

CSH Compute Cluster Node Setup Guide

Zach Hart

July 3, 2017

1 Things to know before you start

So you want to add (or replace) a node to the existing cluster. This should be fairly straightforward, as it's been done a bunch and we've encountered pretty much every possible roadblock that you might encounter. That being said, if you followed this guide and something went wrong, re-read the guide and make sure you didn't miss something. A more condensed version of this guide can be found at the bottom of the [CSH Wiki Entry](#) on the cluster. If you're still having troubles, check out the [SchedMD Troubleshooting Guide](#).

1.1 Unusual Setup Constraints

If the new node is in IH, OpComm should have already let you know of a few network environment constraints you'll need to deal with. The most obvious of which is that you will only be able to access this node from within RIT's Network. For a CSHer, this presents no real challenge - either use the CSH or RIT VPN, or simply use another machine as a bridge into the network (for example, `ssh` into `shell.csh.rit.edu` and then into the node you want to access).

2 Basic Setup Tasks

2.1 Node Provisioning

Currently, our nodes have been provisioned for us by OpComm, and are all LinkedIn Nodes (2 sockets, 12 Cores per socket, 48 Gigabytes of RAM) running RHEL 7. To keep things in the guide simple, we're going to assume the node you're trying to add is identical. If your node is different, note that many things in this guide will need to be tailored. At the minimum, your node should be running RHEL/CentOS, otherwise our setup scripts will fail.

2.2 Initial Setup

When you first get into a node, create the user accounts that you'll want and make sure those users are in the `wheel` group so that your user has access to `sudo`. While it's not completely necessary to have a user account other than root (and the `slurm` and `munge` users that will be created for you), it's generally good practice to keep one around. The next thing that should be done is to update all of the packages on the machine - **however**, you don't need to do this yourself. Running `wget http://sega.csh.rit.edu/compute-node.sh && chmod 755 compute-node.sh` will download and set the permissions on the setup script that we've created, allowing you to execute the script. The script does still need root access, so as root, run `./compute-node.sh` - the script doesn't take too long to run, so sit back and watch for any errors. A quick breakdown of what the script does is as follows

- Update system packages
- Adds `epel-release` to the system
- Installs some desired packages `git`, `vim`, `htop`, etc.
- Grabs RIT's `slurm` repo and CSH's configuration files
- Installs & configures `slurm` and `MUNGE`
- Starts and enables the `slurm` daemon

Keep in mind, that for a more seamless update of the cluster, you should update the `slurm.conf` with the system details of the new node *before* distributing it to the new system. To re-distribute this file to all existing nodes in the cluster, run `slurm-update.py` found in `/opt/slurm-update` on the master node. *Keep in mind* that this file will also need to be updated with the name of the new node - although it can be added *after* you've run it if you've updated `slurm.conf` *before* running `compute-node.sh` on the new node.

2.3 Cleaning up

If you've run `compute-node.sh` without any problems, and `slurmd` is up and running, you can go ahead and remove the `compute-node.sh` script from the machine. All the files that were downloaded by this script (that weren't subsequently moved to their intended location) were placed in `/tmp`, so you don't need to worry about them.

2.4 Networking

You'll likely need to open a port in your firewall for `slurmd` to communicate with `slurmctld` on the master node. To accomplish this, simply run `firewall-cmd --permanent --zone=public --add-port=6818/tcp && firewall-cmd --reload` as root on the new compute node.

2.5 Updating the Cluster

At this point, your new node should be good to go, but the cluster doesn't really have any idea that it exists. To "update" the cluster so-to-speak, on any node run `scontrol reconfigure` as root. At this point, running `sinfo` from any node should give you a brief message about the state of the cluster. Ideally, if no jobs are being run, your new node will be listed alongside the others as `idle`. If your machine has any other state, or the state has an asterisk appended to it, there are some issues - check out the troubleshooting section to see if we know how to fix it.

3 Troubleshooting

This should probably be the first place you look if you're having issues setting up a node - issues with the cluster as a whole are better Googled, or solved by a look through the official troubleshooting guide linked at the top of this guide. The issues here and their solutions are going to be generally specific to CSH's specific setup.

3.1 Common Errors

The `compute-node.sh` setup script fails

There are a couple reasons this might happen, paying attention as the script runs to determine where things might fail is the best way to get a head start on diagnosing this issue. If there are issues with `wget` grabbing files from the master node, make sure that `httpd` is running on `sega.csh.rit.edu`. If there are issues downloading packages from `epel-release` (namely, `fail2ban`), ask OpComm to make sure that the proxy settings for yum were set up correctly. Remember, this script must be run as root in order to work.

`slurmd` is reporting issues with munge or munged

Run `journalctl -xe` or check the slurm logs at `/var/log/slurmd.log`, if the issues appear to do with file ownership, read thoroughly through the [MUNGE setup guide](#). We use the default file permissions, munge files should be owned by the munge user. Once you've confirmed the file permissions, check to see if `munged` is running - If it is, kill the process. Restart the daemon by running `sudo -u munge /usr/sbin/munged`, `munged` must be running as the munge user. Restart the slurm daemon by running `systemctl restart slurmd` as root, and check the logs to see if things are okay now.

A new node can't communicate with the cluster

First, try **pinging** the master node - this will test if there are general networking problems between you and the rest of the cluster. If you're able to send packets back and forth, the issue lies somewhere else. **Make sure that you've opened the correct port in your firewall.** Next, run **ntpdate ntp.rit.edu** - all of the nodes should be synced to RIT's NTP server - keeping the clocks of the nodes as close together as possible is important. If your machine jumps in either direction by more than a fraction of a second, run **scontrol reconfigure** followed by **sinfo** and see if you're able to communicate again. Keep in mind that any state that ends in an asterisk (e.g. **idle***, **down***) means that the master **cannot** communicate properly with the node, and the state that it's assigned to the node is an assumption.

It might be worth it to try and run the command **scontrol update nodename=<nodename> state=resume** to try and force the cluster to assign a new state to the node. If the cluster is able to establish a connection, your machine will change state. If the state is not **idle**, run **scontrol show node** on the node that isn't communicating to see what the status of the node is. If the status changes to any assumed status, it's likely that whatever you've tried hasn't worked - you'll need to try something else. Try running **scontrol ping** from the node that is having problems - If the node believes that the master is down, check **slurm.conf** to make sure that you have the right settings for the master node. If it shows the master as UP, it's time to check the slurmd logs. Logs can be found at **/var/log/slurmd.log**, running **journalctl -xe** and **systemctl status slurmd** may also prove helpful.