电信用户画像平台

环境

IP 设置固定IP

虚拟机使用Nat模式

输入 以管理员账号执行

```
vim /etc/sysconfig/network-scripts/ifcfg-ens33
```

编辑里面的内容

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO="none"
IPADDR=192.168.220.188
NETMASK=255.255.255.0
GATEWAY=192.168.220.2
DNS1=114.114.114.114
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=0d87d2ea-92fd-4855-8cd6-6f0041f05209
DEVICE=ens33
ONBOOT=yes
```

主要修改的内容是

```
BOOTPROTO="none"
IPADDR=192.168.220.188
NETMASK=255.255.255.0
GATEWAY=192.168.220.2
DNS1=114.114.114.114
UUID=0d87d2ea-92fd-4855-8cd6-6f0041f05209
```

保存，重启网卡

```
systemctl restart network
```

hadoop环境搭建

1. hadoop安装文件

   ```
   hadoop-3.3.1.tar.gz
   ```

2. 将软件上传至 ~/tools/目录下
3. 准备工作

   ```
   tar -zxvf hadoop-3.3.1.tar.gz -C ~/training/
   ```

   编辑环境变量

   vim ~/.bash_profile

   ```
   HADOOP_HOME=/root/training/hadoop-3.3.1
   export HADOOP_HOME

   PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
   export PATH
   ```

   source ~/.bash_profile

4. 配置hadoop配置文件

   编辑/etc/hosts文件

   ```
   192.168.220.188 node110
   ```

   伪分布模式：一台
   特点：具备Hadoop的所有功能，在单机模拟一个分布式的环境
   HDFS：NameNode、DataNode、SecondaryNameNode
   Yarn：ResourceManager、NodeManager

   - hadoop-env.sh

     ```
     export HADOOP_HOME=/root/training/jdk1.8.0_162
     ```

   - hdfs-site.xml

□

```
<!--数据块的冗余度-->
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>

<!--禁用了HDFS的权限检查-->
<property>
    <name>dfs.permissions</name>
    <value>false</value>
</property>
```

- core-site.xml

```
<!--NameNode的地址-->
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://node110:9000</value>
</property>

<!--HDFS对应的操作系统目录-->
<property>
    <name>hadoop.tmp.dir</name>
    <value>/root/training/hadoop-3.3.1/tmp</value>
</property>
```

- mapred-site.xml

```
 <!--MR运行的框架-->
 <property>
     <name>mapreduce.framework.name</name>
     <value>yarn</value>
 </property>
```

- yarn-site.xml

```
<!--RM的地址-->
<property>
    <name>yarn.resourcemanager.hostname</name>
    <value>node110</value>
</property>
<!--MapReduce执行的方式是洗牌-->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
```

对namenode格式化

```
hdfs namenode -format
```

配置免密登录

```
ssh-keygen

ssh-copy-id -i .ssh/id_rsa.pub root@node110
```

配置权限

```
vim /etc/profile


export HDFS_NAMENODE_USER=root
export HDFS_DATANODE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
export YARN_RESOURCEMANAGER_USER=root
export YARN_NODEMANAGER_USER=root


source /etc/profile
```

Web Console

```
HDFS:http://192.168.220.188:50070
Yarn:http://192.168.120.188:8088
```

执行WordCount:

```
hadoop jar hadoop-mapreduce-examples-2.7.3.jar wordcount /input/data.txt
/output/wc
```

加载不到主类，解决办法

```
hadoop classpath

/root/training/hadoop-3.3.1/etc/hadoop:/root/training/hadoop-
3.3.1/share/hadoop/common/lib/*:/root/training/hadoop-
3.3.1/share/hadoop/common/*:/root/training/hadoop-
3.3.1/share/hadoop/hdfs:/root/training/hadoop-
3.3.1/share/hadoop/hdfs/lib/*:/root/training/hadoop-
3.3.1/share/hadoop/hdfs/*:/root/training/hadoop-
3.3.1/share/hadoop/mapreduce/*:/root/training/hadoop-
3.3.1/share/hadoop/yarn:/root/training/hadoop-
3.3.1/share/hadoop/yarn/lib/*:/root/training/hadoop-3.3.1/share/hadoop/yarn/*
```

修改yarn-siet.xml文件，该文件在hadoop的安装目录/etc/hadoop 下

在yarn-siet.xml文件的configuration中添加以下配置

```
<property>
    <name>yarn.application.classpath</name>
    <value>
        // 第一步的结果
    </value>
</property>
```

SQOOP配置

安装sqoop的前提是已经具备java和hadoop的环境。

上传sqoop安装包至服务端

解压

```
tar -zxvf sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz -C ~/training/
```

编辑配置文件

```
cd /root/training/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/conf
cp sqoop-env-template.sh sqoop-env.sh

vim sqoop-env.sh
# 加入以下内容
export HADOOP_COMMON_HOME=/root/training/hadoop-3.3.1
export HADOOP_MAPRED_HOME=/root/training/hadoop-3.3.1
```

加入mysql的jdbc驱动包

```
rz -E   # mysql-connector-java-5.1.38.jar
```

将sqoop 添加进入path

```
vim ~/.bash_profile

SQOOP_HOME=/root/training/sqoop-1.4.6.bin__hadoop-2.0.4-alpha
export SQOOP_HOME

PATH=$SQOOP_HOME/bin:$PATH
export PATH
```

1. 测试联通MySQL

    ```
    sqoop list-databases \
    --connect jdbc:mysql://localhost:3306/ \
    --username root --password root
    ```

2. 测试从MySQL导入数据到HDFS

```
sqoop import \
--connect jdbc:mysql://node110:3306/emps \
--username root \
--password root \
--target-dir /sqoopresult \
--table emp \
--m 1
```

PySpark环境

1. 上传并解压Spark的安装包
2. 安装anaconda环境

注意在 /root/trainning/anaconda中安装 Anaconda3-2021.05-Linux-x86_64.sh

```
sh Anaconda3-2021.05-Linux-x86_64.sh
```

3. 配置conda源

```
vim ~/.condarc

channels:
  - defaults
show_channel_urls: true
default_channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

4. 创建pyspark

```
conda create -n pyspark python=3.8

conda activate pyspark

pip install pyspark pymysql jieba
```

5. 配置Spark环境

环境变量

配置Spark由如下5个环境变量需要设置

配置Spark由如下5个环境变量需要设置

```
          -  SPARK_HOME: 表示Spark安装路径在哪里
          -  PYSPARK_PYTHON: 表示Spark想运行Python程序, 那么去哪里找python执行器
          -  JAVA_HOME: 告知Spark Java在哪里
          -  HADOOP_CONF_DIR: 告知Spark Hadoop的配置文件在哪里
          -  HADOOP_HOME: 告知Spark  Hadoop安装在哪里
```

这5个环境变量 都需要配置在: `/etc/profile`中

```
vim /etc/profile

export JAVA_HOME=/root/training/jdk1.8.0_162
export HADOOP_HOME=/root/training/hadoop-3.3.1
export SPARK_HOME=/root/training/spark
export PYSPARK_PYTHON=/root/training/anaconda/envs/pyspark/bin/python3.8
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH

source /etc/profile
```

PYSPARK_PYTHON和 JAVA_HOME 需要同样配置在: `/root/.bashrc`中

```
export JAVA_HOME=/root/training/jdk1.8.0_162
export PYSPARK_PYTHON=/root/training/anaconda/envs/pyspark/bin/python3.8


source ~/.bashrc
```

此时，可以

执行 bin/pyspark

或者使用

```
bin/spark-submit --master local[*]
/root/training/spark/examples/src/main/python/pi.py 10
```

配置spark连接MySQL

需要将jdbc-jar包传入

```
/root/training/anaconda/envs/pyspark/lib/python3.8/site-packages/pyspark/jars/
```

实验内容

1. 输入数据

创建数据表库,执行如下SQL

```sql
create database all_base_data;
use all_base_data;

 CREATE TABLE `calls` (
  `date` date DEFAULT NULL,
  `phone` bigint(20) DEFAULT NULL,
  `desc_num` text,
  `count` int(11) DEFAULT NULL,
  `duration` int(11) DEFAULT NULL,
  `create_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf-8


 CREATE TABLE `call_flags` (
  `desc_num` text,
  `insitution` text,
  `tag` varchar(100),
  `create_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf-8

 CREATE TABLE `apps` (
  `date` date DEFAULT NULL,
  `phone` bigint(20) DEFAULT NULL,
  `desc_num` text,
  `count` int(11) DEFAULT NULL,
  `duration` int(11) DEFAULT NULL,
  `create_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf-8

 CREATE TABLE `app_flags` (
  `desc_num` text,
  `insitution` text,
  `tag` varchar(100),
  `create_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf-8

 CREATE TABLE `sms` (
  `date` date DEFAULT NULL,
  `phone` bigint(20) DEFAULT NULL,
  `desc_num` text,
  `count` int(11) DEFAULT NULL,
  `create_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf-8

 CREATE TABLE `sms_flags` (
  `desc_num` text,
  `insitution` text,
  `tag` varchar(100),
  `create_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf-8
```

sqoop 迁移

sqoop数据迁移脚本

```
#!/bin/bash

db_date=$2
echo $db_date
db_name=dx

import_data(){
 /root/training/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/bin/sqoop import \
--connect jdbc:mysql://node110:3306/$db_name \
--username root \
--password root \
--target-dir /input/dx/db/$db_date/$1 \
--delete-target-dir \
--num-mappers 1 \
--query "$2"' and $CONDITIONS;'
}
import_calls(){
 import_data "call_details" "select date,phone,desc_num,count,duration,create_date from call_details where create_date='$db_date' and 1=1"
}
import_call_flags(){
 import_data "call_flags" "select desc_num,insitution,tag,create_date from call_flags where create_date='$db_date' and 1=1"
}
import_sms(){
 import_data "sms_details" "select date,phone,desc_num,count,create_date from sms_details where create_date='$db_date' and 1=1"
}
import_sms_flags(){
 import_data "sms_flags" "select desc_num,insitution,tag,create_date from sms_flags where create_date='$db_date' and 1=1"
}
import_apps(){
 import_data "app_details" "select date,phone,desc_num,count,duration,create_date from app_details where create_date='$db_date' and 1=1"
}
import_app_flags(){
 import_data "app_flags" "select desc_num,insitution,tag,create_date from app_flags where create_date='$db_date' and 1=1"
}


case $1 in
  "alls")
     import_calls
     import_call_flags
     import_sms
```

```
      import_sms_flags
      import_apps
      import_app_flags
;;
esac
```

Azkaban配置

mkdir /export/servers/azkaban

tar -zxvf azkaban-solo-server-0.1.0-SNAPSHOT.tar.gz –C /export/servers/azkaban/

vim conf/azkaban.properties

default.timezone.id=Asia/Shanghai #修改时区

vim plugins/jobtypes/commonprivate.properties

添加：memCheck.enabled=false

azkaban默认需要3G的内存，剩余内存不足则会报异常

cd azkaban-solo-server-0.1.0-SNAPSHOT/

bin/start-solo.sh

访问Web Server=>http://192.168.220.188:8081/ 默认用户名密码**azkaban**

Azkaban job 编写，调用

1. SQOOP
2. Pyspark