1.1 Setting up Cloudstack

In this lablette, you are going to set up cloudstack on the system.

The instructions we will follow originate at this web page:

<https://docs.cloudstack.apache.org/en/latest/quickinstallationguide/qig.html>

However… I don't want you to ONLY follow these instructions. I want you to AUTOMATE the installation. You will write 5 brief scripts.

1net -- create a network connection for cloudstackmgr1

2bridge -- change the network to be bridged (a requirement of cloudstack)

3common -- make software changes needed by all cloudstack systems

4mgt -- install/configure the management server

5host -- install/configure a KVM hosting system

As you create these scripts, please copy them and save them somewhere they won't be lost if cloudstackmgr1 is deleted.

In case things get really out of hand, I want you to be persuaded that deleting cloudstackmgr1 VM and starting again is not a real problem.

# How to delete cloudstackmgr1

A single command will get rid of cloudstackmgr1 and the disk you created for it.

That command is:

virsh undefine cloudstackmgr1 --remove-all-storage

Oh… but you have to stop the VM first

virsh destroy cloudstackmgr1 (yes, destroy just means stop immediately)

To recreate it, just

cd to /var/lib/libvirt/images

run: qemu-img create -f qcow2 -F qcow2 -b /var/lib/libvirt/boot/20centos7.qcow2 cloudstackmgr1.qcow2

cd to /root

run: mkcloudstackmgr1

# Configure cloudstackmgr1 network

Login to cloudstackmgr1.

You will notice the web instructions call for doing a yum install first thing. We cannot do that because we have no connection to the internet, yet.

So, first things first.

## Script 1net

Write a script called 1net that will configure the ifcfg-eth0 file with a static address.

HINT: you can make your script more functional by 1) keeping a copy of the existing ifcfg-eth0 file (before you overwrite it) and 2) doing error handling. You will not be graded on this -- but if

you want to impress me…

This is a bash script, so you need the shebang line to be:

#! /bin/bash

Follow this line with a number of values we will set to variables. We are taking this extra step so we can re-use this script on a different system later. We will only have to change the IP address information.

theIP=172.16.10.2

theGW=172.16.10.1

theNM=255.255.255.0

theHOST=cloudstackmgr1

Append the following lines to the ifcfg-eth0 file: [HINT you can use echo or a heredoc]

IPADDR=$theIP

GATEWAY=$theGW

NETMASK=$theNM

DNS1=8.8.8.8

Add this line to change the BOOTPROTO line in place:

sed -i 's/BOOTPROTO="dhcp"/BOOTPROTO=static/ /etc/sysconfig/network-scripts/ifcfg-eth0

This script (1net) can also handle configuring the system name. Because this is based on a template, it has no name (localhost). You will change that by adding a line to /etc/hosts and then using the hostnamectl command to set the name.

In your script:

Use echo and >> to append the following line to the file /etc/hosts

"$theIP $theHOST"

The line is simply two of the variables you set at the top of the script.

And now add the following command to your script.

hostnamectl set-hostname $theHOST

Again, we are using the value set to a variable at the top of the script. This way, we don't ever change this code, we just change the value at the top of the script for a different system, and can re-use the same code.

After all these changes, you will need to restart the network

Add to your script a systemctl command to restart the network service.

If you are going to test your script (always a good idea), and you used append to write to files, edit those files and remove your appended data before you test a second time.

Make your script executable for the root user (hint: use chmod).

Run your script.

When it is done, you should be able to:

1. do ip a and see 172.16.10.2 assigned to interface eth0
2. ping www.google.com
3. issue hostname --fqdn and get back the system name

You have now set your system up to be able to complete the cloudstack install.

## Create the bridge script 2bridge

Now that you have basic connectivity to the internet, you can go back through the instructions and do as described there.

In this script, copy the same variable assignments you used in 1net to the top of this script. We will use these here as well.

I am following the instructions now.

Skip the yum upgrade

Add the yum install command to your script.

Create the ifcfg-cloudbr0 file in /etc/sysconfig/network-scripts with the content described. EXCEPT, where it specifies IPADDR=, GATEWAY=, NETMASK=, use the variable as you did in 1net.

You can use echo with multiple lines to create this file, or you can use a heredoc.

You are replacing the content of ifcfg-eth0 so you can overwrite it with echo or a heredoc.

NetworkManager is not installed, so you can skip that step.

The network is already enabled, so you can skip that step.

Do not reboot the system.

You took care of the hostname in 1net, so skip that part.

Your script will restart the network (again) to adopt the new cloudbr0 and eth0 configurations.

## Using SSH to connect

After you have the network functional, you can use the ssh command from outernetwork1 to cloudstackmgr1.

You will need to edit /etc/ssh/sshd\_config on cloudstackmgr1 and make it so root user can log in

(PasswordAuthentication yes and PermitRootLogin yes).

You should verify that you can login as root over ssh. Cloudstack uses this method to connect with its hosting servers.

# Configure Cloudstack software

## Create a common parts script 3common

There are four sections of the quick install guide you can add to a script that will be used for any and all cloudstack systems you build.

These four parts are: SELinux, NTP, Cloudstack repository, and Firewall

Here are some notes on each part:

SELinux:

There are two commands here, setenforce and a re-write of the /etc/selinux/config file.

Take note that the config file is mostly comments (lines that start with "#") and don't affect selinux operation. So, we are really talking about writing a file of two lines.

NTP

The three commands required are given to you. They can be put in your script as is.

Repository

They give the text that is to be written to /etc/yum.repos.d in a file called cloudstack.repo. The file will not exist until you create it.

You can use echo or a heredoc to create it.

Firewall

Oddly, they put the commands to disable the firewall toward the end of the NFS section.

We are going to turn off the system OS firewall everywhere since this is a first install (for most of you). There are more elaborate instructions regarding which ports/protocols to permit which you would need to study if you were implementing Cloudstack in a work environment.

But for now you can just add the systemctl commands to stop and disable firewalld to your 3common script.

## Create a management server install script called 4mgt

These parts you only need to do for the a Cloudstack management server: NFS, Database and DB connector, Install cloudstack-management, Set up databases, download the System template.

NFS: we will be using a network shared storage option cloudstack enables via nfs.

The install requires you to install the software, create a file with two lines of content, create directories, and edit configuration information in two files.

The edit of /etc/idmapd.conf will require an sed command. Since we have not covered sed much, here is the command you can use in your script

sed -i 's/#Domain = local.domain.edu/Domain = wcc.edu/' /etc/idmapd.conf

We will use wcc.edu as a placeholder for a domain name.

The /etc/sysconfig/nfs edits can just be appended to the bottom of the file. Please do this.

Database and DB connector: the management server communicates constantly with the database. The DB can be on another server, but we will keep it all on this one system (as per this quick install guide).

There are some yum/wget/rpm commands to add to your script, and then there is another call for editing a section of a file. Again, I will give you an sed command that will do what is required in your script.

sed -i '/\[mysqld\]/a innodb\_rollback\_on\_timeout=1\ninnodb\_lock\_wait\_timeout=600\nmax\_connections=350\nlog-bin=mysql-bin\nbinlog-format = "ROW"' /etc/my.cnf

This finds the [mysqld] section header and then inserts all the required lines.

You finish the DB part by adding the systemctl commands to your script.

The last part of this section is to install a connector python uses to make calls to the database. This is another yum command.

Install cloudstack-management: now you have all the required pieces for cloudstack-management service to work.

Add the yum command given to your script.

Setup databases

Run the command provided from your script to set up the database, and the second command to setup management.

Install the template

A single really long command is given. Copy the whole command and paste it in your script.

## Create a kvm host install script called 5host

These parts you only do for a hosting system. Normally, a hosting system and the management server don't run on the same system.

But we are doing a single system build. So you will continue with the instructions but put this part in a separate script. Later on we are going to create another hosting system and you will be able to reuse the network, common, and hosting scripts.

Install cloudstack-agent

Google Cloud Platform already loads epel for us, but feel free to include it if you like.

Add the yum install commands to your script.

Configure qemu

This is a single edit of a line you are to find and change. The sed command does just this…

sed -i 's/#vnc\_listen = "0.0.0.0"/vnc\_listen = "0.0.0.0"/' /etc/libvirt/qemu.conf

Add this sed command to your script.

Configure libvirt

First they have you add several lines to /etc/libvirt/libvirtd.conf. You can use echo or a heredoc to add these lines.

There is another line to uncomment… This is another time you will use the sed command. Add this line to your script…

sed -i 's/#LIBVIRTD\_ARGS="--listen"/LIBVIRTD\_ARGS="--listen"/' /etc/sysconfig/libvirtd

And the final thing to add to your script is the systemctl command they supply which restarts the libvirtd service.

And something found in the detailed install

In the quickinstall instructions they leave out an important detail about nested virtualization. There are two lines that need to be added to /etc/cloudstack/agent/agent.properties.

guest.cpu.mode=host-passthrough

guest.cpu.features=vmx

You can append these to the file using echo.