|  |  |
| --- | --- |
| 文档编号：[pxxx - ]xxx - 1xx - ymd - xxxx | 说明（保持文档结构，内容多则加附件） |
| 规则：（[项目代码/编号-]文档类别/简介-组员编号-填写日期-流水号） [...]表示可选项 | |

**Flink集群安装文档**

# 安装环境及介质

* **操作系统：ubuntu-16.04.2-desktop-amd64.iso**
* **虚拟机：VirtualBox-5.1.18**
* **用户口令：smartai/smartai**
* **主机列表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***机器编号*** | ***NAT地址*** | ***Host-only地址*** | ***CPU（逻辑）*** | ***内存（G）*** |
| LTSR005 | dhcp | 192.168.0.15 | 1\*2=2 | 2 |
| LTSR006 | dhcp | 192.168.0.16 | 1\*2=2 | 2 |
| LTSR007 | dhcp | 192.168.0.17 | 1\*2=2 | 2 |
| LTSR008 | dhcp | 192.168.0.18 | 1\*2=2 | 2 |

# 资源下载

|  |  |
| --- | --- |
| ***组件*** | ***下载站点*** |
| ubuntu-16.04.2-desktop-amd64.iso | http://releases.ubuntu.com/16.04/ubuntu-16.04-desktop-amd64.iso |
| VirtualBox-5.1.18-114002.exe | http://download.virtualbox.org/virtualbox/ |
| jdk-8u211-linux-x64.tar.gz | https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8u211-later-5573849.html |
| kafka\_2.11-0.11.0.3.tgz | http://kafka.apache.org/downloads.html |
| zookeeper-3.4.10.tar.gz | http://archive.apache.org/dist/zookeeper/zookeeper-3.4.10/ |
| mysql-server\_5.7.22-1ubuntu16.04\_amd64.deb-bundle.tar | https://downloads.mysql.com/archives/community/ |
| flink-1.9.0-bin-scala\_2.11.tgz | http://archive.apache.org/dist/flink/flink-1.9.0/ |
| flink-1.9.0-src.tgz | http://archive.apache.org/dist/flink/flink-1.9.0/ |
| apache-maven-3.2.5-bin.tar.gz | https://archive.apache.org/dist/maven/maven-3/ |

# 资源规划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **comp** | [**version**](app:ds:version) | LTSR005 | LTSR006 | LTSR007 | LTSR008 |
| OS | ubuntu-16.04.2-desktop-amd64.iso | ubuntu-x64 | ubuntu-x64 | ubuntu-x64 | ubuntu-x64 |
| JDK | jdk-8u211-linux-x64.tar.gz | jvm | jvm | jvm | jvm |
| Zookeeper | zookeeper-3.4.10.tar.gz | zk | zk | zk | N.A |
| Kafka | kafka\_2.11-0.11.0.3.tgz | kafka | kafka | kafka | N.A |
| Flink | flink-1.9.0.tar | flink | flink | flink | 编译Flink |
| MySQL | mysql-server\_5.7.22-1ubuntu16.04\_amd64.deb-bundle.tar | N.A | N.A | N.A | MySQL Server/Client |
| Maven | apache-maven-3.2.5-bin.tar.gz | N.A | N.A | N.A | Maven |

# 系统安装

参考《[安装文档]VirtualBox\_Ubuntu16.04\_x85\_64.docx》。

## 集群配置

设置主机名：

|  |
| --- |
| ***#各节点单独配置***  ***$ sudo*** *vi /etc/hostname*  LTSR005  LTSR006  LTSR007  LTSR008  ***$ sudo*** *vi /etc/hosts*  ***#各节点单独配置***  ***# LTSR005 192.168.0.15***  127.0.0.1 localhost  127.0.1.1 LTSR005  ***# LTSR006 192.168.0.16***  127.0.0.1 localhost  127.0.1.1 LTSR006  ***# LTSR007 192.168.0.17***  127.0.0.1 localhost  127.0.1.1 LTSR007  ***# LTSR008 192.168.0.18***  127.0.0.1 localhost  127.0.1.1 LTSR008  ***#以下所有节点直接复制***  192.168.0.15 LTSR005  192.168.0.16 LTSR006  192.168.0.17 LTSR007  192.168.0.18 LTSR008 |

## 双网卡模式

设置前请保证VirtualBox关于网络配置的全局设定正确无误（参考：[安装文档]VirtualBox\_Ubuntu16.04\_x85\_64.docx --“全局设定（双网卡配置）”），再进行如下设置：

a）网卡1 设置为“网络地址转换（NAT）”，虚拟机可以通过此网卡访问外网。

b）网卡2设置为“仅主机（Host-Only）网络”。

## 配置网络

|  |
| --- |
| ***$ sudo*** *vi /etc/network/interfaces*  ***#各节点单独配置***  ***# LTSR005 192.168.0.15***  auto enp0s8  iface enp0s8 inet static  address 192.168.0.15  netmask 255.255.255.0  auto enp0s3  iface enp0s3 inet dhcp  hwaddress ether 00:AA:BB:CC:DD:15  ***# LTSR006 192.168.0.16***  auto enp0s8  iface enp0s8 inet static  address 192.168.0.16  netmask 255.255.255.0  auto enp0s3  iface enp0s3 inet dhcp  hwaddress ether 00:AA:BB:CC:DD:16  ***# LTSR007 192.168.0.17***  auto enp0s8  iface enp0s8 inet static  address 192.168.0.17  netmask 255.255.255.0  auto enp0s3  iface enp0s3 inet dhcp  hwaddress ether 00:AA:BB:CC:DD:17  ***# LTSR008 192.168.0.18***  auto enp0s8  iface enp0s8 inet static  address 192.168.0.18  netmask 255.255.255.0  auto enp0s3  iface enp0s3 inet dhcp  hwaddress ether 00:AA:BB:CC:DD:18  ***# 验证网络（需重启）***  ***$*** *ping baidu.com* |

# 准备工作

创建相关目录：

|  |
| --- |
| ***# software及modules目录每个节点都需创建，先在节点LTSR005上安装，之后分发到LTSR006、LTSR007***  ***$*** *mkdir -p ~/software*  ***$*** *mkdir -p ~/modules* |

# SSH免密

配置ssh免密，实现节点间的无密码登录 （***注意关闭防火墙***），将实现节点LTSR005、LTSR006、LTSR007互访。

## Step1.LTSR005生成公钥、私钥

（1）以[smartai]身份登录产生密钥（***集群所有机器执行脚本***）

|  |
| --- |
| ***$*** *cd ~*  ***$*** *ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa*  ***##注：以rsa算法，生成公钥、私钥对，-P ''表示空密码。该命令运行完后，会在个人主目录下生成.ssh目录，里面会有二个文件id\_rsa（私钥） ,id\_rsa.pub(公钥)*** |

（2）导入authorized\_keys（***LTSR005执行脚本***）

|  |
| --- |
| ***$*** *cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys*  ***$*** *chmod 600 ~/.ssh/authorized\_keys*  ***$*** *chmod 600 ~/.ssh/id\_rsa* |

（3）[root]用户添加[smartai]用户到sudo（***集群所有机器执行脚本***）

|  |
| --- |
| ***$*** *su*  ***$*** *chmod 755 /etc/sudoers*  ***$*** *vi /etc/sudoers*  smartai ALL=(ALL) ALL  ***$*** *chmod 766 /etc/ssh/sshd\_config* |

（4）[smartai]用户登录服务器修改SSH配置文件

|  |
| --- |
| ***$*** *su smartai*  ***$ sudo*** *vi /etc/ssh/sshd\_config*  ***##配置项如下：***  RSAAuthentication yes ***##启用 RSA 认证***  PubkeyAuthentication yes***##启用公钥私钥配对认证方式***  PermitRootLogin yes***##不限制登录方式***  AuthorizedKeysFile .ssh/authorized\_keys ***##公钥文件路径（和上面生成的文件同）***  ***##设置完之后记得重启SSH服务，才能使刚才设置有效。***  ***$******sudo*** *scp /etc/ssh/sshd\_config smartai@LTSR006:/etc/ssh/sshd\_config*  ***$******sudo*** *scp /etc/ssh/sshd\_config smartai@LTSR007:/etc/ssh/sshd\_config*  ***$ sudo*** */etc/init.d/ssh restart*  ***$ sudo*** */etc/init.d/ssh status* |

（5）验证（***LTSR005执行脚本***）

|  |
| --- |
| ***$*** *ssh localhost*  ***$*** *ssh LTSR005* |

## Step2.在其它机器上生成公钥、密钥

a） 以[smartai]用户身份登录其它二台机器 LTSR006、LTSR007，生成公钥、密钥（如果此步骤在上一步骤中已做，略过a)步骤）

|  |
| --- |
| ***$*** *ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa* |

b） 然后用scp命令，把公钥文件发放给LTSR005

|  |
| --- |
| ***## LTSR006***  ***$*** *scp ~/.ssh/id\_rsa.pub smartai@LTSR005:~/id\_rsa\_02.pub*  ***## LTSR007***  ***$*** *scp ~/.ssh/id\_rsa.pub smartai@LTSR005:~/id\_rsa\_03.pub* |

## Step3.合并所有节点密钥

在LTSR005上将其他节点（LTSR006、LTSR007）上传的id\_rsa\_xx.pub，导入authorized\_keys

|  |
| --- |
| ***$*** *cat ~/id\_rsa\_02.pub >> ~/.ssh/authorized\_keys*  ***$*** *cat ~/id\_rsa\_03.pub >> ~/.ssh/authorized\_keys* |

经过这个步骤，LTSR005上拥有所有集群机器的authorized\_keys

## Step4.分发公钥集合

a）在LTSR005上分发authorized\_keys

|  |
| --- |
| ***##将LTSR005上的“最全”公钥，复制到其它机器***  ***$*** *scp ~/.ssh/authorized\_keys smartai@LTSR006:~/.ssh/authorized\_keys*  ***$*** *scp ~/.ssh/authorized\_keys smartai@LTSR007:~/.ssh/authorized\_keys* |

b）修改其它机器（LTSR006、LTSR007）上authorized\_keys文件的权限

|  |
| --- |
| ***$*** *chmod 600 ~/.ssh/authorized\_keys*  ***$*** *chmod 600 ~/.ssh/id\_rsa*  ***$*** *chmod g-w ~/.ssh*  ***$*** *chmod 700 ~/.ssh* |

## Step5.验证

|  |
| --- |
| ***# 各节点相互登录验证***  ***$*** *ssh LTSR005*  ***$*** *ssh LTSR006*  ***$*** *ssh LTSR007*  ***$*** *exit* |

# JDK安装

根据资源规划每个节点都需要安装JDK环境。

## Step1.解压

|  |
| --- |
| ***$*** *cd ~/software*  ***$*** *tar -zxvf jdk-8u211-linux-x64.tar.gz -C ~/modules*  ***$*** *rm jdk-8u211-linux-x64.tar.gz* |

## Step2.添加环境变量并生效

|  |
| --- |
| ***$*** *vi ~/.bashrc* ***# :$到达行尾添加***  export JAVA\_HOME=/home/smartai/modules/jdk1.8.0\_211  export PATH=$JAVA\_HOME/bin:$PATH  export CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/lib/rt.jar:$JAVA\_HOME/lib/tools.jar  ***$*** *source ~/.bashrc* ***##环境变量生效***  ***$*** *java -version* ***##验证：版本号不正确可能由PATH造成，可以将$JAVA\_HOME/bin放在export PATH最前面***  ***$*** whereis java  ***$*** which java ***##java命令路径*** |

## Step3.分发

|  |
| --- |
| ***$*** *cd ~/modules*  ***$*** *scp -r jdk1.8.0\_211 smartai@LTSR006:/home/smartai/modules/*  ***$*** *scp -r jdk1.8.0\_211 smartai@LTSR007:/home/smartai/modules/*  ***$*** *scp -r jdk1.8.0\_211 smartai@LTSR008:/home/smartai/modules/*  ***# 分发~/.bashrc***  ***$****scp -r ~/.bashrc smartai@LTSR006:/home/smartai/*  ***$*** *scp -r ~/.bashrc smartai@LTSR007:/home/smartai/*  ***$*** *scp -r ~/.bashrc smartai@LTSR008:/home/smartai/*  ***# 各节点激活配置，或者重新登录节点即可激活配置***  ***$*** *source ~/.bashrc* ***##环境变量生效*** |

# 安装Maven

根据资源规划，Maven安装在节点LTSR008上。

|  |
| --- |
| ***$*** *cd ~/software*  ***$*** *tar -zxvf apache-maven-3.2.5-bin.tar.gz -C ~/modules*  ***$*** *rm apache-maven-3.2.5-bin.tar.gz* |

## Step1.配置环境变量

|  |
| --- |
| ***$*** *vi ~/.bashrc* ***# :$到达行尾添加***  export M2\_HOME=/home/smartai/modules/apache-maven-3.2.5  export M2=$M2\_HOME/bin  export PATH=$M2:$PATH  ***$*** *source ~/.bashrc*  ***# 验证***  ***$*** *mvn -v* |

## Step2.配置阿里云镜像

配置~/modules/apache-maven-3.2.5/conf目录下的settings.xml，内容太多，建议下载到本地修改后再提交。

|  |
| --- |
| ***#在<mirrors>标签下添加如下配置***  <mirror>  <id>alimaven</id>  <name>aliyun maven</name>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  <mirrorOf>central</mirrorOf>  </mirror> |

# MySQL安装

根据资源规划，Maven安装在节点LTSR008上。

## Step1.安装服务

|  |
| --- |
| ***$*** *cd ~/software*  ***$*** tar -xvf mysql-server\_5.7.22-1ubuntu16.04\_amd64.deb-bundle.tar  ***$*** rm mysql-server\_5.7.22-1ubuntu16.04\_amd64.deb-bundle.tar  ***# 根据提示下载相关离线包，硬盘中已下载可以直接上传，然后继续下列安装步骤***  ***$ sudo*** dpkg -i mysql-common\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg-preconfigure mysql-community-server\_5.7.22-1ubuntu16.04\_amd64.deb ***#设置密码123456，也可以后续设置***  ***$ sudo*** dpkg -i libmysqlclient20\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg -i libmysqlclient-dev\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg -i libmysqld-dev\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg -i libc6\_2.19-0ubuntu6\_amd64.deb ***#需下载***  ***$ sudo*** dpkg -i libaio1\_0.3.110-2\_amd64.deb ***#需下载***  ***$ sudo*** dpkg -i mysql-community-client\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg -i mysql-client\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg -i mysql-common\_5.7.22-1ubuntu16.04\_amd64.deb  ***$ sudo*** dpkg -i libmecab2\_0.996-1.2ubuntu1\_amd64.deb ***#需下载***  ***$ sudo*** dpkg -i mysql-community-server\_5.7.22-1ubuntu16.04\_amd64.deb  ***# deb包下载站点***  https://pkgs.org/  ***$*** rm \*.deb |

## Step2.修改配置文件

|  |
| --- |
| ***$ sudo*** vi /etc/mysql/mysql.conf.d/mysqld.cnf  # bind-address = 127.0.0.1 # 注释（以支持远程连接）  character-set-server=utf8 # 添加  innodb\_file\_per\_table=1 # 添加 |

## Step3.设置远程连接

|  |
| --- |
| ***$*** mysqladmin -uroot -p password 123456  ***$*** mysql -u root -p123456  ***mysql>*** use mysql;  ***mysql>*** update user set host = '%' where user = 'root';  ***mysql>*** update user set authentication\_string=password("123456") where user="root";  ***mysql>*** update user set authentication\_string=password("123456"),plugin='mysql\_native\_password' where user='root';  ***mysql>*** grant all on \*.\* to root@'%' identified by '123456' with grant option;  ***mysql>*** grant all privileges on \*.\* to 'root'@'%' identified by '123456' with grant option;  ***mysql>*** flush privileges; |

## Step4.卸载服务

|  |
| --- |
| **$ *sudo*** apt-get purge mysql\*  **$ *sudo*** apt autoremove  **$ *sudo*** apt-get remove --purge mysql\*  **$ *sudo*** rm -rf /var/lib/mysql  **$ *sudo*** rm -rf /etc/mysql/\*  **$**dpkg -l | grep mysql |

## Step5.启停服务

|  |
| --- |
| **$ *sudo*** systemctl status mysql  **$ *sudo*** systemctl start mysql  **$ *sudo*** systemctl stop mysql  ***# MySQL服务启动停止***  **$ *sudo*** service mysql stop  **$ *sudo*** service mysql start  **$ *sudo*** service mysql restart  **$ *sudo*** service mysql status |

## Step6.开机启动

|  |
| --- |
| **$ *sudo*** systemctl disable mysql ***#禁止启动***  **$ *sudo*** systemctl enable mysql ***#开机启动*** |

# Zookeeper分布式安装

根据资源规划，Maven安装在节点LTSR005、LTSR006、LTSR007上。

## Step1.解压缩

|  |
| --- |
| ***# 先在节点LTSR005上安装，之后分发到LTSR006、LTSR007***  ***$*** *cd ~/software*  ***$*** *tar -xvf zookeeper-3.4.10.tar.gz -C ~/modules/*  ***$*** *rm zookeeper-3.4.10.tar.gz* |

## Step2.配置

（1）创建data目录、log目录及myid文件。

|  |
| --- |
| ***$*** *mkdir -p ~/modules/zookeeper-3.4.10/data*  ***$*** *mkdir -p ~/modules/zookeeper-3.4.10/log*  ***$*** *cd ~/modules/zookeeper-3.4.10/data*  ***$*** *touch myid*  ***##myid分别写入{1,2,3}，具体那台机器写入什么值根据zoo.cfg中的配置设置，如server.1=xxxx，则myid写入1即可。*** |

（2）修改zoo.cfg。

|  |
| --- |
| ***$*** *cd ~/modules/zookeeper-3.4.10/conf*  ***$*** *cp zoo\_sample.cfg zoo.cfg*  ***$*** *vi zoo.cfg* |

（3）配置zoo.cfg

|  |
| --- |
| ***# 清除掉原来的所有配置，添加下列配置即可（中文注释建议清理掉）***  tickTime=2000***##心跳时间***  initLimit=10***##follow连接leader的初始化连接时间，表示tickTime的倍数***  syncLimit=5***## syncLimit配置表示leader与follower之间发送消息，请求和应答时间长度。如果followe在设置的时间内不能与leader进行通信，那么此follower将被丢弃，tickTime的倍数***  clientPort=2181 ***##客户端连接端口***  dataDir=/home/smartai/modules/zookeeper-3.4.10/data ***##节点数据存储目录，需要提前创建，注意myid添加，用于标识服务器节点***  dataLogDir=/home/smartai/modules/zookeeper-3.4.10/log  server.1=192.168.0.15:2888:3888  server.2=192.168.0.16:2888:3888  server.3=192.168.0.17:2888:3888 |

（4）配置zkEnv.sh。

|  |
| --- |
| ***$*** *vi ~/modules/zookeeper-3.4.10/bin/zkEnv.sh*  export JAVA\_HOME=/usr/local/jdk1.8.0\_221 |

***注：2888为follow连接leader的端口，3888为选举端口，伪集群模式下，由于在一台主机上，故该两个端口不能配置相同，同时clientPort也不同。***

## Step3.分发

|  |
| --- |
| ***$*** *cd ~/modules/*  ***$*** *scp -r zookeeper-3.4.10 smartai@LTSR006:~/modules/*  ***$*** *scp -r zookeeper-3.4.10 smartai@LTSR007:~/modules/* |

## Step4.编辑myid

|  |
| --- |
| ***$****vi ~/modules/zookeeper-3.4.10/data/myid*  ***# LTSR005节点***  1  ***# LTSR006节点***  2  ***# LTSR007节点***  3 |

## Step5.验证

|  |
| --- |
| ***$*** *cd ~/modules/zookeeper-3.4.10/bin*  ***$*** *ls -l /bin/sh*  ***$ sudo*** *ln -sf bash /bin/sh*  ***$*** *sh zkServer.sh start* ***##所有机器开启zk {start|start-foreground|stop|restart|status|upgrade|print-cmd}***  ***$*** *sh zkServer.sh status* ***##查看启动状态***  ***$*** *sh zkServer.sh stop*  ***$*** *jps -m* ***##查看启动进程main方法相关的进程***  ***$*** *./zkCli.sh -server localhost:2181* ***# 客户端工具*** |

# Kafka安装

根据资源规划，Maven安装在节点LTSR005、LTSR006、LTSR007上。

## Step1.解压缩

|  |
| --- |
| ***# 先在节点LTSR005上安装，之后分发到LTSR006、LTSR007***  ***$*** *cd ~/software*  ***$*** *tar -zxvf kafka\_2.11-0.11.0.3.tgz -C ~/modules/*  ***$*** *rm kafka\_2.11-0.11.0.3.tgz* |

## Step2.创建相关目录

|  |
| --- |
| ***$*** *mkdir -p ~/modules/kafka\_2.11-0.11.0.3/tmp/kafka-logs* ***## kafka数据目录***  ***$*** *chmod -R a+w ~/modules/kafka\_2.11-0.11.0.3/tmp/kafka-logs* |

## Step3.配置

（1）server.properties

|  |
| --- |
| ***$*** *vi ~/modules/kafka\_2.11-0.11.0.3/config/server.properties*  ***#节点唯一标识，注意该值为整形，建议设置为IP最后一段***  broker.id=15  ***#默认端口号***  listeners=PLAINTEXT://192.168.0.15:9092  advertised.listeners=PLAINTEXT://192.168.0.15:9092  ***#Kafka数据目录***  log.dirs=/home/smartai/modules/kafka\_2.11-0.11.0.3/tmp/kafka-logs  ***#配置Zookeeper***  zookeeper.connect=192.168.0.15:2181,192.168.0.16:2181,192.168.0.17:2181  ***#配置删除属性***  delete.topic.enable=true |

（2）zookeeper.properties

|  |
| --- |
| ***$*** *vi ~/modules/kafka\_2.11-0.11.0.3/config/zookeeper.properties*  ***#Zookeeper的数据存储路径与Zookeeper集群配置保持一致***  dataDir=/home/smartai/modules/zookeeper-3.4.10/data |

（3）consumer.properties

|  |
| --- |
| ***$*** *vi ~/modules/kafka\_2.11-0.11.0.3/config/consumer.properties*  ***#配置Zookeeper地址***  zookeeper.connect=192.168.0.15:2181,192.168.0.16:2181,192.168.0.17:2181 |

（4）producer.properties

|  |
| --- |
| ***$*** *vi ~/modules/kafka\_2.11-0.11.0.3/config/producer.properties*  ***#配置Kafka集群地址***  bootstrap.servers=192.168.0.15:9092,192.168.0.16:9092,192.168.0.17:9092 |

（5）kafka-run-class.sh

|  |
| --- |
| ***$*** *vi ~/modules/kafka\_2.11-0.11.0.3/bin/kafka-run-class.sh*  ***#行首新增JAVA\_HOME配置***  export JAVA\_HOME=/home/smartai/modules/jdk1.8.0\_211 |

## Step4.分发

|  |
| --- |
| ***$*** *cd ~/modules/*  ***$*** *scp -r kafka\_2.11-0.11.0.3 smartai@LTSR006:/home/smartai/modules/*  ***$*** *scp -r kafka\_2.11-0.11.0.3 smartai@LTSR007:/home/smartai/modules/* |

修改分发节点的server.properties

|  |
| --- |
| ***$*** *vi ~/modules/kafka\_2.11-0.11.0.3/config/server.properties*  ***# LTSR006节点***  broker.id=16  listeners=PLAINTEXT://192.168.0.16:9092  advertised.listeners=PLAINTEXT://192.168.0.16:9092  ***# LTSR007节点***  broker.id=17  listeners=PLAINTEXT://192.168.0.17:9092  advertised.listeners=PLAINTEXT://192.168.0.17:9092 |

## Step5.验证

（1）确保已启动Zookeeper集群。

（2）各个节点启动Kafka集群。

|  |
| --- |
| ***$*** *cd ~/modules/kafka\_2.11-0.11.0.3/*  ***$*** *bin/kafka-server-start.sh config/server.properties >/dev/null 2>&1 &* ***##启动***  ***$*** *bin/kafka-server-stop.sh config/server.properties >/dev/null 2>&1 &* ***##停止*** |

（3）创建topic。

|  |
| --- |
| ***$*** *cd ~/modules/kafka\_2.11-0.11.0.3/*  ***$*** *bin/kafka-topics.sh --zookeeper 192.168.0.15:2181,192.168.0.16:2181,192.168.0.17:2181 --create --topic test --replication-factor 1 --partitions 3* |

（4）查看topic列表。

|  |
| --- |
| ***$*** *cd ~/modules/kafka\_2.11-0.11.0.3/*  ***$*** *bin/kafka-topics.sh --zookeeper 192.168.0.15:2181,192.168.0.16:2181,192.168.0.17:2181 --list* |

（5）生产者生成数据。

|  |
| --- |
| ***$*** *cd ~/modules/kafka\_2.11-0.11.0.3/*  ***$*** *bin/kafka-console-producer.sh --broker-list 192.168.0.15:9092,192.168.0.16:9092,192.168.0.17:9092 --topic test* |

（6）消费者消费数据。

|  |
| --- |
| ***$*** *cd ~/modules/kafka\_2.11-0.11.0.3/*  ***$*** *bin/kafka-console-consumer.sh --zookeeper 192.168.0.15:2181,192.168.0.16:2181,192.168.0.17:2181 --topic test --from-beginning* |

（7）删除topic。

|  |
| --- |
| ***$*** *cd ~/modules/kafka\_2.11-0.11.0.3/*  ***$*** *bin/kafka-topics.sh --delete --zookeeper 192.168.0.15:2181,192.168.0.16:2181,192.168.0.17:2181 --topic test* |

# 源码构建Flink

根据资源规划，在节点LTSR008上编译Flink。

## 方法1（手动下载源码）

|  |  |
| --- | --- |
| ***## 直接在官网下载源码至用户smartai的~/software目录***  ***$*** *cd ~/softwa*re  ***$*** *tar -zxvf flink-1.9.0-src.tgz -C ~*/  ***$*** *cd ~/flink-1.9.0*  ***# 修改flink-runtime-web的pom.xml配置，以提高NPM包下载速度，避免由于网络原因造成延时而编译失败***   |  | | --- | | ***$*** *vi ~/flink-1.9.0/flink-runtime-web /pom.xml*  <plugin>  <groupId>com.github.eirslett</groupId>  <artifactId>frontend-maven-plugin</artifactId>  <version>1.6</version>  <executions>  <execution>  <id>install node and npm</id>  <goals>  <goal>install-node-and-npm</goal>  </goals>  <configuration>  ***<nodeDownloadRoot>http://npm.taobao.org/mirrors/node/</nodeDownloadRoot>***  ***<npmDownloadRoot>http://npm.taobao.org/mirrors/npm/</npmDownloadRoot>***  <nodeVersion>v10.9.0</nodeVersion>  </configuration>  </execution>  <execution>  <id>npm install</id>  <goals>  <goal>npm</goal>  </goals>  <configuration>  <arguments>ci --cache-max=0 --no-save ***--no-bin-links***</arguments>  <environmentVariables>  <HUSKY\_SKIP\_INSTALL>true</HUSKY\_SKIP\_INSTALL>  </environmentVariables>  </configuration>  </execution>  <execution>  <id>npm run build</id>  <goals>  <goal>npm</goal>  </goals>  <configuration>  <arguments>run build</arguments>  </configuration>  </execution>  </executions>  <configuration>  <workingDirectory>web-dashboard</workingDirectory>  </configuration>  </plugin> |   ***$*** *mvn clean install -DskipTests -Dfast*    ***# 将编译后的flink-1.9.0-bin目录打成tar包***  ***$*** *cd ~/flink-1.9.0/flink-dist/target/flink-1.9.0-bin*  ***$*** *tar -cvf flink-1.9.0.tar flink-1.9.0*  ***$*** *scp -r flink-1.9.0.tar smartai@LTSR005:~/software* ***# 分发给LTSR005节点*** |

## 方法2（从远程仓库下载源码）

|  |  |
| --- | --- |
| ***# 安装git后拉取flink源码 （在LTSR008节点运行）***  ***$ sudo*** *apt install git* ***# 根据提示可能要运行sudo apt-get update和sudo apt-get -f install***  ***# 利用码云作为仓库中转可以大大提高下载速度***  ***$*** *cd ~/softwa*re  ***$*** *git clone https://gitee.com/mygiteepolaris/flink.git*  ***$*** *cd flink*  ***$*** *git checkout release-1.9.0*  ***# 修改flink-runtime-web的pom.xml配置，以提高NPM包下载速度，避免由于网络原因造成延时而编译失败***   |  | | --- | | ***$*** *vi ~/software/flink/flink-runtime-web/pom.xml*  <plugin>  <groupId>com.github.eirslett</groupId>  <artifactId>frontend-maven-plugin</artifactId>  <version>1.6</version>  <executions>  <execution>  <id>install node and npm</id>  <goals>  <goal>install-node-and-npm</goal>  </goals>  <configuration>  ***<nodeDownloadRoot>http://npm.taobao.org/mirrors/node/</nodeDownloadRoot>***  ***<npmDownloadRoot>http://npm.taobao.org/mirrors/npm/</npmDownloadRoot>***  <nodeVersion>v10.9.0</nodeVersion>  </configuration>  </execution>  <execution>  <id>npm install</id>  <goals>  <goal>npm</goal>  </goals>  <configuration>  <arguments>ci --cache-max=0 --no-save ***--no-bin-links***</arguments>  <environmentVariables>  <HUSKY\_SKIP\_INSTALL>true</HUSKY\_SKIP\_INSTALL>  </environmentVariables>  </configuration>  </execution>  <execution>  <id>npm run build</id>  <goals>  <goal>npm</goal>  </goals>  <configuration>  <arguments>run build</arguments>  </configuration>  </execution>  </executions>  <configuration>  <workingDirectory>web-dashboard</workingDirectory>  </configuration>  </plugin> |   ***$*** *mvn clean install -DskipTests -Dfast*  ***# 将编译后的flink-1.9.0-bin目录打成tar包***  ***$*** *cd ~/software/flink/flink-dist/target/flink-1.9.0-bin*  ***$*** *tar -cvf flink-1.9.0.tar flink-1.9.0*  ***$*** *scp -r flink-1.9.0.tar smartai@LTSR005:~/software* ***# 分发给LTSR005节点*** |

如果安装过程失败，则需要清理相关目录，再次安装：

|  |
| --- |
| ***$*** *rm -rf ~/.m2/repository*  ***$*** *rm -rf ~/.npm*  ***$*** *rm -rf ~/software/flink* ***# 重新下载源码*** |

# Flink安装

根据资源规划，Flink安装在节点LTSR005、LTSR006、LTSR007上。

## Step1.解压缩

|  |
| --- |
| ***# 先在节点LTSR005上安装，之后分发到LTSR006、LTSR007***  **$***cd ~/software*  **$***tar -xvf flink-1.9.0.tar -C ~/modules*  **$***rm flink-1.9.0.tar* |

## Step2.配置

|  |
| --- |
| **$** *cd ~/modules/flink-1.9.0/conf*  **$** *vi flink-conf.yaml*  ***# flink-conf.yaml直接替换成以下配置 （注意key跟value之间的空格）***  jobmanager.rpc.address: LTSR005  jobmanager.rpc.port: 6123  jobmanager.heap.size: 1024m  taskmanager.heap.size: 1024m  taskmanager.numberOfTaskSlots: 10  taskmanager.memory.preallocate: false  parallelism.default: 1  jobmanager.web.port: 8081  env.java.opts: -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysPreTouch -server -XX:+HeapDumpOnOutOfMemoryError  env.java.home: /home/smartai/modules/jdk1.8.0\_211  ***#LTSR005主节点，LTSR006、LTSR007副节点，***  **$** *vi masters* ***##添加文本：***  LTSR005:8081  **$** *vi slaves* ***##添加文本：***  LTSR006  LTSR007 |

## Step3.分发

|  |
| --- |
| **$** *cd ~/modules*  **$** *scp -r flink-1.9.0 smartai@LTSR006:~/modules/*  **$** *scp -r flink-1.9.0 smartai@LTSR007:~/modules/* |

## Step4.添加环境变量

|  |
| --- |
| **$** *vi ~/.bashrc*  ***# 尾端添加如下配置：***  export FLINK\_HOME=/home/smartai/modules/flink-1.9.0  export PATH=$FLINK\_HOME/bin:$PATH  ***# 分发其余节点***  **$** *scp -r ~/.bashrc smartai@LTSR006:~/*  **$** *scp -r ~/.bashrc smartai@LTSR007:~/*  **$** *source ~/.bashrc* ***##各节点分别执行*** |

## Step5.启停服务

|  |
| --- |
| ***# master节点上启动集群***  **$***cd ~/modules/flink-1.9.0*  **$ *.****/bin/start-cluster.sh*  **$ *.****/bin/stop-cluster.sh*  **$** *jps*  ***#查看进程***  ***# 查看job列表***  **$ *.****/bin/flink list*  ***# 停止job***  **$ *.****/bin/flink cancel ${JobID}* |

## Step6.验证

|  |
| --- |
| ***# 访问入口（谷歌浏览器） ：***[***http://192.168.0.15:8081***](http://192.168.0.15:8081/) |

## Step7.示例

|  |
| --- |
| ***# LTSR006节点添加监听端口***  **$** *nc -l 9000*  ***# 在LTSR005节点运行flink自带的”单词计数”示例***  **$***cd ~/modules/flink-1.9.0*  **$** *./bin/flink run examples/streaming/SocketWindowWordCount.jar --hostname LTSR006 --port 9000*  ***# 在Running Jobs界面可以看到相应的任务生成***    ***# 在监听端口输入一些单词后回车（如下图）***    ***# 回到task Managers页签下，从task的Stdout可以看到单词数量统计结果*** |

# 问题

## GitHub下载慢问题

（1）修改HOSTS文件，添加github相关站点域名。

（2）利用码云作为中转。（推荐）

（3）利用手机4G网络。

参考：https://blog.csdn.net/github\_37847975/article/details/86477343