

Apache CloudStack 4.1.1

Quick Install Guide

Prescriptive instructions for deploying Apache CloudStack

Edition 0



Apache CloudStack

Legal Notice

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

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.

CloudStack® is a registered trademark of the Apache Software Foundation.

Apache CloudStack, the CloudStack word design, the Apache CloudStack word design, and the cloud monkey logo are trademarks of the Apache Software Foundation.

Abstract

This guide is designed to provide a strict environment to guarantee a higher degree of success in initial deployments of Apache CloudStack. All of the elements of the environment will be provided to you. Apache CloudStack is capable of much more complex configurations, but they are beyond the scope of this document.

Preface

1. Document Conventions

- 1.1. Typographic Conventions
- 1.2. Pull-quote Conventions
- 1.3. Notes and Warnings

2. Submitting Feedback and Getting Help

1. Overview

- 1.1. What exactly are we building?
- 1.2. High level overview of the process
- 1.3. Prerequisites

2. Environment

2.1. Operating System

- 2.1.1. Configuring the network
- 2.1.2. Hostname
- 2.1.3. SELinux
- 2.1.4. NTP
- 2.1.5. Configuring the CloudStack Package Repository

2.2. NFS

3. Installation of the management server

- 3.1. Database Installation and Configuration
- 3.2. Installation
- 3.3. System Template Setup

4. KVM Setup and installation

- 4.1. Prerequisites
- 4.2. Installation
 - 4.2.1. KVM Configuration

5. Configuration

- 5.1. UI Access
- 5.2. Setting up a Zone
- 5.3. Pod Configuration
- 5.4. Cluster
 - 5.4.1. Primary Storage
 - 5.4.2. Secondary Storage

A. Revision History

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the [Liberation Fonts](#) set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to the first virtual terminal. Press **Ctrl+Alt+F1** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or Proportional Bold Italic

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;
import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo             echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session.

or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.



Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Submitting Feedback and Getting Help

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a bug: <https://issues.apache.org/jira/browse/CLOUDSTACK> against the component **Doc**.

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Better yet, feel free to submit a patch if you would like to enhance the documentation. Our documentation is, along with the rest of the Apache CloudStack source code, kept in the project's git repository.

The most efficient way to get help with Apache CloudStack is to ask on the mailing lists.

The Apache CloudStack project has mailing lists for users and developers. These are the official channels of communication for the project and are the best way to get answers about using and contributing to CloudStack. It's a good idea to subscribe to the users@cloudstack.apache.org mailing list if you've deployed or are deploying CloudStack into production, and even for test deployments.

The CloudStack developer's mailing list (dev@cloudstack.apache.org) is for discussions about CloudStack development, and is the best list for discussing possible bugs in CloudStack. Anyone contributing to CloudStack should be on this mailing list.

To posts to the lists, you'll need to be subscribed. See the [CloudStack Web site](#) for instructions.

Chapter 1. Overview

1.1. What exactly are we building?

1.2. High level overview of the process

1.3. Prerequisites

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

1.1. What exactly are we building?

This runbook will focus on building a CloudStack cloud using KVM with CentOS 6.4 with NFS storage on a flat layer-2 network utilizing layer-3 network isolation (aka Security Groups), and doing it all on a single piece of hardware.

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

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

1.2. High level overview of the process

Before we actually get to installing CloudStack, we'll start with installing our base operating system, and then configuring that to act as an NFS server for several types of storage. We'll install the management server, download the system VMs, and finally install the agent software. Finally we'll spend a good deal of time configuring the entire cloud in the CloudStack web interface.

1.3. Prerequisites

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

1. At least one computer which supports hardware virtualization.
2. The [CentOS 6.4 x86_64 minimal install CD](#)
3. A /24 network with the gateway being at xxx.xxx.xxx.1, no DHCP should be on this network and none of the computers running CloudStack will have a dynamic address. Again this is done for the sake of simplicity.

Chapter 2. Environment

Chapter 2. Environment

2.1. Operating System

- 2.1.1. Configuring the network
- 2.1.2. Hostname
- 2.1.3. SELinux
- 2.1.4. NTP
- 2.1.5. Configuring the CloudStack Package Repository

2.2. NFS

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

2.1. Operating System

Using the CentOS 6.4 x86_64 minimal install ISO, you'll need to install CentOS on your hardware. The defaults will generally be acceptable for this installation.

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

2.1.1. Configuring the network

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

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

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

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



Hardware Addresses

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

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



IP Addressing

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

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

```
# chkconfig network on
```

```
# service network start
```

2.1.2. Hostname

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

```
# hostname --fqdn
```

At this point it will likely return:

```
localhost
```

```
localhost
```

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

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

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

```
# service network restart
```

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

2.1.3. SELinux

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

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

```
# setenforce 0
```

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

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

2.1.4. NTP

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

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

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

```
# chkconfig ntpd on
```

```
# service ntpd start
```

2.1.5. Configuring the CloudStack Package Repository

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

 **The below repository is not an official Apache CloudStack project repository**

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

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

```
[cloudstack]
name=cloudstack
baseurl=http://cloudstack.appt-get.eu/rhel/4.1/
enabled=1
gpgcheck=0
```

2.2. NFS

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

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

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

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

You will note that we specified two directories that don't exist (yet) on the system. We'll go ahead and create those

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

```
# mkdir /primary
# mkdir /secondary
```

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

```
Domain = cloud.priv
```

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

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

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

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

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

```
# service iptables restart
```

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

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

Chapter 3. Installation of the management server

3.1. Database Installation and Configuration

3.2. Installation

3.3. System Template Setup

Now it is time to start installing CloudStack's management server and some of the related components.

3.1. Database Installation and Configuration

We'll start out by installing **MySQL** and configuring some options to ensure CloudStack runs well.

To install **MySQL** run the following command:

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

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

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

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

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

3.2. Installation

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

```
# yum -y install cloud-client
```

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

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

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

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

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

3.3. System Template Setup

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

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

```
# /usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-tmplt -m  
/secondary -u http://download.cloud.com/templates/acton/acton-systemvm-02062012.qcow2.bz2 -  
h kvm -F
```

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

Chapter 4. KVM Setup and installation

4.1. Prerequisites

4.2. Installation

4.2.1. KVM Configuration

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

4.1. Prerequisites

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

1. [Section 2.1.1, "Configuring the network"](#)
2. [Section 2.1.2, "Hostname"](#)
3. [Section 2.1.3, "SELinux"](#)
4. [Section 2.1.4, "NTP"](#)
5. [Section 2.1.5, "Configuring the CloudStack Package Repository"](#)

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

4.2. Installation

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

```
# yum -y install cloud-agent
```

4.2.1. KVM Configuration

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

4.2.1.1. QEMU Configuration

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

```
vnc_listen=0.0.0.0
```

4.2.1.2. Libvirt Configuration

CloudStack uses libvirt for managing virtual machines. Therefore it is vital that libvirt is configured correctly. Libvirt is a

cloudstack-vm-agent for managing guest machines, therefore the libvirt package is configured correctly, ensuring 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 parameters:

```
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:

```
#LIBVRTD_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
```

4.2.1.3. KVM configuration complete

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

Chapter 5. Configuration

5.1. UI Access

5.2. Setting up a Zone

5.3. Pod Configuration

5.4. Cluster

5.4.1. Primary Storage

5.4.2. Secondary Storage

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

5.1. UI Access

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

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

5.2. Setting up a Zone

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

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

5. Internal DNS2 - we will also set this to '8.8.8.4' for our cloud.



Notes about DNS settings

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

5.3. Pod Configuration

Now that we've added a Zone, the next step that comes up is a prompt for information regarding a pod. Which is looking for 4 items.

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

5.4. Cluster

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

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

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

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

5.4.1. Primary Storage

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

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

5.4.2. Secondary Storage

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

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

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

Revision History

Revision 0-0

Initial creation of book by publican

Fri Jun 28 2013