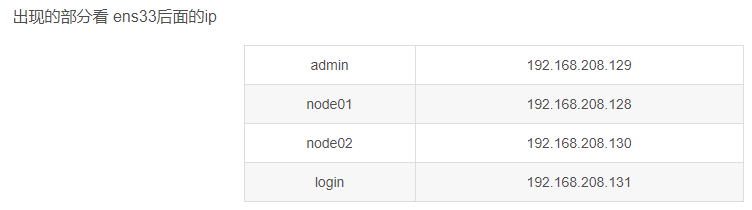
# 系统：**CentOS8**

# 联网，选择自动连接，获得ip



# 启动一个初始化脚本：(主、从 节点都需要)

# 默认源 换成 阿里云镜像，否则有些包下载不了

# 安装一些常用包

# 关闭防火墙

./CentOS8initial.sh

# 在所有节点，关闭SElinux服务，(可选)

vi /etec/selinux/config

#修改成SELINUX=disabled

# 安装必要的包 (主、从 节点都需要安装)

yum install -y epel-release

yum install -y munge munge-libs rng-tools

yum install -y mariadb-devel pam-devel readline-devel

yum install -y rpm-build

yum install -y nfs-utils rpcbind

# 由于munge-devel包在阿里镜像源没有，只能下载到本地再安装

**yum localinstall -y munge-devel-\*.rpm**

（userdel munge？？此时已经有group munge 976了，是否要删除？？）

# 时间同步(主、从 节点都需要)

timedatectl set-timezone Asia/Shanghai

# 在主节点，修改本机名称

hostnamectl set-hostname master

# 在从节点1，修改本机名称

hostnamectl set-hostname node1

# 在从节点2，修改本机名称

hostnamectl set-hostname node2

# 在从节点3，修改本机名称

hostnamectl set-hostname node3

**###主、从 节点全部重启reboot###(可以放在后面一并？)**

# 主、从 节点彼此间直接ssh，而不用输入ip地址

# 在主节点操作：

vi /etc/hosts

末尾添加几行192.168.153.128 master

# 在主节点，将文件分发给其余所有从节点

scp /etc/hosts root@node1:/etc/

scp /etc/hosts root@node2:/etc/

scp /etc/hosts root@node3:/etc/

设置正确的话，此时能看到主机名变了

彼此ssh不用输ip

# **彼此之间免密登录**

**# 在主节点操作**

ssh-keygen

enter enter enter#三次回车

# 在主机操作，使得主机到其它节点免密

ssh-copy-id -i /root/.ssh/id\_rsa.pub master

ssh-copy-id -i /root/.ssh/id\_rsa.pub node1

ssh-copy-id -i /root/.ssh/id\_rsa.pub node2

ssh-copy-id -i /root/.ssh/id\_rsa.pub node3

# 在主机操作，使得各节点彼此之间免密

scp /root/.ssh/\* root@node1:/root/.ssh/

scp /root/.ssh/\* root@node2:/root/.ssh/

scp /root/.ssh/\* root@node3:/root/.ssh/

#查看是否配置好

ssh node1

ssh node2

#不用输入密码直接登录到其他主机代表成功

#退出登录exit

# NFS 文件共享(可选?)

# 在主节点操作

# 创建共享目录

mkdir -p /public

chmod 755 /public

**vi /etc/exports添加一行**

/public/ \*(rw,async,insecure,no\_root\_squash)

#读写，异步同步，不检查端口，root可直接进

# 在每个从节点操作：

**vi /etc/fstab添加一行**

maste5:/home/public /home/public nfs defaults 0 0

# 将从节点的/public文件夹与主节点同步

# 启动文件同步功能

systemctl start nfs-server.service

systemctl enable nfs-server.service

systemctl start rpcbind

systemctl enable rpcbind

从节点**重启**

reboot (之前输错了，系统崩了)

# munge 身份认证包，每个节点都要安装

# **主节点**操作

# 创建随机密钥，取1024个备用

**rngd -r /dev/urandom && create-munge-key**

**dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key**

chown munge: /etc/munge/munge.key && chmod 400 /etc/munge/munge.key

chown -R munge: /var/lib/munge

chown -R munge: /var/run/munge

chown -R munge: /var/log/munge

systemctl start munge

systemctl enable munge

**# 将主节点的key分发给从节点**

**scp /etc/munge/munge.key root@node1:/etc/munge/**

**scp /etc/munge/munge.key root@node2:/etc/munge/**

**scp /etc/munge/munge.key root@node3:/etc/munge/**

# munge 身份认证包

# **从节点**操作:

(**注意别忘了将主节点的密钥复制过来**)

**chown munge: /etc/munge/munge.key && chmod 400 /etc/munge/munge.key**

chown -R munge: **/etc/munge**

chown -R munge: **/var/lib/munge**

chown -R munge: **/var/run/munge**

chown -R munge: **/var/log/munge**

systemctl start munge

systemctl enable munge

# 测试munge是否正常

munge -n | unmunge

# 验证compute node，远程解码

munge -n | ssh node5 unmunge

## **正式安装Slurm** ##

# **在主节点操作**

export SlurmUSER=1191 && groupadd -g $SlurmUSER slurm && useradd -m -c "Slurm workload manager" -d /var/lib/slurm -u $SlurmUSER -g slurm -s /bin/bash slurm

**# 将slurm包，拷贝到/目录**

rpmbuild -ta slurm\*.tar.bz2

(本步骤耗时较多)

# 前往rpmbuild路径，对所有的rpm文件进行安装

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

**yum localinstall slurm-\*.rpm**

# 13个

# 将官方配置文件的模板复制一份，方便修改

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

vi /etc/slurm/slurm.conf

对如下信息修改，修改好保存退出

*ClusterName=cluster#集群名称，可改可不改*

*SlurmctldHost=master#主节点名称*

*# COMPUTE NODES*

*NodeName=node1 CPUs=1 State=UNKNOWN*

*NodeName= node2 CPUs=1 State=UNKNOWN*

*NodeName= node3 CPUs=1 State=UNKNOWN*

*NodeName=master CPUs=8 Boards=1 SocketsPerBoard=2 CoresPerSocket=4 ThreadsPerCore=1 State=UNKNOWN*

*NodeName=node01 CPUs=2 Boards=1 SocketsPerBoard=2 CoresPerSocket=1 ThreadsPerCore=1 State=UNKNOWN*

*NodeName=node02 CPUs=2 Boards=1 SocketsPerBoard=1 CoresPerSocket=2 ThreadsPerCore=1 State=UNKNOWN*

*NodeName=node03 CPUs=4 Boards=1 SocketsPerBoard=2 CoresPerSocket=2 ThreadsPerCore=1 State=UNKNOWN*

*PartitionName=login Nodes=master Default=YES MaxTime=INFINITE State=UP*

*PartitionName=cpu Nodes=node[01-03] Default=YES MaxTime=INFINITE State=UP*

# 创建一些默认文件夹

mkdir /var/spool/slurmctld

chown slurm:slurm /var/spool/slurmctld

chmod 755 /var/spool/slurmctld

mkdir /var/spool/slurmd

chown slurm: /var/spool/slurmd

chmod 755 /var/spool/slurmd

touch /var/log/slurmd.log

touch /var/log/slurmctld.log

touch /var/log/slurmdbd.log

chown slurm:slurm /var/log/slurmd.log

chown slurm:slurm /var/log/slurmctld.log

chown slurm:slurm /var/log/slurmdbd.log

# 启动

systemctl start slurmd.service

systemctl enable slurmd.service

systemctl start slurmctld.service

systemctl enable slurmctld.service

# **在从节点操作**

#创建slurm用户

export SlurmUSER=1191 && groupadd -g $SlurmUSER slurm && useradd -m -c "Slurm workload manager" -d /var/lib/slurm -u $SlurmUSER -g slurm -s /bin/bash slurm

**rpmbuild -ta slurm\*.tar.bz2**

**cd /root/rpmbuild/RPMS/x86\_64/ && yum localinstall slurm\*.rpm &&** ls /etc/slurm/

# 看看有没有配置文件模板，有的话说明装好了

**从节点应该直接套用主节点的配置！！**

从主节点操作，把文件复制过来：

ssh master

scp /etc/slurm/slurm.conf node1:/etc/slurm/

exit

**回到从节点继续:**

mkdir /var/spool/slurmctld

chown slurm:slurm /var/spool/slurmctld

chmod 755 /var/spool/slurmctld

mkdir /var/spool/slurmd

chown slurm: /var/spool/slurmd

chmod 755 /var/spool/slurmd

touch /var/log/slurmd.log

touch /var/log/slurmctld.log

touch /var/log/slurmdbd.log

chown slurm:slurm /var/log/slurmd.log

chown slurm:slurm /var/log/slurmctld.log

chown slurm:slurm /var/log/slurmdbd.log

systemctl start slurmd.service

systemctl enable slurmd.service

systemctl stop slurmctld.service

systemctl disable slurmctld.service

# 莫名其妙的sinfo出错，创建开机自启动脚本

修改vi /etc/rc.d/rc.local 文件,其会在 Linux 系统各项服务都启动完毕之后再被运行。所以你想要自己的脚本在开机后被运行的话，可以将自己脚本路径加到该文件里。

mkdir /run/slurm/conf

cp /etc/slurm/slurm.conf /run/slurm/conf/

chmod +x /etc/rc.d/rc.local

reboot试试

好像从节点不需要，sinfo也能运行

有时候状态不对，手动改成idle

**scontrol update NodeName=节点名 State=idle**

**scontrol update NodeName=node01 State=idle**