Apache CloudStack 4.2.0

CloudStack Installation Guide

Edition 1

cloudstack™
open source cloud computing

Apache CloudStack

Legal Notice

Abstract

Installation Guide for CloudStack.

Chapter 1. Concepts

1.1. What Is CloudStack?

CloudStack is an open source software platform that pools computing resources to build public, private, and hybrid Infrastructure as a Service (IaaS) clouds. CloudStack manages the network, storage, and compute nodes that make up a cloud infrastructure. Use CloudStack to deploy, manage, and configure cloud computing environments.

Typical users are service providers and enterprises. With CloudStack, you can:

- Set up an on-demand, elastic cloud computing service. Service providers can sell self service virtual machine instances, storage volumes, and networking configurations over the Internet.
- Set up an on-premise private cloud for use by employees. Rather than managing virtual machines in the same way as physical machines, with CloudStack an enterprise can offer self-service virtual machines to users without involving IT departments.

1.2. What Can CloudStack Do?

Multiple Hypervisor Support

CloudStack works with a variety of hypervisors, and a single cloud deployment can contain multiple hypervisor implementations. The current release of CloudStack supports pre-packaged enterprise solutions like Citrix XenServer and VMware vSphere, as well as KVM or Xen running on Ubuntu or CentOS.

Massively Scalable Infrastructure Management

CloudStack can manage tens of thousands of servers installed in multiple geographically distributed datacenters. The centralized management server scales linearly, eliminating the need for intermediate cluster-level management servers. No single component failure can cause a cloud-wide outage. Periodic maintenance of the management server can be performed without affecting the functioning of virtual machines running in the cloud.

Automatic Configuration Management

CloudStack automatically configures each guest virtual machine’s networking and storage settings.

CloudStack internally manages a pool of virtual appliances to support the cloud itself. These appliances offer services such as firewalling, routing, DHCP, VPN access, console proxy, storage access, and storage replication. The extensive use of virtual appliances simplifies the installation, configuration, and ongoing management of a cloud deployment.

Graphical User Interface

CloudStack offers an administrator's 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 and Extensibility

CloudStack provides an API that gives programmatic access to all the management features available in the UI. The API is maintained and documented. This API enables the creation of command line tools and new user interfaces to suit particular needs. See the Developer’s Guide and API Reference, both available at Apache CloudStack Guides and Apache CloudStack API Reference respectively.

The CloudStack pluggable allocation architecture allows the creation of new types of allocators for the selection of storage and Hosts. See the Allocator Implementation Guide (http://docs.cloudstack.org/CloudStack_Documentation/Allocator_Implementation_Guide).

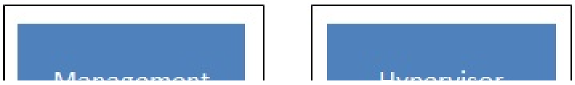
High Availability

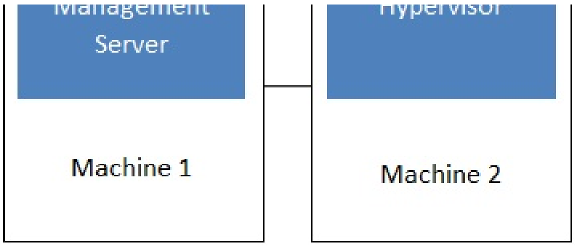
CloudStack has a number of features to increase the availability of the system. The Management Server itself may be deployed in a multi-node installation where the servers are load balanced. MySQL may be configured to use replication to provide for a manual failover in the event of database loss. For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

1.3. Deployment Architecture Overview

A CloudStack installation consists of two parts: the Management Server and the cloud infrastructure that it manages. When you set up and manage a CloudStack cloud, you provision resources such as hosts, storage devices, and IP addresses into the Management Server, and the Management Server manages those resources.

The minimum production 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 advanced networking setups. For information about deployment options, see the "Choosing a Deployment Architecture" section of the CloudStack Installation Guide.

1.3.1. Management Server Overview

The Management Server is the CloudStack software that manages cloud resources. By interacting with the Management Server through its UI or API, you can configure and manage your cloud infrastructure.

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

The machine must meet the system requirements described in System Requirements.

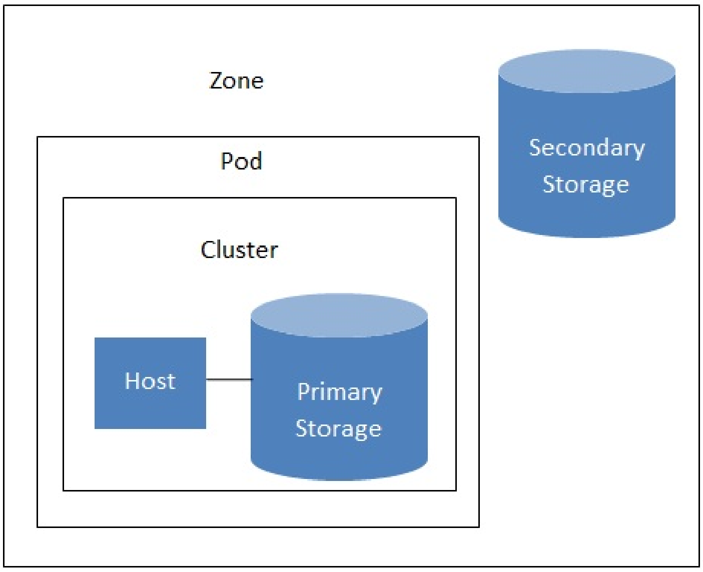
The Management Server:

- Provides the web user interface for the administrator and a reference user interface for end users.
- Provides the APIs for CloudStack.
- Manages the assignment of guest VMs to particular hosts.
- Manages the assignment of public and private IP addresses to particular accounts.
- Manages the allocation of storage to guests as virtual disks.
- Manages snapshots, templates, and ISO images, possibly replicating them across data centers.
- Provides a single point of configuration for the cloud.

1.3.2. Cloud Infrastructure Overview

The Management Server manages one or more zones (typically, datacenters) containing host computers where guest virtual machines will run. The cloud infrastructure is organized as follows:

- Zone: Typically, a zone is equivalent to a single datacenter. A zone consists of one or more pods and secondary storage.
- Pod: A pod is usually one rack of hardware that includes a layer-2 switch and one or more clusters.
- Cluster: A cluster consists of one or more hosts and primary storage.
- Host: A single compute node within a cluster. The hosts are where the actual cloud services run in the form of guest virtual machines.
- Primary storage is associated with a cluster, and it stores the disk volumes for all the VMs running on hosts in that cluster.
- Secondary storage is associated with a zone, and it stores templates, ISO images, and disk volume snapshots.



Nested organization of a zone

More Information

For more information, see documentation on cloud infrastructure concepts.

1.3.3. Networking Overview

CloudStack offers two types of networking scenario:

- 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.

For more details, see Network Setup.

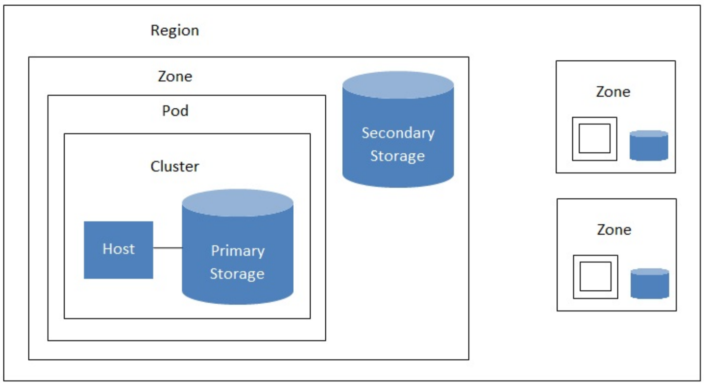
Chapter 2. Cloud Infrastructure Concepts

2.1. 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.



A region with multiple zones

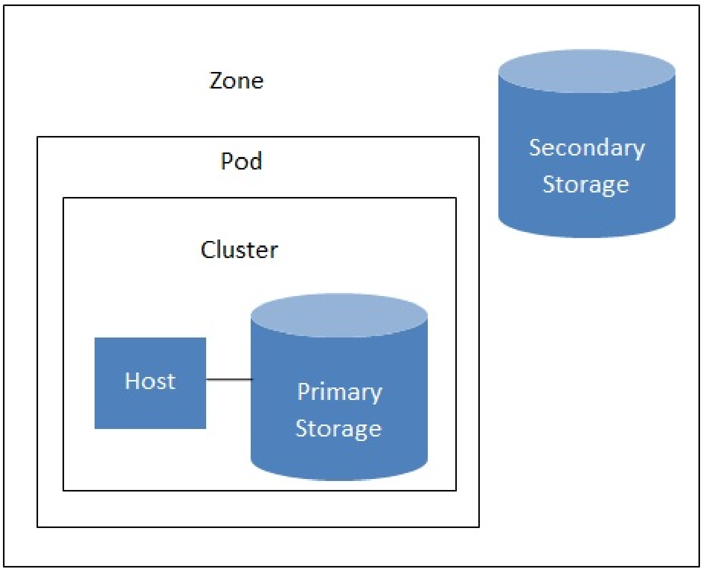
Regions are visible to the end user. When a user starts a guest VM on a particular CloudStack Management Server, the user is implicitly selecting that region for their guest. Users might also be required to copy their private templates to additional regions to enable creation of guest VMs using their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

2.2. 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.



Nested organization of a 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.

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.

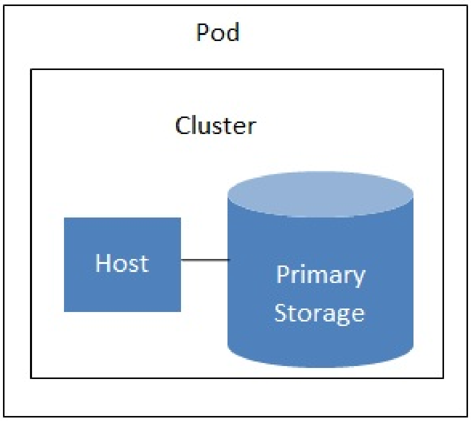
In order to support zone-wide functions for VMware, CloudStack is aware of VMware Datacenters and can map each Datacenter to a CloudStack zone. To enable features like storage live migration and zone-wide primary storage for VMware hosts, CloudStack has to make sure that a zone contains only a single 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.

2.3. About Pods

A pod often represents a single rack. Hosts in the same pod are in the same subnet. A pod is the second-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.



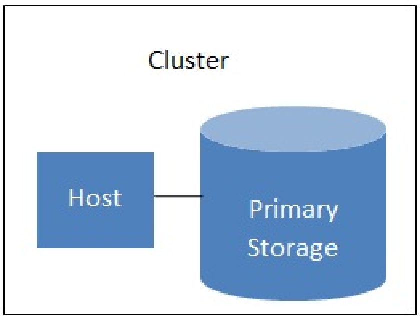
A simple pod

2.4. 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 third-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.



A simple cluster

CloudStack allows multiple clusters in a cloud deployment.

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.

2.5. About Hosts

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

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

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

2.6. About Primary Storage

Primary storage is associated with a cluster and/or a zone. It stores the disk volumes for all of the VMs running on hosts in that cluster. You can add multiple primary storage servers to a cluster or a zone (at least one is required at the cluster level). Primary storage is typically located close to the hosts for increased performance. CloudStack manages the allocation of guest virtual disks to particular primary storage devices.

Primary storage uses the concept of a storage tag. A storage tag is a label that is used to identify the primary storage. Each primary storage can be associated with zero, one, or more storage tags. When a VM is spun up or a data disk attached to a VM for the first time, these tags, if supplied, are used to determine which primary storage can support the VM or data disk (ex. say you need to guarantee a certain number of IOPS to a particular volume).

Primary storage can be either static or dynamic. Static primary storage is what CloudStack has traditionally supported. In this model, the administrator must present CloudStack with a certain amount of preallocated storage (ex. a volume from a SAN) and CloudStack can place many of its volumes on this storage. In the newer, dynamic model, the administrator can present CloudStack with a storage system itself (ex. a SAN). CloudStack, working in concert with a plug-in developed for that storage system, can dynamically create volumes on the storage system. A valuable use for this ability is Quality of Service (QoS). If a volume created in CloudStack can be backed by a dedicated volume on a SAN (i.e. a one-to-one mapping between a SAN volume and a CloudStack volume) and the SAN provides QoS, then CloudStack can provide QoS.

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 to Add Secondary Storage.

2.7. 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.

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.

2.8. 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:

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

2.8.1. 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.

2.8.2. 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.

2.8.3. 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.

2.8.4. 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.

2.8.5. 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.

2.8.6. 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.

Chapter 3. Building from Source

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 4.0.x series is different from the 4.1.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. Getting 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.

3.2. Verifying the downloaded release

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

3.2.1. 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:

```
# gpg --import KEYS
```

3.2.2. GPG

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

```
$ gpg --verify apache-cloudstack-4.0.0-incubating-src.tar.bz2.asc
```

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

3.2.3. 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-4.0.0-incubating-src.tar.bz2 | diff - apache-
cloudstack-4.0.0-incubating-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.

3.2.4. 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-4.0.0-incubating-src.tar.bz2 | diff - apache-
cloudstack-4.0.0-incubating-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.

3.3. 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 (OpenJDK 1.6 or Java 7/OpenJDK 1.7)
3. Apache Web Services Common Utilities (ws-commons-util)
4. MySQL
5. MySQLdb (provides Python database API)
6. Tomcat 6 (not 6.0.35)
7. genisoimage

7. genisoimage
8. rpmbuild or dpkg-dev

3.4. Extracting source

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

```
$ tar -jxvf apache-cloudstack-4.1.0.src.tar.bz2
```

You can now move into the directory:

```
$ cd ./apache-cloudstack-4.1.0-src
```

3.5. 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, which is not currently available in 12.04.1 LTS. So, you'll also need to add a PPA repository that includes Maven 3. After running the command **add-apt-repository**, you will be prompted to continue and a GPG key will be added.

```
$ sudo apt-get update
$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:natecarlson/maven3
$ sudo apt-get update
$ sudo apt-get install ant debhelper openjdk-6-jdk tomcat6 libws-commons-util-java
genisoimage python-mysqldb libcommons-codec-java libcommons-httpclient-java liblog4j1.2-
java maven3
```

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:

```
$ mvn3 -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-4.2.0.amd64.deb
cloudstack-management-4.2.0.amd64.deb
cloudstack-agent-4.2.0.amd64.deb
cloudstack-usage-4.2.0.amd64.deb
cloudstack-awsapi-4.2.0.amd64.deb
cloudstack-cli-4.2.0.amd64.deb
cloudstack-docs-4.2.0.amd64.deb
```

3.5.1. 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
sudo cd /var/www/cloudstack/repo/binary
sudo dpkg-scanpackages . /dev/null | tee Packages | gzip -9 > Packages.gz
```

Note: Override Files

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.)

3.5.2. 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 ./
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.

3.6. Building RPMs from Source

As mentioned previously in Section 3.3, “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-1.6.0-openjdk-devel.x86_64 genisoimage mysql mysql-server ws-commons-
util MySQL-python tomcat6 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 a Maven 3 tarball and uncompress it in your home directory (or whatever location you prefer):

```
$ tar zxvf apache-maven-3.0.4-bin.tar.gz
```

```
$ export PATH=/usr/local/apache-maven-3.0.4//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/jre-1.6.0-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.

> **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.

3.6.1. Generating RPMS

Now that we have the prerequisites and source, you will cd to the **packaging/centos63/** directory.

```
$  cd packaging/centos63
```

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

```
$ ./package.sh
```

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-4.2.0.el6.x86_64.rpm
             cloudstack-awsapi-4.2.0.el6.x86_64.rpm
             cloudstack-cli-4.2.0.el6.x86_64.rpm
             cloudstack-common-4.2.0.el6.x86_64.rpm
             cloudstack-docs-4.2.0.el6.x86_64.rpm
             cloudstack-management-4.2.0.el6.x86_64.rpm
             cloudstack-usage-4.2.0.el6.x86_64.rpm
```

3.6.1.1. 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
```

```
$  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.

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

3.6.1.2. 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.

3.7. 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.

Why Non-OSS?

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 on master branch* 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 **nonoss** option:

   ```
                   $ mvn clean
                   $ mvn install -Dnonoss
   ```

4. Once you've built CloudStack with the **nonoss** profile, you can package it using the Section 3.6, "Building RPMs from Source" or Section 3.5, "Building DEB packages" instructions.

Chapter 4. Installation

4.1. 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.

4.2. Overview of 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 Section 4.3, “Minimum System Requirements”
2. Install the Management Server (choose single-node or multi-node). See Section 4.5, “Management Server Installation”
3. Log in to the UI. See Chapter 5, *User Interface*
4. Add a zone. Includes the first pod, cluster, and host. See Section 6.3, “Adding a Zone”
5. Add more pods (optional). See Section 6.4, “Adding a Pod”
6. Add more clusters (optional). See Section 6.5, “Adding a Cluster”
7. Add more hosts (optional). See Section 6.6, “Adding a Host”
8. Add more primary storage (optional). See Section 6.7, “Add Primary Storage”
9. Add more secondary storage (optional). See Section 6.8, “Add Secondary Storage”
10. Try using the cloud. See Section 6.9, “Initialize and Test”

4.3. Minimum System Requirements

4.3.1. 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 6.3+ or Ubuntu 12.04(.1)
- 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

4.3.2. 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
-

  Note

  If DHCP is used for hosts, ensure that no conflict occurs between DHCP server used for these hosts and the DHCP router created by CloudStack.
- 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.

under Citrix XenServer installation.

- Section 8.1.1, “System Requirements for KVM Hypervisor Hosts”
- Section 8.2.1, “System Requirements for XenServer Hosts”
- Section 8.3.1, “System Requirements for vSphere Hosts”
- Section 8.4.1, “System Requirements for LXC Hosts”

4.4. 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 Section 3.6, “Building RPMs from Source” or Section 3.5, “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.

4.4.1. 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 12.04 LTS (precise) are being built at this time.

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://cloudstack.apt-get.eu/ubuntu precise 4.2
```

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

```
$ wget -O - http://cloudstack.apt-get.eu/release.asc|apt-key add -
```

Now update your local apt cache.

```
$ apt-get update
```

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

4.4.2. 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://cloudstack.apt-get.eu/rhel/4.2/
enabled=1
gpgcheck=0
```

Now you should be able to install CloudStack using Yum.

4.5. Management Server Installation

4.5.1. 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 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

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

4.5.2. 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 www.cloudstack.org
   ```

4. Turn on NTP for time synchronization.

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

   a. Install NTP.

      ```
      yum install ntp
      ```

      ```
      apt-get install openntpd
      ```

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

4.5.3. 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.

4.5.3.1. Install on CentOS/RHEL

We start by installing the required packages:

```
yum install cloudstack-management
```

4.5.3.2. Install on Ubuntu

```
apt-get install cloudstack-mangagement
```

4.5.3.3. 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 vhd-util.

If the Management Server is RHEL or CentOS, copy vhd-util to /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver.

If the Management Server is Ubuntu, copy vhd-util to /usr/share/cloudstack-common/scripts/vm/hypervisor/xenserver.

4.5.4. 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.

4.5.4.1. 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 Section 4.5.4.2, “Install the Database on a Separate Node”.

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

   ```
   yum install mysql-server
   ```

   ```
   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.

3. 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.

   > **Note**
   >
   > On Ubuntu, 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.

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

4. 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.

   ```
   service mysql restart
   ```

5. (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
   ```

6. 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
      ```

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 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 Section 4.5.5, “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 Section 4.5.5, “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 Section 4.5.5, “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>@localhost \
--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.”

> **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.

8. 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
```

9. 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 see the message “CloudStack Management Server setup is done.”

4.5.4.2. 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 Section 4.5.4.1, “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 Section 4.3.1, “Management Server, Database, and Storage System Requirements”.

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

```
yum install mysql-server
```

```
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.

```
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 Section 4.5.5, “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 Section 4.5.5, “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.”

4.5.5. 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.

4.5.6. 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"/>
            <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>
```

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.

4.5.7. 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.

> **Alternative Storage**
>
> 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:

- Section 2.6, “About Primary Storage”
- Section 2.7, “About Secondary Storage”

A production installation typically uses a separate NFS server. See Section 4.5.7.1, “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 Section 4.5.7.2, “Using the Management Server as the NFS Server”.

4.5.7.1. 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
   ```

```
# 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:/nfs/share/secondary /mnt/secondary
   ```

4.5.7.2. 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:

   ```
   $ sudo 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 /primarymount
    # mount -t nfs <management-server-name>:/export/primary /primarymount
    # umount /primarymount
    # mkdir /secondarymount
    # mount -t nfs <management-server-name>:/export/secondary /secondarymount
    # umount /secondarymount
    ```

4.5.8. Prepare and Start 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 Section 4.5.2, “Prepare the Operating System” and Section 3.6, “Building RPMs from Source” or Section 3.5, “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 Section 4.5.4.2, “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.

6. Repeat these steps on each additional Management Server.

7. Be sure to configure a load balancer for the Management Servers. See Section 13.6, “Management Server Load Balancing”.

4.5.9. 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 Section 4.5.5, “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 XenServer:

     ```
     # /usr/lib64/cloud/common/scripts/storage/secondary/cloud-install-sys-tmplt -m
     ```

```
/mnt/secondary -u
http://d21ifhcun6b1t2.cloudfront.net/templates/4.2/systemvmtemplate-2013-07-12-
master-xen.vhd.bz2 -h xenserver -s <optional-management-server-secret-key> -F
```

- For vSphere:

```
# /usr/lib64/cloud/common/scripts/storage/secondary/cloud-install-sys-tmplt -m
/mnt/secondary -u
http://d21ifhcun6b1t2.cloudfront.net/templates/4.2/systemvmtemplate-4.2-vh7.ova -h
vmware -s <optional-management-server-secret-key>  -F
```

- For KVM:

```
# /usr/lib64/cloud/common/scripts/storage/secondary/cloud-install-sys-tmplt -m
/mnt/secondary -u
http://d21ifhcun6b1t2.cloudfront.net/templates/4.2/systemvmtemplate-2013-06-12-
master-kvm.qcow2.bz2 -h kvm -s <optional-management-server-secret-key> -F
```

- For LXC:

```
# /usr/lib64/cloud/common/scripts/storage/secondary/cloud-install-sys-tmplt -m
/mnt/secondary -u http://d21ifhcun6b1t2.cloudfront.net/templates/acton/acton-
systemvm-02062012.qcow2.bz2 -h lxc -s <optional-management-server-secret-key> -F
```

On Ubuntu, use the following path instead:

```
# /usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-tmplt
```

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.

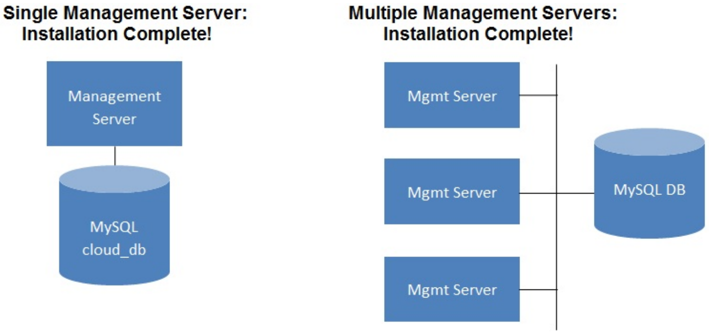
   When the script has finished, unmount secondary storage and remove the created directory.

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

3. Repeat these steps for each secondary storage server.

4.5.10. 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.

Chapter 5. User Interface

5.1. 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.

5.1.1. 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.

5.1.2. 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.

5.1.3. 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 walkthrough 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 Section 5.1.4, "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.

5.1.4. 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. 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.

5.2.1. Creating an Instance Template that Supports SSH Keys

Create a 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 Scriptto 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.

5.2.2. 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
QpbSGDIa11Jsc+XNDx2fgRinDsxXI/zJYXTKRhSl/LIPHBw/brW8vzxhOlSOrwm7
VvemkkgpAkEAwSeEw394LYZiEVv395ar9MLRVTVLwpo54jC4tsOxQCBlloocK
lYaocpk0yBqqOUSBawfIiDCuLXSdvBo1Xz5ICTM19vgvEp/+kMuECQBzm
nVo8b2Gvyagqt/KEQo8wzH2THghZ1qQ1QRhIeJG2aissEacF6bGB2oZ7Igim5L14
4KR7OeEToyCLC2k+02UCQQCrniSnWKtDVoVqeK/zbB32JhW3Wullv5p5zUEcd
KfEEuzcCUIxtJYTahJ1pvlFkQ8anpuxjSEDp8x/18bq3
-----END RSA PRIVATE KEY-----
```

3. Save the file.

5.2.3. 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.

5.2.4. 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 exaple, 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.

5.2.5. 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.

Chapter 6. Steps to Provisioning Your Cloud Infrastructure

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 Chapter 2, *Cloud Infrastructure Concepts*.

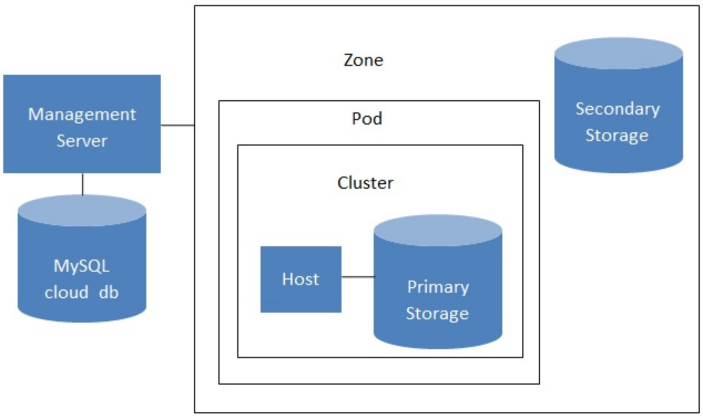
6.1. Overview of Provisioning Steps

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 Section 1.3.2, “Cloud Infrastructure Overview”.

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

1. Define regions (optional). See Section 6.2, “Adding Regions (optional)”.
2. Add a zone to the region. See Section 6.3, “Adding a Zone”.
3. Add more pods to the zone (optional). See Section 6.4, “Adding a Pod”.
4. Add more clusters to the pod (optional). See Section 6.5, “Adding a Cluster”.
5. Add more hosts to the cluster (optional). See Section 6.6, “Adding a Host”.
6. Add primary storage to the cluster. See Section 6.7, “Add Primary Storage”.
7. Add secondary storage to the zone. See Section 6.8, “Add Secondary Storage”.
8. Initialize and test the new cloud. See Section 6.9, “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

6.2. Adding Regions (optional)

Grouping your cloud resources into geographic regions is an optional step when provisioning the cloud. For an overview of regions, see Section 2.1, “About Regions”.

6.2.1. 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.

6.2.2. 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>
   ```

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.

6.2.3. 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.

6.2.4. 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.

6.3. 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 Section 5.1, "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:
   - Section 6.3.1, "Basic Zone Configuration"
   - Section 6.3.2, "Advanced Zone Configuration"

6.3.1. 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 Section 2.3, “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.

6.3.2. 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 Section 2.8.3, “Advanced Zone Network Traffic Types”. This screen starts 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 Section 2.3, “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 Section 2.8.6, “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 Section 2.4, “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 Section 2.5, “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 Section 2.6, “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, 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. 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.
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. 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.
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. 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.
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. 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.
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.

11. In a new zone, CloudStack adds the first secondary storage server for you. For an overview of what secondary storage is, see Section 2.7, “About 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.

12. Click Launch.

6.4. 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 Section 5.1, “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.

6.5. 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.

6.5.1. 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.

6.5.2. 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.

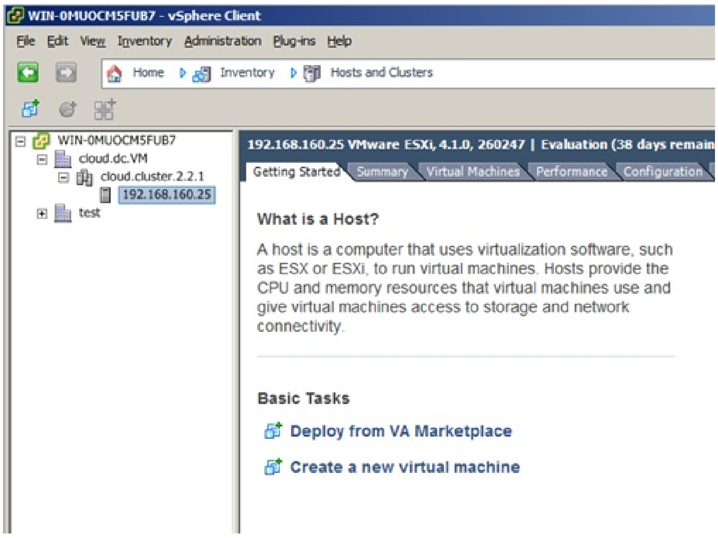
like live migration. Clusters also require shared storage such as NFS or iSCSI.

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.



What is a Host?

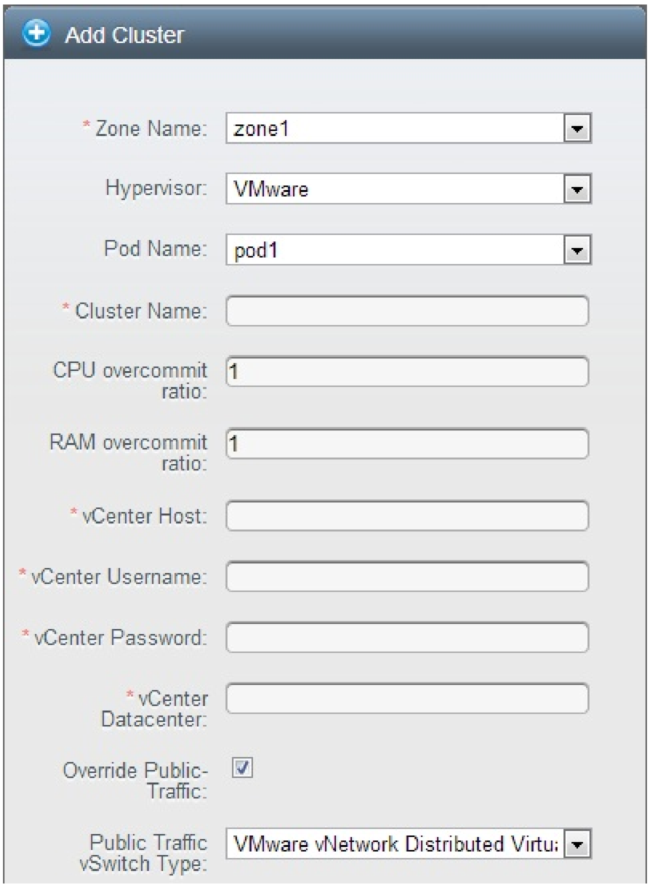
A host is a computer that uses virtualization software, such as ESX or ESXi, to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity.

Basic Tasks

Deploy from VA Marketplace

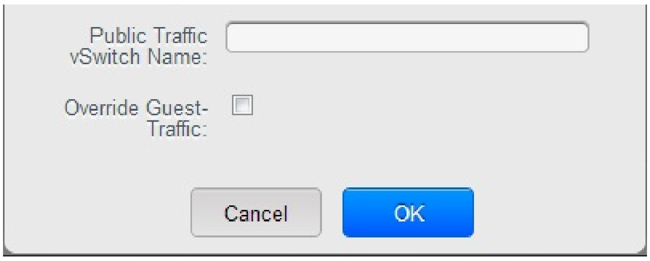
Create a new virtual machine

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.



Add Cluster

* 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: □

- **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.

6.6. 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.
   - Section 6.6.1, “Adding a Host (XenServer or KVM)”
   - Section 6.6.2, “Adding a Host (vSphere)”

6.6.1. Adding a Host (XenServer or KVM)

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

6.6.1.1. 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.

6.6.1.1.1. 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
   ```

6.6.1.1.2. 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

6.6.1.2. 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.

6.6.2. 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.

6.7. Add Primary Storage

6.7.1. System Requirements for Primary Storage

Hardware requirements:

- Any standards-compliant iSCSI 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.

6.7.2. 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.

Note

Primary storage can also be added at the zone level through the CloudStack API (adding zone-level primary storage is not yet supported through the CloudStack UI).
Once primary storage has been added at the zone level, it can be managed through the CloudStack UI.

1. Log in to the CloudStack UI (see Section 5.1, “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.
   - **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".
   - **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.

6.7.3. 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

At this time, a custom storage plug-in, such as the SolidFire storage plug-in, can only be leveraged for data disks (through Disk Offerings).

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 Disk Offerings, an administrator is able to build an environment in which a 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)]

Example URL to add primary storage to CloudStack based on the SolidFire plug-in (note that URL encoding is used with the value of the url key, so '%3A' equals ':','%3B' equals '&' and '%3D' equals '='):

```
http://127.0.0.1:8080/client/api?command=createStoragePool &scope=zone &zoneId=cf4e6ddf-8ae7-4194-8270-
d46733a52b55 &name=SolidFire_121258566 &url=MVIP%3D192.168.138.180%3A443 %3BSVIP%3D192.168.56.7
%3BclusterAdminUsername%3Dadmin %3BclusterAdminPassword%3Dpassword %3BclusterDefaultMinIops%3D200
%3BclusterDefaultMaxIops%3D300 %3BclusterDefaultBurstIopsPercentOfMaxIop%3D2.5 &provider=SolidFire
&tags=SolidFire_SAN_1 &capacityIops=4000000 &capacityBytes=2251799813685248 &hypervisor=Any &response=json
&apiKey=VrrkiZQWFFgSdA6k3DYtoKLcrgQJjZXoSWzicHXt8rYd9Bl47p8L39p0p8vfDpiljtlcMLn_jatMSqCWv5Cs-
Q&signature=wqf8KzcPpY2JmT1Sxk%2F%2BWbgX3l8%3D
```

6.8. Add Secondary Storage

6.8.1. System Requirements for Secondary Storage

- NFS storage appliance or Linux NFS server
- (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.

6.8.2. 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

Be sure there is nothing 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 Section 4.5.7, “Prepare NFS Shares”.
2. Make sure you prepared the system VM template during Management Server installation. See Section 4.5.9, “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, or NFS, 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.

     > **Warning**
     >
     > You can use only a single 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.
   - NFS server. The name of the zone's Secondary Staging Store.
   - Path. The path to the zone's Secondary Staging Store.

6.8.3. 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.

6.9. 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.

Chapter 7. Setting Configuration Parameters

7.1. 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 as a host tag when you add a new host to the cloud.

7.2. 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.

7.3. 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.

7.4. 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.notificationthreshold	The percentage, as a value between 0 and 1, of allocated storage utilization above which alerts are sent that the storage is below the threshold.
cluster	cluster.storage.capacity.notificationthreshold	The percentage, as a value between 0 and 1, of storage utilization above which alerts are sent that the available storage is below the threshold.
cluster	cluster.cpu.allocated.capacity.notificationthreshold	The percentage, as a value between 0 and 1, of cpu utilization above which alerts are sent that the available CPU is below the threshold.
cluster	cluster.memory.allocated.capacity.notificationthreshold	The percentage, as a value between 0 and 1, of memory utilization above which alerts are sent that the available memory is below the threshold.
cluster	cluster.cpu.allocated.capacity.disablethreshold	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.disablethreshold	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.

Chapter 8. Hypervisor Installation

8.1. KVM Hypervisor Host Installation

8.1.1. 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: 6.3
- Ubuntu: 12.04(.1)

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: 0.9.4 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: 0.9.11 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

8.1.2. 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

8.1.3. 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.

8.1.4. 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 Section 6.6, “Adding a Host”. It is recommended that you continue to read the documentation before adding the host!

8.1.4.1. 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 two new 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)
```

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
  ```

Note

host-passthrough may lead to migration failure,if you have this problem,you should use host-model or custom

8.1.5. 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
   ```

8.1.6. 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
      ```

8.1.7. 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.

8.1.7.1. 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!

8.1.7.2. 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.

8.1.7.2.1. 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
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!

8.1.7.2.2. 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!

8.1.8. 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.

8.1.8.1. 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/

8.1.8.2. 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!

8.1.8.3. Configuring the network bridges

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.

8.1.8.3.1. 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
```

8.1.8.3.2. Configure 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!

8.1.9. 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.

8.1.9.1. 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
```

8.1.9.2. 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.

8.1.10. Add the host to CloudStack

The host is now ready to be added to a cluster. This is covered in a later section, see Section 6.6, “Adding a Host”. It is recommended that you continue to read the documentation before adding the host!

8.1.11. Hypervisor Support for Primary Storage

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

	VMware vSphere	Citrix XenServer	KVM	
Format for Disks, Templates, and Snapshots	VMDK	VHD	QCOW2	
iSCSI support	VMFS	Clustered LVM	Yes, via Shared Mountpoint	
Fiber Channel support	VMFS	Yes, via Existing SR	Yes, via Shared Mountpoint	
NFS support	Y	Y	Y	
Local storage support	Y	Y	Y	
Storage over-provisioning	NFS and iSCSI	NFS	NFS	

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.

8.2. Citrix XenServer Installation for CloudStack

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 Section 8.2.11, “Upgrading XenServer Versions”.

8.2.1. 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
- 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-do-date hotfixes can lead to data corruption and lost VMs.

8.2.2. XenServer Installation Steps

1. From https://www.citrix.com/English/ss/downloads/, download the appropriate version of XenServer for your CloudStack version (see Section 8.2.1, “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.

2. After installation, perform the following configuration steps, which are described in the next few sections:

Required	Optional
Section 8.2.3, “Configure XenServer dom0 Memory”	Section 8.2.7, “Install CloudStack XenServer Support Package (CSP)”
Section 8.2.4, “Username and Password”	Set up SR if not using NFS, iSCSI, or local disk; see Section 8.2.8, “Primary Storage Setup for XenServer”
Section 8.2.5, “Time Synchronization”	Section 8.2.9, “iSCSI Multipath Setup for XenServer (Optional)”
Section 8.2.6.1, “Getting and Deploying a License”	Section 8.2.10, “Physical Networking Setup for XenServer”

8.2.3. 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.

8.2.4. Username and Password

All XenServers in a cluster must have the same username and password as configured in CloudStack.

8.2.5. 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
   ```

8.2.6. Licensing

Citrix XenServer Free version provides 30 days usage without a license. Following the 30 day trial, XenServer requires a free activation and license. You can choose to install a license now or skip this step. If you skip this step, you will need to install a license when you activate and license the XenServer.

8.2.6.1. Getting and Deploying a License

If you choose to install a license now you will need to use the XenCenter to activate and get a license.

1. In XenCenter, click Tools > License manager.
2. Select your XenServer and select Activate Free XenServer.
3. Request a license.

You can install the license with XenCenter or using the xe command line tool.

8.2.7. 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.

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.

8.2.8. 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 Section 6.7, “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.

8.2.9. 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

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 Section 6.7, “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.

8.2.10. 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>
```

8.2.10.1. 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 Section 8.2.10.4, “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 Section 4.5, “Management Server Installation”.

If you are using two NICs bonded together to create a public network, see Section 8.2.10.4, “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>
   ```

8.2.10.2. 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.

8.2.10.3. 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
```

8.2.10.4. 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.

8.2.10.4.1. Management Network Bonding

The administrator must bond the management network NICs prior to adding the host to CloudStack.

8.2.10.4.2. 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.

8.2.10.4.3. 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.

8.2.10.4.4. 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.

8.2.10.4.5. 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]
```

8.2.10.4.6. 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.

8.2.11. 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
```

8.3. VMware vSphere Installation and Configuration

If you want to use the VMware vSphere hypervisor to run guest virtual machines, install vSphere on the host(s) in your cloud.

8.3.1. System Requirements for vSphere Hosts

8.3.1.1. Software requirements:

- vSphere and vCenter, both version 4.1 or 5.0.

  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.

Apply All Necessary Hotfixes

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

8.3.1.2. 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

8.3.1.3. 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" at http://pubs.vmware.com/vsp40/wwhelp/wwhimpl/js/html/wwhelp.htm#href=install/c_vc_hw.html.

8.3.1.4. Other requirements:

- VMware vCenter Standard Edition 4.1 or 5.0 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 or 5.0. 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 Section 8.3.5.2, “Configure vCenter Management Network”.
- CloudStack requires ESXi. ESX is not supported.
- All resources used for CloudStack must be used for CloudStack only. CloudStack cannot 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 a cluster in a separate Datacenter in vCenter.
- The cluster that will be managed by CloudStack should not contain any VMs. Do not run the management server, vCenter or any other VMs 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 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.

8.3.2. Preparation Checklist for VMware

For a smoother installation, gather the following information before you start:

- Information listed in Section 8.3.2.1, “vCenter Checklist”
- Information listed in Section 8.3.2.2, “Networking Checklist for VMware”

8.3.2.1. vCenter Checklist

You will need the following information about vCenter.

vCenter Requirement	Value	Notes
vCenter User		This user must have admin privileges.
vCenter User Password		Password for the above user.

vCenter User Password		Password for the above user.
vCenter Datacenter Name		Name of the datacenter.
vCenter Cluster Name		Name of the cluster.

8.3.2.2. Networking Checklist for VMware

You will need the following information about VLAN.

VLAN Information	Value	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.

8.3.3. 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:

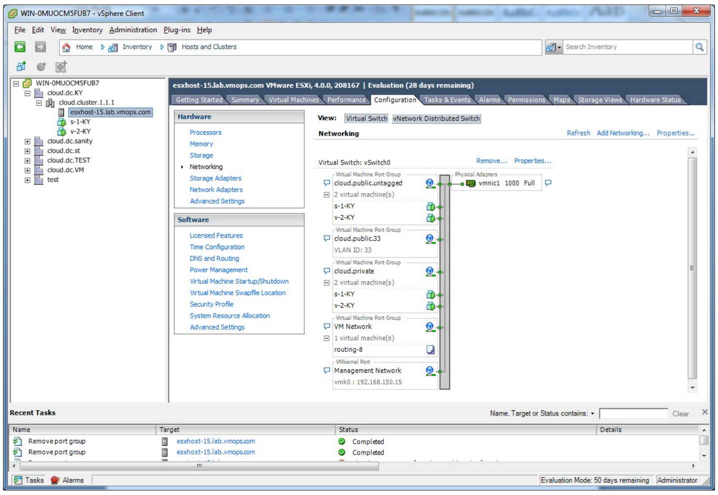
Required	Optional
ESXi host setup	NIC bonding
Configure host physical networking, virtual switch, vCenter Management Network, and extended port range	Multipath storage
Prepare storage for iSCSI	
Configure clusters in vCenter and add hosts to them, or add hosts without clusters to vCenter	

8.3.4. ESXi Host setup

All ESXi hosts should enable CPU hardware virtualization support in BIOS. Please note hardware virtualization support is not enabled by default on most servers.

8.3.5. 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.



WIN-0MUOCM5FUB7 - vSphere Client

esxhost-15.lab.vmops.com VMware ESXi, 4.0.0, 208167 | Evaluation (28 days remaining)

Virtual Switch: vSwitch0

Name	Target	Status	Details
Remove port group	esxhost-15.lab.vmops.com	Completed	
Remove port group	esxhost-15.lab.vmops.com	Completed	

Evaluation Mode: 50 days remaining

In the host configuration tab, click the "Hardware/Networking" link to bring up the networking configuration page as above.

8.3.5.1. Configure Virtual Switch

A default virtual switch vSwitch0 is created. CloudStack requires all ESXi hosts in the cloud to use the same set of virtual switch names. If you change the default virtual switch name, you will need to configure one or more CloudStack configuration variables as well.

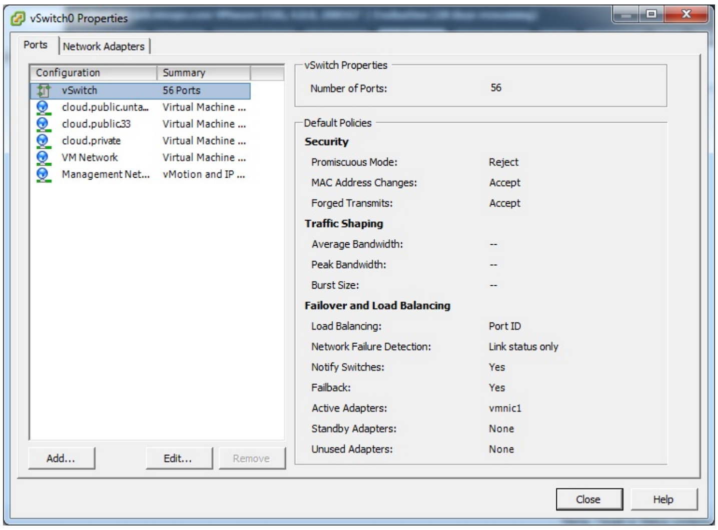
8.3.5.1.1. Separating Traffic

CloudStack allows you to use vCenter to configure three separate networks per ESXi host. These networks are identified by the name of the vSwitch they are connected to. The allowed 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.

8.3.5.1.2. 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).



vSwitch0 Properties

Configuration	Summary
vSwitch	56 Ports
cloud.public.unta...	Virtual Machine ...
cloud.public33	Virtual Machine ...
cloud.private	Virtual Machine ...
VM Network	Virtual Machine ...
Management Net...	vMotion and IP ...

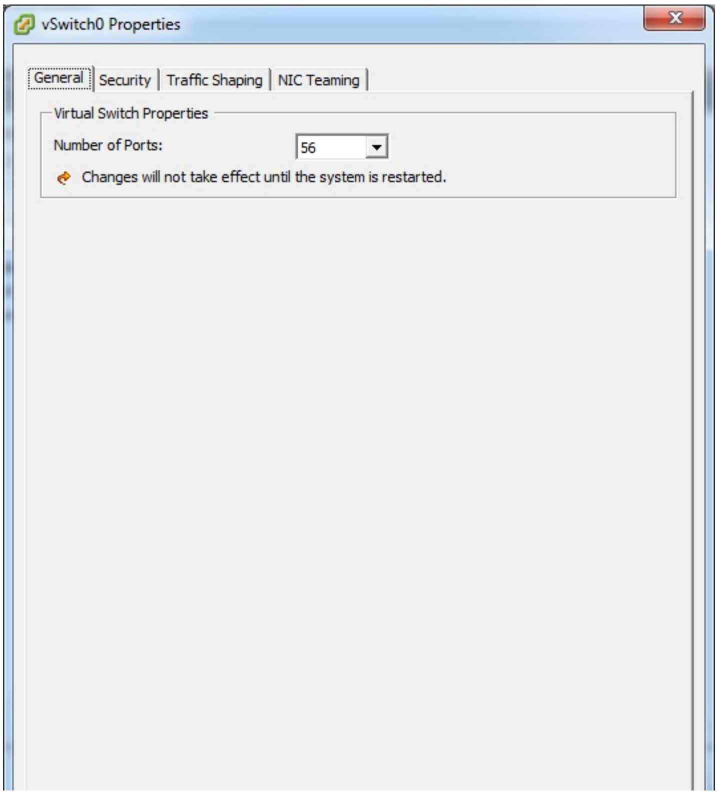
vSwitch Properties

Number of Ports:	56

Default Policies

Security	
Promiscuous Mode:	Reject
MAC Address Changes:	Accept
Forged Transmits:	Accept
Traffic Shaping	
Average Bandwidth:	--
Peak Bandwidth:	--
Burst Size:	--
Failover and Load Balancing	
Load Balancing:	Port ID
Network Failure Detection:	Link status only
Notify Switches:	Yes
Failback:	Yes
Active Adapters:	vmnic1
Standby Adapters:	None
Unused Adapters:	None

In vSwitch properties dialog, select the vSwitch and click Edit. You should see the following dialog:

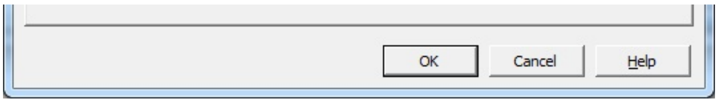


vSwitch0 Properties

Virtual Switch Properties

Number of Ports: 56

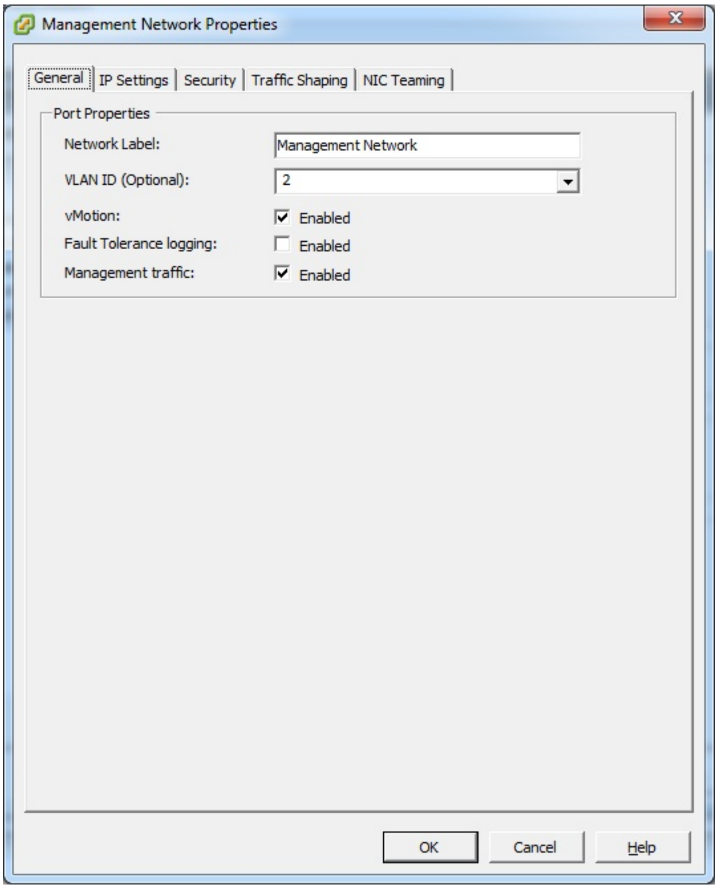
Changes will not take effect until the system is restarted.



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.

8.3.5.2. 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.



Network Label: Management Network

VLAN ID (Optional): 2

vMotion: ☑ Enabled

Fault Tolerance logging: □ Enabled

Management traffic: ☑ Enabled

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.

8.3.5.3. 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
```

8.3.5.4. Configure NIC Bonding for vSphere

NIC bonding on vSphere hosts may be done according to the vSphere installation guide.

8.3.6. Configuring a vSphere Cluster with Nexus 1000v Virtual Switch

8.3.6. 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 Section 8.3, “VMware vSphere Installation and Configuration”

8.3.6.1. 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.

8.3.6.2. 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 Section 8.3.6.3, “Nexus 1000v Virtual Switch Preconfiguration”

8.3.6.3. Nexus 1000v Virtual Switch Preconfiguration

8.3.6.3.1. 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

8.3.6.3.1.1. 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.

8.3.6.3.1.2. 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:

Network Requirements	Value	Notes
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.

8.3.6.3.1.3. VSM Configuration Checklist

You will need the following information about network configuration:

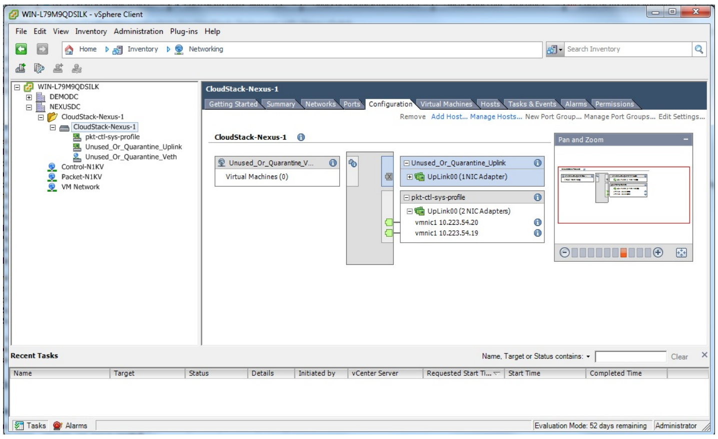
VSM Configuration Parameters Value Notes	Value	Notes
Admin Name and Password		The admin name and password to connect to the VSM appliance. You must specify these credentials while configuring Nexus virtual switch.
Management IP Address		This is the IP address of the VSM appliance. This is the IP address you specify in the virtual switch IP Address field while configuting Nexus virtual switch.
SSL	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.

8.3.6.3.2. 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.


WIN-L79M9QDSILK - vSphere Client
File Edit View Inventory Administration Plug-ins Help
Home Inventory Networking
Search Inventory
WIN-L79M9QDSILK
DEMODC
NEXUSDC
CloudStack-Nexus-1
CloudStack-Nexus-1
pkt-ctl-sys-profile
Unused_Or_Quarantine_Uplink
Unused_Or_Quarantine_Veth
Control-N1KV
Packet-N1KV
VM Network
CloudStack-Nexus-1
Getting Started Summary Networks Ports Configuration Virtual Machines Hosts Tasks & Events Alarms Permissions
Remove Add Host... Manage Hosts... New Port Group... Manage Port Groups... Edit Settings...
CloudStack-Nexus-1
Unused_Or_Quarantine_V...
Virtual Machines (0)
Unused_Or_Quarantine_Uplink
UpLink00 (1 NIC Adapter)
pkt-ctl-sys-profile
UpLink00 (2 NIC Adapters)
vmnic1 10.223.54.20
vmnic1 10.223.54.19
Pan and Zoom
Recent Tasks
Name, Target or Status contains:
Clear
Name
Target
Status
Details
Initiated by
vCenter Server
Requested Start Ti...
Start Time
Completed Time
Tasks
Alarms
Evaluation Mode: 52 days remaining
Administrator

For information on creating a port profile, see Cisco Nexus 1000V Port Profile Configuration Guide.

8.3.6.3.3. 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 Section 8.3.6, “Configuring a vSphere Cluster with Nexus 1000v Virtual Switch”.

8.3.6.3.4. 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.

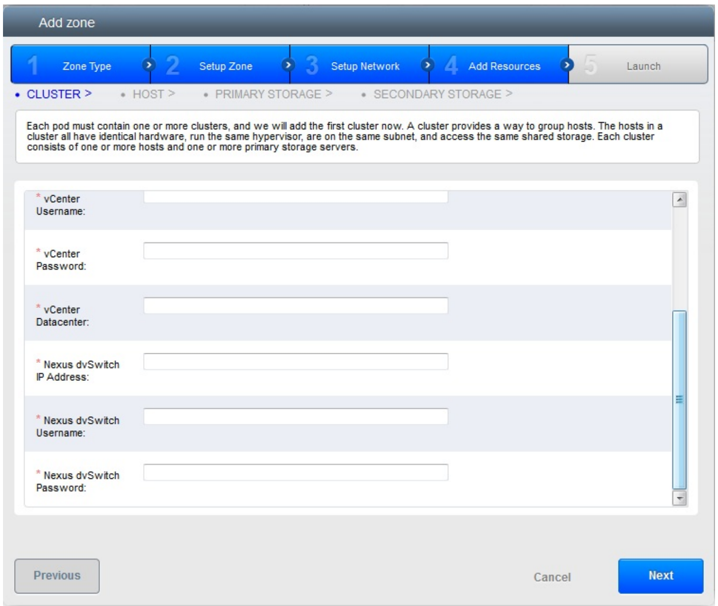
8.3.6.4. Enabling Nexus Virtual Switch in CloudStack

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.

8.3.6.5. Configuring Nexus 1000v Virtual Switch in CloudStack

You can configure Nexus dvSwitch by adding the necessary resources while the zone is being created.



Each pod must contain one or more clusters, and we will add the first cluster now. A cluster provides a way to group hosts. The hosts in a cluster all have identical hardware, run the same hypervisor, are on the same subnet, and access the same shared storage. Each cluster consists of one or more hosts and one or more primary storage servers.

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 Section 6.5.2, “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.

8.3.6.6. 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.

8.3.7. 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.

8.3.7.1. About VMware Distributed Virtual Switch

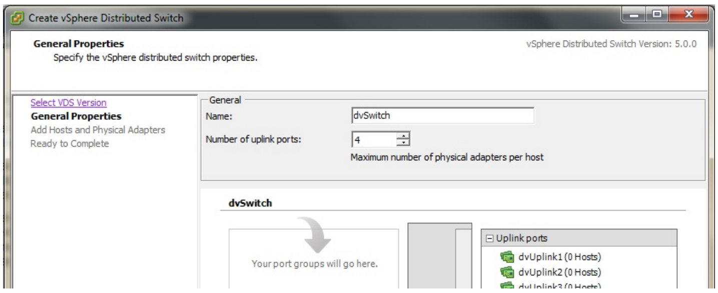
VMware VDS is an aggregation of host-level virtual switches on a VMware vCenter server. VDS abstracts the configuration of individual virtual switches that span across a large number of hosts, and enables centralized provisioning, administration, and monitoring for your entire datacenter from a centralized 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.

8.3.7.2. 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.

8.3.7.3. Preparation Checklist

For a smoother configuration of VMware VDS, note down the VDS name you have added in the datacenter before you start:



Create vSphere Distributed Switch

General Properties

Specify the vSphere distributed switch properties.

vSphere Distributed Switch Version: 5.0.0

General

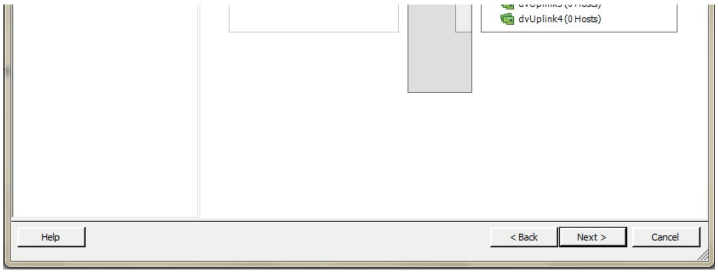
Name: dvSwitch

Number of uplink ports: 4

Maximum number of physical adapters per host

dvSwitch

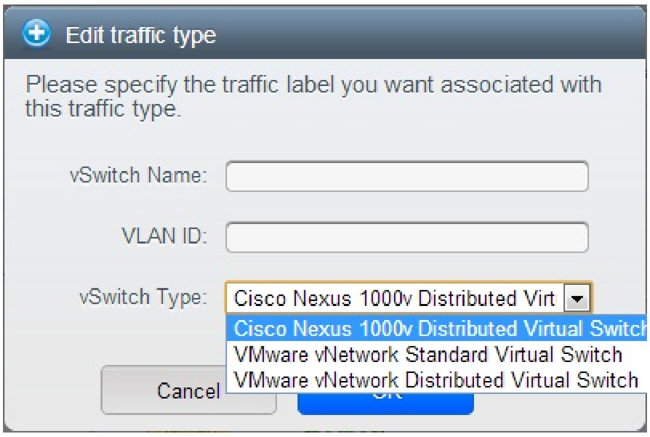
Your port groups will go here.



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.



Edit traffic type

Please specify the traffic label you want associated with this traffic type.

vSwitch Name:

VLAN ID:

vSwitch Type: Cisco Nexus 1000v Distributed Virt

Cisco Nexus 1000v Distributed Virtual Switch
VMware vNetwork Standard Virtual Switch
VMware vNetwork Distributed Virtual Switch
Cancel
OK

- 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

Fields	Name	Description
1	Represents the 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
2	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.
3	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 switch would 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.

8.3.7.4. 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.

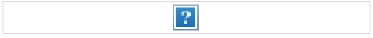
CloudStack supports orchestration of virtual networks in a deployment with a mix of Virtual Distributed Switch, Standard Virtual Switch and Nexus 1000v Virtual Switch.

8.3.7.5. 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 Section 6.5.2, “Add Cluster: vSphere”.

In both these cases, you must specify the following parameters to configure VDS:



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.

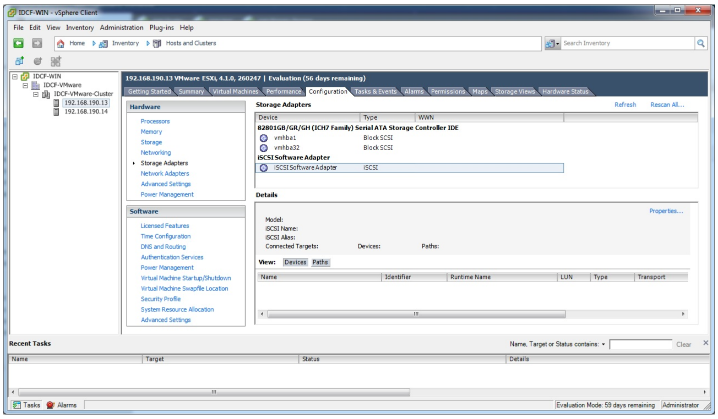
8.3.8. Storage Preparation for vSphere (iSCSI only)

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.

8.3.8.1. Enable iSCSI initiator for ESXi hosts

1. In vCenter, go to hosts and Clusters/Configuration, and click Storage Adapters link. You will see:



192.168.190.13 VMware ESXi, 4.1.0, 260247 | Evaluation (56 days remaining)

Storage Adapters

Device	Type	WWN
82801GB/GR/GH (ICH7 Family) Serial ATA Storage Controller IDE		
vmhba1	Block SCSI	
vmhba32	Block SCSI	
iSCSI Software Adapter		
iSCSI Software Adapter	iSCSI	

Details

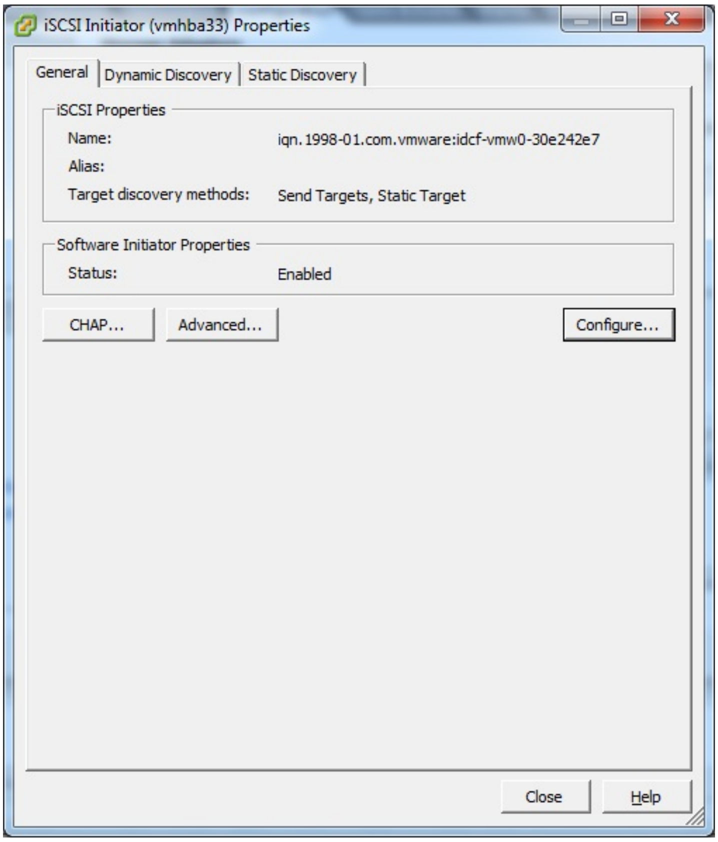
Model:
iSCSI Name:
iSCSI Alias:
Connected Targets: Devices: Paths:

Name	Identifier	Runtime Name	LUN	Type	Transport

Name	Target	Status	Details

Evaluation Mode: 59 days remaining

2. Select iSCSI software adapter and click Properties.



iSCSI Properties

Name: iqn.1998-01.com.vmware:idcf-vmw0-30e242e7

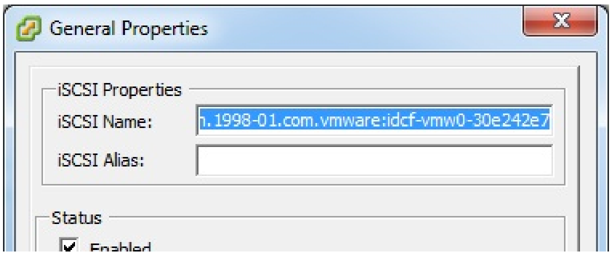
Alias:

Target discovery methods: Send Targets, Static Target

Software Initiator Properties

Status: Enabled

3. Click the Configure... button.



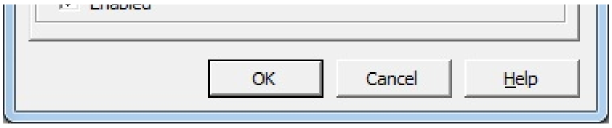
iSCSI Properties

iSCSI Name: n.1998-01.com.vmware:idcf-vmw0-30e242e7

iSCSI Alias:

Status

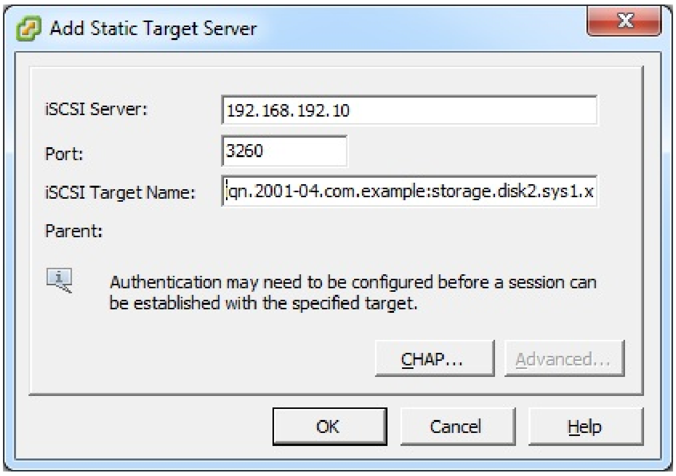
☑ Enabled



4. Check Enabled to enable the initiator.
5. Click OK to save.

8.3.8.2. Add iSCSI target

Under the properties dialog, add the iSCSI target info:



iSCSI Server: 192.168.192.10

Port: 3260

iSCSI Target Name: iqn.2001-04.com.example:storage.disk2.sys1.x

Parent:

Authentication may need to be configured before a session can be established with the specified target.

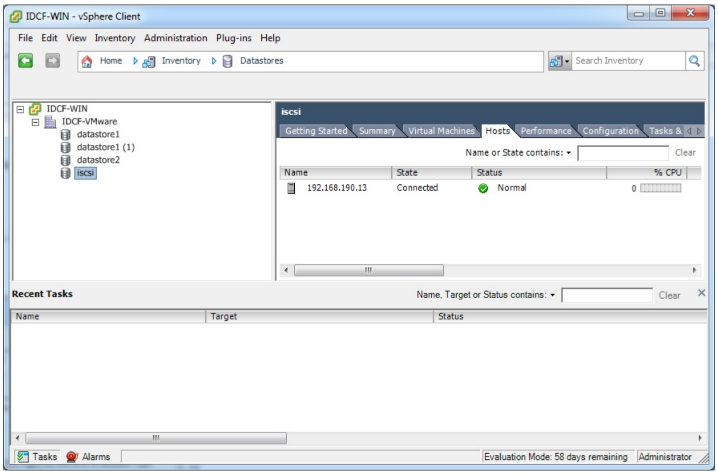
Repeat these steps for all ESXi hosts in the cluster.

8.3.8.3. 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.



Name	State	Status	% CPU
192.168.190.13	Connected	Normal	0

Name	Target	Status

8.3.8.4. Multipathing for vSphere (Optional)

Storage multipathing on vSphere nodes may be done according to the vSphere installation guide.

8.3.9. 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 Section 6.5.2, “Add Cluster: vSphere”).

8.3.10. 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 Section 5.1, “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.

8.4. LXC Installation and Configuration

8.4.1. 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

8.4.2. 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

8.4.3. Prepare the Operating System

8.4.3. 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.

8.4.4. 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 Section 6.6, “Adding a Host”. It is recommended that you continue to read the documentation before adding the host!

8.4.5. 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
   ```

8.4.6. 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
      ```

8.4.7. 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.

8.4.7.1. 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!

8.4.7.2. 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.

8.4.7.2.1. 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
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!

8.4.7.2.2. 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!

8.4.8. 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.

8.4.8.1. 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
```

8.4.8.2. 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.

8.4.9. Add the host to CloudStack

The host is now ready to be added to a cluster. This is covered in a later section, see Section 6.6, “Adding a Host”. It is recommended that you continue to read the documentation before adding the host!

Chapter 9. Additional Installation Options

The next few sections describe CloudStack features above and beyond the basic deployment options.

9.1. 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.

9.1.1. 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.

9.1.2. Steps to Install the Usage Server

1. Run ./install.sh.

   ```
   # ./install.sh
   ```

   You should see a few messages as the installer prepares, followed by a list of choices.

2. Choose "S" to install the Usage Server.

   ```
       > S
   ```

3. Once installed, start the Usage Server with the following command.

   ```
   # service cloudstack-usage start
   ```

The Administration Guide discusses further configuration of the Usage Server.

9.2. 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.

9.3. 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.

password "password". This assumes that master and slave run on the 172.16.1.0/24 network.

```
# mysql -u root
mysql> create user 'cloud-repl'@'172.16.1.%' identified by 'password';
mysql> grant replication slave on *.* TO 'cloud-repl'@'172.16.1.%';
mysql> flush privileges;
mysql> flush tables with read lock;
```

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.

9.3.1. 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:

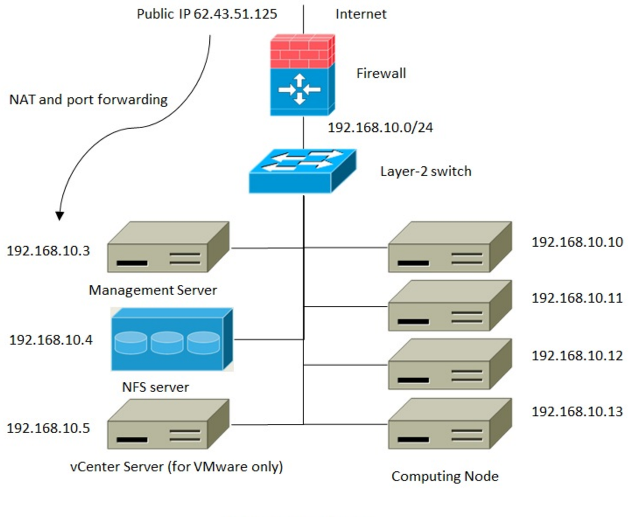
   ```
   # service cloudstack-management start
   ```

Chapter 10. Choosing a Deployment Architecture

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.

10.1. Small-Scale Deployment

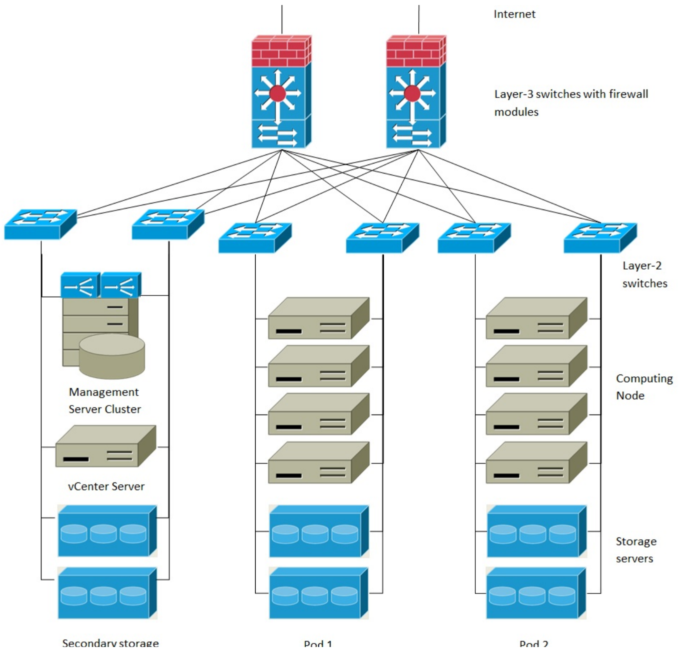


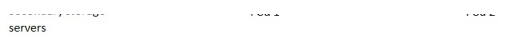
Small-Scale Deployment

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.

10.2. 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.

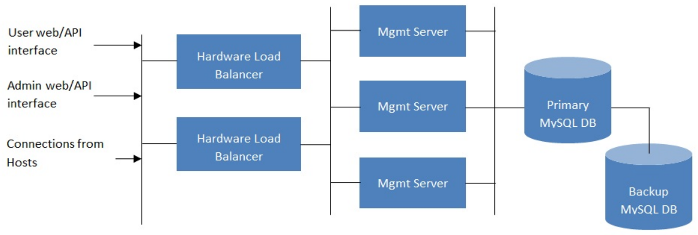
  When the cloud spans multiple zones, the firewalls should enable site-to-site VPN such that servers in different zones can directly reach each other.
- A layer-2 access switch layer is established for each pod. Multiple switches can be stacked to increase port count. In either case, redundant pairs of 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.

10.3. 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.

10.4. 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.



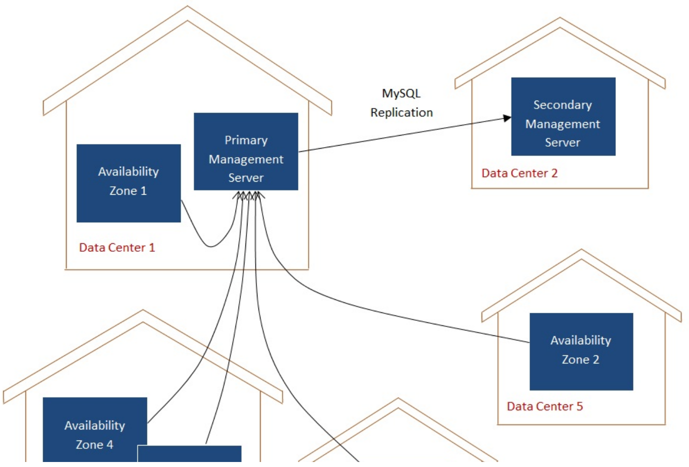
Multi-Node Management Server Deployment

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.

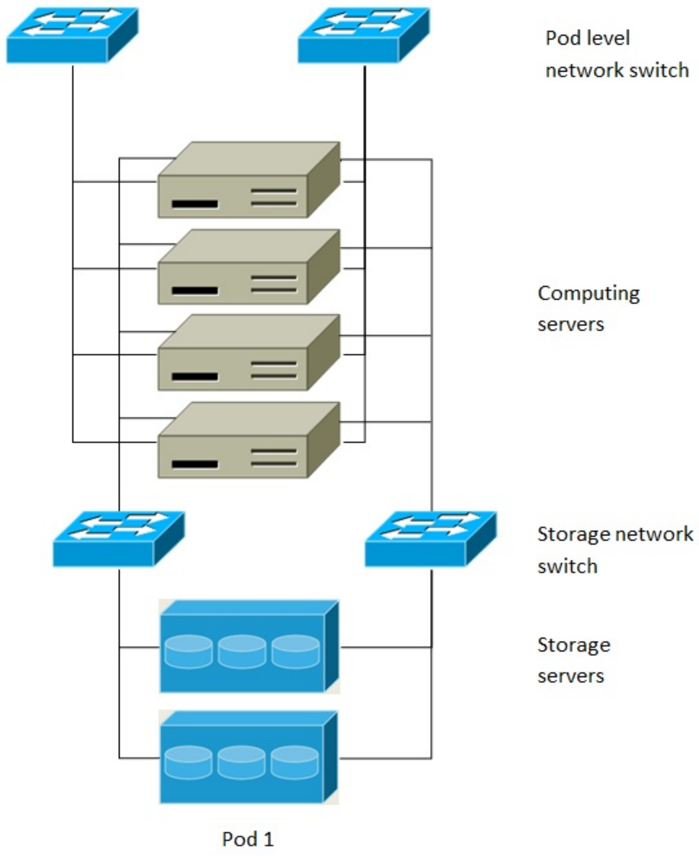
10.5. Multi-Site Deployment

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.

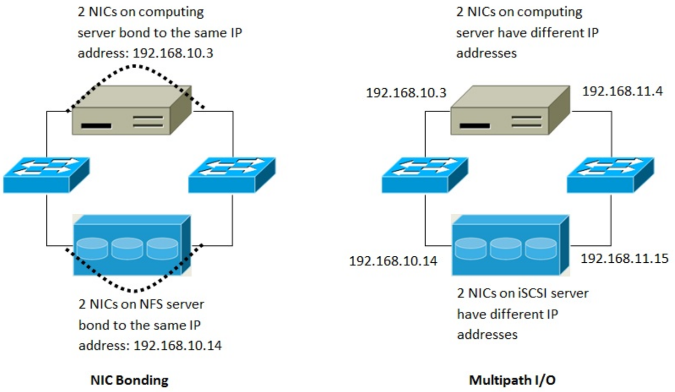


Separate Storage Network

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.


2 NICs on computing server bond to the same IP address: 192.168.10.3
2 NICs on NFS server bond to the same IP address: 192.168.10.14
NIC Bonding
2 NICs on computing server have different IP addresses
192.168.10.3
192.168.11.4
192.168.10.14
192.168.11.15
2 NICs on iSCSI server have different IP addresses
Multipath I/O

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.

Chapter 11. Choosing a Hypervisor: Supported Features

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 6.0.2	vSphere 4.1/5.0	KVM - RHEL 6.2	OVM 2.3	Bare Metal
Network Throttling	Yes	Yes	No	No	N/A
Security groups in zones that use basic networking	Yes	No	Yes	No	No
iSCSI	Yes	Yes	Yes	Yes	N/A
FibreChannel	Yes	Yes	Yes	No	N/A
Local Disk	Yes	Yes	Yes	No	Yes
HA	Yes	Yes (Native)	Yes	Yes	N/A
Snapshots of local disk	Yes	Yes	Yes	No	N/A
Local disk as data disk	No	No	No	No	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

Chapter 12. Amazon Web Services Compatible Interface

12.1. 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).

12.2. 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

12.3. 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 *Chapter 7, Setting 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

12.3.1. 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

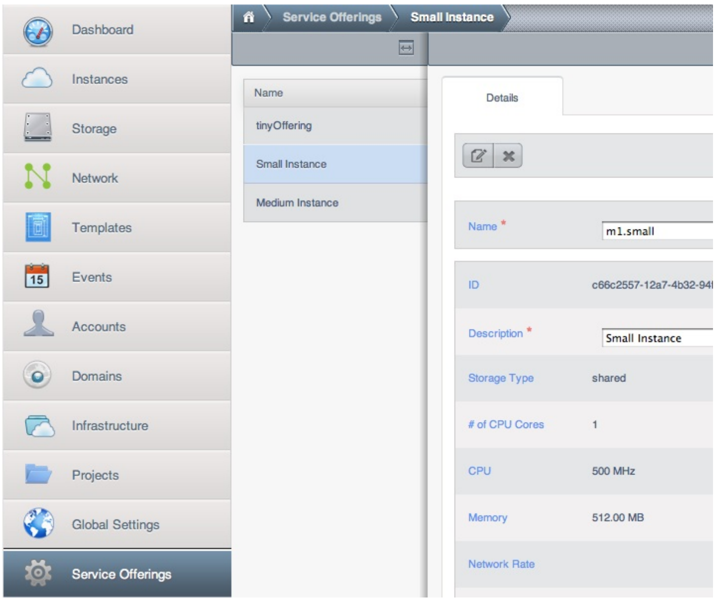
Using the CloudStack API, the easiest is to use the so-called integration port on which you can make unauthenticated calls. In Global Settings set the port to 8096 and subsequently call the *updateConfiguration* method. The following urls shows you how:

```
            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.

12.3.2. 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:



Name *	m1.small
ID	c66c2557-12a7-4b32-94f
Description *	Small Instance
Storage Type	shared
# of CPU Cores	1
CPU	500 MHz
Memory	512.00 MB
Network Rate	

12.3.3. 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:

a. Edit the files /etc/cloudstack/management/server.xml, /etc/cloudstack/management/server-nonssl.xml, and /etc/cloudstack/management/server-ssl.xml.

b. In each file, find the tag <Service name="Catalina7080">. Under this tag, locate <Connector executor="tomcatThreadPool-internal" port= ....<.

c. Change the port to whatever port you want to use, then save the files.

d. Restart the Management Server.

If you re-install CloudStack, you will have to re-enable the services and if need be update the port.

12.4. 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.

12.4.1. AWS API User Registration

Each user must perform a one-time registration. The user follows these steps:

1. Obtain the following by looking in the CloudStack UI, using the API, or asking the cloud administrator:
   - The CloudStack server's publicly available DNS name or IP address
   - The user account's Access key and Secret key
2. Generate a private key and a self-signed X.509 certificate. The user substitutes their own desired storage location for /path/to/… below.

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/path/to/private_key.pem -out /path/to/cert.pem
```

3. Register the user X.509 certificate and Access/Secret keys with the AWS compatible service. If you have the source code of CloudStack go to the awsapi-setup/setup directory and use the Python script cloudstack-aws-api-register. If you do not have the source then download the script using the following command.

```
wget -O cloudstack-aws-api-register "https://git-wip-us.apache.org/repos/asf?
p=cloudstack.git;a=blob_plain;f=awsapi-setup/setup/cloudstack-aws-api-
```

```
register;hb=4.1"
```

Then execute it, using the access and secret keys that were obtained in step 1. An example is shown below.

```
$ cloudstack-aws-api-register --apikey=User’s CloudStack API key --secretkey=User’s
CloudStack Secret key --cert=/path/to/cert.pem --
url=http://CloudStack.server:7080/awsapi
```

Note

A user with an existing AWS certificate could choose to use the same certificate with CloudStack, but note that the certificate would be uploaded to the CloudStack management server database.

12.4.2. 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
```

12.5. 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:

`--connection-timeout TIMEOUT`	Specifies a connection timeout (in seconds). Example: `--connection-timeout 30`
`--request-timeout TIMEOUT`	Specifies a request timeout (in seconds). Example: `--request-timeout 45`

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.

12.6. 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 12.1. 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 12.2. Availability Zone API mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-availability-zones	DescribeAvailabilityZones	listZones

Table 12.3. 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 12.4. 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 12.5. 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 12.6. Instance Attributes Mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-instance-attribute	DescribeInstanceAttribute	listVirtualMachines

Table 12.7. 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 12.8. Passwords API Mapping

EC2 command	SOAP call	CloudStack API call
ec2-get-password	GetPasswordData	getVMPassword

Table 12.9. 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 12.10. 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 12.11. 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

12.7. 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.

12.7.1. 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 12.1. 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")
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. Replace the Access and Secret keys with your own, as well as the endpoint of the service. Be sure to also update the file paths to something that exists on your machine.

Example 12.2. An S3 Boto Example

```
#!/usr/bin/env python

import sys
import os
from boto.s3.key import Key
from boto.s3.connection import S3Connection
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()
```

12.7.2. JClouds Examples

Chapter 13. 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.

13.1. 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

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.

13.2. 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	

13.3. 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.

13.3.1. 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.

13.3.2. 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.

13.4. 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.

Example Configurations

This section contains 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.

13.4.1. 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.

13.4.2. 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.

13.5. 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 Section 13.5.2, "External Guest Firewall Integration for Juniper SRX (Optional)".

13.5.1. 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.

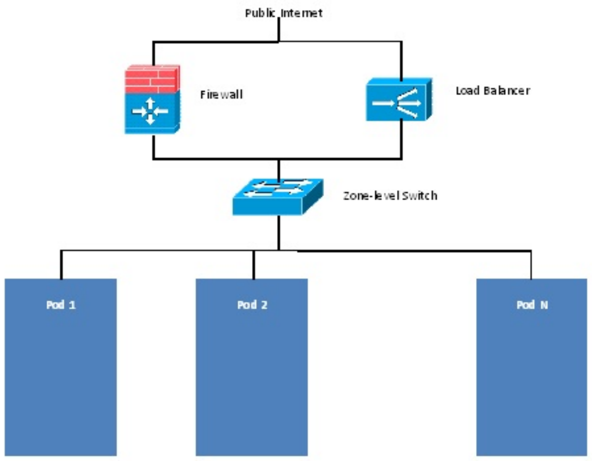
To achieve the above purposes you must set up fixed configurations for the firewall. Firewall rules and policies need not change as users are provisioned into the cloud. Any brand of hardware firewall that supports NAT and site-to-site VPN can be used.

13.5.2. 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 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. 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.

13.5.3. 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.

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

13.5.3.1.1. 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.

13.5.3.1.2. 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 Section 8.3.6, “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 Section 8.3.6, "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.

13.5.3.1.3. Using Cisco ASA 1000v Services

1. Ensure that all the prerequisites are met.

   See Section 13.5.3.1.2, "Prerequisites".

2. Add a VNMC instance.

   See Section 13.5.3.2, "Adding a VNMC Instance".

3. Add a ASA 1000v instance.

   See Section 13.5.3.3, "Adding an ASA 1000v Instance".

4. Create a Network Offering and use Cisco VNMC as the service provider for desired services.

   See Section 13.5.3.4, "Creating a Network Offering Using Cisco ASA 1000v".

5. Create an Isolated Guest Network by using the network offering you just created.

13.5.3.2. 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.

13.5.3.3. 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.

13.5.3.4. 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 Section 15.27.1, “About Virtual Private Clouds”.
    - **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.

13.5.3.5. 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
    ```

13.5.4. 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.

13.6. 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.

13.7. Topology Requirements

13.7.1. Security Requirements

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

13.7.2. 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.

13.7.3. 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.

13.7.4. 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.

13.7.5. Advanced Zone Topology Requirements

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

13.7.6. XenServer Topology Requirements

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

13.7.7. 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.

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.

13.7.8. KVM Topology Requirements

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

13.7.9. LXC Topology Requirements

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

13.8. 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 CloudStack API Developer's Guide.
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.

13.9. 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.

Chapter 14. Storage Setup

CloudStack is designed to work with a wide variety of commodity and enterprise-grade storage. Local disk may be used as well, if supported by the selected hypervisor. Storage type support for guest virtual disks differs based on hypervisor selection.

selection.

	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.

14.1. Small-Scale Setup

In a small-scale setup, a single NFS server can function as both primary and secondary storage. The NFS server just needs to export two separate shares, one for primary storage and the other for secondary storage.

14.2. Secondary Storage

CloudStack is designed to work with any scalable secondary storage system. The only requirement is the secondary storage system supports the NFS protocol.

> **Note**
>
> The storage server should be a machine with a large number of disks. The disks should ideally be managed by a hardware RAID controller. Modern hardware RAID controllers support hot plug functionality independent of the operating system so you can replace faulty disks without impacting the running operating system.

14.3. 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.

14.3.1. 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 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.

14.3.2. 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.

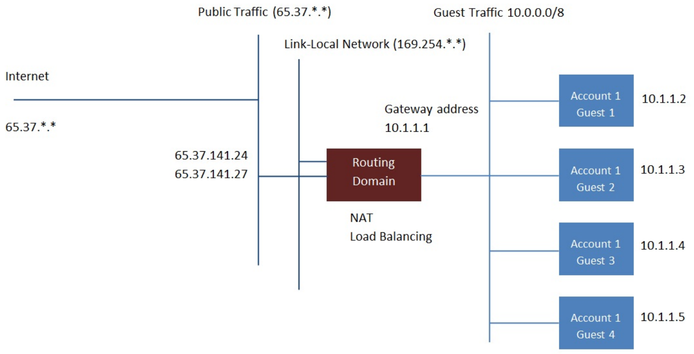
Chapter 15. Managing Networks and Traffic

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.

15.1. 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

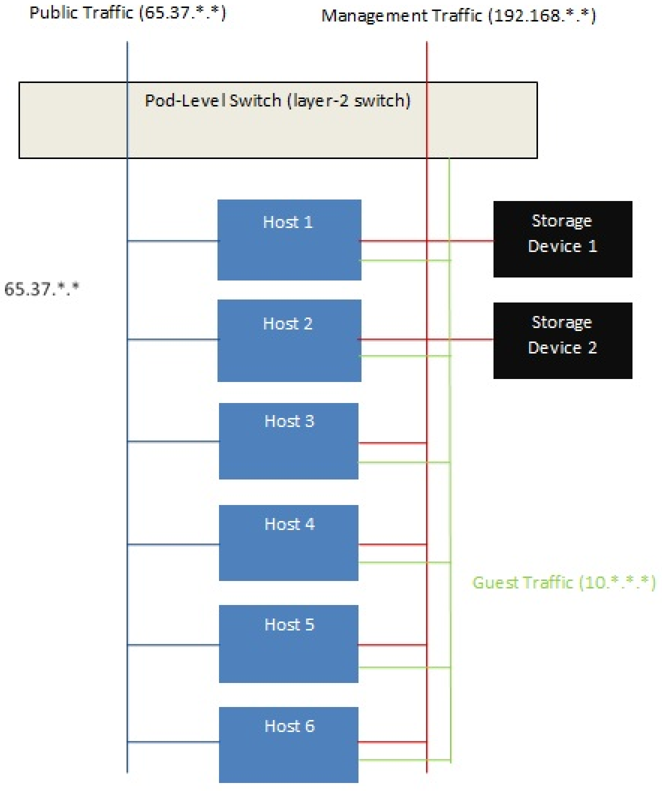
Typically, the Management Server automatically creates a virtual router for each network. A virtual router is a special virtual machine that runs on the hosts. Each virtual router in an isolated network has three network interfaces. If multiple public VLAN is used, the router will have multiple public 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 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

15.2. 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

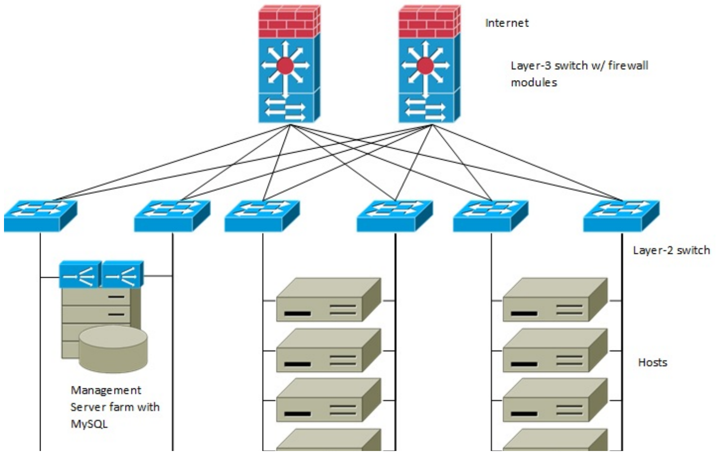
Servers are connected as follows:

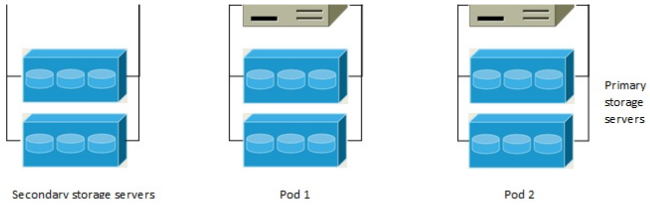
- Storage devices are connected to only the network that carries management traffic.
- Hosts are connected to networks for both management traffic and public traffic.
- Hosts are also connected to one or more networks carrying guest traffic.

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.

15.3. Networking in a Zone

The following figure illustrates the network setup within a single zone.


Internet
Layer-3 switch w/ firewall modules
Layer-2 switch
Hosts
Management Server farm with MySQL



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.

15.4. 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.

15.5. Advanced Zone Physical Network Configuration

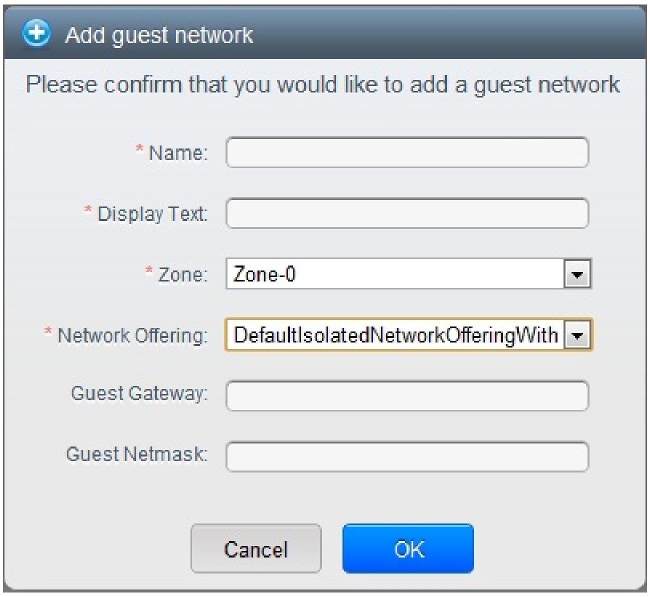
Within a zone that uses advanced networking, you need to tell the Management Server how the physical network is set up to carry different kinds of traffic in isolation.

15.5.1. 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:



Add guest network

Please confirm that you would like to add a guest network

* Name:

* Display Text:

* Zone: Zone-0

* Network Offering: DefaultIsolatedNetworkOfferingWith

Guest Gateway:

Guest Netmask:

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.

15.5.2. 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.

15.5.3. Configuring a Shared Guest Network

1. Log in to the CloudStack UI as administrator.

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.

15.6. 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.

15.6.1. 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.

15.6.2. 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 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.

15.6.2.1. Prerequisites

Ensure that vm-tools are running on guest VMs for adding or removing networks to work on VMware hypervisor.

15.6.2.2. 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)

15.6.2.3. 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.

15.6.2.4. 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.

15.6.3. 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.

15.7. 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.

15.7.1. 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.

15.7.2. 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.

15.7.3. 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.

15.7.4. 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.

15.8. 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.

15.8.1. 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:
   1. Locate the IP range you want to work with.
   2. Click Add Account button.

      The Add Account dialog is displayed.
   3. 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:
      1. Specify the following:
         - **Gateway**
         - **Netmask**
         - **VLAN**
         - **Start IP**
         - **End IP**
         - **Account**: Perform the following:
           1. Click Account.

              The Add Account page is displayed.
           2. Specify the following:

              Account: The account to which you want to assign an IP address range.

              Domain: The domain associated with the account.
           3. Click OK.
      2. Click Add.

15.8.2. 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.

15.9. 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.

15.9.1. Use Cases

15.9.1. 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.

15.9.2. Guidelines

To prevent IP conflict, configure different subnets when multiple networks are connected to the same VM.

15.9.3. 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.

15.9.4. 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.

15.10. 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.

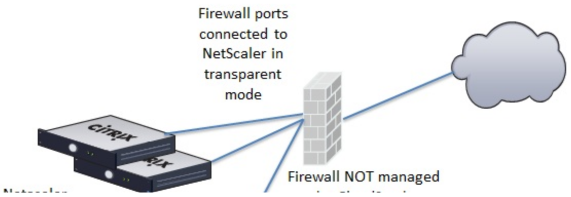
Use the **`deleteVlanRange`** API to delete IP ranges. This operation fails if an IP from the remove range is in use. If the remove range contains the IP address on which the DHCP server is running, CloudStack acquires a new IP from the same subnet. If no IP is available in the subnet, the remove operation fails.

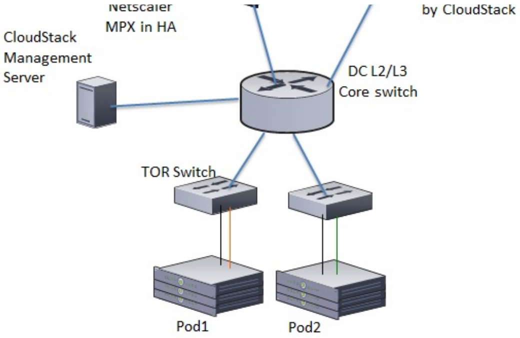
This feature is supported on KVM, xenServer, and VMware hypervisors.

15.11. About Elastic IP

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 the Administration Guide.

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.

15.12. Portable IPs

15.12.1. 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.

15.12.2. 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.

15.12.3. 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.

15.12.4. 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
```

15.13. 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.

15.13.1. Prerequisites and Guidelines

- This feature can only be implemented:

  on IPv4 addresses

  if virtual router is the DHCP provider

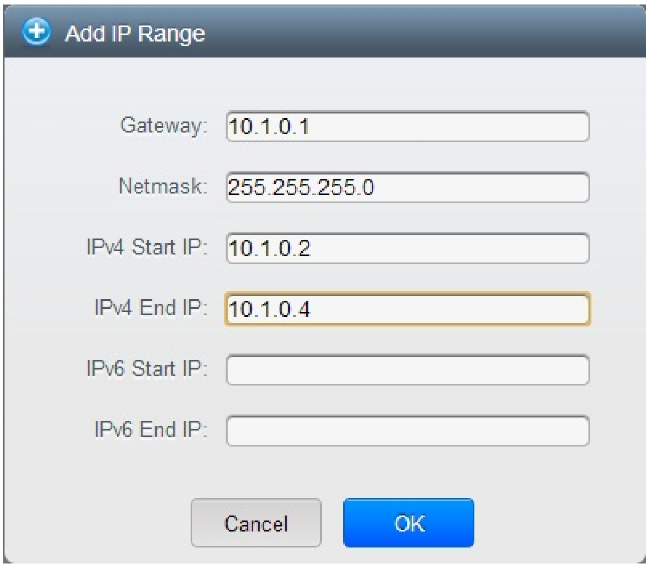
  on KVM, xenServer, and VMware hypervisors
- Manually configure the gateway of the new subnet before adding the IP range.
- CloudStack supports only one gateway for a subnet; overlapping subnets are not currently supported

15.13.2. 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:



Add IP Range

Gateway: 10.1.0.1

Netmask: 255.255.255.0

IPv4 Start IP: 10.1.0.2

IPv4 End IP: 10.1.0.4

IPv6 Start IP:

IPv6 End IP:

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.

15.14. 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.

15.14.1. 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)

15.14.2. Prerequisites

- Use a PVLAN supported switch.

  See Private VLAN Catalyst Switch Support Matrixfor 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.

15.14.3. 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 Section 15.14.1, “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.

15.15. Security Groups

15.15.1. About Security Groups

15.15.1. 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.

15.15.2. 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.

15.15.3. 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.

15.15.4. Enabling Security Groups

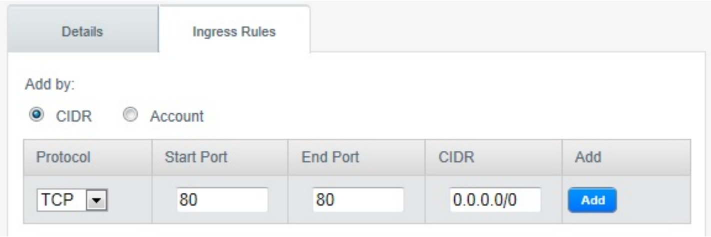
In order for security groups to function in a zone, the security groups feature must first be enabled for the zone. The administrator can do this when creating a new zone, by selecting a network offering that includes security groups. The procedure is described in Basic Zone Configuration in the Advanced Installation Guide. The administrator can not enable security groups for an existing zone, only when creating a new zone.

15.15.5. 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:



Add by: ● CIDR ○ Account

Protocol	Start Port	End Port	CIDR	Add
TCP	80	80	0.0.0.0/0	Add

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.

15.16. 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.

15.16.1. 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 Section 15.11, "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 Section 13.5.4, "External Guest Load Balancer Integration (Optional)".

The Citrix NetScaler comes in three varieties. The following table summarizes how these variants are treated in CloudStack.

CloudStack.

NetScaler ADC Type	Description of Capabilities	CloudStack Supported Features
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.

15.16.2. 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
   ```

15.16.3. Initial Setup of External Firewalls and Load Balancers

15.16.3. 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).

15.16.4. 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.

15.16.5. 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.

15.16.5.1. 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 Section 15.16.6, “Configuring AutoScale”.
   - **Health Check**: (Optional; NetScaler load balancers only) Click Configure and fill in the characteristics of the health check policy. See Section 15.16.5.3, “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.

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.

15.16.5.2. 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.

15.16.5.3. 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 Section 15.16.5.1, “Adding a Load Balancer Rule”.

15.16.6. 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 73.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

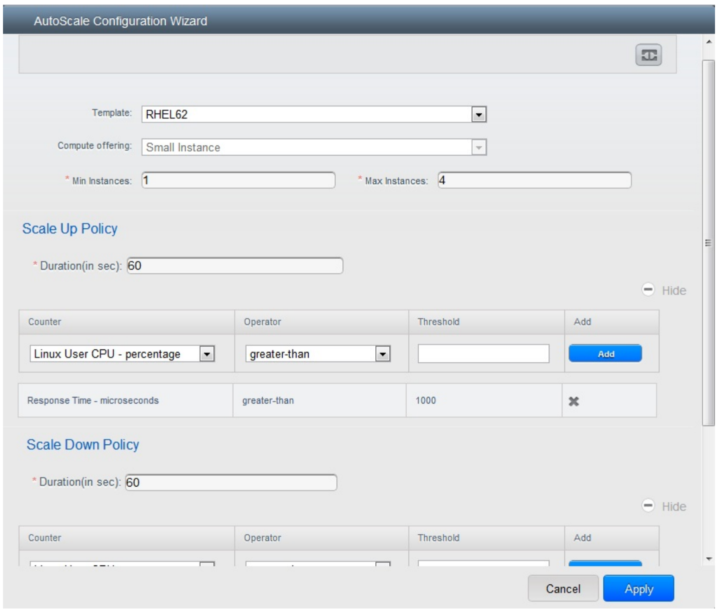
  If the application is not running, the NetScaler device considers the VM as ineffective and continues provisioning the VMs unconditionally until the resource limit is exhausted.

- 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 Section 15.16.2, “Configuring SNMP Community String on a RHEL Server” 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:



AutoScale Configuration Wizard

Template: RHEL62

Compute offering: Small Instance

* Min Instances: 1

* Max Instances: 4

Scale Up Policy

* Duration(in sec): 60

Counter	Operator	Threshold	Add
Linux User CPU - percentage	greater-than		Add
Response Time - microseconds	greater-than	1000	✖

Scale Down Policy

* Duration(in sec): 60

Counter	Operator	Threshold	Add

- **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.

15.17. 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.

15.17.1. 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.

15.17.1.1. Components of GSLB

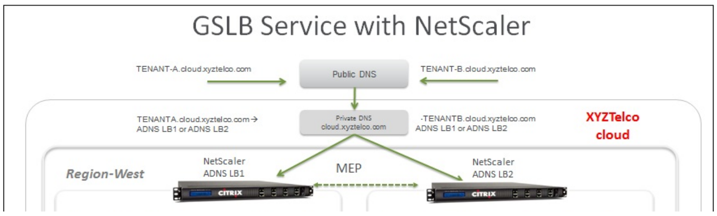
A typical GSLB environment is comprised of the following components:

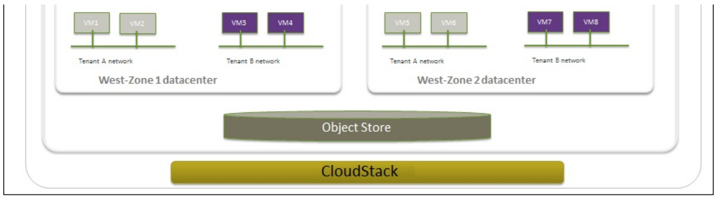
- **GSLB Site**: In CloudStack terminology, GSLB sites are represented by zones that are mapped to data centers, each of which has various network appliances. Each GSLB site is managed by a NetScaler appliance that is local to that site. Each of these appliances treats its own site as the local site 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.

15.17.1.2. 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.

15.17.2. 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 Section 15.17.2.2, “Enabling GSLB in NetScaler”.

As a domain administrator/ user perform the following:

1. Add a GSLB rule on both the sites.

   See Section 15.17.2.3, “Adding a GSLB Rule”.
2. Assign load balancer rules.

   See Section 15.17.2.4, “Assigning Load Balancing Rules to GSLB”.

15.17.2.1. 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.

15.17.2.2. 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.

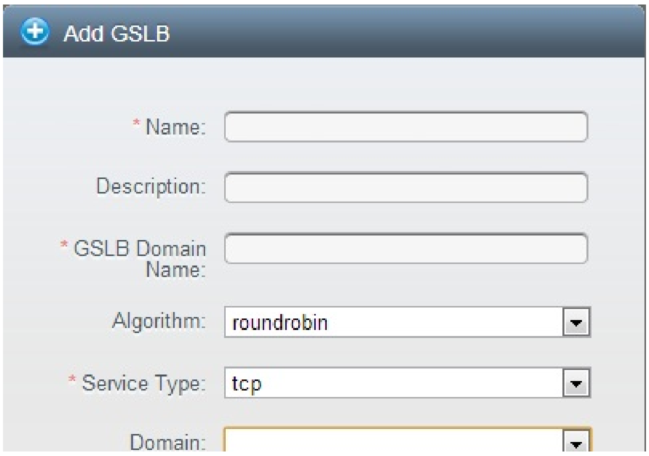
   You might have to scroll down to see this.
7. Click NetScaler.
8. Click Add NetScaler device and provide the following:

   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.

15.17.2.3. 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:



Add GSLB

* Name:

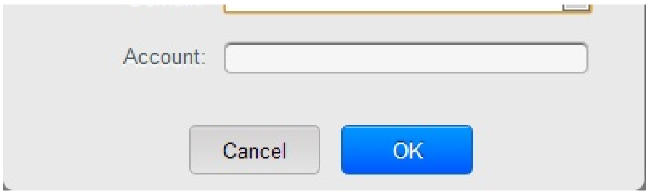
Description:

* GSLB Domain Name:

Algorithm: roundrobin

* Service Type: tcp

Domain:



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.

15.17.2.4. 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.

15.17.3. 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.

15.18. 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 Section 15.10, “About Multiple IP Ranges”.

15.19. 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 Section 15.12, “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.

15.20. 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. ✖

15.21. Static NAT

15.21. 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.

15.21.1. 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.

15.22. 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 Section 15.22.2, “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.

15.22.1. 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 Section 15.15.2, “Adding a Security Group”.

In an advanced zone, you can also create egress firewall rules by using the virtual router. For more information, see Section 15.22.2, “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.

15.22.2. 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.

15.22.2.1. Prerequisites and Guidelines

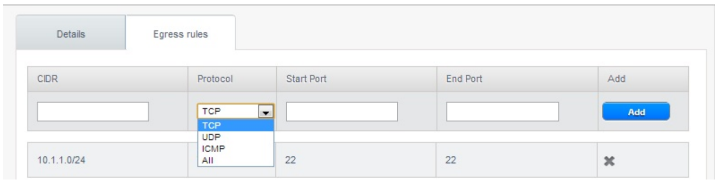
Consider the following scenarios to apply egress firewall rules:

- Egress firewall rules are supported on Juniper SRX and virtual router.

- 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.

15.22.2.2. 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	Protocol	Start Port	End Port	Add
	TCP			Add
10.1.1.0/24		22	22	✖

   - **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.

15.22.2.3. 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.

15.22.3. 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 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 Section 15.19, “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.

15.23. 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.

15.24. 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.

15.25. 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 Section 15.25.4, “Setting Up a Site-to-Site VPN Connection”

15.25.1. 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.

15.25.2. Using Remote Access VPN with Windows

The procedure to use VPN varies by Windows version. Generally, the user must edit the VPN properties and make sure that the default route is not the VPN. The following steps are for Windows L2TP clients on Windows Vista. The commands should be similar for other Windows versions.

1. Log in to the CloudStack UI and click on the source NAT IP for the account. The VPN tab should display the IPsec preshared key. Make a note of this and the source NAT IP. The UI also lists one or more users and their passwords. Choose one of these users, or, if none exists, add a user and password.
2. On the Windows box, go to Control Panel, then select Network and Sharing center. Click Setup a connection or network.
3. In the next dialog, select No, create a new connection.
4. In the next dialog, select Use my Internet Connection (VPN).
5. In the next dialog, enter the source NAT IP from step 1 and give the connection a name. Check Don't connect now.
6. In the next dialog, enter the user name and password selected in step 1.
7. Click Create.
8. Go back to the Control Panel and click Network Connections to see the new connection. The connection is not active yet.
9. Right-click the new connection and select Properties. In the Properties dialog, select the Networking tab.
10. In Type of VPN, choose L2TP IPsec VPN, then click IPsec settings. Select Use preshared key. Enter the preshared key from step 1.
11. The connection is ready for activation. Go back to Control Panel -> Network Connections and double-click the created connection.
12. Enter the user name and password from step 1.

15.25.3. Using Remote Access VPN with Mac OS X

First, be sure you've configured the VPN settings in your CloudStack install. This section is only concerned with connecting via Mac OS X to your VPN.

Note, these instructions were written on Mac OS X 10.7.5. They may differ slightly in older or newer releases of Mac OS X.

1. On your Mac, open System Preferences and click Network.
2. Make sure Send all traffic over VPN connection is not checked.
3. If your preferences are locked, you'll need to click the lock in the bottom left-hand corner to make any changes and provide your administrator credentials.
4. You will need to create a new network entry. Click the plus icon on the bottom left-hand side and you'll see a dialog that says "Select the interface and enter a name for the new service." Select VPN from the Interface drop-down menu, and "L2TP over IPSec" for the VPN Type. Enter whatever you like within the "Service Name" field.
5. You'll now have a new network interface with the name of whatever you put in the "Service Name" field. For the purposes of this example, we'll assume you've named it "CloudStack." Click on that interface and provide the IP address of the interface for your VPN under the Server Address field, and the user name for your VPN under Account Name.
6. Click Authentication Settings, and add the user's password under User Authentication and enter the pre-shared IPSec key in the Shared Secret field under Machine Authentication. Click OK.
7. You may also want to click the "Show VPN status in menu bar" but that's entirely optional.
8. Now click "Connect" and you will be connected to the CloudStack VPN.

15.25.4. 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. 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

Note

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 Section 15.27, "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.

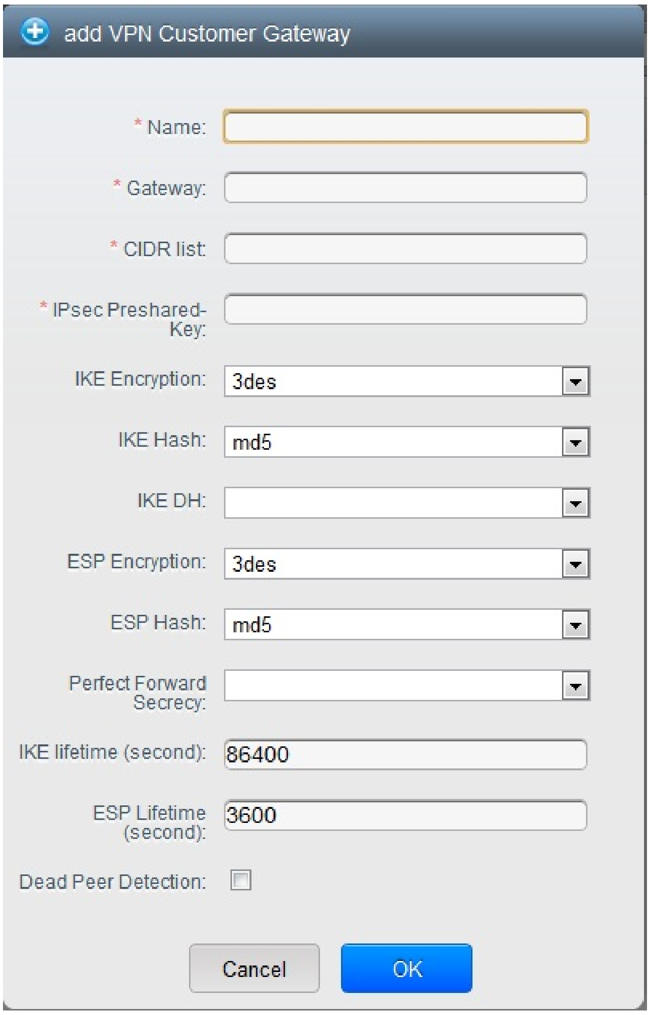
15.25.4.1. Creating and Updating a VPN Customer Gateway

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 site-to-site VPN.



add VPN Customer Gateway

* Name:	
* Gateway:	
* CIDR list:	
* IPsec Preshared-Key:	
IKE Encryption:	3des
IKE Hash:	md5
IKE DH:	
ESP Encryption:	3des
ESP Hash:	md5
Perfect Forward Secrecy:	
IKE lifetime (second):	86400
ESP Lifetime (second):	3600
Dead Peer Detection:	□

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.

> **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.

15.25.4.2. 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.

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.

   If you are creating the VPN gateway for the first time, selecting Site-to-Site VPN prompts you to create a VPN gateway.
7. 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

15.25.4.3. 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:

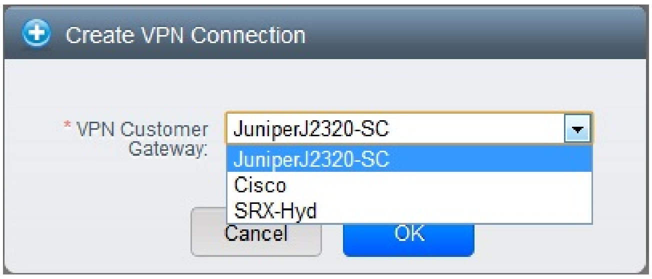
   - Internal LB
   - Public LB IP
   - Static NAT
   - Virtual Machines
   - CIDR

   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, then 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

15.25.4.4. 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. ↻

15.26. 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 Section 15.25.4.2, “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 Section 15.27.5, “Adding a Private Gateway to a VPC”.

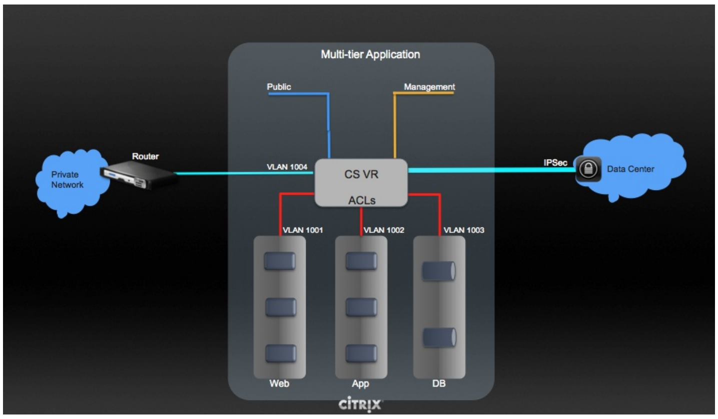
- Both administrators and users can create various possible destinations-gateway combinations. However, only one gateway of each type can be used in a deployment.

  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 Section 15.27, “Configuring a Virtual Private Cloud”.

15.27. Configuring a Virtual Private Cloud

15.27.1. 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 Section 15.27.5, “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 Section 15.25.4, “Setting Up a Site-to-Site VPN Connection”.
- **Customer Gateway**: The customer side of a VPN Connection. For more information, see Section 15.25.4.1, “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 Section 15.27.10, “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 Section 15.27.4, “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:

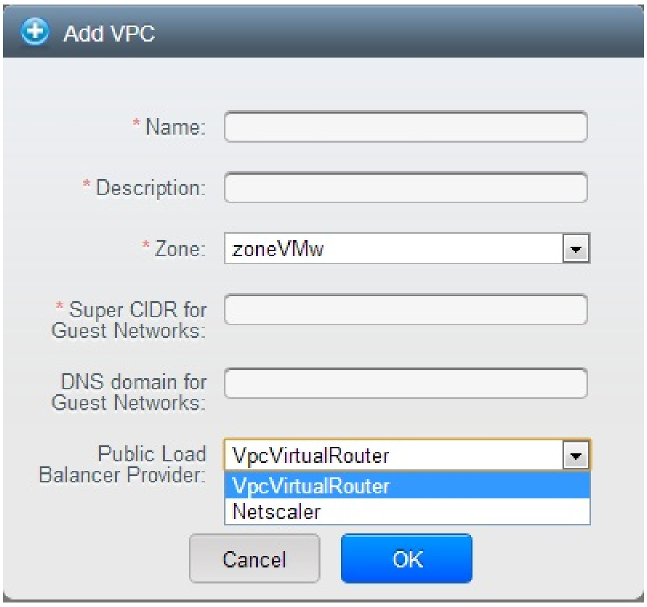
  That IP can't be used by more than one tier at a time in the VPC. For example, if you have tiers A and B, and a public IP1, you can create a port forwarding rule by using the IP either for A or B, but not for both.

  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.

15.27.2. 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:



Add VPC

* Name:

* Description:

* Zone: zoneVMw

* Super CIDR for Guest Networks:

DNS domain for Guest Networks:

Public Load Balancer Provider: VpcVirtualRouter

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.

15.27.3. 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.

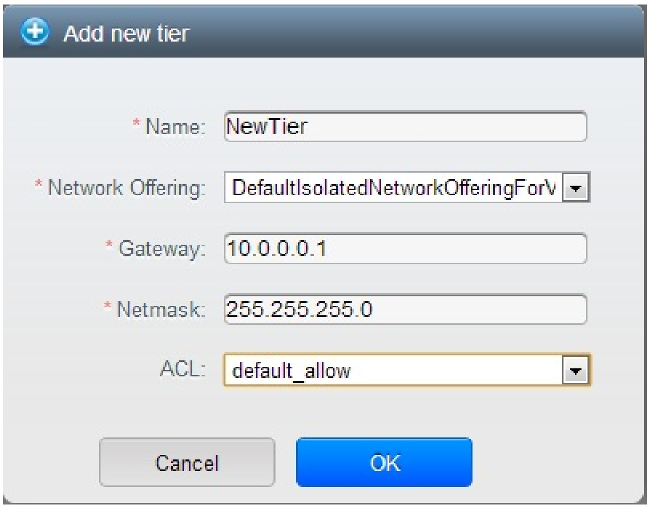
1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network.
3. In the Select view, select VPC.

   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:



Add new tier

* Name: NewTier

* Network Offering: DefaultIsolatedNetworkOfferingForV

* Gateway: 10.0.0.0.1

* Netmask: 255.255.255.0

ACL: default_allow

   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 the Assigning VLANs to Isolated Networks section in the CloudStack Administration Guide.
   - **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.

15.27.4. Configuring Network Access Control List

Define Network Access Control List (ACL) on the VPC virtual router to control incoming (ingress) and outgoing (egress)

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.

15.27.4.1. 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

15.27.4.2. 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.

15.27.4.3. 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.

- **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.

15.27.4.4. 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.

15.27.4.5. 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.

15.27.5. 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.

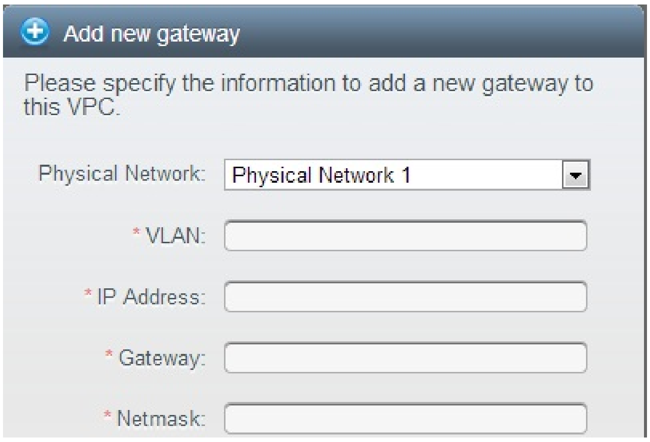
   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to configure load balancing rules.

   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:



Add new gateway

Please specify the information to add a new gateway to this VPC.

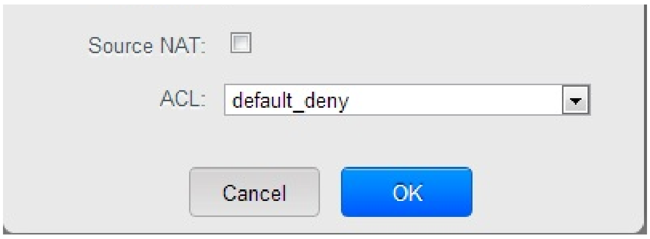
Physical Network: Physical Network 1

* VLAN:

* IP Address:

* Gateway:

* Netmask:



Source NAT: □

ACL: default_deny

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 Section 15.27.5.1, "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 Section 15.27.5.2, "ACL on Private Gateway".

   The new gateway appears in the list. You can repeat these steps to add more gateway for this VPC.

15.27.5.1. 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.

15.27.5.2. 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.

15.27.5.3. 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.

15.27.5.4. 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.

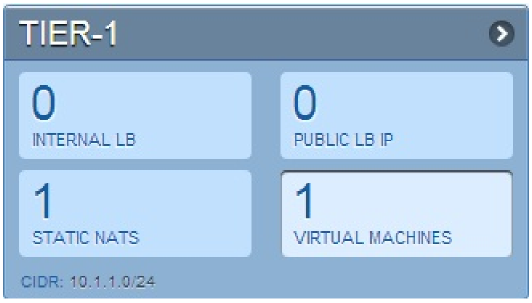
15.27.6. Deploying VMs to the Tier

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.

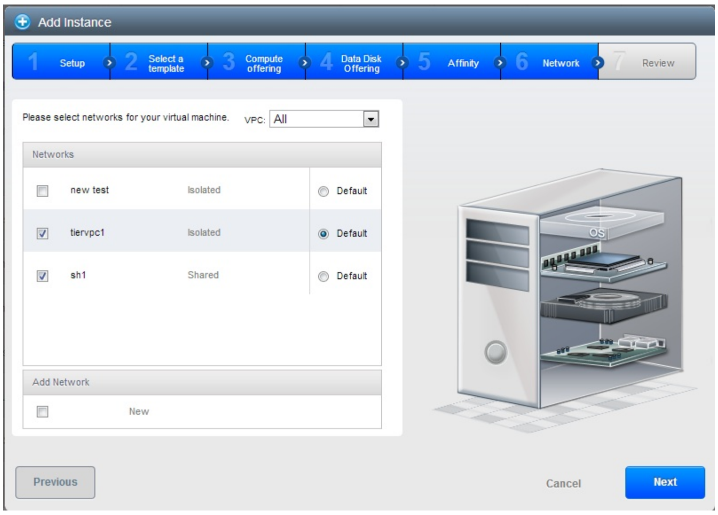
   Follow the on-screen instruction to add an instance. For information on adding an instance, see the Installation Guide.

15.27.7. 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.



Please select networks for your virtual machine. VPC: All

Networks

□	new test	Isolated	○ Default
☑	tiervpc1	Isolated	● Default
☑	sh1	Shared	○ Default

Add Network

□ New

8. Click Next, review the configuration and click Launch.

   Your VM will be deployed to the selected VPC tier and shared network.

15.27.8. 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

   - 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.

15.27.9. 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 ✖

15.27.10. 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

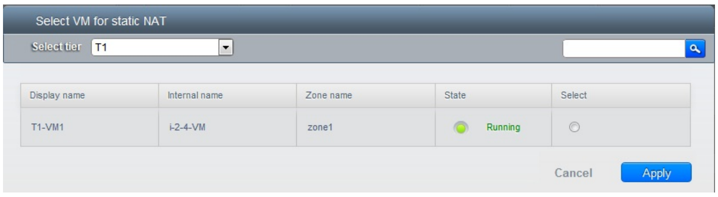
   The following router information is displayed:

   - 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:



Select VM for static NAT

Select tier: T1

Display name	Internal name	Zone name	State	Select
T1-VM1	i-2-4-VM	zone1	● Running	○

9. Select the tier and the destination VM, then click Apply.

15.27.11. 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.

15.27.11.1. 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.

15.27.11.1.1. 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 Section 15.27.11.1.2, “Creating a Network Offering for External LB”.
3. Create a VPC with Netscaler as the Public LB provider.

   For more information, see Section 15.27.2, “Adding a Virtual Private Cloud”.
4. For the VPC, acquire an IP.
5. Create an external load balancing rule and apply, as given in Section 15.27.11.1.3, “Creating an External LB Rule”.

15.27.11.1.2. 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 Section 15.27.1, “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. 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.

15.27.11.1.3. 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.

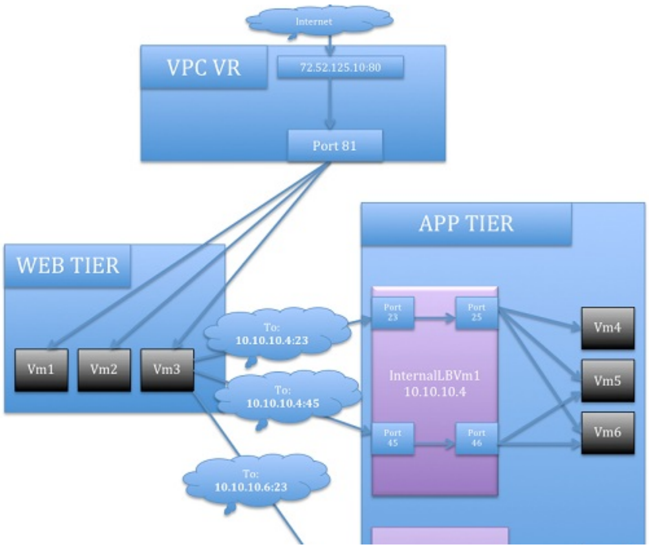
The new load balancing rule appears in the list. You can repeat these steps to add more load balancing rules for this IP address.

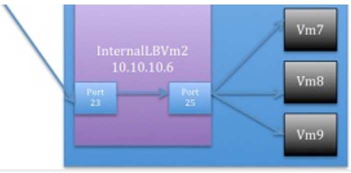
15.27.11.2. Load Balancing Across Tiers

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 Section 15.27.11, “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.

15.27.11.2.1. 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.





15.27.11.2.2. 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.

15.27.11.2.3. Enabling Internal LB on a VPC Tier

1. Create a network offering, as given in Section 15.27.11.2.5, “Creating an Internal LB Rule”.
2. Create an internal load balancing rule and apply, as given in Section 15.27.11.2.5, “Creating an Internal LB Rule”.

15.27.11.2.4. 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 Section 15.27.1, “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.

15.27.11.2.5. 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

15.27.12. 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.

15.27.13. 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.

15.27.14. 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.

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. ↻

15.28. 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.

15.28.1. 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.

15.28.2. Creating a Persistent Guest Network

To create a persistent network, perform the following:

1. Create a network offering with the Persistent option enabled.

   See the Administration Guide.
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.

Chapter 16. 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.

16.1. 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.

16.2. 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.

16.3. 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.

Revision History

Revision 1-0 **October 5 2012** **Jessica Tomechak, Radhika PC, Wido den Hollander**

Initial publication