Table of Contents

ABOUT THIS GUIDE .............................................................................................................................. 5
LABORATORY INSTALLATION GUIDE .................................................................................................... 7
1.1 HARDWARE REQUIREMENTS ............................................................................................................. 8
1.2 SOFT REQUIREMENTS ......................................................................................................................... 9
1.3 INFRASTRUCTURE DESIGN .................................................................................................................. 9
1.4 MANAGEMENT SERVER SETUP OVERVIEW .................................................................................... 10
1.5 PREPARING THE MANAGEMENT SERVER OS ................................................................................. 10
1.6 CLOUDSTACK INSTALLATION ........................................................................................................... 12
1.7 PREPARE NFS SHARES ....................................................................................................................... 12
1.8 INSTALLATION OF DATABASE SERVER .......................................................................................... 14
1.9 KVM HYPERVISOR SETUP ................................................................................................................. 15
1.10 PREPARING THE OPERATING SYSTEM ............................................................................................ 15
1.11 INSTALLATION AND CONFIGURATION OF THE CLOUDSTACK AGENT ....................................... 16
1.12 INSTALL AND CONFIGURE LIBVIRT ............................................................................................... 17
1.13 CONFIGURING THE NETWORKING ................................................................................................. 17
1.14 ADDING ZONE ................................................................................................................................... 20

DOMAIN HIERARCHY TO MANAGE USER ACCOUNTS AND RESOURCE LIMITS ................................. 34
1.15 OVERVIEW ........................................................................................................................................ 35
1.16 TOPICS COVERED ............................................................................................................................. 35
1.17 ACCESSING THE CONTROL PANEL ............................................................................................... 35
1.18 CREATING A DOMAIN .................................................................................................................... 36
1.19 LIMITING RESOURCES AT DOMAIN LEVEL .................................................................................... 37
1.20 ADDING A DOMAIN ADMINISTRATOR ACCOUNT .......................................................................... 38
1.21 ADDING A USER ACCOUNT AND SETTING LIMITS ......................................................................... 40

LAUNCHING, RESIZING, CONFIGURING, AND MANAGING INSTANCES ............................................ 42
1.22 OVERVIEW ........................................................................................................................................ 43
1.23 TOPICS COVERED ............................................................................................................................. 43
1.24 ACCESSING THE CONTROL PANEL ............................................................................................... 43
1.25 REGISTERING A NEW TEMPLATE ....................................................................................................... 43
1.26 CREATE A GUEST NETWORK AND MANAGE EGRESS RULES ....................................................... 46
1.27 LAUNCHING A WEB SERVER ........................................................................................................... 49
1.28 ENABLING HTTP PORT REDIRECT TO EXPOSE THE WEB SERVER TO PUBLIC ACCESS .......... 50
1.29 SCALING UP/DOWN INSTANCES RESOURCES ............................................................................ 52
1.30 REMOVING AND RECOVERING INSTANCES .................................................................................. 55

VIRTUAL PRIVATE CLOUD .................................................................................................................... 60
1.31 OVERVIEW ........................................................................................................................................ 61
1.32 TOPICS COVERED ............................................................................................................................. 61
1.33 ARCHITECTURE ................................................................................................................................. 61
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.34</td>
<td>Creating a VPC</td>
<td>61</td>
</tr>
<tr>
<td>1.35</td>
<td>Creating ACL Lists</td>
<td>63</td>
</tr>
<tr>
<td>1.36</td>
<td>Creating Tiers</td>
<td>67</td>
</tr>
<tr>
<td>1.37</td>
<td>Creating Backend Instances</td>
<td>69</td>
</tr>
<tr>
<td>1.38</td>
<td>Creating and Configuring the Internal Load Balancer</td>
<td>70</td>
</tr>
<tr>
<td>1.39</td>
<td>Creating Frontend Instances</td>
<td>73</td>
</tr>
<tr>
<td>1.40</td>
<td>Creating and Configuring the External Load Balancer</td>
<td>74</td>
</tr>
<tr>
<td>1.41</td>
<td>Accessing the Web Service Hosted Within the VPC</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td><strong>SUMMARY</strong></td>
<td>80</td>
</tr>
</tbody>
</table>
About this Guide
Conducting a PoC of new technology can be a complex task. You need to ensure that you are doing everything correctly to be confident that the technology will fit your use case and business needs. Testing a new virtualization management solution follows a specific approach and must-do steps. This Lab Guide offers a series of use cases designed to aid in the proof-of-concept evaluation process. This POC guide gives you all the guidance you need to perform a successful Proof of Concept of Apache CloudStack. By following this guide, you will be prepared for what to expect from the technology and how to move into production. The guide gives detailed configuration information.

At the end of this PoC, you will have a highly available, reliable and flexible CloudStack-powered cloud. Following this guide should allow you to feel confident enough in setting up and managing a CloudStack IaaS environment and should give a smooth implementation of the cloud orchestration layer in your infrastructure.

Let’s get started!

**Conventions**

The following conventions are used to highlight important areas and necessary inputs:

- Highlight a section of interest.

**Highlighted text** that is related to an important area in the GUI or Component/Feature.

- Highlight a button/item from the list in the GUI that requires direct interaction.

- Numerical sequence when many steps are illustrated by a single screenshot.

**Highlighted text** which can be copied and pasted directly into the GUI

**Highlight text which can be copied and pasted directly into the command prompt**

**Text highlighted from command prompt output.**

- Highlights an observation that provides additional information.

- Highlights a warning that provides additional information.
Laboratory Installation Guide
Warning

This is a basic Apache CloudStack setup to be used only for a Proof-of-Concept purpose.

Hardware requirements

In order to have a working architecture for Apache CloudStack evaluation, the following hardware will be minimally required.

Storage

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary and Secondary storage created as NFS exports on the CloudStack Management server</td>
<td></td>
</tr>
<tr>
<td>500GB of RAID based storage on CloudStack Management server</td>
<td></td>
</tr>
</tbody>
</table>

Hosts

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. hosts (per cluster required)</td>
<td>3</td>
</tr>
<tr>
<td>Clusters/Pods (1 cluster per pod)</td>
<td>1</td>
</tr>
<tr>
<td>No. cores (per host)</td>
<td>8</td>
</tr>
<tr>
<td>Memory (per host)</td>
<td>32-64GB</td>
</tr>
<tr>
<td>Local storage</td>
<td>Disk to support Hypervisor/OS</td>
</tr>
<tr>
<td>Network Interfaces</td>
<td>2 Ethernet cards</td>
</tr>
<tr>
<td>Network throughput</td>
<td>1Gb/s</td>
</tr>
</tbody>
</table>

Management Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. cores</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>16GB</td>
</tr>
<tr>
<td>Local storage</td>
<td>150GB for OS + 500GB for primary and secondary storage of RAID based storage</td>
</tr>
<tr>
<td>Network Interfaces</td>
<td>1 Ethernet card</td>
</tr>
<tr>
<td>Network throughput</td>
<td>1Gb/s</td>
</tr>
</tbody>
</table>

Networking

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Switches</td>
<td>1</td>
</tr>
<tr>
<td>VLAN</td>
<td>802.1q support required for advanced network zones.</td>
</tr>
<tr>
<td>No. ports</td>
<td>Enough ports to connect 2 interfaces on each host considering 1 port for public/guest networks and 1 port for storage/management (+) 1 interface for storage/management network for the CloudStack management server.</td>
</tr>
<tr>
<td>Throughput</td>
<td>1 Gb/s</td>
</tr>
</tbody>
</table>
Liberatory Install Guide

**Soft Requirements**

**IP Address / VLAN Space**

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity/Description</th>
<th>Network IP Address</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Network</td>
<td>0 public addresses – /24 network IP range - RFC 1918 addresses routable within POC environment</td>
<td>10.0.48.0/24</td>
<td>48</td>
</tr>
<tr>
<td>Guest Network</td>
<td>20x VLANS dedicated to CloudStack use.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Management and Storage Network</td>
<td>40x RFC 1918 addresses</td>
<td>10.0.32.0/24</td>
<td>49</td>
</tr>
</tbody>
</table>

**Hostname and IP addresses**

<table>
<thead>
<tr>
<th>Host</th>
<th>hostname</th>
<th>IP Address</th>
<th>Netmask</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudStack Management Server</td>
<td>mgmt..local</td>
<td>10.0.32.100</td>
<td>255.255.240.0</td>
</tr>
<tr>
<td>KVM Host 01</td>
<td>host01.local</td>
<td>10.0.33.1</td>
<td>255.255.240.0</td>
</tr>
<tr>
<td>KVM Host 02</td>
<td>host02.local</td>
<td>10.0.33.2</td>
<td>255.255.240.0</td>
</tr>
<tr>
<td>KVM Host 03</td>
<td>host03.local</td>
<td>10.0.33.3</td>
<td>255.255.240.0</td>
</tr>
<tr>
<td>Reserved System VM</td>
<td>-</td>
<td>10.0.34.1 – 10.0.34.40</td>
<td>255.255.240.0</td>
</tr>
</tbody>
</table>

**Infrastructure Design**

**Physical Layout**

![Physical Layout Diagram](image-url)
Logical Layout

Management Server Setup Overview

- **Warning**
  To install the management server, make sure you have a server described in the hardware requirements.

- **Warning**
  In this guide, will be used Linux CentOS 8 for management servers and hosts. Make sure you have sufficient space to store the OS and the secondary storage mount point.

- **Warning**
  Before continuing, make sure that you have applied the latest updates to your host.

Preparing the Management Server OS

1. Log in to manager server OS as root.

2. Edit the `/etc/hosts` file and add the following lines. If you prefer, you can add these entries in your internal DNS server.
3. Now, check for a fully qualified hostname.

```
hostname-f
```

**Note**
This should return a fully qualified hostname as “mgmt.local”.

4. Turn on NTP for time synchronization.

```
yum -y install chrony
systemctl enable --now chronyd
```

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

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

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

Change the following line

```
SELINUX=enforcing
```

To this

```
SELINUX=permissive
```

6. Set SELinux to permissive starting immediately, without requiring a system reboot, running the following command.

```
setenforce permissive
```

7. For a proof-of-concept propose, firewalld/iptables will not be necessary. To disable it, run the following command.

```
systemctl stop firewalld
systemctl disable firewalld
```
CloudStack Installation

1. Add the CloudStack repository creating `/etc/yum.repos.d/cloudstack.repo` file and inserting the following information.

```
[cloudstack]
name=cloudstack
baseurl=http://download.cloudstack.org/centos/$releasever/4.16/
enabled=1
gpgcheck=1
gpgkey=https://download.cloudstack.org/RPM-GPG-KEY
```

2. Now, to install cloudstack, run the following command.

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

**Warning**

CloudStack 4.16 requires Java 11 JRE. Installing CloudStack packages will automatically install Java 11, but it’s good to explicitly confirm that the Java 11 is the selected/active one (in case you had a previous Java version already installed) with `alternatives --config java` after Apache CloudStack packages are already installed.

**Note**

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

Prepare NFS Shares

1. Install the `nfs-utils` package.

```
yum -y install nfs-utils quota-rpc
```

2. Create a NFS share for primary and secondary storage running the command as follows.

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

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

```
vi /etc/exports
```

4. Insert the following line.

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

5. Now, export the `/export` directory running the following command.

```
exportfs -a
```

Laboratory Install Guide

vi /etc/nfs.conf

Ensure the parameters are like the following lines.

```plaintext
[general]
[exportfs]
[gssd]
use-gss-proxy=1
[lockd]
port=32803
udp-port=32769
[mountd]
port=892
[nfsdcl]
[nfsdcltrack]
[nfsd]
[statd]
port=662
outgoing-port=2020
[sm-notify]
```

7. Also, edit the /etc/sysconfig/rpc-rquotad file:

`vi /etc/sysconfig/rpc-rquotad`

Ensure the parameters are like the following line.

```
RPCRQUOTADOPTS="-p 875"
```

8. Enable and start the services.

```bash
systemctl enable --now nfs-server.service
systemctl enable --now rpc-rquotad
systemctl enable nfs-server.service
systemctl start rpc-rquotad
```

1. **Note**

After restarting nfs and rpcbind, only these seven ports are needed for setting up NFS server.

9. The ports used by NFS RPC-based service can be listed by:

`rpcinfo -p`

This is a sample output of this command:
### Installation of Database Server

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

1. Install mysql-server running the following command:
   ```bash
   yum -y install mysql-server
   systemctl enable --now mysqld
   ```
2. Open the `/etc/my.cnf.d/mysql-server.cnf` configuration file.
3. Insert the following lines in the `[mysqld]` section.
   ```ini
   [mysqld]
   server_id=1
   innodb_rollback_on_timeout=1
   innodb_lock_wait_timeout=600
   max_connections=350
   log-bin=mysql-bin
   binlog-format = 'ROW'
   ```
4. Now, start the MySQL server to put the new configuration into effect.
   ```bash
   systemctl start mysqld
   ```
5. Run the following command to setup a CloudStack database.
   ```bash
   cloudstack-setup-databases cloud:password@localhost --deploy-as=root
   ```

   It will set the database with the following informations.

<table>
<thead>
<tr>
<th>Database</th>
<th>User</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloud</td>
<td>cloud</td>
<td>password</td>
</tr>
</tbody>
</table>
6. Now, configure the OS and start the Management Server:

   - `cloudstack-setup-management`
   - `systemctl enable cloudstack-management`
   - `systemctl start cloudstack-management`

   **Note**
   The Management Server should now be running.

7. Secondary storage must be seeded with a template that is used for CloudStack system VMs. This process will need up to 30 minutes to run. To seed the template, run the following command:

```
```

**KVM Hypervisor Setup**

**Note**
To install the KVM hosts, make sure you have a host described in the hardware requirements.

**Note**
Before continuing, make sure that you have applied the latest updates to your host.

**Warning**
Repeat all of following steps on every hypervisor host.

The procedure for installing a KVM Hypervisor Host is:
- Prepare the Operating System
- Install and configure libvirt
- Configure Security Policies (SELinux)
- Install and configure the Agent

**Preparing the Operating System**

1. Log in to your OS as root.

2. Open `/etc/hosts` and add the following lines. If you prefer, you can add these entries in your internal DNS server.

```
10.0.32.10 mgmt.local mgmt secondary-storage.local primary-storage.local
10.0.33.1 host01.local host01
10.0.33.2 host02.local host02
10.0.33.3 host03.local host03
```

3. Now, check for a fully qualified hostname.

```
hostname -f
```
4. Turn on NTP for time synchronization.
   
yum -y install chrony
   systemctl enable --now chronyd

5. Set the SELINUX variable in /etc/selinux/config to **permissive**. This ensures that the permissive setting will be maintained after a system reboot.
   
   vi /etc/selinux/config
   Change the following line
   
   SELINUX=enforcing
   To this
   
   SELINUX=permissive

6. Set SELinux to permissive starting immediately, without requiring a system reboot, running the following command.
   
   setenforce permissive

7. For a proof-of-concept propose, firewalld/iptables will not be necessary. To disable it, run the following command.
   
   yum -y install iptables-services
   systemctl stop firewalld
   systemctl disable firewalld

### Installation and Configuration of the CloudStack Agent

1. Add the CloudStack repository by creating `/etc/yum.repos.d/cloudstack.repo` file and inserting the following information.
   
   [cloudstack]
   name=cloudstack
   baseurl=http://download.cloudstack.org/centos/$releasever/4.16/
   enabled=1
   gpgcheck=1
   gpgkey=https://download.cloudstack.org/RPM-GPG-KEY

2. Now, install the cloudstack-agent.
   
   yum -y install -y epel-release
   yum -y install cloudstack-agent
CloudStack 4.16 requires Java 11 JRE. Installing CloudStack packages will automatically install Java 11, but it’s good to explicitly confirm that the Java 11 is the selected/active one (in case you had a previous Java version already installed) with `alternatives --config java` after CloudStack packages are already installed.

## Install and Configure libvirt

### Note

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

### Note

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`

1. Open the file `/etc/libvirt/libvirtd.conf` and set the following parameters.

   ```
   Listen_tls = 0
   listen_tcp = 1
   tcp_port = 16509
   auth_tcp = "none"
   mdns_adv = 0
   ```

### Note

Turning on “listen_tcp” in libvirtd.conf is not enough, we have to change the parameters as well modifying `/etc/sysconfig/libvirtd`.

2. As Linux CentOS 8 comes with systemd socket activation effectively breaking `libvirtd-–listen`, disable that running the following command:

   ```
   systemctl mask libvirtd.socket libvirtd-ro.socket libvirtd-admin.socket libvirtd-tls.socket libvirtd-tcp.socket
   ```

3. Open `/etc/sysconfig/libvirtd` and uncomment the following line:

   ```
   LIBVIRTD_ARGS"--listen"
   ```

4. Restart libvirt.

   ```
   systemctl restart libvirtd
   ```

## Configuring the Networking

### Warning

This is a very important section, please make sure you read this thoroughly.
CloudStack uses the network bridges in conjunction with KVM to connect the guest instances to each other and the outside world. They also are used to connect the System VMs to your infrastructure. By default, these bridges are called cloudbr0 and cloudbr1.

Ensure that the interfaces names to be used for configuring the bridges match one of the following patterns: 'eth*', 'bond*', 'team*', 'vlan*', 'em*', 'p*p*', 'ens*', 'eno*', 'enp*', 'enx*'. Otherwise, the KVM agent will not be able to configure the bridges properly.

It is essential that you keep the configuration consistent across all your hypervisors.

In the Advanced networking mode, the most common case is to have (at least) two physical interfaces per hypervisor-host. We will use the interface eth0 linked to the bridge 'cloudbr0' using the untagged (native) VLAN for hypervisor management. Additionally, we configure the second interface for usage with the bridge 'cloudbr1' for public and guest traffic. This time there are no VLANs applied by us. CloudStack will add the VLANs as required during actual use.

1. Install the bridge-utils by running the following command:
   ```
   yum -y install bridge-utils
   ```

2. Configure the eth0 network interface.
   ```
   vi /etc/sysconfig/network-scripts/ifcfg-eth0
   ```
   Make sure it looks similar to:
   ```
   DEVICE=eth0
   HWADDR=00:04:xx:xx:xx:xx
   ONBOOT=yes
   HOTPLUG=no
   BOOTPROTO=none
   TYPE=Ethernet
   BRIDGE=cloudbr0
   ```

3. We now have to configure the second network-interface for use in public and guest VLANs:
   ```
   vi /etc/sysconfig/network-scripts/ifcfg-eth1
   ```
   Make sure it looks similar to:
   ```
   DEVICE=eth1
   HWADDR=00:04:xx:xx:xx:xx
   ONBOOT=yes
   HOTPLUG=no
   BOOTPROTO=none
   TYPE=Ethernet
   BRIDGE=cloudbr1
   ```
4. Now we configure cloudbr0 and include the Management IP of the hypervisor.

```
vi /etc/sysconfig/network-scripts/ifcfg-cloudbr0
```

Make sure it looks similar to:

```
DEVICE=cloudbr0
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=None
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
IPADDR=10.0.33.x
GATEWAY=10.0.32.1
NETMASK=255.255.240.0
STP=yes
```

5. Configure ‘cloudbr1’ as a plain bridge without an IP address or dedicated VLAN configuration.

```
vi /etc/sysconfig/network-scripts/ifcfg-cloudbr1
```

Make sure it looks similar to:

```
DEVICE=cloudbr1
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=None
IPV6INIT=no
IPV6_AUTOCONF=no
DELAY=5
STP=yes
```

6. Run the following command to restart the network:

```
systemctl restart NetworkManager
```

7. Run the following command to show the bridge configuration and see if everything is right:
**brctl show**

The command should return the following output:

<table>
<thead>
<tr>
<th>bridge name</th>
<th>bridge id</th>
<th>STP enabled</th>
<th>interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloudbr0</td>
<td>8000.1e0032000265</td>
<td>yes</td>
<td>eth0</td>
</tr>
<tr>
<td>cloudbr1</td>
<td>8000.020045c625cb</td>
<td>yes</td>
<td>eth1</td>
</tr>
<tr>
<td>virbr0</td>
<td>8000.525400774409</td>
<td>yes</td>
<td>virbr0-nic</td>
</tr>
</tbody>
</table>

Adding Zone

**Note**

Now we will create each one of the components represented below.

1. To access the CloudStack UI, open this URL http:// 10.0.32.10:8080/client in your web browser.
2. Login with the following credentials and click *Login*.

    Username: admin
    Password: password
    Domain: Blank
3. In the left part of the navigation pane, click **Infrastructure > Zones** and then click **Add Zone**.

4. In the wizard, select **Advanced** to create an Advanced Zone and click **Next**.
5. Fill in the informations required for Zone Details as follows and then click Next.

- Name: poc-zone
- IPv4 DNS1: 8.8.8.8
- Internal DNS 1: 8.8.4.4
- Hypervisor: KVM
- Guest CIDR: 10.1.1.0/24

Note
For more info about Advanced Zone, you can reach here:
6. On this screen, we will configure traffic types for the hosts’ physical networks. Click *Add Physical Network* to create a new physical network and configure the *Traffic Types* as follows:
8. Click *edit* in each *traffic type* item and set the *traffic label* as follows and, after all traffic label was defined, click *Next*.

<table>
<thead>
<tr>
<th>Name</th>
<th>Traffic Type</th>
<th>Traffic label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Network 1</td>
<td>Management</td>
<td>cloudbr0</td>
</tr>
<tr>
<td>Physical Network 2</td>
<td>Public, Guest</td>
<td>cloudbr1</td>
</tr>
</tbody>
</table>
9. Now we will configure the Public traffic and add public IPs to be used by the Virtual Routers and System VMs in the public NIC. Fill in the form as follows, click Add and then click Next.
10. Fill in the form to create and setup the POD and then click Next.

**Note**
A Pod represents a Rack in the Data Center. The network IPs used will be from the management network.

- **Pod name:** POD1
- **Rerverd system gateway:** 10.0.32.1
- **Rerverd system netmask:** 255.255.240.0
- **Start Rerverd system IP:** 10.0.34.1
- **End Rerverd system IP:** 10.0.34.40
Each zone must contain one or more pods, and we will add the first pod now. A pod contains hosts and primary storage servers, which you will add in a later step. First, configure a range of reserved IP addresses for CloudStack’s internal management traffic. The reserved IP range must be unique for each zone in the cloud.

- **Pod name**: POD1
- **Reserved system gateway**: 10.0.32.1
- **Reserved system netmask**: 255.255.240.0
- **Start Reserved system IP**: 10.0.34.1
- **End Reserved system IP**: 10.0.34.40
11. Now, configure the *Guest Traffic* VLANs and then click *Next*.

**VLAN/VNI Range:** 101 - 120

12. Fill in the form with the *cluster name* and then click *Next*.

**Cluster Name:** cluster01
13. Add the first host and then click Next. The others hosts will be added after the Zone is created.

- **Host Name:** host01.local
- **Username:** root
- **Password:** <the root password>
- **Tags:** Blank

14. Fill in the form as follows to add a NFS Primary Storage and then click Next.

- **Name:** Primary Storage
- **Scope:** Cluster
- **Protocol:** nfs
- **Server:** 192.168.32.10
- **Path:** /export/primary
- **Storage Tags:** Blank
15. Fill in the form as follows to add a NFS Secondary Storage and then click Next.

- Provider: Secondary Storage
- Name: Cluster
- Server: 192.168.32.10
- Path: /export/secondary
16. Now, click *Launch Zone* to proceed.

The Zone will be ready after all resources is configured. Click *Enable Zone* to finish the wizard.
17. We need now to add the remaining hosts. In the left navigation pane, click *Infrastructure* > *Hosts* and then click *Add Host.*
18. Fill in the form as follows and click OK.

- **Zone Name**: Poc Zone
- **Pod name**: POD1
- **Cluster name**: cluster01
- **Host Name**: host02.local
- **Username**: root
- **Password**: <The root password>
- **Host Tags**: Blank

Repeat this step to add the host03.
Domain Hierarchy to Manage User Accounts and Resource Limits
Overview

CloudStack implements domain hierarchies to logically isolate the user accounts. This model can be used to define, for example, departments within the same organization if used to build private clouds or, different customers when it used to build public clouds. You could also have a specific domain for sales partners or customers for example.

A domain can contain multiple user accounts. In Apache CloudStack, a user account assumes a profile defined in roles. There is a set of predefined roles for the most common use cases, these being the main:

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root Admin</td>
<td>Manage the entire platform, including physical and logical resources from all domains and accounts.</td>
</tr>
<tr>
<td>Domain Admin</td>
<td>Manage all logical resources under the domain and adjacent sub-domains including user accounts, sub-domains and all related virtual computing resources.</td>
</tr>
<tr>
<td>User</td>
<td>Manage virtual computing resources related to your own account.</td>
</tr>
</tbody>
</table>

Topics covered

- Managing Domain
- Setting resource limits
- Managing User Accounts

Accessing the Control Panel

18. Open your browser and access the CloudStack Panel at http://192.168.32.10:8080/client.

19. Login with the following credentials and click Login.

Username: admin
Password: password
Domain: Blank
Domain Hierarchy to Manage User Accounts and Resource Limits

Creating a Domain

**Note**
In the domain and user accounts level, it is possible to distribute the computational resources to be used for the users, limiting their compute capacity. When a domain account is allocated at the domain level, the domain administrator user can distribute the resources for other regular user accounts and the adjacent sub-domains. It allows users to have granular control over allocated resources in many hierarchical levels.

20. In the left navigation pane, click *Domains*.

21. Now click *Add Domain*.
22. Fill in the form as follows and then click \textit{OK}:

Name: \textit{POC}

- Network Domain: Blank
- Domain: Blank

Limiting resources at Domain level

23. Expand the \textit{domain tree} and click \textit{POC} and then click \textit{Configure Limits}.

\begin{itemize}
  \item \textbf{Note}\textbf{\textit{\footnotesize}}
  \footnotesize This screen shows the computing resources limits in the domain level. By default, the resources are configured with \textit{-1}, that means \textit{unlimited resource}.
\end{itemize}

24. Set up the domain resources as follows and then click \textit{Submit}:

\begin{align*}
\text{Max. User VMs:} & \quad 10 \\
\text{Max. Public IPs:} & \quad 5 \\
\text{Max. Volumes:} & \quad -1 \\
\text{Max. Snapshots:} & \quad -1 \\
\text{Max. Templates:} & \quad -1
\end{align*}
Domain Hierarchy to Manage User Accounts and Resource Limits

Max. Networks: 5
Max. VPCs: 2
Max. CPU Cores: -1
Max. Memory (MiB): 4096
Max. Primary Storage (GiB): 50
Max. Secondary Storage (GiB): -1

Adding a Domain Administrator Account

25. In the left of navigation pane, click Accounts and then click Add Account.

26. Fill in the form as follows and then click OK.

Role: Domain Admin (Domain Admin)
Username: admin
Password: password
Confirm Password: password
Email: admin@poc.zone
First Name: Administrator
Last Name: PoC Domain
Domain: ROOT/POC
Account: admin
Timezone: <select your timezone>
Network Domain: Blank
27. Click the admin where the corresponding role is Domain Admin.

28. The domain admin account will be used only for administrative proposes, then no resource will be available for this account. Fill in the form as follows and then click Submit.

Max. User VMs: 0
Max. Public IPs: 0
Max. Volumes: 0
Max. Snapshots: 0
Max. Templates: 0
Max. Networks: 0
Max. VPCs: 0
Max. CPU Cores: 0
Max. Memory (MiB): 0
Max. Primary Storage (GiB): 0
Max. Secondary Storage (GiB): 0

**Adding a User Account and setting Limits**

29. Logout and login with the following credentials:

Username: admin
Password: password
Domain: poc

30. In the left of the navigation pane, click **Accounts** and then click **Add Account**.

31. Fill in the form as follows and then click **OK**:

Role: User
Username: user-1
Password: password
Confirm Password: password
Email: user-1@poc.zone
First Name: User-1
Last Name: PoC Domain
Domain: ROOT/POC
Account: poc-user-account
Timezone: <select yours>
Network Domain: Blank
32. Select *poc-user-account* and then click *Configure Limits*.

33. Fill in the form as follows and then click *Submit*:

- Max. User VMs: 10
- Max. Public IPs: 5
- Max. Volumes: -1
- Max. Snapshots: -1
- Max. Templates: -1
- Max. Networks: 5
- Max. VPCs: 2
- Max. CPU Cores: -1
- Max. Memory (MiB): 4096
- Max. Primary Storage (GiB): 50
- Max. Secondary Storage (GiB): -1
Launching, Resizing, Configuring, and Managing Instances
Overview

This use case provides you with a basic overview of launching, resizing, and managing an Apache CloudStack instance.

Apache CloudStack has a UI that allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Apache CloudStack proven computing environment. Apache CloudStack reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Topics Covered

After you have finished this practical exercise, you will then take the following Proof-of-Concept tests:

- Registering a new template
- Creating an isolated network
- Modifying egress firewall rules
- Launching an instance
- Using Userdata
- Managing Firewall
- Scaling up/down Instance
- Instance console access
- Terminate instance
- Recover instance
- Expunge instance

Accessing the Control Panel

1. Login with the following credentials:

   Username:  user-1
   Password:  password
   Domain: poc

Registering a New Template

Note

An Apache CloudStack Template provides a root disk with pre-installed OS to launch an Instance, which is a virtual machine in the cloud. It may contain a preinstalled and configured application server.

2. In the left of the navigation pane, click Images > Templates and then click Register Template from URL.
Note
You will use a pre-configured template including cloudinit installed and configured that allows you to run commands during the instance startup.

3. Fill in the form as follows and then click OK:
   URL: http://dl.openvm.eu/cloudstack/centos/x86_64/centos-7-kvm.qcow2.bz2
   Name: CentOS 7
   Description: CentOS 7.0(64-bit)
   Zone: poc-zone
   Hypervisor: KVM
   Format: QCOW2
   Root disk controller: osdefaults
   OS Type: CentOS 7.2
   Extratable: None
   Dynamically Scalable: Yes
   Password Enabled: Yes
   HVM: Yes
   Public: None
4. From the previously registered template, select **CentOS 7** to check the download status:

5. Refresh this page clicking *refresh* every 30 seconds until the download is complete.
Create a guest network and manage egress rules

1. **Note**
   In the domain and user accounts level, it is possible to distribute the computational resources to be used for the users, limiting their compute capacity. When a domain account is allocated at the domain level, the domain administrator user can distribute the resources for other regular user accounts and the adjacent sub-domains. It allows users to have granular control over allocated resources in many hierarchical levels.

2. **Note**
   An Apache CloudStack guest network, provides a VLAN-isolated layer that connect the instance network to the Virtual Router gateway.

3. **Note**
   The Virtual Router will do the network communication between instances and the public network. This also controls firewall rules, userdata, metadata, dhcp for instances, load balancing, and TCP/UDP port forwarding.

6. In the left navigation pane, click *Networks > Guest networks* and then click *Add network*.

7. Select *Isolated*, and fill in the form as follows and then click *OK*:
### Launching, Resizing, Configuring and Managing Instances

<table>
<thead>
<tr>
<th>Name:</th>
<th>my-isolated-network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>My Isolated Network</td>
</tr>
<tr>
<td>Zone:</td>
<td>poc-zone</td>
</tr>
<tr>
<td>Network Offering:</td>
<td>Offering for Isolated networks with Source Nat service enabled</td>
</tr>
<tr>
<td>External Id:</td>
<td>Blank</td>
</tr>
<tr>
<td>Gateway:</td>
<td>Blank</td>
</tr>
<tr>
<td>Netmask:</td>
<td>Blank</td>
</tr>
<tr>
<td>Network Domain:</td>
<td>Blank</td>
</tr>
</tbody>
</table>

1. Click **my-isolated-network**

2. **OK**

8. Click **my-isolated-network** and then copy the CIDR address.
9. Click **Egress Rules** and add a rule that allows instances to access the internet as follows and then click **Add**.

   - **Source CIDR:** 10.1.1.0/24
   - **Destination CIDR:** 0.0.0.0/0
   - **Protocol:** All

10. Below you can see the egress rule added.
Launching a Web Server

**Note**
An Apache CloudStack instance is a virtual machine created from a template. In this task you will install a new instance to suit your requirements. This includes the CPU, memory, disk and network requirements to have an instance ready to deliver web services.

**Note**
cloud-init is already installed and configured in the Template. Therefore, cloud-init is designed to make additional settings to the instance's operating system at boot time. It extends the integration enabling users to send commands, create users, set password, add SSH keys during the instance boot using the CloudStack userdata and metadata.

11. In the left navigation pane, click *Compute > Instances* and then click *Add Instance*.

12. Fill in the form as follows and then click *Launch Instance*:

- **Zone**: poc-zone
- **Template/ISO**
  - Community: CentOS 7.0 (64bit)
- **Compute Offering**: Small Instance
- **Advanced Mode**: Yes
- **Userdata:**

```bash
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h2>Hello from your new WebServer!</h2></html>' > /var/www/html/index.html
```

Name: WebServer01
Group: None
Keyboard Language: None
Start Instance: Yes

---

**Enabling HTTP port redirect to expose the Web Server to public access**

13. In the left navigation pane, click **Network > Guest networks** and then click *my-isolated-network*.

14. Select **Public IP Address** and then click *source nat IP*.
15. Select **Port Forwarding** and add a rule to redirect the HTTP port as follows and click **Add**.

- **Private Port:** Start: 80 – End: 80
- **Public Port:** Start: 80 – End: 80

16. It will open a list of instances available on the **my-isolated-network** network. Select **WebServer01** and click **OK**.

17. Click **Firewall** to open the HTTP port and fill in the form as follows and then click **Add**.
Launching, Resizing, Configuring and Managing Instances

Source CIDR: 0.0.0.0/0
Protocol: TCP
Start Port: 80
End Port: 80

18. The rule added will be shown in the firewall list.

19. In Details copy the IP address.

20. Open a new tab in the browser and paste the address copied and then press enter:

Hello from your new WebServer!

① Note
The Web Server is now running.

Scaling Up/Down Instances Resources

① Note
Scaling out/in is the ability to scale by adding/removing resource instances (e.g., virtual machine), whereas scaling up/down is the ability to scale by changing allocated resources (e.g., memory/CPU/storage capacity). In Apache CloudStack, the characteristics of an instance are inherited from Compute Offerings, which control CPU capacity, memory, network rate, root disk size, high availability, host and storage tags, Deployment planner, GPU, Zone and sharing level.

② Note
Compute offering can only be created by domain or root administrators. User accounts can only consume them for use in their respective resources.
21. Logout and then login with the following credentials:
   Username: admin
   Password: password
   Domain: poc

22. In the left navigation pane, click Service Offering > Compute Offering and then click Add Compute Offering.

23. Fill in the form as follows and then click OK:

   Name: Poc Instance
   Description: Personal PoC Instance
   Storage Type: Shared
   Provisioning Type: Thin Provisioning
   Write-cache Type: No disk cache
   Compute Offering Type: Fixed Offering
   CPU Cores: 2
   CPU (in Mhz): 1200
   Memory (in MB): 768
   Network Rate (Mb/s): Blank
   Root disk size (GB): Blank
   QoS Type: None
   Offer HA: Yes
   CPU Cap: No
   Volatile: No
24. Logout, and then login with the following credentials:
   Username: user-1
   Password: password
   Domain: poc

25. Stop the instance and then click Scale Instance.
26. Select *Poc Instance* Compute Offering and then click *OK*.

27. The instance will be scaled with the values inherited from the Compute Offering *Poc Instance*.

Removing and Recovering Instances

1 Note
When an Apache CloudStack instance is no longer needed, it can be terminated by any user. By default, the regular user account can only terminate the instance, but can’t remove it permanently from the cloud infrastructure. It is controlled by Global Setting variable “allow.user.view.destroyed.vm”. The resource remains available but, only root or domain admin can recover or eliminate the resource definitively. If no action is taken within the period defined in the Global Setting variable “event.purge.interval”, the event purge thread will eliminate permanently the resource from the infrastructure. The Domain and Account Limits continue to be used until the purge is performed.

28. In the left navigation pane, click *Compute > Instances*, select *WebServer01*, click *Destroy Instance* and then, click *OK*. 
Launching, Resizing, Configuring and Managing Instances

Note
Since the instance was dropped by a regular user account and, although in the user’s view it appears to be permanently excluded, an admin account (root or domain) can recover it.

29. Logout, and then login with the following credentials:
30. Username: admin
    Password: password
    Domain: poc

31. In the left navigation pane, click *Compute > Instances*

Note
A domain admin account can see destroyed instances and can take actions to either purge or recover as seen below:
32. To recover the instance removed by user, click \textit{Recover Instance} to proceed. After this, the instance will be available again.

33. Logout, and then login with the following credentials:
   \begin{itemize}
   \item Username: \textit{user-1}
   \item Password: \textit{password}
   \item Domain: \textit{poc}
   \end{itemize}

34. In the left navigation pane, click \textit{Compute} > \textit{Instances} to view the recovered instance.

35. Select the \textit{WebServer01} instance, followed by clicking \textit{Destroy Instance} and then click \textit{OK}. 
36. Logout, and then login with the following credentials:

   Username: admin
   Password: password
   Domain: poc

37. In the left navigation pane, click Compute > Instances, followed by clicking WebServer01 instance and, click Destroy Instance, and in the dialog box, select Expunge and click OK.
Launching, Resizing, Configuring and Managing Instances

![Image of CloudStack interface]

**Note**
When *Expunge* is selected, the process will be irreversible and no longer possible to recover the instance, unless there is a backup of the instance's root disk.

**Note**
If a data disk is attached to the instance, it will not be purged.
Virtual Private Cloud
Virtual Private Cloud

Overview
Virtual Private Cloud lets you provision an architecture that resembles a traditional physical network.
VPC implements:
- Tiering isolation
- ACL
- Site-to-site IPsec VPN
- Client VPN
- Internal and External Load Balancer

Topics covered
● How to create VPC
● How to create ACLs List
● How to subneting VPC
● How to create internal and external Load Balancer

Architecture
In this use case, you will implement a simple architecture that will demonstrate the ability to deliver services using the VPC components. We will define a VPC CIDR, subnetting it into 2 tiers; the first will be the frontend that will deliver the content through load balancing, and second subnet, will be the backend that will have access through an internal load balancing accessible only for the frontend tier. To demonstrate the balancing ability, it will be necessary to implement 3 backend and 2 frontend instances. The content will be delivered using load balancer round-robin algorithms, with a combination for each user request of a set of different variables, that is, for a request coming from an external user accessing the service. For this, a set of different frontend and backend instances will be used considering the algorithm in each request.

Creating a VPC
1. Open the CloudStack control panel and login with the following credentials:
2. Username: *user-1*
   
   Password: *password*
   
   Domain: *poc*

3. In the left of the navigation pane, click *Network > VPC* and then click *Add VPC*.

4. Fill in the form as follows and then click *OK*.

   - **Name:** *My VPC*
   - **Description:** *My VPC*
   - **Zone:** *poc-zone*
   - **CIDR:** *192.168.0.0/24*
   - **Network Domain:** Blank
   - **VPC Offering:** *Default VPC Offering*
   - **Start:** *yes*
Creating ACL Lists

**Note**
In CloudStack terminology, Network ACL is a group of Network ACL items. Network ACL items are nothing but numbered rules that are evaluated in order, starting with the lowest numbered rule. These rules determine whether traffic is allowed in or out of any tier associated with the network ACL. You need to add the Network ACL items to the Network ACL, then associate the Network ACL with a tier. Network ACL is associated with a VPC and can be assigned to multiple VPC tiers within a VPC. A Tier is associated with a Network ACL at all the times. Each tier can be associated with only one ACL.

**Note**
The default Network ACL is used when no ACL is associated. Default behaviour is all the incoming traffic is blocked and outgoing traffic is allowed from the tiers. Default network ACL cannot be removed or modified. You will add rules for both ACLs lists after all VPC components are created.

5. In the VPC list, click *My VPC* and then click *Network ACL Lists*. A list of default ACLs will be listed.
6. Now, you will add two new ACLs, one for Frontend and another for Backend. Click *Add Network ACL List*, fill in the form as follows and then click *OK*:

**ACL List Name:** ACL-FE  
**Description:** Frontend ACL List

7. Now, click again the *Add Network ACL List*, fill in the form as follows and then click *OK*: 

---

64
8. Click **ACL-FE > ACL List Rules** to add an ACL to deny all incoming traffic from backend tier by clicking **Add ACL** and fill in the form as follows and then click **Ok**.

- **#Rule:** 1
- **CIDR List:** 192.168.0.128/25
- **Action:** Deny
- **Protocol:** All
- **Traffic Type:** Ingress
- **Description:** Deny all ingress traffic from backend tier.
Virtual Private Cloud

9. Go back one page in your browser to return to ACL lists and click `ACL-BE > ACL List Rules` to add an ACL to allow the incoming http port (80 traffic from the frontend tier) by clicking `Add ACL` and fill in the form as follows and then click `Ok`.

| #Rule: | 1 |
| CIDR List: | 192.168.0.0/25 |
| Action: | Allow |
| Protocol: | TCP |
| Start Port: | 80 |
| End Port: | 80 |
| Traffic Type: | Ingress |
| Description: | Allow http ingress traffic from frontend tier. |

10. Click again in Add ACL to deny any other ingress traffic from frontend and fill in the form as follows and then click `Ok`.

| #Rule: | 1 |
| CIDR List: | 192.168.0.0/25 |
| Action: | Allow |
| Protocol: | TCP |
| Start Port: | 80 |
| End Port: | 80 |
| Traffic Type: | Ingress |
| Description: | Allow http ingress traffic from frontend tier. |
Virtual Private Cloud

#Rule: 2
CIDR List: 192.168.0.0/25
Action: Deny
Protocol: All
Traffic Type: Ingress
Description: Deny all ingress traffic from frontend tier.

Creating Tiers
11. In the left navigation pane, click Network > VPC and then click My VPC.

1. Click Network
2. Click VPC
3. Click My VPC

12. Click Networks tab and then click Add New Tier.
13. Fill in the form as follows then click OK.

Name: Frontend Tier  
Network Offering: Offering for Isolated Vpc networks with Source Nat service enabled  
Gateway: 192.168.0.1  
Netmask: 255.255.255.128  
External Id: Blank  
ACL: ACL-FE

14. Click again over Add New Tier, fill in the form as follows and then click OK.

Name: Backend Tier  
Network Offering: Offering for Isolated Vpc networks with Internal LB support  
Gateway: 192.168.0.129  
Netmask: 255.255.255.128
Virtual Private Cloud

External Id: Blank
ACL: **ACL-BE**

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Backend Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Offering</td>
<td>Offering for isolated VPC networks with internal L3 support</td>
</tr>
<tr>
<td>Gateway</td>
<td>192.168.0.129</td>
</tr>
<tr>
<td>Network Mask</td>
<td>255.255.255.128</td>
</tr>
<tr>
<td>External Id</td>
<td>ID of the network in an external system</td>
</tr>
<tr>
<td>ACL</td>
<td>ACL-BE (Backend ACL List )</td>
</tr>
</tbody>
</table>
```

**Note**
You created both tiers and each uses a guest network VLAN where both VLAN are connected to the Virtual Router that implements static routing between tiers.

Creating Backend Instances

15. In the left navigation pane, click *Compute* to expand the menu, and click *Instances*.

16. Click *Add Instance*.
1. **Note**

To create each of the three backend instances, change only the Name field as follows (backend-instance-01, backend-instance-02, backend-instance-03).

17. Fill in the forms for each instance (backend-instance-01, backend-instance-02, backend-instance-03) as follows and then click **Launch Instance**:

- **Zone:** poc-zone
- **Template/ISO**
  - Community: CentOS 7.0 (64bit)
- **Compute Offering:** Small Instance
- **Networks**
  - Backend Tier: Yes
  - Frontend Tier: No
- **Compute Offering:** Small Instance
- **Advanced Mode:** Yes
  - **Userdata:**

```bash
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
hostname > /var/www/html/index.html
```

- **Name:** backend-instance-0[1,2,3]
- **Group:** Blank
- **Keyboard Language:** Blank
- **Start Instance:** Yes

### Creating and configuring the Internal Load Balancer

18. In the left navigation pane, click **Network** > **VPC** and click **My VPC**.

19. Click **Networks** tab, expand **Internal LB** menu and then, click **Add Internal LB**.
20. Fill in the form as follows and click **OK**.

Name: **Backend LB service**  
Description: **Internal LB service for Backend service**  
Source IP Address: **Blank**  
Source Port: **80**  
Instance Port: **80**  
Algorithm: **Round-robin**

21. Copy the **Source IP Address** in the Internal LB list and paste it in a text editor.
22. Click **Backend LB Service**.

23. Click **Assigned Instances** and then click **Assign Instance**.

24. Select all backend instances and then click **Ok**.
Creating Frontend Instances
25. In the left navigation pane, click Compute > Instances and then, click Add Instance.

![Image of CloudStack interface showing Compute and Instances options]

---

Note
To create each of two frontend instances, change `backend_endpoint` variable in the beginning of the UserData field for the Source IP Address from your text editor. Also change the Name field as follows (frontend-instance-01, frontend-instance-02).

26. Fill in the forms for each instance (frontend-instance-01, frontend-instance-02) as follows and then click Launch Instance:

<table>
<thead>
<tr>
<th>Zone:</th>
<th>poc-zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template/ISO</td>
<td>CentOS 7.0 (64bit)</td>
</tr>
<tr>
<td>Compute Offering:</td>
<td>Small Instance</td>
</tr>
<tr>
<td>Networks:</td>
<td></td>
</tr>
<tr>
<td>- Backend Tier:</td>
<td>No</td>
</tr>
<tr>
<td>- Frontend Tier:</td>
<td>Yes</td>
</tr>
<tr>
<td>Advanced Mode:</td>
<td>Yes</td>
</tr>
<tr>
<td>UserData:</td>
<td></td>
</tr>
</tbody>
</table>

```bash
# ! /bin/bash
backend_endpoint="Internal LB Source IP Address"
yum -y install httpd
cat << EOF > /etc/httpd/conf.d/cgi-enabled.conf
<Directory "/var/www/html/frontend">
  Options +ExecCGI
  AddHandler cgi-script .py
</Directory>
EOF
mkdir /var/www/html/frontend
cat << EOF > /var/www/html/frontend/index.py
#!/usr/bin/env python
import requests
import socket
h = socket.gethostname()
EOF
```
Virtual Private Cloud

```
r = requests.get("http://${backend_endpoint}/")
print "Content-type: text/html\n\n"
print "<html>\n<body>"
print "<meta http-equiv="refresh" content="10="/">
print "<div style="width: 100%; font-size: 40px; font-weight:
bold; text-align: center;">"
print('%s - %s' % (h, r.text))
print "</div>\n</body>\n</html>"
EOF
chmod 705 /var/www/html/frontend/index.py
systemctl enable httpd
systemctl start httpd
```

Name: frontend-instance-0x
Group: None
Keyboard Language: None
Start Instance: Yes

Creating and Configuring the External Load Balancer

27. In the left navigation pane, click Network > VPC and then click My VPC.

28. Click Public IP Address > Acquire New IP, and then click OK.
29. Click the added IP (not the source-nat).

30. Copy the IP Address showed in the Details tab and past it in a text editor. It will be used later.

31. Click Load Balancing, fill in the form as follows and then click Add.
32. Select the tier *Frontend Tier* and select all instances in the list as follows and then click *OK*. 
Accessing the Web Service Hosted Within the VPC

33. Open a new tab in your browser and paste the IP Address copied in the step #30 followed by /frontend/index.py
   Example:  http://IP_Address/frontend/index.py

**Note**
The page will be reloaded in each 10 seconds and you can see the frontend and backend changing in each request.

**Note**
One of the objectives of the cloud-init installed in the template is to allow a set of scripts to be executed when first boot is performed during the instance creation. Below explanation of what runs on Userdata.

**Userdata in the Frontend instances.**

1. In the first script line, a `/bin/bash` was called to interpret the script:

   ```bash
   #!/bin/bash
   ```

2. The variable `backend_endpoint` was settled with the value of the source ip address of the internal LB:

   ```bash
   backend_endpoint="192.168.0.x"
   ```

3. The web server `httpd` was installed:

   ```bash
   yum -y install httpd
   ```

4. The file `/etc/httpd/conf.d/cgi-enabled.conf` was wrote to enable the directory `/var/www/html/frontend` to execute cgi python script:

   ```bash
   cat << EOF > /etc/httpd/conf.d/cgi-enabled.conf
   <Directory "/var/www/html/frontend">
   Options +ExecCGI
   AddHandler cgi-script .py
   </Directory>
   EOF
   ```

5. The directory `/var/www/html/frontend` was created where posteriorly the cgi python script will be written:

   ```bash
   mkdir /var/www/html/frontend
   ```
6. The python script is written in the directory:

```
cat << EOF > /var/www/html/frontend/index.py
```

7. This is the content of the cgi python script commented:

```python
#!/usr/bin/env python
import requests
import socket

# Request call to the Internal LB
r = requests.get("http://${backend_endpoint}/")

# Get the instance hostname
h = socket.gethostname()

# Print the HTML code
print "Content-type: text/html\n\n"
print "<html>
<body>

<meta http-equiv="refresh" content="10"/>

<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center;">"

# Here, print the values of hostname and Internal LB request
print('%s - %s' % (h, r.text))
print "</div>\n</body>\n</html>"

EOF
```

8. End of file of the cgi python script:

```
EOF
```

9. Change the cgi python script permission to be executable:

```
chmod 705 /var/www/html/frontend/index.py
```

10. Enable the httpd service:

```
systemctl enable httpd
```

11. Start the httpd service:

```
systemctl start httpd
```
**Userdata in the Backend instances.**

1. *In the first script line, a /bin/bash was called to interpret the script:*
   ```bash
   #!/bin/bash
   ```

2. *The web server httpd was installed:*
   ```bash
   yum -y install httpd
   ```

3. *Enable the httpd service:*
   ```bash
   systemctl enable httpd
   ```

4. *Start the httpd service:*
   ```bash
   systemctl start httpd
   ```

5. *Write the instance hostname to the file /var/www/html/index.html*
   ```bash
   systemctl start httpd
   ```
Summary
Apache CloudStack is the leading open-source cloud orchestration platform used by many of the world’s largest public and private clouds. It is a multi-hypervisor, multi-tenant, high-availability Infrastructure as a Service cloud management platform.

Apache CloudStack is software that provides a cloud orchestration layer, giving automation of the creation, provisioning and configuration of IaaS components (such as virtual servers). It turns existing virtual infrastructure into a cloud-based Infrastructure as a Service (IaaS) platform. Because CloudStack leverages existing infrastructure, the cost and time for the organization to build a multi-tenant IaaS platform are greatly reduced.

Among the most significant advantages of the virtualization management platform is the simplicity and ease of use it brings, even for large-scale environments. With CloudStack, you can orchestrate hosted public, on-premise clouds and hybrid environments without the need of engaging a huge operations team to support them in the long term.

As more and more companies build on-premises clouds or enter the service provider market with public clouds, the more they will need the right set of tools to successfully build, manage and scale their Infrastructure as a Service (IaaS) platform. However – choosing the right technology stack can be a difficult decision. There are several aspects that should be considered, such as planning for future growth and demand, team size, budget, project timeframe, previous experience, available hardware and the underlying infrastructure already in place.

After completion of a successful CloudStack PoC by following all of the described steps in this guide, you will feel confident to implement CloudStack in your production environment and benefit from all its advantages!
About Apache CloudStack

Apache CloudStack is the leading open-source cloud orchestration platform, in use by many of the world’s largest public and private clouds. It is a multi-hypervisor, multi-tenant, high-availability Infrastructure as a Service cloud management platform. CloudStack is software that provides a cloud orchestration layer, giving automation of the creation, provisioning and configuration of IaaS components.

CloudStack turns an existing virtual infrastructure into a cloud-based infrastructure as a Service (IaaS) platform. The fact CloudStack leverages existing infrastructure means that the cost and time for an organisation to build a multi-tenant IaaS platform is greatly reduced.

cloudstack.apache.org/

Need help with Apache CloudStack?

Get In Touch