Move to...

[Mac][Linux] Class 2: Part 2.2 How to set up hadoop environment

注意：配置的时候如果遇到了问题，请贴输入的命令框里面的代码和结果的截图，这样老师才能准确知道问题更好的解决。

PS: 本教程针对 Mac 和 Linux 用户都适用

配置Hadoop环境

装好了docker之后下面我们就可以来配置Hadoop的环境啦。

怎么检测docker是否配置好？

$ sudo docker info

如果看到类似下面的界面就说明已经配置好。

```
➜  ~ docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 1.12.0
Storage Driver: aufs
 Root Dir: /var/lib/docker/aufs
 Backing Filesystem: extfs
 Dirs: 5
 Dirperm1 Supported: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: null host bridge overlay
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Security Options: seccomp
Kernel Version: 4.4.15-moby
Operating System: Alpine Linux v3.4
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 1.954 GiB
Name: moby
ID: 5QTR:HUY3:QHZG:52NJ:ZUEW:XG7S:VDVT:MZRJ:3PL3:6ZLN:BN5I:YHCQ
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 27
 Goroutines: 72
 System Time: 2016-09-05T23:39:32.920278293Z
 EventsListeners: 1
Registry: https://index.docker.io/v1/
Insecure Registries:
 127.0.0.0/8
```

- [ ]

  配置一个hadoop环境，该环境中有一namenode（主机），两个datanode（子机）

$ **mkdir bigdata-class2** # 创建一个目录

$ **cd bigdata-class2** # 进入 文件夹

$ `sudo docker pull joway/hadoop-cluster`# pull docker image ，接下来需要输入密码，需要管理员权限

There is no `sudo` command in Windows. The nearest equivalent is "run as # # administrator." http://stackoverflow.com/questions/9652720/how-to-run-sudo-command-in-windows .

$ `git clone https://github.com/joway/hadoop-cluster-docker` # 把 github repository 复制到本地

$ `ls` # 检测 本地有一个hadoop cluster docker 的文件夹

```
➜ bigdata-class2 ls
hadoop-cluster-docker
```

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

$ `cd hadoop-cluster-docker` # 进入到 hadoop-cluster-docker 文件夹

$ sudo ./start-container.sh # 运行， 如果你看到下面的类似的输出就恭喜你运行成功， 表示的意思是我们启动了1个name node叫做master, 2个data node 叫做hadoop-slave，**这段代码在每次使用Docker的时候必须运行**

```
➜ hadoop-cluster-docker git:(master) sudo ./start-container.sh
start hadoop-master container...
start hadoop-slave1 container...
start hadoop-slave2 container...
root@hadoop-master:~#
```

$ `./start-hadoop.sh` # 现在hadoop的环境已经被启动了，输入下行代码进行test，**每次需要使用Hadoop的时候必须运行**

$ `./run-wordcount.sh`

应该看到如下结果：

input file1.txt:
Hello Hadoop
input file2.txt:
Hello Docker
wordcount output:
Docker 1
Hadoop 1
Hello 2

```
        at org.apache.hadoop.ipc.Client.get
        at org.apache.hadoop.ipc.Client.cal
        ... 25 more

input file1.txt:
Hello Hadoop

input file2.txt:
Hello Docker

wordcount output:
Docker  1
Hadoop  1
Hello   2
```

如果你完成了以上步骤那么恭喜你你已经完成了配置工作。

----------我是萌萌的分割线----------

测试 Hadoop

-

  现在以WordCount为例，用一下指令进行编译和运行 （仍然在docker里面运行）

下面命令在docker里面运行(即下面的截图环境中)，不是在本地。

```
-rw-r--r-- 1 root root 3059 Sep  6 20:14 wc.jar
root@hadoop-master:~#
```

$ export JAVA_HOME=/usr/java/default # 配置 java home

$ export PATH=${JAVA_HOME}/bin:${PATH} # 配置路径

$ export HADOOP_CLASSPATH=/usr/lib/jvm/java-7-openjdk-amd64/lib/tools.jar # 配置hadoop 路径

$ # copy 输入下面代码，记得一定直接copy 全部到命令框里面

```
echo "
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper extends
            Mapper<Object, Text, Text, IntWritable> {

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class IntSumReducer extends
            Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
                Context context) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = new Job(conf, \"word count\");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
```

```
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
" > WordCount.java
```

$ hadoop com.sun.tools.javac.Main WordCount.java # 根据java文件生成class文件

$ jar cf wc.jar WordCount*.class # 打包class文件

$ mkdir input # 本地创建一个文件夹

$ echo "Hello Docker" >input/file2.txt # # 创建 input file2 在本地

$ echo "Hello Hadoop" >input/file1.txt # 创建 input file1 在本地h

$ hdfs dfs -mkdir -p input # hdfs 上面创建一个input文件夹

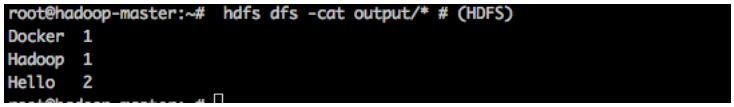
$ hdfs dfs -put ./input/* input # 把本地的input 文件夹内容上传到 hdfs上面

$ hdfs dfs -ls input # 检查input文件是否存hdfs上面了

$ hdfs dfs -rmr output # (**如果output文件夹没有创建，则不需要进行这一步，如果output已经有了，必须要运行这一步先删除他**)

$ hadoop jar wc.jar WordCount input output

$ hdfs dfs -cat output/* # (查看HDFS上面结果, 如果看到如下结果，恭喜你成功啦)



----------我是萌萌的分割线----------

本地与docker hadoop同步

1

如何直接复制本地的src文件夹到hadoop上呢?

方法一 (推荐): 使用volume

ps: 如果你使用的仍旧是旧有的镜像, 建议重新按照本文档开始操作，以切换到新镜像，若你仍旧希望在旧镜像基础上work, 同步方式只能采取方式二中的SCP来进行

新的镜像(joway/hadoop-cluster)开启了docker vulume, 如果你使用的是Mac/Linux, 请检查你本地的 ~/src/ 目录, 你可以在该目录下执行:

touch test.txt

然后在docker的terminal里执行 ls, 可以看见有一个src的目录, 执行 cd src , 再执行ls, 如果看到一个test.txt的目录, 说明你本地~/src/ 和 docker hadoop master 里的 /root/src/ 目录已经同步, 接下在你可以在本地 ~/src/ 中做任何修改, 然后在docker里去跑你的代码。

如果你使用的是Windows, 使用的是旧镜像，重新按照本文档进行操作。把需要传输的文件放入big-data-class文件夹内src目录下，文件会同步到Docker虚拟机中的/root/src目录下

想细节理解volume 请看这个文档： https://docs.docker.com/engine/tutorials/dockervolumes/

方法二: SCP

先到你本地保存代码的路径下面运行下面代码

$ pwd

得到当前文件路径， 比如老师现在命令框在 /Users/Zhaomin/Documents/workspace/InvertedIndex/src

这个路径下面所以pwd后可以看到

/Users/Zhaomin/Documents/workspace/InvertedIndex/src

首先在本地跑以下两条命令

$ ifconfig | grep inet | grep broadcast #（得到本机IP） 下面命令的 加了白色线的就是 本机IP

```
bash-3.2$  ifconfig | grep inet  | grep broadcast
        inet 100.110.214.240 netmask 0xffffc000 broadcast 100.110.255.255
```

$ echo $USER # 得到当前机器的名字

老师对应的机器名字是 **Zhaomin**

再到hadoop机器上进行复制工作， 到docker 运行terminal 里面

$ mkdir code # 创建要放入的代码的目录

$ scp -r 当前机器名字@ip_address:local_directory/* /root/code # scp 命令 是把一个机器里面的文件传输到另一个机器比如hadoop机器的命令 ,详细介绍 http://www.cnblogs.com/peida/archive/2013/03/15/2960802.html

举一个例子: scp -r Zhaomin@100.110.214.240:/Users/Zhaomin/Documents/workspace/InvertedIndex/* /root/code 如果hdfs上面已经有了就不需要了，否则会报错。

----------我是萌萌的分割线----------

其它

- ▢

你也可以配置一个hadoop环境有n个datanode（n可以自己定义,假设n=5），方法如下

$ sudo docker pull joway/hadoop-cluster

$ git clone https://github.com/joway/hadoop-cluster-docker

$ sudo docker network create --driver=bridge hadoop

$ cd hadoop-cluster-docker

$ sudo ./resize-cluster.sh 5

$ sudo ./start-container.sh 5

常见问题:

如果碰到Docker command can't connect to docker daemon这个问题，需要把 current user 加到 docker group, sudo usermod -aG docker current_user. 具体看stackoverflow http://stackoverflow.com/question...

如何判断当前terminal环境是本地环境还是docker内环境?

开头有root@hadoop-master, 则说明是在docker 的hadoop master 容器内

其它类似 username / macbook 之类打头的说明你的当前环境是本地机器

```
start hadoop-master container...
mkdir: /Users/Tongtong/src/: File exists
start hadoop-slave1 container...
start hadoop-slave2 container...
Error response from daemon: Container 52d52efa7a493602aa2ea56265366f270e8aaeed0
4af7613112ace92ed4de7e is not running
```

这个问题说master启动所需要的端口被占用了，所以master无法启动，所以

先查看占用端口的程序:

$ lsof -t -itcp:50070

$ lsof -t -itcp:8088

你会得到两个程序的PID，然后关闭程序:

$ kill -9 PID

再重新启动脚本文件

如果遇到下面问题

```
127.0.0.0/8
16/09/12 04:57:20 INFO mapreduce.Job: Task Id : attempt_1473656201503_0001_m_0000
Exception from container-launch.
Container id: container_1473656201503_0001_01_000002
Exit code: 1
Stack trace: ExitCodeException exitCode=1:
        at org.apache.hadoop.util.Shell.runCommand(Shell.java:545)
        at org.apache.hadoop.util.Shell.run(Shell.java:456)
        at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:7
        at org.apache.hadoop.yarn.server.nodemanager.DefaultContainerExecutor.lau
        at org.apache.hadoop.yarn.server.nodemanager.containermanager.launcher.Co
        at org.apache.hadoop.yarn.server.nodemanager.containermanager.launcher.Co
        at java.util.concurrent.FutureTask.run(FutureTask.java:262)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.ja
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.j
        at java.lang.Thread.run(Thread.java:745)


Container exited with a non-zero exit code 1

16/09/12 04:57:36 INFO mapreduce.Job:  map 50% reduce 0%
16/09/12 04:57:36 INFO mapreduce.Job: Task Id : attempt_14736562 0001_m_00000
Error: unable to create new native thread
```

原因是hadoop 默认配置会以 8Gb 内存 4 CPU 来跑, 导致本地机器内存不足

解决方法:

如果你的机器内存 >4g, 可以把docker 内存配大点，最好配>=4G, 这样它执行速度就会很快了

更新容器镜像方法:

$ docker rm $(docker ps -a -q) -f

$ docker rmi -f joway/hadoop-cluster

$ docker pull joway/hadoop-cluster

把运行中的容器都杀死，镜像重新 pull， 然后在那个目录下 ./start-container.sh 重启开起来

问题: docker 配置好之后 MAC 上 RUN wordcount.sh 得到错误 CONNECTION REFUSED

```
mmao-mba:bigdata-class2 mmao$ cd hadoop-cluster-docker
mmao-mba:hadoop-cluster-docker mmao$  sudo ./start-container.sh
start hadoop-master container...
start hadoop-slave1 container...
start hadoop-slave2 container...
root@hadoop-master:~# ./run-wordcount.sh
mkdir: Call From hadoop-master/172.18.0.2 to hadoop-master:9000 failed on connec
tion exception: java.net.ConnectException: Connection refused; For more details
see:  http://wiki.apache.org/hadoop/ConnectionRefused
put: Call From hadoop-master/172.18.0.2 to hadoop-master:9000 failed on connecti
on exception: java.net.ConnectException: Connection refused; For more details se
e:  http://wiki.apache.org/hadoop/ConnectionRefused
16/10/16 20:25:14 INFO client.RMProxy: Connecting to ResourceManager at hadoop-m
aster/172.18.0.2:8032
Exception in thread "main" java.net.ConnectException: Call From hadoop-master/17
2.18.0.2 to hadoop-master:9000 failed on connection exception: java.net.ConnectE
xception: Connection refused; For more details see:  http://wiki.apache.org/hado
op/ConnectionRefused
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstruct
orAccessorImpl.java:57)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingC
onstructorAccessorImpl.java:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:526)
```

解决方法:

你没有运行./start-hadoop

283212213223

Sign In

Block Quote
Pull Quote