**Spark的standalone模式部署**

**目前我们有四台主机，分别名为**

**外网IP 内网IP**

**SRP01 210.209.117.148 10.0.2.33**

**SRP02 210.209.117.219 10.0.2.53**

**SRP03 113.10.168.178 10.0.2.54**

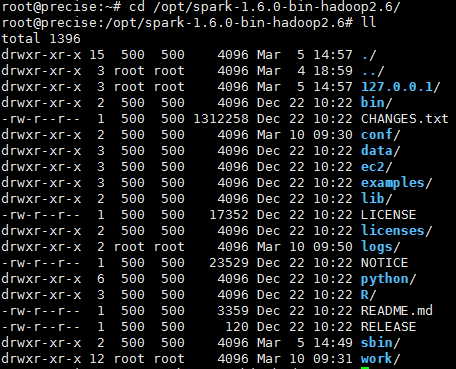
**SRP04 113.10.137.47 10.0.2.61**

**SRP01同时作为主节点master与工作节点worker，其余三台机器作为工作节点worker，在SRP01下启动和停止整个集群。另外特别注意的是，每一个步骤若无特殊说明，则默认必须对每一台主机进行相同的配置，尤其是：spark必须安装在四台主机的同一个目录下。**

**我们将分部分叙述部署的详细过程，此文档今后亦将不断完善。**

**第一部分：各主机上基本环境的部署**

1. 首先需要安装Java程序的运行环境，因为scala程序的运行需要JVM的支持，同时为了以后编写Java程序方便，可以直接安装JDK，在ubuntu下，使用命令apt-get install openjdk-7-jdk；
2. 安装spark，先获取spark的URL，然后使用wget命令下载spark，存放在/opt目录下；
3. 使用命令tar zxvf spark-1.6.0-bin-hadoop2.6.tgz解压spark压缩包；
4. 使用命令cd spark-1.6.0-bin-hadoop2.6进入spark的目录，可以看到spark下有如下目录：

./bin目录下存放spark可执行文件，例如spark-shell，spark-sql，pyspark等，./sbin目录下存放关于集群管理的脚本，例如start-all.sh，stop-all.sh，start-master.sh，start-slave.sh等。这两个目录经常会用到，我们可以将这两个目录路径导出到环境变量中去。另外，./conf下存放spark的配置文件，我们稍后将会修改里面的文件，用以配置集群。

（左图为步骤四图）

1. 导出环境变量，使用命令vi /etc/profile修改bash的环境配置：在/etc/profile下追加以下两行，这样每次登录都能直接执行步骤4中提到的目录中的命令和脚本，而不必进入该目录。

（左图为步骤五图）

1. 为避免spark的shell输出太多我们不需要的信息，可以修改spark日志的级别，这一步也许是必须的，因为不修改的话，在spark的shell中的每一步操作都会输出很多无用信息，干扰我们使用：进入目录/opt/spark-1.6.0-bin-hadoop2.6/conf，使用命令cp log4j.properties.template log4j.properties基于模板建立我们的偏好配置文件，用vi修改log4j.properties中的rootCategory项，将INFO改成WARN。

（左图为步骤六图）

**第二部分：standalone模式的部署**

1. 设置好从主节点到其他机器的SSH无密码登录，这需要在所有的机器上都有相同的用户账号，这里我们用root，先在主节点上生成SSH私钥，再将其放到所有工作节点中去，具体步骤如下：

第一步：进入SRP01的root的家目录，执行命令ssh-keygen -t dsa，接受所有默认选项，这将会在家目录下生成.ssh/的隐藏文件夹，文件夹里有authorized\_keys，id\_dsa，id\_dsa.pub三个文件，其中id\_dsa.pub是我们需要的公钥；

第二步：分别在SRP02，SRP03，SRP04三台主机root的家目录下手动建立.ssh文件夹，同时使用命令chmod 700 .ssh将权限修改成700，这是linux的安全要求；

第三步：使用scp命令将SRP01中的id\_dsa.pub复制到另外三台主机对应的.ssh目录下，

比如到SRP02：scp ~/.ssh/id\_dsa.pub [root@210.209.117.219:~/.ssh/](mailto:root@210.209.117.219:~/.ssh/)

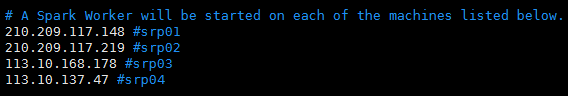
依次替换命令中的ip地址为SRP03与SRP04的外网ip即可。

第四步：在SRP02~04上重定向生成authorized\_keys：使用命令cat ~/.ssh/id\_dsa.pub >> ~/.ssh/ authorized\_keys，并修改权限：使用命令chmod 644 ~/.ssh/authorized\_keys，这个权限也是linux要求的，若权限不对，则登录需要密码；

至此，ssh无密码登录已经配置完成。

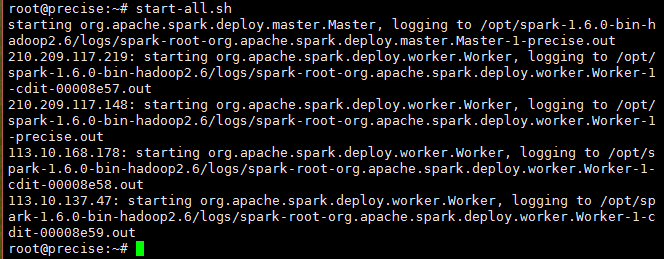
1. 在主机SRP01上，进入/opt/spark-1.6.0-bin-hadoop2.6/conf，基于slaves.template创建slaves文件，使用vi在此文件中分别添加SRP01~04四台主机的主机名或者ip地址（因为SRP01同时作为worker），注意若这里填写的是主机名，则需要在SRP01的/etc/hosts文件内填写SRP01~04的ip，否则会无法解析。而这里为了方便，我们直接在slaves填写SRP01~04的ip。

**特别注意，这里务必填写主机的外网ip**！如下图：



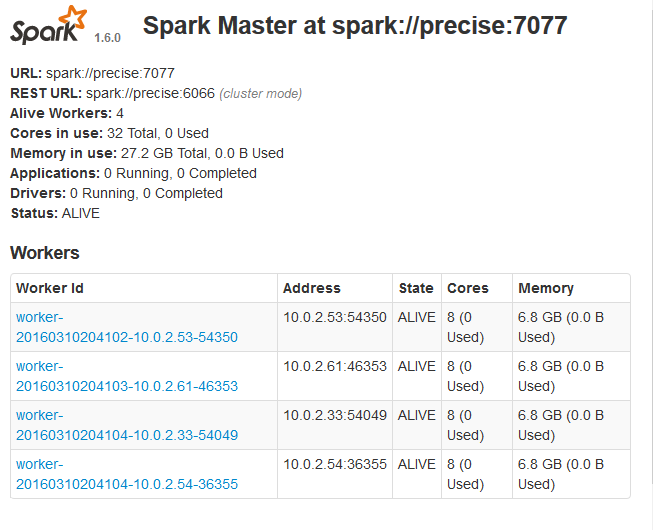
**第三部分：集群的启动和停止，作业提交**

1. 在主机SRP01上，启动脚本start-all.sh:

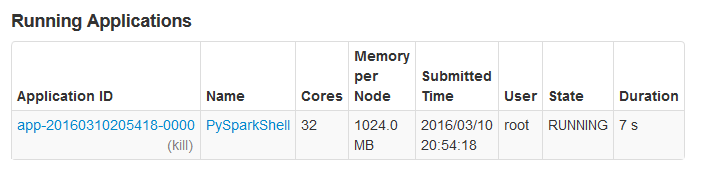


可以看到spark先启动了master，然后再启动了四个worker，相关的日志信息也被输出了，出现问题时可以查找日志。

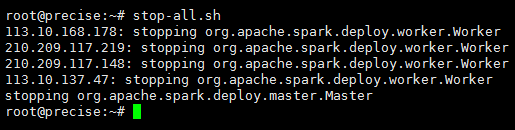
1. 在浏览器中输入http://210.209.117.148:8080/，来到spark的WebUI，这里可以看到集群的详细信息，注意，默认情况下，各个worker的ip是用内网ip来表征的：



1. 我们可以在任意一个worker上提交应用，那么，这个提交应用的worker将会成为驱动器节点，其余三个是执行器节点，例如，可以尝试在SRP02上提交pyspark的应用，在SRP02上输入：pyspark --master spark://210.209.117.148:7077（当中ip是主节点master的外网ip），然后我们就能看到应用的运行情况了：



1. 在SRP01下，通过执行脚本stop-all.sh就能停止整个集群，同时WebUI将不能再打开，在集群停止时，同样是先停掉四个worker，再停掉master，假如集群中某个worker之前没有被启动，则系统会在该worker的ip后显示“no worker to stop ```”。



**第四部分：尚未明确**

位于conf目录下的spark-env.sh是个非常重要的文件，但上面我们并没有对它进行配置，该文件用于设置端口，yarn管理器，ip地址等，内部有一个变量SPARK\_LOCAL\_IP，这个变量有点奇怪，SPARK\_MASTER\_IP我们也没有设置，但不影响集群的工作。

根据learning spark中的叙述，spark-env.sh在spark启动时会被sourced，最开始搭建环境的时候，我在spark-env.sh下添加过如下的命令：

export SPARK\_LOCAL\_IP=x.x.x.x

export SPARK\_MASTER\_IP=x.x.x.x

在启动spark后，按下ctrl+Z将其切换到后台，输入env查看环境变量，也没有SPARK\_LOCAL\_IP和SPARK\_MASTER\_IP。