FBDP lab 2

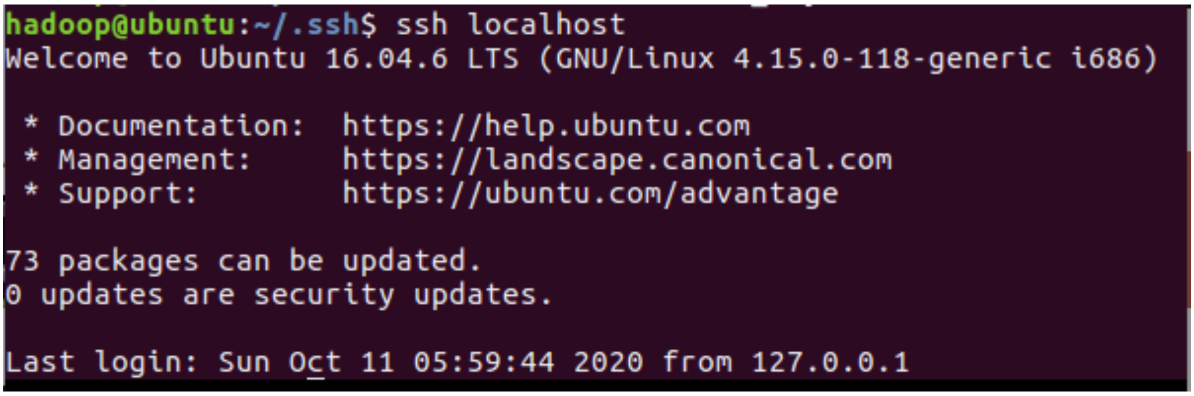
[redacted]

单节点实现 Local (Standalone) Mode

- 创建Hadoop用户

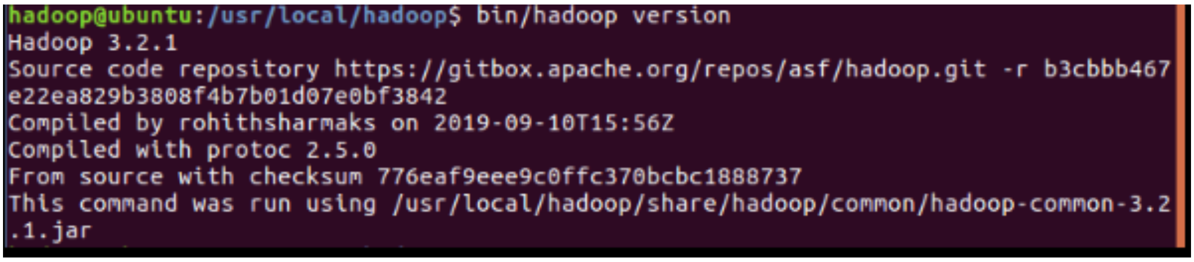
```
sudo useradd -m hadoop -s /bin/bash
sudo passwd hadoop
sudo adduser hadoop sudo ## add root for user
```

- 设置SSH免密登录
  原因： Hadoop中的NameNode将通过SSH登录实现集群中所有机器的启动，但并没有提供SSH输入密码登录的形式，因此为顺利登陆每台机器，需要将所有机器均配置为NameNode可以免密登录。
  - 设置完成后 输入 `ssh localhost` 即可免密登录。



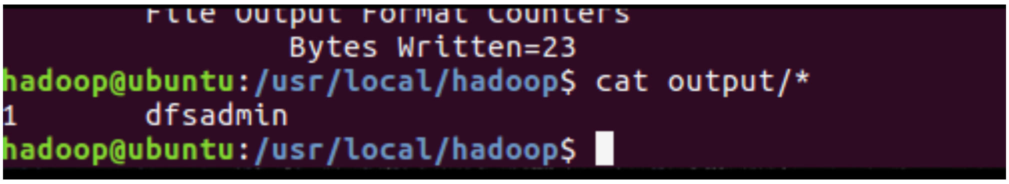
```
hadoop@ubuntu:~/.ssh$ ssh localhost
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-118-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

73 packages can be updated.
0 updates are security updates.

Last login: Sun Oct 11 05:59:44 2020 from 127.0.0.1
```

- 单机Hadoop配置



```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop version
Hadoop 3.2.1
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r b3cbbb467
e22ea829b3808f4b7b01d07e0bf3842
Compiled by rohithsharmaks on 2019-09-10T15:56Z
Compiled with protoc 2.5.0
From source with checksum 776eaf9eee9c0ffc370bcbc1888737
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.2
.1.jar
```

- 单节点实现结果



```
        File Output Format Counters
                Bytes Written=23
hadoop@ubuntu:/usr/local/hadoop$ cat output/*
1       dfsadmin
hadoop@ubuntu:/usr/local/hadoop$ 
```

单机伪分布式 Pesudo-Distributed Mode

- Hadoop将在单节点上已伪分布式的方式运行，进程以分离的Java进程运行，节点既作为NameNode也作为DataNode，同时运行过程中读取的是HDFS中的文件。

> 了解Hadoop下目录的基本定义和结构：
>
> - Hadoop的配置文件位于/usr/local/hadoop/etc/hadoop/中，伪分布式需要修改2个配置文件core-site.xml和hdfs-site.xml；
> - Hadoop的配置文件为xml，每个配置以声明property的name和value的方式来实现
> - sbin：hadoop管理脚本所在目录，主要包含hdfs和yarn中各类服务的启动、关闭脚本
> - bin：hadoop最基本的管理脚本和使用脚本所在目录，是sbin目录下管理脚本的基础实现，用户可以直接使用这些脚本管理和使用hadoop
> - etc：配置文件存放的目录，包括core-site.xml,hdfs-site.xml,mapred-site.xml等从hadoop1.x继承而来的配置文件和yarn-site.xml等hadoop2.x新增的配置文件
>
> | 文件名称 | 格式 | 描述 |
> |---|---|---|
> | hadoop-env.sh | Bash脚本 | 记录配置Hadoop运行所需的环境变量，以运行Hadoop |
> | core-site.xml | Hadoop配置XML | Hadoop core的配置项，例如HDFS和MapReduce 常用的I/O设置等 |
> | hdfs-site.xml | Hadoop配置XML | Hadoop的守护进程的配置项，包括NameNode、SecondaryNameNode和DataNode等 |
> | mapred-site.xml | Hadoop配置XML | MapReduce守护进程的配置项，包括JobTracker 和TaskTracker |
> | masters | 纯文本 | 运行SecondaryNameNode的机器列表(每行一 个) |
> | slaves | 纯文本 | 运行DataNode和TaskTracker的机器列表(每行 一个) |
> | hadoop-metrics.properties | Java属性 | 控制metrics在Hadoop上如何发布的属性 |
>
> - include：对外提供的编程库头文件（具体动态库和静态库在lib目录中，这些头文件均是用c++定义的，通常用于c++程序访问hdfs或者编写mapreduce程序）
> - Lib：该目录包含了hadoop对外提供的才变成动态库和静态库，与include目录中的头文件结合使用
> - libexec：各个服务对应的shell配置文件所在目录，可用于配置日志输出目录、启动参数等信息
> - share：hadoop各个模块编译后的jar包所在目录

配置伪分布需要修改core-site.xml及hdfs-site.xml

core-site.xml配置

```
 <configuration>
   <property>
        <name>hadoop.tmp.dir</name>
        <value>/usr/local/hadoop/tmp</value># 可以不加file:
        <description>Abase for other temporary directories.</description>
   </property>
   <property>
     <name>fs.defaultFS</name>
     <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

- hadoop.tmp.dir表示存放临时数据的目录，即包括NameNode的数据，也包 括DataNode的数据。该路径任意指定，只要实际存在该文件夹即可
- name为fs.defaultFS的值，表示hdfs路径的逻辑名称

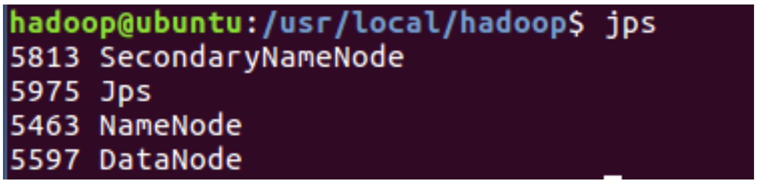
遇到问题：因为无法启动NameNode多次重新尝试后发现格式化NameNode时会失败，上网查报错后发现多次格式化/启动NameNode失败的情况时需要手动清空该文件夹hadoop.tmp.dir对应的文件夹。

hdfs-site.xml配置

```
<configuration>
   <property>
       <name>dfs.replication</name>
       <value>1</value>
   </property>
   <property>
       <name>dfs.namenode.name.dir</name>
       <value>/usr/local/hadoop/hdfs/name</value>
    </property>
   <property>
       <name>dfs.datanode.data.dir</name>
       <value>/usr/local/hadoop/hdfs/data</value>
   </property>
</configuration>
```

- dfs.replication表示副本的数量，在伪分布式中设置为1
- dfs.namenode.name.dir表示本地磁盘目录，是存储fsimage文件的地方
- dfs.datanode.data.dir表示本地磁盘目录，HDFS数据存放block的地方

执行Jps命令的返回结果：



```
hadoop@ubuntu:/usr/local/hadoop$ jps
5813 SecondaryNameNode
5975 Jps
5463 NameNode
5597 DataNode
```

遇到问题：

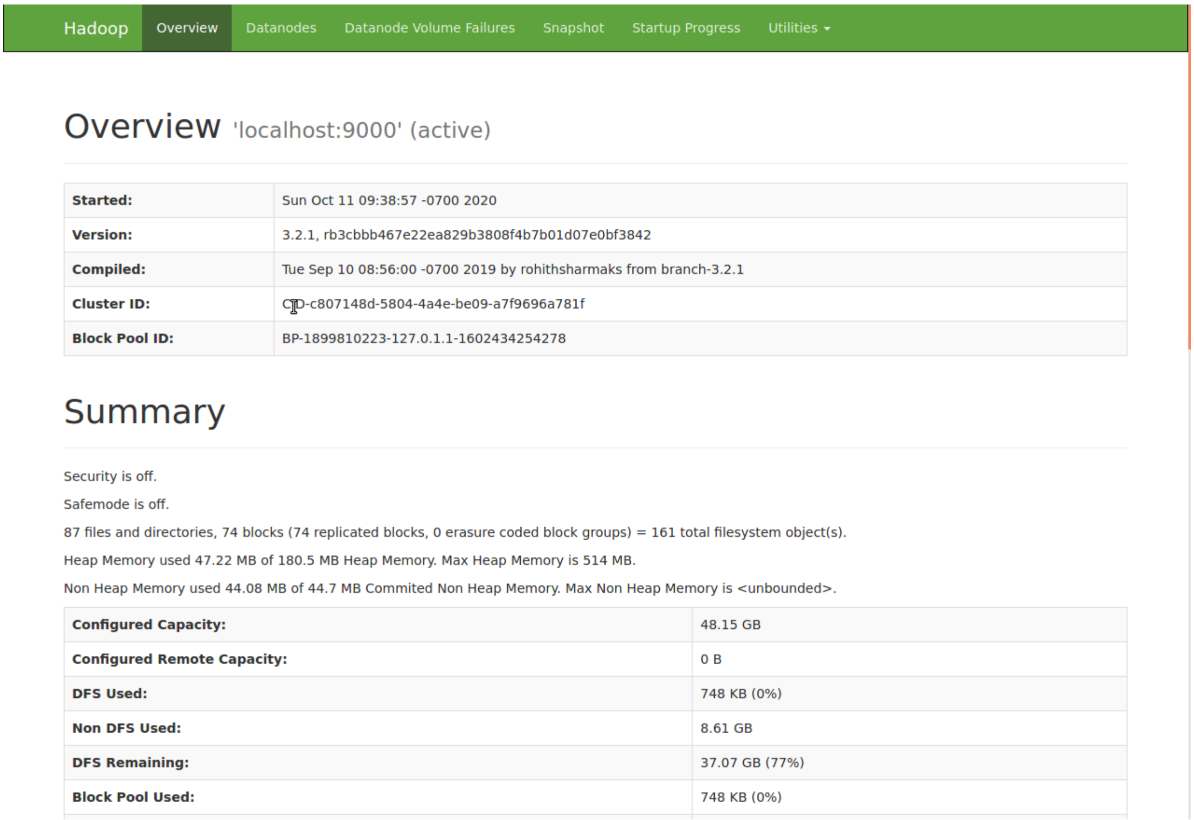
最开始并没有建立hdfs-site.xml中datanode和namenode的两个临时文件夹，运行jps发现NameNode及DataNode无法启动，后来在网上寻求解决方案后加上，但第一个解决方案中的目录建立在/usr/local/hadoop/tmp下，后来出现NameNode无法正常启动；最终经过诸多尝试和求助（甚至在不断格式化和重启节点的过程中出现Java运行内存不足于是killall重新来过了一次）后发现新建文件夹单独存储这两部分文件即可解决

Supplement：

执行Jps后NameNode未启动的解决方案

- stop-all.sh
- 删除core-site.xml下配置的hadoop.tmp.dir所指向的目录并建立新的空目录
    - `rm -fr tmp/*`
    - 可能需要对目录的权限进行修改`sudo chmod -R 777 ./tmp`
- 运行hdfs namenode -format进行格式化
- 运行start-all.sh
- 再次执行jps

最终localhost返回界面：



Overview 'localhost:9000' (active)

Started:	Sun Oct 11 09:38:57 -0700 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 08:56:00 -0700 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-c807148d-5804-4a4e-be09-a7f9696a781f
Block Pool ID:	BP-1899810223-127.0.1.1-1602434254278

Summary

Security is off.

Safemode is off.

87 files and directories, 74 blocks (74 replicated blocks, 0 erasure coded block groups) = 161 total filesystem object(s).

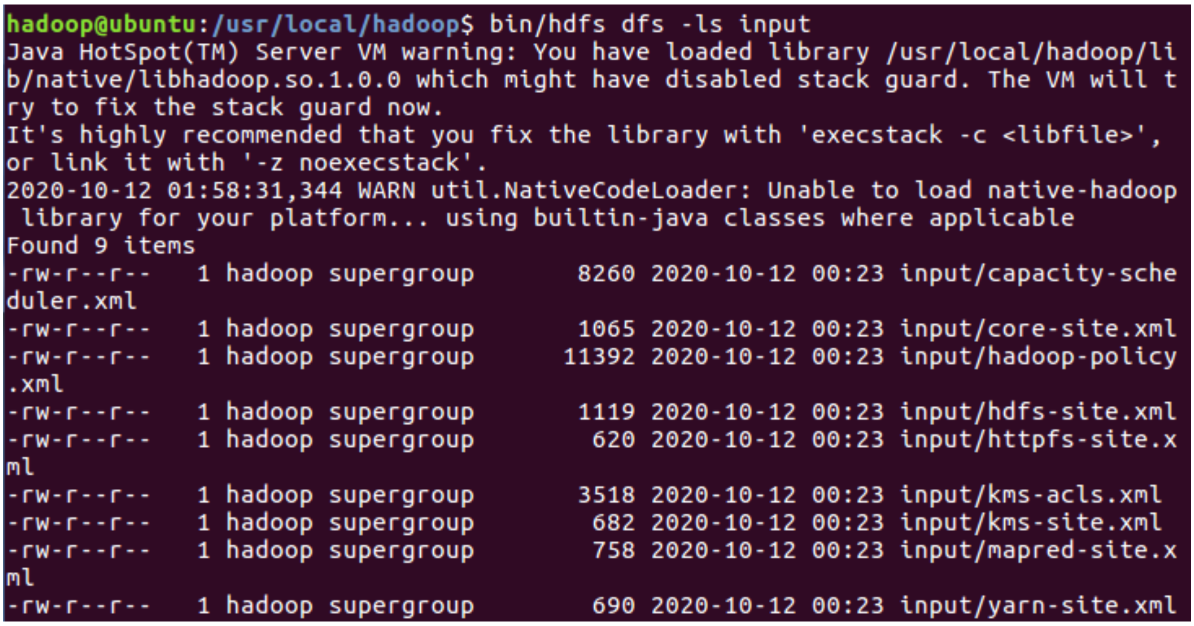
Heap Memory used 47.22 MB of 180.5 MB Heap Memory. Max Heap Memory is 514 MB.

Non Heap Memory used 44.08 MB of 44.7 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	48.15 GB
Configured Remote Capacity:	0 B
DFS Used:	748 KB (0%)
Non DFS Used:	8.61 GB
DFS Remaining:	37.07 GB (77%)
Block Pool Used:	748 KB (0%)

在HDFS文件系统运行示例：

```
$ bin/hdfs dfs -mkdir -p /user/hadoop/input
$ bin/hdfs dfs -put ./etc/hadoop/*.xml /user/hadoop/input/
$ bin/hdfs dfs -ls  /user/hadoop/input/ #查看文件列表
```

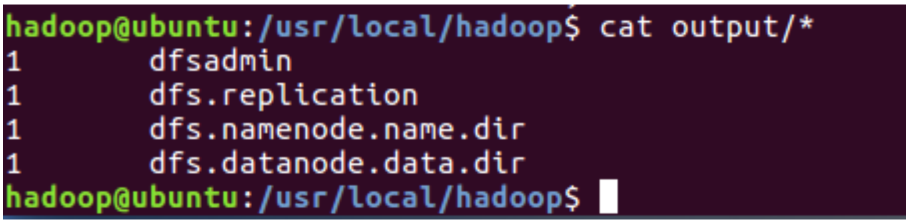


```
hadoop@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -ls input
Java HotSpot(TM) Server VM warning: You have loaded library /usr/local/hadoop/li
b/native/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will t
ry to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>',
or link it with '-z noexecstack'.
2020-10-12 01:58:31,344 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Found 9 items
-rw-r--r--   1 hadoop supergroup       8260 2020-10-12 00:23 input/capacity-sche
duler.xml
-rw-r--r--   1 hadoop supergroup       1065 2020-10-12 00:23 input/core-site.xml
-rw-r--r--   1 hadoop supergroup      11392 2020-10-12 00:23 input/hadoop-policy
.xml
-rw-r--r--   1 hadoop supergroup       1119 2020-10-12 00:23 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2020-10-12 00:23 input/httpfs-site.x
ml
-rw-r--r--   1 hadoop supergroup       3518 2020-10-12 00:23 input/kms-acls.xml
-rw-r--r--   1 hadoop supergroup        682 2020-10-12 00:23 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        758 2020-10-12 00:23 input/mapred-site.x
ml
-rw-r--r--   1 hadoop supergroup        690 2020-10-12 00:23 input/yarn-site.xml
```

将单机中创建的本地input和output文件夹均删除后利用伪分布式运行MapReduce（以此验证伪分布式将读取HDFS中的文件）

```
$ /bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar grep input output 'dfs[a-z.]+'
$ /bin/hdfs dfs -cat output/* #查看运行结果的命令
$ rm -r ./output    # 先删除本地的 output 文件夹（如果存在）
$ /bin/hdfs dfs -get output ./output     # 将 HDFS 上的 output 文件夹拷贝到本机
$ cat ./output/*
#如果需要再次执行则需要执行删除output文件夹的命令
$ /bin/hdfs dfs -rm -r output    # 删除 output 文件夹
$ /sbin/stop-dfs.sh # 关闭Hadoop
```

运行结果如下：



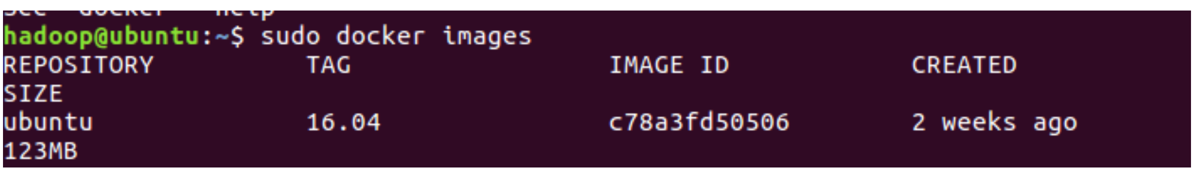
```
hadoop@ubuntu:/usr/local/hadoop$ cat output/*
1       dfsadmin
1       dfs.replication
1       dfs.namenode.name.dir
1       dfs.datanode.data.dir
hadoop@ubuntu:/usr/local/hadoop$ 
```

> 注意：下次重启Hadoop时无需进行NameNode初始化，只需要运行`./sbin/start-dfs.sh`就可以了！

Fully-Distributed Mode

首先构建docker

```
sudo service docker start #启动docker服务
sudo docker images #查看已经下载的镜像
```



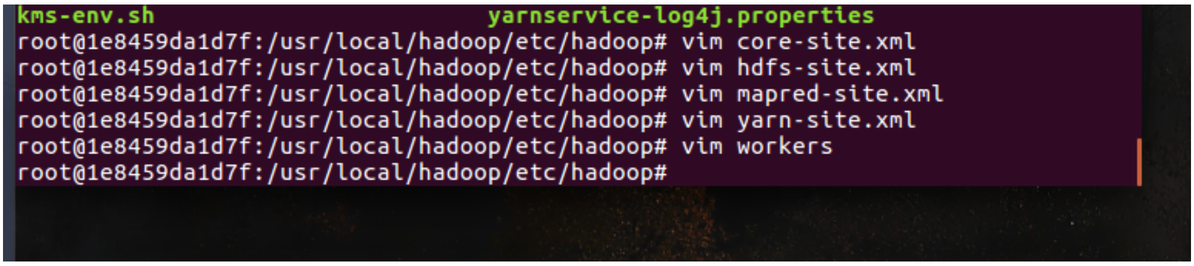
```
hadoop@ubuntu:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
ubuntu              16.04               c78a3fd50506        2 weeks ago
123MB
```

启动容器

```
sudo docker ps -a #查看本机所有容器
#启动容器
sudo docker start fab4da838c2f
st fab4da838c2f /bin/bash
sudo docker ps #查看正在运行的容器
sudo docker stop  fab4da838c2f #关闭某个容器
```

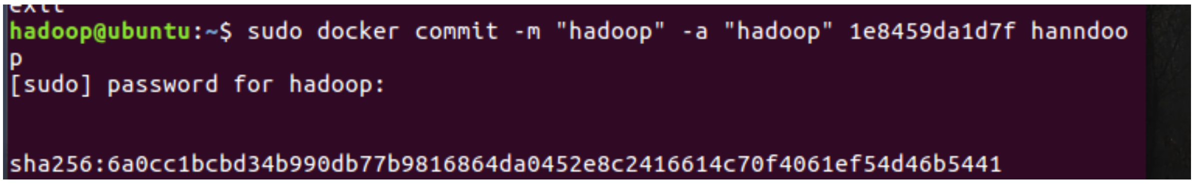
在容器中安装openjdk8，hadoop

配置hadoop



```
kms-env.sh                                yarnservice-log4j.properties
root@1e8459da1d7f:/usr/local/hadoop/etc/hadoop# vim core-site.xml
root@1e8459da1d7f:/usr/local/hadoop/etc/hadoop# vim hdfs-site.xml
root@1e8459da1d7f:/usr/local/hadoop/etc/hadoop# vim mapred-site.xml
root@1e8459da1d7f:/usr/local/hadoop/etc/hadoop# vim yarn-site.xml
root@1e8459da1d7f:/usr/local/hadoop/etc/hadoop# vim workers
root@1e8459da1d7f:/usr/local/hadoop/etc/hadoop#
```

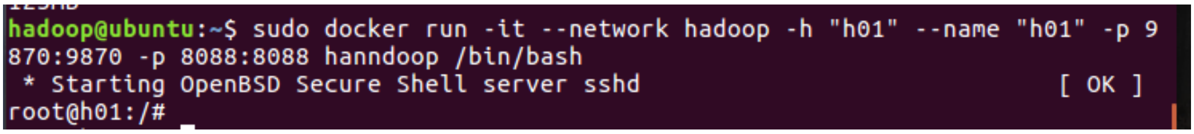
将容器导出为镜像



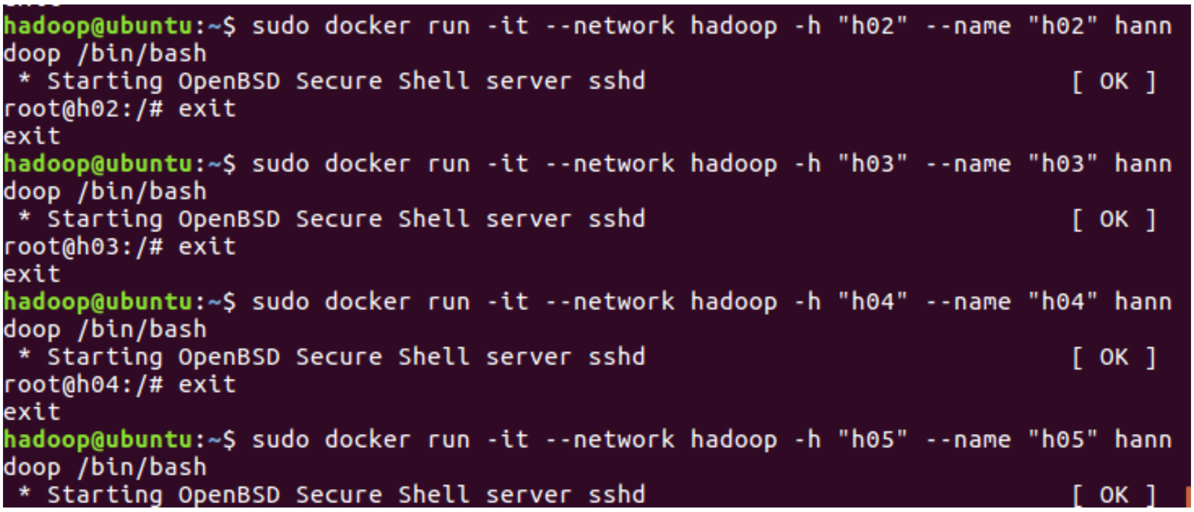
```
hadoop@ubuntu:~$ sudo docker commit -m "hadoop" -a "hadoop" 1e8459da1d7f hanndoo
p
[sudo] password for hadoop:

sha256:6a0cc1bcbd34b990db77b9816864da0452e8c2416614c70f4061ef54d46b5441
```

启动5个终端：

```
sudo docker run -it --network hadoop -h "h01" --name "h01" -p 9870:9870 -p
8088:8088 hanndoop /bin/bash
```

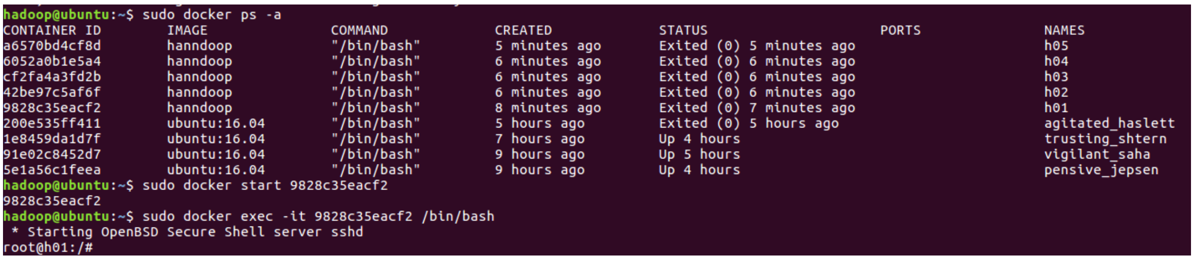


```
hadoop@ubuntu:~$ sudo docker run -it --network hadoop -h "h01" --name "h01" -p 9
870:9870 -p 8088:8088 hanndoop /bin/bash
 * Starting OpenBSD Secure Shell server sshd                            [ OK ]
root@h01:/#
```



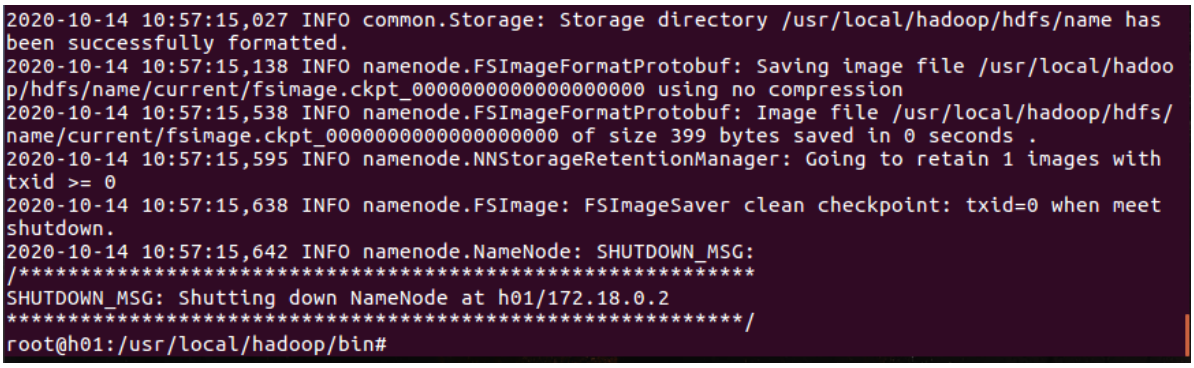
```
hadoop@ubuntu:~$ sudo docker run -it --network hadoop -h "h02" --name "h02" hann
doop /bin/bash
 * Starting OpenBSD Secure Shell server sshd                            [ OK ]
root@h02:/# exit
exit
hadoop@ubuntu:~$ sudo docker run -it --network hadoop -h "h03" --name "h03" hann
doop /bin/bash
 * Starting OpenBSD Secure Shell server sshd                            [ OK ]
root@h03:/# exit
exit
hadoop@ubuntu:~$ sudo docker run -it --network hadoop -h "h04" --name "h04" hann
doop /bin/bash
 * Starting OpenBSD Secure Shell server sshd                            [ OK ]
root@h04:/# exit
exit
hadoop@ubuntu:~$ sudo docker run -it --network hadoop -h "h05" --name "h05" hann
doop /bin/bash
 * Starting OpenBSD Secure Shell server sshd                            [ OK ]
```

查看刚刚建立的终端并进入h01：

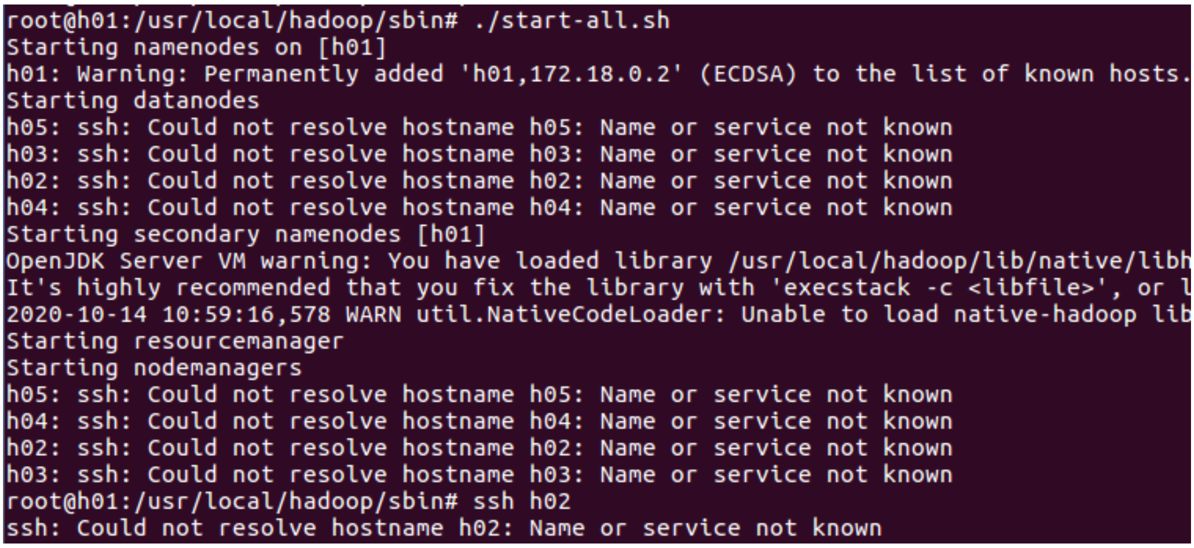


```
hadoop@ubuntu:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                     PORTS               NAMES
a6570bd4cf8d        hanndoop            "/bin/bash"         5 minutes ago       Exited (0) 5 minutes ago                       h05
6052a0b1e5a4        hanndoop            "/bin/bash"         6 minutes ago       Exited (0) 6 minutes ago                       h04
cf2fa4a3fd2b        hanndoop            "/bin/bash"         6 minutes ago       Exited (0) 6 minutes ago                       h03
42be97c5af6f        hanndoop            "/bin/bash"         6 minutes ago       Exited (0) 6 minutes ago                       h02
9828c35eacf2        hanndoop            "/bin/bash"         8 minutes ago       Exited (0) 7 minutes ago                       h01
200e535ff411        ubuntu:16.04        "/bin/bash"         5 hours ago         Exited (0) 5 hours ago                         agitated_haslett
1e8459da1d7f        ubuntu:16.04        "/bin/bash"         7 hours ago         Up 4 hours                                     trusting_shtern
91e02c8452d7        ubuntu:16.04        "/bin/bash"         9 hours ago         Up 5 hours                                     vigilant_saha
5e1a56c1feea        ubuntu:16.04        "/bin/bash"         9 hours ago         Up 4 hours                                     pensive_jepsen
hadoop@ubuntu:~$ sudo docker start 9828c35eacf2
9828c35eacf2
hadoop@ubuntu:~$ sudo docker exec -it 9828c35eacf2 /bin/bash
 * Starting OpenBSD Secure Shell server sshd
root@h01:/#
```

在h01中对NameNode进行格式化：

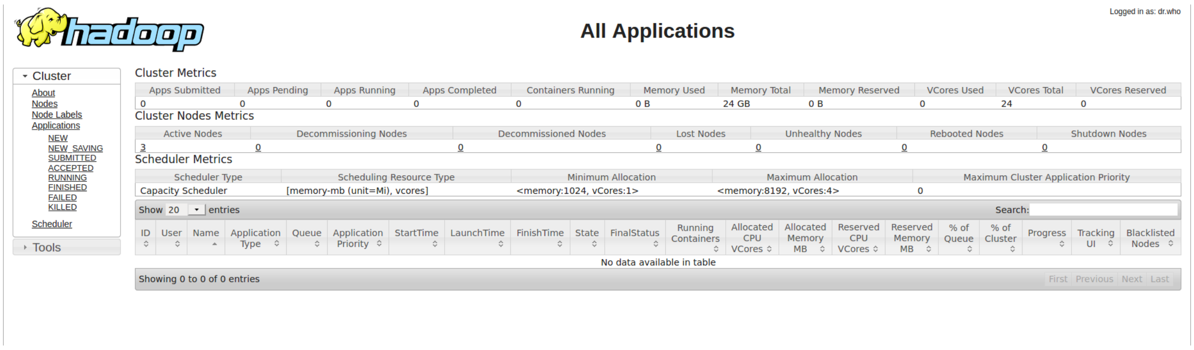


```
2020-10-14 10:57:15,027 INFO common.Storage: Storage directory /usr/local/hadoop/hdfs/name has
been successfully formatted.
2020-10-14 10:57:15,138 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoo
p/hdfs/name/current/fsimage.ckpt_0000000000000000000 using no compression
2020-10-14 10:57:15,538 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/hdfs/
name/current/fsimage.ckpt_0000000000000000000 of size 399 bytes saved in 0 seconds .
2020-10-14 10:57:15,595 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with
txid >= 0
2020-10-14 10:57:15,638 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet
shutdown.
2020-10-14 10:57:15,642 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at h01/172.18.0.2
************************************************************/
root@h01:/usr/local/hadoop/bin#
```



```
root@h01:/usr/local/hadoop/sbin# ./start-all.sh
Starting namenodes on [h01]
h01: Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts.
Starting datanodes
h05: ssh: Could not resolve hostname h05: Name or service not known
h03: ssh: Could not resolve hostname h03: Name or service not known
h02: ssh: Could not resolve hostname h02: Name or service not known
h04: ssh: Could not resolve hostname h04: Name or service not known
Starting secondary namenodes [h01]
OpenJDK Server VM warning: You have loaded library /usr/local/hadoop/lib/native/libh
It's highly recommended that you fix the library with 'execstack -c <libfile>', or l
2020-10-14 10:59:16,578 WARN util.NativeCodeLoader: Unable to load native-hadoop lib
Starting resourcemanager
Starting nodemanagers
h05: ssh: Could not resolve hostname h05: Name or service not known
h04: ssh: Could not resolve hostname h04: Name or service not known
h02: ssh: Could not resolve hostname h02: Name or service not known
h03: ssh: Could not resolve hostname h03: Name or service not known
root@h01:/usr/local/hadoop/sbin# ssh h02
ssh: Could not resolve hostname h02: Name or service not known
```

localhost:8088



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	24 GB	0 B	0	24	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
3	0	0	0	0	0	0

Scheduler Metrics

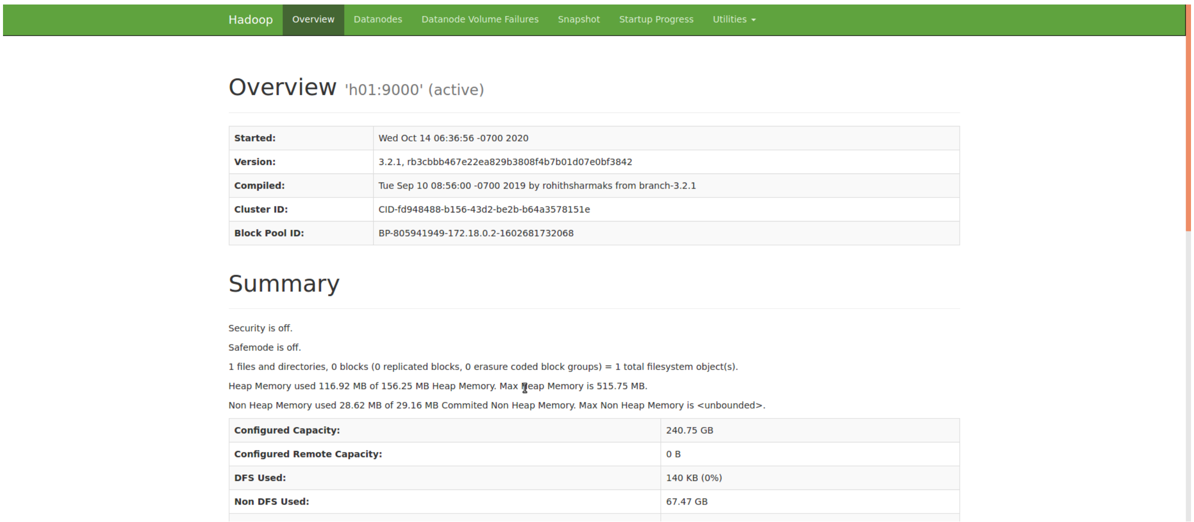
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																				

Showing 0 to 0 of 0 entries First Previous Next Last

localhost:9870



Overview 'h01:9000' (active)

Started:	Wed Oct 14 06:36:56 -0700 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 08:56:00 -0700 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-fd948488-b156-43d2-be2b-b64a3578151e
Block Pool ID:	BP-805941949-172.18.0.2-1602681732068

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 116.92 MB of 156.25 MB Heap Memory. Max Heap Memory is 515.75 MB.

Non Heap Memory used 28.62 MB of 29.16 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	240.75 GB
Configured Remote Capacity:	0 B
DFS Used:	140 KB (0%)
Non DFS Used:	67.47 GB

遇到问题：

1. ssh免密登录设置后仍然不能连接

解决方案：在h01 etc/profile中增加各节点对应的ip地址后重试即可（for reference)

> h04 172.18.0.3 h05 172.18.0.4 h03 172.18.0.5
>
> h02 172.18.0.6 --可以在各容器内通过ifconfig查看ip地址

2. NameNode /DataNode 无法启动
   若出现配置文件问题需要重新格式化，则需要参照伪分布式操作内容将NameNode对应的hdfs临时文件和tmp内临时文件全部手动清除，再将各datanode内tmp/dfs/data文件夹手动清空后重新进行格式化，偷懒的话会出现NameNode或DataNode无法成功启动的情况（别问我是怎么知道的）
3. localhost:9870 打不开

需要在hdfs-site.xml中添加配置如下

```
<property>
  <name>dfs.http-address</name>
  <value>localhost:9870</value>
</property>
```

之后stop-all.sh 再重新启动即可解决！

docker实现多节点示例

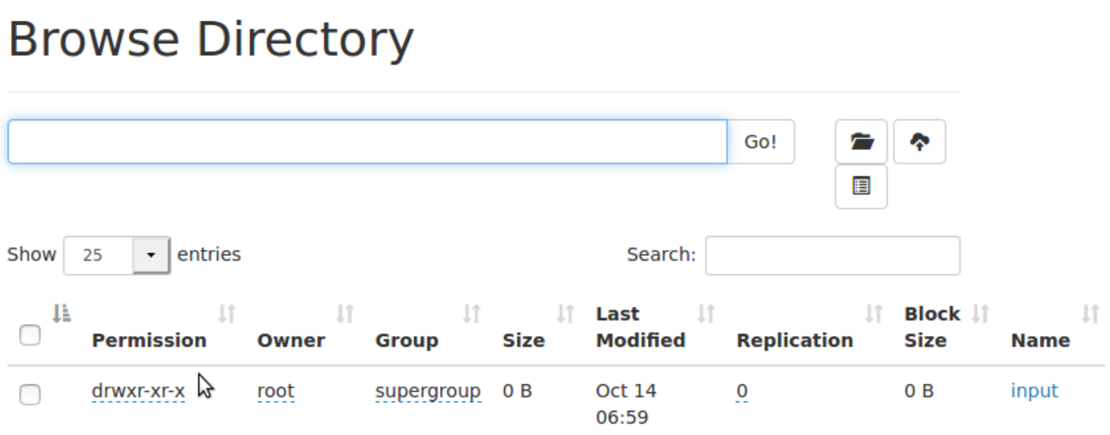
把license作为需要统计的文件

```
root@h01:/usr/local/hadoop# cat LICENSE.txt > file1.txt
root@h01:/usr/local/hadoop# ls
```

在 HDFS 中创建 input 文件夹

```
root@h01:/usr/local/hadoop/bin# ./hadoop fs -mkdir /input
root@h01:/usr/local/hadoop/bin#
```

在网页端可以看到生成的新文件夹input



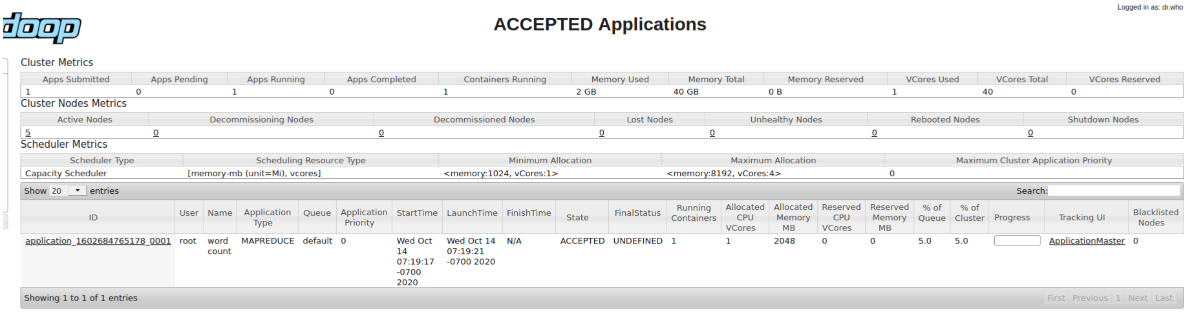
Browse Directory

Show 25 entries

Search:

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	drwxr-xr-x	root	supergroup	0 B	Oct 14 06:59	0	0 B	input

最终运行示例：



Logged in as: dr.who

ACCEPTED Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
1	0	1	0	1	2 GB	40 GB	0 B	1	40	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
5	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1602684765178_0001	root	word count	MAPREDUCE	default	0	Wed Oct 14 07:19:17 -0700 2020	Wed Oct 14 07:19:21 -0700 2020	N/A	ACCEPTED	UNDEFINED	1	1	2048	0	0	5.0	5.0		ApplicationMaster	0

Showing 1 to 1 of 1 entries

（但是似乎跑不动orz）

Reference

（鞠躬鸣谢）

> - [Hadoop学习笔记(1)-Hadoop在Ubuntu的安装和使用]
> - [大数据-运行Hadoop伪分布式实例]
> - [基于 Docker 构建 Hadoop 平台]