Apache CloudStack 4.0.0-incubating

CloudStack Installation Guide

Apache CloudStack

Apache CloudStack 4.0.0-incubating CloudStack Installation Guide

Author Apache CloudStack

Apache CloudStack is an effort undergoing incubation at The Apache Software Foundation (ASF).

Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

Installation Guide for CloudStack.

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 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*[1] and *Apache CloudStack API Reference*[2] 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

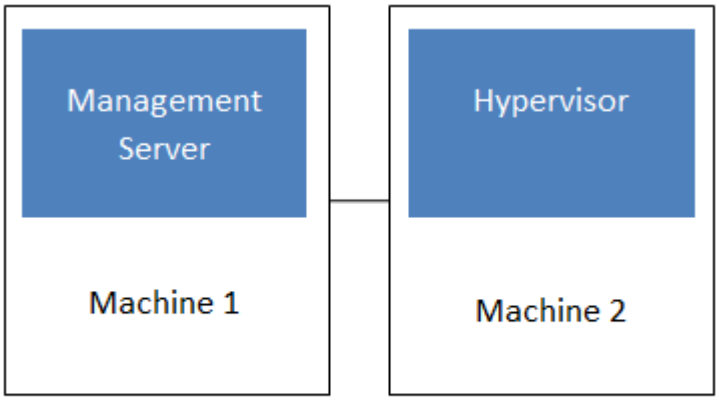
CloudStack has a number of features to increase the availability of the system. The Management Server itself may be deployed in a multi-node installation where the servers are load balanced. MySQL may be configured to use replication to provide for a manual failover in the event of database loss. For 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).

[1] http://incubator.apache.org/cloudstack/docs
[2] http://incubator.apache.org/cloudstack/docs/api



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 Choosing a Deployment Architecture.

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:

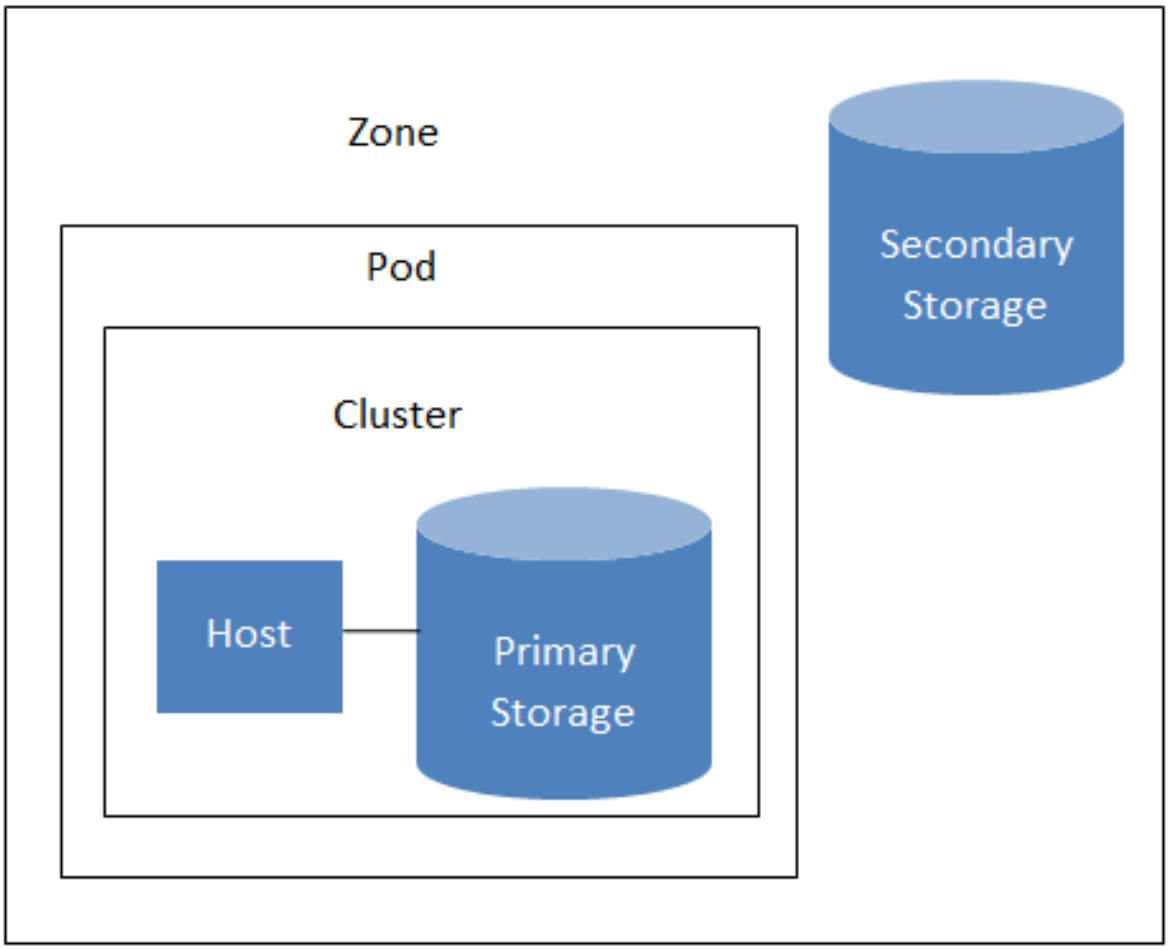
- Provides the web user interface for the administrator and a reference user interface for end users.
- Provides the APIs for CloudStack.
- Manages the assignment of guest VMs to particular hosts.
- Manages the assignment of public and private IP addresses to particular accounts.
- Manages the allocation 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.

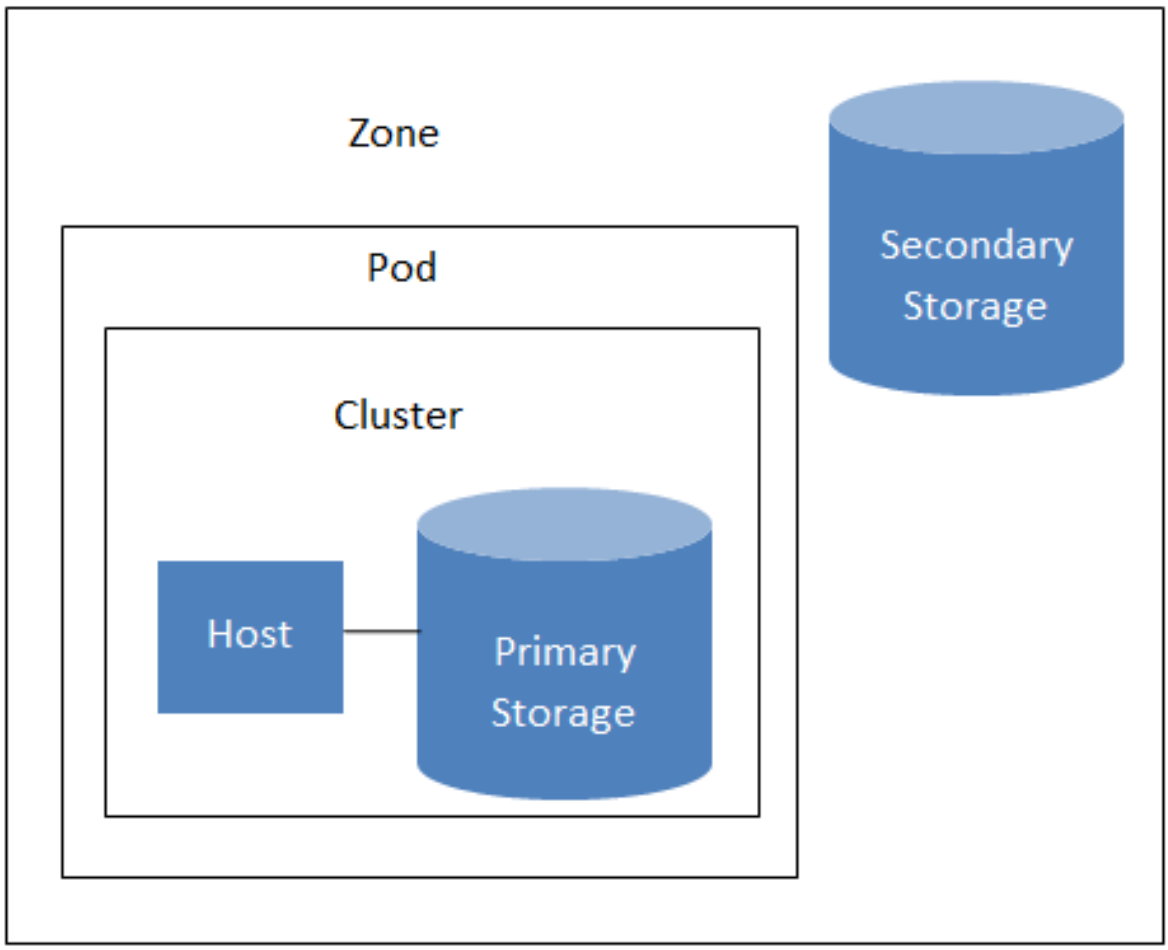
Cloud Infrastructure Concepts

2.1. About Zones

A zone is the 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.
- 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 a zone.
- How many clusters to place in each pod.
- How many hosts to place in each cluster.
- How many primary storage servers to place in each cluster and total capacity for the storage servers.
- How much secondary storage to deploy in 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.

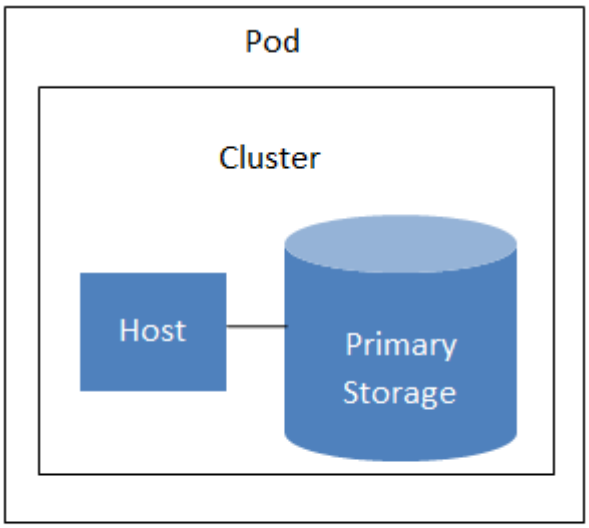
2.2. About Pods

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.

Pods are not visible to the end user.

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



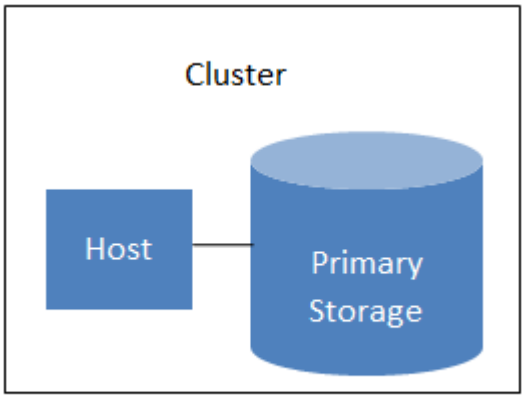
A simple pod

2.3. 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. Administrator must register the vCenter server with CloudStack. There may be multiple vCenter servers per zone. Each vCenter server may manage multiple VMware clusters.

2.4. 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

2.5. About Primary Storage

Primary storage is associated with a cluster, and it stores the disk volumes for all the VMs running on hosts in that cluster. You can add multiple primary storage servers to a cluster. At least one is required. It is typically located close to the hosts for increased performance.

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

- 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.

2.6. About Secondary Storage

Secondary storage is associated with a zone, and it 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 zone-based NFS secondary storage are available to all hosts in the zone. CloudStack manages the allocation of guest virtual disks to particular primary storage devices.

To make items in secondary storage available to all hosts throughout the cloud, you can add OpenStack Object Storage (Swift, *swift.openstack.org*[1]) in addition to the zone-based NFS secondary storage. When using Swift, you configure Swift storage for the entire CloudStack, then set up NFS secondary storage for each zone as usual. 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. The Swift storage acts as a cloud-wide resource, making templates and other data available to any zone in the cloud. There is no hierarchy in the Swift storage, just one Swift container per storage

[1] http://swift.openstack.org

object. Any secondary storage in the whole cloud can pull a container from Swift at need. 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.

2.7. About Physical Networks

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

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

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

2.7.1. Configurable Characteristics of Physical Networks

CloudStack provides configuration settings you can use to set up a physical network in a zone, including:

- What type of network traffic it carries (guest, public, management, storage)
- VLANs
- Unique name that the hypervisor can use to find that particular network
- Enabled or disabled. When a network is first set up, it is disabled – not in use yet. The administrator sets the physical network to enabled, and it begins to be used. The administrator can later disable the network again, which prevents any new virtual networks from being created on that physical network; the existing network traffic continues even though the state is disabled.
- Speed
- Tags, so network offerings can be matched to physical networks
- Isolation method

2.7.2. 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. 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.

2.7.3. 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.

2.7.4. 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. 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.

2.7.5. 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.

2.7.6. 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.

2.7.7. 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 Working with System Virtual Machines 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.

Building from Source

The official CloudStack release is always in source code form. While there may exist convenience binaries in various forms from a number of places, the source is the canonical release will be source. In this document we'll cover acquiring the source release, building that into binary, deployable packages.

While building and deploying directly from source is certainly possible, the reality of Infrastructure-as-a-Service cloud computing implies a need to deploy packages on a potentially large number of systems, which RPMs and DEBs fill nicely.

Building and deploying directly from source is thus outside the scope of this document, but is documented in the INSTALL.md file in the release.

3.1. Getting the release

You can download the latest CloudStack release from the *Apache CloudStack project download page*[1].

You'll notice several links under the 'Latest release' section.

1. *apache-cloudstack-4.0.0-incubating-src.tar.bz2*[2] - This is the link to the release itself.
2. *PGP*[3] - This is a detached cryptographic signature that can be used to help verify the authenticity of the release.
3. *MD5*[4] - An MD5 hash of the release to aid in verify the validity of the release download.
4. *SHA512*[5] - 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*[6] file.

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

```
# 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:

[1] http://incubator.apache.org/cloudstack/downloads.html
[2] http://www.apache.org/dyn/closer.cgi/dist/incubator/cloudstack/releases/4.0.0-incubating/apache-cloudstack-4.0.0-incubating-src.tar.bz2
[3] http://www.apache.org/dist/incubator/cloudstack/releases/4.0.0-incubating/apache-cloudstack-4.0.0-incubating-src.tar.bz2.asc
[4] http://www.apache.org/dist/incubator/cloudstack/releases/4.0.0-incubating/apache-cloudstack-4.0.0-incubating-src.tar.bz2.md5
[5] http://www.apache.org/dist/incubator/cloudstack/releases/4.0.0-incubating/apache-cloudstack-4.0.0-incubating-src.tar.bz2.sha
[6] http://www.apache.org/dist/dev/incubator/cloudstack/KEYS

```
$ 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.

The minimum bootstrapped prerequisites for building CloudStack includes the following:

1. ant
2. maven (version 3)
3. Java (Java 6/OpenJDK 1.6)
4. 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-4.0.0-incubating-src.tar.bz2
```

You can now move into the directory:

```
$ cd ./apache-cloudstack-4.0.0-incubating-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-buildpackge -uc -us
```

This command will build 16 Debian packages. You should have all of the following:

```
cloud-agent_4.0.0-incubating_amd64.deb
cloud-agent-deps_4.0.0-incubating_amd64.deb
cloud-agent-libs_4.0.0-incubating_amd64.deb
cloud-awsapi_4.0.0-incubating_amd64.deb
cloud-cli_4.0.0-incubating_amd64.deb
cloud-client_4.0.0-incubating_amd64.deb
cloud-client-ui_4.0.0-incubating_amd64.deb
cloud-core_4.0.0-incubating_amd64.deb
cloud-deps_4.0.0-incubating_amd64.deb
cloud-python_4.0.0-incubating_amd64.deb
cloud-scripts_4.0.0-incubating_amd64.deb
cloud-server_4.0.0-incubating_amd64.deb
cloud-setup_4.0.0-incubating_amd64.deb
cloud-system-iso_4.0.0-incubating_amd64.deb
cloud-usage_4.0.0-incubating_amd64.deb
cloud-utils_4.0.0-incubating_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 ./
```

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

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 the following command:

```
$ mvn -P deps
```

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

```
$ ./waf rpm
```

Once this completes, you should find assembled RPMs in **artifacts/rpmbuild/RPMS/x86_64**

3.6.1. Creating a yum repo

While RPMs is an ideal packaging format - it's most easily consumed from yum repositories over a network. We'll move into the directory with the newly created RPMs by issuing the following command:

```
$ cd artifacts/rpmbuild/RPMS/x86_64
```

Next we'll issue a command to create the repository metadata by issuing the following command:

```
$ createrepo ./
```

The files and directories within our current working directory can now be uploaded to a web server and serve as a yum repository

3.6.2. Configuring your systems to use your new yum repository

Now that your yum repository is populated with RPMs and metadata we need to configure our machines that need to install CloudStack. We will create a file at **`/etc/yum.repos.d/cloudstack.repo`** with the following content:

```
[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.

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 6.2, “Adding a Zone”*
5. Add more pods (optional). See *Section 6.3, “Adding a Pod”*
6. Add more clusters (optional). See *Section 6.4, “Adding a Cluster”*
7. Add more hosts (optional). See *Section 6.5, “Adding a Host”*
8. Add more primary storage (optional). See *Section 6.6, “Add Primary Storage”*
9. Add more secondary storage (optional). See *Section 6.7, “Add Secondary Storage”*
10. Try using the cloud. See *Section 6.8, “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
- 50 GB of local disk (When running secondary storage on the management server 500GB is 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 homogenous. 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.

- *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"*

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"* or *Section 3.5, "Building DEB packages"* you may find pre-built DEB and RPM packages for your convience linked from the *downloads*[1] 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 4.0
```

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

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

[1] http://incubator.apache.org/cloudstack/downloads.html

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/4.0/
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.
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 "managament1.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. 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*[2].

If the Management Server is RHEL or CentOS, copy vhd-util to /usr/lib64/cloud/common/scripts/vm/hypervisor/xenserver.

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

4.5.3.2. Install on CentOS/RHEL

We start by installing the required packages:

```
yum install cloud-client
```

4.5.3.3. Install on Ubuntu

```
apt-get install cloud-client
```

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

[2] http://download.cloud.com.s3.amazonaws.com/tools/vhd-util

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 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 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'
```

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 mysqld restart
```

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

Warning

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

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

```
mysql_secure_installation
```

5. 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”*.

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

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

6. 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
   ```

   Note

   This type of single-machine setup is recommended only for a trial installation.

7. 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.

   ```
   # cloud-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 mysqld 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”*.

```
cloud-setup-databases cloud:<dbpassword>@<ip address mysql server> \
--deploy-as=root:<password> \
-e <encryption_type> \
-m <management_server_key> \
-k <database_key>
```

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 new 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 (cloud-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. 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 a Ceph RDB cluster, GlusterFS, iSCSI, and otthers.

The requirements for primary and secondary storage are described in:

- *Section 2.5, “About Primary Storage”*
- *Section 2.6, “About Secondary Storage”*

A production installation typically uses a separate NFS server. See *Section 4.5.6.1, “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 4.5.6.2, “Using the Management Server as the NFS Server”*.

4.5.6.1. 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
```

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

```
# vi /etc/exports
```

   Insert the following line.

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

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
```

4.5.6.2. 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 Ubuntu/Debian systems, you'll need to install the nfs-kernel-server package:

```
$ sudo apt-get install nfs-kernel-server
```

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. For example:

```
# vi /etc/exports
```

Insert the following line.

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

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
```

4.5.7. 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"* 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*[3]

   If the Management Server is RHEL or CentOS, copy vhd-util to /usr/lib64/cloud/common/scripts/vm/hypervisor/xenserver.

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

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"*.)

   ```
   # cloud-setup-databases cloud:dbpassword@dbhost -e encryption_type -
   m management_server_key -k database_key
   ```

5. Configure the OS and start the Management Server:

[3] http://download.cloud.com.s3.amazonaws.com/tools/vhd-util

```
# cloud-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 Management Server Load Balancing.

4.5.8. 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://download.cloud.com/templates/acton/acton-systemvm-02062012.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://download.cloud.com/templates/burbank/burbank-systemvm-08012012.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://download.cloud.com/templates/acton/acton-
     systemvm-02062012.qcow2.bz2 -h kvm -s <optional-management-server-secret-key> -F
     ```

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

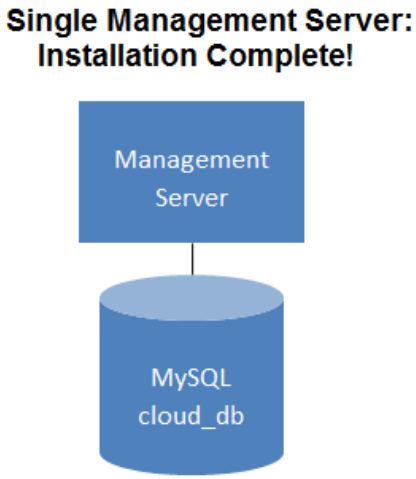
   When the script has finished, unmount secondary storage and remove the created directory.

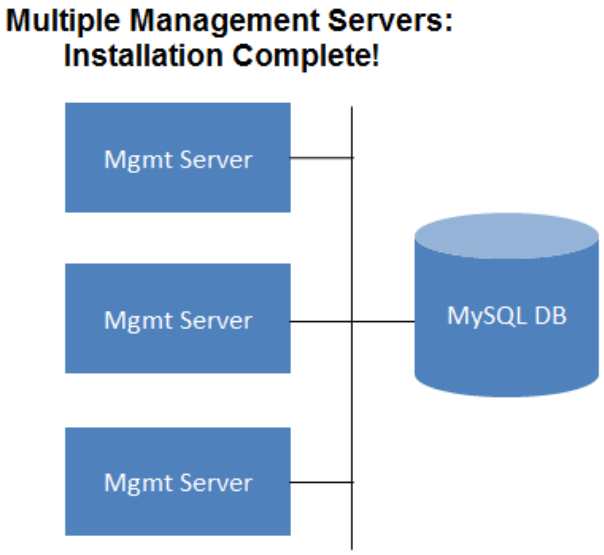
   ```
   # umount /mnt/secondary
   # rmdir /mnt/secondary
   ```

3. Repeat these steps for each secondary storage server.

4.5.9. 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.

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 Script*[1] to the instance you have created.

```
wget http://downloads.sourceforge.net/project/cloudstack/SSH%20Key%20Gen%20Script/cloud-
set-guest-sshkey.in?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fcloudstack%2Ffiles%2FSSH
%2520Key%2520Gen%2520Script%2F&ts=1331225219&use_mirror=iweb
```

3. Copy the file to /etc/init.d.

```
cp cloud-set-guest-sshkey.in /etc/init.d/
```

4. Give the necessary permissions on the script:

```
chmod +x /etc/init.d/cloud-set-guest-sshkey.in
```

[1] http://sourceforge.net/projects/cloudstack/files/SSH%20Key%20Gen%20Script/

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 numbet>/?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.

Steps to Provisioning Your Cloud Infrastructure

This section tells how to add 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*.

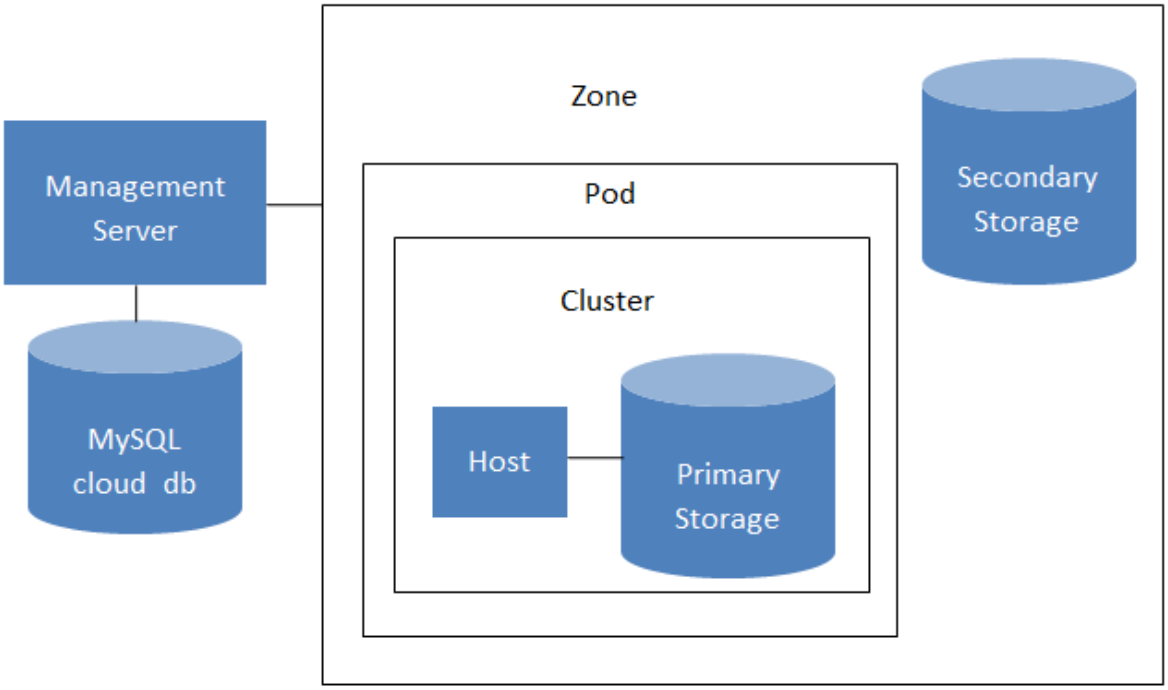
6.1. Overview of Provisioning Steps

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. Change the root password. See *Section 5.1.4, "Changing the Root Password"*.
2. Add a zone. See *Section 6.2, "Adding a Zone"*.
3. Add more pods (optional). See *Section 6.3, "Adding a Pod"*.
4. Add more clusters (optional). See *Section 6.4, "Adding a Cluster"*.
5. Add more hosts (optional). See *Section 6.5, "Adding a Host"*.
6. Add primary storage. See *Section 6.6, "Add Primary Storage"*.
7. Add secondary storage. See *Section 6.7, "Add Secondary Storage"*.
8. Initialize and test the new cloud. See *Section 6.8, "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 a Zone

These steps assume you have already logged in to the CloudStack UI. See *Section 5.1, "Log In to the UI"*.

1. (Optional) If you are going to use Swift for cloud-wide secondary storage, you need to add it before you add zones.
   a. Log in to the CloudStack UI as administrator.
   b. If this is your first time visiting the UI, you will see the guided tour splash screen. Choose "Experienced user." The Dashboard appears.
   c. In the left navigation bar, click Global Settings.
   d. In the search box, type swift.enable and click the search button.
   e. Click the edit button and set swift.enable to true. 
   f. Restart the Management Server.

      ```
      # service cloud-management restart
      ```

   g. Refresh the CloudStack UI browser tab and log back in.
2. In the left navigation, choose Infrastructure.
3. On Zones, click View More.

4. (Optional) If you are using Swift storage, click Enable Swift. Provide the following:
   - **URL.** The Swift URL.
   - **Account.** The Swift account.
   - **Username.** The Swift account's username.
   - **Key.** The Swift key.
5. Click Add Zone. The zone creation wizard will appear.
6. 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.

   For more information about the network types, see Network Setup.
7. The rest of the steps differ depending on whether you chose Basic or Advanced. Continue with the steps that apply to you:
   - *Section 6.2.1, "Basic Zone Configuration"*
   - *Section 6.2.2, "Advanced Zone Configuration"*

6.2.1. 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. (Introduced in version 3.0.1) 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 2.2, “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.

6.2.2. 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 2.7.4, “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.

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 2.2, "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 2.7.7, "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 2.3, "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 2.4, "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 2.5, "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 2.6, "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.
    - **Path.** The exported path from the server.

12. Click Launch.

6.3. 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.

6.4. 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.

6.4.1. 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.

6.4.2. Add Cluster: vSphere

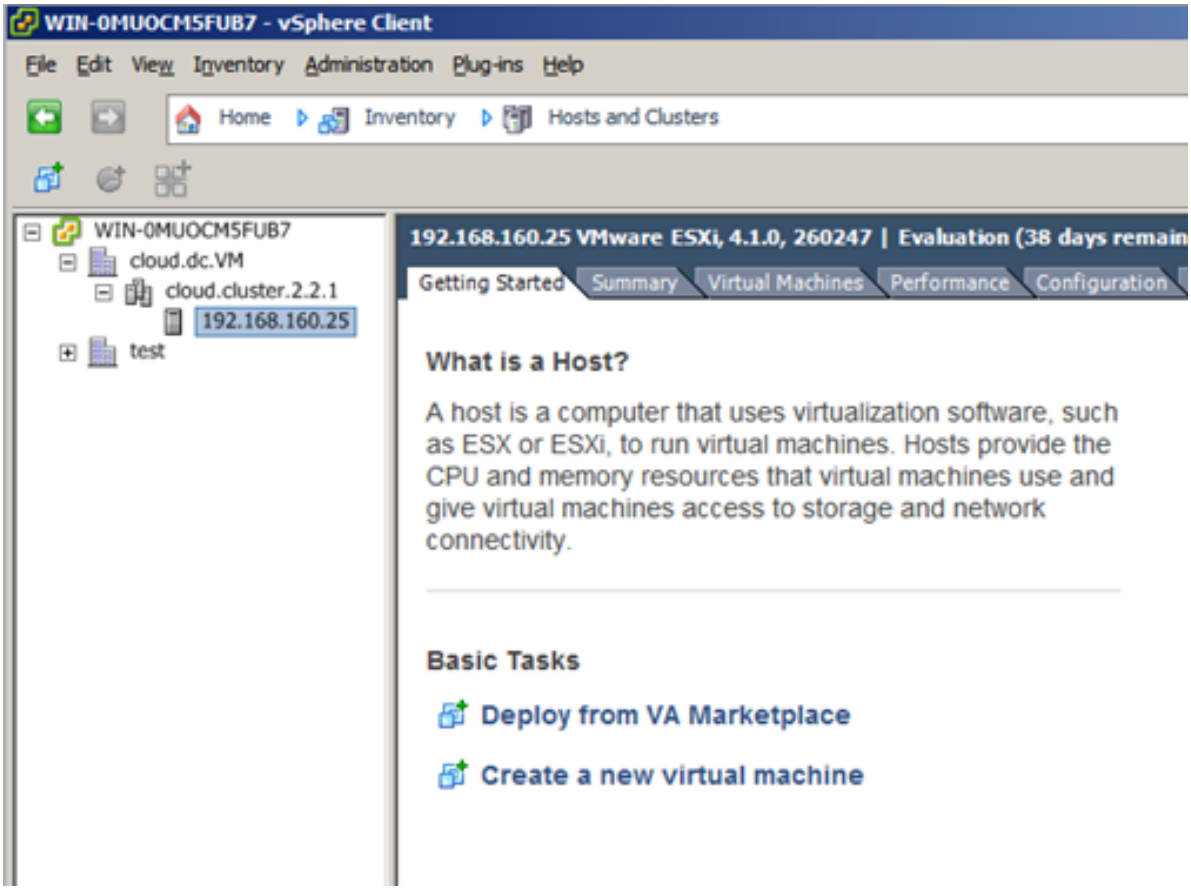
Host management for vSphere is done through a combination of vCenter and the CloudStack admin UI. CloudStack requires that all hosts be in a CloudStack cluster, but the cluster may consist of a single host. As an administrator you must decide if you would like to use clusters of one host or of multiple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

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

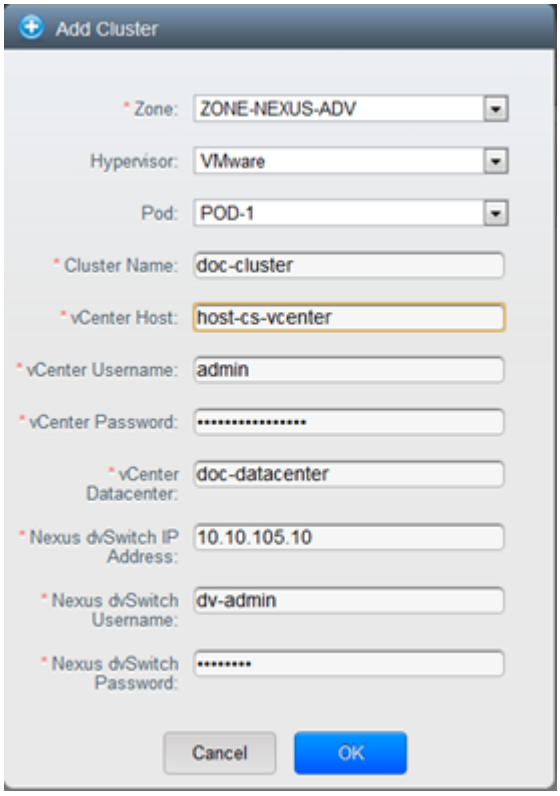
- Do not put more than 8 hosts in a vSphere cluster
- Make sure the hypervisor hosts do not have any VMs already running before you add them to CloudStack.

To add a vSphere cluster to CloudStack:

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



2. Log in to the UI.
3. In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone in which you want to add the cluster.
4. Click the Compute tab, and click View All on Pods. Choose the pod to which you want to add the cluster.
5. Click View Clusters.
6. Click Add Cluster.
7. In Hypervisor, choose VMware.
8. Provide the following information in the dialog. The fields below make reference to values from vCenter.
    - Cluster Name. Enter the name of the cluster you created in vCenter. For example, "cloud.cluster.2.2.1"
    - vCenter Host. Enter the hostname or IP address of the vCenter server.
    - vCenter Username. 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".

- 

  There might be a slight delay while the cluster is provisioned. It will automatically display in the UI

6.5. 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 6.5.1, “Adding a Host (XenServer or KVM)”*
   - *Section 6.5.2, “Adding a Host (vSphere)”*

6.5.1. Adding a Host (XenServer or KVM)

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

6.5.1.1. 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.

6.5.1.1.1. 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/lib64/cloud/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
```

6.5.1.1.2. 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.

6.5.1.2. Adding a XenServer or KVM Host

- 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.
- Log in to the CloudStack UI as administrator.
- In the left navigation, choose Infrastructure. In Zones, click View More, then click the zone in which you want to add the host.
- Click the Compute tab. In the Clusters node, click View All.
- Click the cluster where you want to add the host.
- Click View Hosts.
- Click Add Host.
- 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.
- Repeat for additional hosts.

6.5.2. 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.

6.6. Add Primary Storage

6.6.1. 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.

6.6.2. Adding Primary Stroage

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

Be sure there is nothing stored on the server. Adding the server to CloudStack will destroy any existing data.

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.
   - **Pod.** The pod for the storage device.
   - **Cluster.** 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. Add Secondary Storage

6.7.1. 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.

6.7.2. 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. If you are going to use Swift for cloud-wide secondary storage, you must add the Swift storage to CloudStack before you add the local zone secondary storage servers. See *Section 6.2, “Adding a Zone”*.
2. To prepare for local zone secondary storage, you should have created and mounted an NFS share during Management Server installation. See *Section 4.5.6, “Prepare NFS Shares”*.

3. Make sure you prepared the system VM template during Management Server installation. See *Section 4.5.8, “Prepare the System VM Template”*.

4. Now that the secondary storage server for per-zone storage is prepared, add it to CloudStack. Secondary storage is added as part of the procedure for adding a new zone. See *Section 6.2, “Adding a Zone”*.

6.8. 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 VM’s progress in the Instances screen.

4. To use the VM, click the View Console button. 

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.

Setting Global Configuration Parameters

CloudStack provides parameters that you can set to control many aspects of the cloud. When CloudStack is first installed, and periodically thereafter, you might need to modify these settings.

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. 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.

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

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 cloud-agent
```

In Ubuntu:

```
$ apt-get install cloud-agent
```

The host is now ready to be added to a cluster. This is covered in a later section, see *Section 6.5, "Adding a Host"*. It is recommended that you continue to read the documentation before adding the host!

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 cloud-agent and should already be installed.

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

   Set the following paramaters:

   ```
   listen_tls = 0
   ```

   ```
   listen_tcp = 1
   ```

   ```
   tcp_port = 16059
   ```

   ```
   auth_tcp = "none"
   ```

   ```
   mdns_adv = 0
   ```

2. Turning on "listen_tcp" in libvirtd.conf is not enough, we have to change the parameters as well:

   On RHEL or CentOS modify **`/etc/sysconfig/libvirtd`**:

   Uncomment the following line:

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

   On Ubuntu: modify **`/etc/init/libvirt-bin.conf`**

   Change the following line (at the end of the file):

   ```
   exec /usr/sbin/libvirtd -d
   ```

to (just add -l)

```
exec /usr/sbin/libvirtd -d -l
```

3. 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.

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 similair 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-Adress

```
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. 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.

8.1.8.1. 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
```

8.1.8.2. 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.

8.1.9. 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 6.5, "Adding a Host"*. It is recommended that you continue to read the documentation before adding the host!

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 homogenous. 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.

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 6.6, "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 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 6.6, "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.

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/lib64/cloud/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 cloud-management start
# service cloud-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/lib64/cloud/common/scripts/vm/hypervisor/xenserver/xenserver60/NFSSR.py	/opt/xensource/sm/NFSSR.py
/usr/lib64/cloud/common/scripts/vm/hypervisor/xenserver/setupxenserver.sh	/opt/xensource/bin/setupxenserver.sh

Copy this Management Server file...	...to this location on the XenServer host
/usr/lib64/cloud/common/scripts/vm/hypervisor/xenserver/make_migratable.sh	/opt/xensource/bin/make_migratable.sh
/usr/lib64/cloud/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 homogenous. 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 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:

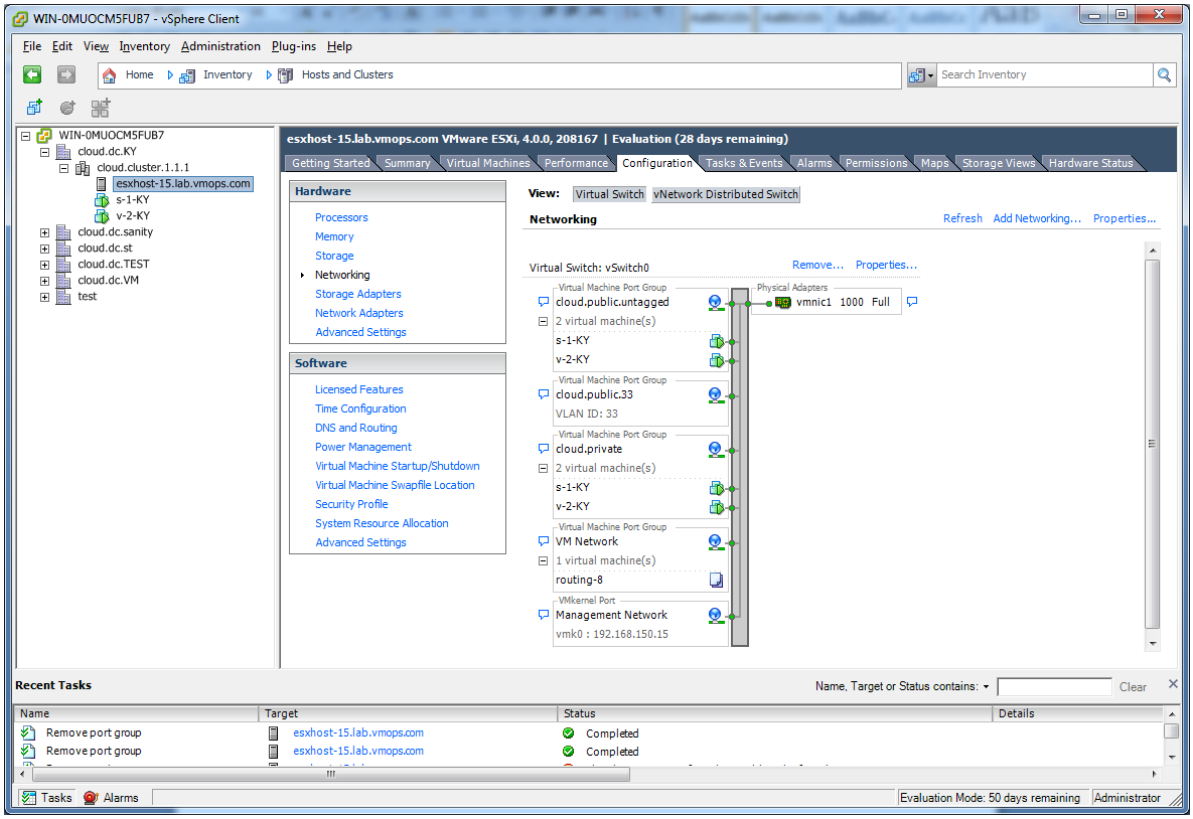
Required	Optional
ESXi host setup	NIC bonding
Configure host physical networking, virtual switch, vCenter Management Network, and extended port range	Multipath storage
Prepare storage for iSCSI	
Configure clusters in vCenter and add hosts to them, or add hosts without clusters to vCenter	

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.



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.

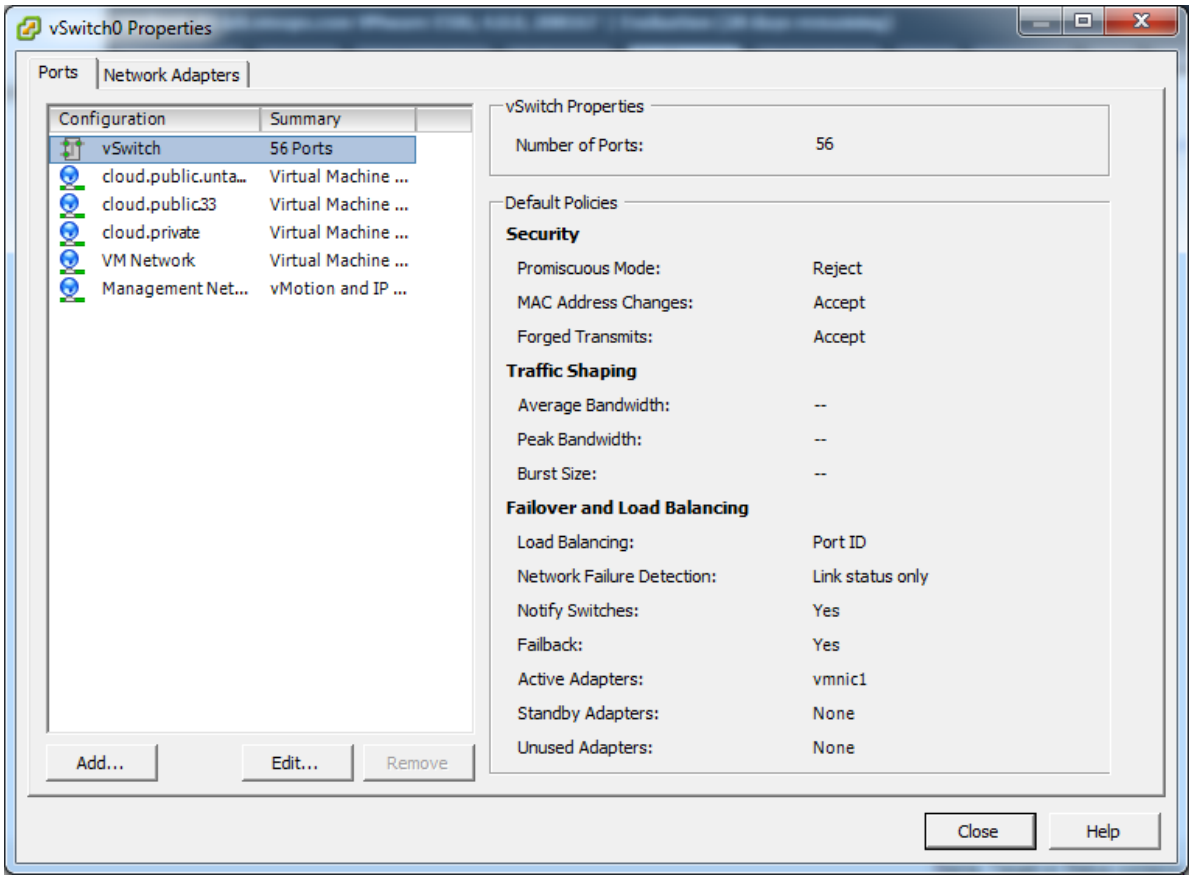
8.3.5.1.1. Separating Traffic

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.

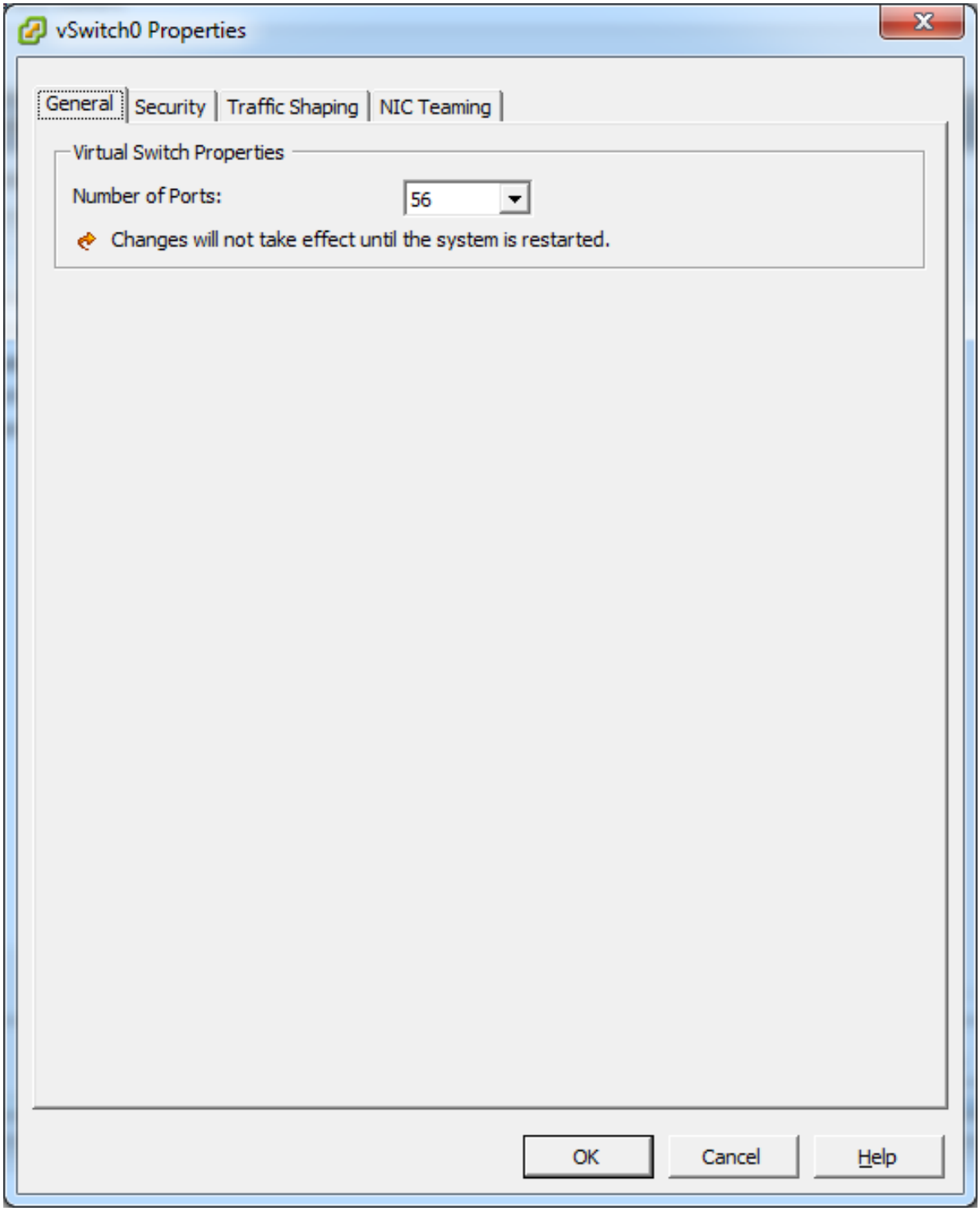
If you want to separate traffic in this way you should first create and configure vSwitches in vCenter according to the vCenter instructions. Take note of the vSwitch names you have used for each traffic type. You will configure CloudStack to use these vSwitches.

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).



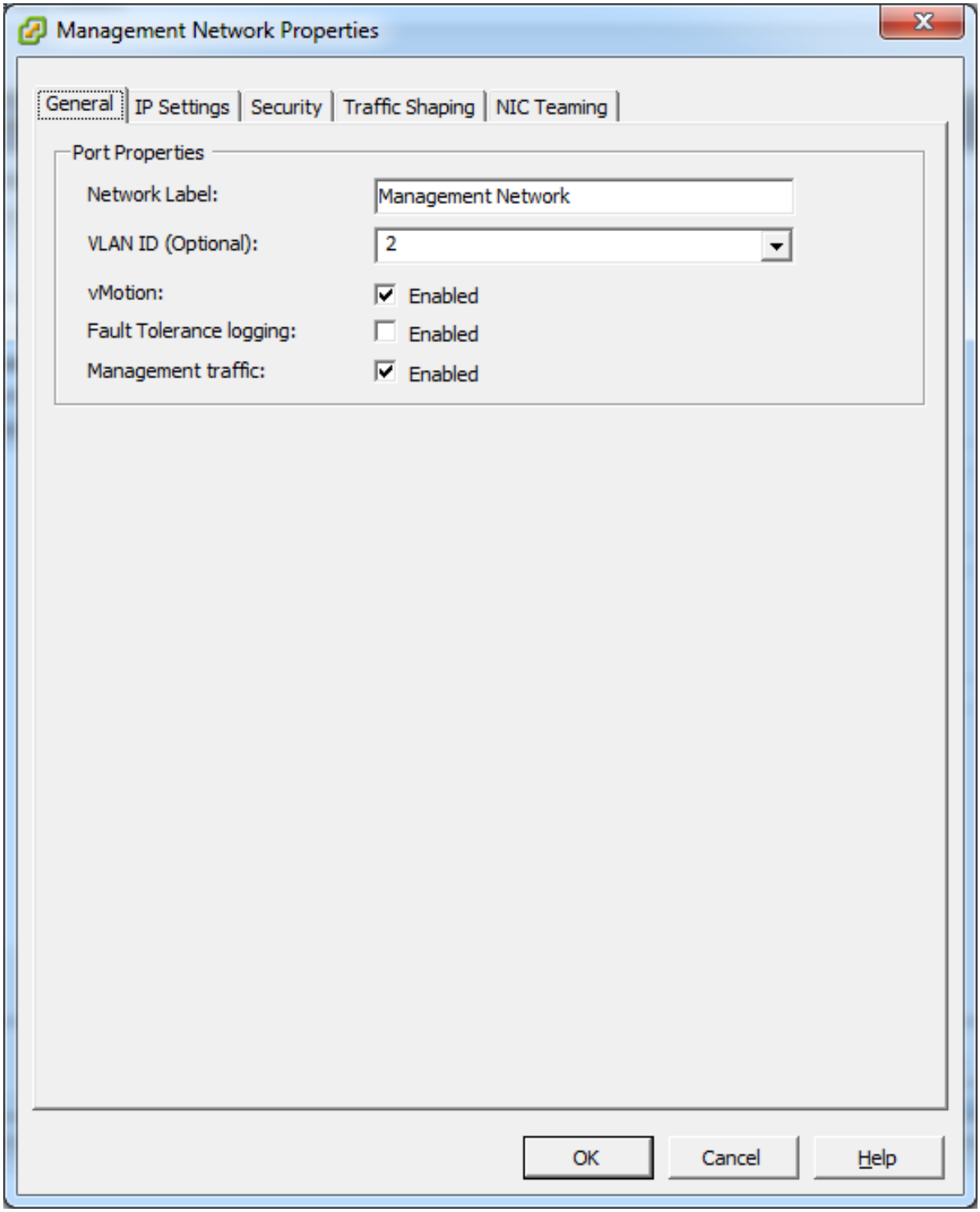
In vSwitch properties dialog, select the vSwitch and click Edit. You should see the following dialog:



In this dialog, you can change the number of switch ports. After you've done that, ESXi hosts are required to reboot in order for the setting to take effect.

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.



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.

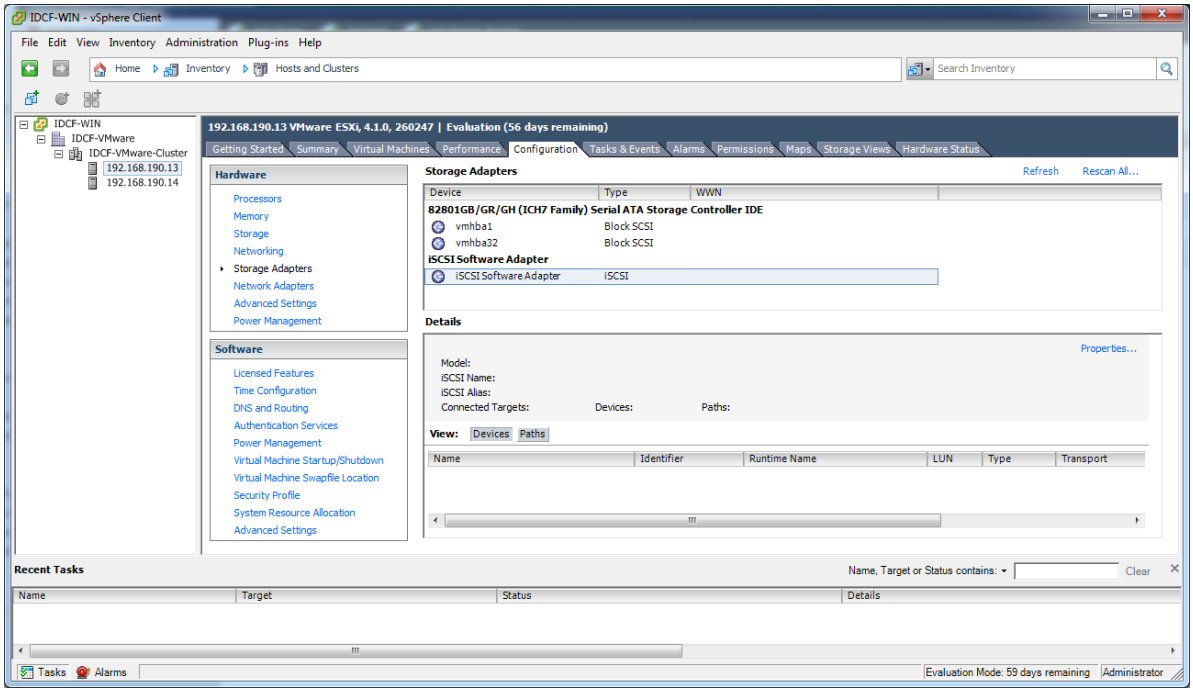
8.3.6. Storage Preparation for vSphere (iSCSI only)

Use of iSCSI requires preparatory work in vCenter. You must add an iSCSI target and create an iSCSI datastore.

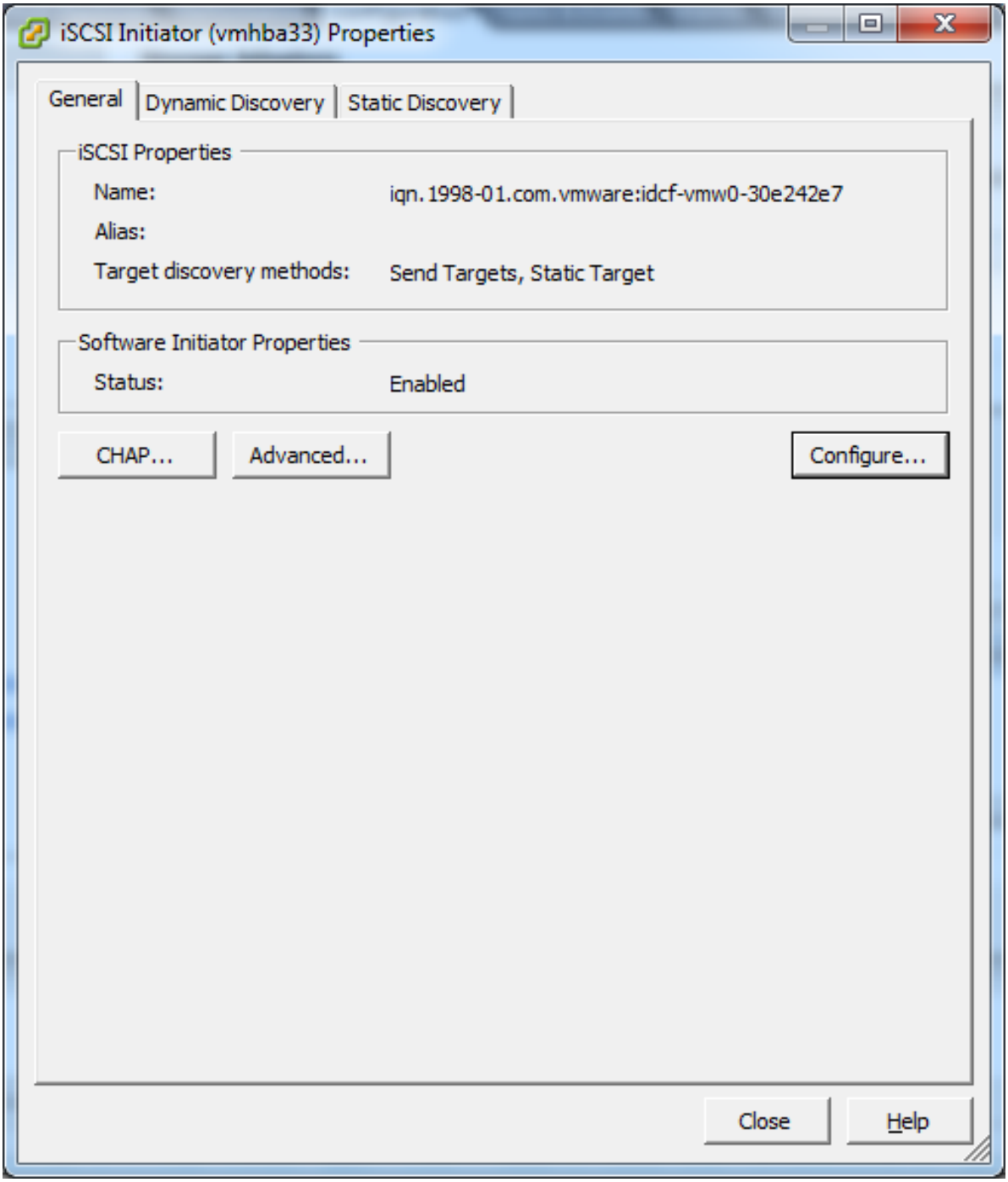
If you are using NFS, skip this section.

8.3.6.1. Enable iSCSI initiator for ESXi hosts

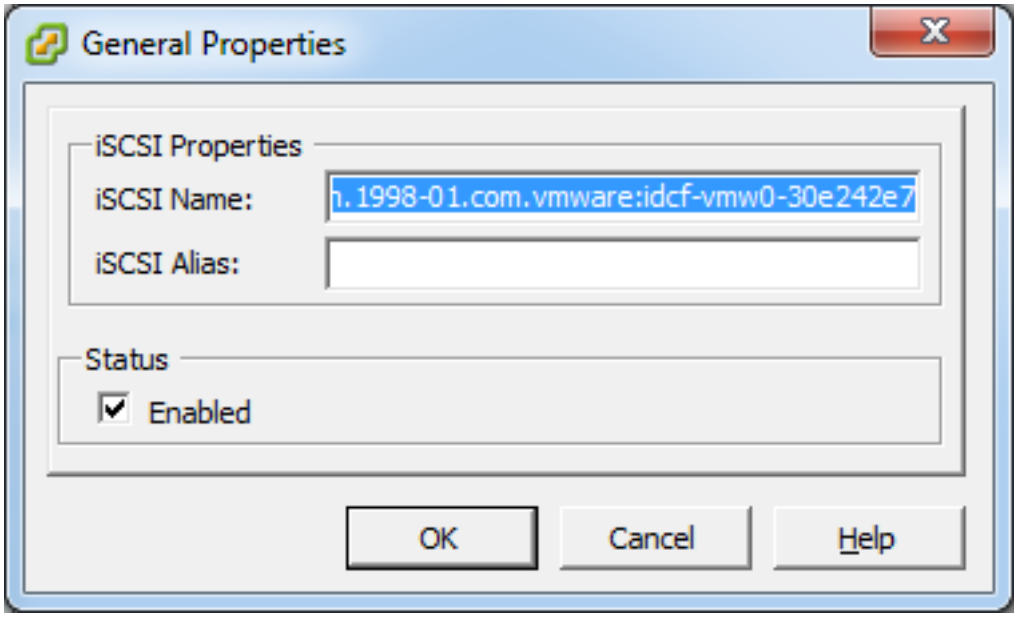
1. In vCenter, go to hosts and Clusters/Configuration, and click Storage Adapters link. You will see:



2. Select iSCSI software adapter and click Properties.



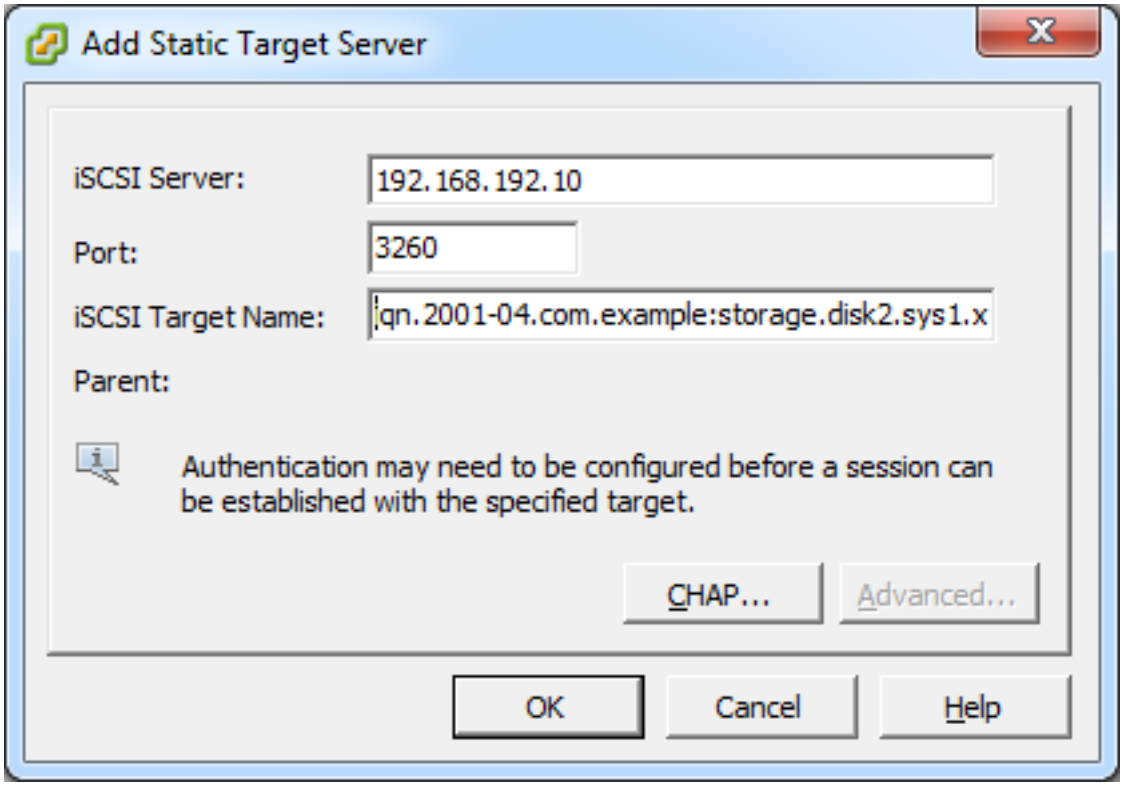
3. Click the Configure... button.



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

8.3.6.2. Add iSCSI target

Under the properties dialog, add the iSCSI target info:



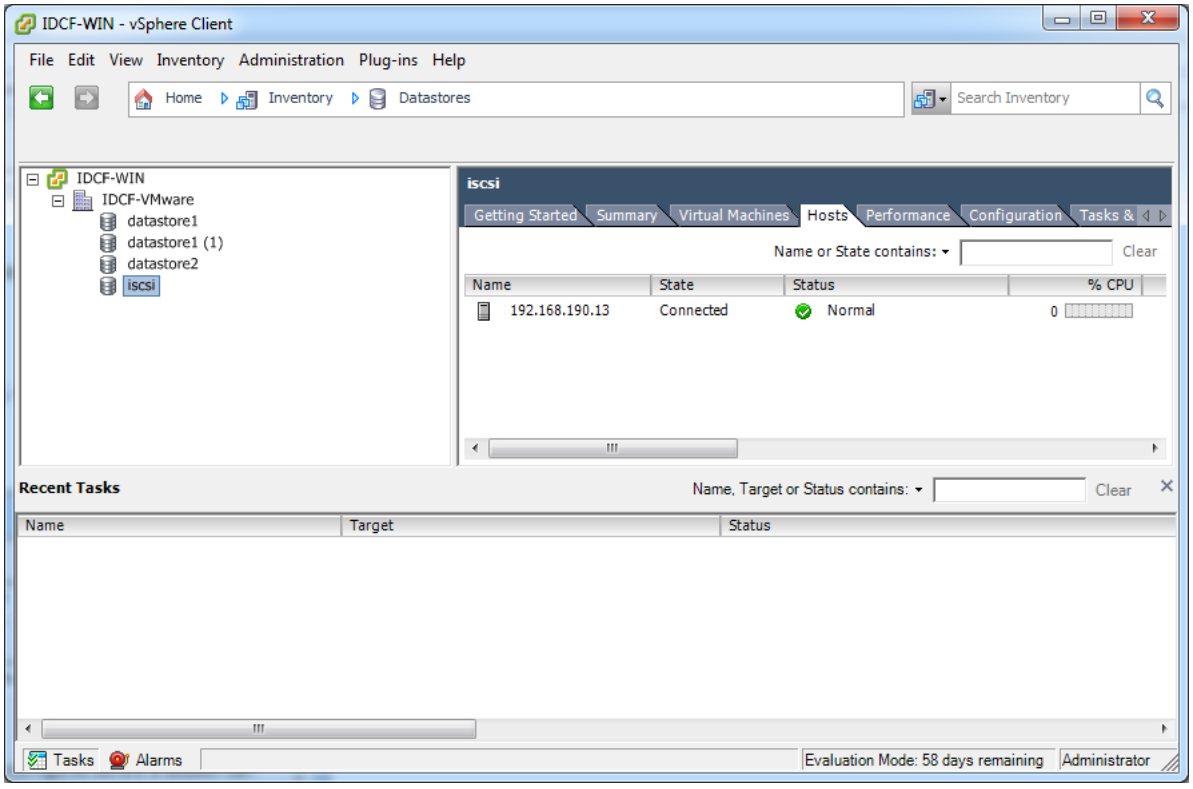
Repeat these steps for all ESXi hosts in the cluster.

8.3.6.3. 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.



8.3.6.4. Multipathing for vSphere (Optional)

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

8.3.7. 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 6.4.2, “Add Cluster: vSphere”*).

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 cloud-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:

   ```
   # service mysqld 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.

   ```
   # 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.

```
# service mysqld 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:

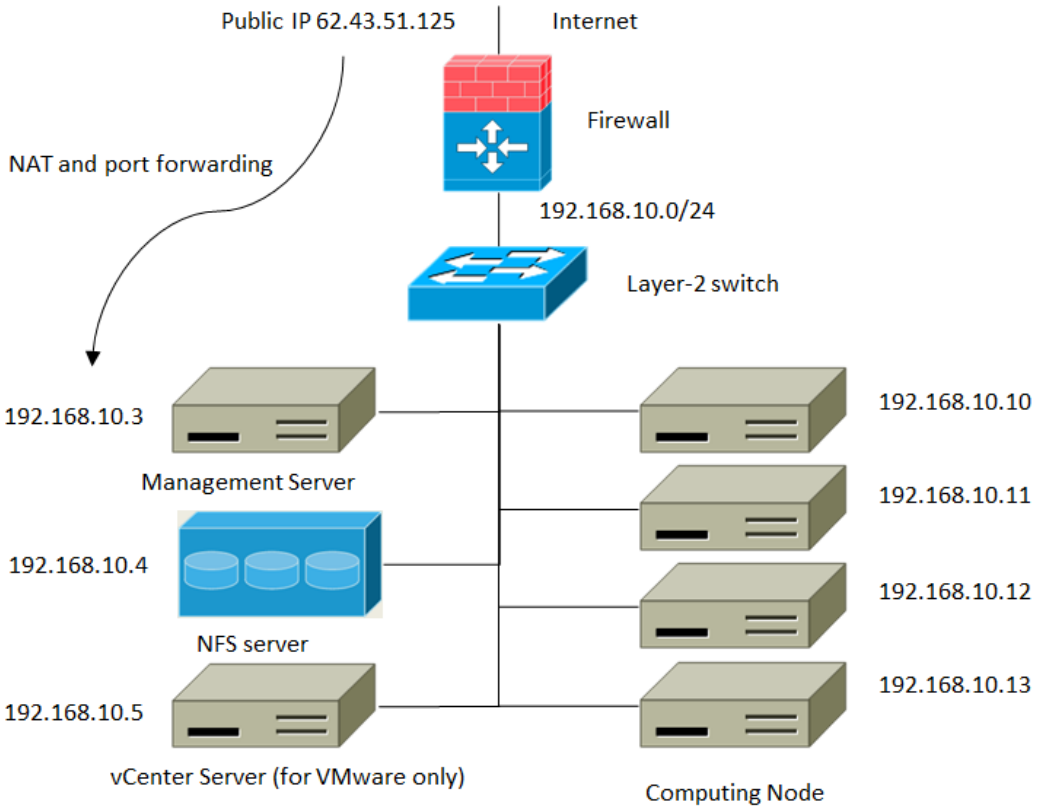
1. Stop the Management Servers (via service cloud-management stop).
2. Change the replica's configuration to be a master and restart it.
3. Ensure that the replica's port 3306 is open to the Management Servers.
4. Make a change so that the Management Server uses the new database. The simplest process here is to put the IP address of the new database server into each Management Server's /etc/cloud/management/db.properties.
5. Restart the Management Servers:

```
# service cloud-management start
```

Choosing a Deployment Architecture

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

10.1. Small-Scale Deployment

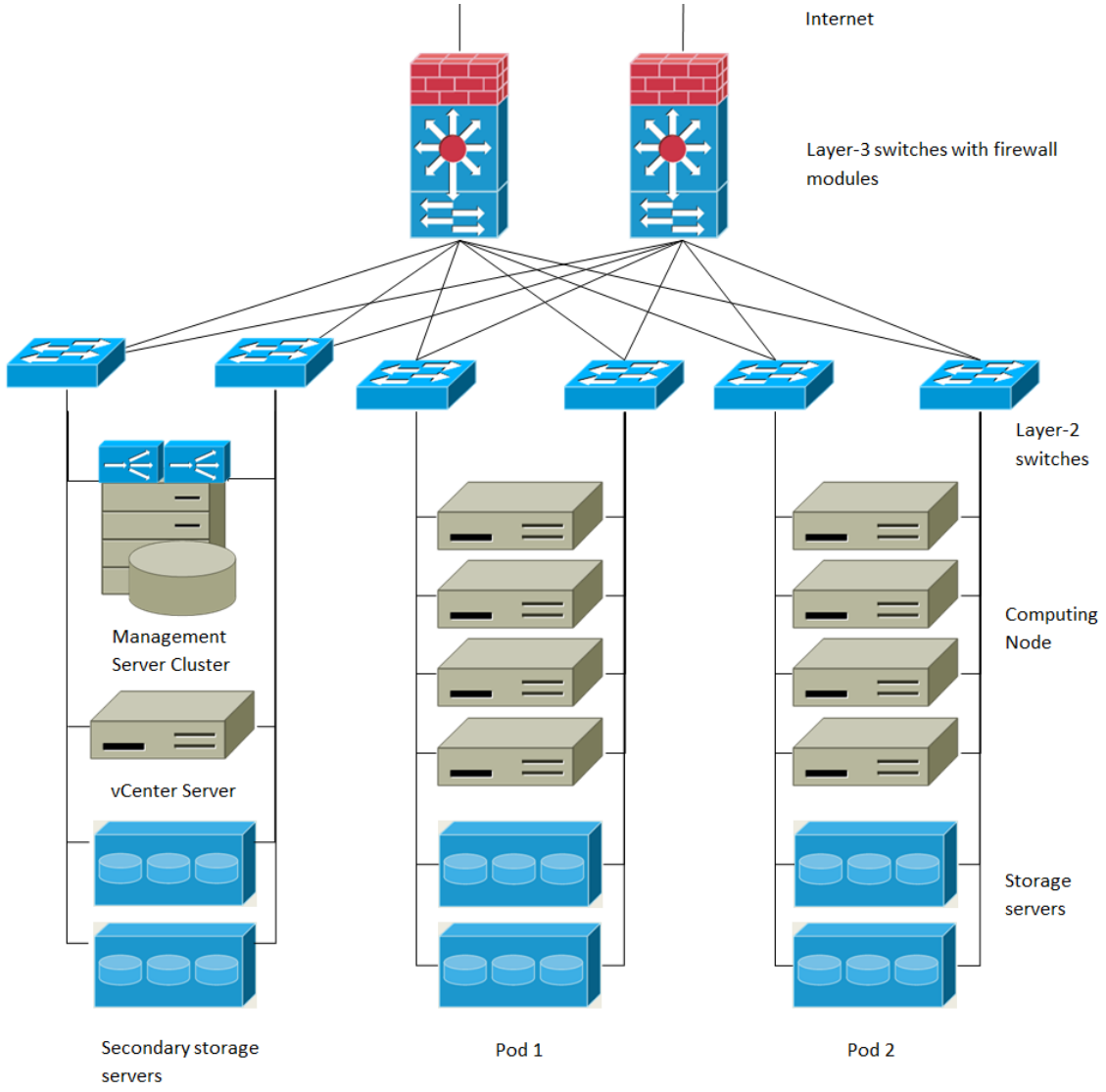


Small-Scale Deployment

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.

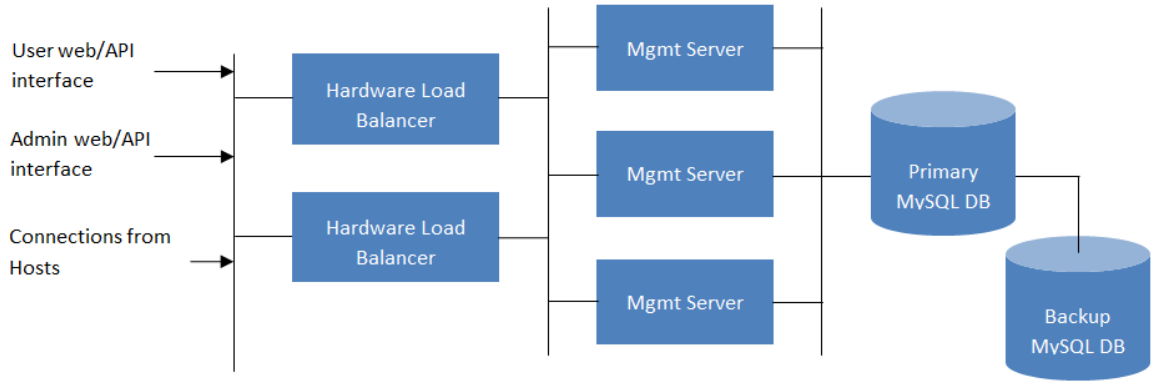
- The Management Server cluster (including front-end load balancers, Management Server nodes, and the MySQL database) is connected to the management network through a pair of load balancers.
- Secondary storage servers are connected to the management network.
- Each pod contains storage and computing 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.



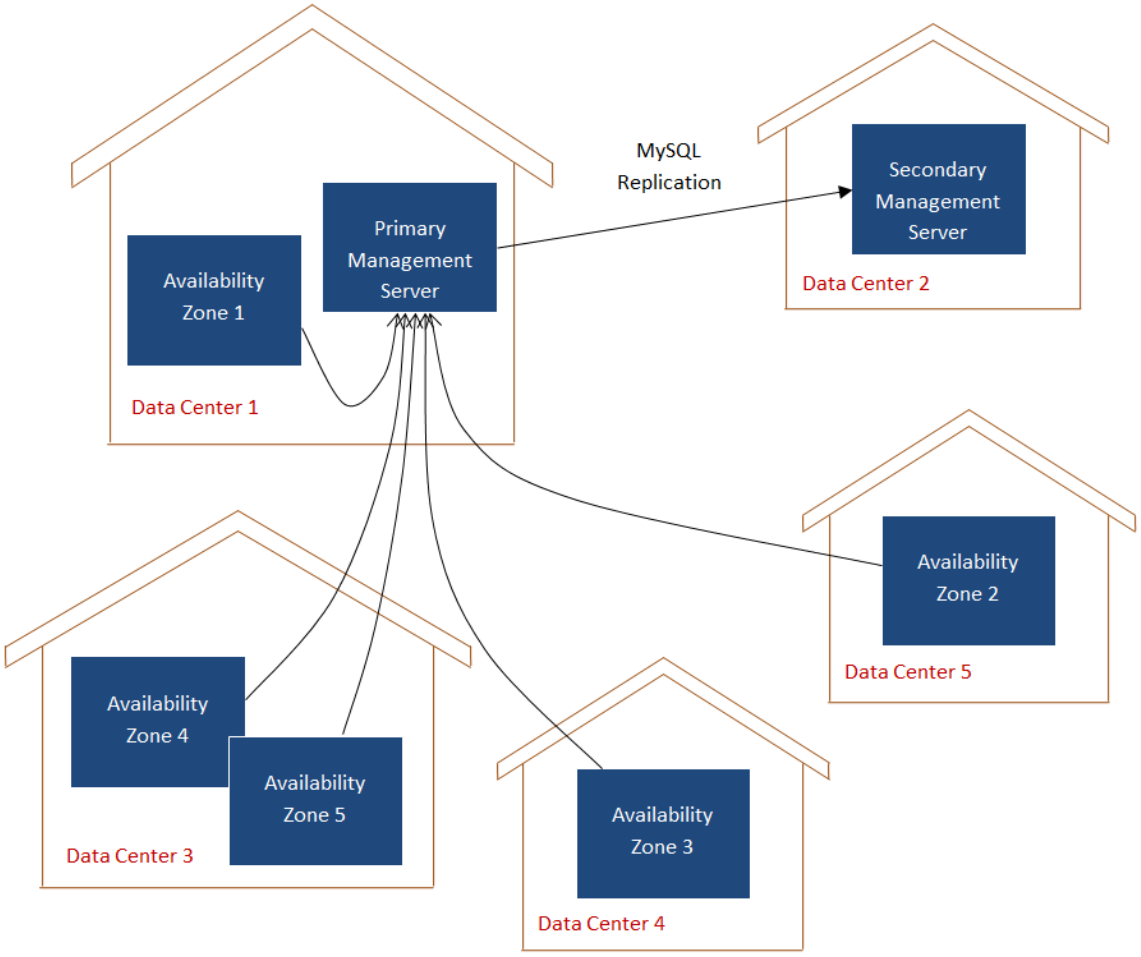
Multi-Node Management Server Deployment

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.

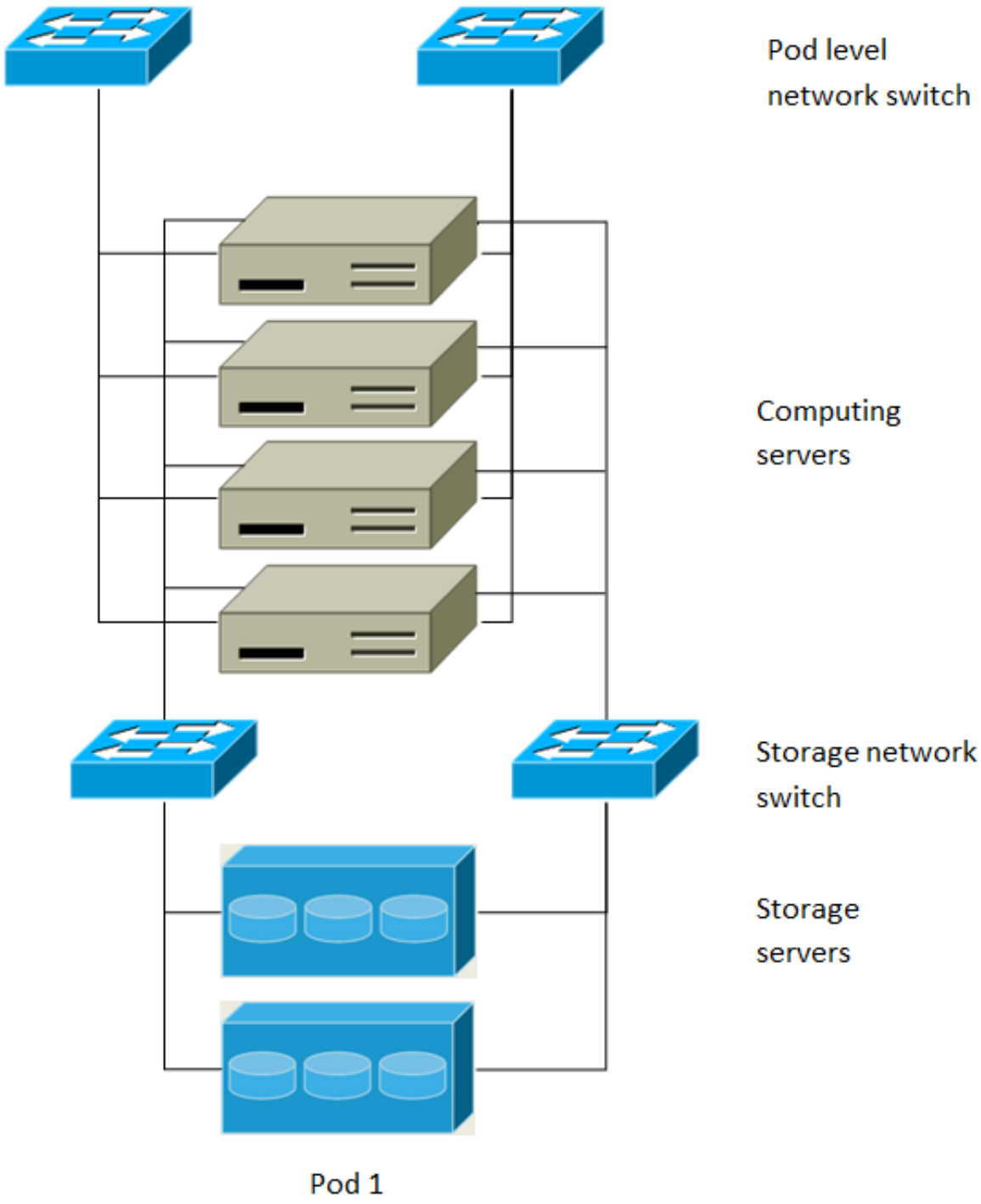
10.5. Multi-Site Deployment

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.



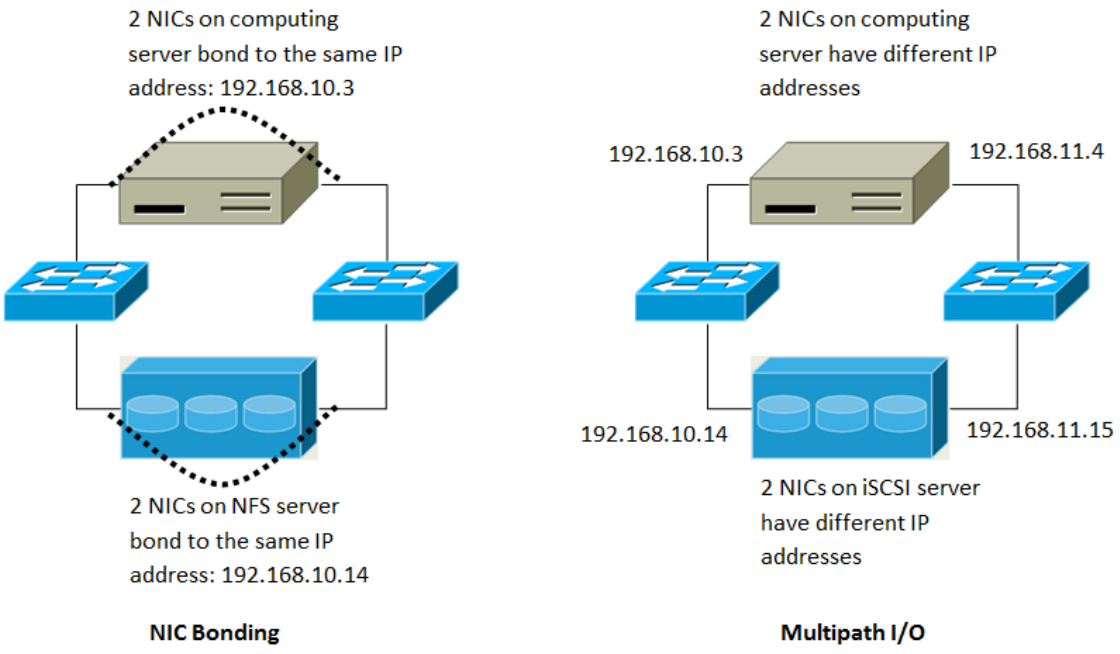
Separate Storage Network

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.



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.

Amazon Web Services Compatible Interface

11.1. 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 Balacing (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).

11.2. 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

11.3. 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 Global Configuration Parameters*.

2. Create a set of CloudStack service offerings with names that match the Amazon service offerings. You can do this through the CloudStack UI as described in the Administration Guide.

   Warning

   Be sure you have included the Amazon default service offering, m1.small. As well as any EC2 instance types that you will use.

3. If you did not already do so when you set the configuration parameter in step 1, restart the Management Server.

   ```
   # service cloud-management restart
   ```

The following sections provides details to perform these steps

11.3.1. 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

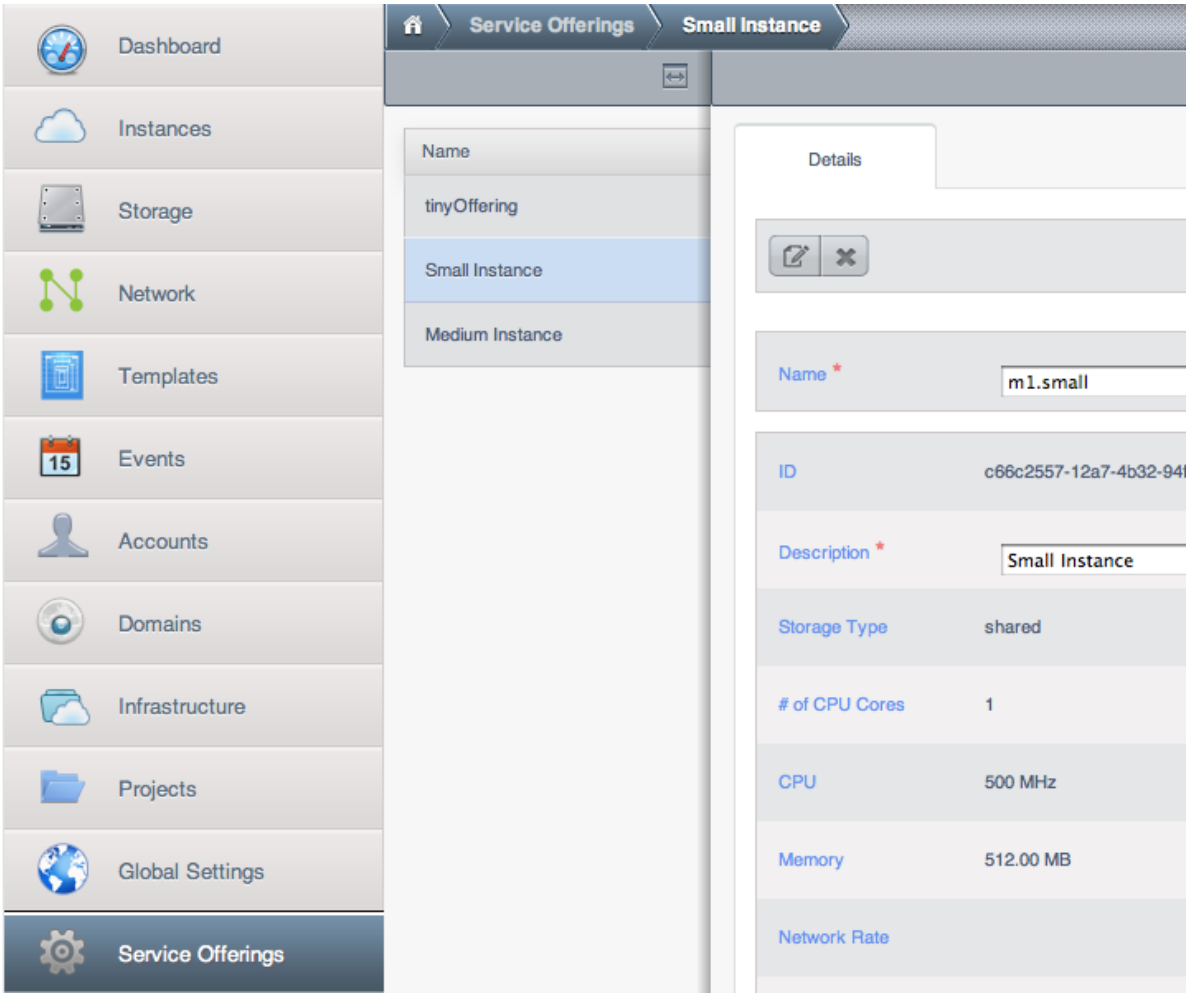
Using the CloudStack API, the easiest is to use the so-called integration port on which you can make unauthenticated calls. In Global Settings set the port to 8096 and subsequently call the *updateConfiguration* method. The following urls shows you how:

```
            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.

11.3.2. Creating EC2 Compatible Service Offerings

You will also need to define compute service offerings with names compatible with the *Amazon EC2 instance types*[1] 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:



[1] http://aws.amazon.com/ec2/instance-types/

11.3.3. 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/cloud/management/server.xml, /etc/cloud/management/server-nonssl.xml, and /etc/cloud/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.

11.4. 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.

11.4.1. 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=incubator-cloudstack.git;a=blob_plain;f=awsapi-setup/setup/cloudstack-aws-api-register;hb=HEAD"[2]
```

Then execute it, using the parameter values 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.

11.4.2. 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
```

11.5. 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:

[2] https://git-wip-us.apache.org/repos/asf?p=incubator-cloudstack.git;a=blob_plain;f=awsapi-setup/setup/cloudstack-aws-api-register;hb=HEAD

	`--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.

11.6. 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 11.1. 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 11.2. Availability Zone API mapping

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

Table 11.3. 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 11.4. Image Attributes API mapping

EC2 command	SOAP call	CloudStack API call
ec2-describe-image-attribute	DescribeImageAttribute	listTemplatePermissions
ec2-modify-image-attribute	ModifyImageAttribute	updateTemplatePermissions

EC2 command	SOAP call	CloudStack API call
ec2-reset-image-attribute	ResetImageAttribute	updateTemplatePermissions

Table 11.5. 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 11.6. Instance Attributes Mapping

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

Table 11.7. 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 11.8. Passwords API Mapping

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

Table 11.9. 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 11.10. Snapshots API Mapping

EC2 command	SOAP call	CloudStack API call
ec2-create-snapshot	CreateSnapshot	createSnapshot
ec2-delete-snapshot	DeleteSnapshot	deleteSnapshot
ec2-describe-snapshots	DescribeSnapshots	listSnapshots

Table 11.11. 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 command	SOAP call	CloudStack API call
ec2-describe-volume	DescribeVolume	listVolumes
ec2-detach-volume	DetachVolume	detachVolume

11.7. 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.

11.7.1. 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 11.1. An EC2 Boto example

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

import sys
import os
import boto
import boto.ec2

region = boto.ec2.regioninfo.RegionInfo(name="ROOT",endpoint="localhost")
apikey='GwNnpUPrO6KgIdZu01z_ZhhZnKjtSdRwuYd4DvpzvFpyxGMvrzno2q05MB0ViBoFYtdqKd'
secretkey='t4eXLEYWw7chBhDlaKf38adCMSHx_wlds6JfSx3z9fSpSOm0AbP9Moj0oGIzy2LSC8iw'

def main():
 '''Establish connection to EC2 cloud'''
        conn =boto.connect_ec2(aws_access_key_id=apikey,
                        aws_secret_access_key=secretkey,
                        is_secure=False,
                        region=region,
                        port=7080,
                        path="/awsapi",
                        api_version="2010-11-15")

        '''Get list of images that I own'''
 images = conn.get_all_images()
 print images
 myimage = images[0]
 '''Pick an instance type'''
 vm_type='m1.small'
 reservation = myimage.run(instance_type=vm_type,security_groups=['default'])

if __name__ == '__main__':
 main()
```

Second is an S3 example. 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 11.2. 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()
```

11.7.2. JClouds Examples

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.

12.1. Basic and Advanced Networking

CloudStack provides two styles of networking:.

Basic

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

Advanced

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

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

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

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

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

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

12.2. 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	

12.3. 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.

12.3.1. 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.

12.3.2. 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.

12.4. 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.

12.4.1. 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.

12.4.2. 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.

12.5. 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 12.5.2, "External Guest Firewall Integration for Juniper SRX (Optional)"*.

12.5.1. 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.

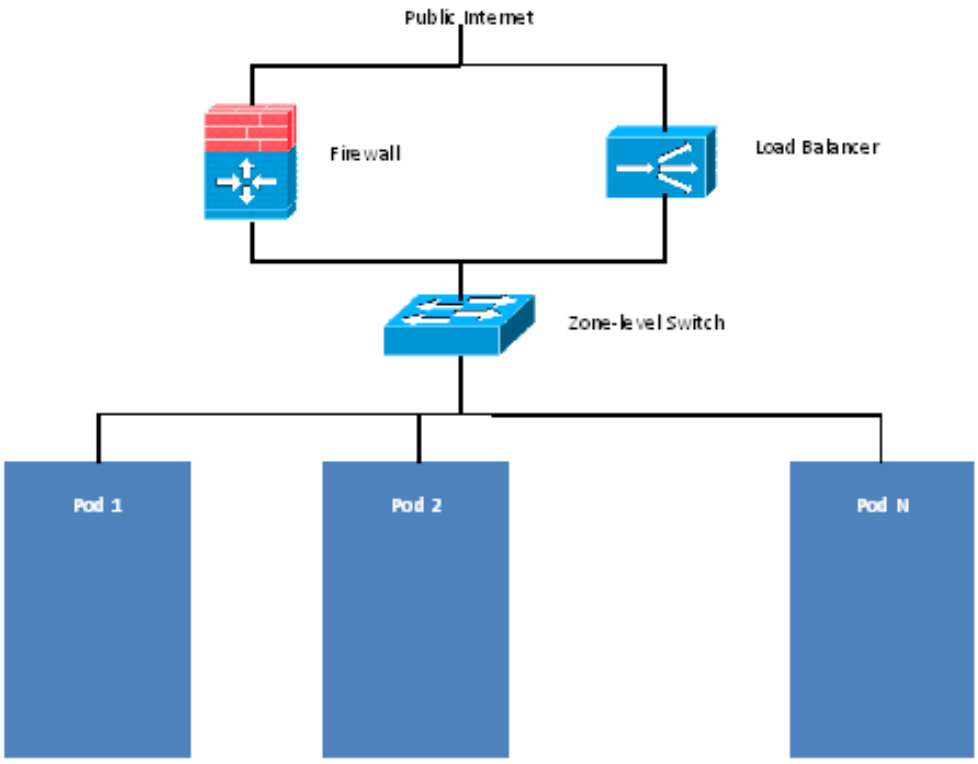
12.5.2. External Guest Firewall Integration for Juniper SRX (Optional)

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.

12.5.3. 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.

12.6. Setting Zone VLAN and Running VM Maximums

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

12.7. Topology Requirements

12.7.1. Security Requirements

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

12.7.2. 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.

12.7.3. 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.

12.7.4. 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.

12.7.5. Advanced Zone Topology Requirements

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

12.7.6. XenServer Topology Requirements

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

12.7.7. 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.
- 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.

12.7.8. KVM Topology Requirements

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

12.8. 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*[1].
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.

[1] http://inmon.com.

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*[2].

   For information about how to call the CloudStack API, see the Developer's Guide at *CloudStack API Developer's Guide*[3].

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.

12.9. 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.

[2] http://incubator.apache.org/cloudstack/docs/api/index.html
[3] http://incubator.apache.org/cloudstack/docs/en-US/Apache_CloudStack/4.0.0-incubating/html/API_Developers_Guide/index.html

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.

13.1. 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.

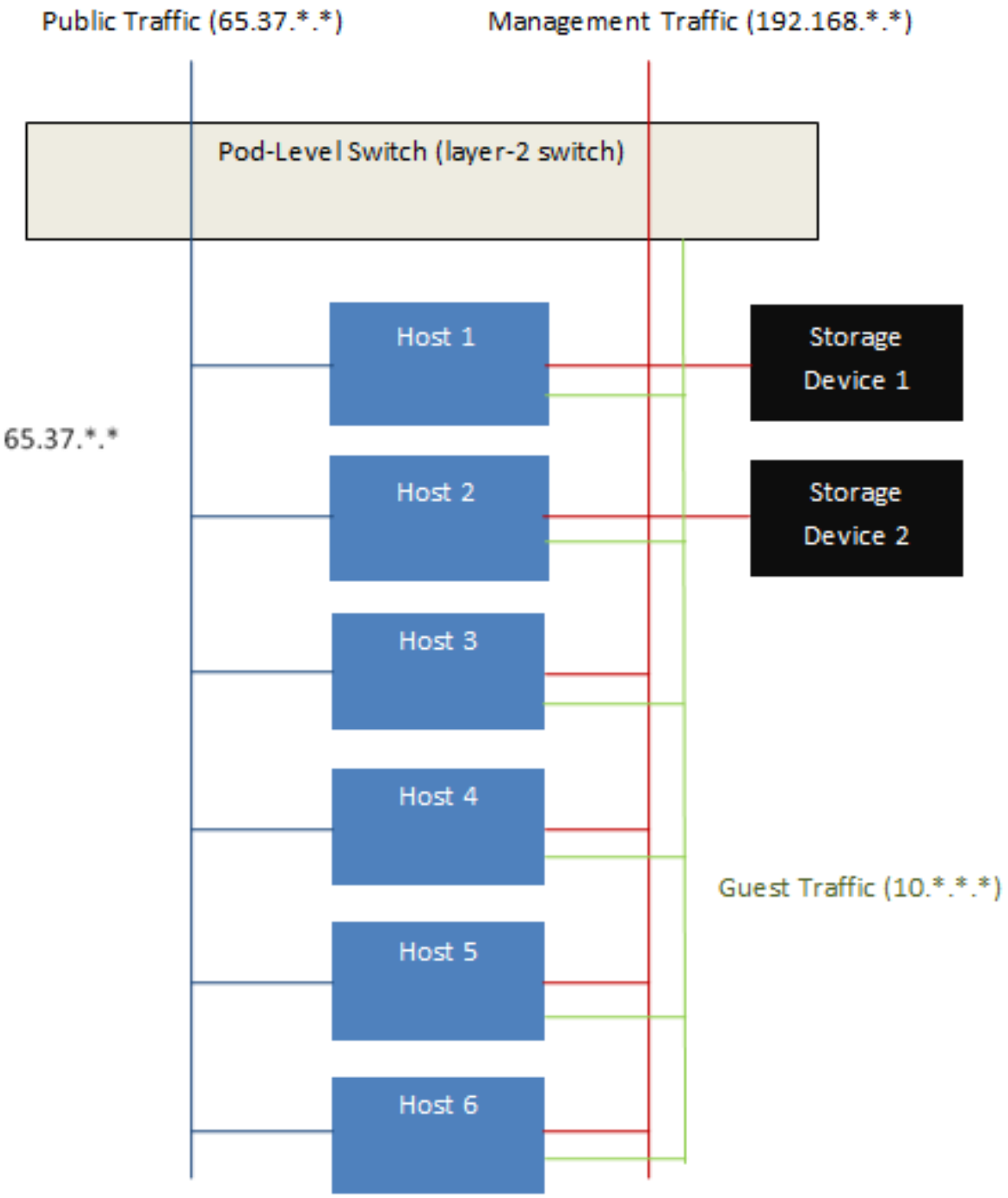
The Management Server automatically creates a virtual router for each network. A virtual router is a special virtual machine that runs on the hosts. Each virtual router has three network interfaces. Its eth0 interface serves as the gateway for the guest traffic and has the IP address of 10.1.1.1. Its eth1 interface is used by the system to configure the virtual router. Its eth2 interface is assigned a public IP address for public traffic.

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

13.2. Networking in a Pod

Figure 2 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

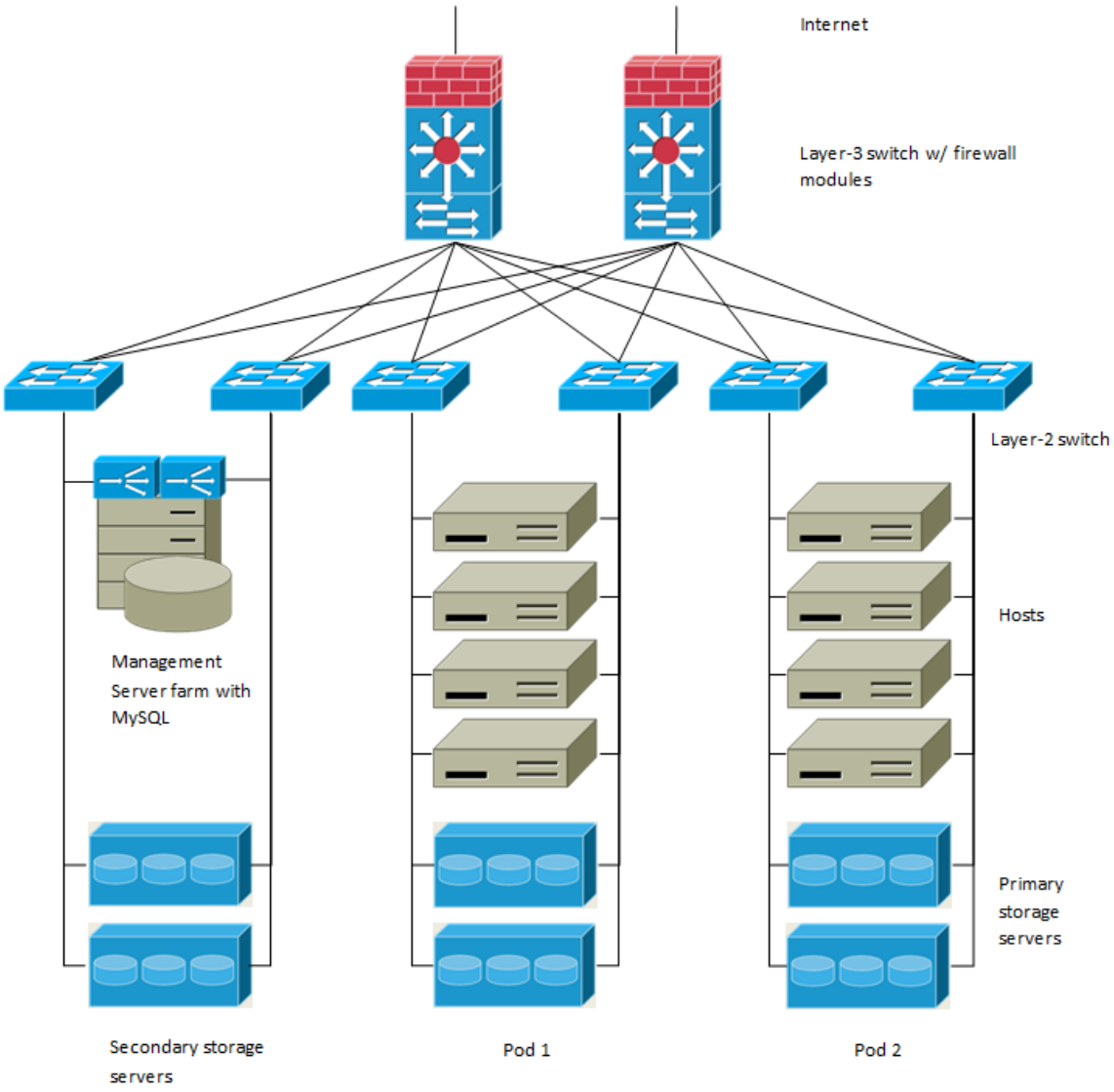
Servers are connected as follows:

- 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.

13.3. Networking in a Zone

Figure 3 illustrates the network setup within a single zone.



Networking setup in a zone

A firewall for management traffic operates in the NAT mode. The network typically is assigned IP addresses in the 192.168.0.0/16 Class B private address space. Each pod is assigned IP addresses in the 192.168.*.0/24 Class C private address space.

Each zone has its own set of public IP addresses. Public IP addresses from different zones do not overlap.

13.4. 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.

13.5. Advanced Zone Physical Network Configuration

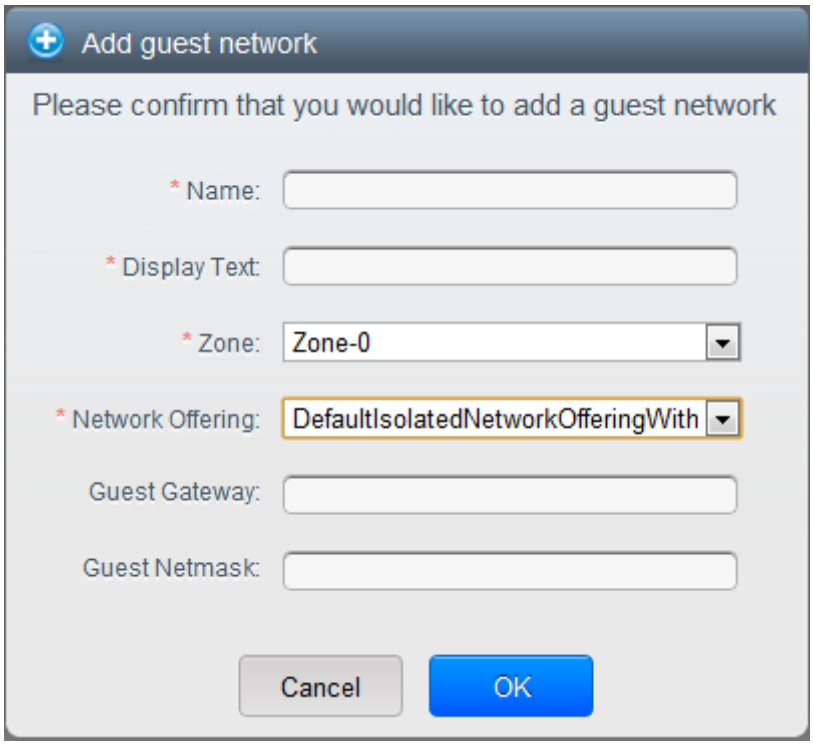
Within a zone that uses advanced networking, you need to tell the Management Server how the physical network is set up to carry different kinds of traffic in isolation.

13.5.1. 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:



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.

13.5.2. 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.

13.6. 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.

13.6.1. 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.

13.6.2. 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.

- Log in to the CloudStack UI as an administrator or end user.

- 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. See Stopping and Starting VMs. Then return here and continue to the next step
- In the left navigation, choose Network
- Click the name of the network you want to modify  .
- In Network Offering, choose the new network offering, then click Apply.
- A prompt appears 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. Choose No to proceed with the change.
- Wait for the update to complete. Don't try to restart VMs until after the network change is complete.
- If you stopped any VMs in step 2, restart them.

13.7. Security Groups

13.7.1. 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 CloudStack 3.0.3 - 3.0.5, security groups are supported only in zones that use basic networking.

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.

13.7.2. 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.

13.7.3. 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.

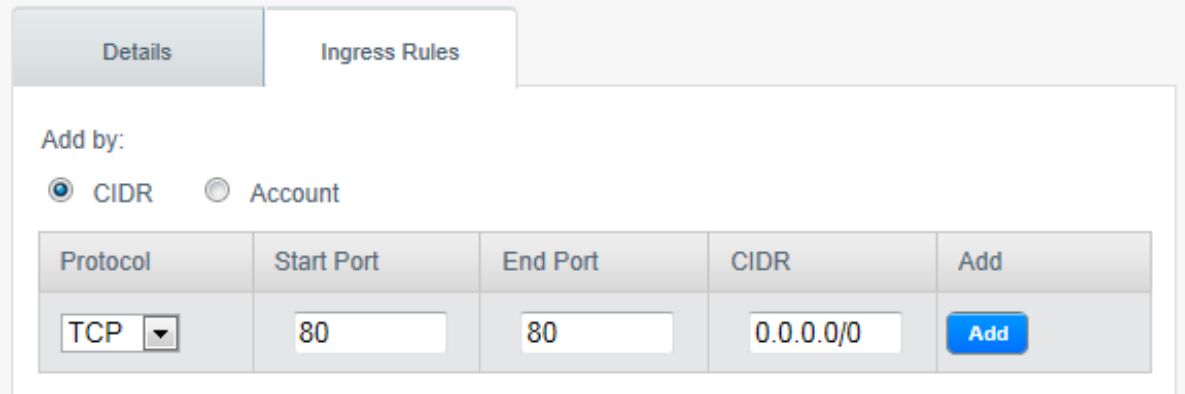
13.7.4. Adding Ingress and Egress Rules to a Security Group

1. Log in to the CloudStack UI as an administrator or end user.
2. In the left navigation, choose Network
3. In Select view, choose Security Groups, then click the security group you want .
4. To add an ingress rule, click the Ingress Rules tab and fill out the following fields to specify what network traffic is allowed into VM instances in this security group. If no ingress rules are specified, then no traffic will be allowed in, except for responses to any traffic that has been allowed out through an egress rule.

   - **Add by CIDR/Account**. Indicate whether the source of the traffic will be defined by IP address (CIDR) or an existing security group in a CloudStack account (Account). Choose Account if you want to allow incoming traffic from all VMs in another security group
   - **Protocol**. The networking protocol that sources will use to send traffic to the security group. TCP and UDP are typically used for data exchange and end-user communications. ICMP is typically used to send error messages or network monitoring data.
   - **Start Port, End Port**. (TCP, UDP only) A range of listening ports that are the destination for the incoming traffic. If you are opening a single port, use the same number in both fields.
   - **ICMP Type, ICMP Code**. (ICMP only) The type of message and error code that will be accepted.
   - **CIDR**. (Add by CIDR only) To accept only traffic from IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. The CIDR is the base IP address of the incoming traffic. For example, 192.168.0.0/22. To allow all CIDRs, set to 0.0.0.0/0.

- **Account, Security Group**. (Add by Account only) To accept only traffic from another security group, enter the CloudStack account and name of a security group that has already been defined in that account. To allow traffic between VMs within the security group you are editing now, enter the same name you used in step 7.

The following example allows inbound HTTP access from anywhere:



5. To add an egress rule, click the Egress Rules tab and fill out the following fields to specify what type of traffic is allowed to be sent out of VM instances in this security group. If no egress rules are specified, then all traffic will be allowed out. Once egress rules are specified, the following types of traffic are allowed out: traffic specified in egress rules; queries to DNS and DHCP servers; and responses to any traffic that has been allowed in through an ingress rule

   - **Add by CIDR/Account**. Indicate whether the destination of the traffic will be defined by IP address (CIDR) or an existing security group in a CloudStack account (Account). Choose Account if you want to allow outgoing traffic to all VMs in another security group.
   - **Protocol**. The networking protocol that VMs will use to send outgoing traffic. TCP and UDP are typically used for data exchange and end-user communications. ICMP is typically used to send error messages or network monitoring data.
   - **Start Port, End Port**. (TCP, UDP only) A range of listening ports that are the destination for the outgoing traffic. If you are opening a single port, use the same number in both fields.
   - **ICMP Type, ICMP Code**. (ICMP only) The type of message and error code that will be sent
   - **CIDR**. (Add by CIDR only) To send traffic only to IP addresses within a particular address block, enter a CIDR or a comma-separated list of CIDRs. The CIDR is the base IP address of the destination. For example, 192.168.0.0/22. To allow all CIDRs, set to 0.0.0.0/0.
   - **Account, Security Group**. (Add by Account only) To allow traffic to be sent to another security group, enter the CloudStack account and name of a security group that has already been defined in that account. To allow traffic between VMs within the security group you are editing now, enter its name.

6. Click Add.

13.8. 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.

13.9. 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.

13.10. 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.

13.11. 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, 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 or static NAT rules.

13.12. Releasing an 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 the IP address you want to release.
6. Click the Release IP button  .

13.13. 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.

13.14. IP Forwarding and Firewalling

By default, all incoming traffic to the public IP address is rejected. All outgoing traffic from the guests is translated via NAT to the public IP address and is allowed.

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.

For the steps to implement these rules, see Firewall Rules and Port Forwarding.

13.15. 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.

13.16. 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.

13.17. 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.

13.17.1. Configuring 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 addressess 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.

13.17.2. Using 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.

13.17.3. Using VPN with Mac OS X

In Mac OS X, in Network Preferences - Advanced, make sure Send all traffic over VPN connection is not checked.

13.17.4. 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 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

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 13.19, "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.

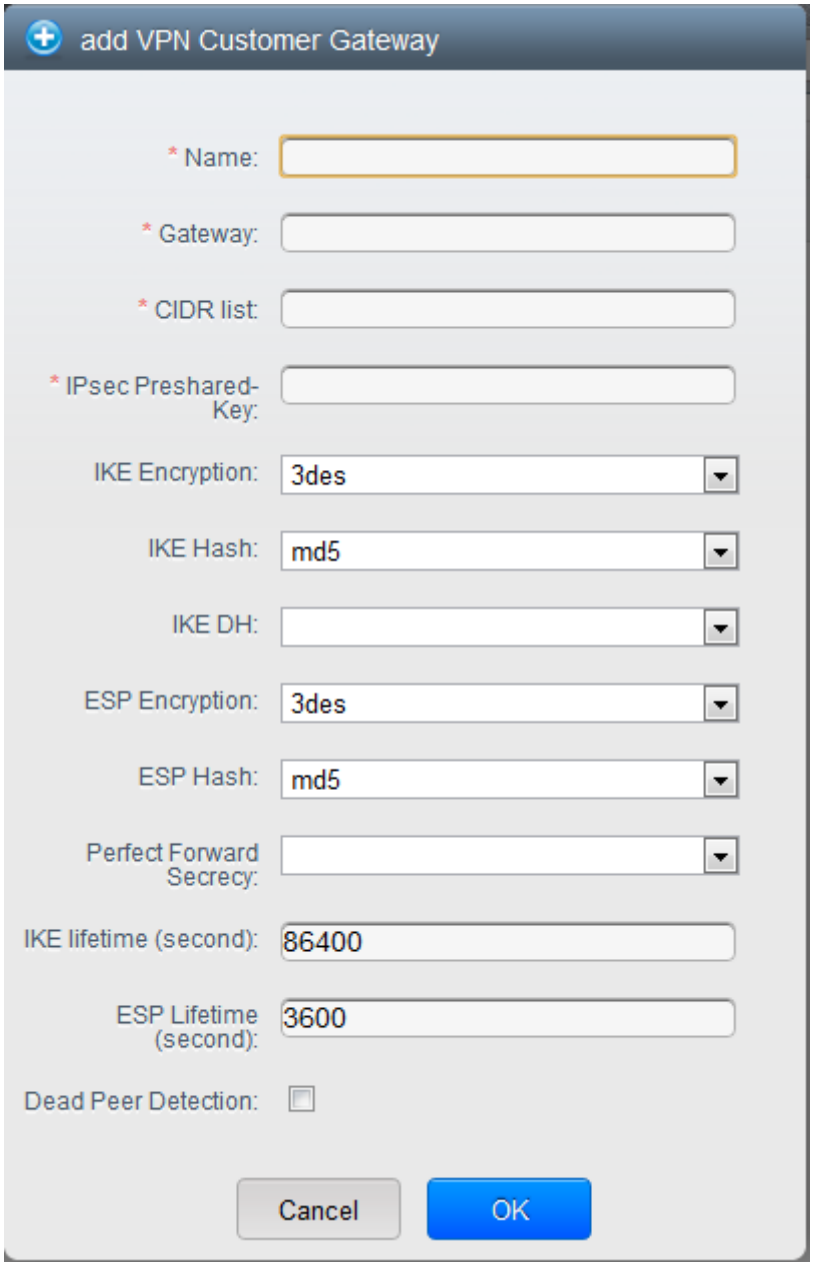
13.17.4.1. Creating and Updating a VPN Customer Gateway

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.



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.

13.17.4.2. 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.
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.

   The following options are displayed.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN
   - Network ACLs

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

13.17.4.3. Creating 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 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.

   The following options are displayed.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN

- Network ASLs

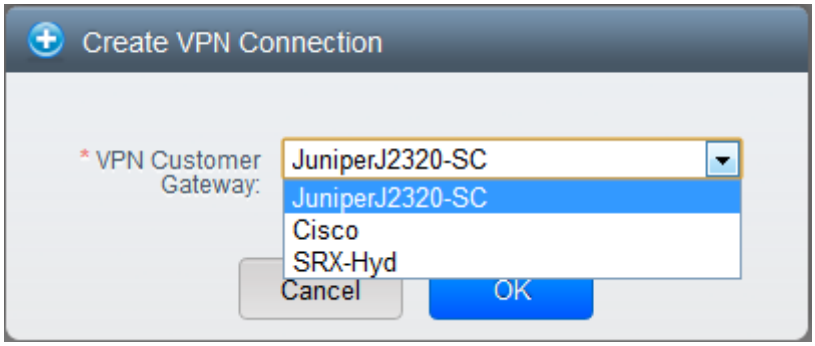
6. Select Site-to-Site VPN.

   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

13.17.4.4. 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.

   The following options are displayed.

- IP Addresses
- Gateways
- Site-to-Site VPN
- Network ASLs

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.
   

13.18. About Inter-VLAN Routing

Inter-VLAN Routing 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 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

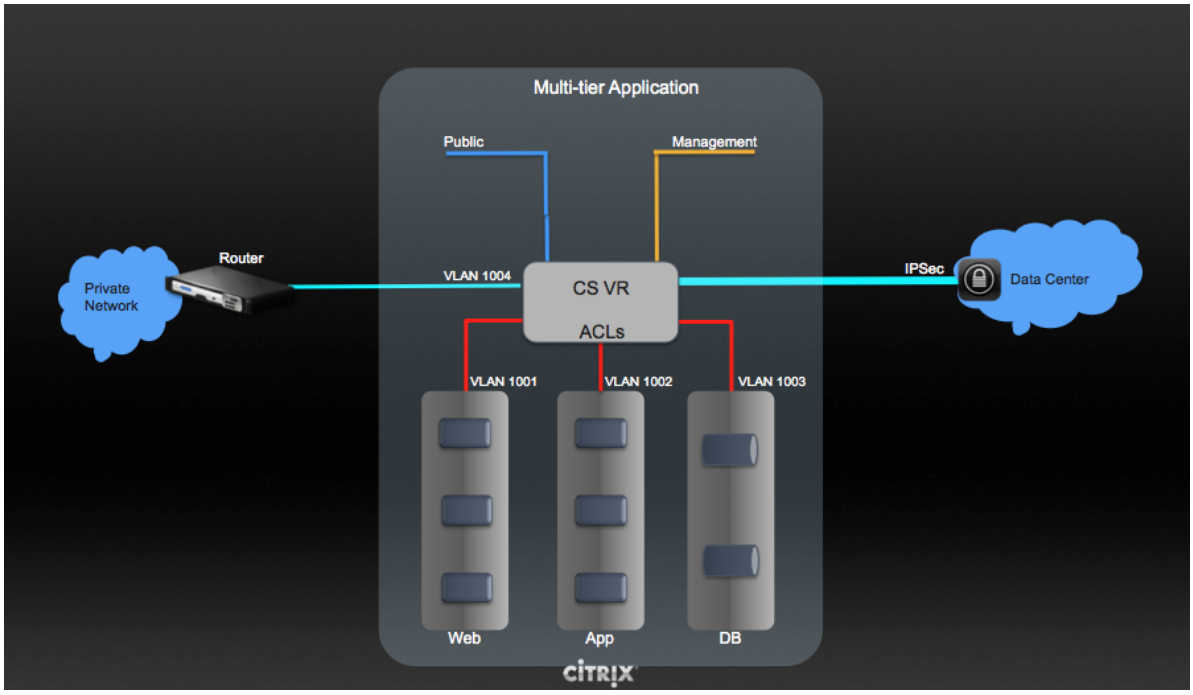
  A VLAN allocated for an account cannot be shared between multiple accounts.

- 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 13.17.4.2, "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 13.19.5, "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 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 13.19, "Configuring a Virtual Private Cloud"*.

13.19. Configuring a Virtual Private Cloud

13.19.1. 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 13.19.5, “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 13.17.4, “Setting Up a Site-to-Site VPN Connection”*.
- **Customer Gateway**: The customer side of a VPN Connection. For more information, see *Section 13.17.4.1, “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 13.19.9, “Enabling or Disabling Static NAT on a VPC”*.

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:

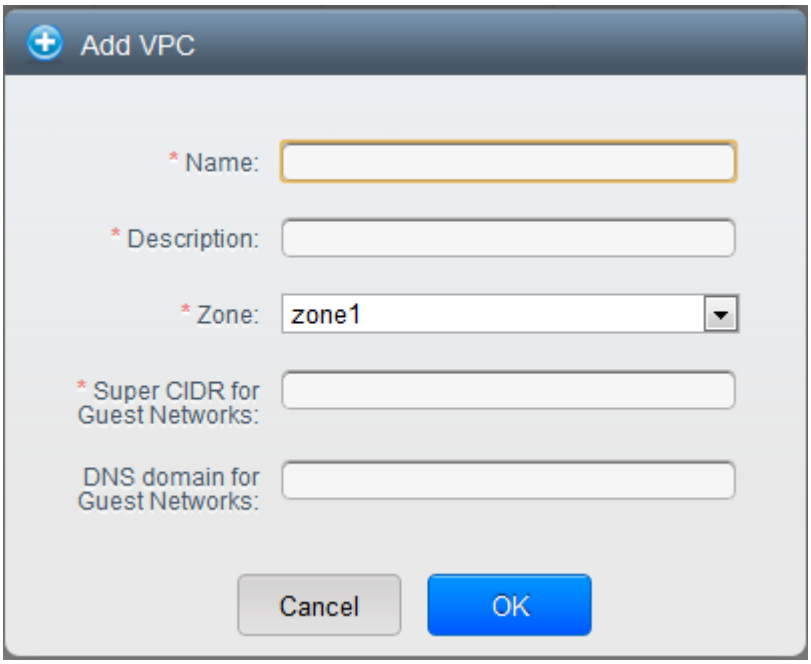
- A VPC, by default, is created in the enabled state.
- A VPC can be created in Advance zone only, and can't belong to more than one zone at a time.
- The default number of VPCs an account can create is 20. However, you can change it by using the max.account.vpcs global parameter, which controls the 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 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.

13.19.2. 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:



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.

13.19.3. 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.

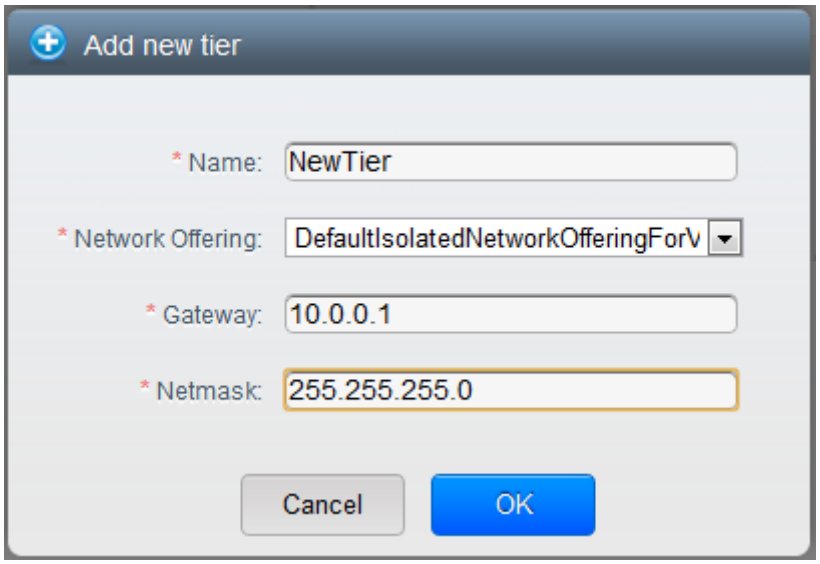
   All the VPC that you have created for the account is listed in the page.

   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.

   The Add new tier dialog is displayed, as follows:



   If you have already created tiers, the VPC diagram is displayed. Click Create Tier to add a new tier.

5. 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: 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.
- **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.

6. Click OK.
7. Continue with configuring access control list for the tier.

13.19.4. Configuring Access Control List

Define Network Access Control List (ACL) on the VPC virtual router to control incoming (ingress) and outgoing (egress) traffic between the VPC tiers, and the tiers and Internet. By default, all incoming and outgoing traffic to the guest networks 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.

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 Settings icon.

   The following options are displayed.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN
   - Network ACLs
5. Select Network ACLs.

   The Network ACLs page is displayed.
6. Click Add Network ACLs.

   To add an ACL rule, fill in the following fields to specify what kind of network traffic is allowed in this tier.

   - **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.
   - **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.

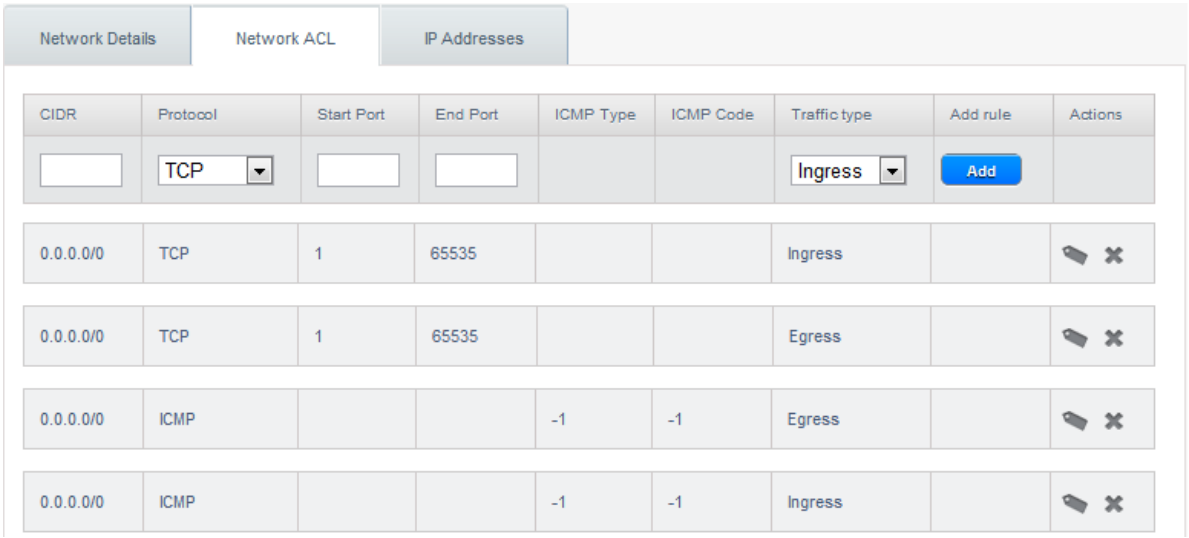
- **Start Port**, **End Port** (TCP, UDP only): A range of listening ports that are the destination for the incoming traffic. If you are opening a single port, use the same number in both fields.
- **Select Tier**: Select the tier for which you want to add this ACL rule.
- **ICMP Type**, **ICMP Code** (ICMP only): The type of message and error code that will be sent.
- **Traffic Type**: Select the traffic type you want to apply.
  - **Egress**: To add an egress rule, select Egress from the Traffic type drop-down box and click Add. This specifies what type of traffic is allowed to be sent out of VM instances in this tier. If no egress rules are specified, all traffic from the tier is allowed out at the VPC virtual router. Once egress rules are specified, only the traffic specified in egress rules and the responses to any traffic that has been allowed in through an ingress rule are allowed out. No egress rule is required for the VMs in a tier to communicate with each other.
  - **Ingress**: To add an ingress rule, select Ingress from the Traffic type drop-down box and click Add. This specifies what network traffic is allowed into the VM instances in this tier. 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.

Note

By default, all incoming and outgoing traffic to the guest networks is blocked. To open the ports, create a new network ACL.

7. Click Add. The ACL rule is added.

   To view the list of ACL rules you have added, click the desired tier from the Network ACLs page, then select the Network ACL tab.



   You can edit the tags assigned to the ACL rules and delete the ACL rules you have created. Click the appropriate button in the Actions column.

13.19.5. 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. 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.

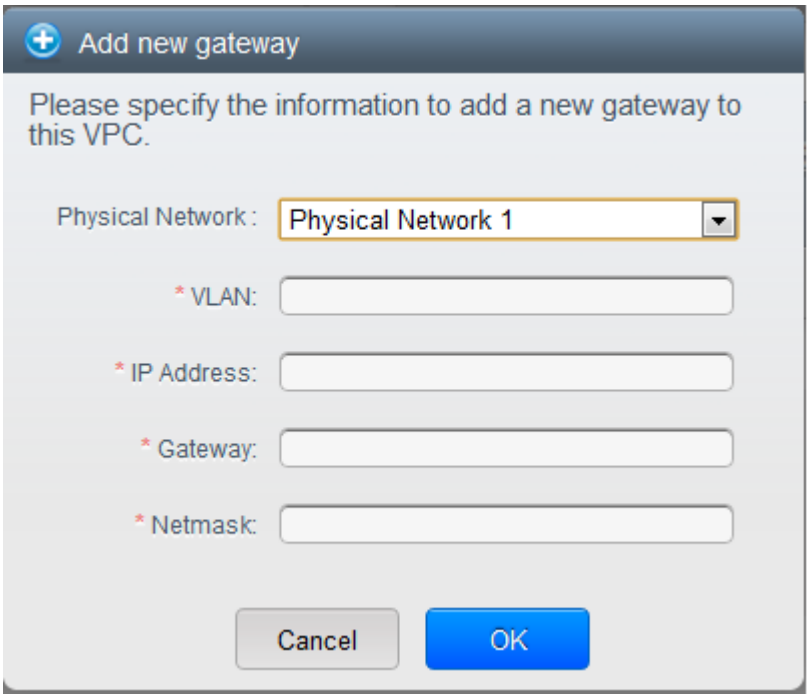
   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to configure load balancing rules.

   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.

   - IP Addresses
   - Private Gateways
   - Site-to-Site VPN
   - Network ACLs
6. Select Private Gateways.

   The Gateways page is displayed.
7. Click Add new gateway:



8. Specify the following:

- **Physical Network**: The physical network you have created in the zone.
- **IP Address**: The IP address associated with the VPC gateway.
- **Gateway**: The gateway through which the traffic is routed to and from the VPC.
- **Netmask**: The netmask associated with the VPC gateway.
- **VLAN**: The VLAN associated with the VPC gateway.

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

13.19.6. 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 created are listed.
5. Click the Add VM button of the tier for 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 Adding Instances section in the Installation Guide.

13.19.7. 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.
5. Click the Settings icon.

   The following options are displayed.

   - IP Addresses

- Gateways
- Site-to-Site VPN
- Network ACLs

6. Select IP Addresses.

   The IP Addresses page is displayed.

7. 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.

13.19.8. 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.

5. Click the Settings icon.

   The following options are displayed.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN
   - Network ACLs

6. Select IP Addresses.

   The IP Addresses page is displayed.

7. Click the IP you want to release.

8. In the Details tab, click the Release IP button 

13.19.9. 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.

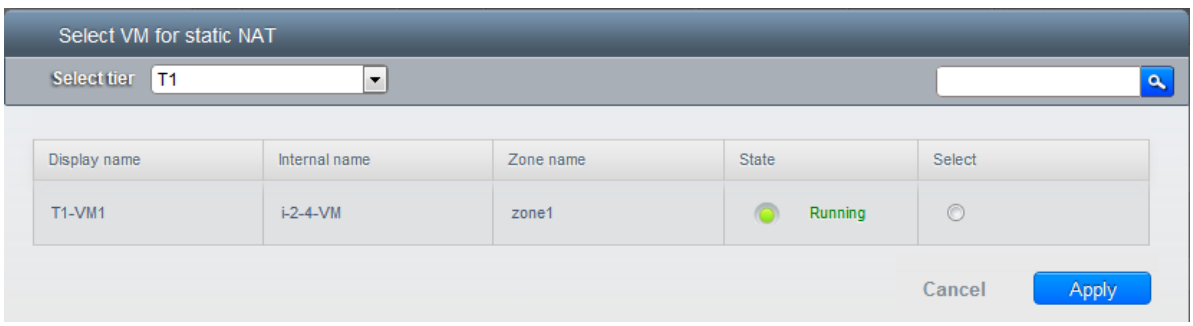
   All the VPCs that you have created for the account is listed in the page.
4. Click the Configure button of the VPC to which you want to deploy the VMs.

   The VPC page is displayed where all the tiers you created are listed in a diagram.
5. Click the Settings icon.

   The following options are displayed.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN
   - Network ACLs
6. Select IP Addresses.

   The IP Addresses page is displayed.
7. Click the IP you want to work with.
8. 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.
9. If you are enabling static NAT, a dialog appears as follows:



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

13.19.10. Adding Load Balancing Rules on a VPC

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 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 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.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN
   - Network ACLs
6. Select IP Addresses.

   The IP Addresses page is displayed.
7. Click the IP address for which you want to create the rule, then click the Configuration tab.
8. In the Load Balancing node of the diagram, click View All.
9. Select the tier to which you want to apply the rule.

   Note

   In a VPC, the load balancing service is supported only on a single tier.

10. 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.

13.19.11. 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.
5. Click the Settings icon.

   The following options are displayed.

   - IP Addresses
   - Gateways
   - Site-to-Site VPN
   - Network ACLs
6. Choose an existing IP address or acquire a new IP address. Click the name of the IP address in the list.

   The IP Addresses page is displayed.
7. Click the IP address for which you want to create the rule, then click the Configuration tab.
8. In the Port Forwarding node of the diagram, click View All.
9. Select the tier to which you want to apply the rule.
10. 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.

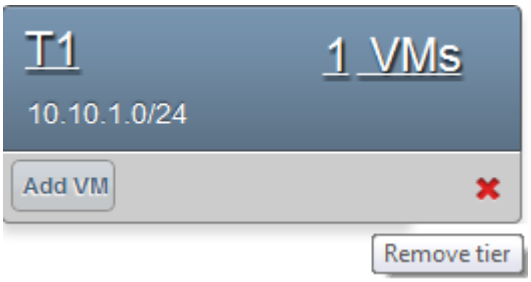
13.19.12. Removing Tiers

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. Click the Remove VPC button:

   Wait for some time for the tier to be removed.

13.19.13. Editing, Restarting, and Removing a Virtual Private Cloud

Note

Ensure that all the tiers are removed before you remove a VPC.

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

3. In the Select view, select VPC.

   All the VPCs that you have created for the account is listed in the page.

4. Select the VPC you want to work with.

5. To remove, click the Remove VPC button 

   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. 

Appendix A. Revision History

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

Initial publication