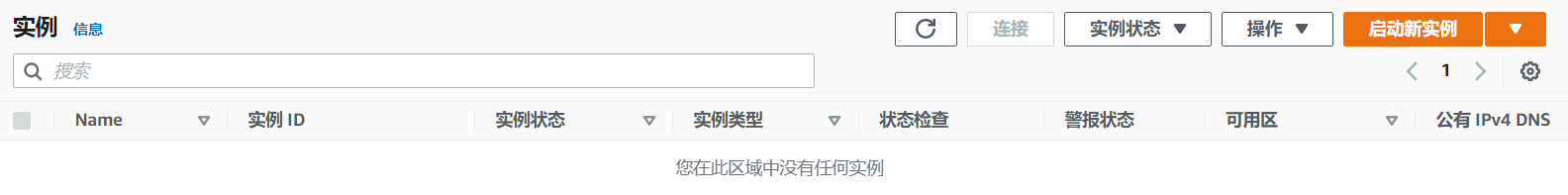
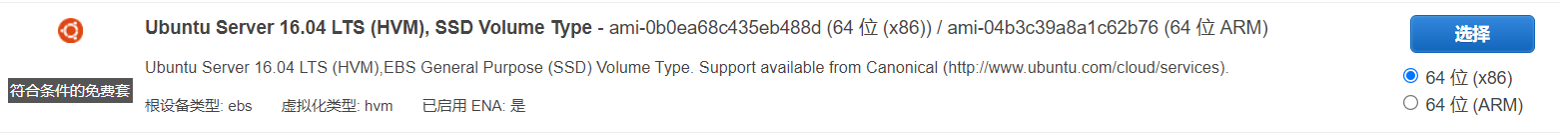


Hadoop实验报告

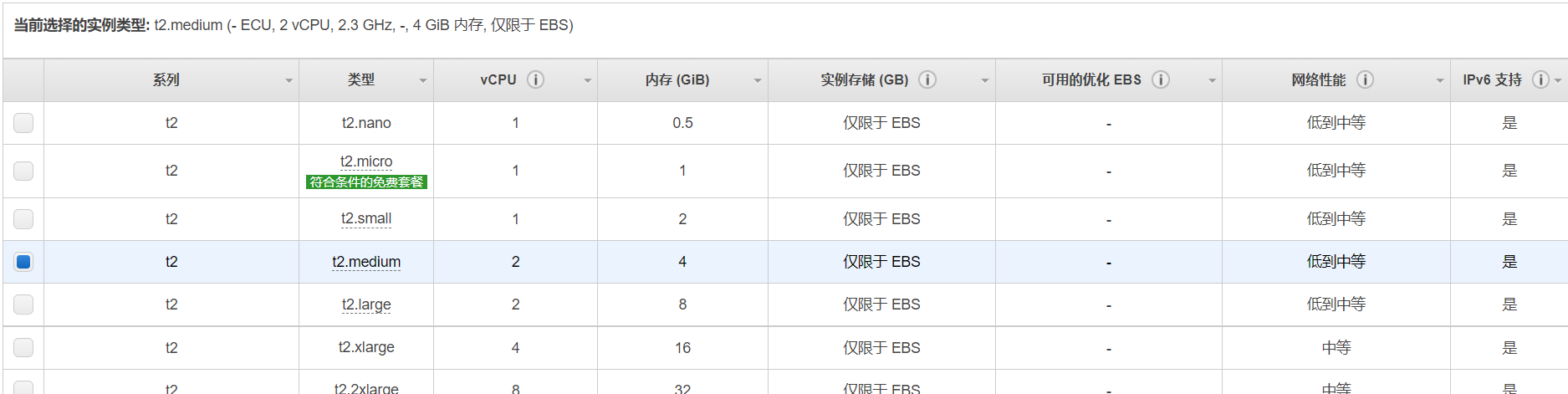
# 创建新实例



刚开始选择Ubuntu，后面发现无法启动实例，于是就换成了最上面的Linux2操作系统。

# 选择实例的时候刚开始选择了micro，最后选择的是medium。





# 选择启动三台实例。



# 刚开始选择存储卷大小15GB，后来选择的是8GB。





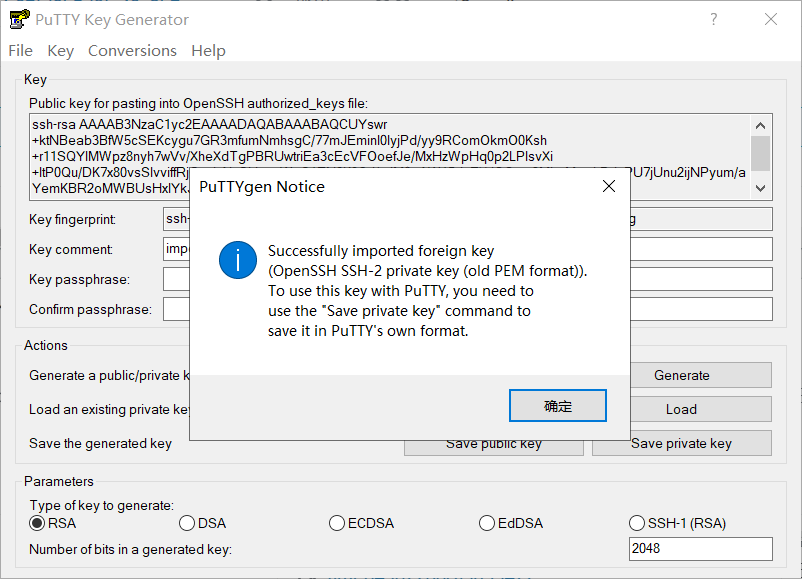
# 在启动实例前的最后一步，因为是windows系统，不使用已有密钥对，点击勾选创建新的密钥对。然后对生成的文件命名，即myrsa.pem。



# 启动实例，将实例命名为master，slaver01，slaver02。



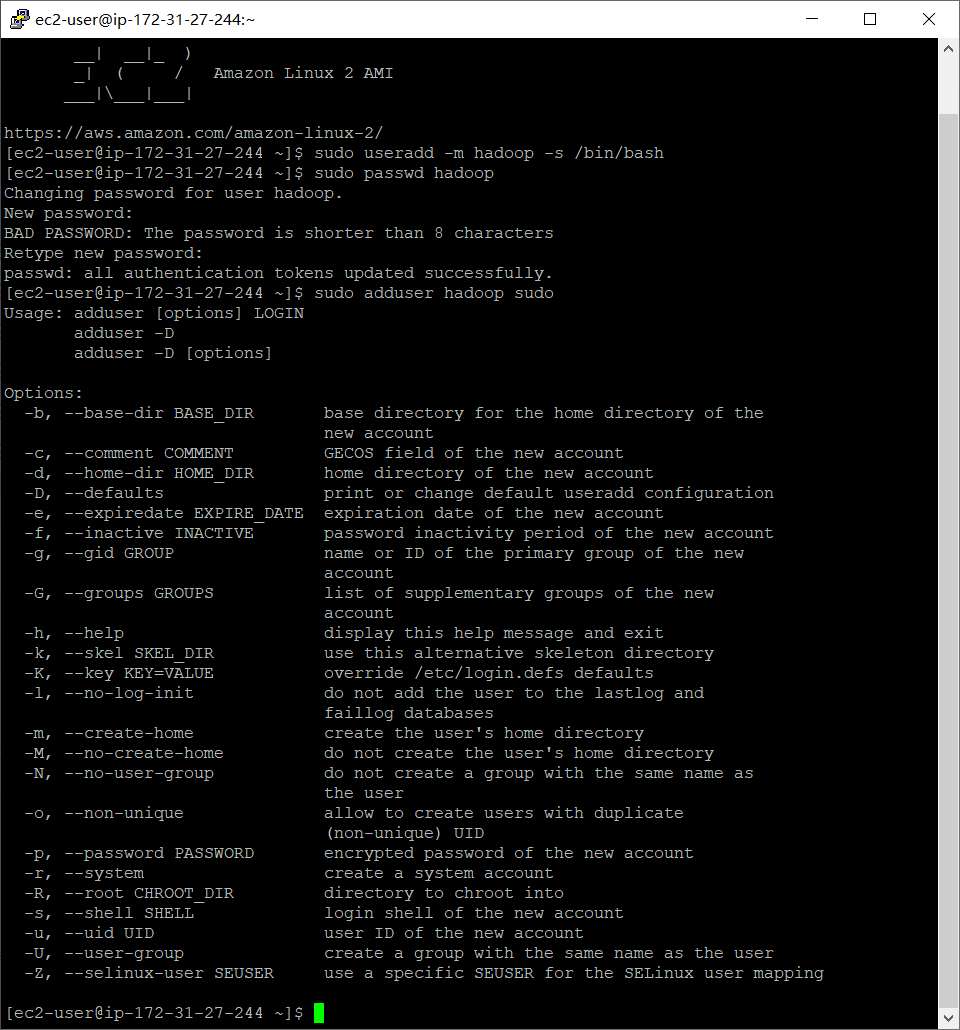
# 因为是windows系统，所以我要用putty进行连接。因此还要先用putty generator将pem文件进行格式转换，转换成.ppk文件。



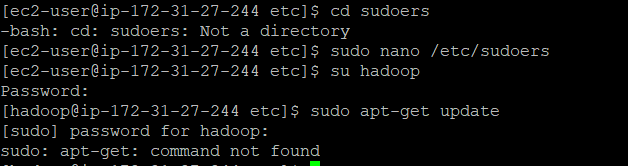
# 然后利用putty进行连接，如下图，连接成功。



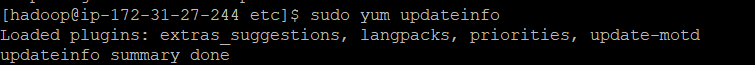
# 输入用户名ec2-user，直接进入欢迎界面，执行linux命令建hadoop用户。



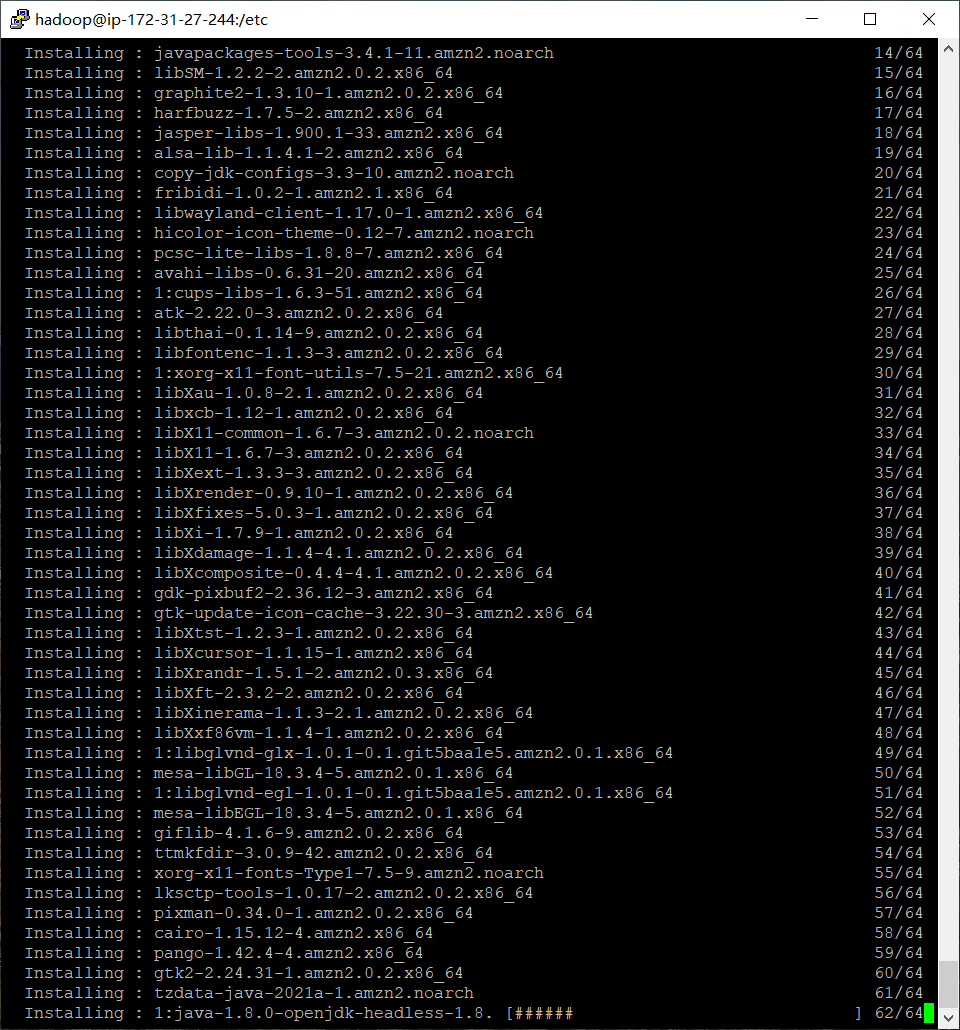
# 执行命令进入hadoop用户。



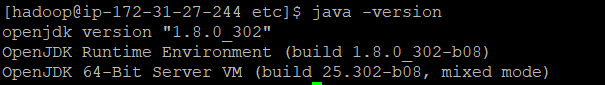
# 因为没用Ubuntu系统，所以apt-get命令是不生效的，因此用yum来代替。



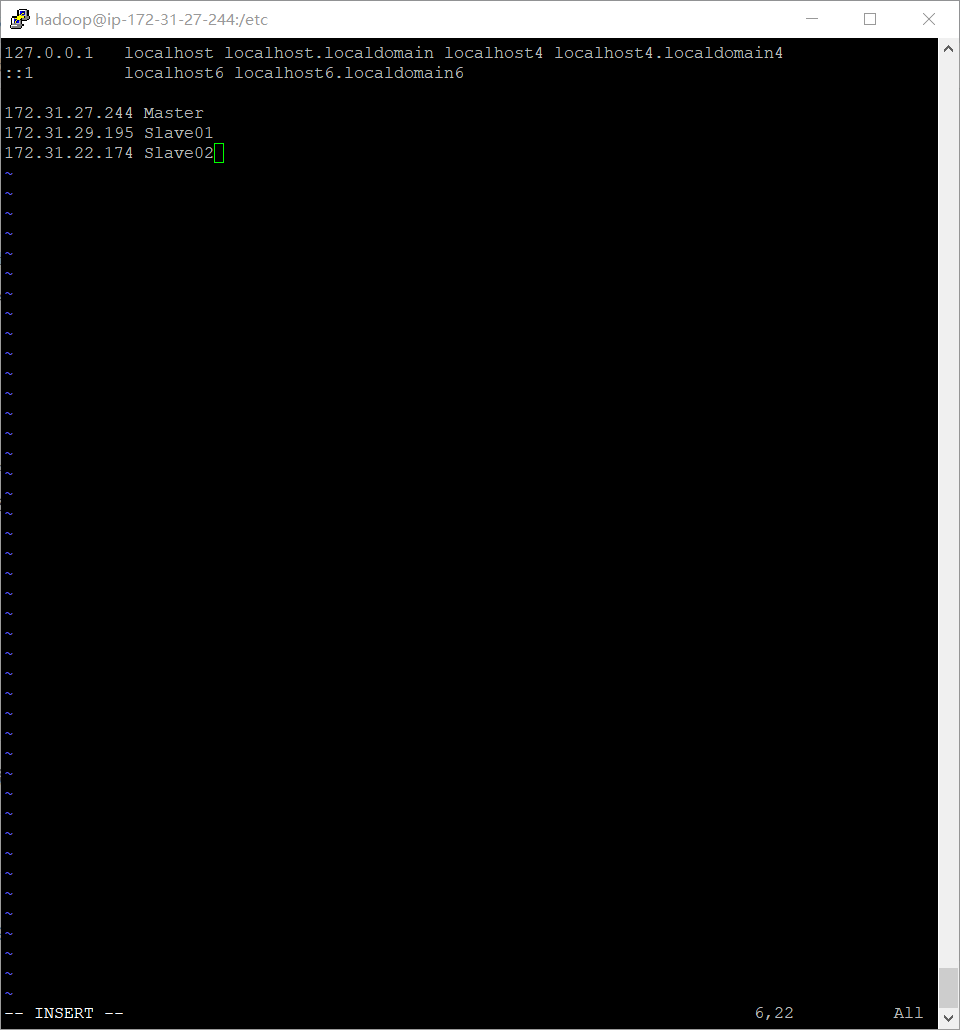
# 接下来是要下载java1.8，我从官网找到Java-openjdk1.8，并将它的url复制，用linux的wget命令进行在线下载。如下是下载过程的截图。



# 检查Java下载结果。可以看到版本号能被打印。并且Java的环境变量也自动配置好了，用javac也能直接编译.Java类型文件。



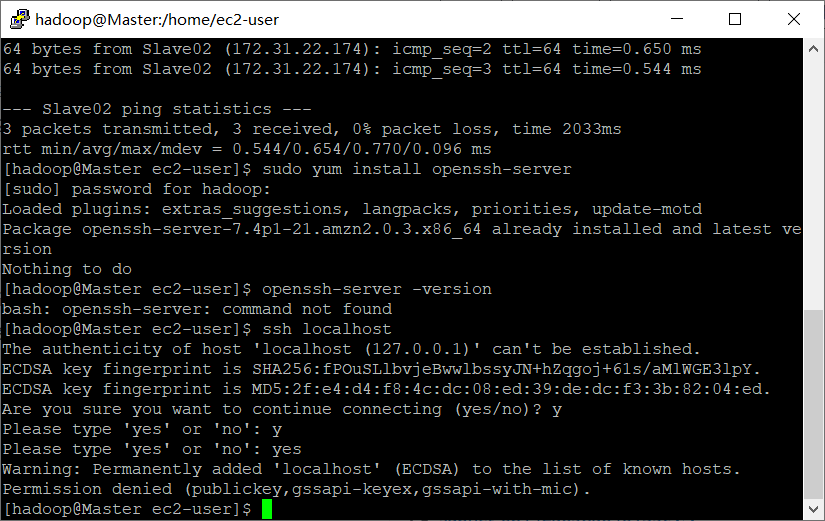
# 然后是进入/etc/hosts去分别加入主从的IP。



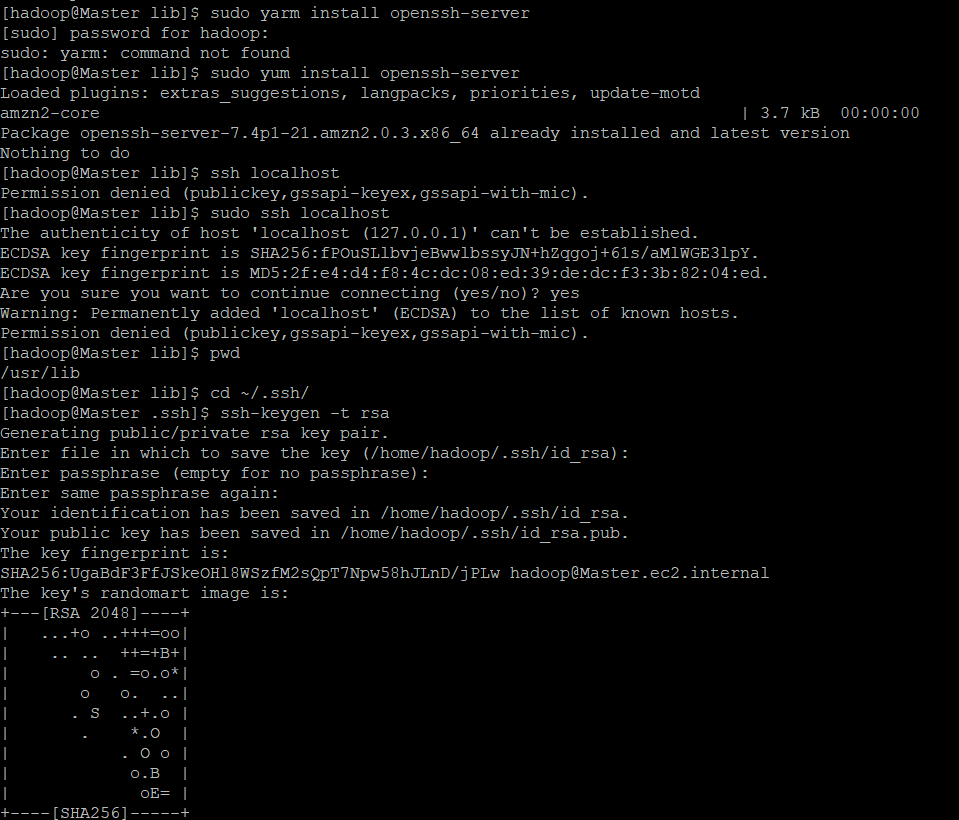
# 在实例那边的安全组添加入站规则。添加所有流量，端口范围全部。



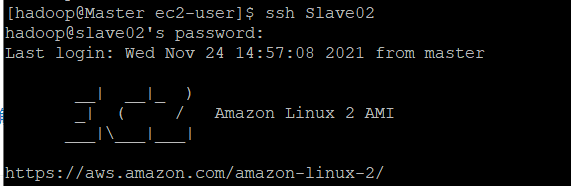
# ssh连接localhost



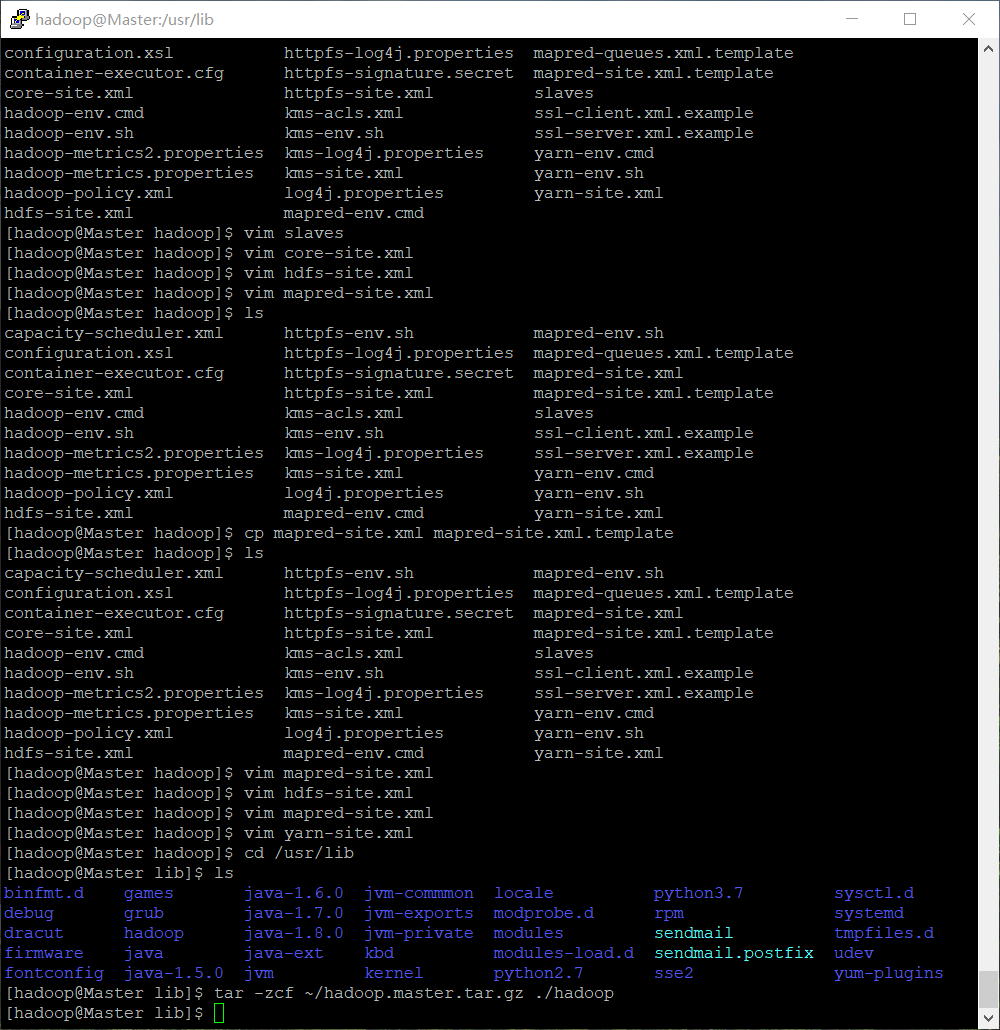
# 生成密钥，将密钥保存在id\_rsa.pub文件中。



# 刚开始ssh连接slave01，slave02的时候，发现不能连接。之后去找解决方法，发现将/etc/ssh/sshd\_config 文件中的PasswordAuthentication 改为yes。就可以连接了，如下图，连接Slave02。

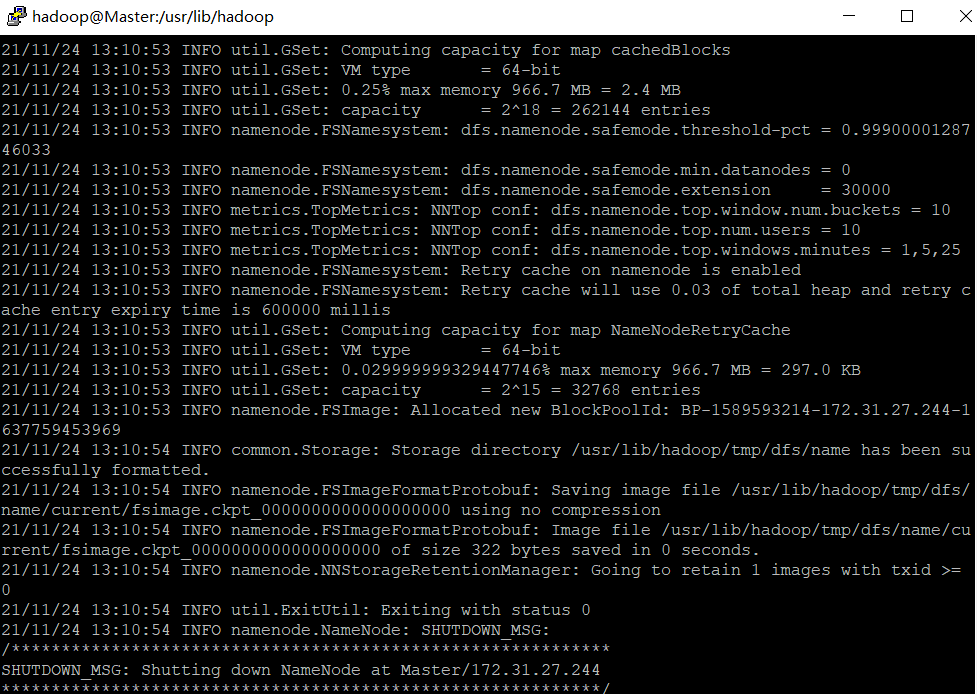


# 在hadoop文件夹下修改一些配置文件。然后退出来，将hadoop进行打包，传送给其他两个slave。

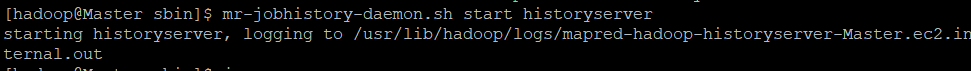


# 然后分别在两个slave节点安装hadoop，并赋予权限。（此处没截图）

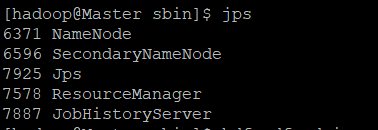
# 将hadoop初始化。可以看到succeed，并且status=0，执行成功。



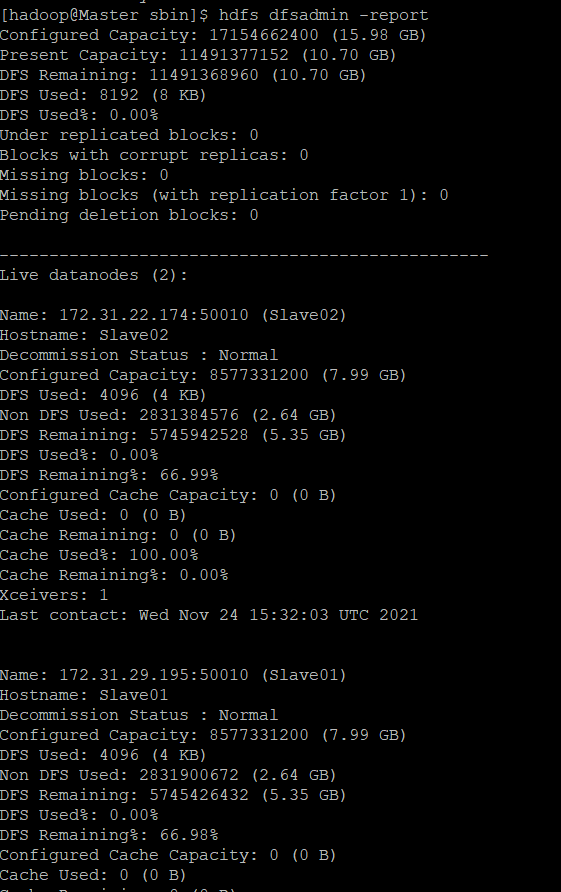
# 接着可以启动 Hadoop 和 Yarn 了，启动需要在 Master 节点上进行。然后顺利成功之后，执行语句打开jobhistory。



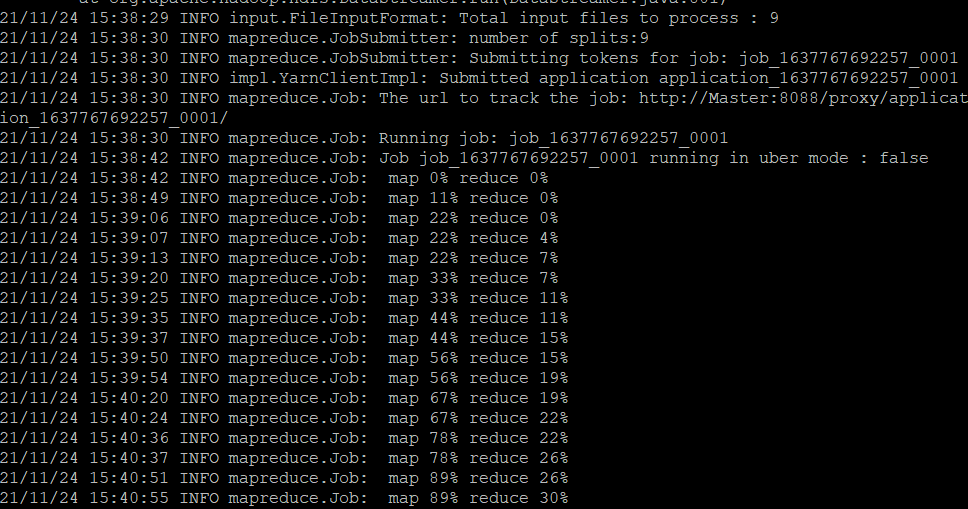
# 如下图状态，能看到jps下出现五个项，操作成功。



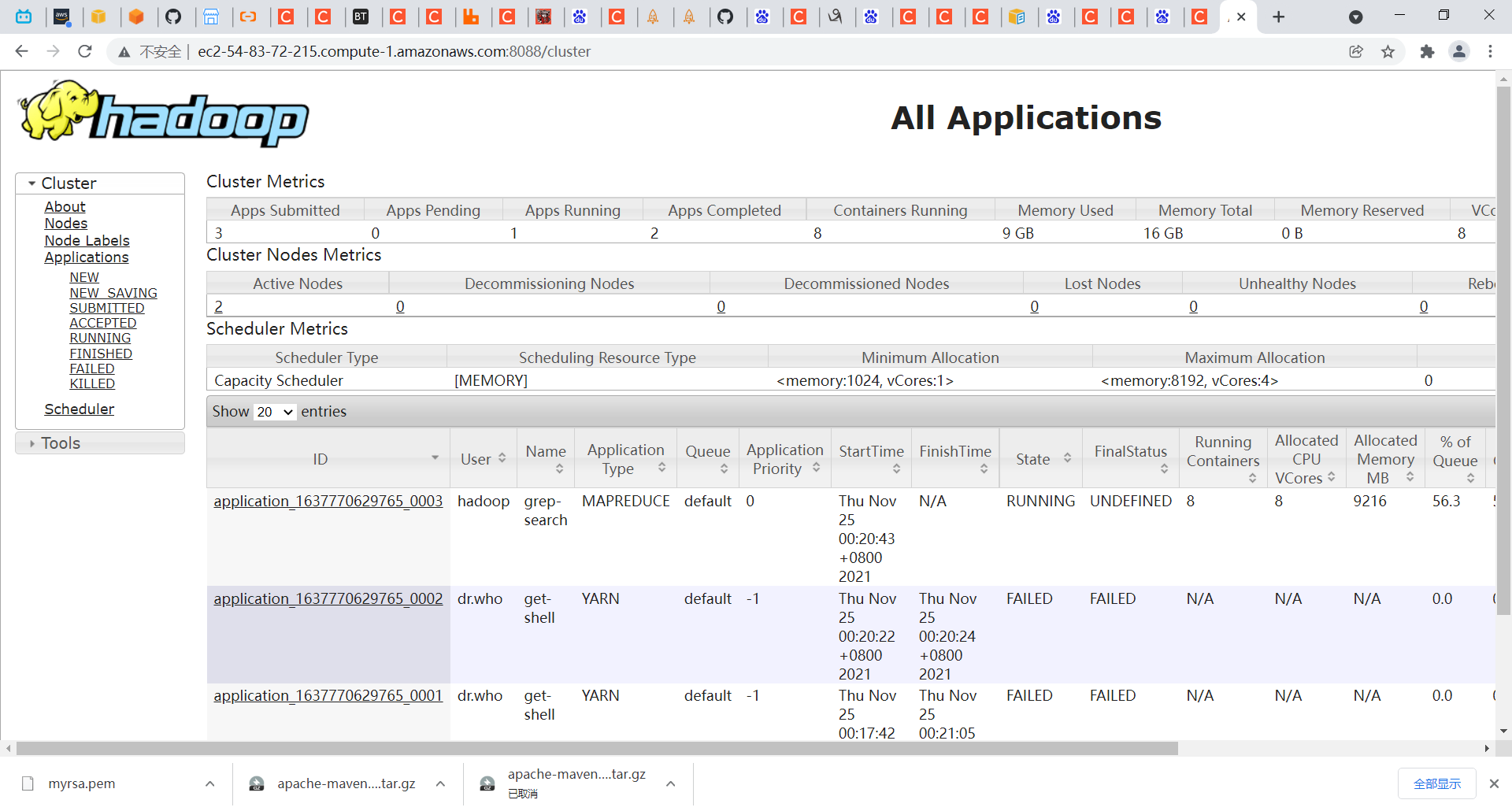
# 通过命令 hdfs dfsadmin -report 可查看集群状态，其中 Live datanodes (2) 表明两个从节点都已正常启动。

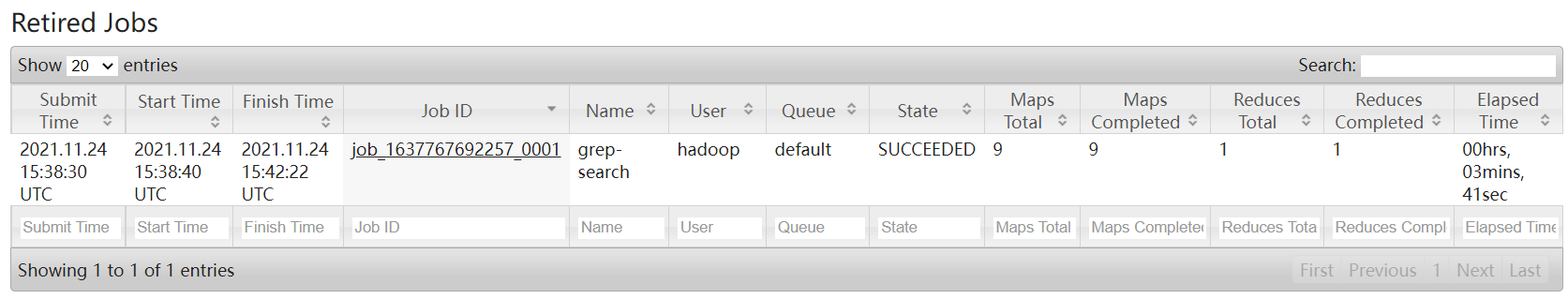


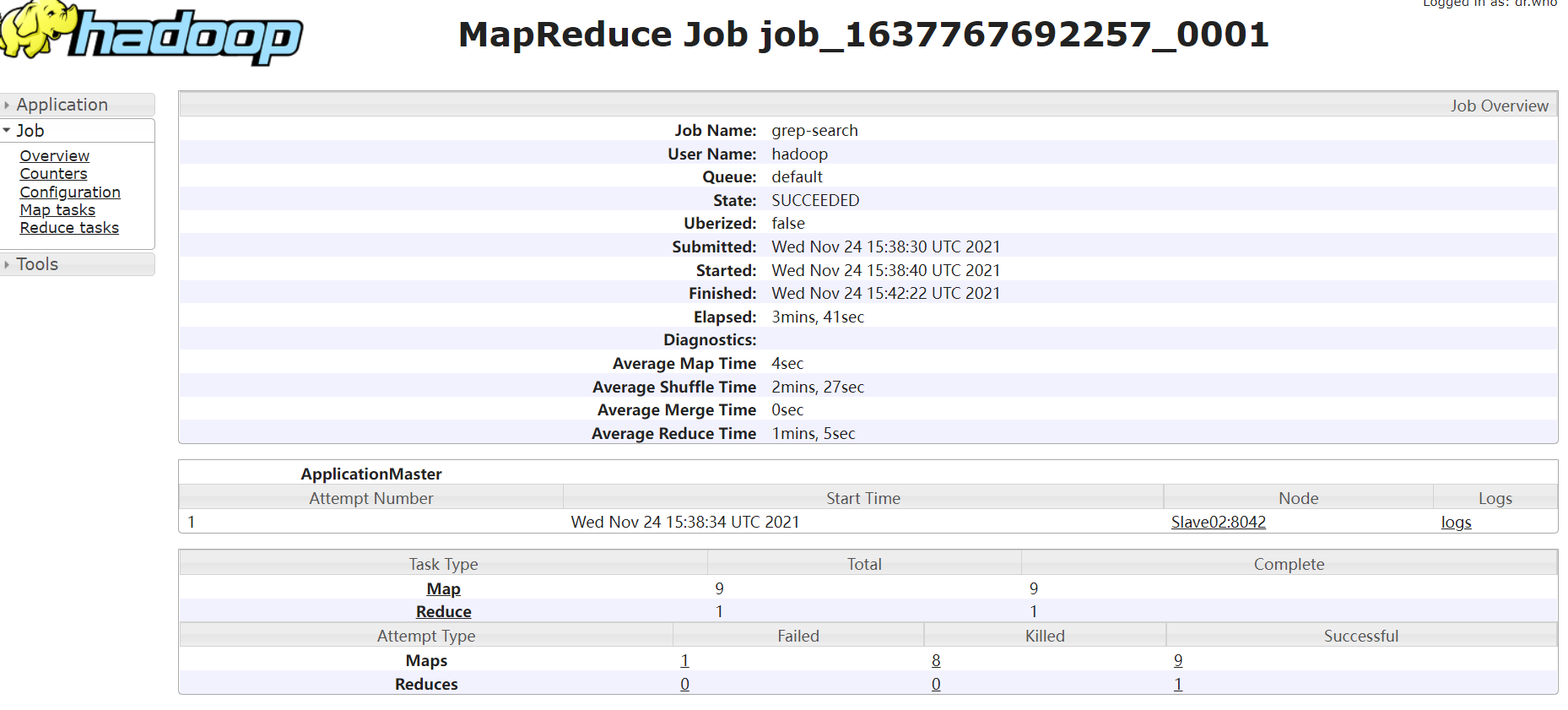
# 执行分布式实例，跑到了89% 30%。

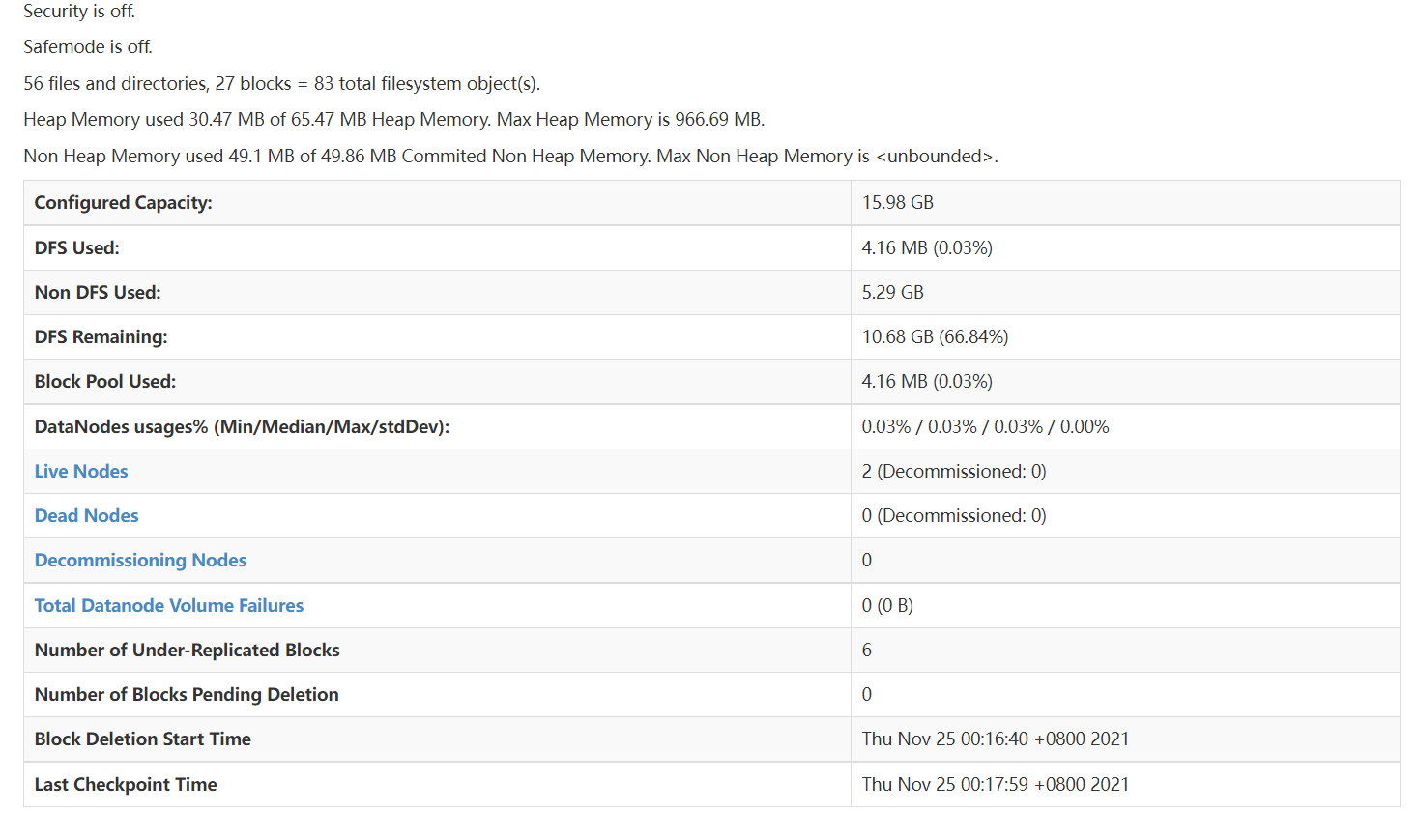


# 可以通过三个地址查看 hadoop 的 web UI，分别如下图。可以看到当前执行的进程，以及资源占用率等。









# 后面89%卡住 ，等太久了，我就先ctrl+c将他中断退出。然后去寻找原因，发现是内存的问题。这两种错误的本质是一样的，就是在运行jar包时，节点为期分配的内存不够，且也没有指定最大最小值。在集群的yarn-site.xml中添加如下配置：

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>4096</value>

</property>

<property>

<name>yarn.scheduler.minimum-allocation-mb</name>

<value>2048</value>

</property>

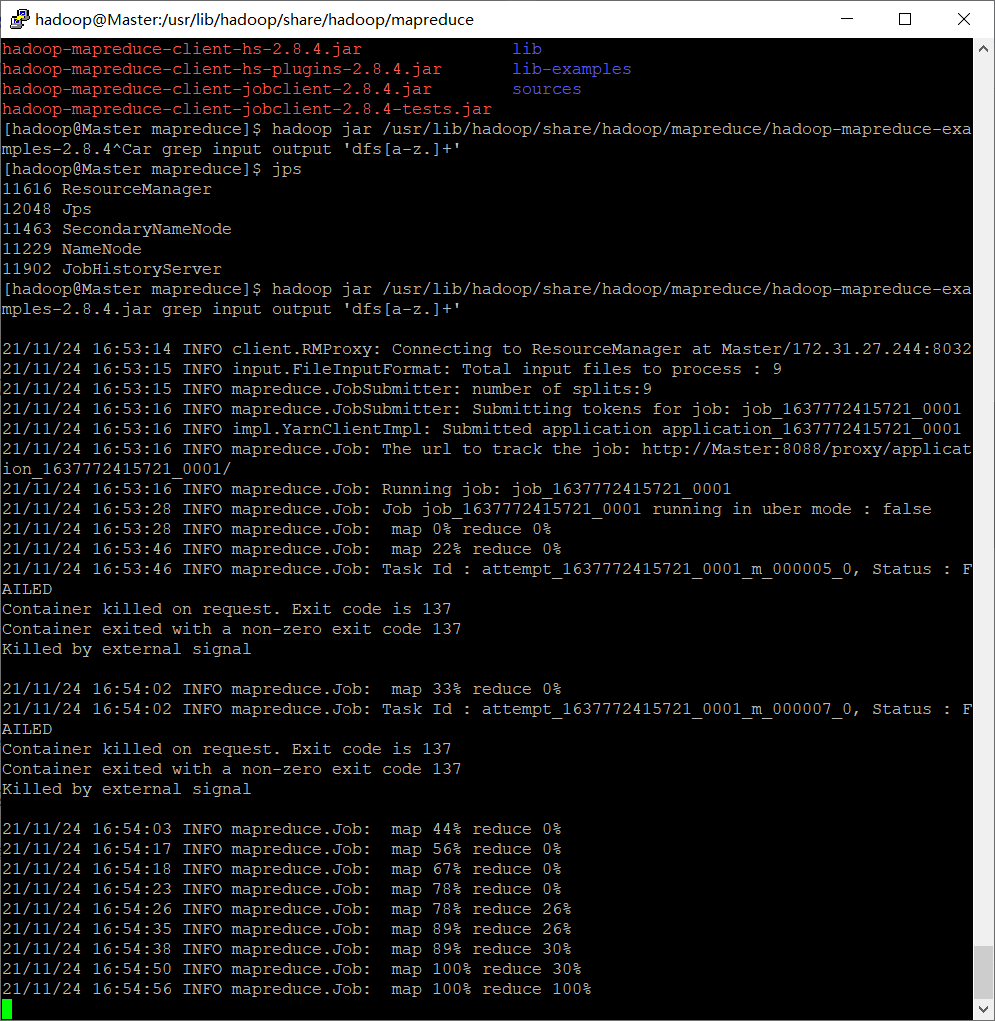
<property>

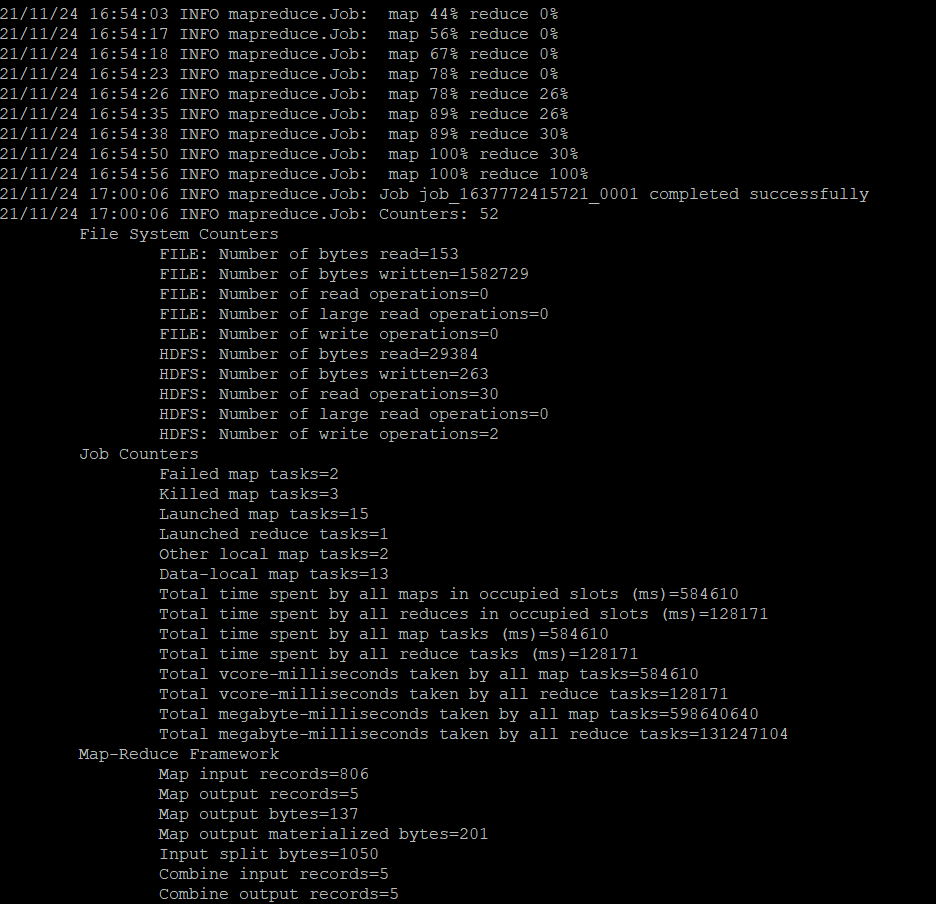
<name>yarn.nodemanager.vmem-pmem-ratio</name>

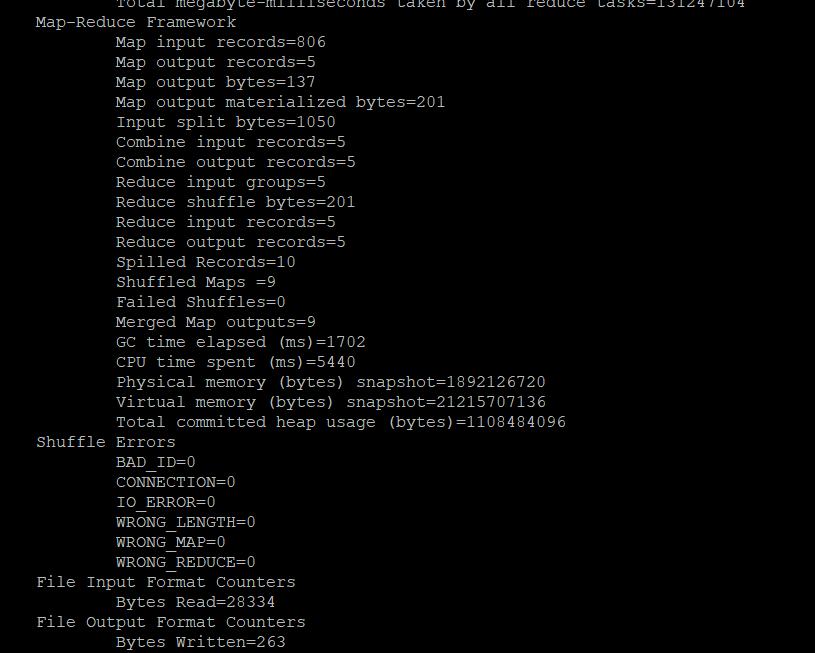
<value>2.1</value>

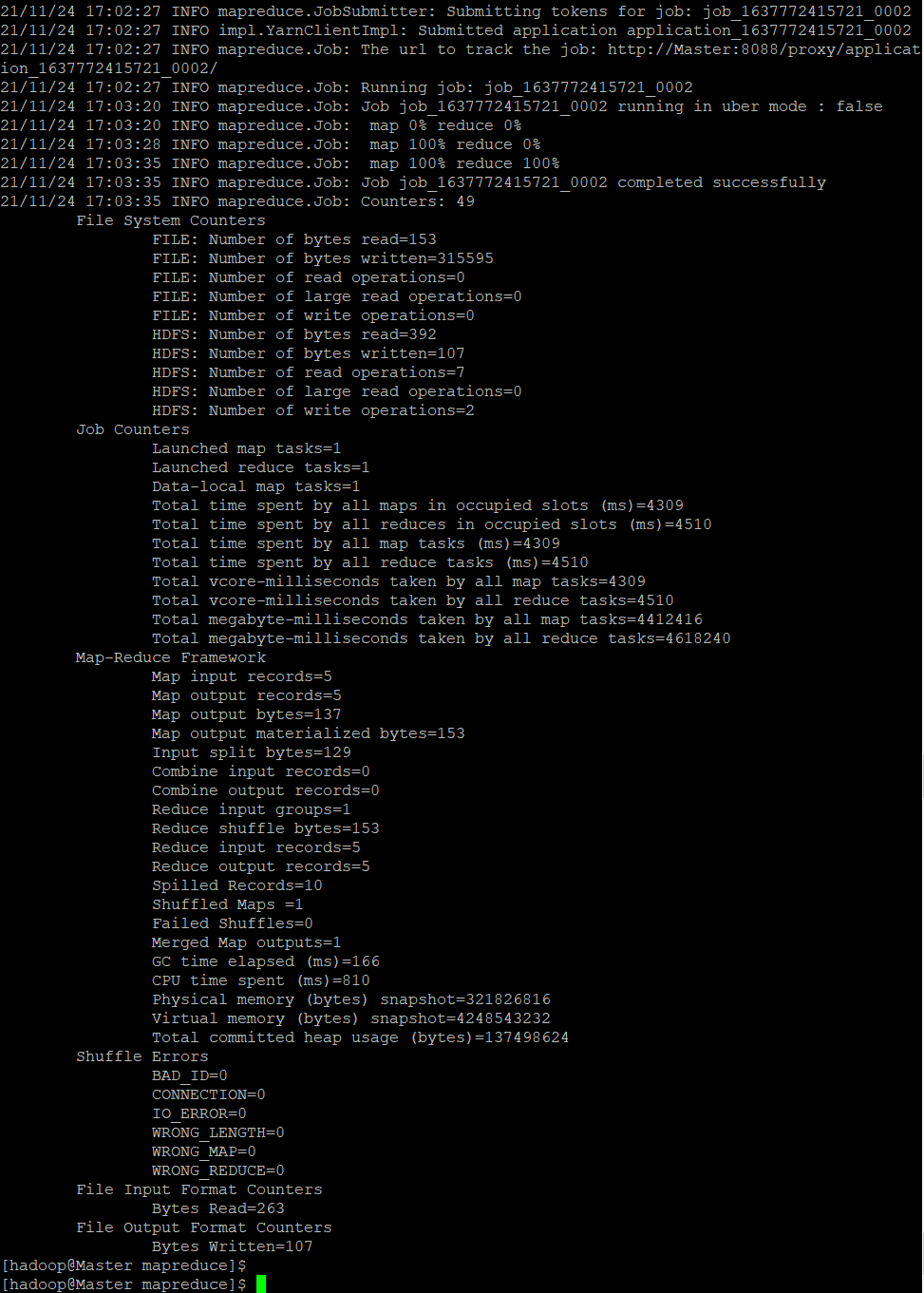
</property>

# 重新启动集群，运行jar包即可，可以看到已经达到了双100%。执行成功。Counters=52

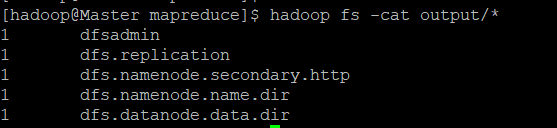








# 然后将output结果打印出来



# 执行完毕，进行关闭操作。

