1 Branch 0 Tags Go to file t Go to file + Add file Code ···

scottgriffinm grammar 1b85808 · 11 months ago

LICENSE	Initial commit	2 years ago
README.md	grammar	11 months ago

README MIT license

Slurm for Dummies

A step-by-step guide on how to setup Slurm HPC clusters written for dummies by dummies from the 2023 University of Iowa Quantitative Finance Club under the advisory of Professor John Lewis Jr. We are by no means experts, but what is enclosed herein was learned through grueling trial and error. The primary contributers to this guide are Scott Griffin (scott-griffin@uiowa.edu) and Sergio Martelo (sergio-martelo@uiowa.edu).

Table of Contents

- Overview
- Setup Private Network and Install Ubuntu
- Setup SSH on Each Computer
- Setup Munge
- Setup Slurm
- Other Resources
- FAQ

Overview

These are the steps we followed to setup our Slurm cluster. It is important that you follow the steps in the sequence as they are written. Again, this is just what worked for us on fresh installs of Ubuntu 22.04.03 LTS.

> IMPORTANT: Steps marked with **(CONTROLLER NODE)** are just performed on your controller node and steps marked with **(WORKERS)** are just performed in your worker nodes. Steps that aren't marked are performed in both.

1. Install Ubuntu on all computers, make sure all users have the same name, configure a private network with DHCP static IP addresses, update /etc/hosts file to include all computers.
2. Setup SSH on Each Computer
3. **(CONTROLLER NODE)** Setup Munge on your controller node first.
4. **(WORKER NODES)** Setup Munge on each of the worker nodes.
5. Setup Slurm on all machines. Make sure to follow the controller node instructions for the controller node and the worker node instructions for the worker nodes.

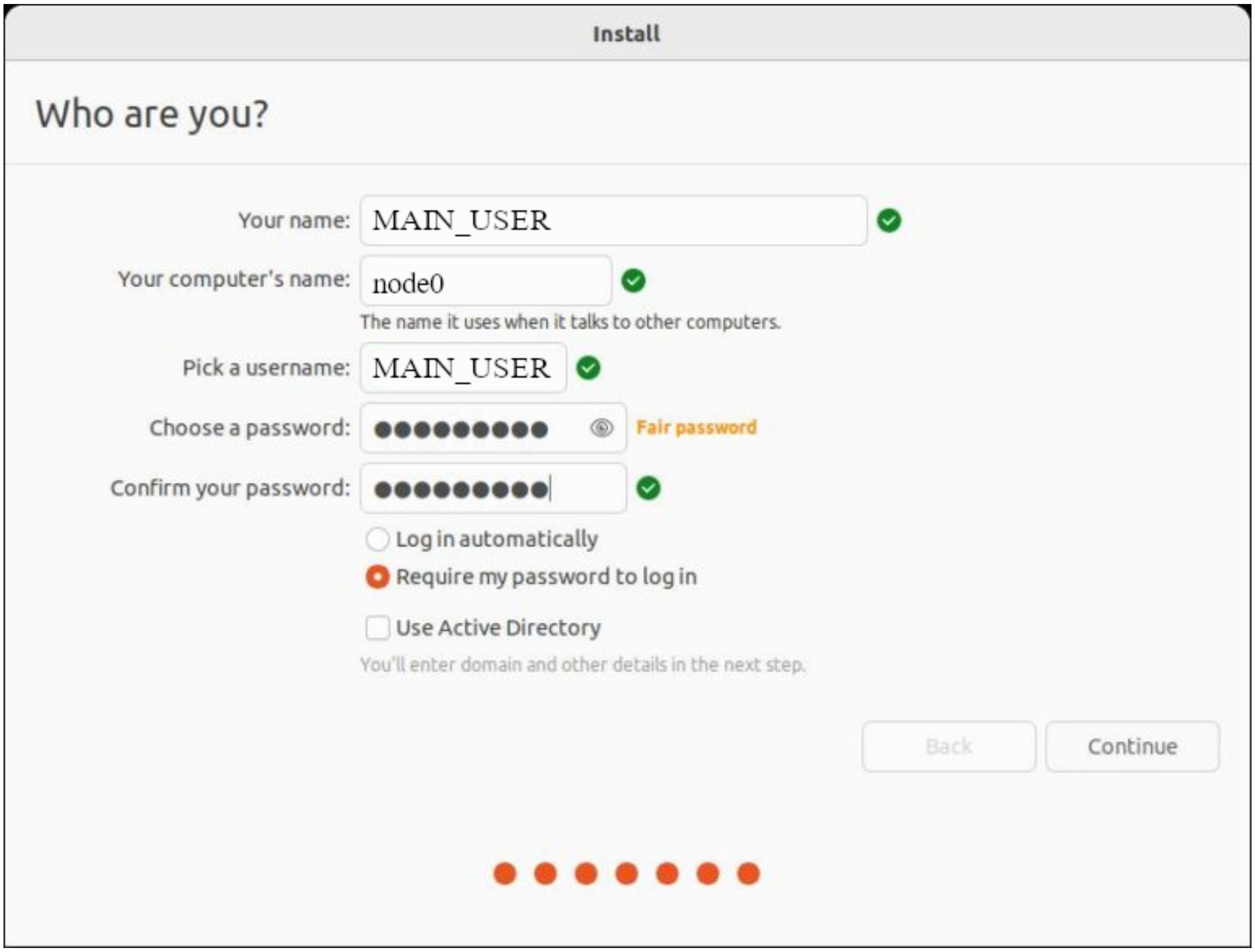
node and the worker node instructions for the worker nodes.

Setup Private Network and Install Ubuntu

Install Ubuntu 22.04 on all computers in the cluster.

> We recommend you turn off any sort of inactivity shutdown timer on all computers.

Make sure the first user's name on each computer is the same. We'll will call this user MAIN_USER from now on. Each computer will be called node0, node1, etc.



Install

Who are you?

Your name: MAIN_USER

Your computer's name: node0

The name it uses when it talks to other computers.

Pick a username: MAIN_USER

Choose a password: ●●●●●●●●●● Fair password

Confirm your password: ●●●●●●●●●●

○ Log in automatically

● Require my password to log in

□ Use Active Directory

You'll enter domain and other details in the next step.

Create a private network and update your router's DHCP static IP settings, manually entering each computer's MAC address with their IP respective address. Make sure that all computers on the cluster have each other in their known hosts file. This file can be found at `/etc/hosts`. To add a known host to the file, you have to add the hosts IP address and the hosts alias separated by a space on a newline in the file. Our `/etc/hosts` file looked something like this:

```
127.0.0.1 localhost
XXX.XXX.XX.XX0  node0
XXX.XXX.XX.XX1  node1
XXX.XXX.XX.XX2  node2
XXX.XXX.XX.XX3  node3
XXX.XXX.XX.XX4  node4
```

> Note that the Xs here stand for numbers in our IP addresses. The aliases (node0, node1, etc.) are also arbitrary, you can name your nodes whatever you like. There will likely be other networking

> arbitrary, you can name your nodes whatever you like. There will likely be other networking configurations in this file, leave them unchanged.

Run the following commands in your shell on each computer to update and upgrade all packages in that system.

```
$ sudo apt update
$ sudo apt upgrade
```

Setup SSH

Seting up SSH is pretty simple. You just run the following command:

```
$ sudo apt install openssh-server openssh-client
```

Then, to test whether it was installed correctly, we can attempt to SSH into another computer in the cluster like so:

```
$ ssh <hostname>                         (<hostname> would be the alias of another
cluster. For example, node1 if you're on node0)
```

If SSH is succesful, you should know be in a remote shell connected to the host with name `<hostname>` . If you want to learn more about SSH, visit this page.

Remember to do this on each computer.

Setup Munge

Installing Munge is pretty straightforward once you figure out what you're doing. However, the one thing that can get tricky is the file permissions, so make sure you follow the steps in order. Also, we recommend configuring the controller node first and then configuring the worker nodes.

Controller node

First, run the following command to install the munge packages.

```
$ sudo apt install munge libmunge2 libmunge-dev
```

This should install successfully as long as you're connected to the internet. To test your installation, you can run the following command:

```
$ munge -n | unmunge | grep STATUS
```

You should see something like `STATUS: SUCCESS` . Now, you have Munge correctly installed and there should be a Munge key at `/etc/munge/munge.key'. If you don't see one, then you should be able to create one manually by running the following command:

```
$ sudo /usr/sbin/mungekey
$ sudo /usr/sbin/mungekey
```

> With future updates of munge, slurm, and ubuntu, specific file locations may change.

Now, we have to ensure all of the munge files have the correct permissions. This just entails giving the munge user ownership over all the munge files. You don't have to create the munge user manually since it should have been created by munge when we installed the packages above. In fact, we recommend saving yourself the trouble and not creating the user yourself. We had a lot of troubles stem from trying to create it ourselves.

To Setup the correct permissions, use the following commands:

```
$ sudo chown -R munge: /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/
$ sudo chmod 0700 /etc/munge/ /var/log/munge/ /var/lib/munge/
$ sudo chmod 0755 /run/munge/
$ sudo chmod 0700 /etc/munge/munge.key
$ sudo chown -R munge: /etc/munge/munge.key
```

Next, we need to restart the munge service and configure it to run at startup. We do that like so:

```
$ systemctl enable munge
$ systemctl restart munge
```

You can investigate munge service errors with:

```
$ systemctl status munge
```

Or

```
$ sudo nano /var/log/munge/munged.log
```

That's it! Now, you can go ahead and Setup your worker nodes. Also, for convenience you can now save your `munge.key` located at `/etc/munge/' to an easily accessible location. You will need to copy that key over to the other nodes in the cluster when setting them up. We go over that in detail next.

Worker nodes

For each worker node we follow the same procedure. Similar to the controller node, you first install munge, like so:

```
$ sudo apt install munge libmunge2 libmunge-dev
```

We check if munge is installed correctly, like so:

```
$ munge -n | unmunge | grep STATUS
```

Again, you should see something like `STATUS: SUCCESS` . Now, munge is correctly installed on this node, however we still need to copy our controller node's key to this node. To do that, simply replace the worker node's `munge.key` file located at `/etc/munge/' with the controller node's` munge.key` file. The most straightforward way we found to do this was to put the controller node's 'munge.key' onto a USB drive and then plug the USB drive into the worker node.

Once you have swapped out `munge.key`, we need to make sure munge's permissions are correct on this worker node. We do that like so:

```
$ sudo chown -R munge: /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/
$ sudo chmod 0700 /etc/munge/ /var/log/munge/ /var/lib/munge/
$ sudo chmod 0755 /run/munge/
$ sudo chmod 0700 /etc/munge/munge.key
$ sudo chown -R munge: /etc/munge/munge.key
```

Next, we start the munge service and configure it to start at startup.

```
$ systemctl enable munge
$ systemctl restart munge
```

Again, you can investigate munge service errors with:

```
$ systemctl status munge
```

Or

```
$ sudo nano /var/log/munge/munged.log
```

Now, we can test the munge connection to the controller node, like so:

```
$ munge -n | ssh <CONTROLLER_NODE> unmunge
```

Make sure to replace `<CONTROLLER_NODE>` with host alias of your controller node. If this is successful, you should see the munge status of the controller node. If you get an error, try restarting the munge service on the controller node.

Setup Slurm

The process to install and Setup Slurm is almost the same in the controller node and the worker nodes. The only significant difference is which service we have to start and enable. First, on all nodes, install the required packages with:

```
$ sudo apt install slurm-wlm
```

Controller node

To configure Slurm on your controller node do the following.

Use slurm's handy configuration file generator located at `/usr/share/doc/slurmctld/slurm-wlm-configurator.html` to create your configuration file. You can open the configurator file with your browser `configurator.html` to create your configuration file. You can open the configurator file with your browser.

> Slurm configuration files are a complicated topic and what values you have to fill in is specific to your machines. If you want to learn more about it, go here.

You don't have to fill out all of the fields in the configuration tool since a lot of them can be left to their defaults. The following fields are the once we had to manually configure:

- ClusterName: `<YOUR-CLUSTER-NAME>`
- SlurmctldHost: `<CONTROLLER-NODE-NAME>`
- NodeName: `<WORKER-NODE-NAME>` [1-4] (this would mean that you have four worker nodes called `<WORKER-NODE-NAME>1` , `<WORKER-NODE-NAME>2` , `<WORKER-NODE-NAME>3` , `<WORKER-NODE-NAME>4`)
- Enter values for CPUs, Sockets, CoresPerSocket, and ThreadsPerCore according to $ lscpu (run on a worker node computer)
- ProctrackType: LinuxProc

Once you press the `submit` button at the bottom of the configuration tool your configuration file text will appear in your browser. Copy this text into a new /etc/slurm/slurm.conf file and save.

```
$ sudo nano /etc/slurm/slurm.conf
```

At this point you should copy the text from your created slurm.conf to each worker node's /etc/slurm/slurm.conf. We found the best way to do this was to copy our created slurm.conf file to a thumbdrive, then use the previous command on each worker node to create the slurm.conf file and then copy the text from our thumbdrive slurm.conf and save.

Now, we have to start the slurm controller node service and configure it to start at startup, like so:

```
$ systemctl enable slurmctld
$ systemctl restart slurmctld
```

You can now check your slurm installation is runnning and your cluster is Setup with the following commands:

```
$ systemctl status slurmctld # returns status of slurm service
$ sinfo         # returns cluster information
```

Once you have your worker nodes Setup, you can also check the cluster is correctly Setup by running:

```
$ srun hostname
```

Where `<NUMBER-OF-NODES>` is the number of worker nodes that are currently Setup. If you followed all of the steps correctly, this should return the name of all of your nodes.

Worker nodes

We follow a similar procedure to the controller node for each worker node. Be sure to copy the text from your created slurm.conf to each worker node's /etc/slurm/slurm.conf. We found the best way to do this was to copy our created slurm.conf file to a thumbdrive, then use the following command on each worker node to create the slurm.conf file and then copy the text from our thumbdrive slurm.conf and save.

```
$ sudo nano /etc/slurm/slurm.conf
```

Now, we start the slurm worker node service and configure it to start at startup.

```
$ systemctl enable slurmd
$ systemctl restart slurmd
```

Then, we can verify slurm is Setup correctly and running like so:

```
$ systemctl status slurmd
```

As long as you got no errors, your slurm worker node should now be setup. You can check that it is running correctly by using the `sinfo` or `srun` commands on your controller node.

```
$ srun hostname
```

You can investigate errors in more detail by looking in the slurm log file:

```
$ sudo nano /var/log/slurm.slurmd.log
```

Other Resources

These are some resources we found helpful along the way.

- Munge docs by Chris Dunlap
- Slurm docs from SchedMD
- Great blog we used to help Setup our Slurm cluster by Bodun Hu

FAQ

What is Slurm?

Slurm is a cluster managament and job scheduling system for Linux clusters. It has very extensive documentation that can be found here.

Why did we write this?

We are a group of students from the University of Iowa Quant Finance Club who struggled for weeks with setting up a Slurm cluster. We made every mistake in the book and looked everywhere for guides on how to setup clusters, but the guides we found were either going above our heads or missing critical information. So, we decided to document our process and put it on the web and, hopefully, it'll be able to help other students/practitioners Setup HPC clusters.

Packages

No packages published

Releases

No releases published

Contributors 2

scottgriffinm Scott Griffin

SergioMEV Sergio Martelo