Apache CloudStack Documentation

Release 4.11.1.0

Paul Angus

Feb 16, 2019

Contents:

1 CloudStack Concepts and Terminology **3**
- 1.1 Concepts and Terminolgy . . . 3
  - 1.1.1 What is Apache CloudStack? . . . 3
  - 1.1.2 What can Apache CloudStack do? . . . 3
  - 1.1.3 Deployment Architecture Overview . . . 4
  - 1.1.4 CloudStack Terminology . . . 6
- 1.2 Choosing a Deployment Architecture . . . 14
  - 1.2.1 Small-Scale Deployment . . . 15
  - 1.2.2 Large-Scale Redundant Setup . . . 16
  - 1.2.3 Separate Storage Network . . . 17
  - 1.2.4 Multi-Node Management Server . . . 17
  - 1.2.5 Multi-Site Deployment . . . 17
  - 1.2.6 Choosing a Hypervisor . . . 20
  - 1.2.7 Best Practices . . . 22
- 1.3 Network Setup . . . 23
  - 1.3.1 Basic and Advanced Networking . . . 23
  - 1.3.2 VLAN Allocation Example . . . 24
  - 1.3.3 Example Hardware Configuration . . . 24
  - 1.3.4 Layer-2 Switch . . . 25
  - 1.3.5 Hardware Firewall . . . 26
  - 1.3.6 Management Server Load Balancing . . . 34
  - 1.3.7 Topology Requirements . . . 34
  - 1.3.8 Guest Network Usage Integration for Traffic Sentinel . . . 36
  - 1.3.9 Setting Zone VLAN and Running VM Maximums . . . 36
- 1.4 Storage Setup . . . 37
  - 1.4.1 Introduction . . . 37
  - 1.4.2 Configurations . . . 37
  - 1.4.3 Storage Architecture . . . 38
  - 1.4.4 CloudStack Networking For Storage . . . 38

2 Quick Installation Guide **45**
- 2.1 Overview . . . 45
  - 2.1.1 What exactly are we building? . . . 45
  - 2.1.2 High level overview of the process . . . 45
  - 2.1.3 Prerequisites . . . 45
- 2.2 Environment . . . 46

2.2.1 Operating System . . . 46
2.2.2 NFS . . . 48
2.3 Management Server Installation . . . 49
2.3.1 Database Installation and Configuration . . . 49
2.3.2 MySQL connector Installation . . . 50
2.3.3 Installation . . . 50
2.3.4 System Template Setup . . . 50
2.4 KVM Setup and Installation . . . 51
2.4.1 Prerequisites . . . 51
2.4.2 Installation . . . 51
2.4.3 KVM Configuration . . . 51
2.5 Configuration . . . 52
2.5.1 UI Access . . . 52
2.5.2 Setting up a Zone . . . 53
2.5.3 Pod Configuration . . . 53
2.5.4 Cluster . . . 53

3 Installation Guide **55**
3.1 Building from Source . . . 55
3.1.1 Introduction . . . 55
3.1.2 Downloading the release . . . 55
3.1.3 Verifying the downloaded release . . . 56
3.1.4 Prerequisites for building Apache CloudStack . . . 56
3.1.5 Extracting source . . . 57
3.1.6 Install new MySQL connector . . . 57
3.1.7 Building DEB packages . . . 58
3.1.8 Building RPMs from Source . . . 59
3.1.9 Building Non-OSS . . . 61
3.2 General Installation . . . 61
3.2.1 Installation overview . . . 61
3.2.2 Management Server Installation . . . 63
3.3 Configuration . . . 76
3.3.1 Configuring your CloudStack Installation . . . 76
3.4 Hypervisor Setup . . . 101
3.4.1 Host Hyper-V Installation . . . 101
3.4.2 Host KVM Installation . . . 104
3.4.3 Host LXC Installation . . . 119
3.4.4 Host VMware vSphere Installation . . . 127
3.4.5 Host Citrix XenServer Installation . . . 151
3.5 Optional Installation . . . 161
3.5.1 Additional Installation Options . . . 161
3.5.2 About Password and Key Encryption . . . 172

4 Upgrading CloudStack **175**
4.1 Upgrade Instruction from 4.11.0.0 . . . 175
4.1.1 Update System-VM templates . . . 176
4.1.2 Packages repository . . . 178
4.1.3 Database Preparation . . . 178
4.1.4 Management Server on Ubuntu . . . 178
4.1.5 Java 8 JRE on Ubuntu . . . 179
4.1.6 Management Server on CentOS/RHEL . . . 179
4.1.7 Hypervisor: XenServer . . . 181
4.1.8 Hypervisor: VMware . . . 181
4.1.9 Hypervisor: KVM . . . 182

4.1.10 Restart management services . . . 183
4.2 Upgrade Instruction from 4.10.x . . . 183
4.2.1 Update System-VM templates . . . 183
4.2.2 Packages repository . . . 185
4.2.3 Database Preparation . . . 185
4.2.4 Management Server on Ubuntu . . . 185
4.2.5 Java 8 JRE on Ubuntu . . . 186
4.2.6 Management Server on CentOS/RHEL . . . 186
4.2.7 Hypervisor: XenServer . . . 188
4.2.8 Hypervisor: VMware . . . 188
4.2.9 Hypervisor: KVM . . . 189
4.2.10 Restart management services . . . 190
4.3 Upgrade Instruction from 4.9.x . . . 190
4.3.1 Update System-VM templates . . . 190
4.3.2 Packages repository . . . 192
4.3.3 Database Preparation . . . 192
4.3.4 Management Server on Ubuntu . . . 192
4.3.5 Java 8 JRE on Ubuntu . . . 193
4.3.6 Management Server on CentOS/RHEL . . . 193
4.3.7 Hypervisor: XenServer . . . 195
4.3.8 Hypervisor: VMware . . . 195
4.3.9 Hypervisor: KVM . . . 196
4.3.10 Restart management services . . . 197
4.4 Upgrade Instruction from 4.8.x . . . 197
4.4.1 Update System-VM templates . . . 197
4.4.2 Packages repository . . . 199
4.4.3 Database Preparation . . . 199
4.4.4 Management Server on Ubuntu . . . 199
4.4.5 Java 8 JRE on Ubuntu . . . 200
4.4.6 Management Server on CentOS/RHEL . . . 200
4.4.7 Hypervisor: XenServer . . . 202
4.4.8 Hypervisor: VMware . . . 202
4.4.9 Hypervisor: KVM . . . 203
4.4.10 Restart management services . . . 204
4.5 Upgrade Instruction from 4.7.x . . . 204
4.5.1 Packages repository . . . 204
4.5.2 Update System-VM templates . . . 204
4.5.3 Database Preparation . . . 207
4.5.4 Management Server on Ubuntu . . . 207
4.5.5 Java 8 JRE on Ubuntu . . . 207
4.5.6 Management Server on CentOS/RHEL . . . 208
4.5.7 Hypervisor: XenServer . . . 209
4.5.8 Hypervisor: VMware . . . 210
4.5.9 Hypervisor: KVM . . . 210
4.5.10 Restart management services . . . 211
4.6 Upgrade Instruction from 4.6.x . . . 212
4.6.1 Packages repository . . . 212
4.6.2 Update System-VM templates . . . 212
4.6.3 Database Preparation . . . 214
4.6.4 Management Server on Ubuntu . . . 214
4.6.5 Java 8 JRE on Ubuntu . . . 214
4.6.6 Management Server on CentOS/RHEL . . . 215
4.6.7 Hypervisor: XenServer . . . 216
4.6.8 Hypervisor: VMware . . . 217

4.6.9 Hypervisor: KVM . . . 218
4.6.10 Restart management services . . . 218
4.7 Upgrade Instruction from 4.5.x . . . 219
4.7.1 Packages repository . . . 219
4.7.2 Update System-VM templates . . . 219
4.7.3 Database Preparation . . . 221
4.7.4 Management Server on Ubuntu . . . 221
4.7.5 Java 8 JRE on Ubuntu . . . 221
4.7.6 Management Server on CentOS/RHEL . . . 222
4.7.7 Hypervisor: XenServer . . . 223
4.7.8 Hypervisor: VMware . . . 224
4.7.9 Hypervisor: KVM . . . 225
4.7.10 Restart management services . . . 225
4.7.11 System-VMs and Virtual-Routers . . . 226
4.8 Upgrade Instruction from 4.4.x . . . 226
4.8.1 Packages repository . . . 227
4.8.2 Update System-VM templates . . . 227
4.8.3 Database Preparation . . . 229
4.8.4 Management Server on Ubuntu . . . 229
4.8.5 Java 8 JRE on Ubuntu . . . 229
4.8.6 Management Server on CentOS/RHEL . . . 230
4.8.7 Hypervisor: XenServer . . . 231
4.8.8 Hypervisor: VMware . . . 232
4.8.9 Hypervisor: KVM . . . 233
4.8.10 Restart management services . . . 234
4.8.11 System-VMs and Virtual-Routers . . . 234
4.9 Upgrade Instruction from 4.3.x . . . 235
4.9.1 Packages repository . . . 235
4.9.2 Update System-VM templates . . . 235
4.9.3 Database Preparation . . . 238
4.9.4 Management Server on Ubuntu . . . 238
4.9.5 Java 8 JRE on Ubuntu . . . 238
4.9.6 Management Server on CentOS/RHEL . . . 239
4.9.7 Hypervisor: XenServer . . . 240
4.9.8 Hypervisor: VMware . . . 241
4.9.9 Hypervisor: KVM . . . 242
4.9.10 Restart management services . . . 243
4.9.11 System-VMs and Virtual-Routers . . . 243

5 Usage Guide **245**
5.1 User Interface . . . 245
5.1.1 Log In to the UI . . . 245
5.2 Managing Accounts, Users and Domains . . . 247
5.2.1 Roles, Accounts, Users, and Domains . . . 247
5.2.2 Using Dynamic Roles . . . 248
5.2.3 Dedicating Resources to Accounts and Domains . . . 249
5.2.4 How to Dedicate a Zone, Cluster, Pod, or Host to an Account or Domain . . . 250
5.2.5 Using an LDAP Server for User Authentication . . . 250
5.2.6 Using a SAML 2.0 Identity Provider for User Authentication . . . 252
5.3 Using Projects to Organize User Resources . . . 253
5.3.1 Overview of Projects . . . 253
5.3.2 Configuring Projects . . . 254
5.3.3 Creating a New Project . . . 256
5.3.4 Adding Members to a Project . . . 256

5.3.5 Accepting a Membership Invitation . . . 257
5.3.6 Suspending or Deleting a Project . . . 257
5.3.7 Using the Project View . . . 258
5.4 Service Offerings . . . 258
5.4.1 Service Offerings, Disk Offerings, Network Offerings, and Templates . . . 258
5.4.2 Compute and Disk Service Offerings . . . 259
5.4.3 System Service Offerings . . . 263
5.4.4 Network Throttling . . . 264
5.4.5 Changing the Default System Offering for System VMs . . . 265
5.5 Setting up Networking for Users . . . 266
5.5.1 Overview of Setting Up Networking for Users . . . 266
5.5.2 About Virtual Networks . . . 267
5.5.3 Network Service Providers . . . 267
5.5.4 Network Offerings . . . 268
5.5.5 Configuring AutoScale without using NetScaler . . . 271
5.6 Working with Virtual Machines . . . 275
5.6.1 Working with Virtual Machines . . . 275
5.7 Working with Templates . . . 295
5.7.1 Creating Templates: Overview . . . 295
5.7.2 Requirements for Templates . . . 295
5.7.3 Best Practices for Templates . . . 295
5.7.4 The Default Template . . . 295
5.7.5 Private and Public Templates . . . 296
5.7.6 Creating a Template from an Existing Virtual Machine . . . 296
5.7.7 Creating a Template from a Snapshot . . . 297
5.7.8 Uploading Templates . . . 297
5.7.9 vSphere Templates and ISOs . . . 297
5.7.10 Exporting Templates . . . 298
5.7.11 Creating a Linux Template . . . 298
5.7.12 Creating a Windows Template . . . 302
5.7.13 Importing Amazon Machine Images . . . 306
5.7.14 Converting a Hyper-V VM to a Template . . . 309
5.7.15 Adding Password Management to Your Templates . . . 310
5.7.16 Deleting Templates . . . 311
5.8 Working with Hosts . . . 311
5.8.1 Adding Hosts . . . 311
5.8.2 Scheduled Maintenance and Maintenance Mode for Hosts . . . 311
5.8.3 Disabling and Enabling Zones, Pods, and Clusters . . . 312
5.8.4 Removing Hosts . . . 312
5.8.5 Re-Installing Hosts . . . 313
5.8.6 Maintaining Hypervisors on Hosts . . . 313
5.8.7 Changing Host Password . . . 313
5.8.8 Over-Provisioning and Service Offering Limits . . . 314
5.8.9 VLAN Provisioning . . . 317
5.8.10 Out-of-band Management . . . 318
5.8.11 Security . . . 319
5.8.12 Server Address Usage . . . 320
5.8.13 Securing Process . . . 320
5.9 Working with Storage . . . 321
5.9.1 Storage Overview . . . 321
5.9.2 Primary Storage . . . 321
5.9.3 Secondary Storage . . . 323
5.9.4 Working With Volumes . . . 323
5.9.5 Working with Volume Snapshots . . . 329

5.10 Working with System Virtual Machines . . . 331
5.10.1 The System VM Template . . . 331
5.10.2 Changing the Default System VM Template . . . 332
5.10.3 Multiple System VM Support for VMware . . . 332
5.10.4 Console Proxy . . . 332
5.10.5 Virtual Router . . . 335
5.10.6 Secondary Storage VM . . . 338
5.11 Working with Usage . . . 339
5.11.1 Working with Usage . . . 339
5.12 Managing Networks and Traffic . . . 350
5.12.1 Guest Traffic . . . 350
5.12.2 Networking in a Pod . . . 351
5.12.3 Networking in a Zone . . . 352
5.12.4 Basic Zone Physical Network Configuration . . . 353
5.12.5 Advanced Zone Physical Network Configuration . . . 353
5.12.6 Using Multiple Guest Networks . . . 356
5.12.7 IP Reservation in Isolated Guest Networks . . . 358
5.12.8 Reserving Public IP Addresses and VLANs for Accounts . . . 360
5.12.9 Configuring Multiple IP Addresses on a Single NIC . . . 361
5.12.10 About Multiple IP Ranges . . . 362
5.12.11 About Elastic IPs . . . 363
5.12.12 Portable IPs . . . 364
5.12.13 Multiple Subnets in Shared Network . . . 366
5.12.14 Isolation in Advanced Zone Using Private VLAN . . . 367
5.12.15 Security Groups . . . 370
5.12.16 External Firewalls and Load Balancers . . . 372
5.12.17 Global Server Load Balancing Support . . . 381
5.12.18 Guest IP Ranges . . . 388
5.12.19 Acquiring a New IP Address . . . 388
5.12.20 Releasing an IP Address . . . 388
5.12.21 Static NAT . . . 389
5.12.22 IP Forwarding and Firewalling . . . 389
5.12.23 IP Load Balancing . . . 392
5.12.24 DNS and DHCP . . . 393
5.12.25 Remote Access VPN . . . 393
5.12.26 About Inter-VLAN Routing (nTier Apps) . . . 402
5.12.27 Configuring a Virtual Private Cloud . . . 403
5.12.28 Persistent Networks . . . 424
5.12.29 Setup a Palo Alto Networks Firewall . . . 425
5.12.30 Using Remote Access VPN . . . 431
5.13 Managing the Cloud . . . 439
5.13.1 Using Tags to Organize Resources in the Cloud . . . 439
5.13.2 Reporting CPU Sockets . . . 440
5.13.3 Changing the Database Configuration . . . 440
5.13.4 Changing the Database Password . . . 440
5.13.5 File encryption type . . . 441
5.13.6 Administrator Alerts . . . 441
5.13.7 Customizing the Network Domain Name . . . 444
5.13.8 Stopping and Restarting the Management Server . . . 445
5.14 System Reliability and Availability . . . 445
5.14.1 HA for Management Server . . . 445
5.14.2 Management Server Load Balancing . . . 446
5.14.3 HA-Enabled Virtual Machines . . . 446
5.14.4 HA for Hosts . . . 446

5.14.5 Primary Storage Outage and Data Loss . . . 447
5.14.6 Secondary Storage Outage and Data Loss . . . 447
5.14.7 Database High Availability . . . 447
5.15 Tuning . . . 449
5.15.1 Tuning . . . 449
5.16 Events and Troubleshooting . . . 450
5.16.1 Event Notification . . . 450
5.16.2 TroubleShooting . . . 455

6 Developers Guide **465**
6.1 CloudStack Installation from GIT repo for Developers . . . 465
6.1.1 Prerequisites . . . 465
6.1.2 Installing from Source . . . 468
6.1.3 Using the Simulator . . . 469
6.1.4 Using DevCloud . . . 469
6.1.5 Building Packages . . . 470
6.1.6 The CloudStack API . . . 471
6.1.7 Testing the AWS API interface . . . 473
6.1.8 Conclusions . . . 474
6.2 Programmer Guide . . . 474
6.2.1 The CloudStack API . . . 474
6.2.2 Event Types . . . 484
6.2.3 Time Zones . . . 490
6.3 Plugins . . . 491
6.3.1 Storage Plugins . . . 491
6.3.2 Third Party UI Plugins . . . 496
6.4 Allocators . . . 503
6.4.1 Implementing a custom HostAllocator . . . 503
6.4.2 Implementing a custom StoragePoolAllocator . . . 505
6.5 Deploying CloudStack with Ansible . . . 507
6.5.1 What is Ansible . . . 507
6.5.2 There's already Chef and Puppet, so what's the fuss about Ansible? . . . 507
6.5.3 So let's see something . . . 508
6.5.4 Installing Ansible . . . 508
6.5.5 Playbooks . . . 508
6.5.6 Modules . . . 508
6.5.7 Planning . . . 509
6.5.8 MySQL . . . 509
6.5.9 CloudStack Management server service . . . 510
6.5.10 System VM Templates: . . . 512
6.5.11 Bringing it all together . . . 512
6.5.12 How is this example different from a production deployment? . . . 513
6.5.13 Acknowledgements . . . 513
6.6 Getting Help . . . 513
6.6.1 Documentation Available . . . 514
6.6.2 Books . . . 514
6.6.3 Commercial support . . . 515

7 Plugins Guide **517**
7.1 The Cloudian Connector Plugin . . . 517
7.1.1 Introduction to the Cloudian Connector Plugin . . . 517
7.1.2 Connector Overview . . . 518
7.1.3 Configuring the Cloudian Connector . . . 519
7.1.4 Cloudian as CloudStack Secondary Storage . . . 521

7.1.5 Adding Cloudian as CloudStack Secondary Storage . . . 522
7.1.6 Revision History . . . 525
7.2 The Nicira NVP Plugin . . . 525
7.2.1 Introduction to the Nicira NVP Plugin . . . 525
7.2.2 Configuring the Nicira NVP Plugin . . . 526
7.2.3 Using the Nicira NVP plugin with VPC . . . 528
7.2.4 Troubleshooting the Nicira NVP Plugin . . . 530
7.2.5 Revision History . . . 531
7.3 The Nuage VSP Plugin . . . 531
7.3.1 Introduction . . . 531
7.3.2 Configuring The Nuage VSP Plugin . . . 532
7.3.3 Using The Nuage VSP Plugin . . . 538
7.3.4 Dedicated Features Provided by The Nuage VSP Plugin . . . 541
7.3.5 Running The Nuage VSP Plugin Specific Marvin Tests . . . 544
7.3.6 Appendix . . . 545
7.4 The VXLAN Plugin . . . 547
7.4.1 System Requirements for VXLAN . . . 547
7.4.2 Linux Distributions that meet the requirements . . . 548
7.4.3 Configure PRODUCT to use VXLAN Plugin . . . 550
7.5 The OVS Plugin . . . 553
7.5.1 Introduction to the OVS Plugin . . . 553
7.5.2 Configuring the OVS Plugin . . . 557
7.5.3 Using the OVS plugin with VPC . . . 560
7.5.4 Revision History . . . 560
7.6 IPv6 Support in CloudStack . . . 560
7.6.1 Prerequisites and Guidelines . . . 562
7.6.2 Limitations of IPv6 in CloudStack . . . 562
7.6.3 Guest VM Configuration for DHCPv6 . . . 563
7.7 Quota Plugin . . . 564
7.7.1 Enabling the Quota Service . . . 564
7.7.2 Quota Tariff . . . 565
7.7.3 Quota Credits . . . 565
7.7.4 Quota Balance . . . 566
7.7.5 Quota Statement . . . 566
7.7.6 Quota Monthly Statement . . . 566
7.7.7 Quota Alert Management . . . 566

8 Release Notes **567**
8.1 What's New in 4.11.1.0 . . . 567
8.1.1 What's New in 4.11.1.0 . . . 567
8.1.2 What's New in 4.11.0.0 . . . 567
8.2 Issues Fixed in 4.11.1.0 . . . 569
8.2.1 Issues Fixed in 4.11.1.0 . . . 569
8.2.2 Issues Fixed in 4.11.0.0 . . . 572
8.3 Compatibility Matrix . . . 577
8.3.1 Supported OS Versions for Management Server . . . 577
8.3.2 Supported Hypervisor Versions . . . 577
8.3.3 Supported External Devices . . . 578
8.3.4 Supported Browsers . . . 578
8.3.5 Notice Of Management OSes and Hypervisors to be Deprecated . . . 579
8.4 API Changes Introduced in 4.11.1.0 . . . 579
8.4.1 New API Commands . . . 579
8.4.2 Parameters Changed API Commands . . . 580
8.5 Known Issues in 4.11 . . . 592

9 Indices and Tables 601

We have a number of guides, starting with a guide to cloudstack's terminology and concepts, moving through some information about possible topologies. We then have a quick start guide to help you get a very simple cloudstack up and running. Followed by the full installation guide, an administrator's guide and then further detailed guides on complex configurations.

Information can also be found at CloudStack's wiki https://cwiki.apache.org/confluence/display/CLOUDSTACK/Home and on cloudstack mailing lists http://cloudstack.apache.org/mailing-lists.html

CHAPTER 1

CloudStack Concepts and Terminology

This is the Apache CloudStack installation guide. In this guide we first go through some design and architectural to build your cloud.

1.1 Concepts and Terminolgy

1.1.1 What is Apache CloudStack?

Apache CloudStack is an open source Infrastructure-as-a-Service platform that manages and orchestrates pools of storage, network, and computer resources to build a public or private IaaS compute cloud.

With CloudStack you can:

- Set up an on-demand elastic cloud computing service.
- Allow end-users to provision resources

1.1.2 What can Apache CloudStack do?

Multiple Hypervisor Support

CloudStack works with a variety of hypervisors and hypervisor-like technologies. A single cloud can contain multiple hypervisor implementations. As of the current release CloudStack supports:

- BareMetal (via IPMI)
- Hyper-V
- KVM
- LXC
- vSphere (via vCenter)
- Xenserver

- Xen Project

Massively Scalable Infrastructure Management

CloudStack can manage tens of thousands of physical servers installed in geographically distributed datacenters. The management server scales near-linearly eliminating the need for cluster-level management servers. Maintenance or other outages of the management server can occur without affecting the virtual machines running in the cloud.

Automatic Cloud Configuration Management

CloudStack automatically configures the network and storage settings for each virtual machine deployment. Internally, a pool of virtual appliances support the operation of configuration of the cloud itself. These appliances offer services such as firewalling, routing, DHCP, VPN, console proxy, storage access, and storage replication. The extensive use of horizontally scalable virtual machines simplifies the installation and ongoing operation of a cloud.

Graphical User Interface

CloudStack offers an administrators web interface used for provisioning and managing the cloud, as well as an end-user's Web interface, used for running VMs and managing VM templates. The UI can be customized to reflect the desired service provider or enterprise look and feel.

API

CloudStack provides a REST-like API for the operation, management and use of the cloud.

AWS EC2 API Support

CloudStack provides an EC2 API translation layer to permit the common EC2 tools to be used in the use of a CloudStack cloud.

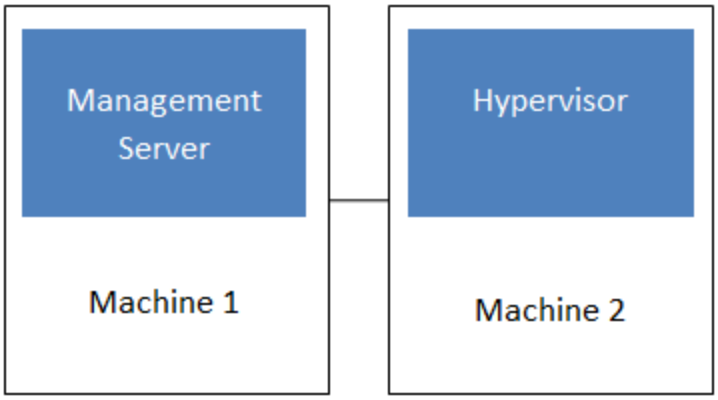
High Availability

CloudStack has a number of features to increase the availability of the system. The Management Server itself may be deployed in a multi-node installation where the servers are load balanced. MySQL may be configured to use replication to provide for failover in the event of database loss. For the hosts, CloudStack supports NIC bonding and the use of separate networks for storage as well as iSCSI Multipath.

1.1.3 Deployment Architecture Overview

Generally speaking, most CloudStack deployments consist of the management server and the resources to be managed. During deployment you inform the management server of the resources to be managed, such as IP address blocks, storage devices, hypervisors, and VLANs.

The minimum installation consists of one machine running the CloudStack Management Server and another machine to act as the cloud infrastructure (in this case, a very simple infrastructure consisting of one host running hypervisor software). In its smallest deployment, a single machine can act as both the Management Server and the hypervisor host (using the KVM hypervisor).



Simplified view of a basic deployment

A more full-featured installation consists of a highly-available multi-node Management Server installation and up to tens of thousands of hosts using any of several networking technologies.

Management Server Overview

The management server orchestrates and allocates the resources in your cloud deployment.

The management server typically runs on a dedicated machine or as a virtual machine. It controls allocation of virtual machines to hosts and assigns storage and IP addresses to the virtual machine instances. The Management Server runs in an Apache Tomcat container and requires a MySQL database for persistence.

The management server:

- Provides the web interface for both the adminstrator and end user.
- Provides the API interfaces for both the CloudStack API as well as the EC2 interface.
- Manages the assignment of guest VMs to a specific compute resource
- Manages the assignment of public and private IP addresses.
- Allocates storage during the VM instantiation process.
- Manages snapshots, disk images (templates), and ISO images.
- Provides a single point of configuration for your cloud.

Cloud Infrastructure Overview

Resources within the cloud are managed as follows:

- Regions: A collection of one or more geographically proximate zones managed by one or more management servers.
- Zones: Typically, a zone is equivalent to a single datacenter. A zone consists of one or more pods and secondary storage.
- Pods: A pod is usually a rack, or row of racks that includes a layer-2 switch and one or more clusters.
- Clusters: A cluster consists of one or more homogenous hosts and primary storage.
- Host: A single compute node within a cluster; often a hypervisor.

- Primary Storage: A storage resource typically provided to a single cluster for the actual running of instance disk images. (Zone-wide primary storage is an option, though not typically used.)
- Secondary Storage: A zone-wide resource which stores disk templates, ISO images, and snapshots.

Networking Overview

CloudStack offers many types of networking, but they typically fall into one of two scenarios:

- Basic: Most analogous to AWS-classic style networking. Provides a single flat layer-2 network where guest isolation is provided at layer-3 by the hypervisors bridge device.
- Advanced: This typically uses layer-2 isolation such as VLANs, though this category also includes SDN technologies such as Nicira NVP.

1.1.4 CloudStack Terminology

About Regions

To increase reliability of the cloud, you can optionally group resources into multiple geographic regions. A region is the largest available organizational unit within a CloudStack deployment. A region is made up of several availability zones, where each zone is roughly equivalent to a datacenter. Each region is controlled by its own cluster of Management Servers, running in one of the zones. The zones in a region are typically located in close geographical proximity. Regions are a useful technique for providing fault tolerance and disaster recovery.

By grouping zones into regions, the cloud can achieve higher availability and scalability. User accounts can span regions, so that users can deploy VMs in multiple, widely-dispersed regions. Even if one of the regions becomes unavailable, the services are still available to the end-user through VMs deployed in another region. And by grouping communities of zones under their own nearby Management Servers, the latency of communications within the cloud is reduced compared to managing widely-dispersed zones from a single central Management Server.

Usage records can also be consolidated and tracked at the region level, creating reports or invoices for each geographic region.

Regions are visible to the end user. When a user starts a guest VM on a particular CloudStack Management Server, the user is implicitly selecting that region for their guest. Users might also be required to copy their private templates to additional regions to enable creation of guest VMs using their templates in those regions.

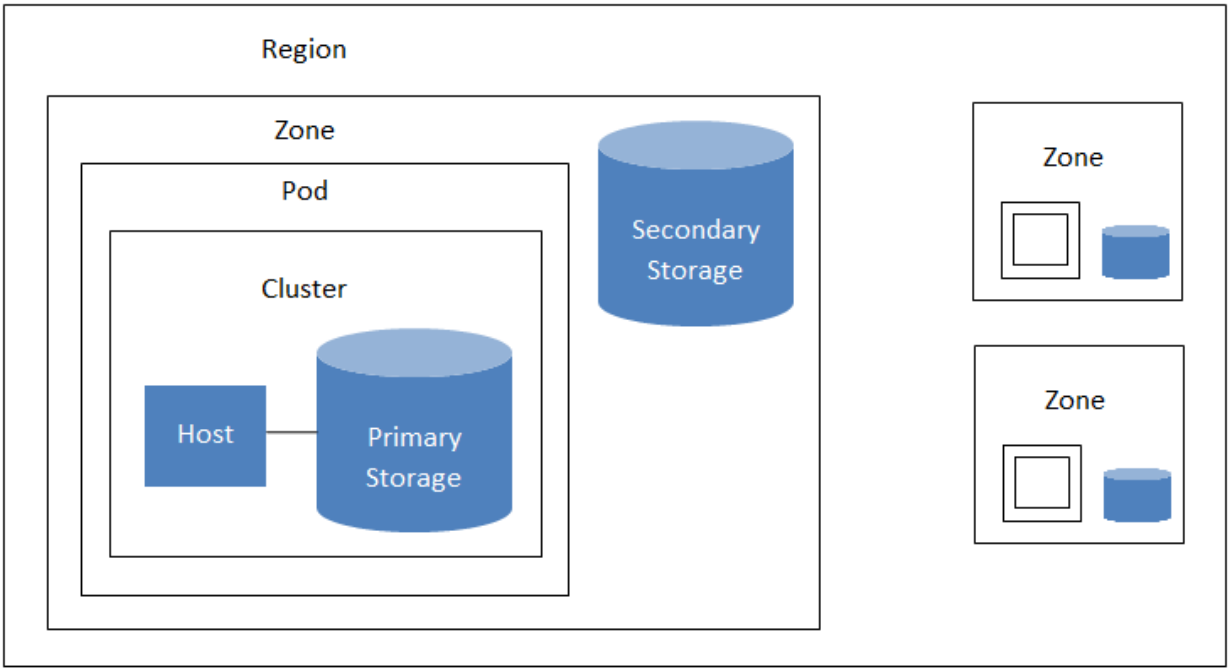
About Zones

A zone is the second largest organizational unit within a CloudStack deployment. A zone typically corresponds to a single datacenter, although it is permissible to have multiple zones in a datacenter. The benefit of organizing infrastructure into zones is to provide physical isolation and redundancy. For example, each zone can have its own power supply and network uplink, and the zones can be widely separated geographically (though this is not required).

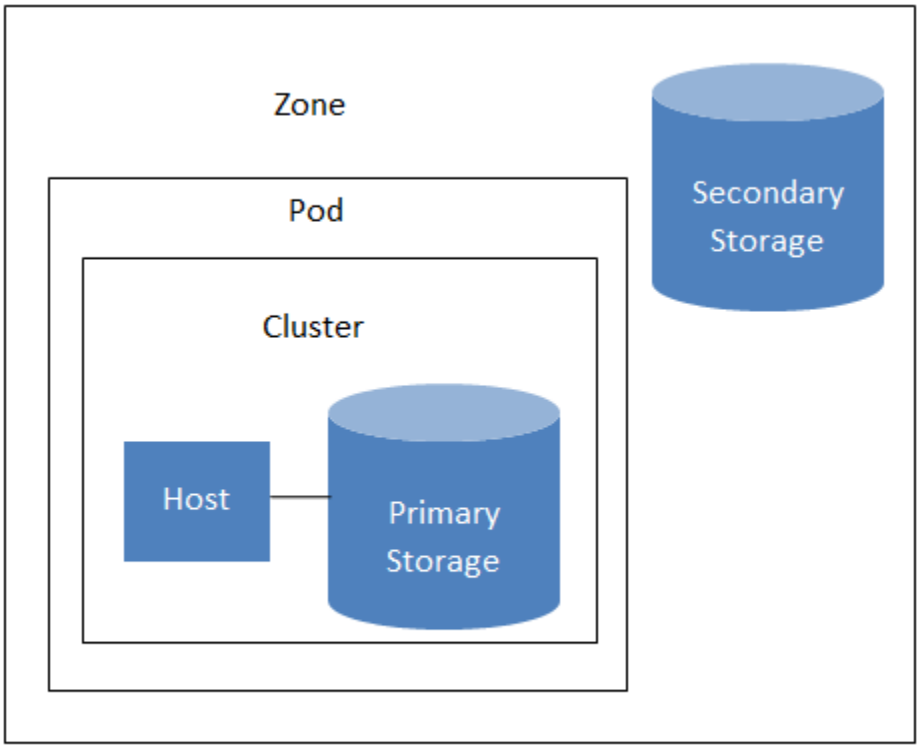
A zone consists of:

- One or more pods. Each pod contains one or more clusters of hosts and one or more primary storage servers.
- A zone may contain one or more primary storage servers, which are shared by all the pods in the zone.
- Secondary storage, which is shared by all the pods in the zone.

Zones are visible to the end user. When a user starts a guest VM, the user must select a zone for their guest. Users might also be required to copy their private templates to additional zones to enable creation of guest VMs using their templates in those zones.



A region with multiple zones



Nested organization of a zone

Zones can be public or private. Public zones are visible to all users. This means that any user may create a guest in that zone. Private zones are reserved for a specific domain. Only users in that domain or its subdomains may create guests in that zone.

Hosts in the same zone are directly accessible to each other without having to go through a firewall. Hosts in different zones can access each other through statically configured VPN tunnels.

For each zone, the administrator must decide the following.

- How many pods to place in each zone.
- How many clusters to place in each pod.
- How many hosts to place in each cluster.
- (Optional) How many primary storage servers to place in each zone and total capacity for these storage servers.
- How many primary storage servers to place in each cluster and total capacity for these storage servers.
- How much secondary storage to deploy in a zone.

When you add a new zone using the CloudStack UI, you will be prompted to configure the zone's physical network and add the first pod, cluster, host, primary storage, and secondary storage.

In order to support zone-wide functions for VMware, CloudStack is aware of VMware Datacenters and can map each Datacenter to a CloudStack zone. To enable features like storage live migration and zone-wide primary storage for VMware hosts, CloudStack has to make sure that a zone contains only a single VMware Datacenter. Therefore, when you are creating a new CloudStack zone, you can select a VMware Datacenter for the zone. If you are provisioning multiple VMware Datacenters, each one will be set up as a single zone in CloudStack.

Note: If you are upgrading from a previous CloudStack version, and your existing deployment contains a zone with clusters from multiple VMware Datacenters, that zone will not be forcibly migrated to the new model. It will continue to function as before. However, any new zone-wide operations, such as zone-wide primary storage and live storage migration, will not be available in that zone.

About Pods

A pod often represents a single rack. Hosts in the same pod are in the same subnet. A pod is the third-largest organizational unit within a CloudStack deployment. Pods are contained within zones. Each zone can contain one or more pods. A pod consists of one or more clusters of hosts and one or more primary storage servers. Pods are not visible to the end user.

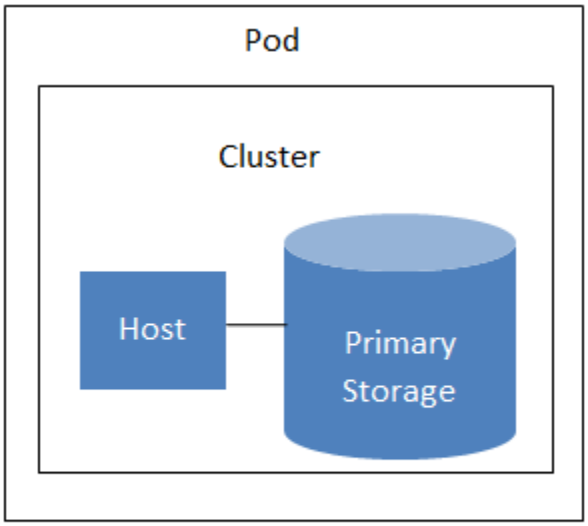
About Clusters

A cluster provides a way to group hosts. To be precise, a cluster is a XenServer server pool, a set of KVM servers, , or a VMware cluster preconfigured in vCenter. The hosts in a cluster all have identical hardware, run the same hypervisor, are on the same subnet, and access the same shared primary storage. Virtual machine instances (VMs) can be live-migrated from one host to another within the same cluster, without interrupting service to the user.

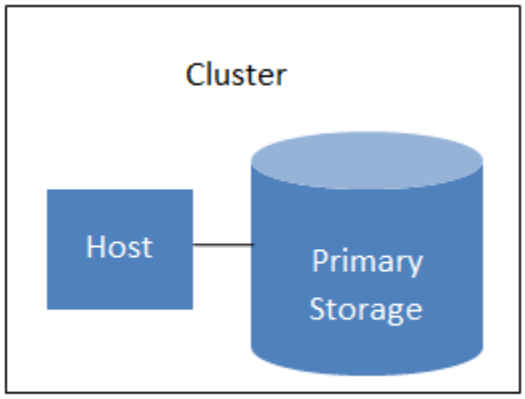
A cluster is the fourth-largest organizational unit within a CloudStack deployment. Clusters are contained within pods, and pods are contained within zones. Size of the cluster is limited by the underlying hypervisor, although the CloudStack recommends less in most cases; see Best Practices.

A cluster consists of one or more hosts and one or more primary storage servers.

CloudStack allows multiple clusters in a cloud deployment.



A simple pod



A simple cluster

Even when local storage is used exclusively, clusters are still required organizationally, even if there is just one host per cluster.

When VMware is used, every VMware cluster is managed by a vCenter server. An Administrator must register the vCenter server with CloudStack. There may be multiple vCenter servers per zone. Each vCenter server may manage multiple VMware clusters.

About Hosts

A host is a single computer. Hosts provide the computing resources that run guest virtual machines. Each host has hypervisor software installed on it to manage the guest VMs. For example, a host can be a Citrix XenServer server, a Linux KVM-enabled server, an ESXi server, or a Windows Hyper-V server.

The host is the smallest organizational unit within a CloudStack deployment. Hosts are contained within clusters, clusters are contained within pods, pods are contained within zones, and zones can be contained within regions.

Hosts in a CloudStack deployment:

- Provide the CPU, memory, storage, and networking resources needed to host the virtual machines
- Interconnect using a high bandwidth TCP/IP network and connect to the Internet
- May reside in multiple data centers across different geographic locations
- May have different capacities (different CPU speeds, different amounts of RAM, etc.), although the hosts within a cluster must all be homogeneous

Additional hosts can be added at any time to provide more capacity for guest VMs.

CloudStack automatically detects the amount of CPU and memory resources provided by the hosts.

Hosts are not visible to the end user. An end user cannot determine which host their guest has been assigned to.

For a host to function in CloudStack, you must do the following:

- Install hypervisor software on the host
- Assign an IP address to the host
- Ensure the host is connected to the CloudStack Management Server.

About Primary Storage

Primary storage is associated with a cluster, and it stores virtual disks for all the VMs running on hosts in that cluster. On KVM and VMware, you can provision primary storage on a per-zone basis.

You can add multiple primary storage servers to a cluster or zone. At least one is required. It is typically located close to the hosts for increased performance. CloudStack manages the allocation of guest virtual disks to particular primary storage devices.

It is useful to set up zone-wide primary storage when you want to avoid extra data copy operations. With cluster-based primary storage, data in the primary storage is directly available only to VMs within that cluster. If a VM in a different cluster needs some of the data, it must be copied from one cluster to another, using the zone's secondary storage as an intermediate step. This operation can be unnecessarily time-consuming.

For Hyper-V, SMB/CIFS storage is supported. Note that Zone-wide Primary Storage is not supported in Hyper-V.

Ceph/RBD storage is only supported by the KVM hypervisor. It can be used as Zone-wide Primary Storage.

CloudStack is designed to work with all standards-compliant iSCSI and NFS servers that are supported by the underlying hypervisor, including, for example:

- SolidFire for iSCSI

- Dell EqualLogic™ for iSCSI
- Network Appliances filers for NFS and iSCSI
- Scale Computing for NFS

If you intend to use only local disk for your installation, you can skip adding separate primary storage.

About Secondary Storage

Secondary storage stores the following:

- Templates — OS images that can be used to boot VMs and can include additional configuration information, such as installed applications
- ISO images — disc images containing data or bootable media for operating systems
- Disk volume snapshots — saved copies of VM data which can be used for data recovery or to create new templates

The items in secondary storage are available to all hosts in the scope of the secondary storage, which may be defined as per zone or per region.

To make items in secondary storage available to all hosts throughout the cloud, you can add object storage in addition to the zone-based NFS Secondary Staging Store. It is not necessary to copy templates and snapshots from one zone to another, as would be required when using zone NFS alone. Everything is available everywhere.

For Hyper-V hosts, SMB/CIFS storage is supported.

CloudStack provides plugins that enable both OpenStack Object Storage (Swift, swift.openstack.org) and Amazon Simple Storage Service (S3) object storage. When using one of these storage plugins, you configure Swift or S3 storage for the entire CloudStack, then set up the NFS Secondary Staging Store for each zone. The NFS storage in each zone acts as a staging area through which all templates and other secondary storage data pass before being forwarded to Swift or S3. The backing object storage acts as a cloud-wide resource, making templates and other data available to any zone in the cloud.

Warning: Heterogeneous Secondary Storage is not supported in Regions. For example, you cannot set up multiple zones, one using NFS secondary and the other using S3 or Swift secondary.

About Physical Networks

Part of adding a zone is setting up the physical network. One or (in an advanced zone) more physical networks can be associated with each zone. The network corresponds to a NIC on the hypervisor host. Each physical network can carry one or more types of network traffic. The choices of traffic type for each network vary depending on whether you are creating a zone with basic networking or advanced networking.

A physical network is the actual network hardware and wiring in a zone. A zone can have multiple physical networks. An administrator can:

- Add/Remove/Update physical networks in a zone
- Configure VLANs on the physical network
- Configure a name so the network can be recognized by hypervisors
- Configure the service providers (firewalls, load balancers, etc.) available on a physical network
- Configure the IP addresses trunked to a physical network
- Specify what type of traffic is carried on the physical network, as well as other properties like network speed

Basic Zone Network Traffic Types

When basic networking is used, there can be only one physical network in the zone. That physical network carries the following traffic types:

- Guest. When end users run VMs, they generate guest traffic. The guest VMs communicate with each other over a network that can be referred to as the guest network. Each pod in a basic zone is a broadcast domain, and therefore each pod has a different IP range for the guest network. The administrator must configure the IP range for each pod.
- Management. When CloudStack's internal resources communicate with each other, they generate management traffic. This includes communication between hosts, system VMs (VMs used by CloudStack to perform various tasks in the cloud), and any other component that communicates directly with the CloudStack Management Server. You must configure the IP range for the system VMs to use.

Note: We strongly recommend the use of separate NICs for management traffic and guest traffic.

- Public. Public traffic is generated when VMs in the cloud access the Internet. Publicly accessible IPs must be allocated for this purpose. End users can use the CloudStack UI to acquire these IPs to implement NAT between their guest network and the public network, as described in Acquiring a New IP Address.
- Storage. While labeled "storage" this is specifically about secondary storage, and doesn't affect traffic for primary storage. This includes traffic such as VM templates and snapshots, which is sent between the secondary storage VM and secondary storage servers. CloudStack uses a separate Network Interface Controller (NIC) named storage NIC for storage network traffic. Use of a storage NIC that always operates on a high bandwidth network allows fast template and snapshot copying. You must configure the IP range to use for the storage network.

In a basic network, configuring the physical network is fairly straightforward. In most cases, you only need to configure one guest network to carry traffic that is generated by guest VMs. If you use a NetScaler load balancer and enable its elastic IP and elastic load balancing (EIP and ELB) features, you must also configure a network to carry public traffic. CloudStack takes care of presenting the necessary network configuration steps to you in the UI when you add a new zone.

Basic Zone Guest IP Addresses

When basic networking is used, CloudStack will assign IP addresses in the CIDR of the pod to the guests in that pod. The administrator must add a Direct IP range on the pod for this purpose. These IPs are in the same VLAN as the hosts.

Advanced Zone Network Traffic Types

When advanced networking is used, there can be multiple physical networks in the zone. Each physical network can carry one or more traffic types, and you need to let CloudStack know which type of network traffic you want each network to carry. The traffic types in an advanced zone are:

- Guest. When end users run VMs, they generate guest traffic. The guest VMs communicate with each other over a network that can be referred to as the guest network. This network can be isolated or shared. In an isolated guest network, the administrator needs to reserve VLAN ranges to provide isolation for each CloudStack account's network (potentially a large number of VLANs). In a shared guest network, all guest VMs share a single network.
- Management. When CloudStack's internal resources communicate with each other, they generate management traffic. This includes communication between hosts, system VMs (VMs used by CloudStack to perform various

tasks in the cloud), and any other component that communicates directly with the CloudStack Management Server. You must configure the IP range for the system VMs to use.

- Public. Public traffic is generated when VMs in the cloud access the Internet. Publicly accessible IPs must be allocated for this purpose. End users can use the CloudStack UI to acquire these IPs to implement NAT between their guest network and the public network, as described in "Acquiring a New IP Address" in the Administration Guide.
- Storage. While labeled "storage" this is specifically about secondary storage, and doesn't affect traffic for primary storage. This includes traffic such as VM templates and snapshots, which is sent between the secondary storage VM and secondary storage servers. CloudStack uses a separate Network Interface Controller (NIC) named storage NIC for storage network traffic. Use of a storage NIC that always operates on a high bandwidth network allows fast template and snapshot copying. You must configure the IP range to use for the storage network.

These traffic types can each be on a separate physical network, or they can be combined with certain restrictions. When you use the Add Zone wizard in the UI to create a new zone, you are guided into making only valid choices.

Advanced Zone Guest IP Addresses

When advanced networking is used, the administrator can create additional networks for use by the guests. These networks can span the zone and be available to all accounts, or they can be scoped to a single account, in which case only the named account may create guests that attach to these networks. The networks are defined by a VLAN ID, IP range, and gateway. The administrator may provision thousands of these networks if desired. Additionally, the administrator can reserve a part of the IP address space for non-CloudStack VMs and servers.

Advanced Zone Public IP Addresses

When advanced networking is used, the administrator can create additional networks for use by the guests. These networks can span the zone and be available to all accounts, or they can be scoped to a single account, in which case only the named account may create guests that attach to these networks. The networks are defined by a VLAN ID, IP range, and gateway. The administrator may provision thousands of these networks if desired.

System Reserved IP Addresses

In each zone, you need to configure a range of reserved IP addresses for the management network. This network carries communication between the CloudStack Management Server and various system VMs, such as Secondary Storage VMs, Console Proxy VMs, and DHCP.

The reserved IP addresses must be unique across the cloud. You cannot, for example, have a host in one zone which has the same private IP address as a host in another zone.

The hosts in a pod are assigned private IP addresses. These are typically RFC1918 addresses. The Console Proxy and Secondary Storage system VMs are also allocated private IP addresses in the CIDR of the pod that they are created in.

Make sure computing servers and Management Servers use IP addresses outside of the System Reserved IP range. For example, suppose the System Reserved IP range starts at 192.168.154.2 and ends at 192.168.154.7. CloudStack can use .2 to .7 for System VMs. This leaves the rest of the pod CIDR, from .8 to .254, for the Management Server and hypervisor hosts.

In all zones:

Provide private IPs for the system in each pod and provision them in CloudStack.

For KVM and XenServer, the recommended number of private IPs per pod is one per host. If you expect a pod to grow, add enough private IPs now to accommodate the growth.

In a zone that uses advanced networking:

For zones with advanced networking, we recommend provisioning enough private IPs for your total number of customers, plus enough for the required CloudStack System VMs. Typically, about 10 additional IPs are required for the System VMs. For more information about System VMs, see the section on working with SystemVMs in the Administrator's Guide.

When advanced networking is being used, the number of private IP addresses available in each pod varies depending on which hypervisor is running on the nodes in that pod. Citrix XenServer and KVM use link-local addresses, which in theory provide more than 65,000 private IP addresses within the address block. As the pod grows over time, this should be more than enough for any reasonable number of hosts as well as IP addresses for guest virtual routers. VMWare ESXi, by contrast uses any administrator-specified subnetting scheme, and the typical administrator provides only 255 IPs per pod. Since these are shared by physical machines, the guest virtual router, and other entities, it is possible to run out of private IPs when scaling up a pod whose nodes are running ESXi.

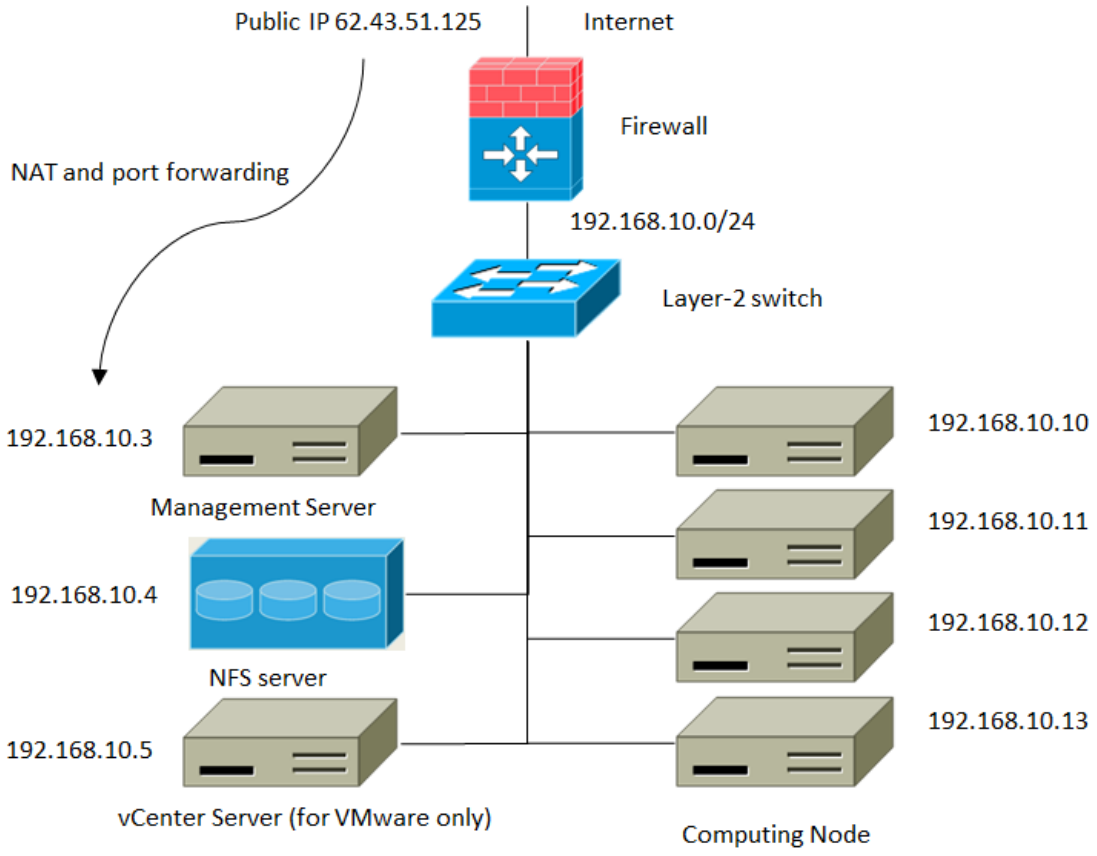
To ensure adequate headroom to scale private IP space in an ESXi pod that uses advanced networking, use one or both of the following techniques:

- Specify a larger CIDR block for the subnet. A subnet mask with a /20 suffix will provide more than 4,000 IP addresses.
- Create multiple pods, each with its own subnet. For example, if you create 10 pods and each pod has 255 IPs, this will provide 2,550 IP addresses.

1.2 Choosing a Deployment Architecture

The architecture used in a deployment will vary depending on the size and purpose of the deployment. This section contains examples of deployment architecture, including a small-scale deployment useful for test and trial deployments and a fully-redundant large-scale setup for production deployments.

1.2.1 Small-Scale Deployment

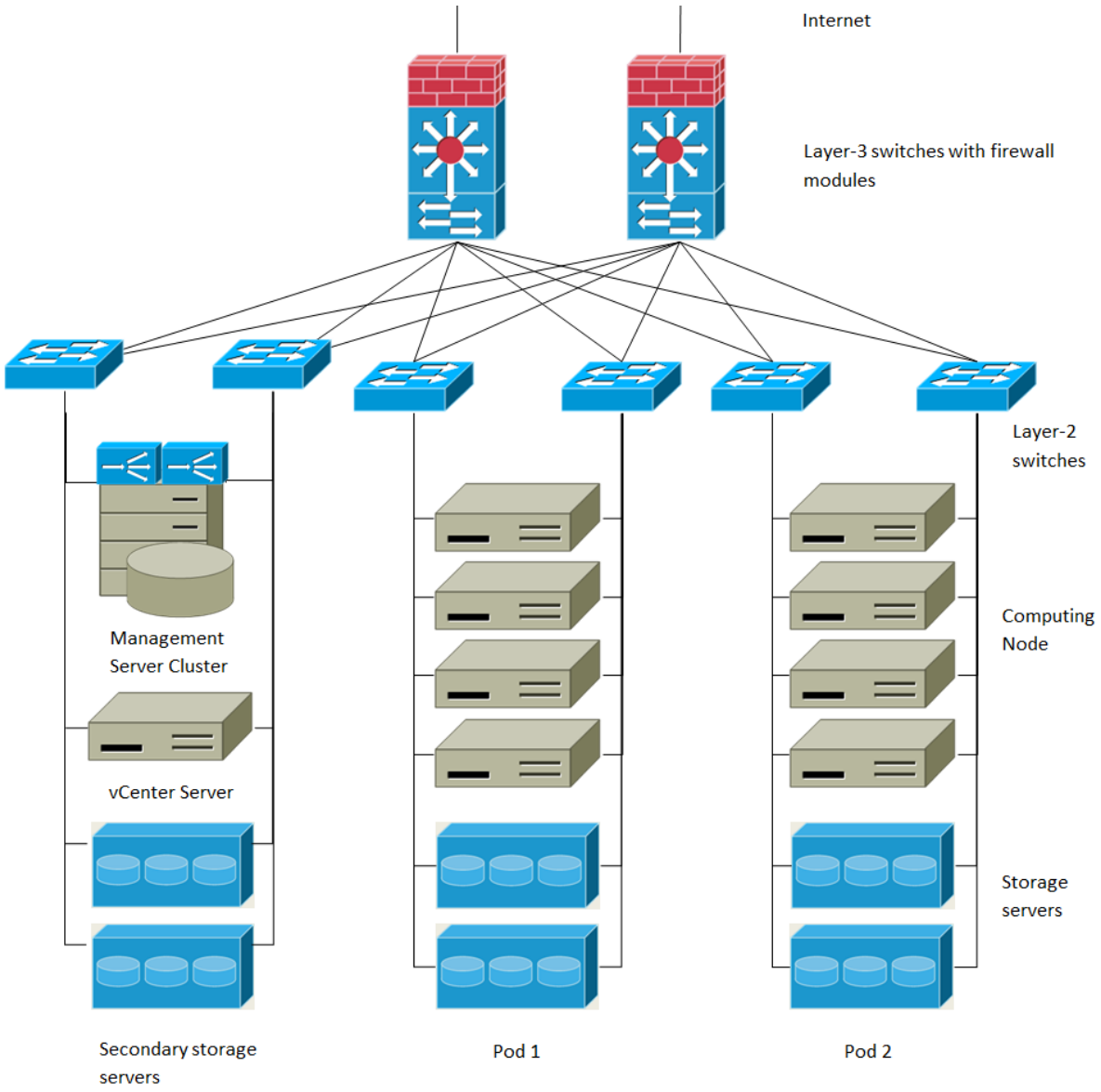


Small-Scale Deployment

This diagram illustrates the network architecture of a small-scale CloudStack deployment.

- A firewall provides a connection to the Internet. The firewall is configured in NAT mode. The firewall forwards HTTP requests and API calls from the Internet to the Management Server. The Management Server resides on the management network.
- A layer-2 switch connects all physical servers and storage.
- A single NFS server functions as both the primary and secondary storage.
- The Management Server is connected to the management network.

1.2.2 Large-Scale Redundant Setup



Large-Scale Redundant Deployment

This diagram illustrates the network architecture of a large-scale CloudStack deployment.

- A layer-3 switching layer is at the core of the data center. A router redundancy protocol like VRRP should be deployed. Typically high-end core switches also include firewall modules. Separate firewall appliances may also be used if the layer-3 switch does not have integrated firewall capabilities. The firewalls are configured in NAT mode. The firewalls provide the following functions:
  - Forwards HTTP requests and API calls from the Internet to the Management Server. The Management Server resides on the management network.
  - When the cloud spans multiple zones, the firewalls should enable site-to-site VPN such that servers in different zones can directly reach each other.

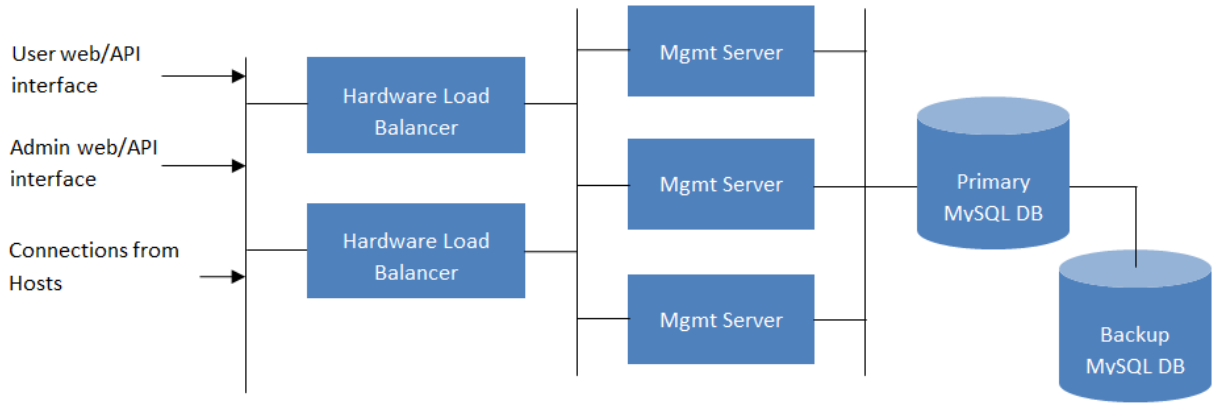
- A layer-2 access switch layer is established for each pod. Multiple switches can be stacked to increase port count. In either case, redundant pairs of layer-2 switches should be deployed.
- The Management Server cluster (including front-end load balancers, Management Server nodes, and the MySQL database) is connected to the management network through a pair of load balancers.
- Secondary storage servers are connected to the management network.
- Each pod contains storage and computing servers. Each storage and computing server should have redundant NICs connected to separate layer-2 access switches.

1.2.3 Separate Storage Network

In the large-scale redundant setup described in the previous section, storage traffic can overload the management network. A separate storage network is optional for deployments. Storage protocols such as iSCSI are sensitive to network delays. A separate storage network ensures guest network traffic contention does not impact storage performance.

1.2.4 Multi-Node Management Server

The CloudStack Management Server is deployed on one or more front-end servers connected to a single MySQL database. Optionally a pair of hardware load balancers distributes requests from the web. A backup management server set may be deployed using MySQL replication at a remote site to add DR capabilities.



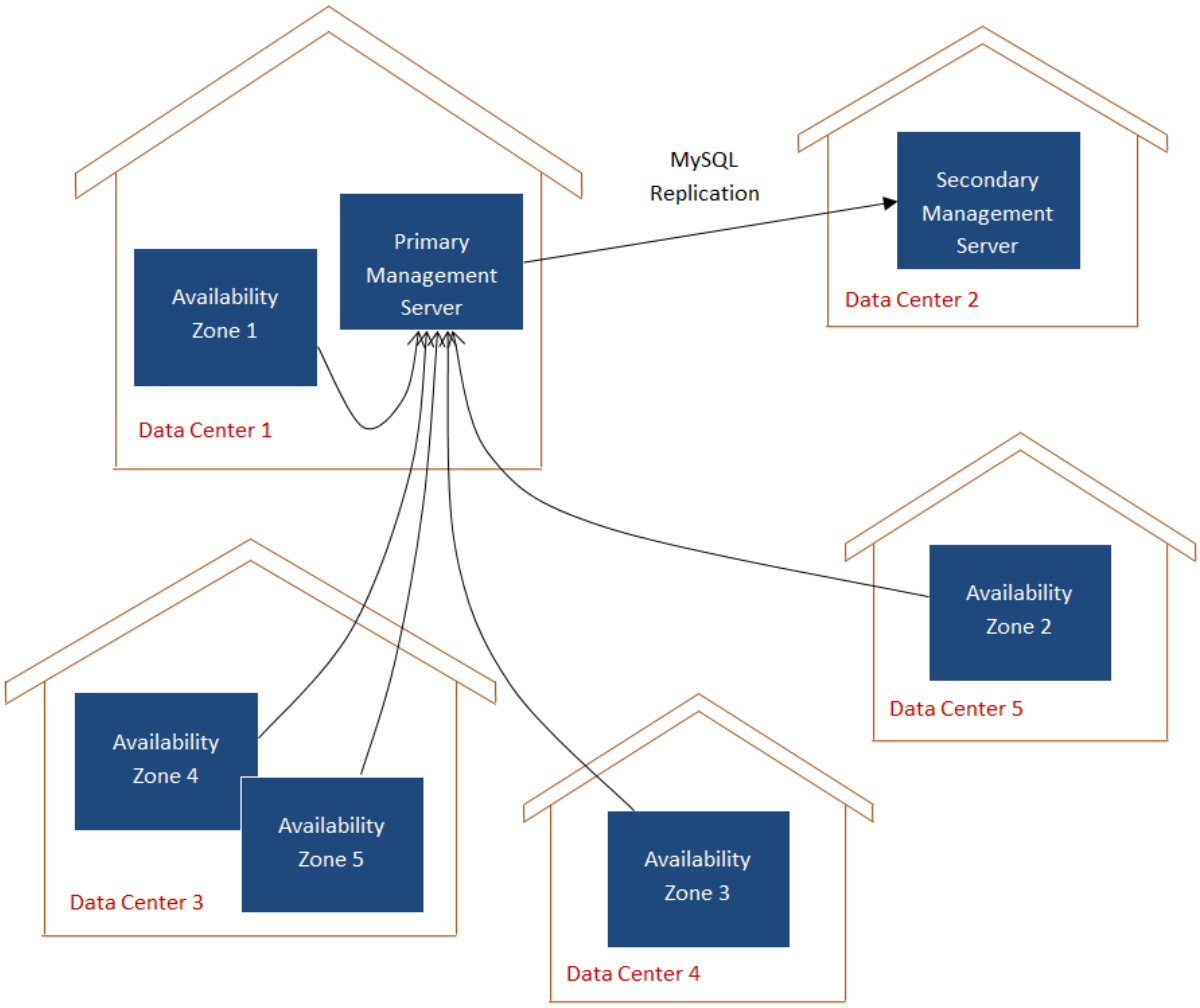
Multi-Node Management Server Deployment

The administrator must decide the following.

- Whether or not load balancers will be used.
- How many Management Servers will be deployed.
- Whether MySQL replication will be deployed to enable disaster recovery.

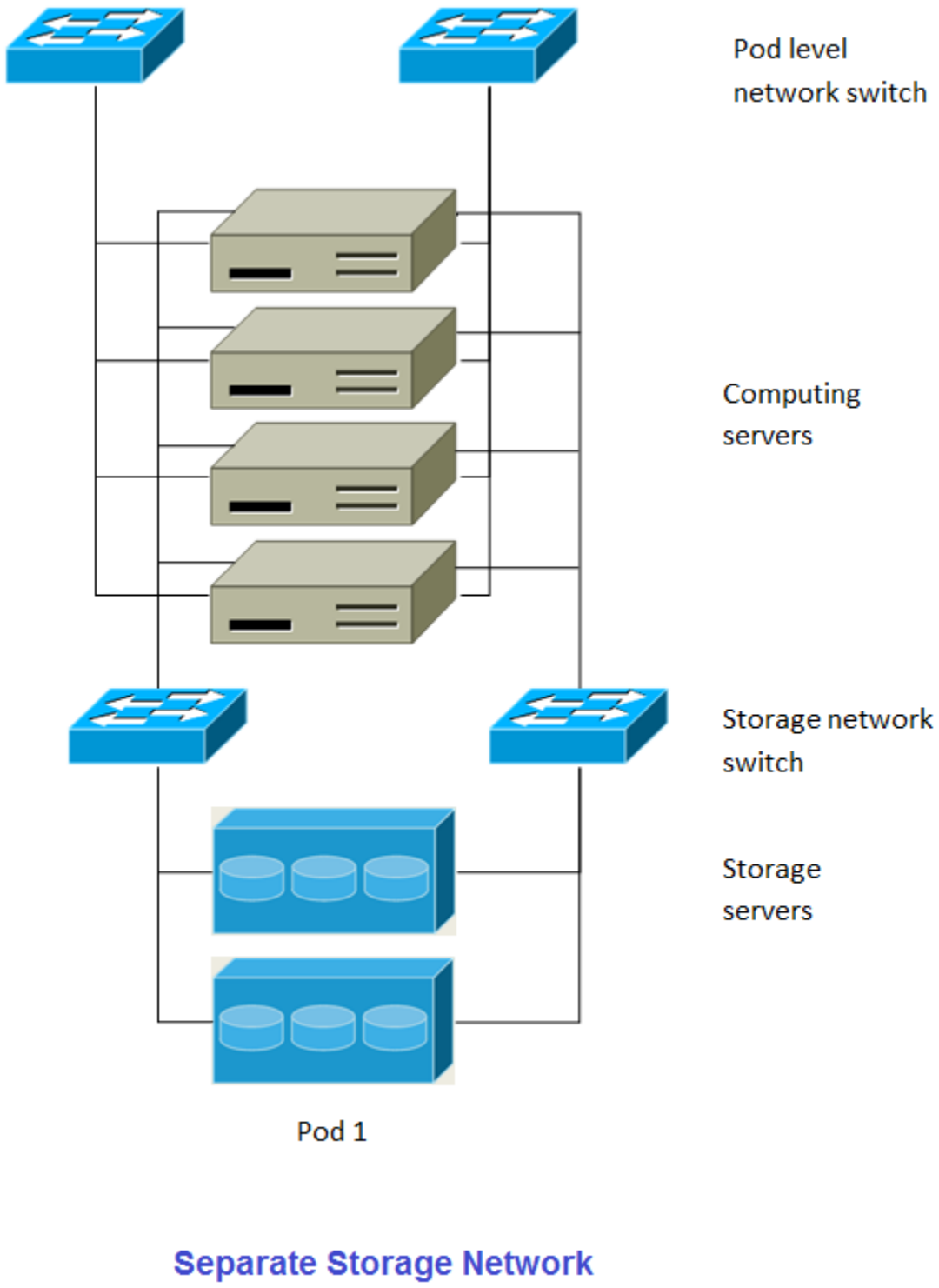
1.2.5 Multi-Site Deployment

The CloudStack platform scales well into multiple sites through the use of zones. The following diagram shows an example of a multi-site deployment.



Example of a Multi-Site Deployment

Data Center 1 houses the primary Management Server as well as zone 1. The MySQL database is replicated in real time to the secondary Management Server installation in Data Center 2.

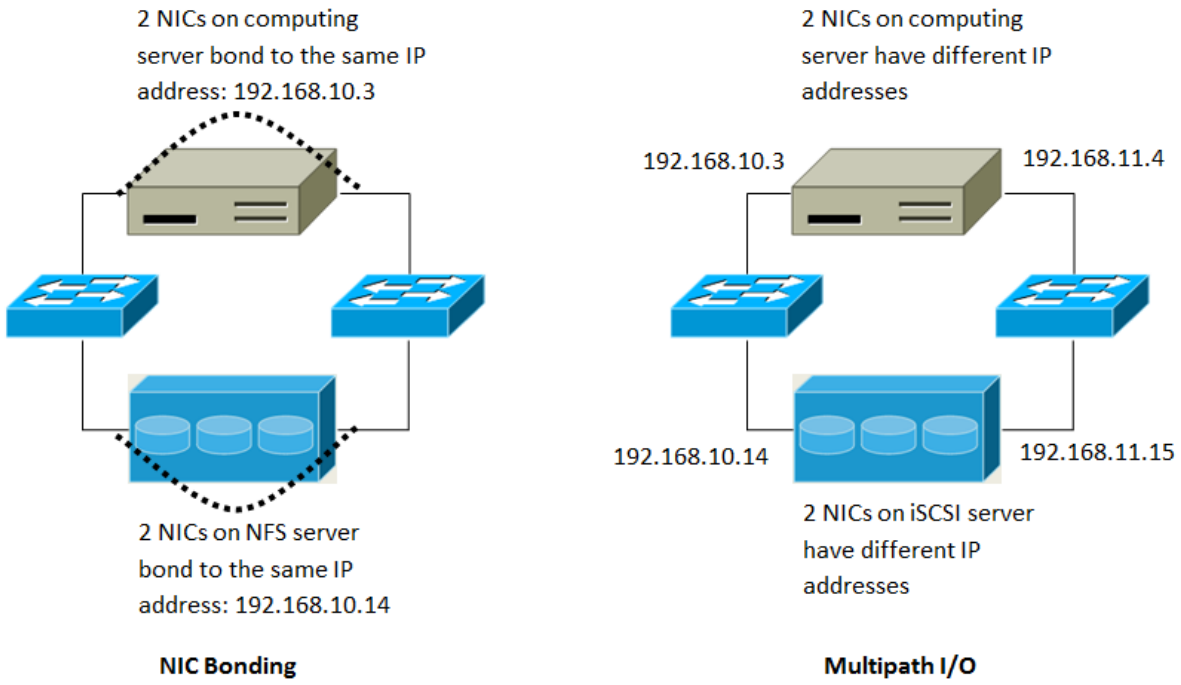


This diagram illustrates a setup with a separate storage network. Each server has four NICs, two connected to pod-level network switches and two connected to storage network switches.

There are two ways to configure the storage network:

- Bonded NIC and redundant switches can be deployed for NFS. In NFS deployments, redundant switches and bonded NICs still result in one network (one CIDR block+ default gateway address).

- iSCSI can take advantage of two separate storage networks (two CIDR blocks each with its own default gateway). Multipath iSCSI client can failover and load balance between separate storage networks.



NIC Bonding and Multipath I/O

This diagram illustrates the differences between NIC bonding and Multipath I/O (MPIO). NIC bonding configuration involves only one network. MPIO involves two separate networks.

1.2.6 Choosing a Hypervisor

CloudStack supports many popular hypervisors. Your cloud can consist entirely of hosts running a single hypervisor, or you can use multiple hypervisors. Each cluster of hosts must run the same hypervisor.

You might already have an installed base of nodes running a particular hypervisor, in which case, your choice of hypervisor has already been made. If you are starting from scratch, you need to decide what hypervisor software best suits your needs. A discussion of the relative advantages of each hypervisor is outside the scope of our documentation. However, it will help you to know which features of each hypervisor are supported by CloudStack. The following table provides this information.

Feature	XenServer	vSphere	KVM - RHEL	LXC	HyperV	Bare Metal
Network Throttling	Yes	Yes	No	No	?	N/A
Security groups in zones that use basic networking	Yes	No	Yes	Yes	?	No
iSCSI	Yes	Yes	Yes	Yes	Yes	N/A
FibreChannel	Yes	Yes	Yes	Yes	Yes	N/A
Local Disk	Yes	Yes	Yes	Yes	Yes	Yes
HA	Yes	Yes (Native)	Yes	?	Yes	N/A
Snapshots of local disk	Yes	Yes	Yes	?	?	N/A
Local disk as data disk	Yes	No	Yes	Yes	Yes	N/A
Work load balancing	No	DRS	No	No	?	N/A
Manual live migration of VMs from host to host	Yes	Yes	Yes	?	Yes	N/A
Conserve management traffic IP address by using link local network to communicate with virtual router	Yes	No	Yes	Yes	?	N/A

Hypervisor Support for Primary Storage

The following table shows storage options and parameters for different hypervisors.

Primary Storage Type	XenServer	vSphere	KVM - RHEL	LXC	HyperV
Format for Disks, Templates, and Snapshots	VHD	VMDK	QCOW2		VHD
iSCSI support	CLVM	VMFS	Yes via Shared Mountpoint	Yes via Shared Mountpoint	No
Fiber Channel support	Yes, Via existing SR	VMFS	Yes via Shared Mountpoint	Yes via Shared Mountpoint	No
NFS support	Yes	Yes	Yes	Yes	No
Local storage support	Yes	Yes	Yes	Yes	Yes
Storage over-provisioning	NFS	NFS and iSCSI	NFS		No
SMB/CIFS	No	No	No	No	Yes

XenServer uses a clustered LVM system to store VM images on iSCSI and Fiber Channel volumes and does not support over-provisioning in the hypervisor. The storage server itself, however, can support thin-provisioning. As a result the CloudStack can still support storage over-provisioning by running on thin-provisioned storage volumes.

KVM supports "Shared Mountpoint" storage. A shared mountpoint is a file system path local to each server in a given cluster. The path must be the same across all Hosts in the cluster, for example /mnt/primary1. This shared mountpoint is assumed to be a clustered filesystem such as OCFS2. In this case the CloudStack does not attempt to mount or unmount the storage as is done with NFS. The CloudStack requires that the administrator insure that the storage is available

With NFS storage, CloudStack manages the overprovisioning. In this case the global configuration parameter storage.overprovisioning.factor controls the degree of overprovisioning. This is independent of hypervisor type.

Local storage is an option for primary storage for vSphere, XenServer, and KVM. When the local disk option is enabled, a local disk storage pool is automatically created on each host. To use local storage for the System Virtual Machines (such as the Virtual Router), set system.vm.use.local.storage to true in global configuration.

CloudStack supports multiple primary storage pools in a Cluster. For example, you could provision 2 NFS servers in primary storage. Or you could provision 1 iSCSI LUN initially and then add a second iSCSI LUN when the first approaches capacity.

1.2.7 Best Practices

Deploying a cloud is challenging. There are many different technology choices to make, and CloudStack is flexible enough in its configuration that there are many possible ways to combine and configure the chosen technology. This section contains suggestions and requirements about cloud deployments.

These should be treated as suggestions and not absolutes. However, we do encourage anyone planning to build a cloud outside of these guidelines to seek guidance and advice on the project mailing lists.

Process Best Practices

- A staging system that models the production environment is strongly advised. It is critical if customizations have been applied to CloudStack.
- Allow adequate time for installation, a beta, and learning the system. Installs with basic networking can be done in hours. Installs with advanced networking usually take several days for the first attempt, with complicated installations taking longer. For a full production system, allow at least 4-8 weeks for a beta to work through all of the integration issues. You can get help from fellow users on the cloudstack-users mailing list.

Setup Best Practices

- Each host should be configured to accept connections only from well-known entities such as the CloudStack Management Server or your network monitoring software.
- Use multiple clusters per pod if you need to achieve a certain switch density.
- Primary storage mountpoints or LUNs should not exceed 6 TB in size. It is better to have multiple smaller primary storage elements per cluster than one large one.
- When exporting shares on primary storage, avoid data loss by restricting the range of IP addresses that can access the storage. See "Linux NFS on Local Disks and DAS" or "Linux NFS on iSCSI".
- NIC bonding is straightforward to implement and provides increased reliability.
- 10G networks are generally recommended for storage access when larger servers that can support relatively more VMs are used.
- Host capacity should generally be modeled in terms of RAM for the guests. Storage and CPU may be overprovisioned. RAM may not. RAM is usually the limiting factor in capacity designs.
- (XenServer) Configure the XenServer dom0 settings to allocate more memory to dom0. This can enable XenServer to handle larger numbers of virtual machines. We recommend 2940 MB of RAM for XenServer dom0. For instructions on how to do this, see http://support.citrix.com/article/CTX126531. The article refers to XenServer 5.6, but the same information applies to XenServer 6.0.

Maintenance Best Practices

- Monitor host disk space. Many host failures occur because the host's root disk fills up from logs that were not rotated adequately.

- Monitor the total number of VM instances in each cluster, and disable allocation to the cluster if the total is approaching the maximum that the hypervisor can handle. Be sure to leave a safety margin to allow for the possibility of one or more hosts failing, which would increase the VM load on the other hosts as the VMs are redeployed. Consult the documentation for your chosen hypervisor to find the maximum permitted number of VMs per host, then use CloudStack global configuration settings to set this as the default limit. Monitor the VM activity in each cluster and keep the total number of VMs below a safe level that allows for the occasional host failure. For example, if there are N hosts in the cluster, and you want to allow for one host in the cluster to be down at any given time, the total number of VM instances you can permit in the cluster is at most (N-1) * (per-host-limit). Once a cluster reaches this number of VMs, use the CloudStack UI to disable allocation to the cluster.

Warning: The lack of up-do-date hotfixes can lead to data corruption and lost VMs.

Be sure all the hotfixes provided by the hypervisor vendor are applied. Track the release of hypervisor patches through your hypervisor vendor's support channel, and apply patches as soon as possible after they are released. CloudStack will not track or notify you of required hypervisor patches. It is essential that your hosts are completely up to date with the provided hypervisor patches. The hypervisor vendor is likely to refuse to support any system that is not up to date with patches.

1.3 Network Setup

Achieving the correct networking setup is crucial to a successful CloudStack installation. This section contains information to help you make decisions and follow the right procedures to get your network set up correctly.

1.3.1 Basic and Advanced Networking

CloudStack provides two styles of networking:.

Basic For AWS-style networking. Provides a single network where guest isolation can be provided through layer-3 means such as security groups (IP address source filtering).

Advanced For more sophisticated network topologies. This network model provides the most flexibility in defining guest networks, but requires more configuration steps than basic networking.

Each zone has either basic or advanced networking. Once the choice of networking model for a zone has been made and configured in CloudStack, it can not be changed. A zone is either basic or advanced for its entire lifetime.

The following table compares the networking features in the two networking models.

Networking Feature	Basic Network	Advanced Network
Number of networks	Single network	Multiple networks
Firewall type	Physical	Physical and Virtual
Load balancer	Physical	Physical and Virtual
Isolation type	Layer 3	Layer 2 and Layer 3
VPN support	No	Yes
Port forwarding	Physical	Physical and Virtual
1:1 NAT	Physical	Physical and Virtual
Source NAT	No	Physical and Virtual
Userdata	Yes	Yes
Network usage monitoring	sFlow / netFlow at physical router	Hypervisor and Virtual Router
DNS and DHCP	Yes	Yes

The two types of networking may be in use in the same cloud. However, a given zone must use either Basic Networking or Advanced Networking.

Different types of network traffic can be segmented on the same physical network. Guest traffic can also be segmented by account. To isolate traffic, you can use separate VLANs. If you are using separate VLANs on a single physical network, make sure the VLAN tags are in separate numerical ranges.

1.3.2 VLAN Allocation Example

VLANs are required for public and guest traffic. The following is an example of a VLAN allocation scheme:

VLAN IDs	Traffic type	Scope
less than 500	Management traffic. Reserved for administrative purposes.	CloudStack software can access this, hypervisors, system VMs.
500-599	VLAN carrying public traffic.	CloudStack accounts.
600-799	VLANs carrying guest traffic.	CloudStack accounts. Account-specific VLAN is chosen from this pool.
800-899	VLANs carrying guest traffic.	CloudStack accounts. Account-specific VLAN chosen by CloudStack admin to assign to that account.
900-999	VLAN carrying guest traffic	CloudStack accounts. Can be scoped by project, domain, or all accounts.
greater than 1000	Reserved for future use	

1.3.3 Example Hardware Configuration

This section contains an example configuration of specific switch models for zone-level layer-3 switching. It assumes VLAN management protocols, such as VTP or GVRP, have been disabled. The example scripts must be changed appropriately if you choose to use VTP or GVRP.

Dell 62xx

The following steps show how a Dell 62xx is configured for zone-level layer-3 switching. These steps assume VLAN 201 is used to route untagged private IPs for pod 1, and pod 1's layer-2 switch is connected to Ethernet port 1/g1.

The Dell 62xx Series switch supports up to 1024 VLANs.

1. Configure all the VLANs in the database.

```
vlan database
vlan 200-999
exit
```

2. Configure Ethernet port 1/g1.

```
interface ethernet 1/g1
switchport mode general
switchport general pvid 201
switchport general allowed vlan add 201 untagged
switchport general allowed vlan add 300-999 tagged
exit
```

The statements configure Ethernet port 1/g1 as follows:

- VLAN 201 is the native untagged VLAN for port 1/g1.
- All VLANs (300-999) are passed to all the pod-level layer-2 switches.

Cisco 3750

The following steps show how a Cisco 3750 is configured for zone-level layer-3 switching. These steps assume VLAN 201 is used to route untagged private IPs for pod 1, and pod 1's layer-2 switch is connected to GigabitEthernet1/0/1.

1. Setting VTP mode to transparent allows us to utilize VLAN IDs above 1000. Since we only use VLANs up to 999, vtp transparent mode is not strictly required.

```
vtp mode transparent
vlan 200-999
exit
```

2. Configure GigabitEthernet1/0/1.

```
interface GigabitEthernet1/0/1
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk native vlan 201
exit
```

The statements configure GigabitEthernet1/0/1 as follows:

- VLAN 201 is the native untagged VLAN for port GigabitEthernet1/0/1.
- Cisco passes all VLANs by default. As a result, all VLANs (300-999) are passed to all the pod-level layer-2 switches.

1.3.4 Layer-2 Switch

The layer-2 switch is the access switching layer inside the pod.

- It should trunk all VLANs into every computing host.
- It should switch traffic for the management network containing computing and storage hosts. The layer-3 switch will serve as the gateway for the management network.

The following sections contain example configurations for specific switch models for pod-level layer-2 switching. It assumes VLAN management protocols such as VTP or GVRP have been disabled. The scripts must be changed appropriately if you choose to use VTP or GVRP.

Dell 62xx

The following steps show how a Dell 62xx is configured for pod-level layer-2 switching.

1. Configure all the VLANs in the database.

```
vlan database
vlan 300-999
exit
```

2. VLAN 201 is used to route untagged private IP addresses for pod 1, and pod 1 is connected to this layer-2 switch.

```
interface range ethernet all
switchport mode general
switchport general allowed vlan add 300-999 tagged
exit
```

The statements configure all Ethernet ports to function as follows:

- All ports are configured the same way.
- All VLANs (300-999) are passed through all the ports of the layer-2 switch.

Cisco 3750

The following steps show how a Cisco 3750 is configured for pod-level layer-2 switching.

1. Setting VTP mode to transparent allows us to utilize VLAN IDs above 1000. Since we only use VLANs up to 999, vtp transparent mode is not strictly required.

```
vtp mode transparent
vlan 300-999
exit
```

2. Configure all ports to dot1q and set 201 as the native VLAN.

```
interface range GigabitEthernet 1/0/1-24
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk native vlan 201
exit
```

By default, Cisco passes all VLANs. Cisco switches complain of the native VLAN IDs are different when 2 ports are connected together. That's why you must specify VLAN 201 as the native VLAN on the layer-2 switch.

1.3.5 Hardware Firewall

All deployments should have a firewall protecting the management server; see Generic Firewall Provisions. Optionally, some deployments may also have a Juniper SRX firewall that will be the default gateway for the guest networks; see *External Guest Firewall Integration for Juniper SRX (Optional)*

Generic Firewall Provisions

The hardware firewall is required to serve two purposes:

- Protect the Management Servers. NAT and port forwarding should be configured to direct traffic from the public Internet to the Management Servers.
- Route management network traffic between multiple zones. Site-to-site VPN should be configured between multiple zones.

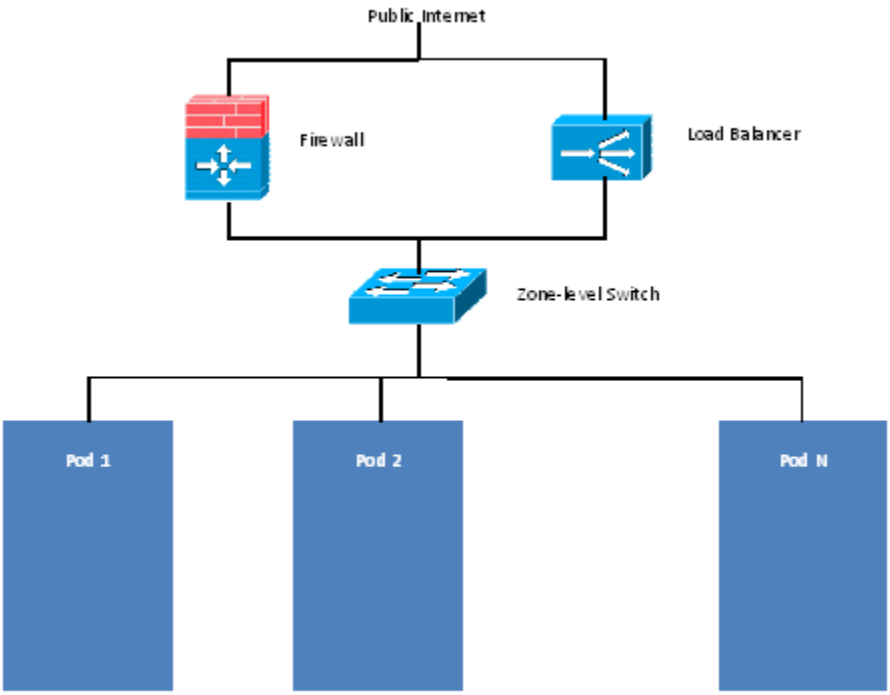
To achieve the above purposes you must set up fixed configurations for the firewall. Firewall rules and policies need not change as users are provisioned into the cloud. Any brand of hardware firewall that supports NAT and site-to-site VPN can be used.

External Guest Firewall Integration for Juniper SRX (Optional)

Note: Available only for guests using advanced networking.

CloudStack provides for direct management of the Juniper SRX series of firewalls. This enables CloudStack to establish static NAT mappings from public IPs to guest VMs, and to use the Juniper device in place of the virtual router for firewall services. You can have one or more Juniper SRX per zone. This feature is optional. If Juniper integration is not provisioned, CloudStack will use the virtual router for these services.

The Juniper SRX can optionally be used in conjunction with an external load balancer. External Network elements can be deployed in a side-by-side or inline configuration.



CloudStack requires the Juniper SRX firewall to be configured as follows:

Note: Supported SRX software version is 10.3 or higher.

1. Install your SRX appliance according to the vendor's instructions.
2. Connect one interface to the management network and one interface to the public network. Alternatively, you can connect the same interface to both networks and a use a VLAN for the public network.
3. Make sure "vlan-tagging" is enabled on the private interface.
4. Record the public and private interface names. If you used a VLAN for the public interface, add a ".[VLAN TAG]" after the interface name. For example, if you are using ge-0/0/3 for your public interface and VLAN tag 301, your public interface name would be "ge-0/0/3.301". Your private interface name should always be untagged because the CloudStack software automatically creates tagged logical interfaces.

5. Create a public security zone and a private security zone. By default, these will already exist and will be called "untrust" and "trust". Add the public interface to the public zone and the private interface to the private zone. Note down the security zone names.
6. Make sure there is a security policy from the private zone to the public zone that allows all traffic.
7. Note the username and password of the account you want the CloudStack software to log in to when it is programming rules.
8. Make sure the "ssh" and "xnm-clear-text" system services are enabled.
9. If traffic metering is desired:
   (a) Create an incoming firewall filter and an outgoing firewall filter. These filters should be the same names as your public security zone name and private security zone name respectively. The filters should be set to be "interface-specific". For example, here is the configuration where the public zone is "untrust" and the private zone is "trust":

```
root@cloud-srx# show firewall
filter trust {
    interface-specific;
}
filter untrust {
    interface-specific;
}
```

   (b) Add the firewall filters to your public interface. For example, a sample configuration output (for public interface ge-0/0/3.0, public security zone untrust, and private security zone trust) is:

```
ge-0/0/3 {
    unit 0 {
        family inet {
            filter {
                input untrust;
                output trust;
            }
            address 172.25.0.252/16;
        }
    }
}
```

10. Make sure all VLANs are brought to the private interface of the SRX.
11. After the CloudStack Management Server is installed, log in to the CloudStack UI as administrator.
12. In the left navigation bar, click Infrastructure.
13. In Zones, click View More.
14. Choose the zone you want to work with.
15. Click the Network tab.
16. In the Network Service Providers node of the diagram, click Configure. (You might have to scroll down to see this.)
17. Click SRX.
18. Click the Add New SRX button (+) and provide the following:
    - IP Address: The IP address of the SRX.
    - Username: The user name of the account on the SRX that CloudStack should use.

- Password: The password of the account.
- Public Interface. The name of the public interface on the SRX. For example, ge-0/0/2. A ".x" at the end of the interface indicates the VLAN that is in use.
- Private Interface: The name of the private interface on the SRX. For example, ge-0/0/1.
- Usage Interface: (Optional) Typically, the public interface is used to meter traffic. If you want to use a different interface, specify its name here
- Number of Retries: The number of times to attempt a command on the SRX before failing. The default value is 2.
- Timeout (seconds): The time to wait for a command on the SRX before considering it failed. Default is 300 seconds.
- Public Network: The name of the public network on the SRX. For example, trust.
- Private Network: The name of the private network on the SRX. For example, untrust.
- Capacity: The number of networks the device can handle
- Dedicated: When marked as dedicated, this device will be dedicated to a single account. When Dedicated is checked, the value in the Capacity field has no significance implicitly, its value is 1

19. Click OK.
20. Click Global Settings. Set the parameter external.network.stats.interval to indicate how often you want CloudStack to fetch network usage statistics from the Juniper SRX. If you are not using the SRX to gather network usage statistics, set to 0.

External Guest Firewall Integration for Cisco VNMC (Optional)

Cisco Virtual Network Management Center (VNMC) provides centralized multi-device and policy management for Cisco Network Virtual Services. You can integrate Cisco VNMC with CloudStack to leverage the firewall and NAT service offered by ASA 1000v Cloud Firewall. Use it in a Cisco Nexus 1000v dvSwitch-enabled cluster in CloudStack. In such a deployment, you will be able to:

- Configure Cisco ASA 1000v firewalls. You can configure one per guest network.
- Use Cisco ASA 1000v firewalls to create and apply security profiles that contain ACL policy sets for both ingress and egress traffic.
- Use Cisco ASA 1000v firewalls to create and apply Source NAT, Port Forwarding, and Static NAT policy sets.

CloudStack supports Cisco VNMC on Cisco Nexus 1000v dvSwich-enabled VMware hypervisors.

Using Cisco ASA 1000v Firewall, Cisco Nexus 1000v dvSwitch, and Cisco VNMC in a Deployment

Guidelines

- Cisco ASA 1000v firewall is supported only in Isolated Guest Networks.
- Cisco ASA 1000v firewall is not supported on VPC.
- Cisco ASA 1000v firewall is not supported for load balancing.
- When a guest network is created with Cisco VNMC firewall provider, an additional public IP is acquired along with the Source NAT IP. The Source NAT IP is used for the rules, whereas the additional IP is used to for the ASA outside interface. Ensure that this additional public IP is not released. You can identify this IP as soon as the network is in implemented state and before acquiring any further public IPs. The additional IP is the one that

is not marked as Source NAT. You can find the IP used for the ASA outside interface by looking at the Cisco VNMC used in your guest network.

- Use the public IP address range from a single subnet. You cannot add IP addresses from different subnets.
- Only one ASA instance per VLAN is allowed because multiple VLANS cannot be trunked to ASA ports. Therefore, you can use only one ASA instance in a guest network.
- Only one Cisco VNMC per zone is allowed.
- Supported only in Inline mode deployment with load balancer.
- The ASA firewall rule is applicable to all the public IPs in the guest network. Unlike the firewall rules created on virtual router, a rule created on the ASA device is not tied to a specific public IP.
- Use a version of Cisco Nexus 1000v dvSwitch that support the vservice command. For example: nexus-1000v.4.2.1.SV1.5.2b.bin

  Cisco VNMC requires the vservice command to be available on the Nexus switch to create a guest network in CloudStack.

Prerequisites

1. Configure Cisco Nexus 1000v dvSwitch in a vCenter environment.

   Create Port profiles for both internal and external network interfaces on Cisco Nexus 1000v dvSwitch. Note down the inside port profile, which needs to be provided while adding the ASA appliance to CloudStack.

   For information on configuration, see *Configuring a vSphere Cluster with Nexus 1000v Virtual Switch*.

2. Deploy and configure Cisco VNMC.

   For more information, see Installing Cisco Virtual Network Management Center and Configuring Cisco Virtual Network Management Center.

3. Register Cisco Nexus 1000v dvSwitch with Cisco VNMC.

   For more information, see Registering a Cisco Nexus 1000V with Cisco VNMC.

4. Create Inside and Outside port profiles in Cisco Nexus 1000v dvSwitch.

   For more information, see *Configuring a vSphere Cluster with Nexus 1000v Virtual Switch*.

5. Deploy and Cisco ASA 1000v appliance.

   For more information, see Setting Up the ASA 1000V Using VNMC.

   Typically, you create a pool of ASA 1000v appliances and register them with CloudStack.

   Specify the following while setting up a Cisco ASA 1000v instance:

   - VNMC host IP.
   - Ensure that you add ASA appliance in VNMC mode.
   - Port profiles for the Management and HA network interfaces. This need to be pre-created on Cisco Nexus 1000v dvSwitch.
   - Internal and external port profiles.
   - The Management IP for Cisco ASA 1000v appliance. Specify the gateway such that the VNMC IP is reachable.
   - Administrator credentials
   - VNMC credentials

6. Register Cisco ASA 1000v with VNMC.

   After Cisco ASA 1000v instance is powered on, register VNMC from the ASA console.

Using Cisco ASA 1000v Services

1. Ensure that all the prerequisites are met.

   See *"Prerequisites"*.
2. Add a VNMC instance.

   See *"Adding a VNMC Instance"*.
3. Add a ASA 1000v instance.

   See adding-an-asa-1000v-instance.
4. Create a Network Offering and use Cisco VNMC as the service provider for desired services.

   See *Creating a Network Offering Using Cisco ASA 1000v*.
5. Create an Isolated Guest Network by using the network offering you just created.

Adding a VNMC Instance

1. Log in to the CloudStack UI as administrator.
2. In the left navigation bar, click Infrastructure.
3. In Zones, click View More.
4. Choose the zone you want to work with.
5. Click the Physical Network tab.
6. In the Network Service Providers node of the diagram, click Configure.

   You might have to scroll down to see this.
7. Click Cisco VNMC.
8. Click View VNMC Devices.
9. Click the Add VNMC Device and provide the following:
   - Host: The IP address of the VNMC instance.
   - Username: The user name of the account on the VNMC instance that CloudStack should use.
   - Password: The password of the account.
10. Click OK.

Adding an ASA 1000v Instance

1. Log in to the CloudStack UI as administrator.
2. In the left navigation bar, click Infrastructure.
3. In Zones, click View More.
4. Choose the zone you want to work with.

5. Click the Physical Network tab.
6. In the Network Service Providers node of the diagram, click Configure.

   You might have to scroll down to see this.
7. Click Cisco VNMC.
8. Click View ASA 1000v.
9. Click the Add CiscoASA1000v Resource and provide the following:
   - **Host**: The management IP address of the ASA 1000v instance. The IP address is used to connect to ASA 1000V.
   - **Inside Port Profile**: The Inside Port Profile configured on Cisco Nexus1000v dvSwitch.
   - **Cluster**: The VMware cluster to which you are adding the ASA 1000v instance.

     Ensure that the cluster is Cisco Nexus 1000v dvSwitch enabled.
10. Click OK.

Creating a Network Offering Using Cisco ASA 1000v

To have Cisco ASA 1000v support for a guest network, create a network offering as follows:

1. Log in to the CloudStack UI as a user or admin.
2. From the Select Offering drop-down, choose Network Offering.
3. Click Add Network Offering.
4. In the dialog, make the following choices:
   - **Name**: Any desired name for the network offering.
   - **Description**: A short description of the offering that can be displayed to users.
   - **Network Rate**: Allowed data transfer rate in MB per second.
   - **Traffic Type**: The type of network traffic that will be carried on the network.
   - **Guest Type**: Choose whether the guest network is isolated or shared.
   - **Persistent**: Indicate whether the guest network is persistent or not. The network that you can provision without having to deploy a VM on it is termed persistent network.
   - **VPC**: This option indicate whether the guest network is Virtual Private Cloud-enabled. A Virtual Private Cloud (VPC) is a private, isolated part of CloudStack. A VPC can have its own virtual network topology that resembles a traditional physical network. For more information on VPCs, see :ref: *about-vpc*.
   - **Specify VLAN**: (Isolated guest networks only) Indicate whether a VLAN should be specified when this offering is used.
   - **Supported Services**: Use Cisco VNMC as the service provider for Firewall, Source NAT, Port Forwarding, and Static NAT to create an Isolated guest network offering.
   - **System Offering**: Choose the system service offering that you want virtual routers to use in this network.
   - **Conserve mode**: Indicate whether to use conserve mode. In this mode, network resources are allocated only when the first virtual machine starts in the network.
5. Click OK

   The network offering is created.

Reusing ASA 1000v Appliance in new Guest Networks

You can reuse an ASA 1000v appliance in a new guest network after the necessary cleanup. Typically, ASA 1000v is cleaned up when the logical edge firewall is cleaned up in VNMC. If this cleanup does not happen, you need to reset the appliance to its factory settings for use in new guest networks. As part of this, enable SSH on the appliance and store the SSH credentials by registering on VNMC.

1. Open a command line on the ASA appliance:
   (a) Run the following:

   ```
   ASA1000V(config)# reload
   ```

   You are prompted with the following message:

   ```
   System config has been modified. Save? [Y]es/[N]o:"
   ```

   (b) Enter N.

   You will get the following confirmation message:

   ```
   "Proceed with reload? [confirm]"
   ```

   (c) Restart the appliance.
2. Register the ASA 1000v appliance with the VNMC:

   ```
   ASA1000V(config)# vnmc policy-agent
   ASA1000V(config-vnmc-policy-agent)# registration host vnmc_ip_address
   ASA1000V(config-vnmc-policy-agent)# shared-secret key where key is the shared␣
   ↪secret for authentication of the ASA 1000V connection to the Cisco VNMC
   ```

External Guest Load Balancer Integration (Optional)

CloudStack can optionally use a Citrix NetScaler or BigIP F5 load balancer to provide load balancing services to guests. If this is not enabled, CloudStack will use the software load balancer in the virtual router.

To install and enable an external load balancer for CloudStack management:

1. Set up the appliance according to the vendor's directions.
2. Connect it to the networks carrying public traffic and management traffic (these could be the same network).
3. Record the IP address, username, password, public interface name, and private interface name. The interface names will be something like "1.1" or "1.2".
4. Make sure that the VLANs are trunked to the management network interface.
5. After the CloudStack Management Server is installed, log in as administrator to the CloudStack UI.
6. In the left navigation bar, click Infrastructure.
7. In Zones, click View More.
8. Choose the zone you want to work with.
9. Click the Network tab.
10. In the Network Service Providers node of the diagram, click Configure. (You might have to scroll down to see this.)
11. Click NetScaler or F5.

12. Click the Add button (+) and provide the following:

    For NetScaler:

    - IP Address: The IP address of the SRX.
    - Username/Password: The authentication credentials to access the device. CloudStack uses these credentials to access the device.
    - Type: The type of device that is being added. It could be F5 Big Ip Load Balancer, NetScaler VPX, NetScaler MPX, or NetScaler SDX. For a comparison of the NetScaler types, see the CloudStack Administration Guide.
    - Public interface: Interface of device that is configured to be part of the public network.
    - Private interface: Interface of device that is configured to be part of the private network.
    - Number of retries. Number of times to attempt a command on the device before considering the operation failed. Default is 2.
    - Capacity: The number of networks the device can handle.
    - Dedicated: When marked as dedicated, this device will be dedicated to a single account. When Dedicated is checked, the value in the Capacity field has no significance implicitly, its value is 1.
13. Click OK.

The installation and provisioning of the external load balancer is finished. You can proceed to add VMs and NAT or load balancing rules.

1.3.6 Management Server Load Balancing

CloudStack can use a load balancer to provide a virtual IP for multiple Management Servers. The administrator is responsible for creating the load balancer rules for the Management Servers. The application requires persistence or stickiness across multiple sessions. The following chart lists the ports that should be load balanced and whether or not persistence is required.

Even if persistence is not required, enabling it is permitted.

Source Port	Destination Port	Protocol	Persistence Required?
80 or 443	8080 (or 20400 with AJP)	HTTP (or AJP)	Yes
8250	8250	TCP	Yes
8096	8096	HTTP	No

In addition to above settings, the administrator is responsible for setting the 'host' global config value from the management server IP to load balancer virtual IP address. If the 'host' value is not set to the VIP for Port 8250 and one of your management servers crashes, the UI is still available but the system VMs will not be able to contact the management server.

1.3.7 Topology Requirements

Security Requirements

The public Internet must not be able to access port 8096 or port 8250 on the Management Server.

Runtime Internal Communications Requirements

- The Management Servers communicate with each other to coordinate tasks. This communication uses TCP on ports 8250 and 9090.
- The console proxy VMs connect to all hosts in the zone over the management traffic network. Therefore the management traffic network of any given pod in the zone must have connectivity to the management traffic network of all other pods in the zone.
- The secondary storage VMs and console proxy VMs connect to the Management Server on port 8250. If you are using multiple Management Servers, the load balanced IP address of the Management Servers on port 8250 must be reachable.

Storage Network Topology Requirements

The secondary storage NFS export is mounted by the secondary storage VM. Secondary storage traffic goes over the management traffic network, even if there is a separate storage network. Primary storage traffic goes over the storage network, if available. If you choose to place secondary storage NFS servers on the storage network, you must make sure there is a route from the management traffic network to the storage network.

External Firewall Topology Requirements

When external firewall integration is in place, the public IP VLAN must still be trunked to the Hosts. This is required to support the Secondary Storage VM and Console Proxy VM.

Advanced Zone Topology Requirements

With Advanced Networking, separate subnets must be used for private and public networks.

XenServer Topology Requirements

The Management Servers communicate with XenServer hosts on ports 22 (ssh), 80 (HTTP), and 443 (HTTPs).

VMware Topology Requirements

- The Management Server and secondary storage VMs must be able to access vCenter and all ESXi hosts in the zone. To allow the necessary access through the firewall, keep port 443 open.
- The Management Servers communicate with VMware vCenter servers on port 443 (HTTPs).
- The Management Servers communicate with the System VMs on port 3922 (ssh) on the management traffic network.

Hyper-V Topology Requirements

CloudStack Management Server communicates with Hyper-V Agent by using HTTPS. For secure communication between the Management Server and the Hyper-V host, open port 8250.

KVM Topology Requirements

The Management Servers communicate with KVM hosts on port 22 (ssh).

LXC Topology Requirements

The Management Servers communicate with LXC hosts on port 22 (ssh).

1.3.8 Guest Network Usage Integration for Traffic Sentinel

To collect usage data for a guest network, CloudStack needs to pull the data from an external network statistics collector installed on the network. Metering statistics for guest networks are available through CloudStack's integration with inMon Traffic Sentinel.

Traffic Sentinel is a network traffic usage data collection package. CloudStack can feed statistics from Traffic Sentinel into its own usage records, providing a basis for billing users of cloud infrastructure. Traffic Sentinel uses the traffic monitoring protocol sFlow. Routers and switches generate sFlow records and provide them for collection by Traffic Sentinel, then CloudStack queries the Traffic Sentinel database to obtain this information

To construct the query, CloudStack determines what guest IPs were in use during the current query interval. This includes both newly assigned IPs and IPs that were assigned in a previous time period and continued to be in use. CloudStack queries Traffic Sentinel for network statistics that apply to these IPs during the time period they remained allocated in CloudStack. The returned data is correlated with the customer account that owned each IP and the timestamps when IPs were assigned and released in order to create billable metering records in CloudStack. When the Usage Server runs, it collects this data.

To set up the integration between CloudStack and Traffic Sentinel:

1. On your network infrastructure, install Traffic Sentinel and configure it to gather traffic data. For installation and configuration steps, see inMon documentation at Traffic Sentinel Documentation.
2. In the Traffic Sentinel UI, configure Traffic Sentinel to accept script querying from guest users. CloudStack will be the guest user performing the remote queries to gather network usage for one or more IP addresses.

   Click File > Users > Access Control > Reports Query, then select Guest from the drop-down list.
3. On CloudStack, add the Traffic Sentinel host by calling the CloudStack API command addTrafficMonitor. Pass in the URL of the Traffic Sentinel as protocol + host + port (optional); for example, http://10.147.28.100:8080. For the addTrafficMonitor command syntax, see the API Reference at API Documentation.

   For information about how to call the CloudStack API, see the Developer's Guide at the CloudStack API Developer's Guide *The CloudStack API*
4. Log in to the CloudStack UI as administrator.
5. Select Configuration from the Global Settings page, and set the following:

   direct.network.stats.interval: How often you want CloudStack to query Traffic Sentinel.

1.3.9 Setting Zone VLAN and Running VM Maximums

In the external networking case, every VM in a zone must have a unique guest IP address. There are two variables that you need to consider in determining how to configure CloudStack to support this: how many Zone VLANs do you expect to have and how many VMs do you expect to have running in the Zone at any one time.

Use the following table to determine how to configure CloudStack for your deployment.

guest.vlan.bits	Maximum Running VMs per Zone	Maximum Zone VLANs
12	4096	4094
11	8192	2048
10	16384	1024
10	32768	512

Based on your deployment's needs, choose the appropriate value of guest.vlan.bits. Set it as described in Edit the Global Configuration Settings (Optional) section and restart the Management Server.

1.4 Storage Setup

1.4.1 Introduction

Primary Storage

CloudStack is designed to work with a wide variety of commodity and enterprise-rated storage systems. CloudStack can also leverage the local disks within the hypervisor hosts if supported by the selected hypervisor. Storage type support for guest virtual disks differs based on hypervisor selection.

Storage Type	XenServer	vSphere	KVM
NFS	Supported	Supported	Supported
iSCSI	Supported	Supported via VMFS	Supported via Clustered Filesystems
Fiber Channel	Supported via Pre-existing SR	Supported	Supported via Clustered Filesystems
Local Disk	Supported	Supported	Supported

The use of the Cluster Logical Volume Manager (CLVM) for KVM is not officially supported with CloudStack.

Secondary Storage

CloudStack is designed to work with any scalable secondary storage system. The only requirement is that the secondary storage system supports the NFS protocol. For large, multi-zone deployments, S3 compatible storage is also supported for secondary storage. This allows for secondary storage which can span an entire region, however an NFS staging area must be maintained in each zone as most hypervisors are not capable of directly mounting S3 type storage.

1.4.2 Configurations

Small-Scale Setup

In a small-scale setup, a single NFS server can function as both primary and secondary storage. The NFS server must export two separate shares, one for primary storage and the other for secondary storage. This could be a VM or physical host running an NFS service on a Linux OS or a virtual software appliance. Disk and network performance are still important in a small scale setup to get a good experience when deploying, running or snapshotting VMs.

Large-Scale Setup

In large-scale environments primary and secondary storage typically consist of independent physical storage arrays.

Primary storage is likely to have to support mostly random read/write I/O once a template has been deployed. Secondary storage is only going to experience sustained sequential reads or writes.

In clouds which will experience a large number of users taking snapshots or deploying VMs at the same time, secondary storage performance will be important to maintain a good user experience.

It is important to start the design of your storage with the a rough profile of the workloads which it will be required to support. Care should be taken to consider the IOPS demands of your guest VMs as much as the volume of data to be stored and the bandwidth (MB/s) available at the storage interfaces.

1.4.3 Storage Architecture

There are many different storage types available which are generally suitable for CloudStack environments. Specific use cases should be considered when deciding the best one for your environment and financial constraints often make the 'perfect' storage architecture economically unrealistic.

Broadly, the architectures of the available primary storage types can be split into 3 types:

Local Storage

Local storage works best for pure 'cloud-era' workloads which rarely need to be migrated between storage pools and where HA of individual VMs is not required. As SSDs become more mainstream/affordable, local storage based VMs can now be served with the size of IOPS which previously could only be generated by large arrays with 10s of spindles. Local storage is highly scalable because as you add hosts you would add the same proportion of storage. Local Storage is relatively inefficent as it can not take advantage of linked clones or any deduplication.

'Traditional' node-based Shared Storage

Traditional node-based storage are arrays which consist of a controller/controller pair attached to a number of disks in shelves. Ideally a cloud architecture would have one of these physical arrays per CloudStack pod to limit the 'blast-radius' of a failure to a single pod. This is often not economically viable, however one should look to try to reduce the scale of any incident relative to any zone with any single array where possible. The use of shared storage enables workloads to be immediately restarted on an alternate host should a host fail. These shared storage arrays often have the ability to create 'tiers' of storage utilising say large SATA disks, 15k SAS disks and SSDs. These differently performing tiers can then be presented as different offerings to users. The sizing of an array should take into account the IOPS required by the workload as well as the volume of data to be stored. One should also consider the number of VMs which a storage array will be expected to support, and the maximum network bandwidth possible through the controllers.

Clustered Shared Storage

Clustered shared storage arrays are the new generation of storage which do not have a single set of interfaces where data enters and exits the array. Instead it is distributed between all of the active nodes giving greatly improved scalability and performance. Some shared storage arrays enable all data to continue to be accessible even in the event of the loss of an entire node.

The network topology should be carefully considered when using clustered shared storage to avoid creating bottlenecks in the network fabric.

Network Configuration For Storage

Care should be taken when designing your cloud to take into consideration not only the performance of your disk arrays but also the bandwidth available to move that traffic between the switch fabric and the array interfaces.

1.4.4 CloudStack Networking For Storage

The first thing to understand is the process of provisioning primary storage. When you create a primary storage pool for any given cluster, the CloudStack management server tells each hosts' hypervisor to mount the NFS share or (iSCSI LUN). The storage pool will be presented within the hypervisor as a datastore (VMware), storage repository (XenServer/XCP) or a mount point (KVM), the important point is that it is the hypervisor itself that communicates with the primary storage, the CloudStack management server only communicates with the host hypervisor. Now, all

hypervisors communicate with the outside world via some kind of management interface – think VMKernel port on ESXi or 'Management Interface' on XenServer. As the CloudStack management server needs to communicate with the hypervisor in the host, this management interface must be on the CloudStack 'management' or 'private' network. There may be other interfaces configured on your host carrying guest and public traffic to/from VMs within the hosts but the hypervisor itself doesn't/can't communicate over these interfaces.

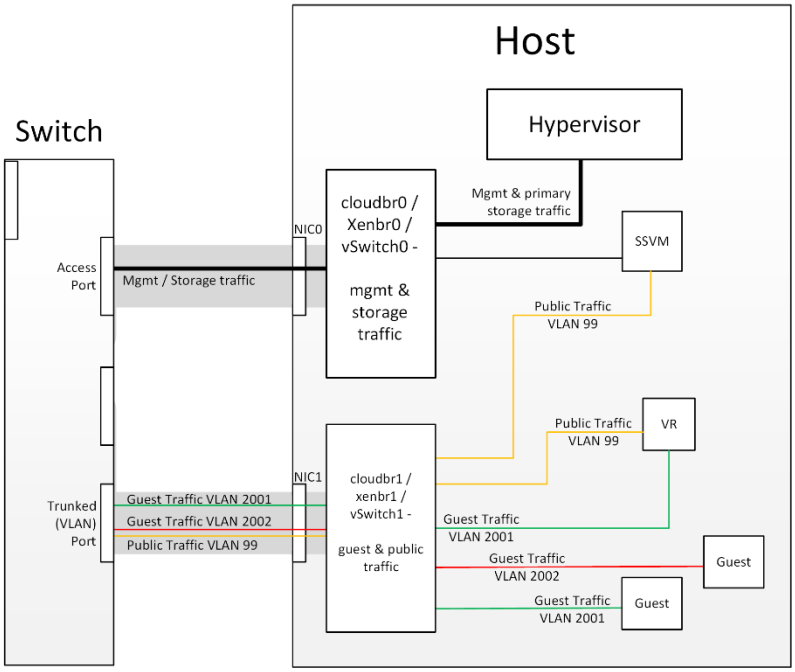


Figure 1: Hypervisor communications

Separating Primary Storage traffic For those from a pure virtualisation background, the concept of creating a specific interface for storage traffic will not be new; it has long been best practice for iSCSI traffic to have a dedicated switch fabric to avoid any latency or contention issues. Sometimes in the cloud(Stack) world we forget that we are simply orchestrating processes that the hypervisors already carry out and that many 'normal' hypervisor configurations still apply. The logical reasoning which explains how this splitting of traffic works is as follows:

1. If you want an additional interface over which the hypervisor can communicate (excluding teamed or bonded interfaces) you need to give it an IP address.
2. The mechanism to create an additional interface that the hypervisor can use is to create an additional management interface
3. So that the hypervisor can differentiate between the management interfaces they have to be in different (non-overlapping) subnets
4. In order for the 'primary storage' management interface to communicate with the primary storage, the interfaces on the primary storage arrays must be in the same CIDR as the 'primary storage' management interface.
5. Therefore the primary storage must be in a different subnet to the management network

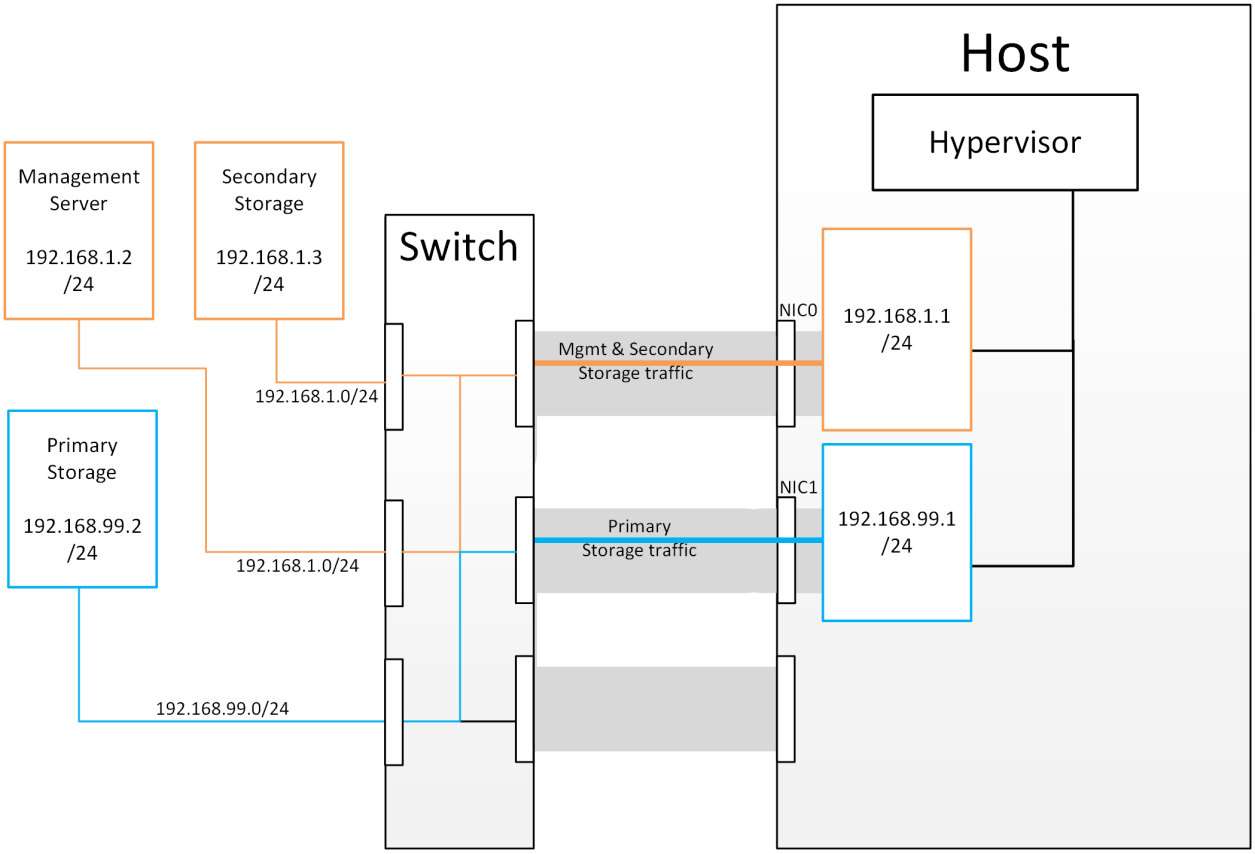


Figure 2: Subnetting of Storage Traffic

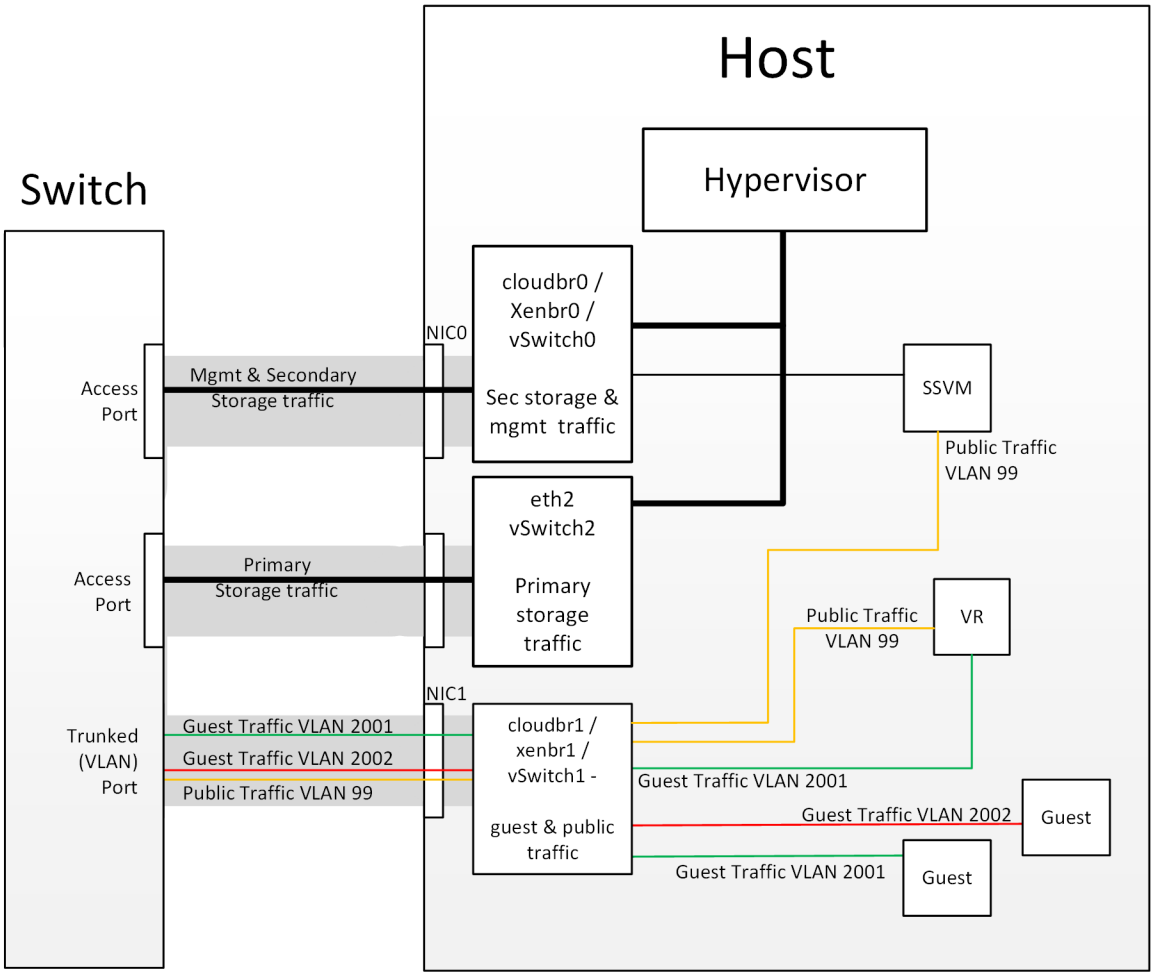


Figure 3: Hypervisor Communications with Separated Storage Traffic

Other Primary Storage Types If you are using PreSetup or SharedMountPoints to connect to IP based storage then the same principles apply; if the primary storage and 'primary storage interface' are in a different subnet to the 'management subnet' then the hypervisor will use the 'primary storage interface' to communicate with the primary storage.

Small-Scale Example Configurations

In this section we go through a few examples of how to set up storage to work properly on a few types of NFS and iSCSI storage systems.

Linux NFS on Local Disks and DAS

This section describes how to configure an NFS export on a standard Linux installation. The exact commands might vary depending on the operating system version.

1. Install the RHEL/CentOS distribution on the storage server.
2. If the root volume is more than 2 TB in size, create a smaller boot volume to install RHEL/CentOS. A root volume of 20 GB should be sufficient.
3. After the system is installed, create a directory called /export. This can each be a directory in the root partition itself or a mount point for a large disk volume.

4. If you have more than 16TB of storage on one host, create multiple EXT3 file systems and multiple NFS exports. Individual EXT3 file systems cannot exceed 16TB.
5. After /export directory is created, run the following command to configure it as an NFS export.

```
# echo "/export <CIDR>(rw,async,no_root_squash,no_subtree_check)" >> /etc/exports
```

   Adjust the above command to suit your deployment needs.

- **Limiting NFS export.** It is highly recommended that you limit the NFS export to a particular subnet by specifying a subnet mask (e.g.,"192.168.1.0/24"). By allowing access from only within the expected cluster, you avoid having non-pool member mount the storage. The limit you place must include the management network(s) and the storage network(s). If the two are the same network then one CIDR is sufficient. If you have a separate storage network you must provide separate CIDR's for both or one CIDR that is broad enough to span both.

  The following is an example with separate CIDRs:

```
/export 192.168.1.0/24(rw,async,no_root_squash,no_subtree_check) 10.50.1.0/
↪24(rw,async,no_root_squash,no_subtree_check)
```

- **Removing the async flag.** The async flag improves performance by allowing the NFS server to respond before writes are committed to the disk. Remove the async flag in your mission critical production deployment.

6. Run the following command to enable NFS service.

```
# chkconfig nfs on
```

7. Edit the /etc/sysconfig/nfs file and uncomment the following lines.

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

8. Edit the /etc/sysconfig/iptables file and add the following lines at the beginning of the INPUT chain.

```
-A INPUT -m state --state NEW -p udp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 32803 -j ACCEPT
-A INPUT -m state --state NEW -p udp --dport 32769 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 892 -j ACCEPT
-A INPUT -m state --state NEW -p udp --dport 892 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 875 -j ACCEPT
-A INPUT -m state --state NEW -p udp --dport 875 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 662 -j ACCEPT
-A INPUT -m state --state NEW -p udp --dport 662 -j ACCEPT
```

9. Reboot the server.

   An NFS share called /export is now set up.

Note: When copying and pasting a command, be sure the command has pasted as a single line before executing. Some document viewers may introduce unwanted line breaks in copied text.

Linux NFS on iSCSI

Use the following steps to set up a Linux NFS server export on an iSCSI volume. These steps apply to RHEL/CentOS 5 distributions.

1. Install iscsiadm.

```
# yum install iscsi-initiator-utils
# service iscsi start
# chkconfig --add iscsi
# chkconfig iscsi on
```

2. Discover the iSCSI target.

```
# iscsiadm -m discovery -t st -p <iSCSI Server IP address>:3260
```

For example:

```
# iscsiadm -m discovery -t st -p 172.23.10.240:3260 172.23.10.240:3260,1 iqn.2001-
↪05.com.equallogic:0-8a0906-83bcb3401-16e0002fd0a46f3d-rhel5-test
```

3. Log in.

```
# iscsiadm -m node -T <Complete Target Name> -l -p <Group IP>:3260
```

For example:

```
# iscsiadm -m node -l -T iqn.2001-05.com.equallogic:83bcb3401-16e0002fd0a46f3d-
↪rhel5-test -p 172.23.10.240:3260
```

4. Discover the SCSI disk. For example:

```
# iscsiadm -m session -P3 | grep Attached
Attached scsi disk sdb State: running
```

5. Format the disk as ext3 and mount the volume.

```
# mkfs.ext3 /dev/sdb
# mkdir -p /export
# mount /dev/sdb /export
```

6. Add the disk to /etc/fstab to make sure it gets mounted on boot.

```
/dev/sdb /export ext3 _netdev 0 0
```

Now you can set up /export as an NFS share.

- **Limiting NFS export.** In order to avoid data loss, it is highly recommended that you limit the NFS export to a particular subnet by specifying a subnet mask (e.g.,"192.168.1.0/24"). By allowing access from only within the expected cluster, you avoid having non-pool member mount the storage and inadvertently delete all its data. The limit you place must include the management network(s) and the storage network(s). If the two are the same network then one CIDR is sufficient. If you have a separate storage network you must provide separate CIDRs for both or one CIDR that is broad enough to span both.

  The following is an example with separate CIDRs:

```
/export 192.168.1.0/24(rw,async,no_root_squash,no_subtree_check) 10.50.1.0/24(rw,
↪async,no_root_squash,no_subtree_check)
```

- **Removing the async flag.** The async flag improves performance by allowing the NFS server to respond before writes are committed to the disk. Remove the async flag in your mission critical production deployment.

CHAPTER 2

Quick Installation Guide

2.1 Overview

2.1.1 What exactly are we building?

Infrastructure-as-a-Service (IaaS) clouds can be a complex thing to build, and by definition they have a plethora of options, which often lead to confusion for even experienced admins who are newcomers to building cloud platforms. The goal for this runbook is to provide a straightforward set of instructions to get you up and running with CloudStack with a minimum amount of trouble.

2.1.2 High level overview of the process

This runbook will focus on building a CloudStack cloud using KVM on CentOS 6.8 with NFS storage on a flat layer-2 network utilizing layer-3 network isolation (aka Security Groups), and doing it all on a single piece of hardware.

KVM, or Kernel-based Virtual Machine is a virtualization technology for the Linux kernel. KVM supports native virtualization atop processors with hardware virtualization extensions.

Security Groups act as distributed firewalls that control access to a group of virtual machines.

2.1.3 Prerequisites

To complete this runbook you'll need the following items:

1. At least one computer which supports and has enabled hardware virtualization.
2. The CentOS 6.8 x86_64 minimal install CD
3. A /24 network with the gateway being at xxx.xxx.xxx.1, no DHCP should be on this network and none of the computers running CloudStack will have a dynamic address. Again this is done for the sake of simplicity.

2.2 Environment

Before you begin , you need to prepare the environment before you install CloudStack. We will go over the steps to prepare now.

2.2.1 Operating System

Using the CentOS 6.8 x86_64 minimal install ISO, you'll need to install CentOS 6 on your hardware. The defaults will generally be acceptable for this installation.

Once this installation is complete, you'll want to connect to your freshly installed machine via SSH as the root user. Note that you should not allow root logins in a production environment, so be sure to turn off remote logins once you have finished the installation and configuration.

Configuring the network

By default the network will not come up on your hardware and you will need to configure it to work in your environment. Since we specified that there will be no DHCP server in this environment we will be manually configuring your network interface. We will assume, for the purposes of this exercise, that eth0 is the only network interface that will be connected and used.

Connecting via the console you should login as root. Check the file /etc/sysconfig/network-scripts/ifcfg-eth0, it will look like this by default:

```
DEVICE="eth0"
HWADDR="52:54:00:B9:A6:C0"
NM_CONTROLLED="yes"
ONBOOT="no"
```

Unfortunately, this configuration will not permit you to connect to the network, and is also unsuitable for our purposes with CloudStack. We want to configure that file so that it specifies the IP address, netmask, etc., as shown in the following example:

Note: You should not use the Hardware Address (aka the MAC address) from our example for your configuration. It is network interface specific, so you should keep the address already provided in the HWADDR directive.

```
DEVICE=eth0
HWADDR=52:54:00:B9:A6:C0
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
IPADDR=172.16.10.2
NETMASK=255.255.255.0
GATEWAY=172.16.10.1
DNS1=8.8.8.8
DNS2=8.8.4.4
```

Note: IP Addressing - Throughout this document we are assuming that you will have a /24 network for your CloudStack implementation. This can be any RFC 1918 network. However, we are assuming that you will match the machine address that we are using. Thus we may use 172.16.10.2 and because you might be using the 192.168.55.0/24 network you would use 192.168.55.2

Now that we have the configuration files properly set up, we need to run a few commands to start up the network:

```
# chkconfig network on

# service network start
```

Hostname

CloudStack requires that the hostname be properly set. If you used the default options in the installation, then your hostname is currently set to localhost.localdomain. To test this we will run:

```
# hostname --fqdn
```

At this point it will likely return:

```
localhost
```

To rectify this situation - we'll set the hostname by editing the /etc/hosts file so that it follows a similar format to this example:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
172.16.10.2 srvr1.cloud.priv
```

After you've modified that file, go ahead and restart the network using:

```
# service network restart
```

Now recheck with the hostname –fqdn command and ensure that it returns a FQDN response

SELinux

At the moment, for CloudStack to work properly SELinux must be set to permissive. We want to both configure this for future boots and modify it in the current running system.

To configure SELinux to be permissive in the running system we need to run the following command:

```
# setenforce 0
```

To ensure that it remains in that state we need to configure the file /etc/selinux/config to reflect the permissive state, as shown in this example:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

NTP

NTP configuration is a necessity for keeping all of the clocks in your cloud servers in sync. However, NTP is not installed by default. So we'll install and and configure NTP at this stage. Installation is accomplished as follows:

```
# yum -y install ntp
```

The actual default configuration is fine for our purposes, so we merely need to enable it and set it to start on boot as follows:

```
# chkconfig ntpd on
# service ntpd start
```

Configuring the CloudStack Package Repository

We need to configure the machine to use a CloudStack package repository.

Note: The Apache CloudStack official releases are source code. As such there are no 'official' binaries available. The full installation guide describes how to take the source release and generate RPMs and and yum repository. This guide attempts to keep things as simple as possible, and thus we are using one of the community-provided yum repositories.

To add the CloudStack repository, create /etc/yum.repos.d/cloudstack.repo and insert the following information.

```
[cloudstack]
name=cloudstack
baseurl=http://download.cloudstack.org/centos/6/|version|/
enabled=1
gpgcheck=0
```

2.2.2 NFS

Our configuration is going to use NFS for both primary and secondary storage. We are going to go ahead and setup two NFS shares for those purposes. We'll start out by installing nfs-utils.

```
# yum -y install nfs-utils
```

We now need to configure NFS to serve up two different shares. This is handled comparatively easily in the /etc/exports file. You should ensure that it has the following content:

```
/export/secondary *(rw,async,no_root_squash,no_subtree_check)
/export/primary *(rw,async,no_root_squash,no_subtree_check)
```

You will note that we specified two directories that don't exist (yet) on the system. We'll go ahead and create those directories and set permissions appropriately on them with the following commands:

```
# mkdir -p /export/primary
# mkdir /export/secondary
```

CentOS 6.x releases use NFSv4 by default. NFSv4 requires that domain setting matches on all clients. In our case, the domain is cloud.priv, so ensure that the domain setting in /etc/idmapd.conf is uncommented and set as follows: Domain = cloud.priv

Now you'll need uncomment the configuration values in the file /etc/sysconfig/nfs

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

Now we need to configure the firewall to permit incoming NFS connections. Edit the file /etc/sysconfig/iptables

```
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p udp --dport 111 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p tcp --dport 111 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p tcp --dport 2049 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p tcp --dport 32803 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p udp --dport 32769 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p tcp --dport 892 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p udp --dport 892 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p tcp --dport 875 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p udp --dport 875 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p tcp --dport 662 -j ACCEPT
-A INPUT -s 172.16.10.0/24 -m state --state NEW -p udp --dport 662 -j ACCEPT
```

Now you can restart the iptables service with the following command:

```
# service iptables restart
```

We now need to configure the nfs service to start on boot and actually start it on the host by executing the following commands:

```
# service rpcbind start
# service nfs start
# chkconfig rpcbind on
# chkconfig nfs on
```

2.3 Management Server Installation

We're going to install the CloudStack management server and surrounding tools.

2.3.1 Database Installation and Configuration

We'll start with installing MySQL and configuring some options to ensure it runs well with CloudStack.

Install by running the following command:

```
# yum -y install mysql-server
```

With MySQL now installed we need to make a few configuration changes to /etc/my.cnf. Specifically we need to add the following options to the [mysqld] section:

```
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
max_connections=350
log-bin=mysql-bin
binlog-format = 'ROW'
```

Now that MySQL is properly configured we can start it and configure it to start on boot as follows:

```
# service mysqld start
# chkconfig mysqld on
```

2.3.2 MySQL connector Installation

Install Python MySQL connector using the official MySQL packages repository. Create the file `/etc/yum.repos.d/mysql.repo` with the following content:

```
[mysql-connectors-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

2.3.3 Installation

We are now going to install the management server. We do that by executing the following command:

```
# yum -y install cloudstack-management
```

With the application itself installed we can now setup the database, we'll do that with the following command and options:

```
# cloudstack-setup-databases cloud:password@localhost --deploy-as=root
```

When this process is finished, you should see a message like "CloudStack has successfully initialized the database."

Now that the database has been created, we can take the final step in setting up the management server by issuing the following command:

```
# cloudstack-setup-management
```

If the servlet container is Tomcat7 the argument –tomcat7 must be used.

2.3.4 System Template Setup

CloudStack uses a number of system VMs to provide functionality for accessing the console of virtual machines, providing various networking services, and managing various aspects of storage. This step will acquire those system images ready for deployment when we bootstrap your cloud.

Now we need to download the system VM template and deploy that to the share we just mounted. The management server includes a script to properly manipulate the system VMs images.

```
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-
↪tmplt -m /export/secondary -u http://download.cloudstack.org/systemvm/4.
6/systemvm64template-4.6.0-kvm.qcow2.bz2 -h kvm -F
```

That concludes our setup of the management server. We still need to configure CloudStack, but we will do that after we get our hypervisor set up.

2.4 KVM Setup and Installation

KVM is the hypervisor we'll be using - we will recover the initial setup which has already been done on the hypervisor host and cover installation of the agent software, you can use the same steps to add additional KVM nodes to your CloudStack environment.

2.4.1 Prerequisites

We explicitly are using the management server as a compute node as well, which means that we have already performed many of the prerequisite steps when setting up the management server, but we will list them here for clarity. Those steps are:

Configuring the network

Hostname

SELinux

NTP

Configuring the CloudStack Package Repository

You shouldn't need to do that for the management server, of course, but any additional hosts will need for you to complete the above steps.

2.4.2 Installation

Installation of the KVM agent is trivial with just a single command, but afterwards we'll need to configure a few things.

```
# yum -y install cloudstack-agent
```

2.4.3 KVM Configuration

We have two different parts of KVM to configure, libvirt, and QEMU.

QEMU Configuration

KVM configuration is relatively simple at only a single item. We need to edit the QEMU VNC configuration. This is done by editing /etc/libvirt/qemu.conf and ensuring the following line is present and uncommented.

```
vnc_listen=0.0.0.0
```

Libvirt Configuration

CloudStack uses libvirt for managing virtual machines. Therefore it is vital that libvirt is configured correctly. Libvirt is a dependency of cloud-agent and should already be installed.

1. In order to have live migration working libvirt has to listen for unsecured TCP connections. We also need to turn off libvirts attempt to use Multicast DNS advertising. Both of these settings are in /etc/libvirt/libvirtd.conf

   Set the following paramaters:

```
listen_tls = 0
listen_tcp = 1
tcp_port = "16059"
auth_tcp = "none"
mdns_adv = 0
```

2. Turning on "listen_tcp" in libvirtd.conf is not enough, we have to change the parameters as well we also need to modify /etc/sysconfig/libvirtd:

   Uncomment the following line:

```
#LIBVIRTD_ARGS="--listen"
```

3. Restart libvirt

```
# service libvirtd restart
```

KVM configuration complete

For the sake of completeness you should check if KVM is running OK on your machine:

```
# lsmod | grep kvm
kvm_intel               55496  0
kvm                    337772  1 kvm_intel
```

That concludes our installation and configuration of KVM, and we'll now move to using the CloudStack UI for the actual configuration of our cloud.

2.5 Configuration

As we noted before we will be using security groups to provide isolation and by default that implies that we'll be using a flat layer-2 network. It also means that the simplicity of our setup means that we can use the quick installer.

2.5.1 UI Access

To get access to CloudStack's web interface, merely point your browser to http://172.16.10.2:8080/client The default username is 'admin', and the default password is 'password'. You should see a splash screen that allows you to choose several options for setting up CloudStack. You should choose the Continue with Basic Setup option.

You should now see a prompt requiring you to change the password for the admin user. Please do so.

2.5.2 Setting up a Zone

A zone is the largest organization entity in CloudStack - and we'll be creating one, this should be the screen that you see in front of you now. And for us there are 5 pieces of information that we need.

1. Name - we will set this to the ever-descriptive 'Zone1' for our cloud.
2. Public DNS 1 - we will set this to `8.8.8.8` for our cloud.
3. Public DNS 2 - we will set this to `8.8.4.4` for our cloud.
4. Internal DNS1 - we will also set this to `8.8.8.8` for our cloud.
5. Internal DNS2 - we will also set this to `8.8.4.4` for our cloud.

Note: CloudStack distinguishes between internal and public DNS. Internal DNS is assumed to be capable of resolving internal-only hostnames, such as your NFS server's DNS name. Public DNS is provided to the guest VMs to resolve public IP addresses. You can enter the same DNS server for both types, but if you do so, you must make sure that both internal and public IP addresses can route to the DNS server. In our specific case we will not use any names for resources internally, and we have indeed them set to look to the same external resource so as to not add a namerserver setup to our list of requirements.

2.5.3 Pod Configuration

Now that we've added a Zone, the next step that comes up is a prompt for information regading a pod. Which is looking for several items.

1. Name - We'll use `Pod1` for our cloud.
2. Gateway - We'll use `172.16.10.1` as our gateway
3. Netmask - We'll use `255.255.255.0`
4. Start/end reserved system IPs - we will use `172.16.10.10-172.16.10.20`
5. Guest gateway - We'll use `172.16.10.1`
6. Guest netmask - We'll use `255.255.255.0`
7. Guest start/end IP - We'll use `172.16.10.30-172.16.10.200`

2.5.4 Cluster

Now that we've added a Zone, we need only add a few more items for configuring the cluster.

1. Name - We'll use `Cluster1`
2. Hypervisor - Choose `KVM`

You should be prompted to add the first host to your cluster at this point. Only a few bits of information are needed.

1. Hostname - we'll use the IP address `172.16.10.2` since we didn't set up a DNS server.
2. Username - we'll use `root`
3. Password - enter the operating system password for the root user

Primary Storage

With your cluster now setup - you should be prompted for primary storage information. Choose NFS as the storage type and then enter the following values in the fields:

1. Name - We'll use `Primary1`
2. Server - We'll be using the IP address `172.16.10.2`
3. Path - Well define `/export/primary` as the path we are using

Secondary Storage

If this is a new zone, you'll be prompted for secondary storage information - populate it as follows:

1. NFS server - We'll use the IP address `172.16.10.2`
2. Path - We'll use `/export/secondary`

Now, click Launch and your cloud should begin setup - it may take several minutes depending on your internet connection speed for setup to finalize.

That's it, you are done with installation of your Apache CloudStack cloud.

CHAPTER 3

Installation Guide

This is the Apache CloudStack installation guide

3.1 Building from Source

3.1.1 Introduction

The official CloudStack release is always in source code form. You will likely be able to find "convenience binaries," the source is the canonical release. In this section, we'll cover acquiring the source release and building that so that you can deploy it using Maven or create Debian packages or RPMs.

Note that building and deploying directly from source is typically not the most efficient way to deploy an IaaS. However, we will cover that method as well as building RPMs or Debian packages for deploying CloudStack.

The instructions here are likely version-specific. That is, the method for building from source for the 4.7.x series is different from the 4.2.x series.

If you are working with a unreleased version of CloudStack, see the INSTALL.md file in the top-level directory of the release.

3.1.2 Downloading the release

You can download the latest CloudStack release from the Apache CloudStack project download page.

Prior releases are available via archive.apache.org as well. See the downloads page for more information on archived releases.

You'll notice several links under the 'Latest release' section. A link to a file ending in `tar.bz2`, as well as a PGP/GPG signature, MD5, and SHA512 file.

- The `tar.bz2` file contains the Bzip2-compressed tarball with the source code.
- The `.asc` file is a detached cryptographic signature that can be used to help verify the authenticity of the release.

- The `.md5` file is an MD5 hash of the release to aid in verify the validity of the release download.
- The `.sha` file is a SHA512 hash of the release to aid in verify the validity of the release download.

3.1.3 Verifying the downloaded release

There are a number of mechanisms to check the authenticity and validity of a downloaded release.

Getting the KEYS

To enable you to verify the GPG signature, you will need to download the KEYS file.

You next need to import those keys, which you can do by running:

```
$ wget http://www.apache.org/dist/cloudstack/KEYS
$ gpg --import KEYS
```

GPG

The CloudStack project provides a detached GPG signature of the release. To check the signature, run the following command:

```
$ gpg --verify apache-cloudstack-4.11.1.0-src.tar.bz2.asc
```

If the signature is valid you will see a line of output that contains 'Good signature'.

MD5

In addition to the cryptographic signature, CloudStack has an MD5 checksum that you can use to verify the download matches the release. You can verify this hash by executing the following command:

```
$ gpg --print-md MD5 apache-cloudstack-4.11.1.0-src.tar.bz2 | diff - apache-
↪cloudstack-4.11.1.0-src.tar.bz2.md5
```

If this successfully completes you should see no output. If there is any output from them, then there is a difference between the hash you generated locally and the hash that has been pulled from the server.

SHA512

In addition to the MD5 hash, the CloudStack project provides a SHA512 cryptographic hash to aid in assurance of the validity of the downloaded release. You can verify this hash by executing the following command:

```
$ gpg --print-md SHA512 apache-cloudstack-4.11.1.0-src.tar.bz2 | diff -␣
↪apache-cloudstack-4.11.1.0-src.tar.bz2.sha
```

If this command successfully completes you should see no output. If there is any output from them, then there is a difference between the hash you generated locally and the hash that has been pulled from the server.

3.1.4 Prerequisites for building Apache CloudStack

There are a number of prerequisites needed to build CloudStack. This document assumes compilation on a Linux system that uses RPMs or DEBs for package management.

You will need, at a minimum, the following to compile CloudStack:

1. Maven (version 3)
2. Java (Java 8/OpenJDK 1.8)
3. Apache Web Services Common Utilities (ws-commons-util)
4. MySQL
5. MySQLdb (provides Python database API)
6. genisoimage
7. rpmbuild or dpkg-dev

3.1.5 Extracting source

Extracting the CloudStack release is relatively simple and can be done with a single command as follows:

```
$ tar -jxvf apache-cloudstack-4.11.1.0-src.tar.bz2
```

You can now move into the directory:

```
$ cd ./apache-cloudstack-4.11.1.0-src
```

3.1.6 Install new MySQL connector

Install Python MySQL connector using the official MySQL packages repository.

MySQL connector APT repository

Install the following package provided by MySQL to enable official repositories:

```
wget http://dev.mysql.com/get/mysql-apt-config_0.7.3-1_all.deb
sudo dpkg -i mysql-apt-config_0.7.3-1_all.deb
```

Make sure to activate the repository for MySQL connectors.

```
sudo apt-get update
sudo apt-get install mysql-connector-python
```

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

3.1.7 Building DEB packages

In addition to the bootstrap dependencies, you'll also need to install several other dependencies. Note that we recommend using Maven 3.

```
$ sudo apt-get update
$ sudo apt-get install python-software-properties
$ sudo apt-get update
$ sudo apt-get install debhelper openjdk-8-jdk libws-commons-util-java genisoimage␣
↪libcommons-codec-java libcommons-httpclient-java liblog4j1.2-java maven
```

While we have defined, and you have presumably already installed the bootstrap prerequisites, there are a number of build time prerequisites that need to be resolved. CloudStack uses maven for dependency resolution. You can resolve the buildtime depdencies for CloudStack by running:

```
$ mvn -P deps
```

Now that we have resolved the dependencies we can move on to building CloudStack and packaging them into DEBs by issuing the following command.

```
$ dpkg-buildpackage -uc -us
```

This command will build the following debian packages. You should have all of the following:

```
cloudstack-common-4.11.1.0.amd64.deb
cloudstack-management-4.11.1.0.amd64.deb
cloudstack-agent-4.11.1.0.amd64.deb
cloudstack-usage-4.11.1.0.amd64.deb
cloudstack-cli-4.11.1.0.amd64.deb
```

Setting up an APT repo

After you've created the packages, you'll want to copy them to a system where you can serve the packages over HTTP. You'll create a directory for the packages and then use `dpkg-scanpackages` to create `Packages.gz`, which holds information about the archive structure. Finally, you'll add the repository to your system(s) so you can install the packages using APT.

The first step is to make sure that you have the **dpkg-dev** package installed. This should have been installed when you pulled in the **debhelper** application previously, but if you're generating `Packages.gz` on a different system, be sure that it's installed there as well.

```
$ sudo apt-get install dpkg-dev
```

The next step is to copy the DEBs to the directory where they can be served over HTTP. We'll use `/var/www/cloudstack/repo` in the examples, but change the directory to whatever works for you.

```
$ sudo mkdir -p /var/www/cloudstack/repo/binary
$ sudo cp *.deb /var/www/cloudstack/repo/binary
$ cd /var/www/cloudstack/repo/binary
$ sudo sh -c 'dpkg-scanpackages . /dev/null | tee Packages | gzip -9 >␣
↪Packages.gz'
```

Note: You can safely ignore the warning about a missing override file.

Now you should have all of the DEB packages and `Packages.gz` in the `binary` directory and available over HTTP. (You may want to use `wget` or `curl` to test this before moving on to the next step.)

Configuring your machines to use the APT repository

Now that we have created the repository, you need to configure your machine to make use of the APT repository. You can do this by adding a repository file under `/etc/apt/sources.list.d`. Use your preferred editor to create `/etc/apt/sources.list.d/cloudstack.list` with this line:

```
deb http://server.url/cloudstack/repo/binary ./
```

Now that you have the repository info in place, you'll want to run another update so that APT knows where to find the CloudStack packages.

```
$ sudo apt-get update
```

You can now move on to the instructions under Install on Ubuntu.

3.1.8 Building RPMs from Source

As mentioned previously in *"Prerequisites for building Apache CloudStack"*, you will need to install several prerequisites before you can build packages for CloudStack. Here we'll assume you're working with a 64-bit build of CentOS or Red Hat Enterprise Linux.

```
# yum groupinstall "Development Tools"
```

```
# yum install java-1.8.0-openjdk-devel.x86_64 genisoimage mysql mysql-server ws-
↪commons-util MySQL-python createrepo
```

Next, you'll need to install build-time dependencies for CloudStack with Maven. We're using Maven 3, so you'll want to grab Maven 3.0.5 (Binary tar.gz) and uncompress it in your home directory (or whatever location you prefer):

```
$ cd ~
$ tar zxvf apache-maven-3.0.5-bin.tar.gz
```

```
$ export PATH=~/apache-maven-3.0.5/bin:$PATH
```

Maven also needs to know where Java is, and expects the JAVA_HOME environment variable to be set:

```
$ export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk.x86_64
```

Verify that Maven is installed correctly:

```
$ mvn --version
```

You probably want to ensure that your environment variables will survive a logout/reboot. Be sure to update `~/.bashrc` with the PATH and JAVA_HOME variables.

Building RPMs for CloudStack is fairly simple. Assuming you already have the source downloaded and have uncompressed the tarball into a local directory, you're going to be able to generate packages in just a few minutes.

Note: Packaging has changed. If you've created packages for CloudStack previously, you should be aware that the process has changed considerably since the project has moved to using Apache Maven. Please be sure to follow the steps in this section closely.

Generating RPMS

Now that we have the prerequisites and source, you will cd to the *packaging/* directory.

```
$ cd packaging/
```

Generating RPMs is done using the `package.sh` script:

```
$ ./package.sh -d centos63
```

For other supported options(like centos7), run `./package.sh --help`

That will run for a bit and then place the finished packages in `dist/rpmbuild/RPMS/x86_64/`.

You should see the following RPMs in that directory:

```
cloudstack-agent-4.11.1.0.el6.x86_64.rpm
cloudstack-cli-4.11.1.0.el6.x86_64.rpm
cloudstack-common-4.11.1.0.el6.x86_64.rpm
cloudstack-management-4.11.1.0.el6.x86_64.rpm
cloudstack-usage-4.11.1.0.el6.x86_64.rpm
```

Creating a yum repo

While RPMs is a useful packaging format - it's most easily consumed from Yum repositories over a network. The next step is to create a Yum Repo with the finished packages:

```
$ mkdir -p ~/tmp/repo

$ cd ../..
$ cp dist/rpmbuild/RPMS/x86_64/*rpm ~/tmp/repo/

$ createrepo /tmp/repo
```

The files and directories within `~/tmp/repo` can now be uploaded to a web server and serve as a yum repository.

Configuring your systems to use your new yum repository

Now that your yum repository is populated with RPMs and metadata we need to configure the machines that need to install CloudStack. Create a file named `/etc/yum.repos.d/cloudstack.repo` with this information:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://webserver.tld/path/to/repo
enabled=1
gpgcheck=0
```

Completing this step will allow you to easily install CloudStack on a number of machines across the network.

3.1.9 Building Non-OSS

If you need support for the VMware, NetApp, F5, NetScaler, SRX, or any other non-Open Source Software (nonoss) plugins, you'll need to download a few components on your own and follow a slightly different procedure to build from source.

> **Warning:** Some of the plugins supported by CloudStack cannot be distributed with CloudStack for licensing reasons. In some cases, some of the required libraries/JARs are under a proprietary license. In other cases, the required libraries may be under a license that's not compatible with Apache's licensing guidelines for third-party products.

1. To build the Non-OSS plugins, you'll need to have the requisite JARs installed under the `deps` directory.

   Because these modules require dependencies that can't be distributed with CloudStack you'll need to download them yourself. Links to the most recent dependencies are listed on the *How to build CloudStack* page on the wiki.

2. You may also need to download vhd-util, which was removed due to licensing issues. You'll copy vhd-util to the `scripts/vm/hypervisor/xenserver/` directory.

3. Once you have all the dependencies copied over, you'll be able to build CloudStack with the `noredist` option:

```
$ mvn clean
$ mvn install -Dnoredist
```

1. Once you've built CloudStack with the `noredist` profile, you can package it using the *"Building RPMs from Source"* or *"Building DEB packages"* instructions.

3.2 General Installation

3.2.1 Installation overview

- *Introduction*
  - *Who Should Read This*
  - *Installation Steps*
- *Minimum System Requirements*
  - *Management Server, Database, and Storage System Requirements*
  - *Host/Hypervisor System Requirements*
- *Package Repository*

Introduction

Who Should Read This

For those who have already gone through a design phase and planned a more sophisticated deployment, or those who are ready to start scaling up a trial installation. With the following procedures, you can start using the more powerful

features of CloudStack, such as advanced VLAN networking, high availability, additional network elements such as load balancers and firewalls, and support for multiple hypervisors including Citrix XenServer, KVM, and VMware vSphere.

Installation Steps

For anything more than a simple trial installation, you will need guidance for a variety of configuration choices. It is strongly recommended that you read the following:

- Choosing a Deployment Architecture
- Choosing a Hypervisor: Supported Features
- Network Setup
- Storage Setup
- Best Practices

1. Make sure you have the required hardware ready. See *Minimum System Requirements*
2. Install the Management Server (choose single-node or multi-node). See *Management Server Installation*
3. Configure your cloud. See *Configuring your CloudStack Installation*
   (a) Using CloudStack UI. See **User Interface* :ref:'log-in-to-ui*
   (b) Add a zone. Includes the first pod, cluster, and host. See *Adding a Zone*
   (c) Add more pods (optional). See *Adding a Pod*
   (d) Add more clusters (optional). See *Adding a Cluster*
   (e) Add more hosts (optional). See *Adding a Host*
   (f) Add more primary storage (optional). See *Add Primary Storage*
   (g) Add more secondary storage (optional). See *Add Secondary Storage*
4. Try using the cloud. See *Initialize and Test*

Minimum System Requirements

Management Server, Database, and Storage System Requirements

The machines that will run the Management Server and MySQL database must meet the following requirements. The same machines can also be used to provide primary and secondary storage, such as via localdisk or NFS. The Management Server may be placed on a virtual machine.

- Operating system:
  - Preferred: CentOS/RHEL 7.2+, CentOS/RHEL 6.8+ or Ubuntu 14.04(.2) or higher
- 64-bit x86 CPU (more cores results in better performance)
- 4 GB of memory
- 250 GB of local disk (more results in better capability; 500 GB recommended)
- At least 1 NIC
- Statically allocated IP address
- Fully qualified domain name as returned by the hostname command

Host/Hypervisor System Requirements

The host is where the cloud services run in the form of guest virtual machines. Each host is one machine that meets the following requirements:

- Must support HVM (Intel-VT or AMD-V enabled).
- 64-bit x86 CPU (more cores results in better performance)
- Hardware virtualization support required
- 4 GB of memory
- 36 GB of local disk
- At least 1 NIC
- Latest hotfixes applied to hypervisor software
- When you deploy CloudStack, the hypervisor host must not have any VMs already running
- All hosts within a cluster must be homogeneous. The CPUs must be of the same type, count, and feature flags.

Hosts have additional requirements depending on the hypervisor. See the requirements listed at the top of the Installation section for your chosen hypervisor:

Warning: Be sure you fulfill the additional hypervisor requirements and installation steps provided in this Guide. Hypervisor hosts must be properly prepared to work with CloudStack. For example, the requirements for XenServer are listed under Citrix XenServer Installation.

Package Repository

CloudStack is only distributed from source from the official Apache mirrors. However, members of the CloudStack community may build convenience binaries so that users can install Apache CloudStack without needing to build from source.

If you didn't follow the steps to build your own packages from source in the sections for "Building RPMs from Source" or "Building DEB packages" you may find pre-built DEB and RPM packages for your convenience linked from the downloads page.

Note: These repositories contain both the Management Server and KVM Hypervisor packages.

3.2.2 Management Server Installation

Overview

This section describes installing the Management Server. There are two slightly different installation flows, depending on how many Management Server nodes will be in your cloud:

- A single Management Server node, with MySQL on the same node.
- Multiple Management Server nodes, with MySQL on a node separate from the Management Servers.

In either case, each machine must meet the system requirements described in *Minimum System Requirements*.

Warning: For the sake of security, be sure the public Internet can not access port 8096 or port 8250 on the Management Server.

The procedure for installing the Management Server is:

1. Prepare the Operating System
2. (XenServer only) Download and install vhd-util.
3. Install the First Management Server
4. Install and Configure the MySQL database
5. Prepare NFS Shares
6. Prepare and Start Additional Management Servers (optional)
7. Prepare the System VM Template

Prepare the Operating System

The OS must be prepared to host the Management Server using the following steps. These steps must be performed on each Management Server node.

1. Log in to your OS as root.
2. Check for a fully qualified hostname.

   ```
   hostname --fqdn
   ```

   This should return a fully qualified hostname such as "management1.lab.example.org". If it does not, edit /etc/hosts so that it does.

3. Make sure that the machine can reach the Internet.

   ```
   ping cloudstack.apache.org
   ```

4. Turn on NTP for time synchronization.

   Note: NTP is required to synchronize the clocks of the servers in your cloud.

   Install NTP.

   ```
   yum install ntp
   ```

   ```
   sudo apt-get install openntpd
   ```

5. Repeat all of these steps on every host where the Management Server will be installed.

Install the Management Server on the First Host

The first step in installation, whether you are installing the Management Server on one host or many, is to install the software on a single node.

Note: If you are planning to install the Management Server on multiple nodes for high availability, do not proceed to the additional nodes yet. That step will come later.

The CloudStack Management server can be installed using either RPM or DEB packages. These packages will depend on everything you need to run the Management server.

Configure package repository

CloudStack is only distributed from source from the official mirrors. However, members of the CloudStack community may build convenience binaries so that users can install Apache CloudStack without needing to build from source.

If you didn't follow the steps to build your own packages from source in the sections for "Building RPMs from Source" or "Building DEB packages" you may find pre-built DEB and RPM packages for your convenience linked from the downloads page.

Note: These repositories contain both the Management Server and KVM Hypervisor packages.

RPM package repository

There is a RPM package repository for CloudStack so you can easily install on RHEL based platforms.

If you're using an RPM-based system, you'll want to add the Yum repository so that you can install CloudStack with Yum.

Yum repository information is found under `/etc/yum.repos.d`. You'll see several `.repo` files in this directory, each one denoting a specific repository.

To add the CloudStack repository, create `/etc/yum.repos.d/cloudstack.repo` and insert the following information.

```
[cloudstack]
name=cloudstack
baseurl=http://download.cloudstack.org/centos/$releasever/4.11/
enabled=1
gpgcheck=0
```

Now you should now be able to install CloudStack using Yum.

DEB package repository

You can add a DEB package repository to your apt sources with the following commands. Please note that only packages for Ubuntu 14.04 LTS (Trusty) and Ubuntu 16.04 (Xenial) are being built at this time. DISCLAIMER: Ubuntu 12.04 (Precise) is no longer supported.

Use your preferred editor and open (or create) `/etc/apt/sources.list.d/cloudstack.list`. Add the community provided repository to the file:

```
deb http://download.cloudstack.org/ubuntu trusty 4.11
```

We now have to add the public key to the trusted keys.

```
sudo wget -O - http://download.cloudstack.org/release.asc|apt-key add -
```

Now update your local apt cache.

```
sudo apt-get update
```

Your DEB package repository should now be configured and ready for use.

Install on CentOS/RHEL

```
yum install cloudstack-management
```

Install on Ubuntu

```
sudo apt-get install cloudstack-management
```

Downloading vhd-util

This procedure is required only for installations where XenServer is installed on the hypervisor hosts.

Before setting up the Management Server, download vhd-util from http://download.cloudstack.org/tools/vhd-util. and copy it into `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver` of the Management Server.

Install the database server

The CloudStack management server uses a MySQL database server to store its data. When you are installing the management server on a single node, you can install the MySQL server locally. For an installation that has multiple management server nodes, we assume the MySQL database also runs on a separate node.

CloudStack has been tested with MySQL 5.1 and 5.5. These versions are included in RHEL/CentOS and Ubuntu.

Install the Database on the Management Server Node

This section describes how to install MySQL on the same machine with the Management Server. This technique is intended for a simple deployment that has a single Management Server node. If you have a multi-node Management Server deployment, you will typically use a separate node for MySQL. See *Install the Database on a Separate Node*.

1. Install MySQL from the package repository of your distribution:

   ```
   yum install mysql-server
   ```

   ```
   sudo apt-get install mysql-server
   ```

2. Open the MySQL configuration file. The configuration file is `/etc/my.cnf` or `/etc/mysql/my.cnf`, depending on your OS.

   Insert the following lines in the `[mysqld]` section.

   You can put these lines below the datadir line. The max_connections parameter should be set to 350 multiplied by the number of Management Servers you are deploying. This example assumes one Management Server.

```
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
max_connections=350
log-bin=mysql-bin
binlog-format = 'ROW'
```

Note: You can also create a file `/etc/mysql/conf.d/cloudstack.cnf` and add these directives there. Don't forget to add `[mysqld]` on the first line of the file.

3. Start or restart MySQL to put the new configuration into effect.

   On RHEL/CentOS, MySQL doesn't automatically start after installation. Start it manually.

```
service mysqld start
```

   On Ubuntu, restart MySQL.

```
sudo service mysql restart
```

4. (CentOS and RHEL only; not required on Ubuntu)

   Warning: On RHEL and CentOS, MySQL does not set a root password by default. It is very strongly recommended that you set a root password as a security precaution.

   Run the following command to secure your installation. You can answer "Y" to all questions.

```
mysql_secure_installation
```

5. CloudStack can be blocked by security mechanisms, such as SELinux. Disable SELinux to ensure + that the Agent has all the required permissions.

   Configure SELinux (RHEL and CentOS):

   (a) Check whether SELinux is installed on your machine. If not, you can skip this section.

   In RHEL or CentOS, SELinux is installed and enabled by default. You can verify this with:

```
rpm -qa | grep selinux
```

   (b) Set the SELINUX variable in `/etc/selinux/config` to "permissive". This ensures that the permissive setting will be maintained after a system reboot.

   In RHEL or CentOS:

```
vi /etc/selinux/config
```

   Change the following line

```
SELINUX=enforcing
```

   to this:

```
SELINUX=permissive
```

   (c) Set SELinux to permissive starting immediately, without requiring a system reboot.

```
setenforce permissive
```

6. Set up the database. The following command creates the "cloud" user on the database.

```
cloudstack-setup-databases cloud:<dbpassword>@localhost --deploy-as=root:
↪<password> -e <encryption_type> -m <management_server_key> -k <database_
↪key> -i <management_server_ip>
```

   - In dbpassword, specify the password to be assigned to the "cloud" user. You can choose to provide no password although that is not recommended.
   - In deploy-as, specify the username and password of the user deploying the database. In the following command, it is assumed the root user is deploying the database and creating the "cloud" user.
   - (Optional) For encryption_type, use file or web to indicate the technique used to pass in the database encryption password. Default: file. See *About Password and Key Encryption*.
   - (Optional) For management_server_key, substitute the default key that is used to encrypt confidential parameters in the CloudStack properties file. Default: password. It is highly recommended that you replace this with a more secure value. See *About Password and Key Encryption*.
   - (Optional) For database_key, substitute the default key that is used to encrypt confidential parameters in the CloudStack database. Default: password. It is highly recommended that you replace this with a more secure value. See *About Password and Key Encryption*.
   - (Optional) For management_server_ip, you may explicitly specify cluster management server node IP. If not specified, the local IP address will be used.

   When this script is finished, you should see a message like "Successfully initialized the database."

   Note: If the script is unable to connect to the MySQL database, check the "localhost" loopback address in `/etc/hosts`. It should be pointing to the IPv4 loopback address "127.0.0.1" and not the IPv6 loopback address `::1`. Alternatively, reconfigure MySQL to bind to the IPv6 loopback interface.

7. If you are running the KVM hypervisor on the same machine with the Management Server, edit /etc/sudoers and add the following line:

```
Defaults:cloud !requiretty
```

8. Now that the database is set up, you can finish configuring the OS for the Management Server. This command will set up iptables, sudoers, and start the Management Server.

```
cloudstack-setup-management
```

   You should get the output message "CloudStack Management Server setup is done." If the servlet container is Tomcat7 the argument –tomcat7 must be used.

Install the Database on a Separate Node

This section describes how to install MySQL on a standalone machine, separate from the Management Server. This technique is intended for a deployment that includes several Management Server nodes. If you have a single-node Management Server deployment, you will typically use the same node for MySQL. See *"Install the Database on the Management Server Node"*.

Note: The management server doesn't require a specific distribution for the MySQL node. You can use a distribution or Operating System of your choice. Using the same distribution as the management server is recommended, but not required. See *"Management Server, Database, and Storage System Requirements"*.

1. Install MySQL from the package repository from your distribution:

```
yum install mysql-server
```

```
sudo apt-get install mysql-server
```

2. Edit the MySQL configuration (/etc/my.cnf or /etc/mysql/my.cnf, depending on your OS) and insert the following lines in the [mysqld] section. You can put these lines below the datadir line. The max_connections parameter should be set to 350 multiplied by the number of Management Servers you are deploying. This example assumes two Management Servers.

   Note: On Ubuntu, you can also create /etc/mysql/conf.d/cloudstack.cnf file and add these directives there. Don't forget to add [mysqld] on the first line of the file.

```
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
max_connections=700
log-bin=mysql-bin
binlog-format = 'ROW'
bind-address = 0.0.0.0
```

3. Start or restart MySQL to put the new configuration into effect.

   On RHEL/CentOS, MySQL doesn't automatically start after installation. Start it manually.

```
service mysqld start
```

   On Ubuntu, restart MySQL.

```
sudo service mysql restart
```

4. (CentOS and RHEL only; not required on Ubuntu)

   Warning: On RHEL and CentOS, MySQL does not set a root password by default. It is very strongly recommended that you set a root password as a security precaution. Run the following command to secure your installation. You can answer "Y" to all questions except "Disallow root login remotely?". Remote root login is required to set up the databases.

```
mysql_secure_installation
```

5. If a firewall is present on the system, open TCP port 3306 so external MySQL connections can be established.

   On Ubuntu, UFW is the default firewall. Open the port with this command:

```
ufw allow mysql
```

   On RHEL/CentOS:

   (a) Edit the /etc/sysconfig/iptables file and add the following line at the beginning of the INPUT chain.

```
-A INPUT -p tcp --dport 3306 -j ACCEPT
```

(b) Now reload the iptables rules.

```
service iptables restart
```

6. Return to the root shell on your first Management Server.
7. Set up the database. The following command creates the cloud user on the database.
   - In dbpassword, specify the password to be assigned to the cloud user. You can choose to provide no password.
   - In deploy-as, specify the username and password of the user deploying the database. In the following command, it is assumed the root user is deploying the database and creating the cloud user.
   - (Optional) For encryption_type, use file or web to indicate the technique used to pass in the database encryption password. Default: file. See *About Password and Key Encryption*.
   - (Optional) For management_server_key, substitute the default key that is used to encrypt confidential parameters in the CloudStack properties file. Default: password. It is highly recommended that you replace this with a more secure value. See About Password and Key Encryption.
   - (Optional) For database_key, substitute the default key that is used to encrypt confidential parameters in the CloudStack database. Default: password. It is highly recommended that you replace this with a more secure value. See *About Password and Key Encryption*.
   - (Optional) For management_server_ip, you may explicitly specify cluster management server node IP. If not specified, the local IP address will be used.

```
cloudstack-setup-databases cloud:<dbpassword>@<ip address mysql server> -
↪-deploy-as=root:<password> -e <encryption_type> -m <management_server_
↪key> -k <database_key> -i <management_server_ip>
```

When this script is finished, you should see a message like "Successfully initialized the database."

8. Now that the database is set up, you can finish configuring the OS for the Management Server. This command will set up iptables, sudoers, and start the Management Server.

```
cloudstack-setup-management
```

You should get the output message "CloudStack Management Server setup is done."

Prepare NFS Shares

CloudStack needs a place to keep primary and secondary storage (see Cloud Infrastructure Overview). Both of these can be NFS shares. This section tells how to set up the NFS shares before adding the storage to CloudStack.

Note: NFS is not the only option for primary or secondary storage. For example, you may use Ceph RBD, GlusterFS, iSCSI, and others. The choice of storage system will depend on the choice of hypervisor and whether you are dealing with primary or secondary storage.

The requirements for primary and secondary storage are described in:

- *Primary Storage*
- *Secondary Storage*

A production installation typically uses a separate NFS server. See *Using a Separate NFS Server*.

You can also use the Management Server node as the NFS server. This is more typical of a trial installation, but is technically possible in a larger deployment. See *Using the Management Server as the NFS Server*.

Using a Separate NFS Server

This section tells how to set up NFS shares for secondary and (optionally) primary storage on an NFS server running on a separate node from the Management Server.

The exact commands for the following steps may vary depending on your operating system version.

Warning: (KVM only) Ensure that no volume is already mounted at your NFS mount point.

1. On the storage server, create an NFS share for secondary storage and, if you are using NFS for primary storage as well, create a second NFS share. For example:

```
mkdir -p /export/primary
mkdir -p /export/secondary
```

2. To configure the new directories as NFS exports, edit /etc/exports. Export the NFS share(s) with rw,async,no_root_squash,no_subtree_check. For example:

```
vi /etc/exports
```

   Insert the following line.

```
/export  *(rw,async,no_root_squash,no_subtree_check)
```

3. Export the /export directory.

```
exportfs -a
```

4. On the management server, create a mount point for secondary storage. For example:

```
mkdir -p /mnt/secondary
```

5. Mount the secondary storage on your Management Server. Replace the example NFS server name and NFS share paths below with your own.

```
mount -t nfs nfsservername:/export/secondary /mnt/secondary
```

Using the Management Server as the NFS Server

This section tells how to set up NFS shares for primary and secondary storage on the same node with the Management Server. This is more typical of a trial installation, but is technically possible in a larger deployment. It is assumed that you will have less than 16TB of storage on the host.

The exact commands for the following steps may vary depending on your operating system version.

1. On RHEL/CentOS systems, you'll need to install the nfs-utils package:

```
yum install nfs-utils
```

2. On the Management Server host, create two directories that you will use for primary and secondary storage. For example:

```
mkdir -p /export/primary
mkdir -p /export/secondary
```

3. To configure the new directories as NFS exports, edit /etc/exports. Export the NFS share(s) with rw,async,no_root_squash,no_subtree_check. For example:

```
vi /etc/exports
```

Insert the following line.

```
/export  *(rw,async,no_root_squash,no_subtree_check)
```

4. Export the /export directory.

```
exportfs -a
```

5. Edit the /etc/sysconfig/nfs file.

```
vi /etc/sysconfig/nfs
```

Uncomment the following lines:

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

6. Edit the /etc/sysconfig/iptables file.

```
vi /etc/sysconfig/iptables
```

Add the following lines at the beginning of the INPUT chain, where <NETWORK> is the network that you'll be using:

```
-A INPUT -s <NETWORK> -m state --state NEW -p udp --dport 111 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p tcp --dport 111 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p tcp --dport 2049 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p tcp --dport 32803 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p udp --dport 32769 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p tcp --dport 892 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p udp --dport 892 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p tcp --dport 875 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p udp --dport 875 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p tcp --dport 662 -j ACCEPT
-A INPUT -s <NETWORK> -m state --state NEW -p udp --dport 662 -j ACCEPT
```

7. Run the following commands:

```
service iptables restart
service iptables save
```

8. If NFS v4 communication is used between client and server, add your domain to /etc/idmapd.conf on both the hypervisor host and Management Server.

```
vi /etc/idmapd.conf
```

Remove the character # from the beginning of the Domain line in idmapd.conf and replace the value in the file with your own domain. In the example below, the domain is company.com.

```
Domain = company.com
```

9. Reboot the Management Server host.

   Two NFS shares called /export/primary and /export/secondary are now set up.

10. It is recommended that you test to be sure the previous steps have been successful.

    (a) Log in to the hypervisor host.

    (b) Be sure NFS and rpcbind are running. The commands might be different depending on your OS. For example:

    ```
    service rpcbind start
    service nfs start
    chkconfig nfs on
    chkconfig rpcbind on
    reboot
    ```

    (c) Log back in to the hypervisor host and try to mount the /export directories. For example, substitute your own management server name:

    ```
    mkdir /primary
    mount -t nfs <management-server-name>:/export/primary
    umount /primary
    mkdir /secondary
    mount -t nfs <management-server-name>:/export/secondary
    umount /secondary
    ```

Additional Management Servers

For your second and subsequent Management Servers, you will install the Management Server software, connect it to the database, and set up the OS for the Management Server.

1. Perform the steps in *"Prepare the Operating System"* and "Building RPMs from Source" or "Building DEB packages" as appropriate.

2. This step is required only for installations where XenServer is installed on the hypervisor hosts.

   Download vhd-util from vhd-util

   Copy vhd-util to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

3. Ensure that necessary services are started and set to start on boot.

   ```
   service rpcbind start
   service nfs start
   chkconfig nfs on
   chkconfig rpcbind on
   ```

4. Configure the database client. Note the absence of the –deploy-as argument in this case. (For more details about the arguments to this command, see *Install the Database on a Separate Node*.)

   ```
   cloudstack-setup-databases cloud:dbpassword@dbhost -e encryption_type -m␣
   ↪management_server_key -k database_key -i management_server_ip
   ```

5. Configure the OS and start the Management Server:

```
cloudstack-setup-management
```

   The Management Server on this node should now be running. If the servlet container is Tomcat7 the argument –tomcat7 must be used.

6. Repeat these steps on each additional Management Server.
7. Be sure to configure a load balancer for the Management Servers. See *Management Server Load Balancing*

Prepare the System VM Template

Secondary storage must be seeded with a template that is used for CloudStack system VMs.

Note: When copying and pasting a command, be sure the command has pasted as a single line before executing. Some document viewers may introduce unwanted line breaks in copied text.

---

1. On the Management Server, run one or more of the following `cloud-install-sys-tmplt` commands to retrieve and decompress the system VM template. Run the command for each hypervisor type that you expect end users to run in this Zone.

   If your secondary storage mount point is not named `/mnt/secondary`, substitute your own mount point name.

   If you set the CloudStack database encryption type to "web" when you set up the database, you must now add the parameter `-s <management-server-secret-key>`. See *About Password and Key Encryption*.

   This process will require approximately 5 GB of free space on the local file system and up to 30 minutes each time it runs.

   - For Hyper-V

     ```
     /usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-
     ↪sys-tmplt -m /mnt/secondary -u http://download.cloudstack.org/
     systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip -h hyperv -s
     ↪<optional-management-server-secret-key> -F
     ```

   - For XenServer:

     ```
     /usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-
     ↪sys-tmplt -m /mnt/secondary -u http://download.cloudstack.org/
     systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 -h xenserver -s
     ↪<optional-management-server-secret-key> -F
     ```

   - For vSphere:

     ```
     /usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-
     ↪sys-tmplt -m /mnt/secondary -u http://download.cloudstack.org/
     systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova -h vmware -s
     ↪<optional-management-server-secret-key> -F
     ```

   - For KVM:

     ```
     /usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-
     ↪sys-tmplt -m /mnt/secondary -u http://download.cloudstack.org/
     systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 -h kvm -s
     ↪<optional-management-server-secret-key> -F
     ```

- For LXC:

```
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-
↪sys-tmplt -m /mnt/secondary -u http://download.cloudstack.org/
systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 -h lxc -s
↪<optional-management-server-secret-key> -F
```

- For OVM3:

```
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-
↪sys-tmplt -m /mnt/secondary -u http://download.cloudstack.org/
systemvm/4.11/systemvmtemplate-4.11.0-ovm.raw.bz2 -h ovm3 -s
↪<optional-management-server-secret-key> -F
```

2. If you are using a separate NFS server, perform this step. If you are using the Management Server as the NFS server, you MUST NOT perform this step.

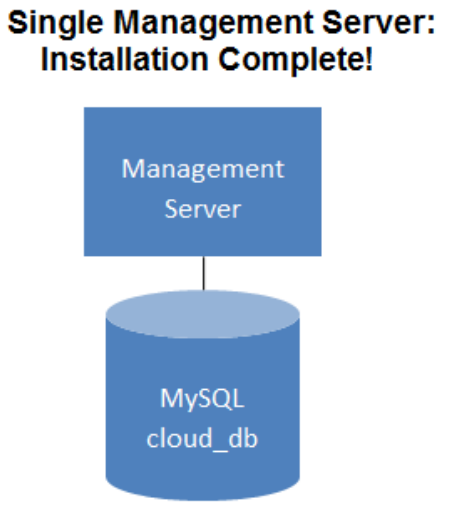
   When the script has finished, unmount secondary storage and remove the created directory.

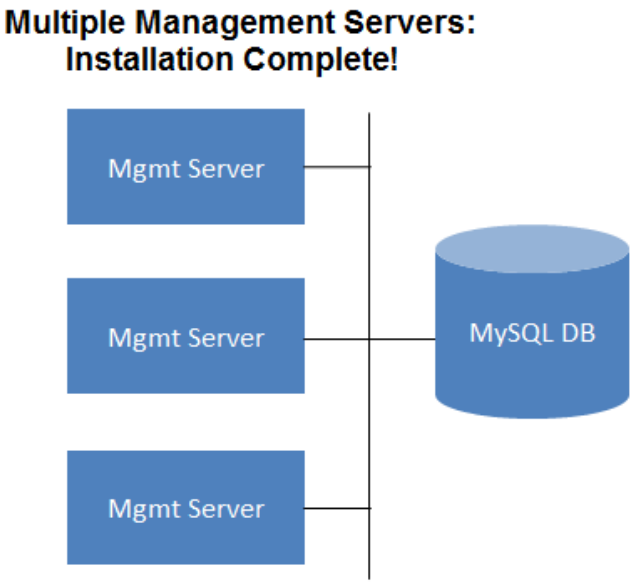
```
umount /mnt/secondary
rmdir /mnt/secondary
```

3. Repeat these steps for each secondary storage server.

Installation Complete! Next Steps

Congratulations! You have now installed CloudStack Management Server and the database it uses to persist system data.





What should you do next?

- Even without adding any cloud infrastructure, you can run the UI to get a feel for what's offered and how you will interact with CloudStack on an ongoing basis. See Log In to the UI.
- When you're ready, add the cloud infrastructure and try running some virtual machines on it, so you can watch how CloudStack manages the infrastructure. See Provision Your Cloud Infrastructure.

3.3 Configuration

3.3.1 Configuring your CloudStack Installation

This section tells how to add regions, zones, pods, clusters, hosts, storage, and networks to your cloud. If you are unfamiliar with these entities, please begin by looking through *Cloud Infrastructure Overview*

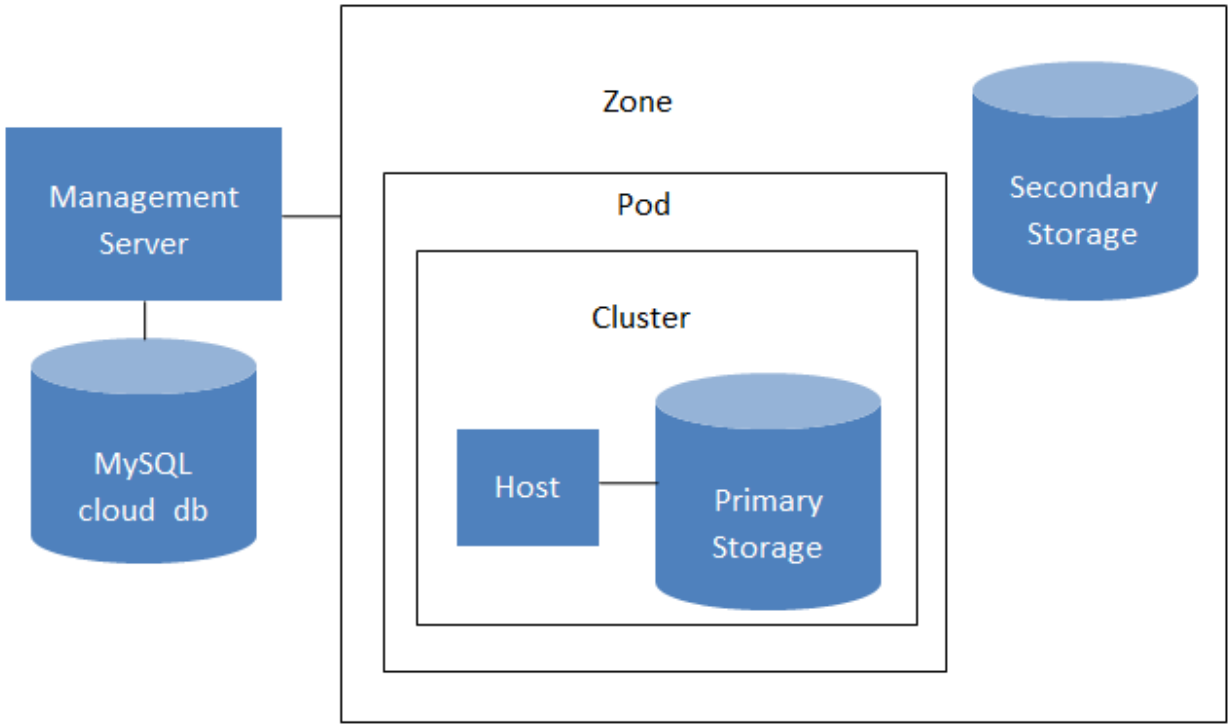
Overview of Provisioning Steps

After the Management Server is installed and running, you can add the compute resources for it to manage. For an overview of how a CloudStack cloud infrastructure is organized, see *Cloud Infrastructure Overview*

To provision the cloud infrastructure, or to scale it up at any time, follow these procedures:

1. Define regions (optional). See *Adding Regions (optional)*.
2. Add a zone to the region. See *Adding a Zone*.
3. Add more pods to the zone (optional). See *Adding a Pod*.
4. Add more clusters to the pod (optional). See *Adding a Cluster*.
5. Add more hosts to the cluster (optional). See *Adding a Host*.
6. Add primary storage to the cluster. See *Add Primary Storage*.
7. Add secondary storage to the zone. See *Add Secondary Storage*.
8. Initialize and test the new cloud. See *Initialize and Test*.

When you have finished these steps, you will have a deployment with the following basic structure:



Conceptual view of a basic deployment

Adding Regions (optional)

Grouping your cloud resources into geographic regions is an optional step when provisioning the cloud. For an overview of regions, see *About Regions*

The First Region: The Default Region

If you do not take action to define regions, then all the zones in your cloud will be automatically grouped into a single default region. This region is assigned the region ID of 1. You can change the name or URL of the default region by displaying the region in the CloudStack UI and clicking the Edit button.

Adding a Region

Use these steps to add a second region in addition to the default region.

1. Each region has its own CloudStack instance. Therefore, the first step of creating a new region is to install the Management Server software, on one or more nodes, in the geographic area where you want to set up the new region. Use the steps in the Installation guide. When you come to the step where you set up the database, use the additional command-line flag `-r <region_id>` to set a region ID for the new region. The default region is automatically assigned a region ID of 1, so your first additional region might be region 2.

```
# cloudstack-setup-databases cloud:<dbpassword>@localhost --deploy-as=root:
↪<password> -e <encryption_type> -m <management_server_key> -k <database_key> -r
↪<region_id>
```

(continues on next page)

(continued from previous page)

2. By the end of the installation procedure, the Management Server should have been started. Be sure that the Management Server installation was successful and complete.
3. Now add the new region to region 1 in CloudStack.
   (a) Log in to CloudStack in the first region as root administrator (that is, log in to <region.1.IP.address>:8080/client).
   (b) In the left navigation bar, click Regions.
   (c) Click Add Region. In the dialog, fill in the following fields:
      - ID. A unique identifying number. Use the same number you set in the database during Management Server installation in the new region; for example, 2.
      - Name. Give the new region a descriptive name.
      - Endpoint. The URL where you can log in to the Management Server in the new region. This has the format <region.2.IP.address>:8080/client.
4. Now perform the same procedure in reverse. Log in to region 2, and add region 1.
5. Copy the account, user, and domain tables from the region 1 database to the region 2 database.

   In the following commands, it is assumed that you have set the root password on the database, which is a CloudStack recommended best practice. Substitute your own MySQL root password.

   (a) First, run this command to copy the contents of the database:

   ```
   # mysqldump -u root -p<mysql_password> -h <region1_db_host> cloud account␣
   ↪user domain > region1.sql
   ```

   (b) Then run this command to put the data onto the region 2 database:

   ```
   # mysql -u root -p<mysql_password> -h <region2_db_host> cloud < region1.sql
   ```

6. Remove project accounts. Run these commands on the region 2 database:

   ```
   # mysql> delete from account where type = 5;
   ```

7. Set the default zone as null:

   ```
   # mysql> update account set default_zone_id = null;
   ```

8. Restart the Management Servers in region 2.

Adding Third and Subsequent Regions

To add the third region, and subsequent additional regions, the steps are similar to those for adding the second region. However, you must repeat certain steps additional times for each additional region:

1. Install CloudStack in each additional region. Set the region ID for each region during the database setup step.

   ```
   cloudstack-setup-databases cloud:<dbpassword>@localhost --deploy-as=root:
   ↪<password> -e <encryption_type> -m <management_server_key> -k <database_key> -r
   ↪<region_id>
   ```

2. Once the Management Server is running, add your new region to all existing regions by repeatedly using the Add Region button in the UI. For example, if you were adding region 3:
   (a) Log in to CloudStack in the first region as root administrator (that is, log in to <region.1.IP.address>:8080/client), and add a region with ID 3, the name of region 3, and the endpoint <region.3.IP.address>:8080/client.
   (b) Log in to CloudStack in the second region as root administrator (that is, log in to <region.2.IP.address>:8080/client), and add a region with ID 3, the name of region 3, and the endpoint <region.3.IP.address>:8080/client.
3. Repeat the procedure in reverse to add all existing regions to the new region. For example, for the third region, add the other two existing regions:
   (a) Log in to CloudStack in the third region as root administrator (that is, log in to <region.3.IP.address>:8080/client).
   (b) Add a region with ID 1, the name of region 1, and the endpoint <region.1.IP.address>:8080/client.
   (c) Add a region with ID 2, the name of region 2, and the endpoint <region.2.IP.address>:8080/client.
4. Copy the account, user, and domain tables from any existing region's database to the new region's database.

   In the following commands, it is assumed that you have set the root password on the database, which is a CloudStack recommended best practice. Substitute your own MySQL root password.

   (a) First, run this command to copy the contents of the database:

   ```
   # mysqldump -u root -p<mysql_password> -h <region1_db_host> cloud account␣
   ↪user domain > region1.sql
   ```

   (b) Then run this command to put the data onto the new region's database. For example, for region 3:

   ```
   # mysql -u root -p<mysql_password> -h <region3_db_host> cloud < region1.sql
   ```

5. Remove project accounts. Run these commands on the region 3 database:

   ```
   mysql> delete from account where type = 5;
   ```

6. Set the default zone as null:

   ```
   mysql> update account set default_zone_id = null;
   ```

7. Restart the Management Servers in the new region.

Deleting a Region

Log in to each of the other regions, navigate to the one you want to delete, and click Remove Region. For example, to remove the third region in a 3-region cloud:

1. Log in to <region.1.IP.address>:8080/client.
2. In the left navigation bar, click Regions.
3. Click the name of the region you want to delete.
4. Click the Remove Region button.
5. Repeat these steps for <region.2.IP.address>:8080/client.

Adding a Zone

When you add a new zone, you will be prompted to configure the zone's physical network and add the first pod, cluster, host, primary storage, and secondary storage.

1. Log in to the CloudStack UI as the root administrator. See *Log In to the UI*.
2. In the left navigation, choose Infrastructure.
3. On Zones, click View More.
4. Click Add Zone. The zone creation wizard will appear.
5. Choose one of the following network types:
   - **Basic.** For AWS-style networking. Provides a single network where each VM instance is assigned an IP directly from the network. Guest isolation can be provided through layer-3 means such as security groups (IP address source filtering).
   - **Advanced.** For more sophisticated network topologies. This network model provides the most flexibility in defining guest networks and providing custom network offerings such as firewall, VPN, or load balancer support.
6. The rest of the steps differ depending on whether you chose Basic or Advanced. Continue with the steps that apply to you:
   - *"Basic Zone Configuration"*
   - *"Advanced Zone Configuration"*

Basic Zone Configuration

1. After you select Basic in the Add Zone wizard and click Next, you will be asked to enter the following details. Then click Next.
   - **Name.** A name for the zone.
   - **DNS 1 and 2.** These are DNS servers for use by guest VMs in the zone. These DNS servers will be accessed via the public network you will add later. The public IP addresses for the zone must have a route to the DNS server named here.
   - **Internal DNS 1 and Internal DNS 2.** These are DNS servers for use by system VMs in the zone (these are VMs used by CloudStack itself, such as virtual routers, console proxies, and Secondary Storage VMs.) These DNS servers will be accessed via the management traffic network interface of the System VMs. The private IP address you provide for the pods must have a route to the internal DNS server named here.
   - **Hypervisor.** (Introduced in version 3.0.1) Choose the hypervisor for the first cluster in the zone. You can add clusters with different hypervisors later, after you finish adding the zone.
   - **Network Offering.** Your choice here determines what network services will be available on the network for guest VMs.

Network Offering	Description
DefaultSharedNetworkOfferingWithSGService	If you want to enable security groups for guest traffic isolation, choose this. (See Using Security Groups to Control Traffic to VMs.)
DefaultSharedNetworkOffering	If you do not need security groups, choose this.
DefaultSharedNetscalerEIPandELBNetworkOffering	If you have installed a Citrix NetScaler appliance as part of your zone network, and you will be using its Elastic IP and Elastic Load Balancing features, choose this. With the EIP and ELB features, a basic zone with security groups enabled can offer 1:1 static NAT and load balancing.

- **Network Domain.** (Optional) If you want to assign a special domain name to the guest VM network, specify the DNS suffix.
- **Public.** A public zone is available to all users. A zone that is not public will be assigned to a particular domain. Only users in that domain will be allowed to create guest VMs in this zone.

2. Choose which traffic types will be carried by the physical network.

   The traffic types are management, public, guest, and storage traffic. For more information about the types, roll over the icons to display their tool tips, or see Basic Zone Network Traffic Types. This screen starts out with some traffic types already assigned. To add more, drag and drop traffic types onto the network. You can also change the network name if desired.

3. Assign a network traffic label to each traffic type on the physical network. These labels must match the labels you have already defined on the hypervisor host. To assign each label, click the Edit button under the traffic type icon. A popup dialog appears where you can type the label, then click OK.

   These traffic labels will be defined only for the hypervisor selected for the first cluster. For all other hypervisors, the labels can be configured after the zone is created.

4. Click Next.

5. (NetScaler only) If you chose the network offering for NetScaler, you have an additional screen to fill out. Provide the requested details to set up the NetScaler, then click Next.

   - **IP address.** The NSIP (NetScaler IP) address of the NetScaler device.
   - **Username/Password.** The authentication credentials to access the device. CloudStack uses these credentials to access the device.
   - **Type.** NetScaler device type that is being added. It could be NetScaler VPX, NetScaler MPX, or NetScaler SDX. For a comparison of the types, see About Using a NetScaler Load Balancer.
   - **Public interface.** Interface of NetScaler that is configured to be part of the public network.
   - **Private interface.** Interface of NetScaler that is configured to be part of the private network.
   - **Number of retries.** Number of times to attempt a command on the device before considering the operation failed. Default is 2.
   - **Capacity.** Number of guest networks/accounts that will share this NetScaler device.
   - **Dedicated.** When marked as dedicated, this device will be dedicated to a single account. When Dedicated is checked, the value in the Capacity field has no significance – implicitly, its value is 1.

6. (NetScaler only) Configure the IP range for public traffic. The IPs in this range will be used for the static NAT capability which you enabled by selecting the network offering for NetScaler with EIP and ELB. Enter the

following details, then click Add. If desired, you can repeat this step to add more IP ranges. When done, click Next.

- **Gateway.** The gateway in use for these IP addresses.
- **Netmask.** The netmask associated with this IP range.
- **VLAN.** The VLAN that will be used for public traffic.
- **Start IP/End IP.** A range of IP addresses that are assumed to be accessible from the Internet and will be allocated for access to guest VMs.

7. In a new zone, CloudStack adds the first pod for you. You can always add more pods later. For an overview of what a pod is, see *About Pods*

   To configure the first pod, enter the following, then click Next:

   - **Pod Name.** A name for the pod.
   - **Reserved system gateway.** The gateway for the hosts in that pod.
   - **Reserved system netmask.** The network prefix that defines the pod's subnet. Use CIDR notation.
   - **Start/End Reserved System IP.** The IP range in the management network that CloudStack uses to manage various system VMs, such as Secondary Storage VMs, Console Proxy VMs, and DHCP. For more information, see System Reserved IP Addresses.

8. Configure the network for guest traffic. Provide the following, then click Next:

   - **Guest gateway.** The gateway that the guests should use.
   - **Guest netmask.** The netmask in use on the subnet the guests will use.
   - **Guest start IP/End IP.** Enter the first and last IP addresses that define a range that CloudStack can assign to guests.
     - We strongly recommend the use of multiple NICs. If multiple NICs are used, they may be in a different subnet.
     - If one NIC is used, these IPs should be in the same CIDR as the pod CIDR.

9. In a new pod, CloudStack adds the first cluster for you. You can always add more clusters later. For an overview of what a cluster is, see About Clusters.

   To configure the first cluster, enter the following, then click Next:

   - **Hypervisor.** (Version 3.0.0 only; in 3.0.1, this field is read only) Choose the type of hypervisor software that all hosts in this cluster will run. If you choose VMware, additional fields appear so you can give information about a vSphere cluster. For vSphere servers, we recommend creating the cluster of hosts in vCenter and then adding the entire cluster to CloudStack. See Add Cluster: vSphere.
   - **Cluster name.** Enter a name for the cluster. This can be text of your choosing and is not used by CloudStack.

10. In a new cluster, CloudStack adds the first host for you. You can always add more hosts later. For an overview of what a host is, see About Hosts.

    Note: When you add a hypervisor host to CloudStack, the host must not have any VMs already running.

    Before you can configure the host, you need to install the hypervisor software on the host. You will need to know which version of the hypervisor software version is supported by CloudStack and what additional configuration is required to ensure the host will work with CloudStack. To find these installation details, see:

    - Citrix XenServer Installation and Configuration

- VMware vSphere Installation and Configuration
- KVM vSphere Installation and Configuration

To configure the first host, enter the following, then click Next:

- **Host Name.** The DNS name or IP address of the host.
- **Username.** The username is root.
- **Password.** This is the password for the user named above (from your XenServer or KVM install).
- **Host Tags.** (Optional) Any labels that you use to categorize hosts for ease of maintenance. For example, you can set this to the cloud's HA tag (set in the ha.tag global configuration parameter) if you want this host to be used only for VMs with the "high availability" feature enabled. For more information, see HA-Enabled Virtual Machines as well as HA for Hosts.

11. In a new cluster, CloudStack adds the first primary storage server for you. You can always add more servers later. For an overview of what primary storage is, see About Primary Storage.

    To configure the first primary storage server, enter the following, then click Next:

    - **Name.** The name of the storage device.
    - **Protocol.** For XenServer, choose either NFS, iSCSI, or PreSetup. For KVM, choose NFS, SharedMountPoint,CLVM, or RBD. For vSphere choose either VMFS (iSCSI or FiberChannel) or NFS. The remaining fields in the screen vary depending on what you choose here.

Advanced Zone Configuration

1. After you select Advanced in the Add Zone wizard and click Next, you will be asked to enter the following details. Then click Next.

    - **Name.** A name for the zone.
    - **DNS 1 and 2.** These are DNS servers for use by guest VMs in the zone. These DNS servers will be accessed via the public network you will add later. The public IP addresses for the zone must have a route to the DNS server named here.
    - **Internal DNS 1 and Internal DNS 2.** These are DNS servers for use by system VMs in the zone(these are VMs used by CloudStack itself, such as virtual routers, console proxies,and Secondary Storage VMs.) These DNS servers will be accessed via the management traffic network interface of the System VMs. The private IP address you provide for the pods must have a route to the internal DNS server named here.
    - **Network Domain.** (Optional) If you want to assign a special domain name to the guest VM network, specify the DNS suffix.
    - **Guest CIDR.** This is the CIDR that describes the IP addresses in use in the guest virtual networks in this zone. For example, 10.1.1.0/24. As a matter of good practice you should set different CIDRs for different zones. This will make it easier to set up VPNs between networks in different zones.
    - **Hypervisor.** (Introduced in version 3.0.1) Choose the hypervisor for the first cluster in the zone. You can add clusters with different hypervisors later, after you finish adding the zone.
    - **Public.** A public zone is available to all users. A zone that is not public will be assigned to a particular domain. Only users in that domain will be allowed to create guest VMs in this zone.

2. Choose which traffic types will be carried by the physical network.

    The traffic types are management, public, guest, and storage traffic. For more information about the types, roll over the icons to display their tool tips, or see *Advanced Zone Network Traffic Types*. This screenstarts out with one network already configured. If you have multiple physical networks, you need to add more. Drag and drop

traffic types onto a greyed-out network and it will become active. You can move the traffic icons from one network to another; for example, if the default traffic types shown for Network 1 do not match your actual setup, you can move them down. You can also change the network names if desired.

3. (Introduced in version 3.0.1) Assign a network traffic label to each traffic type on each physical network. These labels must match the labels you have already defined on the hypervisor host. To assign each label, click the Edit button under the traffic type icon within each physical network. A popup dialog appears where you can type the label, then click OK.

   These traffic labels will be defined only for the hypervisor selected for the first cluster. For all other hypervisors, the labels can be configured after the zone is created.

   (VMware only) If you have enabled Nexus dvSwitch in the environment, you must specify the corresponding Ethernet port profile names as network traffic label for each traffic type on the physical network. For more information on Nexus dvSwitch, see Configuring a vSphere Cluster with Nexus 1000v Virtual Switch in the Installation Guide. If you have enabled VMware dvSwitch in the environment, you must specify the corresponding Switch name as network traffic label for each traffic type on the physical network. For more information, see Configuring a VMware Datacenter with VMware Distributed Virtual Switch in the Installation Guide.

4. Click Next.

5. Configure the IP range for public Internet traffic. Enter the following details, then click Add. If desired, you can repeat this step to add more public Internet IP ranges. When done, click Next.

   - **Gateway.** The gateway in use for these IP addresses.
   - **Netmask.** The netmask associated with this IP range.
   - **VLAN.** The VLAN that will be used for public traffic.
   - **Start IP/End IP.** A range of IP addresses that are assumed to be accessible from the Internet and will be allocated for access to guest networks.

6. In a new zone, CloudStack adds the first pod for you. You can always add more pods later. For an overview of what a pod is, see *About Pods*

   To configure the first pod, enter the following, then click Next:

   - **Pod Name.** A name for the pod.
   - **Reserved system gateway.** The gateway for the hosts in that pod.
   - **Reserved system netmask.** The network prefix that defines the pod's subnet. Use CIDR notation.
   - **Start/End Reserved System IP.** The IP range in the management network that CloudStack uses to manage various system VMs, such as Secondary Storage VMs, Console Proxy VMs, and DHCP. For more information, see *System Reserved IP Addresses*

7. Specify a range of VLAN IDs to carry guest traffic for each physical network (see VLAN Allocation Example), then click Next.

8. In a new pod, CloudStack adds the first cluster for you. You can always add more clusters later. For an overview of what a cluster is, see *About Clusters*

   To configure the first cluster, enter the following, then click Next:

   - **Hypervisor.** (Version 3.0.0 only; in 3.0.1, this field is read only) Choose the type of hypervisor software that all hosts in this cluster will run. If you choose VMware, additional fields appear so you can give information about a vSphere cluster. For vSphere servers, we recommend creating the cluster of hosts in vCenter and then adding the entire cluster to CloudStack. See Add Cluster: vSphere .
   - **Cluster name.** Enter a name for the cluster. This can be text of your choosing and is not used by CloudStack.

9. In a new cluster, CloudStack adds the first host for you. You can always add more hosts later. For an overview of what a host is, see *About Hosts*.

   Note: When you deploy CloudStack, the hypervisor host must not have any VMs already running.

   Before you can configure the host, you need to install the hypervisor software on the host. You will need to know which version of the hypervisor software version is supported by CloudStack and what additional configuration is required to ensure the host will work with CloudStack. To find these installation details, see:

   - Citrix XenServer Installation for CloudStack
   - VMware vSphere Installation and Configuration
   - KVM Installation and Configuration

   To configure the first host, enter the following, then click Next:

   - **Host Name.** The DNS name or IP address of the host.
   - **Username.** Usually root.
   - **Password.** This is the password for the user named above (from your XenServer or KVM install).
   - **Host Tags.** (Optional) Any labels that you use to categorize hosts for ease of maintenance. For example, you can set to the cloud's HA tag (set in the ha.tag global configuration parameter) if you want this host to be used only for VMs with the "high availability" feature enabled. For more information, see HA-Enabled Virtual Machines as well as HA for Hosts, both in the Administration Guide.

10. In a new cluster, CloudStack adds the first primary storage server for you. You can always add more servers later. For an overview of what primary storage is, see *Primary Storage*

    To configure the first primary storage server, enter the following, then click Next:

    - **Name.** The name of the storage device.
    - **Protocol.** For XenServer, choose either NFS, iSCSI, or PreSetup. For KVM, choose NFS, SharedMountPoint, CLVM, and RBD. For vSphere choose either VMFS (iSCSI or FiberChannel) or NFS. The remaining fields in the screen vary depending on what you choose here.

NFS	– **Server.** The IP address or DNS name of the storage device. – **Path.** The exported path from the server. – **Tags (optional).** The comma-separated list of tags for this storage device. It should be an equivalent set or superset of the tags on your disk offerings.
iSCSI	– **Server.** The IP address or DNS name of the storage device. – **Target IQN.** The IQN of the target. For example, iqn.1986-03.com.sun:02:01ec9bb549-1271378984. – **Lun.** The LUN number. For example, 3. – **Tags (optional).** The comma-separated list of tags for this storage device. It should be an equivalent set or superset of the tags on your disk offerings.
preSetup	– **Server.** The IP address or DNS name of the storage device. – **SR Name-Label.** Enter the name-label of the SR that has been set up outside CloudStack. – **Tags (optional).** The comma-separated list of tags for this storage device. It should be an equivalent set or superset of the tags on your disk offerings.
SharedMountPoint	– **Path.** The path on each host that is where this primary storage is mounted. For example, "/mnt/primary". – **Tags (optional).** The comma-separated list of tags for this storage device. It should be an equivalent set or superset of the tags on your disk offerings.
VMFS	– **Server.** The IP address or DNS name of the vCenter server. – **Path.** A combination of the datacenter name and the datastore name. The format is "/" datacenter name "/" datastore name. For example, "/cloud.dc.VM/cluster1datastore". – **Tags (optional).** The comma-separated list of tags for this storage device. It should be an equivalent set or superset of the tags on your disk offerings.

The tag sets on primary storage across clusters in a Zone must be identical. For example, if cluster A provides primary storage that has tags T1 and T2, all other clusters in the Zone must also provide primary

storage that has tags T1 and T2.

1. In a new zone, CloudStack connects the first secondary storage server for you. For an overview of what secondary storage is, see *Secondary Storage*

   Before you can fill out this screen, you need to prepare the secondary storage by setting up NFS shares and installing the latest CloudStack System VM template. See Adding Secondary Storage :

   - **NFS Server.** The IP address of the server or fully qualified domain name of the server.
   - **Path.** The exported path from the server.

2. Click Launch.

Adding a Pod

When you created a new zone, CloudStack adds the first pod for you. You can add more pods at any time using the procedure in this section.

1. Log in to the CloudStack UI. See *Log In to the UI*.
2. In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone to which you want to add a pod.
3. Click the Compute and Storage tab. In the Pods node of the diagram, click View All.
4. Click Add Pod.
5. Enter the following details in the dialog.
   - **Name.** The name of the pod.
   - **Gateway.** The gateway for the hosts in that pod.
   - **Netmask.** The network prefix that defines the pod's subnet. Use CIDR notation.
   - **Start/End Reserved System IP.** The IP range in the management network that CloudStack uses to manage various system VMs, such as Secondary Storage VMs, Console Proxy VMs, and DHCP. For more information, see System Reserved IP Addresses.
6. Click OK.

Adding a Cluster

You need to tell CloudStack about the hosts that it will manage. Hosts exist inside clusters, so before you begin adding hosts to the cloud, you must add at least one cluster.

Add Cluster: KVM or XenServer

These steps assume you have already installed the hypervisor on the hosts and logged in to the CloudStack UI.

1. In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone in which you want to add the cluster.
2. Click the Compute tab.
3. In the Clusters node of the diagram, click View All.
4. Click Add Cluster.
5. Choose the hypervisor type for this cluster.

6. Choose the pod in which you want to create the cluster.
7. Enter a name for the cluster. This can be text of your choosing and is not used by CloudStack.
8. Click OK.

Add Cluster: vSphere

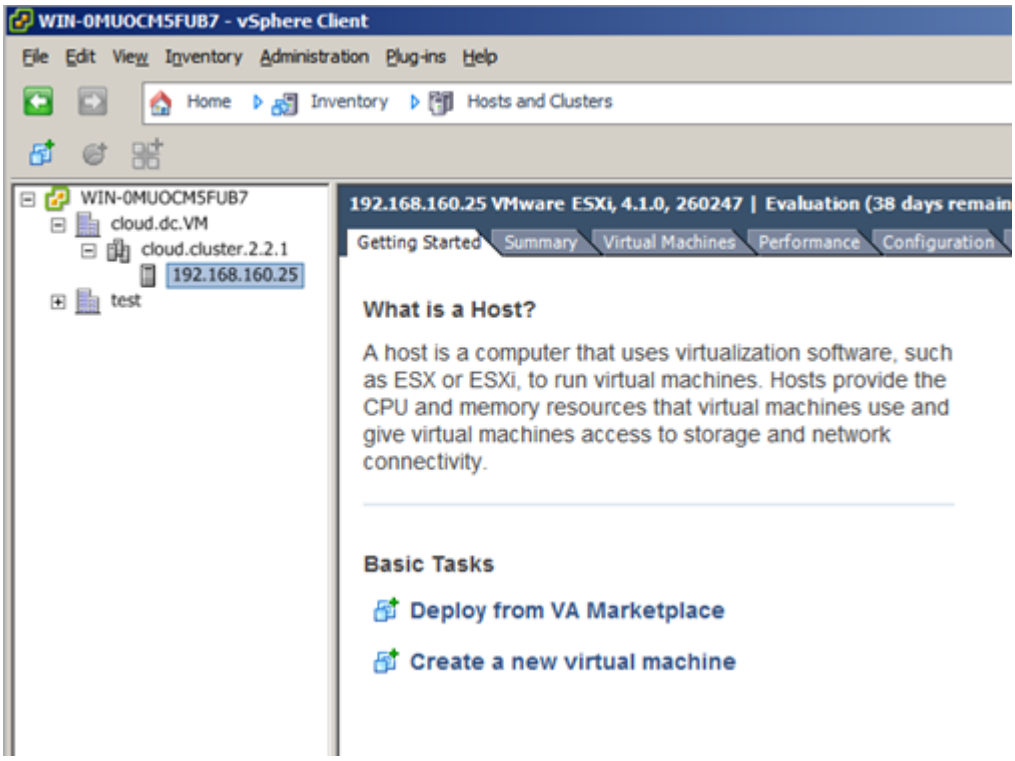
Host management for vSphere is done through a combination of vCenter and the CloudStack admin UI. CloudStack requires that all hosts be in a CloudStack cluster, but the cluster may consist of a single host. As an administrator you must decide if you would like to use clusters of one host or of multiple hosts. Clusters of multiple hosts allow for features like live migration. Clusters also require shared storage such as NFS or iSCSI.

For vSphere servers, we recommend creating the cluster of hosts in vCenter and then adding the entire cluster to CloudStack. Follow these requirements:

- Do not put more than 8 hosts in a vSphere cluster
- Make sure the hypervisor hosts do not have any VMs already running before you add them to CloudStack.

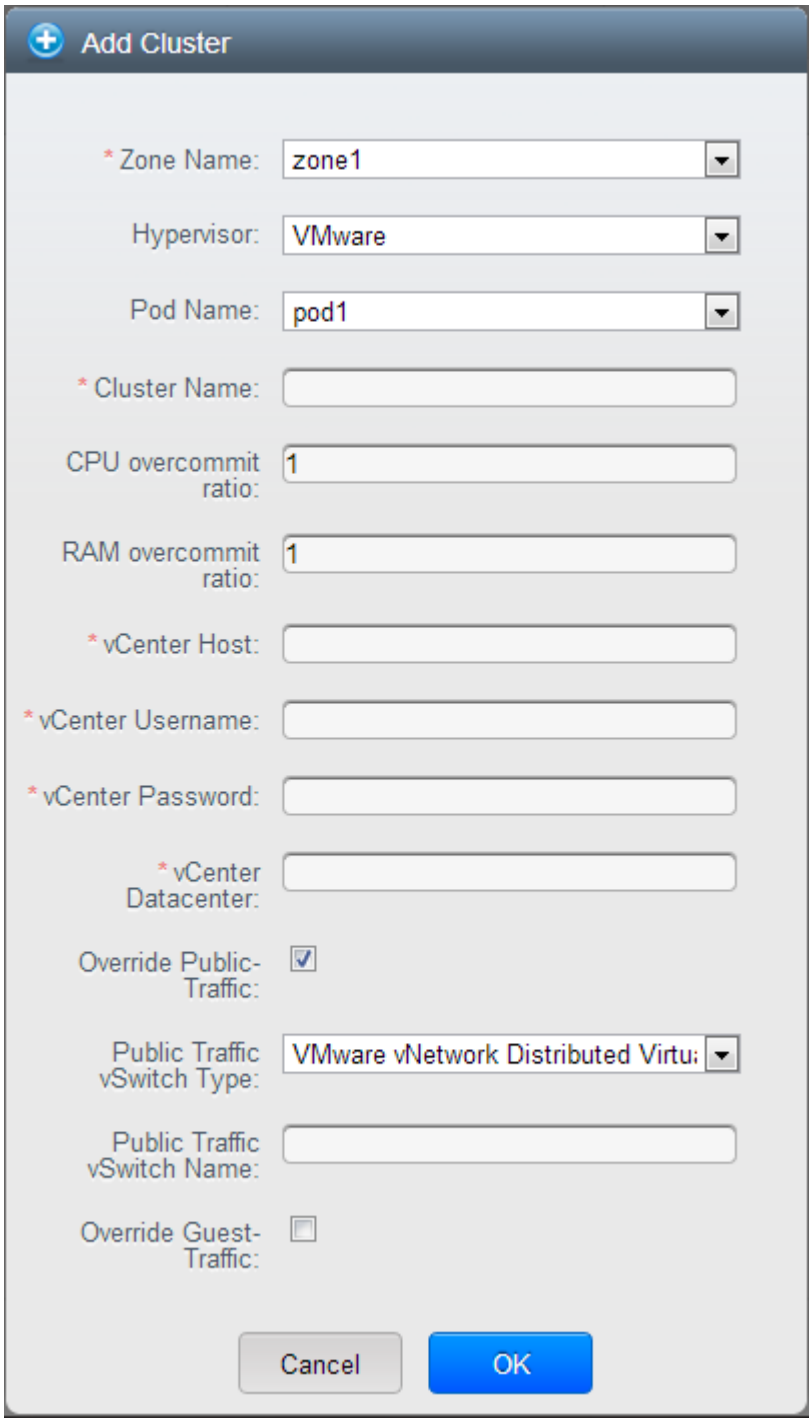
To add a vSphere cluster to CloudStack:

1. Create the cluster of hosts in vCenter. Follow the vCenter instructions to do this. You will create a cluster that looks something like this in vCenter.



2. Log in to the UI.
3. In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone in which you want to add the cluster.
4. Click the Compute tab, and click View All on Pods. Choose the pod to which you want to add the cluster.
5. Click View Clusters.
6. Click Add Cluster.

7. In Hypervisor, choose VMware.
8. Provide the following information in the dialog. The fields below make reference to the values from vCenter.



- **Cluster Name**: Enter the name of the cluster you created in vCenter. For example, "cloud.cluster.2.2.1"
- **vCenter Username**: Enter the username that CloudStack should use to connect to vCenter. This user must have all the administrative privileges.
- **CPU overcommit ratio**: Enter the CPU overcommit ratio for the cluster. The value you enter determines the CPU consumption of each VM in the selected cluster. By increasing the over-provisioning ratio, more

resource capacity will be used. If no value is specified, the value is defaulted to 1, which implies no over-provisioning is done.

- **RAM overcommit ratio**: Enter the RAM overcommit ratio for the cluster. The value you enter determines the memory consumption of each VM in the selected cluster. By increasing the over-provisioning ratio, more resource capacity will be used. If no value is specified, the value is defaulted to 1, which implies no over-provisioning is done.
- **vCenter Host**: Enter the hostname or IP address of the vCenter server.
- **vCenter Password**: Enter the password for the user named above.
- **vCenter Datacenter**: Enter the vCenter datacenter that the cluster is in. For example, "cloud.dc.VM".
- **Override Public Traffic**: Enable this option to override the zone-wide public traffic for the cluster you are creating.
- **Public Traffic vSwitch Type**: This option is displayed only if you enable the Override Public Traffic option. Select a desirable switch. If the vmware.use.dvswitch global parameter is true, the default option will be VMware vNetwork Distributed Virtual Switch.

  If you have enabled Nexus dvSwitch in the environment, the following parameters for dvSwitch configuration are displayed:

  - Nexus dvSwitch IP Address: The IP address of the Nexus VSM appliance.
  - Nexus dvSwitch Username: The username required to access the Nexus VSM appliance.
  - Nexus dvSwitch Password: The password associated with the username specified above.
- **Override Guest Traffic**: Enable this option to override the zone-wide guest traffic for the cluster you are creating.
- **Guest Traffic vSwitch Type**: This option is displayed only if you enable the Override Guest Traffic option. Select a desirable switch.

  If the vmware.use.dvswitch global parameter is true, the default option will be VMware vNetwork Distributed Virtual Switch.

  If you have enabled Nexus dvSwitch in the environment, the following parameters for dvSwitch configuration are displayed:

  - Nexus dvSwitch IP Address: The IP address of the Nexus VSM appliance.
  - Nexus dvSwitch Username: The username required to access the Nexus VSM appliance.
  - Nexus dvSwitch Password: The password associated with the username specified above.
- There might be a slight delay while the cluster is provisioned. It will automatically display in the UI.

Adding a Host

1. Before adding a host to the CloudStack configuration, you must first install your chosen hypervisor on the host. CloudStack can manage hosts running VMs under a variety of hypervisors.

   The CloudStack Installation Guide provides instructions on how to install each supported hypervisor and configure it for use with CloudStack. See the appropriate section in the Installation Guide for information about which version of your chosen hypervisor is supported, as well as crucial additional steps to configure the hypervisor hosts for use with CloudStack.

> **Warning:** Be sure you have performed the additional CloudStack-specific configuration steps described in the hypervisor installation section for your particular hypervisor.

2. Now add the hypervisor host to CloudStack. The technique to use varies depending on the hypervisor.
   - *Adding a Host (XenServer or KVM)*
   - *Adding a Host (vSphere)*

Adding a Host (XenServer or KVM)

XenServer and KVM hosts can be added to a cluster at any time.

Requirements for XenServer and KVM Hosts

> **Warning:** Make sure the hypervisor host does not have any VMs already running before you add it to CloudStack.

Configuration requirements:

- Each cluster must contain only hosts with the identical hypervisor.
- For XenServer, do not put more than 8 hosts in a cluster.
- For KVM, do not put more than 16 hosts in a cluster.

For hardware requirements, see the installation section for your hypervisor in the CloudStack Installation Guide.

XenServer Host Additional Requirements

If network bonding is in use, the administrator must cable the new host identically to other hosts in the cluster.

For all additional hosts to be added to the cluster, run the following command. This will cause the host to join the master in a XenServer pool.

```
# xe pool-join master-address=[master IP] master-username=root master-password=[your␣
↪password]
```

> **Note:** When copying and pasting a command, be sure the command has pasted as a single line before executing. Some document viewers may introduce unwanted line breaks in copied text.

With all hosts added to the XenServer pool, run the cloud-setup-bond script. This script will complete the configuration and setup of the bonds on the new hosts in the cluster.

1. Copy the script from the Management Server in /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver/cloud-setup-bonding.sh to the master host and ensure it is executable.
2. Run the script:

   ```
   # ./cloud-setup-bonding.sh
   ```

KVM Host Additional Requirements

- If shared mountpoint storage is in use, the administrator should ensure that the new host has all the same mountpoints (with storage mounted) as the other hosts in the cluster.
- Make sure the new host has the same network configuration (guest, private, and public network) as other hosts in the cluster.
- If you are using OpenVswitch bridges edit the file agent.properties on the KVM host and set the parameter network.bridge.type to openvswitch before adding the host to CloudStack
- If you're using a non-root user to add a KVM host, please add the user to sudoers file:

```
cloudstack ALL=NOPASSWD: /usr/bin/cloudstack-setup-agent
defaults:cloudstack !requiretty
```

Adding a XenServer or KVM Host

1. If you have not already done so, install the hypervisor software on the host. You will need to know which version of the hypervisor software version is supported by CloudStack and what additional configuration is required to ensure the host will work with CloudStack. To find these installation details, see the appropriate section for your hypervisor in the CloudStack Installation Guide.
2. Log in to the CloudStack UI as administrator.
3. In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone in which you want to add the host.
4. Click the Compute tab. In the Clusters node, click View All.
5. Click the cluster where you want to add the host.
6. Click View Hosts.
7. Click Add Host.
8. Provide the following information.
   - Host Name. The DNS name or IP address of the host.
   - Username. Usually root.
   - Password. This is the password for the user from your XenServer or KVM install).
   - Host Tags (Optional). Any labels that you use to categorize hosts for ease of maintenance. For example, you can set to the cloud's HA tag (set in the ha.tag global configuration parameter) if you want this host to be used only for VMs with the "high availability" feature enabled. For more information, see HA-Enabled Virtual Machines as well as HA for Hosts.

   There may be a slight delay while the host is provisioned. It should automatically display in the UI.
9. Repeat for additional hosts.

Adding a Host (vSphere)

For vSphere servers, we recommend creating the cluster of hosts in vCenter and then adding the entire cluster to CloudStack. See Add Cluster: vSphere.

Add Primary Storage

System Requirements for Primary Storage

Hardware requirements:

- Any standards-compliant iSCSI, SMB, or NFS server that is supported by the underlying hypervisor.
- The storage server should be a machine with a large number of disks. The disks should ideally be managed by a hardware RAID controller.
- Minimum required capacity depends on your needs.

When setting up primary storage, follow these restrictions:

- Primary storage cannot be added until a host has been added to the cluster.
- If you do not provision shared primary storage, you must set the global configuration parameter system.vm.local.storage.required to true, or else you will not be able to start VMs.

Adding Primary Storage

When you create a new zone, the first primary storage is added as part of that procedure. You can add primary storage servers at any time, such as when adding a new cluster or adding more servers to an existing cluster.

> **Warning:** When using preallocated storage for primary storage, be sure there is nothing on the storage (ex. you have an empty SAN volume or an empty NFS share). Adding the storage to CloudStack will destroy any existing data.

1. Log in to the CloudStack UI *Log In to the UI*.
2. In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone in which you want to add the primary storage.
3. Click the Compute tab.
4. In the Primary Storage node of the diagram, click View All.
5. Click Add Primary Storage.
6. Provide the following information in the dialog. The information required varies depending on your choice in Protocol.
   - **Scope.** Indicate whether the storage is available to all hosts in the zone or only to hosts in a single cluster.
   - **Pod.** (Visible only if you choose Cluster in the Scope field.) The pod for the storage device.
   - **Cluster.** (Visible only if you choose Cluster in the Scope field.) The cluster for the storage device.
   - **Name.** The name of the storage device.
   - **Protocol.** For XenServer, choose either NFS, iSCSI, or PreSetup. For KVM, choose NFS or SharedMountPoint. For vSphere choose either VMFS (iSCSI or FiberChannel) or NFS. For Hyper-V, choose SMB.
   - **Server (for NFS, iSCSI, or PreSetup).** The IP address or DNS name of the storage device.
   - **Server (for VMFS).** The IP address or DNS name of the vCenter server.
   - **Path (for NFS).** In NFS this is the exported path from the server.

- **Path (for VMFS).** In vSphere this is a combination of the datacenter name and the datastore name. The format is "/" datacenter name "/" datastore name. For example, "/cloud.dc.VM/cluster1datastore".
- **Path (for SharedMountPoint).** With KVM this is the path on each host that is where this primary storage is mounted. For example, "/mnt/primary".
- **SMB Username** (for SMB/CIFS): Applicable only if you select SMB/CIFS provider. The username of the account which has the necessary permissions to the SMB shares. The user must be part of the Hyper-V administrator group.
- **SMB Password** (for SMB/CIFS): Applicable only if you select SMB/CIFS provider. The password associated with the account.
- **SMB Domain**(for SMB/CIFS): Applicable only if you select SMB/CIFS provider. The Active Directory domain that the SMB share is a part of.
- **SR Name-Label (for PreSetup).** Enter the name-label of the SR that has been set up outside CloudStack.
- **Target IQN (for iSCSI).** In iSCSI this is the IQN of the target. For example, iqn.1986-03.com.sun:02:01ec9bb549-1271378984.
- **Lun # (for iSCSI).** In iSCSI this is the LUN number. For example, 3.
- **Tags (optional).** The comma-separated list of tags for this storage device. It should be an equivalent set or superset of the tags on your disk offerings..

The tag sets on primary storage across clusters in a Zone must be identical. For example, if cluster A provides primary storage that has tags T1 and T2, all other clusters in the Zone must also provide primary storage that has tags T1 and T2.

7. Click OK.

Configuring a Storage Plug-in

Note: Primary storage that is based on a custom plug-in (ex. SolidFire) must be added through the CloudStack API (described later in this section). There is no support at this time through the CloudStack UI to add this type of primary storage (although most of its features are available through the CloudStack UI).

Note: The SolidFire storage plug-in for CloudStack is part of the standard CloudStack install. There is no additional work required to add this component.

Adding primary storage that is based on the SolidFire plug-in enables CloudStack to provide hard quality-of-service (QoS) guarantees.

When used with Compute or Disk Offerings, an administrator is able to build an environment in which a root or data disk that a user creates leads to the dynamic creation of a SolidFire volume, which has guaranteed performance. Such a SolidFire volume is associated with one (and only ever one) CloudStack volume, so performance of the CloudStack volume does not vary depending on how heavily other tenants are using the system.

The createStoragePool API has been augmented to support plugable storage providers. The following is a list of parameters to use when adding storage to CloudStack that is based on the SolidFire plug-in:

- command=createStoragePool
- scope=zone
- zoneId=[your zone id]

- name=[name for primary storage]
- hypervisor=Any
- provider=SolidFire
- capacityIops=[whole number of IOPS from the SAN to give to CloudStack]
- capacityBytes=[whole number of bytes from the SAN to give to CloudStack]

The url parameter is somewhat unique in that its value can contain additional key/value pairs.

url=[key/value pairs detailed below (values are URL encoded; for example, '=' is represented as '%3D')]

- MVIP%3D[Management Virtual IP Address] (can be suffixed with :[port number])
- SVIP%3D[Storage Virtual IP Address] (can be suffixed with :[port number])
- clusterAdminUsername%3D[cluster admin's username]
- clusterAdminPassword%3D[cluster admin's password]
- clusterDefaultMinIops%3D[Min IOPS (whole number) to set for a volume; used if Min IOPS is not specified by administrator or user]
- clusterDefaultMaxIops%3D[Max IOPS (whole number) to set for a volume; used if Max IOPS is not specified by administrator or user]
- clusterDefaultBurstIopsPercentOfMaxIops%3D[Burst IOPS is determined by (Min IOPS * clusterDefaultBurstIopsPercentOfMaxIops parameter) (can be a decimal value)]

Add Secondary Storage

System Requirements for Secondary Storage

- NFS storage appliance or Linux NFS server
- SMB/CIFS (Hyper-V)
- (Optional) OpenStack Object Storage (Swift) (see http://swift.openstack.org)
- 100GB minimum capacity
- A secondary storage device must be located in the same zone as the guest VMs it serves.
- Each Secondary Storage server must be available to all hosts in the zone.

Adding Secondary Storage

When you create a new zone, the first secondary storage is added as part of that procedure. You can add secondary storage servers at any time to add more servers to an existing zone.

> **Warning:** Ensure that nothing is stored on the server. Adding the server to CloudStack will destroy any existing data.

1. To prepare for the zone-based Secondary Staging Store, you should have created and mounted an NFS share during Management Server installation. See *Prepare NFS Shares*.

   If you are using an Hyper-V host, ensure that you have created a SMB share.

2. Make sure you prepared the system VM template during Management Server installation. See "Prepare the System VM Template".
3. Log in to the CloudStack UI as root administrator.
4. In the left navigation bar, click Infrastructure.
5. In Secondary Storage, click View All.
6. Click Add Secondary Storage.
7. Fill in the following fields:
   - Name. Give the storage a descriptive name.
   - Provider. Choose S3, Swift, NFS, or CIFS then fill in the related fields which appear. The fields will vary depending on the storage provider; for more information, consult the provider's documentation (such as the S3 or Swift website). NFS can be used for zone-based storage, and the others for region-wide storage. For Hyper-V, select SMB/CIFS.

     Warning: Heterogeneous Secondary Storage is not supported in Regions. You can use only a single NFS, S3, or Swift account per region.

   - Create NFS Secondary Staging Store. This box must always be checked.

     Warning: Even if the UI allows you to uncheck this box, do not do so. This checkbox and the three fields below it must be filled in. Even when Swift or S3 is used as the secondary storage provider, an NFS staging storage in each zone is still required.

   - Zone. The zone where the NFS Secondary Staging Store is to be located.
   - **SMB Username**: Applicable only if you select SMB/CIFS provider. The username of the account which has the necessary permissions to the SMB shares. The user must be part of the Hyper-V administrator group.
   - **SMB Password**: Applicable only if you select SMB/CIFS provider. The password associated with the account.
   - **SMB Domain**: Applicable only if you select SMB/CIFS provider. The Active Directory domain that the SMB share is a part of.
   - NFS server. The name of the zone's Secondary Staging Store.
   - Path. The path to the zone's Secondary Staging Store.

Adding an NFS Secondary Staging Store for Each Zone

Every zone must have at least one NFS store provisioned; multiple NFS servers are allowed per zone. To provision an NFS Staging Store for a zone:

1. Log in to the CloudStack UI as root administrator.
2. In the left navigation bar, click Infrastructure.
3. In Secondary Storage, click View All.
4. In Select View, choose Secondary Staging Store.
5. Click the Add NFS Secondary Staging Store button.

6. Fill out the dialog box fields, then click OK:
   - Zone. The zone where the NFS Secondary Staging Store is to be located.
   - NFS server. The name of the zone's Secondary Staging Store.
   - Path. The path to the zone's Secondary Staging Store.

Initialize and Test

After everything is configured, CloudStack will perform its initialization. This can take 30 minutes or more, depending on the speed of your network. When the initialization has completed successfully, the administrator's Dashboard should be displayed in the CloudStack UI.

1. Verify that the system is ready. In the left navigation bar, select Templates. Click on the CentOS 5.5 (64bit) no Gui (KVM) template. Check to be sure that the status is "Download Complete." Do not proceed to the next step until this status is displayed.
2. Go to the Instances tab, and filter by My Instances.
3. Click Add Instance and follow the steps in the wizard.
   (a) Choose the zone you just added.
   (b) In the template selection, choose the template to use in the VM. If this is a fresh installation, likely only the provided CentOS template is available.
   (c) Select a service offering. Be sure that the hardware you have allows starting the selected service offering.
   (d) In data disk offering, if desired, add another data disk. This is a second volume that will be available to but not mounted in the guest. For example, in Linux on XenServer you will see /dev/xvdb in the guest after rebooting the VM. A reboot is not required if you have a PV-enabled OS kernel in use.
   (e) In default network, choose the primary network for the guest. In a trial installation, you would have only one option here.
   (f) Optionally give your VM a name and a group. Use any descriptive text you would like.
   (g) Click Launch VM. Your VM will be created and started. It might take some time to download the template and complete the VM startup. You can watch the VMâ€™s progress in the Instances screen.
4. To use the VM, click the View Console button.

   For more information about using VMs, including instructions for how to allow incoming network traffic to the VM, start, stop, and delete VMs, and move a VM from one host to another, see Working With Virtual Machines in the Administratorâ€™s Guide.

Congratulations! You have successfully completed a CloudStack Installation.

If you decide to grow your deployment, you can add more hosts, primary storage, zones, pods, and clusters.

Configuration Parameters

About Configuration Parameters

CloudStack provides a variety of settings you can use to set limits, configure features, and enable or disable features in the cloud. Once your Management Server is running, you might need to set some of these configuration parameters, depending on what optional features you are setting up. You can set default values at the global level, which will be in effect throughout the cloud unless you override them at a lower level. You can make local settings, which will override the global configuration parameter values, at the level of an account, zone, cluster, or primary storage.

The documentation for each CloudStack feature should direct you to the names of the applicable parameters. The following table shows a few of the more useful parameters.

Field	Value
management.network.cidr	A CIDR that describes the network that the management CIDRs reside on. This variable must be set for deployments that use vSphere. It is recommended to be set for other deployments as well. Example: 192.168.3.0/24.
xen.setup.multipath	For XenServer nodes, this is a true/false variable that instructs CloudStack to enable iSCSI multipath on the XenServer Hosts when they are added. This defaults to false. Set it to true if you would like CloudStack to enable multipath.If this is true for a NFS-based deployment multipath will still be enabled on the XenServer host. However, this does not impact NFS operation and is harmless.
secstorage.allowed.internal.sites	This is used to protect your internal network from rogue attempts to download arbitrary files using the template download feature. This is a comma-separated list of CIDRs. If a requested URL matches any of these CIDRs the Secondary Storage VM will use the private network interface to fetch the URL. Other URLs will go through the public interface. We suggest you set this to 1 or 2 hardened internal machines where you keep your templates. For example, set it to 192.168.1.66/32.
use.local.storage	Determines whether CloudStack will use storage that is local to the Host for data disks, templates, and snapshots. By default CloudStack will not use this storage. You should change this to true if you want to use local storage and you understand the reliability and feature drawbacks to choosing local storage.
host	This is the IP address of the Management Server. If you are using multiple Management Servers you should enter a load balanced IP address that is reachable via the private network.
default.page.size	Maximum number of items per page that can be returned by a CloudStack API command. The limit applies at the cloud level and can vary from cloud to cloud. You can override this with a lower value on a particular API call by using the page and pagesize API command parameters. For more information, see the Developer's Guide. Default: 500.
ha.tag	The label you want to use throughout the cloud to designate certain hosts as dedicated HA hosts. These hosts will be used only for HA-enabled VMs that are restarting due to the failure of another host. For example, you could set this to ha_host. Specify the ha.tag value asa host tag when you add a new host to the cloud.
vmware.vcenter.session.timeout	Determines the vCenter session timeout value by using this parameter. The default value is 20 minutes. Increase the timeout value to avoid timeout errors in VMware deployments because certain VMware operations take more than 20 minutes.

Setting Global Configuration Parameters

Use the following steps to set global configuration parameters. These values will be the defaults in effect throughout your CloudStack deployment.

1. Log in to the UI as administrator.
2. In the left navigation bar, click Global Settings.
3. In Select View, choose one of the following:
   - Global Settings. This displays a list of the parameters with brief descriptions and current values.
   - Hypervisor Capabilities. This displays a list of hypervisor versions with the maximum number of guests supported for each.
4. Use the search box to narrow down the list to those you are interested in.
5. In the Actions column, click the Edit icon to modify a value. If you are viewing Hypervisor Capabilities, you must click the name of the hypervisor first to display the editing screen.

Setting Local Configuration Parameters

Use the following steps to set local configuration parameters for an account, zone, cluster, or primary storage. These values will override the global configuration settings.

1. Log in to the UI as administrator.
2. In the left navigation bar, click Infrastructure or Accounts, depending on where you want to set a value.
3. Find the name of the particular resource that you want to work with. For example, if you are in Infrastructure, click View All on the Zones, Clusters, or Primary Storage area.
4. Click the name of the resource where you want to set a limit.
5. Click the Settings tab.
6. Use the search box to narrow down the list to those you are interested in.
7. In the Actions column, click the Edit icon to modify a value.

Granular Global Configuration Parameters

The following global configuration parameters have been made more granular. The parameters are listed under three different scopes: account, cluster, and zone.

Field	Field	Value
account	remote.access.vpn.client.iprange	The range of IPs to be allocated to remotely access the VPN clients. The first IP in the range is used by the VPN server.
account	allow.public.user.templates	If false, users will not be able to create public templates.
account	use.system.public.ips	If true and if an account has one or more dedicated public IP ranges, IPs are acquired from the system pool after all the IPs dedicated to the account have been consumed.
account	use.system.guest.vlans	If true and if an account has one or more dedicated guest VLAN ranges, VLANs are allocated from the system pool after all the VLANs dedicated to the account have been consumed.
cluster	cluster.storage.allocated.capacity.notification.threshold	The percentage, as a value between 0 and 1, of allocated storage utilization above which alerts are emitted that the storage is below the threshold.
cluster	cluster.storage.capacity.notification.threshold	The percentage, as a value between 0 and 1, of storage utilization above which alerts are emitted that the available storage is below the threshold.
cluster	cluster.cpu.allocated.capacity.notification.threshold	The percentage, as a value between 0 and 1, of cpu utilization above which alerts are emitted that the available CPU is below the threshold.
cluster	cluster.memory.allocated.capacity.notification.threshold	The percentage, as a value between 0 and 1, of memory utilization above which alerts are emitted that the available memory is below the threshold.
cluster	cluster.cpu.allocated.capacity.disable.threshold	The percentage, as a value between 0 and 1, of CPU utilization above which allocators will disable that cluster from further usage. Keep the corresponding notification threshold lower than this value to be notified beforehand.
cluster	cluster.memory.allocated.capacity.disable.threshold	The percentage, as a value between 0 and 1, of memory utilization above which allocators will disable that cluster from further usage. Keep the corresponding notification threshold lower than this value to be notified beforehand.
cluster	cpu.overprovisioning.factor	Used for CPU over-provisioning calculation; the available CPU will be the mathematical product of actualCpuCapacity and cpu.overprovisioning.factor.
cluster	mem.overprovisioning.factor	Used for memory over-provisioning calculation.
cluster	vmware.reserve.cpu	Specify whether or not to reserve CPU when not over-provisioning; In case of CPU over-provisioning, CPU is always reserved.
cluster	vmware.reserve.mem	Specify whether or not to reserve memory when not over-provisioning; In case of memory over-provisioning memory is always reserved.
zone	pool.storage.allocated.capacity.disablethreshold	The percentage, as a value between 0 and 1, of allocated storage utilization above which allocators will disable that pool because the available allocated storage is below the threshold.
zone	pool.storage.capacity.disablethreshold	The percentage, as a value between 0 and 1, of storage utilization above which allocators will disable the pool because the available storage capacity is below the threshold.
zone	storage.overprovisioning.factor	Used for storage over-provisioning calculation; available storage will be the mathematical product of actualStorageSize and storage.overprovisioning.factor.
zone	network.throttling.rate	Default data transfer rate in megabits per second allowed in a network.
zone	guest.domain.suffix	Default domain name for VMs inside a virtual networks with a router.
zone	router.template.xen	Name of the default router template on Xenserver.
zone	router.template.kvm	Name of the default router template on KVM.
zone	router.template.vmware	Name of the default router template on VMware.
zone	enable.dynamic.scale.vm	Enable or diable dynamically scaling of a VM.
zone	use.external.dns	Bypass internal DNS, and use the external DNS1 and DNS2
zone	blacklisted.routes	Routes that are blacklisted cannot be used for creating static routes for a VPC Private Gateway.

3.4 Hypervisor Setup

3.4.1 Host Hyper-V Installation

If you want to use Hyper-V hypervisor to run guest virtual machines, install Hyper-V on the hosts in your cloud. The instructions in this section doesn't duplicate Hyper-V Installation documentation. It provides the CloudStack-specific steps that are needed to prepare a Hyper-V host to work with CloudStack.

System Requirements for Hyper-V Hypervisor Hosts

Supported Operating Systems for Hyper-V Hosts

- Windows Server 2012 R2 Standard
- Windows Server 2012 R2 Datacenter
- Hyper-V 2012 R2

Minimum System Requirements for Hyper-V Hosts

- 1.4 GHz 64-bit processor with hardware-assisted virtualization.
- 800 MB of RAM
- 32 GB of disk space
- Gigabit (10/100/1000baseT) Ethernet adapter

Supported Storage

- Primary Storage: Server Message Block (SMB) Version 3, Local
- Secondary Storage: SMB

Preparation Checklist for Hyper-V

For a smoother installation, gather the following information before you start:

Hyper-V Requiremen ts	Value	Description
Server Roles	Hyper-V	After the Windows Server 2012 R2 installation, ensure that Hyper-V is selected from Server Roles. For more information, see Installing Hyper-V.
Share Location	New folders in the /Share director y	Ensure that folders are created for Primary and Secondary storage. The SMB share and the hosts should be part of the same domain. If you are using Windows SMB share, the location of the file share for the Hyper-V deployment will be the new folder created in the \Shares on the selected volume. You can create sub-folders for both PRODUCT Primary and Secondary storage within the share location. When you select the profile for the file shares, ensure that you select SMB Share -Applications. This creates the file shares with settings appropriate for Hyper-V.
Domain and Hosts		Hosts should be part of the same Active Directory domain.
Hyper-V Users	Full control	Full control on the SMB file share.
Virtual Switch		If you are using Hyper-V 2012 R2, manually create an external virtual switch before adding the host to PRODUCT. If the Hyper-V host is added to the Hyper-V manager, select the host, then click Virtual Switch Manager, then New Virtual Switch. In the External Network, select the desired NIC adapter and click Apply. If you are using Windows 2012 R2, virtual switch is created automatically.
Virtual Switch Name		Take a note of the name of the virtual switch. You need to specify that when configuring PRODUCT physical network labels.
Hyper-V Domain Users		• Add the Hyper-V domain users to the Hyper-V Administrators group. • A domain user should have full control on the SMB share that is exported for primary and secondary storage. • This domain user should be part of the Hyper-V Administrators and Local Administrators group on the Hyper-V hosts that are to be managed by PRODUCT. • The Hyper-V Agent service runs with the credentials of this domain user account.

Hyper-V Installation Steps

1. Download the operating system from Windows Server 2012 R2.
2. Install it on the host as given in Install and Deploy Windows Server 2012 R2.
3. Post installation, ensure that you enable Hyper-V role in the server.
4. If no Active Directory domain exists in your deployment, create one and add users to the domain.
5. In the Active Directory domain, ensure that all the Hyper-v hosts are added so that all the hosts are part of the domain.
6. Add the domain user to the following groups on the Hyper-V host: Hyper-V Administrators and Local Administrators.

Installing the CloudStack Agent on a Hyper-V Host

The Hyper-V Agent helps CloudStack perform operations on the Hyper-V hosts. This Agent communicates with the Management Server and controls all the instances on the host. Each Hyper-V host must have the Hyper-V Agent installed on it for successful interaction between the host and CloudStack. The Hyper-V Agent runs as a Windows service. Install the Agent on each host using the following steps.

CloudStack Management Server communicates with Hyper-V Agent by using HTTPS. For secure communication between the Management Server and the host, install a self-signed certificate on port 8250.

Note: The Agent installer automatically perform this operation. You have not selected this option during the Agent installation, it can also be done manually as given in step 1.

1. Create and add a self-signed SSL certificate on port 8250:
   (a) Create A self-signed SSL certificate:

   ```
   # New-SelfSignedCertificate -DnsName apachecloudstack -
   ↪CertStoreLocation Cert:LocalMachineMy
   ```

   This command creates the self-signed certificate and add that to the certificate store `LocalMachine\My`.

   (b) Add the created certificate to port 8250 for https communication:

   ```
   netsh http add sslcert ipport=0.0.0.0:8250 certhash=<thumbprint> appid="
   ↪{727beb1c-6e7c-49b2-8fbd-f03dbe481b08}"
   ```

   Thumbprint is the thumbprint of the certificate you created.
2. Build the CloudStack Agent for Hyper-V as given in Building CloudStack Hyper-V Agent.
3. As an administrator, run the installer.
4. Provide the Hyper-V admin credentials when prompted.

   When the agent installation is finished, the agent runs as a service on the host machine.

Physical Network Configuration for Hyper-V

You should have a plan for how the hosts will be cabled and which physical NICs will carry what types of traffic. By default, CloudStack will use the device that is used for the default route.

If you are using Hyper-V 2012 R2, manually create an external virtual switch before adding the host to CloudStack. If the Hyper-V host is added to the Hyper-V manager, select the host, then click Virtual Switch Manager, then New Virtual Switch. In the External Network, select the desired NIC adapter and click Apply.

If you are using Windows 2012 R2, virtual switch is created automatically.

Storage Preparation for Hyper-V (Optional)

CloudStack allows administrators to set up shared Primary Storage and Secondary Storage that uses SMB.

1. Create a SMB storage and expose it over SMB Version 3.

   For more information, see Deploying Hyper-V over SMB.

   You can also create and export SMB share using Windows. After the Windows Server 2012 R2 installation, select File and Storage Services from Server Roles to create an SMB file share. For more information, see Creating an SMB File Share Using Server Manager.

2. Add the SMB share to the Active Directory domain.

   The SMB share and the hosts managed by CloudStack need to be in the same domain. However, the storage should be accessible from the Management Server with the domain user privileges.

3. While adding storage to CloudStack, ensure that the correct domain, and credentials are supplied. This user should be able to access the storage from the Management Server.

3.4.2 Host KVM Installation

System Requirements for KVM Hypervisor Hosts

KVM is included with a variety of Linux-based operating systems. Although you are not required to run these distributions, the following are recommended:

- CentOS / RHEL: 7.X
- Ubuntu: 14.04

The main requirement for KVM hypervisors is the libvirt and Qemu version. No matter what Linux distribution you are using, make sure the following requirements are met:

- libvirt: 1.2.0 or higher
- Qemu/KVM: 2.0 or higher

The default bridge in CloudStack is the Linux native bridge implementation (bridge module). CloudStack includes an option to work with OpenVswitch, the requirements are listed below

- libvirt: 1.2.0 or higher
- openvswitch: 1.7.1 or higher

In addition, the following hardware requirements apply:

- Within a single cluster, the hosts must be of the same distribution version.
- All hosts within a cluster must be homogenous. The CPUs must be of the same type, count, and feature flags.
- Must support HVM (Intel-VT or AMD-V enabled)
- 64-bit x86 CPU (more cores results in better performance)
- 4 GB of memory

- At least 1 NIC
- When you deploy CloudStack, the hypervisor host must not have any VMs already running. These will be destroy by CloudStack.

KVM Installation Overview

If you want to use the Linux Kernel Virtual Machine (KVM) hypervisor to run guest virtual machines, install KVM on the host(s) in your cloud. The material in this section doesn't duplicate KVM installation docs. It provides the CloudStack-specific steps that are needed to prepare a KVM host to work with CloudStack.

Warning: Before continuing, make sure that you have applied the latest updates to your host.

Warning: It is NOT recommended to run services on this host not controlled by CloudStack.

The procedure for installing a KVM Hypervisor Host is:

1. Prepare the Operating System
2. Install and configure libvirt
3. Configure Security Policies (AppArmor and SELinux)
4. Install and configure the Agent

Prepare the Operating System

The OS of the Host must be prepared to host the CloudStack Agent and run KVM instances.

1. Log in to your OS as root.
2. Check for a fully qualified hostname.

   ```
   $ hostname --fqdn
   ```

   This should return a fully qualified hostname such as "kvm1.lab.example.org". If it does not, edit /etc/hosts so that it does.

3. Make sure that the machine can reach the Internet.

   ```
   $ ping www.cloudstack.org
   ```

4. Turn on NTP for time synchronization.

   Note: NTP is required to synchronize the clocks of the servers in your cloud. Unsynchronized clocks can cause unexpected problems.

   (a) Install NTP

   ```
   $ yum install ntp
   ```

   ```
   $ apt-get install openntpd
   ```

5. Repeat all of these steps on every hypervisor host.

Install and configure the Agent

To manage KVM instances on the host CloudStack uses a Agent. This Agent communicates with the Management server and controls all the instances on the host.

First we start by installing the agent:

In RHEL or CentOS:

```
$ yum install cloudstack-agent
```

In Ubuntu:

```
$ apt-get install cloudstack-agent
```

The host is now ready to be added to a cluster. This is covered in a later section, see *Adding a Host*. It is recommended that you continue to read the documentation before adding the host!

If you're using a non-root user to add the KVM host, please add the user to sudoers file:

```
cloudstack ALL=NOPASSWD: /usr/bin/cloudstack-setup-agent
defaults:cloudstack !requiretty
```

Configure CPU model for KVM guest (Optional)

In additional,the CloudStack Agent allows host administrator to control the guest CPU model which is exposed to KVM instances. By default, the CPU model of KVM instance is likely QEMU Virtual CPU version x.x.x with least CPU features exposed. There are a couple of reasons to specify the CPU model:

- To maximise performance of instances by exposing new host CPU features to the KVM instances;
- To ensure a consistent default CPU across all machines,removing reliance of variable QEMU defaults;

For the most part it will be sufficient for the host administrator to specify the guest CPU config in the per-host configuration file (/etc/cloudstack/agent/agent.properties). This will be achieved by introducing following configuration parameters:

```
guest.cpu.mode=custom|host-model|host-passthrough
guest.cpu.model=from /usr/share/libvirt/cpu_map.xml(only valid when guest.cpu.
↪mode=custom)
guest.cpu.features=vmx ept aes smx mmx ht (space separated list of cpu flags to apply)
```

There are three choices to fulfill the cpu model changes:

1. **custom:** you can explicitly specify one of the supported named model in /usr/share/libvirt/cpu_map.xml
2. **host-model:** libvirt will identify the CPU model in /usr/share/libvirt/cpu_map.xml which most closely matches the host, and then request additional CPU flags to complete the match. This should give close to maximum functionality/performance, which maintaining good reliability/compatibility if the guest is migrated to another host with slightly different host CPUs.
3. **host-passthrough:** libvirt will tell KVM to passthrough the host CPU with no modifications. The difference to host-model, instead of just matching feature flags, every last detail of the host CPU is matched. This gives absolutely best performance, and can be important to some apps which check low level CPU details, but it comes at a cost with respect to migration: the guest can only be migrated to an exactly matching host CPU.

Here are some examples:

- custom

```
guest.cpu.mode=custom
guest.cpu.model=SandyBridge
```

- host-model

```
guest.cpu.mode=host-model
```

- host-passthrough

```
guest.cpu.mode=host-passthrough
guest.cpu.features=vmx
```

Note: host-passthrough may lead to migration failure,if you have this problem, you should use host-model or custom. guest.cpu.features will force cpu features as a required policy so make sure to put only those features that are provided by the host CPU.

Install and Configure libvirt

CloudStack uses libvirt for managing virtual machines. Therefore it is vital that libvirt is configured correctly. Libvirt is a dependency of cloudstack-agent and should already be installed.

1. In order to have live migration working libvirt has to listen for unsecured TCP connections. We also need to turn off libvirts attempt to use Multicast DNS advertising. Both of these settings are in `/etc/libvirt/libvirtd.conf`

   Set the following parameters:

```
listen_tls = 0
```

```
listen_tcp = 1
```

```
tcp_port = "16509"
```

```
auth_tcp = "none"
```

```
mdns_adv = 0
```

2. Turning on "listen_tcp" in libvirtd.conf is not enough, we have to change the parameters as well:

   On RHEL or CentOS modify `/etc/sysconfig/libvirtd`:

   Uncomment the following line:

```
#LIBVIRTD_ARGS="--listen"
```

   On Ubuntu 14.04: modify `/etc/default/libvirt-bin`

   Add "-l" to the following line

```
libvirtd_opts="-d"
```

so it looks like:

```
libvirtd_opts="-d -l"
```

And modify `/etc/init/libvirt-bin.conf`

Add "-l" to the following line

```
env libvirtd_opts="-d"
```

so it looks like:

```
env libvirtd_opts="-d -l"
```

On Ubuntu 16.04: just modify `/etc/init/libvirt-bin.conf`

Add "-l" to the following line

```
env libvirtd_opts="-d"
```

so it looks like:

```
env libvirtd_opts="-d -l"
```

3. Restart libvirt

   In RHEL or CentOS:

   ```
   $ service libvirtd restart
   ```

   In Ubuntu:

   ```
   $ service libvirt-bin restart
   ```

Configure the Security Policies

CloudStack does various things which can be blocked by security mechanisms like AppArmor and SELinux. These have to be disabled to ensure the Agent has all the required permissions.

1. Configure SELinux (RHEL and CentOS)

   (a) Check to see whether SELinux is installed on your machine. If not, you can skip this section.

   In RHEL or CentOS, SELinux is installed and enabled by default. You can verify this with:

   ```
   $ rpm -qa | grep selinux
   ```

   (b) Set the SELINUX variable in `/etc/selinux/config` to "permissive". This ensures that the permissive setting will be maintained after a system reboot.

   In RHEL or CentOS:

   ```
   $ vi /etc/selinux/config
   ```

   Change the following line

   ```
   SELINUX=enforcing
   ```

   to this

```
SELINUX=permissive
```

(c) Then set SELinux to permissive starting immediately, without requiring a system reboot.

```
$ setenforce permissive
```

2. Configure Apparmor (Ubuntu)

(a) Check to see whether AppArmor is installed on your machine. If not, you can skip this section.

In Ubuntu AppArmor is installed and enabled by default. You can verify this with:

```
$ dpkg --list 'apparmor'
```

(b) Disable the AppArmor profiles for libvirt

```
$ ln -s /etc/apparmor.d/usr.sbin.libvirtd /etc/apparmor.d/disable/
```

```
$ ln -s /etc/apparmor.d/usr.lib.libvirt.virt-aa-helper /etc/apparmor.d/
↪disable/
```

```
$ apparmor_parser -R /etc/apparmor.d/usr.sbin.libvirtd
```

```
$ apparmor_parser -R /etc/apparmor.d/usr.lib.libvirt.virt-aa-helper
```

Configuring the Networking

Warning: This is a very important section, please make sure you read this thoroughly.

Note: This section details how to configure bridges using the native implementation in Linux. Please refer to the next section if you intend to use OpenVswitch

CloudStack uses the network bridges in conjunction with KVM to connect the guest instances to each other and the outside world. They also are used to connect the System VMs to your infrastructure.

By default these bridges are called *cloudbr0* and *cloudbr1* etc, but this can be changed to be more description.

Warning: It is essential that you keep the configuration consistent across all of your hypervisors.

There are many ways to configure your networking. Even within the scope of a given network mode. Below are a few simple examples.

Network example for Basic Networks

In the Basic networking, all of the guests in a given pod will be on the same VLAN/subnet. It is common to use the native (untagged) VLAN for the private/management network, so in this example we will have two VLANs, one (native) for your private/management network and one for the guest network.

We assume that the hypervisor has one NIC (eth0) with one tagged VLAN trunked from the switch:

1. Native VLAN for management network (cloudbr0)
2. VLAN 200 for guest network of the instances (cloudbr1)

In this the following example we give the Hypervisor the IP-Address 192.168.42.11/24 with the gateway 192.168.42.1

Note: The Hypervisor and Management server don't have to be in the same subnet

Configuring the Network Bridges for Basic Networks

It depends on the distribution you are using how to configure these, below you'll find examples for RHEL/CentOS and Ubuntu.

Note: The goal is to have two bridges called 'cloudbr0' and 'cloudbr1' after this section. This should be used as a guideline only. The exact configuration will depend on your network layout.

Configure RHEL or CentOS for Basic Networks

The required packages were installed when libvirt was installed, we can proceed to configuring the network.

First we configure eth0

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Make sure it looks similar to:

```
DEVICE=eth0
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
BRIDGE=cloudbr0
```

We now have to configure the VLAN interfaces:

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0.200
```

```
DEVICE=eth0.200
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
VLAN=yes
BRIDGE=cloudbr1
```

Now that we have the VLAN interfaces configured we can add the bridges on top of them.

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr0
```

Now we configure cloudbr0 and include the Management IP of the hypervisor.

Note: The management IP of the hypervisor doesn't have to be in same subnet/VLAN as the management network, but its quite common.

```
DEVICE=cloudbr0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
IPADDR=192.168.42.11
GATEWAY=192.168.42.1
NETMASK=255.255.255.0
STP=yes
```

We configure cloudbr1 as a plain bridge without an IP address

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr1
```

```
DEVICE=cloudbr1
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configure Ubuntu for Basic Networks

All the required packages were installed when you installed libvirt, so we only have to configure the network.

```
$ vi /etc/network/interfaces
```

Modify the interfaces file to look like this:

```
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual

auto eth0.200
```

(continues on next page)

(continued from previous page)

```
iface eth0 inet manual

# management network
auto cloudbr0
iface cloudbr0 inet static
    bridge_ports eth0
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
    address 192.168.42.11
    netmask 255.255.255.240
    gateway 192.168.42.1
    dns-nameservers 8.8.8.8 8.8.4.4
    dns-domain lab.example.org

# guest network
auto cloudbr1
iface cloudbr1 inet manual
    bridge_ports eth0.200
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Network Example for Advanced Networks

In the Advanced networking mode is most common to have (at least) two physical interfaces. In this example we will again have the hypervisor management interface on cloudbr0 on the untagged (native) VLAN. But now we will have a bridge on top of our additional interface (eth1) for public and guest traffic with no VLANs applied by us - CloudStack will add the VLANs as required.

We again give the Hypervisor the IP-Address 192.168.42.11/24 with the gateway 192.168.42.1

Note: The Hypervisor and Management server don't have to be in the same subnet

Configuring the Network Bridges for Advanced Networks

It depends on the distribution you are using how to configure these, below you'll find examples for RHEL/CentOS and Ubuntu.

Note: The goal is to have two bridges called 'cloudbr0' and 'cloudbr1' after this section. This should be used as a guideline only. The exact configuration will depend on your network layout.

Configure RHEL/CentOS for Advanced Networks

The required packages were installed when libvirt was installed, we can proceed to configuring the network.

First we configure eth0

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Make sure it looks similar to:

```
DEVICE=eth0
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
BRIDGE=cloudbr0
```

We now have to configure the VLAN interfaces:

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
DEVICE=eth1
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
BRIDGE=cloudbr1
```

Now we have the VLAN interfaces configured we can add the bridges on top of them.

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr0
```

Now we configure cloudbr0 and include the Management IP of the hypervisor.

Note: The management IP of the hypervisor doesn't have to be in same subnet/VLAN as the management network, but its quite common.

```
DEVICE=cloudbr0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
IPADDR=192.168.42.11
GATEWAY=192.168.42.1
NETMASK=255.255.255.0
STP=yes
```

We configure cloudbr1 as a plain bridge without an IP address

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr1
```

```
DEVICE=cloudbr1
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configure Ubuntu for Advanced Networks

All the required packages were installed when you installed libvirt, so we only have to configure the network.

```
$ vi /etc/network/interfaces
```

Modify the interfaces file to look like this:

```
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual

# The second network interface
auto eth1
iface eth1 inet manual

# management network
auto cloudbr0
iface cloudbr0 inet static
    bridge_ports eth0
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
    address 192.168.42.11
    netmask 255.255.255.240
    gateway 192.168.42.1
    dns-nameservers 8.8.8.8 8.8.4.4
    dns-domain lab.example.org

# guest network
auto cloudbr1
iface cloudbr1 inet manual
    bridge_ports eth1
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configure the network using OpenVswitch

Warning: This is a very important section, please make sure you read this thoroughly.

In order to forward traffic to your instances you will need at least two bridges: *public* and *private*.

By default these bridges are called *cloudbr0* and *cloudbr1*, but you do have to make sure they are available on each hypervisor.

The most important factor is that you keep the configuration consistent on all your hypervisors.

Preparing

To make sure that the native bridge module will not interfere with openvswitch the bridge module should be added to the blacklist. See the modprobe documentation for your distribution on where to find the blacklist. Make sure the module is not loaded either by rebooting or executing rmmod bridge before executing next steps.

The network configurations below depend on the ifup-ovs and ifdown-ovs scripts which are part of the openvswitch installation. They should be installed in /etc/sysconfig/network-scripts/

OpenVswitch Network example

There are many ways to configure your network. In the Basic networking mode you should have two VLANs, one for your private network and one for the public network.

We assume that the hypervisor has one NIC (eth0) with three tagged VLANs:

1. VLAN 100 for management of the hypervisor
2. VLAN 200 for public network of the instances (cloudbr0)
3. VLAN 300 for private network of the instances (cloudbr1)

On VLAN 100 we give the Hypervisor the IP-Address 192.168.42.11/24 with the gateway 192.168.42.1

Note: The Hypervisor and Management server don't have to be in the same subnet

Configuring the network bridges for OpenVswitch

It depends on the distribution you are using how to configure these, below you'll find examples for RHEL/CentOS.

Note: The goal is to have three bridges called 'mgmt0', 'cloudbr0' and 'cloudbr1' after this section. This should be used as a guideline only. The exact configuration will depend on your network layout.

Configure OpenVswitch

The network interfaces using OpenVswitch are created using the ovs-vsctl command. This command will configure the interfaces and persist them to the OpenVswitch database.

First we create a main bridge connected to the eth0 interface. Next we create three fake bridges, each connected to a specific vlan tag.

```
# ovs-vsctl add-br cloudbr
# ovs-vsctl add-port cloudbr eth0
# ovs-vsctl set port cloudbr trunks=100,200,300
# ovs-vsctl add-br mgmt0 cloudbr 100
# ovs-vsctl add-br cloudbr0 cloudbr 200
# ovs-vsctl add-br cloudbr1 cloudbr 300
```

Configure OpenVswitch in RHEL or CentOS

The required packages were installed when openvswitch and libvirt were installed, we can proceed to configuring the network.

First we configure eth0

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Make sure it looks similar to:

```
DEVICE=eth0
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
```

We have to configure the base bridge with the trunk.

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr
```

```
DEVICE=cloudbr
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
DEVICETYPE=ovs
TYPE=OVSBridge
```

We now have to configure the three VLAN bridges:

```
$ vi /etc/sysconfig/network-scripts/ifcfg-mgmt0
```

```
DEVICE=mgmt0
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=static
DEVICETYPE=ovs
TYPE=OVSBridge
IPADDR=192.168.42.11
GATEWAY=192.168.42.1
NETMASK=255.255.255.0
```

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr0
```

```
DEVICE=cloudbr0
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
DEVICETYPE=ovs
TYPE=OVSBridge
```

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr1
```

```
DEVICE=cloudbr1
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=OVSBridge
DEVICETYPE=ovs
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configuring the firewall

The hypervisor needs to be able to communicate with other hypervisors and the management server needs to be able to reach the hypervisor.

In order to do so we have to open the following TCP ports (if you are using a firewall):

1. 22 (SSH)
2. 1798
3. 16509, 16514 (libvirt)
4. 5900 - 6100 (VNC consoles)
5. 49152 - 49216 (libvirt live migration)

It depends on the firewall you are using how to open these ports. Below you'll find examples how to open these ports in RHEL/CentOS and Ubuntu.

Open ports in RHEL/CentOS

RHEL and CentOS use iptables for firewalling the system, you can open extra ports by executing the following iptable commands:

```
$ iptables -I INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 1798 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 16509 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 16514 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 5900:6100 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 49152:49216 -j ACCEPT
```

These iptable settings are not persistent accross reboots, we have to save them first.

```
$ iptables-save > /etc/sysconfig/iptables
```

Open ports in Ubuntu

The default firewall under Ubuntu is UFW (Uncomplicated FireWall), which is a Python wrapper around iptables.

To open the required ports, execute the following commands:

```
$ ufw allow proto tcp from any to any port 22
```

```
$ ufw allow proto tcp from any to any port 1798
```

```
$ ufw allow proto tcp from any to any port 16509
```

```
$ ufw allow proto tcp from any to any port 16514
```

```
$ ufw allow proto tcp from any to any port 5900:6100
```

```
$ ufw allow proto tcp from any to any port 49152:49216
```

Note: By default UFW is not enabled on Ubuntu. Executing these commands with the firewall disabled does not enable the firewall.

Additional Packages Required for Features

Secondary Storage Bypass

New in 4.11 is the ability to bypass storing a template on secondary storage, and instead directly downloading a 'template' from an alternate remote location. In order to facilitate this the **Aria2** (https://aria2.github.io/) package must be installed on all of your KVM hosts.

As this package often is not available in standard distribution repos, you will need to install the package from your preferred source.

Live Migration

CloudStack uses the qemu-img to perform live migrations. In CentOS > 6.3, the qemu-img supplied by RedHat/CentOS ceased to include a '-s' switch which performs snapshots. The '-s' switch has been restored in latest CentOS/RHEL 7.x versions.

In order to be able to perform live migrations on CentOS 6.x (greater than 6.3) you must replace your version of qemu-img with one which has been patched to include the '-s' switch.

Add the host to CloudStack

The host is now ready to be added to a cluster. This is covered in a later section, see *Adding a Host*. It is recommended that you continue to read the documentation before adding the host!

3.4.3 Host LXC Installation

System Requirements for LXC Hosts

LXC requires the Linux kernel cgroups functionality which is available starting 2.6.24. Although you are not required to run these distributions, the following are recommended:

- CentOS / RHEL: 6.3
- Ubuntu: 12.04(.1)

The main requirement for LXC hypervisors is the libvirt and Qemu version. No matter what Linux distribution you are using, make sure the following requirements are met:

- libvirt: 1.0.0 or higher
- Qemu/KVM: 1.0 or higher

The default bridge in CloudStack is the Linux native bridge implementation (bridge module). CloudStack includes an option to work with OpenVswitch, the requirements are listed below

- libvirt: 1.0.0 or higher
- openvswitch: 1.7.1 or higher

In addition, the following hardware requirements apply:

- Within a single cluster, the hosts must be of the same distribution version.
- All hosts within a cluster must be homogenous. The CPUs must be of the same type, count, and feature flags.
- Must support HVM (Intel-VT or AMD-V enabled)
- 64-bit x86 CPU (more cores results in better performance)
- 4 GB of memory
- At least 1 NIC
- When you deploy CloudStack, the hypervisor host must not have any VMs already running

LXC Installation Overview

LXC does not have any native system VMs, instead KVM will be used to run system VMs. This means that your host will need to support both LXC and KVM, thus most of the installation and configuration will be identical to the KVM installation. The material in this section doesn't duplicate KVM installation docs. It provides the CloudStack-specific steps that are needed to prepare a KVM host to work with CloudStack.

Warning: Before continuing, make sure that you have applied the latest updates to your host.

Warning: It is NOT recommended to run services on this host not controlled by CloudStack.

The procedure for installing an LXC Host is:

1. Prepare the Operating System
2. Install and configure libvirt
3. Configure Security Policies (AppArmor and SELinux)
4. Install and configure the Agent

Prepare the Operating System

The OS of the Host must be prepared to host the CloudStack Agent and run KVM instances.

1. Log in to your OS as root.
2. Check for a fully qualified hostname.

   ```
   $ hostname --fqdn
   ```

   This should return a fully qualified hostname such as "kvm1.lab.example.org". If it does not, edit /etc/hosts so that it does.
3. Make sure that the machine can reach the Internet.

   ```
   $ ping www.cloudstack.org
   ```
4. Turn on NTP for time synchronization.

   Note: NTP is required to synchronize the clocks of the servers in your cloud. Unsynchronized clocks can cause unexpected problems.

   (a) Install NTP

   ```
   $ yum install ntp
   ```

   ```
   $ apt-get install openntpd
   ```
5. Repeat all of these steps on every hypervisor host.

Install and configure the Agent

To manage LXC instances on the host CloudStack uses a Agent. This Agent communicates with the Management server and controls all the instances on the host.

First we start by installing the agent:

In RHEL or CentOS:

```
$ yum install cloudstack-agent
```

In Ubuntu:

```
$ apt-get install cloudstack-agent
```

Next step is to update the Agent configuration setttings. The settings are in `/etc/cloudstack/agent/agent.properties`

1. Set the Agent to run in LXC mode:

   ```
   hypervisor.type=lxc
   ```

2. Optional: If you would like to use direct networking (instead of the default bridge networking), configure these lines:

   ```
   libvirt.vif.driver=com.cloud.hypervisor.kvm.resource.DirectVifDriver
   ```

   ```
   network.direct.source.mode=private
   ```

   ```
   network.direct.device=eth0
   ```

The host is now ready to be added to a cluster. This is covered in a later section, see *Adding a Host*. It is recommended that you continue to read the documentation before adding the host!

Install and Configure libvirt

CloudStack uses libvirt for managing virtual machines. Therefore it is vital that libvirt is configured correctly. Libvirt is a dependency of cloudstack-agent and should already be installed.

1. In order to have live migration working libvirt has to listen for unsecured TCP connections. We also need to turn off libvirts attempt to use Multicast DNS advertising. Both of these settings are in `/etc/libvirt/libvirtd.conf`

   Set the following parameters:

   ```
   listen_tls = 0
   ```

   ```
   listen_tcp = 1
   ```

   ```
   tcp_port = "16509"
   ```

   ```
   auth_tcp = "none"
   ```

   ```
   mdns_adv = 0
   ```

2. Turning on "listen_tcp" in libvirtd.conf is not enough, we have to change the parameters as well:

   On RHEL or CentOS modify `/etc/sysconfig/libvirtd`:

   Uncomment the following line:

```
#LIBVIRTD_ARGS="--listen"
```

   On Ubuntu: modify `/etc/default/libvirt-bin`

   Add "-l" to the following line

```
libvirtd_opts="-d"
```

   so it looks like:

```
libvirtd_opts="-d -l"
```

3. In order to have the VNC Console work we have to make sure it will bind on 0.0.0.0. We do this by editing `/etc/libvirt/qemu.conf`

   Make sure this parameter is set:

```
vnc_listen = "0.0.0.0"
```

4. Restart libvirt

   In RHEL or CentOS:

```
$ service libvirtd restart
```

   In Ubuntu:

```
$ service libvirt-bin restart
```

Configure the Security Policies

CloudStack does various things which can be blocked by security mechanisms like AppArmor and SELinux. These have to be disabled to ensure the Agent has all the required permissions.

1. Configure SELinux (RHEL and CentOS)

   (a) Check to see whether SELinux is installed on your machine. If not, you can skip this section.

   In RHEL or CentOS, SELinux is installed and enabled by default. You can verify this with:

```
$ rpm -qa | grep selinux
```

   (b) Set the SELINUX variable in `/etc/selinux/config` to "permissive". This ensures that the permissive setting will be maintained after a system reboot.

   In RHEL or CentOS:

```
$ vi /etc/selinux/config
```

   Change the following line

```
SELINUX=enforcing
```

   to this

```
SELINUX=permissive
```

(c) Then set SELinux to permissive starting immediately, without requiring a system reboot.

```
$ setenforce permissive
```

2. Configure Apparmor (Ubuntu)

   (a) Check to see whether AppArmor is installed on your machine. If not, you can skip this section.

   In Ubuntu AppArmor is installed and enabled by default. You can verify this with:

```
$ dpkg --list 'apparmor'
```

   (b) Disable the AppArmor profiles for libvirt

```
$ ln -s /etc/apparmor.d/usr.sbin.libvirtd /etc/apparmor.d/disable/
```

```
$ ln -s /etc/apparmor.d/usr.lib.libvirt.virt-aa-helper /etc/apparmor.d/
↪disable/
```

```
$ apparmor_parser -R /etc/apparmor.d/usr.sbin.libvirtd
```

```
$ apparmor_parser -R /etc/apparmor.d/usr.lib.libvirt.virt-aa-helper
```

Configure the network bridges

Warning: This is a very important section, please make sure you read this thoroughly.

Note: This section details how to configure bridges using the native implementation in Linux. Please refer to the next section if you intend to use OpenVswitch

In order to forward traffic to your instances you will need at least two bridges: *public* and *private*.

By default these bridges are called *cloudbr0* and *cloudbr1*, but you do have to make sure they are available on each hypervisor.

The most important factor is that you keep the configuration consistent on all your hypervisors.

Network example

There are many ways to configure your network. In the Basic networking mode you should have two (V)LAN's, one for your private network and one for the public network.

We assume that the hypervisor has one NIC (eth0) with three tagged VLAN's:

1. VLAN 100 for management of the hypervisor
2. VLAN 200 for public network of the instances (cloudbr0)
3. VLAN 300 for private network of the instances (cloudbr1)

On VLAN 100 we give the Hypervisor the IP-Address 192.168.42.11/24 with the gateway 192.168.42.1

Note: The Hypervisor and Management server don't have to be in the same subnet!

Configuring the network bridges

It depends on the distribution you are using how to configure these, below you'll find examples for RHEL/CentOS and Ubuntu.

Note: The goal is to have two bridges called 'cloudbr0' and 'cloudbr1' after this section. This should be used as a guideline only. The exact configuration will depend on your network layout.

Configure in RHEL or CentOS

The required packages were installed when libvirt was installed, we can proceed to configuring the network.

First we configure eth0

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Make sure it looks similar to:

```
DEVICE=eth0
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
```

We now have to configure the three VLAN interfaces:

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0.100
```

```
DEVICE=eth0.100
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
VLAN=yes
IPADDR=192.168.42.11
GATEWAY=192.168.42.1
NETMASK=255.255.255.0
```

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0.200
```

```
DEVICE=eth0.200
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
```

(continues on next page)

(continued from previous page)

```
BOOTPROTO=none
TYPE=Ethernet
VLAN=yes
BRIDGE=cloudbr0
```

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0.300
```

```
DEVICE=eth0.300
HWADDR=00:04:xx:xx:xx:xx
ONBOOT=yes
HOTPLUG=no
BOOTPROTO=none
TYPE=Ethernet
VLAN=yes
BRIDGE=cloudbr1
```

Now we have the VLAN interfaces configured we can add the bridges on top of them.

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr0
```

Now we just configure it is a plain bridge without an IP-Address

```
DEVICE=cloudbr0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

We do the same for cloudbr1

```
$ vi /etc/sysconfig/network-scripts/ifcfg-cloudbr1
```

```
DEVICE=cloudbr1
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configure in Ubuntu

All the required packages were installed when you installed libvirt, so we only have to configure the network.

```
$ vi /etc/network/interfaces
```

Modify the interfaces file to look like this:

```
auto lo
iface lo inet loopback

# The primary network interface
auto eth0.100
iface eth0.100 inet static
    address 192.168.42.11
    netmask 255.255.255.240
    gateway 192.168.42.1
    dns-nameservers 8.8.8.8 8.8.4.4
    dns-domain lab.example.org

# Public network
auto cloudbr0
iface cloudbr0 inet manual
    bridge_ports eth0.200
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1

# Private network
auto cloudbr1
iface cloudbr1 inet manual
    bridge_ports eth0.300
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configuring the firewall

The hypervisor needs to be able to communicate with other hypervisors and the management server needs to be able to reach the hypervisor.

In order to do so we have to open the following TCP ports (if you are using a firewall):

1. 22 (SSH)
2. 1798
3. 16509 (libvirt)
4. 5900 - 6100 (VNC consoles)
5. 49152 - 49216 (libvirt live migration)

It depends on the firewall you are using how to open these ports. Below you'll find examples how to open these ports in RHEL/CentOS and Ubuntu.

Open ports in RHEL/CentOS

RHEL and CentOS use iptables for firewalling the system, you can open extra ports by executing the following iptable commands:

```
$ iptables -I INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 1798 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 16509 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 5900:6100 -j ACCEPT
```

```
$ iptables -I INPUT -p tcp -m tcp --dport 49152:49216 -j ACCEPT
```

These iptable settings are not persistent accross reboots, we have to save them first.

```
$ iptables-save > /etc/sysconfig/iptables
```

Open ports in Ubuntu

The default firewall under Ubuntu is UFW (Uncomplicated FireWall), which is a Python wrapper around iptables.

To open the required ports, execute the following commands:

```
$ ufw allow proto tcp from any to any port 22
```

```
$ ufw allow proto tcp from any to any port 1798
```

```
$ ufw allow proto tcp from any to any port 16509
```

```
$ ufw allow proto tcp from any to any port 5900:6100
```

```
$ ufw allow proto tcp from any to any port 49152:49216
```

Note: By default UFW is not enabled on Ubuntu. Executing these commands with the firewall disabled does not enable the firewall.

Add the host to CloudStack

The host is now ready to be added to a cluster. This is covered in a later section, see *Adding a Host*. It is recommended that you continue to read the documentation before adding the host!

3.4.4 Host VMware vSphere Installation

If you want to use the VMware vSphere hypervisor to run guest virtual machines, install vSphere on the host(s) in your cloud.

System Requirements for vSphere Hosts

Software requirements:

- vSphere and vCenter, versions 4.1, 5.0, 5.1 or 5.5.

  vSphere Standard is recommended. Note however that customers need to consider the CPU constraints in place with vSphere licensing. See http://www.vmware.com/files/pdf/vsphere_pricing.pdf and discuss with your VMware sales representative.

  vCenter Server Standard is recommended.

- Be sure all the hotfixes provided by the hypervisor vendor are applied. Track the release of hypervisor patches through your hypervisor vendor's support channel, and apply patches as soon as possible after they are released. CloudStack will not track or notify you of required hypervisor patches. It is essential that your hosts are completely up to date with the provided hypervisor patches. The hypervisor vendor is likely to refuse to support any system that is not up to date with patches.

Warning: Apply All Necessary Hotfixes. The lack of up-do-date hotfixes can lead to data corruption and lost VMs.

Hardware requirements:

- The host must be certified as compatible with vSphere. See the VMware Hardware Compatibility Guide at http://www.vmware.com/resources/compatibility/search.php.
- All hosts must be 64-bit and must support HVM (Intel-VT or AMD-V enabled).
- All hosts within a cluster must be homogenous. That means the CPUs must be of the same type, count, and feature flags.
- 64-bit x86 CPU (more cores results in better performance)
- Hardware virtualization support required
- 4 GB of memory
- 36 GB of local disk
- At least 1 NIC
- Statically allocated IP Address

vCenter Server requirements:

- Processor - 2 CPUs 2.0GHz or higher Intel or AMD x86 processors. Processor requirements may be higher if the database runs on the same machine.
- Memory - 3GB RAM. RAM requirements may be higher if your database runs on the same machine.
- Disk storage - 2GB. Disk requirements may be higher if your database runs on the same machine.
- Microsoft SQL Server 2005 Express disk requirements. The bundled database requires up to 2GB free disk space to decompress the installation archive.
- Networking - 1Gbit or 10Gbit.

For more information, see "vCenter Server and the vSphere Client Hardware Requirements".

Other requirements:

- VMware vCenter Standard Edition 4.1, 5.0, 5.1 or 5.5 must be installed and available to manage the vSphere hosts.
- vCenter must be configured to use the standard port 443 so that it can communicate with the CloudStack Management Server.
- You must re-install VMware ESXi if you are going to re-use a host from a previous install.
- CloudStack requires VMware vSphere 4.1, 5.0, 5.1 or 5.5. VMware vSphere 4.0 is not supported.
- All hosts must be 64-bit and must support HVM (Intel-VT or AMD-V enabled). All hosts within a cluster must be homogeneous. That means the CPUs must be of the same type, count, and feature flags.
- The CloudStack management network must not be configured as a separate virtual network. The CloudStack management network is the same as the vCenter management network, and will inherit its configuration. See *Configure vCenter Management Network*.
- CloudStack requires ESXi and vCenter. ESX is not supported.
- Ideally all resources used for CloudStack must be used for CloudStack only. CloudStack should not share instance of ESXi or storage with other management consoles. Do not share the same storage volumes that will be used by CloudStack with a different set of ESXi servers that are not managed by CloudStack.
- Put all target ESXi hypervisors in dedicated clusters in a separate Datacenter in vCenter.
- Ideally clusters that will be managed by CloudStack should not contain any other VMs. Do not run the management server or vCenter on the cluster that is designated for CloudStack use. Create a separate cluster for use of CloudStack and make sure that they are no VMs in this cluster.
- All of the required VLANs must be trunked into all network switches that are connected to the ESXi hypervisor hosts. These would include the VLANs for Management, Storage, vMotion, and guest VLANs. The guest VLAN (used in Advanced Networking; see Network Setup) is a contiguous range of VLANs that will be managed by CloudStack.

Preparation Checklist for VMware

For a smoother installation, gather the following information before you start:

- Information listed in *vCenter Checklist*
- Information listed in *Networking Checklist for VMware*

vCenter Checklist

You will need the following information about vCenter.

vCenter Requirement	Notes
vCenter User	This user must have admin privileges.
vCenter User Password	Password for the above user.
vCenter Datacenter Name	Name of the datacenter.
vCenter Cluster Name	Name of the cluster.

Networking Checklist for VMware

You will need the following information about your VLANs.

VLAN Information	Notes
ESXi VLAN	VLAN on which all your ESXi hypervisors reside.
ESXI VLAN IP Address	IP Address Range in the ESXi VLAN. One address per Virtual Router is used from this range.
ESXi VLAN IP Gateway	
ESXi VLAN Netmask	
Management Server VLAN	VLAN on which the CloudStack Management server is installed.
Public VLAN	VLAN for the Public Network.
Public VLAN Gateway	
Public VLAN Netmask	
Public VLAN IP Address Range	Range of Public IP Addresses available for CloudStack use. These addresses will be used for virtual router on CloudStack to route private traffic to external networks.
VLAN Range for Customer use	A contiguous range of non-routable VLANs. One VLAN will be assigned for each customer.

vSphere Installation Steps

1. If you haven't already, you'll need to download and purchase vSphere from the VMware Website (https://www.vmware.com/tryvmware/index.php?p=vmware-vsphere&lp=1) and install it by following the VMware vSphere Installation Guide.
2. Following installation, perform the following configuration, which are described in the next few sections:

Required	Optional
ESXi host setup	NIC bonding
Configure host physical networking,virtual switch, vCenter Management Network, and extended port range	Multipath storage
Prepare storage for iSCSI	
Configure clusters in vCenter and add hosts to them, or add hosts without clusters to vCenter	

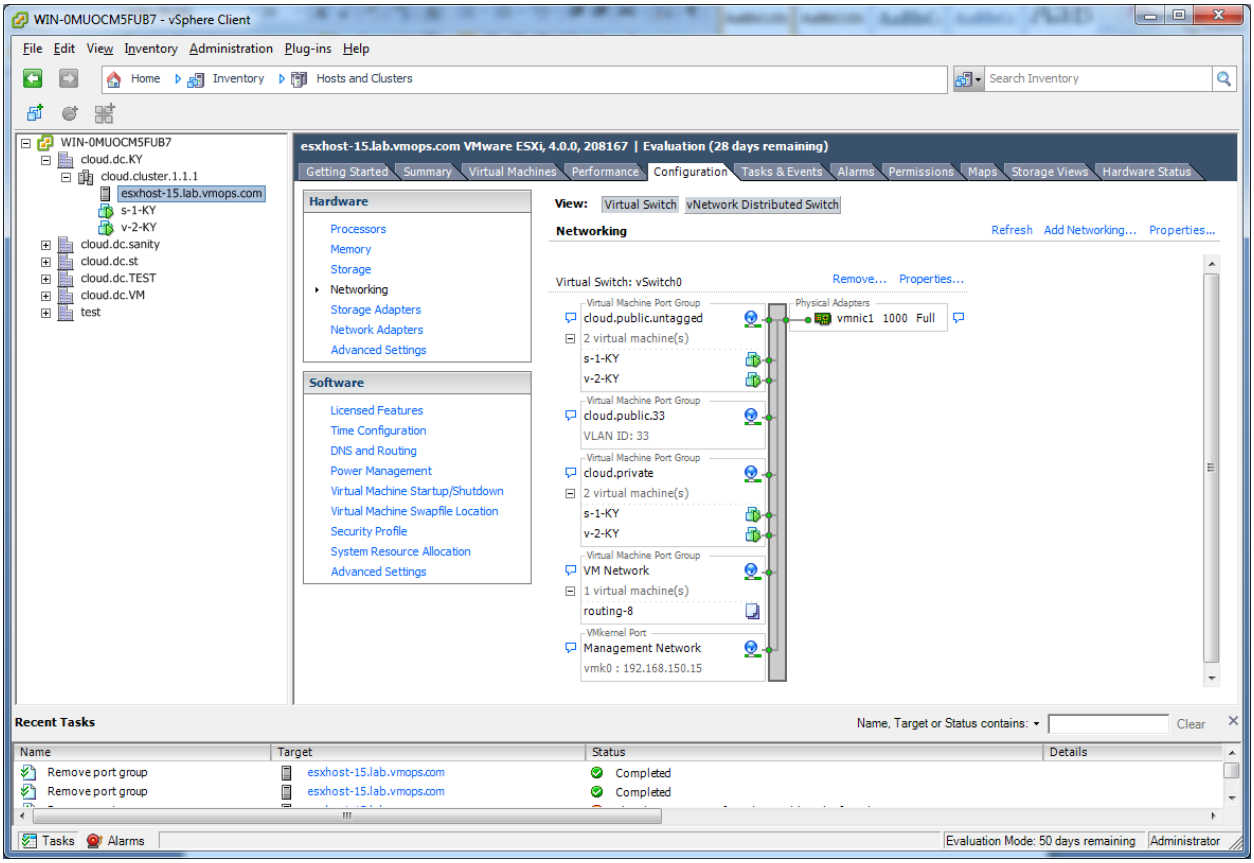
ESXi Host setup

All ESXi hosts should have CPU hardware virtualization support enabled in the BIOS. Please note hardware virtualization support is not enabled by default on most servers.

Physical Host Networking

You should have a plan for cabling the vSphere hosts. Proper network configuration is required before adding a vSphere host to CloudStack. To configure an ESXi host, you can use vClient to add it as standalone host to vCenter

first. Once you see the host appearing in the vCenter inventory tree, click the host node in the inventory tree, and navigate to the Configuration tab.



In the host configuration tab, click the "Hardware/Networking" link to bring up the networking configuration page as above.

Configure Virtual Switch

During the initial installation of an ESXi host a default virtual switch vSwitch0 is created. You may need to create additional vSwiches depending on your required architecture. CloudStack requires all ESXi hosts in the cloud to use consistently named virtual switches. If you change the default virtual switch name, you will need to configure one or more CloudStack configuration variables as well.

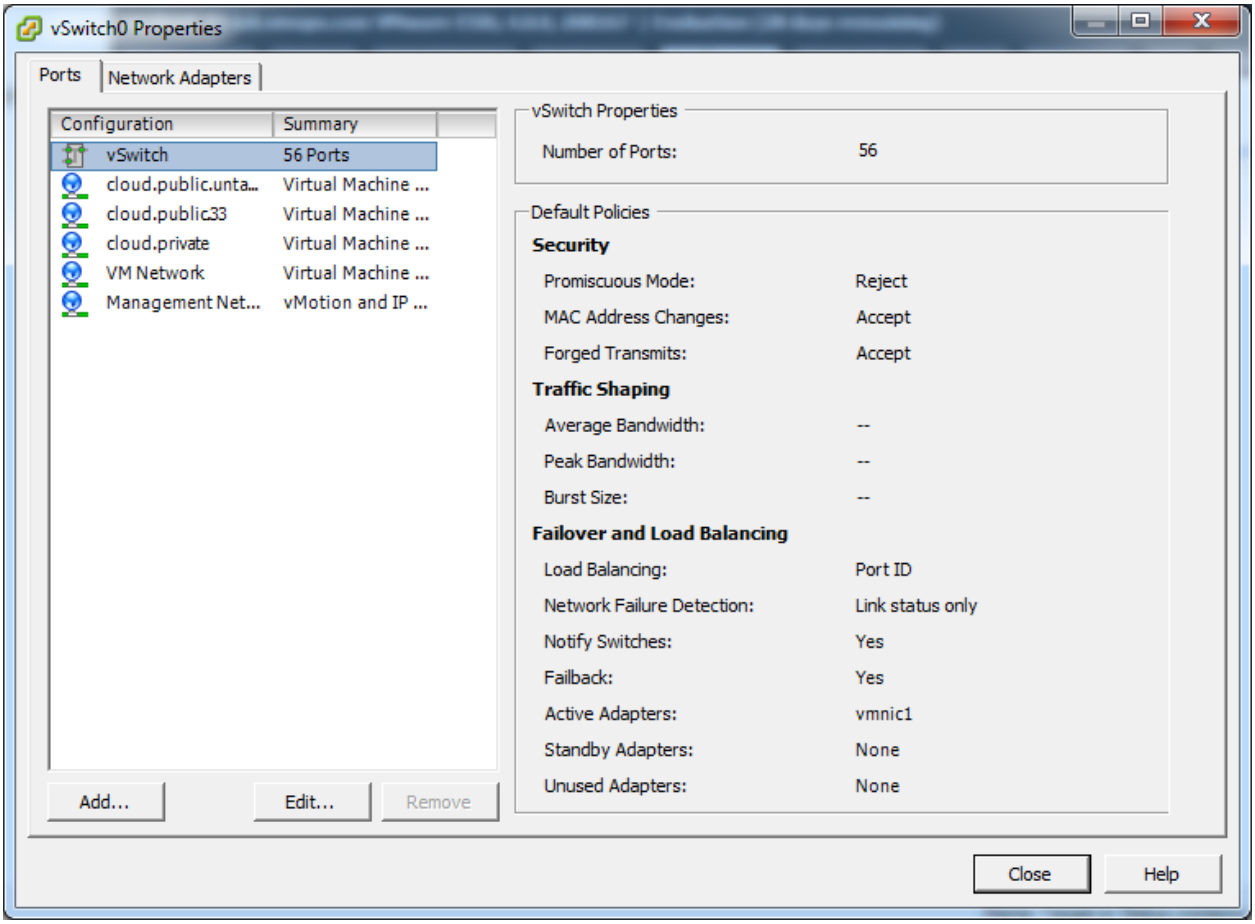
Separating Traffic

CloudStack allows you to configure three separate networks per ESXi host. CloudStack identifies these networks by the name of the vSwitch they are connected to. The networks for configuration are public (for traffic to/from the public internet), guest (for guest-guest traffic), and private (for management and usually storage traffic). You can use the default virtual switch for all three, or create one or two other vSwitches for those traffic types.

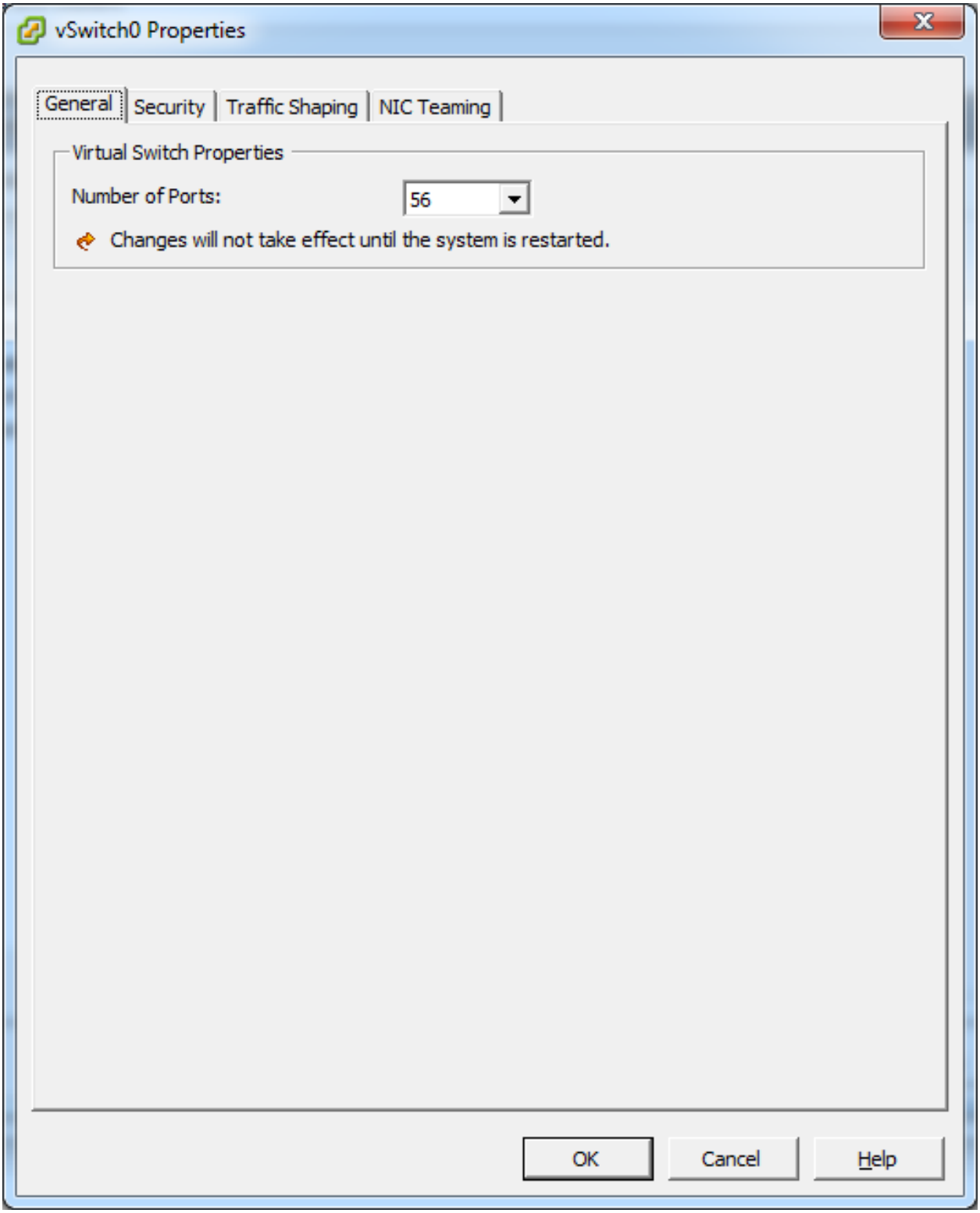
If you want to separate traffic in this way you should first create and configure vSwitches in vCenter according to the vCenter instructions. Take note of the vSwitch names you have used for each traffic type. You will configure CloudStack to use these vSwitches.

Increasing Ports

By default a virtual switch on ESXi hosts is created with 56 ports. We recommend setting it to 4088, the maximum number of ports allowed. To do that, click the "Properties…" link for virtual switch (note this is not the Properties link for Networking).



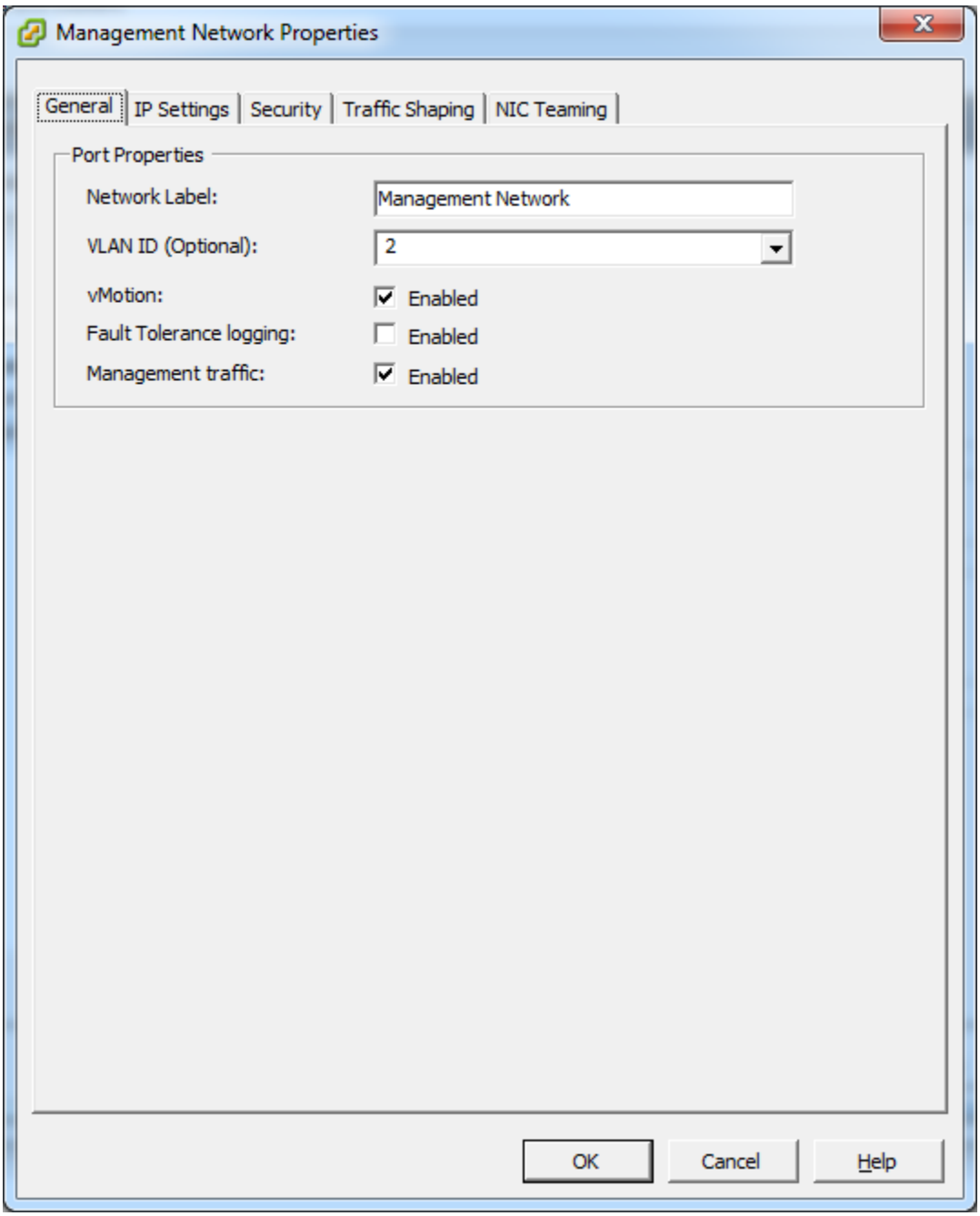
In vSwitch properties dialog, select the vSwitch and click Edit. You should see the following dialog:



In this dialog, you can change the number of switch ports. After you've done that, ESXi hosts are required to reboot in order for the setting to take effect.

Configure vCenter Management Network

In the vSwitch properties dialog box, you may see a vCenter management network. This same network will also be used as the CloudStack management network. CloudStack requires the vCenter management network to be configured properly. Select the management network item in the dialog, then click Edit.



Make sure the following values are set:

- VLAN ID set to the desired ID
- vMotion enabled.
- Management traffic enabled.

If the ESXi hosts have multiple VMKernel ports, and ESXi is not using the default value "Management Network" as the management network name, you must follow these guidelines to configure the management network port group so that CloudStack can find it:

- Use one label for the management network port across all ESXi hosts.
- In the CloudStack UI, go to Configuration - Global Settings and set vmware.management.portgroup to the management network label from the ESXi hosts.

Extend Port Range for CloudStack Console Proxy

(Applies only to VMware vSphere version 4.x)

You need to extend the range of firewall ports that the console proxy works with on the hosts. This is to enable the console proxy to work with VMware-based VMs. The default additional port range is 59000-60000. To extend the port range, log in to the VMware ESX service console on each host and run the following commands:

```
esxcfg-firewall -o 59000-60000,tcp,in,vncextras
esxcfg-firewall -o 59000-60000,tcp,out,vncextras
```

Configure NIC Bonding for vSphere

NIC bonding on vSphere hosts may be done according to the vSphere installation guide.

Configuring a vSphere Cluster with Nexus 1000v Virtual Switch

CloudStack supports Cisco Nexus 1000v dvSwitch (Distributed Virtual Switch) for virtual network configuration in a VMware vSphere environment. This section helps you configure a vSphere cluster with Nexus 1000v virtual switch in a VMware vCenter environment. For information on creating a vSphere cluster, see *"VMware vSphere Installation and Configuration"*

About Cisco Nexus 1000v Distributed Virtual Switch

The Cisco Nexus 1000V virtual switch is a software-based virtual machine access switch for VMware vSphere environments. It can span multiple hosts running VMware ESXi 4.0 and later. A Nexus virtual switch consists of two components: the Virtual Supervisor Module (VSM) and the Virtual Ethernet Module (VEM). The VSM is a virtual appliance that acts as the switch's supervisor. It controls multiple VEMs as a single network device. The VSM is installed independent of the VEM and is deployed in redundancy mode as pairs or as a standalone appliance. The VEM is installed on each VMware ESXi server to provide packet-forwarding capability. It provides each virtual machine with dedicated switch ports. This VSM-VEM architecture is analogous to a physical Cisco switch's supervisor (standalone or configured in high-availability mode) and multiple linecards architecture.

Nexus 1000v switch uses vEthernet port profiles to simplify network provisioning for virtual machines. There are two types of port profiles: Ethernet port profile and vEthernet port profile. The Ethernet port profile is applied to the physical uplink ports-the NIC ports of the physical NIC adapter on an ESXi server. The vEthernet port profile is associated with the virtual NIC (vNIC) that is plumbed on a guest VM on the ESXi server. The port profiles help the network administrators define network policies which can be reused for new virtual machines. The Ethernet port profiles are created on the VSM and are represented as port groups on the vCenter server.

Prerequisites and Guidelines

This section discusses prerequisites and guidelines for using Nexus virtual switch in CloudStack. Before configuring Nexus virtual switch, ensure that your system meets the following requirements:

- A cluster of servers (ESXi 4.1 or later) is configured in the vCenter.
- Each cluster managed by CloudStack is the only cluster in its vCenter datacenter.
- A Cisco Nexus 1000v virtual switch is installed to serve the datacenter that contains the vCenter cluster. This ensures that CloudStack doesn't have to deal with dynamic migration of virtual adapters or networks across other existing virtual switches. See Cisco Nexus 1000V Installation and Upgrade Guide for guidelines on how to install the Nexus 1000v VSM and VEM modules.
- The Nexus 1000v VSM is not deployed on a vSphere host that is managed by CloudStack.
- When the maximum number of VEM modules per VSM instance is reached, an additional VSM instance is created before introducing any more ESXi hosts. The limit is 64 VEM modules for each VSM instance.
- CloudStack expects that the Management Network of the ESXi host is configured on the standard vSwitch and searches for it in the standard vSwitch. Therefore, ensure that you do not migrate the management network to Nexus 1000v virtual switch during configuration.
- All information given in *Nexus 1000v Virtual Switch Preconfiguration*

Nexus 1000v Virtual Switch Preconfiguration

Preparation Checklist

For a smoother configuration of Nexus 1000v switch, gather the following information before you start:

- vCenter credentials
- Nexus 1000v VSM IP address
- Nexus 1000v VSM Credentials
- Ethernet port profile names

vCenter Credentials Checklist

You will need the following information about vCenter:

Nexus vSwitch Requirements	Value	Notes
vCenter IP		The IP address of the vCenter.
Secure HTTP Port Number	443	Port 443 is configured by default; however, you can change the port if needed.
vCenter User ID		The vCenter user with administrator-level privileges. The vCenter User ID is required when you configure the virtual switch in CloudStack.
vCenter Password		The password for the vCenter user specified above. The password for this vCenter user is required when you configure the switch in CloudStack.

Network Configuration Checklist

The following information specified in the Nexus Configure Networking screen is displayed in the Details tab of the Nexus dvSwitch in the CloudStack UI:

Control Port Group VLAN ID The VLAN ID of the Control Port Group. The control VLAN is used for communication between the VSM and the VEMs.

Management Port Group VLAN ID The VLAN ID of the Management Port Group. The management VLAN corresponds to the mgmt0 interface that is used to establish and maintain the connection between the VSM and VMware vCenter Server.

Packet Port Group VLAN ID The VLAN ID of the Packet Port Group. The packet VLAN forwards relevant data packets from the VEMs to the VSM.

Note: The VLANs used for control, packet, and management port groups can be the same.

For more information, see Cisco Nexus 1000V Getting Started Guide.

VSM Configuration Checklist

You will need the following VSM configuration parameters:

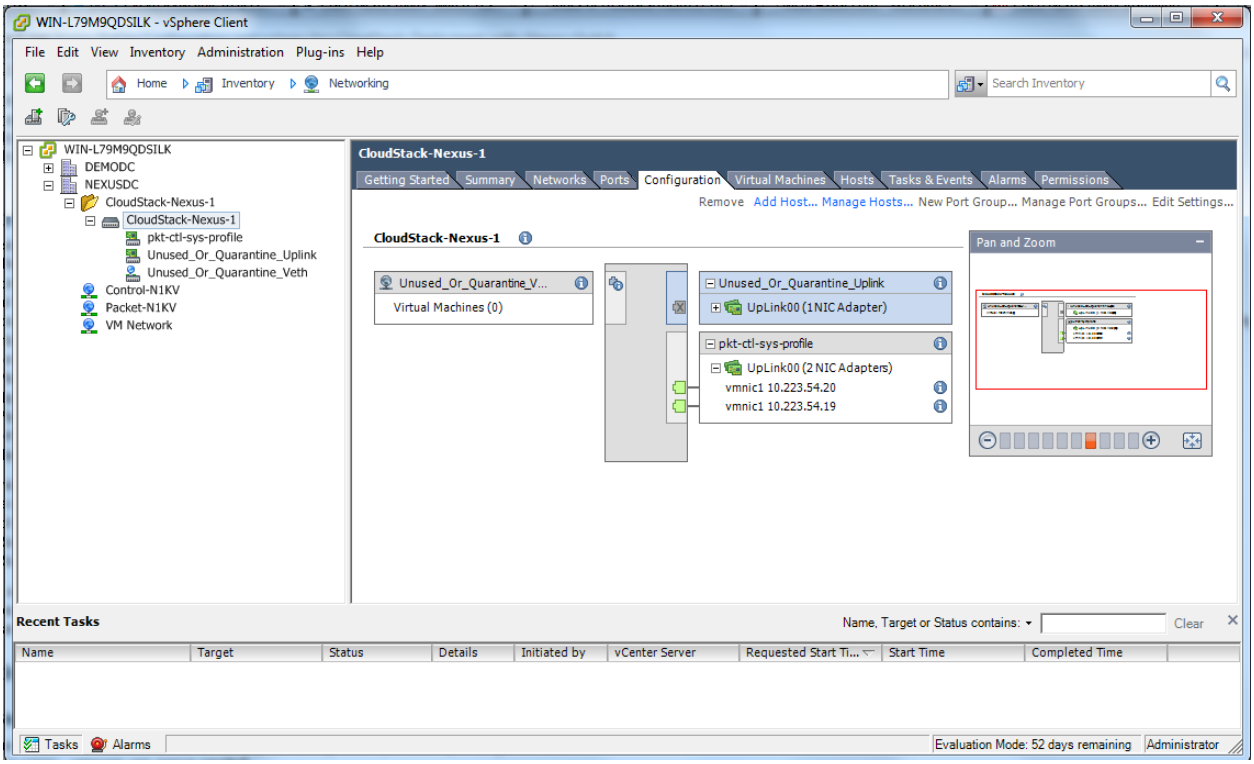
Admin Name and Password The admin name and password to connect to the VSM appliance. You must specify these credentials while configuring Nexus virtual switch.

Management IP Address This is the IP address of the VSM appliance. This is the IP address you specify in the virtual switch IP Address field while configuting Nexus virtual switch.

SSL Should be set to Enable.Always enable SSL. SSH is usually enabled by default during the VSM installation. However, check whether the SSH connection to the VSM is working, without which CloudStack failes to connect to the VSM.

Creating a Port Profile

- Whether you create a Basic or Advanced zone configuration, ensure that you always create an Ethernet port profile on the VSM after you install it and before you create the zone.
  - The Ethernet port profile created to represent the physical network or networks used by an Advanced zone configuration trunk all the VLANs including guest VLANs, the VLANs that serve the native VLAN, and the packet/control/data/management VLANs of the VSM.
  - The Ethernet port profile created for a Basic zone configuration does not trunk the guest VLANs because the guest VMs do not get their own VLANs provisioned on their network interfaces in a Basic zone.
- An Ethernet port profile configured on the Nexus 1000v virtual switch should not use in its set of system VLANs, or any of the VLANs configured or intended to be configured for use towards VMs or VM resources in the CloudStack environment.
- You do not have to create any vEthernet port profiles – CloudStack does that during VM deployment.
- Ensure that you create required port profiles to be used by CloudStack for different traffic types of CloudStack, such as Management traffic, Guest traffic, Storage traffic, and Public traffic. The physical networks configured during zone creation should have a one-to-one relation with the Ethernet port profiles.



For information on creating a port profile, see Cisco Nexus 1000V Port Profile Configuration Guide.

Assigning Physical NIC Adapters

Assign ESXi host's physical NIC adapters, which correspond to each physical network, to the port profiles. In each ESXi host that is part of the vCenter cluster, observe the physical networks assigned to each port profile and note down the names of the port profile for future use. This mapping information helps you when configuring physical networks during the zone configuration on CloudStack. These Ethernet port profile names are later specified as VMware Traffic Labels for different traffic types when configuring physical networks during the zone configuration. For more information on configuring physical networks, see *Configuring a vSphere Cluster with Nexus 1000v Virtual Switch*.

Adding VLAN Ranges

Determine the public VLAN, System VLAN, and Guest VLANs to be used by the CloudStack. Ensure that you add them to the port profile database. Corresponding to each physical network, add the VLAN range to port profiles. In the VSM command prompt, run the switchport trunk allowed vlan<range> command to add the VLAN ranges to the port profile.

For example:

```
switchport trunk allowed vlan 1,140-147,196-203
```

In this example, the allowed VLANs added are 1, 140-147, and 196-203

You must also add all the public and private VLANs or VLAN ranges to the switch. This range is the VLAN range you specify in your zone.

Note: Before you run the vlan command, ensure that the configuration mode is enabled in Nexus 1000v virtual

switch.

For example:

If you want the VLAN 200 to be used on the switch, run the following command:

```
vlan 200
```

If you want the VLAN range 1350-1750 to be used on the switch, run the following command:

```
vlan 1350-1750
```

Refer to Cisco Nexus 1000V Command Reference of specific product version.

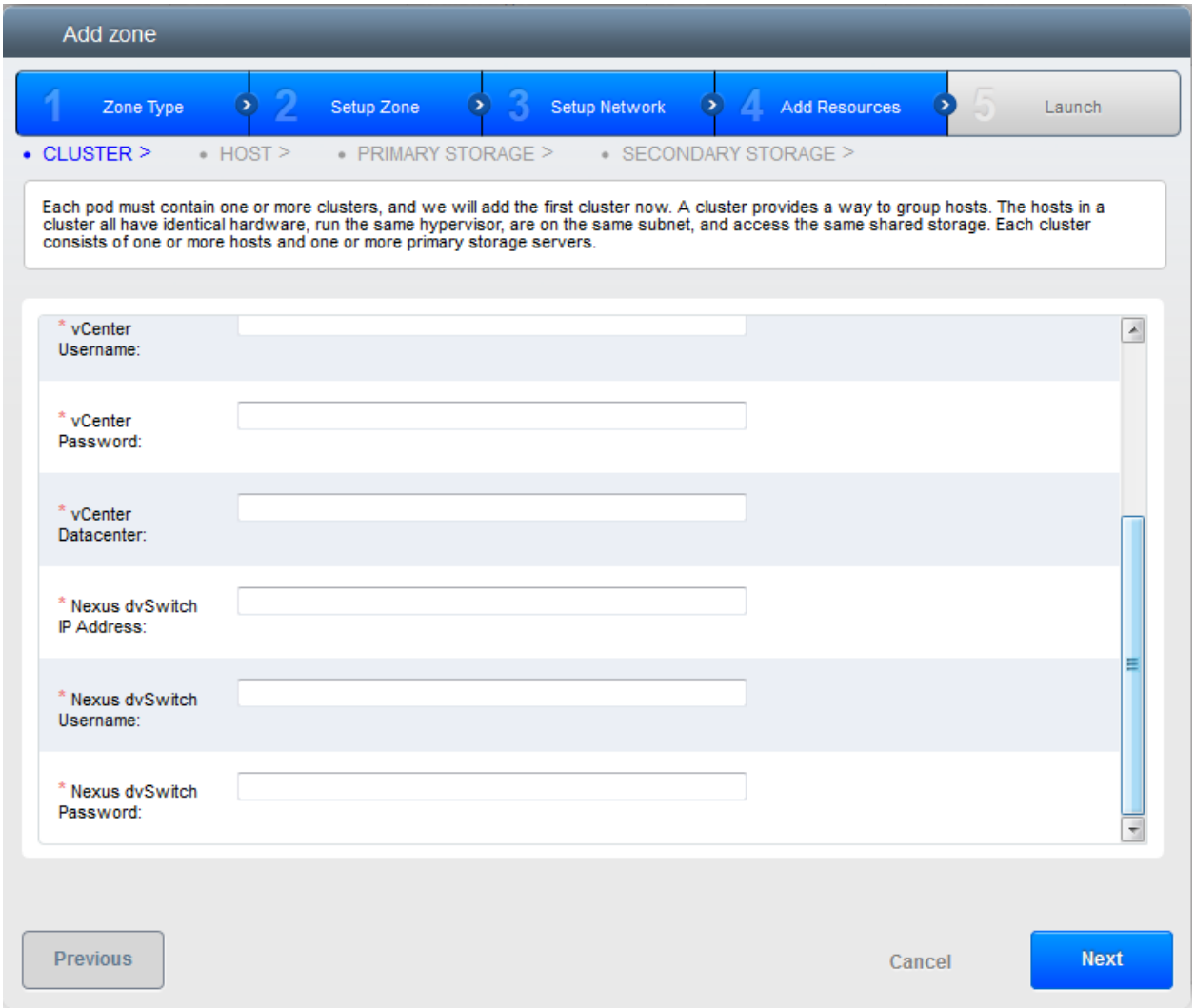
Enabling Nexus Virtual Switch in CloudStack

To make a CloudStack deployment Nexus enabled, you must set the vmware.use.nexus.vswitch parameter true by using the Global Settings page in the CloudStack UI. Unless this parameter is set to "true" and restart the management server, you cannot see any UI options specific to Nexus virtual switch, and CloudStack ignores the Nexus virtual switch specific parameters specified in the AddTrafficTypeCmd, UpdateTrafficTypeCmd, and AddClusterCmd API calls.

Unless the CloudStack global parameter "vmware.use.nexus.vswitch" is set to "true", CloudStack by default uses VMware standard vSwitch for virtual network infrastructure. In this release, CloudStack doesn't support configuring virtual networks in a deployment with a mix of standard vSwitch and Nexus 1000v virtual switch. The deployment can have either standard vSwitch or Nexus 1000v virtual switch.

Configuring Nexus 1000v Virtual Switch in CloudStack

You can configure Nexus dvSwitch by adding the necessary resources while the zone is being created.



After the zone is created, if you want to create an additional cluster along with Nexus 1000v virtual switch in the existing zone, use the Add Cluster option. For information on creating a cluster, see "Add Cluster: vSphere".

In both these cases, you must specify the following parameters to configure Nexus virtual switch:

Parameters	Description
Cluster Name	Enter the name of the cluster you created in vCenter. For example,"cloud.cluster".
vCenter Host	Enter the host name or the IP address of the vCenter host where you have deployed the Nexus virtual switch.
vCenter User name	Enter the username that CloudStack should use to connect to vCenter. This user must have all administrative privileges.
vCenter Password	Enter the password for the user named above.
vCenter Datacenter	Enter the vCenter datacenter that the cluster is in. For example, "cloud.dc.VM".
Nexus dvSwitch IP Address	The IP address of the VSM component of the Nexus 1000v virtual switch.
Nexus dvSwitch Username	The admin name to connect to the VSM appliance.
Nexus dvSwitch Password	The corresponding password for the admin user specified above.

Removing Nexus Virtual Switch

1. In the vCenter datacenter that is served by the Nexus virtual switch, ensure that you delete all the hosts in the corresponding cluster.
2. Log in with Admin permissions to the CloudStack administrator UI.
3. In the left navigation bar, select Infrastructure.
4. In the Infrastructure page, click View all under Clusters.
5. Select the cluster where you want to remove the virtual switch.
6. In the dvSwitch tab, click the name of the virtual switch.
7. In the Details page, click Delete Nexus dvSwitch icon.

   Click Yes in the confirmation dialog box.

Configuring a VMware Datacenter with VMware Distributed Virtual Switch

CloudStack supports VMware vNetwork Distributed Switch (VDS) for virtual network configuration in a VMware vSphere environment. This section helps you configure VMware VDS in a CloudStack deployment. Each vCenter server instance can support up to 128 VDS instances and each VDS instance can manage up to 500 VMware hosts.

About VMware Distributed Virtual Switch

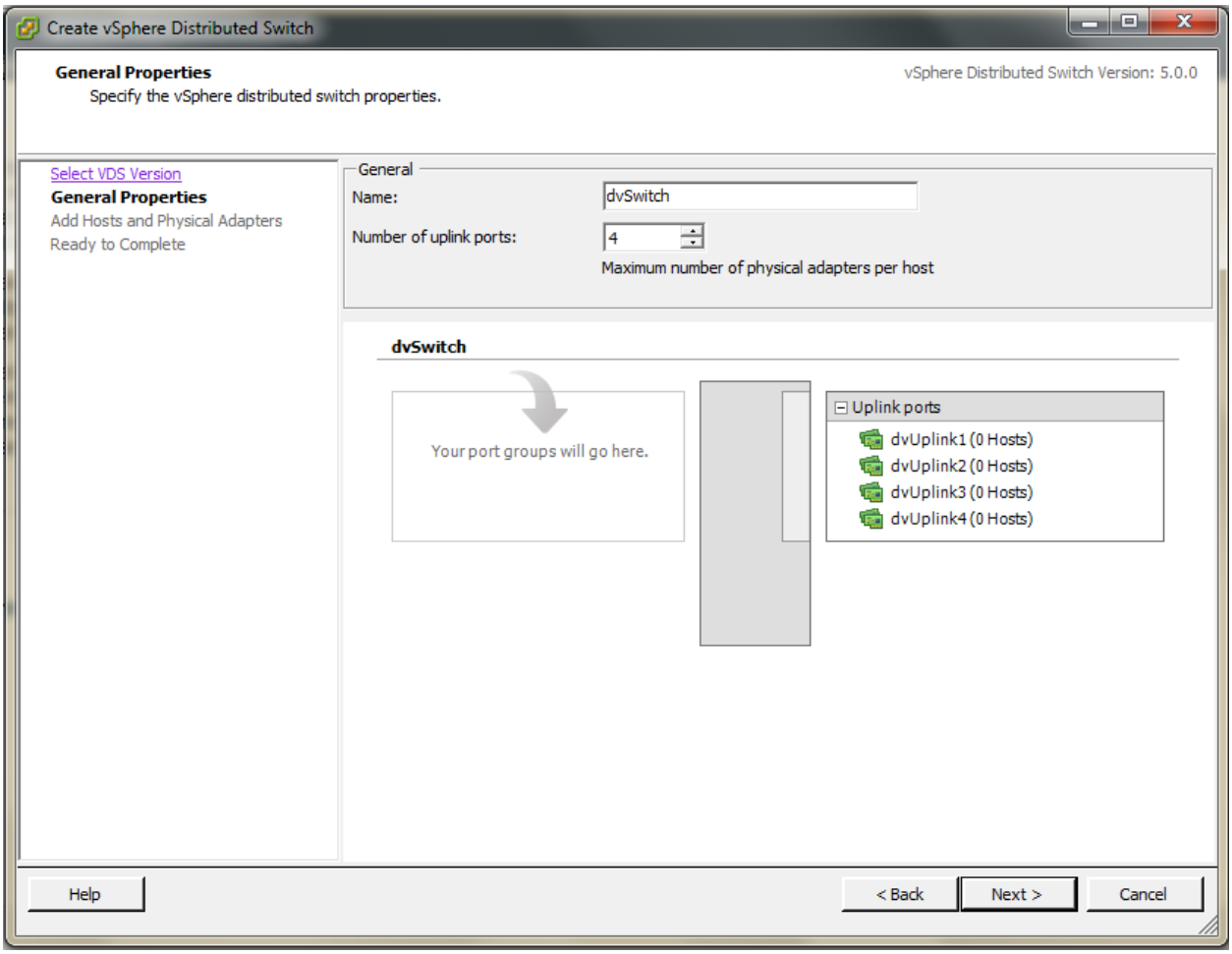
VMware VDS is an aggregation of host-level virtual switches on a VMware vCenter server. VDS abstracts the configuration of individual virtual switches that span across a large number of hosts, and enables centralized provisioning, administration, and monitoring for your entire datacenter from a centralized interface. In effect, a VDS acts as a single virtual switch at the datacenter level and manages networking for a number of hosts in a datacenter from a centralized VMware vCenter server. Each VDS maintains network runtime state for VMs as they move across multiple hosts, enabling inline monitoring and centralized firewall services. A VDS can be deployed with or without Virtual Standard Switch and a Nexus 1000V virtual switch.

Prerequisites and Guidelines

- VMware VDS is supported only on Public and Guest traffic in CloudStack.
- VMware VDS does not support multiple VDS per traffic type. If a user has many VDS switches, only one can be used for Guest traffic and another one for Public traffic.
- Additional switches of any type can be added for each cluster in the same zone. While adding the clusters with different switch type, traffic labels is overridden at the cluster level.
- Management and Storage network does not support VDS. Therefore, use Standard Switch for these networks.
- When you remove a guest network, the corresponding dvportgroup will not be removed on the vCenter. You must manually delete them on the vCenter.

Preparation Checklist

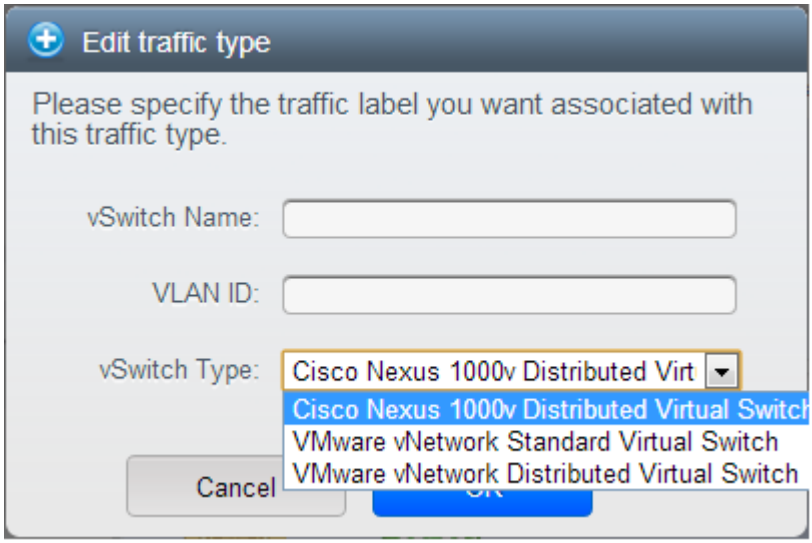
For a smoother configuration of VMware VDS, note down the VDS name you have added in the datacenter before you start:



Use this VDS name in the following:

- The switch name in the Edit traffic label dialog while configuring a public and guest traffic during zone creation.

  During a zone creation, ensure that you select VMware vNetwork Distributed Virtual Switch when you configure guest and public traffic type.



- The Public Traffic vSwitch Type field when you add a VMware VDS-enabled cluster.
- The switch name in the traffic label while updating the switch type in a zone.

Traffic label format in the last case is [["Name of vSwitch/dvSwitch/EthernetPortProfile"][,"VLAN ID"[,"vSwitch Type"]]]

The possible values for traffic labels are:

- empty string
- dvSwitch0
- dvSwitch0,200
- dvSwitch1,300,vmwaredvs
- myEthernetPortProfile,,nexusdvs
- dvSwitch0,,vmwaredvs

The three fields to fill in are:

- Name of the virtual / distributed virtual switch at vCenter.

  The default value depends on the type of virtual switch:

  vSwitch0: If type of virtual switch is VMware vNetwork Standard virtual switch

  dvSwitch0: If type of virtual switch is VMware vNetwork Distributed virtual switch

  epp0: If type of virtual switch is Cisco Nexus 1000v Distributed virtual switch

- VLAN ID to be used for this traffic wherever applicable.

  This field would be used for only public traffic as of now. In case of guest traffic this field would be ignored and could be left empty for guest traffic. By default empty string would be assumed which translates to untagged VLAN for that specific traffic type.

- Type of virtual switch. Specified as string.

  Possible valid values are vmwaredvs, vmwaresvs, nexusdvs.

  vmwaresvs: Represents VMware vNetwork Standard virtual switch

  vmwaredvs: Represents VMware vNetwork distributed virtual switch

  nexusdvs: Represents Cisco Nexus 1000v distributed virtual switch.

  If nothing specified (left empty), zone-level default virtual switchwould be defaulted, based on the value of global parameter you specify.

  Following are the global configuration parameters:

  vmware.use.dvswitch: Set to true to enable any kind (VMware DVS and Cisco Nexus 1000v) of distributed virtual switch in a CloudStack deployment. If set to false, the virtual switch that can be used in that CloudStack deployment is Standard virtual switch.

  vmware.use.nexus.vswitch: This parameter is ignored if vmware.use.dvswitch is set to false. Set to true to enable Cisco Nexus 1000v distributed virtual switch in a CloudStack deployment.

Enabling Virtual Distributed Switch in CloudStack

To make a CloudStack deployment VDS enabled, set the vmware.use.dvswitch parameter to true by using the Global Settings page in the CloudStack UI and restart the Management Server. Unless you enable the vmware.use.dvswitch parameter, you cannot see any UI options specific to VDS, and CloudStack ignores the VDS-specific parameters that

you specify. Additionally, CloudStack uses VDS for virtual network infrastructure if the value of vmware.use.dvswitch parameter is true and the value of vmware.use.nexus.dvswitch parameter is false. Another global parameter that defines VDS configuration is vmware.ports.per.dvportgroup. This is the default number of ports per VMware dvPortGroup in a VMware environment. Default value is 256. This number directly associated with the number of guest network you can create.

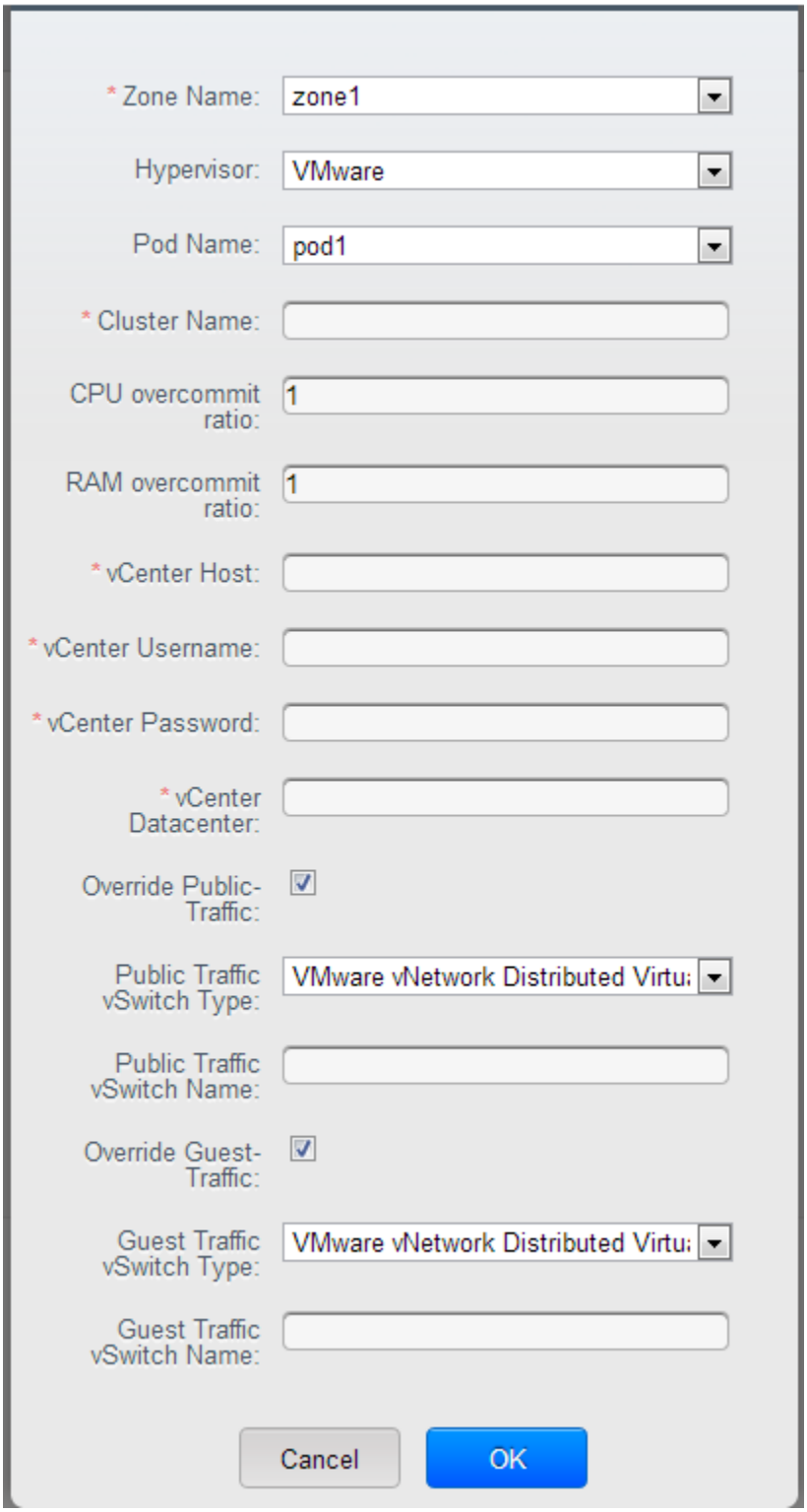
CloudStack supports orchestration of virtual networks in a deployment with a mix of Virtual Distributed Switch, Standard Virtual Switch and Nexus 1000v Virtual Switch.

Configuring Distributed Virtual Switch in CloudStack

You can configure VDS by adding the necessary resources while a zone is created.

Alternatively, at the cluster level, you can create an additional cluster with VDS enabled in the existing zone. Use the Add Cluster option. For information as given in "Add Cluster: vSphere".

In both these cases, you must specify the following parameters to configure VDS:


* Zone Name:
zone1
Hypervisor:
VMware
Pod Name:
pod1
* Cluster Name:
CPU overcommit ratio:
1
RAM overcommit ratio:
1
* vCenter Host:
* vCenter Username:
* vCenter Password:
* vCenter Datacenter:
Override Public-Traffic:
Public Traffic vSwitch Type:
VMware vNetwork Distributed Virtu:
Public Traffic vSwitch Name:
Override Guest-Traffic:
Guest Traffic vSwitch Type:
VMware vNetwork Distributed Virtu:
Guest Traffic vSwitch Name:
Cancel
OK

Parameters Description	
Cluster Name	Enter the name of the cluster you created in vCenter. For example, "cloudcluster".
vCenter Host	Enter the name or the IP address of the vCenter host where you have deployed the VMware VDS.
vCenter User name	Enter the username that CloudStack should use to connect to vCenter. This user must have all administrative privileges.
vCenter Password	Enter the password for the user named above.
vCenter Datacenter	Enter the vCenter datacenter that the cluster is in. For example, "clouddcVM".
Override Public Traffic	Enable this option to override the zone-wide public traffic for the cluster you are creating.
Public Traffic vSwitch Type	This option is displayed only if you enable the Override Public Traffic option. Select VMware vNetwork Distributed Virtual Switch. If the vmware.use.dvswitch global parameter is true, the default option will be VMware vNetwork Distributed Virtual Switch.
Public Traffic vSwitch Name	Name of virtual switch to be used for the public traffic.
Override Guest Traffic	Enable the option to override the zone-wide guest traffic for the cluster you are creating.
Guest Traffic vSwitch Type	This option is displayed only if you enable the Override Guest Traffic option. Select VMware vNetwork Distributed Virtual Switch. If the vmware.use.dvswitch global parameter is true, the default option will be VMware vNetwork Distributed Virtual Switch.
Guest Traffic vSwitch Name	Name of virtual switch to be used for guest traffic.

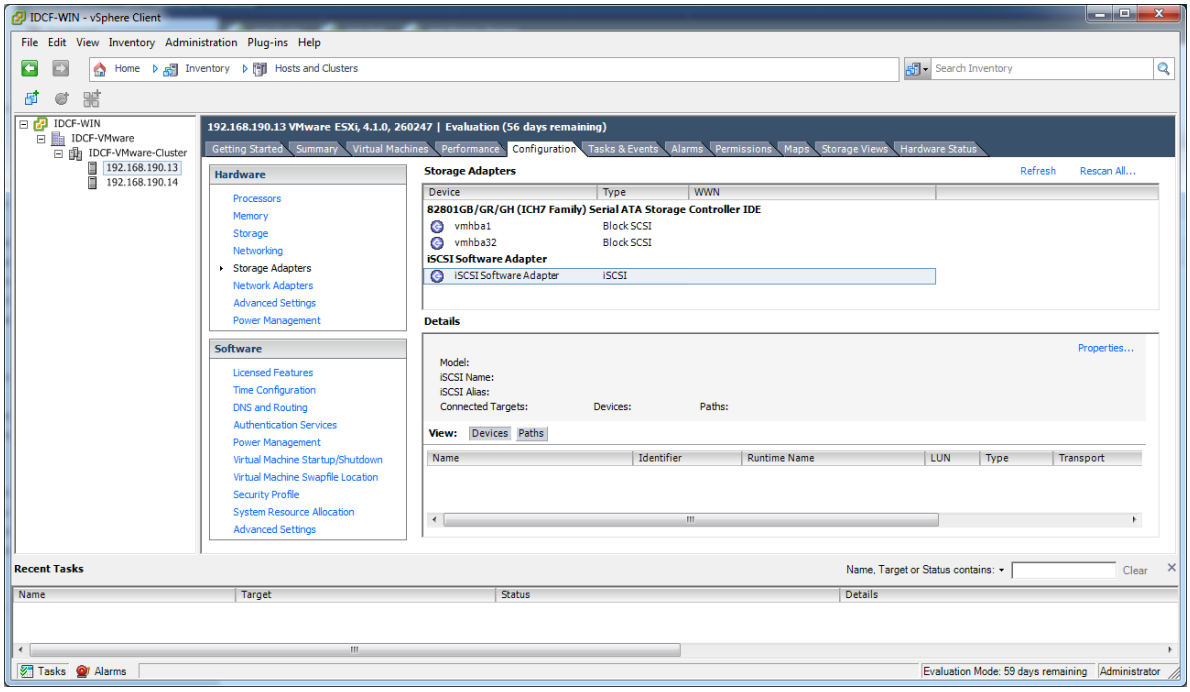
Storage Preparation for vSphere (iSCSI only)

Use of iSCSI requires preparatory work in vCenter. You must add an iSCSI target and create an iSCSI datastore.

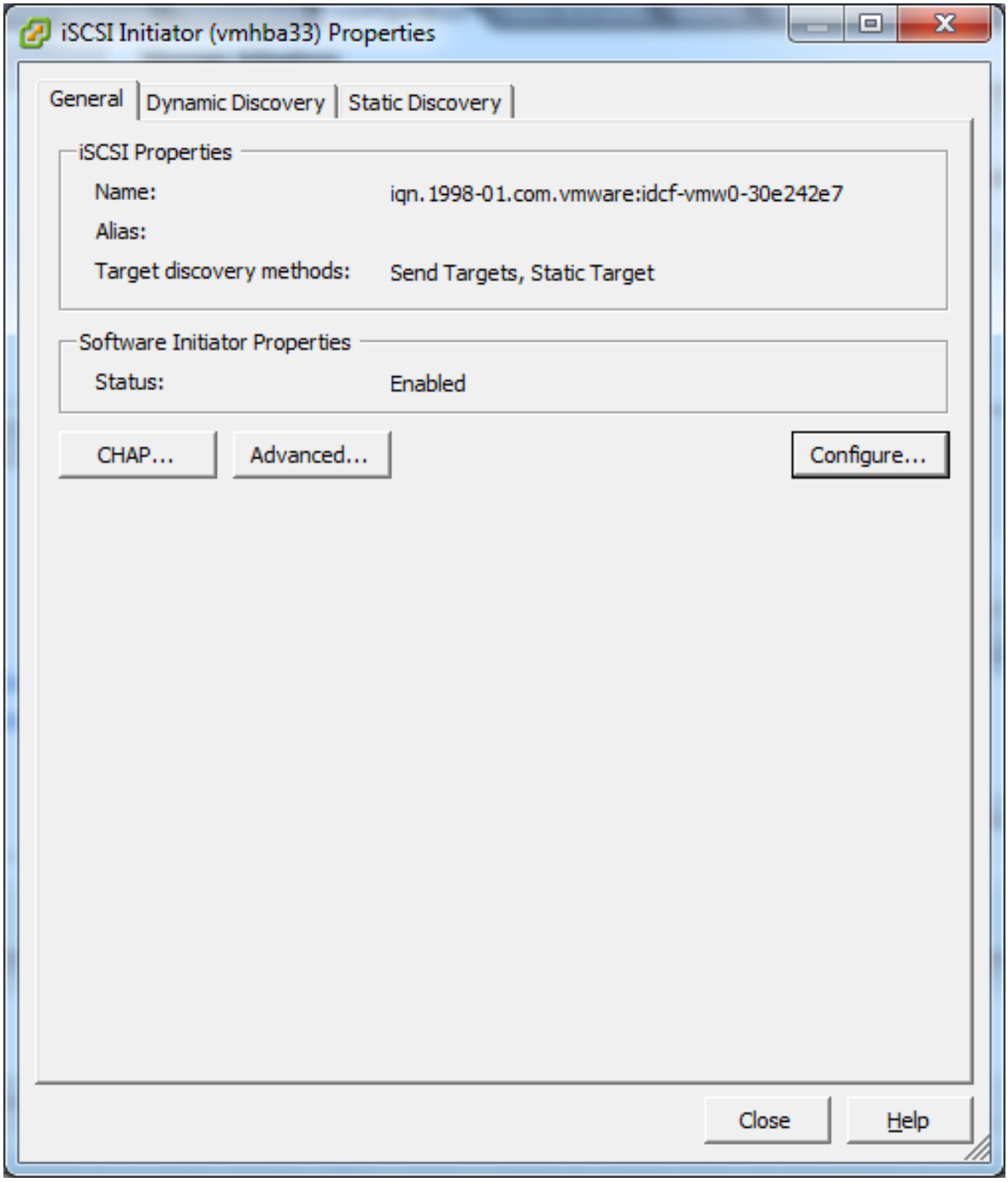
If you are using NFS, skip this section.

Enable iSCSI initiator for ESXi hosts

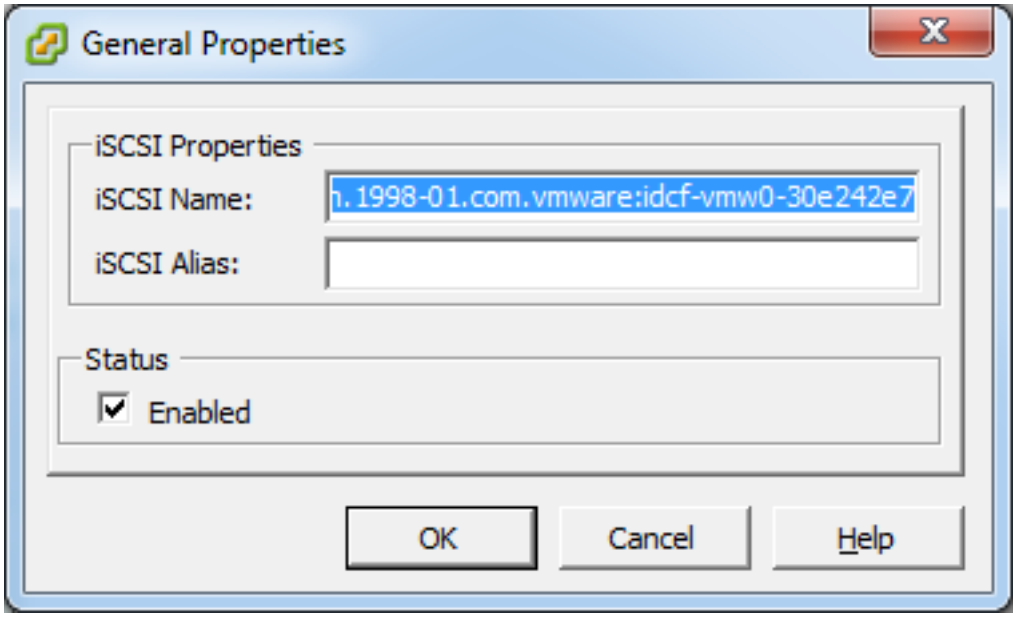
1. In vCenter, go to hosts and Clusters/Configuration, and click Storage Adapters link. You will see:



2. Select iSCSI software adapter and click Properties.



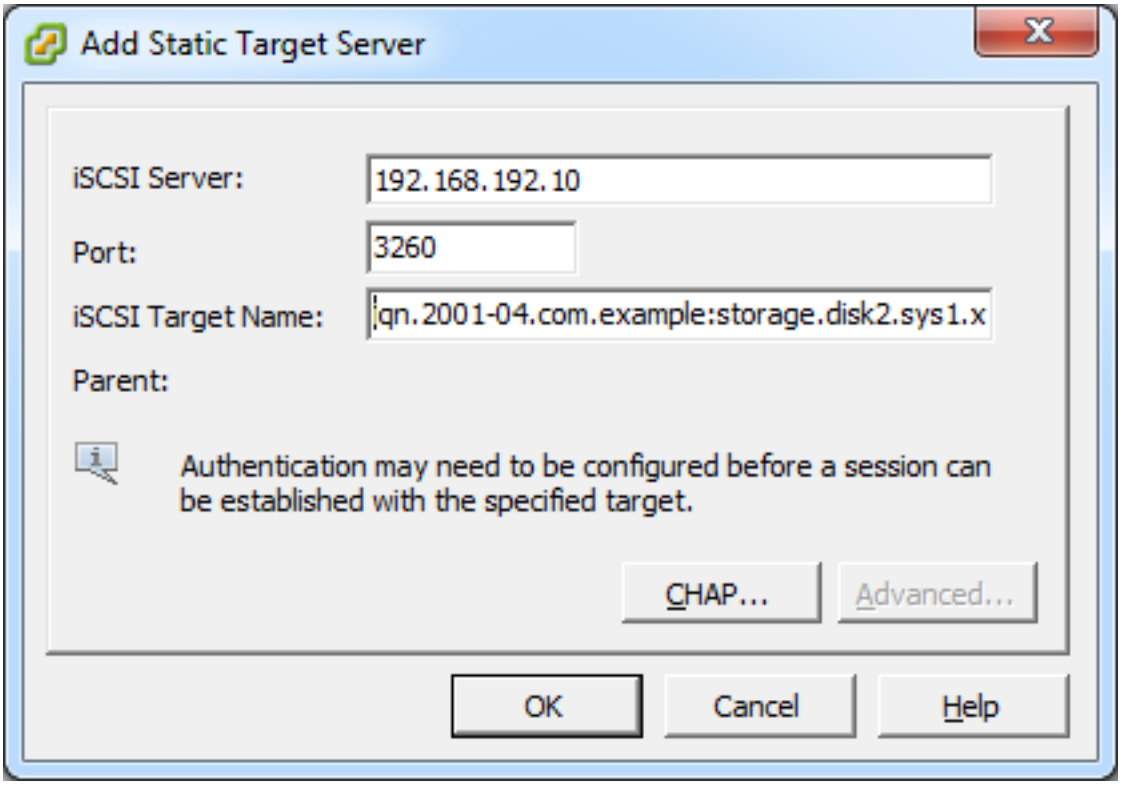
3. Click the Configure… button.



4. Check Enabled to enable the initiator.
5. Click OK to save.

Add iSCSI target

Under the properties dialog, add the iSCSI target info:



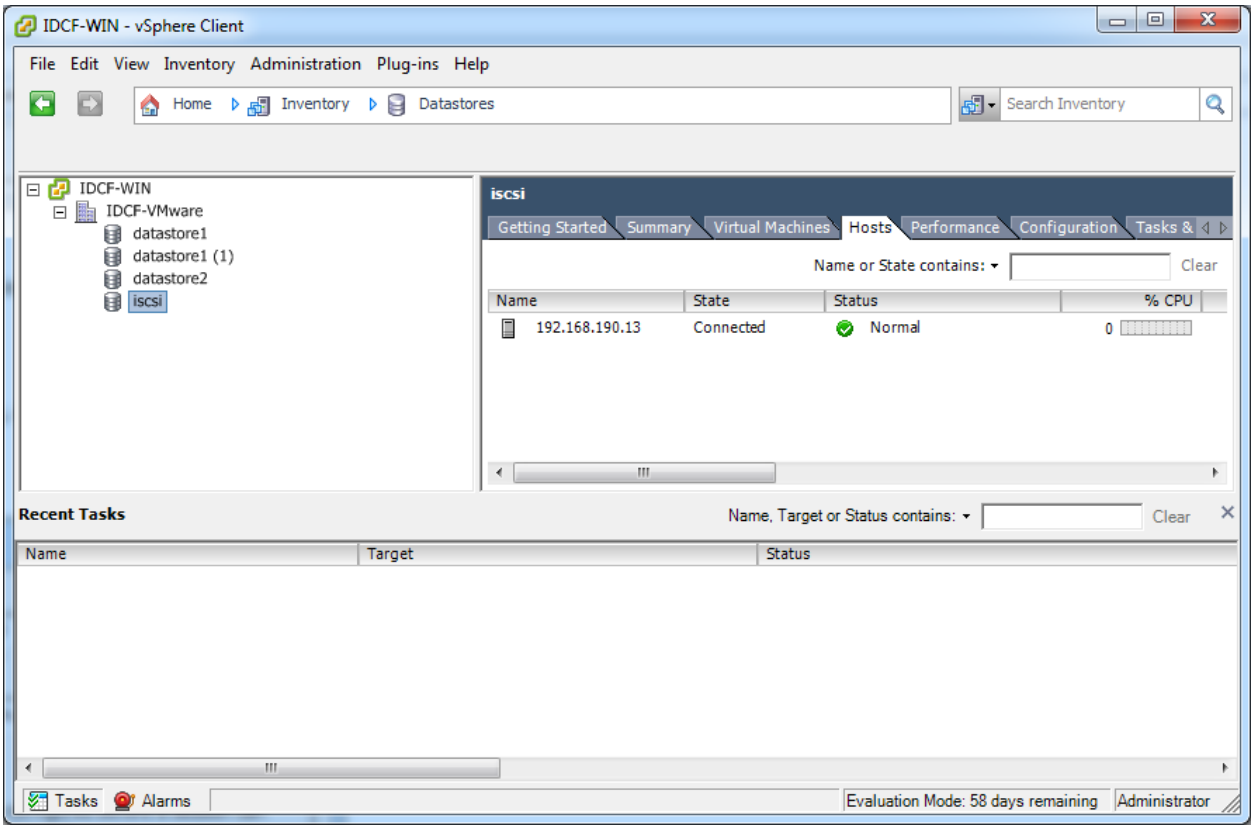
Repeat these steps for all ESXi hosts in the cluster.

Create an iSCSI datastore

You should now create a VMFS datastore. Follow these steps to do so:

1. Select Home/Inventory/Datastores.
2. Right click on the datacenter node.
3. Choose Add Datastore… command.
4. Follow the wizard to create a iSCSI datastore.

This procedure should be done on one host in the cluster. It is not necessary to do this on all hosts.



Multipathing for vSphere (Optional)

Storage multipathing on vSphere nodes may be done according to the vSphere installation guide.

Add Hosts or Configure Clusters (vSphere)

Use vCenter to create a vCenter cluster and add your desired hosts to the cluster. You will later add the entire cluster to CloudStack. (see "Add Cluster: vSphere").

Applying Hotfixes to a VMware vSphere Host

1. Disconnect the VMware vSphere cluster from CloudStack. It should remain disconnected long enough to apply the hotfix on the host.

(a) Log in to the CloudStack UI as root.

See *Log In to the UI*

(b) Navigate to the VMware cluster, click Actions, and select Unmanage.

(c) Watch the cluster status until it shows Unmanaged.

2. Perform the following on each of the ESXi hosts in the cluster:

(a) Move each of the ESXi hosts in the cluster to maintenance mode.

(b) Ensure that all the VMs are migrated to other hosts in that cluster.

(c) If there is only one host in that cluster, shutdown all the VMs and move the host into maintenance mode.

(d) Apply the patch on the ESXi host.

(e) Restart the host if prompted.

(f) Cancel the maintenance mode on the host.

3. Reconnect the cluster to CloudStack:

(a) Log in to the CloudStack UI as root.

(b) Navigate to the VMware cluster, click Actions, and select Manage.

(c) Watch the status to see that all the hosts come up. It might take several minutes for the hosts to come up.

Alternatively, verify the host state is properly synchronized and updated in the CloudStack database.

3.4.5 Host Citrix XenServer Installation

If you want to use the Citrix XenServer hypervisor to run guest virtual machines, install XenServer 6.0 or XenServer 6.0.2 on the host(s) in your cloud. For an initial installation, follow the steps below. If you have previously installed XenServer and want to upgrade to another version, see *Upgrading XenServer Versions*.

System Requirements for XenServer Hosts

- The host must be certified as compatible with one of the following. See the Citrix Hardware Compatibility Guide: http://hcl.xensource.com
  - XenServer 5.6 SP2
  - XenServer 6.0
  - XenServer 6.0.2
  - XenServer 6.1.0
  - XenServer 6.2.0
  - XenServer 6.5.0
- You must re-install Citrix XenServer if you are going to re-use a host from a previous install.
- Must support HVM (Intel-VT or AMD-V enabled)
- Be sure all the hotfixes provided by the hypervisor vendor are applied. Track the release of hypervisor patches through your hypervisor vendor's support channel, and apply patches as soon as possible after they are released. CloudStack will not track or notify you of required hypervisor patches. It is essential that your hosts are completely up to date with the provided hypervisor patches. The hypervisor vendor is likely to refuse to support any system that is not up to date with patches.

- All hosts within a cluster must be homogeneous. The CPUs must be of the same type, count, and feature flags.
- Must support HVM (Intel-VT or AMD-V enabled in BIOS)
- 64-bit x86 CPU (more cores results in better performance)
- Hardware virtualization support required
- 4 GB of memory
- 36 GB of local disk
- At least 1 NIC
- Statically allocated IP Address
- When you deploy CloudStack, the hypervisor host must not have any VMs already running

Warning: The lack of up-to-date hotfixes can lead to data corruption and lost VMs.

XenServer Installation Steps

1. From https://www.citrix.com/English/ss/downloads/, download the appropriate version of XenServer for your CloudStack version (see *"System Requirements for XenServer Hosts"*). Install it using the Citrix XenServer Installation Guide.

   Older Versions of XenServer:

   Note that you can download the most recent release of XenServer without having a Citrix account. If you wish to download older versions, you will need to create an account and look through the download archives.

Configure XenServer dom0 Memory

Configure the XenServer dom0 settings to allocate more memory to dom0. This can enable XenServer to handle larger numbers of virtual machines. We recommend 2940 MB of RAM for XenServer dom0. For instructions on how to do this, see http://support.citrix.com/article/CTX126531. The article refers to XenServer 5.6, but the same information applies to XenServer 6.0.

Username and Password

All XenServers in a cluster must have the same username and password as configured in CloudStack.

Time Synchronization

The host must be set to use NTP. All hosts in a pod must have the same time.

1. Install NTP.

   ```
   # yum install ntp
   ```

2. Edit the NTP configuration file to point to your NTP server.

   ```
   # vi /etc/ntp.conf
   ```

   Add one or more server lines in this file with the names of the NTP servers you want to use. For example:

```
server 0.xenserver.pool.ntp.org
server 1.xenserver.pool.ntp.org
server 2.xenserver.pool.ntp.org
server 3.xenserver.pool.ntp.org
```

3. Restart the NTP client.

```
# service ntpd restart
```

4. Make sure NTP will start again upon reboot.

```
# chkconfig ntpd on
```

Install CloudStack XenServer Support Package (CSP)

(Optional)

To enable security groups, elastic load balancing, and elastic IP on XenServer, download and install the CloudStack XenServer Support Package (CSP). After installing XenServer, perform the following additional steps on each XenServer host.

For XenServer 6.1:

CSP functionality is already present in XenServer 6.1

1. Run the below command

```
xe-switch-network-backend bridge
```

2. update sysctl.conf with the following

```
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-arptables = 1

$ sysctl -p /etc/sysctl.conf
```

For XenServer 6.0.2, 6.0, 5.6 SP2:

1. Download the CSP software onto the XenServer host from one of the following links:

   For XenServer 6.0.2:

   http://download.cloud.com/releases/3.0.1/XS-6.0.2/xenserver-cloud-supp.tgz

   For XenServer 5.6 SP2:

   http://download.cloud.com/releases/2.2.0/xenserver-cloud-supp.tgz

   For XenServer 6.0:

   http://download.cloud.com/releases/3.0/xenserver-cloud-supp.tgz

2. Extract the file:

```
# tar xf xenserver-cloud-supp.tgz
```

3. Run the following script:

```
# xe-install-supplemental-pack xenserver-cloud-supp.iso
```

4. If the XenServer host is part of a zone that uses basic networking, disable Open vSwitch (OVS):

```
# xe-switch-network-backend  bridge
```

   Restart the host machine when prompted.

The XenServer host is now ready to be added to CloudStack.

Primary Storage Setup for XenServer

CloudStack natively supports NFS, iSCSI and local storage. If you are using one of these storage types, there is no need to create the XenServer Storage Repository ("SR").

If, however, you would like to use storage connected via some other technology, such as FiberChannel, you must set up the SR yourself. To do so, perform the following steps. If you have your hosts in a XenServer pool, perform the steps on the master node. If you are working with a single XenServer which is not part of a cluster, perform the steps on that XenServer.

1. Connect FiberChannel cable to all hosts in the cluster and to the FiberChannel storage host.
2. Rescan the SCSI bus. Either use the following command or use XenCenter to perform an HBA rescan.

```
# scsi-rescan
```

3. Repeat step 2 on every host.
4. Check to be sure you see the new SCSI disk.

```
# ls /dev/disk/by-id/scsi-360a98000503365344e6f6177615a516b -l
```

   The output should look like this, although the specific file name will be different (scsi-<scsiID>):

```
lrwxrwxrwx 1 root root 9 Mar 16 13:47
/dev/disk/by-id/scsi-360a98000503365344e6f6177615a516b -> ../../sdc
```

5. Repeat step 4 on every host.
6. On the storage server, run this command to get a unique ID for the new SR.

```
# uuidgen
```

   The output should look like this, although the specific ID will be different:

```
e6849e96-86c3-4f2c-8fcc-350cc711be3d
```

7. Create the FiberChannel SR. In name-label, use the unique ID you just generated.

```
# xe sr-create type=lvmohba shared=true
device-config:SCSIid=360a98000503365344e6f6177615a516b
name-label="e6849e96-86c3-4f2c-8fcc-350cc711be3d"
```

   This command returns a unique ID for the SR, like the following example (your ID will be different):

```
7a143820-e893-6c6a-236e-472da6ee66bf
```

8. To create a human-readable description for the SR, use the following command. In uuid, use the SR ID returned by the previous command. In name-description, set whatever friendly text you prefer.

```
# xe sr-param-set uuid=7a143820-e893-6c6a-236e-472da6ee66bf name-description=
↪"Fiber Channel storage repository"
```

   Make note of the values you will need when you add this storage to CloudStack later (see "Add Primary Storage"). In the Add Primary Storage dialog, in Protocol, you will choose PreSetup. In SR Name-Label, you will enter the name-label you set earlier (in this example, e6849e96-86c3-4f2c-8fcc-350cc711be3d).

9. (Optional) If you want to enable multipath I/O on a FiberChannel SAN, refer to the documentation provided by the SAN vendor.

iSCSI Multipath Setup for XenServer (Optional)

When setting up the storage repository on a Citrix XenServer, you can enable multipath I/O, which uses redundant physical components to provide greater reliability in the connection between the server and the SAN. To enable multipathing, use a SAN solution that is supported for Citrix servers and follow the procedures in Citrix documentation. The following links provide a starting point:

- http://support.citrix.com/article/CTX118791
- http://support.citrix.com/article/CTX125403

You can also ask your SAN vendor for advice about setting up your Citrix repository for multipathing.

Make note of the values you will need when you add this storage to the CloudStack later (see "Add Primary Storage"). In the Add Primary Storage dialog, in Protocol, you will choose PreSetup. In SR Name-Label, you will enter the same name used to create the SR.

If you encounter difficulty, address the support team for the SAN provided by your vendor. If they are not able to solve your issue, see Contacting Support.

Physical Networking Setup for XenServer

Once XenServer has been installed, you may need to do some additional network configuration. At this point in the installation, you should have a plan for what NICs the host will have and what traffic each NIC will carry. The NICs should be cabled as necessary to implement your plan.

If you plan on using NIC bonding, the NICs on all hosts in the cluster must be cabled exactly the same. For example, if eth0 is in the private bond on one host in a cluster, then eth0 must be in the private bond on all hosts in the cluster.

The IP address assigned for the management network interface must be static. It can be set on the host itself or obtained via static DHCP.

CloudStack configures network traffic of various types to use different NICs or bonds on the XenServer host. You can control this process and provide input to the Management Server through the use of XenServer network name labels. The name labels are placed on physical interfaces or bonds and configured in CloudStack. In some simple cases the name labels are not required.

When configuring networks in a XenServer environment, network traffic labels must be properly configured to ensure that the virtual interfaces are created by CloudStack are bound to the correct physical device. The name-label of the XenServer network must match the XenServer traffic label specified while creating the CloudStack network. This is set by running the following command:

```
xe network-param-set uuid=<network id> name-label=<CloudStack traffic label>
```

Configuring Public Network with a Dedicated NIC for XenServer (Optional)

CloudStack supports the use of a second NIC (or bonded pair of NICs, described in *NIC Bonding for XenServer (Optional)*) for the public network. If bonding is not used, the public network can be on any NIC and can be on different NICs on the hosts in a cluster. For example, the public network can be on eth0 on node A and eth1 on node B. However, the XenServer name-label for the public network must be identical across all hosts. The following examples set the network label to "cloud-public". After the management server is installed and running you must configure it with the name of the chosen network label (e.g. "cloud-public"); this is discussed in "Management Server Installation".

If you are using two NICs bonded together to create a public network, see *NIC Bonding for XenServer (Optional)*.

If you are using a single dedicated NIC to provide public network access, follow this procedure on each new host that is added to CloudStack before adding the host.

1. Run xe network-list and find the public network. This is usually attached to the NIC that is public. Once you find the network make note of its UUID. Call this <UUID-Public>.
2. Run the following command.

```
# xe network-param-set name-label=cloud-public uuid=<UUID-Public>
```

Configuring Multiple Guest Networks for XenServer (Optional)

CloudStack supports the use of multiple guest networks with the XenServer hypervisor. Each network is assigned a name-label in XenServer. For example, you might have two networks with the labels "cloud-guest" and "cloud-guest2". After the management server is installed and running, you must add the networks and use these labels so that CloudStack is aware of the networks.

Follow this procedure on each new host before adding the host to CloudStack:

1. Run xe network-list and find one of the guest networks. Once you find the network make note of its UUID. Call this <UUID-Guest>.
2. Run the following command, substituting your own name-label and uuid values.

```
# xe network-param-set name-label=<cloud-guestN> uuid=<UUID-Guest>
```

3. Repeat these steps for each additional guest network, using a different name-label and uuid each time.

Separate Storage Network for XenServer (Optional)

You can optionally set up a separate storage network. This should be done first on the host, before implementing the bonding steps below. This can be done using one or two available NICs. With two NICs bonding may be done as above. It is the administrator's responsibility to set up a separate storage network.

Give the storage network a different name-label than what will be given for other networks.

For the separate storage network to work correctly, it must be the only interface that can ping the primary storage device's IP address. For example, if eth0 is the management network NIC, ping -I eth0 <primary storage device IP> must fail. In all deployments, secondary storage devices must be pingable from the management network NIC or bond. If a secondary storage device has been placed on the storage network, it must also be pingable via the storage network NIC or bond on the hosts as well.

You can set up two separate storage networks as well. For example, if you intend to implement iSCSI multipath, dedicate two non-bonded NICs to multipath. Each of the two networks needs a unique name-label.

If no bonding is done, the administrator must set up and name-label the separate storage network on all hosts (masters and slaves).

Here is an example to set up eth5 to access a storage network on 172.16.0.0/24.

```
# xe pif-list host-name-label='hostname' device=eth5
uuid(RO): ab0d3dd4-5744-8fae-9693-a022c7a3471d
device ( RO): eth5
#xe pif-reconfigure-ip DNS=172.16.3.3 gateway=172.16.0.1 IP=172.16.0.55 mode=static␣
↪netmask=255.255.255.0 uuid=ab0d3dd4-5744-8fae-9693-a022c7a3471d
```

NIC Bonding for XenServer (Optional)

XenServer supports Source Level Balancing (SLB) NIC bonding. Two NICs can be bonded together to carry public, private, and guest traffic, or some combination of these. Separate storage networks are also possible. Here are some example supported configurations:

- 2 NICs on private, 2 NICs on public, 2 NICs on storage
- 2 NICs on private, 1 NIC on public, storage uses management network
- 2 NICs on private, 2 NICs on public, storage uses management network
- 1 NIC for private, public, and storage

All NIC bonding is optional.

XenServer expects all nodes in a cluster will have the same network cabling and same bonds implemented. In an installation the master will be the first host that was added to the cluster and the slave hosts will be all subsequent hosts added to the cluster. The bonds present on the master set the expectation for hosts added to the cluster later. The procedure to set up bonds on the master and slaves are different, and are described below. There are several important implications of this:

- You must set bonds on the first host added to a cluster. Then you must use xe commands as below to establish the same bonds in the second and subsequent hosts added to a cluster.
- Slave hosts in a cluster must be cabled exactly the same as the master. For example, if eth0 is in the private bond on the master, it must be in the management network for added slave hosts.

Management Network Bonding

The administrator must bond the management network NICs prior to adding the host to CloudStack.

Creating a Private Bond on the First Host in the Cluster

Use the following steps to create a bond in XenServer. These steps should be run on only the first host in a cluster. This example creates the cloud-private network with two physical NICs (eth0 and eth1) bonded into it.

1. Find the physical NICs that you want to bond together.

   ```
   # xe pif-list host-name-label='hostname' device=eth0
   # xe pif-list host-name-label='hostname' device=eth1
   ```

   These command shows the eth0 and eth1 NICs and their UUIDs. Substitute the ethX devices of your choice. Call the UUID's returned by the above command slave1-UUID and slave2-UUID.

2. Create a new network for the bond. For example, a new network with name "cloud-private".

   This label is important. CloudStack looks for a network by a name you configure. You must use the same name-label for all hosts in the cloud for the management network.

```
# xe network-create name-label=cloud-private
# xe bond-create network-uuid=[uuid of cloud-private created above]
pif-uuids=[slave1-uuid],[slave2-uuid]
```

Now you have a bonded pair that can be recognized by CloudStack as the management network.

Public Network Bonding

Bonding can be implemented on a separate, public network. The administrator is responsible for creating a bond for the public network if that network will be bonded and will be separate from the management network.

Creating a Public Bond on the First Host in the Cluster

These steps should be run on only the first host in a cluster. This example creates the cloud-public network with two physical NICs (eth2 and eth3) bonded into it.

1. Find the physical NICs that you want to bond together.

```
# xe pif-list host-name-label='hostname' device=eth2
# xe pif-list host-name-label='hostname' device=eth3
```

   These command shows the eth2 and eth3 NICs and their UUIDs. Substitute the ethX devices of your choice. Call the UUID's returned by the above command slave1-UUID and slave2-UUID.

2. Create a new network for the bond. For example, a new network with name "cloud-public".

   This label is important. CloudStack looks for a network by a name you configure. You must use the same name-label for all hosts in the cloud for the public network.

```
# xe network-create name-label=cloud-public
# xe bond-create network-uuid=[uuid of cloud-public created above]
pif-uuids=[slave1-uuid],[slave2-uuid]
```

Now you have a bonded pair that can be recognized by CloudStack as the public network.

Adding More Hosts to the Cluster

With the bonds (if any) established on the master, you should add additional, slave hosts. Run the following command for all additional hosts to be added to the cluster. This will cause the host to join the master in a single XenServer pool.

```
# xe pool-join master-address=[master IP] master-username=root
master-password=[your password]
```

Complete the Bonding Setup Across the Cluster

With all hosts added to the pool, run the cloud-setup-bond script. This script will complete the configuration and set up of the bonds across all hosts in the cluster.

1. Copy the script from the Management Server in /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver/cloud-setup-bonding.sh to the master host and ensure it is executable.
2. Run the script:

```
# ./cloud-setup-bonding.sh
```

Now the bonds are set up and configured properly across the cluster.

Upgrading XenServer Versions

This section tells how to upgrade XenServer software on CloudStack hosts. The actual upgrade is described in XenServer documentation, but there are some additional steps you must perform before and after the upgrade.

Note: Be sure the hardware is certified compatible with the new version of XenServer.

To upgrade XenServer:

1. Upgrade the database. On the Management Server node:
   (a) Back up the database:

```
# mysqldump --user=root --databases cloud > cloud.backup.sql
# mysqldump --user=root --databases cloud_usage > cloud_usage.backup.sql
```

   (b) You might need to change the OS type settings for VMs running on the upgraded hosts.
      - If you upgraded from XenServer 5.6 GA to XenServer 5.6 SP2, change any VMs that have the OS type CentOS 5.5 (32-bit), Oracle Enterprise Linux 5.5 (32-bit), or Red Hat Enterprise Linux 5.5 (32-bit) to Other Linux (32-bit). Change any VMs that have the 64-bit versions of these same OS types to Other Linux (64-bit).
      - If you upgraded from XenServer 5.6 SP2 to XenServer 6.0.2, change any VMs that have the OS type CentOS 5.6 (32-bit), CentOS 5.7 (32-bit), Oracle Enterprise Linux 5.6 (32-bit), Oracle Enterprise Linux 5.7 (32-bit), Red Hat Enterprise Linux 5.6 (32-bit) , or Red Hat Enterprise Linux 5.7 (32-bit) to Other Linux (32-bit). Change any VMs that have the 64-bit versions of these same OS types to Other Linux (64-bit).
      - If you upgraded from XenServer 5.6 to XenServer 6.0.2, do all of the above.
   (c) Restart the Management Server and Usage Server. You only need to do this once for all clusters.

```
# service cloudstack-management start
# service cloudstack-usage start
```

2. Disconnect the XenServer cluster from CloudStack.
   (a) Log in to the CloudStack UI as root.
   (b) Navigate to the XenServer cluster, and click Actions – Unmanage.
   (c) Watch the cluster status until it shows Unmanaged.
3. Log in to one of the hosts in the cluster, and run this command to clean up the VLAN:

```
# . /opt/xensource/bin/cloud-clean-vlan.sh
```

4. Still logged in to the host, run the upgrade preparation script:

```
# /opt/xensource/bin/cloud-prepare-upgrade.sh
```

Troubleshooting: If you see the error "can't eject CD," log in to the VM and umount the CD, then run the script again.

5. Upgrade the XenServer software on all hosts in the cluster. Upgrade the master first.

   (a) Live migrate all VMs on this host to other hosts. See the instructions for live migration in the Administrator's Guide.

   Troubleshooting: You might see the following error when you migrate a VM:

```
[root@xenserver-qa-2-49-4 ~]# xe vm-migrate live=true host=xenserver-qa-2-49-
↪5 vm=i-2-8-VM
You attempted an operation on a VM which requires PV drivers to be installed␣
↪but the drivers were not detected.
vm: b6cf79c8-02ee-050b-922f-49583d9f1a14 (i-2-8-VM)
```

   To solve this issue, run the following:

```
# /opt/xensource/bin/make_migratable.sh  b6cf79c8-02ee-050b-922f-49583d9f1a14
```

   (b) Reboot the host.

   (c) Upgrade to the newer version of XenServer. Use the steps in XenServer documentation.

   (d) After the upgrade is complete, copy the following files from the management server to this host, in the directory locations shown below:

Copy this Management Server file	To this location on the XenServer host
/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver/xenserver60/NFSSR.py	/opt/xensource/sm/NFSSR.py
/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver/setupxenserver.sh	/opt/xensource/bin/setupxenserver.sh
/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver/make_migratable.sh	/opt/xensource/bin/make_migratable.sh
/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver/cloud-clean-vlan.sh	/opt/xensource/bin/cloud-clean-vlan.sh

   (e) Run the following script:

```
# /opt/xensource/bin/setupxenserver.sh
```

   Troubleshooting: If you see the following error message, you can safely ignore it.

```
mv: cannot stat /etc/cron.daily/logrotate: No such file or directory
```

   (f) Plug in the storage repositories (physical block devices) to the XenServer host:

```
# for pbd in xe pbd-list currently-attached=false| grep ^uuid | awk '
↪{print $NF}'; do xe pbd-plug uuid=$pbd ; done
```

   Note: If you add a host to this XenServer pool, you need to migrate all VMs on this host to other hosts, and eject this host from XenServer pool.

6. Repeat these steps to upgrade every host in the cluster to the same version of XenServer.

7. Run the following command on one host in the XenServer cluster to clean up the host tags:

```
# for host in $(xe host-list | grep ^uuid | awk '{print $NF}') ; do xe host-param-
↪clear uuid=$host param-name=tags; done;
```

Note: When copying and pasting a command, be sure the command has pasted as a single line before executing. Some document viewers may introduce unwanted line breaks in copied text.

8. Reconnect the XenServer cluster to CloudStack.
   (a) Log in to the CloudStack UI as root.
   (b) Navigate to the XenServer cluster, and click Actions – Manage.
   (c) Watch the status to see that all the hosts come up.
9. After all hosts are up, run the following on one host in the cluster:

```
# /opt/xensource/bin/cloud-clean-vlan.sh
```

3.5 Optional Installation

3.5.1 Additional Installation Options

The next few sections describe CloudStack features above and beyond the basic deployment options.

Installing the Usage Server (Optional)

You can optionally install the Usage Server once the Management Server is configured properly. The Usage Server takes data from the events in the system and enables usage-based billing for accounts.

When multiple Management Servers are present, the Usage Server may be installed on any number of them. The Usage Servers will coordinate usage processing. A site that is concerned about availability should install Usage Servers on at least two Management Servers.

Requirements for Installing the Usage Server

- The Management Server must be running when the Usage Server is installed.
- The Usage Server must be installed on the same server as a Management Server.

Steps to Install the Usage Server

1. Package repository should already being configured. Refer to Configure Package Repository
2. Install package cloudstack-usage

   On RHEL/CentOS systems, use:

```
# yum install cloudstack-usage
```

   On Debian/Ubuntu systems, use:

```
# apt-get install cloudstack-usage
```

3. Once installed, start the Usage Server with the following command.

```
# service cloudstack-usage start
```

4. Enable the service at boot

   On RHEL/CentOS systems, use:

```
# chkconfig cloudstack-usage on
```

   On Debian/Ubuntu systems, use:

```
# update-rc.d cloudstack-usage defaults
```

Working with Usage discusses further configuration of the Usage Server.

SSL (Optional)

CloudStack provides HTTP access in its default installation. There are a number of technologies and sites which choose to implement SSL. As a result, we have left CloudStack to expose HTTP under the assumption that a site will implement its typical practice.

CloudStack uses Tomcat as its servlet container. For sites that would like CloudStack to terminate the SSL session, Tomcat's SSL access may be enabled. Tomcat SSL configuration is described at http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html.

Database Replication (Optional)

CloudStack supports database replication from one MySQL node to another. This is achieved using standard MySQL replication. You may want to do this as insurance against MySQL server or storage loss. MySQL replication is implemented using a master/slave model. The master is the node that the Management Servers are configured to use. The slave is a standby node that receives all write operations from the master and applies them to a local, redundant copy of the database. The following steps are a guide to implementing MySQL replication.

Note: Creating a replica is not a backup solution. You should develop a backup procedure for the MySQL data that is distinct from replication.

1. Ensure that this is a fresh install with no data in the master.
2. Edit my.cnf on the master and add the following in the [mysqld] section below datadir.

```
log_bin=mysql-bin
server_id=1
```

   The server_id must be unique with respect to other servers. The recommended way to achieve this is to give the master an ID of 1 and each slave a sequential number greater than 1, so that the servers are numbered 1, 2, 3, etc.

3. Restart the MySQL service. On RHEL/CentOS systems, use:

```
# service mysqld restart
```

   On Debian/Ubuntu systems, use:

```
# service mysql restart
```

4. Create a replication account on the master and give it privileges. We will use the "cloud-repl" user with the password "password". This assumes that master and slave run on the 172.16.1.0/24 network.
5. Leave the current MySQL session running.
6. In a new shell start a second MySQL session.
7. Retrieve the current position of the database.

```
# mysql -u root
mysql> show master status;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |      412 |              |                  |
+------------------+----------+--------------+------------------+
```

8. Note the file and the position that are returned by your instance.
9. Exit from this session.
10. Complete the master setup. Returning to your first session on the master, release the locks and exit MySQL.

```
mysql> unlock tables;
```

11. Install and configure the slave. On the slave server, run the following commands.

```
# yum install mysql-server
# chkconfig mysqld on
```

12. Edit my.cnf and add the following lines in the [mysqld] section below datadir.

```
server_id=2
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
```

13. Restart MySQL. Use "mysqld" on RHEL/CentOS systems:

```
# service mysqld restart
```

On Ubuntu/Debian systems use "mysql."

```
# service mysql restart
```

14. Instruct the slave to connect to and replicate from the master. Replace the IP address, password, log file, and position with the values you have used in the previous steps.

```
mysql> change master to
    -> master_host='172.16.1.217',
    -> master_user='cloud-repl',
    -> master_password='password',
    -> master_log_file='mysql-bin.000001',
    -> master_log_pos=412;
```

15. Then start replication on the slave.

```
mysql> start slave;
```

16. Optionally, open port 3306 on the slave as was done on the master earlier.

    This is not required for replication to work. But if you choose not to do this, you will need to do it when failover to the replica occurs.

Failover

This will provide for a replicated database that can be used to implement manual failover for the Management Servers. CloudStack failover from one MySQL instance to another is performed by the administrator. In the event of a database failure you should:

1. Stop the Management Servers (via service cloudstack-management stop).
2. Change the replica's configuration to be a master and restart it.
3. Ensure that the replica's port 3306 is open to the Management Servers.
4. Make a change so that the Management Server uses the new database. The simplest process here is to put the IP address of the new database server into each Management Server's /etc/cloudstack/management/db.properties.
5. Restart the Management Servers:

```
# service cloudstack-management start
```

Amazon Web Services Interface

Amazon Web Services Compatible Interface

CloudStack can translate Amazon Web Services (AWS) API calls to native CloudStack API calls so that users can continue using existing AWS-compatible tools. This translation service runs as a separate web application in the same tomcat server as the management server of CloudStack, listening on a different port. The Amazon Web Services (AWS) compatible interface provides the EC2 SOAP and Query APIs as well as the S3 REST API.

Note: This service was previously enabled by separate software called CloudBridge. It is now fully integrated with the CloudStack management server.

Warning: The compatible interface for the EC2 Query API and the S3 API are Work In Progress. The S3 compatible API offers a way to store data on the management server file system, it is not an implementation of the S3 backend.

Limitations

- Supported only in zones that use basic networking.
- Available in fresh installations of CloudStack. Not available through upgrade of previous versions.
- Features such as Elastic IP (EIP) and Elastic Load Balancing (ELB) are only available in an infrastructure with a Citrix NetScaler device. Users accessing a Zone with a NetScaler device will need to use a NetScaler-enabled network offering (DefaultSharedNetscalerEIP and ELBNetworkOffering).

Supported API Version

- The EC2 interface complies with Amazon's WDSL version dated November 15, 2010, available at http://ec2.amazonaws.com/doc/2010-11-15/.
- The interface is compatible with the EC2 command-line tools *EC2 tools v. 1.3.6230*, which can be downloaded at http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-62308.zip.

Note: Work is underway to support a more recent version of the EC2 API

Enabling the EC2 and S3 Compatible Interface

The software that provides AWS API compatibility is installed along with CloudStack. You must enable the services and perform some setup steps prior to using it.

1. Set the global configuration parameters for each service to true. See *Setting Global Configuration Parameters*.
2. Create a set of CloudStack service offerings with names that match the Amazon service offerings. You can do this through the CloudStack UI as described in the Administration Guide.

   Warning: Be sure you have included the Amazon default service offering, m1.small. As well as any EC2 instance types that you will use.

3. If you did not already do so when you set the configuration parameter in step 1, restart the Management Server.

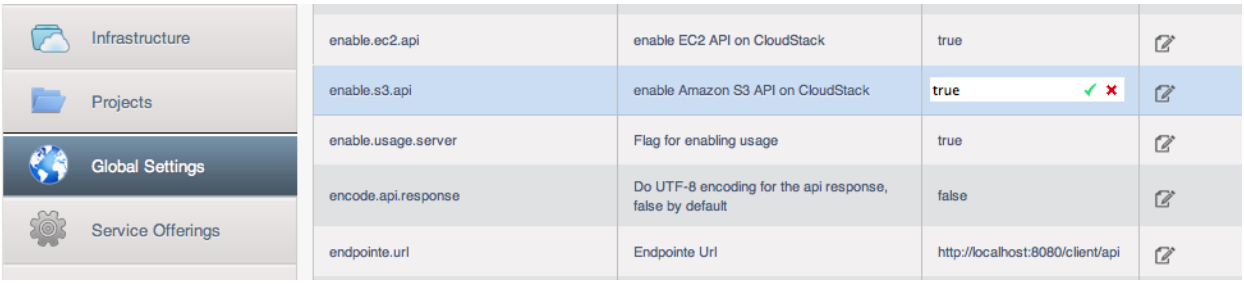
```
# service cloudstack-management restart
```

The following sections provides details to perform these steps

Enabling the Services

To enable the EC2 and S3 compatible services you need to set the configuration variables *enable.ec2.api* and *enable.s3.api* to true. You do not have to enable both at the same time. Enable the ones you need. This can be done via the CloudStack GUI by going in *Global Settings* or via the API.

The snapshot below shows you how to use the GUI to enable these services



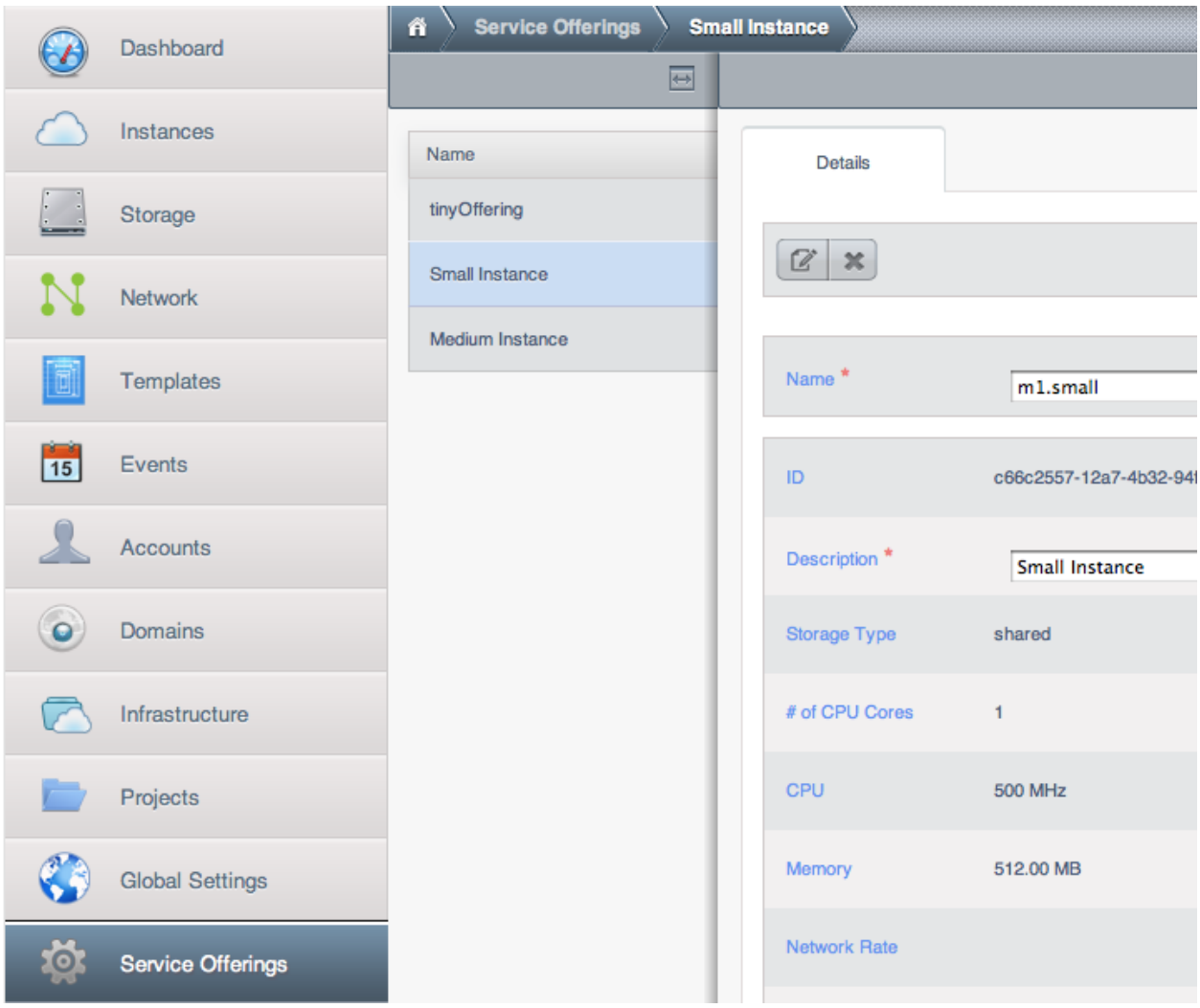
Using the CloudStack API, the easiest is to use the so-called integration port on which you can make unauthenticated calls. In Global Settings set the port to 8096 and subsequently call the *updateConfiguration* method. The following urls shows you how:

```
http://localhost:8096/client/api?command=updateConfiguration&name=enable.ec2.api&
↪value=true
http://localhost:8096/client/api?command=updateConfiguration&name=enable.ec2.api&
↪value=true
```

Once you have enabled the services, restart the server.

Creating EC2 Compatible Service Offerings

You will also need to define compute service offerings with names compatible with the Amazon EC2 instance types API names (e.g m1.small,m1.large). This can be done via the CloudStack GUI. Go under *Service Offerings* select *Compute offering* and either create a new compute offering or modify an existing one, ensuring that the name matches an EC2 instance type API name. The snapshot below shows you how:



Modifying the AWS API Port

Note: (Optional) The AWS API listens for requests on port 7080. If you prefer AWS API to listen on another port, you can change it as follows:

1. Edit the files `/etc/cloudstack/management/server.xml`, `/etc/cloudstack/management/server-nonssl.xml`, and `/etc/cloudstack/management/server-ssl.xml`.
2. In each file, find the tag <Service name="Catalina7080">. Under this tag, locate <Connector executor="tomcatThreadPool-internal" port= . . . .<.
3. Change the port to whatever port you want to use, then save the files.
4. Restart the Management Server.

If you re-install CloudStack, you will have to re-enable the services and if need be update the port.

AWS API User Setup

In general, users need not be aware that they are using a translation service provided by CloudStack. They only need to send AWS API calls to CloudStack's endpoint, and it will translate the calls to the native CloudStack API. Users of the Amazon EC2 compatible interface will be able to keep their existing EC2 tools and scripts and use them with their CloudStack deployment, by specifying the endpoint of the management server and using the proper user credentials. In order to do this, each user must perform the following configuration steps:

- Generate user credentials.
- Register with the service.
- For convenience, set up environment variables for the EC2 SOAP command-line tools.

AWS API Command-Line Tools Setup

To use the EC2 command-line tools, the user must perform these steps:

1. Be sure you have the right version of EC2 Tools. The supported version is available at http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-62308.zip.
2. Set up the EC2 environment variables. This can be done every time you use the service or you can set them up in the proper shell profile. Replace the endpoint (i.e EC2_URL) with the proper address of your CloudStack management server and port. In a bash shell do the following.

```
$ export EC2_CERT=/path/to/cert.pem
$ export EC2_PRIVATE_KEY=/path/to/private_key.pem
$ export EC2_URL=http://localhost:7080/awsapi
$ export EC2_HOME=/path/to/EC2_tools_directory
```

Using Timeouts to Ensure AWS API Command Completion

The Amazon EC2 command-line tools have a default connection timeout. When used with CloudStack, a longer timeout might be needed for some commands. If you find that commands are not completing due to timeouts, you can specify a custom timeouts. You can add the following optional command-line parameters to any CloudStack-supported EC2 command:

Specifies a connection timeout (in seconds)

```
--connection-timeout TIMEOUT
```

Specifies a request timeout (in seconds)

```
--request-timeout TIMEOUT
```

Example:

```
ec2-run-instances 2 -z us-test1 -n 1-3 --connection-timeout 120 --request-timeout 120
```

Note: The timeouts optional arguments are not specific to CloudStack.

Supported AWS API Calls

The following Amazon EC2 commands are supported by CloudStack when the AWS API compatible interface is enabled. For a few commands, there are differences between the CloudStack and Amazon EC2 versions, and these differences are noted. The underlying SOAP call for each command is also given, for those who have built tools using those calls.

Table 1. Elastic IP API mapping

EC2 command	SOAP call	CloudStack API call
ec2-allocate-address	AllocateAddress	associateIpAddress
ec2-associate-address	AssociateAddress	enableStaticNat
ec2-describe-addresses	DescribeAddresses	listPublicIpAddresses
ec2-diassociate-address	DisassociateAddress	disableStaticNat
ec2-release-address	ReleaseAddress	disassociateIpAddress

Table 2. Availability Zone API mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-availability-zones	DescribeAvailabilityZones	listZones

Table 3. Images API mapping

EC2 command	SOAP call	CloudStack API call
ec2-create-image	CreateImage	createTemplate
ec2-deregister	DeregisterImage	DeleteTemplate
ec2-describe-images	DescribeImages	listTemplates
ec2-register	RegisterImage	registerTemplate

Table 4. Image Attributes API mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-image-attribute	DescribeImageAttribute	listTemplatePermissions
ec2-modify-image-attribute	ModifyImageAttribute	updateTemplatePermissions
ec2-reset-image-attribute	ResetImageAttribute	updateTemplatePermissions

Table 5. Instances API mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-instances	DescribeInstances	listVirtualMachines
ec2-run-instances	RunInstances	deployVirtualMachine
ec2-reboot-instances	RebootInstances	rebootVirtualMachine
ec2-start-instances	StartInstances	startVirtualMachine
ec2-stop-instances	StopInstances	stopVirtualMachine
ec2-terminate-instances	TerminateInstances	destroyVirtualMachine

Table 6. Instance Attributes Mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-instance-attribute	DescribeInstanceAttribute	listVirtualMachines

Table 7. Keys Pairs Mapping

EC2 command	SOAP call	CloudStack API call
ec2-add-keypair	CreateKeyPair	createSSHKeyPair
ec2-delete-keypair	DeleteKeyPair	deleteSSHKeyPair
ec2-describe-keypairs	DescribeKeyPairs	listSSHKeyPairs
ec2-import-keypair	ImportKeyPair	registerSSHKeyPair

Table 8. Passwords API Mapping

EC2 command	SOAP call	CloudStack API call
ec2-get-password	GetPasswordData	getVMPassword

Table 9. Security Groups API Mapping

EC2 command	SOAP call	CloudStack API call
ec2-authorize	AuthorizeSecurityGroupIngress	authorizeSecurityGroupIngress
ec2-add-group	CreateSecurityGroup	createSecurityGroup
ec2-delete-group	DeleteSecurityGroup	deleteSecurityGroup
ec2-describe-group	DescribeSecurityGroups	listSecurityGroups
ec2-revoke	RevokeSecurityGroupIngress	revokeSecurityGroupIngress

Table 10. Snapshots API Mapping

EC2 command	SOAP call	CloudStack API call
ec2-create-snapshot	CreateSnapshot	createSnapshot
ec2-delete-snapshot	DeleteSnapshot	deleteSnapshot
ec2-describe-snapshots	DescribeSnapshots	listSnapshots

Table 11. Volumes API Mapping

EC2 command	SOAP call	CloudStack API call
ec2-attach-volume	AttachVolume	attachVolume
ec2-create-volume	CreateVolume	createVolume
ec2-delete-volume	DeleteVolume	deleteVolume
ec2-describe-volume	DescribeVolume	listVolumes
ec2-detach-volume	DetachVolume	detachVolume

Examples

There are many tools available to interface with a AWS compatible API. In this section we provide a few examples that users of CloudStack can build upon.

Boto Examples

Boto is one of them. It is a Python package available at https://github.com/boto/boto. In this section we provide two examples of Python scripts that use Boto and have been tested with the CloudStack AWS API Interface.

First is an EC2 example. Replace the Access and Secret Keys with your own and update the endpoint.

Example 1. An EC2 Boto example

```
#!/usr/bin/env python

import sys
import os
import boto
import boto.ec2

region = boto.ec2.regioninfo.RegionInfo(name="ROOT",endpoint="localhost")
apikey='GwNnpUPrO6KgIdZu01z_ZhhZnKjtSdRwuYd4DvpzvFpyxGMvrzno2q05MB0ViBoFYtdqKd'
secretkey='t4eXLEYWw7chBhDlaKf38adCMSHx_wlds6JfSx3z9fSpSOm0AbP9Moj0oGIzy2LSC8iw'

def main():
    '''Establish connection to EC2 cloud'''
    conn = boto.connect_ec2(aws_access_key_id=apikey,
                            aws_secret_access_key=secretkey,
                            is_secure=False,
                            region=region,
                            port=7080,
                            path="/awsapi",
                            api_version="2010-11-15")

    '''Get list of images that I own'''
    images = conn.get_all_images()
    print images
    myimage = images[0]
    '''Pick an instance type'''
    vm_type='m1.small'
    reservation = myimage.run(instance_type=vm_type,security_groups=['default'])

if __name__ == '__main__':
     main()
```

Second is an S3 example. The S3 interface in CloudStack is obsolete. If you need an S3 interface you should look at systems like RiakCS, Ceph or GlusterFS. This example is here for completeness and can be adapted to other S3 endpoint.

Example 2. An S3 Boto Example

```
#!/usr/bin/env python

import sys
import os
from boto.s3.key import Key
from boto.s3.connection import S3Connection
```

(continues on next page)

(continued from previous page)

```
from boto.s3.connection import OrdinaryCallingFormat

apikey='ChOw-pwdcCFy6fpeyv6kUaR0NnhzmG3tE7HLN2z3OB_s-ogF5HjZtN4rnzKnq2UjtnHeg_yLA5gOw'
secretkey='IMY8R7CJQiSGFk4cHwfXXN3DUFXz07cCiU80eM3MCmfLs7kusgyOfm0g9qzXRXhoAPCH-
↪IRxXc3w'

cf=OrdinaryCallingFormat()

def main():
    '''Establish connection to S3 service'''
    conn = S3Connection(aws_access_key_id=apikey,aws_secret_access_key=secretkey, \
                        is_secure=False, \
                        host='localhost', \
                        port=7080, \
                        calling_format=cf, \
                        path="/awsapi/rest/AmazonS3")

    try:
        bucket=conn.create_bucket('cloudstack')
        k = Key(bucket)
        k.key = 'test'
        try:
            k.set_contents_from_filename('/Users/runseb/Desktop/s3cs.py')
        except:
            print 'could not write file'
            pass
    except:
        bucket = conn.get_bucket('cloudstack')
        k = Key(bucket)
        k.key = 'test'
        try:
            k.get_contents_to_filename('/Users/runseb/Desktop/foobar')
        except:
            print 'Could not get file'
            pass

    try:
        bucket1=conn.create_bucket('teststring')
        k=Key(bucket1)
        k.key('foobar')
        k.set_contents_from_string('This is my silly test')
    except:
        bucket1=conn.get_bucket('teststring')
        k = Key(bucket1)
        k.key='foobar'
        k.get_contents_as_string()

if __name__ == '__main__':
    main()
```

3.5.2 About Password and Key Encryption

CloudStack stores several sensitive passwords and secret keys that are used to provide security. These values are always automatically encrypted:

- Database secret key

- Database password
- SSH keys
- Compute node root password
- VPN password
- User API secret key
- VNC password

CloudStack uses the Java Simplified Encryption (JASYPT) library. The data values are encrypted and decrypted using a database secret key, which is stored in one of CloudStack's internal properties files along with the database password. The other encrypted values listed above, such as SSH keys, are in the CloudStack internal database.

Of course, the database secret key itself can not be stored in the open – it must be encrypted. How then does CloudStack read it? A second secret key must be provided from an external source during Management Server startup. This key can be provided in one of two ways: loaded from a file or provided by the CloudStack administrator. The CloudStack database has a configuration setting that lets it know which of these methods will be used. If the encryption type is set to "file," the key must be in a file in a known location. If the encryption type is set to "web," the administrator runs the utility com.cloud.utils.crypt.EncryptionSecretKeySender, which relays the key to the Management Server over a known port.

The encryption type, database secret key, and Management Server secret key are set during CloudStack installation. They are all parameters to the CloudStack database setup script (cloudstack-setup-databases). The default values are file, password, and password. It is, of course, highly recommended that you change these to more secure keys.

Changing the Default Password Encryption

Passwords are encoded when creating or updating users. CloudStack allows you to determine the default encoding and authentication mechanism for admin and user logins. Two new configurable lists have been introduced—userPasswordEncoders and userAuthenticators. userPasswordEncoders allows you to configure the order of preference for encoding passwords, whereas userAuthenticators allows you to configure the order in which authentication schemes are invoked to validate user passwords.

Additionally, the plain text user authenticator has been modified not to convert supplied passwords to their md5 sums before checking them with the database entries. It performs a simple string comparison between retrieved and supplied login passwords instead of comparing the retrieved md5 hash of the stored password against the supplied md5 hash of the password because clients no longer hash the password. The following method determines what encoding scheme is used to encode the password supplied during user creation or modification.

When a new user is created, the user password is encoded by using the first valid encoder loaded as per the sequence specified in the `UserPasswordEncoders` property in the `ComponentContext.xml` or `nonossComponentContext.xml` files. The order of authentication schemes is determined by the `UserAuthenticators` property in the same files. If Non-OSS components, such as VMware environments, are to be deployed, modify the `UserPasswordEncoders` and `UserAuthenticators` lists in the `nonossComponentContext.xml` file, for OSS environments, such as XenServer or KVM, modify the `ComponentContext.xml` file. It is recommended to make uniform changes across both the files. When a new authenticator or encoder is added, you can add them to this list. While doing so, ensure that the new authenticator or encoder is specified as a bean in both these files. The administrator can change the ordering of both these properties as preferred to change the order of schemes. Modify the following list properties available in `client/tomcatconf/nonossComponentContext.xml.in` or `client/tomcatconf/componentContext.xml.in` as applicable, to the desired order:

```
<property name="UserAuthenticators">
   <list>
      <ref bean="SHA256SaltedUserAuthenticator"/>
```

(continues on next page)

(continued from previous page)

```
      <ref bean="MD5UserAuthenticator"/>
      <ref bean="LDAPUserAuthenticator"/>
      <ref bean="PlainTextUserAuthenticator"/>
   </list>
</property>
<property name="UserPasswordEncoders">
   <list>
      <ref bean="SHA256SaltedUserAuthenticator"/>
      <ref bean="MD5UserAuthenticator"/>
      <ref bean="LDAPUserAuthenticator"/>
      <ref bean="PlainTextUserAuthenticator"/>
   </list>
</property>
```

In the above default ordering, SHA256Salt is used first for `UserPasswordEncoders`. If the module is found and encoding returns a valid value, the encoded password is stored in the user table's password column. If it fails for any reason, the MD5UserAuthenticator will be tried next, and the order continues. For `UserAuthenticators`, SHA256Salt authentication is tried first. If it succeeds, the user is logged into the Management server. If it fails, md5 is tried next, and attempts continues until any of them succeeds and the user logs in . If none of them works, the user is returned an invalid credential message.

CHAPTER 4

Upgrading CloudStack

This document contains the instructions for upgrading CloudStack from prior releases, to the current release. Please read through all sections carefully before starting.

Note: For information on the API changes and issues fixed in this release, please see the Release Notes section of the documentation

Contents:

4.1 Upgrade Instruction from 4.11.0.0

This section will guide you from CloudStack 4.11.0.0 to latest CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Add package repository for MySQL connector
3. Upgrade CloudStack management server(s)
4. Update hypervisors specific dependencies

4.1.1 Update System-VM templates

1. While running the existing 4.11.0.0 system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.1.2 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.1.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.1.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.1.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.10
```

We'll change it to point to the new package repository:

`deb http://download.cloudstack.org/ubuntu precise 4.11`

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.1.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/centos/6/4.10/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/|version|/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

2. If you use CloudStack usage server

```
$ sudo yum upgrade cloudstack-usage
```

4.1.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http://download.cloudstack.org/tools/vhd-util
```

4.1.8 Hypervisor: VMware

Warning: For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*.

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.1.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.1.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.2 Upgrade Instruction from 4.10.x

This section will guide you from CloudStack 4.10.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Add package repository for MySQL connector
3. Upgrade CloudStack management server(s)
4. Update hypervisors specific dependencies

Apache CloudStack 4.10.0.0 users who are upgrading to 4.11.0.0 should read the following discussion and workaround for a db-upgrade issue: http://markmail.org/message/f42kqr3mx4r4hgih

4.2.1 Update System-VM templates

1. While running the existing 4.10.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.2.2 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.2.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.2.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.2.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.10
```

We'll change it to point to the new package repository:

`deb http://download.cloudstack.org/ubuntu precise 4.11`

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.2.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/centos/6/4.10/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/|version|/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

2. If you use CloudStack usage server

```
$ sudo yum upgrade cloudstack-usage
```

4.2.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http://download.cloudstack.org/tools/vhd-util
```

4.2.8 Hypervisor: VMware

Warning: For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.2.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.2.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.3 Upgrade Instruction from 4.9.x

This section will guide you from CloudStack 4.9.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Add package repository for MySQL connector
3. Upgrade CloudStack management server(s)
4. Update hypervisors specific dependencies

4.3.1 Update System-VM templates

1. While running the existing 4.9.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.3.2 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.3.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.3.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.3.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.8
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.3.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/centos/6/4.8/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

2. If you use CloudStack usage server

```
$ sudo yum upgrade cloudstack-usage
```

4.3.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http://download.cloudstack.org/tools/vhd-util
```

4.3.8 Hypervisor: VMware

> **Warning:** For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.3.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.3.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.4 Upgrade Instruction from 4.8.x

This section will guide you from CloudStack 4.8.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Add package repository for MySQL connector
3. Upgrade CloudStack management server(s)
4. Update hypervisors specific dependencies

4.4.1 Update System-VM templates

1. While running the existing 4.8.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.4.2 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.4.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.4.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.4.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.8
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.4.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/centos/6/4.8/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

2. If you use CloudStack usage server

```
$ sudo yum upgrade cloudstack-usage
```

4.4.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http://download.cloudstack.org/tools/vhd-util
```

4.4.8 Hypervisor: VMware

Warning: For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.4.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.4.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.5 Upgrade Instruction from 4.7.x

This section will guide you from CloudStack 4.7.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Add package repository for MySQL connector
3. Upgrade CloudStack management server(s)
4. Update hypervisors specific dependencies

4.5.1 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.5.2 Update System-VM templates

1. While running the existing 4.7.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.

5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.5.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.5.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.5.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.7
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.5.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/rhel/4.7/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

2. If you use CloudStack usage server

   ```
   $ sudo yum upgrade cloudstack-usage
   ```

4.5.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

`wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http://download.cloudstack.org/tools/vhd-util`

4.5.8 Hypervisor: VMware

> **Warning:** For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.5.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.5.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.6 Upgrade Instruction from 4.6.x

This section will guide you from CloudStack 4.6.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Add package repository for MySQL connector
3. Upgrade CloudStack management server(s)
4. Update hypervisors specific dependencies

4.6.1 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.6.2 Update System-VM templates

1. While running the existing 4.6.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.6.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.6.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.6.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.6
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.6.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/rhel/4.6/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Remove the deprecated dependency for awsapi.

   ```
   $ sudo rpm -e --nodeps cloudstack-awsapi
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo yum upgrade cloudstack-usage
   ```

4.6.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http:
//download.cloudstack.org/tools/vhd-util
```

4.6.8 Hypervisor: VMware

> **Warning:** For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.6.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.6.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.7 Upgrade Instruction from 4.5.x

This section will guide you from CloudStack 4.5.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Install new systemvm template
3. Add package repository for MySQL connector
4. Upgrade CloudStack management server(s)
5. Update hypervisors specific dependencies

4.7.1 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.7.2 Update System-VM templates

1. While running the existing 4.5.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.7.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.7.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.7.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.5
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.7.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/rhel/4.5/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Remove the deprecated dependency for awsapi.

   ```
   $ sudo rpm -e --nodeps cloudstack-awsapi
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo yum upgrade cloudstack-usage
   ```

4.7.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http:
//download.cloudstack.org/tools/vhd-util
```

4.7.8 Hypervisor: VMware

> **Warning:** For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

```
$ sudo service cloudstack-management stop
```

2. Generate the encrypted equivalent of your vCenter password:

```
$ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
↪verbose=false
```

Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

1. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

2. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_'
where id = _id_from_step_2_;
```

3. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

4. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

5. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

6. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.7.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.7.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.7.11 System-VMs and Virtual-Routers

Once you've upgraded the packages on your management servers, you'll need to restart the system VMs. Ensure that the admin port is set to 8096 by using the "integration.api.port" global parameter. This port is used by the cloud-sysvmadm script at the end of the upgrade procedure. For information about how to set this parameter, see *configuration parameters* Changing this parameter will require management server restart. Also make sure port 8096 is open in your local host firewall to do this.

There is a script that will do this for you, all you need to do is run the script and supply the IP address for your MySQL instance and your MySQL credentials:

```
# nohup cloudstack-sysvmadm -d IPaddress -u cloud -p password -a > sysvm.log 2>&1 &
```

You can monitor the log for progress. The process of restarting the system VMs can take an hour or more.

```
# tail -f sysvm.log
```

The output to `sysvm.log` will look something like this:

```
Stopping and starting 1 secondary storage vm(s)...
Done stopping and starting secondary storage vm(s)
Stopping and starting 1 console proxy vm(s)...
Done stopping and starting console proxy vm(s).
Stopping and starting 4 running routing vm(s)...
Done restarting router(s).
```

4.8 Upgrade Instruction from 4.4.x

This section will guide you from CloudStack 4.4.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Install new systemvm template
3. Add package repository for MySQL connector
4. Upgrade CloudStack management server(s)
5. Update hypervisors specific dependencies

4.8.1 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.8.2 Update System-VM templates

1. While running the existing 4.4.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.
4. Click Register template.

   The Register template dialog box is displayed.
5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.8.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.8.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.8.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.4
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

```
$ sudo apt-get update
```

2. **Now that you have the repository configured, it's time to upgrade** the `cloudstack-management` package.

```
$ sudo apt-get upgrade cloudstack-management
```

3. If you use CloudStack usage server

```
$ sudo apt-get upgrade cloudstack-usage
```

4.8.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/rhel/4.4/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Remove the deprecated dependency for awsapi.

   ```
   $ sudo rpm -e --nodeps cloudstack-awsapi
   ```

2. **Now that you have the repository configured, it's time to upgrade the** `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo yum upgrade cloudstack-usage
   ```

4.8.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http:
//download.cloudstack.org/tools/vhd-util
```

XenServer HA

As of Apache CloudStack 4.4, CloudStack is not responsible to promote a new pool master on a Citrix XenServer pool. In case of failure of the pool master host, the responsability of electing a new pool master as been delegated back to the HA feature of XenServer. CloudStack remain responsible to honored HA capability for Compute Offerings of instances. The XenServer HA feature must be enabled only for the pool master, not for virtual-machines.

Make sure XenServer has enabled HA on the pool.

To test if poolHA is currently turned on:

```
xe pool-list params=all | grep -E "ha-enabled|ha-config"
```

Output when poolHA is ON:

```
ha-enabled ( RO): true
ha-configuration ( RO): timeout: 180
```

Output when poolHA is OFF:

```
ha-enabled ( RO): false
ha-configuration ( RO):
```

To enable poolHA, use something like this:

```
xe pool-enable-ha heartbeat-sr-uuids={SR-UUID} ha-config:timeout=180
```

Please refer to the XenServer documentation, as there are multiple ways of configuring it either on NFS, iSCSI or Fibre Channel. Be aware though, that the timeout setting is not documented. The default is 30 seconds so you may want to bump that towards 120-180 seconds.

4.8.8 Hypervisor: VMware

> **Warning:** For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

   ```
   $ sudo service cloudstack-management stop
   ```

2. Generate the encrypted equivalent of your vCenter password:

   ```
   $ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
   ↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
   ↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
   ↪verbose=false
   ```

   Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

3. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

4. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_' where id = _
↪id_from_step_2_;
```

5. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

6. **Find the ID of the correct row of vmware_data_center that you** want to update

```
select * from cloud.vmware_data_center;
```

7. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_'
where id = _id_from_step_5_;
```

8. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.8.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. **Verify that the file `/etc/cloudstack/agent/environment.properties` has a** line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.8.10 Restart management services

1. If upgrading fresh installation of 4.4.0

   If you are upgrading fresh installation of CloudStack 4.4.0, the following MySQL command must be executed before restarting the management server. If the system was running pre 4.4 and then upgraded to 4.4.0, the MySQL command is not required. Refer to: CLOUDSTACK-7813

```
use cloud;
ALTER TABLE `snapshot_policy` ADD `display` TINYINT( 1 ) NOT NULL DEFAULT '1';
```

2. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

3. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.8.11 System-VMs and Virtual-Routers

Once you've upgraded the packages on your management servers, you'll need to restart the system VMs. Ensure that the admin port is set to 8096 by using the "integration.api.port" global parameter. This port is used by the cloud-sysvmadm script at the end of the upgrade procedure. For information about how to set this parameter, see *configuration parameters* Changing this parameter will require management server restart. Also make sure port 8096 is open in your local host firewall to do this.

There is a script that will do this for you, all you need to do is run the script and supply the IP address for your MySQL instance and your MySQL credentials:

```
# nohup cloudstack-sysvmadm -d IPaddress -u cloud -p password -a > sysvm.log 2>&1 &
```

You can monitor the log for progress. The process of restarting the system VMs can take an hour or more.

```
# tail -f sysvm.log
```

The output to `sysvm.log` will look something like this:

```
Stopping and starting 1 secondary storage vm(s)...
Done stopping and starting secondary storage vm(s)
Stopping and starting 1 console proxy vm(s)...
Done stopping and starting console proxy vm(s).
Stopping and starting 4 running routing vm(s)...
Done restarting router(s).
```

4.9 Upgrade Instruction from 4.3.x

This section will guide you from CloudStack 4.3.x to CloudStack 4.11.1.0.

Any steps that are hypervisor-specific will be called out with a note.

We recommend reading through this section once or twice before beginning your upgrade procedure, and working through it on a test system before working on a production system.

Note: The following upgrade instructions should be performed regardless of hypervisor type.

Upgrade Steps:

1. Backup CloudStack database (MySQL)
2. Install new systemvm template
3. Add package repository for MySQL connector
4. Upgrade CloudStack management server(s)
5. Update hypervisors specific dependencies

4.9.1 Packages repository

Most users of CloudStack manage the installation and upgrades of CloudStack with one of Linux's predominant package systems, RPM or APT. This guide assumes you'll be using RPM and Yum (for Red Hat Enterprise Linux or CentOS), or APT and Debian packages (for Ubuntu).

Create RPM or Debian packages (as appropriate) and a repository from the 4.11.1.0 source, or check the Apache CloudStack downloads page at http://cloudstack.apache.org/downloads.html for package repositories supplied by community members. You will need them for *Management Server on Ubuntu* or *Management Server on CentOS/RHEL* and *Hypervisor: KVM* hosts upgrade.

Instructions for creating packages from the CloudStack source are in the CloudStack Installation Guide.

4.9.2 Update System-VM templates

1. While running the existing 4.3.x system, log in to the UI as root administrator.
2. In the left navigation bar, click Templates.
3. In Select view, click Templates.

4. Click Register template.

   The Register template dialog box is displayed.

5. In the Register template dialog box, specify the following values (do not change these):

Hypervisor	Description
XenServer	Name: systemvm-xenserver-4.11.1 Description: systemvm-xenserver-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: XenServer Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
KVM	Name: systemvm-kvm-4.11.1 Description: systemvm-kvm-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2 Zone: Choose the zone where this hypervisor is used Hypervisor: KVM Format: QCOW2 OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
VMware	Name: systemvm-vmware-4.11.1 Description: systemvm-vmware-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova Zone: Choose the zone where this hypervisor is used Hypervisor: VMware Format: OVA OS Type: Other Linux 64-bit (or Debian 8.0 or 9.0 64-bit) Extractable: no Password Enabled: no Public: no Featured: no Routing: no
HyperV	Name: systemvm-hyperv-4.11.1 Description: systemvm-hyperv-4.11.1 URL: http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip Zone: Choose the zone where this hypervisor is used Hypervisor: Hyper-V Format: VHD OS Type: Debian GNU/Linux 7.0 (64-bit) (or the highest Debian release number available in the dropdown) Extractable: no Password Enabled: no Public: no Featured: no Routing: no

6. Watch the screen to be sure that the template downloads successfully and enters the **READY** state. Do not proceed until this is successful.

4.9.3 Database Preparation

Backup current database

1. Stop your management server or servers. Run this on all management server hosts:

```
$ sudo service cloudstack-management stop
```

2. If you are running a usage server or usage servers, stop those as well:

```
$ sudo service cloudstack-usage stop
```

3. Make a backup of your MySQL database. If you run into any issues or need to roll back the upgrade, this will assist in debugging or restoring your existing environment. You'll be prompted for your password.

```
$ mysqldump -u root -p cloud > cloud-backup_`date '+%Y-%m-%d'.sql
$ mysqldump -u root -p cloud_usage > cloud_usage-backup_`date '+%Y-%m-%d'.
↪sql
```

4. **(KVM Only)** If primary storage of type local storage is in use, the path for this storage needs to be verified to ensure it passes new validation. Check local storage by querying the cloud.storage_pool table:

```
$ mysql -u cloud -p -e "select id,name,path from cloud.storage_pool where pool_
↪type='Filesystem'"
```

If local storage paths are found to have a trailing forward slash, remove it:

```
$ mysql -u cloud -p -e 'update cloud.storage_pool set path="/var/lib/libvirt/
↪images" where path="/var/lib/libvirt/images/"';
```

4.9.4 Management Server on Ubuntu

If you are using Ubuntu, follow this procedure to upgrade your packages. If not, skip to step *Management Server on CentOS/RHEL*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and APT repository, substitute your own URL for the ones used in these examples.

The first order of business will be to change the sources list for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent. (No changes should be necessary for hosts that are running VMware or Xen.)

4.9.5 Java 8 JRE on Ubuntu

CloudStack 4.11 requires installation of Java 8 JRE from an external PPA such as openjdk-r for Ubuntu distributions where the openjdk-8 packages are not available from the main repositories such as on Ubuntu 14.04. The PPA can be added before installation/upgrade:

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
```

Users can also choose to install Java 8 distribution from Oracle, or Xulu-8 OpenJDK distribution from Azul.

CloudStack apt repository

Start by opening `/etc/apt/sources.list.d/cloudstack.list` on any systems that have CloudStack packages installed.

This file should have one line, which contains:

```
deb http://download.cloudstack.org/ubuntu precise 4.3
```

We'll change it to point to the new package repository:

```
deb http://download.cloudstack.org/ubuntu precise 4.9
```

Setup the public key for the above repository:

```
wget -qO - http://download.cloudstack.org/release.asc | sudo apt-key add -
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Now update your apt package list:

   ```
   $ sudo apt-get update
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management` package.

   ```
   $ sudo apt-get upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo apt-get upgrade cloudstack-usage
   ```

4.9.6 Management Server on CentOS/RHEL

If you are using CentOS or RHEL, follow this procedure to upgrade your packages. If not, skip to hypervisors section *Hypervisor: XenServer*.

Note: Community Packages: This section assumes you're using the community supplied packages for CloudStack. If you've created your own packages and yum repository, substitute your own URL for the ones used in these examples.

Install new MySQL connector

Apache CloudStack 4.11.1.0 require an upgrade of the MySQL connector on CentOS. Starting with 4.9.0, cloudstack-management RPM's now depend on `mysql-connector-python` package.

MySQL connector RPM repository

Add a new yum repo `/etc/yum.repos.d/mysql.repo`:

```
[mysql-community]
name=MySQL Community connectors
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/$releasever/$basearch/
enabled=1
gpgcheck=1
```

Import GPG public key from MySQL:

```
rpm --import http://repo.mysql.com/RPM-GPG-KEY-mysql
```

Install mysql-connector

```
yum install mysql-connector-python
```

CloudStack RPM repository

The first order of business will be to change the yum repository for each system with CloudStack packages. This means all management servers, and any hosts that have the KVM agent.

(No changes should be necessary for hosts that are running VMware or Xen.)

Start by opening `/etc/yum.repos.d/cloudstack.repo` on any systems that have CloudStack packages installed.

This file should have content similar to the following:

```
[apache-cloudstack]
name=Apache CloudStack
baseurl=http://download.cloudstack.org/rhel/4.3/
enabled=1
gpgcheck=0
```

If you are using the community provided package repository, change the base url to `http://download.cloudstack.org/centos/$releasever/4.9/`.

Setup the GPG public key if you wish to enable `gpgcheck=1`:

```
rpm --import http://download.cloudstack.org/RPM-GPG-KEY
```

If you're using your own package repository, change this line to read as appropriate for your 4.11 repository.

1. Remove the deprecated dependency for awsapi.

   ```
   $ sudo rpm -e --nodeps cloudstack-awsapi
   ```

2. Now that you have the repository configured, it's time to upgrade the `cloudstack-management`.

   ```
   $ sudo yum upgrade cloudstack-management
   ```

3. If you use CloudStack usage server

   ```
   $ sudo yum upgrade cloudstack-usage
   ```

4.9.7 Hypervisor: XenServer

(XenServer only) Copy vhd-utils file on CloudStack management servers. Copy the file vhd-utils to `/usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver`.

```
wget -P /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver http:
//download.cloudstack.org/tools/vhd-util
```

XenServer HA

As of Apache CloudStack 4.4, CloudStack is not responsible to promote a new pool master on a Citrix XenServer pool. In case of failure of the pool master host, the responsability of electing a new pool master as been delegated back to the HA feature of XenServer. CloudStack remain responsible to honored HA capability for Compute Offerings of instances. The XenServer HA feature must be enabled only for the pool master, not for virtual-machines.

Make sure XenServer has enabled HA on the pool.

To test if poolHA is currently turned on:

```
xe pool-list params=all | grep -E "ha-enabled|ha-config"
```

Output when poolHA is ON:

```
ha-enabled ( RO): true
ha-configuration ( RO): timeout: 180
```

Output when poolHA is OFF:

```
ha-enabled ( RO): false
ha-configuration ( RO):
```

To enable poolHA, use something like this:

```
xe pool-enable-ha heartbeat-sr-uuids={SR-UUID} ha-config:timeout=180
```

Please refer to the XenServer documentation, as there are multiple ways of configuring it either on NFS, iSCSI or Fibre Channel. Be aware though, that the timeout setting is not documented. The default is 30 seconds so you may want to bump that towards 120-180 seconds.

4.9.8 Hypervisor: VMware

> **Warning:** For VMware hypervisor CloudStack management server packages must be build using "noredist". Refer to *Building Non-OSS*

(VMware only) Additional steps are required for each VMware cluster. These steps will not affect running guests in the cloud. These steps are required only for clouds using VMware clusters:

1. Stop the Management Server:

   ```
   $ sudo service cloudstack-management stop
   ```

2. Generate the encrypted equivalent of your vCenter password:

   ```
   $ java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.2.jar␣
   ↪org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh input="_
   ↪your_vCenter_password_" password="cat /etc/cloudstack/management/key"␣
   ↪verbose=false
   ```

   Store the output from this step, we need to add this in cluster_details table and vmware_data_center tables in place of the plain text password

3. Find the ID of the row of cluster_details table that you have to update:

```
$ mysql -u <username> -p<password>
```

```
select * from cloud.cluster_details;
```

4. Update the plain text password with the encrypted one

```
update cloud.cluster_details set value = '_ciphertext_from_step_1_' where id = _
↪id_from_step_2_;
```

5. Confirm that the table is updated:

```
select * from cloud.cluster_details;
```

6. Find the ID of the correct row of vmware_data_center that you want to update

```
select * from cloud.vmware_data_center;
```

7. update the plain text password with the encrypted one:

```
update cloud.vmware_data_center set password = '_ciphertext_from_step_1_' where␣
↪id = _id_from_step_5_;
```

8. Confirm that the table is updated:

```
select * from cloud.vmware_data_center;
```

4.9.9 Hypervisor: KVM

KVM on Ubuntu

(KVM only) Additional steps are required for each KVM host. These steps will not affect running guests in the cloud. These steps are required only for clouds using KVM as hosts and only on the KVM hosts.

1. Configure the *APT repo* as detailed above.
2. Stop the running agent.

```
$ sudo service cloudstack-agent stop
```

3. Update the agent software.

```
$ sudo apt-get upgrade cloudstack-agent
```

4. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

5. Start the agent.

```
$ sudo service cloudstack-agent start
```

KVM on CentOS/RHEL

For KVM hosts, upgrade the `cloudstack-agent` package

1. Configure the *CloudStack RPM repository* as detailed above.

```
$ sudo yum upgrade cloudstack-agent
```

2. Verify that the file `/etc/cloudstack/agent/environment.properties` has a line that reads:

```
paths.script=/usr/share/cloudstack-common
```

   If not, add the line.

3. Restart the agent:

```
$ sudo service cloudstack-agent stop
$ sudo killall jsvc
$ sudo service cloudstack-agent start
```

4.9.10 Restart management services

1. Now it's time to start the management server

```
$ sudo service cloudstack-management start
```

2. If you use it, start the usage server

```
$ sudo service cloudstack-usage start
```

4.9.11 System-VMs and Virtual-Routers

Once you've upgraded the packages on your management servers, you'll need to restart the system VMs. Ensure that the admin port is set to 8096 by using the "integration.api.port" global parameter. This port is used by the cloud-sysvmadm script at the end of the upgrade procedure. For information about how to set this parameter, see *configuration parameters* Changing this parameter will require management server restart. Also make sure port 8096 is open in your local host firewall to do this.

There is a script that will do this for you, all you need to do is run the script and supply the IP address for your MySQL instance and your MySQL credentials:

```
# nohup cloudstack-sysvmadm -d IPaddress -u cloud -p password -a > sysvm.log 2>&1 &
```

You can monitor the log for progress. The process of restarting the system VMs can take an hour or more.

```
# tail -f sysvm.log
```

The output to `sysvm.log` will look something like this:

```
Stopping and starting 1 secondary storage vm(s)...
Done stopping and starting secondary storage vm(s)
Stopping and starting 1 console proxy vm(s)...
Done stopping and starting console proxy vm(s).
Stopping and starting 4 running routing vm(s)...
Done restarting router(s).
```

CHAPTER 5

Usage Guide

This guide is aimed at Administrators of a CloudStack based Cloud

5.1 User Interface

5.1.1 Log In to the UI

CloudStack provides a web-based UI that can be used by both administrators and end users. The appropriate version of the UI is displayed depending on the credentials used to log in. The UI is available in popular browsers including IE7, IE8, IE9, Firefox 3.5+, Firefox 4, Safari 4, and Safari 5. The URL is: (substitute your own management server IP address)

```
http://<management-server-ip-address>:8080/client
```

On a fresh Management Server installation, a guided tour splash screen appears. On later visits, you'll see a login screen where you specify the following to proceed to your Dashboard:

Username -> The user ID of your account. The default username is admin.

Password -> The password associated with the user ID. The password for the default username is password.

Domain -> If you are a root user, leave this field blank.

If you are a user in the sub-domains, enter the full path to the domain, excluding the root domain.

For example, suppose multiple levels are created under the root domain, such as Comp1/hr. The users in the Comp1 domain should enter Comp1 in the Domain field, whereas the users in the Comp1/sales domain should enter Comp1/sales.

For more guidance about the choices that appear when you log in to this UI, see Logging In as the Root Administrator.

End User's UI Overview

The CloudStack UI helps users of cloud infrastructure to view and use their cloud resources, including virtual machines, templates and ISOs, data volumes and snapshots, guest networks, and IP addresses. If the user is a member or

administrator of one or more CloudStack projects, the UI can provide a project-oriented view.

Root Administrator's UI Overview

The CloudStack UI helps the CloudStack administrator provision, view, and manage the cloud infrastructure, domains, user accounts, projects, and configuration settings. The first time you start the UI after a fresh Management Server installation, you can choose to follow a guided tour to provision your cloud infrastructure. On subsequent logins, the dashboard of the logged-in user appears. The various links in this screen and the navigation bar on the left provide access to a variety of administrative functions. The root administrator can also use the UI to perform all the same tasks that are present in the end-user's UI.

Logging In as the Root Administrator

After the Management Server software is installed and running, you can run the CloudStack user interface. This UI is there to help you provision, view, and manage your cloud infrastructure.

1. Open your favorite Web browser and go to this URL. Substitute the IP address of your own Management Server:

   ```
   http://<management-server-ip-address>:8080/client
   ```

   After logging into a fresh Management Server installation, a guided tour splash screen appears. On later visits, you'll be taken directly into the Dashboard.

2. If you see the first-time splash screen, choose one of the following.

   - **Continue with basic setup.** Choose this if you're just trying CloudStack, and you want a guided walk-through of the simplest possible configuration so that you can get started right away. We'll help you set up a cloud with the following features: a single machine that runs CloudStack software and uses NFS to provide storage; a single machine running VMs under the XenServer or KVM hypervisor; and a shared public network.

     The prompts in this guided tour should give you all the information you need, but if you want just a bit more detail, you can follow along in the Trial Installation Guide.

   - **I have used CloudStack before.** Choose this if you have already gone through a design phase and planned a more sophisticated deployment, or you are ready to start scaling up a trial cloud that you set up earlier with the basic setup screens. In the Administrator UI, you can start using the more powerful features of CloudStack, such as advanced VLAN networking, high availability, additional network elements such as load balancers and firewalls, and support for multiple hypervisors including Citrix XenServer, KVM, and VMware vSphere.

     The root administrator Dashboard appears.

3. You should set a new root administrator password. If you chose basic setup, you'll be prompted to create a new password right away. If you chose experienced user, use the steps in *Changing the Root Password*.

Warning: You are logging in as the root administrator. This account manages the CloudStack deployment, including physical infrastructure. The root administrator can modify configuration settings to change basic functionality, create or delete user accounts, and take many actions that should be performed only by an authorized person. Please change the default password to a new, unique password.

Changing the Root Password

During installation and ongoing cloud administration, you will need to log in to the UI as the root administrator. The root administrator account manages the CloudStack deployment, including physical infrastructure. The root administrator can modify configuration settings to change basic functionality, create or delete user accounts, and take many actions that should be performed only by an authorized person. When first installing CloudStack, be sure to change the default password to a new, unique value.

1. Open your favorite Web browser and go to this URL. Substitute the IP address of your own Management Server:

   ```
   http://<management-server-ip-address>:8080/client
   ```

2. Log in to the UI using the current root user ID and password. The default is admin, password.
3. Click Accounts.
4. Click the admin account name.
5. Click View Users.
6. Click the admin user name.
7. Click the Change Password button.
8. Type the new password, and click OK.

5.2 Managing Accounts, Users and Domains

5.2.1 Roles, Accounts, Users, and Domains

Roles

A role represents a set of allowed functions. All CloudStack accounts have a role attached to them that enforce access rules on them to be allowed or disallowed to make an API request. Typically there are four default roles: root admin, resource admin, domain admin and user.

Accounts

An account typically represents a customer of the service provider or a department in a large organization. Multiple users can exist in an account.

Domains

Accounts are grouped by domains. Domains usually contain multiple accounts that have some logical relationship to each other and a set of delegated administrators with some authority over the domain and its subdomains. For example, a service provider with several resellers could create a domain for each reseller.

For each account created, the Cloud installation creates three different types of user accounts: root administrator, domain administrator, and user.

Users

Users are like aliases in the account. Users in the same account are not isolated from each other, but they are isolated from users in other accounts. Most installations need not surface the notion of users; they just have one user per account. The same user cannot belong to multiple accounts.

Username is unique in a domain across accounts in that domain. The same username can exist in other domains, including sub-domains. Domain name can repeat only if the full pathname from root is unique. For example, you can create root/d1, as well as root/foo/d1, and root/sales/d1.

Administrators are accounts with special privileges in the system. There may be multiple administrators in the system. Administrators can create or delete other administrators, and change the password for any user in the system.

Domain Administrators

Domain administrators can perform administrative operations for users who belong to that domain. Domain administrators do not have visibility into physical servers or other domains.

Root Administrator

Root administrators have complete access to the system, including managing templates, service offerings, customer care administrators, and domains

Resource Ownership

Resources belong to the account, not individual users in that account. For example, billing, resource limits, and so on are maintained by the account, not the users. A user can operate on any resource in the account provided the user has privileges for that operation. The privileges are determined by the role. A root administrator can change the ownership of any virtual machine from one account to any other account by using the assignVirtualMachine API. A domain or sub-domain administrator can do the same for VMs within the domain from one account to any other account in the domain or any of its sub-domains.

5.2.2 Using Dynamic Roles

In addition to the four default roles, the dynamic role-based API checker feature allows CloudStack root admins to create new roles with customized permissions. The allow/deny rules can be configured dynamically during runtime without restarting the management server(s).

For backward compatiblity, all roles resolve to one of the four role types: admin, resource admin, domain admin and user. A new role can be created using the roles tab in the UI and specifying a name, a role type and optionally a description.

Role specific rules can be configured through the rules tab on role specific details page. A rule is either an API name or a wildcard string that are one of allow or deny permission and optionally a description.

When a user makes an API request, the backend checks the requested API against configured rules (in the order the rules were configured) for the caller user-account's role. It will iterate through the rules and would allow the API request if the API matches an allow rule, else if it matches a deny rule it would deny the request. Next, if the request API fails to match any of the configured rules it would allow if the requested API's default authorized annotaions allow that user role type and finally deny the user API request if it fails to be explicitly allowed/denied by the role permission rules or the default API authorize annotations. Note: to avoid root admin being locked out of the system, all root admin accounts are allowed all APIs.

The dynamic-roles feature is enabled by default only for all new CloudStack installations since version 4.9.x.

After an upgrade, existing deployments can be migrated to use this feature by running a migration tool by the CloudStack admin. The migration tool is located at `/usr/share/cloudstack-common/scripts/util/migrate-dynamicroles.py`.

NOTE: If you have not changed your commands.properties file at any time, then it is recommended to use the -D (default) option as otherwise new API commands may not be added to the dynamic roles database.

During migration, this tool enables an internal flag in the database, copies existing static role-based rules from provided commands.properties file (typically at `/etc/cloudstack/management/commands.properties`) to the database and renames the commands.properties file (typically to /etc/cloudstack/management/commands.properties.deprecated). The migration process does not require restarting the management server(s).

Usage: `migrate-dynamicroles.py` [options] [-h for help]

Options:

-b DB	The name of the database, default: cloud
-u USER	User name a MySQL user with privileges on cloud database, default: cloud
-p PASSWORD	Password of a MySQL user with privileges on cloud database
-H HOST	Host or IP of the MySQL server
-P PORT	Host or IP of the MySQL server, default: 3306
-f FILE	The commands.properties file, default: /etc/cloudstack/management/commands.properties
-d	Dry run and debug operations this tool will perform
-D	Use the default configuration for Dynamic Roles (does not import commands.properties)

Example:

```
sudo python /usr/share/cloudstack-common/scripts/util/migrate-dynamicroles.py -u␣
↪cloud -p cloud -H localhost -P 3306 -f /etc/cloudstack/management/commands.
↪properties

sudo python /usr/share/cloudstack-common/scripts/util/migrate-dynamicroles.py -u␣
↪cloud -p cloud -H localhost -P 3306 -D
```

If you've multiple management servers, remove or rename the commands.properties file on all management servers typically in /etc/cloudstack/management path, after running the migration tool for the first management server

5.2.3 Dedicating Resources to Accounts and Domains

The root administrator can dedicate resources to a specific domain or account that needs private infrastructure for additional security or performance guarantees. A zone, pod, cluster, or host can be reserved by the root administrator for a specific domain or account. Only users in that domain or its subdomain may use the infrastructure. For example, only users in a given domain can create guests in a zone dedicated to that domain.

There are several types of dedication available:

- Explicit dedication. A zone, pod, cluster, or host is dedicated to an account or domain by the root administrator during initial deployment and configuration.
- Strict implicit dedication. A host will not be shared across multiple accounts. For example, strict implicit dedication is useful for deployment of certain types of applications, such as desktops, where no host can be shared between different accounts without violating the desktop software's terms of license.

- Preferred implicit dedication. The VM will be deployed in dedicated infrastructure if possible. Otherwise, the VM can be deployed in shared infrastructure.

5.2.4 How to Dedicate a Zone, Cluster, Pod, or Host to an Account or Domain

For explicit dedication: When deploying a new zone, pod, cluster, or host, the root administrator can click the Dedicated checkbox, then choose a domain or account to own the resource.

To explicitly dedicate an existing zone, pod, cluster, or host: log in as the root admin, find the resource in the UI, and click the Dedicate button.

For implicit dedication: The administrator creates a compute service offering and in the Deployment Planner field, chooses ImplicitDedicationPlanner. Then in Planner Mode, the administrator specifies either Strict or Preferred, depending on whether it is permissible to allow some use of shared resources when dedicated resources are not available. Whenever a user creates a VM based on this service offering, it is allocated on one of the dedicated hosts.

How to Use Dedicated Hosts

To use an explicitly dedicated host, use the explicit-dedicated type of affinity group (see "Affinity Groups"). For example, when creating a new VM, an end user can choose to place it on dedicated infrastructure. This operation will succeed only if some infrastructure has already been assigned as dedicated to the user's account or domain.

Behavior of Dedicated Hosts, Clusters, Pods, and Zones

The administrator can live migrate VMs away from dedicated hosts if desired, whether the destination is a host reserved for a different account/domain or a host that is shared (not dedicated to any particular account or domain). CloudStack will generate an alert, but the operation is allowed.

Dedicated hosts can be used in conjunction with host tags. If both a host tag and dedication are requested, the VM will be placed only on a host that meets both requirements. If there is no dedicated resource available to that user that also has the host tag requested by the user, then the VM will not deploy.

If you delete an account or domain, any hosts, clusters, pods, and zones that were dedicated to it are freed up. They will now be available to be shared by any account or domain, or the administrator may choose to re-dedicate them to a different account or domain.

System VMs and virtual routers affect the behavior of host dedication. System VMs and virtual routers are owned by the CloudStack system account, and they can be deployed on any host. They do not adhere to explicit dedication. The presence of system vms and virtual routers on a host makes it unsuitable for strict implicit dedication. The host can not be used for strict implicit dedication, because the host already has VMs of a specific account (the default system account). However, a host with system VMs or virtual routers can be used for preferred implicit dedication.

5.2.5 Using an LDAP Server for User Authentication

You can use an external LDAP server such as Microsoft Active Directory or ApacheDS to authenticate CloudStack end-users. CloudStack will search the external LDAP directory tree starting at a specified base directory and gets user info such as first name, last name, email and username.

To authenticate, username and password entered by the user are used. Cloudstack does a search for a user with the given username. If it exists, it does a bind request with DN and password.

To set up LDAP authentication in CloudStack, call the CloudStack API command `addLdapConfiguration` and provide Hostname or IP address and listening port of the LDAP server. You could configure multiple servers as well. These are expected to be replicas. If one fails, the next one is used.

The following global configurations should also be configured (the default values are for openldap)

- `ldap.basedn`: Sets the basedn for LDAP. Ex: **OU=APAC,DC=company,DC=com**
- `ldap.bind.principal`, `ldap.bind.password`: DN and password for a user who can list all the users in the above basedn. Ex: **CN=Administrator, OU=APAC, DC=company, DC=com**
- `ldap.user.object`: object type of users within LDAP. Defaults value is **user** for AD and **interorgperson** for openldap.
- `ldap.email.attribute`: email attribute within ldap for a user. Default value for AD and openldap is **mail**.
- `ldap.firstname.attribute`: firstname attribute within ldap for a user. Default value for AD and openldap is **givenname**.
- `ldap.lastname.attribute`: lastname attribute within ldap for a user. Default value for AD and openldap is **sn**.
- `ldap.username.attribute`: username attribute for a user within LDAP. Default value is **SAMAccountName** for AD and **uid** for openldap.

Restricting LDAP users to a group:

- `ldap.search.group.principle`: this is optional and if set only users from this group are listed.

LDAP SSL:

If the LDAP server requires SSL, you need to enable the below configurations. Before enabling SSL for LDAP, you need to get the certificate which the LDAP server is using and add it to a trusted keystore. You will need to know the path to the keystore and the password.

- `ldap.truststore` : truststore path
- `ldap.truststore.password` : truststore password

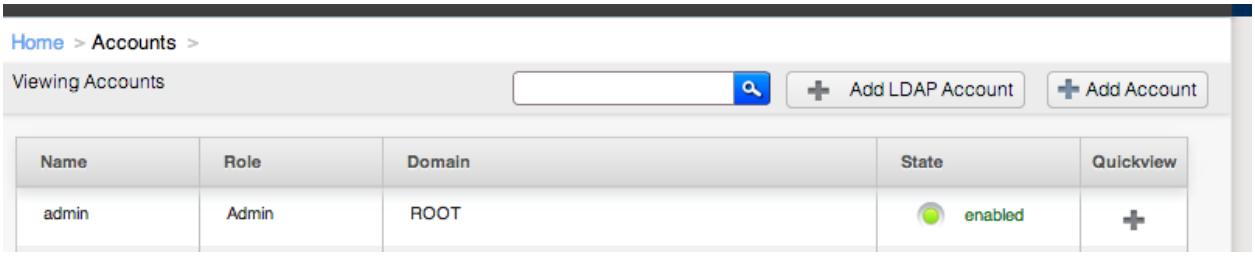
LDAP groups:

- `ldap.group.object`: object type of groups within LDAP. Default value is group for AD and **groupOfUniqueNames** for openldap.
- `ldap.group.user.uniquemember`: attribute for uniquemembers within a group. Default value is **member** for AD and **uniquemember** for openldap.

Once configured, on Add Account page, you will see an "Add LDAP Account" button which opens a dialog and the selected users can be imported.

You could also use api commands: `listLdapUsers`, `ldapCreateAccount` and `importLdapUsers`.

Once LDAP is enabled, the users will not be allowed to changed password directly in cloudstack.



5.2.6 Using a SAML 2.0 Identity Provider for User Authentication

You can use a SAML 2.0 Identity Provider with CloudStack for user authentication. This will require enabling the SAML 2.0 service provider plugin in CloudStack. To do that first, enable the SAML plugin by setting `saml2.enabled` to `true` and restart management server.

Starting 4.5.2, the SAML plugin uses an authorization workflow where users should be authorized by an admin using `authorizeSamlSso` API before those users can use Single Sign On against a specific IDP. This can be done by ticking the enable SAML Single Sign On checkbox and selecting a IDP when adding or importing users. For existing users, admin can go to the user's page and click on configure SAML SSO option to enable/disable SSO for a user and select a Identity Provider. A user can be authorized to authenticate against only one IDP.

The CloudStack service provider metadata is accessible using the `getSPMetadata` API command, or from the URL http://acs-server:8080/client/api?command=getSPMetadata where acs-server is the domain name or IP address of the management server. The IDP administrator can get the SP metadata from CloudStack and add it to their IDP server.

To start a SAML 2.0 Single Sign-On authentication, on the login page users need to select the Identity Provider or Institution/Department they can authenticate with and click on Login button. This action call the `samlsso` API command which will redirect the user to the Identity Provider's login page. Upon successful authentication, the IdP will redirect the user to CloudStack. In case a user has multiple user accounts with the same username (across domains) for the same authorized IDP, that user would need to specify domainpath after selecting their IDP server from the dropdown list. By default, users don't need to specify any domain path. After a user is successfully authenticated by an IDP server, the SAML authentication plugin finds user accounts whose username match the username attribute value returned by the SAML authentication response; it fails only when it finds that there are multiple user accounts with the same user name for the specific IDP otherwise the unique useraccount is allowed to proceed and the user is logged into their account.

Limitations:

- The plugin uses a user attribute returned by the IDP server in the SAML response to find and map the authorized user in CloudStack. The default attribute is *uid*.
- The SAML authentication plugin supports HTTP-Redirect and HTTP-Post bindings.
- Tested with Shibboleth 2.4, SSOCircle, Microsoft ADFS, OneLogin, Feide OpenIDP, PingIdentity.

The following global configuration should be configured:

- `saml2.enabled`: Indicates whether SAML SSO plugin is enabled or not true. Default is **false**
- `saml2.sp.id`: SAML2 Service Provider Identifier string
- `saml2.idp.metadata.url`: SAML2 Identity Provider Metadata XML Url or Filename. If a URL is not provided, it will look for a file in the config directory /etc/cloudstack/management
- `saml2.default.idpid`: The default IdP entity ID to use only in case of multiple IdPs
- `saml2.sigalg`: The algorithm to use to when signing a SAML request. Default is SHA1, allowed algorithms: SHA1, SHA256, SHA384, SHA512.

- `saml2.redirect.url`: The CloudStack UI url the SSO should redirected to when successful. Default is **http://localhost:8080/client**
- `saml2.sp.org.name`: SAML2 Service Provider Organization Name
- `saml2.sp.org.url`: SAML2 Service Provider Organization URL
- `saml2.sp.contact.email`: SAML2 Service Provider Contact Email Address
- `saml2.sp.contact.person`: SAML2 Service Provider Contact Person Name
- `saml2.sp.slo.url`: SAML2 CloudStack Service Provider Single Log Out URL
- `saml2.sp.sso.url`: SAML2 CloudStack Service Provider Single Sign On URL
- `saml2.user.attribute`: Attribute name to be looked for in SAML response that will contain the username. Default is **uid**
- `saml2.timeout`: SAML2 IDP Metadata refresh interval in seconds, minimum value is set to 300. Default is 1800

5.3 Using Projects to Organize User Resources

5.3.1 Overview of Projects

Projects are used to organize people and resources. CloudStack users within a single domain can group themselves into project teams so they can collaborate and share virtual resources such as VMs, snapshots, templates, data disks, and IP addresses. CloudStack tracks resource usage per project as well as per user, so the usage can be billed to either a user account or a project. For example, a private cloud within a software company might have all members of the QA department assigned to one project, so the company can track the resources used in testing while the project members can more easily isolate their efforts from other users of the same cloud

You can configure CloudStack to allow any user to create a new project, or you can restrict that ability to just CloudStack administrators. Once you have created a project, you become that project's administrator, and you can add others within your domain to the project. CloudStack can be set up either so that you can add people directly to a project, or so that you have to send an invitation which the recipient must accept. Project members can view and manage all virtual resources created by anyone in the project (for example, share VMs). A user can be a member of any number of projects and can switch views in the CloudStack UI to show only project-related information, such as project VMs, fellow project members, project-related alerts, and so on.

The project administrator can pass on the role to another project member. The project administrator can also add more members, remove members from the project, set new resource limits (as long as they are below the global defaults set by the CloudStack administrator), and delete the project. When the administrator removes a member from the project, resources created by that user, such as VM instances, remain with the project. This brings us to the subject of resource ownership and which resources can be used by a project.

Resources created within a project are owned by the project, not by any particular CloudStack account, and they can be used only within the project. A user who belongs to one or more projects can still create resources outside of those projects, and those resources belong to the user's account; they will not be counted against the project's usage or resource limits. You can create project-level networks to isolate traffic within the project and provide network services such as port forwarding, load balancing, VPN, and static NAT. A project can also make use of certain types of resources from outside the project, if those resources are shared. For example, a shared network or public template is available to any project in the domain. A project can get access to a private template if the template's owner will grant permission. A project can use any service offering or disk offering available in its domain; however, you can not create private service and disk offerings at the project level..

5.3.2 Configuring Projects

Before CloudStack users start using projects, the CloudStack administrator must set up various systems to support them, including membership invitations, limits on project resources, and controls on who can create projects.

Setting Up Invitations

CloudStack can be set up either so that project administrators can add people directly to a project, or so that it is necessary to send an invitation which the recipient must accept. The invitation can be sent by email or through the user's CloudStack account. If you want administrators to use invitations to add members to projects, turn on and set up the invitations feature in CloudStack.

1. Log in as administrator to the CloudStack UI.
2. In the left navigation, click Global Settings.
3. In the search box, type project and click the search button.
4. In the search results, you can see a few other parameters you need to set to control how invitations behave. The table below shows global configuration parameters related to project invitations. Click the edit button to set each parameter.

Configuration Parameters	Description
project.invite.required	Set to true to turn on the invitations feature.
project.email.sender	The email address to show in the From field of invitation emails.
project.invite.timeout	Amount of time to allow for a new member to respond to the invitation.
project.smtp.host	Name of the host that acts as an email server to handle invitations.
project.smtp.password	(Optional) Password required by the SMTP server. You must also set project.smtp.username and set project.smtp.useAuth to true.
project.smtp.port	SMTP server's listening port.
project.smtp.useAuth	Set to true if the SMTP server requires a username and password.
project.smtp.username	(Optional) User name required by the SMTP server for authentication. You must also set project.smtp.password and set project.smtp.useAuth to true..

5. Restart the Management Server:

```
service cloudstack-management restart
```

Setting Resource Limits for Projects

The CloudStack administrator can set global default limits to control the amount of resources that can be owned by each project in the cloud. This serves to prevent uncontrolled usage of resources such as snapshots, IP addresses, and virtual machine instances. Domain administrators can override these resource limits for individual projects with their domains, as long as the new limits are below the global defaults set by the CloudStack root administrator. The root administrator can also set lower resource limits for any project in the cloud

Setting Per-Project Resource Limits

The CloudStack root administrator or the domain administrator of the domain where the project resides can set new resource limits for an individual project. The project owner can set resource limits only if the owner is also a domain or root administrator.

The new limits must be below the global default limits set by the CloudStack administrator (as described in *"Setting Resource Limits for Projects"*). If the project already owns more of a given type of resource than the new maximum, the resources are not affected; however, the project can not add any new resources of that type until the total drops below the new limit.

1. Log in as administrator to the CloudStack UI.
2. In the left navigation, click Projects.
3. In Select View, choose Projects.
4. Click the name of the project you want to work with.
5. Click the Resources tab. This tab lists the current maximum amount that the project is allowed to own for each type of resource.
6. Type new values for one or more resources.
7. Click Apply.

Setting the Global Project Resource Limits

1. Log in as administrator to the CloudStack UI.
2. In the left navigation, click Global Settings.
3. In the search box, type max.projects and click the search button.
4. In the search results, you will see the parameters you can use to set per-project maximum resource amounts that apply to all projects in the cloud. No project can have more resources, but an individual project can have lower limits. Click the edit button to set each parameter.

max.project.publicips	Maximum number of public IP addresses that can be owned by any project in the cloud. See About Public IP Addresses.
max.project.snapshots	Maximum number of snapshots that can be owned by any project in the cloud. See Working with Snapshots.
max.project.templates	Maximum number of templates that can be owned by any project in the cloud. See Working with Templates.
max.project.uservms	Maximum number of guest virtual machines that can be owned by any project in the cloud. See Working With Virtual Machines.
max.project.volumes	Maximum number of data volumes that can be owned by any project in the cloud. See Working with Volumes.

5. Restart the Management Server.

```
# service cloudstack-management restart
```

Setting Project Creator Permissions

You can configure CloudStack to allow any user to create a new project, or you can restrict that ability to just CloudStack administrators.

1. Log in as administrator to the CloudStack UI.
2. In the left navigation, click Global Settings.
3. In the search box, type allow.user.create.projects.

4. Click the edit button to set the parameter.

   `allow.user.create.projects`

   Set to true to allow end users to create projects. Set to false if you want only the CloudStack root administrator and domain administrators to create projects.

5. Restart the Management Server.

```
# service cloudstack-management restart
```

5.3.3 Creating a New Project

CloudStack administrators and domain administrators can create projects. If the global configuration parameter allow.user.create.projects is set to true, end users can also create projects.

1. Log in as administrator to the CloudStack UI.
2. In the left navigation, click Projects.
3. In Select view, click Projects.
4. Click New Project.
5. Give the project a name and description for display to users, then click Create Project.
6. A screen appears where you can immediately add more members to the project. This is optional. Click Next when you are ready to move on.
7. Click Save.

5.3.4 Adding Members to a Project

New members can be added to a project by the project's administrator, the domain administrator of the domain where the project resides or any parent domain, or the CloudStack root administrator. There are two ways to add members in CloudStack, but only one way is enabled at a time:

- If invitations have been enabled, you can send invitations to new members.
- If invitations are not enabled, you can add members directly through the UI.

Sending Project Membership Invitations

Use these steps to add a new member to a project if the invitations feature is enabled in the cloud as described in *"Setting Up Invitations"*. If the invitations feature is not turned on, use the procedure in Adding Project Members From the UI.

1. Log in to the CloudStack UI.
2. In the left navigation, click Projects.
3. In Select View, choose Projects.
4. Click the name of the project you want to work with.
5. Click the Invitations tab.
6. In Add by, select one of the following:

(a) Account – The invitation will appear in the user's Invitations tab in the Project View. See Using the Project View.

(b) Email – The invitation will be sent to the user's email address. Each emailed invitation includes a unique code called a token which the recipient will provide back to CloudStack when accepting the invitation. Email invitations will work only if the global parameters related to the SMTP server have been set. See *"Setting Up Invitations"*.

7. Type the user name or email address of the new member you want to add, and click Invite. Type the CloudStack user name if you chose Account in the previous step. If you chose Email, type the email address. You can invite only people who have an account in this cloud within the same domain as the project. However, you can send the invitation to any email address.
8. To view and manage the invitations you have sent, return to this tab. When an invitation is accepted, the new member will appear in the project's Accounts tab.

Adding Project Members From the UI

The steps below tell how to add a new member to a project if the invitations feature is not enabled in the cloud. If the invitations feature is enabled cloud,as described in *"Setting Up Invitations"*, use the procedure in *"Sending Project Membership Invitations"*.

1. Log in to the CloudStack UI.
2. In the left navigation, click Projects.
3. In Select View, choose Projects.
4. Click the name of the project you want to work with.
5. Click the Accounts tab. The current members of the project are listed.
6. Type the account name of the new member you want to add, and click Add Account. You can add only people who have an account in this cloud and within the same domain as the project.

5.3.5 Accepting a Membership Invitation

If you have received an invitation to join a CloudStack project, and you want to accept the invitation, follow these steps:

1. Log in to the CloudStack UI.
2. In the left navigation, click Projects.
3. In Select View, choose Invitations.
4. If you see the invitation listed onscreen, click the Accept button.

   Invitations listed on screen were sent to you using your CloudStack account name.
5. If you received an email invitation, click the Enter Token button, and provide the project ID and unique ID code (token) from the email.

5.3.6 Suspending or Deleting a Project

When a project is suspended, it retains the resources it owns, but they can no longer be used. No new resources or members can be added to a suspended project.

When a project is deleted, its resources are destroyed, and member accounts are removed from the project. The project's status is shown as Disabled pending final deletion.

A project can be suspended or deleted by the project administrator, the domain administrator of the domain the project belongs to or of its parent domain, or the CloudStack root administrator.

1. Log in to the CloudStack UI.
2. In the left navigation, click Projects.
3. In Select View, choose Projects.
4. Click the name of the project.
5. Click one of the buttons:

   To delete, use

   To suspend, use

5.3.7 Using the Project View

If you are a member of a project, you can use CloudStack's project view to see project members, resources consumed, and more. The project view shows only information related to one project. It is a useful way to filter out other information so you can concentrate on a project status and resources.

1. Log in to the CloudStack UI.
2. Click Project View.
3. The project dashboard appears, showing the project's VMs, volumes, users, events, network settings, and more. From the dashboard, you can:
   - Click the Accounts tab to view and manage project members. If you are the project administrator, you can add new members, remove members, or change the role of a member from user to admin. Only one member at a time can have the admin role, so if you set another user's role to admin, your role will change to regular user.
   - (If invitations are enabled) Click the Invitations tab to view and manage invitations that have been sent to new project members but not yet accepted. Pending invitations will remain in this list until the new member accepts, the invitation timeout is reached, or you cancel the invitation.

5.4 Service Offerings

In addition to the physical and logical infrastructure of your cloud and the CloudStack software and servers, you also need a layer of user services so that people can actually make use of the cloud. This means not just a user UI, but a set of options and resources that users can choose from, such as templates for creating virtual machines, disk storage, and more. If you are running a commercial service, you will be keeping track of what services and resources users are consuming and charging them for that usage. Even if you do not charge anything for people to use your cloud – say, if the users are strictly internal to your organization, or just friends who are sharing your cloud – you can still keep track of what services they use and how much of them.

5.4.1 Service Offerings, Disk Offerings, Network Offerings, and Templates

A user creating a new instance can make a variety of choices about its characteristics and capabilities. CloudStack provides several ways to present users with choices when creating a new instance:

- Service Offerings, defined by the CloudStack administrator, provide a choice of CPU speed, number of CPUs, RAM size, tags on the root disk, and other choices. See Creating a New Compute Offering.

- Disk Offerings, defined by the CloudStack administrator, provide a choice of disk size and IOPS (Quality of Service) for primary data storage. See Creating a New Disk Offering.
- Network Offerings, defined by the CloudStack administrator, describe the feature set that is available to end users from the virtual router or external networking devices on a given guest network. See Network Offerings.
- Templates, defined by the CloudStack administrator or by any CloudStack user, are the base OS images that the user can choose from when creating a new instance. For example, CloudStack includes CentOS as a template. See Working with Templates.

In addition to these choices that are provided for users, there is another type of service offering which is available only to the CloudStack root administrator, and is used for configuring virtual infrastructure resources. For more information, see Upgrading a Virtual Router with System Service Offerings.

5.4.2 Compute and Disk Service Offerings

A service offering is a set of virtual hardware features such as CPU core count and speed, memory, and disk size. The CloudStack administrator can set up various offerings, and then end users choose from the available offerings when they create a new VM. Based on the user's selected offering, CloudStack emits usage records that can be integrated with billing systems.

Some characteristics of service offerings must be defined by the CloudStack administrator, and others can be left undefined so that the end-user can enter their own desired values. This is useful to reduce the number of offerings the CloudStack administrator has to define. Instead of defining a compute offering for every imaginable combination of values that a user might want, the administrator can define offerings that provide some flexibility to the users and can serve as the basis for several different VM configurations.

A service offering includes the following elements:

- CPU, memory, and network resource guarantees
- How resources are metered
- How the resource usage is charged
- How often the charges are generated

For example, one service offering might allow users to create a virtual machine instance that is equivalent to a 1 GHz Intel® Core™ 2 CPU, with 1 GB memory at $0.20/hour, with network traffic metered at $0.10/GB.

CloudStack separates service offerings into compute offerings and disk offerings. The compute service offering specifies:

- Guest CPU (optional). If not defined by the CloudStack administrator, users can pick the CPU attributes.
- Guest RAM (optional). If not defined by the CloudStack administrator, users can pick the RAM.
- Guest Networking type (virtual or direct)
- Tags on the root disk

The disk offering specifies:

- Disk size (optional). If not defined by the CloudStack administrator, users can pick the disk size.
- Tags on the data disk

Custom Compute Offering

CloudStack provides you the flexibility to specify the desired values for the number of CPU, CPU speed, and memory while deploying a VM. As an admin, you create a Compute Offering by marking it as custom, and the users will be

able to customize this dynamic Compute Offering by specifying the memory, and CPU at the time of VM creation or upgrade. Custom Compute Offering is same as the normal Compute Offering except that the values of the dynamic parameters will be set to zeros in the given set of templates. Use this offering to deploy VM by specifying custom values for the dynamic parameters. Memory, CPU and number of CPUs are considered as dynamic parameters.

Dynamic Compute Offerings can be used in following cases: deploying a VM, changing the compute offering of a stopped VM and running VMs, which is nothing but scaling up. To support this feature a new field, Custom, has been added to the Create Compute Offering page. If the Custom field is checked, the user will be able to create a custom Compute Offering by filling in the desired values for number of CPU, CPU speed, and memory. See ? for more information on this.

Recording Usage Events for Dynamically Assigned Resources.

To support this feature, usage events has been enhanced to register events for dynamically assigned resources. Usage events are registered when a VM is created from a custom compute offering, and upon changing the compute offering of a stopped or running VM. The values of the parameters, such as CPU, speed, RAM are recorded.

Creating a New Compute Offering

To create a new compute offering:

1. Log in with admin privileges to the CloudStack UI.
2. In the left navigation bar, click Service Offerings.
3. In Select Offering, choose Compute Offering.
4. Click Add Compute Offering.
5. In the dialog, make the following choices:
    - **Name**: Any desired name for the service offering.
    - **Description**: A short description of the offering that can be displayed to users
    - **Storage type**: The type of disk that should be allocated. Local allocates from storage attached directly to the host where the system VM is running. Shared allocates from storage accessible via NFS.
    - **Custom**: Custom compute offerings can be used in following cases: deploying a VM, changing the compute offering of a stopped VM and running VMs, which is nothing but scaling up.

      If the Custom field is checked, the end-user must fill in the desired values for number of CPU, CPU speed, and RAM Memory when using a custom compute offering. When you check this box, those three input fields are hidden in the dialog box.
    - **# of CPU cores**: The number of cores which should be allocated to a system VM with this offering. If Custom is checked, this field does not appear.
    - **CPU (in MHz)**: The CPU speed of the cores that the system VM is allocated. For example, "2000" would provide for a 2 GHz clock. If Custom is checked, this field does not appear.
    - **Memory (in MB)**: The amount of memory in megabytes that the system VM should be allocated. For example, "2048" would provide for a 2 GB RAM allocation. If Custom is checked, this field does not appear.
    - **Network Rate**: Allowed data transfer rate in MB per second.
    - **Disk Read Rate**: Allowed disk read rate in bits per second.
    - **Disk Write Rate**: Allowed disk write rate in bits per second.
    - **Disk Read Rate**: Allowed disk read rate in IOPS (input/output operations per second).
    - **Disk Write Rate**: Allowed disk write rate in IOPS (input/output operations per second).

- **Offer HA**: If yes, the administrator can choose to have the system VM be monitored and as highly available as possible.
- **QoS Type**: Three options: Empty (no Quality of Service), hypervisor (rate limiting enforced on the hypervisor side), and storage (guaranteed minimum and maximum IOPS enforced on the storage side). If leveraging QoS, make sure that the hypervisor or storage system supports this feature.
- **Custom IOPS**: If checked, the user can set their own IOPS. If not checked, the root administrator can define values. If the root admin does not set values when using storage QoS, default values are used (the defauls can be overridden if the proper parameters are passed into CloudStack when creating the primary storage in question).
- **Min IOPS**: Appears only if storage QoS is to be used. Set a guaranteed minimum number of IOPS to be enforced on the storage side.
- **Max IOPS**: Appears only if storage QoS is to be used. Set a maximum number of IOPS to be enforced on the storage side (the system may go above this limit in certain circumstances for short intervals).
- **Hypervisor Snapshot Reserve**: For managed storage only. This is a value that is a percentage of the size of the root disk. For example: if the root disk is 20 GB and Hypervisor Snapshot Reserve is 200%, the storage volume that backs the storage repository (XenServer) or datastore (VMware) in question is sized at 60 GB (20 GB + (20 GB * 2)). This enables space for hypervisor snapshots in addition to the virtual disk that represents the root disk. This does not apply for KVM.
- **Storage Tags**: The tags that should be associated with the primary storage used by the system VM.
- **Host Tags**: (Optional) Any tags that you use to organize your hosts
- **CPU cap**: Whether to limit the level of CPU usage even if spare capacity is available.
- **Public**: Indicate whether the service offering should be available all domains or only some domains. Choose Yes to make it available to all domains. Choose No to limit the scope to a subdomain; CloudStack will then prompt for the subdomain's name.
- **isVolatile**: If checked, VMs created from this service offering will have their root disks reset upon reboot. This is useful for secure environments that need a fresh start on every boot and for desktops that should not retain state.
- **Deployment Planner**: Choose the technique that you would like CloudStack to use when deploying VMs based on this service offering.

  First Fit places new VMs on the first host that is found having sufficient capacity to support the VM's requirements.

  User Dispersing makes the best effort to evenly distribute VMs belonging to the same account on different clusters or pods.

  User Concentrated prefers to deploy VMs belonging to the same account within a single pod.

  Implicit Dedication will deploy VMs on private infrastructure that is dedicated to a specific domain or account. If you choose this planner, then you must also pick a value for Planner Mode. See "Dedicating Resources to Accounts and Domains".

  Bare Metal is used with bare metal hosts. See Bare Metal Installation in the Installation Guide.
- **Planner Mode**: Used when ImplicitDedicationPlanner is selected in the previous field. The planner mode determines how VMs will be deployed on private infrastructure that is dedicated to a single domain or account.

  Strict: A host will not be shared across multiple accounts. For example, strict implicit dedication is useful for deployment of certain types of applications, such as desktops, where no host can be shared between different accounts without violating the desktop software's terms of license.

Preferred: The VM will be deployed in dedicated infrastructure if possible. Otherwise, the VM can be deployed in shared infrastructure.

- **GPU: Assign a physical GPU(GPU-passthrough) or a portion of a physicalGPU** GPU card(vGPU) to the guest VM. It allows graphical applications to run on the VM. Select the card from the supported list of cards.

  The options given are NVIDIA GRID K1 and NVIDIA GRID K2. These are vGPU capable cards that allow multiple vGPUs on a single physical GPU. If you want to use a card other than these, follow the instructions in the **"GPU and vGPU support for CloudStack Guest VMs"** page in the Cloudstack Version 4.4 Design Docs found in the Cloudstack Wiki.

- **vGPU Type**: Represents the type of virtual GPU to be assigned to a guest VM. In this case, only a portion of a physical GPU card (vGPU) is assigned to the guest VM.

  Additionally, the **passthrough vGPU** type is defined to represent a physical GPU device. A **passthrough vGPU** can directly be assigned to a single guest VM. In this case, a physical GPU device is exclusively allotted to a single guest VM.

6. Click Add.

Creating a New Disk Offering

To create a new disk offering:

1. Log in with admin privileges to the CloudStack UI.
2. In the left navigation bar, click Service Offerings.
3. In Select Offering, choose Disk Offering.
4. Click Add Disk Offering.
5. In the dialog, make the following choices:
   - **Name**: Any desired name for the disk offering.
   - **Description**: A short description of the offering that can be displayed to users
   - **Custom Disk Size**: If checked, the user can set their own disk size. If not checked, the root administrator must define a value in Disk Size.
   - **Disk Size**: Appears only if Custom Disk Size is not selected. Define the volume size in GB (2^30 1GB = 1,073,741,824 Bytes).
   - **QoS Type**: Three options: Empty (no Quality of Service), hypervisor (rate limiting enforced on the hypervisor side), and storage (guaranteed minimum and maximum IOPS enforced on the storage side). If leveraging QoS, make sure that the hypervisor or storage system supports this feature.
   - **Custom IOPS**: If checked, the user can set their own IOPS. If not checked, the root administrator can define values. If the root admin does not set values when using storage QoS, default values are used (the defauls can be overridden if the proper parameters are passed into CloudStack when creating the primary storage in question).
   - **Min IOPS**: Appears only if storage QoS is to be used. Set a guaranteed minimum number of IOPS to be enforced on the storage side.
   - **Max IOPS**: Appears only if storage QoS is to be used. Set a maximum number of IOPS to be enforced on the storage side (the system may go above this limit in certain circumstances for short intervals).
   - **Hypervisor Snapshot Reserve**: For managed storage only. This is a value that is a percentage of the size of the data disk. For example: if the data disk is 20 GB and Hypervisor Snapshot Reserve is 200%, the storage volume that backs the storage repository (XenServer) or datastore (VMware) in question is sized

at 60 GB (20 GB + (20 GB * 2)). This enables space for hypervisor snapshots in addition to the virtual disk that represents the data disk. This does not apply for KVM.

- **(Optional)Storage Tags**: The tags that should be associated with the primary storage for this disk. Tags are a comma separated list of attributes of the storage. For example "ssd,blue". Tags are also added on Primary Storage. CloudStack matches tags on a disk offering to tags on the storage. If a tag is present on a disk offering that tag (or tags) must also be present on Primary Storage for the volume to be provisioned. If no such primary storage exists, allocation from the disk offering will fail..
- **Public**: Indicate whether the service offering should be available all domains or only some domains. Choose Yes to make it available to all domains. Choose No to limit the scope to a subdomain; CloudStack will then prompt for the subdomain's name.

6. Click Add.

Modifying or Deleting a Service Offering

Service offerings cannot be changed once created. This applies to both compute offerings and disk offerings.

A service offering can be deleted. If it is no longer in use, it is deleted immediately and permanently. If the service offering is still in use, it will remain in the database until all the virtual machines referencing it have been deleted. After deletion by the administrator, a service offering will not be available to end users that are creating new instances.

5.4.3 System Service Offerings

System service offerings provide a choice of CPU speed, number of CPUs, tags, and RAM size, just as other service offerings do. But rather than being used for virtual machine instances and exposed to users, system service offerings are used to change the default properties of virtual routers, console proxies, and other system VMs. System service offerings are visible only to the CloudStack root administrator. CloudStack provides default system service offerings. The CloudStack root administrator can create additional custom system service offerings.

When CloudStack creates a virtual router for a guest network, it uses default settings which are defined in the system service offering associated with the network offering. You can upgrade the capabilities of the virtual router by applying a new network offering that contains a different system service offering. All virtual routers in that network will begin using the settings from the new service offering.

Creating a New System Service Offering

To create a system service offering:

1. Log in with admin privileges to the CloudStack UI.
2. In the left navigation bar, click Service Offerings.
3. In Select Offering, choose System Offering.
4. Click Add System Service Offering.
5. In the dialog, make the following choices:
   - Name. Any desired name for the system offering.
   - Description. A short description of the offering that can be displayed to users
   - System VM Type. Select the type of system virtual machine that this offering is intended to support.
   - Storage type. The type of disk that should be allocated. Local allocates from storage attached directly to the host where the system VM is running. Shared allocates from storage accessible via NFS.

- # of CPU cores. The number of cores which should be allocated to a system VM with this offering
- CPU (in MHz). The CPU speed of the cores that the system VM is allocated. For example, "2000" would provide for a 2 GHz clock.
- Memory (in MB). The amount of memory in megabytes that the system VM should be allocated. For example, "2048" would provide for a 2 GB RAM allocation.
- Network Rate. Allowed data transfer rate in MB per second.
- Offer HA. If yes, the administrator can choose to have the system VM be monitored and as highly available as possible.
- Storage Tags. The tags that should be associated with the primary storage used by the system VM.
- Host Tags. (Optional) Any tags that you use to organize your hosts
- CPU cap. Whether to limit the level of CPU usage even if spare capacity is available.
- Public. Indicate whether the service offering should be available all domains or only some domains. Choose Yes to make it available to all domains. Choose No to limit the scope to a subdomain; CloudStack will then prompt for the subdomain's name.

6. Click Add.

5.4.4 Network Throttling

Network throttling is the process of controlling the network access and bandwidth usage based on certain rules. CloudStack controls this behaviour of the guest networks in the cloud by using the network rate parameter. This parameter is defined as the default data transfer rate in Mbps (Megabits Per Second) allowed in a guest network. It defines the upper limits for network utilization. If the current utilization is below the allowed upper limits, access is granted, else revoked.

You can throttle the network bandwidth either to control the usage above a certain limit for some accounts, or to control network congestion in a large cloud environment. The network rate for your cloud can be configured on the following:

- Network Offering
- Service Offering
- Global parameter

If network rate is set to NULL in service offering, the value provided in the vm.network.throttling.rate global parameter is applied. If the value is set to NULL for network offering, the value provided in the network.throttling.rate global parameter is considered.

For the default public, storage, and management networks, network rate is set to 0. This implies that the public, storage, and management networks will have unlimited bandwidth by default. For default guest networks, network rate is set to NULL. In this case, network rate is defaulted to the global parameter value.

The following table gives you an overview of how network rate is applied on different types of networks in CloudStack.

Networks	Network Rate Is Taken from
Guest network of Virtual Router	Guest Network Offering
Public network of Virtual Router	Guest Network Offering
Storage network of Secondary Storage VM	System Network Offering
Management network of Secondary Storage VM	System Network Offering
Storage network of Console Proxy VM	System Network Offering
Management network of Console Proxy VM	System Network Offering
Storage network of Virtual Router	System Network Offering
Management network of Virtual Router	System Network Offering
Public network of Secondary Storage VM	System Network Offering
Public network of Console Proxy VM	System Network Offering
Default network of a guest VM	Compute Offering
Additional networks of a guest VM	Corresponding Network Offerings

A guest VM must have a default network, and can also have many additional networks. Depending on various parameters, such as the host and virtual switch used, you can observe a difference in the network rate in your cloud. For example, on a VMware host the actual network rate varies based on where they are configured (compute offering, network offering, or both); the network type (shared or isolated); and traffic direction (ingress or egress).

The network rate set for a network offering used by a particular network in CloudStack is used for the traffic shaping policy of a port group, for example: port group A, for that network: a particular subnet or VLAN on the actual network. The virtual routers for that network connects to the port group A, and by default instances in that network connects to this port group. However, if an instance is deployed with a compute offering with the network rate set, and if this rate is used for the traffic shaping policy of another port group for the network, for example port group B, then instances using this compute offering are connected to the port group B, instead of connecting to port group A.

The traffic shaping policy on standard port groups in VMware only applies to the egress traffic, and the net effect depends on the type of network used in CloudStack. In shared networks, ingress traffic is unlimited for CloudStack, and egress traffic is limited to the rate that applies to the port group used by the instance if any. If the compute offering has a network rate configured, this rate applies to the egress traffic, otherwise the network rate set for the network offering applies. For isolated networks, the network rate set for the network offering, if any, effectively applies to the ingress traffic. This is mainly because the network rate set for the network offering applies to the egress traffic from the virtual router to the instance. The egress traffic is limited by the rate that applies to the port group used by the instance if any, similar to shared networks.

For example:

Network rate of network offering = 10 Mbps

Network rate of compute offering = 200 Mbps

In shared networks, ingress traffic will not be limited for CloudStack, while egress traffic will be limited to 200 Mbps. In an isolated network, ingress traffic will be limited to 10 Mbps and egress to 200 Mbps.

5.4.5 Changing the Default System Offering for System VMs

You can manually change the system offering for a particular System VM. Additionally, as a CloudStack administrator, you can also change the default system offering used for System VMs.

1. Create a new system offering.

   For more information, see Creating a New System Service Offering.

2. Back up the database:

```
mysqldump -u root -p cloud | bzip2 > cloud_backup.sql.bz2
```

3. Open an MySQL prompt:

```
mysql -u cloud -p cloud
```

4. Run the following queries on the cloud database.

   (a) In the disk_offering table, identify the original default offering and the new offering you want to use by default.

   Take a note of the ID of the new offering.

```
select id,name,unique_name,type from disk_offering;
```

   (b) For the original default offering, set the value of unique_name to NULL.

```
# update disk_offering set unique_name = NULL where id = 10;
```

   Ensure that you use the correct value for the ID.

   (c) For the new offering that you want to use by default, set the value of unique_name as follows:

   For the default Console Proxy VM (CPVM) offering,set unique_name to 'Cloud.com-ConsoleProxy'. For the default Secondary Storage VM (SSVM) offering, set unique_name to 'Cloud.com-SecondaryStorage'. For example:

```
update disk_offering set unique_name = 'Cloud.com-ConsoleProxy' where id = 16;
```

5. Restart CloudStack Management Server. Restarting is required because the default offerings are loaded into the memory at startup.

```
service cloudstack-management restart
```

6. Destroy the existing CPVM or SSVM offerings and wait for them to be recreated. The new CPVM or SSVM are configured with the new offering.

5.5 Setting up Networking for Users

5.5.1 Overview of Setting Up Networking for Users

People using cloud infrastructure have a variety of needs and preferences when it comes to the networking services provided by the cloud. As a CloudStack administrator, you can do the following things to set up networking for your users:

- Set up physical networks in zones
- Set up several different providers for the same service on a single physical network (for example, both Cisco and Juniper firewalls)
- Bundle different types of network services into network offerings, so users can choose the desired network services for any given virtual machine
- Add new network offerings as time goes on so end users can upgrade to a better class of service on their network
- Provide more ways for a network to be accessed by a user, such as through a project of which the user is a member

5.5.2 About Virtual Networks

A virtual network is a logical construct that enables multi-tenancy on a single physical network. In CloudStack a virtual network can be shared or isolated.

Isolated Networks

An isolated network can be accessed only by virtual machines of a single account. Isolated networks have the following properties.

- Resources such as VLAN are allocated and garbage collected dynamically
- There is one network offering for the entire network
- The network offering can be upgraded or downgraded but it is for the entire network

For more information, see "Configure Guest Traffic in an Advanced Zone".

Shared Networks

A shared network can be accessed by virtual machines that belong to many different accounts. Network Isolation on shared networks is accomplished by using techniques such as security groups, which is supported only in Basic zones in CloudStack 3.0.3 and later versions.

- Shared Networks are created by the administrator
- Shared Networks can be designated to a certain domain
- Shared Network resources such as VLAN and physical network that it maps to are designated by the administrator
- Shared Networks can be isolated by security groups
- Public Network is a shared network that is not shown to the end users
- Source NAT per zone is not supported in Shared Network when the service provider is virtual router. However, Source NAT per account is supported. For information, see "Configuring a Shared Guest Network".

Runtime Allocation of Virtual Network Resources

When you define a new virtual network, all your settings for that network are stored in CloudStack. The actual network resources are activated only when the first virtual machine starts in the network. When all virtual machines have left the virtual network, the network resources are garbage collected so they can be allocated again. This helps to conserve network resources.

5.5.3 Network Service Providers

Note: For the most up-to-date list of supported network service providers, see the CloudStack UI or call *listNetworkServiceProviders*.

A service provider (also called a network element) is hardware or virtual appliance that makes a network service possible; for example, a firewall appliance can be installed in the cloud to provide firewall service. On a single network, multiple providers can provide the same network service. For example, a firewall service may be provided by Cisco or Juniper devices in the same physical network.

You can have multiple instances of the same service provider in a network (say, more than one Juniper SRX device).

If different providers are set up to provide the same service on the network, the administrator can create network offerings so users can specify which network service provider they prefer (along with the other choices offered in network offerings). Otherwise, CloudStack will choose which provider to use whenever the service is called for.

Supported Network Service Providers

CloudStack ships with an internal list of the supported service providers, and you can choose from this list when creating a network offering.

	Virtual Router	Citrix NetScaler	Juniper SRX	F5 BigIP	Host based (KVM/Xen)
Remote Access VPN	Yes	No	No	No	No
DNS/DHCP/User Data	Yes	No	No	No	No
Firewall	Yes	No	Yes	No	No
Load Balancing	Yes	Yes	No	Yes	No
Elastic IP	No	Yes	No	No	No
Elastic LB	No	Yes	No	No	No
Source NAT	Yes	No	Yes	No	No
Static NAT	Yes	Yes	Yes	No	No
Port Forwarding	Yes	No	Yes	No	No

5.5.4 Network Offerings

Note: For the most up-to-date list of supported network services, see the CloudStack UI or call listNetworkServices.

A network offering is a named set of network services, such as:

- DHCP
- DNS
- Source NAT
- Static NAT
- Port Forwarding
- Load Balancing
- Firewall
- VPN
- (Optional) Name one of several available providers to use for a given service, such as Juniper for the firewall
- (Optional) Network tag to specify which physical network to use

When creating a new VM, the user chooses one of the available network offerings, and that determines which network services the VM can use.

The CloudStack administrator can create any number of custom network offerings, in addition to the default network offerings provided by CloudStack. By creating multiple custom network offerings, you can set up your cloud to offer different classes of service on a single multi-tenant physical network. For example, while the underlying physical wiring may be the same for two tenants, tenant A may only need simple firewall protection for their website, while

tenant B may be running a web server farm and require a scalable firewall solution, load balancing solution, and alternate networks for accessing the database backend.

Note: If you create load balancing rules while using a network service offering that includes an external load balancer device such as NetScaler, and later change the network service offering to one that uses the CloudStack virtual router, you must create a firewall rule on the virtual router for each of your existing load balancing rules so that they continue to function.

When creating a new virtual network, the CloudStack administrator chooses which network offering to enable for that network. Each virtual network is associated with one network offering. A virtual network can be upgraded or downgraded by changing its associated network offering. If you do this, be sure to reprogram the physical network to match.

CloudStack also has internal network offerings for use by CloudStack system VMs. These network offerings are not visible to users but can be modified by administrators.

Creating a New Network Offering

To create a network offering:

1. Log in with admin privileges to the CloudStack UI.
2. In the left navigation bar, click Service Offerings.
3. In Select Offering, choose Network Offering.
4. Click Add Network Offering.
5. In the dialog, make the following choices:
   - **Name**. Any desired name for the network offering.
   - **Description**. A short description of the offering that can be displayed to users.
   - **Network Rate**. Allowed data transfer rate in MB per second.
   - **Guest Type**. Choose whether the guest network is isolated or shared.

     For a description of this term, see *"About Virtual Networks"*.
   - **Persistent**. Indicate whether the guest network is persistent or not. The network that you can provision without having to deploy a VM on it is termed persistent network. For more information, see "Persistent Networks".
   - **Specify VLAN**. (Isolated guest networks only) Indicate whether a VLAN could be specified when this offering is used. If you select this option and later use this network offering while creating a VPC tier or an isolated network, you will be able to specify a VLAN ID for the network you create.
   - **VPC**. This option indicate whether the guest network is Virtual Private Cloud-enabled. A Virtual Private Cloud (VPC) is a private, isolated part of CloudStack. A VPC can have its own virtual network topology that resembles a traditional physical network. For more information on VPCs, see "About Virtual Private Clouds".
   - **Supported Services**. Select one or more of the possible network services. For some services, you must also choose the service provider; for example, if you select Load Balancer, you can choose the CloudStack virtual router or any other load balancers that have been configured in the cloud. Depending on which services you choose, additional fields may appear in the rest of the dialog box.

     Based on the guest network type selected, you can see the following supported services:

Supported Services	Description	Isolated	Shared
DHCP	For more information, see "DNS and DHCP".	Supported	Supported
DNS	For more information, see "DNS and DHCP".	Supported	Supported
Load Balancer	If you select Load Balancer, you can choose the CloudStack virtual router or any other load balancers that have been configured in the cloud.	Supported	Supported
Firewall	For more information, see the Administration Guide.	Supported	Supported
Source NAT	If you select Source NAT, you can choose the CloudStack virtual router or any other Source NAT providers that have been configured in the cloud.	Supported	Supported
Static NAT	If you select Static NAT, you can choose the CloudStack virtual router or any other Static NAT providers that have been configured in the cloud.	Supported	Supported
Port Forwarding	If you select Port Forwarding, you can choose the CloudStack virtual router or any other Port Forwarding providers that have been configured in the cloud.	Supported	Not Supported
VPN	For more information, see "Remote Access VPN".	Supported	Not Supported
User Data	For more information, see "User Data and Meta Data".	Not Supported	Supported
Network ACL	For more information, see "Configuring Network Access Control List".	Supported	Not Supported
Security Groups	For more information, see "Adding a Security Group".	Not Supported	Supported

- **System Offering**. If the service provider for any of the services selected in Supported Services is a virtual router, the System Offering field appears. Choose the system service offering that you want virtual routers to use in this network. For example, if you selected Load Balancer in Supported Services and selected a virtual router to provide load balancing, the System Offering field appears so you can choose between the CloudStack default system service offering and any custom system service offerings that have been defined by the CloudStack root administrator.

  For more information, see "System Service Offerings".

- **LB Isolation**: Specify what type of load balancer isolation you want for the network: Shared or Dedicated.

  Dedicated: If you select dedicated LB isolation, a dedicated load balancer device is assigned for the network from the pool of dedicated load balancer devices provisioned in the zone. If no sufficient dedicated load balancer devices are available in the zone, network creation fails. Dedicated device is a good choice for the high-traffic networks that make full use of the device's resources.

  Shared: If you select shared LB isolation, a shared load balancer device is assigned for the network from the pool of shared load balancer devices provisioned in the zone. While provisioning CloudStack picks the shared load balancer device that is used by the least number of accounts. Once the device reaches its maximum capacity, the device will not be allocated to a new account.

- **Mode**: You can select either Inline mode or Side by Side mode:

  Inline mode: Supported only for Juniper SRX firewall and BigF5 load balancer devices. In inline mode, a firewall device is placed in front of a load balancing device. The firewall acts as the gateway for all the incoming traffic, then redirect the load balancing traffic to the load balancer behind it. The load balancer in this case will not have the direct access to the public network.

  Side by Side: In side by side mode, a firewall device is deployed in parallel with the load balancer device. So the traffic to the load balancer public IP is not routed through the firewall, and therefore, is exposed to the public network.

- **Associate Public IP**: Select this option if you want to assign a public IP address to the VMs deployed in the guest network. This option is available only if

  - Guest network is shared.
  - StaticNAT is enabled.
  - Elastic IP is enabled.

  For information on Elastic IP, see "About Elastic IP".

- **Redundant router capability**: Available only when Virtual Router is selected as the Source NAT provider. Select this option if you want to use two virtual routers in the network for uninterrupted connection: one operating as the master virtual router and the other as the backup. The master virtual router receives requests from and sends responses to the user's VM. The backup virtual router is activated only when the master is down. After the failover, the backup becomes the master virtual router. CloudStack deploys the routers on different hosts to ensure reliability if one host is down.

- **Conserve mode**: Indicate whether to use conserve mode. In this mode, network resources are allocated only when the first virtual machine starts in the network. When conservative mode is off, the public IP can only be used for a single service. For example, a public IP used for a port forwarding rule cannot be used for defining other services, such as StaticNAT or load balancing. When the conserve mode is on, you can define more than one service on the same public IP.

  Note: If StaticNAT is enabled, irrespective of the status of the conserve mode, no port forwarding or load balancing rule can be created for the IP. However, you can add the firewall rules by using the createFirewallRule command.

- **Tags**: Network tag to specify which physical network to use.

- **Default egress policy**: Configure the default policy for firewall egress rules. Options are Allow and Deny. Default is Allow if no egress policy is specified, which indicates that all the egress traffic is accepted when a guest network is created from this offering.

  To block the egress traffic for a guest network, select Deny. In this case, when you configure an egress rules for an isolated guest network, rules are added to allow the specified traffic.

6. Click Add.

5.5.5 Configuring AutoScale without using NetScaler

Warning: This feature is currently only available on the master branch and will be released in the 4.4 release.

What is AutoScaling?

AutoScaling allows you to scale your back-end services or application VMs up or down seamlessly and automatically according to the conditions you define. With AutoScaling enabled, you can ensure that the number of VMs you are using seamlessly scale up when demand increases, and automatically decreases when demand subsides. Thus it helps you save compute costs by terminating underused VMs automatically and launching new VMs when you need them, without the need for manual intervention.

Hypervisor support

At that time, AutoScaling without NetScaler only supports for Xenserver. We are working to support KVM also.

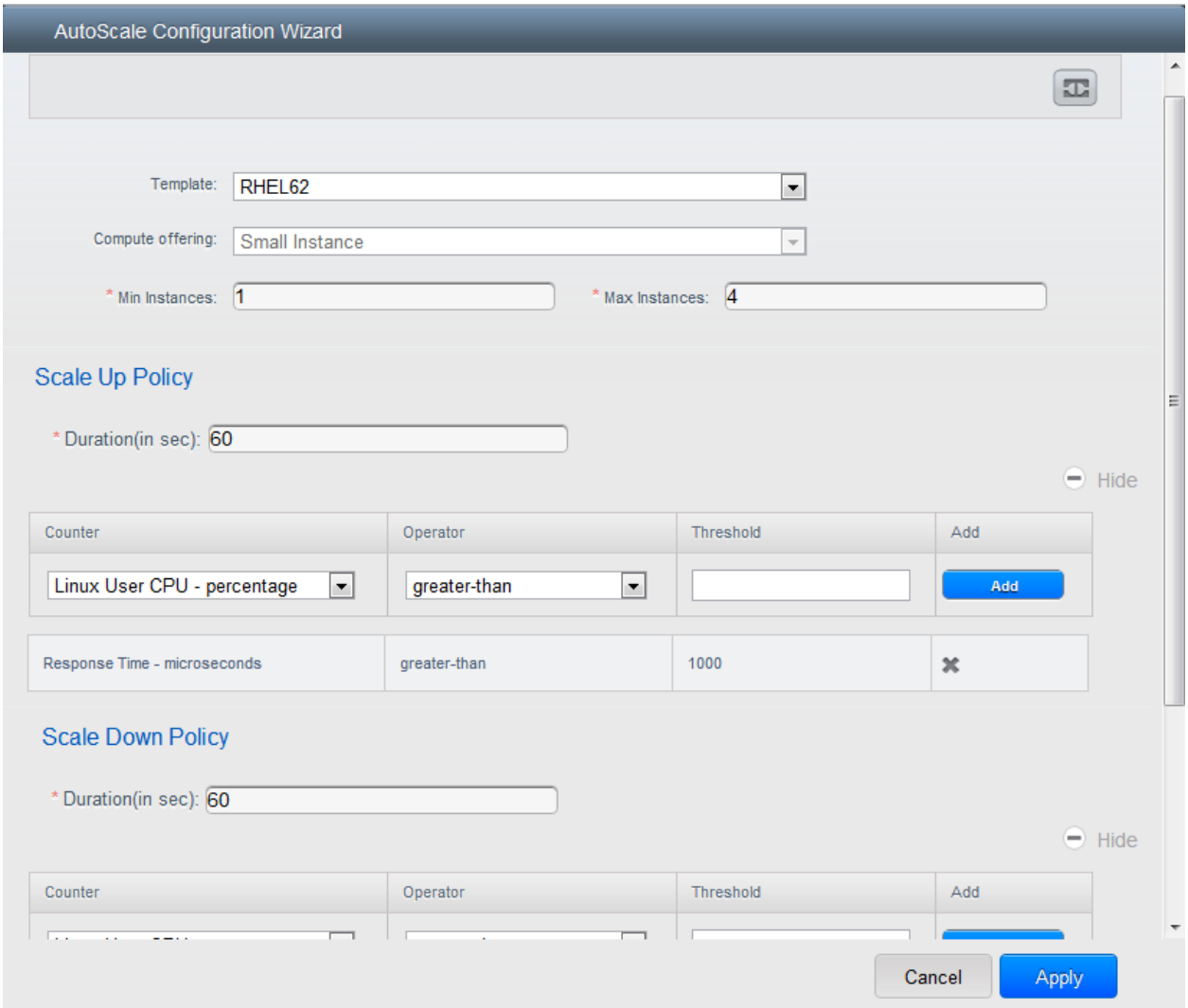
Prerequisites

Before you configure an AutoScale rule, consider the following:

- Ensure that the necessary template is prepared before configuring AutoScale. Firstly you must install the PV-driver, which helps Xenserver collect performance parameters (CPU and memory) into VMs. Beside, When a VM is deployed by using a template and when it comes up, the application should be up and running.

Configuration

Specify the following:



- Template: A template consists of a base OS image and application. A template is used to provision the new instance of an application on a scaleup action. When a VM is deployed from a template, the VM can start taking the traffic from the load balancer without any admin intervention. For example, if the VM is deployed for a Web service, it should have the Web server running, the database connected, and so on.
- Compute offering: A predefined set of virtual hardware attributes, including CPU speed, number of CPUs, and RAM size, that the user can select when creating a new virtual machine instance. Choose one of the compute offerings to be used while provisioning a VM instance as part of scaleup action.
- Min Instance: The minimum number of active VM instances that is assigned to a load balancing rule. The active VM instances are the application instances that are up and serving the traffic, and are being load balanced. This parameter ensures that a load balancing rule has at least the configured number of active VM instances are available to serve the traffic.
- Max Instance: Maximum number of active VM instances that should be assigned to a load balancing rule. This parameter defines the upper limit of active VM instances that can be assigned to a load balancing rule.

  Specifying a large value for the maximum instance parameter might result in provisioning large number of VM instances, which in turn leads to a single load balancing rule exhausting the VM instances limit specified at the account or domain level.

Specify the following scale-up and scale-down policies:

- Duration: The duration, in seconds, for which the conditions you specify must be true to trigger a scaleup action. The conditions defined should hold true for the entire duration you specify for an AutoScale action to be invoked.
- Counter: The performance counters expose the state of the monitored instances. We added two new counter to work with that feature:
  - Linux User CPU [native] - percentage
  - Linux User RAM [native] - percentage

  Remember to choose one of them. If you choose anything else, the autoscaling will not work.
- Operator: The following five relational operators are supported in AutoScale feature: Greater than, Less than, Less than or equal to, Greater than or equal to, and Equal to.
- Threshold: Threshold value to be used for the counter. Once the counter defined above breaches the threshold value, the AutoScale feature initiates a scaleup or scaledown action.
- Add: Click Add to add the condition.

  Additionally, if you want to configure the advanced settings, click Show advanced settings, and specify the following:
- Polling interval: Frequency in which the conditions, combination of counter, operator and threshold, are to be evaluated before taking a scale up or down action. The default polling interval is 30 seconds.
- Quiet Time: This is the cool down period after an AutoScale action is initiated. The time includes the time taken to complete provisioning a VM instance from its template and the time taken by an application to be ready to serve traffic. This quiet time allows the fleet to come up to a stable state before any action can take place. The default is 300 seconds.
- Destroy VM Grace Period: The duration in seconds, after a scaledown action is initiated, to wait before the VM is destroyed as part of scaledown action. This is to ensure graceful close of any pending sessions or transactions being served by the VM marked for destroy. The default is 120 seconds.
- Apply: Click Apply to create the AutoScale configuration.

Disabling and Enabling an AutoScale Configuration

If you want to perform any maintenance operation on the AutoScale VM instances, disable the AutoScale configuration. When the AutoScale configuration is disabled, no scaleup or scaledown action is performed. You can use this downtime for the maintenance activities. To disable the AutoScale configuration, click the Disable AutoScale button.

The button toggles between enable and disable, depending on whether AutoScale is currently enabled or not. After the maintenance operations are done, you can enable the AutoScale configuration back. To enable, open the AutoScale configuration page again, then click the Enable AutoScale button.

Updating an AutoScale Configuration

You can update the various parameters and add or delete the conditions in a scaleup or scaledown rule. Before you update an AutoScale configuration, ensure that you disable the AutoScale load balancer rule by clicking the Disable AutoScale button.

After you modify the required AutoScale parameters, click Apply. To apply the new AutoScale policies, open the AutoScale configuration page again, then click the Enable AutoScale button.

Runtime Considerations

An administrator should not assign a VM to a load balancing rule which is configured for AutoScale.

Making API calls outside the context of AutoScale, such as destroyVM, on an autoscaled VM leaves the load balancing configuration in an inconsistent state. Though VM is destroyed from the load balancer rule, it continues be showed as a service assigned to a rule inside the context of AutoScale.

5.6 Working with Virtual Machines

5.6.1 Working with Virtual Machines

About Working with Virtual Machines

CloudStack provides administrators with complete control over the lifecycle of all guest VMs executing in the cloud. CloudStack provides several guest management operations for end users and administrators. VMs may be stopped, started, rebooted, and destroyed.

Guest VMs have a name and group. VM names and groups are opaque to CloudStack and are available for end users to organize their VMs. Each VM can have three names for use in different contexts. Only two of these names can be controlled by the user:

- Instance name – a unique, immutable ID that is generated by CloudStack and can not be modified by the user. This name conforms to the requirements in IETF RFC 1123.
- Display name – the name displayed in the CloudStack web UI. Can be set by the user. Defaults to instance name.
- Name – host name that the DHCP server assigns to the VM. Can be set by the user. Defaults to instance name

Note: You can append the display name of a guest VM to its internal name. For more information, see *"Appending a Display Name to the Guest VM's Internal Name"*.

Guest VMs can be configured to be Highly Available (HA). An HA-enabled VM is monitored by the system. If the system detects that the VM is down, it will attempt to restart the VM, possibly on a different host. For more information, see HA-Enabled Virtual Machines on

Each new VM is allocated one public IP address. When the VM is started, CloudStack automatically creates a static NAT between this public IP address and the private IP address of the VM.

If elastic IP is in use (with the NetScaler load balancer), the IP address initially allocated to the new VM is not marked as elastic. The user must replace the automatically configured IP with a specifically acquired elastic IP, and set up the static NAT mapping between this new IP and the guest VM's private IP. The VM's original IP address is then released and returned to the pool of available public IPs. Optionally, you can also decide not to allocate a public IP to a VM in an EIP-enabled Basic zone. For more information on Elastic IP, see "About Elastic IP".

CloudStack cannot distinguish a guest VM that was shut down by the user (such as with the "shutdown" command in Linux) from a VM that shut down unexpectedly. If an HA-enabled VM is shut down from inside the VM, CloudStack will restart it. To shut down an HA-enabled VM, you must go through the CloudStack UI or API.

Best Practices for Virtual Machines

For VMs to work as expected and provide excellent service, follow these guidelines.

Monitor VMs for Max Capacity

The CloudStack administrator should monitor the total number of VM instances in each cluster, and disable allocation to the cluster if the total is approaching the maximum that the hypervisor can handle. Be sure to leave a safety margin to allow for the possibility of one or more hosts failing, which would increase the VM load on the other hosts as the VMs are automatically redeployed. Consult the documentation for your chosen hypervisor to find the maximum permitted number of VMs per host, then use CloudStack global configuration settings to set this as the default limit. Monitor the VM activity in each cluster at all times. Keep the total number of VMs below a safe level that allows for the occasional host failure. For example, if there are N hosts in the cluster, and you want to allow for one host in the cluster to be down at any given time, the total number of VM instances you can permit in the cluster is at most (N-1) * (per-host-limit). Once a cluster reaches this number of VMs, use the CloudStack UI to disable allocation of more VMs to the cluster.

Install Required Tools and Drivers

Be sure the following are installed on each VM:

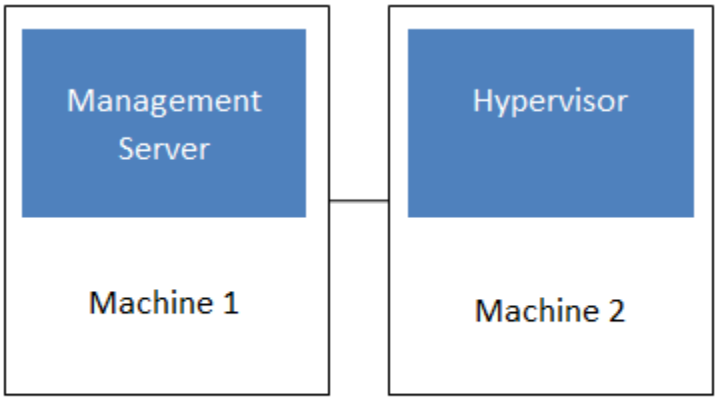
- For XenServer, install PV drivers and Xen tools on each VM. This will enable live migration and clean guest shutdown. Xen tools are required in order for dynamic CPU and RAM scaling to work.
- For vSphere, install VMware Tools on each VM. This will enable console view to work properly. VMware Tools are required in order for dynamic CPU and RAM scaling to work.

To be sure that Xen tools or VMware Tools is installed, use one of the following techniques:

- Create each VM from a template that already has the tools installed; or,
- When registering a new template, the administrator or user can indicate whether tools are installed on the template. This can be done through the UI or using the updateTemplate API; or,
- If a user deploys a virtual machine with a template that does not have Xen tools or VMware Tools, and later installs the tools on the VM, then the user can inform CloudStack using the updateVirtualMachine API. After installing the tools and updating the virtual machine, stop and start the VM.

VM Lifecycle

Virtual machines can be in the following states:



Simplified view of a basic deployment

Once a virtual machine is destroyed, it cannot be recovered. All the resources used by the virtual machine will be reclaimed by the system. This includes the virtual machine's IP address.

A stop will attempt to gracefully shut down the operating system, which typically involves terminating all the running applications. If the operation system cannot be stopped, it will be forcefully terminated. This has the same effect as pulling the power cord to a physical machine.

A reboot is a stop followed by a start.

CloudStack preserves the state of the virtual machine hard disk until the machine is destroyed.

A running virtual machine may fail because of hardware or network issues. A failed virtual machine is in the down state.

The system places the virtual machine into the down state if it does not receive the heartbeat from the hypervisor for three minutes.

The user can manually restart the virtual machine from the down state.

The system will start the virtual machine from the down state automatically if the virtual machine is marked as HA-enabled.

Creating VMs

Virtual machines are usually created from a template. Users can also create blank virtual machines. A blank virtual machine is a virtual machine without an OS template. Users can attach an ISO file and install the OS from the CD/DVD-ROM.

Note: You can create a VM without starting it. You can determine whether the VM needs to be started as part of the VM deployment. A request parameter, startVM, in the deployVm API provides this feature. For more information, see the Developer's Guide.

To create a VM from a template:

1. Log in to the CloudStack UI as an administrator or user.
2. In the left navigation bar, click Instances.
3. Click Add Instance.
4. Select a zone.
5. Select a template, then follow the steps in the wizard. For more information about how the templates came to be in this list, see *Working with Templates*.
6. Be sure that the hardware you have allows starting the selected service offering.
7. Click Submit and your VM will be created and started.

   Note: For security reason, the internal name of the VM is visible only to the root admin.

To create a VM from an ISO:

Note: (XenServer) Windows VMs running on XenServer require PV drivers, which may be provided in the template or added after the VM is created. The PV drivers are necessary for essential management functions such as mounting additional volumes and ISO images, live migration, and graceful shutdown.

1. Log in to the CloudStack UI as an administrator or user.
2. In the left navigation bar, click Instances.
3. Click Add Instance.
4. Select a zone.
5. Select ISO Boot, and follow the steps in the wizard.
6. Click Submit and your VM will be created and started.

Accessing VMs

Any user can access their own virtual machines. The administrator can access all VMs running in the cloud.

To access a VM through the CloudStack UI:

1. Log in to the CloudStack UI as a user or admin.
2. Click Instances, then click the name of a running VM.
3. Click the View Console button .

To access a VM directly over the network:

1. The VM must have some port open to incoming traffic. For example, in a basic zone, a new VM might be assigned to a security group which allows incoming traffic. This depends on what security group you picked when creating the VM. In other cases, you can open a port by setting up a port forwarding policy. See "IP Forwarding and Firewalling".
2. If a port is open but you can not access the VM using ssh, it's possible that ssh is not already enabled on the VM. This will depend on whether ssh is enabled in the template you picked when creating the VM. Access the VM through the CloudStack UI and enable ssh on the machine using the commands for the VM's operating system.
3. If the network has an external firewall device, you will need to create a firewall rule to allow access. See "IP Forwarding and Firewalling".

Stopping and Starting VMs

Once a VM instance is created, you can stop, restart, or delete it as needed. In the CloudStack UI, click Instances, select the VM, and use the Stop, Start, Reboot, and Destroy buttons.

Assigning VMs to Hosts

At any point in time, each virtual machine instance is running on a single host. How does CloudStack determine which host to place a VM on? There are several ways:

- Automatic default host allocation. CloudStack can automatically pick the most appropriate host to run each virtual machine.
- Instance type preferences. CloudStack administrators can specify that certain hosts should have a preference for particular types of guest instances. For example, an administrator could state that a host should have a preference to run Windows guests. The default host allocator will attempt to place guests of that OS type on such hosts first. If no such host is available, the allocator will place the instance wherever there is sufficient physical capacity.
- Vertical and horizontal allocation. Vertical allocation consumes all the resources of a given host before allocating any guests on a second host. This reduces power consumption in the cloud. Horizontal allocation places a guest on each host in a round-robin fashion. This may yield better performance to the guests in some cases.

- End user preferences. Users can not control exactly which host will run a given VM instance, but they can specify a zone for the VM. CloudStack is then restricted to allocating the VM only to one of the hosts in that zone.
- Host tags. The administrator can assign tags to hosts. These tags can be used to specify which host a VM should use. The CloudStack administrator decides whether to define host tags, then create a service offering using those tags and offer it to the user.
- Affinity groups. By defining affinity groups and assigning VMs to them, the user or administrator can influence (but not dictate) which VMs should run on separate hosts. This feature is to let users specify that certain VMs won't be on the same host.
- CloudStack also provides a pluggable interface for adding new allocators. These custom allocators can provide any policy the administrator desires.

Affinity Groups

By defining affinity groups and assigning VMs to them, the user or administrator can influence (but not dictate) which VMs should run on separate hosts. This feature is to let users specify that VMs with the same "host anti-affinity" type won't be on the same host. This serves to increase fault tolerance. If a host fails, another VM offering the same service (for example, hosting the user's website) is still up and running on another host.

The scope of an affinity group is per user account.

Creating a New Affinity Group

To add an affinity group:

1. Log in to the CloudStack UI as an administrator or user.
2. In the left navigation bar, click Affinity Groups.
3. Click Add affinity group. In the dialog box, fill in the following fields:
    - Name. Give the group a name.
    - Description. Any desired text to tell more about the purpose of the group.
    - Type. The only supported type shipped with CloudStack is Host Anti-Affinity. This indicates that the VMs in this group should avoid being placed on the same host with each other. If you see other types in this list, it means that your installation of CloudStack has been extended with customized affinity group plugins.

Assign a New VM to an Affinity Group

To assign a new VM to an affinity group:

- Create the VM as usual, as described in "Creating VMs". In the Add Instance wizard, there is a new Affinity tab where you can select the affinity group.

Change Affinity Group for an Existing VM

To assign an existing VM to an affinity group:

1. Log in to the CloudStack UI as an administrator or user.
2. In the left navigation bar, click Instances.

3. Click the name of the VM you want to work with.
4. Stop the VM by clicking the Stop button.
5. Click the Change Affinity button.

View Members of an Affinity Group

To see which VMs are currently assigned to a particular affinity group:

1. In the left navigation bar, click Affinity Groups.
2. Click the name of the group you are interested in.
3. Click View Instances. The members of the group are listed.

   From here, you can click the name of any VM in the list to access all its details and controls.

Delete an Affinity Group

To delete an affinity group:

1. In the left navigation bar, click Affinity Groups.
2. Click the name of the group you are interested in.
3. Click Delete.

   Any VM that is a member of the affinity group will be disassociated from the group. The former group members will continue to run normally on the current hosts, but if the VM is restarted, it will no longer follow the host allocation rules from its former affinity group.

Virtual Machine Snapshots

(Supported on VMware and XenServer)

In addition to the existing CloudStack ability to snapshot individual VM volumes, you can take a VM snapshot to preserve all the VM's data volumes as well as (optionally) its CPU/memory state. This is useful for quick restore of a VM. For example, you can snapshot a VM, then make changes such as software upgrades. If anything goes wrong, simply restore the VM to its previous state using the previously saved VM snapshot.

The snapshot is created using the hypervisor's native snapshot facility. The VM snapshot includes not only the data volumes, but optionally also whether the VM is running or turned off (CPU state) and the memory contents. The snapshot is stored in CloudStack's primary storage.

VM snapshots can have a parent/child relationship. Each successive snapshot of the same VM is the child of the snapshot that came before it. Each time you take an additional snapshot of the same VM, it saves only the differences between the current state of the VM and the state stored in the most recent previous snapshot. The previous snapshot becomes a parent, and the new snapshot is its child. It is possible to create a long chain of these parent/child snapshots, which amount to a "redo" record leading from the current state of the VM back to the original.

If you need more information about VM snapshots on VMware, check out the VMware documentation and the VMware Knowledge Base, especially Understanding virtual machine snapshots.

Limitations on VM Snapshots

- If a VM has some stored snapshots, you can't attach new volume to the VM or delete any existing volumes. If you change the volumes on the VM, it would become impossible to restore the VM snapshot which was created with the previous volume structure. If you want to attach a volume to such a VM, first delete its snapshots.
- VM snapshots which include both data volumes and memory can't be kept if you change the VM's service offering. Any existing VM snapshots of this type will be discarded.
- You can't make a VM snapshot at the same time as you are taking a volume snapshot.
- You should use only CloudStack to create VM snapshots on hosts managed by CloudStack. Any snapshots that you make directly on the hypervisor will not be tracked in CloudStack.

Configuring VM Snapshots

The cloud administrator can use global configuration variables to control the behavior of VM snapshots. To set these variables, go through the Global Settings area of the CloudStack UI.

Configuration Setting Name

Description

vmsnapshots.max

The maximum number of VM snapshots that can be saved for any given virtual machine in the cloud. The total possible number of VM snapshots in the cloud is (number of VMs) * vmsnapshots.max. If the number of snapshots for any VM ever hits the maximum, the older ones are removed by the snapshot expunge job.

vmsnapshot.create.wait

Number of seconds to wait for a snapshot job to succeed before declaring failure and issuing an error.

Using VM Snapshots

To create a VM snapshot using the CloudStack UI:

1. Log in to the CloudStack UI as a user or administrator.
2. Click Instances.
3. Click the name of the VM you want to snapshot.
4. Click the Take VM Snapshot button.

   Note: If a snapshot is already in progress, then clicking this button will have no effect.

5. Provide a name and description. These will be displayed in the VM Snapshots list.
6. (For running VMs only) If you want to include the VM's memory in the snapshot, click the Memory checkbox. This saves the CPU and memory state of the virtual machine. If you don't check this box, then only the current state of the VM disk is saved. Checking this box makes the snapshot take longer.
7. Quiesce VM: check this box if you want to quiesce the file system on the VM before taking the snapshot. Not supported on XenServer when used with CloudStack-provided primary storage.

When this option is used with CloudStack-provided primary storage, the quiesce operation is performed by the underlying hypervisor (VMware is supported). When used with another primary storage vendor's plugin, the quiesce operation is provided according to the vendor's implementation.

8. Click OK.

To delete a snapshot or restore a VM to the state saved in a particular snapshot:

1. Navigate to the VM as described in the earlier steps.
2. Click View VM Snapshots.
3. In the list of snapshots, click the name of the snapshot you want to work with.
4. Depending on what you want to do:

   To delete the snapshot, click the Delete button.

   To revert to the snapshot, click the Revert button.

Note: VM snapshots are deleted automatically when a VM is destroyed. You don't have to manually delete the snapshots in this case.

Changing the VM Name, OS, or Group

After a VM is created, you can modify the display name, operating system, and the group it belongs to.

To access a VM through the CloudStack UI:

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation, click Instances.
3. Select the VM that you want to modify.
4. Click the Stop button to stop the VM.
5. Click Edit.
6. Make the desired changes to the following:
7. **Display name**: Enter a new display name if you want to change the name of the VM.
8. **OS Type**: Select the desired operating system.
9. **Group**: Enter the group name for the VM.
10. Click Apply.

Appending a Display Name to the Guest VM's Internal Name

Every guest VM has an internal name. The host uses the internal name to identify the guest VMs. CloudStack gives you an option to provide a guest VM with a display name. You can set this display name as the internal name so that the vCenter can use it to identify the guest VM. A new global parameter, vm.instancename.flag, has now been added to achieve this functionality.

The default format of the internal name is i-<user_id>-<vm_id>-<instance.name>, where instance.name is a global parameter. However, If vm.instancename.flag is set to true, and if a display name is provided during the creation of a

guest VM, the display name is appended to the internal name of the guest VM on the host. This makes the internal name format as i-<user_id>-<vm_id>-<displayName>. The default value of vm.instancename.flag is set to false. This feature is intended to make the correlation between instance names and internal names easier in large data center deployments.

The following table explains how a VM name is displayed in different scenarios.

User-Provided Display Name	vm.instancename.flag	Hostname on the VM	Name on vCenter	Internal Name
Yes	True	Display name	i-<user_id>-<vm_id>-displayName	i-<user_id>-<vm_id>-displayName
No	True	UUID	i-<user_id>-<vm_id>-<instance.name>	i-<user_id>-<vm_id>-<instance.name>
Yes	False	Display name	i-<user_id>-<vm_id>-<instance.name>	i-<user_id>-<vm_id>-<instance.name>
No	False	UUID	i-<user_id>-<vm_id>-<instance.name>	i-<user_id>-<vm_id>-<instance.name>

Changing the Service Offering for a VM

To upgrade or downgrade the level of compute resources available to a virtual machine, you can change the VM's compute offering.

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation, click Instances.
3. Choose the VM that you want to work with.
4. (Skip this step if you have enabled dynamic VM scaling; see *CPU and Memory Scaling for Running VMs*.)

   Click the Stop button to stop the VM.
5. Click the Change Service button.

   The Change service dialog box is displayed.
6. Select the offering you want to apply to the selected VM.
7. Click OK.

CPU and Memory Scaling for Running VMs

(Supported on VMware and XenServer)

It is not always possible to accurately predict the CPU and RAM requirements when you first deploy a VM. You might need to increase these resources at any time during the life of a VM. You can dynamically modify CPU and RAM levels to scale up these resources for a running VM without incurring any downtime.

Dynamic CPU and RAM scaling can be used in the following cases:

- User VMs on hosts running VMware and XenServer.
- System VMs on VMware.
- VMware Tools or XenServer Tools must be installed on the virtual machine.

- The new requested CPU and RAM values must be within the constraints allowed by the hypervisor and the VM operating system.
- New VMs that are created after the installation of CloudStack 4.2 can use the dynamic scaling feature. If you are upgrading from a previous version of CloudStack, your existing VMs created with previous versions will not have the dynamic scaling capability unless you update them using the following procedure.

Updating Existing VMs

If you are upgrading from a previous version of CloudStack, and you want your existing VMs created with previous versions to have the dynamic scaling capability, update the VMs using the following steps:

1. Make sure the zone-level setting enable.dynamic.scale.vm is set to true. In the left navigation bar of the CloudStack UI, click Infrastructure, then click Zones, click the zone you want, and click the Settings tab.
2. Install Xen tools (for XenServer hosts) or VMware Tools (for VMware hosts) on each VM if they are not already installed.
3. Stop the VM.
4. Click the Edit button.
5. Click the Dynamically Scalable checkbox.
6. Click Apply.
7. Restart the VM.

Configuring Dynamic CPU and RAM Scaling

To configure this feature, use the following new global configuration variables:

- enable.dynamic.scale.vm: Set to True to enable the feature. By default, the feature is turned off.
- scale.retry: How many times to attempt the scaling operation. Default = 2.

How to Dynamically Scale CPU and RAM

To modify the CPU and/or RAM capacity of a virtual machine, you need to change the compute offering of the VM to a new compute offering that has the desired CPU and RAM values. You can use the same steps described above in *"Changing the Service Offering for a VM"*, but skip the step where you stop the virtual machine. Of course, you might have to create a new compute offering first.

When you submit a dynamic scaling request, the resources will be scaled up on the current host if possible. If the host does not have enough resources, the VM will be live migrated to another host in the same cluster. If there is no host in the cluster that can fulfill the requested level of CPU and RAM, the scaling operation will fail. The VM will continue to run as it was before.

Limitations

- You can not do dynamic scaling for system VMs on XenServer.
- CloudStack will not check to be sure that the new CPU and RAM levels are compatible with the OS running on the VM.

- When scaling memory or CPU for a Linux VM on VMware, you might need to run scripts in addition to the other steps mentioned above. For more information, see Hot adding memory in Linux (1012764) in the VMware Knowledge Base.
- (VMware) If resources are not available on the current host, scaling up will fail on VMware because of a known issue where CloudStack and vCenter calculate the available capacity differently. For more information, see https://issues.apache.org/jira/browse/CLOUDSTACK-1809.
- On VMs running Linux 64-bit and Windows 7 32-bit operating systems, if the VM is initially assigned a RAM of less than 3 GB, it can be dynamically scaled up to 3 GB, but not more. This is due to a known issue with these operating systems, which will freeze if an attempt is made to dynamically scale from less than 3 GB to more than 3 GB.

Resetting the Virtual Machine Root Volume on Reboot

For secure environments, and to ensure that VM state is not persisted across reboots, you can reset the root disk. For more information, see "Reset VM to New Root Disk on Reboot".

Moving VMs Between Hosts (Manual Live Migration)

The CloudStack administrator can move a running VM from one host to another without interrupting service to users or going into maintenance mode. This is called manual live migration, and can be done under the following conditions:

- The root administrator is logged in. Domain admins and users can not perform manual live migration of VMs.
- The VM is running. Stopped VMs can not be live migrated.
- The destination host must have enough available capacity. If not, the VM will remain in the "migrating" state until memory becomes available.
- (KVM) The VM must not be using local disk storage. (On XenServer and VMware, VM live migration with local disk is enabled by CloudStack support for XenMotion and vMotion.)
- (KVM) The destination host must be in the same cluster as the original host. (On XenServer and VMware, VM live migration from one cluster to another is enabled by CloudStack support for XenMotion and vMotion.)

To manually live migrate a virtual machine

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation, click Instances.
3. Choose the VM that you want to migrate.
4. Click the Migrate Instance button.
5. From the list of suitable hosts, choose the one to which you want to move the VM.

   Note: If the VM's storage has to be migrated along with the VM, this will be noted in the host list. CloudStack will take care of the storage migration for you.

6. Click OK.

Deleting VMs

Users can delete their own virtual machines. A running virtual machine will be abruptly stopped before it is deleted. Administrators can delete any virtual machines.

To delete a virtual machine:

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation, click Instances.
3. Choose the VM that you want to delete.
4. Click the Destroy Instance button.

Working with ISOs

CloudStack supports ISOs and their attachment to guest VMs. An ISO is a read-only file that has an ISO/CD-ROM style file system. Users can upload their own ISOs and mount them on their guest VMs.

ISOs are uploaded based on a URL. HTTP is the supported protocol. Once the ISO is available via HTTP specify an upload URL such as http://my.web.server/filename.iso.

ISOs may be public or private, like templates.ISOs are not hypervisor-specific. That is, a guest on vSphere can mount the exact same image that a guest on KVM can mount.

ISO images may be stored in the system and made available with a privacy level similar to templates. ISO images are classified as either bootable or not bootable. A bootable ISO image is one that contains an OS image. CloudStack allows a user to boot a guest VM off of an ISO image. Users can also attach ISO images to guest VMs. For example, this enables installing PV drivers into Windows. ISO images are not hypervisor-specific.

Adding an ISO

To make additional operating system or other software available for use with guest VMs, you can add an ISO. The ISO is typically thought of as an operating system image, but you can also add ISOs for other types of software, such as desktop applications that you want to be installed as part of a template.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation bar, click Templates.
3. In Select View, choose ISOs.
4. Click Add ISO.
5. In the Add ISO screen, provide the following:
   - **Name**: Short name for the ISO image. For example, CentOS 6.2 64-bit.
   - **Description**: Display test for the ISO image. For example, CentOS 6.2 64-bit.
   - **URL**: The URL that hosts the ISO image. The Management Server must be able to access this location via HTTP. If needed you can place the ISO image directly on the Management Server
   - **Zone**: Choose the zone where you want the ISO to be available, or All Zones to make it available throughout CloudStack.
   - **Bootable**: Whether or not a guest could boot off this ISO image. For example, a CentOS ISO is bootable, a Microsoft Office ISO is not bootable.

- **OS Type**: This helps CloudStack and the hypervisor perform certain operations and make assumptions that improve the performance of the guest. Select one of the following.
  - If the operating system of your desired ISO image is listed, choose it.
  - If the OS Type of the ISO is not listed or if the ISO is not bootable, choose Other.
  - (XenServer only) If you want to boot from this ISO in PV mode, choose Other PV (32-bit) or Other PV (64-bit)
  - (KVM only) If you choose an OS that is PV-enabled, the VMs created from this ISO will have a SCSI (virtio) root disk. If the OS is not PV-enabled, the VMs will have an IDE root disk. The PV-enabled types are:
    * Fedora 13
    * Fedora 12
    * Fedora 11
    * Fedora 10
    * Fedora 9
    * Other PV
    * Debian GNU/Linux
    * CentOS 5.3
    * CentOS 5.4
    * CentOS 5.5
    * Red Hat Enterprise Linux 5.3
    * Red Hat Enterprise Linux 5.4
    * Red Hat Enterprise Linux 5.5
    * Red Hat Enterprise Linux 6

  Note: It is not recommended to choose an older version of the OS than the version in the image. For example, choosing CentOS 5.4 to support a CentOS 6.2 image will usually not work. In these cases, choose Other.

- **Extractable**: Choose Yes if the ISO should be available for extraction.
- **Public**: Choose Yes if this ISO should be available to other users.
- **Featured**: Choose Yes if you would like this ISO to be more prominent for users to select. The ISO will appear in the Featured ISOs list. Only an administrator can make an ISO Featured.

6. Click OK.

   The Management Server will download the ISO. Depending on the size of the ISO, this may take a long time. The ISO status column will display Ready once it has been successfully downloaded into secondary storage. Clicking Refresh updates the download percentage.

7. **Important**: Wait for the ISO to finish downloading. If you move on to the next task and try to use the ISO right away, it will appear to fail. The entire ISO must be available before CloudStack can work with it.

Attaching an ISO to a VM

1. In the left navigation, click Instances.
2. Choose the virtual machine you want to work with.
3. Click the Attach ISO button.
4. In the Attach ISO dialog box, select the desired ISO.
5. Click OK.

Changing a VM's Base Image

Every VM is created from a base image, which is a template or ISO which has been created and stored in CloudStack. Both cloud administrators and end users can create and modify templates, ISOs, and VMs.

In CloudStack, you can change an existing VM's base image from one template to another, or from one ISO to another. (You can not change from an ISO to a template, or from a template to an ISO).

For example, suppose there is a template based on a particular operating system, and the OS vendor releases a software patch. The administrator or user naturally wants to apply the patch and then make sure existing VMs start using it. Whether a software update is involved or not, it's also possible to simply switch a VM from its current template to any other desired template.

To change a VM's base image, call the restoreVirtualMachine API command and pass in the virtual machine ID and a new template ID. The template ID parameter may refer to either a template or an ISO, depending on which type of base image the VM was already using (it must match the previous type of image). When this call occurs, the VM's root disk is first destroyed, then a new root disk is created from the source designated in the template ID parameter. The new root disk is attached to the VM, and now the VM is based on the new template.

You can also omit the template ID parameter from the restoreVirtualMachine call. In this case, the VM's root disk is destroyed and recreated, but from the same template or ISO that was already in use by the VM.

Using SSH Keys for Authentication

In addition to the username and password authentication, CloudStack supports using SSH keys to log in to the cloud infrastructure for additional security. You can use the createSSHKeyPair API to generate the SSH keys.

Because each cloud user has their own SSH key, one cloud user cannot log in to another cloud user's instances unless they share their SSH key files. Using a single SSH key pair, you can manage multiple instances.

Creating an Instance Template that Supports SSH Keys

Create an instance template that supports SSH Keys.

1. Create a new instance by using the template provided by cloudstack.

   For more information on creating a new instance, see

2. Download the cloudstack script from The SSH Key Gen Script to the instance you have created.

```
wget http://downloads.sourceforge.net/project/cloudstack/SSH%20Key%20Gen%20Script/
↪cloud-set-guest-sshkey.in?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fcloudstack
↪%2Ffiles%2FSSH%2520Key%2520Gen%2520Script%2F&ts=1331225219&use_mirror=iweb
```

3. Copy the file to /etc/init.d.

```
cp cloud-set-guest-sshkey.in /etc/init.d/
```

4. Give the necessary permissions on the script:

```
chmod +x /etc/init.d/cloud-set-guest-sshkey.in
```

5. Run the script while starting up the operating system:

```
chkconfig --add cloud-set-guest-sshkey.in
```

6. Stop the instance.

Creating the SSH Keypair

You must make a call to the createSSHKeyPair api method. You can either use the CloudStack Python API library or the curl commands to make the call to the cloudstack api.

For example, make a call from the cloudstack server to create a SSH keypair called "keypair-doc" for the admin account in the root domain:

Note: Ensure that you adjust these values to meet your needs. If you are making the API call from a different server, your URL/PORT will be different, and you will need to use the API keys.

1. Run the following curl command:

```
curl --globoff "http://localhost:8096/?command=createSSHKeyPair&name=keypair-doc&
↪account=admin&domainid=5163440e-c44b-42b5-9109-ad75cae8e8a2"
```

The output is something similar to what is given below:

```
<?xml version="1.0" encoding="ISO-8859-1"?><createsshkeypairresponse cloud-stack-
↪version="3.0.0.20120228045507"><keypair><name>keypair-doc</name><fingerprint>
↪f6:77:39:d5:5e:77:02:22:6a:d8:7f:ce:ab:cd:b3:56</fingerprint><privatekey>-----
↪BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCSydmnQ67jP6lNoXdX3noZjQdrMAWNQZ7y5SrEu4wDxplvhYci
dXYBeZVwakDVsU2MLGl/K+wefwefwefwefwefJyKJaogMKn7BperPD6n1wIDAQAB
AoGAdXaJ7uyZKeRDoy6wA0UmF0kSPbMZCR+UTIHNkS/E0/4U+6lhMokmFSHtu
mfDZ1kGGDYhMsdytjDBztljawfawfeawefawfawfawQQDCjEsoRdgkduTy
QpbSGDIa11Jsc+XNDx2fgRinDsxXI/zJYXTKRhSl/LIPHBw/brW8vzxhOlSOrwm7
VvemkkgpAkEAwSeEw394LYZiEVv395ar9MLRVTVLwpo54jC4tsOxQCBlloocK
lYaocpk0yBqqOUSBawfIiDCuLXSdvBo1Xz5ICTM19vgvEp/+kMuECQBzm
nVo8b2Gvyagqt/KEQo8wzH2THghZ1qQ1QRhIeJG2aissEacF6bGB2oZ7Igim5L14
4KR7OeEToyCLC2k+02UCQQCrniSnWKtDVoVqeK/zbB32JhW3Wullv5p5zUEcd
KfEEuzcCUIxtJYTahJ1pvlFkQ8anpuxjSEDp8x/18bq3
-----END RSA PRIVATE KEY-----
</privatekey></keypair></createsshkeypairresponse>
```

2. Copy the key data into a file. The file looks like this:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCSydmnQ67jP6lNoXdX3noZjQdrMAWNQZ7y5SrEu4wDxplvhYci
dXYBeZVwakDVsU2MLGl/K+wefwefwefwefwefJyKJaogMKn7BperPD6n1wIDAQAB
AoGAdXaJ7uyZKeRDoy6wA0UmF0kSPbMZCR+UTIHNkS/E0/4U+6lhMokmFSHtu
mfDZ1kGGDYhMsdytjDBztljawfawfeawefawfawfawQQDCjEsoRdgkduTy
```

(continues on next page)

(continued from previous page)

```
QpbSGDIa11Jsc+XNDx2fgRinDsxXI/zJYXTKRhSl/LIPHBw/brW8vzxhOlSOrwm7
VvemkkgpAkEAwSeEw394LYZiEVv395ar9MLRVTVLwpo54jC4tsOxQCBlloocK
lYaocpk0yBqqOUSBawfIiDCuLXSdvBo1Xz5ICTM19vgvEp/+kMuECQBzm
nVo8b2Gvyagqt/KEQo8wzH2THghZ1qQ1QRhIeJG2aissEacF6bGB2oZ7Igim5L14
4KR7OeEToyCLC2k+02UCQQCrniSnWKtDVoVqeK/zbB32JhW3Wullv5p5zUEcd
KfEEuzcCUIxtJYTahJ1pvlFkQ8anpuxjSEDp8x/18bq3
-----END RSA PRIVATE KEY-----
```

3. Save the file.

Creating an Instance

After you save the SSH keypair file, you must create an instance by using the template that you created at *Section 5.2.1, " Creating an Instance Template that Supports SSH Keys"*. Ensure that you use the same SSH key name that you created at *Section 5.2.2, "Creating the SSH Keypair"*.

Note: You cannot create the instance by using the GUI at this time and associate the instance with the newly created SSH keypair.

A sample curl command to create a new instance is:

```
curl --globoff http://localhost:<port number>/?command=deployVirtualMachine&
↪zoneId=1&serviceOfferingId=18727021-7556-4110-9322-d625b52e0813&
↪templateId=e899c18a-ce13-4bbf-98a9-625c5026e0b5&securitygroupids=ff03f02f-
↪9e3b-48f8-834d-91b822da40c5&account=admin&domainid=1&keypair=keypair-doc
```

Substitute the template, service offering and security group IDs (if you are using the security group feature) that are in your cloud environment.

Logging In Using the SSH Keypair

To test your SSH key generation is successful, check whether you can log in to the cloud setup.

For example, from a Linux OS, run:

```
ssh -i ~/.ssh/keypair-doc <ip address>
```

The -i parameter tells the ssh client to use a ssh key found at ~/.ssh/keypair-doc.

Resetting SSH Keys

With the API command resetSSHKeyForVirtualMachine, a user can set or reset the SSH keypair assigned to a virtual machine. A lost or compromised SSH keypair can be changed, and the user can access the VM by using the new keypair. Just create or register a new keypair, then call resetSSHKeyForVirtualMachine.

User-Data and Meta-Data

CloudStack provides API access to attach up to 2KB of data after base64 encoding to a deployed VM. Using HTTP POST(via POST body), you can send up to 32K of data after base64 encoding. Deployed VMs also have access to instance metadata via the virtual router.

Create virtual machine thru the API: deployVirtualMachine using the parameter `userdata=` to include user-data formated in base64.

Accessed user-data from VM. Once the IP address of the virtual router is known, use the following steps to retrieve user-data:

1. Run the following command to find the virtual router.

```
# cat /var/lib/dhclient/dhclient-eth0.leases | grep dhcp-server-identifier | tail␣
↪-1
```

2. Access user-data by running the following command using the result of the above command

```
# curl http://10.1.1.1/latest/user-data
```

Meta Data can be accessed similarly, using a URL of the form `http://10.1.1.1/latest/meta-data/{metadata type}`. (For backwards compatibility, the previous URL `http://10.1.1.1/latest/{metadata type}` is also supported.) For metadata type, use one of the following:

- `service-offering`. A description of the VMs service offering
- `availability-zone`. The Zone name
- `local-ipv4`. The guest IP of the VM
- `local-hostname`. The hostname of the VM
- `public-ipv4`. The first public IP for the router. (E.g. the first IP of eth2)
- `public-hostname`. This is the same as public-ipv4
- `instance-id`. The instance name of the VM

Using Cloud-Init

Cloud-Init can be use to access an interpret user-data from virtual machines. Cloud-Init be installed into templates and also require CloudStack password and sshkey scripts (adding-password-management-to-templates and using ssh keys). User password management and `resetSSHKeyForVirtualMachine` API are not yet supported by cloud-init.

1. Install cloud-init package into a template:

```
# yum install cloud-init
  or
$ sudo apt-get install cloud-init
```

2. Create datasource configuration file: `/etc/cloud/cloud.cfg.d/99_cloudstack.cfg`

```
datasource:
  CloudStack: {}
  None: {}
datasource_list:
  - CloudStack
```

user-data example

This example use cloud-init to Upgrade Operating-System of the newly created VM:

```
#cloud-config

# Upgrade the instance on first boot
# (ie run apt-get upgrade)
#
# Default: false
# Aliases: apt_upgrade
package_upgrade: true
```

base64 formated:


```
I2Nsb3VkLWNvbmZpZw0KDQojIFVwZ3JhZGUgdGhlIGluc3RhbmNlIG9uIGZpcnN0IGJvb3QNCiMgKGllIHJ1biBhcHQtZ2V0IHVw
```


Refer to Cloud-Init CloudStack datasource documentation for latest capabilities. Cloud-Init and Cloud-Init CloudStack datasource are not supported by Apache CloudStack community.

Assigning GPU/vGPU to Guest VMs

CloudStack can deploy guest VMs with Graphics Processing Unit (GPU) or Virtual Graphics Processing Unit (vGPU) capabilities on XenServer hosts. At the time of VM deployment or at a later stage, you can assign a physical GPU (known as GPU-passthrough) or a portion of a physical GPU card (vGPU) to a guest VM by changing the Service Offering. With this capability, the VMs running on CloudStack meet the intensive graphical processing requirement by means of the high computation power of GPU/vGPU, and CloudStack users can run multimedia rich applications, such as Auto-CAD, that they otherwise enjoy at their desk on a virtualized environment. CloudStack leverages the XenServer support for NVIDIA GRID Kepler 1 and 2 series to run GPU/vGPU enabled VMs. NVIDIA GRID cards allows sharing a single GPU cards among multiple VMs by creating vGPUs for each VM. With vGPU technology, the graphics commands from each VM are passed directly to the underlying dedicated GPU, without the intervention of the hypervisor. This allows the GPU hardware to be time-sliced and shared across multiple VMs. XenServer hosts use the GPU cards in following ways:

GPU passthrough: GPU passthrough represents a physical GPU which can be directly assigned to a VM. GPU passthrough can be used on a hypervisor alongside GRID vGPU, with some restrictions: A GRID physical GPU can either host GRID vGPUs or be used as passthrough, but not both at the same time.

GRID vGPU: GRID vGPU enables multiple VMs to share a single physical GPU. The VMs run an NVIDIA driver stack and get direct access to the GPU. GRID physical GPUs are capable of supporting multiple virtual GPU devices (vGPUs) that can be assigned directly to guest VMs. Guest VMs use GRID virtual GPUs in the same manner as a physical GPU that has been passed through by the hypervisor: an NVIDIA driver loaded in the guest VM provides direct access to the GPU for performance-critical fast paths, and a paravirtualized interface to the GRID Virtual GPU Manager, which is used for nonperformant management operations. NVIDIA GRID Virtual GPU Manager for XenServer runs in dom0. CloudStack provides you with the following capabilities:

- Adding XenServer hosts with GPU/vGPU capability provisioned by the administrator.
- Creating a Compute Offering with GPU/vGPU capability.
- Deploying a VM with GPU/vGPU capability.
- Destroying a VM with GPU/vGPU capability.
- Allowing an user to add GPU/vGPU support to a VM without GPU/vGPU support by changing the Service Offering and vice-versa.
- Migrating VMs (cold migration) with GPU/vGPU capability.
- Managing GPU cards capacity.
- Querying hosts to obtain information about the GPU cards, supported vGPU types in case of GRID cards, and capacity of the cards.

Prerequisites and System Requirements

Before proceeding, ensure that you have these prerequisites:

- The vGPU-enabled XenServer 6.2 and later versions. For more information, see Citrix 3D Graphics Pack.
- GPU/vPGU functionality is supported for following HVM guest operating systems: For more information, see Citrix 3D Graphics Pack.
- Windows 7 (x86 and x64)
- Windows Server 2008 R2
- Windows Server 2012
- Windows 8 (x86 and x64)
- Windows 8.1 ("Blue") (x86 and x64)
- Windows Server 2012 R2 (server equivalent of "Blue")
- CloudStack does not restrict the deployment of GPU-enabled VMs with guest OS types that are not supported by XenServer for GPU/vGPU functionality. The deployment would be successful and a GPU/vGPU will also get allocated for VMs; however, due to missing guest OS drivers, VM would not be able to leverage GPU resources. Therefore, it is recommended to use GPU-enabled service offering only with supported guest OS.
- NVIDIA GRID K1 (16 GiB video RAM) AND K2 (8 GiB of video RAM) cards supports homogeneous virtual GPUs, implies that at any given time, the vGPUs resident on a single physical GPU must be all of the same type. However, this restriction doesn't extend across physical GPUs on the same card. Each physical GPU on a K1 or K2 may host different types of virtual GPU at the same time. For example, a GRID K2 card has two physical GPUs, and supports four types of virtual GPU; GRID K200, GRID K220Q, GRID K240Q, AND GRID K260Q.
- NVIDIA driver must be installed to enable vGPU operation as for a physical NVIDIA GPU.
- XenServer tools are installed in the VM to get maximum performance on XenServer, regardless of type of vGPU you are using. Without the optimized networking and storage drivers that the XenServer tools provide, remote graphics applications running on GRID vGPU will not deliver maximum performance.
- To deliver high frames from multiple heads on vGPU, install XenDesktop with HDX 3D Pro remote graphics.

Before continuing with configuration, consider the following:

- Deploying VMs GPU/vGPU capability is not supported if hosts are not available with enough GPU capacity.
- A Service Offering cannot be created with the GPU values that are not supported by CloudStack UI. However, you can make an API call to achieve this.
- Dynamic scaling is not supported. However, you can choose to deploy a VM without GPU support, and at a later point, you can change the system offering to upgrade to the one with vGPU. You can achieve this by offline upgrade: stop the VM, upgrade the Service Offering to the one with vGPU, then start the VM.
- Live migration of GPU/vGPU enabled VM is not supported.
- Limiting GPU resources per Account/Domain is not supported.
- Disabling GPU at Cluster level is not supported.
- Notification thresholds for GPU resource is not supported.

Supported GPU Devices

Device	Type
GPU	• Group of NVIDIA Corporation GK107GL [GRID K1] GPUs • Group of NVIDIA Corporation GK104GL [GRID K2] GPUs • Any other GPU Group
vGPU	• GRID K100 • GRID K120Q • GRID K140Q • GRID K200 • GRID K220Q • GRID K240Q • GRID K260Q

GPU/vGPU Assignment Workflow

CloudStack follows the below sequence of operations to provide GPU/vGPU support for VMs:

1. Ensure that XenServer host is ready with GPU installed and configured. For more information, see Citrix 3D Graphics Pack.
2. Add the host to CloudStack. CloudStack checks if the host is GPU-enabled or not. CloudStack queries the host and detect if it's GPU enabled.
3. Create a compute offering with GPU/vGPU support: For more information, see *Creating a New Compute Offering*..
4. Continue with any of the following operations:
   - Deploy a VM.

     Deploy a VM with GPU/vGPU support by selecting appropriate Service Offering. CloudStack decide which host to choose for VM deployment based on following criteria:

     - Host has GPU cards in it. In case of vGPU, CloudStack checks if cards have the required vGPU type support and enough capacity available. Having no appropriate hosts results in an InsufficientServerCapacity exception.
     - Alternately, you can choose to deploy a VM without GPU support, and at a later point, you can change the system offering. You can achieve this by offline upgrade: stop the VM, upgrade the Service Offering to the one with vGPU, then start the VM. In this case, CloudStack gets a list of hosts which have enough capacity to host the VM. If there is a GPU-enabled host, CloudStack reorders this host list and place the GPU-enabled hosts at the bottom of the list.
   - Migrate a VM.

     CloudStack searches for hosts available for VM migration, which satisfies GPU requirement. If the host is available, stop the VM in the current host and perform the VM migration task. If the VM migration is successful, the remaining GPU capacity is updated for both the hosts accordingly.
   - Destroy a VM.

GPU resources are released automatically when you stop a VM. Once the destroy VM is successful, CloudStack will make a resource call to the host to get the remaining GPU capacity in the card and update the database accordingly.

5.7 Working with Templates

A template is a reusable configuration for virtual machines. When users launch VMs, they can choose from a list of templates in CloudStack.

Specifically, a template is a virtual disk image that includes one of a variety of operating systems, optional additional software such as office applications, and settings such as access control to determine who can use the template. Each template is associated with a particular type of hypervisor, which is specified when the template is added to CloudStack.

CloudStack ships with a default template. In order to present more choices to users, CloudStack administrators and users can create templates and add them to CloudStack.

5.7.1 Creating Templates: Overview

CloudStack ships with a default template for the CentOS operating system. There are a variety of ways to add more templates. Administrators and end users can add templates. The typical sequence of events is:

1. Launch a VM instance that has the operating system you want. Make any other desired configuration changes to the VM.
2. Stop the VM.
3. Convert the volume into a template.

There are other ways to add templates to CloudStack. For example, you can take a snapshot of the VM's volume and create a template from the snapshot, or import a VHD from another system into CloudStack.

The various techniques for creating templates are described in the next few sections.

5.7.2 Requirements for Templates

- For XenServer, install PV drivers / Xen tools on each template that you create. This will enable live migration and clean guest shutdown.
- For vSphere, install VMware Tools on each template that you create. This will enable console view to work properly.

5.7.3 Best Practices for Templates

If you plan to use large templates (100 GB or larger), be sure you have a 10-gigabit network to support the large templates. A slower network can lead to timeouts and other errors when large templates are used.

5.7.4 The Default Template

CloudStack includes a CentOS template. This template is downloaded by the Secondary Storage VM after the primary and secondary storage are configured. You can use this template in your production deployment or you can delete it and use custom templates.

The root password for the default template is "password".

A default template is provided for each of XenServer, KVM, and vSphere. The templates that are downloaded depend on the hypervisor type that is available in your cloud. Each template is approximately 2.5 GB physical size.

The default template includes the standard iptables rules, which will block most access to the template excluding ssh.

```
# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere        anywhere       icmp any
ACCEPT     esp  --  anywhere        anywhere
ACCEPT     ah   --  anywhere        anywhere
ACCEPT     udp  --  anywhere        224.0.0.251    udp dpt:mdns
ACCEPT     udp  --  anywhere        anywhere       udp dpt:ipp
ACCEPT     tcp  --  anywhere        anywhere       tcp dpt:ipp
ACCEPT     all  --  anywhere        anywhere       state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere        anywhere       state NEW tcp dpt:ssh
REJECT     all  --  anywhere        anywhere       reject-with icmp-host-
```

5.7.5 Private and Public Templates

When a user creates a template, it can be designated private or public.

Private templates are only available to the user who created them. By default, an uploaded template is private.

When a user marks a template as "public," the template becomes available to all users in all accounts in the user's domain, as well as users in any other domains that have access to the Zone where the template is stored. This depends on whether the Zone, in turn, was defined as private or public. A private Zone is assigned to a single domain, and a public Zone is accessible to any domain. If a public template is created in a private Zone, it is available only to users in the domain assigned to that Zone. If a public template is created in a public Zone, it is available to all users in all domains.

5.7.6 Creating a Template from an Existing Virtual Machine

Once you have at least one VM set up in the way you want, you can use it as the prototype for other VMs.

1. Create and start a virtual machine using any of the techniques given in "Creating VMs".
2. Make any desired configuration changes on the running VM, then click Stop.
3. Wait for the VM to stop. When the status shows Stopped, go to the next step.
4. Go into "View Volumes" and select the Volume having the type "ROOT".
5. Click Create Template and provide the following:

- **Name and Display Text**. These will be shown in the UI, so choose something descriptive.
- **OS Type**. This helps CloudStack and the hypervisor perform certain operations and make assumptions that improve the performance of the guest. Select one of the following.
  - If the operating system of the stopped VM is listed, choose it.
  - If the OS type of the stopped VM is not listed, choose Other.
  - If you want to boot from this template in PV mode, choose Other PV (32-bit) or Other PV (64-bit). This choice is available only for XenServere:

    Note: Generally you should not choose an older version of the OS than the version in the image. For example, choosing CentOS 5.4 to support a CentOS 6.2 image will in general not work. In those cases you should choose Other.

- **Public**. Choose Yes to make this template accessible to all users of this CloudStack installation. The template will appear in the Community Templates list. See *"Private and Public Templates"*.
- **Password Enabled**. Choose Yes if your template has the CloudStack password change script installed. See adding-password-management-to-templates.

6. Click Add.

The new template will be visible in the Templates section when the template creation process has been completed. The template is then available when creating a new VM.

5.7.7 Creating a Template from a Snapshot

If you do not want to stop the VM in order to use the Create Template menu item (as described in *"Creating a Template from an Existing Virtual Machine"*), you can create a template directly from any snapshot through the CloudStack UI.

5.7.8 Uploading Templates

5.7.9 vSphere Templates and ISOs

If you are uploading a template that was created using vSphere Client, be sure the OVA file does not contain an ISO. If it does, the deployment of VMs from the template will fail.

Templates are uploaded based on a URL. HTTP is the supported access protocol. Templates are frequently large files. You can optionally gzip them to decrease upload times.

To upload a template:

1. In the left navigation bar, click Templates.
2. Click Register Template.
3. Provide the following:
   - **Name and Description**. These will be shown in the UI, so choose something descriptive.
   - **URL**. The Management Server will download the file from the specified URL, such as `http://my.web.server/filename.vhd.gz`.
   - **Zone**. Choose the zone where you want the template to be available, or All Zones to make it available throughout CloudStack.

- **OS Type**: This helps CloudStack and the hypervisor perform certain operations and make assumptions that improve the performance of the guest. Select one of the following:
  - If the operating system of the stopped VM is listed, choose it.
  - If the OS type of the stopped VM is not listed, choose Other.

  Note: You should not choose an older version of the OS than the version in the image. For example, choosing CentOS 5.4 to support a CentOS 6.2 image will in general not work. In those cases you should choose Other.

- **Hypervisor**: The supported hypervisors are listed. Select the desired one.
- **Format**. The format of the template upload file, such as VHD or OVA.
- **Password Enabled**. Choose Yes if your template has the CloudStack password change script installed. See adding-password-management-to-templates.
- **Extractable**. Choose Yes if the template is available for extraction. If this option is selected, end users can download a full image of a template.
- **Public**. Choose Yes to make this template accessible to all users of this CloudStack installation. The template will appear in the Community Templates list. See *"Private and Public Templates"*.
- **Featured**. Choose Yes if you would like this template to be more prominent for users to select. The template will appear in the Featured Templates list. Only an administrator can make a template Featured.

5.7.10 Exporting Templates

End users and Administrators may export templates from the CloudStack. Navigate to the template in the UI and choose the Download function from the Actions menu.

5.7.11 Creating a Linux Template

Linux templates should be prepared using this documentation in order to prepare your linux VMs for template deployment. For ease of documentation, the VM which you are configuring the template on will be referred to as "Template Master". This guide currently covers legacy setups which do not take advantage of UserData and cloud-init and assumes openssh-server is installed during installation.

An overview of the procedure is as follow:

1. Upload your Linux ISO.

   For more information, see "Adding an ISO".

2. Create a VM Instance with this ISO.

   For more information, see "Creating VMs".

3. Prepare the Linux VM
4. Create a template from the VM.

   For more information, see *"Creating a Template from an Existing Virtual Machine"*.

System preparation for Linux

The following steps will prepare a basic Linux installation for templating.

1. **Installation**

   It is good practice to name your VM something generic during installation, this will ensure components such as LVM do not appear unique to a machine. It is recommended that the name of "localhost" is used for installation.

   > **Warning:** For CentOS, it is necessary to take unique identification out of the interface configuration file, for this edit /etc/sysconfig/network-scripts/ifcfg-eth0 and change the content to the following.

   ```
   DEVICE=eth0
   TYPE=Ethernet
   BOOTPROTO=dhcp
   ONBOOT=yes
   ```

   The next steps updates the packages on the Template Master.

   - Ubuntu

     ```
     sudo -i
     apt-get update
     apt-get upgrade -y
     apt-get install -y acpid ntp
     reboot
     ```

   - CentOS

     ```
     ifup eth0
     yum update -y
     reboot
     ```

2. **Password management**

   > **Note:** If preferred, custom users (such as ones created during the Ubuntu installation) should be removed. First ensure the root user account is enabled by giving it a password and then login as root to continue.

   ```
   sudo passwd root
   logout
   ```

   As root, remove any custom user accounts created during the installation process.

   ```
   deluser myuser --remove-home
   ```

   See adding-password-management-to-templates for instructions to setup the password management script, this will allow CloudStack to change your root password from the web interface.

3. **Hostname Management**

   CentOS configures the hostname by default on boot. Unfortunately Ubuntu does not have this functionality, for Ubuntu installations use the following steps.

   - Ubuntu

The hostname of a Templated VM is set by a custom script in */etc/dhcp/dhclient-exit-hooks.d*, this script first checks if the current hostname is localhost, if true, it will get the host-name, domain-name and fixed-ip from the DHCP lease file and use those values to set the hostname and append the */etc/hosts* file for local hostname resolution. Once this script, or a user has changed the hostname from localhost, it will no longer adjust system files regardless of its new hostname. The script also recreates openssh-server keys, which should have been deleted before templating (shown below). Save the following script to */etc/dhcp/dhclient-exit-hooks.d/sethostname*, and adjust the permissions.

```
#!/bin/sh
# dhclient change hostname script for Ubuntu
oldhostname=$(hostname -s)
if [ $oldhostname = 'localhost' ]
then
    sleep 10 # Wait for configuration to be written to disk
    hostname=$(cat /var/lib/dhcp/dhclient.eth0.leases  |  awk ' /host-name/ {␣
↪host = $3 }  END { printf host } ' | sed     's/[";]//g' )
    fqdn="$hostname.$(cat /var/lib/dhcp/dhclient.eth0.leases  |  awk ' /
↪domain-name/ { domain = $3 }  END { printf     domain } ' | sed 's/[";]//g')
↪"
    ip=$(cat /var/lib/dhcp/dhclient.eth0.leases  |  awk ' /fixed-address/ {␣
↪lease = $2 }  END { printf lease } ' | sed     's/[";]//g')
    echo "cloudstack-hostname: Hostname _localhost_ detected. Changing␣
↪hostname and adding hosts."
    printf " Hostname: $hostname\n FQDN: $fqdn\n IP: $ip"
    # Update /etc/hosts
    awk -v i="$ip" -v f="$fqdn" -v h="$hostname" "/^127/{x=1} !/^127/ && x {␣
↪x=0; print i,f,h; } { print $0; }" /etc/hosts > /etc/hosts.dhcp.tmp
    mv /etc/hosts /etc/hosts.dhcp.bak
    mv /etc/hosts.dhcp.tmp /etc/hosts
    # Rename Host
    echo $hostname > /etc/hostname
    hostname -b -F /etc/hostname
    echo $hostname > /proc/sys/kernel/hostname
    # Recreate SSH2
    export DEBIAN_FRONTEND=noninteractive
    dpkg-reconfigure openssh-server
fi
### End of Script ###

chmod 774  /etc/dhcp/dhclient-exit-hooks.d/sethostname
```

Warning: The following steps should be run when you are ready to template your Template Master. If the Template Master is rebooted during these steps you will have to run all the steps again. At the end of this process the Template Master should be shutdown and the template created in order to create and deploy the final template.

4. **Remove the udev persistent device rules**

   This step removes information unique to your Template Master such as network MAC addresses, lease files and CD block devices, the files are automatically generated on next boot.

   - Ubuntu

```
rm -f /etc/udev/rules.d/70*
rm -f /var/lib/dhcp/dhclient.*
```

- CentOS

```
rm -f /etc/udev/rules.d/70*
rm -f /var/lib/dhclient/*
```

5. **Remove SSH Keys**

   This step is to ensure all your Templated VMs do not have the same SSH keys, which would decrease the security of the machines dramatically.

```
rm -f /etc/ssh/*key*
```

6. **Cleaning log files**

   It is good practice to remove old logs from the Template Master.

```
cat /dev/null > /var/log/audit/audit.log 2>/dev/null
cat /dev/null > /var/log/wtmp 2>/dev/null
logrotate -f /etc/logrotate.conf 2>/dev/null
rm -f /var/log/*-* /var/log/*.gz 2>/dev/null
```

7. **Setting hostname**

   In order for the Ubuntu DHCP script to function and the CentOS dhclient to set the VM hostname they both require the Template Master's hostname to be "localhost", run the following commands to change the hostname.

```
hostname localhost
echo "localhost" > /etc/hostname
```

8. **Set user password to expire**

   This step forces the user to change the password of the VM after the template has been deployed.

```
passwd --expire root
```

9. **Clearing User History**

   The next step clears the bash commands you have just run.

```
history -c
unset HISTFILE
```

10. **Shutdown the VM**

    Your now ready to shutdown your Template Master and create a template!

```
halt -p
```

11. **Create the template!**

    You are now ready to create the template, for more information see *"Creating a Template from an Existing Virtual Machine"*.

Note: Templated VMs for both Ubuntu and CentOS may require a reboot after provisioning in order to pickup the hostname.

5.7.12 Creating a Windows Template

Windows templates must be prepared with Sysprep before they can be provisioned on multiple machines. Sysprep allows you to create a generic Windows template and avoid any possible SID conflicts.

Note: (XenServer) Windows VMs running on XenServer require PV drivers, which may be provided in the template or added after the VM is created. The PV drivers are necessary for essential management functions such as mounting additional volumes and ISO images, live migration, and graceful shutdown.

An overview of the procedure is as follows:

1. Upload your Windows ISO.

   For more information, see "Adding an ISO".

2. Create a VM Instance with this ISO.

   For more information, see "Creating VMs".

3. Follow the steps in Sysprep for Windows Server 2008 R2 (below) or Sysprep for Windows Server 2003 R2, depending on your version of Windows Server

4. The preparation steps are complete. Now you can actually create the template as described in Creating the Windows Template.

System Preparation for Windows Server 2008 R2

For Windows 2008 R2, you run Windows System Image Manager to create a custom sysprep response XML file. Windows System Image Manager is installed as part of the Windows Automated Installation Kit (AIK). Windows AIK can be downloaded from Microsoft Download Center.

Use the following steps to run sysprep for Windows 2008 R2:

Note: The steps outlined here are derived from the excellent guide by Charity Shelbourne, originally published at Windows Server 2008 Sysprep Mini-Setup.

1. Download and install the Windows AIK

   Note: Windows AIK should not be installed on the Windows 2008 R2 VM you just created. Windows AIK should not be part of the template you create. It is only used to create the sysprep answer file.

2. Copy the install.wim file in the \sources directory of the Windows 2008 R2 installation DVD to the hard disk. This is a very large file and may take a long time to copy. Windows AIK requires the WIM file to be writable.

3. Start the Windows System Image Manager, which is part of the Windows AIK.

4. In the Windows Image pane, right click the Select a Windows image or catalog file option to load the install.wim file you just copied.

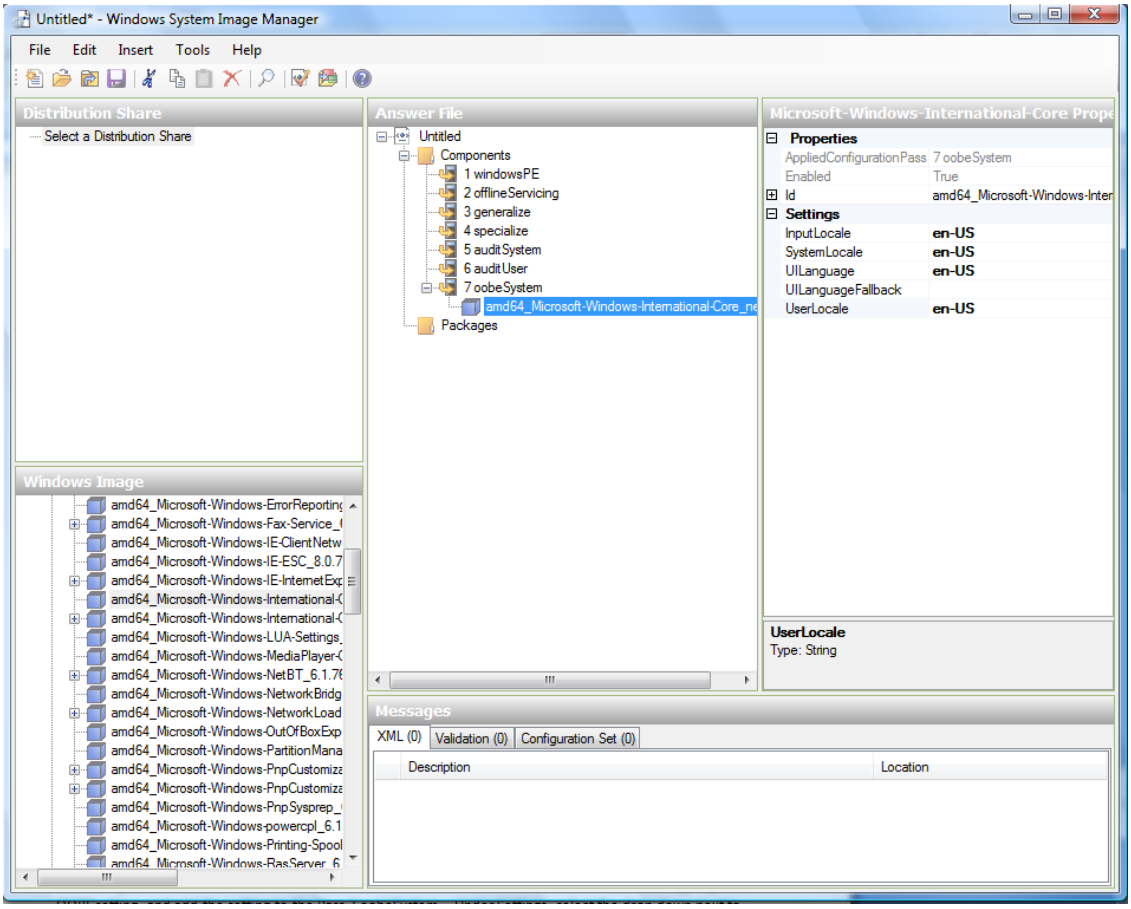
5. Select the Windows 2008 R2 Edition.

   You may be prompted with a warning that the catalog file cannot be opened. Click Yes to create a new catalog file.

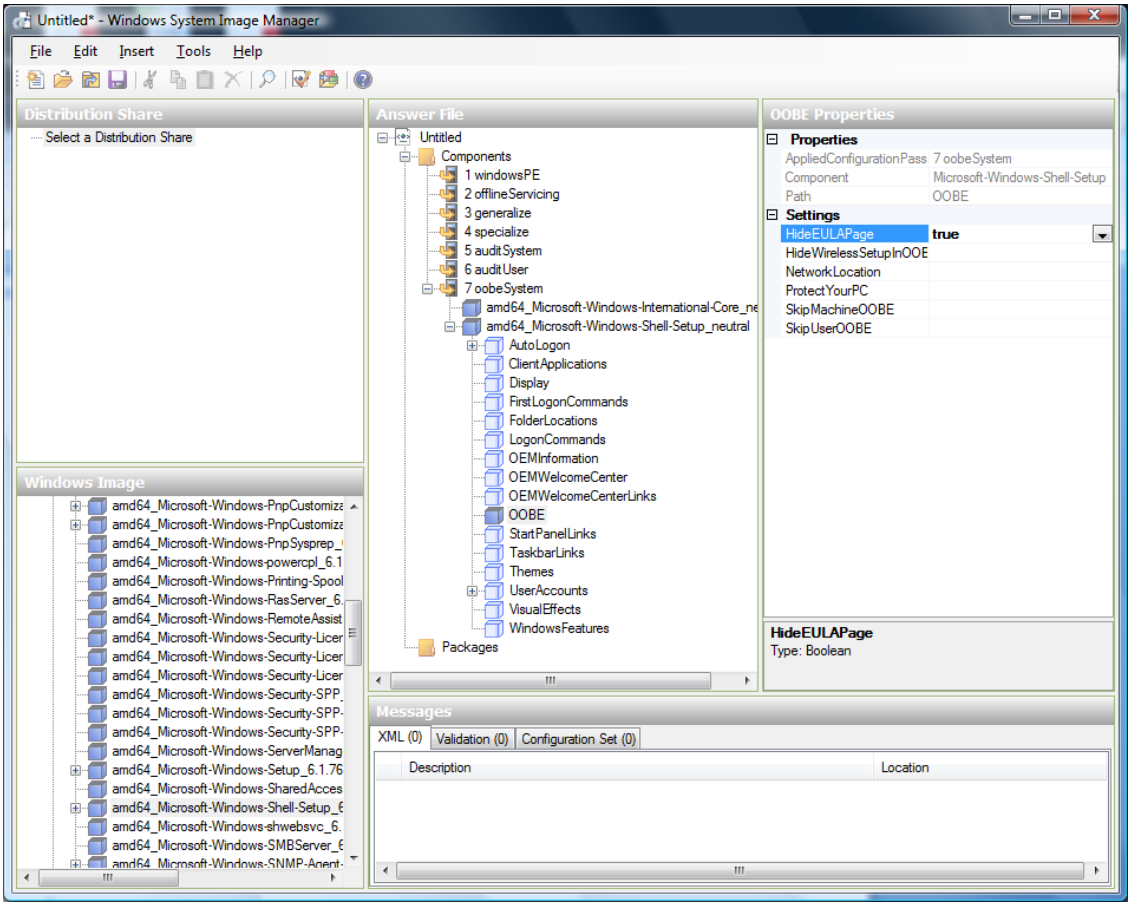
6. In the Answer File pane, right click to create a new answer file.

7. Generate the answer file from the Windows System Image Manager using the following steps:

(a) The first page you need to automate is the Language and Country or Region Selection page. To automate this, expand Components in your Windows Image pane, right-click and add the Microsoft-Windows-International-Core setting to Pass 7 oobeSystem. In your Answer File pane, configure the InputLocale, SystemLocale, UILanguage, and UserLocale with the appropriate settings for your language and country or region. Should you have a question about any of these settings, you can right-click on the specific setting and select Help. This will open the appropriate CHM help file with more information, including examples on the setting you are attempting to configure.

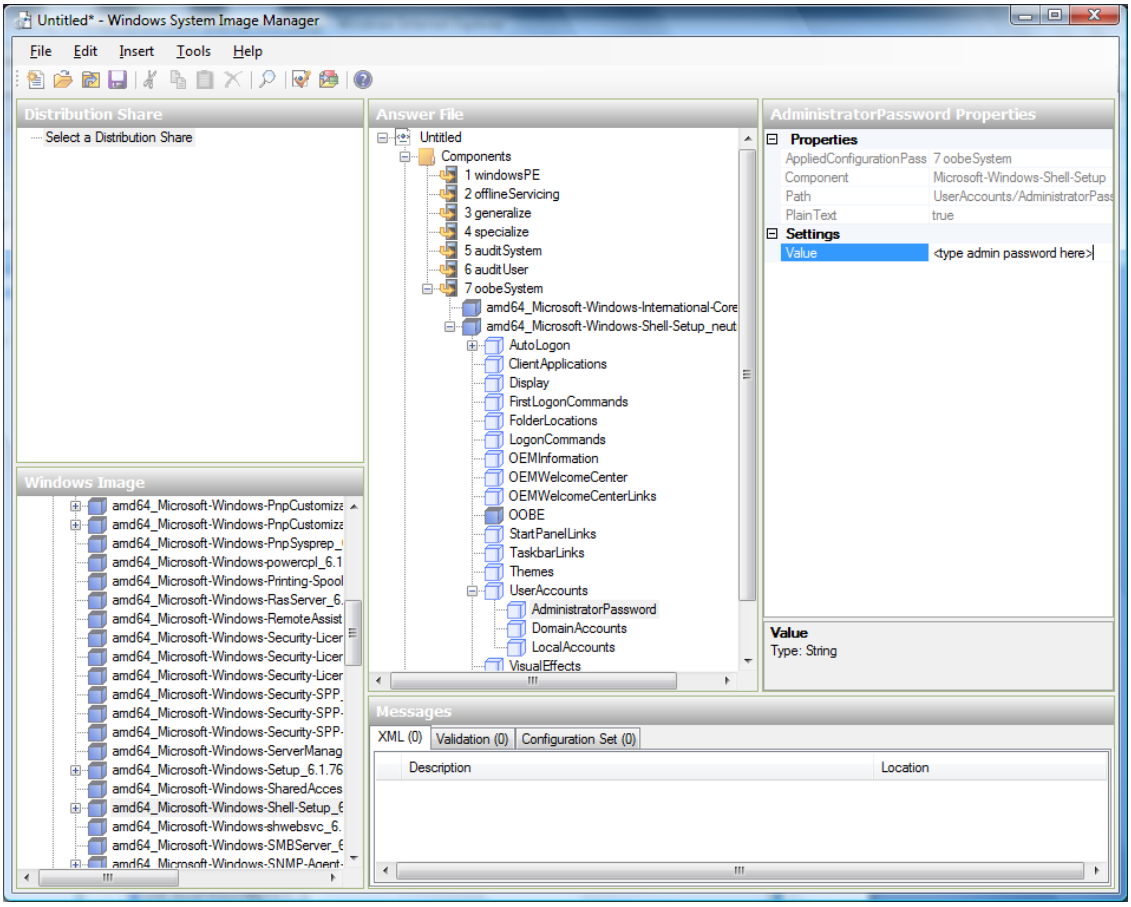


(b) You need to automate the Software License Terms Selection page, otherwise known as the End-User License Agreement (EULA). To do this, expand the Microsoft-Windows-Shell-Setup component. Highlight the OOBE setting, and add the setting to the Pass 7 oobeSystem. In Settings, set HideEULAPage true.



(c) Make sure the license key is properly set. If you use MAK key, you can just enter the MAK key on the Windows 2008 R2 VM. You need not input the MAK into the Windows System Image Manager. If you use KMS host for activation you need not enter the Product Key. Details of Windows Volume Activation can be found at http://technet.microsoft.com/en-us/library/bb892849.aspx

(d) You need to automate is the Change Administrator Password page. Expand the Microsoft-Windows-Shell-Setup component (if it is not still expanded), expand UserAccounts, right-click on AdministratorPassword, and add the setting to the Pass 7 oobeSystem configuration pass of your answer file. Under Settings, specify a password next to Value.



You may read the AIK documentation and set many more options that suit your deployment. The steps above are the minimum needed to make Windows unattended setup work.

8. Save the answer file as unattend.xml. You can ignore the warning messages that appear in the validation window.
9. Copy the unattend.xml file into the c:\windows\system32\sysprep directory of the Windows 2008 R2 Virtual Machine
10. Once you place the unattend.xml file in c:\windows\system32\sysprep directory, you run the sysprep tool as follows:

```
cd c:\Windows\System32\sysprep
sysprep.exe /oobe /generalize /shutdown
```

The Windows 2008 R2 VM will automatically shut down after sysprep is complete.

System Preparation for Windows Server 2003 R2

Earlier versions of Windows have a different sysprep tool. Follow these steps for Windows Server 2003 R2.

1. Extract the content of \support\tools\deploy.cab on the Windows installation CD into a directory called c:\sysprep on the Windows 2003 R2 VM.
2. Run c:\sysprep\setupmgr.exe to create the sysprep.inf file.
   (a) Select Create New to create a new Answer File.
   (b) Enter "Sysprep setup" for the Type of Setup.

(c) Select the appropriate OS version and edition.

(d) On the License Agreement screen, select "Yes fully automate the installation".

(e) Provide your name and organization.

(f) Leave display settings at default.

(g) Set the appropriate time zone.

(h) Provide your product key.

(i) Select an appropriate license mode for your deployment

(j) Select "Automatically generate computer name".

(k) Type a default administrator password. If you enable the password reset feature, the users will not actually use this password. This password will be reset by the instance manager after the guest boots up.

(l) Leave Network Components at "Typical Settings".

(m) Select the "WORKGROUP" option.

(n) Leave Telephony options at default.

(o) Select appropriate Regional Settings.

(p) Select appropriate language settings.

(q) Do not install printers.

(r) Do not specify "Run Once commands".

(s) You need not specify an identification string.

(t) Save the Answer File as c:\sysprep\sysprep.inf.

3. Run the following command to sysprep the image:

```
c:\sysprep\sysprep.exe -reseal -mini -activated
```

After this step the machine will automatically shut down

5.7.13 Importing Amazon Machine Images

The following procedures describe how to import an Amazon Machine Image (AMI) into CloudStack when using the XenServer hypervisor.

Assume you have an AMI file and this file is called CentOS_6.2_x64. Assume further that you are working on a CentOS host. If the AMI is a Fedora image, you need to be working on a Fedora host initially.

You need to have a XenServer host with a file-based storage repository (either a local ext3 SR or an NFS SR) to convert to a VHD once the image file has been customized on the Centos/Fedora host.

Note: When copying and pasting a command, be sure the command has pasted as a single line before executing. Some document viewers may introduce unwanted line breaks in copied text.

To import an AMI:

1. Set up loopback on image file:

```
# mkdir -p /mnt/loop/centos62
# mount -o loop  CentOS_6.2_x64 /mnt/loop/centos54
```

2. Install the kernel-xen package into the image. This downloads the PV kernel and ramdisk to the image.

```
# yum -c /mnt/loop/centos54/etc/yum.conf --installroot=/mnt/loop/centos62/ -y␣
↪install kernel-xen
```

3. Create a grub entry in /boot/grub/grub.conf.

```
# mkdir -p /mnt/loop/centos62/boot/grub
# touch /mnt/loop/centos62/boot/grub/grub.conf
# echo "" > /mnt/loop/centos62/boot/grub/grub.conf
```

4. Determine the name of the PV kernel that has been installed into the image.

```
# cd /mnt/loop/centos62
# ls lib/modules/
2.6.16.33-xenU  2.6.16-xenU  2.6.18-164.15.1.el5xen  2.6.18-164.6.1.el5.centos.
↪plus  2.6.18-xenU-ec2-v1.0  2.6.21.7-2.fc8xen  2.6.31-302-ec2
# ls boot/initrd*
boot/initrd-2.6.18-164.6.1.el5.centos.plus.img boot/initrd-2.6.18-164.15.1.el5xen.
↪img
# ls boot/vmlinuz*
boot/vmlinuz-2.6.18-164.15.1.el5xen  boot/vmlinuz-2.6.18-164.6.1.el5.centos.plus ␣
↪boot/vmlinuz-2.6.18-xenU-ec2-v1.0  boot/vmlinuz-2.6.21-2952.fc8xen
```

Xen kernels/ramdisk always end with "xen". For the kernel version you choose, there has to be an entry for that version under lib/modules, there has to be an initrd and vmlinuz corresponding to that. Above, the only kernel that satisfies this condition is 2.6.18-164.15.1.el5xen.

5. Based on your findings, create an entry in the grub.conf file. Below is an example entry.

```
default=0
timeout=5
hiddenmenu
title CentOS (2.6.18-164.15.1.el5xen)
   root (hd0,0)
   kernel /boot/vmlinuz-2.6.18-164.15.1.el5xen ro root=/dev/xvda
   initrd /boot/initrd-2.6.18-164.15.1.el5xen.img
```

6. Edit etc/fstab, changing "sda1" to "xvda" and changing "sdb" to "xvdb".

```
# cat etc/fstab
/dev/xvda  /         ext3    defaults        1 1
/dev/xvdb  /mnt      ext3    defaults        0 0
none       /dev/pts  devpts  gid=5,mode=620  0 0
none       /proc     proc    defaults        0 0
none       /sys      sysfs   defaults        0 0
```

7. Enable login via the console. The default console device in a XenServer system is xvc0. Ensure that etc/inittab and etc/securetty have the following lines respectively:

```
# grep xvc0 etc/inittab
co:2345:respawn:/sbin/agetty xvc0 9600 vt100-nav
# grep xvc0 etc/securetty
xvc0
```

8. Ensure the ramdisk supports PV disk and PV network. Customize this for the kernel version you have determined above.

```
# chroot /mnt/loop/centos54
# cd /boot/
# mv initrd-2.6.18-164.15.1.el5xen.img initrd-2.6.18-164.15.1.el5xen.img.bak
# mkinitrd -f /boot/initrd-2.6.18-164.15.1.el5xen.img --with=xennet --
↪preload=xenblk --omit-scsi-modules 2.6.18-164.15.1.el5xen
```

9. Change the password.

```
# passwd
Changing password for user root.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

10. Exit out of chroot.

```
# exit
```

11. Check *etc/ssh/sshd_config* for lines allowing ssh login using a password.

```
# egrep "PermitRootLogin|PasswordAuthentication" /mnt/loop/centos54/etc/ssh/sshd_
↪config
PermitRootLogin yes
PasswordAuthentication yes
```

12. If you need the template to be enabled to reset passwords from the CloudStack UI or API, install the password change script into the image at this point. See adding-password-management-to-templates.

13. Unmount and delete loopback mount.

```
# umount /mnt/loop/centos54
# losetup -d /dev/loop0
```

14. Copy the image file to your XenServer host's file-based storage repository. In the example below, the Xenserver is "xenhost". This XenServer has an NFS repository whose uuid is a9c5b8c8-536b-a193-a6dc-51af3e5ff799.

```
# scp CentOS_6.2_x64 xenhost:/var/run/sr-mount/a9c5b8c8-536b-a193-a6dc-
↪51af3e5ff799/
```

15. Log in to the Xenserver and create a VDI the same size as the image.

```
[root@xenhost ~]# cd /var/run/sr-mount/a9c5b8c8-536b-a193-a6dc-51af3e5ff799
[root@xenhost a9c5b8c8-536b-a193-a6dc-51af3e5ff799]#  ls -lh CentOS_6.2_x64
-rw-r--r-- 1 root root 10G Mar 16 16:49 CentOS_6.2_x64
[root@xenhost a9c5b8c8-536b-a193-a6dc-51af3e5ff799]# xe vdi-create virtual-
↪size=10GiB sr-uuid=a9c5b8c8-536b-a193-a6dc-51af3e5ff799 type=user name-label=
↪"Centos 6.2 x86_64"
cad7317c-258b-4ef7-b207-cdf0283a7923
```

16. Import the image file into the VDI. This may take 10–20 minutes.

```
[root@xenhost a9c5b8c8-536b-a193-a6dc-51af3e5ff799]# xe vdi-import␣
↪filename=CentOS_6.2_x64 uuid=cad7317c-258b-4ef7-b207-cdf0283a7923
```

17. Locate a the VHD file. This is the file with the VDI's UUID as its name. Compress it and upload it to your web server.

```
[root@xenhost a9c5b8c8-536b-a193-a6dc-51af3e5ff799]# bzip2 -c cad7317c-258b-4ef7-
↪b207-cdf0283a7923.vhd > CentOS_6.2_x64.vhd.bz2
[root@xenhost a9c5b8c8-536b-a193-a6dc-51af3e5ff799]# scp CentOS_6.2_x64.vhd.bz2␣
↪webserver:/var/www/html/templates/
```

5.7.14 Converting a Hyper-V VM to a Template

To convert a Hyper-V VM to a XenServer-compatible CloudStack template, you will need a standalone XenServer host with an attached NFS VHD SR. Use whatever XenServer version you are using with CloudStack, but use XenCenter 5.6 FP1 or SP2 (it is backwards compatible to 5.6). Additionally, it may help to have an attached NFS ISO SR.

For Linux VMs, you may need to do some preparation in Hyper-V before trying to get the VM to work in XenServer. Clone the VM and work on the clone if you still want to use the VM in Hyper-V. Uninstall Hyper-V Integration Components and check for any references to device names in /etc/fstab:

1. From the linux_ic/drivers/dist directory, run make uninstall (where "linux_ic" is the path to the copied Hyper-V Integration Components files).
2. Restore the original initrd from backup in /boot/ (the backup is named *.backup0).
3. Remove the "hdX=noprobe" entries from /boot/grub/menu.lst.
4. Check /etc/fstab for any partitions mounted by device name. Change those entries (if any) to mount by LABEL or UUID. You can get that information with the blkid command.

The next step is make sure the VM is not running in Hyper-V, then get the VHD into XenServer. There are two options for doing this.

Option one:

1. Import the VHD using XenCenter. In XenCenter, go to Tools>Virtual Appliance Tools>Disk Image Import.
2. Choose the VHD, then click Next.
3. Name the VM, choose the NFS VHD SR under Storage, enable "Run Operating System Fixups" and choose the NFS ISO SR.
4. Click Next, then Finish. A VM should be created.

Option two:

1. Run XenConvert, under From choose VHD, under To choose XenServer. Click Next.
2. Choose the VHD, then click Next.
3. Input the XenServer host info, then click Next.
4. Name the VM, then click Next, then Convert. A VM should be created.

Once you have a VM created from the Hyper-V VHD, prepare it using the following steps:

1. Boot the VM, uninstall Hyper-V Integration Services, and reboot.
2. Install XenServer Tools, then reboot.
3. Prepare the VM as desired. For example, run sysprep on Windows VMs. See *"Creating a Windows Template"*.

Either option above will create a VM in HVM mode. This is fine for Windows VMs, but Linux VMs may not perform optimally. Converting a Linux VM to PV mode will require additional steps and will vary by distribution.

1. Shut down the VM and copy the VHD from the NFS storage to a web server; for example, mount the NFS share on the web server and copy it, or from the XenServer host use sftp or scp to upload it to the web server.
2. In CloudStack, create a new template using the following values:

- URL. Give the URL for the VHD
- OS Type. Use the appropriate OS. For PV mode on CentOS, choose Other PV (32-bit) or Other PV (64-bit). This choice is available only for XenServer.
- Hypervisor. XenServer
- Format. VHD

The template will be created, and you can create instances from it.

5.7.15 Adding Password Management to Your Templates

CloudStack provides an optional password reset feature that allows users to set a temporary admin or root password as well as reset the existing admin or root password from the CloudStack UI.

To enable the Reset Password feature, you will need to download an additional script to patch your template. When you later upload the template into CloudStack, you can specify whether reset admin/root password feature should be enabled for this template.

The password management feature works always resets the account password on instance boot. The script does an HTTP call to the virtual router to retrieve the account password that should be set. As long as the virtual router is accessible the guest will have access to the account password that should be used. When the user requests a password reset the management server generates and sends a new password to the virtual router for the account. Thus an instance reboot is necessary to effect any password changes.

If the script is unable to contact the virtual router during instance boot it will not set the password but boot will continue normally.

Linux OS Installation

Use the following steps to begin the Linux OS installation:

1. Download the script file cloud-set-guest-password:
   - http://download.cloud.com/templates/4.2/bindir/cloud-set-guest-password.in
2. Rename the file:

   ```
   mv cloud-set-guest-password.in cloud-set-guest-password
   ```

3. Copy this file to /etc/init.d.

   On some Linux distributions, copy the file to `/etc/rc.d/init.d`.

4. Run the following command to make the script executable:

   ```
   chmod +x /etc/init.d/cloud-set-guest-password
   ```

5. Depending on the Linux distribution, continue with the appropriate step.

   On Fedora, CentOS/RHEL, and Debian, run:

   ```
   chkconfig --add cloud-set-guest-password
   ```

Windows OS Installation

Download the installer, CloudInstanceManager.msi, from the Download page and run the installer in the newly created Windows VM.

5.7.16 Deleting Templates

Templates may be deleted. In general, when a template spans multiple Zones, only the copy that is selected for deletion will be deleted; the same template in other Zones will not be deleted. The provided CentOS template is an exception to this. If the provided CentOS template is deleted, it will be deleted from all Zones.

When templates are deleted, the VMs instantiated from them will continue to run. However, new VMs cannot be created based on the deleted template.

5.8 Working with Hosts

5.8.1 Adding Hosts

Additional hosts can be added at any time to provide more capacity for guest VMs. For requirements and instructions, see *Adding a Host*.

5.8.2 Scheduled Maintenance and Maintenance Mode for Hosts

You can place a host into maintenance mode. When maintenance mode is activated, the host becomes unavailable to receive new guest VMs, and the guest VMs already running on the host are seamlessly migrated to another host not in maintenance mode. This migration uses live migration technology and does not interrupt the execution of the guest.

vCenter and Maintenance Mode

To enter maintenance mode on a vCenter host, both vCenter and CloudStack must be used in concert. CloudStack and vCenter have separate maintenance modes that work closely together.

1. Place the host into CloudStack's "scheduled maintenance" mode. This does not invoke the vCenter maintenance mode, but only causes VMs to be migrated off the host

   When the CloudStack maintenance mode is requested, the host first moves into the Prepare for Maintenance state. In this state it cannot be the target of new guest VM starts. Then all VMs will be migrated off the server. Live migration will be used to move VMs off the host. This allows the guests to be migrated to other hosts with no disruption to the guests. After this migration is completed, the host will enter the Ready for Maintenance mode.

2. Wait for the "Ready for Maintenance" indicator to appear in the UI.

3. Now use vCenter to perform whatever actions are necessary to maintain the host. During this time, the host cannot be the target of new VM allocations.

4. When the maintenance tasks are complete, take the host out of maintenance mode as follows:

   (a) First use vCenter to exit the vCenter maintenance mode.

   This makes the host ready for CloudStack to reactivate it.

   (b) Then use CloudStack's administrator UI to cancel the CloudStack maintenance mode

   When the host comes back online, the VMs that were migrated off of it may be migrated back to it manually and new VMs can be added.

XenServer and Maintenance Mode

For XenServer, you can take a server offline temporarily by using the Maintenance Mode feature in XenCenter. When you place a server into Maintenance Mode, all running VMs are automatically migrated from it to another host in the same pool. If the server is the pool master, a new master will also be selected for the pool. While a server is Maintenance Mode, you cannot create or start any VMs on it.

To place a server in Maintenance Mode:

1. In the Resources pane, select the server, then do one of the following:
   - Right-click, then click Enter Maintenance Mode on the shortcut menu.
   - On the Server menu, click Enter Maintenance Mode.
2. Click Enter Maintenance Mode.

The server's status in the Resources pane shows when all running VMs have been successfully migrated off the server.

To take a server out of Maintenance Mode:

1. In the Resources pane, select the server, then do one of the following:
   - Right-click, then click Exit Maintenance Mode on the shortcut menu.
   - On the Server menu, click Exit Maintenance Mode.
2. Click Exit Maintenance Mode.

5.8.3 Disabling and Enabling Zones, Pods, and Clusters

You can enable or disable a zone, pod, or cluster without permanently removing it from the cloud. This is useful for maintenance or when there are problems that make a portion of the cloud infrastructure unreliable. No new allocations will be made to a disabled zone, pod, or cluster until its state is returned to Enabled. When a zone, pod, or cluster is first added to the cloud, it is Disabled by default.

To disable and enable a zone, pod, or cluster:

1. Log in to the CloudStack UI as administrator
2. In the left navigation bar, click Infrastructure.
3. In Zones, click View More.
4. If you are disabling or enabling a zone, find the name of the zone in the list, and click the Enable/Disable button.
5. If you are disabling or enabling a pod or cluster, click the name of the zone that contains the pod or cluster.
6. Click the Compute tab.
7. In the Pods or Clusters node of the diagram, click View All.
8. Click the pod or cluster name in the list.
9. Click the Enable/Disable button.

5.8.4 Removing Hosts

Hosts can be removed from the cloud as needed. The procedure to remove a host depends on the hypervisor type.

Removing XenServer and KVM Hosts

A node cannot be removed from a cluster until it has been placed in maintenance mode. This will ensure that all of the VMs on it have been migrated to other Hosts. To remove a Host from the cloud:

1. Place the node in maintenance mode.

   See *"Scheduled Maintenance and Maintenance Mode for Hosts"*.

2. For KVM, stop the cloud-agent service.
3. Use the UI option to remove the node.

   Then you may power down the Host, re-use its IP address, re-install it, etc

Removing vSphere Hosts

To remove this type of host, first place it in maintenance mode, as described in *"Scheduled Maintenance and Maintenance Mode for Hosts"*. Then use CloudStack to remove the host. CloudStack will not direct commands to a host that has been removed using CloudStack. However, the host may still exist in the vCenter cluster.

5.8.5 Re-Installing Hosts

You can re-install a host after placing it in maintenance mode and then removing it. If a host is down and cannot be placed in maintenance mode, it should still be removed before the re-install.

5.8.6 Maintaining Hypervisors on Hosts

When running hypervisor software on hosts, be sure all the hotfixes provided by the hypervisor vendor are applied. Track the release of hypervisor patches through your hypervisor vendor's support channel, and apply patches as soon as possible after they are released. CloudStack will not track or notify you of required hypervisor patches. It is essential that your hosts are completely up to date with the provided hypervisor patches. The hypervisor vendor is likely to refuse to support any system that is not up to date with patches.

Note: The lack of up-do-date hotfixes can lead to data corruption and lost VMs.

(XenServer) For more information, see Highly Recommended Hotfixes for XenServer in the CloudStack Knowledge Base.

5.8.7 Changing Host Password

The password for a XenServer Node, KVM Node, or vSphere Node may be changed in the database. Note that all Nodes in a Cluster must have the same password.

To change a Node's password:

1. Identify all hosts in the cluster.
2. Change the password on all hosts in the cluster. Now the password for the host and the password known to CloudStack will not match. Operations on the cluster will fail until the two passwords match.
3. if the password in the database is encrypted, it is (likely) necessary to encrypt the new password using the database key before adding it to the database.

```
java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.0.jar \
org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI \
encrypt.sh input="newrootpassword" \
password="databasekey" \
verbose=false
```

4. Get the list of host IDs for the host in the cluster where you are changing the password. You will need to access the database to determine these host IDs. For each hostname "h" (or vSphere cluster) that you are changing the password for, execute:

```
mysql> SELECT id FROM cloud.host WHERE name like '%h%';
```

5. This should return a single ID. Record the set of such IDs for these hosts. Now retrieve the host_details row id for the host

```
mysql> SELECT * FROM cloud.host_details WHERE name='password' AND host_id=
↪{previous step ID};
```

6. Update the passwords for the host in the database. In this example, we change the passwords for hosts with host IDs 5 and 12 and host_details IDs 8 and 22 to "password".

```
mysql> UPDATE cloud.host_details SET value='password' WHERE id=8 OR id=22;
```

5.8.8 Over-Provisioning and Service Offering Limits

(Supported for XenServer, KVM, and VMware)

CPU and memory (RAM) over-provisioning factors can be set for each cluster to change the number of VMs that can run on each host in the cluster. This helps optimize the use of resources. By increasing the over-provisioning ratio, more resource capacity will be used. If the ratio is set to 1, no over-provisioning is done.

The administrator can also set global default over-provisioning ratios in the cpu.overprovisioning.factor and mem.overprovisioning.factor global configuration variables. The default value of these variables is 1: over-provisioning is turned off by default.

Over-provisioning ratios are dynamically substituted in CloudStack's capacity calculations. For example:

Capacity = 2 GB Over-provisioning factor = 2 Capacity after over-provisioning = 4 GB

With this configuration, suppose you deploy 3 VMs of 1 GB each:

Used = 3 GB Free = 1 GB

The administrator can specify a memory over-provisioning ratio, and can specify both CPU and memory over-provisioning ratios on a per-cluster basis.

In any given cloud, the optimum number of VMs for each host is affected by such things as the hypervisor, storage, and hardware configuration. These may be different for each cluster in the same cloud. A single global over-provisioning setting can not provide the best utilization for all the different clusters in the cloud. It has to be set for the lowest common denominator. The per-cluster setting provides a finer granularity for better utilization of resources, no matter where the CloudStack placement algorithm decides to place a VM.

The overprovisioning settings can be used along with dedicated resources (assigning a specific cluster to an account) to effectively offer different levels of service to different accounts. For example, an account paying for a more expensive level of service could be assigned to a dedicated cluster with an over-provisioning ratio of 1, and a lower-paying account to a cluster with a ratio of 2.

When a new host is added to a cluster, CloudStack will assume the host has the capability to perform the CPU and RAM over-provisioning which is configured for that cluster. It is up to the administrator to be sure the host is actually suitable for the level of over-provisioning which has been set.

Limitations on Over-Provisioning in XenServer and KVM

- In XenServer, due to a constraint of this hypervisor, you can not use an over-provisioning factor greater than 4.
- The KVM hypervisor can not manage memory allocation to VMs dynamically. CloudStack sets the minimum and maximum amount of memory that a VM can use. The hypervisor adjusts the memory within the set limits based on the memory contention.

Requirements for Over-Provisioning

Several prerequisites are required in order for over-provisioning to function properly. The feature is dependent on the OS type, hypervisor capabilities, and certain scripts. It is the administrator's responsibility to ensure that these requirements are met.

Balloon Driver

All VMs should have a balloon driver installed in them. The hypervisor communicates with the balloon driver to free up and make the memory available to a VM.

XenServer

The balloon driver can be found as a part of xen pv or PVHVM drivers. The xen pvhvm drivers are included in upstream linux kernels 2.6.36+.

VMware

The balloon driver can be found as a part of the VMware tools. All the VMs that are deployed in a over-provisioned cluster should have the VMware tools installed.

KVM

All VMs are required to support the virtio drivers. These drivers are installed in all Linux kernel versions 2.6.25 and greater. The administrator must set CONFIG_VIRTIO_BALLOON=y in the virtio configuration.

Hypervisor capabilities

The hypervisor must be capable of using the memory ballooning.

XenServer

The DMC (Dynamic Memory Control) capability of the hypervisor should be enabled. Only XenServer Advanced and above versions have this feature.

VMware, KVM

Memory ballooning is supported by default.

Setting Over-Provisioning Ratios

There are two ways the root admin can set CPU and RAM over-provisioning ratios. First, the global configuration settings cpu.overprovisioning.factor and mem.overprovisioning.factor will be applied when a new cluster is created. Later, the ratios can be modified for an existing cluster.

Only VMs deployed after the change are affected by the new setting. If you want VMs deployed before the change to adopt the new over-provisioning ratio, you must stop and restart the VMs. When this is done, CloudStack recalculates or scales the used and reserved capacities based on the new over-provisioning ratios, to ensure that CloudStack is correctly tracking the amount of free capacity.

Note: It is safer not to deploy additional new VMs while the capacity recalculation is underway, in case the new values for available capacity are not high enough to accommodate the new VMs. Just wait for the new used/available values to become available, to be sure there is room for all the new VMs you want.

To change the over-provisioning ratios for an existing cluster:

1. Log in as administrator to the CloudStack UI.
2. In the left navigation bar, click Infrastructure.
3. Under Clusters, click View All.
4. Select the cluster you want to work with, and click the Edit button.
5. Fill in your desired over-provisioning multipliers in the fields CPU overcommit ratio and RAM overcommit ratio. The value which is intially shown in these fields is the default value inherited from the global configuration settings.

   Note: In XenServer, due to a constraint of this hypervisor, you can not use an over-provisioning factor greater than 4.

Service Offering Limits and Over-Provisioning

Service offering limits (e.g. 1 GHz, 1 core) are strictly enforced for core count. For example, a guest with a service offering of one core will have only one core available to it regardless of other activity on the Host.

Service offering limits for gigahertz are enforced only in the presence of contention for CPU resources. For example, suppose that a guest was created with a service offering of 1 GHz on a Host that has 2 GHz cores, and that guest is the only guest running on the Host. The guest will have the full 2 GHz available to it. When multiple guests are attempting to use the CPU a weighting factor is used to schedule CPU resources. The weight is based on the clock speed in the service offering. Guests receive a CPU allocation that is proportionate to the GHz in the service offering. For example, a guest created from a 2 GHz service offering will receive twice the CPU allocation as a guest created from a 1 GHz service offering. CloudStack does not perform memory over-provisioning.

5.8.9 VLAN Provisioning

CloudStack automatically creates and destroys interfaces bridged to VLANs on the hosts. In general the administrator does not need to manage this process.

CloudStack manages VLANs differently based on hypervisor type. For XenServer or KVM, the VLANs are created on only the hosts where they will be used and then they are destroyed when all guests that require them have been terminated or moved to another host.

For vSphere the VLANs are provisioned on all hosts in the cluster even if there is no guest running on a particular Host that requires the VLAN. This allows the administrator to perform live migration and other functions in vCenter without having to create the VLAN on the destination Host. Additionally, the VLANs are not removed from the Hosts when they are no longer needed.

You can use the same VLANs on different physical networks provided that each physical network has its own underlying layer-2 infrastructure, such as switches. For example, you can specify VLAN range 500 to 1000 while deploying physical networks A and B in an Advanced zone setup. This capability allows you to set up an additional layer-2 physical infrastructure on a different physical NIC and use the same set of VLANs if you run out of VLANs. Another advantage is that you can use the same set of IPs for different customers, each one with their own routers and the guest networks on different physical NICs.

VLAN Allocation Example

VLANs are required for public and guest traffic. The following is an example of a VLAN allocation scheme:

VLAN IDs	Traffic type	Scope
less than 500	Management traffic.	Reserved for administrative purposes. CloudStack software can access this, hypervisors, system VMs.
500-599	VLAN carrying public traffic.	CloudStack accounts.
600-799	VLANs carrying guest traffic.	CloudStack accounts. Account-specific VLAN is chosen from this pool.
800-899	VLANs carrying guest traffic.	CloudStack accounts. Account-specific VLAN chosen by CloudStack admin to assign to that account.
900-999	VLAN carrying guest traffic	CloudStack accounts. Can be scoped by project, domain, or all accounts.
greater than 1000	Reserved for future use	

Adding Non Contiguous VLAN Ranges

CloudStack provides you with the flexibility to add non contiguous VLAN ranges to your network. The administrator can either update an existing VLAN range or add multiple non contiguous VLAN ranges while creating a zone. You can also use the UpdatephysicalNetwork API to extend the VLAN range.

1. Log in to the CloudStack UI as an administrator or end user.
2. Ensure that the VLAN range does not already exist.
3. In the left navigation, choose Infrastructure.
4. On Zones, click View More, then click the zone to which you want to work with.
5. Click Physical Network.
6. In the Guest node of the diagram, click Configure.

7. Click Edit .

   The VLAN Ranges field now is editable.

8. Specify the start and end of the VLAN range in comma-separated list.

   Specify all the VLANs you want to use, VLANs not specified will be removed if you are adding new ranges to the existing list.

9. Click Apply.

Assigning VLANs to Isolated Networks

CloudStack provides you the ability to control VLAN assignment to Isolated networks. As a Root admin, you can assign a VLAN ID when a network is created, just the way it's done for Shared networks.

The former behaviour also is supported — VLAN is randomly allocated to a network from the VNET range of the physical network when the network turns to Implemented state. The VLAN is released back to the VNET pool when the network shuts down as a part of the Network Garbage Collection. The VLAN can be re-used either by the same network when it is implemented again, or by any other network. On each subsequent implementation of a network, a new VLAN can be assigned.

Only the Root admin can assign VLANs because the regular users or domain admin are not aware of the physical network topology. They cannot even view what VLAN is assigned to a network.

To enable you to assign VLANs to Isolated networks,

1. Create a network offering by specifying the following:

   - **Guest Type**: Select Isolated.
   - **Specify VLAN**: Select the option.

   For more information, see the CloudStack Installation Guide.

2. Using this network offering, create a network.

   You can create a VPC tier or an Isolated network.

3. Specify the VLAN when you create the network.

   When VLAN is specified, a CIDR and gateway are assigned to this network and the state is changed to Setup. In this state, the network will not be garbage collected.

Note: You cannot change a VLAN once it's assigned to the network. The VLAN remains with the network for its entire life cycle.

5.8.10 Out-of-band Management

CloudStack provides Root admins the ability to configure and use supported out-of-band management interface (e.g. IPMI, iLO, DRAC, etc.) on a physical host to manage host power operations such as on, off, reset etc. By default, IPMI 2.0 baseboard controller are supported out of the box with `IPMITOOL` out-of-band management driver in CloudStack that uses `ipmitool` for performing IPMI 2.0 management operations.

Following are some global settings that control various aspects of this feature.

Global setting	Default values	Description
outofbandmanagement.action.timeout	60	The out of band management action timeout in seconds, configurable per cluster
outofbandmanagement.ipmitool.interface	lanplus	The out of band management IpmiTool driver interface to use. Valid values are: lan, lanplus etc
outofbandmanagement.ipmitool.path	/usr/bin/ipmitool	The out of band management ipmitool path used by the IpmiTool driver
outofbandmanagement.ipmitool.retries	1	The out of band management IpmiTool driver retries option -R
outofbandmanagement.sync.poolsize	50	The out of band management background sync thread pool size 50

A change in `outofbandmanagement.sync.poolsize` settings requires restarting of management server(s) as the thread pool and a background (power state) sync thread are configured during load time when CloudStack management server starts. Rest of the global settings can be changed without requiring restarting of management server(s).

The `outofbandmanagement.sync.poolsize` is the maximum number of ipmitool background power state scanners that can run at a time. Based on the maximum number of hosts you've, you can increase/decrease the value depending on how much stress your management server host can endure. It will take atmost number of total out-of-band-management enabled hosts in a round * `outofbandmanagement.action.timeout` / `outofbandmanagement.sync.poolsize` seconds to complete a background power-state sync scan in a single round.

In order to use this feature, the Root admin needs to first configure out-of-band management for a host using either the UI or the `configureOutOfBandManagement` API. Next, the Root admin needs to enable it. The feature can be enabled or disabled across a zone or a cluster or a host,

Once out-of-band management is configured and enabled for a host (and provided not disabled at zone or cluster level), Root admins would be able to issue power management actions such as on, off, reset, cycle, soft and status.

If a host is in maintenance mode, Root admins are still allowed to perform power management actions but in the UI a warning is displayed.

5.8.11 Security

Starting 4.11, CloudStack has an inbuilt certicate authority (CA) framework and a default 'root' CA provider which acts as a self-signed CA. The CA framework participates in certificate issuance, renewal, revocation, and propagation of certificates during setup of a host. This framework is primary used to secure communications between CloudStack management server(s), the KVM/LXC/SSVM/CPVM agent(s) and peer management server(s).

Following are some global settings that control various aspects of this feature.

Global setting	Description
ca.framework.provider.plugin	The configured CA provider plugin
ca.framework.cert.keysize	The key size used for certificate generation
ca.framework.cert.signature.algorithm	The certificate signature algorithm
ca.framework.cert.validity.period	Certificate validity in days
ca.framework.cert.automatic.renewal	Whether to auto-renew expiring certificate on hosts
ca.framework.background.task.delay	The delay between each CA background task round in seconds
ca.framework.cert.expiry.alert.period	The number of days to check and alert expiring certificates
ca.plugin.root.private.key	(hidden/encrypted in database) Auto-generated CA private key
ca.plugin.root.public.key	(hidden/encrypted in database) CA public key
ca.plugin.root.ca.certificate	(hidden/encrypted in database) CA certificate
ca.plugin.root.issuer.dn	The CA issue distinguished name used by the root CA provider
ca.plugin.root.auth.strictness	Setting to enforce two-way SSL authentication and trust validation
ca.plugin.root.allow.expired.cert	Setting to allow clients with expired certificates

A change in `ca.framework.background.task.delay` settings requires restarting of management server(s) as the thread pool and a background tasks are configured only when CloudStack management server(s) start.

After upgrade to CloudStack 4.11+, the CA framework will by default use the `root` CA provider. This CA provider will auto-generate its private/public keys and CA certificate on first boot post-upgrade. For freshly installed environments, the `ca.plugin.root.auth.strictness` setting will be `true` to enforce two-way SSL authentication and trust validation between client and server components, however, it will be `false` on upgraded environments to be backward compatible with legacy behaviour of trusting all clients and servers, and one-way SSL authentication. Upgraded/existing environments can use the `provisionCertificate` API to renew/setup certificates for already connected agents/hosts, and once all the agents/hosts are secured they may enforce authentication and validation strictness by setting `ca.plugin.root.auth.strictness` to `true` and restarting the management server(s).

5.8.12 Server Address Usage

Historically, when multiple management servers are used a tcp-LB is used on port 8250 (default) of the management servers and the VIP/LB-IP is used as the `host` setting to be used by various CloudStack agents such as the KVM, CPVM, SSVM agents, who connect to the `host` on port 8250. However, starting CloudStack 4.11+ the `host` setting can accept comma separated list of management server IPs to which new CloudStack hosts/agents will get a shuffled list of the same to which they can cycle reconnections in a round-robin way.

5.8.13 Securing Process

Agents while making connections/reconnections to management server will only validate server certificate and be able to present client certificate (issued to them) when `cloud.jks` is accessible to them. On older hosts that are setup prior to this feature the keystore won't be available, however, they can still connect to management server(s) if `ca.plugin.root.auth.strictness` is set to `false`. Management server(s) will check and setup their own `cloud.jks` keystore on startup, this keystore will be used for connecting to peer management server(s).

When a new host is being setup, such as adding a KVM host or starting a systemvm host, the CA framework kicks in and uses ssh to execute `keystore-setup` to generate a new keystore file `cloud.jks.new`, save a random passphrase of the keystore in the agent's properties file and a CSR `cloud.csr` file. The CSR is then used to issue certificate for that agent/host and ssh is used to execute `keystore-cert-import` to import the issued certificate along with the CA certificate(s), the keystore is that renamed as `cloud.jks` replacing an previous keystore in-use. During this process, keys and certificates files are also stored in `cloud.key`, `cloud.crt`, `cloud.ca.crt` in the agent's configuration directory.

When hosts are added out-of-band, for example a KVM host that is setup first outside of CloudStack and added to a cluster, the keystore file will not be available however the keystore and security could be setup by using keystore utility scripts manually. The `keystore-setup` can be ran first to generate a keystore and a CSR, then CloudStack CA can be used to issue certificate by providing the CSR to the `issueCertificate` API, and finally issued certificate and CA certificate(s) can be imported to the keystore using `keystore-cert-import` script.

Following lists the usage of these scripts, when using these script use full paths, use the final keystore filename as `cloud.jks`, and the certificate/key content need to be encoded and provided such that newlines are replace with ^ and space are replaced with ~:

```
keystore-setup <properties file> <keystore file> <passphrase> <validity> <csr file>

keystore-cert-import <properties file> <keystore file> <mode: ssh|agent> <cert file>
↪<cert content> <ca-cert file> <ca-cert content> <private-key file> <private key␣
↪content:optional>
```

Starting 4.11.1, a KVM host is considered secured when it has its keystore and certificates setup for both the agent and libvirtd process. A secured host will only allow and initiate TLS enabled live VM migration. This requires libvirtd to listen on default port 16514, and the port to be allowed in the firewall rules. Certificate renewal (using the `provisionCertificate` API) will restart both the libvirtd process and agent after deploying new certificates.

5.9 Working with Storage

5.9.1 Storage Overview

CloudStack defines two types of storage: primary and secondary. Primary storage can be accessed by either iSCSI or NFS. Additionally, direct attached storage may be used for primary storage. Secondary storage is always accessed using NFS.

There is no ephemeral storage in CloudStack. All volumes on all nodes are persistent.

5.9.2 Primary Storage

This section gives technical details about CloudStack primary storage. For more information about the concepts behind primary storage see *Primary Storage* . For information about how to install and configure primary storage through the CloudStack UI, see the in the Installation Guide.

Best Practices for Primary Storage

- The speed of primary storage will impact guest performance. If possible, choose smaller, higher RPM drives or SSDs for primary storage.
- There are two ways CloudStack can leverage primary storage:

  Static: This is CloudStack's traditional way of handling storage. In this model, a preallocated amount of storage (ex. a volume from a SAN) is given to CloudStack. CloudStack then permits many of its volumes to be created on this storage (can be root and/or data disks). If using this technique, ensure that nothing is stored on the storage. Adding the storage to CloudStack will destroy any existing data.

  Dynamic: This is a newer way for CloudStack to manage storage. In this model, a storage system (rather than a preallocated amount of storage) is given to CloudStack. CloudStack, working in concert with a storage plug-in, dynamically creates volumes on the storage system and each volume on the storage system maps to a single CloudStack volume. This is highly useful for features such as storage Quality of Service. Currently this feature is supported for data disks (Disk Offerings).

Runtime Behavior of Primary Storage

Root volumes are created automatically when a virtual machine is created. Root volumes are deleted when the VM is destroyed. Data volumes can be created and dynamically attached to VMs. Data volumes are not deleted when VMs are destroyed.

Administrators should monitor the capacity of primary storage devices and add additional primary storage as needed. See the Advanced Installation Guide.

Administrators add primary storage to the system by creating a CloudStack storage pool. Each storage pool is associated with a cluster or a zone.

With regards to data disks, when a user executes a Disk Offering to create a data disk, the information is initially written to the CloudStack database only. Upon the first request that the data disk be attached to a VM, CloudStack determines what storage to place the volume on and space is taken from that storage (either from preallocated storage or from a storage system (ex. a SAN), depending on how the primary storage was added to CloudStack).

Hypervisor Support for Primary Storage

The following table shows storage options and parameters for different hypervisors.

Storage media \ hypervisor	VMware vSphere	Citrix XenServer	KVM	Hyper-V
Format for Disks, Templates, and Snapshots	VMDK	VHD	QCOW2	VHD Snapshots are not supported.
iSCSI support	VMFS	Clustered LVM	Yes, via Shared Mountpoint	No
Fiber Channel support	VMFS	Yes, via Existing SR	Yes, via Shared Mountpoint	No
NFS support	Yes	Yes	Yes	No
Local storage support	Yes	Yes	Yes	Yes
Storage over-provisioning	NFS and iSCSI	NFS	NFS	No
SMB/CIFS	No	No	No	Yes
Ceph/RBD	No	No	Yes	No

XenServer uses a clustered LVM system to store VM images on iSCSI and Fiber Channel volumes and does not support over-provisioning in the hypervisor. The storage server itself, however, can support thin-provisioning. As a result the CloudStack can still support storage over-provisioning by running on thin-provisioned storage volumes.

KVM supports "Shared Mountpoint" storage. A shared mountpoint is a file system path local to each server in a given cluster. The path must be the same across all Hosts in the cluster, for example /mnt/primary1. This shared mountpoint is assumed to be a clustered filesystem such as OCFS2. In this case the CloudStack does not attempt to mount or unmount the storage as is done with NFS. The CloudStack requires that the administrator insure that the storage is available

With NFS storage, CloudStack manages the overprovisioning. In this case the global configuration parameter storage.overprovisioning.factor controls the degree of overprovisioning. This is independent of hypervisor type.

Local storage is an option for primary storage for vSphere, XenServer, and KVM. When the local disk option is enabled, a local disk storage pool is automatically created on each host. To use local storage for the System Virtual Machines (such as the Virtual Router), set system.vm.use.local.storage to true in global configuration.

CloudStack supports multiple primary storage pools in a Cluster. For example, you could provision 2 NFS servers in primary storage. Or you could provision 1 iSCSI LUN initially and then add a second iSCSI LUN when the first approaches capacity.

Storage Tags

Storage may be "tagged". A tag is a text string attribute associated with primary storage, a Disk Offering, or a Service Offering. Tags allow administrators to provide additional information about the storage. For example, that is a "SSD" or it is "slow". Tags are not interpreted by CloudStack. They are matched against tags placed on service and disk offerings. CloudStack requires all tags on service and disk offerings to exist on the primary storage before it allocates root or data disks on the primary storage. Service and disk offering tags are used to identify the requirements of the storage that those offerings have. For example, the high end service offering may require "fast" for its root disk volume.

The interaction between tags, allocation, and volume copying across clusters and pods can be complex. To simplify the situation, use the same set of tags on the primary storage for all clusters in a pod. Even if different devices are used to present those tags, the set of exposed tags can be the same.

Maintenance Mode for Primary Storage

Primary storage may be placed into maintenance mode. This is useful, for example, to replace faulty RAM in a storage device. Maintenance mode for a storage device will first stop any new guests from being provisioned on the storage device. Then it will stop all guests that have any volume on that storage device. When all such guests are stopped the storage device is in maintenance mode and may be shut down. When the storage device is online again you may cancel maintenance mode for the device. The CloudStack will bring the device back online and attempt to start all guests that were running at the time of the entry into maintenance mode.

5.9.3 Secondary Storage

This section gives concepts and technical details about CloudStack secondary storage. For information about how to install and configure secondary storage through the CloudStack UI, see the Advanced Installation Guide. about-secondary-storage>'_

5.9.4 Working With Volumes

A volume provides storage to a guest VM. The volume can provide for a root disk or an additional data disk. CloudStack supports additional volumes for guest VMs.

Volumes are created for a specific hypervisor type. A volume that has been attached to guest using one hypervisor type (e.g, XenServer) may not be attached to a guest that is using another hypervisor type, for example:vSphere, KVM. This is because the different hypervisors use different disk image formats.

CloudStack defines a volume as a unit of storage available to a guest VM. Volumes are either root disks or data disks. The root disk has "/" in the file system and is usually the boot device. Data disks provide for additional storage, for example: "/opt" or "D:". Every guest VM has a root disk, and VMs can also optionally have a data disk. End users can mount multiple data disks to guest VMs. Users choose data disks from the disk offerings created by administrators. The user can create a template from a volume as well; this is the standard procedure for private template creation. Volumes are hypervisor-specific: a volume from one hypervisor type may not be used on a guest of another hypervisor type.

Note: CloudStack supports attaching up to

- 13 data disks on XenServer hypervisor versions 6.0 and above, And all versions of VMware.
- 64 data disks on Hyper-V.
- 6 data disks on other hypervisor types.

Creating a New Volume

You can add more data disk volumes to a guest VM at any time, up to the limits of your storage capacity. Both CloudStack administrators and users can add volumes to VM instances. When you create a new volume, it is stored as an entity in CloudStack, but the actual storage resources are not allocated on the physical storage device until you attach the volume. This optimization allows the CloudStack to provision the volume nearest to the guest that will use it when the first attachment is made.

Using Local Storage for Data Volumes

You can create data volumes on local storage (supported with XenServer, KVM, and VMware). The data volume is placed on the same host as the VM instance that is attached to the data volume. These local data volumes can be attached to virtual machines, detached, re-attached, and deleted just as with the other types of data volume.

Local storage is ideal for scenarios where persistence of data volumes and HA is not required. Some of the benefits include reduced disk I/O latency and cost reduction from using inexpensive local disks.

In order for local volumes to be used, the feature must be enabled for the zone.

You can create a data disk offering for local storage. When a user creates a new VM, they can select this disk offering in order to cause the data disk volume to be placed in local storage.

You can not migrate a VM that has a volume in local storage to a different host, nor migrate the volume itself away to a different host. If you want to put a host into maintenance mode, you must first stop any VMs with local data volumes on that host.

To Create a New Volume

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation bar, click Storage.
3. In Select View, choose Volumes.
4. To create a new volume, click Add Volume, provide the following details, and click OK.
   - Name. Give the volume a unique name so you can find it later.
   - Availability Zone. Where do you want the storage to reside? This should be close to the VM that will use the volume.
   - Disk Offering. Choose the characteristics of the storage.

   The new volume appears in the list of volumes with the state "Allocated." The volume data is stored in CloudStack, but the volume is not yet ready for use
5. To start using the volume, continue to Attaching a Volume

Uploading an Existing Volume to a Virtual Machine

Existing data can be made accessible to a virtual machine. This is called uploading a volume to the VM. For example, this is useful to upload data from a local file system and attach it to a VM. Root administrators, domain administrators, and end users can all upload existing volumes to VMs.

The upload is performed using HTTP. The uploaded volume is placed in the zone's secondary storage

You cannot upload a volume if the preconfigured volume limit has already been reached. The default limit for the cloud is set in the global configuration parameter max.account.volumes, but administrators can also set per-domain limits that are different from the global default. See Setting Usage Limits

To upload a volume:

1. (Optional) Create an MD5 hash (checksum) of the disk image file that you are going to upload. After uploading the data disk, CloudStack will use this value to verify that no data corruption has occurred.
2. Log in to the CloudStack UI as an administrator or user
3. In the left navigation bar, click Storage.
4. Click Upload Volume.
5. Provide the following:
   - Name and Description. Any desired name and a brief description that can be shown in the UI.
   - Availability Zone. Choose the zone where you want to store the volume. VMs running on hosts in this zone can attach the volume.
   - Format. Choose one of the following to indicate the disk image format of the volume.

Hypervisor	Disk Image Format
XenServer	VHD
VMware	OVA
KVM	QCOW2

   - URL. The secure HTTP or HTTPS URL that CloudStack can use to access your disk. The type of file at the URL must match the value chosen in Format. For example, if Format is VHD, the URL might look like the following:

     `http://yourFileServerIP/userdata/myDataDisk.vhd`
   - MD5 checksum. (Optional) Use the hash that you created in step 1.
6. Wait until the status of the volume shows that the upload is complete. Click Instances - Volumes, find the name you specified in step 5, and make sure the status is Uploaded.

Attaching a Volume

You can attach a volume to a guest VM to provide extra disk storage. Attach a volume when you first create a new volume, when you are moving an existing volume from one VM to another, or after you have migrated a volume from one storage pool to another.

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation, click Storage.
3. In Select View, choose Volumes.
4. Click the volume name in the Volumes list, then click the Attach Disk button
5. In the Instance popup, choose the VM to which you want to attach the volume. You will only see instances to which you are allowed to attach volumes; for example, a user will see only instances created by that user, but the administrator will have more choices.
6. When the volume has been attached, you should be able to see it by clicking Instances, the instance name, and View Volumes.

Detaching and Moving Volumes

Note: This procedure is different from moving volumes from one storage pool to another as described in *"VM Storage Migration"*.

A volume can be detached from a guest VM and attached to another guest. Both CloudStack administrators and users can detach volumes from VMs and move them to other VMs.

If the two VMs are in different clusters, and the volume is large, it may take several minutes for the volume to be moved to the new VM.

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation bar, click Storage, and choose Volumes in Select View. Alternatively, if you know which VM the volume is attached to, you can click Instances, click the VM name, and click View Volumes.
3. Click the name of the volume you want to detach, then click the Detach Disk button.
4. To move the volume to another VM, follow the steps in *"Attaching a Volume"*.

VM Storage Migration

Supported in XenServer, KVM, and VMware.

Note: This procedure is different from moving disk volumes from one VM to another as described in *"Detaching and Moving Volumes"*.

You can migrate a virtual machine's root disk volume or any additional data disk volume from one storage pool to another in the same zone.

You can use the storage migration feature to achieve some commonly desired administration goals, such as balancing the load on storage pools and increasing the reliability of virtual machines by moving them away from any storage pool that is experiencing issues.

On XenServer and VMware, live migration of VM storage is enabled through CloudStack support for XenMotion and vMotion. Live storage migration allows VMs to be moved from one host to another, where the VMs are not located on storage shared between the two hosts. It provides the option to live migrate a VM's disks along with the VM itself. It is possible to migrate a VM from one XenServer resource pool / VMware cluster to another, or to migrate a VM whose disks are on local storage, or even to migrate a VM's disks from one storage repository to another, all while the VM is running.

Note: Because of a limitation in VMware, live migration of storage for a VM is allowed only if the source and target storage pool are accessible to the source host; that is, the host where the VM is running when the live migration operation is requested.

Migrating a Data Volume to a New Storage Pool

There are two situations when you might want to migrate a disk:

- Move the disk to new storage, but leave it attached to the same running VM.
- Detach the disk from its current VM, move it to new storage, and attach it to a new VM.

Migrating Storage For a Running VM

(Supported on XenServer and VMware)

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation bar, click Instances, click the VM name, and click View Volumes.
3. Click the volume you want to migrate.
4. Detach the disk from the VM. See *"Detaching and Moving Volumes"* but skip the "reattach" step at the end. You will do that after migrating to new storage.
5. Click the Migrate Volume button and choose the destination from the dropdown list.
6. Watch for the volume status to change to Migrating, then back to Ready.

Migrating Storage and Attaching to a Different VM

1. Log in to the CloudStack UI as a user or admin.
2. Detach the disk from the VM. See *"Detaching and Moving Volumes"* but skip the "reattach" step at the end. You will do that after migrating to new storage.
3. Click the Migrate Volume button and choose the destination from the dropdown list.
4. Watch for the volume status to change to Migrating, then back to Ready. You can find the volume by clicking Storage in the left navigation bar. Make sure that Volumes is displayed at the top of the window, in the Select View dropdown.
5. Attach the volume to any desired VM running in the same cluster as the new storage server. See *"Attaching a Volume"*

Migrating a VM Root Volume to a New Storage Pool

(XenServer, VMware) You can live migrate a VM's root disk from one storage pool to another, without stopping the VM first.

(KVM) When migrating the root disk volume, the VM must first be stopped, and users can not access the VM. After migration is complete, the VM can be restarted.

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation bar, click Instances, and click the VM name.
3. (KVM only) Stop the VM.
4. Click the Migrate button and choose the destination from the dropdown list.

   Note: If the VM's storage has to be migrated along with the VM, this will be noted in the host list. CloudStack will take care of the storage migration for you.

5. Watch for the volume status to change to Migrating, then back to Running (or Stopped, in the case of KVM). This can take some time.
6. (KVM only) Restart the VM.

Resizing Volumes

CloudStack provides the ability to resize data disks; CloudStack controls volume size by using disk offerings. This provides CloudStack administrators with the flexibility to choose how much space they want to make available to the end users. Volumes within the disk offerings with the same storage tag can be resized. For example, if you only want to offer 10, 50, and 100 GB offerings, the allowed resize should stay within those limits. That implies if you define a 10 GB, a 50 GB and a 100 GB disk offerings, a user can upgrade from 10 GB to 50 GB, or 50 GB to 100 GB. If you create a custom-sized disk offering, then you have the option to resize the volume by specifying a new, larger size.

Additionally, using the resizeVolume API, a data volume can be moved from a static disk offering to a custom disk offering with the size specified. This functionality allows those who might be billing by certain volume sizes or disk offerings to stick to that model, while providing the flexibility to migrate to whatever custom size necessary.

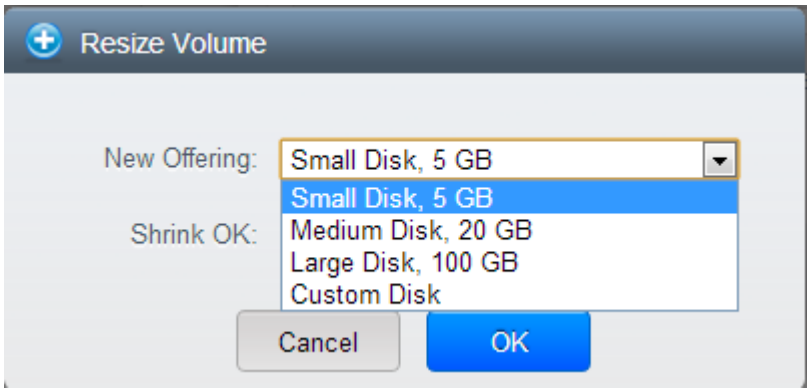
This feature is supported on KVM, XenServer, and VMware hosts. However, shrinking volumes is not supported on VMware hosts.

Before you try to resize a volume, consider the following:

- The VMs associated with the volume are stopped.
- The data disks associated with the volume are removed.
- When a volume is shrunk, the disk associated with it is simply truncated, and doing so would put its content at risk of data loss. Therefore, resize any partitions or file systems before you shrink a data disk so that all the data is moved off from that disk.

To resize a volume:

1. Log in to the CloudStack UI as a user or admin.
2. In the left navigation bar, click Storage.
3. In Select View, choose Volumes.
4. Select the volume name in the Volumes list, then click the Resize Volume button
5. In the Resize Volume pop-up, choose desired characteristics for the storage.



   (a) If you select Custom Disk, specify a custom size.

   (b) Click Shrink OK to confirm that you are reducing the size of a volume.

   This parameter protects against inadvertent shrinking of a disk, which might lead to the risk of data loss. You must sign off that you know what you are doing.

6. Click OK.

Reset VM to New Root Disk on Reboot

You can specify that you want to discard the root disk and create a new one whenever a given VM is rebooted. This is useful for secure environments that need a fresh start on every boot and for desktops that should not retain state. The IP address of the VM will not change due to this operation.

To enable root disk reset on VM reboot:

When creating a new service offering, set the parameter isVolatile to True. VMs created from this service offering will have their disks reset upon reboot. See "Creating a New Compute Offering".

Volume Deletion and Garbage Collection

The deletion of a volume does not delete the snapshots that have been created from the volume

When a VM is destroyed, data disk volumes that are attached to the VM are not deleted.

Volumes are permanently destroyed using a garbage collection process. The global configuration variables expunge.delay and expunge.interval determine when the physical deletion of volumes will occur.

- *expunge.delay*: determines how old the volume must be before it is destroyed, in seconds
- *expunge.interval*: determines how often to run the garbage collection check

Administrators should adjust these values depending on site policies around data retention.

5.9.5 Working with Volume Snapshots

(Supported for the following hypervisors: **XenServer**, **VMware vSphere**, and **KVM**)

CloudStack supports snapshots of disk volumes. Snapshots are a point-in-time capture of virtual machine disks. Memory and CPU states are not captured. If you are using the Oracle VM hypervisor, you can not take snapshots, since OVM does not support them.

Snapshots may be taken for volumes, including both root and data disks (except when the Oracle VM hypervisor is used, which does not support snapshots). The administrator places a limit on the number of stored snapshots per user. Users can create new volumes from the snapshot for recovery of particular files and they can create templates from snapshots to boot from a restored disk.

Users can create snapshots manually or by setting up automatic recurring snapshot policies. Users can also create disk volumes from snapshots, which may be attached to a VM like any other disk volume. Snapshots of both root disks and data disks are supported. However, CloudStack does not currently support booting a VM from a recovered root disk. A disk recovered from snapshot of a root disk is treated as a regular data disk; the data on recovered disk can be accessed by attaching the disk to a VM.

A completed snapshot is copied from primary storage to secondary storage, where it is stored until deleted or purged by newer snapshot.

How to Snapshot a Volume

1. Log in to the CloudStack UI as a user or administrator.
2. In the left navigation bar, click Storage.
3. In Select View, be sure Volumes is selected.
4. Click the name of the volume you want to snapshot.

5. Click the Snapshot button.

Automatic Snapshot Creation and Retention

(Supported for the following hypervisors: **XenServer**, **VMware vSphere**, and **KVM**)

Users can set up a recurring snapshot policy to automatically create multiple snapshots of a disk at regular intervals. Snapshots can be created on an hourly, daily, weekly, or monthly interval. One snapshot policy can be set up per disk volume. For example, a user can set up a daily snapshot at 02:30.

With each snapshot schedule, users can also specify the number of scheduled snapshots to be retained. Older snapshots that exceed the retention limit are automatically deleted. This user-defined limit must be equal to or lower than the global limit set by the CloudStack administrator. See "Globally Configured Limits". The limit applies only to those snapshots that are taken as part of an automatic recurring snapshot policy. Additional manual snapshots can be created and retained.

Incremental Snapshots and Backup

Snapshots are created on primary storage where a disk resides. After a snapshot is created, it is immediately backed up to secondary storage and removed from primary storage for optimal utilization of space on primary storage.

CloudStack does incremental backups for some hypervisors. When incremental backups are supported, every N backup is a full backup.

	VMware vSphere	Citrix XenServer	KVM
Support incremental backup	No	Yes	No

Volume Status

When a snapshot operation is triggered by means of a recurring snapshot policy, a snapshot is skipped if a volume has remained inactive since its last snapshot was taken. A volume is considered to be inactive if it is either detached or attached to a VM that is not running. CloudStack ensures that at least one snapshot is taken since the volume last became inactive.

When a snapshot is taken manually, a snapshot is always created regardless of whether a volume has been active or not.

Snapshot Restore

There are two paths to restoring snapshots. Users can create a volume from the snapshot. The volume can then be mounted to a VM and files recovered as needed. Alternatively, a template may be created from the snapshot of a root disk. The user can then boot a VM from this template to effect recovery of the root disk.

Snapshot Job Throttling

When a snapshot of a virtual machine is requested, the snapshot job runs on the same host where the VM is running or, in the case of a stopped VM, the host where it ran last. If many snapshots are requested for VMs on a single host, this can lead to problems with too many snapshot jobs overwhelming the resources of the host.

To address this situation, the cloud's root administrator can throttle how many snapshot jobs are executed simultaneously on the hosts in the cloud by using the global configuration setting concurrent.snapshots.threshold.perhost. By

using this setting, the administrator can better ensure that snapshot jobs do not time out and hypervisor hosts do not experience performance issues due to hosts being overloaded with too many snapshot requests.

Set concurrent.snapshots.threshold.perhost to a value that represents a best guess about how many snapshot jobs the hypervisor hosts can execute at one time, given the current resources of the hosts and the number of VMs running on the hosts. If a given host has more snapshot requests, the additional requests are placed in a waiting queue. No new snapshot jobs will start until the number of currently executing snapshot jobs falls below the configured limit.

The admin can also set job.expire.minutes to place a maximum on how long a snapshot request will wait in the queue. If this limit is reached, the snapshot request fails and returns an error message.

VMware Volume Snapshot Performance

When you take a snapshot of a data or root volume on VMware, CloudStack uses an efficient storage technique to improve performance.

A snapshot is not immediately exported from vCenter to a mounted NFS share and packaged into an OVA file format. This operation would consume time and resources. Instead, the original file formats (e.g., VMDK) provided by vCenter are retained. An OVA file will only be created as needed, on demand. To generate the OVA, CloudStack uses information in a properties file (*.ova.meta) which it stored along with the original snapshot data.

Note: For upgrading customers: This process applies only to newly created snapshots after upgrade to CloudStack 4.2. Snapshots that have already been taken and stored in OVA format will continue to exist in that format, and will continue to work as expected.

5.10 Working with System Virtual Machines

CloudStack uses several types of system virtual machines to perform tasks in the cloud. In general CloudStack manages these system VMs and creates, starts, and stops them as needed based on scale and immediate needs. However, the administrator should be aware of them and their roles to assist in debugging issues.

5.10.1 The System VM Template

The System VMs come from a single template. The System VM has the following characteristics:

- Debian 7.8 ("wheezy"), 3.2.0 kernel with the latest security patches from the Debian security APT repository
- Has a minimal set of packages installed thereby reducing the attack surface
- 64-bit for enhanced performance on Xen/VMWare
- pvops kernel with Xen PV drivers, KVM virtio drivers, and VMware tools for optimum performance on all hypervisors
- Xen tools inclusion allows performance monitoring
- Latest versions of HAProxy, iptables, IPsec, and Apache from debian repository ensures improved security and speed
- Latest version of JRE from Sun/Oracle ensures improved security and speed

5.10.2 Changing the Default System VM Template

Using the 64-bit template should be use with a System Offering of at least 512MB of memory.

1. Based on the hypervisor you use, download the 64-bit template from the following location:

Hypervisor	Download Location
XenServer	http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-xen.vhd.bz2
KVM	http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-kvm.qcow2.bz2
VMware	http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-vmware.ova
Hyper-V	http://download.cloudstack.org/systemvm/4.11/systemvmtemplate-4.11.1-hyperv.vhd.zip

2. As an administrator, log in to the CloudStack UI
3. Register the 64 bit template.

   For example: KVM64bitTemplate
4. While registering the template, select Routing.
5. Navigate to Infrastructure > Zone > Settings.
6. Set the name of the 64-bit template, KVM64bitTemplate, in the *''router.template.kvm''* global parameter.

   If you are using a XenServer 64-bit template, set the name in the *''router.template.xen''* global parameter.

   Any new virtual router created in this Zone automatically picks up this template.
7. Restart the Management Server.

5.10.3 Multiple System VM Support for VMware

Every CloudStack zone has single System VM for template processing tasks such as downloading templates, uploading templates, and uploading ISOs. In a zone where VMware is being used, additional System VMs can be launched to process VMware-specific tasks such as taking snapshots and creating private templates. The CloudStack management server launches additional System VMs for VMware-specific tasks as the load increases. The management server monitors and weights all commands sent to these System VMs and performs dynamic load balancing and scaling-up of more System VMs.

5.10.4 Console Proxy

The Console Proxy is a type of System Virtual Machine that has a role in presenting a console view via the web UI. It connects the user's browser to the VNC port made available via the hypervisor for the console of the guest. Both the administrator and end user web UIs offer a console connection.

Clicking a console icon brings up a new window. The AJAX code downloaded into that window refers to the public IP address of a console proxy VM. There is exactly one public IP address allocated per console proxy VM. The AJAX application connects to this IP. The console proxy then proxies the connection to the VNC port for the requested VM on the Host hosting the guest.

Note: The hypervisors will have many ports assigned to VNC usage so that multiple VNC sessions can occur simultaneously.

There is never any traffic to the guest virtual IP, and there is no need to enable VNC within the guest.

The console proxy VM will periodically report its active session count to the Management Server. The default reporting interval is five seconds. This can be changed through standard Management Server configuration with the parameter consoleproxy.loadscan.interval.

Assignment of guest VM to console proxy is determined by first determining if the guest VM has a previous session associated with a console proxy. If it does, the Management Server will assign the guest VM to the target Console Proxy VM regardless of the load on the proxy VM. Failing that, the first available running Console Proxy VM that has the capacity to handle new sessions is used.

Console proxies can be restarted by administrators but this will interrupt existing console sessions for users.

Using a SSL Certificate for the Console Proxy

By default, the console viewing functionality uses plaintext HTTP. In any production environment, the console proxy connection should be encrypted via SSL at the mininum.

A CloudStack administrator has 2 ways to secure the console proxy communication with SSL:

- Set up a SSL wild-card certificate and domain name resolution
- Set up SSL certificate for specific FQDN and configure load-balancer

Changing the Console Proxy SSL Certificate and Domain

The administrator can configure SSL encryption by selecting a domain and uploading a new SSL certificate and private key. The domain must run a DNS service that is capable of resolving queries for addresses of the form aaa-bbb-ccc-ddd.your.domain to an IPv4 IP address in the form aaa.bbb.ccc.ddd, for example, 202.8.44.1. To change the console proxy domain, SSL certificate, and private key:

1. Set up dynamic name resolution or populate all possible DNS names in your public IP range into your existing DNS server with the format aaa-bbb-ccc-ddd.consoleproxy.company.com -> aaa.bbb.ccc.ddd.

   Note: In these steps you will notice *consoleproxy.company.com* -For security best practices, we recommend creating a wildcard SSL certificate on a separate subdomain so in the event that the certificate is compromised, a malicious user cannot impersonate a company.com domain.

2. Generate the private key and certificate signing request (CSR). When you are using openssl to generate private/public key pairs and CSRs, for the private key that you are going to paste into the CloudStack UI, be sure to convert it into PKCS#8 format.

   (a) Generate a new 2048-bit private key

   ```
   openssl genrsa -des3 -out yourprivate.key 2048
   ```

   (b) Generate a new certificate CSR. Ensure the creation of a wildcard certificate, eg `*.consoleproxy.company.com`

   ```
   openssl req -new -key yourprivate.key -out yourcertificate.csr
   ```

   (c) Head to the website of your favorite trusted Certificate Authority, purchase an SSL certificate, and submit the CSR. You should receive a valid certificate in return

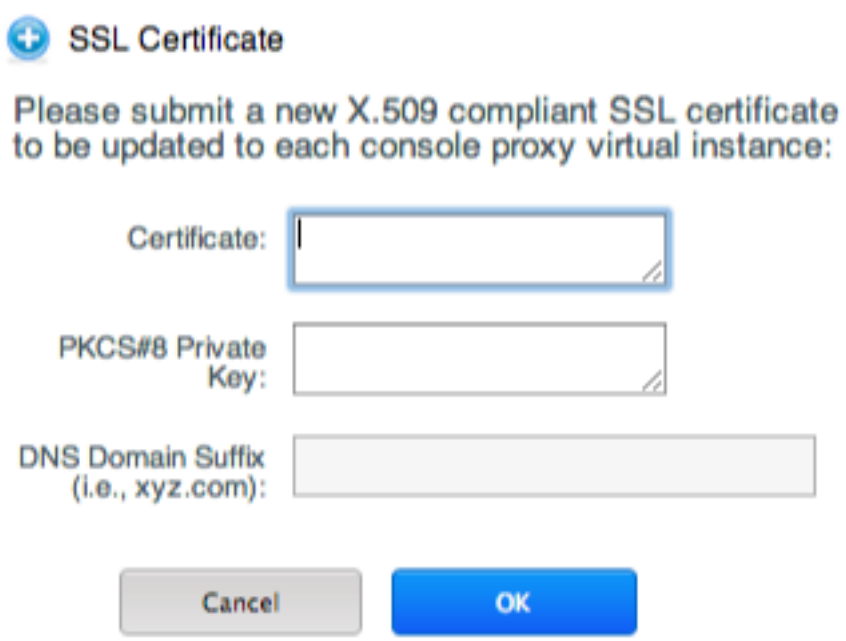
   (d) Convert your private key format into PKCS#8 encrypted format.

   ```
   openssl pkcs8 -topk8 -in yourprivate.key -out yourprivate.pkcs8.encrypted.key
   ```

(e) Convert your PKCS#8 encrypted private key into the PKCS#8 format that is compliant with CloudStack

```
openssl pkcs8 -in yourprivate.pkcs8.encrypted.key -out yourprivate.pkcs8.key
```

3. In the Update SSL Certificate screen of the CloudStack UI, paste the following:
   - The certificate you've just generated.
   - The private key you've just generated.
   - The desired domain name, prefixed with `*.`; for example, `*.consoleproxy.company.com`



4. This stops all currently running console proxy VMs, then restarts them with the new certificate and key. Users might notice a brief interruption in console availability.

The Management Server generates URLs of the form "aaa-bbb-ccc-ddd.consoleproxy.company.com" after this change is made. The new console requests will be served with the new DNS domain name, certificate, and key.

Uploading ROOT CA and Intermediate CA

If you need to upload custom certificate with ROOT CA and intermediate CA, you can find more details here: https://cwiki.apache.org/confluence/display/CLOUDSTACK/Procedure+to+Replace+realhostip.com+with+Your+Own+Domain+Name

IMPORTANT NOTES:

In order to avoid errors and problems while uploading custom certificates, please check following:

1. While doing URL encoding of ROOT CA and any Intermediate CA, be sure that the plus signs ("+") inside certificates are not replaced by space (" "), because some URL/string encoding tools tend to do that.

2. If you are renewing certificates it might happen you need to upload new ROOT CA and Intermediate CA, together with new Server Certificate and key. In this case please be sure to use same names for certificates during API upload of certificate, example:

http://123.123.123.123:8080/client/api?command=uploadCustomCertificate&...&name=root1...
http://123.123.123.123:8080/client/api?command=uploadCustomCertificate&...&name=intermed1...

Here names are "root1" and "intermed1". If you used other names previously, please check the cloud.keystore table to obtain used names.

If you still have problems and folowing errors in management.log while destroying CPVM:

- Unable to build keystore for CPVMCertificate due to CertificateException
- Cold not find and construct a valid SSL certificate

that means that still some of the Root/intermediate/server certificates or the key is not in a good format, or incorrectly encoded or multiply Root CA/Intemediate CA present in database by mistake.

Other way to renew Certificates (Root,Intermediates,Server certificates and key) - although not recommended unless you fill comfortable - is to directly edit the database, while still respect the main requirement that the private key is PKCS8 encoded, while Root CA, Intemediate and Server certificates are still in default PEM format (no URL encoding needed here). After editing the database, please restart management server, and destroy SSVM and CPVM after that, so the new SSVM and CPVM with new certificates are created.

Load-balancing Console Proxies

An alternative to using dynamic DNS or creating a range of DNS entries as described in the last section would be to create a SSL certificate for a specific domain name, configure CloudStack to use that particular FQDN, and then configure a load balancer to load balance the console proxy's IP address behind the FQDN. As the functionality for this is still new, please see https://cwiki.apache.org/confluence/display/CLOUDSTACK/Realhost+IP+changes for more details.

5.10.5 Virtual Router

The virtual router is a type of System Virtual Machine. The virtual router is one of the most frequently used service providers in CloudStack. The end user has no direct access to the virtual router. Users can ping the virtual router and take actions that affect it (such as setting up port forwarding), but users do not have SSH access into the virtual router.

There is no mechanism for the administrator to log in to the virtual router. Virtual routers can be restarted by administrators, but this will interrupt public network access and other services for end users. A basic test in debugging networking issues is to attempt to ping the virtual router from a guest VM. Some of the characteristics of the virtual router are determined by its associated system service offering.

Configuring the Virtual Router

You can set the following:

- IP range
- Supported network services
- Default domain name for the network serviced by the virtual router
- Gateway IP address
- How often CloudStack fetches network usage statistics from CloudStack virtual routers. If you want to collect traffic metering data from the virtual router, set the global configuration parameter router.stats.interval. If you are not using the virtual router to gather network usage statistics, set it to 0.

Upgrading a Virtual Router with System Service Offerings

When CloudStack creates a virtual router, it uses default settings which are defined in a default system service offering. See *"System Service Offerings"*. All the virtual routers in a single guest network use the same system service offering. You can upgrade the capabilities of the virtual router by creating and applying a custom system service offering.

1. Define your custom system service offering. See *"Creating a New System Service Offering"*. In System VM Type, choose Domain Router.
2. Associate the system service offering with a network offering. See "Creating a New Network Offering".
3. Apply the network offering to the network where you want the virtual routers to use the new system service offering. If this is a new network, follow the steps in Adding an Additional Guest Network on page 66. To change the service offering for existing virtual routers, follow the steps in "Changing the Network Offering on a Guest Network".

Best Practices for Virtual Routers

- WARNING: Restarting a virtual router from a hypervisor console deletes all the iptables rules. To work around this issue, stop the virtual router and start it from the CloudStack UI.
- **Warning:** Do not use the destroyRouter API when only one router is available in the network, because restartNetwork API with the cleanup=false parameter can't recreate it later. If you want to destroy and recreate the single router available in the network, use the restartNetwork API with the cleanup=true parameter.

Service Monitoring Tool for Virtual Router

Various services running on the CloudStack virtual routers can be monitored by using a Service Monitoring tool. The tool ensures that services are successfully running until CloudStack deliberately disables them. If a service goes down, the tool automatically restarts the service, and if that does not help bringing up the service, an alert as well as an event is generated indicating the failure. A new global parameter, `network.router.enableservicemonitoring`, has been introduced to control this feature. The default value is false, implies, monitoring is disabled. When you enable, ensure that the Management Server and the router are restarted.

Monitoring tool can help to start a VR service, which is crashed due to an unexpected reason. For example:

- The services crashed due to defects in the source code.
- The services that are terminated by the OS when memory or CPU is not sufficiently available for the service.

Note: Only those services with daemons are monitored. The services that are failed due to errors in the service/daemon configuration file cannot be restarted by the Monitoring tool. VPC networks are not supported.

The following services are monitored in a VR:

- DNS
- HA Proxy
- SSH
- Apache Web Server

The following networks are supported:

- Isolated Networks

- Shared Networks in both Advanced and Basic zone

Note: VPC networks are not supported

This feature is supported on the following hypervisors: XenServer, VMware, and KVM.

Enhanced Upgrade for Virtual Routers

Upgrading VR is made flexible. The CloudStack administrators will be able to control the sequence of the VR upgrades. The sequencing is based on Infrastructure hierarchy, such as by Cluster, Pod, or Zone, and Administrative (Account) hierarchy, such as by Tenant or Domain. As an administrator, you can also determine when a particular customer service, such as VR, is upgraded within a specified upgrade interval. Upgrade operation is enhanced to increase the upgrade speed by allowing as many upgrade operations in parallel as possible.

During the entire duration of the upgrade, users cannot launch new services or make changes to an existing service.

Additionally, using multiple versions of VRs in a single instance is supported. In the Details tab of a VR, you can view the version and whether it requires upgrade. During the Management Server upgrade, CloudStack checks whether VR is at the latest version before performing any operation on the VR. To support this, a new global parameter, *''router.version.check''*, has been added. This parameter is set to true by default, which implies minimum required version is checked before performing any operation. No operation is performed if the VR is not at the required version. Services of the older version VR continue to be available, but no further operations can be performed on the VR until it is upgraded to the latest version. This will be a transient state until the VR is upgraded. This will ensure that the availability of VR services and VR state is not impacted due to the Management Server upgrade.

The following service will be available even if the VR is not upgraded. However, no changes for any of the services can be sent to the VR, until it is upgraded:

- SecurityGroup
- UserData
- DHCP
- DNS
- LB
- Port Forwarding
- VPN
- Static NAT
- Source NAT
- Firewall
- Gateway
- NetworkACL

Supported Virtual Routers

- VR
- VPC VR
- Redundant VR

Upgrading Virtual Routers

1. Download the latest System VM template.
2. Download the latest System VM to all the primary storage pools.
3. Upgrade the Management Server.
4. Upgrade CPVM and SSVM either from the UI or by using the following script:

   ```
   # cloudstack-sysvmadm -d <IP address> -u cloud -p -s
   ```

   Even when the VRs are still on older versions, existing services will continue to be available to the VMs. The Management Server cannot perform any operations on the VRs until they are upgraded.
5. Selectively upgrade the VRs:
   (a) Log in to the CloudStack UI as the root administrator.
   (b) In the left navigation, choose Infrastructure.
   (c) On Virtual Routers, click View More.

   All the VRs are listed in the Virtual Routers page.

   (d) In Select View drop-down, select desired grouping based on your requirement.

   You can use either of the following:

   - Group by zone
   - Group by pod
   - Group by cluster
   - Group by account

   (e) Click the group which has the VRs to be upgraded.

   For example, if you have selected Group by zone, select the name of the desired zone.

   (f) Click the Upgrade button to upgrade all the VRs.
   (g) Click OK to confirm.

5.10.6 Secondary Storage VM

In addition to the hosts, CloudStack's Secondary Storage VM mounts and writes to secondary storage.

Submissions to secondary storage go through the Secondary Storage VM. The Secondary Storage VM can retrieve templates and ISO images from URLs using a variety of protocols.

The secondary storage VM provides a background task that takes care of a variety of secondary storage activities: downloading a new template to a Zone, copying templates between Zones, and snapshot backups.

The administrator can log in to the secondary storage VM if needed.

5.11 Working with Usage

5.11.1 Working with Usage

The Usage Server is an optional, separately-installed part of CloudStack that provides aggregated usage records which you can use to create billing integration for CloudStack. The Usage Server works by taking data from the events log and creating summary usage records that you can access using the listUsageRecords API call.

The usage records show the amount of resources, such as VM run time or template storage space, consumed by guest instances.

The Usage Server runs at least once per day. It can be configured to run multiple times per day.

Configuring the Usage Server

To configure the usage server:

1. Be sure the Usage Server has been installed. This requires extra steps beyond just installing the CloudStack software. See Installing the Usage Server (Optional) in the Advanced Installation Guide.
2. Log in to the CloudStack UI as administrator.
3. Click Global Settings.
4. In Search, type usage. Find the configuration parameter that controls the behavior you want to set. See the table below for a description of the available parameters.
5. In Actions, click the Edit icon.
6. Type the desired value and click the Save icon.
7. Restart the Management Server (as usual with any global configuration change) and also the Usage Server:

```
# service cloudstack-management restart
# service cloudstack-usage restart
```

The following table shows the global configuration settings that control the behavior of the Usage Server.

Parameter Name Description

enable.usage.server Whether the Usage Server is active.

usage.aggregation.timezone

Time zone of usage records. Set this if the usage records and daily job execution are in different time zones. For example, with the following settings, the usage job will run at PST 00:15 and generate usage records for the 24 hours from 00:00:00 GMT to 23:59:59 GMT:

```
usage.stats.job.exec.time = 00:15
usage.execution.timezone = PST
usage.aggregation.timezone = GMT
```

Valid values for the time zone are specified in the *Time Zones* section

Default: GMT

usage.execution.timezone

The time zone of usage.stats.job.exec.time. Valid values for the time zone are specified in the *Time Zones* section

Default: The time zone of the management server.

usage.sanity.check.interval

The number of days between sanity checks. Set this in order to periodically search for records with erroneous data before issuing customer invoices. For example, this checks for VM usage records created after the VM was destroyed, and similar checks for templates, volumes, and so on. It also checks for usage times longer than the aggregation range. If any issue is found, the alert ALERT_TYPE_USAGE_SANITY_RESULT = 21 is sent.

usage.stats.job.aggregation.range

The time period in minutes between Usage Server processing jobs. For example, if you set it to 1440, the Usage Server will run once per day. If you set it to 600, it will run every ten hours. In general, when a Usage Server job runs, it processes all events generated since usage was last run.

There is special handling for the case of 1440 (once per day). In this case the Usage Server does not necessarily process all records since Usage was last run. CloudStack assumes that you require processing once per day for the previous, complete day's records. For example, if the current day is October 7, then it is assumed you would like to process records for October 6, from midnight to midnight. CloudStack assumes this "midnight to midnight" is relative to the usage.execution.timezone.

Default: 1440

usage.stats.job.exec.time

The time when the Usage Server processing will start. It is specified in 24-hour format (HH:MM) in the time zone of the server, which should be GMT. For example, to start the Usage job at 10:30 GMT, enter "10:30".

If usage.stats.job.aggregation.range is also set, and its value is not 1440, then its value will be added to usage.stats.job.exec.time to get the time to run the Usage Server job again. This is repeated until 24 hours have elapsed, and the next day's processing begins again at usage.stats.job.exec.time.

Default: 00:15.

For example, suppose that your server is in GMT, your user population is predominantly in the East Coast of the United States, and you would like to process usage records every night at 2 AM local (EST) time. Choose these settings:

- enable.usage.server = true
- usage.execution.timezone = America/New_York
- usage.stats.job.exec.time = 07:00. This will run the Usage job at 2:00 AM EST. Note that this will shift by an hour as the East Coast of the U.S. enters and exits Daylight Savings Time.
- usage.stats.job.aggregation.range = 1440

With this configuration, the Usage job will run every night at 2 AM EST and will process records for the previous day's midnight-midnight as defined by the EST (America/New_York) time zone.

Note: Because the special value 1440 has been used for usage.stats.job.aggregation.range, the Usage Server will ignore the data between midnight and 2 AM. That data will be included in the next day's run.

Setting Usage Limits

CloudStack provides several administrator control points for capping resource usage by users. Some of these limits are global configuration parameters. Others are applied at the ROOT domain and may be overridden on a per-account basis.

Globally Configured Limits

In a zone, the guest virtual network has a 24 bit CIDR by default. This limits the guest virtual network to 254 running instances. It can be adjusted as needed, but this must be done before any instances are created in the zone. For example, 10.1.1.0/22 would provide for ~1000 addresses.

The following table lists limits set in the Global Configuration:

Parameter Name	Definition
max.account.public.ips	Number of public IP addresses that can be owned by an account
max.account.snapshots	Number of snapshots that can exist for an account
max.account.templates	Number of templates that can exist for an account
max.account.user.vms	Number of virtual machine instances that can exist for an account
max.account.volumes	Number of disk volumes that can exist for an account
max.template.iso.size	Maximum size for a downloaded template or ISO in GB
max.volume.size.gb	Maximum size for a volume in GB
network.throttling.rate	Default data transfer rate in megabits per second allowed per user (supported on XenServer)
snapshot.max.hourly	Maximum recurring hourly snapshots to be retained for a volume. If the limit is reached, early snapshots from the start of the hour are deleted so that newer ones can be saved. This limit does not apply to manual snapshots. If set to 0, recurring hourly snapshots can not be scheduled
snapshot.max.daily	Maximum recurring daily snapshots to be retained for a volume. If the limit is reached, snapshots from the start of the day are deleted so that newer ones can be saved. This limit does not apply to manual snapshots. If set to 0, recurring daily snapshots can not be scheduled
snapshot.max.weekly	Maximum recurring weekly snapshots to be retained for a volume. If the limit is reached, snapshots from the beginning of the week are deleted so that newer ones can be saved. This limit does not apply to manual snapshots. If set to 0, recurring weekly snapshots can not be scheduled
snapshot.max.monthly	Maximum recurring monthly snapshots to be retained for a volume. If the limit is reached, snapshots from the beginning of the month are deleted so that newer ones can be saved. This limit does not apply to manual snapshots. If set to 0, recurring monthly snapshots can not be scheduled.

To modify global configuration parameters, use the global configuration screen in the CloudStack UI. See Setting Global Configuration Parameters

Limiting Resource Usage

CloudStack allows you to control resource usage based on the types of resources, such as CPU, RAM, Primary storage, and Secondary storage. A new set of resource types has been added to the existing pool of resources to support the new customization model—need-basis usage, such as large VM or small VM. The new resource types are now broadly classified as CPU, RAM, Primary storage, and Secondary storage. The root administrator is able to impose resource usage limit by the following resource types for Domain, Project, and Accounts.

- CPUs
- Memory (RAM)
- Primary Storage (Volumes)
- Secondary Storage (Snapshots, Templates, ISOs)

To control the behaviour of this feature, the following configuration parameters have been added:

Parameter Name	Description
max.account.cpus	Maximum number of CPU cores that can be used for an account. Default is 40.
max.account.ram (MB)	Maximum RAM that can be used for an account. Default is 40960.
max.account.primary.storage (GB)	Maximum primary storage space that can be used for an account. Default is 200.
max.account.secondary.storage (GB)	Maximum secondary storage space that can be used for an account. Default is 400.
max.project.cpus	Maximum number of CPU cores that can be used for an account. Default is 40.
max.project.ram (MB)	Maximum RAM that can be used for an account. Default is 40960.
max.project.primary.storage (GB)	Maximum primary storage space that can be used for an account. Default is 200.
max.project.secondary.storage (GB)	Maximum secondary storage space that can be used for an account. Default is 400.

User Permission

The root administrator, domain administrators and users are able to list resources. Ensure that proper logs are maintained in the `vmops.log` and `api.log` files.

- The root admin will have the privilege to list and update resource limits.
- The domain administrators are allowed to list and change these resource limits only for the sub-domains and accounts under their own domain or the sub-domains.
- The end users will the privilege to list resource limits. Use the listResourceLimits API.

Limit Usage Considerations

- Primary or Secondary storage space refers to the stated size of the volume and not the physical size— the actual consumed size on disk in case of thin provisioning.
- If the admin reduces the resource limit for an account and set it to less than the resources that are currently being consumed, the existing VMs/templates/volumes are not destroyed. Limits are imposed only if the user under that account tries to execute a new operation using any of these resources. For example, the existing behavior in the case of a VM are:
  - migrateVirtualMachine: The users under that account will be able to migrate the running VM into any other host without facing any limit issue.
  - recoverVirtualMachine: Destroyed VMs cannot be recovered.
- For any resource type, if a domain has limit X, sub-domains or accounts under that domain can have there own limits. However, the sum of resource allocated to a sub-domain or accounts under the domain at any point of time should not exceed the value X.

  For example, if a domain has the CPU limit of 40 and the sub-domain D1 and account A1 can have limits of 30 each, but at any point of time the resource allocated to D1 and A1 should not exceed the limit of 40.
- If any operation needs to pass through two of more resource limit check, then the lower of 2 limits will be enforced, For example: if an account has the VM limit of 10 and CPU limit of 20, and a user under that account requests 5 VMs of 4 CPUs each. The user can deploy 5 more VMs because VM limit is 10. However, the user cannot deploy any more instances because the CPU limit has been exhausted.

Limiting Resource Usage in a Domain

CloudStack allows the configuration of limits on a domain basis. With a domain limit in place, all users still have their account limits. They are additionally limited, as a group, to not exceed the resource limits set on their domain. Domain limits aggregate the usage of all accounts in the domain as well as all the accounts in all the sub-domains of that domain. Limits set at the root domain level apply to the sum of resource usage by the accounts in all the domains and sub-domains below that root domain.

To set a domain limit:

1. Log in to the CloudStack UI.
2. In the left navigation tree, click Domains.
3. Select the domain you want to modify. The current domain limits are displayed.

   A value of -1 shows that there is no limit in place.
4. Click the Edit button
5. Edit the following as per your requirement:
   - Parameter Name
   - Description
   - Instance Limits

     The number of instances that can be used in a domain.
   - Public IP Limits

     The number of public IP addresses that can be used in a domain.
   - Volume Limits

     The number of disk volumes that can be created in a domain.
   - Snapshot Limits

     The number of snapshots that can be created in a domain.
   - Template Limits

     The number of templates that can be registered in a domain.
   - VPC limits

     The number of VPCs that can be created in a domain.
   - CPU limits

     The number of CPU cores that can be used for a domain.
   - Memory limits (MB)

     The number of RAM that can be used for a domain.
   - Primary Storage limits (GB)

     The primary storage space that can be used for a domain.
   - Secondary Storage limits (GB)

     The secondary storage space that can be used for a domain.
6. Click Apply.

Default Account Resource Limits

You can limit resource use by accounts. The default limits are set by using Global configuration parameters, and they affect all accounts within a cloud. The relevant parameters are those beginning with max.account, for example: max.account.snapshots.

To override a default limit for a particular account, set a per-account resource limit.

1. Log in to the CloudStack UI.
2. In the left navigation tree, click Accounts.
3. Select the account you want to modify. The current limits are displayed.

   A value of -1 shows that there is no limit in place.

4. Click the Edit button.
5. Edit the following as per your requirement:
   - Parameter Name
   - Description
   - Instance Limits

     The number of instances that can be used in an account.

     The default is 20.
   - Public IP Limits

     The number of public IP addresses that can be used in an account.

     The default is 20.
   - Volume Limits

     The number of disk volumes that can be created in an account.

     The default is 20.
   - Snapshot Limits

     The number of snapshots that can be created in an account.

     The default is 20.
   - Template Limits

     The number of templates that can be registered in an account.

     The default is 20.
   - VPC limits

     The number of VPCs that can be created in an account.

     The default is 20.
   - CPU limits

     The number of CPU cores that can be used for an account.

     The default is 40.

- Memory limits (MB)

  The number of RAM that can be used for an account.

  The default is 40960.
- Primary Storage limits (GB)

  The primary storage space that can be used for an account.

  The default is 200.
- Secondary Storage limits (GB)

  The secondary storage space that can be used for an account.

  The default is 400.

6. Click Apply.

Usage Record Format

Virtual Machine Usage Record Format

For running and allocated virtual machine usage, the following fields exist in a usage record:

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for VM running time)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- virtualMachineId – The ID of the virtual machine
- name – The name of the virtual machine
- offeringid – The ID of the service offering
- templateid – The ID of the template or the ID of the parent template. The parent template value is present when the current template was created from a volume.
- usageid – Virtual machine
- type – Hypervisor
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

Network Usage Record Format

For network usage (bytes sent/received), the following fields exist in a usage record.

- account – name of the account
- accountid – ID of the account

- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – Device ID (virtual router ID or external device ID)
- type – Device type (domain router, external load balancer, etc.)
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

IP Address Usage Record Format

For IP address usage the following fields exist in a usage record.

- account - name of the account
- accountid - ID of the account
- domainid - ID of the domain in which this account resides
- zoneid - Zone where the usage occurred
- description - A string describing what the usage record is tracking
- usage - String representation of the usage, including the units of usage
- usagetype - A number representing the usage type (see Usage Types)
- rawusage - A number representing the actual usage in hours
- usageid - IP address ID
- startdate, enddate - The range of time for which the usage is aggregated; see Dates in the Usage Record
- issourcenat - Whether source NAT is enabled for the IP address
- iselastic - True if the IP address is elastic.

Disk Volume Usage Record Format

For disk volumes, the following fields exist in a usage record.

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – The volume ID

- offeringid – The ID of the disk offering
- type – Hypervisor
- templateid – ROOT template ID
- size – The amount of storage allocated
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

Template, ISO, and Snapshot Usage Record Format

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – The ID of the the template, ISO, or snapshot
- offeringid – The ID of the disk offering
- templateid – – Included only for templates (usage type 7). Source template ID.
- size – Size of the template, ISO, or snapshot
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

Load Balancer Policy or Port Forwarding Rule Usage Record Format

- account - name of the account
- accountid - ID of the account
- domainid - ID of the domain in which this account resides
- zoneid - Zone where the usage occurred
- description - A string describing what the usage record is tracking
- usage - String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype - A number representing the usage type (see Usage Types)
- rawusage - A number representing the actual usage in hours
- usageid - ID of the load balancer policy or port forwarding rule
- usagetype - A number representing the usage type (see Usage Types)
- startdate, enddate - The range of time for which the usage is aggregated; see Dates in the Usage Record

Network Offering Usage Record Format

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – ID of the network offering
- usagetype – A number representing the usage type (see Usage Types)
- offeringid – Network offering ID
- virtualMachineId – The ID of the virtual machine
- virtualMachineId – The ID of the virtual machine
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

VPN User Usage Record Format

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – VPN user ID
- usagetype – A number representing the usage type (see Usage Types)
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

Usage Types

The following table shows all usage types.

Type ID	Type Name	Description
1	RUNNING_VM	Tracks the total running time of a VM per usage record period. If the VM is upgraded during the usage period, you will get a separate Usage Record for the new upgraded VM.
2	ALLOCATED_VM	Tracks the total time the VM has been created to the time when it has been destroyed. This usage type is also useful in determining usage for specific templates such as Windows-based templates.
3	IP_ADDRESS	Tracks the public IP address owned by the account.
4	NETWORK_BYTES_SENT	Tracks the total number of bytes sent by all the VMs for an account. Cloud.com does not currently track network traffic per VM.
5	NETWORK_BYTES_RECEIVED	Tracks the total number of bytes received by all the VMs for an account. Cloud.com does not currently track network traffic per VM.
6	VOLUME	Tracks the total time a disk volume has been created to the time when it has been destroyed.
7	TEMPLATE	Tracks the total time a template (either created from a snapshot or uploaded to the cloud) has been created to the time it has been destroyed. The size of the template is also returned.
8	ISO	Tracks the total time an ISO has been uploaded to the time it has been removed from the cloud. The size of the ISO is also returned.
9	SNAPSHOT	Tracks the total time from when a snapshot has been created to the time it have been destroyed.
11	LOAD_BALANCER_POLICY	Tracks the total time a load balancer policy has been created to the time it has been removed. Cloud.com does not track whether a VM has been assigned to a policy.
12	PORT_FORWARDING_RULE	Tracks the time from when a port forwarding rule was created until the time it was removed.
13	NETWORK_OFFERING	The time from when a network offering was assigned to a VM until it is removed.
14	VPN_USERS	The time from when a VPN user is created until it is removed.

Example response from listUsageRecords

All CloudStack API requests are submitted in the form of a HTTP GET/POST with an associated command and any parameters. A request is composed of the following whether in HTTP or HTTPS:

```
<listusagerecordsresponse>
   <count>1816</count>
   <usagerecord>
      <account>user5</account>
      <accountid>10004</accountid>
      <domainid>1</domainid>
      <zoneid>1</zoneid>
      <description>i-3-4-WC running time (ServiceOffering: 1) (Template: 3)</
↪description>
      <usage>2.95288 Hrs</usage>
      <usagetype>1</usagetype>
      <rawusage>2.95288</rawusage>
      <virtualmachineid>4</virtualmachineid>
      <name>i-3-4-WC</name>
      <offeringid>1</offeringid>
      <templateid>3</templateid>
      <usageid>245554</usageid>
      <type>XenServer</type>
      <startdate>2009-09-15T00:00:00-0700</startdate>
```

(continues on next page)

(continued from previous page)

```
      <enddate>2009-09-18T16:14:26-0700</enddate>
   </usagerecord>

   ... (1,815 more usage records)
</listusagerecordsresponse>
```

Dates in the Usage Record

Usage records include a start date and an end date. These dates define the period of time for which the raw usage number was calculated. If daily aggregation is used, the start date is midnight on the day in question and the end date is 23:59:59 on the day in question (with one exception; see below). A virtual machine could have been deployed at noon on that day, stopped at 6pm on that day, then started up again at 11pm. When usage is calculated on that day, there will be 7 hours of running VM usage (usage type 1) and 12 hours of allocated VM usage (usage type 2). If the same virtual machine runs for the entire next day, there will 24 hours of both running VM usage (type 1) and allocated VM usage (type 2).

Note: The start date is not the time a virtual machine was started, and the end date is not the time when a virtual machine was stopped. The start and end dates give the time range within which usage was calculated.

For network usage, the start date and end date again define the range in which the number of bytes transferred was calculated. If a user downloads 10 MB and uploads 1 MB in one day, there will be two records, one showing the 10 megabytes received and one showing the 1 megabyte sent.

There is one case where the start date and end date do not correspond to midnight and 11:59:59pm when daily aggregation is used. This occurs only for network usage records. When the usage server has more than one day's worth of unprocessed data, the old data will be included in the aggregation period. The start date in the usage record will show the date and time of the earliest event. For other types of usage, such as IP addresses and VMs, the old unprocessed data is not included in daily aggregation.

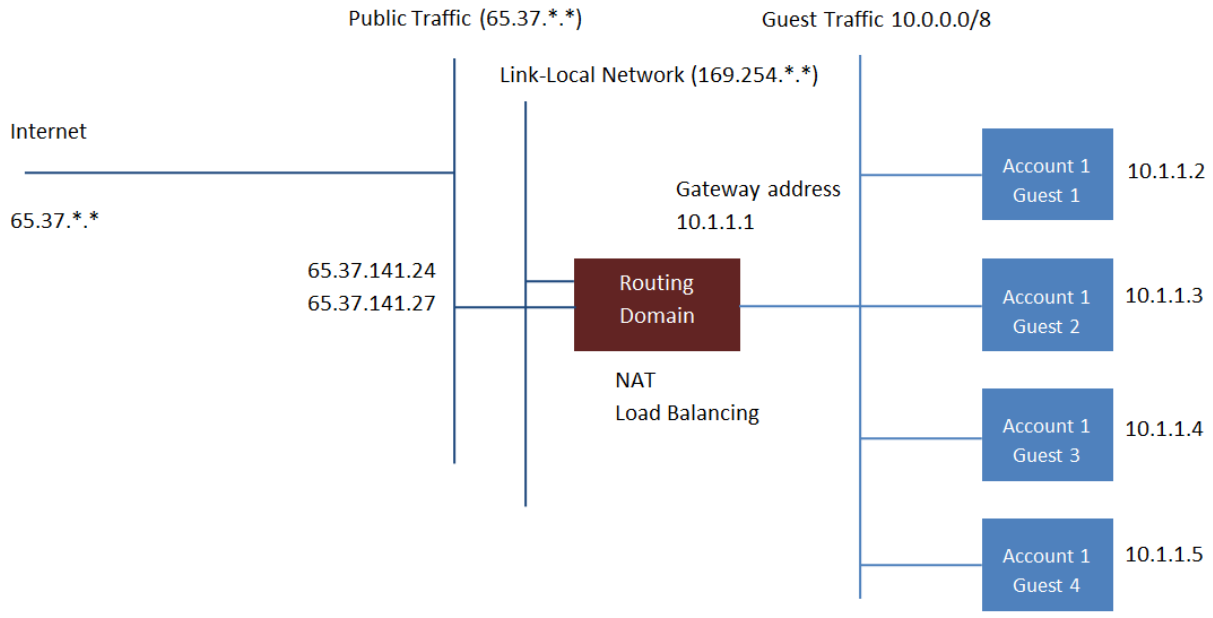
5.12 Managing Networks and Traffic

In a CloudStack, guest VMs can communicate with each other using shared infrastructure with the security and user perception that the guests have a private LAN. The CloudStack virtual router is the main component providing networking features for guest traffic.

5.12.1 Guest Traffic

A network can carry guest traffic only between VMs within one zone. Virtual machines in different zones cannot communicate with each other using their IP addresses; they must communicate with each other by routing through a public IP address.

See a typical guest traffic setup given below:



Guest Traffic Setup

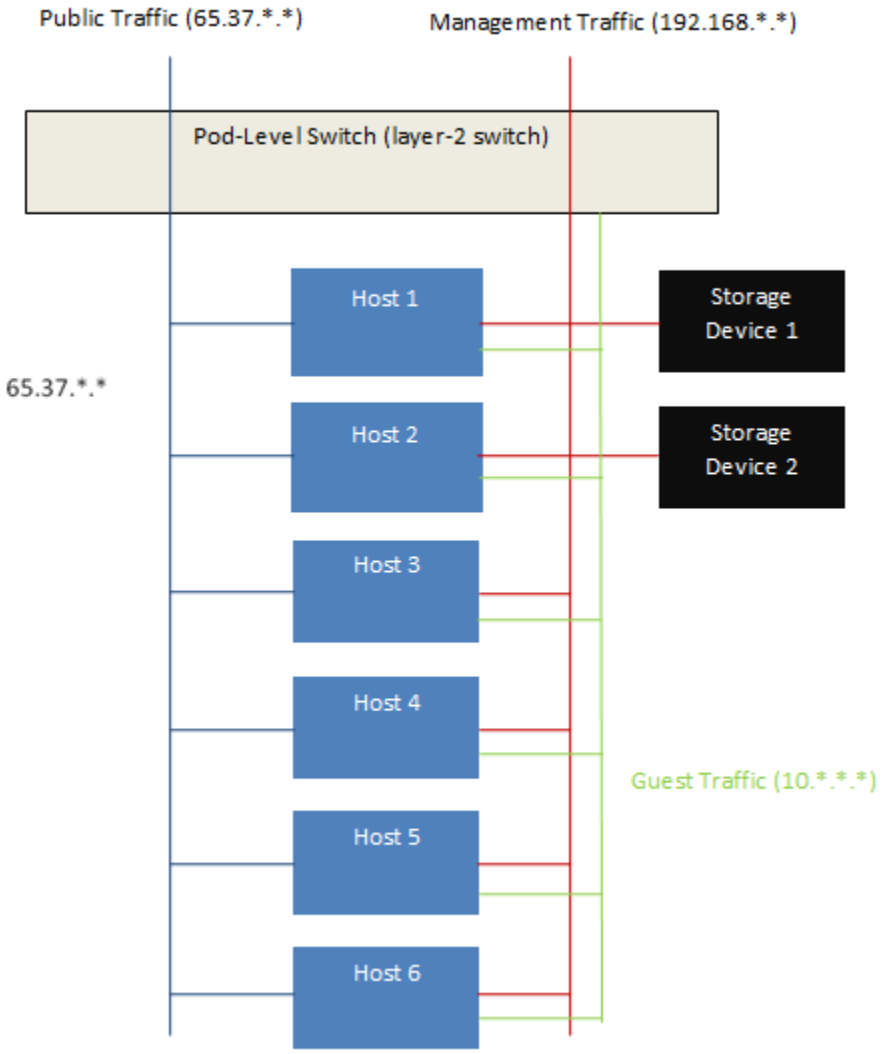
Typically, the Management Server automatically creates a virtual router for each network. A virtual router is a special virtual machine that runs on the hosts. Each virtual router in an isolated network has three network interfaces. If multiple public VLAN is used, the router will have multiple public interfaces. Its eth0 interface serves as the gateway for the guest traffic and has the IP address of 10.1.1.1. Its eth1 interface is used by the system to configure the virtual router. Its eth2 interface is assigned a public IP address for public traffic. If multiple public VLAN is used, the router will have multiple public interfaces.

The virtual router provides DHCP and will automatically assign an IP address for each guest VM within the IP range assigned for the network. The user can manually reconfigure guest VMs to assume different IP addresses.

Source NAT is automatically configured in the virtual router to forward outbound traffic for all guest VMs

5.12.2 Networking in a Pod

The figure below illustrates network setup within a single pod. The hosts are connected to a pod-level switch. At a minimum, the hosts should have one physical uplink to each switch. Bonded NICs are supported as well. The pod-level switch is a pair of redundant gigabit switches with 10 G uplinks.



Network Setup within a Single Pod – Logical View

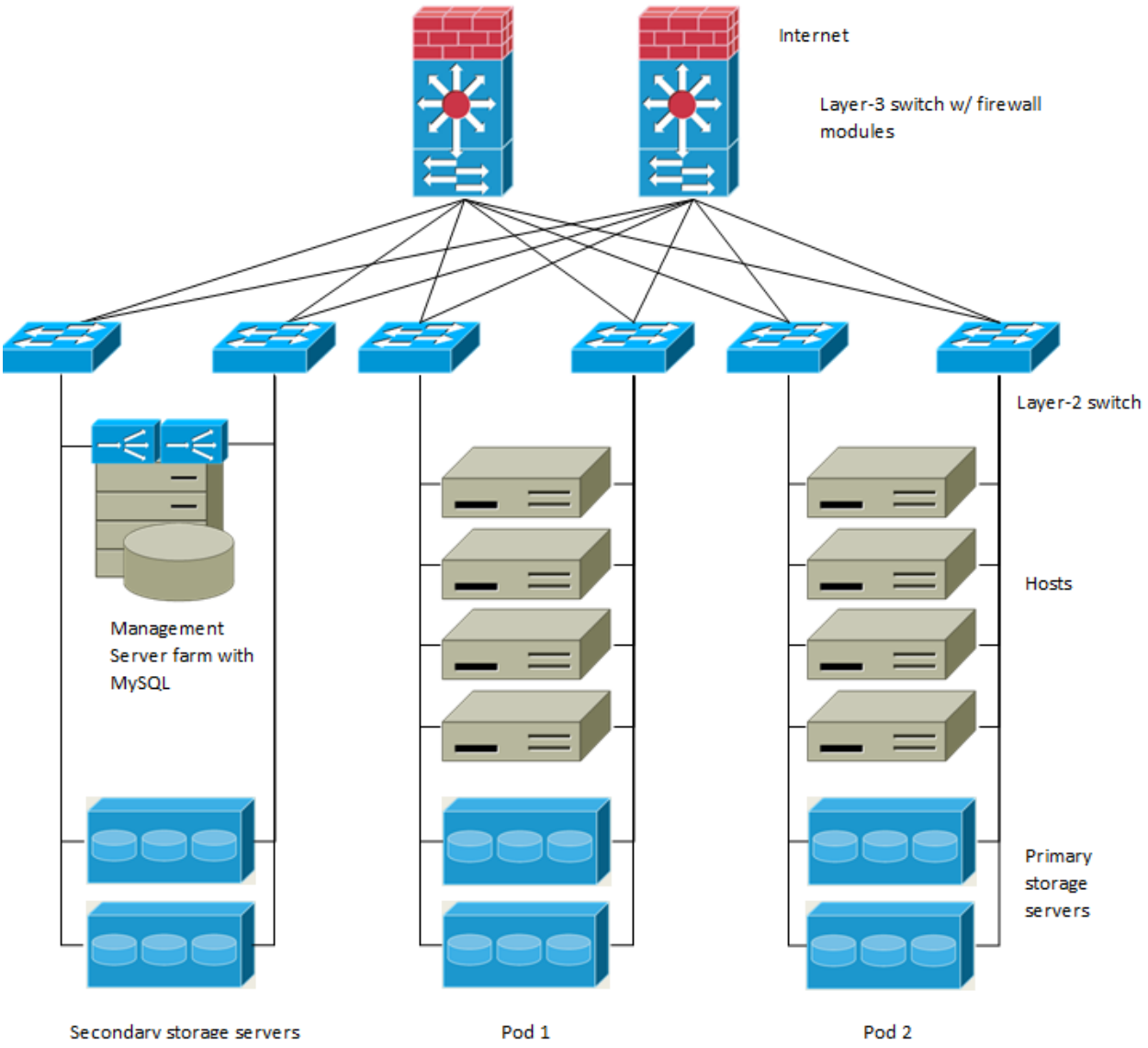
Servers are connected as follows:

- Storage devices are connected to only the network that carries management traffic.
- Hosts are connected to networks for both management traffic and public traffic.
- Hosts are also connected to one or more networks carrying guest traffic.

We recommend the use of multiple physical Ethernet cards to implement each network interface as well as redundant switch fabric in order to maximize throughput and improve reliability.

5.12.3 Networking in a Zone

The following figure illustrates the network setup within a single zone.



A firewall for management traffic operates in the NAT mode. The network typically is assigned IP addresses in the 192.168.0.0/16 Class B private address space. Each pod is assigned IP addresses in the 192.168.*.0/24 Class C private address space.

Each zone has its own set of public IP addresses. Public IP addresses from different zones do not overlap.

5.12.4 Basic Zone Physical Network Configuration

In a basic network, configuring the physical network is fairly straightforward. You only need to configure one guest network to carry traffic that is generated by guest VMs. When you first add a zone to CloudStack, you set up the guest network through the Add Zone screens.

5.12.5 Advanced Zone Physical Network Configuration

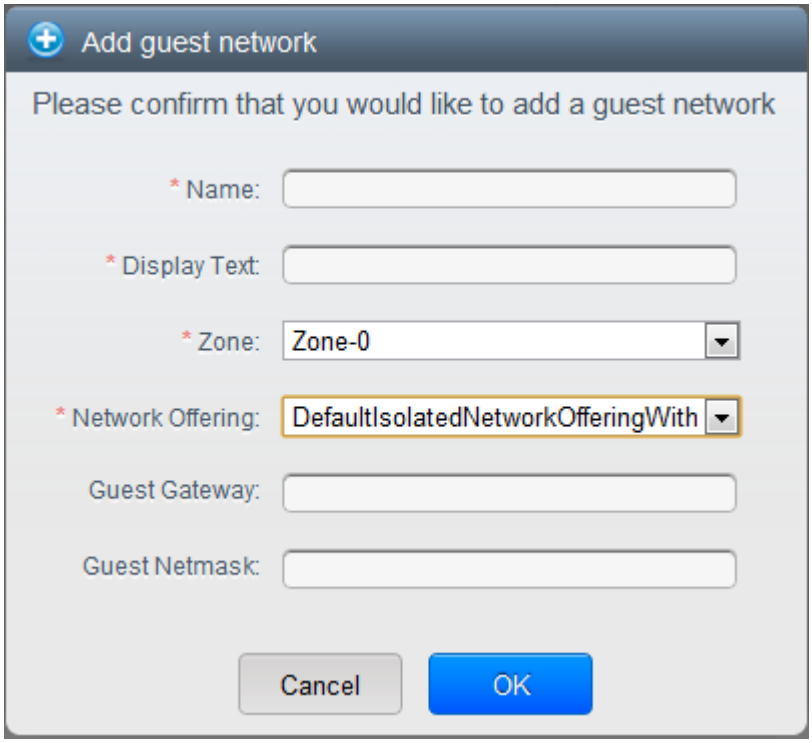
Within a zone that uses advanced networking, you need to tell the Management Server how the physical network is set up to carry different kinds of traffic in isolation.

Configure Guest Traffic in an Advanced Zone

These steps assume you have already logged in to the CloudStack UI. To configure the base guest network:

1. In the left navigation, choose Infrastructure. On Zones, click View More, then click the zone to which you want to add a network.
2. Click the Network tab.
3. Click Add guest network.

   The Add guest network window is displayed:



4. Provide the following information:
   - **Name**: The name of the network. This will be user-visible
   - **Display Text**: The description of the network. This will be user-visible
   - **Zone**: The zone in which you are configuring the guest network.
   - **Network offering**: If the administrator has configured multiple network offerings, select the one you want to use for this network
   - **Guest Gateway**: The gateway that the guests should use
   - **Guest Netmask**: The netmask in use on the subnet the guests will use
5. Click OK.

Configure Public Traffic in an Advanced Zone

In a zone that uses advanced networking, you need to configure at least one range of IP addresses for Internet traffic.

Configuring a Shared Guest Network

1. Log in to the CloudStack UI as administrator.
2. In the left navigation, choose Infrastructure.
3. On Zones, click View More.
4. Click the zone to which you want to add a guest network.
5. Click the Physical Network tab.
6. Click the physical network you want to work with.
7. On the Guest node of the diagram, click Configure.
8. Click the Network tab.
9. Click Add guest network.

   The Add guest network window is displayed.
10. Specify the following:
    - **Name**: The name of the network. This will be visible to the user.
    - **Description**: The short description of the network that can be displayed to users.
    - **VLAN ID**: The unique ID of the VLAN.
    - **Isolated VLAN ID**: The unique ID of the Secondary Isolated VLAN.
    - **Scope**: The available scopes are Domain, Account, Project, and All.
      - **Domain**: Selecting Domain limits the scope of this guest network to the domain you specify. The network will not be available for other domains. If you select Subdomain Access, the guest network is available to all the sub domains within the selected domain.
      - **Account**: The account for which the guest network is being created for. You must specify the domain the account belongs to.
      - **Project**: The project for which the guest network is being created for. You must specify the domain the project belongs to.
      - **All**: The guest network is available for all the domains, account, projects within the selected zone.
    - **Network Offering**: If the administrator has configured multiple network offerings, select the one you want to use for this network.
    - **Gateway**: The gateway that the guests should use.
    - **Netmask**: The netmask in use on the subnet the guests will use.
    - **IP Range**: A range of IP addresses that are accessible from the Internet and are assigned to the guest VMs.

      If one NIC is used, these IPs should be in the same CIDR in the case of IPv6.
    - **IPv6 CIDR**: The network prefix that defines the guest network subnet. This is the CIDR that describes the IPv6 addresses in use in the guest networks in this zone. To allot IP addresses from within a particular address block, enter a CIDR.
    - **Network Domain**: A custom DNS suffix at the level of a network. If you want to assign a special domain name to the guest VM network, specify a DNS suffix.
11. Click OK to confirm.

5.12.6 Using Multiple Guest Networks

In zones that use advanced networking, additional networks for guest traffic may be added at any time after the initial installation. You can also customize the domain name associated with the network by specifying a DNS suffix for each network.

A VM's networks are defined at VM creation time. A VM cannot add or remove networks after it has been created, although the user can go into the guest and remove the IP address from the NIC on a particular network.

Each VM has just one default network. The virtual router's DHCP reply will set the guest's default gateway as that for the default network. Multiple non-default networks may be added to a guest in addition to the single, required default network. The administrator can control which networks are available as the default network.

Additional networks can either be available to all accounts or be assigned to a specific account. Networks that are available to all accounts are zone-wide. Any user with access to the zone can create a VM with access to that network. These zone-wide networks provide little or no isolation between guests.Networks that are assigned to a specific account provide strong isolation.

Adding an Additional Guest Network

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click Add guest network. Provide the following information:
   - **Name**: The name of the network. This will be user-visible.
   - **Display Text**: The description of the network. This will be user-visible.
   - **Zone**. The name of the zone this network applies to. Each zone is a broadcast domain, and therefore each zone has a different IP range for the guest network. The administrator must configure the IP range for each zone.
   - **Network offering**: If the administrator has configured multiple network offerings, select the one you want to use for this network.
   - **Guest Gateway**: The gateway that the guests should use.
   - **Guest Netmask**: The netmask in use on the subnet the guests will use.
4. Click Create.

Reconfiguring Networks in VMs

CloudStack provides you the ability to move VMs between networks and reconfigure a VM's network. You can remove a VM from a network and add to a new network. You can also change the default network of a virtual machine. With this functionality, hybrid or traditional server loads can be accommodated with ease.

This feature is supported on XenServer, VMware, and KVM hypervisors.

Prerequisites

Ensure that vm-tools are running on guest VMs for adding or removing networks to work on VMware hypervisor.

Adding a Network

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, click Instances.
3. Choose the VM that you want to work with.
4. Click the NICs tab.
5. Click Add network to VM.

   The Add network to VM dialog is displayed.
6. In the drop-down list, select the network that you would like to add this VM to.

   A new NIC is added for this network. You can view the following details in the NICs page:

   - ID
   - Network Name
   - Type
   - IP Address
   - Gateway
   - Netmask
   - Is default
   - CIDR (for IPv6)

Removing a Network

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, click Instances.
3. Choose the VM that you want to work with.
4. Click the NICs tab.
5. Locate the NIC you want to remove.
6. Click Remove NIC button.
7. Click Yes to confirm.

Selecting the Default Network

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, click Instances.
3. Choose the VM that you want to work with.
4. Click the NICs tab.
5. Locate the NIC you want to work with.
6. Click the Set default NIC button. .

7. Click Yes to confirm.

Changing the Network Offering on a Guest Network

A user or administrator can change the network offering that is associated with an existing guest network.

1. Log in to the CloudStack UI as an administrator or end user.
2. If you are changing from a network offering that uses the CloudStack virtual router to one that uses external devices as network service providers, you must first stop all the VMs on the network.
3. In the left navigation, choose Network.
4. Click the name of the network you want to modify.
5. In the Details tab, click Edit.
6. In Network Offering, choose the new network offering, then click Apply.

   A prompt is displayed asking whether you want to keep the existing CIDR. This is to let you know that if you change the network offering, the CIDR will be affected.

   If you upgrade between virtual router as a provider and an external network device as provider, acknowledge the change of CIDR to continue, so choose Yes.
7. Wait for the update to complete. Don't try to restart VMs until the network change is complete.
8. If you stopped any VMs, restart them.

5.12.7 IP Reservation in Isolated Guest Networks

In isolated guest networks, a part of the guest IP address space can be reserved for non-CloudStack VMs or physical servers. To do so, you configure a range of Reserved IP addresses by specifying the CIDR when a guest network is in Implemented state. If your customers wish to have non-CloudStack controlled VMs or physical servers on the same network, they can share a part of the IP address space that is primarily provided to the guest network.

In an Advanced zone, an IP address range or a CIDR is assigned to a network when the network is defined. The CloudStack virtual router acts as the DHCP server and uses CIDR for assigning IP addresses to the guest VMs. If you decide to reserve CIDR for non-CloudStack purposes, you can specify a part of the IP address range or the CIDR that should only be allocated by the DHCP service of the virtual router to the guest VMs created in CloudStack. The remaining IPs in that network are called Reserved IP Range. When IP reservation is configured, the administrator can add additional VMs or physical servers that are not part of CloudStack to the same network and assign them the Reserved IP addresses. CloudStack guest VMs cannot acquire IPs from the Reserved IP Range.

IP Reservation Considerations

Consider the following before you reserve an IP range for non-CloudStack machines:

- IP Reservation is supported only in Isolated networks.
- IP Reservation can be applied only when the network is in Implemented state.
- No IP Reservation is done by default.
- Guest VM CIDR you specify must be a subset of the network CIDR.

- Specify a valid Guest VM CIDR. IP Reservation is applied only if no active IPs exist outside the Guest VM CIDR.

  You cannot apply IP Reservation if any VM is alloted with an IP address that is outside the Guest VM CIDR.

- To reset an existing IP Reservation, apply IP reservation by specifying the value of network CIDR in the CIDR field.

  For example, the following table describes three scenarios of guest network creation:

Case	CIDR	Network CIDR	Reserved IP Range for Non-CloudStack VMs	Description
1	10.1.1.0/24	None	None	No IP Reservation.
2	10.1.1.0/26	10.1.1.0/24	10.1.1.64 to 10.1.1.254	IP Reservation configured by the UpdateNetwork API with guestvmcidr=10.1.1.0/26 or enter 10.1.1.0/26 in the CIDR field in the UI.
3	10.1.1.0/24	None	None	Removing IP Reservation by the UpdateNetwork API with guestvmcidr=10.1.1.0/24 or enter 10.1.1.0/24 in the CIDR field in the UI.

Limitations

- The IP Reservation is not supported if active IPs that are found outside the Guest VM CIDR.
- Upgrading network offering which causes a change in CIDR (such as upgrading an offering with no external devices to one with external devices) IP Reservation becomes void if any. Reconfigure IP Reservation in the new re-implemeted network.

Best Practices

Apply IP Reservation to the guest network as soon as the network state changes to Implemented. If you apply reservation soon after the first guest VM is deployed, lesser conflicts occurs while applying reservation.

Reserving an IP Range

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network you want to modify.
4. In the Details tab, click Edit.

   The CIDR field changes to editable one.
5. In CIDR, specify the Guest VM CIDR.
6. Click Apply.

   Wait for the update to complete. The Network CIDR and the Reserved IP Range are displayed on the Details page.

5.12.8 Reserving Public IP Addresses and VLANs for Accounts

CloudStack provides you the ability to reserve a set of public IP addresses and VLANs exclusively for an account. During zone creation, you can continue defining a set of VLANs and multiple public IP ranges. This feature extends the functionality to enable you to dedicate a fixed set of VLANs and guest IP addresses for a tenant.

Note that if an account has consumed all the VLANs and IPs dedicated to it, the account can acquire two more resources from the system. CloudStack provides the root admin with two configuration parameter to modify this default behavior: use.system.public.ips and use.system.guest.vlans. These global parameters enable the root admin to disallow an account from acquiring public IPs and guest VLANs from the system, if the account has dedicated resources and these dedicated resources have all been consumed. Both these configurations are configurable at the account level.

This feature provides you the following capabilities:

- Reserve a VLAN range and public IP address range from an Advanced zone and assign it to an account
- Disassociate a VLAN and public IP address range from an account
- View the number of public IP addresses allocated to an account
- Check whether the required range is available and is conforms to account limits.

  The maximum IPs per account limit cannot be superseded.

Dedicating IP Address Ranges to an Account

1. Log in to the CloudStack UI as administrator.
2. In the left navigation bar, click Infrastructure.
3. In Zones, click View All.
4. Choose the zone you want to work with.
5. Click the Physical Network tab.
6. In the Public node of the diagram, click Configure.
7. Click the IP Ranges tab.

   You can either assign an existing IP range to an account, or create a new IP range and assign to an account.
8. To assign an existing IP range to an account, perform the following:
   (a) Locate the IP range you want to work with.
   (b) Click Add Account button.

   The Add Account dialog is displayed.
   (c) Specify the following:
   - **Account**: The account to which you want to assign the IP address range.
   - **Domain**: The domain associated with the account.

   To create a new IP range and assign an account, perform the following:
   i. Specify the following:
   - **Gateway**
   - **Netmask**
   - **VLAN**

- **Start IP**
- **End IP**
- **Account**: Perform the following:

  A. Click Account.

     The Add Account page is displayed.

  B. Specify the following:

     – **Account**: The account to which you want to assign an IP address range.

     – **Domain**: The domain associated with the account.

  C. Click OK.

ii. Click Add.

Dedicating VLAN Ranges to an Account

1. After the CloudStack Management Server is installed, log in to the CloudStack UI as administrator.
2. In the left navigation bar, click Infrastructure.
3. In Zones, click View All.
4. Choose the zone you want to work with.
5. Click the Physical Network tab.
6. In the Guest node of the diagram, click Configure.
7. Select the Dedicated VLAN Ranges tab.
8. Click Dedicate VLAN Range.

   The Dedicate VLAN Range dialog is displayed.
9. Specify the following:
   - **VLAN Range**: The VLAN range that you want to assign to an account.
   - **Account**: The account to which you want to assign the selected VLAN range.
   - **Domain**: The domain associated with the account.

5.12.9 Configuring Multiple IP Addresses on a Single NIC

CloudStack provides you the ability to associate multiple private IP addresses per guest VM NIC. In addition to the primary IP, you can assign additional IPs to the guest VM NIC. This feature is supported on all the network configurations: Basic, Advanced, and VPC. Security Groups, Static NAT and Port forwarding services are supported on these additional IPs.

As always, you can specify an IP from the guest subnet; if not specified, an IP is automatically picked up from the guest VM subnet. You can view the IPs associated with for each guest VM NICs on the UI. You can apply NAT on these additional guest IPs by using network configuration option in the CloudStack UI. You must specify the NIC to which the IP should be associated.

This feature is supported on XenServer, KVM, and VMware hypervisors. Note that Basic zone security groups are not supported on VMware.

Use Cases

Some of the use cases are described below:

- Network devices, such as firewalls and load balancers, generally work best when they have access to multiple IP addresses on the network interface.
- Moving private IP addresses between interfaces or instances. Applications that are bound to specific IP addresses can be moved between instances.
- Hosting multiple SSL Websites on a single instance. You can install multiple SSL certificates on a single instance, each associated with a distinct IP address.

Guidelines

To prevent IP conflict, configure different subnets when multiple networks are connected to the same VM.

Assigning Additional IPs to a VM

1. Log in to the CloudStack UI.
2. In the left navigation bar, click Instances.
3. Click the name of the instance you want to work with.
4. In the Details tab, click NICs.
5. Click View Secondary IPs.
6. Click Acquire New Secondary IP, and click Yes in the confirmation dialog.

   You need to configure the IP on the guest VM NIC manually. CloudStack will not automatically configure the acquired IP address on the VM. Ensure that the IP address configuration persist on VM reboot.

   Within a few moments, the new IP address should appear with the state Allocated. You can now use the IP address in Port Forwarding or StaticNAT rules.

Port Forwarding and StaticNAT Services Changes

Because multiple IPs can be associated per NIC, you are allowed to select a desired IP for the Port Forwarding and StaticNAT services. The default is the primary IP. To enable this functionality, an extra optional parameter 'vmguestip' is added to the Port forwarding and StaticNAT APIs (enableStaticNat, createIpForwardingRule) to indicate on what IP address NAT need to be configured. If vmguestip is passed, NAT is configured on the specified private IP of the VM. if not passed, NAT is configured on the primary IP of the VM.

5.12.10 About Multiple IP Ranges

Note: The feature can only be implemented on IPv4 addresses.

CloudStack provides you with the flexibility to add guest IP ranges from different subnets in Basic zones and security groups-enabled Advanced zones. For security groups-enabled Advanced zones, it implies multiple subnets can be added to the same VLAN. With the addition of this feature, you will be able to add IP address ranges from the same subnet or from a different one when IP address are exhausted. This would in turn allows you to employ higher

number of subnets and thus reduce the address management overhead. To support this feature, the capability of `createVlanIpRange` API is extended to add IP ranges also from a different subnet.

Ensure that you manually configure the gateway of the new subnet before adding the IP range. Note that CloudStack supports only one gateway for a subnet; overlapping subnets are not currently supported.

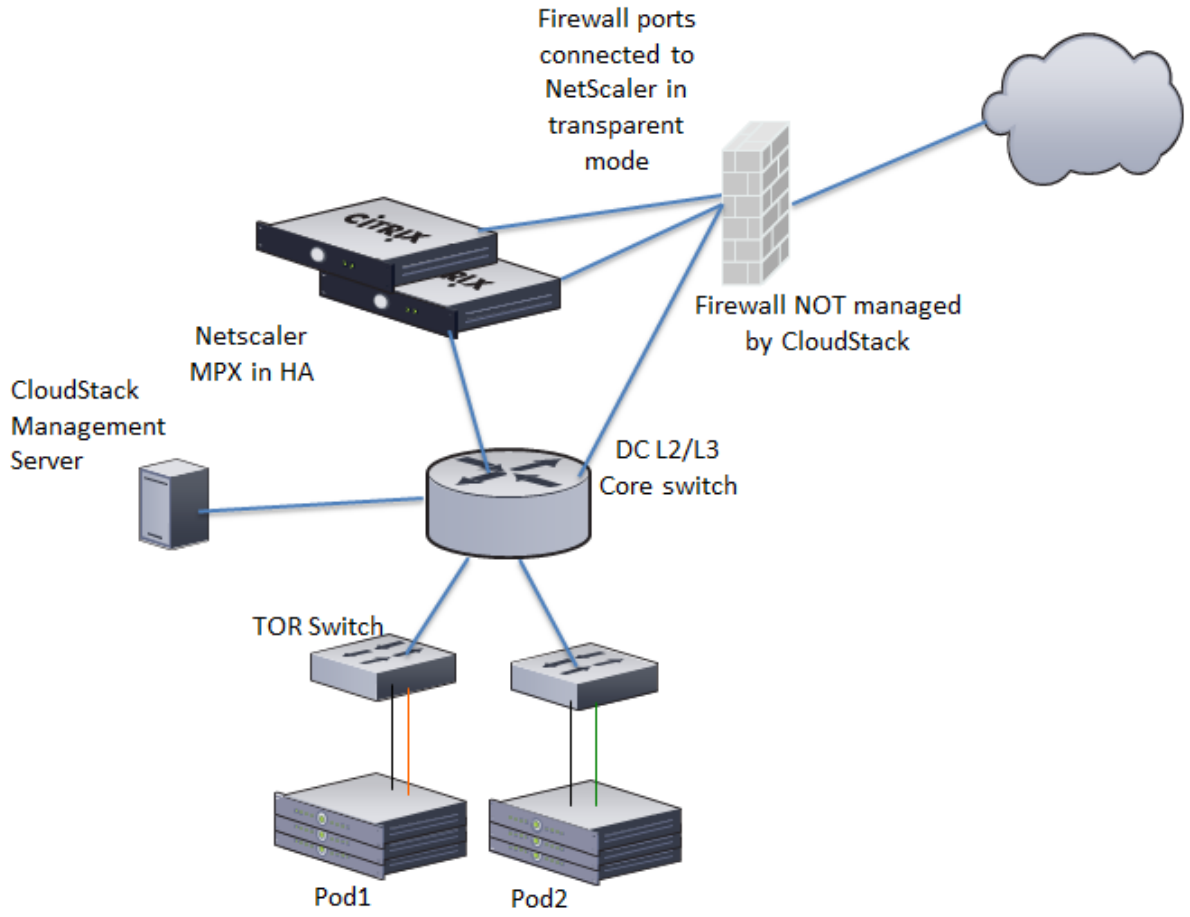
Use the `deleteVlanRange` API to delete IP ranges. This operation fails if an IP from the remove range is in use. If the remove range contains the IP address on which the DHCP server is running, CloudStack acquires a new IP from the same subnet. If no IP is available in the subnet, the remove operation fails.

This feature is supported on KVM, xenServer, and VMware hypervisors.

5.12.11 About Elastic IPs

Elastic IP (EIP) addresses are the IP addresses that are associated with an account, and act as static IP addresses. The account owner has the complete control over the Elastic IP addresses that belong to the account. As an account owner, you can allocate an Elastic IP to a VM of your choice from the EIP pool of your account. Later if required you can reassign the IP address to a different VM. This feature is extremely helpful during VM failure. Instead of replacing the VM which is down, the IP address can be reassigned to a new VM in your account.

Similar to the public IP address, Elastic IP addresses are mapped to their associated private IP addresses by using StaticNAT. The EIP service is equipped with StaticNAT (1:1) service in an EIP-enabled basic zone. The default network offering, DefaultSharedNetscalerEIPandELBNetworkOffering, provides your network with EIP and ELB network services if a NetScaler device is deployed in your zone. Consider the following illustration for more details.



In the illustration, a NetScaler appliance is the default entry or exit point for the CloudStack instances, and firewall is the default entry or exit point for the rest of the data center. Netscaler provides LB services and staticNAT service to the guest networks. The guest traffic in the pods and the Management Server are on different subnets / VLANs. The policy-based routing in the data center core switch sends the public traffic through the NetScaler, whereas the rest of the data center goes through the firewall.

The EIP work flow is as follows:

- When a user VM is deployed, a public IP is automatically acquired from the pool of public IPs configured in the zone. This IP is owned by the VM's account.
- Each VM will have its own private IP. When the user VM starts, Static NAT is provisioned on the NetScaler device by using the Inbound Network Address Translation (INAT) and Reverse NAT (RNAT) rules between the public IP and the private IP.

  Note: Inbound NAT (INAT) is a type of NAT supported by NetScaler, in which the destination IP address is replaced in the packets from the public network, such as the Internet, with the private IP address of a VM in the private network. Reverse NAT (RNAT) is a type of NAT supported by NetScaler, in which the source IP address is replaced in the packets generated by a VM in the private network with the public IP address.

- This default public IP will be released in two cases:
  - When the VM is stopped. When the VM starts, it again receives a new public IP, not necessarily the same one allocated initially, from the pool of Public IPs.
  - The user acquires a public IP (Elastic IP). This public IP is associated with the account, but will not be mapped to any private IP. However, the user can enable Static NAT to associate this IP to the private IP of a VM in the account. The Static NAT rule for the public IP can be disabled at any time. When Static NAT is disabled, a new public IP is allocated from the pool, which is not necessarily be the same one allocated initially.

For the deployments where public IPs are limited resources, you have the flexibility to choose not to allocate a public IP by default. You can use the Associate Public IP option to turn on or off the automatic public IP assignment in the EIP-enabled Basic zones. If you turn off the automatic public IP assignment while creating a network offering, only a private IP is assigned to a VM when the VM is deployed with that network offering. Later, the user can acquire an IP for the VM and enable static NAT.

For more information on the Associate Public IP option, see "Creating a New Network Offering".

Note: The Associate Public IP feature is designed only for use with user VMs. The System VMs continue to get both public IP and private by default, irrespective of the network offering configuration.

New deployments which use the default shared network offering with EIP and ELB services to create a shared network in the Basic zone will continue allocating public IPs to each user VM.

5.12.12 Portable IPs

About Portable IP

Portable IPs in CloudStack are region-level pool of IPs, which are elastic in nature, that can be transferred across geographically separated zones. As an administrator, you can provision a pool of portable public IPs at region level and are available for user consumption. The users can acquire portable IPs if admin has provisioned portable IPs at the region level they are part of. These IPs can be use for any service within an advanced zone. You can also use portable IPs for EIP services in basic zones.

The salient features of Portable IP are as follows:

- IP is statically allocated
- IP need not be associated with a network
- IP association is transferable across networks
- IP is transferable across both Basic and Advanced zones
- IP is transferable across VPC, non-VPC isolated and shared networks
- Portable IP transfer is available only for static NAT.

Guidelines

Before transferring to another network, ensure that no network rules (Firewall, Static NAT, Port Forwarding, and so on) exist on that portable IP.

Configuring Portable IPs

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, click Regions.
3. Choose the Regions that you want to work with.
4. Click View Portable IP.
5. Click Portable IP Range.

   The Add Portable IP Range window is displayed.
6. Specify the following:
   - **Start IP/ End IP**: A range of IP addresses that are accessible from the Internet and will be allocated to guest VMs. Enter the first and last IP addresses that define a range that CloudStack can assign to guest VMs.
   - **Gateway**: The gateway in use for the Portable IP addresses you are configuring.
   - **Netmask**: The netmask associated with the Portable IP range.
   - **VLAN**: The VLAN that will be used for public traffic.
7. Click OK.

Acquiring a Portable IP

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network where you want to work with.
4. Click View IP Addresses.
5. Click Acquire New IP.

   The Acquire New IP window is displayed.
6. Specify whether you want cross-zone IP or not.

7. Click Yes in the confirmation dialog.

   Within a few moments, the new IP address should appear with the state Allocated. You can now use the IP address in port forwarding or static NAT rules.

Transferring Portable IP

An IP can be transferred from one network to another only if Static NAT is enabled. However, when a portable IP is associated with a network, you can use it for any service in the network.

To transfer a portable IP across the networks, execute the following API:

```
http://localhost:8096/client/api?command=enableStaticNat&response=json&
↪ipaddressid=a4bc37b2-4b4e-461d-9a62-b66414618e36&virtualmachineid=a242c476-ef37-
↪441e-9c7b-b303e2a9cb4f&networkid=6e7cd8d1-d1ba-4c35-bdaf-333354cbd49810
```

Replace the UUID with appropriate UUID. For example, if you want to transfer a portable IP to network X and VM Y in a network, execute the following:

```
http://localhost:8096/client/api?command=enableStaticNat&response=json&
↪ipaddressid=a4bc37b2-4b4e-461d-9a62-b66414618e36&virtualmachineid=Y&networkid=X
```

5.12.13 Multiple Subnets in Shared Network

CloudStack provides you with the flexibility to add guest IP ranges from different subnets in Basic zones and security groups-enabled Advanced zones. For security groups-enabled Advanced zones, it implies multiple subnets can be added to the same VLAN. With the addition of this feature, you will be able to add IP address ranges from the same subnet or from a different one when IP address are exhausted. This would in turn allows you to employ higher number of subnets and thus reduce the address management overhead. You can delete the IP ranges you have added.

Prerequisites and Guidelines

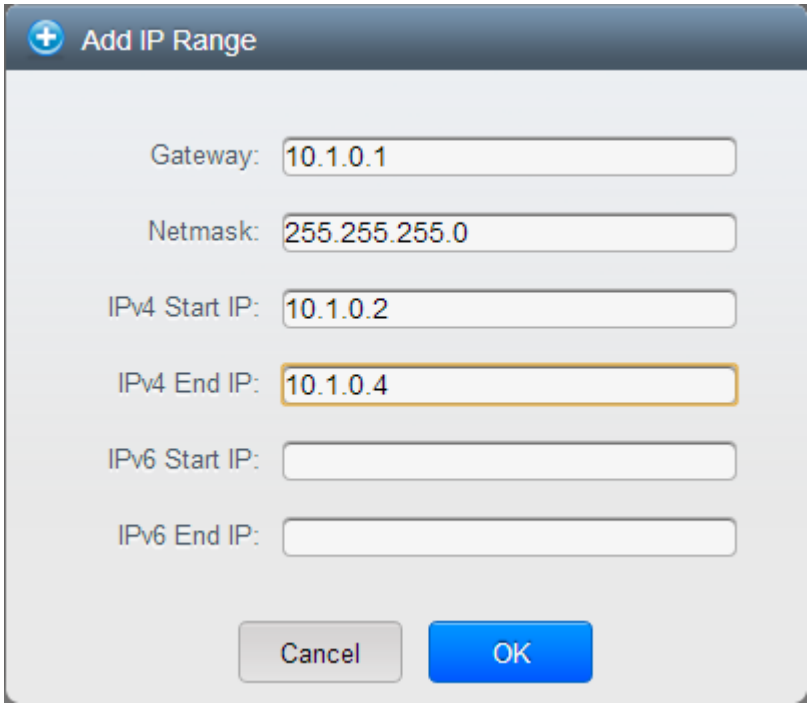
- This feature can only be implemented:
  - on IPv4 addresses
  - if virtual router is the DHCP provider
  - on KVM, xenServer, and VMware hypervisors
- Manually configure the gateway of the new subnet before adding the IP range.
- CloudStack supports only one gateway for a subnet; overlapping subnets are not currently supported

Adding Multiple Subnets to a Shared Network

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Infrastructure.
3. On Zones, click View More, then click the zone to which you want to work with..
4. Click Physical Network.
5. In the Guest node of the diagram, click Configure.
6. Click Networks.

7. Select the networks you want to work with.
8. Click View IP Ranges.
9. Click Add IP Range.

   The Add IP Range dialog is displayed, as follows:



10. Specify the following:

    All the fields are mandatory.

    - **Gateway**: The gateway for the tier you create. Ensure that the gateway is within the Super CIDR range that you specified while creating the VPC, and is not overlapped with the CIDR of any existing tier within the VPC.
    - **Netmask**: The netmask for the tier you create.

      For example, if the VPC CIDR is 10.0.0.0/16 and the network tier CIDR is 10.0.1.0/24, the gateway of the tier is 10.0.1.1, and the netmask of the tier is 255.255.255.0.
    - **Start IP/ End IP**: A range of IP addresses that are accessible from the Internet and will be allocated to guest VMs. Enter the first and last IP addresses that define a range that CloudStack can assign to guest VMs .
11. Click OK.

5.12.14 Isolation in Advanced Zone Using Private VLAN

Isolation of guest traffic in shared networks can be achieved by using Private VLANs (PVLAN). PVLANs provide Layer 2 isolation between ports within the same VLAN. In a PVLAN-enabled shared network, a user VM cannot reach other user VM though they can reach the DHCP server and gateway, this would in turn allow users to control traffic within a network and help them deploy multiple applications without communication between application as well as prevent communication with other users' VMs.

- Isolate VMs in a shared networks by using Private VLANs.

- Supported on KVM, XenServer, and VMware hypervisors
- PVLAN-enabled shared network can be a part of multiple networks of a guest VM.

About Private VLAN

In an Ethernet switch, a VLAN is a broadcast domain where hosts can establish direct communication with each another at Layer 2. Private VLAN is designed as an extension of VLAN standard to add further segmentation of the logical broadcast domain. A regular VLAN is a single broadcast domain, whereas a private VLAN partitions a larger VLAN broadcast domain into smaller sub-domains. A sub-domain is represented by a pair of VLANs: a Primary VLAN and a Secondary VLAN. The original VLAN that is being divided into smaller groups is called Primary, which implies that all VLAN pairs in a private VLAN share the same Primary VLAN. All the secondary VLANs exist only inside the Primary. Each Secondary VLAN has a specific VLAN ID associated to it, which differentiates one sub-domain from another.

Three types of ports exist in a private VLAN domain, which essentially determine the behaviour of the participating hosts. Each ports will have its own unique set of rules, which regulate a connected host's ability to communicate with other connected host within the same private VLAN domain. Configure each host that is part of a PVLAN pair can be by using one of these three port designation:

- **Promiscuous**: A promiscuous port can communicate with all the interfaces, including the community and isolated host ports that belong to the secondary VLANs. In Promiscuous mode, hosts are connected to promiscuous ports and are able to communicate directly with resources on both primary and secondary VLAN. Routers, DHCP servers, and other trusted devices are typically attached to promiscuous ports.
- **Isolated VLANs**: The ports within an isolated VLAN cannot communicate with each other at the layer-2 level. The hosts that are connected to Isolated ports can directly communicate only with the Promiscuous resources. If your customer device needs to have access only to a gateway router, attach it to an isolated port.
- **Community VLANs**: The ports within a community VLAN can communicate with each other and with the promiscuous ports, but they cannot communicate with the ports in other communities at the layer-2 level. In a Community mode, direct communication is permitted only with the hosts in the same community and those that are connected to the Primary PVLAN in promiscuous mode. If your customer has two devices that need to be isolated from other customers' devices, but to be able to communicate among themselves, deploy them in community ports.

For further reading:

- Understanding Private VLANs
- Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment
- Private VLAN (PVLAN) on vNetwork Distributed Switch - Concept Overview (1010691)

Prerequisites

- Use a PVLAN supported switch.

  See Private VLAN Catalyst Switch Support Matrix for more information.
- All the layer 2 switches, which are PVLAN-aware, are connected to each other, and one of them is connected to a router. All the ports connected to the host would be configured in trunk mode. Open Management VLAN, Primary VLAN (public) and Secondary Isolated VLAN ports. Configure the switch port connected to the router in PVLAN promiscuous trunk mode, which would translate an isolated VLAN to primary VLAN for the PVLAN-unaware router.

  Note that only Cisco Catalyst 4500 has the PVLAN promiscuous trunk mode to connect both normal VLAN and PVLAN to a PVLAN-unaware switch. For the other Catalyst PVLAN support switch, connect the switch to upper switch by using cables, one each for a PVLAN pair.

- Configure private VLAN on your physical switches out-of-band.
- Before you use PVLAN on XenServer and KVM, enable Open vSwitch (OVS).

Note: OVS on XenServer and KVM does not support PVLAN natively. Therefore, CloudStack managed to simulate PVLAN on OVS for XenServer and KVM by modifying the flow table.

Creating a PVLAN-Enabled Guest Network

1. Log in to the CloudStack UI as administrator.
2. In the left navigation, choose Infrastructure.
3. On Zones, click View More.
4. Click the zone to which you want to add a guest network.
5. Click the Physical Network tab.
6. Click the physical network you want to work with.
7. On the Guest node of the diagram, click Configure.
8. Click the Network tab.
9. Click Add guest network.

   The Add guest network window is displayed.
10. Specify the following:
    - **Name**: The name of the network. This will be visible to the user.
    - **Description**: The short description of the network that can be displayed to users.
    - **VLAN ID**: The unique ID of the VLAN.
    - **Secondary Isolated VLAN ID**: The unique ID of the Secondary Isolated VLAN.

      For the description on Secondary Isolated VLAN, see *About Private VLAN"*.
    - **Scope**: The available scopes are Domain, Account, Project, and All.
      - **Domain**: Selecting Domain limits the scope of this guest network to the domain you specify. The network will not be available for other domains. If you select Subdomain Access, the guest network is available to all the sub domains within the selected domain.
      - **Account**: The account for which the guest network is being created for. You must specify the domain the account belongs to.
      - **Project**: The project for which the guest network is being created for. You must specify the domain the project belongs to.
      - **All**: The guest network is available for all the domains, account, projects within the selected zone.
    - **Network Offering**: If the administrator has configured multiple network offerings, select the one you want to use for this network.
    - **Gateway**: The gateway that the guests should use.
    - **Netmask**: The netmask in use on the subnet the guests will use.
    - **IP Range**: A range of IP addresses that are accessible from the Internet and are assigned to the guest VMs.

- **Network Domain**: A custom DNS suffix at the level of a network. If you want to assign a special domain name to the guest VM network, specify a DNS suffix.

11. Click OK to confirm.

5.12.15 Security Groups

About Security Groups

Security groups provide a way to isolate traffic to VMs. A security group is a group of VMs that filter their incoming and outgoing traffic according to a set of rules, called ingress and egress rules. These rules filter network traffic according to the IP address that is attempting to communicate with the VM. Security groups are particularly useful in zones that use basic networking, because there is a single guest network for all guest VMs. In advanced zones, security groups are supported only on the KVM hypervisor.

Note: In a zone that uses advanced networking, you can instead define multiple guest networks to isolate traffic to VMs.

Each CloudStack account comes with a default security group that denies all inbound traffic and allows all outbound traffic. The default security group can be modified so that all new VMs inherit some other desired set of rules.

Any CloudStack user can set up any number of additional security groups. When a new VM is launched, it is assigned to the default security group unless another user-defined security group is specified. A VM can be a member of any number of security groups. Once a VM is assigned to a security group, it remains in that group for its entire lifetime; you can not move a running VM from one security group to another.

You can modify a security group by deleting or adding any number of ingress and egress rules. When you do, the new rules apply to all VMs in the group, whether running or stopped.

If no ingress rules are specified, then no traffic will be allowed in, except for responses to any traffic that has been allowed out through an egress rule.

Adding a Security Group

A user or administrator can define a new security group.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In Select view, choose Security Groups.
4. Click Add Security Group.
5. Provide a name and description.
6. Click OK.

   The new security group appears in the Security Groups Details tab.
7. To make the security group useful, continue to Adding Ingress and Egress Rules to a Security Group.

Security Groups in Advanced Zones (KVM Only)

CloudStack provides the ability to use security groups to provide isolation between guests on a single shared, zone-wide network in an advanced zone where KVM is the hypervisor. Using security groups in advanced zones rather than multiple VLANs allows a greater range of options for setting up guest isolation in a cloud.

Limitations

The following are not supported for this feature:

- Two IP ranges with the same VLAN and different gateway or netmask in security group-enabled shared network.
- Two IP ranges with the same VLAN and different gateway or netmask in account-specific shared networks.
- Multiple VLAN ranges in security group-enabled shared network.
- Multiple VLAN ranges in account-specific shared networks.

Security groups must be enabled in the zone in order for this feature to be used.

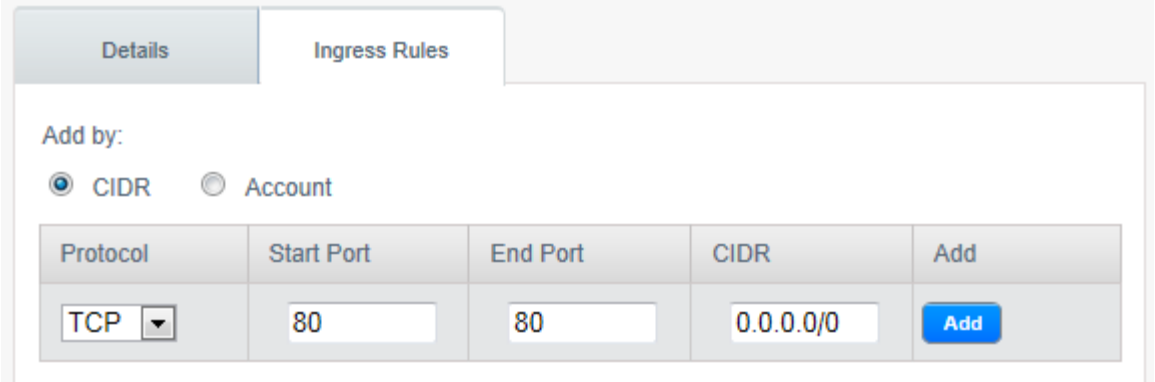
Enabling Security Groups

In order for security groups to function in a zone, the security groups feature must first be enabled for the zone. The administrator can do this when creating a new zone, by selecting a network offering that includes security groups. The procedure is described in Basic Zone Configuration in the Advanced Installation Guide. The administrator can not enable security groups for an existing zone, only when creating a new zone.

Adding Ingress and Egress Rules to a Security Group

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network
3. In Select view, choose Security Groups, then click the security group you want.
4. To add an ingress rule, click the Ingress Rules tab and fill out the following fields to specify what network traffic is allowed into VM instances in this security group. If no ingress rules are specified, then no traffic will be allowed in, except for responses to any traffic that has been allowed out through an egress rule.
   - **Add by CIDR/Account**. Indicate whether the source of the traffic will be defined by IP address (CIDR) or an existing security group in a CloudStack account (Account). Choose Account if you want to allow incoming traffic from all VMs in another security group
   - **Protocol**. The networking protocol that sources will use to send traffic to the security group. TCP and UDP are typically used for data exchange and end-user communications. ICMP is typically used to send error messages or network monitoring data.
   - **Start Port, End Port**. (TCP, UDP only) A range of listening ports that are the destination for the incoming traffic. If you are opening a single port, use the same number in both fields.
   - **ICMP Type, ICMP Code**. (ICMP only) The type of message and error code that will be accepted.
   - **CIDR**. (Add by CIDR only) To accept only traffic from IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. The CIDR is the base IP address of the incoming traffic. For example, 192.168.0.0/22. To allow all CIDRs, set to 0.0.0.0/0.
   - **Account, Security Group**. (Add by Account only) To accept only traffic from another security group, enter the CloudStack account and name of a security group that has already been defined in that account. To allow traffic between VMs within the security group you are editing now, enter the same name you used in step 7.

   The following example allows inbound HTTP access from anywhere:



5. To add an egress rule, click the Egress Rules tab and fill out the following fields to specify what type of traffic is allowed to be sent out of VM instances in this security group. If no egress rules are specified, then all traffic will be allowed out. Once egress rules are specified, the following types of traffic are allowed out: traffic specified in egress rules; queries to DNS and DHCP servers; and responses to any traffic that has been allowed in through an ingress rule
   - **Add by CIDR/Account**. Indicate whether the destination of the traffic will be defined by IP address (CIDR) or an existing security group in a CloudStack account (Account). Choose Account if you want to allow outgoing traffic to all VMs in another security group.
   - **Protocol**. The networking protocol that VMs will use to send outgoing traffic. TCP and UDP are typically used for data exchange and end-user communications. ICMP is typically used to send error messages or network monitoring data.
   - **Start Port, End Port**. (TCP, UDP only) A range of listening ports that are the destination for the outgoing traffic. If you are opening a single port, use the same number in both fields.
   - **ICMP Type, ICMP Code**. (ICMP only) The type of message and error code that will be sent
   - **CIDR**. (Add by CIDR only) To send traffic only to IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. The CIDR is the base IP address of the destination. For example, 192.168.0.0/22. To allow all CIDRs, set to 0.0.0.0/0.
   - **Account, Security Group**. (Add by Account only) To allow traffic to be sent to another security group, enter the CloudStack account and name of a security group that has already been defined in that account. To allow traffic between VMs within the security group you are editing now, enter its name.
6. Click Add.

5.12.16 External Firewalls and Load Balancers

CloudStack is capable of replacing its Virtual Router with an external Juniper SRX device and an optional external NetScaler or F5 load balancer for gateway and load balancing services. In this case, the VMs use the SRX as their gateway.

About Using a NetScaler Load Balancer

Citrix NetScaler is supported as an external network element for load balancing in zones that use isolated networking in advanced zones. Set up an external load balancer when you want to provide load balancing through means other than CloudStack's provided virtual router.

Note: In a Basic zone, load balancing service is supported only if Elastic IP or Elastic LB services are enabled.

When NetScaler load balancer is used to provide EIP or ELB services in a Basic zone, ensure that all guest VM traffic must enter and exit through the NetScaler device. When inbound traffic goes through the NetScaler device, traffic is routed by using the NAT protocol depending on the EIP/ELB configured on the public IP to the private IP. The traffic that is originated from the guest VMs usually goes through the layer 3 router. To ensure that outbound traffic goes through NetScaler device providing EIP/ELB, layer 3 router must have a policy-based routing. A policy-based route must be set up so that all traffic originated from the guest VM's are directed to NetScaler device. This is required to ensure that the outbound traffic from the guest VM's is routed to a public IP by using NAT.For more information on Elastic IP, see *"About Elastic IP"*.

The NetScaler can be set up in direct (outside the firewall) mode. It must be added before any load balancing rules are deployed on guest VMs in the zone.

The functional behavior of the NetScaler with CloudStack is the same as described in the CloudStack documentation for using an F5 external load balancer. The only exception is that the F5 supports routing domains, and NetScaler does not. NetScaler can not yet be used as a firewall.

To install and enable an external load balancer for CloudStack management, see External Guest Load Balancer Integration in the Installation Guide.

The Citrix NetScaler comes in three varieties. The following summarizes how these variants are treated in CloudStack.

MPX

- Physical appliance. Capable of deep packet inspection. Can act as application firewall and load balancer
- In advanced zones, load balancer functionality fully supported without limitation. In basic zones, static NAT, elastic IP (EIP), and elastic load balancing (ELB) are also provided.

VPX

- Virtual appliance. Can run as VM on XenServer, ESXi, and Hyper-V hypervisors. Same functionality as MPX
- Supported on ESXi and XenServer. Same functional support as for MPX. CloudStack will treat VPX and MPX as the same device type.

SDX

- Physical appliance. Can create multiple fully isolated VPX instances on a single appliance to support multi-tenant usage
- CloudStack will dynamically provision, configure, and manage the life cycle of VPX instances on the SDX. Provisioned instances are added into CloudStack automatically - no manual configuration by the administrator is required. Once a VPX instance is added into CloudStack, it is treated the same as a VPX on an ESXi host.

Configuring SNMP Community String on a RHEL Server

The SNMP Community string is similar to a user id or password that provides access to a network device, such as router. This string is sent along with all SNMP requests. If the community string is correct, the device responds with the requested information. If the community string is incorrect, the device discards the request and does not respond.

The NetScaler device uses SNMP to communicate with the VMs. You must install SNMP and configure SNMP Community string for a secure communication between the NetScaler device and the RHEL machine.

1. Ensure that you installed SNMP on RedHat. If not, run the following command:

```
yum install net-snmp-utils
```

2. Edit the /etc/snmp/snmpd.conf file to allow the SNMP polling from the NetScaler device.
   (a) Map the community name into a security name (local and mynetwork, depending on where the request is coming from):

Note: Use a strong password instead of public when you edit the following table.

```
#          sec.name   source        community
com2sec    local      localhost     public
com2sec    mynetwork  0.0.0.0       public
```

Note: Setting to 0.0.0.0 allows all IPs to poll the NetScaler server.

(b) Map the security names into group names:

```
#        group.name    sec.model  sec.name
group    MyRWGroup     v1         local
group    MyRWGroup     v2c        local
group    MyROGroup     v1         mynetwork
group    MyROGroup     v2c        mynetwork
```

(c) Create a view to allow the groups to have the permission to:

```
incl/excl subtree mask view all included .1
```

(d) Grant access with different write permissions to the two groups to the view you created.

```
# context     sec.model     sec.level      prefix     read     write    notif
  access      MyROGroup ""  any noauth     exact      all      none     none
  access      MyRWGroup ""  any noauth     exact      all      all      all
```

3. Unblock SNMP in iptables.

```
iptables -A INPUT -p udp --dport 161 -j ACCEPT
```

4. Start the SNMP service:

```
service snmpd start
```

5. Ensure that the SNMP service is started automatically during the system startup:

```
chkconfig snmpd on
```

Initial Setup of External Firewalls and Load Balancers

When the first VM is created for a new account, CloudStack programs the external firewall and load balancer to work with the VM. The following objects are created on the firewall:

- A new logical interface to connect to the account's private VLAN. The interface IP is always the first IP of the account's private subnet (e.g. 10.1.1.1).
- A source NAT rule that forwards all outgoing traffic from the account's private VLAN to the public Internet, using the account's public IP address as the source address
- A firewall filter counter that measures the number of bytes of outgoing traffic for the account

The following objects are created on the load balancer:

- A new VLAN that matches the account's provisioned Zone VLAN

- A self IP for the VLAN. This is always the second IP of the account's private subnet (e.g. 10.1.1.2).

Ongoing Configuration of External Firewalls and Load Balancers

Additional user actions (e.g. setting a port forward) will cause further programming of the firewall and load balancer. A user may request additional public IP addresses and forward traffic received at these IPs to specific VMs. This is accomplished by enabling static NAT for a public IP address, assigning the IP to a VM, and specifying a set of protocols and port ranges to open. When a static NAT rule is created, CloudStack programs the zone's external firewall with the following objects:

- A static NAT rule that maps the public IP address to the private IP address of a VM.
- A security policy that allows traffic within the set of protocols and port ranges that are specified.
- A firewall filter counter that measures the number of bytes of incoming traffic to the public IP.

The number of incoming and outgoing bytes through source NAT, static NAT, and load balancing rules is measured and saved on each external element. This data is collected on a regular basis and stored in the CloudStack database.

Load Balancer Rules

A CloudStack user or administrator may create load balancing rules that balance traffic received at a public IP to one or more VMs. A user creates a rule, specifies an algorithm, and assigns the rule to a set of VMs.

Note: If you create load balancing rules while using a network service offering that includes an external load balancer device such as NetScaler, and later change the network service offering to one that uses the CloudStack virtual router, you must create a firewall rule on the virtual router for each of your existing load balancing rules so that they continue to function.

Adding a Load Balancer Rule

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network where you want to load balance the traffic.
4. Click View IP Addresses.
5. Click the IP address for which you want to create the rule, then click the Configuration tab.
6. In the Load Balancing node of the diagram, click View All.

   In a Basic zone, you can also create a load balancing rule without acquiring or selecting an IP address. CloudStack internally assign an IP when you create the load balancing rule, which is listed in the IP Addresses page when the rule is created.

   To do that, select the name of the network, then click Add Load Balancer tab. Continue with #7.
7. Fill in the following:
   - **Name**: A name for the load balancer rule.
   - **Public Port**: The port receiving incoming traffic to be balanced.
   - **Private Port**: The port that the VMs will use to receive the traffic.

- **Algorithm**: Choose the load balancing algorithm you want CloudStack to use. CloudStack supports a variety of well-known algorithms. If you are not familiar with these choices, you will find plenty of information about them on the Internet.
- **Stickiness**: (Optional) Click Configure and choose the algorithm for the stickiness policy. See Sticky Session Policies for Load Balancer Rules.
- **AutoScale**: Click Configure and complete the AutoScale configuration as explained in *Configuring AutoScale*.
- **Health Check**: (Optional; NetScaler load balancers only) Click Configure and fill in the characteristics of the health check policy. See *Health Checks for Load Balancer Rules*.
  - **Ping path (Optional)**: Sequence of destinations to which to send health check queries. Default: / (all).
  - **Response time (Optional)**: How long to wait for a response from the health check (2 - 60 seconds). Default: 5 seconds.
  - **Interval time (Optional)**: Amount of time between health checks (1 second - 5 minutes). Default value is set in the global configuration parameter lbrule_health check_time_interval.
  - **Healthy threshold (Optional)**: Number of consecutive health check successes that are required before declaring an instance healthy. Default: 2.
  - **Unhealthy threshold (Optional)**: Number of consecutive health check failures that are required before declaring an instance unhealthy. Default: 10.

8. Click Add VMs, then select two or more VMs that will divide the load of incoming traffic, and click Apply.

   The new load balancer rule appears in the list. You can repeat these steps to add more load balancer rules for this IP address.

Sticky Session Policies for Load Balancer Rules

Sticky sessions are used in Web-based applications to ensure continued availability of information across the multiple requests in a user's session. For example, if a shopper is filling a cart, you need to remember what has been purchased so far. The concept of "stickiness" is also referred to as persistence or maintaining state.

Any load balancer rule defined in CloudStack can have a stickiness policy. The policy consists of a name, stickiness method, and parameters. The parameters are name-value pairs or flags, which are defined by the load balancer vendor. The stickiness method could be load balancer-generated cookie, application-generated cookie, or source-based. In the source-based method, the source IP address is used to identify the user and locate the user's stored data. In the other methods, cookies are used. The cookie generated by the load balancer or application is included in request and response URLs to create persistence. The cookie name can be specified by the administrator or automatically generated. A variety of options are provided to control the exact behavior of cookies, such as how they are generated and whether they are cached.

For the most up to date list of available stickiness methods, see the CloudStack UI or call listNetworks and check the SupportedStickinessMethods capability.

Health Checks for Load Balancer Rules

(NetScaler load balancer only; requires NetScaler version 10.0)

Health checks are used in load-balanced applications to ensure that requests are forwarded only to running, available services. When creating a load balancer rule, you can specify a health check policy. This is in addition to specifying

the stickiness policy, algorithm, and other load balancer rule options. You can configure one health check policy per load balancer rule.

Any load balancer rule defined on a NetScaler load balancer in CloudStack can have a health check policy. The policy consists of a ping path, thresholds to define "healthy" and "unhealthy" states, health check frequency, and timeout wait interval.

When a health check policy is in effect, the load balancer will stop forwarding requests to any resources that are found to be unhealthy. If the resource later becomes available again, the periodic health check will discover it, and the resource will once again be added to the pool of resources that can receive requests from the load balancer. At any given time, the most recent result of the health check is displayed in the UI. For any VM that is attached to a load balancer rule with a health check configured, the state will be shown as UP or DOWN in the UI depending on the result of the most recent health check.

You can delete or modify existing health check policies.

To configure how often the health check is performed by default, use the global configuration setting healthcheck.update.interval (default value is 600 seconds). You can override this value for an individual health check policy.

For details on how to set a health check policy using the UI, see *Adding a Load Balancer Rule*.

Configuring AutoScale

AutoScaling allows you to scale your back-end services or application VMs up or down seamlessly and automatically according to the conditions you define. With AutoScaling enabled, you can ensure that the number of VMs you are using seamlessly scale up when demand increases, and automatically decreases when demand subsides. Thus it helps you save compute costs by terminating underused VMs automatically and launching new VMs when you need them, without the need for manual intervention.

NetScaler AutoScaling is designed to seamlessly launch or terminate VMs based on user-defined conditions. Conditions for triggering a scaleup or scaledown action can vary from a simple use case like monitoring the CPU usage of a server to a complex use case of monitoring a combination of server's responsiveness and its CPU usage. For example, you can configure AutoScaling to launch an additional VM whenever CPU usage exceeds 80 percent for 15 minutes, or to remove a VM whenever CPU usage is less than 20 percent for 30 minutes.

CloudStack uses the NetScaler load balancer to monitor all aspects of a system's health and work in unison with CloudStack to initiate scale-up or scale-down actions.

Note: AutoScale is supported on NetScaler Release 10 Build 74.4006.e and beyond.

Prerequisites

Before you configure an AutoScale rule, consider the following:

- Ensure that the necessary template is prepared before configuring AutoScale. When a VM is deployed by using a template and when it comes up, the application should be up and running.

  Note: If the application is not running, the NetScaler device considers the VM as ineffective and continues provisioning the VMs unconditionally until the resource limit is exhausted.

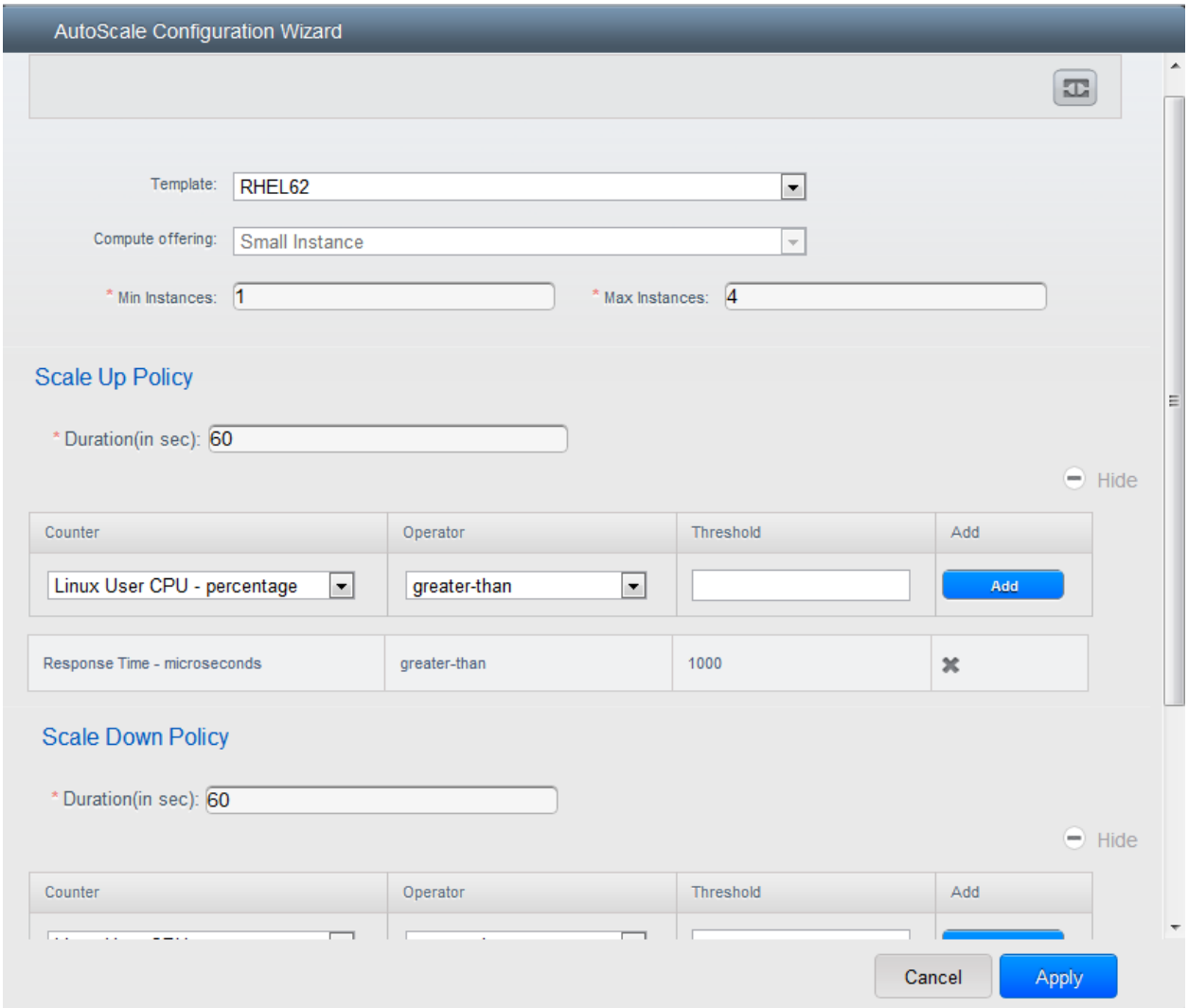
- Deploy the templates you prepared. Ensure that the applications come up on the first boot and is ready to take the traffic. Observe the time requires to deploy the template. Consider this time when you specify the quiet time while configuring AutoScale.

- The AutoScale feature supports the SNMP counters that can be used to define conditions for taking scale up or scale down actions. To monitor the SNMP-based counter, ensure that the SNMP agent is installed in the template used for creating the AutoScale VMs, and the SNMP operations work with the configured SNMP community and port by using standard SNMP managers. For example, see *"Configuring SNMP Community String on a RHELServer"* to configure SNMP on a RHEL machine.
- Ensure that the endpointe.url parameter present in the Global Settings is set to the Management Server API URL. For example, `http://10.102.102.22:8080/client/api`. In a multi-node Management Server deployment, use the virtual IP address configured in the load balancer for the management server's cluster. Additionally, ensure that the NetScaler device has access to this IP address to provide AutoScale support.

  If you update the endpointe.url, disable the AutoScale functionality of the load balancer rules in the system, then enable them back to reflect the changes. For more information see *Updating an AutoScale Configuration*.
- If the API Key and Secret Key are regenerated for an AutoScale user, ensure that the AutoScale functionality of the load balancers that the user participates in are disabled and then enabled to reflect the configuration changes in the NetScaler.
- In an advanced Zone, ensure that at least one VM should be present before configuring a load balancer rule with AutoScale. Having one VM in the network ensures that the network is in implemented state for configuring AutoScale.

Configuration

Specify the following:



- **Template**: A template consists of a base OS image and application. A template is used to provision the new instance of an application on a scaleup action. When a VM is deployed from a template, the VM can start taking the traffic from the load balancer without any admin intervention. For example, if the VM is deployed for a Web service, it should have the Web server running, the database connected, and so on.
- **Compute offering**: A predefined set of virtual hardware attributes, including CPU speed, number of CPUs, and RAM size, that the user can select when creating a new virtual machine instance. Choose one of the compute offerings to be used while provisioning a VM instance as part of scaleup action.
- **Min Instance**: The minimum number of active VM instances that is assigned to a load balancing rule. The active VM instances are the application instances that are up and serving the traffic, and are being load balanced. This parameter ensures that a load balancing rule has at least the configured number of active VM instances are available to serve the traffic.

Note: If an application, such as SAP, running on a VM instance is down for some reason, the VM is then not counted as part of Min Instance parameter, and the AutoScale feature initiates a scaleup action if the number of active VM instances is below the configured value. Similarly, when an application instance comes up from its earlier down state, this application instance is counted as part of the active instance count and the AutoScale process initiates a scaledown action when the active instance count breaches the Max instance value.

- **Max Instance**: Maximum number of active VM instances that **should be assigned to**a load balancing rule. This parameter defines the upper limit of active VM instances that can be assigned to a load balancing rule.

  Specifying a large value for the maximum instance parameter might result in provisioning large number of VM instances, which in turn leads to a single load balancing rule exhausting the VM instances limit specified at the account or domain level.

  Note: If an application, such as SAP, running on a VM instance is down for some reason, the VM is not counted as part of Max Instance parameter. So there may be scenarios where the number of VMs provisioned for a scaleup action might be more than the configured Max Instance value. Once the application instances in the VMs are up from an earlier down state, the AutoScale feature starts aligning to the configured Max Instance value.

Specify the following scale-up and scale-down policies:

- **Duration**: The duration, in seconds, for which the conditions you specify must be true to trigger a scaleup action. The conditions defined should hold true for the entire duration you specify for an AutoScale action to be invoked.
- **Counter**: The performance counters expose the state of the monitored instances. By default, CloudStack offers four performance counters: Three SNMP counters and one NetScaler counter. The SNMP counters are Linux User CPU, Linux System CPU, and Linux CPU Idle. The NetScaler counter is ResponseTime. The root administrator can add additional counters into CloudStack by using the CloudStack API.
- **Operator**: The following five relational operators are supported in AutoScale feature: Greater than, Less than, Less than or equal to, Greater than or equal to, and Equal to.
- **Threshold**: Threshold value to be used for the counter. Once the counter defined above breaches the threshold value, the AutoScale feature initiates a scaleup or scaledown action.
- **Add**: Click Add to add the condition.

Additionally, if you want to configure the advanced settings, click Show advanced settings, and specify the following:

- **Polling interval**: Frequency in which the conditions, combination of counter, operator and threshold, are to be evaluated before taking a scale up or down action. The default polling interval is 30 seconds.
- **Quiet Time**: This is the cool down period after an AutoScale action is initiated. The time includes the time taken to complete provisioning a VM instance from its template and the time taken by an application to be ready to serve traffic. This quiet time allows the fleet to come up to a stable state before any action can take place. The default is 300 seconds.
- **Destroy VM Grace Period**: The duration in seconds, after a scaledown action is initiated, to wait before the VM is destroyed as part of scaledown action. This is to ensure graceful close of any pending sessions or transactions being served by the VM marked for destroy. The default is 120 seconds.
- **Security Groups**: Security groups provide a way to isolate traffic to the VM instances. A security group is a group of VMs that filter their incoming and outgoing traffic according to a set of rules, called ingress and egress rules. These rules filter network traffic according to the IP address that is attempting to communicate with the VM.
- **Disk Offerings**: A predefined set of disk size for primary data storage.
- **SNMP Community**: The SNMP community string to be used by the NetScaler device to query the configured counter value from the provisioned VM instances. Default is public.
- **SNMP Port**: The port number on which the SNMP agent that run on the provisioned VMs is listening. Default port is 161.

- **User**: This is the user that the NetScaler device use to invoke scaleup and scaledown API calls to the cloud. If no option is specified, the user who configures AutoScaling is applied. Specify another user name to override.
- **Apply**: Click Apply to create the AutoScale configuration.

Disabling and Enabling an AutoScale Configuration

If you want to perform any maintenance operation on the AutoScale VM instances, disable the AutoScale configuration. When the AutoScale configuration is disabled, no scaleup or scaledown action is performed. You can use this downtime for the maintenance activities. To disable the AutoScale configuration, click the Disable AutoScale button.

The button toggles between enable and disable, depending on whether AutoScale is currently enabled or not. After the maintenance operations are done, you can enable the AutoScale configuration back. To enable, open the AutoScale configuration page again, then click the Enable AutoScale button.

Updating an AutoScale Configuration

You can update the various parameters and add or delete the conditions in a scaleup or scaledown rule. Before you update an AutoScale configuration, ensure that you disable the AutoScale load balancer rule by clicking the Disable AutoScale button.

After you modify the required AutoScale parameters, click Apply. To apply the new AutoScale policies, open the AutoScale configuration page again, then click the Enable AutoScale button.

Runtime Considerations

- An administrator should not assign a VM to a load balancing rule which is configured for AutoScale.
- Before a VM provisioning is completed if NetScaler is shutdown or restarted, the provisioned VM cannot be a part of the load balancing rule though the intent was to assign it to a load balancing rule. To workaround, rename the AutoScale provisioned VMs based on the rule name or ID so at any point of time the VMs can be reconciled to its load balancing rule.
- Making API calls outside the context of AutoScale, such as destroyVM, on an autoscaled VM leaves the load balancing configuration in an inconsistent state. Though VM is destroyed from the load balancer rule, NetScaler continues to show the VM as a service assigned to a rule.

5.12.17 Global Server Load Balancing Support

CloudStack supports Global Server Load Balancing (GSLB) functionalities to provide business continuity, and enable seamless resource movement within a CloudStack environment. CloudStack achieve this by extending its functionality of integrating with NetScaler Application Delivery Controller (ADC), which also provides various GSLB capabilities, such as disaster recovery and load balancing. The DNS redirection technique is used to achieve GSLB in CloudStack.

In order to support this functionality, region level services and service provider are introduced. A new service 'GSLB' is introduced as a region level service. The GSLB service provider is introduced that will provider the GSLB service. Currently, NetScaler is the supported GSLB provider in CloudStack. GSLB functionality works in an Active-Active data center environment.

About Global Server Load Balancing

Global Server Load Balancing (GSLB) is an extension of load balancing functionality, which is highly efficient in avoiding downtime. Based on the nature of deployment, GSLB represents a set of technologies that is used for various purposes, such as load sharing, disaster recovery, performance, and legal obligations. With GSLB, workloads can be distributed across multiple data centers situated at geographically separated locations. GSLB can also provide an alternate location for accessing a resource in the event of a failure, or to provide a means of shifting traffic easily to simplify maintenance, or both.

Components of GSLB

A typical GSLB environment is comprised of the following components:

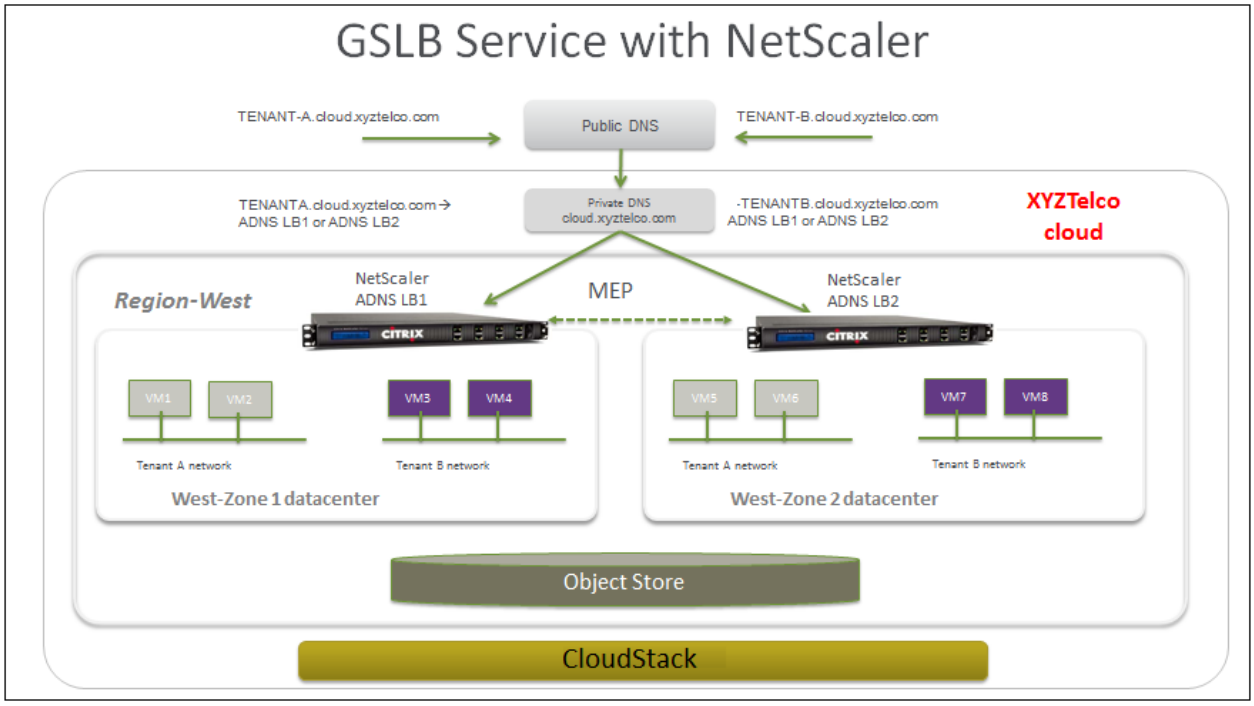
- **GSLB Site**: In CloudStack terminology, GSLB sites are represented by zones that are mapped to data centers, each of which has various network appliances. Each GSLB site is managed by a NetScaler appliance that is local to that site. Each of these appliances treats its own site as the local site and all other sites, managed by other appliances, as remote sites. It is the central entity in a GSLB deployment, and is represented by a name and an IP address.
- **GSLB Services**: A GSLB service is typically represented by a load balancing or content switching virtual server. In a GSLB environment, you can have a local as well as remote GSLB services. A local GSLB service represents a local load balancing or content switching virtual server. A remote GSLB service is the one configured at one of the other sites in the GSLB setup. At each site in the GSLB setup, you can create one local GSLB service and any number of remote GSLB services.
- **GSLB Virtual Servers**: A GSLB virtual server refers to one or more GSLB services and balances traffic between traffic across the VMs in multiple zones by using the CloudStack functionality. It evaluates the configured GSLB methods or algorithms to select a GSLB service to which to send the client requests. One or more virtual servers from different zones are bound to the GSLB virtual server. GSLB virtual server does not have a public IP associated with it, instead it will have a FQDN DNS name.
- **Load Balancing or Content Switching Virtual Servers**: According to Citrix NetScaler terminology, a load balancing or content switching virtual server represents one or many servers on the local network. Clients send their requests to the load balancing or content switching virtual server's virtual IP (VIP) address, and the virtual server balances the load across the local servers. After a GSLB virtual server selects a GSLB service representing either a local or a remote load balancing or content switching virtual server, the client sends the request to that virtual server's VIP address.
- **DNS VIPs**: DNS virtual IP represents a load balancing DNS virtual server on the GSLB service provider. The DNS requests for domains for which the GSLB service provider is authoritative can be sent to a DNS VIP.
- **Authoritative DNS**: ADNS (Authoritative Domain Name Server) is a service that provides actual answer to DNS queries, such as web site IP address. In a GSLB environment, an ADNS service responds only to DNS requests for domains for which the GSLB service provider is authoritative. When an ADNS service is configured, the service provider owns that IP address and advertises it. When you create an ADNS service, the NetScaler responds to DNS queries on the configured ADNS service IP and port.

How Does GSLB Works in CloudStack?

Global server load balancing is used to manage the traffic flow to a web site hosted on two separate zones that ideally are in different geographic locations. The following is an illustration of how GLSB functionality is provided in CloudStack: An organization, xyztelco, has set up a public cloud that spans two zones, Zone-1 and Zone-2, across geographically separated data centers that are managed by CloudStack. Tenant-A of the cloud launches a highly available solution by using xyztelco cloud. For that purpose, they launch two instances each in both the zones: VM1 and VM2 in Zone-1 and VM5 and VM6 in Zone-2. Tenant-A acquires a public IP, IP-1 in Zone-1, and configures a

load balancer rule to load balance the traffic between VM1 and VM2 instances. CloudStack orchestrates setting up a virtual server on the LB service provider in Zone-1. Virtual server 1 that is set up on the LB service provider in Zone-1 represents a publicly accessible virtual server that client reaches at IP-1. The client traffic to virtual server 1 at IP-1 will be load balanced across VM1 and VM2 instances.

Tenant-A acquires another public IP, IP-2 in Zone-2 and sets up a load balancer rule to load balance the traffic between VM5 and VM6 instances. Similarly in Zone-2, CloudStack orchestrates setting up a virtual server on the LB service provider. Virtual server 2 that is setup on the LB service provider in Zone-2 represents a publicly accessible virtual server that client reaches at IP-2. The client traffic that reaches virtual server 2 at IP-2 is load balanced across VM5 and VM6 instances. At this point Tenant-A has the service enabled in both the zones, but has no means to set up a disaster recovery plan if one of the zone fails. Additionally, there is no way for Tenant-A to load balance the traffic intelligently to one of the zones based on load, proximity and so on. The cloud administrator of xyztelco provisions a GSLB service provider to both the zones. A GSLB provider is typically an ADC that has the ability to act as an ADNS (Authoritative Domain Name Server) and has the mechanism to monitor health of virtual servers both at local and remote sites. The cloud admin enables GSLB as a service to the tenants that use zones 1 and 2.



Tenant-A wishes to leverage the GSLB service provided by the xyztelco cloud. Tenant-A configures a GSLB rule to load balance traffic across virtual server 1 at Zone-1 and virtual server 2 at Zone-2. The domain name is provided as A.xyztelco.com. CloudStack orchestrates setting up GSLB virtual server 1 on the GSLB service provider at Zone-1. CloudStack binds virtual server 1 of Zone-1 and virtual server 2 of Zone-2 to GLSB virtual server 1. GSLB virtual server 1 is configured to start monitoring the health of virtual server 1 and 2 in Zone-1. CloudStack will also orchestrate setting up GSLB virtual server 2 on GSLB service provider at Zone-2. CloudStack will bind virtual server 1 of Zone-1 and virtual server 2 of Zone-2 to GLSB virtual server 2. GSLB virtual server 2 is configured to start monitoring the health of virtual server 1 and 2. CloudStack will bind the domain A.xyztelco.com to both the GSLB virtual server 1 and 2. At this point, Tenant-A service will be globally reachable at A.xyztelco.com. The private DNS server for the domain xyztelcom.com is configured by the admin out-of-band to resolve the domain A.xyztelco.com to the GSLB providers at both the zones, which are configured as ADNS for the domain A.xyztelco.com. A client when sends a DNS request to resolve A.xyztelcom.com, will eventually get DNS delegation to the address of GSLB providers at zone 1 and 2. A client DNS request will be received by the GSLB provider. The GSLB provider, depending on the domain for which it needs to resolve, will pick up the GSLB virtual server associated with the domain. Depending on the health of the virtual servers being load balanced, DNS request for the domain will be resolved to the public IP associated with the selected virtual server.

Configuring GSLB

To configure a GSLB deployment, you must first configure a standard load balancing setup for each zone. This enables you to balance load across the different servers in each zone in the region. Then on the NetScaler side, configure both NetScaler appliances that you plan to add to each zone as authoritative DNS (ADNS) servers. Next, create a GSLB site for each zone, configure GSLB virtual servers for each site, create GLSB services, and bind the GSLB services to the GSLB virtual servers. Finally, bind the domain to the GSLB virtual servers. The GSLB configurations on the two appliances at the two different zones are identical, although each sites load-balancing configuration is specific to that site.

Perform the following as a cloud administrator. As per the example given above, the administrator of xyztelco is the one who sets up GSLB:

1. In the cloud.dns.name global parameter, specify the DNS name of your tenant's cloud that make use of the GSLB service.
2. On the NetScaler side, configure GSLB as given in Configuring Global Server Load Balancing (GSLB):
   (a) Configuring a standard load balancing setup.
   (b) Configure Authoritative DNS, as explained in Configuring an Authoritative DNS Service.
   (c) Configure a GSLB site with site name formed from the domain name details.

       Configure a GSLB site with the site name formed from the domain name.

       As per the example given above, the site names are A.xyztelco.com and B.xyztelco.com.

       For more information, see Configuring a Basic GSLB Site.
   (d) Configure a GSLB virtual server.

       For more information, see Configuring a GSLB Virtual Server.
   (e) Configure a GSLB service for each virtual server.

       For more information, see Configuring a GSLB Service.
   (f) Bind the GSLB services to the GSLB virtual server.

       For more information, see Binding GSLB Services to a GSLB Virtual Server.
   (g) Bind domain name to GSLB virtual server. Domain name is obtained from the domain details.

       For more information, see Binding a Domain to a GSLB Virtual Server.
3. In each zone that are participating in GSLB, add GSLB-enabled NetScaler device.

   For more information, see *Enabling GSLB in NetScaler*.

As a domain administrator/ user perform the following:

1. Add a GSLB rule on both the sites.

   See "*Adding a GSLB Rule*".
2. Assign load balancer rules.

   See "*Assigning Load Balancing Rules to GSLB*".

Prerequisites and Guidelines

- The GSLB functionality is supported both Basic and Advanced zones.
- GSLB is added as a new network service.

- GSLB service provider can be added to a physical network in a zone.
- The admin is allowed to enable or disable GSLB functionality at region level.
- The admin is allowed to configure a zone as GSLB capable or enabled.

  A zone shall be considered as GSLB capable only if a GSLB service provider is provisioned in the zone.
- When users have VMs deployed in multiple availability zones which are GSLB enabled, they can use the GSLB functionality to load balance traffic across the VMs in multiple zones.
- The users can use GSLB to load balance across the VMs across zones in a region only if the admin has enabled GSLB in that region.
- The users can load balance traffic across the availability zones in the same region or different regions.
- The admin can configure DNS name for the entire cloud.
- The users can specify an unique name across the cloud for a globally load balanced service. The provided name is used as the domain name under the DNS name associated with the cloud.

  The user-provided name along with the admin-provided DNS name is used to produce a globally resolvable FQDN for the globally load balanced service of the user. For example, if the admin has configured xyztelco.com as the DNS name for the cloud, and user specifies 'foo' for the GSLB virtual service, then the FQDN name of the GSLB virtual service is foo.xyztelco.com.
- While setting up GSLB, users can select a load balancing method, such as round robin, for using across the zones that are part of GSLB.
- The user shall be able to set weight to zone-level virtual server. Weight shall be considered by the load balancing method for distributing the traffic.
- The GSLB functionality shall support session persistence, where series of client requests for particular domain name is sent to a virtual server on the same zone.

  Statistics is collected from each GSLB virtual server.

Enabling GSLB in NetScaler

In each zone, add GSLB-enabled NetScaler device for load balancing.

1. Log in as administrator to the CloudStack UI.
2. In the left navigation bar, click Infrastructure.
3. In Zones, click View More.
4. Choose the zone you want to work with.
5. Click the Physical Network tab, then click the name of the physical network.
6. In the Network Service Providers node of the diagram, click Configure.

   You might have to scroll down to see this.
7. Click NetScaler.
8. Click Add NetScaler device and provide the following:

   For NetScaler:

   - **IP Address**: The IP address of the SDX.
   - **Username/Password**: The authentication credentials to access the device. CloudStack uses these credentials to access the device.

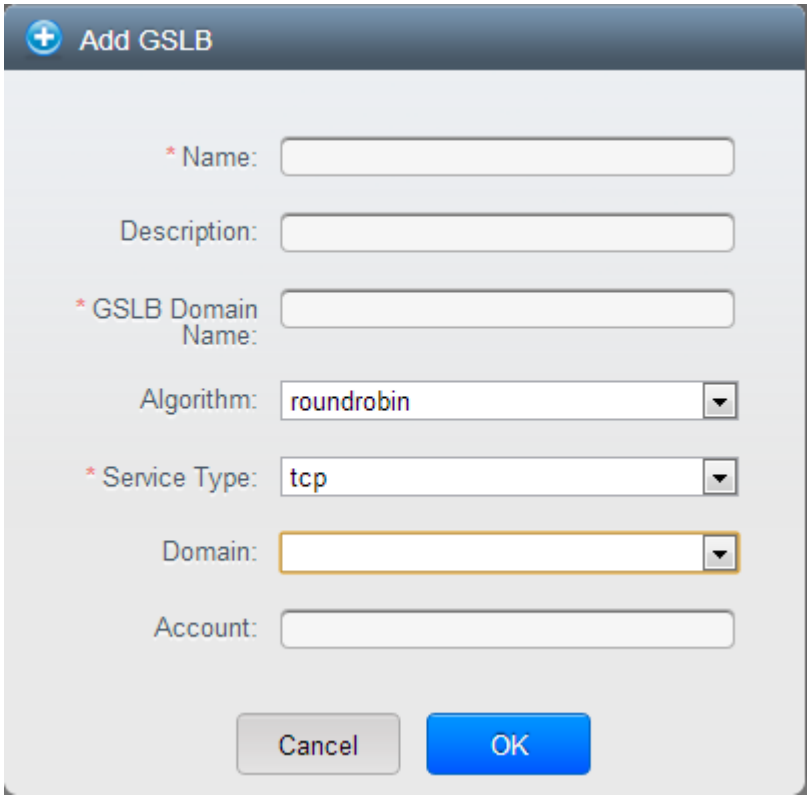
- **Type**: The type of device that is being added. It could be F5 Big Ip Load Balancer, NetScaler VPX, NetScaler MPX, or NetScaler SDX. For a comparison of the NetScaler types, see the CloudStack Administration Guide.
- **Public interface**: Interface of device that is configured to be part of the public network.
- **Private interface**: Interface of device that is configured to be part of the private network.
- **GSLB service**: Select this option.
- **GSLB service Public IP**: The public IP address of the NAT translator for a GSLB service that is on a private network.
- **GSLB service Private IP**: The private IP of the GSLB service.
- **Number of Retries**. Number of times to attempt a command on the device before considering the operation failed. Default is 2.
- **Capacity**: The number of networks the device can handle.
- **Dedicated**: When marked as dedicated, this device will be dedicated to a single account. When Dedicated is checked, the value in the Capacity field has no significance implicitly, its value is 1.

9. Click OK.

Adding a GSLB Rule

1. Log in to the CloudStack UI as a domain administrator or user.
2. In the left navigation pane, click Region.
3. Select the region for which you want to create a GSLB rule.
4. In the Details tab, click View GSLB.
5. Click Add GSLB.

   The Add GSLB page is displayed as follows:



6. Specify the following:
   - **Name**: Name for the GSLB rule.
   - **Description**: (Optional) A short description of the GSLB rule that can be displayed to users.
   - **GSLB Domain Name**: A preferred domain name for the service.
   - **Algorithm**: (Optional) The algorithm to use to load balance the traffic across the zones. The options are Round Robin, Least Connection, and Proximity.
   - **Service Type**: The transport protocol to use for GSLB. The options are TCP and UDP.
   - **Domain**: (Optional) The domain for which you want to create the GSLB rule.
   - **Account**: (Optional) The account on which you want to apply the GSLB rule.
7. Click OK to confirm.

Assigning Load Balancing Rules to GSLB

1. Log in to the CloudStack UI as a domain administrator or user.
2. In the left navigation pane, click Region.
3. Select the region for which you want to create a GSLB rule.
4. In the Details tab, click View GSLB.
5. Select the desired GSLB.
6. Click view assigned load balancing.
7. Click assign more load balancing.

8. Select the load balancing rule you have created for the zone.
9. Click OK to confirm.

Known Limitation

Currently, CloudStack does not support orchestration of services across the zones. The notion of services and service providers in region are to be introduced.

5.12.18 Guest IP Ranges

The IP ranges for guest network traffic are set on a per-account basis by the user. This allows the users to configure their network in a fashion that will enable VPN linking between their guest network and their clients.

In shared networks in Basic zone and Security Group-enabled Advanced networks, you will have the flexibility to add multiple guest IP ranges from different subnets. You can add or remove one IP range at a time. For more information, see *"About Multiple IP Ranges"*.

5.12.19 Acquiring a New IP Address

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network where you want to work with.
4. Click View IP Addresses.
5. Click Acquire New IP.

   The Acquire New IP window is displayed.
6. Specify whether you want cross-zone IP or not.

   If you want Portable IP click Yes in the confirmation dialog. If you want a normal Public IP click No.

   For more information on Portable IP, see *"Portable IPs"*.

   Within a few moments, the new IP address should appear with the state Allocated. You can now use the IP address in port forwarding or static NAT rules.

5.12.20 Releasing an IP Address

When the last rule for an IP address is removed, you can release that IP address. The IP address still belongs to the VPC; however, it can be picked up for any guest network again.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network where you want to work with.
4. Click View IP Addresses.
5. Click the IP address you want to release.
6. Click the Release IP button.

5.12.21 Static NAT

A static NAT rule maps a public IP address to the private IP address of a VM in order to allow Internet traffic into the VM. The public IP address always remains the same, which is why it is called static NAT. This section tells how to enable or disable static NAT for a particular IP address.

Enabling or Disabling Static NAT

If port forwarding rules are already in effect for an IP address, you cannot enable static NAT to that IP.

If a guest VM is part of more than one network, static NAT rules will function only if they are defined on the default network.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network where you want to work with.
4. Click View IP Addresses.
5. Click the IP address you want to work with.
6. Click the Static NAT button.

   The button toggles between Enable and Disable, depending on whether static NAT is currently enabled for the IP address.
7. If you are enabling static NAT, a dialog appears where you can choose the destination VM and click Apply.

5.12.22 IP Forwarding and Firewalling

By default, all incoming traffic to the public IP address is rejected. All outgoing traffic from the guests is also blocked by default.

To allow outgoing traffic, follow the procedure in *Egress Firewall Rules in an Advanced Zone*.

To allow incoming traffic, users may set up firewall rules and/or port forwarding rules. For example, you can use a firewall rule to open a range of ports on the public IP address, such as 33 through 44. Then use port forwarding rules to direct traffic from individual ports within that range to specific ports on user VMs. For example, one port forwarding rule could route incoming traffic on the public IP's port 33 to port 100 on one user VM's private IP.

Firewall Rules

By default, all incoming traffic to the public IP address is rejected by the firewall. To allow external traffic, you can open firewall ports by specifying firewall rules. You can optionally specify one or more CIDRs to filter the source IPs. This is useful when you want to allow only incoming requests from certain IP addresses.

You cannot use firewall rules to open ports for an elastic IP address. When elastic IP is used, outside access is instead controlled through the use of security groups. See *"Adding a Security Group"*.

In an advanced zone, you can also create egress firewall rules by using the virtual router. For more information, see "*Egress Firewall Rules in an Advanced Zone*".

Firewall rules can be created using the Firewall tab in the Management Server UI. This tab is not displayed by default when CloudStack is installed. To display the Firewall tab, the CloudStack administrator must set the global configuration parameter firewall.rule.ui.enabled to "true."

To create a firewall rule:

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. Click the name of the network where you want to work with.
4. Click View IP Addresses.
5. Click the IP address you want to work with.
6. Click the Configuration tab and fill in the following values.
   - **Source CIDR**: (Optional) To accept only traffic from IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. Example: 192.168.0.0/22. Leave empty to allow all CIDRs.
   - **Protocol**: The communication protocol in use on the opened port(s).
   - **Start Port and End Port**: The port(s) you want to open on the firewall. If you are opening a single port, use the same number in both fields
   - **ICMP Type and ICMP Code**: Used only if Protocol is set to ICMP. Provide the type and code required by the ICMP protocol to fill out the ICMP header. Refer to ICMP documentation for more details if you are not sure what to enter
7. Click Add.

Egress Firewall Rules in an Advanced Zone

The egress traffic originates from a private network to a public network, such as the Internet. By default, the egress traffic is blocked in default network offerings, so no outgoing traffic is allowed from a guest network to the Internet. However, you can control the egress traffic in an Advanced zone by creating egress firewall rules. When an egress firewall rule is applied, the traffic specific to the rule is allowed and the remaining traffic is blocked. When all the firewall rules are removed the default policy, Block, is applied.

Prerequisites and Guidelines

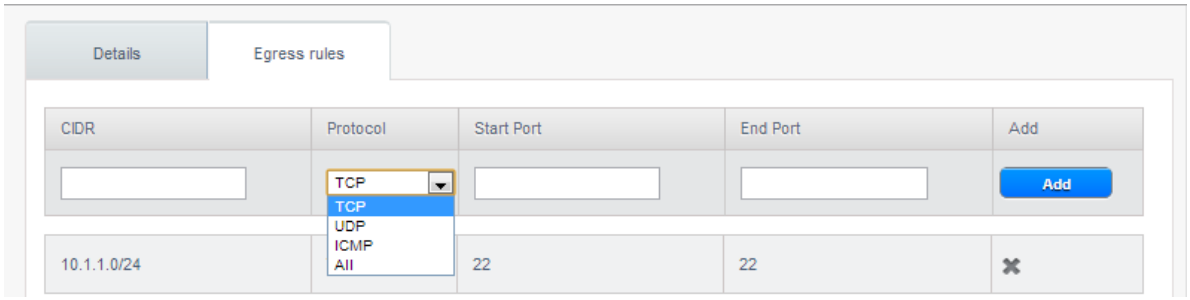
Consider the following scenarios to apply egress firewall rules:

- Egress firewall rules are supported on Juniper SRX and virtual router.
- The egress firewall rules are not supported on shared networks.
- Allow the egress traffic from specified source CIDR. The Source CIDR is part of guest network CIDR.
- Allow the egress traffic with protocol TCP,UDP,ICMP, or ALL.
- Allow the egress traffic with protocol and destination port range. The port range is specified for TCP, UDP or for ICMP type and code.
- The default policy is Allow for the new network offerings, whereas on upgrade existing network offerings with firewall service providers will have the default egress policy Deny.

Configuring an Egress Firewall Rule

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In Select view, choose Guest networks, then click the Guest network you want.

4. To add an egress rule, click the Egress rules tab and fill out the following fields to specify what type of traffic is allowed to be sent out of VM instances in this guest network:



   - **CIDR**: (Add by CIDR only) To send traffic only to the IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. The CIDR is the base IP address of the destination. For example, 192.168.0.0/22. To allow all CIDRs, set to 0.0.0.0/0.
   - **Protocol**: The networking protocol that VMs uses to send outgoing traffic. The TCP and UDP protocols are typically used for data exchange and end-user communications. The ICMP protocol is typically used to send error messages or network monitoring data.
   - **Start Port, End Port**: (TCP, UDP only) A range of listening ports that are the destination for the outgoing traffic. If you are opening a single port, use the same number in both fields.
   - **ICMP Type, ICMP Code**: (ICMP only) The type of message and error code that are sent.

5. Click Add.

Configuring the Default Egress Policy

The default egress policy for Isolated guest network is configured by using Network offering. Use the create network offering option to determine whether the default policy should be block or allow all the traffic to the public network from a guest network. Use this network offering to create the network. If no policy is specified, by default all the traffic is allowed from the guest network that you create by using this network offering.

You have two options: Allow and Deny.

Allow

If you select Allow for a network offering, by default egress traffic is allowed. However, when an egress rule is configured for a guest network, rules are applied to block the specified traffic and rest are allowed. If no egress rules are configured for the network, egress traffic is accepted.

Deny

If you select Deny for a network offering, by default egress traffic for the guest network is blocked. However, when an egress rules is configured for a guest network, rules are applied to allow the specified traffic. While implementing a guest network, CloudStack adds the firewall egress rule specific to the default egress policy for the guest network.

This feature is supported only on virtual router and Juniper SRX.

1. Create a network offering with your desirable default egress policy:
   (a) Log in with admin privileges to the CloudStack UI.
   (b) In the left navigation bar, click Service Offerings.

(c) In Select Offering, choose Network Offering.

(d) Click Add Network Offering.

(e) In the dialog, make necessary choices, including firewall provider.

(f) In the Default egress policy field, specify the behaviour.

(g) Click OK.

2. Create an isolated network by using this network offering.

   Based on your selection, the network will have the egress public traffic blocked or allowed.

Port Forwarding

A port forward service is a set of port forwarding rules that define a policy. A port forward service is then applied to one or more guest VMs. The guest VM then has its inbound network access managed according to the policy defined by the port forwarding service. You can optionally specify one or more CIDRs to filter the source IPs. This is useful when you want to allow only incoming requests from certain IP addresses to be forwarded.

A guest VM can be in any number of port forward services. Port forward services can be defined but have no members. If a guest VM is part of more than one network, port forwarding rules will function only if they are defined on the default network

You cannot use port forwarding to open ports for an elastic IP address. When elastic IP is used, outside access is instead controlled through the use of security groups. See Security Groups.

To set up port forwarding:

1. Log in to the CloudStack UI as an administrator or end user.
2. If you have not already done so, add a public IP address range to a zone in CloudStack. See Adding a Zone and Pod in the Installation Guide.
3. Add one or more VM instances to CloudStack.
4. In the left navigation bar, click Network.
5. Click the name of the guest network where the VMs are running.
6. Choose an existing IP address or acquire a new IP address. See *"Acquiring a New IP Address"*. Click the name of the IP address in the list.
7. Click the Configuration tab.
8. In the Port Forwarding node of the diagram, click View All.
9. Fill in the following:
   - **Public Port**: The port to which public traffic will be addressed on the IP address you acquired in the previous step.
   - **Private Port**: The port on which the instance is listening for forwarded public traffic.
   - **Protocol**: The communication protocol in use between the two ports
10. Click Add.

5.12.23 IP Load Balancing

The user may choose to associate the same public IP for multiple guests. CloudStack implements a TCP-level load balancer with the following policies.

- Round-robin
- Least connection
- Source IP

This is similar to port forwarding but the destination may be multiple IP addresses.

5.12.24 DNS and DHCP

The Virtual Router provides DNS and DHCP services to the guests. It proxies DNS requests to the DNS server configured on the Availability Zone.

5.12.25 Remote Access VPN

CloudStack account owners can create virtual private networks (VPN) to access their virtual machines. If the guest network is instantiated from a network offering that offers the Remote Access VPN service, the virtual router (based on the System VM) is used to provide the service. CloudStack provides a L2TP-over-IPsec-based remote access VPN service to guest virtual networks. Since each network gets its own virtual router, VPNs are not shared across the networks. VPN clients native to Windows, Mac OS X and iOS can be used to connect to the guest networks. The account owner can create and manage users for their VPN. CloudStack does not use its account database for this purpose but uses a separate table. The VPN user database is shared across all the VPNs created by the account owner. All VPN users get access to all VPNs created by the account owner.

Note: Make sure that not all traffic goes through the VPN. That is, the route installed by the VPN should be only for the guest network and not for all traffic.

- **Road Warrior / Remote Access**. Users want to be able to connect securely from a home or office to a private network in the cloud. Typically, the IP address of the connecting client is dynamic and cannot be preconfigured on the VPN server.
- **Site to Site**. In this scenario, two private subnets are connected over the public Internet with a secure VPN tunnel. The cloud user's subnet (for example, an office network) is connected through a gateway to the network in the cloud. The address of the user's gateway must be preconfigured on the VPN server in the cloud. Note that although L2TP-over-IPsec can be used to set up Site-to-Site VPNs, this is not the primary intent of this feature. For more information, see "*Setting Up a Site-to-Site VPN Connection*".

Configuring Remote Access VPN

To set up VPN for the cloud:

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, click Global Settings.
3. Set the following global configuration parameters.
    - remote.access.vpn.client.ip.range - The range of IP addresses to be allocated to remote access VPN clients. The first IP in the range is used by the VPN server.
    - remote.access.vpn.psk.length - Length of the IPSec key.
    - remote.access.vpn.user.limit - Maximum number of VPN users per account.

To enable VPN for a particular network:

1. Log in as a user or administrator to the CloudStack UI.
2. In the left navigation, click Network.
3. Click the name of the network you want to work with.
4. Click View IP Addresses.
5. Click one of the displayed IP address names.
6. Click the Enable VPN button.

   The IPsec key is displayed in a popup window.

Configuring Remote Access VPN in VPC

On enabling Remote Access VPN on a VPC, any VPN client present outside the VPC can access VMs present in the VPC by using the Remote VPN connection. The VPN client can be present anywhere except inside the VPC on which the user enabled the Remote Access VPN service.

To enable VPN for a VPC:

1. Log in as a user or administrator to the CloudStack UI.
2. In the left navigation, click Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC.

   For each tier, the following options are displayed:

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists
5. In the Router node, select Public IP Addresses.

   The IP Addresses page is displayed.
6. Click Source NAT IP address.
7. Click the Enable VPN button.

   Click OK to confirm. The IPsec key is displayed in a pop-up window.

Now, you need to add the VPN users.

1. Click the Source NAT IP.

2. Select the VPN tab.
3. Add the username and the corresponding password of the user you wanted to add.
4. Click Add.
5. Repeat the same steps to add the VPN users.

Setting Up a Site-to-Site VPN Connection

A Site-to-Site VPN connection helps you establish a secure connection from an enterprise datacenter to the cloud infrastructure. This allows users to access the guest VMs by establishing a VPN connection to the virtual router of the account from a device in the datacenter of the enterprise. You can also establish a secure connection between two VPC setups or high availability zones in your environment. Having this facility eliminates the need to establish VPN connections to individual VMs.

The difference from Remote VPN is that Site-to-site VPNs connects entire networks to each other, for example, connecting a branch office network to a company headquarters network. In a site-to-site VPN, hosts do not have VPN client software; they send and receive normal TCP/IP traffic through a VPN gateway.

The supported endpoints on the remote datacenters are:

- Cisco ISR with IOS 12.4 or later
- Juniper J-Series routers with JunOS 9.5 or later
- CloudStack virtual routers

Note: In addition to the specific Cisco and Juniper devices listed above, the expectation is that any Cisco or Juniper device running on the supported operating systems are able to establish VPN connections.

To set up a Site-to-Site VPN connection, perform the following:

1. Create a Virtual Private Cloud (VPC).

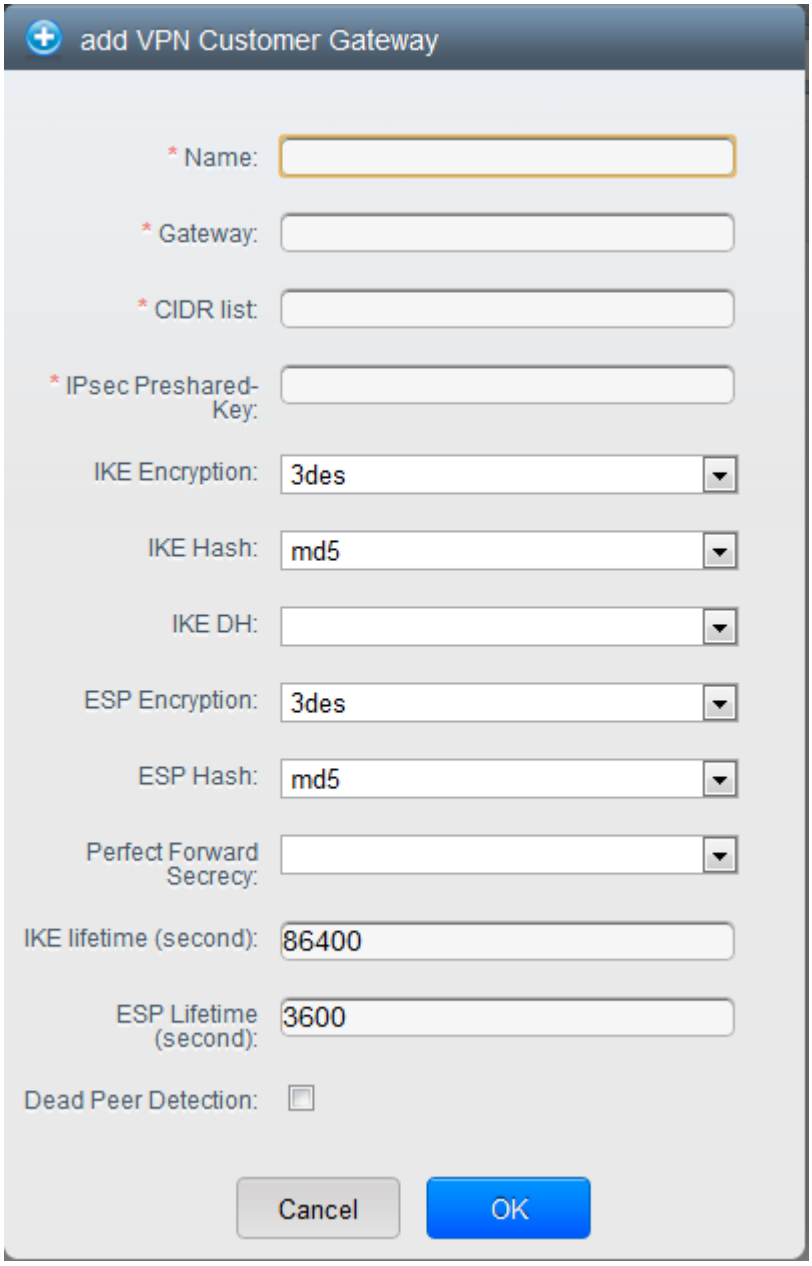
   See "*Configuring a Virtual Private Cloud*".
2. Create a VPN Customer Gateway.
3. Create a VPN gateway for the VPC that you created.
4. Create VPN connection from the VPC VPN gateway to the customer VPN gateway.

Creating and Updating a VPN Customer Gateway

Note: A VPN customer gateway can be connected to only one VPN gateway at a time.

To add a VPN Customer Gateway:

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPN Customer Gateway.
4. Click Add VPN Customer Gateway.



Provide the following information:

- **Name**: A unique name for the VPN customer gateway you create.
- **Gateway**: The IP address for the remote gateway.
- **CIDR list**: The guest CIDR list of the remote subnets. Enter a CIDR or a comma-separated list of CIDRs. Ensure that a guest CIDR list is not overlapped with the VPC's CIDR, or another guest CIDR. The CIDR must be RFC1918-compliant.
- **IPsec Preshared Key**: Preshared keying is a method where the endpoints of the VPN share a secret key. This key value is used to authenticate the customer gateway and the VPC VPN gateway to each other. The sequence cannot contain a newline or double-quote.

Note: The IKE peers (VPN end points) authenticate each other by computing and sending a keyed hash

of data that includes the Preshared key. If the receiving peer is able to create the same hash independently by using its Preshared key, it knows that both peers must share the same secret, thus authenticating the customer gateway.

- **IKE Encryption**: The Internet Key Exchange (IKE) policy for phase-1. The supported encryption algorithms are AES128, AES192, AES256, and 3DES. Authentication is accomplished through the Preshared Keys.

  Note: The phase-1 is the first phase in the IKE process. In this initial negotiation phase, the two VPN endpoints agree on the methods to be used to provide security for the underlying IP traffic. The phase-1 authenticates the two VPN gateways to each other, by confirming that the remote gateway has a matching Preshared Key.

- **IKE Hash**: The IKE hash for phase-1. The supported hash algorithms are SHA1 and MD5.
- **IKE DH**: A public-key cryptography protocol which allows two parties to establish a shared secret over an insecure communications channel. The 1536-bit Diffie-Hellman group is used within IKE to establish session keys. The supported options are None, Group-5 (1536-bit) and Group-2 (1024-bit).
- **ESP Encryption**: Encapsulating Security Payload (ESP) algorithm within phase-2. The supported encryption algorithms are AES128, AES192, AES256, and 3DES.

  Note: The phase-2 is the second phase in the IKE process. The purpose of IKE phase-2 is to negotiate IPSec security associations (SA) to set up the IPSec tunnel. In phase-2, new keying material is extracted from the Diffie-Hellman key exchange in phase-1, to provide session keys to use in protecting the VPN data flow.

- **ESP Hash**: Encapsulating Security Payload (ESP) hash for phase-2. Supported hash algorithms are SHA1 and MD5.
- **Perfect Forward Secrecy**: Perfect Forward Secrecy (or PFS) is the property that ensures that a session key derived from a set of long-term public and private keys will not be compromised. This property enforces a new Diffie-Hellman key exchange. It provides the keying material that has greater key material life and thereby greater resistance to cryptographic attacks. The available options are None, Group-5 (1536-bit) and Group-2 (1024-bit). The security of the key exchanges increase as the DH groups grow larger, as does the time of the exchanges.

  Note: When PFS is turned on, for every negotiation of a new phase-2 SA the two gateways must generate a new set of phase-1 keys. This adds an extra layer of protection that PFS adds, which ensures if the phase-2 SA's have expired, the keys used for new phase-2 SA's have not been generated from the current phase-1 keying material.

- **IKE Lifetime (seconds)**: The phase-1 lifetime of the security association in seconds. Default is 86400 seconds (1 day). Whenever the time expires, a new phase-1 exchange is performed.
- **ESP Lifetime (seconds)**: The phase-2 lifetime of the security association in seconds. Default is 3600 seconds (1 hour). Whenever the value is exceeded, a re-key is initiated to provide a new IPsec encryption and authentication session keys.
- **Dead Peer Detection**: A method to detect an unavailable Internet Key Exchange (IKE) peer. Select this option if you want the virtual router to query the liveliness of its IKE peer at regular intervals. It's recommended to have the same configuration of DPD on both side of VPN connection.

5. Click OK.

Updating and Removing a VPN Customer Gateway

You can update a customer gateway either with no VPN connection, or related VPN connection is in error state.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPN Customer Gateway.
4. Select the VPN customer gateway you want to work with.
5. To modify the required parameters, click the Edit VPN Customer Gateway button
6. To remove the VPN customer gateway, click the Delete VPN Customer Gateway button
7. Click OK.

Creating a VPN gateway for the VPC

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.

   For each tier, the following options are displayed:

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists
5. Select Site-to-Site VPN.

   If you are creating the VPN gateway for the first time, selecting Site-to-Site VPN prompts you to create a VPN gateway.

6. In the confirmation dialog, click Yes to confirm.

   Within a few moments, the VPN gateway is created. You will be prompted to view the details of the VPN gateway you have created. Click Yes to confirm.

   The following details are displayed in the VPN Gateway page:

   - IP Address
   - Account
   - Domain

Creating a VPN Connection

Note: CloudStack supports creating up to 8 VPN connections.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you create for the account are listed in the page.
4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.
5. Click the Settings icon.

   For each tier, the following options are displayed:

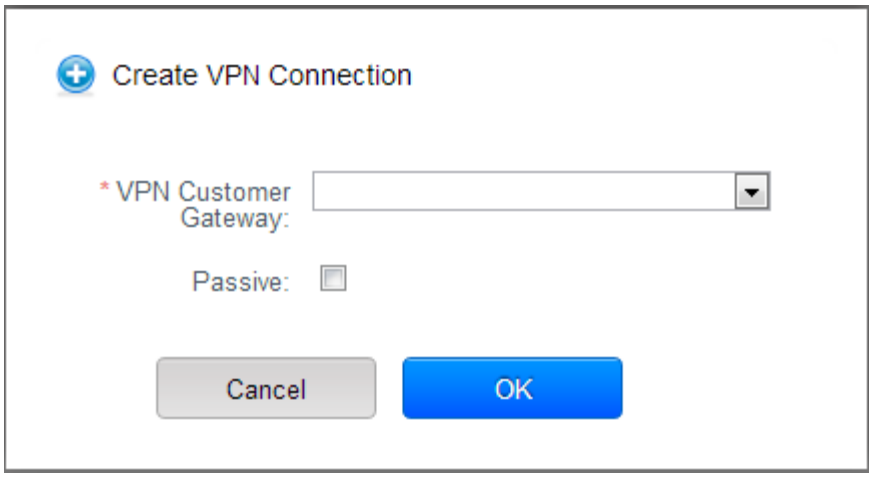
   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists
6. Select Site-to-Site VPN.

   The Site-to-Site VPN page is displayed.
7. From the Select View drop-down, ensure that VPN Connection is selected.
8. Click Create VPN Connection.

   The Create VPN Connection dialog is displayed:



9. Select the desired customer gateway.
10. Select Passive if you want to establish a connection between two VPC virtual routers.

    If you want to establish a connection between two VPC virtual routers, select Passive only on one of the VPC virtual routers, which waits for the other VPC virtual router to initiate the connection. Do not select Passive on the VPC virtual router that initiates the connection.
11. Click OK to confirm.

    Within a few moments, the VPN Connection is displayed.

    The following information on the VPN connection is displayed:

    - IP Address
    - Gateway
    - State
    - IPSec Preshared Key
    - IKE Policy
    - ESP Policy

Site-to-Site VPN Connection Between VPC Networks

CloudStack provides you with the ability to establish a site-to-site VPN connection between CloudStack virtual routers. To achieve that, add a passive mode Site-to-Site VPN. With this functionality, users can deploy applications in multiple Availability Zones or VPCs, which can communicate with each other by using a secure Site-to-Site VPN Tunnel.

This feature is supported on all the hypervisors.

1. Create two VPCs. For example, VPC A and VPC B.

   For more information, see "*Configuring a Virtual Private Cloud*".
2. Create VPN gateways on both the VPCs you created.

   For more information, see *"Creating a VPN gateway for the VPC"*.
3. Create VPN customer gateway for both the VPCs.

   For more information, see *"Creating and Updating a VPN Customer Gateway"*.

4. Enable a VPN connection on VPC A in passive mode.

   For more information, see *"Creating a VPN Connection"*.

   Ensure that the customer gateway is pointed to VPC B. The VPN connection is shown in the Disconnected state.

5. Enable a VPN connection on VPC B.

   Ensure that the customer gateway is pointed to VPC A. Because virtual router of VPC A, in this case, is in passive mode and is waiting for the virtual router of VPC B to initiate the connection, VPC B virtual router should not be in passive mode.

   The VPN connection is shown in the Disconnected state.

   Creating VPN connection on both the VPCs initiates a VPN connection. Wait for few seconds. The default is 30 seconds for both the VPN connections to show the Connected state.

Restarting and Removing a VPN Connection

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.

5. Click the Settings icon.

   For each tier, the following options are displayed:

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists

6. Select Site-to-Site VPN.

   The Site-to-Site VPN page is displayed.

7. From the Select View drop-down, ensure that VPN Connection is selected.

   All the VPN connections you created are displayed.

8. Select the VPN connection you want to work with.

   The Details tab is displayed.

9. To remove a VPN connection, click the Delete VPN connection button

   To restart a VPN connection, click the Reset VPN connection button present in the Details tab.

5.12.26 About Inter-VLAN Routing (nTier Apps)

Inter-VLAN Routing (nTier Apps) is the capability to route network traffic between VLANs. This feature enables you to build Virtual Private Clouds (VPC), an isolated segment of your cloud, that can hold multi-tier applications. These tiers are deployed on different VLANs that can communicate with each other. You provision VLANs to the tiers your create, and VMs can be deployed on different tiers. The VLANs are connected to a virtual router, which facilitates communication between the VMs. In effect, you can segment VMs by means of VLANs into different networks that can host multi-tier applications, such as Web, Application, or Database. Such segmentation by means of VLANs logically separate application VMs for higher security and lower broadcasts, while remaining physically connected to the same device.

This feature is supported on XenServer, KVM, and VMware hypervisors.

The major advantages are:

- The administrator can deploy a set of VLANs and allow users to deploy VMs on these VLANs. A guest VLAN is randomly alloted to an account from a pre-specified set of guest VLANs. All the VMs of a certain tier of an account reside on the guest VLAN allotted to that account.

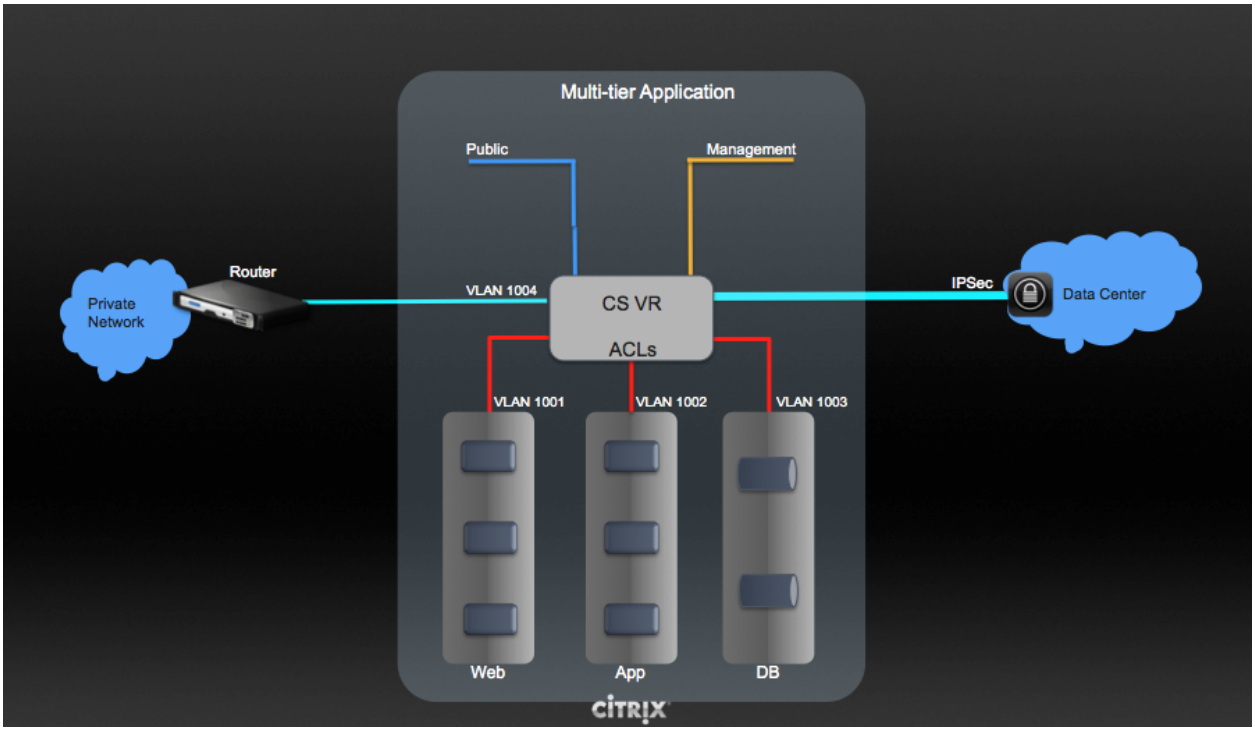
  Note: A VLAN allocated for an account cannot be shared between multiple accounts.

- The administrator can allow users create their own VPC and deploy the application. In this scenario, the VMs that belong to the account are deployed on the VLANs allotted to that account.
- Both administrators and users can create multiple VPCs. The guest network NIC is plugged to the VPC virtual router when the first VM is deployed in a tier.
- The administrator can create the following gateways to send to or receive traffic from the VMs:
  - **VPN Gateway**: For more information, see *"Creating a VPN gateway for the VPC"*.
  - **Public Gateway**: The public gateway for a VPC is added to the virtual router when the virtual router is created for VPC. The public gateway is not exposed to the end users. You are not allowed to list it, nor allowed to create any static routes.
  - **Private Gateway**: For more information, see "*Adding a Private Gateway to a VPC*".
- Both administrators and users can create various possible destinations-gateway combinations. However, only one gateway of each type can be used in a deployment.

  For example:

  - **VLANs and Public Gateway**: For example, an application is deployed in the cloud, and the Web application VMs communicate with the Internet.
  - **VLANs, VPN Gateway, and Public Gateway**: For example, an application is deployed in the cloud; the Web application VMs communicate with the Internet; and the database VMs communicate with the on-premise devices.
- The administrator can define Network Access Control List (ACL) on the virtual router to filter the traffic among the VLANs or between the Internet and a VLAN. You can define ACL based on CIDR, port range, protocol, type code (if ICMP protocol is selected) and Ingress/Egress type.

The following figure shows the possible deployment scenarios of a Inter-VLAN setup:



To set up a multi-tier Inter-VLAN deployment, see "*Configuring a Virtual Private Cloud*".

5.12.27 Configuring a Virtual Private Cloud

About Virtual Private Clouds

CloudStack Virtual Private Cloud is a private, isolated part of CloudStack. A VPC can have its own virtual network topology that resembles a traditional physical network. You can launch VMs in the virtual network that can have private addresses in the range of your choice, for example: 10.0.0.0/16. You can define network tiers within your VPC network range, which in turn enables you to group similar kinds of instances based on IP address range.

For example, if a VPC has the private range 10.0.0.0/16, its guest networks can have the network ranges 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24, and so on.

Major Components of a VPC

A VPC is comprised of the following network components:

- **VPC**: A VPC acts as a container for multiple isolated networks that can communicate with each other via its virtual router.
- **Network Tiers**: Each tier acts as an isolated network with its own VLANs and CIDR list, where you can place groups of resources, such as VMs. The tiers are segmented by means of VLANs. The NIC of each tier acts as its gateway.
- **Virtual Router**: A virtual router is automatically created and started when you create a VPC. The virtual router connect the tiers and direct traffic among the public gateway, the VPN gateways, and the NAT instances. For each tier, a corresponding NIC and IP exist in the virtual router. The virtual router provides DNS and DHCP services through its IP.

- **Public Gateway**: The traffic to and from the Internet routed to the VPC through the public gateway. In a VPC, the public gateway is not exposed to the end user; therefore, static routes are not support for the public gateway.
- **Private Gateway**: All the traffic to and from a private network routed to the VPC through the private gateway. For more information, see "*Adding a Private Gateway to a VPC*".
- **VPN Gateway**: The VPC side of a VPN connection.
- **Site-to-Site VPN Connection**: A hardware-based VPN connection between your VPC and your datacenter, home network, or co-location facility. For more information, see "*Setting Up a Site-to-Site VPN Connection*".
- **Customer Gateway**: The customer side of a VPN Connection. For more information, see *"Creating and Updating a VPN Customer Gateway"*.
- **NAT Instance**: An instance that provides Port Address Translation for instances to access the Internet via the public gateway. For more information, see "*Enabling or Disabling Static NAT on a VPC*".
- **Network ACL**: Network ACL is a group of Network ACL items. Network ACL items are nothing but numbered rules that are evaluated in order, starting with the lowest numbered rule. These rules determine whether traffic is allowed in or out of any tier associated with the network ACL. For more information, see "*Configuring Network Access Control List*".

Network Architecture in a VPC

In a VPC, the following four basic options of network architectures are present:

- VPC with a public gateway only
- VPC with public and private gateways
- VPC with public and private gateways and site-to-site VPN access
- VPC with a private gateway only and site-to-site VPN access

Connectivity Options for a VPC

You can connect your VPC to:

- The Internet through the public gateway.
- The corporate datacenter by using a site-to-site VPN connection through the VPN gateway.
- Both the Internet and your corporate datacenter by using both the public gateway and a VPN gateway.

VPC Network Considerations

Consider the following before you create a VPC:

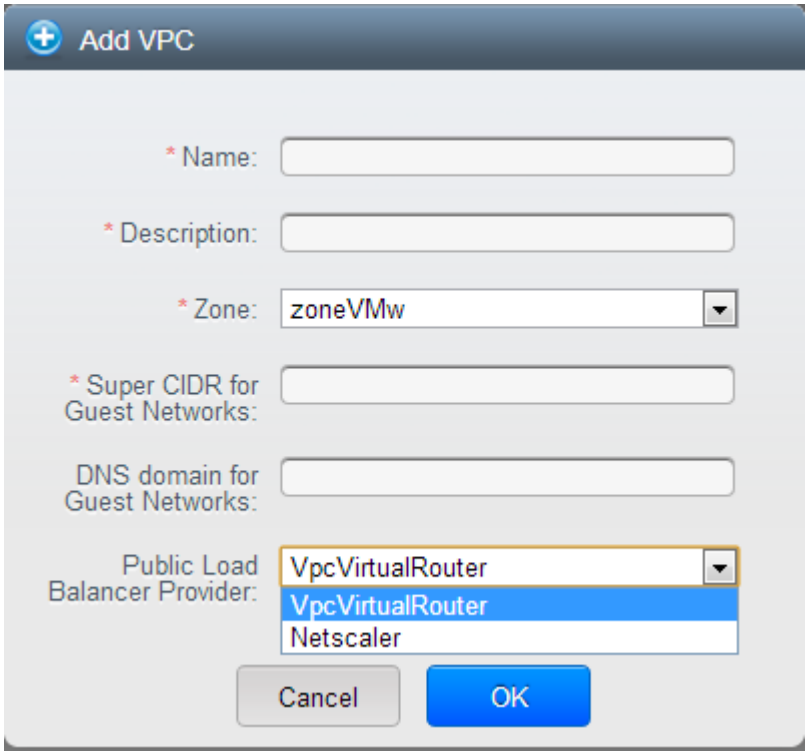
- A VPC, by default, is created in the enabled state.
- A VPC can be created in Advance zone only, and can't belong to more than one zone at a time.
- The default number of VPCs an account can create is 20. However, you can change it by using the max.account.vpcs global parameter, which controls the maximum number of VPCs an account is allowed to create.
- The default number of tiers an account can create within a VPC is 3. You can configure this number by using the vpc.max.networks parameter.

- Each tier should have an unique CIDR in the VPC. Ensure that the tier's CIDR should be within the VPC CIDR range.
- A tier belongs to only one VPC.
- All network tiers inside the VPC should belong to the same account.
- When a VPC is created, by default, a SourceNAT IP is allocated to it. The Source NAT IP is released only when the VPC is removed.
- A public IP can be used for only one purpose at a time. If the IP is a sourceNAT, it cannot be used for StaticNAT or port forwarding.
- The instances can only have a private IP address that you provision. To communicate with the Internet, enable NAT to an instance that you launch in your VPC.
- Only new networks can be added to a VPC. The maximum number of networks per VPC is limited by the value you specify in the vpc.max.networks parameter. The default value is three.
- The load balancing service can be supported by only one tier inside the VPC.
- If an IP address is assigned to a tier:
  - That IP can't be used by more than one tier at a time in the VPC. For example, if you have tiers A and B, and a public IP1, you can create a port forwarding rule by using the IP either for A or B, but not for both.
  - That IP can't be used for StaticNAT, load balancing, or port forwarding rules for another guest network inside the VPC.
- Remote access VPN is not supported in VPC networks.

Adding a Virtual Private Cloud

When creating the VPC, you simply provide the zone and a set of IP addresses for the VPC network address space. You specify this set of addresses in the form of a Classless Inter-Domain Routing (CIDR) block.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.
4. Click Add VPC. The Add VPC page is displayed as follows:



Provide the following information:

- **Name**: A short name for the VPC that you are creating.
- **Description**: A brief description of the VPC.
- **Zone**: Choose the zone where you want the VPC to be available.
- **Super CIDR for Guest Networks**: Defines the CIDR range for all the tiers (guest networks) within a VPC. When you create a tier, ensure that its CIDR is within the Super CIDR value you enter. The CIDR must be RFC1918 compliant.
- **DNS domain for Guest Networks**: If you want to assign a special domain name, specify the DNS suffix. This parameter is applied to all the tiers within the VPC. That implies, all the tiers you create in the VPC belong to the same DNS domain. If the parameter is not specified, a DNS domain name is generated automatically.
- **Public Load Balancer Provider**: You have two options: VPC Virtual Router and Netscaler.

5. Click OK.

Adding Tiers

Tiers are distinct locations within a VPC that act as isolated networks, which do not have access to other tiers by default. Tiers are set up on different VLANs that can communicate with each other by using a virtual router. Tiers provide inexpensive, low latency network connectivity to other tiers within the VPC.

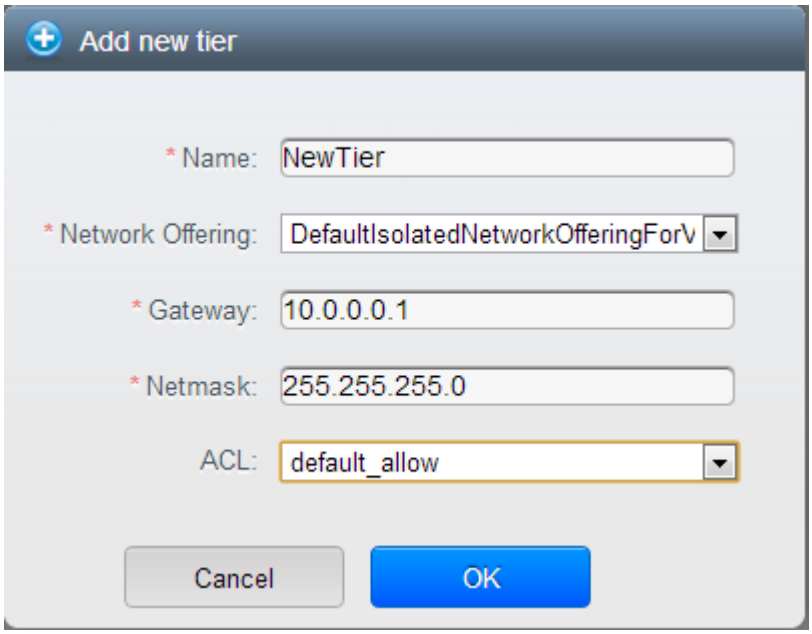
1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPC that you have created for the account is listed in the page.

Note: The end users can see their own VPCs, while root and domain admin can see any VPC they are authorized to see.

4. Click the Configure button of the VPC for which you want to set up tiers.
5. Click Create network.

   The Add new tier dialog is displayed, as follows:



   If you have already created tiers, the VPC diagram is displayed. Click Create Tier to add a new tier.

6. Specify the following:

   All the fields are mandatory.

   - **Name**: A unique name for the tier you create.
   - **Network Offering**: The following default network offerings are listed: Internal LB, DefaultIsolatedNetworkOfferingForVpcNetworksNoLB, DefaultIsolatedNetworkOfferingForVpcNetworks

     In a VPC, only one tier can be created by using LB-enabled network offering.
   - **Gateway**: The gateway for the tier you create. Ensure that the gateway is within the Super CIDR range that you specified while creating the VPC, and is not overlapped with the CIDR of any existing tier within the VPC.
   - **VLAN**: The VLAN ID for the tier that the root admin creates.

     This option is only visible if the network offering you selected is VLAN-enabled.

     For more information, see "Assigning VLANs to Isolated Networks".
   - **Netmask**: The netmask for the tier you create.

     For example, if the VPC CIDR is 10.0.0.0/16 and the network tier CIDR is 10.0.1.0/24, the gateway of the tier is 10.0.1.1, and the netmask of the tier is 255.255.255.0.

7. Click OK.
8. Continue with configuring access control list for the tier.

Configuring Network Access Control List

Define Network Access Control List (ACL) on the VPC virtual router to control incoming (ingress) and outgoing (egress) traffic between the VPC tiers, and the tiers and Internet. By default, all incoming traffic to the guest networks is blocked and all outgoing traffic from guest networks is allowed, once you add an ACL rule for outgoing traffic, then only outgoing traffic specified in this ACL rule is allowed, the rest is blocked. To open the ports, you must create a new network ACL. The network ACLs can be created for the tiers only if the NetworkACL service is supported.

About Network ACL Lists

In CloudStack terminology, Network ACL is a group of Network ACL items. Network ACL items are nothing but numbered rules that are evaluated in order, starting with the lowest numbered rule. These rules determine whether traffic is allowed in or out of any tier associated with the network ACL. You need to add the Network ACL items to the Network ACL, then associate the Network ACL with a tier. Network ACL is associated with a VPC and can be assigned to multiple VPC tiers within a VPC. A Tier is associated with a Network ACL at all the times. Each tier can be associated with only one ACL.

The default Network ACL is used when no ACL is associated. Default behavior is all the incoming traffic is blocked and outgoing traffic is allowed from the tiers. Default network ACL cannot be removed or modified. Contents of the default Network ACL is:

Rule	Protocol	Traffic type	Action	CIDR
1	All	Ingress	Deny	0.0.0.0/0
2	All	Egress	Deny	0.0.0.0/0

Creating ACL Lists

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC.

   For each tier, the following options are displayed:

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists

5. Select Network ACL Lists.

   The following default rules are displayed in the Network ACLs page: default_allow, default_deny.

6. Click Add ACL Lists, and specify the following:

   - **ACL List Name**: A name for the ACL list.
   - **Description**: A short description of the ACL list that can be displayed to users.

Creating an ACL Rule

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Click the Configure button of the VPC.
5. Select Network ACL Lists.

   In addition to the custom ACL lists you have created, the following default rules are displayed in the Network ACLs page: default_allow, default_deny.

6. Select the desired ACL list.
7. Select the ACL List Rules tab.

   To add an ACL rule, fill in the following fields to specify what kind of network traffic is allowed in the VPC.

   - **Rule Number**: The order in which the rules are evaluated.
   - **CIDR**: The CIDR acts as the Source CIDR for the Ingress rules, and Destination CIDR for the Egress rules. To accept traffic only from or to the IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. The CIDR is the base IP address of the incoming traffic. For example, 192.168.0.0/22. To allow all CIDRs, set to 0.0.0.0/0.
   - **Action**: What action to be taken. Allow traffic or block.
   - **Protocol**: The networking protocol that sources use to send traffic to the tier. The TCP and UDP protocols are typically used for data exchange and end-user communications. The ICMP protocol is typically used to send error messages or network monitoring data. All supports all the traffic. Other option is Protocol Number.
   - **Start Port**, **End Port** (TCP, UDP only): A range of listening ports that are the destination for the incoming traffic. If you are opening a single port, use the same number in both fields.
   - **Protocol Number**: The protocol number associated with IPv4 or IPv6. For more information, see Protocol Numbers.
   - **ICMP Type**, **ICMP Code** (ICMP only): The type of message and error code that will be sent.
   - **Traffic Type**: The type of traffic: Incoming or outgoing.

8. Click Add. The ACL rule is added.

   You can edit the tags assigned to the ACL rules and delete the ACL rules you have created. Click the appropriate button in the Details tab.

Creating a Tier with Custom ACL List

1. Create a VPC.
2. Create a custom ACL list.
3. Add ACL rules to the ACL list.
4. Create a tier in the VPC.

   Select the desired ACL list while creating a tier.
5. Click OK.

Assigning a Custom ACL List to a Tier

1. Create a VPC.
2. Create a tier in the VPC.
3. Associate the tier with the default ACL rule.
4. Create a custom ACL list.
5. Add ACL rules to the ACL list.
6. Select the tier for which you want to assign the custom ACL.
7. Click the Replace ACL List icon.

   The Replace ACL List dialog is displayed.
8. Select the desired ACL list.
9. Click OK.

Adding a Private Gateway to a VPC

A private gateway can be added by the root admin only. The VPC private network has 1:1 relationship with the NIC of the physical network. You can configure multiple private gateways to a single VPC. No gateways with duplicated VLAN and IP are allowed in the same data center.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to configure load balancing rules.

   The VPC page is displayed where all the tiers you created are listed in a diagram.
5. Click the Settings icon.

   The following options are displayed.

   - Internal LB
   - Public LB IP
   - Static NAT

- Virtual Machines
- CIDR

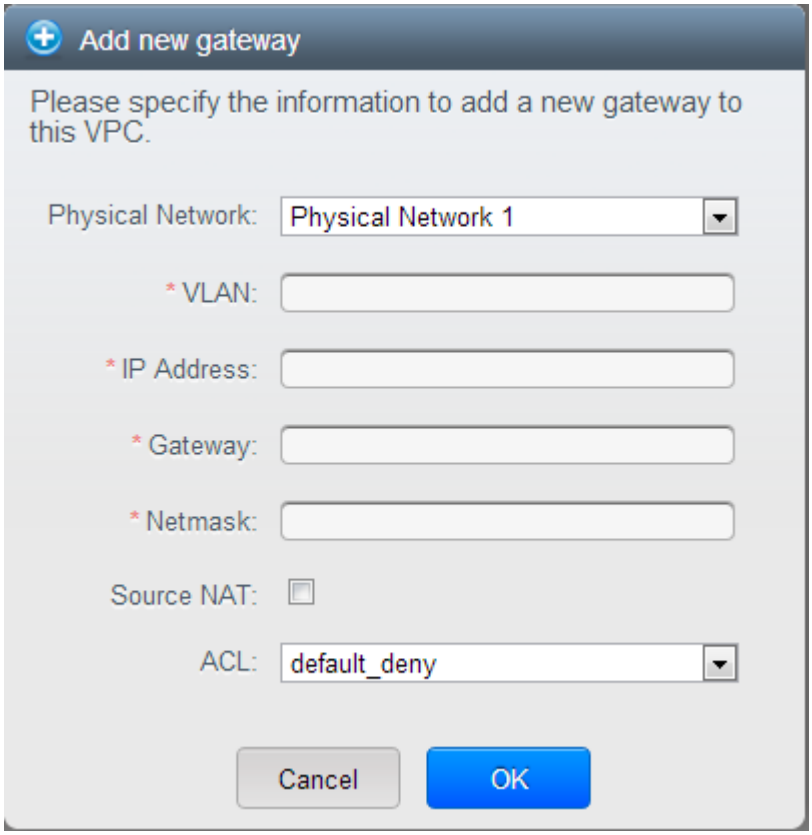
The following router information is displayed:

- Private Gateways
- Public IP Addresses
- Site-to-Site VPNs
- Network ACL Lists

6. Select Private Gateways.

   The Gateways page is displayed.

7. Click Add new gateway:



8. Specify the following:

   - **Physical Network**: The physical network you have created in the zone.
   - **IP Address**: The IP address associated with the VPC gateway.
   - **Gateway**: The gateway through which the traffic is routed to and from the VPC.
   - **Netmask**: The netmask associated with the VPC gateway.
   - **VLAN**: The VLAN associated with the VPC gateway.
   - **Source NAT**: Select this option to enable the source NAT service on the VPC private gateway.

     See "*Source NAT on Private Gateway*".

- **ACL**: Controls both ingress and egress traffic on a VPC private gateway. By default, all the traffic is blocked.

  See "*ACL on Private Gateway*".

The new gateway appears in the list. You can repeat these steps to add more gateway for this VPC.

Source NAT on Private Gateway

You might want to deploy multiple VPCs with the same super CIDR and guest tier CIDR. Therefore, multiple guest VMs from different VPCs can have the same IPs to reach a enterprise data center through the private gateway. In such cases, a NAT service need to be configured on the private gateway to avoid IP conflicts. If Source NAT is enabled, the guest VMs in VPC reaches the enterprise network via private gateway IP address by using the NAT service.

The Source NAT service on a private gateway can be enabled while adding the private gateway. On deletion of a private gateway, source NAT rules specific to the private gateway are deleted.

To enable source NAT on existing private gateways, delete them and create afresh with source NAT.

ACL on Private Gateway

The traffic on the VPC private gateway is controlled by creating both ingress and egress network ACL rules. The ACLs contains both allow and deny rules. As per the rule, all the ingress traffic to the private gateway interface and all the egress traffic out from the private gateway interface are blocked.

You can change this default behaviour while creating a private gateway. Alternatively, you can do the following:

1. In a VPC, identify the Private Gateway you want to work with.
2. In the Private Gateway page, do either of the following:
   - Use the Quickview. See 3.
   - Use the Details tab. See 4 through .
3. In the Quickview of the selected Private Gateway, click Replace ACL, select the ACL rule, then click OK
4. Click the IP address of the Private Gateway you want to work with.
5. In the Detail tab, click the Replace ACL button.

   The Replace ACL dialog is displayed.
6. select the ACL rule, then click OK.

   Wait for few seconds. You can see that the new ACL rule is displayed in the Details page.

Creating a Static Route

CloudStack enables you to specify routing for the VPN connection you create. You can enter one or CIDR addresses to indicate which traffic is to be routed back to the gateway.

1. In a VPC, identify the Private Gateway you want to work with.
2. In the Private Gateway page, click the IP address of the Private Gateway you want to work with.
3. Select the Static Routes tab.
4. Specify the CIDR of destination network.

5. Click Add.

   Wait for few seconds until the new route is created.

Blacklisting Routes

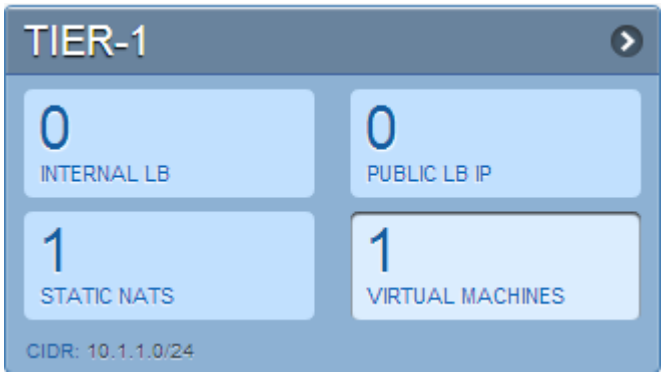
CloudStack enables you to block a list of routes so that they are not assigned to any of the VPC private gateways. Specify the list of routes that you want to blacklist in the `blacklisted.routes` global parameter. Note that the parameter update affects only new static route creations. If you block an existing static route, it remains intact and continue functioning. You cannot add a static route if the route is blacklisted for the zone.

Deploying VMs to the Tier

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you have created are listed.
5. Click Virtual Machines tab of the tier to which you want to add a VM.

   The Add Instance page is displayed.

   Follow the on-screen instruction to add an instance. For information on adding an instance, see the Installation Guide.

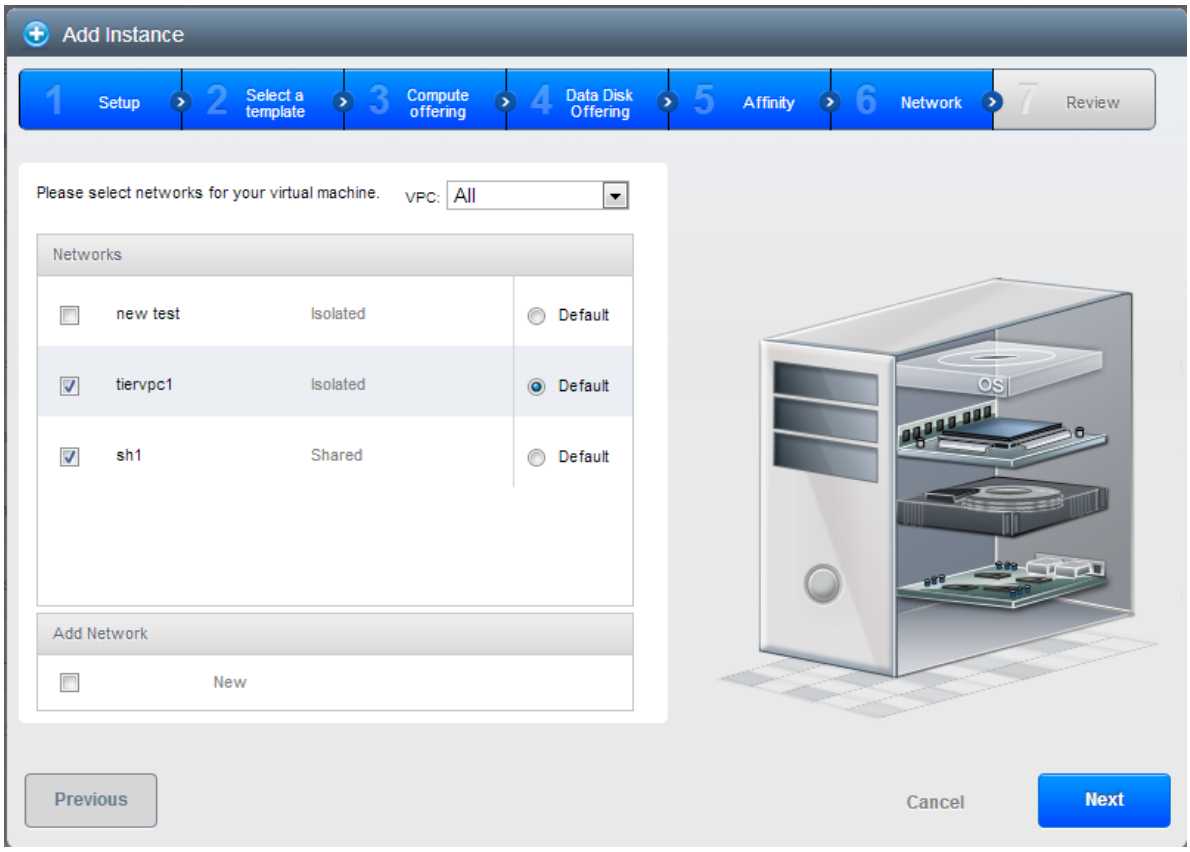
Deploying VMs to VPC Tier and Shared Networks

CloudStack allows you deploy VMs on a VPC tier and one or more shared networks. With this feature, VMs deployed in a multi-tier application can receive monitoring services via a shared network provided by a service provider.

1. Log in to the CloudStack UI as an administrator.
2. In the left navigation, choose Instances.
3. Click Add Instance.
4. Select a zone.
5. Select a template or ISO, then follow the steps in the wizard.

6. Ensure that the hardware you have allows starting the selected service offering.
7. Under Networks, select the desired networks for the VM you are launching.

   You can deploy a VM to a VPC tier and multiple shared networks.



8. Click Next, review the configuration and click Launch.

   Your VM will be deployed to the selected VPC tier and shared network.

Acquiring a New IP Address for a VPC

When you acquire an IP address, all IP addresses are allocated to VPC, not to the guest networks within the VPC. The IPs are associated to the guest network only when the first port-forwarding, load balancing, or Static NAT rule is created for the IP or the network. IP can't be associated to more than one network at a time.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.

   The following options are displayed.

   - Internal LB
   - Public LB IP

- Static NAT
- Virtual Machines
- CIDR

The following router information is displayed:

- Private Gateways
- Public IP Addresses
- Site-to-Site VPNs
- Network ACL Lists

5. Select IP Addresses.

   The Public IP Addresses page is displayed.

6. Click Acquire New IP, and click Yes in the confirmation dialog.

   You are prompted for confirmation because, typically, IP addresses are a limited resource. Within a few moments, the new IP address should appear with the state Allocated. You can now use the IP address in port forwarding, load balancing, and static NAT rules.

Releasing an IP Address Alloted to a VPC

The IP address is a limited resource. If you no longer need a particular IP, you can disassociate it from its VPC and return it to the pool of available addresses. An IP address can be released from its tier, only when all the networking (port forwarding, load balancing, or StaticNAT) rules are removed for this IP address. The released IP address will still belongs to the same VPC.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Click the Configure button of the VPC whose IP you want to release.

   The VPC page is displayed where all the tiers you created are listed in a diagram.

   The following options are displayed.

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists

5. Select Public IP Addresses.

   The IP Addresses page is displayed.

6. Click the IP you want to release.

7. In the Details tab, click the Release IP button

Enabling or Disabling Static NAT on a VPC

A static NAT rule maps a public IP address to the private IP address of a VM in a VPC to allow Internet traffic to it. This section tells how to enable or disable static NAT for a particular IP address in a VPC.

If port forwarding rules are already in effect for an IP address, you cannot enable static NAT to that IP.

If a guest VM is part of more than one network, static NAT rules will function only if they are defined on the default network.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.

   For each tier, the following options are displayed.

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

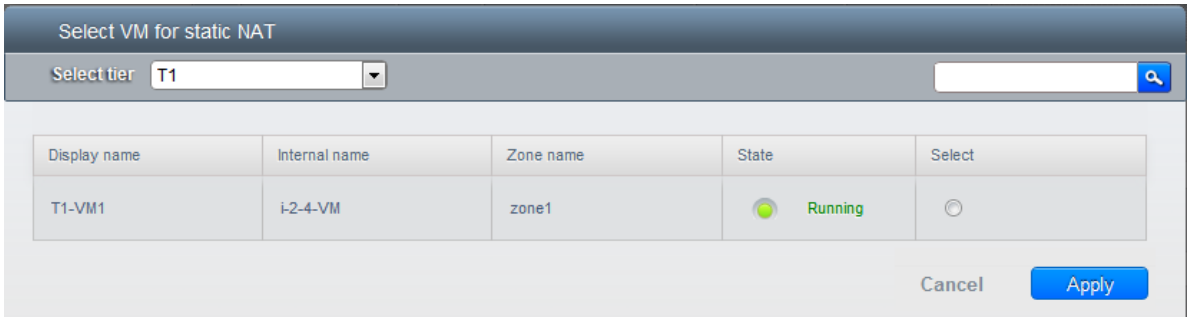
   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists

5. In the Router node, select Public IP Addresses.

   The IP Addresses page is displayed.

6. Click the IP you want to work with.

7. In the Details tab,click the Static NAT button. The button toggles between Enable and Disable, depending on whether static NAT is currently enabled for the IP address.
8. If you are enabling static NAT, a dialog appears as follows:



9. Select the tier and the destination VM, then click Apply.

Adding Load Balancing Rules on a VPC

In a VPC, you can configure two types of load balancing: external LB and internal LB. External LB is nothing but a LB rule created to redirect the traffic received at a public IP of the VPC virtual router. The traffic is load balanced within a tier based on your configuration. Citrix NetScaler and VPC virtual router are supported for external LB. When you use internal LB service, traffic received at a tier is load balanced across different VMs within that tier. For example, traffic reached at Web tier is redirected to another VM in that tier. External load balancing devices are not supported for internal LB. The service is provided by a internal LB VM configured on the target tier.

Load Balancing Within a Tier (External LB)

A CloudStack user or administrator may create load balancing rules that balance traffic received at a public IP to one or more VMs that belong to a network tier that provides load balancing service in a VPC. A user creates a rule, specifies an algorithm, and assigns the rule to a set of VMs within a tier.

Enabling NetScaler as the LB Provider on a VPC Tier

1. Add and enable Netscaler VPX in dedicated mode.

   Netscaler can be used in a VPC environment only if it is in dedicated mode.
2. Create a network offering, as given in "*Creating a Network Offering for External LB*".
3. Create a VPC with Netscaler as the Public LB provider.

   For more information, see *"Adding a Virtual Private Cloud"*.
4. For the VPC, acquire an IP.
5. Create an external load balancing rule and apply, as given in *Creating an External LB Rule*.

Creating a Network Offering for External LB

To have external LB support on VPC, create a network offering as follows:

1. Log in to the CloudStack UI as a user or admin.
2. From the Select Offering drop-down, choose Network Offering.
3. Click Add Network Offering.
4. In the dialog, make the following choices:

- **Name**: Any desired name for the network offering.
- **Description**: A short description of the offering that can be displayed to users.
- **Network Rate**: Allowed data transfer rate in MB per second.
- **Traffic Type**: The type of network traffic that will be carried on the network.
- **Guest Type**: Choose whether the guest network is isolated or shared.
- **Persistent**: Indicate whether the guest network is persistent or not. The network that you can provision without having to deploy a VM on it is termed persistent network.
- **VPC**: This option indicate whether the guest network is Virtual Private Cloud-enabled. A Virtual Private Cloud (VPC) is a private, isolated part of CloudStack. A VPC can have its own virtual network topology that resembles a traditional physical network. For more information on VPCs, see :ref: *about-vpc*.
- **Specify VLAN**: (Isolated guest networks only) Indicate whether a VLAN should be specified when this offering is used.
- **Supported Services**: Select Load Balancer. Use Netscaler or VpcVirtualRouter.
- **Load Balancer Type**: Select Public LB from the drop-down.
- **LB Isolation**: Select Dedicated if Netscaler is used as the external LB provider.
- **System Offering**: Choose the system service offering that you want virtual routers to use in this network.
- **Conserve mode**: Indicate whether to use conserve mode. In this mode, network resources are allocated only when the first virtual machine starts in the network.

5. Click OK and the network offering is created.

Creating an External LB Rule

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC, for which you want to configure load balancing rules.

   The VPC page is displayed where all the tiers you created listed in a diagram.

   For each tier, the following options are displayed:

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs

- Network ACL Lists

5. In the Router node, select Public IP Addresses.

   The IP Addresses page is displayed.

6. Click the IP address for which you want to create the rule, then click the Configuration tab.
7. In the Load Balancing node of the diagram, click View All.
8. Select the tier to which you want to apply the rule.
9. Specify the following:
   - **Name**: A name for the load balancer rule.
   - **Public Port**: The port that receives the incoming traffic to be balanced.
   - **Private Port**: The port that the VMs will use to receive the traffic.
   - **Algorithm**. Choose the load balancing algorithm you want CloudStack to use. CloudStack supports the following well-known algorithms:
     - Round-robin
     - Least connections
     - Source
   - **Stickiness**. (Optional) Click Configure and choose the algorithm for the stickiness policy. See Sticky Session Policies for Load Balancer Rules.
   - **Add VMs**: Click Add VMs, then select two or more VMs that will divide the load of incoming traffic, and click Apply.

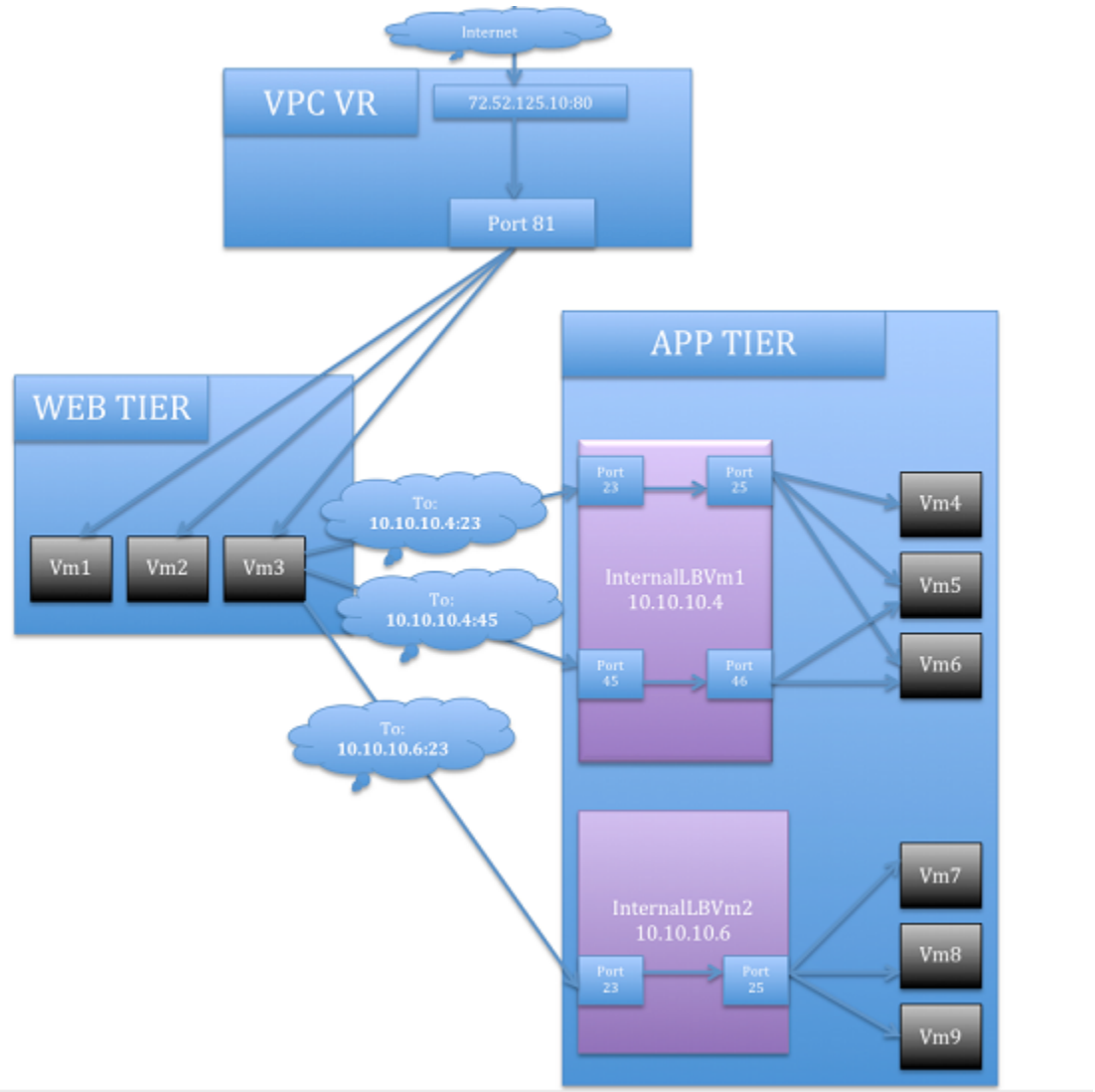
The new load balancing rule appears in the list. You can repeat these steps to add more load balancing rules for this IP address.

Load Balancing Across Tiers

CloudStack supports sharing workload across different tiers within your VPC. Assume that multiple tiers are set up in your environment, such as Web tier and Application tier. Traffic to each tier is balanced on the VPC virtual router on the public side, as explained in *"Adding Load Balancing Rules on a VPC"*. If you want the traffic coming from the Web tier to the Application tier to be balanced, use the internal load balancing feature offered by CloudStack.

How Does Internal LB Work in VPC?

In this figure, a public LB rule is created for the public IP 72.52.125.10 with public port 80 and private port 81. The LB rule, created on the VPC virtual router, is applied on the traffic coming from the Internet to the VMs on the Web tier. On the Application tier two internal load balancing rules are created. An internal LB rule for the guest IP 10.10.10.4 with load balancer port 23 and instance port 25 is configured on the VM, InternalLBVM1. Another internal LB rule for the guest IP 10.10.10.4 with load balancer port 45 and instance port 46 is configured on the VM, InternalLBVM1. Another internal LB rule for the guest IP 10.10.10.6, with load balancer port 23 and instance port 25 is configured on the VM, InternalLBVM2.



Guidelines

- Internal LB and Public LB are mutually exclusive on a tier. If the tier has LB on the public side, then it can't have the Internal LB.
- Internal LB is supported just on VPC networks in CloudStack 4.2 release.
- Only Internal LB VM can act as the Internal LB provider in CloudStack 4.2 release.
- Network upgrade is not supported from the network offering with Internal LB to the network offering with Public LB.
- Multiple tiers can have internal LB support in a VPC.
- Only one tier can have Public LB support in a VPC.

Enabling Internal LB on a VPC Tier

1. Create a network offering, as given in *Creating a Network Offering for Internal LB*.

2. Create an internal load balancing rule and apply, as given in *Creating an Internal LB Rule*.

Creating a Network Offering for Internal LB

To have internal LB support on VPC, either use the default offering, DefaultIsolatedNetworkOfferingForVpcNetworksWithInternalLB, or create a network offering as follows:

1. Log in to the CloudStack UI as a user or admin.
2. From the Select Offering drop-down, choose Network Offering.
3. Click Add Network Offering.
4. In the dialog, make the following choices:
   - **Name**: Any desired name for the network offering.
   - **Description**: A short description of the offering that can be displayed to users.
   - **Network Rate**: Allowed data transfer rate in MB per second.
   - **Traffic Type**: The type of network traffic that will be carried on the network.
   - **Guest Type**: Choose whether the guest network is isolated or shared.
   - **Persistent**: Indicate whether the guest network is persistent or not. The network that you can provision without having to deploy a VM on it is termed persistent network.
   - **VPC**: This option indicate whether the guest network is Virtual Private Cloud-enabled. A Virtual Private Cloud (VPC) is a private, isolated part of CloudStack. A VPC can have its own virtual network topology that resembles a traditional physical network. For more information on VPCs, see *"About Virtual Private Clouds"*.
   - **Specify VLAN**: (Isolated guest networks only) Indicate whether a VLAN should be specified when this offering is used.
   - **Supported Services**: Select Load Balancer. Select `InternalLbVM` from the provider list.
   - **Load Balancer Type**: Select Internal LB from the drop-down.
   - **System Offering**: Choose the system service offering that you want virtual routers to use in this network.
   - **Conserve mode**: Indicate whether to use conserve mode. In this mode, network resources are allocated only when the first virtual machine starts in the network.
5. Click OK and the network offering is created.

Creating an Internal LB Rule

When you create the Internal LB rule and applies to a VM, an Internal LB VM, which is responsible for load balancing, is created.

You can view the created Internal LB VM in the Instances page if you navigate to **Infrastructure > Zones >** <zone_name> > <physical_network_name> > **Network Service Providers > Internal LB VM**. You can manage the Internal LB VMs as and when required from the location.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Locate the VPC for which you want to configure internal LB, then click Configure.

   The VPC page is displayed where all the tiers you created listed in a diagram.

5. Locate the Tier for which you want to configure an internal LB rule, click Internal LB.

   In the Internal LB page, click Add Internal LB.

6. In the dialog, specify the following:

   - **Name**: A name for the load balancer rule.
   - **Description**: A short description of the rule that can be displayed to users.
   - **Source IP Address**: (Optional) The source IP from which traffic originates. The IP is acquired from the CIDR of that particular tier on which you want to create the Internal LB rule. If not specified, the IP address is automatically allocated from the network CIDR.

     For every Source IP, a new Internal LB VM is created for load balancing.
   - **Source Port**: The port associated with the source IP. Traffic on this port is load balanced.
   - **Instance Port**: The port of the internal LB VM.
   - **Algorithm**. Choose the load balancing algorithm you want CloudStack to use. CloudStack supports the following well-known algorithms:
     - Round-robin
     - Least connections
     - Source

Adding a Port Forwarding Rule on a VPC

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.

   For each tier, the following options are displayed:

   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   The following router information is displayed:

   - Private Gateways
   - Public IP Addresses
   - Site-to-Site VPNs
   - Network ACL Lists

5. In the Router node, select Public IP Addresses.

   The IP Addresses page is displayed.

6. Click the IP address for which you want to create the rule, then click the Configuration tab.
7. In the Port Forwarding node of the diagram, click View All.
8. Select the tier to which you want to apply the rule.
9. Specify the following:
   - **Public Port**: The port to which public traffic will be addressed on the IP address you acquired in the previous step.
   - **Private Port**: The port on which the instance is listening for forwarded public traffic.
   - **Protocol**: The communication protocol in use between the two ports.
     - TCP
     - UDP
   - **Add VM**: Click Add VM. Select the name of the instance to which this rule applies, and click Apply.

     You can test the rule by opening an SSH session to the instance.

Removing Tiers

You can remove a tier from a VPC. A removed tier cannot be revoked. When a tier is removed, only the resources of the tier are expunged. All the network rules (port forwarding, load balancing and staticNAT) and the IP addresses associated to the tier are removed. The IP address still be belonging to the same VPC.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPC that you have created for the account is listed in the page.

4. Click the Configure button of the VPC for which you want to set up tiers.

   The Configure VPC page is displayed. Locate the tier you want to work with.

5. Select the tier you want to remove.
6. In the Network Details tab, click the Delete Network button.

   Click Yes to confirm. Wait for some time for the tier to be removed.

Editing, Restarting, and Removing a Virtual Private Cloud

Note: Ensure that all the tiers are removed before you remove a VPC.

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Select the VPC you want to work with.

5. In the Details tab, click the Remove VPC button

   You can remove the VPC by also using the remove button in the Quick View.

   You can edit the name and description of a VPC. To do that, select the VPC, then click the Edit button.

   To restart a VPC, select the VPC, then click the Restart button.

5.12.28 Persistent Networks

The network that you can provision without having to deploy any VMs on it is called a persistent network. A persistent network can be part of a VPC or a non-VPC environment.

When you create other types of network, a network is only a database entry until the first VM is created on that network. When the first VM is created, a VLAN ID is assigned and the network is provisioned. Also, when the last VM is destroyed, the VLAN ID is released and the network is no longer available. With the addition of persistent network, you will have the ability to create a network in CloudStack in which physical devices can be deployed without having to run any VMs. Additionally, you can deploy physical devices on that network.

One of the advantages of having a persistent network is that you can create a VPC with a tier consisting of only physical devices. For example, you might create a VPC for a three-tier application, deploy VMs for Web and Application tier, and use physical machines for the Database tier. Another use case is that if you are providing services by using physical hardware, you can define the network as persistent and therefore even if all its VMs are destroyed the services will not be discontinued.

Persistent Network Considerations

- Persistent network is designed for isolated networks.
- All default network offerings are non-persistent.
- A network offering cannot be editable because changing it affects the behavior of the existing networks that were created using this network offering.
- When you create a guest network, the network offering that you select defines the network persistence. This in turn depends on whether persistent network is enabled in the selected network offering.
- An existing network can be made persistent by changing its network offering to an offering that has the Persistent option enabled. While setting this property, even if the network has no running VMs, the network is provisioned.
- An existing network can be made non-persistent by changing its network offering to an offering that has the Persistent option disabled. If the network has no running VMs, during the next network garbage collection run the network is shut down.
- When the last VM on a network is destroyed, the network garbage collector checks if the network offering associated with the network is persistent, and shuts down the network only if it is non-persistent.

Creating a Persistent Guest Network

To create a persistent network, perform the following:

1. Create a network offering with the Persistent option enabled.

   See "Creating a New Network Offering".

2. Select Network from the left navigation pane.
3. Select the guest network that you want to offer this network service to.
4. Click the Edit button.
5. From the Network Offering drop-down, select the persistent network offering you have just created.
6. Click OK.

5.12.29 Setup a Palo Alto Networks Firewall

Functionality Provided

This implementation enables the orchestration of a Palo Alto Networks Firewall from within CloudStack UI and API.

The following features are supported:

- List/Add/Delete Palo Alto Networks service provider
- List/Add/Delete Palo Alto Networks network service offering
- List/Add/Delete Palo Alto Networks network using the above service offering
- Add an instance to a Palo Alto Networks network
- Source NAT management on network create and delete
- List/Add/Delete Ingress Firewall rule
- List/Add/Delete Egress Firewall rule (both 'Allow' and 'Deny' default rules supported)
- List/Add/Delete Port Forwarding rule
- List/Add/Delete Static NAT rule
- Apply a Threat Profile to all firewall rules (more details in the Additional Features section)
- Apply a Log Forwarding profile to all firewall rules (more details in the Additional Features section)

Initial Palo Alto Networks Firewall Configuration

Anatomy of the Palo Alto Networks Firewall

- In **'Network > Interfaces'** there is a list of physical interfaces as well as aggregated physical interfaces which are used for managing traffic in and out of the Palo Alto Networks Firewall device.
- In **'Network > Zones'** there is a list of the different configuration zones. This implementation will use two zones; a public (defaults to 'untrust') and private (defaults to 'trust') zone.
- In **'Network > Virtual Routers'** there is a list of VRs which handle traffic routing for the Palo Alto Firewall. We only use a single Virtual Router on the firewall and it is used to handle all the routing to the next network hop.
- In **'Objects > Security Profile Groups'** there is a list of profiles which can be applied to firewall rules. These profiles are used to better understand the types of traffic that is flowing through your network. Configured when you add the firewall provider to CloudStack.
- In **'Objects > Log Forwarding'** there is a list of profiles which can be applied to firewall rules. These profiles are used to better track the logs generated by the firewall. Configured when you add the firewall provider to CloudStack.

- In **'Policies > Security'** there is a list of firewall rules that are currently configured. You will not need to modify this section because it will be completely automated by CloudStack, but you can review the firewall rules which have been created here.
- In **'Policies > NAT'** there is a list of the different NAT rules. You will not need to modify this section because it will be completely automated by CloudStack, but you can review the different NAT rules that have been created here. Source NAT, Static NAT and Destination NAT (Port Forwarding) rules will show up in this list.

Configure the Public / Private Zones on the firewall

No manual configuration is required to setup these zones because CloudStack will configure them automatically when you add the Palo Alto Networks firewall device to CloudStack as a service provider. This implementation depends on two zones, one for the public side and one for the private side of the firewall.

- The public zone (defaults to 'untrust') will contain all of the public interfaces and public IPs.
- The private zone (defaults to 'trust') will contain all of the private interfaces and guest network gateways.

The NAT and firewall rules will be configured between these zones.

Configure the Public / Private Interfaces on the firewall

This implementation supports standard physical interfaces as well as grouped physical interfaces called aggregated interfaces. Both standard interfaces and aggregated interfaces are treated the same, so they can be used interchangeably. For this document, we will assume that we are using 'ethernet1/1' as the public interface and 'ethernet1/2' as the private interface. If aggregated interfaces where used, you would use something like 'ae1' and 'ae2' as the interfaces.

This implementation requires that the 'Interface Type' be set to 'Layer3' for both the public and private interfaces. If you want to be able to use the 'Untagged' VLAN tag for public traffic in CloudStack, you will need to enable support for it in the public 'ethernet1/1' interface (details below).

Steps to configure the Public Interface:

1. Log into Palo Alto Networks Firewall
2. Navigate to 'Network > Interfaces'
3. Click on 'ethernet1/1' (for aggregated ethernet, it will probably be called 'ae1')
4. Select 'Layer3' from the 'Interface Type' list
5. Click 'Advanced'
6. Check the 'Untagged Subinterface' check-box
7. Click 'OK'

Steps to configure the Private Interface:

1. Click on 'ethernet1/2' (for aggregated ethernet, it will probably be called 'ae2')
2. Select 'Layer3' from the 'Interface Type' list
3. Click 'OK'

Configure a Virtual Router on the firewall

The Virtual Router on the Palo Alto Networks Firewall is not to be confused with the Virtual Routers that CloudStack provisions. For this implementation, the Virtual Router on the Palo Alto Networks Firewall will ONLY handle the upstream routing from the Firewall to the next hop.

Steps to configure the Virtual Router:

1. Log into Palo Alto Networks Firewall
2. Navigate to 'Network > Virtual Routers'
3. Select the 'default' Virtual Router or Add a new Virtual Router if there are none in the list
   - If you added a new Virtual Router, you will need to give it a 'Name'
4. Navigate to 'Static Routes > IPv4'
5. 'Add' a new static route
   - **Name**: next_hop (you can name it anything you want)
   - **Destination**: 0.0.0.0/0 (send all traffic to this route)
   - **Interface**: ethernet1/1 (or whatever you set your public interface as)
   - **Next Hop**: (specify the gateway IP for the next hop in your network)
   - Click 'OK'
6. Click 'OK'

Configure the default Public Subinterface

The current implementation of the Palo Alto Networks firewall integration uses CIDRs in the form of 'w.x.y.z/32' for the public IP addresses that CloudStack provisions. Because no broadcast or gateway IPs are in this single IP range, there is no way for the firewall to route the traffic for these IPs. To route the traffic for these IPs, we create a single subinterface on the public interface with an IP and a CIDR which encapsulates the CloudStack public IP range. This IP will need to be inside the subnet defined by the CloudStack public range netmask, but outside the CloudStack public IP range. The CIDR should reflect the same subnet defined by the CloudStack public range netmask. The name of the subinterface is determined by the VLAN configured for the public range in CloudStack.

To clarify this concept, we will use the following example.

Example CloudStack Public Range Configuration:

- **Gateway**: 172.30.0.1
- **Netmask**: 255.255.255.0
- **IP Range**: 172.30.0.100 - 172.30.0.199
- **VLAN**: Untagged

Configure the Public Subinterface:

1. Log into Palo Alto Networks Firewall
2. Navigate to 'Network > Interfaces'
3. Select the 'ethernet1/1' line (not clicking on the name)
4. Click 'Add Subinterface' at the bottom of the window
5. Enter 'Interface Name': 'ethernet1/1' . '9999'
   - 9999 is used if the CloudStack public range VLAN is 'Untagged'
   - If the CloudStack public range VLAN is tagged (eg: 333), then the name will reflect that tag

6. The 'Tag' is the VLAN tag that the traffic is sent to the next hop with, so set it accordingly. If you are passing 'Untagged' traffic from CloudStack to your next hop, leave it blank. If you want to pass tagged traffic from CloudStack, specify the tag.
7. Select 'default' from the 'Config > Virtual Router' drop-down (assuming that is what your virtual router is called)
8. Click the 'IPv4' tab
9. Select 'Static' from the 'Type' radio options
10. Click 'Add' in the 'IP' section
11. Enter '172.30.0.254/24' in the new line
    - The IP can be any IP outside the CloudStack public IP range, but inside the CloudStack public range netmask (it can NOT be the gateway IP)
    - The subnet defined by the CIDR should match the CloudStack public range netmask
12. Click 'OK'

Commit configuration on the Palo Alto Networks Firewall

In order for all the changes we just made to take effect, we need to commit the changes.

1. Click the 'Commit' link in the top right corner of the window
2. Click 'OK' in the commit window overlay
3. Click 'Close' to the resulting commit status window after the commit finishes

Setup the Palo Alto Networks Firewall in CloudStack

Add the Palo Alto Networks Firewall as a Service Provider

1. Navigate to 'Infrastructure > Zones > ZONE_NAME > Physical Network > NETWORK_NAME (guest) > Configure; Network Service Providers'
2. Click on 'Palo Alto' in the list
3. Click 'View Devices'
4. Click 'Add Palo Alto Device'
5. Enter your configuration in the overlay. This example will reflect the details previously used in this guide.
    - **IP Address**: (the IP of the Palo Alto Networks Firewall)
    - **Username**: (the admin username for the firewall)
    - **Password**: (the admin password for the firewall)
    - **Type**: Palo Alto Firewall
    - **Public Interface**: ethernet1/1 (use what you setup earlier as the public interface if it is different from my examples)
    - **Private Interface**: ethernet1/2 (use what you setup earlier as the private interface if it is different from my examples)
    - **Number of Retries**: 2 (the default is fine)

- **Timeout**: 300 (the default is fine)
- **Public Network**: untrust (this is the public zone on the firewall and did not need to be configured)
- **Private Network**: trust (this is the private zone on the firewall and did not need to be configured)
- **Virtual Router**: default (this is the name of the Virtual Router we setup on the firewall)
- **Palo Alto Threat Profile**: (not required. name of the 'Security Profile Groups' to apply. more details in the 'Additional Features' section)
- **Palo Alto Log Profile**: (not required. name of the 'Log Forwarding' profile to apply. more details in the 'Additional Features' section)
- **Capacity**: (not required)
- **Dedicated**: (not required)

6. Click 'OK'
7. Click on 'Palo Alto' in the breadcrumbs to go back one screen.
8. Click on 'Enable Provider'

Add a Network Service Offering to use the new Provider

There are 6 'Supported Services' that need to be configured in the network service offering for this functionality. They are DHCP, DNS, Firewall, Source NAT, Static NAT and Port Forwarding. For the other settings, there are probably additional configurations which will work, but I will just document a common case.

1. Navigate to 'Service Offerings'
2. In the drop-down at the top, select 'Network Offerings'
3. Click 'Add Network Offering'
   - **Name**: (name it whatever you want)
   - **Description**: (again, can be whatever you want)
   - **Guest Type**: Isolated
   - **Supported Services**:
     - **DHCP**: Provided by 'VirtualRouter'
     - **DNS**: Provided by 'VirtualRouter'
     - **Firewall**: Provided by 'PaloAlto'
     - **Source NAT**: Provided by 'PaloAlto'
     - **Static NAT**: Provided by 'PaloAlto'
     - **Port Forwarding**: Provided by 'PaloAlto'
   - **System Offering for Router**: System Offering For Software Router
   - **Supported Source NAT Type**: Per account (this is the only supported option)
   - **Default egress policy**: (both 'Allow' and 'Deny' are supported)
4. Click 'OK'
5. Click on the newly created service offering

6. Click 'Enable network offering'

When adding networks in CloudStack, select this network offering to use the Palo Alto Networks firewall.

Additional Features

In addition to the standard functionality exposed by CloudStack, we have added a couple additional features to this implementation. We did not add any new screens to CloudStack, but we have added a couple fields to the 'Add Palo Alto Service Provider' screen which will add functionality globally for the device.

Palo Alto Networks Threat Profile

This feature allows you to specify a 'Security Profile Group' to be applied to all of the firewall rules which are created on the Palo Alto Networks firewall device.

To create a 'Security Profile Group' on the Palo Alto Networks firewall, do the following:

1. Log into the Palo Alto Networks firewall
2. Navigate to 'Objects > Security Profile Groups'
3. Click 'Add' at the bottom of the page to add a new group
4. Give the group a Name and specify the profiles you would like to include in the group
5. Click 'OK'
6. Click the 'Commit' link in the top right of the screen and follow the on screen instructions

Once you have created a profile, you can reference it by Name in the 'Palo Alto Threat Profile' field in the 'Add the Palo Alto Networks Firewall as a Service Provider' step.

Palo Alto Networks Log Forwarding Profile

This feature allows you to specify a 'Log Forwarding' profile to better manage where the firewall logs are sent to. This is helpful for keeping track of issues that can arise on the firewall.

To create a 'Log Forwarding' profile on the Palo Alto Networks Firewall, do the following:

1. Log into the Palo Alto Networks firewall
2. Navigate to 'Objects > Log Forwarding'
3. Click 'Add' at the bottom of the page to add a new profile
4. Give the profile a Name and specify the details you want for the traffic and threat settings
5. Click 'OK'
6. Click the 'Commit' link in the top right of the screen and follow the on screen instructions

Once you have created a profile, you can reference it by Name in the 'Palo Alto Log Profile' field in the 'Add the Palo Alto Networks Firewall as a Service Provider' step.

Limitations

- The implementation currently only supports a single public IP range in CloudStack
- Usage tracking is not yet implemented

5.12.30 Using Remote Access VPN

Clients

Per Operating System instructions

- *Mac OSX*
- *Microsoft Windows 8*

Remote Access VPN connection to VPC or Guest Network to access Instances and applications. This section consider you have enable Remonte acccess VPN, refer to: *Remote Access VPN*.

When connected to a VPC via VPN, the client have access to all Tiers.

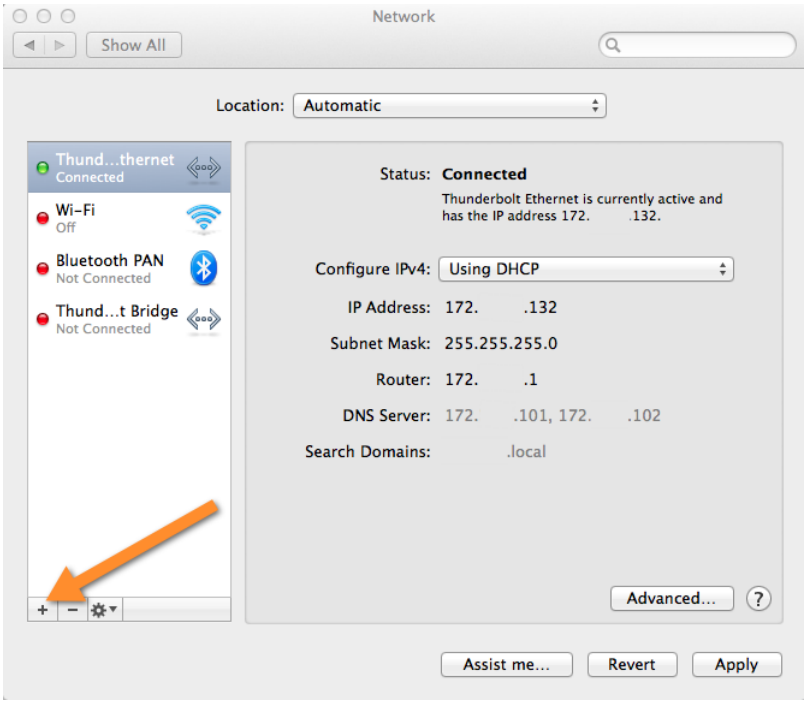
Following information is required to confiture VPN client:

- `Public IP`: source NAT with VPN enabled.
- `IPsec pre-shared key`: Provide at the VPN activation.
- `Username` VPN account username.
- `Password` VPN account password.

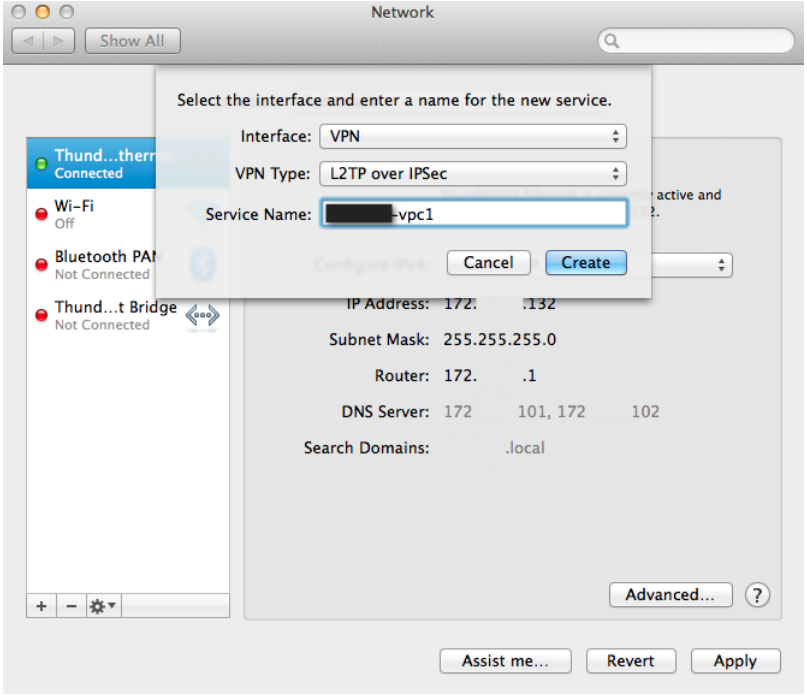
Mac OSX

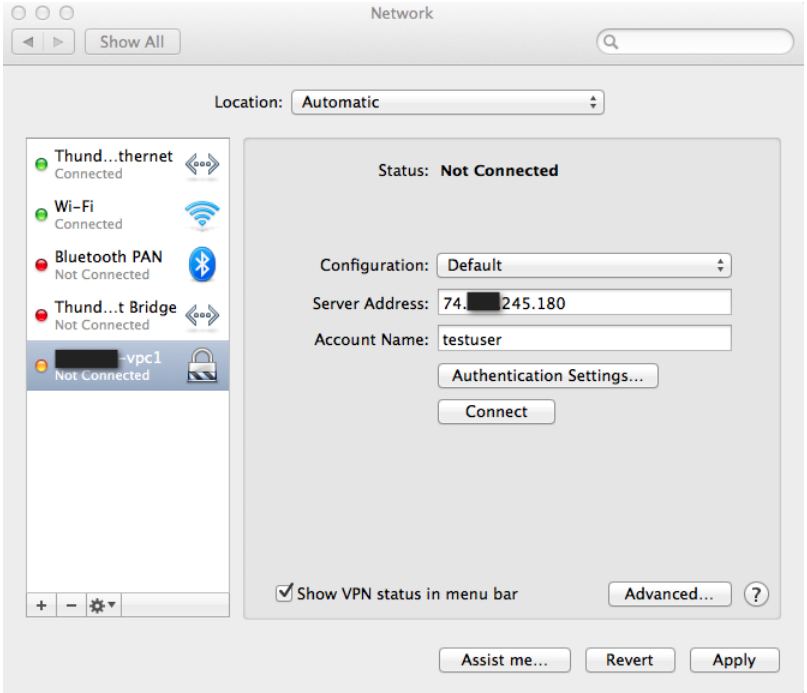
Mac OSX provide native IPsec VPN client.

1. Into System Preferences -> Network
2. Click "+" button and add a VPN:
    - Interface: VPN
    - VPN Type: L2TP over IPSec
    - Service Name: (ex: test-vpc1)

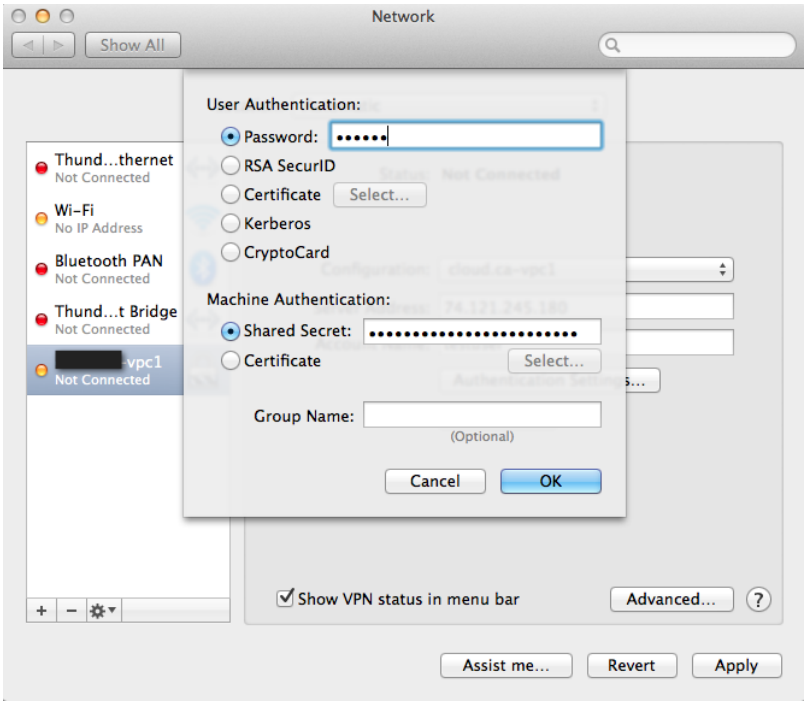


3. Configure L2TP over IPsec

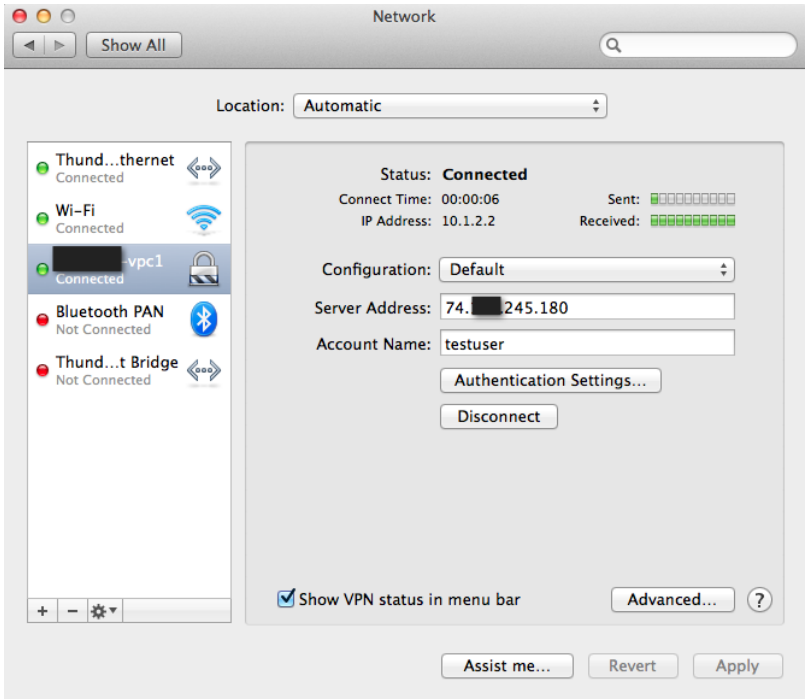




4. Inside Authentication Settings. . .



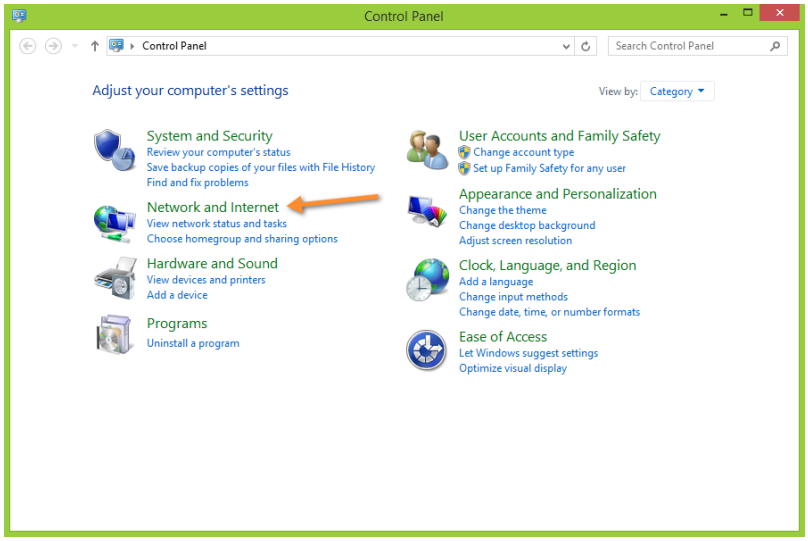
5. Connect into VPN
   (a) Click Apply to apply Network configuration changes.
   (b) Click Connect to initiate VPN connection.

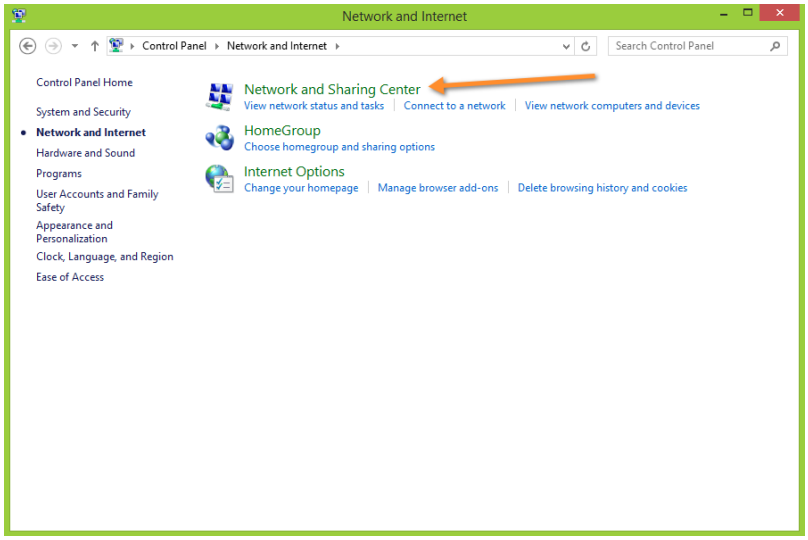


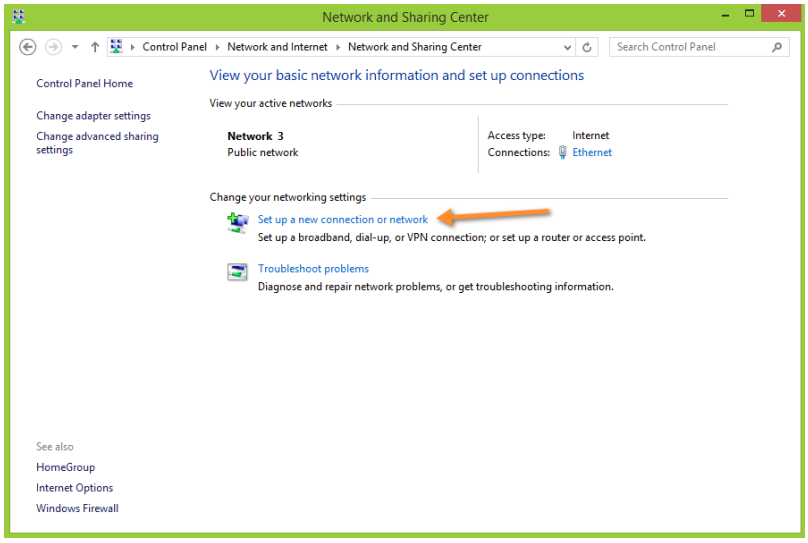
Microsoft Windows 8

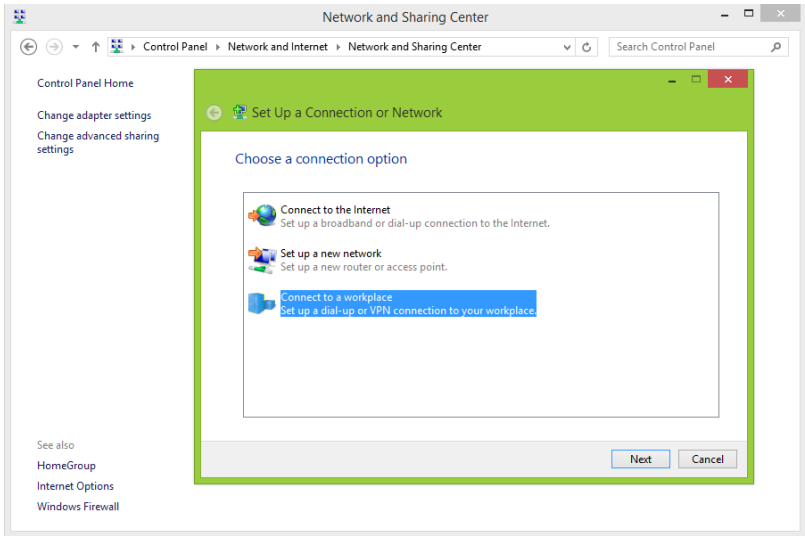
Following instruction have been perform using Windows 8.1 using Native VPN client.

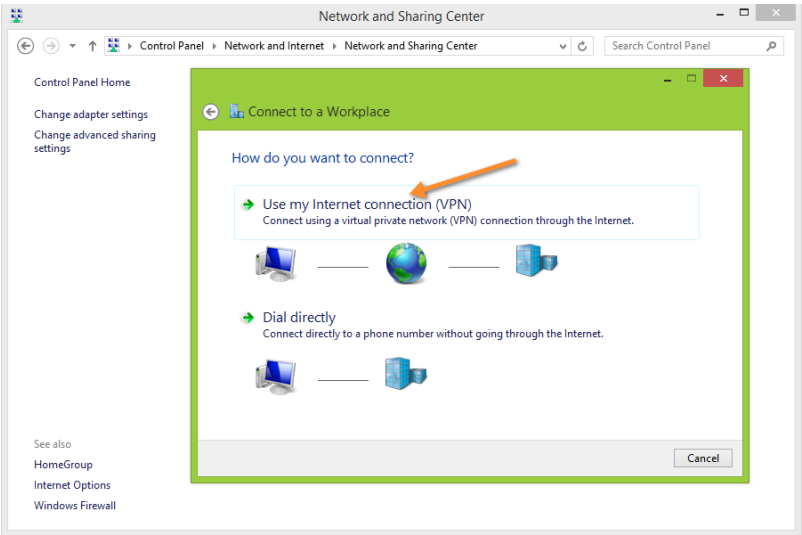
1. Create network VPN connection

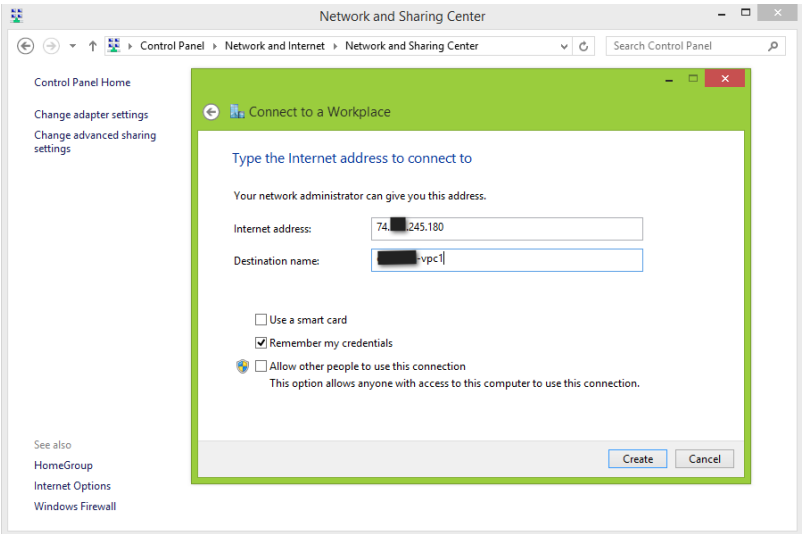



Network and Internet
Control Panel ▸ Network and Internet ▸
Search Control Panel
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts and Family Safety
Appearance and Personalization
Clock, Language, and Region
Ease of Access
Network and Sharing Center
View network status and tasks | Connect to a network | View network computers and devices
HomeGroup
Choose homegroup and sharing options
Internet Options
Change your homepage | Manage browser add-ons | Delete browsing history and cookies

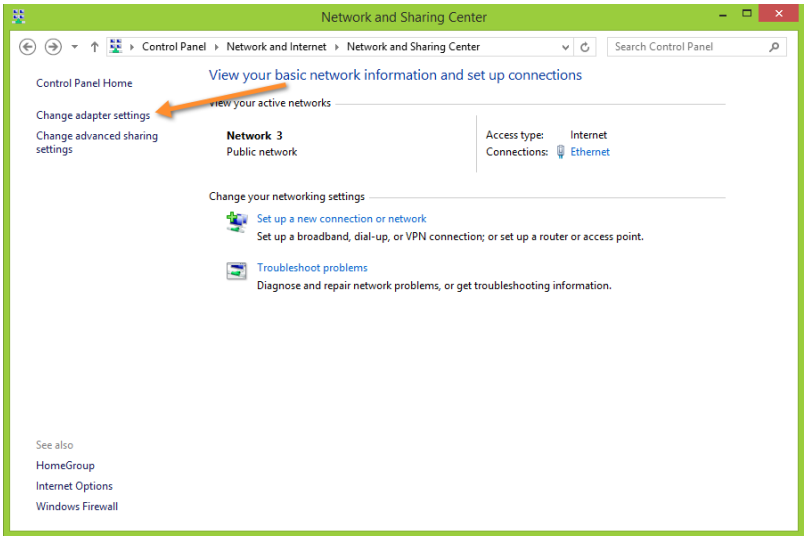

Network and Sharing Center
Control Panel ▸ Network and Internet ▸ Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
View your basic network information and set up connections
View your active networks
Network 3
Public network
Access type: Internet
Connections: Ethernet
Change your networking settings
Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.
Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.
See also
HomeGroup
Internet Options
Windows Firewall

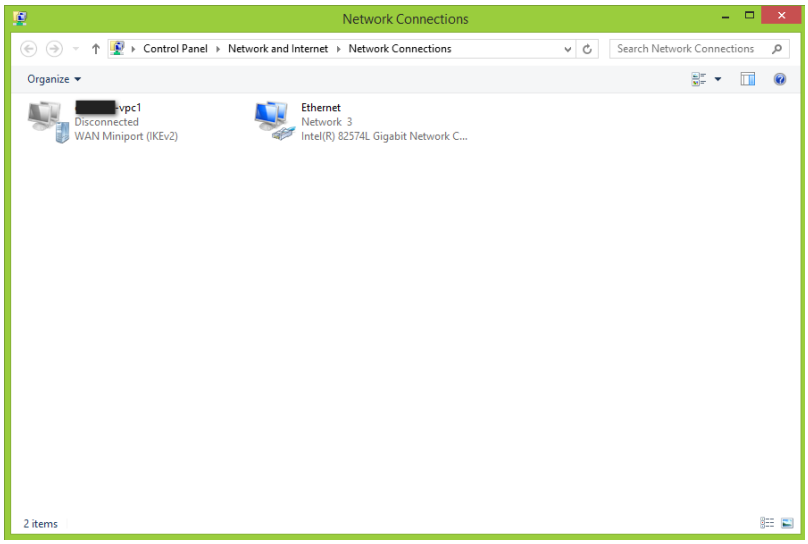

Network and Sharing Center
Control Panel ▸ Network and Internet ▸ Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
Set Up a Connection or Network
Choose a connection option
Connect to the Internet
Set up a broadband or dial-up connection to the Internet.
Set up a new network
Set up a new router or access point.
Connect to a workplace
Set up a dial-up or VPN connection to your workplace.
Next
Cancel
See also
HomeGroup
Internet Options
Windows Firewall

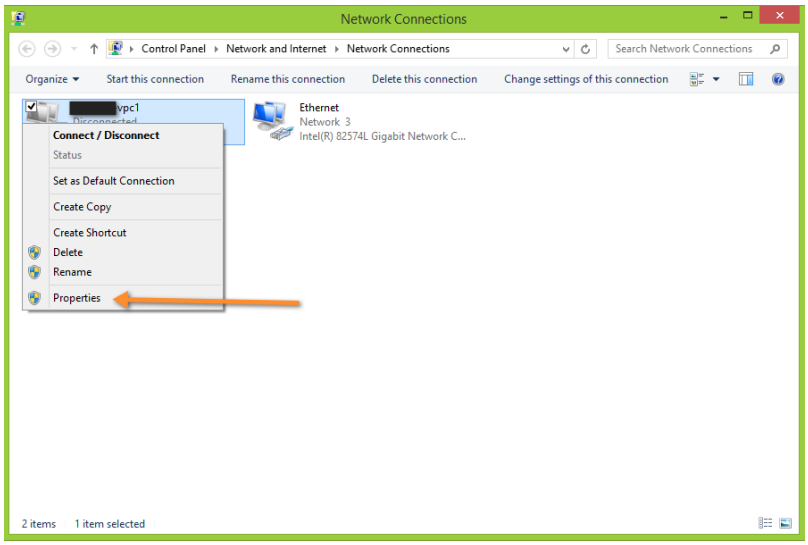


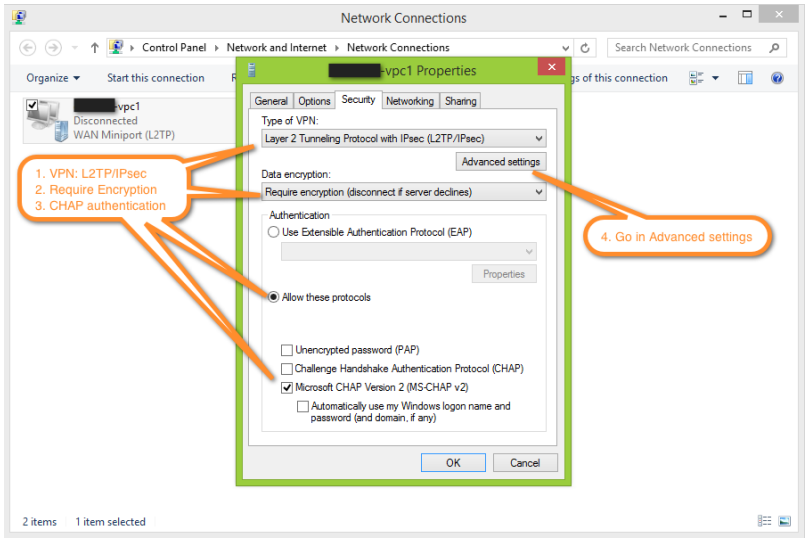


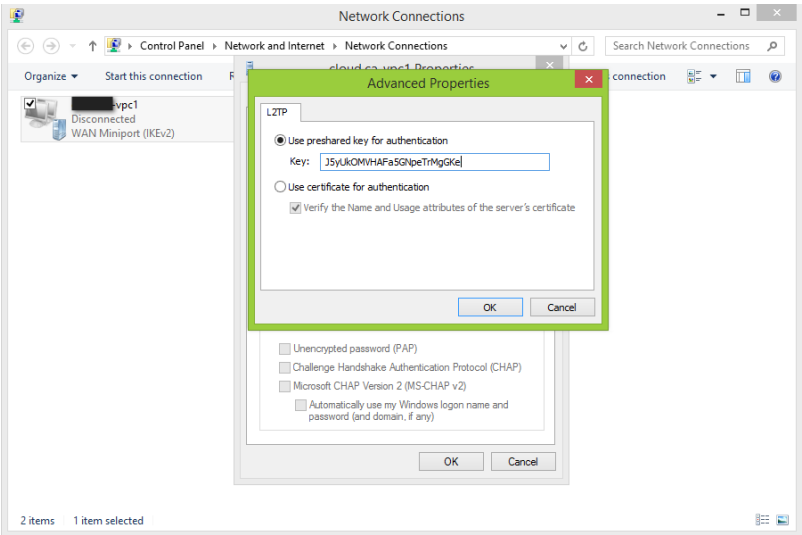
2. Configure VPN settings



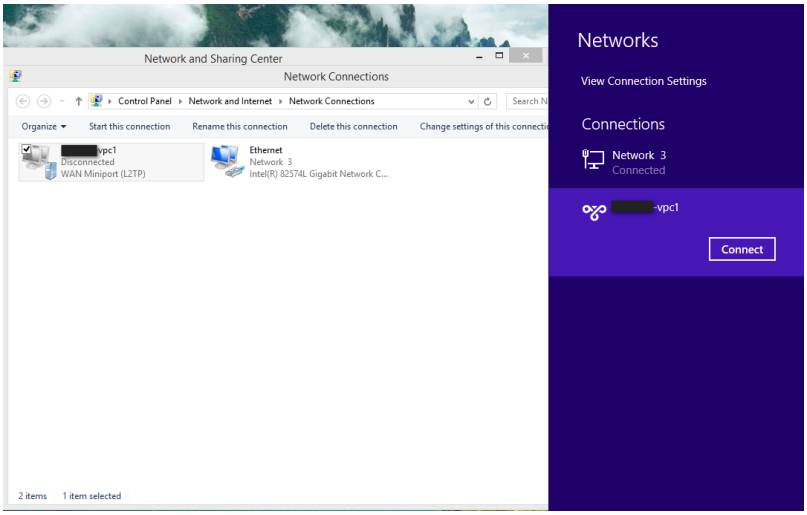

Network Connections
Control Panel ▸ Network and Internet ▸ Network Connections
Search Network Connections
Organize
[redacted]vpc1
Disconnected
WAN Miniport (IKEv2)
Ethernet
Network 3
Intel(R) 82574L Gigabit Network C...
2 items

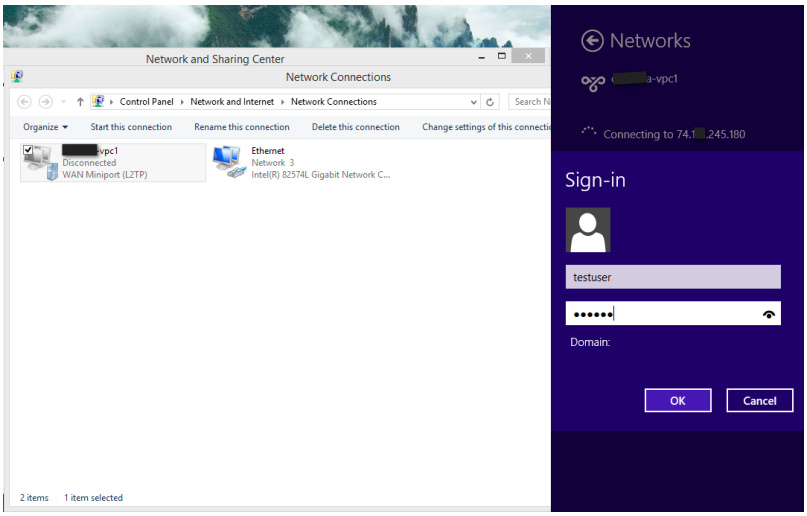

Network Connections
Control Panel ▸ Network and Internet ▸ Network Connections
Search Network Connections
Organize
Start this connection
Rename this connection
Delete this connection
Change settings of this connection
[redacted]vpc1
Ethernet
Network 3
Intel(R) 82574L Gigabit Network C...
Connect / Disconnect
Status
Set as Default Connection
Create Copy
Create Shortcut
Delete
Rename
Properties
2 items
1 item selected

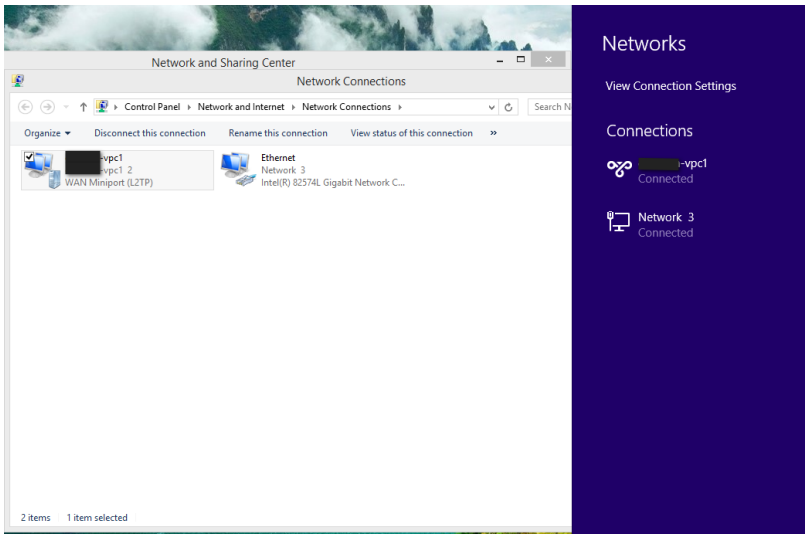

Network Connections
Control Panel ▸ Network and Internet ▸ Network Connections
Search Network Connections
Organize
Start this connection
[redacted]vpc1 Properties
General
Options
Security
Networking
Sharing
[redacted]vpc1
Disconnected
WAN Miniport (L2TP)
Type of VPN:
Layer 2 Tunneling Protocol with IPsec (L2TP/IPsec)
Advanced settings
Data encryption:
Require encryption (disconnect if server declines)
Authentication
Use Extensible Authentication Protocol (EAP)
Properties
Allow these protocols
Unencrypted password (PAP)
Challenge Handshake Authentication Protocol (CHAP)
Microsoft CHAP Version 2 (MS-CHAP v2)
Automatically use my Windows logon name and password (and domain, if any)
OK
Cancel
1. VPN: L2TP/IPsec
2. Require Encryption
3. CHAP authentication
4. Go in Advanced settings
2 items
1 item selected



3. Initiate VPN connection







5.13 Managing the Cloud

5.13.1 Using Tags to Organize Resources in the Cloud

A tag is a key-value pair that stores metadata about a resource in the cloud. Tags are useful for categorizing resources. For example, you can tag a user VM with a value that indicates the user's city of residence. In this case, the key would be "city" and the value might be "Toronto" or "Tokyo." You can then request CloudStack to find all resources that have a given tag; for example, VMs for users in a given city.

You can tag a user virtual machine, volume, snapshot, guest network, template, ISO, firewall rule, port forwarding rule, public IP address, security group, load balancer rule, project, VPC, network ACL, or static route. You can not tag a remote access VPN.

You can work with tags through the UI or through the API commands createTags, deleteTags, and listTags. You can define multiple tags for each resource. There is no limit on the number of tags you can define. Each tag can be up to 255 characters long. Users can define tags on the resources they own, and administrators can define tags on any resources in the cloud.

An optional input parameter, "tags," exists on many of the list* API commands. The following example shows how to use this new parameter to find all the volumes having tag region=canada OR tag city=Toronto:

```
command=listVolumes
   &listAll=true
   &tags[0].key=region
   &tags[0].value=canada
   &tags[1].key=city
   &tags[1].value=Toronto
```

The following API commands have the "tags" input parameter:

- listVirtualMachines
- listVolumes
- listSnapshots
- listNetworks
- listTemplates

- listIsos
- listFirewallRules
- listPortForwardingRules
- listPublicIpAddresses
- listSecurityGroups
- listLoadBalancerRules
- listProjects
- listVPCs
- listNetworkACLs
- listStaticRoutes

5.13.2 Reporting CPU Sockets

PRODUCT manages different types of hosts that contains one or more physical CPU sockets. CPU socket is considered as a unit of measure used for licensing and billing cloud infrastructure. PRODUCT provides both UI and API support to collect the CPU socket statistics for billing purpose. The Infrastructure tab has a new tab for CPU sockets. You can view the statistics for CPU sockets managed by PRODUCT, which in turn reflects the size of the cloud. The CPU Socket page will give you the number of hosts and sockets used for each host type.

1. Log in to the PRODUCT UI.
2. In the left navigation bar, click Infrastructure.
3. On CPU Sockets, click View all.

   The CPU Socket page is displayed. The page shows the number of hosts and CPU sockets based on hypervisor types.

5.13.3 Changing the Database Configuration

The CloudStack Management Server stores database configuration information (e.g., hostname, port, credentials) in the file `/etc/cloudstack/management/db.properties`. To effect a change, edit this file on each Management Server, then restart the Management Server.

5.13.4 Changing the Database Password

You may need to change the password for the MySQL account used by CloudStack. If so, you'll need to change the password in MySQL, and then add the encrypted password to `/etc/cloudstack/management/db.properties`.

1. Before changing the password, you'll need to stop CloudStack's management server and the usage engine if you've deployed that component.

   ```
   # service cloudstack-management stop
   # service cloudstack-usage stop
   ```

2. Next, you'll update the password for the CloudStack user on the MySQL server.

   ```
   # mysql -u root -p
   ```

At the MySQL shell, you'll change the password and flush privileges:

```
update mysql.user set password=PASSWORD("newpassword123") where User='cloud';
flush privileges;
quit;
```

3. The next step is to encrypt the password and copy the encrypted password to CloudStack's database configuration (`/etc/cloudstack/management/db.properties`).

```
# java -classpath /usr/share/cloudstack-common/lib/jasypt-1.9.0.jar \ org.jasypt.
↪intf.cli.JasyptPBEStringEncryptionCLI encrypt.sh \ input="newpassword123"␣
↪password="`cat /etc/cloudstack/management/key`" \ verbose=false
```

5.13.5 File encryption type

Note that this is for the file encryption type. If you're using the web encryption type then you'll use password="management_server_secret_key"

1. Now, you'll update `/etc/cloudstack/management/db.properties` with the new ciphertext. Open `/etc/cloudstack/management/db.properties` in a text editor, and update these parameters:

```
db.cloud.password=ENC(encrypted_password_from_above)
db.usage.password=ENC(encrypted_password_from_above)
```

2. After copying the new password over, you can now start CloudStack (and the usage engine, if necessary).

```
# service cloudstack-management start
# service cloud-usage start
```

5.13.6 Administrator Alerts

The system provides alerts and events to help with the management of the cloud. Alerts are notices to an administrator, generally delivered by e-mail, notifying the administrator that an error has occurred in the cloud. Alert behavior is configurable.

Events track all of the user and administrator actions in the cloud. For example, every guest VM start creates an associated event. Events are stored in the Management Server's database.

Emails will be sent to administrators under the following circumstances:

- The Management Server cluster runs low on CPU, memory, or storage resources
- The Management Server loses heartbeat from a Host for more than 3 minutes
- The Host cluster runs low on CPU, memory, or storage resources

Sending Alerts to External SNMP and Syslog Managers

In addition to showing administrator alerts on the Dashboard in the CloudStack UI and sending them in email, CloudStack can also send the same alerts to external SNMP or Syslog management software. This is useful if you prefer to use an SNMP or Syslog manager to monitor your cloud.

The alerts which can be sent are:

The following is the list of alert type numbers. The current alerts can be found by calling listAlerts.

```
MEMORY = 0 // Available Memory below configured threshold
```

```
CPU = 1 // Unallocated CPU below configured threshold
```

```
STORAGE =2 // Available Storage below configured threshold
```

```
STORAGE_ALLOCATED = 3 // Remaining unallocated Storage is below configured threshold
```

```
PUBLIC_IP = 4 // Number of unallocated virtual network public IPs is below configured␣
↪threshold
```

```
PRIVATE_IP = 5 // Number of unallocated private IPs is below configured threshold
```

```
SECONDARY_STORAGE = 6 //  Available Secondary Storage in availability zone is below␣
↪configured threshold
```

```
HOST = 7 // Host related alerts like host disconnected
```

```
USERVM = 8 // User VM stopped unexpectedly
```

```
DOMAIN_ROUTER = 9 // Domain Router VM stopped unexpectedly
```

```
CONSOLE_PROXY = 10 // Console Proxy VM stopped unexpectedly
```

```
ROUTING = 11 // Lost connection to default route (to the gateway)
```

```
STORAGE_MISC = 12 // Storage issue in system VMs
```

```
USAGE_SERVER = 13 // No usage server process running
```

```
MANAGMENT_NODE = 14 // Management network CIDR is not configured originally
```

```
DOMAIN_ROUTER_MIGRATE = 15 // Domain Router VM Migration was unsuccessful
```

```
CONSOLE_PROXY_MIGRATE = 16 // Console Proxy VM Migration was unsuccessful
```

```
USERVM_MIGRATE = 17 // User VM Migration was unsuccessful
```

```
VLAN = 18 // Number of unallocated VLANs is below configured threshold in␣
↪availability zone
```

```
SSVM = 19 // SSVM stopped unexpectedly
```

```
USAGE_SERVER_RESULT = 20 // Usage job failed
```

```
STORAGE_DELETE = 21 // Failed to delete storage pool
```

```
UPDATE_RESOURCE_COUNT = 22 // Failed to update the resource count
```

```
USAGE_SANITY_RESULT = 23 // Usage Sanity Check failed
```

```
DIRECT_ATTACHED_PUBLIC_IP = 24 // Number of unallocated shared network IPs is low in␣
↪availability zone
```

```
LOCAL_STORAGE = 25 // Remaining unallocated Local Storage is below configured␣
↪threshold
```

```
RESOURCE_LIMIT_EXCEEDED = 26 //Generated when the resource limit exceeds the limit.␣
↪Currently used for recurring snapshots only
```

You can also display the most up to date list by calling the API command `listAlerts`.

SNMP Alert Details

The supported protocol is SNMP version 2.

Each SNMP trap contains the following information: message, podId, dataCenterId, clusterId, and generationTime.

Syslog Alert Details

CloudStack generates a syslog message for every alert. Each syslog message incudes the fields alertType, message, podId, dataCenterId, and clusterId, in the following format. If any field does not have a valid value, it will not be included.

```
Date severity_level Management_Server_IP_Address/Name  alertType:: value␣
↪dataCenterId:: value  podId:: value  clusterId:: value  message:: value
```

For example:

```
Mar  4 10:13:47    WARN    localhost    alertType:: managementNode message::␣
↪Management server node 127.0.0.1 is up
```

Configuring SNMP and Syslog Managers

To configure one or more SNMP managers or Syslog managers to receive alerts from CloudStack:

1. For an SNMP manager, install the CloudStack MIB file on your SNMP manager system. This maps the SNMP OIDs to trap types that can be more easily read by users. The file must be publicly available. For more information on how to install this file, consult the documentation provided with the SNMP manager.
2. Edit the file /etc/cloudstack/management/log4j-cloud.xml.

   ```
   # vi /etc/cloudstack/management/log4j-cloud.xml
   ```

3. Add an entry using the syntax shown below. Follow the appropriate example depending on whether you are adding an SNMP manager or a Syslog manager. To specify multiple external managers, separate the IP addresses and other configuration values with commas (,).

Note: The recommended maximum number of SNMP or Syslog managers is 20 for each.

The following example shows how to configure two SNMP managers at IP addresses 10.1.1.1 and 10.1.1.2. Substitute your own IP addresses, ports, and communities. Do not change the other values (name, threshold, class, and layout values).

```
<appender name="SNMP" class="org.apache.cloudstack.alert.snmp.SnmpTrapAppender">
  <param name="Threshold" value="WARN"/>  <!-- Do not edit. The alert feature␣
↪assumes WARN. -->
  <param name="SnmpManagerIpAddresses" value="10.1.1.1,10.1.1.2"/>
  <param name="SnmpManagerPorts" value="162,162"/>
  <param name="SnmpManagerCommunities" value="public,public"/>
  <layout class="org.apache.cloudstack.alert.snmp.SnmpEnhancedPatternLayout"> <!--
↪ Do not edit -->
    <param name="PairDelimeter" value="//"/>
    <param name="KeyValueDelimeter" value="::"/>
  </layout>
</appender>
```

The following example shows how to configure two Syslog managers at IP addresses 10.1.1.1 and 10.1.1.2. Substitute your own IP addresses. You can set Facility to any syslog-defined value, such as LOCAL0 - LOCAL7. Do not change the other values.

```
<appender name="ALERTSYSLOG">
  <param name="Threshold" value="WARN"/>
  <param name="SyslogHosts" value="10.1.1.1,10.1.1.2"/>
  <param name="Facility" value="LOCAL6"/>
  <layout>
    <param name="ConversionPattern" value=""/>
  </layout>
</appender>
```

4. If your cloud has multiple Management Server nodes, repeat these steps to edit log4j-cloud.xml on every instance.
5. If you have made these changes while the Management Server is running, wait a few minutes for the change to take effect.

Troubleshooting: If no alerts appear at the configured SNMP or Syslog manager after a reasonable amount of time, it is likely that there is an error in the syntax of the <appender> entry in log4j-cloud.xml. Check to be sure that the format and settings are correct.

Deleting an SNMP or Syslog Manager

To remove an external SNMP manager or Syslog manager so that it no longer receives alerts from CloudStack, remove the corresponding entry from the file `/etc/cloudstack/management/log4j-cloud.xml`.

5.13.7 Customizing the Network Domain Name

The root administrator can optionally assign a custom DNS suffix at the level of a network, account, domain, zone, or entire CloudStack installation, and a domain administrator can do so within their own domain. To specify a custom domain name and put it into effect, follow these steps.

1. Set the DNS suffix at the desired scope

- At the network level, the DNS suffix can be assigned through the UI when creating a new network, as described in "Adding an Additional Guest Network" or with the updateNetwork command in the CloudStack API.
- At the account, domain, or zone level, the DNS suffix can be assigned with the appropriate CloudStack API commands: createAccount, editAccount, createDomain, editDomain, createZone, or editZone.
- At the global level, use the configuration parameter guest.domain.suffix. You can also use the CloudStack API command updateConfiguration. After modifying this global configuration, restart the Management Server to put the new setting into effect.

2. To make the new DNS suffix take effect for an existing network, call the CloudStack API command updateNetwork. This step is not necessary when the DNS suffix was specified while creating a new network.

The source of the network domain that is used depends on the following rules.

- For all networks, if a network domain is specified as part of a network's own configuration, that value is used.
- For an account-specific network, the network domain specified for the account is used. If none is specified, the system looks for a value in the domain, zone, and global configuration, in that order.
- For a domain-specific network, the network domain specified for the domain is used. If none is specified, the system looks for a value in the zone and global configuration, in that order.
- For a zone-specific network, the network domain specified for the zone is used. If none is specified, the system looks for a value in the global configuration.

5.13.8 Stopping and Restarting the Management Server

The root administrator will need to stop and restart the Management Server from time to time.

For example, after changing a global configuration parameter, a restart is required. If you have multiple Management Server nodes, restart all of them to put the new parameter value into effect consistently throughout the cloud..

To stop the Management Server, issue the following command at the operating system prompt on the Management Server node:

```
# service cloudstack-management stop
```

To start the Management Server:

```
# service cloudstack-management start
```

5.14 System Reliability and Availability

5.14.1 HA for Management Server

The CloudStack Management Server should be deployed in a multi-node configuration such that it is not susceptible to individual server failures. The Management Server itself (as distinct from the MySQL database) is stateless and may be placed behind a load balancer.

Normal operation of Hosts is not impacted by an outage of all Management Serves. All guest VMs will continue to work.

When the Management Server is down, no new VMs can be created, and the end user and admin UI, API, dynamic load distribution, and HA will cease to work.

5.14.2 Management Server Load Balancing

CloudStack can use a load balancer to provide a virtual IP for multiple Management Servers. The administrator is responsible for creating the load balancer rules for the Management Servers. The application requires persistence or stickiness across multiple sessions. The following chart lists the ports that should be load balanced and whether or not persistence is required.

Even if persistence is not required, enabling it is permitted.

Source Port	Destination Port	Protocol	Persistence Required?
80 or 443	8080 (or 20400 with AJP)	HTTP (or AJP)	Yes
8250	8250	TCP	Yes
8096	8096	HTTP	No

In addition to above settings, the administrator is responsible for setting the 'host' global config value from the management server IP to load balancer virtual IP address. If the 'host' value is not set to the VIP for Port 8250 and one of your management servers crashes, the UI is still available but the system VMs will not be able to contact the management server.

5.14.3 HA-Enabled Virtual Machines

The user can specify a virtual machine as HA-enabled. By default, all virtual router VMs and Elastic Load Balancing VMs are automatically configured as HA-enabled. When an HA-enabled VM crashes, CloudStack detects the crash and restarts the VM automatically within the same Availability Zone. HA is never performed across different Availability Zones. CloudStack has a conservative policy towards restarting VMs and ensures that there will never be two instances of the same VM running at the same time. The Management Server attempts to start the VM on another Host in the same cluster.

HA features work with iSCSI or NFS primary storage. HA with local storage is not supported.

5.14.4 HA for Hosts

The user can specify a virtual machine as HA-enabled. By default, all virtual router VMs and Elastic Load Balancing VMs are automatically configured as HA-enabled. When an HA-enabled VM crashes, CloudStack detects the crash and restarts the VM automatically within the same Availability Zone. HA is never performed across different Availability Zones. CloudStack has a conservative policy towards restarting VMs and ensures that there will never be two instances of the same VM running at the same time. The Management Server attempts to start the VM on another Host in the same cluster.

HA features work with iSCSI or NFS primary storage. HA with local storage is not supported.

Dedicated HA Hosts

One or more hosts can be designated for use only by HA-enabled VMs that are restarting due to a host failure. Setting up a pool of such dedicated HA hosts as the recovery destination for all HA-enabled VMs is useful to:

- Make it easier to determine which VMs have been restarted as part of the CloudStack high-availability function. If a VM is running on a dedicated HA host, then it must be an HA-enabled VM whose original host failed. (With one exception: It is possible for an administrator to manually migrate any VM to a dedicated HA host.).
- Keep HA-enabled VMs from restarting on hosts which may be reserved for other purposes.

The dedicated HA option is set through a special host tag when the host is created. To allow the administrator to dedicate hosts to only HA-enabled VMs, set the global configuration variable ha.tag to the desired tag (for example, "ha_host"), and restart the Management Server. Enter the value in the Host Tags field when adding the host(s) that you want to dedicate to HA-enabled VMs.

Note: If you set ha.tag, be sure to actually use that tag on at least one host in your cloud. If the tag specified in ha.tag is not set for any host in the cloud, the HA-enabled VMs will fail to restart after a crash.

5.14.5 Primary Storage Outage and Data Loss

When a primary storage outage occurs the hypervisor immediately stops all VMs stored on that storage device. Guests that are marked for HA will be restarted as soon as practical when the primary storage comes back on line. With NFS, the hypervisor may allow the virtual machines to continue running depending on the nature of the issue. For example, an NFS hang will cause the guest VMs to be suspended until storage connectivity is restored.Primary storage is not designed to be backed up. Individual volumes in primary storage can be backed up using snapshots.

5.14.6 Secondary Storage Outage and Data Loss

For a Zone that has only one secondary storage server, a secondary storage outage will have feature level impact to the system but will not impact running guest VMs. It may become impossible to create a VM with the selected template for a user. A user may also not be able to save snapshots or examine/restore saved snapshots. These features will automatically be available when the secondary storage comes back online.

Secondary storage data loss will impact recently added user data including templates, snapshots, and ISO images. Secondary storage should be backed up periodically. Multiple secondary storage servers can be provisioned within each zone to increase the scalability of the system.

5.14.7 Database High Availability

To help ensure high availability of the databases that store the internal data for CloudStack, you can set up database replication. This covers both the main CloudStack database and the Usage database. Replication is achieved using the MySQL connector parameters and two-way replication. Tested with MySQL 5.1 and 5.5.

How to Set Up Database Replication

Database replication in CloudStack is provided using the MySQL replication capabilities. The steps to set up replication can be found in the MySQL documentation (links are provided below). It is suggested that you set up two-way replication, which involves two database nodes. In this case, for example, you might have node1 and node2.

You can also set up chain replication, which involves more than two nodes. In this case, you would first set up two-way replication with node1 and node2. Next, set up one-way replication from node2 to node3. Then set up one-way replication from node3 to node4, and so on for all the additional nodes.

References:

- http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html
- https://wikis.oracle.com/display/CommSuite/MySQL+High+Availability+and+Replication+Information+For+Calendar+Server

Configuring Database High Availability

To control the database high availability behavior, use the following configuration settings in the file /etc/cloudstack/management/db.properties.

Required Settings

Be sure you have set the following in db.properties:

- `db.ha.enabled`: set to true if you want to use the replication feature.

  Example: `db.ha.enabled=true`

- `db.cloud.slaves`: set to a comma-delimited set of slave hosts for the cloud database. This is the list of nodes set up with replication. The master node is not in the list, since it is already mentioned elsewhere in the properties file.

  Example: `db.cloud.slaves=node2,node3,node4`

- `db.usage.slaves`: set to a comma-delimited set of slave hosts for the usage database. This is the list of nodes set up with replication. The master node is not in the list, since it is already mentioned elsewhere in the properties file.

  Example: `db.usage.slaves=node2,node3,node4`

Optional Settings

The following settings must be present in db.properties, but you are not required to change the default values unless you wish to do so for tuning purposes:

- `db.cloud.secondsBeforeRetryMaster`: The number of seconds the MySQL connector should wait before trying again to connect to the master after the master went down. Default is 1 hour. The retry might happen sooner if db.cloud.queriesBeforeRetryMaster is reached first.

  Example: `db.cloud.secondsBeforeRetryMaster=3600`

- `db.cloud.queriesBeforeRetryMaster`: The minimum number of queries to be sent to the database before trying again to connect to the master after the master went down. Default is 5000. The retry might happen sooner if db.cloud.secondsBeforeRetryMaster is reached first.

  Example: `db.cloud.queriesBeforeRetryMaster=5000`

- `db.cloud.initialTimeout`: Initial time the MySQL connector should wait before trying again to connect to the master. Default is 3600.

  Example: `db.cloud.initialTimeout=3600`

Limitations on Database High Availability

The following limitations exist in the current implementation of this feature.

- Slave hosts can not be monitored through CloudStack. You will need to have a separate means of monitoring.
- Events from the database side are not integrated with the CloudStack Management Server events system.
- You must periodically perform manual clean-up of bin log files generated by replication on database nodes. If you do not clean up the log files, the disk can become full.

5.15 Tuning

5.15.1 Tuning

This section provides tips on how to improve the performance of your cloud.

Performance Monitoring

Host and guest performance monitoring is available to end users and administrators. This allows the user to monitor their utilization of resources and determine when it is appropriate to choose a more powerful service offering or larger disk.

Increase Management Server Maximum Memory

If the Management Server is subject to high demand, the default maximum JVM memory allocation can be insufficient. To increase the memory:

1. Edit the Tomcat configuration file:

```
/etc/cloudstack/management/tomcat6.conf
```

2. Change the command-line parameter -XmxNNNm to a higher value of N.

   For example, if the current value is -Xmx128m, change it to -Xmx1024m or higher.

3. To put the new setting into effect, restart the Management Server.

```
# service cloudstack-management restart
```

For more information about memory issues, see "FAQ: Memory" at Tomcat Wiki.

Set Database Buffer Pool Size

It is important to provide enough memory space for the MySQL database to cache data and indexes:

1. Edit the MySQL configuration file:

```
/etc/my.cnf
```

2. Insert the following line in the [mysqld] section, below the datadir line. Use a value that is appropriate for your situation. We recommend setting the buffer pool at 40% of RAM if MySQL is on the same server as the management server or 70% of RAM if MySQL has a dedicated server. The following example assumes a dedicated server with 1024M of RAM.

```
innodb_buffer_pool_size=700M
```

3. Restart the MySQL service.

```
# service mysqld restart
```

For more information about the buffer pool, see "The InnoDB Buffer Pool" at MySQL Reference Manual.

Set and Monitor Total VM Limits per Host

The CloudStack administrator should monitor the total number of VM instances in each cluster, and disable allocation to the cluster if the total is approaching the maximum that the hypervisor can handle. Be sure to leave a safety margin to allow for the possibility of one or more hosts failing, which would increase the VM load on the other hosts as the VMs are automatically redeployed. Consult the documentation for your chosen hypervisor to find the maximum permitted number of VMs per host, then use CloudStack global configuration settings to set this as the default limit. Monitor the VM activity in each cluster at all times. Keep the total number of VMs below a safe level that allows for the occasional host failure. For example, if there are N hosts in the cluster, and you want to allow for one host in the cluster to be down at any given time, the total number of VM instances you can permit in the cluster is at most (N-1) * (per-host-limit). Once a cluster reaches this number of VMs, use the CloudStack UI to disable allocation of more VMs to the cluster.

Configure XenServer dom0 Memory

Configure the XenServer dom0 settings to allocate more memory to dom0. This can enable XenServer to handle larger numbers of virtual machines. We recommend 2940 MB of RAM for XenServer dom0. For instructions on how to do this, see Citrix Knowledgebase Article.The article refers to XenServer 5.6, but the same information applies to XenServer 6

5.16 Events and Troubleshooting

5.16.1 Event Notification

An event is essentially a significant or meaningful change in the state of both virtual and physical resources associated with a cloud environment. Events are used by monitoring systems, usage and billing systems, or any other event-driven workflow systems to discern a pattern and make the right business decision. In CloudStack an event could be a state change of virtual or physical resources, an action performed by an user (action events), or policy based events (alerts).

Event Logs

There are two types of events logged in the CloudStack Event Log. Standard events log the success or failure of an event and can be used to identify jobs or processes that have failed. There are also long running job events. Events for asynchronous jobs log when a job is scheduled, when it starts, and when it completes. Other long running synchronous jobs log when a job starts, and when it completes. Long running synchronous and asynchronous event logs can be used to gain more information on the status of a pending job or can be used to identify a job that is hanging or has not started. The following sections provide more information on these events..

Notification

Event notification framework provides a means for the Management Server components to publish and subscribe to CloudStack events. Event notification is achieved by implementing the concept of event bus abstraction in the Management Server.

A new event for state change, resource state change, is introduced as part of Event notification framework. Every resource, such as user VM, volume, NIC, network, public IP, snapshot, and template, is associated with a state machine and generates events as part of the state change. That implies that a change in the state of a resource results in a state change event, and the event is published in the corresponding state machine on the event bus. All the CloudStack events (alerts, action events, usage events) and the additional category of resource state change events, are published on to the events bus.

Implementations

An event bus is introduced in the Management Server that allows the CloudStack components and extension plug-ins to subscribe to the events by using the Advanced Message Queuing Protocol (AMQP) client. In CloudStack, a default implementation of event bus is provided as a plug-in that uses the RabbitMQ AMQP client. The AMQP client pushes the published events to a compatible AMQP server. Therefore all the CloudStack events are published to an exchange in the AMQP server.

Additionally, both an in-memory implementation and an Apache Kafka implementation are also available.

Use Cases

The following are some of the use cases:

- Usage or Billing Engines: A third-party cloud usage solution can implement a plug-in that can connects to CloudStack to subscribe to CloudStack events and generate usage data. The usage data is consumed by their usage software.
- AMQP plug-in can place all the events on the a message queue, then a AMQP message broker can provide topic-based notification to the subscribers.
- Publish and Subscribe notification service can be implemented as a pluggable service in CloudStack that can provide rich set of APIs for event notification, such as topics-based subscription and notification. Additionally, the pluggable service can deal with multi-tenancy, authentication, and authorization issues.

AMQP Configuration

As a CloudStack administrator, perform the following one-time configuration to enable event notification framework. At run time no changes can control the behaviour.

1. Create the folder `/etc/cloudstack/management/META-INF/cloudstack/core`
2. Inside that folder, open `spring-event-bus-context.xml`.
3. Define a bean named `eventNotificationBus` as follows:
   - name : Specify a name for the bean.
   - server : The name or the IP address of the RabbitMQ AMQP server.
   - port : The port on which RabbitMQ server is running.
   - username : The username associated with the account to access the RabbitMQ server.
   - password : The password associated with the username of the account to access the RabbitMQ server.
   - exchange : The exchange name on the RabbitMQ server where CloudStack events are published.

   A sample bean is given below:

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/
↪schema/aop/spring-aop-3.0.xsd
http://www.springframework.org/schema/context
```

(continues on next page)

(continued from previous page)

```
http://www.springframework.org/schema/context/spring-context-3.0.xsd">
   <bean id="eventNotificationBus" class="org.apache.cloudstack.mom.rabbitmq.
↪RabbitMQEventBus">
      <property name="name" value="eventNotificationBus"/>
      <property name="server" value="127.0.0.1"/>
      <property name="port" value="5672"/>
      <property name="username" value="guest"/>
      <property name="password" value="guest"/>
      <property name="exchange" value="cloudstack-events"/>
   </bean>
</beans>
```

The `eventNotificationBus` bean represents the `org.apache.cloudstack.mom.rabbitmq.RabbitMQEventBus` class.

If you want to use encrypted values for the username and password, you have to include a bean to pass those as variables from a credentials file.

A sample is given below

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/aop http://www.springframework.
↪org/schema/aop/spring-aop-3.0.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.0.xsd"
>

   <bean id="eventNotificationBus" class="org.apache.cloudstack.mom.rabbitmq.
↪RabbitMQEventBus">
      <property name="name" value="eventNotificationBus"/>
      <property name="server" value="127.0.0.1"/>
      <property name="port" value="5672"/>
      <property name="username" value="${username}"/>
      <property name="password" value="${password}"/>
      <property name="exchange" value="cloudstack-events"/>
   </bean>

   <bean id="environmentVariablesConfiguration" class="org.jasypt.encryption.
↪pbe.config.EnvironmentStringPBEConfig">
      <property name="algorithm" value="PBEWithMD5AndDES" />
      <property name="passwordEnvName" value="APP_ENCRYPTION_PASSWORD" />
   </bean>

   <bean id="configurationEncryptor" class="org.jasypt.encryption.pbe.
↪StandardPBEStringEncryptor">
      <property name="config" ref="environmentVariablesConfiguration" />
   </bean>

   <bean id="propertyConfigurer" class="org.jasypt.spring3.properties.
↪EncryptablePropertyPlaceholderConfigurer">
      <constructor-arg ref="configurationEncryptor" />
      <property name="location" value="classpath:/cred.properties" />
```

(continues on next page)

(continued from previous page)

```
    </bean>
</beans>
```

Create a new file in the same folder called `cred.properties` and the specify the values for username and password as jascrypt encrypted strings

Sample, with `guest` as values for both fields:

```
username=nh2XrM7jWHMG4VQK18iiBQ==
password=nh2XrM7jWHMG4VQK18iiBQ==
```

4. Restart the Management Server.

Kafka Configuration

As a CloudStack administrator, perform the following one-time configuration to enable event notification framework. At run time no changes can control the behaviour.

1. Create an appropriate configuration file in `/etc/cloudstack/management/kafka.producer.properties` which contains valid kafka configuration properties as documented in http://kafka.apache.org/documentation.html#newproducerconfigs The properties may contain an additional `topic` property which if not provided will default to `cloudstack`. While `key.serializer` and `value.serializer` are usually required for a producer to correctly start, they may be omitted and will default to `org.apache.kafka.common.serialization.StringSerializer`.
2. Create the folder `/etc/cloudstack/management/META-INF/cloudstack/core`
3. Inside that folder, open `spring-event-bus-context.xml`.
4. Define a bean named `eventNotificationBus` with a single `name` attribute, A sample bean is given below:

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-
↪beans-3.0.xsd
                           http://www.springframework.org/schema/aop http://www.
↪springframework.org/schema/aop/spring-aop-3.0.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-
↪context-3.0.xsd">
   <bean id="eventNotificationBus" class="org.apache.cloudstack.mom.kafka.
↪KafkaEventBus">
     <property name="name" value="eventNotificationBus"/>
   </bean>
 </beans>
```

5. Restart the Management Server.

Standard Events

The events log records three types of standard events.

- INFO. This event is generated when an operation has been successfully performed.

- WARN. This event is generated in the following circumstances.
  - When a network is disconnected while monitoring a template download.
  - When a template download is abandoned.
  - When an issue on the storage server causes the volumes to fail over to the mirror storage server.
- ERROR. This event is generated when an operation has not been successfully performed

Long Running Job Events

The events log records three types of standard events.

- INFO. This event is generated when an operation has been successfully performed.
- WARN. This event is generated in the following circumstances.
  - When a network is disconnected while monitoring a template download.
  - When a template download is abandoned.
  - When an issue on the storage server causes the volumes to fail over to the mirror storage server.
- ERROR. This event is generated when an operation has not been successfully performed

Event Log Queries

Database logs can be queried from the user interface. The list of events captured by the system includes:

- Virtual machine creation, deletion, and on-going management operations
- Virtual router creation, deletion, and on-going management operations
- Template creation and deletion
- Network/load balancer rules creation and deletion
- Storage volume creation and deletion
- User login and logout

Deleting and Archiving Events and Alerts

CloudStack provides you the ability to delete or archive the existing alerts and events that you no longer want to implement. You can regularly delete or archive any alerts or events that you cannot, or do not want to resolve from the database.

You can delete or archive individual alerts or events either directly by using the Quickview or by using the Details page. If you want to delete multiple alerts or events at the same time, you can use the respective context menu. You can delete alerts or events by category for a time period. For example, you can select categories such as **USER.LOGOUT**, **VM.DESTROY**, **VM.AG.UPDATE**, **CONFIGURATION.VALUE.EDI**, and so on. You can also view the number of events or alerts archived or deleted.

In order to support the delete or archive alerts, the following global parameters have been added:

- **alert.purge.delay**: The alerts older than specified number of days are purged. Set the value to 0 to never purge alerts automatically.
- **alert.purge.interval**: The interval in seconds to wait before running the alert purge thread. The default is 86400 seconds (one day).

Note: Archived alerts or events cannot be viewed in the UI or by using the API. They are maintained in the database for auditing or compliance purposes.

Permissions

Consider the following:

- The root admin can delete or archive one or multiple alerts or events.
- The domain admin or end user can delete or archive one or multiple events.

Procedure

1. Log in as administrator to the CloudStack UI.
2. In the left navigation, click Events.
3. Perform either of the following:
    - To archive events, click Archive Events, and specify event type and date.
    - To archive events, click Delete Events, and specify event type and date.
4. Click OK.

5.16.2 TroubleShooting

Working with Server Logs

The CloudStack Management Server logs all web site, middle tier, and database activities for diagnostics purposes in */var/log/cloudstack/management/*. The CloudStack logs a variety of error messages. We recommend this command to find the problematic output in the Management Server log:.

Note: When copying and pasting a command, be sure the command has pasted as a single line before executing. Some document viewers may introduce unwanted line breaks in copied text.

```
grep -i -E 'exception|unable|fail|invalid|leak|warn|error' /var/log/cloudstack/
↪management/management-server.log
```

The CloudStack processes requests with a Job ID. If you find an error in the logs and you are interested in debugging the issue you can grep for this job ID in the management server log. For example, suppose that you find the following ERROR message:

```
2010-10-04 13:49:32,595 ERROR [cloud.vm.UserVmManagerImpl] (Job-Executor-11:job-1076)␣
↪Unable to find any host for [User|i-8-42-VM-untagged]
```

Note that the job ID is 1076. You can track back the events relating to job 1076 with the following grep:

```
grep "job-1076)" management-server.log
```

The CloudStack Agent Server logs its activities in */var/log/cloudstack/agent/*.

Data Loss on Exported Primary Storage

Symptom

Loss of existing data on primary storage which has been exposed as a Linux NFS server export on an iSCSI volume.

Cause

It is possible that a client from outside the intended pool has mounted the storage. When this occurs, the LVM is wiped and all data in the volume is lost

Solution

When setting up LUN exports, restrict the range of IP addresses that are allowed access by specifying a subnet mask. For example:

```
echo "/export 192.168.1.0/24(rw,async,no_root_squash,no_subtree_check)" > /etc/exports
```

Adjust the above command to suit your deployment needs.

More Information

See the export procedure in the "Secondary Storage" section of the CloudStack Installation Guide

Recovering a Lost Virtual Router

Symptom

A virtual router is running, but the host is disconnected. A virtual router no longer functions as expected.

Cause

The Virtual router is lost or down.

Solution

If you are sure that a virtual router is down forever, or no longer functions as expected, destroy it. You must create one afresh while keeping the backup router up and running (it is assumed this is in a redundant router setup):

- Force stop the router. Use the stopRouter API with forced=true parameter to do so.
- Before you continue with destroying this router, ensure that the backup router is running. Otherwise the network connection will be lost.
- Destroy the router by using the destroyRouter API.

Recreate the missing router by using the restartNetwork API with cleanup=false parameter. For more information about redundant router setup, see Creating a New Network Offering.

For more information about the API syntax, see the API Reference at http://cloudstack.apache.org/docs/api/.

Maintenance mode not working on vCenter

Symptom

Host was placed in maintenance mode, but still appears live in vCenter.

Cause

The CloudStack administrator UI was used to place the host in scheduled maintenance mode. This mode is separate from vCenter's maintenance mode.

Solution

Use vCenter to place the host in maintenance mode.

Unable to deploy VMs from uploaded vSphere template

Symptom

When attempting to create a VM, the VM will not deploy.

Cause

If the template was created by uploading an OVA file that was created using vSphere Client, it is possible the OVA contained an ISO image. If it does, the deployment of VMs from the template will fail.

Solution

Remove the ISO and re-upload the template.

Unable to power on virtual machine on VMware

Symptom

Virtual machine does not power on. You might see errors like:

- Unable to open Swap File
- Unable to access a file since it is locked
- Unable to access Virtual machine configuration

Cause

A known issue on VMware machines. ESX hosts lock certain critical virtual machine files and file systems to prevent concurrent changes. Sometimes the files are not unlocked when the virtual machine is powered off. When a virtual machine attempts to power on, it can not access these critical files, and the virtual machine is unable to power on.

Solution

See the following:

VMware Knowledge Base Article

Load balancer rules fail after changing network offering

Symptom

After changing the network offering on a network, load balancer rules stop working.

Cause

Load balancing rules were created while using a network service offering that includes an external load balancer device such as NetScaler, and later the network service offering changed to one that uses the CloudStack virtual router.

Solution

Create a firewall rule on the virtual router for each of your existing load balancing rules so that they continue to function.

Troubleshooting Internet Traffic

Below are a few troubleshooting steps to check whats going wrong with your network…

Trouble Shooting Steps

1. The switches have to be configured correctly to pass VLAN traffic. You can verify if VLAN traffic is working by bringing up a tagged interface on the hosts and pinging between them as below…

   On *host1 (kvm1)*

   ```
   kvm1 ~$ vconfig add eth0 64
   kvm1 ~$ ifconfig eth0.64 1.2.3.4 netmask 255.255.255.0 up
   kvm1 ~$ ping 1.2.3.5
   ```

   On *host2 (kvm2)*

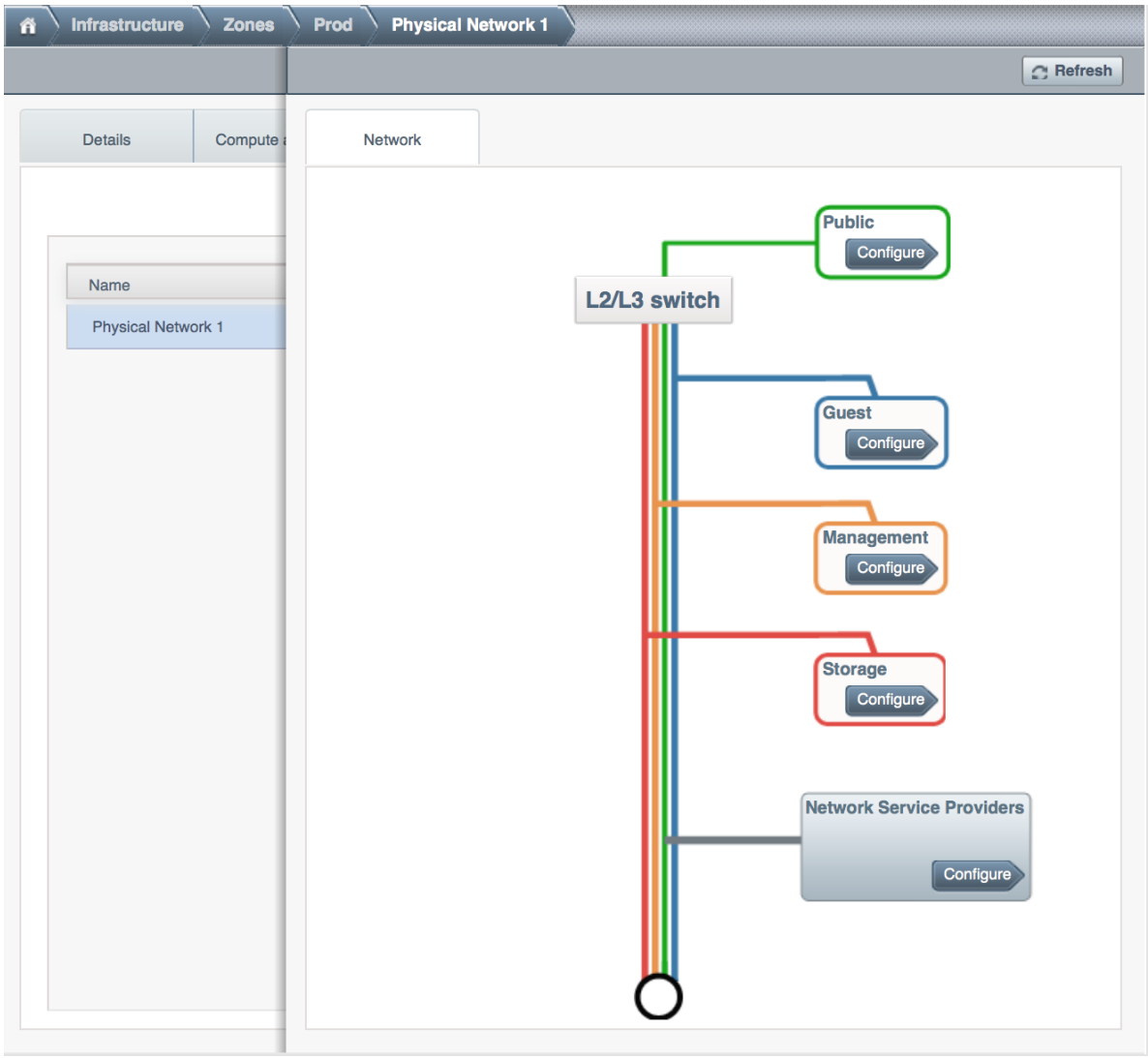
   ```
   kvm2 ~$ vconfig add eth0 64
   kvm2 ~$ ifconfig eth0.64 1.2.3.5 netmask 255.255.255.0 up
   kvm2 ~$ ping 1.2.3.4
   ```

   If the pings dont work, run *tcpdump(8)* all over the place to check who is gobbling up the packets. Ultimately, if the switches are not configured correctly, CloudStack networking wont work so fix the physical networking issues before you proceed to the next steps

2. Ensure Traffic Labels are set for the Zone.

   Traffic labels need to be set for all hypervisors including XenServer, KVM and VMware types. You can configure traffic labels when you creating a new zone from the *Add Zone Wizard*.

On an existing zone, you can modify the traffic labels by going to *Infrastructure, Zones, Physical Network* tab.



List labels using *CloudMonkey*

```
acs-manager ~$ cloudmonkey list traffictypes physicalnetworkid=41cb7ff6-8eb2-4630-
↪b577-1da25e0e1145
count = 4
traffictype:
id = cd0915fe-a660-4a82-9df7-34aebf90003e
kvmnetworklabel = cloudbr0
physicalnetworkid = 41cb7ff6-8eb2-4630-b577-1da25e0e1145
traffictype = Guest
xennetworklabel = MGMT
=========================================================
id = f5524b8f-6605-41e4-a982-81a356b2a196
kvmnetworklabel = cloudbr0
physicalnetworkid = 41cb7ff6-8eb2-4630-b577-1da25e0e1145
traffictype = Management
xennetworklabel = MGMT
=========================================================
id = 266bad0e-7b68-4242-b3ad-f59739346cfd
kvmnetworklabel = cloudbr0
```

(continues on next page)

(continued from previous page)

```
physicalnetworkid = 41cb7ff6-8eb2-4630-b577-1da25e0e1145
traffictype = Public
xennetworklabel = MGMT
=========================================================
id = a2baad4f-7ce7-45a8-9caf-a0b9240adf04
kvmnetworklabel = cloudbr0
physicalnetworkid = 41cb7ff6-8eb2-4630-b577-1da25e0e1145
traffictype = Storage
xennetworklabel = MGMT
==========================================================
```

3. KVM traffic labels require to be named as *"cloudbr0"*, *"cloudbr2"*, *"cloudbrN"* etc and the corresponding bridge must exist on the KVM hosts. If you create labels/bridges with any other names, CloudStack (atleast earlier versions did) seems to ignore them. CloudStack does not create the physical bridges on the KVM hosts, you need to create them **before** before adding the host to Cloudstack.

```
kvm1 ~$ ifconfig cloudbr0
cloudbr0  Link encap:Ethernet  HWaddr 00:0C:29:EF:7D:78
   inet addr:192.168.44.22  Bcast:192.168.44.255  Mask:255.255.255.0
   inet6 addr: fe80::20c:29ff:feef:7d78/64 Scope:Link
   UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
   RX packets:92435 errors:0 dropped:0 overruns:0 frame:0
   TX packets:50596 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:0
   RX bytes:94985932 (90.5 MiB)  TX bytes:61635793 (58.7 MiB)
```

4. The Virtual Router, SSVM, CPVM *public* interface would be bridged to a physical interface on the host. In the example below, *cloudbr0* is the public interface and CloudStack has correctly created the virtual interfaces bridge. This virtual interface to physical interface mapping is done automatically by CloudStack using the traffic label settings for the Zone. If you have provided correct settings and still dont have a working working Internet, check the switching layer before you debug any further. You can verify traffic using tcpdump on the virtual, physical and bridge interfaces.

```
kvm-host1 ~$ brctl show
bridge name  bridge id           STP enabled interfaces
breth0-64    8000.000c29ef7d78   no          eth0.64
                                             vnet2
cloud0       8000.fe00a9fe0219   no          vnet0
cloudbr0     8000.000c29ef7d78   no          eth0
                                             vnet1
                                             vnet3
virbr0       8000.5254008e321a   yes         virbr0-nic
```

```
xenserver1 ~$ brctl show
bridge name  bridge id           STP enabled interfaces
xapi0    0000.e2b76d0a1149       no          vif1.0
xenbr0   0000.000c299b54dc       no          eth0
                                            xapi1
                                            vif1.1
                                            vif1.2
```

5. Pre-create labels on the XenServer Hosts. Similar to KVM bridge setup, traffic labels must also be pre-created on the XenServer hosts before adding them to CloudStack.

```
xenserver1 ~$ xe network-list
```

(continues on next page)

(continued from previous page)

```
uuid ( RO)                : aaa-bbb-ccc-ddd
          name-label ( RW): MGMT
    name-description ( RW):
              bridge ( RO): xenbr0
```

6. The Internet would be accessible from both the SSVM and CPVM instances by default. Their public IPs will also be directly pingable from the Internet. Please note that these test would work only if your switches and traffic labels are configured correctly for your environment. If your SSVM/CPVM cant reach the Internet, its very unlikely that the Virtual Router (VR) can also the reach the Internet suggesting that its either a switching issue or incorrectly assigned traffic labels. Fix the SSVM/CPVM issues before you debug VR issues.

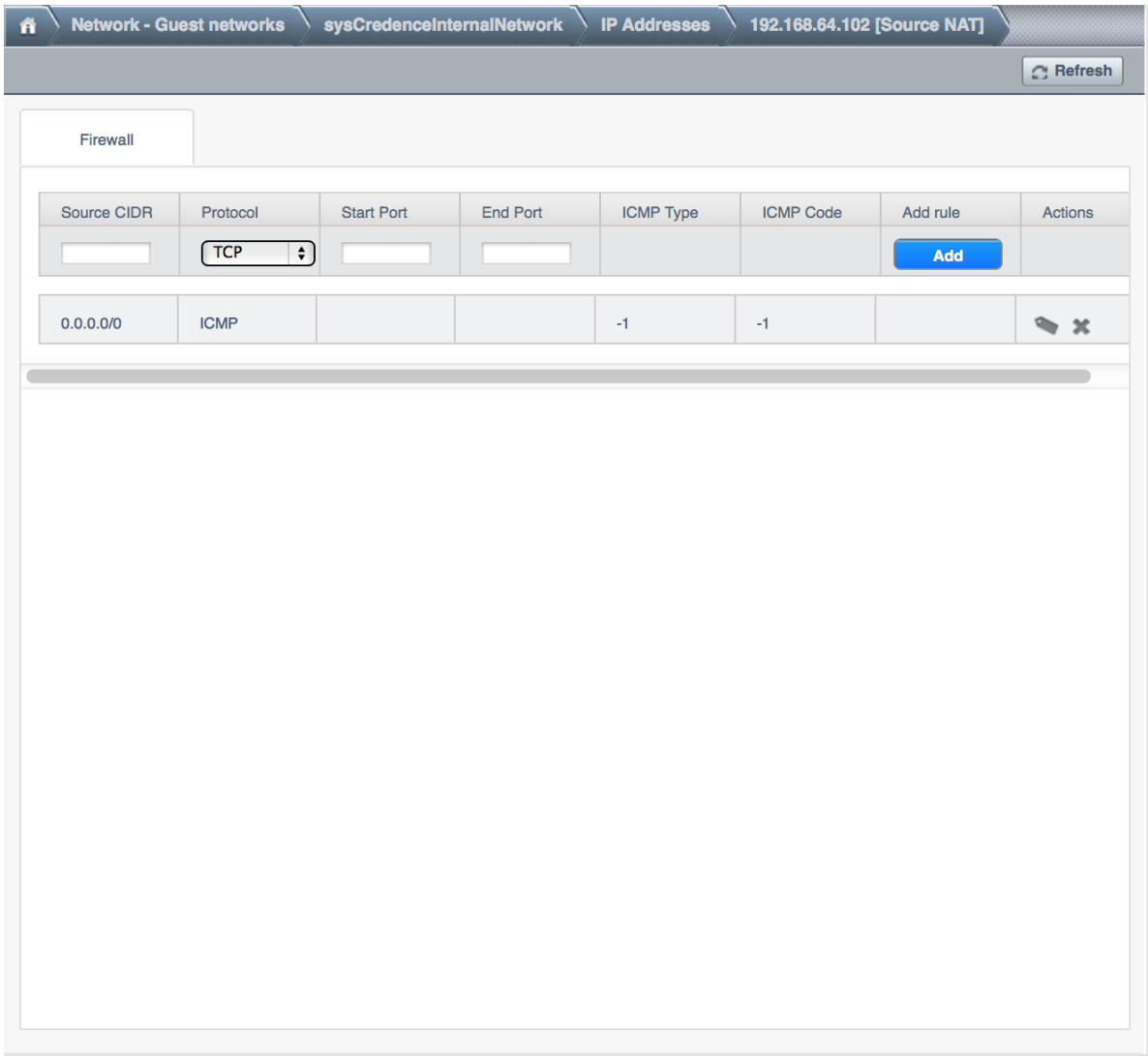
```
root@s-1-VM:~# ping -c 3 google.com
PING google.com (74.125.236.164): 56 data bytes
64 bytes from 74.125.236.164: icmp_seq=0 ttl=55 time=26.932 ms
64 bytes from 74.125.236.164: icmp_seq=1 ttl=55 time=29.156 ms
64 bytes from 74.125.236.164: icmp_seq=2 ttl=55 time=25.000 ms
--- google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 25.000/27.029/29.156/1.698 ms
```

```
root@v-2-VM:~# ping -c 3 google.com
PING google.com (74.125.236.164): 56 data bytes
64 bytes from 74.125.236.164: icmp_seq=0 ttl=55 time=32.125 ms
64 bytes from 74.125.236.164: icmp_seq=1 ttl=55 time=26.324 ms
64 bytes from 74.125.236.164: icmp_seq=2 ttl=55 time=37.001 ms
--- google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 26.324/31.817/37.001/4.364 ms
```

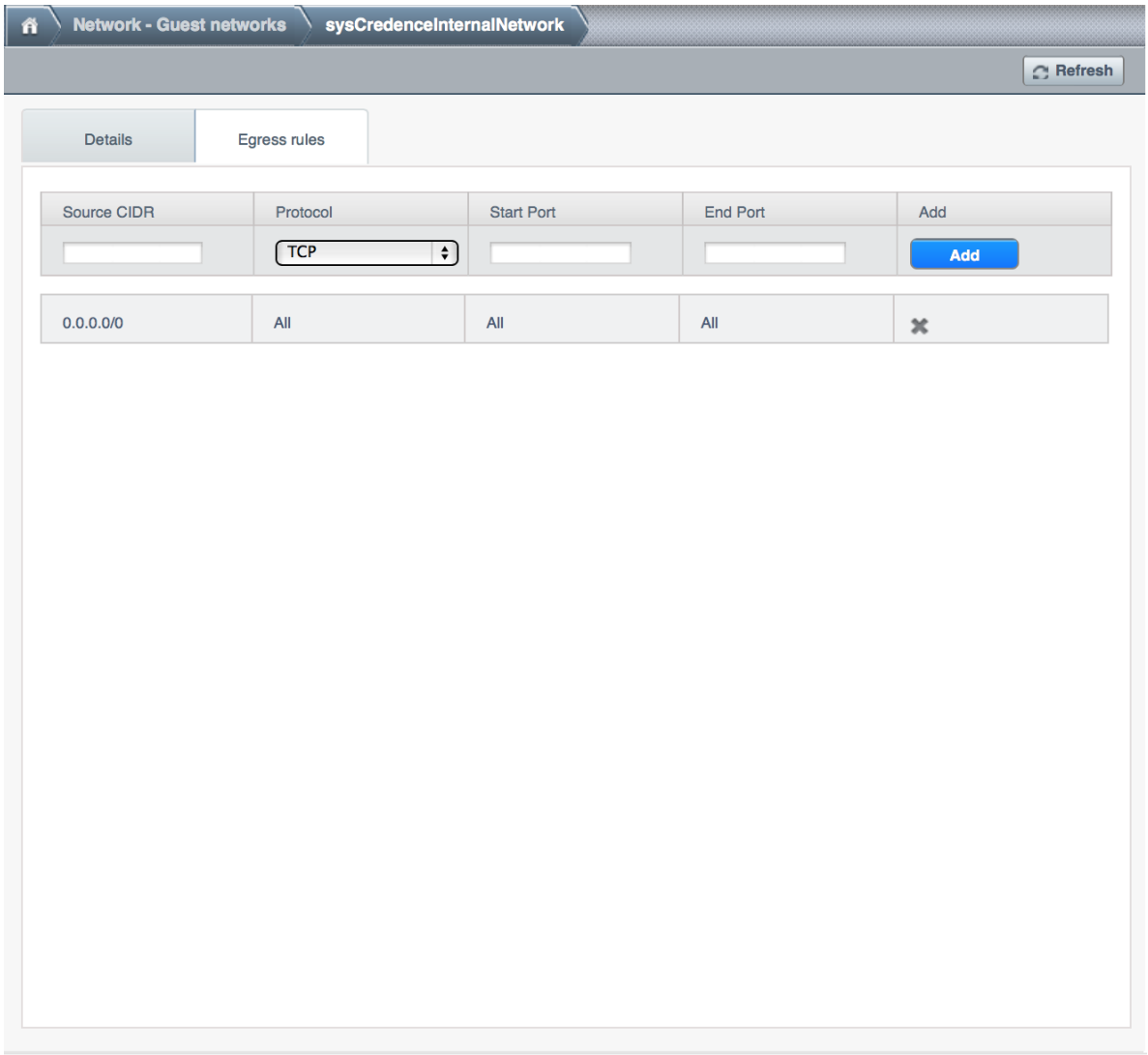
7. The Virtual Router (VR) should also be able to reach the Internet without having any Egress rules. The Egress rules only control forwarded traffic and not traffic that originates on the VR itself.

```
root@r-4-VM:~# ping -c 3 google.com
PING google.com (74.125.236.164): 56 data bytes
64 bytes from 74.125.236.164: icmp_seq=0 ttl=55 time=28.098 ms
64 bytes from 74.125.236.164: icmp_seq=1 ttl=55 time=34.785 ms
64 bytes from 74.125.236.164: icmp_seq=2 ttl=55 time=69.179 ms
--- google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.098/44.021/69.179/17.998 ms
```

8. However, the Virtual Router's (VR) Source NAT Public IP address **WONT** be reachable until appropriate Ingress rules are in place. You can add *Ingress* rules under *Network, Guest Network, IP Address, Firewall* setting page.



9. The VM Instances by default wont be able to access the Internet. Add Egress rules to permit traffic.



10. Some users have reported that flushing IPTables rules (or changing routes) on the SSVM, CPVM or the Virtual Router makes the Internet work. This is not expected behaviour and suggests that your networking settings are incorrect. No IPtables/route changes are required on the SSVM, CPVM or the VR. Go back and double check all your settings.

In a vast majority of the cases, the problem has turned out to be at the switching layer where the L3 switches were configured incorrectly.

This section was contibuted by Shanker Balan and was originally published on Shapeblue's blog

CHAPTER 6

Developers Guide

This is the Apache CloudStack developers guide. This section gives information for those wishing to develop CloudStack either contributing to the CloudStack core software or writing external plugins. Futher information can also be found at CloudStack's wiki https://cwiki.apache.org/confluence/display/CLOUDSTACK/Home and on the CloudStack mailing lists http://cloudstack.apache.org/mailing-lists.html

6.1 CloudStack Installation from GIT repo for Developers

This book is aimed at CloudStack developers who need to build the code. These instructions are valid on a Ubuntu 12.04 and CentOS 6.4 systems and were tested with the 4.2 release of Apache CloudStack, please adapt them if you are on a different operating system or using a newer/older version of CloudStack. This book is composed of the following sections:

1. Installation of the prerequisites
2. Compiling and installation from source
3. Using the CloudStack simulator
4. Installation with DevCloud the CloudStack sandbox
5. Building your own packages
6. The CloudStack API
7. Testing the AWS API interface

6.1.1 Prerequisites

In this section we'll look at installing the dependencies you'll need for Apache CloudStack development.

On Ubuntu 12.04

First update and upgrade your system:

```
apt-get update
apt-get upgrade
```

NTP might already be installed, check it with `service ntp status`. If it's not then install NTP to synchronize the clocks:

```
apt-get install openntpd
```

Install `openjdk`. As we're using Linux, OpenJDK is our first choice.

```
apt-get install openjdk-7-jdk
```

Install `tomcat6`, note that the new version of tomcat on Ubuntu is the 6.0.35 version.

```
apt-get install tomcat6
```

Next, we'll install MySQL if it's not already present on the system.

```
apt-get install mysql-server
```

Remember to set the correct `mysql` password in the CloudStack properties file. Mysql should be running but you can check it's status with:

```
service mysql status
```

Developers wanting to build CloudStack from source will want to install the following additional packages. If you dont' want to build from source just jump to the next section.

Install `git` to later clone the CloudStack source code:

```
apt-get install git
```

Install `Maven` to later build CloudStack

```
apt-get install maven
```

This should have installed Maven 3.0, check the version number with `mvn --version`

A little bit of Python can be used (e.g simulator), install the Python package management tools:

```
apt-get install python-pip python-setuptools
```

Finally install `mkisofs` with:

```
apt-get install genisoimage
```

On CentOS 6.4

First update and upgrade your system:

```
yum -y update
yum -y upgrade
```

If not already installed, install NTP for clock synchornization

```
yum -y install ntp
```

Install `openjdk`. As we're using Linux, OpenJDK is our first choice.

```
yum -y install java-1.7.0-openjdk-devel
```

Install `tomcat6`, note that the version of tomcat6 in the default CentOS 6.4 repo is 6.0.24, so we will grab the 6.0.35 version. The 6.0.24 version will be installed anyway as a dependency to cloudstack.

```
wget https://archive.apache.org/dist/tomcat/tomcat-6/v6.0.35/bin/apache-tomcat-6.0.35.
↪tar.gz
tar xzvf apache-tomcat-6.0.35.tar.gz -C /usr/local
```

Setup tomcat6 system wide by creating a file `/etc/profile.d/tomcat.sh` with the following content:

```
export CATALINA_BASE=/usr/local/apache-tomcat-6.0.35
export CATALINA_HOME=/usr/local/apache-tomcat-6.0.35
```

Next, we'll install MySQL if it's not already present on the system.

```
yum -y install mysql mysql-server
```

Remember to set the correct `mysql` password in the CloudStack properties file. Mysql should be running but you can check it's status with:

```
service mysqld status
```

Install `git` to later clone the CloudStack source code:

```
yum -y install git
```

Install `Maven` to later build CloudStack. Grab the 3.0.5 release from the Maven website

```
wget http://mirror.cc.columbia.edu/pub/software/apache/maven/maven-3/3.0.5/binaries/
↪apache-maven-3.0.5-bin.tar.gz
tar xzf apache-maven-3.0.5-bin.tar.gz -C /usr/local
cd /usr/local
ln -s apache-maven-3.0.5 maven
```

Setup Maven system wide by creating a `/etc/profile.d/maven.sh` file with the following content:

```
export M2_HOME=/usr/local/maven
export PATH=${M2_HOME}/bin:${PATH}
```

Log out and log in again and you will have maven in your PATH:

```
mvn --version
```

This should have installed Maven 3.0, check the version number with `mvn --version`

A little bit of Python can be used (e.g simulator), install the Python package management tools:

```
yum -y install python-setuptools
```

To install python-pip you might want to setup the Extra Packages for Enterprise Linux (EPEL) repo

```
cd /tmp
wget http://mirror-fpt-telecom.fpt.net/fedora/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
```

Then update you repository cache `yum update` and install pip `yum -y install python-pip`

Finally install `mkisofs` with:

```
yum -y install genisoimage
```

6.1.2 Installing from Source

CloudStack uses git for source version control, if you know little about git is a good start. Once you have git setup on your machine, pull the source with:

```
git clone https://git-wip-us.apache.org/repos/asf/cloudstack.git
```

To build the latest stable release:

```
git checkout 4.8
```

To compile Apache CloudStack, go to the cloudstack source folder and run:

```
mvn -Pdeveloper,systemvm clean install
```

If you want to skip the tests add `-DskipTests` to the command above. Do NOT use `-Dmaven.test.skip=true` because that will break the build.

You will have made sure to set the proper db password in `utils/conf/db.properties`

Deploy the database next:

```
mvn -P developer -pl developer -Ddeploydb
```

Run Apache CloudStack with jetty for testing. Note that `tomcat` maybe be running on port 8080, stop it before you use `jetty`

```
mvn -pl :cloud-client-ui jetty:run
```

Log Into Apache CloudStack:

Open your Web browser and use this URL to connect to CloudStack:

```
http://localhost:8080/client/
```

Replace `localhost` with the IP of your management server if need be.

Note: If you have iptables enabled, you may have to open the ports used by CloudStack. Specifically, ports 8080, 8250, and 9090.

You can now start configuring a Zone, playing with the API. Of course we did not setup any infrastructure, there is no storage, no hypervisors…etc. However you can run tests using the simulator. The following section shows you how to use the simulator so that you don't have to setup a physical infrastructure.

6.1.3 Using the Simulator

CloudStack comes with a simulator based on Python bindings called *Marvin*. Marvin is available in the CloudStack source code or on Pypi. With Marvin you can simulate your data center infrastructure by providing CloudStack with a configuration file that defines the number of zones/pods/clusters/hosts, types of storage etc. You can then develop and test the CloudStack management server *as if* it was managing your production infrastructure.

Do a clean build:

```
mvn -Pdeveloper -Dsimulator -DskipTests clean install
```

Deploy the database:

```
mvn -Pdeveloper -pl developer -Ddeploydb
mvn -Pdeveloper -pl developer -Ddeploydb-simulator
```

Install marvin. Note that you will need to have installed `pip` properly in the prerequisites step.

```
pip install tools/marvin/dist/Marvin-|release|.tar.gz
```

Stop jetty (from any previous runs)

```
mvn -pl :cloud-client-ui jetty:stop
```

Start jetty

```
mvn -pl client jetty:run
```

Setup a basic zone with Marvin. In a separate shell://

```
mvn -Pdeveloper,marvin.setup -Dmarvin.config=setup/dev/basic.cfg -pl :cloud-marvin␣
↪integration-test
```

At this stage log in the CloudStack management server at http://localhost:8080/client with the credentials admin/password, you should see a fully configured basic zone infrastructure. To simulate an advanced zone replace `basic.cfg` with `advanced.cfg`.

You can now run integration tests, use the API etc…

6.1.4 Using DevCloud

The Installing from source section will only get you to the point of runnign the management server, it does not get you any hypervisors. The simulator section gets you a simulated datacenter for testing. With DevCloud you can run at least one hypervisor and add it to your management server the way you would a real physical machine.

DevCloud is the CloudStack sandbox, the standard version is a VirtualBox based image. There is also a KVM based image for it. Here we only show steps with the VirtualBox image. For KVM see the wiki.

** DevCloud Pre-requisites

1. Install VirtualBox on your machine
2. Run VirtualBox and under >Preferences create a *host-only interface* on which you disable the DHCP server
3. Download the DevCloud image
4. In VirtualBox, under File > Import Appliance import the DevCloud image.
5. Verify the settings under > Settings and check the `enable PAE` option in the processor menu

6. Once the VM has booted try to `ssh` to it with credentials: `root/password`

   ssh root@192.168.56.10

Adding DevCloud as an Hypervisor

Picking up from a clean build:

```
mvn -Pdeveloper,systemvm clean install
mvn -P developer -pl developer,tools/devcloud -Ddeploydb
```

At this stage install marvin similarly than with the simulator:

```
pip install tools/marvin/dist/Marvin-|release|.tar.gz
```

Start the management server

```
mvn -pl client jetty:run
```

Then you are going to configure CloudStack to use the running DevCloud instance:

```
cd tools/devcloud
python ../marvin/marvin/deployDataCenter.py -i devcloud.cfg
```

If you are curious, check the `devcloud.cfg` file and see how the data center is defined: 1 Zone, 1 Pod, 1 Cluster, 1 Host, 1 primary Storage, 1 Seondary Storage, all provided by Devcloud.

You can now log in the management server at `http://localhost:8080/client` and start experimenting with the UI or the API.

Do note that the management server is running in your local machine and that DevCloud is used only as a n Hypervisor. You could potentially run the management server within DevCloud as well, or memory granted, run multiple DevClouds.

6.1.5 Building Packages

Working from source is necessary when developing CloudStack. As mentioned earlier this is not primarily intended for users. However some may want to modify the code for their own use and specific infrastructure. The may also need to build their own packages for security reasons and due to network connectivity constraints. This section shows you the gist of how to build packages. We assume that the reader will know how to create a repository to serve this packages. The complete documentation is available in the *Building DEB packages* section.

To build debian packages you will need couple extra packages that we did not need to install for source compilation:

```
apt-get install python-mysqldb
apt-get install debhelper
```

Then build the packages with:

```
dpkg-buildpackage -uc -us
```

One directory up from the CloudStack root dir you will find:

```
cloudstack_|release|_amd64.changes
cloudstack_|release|.dsc
cloudstack_|release|.tar.gz
```

(continues on next page)

(continued from previous page)

```
cloudstack-agent_|release|_all.deb
cloudstack-awsapi_|release|_all.deb
cloudstack-cli_|release|_all.deb
cloudstack-common_|release|_all.deb
cloudstack-docs_|release|_all.deb
cloudstack-management_|release|_all.deb
cloudstack-usage_|release|_all.deb
```

Of course the community provides a repository for these packages and you can use it instead of building your own packages and putting them in your own repo. Instructions on how to use this community repository are available in the installation book.

6.1.6 The CloudStack API

The CloudStack API is a query based API using http that return results in XML or JSON. It is used to implement the default web UI. This API is not a standard like OGF OCCI or DMTF CIMI but is easy to learn. Mapping exists between the AWS API and the CloudStack API as will be seen in the next section. Recently a Google Compute Engine interface was also developed that maps the GCE REST API to the CloudStack API described here. The API docs are a good start to learn the extent of the API. Multiple clients exist on github to use this API, you should be able to find one in your favorite language. The reference documentation for the API and changes that might occur from version to version is availble on-line. This short section is aimed at providing a quick summary to give you a base understanding of how to use this API. As a quick start, a good way to explore the API is to navigate the dashboard with a firebug console (or similar developer console) to study the queries.

In a succint statement, the CloudStack query API can be used via http GET requests made against your cloud endpoint (e.g http://localhost:8080/client/api). The API name is passed using the `command` key and the various parameters for this API call are passed as key value pairs. The request is signed using the access key and secret key of the user making the call. Some calls are synchronous while some are asynchronous, this is documented in the API docs. Asynchronous calls return a `jobid`, the status and result of a job can be queried with the `queryAsyncJobResult` call. Let's get started and give an example of calling the `listUsers` API in Python.

First you will need to generate keys to make requests. Going through the dashboard, go under `Accounts` select the appropriate account then click on `Show Users` select the intended users and generate keys using the `Generate Keys` icon. You will see an `API Key` and `Secret Key` field being generated. The keys will be of the form:

```
API Key : XzAz0uC0t888gOzPs3HchY72qwDc7pUPIO8LxC-VkIHo4C3fvbEBY_Ccj8fo3mBapN5qRDg_0_
↪EbGdbxi8oy1A
Secret Key: zmBOXAXPlfb-LIygOxUVblAbz7E47eukDS_0JYUxP3JAmknOYo56T0R-
↪AcM7rK7SMyo11Y6XW22gyuXzOdiybQ
```

Open a Python shell and import the basic modules necessary to make the request. Do note that this request could be made many different ways, this is just a low level example. The `urllib*` modules are used to make the http request and do url encoding. The `hashlib` module gives us the sha1 hash function. It used to geenrate the `hmac` (Keyed Hashing for Message Authentication) using the secretkey. The result is encoded using the `base64` module.

```
$python
Python 2.7.3 (default, Nov 17 2012, 19:54:34)
[GCC 4.2.1 Compatible Apple Clang 4.1 ((tags/Apple/clang-421.11.66))] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import urllib2
>>> import urllib
>>> import hashlib
>>> import hmac
>>> import base64
```

Define the endpoint of the Cloud, the command that you want to execute, the type of the response (i.e XML or JSON) and the keys of the user. Note that we do not put the secretkey in our request dictionary because it is only used to compute the hmac.

```
>>> baseurl='http://localhost:8080/client/api?'
>>> request={}
>>> request['command']='listUsers'
>>> request['response']='json'
>>> request['apikey']='plgWJfZK4gyS3mOMTVmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
↪kAYMmAiLqzzq1ElZLYq_u38zCm0bewzGUdP66mg'
>>> secretkey='VDaACYb0LV9eNjTetIOElcVQkvJck_J_QljX_
↪FcHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3akA3KBQ'
```

Build the base request string, the combination of all the key/pairs of the request, url encoded and joined with ampersand.

```
>>> request_str='&'.join(['='.join([k,urllib.quote_plus(request[k])]) for k in␣
↪request.keys()])
>>> request_str
'apikey=plgWJfZK4gyS3mOMTVmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-kAYMmAiLqzzq1ElZLYq_
↪u38zCm0bewzGUdP66mg&command=listUsers&response=json'
```

Compute the signature with hmac, do a 64 bit encoding and a url encoding, the string used for the signature is similar to the base request string shown above but the keys/values are lower cased and joined in a sorted order

```
>>> sig_str='&'.join(['='.join([k.lower(),urllib.quote_plus(request[k].lower().
↪replace('+','%20'))])for k in sorted(request.iterkeys())])
>>> sig_str
'apikey=plgwjfzk4gys3momtvmjuvg-x-jlwlnfauj9gabbbf9edm-kaymmailqzzq1elzlyq_
↪u38zcm0bewzgudp66mg&command=listusers&response=json'
>>> sig=hmac.new(secretkey,sig_str,hashlib.sha1).digest()
>>> sig
'M:]\x0e\xaf\xfb\x8f\xf2y\xf1p\x91\x1e\x89\x8a\xa1\x05\xc4A\xdb'
>>> sig=base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).digest())
>>> sig
'TTpdDq/7j/J58XCRHomKoQXEQds=\n'
>>> sig=base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).digest()).strip()
>>> sig
'TTpdDq/7j/J58XCRHomKoQXEQds='
>>> sig=urllib.quote_plus(base64.encodestring(hmac.new(secretkey,sig_str,hashlib.
↪sha1).digest()).strip())
```

Finally, build the entire string by joining the baseurl, the request str and the signature. Then do an http GET:

```
>>> req=baseurl+request_str+'&signature='+sig
>>> req
'http://localhost:8080/client/api?apikey=plgWJfZK4gyS3mOMTVmjUVg-X-
↪jlWlnfaUJ9GAbBbf9EdM-kAYMmAiLqzzq1ElZLYq_u38zCm0bewzGUdP66mg&command=listUsers&
↪response=json&signature=TTpdDq%2F7j%2FJ58XCRHomKoQXEQds%3D'
>>> res=urllib2.urlopen(req)
>>> res.read()
{
   "listusersresponse" : {
      "count":1 ,
      "user" : [
         {
            "id":"7ed6d5da-93b2-4545-a502-23d20b48ef2a",
            "username":"admin",
```

(continues on next page)

(continued from previous page)

```
            "firstname":"admin",
            "lastname":"cloud",
            "created":"2012-07-05T12:18:27-0700",
            "state":"enabled",
            "account":"admin",
            "accounttype":1,
            "domainid":"8a111e58-e155-4482-93ce-84efff3c7c77",
            "domain":"ROOT",
            "apikey":"plgWJfZK4gyS3mOMTVmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
↪kAYMmAiLqzzq1ElZLYq_u38zCm0bewzGUdP66mg",
            "secretkey":"VDaACYb0LV9eNjTetIOElcVQkvJck_J_QljX_
↪FcHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3akA3KBQ",
            "accountid":"7548ac03-af1d-4c1c-9064-2f3e2c0eda0d"
        }
    ]
  }
}
```

All the clients that you will find on github will implement this signature technique, you should not have to do it by hand. Now that you have explored the API through the UI and that you understand how to make low level calls, pick your favorite client of use CloudMonkey. CloudMonkey is a sub-project of Apache CloudStack and gives operators/developers the ability to use any of the API methods. It has nice auto-completion and help feature as well as an API discovery mechanism since 4.2.

6.1.7 Testing the AWS API interface

While the native CloudStack API is not a standard, CloudStack provides a AWS EC2 compatible interface. It has the great advantage that existing tools written with EC2 libraries can be re-used against a CloudStack based cloud. In the installation books we described how to run this interface from installing packages. In this section we show you how to compile the interface with `maven` and test it with Python boto module.

Starting from a running management server (with DevCloud for instance), start the AWS API interface in a separate shell with:

```
mvn -Pawsapi -pl :cloud-awsapi jetty:run
```

Log into the CloudStack UI `http://localhost:8080/client`, go to *Service Offerings* and edit one of the compute offerings to have the name `m1.small` or any of the other AWS EC2 instance types.

With access and secret keys generated for a user you should now be able to use Python Boto module:

```
import boto
import boto.ec2

accesskey="2IUSA5xylbsPSnBQFoWXKg3RvjHgsufcKhC1SeiCbeEc0obKwUlwJamB_
↪gFmMJkFHYHTIafpUx0pHcfLvt-dzw"
secretkey=
↪"oxV5Dhhk5ufNowey7OVHgWxCBVS4deTl9qL0EqMthfPBuy3ScHPo2fifDxw1aXeL5cyH10hnLOKjyKphcXGeDA
↪"

region = boto.ec2.regioninfo.RegionInfo(name="ROOT", endpoint="localhost")
conn = boto.connect_ec2(aws_access_key_id=accesskey, aws_secret_access_key=secretkey,␣
↪is_secure=False, region=region, port=7080, path="/awsapi", api_version="2012-08-15")

images=conn.get_all_images()
```

(continues on next page)

(continued from previous page)

```
print images

res = images[0].run(instance_type='m1.small',security_groups=['default'])
```

Note the new `api_version` number in the connection object and also note that there was no user registration to make like in previous CloudStack releases.

6.1.8 Conclusions

CloudStack is a mostly Java application running with Tomcat and Mysql. It consists of a management server and depending on the hypervisors being used, an agent installed on the hypervisor farm. To complete a Cloud infrastructure however you will also need some Zone wide storage a.k.a Secondary Storage and some Cluster wide storage a.k.a Primary storage. The choice of hypervisor, storage solution and type of Zone (i.e Basic vs. Advanced) will dictate how complex your installation can be. As a quick start, you might want to consider KVM+NFS and a Basic Zone.

If you've run into any problems with this, please ask on the cloudstack-dev mailing list.

6.2 Programmer Guide

This guide shows how to develop CloudStack, use the API for operation and integration, access the usage data and use CloudStack specific tools to ease development, testing and integration.

6.2.1 The CloudStack API

Getting Started

To get started using the CloudStack API, you should have the following:

- URL of the CloudStack server you wish to integrate with.
- Both the API Key and Secret Key for an account. This should have been generated by the administrator of the cloud instance and given to you.
- Familiarity with HTTP GET/POST and query strings.
- Knowledge of either XML or JSON.
- Knowledge of a programming language that can generate HTTP requests; for example, Java or PHP.

Roles

The CloudStack API supports three access roles:

1. Root Admin. Access to all features of the cloud, including both virtual and physical resource management.
2. Domain Admin. Access to only the virtual resources of the clouds that belong to the administrator's domain.
3. User. Access to only the features that allow management of the user's virtual instances, storage, and network.

API Reference Documentation

You can find all the API reference documentation at the below site:

http://cloudstack.apache.org/docs/api/

Making API Requests

All CloudStack API requests are submitted in the form of a HTTP GET/POST with an associated command and any parameters. A request is composed of the following whether in HTTP or HTTPS:

- CloudStack API URL: This is the web services API entry point(for example, http://www.example.com:8080/client/api)
- Command: The web services command you wish to execute, such as start a virtual machine or create a disk volume
- Parameters: Any additional required or optional parameters for the command

A sample API GET request looks like the following:

```
http://localhost:8080/client/api?command=deployVirtualMachine&serviceOfferingId=1&
↪diskOfferingId=1&templateId=2&zoneId=4&apiKey=miVr6X7u6bN_sdahOBpjNejPgEsT35eXq-
↪jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU
↪%2FcaiK8RAP0O1hU%3D
```

Or in a more readable format:

```
1. http://localhost:8080/client/api
2. ?command=deployVirtualMachine
3. &serviceOfferingId=1
4. &diskOfferingId=1
5. &templateId=2
6. &zoneId=4
7. &apiKey=miVr6X7u6bN_sdahOBpjNejPgEsT35eXqjB8CG20YI3yaxXcgpyuaIRmFI_
↪EJTVwZ0nUkkJbPmY3y2bciKwFQ
8. &signature=Lxx1DM40AjcXU%2FcaiK8RAP0O1hU%3D
```

The first line is the CloudStack API URL. This is the Cloud instance you wish to interact with.

The second line refers to the command you wish to execute. In our example, we are attempting to deploy a fresh new virtual machine. It is preceded by a (?) to separate itself from the CloudStack API URL.

Lines 3-6 are the parameters for this given command. To see the command and its request parameters, please refer to the appropriate section in the CloudStack API documentation. Each parameter field-value pair (field=value) is preceded by an ampersand character (&).

Line 7 is the user API Key that uniquely identifies the account. See Signing API Requests on page 7.

Line 8 is the signature hash created to authenticate the user account executing the API command.

Signing API Requests

Whether you access the CloudStack API with HTTP or HTTPS, it must still be signed so that CloudStack can verify the caller has been authenticated and authorized to execute the command. Make sure that you have both the API Key and Secret Key provided by the CloudStack administrator for your account before proceeding with the signing process.

To show how to sign a request, we will re-use the previous example.

```
http://http://localhost:8080/client/api?command=deployVirtualMachine&
↪serviceOfferingId=1&diskOfferingId=1&templateId=2&zoneId=4&apiKey=miVr6X7u6bN_
↪sdahOBpjNejPgEsT35eXq-jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&
↪signature=Lxx1DM40AjcXU%2FcaiK8RAP0O1hU%3D
```

Breaking this down, we have several distinct parts to this URL.

- Base URL: This is the base URL to the CloudStack Management Server.

  ```
  http://localhost:8080
  ```

- API Path: This is the path to the API Servlet that processes the incoming requests.

  ```
  /client/api?
  ```

- Command String: This part of the query string comprises of the command, its parameters, and the API Key that identifies the account.

  Note: As with all query string parameters of field-value pairs, the "field" component is case insensitive while all "value" values are case sensitive.

- Signature: This is the signature of the command string that is generated using a combination of the user's Secret Key and the HMAC SHA-1 hashing algorithm.

  ```
  &signature=Lxx1DM40AjcXU%2FcaiK8RAP0O1hU%3D
  ```

Every API request has the format Base URL+API Path+Command String+Signature.

To generate the signature.

1. For each field-value pair (as separated by a '&') in the Command String, URL encode each value so that it can be safely sent via HTTP GET.

   Note: Make sure all spaces are encoded as "%20" rather than "+".

2. Lower case the entire Command String and sort it alphabetically via the field for each field-value pair. The result of this step would look like the following.

   ```
   apikey=mivr6x7u6bn_sdahobpjnejpgest35exq-jb8cg20yi3yaxxcgpyuairmfi_
   ↪ejtvwz0nukkjbpmy3y2bcikwfq&command=deployvirtualmachine&diskofferingid=1&
   ↪serviceofferingid=1&templateid=2&zoneid=4
   ```

3. Take the sorted Command String and run it through the HMAC SHA-1 hashing algorithm (most programming languages offer a utility method to do this) with the user's Secret Key. Base64 encode the resulting byte array in UTF-8 so that it can be safely transmitted via HTTP. The final string produced after Base64 encoding should be "Lxx1DM40AjcXU%2FcaiK8RAP0O1hU%3D".

   By reconstructing the final URL in the format Base URL+API Path+Command String+Signature, the final URL should look like:

   ```
   http://localhost:8080/client/api?command=deployVirtualMachine&serviceOfferingId=1&
   ↪diskOfferingId=1&templateId=2&zoneId=4&apiKey=miVr6X7u6bN_sdahOBpjNejPgEsT35eXq-
   ↪jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU
   ↪%2FcaiK8RAP0O1hU%3D
   ```

How to sign an API call with Python

To illustrate the procedure used to sign API calls we present a step by step interactive session using Python.

First import the required modules:

```
$python
Python 2.7.3 (default, Nov 17 2012, 19:54:34)
[GCC 4.2.1 Compatible Apple Clang 4.1 ((tags/Apple/clang-421.11.66))] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import urllib2
>>> import urllib
>>> import hashlib
>>> import hmac
>>> import base64
```

Define the endpoint of the Cloud, the command that you want to execute and the keys of the user.

```
>>> baseurl='http://localhost:8080/client/api?'
>>> request={}
>>> request['command']='listUsers'
>>> request['response']='json'
>>> request['apikey']='plgWJfZK4gyS3mOMTVmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
↪kAYMmAiLqzzq1ElZLYq_u38zCm0bewzGUdP66mg'
>>> secretkey='VDaACYb0LV9eNjTetIOElcVQkvJck_J_QljX_
↪FcHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3akA3KBQ'
```

Build the request string:

```
>>> request_str='&'.join(['='.join([k,urllib.quote_plus(request[k])]) for k in␣
↪request.keys()])
>>> request_str
'apikey=plgWJfZK4gyS3mOMTVmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-kAYMmAiLqzzq1ElZLYq_
↪u38zCm0bewzGUdP66mg&command=listUsers&response=json'
```

Compute the signature with hmac, do a 64 bit encoding and a url encoding:

```
>>> sig_str='&'.join(['='.join([k.lower(),urllib.quote_plus(request[k].
↪lower().replace('+','%20'))])for k in sorted(request.iterkeys())])
>>> sig_str 'apikey=plgwjfzk4gys3momtvmjuvg-x-jlwlnfauj9gabbbf9edm-
↪kaymmailqzzq1elzlyq_u38zcm0bewzgudp66mg&command=listusers&response=json'
>>> sig=hmac.new(secretkey,sig_str,hashlib.sha1)
>>> sig
<hmac.HMAC instance at 0x10d91d680>
>>> sig=hmac.new(secretkey,sig_str,hashlib.sha1).digest()
>>> sig
'M:]x0exafxfbx8fxf2yxf1px91x1ex89x8axa1x05xc4Axdb'
>>> sig=base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).digest())
>>> sig
'TTpdDq/7j/J58XCRHomKoQXEQds=n'
>>> sig=base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).
↪digest()).strip()
>>> sig
'TTpdDq/7j/J58XCRHomKoQXEQds='
>>> sig=urllib.quote_plus(base64.encodestring(hmac.new(secretkey,sig_str,
↪hashlib.sha1).digest()).strip())
```

Finally, build the entire string and do an http GET:

```
>>> req=baseurl+request_str+'&signature='+sig
>>> req
'http://localhost:8080/client/api?apikey=plgWJfZK4gyS3mOMTVmjUVg-X-
↪jlWlnfaUJ9GAbBbf9EdM-kAYMmAiLqzzq1ElZLYq_u38zCm0bewzGUdP66mg&command=listUsers&
↪response=json&signature=TTpdDq%2F7j%2FJ58XCRHomKoQXEQds%3D'
>>> res=urllib2.urlopen(req)
>>> res.read()
'{ "listusersresponse" : { "count":3 ,"user" : [  {"id":"7ed6d5da-93b2-4545-a502-
↪23d20b48ef2a","username":"admin","firstname":"admin","lastname":"cloud","created":
↪"2012-07-05T12:18:27-0700","state":"enabled","account":"admin","accounttype":1,
↪"domainid":"8a111e58-e155-4482-93ce-84efff3c7c77","domain":"ROOT","apikey":
↪"plgWJfZK4gyS3mOMTVmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-kAYMmAiLqzzq1ElZLYq_
↪u38zCm0bewzGUdP66mg","secretkey":"VDaACYb0LV9eNjTetIOElcVQkvJck_J_QljX_
↪FcHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3akA3KBQ","accountid":"7548ac03-af1d-4c1c-
↪9064-2f3e2c0eda0d"}, {"id":"1fea6418-5576-4989-a21e-4790787bbee3","username":"runseb
↪","firstname":"foobar","lastname":"goa","email":"joe@smith.com","created":"2013-04-
↪10T16:52:06-0700","state":"enabled","account":"admin","accounttype":1,"domainid":
↪"8a111e58-e155-4482-93ce-84efff3c7c77","domain":"ROOT","apikey":
↪"Xhsb3MewjJQaXXMszRcLvQI9_NPy_
↪UcbDj1QXikkVbDC9MDSPwWdtZ1bUY1H7JBEYTtDDLY3yuchCeW778GkBA","secretkey":
↪"gIsgmi8C5YwxMHjX5o51pSe0kqs6JnKriw0jJBLceY5bgnfzKjL4aM6ctJX-
↪i1ddQIHJLbLJDK9MRzsKk6xZ_w","accountid":"7548ac03-af1d-4c1c-9064-2f3e2c0eda0d"}, {
↪"id":"52f65396-183c-4473-883f-a37e7bb93967","username":"toto","firstname":"john",
↪"lastname":"smith","email":"john@smith.com","created":"2013-04-23T04:27:22-0700",
↪"state":"enabled","account":"admin","accounttype":1,"domainid":"8a111e58-e155-4482-
↪93ce-84efff3c7c77","domain":"ROOT","apikey":"THaA6fFWS_OmvU8od201omxFC8yKNL_
↪Hc5ZCS77LFCJsRzSx48JyZucbUul6XYbEg-ZyXMl_wuEpECzK-wKnow","secretkey":
↪"O5ywpqJorAsEBKR_5jEvrtGHfWL1Y_j1E4Z_
↪iCr8OKCYcsPIOdVcfzjJQ8YqK0a5EzSpoRrjOFiLsG0hQrYnDA","accountid":"7548ac03-af1d-4c1c-
↪9064-2f3e2c0eda0d"} ] } }'
```

Enabling API Call Expiration

You can set an expiry timestamp on API calls to prevent replay attacks over non-secure channels, such as HTTP. The server tracks the expiry timestamp you have specified and rejects all the subsequent API requests that come in after this validity period.

To enable this feature, add the following parameters to the API request:

- signatureVersion=3: If the signatureVersion parameter is missing or is not equal to 3, the expires parameter is ignored in the API request.
- expires=YYYY-MM-DDThh:mm:ssZ: Specifies the date and time at which the signature included in the request is expired. The timestamp is expressed in the YYYY-MM-DDThh:mm:ssZ format, as specified in the ISO 8601 standard.

For example:

```
expires=2011-10-10T12:00:00+0530
```

A sample API request with expiration is given below:

```
http://<IPAddress>:8080/client/api?command=listZones&signatureVersion=3&expires=2011-
↪10-10T12:00:00+0530&apiKey=miVr6X7u6bN_sdahOBpjNejPgEsT35eXq-
↪jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU
↪%2FcaiK8RAP0O1hU%3D
```

Limiting the Rate of API Requests

You can limit the rate at which API requests can be placed for each account. This is useful to avoid malicious attacks on the Management Server, prevent performance degradation, and provide fairness to all accounts.

If the number of API calls exceeds the threshold, an error message is returned for any additional API calls. The caller will have to retry these API calls at another time.

Configuring the API Request Rate

To control the API request rate, use the following global configuration settings:

- *api.throttling.enabled* - Enable/Disable API throttling. By default, this setting is false, so API throttling is not enabled.
- *api.throttling.interval* (in seconds) - Time interval during which the number of API requests is to be counted. When the interval has passed, the API count is reset to 0.
- *api.throttling.max* - Maximum number of APIs that can be placed within the *api.throttling.interval* period.
- *api.throttling.cachesize* - Cache size for storing API counters. Use a value higher than the total number of accounts managed by the cloud. One cache entry is needed for each account, to store the running API total for that account.

Limitations on API Throttling

The following limitations exist in the current implementation of this feature.

Note: Even with these limitations, CloudStack is still able to effectively use API throttling to avoid malicious attacks causing denial of service.

- In a deployment with multiple Management Servers, the cache is not synchronized across them. In this case, CloudStack might not be able to ensure that only the exact desired number of API requests are allowed. In the worst case, the number of API calls that might be allowed is (number of Management Servers) * (api.throttling.max).
- The API commands resetApiLimit and getApiLimit are limited to the Management Server where the API is invoked.

API Responses

Response Formats: XML and JSON

CloudStack supports two formats as the response to an API call. The default response is XML. If you would like the response to be in JSON, add *&response=json* to the Command String.

The two response formats differ in how they handle blank fields. In JSON, if there is no value for a response field, it will not appear in the response. If all the fields were empty, there might be no response at all. In XML, even if there is no value to be returned, an empty field will be returned as a placeholder XML element.

Sample XML Response:

```
<listipaddressesresponse>
   <allocatedipaddress>
   <ipaddress>192.168.10.141</ipaddress>
   <allocated>2009-09-18T13:16:10-0700</allocated>
   <zoneid>4</zoneid>
      <zonename>WC</zonename>
      <issourcenat>true</issourcenat>
   </allocatedipaddress>
</listipaddressesresponse>
```

Sample JSON Response:

```
{ "listipaddressesresponse" :
  { "allocatedipaddress" :
    [
      {
        "ipaddress" : "192.168.10.141",
        "allocated" : "2009-09-18T13:16:10-0700",
        "zoneid" : "4",
        "zonename" : "WC",
        "issourcenat" : "true"
      }
    ]
  }
}
```

Maximum Result Pages Returned

For each cloud, there is a default upper limit on the number of results that any API command will return in a single page. This is to help prevent overloading the cloud servers and prevent DOS attacks. For example, if the page size limit is 500 and a command returns 10,000 results, the command will return 20 pages.

The default page size limit can be different for each cloud. It is set in the global configuration parameter *default.page.size*. If your cloud has many users with lots of VMs, you might need to increase the value of this parameter. At the same time, be careful not to set it so high that your site can be taken down by an enormous return from an API call. For more information about how to set global configuration parameters, see "Describe Your Deployment" in the Installation Guide.

To decrease the page size limit for an individual API command, override the global setting with the page and pagesize parameters, which are available in any list* command (listCapabilities, listDiskOfferings, etc.).

- Both parameters must be specified together.
- The value of the pagesize parameter must be smaller than the value of default.page.size. That is, you can not increase the number of possible items in a result page, only decrease it.

For syntax information on the list* commands, see the API Reference.

Error Handling

If an error occurs while processing an API request, the appropriate response in the format specified is returned. Each error response consists of an error code and an error text describing what possibly can go wrong. Below is a list of possible error codes:

You can now find the CloudStack-specific error code in the exception response for each type of exception. The following list of error codes is added to the new class named CSExceptionErrorCode.

4250 : "com.cloud.utils.exception.CloudRuntimeException"

4255 : "com.cloud.utils.exception.ExceptionUtil"

4260 : "com.cloud.utils.exception.ExecutionException"

4265 : "com.cloud.utils.exception.HypervisorVersionChangedException"

4270 : "com.cloud.utils.exception.RuntimeCloudException"

4275 : "com.cloud.exception.CloudException"

4280 : "com.cloud.exception.AccountLimitException"

4285 : "com.cloud.exception.AgentUnavailableException"

4290 : "com.cloud.exception.CloudAuthenticationException"

4295 : "com.cloud.exception.CloudExecutionException"

4300 : "com.cloud.exception.ConcurrentOperationException"

4305 : "com.cloud.exception.ConflictingNetworkSettingsException"

4310 : "com.cloud.exception.DiscoveredWithErrorException"

4315 : "com.cloud.exception.HAStateException"

4320 : "com.cloud.exception.InsufficientAddressCapacityException"

4325 : "com.cloud.exception.InsufficientCapacityException"

4330 : "com.cloud.exception.InsufficientNetworkCapacityException"

4335 : "com.cloud.exception.InsufficientServerCapacityException"

4340 : "com.cloud.exception.InsufficientStorageCapacityException"

4345 : "com.cloud.exception.InternalErrorException"

4350 : "com.cloud.exception.InvalidParameterValueException"

4355 : "com.cloud.exception.ManagementServerException"

4360 : "com.cloud.exception.NetworkRuleConflictException"

4365 : "com.cloud.exception.PermissionDeniedException"

4370 : "com.cloud.exception.ResourceAllocationException"

4375 : "com.cloud.exception.ResourceInUseException"

4380 : "com.cloud.exception.ResourceUnavailableException"

4385 : "com.cloud.exception.StorageUnavailableException"

4390 : "com.cloud.exception.UnsupportedServiceException"

4395 : "com.cloud.exception.VirtualMachineMigrationException"

4400 : "com.cloud.exception.AccountLimitException"

4405 : "com.cloud.exception.AgentUnavailableException"

4410 : "com.cloud.exception.CloudAuthenticationException"

4415 : "com.cloud.exception.CloudException"

4420 : "com.cloud.exception.CloudExecutionException"

4425 : "com.cloud.exception.ConcurrentOperationException"

4430 : "com.cloud.exception.ConflictingNetworkSettingsException"

4435 : "com.cloud.exception.ConnectionException"

4440 : "com.cloud.exception.DiscoveredWithErrorException"

4445 : "com.cloud.exception.DiscoveryException"

4450 : "com.cloud.exception.HAStateException"

4455 : "com.cloud.exception.InsufficientAddressCapacityException"

4460 : "com.cloud.exception.InsufficientCapacityException"

4465 : "com.cloud.exception.InsufficientNetworkCapacityException"

4470 : "com.cloud.exception.InsufficientServerCapacityException"

4475 : "com.cloud.exception.InsufficientStorageCapacityException"

4480 : "com.cloud.exception.InsufficientVirtualNetworkCapcityException"

4485 : "com.cloud.exception.InternalErrorException"

4490 : "com.cloud.exception.InvalidParameterValueException"

4495 : "com.cloud.exception.ManagementServerException"

4500 : "com.cloud.exception.NetworkRuleConflictException"

4505 : "com.cloud.exception.PermissionDeniedException"

4510 : "com.cloud.exception.ResourceAllocationException"

4515 : "com.cloud.exception.ResourceInUseException"

4520 : "com.cloud.exception.ResourceUnavailableException"

4525 : "com.cloud.exception.StorageUnavailableException"

4530 : "com.cloud.exception.UnsupportedServiceException"

4535 : "com.cloud.exception.VirtualMachineMigrationException"

9999 : "org.apache.cloudstack.api.ServerApiException"

An HTTP error code of 401 is always returned if API request was rejected due to bad signatures, missing API Keys, or the user simply did not have the permissions to execute the command.

Asynchronous Commands

Asynchronous commands were introduced in CloudStack 2.x. Commands are designated as asynchronous when they can potentially take a long period of time to complete such as creating a snapshot or disk volume. They differ from synchronous commands by the following:

- They are identified in the API Reference by an (A).
- They will immediately return a job ID to refer to the job that will be responsible in processing the command.
- If executed as a "create" resource command, it will return the resource ID as well as the job ID.

  You can periodically check the status of the job by making a simple API call to the command, *queryAsyncJobResult* and passing in the job ID.

Job Status

The key to using an asynchronous command is the job ID that is returned immediately once the command has been executed. With the job ID, you can periodically check the job status by making calls to queryAsyncJobResult command. The command will return three possible job status integer values:

- 0 - Job is still in progress. Continue to periodically poll for any status changes.
- 1 - Job has successfully completed. The job will return any successful response values associated with command that was originally executed.
- 2 - Job has failed to complete. Please check the "jobresultcode" tag for failure reason code and "jobresult" for the failure reason.

Example

The following shows an example of using an asynchronous command. Assume the API command:

```
command=deployVirtualMachine&zoneId=1&serviceOfferingId=1&diskOfferingId=1&
↪templateId=1
```

CloudStack will immediately return a job ID and any other additional data.

```
<deployvirtualmachineresponse>
   <jobid>1</jobid>
   <id>100</id>
</deployvirtualmachineresponse>
```

Using the job ID, you can periodically poll for the results by using the queryAsyncJobResult command.

```
command=queryAsyncJobResult&jobId=1
```

Three possible results could come from this query.

Job is still pending:

```
<queryasyncjobresult>
   <jobid>1</jobid>
   <jobstatus>0</jobstatus>
   <jobprocstatus>1</jobprocstatus>
</queryasyncjobresult>
```

Job has succeeded:

```
<queryasyncjobresultresponse cloud-stack-version="3.0.1.6">
   <jobid>1</jobid>
   <jobstatus>1</jobstatus>
   <jobprocstatus>0</jobprocstatus>
   <jobresultcode>0</jobresultcode>
   <jobresulttype>object</jobresulttype>
   <jobresult>
      <virtualmachine>
         <id>450</id>
         <name>i-2-450-VM</name>
         <displayname>i-2-450-VM</displayname>
         <account>admin</account>
         <domainid>1</domainid>
         <domain>ROOT</domain>
```

(continues on next page)

(continued from previous page)

```
        <created>2011-03-10T18:20:25-0800</created>
        <state>Running</state>
        <haenable>false</haenable>
        <zoneid>1</zoneid>
        <zonename>San Jose 1</zonename>
        <hostid>2</hostid>
        <hostname>905-13.sjc.lab.vmops.com</hostname>
        <templateid>1</templateid>
        <templatename>CentOS 5.3 64bit LAMP</templatename>
        <templatedisplaytext>CentOS 5.3 64bit LAMP</templatedisplaytext>
        <passwordenabled>false</passwordenabled>
        <serviceofferingid>1</serviceofferingid>
        <serviceofferingname>Small Instance</serviceofferingname>
        <cpunumber>1</cpunumber>
        <cpuspeed>500</cpuspeed>
        <memory>512</memory>
        <guestosid>12</guestosid>
        <rootdeviceid>0</rootdeviceid>
        <rootdevicetype>NetworkFilesystem</rootdevicetype>
        <nic>
           <id>561</id>
           <networkid>205</networkid>
           <netmask>255.255.255.0</netmask>
           <gateway>10.1.1.1</gateway>
           <ipaddress>10.1.1.225</ipaddress>
           <isolationuri>vlan://295</isolationuri>
           <broadcasturi>vlan://295</broadcasturi>
           <traffictype>Guest</traffictype>
           <type>Virtual</type>
           <isdefault>true</isdefault>
        </nic>
        <hypervisor>XenServer</hypervisor>
      </virtualmachine>
   </jobresult>
</queryasyncjobresultresponse>
```

Job has failed:

```
<queryasyncjobresult>
   <jobid>1</jobid>
   <jobstatus>2</jobstatus>
   <jobprocstatus>0</jobprocstatus>
   <jobresultcode>551</jobresultcode>
   <jobresulttype>text</jobresulttype>
   <jobresult>Unable to deploy virtual machine id = 100 due to not enough capacity</
↪jobresult>
</queryasyncjobresult>
```

6.2.2 Event Types

Types	Events
VM	VM.CREATE VM.DESTROY VM.START VM.STOP VM.REBOOT VM.UPDATE VM.UPGRADE VM.DYNAMIC.SCALE VM.RESETPASSWORD VM.RESETSSHKEY VM.MIGRATE VM.MOVE VM.RESTORE
Domain Router	ROUTER.CREATE ROUTER.DESTROY ROUTER.START ROUTER.STOP ROUTER.REBOOT ROUTER.HA ROUTER.UPGRADE
Console proxy	PROXY.CREATE PROXY.DESTROY PROXY.START PROXY.STOP PROXY.REBOOT PROXY.HA
VNC Console Events	VNC.CONNECT VNC.DISCONNECT
Network Events	NET.IPASSIGN NET.IPRELEASE PORTABLE.IPASSIGN PORTABLE.IPRELEASE NET.RULEADD NET.RULEDELETE NET.RULEMODIFY NETWORK.CREATE NETWORK.DELETE NETWORK.UPDATE FIREWALL.OPEN FIREWALL.CLOSE
NIC Events	NIC.CREATE NIC.DELETE NIC.UPDATE NIC.DETAIL.ADD NIC.DETAIL.UPDATE NIC.DETAIL.REMOVE

Continued on next page

Table 1 – continued from previous page

Types	Events
Load Balancers	LB.ASSIGN.TO.RULE LB.REMOVE.FROM.RULE LB.CREATE LB.DELETE LB.STICKINESSPOLICY.CREATE LB.STICKINESSPOLICY.DELETE LB.HEALTHCHECKPOLICY.CREATE LB.HEALTHCHECKPOLICY.DELETE LB.UPDATE
Global Load Balancer rules	GLOBAL.LB.ASSIGN GLOBAL.LB.REMOVE GLOBAL.LB.CREATE GLOBAL.LB.DELETE GLOBAL.LB.UPDATE
Account events	ACCOUNT.ENABLE ACCOUNT.DISABLE ACCOUNT.CREATE ACCOUNT.DELETE ACCOUNT.UPDATE ACCOUNT.MARK.DEFAULT.ZONE
UserVO Events	USER.LOGIN USER.LOGOUT USER.CREATE USER.DELETE USER.DISABLE USER.UPDATE USER.ENABLE USER.LOCK
Registering SSH keypair events	REGISTER.SSH.KEYPAIR
Register for user API and secret keys	REGISTER.USER.KEY
Template Events	TEMPLATE.CREATE TEMPLATE.DELETE TEMPLATE.UPDATE TEMPLATE.DOWNLOAD.START TEMPLATE.DOWNLOAD.SUCCESS TEMPLATE.DOWNLOAD.FAILED TEMPLATE.COPY TEMPLATE.EXTRACT TEMPLATE.UPLOAD TEMPLATE.CLEANUP
Volume Events	VOLUME.CREATE VOLUME.DELETE VOLUME.ATTACH VOLUME.DETACH VOLUME.EXTRACT VOLUME.UPLOAD VOLUME.MIGRATE VOLUME.RESIZE VOLUME.DETAIL.UPDATE VOLUME.DETAIL.ADD VOLUME.DETAIL.REMOVE

Continued on next page

Table 1 – continued from previous page

Types	Events
Domains	DOMAIN.CREATE DOMAIN.DELETE DOMAIN.UPDATE
Snapshots	SNAPSHOT.CREATE SNAPSHOT.DELETE SNAPSHOTPOLICY.CREATE SNAPSHOTPOLICY.UPDATE SNAPSHOTPOLICY.DELETE
ISO	ISO.CREATE ISO.DELETE ISO.COPY ISO.ATTACH ISO.DETACH ISO.EXTRACT ISO.UPLOAD
SSVM	SSVM.CREATE SSVM.DESTROY SSVM.START SSVM.STOP SSVM.REBOOT SSVM.HA
Service Offerings	SERVICE.OFFERING.CREATE SERVICE.OFFERING.EDIT SERVICE.OFFERING.DELETE
Disk Offerings	DISK.OFFERING.CREATE DISK.OFFERING.EDIT DISK.OFFERING.DELETE
Network offerings	NETWORK.OFFERING.CREATE NETWORK.OFFERING.ASSIGN NETWORK.OFFERING.EDIT NETWORK.OFFERING.REMOVE NETWORK.OFFERING.DELETE
Pods	POD.CREATE POD.EDIT POD.DELETE
Zones	ZONE.CREATE ZONE.EDIT ZONE.DELETE
VLANs/IP ranges	VLAN.IP.RANGE.CREATE VLAN.IP.RANGE.DELETE VLAN.IP.RANGE.DEDICATE VLAN.IP.RANGE.RELEASE STORAGE.IP.RANGE.CREATE STORAGE.IP.RANGE.DELETE STORAGE.IP.RANGE.UPDATE
Configuration Table	CONFIGURATION.VALUE.EDIT

Continued on next page

Table 1 – continued from previous page

Types	Events
Security Groups	SG.AUTH.INGRESS SG.REVOKE.INGRESS SG.AUTH.EGRESS SG.REVOKE.EGRESS SG.CREATE SG.DELETE SG.ASSIGN SG.REMOVE
Host	HOST.RECONNECT
Maintenance	MAINT.CANCEL MAINT.CANCEL.PS MAINT.PREPARE MAINT.PREPARE.PS
VPN	VPN.REMOTE.ACCESS.CREATE VPN.REMOTE.ACCESS.DESTROY VPN.USER.ADD VPN.USER.REMOVE VPN.S2S.VPN.GATEWAY.CREATE VPN.S2S.VPN.GATEWAY.DELETE VPN.S2S.CUSTOMER.GATEWAY.CREATE VPN.S2S.CUSTOMER.GATEWAY.DELETE VPN.S2S.CUSTOMER.GATEWAY.UPDATE VPN.S2S.CONNECTION.CREATE VPN.S2S.CONNECTION.DELETE VPN.S2S.CONNECTION.RESET
Network	NETWORK.RESTART
Custom certificates	UPLOAD.CUSTOM.CERTIFICATE
OneToOnenat	STATICNAT.ENABLE STATICNAT.DISABLE ZONE.VLAN.ASSIGN ZONE.VLAN.RELEASE
Projects	PROJECT.CREATE PROJECT.UPDATE PROJECT.DELETE PROJECT.ACTIVATE PROJECT.SUSPEND PROJECT.ACCOUNT.ADD PROJECT.INVITATION.UPDATE PROJECT.INVITATION.REMOVE PROJECT.ACCOUNT.REMOVE
Network as a Service	NETWORK.ELEMENT.CONFIGURE
Physical Network Events	PHYSICAL.NETWORK.CREATE PHYSICAL.NETWORK.DELETE PHYSICAL.NETWORK.UPDATE
Physical Network Service Provider Events	SERVICE.PROVIDER.CREATE SERVICE.PROVIDER.DELETE SERVICE.PROVIDER.UPDATE
Physical Network Traffic Type Events	TRAFFIC.TYPE.CREATE TRAFFIC.TYPE.DELETE TRAFFIC.TYPE.UPDATE

Continued on next page

Table 1 – continued from previous page

Types	Events
External network device events	PHYSICAL.LOADBALANCER.ADD PHYSICAL.LOADBALANCER.DELETE PHYSICAL.LOADBALANCER.CONFIGURE
External switch management device events For example: Cisco Nexus 1000v Virtual Supervisor Module.	SWITCH.MGMT.ADD SWITCH.MGMT.DELETE SWITCH.MGMT.CONFIGURE SWITCH.MGMT.ENABLE SWITCH.MGMT.DISABLE PHYSICAL.FIREWALL.ADD PHYSICAL.FIREWALL.DELETE PHYSICAL.FIREWALL.CONFIGURE
VPC	VPC.CREATE VPC.UPDATE VPC.DELETE VPC.RESTART
Network ACL	NETWORK.ACL.CREATE NETWORK.ACL.DELETE NETWORK.ACL.REPLACE NETWORK.ACL.ITEM.CREATE NETWORK.ACL.ITEM.UPDATE NETWORK.ACL.ITEM.DELETE
VPC offerings	VPC.OFFERING.CREATE VPC.OFFERING.UPDATE VPC.OFFERING.DELETE
Private gateway	PRIVATE.GATEWAY.CREATE PRIVATE.GATEWAY.DELETE
Static routes	STATIC.ROUTE.CREATE STATIC.ROUTE.DELETE
Tag-related events	CREATE_TAGS DELETE_TAGS
Meta data-related events	CREATE_RESOURCE_DETAILS DELETE_RESOURCE_DETAILS
VM snapshot events	VMSNAPSHOT.CREATE VMSNAPSHOT.DELETE VMSNAPSHOT.REVERTTO
External network device events	PHYSICAL.NVPCONTROLLER.ADD PHYSICAL.NVPCONTROLLER.DELETE PHYSICAL.NVPCONTROLLER.CONFIGURE

Continued on next page

Table 1 – continued from previous page

Types	Events
AutoScale	COUNTER.CREATE COUNTER.DELETE CONDITION.CREATE CONDITION.DELETE AUTOSCALEPOLICY.CREATE AUTOSCALEPOLICY.UPDATE AUTOSCALEPOLICY.DELETE AUTOSCALEVMPROFILE.CREATE AUTOSCALEVMPROFILE.DELETE AUTOSCALEVMPROFILE.UPDATE AUTOSCALEVMGROUP.CREATE AUTOSCALEVMGROUP.DELETE AUTOSCALEVMGROUP.UPDATE AUTOSCALEVMGROUP.ENABLE AUTOSCALEVMGROUP.DISABLE PHYSICAL.DHCP.ADD PHYSICAL.DHCP.DELETE PHYSICAL.PXE.ADD PHYSICAL.PXE.DELETE AG.CREATE AG.DELETE AG.ASSIGN AG.REMOVE VM.AG.UPDATE INTERNALLBVM.START INTERNALLBVM.STOP HOST.RESERVATION.RELEASE
Dedicated guest vlan range	GUESTVLANRANGE.DEDICATE GUESTVLANRANGE.RELEASE PORTABLE.IP.RANGE.CREATE PORTABLE.IP.RANGE.DELETE PORTABLE.IP.TRANSFER
Dedicated Resources	DEDICATE.RESOURCE DEDICATE.RESOURCE.RELEASE VM.RESERVATION.CLEANUP UCS.ASSOCIATEPROFILE UCS.DISASSOCIATEPROFILE

6.2.3 Time Zones

The following time zone identifiers are accepted by PRODUCT. There are several places that have a time zone as a required or optional parameter. These include scheduling recurring snapshots, creating a user, and specifying the usage time zone in the Configuration table.

Etc/GMT+12	Etc/GMT+11	Pacific/Samoa
Pacific/Honolulu	US/Alaska	America/Los_Angeles
Mexico/BajaNorte	US/Arizona	US/Mountain
America/Chihuahua	America/Chicago	America/Costa_Rica
America/Mexico_City	Canada/Saskatchewan	America/Bogota
America/New_York	America/Caracas	America/Asuncion
America/Cuiaba	America/Halifax	America/La_Paz
America/Santiago	America/St_Johns	America/Araguaina
America/Argentina/Buenos_Aires	America/Cayenne	America/Godthab
America/Montevideo	Etc/GMT+2	Atlantic/Azores
Atlantic/Cape_Verde	Africa/Casablanca	Etc/UTC
Atlantic/Reykjavik	Europe/London	CET
Europe/Bucharest	Africa/Johannesburg	Asia/Beirut
Africa/Cairo	Asia/Jerusalem	Europe/Minsk
Europe/Moscow	Africa/Nairobi	Asia/Karachi
Asia/Kolkata	Asia/Bangkok	Asia/Shanghai
Asia/Kuala_Lumpur	Australia/Perth	Asia/Taipei
Asia/Tokyo	Asia/Seoul	Australia/Adelaide
Australia/Darwin	Australia/Brisbane	Australia/Canberra
Pacific/Guam	Pacific/Auckland	

6.3 Plugins

6.3.1 Storage Plugins

This section gives an outline of how to implement a plugin to integrate a third-party storage provider. For details and an example, you will need to read the code.

Note: Example code is available at: *plugins/storage/volume/sample*

Third party storage providers can integrate with CloudStack to provide either primary storage or secondary storage. For example, CloudStack provides plugins for Amazon Simple Storage Service (S3) or OpenStack Object Storage (Swift). The S3 plugin can be used for any object storage that supports the Amazon S3 interface.

Additional third party object storages that do not support the S3 interface can be integrated with CloudStack by writing plugin software that uses the object storage plugin framework. Several new interfaces are available so that storage providers can develop vendor-specific plugins based on well-defined contracts that can be seamlessly managed by CloudStack.

Artifacts such as templates, ISOs and snapshots are kept in storage which CloudStack refers to as secondary storage. To improve scalability and performance, as when a number of hosts access secondary storage concurrently, object storage can be used for secondary storage. Object storage can also provide built-in high availability capability. When using object storage, access to secondary storage data can be made available across multiple zones in a region. This is a huge benefit, as it is no longer necessary to copy templates, snapshots etc. across zones as would be needed in an environment using only zone-based NFS storage.

The user enables a storage plugin through the UI. A new dialog box choice is offered to select the storage provider. Depending on which provider is selected, additional input fields may appear so that the user can provide the additional details required by that provider, such as a user name and password for a third-party storage account.

Overview of How to Write a Storage Plugin

To add a third-party storage option to CloudStack, follow these general steps (explained in more detail later in this section):

1. Implement the following interfaces in Java:
   - DataStoreDriver
   - DataStoreLifecycle
   - DataStoreProvider
   - VMSnapshotStrategy (if you want to customize the VM snapshot functionality)
2. Hardcode your plugin's required additional input fields into the code for the Add Secondary Storage or Add Primary Storage dialog box.
3. Place your .jar file in *plugins/storage/volume/* or *plugins/storage/image/*.
4. Edit */client/tomcatconf/componentContext.xml.in*.
5. Edit *client/pom.xml*.

Implementing DataStoreDriver

DataStoreDriver contains the code that CloudStack will use to provision the object store, when needed.

You must implement the following methods:

- getTO()
- getStoreTO()
- createAsync()
- deleteAsync()

The following methods are optional:

- resize()
- canCopy() is optional. If you set it to true, then you must implement copyAsync().

Implementing DataStoreLifecycle

DataStoreLifecycle contains the code to manage the storage operations for ongoing use of the storage. Several operations are needed, like create, maintenance mode, delete, etc.

You must implement the following methods:

- initialize()
- maintain()
- cancelMaintain()
- deleteDataStore()
- Implement one of the attach*() methods depending on what scope you want the storage to have: attachHost(), attachCluster(), or attachZone().

Implementing DataStoreProvider

DataStoreProvider contains the main code of the data store.

You must implement the following methods:

- getDatastoreLifeCycle()
- getDataStoreDriver()
- getTypes(). Returns one or more types of storage for which this data store provider can be used. For secondary object storage, return IMAGE, and for a Secondary Staging Store, return ImageCache.
- configure(). First initialize the lifecycle implementation and the driver implementation, then call registerDriver() to register the new object store provider instance with CloudStack.
- getName(). Returns the unique name of your provider; for example, this can be used to get the name to display in the UI.

The following methods are optional:

- getHostListener() is optional; it's for monitoring the status of the host.

Implementing VMSnapshotStrategy

VMSnapshotStrategy has the following methods:

- takeVMSnapshot()
- deleteVMSnapshot()
- revertVMSnapshot()
- canHandle(). For a given VM snapshot, tells whether this implementation of VMSnapshotStrategy can handle it.

Place the .jar File in the Right Directory

For a secondary storage plugin, place your .jar file here:

```
plugins/storage/image/
```

For a primary storage plugin, place your .jar file here:

```
plugins/storage/volume/
```

Edit Configuration Files

First, edit the following file tell CloudStack to include your .jar file. Add a line to this file to tell the CloudStack Management Server that it now has a dependency on your code:

```
client/pom.xml
```

Place some facts about your code in the following file so CloudStack can run it:

```
/client/tomcatconf/componentContext.xml.in
```

In the section "Deployment configurations of various adapters," add this:

```
<bean>id="some unique ID" class="package name of your implementation of␣
↪DataStoreProvider"</bean>
```

In the section "Storage Providers," add this:

```
<property name="providers">
   <ref local="same ID from the bean tag's id attribute">
</property>
```

Minimum Required Interfaces

The classes, interfaces, and methods used by CloudStack from the Amazon Web Services (AWS) Java SDK are listed in this section. An object storage that supports the S3 interface is minimally required to support the below in order to be compatible with CloudStack.

Interface AmazonS3

http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/s3/AmazonS3.html

Modifier and Type	Method and Description
Bucket	createBucket(String bucketName) Creates a new Amazon S3 bucket with the specified name in the default (US) region, Region.US_Standard.
void	deleteObject(String bucketName, String key) Deletes the specified object in the specified bucket.
ObjectMetadata	getObject(GetObjectRequest getObjectRequest, File destinationFile) Gets the object metadata for the object stored in Amazon S3 under the specified bucket and key, and saves the object contents to the specified file.
S3Object	getObject(String bucketName, String key) Gets the object stored in Amazon S3 under the specified bucket and key.
URL	generatePresignedUrl(String bucketName, String key, Date expiration, HttpMethod method) Returns a pre-signed URL for accessing an Amazon S3 resource.
void	deleteBucket(String bucketName) Deletes the specified bucket.
List<Bucket>	listBuckets() Returns a list of all Amazon S3 buckets that the authenticated sender of the request owns.
ObjectListing	listObjects(String bucketName, String prefix) Returns a list of summary information about the objects in the specified bucket.
PutObjectResult	putObject(PutObjectRequest putObjectRequest) Uploads a new object to the specified Amazon S3 bucket.
PutObjectResult	putObject(String bucketName, String key, File file) Uploads the specified file to Amazon S3 under the specified bucket and key name.
PutObjectResult	putObject(String bucketName, String key, InputStream input, ObjectMetadata metadata) Uploads the specified input stream and object metadata to Amazon S3 under the specified bucket and key name.
void	setEndpoint(String endpoint) Overrides the default endpoint for this client.
void	setObjectAcl(String bucketName, String key, CannedAccessControlList acl) Sets the CannedAccessControlList for the specified object in Amazon S3 using one of the pre-configured CannedAccessControlLists.

Class TransferManager

http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/s3/transfer/TransferManager.html

Modifier and Type	Method and Description
Upload	upload(PutObjectRequest putObjectRequest) Schedules a new transfer to upload data to Amazon S3.

Class PutObjectRequest

http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/s3/model/PutObjectRequest.html

Modifier and Type	Method and Description
Upload	upload(PutObjectRequest putObjectRequest) Schedules a new transfer to upload data to Amazon S3.

6.3.2 Third Party UI Plugins

Using the new third-party plugin framework, you can write and install extensions to CloudStack. The installed and enabled plugins will appear in the UI alongside the other features. The code for the plugin is simply placed in a special directory within CloudStack's installed code at any time after CloudStack installation. The new plugin appears only when it is enabled by the cloud administrator.

The left navigation bar of the CloudStack UI has a new Plugins button to help you work with UI plugins.

How to Write a Plugin: Overview

The basic procedure for writing a plugin is:

1. Write the code and create the other files needed. You will need the plugin code itself (in Javascript), a thumbnail image, the plugin listing, and a CSS file.

   All UI plugins have the following set of files:

```
+-- cloudstack/
  +-- ui/
    +-- plugins/
      +-- csMyFirstPlugin/
        +-- config.js            --> Plugin metadata (title, author, vendor URL,␣
↪etc.)
        +-- icon.png             --> Icon, shown on side nav bar and plugin␣
↪listing
                                     (should be square, and ~50x50px)
        +-- csMyFirstPlugin.css  --> CSS file, loaded automatically when plugin␣
↪loads
        +-- csMyFirstPlugin.js   --> Main JS file, containing plugin code
```

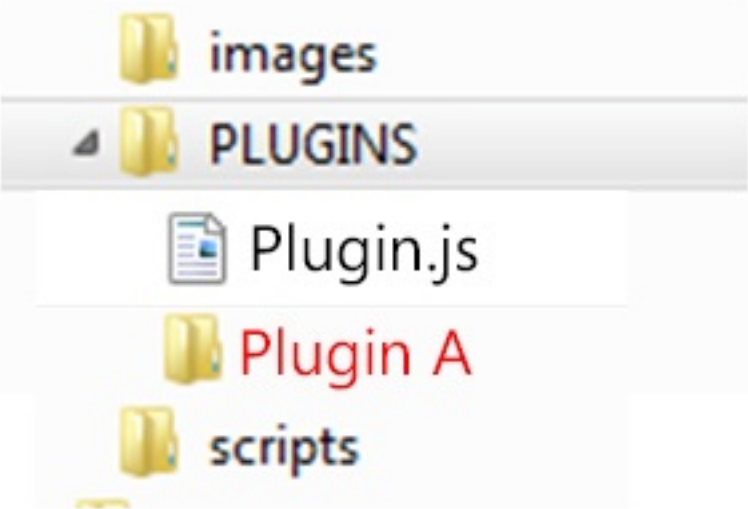
   The same files must also be present at */tomcat/webapps/client/plugins*.

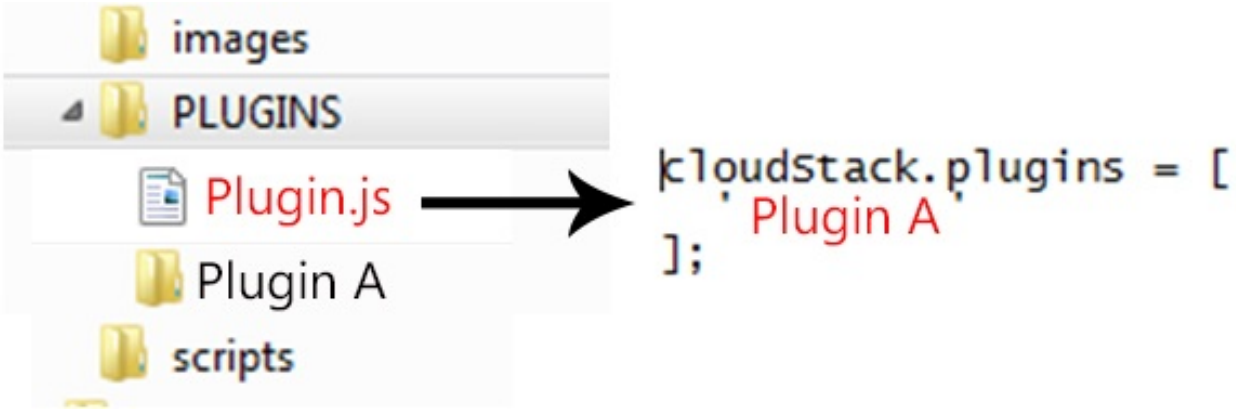
2. The CloudStack administrator adds the folder containing your plugin code under the CloudStack PLUGINS folder.
3. The administrator also adds the name of your plugin to the plugin.js file in the PLUGINS folder.
4. The next time the user refreshes the UI in the browser, your plugin will appear in the left navigation bar.

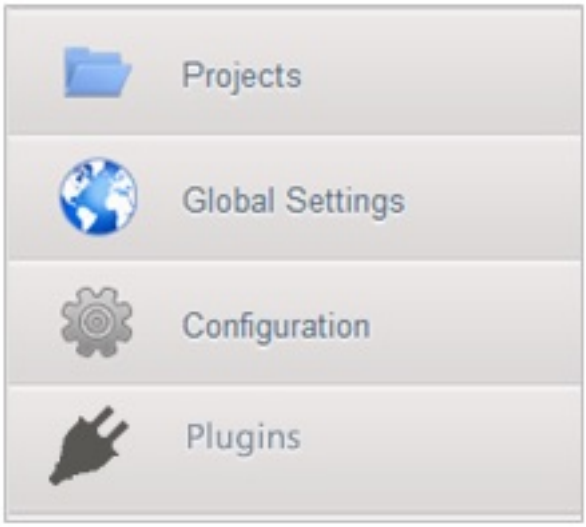
How to Write a Plugin: Implementation Details

This section requires an understanding of JavaScript and the CloudStack API. You don't need knowledge of specific frameworks for this tutorial (jQuery, etc.), since the CloudStack UI handles the front-end rendering for you.

Plugin A
plugin.js
File Hierarchy
-plugin
-thumbnail
-description
-etc...


images
PLUGINS
Plugin.js
Plugin A
scripts


images
PLUGINS
Plugin.js
Plugin A
scripts
cloudStack.plugins = [
'Plugin A'
];


Projects
Global Settings
Configuration
Plugins

There is much more to the CloudStack UI framework than can be described here. The UI is very flexible to handle many use cases, so there are countless options and variations. The best reference right now is to read the existing code for the main UI, which is in the /ui folder. Plugins are written in a very similar way to the main UI.

1. **Create the directory to hold your plugin.**

   All plugins are composed of set of required files in the directory /ui/plugins/pluginID, where pluginID is a short name for your plugin. It's recommended that you prefix your folder name (for example, bfMyPlugin) to avoid naming conflicts with other people's plugins.

   In this example, the plugin is named csMyFirstPlugin.

```
$ cd cloudstack/ui/plugins
$ mkdir csMyFirstPlugin
$ ls -l

total 8
drwxr-xr-x  2 bgregory  staff   68 Feb 11 14:44 csMyFirstPlugin
-rw-r--r--  1 bgregory  staff  101 Feb 11 14:26 plugins.js
```

2. **Change to your new plugin directory.**

```
$ cd csMyFirstPlugin
```

3. **Set up the listing.**

   Add the file *config.js*, using your favorite editor.

```
$ vi config.js
```

   Add the following content to config.js. This information will be displayed on the plugin listing page in the UI:

```
(function (cloudStack) {
  cloudStack.plugins.csMyFirstPlugin.config = {
    title: 'My first plugin',
    desc: 'Tutorial plugin',
    externalLink: 'http://www.cloudstack.org/',
    authorName: 'Test Plugin Developer',
    authorEmail: 'plugin.developer@example.com'
  };
}(cloudStack));
```

4. **Add a new main section.**

   Add the file csMyFirstPlugin.js, using your favorite editor.

```
$ vi csMyFirstPlugin.js
```

   Add the following content to csMyFirstPlugin.js:

```
(function (cloudStack) {
  cloudStack.plugins.csMyFirstPlugin = function(plugin) {
    plugin.ui.addSection({
      id: 'csMyFirstPlugin',
      title: 'My Plugin',
      preFilter: function(args) {
        return isAdmin();
      },
      show: function() {
```

(continues on next page)

(continued from previous page)

```
        return $('<div>').html('Content will go here');
      }
    });
  };
}(cloudStack));
```

5. **Register the plugin.**

   You now have the minimal content needed to run the plugin, so you can activate the plugin in the UI by adding it to plugins.js. First, edit the file:

```
$ cd cloudstack/ui/plugins
$ vi plugins.js
```

   Now add the following to plugins.js:

```
(function($, cloudStack) {
  cloudStack.plugins = [
    'csMyFirstPlugin'
  ];
}(jQuery, cloudStack));
```

6. **Check the plugin in the UI.**

   First, copy all the plugin code that you have created so far to */tomcat/webapps/client/plugins*. Then refresh the browser and click Plugins in the side navigation bar. You should see your new plugin.

7. **Make the plugin do something.**

   Right now, you just have placeholder content in the new plugin. It's time to add real code. In this example, you will write a basic list view, which renders data from an API call. You will list all virtual machines owned by the logged-in user. To do this, replace the 'show' function in the plugin code with a 'listView' block, containing the required syntax for a list view. To get the data, use the listVirtualMachines API call. Without any parameters, it will return VMs only for your active user. Use the provided 'apiCall' helper method to handle the server call. Of course, you are free to use any other method for making the AJAX call (for example, jQuery's $.ajax method).

   First, open your plugin's JavaScript source file in your favorite editor:

```
$ cd csMyFirstPlugin
$ vi csMyFirstPlugin.js
```

   Add the following code in csMyFirstPlugin.js:

```
(function (cloudStack) {
  cloudStack.plugins.csMyFirstPlugin = function(plugin) {
    plugin.ui.addSection({
      id: 'csMyFirstPlugin',
      title: 'My Plugin',
      preFilter: function(args) {
        return isAdmin();
      },

      // Render page as a list view
      listView: {
        id: 'testPluginInstances',
        fields: {
          name: { label: 'label.name' },
          instancename: { label: 'label.internal.name' },
```

(continues on next page)

(continued from previous page)

```
          displayname: { label: 'label.display.name' },
          zonename: { label: 'label.zone.name' }
        },
        dataProvider: function(args) {
          // API calls go here, to retrive the data asynchronously
          //
          // On successful retrieval, call
          // args.response.success({ data: [data array] });
          plugin.ui.apiCall('listVirtualMachines', {
            success: function(json) {
              var vms = json.listvirtualmachinesresponse.virtualmachine;

              args.response.success({ data: vms });
            },
            error: function(errorMessage) {
              args.response.error(errorMessage)
            }
          });
        }
      }
    });
  };
}(cloudStack));
```

8. **Test the plugin.**

   First, copy all the plugin code that you have created so far to */tomcat/webapps/client/plugins*. Then refresh the browser. You can see that your placeholder content was replaced with a list table, containing 4 columns of virtual machine data.

9. **Add an action button.**

   Let's add an action button to the list view, which will reboot the VM. To do this, add an actions block under listView. After specifying the correct format, the actions will appear automatically to the right of each row of data.

```
$ vi csMyFirstPlugin.js
```

   Now add the following new code in csMyFirstPlugin.js. (The dots ... show where we have omitted some existing code for the sake of space. Don't actually cut and paste that part):

```
...
   listView: {
     id: 'testPluginInstances',
     ...

     actions: {
       // The key/ID you specify here will determine what icon is
       // shown in the UI for this action,
       // and will be added as a CSS class to the action's element
       // (i.e., '.action.restart')
       //
       // -- here, 'restart' is a predefined name in CloudStack that will
       // automatically show a 'reboot' arrow as an icon;
       // this can be changed in csMyFirstPlugin.css
       restart: {
         label: 'Restart VM',
```

(continues on next page)

(continued from previous page)

```
        messages: {
          confirm: function() { return 'Are you sure you want to restart this VM?
↪' },
          notification: function() { return 'Rebooted VM' }
        },
        action: function(args) {
          // Get the instance object of the selected row from context
          //
          // -- all currently loaded state is stored in 'context' as objects,
          //    such as the selected list view row,
          //    the selected section, and active user
          //
          // -- for list view actions, the object's key will be the same as
          //    listView.id, specified above;
          //    always make sure you specify an 'id' for the listView,
          //     or else it will be 'undefined!'
          var instance = args.context.testPluginInstances[0];

          plugin.ui.apiCall('rebootVirtualMachine', {
            // These will be appended to the API request
            //
            // i.e., rebootVirtualMachine&id=...
            data: {
              id: instance.id
            },
            success: function(json) {
              args.response.success({
                // This is an async job, so success here only indicates
                // that the job was initiated.
                //
                // To pass the job ID to the notification UI
                // (for checking to see when action is completed),
                // '_custom: { jobID: ... }' needs to always be passed on␣
↪success,
                // in the same format as below
                _custom: { jobId: json.rebootvirtualmachineresponse.jobid }
              });
            },


            error: function(errorMessage) {
              args.response.error(errorMessage); // Cancel action, show error␣
↪message returned
            }
          });
        },

        // Because rebootVirtualMachine is an async job, we need to add
        // a poll function, which will perodically check
        // the management server to see if the job is ready
        // (via pollAsyncJobResult API call)
        //
        // The plugin API provides a helper function, 'plugin.ui.pollAsyncJob',
        /  which will work for most jobs
        // in CloudStack
        notification: {
          poll: plugin.ui.pollAsyncJob
```

(continues on next page)

(continued from previous page)

```
          }
        }
      },

      dataProvider: function(args) {
        ...
...
```

10. **Add the thumbnail icon.**

    Create an icon file; it should be square, about 50x50 pixels, and named *icon.png*. Copy it into the same directory with your plugin code: *cloudstack/ui/plugins/csMyFirstPlugin/icon.png*.

11. **Add the stylesheet.**

    Create a CSS file, with the same name as your *.js* file. Copy it into the same directory with your plugin code: *cloudstack/ui/plugins/csMyFirstPlugin/csMyFirstPlugin.css*.

6.4 Allocators

CloudStack enables administrators to write custom allocators that will choose the Host to place a new guest and the storage host from which to allocate guest virtual disk images.

These are following categories of allocators currently supported:

- HostAllocators - Allows you to create custom rules to determine which physical host to allocate the guest virtual machines on.
- StoragePoolAllocators - Allows you to create custom rules to determine which storage pool to allocate the guest virtual machines on.

6.4.1 Implementing a custom HostAllocator

HostAllocators are written by extending com.cloud.agent.manager.allocator.HostAllocator interface.

HostAllocator Interface

The interface defines the following two methods.

```
/**
  * Checks if the VM can be upgraded to the specified ServiceOffering
  * @param UserVm vm
  * @param ServiceOffering offering
  * @return boolean true if the VM can be upgraded
**/

publicboolean isVirtualMachineUpgradable(final UserVm vm, final ServiceOffering␣
↪offering);

/**
  * Determines which physical hosts are suitable to allocate the guest virtual␣
↪machines on
  *
  * @paramVirtualMachineProfile vmProfile
```

(continues on next page)

(continued from previous page)

```
  * @paramDeploymentPlan plan
  * @paramType type
  * @paramExcludeList avoid
  * @paramint returnUpTo
  * @returnList<Host>List of hosts that are suitable for VM allocation
**/

publicList<Host> allocateTo( VirtualMachineProfile<?extendsVirtualMachine> vmProfile,␣
↪ DeploymentPlan plan, Type type, ExcludeList avoid, intreturnUpTo);
```

A custom HostAllocator can be written by implementing the 'allocateTo' method

Input Parameters for the method 'HostAllocator :: allocateTo'

com.cloud.vm.VirtualMachineProfile vmProfile

VirtualMachineProfile describes one virtual machine. This allows the adapters like Allocators to process the information in the virtual machine and make determinations on what the virtual machine profile should look like before it is actually started on the hypervisor.

HostAllocators can make use of the following information present in the VirtualMachineProfile:

- The ServiceOffering that specifies configuration like requested CPU speed, RAM etc necessary for the guest VM.
- The VirtualMachineTemplate, the template to be used to start the VM.

com.cloud.deploy.DeploymentPlan plan

DeploymentPlan should specify:

- dataCenterId: The data center the VM should deploy in
- podId: The pod the Vm should deploy in; null if no preference
- clusterId: The cluster the VM should deploy in; null if no preference
- poolId: The storage pool the VM should be created in; null if no preference

com.cloud.host.Host.Type type

Type of the Host needed for this guest VM. Currently com.cloud.host.Host.Type interface defines the following Host types:

- Storage
- Routing
- SecondaryStorage
- ConsoleProxy
- ExternalFirewall
- ExternalLoadBalancer

com.cloud.deploy.DeploymentPlanner.ExcludeList avoid

The ExcludeList specifies what datacenters, pods, clusters, hosts, storagePools should not be considered for allocating this guest VM. HostAllocators should avoid the hosts that are mentioned in ExcludeList.hostIds.

- Set Long dcIds;

- Set Long podIds;
- Set Long clusterIds;
- Set Long hostIds;
- Set Long poolIds;

int returnUpTo

This specifies return up to that many available hosts for this guest VM.

To get all possible hosts, set this value to -1.

Reference HostAllocator implementation

Refer com.cloud.agent.manager.allocator.impl.FirstFitAllocator that implements the HostAllocator interface. This allocator checks available hosts in the specified datacenter, Pod, Cluster and considering the given ServiceOffering requirements.

If returnUpTo = 1, this allocator would return the first Host that fits the requirements of the guest VM.

Loading a custom HostAllocator

1. Write a custom HostAllocator class, implementing the interface described above.
2. Package the code into a JAR file and make the JAR available in the classpath of the Management Server/tomcat.
3. Modify the components.xml and components-premium.xml files found in /client/ tomcatconf as follows.
4. Search for 'HostAllocator' in these files.

```
<adapters key="com.cloud.agent.manager.allocator.HostAllocator">
  <adapter name="FirstFit" class="com.cloud.agent.manager.allocator.impl.
↪FirstFitAllocator"/>
</adapters>
```

5. Replace the FirstFitAllocator with your class name. Optionally, you can change the name of the adapter as well.
6. Restart the Management Server.

6.4.2 Implementing a custom StoragePoolAllocator

StoragePoolAllocators are written by extending com.cloud.storage.allocator. StoragePoolAllocator interface.

StoragePoolAllocator Interface

A custom StoragePoolAllocator can be written by implementing the 'allocateTo' method.

```
/**
  * Determines which storage pools are suitable for the guest virtual machine
  * @param DiskProfile dskCh
  * @param VirtualMachineProfile vmProfile
  * @param DeploymentPlan plan
  * @param ExcludeList avoid
  * @param int returnUpTo
```

(continues on next page)

(continued from previous page)

```
  * @return List<StoragePool> List of storage pools that are suitable for the VM
**/

public List<StoragePool> allocateToPool(DiskProfile dskCh, VirtualMachineProfile<?
↪extends VirtualMachine> vm, DeploymentPlan plan, ExcludeList avoid, int returnUpTo);
```

This interface also contains some other methods to support some legacy code. However your custom allocator can extend the existing com.cloud.storage.allocator. AbstractStoragePoolAllocator. This class provides default implementation for all the other interface methods.

Input Parameters for the method 'StoragePoolAllocator :: allocateTo'

com.cloud.vm.DiskProfile dskCh

DiskCharacteristics describes a disk and what functionality is required from it. It specifies the storage pool tags if any to be used while searching for a storage pool.

com.cloud.vm.VirtualMachineProfile vmProfile

VirtualMachineProfile describes one virtual machine. This allows the adapters like Allocators to process the information in the virtual machine and make determinations on what the virtual machine profile should look like before it is actually started on the hypervisor.

StoragePoolAllocators can make use of the following information present in the VirtualMachineProfile:

- The VirtualMachine instance that specifies properties of the guest VM.
- The VirtualMachineTemplate, the template to be used to start the VM.

com.cloud.deploy.DeploymentPlan plan

DeploymentPlan should specify:

- dataCenterId: The data center the VM should deploy in
- podId: The pod the VM should deploy in; null if no preference
- clusterId: The cluster the VM should deploy in; null if no preference
- poolId: The storage pool the VM should be created in; null if no preference

com.cloud.deploy.DeploymentPlanner.ExcludeList avoid

The ExcludeList specifies what datacenters, pods, clusters, hosts, storagePools should not be considered for allocating this guest VM. StoragePoolAllocators should avoid the pools that are mentioned in ExcludeList.poolIds

- Set Long dcIds;
- Set Long podIds;
- Set Long clusterIds;
- Set Long hostIds;
- Set Long poolIds;

int returnUpTo

This specifies return up to that many available pools for this guest VM

To get all possible pools, set this value to -1

Reference StoragePoolAllocator implementation

Refer com.cloud.storage.allocator.FirstFitStoragePoolAllocator that implements the StoragePoolAllocator interface. This allocator checks available pools in the specified datacenter, Pod, Cluster and considering the given DiskProfile characteristics.

If returnUpTo = 1, this allocator would return the first Storage Pool that fits the requirements of the guest VM.

Loading a custom StoragePoolAllocator

1. Write a custom StoragePoolAllocator class, implementing the interface described above.
2. Package the code into a JAR file and make the JAR available in the classpath of the Management Server/tomcat.
3. Modify the components.xml and components-premium.xml files found in /client/ tomcatconf as follows.
4. Search for 'StoragePoolAllocator' in these files.

```
<adapters key="com.cloud.storage.allocator.StoragePoolAllocator">
   <adapter name="Storage" class="com.cloud.storage.allocator.
↪FirstFitStoragePoolAllocator"/>
</adapters>
```

5. Replace the FirstFitStoragePoolAllocator with your class name. Optionally, you can change the name of the adapter as well.
6. Restart the Management Server.

6.5 Deploying CloudStack with Ansible

In this article, Paul Angus Cloud Architect at ShapeBlue takes a look at using Ansible to Deploy an Apache CloudStack cloud.

6.5.1 What is Ansible

Ansible is a deployment and configuration management tool similar in intent to Chef and Puppet. It allows (usually) DevOps teams to orchestrate the deployment and configuration of their environments without having to re-write custom scripts to make changes.

Like Chef and Puppet, Ansible is designed to be idempotent. This means that you determine the state you want a host to be in and Ansible will decide if it needs to act in order to achieve that state.

6.5.2 There's already Chef and Puppet, so what's the fuss about Ansible?

Let's take it as a given that configuration management makes life much easier (and is quite cool), Ansible only needs an SSH connection to the hosts that you're going to manage to get started. While Ansible requires Python 2.4 or greater on the host you're going to manage in order to leverage the vast majority of its functionality, it is able to connect to hosts which don't have Python installed in order to then install Python, so it's not really a problem. This greatly simplifies the deployment procedure for hosts, avoiding the need to pre-install agents onto the clients before the configuration management can take over.

Ansible will allow you to connect as any user to a managed host (with that user's privileges) or by using public/private keys – allowing fully automated management.

There also doesn't need to be a central server to run everything, as long as your playbooks and inventories are in-sync you can create as many Ansible servers as you need (generally a bit of Git pushing and pulling will do the trick).

Finally – its structure and language is pretty simple and clean. I've found it a bit tricky to get the syntax correct for variables in some circumstances, but otherwise I've found it one of the easier tools to get my head around.

6.5.3 So let's see something

For this example we're going to create an Ansible server which will then deploy a CloudStack server. Both of these servers will be CentOS 6.4 virtual machines.

6.5.4 Installing Ansible

Installing Ansible is blessedly easy. We generally prefer to use CentOS so to install Ansible you run the following commands on the Ansible server.

```
# rpm -ivh http://www.mirrorservice.org/sites/dl.fedoraproject.org/pub/epel/6/i386/
↪epel-release-6-8.noarch.rpm
# yum install -y ansible
```

And that's it.

(There is a commercial version which has more features such as callback to request configurations and a RESTful API and also support. The installation of this is different)

By default Ansible uses /etc/ansible to store your playbooks, I tend to move it, but there's no real problem with using the default location. Create yourself a little directory structure to get started with. The documentation recommends something like this:

6.5.5 Playbooks

Ansible uses playbooks to specify the state in which you wish the target host to be in to be able to accomplish its role. Ansible playbooks are written in YAML format.

6.5.6 Modules

To get Ansible to do things you specify the hosts a playbook will act upon and then call modules and supply arguments which determine what Ansible will do to those hosts.

To keep things simple, this example is a cut-down version of a full deployment. This example creates a single management server with a local MySQL server and assumes you have your secondary storage already provisioned somewhere. For this example I'm also not going to include securing the MySQL server, configuring NTP or using Ansible to configure the networking on the hosts either. Although normally we'd use Ansible to do exactly that.

The pre-requisites to this CloudStack build are:

- A CentOS 6.4 host to install CloudStack on
- An IP address already assigned on the ACS management host
- The ACS management host should have a resolvable FQDN (either through DNS or the host file on the ACS management host)
- Internet connectivity on the ACS management host

6.5.7 Planning

The first step I use is to list all of the tasks I think I'll need and group them or split them into logical blocks. So for this deployment of CloudStack I'd start with:

- Configure selinux
- (libselinux-python required for Ansible to work with selinux enabled hosts)
- Install and configure MySQL
- (Python MySQL-DB required for Ansible MySQL module)
- Install cloud-client
- Seed secondary storage

Ansible is built around the idea of hosts having roles, so generally you would group or manage your hosts by their roles. So now to create some roles for these tasks

I've created:

- cloudstack-manager
- mysql

First up we need to tell Ansible where to find our CloudStack management host. In the root Ansible directory there is a file called 'hosts' (/etc/Ansible/hosts) add a section like this:

```
[acs-manager]
xxx.xxx.xxx.xxx
```

where xxx.xxx.xxx.xxx is the ip address of your ACS management host.

6.5.8 MySQL

So let's start with the MySQL server. We'll need to create a task within the mysql role directory called main.yml. The 'task' in this case to have MySQL running and configured on the target host. The contents of the file will look like this:

```
-name: Ensure mysql server is installed

yum: name=mysql-server state=present

- name: Ensure mysql python is installed

yum: name=MySQL-python state=present


- name: Ensure selinux python bindings are installed

yum: name=libselinux-python state=present

- name: Ensure cloudstack specfic my.cnf lines are present

lineinfile: dest=/etc/my.cnf regexp='$item' insertafter="symbolic-links=0 line='$item'

with\_items:

- skip-name-resolve
```

(continues on next page)

(continued from previous page)

```
- default-time-zone='+00:00

- innodb\_rollback\_on\_timeout=1

- innodb\_lock\_wait\_timeout=600

- max\_connections=350

- log-bin=mysql-bin

 - binlog-format = 'ROW'


- name: Ensure MySQL service is started

service: name=mysqld state=started

- name: Ensure MySQL service is enabled at boot

service: name=mysqld enabled=yes



- name: Ensure root password is set

mysql\_user: user=root password=$mysql\_root\_password host=localhost

ignore\_errors: true

- name: Ensure root has sufficient privileges

mysql\_user: login\_user=root login\_password=$mysql\_root\_password user=root host=%␣
↪password=$mysql\_root\_password priv=\*.\*:GRANT,ALL state=present
```

This needs to be saved as */etc/ansible/roles/mysql/tasks/main.yml*

As explained earlier, this playbook in fact describes the state of the host rather than setting out commands to be run. For instance, we specify certain lines which must be in the my.cnf file and allow Ansible to decide whether or not it needs to add them.

Most of the modules are self-explanatory once you see them, but to run through them briefly;

The 'yum' module is used to specify which packages are required, the 'service' module controls the running of services, while the 'mysql_user' module controls mysql user configuration. The 'lineinfile' module controls the contents in a file.

We have a couple of variables which need declaring. You could do that within this playbook or its 'parent' playbook, or as a higher level variable. I'm going to declare them in a higher level playbook. More on this later.

That's enough to provision a MySQL server. Now for the management server.

6.5.9 CloudStack Management server service

For the management server role we create a main.yml task like this:

```
- name: Ensure selinux python bindings are installed

  yum: name=libselinux-python state=present


- name: Ensure the Apache Cloudstack Repo file exists as per template

  template: src=cloudstack.repo.j2 dest=/etc/yum.repos.d/cloudstack.repo


- name: Ensure selinux is in permissive mode

  command: setenforce permissive


- name: Ensure selinux is set permanently

  selinux: policy=targeted state=permissive


-name: Ensure CloudStack packages are installed

  yum: name=cloud-client state=present


- name: Ensure vhdutil is in correct location

  get\_url: url=http://download.cloudstack.org/tools/vhd-util dest=/usr/share/
↪cloudstack-common/scripts/vm/hypervisor/xenserver/vhd-util mode=0755
```

Save this as */etc/ansible/roles/cloudstack-management/tasks/main.yml*

Now we have some new elements to deal with. The Ansible template module uses Jinja2 based templating. As we're doing a simplified example here, the Jinja template for the cloudstack.repo won't have any variables in it, so it would simply look like this:

```
[cloudstack]
name=cloudstack
baseurl=http://download.cloudstack.org/rhel/4.2/
enabled=1
gpgcheck=0
```

This is saved in */etc/ansible/roles/cloudstack-manager/templates/cloudstack.repo.j2*

That gives us the packages installed, we need to set up the database. To do this I've created a separate task called setupdb.yml

```
- name: cloudstack-setup-databases
command: /usr/bin/cloudstack-setup-databases cloud:{{mysql\_cloud\_password }}
↪@localhost -deploy-as=root:{{mysql\_root\_password }}

- name: Setup CloudStack manager
command: /usr/bin/cloudstack-setup-management
```

Save this as: */etc/ansible/roles/cloudstack-management/tasks/setupdb.yml*

As there isn't (as yet) a CloudStack module, Ansible doesn't inherently know whether or not the databases have already been provisioned, therefore this step is not currently idempotent and will overwrite any previously provisioned databases.

There are some more variables here for us to declare later.

6.5.10 System VM Templates:

Finally we would want to seed the system VM templates into the secondary storage. The playbook for this would look as follows:

```
- name: Ensure secondary storage mount exists
  file: path={{ tmp\_nfs\_path }} state=directory


- name: Ensure  NFS storage is mounted
  mount: name={{ tmp\_nfs\_path }} src={{ sec\_nfs\_ip }}:{{sec\_nfs\_path }}␣
↪fstype=nfs state=mounted opts=nolock


- name: Seed secondary storage
  command:
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-tmplt -m {{␣
↪tmp\_nfs\_path }} -u http://download.cloud.com/templates/4.2/systemvmtemplate-2013-
↪06-12-master-kvm.qcow2.bz2 -h kvm -F

  command:
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-tmplt -m {{␣
↪tmp\_nfs\_path }} -u http://download.cloud.com/templates/4.2/systemvmtemplate-2013-
↪07-12-master-xen.vhd.bz2 -h xenserver -F

  command:
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-tmplt -m {{␣
↪tmp\_nfs\_path }} -u http://download.cloud.com/templates/4.2/systemvmtemplate-4.2-
↪vh7.ov -h vmware -F
```

Save this as */etc/ansible/roles/cloudstack-manager/tasks/seedstorage.yml*

Again, there isn't a CloudStack module so Ansible will always run this even if the secondary storage already has the templates in it.

6.5.11 Bringing it all together

Ansible can use playbooks which run other playbooks, this allows us to group these playbooks together and declare variables across all of the individual playbooks. So in the Ansible playbook directory create a file called deploy-cloudstack.yml, which would look like this:

```
-hosts: acs-manager

 vars:

    mysql\_root\_password: Cl0ud5tack
    mysql\_cloud\_password: Cl0ud5tack
    tmp\_nfs\_path: /mnt/secondary
    sec\_nfs\_ip: IP\_OF\_YOUR\_SECONDARY\_STORAGE
    sec\_nfs\_path: PATH\_TO\_YOUR\_SECONDARY\_STORAGE\_MOUNT


 roles:
```

(continues on next page)

(continued from previous page)

```
  - mysql
  - cloudstack-manager

tasks:

  - include: /etc/ansible/roles/cloudstack-manager/tasks/setupdb.yml
  - include: /etc/ansible/roles/cloudstack-manager/tasks/seedstorage.yml
```

Save this as */etc/ansible/deploy-cloudstack.yml* inserting the IP address and path for your secondary storage and changing the passwords if you wish to.

To run this go to the Ansible directory (cd /etc/ansible) and run:

```
# ansible-playbook deploy-cloudstack.yml -k
```

'-k' tells Ansible to ask you for the root password to connect to the remote host.

Now log in to the CloudStack UI on the new management server.

6.5.12 How is this example different from a production deployment?

In a production deployment, the Ansible playbooks would configure multiple management servers connected to master/slave replicating MySQL databases along with any other infrastructure components required and deploy and configure the hypervisor hosts. We would also have a dedicated file describing the hosts in the environment and a dedicated file containing variables which describe the environment.

The advantage of using a configuration management tool such as Ansible is that we can specify components like the MySQL database VIP once and use it multiple times when configuring the MySQL server itself and other components which need to use that information.

6.5.13 Acknowledgements

Thanks to Shanker Balan for introducing me to Ansible and a load of handy hints along the way.

6.6 Getting Help

Need some help getting started? Feel free to ask on the mailing list:

- users@: This list is for users of CloudStack to seek and provide support. This is a moderately high volume list.
- dev@: Where discussions about development and the project itself happen. This is a high volume list.

Or on one of the following IRC channels on irc.freenode.net:

- #cloudstack - General Apache CloudStack conversation and end user support
- #cloudstack-dev - Development discussions
- #cloudstack-meeting - Weekly and ad-hoc meeting room for the Apache CloudStack community

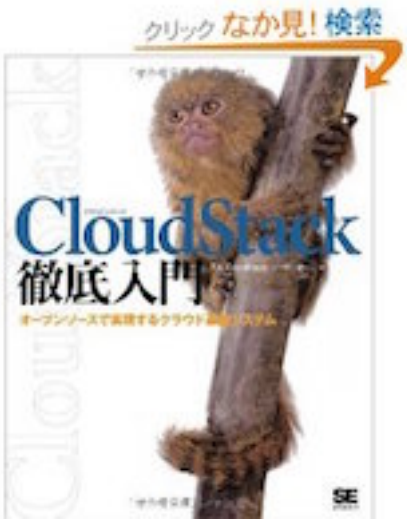
6.6.1 Documentation Available

The following guides are available:

- CloudStack Concepts and Terminology
- Quick Installation Guide
- Installation Guide
- Upgradong CloudStack
- Usage Guide
- Developers Guide
- Plugins Guide
- Release Notes

All at http://docs.cloudstack.apache.org/

6.6.2 Books



6.6.3 Commercial support

Some companies offer commercial support for Apache CloudStack or their own product based on CloudStack.

CHAPTER 7

Plugins Guide

This is the Apache CloudStack Plugins guide. This section gives information for those wishing to develop CloudStack either contributing to the CloudStack core software or writing external plugins

7.1 The Cloudian Connector Plugin

7.1.1 Introduction to the Cloudian Connector Plugin

The Cloudian (HyperStore) Connector is a native CloudStack plugin that allow integration between Apache CloudStack and Cloudian Management Console (CMC). The Connector integrates Cloudian S3 Storage into the CloudStack Management GUI and allows administrators to easily give their CloudStack users access to and manage their own S3 storage areas.

Compatibilty

The following table shows the compatiblity of Cloudian Connector with CloudStack.

Connector Version	CloudStack version	Cloudian Compatibility
4.9_6.2-1	4.9	Version 6.2 and onwards
4.11+	4.11+	Version 6.2 and onwards

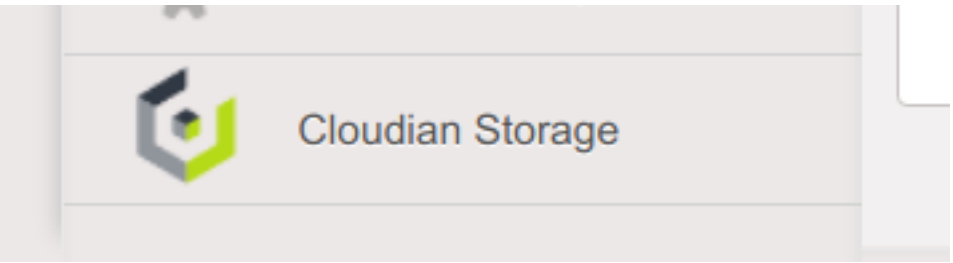
Table: Support Matrix

Note: Starting CloudStack 4.11, the Connector will be part of the CloudStack release and will not need to be externally installed.

7.1.2 Connector Overview

Single-Sign-On Integration

The connector plugin adds a Cloudian Storage button to the CloudStack UI. This button is available for all users on the bottom left of the menu.



When a user clicks this button, a new window or tab (depending on the web browser preferences) is opened for the HyperStore CMC GUI. The CloudStack user is automatically logged in to CMC as the correctly mapped HyperStore user using Single-Sign-On (SSO).

With the connector enabled, when the user clicks 'Log Out' in the CloudStack UI this first logs the user out of CloudStack and then redirects the page to log out any logged-in Cloudian user out of CMC GUI and finally redirects the page to the CloudStack login page.

Single-Sign-On is a technique where CloudStack and HyperStore are configured to trust each other. This is achieved by configuring both HyperStore and the CloudStack connector with the same SSO Shared Key. The CloudStack connector creates a special login URL for CMC which it signs using this shared key. Upon receiving the special signed login URL, CMC validates the request by comparing the signature to its own copy of the shared key and the user is automatically logged in.

User Mapping and Provisioning/De-provisioning

CloudStack domains are mapped to Cloudian Groups. CloudStack accounts within those domains are mapped to Cloudian users. The Cloudian user and group are created on demand if it doesn't already exist when the CloudStack user accesses CMC through the Cloudian Storage button. When accounts and domains are created or removed in CloudStack, they automatically create or remove users or groups in CMC.

CloudStack Entity	Equivalent Cloudian Entity
Account	User
Domain	Group

Table: Mapping Between Cloudian and CloudStack

Note: Adding groups or users directly through Cloudian does not add corresponding CloudStack Domains or Accounts. The integration is driven completely from the CloudStack side.

Special Admin User Mapping

The special CloudStack admin is are mapped to a special HyperStore Admin user account which defaults to the user id admin. As the admin user on HyperStore is configurable, there is a configuration option to control this mapping.

This mapping dictates which HyperStore user is automatically logged in using SSO when the CloudStack admin user clicks "Cloudian Storage".

Note: The Cloudian Admin user default is called admin. Older versions of Cloudian used to use admin@cloudian.com.

DNS Resolution Requirements

The CloudStack Management Server will need to be access the Cloudian admin service. The Cloudian admin service is commonly run on the same nodes as your Cloudian S3 servers. The admin service is used to provision and deprovision Cloudian users and groups automatically by CloudStack.

Additionally, your CloudStack users will need to be able to resolve your CMC server hostname on their desktops so that they can access CMC.

Example domain names that should resolve:

Resolvable Name	Required By	Description
mgmt.abc-cloud.com	User's browser	CloudStack Management Server
cmc.abc-cloud.com	User's browser	Cloudian CMC
admin.abc-cloud.com	Management Server	Cloudian Admin Server

Table: DNS Name Resolution Example

7.1.3 Configuring the Cloudian Connector

Prerequisites

Cloudian ships with SSO disabled by default. You will need to enable it on each CMC server. Additionally, you will need to choose a unique SSO shared key that you will also configure in the CloudStack connector further below.

Edit Puppet config to enable SSO on all CMC servers:

```
# vi /etc/cloudian-[version]-puppet/modules/cmc/templates/mts-ui.properties.
↪erb
  sso.enabled=true
  sso.shared.key=YourSecretKeyHere
```

Note: Once configured in Puppet, you should roll out out to each CMC server and restart CMC services. Please refer to the HyperStore documentation for how to do this.

Connector Configuration

The main way to configure, enable and disable the connector is using the CloudStack global setting. The global settings provide an easy way to configure the connector and synchronize setting across multiple management server(s). The following global setting can be accessed and changed using the CloudStack UI:

Global Setting	Description
cloudian.connector.enabled	Setting to enable/disable the plugin
cloudian.admin.host	The Cloudian admin server host
cloudian.admin.port	The Cloudian admin server port, usually 19443 (https) or 18081 (http)
cloudian.admin.protocol	The Cloudian admin server protocol, http/https
cloudian.validate.ssl	Whether to validate SSL certificate of Cloudian admin service while making API calls
cloudian.admin.user	Basic auth user name for Cloudian admin server
cloudian.admin.password	Basic auth password for Cloudian admin server
cloudian.api.request.timeout	The admin API request timeout in seconds
cloudian.cmc.admin.user	The user id of the CMC admin that maps to CloudStack admin user
cloudian.cmc.host	The Cloudian Management Console hostname
cloudian.cmc.port	The Cloudian Management Console port
cloudian.cmc.protocol	The Cloudian Management Console protocol
cloudian.sso.key	The shared secret as configured in Cloudian CMC

Table: Cloudian Connector Global Settings

Note: Change in only 'cloudian.connector.enabled' setting requires restarting of all the CloudStack management server(s), rest of the setting can be changed dynamically without requiring to restart the CloudStack management server(s).

Enabling the Cloudian Connector

The Cloudian Connector comes disabled by default, enabling the connector is the last step. You should have already configured the Cloudian Connector global settings. To enable the connector, ensure that the global setting "cloudian.connector.enabled" is set to true. Finally, restart each of the management server(s) to reload and enable the connector.

For example, here is how you can restart the CloudStack management server installed on CentOS7:

```
# systemctl restart cloudstack-management
```

Troubleshooting

Most of the trouble you may run into will be configuration related.

There are a few things which can go wrong for SSO. Here are the most common problems and things to check:

- Does the global settings cloudian.cmc.admin.user point to the correct Cloudian (admin) user?
- Is SSO configured and enabled on Cloudian HyperStore CMC?
- Check for errors in the CMC log file.
- Are both CloudStack and HyperStore CMC configured with the same cloudian.sso.key?
- Check the /var/log/cloudstack/management/management-server.log file and search for errors relating to SSO.
- Try access the CMC host directly from the problem users host using the configured cloudian.cmc.host, cloudian.cmc.port and cloudian.cmc.protocol configured in the CloudStack global settings.
- If you log out of the management server and log in again, does the Cloudian Storage button work?

Adding/Deleting Domains or Accounts fails: These operations use the Cloudian Admin Server. It's likely that something has changed with the connection or the admin server is down. Check list:

- Is the admin server alive and listening?
- Try access the admin server host directly from the problem users host using the configured cloudian.admin.host, cloudian.admin.port and cloudian.admin.protocol configured in the CloudStack global settings. Check the configured auth settings cloudian.admin.user and cloudian.admin.password.
- If you're experiencing timeout issues, trying changing the API timeout value defined in cloudian.api.request.timeout global setting.
- Look for errors in the admin log file /var/log/cloudian/cloudian-admin.log.

7.1.4 Cloudian as CloudStack Secondary Storage

This section is a supplementary guide for CloudStack and describes how to configure CloudStack to use Cloudian HyperStore as Secondary Storage. Please also review CloudStack's documentation (Getting Started Guide) for configuring and using S3 as Secondary Storage.

CloudStack, as of version 4.2.1, can utilize Cloudian HyperStore as S3 Secondary Storage out of the box. There is no need for any modifications or to install any connectors. Secondary Storage is used to store ISOs, Templates, Snapshots and Volumes.

Note: CloudStack still requires an NFS Secondary Storage Staging Server with is mentioned in the requirements below.

Requirements:

- CloudStack 4.5+ (installed/configured and running)
- Cloudian HyperStore 5.0 or greater (installed/configured and running)

NFS Secondary Storage Staging Server Requirement

The use of S3 as Secondary Storage for CloudStack also requires an NFS server. The NFS server is required because the various hypervisors cannot yet talk directly to S3 but instead talk through the standard file system API. As such, CloudStack requires an NFS staging server which the Hypervisors use to read and write data from/to. The NFS storage requirements for the staging server are small however as space is only required while objects are staged (moving) between the S3 server and the VMs.

DNS Name Resolution Requirement

All CloudStack Management Servers, system VMs and customer VMs (if required) must be able to resolve your S3 bucket names. Usually, if you already have Cloudian installed and running in your environment, this is already working. At a minimum the following names should resolve to the correct IP addresses using the DNS server that your Management Server and System VMs are using.

Example Name	DNS Name Types
s3.mycloud.com	Cloudian S3 Endpoint
sec.s3.mycloud.com	Bucket for Secondary Storage
s3-admin.mycloud.com	Cloudian Admin Server

Table: Example Domain Names that should Resolve on CloudStack Servers

7.1.5 Adding Cloudian as CloudStack Secondary Storage

Setup a Cloudian User and Bucket for Secondary Storage

S3 Secondary Storage stores the CloudStack templates, snapshots etc in a dedicated S3 Bucket. To properly configure CloudStack you will need to know the S3 Bucket name and how to access your S3 Server (the S3 endpoint, access key and secret key).

Below, we will create a dedicated Cloudian user and a dedicated bucket which we will assign for use as Secondary Storage.

Create a dedicated user/group:

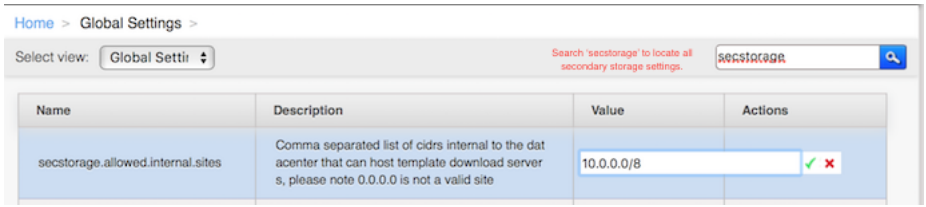
- Login to the Cloudian Management Console (CMC) as the Cloudian admin user.
- Create a new group called cloudstack. Any group name is ok.
- Create a new user called cloudstack in the cloudstack group. Any user name is ok.

Create a dedicated bucket:

- Login to CMC as the cloudstack user created above.
- Create a bucket called secondary. Any bucket name will do.
- On the top menu bar on the right hand side, use the drop down menu under your user name to select Security Credentials and copy and paste your Access and Secret Keys to a note for later use. CloudStack will need these when you attach Cloudian as Secondary Storage in a later step below.

Open Up Access to your S3 Network from Secondary Storage

If your S3 server is on a different network to your Secondary Storage VM, you will need to open up access to the S3 network. This also allows users to download templates from their S3 object store areas.



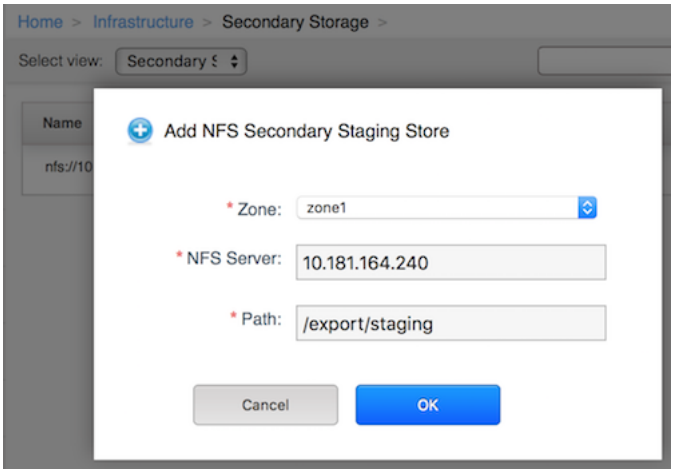
Note: Editing the Global Settings requires you to restart the management server(s).

Add an NFS Secondary Storage Staging Server

As mentioned previously, S3 Secondary Storage currently requires the use of an NFS Secondary Staging Server. Add NFS Secondary Storage Staging Server:

- Login to CloudStack Management Server as the admin user.
- Navigate to Infrastructure → Secondary Storage.
- Click Select View and select Secondary Staging Store.

- Click Add Secondary Staging Store.
- Configure the zone, server and path for your desired secondary staging store. For example nfs.mycloud.com and /export/staging.



Attach Cloudian as Secondary Storage

CloudStack supports using either S3 or NFS as Secondary Storage but not both. The below instructions assume you are not using Secondary Storage on NFS and that you can delete it to add the S3 storage.

Note: Already using NFS for Secondary Storage with CloudStack? You need to migrate your Secondary Storage. Refer to CloudStack's instructions for migrating existing NFS Secondary Storage to an S3 object storage. CloudStack 4.5 onwards supports migrating data via special commands which are described in the Getting Started Guide in a section titled Upgrading from NFS to Object Storage.

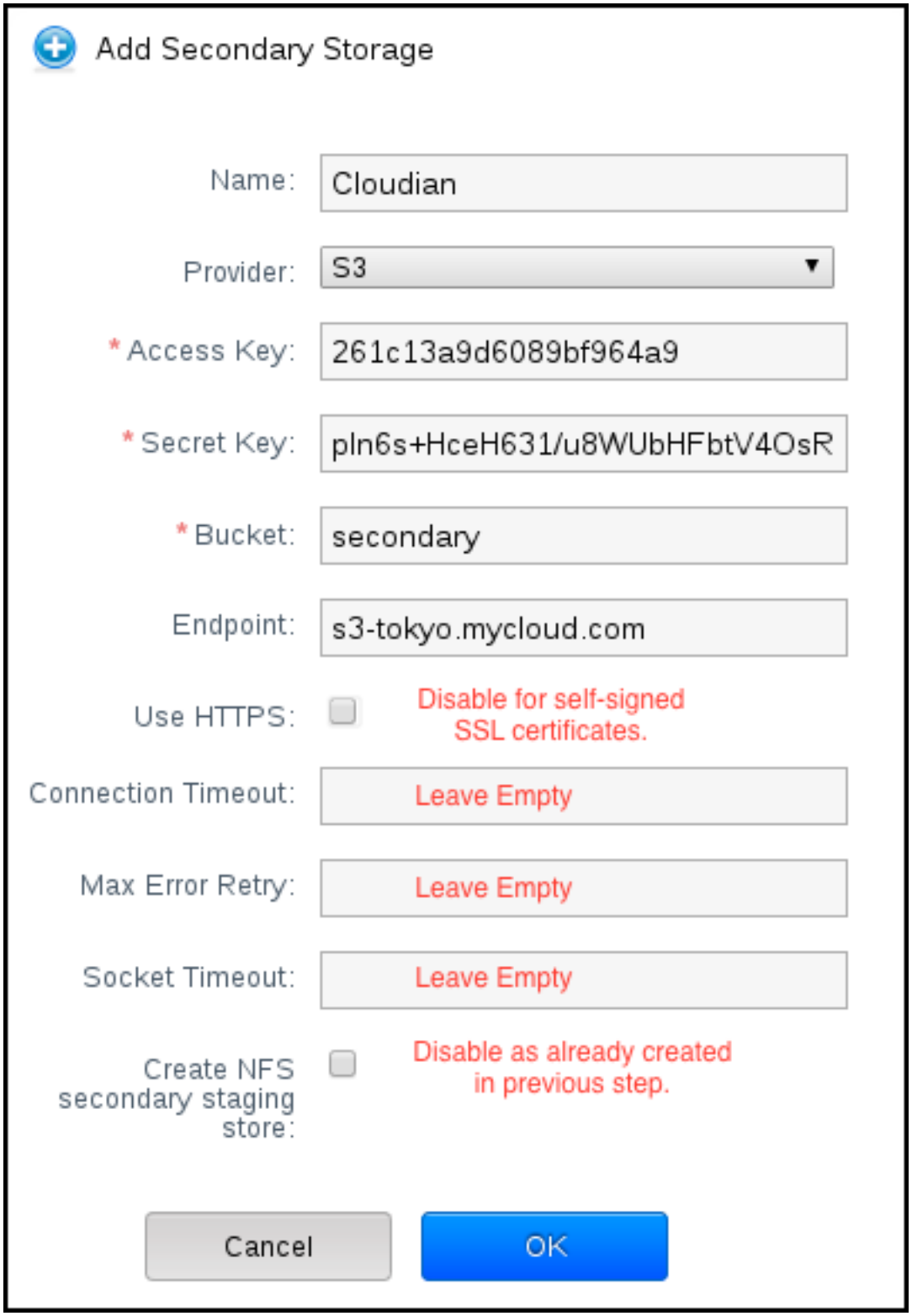
Adding S3 Secondary Storage:

- Login to CloudStack Management Server as the admin user.
- Navigate to Infrastructure → Secondary Storage.
- If it exists, select and delete any existing NFS Secondary Storage server setting. NOTE: Do not do this if you want to migrate existing NFS secondary storage to S3. Instead, see warning above.
- Click the Add Secondary Storage button. This will open up a pop-up form which you can fill out similarly to below.

Note: CloudStack doesn't currently allow you to re-edit the S3 configuration so take time to double check what you enter. If you make a mistake the only options currently are either a) delete and recreate the storage or b) directly edit the entry in the database.

When you have finished adding Cloudian as Secondary Storage in the previous steps, CloudStack will populate the new secondary storage with the system and default templates. This can take some time do download as the templates are quite big.

Note: You can check if the system template and the default template have properly downloaded to the new secondary storage by navigating to Templates, selecting a template, clicking on the Zones tab and checking its Status is Ready


Add Secondary Storage
Name:
Cloudian
Provider:
S3
* Access Key:
261c13a9d6089bf964a9
* Secret Key:
pln6s+HceH631/u8WUbHFbtV4OsR
* Bucket:
secondary
Endpoint:
s3-tokyo.mycloud.com
Use HTTPS:
Disable for self-signed SSL certificates.
Connection Timeout:
Leave Empty
Max Error Retry:
Leave Empty
Socket Timeout:
Leave Empty
Create NFS secondary staging store:
Disable as already created in previous step.
Cancel
OK

100% Downloaded.

Note: Should you continue to have problems, sometimes it is necessary to restart the Secondary Storage VM. You can do this by navigating to Infrastructure, System VMs, selecting and rebooting the Secondary Storage VM.

CloudStack should now ready to use Cloudian HyperStore for S3 Secondary Storage.

7.1.6 Revision History

- Fri Oct 6 2017 Rohit Yadav rohit.yadav@shapeblue.com Documentation created for 4.11.0 version of the Cloudian Connector Plugin

7.2 The Nicira NVP Plugin

7.2.1 Introduction to the Nicira NVP Plugin

The Nicira NVP plugin adds Nicira NVP as one of the available SDN implementations in CloudStack. With the plugin an exisiting Nicira NVP setup can be used by CloudStack to implement isolated guest networks and to provide additional services like routing and NAT.

Features of the Nicira NVP Plugin

The following table lists the CloudStack network services provided by the Nicira NVP Plugin.

Network Service	CloudStack version	NVP version
Virtual Networking	>= 4.0	>= 2.2.1
Source NAT	>= 4.1	>= 3.0.1
Static NAT	>= 4.1	>= 3.0.1
Port Forwarding	>= 4.1	>= 3.0.1

Table: Supported Services

Note: The Virtual Networking service was originally called 'Connectivity' in CloudStack 4.0

The following hypervisors are supported by the Nicira NVP Plugin.

Hypervisor	CloudStack version
XenServer	>= 4.0
KVM	>= 4.1

Table: Supported Hypervisors

Note: Please refer to the Nicira NVP configuration guide on how to prepare the hypervisors for Nicira NVP integration.

7.2.2 Configuring the Nicira NVP Plugin

Prerequisites

Before enabling the Nicira NVP plugin the NVP Controller needs to be configured. Please review the NVP User Guide on how to do that.

Make sure you have the following information ready:

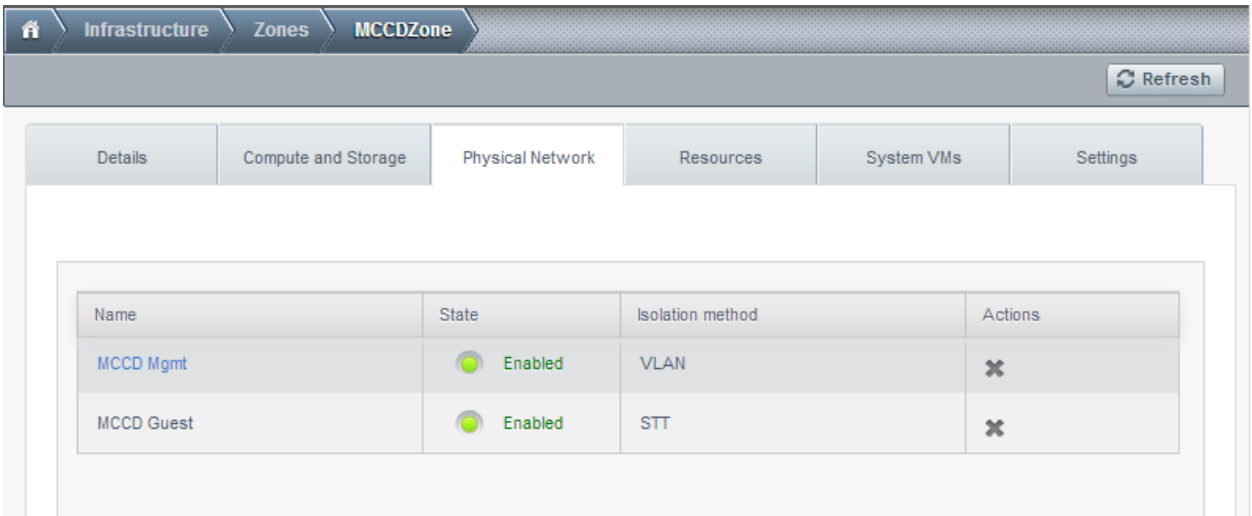
- The IP address of the NVP Controller
- The username to access the API
- The password to access the API
- The UUID of the Transport Zone that contains the hypervisors in this Zone
- The UUID of the Gateway Service used to provide router and NAT services.

Note: The gateway service uuid is optional and is used for Layer 3 services only (SourceNat, StaticNat and PortForwarding)

Zone Configuration

CloudStack needs to have at least one physical network with the isolation method set to "STT". This network should be enabled for the Guest traffic type.

Note: The Guest traffic type should be configured with the traffic label that matches the name of the Integration Bridge on the hypervisor. See the Nicira NVP User Guide for more details on how to set this up in XenServer or KVM.

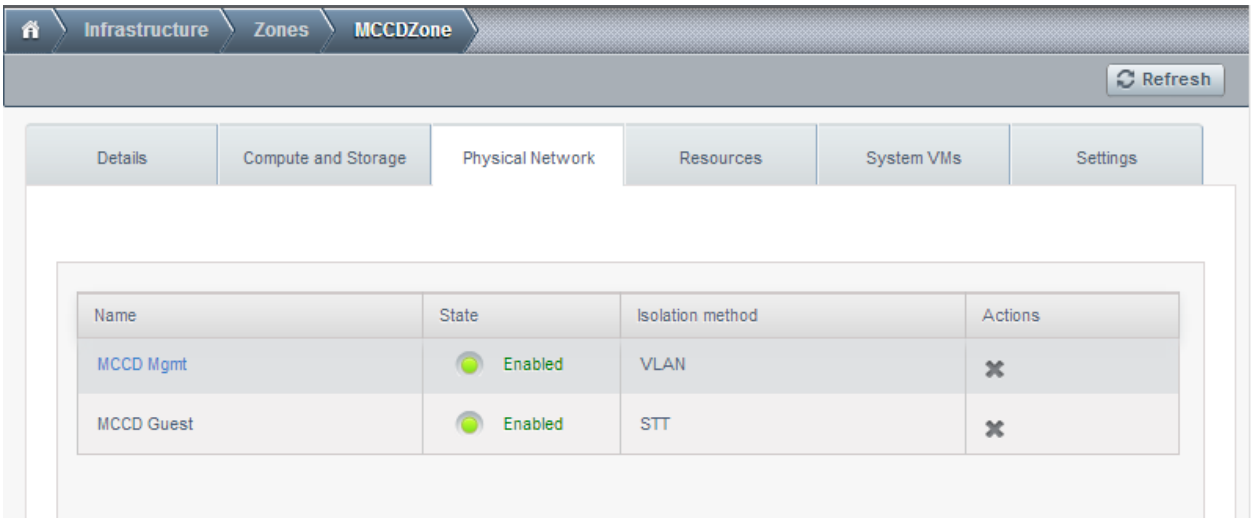


Enabling the service provider

The Nicira NVP provider is disabled by default. Navigate to the "Network Service Providers" configuration of the physical network with the STT isolation type. Navigate to the Nicira NVP provider and press the "Enable Provider"

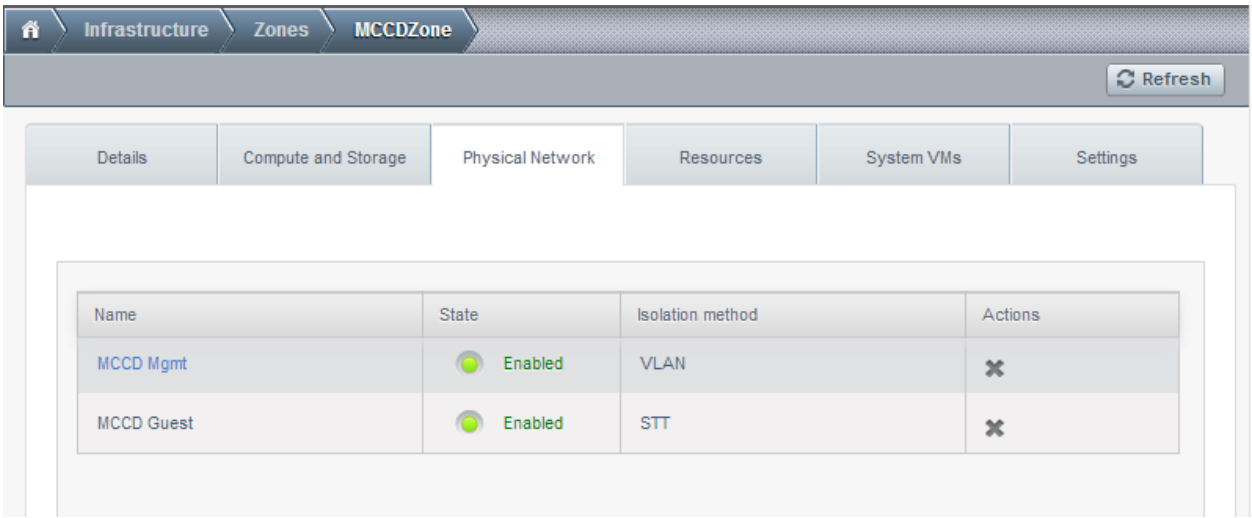
button.

Note: CloudStack 4.0 does not have the UI interface to configure the Nicira NVP plugin. Configuration needs to be done using the API directly.



Device Management

In CloudStack a Nicira NVP setup is considered a "device" that can be added and removed from a physical network. To complete the configuration of the Nicira NVP plugin a device needs to be added to the physical network. Press the "Add NVP Controller" button on the provider panel and enter the configuration details.

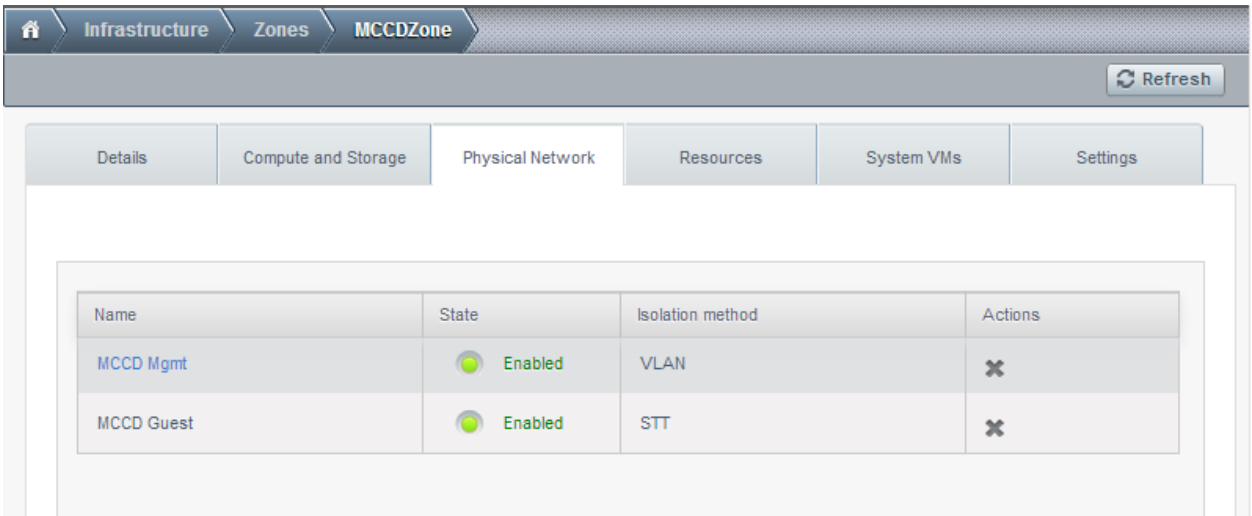


Network Offerings

Using the Nicira NVP plugin requires a network offering with Virtual Networking enabled and configured to use the NiciraNvp element. Typical use cases combine services from the Virtual Router appliance and the Nicira NVP plugin.

Service	Provider
VPN	VirtualRouter
DHCP	VirtualRouter
DNS	VirtualRouter
Firewall	VirtualRouter
Load Balancer	VirtualRouter
User Data	VirtualRouter
Source NAT	VirtualRouter
Static NAT	VirtualRouter
Post Forwarding	VirtualRouter
Virtual Networking	NiciraNVP

Table: Isolated network offering with regular services from the Virtual Router.



Note: The tag in the network offering should be set to the name of the physical network with the NVP provider.

Isolated network with network services. The virtual router is still required to provide network services like dns and dhcp.

Service	Provider
DHCP	VirtualRouter
DNS	VirtualRouter
User Data	VirtualRouter
Source NAT	NiciraNVP
Static NAT	NiciraNVP
Post Forwarding	NiciraNVP
Virtual Networking	NiciraNVP

Table: Isolated network offering with network services

7.2.3 Using the Nicira NVP plugin with VPC

Supported VPC features

The Nicira NVP plugin supports CloudStack VPC to a certain extent. Starting with CloudStack version 4.1 VPCs can be deployed using NVP isolated networks.

It is not possible to use a Nicira NVP Logical Router for as a VPC Router

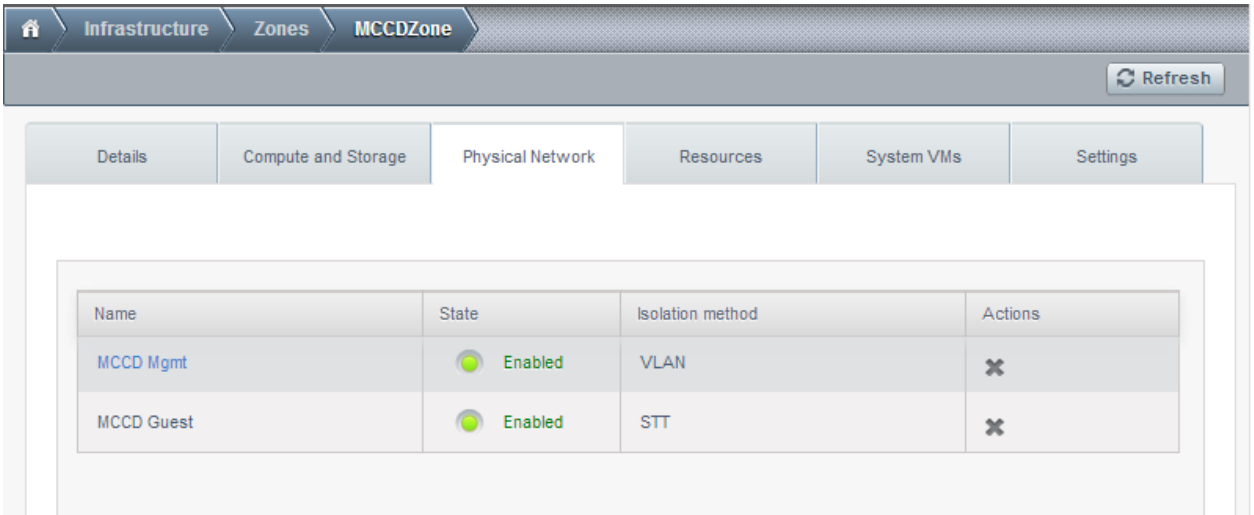
It is not possible to connect a private gateway using a Nicira NVP Logical Switch

VPC Offering with Nicira NVP

To allow a VPC to use the Nicira NVP plugin to provision networks, a new VPC offering needs to be created which allows the Virtual Networking service to be implemented by NiciraNVP.

This is not currently possible with the UI. The API does provide the proper calls to create a VPC offering with Virtual Networking enabled. However due to a limitation in the 4.1 API it is not possible to select the provider for this network service. To configure the VPC offering with the NiciraNVP provider edit the database table 'vpc_offering_service_map' and change the provider to NiciraNvp for the service 'Connectivity'

It is also possible to update the default VPC offering by adding a row to the 'vpc_offering_service_map' with service 'Connectivity' and provider 'NiciraNvp'



Note: When creating a new VPC offering please note that the UI does not allow you to select a VPC offering yet. The VPC needs to be created using the API with the offering UUID.

VPC Network Offerings

The VPC needs specific network offerings with the VPC flag enabled. Otherwise these network offerings are identical to regular network offerings. To allow VPC networks with a Nicira NVP isolated network the offerings need to support the Virtual Networking service with the NiciraNVP provider.

In a typical configuration two network offerings need to be created. One with the loadbalancing service enabled and one without loadbalancing.

Service	Provider
VPN	VpcVirtualRouter
DHCP	VpcVirtualRouter
DNS	VpcVirtualRouter
Load Balancer	VpcVirtualRouter
User Data	VpcVirtualRouter
Source NAT	VpcVirtualRouter
Static NAT	VpcVirtualRouter
Post Forwarding	VpcVirtualRouter
NetworkACL	VpcVirtualRouter
Virtual Networking	NiciraNVP

Table: VPC Network Offering with Loadbalancing

7.2.4 Troubleshooting the Nicira NVP Plugin

UUID References

The plugin maintains several references in the CloudStack database to items created on the NVP Controller.

Every guest network that is created will have its broadcast type set to Lswitch and if the network is in state "Implemented", the broadcast URI will have the UUID of the Logical Switch that was created for this network on the NVP Controller.

The Nics that are connected to one of the Logical Switches will have their Logical Switch Port UUID listed in the nicira_nvp_nic_map table

Note: All devices created on the NVP Controller will have a tag set to domain-account of the owner of the network, this string can be used to search for items in the NVP Controller.

Database tables

The following tables are added to the cloud database for the Nicira NVP Plugin

id	auto incrementing id
logicalswitch	uuid of the logical switch this port is connected to
logicalswitchport	uuid of the logical switch port for this nic
nic	the CloudStack uuid for this nic, reference to the nics table

Table: nicira_nvp_nic_map

id	auto incrementing id
uuid	UUID identifying this device
physical_network_id	the physical network this device is configured on
provider_name	NiciraNVP
device_name	display name for this device
host_id	reference to the host table with the device configuration

Table: external_nicira_nvp_devices

id	auto incrementing id
logicalrouter_uuid	uuid of the logical router
network_id	id of the network this router is linked to

Table: nicira_nvp_router_map

Note: nicira_nvp_router_map is only available in CloudStack 4.1 and above

7.2.5 Revision History

0-0 Wed Oct 03 2012 Hugo Trippaers hugo@apache.org Documentation created for 4.0.0-incubating version of the NVP Plugin 1-0 Wed May 22 2013 Hugo Trippaers hugo@apache.org Documentation updated for CloudStack 4.1.0

7.3 The Nuage VSP Plugin

7.3.1 Introduction

The Nuage VSP Plugin is the Nuage Networks SDN implementation in CloudStack, which integrates with Nuage Networks Virtualized Services Platform (VSP). The plugin can be used by CloudStack to leverage the scalability and rich features of advanced SDN being provided by the Nuage VSP SDN Platform and to implement:

- Isolated Guest Networks
- Virtual Private Clouds (VPC)
- Shared Networks

For more information about Nuage Networks, visit www.nuagenetworks.net.

Supported Features

The following table lists the supported Network services in a CloudStack deployment with NuageVsp being the Connectivity/Virtual Networking provider, with their providers and supported CloudStack versions.

Network Service	Isolated Networks	VPCs	Shared Networks
Virtual Networking	NuageVsp (>=4.5)	NuageVsp (>=4.5)	NuageVsp (>=4.10)
Dhcp	NuageVsp (>=4.5)	NuageVsp (>=4.5)	NuageVsp (>=4.10)
SourceNat	NuageVsp (>=4.10)	NuageVsp (>=4.10)	N/A
StaticNat	NuageVsp (>=4.5)	NuageVsp (>=4.5)	N/A
Firewall	NuageVsp (>=4.5)	N/A	N/A
NetworkACL	N/A	NuageVsp (>=4.5)	N/A
UserData	VirtualRouter (>=4.5)	VpcVirtualRouter (>=4.5)	VirtualRouter (>=4.10)
Dns	VirtualRouter (>=4.10)	VpcVirtualRouter (>=4.10)	N/A
Internal Lb	N/A	InternalLbVm (>=4.9)	N/A

Table: Supported Network Services

Note: The Virtual Networking service was originally called 'Connectivity' in CloudStack 4.0

Supported Hypervisors

The following hypervisors are supported by the Nuage VSP Plugin, with their supported CloudStack versions.

Hypervisor	CloudStack version
KVM 7.x	>= 4.5
VMware ESXi 5.5	>= 4.5
VMware ESXi 6.0	>= 4.9

Table: Supported Hypervisors

Supported Nuage VSP SDN Platform Versions

The following Nuage VSP SDN Platform versions are supported by the Nuage VSP Plugin, with their supported CloudStack versions.

Nuage VSP version	CloudStack version
Nuage VSP v3.2	>= 4.5
Nuage VSP v4.0	>= 4.10

Table: Supported Nuage VSP SDN Platform Versions

7.3.2 Configuring The Nuage VSP Plugin

Prerequisites

Before enabling and using the Nuage VSP Plugin with CloudStack.

1. Verify that the CloudStack deployment (hypervisors) and Nuage VSP SDN Platform version you intend to use is being supported.

Note: Only the release notes for Nuage VSP contain the most up-to-date information on different supported versions. Please check them to verify that the information in this document is up-to-date.

2. Prepare and configure the hypervisors for CloudStack integration with Nuage VSP SDN Platform.

Note: Please refer to the Nuage VSP Install Guide on how to prepare the hypervisors for Nuage VSP SDN Platform integration.

Required Nuage VSD Configuration

When configuring Nuage VSP as the network service provider in a CloudStack Zone, a CSP user must be added in Nuage VSD, and this user must be added to the CMS group. See *Enable Nuage VSP Network Service Provider*.

Note: Nuage VSD is the programmable policy and analytics engine of the Nuage VSP SDN Platform with which the Nuage VSP Plugin interacts.

Zone Configuration

Select VSP Isolation Method

The Nuage VSP solution is NOT supported in Basic zone provisioning mode.

1. When adding a zone, the CloudStack administrator should select **Advanced** mode in the zone wizard.
2. When laying out the physical network configuration during zone provisioning, the **Guest** network traffic should be put in a separate physical network of its own.
3. This physical network carrying the **Guest** traffic should have **VSP** as the **Isolation Method**.

Fig. 1: Setting Isolation Method to VSP

Update Traffic Labels

Guest Traffic Type

Select **Edit** on the **Guest** traffic type panel and update the Traffic Label:

- For KVM, use **alubr0** as the **KVM Traffic Label**.

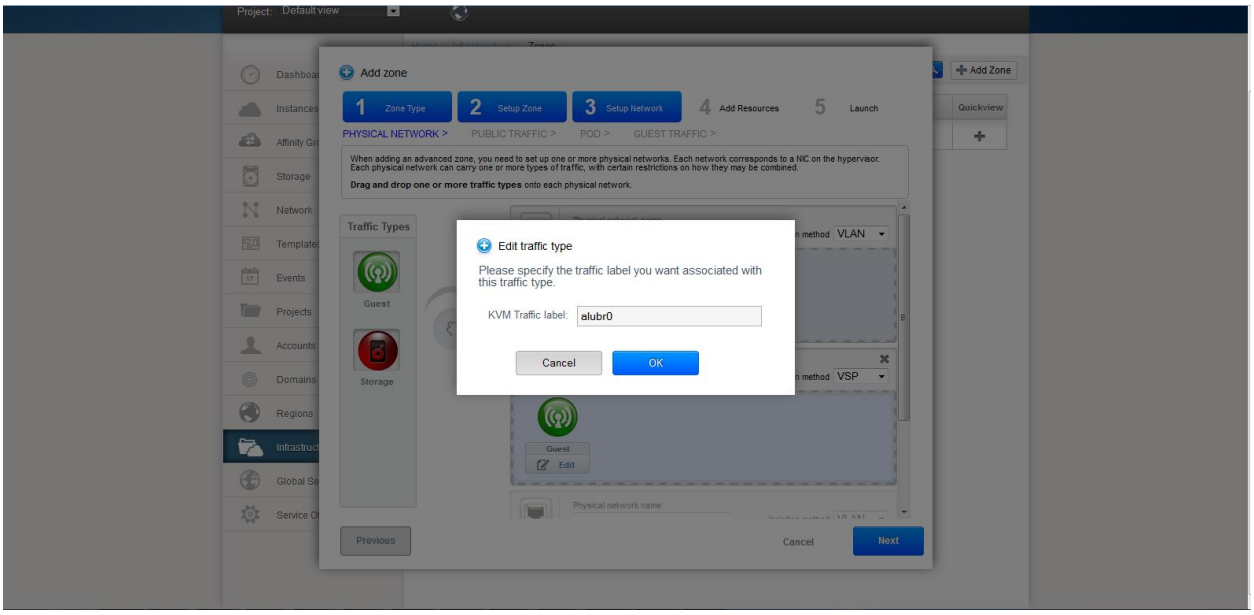


Fig. 2: Specifying the Traffic Type in KVM

- For VMware ESXi, use the switch name used by **dVRS** for guest networking as the **vSwitch Name**, leave the **VLAN ID** field blank, and select **VMware vNetwork Distributed Switch** in the **vSwitch Type** drop down field.

Enable Nuage VSP Network Service Provider

Nuage VSP must be added and enabled as a Network Service Provider in the CloudStack Zone before it can be used.

Step 1 Select **Infrastructure > Zone > [zone name] > Physical Network 2 > Configure Network Service Providers > Nuage Vsp > +**, which brings up the **Add Nuage Virtualized Services Directory (VSD)** panel.

Step 2 Enter the Nuage VSD **Host Name**, **Username** and **Password** that was previously created.

Step 3 Specify the Nuage VSD API version by entering the API version in the appropriate field (format: `v4_0`).

Step 4 *EITHER* Add **Nuage VSD** by clicking the **OK** button,

OR use Nuage VSP API calls to configure Nuage VSP as a Network Service Provider in the CloudStack Zone; see *Configure Nuage VSP API* in the Appendix of this document.

Step 5 Go to **Infrastructure > Zones > [zone name] > Physical Network 2 > Network Service Providers > Nuage Vsp > Devices > Details** tab as shown in the figure "Enabling Nuage VSP Network Service Provider" below. This indicates the state of Nuage VSP Network Service Provider. Enable Nuage VSP Network Service Provider by clicking **Enable**.

Step 6 (Optional) View the Nuage VSP Network Service Provider status on the list of Network Service Providers on the **Infrastructure > Zones > [zone name] > Physical Network 2 > Network Service Providers** page;

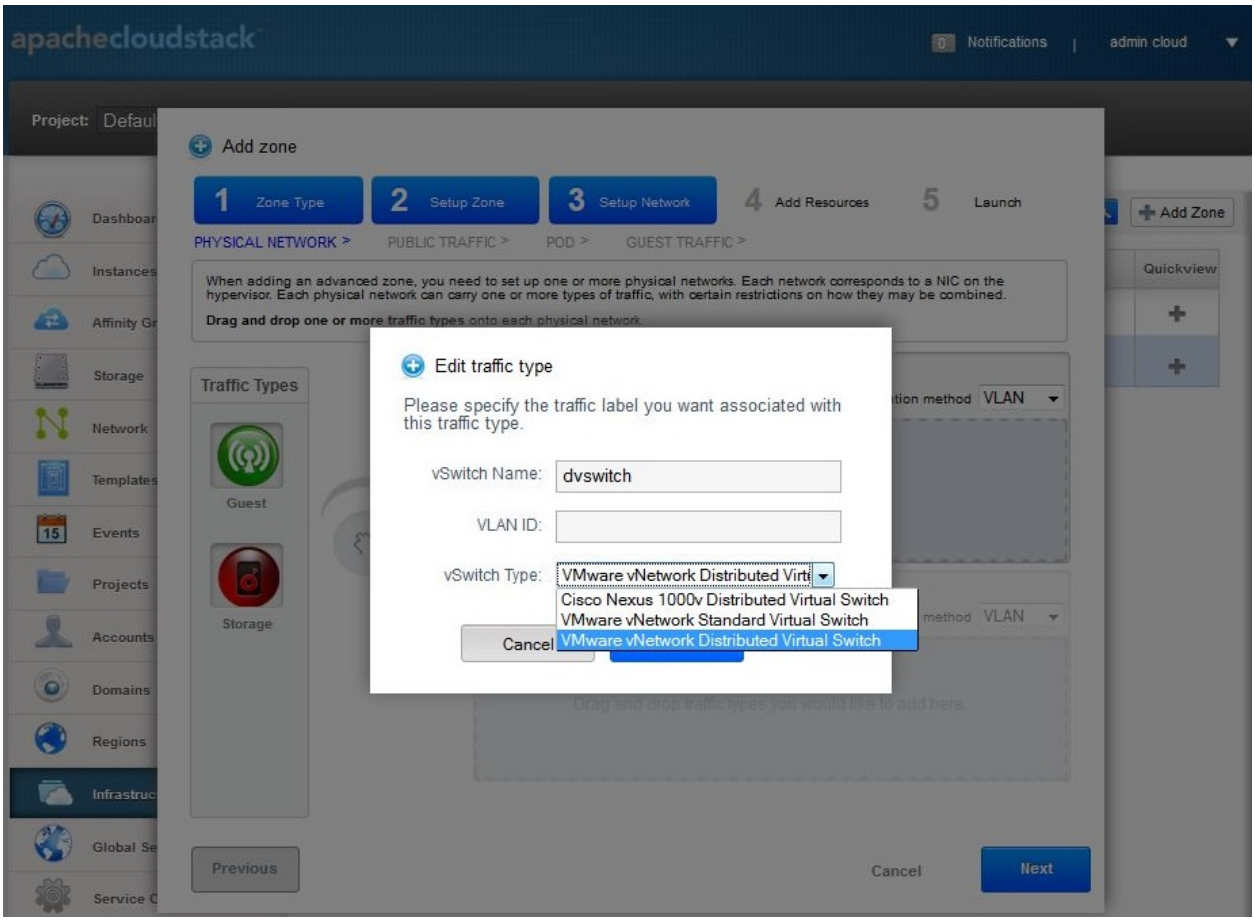


Fig. 3: Specifying the Traffic Type in VMware ESXi

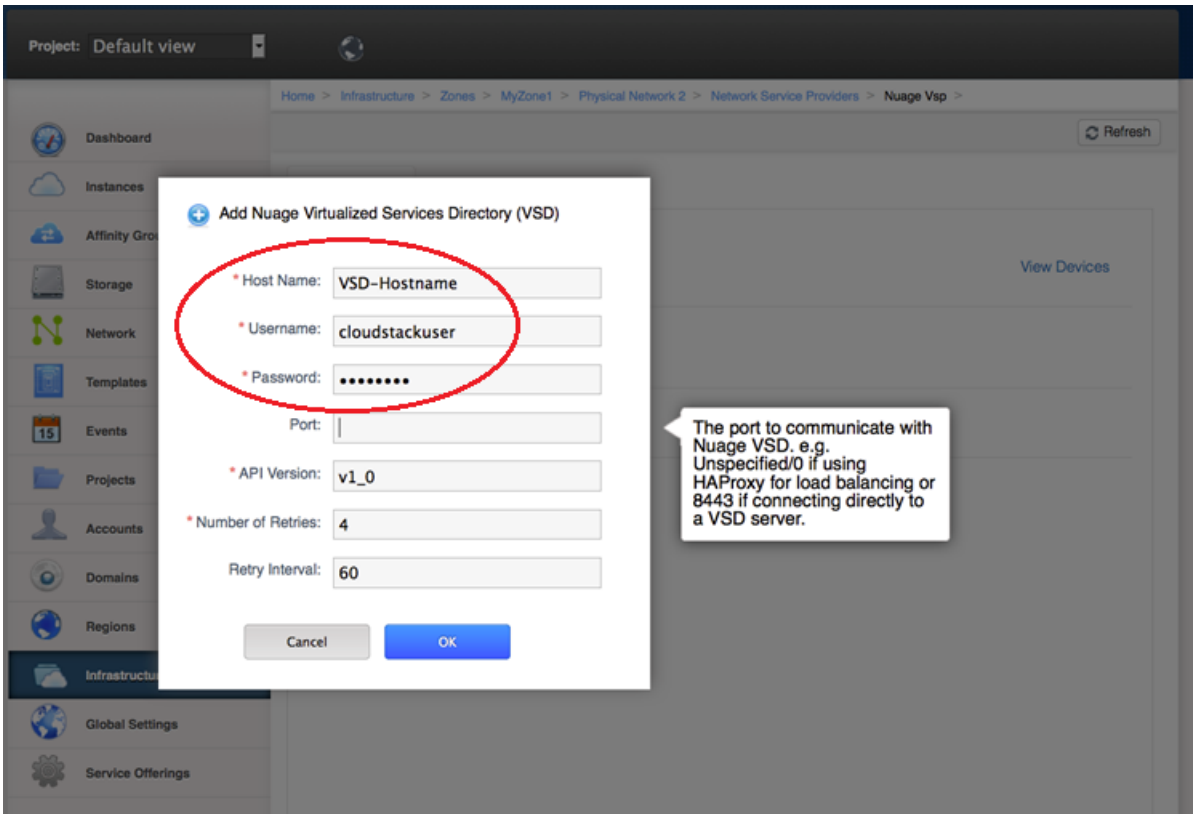


Fig. 4: Adding Nuage VSD as the Network Service Provider

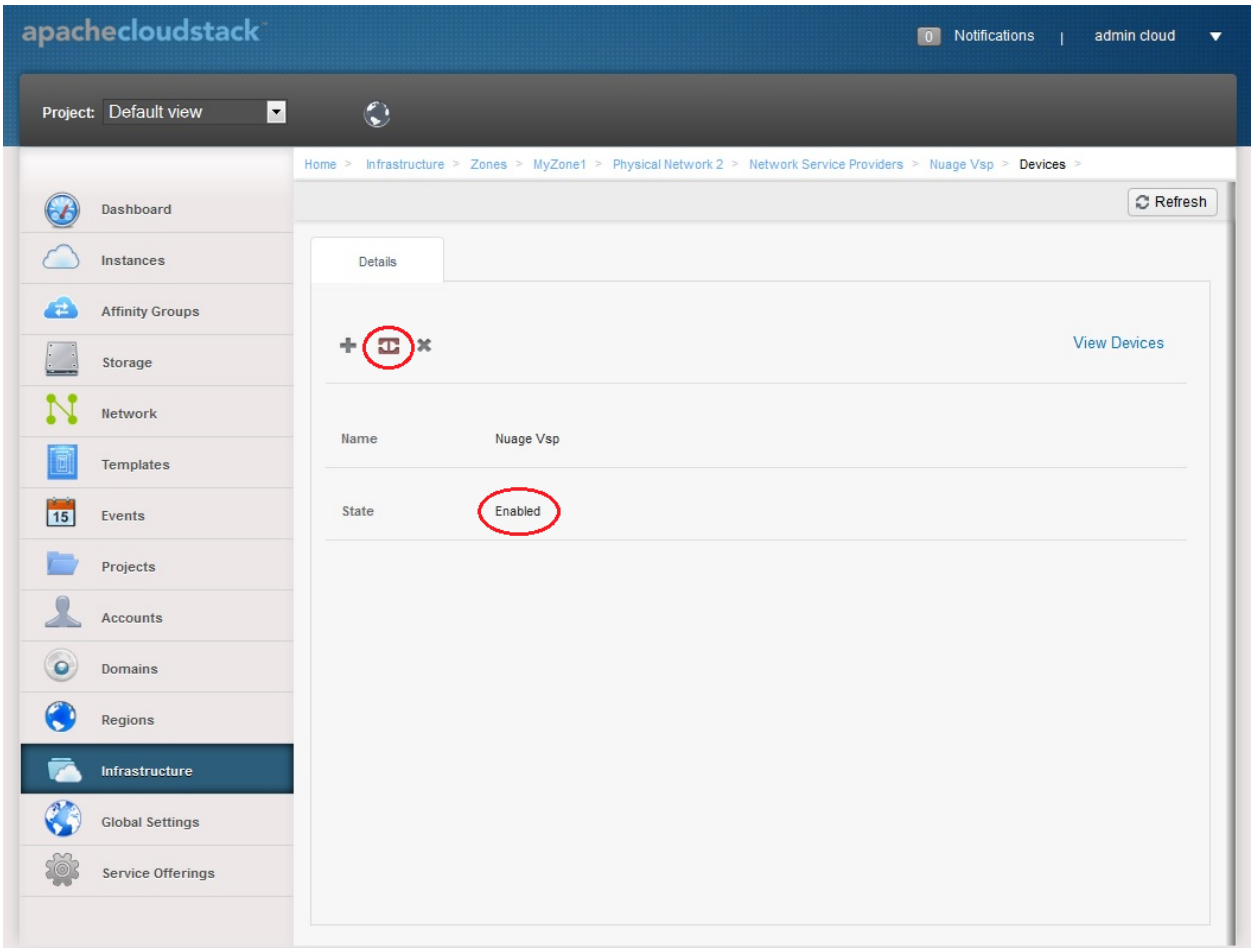


Fig. 5: Enabling Nuage VSP Network Service Provider

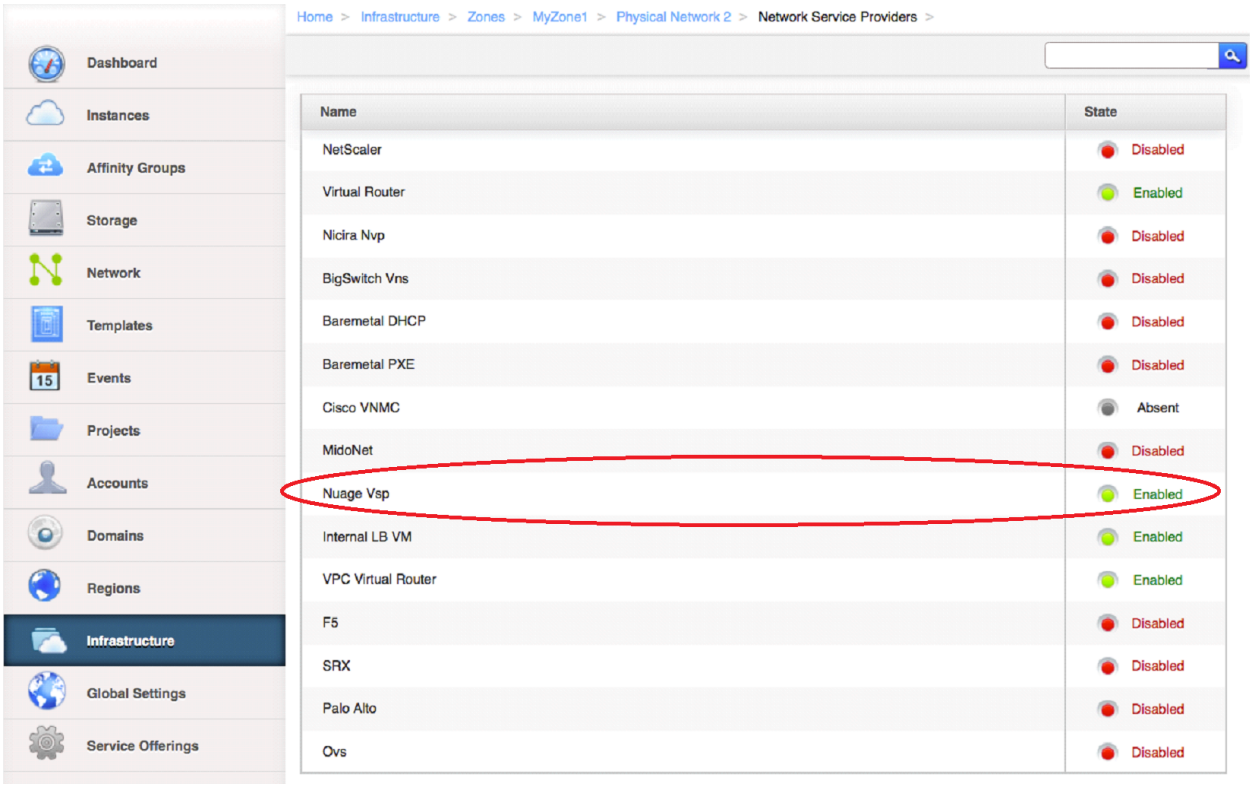


Fig. 6: Viewing Network Service Providers Status

7.3.3 Using The Nuage VSP Plugin

Network Offerings

There are three types of Network Offerings that can be created:

- If Isolated Networks are required, then create a **Isolated** guest type network offering for use with Isolated Networks.
- If VPC deployments are required, then create a new **Isolated** guest type network offering for such deployments.
- If Shared Networks are required, then create a new **Shared** guest type network offering for use with Shared Networks.

Note: **Per Zone** MUST always be selected as the **Supported Source NAT type** when **Source NAT** service is being provided by **NuageVsp**.

Create and Enable Isolated Network Offering

1. Select **Service Offerings > Select Offering: Network Offerings > Add network offering**, which brings up the **Add network offering**.
2. In the **Add network offering** panel, add a **Name** and a **Description** to the network offering. Select **Isolated** as the **Guest Type**. In the **Supported Services** field select services and providers that are supported by the Nuage VSP Plugin for Isolated Networks, see *Supported Features* at the beginning of this document.

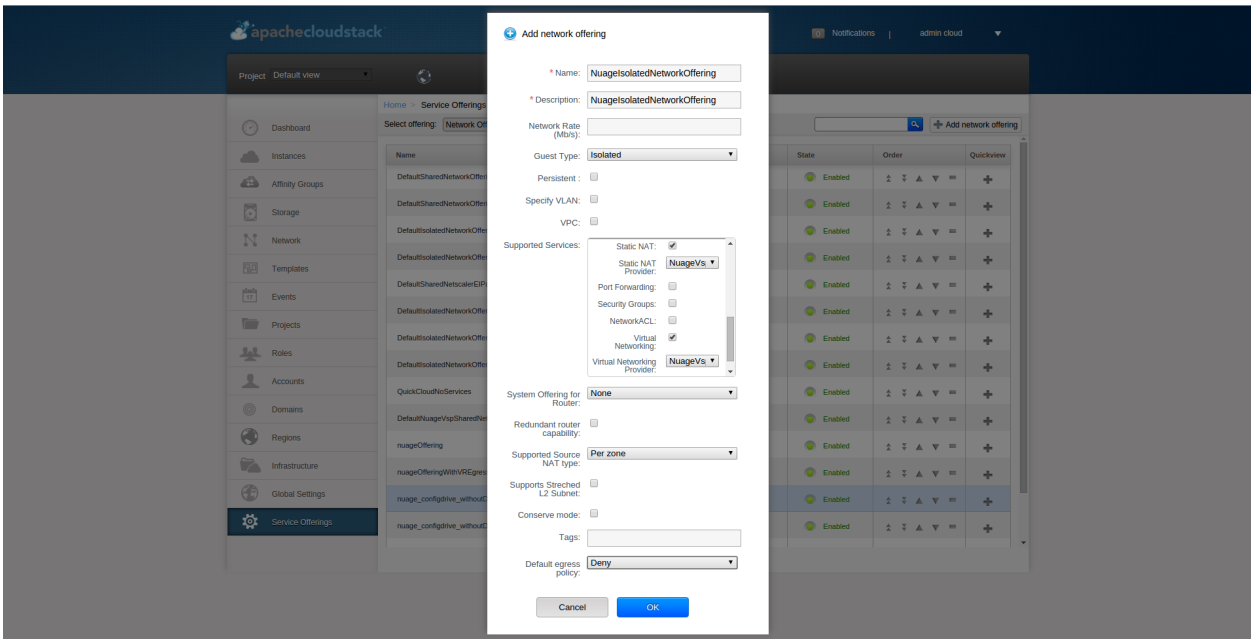


Fig. 7: Creating Isolated Network Offering

3. Click the **OK** button to create the network offering.
4. After the network offering has been successfully created, enable it from the **Service Offerings - Network Offerings** list.

Create and Enable VPC Network Offering

1. Select **Service Offerings > Select Offering: Network Offerings > Add network offering**, which brings up the **Add network offering**.
2. In the **Add network offering** panel, add a **Name** and a **Description** to the network offering. Select **Isolated** as the **Guest Type**. Select the **VPC** field. In the **Supported Services** field select services and providers that are supported by the Nuage VSP Plugin for VPCs, see *Supported Features* at the beginning of this document.
3. Click the **OK** button to create the network offering.
4. After the network offering has been successfully created, enable it from the **Service Offerings - Network Offerings** list.

Create and Enable Shared Network Offering

1. Select **Service Offerings > Select Offering: Network Offerings > Add network offering**, which brings up the **Add network offering**.
2. In the **Add network offering** panel, add a **Name** and a **Description** to the network offering. Select **Shared** as the **Guest Type**. In the **Supported Services** field select services and providers that are supported by the Nuage VSP Plugin for Shared Networks, see *Supported Features* at the beginning of this document.

Note: Selecting the **Supporting Public Access** field in the Shared Network offering enables Public/Internet access to the VMs in the Shared Network.

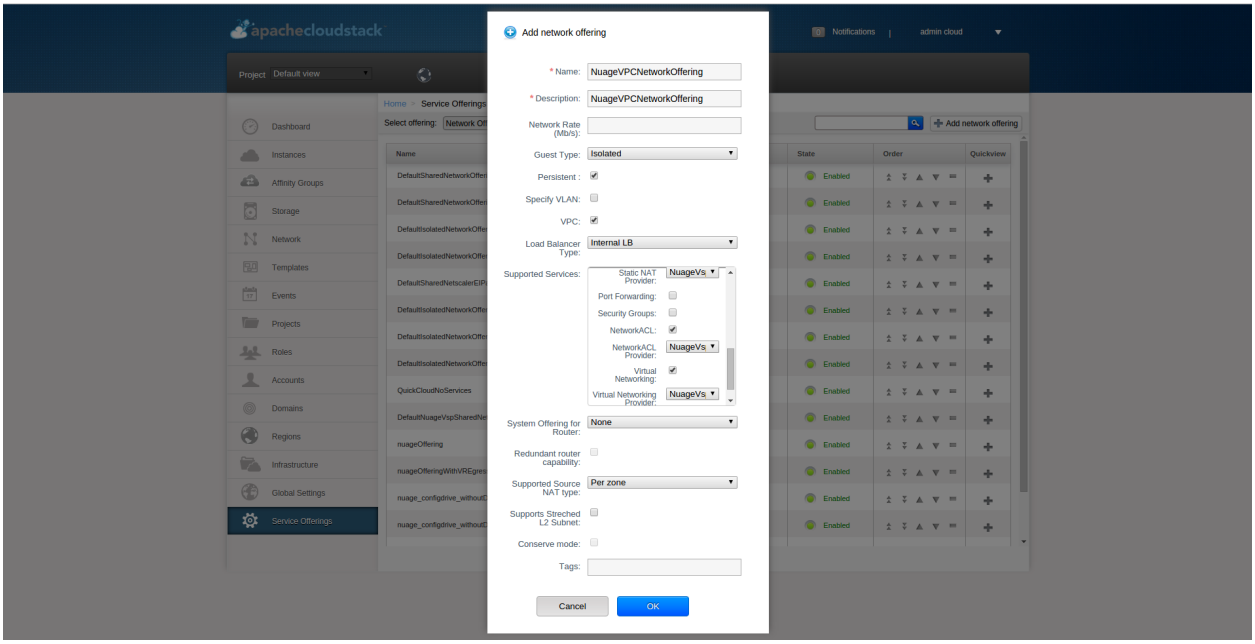


Fig. 8: Creating VPC Network Offering

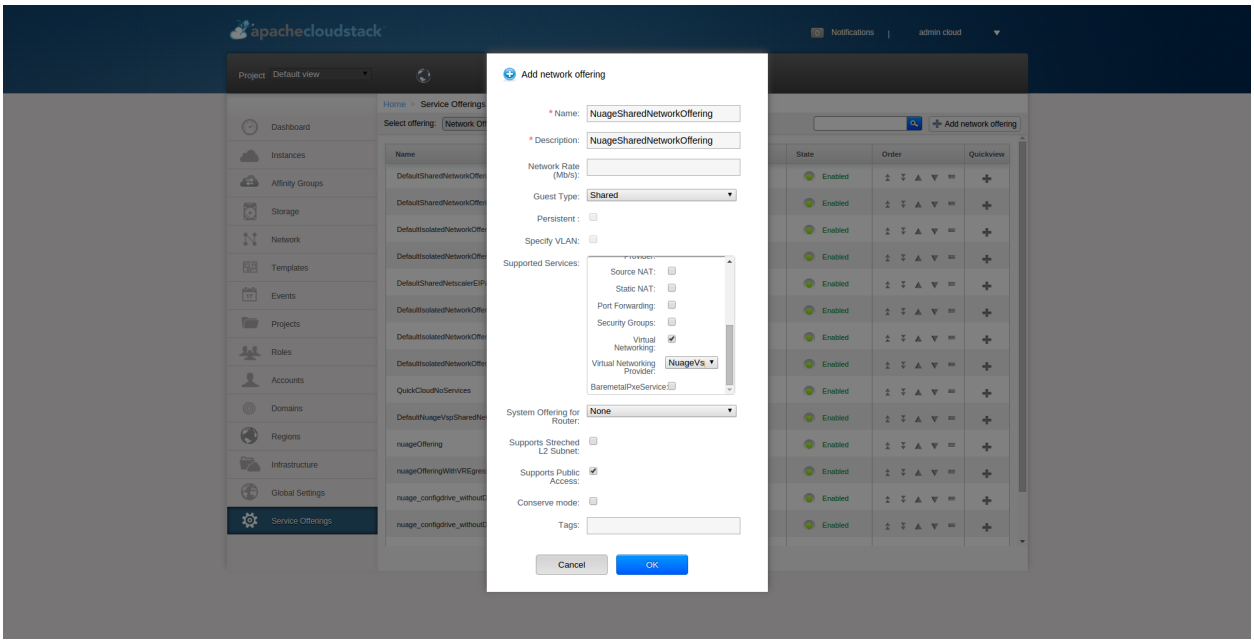


Fig. 9: Creating Shared Network Offering

3. Click the **OK** button to create the network offering.
4. After the network offering has been successfully created, enable it from the **Service Offerings - Network Offerings** list.

VPC Offerings

Pre-created and Enabled Nuage VSP VPC Offering

A VPC offering by the name **Nuage VSP VPC Offering** is pre-created and enabled in the list of **Service Offerings - VPC Offerings** (Select **Service Offerings > Select Offering: VPC Offerings**) which contains all the services and providers that are supported by the Nuage VSP Plugin for VPCs.

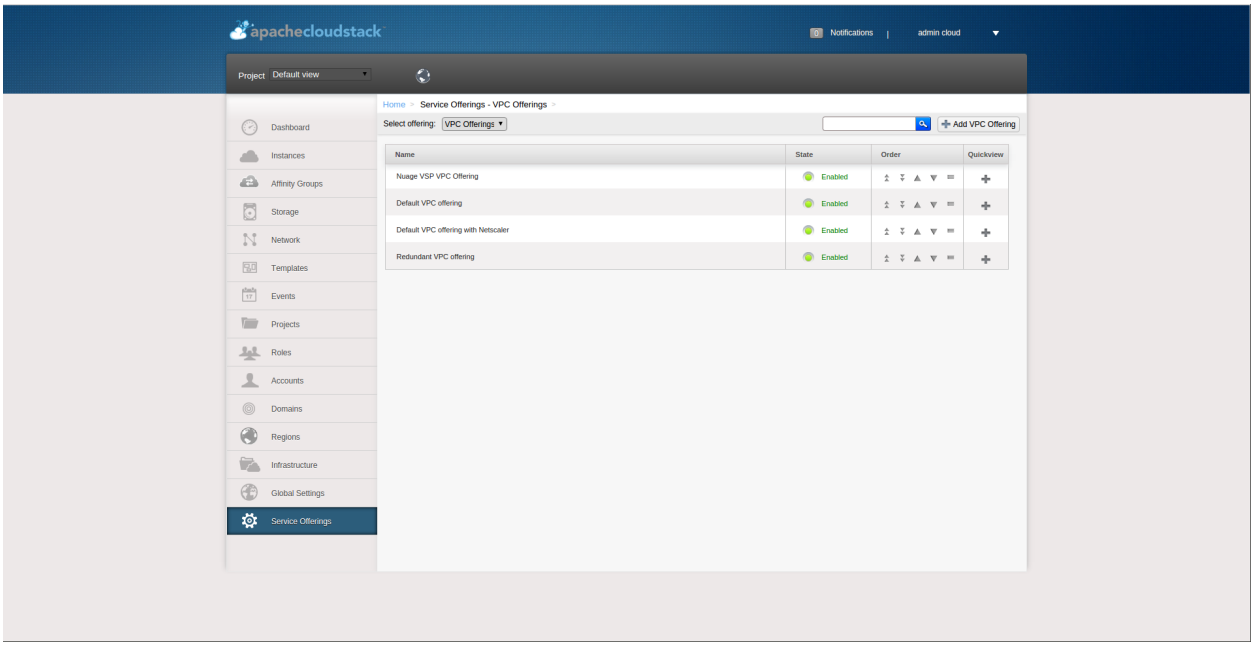


Fig. 10: Pre-created and Enabled Nuage VSP VPC Offering

(Optional) Create and Enable VPC Offering

1. Select **Service Offerings > Select Offering: VPC Offerings > Add VPC Offering**, which brings up the **Add VPC Offering**.
2. In the **Add VPC Offering** panel, add a **Name** and a **Description** to the network offering. In the **Supported Services** field select services and providers that are supported by the Nuage VSP Plugin for VPCs, see *Supported Features* at the beginning of this document.
3. Click the **OK** button to create the VPC Offering.
4. After the VPC Offering has been successfully created, enable it from the **Service Offerings - VPC Offerings** list.

7.3.4 Dedicated Features Provided by The Nuage VSP Plugin

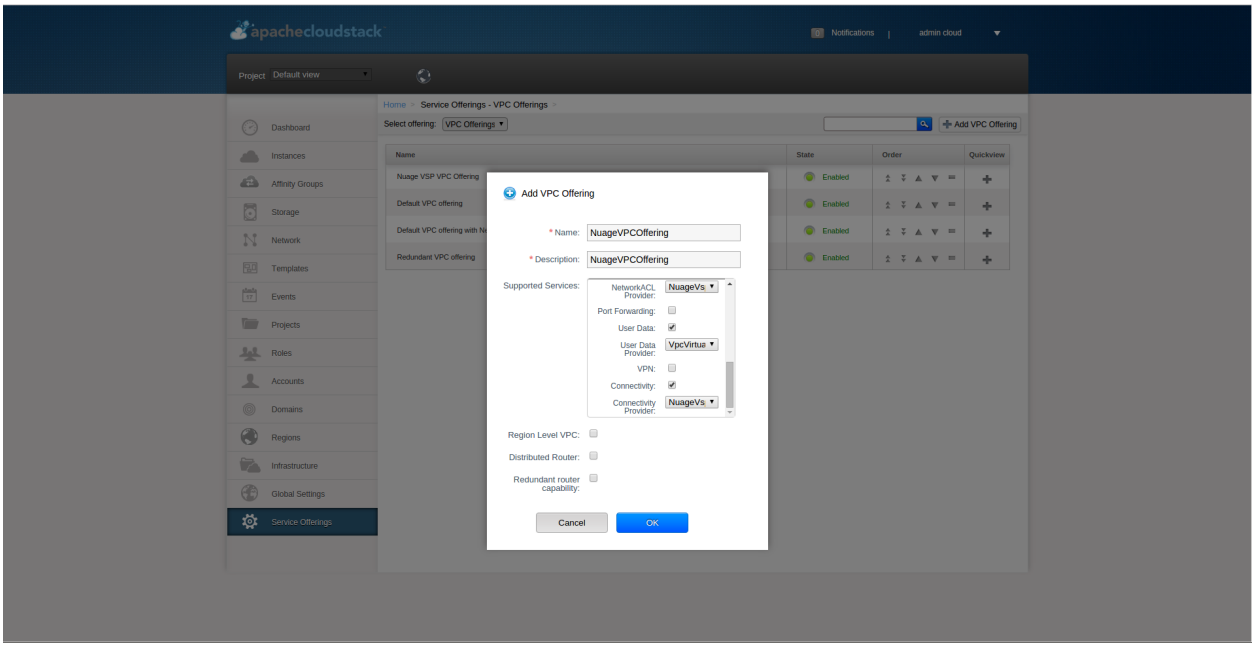


Fig. 11: Creating VPC Offering

Nuage VSP Domain Template Feature Support for CloudStack

All the constructs (parameters and abstractions) defined in a Nuage VSD domain template can be made available to domain instances (i.e. networks) created in CloudStack. To do this, configure the Nuage VSP Plugin to use a pre-created Nuage VSD domain template when instantiating domains (i.e. creating networks). Networks created in CloudStack will then use domain instances created from the domain template.

Typical use-cases are:

- The basic ACLs on the top and bottom that bracket or 'contain' the end-user's ACLs.
- Leakable domains/GRT Leaking (Nuage VSP feature).

To configure a Nuage VSP domain template for use by CloudStack, use the Nuage VSD Architect (VSP's GUI) to create a domain template and configure it in the following CloudStack global settings.

Parameter	Type	Explanation	Supported CloudStack versions
nuagevsp.isolatedntwk.domaintemplate.name	String	Name of the Nuage VSP domain template to use for creating domains for isolated networks	>= 4.5
nuagevsp.vpc.domaintemplate.name	String	Name of the Nuage VSP domain template to use for creating the domain for VPCs	>= 4.5
nuagevsp.sharedntwk.domaintemplate.id	UUID	UUID of the Nuage VSP domain template to use for creating the domain for Shared Networks	>= 4.10

Table: CloudStack Global Settings For Configuring Nuage VSP Domain Template Feature

Nuage VSP Source NAT via the Underlay Feature Support For CloudStack

Supported CloudStack versions: >= 4.10

CloudStack provides Source NAT service to enable guest VMs to send traffic out to the Internet without requiring a Static NAT IP (public IP) assigned to the VM. The Source NAT service must be enabled as part of the network offering used for creating the guest network. When a network is created using this network offering, the first public IP from the assigned public IP range is automatically acquired as the Source NAT IP for the network. All VMs attached to this network then use that Source NAT IP to send traffic to the Internet.

The Nuage VSP Plugin for CloudStack supports CloudStack's native Source NAT service and enhances it by restricting to a minimum the number of public IP addresses assigned to any given tenant. This is achieved by not allocating a Source NAT IP for every network that is created.

The Source NAT service that Nuage VSP calls the Port Address Translation (PAT) feature uses the hypervisor IP as the Source NAT IP address for all VMs in the hypervisor that need to send traffic out to the Internet. Configure this during Nuage VSP installation using the instructions given in the Nuage VSP Install Guide.

This feature is supported for both VPCs and Isolated Networks. In the case of VPCs, Source NAT is applied at the Nuage VSP domain level, therefore there is no customization on the individual VPC network (tier) level.

All VPCs and Isolated networks that are created from a Nuage VSP Source NAT-enabled network offering have this feature enabled automatically. An example Nuage VSP Source NAT-enabled network offering is shown in the figure "Nuage VSP Source NAT-enabled Network Offering" below.

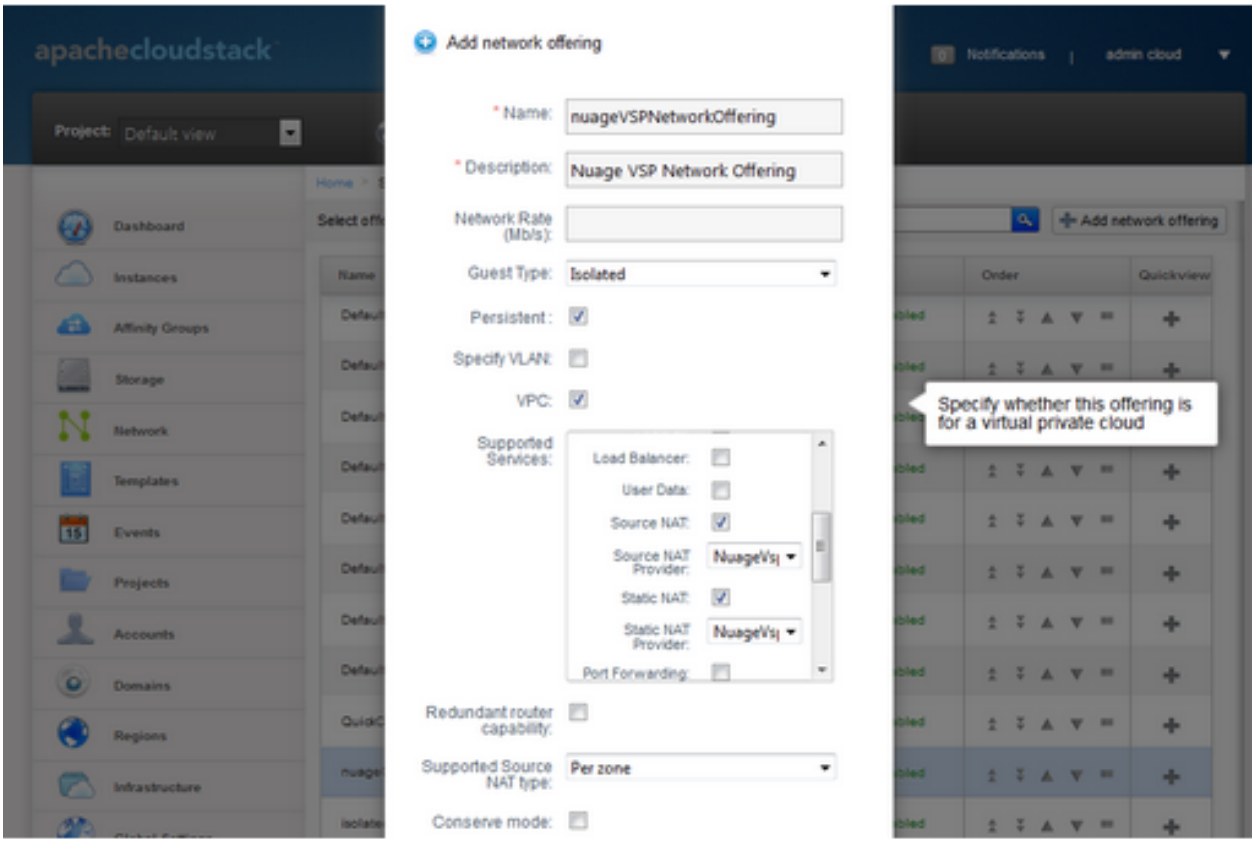


Fig. 12: Nuage VSP Source NAT-enabled Network Offering

Nuage VSP Static NAT via the Underlay Feature Support For CloudStack

Supported CloudStack versions: >= 4.10

Static NAT is supported in Nuage VSP as FIP (Floating IP). Prior to Nuage VSP v3.2, FIP in Nuage VSP required a VXLAN GW/PE to be present in the data center. In Nuage VSP v3.2 and above FIP is supported via the underlay, which removes the requirement for a GW/PE in the DC.

For the Static NAT without GW/PE feature to be operational in the CloudStack plugin, FIP in Nuage VSP must be configured to use the underlay. This operation takes place during Nuage VSP installation; instructions can be found in the Nuage VSP Install Guide.

A new API called `nuageunderlayvlaniprange` has been introduced to enable/disable Static NAT via the Underlay feature support for CloudStack public IP ranges being used for Static NAT service. This API specifies whether the FIP to underlay support is required for the corresponding FIP subnet in Nuage VSD since there is no GW/PE in the data center. When the `nuageunderlayvlaniprange` API has been enabled/disabled for a public IP range and Static NAT is enabled on at-least one of its Public IPs, the plugin creates the corresponding shared FIP subnet in Nuage VSD using the `sharednetworkresources` API with the underlay flag set accordingly. The `nuageunderlayvlaniprange` API usage is shown in the figure "nuageunderlayvlaniprange API Usage" below.

```
[root@csc-1 ~]# cloudmonkey
☁Apache CloudStack 🐵cloudmonkey 5.3.1. Type help or ? to list commands.

Using management server profile: local

(local) 🐵> list vlanipranges
count = 2
vlaniprange:
+--------------+--------------+--------------------------------------+--------------+---------+--------------------------------------+--------------------------------------+--------------+--------------+--------+--------------------------------------
|              networkid       |      endip   | account |            domainid                  |          physicalnetworkid          |    startip   |      vlan    | domain |                zoneid
|     netmask  |              id                     | forvirtualnetwork |   gateway    |
+--------------+--------------+--------------------------------------+--------------+---------+--------------------------------------+--------------------------------------+--------------+--------------+--------+--------------------------------------
| df10f916-936f-46b0-af7f-2e8b62238b82 | 10.100.100.200 |  system | 0f44b44b-95e5-11e6-9cec-faaca6091800 | 66aca5a6-0271-4fd8-9d7a-a877a3fdce37 | 10.100.100.115 | vlan://untagged |  ROOT  | 315bba1f-cf0e-4688-be8
1-fc079f4122b2 | 255.255.255.0 | 30241cb6-9d58-4ffe-9f34-216bbb9f2984 |        True       | 10.100.100.1 |
| df10f916-936f-46b0-af7f-2e8b62238b82 | 10.200.200.200 |  system | 0f44b44b-95e5-11e6-9cec-faaca6091800 | 66aca5a6-0271-4fd8-9d7a-a877a3fdce37 | 10.200.200.100 | vlan://untagged |  ROOT  | 315bba1f-cf0e-4688-be8
1-fc079f4122b2 | 255.255.255.0 | 870e479f-0688-4ae9-9641-c3fbe82ba7ee |        True       | 10.200.200.1 |
+--------------+--------------+--------------------------------------+--------------+---------+--------------------------------------+--------------------------------------+--------------+--------------+--------+--------------------------------------
(local) 🐵>
(local) 🐵>
(local) 🐵> enable nuageunderlayvlaniprange id=870e479f-0688-4ae9-9641-c3fbe82ba7ee

accountid = fed539a9-95e5-11e6-9cec-faaca6091800
cmd = com.cloud.api.commands.EnableNuageUnderlayVlanIpRangeCmd
created = 2016-10-19T06:34:15-0700
jobid = 37c66a87-5eb5-49d3-b5cd-3100dba650f2
jobprocstatus = 0
jobresult:
success = True
jobresultcode = 0
jobresulttype = object
jobstatus = 1
userid = fed54417-95e5-11e6-9cec-faaca6091800
(local) 🐵>
(local) 🐵>
(local) 🐵> list nuageunderlayvlanipranges id=870e479f-0688-4ae9-9641-c3fbe82ba7ee
count = 1
nuagevlaniprange:
+--------------+--------------+--------------------------------------+--------------+---------+--------------------------------------+--------------------------------------+--------------+--------------+--------+--------------------------------------
|              networkid       |      endip   | account |            domainid                  |          physicalnetworkid          |    startip   |      vlan    | domain |                zoneid
|     netmask  |              id                     | forvirtualnetwork | underlay |   gateway    |
+--------------+--------------+--------------------------------------+--------------+---------+--------------------------------------+--------------------------------------+--------------+--------------+--------+--------------------------------------
| df10f916-936f-46b0-af7f-2e8b62238b82 | 10.200.200.200 |  system | 0f44b44b-95e5-11e6-9cec-faaca6091800 | 66aca5a6-0271-4fd8-9d7a-a877a3fdce37 | 10.200.200.100 | vlan://untagged |  ROOT  | 315bba1f-cf0e-4688-be8
1-fc079f4122b2 | 255.255.255.0 | 870e479f-0688-4ae9-9641-c3fbe82ba7ee |        True       |   True   | 10.200.200.1 |
+--------------+--------------+--------------------------------------+--------------+---------+--------------------------------------+--------------------------------------+--------------+--------------+--------+--------------------------------------
(local) 🐵>
```

Fig. 13: nuageunderlayvlaniprange API Usage

By default, the Nuage VSP Plugin creates the corresponding shared FIP subnet in Nuage VSD with the underlay flag set to false (disabled). There is no support for the `nuageunderlayvlaniprange` API from the CloudStack UI.

Note: Enabling/disabling the `nuageunderlayvlaniprange` API for CloudStack public IP ranges is supported only before the Nuage VSP plugin creates the corresponding shared FIP subnet in Nuage VSD. After a shared FIP subnet is created in Nuage VSD, its underlay flag cannot be changed. To change the underlay flag for a given shared FIP subnet, delete the Public vLanIPRange, recreate it and enable/disable the `nuageunderlayvlaniprange` API for it.

7.3.5 Running The Nuage VSP Plugin Specific Marvin Tests

The Nuage VSP Plugin specific Marvin tests can be found under the directory test/integration/plugins/nuagevsp/ in the cloudstack tree.

Here is the list of required Python packages and dependencies to run The Nuage VSP Plugin specific Marvin tests:

- marvin
- vspk
- libVSD
- pyyaml
- netaddr
- futures

Note: vspk is a Python SDK for Nuage VSP's VSD and libVSD is a library that wraps vspk package, which are open sourced and can be found at https://github.com/nuagenetworks.

Here is an example nosetests command to run The Nuage VSP Plugin specific Marvin tests:

```
nosetests --with-marvin --marvin-config=path-to-marvin-config-file/nuage_marvin.cfg␣
↪path-to-marvin-tests/test/integration/plugins/nuagevsp/test_nuage_vsp.py
```

Note: For an example Marvin config file (i.e. nuage_marvin.cfg) required to run The Nuage VSP Plugin specific Marvin tests, refer *Nuage VSP Marvin Config File Format* in the Appendix of this document.

7.3.6 Appendix

Configure Nuage VSP API

To configure Nuage VSP as a Network Service Provider in the CloudStack Zone.

1. Add Nuage VSP as a Network Service Provider in the Physical Network 2:

```
cloudmonkey add networkserviceprovider name=NuageVsp physicalnetworkid=
↪<physicalNetwork2Id>
```

2. Add the Nuage VSD as a Nuage VSP Device in the Physical Network 2:

```
cloudmonkey add nuagevspdevice physicalnetworkid=<physicalNetwork2Id> hostname=
↪<hostnameOfNuageVsp> username=<usernameOfNuageVspUser> password=
↪<passwordOfNuageVspUser> port=<portUsedByNuageVsp> apiversion=<apiVersionOfNuageVsp>
↪ retrycount=<nrOfRetriesOnFailure> retryinterval=<intervalBetweenRetries>
```

Nuage VSP Marvin Config File Format

Format for the Marvin config file required to run The Nuage VSP Plugin specific Marvin tests.

```
{
 "zones": [
     {
         "name": "ZONE1NAME",
         "physical_networks": [
             {
                 "name": "Physical Network 1",
                 "isolationmethods": [
```

(continues on next page)

(continued from previous page)

```
                    "VLAN"
                ]
            },
            {
                "name": "Physical Network 2",
                "isolationmethods": [
                    "VSP"
                ],
                "providers": [
                    {
                        "name": "NuageVsp",
                        "devices": [
                            {
                                "username": "VSDUSERNAME",
                                "retryinterval": "60",
                                "hostname": "VSDSERVER",
                                "apiversion": "VSDVERSION",
                                "retrycount": "4",
                                "password": "VSDUSERPASSWORD",
                                "port": VSDPORT
                            }
                        ]
                    }
                ]
            }
        ],
        "dcInternetConnectivityInfo" : {
            "available": "INTERNETAVAILABLE",
            "httpProxy": "HTTPPROXY",
            "httpsProxy": "HTTPSPROXY"
        }
    },
    {
        "name": "ZONE2NAME",
        "physical_networks": [
            {
                "name": "Physical Network 1",
                "isolationmethods": [
                    "VLAN"
                ]
            },
            {
                "name": "Physical Network 2",
                "isolationmethods": [
                    "VSP"
                ],
                "providers": [
                    {
                        "name": "NuageVsp",
                        "devices": [
                            {
                                "username": "VSDUSERNAME",
                                "retryinterval": "60",
                                "hostname": "VSDSERVER",
                                "apiversion": "VSDVERSION",
                                "retrycount": "4",
                                "password": "VSDUSERPASSWORD",
```

(continues on next page)

(continued from previous page)

```
                                "port": VSDPORT
                            }
                        ]
                    }
                ]
            }
        ],
        "dcInternetConnectivityInfo" : {
            "available": "INTERNETAVAILABLE",
            "httpProxy": "HTTPPROXY",
            "httpsProxy": "HTTPSPROXY"
        }
    }
],
"dbSvr": {
    "dbSvr": "DBSERVER",
    "passwd": "DBPASSWORD",
    "db": "cloud",
    "port": 3306,
    "user": "DBUSERNAME"
},
"logger":
    {
        "LogFolderPath": "/tmp/LOGFOLDERNAME"
    },
"mgtSvr": [
    {
        "mgtSvrIp": "MGNTSERVERIP",
        "port": 8096,
        "user": "MGNTUSERNAME",
        "passwd": "MGNTPASSWORD"
    }
]
}
```

7.4 The VXLAN Plugin

7.4.1 System Requirements for VXLAN

In PRODUCT 4.X.0, this plugin only supports the KVM hypervisor with the standard linux bridge.

The following table lists the requirements for the hypervisor.

Item	Requirement	Note
Hypervisor	KVM	OvsVifDriver is not supported by this plugin in PRODUCT 4.X, use BridgeVifDriver (default).
Linux kernel	version >= 3.7, VXLAN kernel module enabled	It is recommended to use kernel >=3.9, since Linux kernel categorizes the VXLAN driver as experimental <3.9.
iproute2	matches kernel version	

Table: Hypervisor Requirement for VXLAN

7.4.2 Linux Distributions that meet the requirements

The following table lists distributions which meet requirements.

Distribution	Release Version	Kernel Version (Date confirmed)	Note
Ubuntu	13.04	3.8.0 (2013/07/23)	
Fedora	>= 17	3.9.10 (2013/07/23)	Latest kernel packages are available in "update" repository.
CentOS	>= 6.5	2.6.32-431.3.1.el6.x86_64 (2014/01/21)	

Table: List of Linux distributions which meet the hypervisor requirements

Check the capability of your system

To check the capability of your system, execute the following commands.

```
$ sudo modprobe vxlan && echo $?
# Confirm the output is "0".
# If it's non-0 value or error message, your kernel doesn't have VXLAN kernel module.

$ ip link add type vxlan help
# Confirm the output is usage of the command and that it's for VXLAN.
# If it's not, your iproute2 utility doesn't support VXLAN.
```

Important note on MTU size

When new vxlan interfaces are created, kernel will obtain current MTU size of the physical interface (ethX or the bridge) and then create vxlan interface/bridge that are exactly 50 bytes smaller than the MTU on physical interface/bridge. This means that in order to support default MTU size of 1500 bytes inside VM, your vxlan interface/bridge must also have MTU of 1500 bytes, meaning that your physical interface/bridge must have MTU of at least 1550 bytes. In order to configure "jumbo frames" you can i.e. make physical interface/bridge with 9000 bytes MTU, then all the vxlan interfaces will be created with MTU of 8950 bytes, and then MTU size inside VM can be set to 8950 bytes.

Important note on max number of multicast groups (and thus VXLAN interfaces)

Default value of "net.ipv4.igmp_max_memberships" (cat /proc/sys/net/ipv4/igmp_max_memberships) is "20", which means that host can be joined to max 20 multicast groups (attach max 20 multicast IPs on the host). Since all VXLAN (VTEP) interfaces provisioned on host are multicast-based (belong to certain multicast group, and thus has it's own multicast IP that is used as VTEP), this means that you can not provision more than 20 (working) VXLAN interfaces per host. On Linux kernel 3.x you actually can provision more than 20, but ARP request will silently fail and cause client's networking problems On Linux kernel 4.x you can NOT provision (start) more than 20 VXLAN interfaces and error message "No buffer space available" can be observed in Cloudstack Agent logs after provisioning required bridges and VXLAN interfaces. Increase needed parameter to sane value (i.e. 100 or 200) as required. If you need to operate more than 20 VMs from different client's network, this change above is required.

Advanced: Build kernel and iproute2

Even if your system doesn't support VXLAN, you can compile the kernel and iproute2 by yourself. The following procedure is an example for CentOS 6.4.

Build kernel

```
$ sudo yum groupinstall "Development Tools"
$ sudo yum install ncurses-devel hmaccalc zlib-devel binutils-devel elfutils-libelf-
↪devel bc

$ KERNEL_VERSION=3.10.4
# Declare the kernel version you want to build.

$ wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-${KERNEL_VERSION}.tar.xz
$ tar xvf linux-${KERNEL_VERSION}.tar.xz
$ cd linux-${KERNEL_VERSION}
$ cp /boot/config-`uname -r` .config
$ make oldconfig
# You may keep hitting enter and choose the default.

$ make menuconfig
# Dig into "Device Drivers" -> "Network device support",
# then select "Virtual eXtensible Local Area Network (VXLAN)" and hit space.
# Make sure it indicates "<M>" (build as module), then Save and Exit.

# You may also want to check "IPv4 NAT" and its child nodes in "IP: Netfilter␣
↪Configuration"
# and "IPv6 NAT" and its child nodes in "IPv6: Netfilter Configuration".
# In 3.10.4, you can find the options in
# "Networking support" -> "Networking options"
#   -> "Network packet filtering framework (Netfilter)".

$ make # -j N
# You may use -j N option to make the build process parallel and faster,
# generally N = 1 + (cores your machine have).

$ sudo make modules_install
$ sudo make install
# You would get an error like "ERROR: modinfo: could not find module XXXX" here.
# This happens mainly due to config structure changes between kernel versions.
# You can ignore this error, until you find you need the kernel module.
# If you feel uneasy, you can go back to make menuconfig,
# find module XXXX by using '/' key, enable the module, build and install the kernel␣
↪again.

$ sudo vi /etc/grub.conf
# Make sure the new kernel isn't set as the default and the timeout is long enough,
# so you can select the new kernel during boot process.
# It's not a good idea to set the new kernel as the default until you confirm the␣
↪kernel works fine.

$ sudo reboot
# Select the new kernel during the boot process.
```

Build iproute2

```
$ sudo yum install db4-devel

$ git clone git://git.kernel.org/pub/scm/linux/kernel/git/shemminger/iproute2.git
```

(continues on next page)

(continued from previous page)

```
$ cd iproute2
$ git tag
# Find the version that matches the kernel.
# If you built kernel 3.10.4 as above, it would be v3.10.0.

$ git checkout v3.10.0
$ ./configure
$ make # -j N
$ sudo make install
```

Note: Please use rebuild kernel and tools at your own risk.

7.4.3 Configure PRODUCT to use VXLAN Plugin

Configure hypervisor

Configure hypervisor: KVM

In addition to "KVM Hypervisor Host Installation" in "PRODUCT Installation Guide", you have to configure the following item on the host.

Create bridge interface with IPv4 address

This plugin requires an IPv4 address on the KVM host to terminate and originate VXLAN traffic. The address should be assinged to a physical interface or a bridge interface bound to a physical interface. Both a private address or a public address are fine for the purpose. It is not required to be in the same subnet for all hypervisors in a zone, but they should be able to reach each other via IP multicast with UDP/8472 port. A name of a physical interface or a name of a bridge interface bound to a physical interface can be used as a traffic label. Physical interface name fits for almost all cases, but if physical interface name differs per host, you may use a bridge to set a same name. If you would like to use a bridge name as a traffic label, you may create a bridge in this way.

Let `cloudbr1` be the bridge interface for the instances' private network.

Configure in RHEL or CentOS

When you configured the `cloudbr1` interface as below,

```
$ sudo vi /etc/sysconfig/network-scripts/ifcfg-cloudbr1
```

```
DEVICE=cloudbr1
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

you would change the configuration similar to below.

```
DEVICE=cloudbr1
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.0.2.X
NETMASK=255.255.255.0
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

Configure in Ubuntu

When you configured `cloudbr1` as below,

```
$ sudo vi /etc/network/interfaces
```

```
auto lo
iface lo inet loopback

# The primary network interface
auto eth0.100
iface eth0.100 inet static
    address 192.168.42.11
    netmask 255.255.255.240
    gateway 192.168.42.1
    dns-nameservers 8.8.8.8 8.8.4.4
    dns-domain lab.example.org

# Public network
auto cloudbr0
iface cloudbr0 inet manual
    bridge_ports eth0.200
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1

# Private network
auto cloudbr1
iface cloudbr1 inet manual
    bridge_ports eth0.300
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
```

you would change the configuration similar to below.

```
auto lo
iface lo inet loopback

# The primary network interface
auto eth0.100
iface eth0.100 inet static
    address 192.168.42.11
    netmask 255.255.255.240
```

(continues on next page)

(continued from previous page)

```
    gateway 192.168.42.1
    dns-nameservers 8.8.8.8 8.8.4.4
    dns-domain lab.example.org

# Public network
auto cloudbr0
iface cloudbr0 inet manual
    bridge_ports eth0.200
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1

# Private network
auto cloudbr1
iface cloudbr1 inet static
    addres 192.0.2.X
    netmask 255.255.255.0
    bridge_ports eth0.300
    bridge_fd 5
    bridge_stp off
    bridge_maxwait 1
```

Configure iptables to pass XVLAN packets

Since VXLAN uses UDP packet to forward encapsulated the L2 frames, UDP/8472 port must be opened.

Configure in RHEL or CentOS

RHEL and CentOS use iptables for firewalling the system, you can open extra ports by executing the following iptable commands:

```
$ sudo iptables -I INPUT -p udp -m udp --dport 8472 -j ACCEPT
```

These iptable settings are not persistent accross reboots, we have to save them first.

```
$ sudo iptables-save > /etc/sysconfig/iptables
```

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

```
$ sudo service network restart
$ sudo reboot
```

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Configure in Ubuntu

The default firewall under Ubuntu is UFW (Uncomplicated FireWall), which is a Python wrapper around iptables.

To open the required ports, execute the following commands:

```
$ sudo ufw allow proto udp from any to any port 8472
```

Note: By default UFW is not enabled on Ubuntu. Executing these commands with the firewall disabled does not enable the firewall.

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

```
$ sudo service networking restart
$ sudo reboot
```

Warning: Make sure you have an alternative way like IPMI or ILO to reach the machine in case you made a configuration error and the network stops functioning!

Setup zone using VXLAN

In almost all parts of zone setup, you can just follow the advanced zone setup istruction in "PRODUCT Installation Guide" to use this plugin. It is not required to add a network element nor to reconfigure the network offering. The only thing you have to do is configure the physical network to use VXLAN as the isolation method for Guest Network.

Configure the physical network

CloudStack needs to have one physical network for Guest Traffic with the isolation method set to "VXLAN".

Guest Network traffic label should be the name of the physical interface or the name of the bridge interface and the bridge interface and they should have an IPv4 address. See ? for details.

Configure the guest traffic

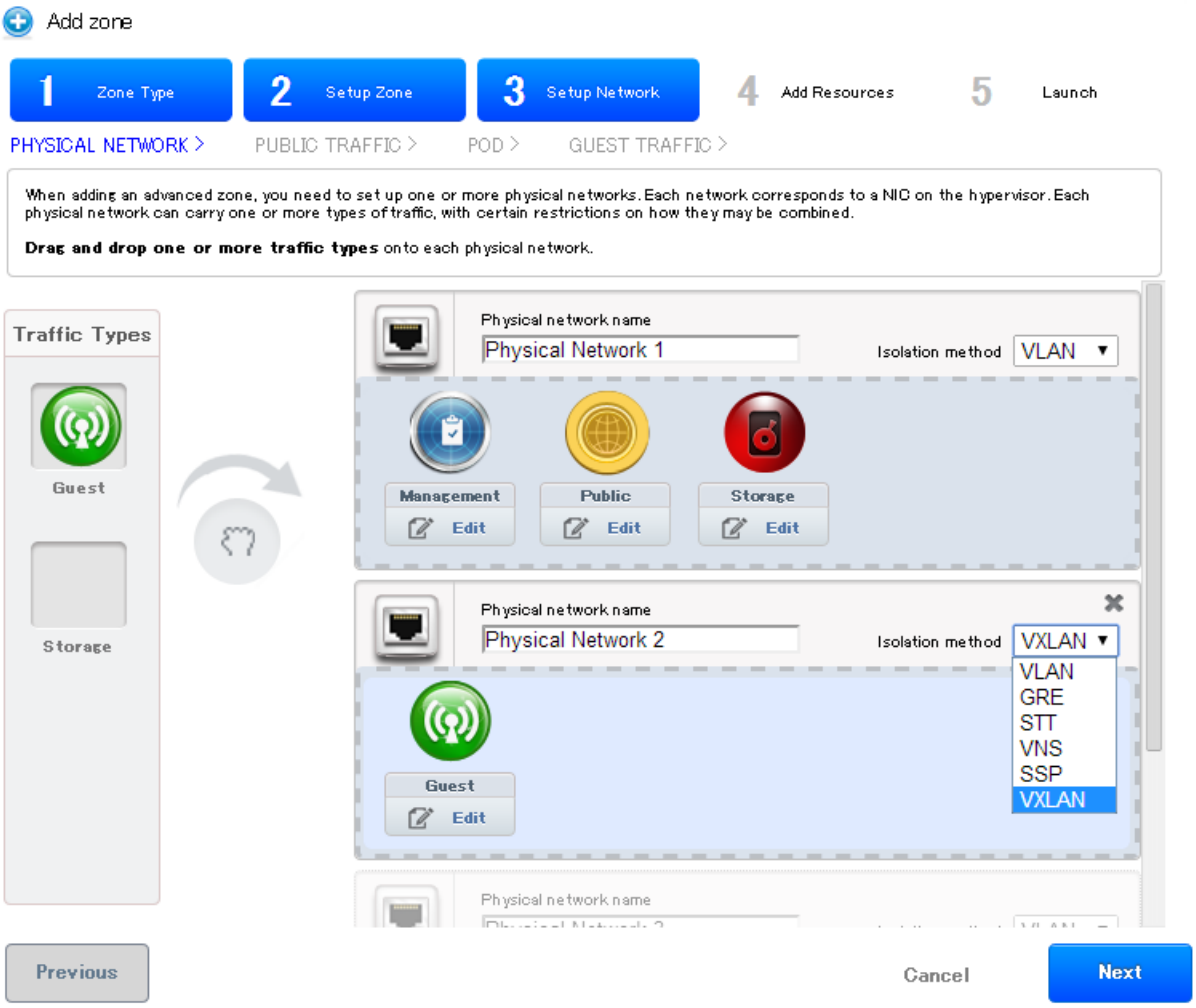
Specify a range of VNIs you would like to use for carrying guest network traffic.

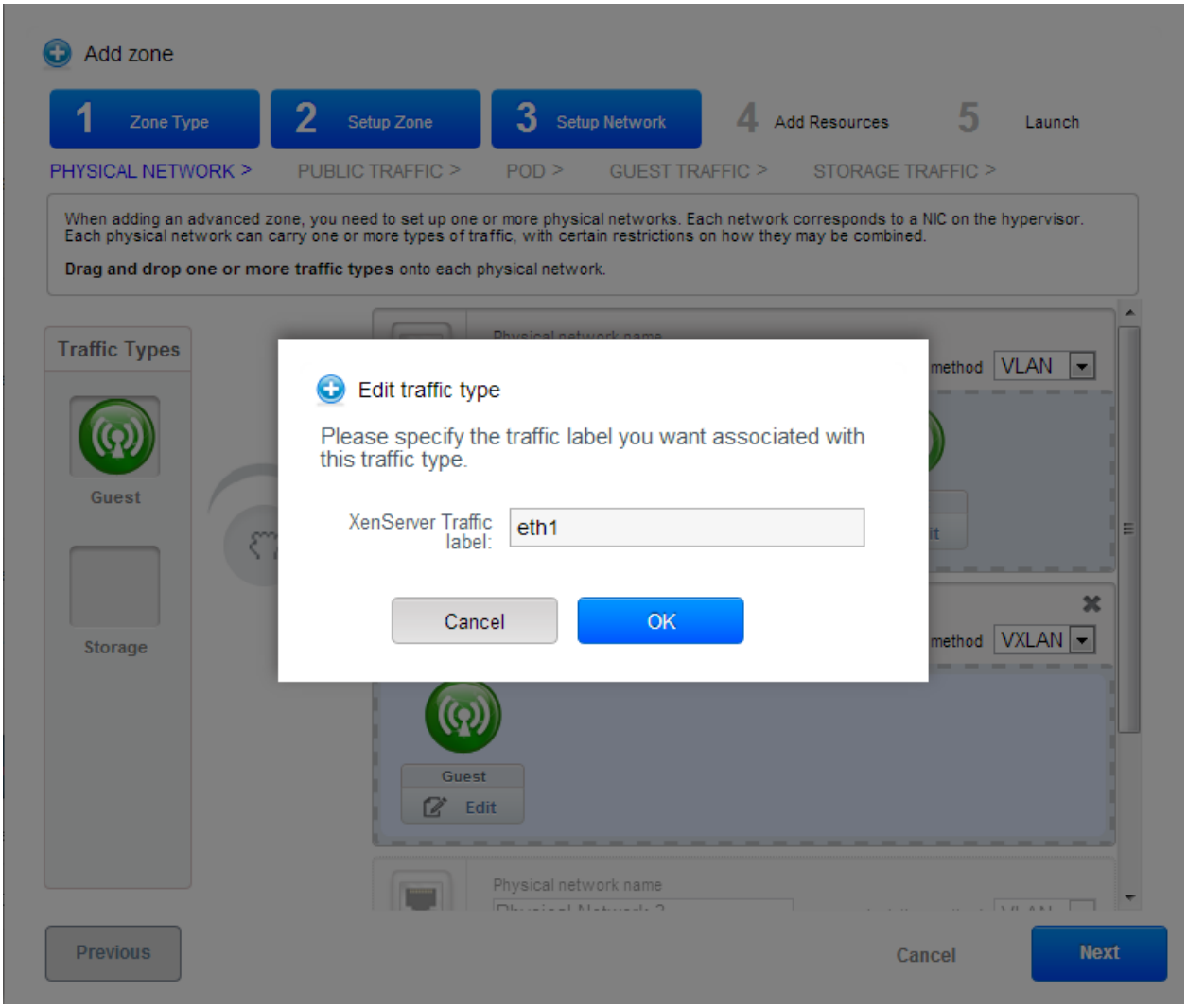
Warning: VNI must be unique per zone and no duplicate VNIs can exist in the zone. Exercise care when designing your VNI allocation policy.

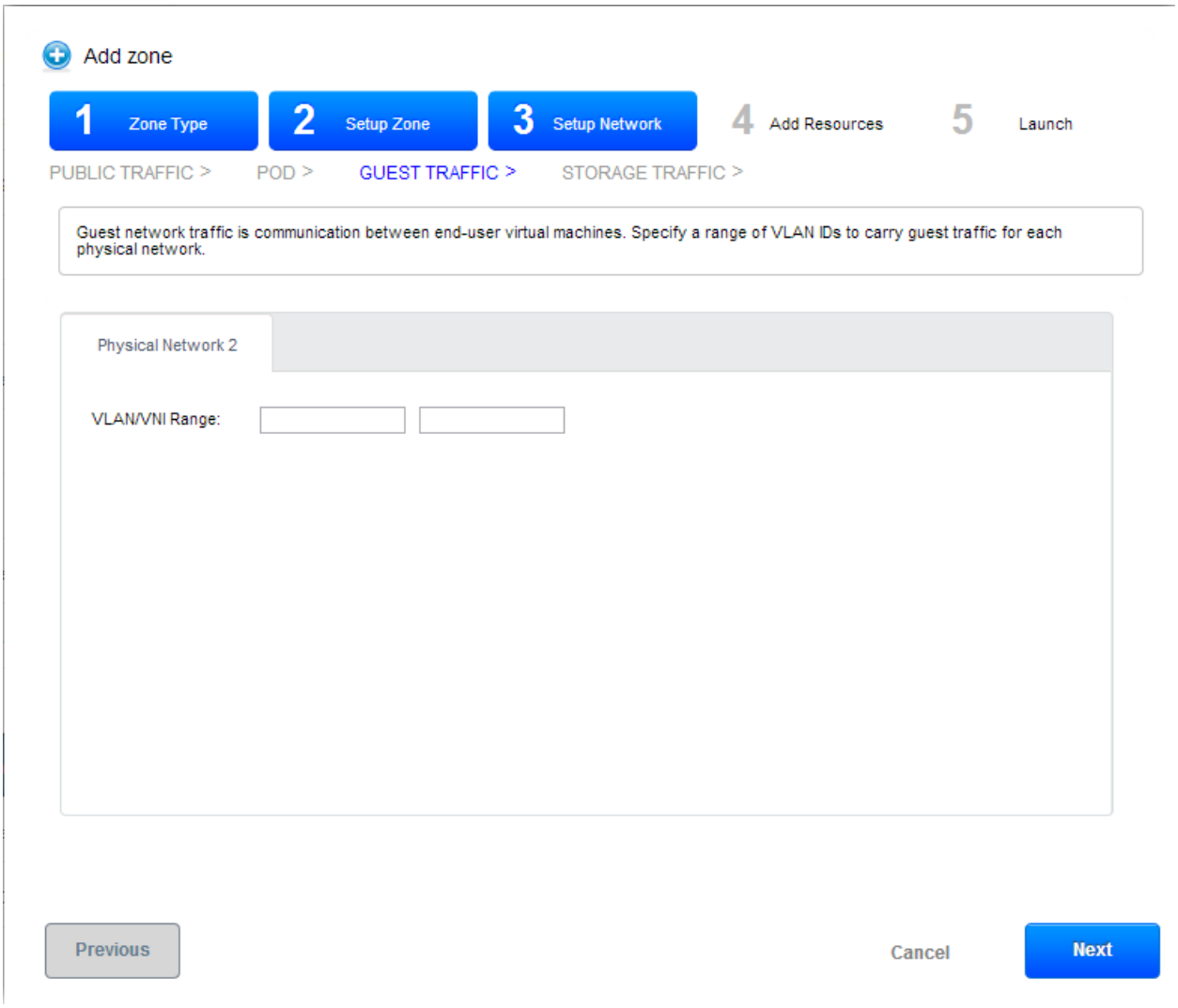
7.5 The OVS Plugin

7.5.1 Introduction to the OVS Plugin

The OVS plugin is the native SDN implementations in CloudStack, using GRE isolation method. The plugin can be used by CloudStack to implement isolated guest networks and to provide additional services like NAT, port forwarding and load balancing.


Add zone
1 Zone Type
2 Setup Zone
3 Setup Network
4 Add Resources
5 Launch
PHYSICAL NETWORK >
PUBLIC TRAFFIC >
POD >
GUEST TRAFFIC >
When adding an advanced zone, you need to set up one or more physical networks. Each network corresponds to a NIC on the hypervisor. Each physical network can carry one or more types of traffic, with certain restrictions on how they may be combined.
Drag and drop one or more traffic types onto each physical network.
Traffic Types
Guest
Storage
Physical network name
Physical Network 1
Isolation method
VLAN
Management
Edit
Public
Edit
Storage
Edit
Physical network name
Physical Network 2
Isolation method
VXLAN
VLAN
GRE
STT
VNS
SSP
VXLAN
Guest
Edit
Physical network name
Previous
Cancel
Next


Add zone
1 Zone Type
2 Setup Zone
3 Setup Network
4 Add Resources
5 Launch
PHYSICAL NETWORK >
PUBLIC TRAFFIC >
POD >
GUEST TRAFFIC >
STORAGE TRAFFIC >
When adding an advanced zone, you need to set up one or more physical networks. Each network corresponds to a NIC on the hypervisor.
Each physical network can carry one or more types of traffic, with certain restrictions on how they may be combined.
Drag and drop one or more traffic types onto each physical network.
Traffic Types
Guest
Storage
Edit traffic type
Please specify the traffic label you want associated with this traffic type.
XenServer Traffic label:
eth1
Cancel
OK
method
VLAN
VXLAN
Guest
Edit
Physical network name
Previous
Cancel
Next


Add zone
1 Zone Type
2 Setup Zone
3 Setup Network
4 Add Resources
5 Launch
PUBLIC TRAFFIC >
POD >
GUEST TRAFFIC >
STORAGE TRAFFIC >
Guest network traffic is communication between end-user virtual machines. Specify a range of VLAN IDs to carry guest traffic for each physical network.
Physical Network 2
VLAN/VNI Range:
Previous
Cancel
Next

Features of the OVS Plugin

The following table lists the CloudStack network services provided by the OVS Plugin.

Network Service	CloudStack version
Virtual Networking	>= 4.0
Static NAT	>= 4.3
Port Forwarding	>= 4.3
Load Balancing	>= 4.3

Table: Supported Services

Note: The Virtual Networking service was originally called 'Connectivity' in CloudStack 4.0

The following hypervisors are supported by the OVS Plugin.

Hypervisor	CloudStack version
XenServer	>= 4.0
KVM	>= 4.3

Table: Supported Hypervisors

7.5.2 Configuring the OVS Plugin

Prerequisites

Before enabling the OVS plugin the hypervisor needs to be install OpenvSwitch. Default, XenServer has already installed OpenvSwitch. However, you must install OpenvSwitch manually on KVM. CentOS 6.4 and OpenvSwitch 1.10 are recommended.

KVM hypervisor:

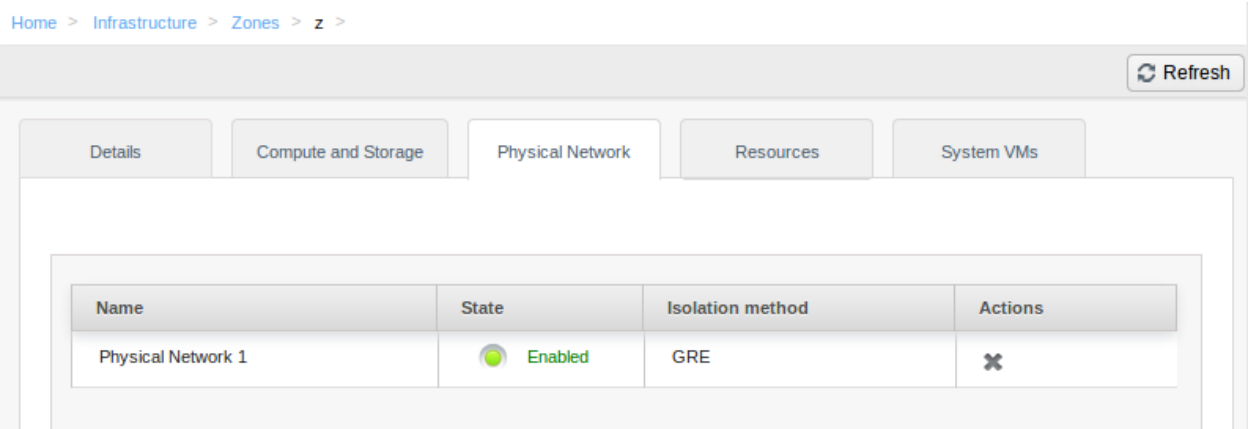
- CentOS 6.4 is recommended.
- To make sure that the native bridge module will not interfere with openvSwitch the bridge module should be added to the blacklist. See the modprobe documentation for your distribution on where to find the blacklist. Make sure the module is not loaded either by rebooting or executing rmmod bridge before executing next steps.

Zone Configuration

CloudStack needs to have at least one physical network with the isolation method set to "GRE". This network should be enabled for the Guest traffic type.

Note: With KVM, the traffic type should be configured with the traffic label that matches the name of the Integration Bridge on the hypervisor. For example, you should set the traffic label as following:

- Management & Storage traffic: cloudbr0
- Guest & Public traffic: cloudbr1 See KVM networking configuration guide for more detail.



Agent Configuration

Note: Only for KVM hypervisor

- Configure network interfaces:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
IPV6INIT=no
NM_CONTROLLED=no
ONBOOT=yes
TYPE=OVSPort
DEVICETYPE=ovs
OVS_BRIDGE=cloudbr0

/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
BOOTPROTO=none
IPV6INIT=no
NM_CONTROLLED=no
ONBOOT=yes
TYPE=OVSPort
DEVICETYPE=ovs
OVS_BRIDGE=cloudbr1

/etc/sysconfig/network-scripts/ifcfg-cloudbr0
DEVICE=cloudbr0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSBridge
BOOTPROTO=static
IPADDR=172.16.10.10
GATEWAY=172.16.10.1
NETMASK=255.255.255.0
HOTPLUG=no

/etc/sysconfig/network-scripts/ifcfg-cloudbr1
DEVICE=cloudbr1
ONBOOT=yes
```

(continues on next page)

(continued from previous page)

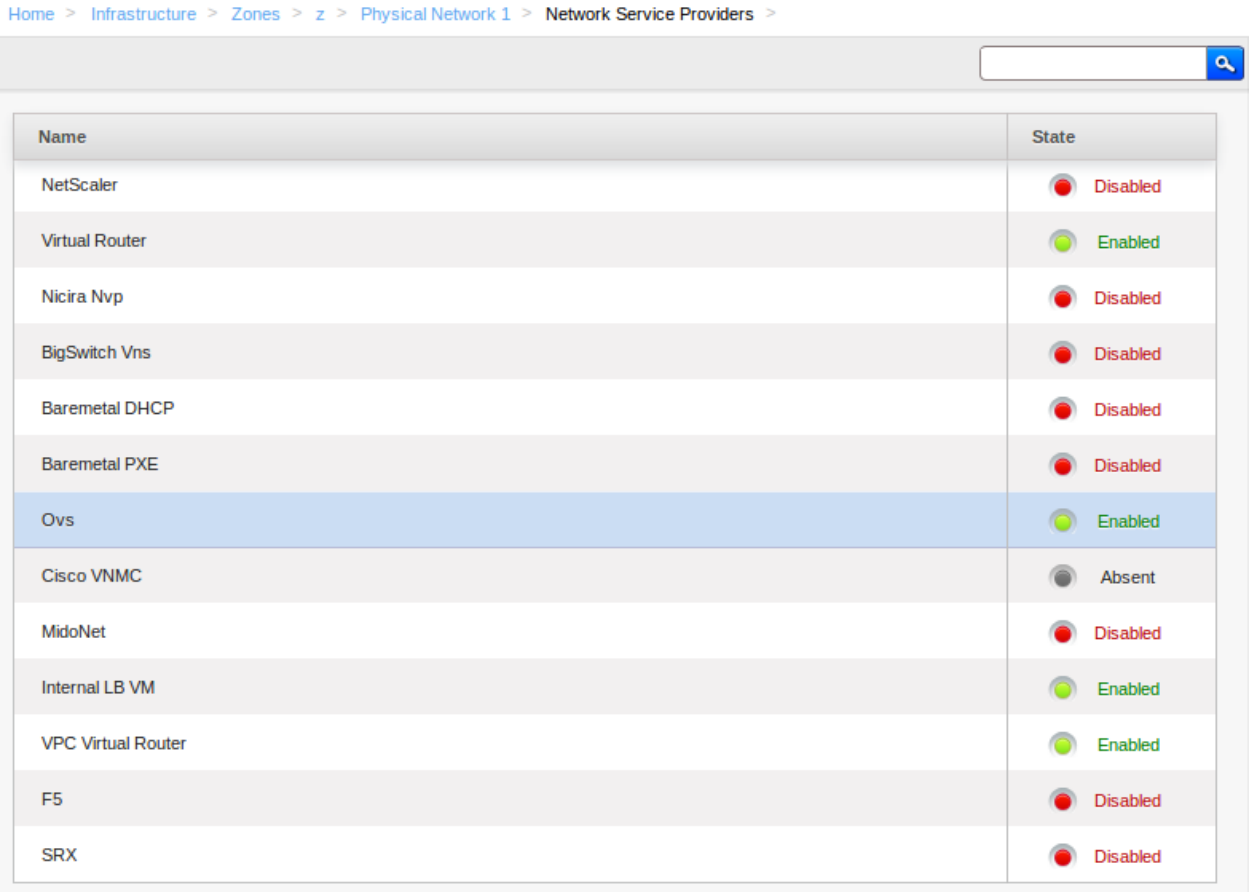
```
DEVICETYPE=ovs
TYPE=OVSBridge
BOOTPROTO=none
HOTPLUG=no

/etc/sysconfig/network
NETWORKING=yes
HOSTNAME=testkvm1
GATEWAY=172.10.10.1
```

- Edit /etc/cloudstack/agent/agent.properties

```
network.bridge.type=openvswitch
libvirt.vif.driver=com.cloud.hypervisor.kvm.resource.OvsVifDriver
```

Enabling the service provider

The OVS provider is disabled by default. Navigate to the "Network Service Providers" configuration of the physical network with the GRE isolation type. Navigate to the OVS provider and press the "Enable Provider" button.



Network Offerings

Using the OVS plugin requires a network offering with Virtual Networking enabled and configured to use the OVS element. Typical use cases combine services from the Virtual Router appliance and the OVS plugin.

Service	Provider
VPN	VirtualRouter
DHCP	VirtualRouter
DNS	VirtualRouter
Firewall	VirtualRouter
Load Balancer	OVS
User Data	VirtualRouter
Source NAT	VirtualRouter
Static NAT	OVS
Post Forwarding	OVS
Virtual Networking	OVS

Table: Isolated network offering with regular services from the Virtual Router.

Note: The tag in the network offering should be set to the name of the physical network with the OVS provider.

Isolated network with network services. The virtual router is still required to provide network services like dns and dhcp.

Service	Provider
DHCP	VirtualRouter
DNS	VirtualRouter
User Data	VirtualRouter
Source NAT	VirtualRouter
Static NAT	OVS
Post Forwarding	OVS
Load Balancing	OVS
Virtual Networking	OVS

Table: Isolated network offering with network services

7.5.3 Using the OVS plugin with VPC

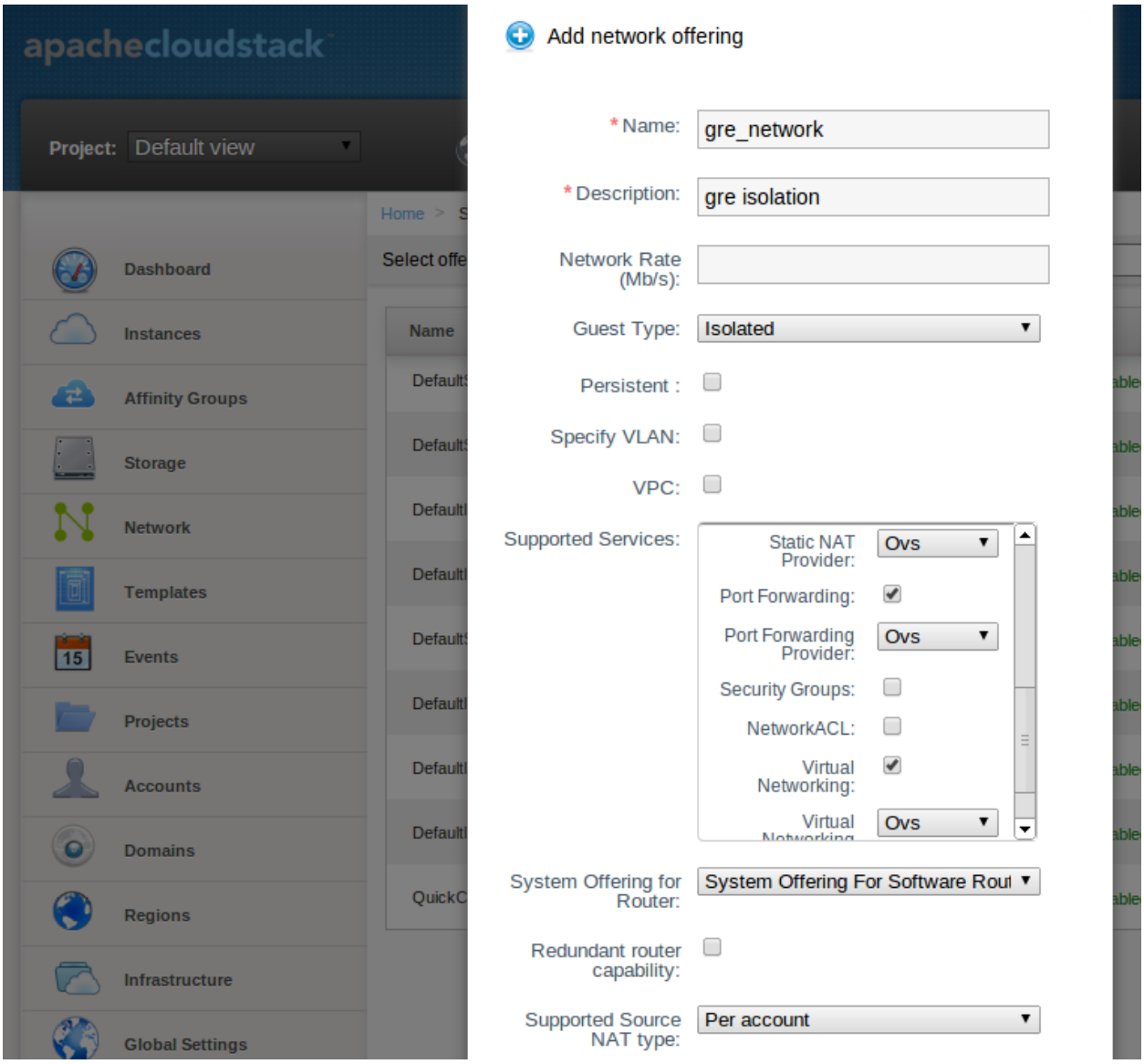
OVS plugin does not work with VPC at that time

7.5.4 Revision History

0-0 Mon Dec 2 2013 Nguyen Anh Tu tuna@apache.org Documentation created for 4.3.0 version of the OVS Plugin

7.6 IPv6 Support in CloudStack

CloudStack supports Internet Protocol version 6 (IPv6), the recent version of the Internet Protocol (IP) that defines routing the network traffic. IPv6 uses a 128-bit address that exponentially expands the current address space that


apachecloudstack
Project: Default view
Dashboard
Instances
Affinity Groups
Storage
Network
Templates
Events
Projects
Accounts
Domains
Regions
Infrastructure
Global Settings
Home
Select offe
Name
Add network offering
* Name: gre_network
* Description: gre isolation
Network Rate (Mb/s):
Guest Type: Isolated
Persistent :
Specify VLAN:
VPC:
Supported Services:
Static NAT Provider: Ovs
Port Forwarding:
Port Forwarding Provider: Ovs
Security Groups:
NetworkACL:
Virtual Networking:
Virtual Networking Ovs
System Offering for Router: System Offering For Software Rout
Redundant router capability:
Supported Source NAT type: Per account

is available to the users. IPv6 addresses consist of eight groups of four hexadecimal digits separated by colons, for example, 5001:0dt8:83a3:1012:1000:8s2e:0870:7454. CloudStack supports IPv6 for public IPs in shared networks. With IPv6 support, VMs in shared networks can obtain both IPv4 and IPv6 addresses from the DHCP server. You can deploy VMs either in a IPv6 or IPv4 network, or in a dual network environment. If IPv6 network is used, the VM generates a link-local IPv6 address by itself, and receives a stateful IPv6 address from the DHCPv6 server.

IPv6 is supported only on KVM and XenServer hypervisors. The IPv6 support is only an experimental feature.

Here's the sequence of events when IPv6 is used:

1. The administrator creates an IPv6 shared network in an advanced zone.
2. The user deploys a VM in an IPv6 shared network.
3. The user VM generates an IPv6 link local address by itself, and gets an IPv6 global or site local address through DHCPv6.

7.6.1 Prerequisites and Guidelines

Consider the following:

- CIDR size must be 64 for IPv6 networks.
- The DHCP client of the guest VMs should support generating DUID based on Link-layer Address (DUID- LL). DUID-LL derives from the MAC address of guest VMs, and therefore the user VM can be identified by using DUID. See Dynamic Host Configuration Protocol for IPv6for more information.
- The gateway of the guest network generates Router Advisement and Response messages to Router Solicitation. The M (Managed Address Configuration) flag of Router Advisement should enable stateful IP address configuration. Set the M flag to where the end nodes receive their IPv6 addresses from the DHCPv6 server as opposed to the router or switch.

  Note: The M flag is the 1-bit Managed Address Configuration flag for Router Advisement. When set, Dynamic Host Configuration Protocol (DHCPv6) is available for address configuration in addition to any IPs set by using stateless address auto-configuration.

- Use the System VM template exclusively designed to support IPv6. Download the System VM template from http://download.cloudstack.org/systemvm/.
- The concept of Default Network applies to IPv6 networks. However, unlike IPv4 CloudStack does not control the routing information of IPv6 in shared network; the choice of Default Network will not affect the routing in the user VM.
- In a multiple shared network, the default route is set by the rack router, rather than the DHCP server, which is out of CloudStack control. Therefore, in order for the user VM to get only the default route from the default NIC, modify the configuration of the user VM, and set non-default NIC's `accept_ra` to 0 explicitly. The `accept_ra` parameter accepts Router Advertisements and auto-configure `/proc/sys/net/ipv6/conf/interface` with received data.

7.6.2 Limitations of IPv6 in CloudStack

The following are not yet supported:

1. Security groups
2. Userdata and metadata
3. Passwords

7.6.3 Guest VM Configuration for DHCPv6

For the guest VMs to get IPv6 address, run dhclient command manually on each of the VMs. Use DUID-LL to set up dhclient.

Note: The IPv6 address is lost when a VM is stopped and started. Therefore, use the same procedure to get an IPv6 address when a VM is stopped and started.

1. Set up dhclient by using DUID-LL.

   Perform the following for DHCP Client 4.2 and above:

   (a) Run the following command on the selected VM to get the dhcpv6 offer from VR:

   ```
   dhclient -6 -D LL <dev>
   ```

   Perform the following for DHCP Client 4.1:

   (a) Open the following to the dhclient configuration file:

   ```
   vi /etc/dhcp/dhclient.conf
   ```

   (b) Add the following to the dhclient configuration file:

   ```
   send dhcp6.client-id = concat(00:03:00, hardware);
   ```

2. Get IPv6 address from DHCP server as part of the system or network restart.

   Based on the operating systems, perform the following:

   On CentOS 6.2:

   (a) Open the Ethernet interface configuration file:

   ```
   vi /etc/sysconfig/network-scripts/ifcfg-eth0
   ```

   The `ifcfg-eth0` file controls the first NIC in a system.

   (b) Make the necessary configuration changes, as given below:

   ```
   DEVICE=eth0
   HWADDR=06:A0:F0:00:00:38
   NM_CONTROLLED=no
   ONBOOT=yes
   BOOTPROTO=dhcp6
   TYPE=Ethernet
   USERCTL=no
   PEERDNS=yes
   IPV6INIT=yes
   DHCPV6C=yes
   ```

   (c) Open the following:

   ```
   vi /etc/sysconfig/network
   ```

   (d) Make the necessary configuration changes, as given below:

```
NETWORKING=yes
HOSTNAME=centos62mgmt.lab.vmops.com
NETWORKING_IPV6=yes
IPV6_AUTOCONF=no
```

On Ubuntu 12.10

(a) Open the following:

```
etc/network/interfaces:
```

(b) Make the necessary configuration changes, as given below:

```
iface eth0 inet6 dhcp
autoconf 0
accept_ra 1
```

7.7 Quota Plugin

Quota service, while allowing for scalability, will make sure that the cloud is not exploited by attacks, careless use and program errors. To address this problem, employ the quota-enforcement service that allows resource usage within certain bounds as defined by policies and available quotas for various entities. Quota service extends the functionality of usage server to provide a measurement for the resources used by the accounts and domains using a common unit referred to as cloud currency in this document. It can be configured to ensure that your usage won't exceed the budget allocated to accounts/domain in cloud currency. It will let user know how much of the cloud resources he is using. It will help the cloud admins, if they want, to ensure that a user does not go beyond his allocated quota. Per usage cycle if an account is found to be exceeding its quota then it is locked. Locking an account means that it will not be able to initiate a new resource allocation request, whether it is more storage or an additional ip. To unlock an account you need to add more credit to it. In case you want the locking to be disabled on global or on account scope those provisions are also provided. Needless to say quota service as well as any action on the account is configurable.

7.7.1 Enabling the Quota Service

Before installing and configuring the quota service you need to make sure that the Usage Server has been installed. This requires extra steps beyond just installing the CloudStack software. See Installing the Usage Server (Optional) in the Advanced Installation Guide.

1. enable.usage.server: Set to true to enable usage server.

The quota plugin is disabled by default. To enable it goto Global Settings and set the following global configuration to true:

1. quota.enable.service

By default Quota service does not lock the accounts that have exceeded the quota usage. To enable quota service to lock accounts set the following global configuration to true:

1. quota.enable.enforcement

The other configurations that are there for quota service are as:

1. quota.currency.symbol : The symbol that is used before any currency figure in various quota forms and reports.
2. quota.usage.smtp.host: Quota SMTP host for sending quota alerts.
3. quota.usage.smtp.port: Quota SMTP port.

4. quota.usage.smtp.user: Quota SMTP user.
5. quota.usage.smtp.password: Quota SMTP password.
6. quota.usage.smtp.sender: Quota SMTP alert sender email address.
7. quota.usage.smtp.useAuth: If true, use secure SMTP authentication when sending emails.
8. quota.usage.smtp.connection.timeout: Quota SMTP server connection timeout duration.

There are several configuration variables that are inherited from usage server, these are listed below:

1. usage.aggregation.timezone

All these are described in details in Usage Server documentation.

Restart the Management Server and the Usage Server to enable the set configuration values.

```
service cloudstack-management restart
service cloudstack-usage restart
```

Once the quota service is running it will calculate the quota balance for each account. The quota usage is calculated as per the quota tariff provided by the site administrator.

7.7.2 Quota Tariff

The following table shows all quota types for which you can specify tariff.

Type ID	Type Name	Tariff Description
1	RUNNING_VM	One month of running Compute-Month
2	ALLOCATED_VM	One month of allocated VM
3	IP_ADDRESS	Quota for a month of allocated IP
4	NETWORK_BYTES_SENT	Quota for 1GB bytes sent
5	NETWORK_BYTES_RECEIVED	Quota for 1GB bytes sent
6	VOLUME	Quota for 1 GB of Volume use for a month
7	TEMPLATE	Quota for 1 GB of Template use for a month
8	ISO	Quota for 1 GB of ISO use for a month
9	SNAPSHOT	Quota for 1 GB of SNAPSHOT use for a month
11	LOAD_BALANCER_POLICY	Quota for load balancer policy month
12	PORT_FORWARDING_RULE	Quota for port forwarding policy month
13	NETWORK_OFFERING	Quota for network Offering for a month
14	VPN_USERS	Quota for VPN usage for a month
15	CPU_CLOCK_RATE	The tariff for using 1 CPU i100 MHz clock
16	CPU_NUMBER	The quota tariff for using 1 virtual CPU.
17	MEMORY	The quota tariff for using 1MB RAM size.

The quota tariff can be listed using listQuotaTariff API.

quotaTariff: Lists all quota tariff plans

The tariff for each of the above can be set by using the updateQuotaTariff API.

7.7.3 Quota Credits

The quota credit (quotaCredit) API lets you add or remove quota currency credits to an account. With this API you can also control the quota enforcement policy at account level. This will enable you to have some accounts where

the quota policy is not enforced. The overall quota enforcement is controlled by the quota.enable.enforcement global setting.

In addition to above the quota API lets you can fine tune the alert generation by specifying the quota threshold for each account. If not explictly stated, the threshold is taken as 80% of the last deposit.

7.7.4 Quota Balance

Quota balance API states the start balance and end balance(optional) from a start date to end date (optional).

7.7.5 Quota Statement

Quota statement for a period consist of the quota usage under various quota types for the given period from a start date to an end date.

7.7.6 Quota Monthly Statement

Quota service emails the monthly quota statement for the last month at the beginning of each month. For this service to work properly you need to ensure that the usage server is running.

7.7.7 Quota Alert Management

Quota module also provides APIs to customize various email templates that are used to alert account owners about quota going down below threshold and quota getting over.

All the above functionality is also available via quota UI plugin.

CHAPTER 8

Release Notes

This document contains information specific to this release of CloudStack, including upgrade instructions from prior releases, new features added to CloudStack, API changes, and issues fixed in the release. For installation instructions, please see the CloudStack Installation Guide. For usage and administration instructions, please see the CloudStack Administrator's Guide.

Contents:

8.1 What's New in 4.11.1.0

8.1.1 What's New in 4.11.1.0

The new 4.11.1.0 version is a 4.11 maintainance release containing over 140 fixes and improvements on the 4.11.0.0 release.

These include the speeding up of virtual router deployments and fixes for corner cases effecting the new config drive and L2 features and some hypervisor specific fixes to improve compatibility with current hypervisor versions including the XCP-ng fork of XenServer.

8.1.2 What's New in 4.11.0.0

Version 4.11.0.0 included more than 400 commits, 220 pull requests that fixes more than 250 issues since the last release. Version 4.11.0.0 was a large release that was worked on for 8 months.

A LOT changed in this release, so this is not a complete list, but here is a quick summary of some of the changes:

- Support for XenServer 7.1, 7.2, 7.3 and 7.4, and support for XCP-ng 7.4.
- Improved support for VMware 6.5.
- Host-HA framework and HA-provider for KVM hosts with and NFS as primary storage, and a new background polling task manager.
- Secure agents communication: new certificate authority framework and a default built-in root CA provider.

- New network type - L2.
- CloudStack metrics exporter for Prometheus.
- Cloudian Hyperstore connector for CloudStack.
- Annotation feature for CloudStack entities such as hosts.
- Separation of volume snapshot creation on primary storage and backing operation on secondary storage.
- Limit admin access from specified CIDRs.
- Expansion of Management IP Range.
- Dedication of public IPs to SSVM and CPVM.
- Support for separate subnet for SSVM and CPVM.
- Bypass secondary storage template copy/transfer for KVM.
- Support for multi-disk OVA template for VMware.
- Storage overprovisioning for local storage.
- LDAP mapping with domain scope, and mapping of LDAP group to an account.
- Move user across accounts.
- Managed storage enhancements.
- Extend config drive support for user data, metadata, and password (Nuage Networks).
- Extra DHCP options support (Nuage Networks).
- Nuage VSP 5.0 support and caching of NuageVsp ID's.
- Nuage domain template selection per VPC and support for network migration.
- Support for watchdog timer to KVM Instances.
- Support for Secondary IPv6 Addresses and Subnets.
- IPv6 Prefix Delegation support in basic networking.
- Ability to specific MAC address while deploying VM or adding a NIC to a VM.
- VMware dvSwitch security policies configuration in network offering
- Allow more than 7 NICs to be added to a VMware VM.
- Network rate usage for guest offering for VRs.
- Usage metrics for VM snapshot on primary storage.
- Enable Netscaler inline mode.
- NCC integration in CloudStack.
- The retirement of the Midonet network plugin.
- Several UI Improvements.
- Embedded Jetty and improved CloudStack management server configuration.
- Improved support for Java 8 for building artifacts/modules, packaging, and in the systemvm template.
- A faster console proxy startup and service availability.
- A new Debian 9 based smaller systemvm template that patches systemvm without requiring reboot.

- Several optimizations and improvements to the virtual router including better support for redundant virtual routers and strongswan provided s2s and remote access vpn.

8.2 Issues Fixed in 4.11.1.0

Apache CloudStack uses Jira to track its issues and Github for pull requests. All new features and bugs for 4.11 have been merged through Github pull requests. A subset of these changes are tracked in Jira, which have a standard naming convention of "CLOUDSTACK-NNNN" where "NNNN" is the issue number.

8.2.1 Issues Fixed in 4.11.1.0

Branches	Github	Jira	Type	Priority	Description
4.11	#2712				reuse ip for non redundant VPC
4.11	#2714				send unsupported answer only when applicable
4.11	#2715				smoketest: Fix test_vm_life_cycle secure migration tes
4.11	#2493	CLOUDSTACK-10326	Bug	Major	Prevent hosts fall into Maintenance when there are runn
4.11	#2716				configdrive: make fewer mountpoints on hosts
4.11	#2681				Source NAT option on Private Gateway
4.11	#2670				Removing an old, unused NetApp plug-in
4.11	#2710				comply with api key constraint
4.11	#2699				remove old config artifacts from update path
4.11	#2705	CLOUDSTACK-10381	Bug	Blocker	[ConfigDrive] Password is missing after reset password
4.11	#2704				ui: fix create VPC dialog box failure when zone is SG e
4.11	#2693				Fix to enable Advanced zones with Security groups and
4.11	#2697				agent: Avoid sudo, renew certificates assuming root
4.11	#2686				Fixes #2685: broken SXM support
4.11	#2696				Fix LibvirtStorageAdaptor.java
4.11	#2683				Add default L2 network offerings
4.11	#2694				Do not send conserve mode param on L2 network offer
4.11	#2688	CLOUDSTACK-10382	Bug	Major	[ConfigDrive] cloud-get-vm-data-configdrive.in is incor
4.11	#2672	CLOUDSTACK-10377	Bug	Major	Nuage VSP regression fails in NetworksWithCleanup te
4.11	#2674				Create unit test cases for 'ConfigDriveBuilder' class
4.11	#2673				Fix test_configdrive.py and test_nuage_configdrive
4.11	#2676				Fix two typos (from uanble to unable).
4.11	#2669				conditional template filter
4.11	#2664				revert dedicate vlan code removal
4.11	#2663				server: Calculate fresh capacity per VM
4.11	#2667	CLOUDSTACK-10375	Bug	Minor	Do not create DefaultNuageVspSharedNetworkOffering
4.11	#2661				Make uploadSslCert a POST request instead of a GET
4.11	#2639	CLOUDSTACK-10276	Bug	Major	View volumes from primary storage not working when
4.11	#2653				Generate MAC address if the MAC in command addNic
4.11	#2656				Only perform certain actions here with managed storag
4.11	#2651	CLOUDSTACK-10290	Bug	Major	Config drive - only supported for secondary storage
4.11	#2473	CLOUDSTACK-10309	Improvement	Minor	VMs with HA enabled, power back on if shutdown fron
4.11	#2655				Handle Ceph.
4.11	#2630				Host Affinity plugin
4.11	#2652				Fix register iso in all zones
4.11	#2629				Fix primary storage count when deleting volumes

Table 1 – continued from previous page

Branches	Github	Jira	Type	Priority	Description
4.11	#2638				agent: Fixes #2633 don't wait for pending tasks on reco
4.11	#2645				config-drive: use hostname of VM instance of internal V
4.11	#2646				Don't skip tests while packaging Centos7
4.11	#2635				router: Fixes #2544 run passwd server on dhcpserver IP
4.11	#2634	CLOUDSTACK-9184	Bug	Major	[VMware] vmware.ports.per.dvportgroup global setting
4.11	#2508	CLOUDSTACK-9114	Improvement	Major	restartnetwork with cleanup should not update/restart be
4.11	#2584				Enhance and cleanup DatabaseUpgradeChecker
4.11	#2600	CLOUDSTACK-10362	Improvement	Major	Inconsistent method names
4.11	#2627				Catch error in packagin script and fail the build
4.11	#2615				config-drive: support user data on L2 networks
4.11	#2632				listostypes: Fixes #2529 return boolean than string in re
4.11	#2621				Backports for 4.11 branch
4.11	#2619				Remove "self-injection" of AccountManagerImpl
4.11	#2607				Allow changing disk offering of VMs' root volume duri
4.11	#2626				bionic: allow Ubuntu 18.04 to be added as KVM host
4.11	#2623				fixes #2611
4.11	#2628				Create upgrade path from 4.9.3.1 to 4.11.1.0
4.11	#2612				[migrateVolume API method] Filter disk offerings based
4.11	#1940	CLOUDSTACK-9781	Bug	Major	ACS records ID in events tables instead of UUID.
4.11	#2608				API: move ostypeid from DB id to DB uuid
4.9*	#2471	CLOUDSTACK-10311	Improvement	Major	Agent Log Rotate variable replace bug
4.11	#2606				When creating a new account (via domain admin) it is p
4.11	#2601	CLOUDSTACK-10364	Improvement	Major	Inconsiste "setXXX" method names.
4.11	#2599	CLOUDSTACK-10363	Improvement	Major	Inconsistent "getXXX" and "listXXX" method names.
4.11	#2598	CLOUDSTACK-10360	Improvement	Major	Inconsistent method name
4.11	#2605				xenserver: Add support for XS 7.3, 7.4 and XCP-ng 7.4
4.11	#2428	CLOUDSTACK-10253	Bug	Minor	JSON response returns boolean as string
4.11	#2536				fix typo c&p bug in externalId feature UI
4.11	#2486	CLOUDSTACK-10323	Improvement	Major	Change disk offering when volume is migrated to differ
4.11	#2422	CLOUDSTACK-10254	Improvement	Major	Require checkstyle to verify package names against dire
4.11	#2566	CLOUDSTACK-10288	Bug	Major	Config drive - Usedata corruption when gzipped
4.11	#2573	CLOUDSTACK-10356	Bug	Major	Fix Some Potential NPE
4.11	#2412	CLOUDSTACK-9677	Improvement	Major	Swift Storage Policy support for Secondary Storage
4.11	#2594				Remove 'NetworkManagerTestComponentLibrary' emp
4.11	#2597				UpdateUserCmd: apiSecretKey refers to itself
4.11	#2498	CLOUDSTACK-10327	Bug	Critical	SSO fails with error "Session Expired", except for root
4.11	#2591	CLOUDSTACK-10359	Improvement	Major	Inconsistent method names
4.11	#2590				network: Fix security groups for CentOS
4.11	#2582				cs-45to411-ugrade-fix: database errors during upgrade t
4.11	#2577				Prevent NPE if guest OS mapping is missing while prio
4.11	#2579				router: fix routing table for marked packets
4.11	#2589				Remove packaging job from pull request template
4.11	#2588				capacity: remove unused threadpool
4.11	#2505	CLOUDSTACK-10333	Improvement	Major	Secure VM Live migration for KVM
4.11	#2580	CLOUDSTACK-10357	Improvement	Minor	Log messages that do not match with their method func
4.11	#2587				Remove empty VPN test class
4.11	#2586				Use double quotes with 'RROUTER' variable in "comm
4.11	#2576				Fix Python code checkstyle execute by "systemvmtestru
4.11	#2562				consoleproxy: use consoleproxy.domain for non-ssl ena

Table 1 – continued from previous page

Branches	Github	Jira	Type	Priority	Description
4.11	#2554				agent: Add logging to libvirt qemu hook and cleanup
4.11	#2511	CLOUDSTACK-10344	Bug	Major	Sometimes a bug happens when moving ACL rules (cha
4.11	#2572				Remove 'todb' in favor of 'encodeURIComponent'.
4.11	#2553				Update inconsistent debugging info in catch block
4.11	#2499				Updates to capacity management
4.11	#2570				Improve README
4.11	#2568				Log command output in CsHelper.execute command
4.11	#2559				Upgrade path 4.11 through 4.11.1 to 4.12
4.11	#2567				[Vmware] Fix for OVF parsing error
4.11	#2563	CLOUDSTACK-10304	Bug	Major	SystemVM - Apache Web Server Version Number Infor
4.11	#2555				Remove 'md5Hashed' variable from Javascript.
4.11	#2390	CLOUDSTACK-10214	Bug	Major	Unable to remove local primary storage
4.11	#2564				[Docs] Fix URL error from installation instructions
4.11	#2557				Add "Fixes: number" to PR template for auto-closing is
4.11	#2404	CLOUDSTACK-10230	Bug	Critical	User is able to change to ?Guest OS type? that has been
4.11	#2550				debian: Use only *-l* for libvirtd default file on debian
4.11	#2560				ui: Make zonal dashboard larger
4.11	#2401	CLOUDSTACK-10226	Bug	Major	CloudStack is not importing Local storage properly
4.11	#2462	CLOUDSTACK-10301	Bug	Major	Allow updating the network ACL list name and Descrip
4.11	#2490				Create database upgrade from 4.11.0.0 to 4.11.1.0 & VI
4.11	#2538				Remove deprecated tomcat configuration file instead of
4.11	#2517				manual mapped ldap fix
4.11	#2552				debian: remove old usage jars during upgrade
4.11	#2535				Create an easy way to enable Java remote Debug for AC
4.11	#2526				add issue template for github issues
4.11	#2522				indicate scope of tests in checklist
4.11	#2519	CLOUDSTACK-10287	Bug	Major	Make el7 rpms to depend on OpenJDK8
4.11	#2520				make Broadcast- and IsolationURI visible to admin
4.11	#2515				Fix Successfully typo
4.11	#2414	CLOUDSTACK-10241	Bug	Major	Duplicated file SRs being created in XenServer pools
4.11	#2512				Only use the host if its Resource State is Enabled.
4.11	#2492				Fix the name of the column used to hold IPv4 range in
4.11	#2496	CLOUDSTACK-10332	Enhancement	Major	Users are not able to change/edit the protocol of an ACL
4.11	#2449	CLOUDSTACK-10278	Bug	Major	Adding a SQL table column is not Idempotent
4.11	#2510	CLOUDSTACK-10334	Improvement	Major	Inadequate information for handling catch clauses
4.11	#2506	CLOUDSTACK-10341	Task	Major	Systemvmtemplate 4.11 changes
4.11	#2513	CLOUDSTACK-10227	Bug	Blocker	Stabilization fixes for master/4.11
4.11	#2465	CLOUDSTACK-10232	Enhancement	Major	SystemVMs and VR to run as HVM on XenServer
4.11	#2438	CLOUDSTACK-10307	Improvement	Minor	Remove unused things from HostDaoImpl
4.11	#2507	CLOUDSTACK-10319	Bug	Major	Prefer TLSv1.2 and deprecate TLS 1.0/1.1
4.11	#2397	CLOUDSTACK-10221	Improvement	Major	Allow specification of IPv6 details when creating Basic
4.11	#2481	CLOUDSTACK-10320	Bug	Major	Invalid pair for response object breaking response parsi
4.11	#2468	CLOUDSTACK-10341	Task	Major	Systemvmtemplate 4.11 changes
4.11	#2504	CLOUDSTACK-10340	Task	Major	Add setter in vminstancevo
4.11	#2497	CLOUDSTACK-10331	Bug	Minor	Error 404 for /client/scripts/vm_snapshots.js
4.11	#2408	CLOUDSTACK-10231	Bug	Major	Asserted fixes for Direct Download on KVM
4.11	#2494	CLOUDSTACK-10329	Enhancement	Major	Button in ACL rules page to export all rules as a CSV fi
4.11	#2495				Fix typo in Packaging script
4.11	#2489	CLOUDSTACK-10330	Improvement	Major	Introduce a standard PULL REQUEST template

Table 1 – continued from previous page

Branches	Github	Jira	Type	Priority	Description
4.11	#2491				Fix "agent-lb" project
4.11	#2469	CLOUDSTACK-10132	Improvement	Major	Extend Multiple Management Servers Support for agen
4.11	#2458	CLOUDSTACK-10296	Bug	Major	Fix timestamp difference in heartbeat script for rVRs
4.11	#2433	CLOUDSTACK-10268	Improvement	Minor	Fix and enhance package script
4.11	#2387	CLOUDSTACK-8855	Bug	Major	Improve Error Message for Host Alert State
4.11	#2482	CLOUDSTACK-10321	Bug	Major	CPU Cap for KVM
4.11	#2483	CLOUDSTACK-10303	Improvement	Major	refactor plugins/nuagevsp tests to run from its own test_
4.11	#2442	CLOUDSTACK-10147	Bug	Major	Disabled Xenserver cluster can still deploy VMs
4.11	#2484				createNetworkACL: number has the wrong doc
4.11	#2475	CLOUDSTACK-10314	Improvement	Minor	Add Text-Field to each ACL Rule
4.11	#2485				Bump the version of Debian net-installer to 9.4.0
4.11	#2480	CLOUDSTACK-10319	Bug	Major	Prefer TLSv1.2 and deprecate TLS 1.0/1.1
4.11	#2470	CLOUDSTACK-10197	Bug	Major	XenServer 7.1: Cannot mount xentool iso from cloudsta
4.11	#2476	CLOUDSTACK-10317	Bug	Minor	In case of multiple-public ips, snat fails to work for add
4.11	#2425	CLOUDSTACK-10240	Improvement	Major	ACS cannot migrate a volume from local to shared stora
4.11	#2448	CLOUDSTACK-10274	Bug	Major	L2 network refused to be designed on VXLAN physica
4.11	#2478	CLOUDSTACK-10318	Bug	Major	Bug on sorting ACL rules list in chrome
4.11	#2437	CLOUDSTACK-10257	Improvement	Minor	Create template/volume does not allow to specify HVM
4.11	#2439	CLOUDSTACK-10259	Bug	Minor	Missing float part of secondary storage data when calcu
4.11	#2392				dateutil: constistency of tzdate input and output
4.11	#2463	CLOUDSTACK-10302	Task	Major	Create database path upgrade from 4.11.0.0 to 4.12.0.0
4.11	#2464	CLOUDSTACK-10299	Bug	Minor	Network listing return error in project mode
4.11	#2244	CLOUDSTACK-10054	Bug	Major	Volume download times out in 3600 seconds
4.11	#2467	CLOUDSTACK-10306	Bug	Minor	Update vmware dependency vim jar to 6.5 version
4.11	#2460	CLOUDSTACK-10298	Bug	Major	Recreation of an earlier deleted Nuage managed isolate
4.11	#2466	CLOUDSTACK-10305	Bug	Major	Rare race condition in KVM migration
4.11	#2443	CLOUDSTACK-9338	Bug	Major	updateResourceCount not accounting resources of VMs
4.11	#2451	CLOUDSTACK-10284	Bug	Major	Creating a snapshot from VM Snapshot generates error
4.11	#2454	CLOUDSTACK-10283	Bug	Major	Use sudo to execute keystore setup/import for kvm agen
4.11	#2457	CLOUDSTACK-10295	Improvement	Major	Marvin: add support for password-enabled templates
4.11	#2456	CLOUDSTACK-10293	Bug	Major	Single view network ACL rules listing
4.11	#2402	CLOUDSTACK-10128	Bug	Critical	Template from snapshot not merging vhd files
4.11	#2432	CLOUDSTACK-10294	Improvement	Major	Updated code-styling and improvements to security_gro
4.11	#2450	CLOUDSTACK-10282	Bug	Major	SystemVM - filrewall rules incorrect
4.11	#2452	CLOUDSTACK-10285	Bug	Critical	4.10.0.0 users face upgrade issues when upgrading to 4.
4.11	#2441	CLOUDSTACK-10261	Bug	Critical	Nuage: Multinic: Libvirt nuage-extenstion metadata co
4.11	#2420	CLOUDSTACK-10247	Bug	Major	L2 network not shared on projects
4.11	#2424	CLOUDSTACK-10251	Bug	Major	HTTPS downloader for Direct Download templates fail
4.11	#2421	CLOUDSTACK-10243	Bug	Major	Updating metadata might hang on VR on "ip rule show"
4.11	#2406	CLOUDSTACK-9663	Improvement	Trivial	updateRole should return updated role as json
4.11	#2445	CLOUDSTACK-10218	Bug	Major	forced network update to a nuage network offering with
4.11	#2444	CLOUDSTACK-10269	Bug	Major	Allow deletion of roles without causing concurrent exce
4.11	#2405	CLOUDSTACK-10146	Bug	Major	Bypass Secondary Storage for KVM templates
4.11	#2398	CLOUDSTACK-10222	Bug	Major	Clean previous snaphosts from primary storage when ta
4.11	#2431	CLOUDSTACK-10225	Improvement	Major	Deprecate com.cloud.utils.StringUtils

8.2.2 Issues Fixed in 4.11.0.0

Branches	Github	Jira	Type	Priority	Description
4.11	#2097	CLOUDSTACK-9813	New Feature	Major	Use configdrive for userdata, metadata & password
4.11	#2146	CLOUDSTACK-4757	New Feature	Minor	Support OVA files with multiple disks for templates
4.11	#2394	CLOUDSTACK-10109	New Feature	Major	Enable dedication of public IPs to SSVM and CPVM
4.11	#2295	CLOUDSTACK-10109	New Feature	Major	Enable dedication of public IPs to SSVM and CPVM
4.11	#2381	CLOUDSTACK-10117	New Feature	Major	LDAP mapping on domain level
4.11	#2368	CLOUDSTACK-10126	New Feature	Major	Separate Subnet for CPVM and SSVM
4.11	#2046	CLOUDSTACK-7958	New Feature	Minor	Limit user login to specific subnets
4.11	#2374	CLOUDSTACK-10024	New Feature	Major	Physical Networking Migration
4.11	#2301	CLOUDSTACK-10121	New Feature	Major	move user API
4.11	#2360	CLOUDSTACK-10189	New Feature	Major	Support for "VSD managed" networks with Nuage Netw
4.11	#2281	CLOUDSTACK-10102	New Feature	Major	New Network Type (L2)
4.11	#2048	CLOUDSTACK-9880	New Feature	Major	Expansion of Management IP Range.
4.11	#2294	CLOUDSTACK-10039	New Feature	Major	Adding IOPS/GB offering
4.11	#2356	CLOUDSTACK-8313	New Feature	Major	Local Storage overprovisioning should be possible
4.11	#2028	CLOUDSTACK-9853	New Feature	Major	IPv6 Prefix Delegation support in Basic Networking
4.11	#1981	CLOUDSTACK-9806	New Feature	Major	Nuage domain template selection per VPC
4.11	#2325	CLOUDSTACK-9998	New Feature	Major	CloudStack exporter for Prometheus
4.11	#2284	CLOUDSTACK-10103	New Feature	Major	Cloudian Connector for CloudStack
4.11	#2297	CLOUDSTACK-9957	New Feature	Major	Annotations on entities
4.11	#2181	CLOUDSTACK-9957	New Feature	Major	Annotations on entities
4.11	#2286	CLOUDSTACK-9993	New Feature	Major	Secure Agent Communications
4.11	#2287	CLOUDSTACK-9998	New Feature	Major	CloudStack exporter for Prometheus
4.11	#2278	CLOUDSTACK-9993	New Feature	Major	Secure Agent Communications
4.11	#1707	CLOUDSTACK-9397	New Feature	Major	Add Watchdog timer to KVM Instances
4.11	#2143	CLOUDSTACK-9949	New Feature	Minor	add ability to specify mac address when deployVirtualM
4.11	#2256	CLOUDSTACK-9782	New Feature	Major	Host HA
4.11	#2239	CLOUDSTACK-9993	New Feature	Major	Secure Agent Communications
4.11	#2222	CLOUDSTACK-10022	New Feature	Minor	Allow domain admin to create and delete subdomains
4.11	#2026	CLOUDSTACK-9861	New Feature	Major	Expire VM snapshots after configured duration
4.11	#2218	CLOUDSTACK-9782	New Feature	Major	Host HA
4.11*	#2407	CLOUDSTACK-10227	Bug	Blocker	Stabilization fixes for master/4.11
4.11	#2403	CLOUDSTACK-10227	Bug	Blocker	Stabilization fixes for master/4.11
4.11	#2396	CLOUDSTACK-10220	Bug	Major	IPv4 NIC alias is not added on Virtual Router in Basic N
4.11	#2362	CLOUDSTACK-10188	Bug	Major	Resource Accounting for primary storage is Broken
4.11	#2393	CLOUDSTACK-10217	Bug	Major	When IPv4 address of Instance is updated DHCP data is
4.11	#2388	CLOUDSTACK-10212	Bug	Major	Changing IPv4 Address of NIC in Basic Networking do
4.11	#2379	CLOUDSTACK-10146	Bug	Major	Bypass Secondary Storage for KVM templates
4.11	#2391	CLOUDSTACK-10215	Bug	Major	Excessive log4j debug level in CPVM, SSVM could lea
4.11	#2139	CLOUDSTACK-9921	Bug	Major	NPE when garbage collector is running
4.11	#2088	CLOUDSTACK-9892	Bug	Critical	Primary storage resource check is broken when using rc
4.11	#2365	CLOUDSTACK-10197	Bug	Major	XenServer 7.1: Cannot mount xentool iso from cloudsta
4.11	#2073	CLOUDSTACK-9896	Bug	Minor	ListDedicatedXXX doesn't respect pagination
4.11	#2386	CLOUDSTACK-9632	Bug	Major	Upgrade bountycastle to 1.55+
4.11	#2315	CLOUDSTACK-9025	Bug	Critical	Unable to deploy VM instance from template if templat
4.11	#2274	CLOUDSTACK-10096	Bug	Major	Can't reset api.integration.port and usage.sanity.check.in
4.11	#2282	CLOUDSTACK-10104	Bug	Major	Optimize database transactions in ListDomain API to in
4.11	#2385	CLOUDSTACK-10211	Bug	Major	test_nuage_public_sharednetwork_userdata fails due to
4.11	#2260	CLOUDSTACK-10065	Bug	Major	Optimize SQL queries in listTemplate API to improve p
4.11	#1740	CLOUDSTACK-9572	Bug	Major	Snapshot on primary storage not cleaned up after Storag
4.11	#2104	CLOUDSTACK-9908	Bug	Major	Primary Storage allocated capacity goes very high after

Table 2 – continued from previou

Branches	Github	Jira	Type	Priority	Description
4.11	#2378	CLOUDSTACK-10205	Bug	Major	LinkDomainToLdap returns internal id
4.11	#1773	CLOUDSTACK-9607	Bug	Major	Preventing template deletion when template is in use.
4.11	#2149	CLOUDSTACK-9932	Bug	Major	Snapshot is getting deleted while volume creation from
4.11	#2156	CLOUDSTACK-9971	Bug	Minor	Bugfix/listaccounts parameter consistency
4.11	#2373	CLOUDSTACK-10202	Bug	Major	createSnapshotPolicy API create multiple entries in DB
4.11	#2344	CLOUDSTACK-10163	Bug	Major	Component tests health check
4.11	#1760	CLOUDSTACK-9593	Bug	Major	User data check is inconsistent with python
4.11	#2352	CLOUDSTACK-10175	Bug	Major	Listing VPCs with a domain account and project id -1 r
4.11	#2347	CLOUDSTACK-10166	Bug	Minor	Cannot add a tag to a NetworkACL (rule not list) in CS
4.11	#2375	CLOUDSTACK-9456	Bug	Major	Migrate master to use Java8 and Spring4
4.11	#2211	CLOUDSTACK-10013	Bug	Major	Migrate to Debian9 systemvmtemplate
4.9*	#2304	CLOUDSTACK-10127	Bug	Critical	4.9 / 4.10 KVM + openvswitch + vpc + static nat / secor
4.11	#2208	CLOUDSTACK-9542	Bug	Major	listNics API does not return data as per API documenta
4.11	#2351	CLOUDSTACK-10173	Bug	Major	Guest/Public nics on VR should pick network rate from
4.11	#2370	CLOUDSTACK-9595	Bug	Major	Transactions are not getting retried in case of database c
4.11	#2366	CLOUDSTACK-10168	Bug	Major	VR duplicate entries in /etc/hosts when reusing VM nar
4.11	#2042	CLOUDSTACK-9875	Bug	Major	Unable to re-apply Explicit dedication to VM
4.11	#2364	CLOUDSTACK-10195	Bug	Minor	CloudStack MySQL HA problem – No database selecte
4.11	#2361	CLOUDSTACK-10190	Bug	Major	Duplicate public VLAN for two different admin accoun
4.11	#2247	CLOUDSTACK-10012	Bug	Major	Migrate to Embedded Jetty based mgmt server
4.11	#1964	CLOUDSTACK-9800	Bug	Major	Enable inline mode for NetScaler device
4.11	#1762	CLOUDSTACK-9595	Bug	Major	Transactions are not getting retried in case of database c
4.11	#2308	CLOUDSTACK-8908	Bug	Major	After copying the template charging for that template is
4.11	#2354	CLOUDSTACK-10176	Bug	Major	VM Start Api Job returns success for failed Job
4.11	#2353	CLOUDSTACK-9986	Bug	Major	Consider overcommit ratios with total/threshold values
4.11	#2358	CLOUDSTACK-9736	Bug	Minor	Incoherent validation and error message when you chan
4.11	#2326	CLOUDSTACK-10184	Bug	Major	Re-work method QuotaResponseBuilderImpl.startOfNe
4.11	#2267	CLOUDSTACK-10077	Bug	Major	Allow account to use the same site-2-site VPN gateway
4.11	#2337	CLOUDSTACK-10157	Bug	Major	Wrong notification while migration
4.11	#2355	CLOUDSTACK-10177	Bug	Major	NPE when programming Security Groups with KVM
4.11	#2349	CLOUDSTACK-10070	Bug	Major	Extend travis run to include more component tests
4.11	#2312	CLOUDSTACK-7793	Bug	Critical	[Snapshots]Create Snaphot with “quiesce” option set to
4.11	#2345	CLOUDSTACK-10164	Bug	Blocker	UI - not able to create a VPC
4.11	#2263	CLOUDSTACK-10070	Bug	Major	Extend travis run to include more component tests
4.11	#2342	CLOUDSTACK-9586	Bug	Critical	When using local storage with Xenserver prepareTempl
4.11	#2124	CLOUDSTACK-9432	Bug	Critical	Dedicate Cluster to Domain always creates an affinity g
4.11	#2322	CLOUDSTACK-10140	Bug	Critical	When template is created from snapshot template.prope
4.11	#2335	CLOUDSTACK-10154	Bug	Major	test failures in smoketest
4.11	#2341	CLOUDSTACK-10160	Bug	Major	KVM VirtIO-SCSI not defined properly in Libvirt XML
4.11	#2321	CLOUDSTACK-10138	Bug	Major	Load br_netfilter in security_group management script
4.11	#2334	CLOUDSTACK-10152	Bug	Major	egress destination cidr with 0.0.0.0/0 is failing
4.11	#2310	CLOUDSTACK-10133	Bug	Major	Local storage overprovisioning for ext file system
4.11	#2303	CLOUDSTACK-10123	Bug	Major	VmWork job gets deleted before the parent job had time
4.11	#2329	CLOUDSTACK-10012	Bug	Major	Migrate to Embedded Jetty based mgmt server
4.11	#2327	CLOUDSTACK-10129	Bug	Trivial	Show instances attached to a network/VR via navigatior
4.11	#2313	CLOUDSTACK-10135	Bug	Major	ACL rules order is not maintained for ACL_OUTBOUN
4.11	#2316	CLOUDSTACK-10137	Bug	Major	Re-installation fails for cloudstack-management
4.11	#2157	CLOUDSTACK-9961	Bug	Major	Accept domain name for gateway while creating Vpncu
4.11	#2306	CLOUDSTACK-10129	Bug	Trivial	Show instances attached to a network/VR via navigatior

Table 2 – continued from previou

Branches	Github	Jira	Type	Priority	Description
4.11	#2273	CLOUDSTACK-10090	Bug	Major	createPortForwardingRule api call accepts 'halt' as Prot
4.11	#2240	CLOUDSTACK-10051	Bug	Major	Mouse Scrolling is not working in instance VM console
4.11	#2291	CLOUDSTACK-10111	Bug	Minor	Fix validation for parameter "vm.password.length"
4.11	#2302	CLOUDSTACK-10122	Bug	Major	Unrelated error message
4.11	#2250	CLOUDSTACK-10057	Bug	Major	ListNetworkOfferingsCmd does not return the correct c
4.11	#2268	CLOUDSTACK-10081	Bug	Major	CloudUtils getDevInfo function only checks for KVM b
4.11	#2293	CLOUDSTACK-10047	Bug	Major	DVSwitch improvements
4.11	#2288	CLOUDSTACK-10107	Bug	Major	VMware VM fails to start if it has more than 7 nics
4.11	#2257	CLOUDSTACK-10060	Bug	Minor	ListUsage API always displays the Virtual size as '0' fo
4.11	#2246	CLOUDSTACK-10046	Bug	Major	checksum is not verified during registerTemplate
4.11	#2074	CLOUDSTACK-9899	Bug	Major	allow download without checking first for MS behind fi
4.11	#2279	CLOUDSTACK-9584	Bug	Major	Increase component tests coverage in Travis run
4.11	#2277	CLOUDSTACK-10099	Bug	Major	GUI invokes migrateVirtualMachine instead of migrate'
4.11	#2269	CLOUDSTACK-10083	Bug	Minor	SSH keys are not created when starting from maintenan
4.11	#876	CLOUDSTACK-8865	Bug	Major	Adding SR doesn't create Storage_pool_host_ref entry
4.11	#1252	CLOUDSTACK-9182	Bug	Major	Some running VMs turned off on manual migration whe
4.11	#2153	CLOUDSTACK-9956	Bug	Major	File search on the vmware datastore may select wrong f
4.11	#2078	CLOUDSTACK-9902	Bug	Minor	consoleproxy.sslEnabled global config variable is not pr
4.11	#2252	CLOUDSTACK-10067	Bug	Major	Fix a case where a user 'ro' or 'roo' exists on the systen
4.11	#2248	CLOUDSTACK-10056	Bug	Minor	Cannot specify root disk controller when creating VM
4.11	#2243	CLOUDSTACK-10019	Bug	Major	template.properties has hardcoded id
4.11	#2261	CLOUDSTACK-10068	Bug	Major	smoketest/test_iso.py reports assertion failure
4.11	#2054	CLOUDSTACK-9886	Bug	Major	After restarting cloudstack-management , It takes time t
4.11	#955	CLOUDSTACK-8969	Bug	Major	VPN customer gateway can't be registered with hostnar
4.9*	#2262	CLOUDSTACK-10069	Bug	Major	During release add sha512 suffix to sha 512 checksum/f
4.11	#2253	CLOUDSTACK-10061	Bug	Major	When starting a VM, make sure it has the correct volum
4.11	#2254	CLOUDSTACK-10058	Bug	Major	Error while opening the Settings tab in Secondary storag
4.11	#1733	CLOUDSTACK-9563	Bug	Major	ExtractTemplate returns malformed URL after migratin
4.11	#2188	CLOUDSTACK-10004	Bug	Major	On deletion, Vmware volume snapshots are left behind
4.11	#914	CLOUDSTACK-8939	Bug	Major	VM Snapshot size with memory is not correctly calcula
4.11	#1985	CLOUDSTACK-9812	Bug	Major	Update "updatePortForwardingRule" pi to include addit
4.11	#2224	CLOUDSTACK-10032	Bug	Major	Database entries for templates created from snapshots d
4.11	#2109	CLOUDSTACK-9922	Bug	Major	Unable to use 8081 port for Load balancing
4.11	#2216	CLOUDSTACK-10027	Bug	Minor	Repeating the same list for Internal LB in VPC
4.11	#2174	CLOUDSTACK-9996	Bug	Major	bug in network resource that juniper srx firewall
4.11	#2186	CLOUDSTACK-10002	Bug	Major	Restart network with cleanup spawns Redundant Router
4.11	#1246	CLOUDSTACK-9165	Bug	Major	unable to use reserved IP range in a network for externa
4.9*	#2241	CLOUDSTACK-10052	Bug	Major	Upgrading to 4.9.2 causes confusion/issues around dyna
4.11	#2221	CLOUDSTACK-10030	Bug	Minor	Public IPs assgined to the VPC are not reacheable from
4.11	#2154	CLOUDSTACK-9967	Bug	Major	[VPC] static nat on additional public subnet ip is not wo
4.11	#1878	CLOUDSTACK-9717	Bug	Major	[VMware] RVRs have mismatching MAC addresses for
4.11	#2013	CLOUDSTACK-9734	Bug	Major	Destroy VM Fails sometimes
4.11	#2159	CLOUDSTACK-9964	Bug	Critical	Snapahots are getting deleted if VM is assigned to anoth
4.11	#2163	CLOUDSTACK-9939	Bug	Major	Volumes stuck in Creating state while restarting the Ma
4.11	#1919	CLOUDSTACK-9763	Bug	Major	vpc: can not ssh to instance after vpc restart
4.11	#2215	CLOUDSTACK-10026	Bug	Major	Page for Internal LB VM stucking while loading
4.11	#2180	CLOUDSTACK-9999	Bug	Major	vpc tiers do not work if vpc has more than 8 tiers
4.11	#2223	CLOUDSTACK-10031	Bug	Major	change default configuration for router.aggregation.com
4.11	#2182	CLOUDSTACK-10000	Bug	Major	Remote access vpn user does not work if user password

Table 2 – continued from previou

Branches	Github	Jira	Type	Priority	Description
4.9*	#2233	CLOUDSTACK-10042	Bug	Major	UI doesn't show ICMP Type and Code for Security Gro
4.11	#2228	CLOUDSTACK-10036	Bug	Major	Decrease timeout of failing unit test HypervisorUtilsTes
4.11	#1774	CLOUDSTACK-9608	Bug	Major	Errored State and Abandoned state Templates are not di
4.11	#2144	CLOUDSTACK-9955	Bug	Minor	Featured Templates/Iso's created by Root/admin user ar
4.9*	#1966	CLOUDSTACK-9801	Bug	Critical	IPSec VPN does not work after vRouter reboot or recrea
4.9*	#2220	CLOUDSTACK-9708	Bug	Major	Router deployment failed due to two threads start VR si
4.11	#1912	CLOUDSTACK-9749	Bug	Critical	cloudstack master vpc_internal_lb feature is broken as p
4.11	#2138	CLOUDSTACK-9944	Bug	Major	In clustered Management Server, Sometimes hosts stays
4.11	#883	CLOUDSTACK-8906	Bug	Major	/var/log/cloud/ doesn't get logrotated on xenserver
4.11	#2119	CLOUDSTACK-9925	Bug	Minor	Quota: fix tariff description for memory. Tariff value is
4.11	#2145	CLOUDSTACK-9697	Bug	Major	Better error message on UI user if tries to shrink the VM
4.11	#2137	CLOUDSTACK-9950	Bug	Major	listUsageRecords doesnt return required fields
4.11	#2008	CLOUDSTACK-9840	Bug	Major	Datetime format of snapshot events is inconsistent with
4.11	#2142	CLOUDSTACK-9954	Bug	Major	Unable to create service offering with networkrate=0
4.11	#2171	CLOUDSTACK-9990	Bug	Minor	Account name is giving null in event tab after successfu
4.9*	#1925	CLOUDSTACK-9751	Bug	Major	if a public IP is assigned to a VM when VR is in starting
4.9*	#1798	CLOUDSTACK-9631	Bug	Major	updateVMAffinityGroup must require one of the list par
4.9*	#2201	CLOUDSTACK-10016	Bug	Major	VPC VR doesn't respond to DNS requests from remote
4.11	#1959	CLOUDSTACK-9786	Bug	Major	API reference guide entry for associateIpAddress needs
4.9*	#1933	CLOUDSTACK-9569	Bug	Critical	VR on shared network not starting on KVM
4.11	#2298	CLOUDSTACK-9620	Improvement	Major	Improvements for Managed Storage
4.11	#2152	CLOUDSTACK-10229	Improvement	Trivial	SSVM logging improvement when using Swift as secon
4.11	#2389	CLOUDSTACK-10213	Improvement	Major	Allow specify SSH key lengh
4.11	#2292	CLOUDSTACK-10108	Improvement	Minor	ConfigKey based approach for reading 'ping' configuara
4.11	#2384	CLOUDSTACK-10210	Improvement	Trivial	remove test file
4.11	#1554	CLOUDSTACK-9602	Improvement	Major	Add resource type name in response
4.11	#2035	CLOUDSTACK-9867	Improvement	Major	VM snapshots - not charged for the primary storage they
4.11	#1934	CLOUDSTACK-9772	Improvement	Major	Perform HEAD request to retrieve header information
4.11	#2348	CLOUDSTACK-10196	Improvement	Minor	Remove ejb-api 3.0 dependency
4.11	#2184	CLOUDSTACK-10003	Improvement	Major	automatic configure juniper srx/vsrx nat loopback
4.11	#2332	CLOUDSTACK-10156	Improvement	Minor	Fix Coverity new problems CID(1349987, 1349986, 13
4.11	#2219	CLOUDSTACK-9989	Improvement	Major	Extend smoketests suite
4.11	#2005	CLOUDSTACK-9450	Improvement	Major	Network Offering for VPC based on DB flag
4.11	#2242	CLOUDSTACK-9958	Improvement	Major	Include tags of resources in listUsageRecords API
4.11	#2158	CLOUDSTACK-9972	Improvement	Major	Enhance listVolume API to include physical size and ut
4.11	#2004	CLOUDSTACK-9832	Improvement	Major	Do not assign public IP NIC to the VPC VR when the V
4.11	#2238	CLOUDSTACK-10053	Improvement	Major	Performance improvement: Caching of external id's
4.11	#2296	CLOUDSTACK-10007	Improvement	Major	Isolation methods are hard coded enum, replace by regis
4.11	#2280	CLOUDSTACK-10101	Improvement	Major	Present the full domain name when listing user's domai
4.11	#2285	CLOUDSTACK-9859	Improvement	Major	Retirement of midonet plugin (final removal ticket)
4.11	#2266	CLOUDSTACK-10073	Improvement	Trivial	KVM host RAM overprovisioning
4.11	#2249	CLOUDSTACK-10007	Improvement	Major	Isolation methods are hard coded enum, replace by regis
4.11	#1443	CLOUDSTACK-9314	Improvement	Trivial	Remove unused code from XenServerStorageProcessor
4.11	#2044	CLOUDSTACK-9877	Improvement	Major	remove fully cloned deleted templates from primary sto
4.11	#2101	CLOUDSTACK-9915	Improvement	Major	ListSnapshots API does not provide virtual size informa
4.11	#2123	CLOUDSTACK-9914	Improvement	Trivial	Alter quota_tariff to support currency values up to 5 dec
4.11	#1936	CLOUDSTACK-9773	Improvement	Major	Don't break API output with non-printable characters
4.11	#2236	CLOUDSTACK-10044	Improvement	Major	Update rule permission of a role permission
4.11	#2193	CLOUDSTACK-10007	Improvement	Major	Isolation methods are hard coded enum, replace by regis

Table 2 – continued from previou

Branches	Github	Jira	Type	Priority	Description
4.11	#2130	CLOUDSTACK-8961	Improvement	Major	Making the VPN user management more intutive
4.11	#2200	CLOUDSTACK-10015	Improvement	Minor	Return storage provider with call to list storage pools
4.11	#2350				Cloudstack 10170 - fixes resource tags security bugs an
4.11	#2383				"isdynamicallyscalable" Field to UpdateTemplate Resp
4.11	#2045				Fix snmptrap alert bug
4.11	#2258				Cloudstack 10064: Secondary storage Usage for upload
4.11	#1637				Command route not available on CentOS 7
4.11	#2367				Fix ACL_INBOUND/OUTBOUND rules for PrivateGa
4.11	#2371				README: Happy Holidays, may the cloud be with you
4.11	#1437				removed unused HypervDummyResourceBase class
4.11	#2346				Add XenServer 7.1 and 7.2 interoperablility
4.11	#2359				doc: replace virutal by virtual (typo)
4.11	#2324				Remove annotation and "depends-on" declaration not n
4.11	#1723				Fix GroupBy (+ having) condition and tests
4.11	#2307				packging: Raise compat mode to 9
4.11	#2245				increased jetty timeout
4.11	#2235				repo has moved
4.11	#2039				rbd: Use libvirt to create new volumes and not rados-jav
4.9*	#2094				Agent logrotation
4.9*	#2212				appliance: fix progress version in Gemfile
4.11	#2205				Add NULL checks for various objects in SolidFire integ
4.11	#1784				CS-505: Marvin test to check VR internal DNS Service
4.11	#1655				Fix ajaxviewer.js to solve console on Firefox
4.11	#2175				4.10 to 4.11 upgrade path
4.10*	#2176				Travis: use oraclejdk8 for 4.10+

8.3 Compatibility Matrix

8.3.1 Supported OS Versions for Management Server

This section lists the operating systems that are supported for running CloudStack Management Server. Note that specific versions of the operating systems are tested, so compatibility with CentOS 6.3 may not indicate compatibility with CentOS 6.2, 6.1 and so on.

- RHEL versions 6.3, 6.5, 6.6 and 7.0
- CentOS versions 6.8, 7
- Ubuntu 14.04, 16.04, 17.04 LTS

Software Requirements

- Java JRE 1.8
- MySQL 5.6, 5.7 (RHEL 7)
- MySQL 5.1 (RHEL 6.x)

8.3.2 Supported Hypervisor Versions

CloudStack supports three hypervisor families, XenServer with XAPI, KVM, and VMware with vSphere.

- CentOS 6.2+, 7.0+ with KVM
- Ubuntu 14.04LTS, 16.04LTS, 18.04LTS with KVM
- Red Hat Enterprise Linux 6.2 with KVM
- XenServer versions 6.1, 6.2 SP1, 6.5, 7.0, 7.1, 7.2 with latest hotfixes, XCP-ng 7.4

  Note: It is now required to enable HA on the XenServer pool in order to recover from a pool-master failure. Please refer to the XenServer documentation.

- VMware versions 5.0 Update 3, 5.1 Update 3, 5.5 Update 3b, 6.0 Update 2, and 6.5 GA
- LXC Host Containers on RHEL 7
- Windows Server 2012 R2 (with Hyper-V Role enabled)
- Hyper-V 2012 R2
- Oracle VM 3.0+
- Bare metal hosts are supported, which have no hypervisor. These hosts can run the following operating systems:
  - RHEL or CentOS, v6.2 or 6.3

    Note: Use libvirt version 0.9.10 for CentOS 6.3

  - Fedora 17
  - Ubuntu 12.04

For more information, see the Hypervisor Compatibility Matrix in the CloudStack Installation Guide.

8.3.3 Supported External Devices

- Netscaler VPX and MPX versions 9.3, 10.1e and 10.5
- Netscaler SDX version 9.3, 10.1e and 10.5
- SRX (Model srx100b) versions 10.3 to 10.4 R7.5
- F5 11.X
- Force 10 Switch version S4810 for Baremetal Advanced Networks

8.3.4 Supported Browsers

The CloudStack Web-based UI should be compatible with any modern browser, but it's possible that some browsers will not render portions of the UI reliably, depending on their support of Web standards. For best results, one of the following browsers recommended:

- Firefox version 31 or later
- Google Chrome version 36.0+
- Safari 6+

8.3.5 Notice Of Management OSes and Hypervisors to be Deprecated

The following hypervisors will no longer be tested from the first release AFTER 1st June 2018. And will be removed from the CloudStack codebase from the first release after 1st November 2018.

- XenServer 6.2
- XenServer 6.5
- vSphere 5.0
- vSphere 5.1

The following hypervisors will no longer be tested from the first release AFTER 1st February 2019. And will be removed from the CloudStack codebase from the first release after 1st July 2019.

- vSphere 5.5

Please see CloudStack Wiki for details.

8.4 API Changes Introduced in 4.11.1.0

For the complete list of API commands and params consult the CloudStack Apidocs.

8.4.1 New API Commands

Name	Description
`provisionCertificate`	Issues and propagates client certificate on a connected host/agent using c
`listElastistorPool`	Lists the pools of elastistor
`deleteServicePackageOffering`	Delete Service Package
`listAnnotations`	Lists annotations.
`enableHAForZone`	Enables HA for a zone
`enableHAForCluster`	Enables HA cluster-wide
`listNuageVspDomainTemplates`	Lists Nuage VSP domain templates
`listElastistorInterface`	Lists the network Interfaces of elastistor
`stopNetScalerVpx`	Stops a NetScalervm.
`disableHAForZone`	Disables HA for a zone
`revokeCertificate`	Revokes certificate using configured CA plugin
`updateSiocInfo`	Update SIOC info
`cloudianSsoLogin`	Generates single-sign-on login url for logged-in CloudStack user to acce
`issueCertificate`	Issues a client certificate using configured or provided CA plugin
`listNetscalerControlCenter`	list control center
`listCAProviders`	Lists available certificate authority providers in CloudStack
`acquirePodIpAddress`	Allocates IP addresses in respective Pod of a Zone
`deleteManagementNetworkIpRange`	Deletes a management network IP range. This action is only allowed wh
`addAnnotation`	add an annotation.
`deployNetscalerVpx`	Creates new NS Vpx
`listElastistorVolume`	Lists the volumes of elastistor
`cloudianIsEnabled`	Checks if the Cloudian Connector is enabled
`listNuageVspGlobalDomainTemplate`	Lists Nuage VSP domain templates
`listHostHAResources`	Lists host HA resources
`enableHAForHost`	Enables HA for a host

Table 3 – continued from previous page

Name	Description
registerNetscalerServicePackage	Registers NCC Service Package
listHostHAProviders	Lists HA providers
listCaCertificate	Lists the CA public certificate(s) as support by the configured/provided C
migrateVPC	moves a vpc to another physical network
configureHAForHost	Configures HA for a host
listRegisteredServicePackages	lists registered service packages
disableHAForCluster	Disables HA cluster-wide
linkAccountToLdap	link a cloudstack account to a group or OU in ldap
associateNuageVspDomainTemplate	associate a vpc with a domain template
moveUser	Moves a user to another account
disableHAForHost	Disables HA for a host
deleteNetscalerControlCenter	Delete Netscaler Control Center
migrateNetwork	moves a network to another physical network
uploadTemplateDirectDownloadCertificate	Upload a certificate for HTTPS direct template download on KVM hosts
registerNetscalerControlCenter	Adds a netscaler control center device
createManagementNetworkIpRange	Creates a Management network IP range.
removeAnnotation	remove an annotation.
releasePodIpAddress	Releases a Pod IP back to the Pod

8.4.2 Parameters Changed API Commands

Name	Description
createPod	**Response:** *New Parameters:* • forsystemvms • vlanid
copyIso	**Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize
listHosts	**Response:** *New Parameters:* • annotation • hostha • lastannotated • username
updateStoragePool	**Response:** *New Parameters:* • allocatediops • provider

Continued on next page

Table 4 – continued from previous page

Name	Description
`rebootSystemVm`	**Response:** *New Parameters:* • `guestvlan` • `publicvlan`
`listNetworks`	**Response:** *New Parameters:* • `externalid`
`updateResourceLimit`	**Response:** *New Parameters:* • `resourcetypename`
`updateHost`	**Request:** *New Parameters:* • `annotation` (optional) **Response:** *New Parameters:* • `annotation` • `hostha` • `lastannotated` • `username`
`uploadVolume`	**Response:** *New Parameters:* • `clusterid` • `clustername` • `physicalsize` • `podid` • `podname` • `utilization` • `virtualsize`
`destroySystemVm`	**Response:** *New Parameters:* • `guestvlan` • `publicvlan`
`scaleSystemVm`	**Response:** *New Parameters:* • `guestvlan` • `publicvlan`
`listLdapConfigurations`	**Request:** *New Parameters:* • `domainid` (optional) **Response:** *New Parameters:* • `domainid`

Continued on next page

Table 4 – continued from previous page

Name	Description
listTemplates	**Request:** *New Parameters:* • parenttemplateid (optional) **Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize
createLoadBalancerRule	**Response:** *New Parameters:* • zonename
updateNetworkOffering	**Request:** *New Parameters:* • tags (optional)
stopSystemVm	**Response:** *New Parameters:* • guestvlan • publicvlan
createNetworkOffering	**Request:** *New Parameters:* • forvpc (optional)
listVolumesMetrics	**Request:** *New Parameters:* • clusterid (optional)
listSslCerts	**Response:** *New Parameters:* • name
listPods	**Response:** *New Parameters:* • forsystemvms • vlanid
listSnapshots	**Response:** *New Parameters:* • virtualsize
listConfigurations	**Request:** *New Parameters:* • domainid (optional)

Continued on next page

Table 4 – continued from previous page

Name	Description
listSystemVms	**Response:** *New Parameters:* • guestvlan • publicvlan
detachVolume	**Response:** *New Parameters:* • clusterid • clustername • physicalsize • podid • podname • utilization • virtualsize
changeServiceForSystemVm	**Response:** *New Parameters:* • guestvlan • publicvlan
createSnapshot	**Request:** *New Parameters:* • asyncbackup (optional) **Response:** *New Parameters:* • virtualsize
listNics	**Response:** *New Parameters:* • extradhcpoption
createSnapshotFromVMSnapshot	**Response:** *New Parameters:* • virtualsize
listStoragePools	**Response:** *New Parameters:* • allocatediops • provider
addNicToVirtualMachine	**Request:** *New Parameters:* • dhcpoptions (optional) • macaddress (optional)

Continued on next page

Table 4 – continued from previous page

Name	Description
listExternalLoadBalancers	**Response:** *New Parameters:* • annotation • hostha • lastannotated • username
updateIso	**Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize
prepareTemplate	**Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize
copyTemplate	**Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize
listNiciraNvpDeviceNetworks	**Response:** *New Parameters:* • externalid
resizeVolume	**Response:** *New Parameters:* • clusterid • clustername • physicalsize • podid • podname • utilization • virtualsize
updateTemplate	**Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize

Continued on next page

Table 4 – continued from previous page

Name	Description
`createVlanIpRange`	**Request:** *New Parameters:* • `forsystemvms` (optional) **Response:** *New Parameters:* • `forsystemvms`
`listPaloAltoFirewallNetworks`	**Response:** *New Parameters:* • `externalid`
`deleteLdapConfiguration`	**Request:** *New Parameters:* • `domainid` (optional) • `port` (optional) **Response:** *New Parameters:* • `domainid`
`updateVolume`	**Response:** *New Parameters:* • `clusterid` • `clustername` • `physicalsize` • `podid` • `podname` • `utilization` • `virtualsize`
`updateVirtualMachine`	**Request:** *New Parameters:* • `dhcpoptionsnetworklist` (optional)
`listDomains`	**Request:** *New Parameters:* • `details` (optional)
`updateNetwork`	**Response:** *New Parameters:* • `externalid`
`deleteTemplate`	**Request:** *New Parameters:* • `forced` (optional)

Continued on next page

Table 4 – continued from previous page

Name	Description
`createTemplate`	**Response:** *New Parameters:* • `childtemplates` • `directdownload` • `parenttemplateid` • `physicalsize`
`updatePortForwardingRule`	**Request:** *New Parameters:* • `privateendport` (optional)
`linkDomainToLdap`	**Request:** *New Parameters:* • `ldapdomain` (required) **Response:** *New Parameters:* • `ldapdomain`
`listSrxFirewallNetworks`	**Response:** *New Parameters:* • `externalid`
`prepareHostForMaintenance`	**Response:** *New Parameters:* • `annotation` • `hostha` • `lastannotated` • `username`
`registerIso`	**Request:** *New Parameters:* • `directdownload` (optional) **Response:** *New Parameters:* • `childtemplates` • `directdownload` • `parenttemplateid` • `physicalsize`
`deployVirtualMachine`	**Request:** *New Parameters:* • `datadiskofferinglist` (optional) • `dhcpoptionsnetworklist` (optional) • `macaddress` (optional)
`listVlanIpRanges`	**Response:** *New Parameters:* • `forsystemvms`

Continued on next page

Table 4 – continued from previous page

Name	Description
`cancelHostMaintenance`	**Response:** *New Parameters:* • `annotation` • `hostha` • `lastannotated` • `username`
`listVolumes`	**Request:** *New Parameters:* • `clusterid` (optional) **Response:** *New Parameters:* • `clusterid` • `clustername` • `physicalsize` • `podid` • `podname` • `utilization` • `virtualsize`
`lockUser`	**Response:** *New Parameters:* • `usersource`
`createNetwork`	**Request:** *New Parameters:* • `bypassvlanoverlapcheck` (optional) • `externalid` (optional) **Response:** *New Parameters:* • `externalid`
`updateUser`	**Response:** *New Parameters:* • `usersource`
`addHost`	**Response:** *New Parameters:* • `annotation` • `hostha` • `lastannotated` • `username`

Continued on next page

Table 4 – continued from previous page

Name	Description
`attachVolume`	**Response:** *New Parameters:* • `clusterid` • `clustername` • `physicalsize` • `podid` • `podname` • `utilization` • `virtualsize`
`listUsers`	**Response:** *New Parameters:* • `usersource`
`listResourceLimits`	**Request:** *New Parameters:* • `resourcetypename` (optional) **Response:** *New Parameters:* • `resourcetypename`
`disableUser`	**Response:** *New Parameters:* • `usersource`
`listIsos`	**Response:** *New Parameters:* • `childtemplates` • `directdownload` • `parenttemplateid` • `physicalsize`
`listNetscalerLoadBalancerNetworks`	**Response:** *New Parameters:* • `externalid`
`startSystemVm`	**Response:** *New Parameters:* • `guestvlan` • `publicvlan`
`migrateVolume`	**Response:** *New Parameters:* • `clusterid` • `clustername` • `physicalsize` • `podid` • `podname` • `utilization` • `virtualsize`

Continued on next page

Table 4 – continued from previous page

Name	Description
listEvents	**Request:** *New Parameters:* • startid (optional)
addLdapConfiguration	**Request:** *New Parameters:* • domainid (optional) **Response:** *New Parameters:* • domainid
updateConfiguration	**Request:** *New Parameters:* • domainid (optional)
dedicatePublicIpRange	**Response:** *New Parameters:* • forsystemvms
revertSnapshot	**Response:** *New Parameters:* • virtualsize
migrateSystemVm	**Response:** *New Parameters:* • guestvlan • publicvlan
updateResourceCount	**Response:** *New Parameters:* • resourcetypename
listBrocadeVcsDeviceNetworks	**Response:** *New Parameters:* • externalid
listUsageRecords	**Request:** *New Parameters:* • includetags (optional)
enableUser	**Response:** *New Parameters:* • usersource

Continued on next page

Table 4 – continued from previous page

Name	Description
registerTemplate	**Request:** *New Parameters:* • directdownload (optional) **Response:** *New Parameters:* • childtemplates • directdownload • parenttemplateid • physicalsize
createStoragePool	**Response:** *New Parameters:* • allocatediops • provider
findStoragePoolsForMigration	**Response:** *New Parameters:* • allocatediops • provider
createVolume	**Response:** *New Parameters:* • clusterid • clustername • physicalsize • podid • podname • utilization • virtualsize
listF5LoadBalancerNetworks	**Response:** *New Parameters:* • externalid
updatePod	**Response:** *New Parameters:* • forsystemvms • vlanid
enableStorageMaintenance	**Response:** *New Parameters:* • allocatediops • provider
createUser	**Response:** *New Parameters:* • usersource

Continued on next page

Table 4 – continued from previous page

Name	Description
updateRolePermission	**Request:** *New Parameters:* • permission (optional) • ruleid (optional) *Changed Parameters:* • ruleorder was 'required' and is now 'optional'
cancelStorageMaintenance	**Response:** *New Parameters:* • allocatediops • provider
updateLoadBalancerRule	**Request:** *New Parameters:* • protocol (optional) **Response:** *New Parameters:* • zonename
reconnectHost	**Response:** *New Parameters:* • annotation • hostha • lastannotated • username
getUser	**Response:** *New Parameters:* • usersource
listLoadBalancerRules	**Response:** *New Parameters:* • zonename
uploadSslCert	**Request:** *New Parameters:* • name (required) **Response:** *New Parameters:* • name
addBaremetalHost	**Response:** *New Parameters:* • annotation • hostha • lastannotated • username

Continued on next page

Table 4 – continued from previous page

Name	Description
`listCapacity`	**Response:** *New Parameters:* • `capacityallocated` • `name`

8.5 Known Issues in 4.11

Apache CloudStack uses Jira to track its issues. All new features and bugs for 4.11 have been tracked in Jira, and have a standard naming convention of "CLOUDSTACK-NNNN" where "NNNN" is the issue number.

For the list of known issues, see Known Issues in 4.11.

Bug ID	Description
CLOUDSTACK-8862	Issuing multiple attach-volume commands simultaneously can be problematic...
CLOUDSTACK-9734	VM Expunge Fails sometimes...
CLOUDSTACK-9712	Establishing Remote access VPN is failing due to mismatch of preshared secrets ...
CLOUDSTACK-9363	Can't start a Xen HVM vm when more than 2 volumes attached...
CLOUDSTACK-9692	Reset password service is not running on Redundant virtual routers....
CLOUDSTACK-9653	listCapacity API shows incorrect output when sortBy=usage option is added...
CLOUDSTACK-9675	Cloudstack Metrics: Miscellaneous bug fixes...
CLOUDSTACK-9727	Password reset discrepancy in RVR when one of the Router is not in Running state...
CLOUDSTACK-9726	state of the rvr dose not change to update failed when updating rvrs in sequence...
CLOUDSTACK-9668	disksizeallocated of PrimaryStorage is different from the total size of a volume...
CLOUDSTACK-9694	Unable to limit the Public IPs in VPC...
CLOUDSTACK-9716	baremetal:rvr:vm deployment gets stuck in starting state (waiting for notificati...
CLOUDSTACK-9701	When host is disabled/removed, capacity_state for local storage in op_host_capac...
CLOUDSTACK-9569	VR on shared network not starting on KVM...
CLOUDSTACK-8284	Primary_storage vlaue is not updating in resource_count table after VM deletion...
CLOUDSTACK-9697	Better error message user if tries to shrink the VM ROOT volume size...
CLOUDSTACK-9693	Cluster status of an unmanaged cluster still shows enabled...
CLOUDSTACK-9687	disksizeallocated of PrimaryStorage is different from the total size of a volume...
CLOUDSTACK-9560	Root volume of deleted VM left unremoved...
CLOUDSTACK-9559	Deleting zone without deleting the secondary storage under the zone should not b...
CLOUDSTACK-8896	Allocated percentage of storage can go beyond 100%...
CLOUDSTACK-9664	updateRole: type can not be changed...
CLOUDSTACK-9672	MySQL HA doesn't work...
CLOUDSTACK-9667	Enable resourcecount.check.interval by default...
CLOUDSTACK-9666	Add configuration validation for the config drive global settings...
CLOUDSTACK-9665	List hosts api dose not report correct cpu and memory usage...
CLOUDSTACK-9628	Fix Template Size in Swift as Secondary Storage...
CLOUDSTACK-9626	Instance fails to start after unsuccesful compute offering upgrade....
CLOUDSTACK-9647	NIC adapter type becomes e1000 , even after changing the global parameter "vmwar...
CLOUDSTACK-9500	VR configuration not clean properly after Public IP release...
CLOUDSTACK-9175	[VMware DRS] Adding new host to DRS cluster does not participate in load balanci...
CLOUDSTACK-9651	Fix Docker image build of simulator, marvin and management-server....
CLOUDSTACK-9589	vmName entries from host_details table for the VM's whose state is Expunging sho...
CLOUDSTACK-8849	Usage job should stop usage generation in case of any exception...
CLOUDSTACK-9558	Cleanup the snapshots on the primary storage of Xenserver after VM/Volume is exp...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-9637	Template create from snapshot does not populate vm_template_details...
CLOUDSTACK-9627	Template Doens't get sync when using Swift as Secondary Storage...
CLOUDSTACK-9546	X-Forwarded-For Headers Not Applied for HTTP Traffic in haproxy...
CLOUDSTACK-9586	When using local storage with Xenserver prepareTemplate does not work with multi...
CLOUDSTACK-9635	fix test_privategw_acl.py...
CLOUDSTACK-8939	VM Snapshot size with memory is not correctly calculated in cloud.usage_event (X...
CLOUDSTACK-9182	Some running VMs turned off on manual migration when auto migration failed while...
CLOUDSTACK-9638	Problems caused when inputting double-byte numbers for custom compute offerings...
CLOUDSTACK-9640	In KVM SSVM and CPVM may use the old cmdline data, if we fail to fetch the new c...
CLOUDSTACK-9641	In KVM SSVM and CPVM may use the old cmdline data, if we fail to fetch the new c...
CLOUDSTACK-9642	API documentation: getVirtualMachineUserData is in the wrong command category 'U...
CLOUDSTACK-9634	fix marvin test test_router_dhcp_opts failure...
CLOUDSTACK-9184	[VMware] vmware.ports.per.dvportgroup global setting is not useful from vCenter ...
CLOUDSTACK-9595	Transactions are not getting retried in case of database deadlock errors...
CLOUDSTACK-9538	Deleting Snapshot From Primary Storage Fails on RBD Storage if you already delet...
CLOUDSTACK-9317	Disabling static NAT on many IPs can leave wrong IPs on the router...
CLOUDSTACK-9593	User data check is inconsistent with python...
CLOUDSTACK-9598	wrong defaut gateway in guest VM with nics in isolated and a shared network...
CLOUDSTACK-9614	Attaching Volume to VM incorrectly checks resource limits...
CLOUDSTACK-9572	Snapshot on primary storage not cleaned up after Storage migration...
CLOUDSTACK-9280	System VM volumes cannot be deleted when there are no system VMs...
CLOUDSTACK-8781	Superfluous field during VPC creation...
CLOUDSTACK-9498	VR CsFile search utility methods fail when search string has char...
CLOUDSTACK-9503	The router script times out resulting in failure of deployment...
CLOUDSTACK-9356	VPC add VPN User fails same error as CLOUDSTACK-8927...
CLOUDSTACK-9017	VPC VR DHCP broken for multihomed guest VMs...
CLOUDSTACK-9585	UI doesn't give an option to select the xentools version for non ROOT users...
CLOUDSTACK-9417	Usage module refactoring...
CLOUDSTACK-9555	when a template is deleted and then copied over again , it is still marked as Re...
CLOUDSTACK-9592	Empty responses from site to site connection status are not handled propertly...
CLOUDSTACK-9601	Database upgrade algorithm is incorrect...
CLOUDSTACK-9596	migrateVirtualMachine API does not respect affinity group assignment...
CLOUDSTACK-9578	6 out of 12 internal Lb rules were added to internal LB with same source ip duri...
CLOUDSTACK-9557	Deploy from VMsnapshot fails with exception if source template is removed or mad...
CLOUDSTACK-9370	Failed to create VPC: Unable to start VPC VR (VM DomainRouter) due to error in ...
CLOUDSTACK-9591	VMware dvSwitch Requires a Dummy, Standard vSwitch...
CLOUDSTACK-9405	listDomains API call takes an extremely long time to respond...
CLOUDSTACK-8288	Deleting Instance deletes unrelated snapshots...
CLOUDSTACK-9577	NPE whle deleting internal LB rules concurrently...
CLOUDSTACK-9579	Internal lb vm display page stuck in loading not showing any vms...
CLOUDSTACK-9580	Unexpected exception while deleting vms concurrently...
CLOUDSTACK-9581	Error in logs while concurrently creating 100 vms...
CLOUDSTACK-9582	Null pointer exceptions while deleting network concurrently...
CLOUDSTACK-9576	Nuage VSP Plugin : NPE while creating vpctier with wrong domain template name...
CLOUDSTACK-9552	KVM Security Groups do not allow DNS over TCP egress...
CLOUDSTACK-9575	ACS 4.9 + VMware/ESXi 6.0: Sometimes VRs are failing to fully come up into runni...
CLOUDSTACK-9226	Wrong number of sockets reported...
CLOUDSTACK-9553	Usage event is not getting recorded for snapshots in a specific scenario...
CLOUDSTACK-9554	Juniper Contrail plug-in is publishing events to wrong message bus...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-9551	Pull KVM agent's tmp folder usage within its own folder structure...
CLOUDSTACK-9571	Management server should fence itself when there are recoverable DB errors OR wh...
CLOUDSTACK-7827	storage migration timeout, loss of data...
CLOUDSTACK-9514	MarvinTests: some host credentials are hardcoded, make them to be picked up from...
CLOUDSTACK-9533	gateway of public IP is not handled correctly when parsing the cmd_line.json to ...
CLOUDSTACK-9529	Marvin Tests do not clean up properly...
CLOUDSTACK-9357	DHCP DNS option is incorrect for Redundant Router config...
CLOUDSTACK-9547	ACS 4.9 + VMware: Unable to remove one of the NICs of a multi-nic guest VM...
CLOUDSTACK-9474	When attaching a pool to an instance the askingSize parameter of pool checker is...
CLOUDSTACK-9542	listNics API does not return data as per API documentation...
CLOUDSTACK-9528	SSVM Downloads (built-in) template multiple times...
CLOUDSTACK-9541	redundant VPC VR: issues when master and backup switch happens on failover...
CLOUDSTACK-9540	createPrivateGateway create private network does not create proper VLAN network ...
CLOUDSTACK-9413	VmIpFetchTask NullPointerException...
CLOUDSTACK-9537	cloudstack can only get network data of eth0 in the xenserver host ...
CLOUDSTACK-9536	PVLAN: DhcpPvlanRules command bieng sent before processing finalize start comman...
CLOUDSTACK-9518	test_01_test_vm_volume_snapshot Smoke Test Fails...
CLOUDSTACK-9521	Multiple Failures in the test_vpc_vpn Smoke Test Suite...
CLOUDSTACK-9520	test_01_primary_storage_iscsi Smoke Test Fails...
CLOUDSTACK-9519	test_01_RVR_Network_FW_PF_SSH_default_routes_egress_true Smoke Test Failure...
CLOUDSTACK-9512	listTemplates ids returns all templates instead of the requested ones...
CLOUDSTACK-9508	Change Script and SshHelper to use Duration instead of long timeout....
CLOUDSTACK-9318	test_privategw_acl is failing intermittently....
CLOUDSTACK-9475	Attaching to PVLAN on VMware dvSwtich fails on VR reboot...
CLOUDSTACK-9490	Cant shrink data volume...
CLOUDSTACK-9483	In developers.html there is a html </br> tag displayed as a plain text ...
CLOUDSTACK-9371	Regular user cannot resize volume...
CLOUDSTACK-8398	Changing compute offering checks account quota instead of project quota...
CLOUDSTACK-9473	Storage pool checker is ignored on resize and migrate volume....
CLOUDSTACK-9472	Taking snapshot on a large VMware volume times out...
CLOUDSTACK-9471	Cross cluster migration did not kick in when HA is enabled and Host goes down...
CLOUDSTACK-8937	Xenserver - VM migration with storage fails in a clustered management server set...
CLOUDSTACK-9144	VMware in Basic Zone: VR never leaves the "Starting" state...
CLOUDSTACK-9454	cloudstack-agent logs rotation outdated...
CLOUDSTACK-9407	vm_network_map table doesnt get cleaned up properly...
CLOUDSTACK-9225	Isolation in Advanced Zone using PVLANs ...
CLOUDSTACK-9431	Network usage stats from VR in VPC are wrong after upgrading to ACS 4.7...
CLOUDSTACK-9433	Change of VM compute offering with additional storage tags not allowed...
CLOUDSTACK-9439	Domain admins can and must create service and disk offerings withouts storage an...
CLOUDSTACK-9434	NPE on attempting account/domain cleanup automation...
CLOUDSTACK-9341	Cannot upload volume when using Swift as secondary storage...
CLOUDSTACK-9206	Input issue on change service offering in Custom...
CLOUDSTACK-9432	Dedicate Cluster to Domain always creates an affinity group owned by the root do...
CLOUDSTACK-9367	Unable to start a HVM VM with more than 2 volumes attached using XenServer 6.5 ...
CLOUDSTACK-9227	service cloudstack-management stop returns [failed] due to log4j:WARN No appende...
CLOUDSTACK-9427	Sudo in wrong place when adding APT repository key...
CLOUDSTACK-9425	Storage statistics shown on CloudStack Primary storage is different from the sta...
CLOUDSTACK-9426	CloudStack does not re-scan for new LUNs for an iSCSI based storage on KVM host...
CLOUDSTACK-9385	Password Server is not running on RvR...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-9421	Cannot add Instance...
CLOUDSTACK-9419	network_domain is a optional param while creating network, still createIpAlias.s...
CLOUDSTACK-9412	NullPointerException in CapacityManagerImpl...
CLOUDSTACK-8921	snapshot_store_ref table should store actual size of back snapshot in secondary ...
CLOUDSTACK-9411	Resize Root Volume UI Element Not Visible By Domain Admins or Users...
CLOUDSTACK-8922	Unable to delete IP tag...
CLOUDSTACK-9253	docker cloudstack simulator "ImportError: No module named marvin" when try to cr...
CLOUDSTACK-8944	Template download possible from new secondary storages before the download is 10...
CLOUDSTACK-9394	HttpTemplateDownloader Causes Hanging Connections...
CLOUDSTACK-9393	Wrong information returned for CheckS2SVpnConnectionsCommand when more than one ...
CLOUDSTACK-9392	Networks with redundant network offerings can be implemented with standalone vir...
CLOUDSTACK-9390	Dettaching data volume from a running vm created with root and data disk fails...
CLOUDSTACK-9384	AutoScalling without netscaller problem...
CLOUDSTACK-9381	updateVmNicIp doesn't update the gateway on NIC if the new IP is from a differen...
CLOUDSTACK-8237	add nic with instance throw java.lang.NullPointerException ...
CLOUDSTACK-8584	Management Server does not start - "cluster node IP should be valid local addres...
CLOUDSTACK-9338	listAccount returns 0 for cputotal and memorytotal if VMs are using a ComputeOff...
CLOUDSTACK-9112	deployVM thread is holding the global lock on network longer and cause delays an...
CLOUDSTACK-8855	Improve Error Message for Host Alert State...
CLOUDSTACK-9372	Cannot create PPTP VPN from guest instance to endpoint outside of the cloud...
CLOUDSTACK-9360	Set guest password not working with redundant routers...
CLOUDSTACK-9224	XenServer local storage added multiple times...
CLOUDSTACK-9346	Password server on VR is not working correctly when using a custom network offer...
CLOUDSTACK-9189	rVPC ACL doesn't recover after cleaning up through the NetworkGarbageCollector...
CLOUDSTACK-9094	Multiple threads are being used to collect the stats from the same VR...
CLOUDSTACK-8775	[HyperV]NPE while attaching Local storage volume to instance whose root volume i...
CLOUDSTACK-8787	Network Update from Standalone VR offering to RVR offering is failing with Runti...
CLOUDSTACK-8877	Show error msg on VPN user add failure....
CLOUDSTACK-8912	listGuestOsMapping doesn't list by id or ostypeid....
CLOUDSTACK-8918	[Install] Db Error after install - Unknown column 'iso_id1' in 'field list'...
CLOUDSTACK-8929	The list of VMs that can be assigned to a load balancer rule is not updated afte...
CLOUDSTACK-9035	[rVR] Password file is stored only with Master when we Reset Password on the VM...
CLOUDSTACK-9176	VMware: Shared datastore is accidentally picked up as a local datastore...
CLOUDSTACK-9330	Cloudstack creates a malformed meta-data file for baremetal instances...
CLOUDSTACK-9329	cloud-set-guest-password doesn't work on CentOS 7...
CLOUDSTACK-9079	ipsec service is not running after restarting virtual router...
CLOUDSTACK-9316	Problem to install in CentOS7 4.8...
CLOUDSTACK-9311	User cant resize VM root disk for XenServer...
CLOUDSTACK-9312	Duplicate instance IPs addresses...
CLOUDSTACK-9310	vpn user creation throwing error , but showing entry for the same user in VR con...
CLOUDSTACK-9309	Adding primary storage pool (basic rbd/DefaultPrimary) doesn't work if the rados...
CLOUDSTACK-9307	You can't mix two different linux distributions in a (KVM) Cluster...
CLOUDSTACK-9303	Cloudstack can't connect to CEPH with "/" in the user pw...
CLOUDSTACK-8845	list snapshot without id is failing with Unable to determine the storage pool of...
CLOUDSTACK-8977	cloudstack UI creates a session for users not yet logged in...
CLOUDSTACK-9295	EGRESS left on ACCEPT on isolated network...
CLOUDSTACK-9292	Failed to create snapshot with Swift on KVM...
CLOUDSTACK-9286	Delete Domain not working: Failed to clean up domain resources and sub domains, ...
CLOUDSTACK-9284	CloudStack usage service tries to get access to "cloud.event_usage" table only v...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-7857	CitrixResourceBase wrongly calculates total memory on hosts with a lot of memory...
CLOUDSTACK-9258	listDomains API fails with NPE when getVolumeTotal is null...
CLOUDSTACK-9247	Templates go into "Not Ready" state after restarting manangement server with Swi...
CLOUDSTACK-9232	Usage data does not reflect changes of VM parameters...
CLOUDSTACK-9212	Cannot Connect to VPN with Public IP on Windows 7 L2TP IPSEC VPN Client...
CLOUDSTACK-9243	createVlanIpRange API unusable because forced to used DB IDs...
CLOUDSTACK-9234	Problem increasing value of vm.password.length global parameter...
CLOUDSTACK-8966	listCapacity produces wrong result for CAPACITY_TYPE_MEMORY and CAPACITY_TYPE_CP...
CLOUDSTACK-6448	VPC router won't be created when a private gateway is defined. ...
CLOUDSTACK-9193	Once password has been fetched, the state does not get updated to "saved_passwor...
CLOUDSTACK-9191	ACS 4.6 Custom Offer Signature mismatch "ERROR : "unable to verify user credenti...
CLOUDSTACK-9190	ACs is falling to identify the version of pure Xen hypervisor + XAPI hosts....
CLOUDSTACK-8806	Powered off VM's not showing up in WebUI...
CLOUDSTACK-9170	Register template in UI does not show zones in dropdown listbox...
CLOUDSTACK-9173	new Quota plugins: CPU Used column is CPU Free column...
CLOUDSTACK-9171	Templates registered with CrossZones have no zone name listed...
CLOUDSTACK-9169	createNetwork API call takes a long time when ispersistent=True...
CLOUDSTACK-9141	Userdata is not validated for valid base64...
CLOUDSTACK-9167	Restore VM - Missing action events for started and completed states...
CLOUDSTACK-9090	Cannot delete zone if it was used and not all elements were cleanly removed...
CLOUDSTACK-8936	wrong values from network.throttling.rate / vm.network.throttling.rate...
CLOUDSTACK-9096	Deleted projects cannot be billed...
CLOUDSTACK-9061	cannot deploy Instance when using Swift as Secondary Storage...
CLOUDSTACK-9089	Static route added to VPC Private Gateway doesn't become active...
CLOUDSTACK-9085	Creation of a instance on a Guest Network with Secondary VLAN fail...
CLOUDSTACK-8807	Cloudstack WebUI sometimes bothers about the selected project, sometimes not...
CLOUDSTACK-9036	IPV6 CIDR not recognized (Parser BUG)...
CLOUDSTACK-9057	upgrade to 4.6 requires 4.5 templates...
CLOUDSTACK-9059	Snapshot on S3 fails when delta is zero...
CLOUDSTACK-9060	Create volume / template from S3 snapshot fails...
CLOUDSTACK-7375	[UI] RBD not available under list of protocols for primary storage during zone c...
CLOUDSTACK-9028	GloboDNS doen´t work with "Shared Networks"...
CLOUDSTACK-8902	Restart Network fails in EIP/ELB zone...
CLOUDSTACK-8994	Activity of the password server isn't logged....
CLOUDSTACK-8889	Primary Storage count for an account does not decrease when a Data Disk is delet...
CLOUDSTACK-8982	Disk Offering properties do no show the domain which are included in...
CLOUDSTACK-8724	Multiple IP's on management server break patchviasocket.pl...
CLOUDSTACK-8945	rp_filter=1 not set on VPC private gateway initially, but is set after restart o...
CLOUDSTACK-8942	snapshot of root drives failing...
CLOUDSTACK-8938	Assigning portforward in Isolated "Offering for Isolated networks with Source Na...
CLOUDSTACK-8914	cannot delete pod, NPE...
CLOUDSTACK-8909	Web Console not working with Hyper-V Windows Server 2012 R2...
CLOUDSTACK-8771	[Automation]Volume migration between pools times out in ACS, but the migration c...
CLOUDSTACK-8782	If pagesize is greater than default.page.size in API call, and default.page.size...
CLOUDSTACK-8846	Performance issue in GUI - API command listVirtualMachines ...
CLOUDSTACK-8839	close concurrent ip disable static nat commands for virtual router will cause so...
CLOUDSTACK-8831	Powered off VM's are not removed from ESXi Host when putting the Host in Mainten...
CLOUDSTACK-7853	Hosts that are temporary Disconnected and get behind on ping (PingTimeout) turn ...
CLOUDSTACK-8747	The agent doesn't reconnect if there are stopped VMs...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-8809	Secondary Storage does not clean-up after time-out...
CLOUDSTACK-8796	the api calll linkdomaintoldap should fail if admin is given and an account isnt...
CLOUDSTACK-7591	Dynamic scaling doesn't work in CloudStack 4.4 with vmware...
CLOUDSTACK-8437	Automation: test_04_create_multiple_networks_with_lb_1_network_offering - Fails...
CLOUDSTACK-8732	Unable to resize RBD volume: "Cannot determine resize type from pool type RBD"...
CLOUDSTACK-8631	[Automation]fixing test/integration/component/test_ss_max_limits.py...
CLOUDSTACK-8142	[Hyper-V] While creating system vms attach systemvm.iso directly from sec storag...
CLOUDSTACK-8448	Attach volume - throws an exception, preferably should give a proper error on UI...
CLOUDSTACK-8770	[HyperV]Proper Message should be displayed when snapshot fails on Hyper-V...
CLOUDSTACK-8768	[HyperV]Migrating volume from cluster wide storage to Zone wide storage or vicev...
CLOUDSTACK-7839	Unable to live migrate an instance to another host in a cluster from which the t...
CLOUDSTACK-7364	NetScaler won't create the Public VLAN and Bind the IP to it...
CLOUDSTACK-7618	Baremetal - AddHost() API docs should include parameters - cpunumber,cpuspeed,me...
CLOUDSTACK-8389	Volume to Template Conversion Broken...
CLOUDSTACK-8442	[VMWARE] VM Cannot be powered on after restoreVirtualMachine ...
CLOUDSTACK-8699	Extra acquired public ip is assigned to wrong eth device...
CLOUDSTACK-8694	monitorServices.py is not running as a cron job in VR...
CLOUDSTACK-8691	deployVirtualMachine should not error when userdata is provided if at least one ...
CLOUDSTACK-8328	NPE while deleteing instance which has custom compute offering...
CLOUDSTACK-8695	Dashboard Alerts for VR Service failures does not contain the service's name...
CLOUDSTACK-8684	Upgrade from 4.3.1 to 4.5.1 does not update resource for existing XenServer 6.0....
CLOUDSTACK-8680	problem parsing RabbitMQ events...
CLOUDSTACK-8679	Changes to RabbitMQ events notification framework not documented anywhere...
CLOUDSTACK-8674	Custom ISO with reboot –eject in kickstart does not get detached at reboot...
CLOUDSTACK-8670	Delay in VM's console...
CLOUDSTACK-8657	java.awt.HeadlessException exception in console proxy on mouse clicks in XenServ...
CLOUDSTACK-8639	fixing calculation mistakes in component/test_ss_domain_limits.py...
CLOUDSTACK-8588	Remove redundant skip test for LXC ...
CLOUDSTACK-8556	Unable to delete attached volume in cleanup...
CLOUDSTACK-8549	Update assert statements in testpath_disable_enable_zone.py testpath ...
CLOUDSTACK-8626	[Automation]fixing test/integration/component/test_ps_max_limits.py for lxc hyp...
CLOUDSTACK-8627	Unable to remove IP from NIC....
CLOUDSTACK-8620	[Automation-lxc]skip test cases if rbd storage is not available in lxc setup ...
CLOUDSTACK-8158	After the host reboots, the system will run out vm management IP, no matter how ...
CLOUDSTACK-8583	[Automation]fixing issue related to script test/integration/component/test_stop...
CLOUDSTACK-8619	Adding secondary IP address results in error...
CLOUDSTACK-8618	Name or displaytext can not be same across different templates...
CLOUDSTACK-8614	Usage records have no valid records for migrated volumes...
CLOUDSTACK-8577	[Automation] fixing script test/integration/component/maint/testpath_disable_en...
CLOUDSTACK-8587	Storage migration issue on secondary storage...
CLOUDSTACK-8578	listVirtualMachines does not return deleted machines when zone is specified...
CLOUDSTACK-8574	Skip testcases including data disk creation for LXC if storagePool type is not R...
CLOUDSTACK-8576	Skip tests as snapshots and template are not supported on LXc...
CLOUDSTACK-8572	Unable to deploy VM as no storage pool found in UP state in setup...
CLOUDSTACK-8555	Skip testcase for HyperV as it doesn't support volume resize operationa...
CLOUDSTACK-8201	KVM Snapshot to Template to New Instance is not working...
CLOUDSTACK-8148	dvSwitch Broken with java.lang.NumberFormatException...
CLOUDSTACK-8558	KVM snapshots are failing at Ubuntu 14.04 LTS...
CLOUDSTACK-8557	Issue while starting Clound-Manager...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-8553	Unable to launch VM from template because of permission issue...
CLOUDSTACK-8550	Attempt to delete already deleted VM...
CLOUDSTACK-8547	Modify hypervisor check in testpath_snapshot_hardning.py testpath...
CLOUDSTACK-8552	Update test_concurrent_snapshots_limits.py asesrt statement...
CLOUDSTACK-8544	IP Stuck in Releasing State Prevents VM Create...
CLOUDSTACK-8532	Modification in setupClass to skip testcases rather than throwing exception...
CLOUDSTACK-8533	Local variable accessed as a class variable...
CLOUDSTACK-8354	[VMware] restoreVirtualMachine should forcefully power off VM...
CLOUDSTACK-8519	SystemVMs do not connect to MS running on Java 8...
CLOUDSTACK-8451	Static Nat show wrong remote IP in VM behind VPC...
CLOUDSTACK-8470	Available Primary Storage Capacity Displayed Incorrectly after Upgrade to ACS 4....
CLOUDSTACK-7907	UI heavily broken...
CLOUDSTACK-8469	wrong global config mount.parent - /var/lib/cloud/mnt ...
CLOUDSTACK-8446	VM reboot operation should make sure there's a VR running...
CLOUDSTACK-8436	Computing offering with High avaliability does not work properly....
CLOUDSTACK-8435	When the ssvm agent restarts, every template generated from a VM snapshot disapp...
CLOUDSTACK-8434	tag filtering hanging on returning values for listVirtualMachines...
CLOUDSTACK-8408	unused i18n keys...
CLOUDSTACK-8173	listCapacity api call returns less response tags than expected...
CLOUDSTACK-8371	Unable to Delete VPC After configuring site-to-site VPN...
CLOUDSTACK-8370	volume download link will not be deleted...
CLOUDSTACK-8358	Cloudstack 4.4.2 Error adding devcloud host IOException scp error: Invalid locat...
CLOUDSTACK-8281	VPN Gateway don't create when create Site-to-Site VPN...
CLOUDSTACK-8297	vnc listen address...
CLOUDSTACK-8228	Allow adding hosts from different subnets in same POD...
CLOUDSTACK-8260	listLBStickinessPolicies with lbruleid as argument gives empty return...
CLOUDSTACK-8242	Cloudstack install Hosts for vmware...
CLOUDSTACK-7449	"CloudRuntimeException: Can not see storage pool" after trying to add a new host...
CLOUDSTACK-8202	Templates /IOS items order list is not persistent...
CLOUDSTACK-8199	Incorrect size when volumes and templates created from image snapshots...
CLOUDSTACK-8189	security group can't enable...
CLOUDSTACK-7640	Failed to delete template that failed to download...
CLOUDSTACK-8185	GUI and failed async commands issue...
CLOUDSTACK-7365	Upgrading without proper systemvm template corrupt cloudstack management server...
CLOUDSTACK-8092	Unable to start instance due to failed to configure ip alias on the router as a ...
CLOUDSTACK-8073	listNetworkACLItem does not return cidrs...
CLOUDSTACK-8004	Xenserver Thin Provisioning...
CLOUDSTACK-7789	I was updated from version 4.4.0 of Apache CloudStack to 4.4.1. It does not work...
CLOUDSTACK-7988	Template status is empty while the template is creating....
CLOUDSTACK-7936	System VM's are getting stuck in starting mode after Hypervisor reboot...
CLOUDSTACK-7858	Implement separate network throttling rate on VR's Public NIC...
CLOUDSTACK-7342	Fail to delete template while using Swift as Secondary Storage...
CLOUDSTACK-7782	The 4.4.1 web UI is missing "Acquire new IP address" buton in NIC section...
CLOUDSTACK-7819	Cannot add tags to project...
CLOUDSTACK-7813	Management server is stuck after upgrade from 4.4.0 to 4.4.1...
CLOUDSTACK-7751	Autoscaling without netscaler...
CLOUDSTACK-7750	Xen server can not mount secondary CIFS storage...
CLOUDSTACK-7578	XenServerInvestigator should do better investigation in case of OVS or other net...
CLOUDSTACK-7406	Templates using Swift provider reports physical size, and not the virtual size i...

Continued on next page

Table 5 – continued from previous page

Bug ID	Description
CLOUDSTACK-7443	Cannot launch SSVMs when using Swift as Secondary Storage…

CHAPTER 9

Indices and Tables

- genindex
- search