

A simple guide to create and configure a Cloudfab's cluster.

What is Cloudfab, and why cloud computing? A Cloud platform to create and manage computer clusters. Is a useful tool for a research environment providing a simple interface and UI to use their services. The current computational trend points to a more intensive and complex situations e.g. deep-learning systems and data analytics on distributed and heterogeneous datasets. At this part of the journey, we have decided to face this computational requirements using clusters of commodity hardware working together as a software tool. This means a distributed system perspective which is a means to an end, making research proposals and publish, publish, publish.

This guide covers a basic set of technical challenges that research student will face, which are: Create a user, Login, Create an experiment, and Configure the cluster. For that keep in mind that we'll use the **OpenStack profile**, and will host our cluster in **Cloudfab's Utah Cluster**. This two concept will come handy as we continue with the tutorial.

Create a user.

Go to <https://www.cloudfab.us>, click on Request an account and create a user and once you are done, email the user to Dr. Rao, so he can add you into the Cloudfab's account.

Login and Create an experiment.

You will be able to login once Dr. Rao has grant you access to the No-SQL-JSON project, please remind him continuously. Once logged into Cloudfab, you can instantiate a computer cluster also known in Cloudfab as experiment.

1. Create an experiment by going to **Actions, Start an experiment** option located the in Cloudfab's main menu.

- Keep the OpenStack profile and click **Next**. Although you can modify the profile, BUT keeping this profile assures you that this tutorial works (we have made some development based on this profile).

New standard image now available: [UBUNTU16-64-STD](#) (Ubuntu 16.04 64-bit); also [UBUNTU16-64-ARM](#) (aarch64) [View Details](#)

Current Usage: 3350.58 Node Hours, **Prev Week: 1008**, **Prev Month: 3957** (30 day rank: 13 of 153 users)

1. Select a Profile 2. Parameterize 3. Finalize

Selected Profile: OpenStack

This profile provides a highly-configurable OpenStack instance with a controller, network manager, and one or more compute nodes (potentially at multiple Cloudlab sites). This profile runs x86 or ARM64 nodes. It sets up OpenStack Liberty, Kilo, or Juno (on Ubuntu 15.10, 15.04, or 14.10) according to your choice, and configures all OpenStack services, pulls in some VM disk images, and creates basic networks accessible via floating IPs. You'll be able to create instances and access them over the Internet in just a few minutes. When you click the Instantiate button, you'll be presented with a list of parameters that you can change to control what your OpenStack instance will look like; **carefully** read the parameter documentation on that page (or in the Instructions) to understand the various features available to you.

[Copy Profile](#) [Show Profile](#) [Change Profile](#)

[Previous](#) [Next](#)

- Indicate the number of nodes in the cluster. The number of nodes in the cluster can be modified in the field **Number of compute nodes** according to Dr. Rao's your needs. If we choose to follow OpenStack's profile we are offered with a $2 + n$ nodes cluster, where n is the number of nodes indicated in the earlier field.

Current Usage: 3350.58 Node Hours, **Prev Week: 1008**, **Prev Month: 3957** (30 day rank: 13 of 153 users)

1. Select a Profile 2. Parameterize 3. Finalize

This profile is parameterized; please make your selections below, and then click to continue.

[Show All Parameter Help](#)

OpenStack Release [Liberty](#)

Number of compute nodes (at Site 1)

Number of public IP addresses

Hardware type of all nodes

Experiment Link Speed of all nodes

[Advanced Parameters](#)

[Previous](#) [Next](#)

- Indicate cluster's name, host, and how to react to a failure (A.K.A. We are almost at the end of creating the cluster! Hang in there buddy!). Here we have to name our almost-newborn cluster, use the **Name** field for that. **IMPORTANT!** **Select Cloudlab Utah** Cluster, since forward configurations depend on this architecture. Next, **check the Ignore Failure** option, this will prevent that Cloudlab cancel your cluster because a node failed to load.

Current Usage: 3350.58 Node Hours, **Prev Week: 1008**, **Prev Month: 3957** (30 day rank: 13 of 153 users)

1. Select a Profile 2. Parameterize 3. Finalize

Profile: OpenStack Version: 32 [Source](#)

Please review the selections below and then click Finish.

Name:

Cluster:

Ignore Failures: ☒

[Check Cluster Status](#)

[Previous](#) [Finish](#)

5. Wait for the cluster to finish and get use to the the graphical interface.

urrent Usage: 3350.85 Node Hours, **Prev Week: 1008**, **Prev Month: 3957** (30 day rank: 13 of 153 users) ⓘ

Your experiment is ready!

Name: CloudLabTest

State: **ready**

Profile: **OpenStack**

Created: Aug 24, 2016 10:05 PM

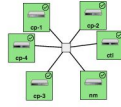
Expires: Aug 25, 2016 2:05 PM (in 16 hours)

Silver

Copy Extend Terminate

Profile Instructions

Topology View | List View | Manifest | Graphs



Click on a node for more options. Click and drag to move things around.

Run Listview Refresh Status

Configuring the cluster

Start a shell and run some code. Once your cluster has been successfully created it is time to configure it! You can take a look at the nodes in the *List View* tab. The data currently useful for us are: Node's ID, SSH command, and Actions. ID refers to the node's or how is known in the cluster. The SSH Command is the actual SSH connection that you can execute in the Terminal. Now is time to configure the cluster and the first thing to do is to login in every computer, **except ctl**, and change the user's password (**Be sure to set the same password in all the machines!**) by executing **sudo passwd <user_name>** where <user_name> is your Cloudlab's user.

ID	Node	Type	SSH command (if you provided your own key)		Actions
cp-4	ms0237	m400	ssh -p 22 d1544@ms0237.utah.cloudlab.us	<input type="checkbox"/>	
cp-2	ms0228	m400	ssh -p 22 d1544@ms0228.utah.cloudlab.us	<input type="checkbox"/>	
nm	ms0230	m400	ssh -p 22 d1544@ms0230.utah.cloudlab.us	<input type="checkbox"/>	
cp-1	ms0204	m400	ssh -p 22 d1544@ms0204.utah.cloudlab.us	<input type="checkbox"/>	
ctl	ms0220	m400	ssh -p 22 d1544@ms0220.utah.cloudlab.us	<input type="checkbox"/>	
cp-3	ms0229	m400	ssh -p 22 d1544@ms0229.utah.cloudlab.us	<input type="checkbox"/>	

Shell

Console

Console Log

Snapshot

Delete Node

This can be done in two options: a) Using Cloudlab's web interface or **b) Using SSH from your local machine**. Using option a) presents a lot of issues, for example you can not copy or paste text in the terminal (it's actually the only problem, and is annoying!). OK, now to configure the cluster. After accessing the machine named nm in your new cluster, we need to follow three simple steps:

- Download the SSHConfigure repository from Github with the command:
`git clone https://github.com/debarron/SSHConfigure.git`

- Configure the sources, in order to successfully configure Hadoop, you need to copy the public server name indicated as the second column in /etc/hosts file. The server name will follow the next convention: nm.PROJECT_NAME.GROUP_NAME.utah.cloudlab.us. **Copy that name in every machine in the cluster, three times** in ... Gotcha you fool! Go to the repository's sources file and change the masterNetworkName's value from something to the actual server name and save the file.
- Execute the script and start Hadoop. Now you need to execute the script using the following convention:
./Master.sh NN_NAME DN_NAME,START,END USER,PASSWORD.

As an example **imagine** that you have this configuration: Namenode's local name = nm, Datanodes = cp-1, cp-2, cp-3, cp-4, user = daniel, password = culiacan, Then the command will look like:

```
./Master.sh nm cp-,1,4 daniel,culiacan
```

After running the Master.sh script you will be able to login on every node in the cluster without having to specify the user or password (try it! ssh cp-1). This is about to finish! You are almost there! Now we need to start Hadoop, run these two commands in the console:

```
$HADOOP_HOME/sbin/start-all.sh
hadoop namenode -format
```

Maybe they will ask you to type yes, or something else so be patient and wait. In order to confirm If everything went well execute: jps in the terminal, you must see the next processes:

- * SecondaryNameNode
- * Jps
- * ResourceManager
- * NameNode

That's it! You are free to go! To go and code! Code! Code! *Pitbull shouting in the background*.

If ...□□□□

There was some error, and the cluster was not properly configure:

- Check the sources file, maybe there is some error with the server's public name.
- Check that all the machines share the same password.
- Check that the Master.sh command is correct.

- Ask for help, email: dl544@mail.umkc.edu