Spark集群安装

整体流程：

- 配置3台虚拟机，一个master两个slave
- master配置好hadoop，拷贝到两个slave
- master配置hadoop集群并启动
- master配置Spark，拷贝到两个slave
- 配置并启动Spark集群

虚拟机配置

在VirtualBox中配置3台虚拟机，采用Ubuntu 16.04

安装流程可以完全参考：https://blog.csdn.net/scene_2015/article/details/83025750

Ubuntu 镜像：

中科大源

http://mirrors.ustc.edu.cn/ubuntu-releases/16.04/

阿里云开源镜像站

http://mirrors.aliyun.com/ubuntu-releases/16.04/

兰州大学开源镜像站

http://mirror.lzu.edu.cn/ubuntu-releases/16.04/

北京理工大学开源

http://mirror.bit.edu.cn/ubuntu-releases/16.04/

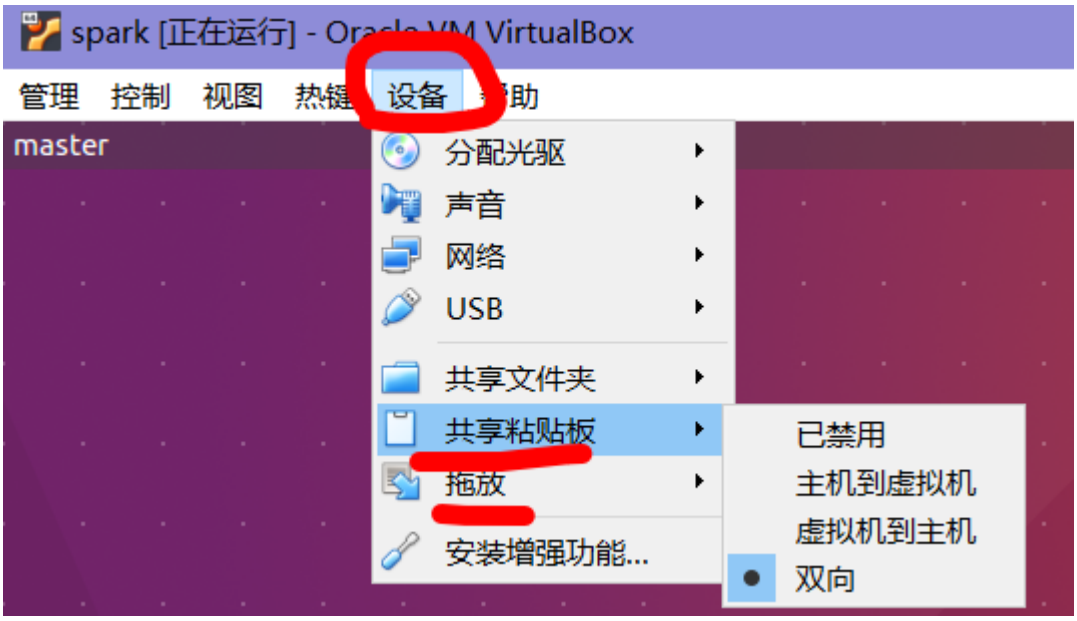
浙江大学

http://mirrors.zju.edu.cn/ubuntu-releases/16.04/

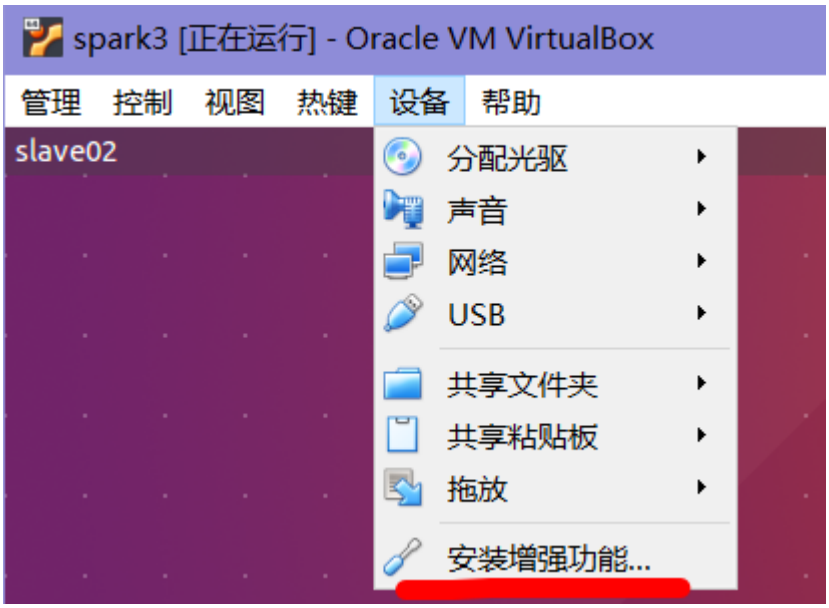
安装VirtualBox Guest Additions

虚拟机不方便进行复制粘贴和文件传递，需要安装VirtualBox Guest Additions

打开虚拟机后，在左上打开“设备”，将共享粘贴板、拖放均选为“双向”

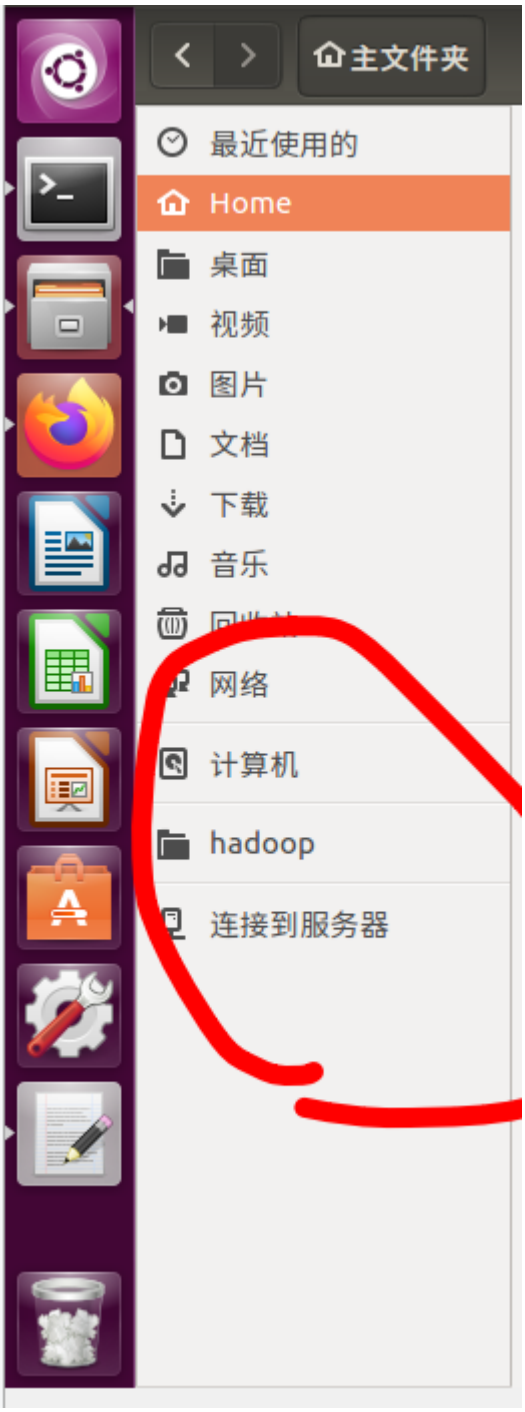


网上有很多下载镜像手动安装的，但是问题很多……最好按如下方法，直接选择安装增强功能，VBox有自带的VirtualBox Guest Additions镜像。



可能遇到报错：

这时检查文件系统有无可弹出的镜像，全部弹出



成功安装后重启系统。

Hadoop单机配置

版本：Hadoop2.7.1

（大部分参考http://dblab.xmu.edu.cn/blog/install-hadoop/，下面**只写了不适合、可以改进、需要补充的部分**

该文档适用于Hadoop2.x，但是Hadoop3.x的配置大同小异，只有默认端口不同，

端口变化参考：https://blog.csdn.net/qq_31454379/article/details/105439752）：

- **不同虚拟机的用户名最好一致，方便后面设置,参考文档中用户名统一为hadoop，用其他名称则要在部分位置修改文档中的指令**
- 安装vim

- ssh免密登陆配置
    - spark和hadoop需要在主机之间大量交换文件，因此要有免密登陆
- jdk安装
    - 可以直接 sudo apt-get install default-jdk
    - 编辑~/.bashrc文件，添加如下内容：
    - ```

      export JAVA_HOME=/usr/lib/jvm/default-java

      ```
    - 让环境变量生效： source ~/.bashrc
- Hadoop配置
    - 镜像下载：http://mirror.bit.edu.cn/apache/hadoop/common/
        - 推荐直接在虚拟机里下载，否则需要虚拟机和宿主创建一个共享文件夹，将宿主下载的镜像给虚拟机使用
    - **Hadoop单机配置、伪分布式部分不是必须，在之后的集群配置中会重新配置。如果做了的话，记得之后重新配置**

      sudo tar -zxf （下载地址）/hadoop-2.7.1.tar.gz -C /usr/local # 解压到/usr/local中 cd /usr/local/ sudo mv ./hadoop-2.7.1/ ./hadoop # 将文件夹名改为hadoop sudo chown -R 用户名 ./hadoop # 修改文件权限

修改XML

修改core-site.xml

```
<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://master:9000</value>
    </property>
</configuration>
```
```

修改hdfs-site.xml：

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</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mapred-site.xml(复制mapred-site.xml.template,再修改文件名)

```
  <configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
  </configuration>
```

修改yarn-site.xml

```
 <configuration>
  <!-- Site specific YARN configuration properties -->
      <property>
          <name>yarn.nodemanager.aux-services</name>
          <value>mapreduce_shuffle</value>
      </property>
      <property>
          <name>yarn.resourcemanager.hostname</name>
          <value>master</value>
      </property>
  </configuration>
```

Hadoop集群

大部分参考http://dblab.xmu.edu.cn/blog/1177-2/

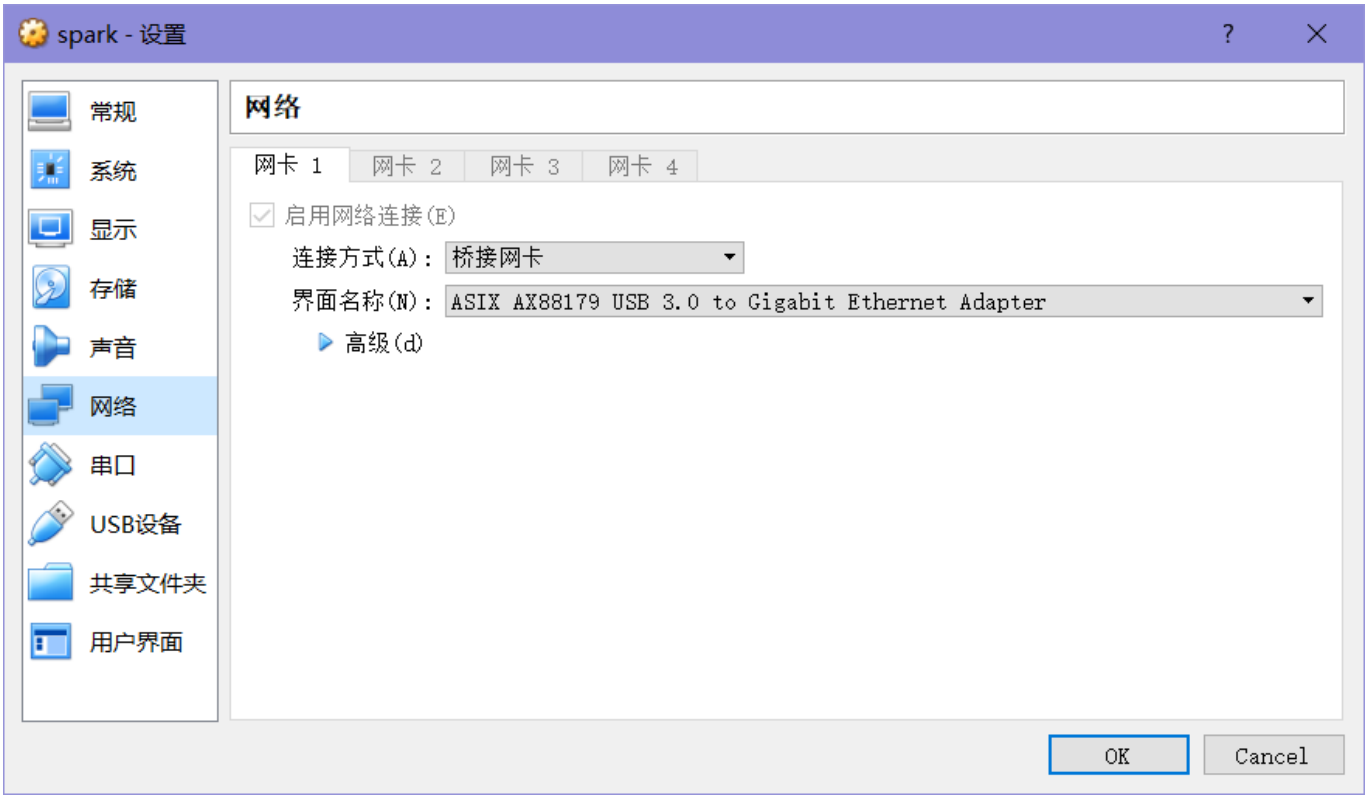
xml配置文件的详细意义可以参考官方文档，或https://blog.csdn.net/qq_25542879/article/details/89554068

部分文件修改需要root权限，记得sudo，如果忘记加权限，可以强制修改文件：
https://blog.csdn.net/u010713545/article/details/46482837

网络设置

首先要设置虚拟机网络，**使虚拟机之间可以互联**，在“设备”->“网络”中设置“桥接网卡”模式：

https://blog.csdn.net/bifengmiaozhuan/article/details/79887692



用ifconfig查看每个虚拟机ip

先设置好三台虚拟机 /etc/hostname 中的主机名（分别为master、slave01、slave02）

然后设置三台虚拟机 /etc/hosts 加入主机名和对应ip，例如：

```
172.17.177.70 master
172.17.186.11 slave01
172.17.159.239 slave02
```

现在应该相互可以ping通：

```
xzy@master:~$ ping slave01
PING slave01 (172.17.186.11) 56(84) bytes of data.
64 bytes from slave01 (172.17.186.11): icmp_seq=1 ttl=64 time=0.346 ms
64 bytes from slave01 (172.17.186.11): icmp_seq=2 ttl=64 time=0.416 ms
64 bytes from slave01 (172.17.186.11): icmp_seq=3 ttl=64 time=0.328 ms
64 bytes from slave01 (172.17.186.11): icmp_seq=4 ttl=64 time=0.522 ms
```

SSH

见http://dblab.xmu.edu.cn/blog/1177-2/

Hadoop配置

见http://dblab.xmu.edu.cn/blog/1177-2/即可，注意之前配置过单机Hadoop的话，core-site.xml一定要把下面的localhost修改成master节点的主机名

```
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
</property>
```

否则slave01和slave02无法访问master的9000端口，log中产生如下报错，之后出现jps能看到datanode，webUI却显示没有datanode的bug

```
RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2020-10-09 19:48:14,107 WARN org.apache.hadoop.hdfs.server.datanode.DataNode: Problem connecting
to server: master/172.17.177.70:9000
2020-10-09 19:48:20,110 INFO org.apache.hadoop.ipc.Client: Retrying connect to server:
master/172.17.177.70:9000. Already tried 0 time(s); retry policy is
RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2020-10-09 19:48:21,111 INFO org.apache.hadoop.ipc.Client: Retrying connect to server:
master/172.17.177.70:9000. Already tried 1 time(s); retry policy is
RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2020-10-09 19:48:22,113 INFO org.apache.hadoop.ipc.Client: Retrying connect to server:
master/172.17.177.70:9000. Already tried 2 time(s); retry policy is
RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
```

在 master 节点主机上执行：

```
cd /usr/local/rm -rf ./hadoop/tmp    # 删除临时文件rm -rf
./hadoop/logs/*   # 删除日志文件
tar -zcf ~/hadoop.master.tar.gz ./hadoop
cd ~
scp ./hadoop.master.tar.gz slave01:/home/用户名
scp ./hadoop.master.tar.gz slave02:/home/用户名
```

在slave01,slave02节点上执行：

```
sudo rm -rf /usr/local/hadoop/ #删掉原来的hadoop
sudo tar -zxf ~/hadoop.master.tar.gz -C /usr/local
sudo chown -R 用户名 /usr/local/hadoop
```

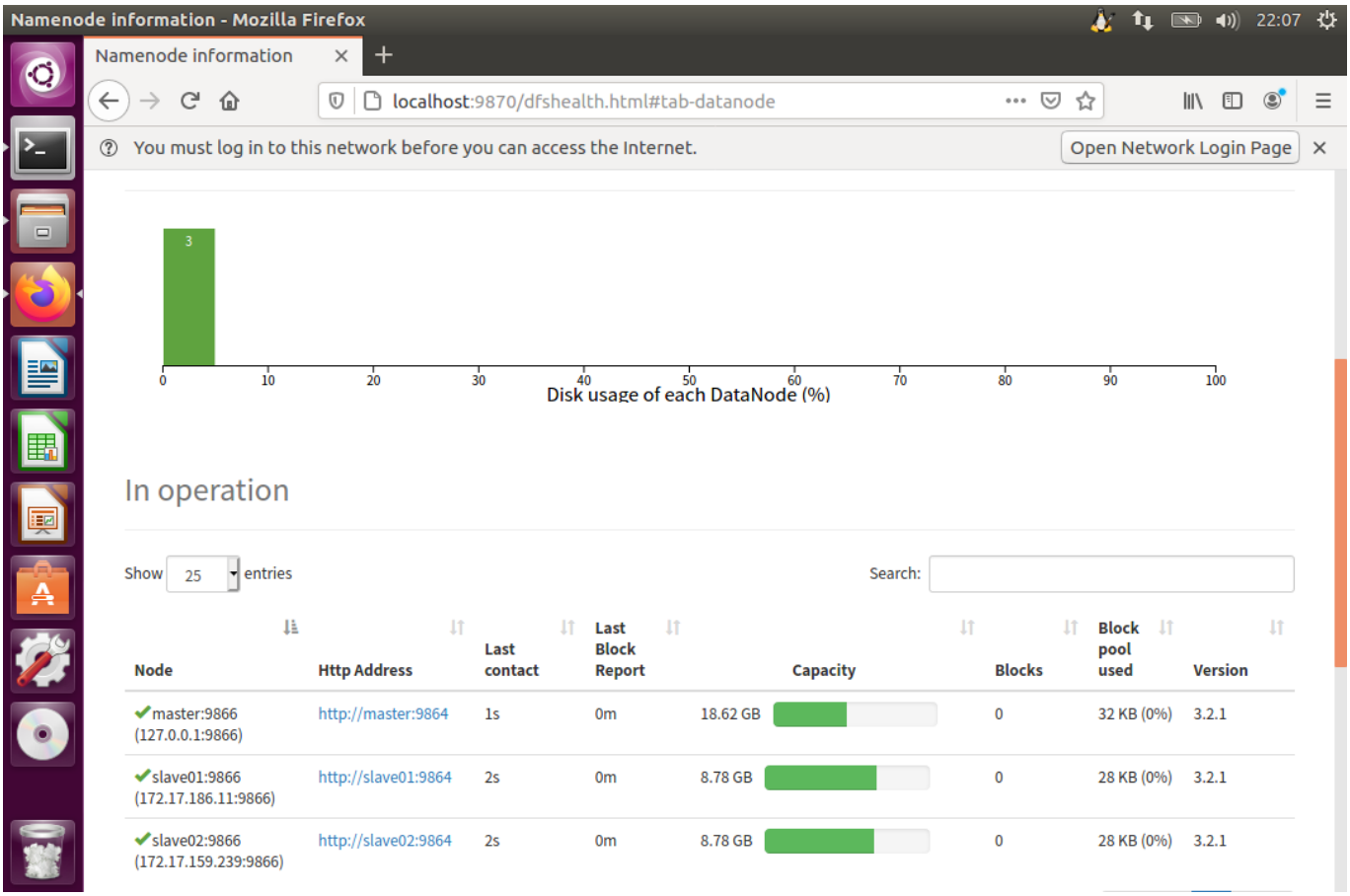
Hadoop集群启动

启动顺序为 namenode格式化->start->stop，之后启动就不需要格式化

```
cd /usr/local/hadoop
bin/hdfs namenode -format
sbin/start-all.sh
```

再次格式化再启动的话，namenode和datanode的ClusterId会产生偏差而出错，解决方法是：将core-site.xml中hadoop.tmp.dir设置的目录下的文件全部删除，然后重新格式化即可。

成功启动后在master的9870端口**（hadoop2.x是50070）**查看：



Spark集群

参考http://dblab.xmu.edu.cn/blog/1307-2/

Spark镜像下载“Pre-build with user-provided Hadoop”版，或者匹配已安装的hadoop的版本

下载完成后，执行如下命令：

```
sudo tar -zxf 下载地址/spark-2.0.2-bin-without-hadoop.tgz -C /usr/local/
cd /usr/local
```

```
sudo mv ./spark-2.0.2-bin-without-hadoop/ ./spark
sudo chown -R 用户名 ./spark
```

配置环境变量

在Mster节点主机的终端中执行如下命令：

```
vim ~/.bashrc
```

在.bashrc添加如下配置：

```
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
```

执行如下命令使得配置立即生效：

```
source ~/.bashrc
```

Spark配置

在Master节点主机上进行如下操作：

- 配置slaves文件 将 slaves.template 拷贝到 slaves

```
cd /usr/local/spark/
cp ./conf/slaves.template ./conf/slaves
```

slaves文件设置Worker节点。编辑slaves内容,把默认内容localhost替换成如下内容：

```
master  #master主机既作为master，也作为worker
slave01
slave02
```

- 配置spark-env.sh文件

  将 spark-env.sh.template 拷贝到 spark-env.sh

  ```
  cp ./conf/spark-env.sh.template ./conf/spark-env.sh
  ```

编辑spark-env.sh,添加如下内容： **(这里尽量写全一些，不容易出BUG)**

```
export SPARK_DIST_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export SCALA_HOME=/usr/lib/scala/scala-2.13.3
export HADOOP_HOME=/usr/local/hadoop

export SPARK_MASTER_IP=master的ip #指定 Spark 集群 Master 节点的 IP 地址;
export SPARK_MASTER_PORT=7077
export SPARK_MASTER_HOST=master的ip
export SPARK_EXECUTOR_MEMORY=1024m  #大小看虚拟机内存
```

配置好后，将Master主机上的/usr/local/spark文件夹复制到各个节点上。在Master主机上执行如下命令：

```
cd /usr/local/tar -zcf ~/spark.master.tar.gz ./spark
cd ~
scp ./spark.master.tar.gz slave01:/home/用户名
scp ./spark.master.tar.gz slave02:/home/用户名
```

在slave01,slave02节点操作：

```
sudo rm -rf /usr/local/spark/
sudo tar -zxf ~/spark.master.tar.gz -C /usr/local
sudo chown -R 用户名 /usr/local/spark
```

启动Hadoop集群

启动Spark集群前，要先启动Hadoop集群。在Master节点主机上运行如下命令：

```
cd /usr/local/hadoop/sbin/start-all.sh
```

启动Spark集群

1. 启动Master节点

   在Master节点主机上运行如下命令： **(这里对启动指令略作修改，解决一个worker启动，但是webUI看不到的BUG)**

   ```
   cd /usr/local/spark/
   sbin/start-master.sh -h 172.17.177.70 (master的ip)
   ```

   在Master节点上运行jps命令，可以看到多了个Master进程：

```
xzy@master:/usr/local/spark$ jps
6436 NodeManager
10394 Jps
3226 DataNode
3434 SecondaryNameNode
3579 ResourceManager
10156 Master
10300 Worker
3100 NameNode
```

2. 启动所有Slave节点

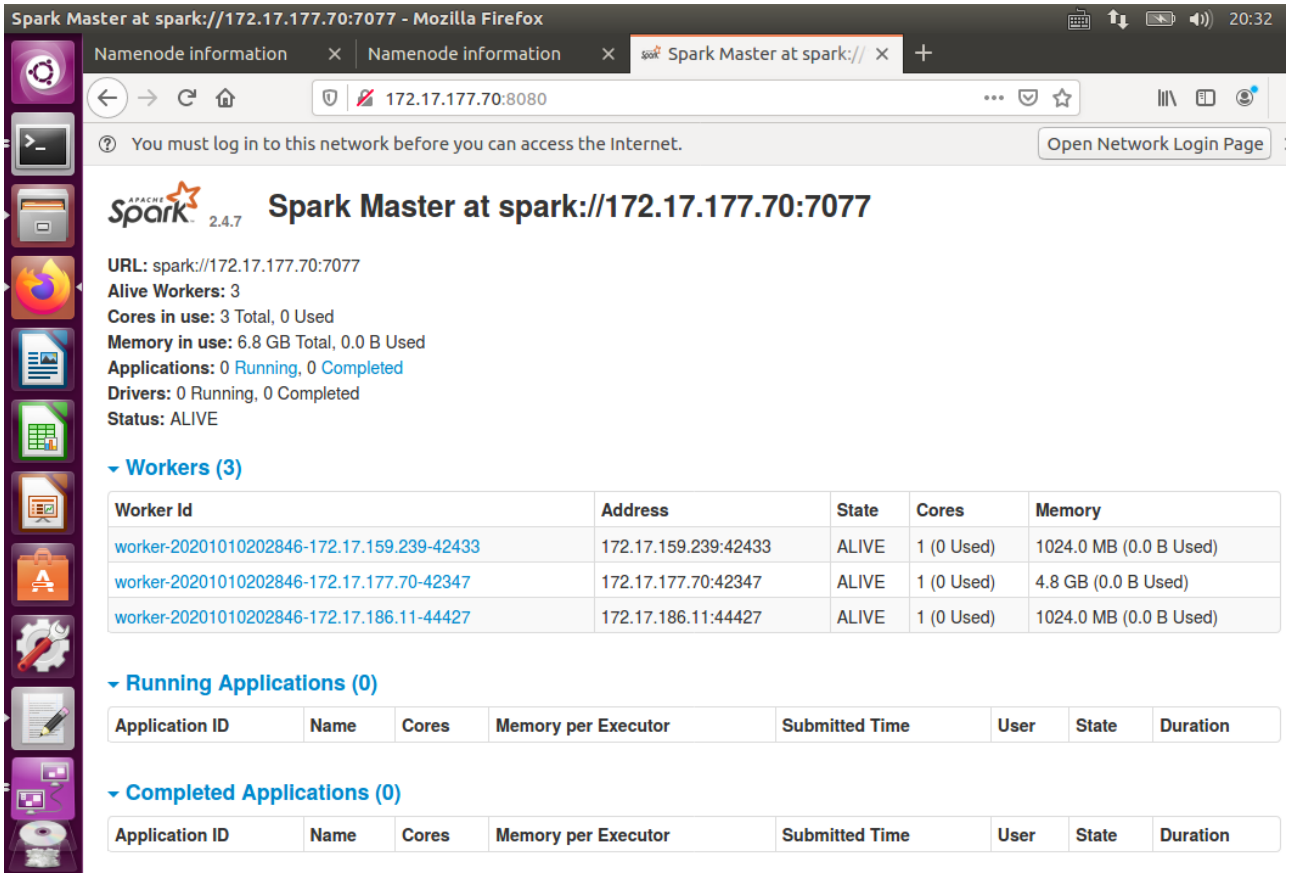
   在Master节点主机上运行如下命令：

   ```
   sbin/start-slaves.sh
   ```

   分别在slave01、slave02节点上运行jps命令，可以看到多了个Worker进程

```
xzy@slave02:~$ jps
5081 Jps
3098 NodeManager
4893 Worker
2959 DataNode
```

3. 在浏览器上查看Spark独立集群管理器的集群信息 在master主机上打开浏览器，访问http://master:8080，如下图：



关闭Spark集群

1. 关闭Master节点

```
sbin/stop-master.sh
```

2. 关闭Worker节点

```
sbin/stop-slaves.sh
```

3. 关闭Hadoop集群

```
cd /usr/local/hadoop/sbin/stop-all.sh
```

worker启动，但是webUI看不到的BUG

Slave无法连接Master，查它的Log，可以得到如下错误。

```
2020-10-09 23:50:54,966 WARN worker.Worker: Failed to connect to master master:7077
org.apache.spark.SparkException: Exception thrown in awaitResult: 
        at org.apache.spark.util.ThreadUtils$.awaitResult(ThreadUtils.scala:302)
        at org.apache.spark.rpc.RpcTimeout.awaitResult(RpcTimeout.scala:75)
        at org.apache.spark.rpc.RpcEnv.setupEndpointRefByURI(RpcEnv.scala:101)
        at org.apache.spark.rpc.RpcEnv.setupEndpointRef(RpcEnv.scala:109)
        at org.apache.spark.deploy.worker.Worker$$anon$1.run(Worker.scala:277)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.io.IOException: Connecting to master/172.17.177.70:7077 timed out (120000 ms)
        at org.apache.spark.network.client.TransportClientFactory.createClient
(TransportClientFactory.java:251)
        at org.apache.spark.network.client.TransportClientFactory.createClient
(TransportClientFactory.java:195)
        at org.apache.spark.rpc.netty.NettyRpcEnv.createClient(NettyRpcEnv.scala:204)
        at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:202)
        at org.apache.spark.rpc.netty.Outbox$$anon$1.call(Outbox.scala:198)
        ... 4 more
2020-10-09 23:51:22,956 INFO worker.Worker: Retrying connection to master (attempt # 9)
```

查看Master进程使用的端口：

```
netstat -tlnp
```

```
tcp6       0      0 :::8033                 :::*                    LISTEN
3579/java
tcp6       0      0 172.17.177.70:7077      :::*                    LISTEN
9577/java
tcp6       0      0 172.17.177.70:8080      :::*                    LISTEN
9577/java
tcp6       0      0 172.17.177.70:8081      :::*                    LISTEN
9719/java
```

是正常启动的

nmap检查master的7077端口，是通的

```
xzy@slave01:~$ nmap -p 7077 master

Starting Nmap 7.01 ( https://nmap.org ) at 2020-10-10 20:27 CST
Nmap scan report for master (172.17.177.70)
Host is up (0.00032s latency).
PORT     STATE SERVICE
7077/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

最后按网上教程，启动使用： sbin/start-master.sh -h 172.17.177.70（master的ip），并补全配置，解决了问题