Install OpenStack Victoria on CentOS 8 With Packstack

By **Josphat Mutai** - January 11, 2021

You can support us by downloading this article as PDF from the Link below.

Download the guide as PDF

OpenStack is a widely deployed cloud operating system for controlling small to large pools of compute servers, storage, and networking resources throughout datacenter / multiple datacenters.

OpenStack comes with a dashboard (Horizon) that gives administrators control of the systems while empowering end users and tenants to provision resources through a web interface. Command line interface and REST API is also available for management and resources provisioning.

The most recent release of OpenStack as of this article writing is Victoria, which is the 22nd release of OpenStack. In the Victoria release there are outstanding new features and improvements. Improvements on integration with Kubernetes, advanced support for FPGAs and solutions for complex networking issues is addressed as well. This guide will walk you through the complete installation of OpenStack Victoria on CentOS 8 using Packstack.

Install OpenStack Victoria on CentOS 8 With Packstack

Packstack is command line utility that uses Puppet modules to deploy various parts of OpenStack on multiple pre-installed servers over SSH automatically. Currently it only supports deployment on CentOS, Red Hat Enterprise Linux (RHEL) and compatible derivatives of both are supported.

I'll be performing the installation on a server with the following hardware.

```
CPU: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz (12 Cores)
Memory: 128GB RAM
Disk: 2 x 1TB SSD
Network: 1Gbit
IPV4 Adresses: 1 x IPV4 + /27 Subnet (30 IPs)
```

The operating system installed on the server where we're performing the deployment is CentOS 8 minimal edition.

Step 1: Set hostname, DNS and Update System

Set correct hostname for the server.

```
sudo hostnamectl set-hostname openstack.example.com
```

Ensure local name resolution is working on your server. Also consider adding an A record if you have a working DNS server in your infrastructure.

```
$ sudo vi /etc/hosts
192.168.10.11 openstack.example.com
```

Then Update system to ensure all packages are latest.

```
$ sudo dnf update -y
```

Reboot the system once it has been upgraded.

```
sudo reboot
```

Step 2: Enable repositories, disable NetworkManager

Enable the repositories that will be required to install OpenStack packages on a CentOS 8 server.

```
sudo dnf -y install epel-release
sudo dnf config-manager --enable PowerTools
```

On CentOS 8 network-scripts is deprecated and not installed by default. You need to install it manually.

```
sudo dnf -y install network-scripts
readlink $(readlink $(which ifup))
sudo touch /etc/sysconfig/disable-deprecation-warnings
```

Disable NetworkManager and Firewalld services then enable network service.

```
sudo systemctl disable --now NetworkManager
sudo systemctl enable network
sudo systemctl start  network
```

Ensure you're using Static IP settings to avoid losing network disconnection. See below example

```
$ sudo vi /etc/sysconfig/network-scripts/ifcfg-eno1
DEVICE=eno1
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.10.11
NETMASK=255.255.255.0
GATEWAY=192.168.10.254
IPV6INIT=no
```

If Firewalld service is running consider disabling it for ease of configurations.

```
sudo systemctl disable --now firewalld
```

Reboot the system to confirm networking is working.

```
sudo reboot
```

Step 3: Add OpenStack Victoria repository

Check available *centos-release-openstack* package releases using the following commands.

```
$ sudo dnf search centos-release-openstack
CentOS-8 - Advanced Virtualization
257 kB/s | 133 kB     00:00
CentOS-8 - Ceph Nautilus
530 kB/s | 388 kB     00:00
CentOS-8 - RabbitMQ 38
239 kB/s | 137 kB     00:00
CentOS-8 - NFV OpenvSwitch
35 kB/s |  16 kB     00:00
CentOS-8 - OpenStack victoria
6.6 MB/s | 2.7 MB     00:00
============================================================ Name
Matched: centos-release-openstack
============================================================
centos-release-openstack-train.noarch : OpenStack from the CentOS
Cloud SIG repo configs
centos-release-openstack-ussuri.noarch : OpenStack from the CentOS
Cloud SIG repo configs
centos-release-openstack-victoria.noarch : OpenStack from the CentOS
Cloud SIG repo configs
```

I'll install Victoria release repository package

```
sudo dnf -y install centos-release-openstack-victoria
```

Update all current OS system packages to ensure in sync.

```
sudo dnf update -y
```

Reboot the system after the upgrade:

```
sudo reboot
```

Step 4: Install Packstack and generate answers file

Install *packstack* which is provided by openstack-packstack package.

```
sudo dnf install -y openstack-packstack
```

Confirm successful installation by querying for the version.

```
$ packstack --version
packstack 17.0.0
```

Command options:

```
$ packstack --help
```

Generate answers file which defines variables that modifies installation of OpenStack services.

```
$ sudo su -
# packstack --os-neutron-ml2-tenant-network-types=vxlan \
--os-neutron-l2-agent=openvswitch \
--os-neutron-ml2-type-drivers=vxlan,flat \
--os-neutron-ml2-mechanism-drivers=openvswitch \
--keystone-admin-passwd=<admin password> \
--nova-libvirt-virt-type=kvm \
--provision-demo=n \
--cinder-volumes-create=n \
--os-heat-install=y \
--os-swift-storage-size=10G \
--gen-answer-file /root/answers.txt
```

Set the Keystone / admin user password `--keystone-admin-passwd`. If you don't have extra storage for Cinder you can use loop device for volume group by *cinder-*

volumes-create=y but performance will not be good. Above are the standard settings but you can pass as many options as it suites your desired deployment.

You can modify the answers file generated to add more options.

```
# vi /root/answers.txt
```

Step 5: Install OpenStack Victoria on CentOS 8 With Packstack

If satisfied with the contents in the answers file initiate deployment of OpenStack Victoria on CentOS 8 With Packstack:

```
 # packstack --answer-file /root/answers.txt --timeout=3000
```

Installation process should be started and may take some time to complete:

```
....
Gathering ssh host keys for Nova migration           [ DONE ]
Preparing Nova Compute entries                       [ DONE ]
Preparing Nova Scheduler entries                     [ DONE ]
Preparing Nova VNC Proxy entries                     [ DONE ]
Preparing OpenStack Network-related Nova entries     [ DONE ]
Preparing Nova Common entries                        [ DONE ]
Preparing Neutron API entries                        [ DONE ]
Preparing Neutron L3 entries                         [ DONE ]
Preparing Neutron L2 Agent entries                   [ DONE ]
Preparing Neutron DHCP Agent entries                 [ DONE ]
Preparing Neutron Metering Agent entries             [ DONE ]
Checking if NetworkManager is enabled and running    [ DONE ]
Preparing OpenStack Client entries                   [ DONE ]
Preparing Horizon entries                            [ DONE ]
Preparing Swift builder entries                      [ DONE ]
Preparing Swift proxy entries                        [ DONE ]
Preparing Swift storage entries                      [ DONE ]
Preparing Heat entries                               [ DONE ]
Preparing Heat CloudFormation API entries            [ DONE ]
Preparing Gnocchi entries                            [ DONE ]
Preparing Redis entries                              [ DONE ]
Preparing Ceilometer entries                         [ DONE ]
Preparing Aodh entries                               [ DONE ]
Preparing Puppet manifests                           [ DONE ]
Copying Puppet modules and manifests                 [ DONE ]
Applying 192.168.10.11_controller.pp
192.168.10.11_controller.pp:                         [ DONE ]
```

```
Applying 192.168.10.11_network.pp
192.168.10.11_network.pp:                              [ DONE ]
Applying 192.168.10.11_compute.pp
192.168.10.11_compute.pp:                              [ DONE ]
Applying Puppet manifests                              [ DONE ]
Finalizing                                             [ DONE ]

 **** Installation completed successfully ******

Additional information:
 * Time synchronization installation was skipped. Please note that
unsynchronized time on server instances might be problem for some
OpenStack components.
 * File /root/keystonerc_admin has been created on OpenStack client
host 192.168.10.11. To use the command line tools you need to source
the file.
 * To access the OpenStack Dashboard browse to
http://192.168.10.11/dashboard .
Please, find your login credentials stored in the keystonerc_admin in
your home directory.
 * The installation log file is available at:
/var/tmp/packstack/20201216-023529-0df1tgus/openstack-setup.log
 * The generated manifests are available at:
/var/tmp/packstack/20201216-023529-0df1tgus/manifests
```

You can now source the keystone admin profile in your terminal session.

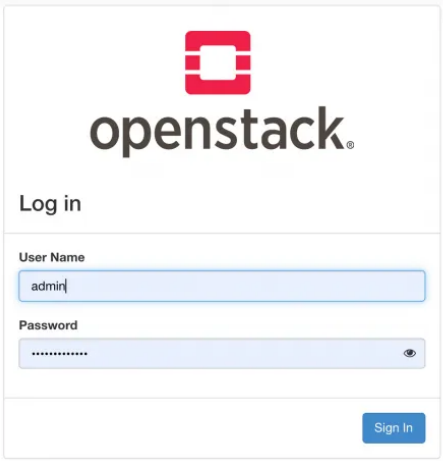
```
source ~/keystonerc_admin
```

Check if you can call the openstack CLI to interact with OpenStack services.

```
$ openstack service list
+----------------------------------+------------+----------------+
| ID                               | Name       | Type           |
+----------------------------------+------------+----------------+
| 016e1a0f299e4188a4ff2f0951041890 | swift      | object-store   |
| 02b03ebfe32a48a8ba1b4eb886fea509 | cinderv2   | volumev2       |
| 0ee374b1619e44dd8c3f1f8c8792b08b | nova       | compute        |
| 4eddc25d9c6c42c29ed4aaf3a690e073 | aodh       | alarming       |
| 51ec76355583449aac07c7570750bfda | heat       | orchestration  |
| 75797c5e394f419f9de85e8f424914fa | neutron    | network        |
| 75e2d698d2114d028769621995232a35 | glance     | image          |
| 84da19176cb84382a7a87d9461ab926e | placement  | placement      |
```

```
| 8d228baf96b24d97934d1f722337f0ee | heat-cfn    | cloudformation |
| 9e944a5b9a3d474ebc60fd85f0c080bd | cinderv3    | volumev3       |
| 9e9507529ec4454daebeb30183a06d16 | gnocchi     | metric         |
| bf915960baff410db3583cc66ee55daa | keystone    | identity       |
| fbb3e1eb3d6b489386648476e1c55877 | ceilometer  | metering       |
+----------------------------------+-------------+----------------+
```

To login to Horizon Dashboard I'll use the URL: http://192.168.10.11/dashboard



Step 6: Configure Neutron Networking

Migrate your primary interface network configurations to a bridge. These are the updated network configurations on my server.

```
$ sudo  vi /etc/sysconfig/network-scripts/ifcfg-eno1
DEVICE=eno1
ONBOOT=yes
TYPE=OVSPort
DEVICETYPE=ovs
OVS_BRIDGE=br-ex

$ sudo vi /etc/sysconfig/network-scripts/ifcfg-br-ex
DEVICE=br-ex
BOOTPROTO=none
ONBOOT=yes
TYPE=OVSBridge
DEVICETYPE=ovs
USERCTL=yes
PEERDNS=yes
IPV6INIT=no
IPADDR=192.168.10.11
```

```
NETMASK=255.255.255.0
GATEWAY=192.168.10.254
```

Create OVS bridge and add the interface.

```
sudo ovs-vsctl add-port br-ex eno1
```

Restart network service for the updates to be effected.

```
sudo systemctl restart network.service
```

Confirm IP address information.

```
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
master ovs-system state UP group default qlen 1000
    link/ether b4:2e:99:47:f1:df brd ff:ff:ff:ff:ff:ff
       valid_lft forever preferred_lft forever
3: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
group default qlen 1000
    link/ether f2:1d:71:2b:ab:66 brd ff:ff:ff:ff:ff:ff
4: br-ex: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UNKNOWN group default qlen 1000
    link/ether b4:2e:99:47:f1:df brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.11/24 scope global br-ex
       valid_lft forever preferred_lft forever
5: br-int: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
default qlen 1000
    link/ether 5a:bc:36:f1:b8:48 brd ff:ff:ff:ff:ff:ff
6: br-tun: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
default qlen 1000
    link/ether 2a:d9:4a:b2:84:47 brd ff:ff:ff:ff:ff:ff
```

Create private network on OpenStack.

```
$ openstack network create private
+---------------------------+--------------------------------------+
```

```
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | UP                                   |
| availability_zone_hints   |                                      |
| availability_zones        |                                      |
| created_at                | 2020-12-16T17:39:11Z                 |
| description               |                                      |
| dns_domain                | None                                 |
| id                        | 03eff42c-0b21-43e6-bbb6-164552279961 |
| ipv4_address_scope        | None                                 |
| ipv6_address_scope        | None                                 |
| is_default                | False                                |
| is_vlan_transparent       | None                                 |
| mtu                       | 1450                                 |
| name                      | private                              |
| port_security_enabled     | True                                 |
| project_id                | f9e4445b9ac14d4da47d0a0451f2e0c9     |
| provider:network_type     | vxlan                                |
| provider:physical_network | None                                 |
| provider:segmentation_id  | 10                                   |
| qos_policy_id             | None                                 |
| revision_number           | 1                                    |
| router:external           | Internal                             |
| segments                  | None                                 |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tags                      |                                      |
| updated_at                | 2020-12-16T17:39:11Z                 |
+---------------------------+--------------------------------------+
```

Create a subnet for the private network:

```
$ openstack subnet create --network private --allocation-pool \
    start=172.10.10.50,end=172.10.10.200 \
    --dns-nameserver 213.133.100.100 --dns-nameserver 213.133.99.99 \
    --subnet-range 172.10.10.0/24 private_subnet

+----------------------+-----------------------------------------
--+
| Field                | Value
|
+----------------------+-----------------------------------------
```

```
--+
| allocation_pools     | 172.10.10.50-172.10.10.200
|
| cidr                 | 172.10.10.0/24
|
| created_at           | 2020-12-16T17:40:11Z
|
| description          |
|
| dns_nameservers      | 213.133.100.100, 213.133.98.98,
213.133.99.99 |
| dns_publish_fixed_ip | None
|
| enable_dhcp          | True
|
| gateway_ip           | 172.10.10.1
|
| host_routes          |
|
| id                   | bd52f697-7e61-4f70-a416-78dde193b0c2
|
| ip_version           | 4
|
| ipv6_address_mode    | None
|
| ipv6_ra_mode         | None
|
| name                 | private_subnet
|
| network_id           | 03eff42c-0b21-43e6-bbb6-164552279961
|
| prefix_length        | None
|
| project_id           | f9e4445b9ac14d4da47d0a0451f2e0c9
|
| revision_number      | 0
|
| segment_id           | None
|
| service_types        |
|
| subnetpool_id        | None
|
```

```
| tags                  |
|
| updated_at            | 2020-12-16T17:40:11Z
|
+-----------------------+-----------------------------------------
--+
```

Create public network:

```
$ openstack network create --provider-network-type flat \
  --provider-physical-network extnet \
  --external public

+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | UP                                   |
| availability_zone_hints   |                                      |
| availability_zones        |                                      |
| created_at                | 2020-12-16T17:47:40Z                 |
| description               |                                      |
| dns_domain                | None                                 |
| id                        | 95cbb9bc-ddcc-412f-9496-3f77dff3f030 |
| ipv4_address_scope        | None                                 |
| ipv6_address_scope        | None                                 |
| is_default                | False                                |
| is_vlan_transparent       | None                                 |
| mtu                       | 1500                                 |
| name                      | public                               |
| port_security_enabled     | True                                 |
| project_id                | f9e4445b9ac14d4da47d0a0451f2e0c9     |
| provider:network_type     | flat                                 |
| provider:physical_network | extnet                               |
| provider:segmentation_id  | None                                 |
| qos_policy_id             | None                                 |
| revision_number           | 1                                    |
| router:external           | External                             |
| segments                  | None                                 |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tags                      |                                      |
```

```
| updated_at                | 2020-12-16T17:47:40Z                 |
+---------------------------+--------------------------------------+
```

Define subnet for the public network. It could be an actual public IP network.

```
$ openstack subnet create --network public --allocation-pool \
  start=192.168.10.100,end=192.168.10.200 --no-dhcp \
  --subnet-range 192.168.10.0/24 public_subnet

+----------------------+--------------------------------------+
| Field                | Value                                |
+----------------------+--------------------------------------+
| allocation_pools     | 192.168.10.100-192.168.10.200        |
| cidr                 | 192.168.10.0/24                      |
| created_at           | 2020-12-16T17:48:40Z                 |
| description          |                                      |
| dns_nameservers      |                                      |
| dns_publish_fixed_ip | None                                 |
| enable_dhcp          | False                                |
| gateway_ip           | 192.168.10.254                       |
| host_routes          |                                      |
| id                   | 0063aaf9-9e3d-4634-a4c7-ddf0e66c2b75 |
| ip_version           | 4                                    |
| ipv6_address_mode    | None                                 |
| ipv6_ra_mode         | None                                 |
| name                 | public_subnet                        |
| network_id           | 95cbb9bc-ddcc-412f-9496-3f77dff3f030 |
| prefix_length        | None                                 |
| project_id           | f9e4445b9ac14d4da47d0a0451f2e0c9     |
| revision_number      | 0                                    |
| segment_id           | None                                 |
| service_types        |                                      |
| subnetpool_id        | None                                 |
| tags                 |                                      |
| updated_at           | 2020-12-16T17:48:40Z                 |
+----------------------+--------------------------------------+
```

Create a router that will connect public and private subnets.

```
$ openstack router create private_router
+-------------------------+--------------------------------------+
| Field                   | Value                                |
+-------------------------+--------------------------------------+
```

```
| admin_state_up          | UP                                   |
| availability_zone_hints |                                      |
| availability_zones      |                                      |
| created_at              | 2020-12-16T17:50:14Z                 |
| description             |                                      |
| distributed             | False                                |
| external_gateway_info   | null                                 |
| flavor_id               | None                                 |
| ha                      | False                                |
| id                      | 0e3d364e-586a-4c17-854d-4e05cddb27fc |
| name                    | private_router                       |
| project_id              | f9e4445b9ac14d4da47d0a0451f2e0c9     |
| revision_number         | 1                                    |
| routes                  |                                      |
| status                  | ACTIVE                               |
| tags                    |                                      |
| updated_at              | 2020-12-16T17:50:14Z                 |
+-------------------------+--------------------------------------+
```

Set external gateway as public network on the router.

```
$ openstack router set --external-gateway public private_router
```

Link private network to the router.

```
$ openstack router add subnet private_router private_subnet
```

Check to ensure network connectivity is working.

```
$ ip netns show
qrouter-0e3d364e-586a-4c17-854d-4e05cddb27fc (id: 1)
qdhcp-03eff42c-0b21-43e6-bbb6-164552279961 (id: 0)

$ ip netns exec qrouter-0e3d364e-586a-4c17-854d-4e05cddb27fc ping -c
1 computingforgeeks.com
PING computingforgeeks.com (104.26.4.192) 56(84) bytes of data.
64 bytes from 104.26.4.192 (104.26.4.192): icmp_seq=1 ttl=57
time=21.10 ms

--- computingforgeeks.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 21.951/21.951/21.951/0.000 ms
```

Step 7: Spin a test instance

Our OpenStack Cloud platform should be ready for use. We'll download Cirros cloud image.

```
mkdir images
cd images
wget http://download.cirros-cloud.net/0.5.1/cirros-0.5.1-x86_64-
disk.img
```

Upload Cirros image to Glance

```
openstack image create --disk-format qcow2 \
  --container-format bare --public \
  --file ./cirros-0.5.1-x86_64-disk.img "Cirros-0.5.1"
```

Confirm image uploaded

```
$ openstack image list
+--------------------------------------+--------------+--------+
| ID                                   | Name         | Status |
+--------------------------------------+--------------+--------+
| 211daeef-eee7-4b13-a778-72c06b8d2c27 | Cirros-0.5.1 | active |
+--------------------------------------+--------------+--------+
```

Create Security Group for all access.

```
openstack security group create permit_all --description "Allow all
ports"
openstack security group rule create --protocol TCP --dst-port
1:65535 --remote-ip 0.0.0.0/0 permit_all
openstack security group rule create --protocol ICMP --remote-ip
0.0.0.0/0 permit_all
```

Create another security group for limited access – standard access ports *ICMP*, *22, 80, 443*

```
openstack security group create limited_access --description "Allow
base ports"
openstack security group rule create --protocol ICMP --remote-ip
0.0.0.0/0 limited_access
openstack security group rule create --protocol TCP --dst-port 22 --
remote-ip 0.0.0.0/0 limited_access
openstack security group rule create --protocol TCP --dst-port 80 --
```

```
remote-ip 0.0.0.0/0 limited_access
openstack security group rule create --protocol TCP --dst-port 443 --
remote-ip 0.0.0.0/0 limited_access
```

List all security groups:

```
$ openstack security group list
```

Confirming. rules in the security group.

```
$ openstack security group show permit_all
$ openstack security group show limited_access
```

Create Private Key

```
$ ssh-keygen # if you don't have ssh keys already
```

Add key to Openstack:

```
$  openstack keypair create --public-key ~/.ssh/id_rsa.pub admin
+-------------+-------------------------------------------------+
| Field       | Value                                           |
+-------------+-------------------------------------------------+
| fingerprint | 19:7b:5c:14:a2:21:7a:a3:dd:56:c6:e4:3a:22:e8:3f |
| name        | admin                                           |
| user_id     | 513f0abd6eba4b0fab2754166f38e0f2                |
+-------------+-------------------------------------------------+
```

Confirm keypair is available on OpenStack:

```
$ openstack keypair list
+-------+-------------------------------------------------+
| Name  | Fingerprint                                     |
+-------+-------------------------------------------------+
| admin | 19:7b:5c:14:a2:21:7a:a3:dd:56:c6:e4:3a:22:e8:3f |
+-------+-------------------------------------------------+
```

Listing available networks:

```
$ openstack network list
+--------------------------------------+---------+------------------
------------------+
| ID                                   | Name    | Subnets
|
```

```
+--------------------------------------+---------+------------------
------------------+
| 03eff42c-0b21-43e6-bbb6-164552279961 | private | bd52f697-7e61-
4f70-a416-78dde193b0c2 |
| 95cbb9bc-ddcc-412f-9496-3f77dff3f030 | public  | 0063aaf9-9e3d-
4634-a4c7-ddf0e66c2b75 |
+--------------------------------------+---------+------------------
------------------+
```

Check available instance flavors:

```
$ openstack flavor list
+----+-----------+-------+------+-----------+-------+-----------+
| ID | Name      |   RAM | Disk | Ephemeral | VCPUs | Is Public |
+----+-----------+-------+------+-----------+-------+-----------+
| 1  | m1.tiny   |   512 |    1 |         0 |     1 | True      |
| 2  | m1.small  |  2048 |   20 |         0 |     1 | True      |
| 3  | m1.medium |  4096 |   40 |         0 |     2 | True      |
| 4  | m1.large  |  8192 |   80 |         0 |     4 | True      |
| 5  | m1.xlarge | 16384 |  160 |         0 |     8 | True      |
+----+-----------+-------+------+-----------+-------+-----------+
```

Let's create an instance on the private network

```
openstack server create \
  --flavor m1.tiny \
  --image "Cirros-0.5.1" \
  --network private \
  --key-name admin \
  --security-group permit_all \
  mycirros
```

Check if the instance is created successfully.

```
$ openstack server list
+--------------------------------------+----------+--------+---------
-----------------------------+-------------+---------+
| ID                                   | Name     | Status | Networks
| Image        | Flavor  |
+--------------------------------------+----------+--------+---------
-----------------------------+-------------+---------+
| 043ba014-1670-4b50-8abf-50210c716611 | mycirros | ACTIVE |
private=172.10.10.113                | Cirros-0.5.1 | m1.tiny |
```

```
+--------------------------------------+----------+--------+---------
-----------------------------+-------------+---------+
```

To associate a floating IP from the public subnet use the guide below:

How To Assign a Floating IP Address to an Instance in OpenStack

Once floating IP is assigned you can ssh to the instance with private key.

```
$ ssh cirros@<floating-ip>
Warning: Permanently added '192.168.10.104' (ECDSA) to the list of
known hosts.
Enter passphrase for key '/Users/jkmutai/.ssh/id_rsa':

$ cat /etc/os-release
NAME=Buildroot
VERSION=2019.02.1-dirty
ID=buildroot
VERSION_ID=2019.02.1
PRETTY_NAME="Buildroot 2019.02.1"
$
$ ping computingforgeeks.com -c 2
PING computingforgeeks.com (104.26.5.192): 56 data bytes
64 bytes from 104.26.5.192: seq=0 ttl=56 time=22.220 ms
64 bytes from 104.26.5.192: seq=1 ttl=56 time=22.190 ms

--- computingforgeeks.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 22.190/22.205/22.220 ms
```

For instance deployment with Terraform check:

Deploy VM instance on OpenStack using Terraform

I hope this article helped you to install OpenStack Victoria on CentOS 8 Server. This deployment method is not for highly Production deployments of OpenStack. Review other OpenStack deployment methods fit for Production setups.

More articles on OpenStack:

How To run CentOS 8 Instance on OpenStack

Install / Run Fedora CoreOS (FCOS) on KVM / OpenStack

How To resize/extend Cinder Volume in OpenStack

How To rename an Instance/Server in OpenStack

You can support us by downloading this article as PDF from the Link below.

Download the guide as PDF

Josphat Mutai

https://computingforgeeks.com/

Founder of Computingforgeeks. Expertise in Virtualization, Cloud, Linux/UNIX Administration, Automation,Storage Systems, Containers, Server Clustering e.t.c.