

Step-by-Step Guide



Apache CloudStack PoC Guide

Installation and Use Cases



apachecloudstack

open source cloud computing

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

1.34	CREATING A VPC.....	61
1.35	CREATING ACL LISTS.....	63
1.36	CREATING TIERS.....	67
1.37	CREATING BACKEND INSTANCES.....	69
1.38	CREATING AND CONFIGURING THE INTERNAL LOAD BALANCER.....	70
1.39	CREATING FRONTEND INSTANCES.....	73
1.40	CREATING AND CONFIGURING THE EXTERNAL LOAD BALANCER.....	74
1.41	ACCESSING THE WEB SERVICE HOSTED WITHIN THE VPC.....	77
	SUMMARY	80

About this Guide

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.

1 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.

Note

Highlights an observation that provides additional information.

Warning

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

Primary and Secondary storage created as NFS exports on the CloudStack Management server
500GB of RAID based storage on CloudStack Management server

Hosts

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

Management Server

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

Networking

Item	Quantity/Description
No. Switches	1
VLAN	802.1q support required for advanced network zones.
No. ports	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.
Throughput	1 Gb/s

Soft Requirements

IP Address / VLAN Space

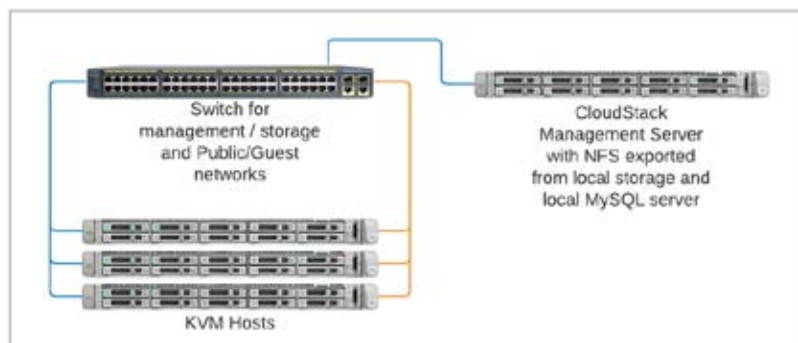
Item	Quantity/Description	Network IP Address	VLAN ID
Public Network	0 public addresses – /24 network IP range - RFC 1918 addresses routable within POC environment	10.0.48.0/24	48
Guest Network	20x VLANs dedicated to CloudStack use.		101-120
Management and Storage Network	40x RFC 1918 addresses	10.0.32.0/24	49

Hostname and IP addresses

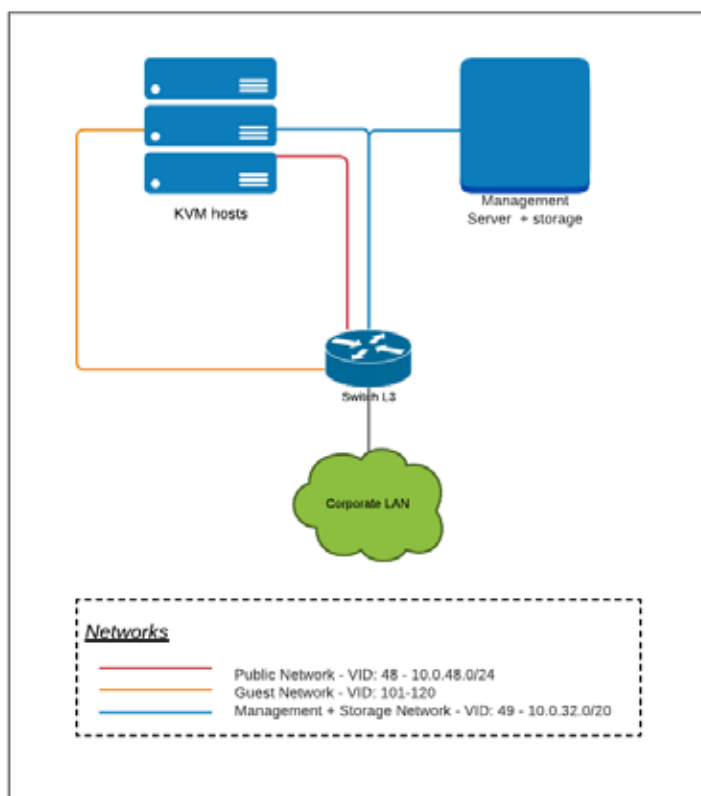
Host	hostname	IP Address	Netmask
CloudStack Management Server	mgmt..local	10.0.32.100	255.255.240.0
KVM Host 01	host01.local	10.0.33.1	255.255.240.0
KVM Host 02	host02.local	10.0.33.2	255.255.240.0
KVM Host 03	host03.local	10.0.33.3	255.255.240.0
Reserved System VM	-	10.0.34.1 – 10.0.34.40	255.255.240.0

Infrastructure Design

Physical Layout



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.

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

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
```

6. Edit the `/etc/nfs.conf` file.

```
vi /etc/nfs.conf
```

Ensure the parameters are like the following lines.

```
[general]
[exportfs]
[gssd]
use-gss-proxy=1
[lockd]
port=32803
udp-port=32769
[mountd]
port=892
[nfsdclld]
[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.

```
RPCRQUOTAD_OPTS="-p 875"
```

8. Enable and start the services.

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

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:

```

program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100011 1 udp 875 rquotad
100011 2 udp 875 rquotad
100011 1 tcp 875 rquotad
100011 2 tcp 875 rquotad

```

Installation of Database Server

① 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:

```

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.

```

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.

```

systemctl start mysqld

```

5. Run the following command to setup a CloudStack database.

```

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

```

It will set the database with the following informations.

Database	cloud
User	cloud
Password	password

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:

```
/usr/share/cloudstack-common/scripts/storage/secondary/cloud-install-sys-tmpl -m /export/secondary -u  
http://download.cloudstack.org/systemvm/4.16/systemvmtemplate-4.16.0-kvm.qcow2.bz2 -h kvm -F
```

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
```

Note

This should return a fully qualified KVM hostname.

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.

```
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
```


ⓘ 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 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:

```
LIBVIRT_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.

Note

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.

Warning

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

Note

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
```

Note

Now we have the interfaces configured and can add the bridges on top of them.

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
```

Note

With this configuration you should be able to restart the network, although a reboot is recommended to see if everything works properly.

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`

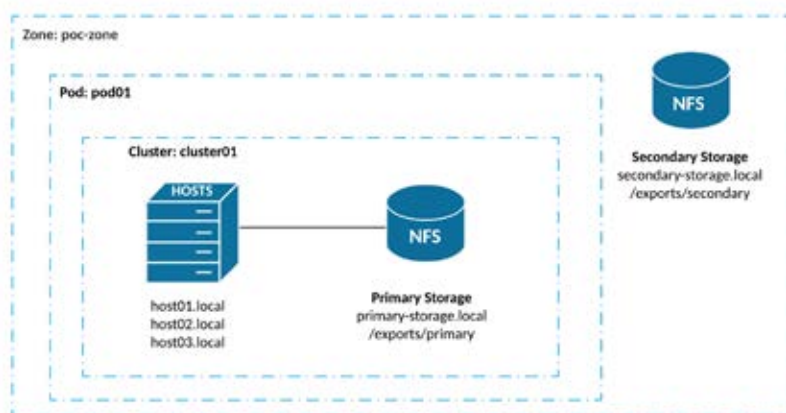
The command should return the following output:

bridge name	bridge id	STP enabled	interfaces
cloudbr0	8000.1e0032000265	yes	eth0
cloudbr1	8000.020045c625cb	yes	eth1
virbr0.	8000.525400774409	yes	virbr0-nic

Adding Zone

① Note

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

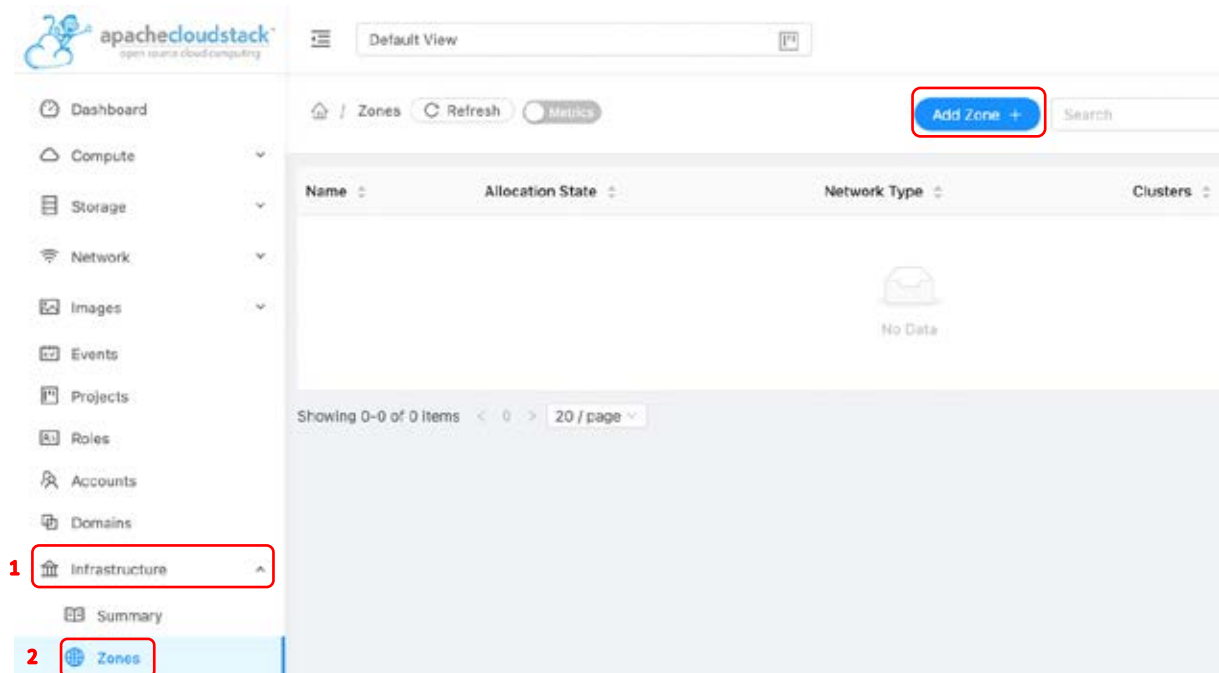


1. To access the CloudStack UI, open this URL `http:// 10.0.32.10:8080/client` in your web browser.
2. Login with the following credentials and click [Login](#).

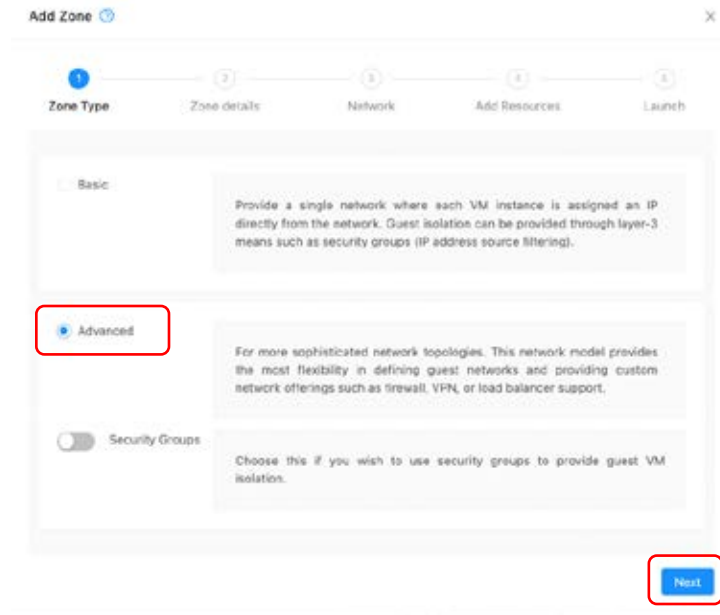
Username: *admin*
 Password: *password*
 Domain: Blank



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*.



Note
For more info about Advanced Zone, you can reach here
http://docs.cloudstack.apache.org/en/latest/adminguide/networking/advanced_zone_config.html.

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*

Add Zone ⓘ ✕

1 2 3 4 5
 Zone Type **Zone details** Network Add Resources Launch

A zone is the largest organizational unit in CloudStack, and it typically corresponds to a single datacenter. Zones provide physical isolation and redundancy. A zone consists of one or more pods (each of which contains hosts and primary storage servers) and a secondary storage server which is shared by all pods in the zone.

Internal DNS 2:

* Hypervisor: ✓

Network Domain:

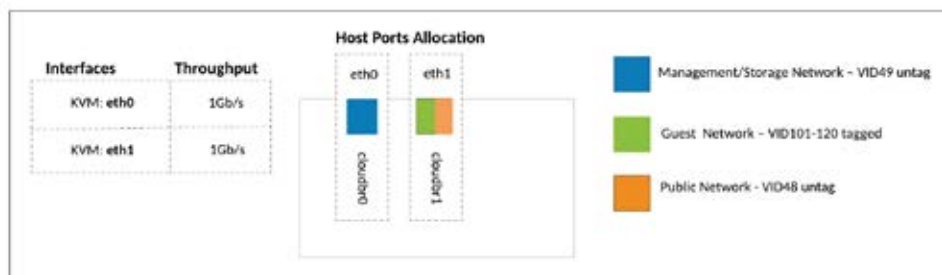
Guest CIDR: ✓

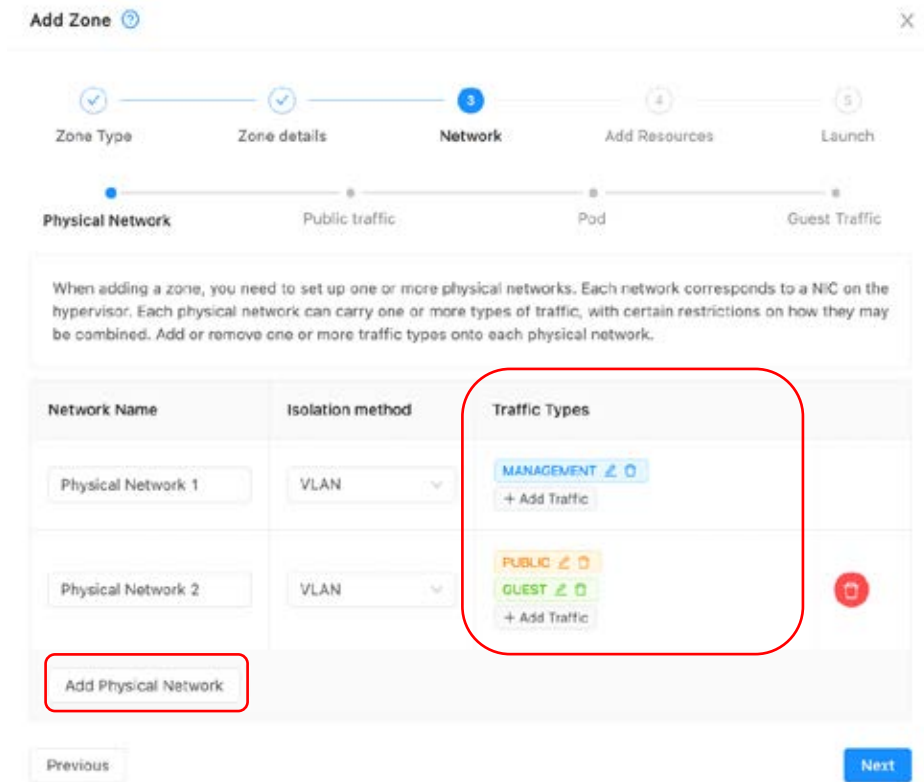
Dedicated:

Enable local storage for User VMs:

Enable local storage for System VMs:

- 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:





- Click *edit* in each *traffic type* item and set the *traffic label* as follows and, after all traffic label was defined, click *Next*.

Name	Traffic Type	Traffic label
Physical Network 1	- Management	cloudbr0
Physical Network 2	- Public	cloudbr1
	- Guest	

Network Name	Isolation method	Traffic Types
Physical Network 1	VLAN	MANAGEMENT + Add Traffic
Physical Network 2	VLAN	PUBLIC GUEST

Edit traffic type ✕

Add Phys

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

Previous * Traffic label: Next

Cancel OK

- 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*.

Add Zone
X

1 ✓

Zone Type

2 ✓

Zone details

3

Network

4

Add Resources

5

Launch

Physical Network **Public traffic** Pod Guest Traffic

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 their public network.

Provide at least one range of IP addresses for internet traffic.

Gateway	Netmask	VLAN/VNI	Start IP	End IP	
No Data					
10.0.48.1	255.255.255.0	48	10.0.48.2	10.0.48.254	Add

Previous
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*
- Rerved system gateway: *10.0.32.1*
- Rerved system netmask: *255.255.240.0*
- Start Rerved system IP: *10.0.34.1*
- End Rerved system IP: *10.0.34.40*

Add Zone ⓘ ✕

Progress: 1. Zone Type (✓) 2. Zone details (✓) 3. Network (3) 4. Add Resources (4) 5. Launch (5)

Physical Network | Public traffic | **Pod** | Guest Traffic

Each zone must contain in 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:	<input type="text" value="POD1"/>	✓
* Reserved system gateway:	<input type="text" value="10.0.32.1"/>	✓
* Reserved system netmask:	<input type="text" value="255.255.240.0"/>	✓
* Start Reserved system IP:	<input type="text" value="10.0.34.1"/>	✓
End Reserved system IP:	<input type="text" value="10.0.34.40"/>	✓

11. Now, configure the *Guest Traffic* VLANs and then click *Next*.

VLAN/VNI Range: *101 - 120*

Add Zone [Close]

Zone Type [✓] — Zone details [✓] — **Network** [3] — Add Resources [4] — Launch [5]

Physical Network — Public traffic — Pod — **Guest Traffic**

Guest network traffic is communication between end-user virtual machines. Specify a range of VLAN IDs to carry guest traffic for each physical network.

VLAN/VNI Range: -

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

Cluster Name: *cluster01*

Add Zone [Close]

Zone Type [✓] — Zone details [✓] — Network [✓] — **Add Resources** [4] — Launch [5]

Cluster — IP Address — Primary Storage — Secondary Storage

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

Cluster Name:

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

Add Zone

Zone Type Zone details Network **Add Resources** Launch

Cluster IP Address Primary Storage Secondary Storage

Each cluster must contain at least one host (computer) for guest VMs to run on, and we will add the first host now. For a host to function in CloudStack, you must install hypervisor software on the host, assign an IP address to the host, and ensure the host is connected to the CloudStack management server.

Give the host's DNS or IP address, the user name (usually root) and password, and any labels you use to categorize hosts.

* Host Name: ✓

* Username: ✓

* Password: ✓

Tags:

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

Add Zone [Close]

Zone Type [✓] Zone details [✓] Network [✓] **Add Resources** [4] Launch [8]

Cluster IP Address **Primary Storage** Secondary Storage

Each cluster must contain one or more primary storage servers, and we will add the first one now. Primary storage contains the disk volumes for all the VMs running on hosts in the cluster. Use any standards-compliant protocol that is supported by the underlying hypervisor.

* Name: Primary Storage [✓]
 Scope: Cluster [✓]
 * Protocol: nfs [✓]
 * Server: primary-storage.local [✓]
 * Path: /export/primary [✓]
 Storage Tags:

Previous **Next**

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*

Add Zone [Close]

Zone Type [✓] Zone details [✓] Network [✓] **Add Resources** [4] Launch [8]

Cluster IP Address Primary Storage **Secondary Storage**

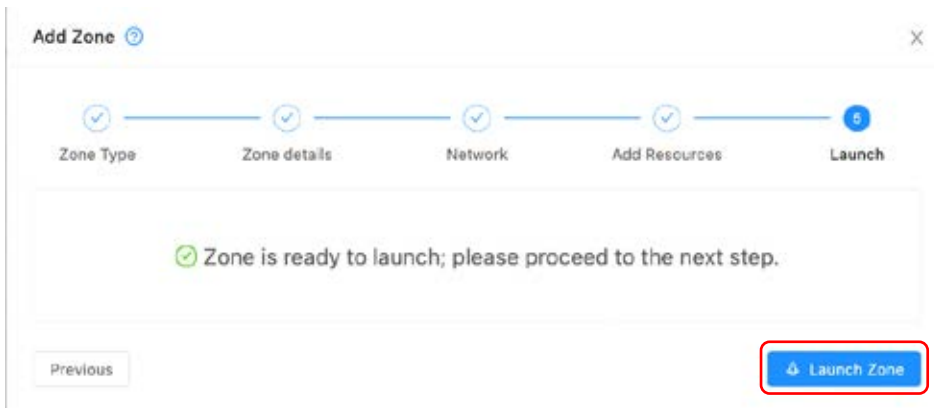
Each zone must have at least one NFS or secondary storage server, and we will add the first one now. Secondary storage stores VM templates, ISO images, and VM disk volume snapshots. This server must be available to all hosts in the zone.

Provide the IP address and exported path.

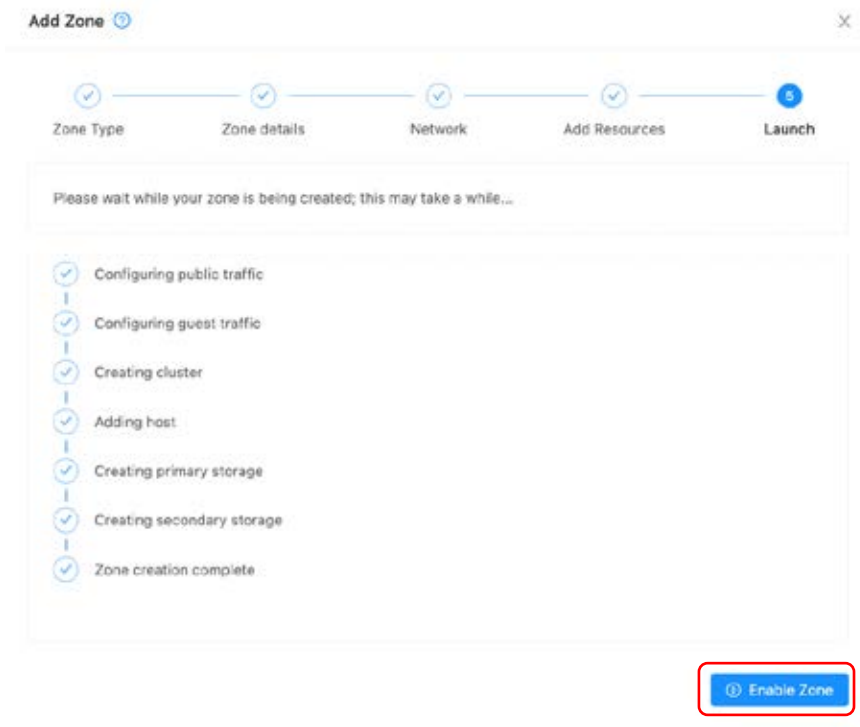
Provider: NFS [✓]
 Name: Secondary Storage [✓]
 * Server: secondary-storage.local [✓]
 * Path: /export/secondary [✓]

Previous **Next**

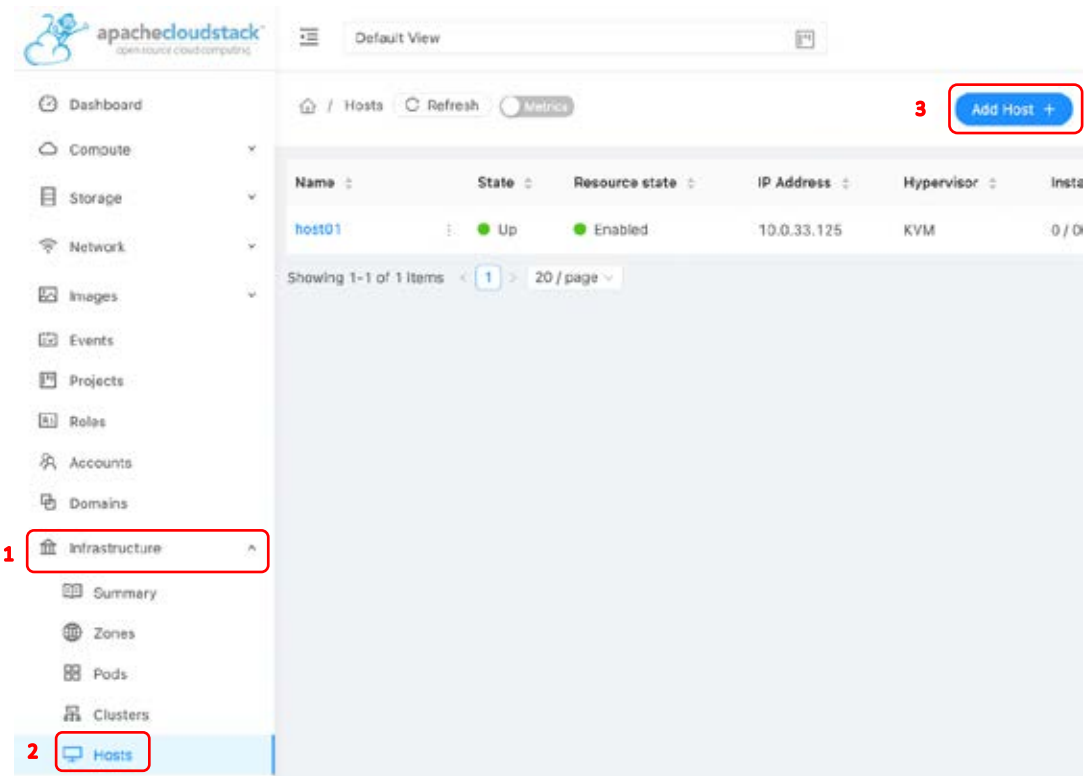
16. Now, click [Launch Zone](#) to proceed.



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

The screenshot shows a dialog box titled "Add Host" with a close button (X) in the top right corner. The dialog contains the following fields and values:

- Zone Name:** A dropdown menu with "Poc Zone" selected.
- Pod name:** A dropdown menu with "POD1" selected.
- Cluster Name:** A dropdown menu with "cluster01" selected.
- Host Name:** A text input field containing "host02.local".
- Username:** A text input field containing "root".
- Password:** A password input field with masked characters (dots).
- Host Tags:** A text input field containing "Est of tags to be added to the host".
- Dedicated:** An unchecked checkbox.

At the bottom of the dialog, there are two buttons: "Cancel" and "OK".

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:

Role	Description
Root Admin	Manage the entire platform, including physical and logical resources from all domains and accounts.
Domain Admin	Manage all logical resources under the domain and adjacent sub-domains including user accounts, sub-domains and all related virtual computing resources.
User	Manage virtual computing resources related to your own account.

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

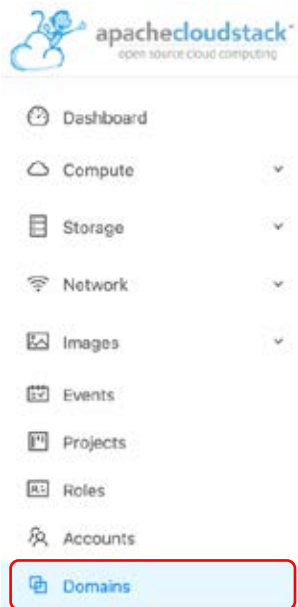


Creating a Domain

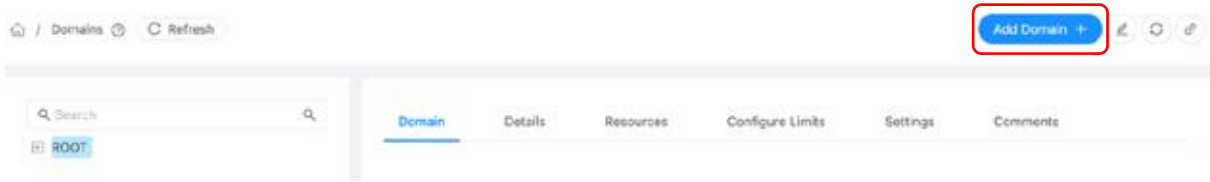
① Note

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.

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



21. Now click *Add Domain*.

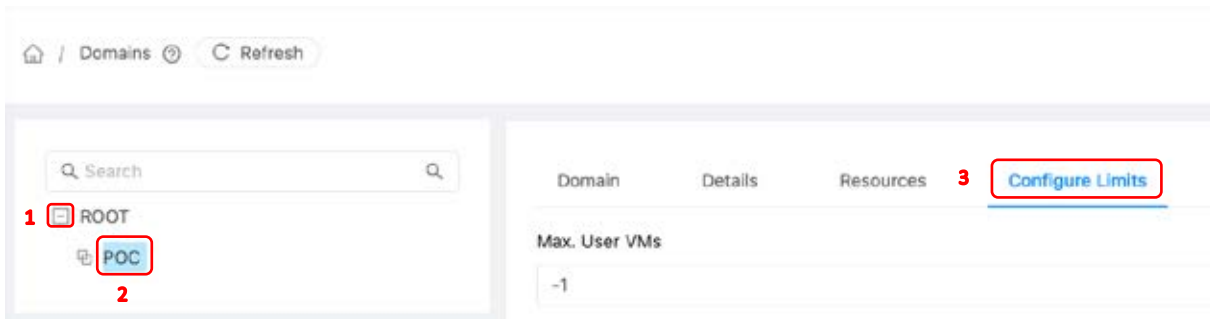


22. Fill in the form as follows and then click *OK*:

Name: *POC*
Network Domain: Blank
Domain: Blank

Limiting resources at Domain level

23. Expand the *domain tree* and click *POC* and then click *Configure Limits*.



Note
This screen shows the computing resources limits in the domain level. By default, the resources are configured with *-1*, that means *unlimited resource*.

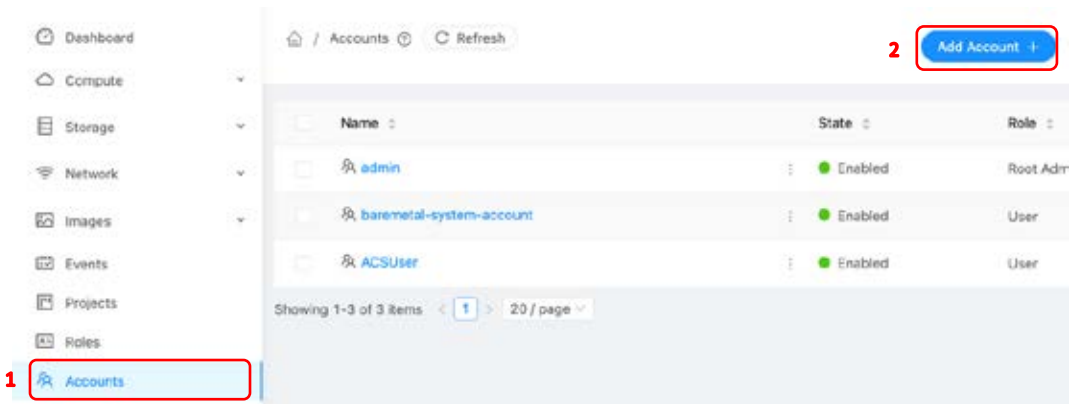
24. Set up the domain resources as follows and then click *Submit*:

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

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

Add Account [Close]

* Role [Down Arrow]
Domain Admin (DomainAdmin)

* Username [Down Arrow]
admin

* Password [Down Arrow] [Eye Icon] * Confirm Password [Down Arrow] [Eye Icon]
.....

* Email [Down Arrow]
admin@poc.local

* First Name [Down Arrow] * Last Name [Down Arrow]
Administrator PoC Domain

* Domain [Down Arrow]
ROOT/POC

Account [Down Arrow]
admin

Timezone [Down Arrow]
GMT [Greenwich Mean Time]

Network Domain [Down Arrow]
Network domain for the account's networks

Cancel **OK**

27. Click the *admin* where the corresponding role is *Domain Admin*.

/ Accounts [Refresh] [Add Account +] Search

Name	State	Role
admin	Enabled	Root Admin
baremetal-system-account	Enabled	User
ACSUser	Enabled	User
admin	Enabled	Domain Admin

28. The domain admin account will be used only for administrative purposes, 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

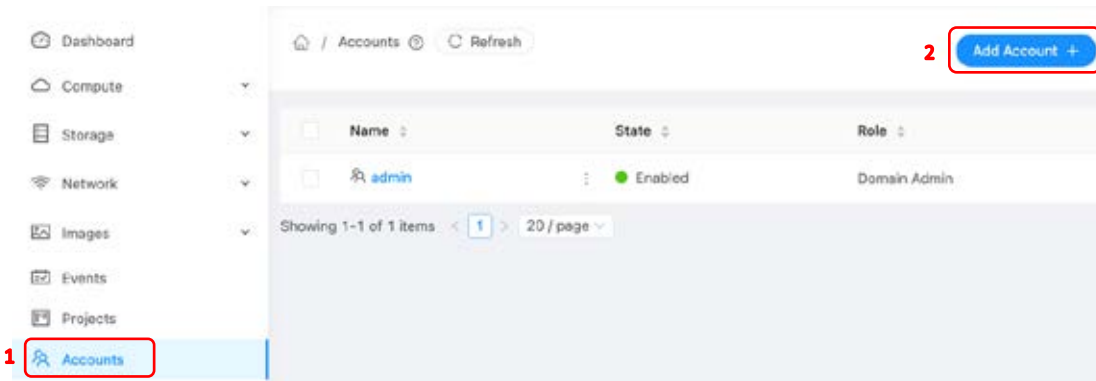
Max. Templates: 0
Max. Networks: 0
Max. VPCs: 0
Max. CPU Cores: 0
Max. Memory (MiB): 0
Max. Primary Storage (GiB): 0
Max. Secondary Storage (GiB): 0

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*

Add Account ⓘ

* Role ⓘ
User (User) ▾

* Username ⓘ
user-1

* Password ⓘ * Confirm Password ⓘ
..... ⓘ

* Email ⓘ
user-1@poc.local

* First Name ⓘ * Last Name ⓘ
User-1 PoC Domain

* Domain ⓘ
ROOT/POC ▾

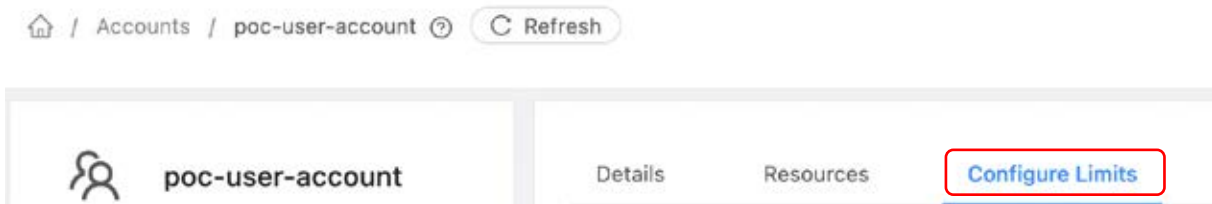
Account ⓘ
poc-user-account

Timezone ⓘ
GMT [Greenwich Mean Time] ▾

Network Domain ⓘ
Network domain for the account's networks.

Cancel **OK**

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
- Expurge instance

Accessing the Control Panel

1. Login with the following credentials:

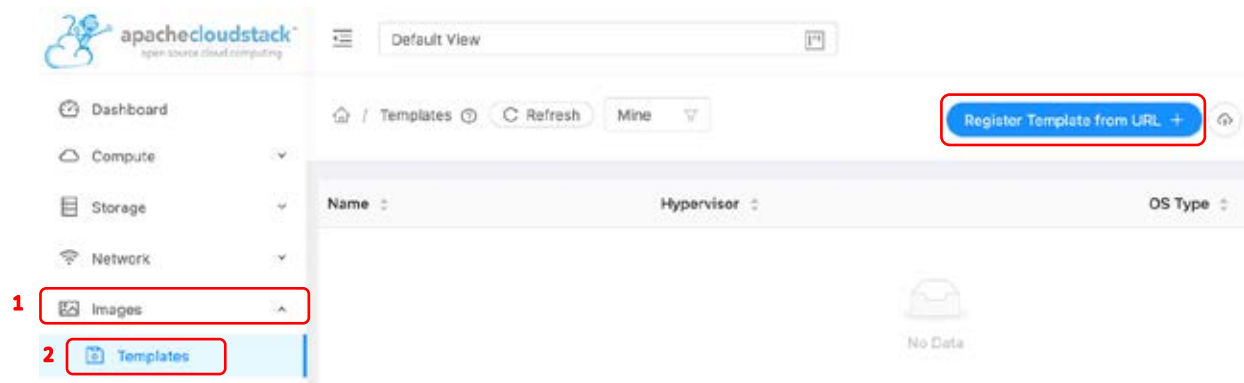
Username: *user-1*
Password: *password*
Domain: *poc*

Registering a New Template

① 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

Register Template from URL

* URL
http://dl.openvm.eu/cloudstack/centos/x86_64/centos-7-kvm.qcow2.bz2

* Name
CentOS 7.0

* Description
CentOS 7.0(64-bit)

* Zone
poc-zone

* Hypervisor
KVM

* Format
QCOW2

Direct Download

* Root disk controller
osdefault

* OS Type
CentOS 7.2

Extractable
 Dynamically Scalable
 Featured
 Routing

Password Enabled
 HVM
 Public

Cancel OK

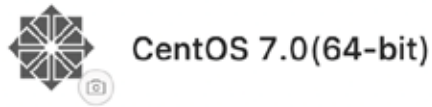
4. From the previews registered template, select [CentOS 7](#) to check the download status:

CentOS 7.0(64-bit)

KVM Dynamically Scalable

Status
 Installing Template

5. Refresh this page clicking [refresh](#) every 30 seconds until the download is complete.



KVM Dynamically Scalable

Status
● Download Complete

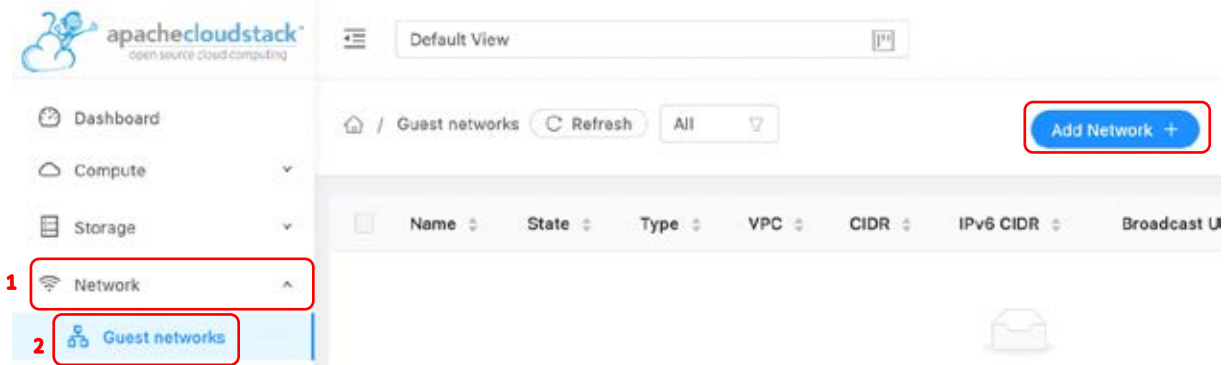
Create a guest network and manage egress rules

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.

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

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*.



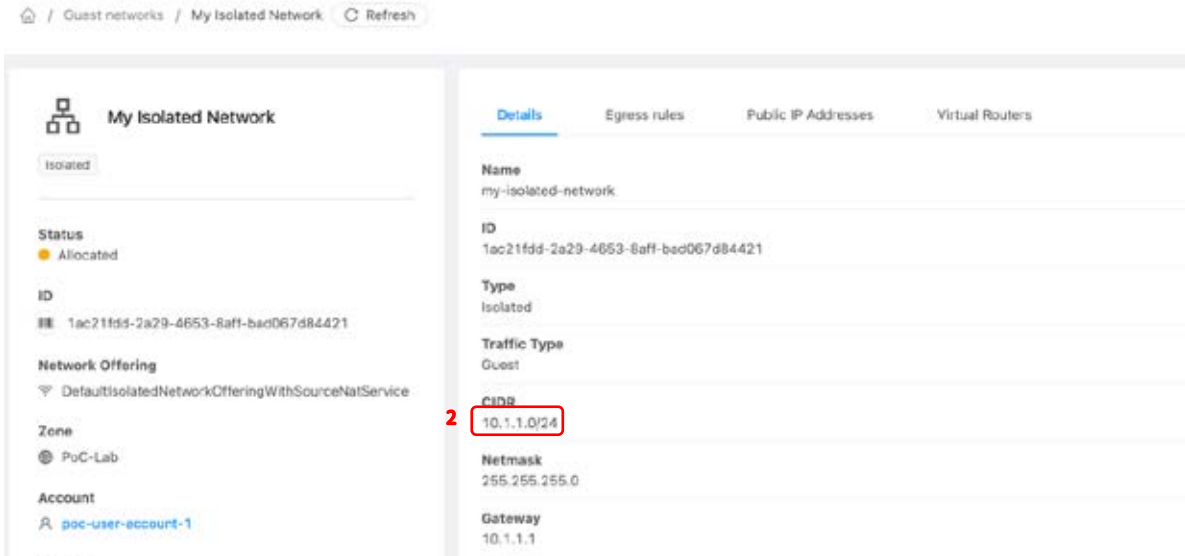
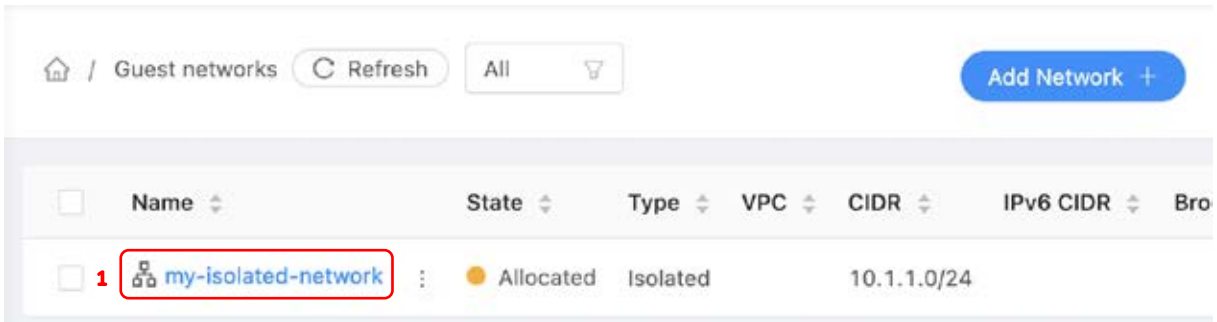
7. Select *Isolated*, and fill in the form as follows and then click *OK*:

Name: **my-isolated-network**
Description: **My Isolated Network**
Zone: **poc-zone**
Network Offering: **Offering for Isolated networks with Source Nat service enabled**
External Id: Blank
Gateway: Blank
Netmask: Blank
Network Domain: Blank

1

2

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

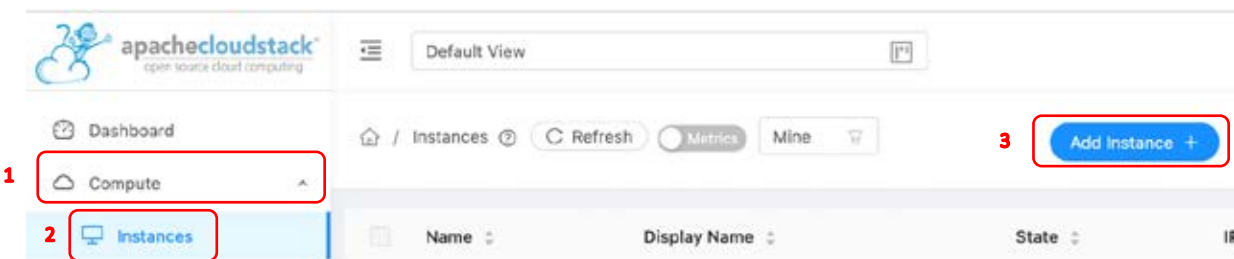
① Note

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.

① 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 SSHkeys during the instance boot using the CloudStack userdata and metadata.

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



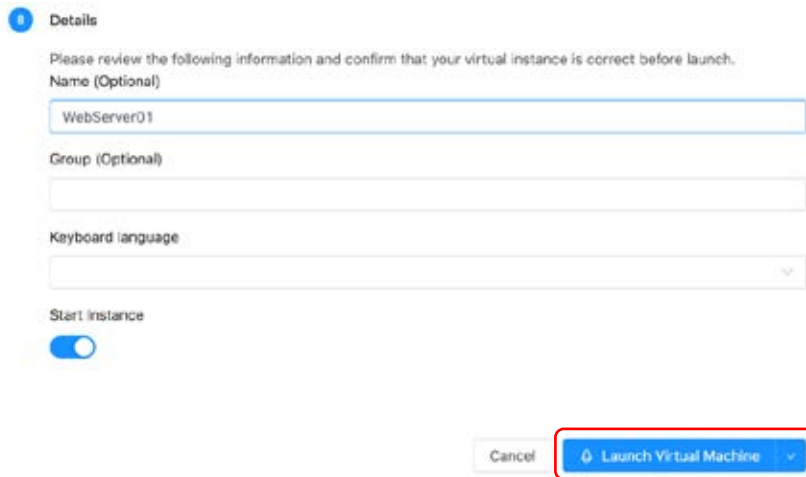
12. Fill in the form as follows and then click *Launch Instance*:

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

- Userdata:

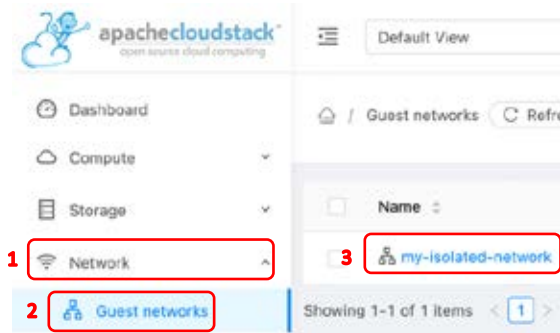
```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h2>Hello from your new
WebServer!</h2></html>' > /var/www/html/index.html
WebServer01
```

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

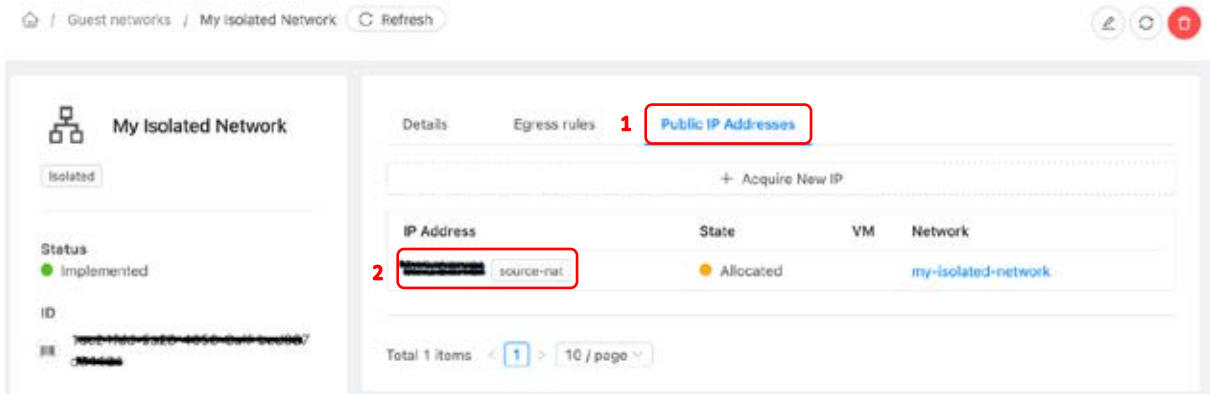


Enabling HTTP port redirect to expose the Web Server to public access

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

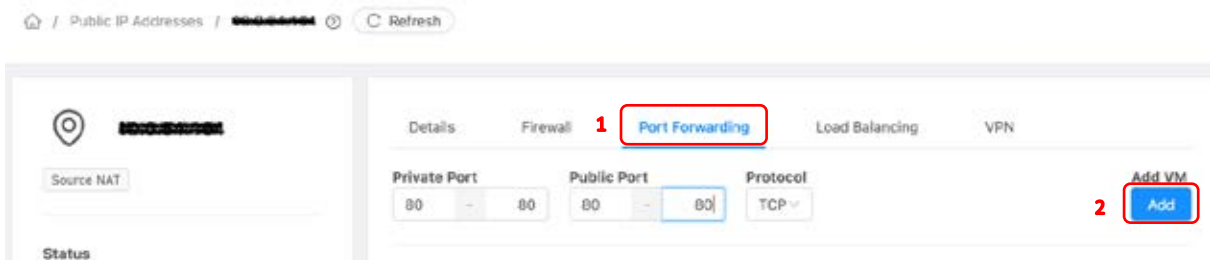


14. Select *Public IP Address* and then click *source nat IP*.

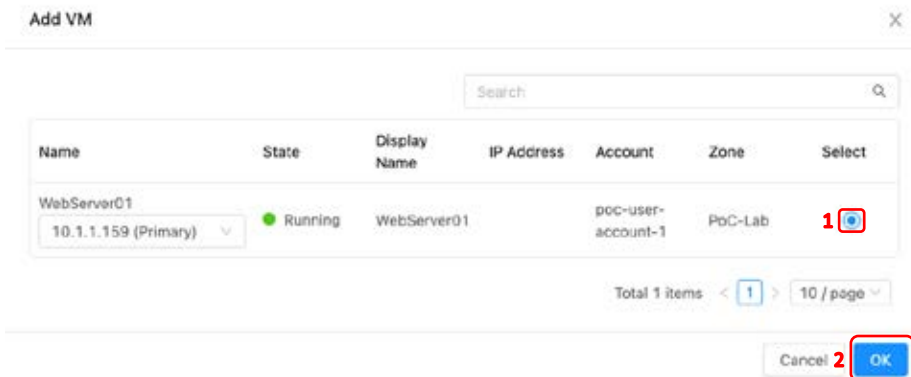


15. Select *Port Forwarding* and add a rule to redirect the HTTP port as follows and click *Add*.

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*.

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

1 Firewall Port Forwarding Load Balancing VPN

Source CIDR: 0.0.0.0/0 Protocol: TCP Start Port: 80 End Port: 80 2 Add

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

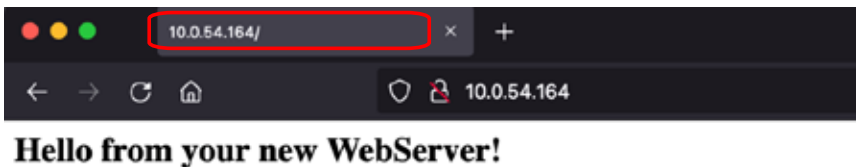
Source CIDR	Protocol	Start Port/ICMP Type	End Port/ICMP Code	State	Action
0.0.0.0/0	TCP	80	80	Active	 

19. In *Details* copy the *IP address*.

1 Details Firewall Port Forwarding Load Balancing VPN

2 IP Address 10.0.54.164

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



Note

The Web Server is now running.

Scaling Up/Down Instances Resources

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.

Note

Compute offering can only be created by domain or root administrators. User accounts can only consume them for use in their respective resources.

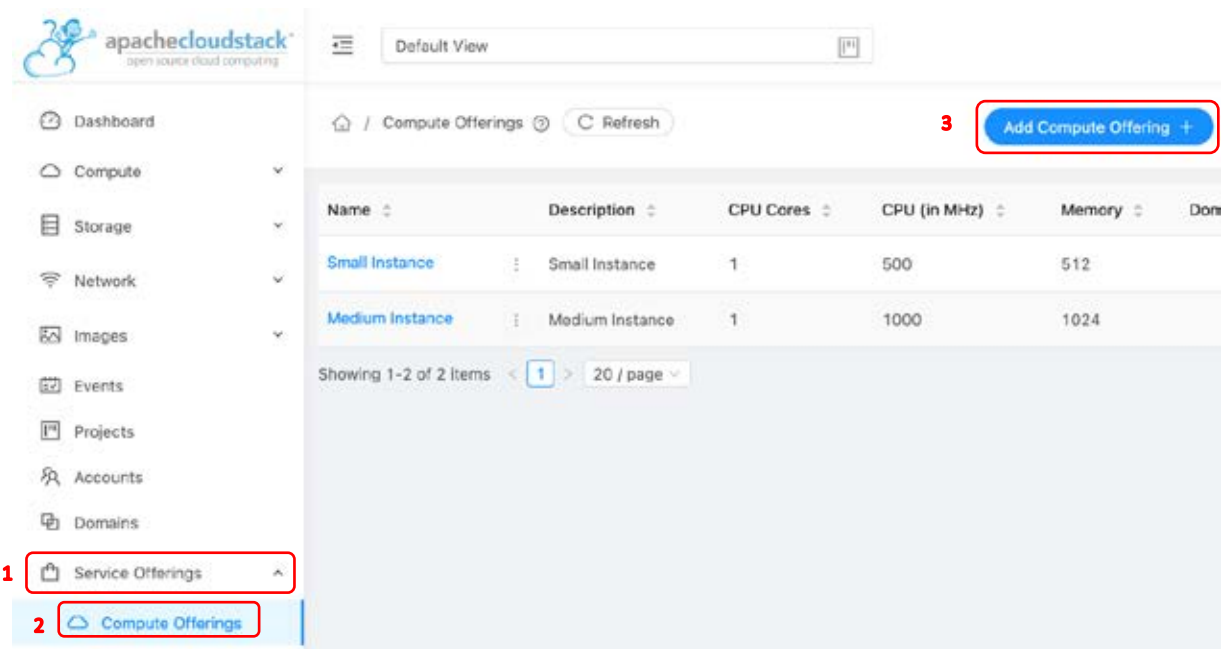
21. Logout and then login with the following credentials:

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*

GPU: *None*
Domain: *ROOT/POC*
Zone: *All Zones*

Offer HA

CPU Cap

Volatile

GPU

None NVIDIA GRID K1 NVIDIA GRID K2

Domain

ROOT/POC

Zone

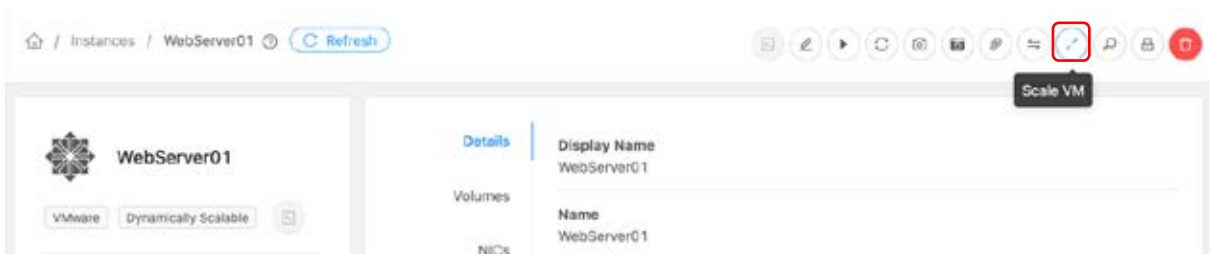
All Zones

Cancel OK

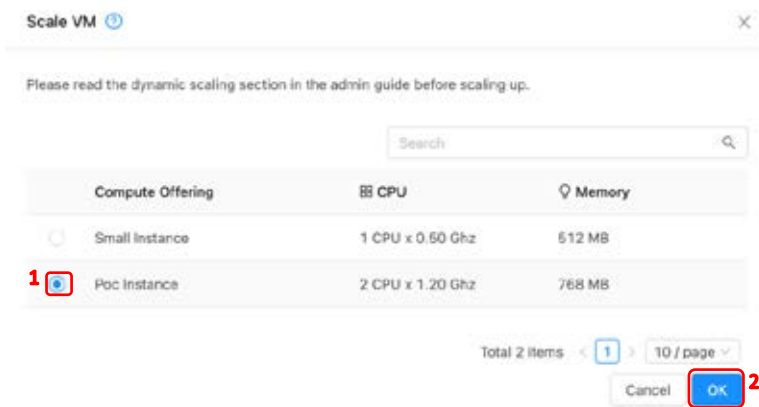
24. Logout, and then login with the following credentials:

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*.

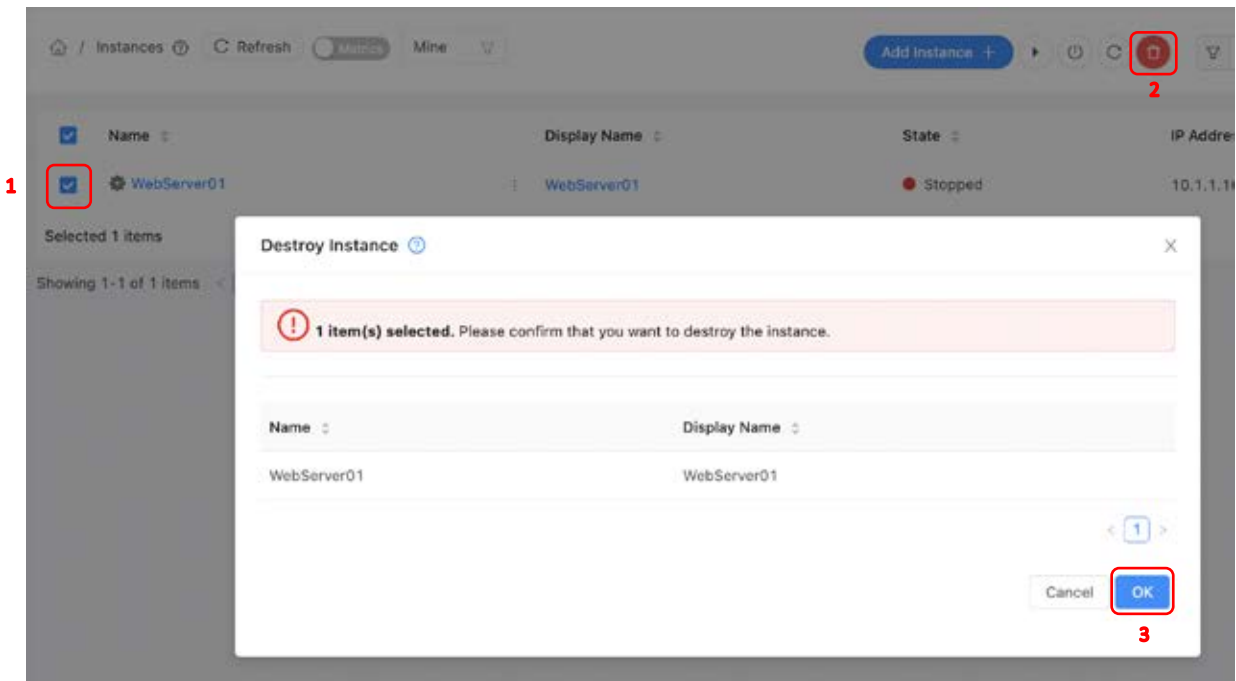


27. The instance will be scaled with the values inherited from the Compute Offering *Poc Instance*.

Removing and Recovering Instances

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*.



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.

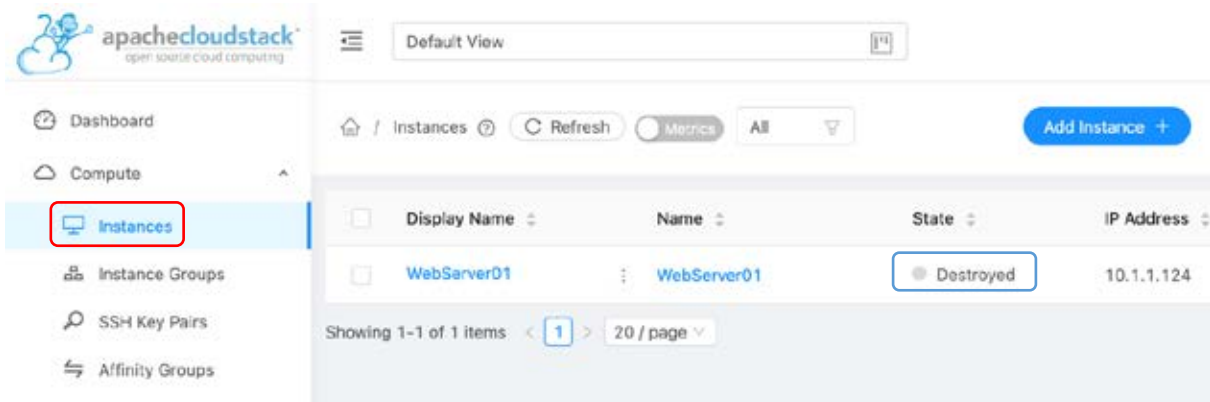
29. Logout, and then login with the following credentials:

30. Username: *admin*

Password: *password*

Domain: *poc*

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

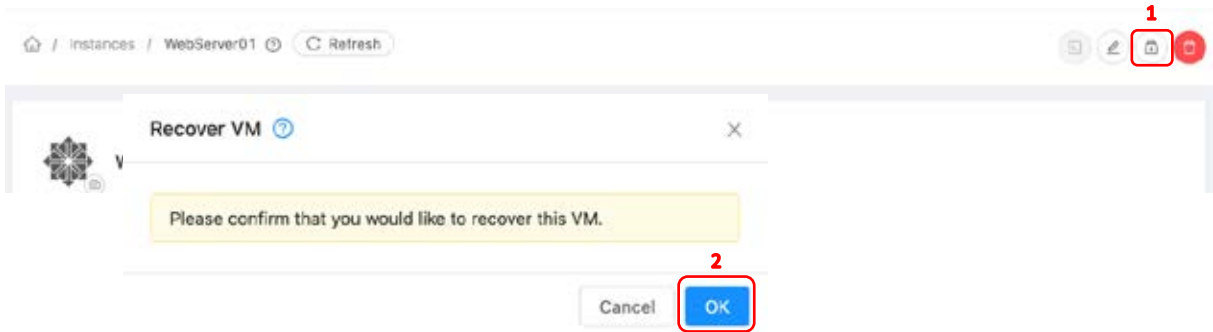


Note

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 *Recover Instance* to proceed. After this, the instance will be available again.



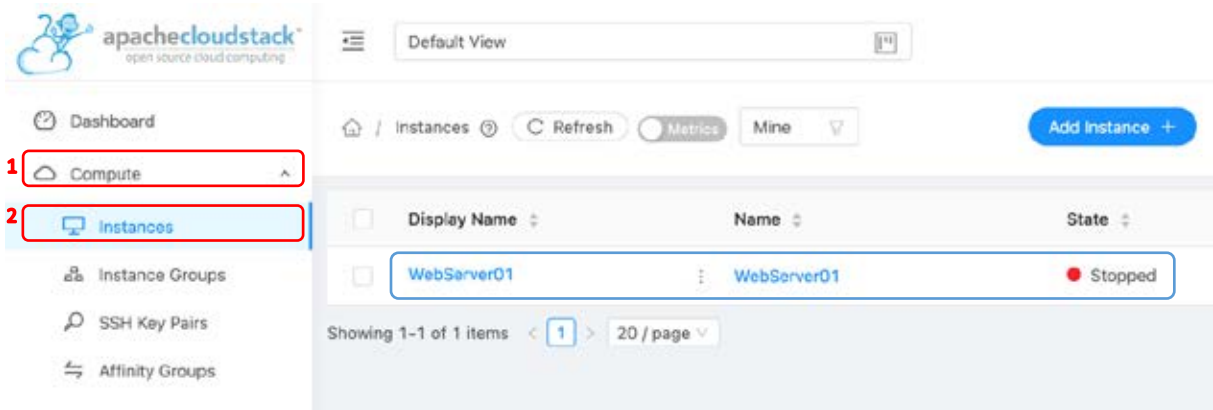
33. Logout, and then login with the following credentials:

Username: *user-1*

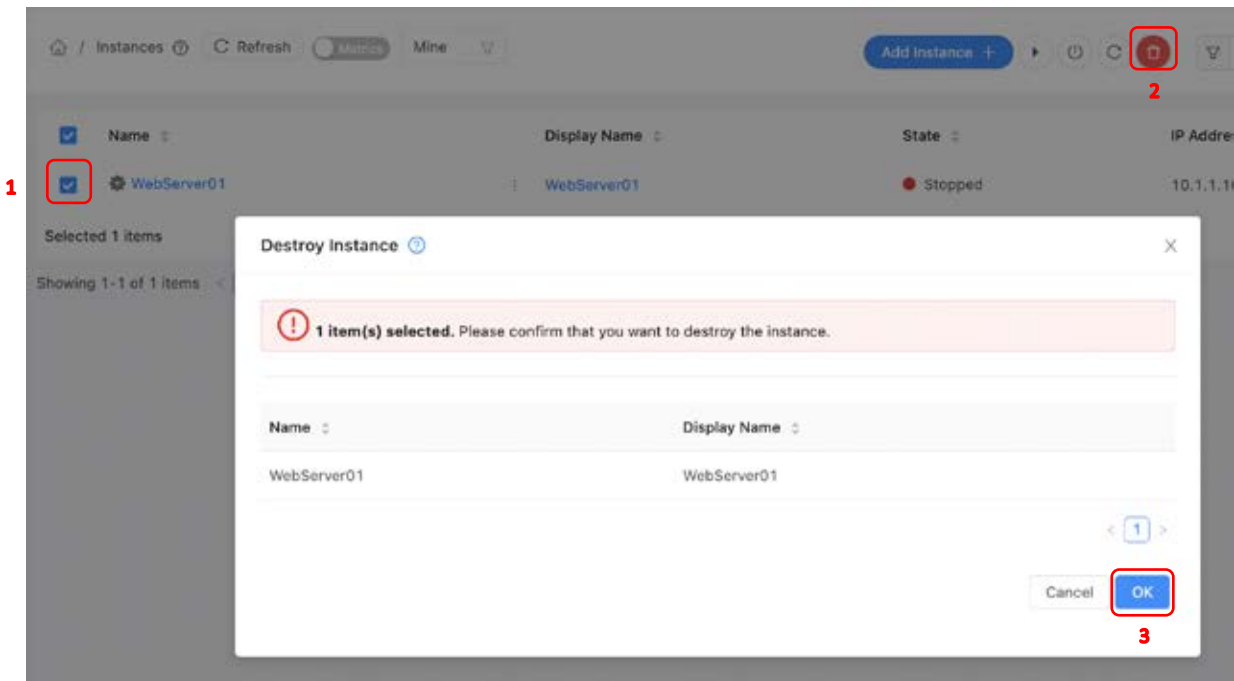
Password: *password*

Domain: *poc*

34. In the left navigation pane, click *Compute > Instances* to view the recovered instance.



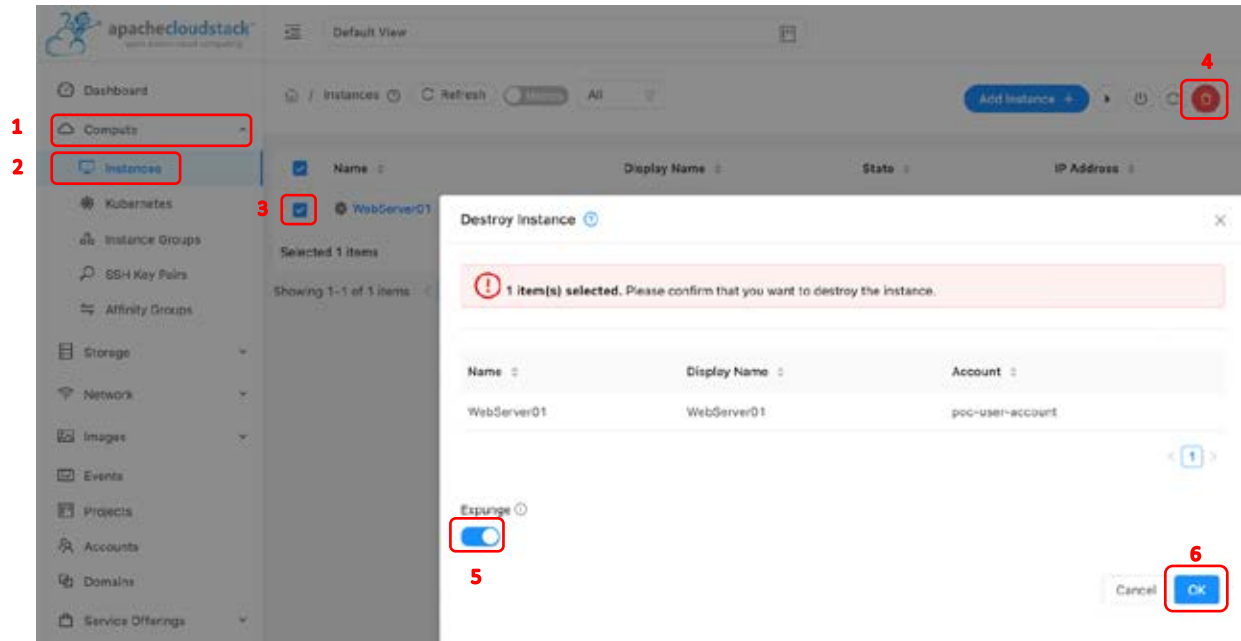
35. Select the *WebServer01* instance, followed by clicking *Destroy Instance* and then click *OK*.



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*.



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

Overview

Virtual Private Cloud lets you provision an architecture that resembles a traditional physical network.

VPC implements:

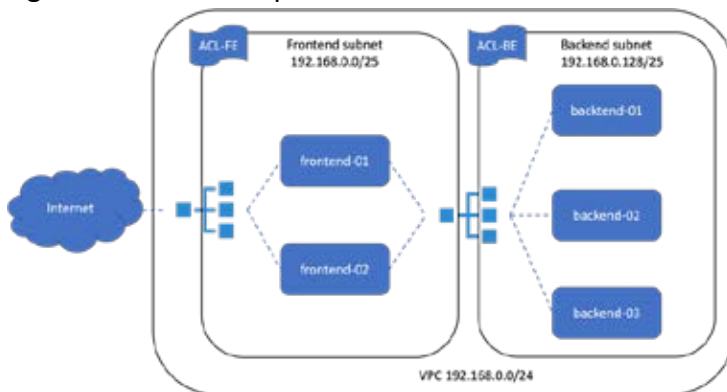
- Tiering isolation
- ACL
- Site-to-site IPsec VPN
- Client VPN
- Internal and External Load Balancer

Topics covered

- How to create VPC
- How to create ACLs List
- How to subnetting VPC
- How to create internal and external Load Balancer

Architecture

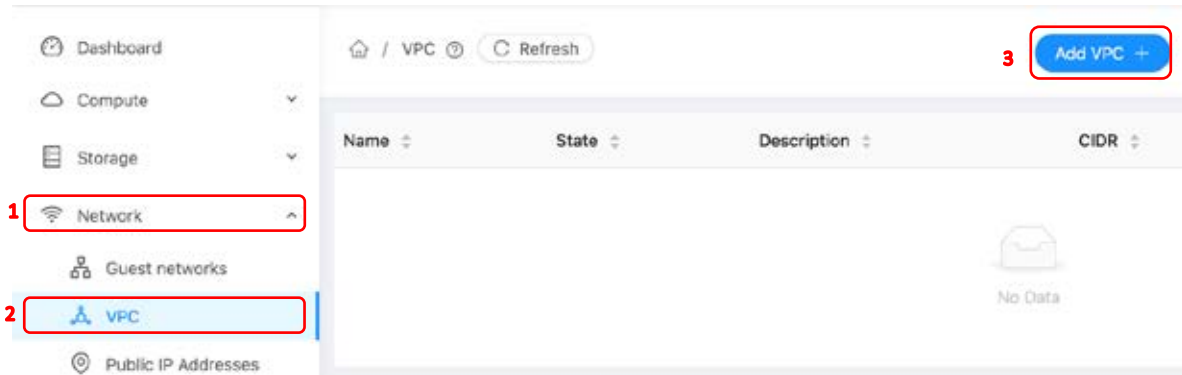
In this use case, you will implement a simple architecture that will demonstrate the ability to deliver services using the VPC components. We will define a VPC CIDR, subnetting it into 2 tiers; the first will be the frontend that will deliver the content through load balancing, and second subnet, will 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

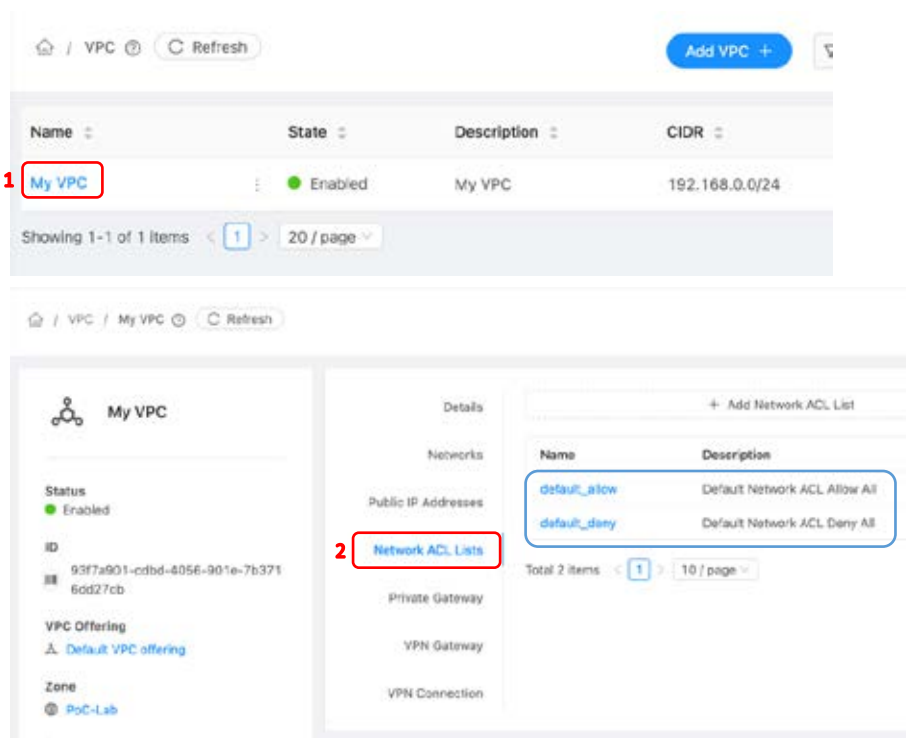
① Note

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.

① 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**:

1

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

Add ACL List X

* ACL List Name :

* Description :

2

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

1

ACL List Name: *ACL-BE*
Description: *Backend ACL List*

ACL List Name:

Description:

Cancel **OK**

- 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](#).

+ Add ACL

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

#Rule:

CIDR list:

Action:

Protocol:

Traffic Type:

Description:

Cancel **OK**

- 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](#).

+ Add ACL

#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.**

Edit rule ✕

#Rule:

CIDR list:

Action:

Protocol:

Start Port:

End Port:

Traffic Type:

Description:

- Click again in Add ACL to deny any other ingress traffic from frontend and fill in the form as follows and then click [Ok](#).

+ Add ACL

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

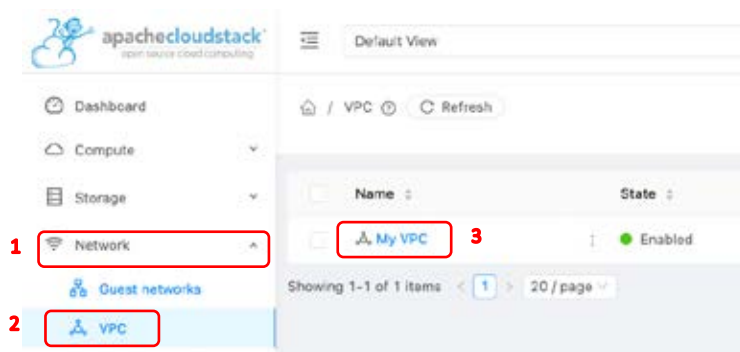
Dialog box titled "Edit rule" with the following fields:

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

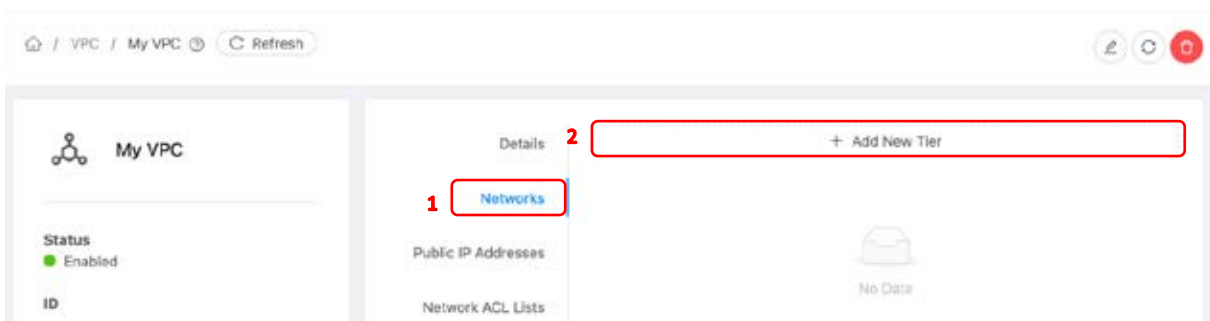
Buttons: Cancel, OK

Creating Tiers

11. In the left navigation pane, click *Network* > *VPC* and then 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*

Add New Tier ✕

* Name ⓘ

* Network Offering ⓘ

* Gateway ⓘ

* Netmask ⓘ

External Id ⓘ

* ACL ⓘ

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*

External Id: Blank
ACL: *ACL-BE*

Add New Tier ✕

* Name ⓘ
Backend Tier

* Network Offering ⓘ
Offering for Isolated Vpc networks with internal LB support

* Gateway ⓘ
192.168.0.129

* Netmask ⓘ
255.255.255.128

External Id ⓘ
ID of the network in an external system

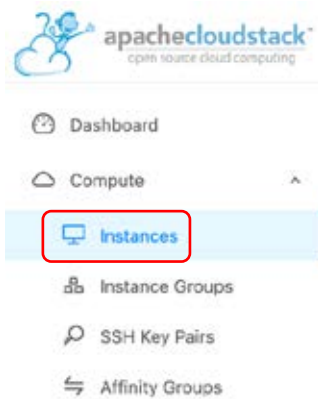
* ACL ⓘ
ACL-BE (Backend ACL List)

Cancel **OK**

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*.



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: *#!/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]*

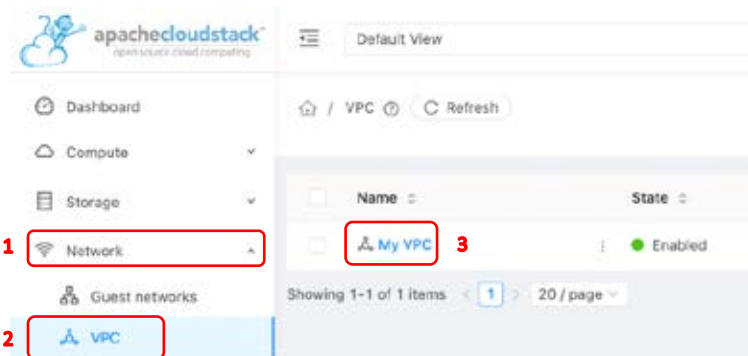
Group: *Blank*

Keyboard Language: *Blank*

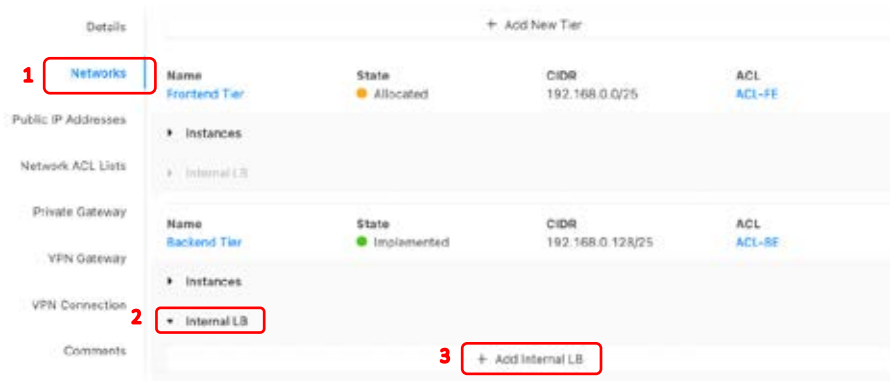
Start Instance: *Yes*

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.

Name	State	CIDR	ACL
Backend Tier	Implemented	192.168.0.128/25	ACL-BE

Instances

Internal LB

+ Add Internal LB

Name	Source IP Address	Algorithm	Account
Backend LB service	192.168.0.168	roundrobin	poc-user-account

22. Click *Backend LB Service*.

Name	State	CIDR	ACL
Backend Tier	Implemented	192.168.0.128/25	ACL-BE

Instances

Internal LB

+ Add Internal LB

Name	Source IP Address	Algorithm	Account
Backend LB service	192.168.0.168	roundrobin	poc-user-account

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

Internal LB / Backend LB service Refresh

2 + Assign VMs

Backend LB service

Details 1 Assigned VMs

24. Select all backend instances and then click *Ok*.

Assign VMs

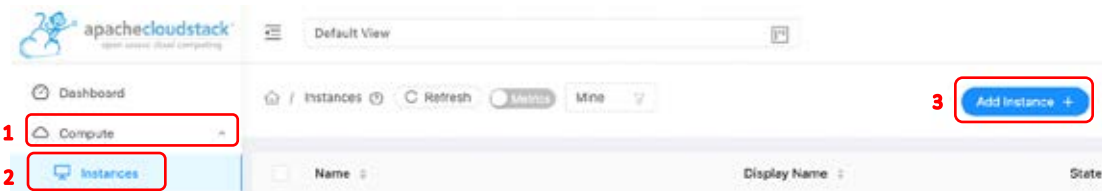
Name	State	Internal name	Display Name	IP Address	Account	Zone Name	Select
backend-instance-03	Running		backend-instance-03		poc-user-account-1	PoC-Lab	<input checked="" type="checkbox"/>
backend-instance-02	Running		backend-instance-02		poc-user-account-1	PoC-Lab	<input checked="" type="checkbox"/>
backend-instance-01	Running		backend-instance-01		poc-user-account-1	PoC-Lab	<input checked="" type="checkbox"/>

Total 3 items < 1 > 10 / page

Cancel OK

Creating Frontend Instances

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



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 follow (*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*:

Zone: *poc-zone*

Template/ISO

- Community: *CentOS 7.0 (64bit)*

Compute Offering: *Small Instance*

Networks:

- Backend Tier: *No*

- Frontend Tier: *Yes*

Advanced Mode: *Yes*

Userdata:

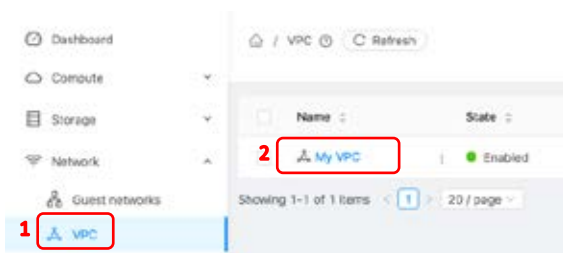
```
#!/bin/bash
backend_endpoint="Internal LB Source IP Address"
yum -y install httpd
cat << EOF > /etc/httpd/conf.d/cgi-enabled.conf
<Directory "/var/www/html/frontend">
Options +ExecCGI
AddHandler cgi-script .py
</Directory>
EOF
mkdir /var/www/html/frontend
cat << EOF > /var/www/html/frontend/index.py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

```
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*.



Acquire New IP ×

Please confirm that you want to acquire new IP

IP Address:

3

29. Click the added IP (not the source-nat).

Details + Acquire New IP

Networks Select Tier: View all

Public IP Addresses

<input type="checkbox"/>	IP Address	State	VM	Network
<input type="checkbox"/>	10.0.0.1 source-nat	Allocated		
<input type="checkbox"/>	10.0.0.1	Allocated		<input type="button" value="✖"/>

Network ACL Lists

Private Gateway

VPN Gateway Total 2 items < 1 > 10 / page

VPN Connection

Comments

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

Details Port Forwarding Load Balancing Comments

IP Address

31. Click *Load Balancing*, fill in the form as follows and then click *Add*.

Details Port Forwarding **Load Balancing** Comments

Name
www

Public Port
80

Private Port
80

Algorithm
Round-robin

Protocol
TCP

Add VMs

Name	Public Port	Private Port	Algorithm	Protocol	State	Stickiness	Add VMs	Action
No Data								

Total 0 items < 0 > 10 / page

32. Select the tier *Frontend Tier* and select all instances in the list as follows and then click *OK*.

Add VMs

Select Tier **Frontend Tier** **1**

Name	State	Display Name	Account	Zone	Select
frontend-instance-02 192.168.0.111 (Prim... x	● Running	frontend-instance-02	poc-user-account	PoC-Lab	2 <input checked="" type="checkbox"/>
frontend-instance-01 192.168.0.67 (Prima... x	● Running	frontend-instance-01	poc-user-account	PoC-Lab	3 <input checked="" type="checkbox"/>

Total 2 items < 1 > 10 / page

4

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*

① Note

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

① 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 fist script line, a `/bin/bash` was called to interpret the script:

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

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

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

3. The web server `httpd` was installed:

```
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:

```
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:

```
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:*

```
#!/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>\n<body>"
print "<meta http-equiv=\"refresh\" content=\"10\"/>"
print "<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>"
```

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:

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

2. The web server `httpd` was installed:

```
yum -y install httpd
```

3. Enable the `httpd` service:

```
systemctl enable httpd
```

4. Start the `httpd` service:

```
systemctl start httpd
```

5. Write the instance hostname to the file `/var/www/html/index.html`

```
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-premise 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 opensource cloud orchestration platform, in use by many of the worlds 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](#)