CREATE THREE VM's 1 HAS MASTER AND OTHER 2 HAS NODE1 AND NODE2 ,ALL MUST HAVE NAT AND HOSTONLY ADAPTOR

PREREQUISITS ON 3 MACHINES:

1>Disable the SELINUX on all vm's.

2>stop firewalld.service

3>update the /etc/hosts file.

4>Configure the passwordless SSH.

ON MASTER:

[root@pbs-master ~]# git clone https://github.com/openpbs/openpbs.git

[root@pbs-master ~]# yum --setopt=group\_package\_types=mandatory,default,optional groupinstall "Development Tools"

[root@pbs-master ~]# mv openpbs/ openpbs-23.06.06

[root@pbs-master ~]# tar -cvf /root/rpmbuild/SOURCES/openpbs-23.06.06.tar.gz openpbs-23.06.06/

[root@pbs-master openpbs-23.06.06]# yum install libtool-ltdl-devel hwloc-devel libedit-devel libical-devel ncurses-devel postgresql-devel postgresql-contrib python3-devel tcl-devel tk-devel zlib-devel expat-devel openssl-devel libXt-devel -y

[root@pbs-master openpbs-23.06.06]# rpmbuild -ba openpbs.spec

[root@pbs-master ~]# cd /root/rpmbuild/RPMS/x86\_64/

output:

openpbs-client-23.06.06-0.x86\_64.rpm openpbs-devel-23.06.06-0.x86\_64.rpm openpbs-server-23.06.06-0.x86\_64.rpm

openpbs-debuginfo-23.06.06-0.x86\_64.rpm openpbs-execution-23.06.06-0.x86\_64.rpm

[root@pbs-master x86\_64]# yum install openpbs-server-23.06.06-0.x86\_64.rpm

[root@pbs-master openpbs-23.06.06]# vi /etc/pbs.conf {YOU CAN CONFIGURE MOM AS WELL}

OUTPUT:

PBS\_EXEC=/opt/pbs

PBS\_SERVER=pbs-master

PBS\_START\_SERVER=1

PBS\_START\_SCHED=1

PBS\_START\_COMM=1

PBS\_START\_MOM=0

PBS\_HOME=/var/spool/pbs

PBS\_CORE\_LIMIT=unlimited

PBS\_SCP=/bin/scp

-----------------

[root@pbs-master openpbs-23.06.06]# cd /root/rpmbuild/RPMS/x86\_64/

[root@pbs-master x86\_64]# chmod 4755 /opt/pbs/sbin/pbs\_iff /opt/pbs/sbin/pbs\_rcp

[root@pbs-master x86\_64]# systemctl restart pbs

OR

[root@pbs-master x86\_64]# /etc/init.d/pbs restart

[root@pbs-master x86\_64]# systemctl restart pbs

[root@pbs-master x86\_64]# systemctl status pbs

OUTPUT:

Active: active (running) since Wed 2023-07-19 20:53:52 IST; 2s ago

[root@pbs-master x86\_64]# /etc/init.d/pbs status

OUTPUT:

pbs\_server is pid 30660

pbs\_sched is pid 29150

pbs\_comm is 29144

[root@pbs-master x86\_64]# . /etc/profile.d/pbs.

OUTPUT:

pbs.csh pbs.sh

[root@pbs-master x86\_64]# which qstat

OUTPUT:

/opt/pbs/bin/qstat

[root@pbs-master x86\_64]# qstat -B

OUTPUT:

Server Max Tot Que Run Hld Wat Trn Ext Status

---------------- ----- ----- ----- ----- ----- ----- ----- ----- -----------

pbs-master 0 0 0 0 0 0 0 0 Active

TROUBLESHOTING:

IF THE COMMAND [root@pbs-master x86\_64]# **qstat –B** IS NOT WORKING THEN RUN FOLLOWING COMMANDS:-

96 rpm -qa |grep openpbs

97 yum autoremove openpbs

98 yum autoremove openpbs-server-23.06.06-0.x86\_64

99 rm -rf /etc/pbs.conf

100 rm -rf /var/spool/pbs

101 cd rpmbuild/RPMS/x86\_64/

102 ls

103 ip a

104 cat /etc/hosts

105 yum install openpbs-server-23.06.06-0.x86\_64.rpm

106 vi /etc/pbs.conf

107 chmod 4755 /opt/pbs/sbin/pbs\_iff /opt/pbs/sbin/pbs\_rcp

108 /etc/init.d/pbs restart

109 /etc/init.d/pbs status

110 qstat –B

OUTPUT:

Server Max Tot Que Run Hld Wat Trn Ext Status

---------------- ----- ----- ----- ----- ----- ----- ----- ----- -----------

pbs-master 0 0 0 0 0 0 0 0 Active

ON NODE 1

yum install git -y

git clone https://github.com/openpbs/openpbs.git

ls

cd openpbs/

./autogen.sh

yum provides \*/autoreconf -y

yum install autoconf automake libtool -y

[root@pbs-node1 openpbs]# ls

aclocal.m4 autom4te.cache buildutils CODE\_OF\_CONDUCT.md configure.ac COPYRIGHT INSTALL m4 Makefile.in openpbs.spec PBS\_License.txt src valgrind.supp

autogen.sh azure-pipelines.yml ci configure CONTRIBUTING.md doc LICENSE Makefile.am openpbs-rpmlintrc openpbs.spec.in README.md test

[root@pbs-node1 openpbs]./configure

[root@pbs-node1 openpbs]./configure --prefix=/opt/pbs

if we not any dependencies then run below commands

[root@pbs-node1 openpbs]yum install openssl\* -y

[root@pbs-node1 openpbs]yum install libXt-devel -y

After that create direcrtory.

[root@pbs-node1 openpbs]mkdir -p /opt/pbs/lib

[root@pbs-node1 openpbs]ls /opt/pbs/

lib

Again configure the below commands

[root@pbs-node1 openpbs]./configure --prefix=/opt/pbs

[root@pbs-node1 openpbs]yum provides \*/autoreconf -y

If you still get any error or not get any dependencies then run below command

[root@pbs-node1 openpbs]# yum install libtool-ltdl-devel hwloc-devel libedit-devel libical-devel ncurses-devel postgresql-devel postgresql-contrib python3-devel tcl-devel tk-devel zlib-devel expat-devel openssl-devel libXt-devel -y

[root@pbs-node1 openpbs]./configure --prefix=/opt/pbs

Again configure with this commands

[root@pbs-node1 openpbs] make (run this command)

[root@pbs-node1 openpbs]# make install

[root@pbs-node1 openpbs]# . /etc/profile.d/pbs.sh

[root@pbs-node1 openpbs]# which qstat

[root@pbs-node1 openpbs]# chmod +x /opt/pbs/etc/pbs.sh

[root@pbs-node1 openpbs]# export PATH=${PATH}:/opt/pbs/bin

[root@pbs-node1 openpbs]# which qstat

/opt/pbs/bin/qstat

[root@pbs-node1 openpbs]# vim /etc/pbs.conf

PBS\_EXEC=/opt/pbs

PBS\_SERVER=pbs-master

PBS\_START\_SERVER=0

PBS\_START\_SCHED=0

PBS\_START\_COMM=0

PBS\_START\_MOM=1

PBS\_HOME=/var/spool/pbs

PBS\_CORE\_LIMIT=unlimited

PBS\_SCP=/bin/scp

[root@pbs-node1 openpbs]# cat /etc/pbs.conf

[root@pbs-node1 openpbs]# systemctl status pbs

================================================================================================================================================================================================================

ON MASTER MACHINE

yum install -y nfs-utils

systemctl start nfs-server rpcbind

systemctl enable nfs-server rpcbind

mkdir /mnt/home

chmod 777 /mnt/home/

systemctl restart nfs

ls /root/rpmbuild/RPMS/x86\_64/

cd /root/rpmbuild/RPMS/x86\_64/

cp \* /home/rpms/

cp \* /mnt/home/rpms/

cd

mkdir /home/rpms

cd /root/rpmbuild/RPMS/x86\_64/

cp \* /home/rpms/

vim /etc/exports

/home \*(rw,sync,no\_root\_squash)

cp \* /home/rpms/history

================================================================================================================================================================================================================

ON NODE 2

yum install -y nfs-utils

showmount -e 192.168.206.160(NAT ip of master)

mkdir /mnt/home

mount 192.168.206.160:/home /mnt/home

ls /root/rpmbuild/RPMS/x86\_64/

showmount -e 192.168.206.160

df -Th

cd /mnt/home/

ls

cd rpms

[root@pbs-node2 rpms]# ls

openpbs-client-23.06.06-0.x86\_64.rpm openpbs-devel-23.06.06-0.x86\_64.rpm openpbs-server-23.06.06-0.x86\_64.rpm

openpbs-debuginfo-23.06.06-0.x86\_64.rpm openpbs-execution-23.06.06-0.x86\_64.rpm

yum install openpbs-execution-23.06.06-0.x86\_64.rpm -y

vim /etc/pbs.conf

PBS\_EXEC=/opt/pbs

PBS\_SERVER=pbs-master

PBS\_START\_SERVER=0

PBS\_START\_SCHED=0

PBS\_START\_COMM=0

PBS\_START\_MOM=1

PBS\_HOME=/var/spool/pbs

PBS\_CORE\_LIMIT=unlimited

PBS\_SCP=/bin/scp

[root@pbs-node2 rpms]# vim /var/spool/pbs/mom\_priv/config

$logevent 0x1ff

#$clientname pbs-node2

$restrict\_user\_maxsysid 999

ON NODE 1

[root@pbs-node2 rpms]# vim /var/spool/pbs/mom\_priv/config

$restrict\_user\_maxsysid 999

ON MASTER MACHINE

[root@pbs-master ~]# vim /var/spool/pbs/server\_priv/nodes

pbs-node1 np=1

pbs-node2 np=1

[root@pbs-master ~]# pbsnodes -a

pbsnodes: Server has no node list

root@pbs-master x86\_64]# qmgr

Max open servers: 49

Qmgr: create node pbs-node1

Qmgr: create node pbs-node2

Qmgr:

Ctlr+c [to exit]

[root@pbs-master x86\_64]# pbsnodes -a

pbs-node1

Mom = pbs-node1

Port = 15002

pbs\_version = 23.06.06

ntype = PBS

state = free

pcpus = 2

resources\_available.arch = linux

resources\_available.host = pbs-node1

resources\_available.mem = 1863028kb

resources\_available.ncpus = 2

resources\_available.vnode = pbs-node1

resources\_assigned.accelerator\_memory = 0kb

resources\_assigned.hbmem = 0kb

resources\_assigned.mem = 0kb

resources\_assigned.naccelerators = 0

resources\_assigned.ncpus = 0

resources\_assigned.vmem = 0kb

resv\_enable = True

sharing = default\_shared

license = l

last\_state\_change\_time = Thu Jul 20 20:24:19 2023

pbs-node2

Mom = pbs-node2

Port = 15002

pbs\_version = 23.06.06

ntype = PBS

state = free

pcpus = 2

resources\_available.arch = linux

resources\_available.host = pbs-node2

resources\_available.mem = 1863028kb

resources\_available.ncpus = 2

resources\_available.vnode = pbs-node2

resources\_assigned.accelerator\_memory = 0kb

resources\_assigned.hbmem = 0kb

resources\_assigned.mem = 0kb

resources\_assigned.naccelerators = 0

resources\_assigned.ncpus = 0

resources\_assigned.vmem = 0kb

resv\_enable = True

sharing = default\_shared

license = l

last\_state\_change\_time = Thu Jul 20 20:24:33 2023

root@pbs-master x86\_64]# vim /etc/pbs.conf

PBS\_EXEC=/opt/pbs

PBS\_SERVER=pbs-master

PBS\_START\_SERVER=1

PBS\_START\_SCHED=1

PBS\_START\_COMM=1

PBS\_START\_MOM=1

PBS\_HOME=/var/spool/pbs

PBS\_CORE\_LIMIT=unlimited

PBS\_SCP=/bin/scp (make MOM=1 )

then add master node

~

root@pbs-master x86\_64]# qmgr

Max open servers: 49

Qmgr: create node pbs-master

[root@pbs-master x86\_64]# pbsnodes -a

pbs-node1

Mom = pbs-node1

Port = 15002

pbs\_version = 23.06.06

ntype = PBS

state = free

pcpus = 2

resources\_available.arch = linux

resources\_available.host = pbs-node1

resources\_available.mem = 1863028kb

resources\_available.ncpus = 2

resources\_available.vnode = pbs-node1

resources\_assigned.accelerator\_memory = 0kb

resources\_assigned.hbmem = 0kb

resources\_assigned.mem = 0kb

resources\_assigned.naccelerators = 0

resources\_assigned.ncpus = 0

resources\_assigned.vmem = 0kb

resv\_enable = True

sharing = default\_shared

license = l

last\_state\_change\_time = Thu Jul 20 20:30:34 2023

pbs-node2

Mom = pbs-node2

Port = 15002

pbs\_version = 23.06.06

ntype = PBS

state = free

pcpus = 2

resources\_available.arch = linux

resources\_available.host = pbs-node2

resources\_available.mem = 1863028kb

resources\_available.ncpus = 2

resources\_available.vnode = pbs-node2

resources\_assigned.accelerator\_memory = 0kb

resources\_assigned.hbmem = 0kb

resources\_assigned.mem = 0kb

resources\_assigned.naccelerators = 0

resources\_assigned.ncpus = 0

resources\_assigned.vmem = 0kb

resv\_enable = True

sharing = default\_shared

license = l

last\_state\_change\_time = Thu Jul 20 20:30:34 2023

pbs-master

Mom = pbs-master

Port = 15002

pbs\_version = 23.06.06

ntype = PBS

state = free

pcpus = 2

resources\_available.arch = linux

resources\_available.host = pbs-master

resources\_available.mem = 3861076kb

resources\_available.ncpus = 2

resources\_available.vnode = pbs-master

resources\_assigned.accelerator\_memory = 0kb

resources\_assigned.hbmem = 0kb

resources\_assigned.mem = 0kb

resources\_assigned.naccelerators = 0

resources\_assigned.ncpus = 0

resources\_assigned.vmem = 0kb

resv\_enable = True

sharing = default\_shared

license = l

last\_state\_change\_time = Thu Jul 20 20:30:37 2023

[root@pbs-master x86\_64]# qstat -B

Server Max Tot Que Run Hld Wat Trn Ext Status

---------------- ----- ----- ----- ----- ----- ----- ----- ----- -----------

pbs-master 0 0 0 0 0 0 0 0 Active

qsub -I (To submit the job)

for that you need to create users on master, node1 and in node2 also

useradd testuser

passwd testuser

Then su - testuser run this on all 3 machine

Then come to the master machine

[root@pbs-master ~]# su - testuser

[testuser@pbs-master ~]$ qsub -I

qsub: waiting for job 2.pbs-master to start

qsub: job 2.pbs-master ready

NOW we submit job to pbs

for that we create in user demo.sh file

1) vim demo.sh

#!/bin/sh

### Set the job name (for your reference)

#PBS -N testjob

### Set the project name, your department code by default

#PBS -P cc

####

#PBS -l select=1:ncpus=1

### Specify "wallclock time" required for this job, hhh:mm:ss

#PBS -l walltime=00:01:00

#PBS -l software=replace\_with\_Your\_software\_name

# After job starts, must goto working directory.

# $PBS\_O\_WORKDIR is the directory from where the job is fired.

echo "==============================="

echo $PBS\_JOBID

#cat $PBS\_NODEFILE

echo "==============================="

cd $PBS\_O\_WORKDIR

:wq

qsub demo.sh

4.pbs-master

qstat

For 2nd job

vim testjob.sh

#!/bin/sh

### Set the job name (for your reference)

#PBS -N testjob

### Set the project name, your department code by default

#PBS -P cc

### Request email when job begins and ends

#PBS -m bea

### Specify email address to use for notification.

#PBS -M $USER@iitd.ac.in

### Specify required no of node in select and specfify required cpu cores in ncpus

#PBS -l select=1:ncpus=1

### Specify "wallclock time" required for this job, hhh:mm:ss

#PBS -l walltime=00:00:05

#PBS -l software=replace\_with\_Your\_software\_name

# After job starts, must goto working directory.

# $PBS\_O\_WORKDIR is the directory from where the job is fired.

echo "==============================="

echo $PBS\_JOBID

#cat $PBS\_NODEFILE

echo "==============================="

hostname

cd $PBS\_O\_WORKDIR

#job

###time -p mpirun -n {n\*m} executable

#NOTE

# The job line is an example : users need to change it to suit their applications

# The PBS select statement picks n nodes each having m free processors

# OpenMPI needs more options such as $PBS\_NODEFILE

:wq

[testuser@pbs-master ~]$ vim testjob.sh

[testuser@pbs-master ~]$ qsub testjob.sh

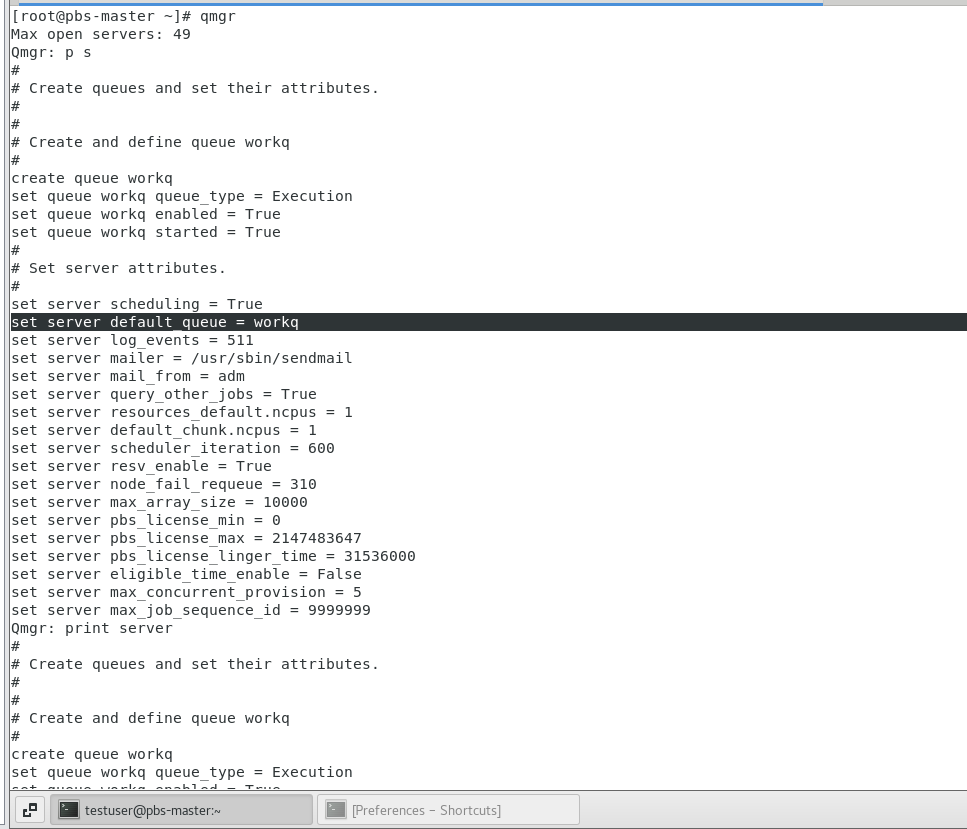
6.pbs-master

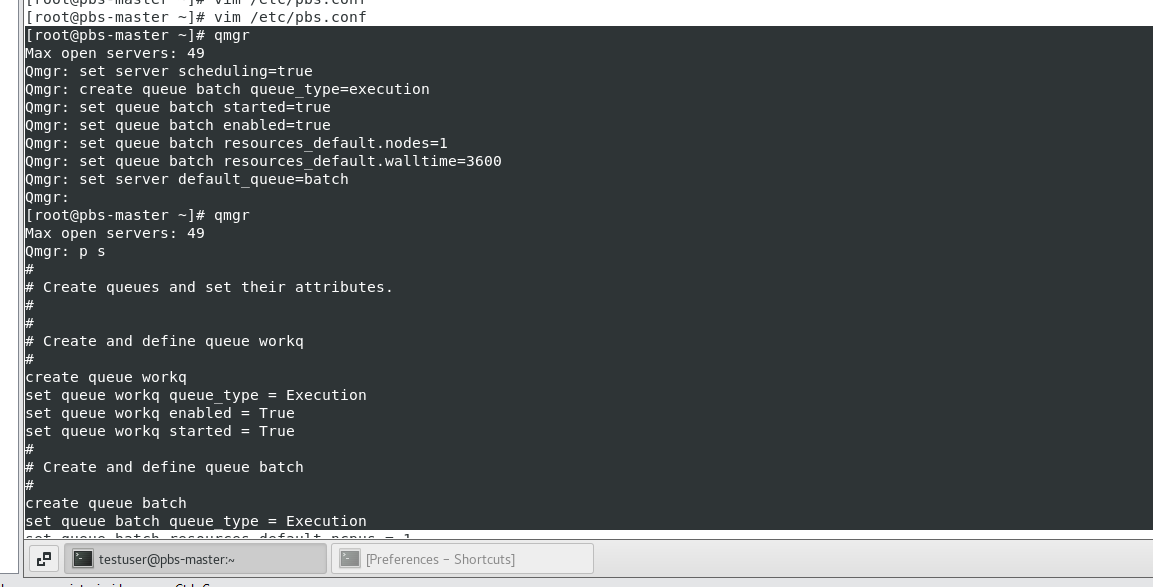
[testuser@pbs-master ~]$ qstat

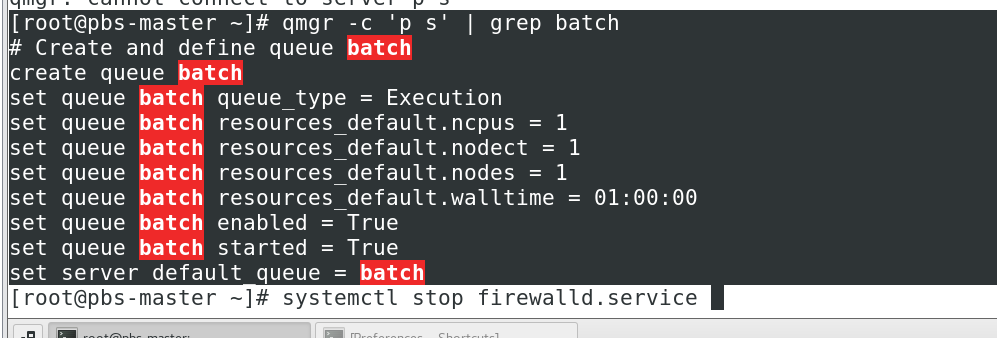
Job id Name User Time Use S Queue

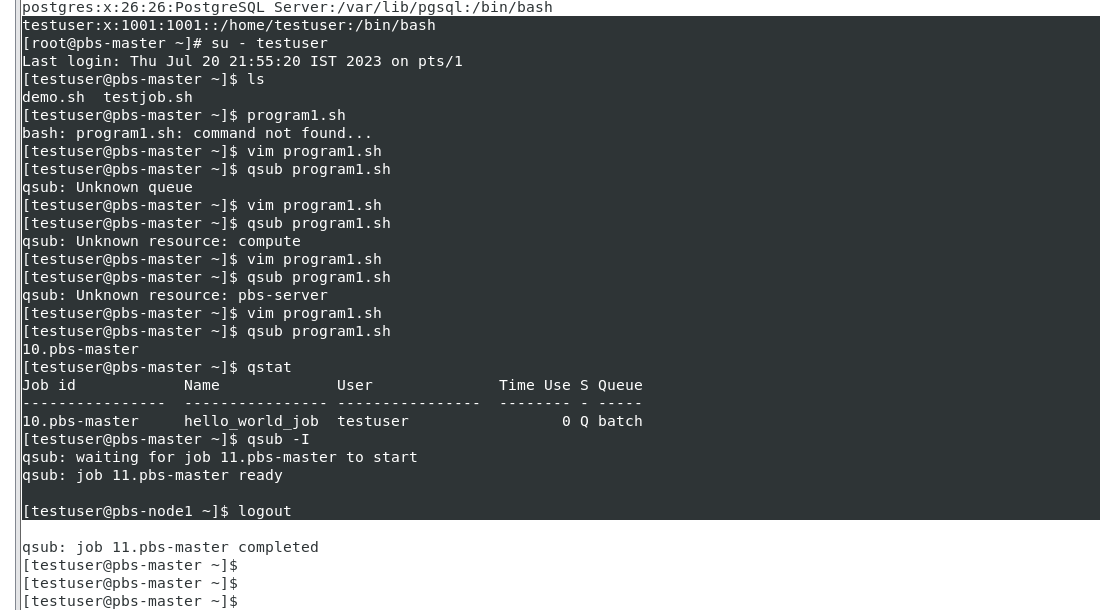
---------------- ---------------- ---------------- -------- - -----

6.pbs-master testjob.sh testuser 00:00:00 E workq









**Program1.sh**

#!/bin/sh

### Job name

#PBS -N hello\_world\_job

### Output files

#PBS -o hello\_world\_job.stdout

#PBS -e hello\_world\_job.stderr

### Queue name

#PBS -q dqueue

### Number of nodes

#PBS -l nodes=4:compute#shared

# Print the default PBS server

echo PBS default server is $PBS\_DEFAULT

# Print the job's working directory and enter it.

echo Working directory is $PBS\_O\_WORKDIR

cd $PBS\_O\_WORKDIR

# Print some other environment information

echo Running on host `hostname`

echo Time is `date`

echo Directory is `pwd`

echo This jobs runs on the following processors:

NODES=`cat $PBS\_NODEFILE`

echo $NODES

# Compute the number of processors

NPROCS=`wc -l < $PBS\_NODEFILE`

echo This job has allocated $NPROCS nodes

# Run hello\_world

for NODE in $NODES; do

ssh $NODE "hello\_world" &

done

# Wait for background jobs to complete.

wait