docker搭建Hadoop+Spark+Hive+HBase操作说明

1. 拉取镜像&进入容器

因为整个镜像比较大，大概有3个多G，所以我将镜像放在我的阿里云上，这样大家可以比较快速地拉取镜像，以后需要使用比较大的镜像时，也可以像我这么做。

```
$ docker pull registry.cn-hangzhou.aliyuncs.com/fbdp2019-hadoop/hadoop:2.0.0
```

拉取完成之后，`docker images`应该可以看到这个镜像。我本地是这样的：

因为我们需要搭建一个集群，节点之间需要相互通信，所以需要构架一个虚拟网络，实现类似于局域网的功能

```
$ sudo docker network create --driver=bridge hadoop
```

以上命令创建了一个名为 Hadoop 的虚拟桥接网络，该虚拟网络内部提供了自动的DNS解析服务。

```
$ sudo docker network ls
```

执行该命令可以看到我们刚刚创建的名为 `hadoop` 的虚拟桥接网络。

接下来我们就可以创建容器并进入容器了。

```
$ docker run -it --network hadoop -h "h01" --name "h01" -p 9870:9870 -p
8088:8088 -p 8080:8080 imageREPOSITORY /bin/bash
```

- `--network hadoop`：使用我们上面创建的名为 `hadoop` 的虚拟桥接网络。
- `-h "h01"`:对应于我配置hadoop时`workers`里面设置的`h01`。
- `-p`：将容器的三个端口`9870,8088,8080`映射到本地，这样我们用`localhost:[端口号]`就可以用dashboard监控我们的服务。
- `imageREPOSITORY`:指定镜像，这里应该就是`registry.cn-hangzhou.aliyuncs.com/fbdp2019-hadoop/hadoop:2.0.0`。

进入容器之后会看到这样的界面

```
* Starting OpenBSD Secure Shell server sshd

                [ OK ]
root@h01:/#
```

2.使用方法

首先说明一下，我本身配的是hadoop集群模式，但是我在主机上搭建三个节点的集群时，因为内存资源不够，所以hadoop自带的wordcount都跑不动，所以放弃了。现在也是集群模式，只不过集群里只有`h01`一个节点。如果需要集群模式的同学请参考这篇博客。只需修改`workers`即可。

```
root@h01:~# cd /usr/local/
root@h01:/usr/local# ls
bin  etc  games  hadoop  hive  include  lib  man  sbin  share  spark-2.4.4
src
```

进入`/usr/local`目录可以看到有`hadoop`,`spark-2.4.4`,`hive`,hadoop的版本是`3.2.1`,spark的版本是`2.4.4`,hive的版本是`3.1.2`,还有`jdk`,`scala`,`mysql`等都已经安装好了。

hadoop使用

执行下面的指令就行

```
root@h01:/usr/local# cd hadoop/bin
root@h01:/usr/local/hadoop/bin# ./hadoop namenode -format    //格式化操作
root@h01:/usr/local/hadoop/bin# cd ../sbin
root@h01:/usr/local/hadoop/sbin# ./start-all.sh     //启动
Starting namenodes on [h01]
Starting datanodes
Starting secondary namenodes [h01]
Starting resourcemanager
Starting nodemanagers
root@h01:/usr/local/hadoop/sbin# jps       //查看
13158 Jps
12663 ResourceManager
12041 NameNode
12809 NodeManager
12411 SecondaryNameNode
12205 DataNode
```

spark使用

执行下面的指令就行

```
root@h01:/usr/local/hadoop/sbin# cd /usr/local/spark-2.4.4/
root@h01:/usr/local/spark-2.4.4# cd sbin
root@h01:/usr/local/spark-2.4.4/sbin# ./start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark-
2.4.4/logs/spark--org.apache.spark.deploy.master.Master-1-h01.out
```

```
h01: starting org.apache.spark.deploy.worker.Worker, logging to
/usr/local/spark-2.4.4/logs/spark-root-org.apache.spark.deploy.worker.Worker-
1-h01.out
root@h01:/usr/local/spark-2.4.4/sbin# jps
15393 Worker
14963 NodeManager
15318 Master
14809 ResourceManager
14185 NameNode
14556 SecondaryNameNode
14349 DataNode
15470 Jps
```

hive使用

- 因为我在测试的时候会遇到路径can not find 的问题，所以首先执行`source /etc/profile`命令，让该文件中的环境变量生效。
- 然后要把`mysql`启动起来，这里我们采用MySQL数据库保存Hive的元数据。

  ```
  $ chown -R mysql:mysql /var/lib/mysql /var/run/mysqld       #做一下初始化
  $ service mysql start                                       #启动mysql服务
  $ root@h01:~# netstat -tap | grep mysql                     #检查mysql状态
  tcp        0      0 localhost:mysql         *:*                     LISTEN
       -
  $ mysql -u root -p
  PASSWORD: 123456
  ```

  这样mysql就启动完成了。

- 接下来执行下面的执行就行

```
root@h01:/usr/local/spark-2.4.4/sbin# cd /usr/local/hive/bin
root@h01:/usr/local/hive/bin# ./hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-
2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = c032ff75-ceaa-4e8a-abe2-065de03d2882

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-
common-3.1.2.jar!/hive-log4j2.properties Async: true
```

注意可能会出现下面这样很多的warning，不影响hive使用，可以忽略(这里我也配了很久。。。。)

```
Sat Nov 30 09:15:57 UTC 2019 WARN: Establishing SSL connection without
server's identity verification is not recommended. According to MySQL 5.5.45+,
5.6.26+ and 5.7.6+ requirements SSL connection must be established by default
if explicit option isn't set. For compliance with existing applications not
using SSL the verifyServerCertificate property is set to 'false'. You need
either to explicitly disable SSL by setting useSSL=false, or set useSSL=true
and provide truststore for server certificate verification.
```

首次进去，执行效果应该是这样的

```
hive> show databases;
OK
default
tc
Time taken: 0.496 seconds, Fetched: 2 row(s)
hive> exit;
```

hbase使用

尝试安装hbase很多次，HMaster总是启动失败，所以找了一个已有的hbase，这是GitHub地址。

```
 $ docker pull registry.cn-hangzhou.aliyuncs.com/fbdp2019-hadoop/hadoop-
hbase:1.0.0
 $ docker run -ti harisekhon/hbase
```

这样就可以直接使用，具体的操作说明请参考这篇博客。