

hadoop-wordcount

211870287 丁旭

September 2023

1 hadoop 环境配置

1. 远程连接 ECS 实例

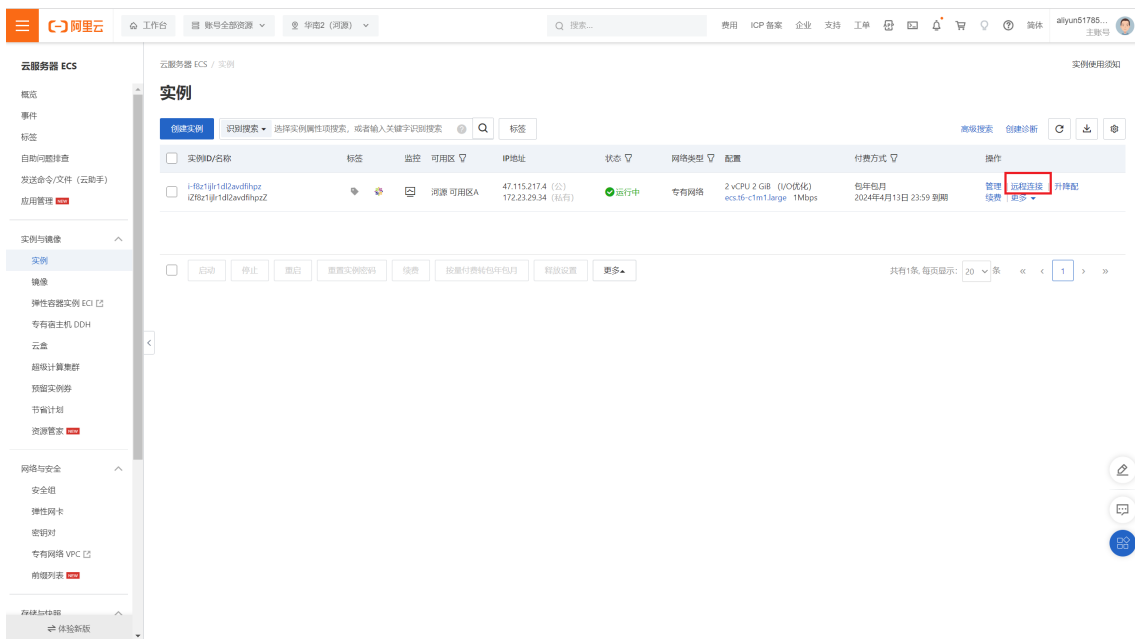


图 1: 云服务器控制台界面

2. 安装 jdk1.8

(a) 下载 JDK 安装包

```
wget https://download.java.net/openjdk/jdk8u41/ri/  
(接)openjdk-8u41-b04-linux-x64-14_jan_2020.tar.gz
```

(b) 解压 JDK 安装包

```
tar -zxvf openjdk-8u41-b04-linux-x64-14_jan_2020.tar.gz
```

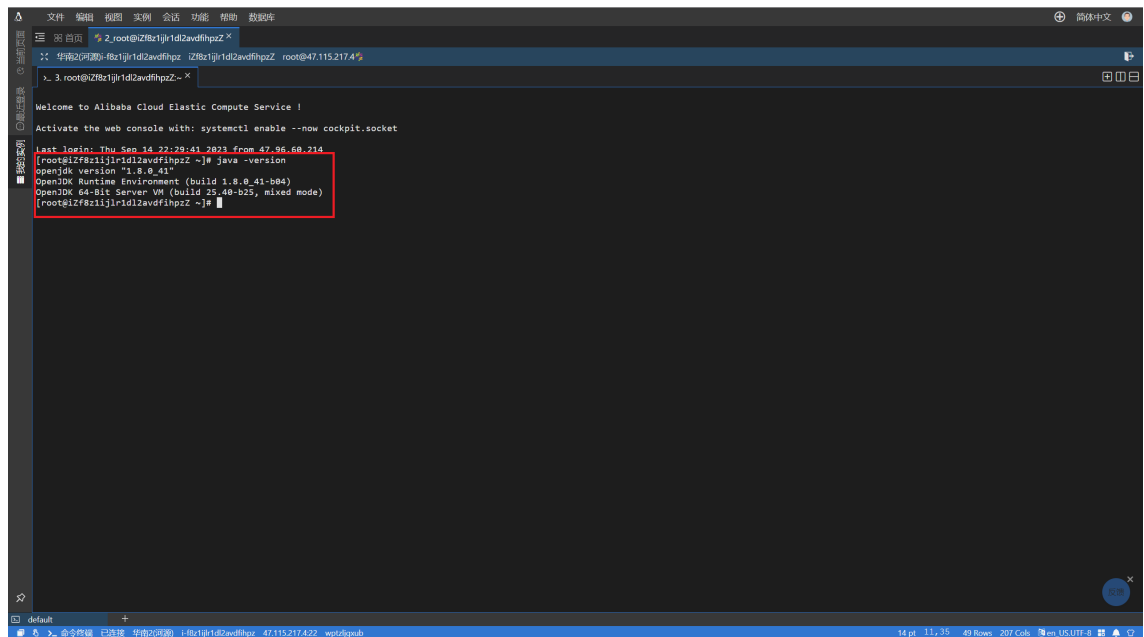
(c) 移动并重命名 JDK 安装包

```
mv java-se-8u41-ri/ /usr/java8
```

(d) 配置 java 环境

```
echo 'export JAVA_HOME=/usr/java8' >> /etc/profile  
echo 'export PATH=$PATH:$JAVA_HOME/bin' >> /etc/profile  
source /etc/profile
```

(e) 查看 java 是否安装



```
Welcome to Alibaba Cloud Elastic Compute Service !  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last login: Thu Sep 14 22:29:41 2023 from 47.86.60.214  
[root@izf8z1j1r1d12avdflhpz ~]# java -version  
openjdk version "1.8.0_41"  
OpenJDK Runtime Environment (build 1.8.0_41-b04)  
OpenJDK 64-bit Server VM (build 25.40-b25, mixed mode)  
[root@izf8z1j1r1d12avdflhpz ~]#
```

图 2: java 验证: java -version

3. 安装 Hadoop

(a) 下载 Hadoop 安装包

```
wget https://mirrors.bfsu.edu.cn/apache/hadoop/common/hadoop-2.10.1/1
```

(b) 解压 Hadoop 安装包至/opt/hadoop

```
tar -zxvf hadoop-2.10.1.tar.gz -C /opt/  
mv /opt/hadoop-2.10.1 /opt/hadoop
```

(c) 配置 Hadoop 环境变量

```
echo 'export HADOOP_HOME=/opt/hadoop/' >> /etc/profile  
echo 'export PATH=$PATH:$HADOOP_HOME/bin' >> /etc/profile  
echo 'export PATH=$PATH:$HADOOP_HOME/sbin' >> /etc/profile  
source /etc/profile
```

(d) 修改配置文件 yarn-env.sh 和 hadoop-env.sh

```
echo "export JAVA_HOME=/usr/java8" >>  
/opt/hadoop/etc/hadoop/yarn-env.sh  
  
echo "export JAVA_HOME=/usr/java8" >>  
/opt/hadoop/etc/hadoop/hadoop-env.sh
```

(e) 验证 Hadoop 是否安装成功

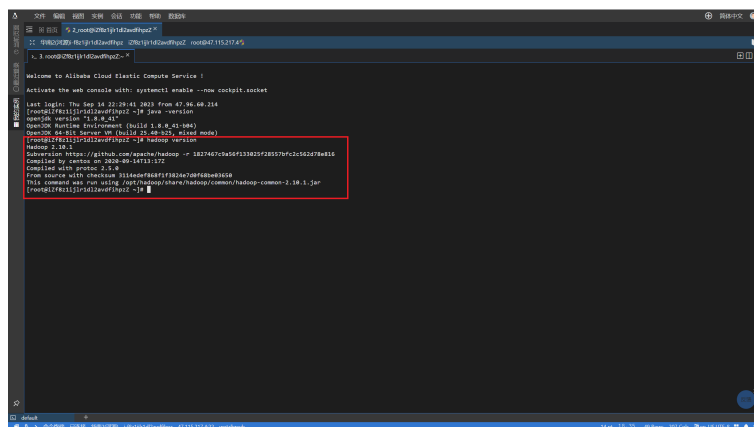


图 3: hadoop 安装验证

4. 配置 Hadoop

进入编辑界面

```
vim /opt/hadoop/etc/hadoop/core-site.xml
```

在<configuration></configuration>节点内，插入如下内容。

```
<property>
    <name>hadoop.tmp.dir</name>
    <value>file:/opt/hadoop/tmp</value>
    <description>location to store temporary files</description>
</property>
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
</property>
```

进入编辑界面

```
vim /opt/hadoop/etc/hadoop/hdfs-site.xml
```

在<configuration></configuration>节点内，插入如下内容

```
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>
<property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/opt/hadoop/tmp/dfs/name</value>
</property>
<property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/opt/hadoop/tmp/dfs/data</value>
</property>
```

5. 配置 SSH 免密登录

创建公钥和私钥

```
ssh-keygen -t rsa
```

执行以下命令，将公钥添加到 authorized_keys 文件中

```
cd .ssh  
cat id_rsa.pub >> authorized_keys
```

2 启动 hadoop

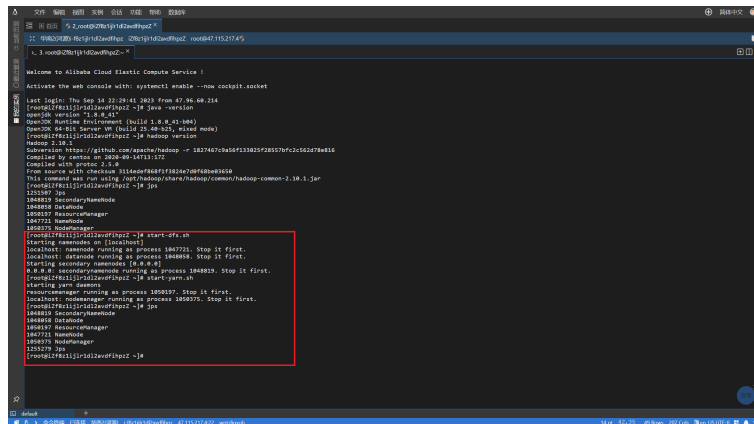
1. 执行以下命令，初始化 namenode

```
hadoop namenode -format
```

2. 依次执行以下命令，启动 Hadoop

```
start-dfs.sh  
start-yarn.sh
```

3. 查看成功启动的进程



```
Welcome to Alibaba Cloud Elastic Compute Service !  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last login: Thu Sep 14 22:29:41 2023 from 47.96.69.214  
[root@iZf6l1j1td2woflhp2 ~]# java -version  
java version "1.8.0_171"  
OpenJDK Runtime Environment (build 1.8.0_171-b01)  
OpenJDK 64-Bit Server VM (build 25.40-b01, mixed mode)  
[root@iZf6l1j1td2woflhp2 ~]# hadoop version  
Hadoop 3.18.1  
Source code: https://github.com/apache/hadoop -r 187467b465f33829f28537bf3c5426784816  
Compiled by centos on 2023-09-14T13:17  
Compiled with OpenJDK 1.8.0_171 on Linux/amd64  
From source with checkout 818a6e889f1f1617048a0b8108  
This command was run using /opt/hadoop/share/hadoop/common/hadoop-common-3.18.1.jar  
[root@iZf6l1j1td2woflhp2 ~]# jps  
115587 jps  
184810 SecondaryNameNode  
184809 DataNode  
184807 ResourceManager  
184771 NameNode  
[root@iZf6l1j1td2woflhp2 ~]# start-dfs.sh  
Starting namenodes on [localhost]  
localhost: namenode running as process 184771. Stop it first.  
localhost: datanode running as process 184808. Stop it first.  
Starting secondary namenodes [184810]  
184810 SecondaryNameNode running as process 184810. Stop it first.  
[root@iZf6l1j1td2woflhp2 ~]# start-yarn.sh  
Starting yarn daemons  
ResourceManager running as process 184807. Stop it first.  
localhost: ResourceManager running as process 184807. Stop it first.  
[root@iZf6l1j1td2woflhp2 ~]# jps  
115587 jps  
184810 SecondaryNameNode  
184809 DataNode  
184807 ResourceManager  
184771 NameNode  
[root@iZf6l1j1td2woflhp2 ~]#
```

图 4: hadoop 启动验证

4. 打开浏览器访问 <http://<ECS 公网 IP>:8088> 和 <http://<ECS 公网 IP>:50070>

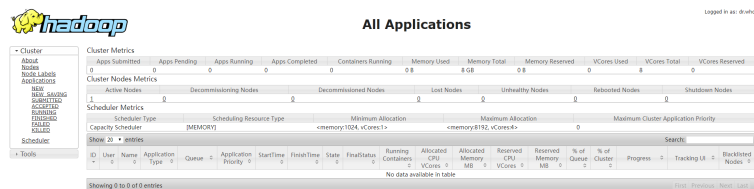


图 5: <http://<ECS 公网 IP>:8088>

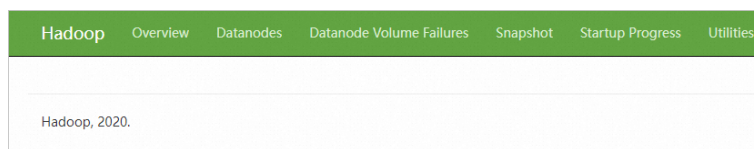


图 6: <http://<ECS 公网 IP>:50070>

3 scala 环境配置

1. 进入到/usr/local/share 文件夹

```
cd /usr/local/share
```

2. 安装包下载

```
wget https://downloads.lightbend.com/scala/2.12.6/scala-2.12.6.tgz
```

3. 解压文件

```
tar -xvzf scala-2.12.6.tgz
```

4. 添加环境变量

```
vim /etc/profile
export PATH="$PATH:/usr/local/share/scala-2.12.6/bin"
source /etc/profile
```

5. scala 环境检验

```
[root@izf8z1ijlr1d12avdfihpzZ ~]# scala
Welcome to Scala 2.12.6 (OpenJDK 64-Bit Server VM, Java 1.8.0_41).
Type in expressions for evaluation. Or try :help.

scala> println("Hello, World!")
Hello, World!

scala>

scala>
```

图 7: scala: hello,world!

4 在 hadoop 环境下编写 scala 的 wordcount

1. 启动 hadoop

```
cd /usr/hadoop/hadoop-2.6.2/
sbin/start-dfs.sh
sbin/start-yarn.sh
```

2. 创建本地数据文件

```
cd ~/
mkdir ~/file
cd file
echo "Hello World , welcome to Big Data Analysis" > test1.txt
echo "Big Data Analysis is a good lesson " > test2.txt
```

3. 创建 scala 代码

```
import java.io.File
import scala.io.Source
object WordCount {
  def main(args: Array[String]): Unit = {
    val dirfile=new File(args(0))
    val files=dirfile.listFiles
    for(file <- files) println(file)
    val listFiles=files.toList
    val wordsMap=scala.collection.mutable.Map[String,Int]()
    listFiles.foreach( file =>Source.fromFile(file).getLines().
    foreach(line=>line.split(" ")).
```

```

        foreach(
            word=>{
                if (wordsMap.contains(word)) {
                    wordsMap(word)+=1
                } else {
                    wordsMap+=(word->1)
                }
            }
        )
    )
    println(wordsMap)
    for((key,value)<-wordsMap) println(key+": "+value)
}
}

```

4. 将数据文件传到 HDFS 的 input 目录下

```

hadoop fs -mkdir /input
hadoop fs -put ~/file/test*.txt /input
sbin/start-yarn.sh

```

5. 运行程序

第一种运行方式

```

hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce
-examples-2.10.1.jar wordcount /input /output

```

第二种运行方式

```

编写 build.sh
#!/bin/bash

```

```

# 设置输出 Jar 文件的名称
JAR_NAME="WordCount.jar"

```



```

# 编译 Scala 文件
scalac WordCount.scala

# 创建临时目录并将编译生成的 .class 文件复制到该目录中
mkdir tmp
find . -name '*.class' -exec cp {} tmp/ \;

# 创建空的 MANIFEST.MF 文件
touch MANIFEST.MF

# 在 MANIFEST.MF 文件中写入主类的信息，替换 "com.example.MainClass" 为你
echo "Main-Class: WordCount" >> MANIFEST.MF

# 打包 .class 文件和 MANIFEST.MF 文件为 Jar 文件
jar cfm $JAR_NAME MANIFEST.MF -C tmp .

# 删除临时目录和 MANIFEST.MF 文件
rm -rf tmp MANIFEST.MF

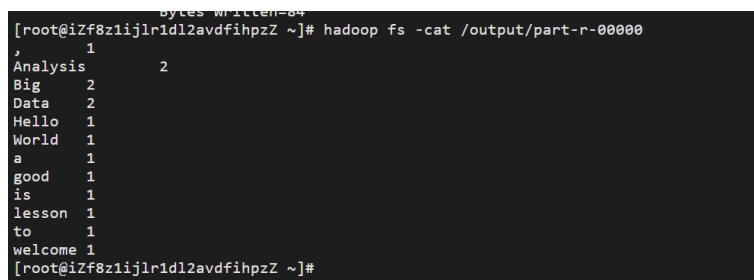
运行 build.sh
sh build.sh
配置 hadoop 的 scala 环境
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/usr/local/share
/scala-2.12.6/lib/*
source ~/.bashrc

在 hadoop 环境运行 scala 的 jar 包
hadoop jar /root/wordcount/WordCount.jar /root/file

```


6. 查看结果

```
hadoop fs -cat /output/part-r-00000
```



```
[root@izf8z1ijlr1dl2avdfihpzZ ~]# hadoop fs -cat /output/part-r-00000
Analysis      2
Big           2
Data          2
Hello         1
World         1
a             1
good          1
is            1
lesson        1
to            1
welcome       1
[root@izf8z1ijlr1dl2avdfihpzZ ~]#
```

图 10: 统计 test1 文件和 test2 文件内所有单词的词频

7. 结果验证

Word	Count
Analysis	2
Big	2
Data	2
Hello	1
World	1
a	1
good	1
is	1
lesson	1
to	1
welcome	1

输出正确，程序运行正常

5 清理——终止实例服务

在云服务器管理控制台停止用例

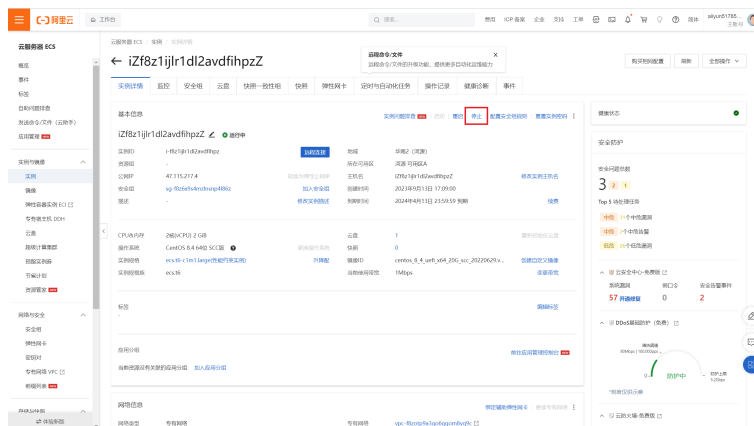


图 11: 终止实例

6 过程中的挑战及如何克服

1. 先需要找出哪个供应商有免费的学生服务，最后在阿里云中免费申请了 7 个月的云
2. 不知道什么是 bda 环境，通过查询资料，最终选择 hadoop 作为作业的 bda 环境
3. 配置环境是最复杂也是最繁琐的一项工程，因为我对这些东西一无所知，通过查询阿里云的官方文档已经搜索 csdn 和 chatgpt，才最终配置好了 hadoop、scala。
4. 不懂如何在 hadoop 中运行 scala 代码，通过查询资料，渐渐明白了 hadoop 和 scala 的关系，将 scala 代码打包成 jar，然后在 Hadoop 中运行。