



Chameleon Cloud Tutorial

National Science Foundation

Program Solicitation # NSF 13-602

CISE Research Infrastructure: Mid-Scale Infrastructure - NSFCloud (CRI: NSFCloud)

Cloud - DevStack Sandbox

Objectives

In this tutorial, we will show you how to install and access DevStack on a Chameleon Cloud bare metal server.

#	Action	Detail	Time (min)
1	Create Chameleon server	You will begin by logging into Chameleon Cloud's "Ironic" interface and creating a new server instance to run the new DevStack sandbox on.	5
2	Create a Dedicated DevStack User	DevStack comes configured out-of-the-box to utilize a dedicated "stack" user, which we will create and configure.	5
3	Launch DevStack	You will finish by creating a single configuration file, then launching DevStack by running a script.	20

Prerequisites

The following prerequisites are expected for successful completion of this tutorial:

- Chameleon Cloud account (<http://chameleoncloud.org/user/register/>)
- SSH client (for Windows users, download PuTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>))
- A basic knowledge of Linux

1. Create a Chameleon Server

Login to <https://ironic.chameleon.tacc.utexas.edu/dashboard/project/instances/> and create a new Chameleon Cloud server from the "Ironic" web interface with the following attributes. If no valid reservation exists, please refer to the Chameleon User Guide

(https://www.chameleoncloud.org/docs/user-guides/technology-preview-user-guide/#provisioning_resources) or this video (<https://goo.gl/veNCdI>) for help on how to create one. See figure 1 for instance creation details.

1. Instance name: **devstack-demo**
2. Availability zone: **Any Availability Zone**
3. Reservation: **<any valid reservation>**
4. Flavor: **baremetal**
5. Instance count: **1**
6. Instance boot source: **Boot from image**
7. Image name: **CC-CentOS7**
8. Click on the “**Access & Security**” tab
9. Select a pre-installed SSH key from the list, or, install one by clicking on “+”
10. Click: **Launch**

Launch Instance

Details *

Access & Security *

Networking *

Post-Creation

Availability Zone

Any Availability Zone

Reservation ?

Launch without reservation

myfirstlease_PaulR (f524c7db-9071-4eed-a364-045cd03b9c10)

Launch without reservation

Flavor * ?

baremetal

Instance Count * ?

1

Instance Boot Source * ?

Boot from image

Image Name

Select Image

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	baremetal
VCPUs	8
Root Disk	128 GB
Ephemeral Disk	0 GB
Total Disk	128 GB
RAM	11,264 MB

Project Limits

Number of Instances0 of 20 Used

Number of VCPUs0 of 160 Used

Total RAM0 of 225,280 MB Used

Cancel

Launch

Figure 1 - Create the Chameleon Cloud Server

The Chameleon Cloud server will begin building. When the server becomes available, click on the “Associate Floating IP” button at the end of its row. Select an available IP address from the list and click on “**Associate**”. See figure 2 below for details. Make note of this new IP address, as we will need it to complete the next step.

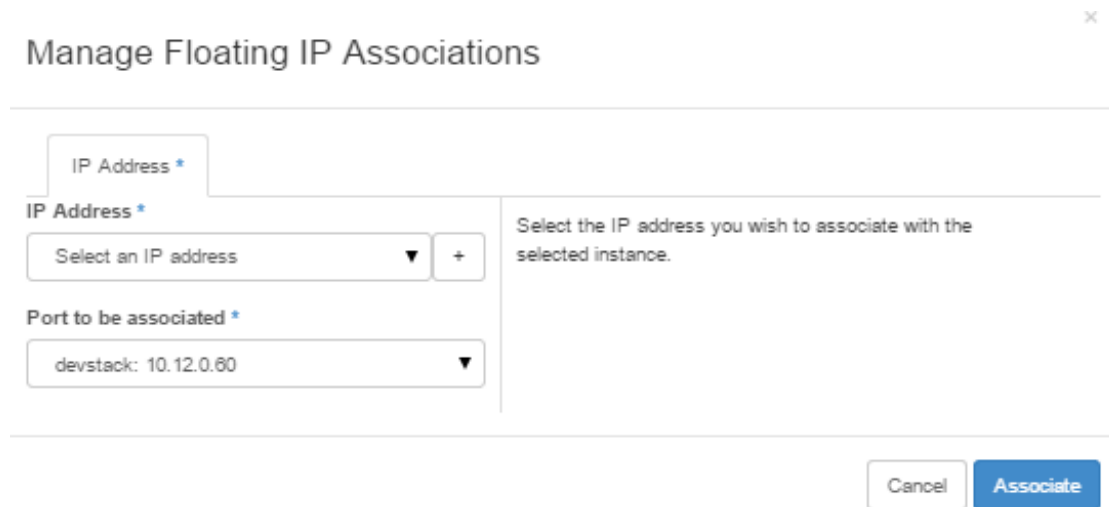


Figure 2 - Associate a Floating IP Address dialog box

Now we can connect to the new server via SSH using the floating IP address we just assigned. If you are a Linux or Mac user, type the following command in a new Terminal window (Windows users: follow along using PuTTY):

```
ssh cc@Floating.IP.Address
```

2. Add a dedicated DevStack user

We need to add a user with sudo privileges in order to install DevStack. Use the following commands to accomplish these tasks:

```
sudo adduser stack
sudo sh -c 'echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers'
```

We now need to copy our SSH public key to the new user, allowing us to login to that account.

```
sudo mkdir /home/stack/.ssh
sudo cp ~/.ssh/authorized_keys /home/stack/.ssh/
sudo chown -R stack:stack /home/stack/.ssh
```

From here on, we should use the new stack user we just created. We can now log out of the cc user account, and log back into the server as the stack user, as seen below:

```
ssh stack@Floating.IP.Address
```

3. Launch DevStack

Once we get logged back onto the server as the stack user, we will execute the following commands in order to download DevStack to its own folder in our home directory.

```
cd ~
git clone https://git.openstack.org/openstack-dev/devstack
cd devstack
```

Next, we want to create a `local.conf` configuration file inside the `devstack` directory, and ensure its contents appear exactly as shown below, except where indicated. In order to create and edit this file, we can either use the built-in text editor `vi`, or download the significantly more user friendly editor named `nano`. If you do not have prior experience using `vi` it is recommended you execute the following instruction in order to download and use `nano` to create the configuration file. Otherwise, create the file using `vi` if you are comfortable doing so.

```
sudo yum -y install nano

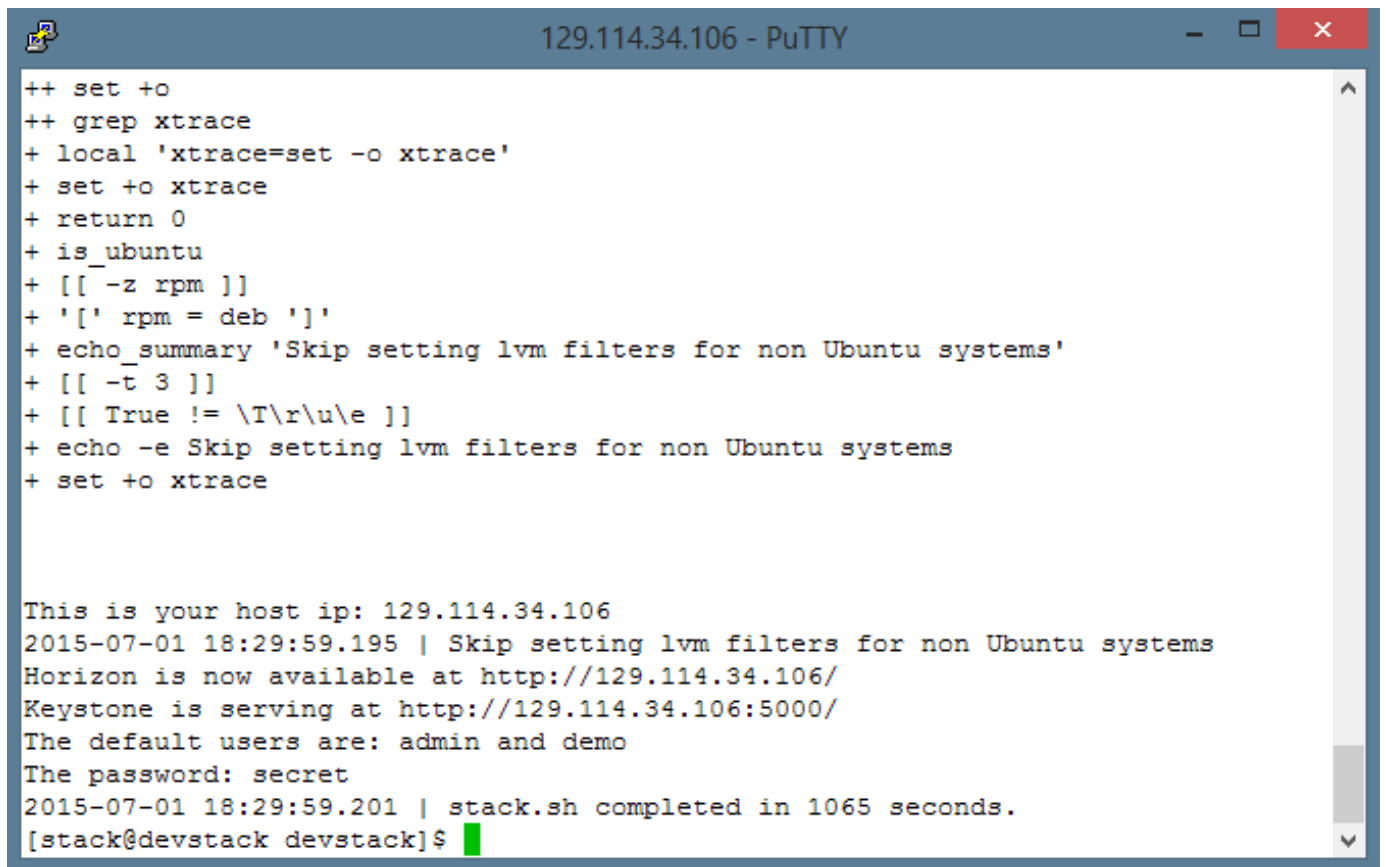
nano ~/devstack/local.conf

| local.conf |
|-----|
| [[local|localrc]] |
| FLOATING_RANGE=10.12.0.240/28 |
| FIXED_RANGE=192.168.1.0/24 |
| FIXED_NETWORK_SIZE=256 |
| FLAT_INTERFACE=eno1 |
| SERVICE_TOKEN=azertytoken |
| ADMIN_PASSWORD=secret |
| MYSQL_PASSWORD=secretdb |
| RABBIT_PASSWORD=stackqueue |
| SERVICE_PASSWORD=$ADMIN_PASSWORD |
| HOST_IP=Your.Server.IP.Address |
```

Ensure that your server's IP address is accurately reflected in the `HOST_IP` field of the configuration file. Note that this is different from the floating IP address we associated with the server. The host IP address can be seen in the Chameleon dashboard directly *above* your instance's floating IP address, and should begin with "10."

Finally, launch DevStack by executing the script: `./stack.sh`

Approximately 20 minutes later, your installation should be finished, and you should receive a message similar to the following:

A screenshot of a PuTTY terminal window titled "129.114.34.106 - PuTTY". The terminal displays the execution of a shell script. The script includes several lines of code for setting environment variables, checking the operating system (Ubuntu vs. RPM), and displaying summary information. The output shows the host IP, a timestamped log entry, the availability of Horizon at a specific URL, the Keystone service URL, default users, and the password. The script concludes with a completion message and a shell prompt.

```
++ set +o
++ grep xtrace
+ local 'xtrace=set -o xtrace'
+ set +o xtrace
+ return 0
+ is_ubuntu
+ [[ -z rpm ]]
+ '[' rpm = deb ']'
+ echo_summary 'Skip setting lvm filters for non Ubuntu systems'
+ [[ -t 3 ]]
+ [[ True != \T\r\u\e ]]
+ echo -e Skip setting lvm filters for non Ubuntu systems
+ set +o xtrace

This is your host ip: 129.114.34.106
2015-07-01 18:29:59.195 | Skip setting lvm filters for non Ubuntu systems
Horizon is now available at http://129.114.34.106/
Keystone is serving at http://129.114.34.106:5000/
The default users are: admin and demo
The password: secret
2015-07-01 18:29:59.201 | stack.sh completed in 1065 seconds.
[stack@devstack devstack]$
```

Figure 3 - Final output of stack.sh script

Your DevStack sandbox is now up and running. You can connect to it using your web browser and the “Horizon is now available at” web address displayed in the output.