Chameleon Cloud Tutorial

National Science Foundation

Program Solicitation # NSF 13-602

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

Hadoop - Sandbox

Objectives

In this tutorial, we will show you how to build a single-node Hadoop cluster on top of a bare metal Chameleon Cloud 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 Hadoop sandbox on.	5
2	Download and Configure Hadoop	Here we will configure our environment as required for Hadoop to install and run properly.	10
3	Run a Sample Map Reduce Program	Once we have have Hadoop up and running, we will execute a simple “grep-like” regular expressions matching program.	5

Prerequisites

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

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

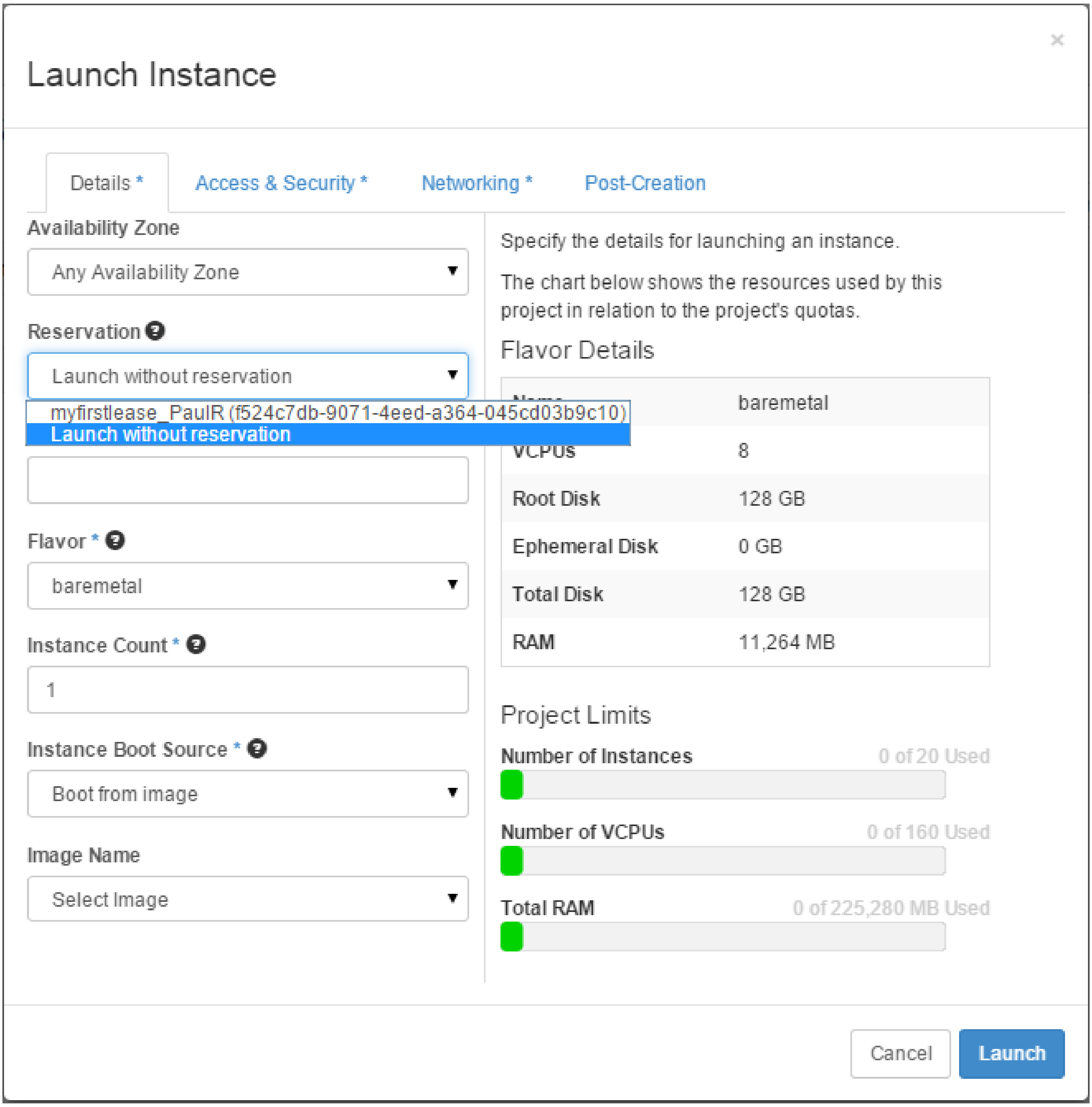
Installation Process

In this tutorial, we will be setting up a one-node cluster and running a sample application on it, producing output.

1. Create a cloud Server

Login to https://ironic.chameleon.tacc.utexas.edu/dashboard/project/instances/ and create a Chameleon Cloud Server from the 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/veNCdl) for how to create one. See figure 1 for details.

1. Instance name: **yourname**
2. Availability zone: **Any Availability Zone**
3. 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

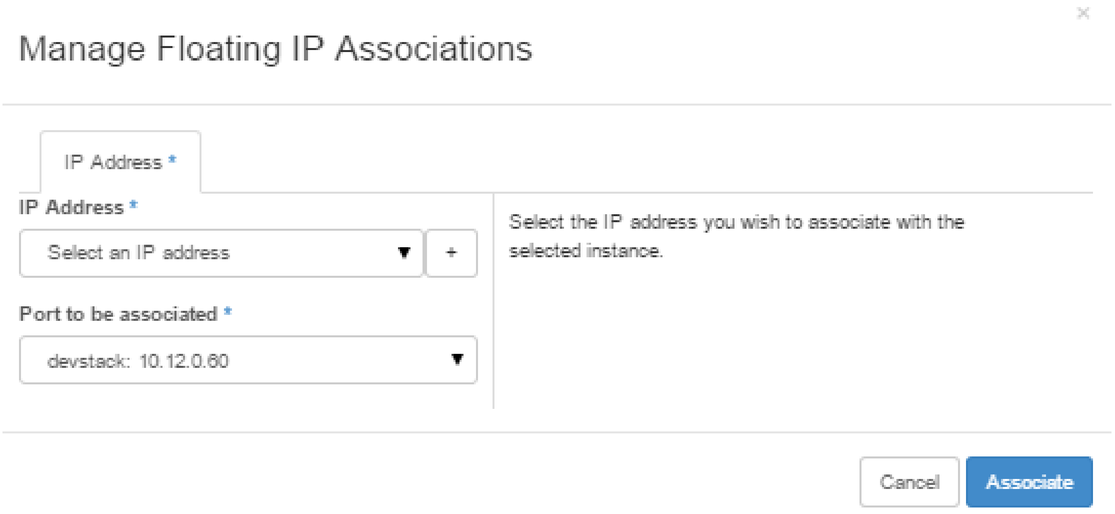
Number of Instances: 0 of 20 Used

Number of VCPUs: 0 of 160 Used

Total RAM: 0 of 225,280 MB Used

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.



Manage Floating IP Associations

IP Address *

Select the IP address you wish to associate with the selected instance.

Port to be associated *

devstack: 10.12.0.60

Figure 2 – Associate a Floating IP Address dialog box

2. Download and Configure Hadoop

SSH into the server using the floating IP address you recorded in the last step:

```
ssh cc<Your Server IP>
```

After logging in, execute the following command to download wget:

```
sudo yum install wget -y
```

When completed, download the Hadoop 2.6.0 binary tarball by executing the next command:

```
wget ftp://apache.mirrors.pair.com/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
```

Extract the contents of the archive:

```
tar -xvzf hadoop-2.6.0.tar.gz
```

We now want to setup passwordless SSH. To do this also need to generate an SSH key pair, which we can accomplish by executing:

```
ssh-keygen -t rsa
```

You will be prompted for a filename and a passphrase during this process. Leave both blank, as seen below:

```
Creating SSH key

Generating public/private rsa key pair.

Enter file in which to save the key (/home/stack/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

We will want to copy the newly created public key to the authorized keys file for this user account:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Prior to running Hadoop, we must install Java:

```
sudo yum install java-1.8.0-openjdk.x86_64 -y
```

Next, we will need to export several environment variables that Hadoop will need to run properly, by typing the following lengthy command:

```
echo export HADOOP_HOME=/home/cc/hadoop-2.6.0 HADOOP_INSTALL=$HADOOP_HOME HADOOP_MAPRED_HOME=$HADOOP_HOME HADOOP_COMMON_HOME=$HADOOP_HOME HADOOP_HDFS_HOME=$HADOOP_HOME YARN_HOME=$HADOOP_HOME HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.45-30.b13.el7_1.x86_64/jre/ >> ~/.bashrc
```

Then, we will ensure these variables remain in our local environment by reloading the .bashrc file:

```
source ~/.bashrc
```

Next, we will change directories into the extracted Hadoop folder and execute the following commands which will, respectively, format a new distributed filesystem and start the Hadoop daemons:

```
cd ~/hadoop-2.6.0
bin/hdfs namenode –format
sbin/start-dfs.sh
sbin/start-yarn.sh
```

Note: You will also be asked twice if you would like to continue connecting to a host whose authenticity cannot be established. Type “**yes**” when so prompted.

```
[ … ]
Starting namenodes on [localhost]
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 4d:e0:bc:c6:1a:f3:11:4a:84:e6:b6:65:b2:05:02:6c.
Are you sure you want to continue connecting (yes/no)? yes
```

3. Run a Sample Map Reduce Program

We will then build the input and output directories which our sample Map Reduce program will use:

```
bin/hadoop fs –put etc/hadoop input
```

Now we can run our sample Map Reduce program:

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar grep input output 'dfs[a-z.]+'
```

And finally, we can gather the results of our program:

```
bin/hadoop fs -cat output/*
```

The example program performs a word search (essentially mimicking grep) on the unpackaged Hadoop configuration directory, searching for matches for the regular expression ‘dfs[a-z.]+’. The output is then displayed on your screen.

Additionally, the server is browsable by directing your web browser to http://Your-Server-IP:50070/ where “Your-Server-IP” represents the floating IP address assigned to your instance. Various server parameters can be found here, including the input and output files related to the sample Map Reduce program.

Finally, when you’re done, you may stop the Hadoop daemons by executing:

```
sbin/stop-dfs.sh
sbin/stop-yarn.sh
```