

Tutorial 2 (2019)

From ACE Lab

Follow the instructions below to complete this tutorial. The tutors are available to answer valid, well thought out questions. The tutors will not touch your keyboard or complete the tutorials for you. Answer questions provided in a text document and email to dmacleod@csir.co.za. Pay special attention to the instructions in red, you will need to demonstrate this to a tutor in order to complete that task.

Part 0 - Create a User account (Only if you do not have LDAP working)

If your LDAP user authentication from tutorial 1 is not working and you wish to rather move on with this tutorial, you can perform the steps below to bypass that LDAP section.

1. On both the headnode and compute node add a user and set a password with:

```
yum install sudo
adduser <name>
passwd <name>
```

2. Now give this user sudo (root privilege) by editing the file `/etc/sudoers` and adding the following to the bottom of the file:

```
<name>    ALL=(ALL)    ALL
```

3. Login to your cluster with the new users: `ssh <name>@<headnode>`

4. To execute a command as root add "sudo" before the command

5. To drop to a root shell execute "sudo su"

Part 1 - NFS

Network File System (NFS) enables you to easily share files over the network. In this section you will setup the NFS service and export a directory on your head node and mount it through the network on your compute node. With this completed, the head and compute nodes will have access to the same files as if they were available locally on the machine. This is useful in HPC where nodes must work together to solve large and complex problems.

1. On your head node: install, start and enable the nfs service

2. Edit `/etc/sysconfig/nfs` with:

```
MOUNTD_NFS_V3="yes"
RPCNFSDARGS="-N 4"
```

3. Edit the `/etc/exports` file and add this line

```
/home    *(rw,async,insecure,no_root_squash)
```

4. Restart the NFS service

QUESTION 1:

Explain the parameters of the above filesystem export?

5. On the compute nodes edit /etc/fstab and add:

```
<headnode>:/home /home nfs _netdev,intr 0 0
```

6. Force a remount of all the filesystems with

```
mount -a
```

7. Print all the mounted filesystems, you should see the NFS mount listed

```
df -h
```

To demonstrate that NFS is working correctly (and understand its purpose): as your user create a file in your home directory on head node. SSH to your compute node and confirm the file you created is present in your home directory on this machine.

With NFS working correctly, it is easy to enable key based SSH authentication. This will allow your user to SSH between the nodes in your cluster without requiring you to enter your password each time.

8. As your user, generate an SSH certificate with

```
ssh-keygen
```

9. Add the public key to the authorized keys file

```
cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

Typically you would need to copy this key to the remote machine to allow it to authenticate you. However, because your home directory is shared via NFS, the key you just generated is present on both systems, hence key based SSH should work.

10. Test password-less login by SSHing into another machine

```
ssh <computenode>
```

Part 2 - Zabbix

Zabbix is an open source solution for network and hardware monitoring. Zabbix includes various graphing and visualisation options which allows you to quickly assess the utilisation and efficiency of even large clusters.

Zabbix Dependencies

1. Install the Zabbix repo and PHP packages

```
rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm
yum clean all
yum install mod_php php-cli php-common php-devel php-pear php-gd php-mbstring php-mysql php-xml php-bcmath
```

QUESTION 2:

What is a repository?

2. Edit PHP configuration file /etc/php.ini, change the following lines:

```
max_execution_time = 600
max_input_time = 600
memory_limit = 256M
post_max_size = 32M
upload_max_filesize = 16M
date.timezone = Africa/Johannesburg
```

3. Restart httpd service

4. Install MariaDB (SQL database for CentOS)

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

5. Start and enable the service

6. Configure the database - there is no default password, set a new admin password and accept all other default options

```
mysql_secure_installation
```

7. Login to the database and run the following queries:

```
mysql -u root -p
create database zabbix character set utf8 collate utf8_bin;
grant all privileges on zabbix.* to zabbix@'localhost' identified by '123qwe';
grant all privileges on zabbix.* to zabbix@'%' identified by '123qwe';
flush privileges;
quit;
```

Zabbix Server

1. Install the server, frontend and agent:

```
yum install zabbix-get zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

2. Import database sample:

```
zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

3. Edit /etc/zabbix/zabbix_server.conf

```
DBHost=localhost
DBPassword=123qwe
```

4. Start and enable the zabbix-server service

Zabbix Agent

On each of your nodes, setup the Zabbix agent

1. yum install zabbix-agent
2. Edit /etc/zabbix/zabbix_agentd.conf with correct information

```
Server=<server_ip>
ServerActive=<server_ip>
Hostname=<hostname>
```

3. Start and enable zabbix-agent service

QUESTION 3:

Explain the client-server model?

Web setup

You will need to tunnel into the cluster as you did for LAM to continue.

1. Visit <http://192.168.0.x/zabbix> with your browser
2. Continue through the setup wizard, enter the zabbix database password you set earlier when prompted.
3. After completing the initial setup, login to the GUI as 'Admin', using the default password of 'zabbix'.
4. Change the default password by clicking the User icon at the top right and selecting 'Change password'.
5. Navigate to Configuration>Hosts and enable your headnode for monitoring (this may take a minute to update).
6. To check that your monitoring is working, navigate to Monitoring>Graphs and select 'CPU Utilization' at the top right.
7. Add your compute node by going to Configuration>Hosts>Add host. Fill in the appropriate details, and set the template to OS Linux and allocate it to the appropriate group.
8. You can create your own custom graph to display multiple properties from both of your nodes. **Demonstrate this to a tutor.**

Part 3 - SLURM Workload Manager

The SLURM Workload Manager (formerly known as Simple Linux Utility for Resource Management), is a free and open-source job scheduler for Linux, used by many of the world's supercomputers/computer clusters. It allows you to manage the resources of a cluster by deciding how users get access for some duration of time so

they can perform work.

Server Setup

1. Setup MariaDB

You have already setup the MariaDB SQL database in the Zabbix section, which will also be used by SLURM.

2. Install Munge packages (authentication service)

```
munge munge-libs munge-devel
```

3. Generate a Munge key for client authentication:

```
/usr/sbin/create-munge-key -r  
chown munge: /etc/munge/munge.key  
chmod 600 /etc/munge/munge.key
```

4. Copy the Munge key to your compute node to allow it to authenticate:

```
scp /etc/munge/munge.key <compute_node>:/etc/munge/munge.key
```

5. Start and enable the Munge service

6. Generate SLURM user

```
useradd slurm  
mkdir -p /var/spool/slurm/d
```

7. Install dependency packages:

```
openssl openssl-devel pam-devel numactl numactl-devel hwloc hwloc-devel lua lua-devel  
readline-devel rrdtool-devel curses-devel man2html libibmad libibumad rpm-build perl-devel  
gcc gcc-c++
```

8. Download SLURM source code tarball from <https://download.schedmd.com/slurm/>

Hint: copying the URL from your browsing then using the 'wget' command allows you to easily download files directly to your VM

Then export the version of the tarball you downloaded to the environment variable VERSION (eg: 19.05). This will make installation easier.

```
export VERSION=<19.05>
```

9. Build RPM packages for SLURM for installation

```
rpmbuild -ta slurm-*.bz2
```

You should have successfully generate SLURM RPMs in the directory /root/rpmbuild/RPMS/x86_64/ 10. Copy these RPMs to your compute node to install later.

QUESTION 4:

What is an RPM file and how do you think it might be useful?

10. Install Slurm server

```
mkdir -p /var/log/slurm /var/spool/slurm/ctld
chown -R slurm: /var/log/slurm /var/spool/slurm/ctld
```

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

```
yum --nogpgcheck localinstall slurm-$VERSION* slurm-devel-$VERSION* slurm-example-configs-$VERSION* \
    slurm-perlapi-$VERSION* slurm-torque-$VERSION* slurm-slurmctld-$VERSION*
```

11. Setup Slurm server

```
cp /etc/slurm/slurm.conf.example /etc/slurm/slurm.conf
```

set appropriate values for:

```
ClusterName=
ControlMachine=
ControlAddr=
```

populate the nodes and partitions:

```
NodeName=<computenode> Sockets=1 NodeAddr=10.0.0.<> CoresPerSocket=<>
PartitionName=debug Nodes=ALL Default=YES MaxTime=INFINITE State=UP
```

12. Start and enable the slurmctld service.**Client Setup****1. Setup Munge**

```
yum install munge munge-libs munge-devel -y
chown -R munge: /etc/munge/munge.key
```

2. Start and enable the Munge service.**3. Test Munge (from head node)**

```
munge -n | ssh <compute> unmunge
remunge
```

4. Install Slurm client

```
useradd slurm
mkdir -p /var/spool/slurm/d
```

Once again export the version of your SLURM to variable VERSION

```
yum --nogpgcheck localinstall slurm-$VERSION* slurm-devel-$VERSION* slurm-perlapi-$VERSION* \  
slurm-torque-$VERSION* slurm-example-configs-$VERSION* \  
slurm-slurmd-$VERSION* slurm-pam_slurm-$VERSION*
```

5. Copy the slurm.conf file you edited on your head node to the same location on your compute node.

6. Start and enable the slurmd service.

Return to your head node. To demonstrate that your scheduler is working, as your user run

```
sinfo
```

you should see your compute node in an idle state.

Start an interactive session on your compute node via the scheduler with

```
srun -N 1 --pty bash
```

you should land on your compute node. Rerun

```
sinfo
```

and

```
squeue
```

to show that your compute node is now allocated to this job.

Retrieved from '[https://www.ace.chpc.ac.za/acewiki/index.php?title=Tutorial_2_\(2019\)&oldid=1645](https://www.ace.chpc.ac.za/acewiki/index.php?title=Tutorial_2_(2019)&oldid=1645)'

- This page was last modified on 2 July 2019, at 13:30.
- This page has been accessed 82 times.