CentOS 6.8 + Hadoop2.7.3

集群环境搭建

技术文档

（版本：V1.0.0）

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 说明 | 作者 |
| V1.0.0 | 2016/11/8 | 创建 | Simon Hoo |
|  |  |  |  |

目 录

一． 环境准备 - 3 -

1.1 服务器及配置 - 3 -

1.2 软件准备 - 4 -

二． 系统环境配置 - 4 -

2.1 系统设置 - 4 -

2.1.1 设置hostname - 4 -

2.1.2 创建用户 - 6 -

2.1.3 安装JDK - 6 -

2.1.4 设置SSH免密码登录 - 7 -

三． 安装Master (Name Node)节点 - 8 -

3.1 解压安装包 - 8 -

3.1.1 解压hadoop安装包 - 8 -

3.2 设置环境变量 - 8 -

3.3 修改$HADOOP\_HOME/etc/hadoop/\*-env.sh - 8 -

3.4 修改$HADOOP\_HOME /etc/hadoop/core-site.xml - 9 -

3.5 修改$HADOOP\_HOME /etc/hadoop/hdfs-site.xml - 9 -

3.6 修改$HADOOP\_HOME /etc/hadoop/yarn-site.xml - 10 -

3.7 修改$HADOOP\_HOME/etc/hadoop/mapred-site.xml - 11 -

3.8 修改$HADOOP\_HOME/etc/hadoop/slaves - 11 -

四． 安装Slave (DataNode)节点 - 11 -

4.1 复制Master上配置好的hadoop到Slave节点上 - 11 -

五． 格式化NameNode - 12 -

六． 启动Hadoop - 12 -

6.1 Master上启动 - 12 -

6.2 JPS检验是否成功启动 - 13 -

6.3 管理界面 - 13 -

七． 测试项目（提交一个MapReduce任务） - 15 -

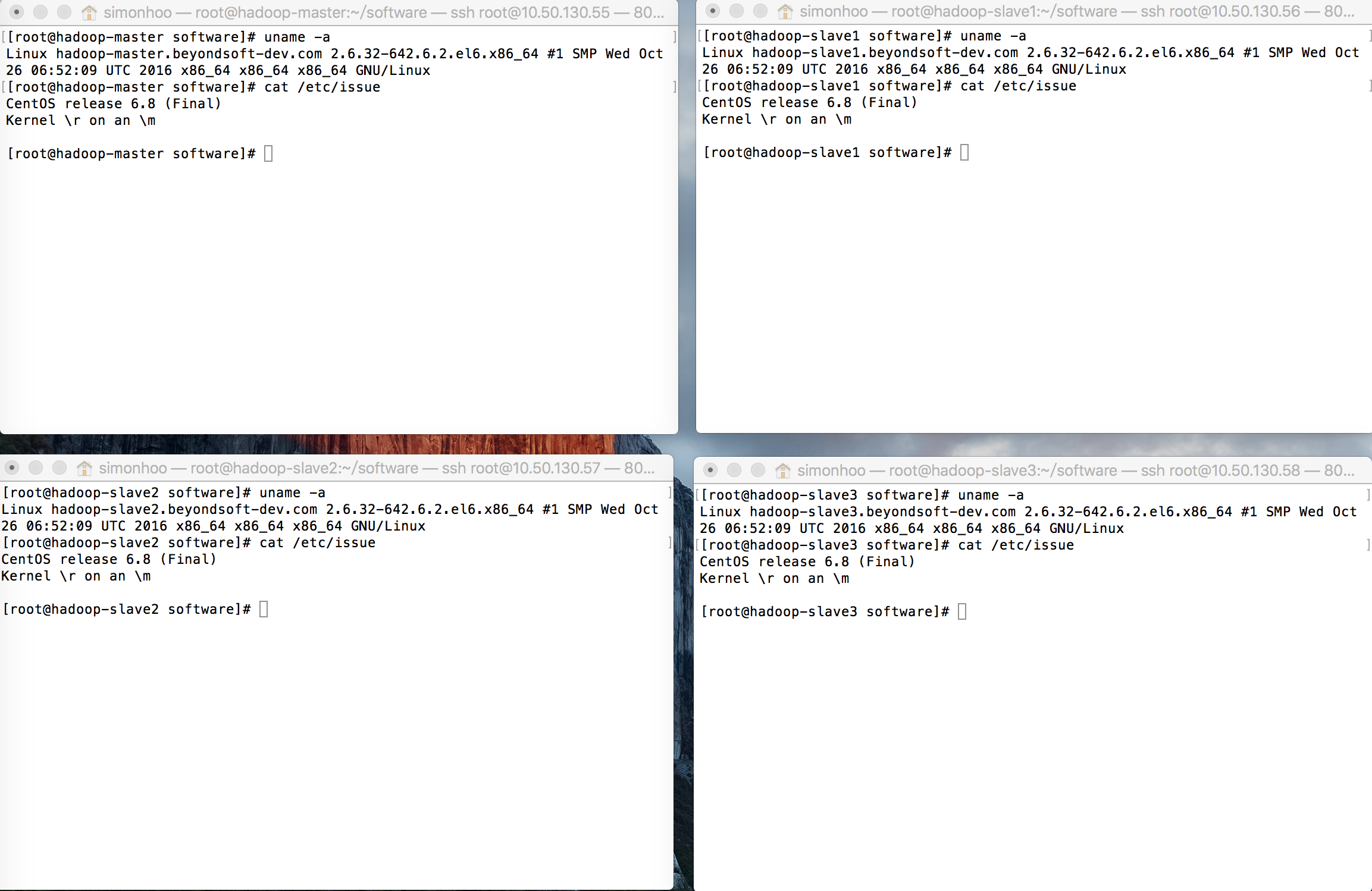
八． 应用程序接入 - 15 -

5.1 创建JAVA程序 - 15 -

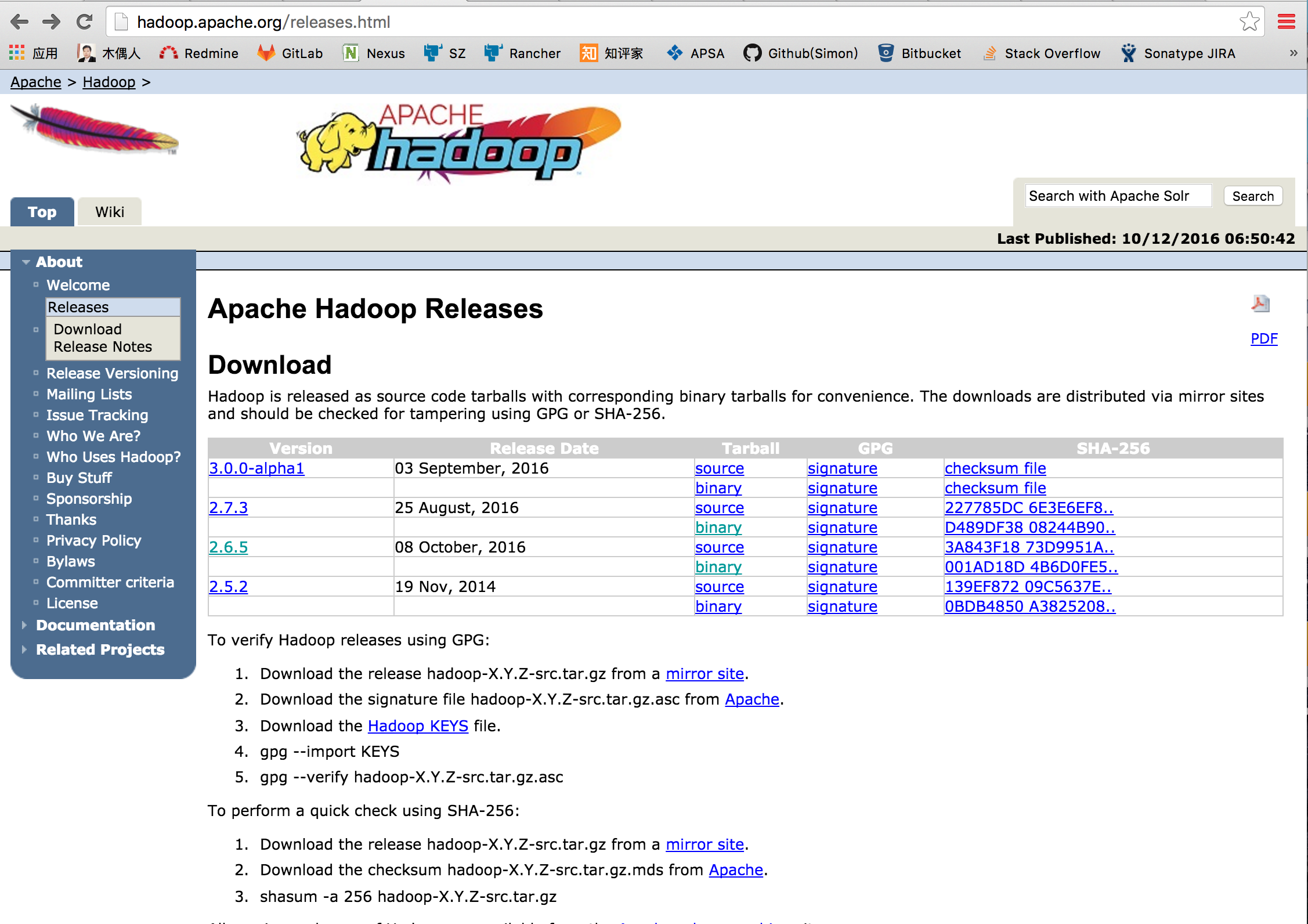
## 环境准备

### 服务器及配置

|  |  |  |  |
| --- | --- | --- | --- |
| 服务器 | IP地址 | 安装软件 | 备注 |
| Hadoop Master  (Name Node) | 10.50.130.55 | Hadoop 2.7.3  JDK 1.7 | CentOS 6.8 64位  CPU: 2x2CPUs  RAM: 8GB  Disk: 50GB |
| Hadoop Slave 1  (Data Node) | 10.50.130.56 | Hadoop 2.7.3  JDK 1.7 | CentOS 6.8 64位  CPU: 2x2CPUs  RAM: 8GB  Disk: 50GB |
| Hadoop Slave 2  (Data Node) | 10.50.130.57 | Hadoop 2.7.3  JDK 1.7 | CentOS 6.8 64位  CPU: 2x2CPUs  RAM: 8GB  Disk: 50GB |
| Hadoop Slave 3  (Data Node) | 10.50.130.58 | Hadoop 2.7.3  JDK 1.7 | CentOS 6.8 64位  CPU: 2x2CPUs  RAM: 8GB  Disk: 50GB |
|  |  |  |  |



### 软件准备



注：JDK自行在ORACLE官网下载。

## 系统环境配置

### 系统设置

#### 2.1.1 设置hostname

Master节点：

[root@localhost ~]# vi /etc/hosts

10.50.130.55 hadoop-master

10.50.130.56 hadoop-slave1

10.50.130.57 hadoop-slave2

10.50.130.58 hadoop-slave3

[root@localhost ~]# vi /etc/sysconfig/network

HOSTNAME = hadoop-master

[root@localhost ~]# vi /etc/selinux/config

SELINUX=disabled

[root@localhost ~]# reboot

Slave1节点：

[root@localhost ~]# vi /etc/hosts

10.50.130.55 hadoop-master

10.50.130.56 hadoop-slave1

10.50.130.57 hadoop-slave2

10.50.130.58 hadoop-slave3

[root@localhost ~]# vi /etc/sysconfig/network

HOSTNAME = hadoop-slave1

[root@localhost ~]# vi /etc/selinux/config

SELINUX=disabled

[root@localhost ~]# reboot

Slave2节点：

[root@localhost ~]# vi /etc/hosts

10.50.130.55 hadoop-master

10.50.130.56 hadoop-slave1

10.50.130.57 hadoop-slave2

10.50.130.58 hadoop-slave3

[root@localhost ~]# vi /etc/sysconfig/network

HOSTNAME = hadoop-slave2

[root@localhost ~]# vi /etc/selinux/config

SELINUX=disabled

[root@localhost ~]# reboot

Slave3节点：

[root@localhost ~]# vi /etc/hosts

10.50.130.55 hadoop-master

10.50.130.56 hadoop-slave1

10.50.130.57 hadoop-slave2

10.50.130.58 hadoop-slave3

[root@localhost ~]# vi /etc/sysconfig/network

HOSTNAME = hadoop-slave3

[root@localhost ~]# vi /etc/selinux/config

SELINUX=disabled

[root@localhost ~]# reboot

#### 2.1.2 创建用户

[root@hadoop-master ~]# useradd hadoop

[root@hadoop-master ~]# passwd Hadoop

#### 2.1.3 安装JDK

[root@hadoop-master ~]# mkdir –p /usr/local/java

[root@hadoop-master ~]# cd /root/software

[root@hadoop-master software]# tar –xzvf jdk-7u79-linux-x64.tar.gz –C /usr/local/java

设置JAVA\_HOME:

[root@hadoop-master ~]# vi /etc/profile

export JAVA\_HOME=/usr/local/java/jdk1.7.0\_79

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA \_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

[root@hadoop-master ~]# source /etc/profile

[root@hadoop-master ~]# java -version

[root@hadoop-master ~]# chown -R hadoop.hadoop /usr/local/java/

#### 2.1.4 设置SSH免密码登录

Master节点：

[root@hadoop-master ~]# su hadoop

[hadoop@hadoop-master ~]$ ssh-keygen -t rsa -P '' –f ~/.ssh/id\_rsa

[hadoop@hadoop-master ~]$ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

[hadoop@hadoop-master ~]$ chmod 600 ~/.ssh/authorized\_keys

复制到Slave节点：

[hadoop@hadoop-master ~]$ scp ~/.ssh/id\_rsa.pub [hadoop@10.50.130.56:/tmp/authorized\_keys](mailto:hadoop@10.50.130.56:/tmp/authorized_keys)

[hadoop@hadoop-master ~]$ scp ~/.ssh/id\_rsa.pub [hadoop@10.50.130.57:/tmp/authorized\_keys](mailto:hadoop@10.50.130.57:/tmp/authorized_keys)

[hadoop@hadoop-master ~]$ scp ~/.ssh/id\_rsa.pub [hadoop@10.50.130.58:/tmp/authorized\_keys](mailto:hadoop@10.50.130.58:/tmp/authorized_keys)

在Slave节点追加 /tmp/authorized\_keys 到 ~/.ssh/authorized\_keys：

[hadoop@hadoop-slave1 ~]$ cat /tmp/authorized\_keys >> ~/.ssh/authorized\_keys

[hadoop@hadoop-slave1 ~]$ chmod 700 ~/.ssh/

[hadoop@hadoop-slave1 ~]$ chmod 600 ~/.ssh/authorized\_keys

[hadoop@hadoop-slave2 ~]$ cat /tmp/authorized\_keys >> ~/.ssh/authorized\_keys

[hadoop@hadoop-slave2 ~]$ chmod 700 ~/.ssh/

[hadoop@hadoop-slave2 ~]$ chmod 600 ~/.ssh/authorized\_keys

[hadoop@hadoop-slave3 ~]$ cat /tmp/authorized\_keys >> ~/.ssh/authorized\_keys

[hadoop@hadoop-slave3 ~]$ chmod 700 ~/.ssh/

[hadoop@hadoop-slave3 ~]$ chmod 600 ~/.ssh/authorized\_keys

在Master节点测试SSH免登录到Slave节点：

[hadoop@hadoop-master ~]$ ssh hadoop-master

[hadoop@hadoop-master ~]$ ssh hadoop-slave1

[hadoop@hadoop-master ~]$ ssh hadoop-slave2

[hadoop@hadoop-master ~]$ ssh hadoop-slave3

当然值得注意的是：首次登陆是需要确认的，hadoop-slave1(1~3)结点首次连接时需要，“YES”确认连接，这意味着hadoop-master结点连接hadoop-slave1(1~3)结点时需要人工询问，无法自动连接，输入yes后成功接入，紧接着注销退出至hadoop-master结点。要实现ssh免密码连接至其它结点，还差一步，只需要再执行一遍ssh hadoop-slave(1~3)，如果没有要求你输入”yes”，就算成功了。

## 安装Master (Name Node)节点

### 解压安装包

#### 3.1.1 解压hadoop安装包

[root@hadoop-master ~]# cd /software

[root@hadoop-master ~]# tar –xzvf hadoop-2.7.3.tar.gz

[root@hadoop-master ~]# cp –r hadoop-2.7.3 /usr/local/hadoop

[root@hadoop-master ~]# chown -R hadoop.hadoop /usr/local/hadoop/

### 设置环境变量

[root@hadoop-master ~]# vi /etc/profile

# Setting HADOOP env

HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

[root@hadoop-master ~]# source /etc/profile

### 修改$HADOOP\_HOME/etc/hadoop/\*-env.sh

[root@hadoop-master ~]# cd $HADOOP\_HOME/etc/hadoop

[root@hadoop-master hadoop]# vi hadoop-env.sh

# The java implementation to use.

#export JAVA\_HOME=${JAVA\_HOME}

export JAVA\_HOME=/usr/local/java/jdk1.7.0\_79

[root@hadoop-master hadoop]# vi mapred-env.sh

export JAVA\_HOME=/usr/local/java/jdk1.7.0\_79

[root@hadoop-master hadoop]# vi yarn-env.sh

export JAVA\_HOME=/usr/local/java/jdk1.7.0\_79

### 修改$HADOOP\_HOME /etc/hadoop/core-site.xml

[root@hadoop-master ~]# cd $HADOOP\_HOME/etc/hadoop

[root@hadoop-master ~]# vi core-site.xml

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>file:/usr/local/hadoop/tmp</value>

<description>Abase for other temporary directories.</description>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoop-master:9000</value>

</property>

</configuration>

### 修改$HADOOP\_HOME /etc/hadoop/hdfs-site.xml

[root@hadoop-master ~]# cd $HADOOP\_HOME/etc/hadoop

[root@hadoop-master ~]# vi hdfs-site.xml

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>hadoop-master:50090</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop/tmp/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop/tmp/dfs/data</value>

</property>

</configuration>

### 修改$HADOOP\_HOME /etc/hadoop/yarn-site.xml

[root@hadoop-master ~]# cd $HADOOP\_HOME/etc/hadoop

[root@hadoop-master ~]# vi yarn-site.xml

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>hadoop-master:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>hadoop-master:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>hadoop-master:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>hadoop-master:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>hadoop-master:8088</value>

</property>

</configuration>

### 修改$HADOOP\_HOME/etc/hadoop/mapred-site.xml

[root@hadoop-master ~]# cd $HADOOP\_HOME/etc/hadoop

[root@hadoop-master ~]# vi mapred-site.xml

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>hadoop-master:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>hadoop-master:19888</value>

</property>

</configuration>

### 修改$HADOOP\_HOME/etc/hadoop/slaves

[root@hadoop-master ~]# cd $HADOOP\_HOME/etc/hadoop

[root@hadoop-master ~]# vi slaves

hadoop-slave1

hadoop-slave2

hadoop-slave3

## 安装Slave (DataNode)节点

### 复制Master上配置好的hadoop到Slave节点上

[hadoop@hadoop-master ~]$ cd /usr/local

[hadoop@hadoop-master local]$ scp –r hadoop root@hadoop-slave1:/usr/local

[hadoop@hadoop-master local]$ scp –r hadoop root@hadoop-slave2:/usr/local

[hadoop@hadoop-master local]$ scp –r hadoop root@hadoop-slave3:/usr/local\

修改Slave节点上的文件归属：

[root@hadoop-slave1 ~]# chown -R hadoop.hadoop /usr/local/hadoop/

[root@hadoop-slave2 ~]# chown -R hadoop.hadoop /usr/local/hadoop/

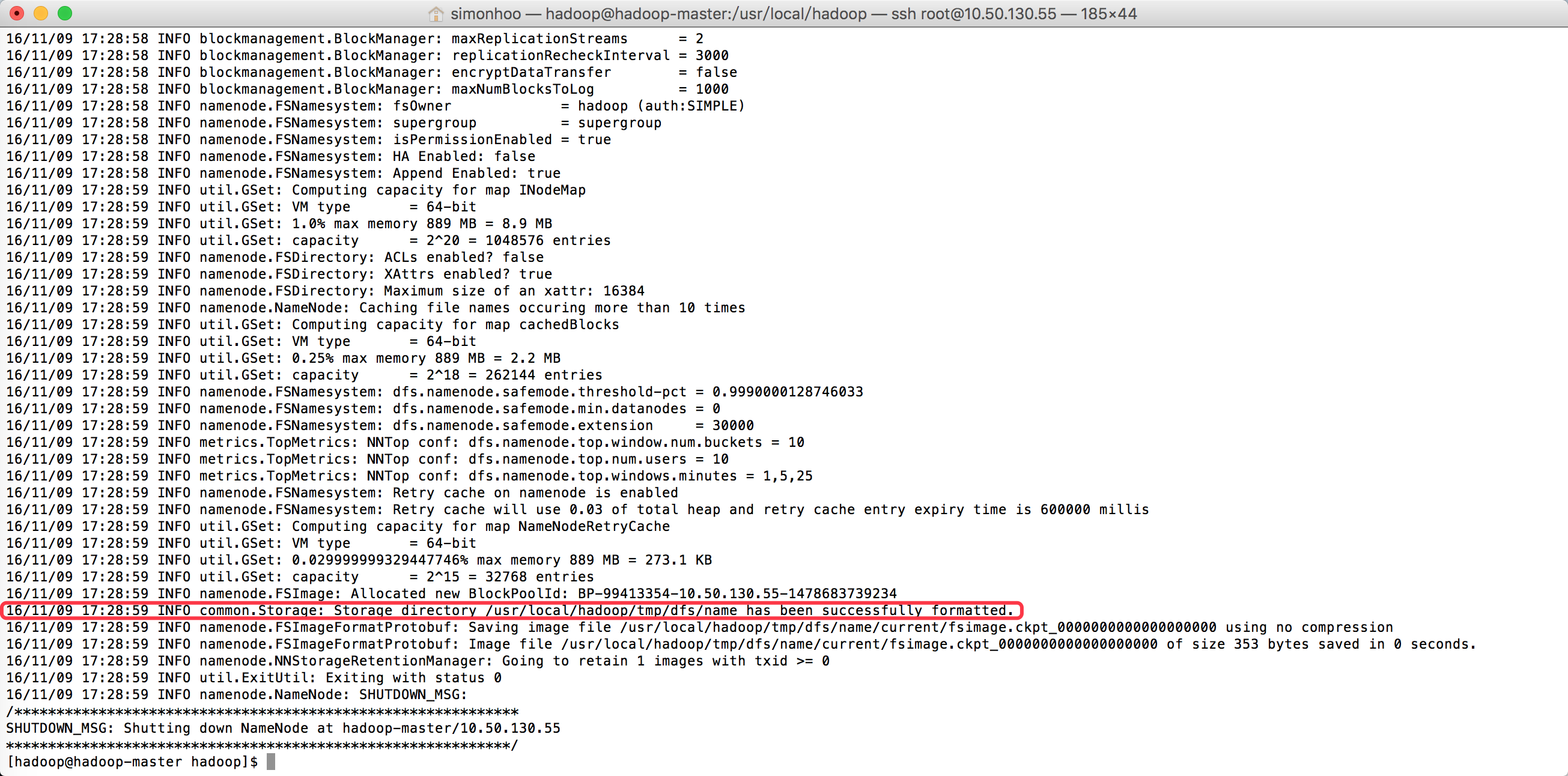
[root@hadoop-slave3 ~]# chown -R hadoop.hadoop /usr/local/hadoop/

## 格式化NameNode

hadoop登录到hadoop-master:

[hadoop@hadoop-master ~]$ cd /usr/local/hadoop

[hadoop@hadoop-master hadoop]$ ./bin/hdfs namenode –format



注意：在你重新格式化分布式文件系统之前，需要将文件系统中的数据先清除，否则，datanode将创建不成功，这一点很重要。

## 启动Hadoop

### Master上启动

hadoop登录到hadoop-master:

[hadoop@hadoop-master ~]$ cd /usr/local/hadoop

[hadoop@hadoop-master hadoop]$ ./sbin/start-all.sh

### JPS检验是否成功启动

hadoop-master:

[hadoop@hadoop-master ~]$ jps

1866 SecondaryNameNode

1671 NameNode

2023 ResourceManager

2290 Jps

hadoop-slave1:

[hadoop@hadoop-slave1 ~]$ jps

1611 Jps

1372 DataNode

1474 NodeManager

hadoop-slave2:

[hadoop@hadoop-slave2 ~]$ jps

1356 DataNode

1458 NodeManager

1595 Jps

hadoop-slave3:

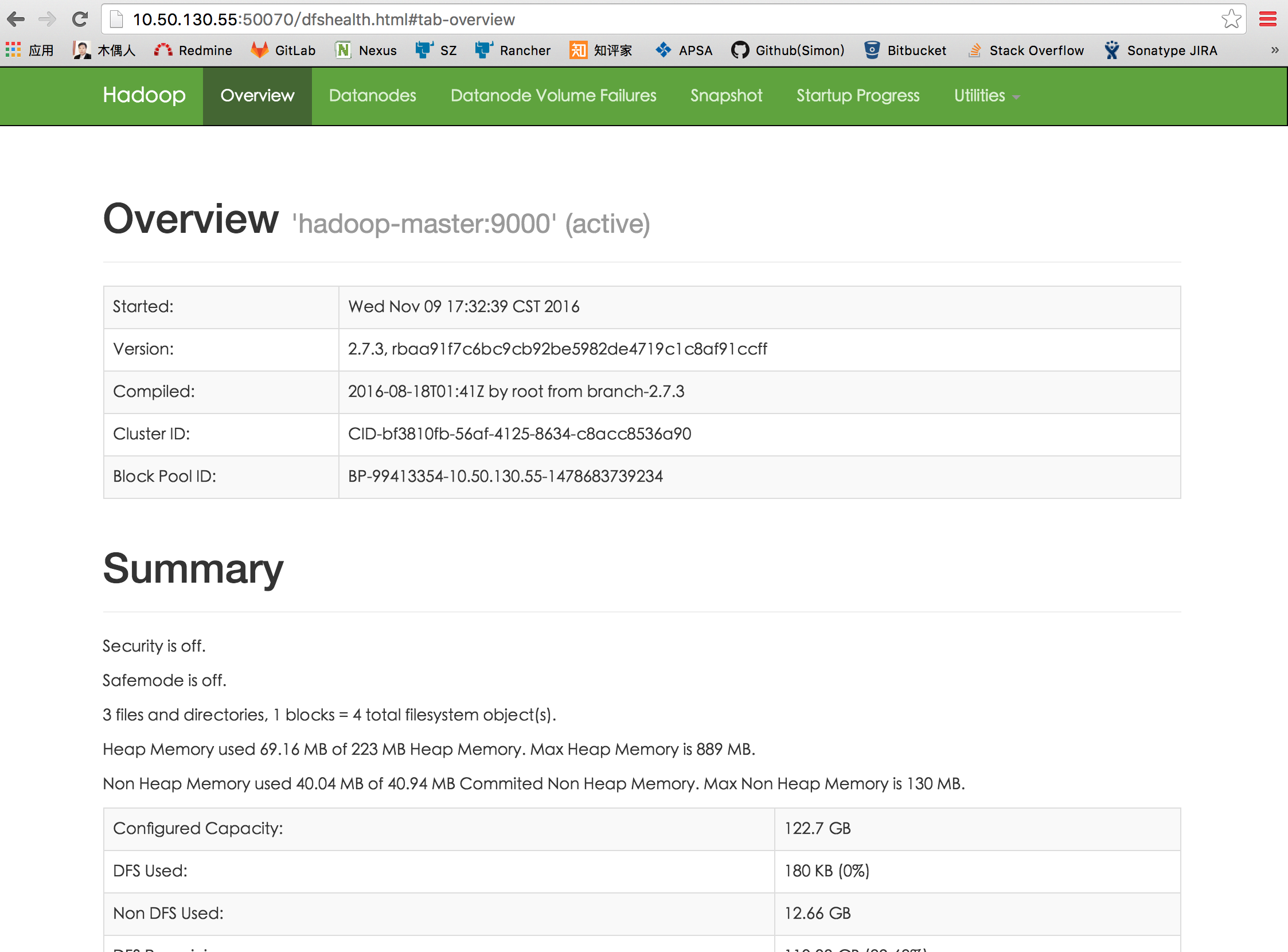
[hadoop@hadoop-slave3 ~]$ jps

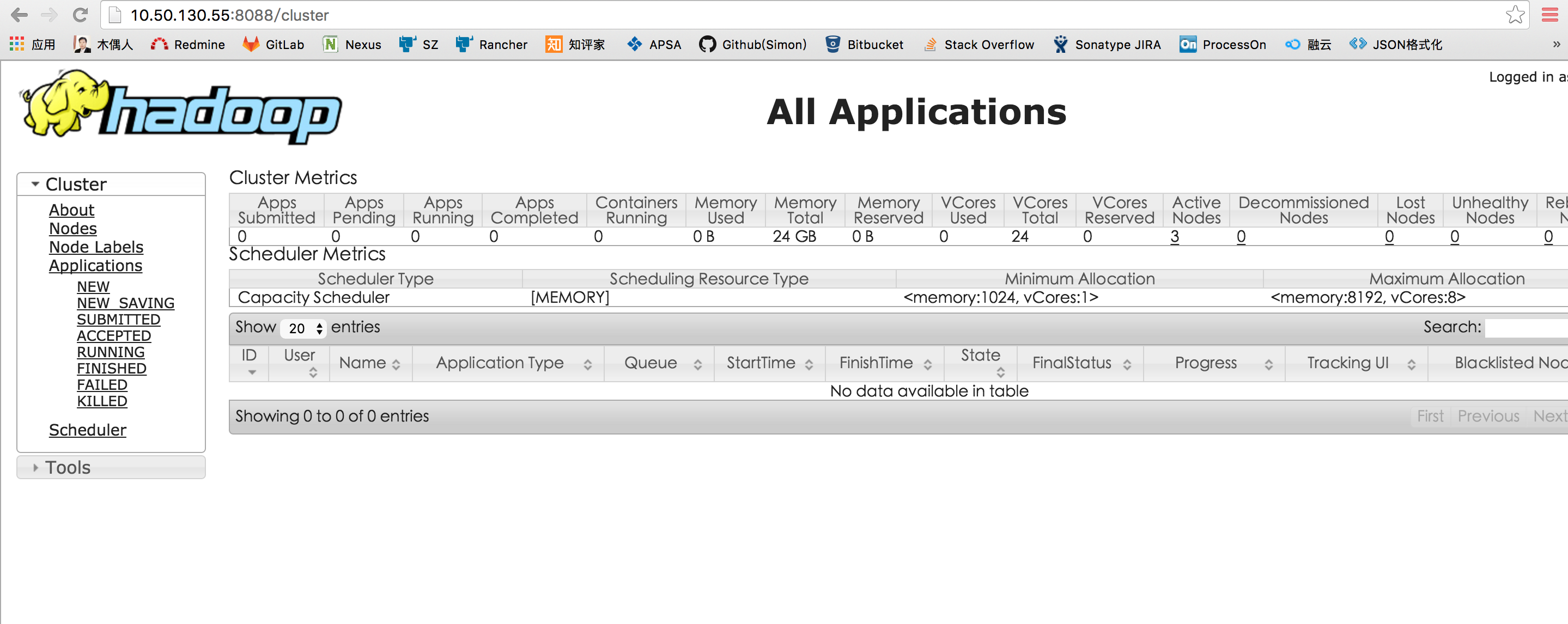
1592 Jps

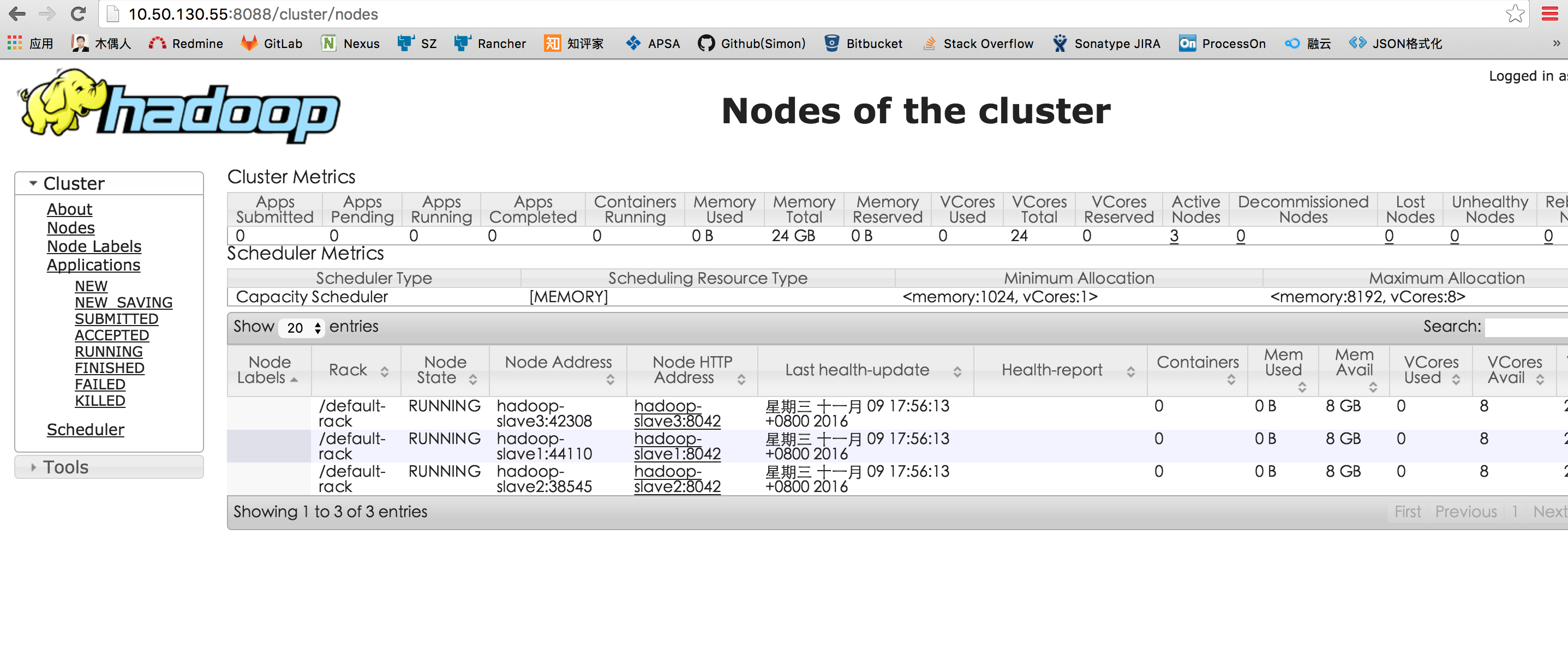
1353 DataNode

1455 NodeManager

### 管理界面







## 测试项目（提交一个MapReduce任务）

创建测试项目：

[hadoop@hadoop-master ~]$ cd /usr/local/Hadoop

[hadoop@hadoop-master hadoop]$ ./bin/hdfs dfs -mkdir /test-project

复制一份文件到分布式文件系统中：

[hadoop@hadoop-master hadoop]$ ./bin/hdfs dfs -copyFromLocal ./LICENSE.txt /test-project

查看在分布式文件系统中是否复制成功：

[hadoop@hadoop-master hadoop]$ ./bin/hdfs dfs -ls /test-project

16/11/09 17:49:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 1 items

-rw-r--r-- 1 hadoop supergroup 84854 2016-11-09 17:48 /test-project/LICENSE.txt

## 应用程序接入

### 创建JAVA程序