# RHCSA for Red Hat OpenStack - using OpenStack Creating and Accessing Instances: Compute and Image Services

## Launching instances

As project demo, list possible ingredients for launching an instance: images, flavors and networks.

source demorc  
openstack image list  
openstack flavor list  
openstack network list

For more information about flavors, try the --long option: openstack flavor list --long

Now launch the instance:

openstack server create --image cirros --flavor 1 --nic net-id=ID-OF-PRIVATE myserver

Have a look at the instance’s properties: Status, Power state, task state, vm state, addresses, flavor, image, security groups, key. Note that the server create command returns immediately, but the instance launch may continue in the background.

Check the instance status by running openstack server list. It should be ACTIVE and it should have an IP address on the private network.

Launch another instance with exactly the same properties:

openstack server create --image cirros --flavor 1 --nic net-id=ID-OF-PRIVATE myserver  
openstack server list

It’s possible to run two servers with same name. Their IDs are by definition unique.

Delete all instance except the two myserver ones openstack server delete INSTANCE1 INSTANCE2 ...

Try to delete one of the myserver instances openstack server delete myserver. This fails because the name is not unique. Run the command a second time but use the ID instead of the name.

Play with different columns and display formats, for example

openstack server list -c ID -c Name -f value  
openstack server list -c Name -c addresses -f yaml

## Snapshotting instances

Snapshot the instance. A snapshot can be taken from a running instance. This is probably OK if the instance is quiet; if it is very busy, you should ensure that the data on the instance’s disk is consistent, for example by shutting it down.

openstack server image create myserver --name alt-cirros

This creates a new image named alt-cirros. Images can have properties, and in this case it is obvious from the properties that this image comes from a server snapshot. The other properties identify the original image, the server, and the user who took the snapshot.

Check if Glance really contains the new image openstack image list

To avoid creating images that contain malware, many clouds limit image uploading to admin users, but snapshots are usually allowed.

Use the snapshot by launching an instance from it:

openstack server create --image cirros --flavor 1 --nic net-id=ID-OF-PRIVATE myserver  
openstack server list

Copy the snapshot to the local filesystem. This can be useful as part of a backup solution or to migrate the instance to another cloud: openstack image save alt-cirros --file myserver.qcow2

You can copy any image this way, not just snapshots.

## Interactive access to an instance

### Console access

Retrieve a URL for an interactive console: openstack console url show myserver

The URL looks similar to this: <http://192.168.1.222:6080/vnc_auto.html?token=b86b7eeb->...

For security reasons, it is valid for 10 minutes only, after which you have to generate another URL. The token component of the URL ensures that not just anybody can access the console.

Paste the URL in a browser. If the console window is just black, click the gray bar on top of it and hit Enter a few times.

Display the instance’s startup log with openstack console log show. To see only the last 20 lines, add option --lines 20.

### Network access

For network access, a server needs open ports in the firewall and a floating IP.

Remove all security groups except default, then create a new security group named *ssh*.

openstack security group list

openstack security group delete ALL\_SECGROUPS\_EXCEPT\_DEFAULT

openstack security group create ssh

List its rules. It should have two egress rules: One for IPv4, the other for IPv6.

openstack security group rule list ssh --long

Add an ingress rules for TCP port 22.

openstack security group rule create --dst-port 22 --protocol tcp --ingress ssh

The ingress and protocol options are not needed, since ingress and TCP are defaults.

You can narrow down rules by specifying a source IP address range, in which case only traffic from that address range is accepted. You can also specify a remote security group, which only accepts traffic from servers in that security group. By default, this is not limited, as indicated by the rule properties remote\_ip\_prefix and remote\_group\_id. Run openstack help security group rule create to explore the required command line options.

Run openstack server add security group myserver ssh to add the security group to the server.

You can create several networks with the same IP address range - since OpenStack isolates them from each other, no conflicts occur. As a consequence, it is not possible to route traffic from outside to a server’s IP address, since the cloud might have several servers with the same address.

The solution for this problem is the floating IP. When you associate a floating IP to a server, the router redirects traffic destined to the floating IP to the correct server. A project can create a few floating IPs and assign them to various servers - they “float” from server to server as servers are deleted and recreated.

List available floating IPs. If there aren’t any, create one. Associate an available floating IP to myserver.

openstack floating ip list

openstack floating ip create public # only if there is no free floating ip

openstack server add floating ip myserver FLOATING\_IP

The commands openstack server list and openstack server show myserver now display the floating IP in addition to the IP address on the private network.

Can you log on?

ssh -l cirros FLOATING\_IP

By default, ssh refuses logging on if the server’s host key doesn’t correspond with its database of recorded host keys. If this is the case, remove ~/.ssh/known\_hosts and try again.

The password for the cirros user is cubswin:) (with the smiley).

Can you ping the instance?

ping FLOATING\_IP

This fails because the firewall blocks ICMP traffic. Add an ICMP rule to the security group and try again.

openstack security group rule create --protocol icmp ssh

ping FLOATING\_IP

Modifying the rule set has an immediate effect.

## Working with SSH keys

When you logged on to the instance via SSH, you still had to provide a password. Since you usually have to manage a large number of instances, password access is not practical. In fact, most cloud images you can download from the likes of Centos or Debian don’t even include accounts with passwords. The standard method of interactive access is via SSH keys.

Start by deleting all instances.

source ~/demorc

openstack server list

openstack server delete ALL\_INSTANCES

You can use existing keypairs or ask Nova to create a keypair. By default, Nova creates a keypair:

openstack keypair create mykey

This generates a keypair, stores the public key in Nova’s database under the name mykey, and outputs the private key. Nova doesn’t store or otherwise remember anything about the private key; it’s the user’s responsibility to manage it.

You can copy-paste the private key into a file, but it’s easier to redirect the output. Delete the keypair from Nova’s database, then recreate it.

openstack keypair delete mykey

openstack keypair create mykey > mykey.pem

You can also generate a keypair with another, e.g. ssh-keygen, and ask Nova to import it in its database. Run ssh-keygen and accept default settings by hitting Enter. By default, ssh-keygen generates RSA keys and stores them in ~/.ssh/id\_rsa and ~/.ssh/id\_rsa.pub.

Import the new keypair.

openstack keypair create myotherkey --public-key ~/.ssh/id\_rsa.pub

Launch two instances, each with a different key.

openstack server create --key-name mykey --image cirros --flavor 1 --nic net-id=PRIVATE myserver

openstack server create --key-name myotherkey --image cirros --flavor 1 --nic net-id=PRIVATE myotherserver

Find the key name in the instance details. Make the instances accessible.

openstack server show myserver

openstack server show myotherserver

openstack server add security group myserver ssh

openstack server add security group myotherserver ssh

openstack floating ip list

openstack floating ip create public # ensure you have two non-associated floating IPs

openstack server add floating ip myserver FLOATING\_IP

openstack server add floating ip myotherserver FLOATING\_IP

Use ping to confirm connectivity, then attempt to log on.

ssh cirros@IP\_OF\_MYOTHERSERVER

The key used by myotherserver, ~/.ssh/id\_rsa, is the default for the ssh client. Therefore, no need to specify it on the command line.

ssh cirros@IP\_OF\_MYSERVER -i mykey.pem

ssh doesn’t accept the private key file, as its permissions are too loose. Tighten them and try again.

chmod 600 mykey.pem

ssh cirros@IP\_OF\_MYSERVER -i mykey.pem

That should work. On myserver, you will find the public key in ~/.ssh/authorized\_keys, where the instance copied it. In the metadata section of the course, you will get some insight into the mechanism by which this is done.

In case you need the public key, you can retrieve it from Nova’s database.

openstack keypair show myotherkey --public key

Compare it with ~/.ssh/id\_rsa.pub and with ~/.ssh/authorized\_keys on the myotherserver instance. The three public keys should be identical.

## Metadata and Userdata

Metadata are a means of providing configuration information to an instance. Metadata can be a simple key-value pair, a script, a generic file that is copied to the instance, and a cloud-config file.

### An instance with a key-value pair

As before, delete all running servers.

Create a server with a key-value pair.

openstack server create ... --property nservers=15 --key-name mykey myserver

Presumably, the instance runs an application that consists of several server processes. This key value pair might contigure the number of those processes.

Note the property in the instance details.

While the instance starts up, add security group and floating IP.

openstack server add security group myserver ssh

openstack floating ip list

openstack floating ip create public # if required

openstack server add floating ip myserver FLOATING\_IP

Check connectivity with ping, then log on.

ssh cirros@FLOATING\_IP -i mykey.pem

Retrieve AWS metadata.

curl http://169.254.169.254/latest/meta-data

Feel free to explore, for example by displaying the flavor:

curl http://169.254.169.254/latest/meta-data/instance-type

The Amazon metadata API is documented at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>.

OpenStack has its own metadata API.

curl http://169.254.169.254/openstack/latest

There are a few JSON files, and two files named password and user\_data. The password function is probably not used by cloud images, and the openstack client doesn’t seem to provide an option to set the password on an instance. user\_data is empty because you didn’t provide the user-data option when launching the instance.

Vendor\_data documentation is at https://docs.openstack.org/nova/latest/user/vendordata.html.

curl http://169.254.169.254/openstack/latest/meta\_data.json

Most metadata is in this file. You should be able to find the nservers key-value pair as well as the SSH keys (twice). Other data is included, such as a random number seed (since virtual machines don’t run on hardware, they have problems generating good random number seeds; instead, they can use the one provided in Nova’s metadata).

The network\_data.json file contains data about the instance’s networks, including implementation information. We will revisit it in the operator course, but feel free to take a look.

Log out from the instance and delete it.

### Running a script at instance launch time

A script gives the user the possibility to modify the instance when it starts up in an entirely open-ended way. Create a simple shell script, for example

cat > myscript.sh <<EOF

#!/bin/sh

echo "I was here" > /home/cirros/iwashere.txt

data >> /home/cirros/iwashere.txt

EOF

This script contains an error. Launch an instance that executes this script when it starts.

openstack server create ... --key-name mykey --user-data myscript.sh myserver

Add the ssh security group and a floating IP, then log on with ssh cirros@FLOATING\_IP -i mykey.pem.

Check if the file iwashere.txt exists and what it contains. It should contain a date, but it doesn’t. Check the user\_data file in the OpenStack metadata.

curl <http://169.254.169.254/openstack/latest/user_data>

Log off and display the console log to find out why iwasthere.txt might not have the expected content. You will find text similar to this:

/run/cirros/datasource/data/user-data: line 4: data: not found

WARN: /etc/rc3.d/S95-cirros-userdata failed

Correct myscript.sh, delete the instance and create a new myserver, this time using the correct script.

Note that the above error message is specific to Cirros. Other cloud images generate other messages when user-data contains an error.

### Injecting a file

In contrast to metadata, files are indeed “injected” to the instance, and not retrieved via the metadata service. Be aware that injected files are limited to a size of 10Kbytes. Furthermore, file injection will be obsoleted at a later Nova release.

Delete myserver and run an instance with a customized motd file.

openstack server delete myserver

echo “Welcome to Cirros!” > motd

openstack server create myserver --file /etc/motd=motd --key-name mykey ...

Add security group and floating IP, then log on with ssh cirros@FLOATING\_IP -i mykey.pem. The message of the day should be displayed.

### Using cloud-config to customize an instance

Since Cirros doesn’t include a cloud-config processor, use a Fedora cloud image for these exercises (Fedora images are among the smallest “real” operating systems). Delete the instance, download the OpenStack image from <http://cloud.fedoraproject.org> and add it to the Glance image store.

openstack server delete myserver

wget http://cloud.fedoraproject.org/....

openstack image create --disk-format qcow2 --file Fedora-Cloud-Base-... fedora

openstack image list

While the image is downloading, explore cloud-init documentation at <http://cloudinit.readthedocs.io> and examples for cloud-config files at <http://cloudinit.readthedocs.io/en/latest/topics/examples.html>.

Create a file that adds a user named *plural* and sets the main domain name server to 8.8.8.8. The first character must be a hash sign. Note that YAML doesn’t accept tab characters: Use spaces for indentation.

To make user *plural* accessible using myotherkey, which corresponds to the default ssh keys under .ssh, include ssh\_authorized\_keys in the cloud-config file. The command below lists the public key and includes it:

cat >> ud.yaml <<EOF

#cloud-config

groups:

- pluralsight

# Add a user to the system. Users are added after groups are added.

# “default” is required to keep the default user on the Fedora image, “fedora”.

users:

- default

- name: plural

gecos: Plural M. Sight, training manager

primary\_group: plural

groups: pluralsight

ssh\_authorized\_keys:

- $(openstack keypair show myotherkey --public-key)

# generate a non-default resolv.conf file

# while cloud-config has a special module for setting up resolv.conf,

# it doesn’t seem to work with Fedora 28. Instead, manually craft the file.

# The pipe sign after the content: key is YAML’s line continuation character.

write\_files:

- content: |

# customized resolv.conf

nameserver 8.8.8.8

path: /etc/resolv.conf

EOF

Check the content of ud.yaml to ensure that the key has been included correctly.

Launch a Fedora instance with this cloud-config file as user data.

openstack server create --image fedora --flavor 2 --key-name mykey --user-data ud.yaml myfedora

A few notes:

* Since Fedora is a bit heavier than Cirros, the tiny flavor is too small. Flavor 2 works.
* mykey will be installed for the default Fedora user named *fedora*, not for *plural*.
* Your cloud is too small to run two Fedora instances at the same time. A second instance will quickly end up in an ERROR state due to lack of resources.
* After deleting an existing Fedora instance, wait a minute until all resources have been released before you run another one.
* Starting up a real OS like Fedora takes several minutes if you run Packstack on a virtual machine. Refer back to the Packstack installation exercise for creating a Packstack configuration that features a CPU with virtualization extensions.

Even on a CPU with virtualization extension, starting up Fedora takes longer than Cirros. Use the console log to check the progress.  
Due to a bug in Newton, redirecting the Fedora console log to a file or a pipe will fail, probably because Fedora outputs non-ASCII data. Use the --lines option instead of redirection: openstack console log show myfedora --lines 33.

Fedora’s console log is different from Cirros. In particular, the console log contains cloud-init output and should include the confirmation that both SSH keys are installed:

ci-info: ++++++++++Authorized keys from /home/plural/.ssh/authorized\_keys for user plural+++++++++++

ci-info: +---------+-------------------------------------------------+---------+-------------------+

ci-info: | Keytype | Fingerprint (md5) | Options | Comment |

ci-info: +---------+-------------------------------------------------+---------+-------------------+

ci-info: | ssh-rsa | de:ea:ce:34:32:27:33:a4:b9:8a:63:43:8f:cb:3b:64 | - | stack@newton.home |

ci-info: +---------+-------------------------------------------------+---------+-------------------+

ci-info: ++++++++++Authorized keys from /home/fedora/.ssh/authorized\_keys for user fedora+++++++++++

ci-info: +---------+-------------------------------------------------+---------+-------------------+

ci-info: | Keytype | Fingerprint (md5) | Options | Comment |

ci-info: +---------+-------------------------------------------------+---------+-------------------+

ci-info: | ssh-rsa | b4:e8:78:db:d1:cc:fa:3f:8f:48:59:67:20:87:37:2b | - | Generated-by-Nova |

ci-info: +---------+-------------------------------------------------+---------+-------------------+

Add security group and floating IP, then try to log on as *plural* ssh plural@172.24.4.230.

Check if /etc/resolv.conf has the expected content. If your network setup allows you to reach 8.8.8.8 from the Packstack server, you can also reach it from the instance.

ping 8.8.8.8

ping www.google.com

Experiment further, e.g. use cloud-config to install a package.

### The config drive

An alternative to the metadata API is the config drive. While some clouds add a config drive to instances by default, this Packstack installation doesn’t. You need to add the config-drive option when launching an instance.

First, delete any instances that might be running. Then launch an instance with config drive.

openstack server create --user-data myscript.sh --property nservers=15 --config-drive true ... myserver

Add security group and floating IP.

ssh cirros@FLOATING\_IP

lsblk

The config drive is a CDROM named /dev/sr0.

sudo mkdir /mnt/config

sudo mount /dev/sr0 /mnt/config

cd /mnt/config

Explore the config drive and compare it with the information you can get from http://169.254.169.254.

This concludes the Compute Service exercises.