Spark-2.3.2-bin-hadoop2.6 镜像构建

一、 实现目的

基于容器技术，搭建 Spark-Hadoop 开发环境，以供开发人员快速掌握并应用到实际开发中，有关 Spark 的优点等知识这里不再赘述。暂且把优点总结如下：

1）技术积累

2）省时省力省钱

二、 环境准备

1） sequenceiq/docker-spark 源码

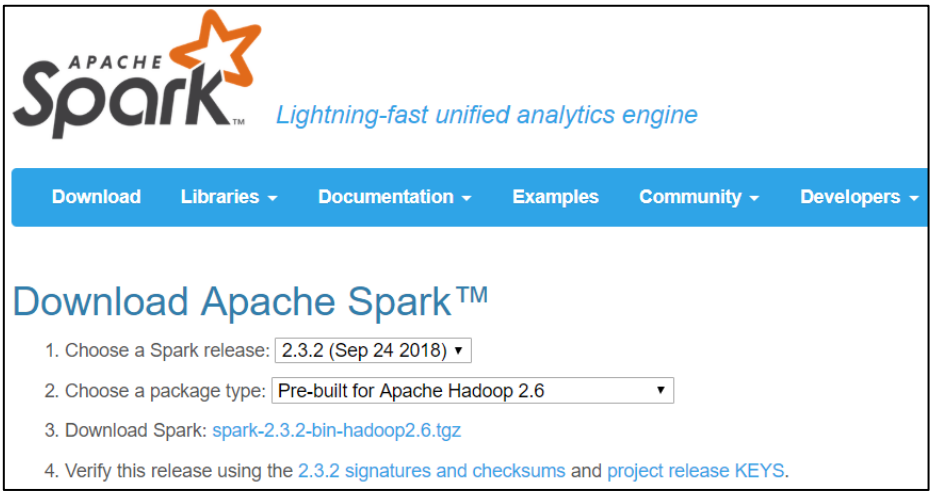
下载 sequenceiq/ docker-spark 镜像构建源码，下载命令如下：

git clone https://github.com/sequenceiq/docker-spark

2） sequenceiq/hadoop-docker:2.6.0 官网镜像

docker pull sequenceiq/hadoop-docker:2.6.0

3） spark-2.3.2-bin-hadoop2.6.tgz



4) jdk-8u25-linux-x64.tar.gz

请自行下载。

三、 镜像构建

1) 目录结构

基础目录结构如下图所示：

```
[root@localhost docker-spark]# tree
.
├── bootstrap.sh
├── Dockerfile
├── jdk-8u25-linux-x64.tar.gz
├── LICENSE
├── README.md
├── spark-2.3.2-bin-hadoop2.6.tgz
├── yarn-remote-client
│   ├── core-site.xml
│   └── yarn-site.xml
└── yarn-site.xml
```

其中，docker-spark 为 sequenceiq/docker-spark 源码目录，Dockerfile 及 yarn-site.xml 如下附件。


yarn-site.xml


Dockerfile

2) 构建命令

docker build --rm -t kngines/spark:2.3.2 .

3) 结果输出

```
Complete!
Removing intermediate container 4304bb84242e
 ---> b64f1cc9f4b6
Step 20/20 : ENTRYPOINT ["/etc/bootstrap.sh"]
 ---> Running in 8b5ad162196e
Removing intermediate container 8b5ad162196e
 ---> 45ad0468703a
Successfully built 45ad0468703a
Successfully tagged kngines/spark:2.3.2
[root@localhost docker-spark]# docker images
REPOSITORY                                  TAG                 IMAGE ID            CREATED             SIZE
kngines/spark                               2.3.2               45ad0468703a        10 seconds ago      3.05GB
```

4） 创建容器

docker run -it -p 8088:8088 -p 8042:8042 -p 4040:4040 -h sandbox kngines/spark:2.3.2 bash

5） 运行模式

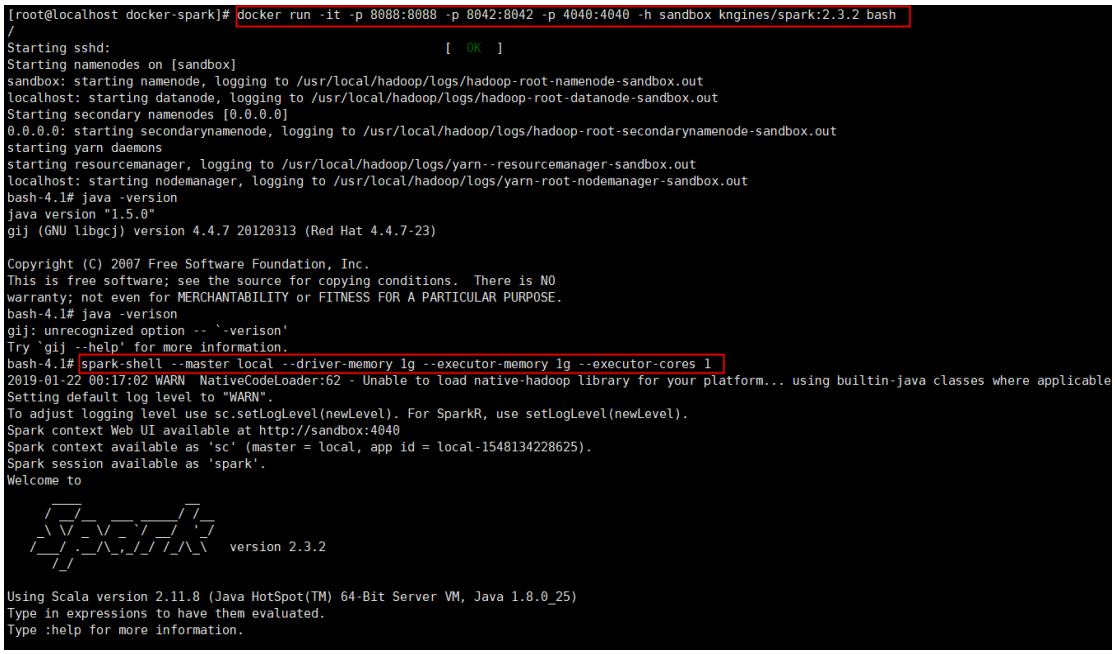
1. Local 模式命令

spark-shell --master local --driver-memory 1g --executor-memory 1g --executor-cores 1

2. Yarn 模式命令

spark-shell --master yarn --deploy-mode client --driver-memory 512m --executor-memory 512m --executor-cores 1

6） 运行效果



四、 测试运行

1) 简单测试

```
scala> sc.parallelize(1 to 1000).count()
res1: Long = 1000

scala>
```

2) 复杂测试（Hadoop-Spark）

```
bash-4.1# ls -l
total 4
-rw-r--r--. 1 root root 37 Jan 22 00:48 hello.txt
bash-4.1# hadoop fs -mkdir -p /spark
bash-4.1# hadoop fs -put hello.txt /spark
```

```
bash-4.1# spark-shell --master yarn --deploy-mode client --driver-memory 512m --executor-memory 512m --executor-cores 1
2019-01-22 00:44:44 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://sandbox:4040
Spark context available as 'sc' (master = yarn, app id = application_1548134203925_0002).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.3.2
      /_/

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_25)
Type in expressions to have them evaluated.
Type :help for more information.

scala> sc.textFile("/spark/hello.txt").flatMap(_.split(",")).map((_,1)).reduceByKey(_+_).saveAsTextFile("/spark/out")
```

```
bash-4.1# hadoop fs -cat /spark/out/p*
(jump,5)
(you,2)
(i,2)
```

说明：

sc 是 SparkContext 对象，提交 spark 程序的入口

textFile("/spark/hello.txt")是从 hdfs 中读取数据

flatMap(_.split(" "))先 map 再压平

map((_,1))将单词和 1 构成元组

reduceByKey(_+_)按照 key 进行 reduce，并将 value 累加

saveAsTextFile("/spark/out")将结果写入到 hdfs 中

相关命令：

hadoop fs -mkdir -p /spark

hadoop fs -put hello.txt /spark

hadoop fs -cat /spark/out/p*

五、 Web UI

3) Spark Web UI