# How to quickly set up Slurm on Ubuntu 20.04 for single node workload scheduling.

https://drtailor.medium.com/how-to-setup-slurm-on-ubuntu-20-04-for-single-node-work-scheduling-6cc909574365

Slurm is an excellent work scheduling tool for High-Performance computing clusters. In addition, it can be an invaluable tool on a local desktop or single server when you need to run several programs at once and queue them up whilst ensuring you don’t overload your computer or server. Furthermore, it can be useful in cases where you share a server with other users or need to run multiple jobs overnight or for weeks! Here I show you how to quickly set up slurm on a single machine with ubuntu 20.04. You will no longer need to make mangled scripts to run multiple programs to avoid going over your hardware limits or dispute with colleagues whose program gets to run first.

Prerequisites

* Basic Linux CLI
* An Ubuntu machine with internet access.

Let us get it installed first with apt, for a basic single machine setup, the only packages needed areslurmctld the control daemon and slurmd the compute node daemon:

$ sudo apt update -y

$ sudo apt install slurmd slurmctld -y

Next, we need to create the slurm.conf file which configures how your slurm queue is set up. Here we use a very simple one: (please adjust the COMPUTE NODES section to your machines specs. e.g. if you have10 cores CPUs=10 and your memory is 32000MB RealMemory=32000 .

$ sudo chmod 777 /etc/slurm-llnl$ sudo cat << EOF > /etc/slurm-llnl/slurm.conf

# slurm.conf file generated by configurator.html.

# Put this file on all nodes of your cluster.

# See the slurm.conf man page for more information.

#

ClusterName=localcluster

SlurmctldHost=localhost

MpiDefault=none

ProctrackType=proctrack/linuxproc

ReturnToService=2

SlurmctldPidFile=/var/run/slurmctld.pid

SlurmctldPort=6817

SlurmdPidFile=/var/run/slurmd.pid

SlurmdPort=6818

SlurmdSpoolDir=/var/lib/slurm-llnl/slurmd

SlurmUser=slurm

StateSaveLocation=/var/lib/slurm-llnl/slurmctld

SwitchType=switch/none

TaskPlugin=task/none

#

# TIMERS

InactiveLimit=0

KillWait=30

MinJobAge=300

SlurmctldTimeout=120

SlurmdTimeout=300

Waittime=0

# SCHEDULING

SchedulerType=sched/backfill

SelectType=select/cons\_tres

SelectTypeParameters=CR\_Core

#

#AccountingStoragePort=

AccountingStorageType=accounting\_storage/none

JobCompType=jobcomp/none

JobAcctGatherFrequency=30

JobAcctGatherType=jobacct\_gather/none

SlurmctldDebug=info

SlurmctldLogFile=/var/log/slurm-llnl/slurmctld.log

SlurmdDebug=info

SlurmdLogFile=/var/log/slurm-llnl/slurmd.log

#

# COMPUTE NODES

NodeName=localhost CPUs=1 RealMemory=500 State=UNKNOWN

PartitionName=LocalQ Nodes=ALL Default=YES MaxTime=INFINITE State=UP

EOF$ sudo chmod 755 /etc/slurm-llnl/

Now lets gets slurm started with systemd:

$ sudo systemctl start slurmctld

$ sudo systemctl start slurmd

Lastly, let's set our machine as idle, so we can start queuing up jobs:

$ sudo scontrol update nodename=localhost state=idle

$ sinfo

PARTITION AVAIL TIMELIMIT NODES STATE NODELIST

LocalQ\* up infinite 1 idle localhost

If successful you see the above and well done, you have got slurm up and running. You now have a queue(or “partition” in slurm lingo) called LocalQ that you can now submit your work to. If you have any issues you can debug it by looking in the logs in /var/log/slurm-llnl/slurmd.log and /var/log/slurm-llnl/slurmctld.log.

Now you have a working slurm queue, if you need to make changes to your config edit the slurm.conf and simply restart slurmctld and slurmd via systemd. For more information about how to use slurm, there are lots of articles online. Just google “how to submit jobs to slurm” and also check out the slurm website. Happy computing!

See [https://slurm.schedmd.com](https://slurm.schedmd.com/) for more information.

# Set up Slurm across Multiple Machines

To install [Slurm](https://slurm.schedmd.com/documentation.html), we need to have admin access to the machine. This post explains how I got Slurm running in multiple Linux servers. All servers are running on Ubuntu 18.04 LTS.

## Setup Munge

First, we need to make sure the clocks, users and groups (UIDs and GIDs) are synchronized across the cluster. We need to create two users: slurm and munge across all servers.

Then, we install [Munge](https://linux.die.net/man/7/munge) for authentication:

$ apt install munge libmunge2 libmunge-dev

To test if munge is installed successfully:

$ munge -n | unmunge | grep STATUS

STATUS: Success (0)

Next, we create a munge authentication key on one of the servers:

$ /usr/sbin/create-munge-key

After we generate munge authentication key, we copy the key /etc/munge/munge.key on that server to all other servers (overwrite the /etc/munge/munge.key on all other servers).

We need to setup the rights for munge accordingly on every server:

$ chown -R munge: /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/

$ chmod 0700 /etc/munge/ /var/log/munge/ /var/lib/munge/

$ chmod 0755 /run/munge/

Then, we enable and start the munge service with (remember to not use sudo when running munge):

$ systemctl enable munge

$ systemctl start munge

You can then test whether munge works properly by executing:

munge -n # Generate a credential on stdout

munge -n | unmunge # Displays information about the MUNGE key

munge -n | ssh somehost unmunge

If everything is setup properly, you shouldn’t see any error messages.

## Setup Slurm

Use apt to install slurm in Ubuntu systems (make sure all nodes have the same slurm versions):

$ apt install slurm-wlm

Next, we need to configure slurm. Since we used package manager to install slurm, the version is lower than the latest release. Thus, it’s preferably to not use the official [Slurm Configuration Tool](https://slurm.schedmd.com/configurator.html). Instead, we can find the corresponding version’s configuration tool at /usr/share/doc/slurmctld/slurm-wlm-configurator.html.

After filling up the required fields in the form, we copy the generated file into /etc/slurm-llnl/slurm.conf on all nodes. Then, you can execute sinfo to check all nodes status. You can also launch jobs to see if it actually works, for example:

srun -N2 -l /bin/hostname

This should print out the hostname for all the nodes in the cluster.

## Add GPU support

To add GPU support, we first create a file gres.conf in /etc/slurm-llnl/. Here is an example on one node:

Name=gpu File=/dev/nvidia0

Name=gpu File=/dev/nvidia1

Name=gpu File=/dev/nvidia2

Then, we add GresTypes=gpu into /etc/slurm-llnl/slurm.conf. Next, we add the GPU information to slurm.conf:

NodeName=node1 Gres=gpu:3 State=UNKNOWN

# Slurm Quick Installation for Cluster on Ubuntu 20.04 很实用！

https://nekodaemon.com/2022/09/02/Slurm-Quick-Installation-for-Cluster-on-Ubuntu-20-04/

## Naming Convention of Nodes

A common cluster should comprise management nodes and compute nodes. This aritcle will take our cluster as an example to demostrate steps to install and configure Slurm. In our case, the management node is called clab-mgt01 while the compute nodes are named from clab01 to clab20 in order.

## Install Dependencies

Execute the following command to install the dependencies on all machines. (clab-all refers to all machines including management and compute nodes).

|  |
| --- |
| clab-all$ sudo apt install slurm-wlm slurm-client munge |

Tips: There are several tools that may help to manage multiple nodes easily:

* iTerm2 (on Mac) / Terminator (on Linux)
* csshX (on Mac) / cssh (on Linux)
* Parallel SSH (at cluster side)

## Generate Slurm Configuration

There is [an official online configuration generator](https://slurm.schedmd.com/configurator.html). And we should carefully check the fields below.

* SlurmctldHost: clab-mgt01 in our case.
* NodeName: clab[01-20] in our case.
* CPUs: It is recommended to leave it blank.
* Sockets: For a dual-socket server we commonly see, it should be 2.
* CoresPerSocket: Number of physical cores per socket.
* ThreadsPerCore: For a regular x86 server, if hyperthreading is enabled, it should be 2, otherwise 1.
* RealMemory: Optional.

Click submit, then we could copy the file content to /etc/slurm-llnl/slurm.conf on all machines.

Tips: Don't forget the shared storage (e.g. NFS storage) on the cluster. We could utilize it to distribute files.

## Distribute Munge Key

Once Munge is installed successfully, the key /etc/munge/munge.key will be automatically generated. It is requried for all machines to hold the same key. Therefore, we could distribute the key on the management node to the remaining nodes including compute nodes and other backup management node if existing.

Tips: Again. We could also utilize the shared storage to distribute the key.

Then make sure the permission and the ownership are correctly set.

|  |
| --- |
| clab-all$ sudo chmod 400 /etc/munge/munge.key  clab-all$ chown munge:munge /etc/munge/munge.key |

## Patch Slurm Cgroup Integration

By default, there Slurm cannot work with Cgroup well. If we start Slurm service right now, we may receive this error shown below.

|  |
| --- |
| error: cgroup namespace 'freezer' not mounted. aborting |

Therefore, by pasting the following content to /etc/slurm/cgroup.conf on compute nodes, this issue can be fixed.

|  |
| --- |
| CgroupMountpoint=/sys/fs/cgroup |

or using this command:

|  |
| --- |
| echo CgroupMountpoint=/sys/fs/cgroup >> /etc/slurm/cgroup.conf |

## Fix Directory Permission

For unknown reasons, the permission of the relevant directory is not set properly, which may lead to this error.

|  |
| --- |
| slurmctld: fatal: mkdir(/var/spool/slurmctld): Permission denied |

The solution is executing the commands below on management nodes.

|  |
| --- |
| clab-mgt$ sudo mkdir -p /var/spool/slurmctld  clab-mgt$ sudo chown slurm:slurm /var/spool/slurmctld/ |

## Start Slurm Service

So far, we have finished the basic configuration. Let us launch Slurm now.

|  |
| --- |
| # On management nodes  clab-mgt$ sudo systemctl enable munge  clab-mgt$ sudo systemctl start munge  clab-mgt$ sudo systemctl enable slurmctld  clab-mgt$ sudo systemctl start slurmctld  # On compute nodes  clab-comp$ sudo systemctl enable munge  clab-comp$ sudo systemctl start munge  clab-comp$ sudo systemctl enable slurmd  clab-comp$ sudo systemctl start slurmd |

Run sinfo and we should see all the compute nodes are ready.

|  |
| --- |
| $ sinfo  PARTITION AVAIL TIMELIMIT NODES STATE NODELIST  debug\* up infinite 20 idle clab[01-20] |

## Debugging Tips

If your Slurm is not working correctly, you could try with these commands to debug.

|  |
| --- |
| clab-mgt$ sudo slurmctld -D  clab-comp$ sudo slurmd -D |

## References

* <https://www.cnblogs.com/aobaxu/p/16195237.html>
* <https://stackoverflow.com/questions/62641323/error-cgroup-namespace-freezer-not-mounted-aborting>
* Author: NekoDaemon
* Link: <https://nekodaemon.com/2022/09/02/Slurm-Quick-Installation-for-Cluster-on-Ubuntu-20-04/>
* Copyright: Original content on this site is licensed under [BY-SA](https://creativecommons.org/licenses/by-sa/4.0/)