Install-Slurm / README.md

Artlands Merge branch 'master' of https://github.com/Artlands/Install-Slurm

ab43abb · 5 years ago

358 lines (261 loc) · 7.54 KB

Preview Code Blame Raw

Introduction

Install Slurm on CentOS-7 Virtual Cluster.

Preparation

1. Connect Virtual Machines
2. Setup NFS Server

Cluster Server and Computing Nodes

List of master node and computing nodes within the cluster.

Hostname	IP Addr
master	10.0.1.5
node1	10.0.1.6
node2	10.0.1.7

(Optional) Delete failed installation of Slurm

Remove database:

```
yum remove mariadb-server mariadb-devel -y
```

Remove Slurm and Munge:

```
yum remove slurm munge munge-libs munge-devel -y
```

Delete the users and corresponding folders:

```
userdel -r slurm
suerdel -r munge
```

Create the global users

Slurm and Munge require consistent UID and GID across every node in the cluster. For all the nodes, before you install Slurm or Munge:

```
export MUNGEUSER=991
groupadd -g $MUNGEUSER munge
useradd  -m -c "MUNGE Uid 'N' Gid Emporium" -d /var/lib/munge -u
$MUNGEUSER -g munge  -s /sbin/nologin munge
export SLURMUSER=992
groupadd -g $SLURMUSER slurm
useradd  -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER
-g slurm  -s /bin/bash slurm
```

Install Munge

Get the latest REPL repository:

```
yum install epel-release -y
```

Install Munge:

```
yum install munge munge-libs munge-devel -y
```

Create a secret key on **master** node. First install rig-tools to properly create the key:

```
yum install rng-tools -y
rngd -r /dev/urandom
/usr/sbin/create-munge-key -r
dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key
chown munge: /etc/munge/munge.key
chmod 400 /etc/munge/munge.key
```

Send this key to all of the compute nodes:

```
scp /etc/munge/munge.key root@10.0.1.6:/etc/munge
scp /etc/munge/munge.key root@10.0.1.7:/etc/munge
```

SSH into every node and correct the permissions as well as start the Munge service:

```
chown -R munge: /etc/munge/ /var/log/munge/
chmod 0700 /etc/munge/ /var/log/munge/
```

```
systemctl enable munge
systemctl start munge
```

To test Munge, try to access another node with Munge from **master** node:

```
munge -n
munge -n | munge
munge -n | ssh 10.0.1.6 unmunge
remunge
```

If you encounter no errors, then Munge is working as expected.

Install Slurm

Install a few dependencies:

```
yum install openssl openssl-devel pam-devel numactl numactl-devel hwloc
hwloc-devel lua lua-devel readline-devel rrdtool-devel ncurses-devel
man2html libibmad libibumad -y
```

Download the latest version of Slurm in the shared folder:

```
cd /nfsshare
wget https://download.schedmd.com/slurm/slurm-19.05.4.tar.bz2
```

If you don't have `rpmbuild` yet:

```
yum install rpm-build
rpmbuild -ta slurm-19.05.4.tar.bz2
```

Check the rpms created by `rpmbuild` :

```
cd /root/rpmbuild/RPMS/x86_64
```

Move the Slurm rpms for installation for all nodes:

```
mkdir /nfsshare/slurm-rpms
cp * /nfsshare/slurm-rpms
```

On every node, install these rpms:

```
yum --nogpgcheck localinstall * -y
```

On the **master** node:

```
vim /etc/slurm/slurm.conf
```

Paste the slurm.conf in Configs and paste it into `slurm.conf` .

Notice: we manually add lines under #COMPUTE NODES.

```
NodeName=node1 NodeAddr=10.0.1.6 CPUs=1 State=UNKNOWN
NodeName=node2 NodeAddr=10.0.1.7 CPUs=1 State=UNKNOWN
```

Now the **master** node has the slurm.conf correctly, we need to send this file to the other compute nodes:

```
scp /etc/slurm/slurm.conf root@10.0.1.6:/etc/slurm/
scp /etc/slurm/slurm.conf root@10.0.1.7:/etc/slurm/
```

On the **master** node, make sure that the **master** has all the right configurations and files:

```
mkdir /var/spool/slurm
chown slurm: /var/spool/slurm/
chmod 755 /var/spool/slurm/
touch /var/log/slurmctld.log
chown slurm: /var/log/slurmctld.log
touch /var/log/slurm_jobacct.log /var/log/slurm/slurm_jobcomp.log
chown slurm: /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
```

On the computing nodes **node[1-2]**, make sure that all the computing nodes have the right configurations and files:

```
mkdir /var/spool/slurm
chown slurm: /var/spool/slurm
chmod 755 /var/spool/slurm
touch /var/log/slurm/slurmd.log
chown slurm: /var/log/slurm/slurmd.log
```

Use the following command to make sure that `slurmd` is configured properly:

```
slurmd -C
```

You should get something like this:

```
NodeName=node1 CPUs=4 Boards=1 SocketsPerBoard=1 CoresPerSocket=4
ThreadsPerCore=1 RealMemory=990 UpTime=0-07:45:41
```

Disable the firewall on the computing nodes **node[1-2]**:

```
systemctl stop firewalld
systemctl disable firewalld
```

On the **master** node, open the default ports that Slurm uses:

```
firewall-cmd --permanent --zone=public --add-port=6817/udp
firewall-cmd --permanent --zone=public --add-port=6817/tcp
firewall-cmd --permanent --zone=public --add-port=6818/udp
firewall-cmd --permanent --zone=public --add-port=6818/tcp
firewall-cmd --permanent --zone=public --add-port=6819/udp
firewall-cmd --permanent --zone=public --add-port=6819/tcp
firewall-cmd --reload
```

If the port freeing does not work, stop the firewall for testing.

Sync clocks on the cluster. On every node:

```
yum install ntp -y
chkconfig ntpd on
ntpdate pool.ntp.org
systemctl start ntpd
```

On the computing nodes **node[1-2]**:

```
systemctl enable slurmd.service
systemctl start slurmd.service
```

```
systemctl status slurmd.service
```

Setting up MariaDB database: master

Install MariaDB:

```
yum install mariadb-server mariadb-devel -y
```

Start the MariaDB service:

```
systemctl enable mariadb
systemctl start mariadb
systemctl status mariadb
```

Create the Slurm database user:

```
mysql
```

In mariaDB:

```
MariaDB[(none)]> GRANT ALL ON slurm_acct_db.* TO 'slurm'@'localhost' IDENTI
MariaDB[(none)]> SHOW VARIABLES LIKE 'have_innodb';
MariaDB[(none)]> FLUSH PRIVILEGES;
MariaDB[(none)]> CREATE DATABASE slurm_acct_db;
MariaDB[(none)]> quit;
```

Verify the databases grants for the *slurm* user:

```
mysql -p -u slurm
```

Tpye password for slurm: `1234` . In mariaDB:

```
MariaDB[(none)]> show grants;
MariaDB[(none)]> quit;
```

Create a new file /etc/my.cnf.d/innodb.cnf containing:

```
[mysqld]
innodb_buffer_pool_size=1024M
```

```
innodb_log_file_size=64M
innodb_lock_wait_timeout=900
```

To implement this change you have to shut down the database and move/remove logfiles:

```
systemctl stop mariadb
mv /var/lib/mysql/ib_logfile? /tmp/
systemctl start mariadb
```

You can check the current setting in MySQL like so:

```
MariaDB[(none)]> SHOW VARIABLES LIKE 'innodb_buffer_pool_size';
```

Create slurmdbd configuration file:

```
vim /etc/slurm/slurmdbd.conf
```

Set up files and permissions:

```
chown slurm: /etc/slurm/slurmdbd.conf
chmod 600 /etc/slurm/slurmdbd.conf
touch /var/log/slurmdbd.log
chown slurm: /var/log/slurmdbd.log
```

Paste the slurmdbd.conf in Configs and paste it into `slurmdbd.conf` .

Some variables are:

```
DbdAddr=localhost
DbdHost=localhost
DbdPort=6819
StoragePass=1234
StorageLoc=slurm_acct_db
```

Try to run *slurndbd* manually to see the log:

```
slurmdbd -D -vvv
```

Terminate the process by Control+C when the testing is OK.

Start the `slurmdbd` service:

```
systemctl enable slurmdbd
systemctl start slurmdbd
systemctl status slurmdbd
```

On the **master** node:

```
systemctl enable slurmctld.service
systemctl start slurmctld.service
systemctl status slurmctld.service
```

Reference: slothparadise, Niflheim, gabrieleiannetti