部署使用设备：

三台 VMware-linux 虚拟机
hadoop-master：Ubuntu18.04
hadoop-slave1：CentOS 7
hadoop-slave2: Ubuntu14
使用分布式方式部署

一、Hadoop 集群安装部署

最开始部署 Hadoop 集群的时候使用伪分布方式，在此种方式下进行调试与报错解决比较方便。**在伪分布式方式运行成功后再改成分布式部署**，更改文件后使用 scp 传送命令传至其他虚拟机。 在此基础上再进行调试。由于计算机资源有限，所以使用了两台原先已经完成安装的虚拟机作为部署节点。

1、更改系统文件 hosts，配置主机路由与主机名，Ping 可联通。

```
prestyan@prestyan-virtual-machine:/etc$ cat hosts
127.0.0.1       localhost
127.0.1.1       prestyan-virtual-machine
192.168.52.129 hadoop

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

2、进入 hadoop 文件配置目录/etc/Hadoop。

```
prestyan@prestyan-virtual-machine:/etc$ cd /opt/hadoop
prestyan@prestyan-virtual-machine:/opt/hadoop$ ls
bin  include  libexec      logs        README.txt  share
etc  lib      LICENSE.txt  NOTICE.txt  sbin
prestyan@prestyan-virtual-machine:/opt/hadoop$ cd etc
prestyan@prestyan-virtual-machine:/opt/hadoop/etc$ ls
hadoop
prestyan@prestyan-virtual-machine:/opt/hadoop/etc$ cd hadoop
prestyan@prestyan-virtual-machine:/opt/hadoop/etc/hadoop$ ls
capacity-scheduler.xml      httpfs-env.sh            mapred-env.sh
configuration.xsl           httpfs-log4j.properties  mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret  mapred-site.xml
core-site.xml               httpfs-site.xml          mapred-site.xml.template
hadoop-env.cmd              kms-acls.xml             slaves
hadoop-env.sh               kms-env.sh               ssl-client.xml.example
hadoop-metrics2.properties  kms-log4j.properties     ssl-server.xml.example
hadoop-metrics.properties   kms-site.xml             yarn-env.cmd
hadoop-policy.xml           log4j.properties         yarn-env.sh
hdfs-site.xml               mapred-env.cmd           yarn-site.xml
prestyan@prestyan-virtual-machine:/opt/hadoop/etc/hadoop$
```

3、更改 hadoop-env.sh 中 java 环境变量，改为新安装的 jdk1.8。

```
# The java implementation to use.
export JAVA_HOME=/home/prestyan/jdk/jdk1.8.0_11
```

4、更改 core-site.xml 代码，编辑临时目录。

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://127.0.0.1:9000</value>
</property>
<property>
<!--这个配置是将hadoop的临时目录改成自定义的目录下-->
<name>hadoop.tmp.dir</name>
<value>/data1/hadoop/tmp</value>
</property>
</configuration>
```

5、更改 hdfs-site.xml 代码，编辑 datanode 存储位置与副本数量。

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.datanode.data.dir</name>
<value>/data1/hadoop/data</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/data1/hadoop/name</value>
</property>
<!-- 指定HDFS副本的数量 -->
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

6、更改 mapred-site.xml 文件。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/etc/hadoop$ cat mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>mapreduce.framework.name
</name>
<value>yarn</value>
</property>
</configuration>
```

7、配置 yarn-site.xml 文件，设置好 yarn 地址。

```
<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>127.0.0.1:8032</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>127.0.0.1:8030</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>127.0.0.1:8031</value>
</property>

</configuration>
```

8、使用 hadoop 命令对 namenode 存储区域初始化。

```
prestyan@prestyan-virtual-machine:/opt/hadoop$ cd bin
prestyan@prestyan-virtual-machine:/opt/hadoop/bin$ ls
container-executor  hadoop  hadoop.cmd  hdfs  hdfs.cmd  mapred  mapred.cmd  rcc  test-container-executor  yarn  yarn.cmd
prestyan@prestyan-virtual-machine:/opt/hadoop/bin$ ./hadoop namenode -format
```

9、初始化成功，返回 0。

```
22/03/13 10:40:28 INFO common.Storage: Storage directory /data1/hadoop/name has been successfully formatted.
22/03/13 10:40:28 INFO namenode.FSImageFormatProtobuf: Saving image file /data1/hadoop/name/current/fsimage.c
22/03/13 10:40:28 INFO namenode.FSImageFormatProtobuf: Image file /data1/hadoop/name/current/fsimage.ckpt_000
22/03/13 10:40:28 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
22/03/13 10:40:28 INFO util.ExitUtil: Exiting with status 0
22/03/13 10:40:28 INFO namenode.NameNode: SHUTDOWN_MSG:
```

10、更改初始化所创建的 data1 文件所属，否则 hadoop 无法使用该文件夹。

```
prestyan@prestyan-virtual-machine:/$ sudo chown -R prestyan:prestyan data1
```

11、可以看到 data1 的所属用户与所属组已经改到了 hadoop 安装用户下。

```
drwxr-xr-x   3 prestyan prestyan      4096 3月    7 15:58 data1/
```

12、进入 sbin 目录，启动 dfs 程序。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
prestyan@localhost's password:
localhost: starting namenode, logging to /opt/hadoop/logs/hadoop-prestyan-nameno
de-prestyan-virtual-machine.out
prestyan@localhost's password:
localhost: starting datanode, logging to /opt/hadoop/logs/hadoop-prestyan-datano
de-prestyan-virtual-machine.out
Starting secondary namenodes [0.0.0.0]
prestyan@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-prestyan
-secondarynamenode-prestyan-virtual-machine.out
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$
```

13、使用 jps 可以看到 dfs 启动成功。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ jps
21360 DataNode
21619 Jps
20939 SecondaryNameNode
20557 NameNode
```

14、启动 yarn 程序，使用 jps 查看可以发现 yarn 也成功启动。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ ./start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop/logs/yarn-prestyan-resourcemana
ger-prestyan-virtual-machine.out
prestyan@localhost's password:
localhost: starting nodemanager, logging to /opt/hadoop/logs/yarn-prestyan-nodem
anager-prestyan-virtual-machine.out
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ jps
21360 DataNode
21681 ResourceManager
22007 NodeManager
20939 SecondaryNameNode
22108 Jps
20557 NameNode
```

15、使用本地浏览器登录网页 hadoop-master:50070 查看 hadoop 启动状态。可以看到基本信息，存货节点数为 1（伪分布）。

Configured Capacity:	19.56 GB
DFS Used:	24 KB
Non DFS Used:	12.38 GB
DFS Remaining:	7.18 GB
DFS Used%:	0%
DFS Remaining%:	36.69%
Block Pool Used:	24 KB
Block Pool Used%:	0%
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	2022/3/13 11:59:32

16、进入 datanode information 界面，可以看到 datanode 状态，说明 hadoop 启动成功。

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
prestyan-virtual-machine (127.0.0.1:50010)	2	In Service	19.56 GB	24 KB	12.38 GB	7.17 GB	0	24 KB (0%)	0	2.6.5

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction

Hadoop, 2016.

Legacy UI

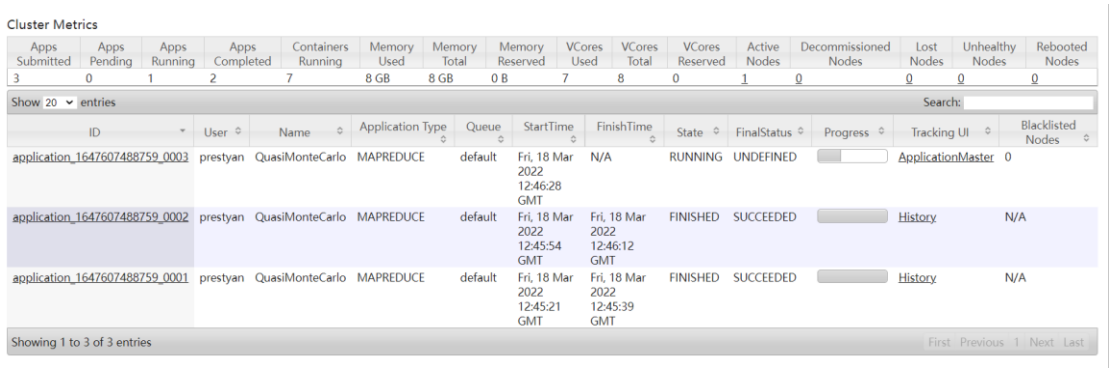
17、使用 hdfs 命令上传文件至/input 文件夹后，尝试使用 hadoop jar 命令运行 wordcount 程序。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/bin$ hadoop jar /opt/hadoop/share/hadoop/mapreduce/hado
op-mapreduce-examples-2.6.5.jar wordcount /input/codes.cpp /output
22/03/18 20:19:10 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
22/03/18 20:19:11 INFO input.FileInputFormat: Total input paths to process : 1
22/03/18 20:19:11 INFO mapreduce.JobSubmitter: number of splits:1
22/03/18 20:19:11 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1647604832830_0001
22/03/18 20:19:11 INFO impl.YarnClientImpl: Submitted application application_1647604832830_0001
22/03/18 20:19:11 INFO mapreduce.Job: The url to track the job: http://prestyan-virtual-machine:8088/
proxy/application_1647604832830_0001/
22/03/18 20:19:11 INFO mapreduce.Job: Running job: job_1647604832830_0001
22/03/18 20:19:17 INFO mapreduce.Job: Job job_1647604832830_0001 running in uber mode : false
22/03/18 20:19:17 INFO mapreduce.Job:  map 0% reduce 0%
22/03/18 20:19:21 INFO mapreduce.Job:  map 100% reduce 0%
22/03/18 20:19:26 INFO mapreduce.Job:  map 100% reduce 100%
22/03/18 20:19:27 INFO mapreduce.Job: Job job_1647604832830_0001 completed successfully
22/03/18 20:19:27 INFO mapreduce.Job: Counters: 49
```

18、可以看到程序运行成功。

```
  File System Counters
        FILE: Number of bytes read=1410
        FILE: Number of bytes written=217235
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=1368
        HDFS: Number of bytes written=1041
        HDFS: Number of read operations=6
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
  Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=1918
        Total time spent by all reduces in occupied slots (ms)=2327
        Total time spent by all map tasks (ms)=1918
        Total time spent by all reduce tasks (ms)=2327
        Total vcore-milliseconds taken by all map tasks=1918
        Total vcore-milliseconds taken by all reduce tasks=2327
        Total megabyte-milliseconds taken by all map tasks=1964032
        Total megabyte-milliseconds taken by all reduce tasks=2382848
  Map-Reduce Framework
        Map input records=50
        Map output records=143
        Map output bytes=1657
        Map output materialized bytes=1410
        Input split bytes=102
```

19、尝试使用 hadoop jar 命令运行 pi 程序，并且进行 50 次 mapreduce 运算，每次迭代五十次。在 hadoop-master:8088 上查看运行状态。



下面更改为分布式：

20、更改 core-site.xml hdfs-site.xml mapred-site.xml yarn-site.xml 文件，使用分布式配置。主机 ip 不能再使用 127.0.0.1，而是要使用本机实际 ip 地址。Core 文件添加 io.file.buffer.size 选项，同时将 yarn 文件中 resourcemanager 与 nodemanager 地址补全。

```
-rw-rw-r-- 1 prestyan prestyan  4436 10月  3  2016 capacity-scheduler.xml
-rw-rw-r-- 1 prestyan prestyan  1128 3月  21 11:05 core-site.xml
-rw-rw-r-- 1 prestyan prestyan  9683 10月  3  2016 hadoop-policy.xml
-rw-rw-r-- 1 prestyan prestyan  1284 3月  21 11:06 hdfs-site.xml
-rw-rw-r-- 1 prestyan prestyan   620 10月  3  2016 httpfs-site.xml
-rw-rw-r-- 1 prestyan prestyan  3523 10月  3  2016 kms-acls.xml
-rw-rw-r-- 1 prestyan prestyan  5511 10月  3  2016 kms-site.xml
-rw-rw-r-- 1 prestyan prestyan  4113 10月  3  2016 mapred-queues.xml.template
-rw-rw-r-- 1 prestyan prestyan  1281 3月  21 11:08 mapred-site.xml
-rw-rw-r-- 1 prestyan prestyan   842 3月   7 12:00 mapred-site.xml.template
-rw-rw-r-- 1 prestyan prestyan  2316 10月  3  2016 ssl-client.xml.example
-rw-rw-r-- 1 prestyan prestyan  2268 10月  3  2016 ssl-server.xml.example
-rw-rw-r-- 1 prestyan prestyan  1939 3月  21 16:05 yarn-site.xml
```

21、更改 ip 如下图所示。

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://192.168.52.131:9000</value>
</property>
<property>
```

22、更改 slaves 文件，删去默认 localhost，改成两台 slave 的 ip 地址。

```
192.168.52.129
192.168.52.128
```

23、将配置好的 hadoop 文件使用 scp 传送至其他主机。并在其他主机上修改好配置。

```
prestyan@hadoop-master:/opt/hadoop/etc/hadoop$ sudo scp -r ../hadoop prestyan@19
2.168.52.129:/home/prestyan
```

24、使用 hadoop 运行圆周率命令的时候报错：找不到主机路由。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.5.jar pi 5 5
Number of Maps  = 5
Samples per Map = 5
22/03/21 15:30:17 INFO hdfs.DFSClient: Exception in createBlockOutputStream
java.net.NoRouteToHostException: 没有到主机的路由
```

25、检查 host 路由发现 host 路由配置全部正确，且能够 ping 通。

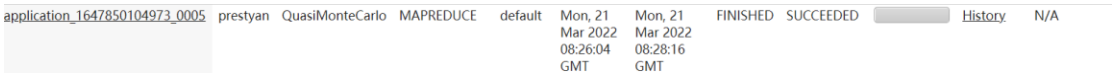
```
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ cat /etc/hosts
127.0.0.1       localhost
#127.0.1.1      prestyan-virtual-machine
192.168.52.131 hadoop-master
192.168.52.129 hadoop-slave1
192.168.52.128 hadoop-slave2

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

26、在检查防火墙的时候发现，CentOS 防火墙是默认开启的，没有关闭，所以可能 hadoop 调度访问被拒绝了，于是使用命令关闭 CentOS 防火墙。

```
[prestyan@hadoop-slave2 bin]$ firewall-cmd --state
running
[prestyan@hadoop-slave2 bin]$ systemctl start firewall.service
Failed to start firewall.service: Unit not found.
[prestyan@hadoop-slave2 bin]$ systemctl start firewalld.service
[prestyan@hadoop-slave2 bin]$ firewall-cmd --state
running
[prestyan@hadoop-slave2 bin]$ systemctl stop firewalld.service
[prestyan@hadoop-slave2 bin]$ firewall-cmd --state
not running
[prestyan@hadoop-slave2 bin]$
```

27、关闭后可以正常跑 hadoop jar，问题解决。



28、在尝试一些比较复杂的运算时，如…… pi 100 100，jar 运行会卡在运行过程当中，有的甚至卡在 running job 处，即分配了节点运行的工作，节点却没有办法正常运行，卡住不动。

```
22/03/21 16:26:04 INFO mapreduce.Job: Running job: job_1647850104973_0005
```

```
Wrote input for Map #97
Wrote input for Map #98
Wrote input for Map #99
Starting Job
22/03/21 16:26:03 INFO client.RMProxy: Connecting to ResourceManager at /192.168.52.131:8032
22/03/21 16:26:04 INFO input.FileInputFormat: Total input paths to process : 100
22/03/21 16:26:04 INFO mapreduce.JobSubmitter: number of splits:100
22/03/21 16:26:04 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1647850104973_0005
22/03/21 16:26:04 INFO impl.YarnClientImpl: Submitted application application_1647850104973_0005
22/03/21 16:26:04 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/applica
22/03/21 16:26:04 INFO mapreduce.Job: Running job: job_1647850104973_0005
22/03/21 16:26:10 INFO mapreduce.Job: Job job_1647850104973_0005 running in uber mode : false
22/03/21 16:26:10 INFO mapreduce.Job:  map 0% reduce 0%
```

29、查阅网上的资料后猜测，问题出在 yarn 分配运行资源的时候。出现此故障的原因应该是，在每个 Docker 分配的内存和 CPU 资源太少，不能满足 Hadoop 和 Hive 运行所需的默认资源需求。于是调整 yarn-site 配置文件，重新分配资源大小。

```
<property>
     <name>yarn.scheduler.minimum-allocation-mb</name>
     <value>2048</value>
 </property>
<property>
        <name>yarn.resourcemanager.webapp.address</name>
        <value>192.168.52.131:8088</value>
</property>
<property>
        <name>yarn.nodemanager.resource.memory-mb</name>
        <value>20480</value>
</property>
 <property>
     <name>yarn.scheduler.maximum-allocation-mb</name>
     <value>4096</value>
 </property>

<property>
     <name>yarn.nodemanager.veme-pmem-ratio</name>
     <value>2.1</value>
 </property>
```

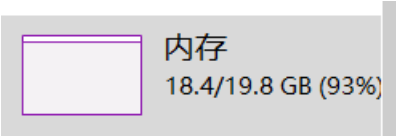
30、重新尝试运行复杂程序。

```
prestyan@prestyan-virtual-machine:/opt/hadoop/sbin$ hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.5.jar pi 100 100
Number of Maps  = 100
```

31、可以看到运行成功。该问题解决。

```
Job Finished in 135.044 seconds
Estimated value of Pi is 3.14080000000000000000
```

32、由于计算机资源限制，每个虚拟机分配到的内存量不大，所以无法再给 yarn 分配更多的内存资源，在运行程序过程中如果运算量太大该进程甚至会被杀死，出现 exit 144 的报错，这是没有办法避免的。



至此，hadoop 分布式配置全部成功完成。

二、Zookeeper 集群安装部署

HBase 需要利用 Zookeeper 作为协同服务，用其来维护配置信息，命名注册，利用其提供分布式同步和组服务，来进行有序管理。所以安装 HBase 之前需要先安装 zookeeper。Zookeeper 使用分布式部署。

1、下载 zookeeper-bin 并且解压，然后进入 conf 目录，找到 zoo.cfg 文件进行配置。

```
[prestyan@localhost apache-zookeeper-3.6.3-bin]$ cd conf/
[prestyan@localhost conf]$ ll
总用量 16
-rw-r--r--. 1 prestyan prestyan  535 4月   9 2021 configuration.xsl
-rw-r--r--. 1 prestyan prestyan 3435 4月   9 2021 log4j.properties
-rw-rw-r--. 1 prestyan prestyan  241 3月  20 13:10 zoo.cfg
-rw-r--r--. 1 prestyan prestyan 1148 4月   9 2021 zoo_sample.cfg
[prestyan@localhost conf]$
```

2、由于需要分布式部署，所以要规定 tickTime、initLimit、syncLimit 的数值，否则无法运行，会出错。然后配置 data 与 log 目录，此目录在 home/prestyan 下，zookeeper 有权直接创建改动，不必再赋予权限。同时配置三台主机的 ip 地址，设置好 id 号。

```
[prestyan@localhost conf]$ cat zoo.cfg
tickTime=2000
dataDir=/home/prestyan/zookeeper/data
dataLogDir=/home/prestyan/zookeeper/dataLog
clientPort=2181
initLimit=10
syncLimit=5

server.1=192.168.52.131:2888:3888
server.2=192.168.52.130:2888:3888
server.3=192.168.52.128:2888:3888

[prestyan@localhost conf]$
```

3、启动 zookeeper，此时不会成功启动，只是让其创建对应文件夹 datadir 目录。然后进入目录 datadir 配置 id，本机为 3。

```
[prestyan@localhost ~]$ cd zookeeper/
[prestyan@localhost zookeeper]$ cd data
[prestyan@localhost data]$ ll
总用量 8
-rw-rw-r--. 1 prestyan prestyan  2 3月   20 20:55 myid
drwxrwxr-x. 2 prestyan prestyan 65 3月   20 20:56 version-2
-rw-rw-r--. 1 prestyan prestyan  4 3月   20 20:56 zookeeper_server.pid
[prestyan@localhost data]$ cat myid
3
[prestyan@localhost data]$
```

4、再次启动 zookeeper，可以看到启动成功。其他主机配置步骤相似，只是要主义改变 myid 数值。

```
[prestyan@localhost bin]$ ./zkServer.sh start
/usr/bin/java
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[prestyan@localhost bin]$
```

5、查看 zkserver 的 status，发现 status 显示报错，可能没有运行。

```
[prestyan@localhost bin]$ ./zkServer.sh status
/usr/bin/java
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Error contacting service. It is probably not running.
[prestyan@localhost bin]$
```

6、查看 log 文件，发现报错问题在于连接不到其中一台主机，拒绝连接。

```
2022-03-18 21:19:55,145 [myid:1] - WARN  [QuorumConnectionThread-[myid=1]-2:QuorumCnxManager@400]
- Cannot open channel to 2 at election address /192.168.52.130:3888
java.net.ConnectException: 拒绝连接 (Connection refused)
```

7、猜测有可能是其他主机没有启动的原因，于是将其他两台主机 zkServer.sh 进程全部启动，可以看到启动成功。

```
ZooKeeper JMX enabled by default
Using config: /opt/zk/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

```
prestyan@prestyan-virtual-machine:/opt/zookeeper/apache-zookeeper-3.6.3-bin/bin$
 ./zkServer.sh start
/usr/bin/java
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

8、再次查看 zookeeper 进程的主机状态，发现选举成功，id 为 3 的节点成为了 follower。连接拒绝问题解决。

```
[prestyan@localhost bin]$ ./zkServer.sh status
/usr/bin/java
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: follower
```

9、查看 2 号主机，发现二号主机连接失败，没有连接上。

```
user@instant-contiki:/opt/zk/apache-zookeeper-3.6.3-bin/bin$ ./zkServer.sh statu
s
ZooKeeper JMX enabled by default
Using config: /opt/zk/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Error contacting service. It is probably not running.
user@instant-contiki:/opt/zk/apache-zookeeper-3.6.3-bin/bin$
```

10、猜测可能是虚拟机 hosts 配置问题导致 zookeeper 找不到该虚拟机。但是查看 hosts 发现配置完善没有遗漏。于是查看该主机 ip，发现主机 ip 发生变化。由于前面改成分布式部署添加新的虚拟机，导致原有的虚拟机 ip 分配发生变化，所以产生连接问题。

```
 7
 8 server.1=192.168.52.131:2888:3888
 9 server.2=192.168.52.130:2888:3888
10 server.3=192.168.52.128:2888:3888
```

```
user@instant-contiki: ~
File  Edit  View  Search  Terminal  Help
[prestyan@localhost ~]$ exit
logout
Connection to 192.168.52.128 closed.
user@instant-contiki:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:8a:23:c5
          inet addr:192.168.52.129  Bcast:192.168.52.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8a:23c5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:861 errors:0 dropped:0 overruns:0 frame:0
          TX packets:217 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:211358 (211.3 KB)  TX bytes:26825 (26.8 KB)
          Interrupt:19 Base address:0x2000
```

11、更改 ip 后重启 zkServer.sh 进程。启动成功。

```
user@instant-contiki:/opt/zk/apache-zookeeper-3.6.3-bin/bin$ ./zkServer.sh resta
rt
ZooKeeper JMX enabled by default
Using config: /opt/zk/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
ZooKeeper JMX enabled by default
Using config: /opt/zk/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Stopping zookeeper ... ./zkServer.sh: line 213: kill: (3388) - No such process
STOPPED
ZooKeeper JMX enabled by default
Using config: /opt/zk/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
user@instant-contiki:/opt/zk/apache-zookeeper-3.6.3-bin/bin$
```

12、查看主机状态，发现成为了 follower，说明 zookeeper 配置成功。

```
user@instant-contiki:/opt/zk/apache-zookeeper-3.6.3-bin/bin$ ./zkServer.sh statu
s
ZooKeeper JMX enabled by default
Using config: /opt/zk/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: follower
user@instant-contiki:/opt/zk/apache-zookeeper-3.6.3-bin/bin$
```

13、查看 1 号主机，根据选举算法，该主机应该被选择成为 leader。查看状态，发现与预期相符合。

```
prestyan@prestyan-virtual-machine:/opt/zookeeper/apache-zookeeper-3.6.3-bin/bin$
 ./zkServer.sh status
/usr/bin/java
ZooKeeper JMX enabled by default
Using config: /opt/zookeeper/apache-zookeeper-3.6.3-bin/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: leader
```

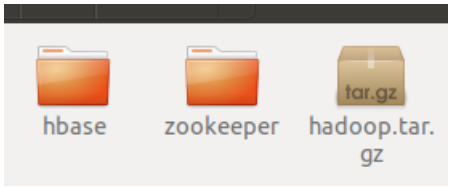
至此 zookeeper 安装部署全部成功完成。

三、HBase 集群安装部署

1、进入清华开源镜像网站下载 HBase 最新稳定版本，并解压到/opt 目录。

```
prestyan@prestyan-virtual-machine:/opt$ ll
总用量 276800
drwxr-xr-x  5 root     root          4096 3月   20 10:12 ./
drwxr-xr-x 27 root     root          4096 3月   12 10:48 ../
drwxrwxr-x 10 prestyan prestyan      4096 3月    7 15:14 hadoop/
drwxr-xr-x  7 root     root          4096 3月   20 10:12 hbase-2.4.11/
-rwxrw-rw-  1 prestyan prestyan 283415422 3月   20 10:05 hbase-2.4.11-bin.tar.gz*
drwxr-xr-x  4 prestyan prestyan      4096 3月   14 10:51 zookeeper/
```

2、解压成功，重命名文件为 hbase，方便访问。



3、进入 conf 目录，编辑 hbase-env.sh 文件。

```
prestyan@hadoop-master:/opt/hbase/conf$ ll
总用量 52
drwxr-xr-x 2 root root 4096 3月   22 10:45 ./
drwxr-xr-x 7 root root 4096 3月   20 10:12 ../
-rw-r--r-- 1 root root 1811 1月   22  2020 hadoop-metrics2-hbase.properties
-rw-r--r-- 1 root root 4284 1月   22  2020 hbase-env.cmd
-rw-r--r-- 1 root root 7699 3月   22 10:42 hbase-env.sh
-rw-r--r-- 1 root root 2257 1月   22  2020 hbase-policy.xml
-rw-r--r-- 1 root root 2615 3月   22 10:25 hbase-site.xml
-rw-r--r-- 1 root root 1169 1月   22  2020 log4j-hbtop.properties
-rw-r--r-- 1 root root 5735 1月   22  2020 log4j.properties
-rw-r--r-- 1 root root   30 3月   22 10:45 regionservers
prestyan@hadoop-master:/opt/hbase/conf$ sudo gedit hbase-env.sh
```

4、添加 java 路径与 hadoop 路径。由于 hadoop 版本更新，取消了 conf 目录，所以该目录可以改成/etc/hadoop，原来 conf 目录下所有文件迁移至/etc/hadoop 下。

```
# The java implementation to use.  Java 1.8+ require
 export JAVA_HOME=/home/prestyan/jdk/jdk1.8.0_11

# Extra Java CLASSPATH elements.  Optional.
 export HBASE_CLASSPATH=/opt/hadoop/conf
```

5、编辑 hbase-site.xml 文件，更改 hbase.rootdir 的名称路径与 hadoop 的 hdfs 一致。此处必须要使用 hostname，不能用 ip 地址，否则无法识别。同时添加 master 节点 ip 与 quorum 节点 ip。

Quorum 节点个数必须为奇数。

```
<property>
 <name>hbase.rootdir</name>
 <value>hdfs://hadoop-master:9000/hbase</value>
</property>


  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
  <property>

<property>
<name>hbase.master</name>
<value>192.168.52.131:60000</value>
</property>

<property>
 <name>hbase.zookeeper.quorum</name>
 <value>192.168.52.131,192.168.52.129,192.168.52.128</value>
</property>
```

6、打开 regionservers 文件，删去默认 localhost，添加 server 节点。

```
192.168.52.129
192.168.52.128
```

7、使用 scp 命令将 hbase 文件夹传送至其他主机，并在其他主机上更改相应配置，例如 dir。

```
prestyan@hadoop-master:/opt/hbase/conf$ sudo scp -r hbase prestyan@192.168.52.12
8:/home/prestyan
```

8、进入 bin 目录启动 hbase 进程，使用 jps 查看，发现进程启动后异常退出终止。于是进入相应 log 目录查看，发现报错 Unexpected KeeperException……

```
2022-03-22 12:42:02,326 INFO  [main] zookeeper.ZooKeeper: Session: 0x0 closed
2022-03-22 12:42:02,326 ERROR [main] regionserver.HRegionServer: Failed construction RegionServer
org.apache.hadoop.hbase.ZooKeeperConnectionException: master:160000x0, quorum=192.168.52.131:2181,192.168.52.129:2181,192.168.52.128:2181, baseZNode=/hbase Unexpected KeeperException creating base node
        at org.apache.hadoop.hbase.zookeeper.ZKWatcher.createBaseZNodes(ZKWatcher.java:260)
```

9、网络上找不到相类似的报错，猜测可能是 hadoop 或者 zkserver 启动启动失败。仔细查看上图 log 报错位置，发现错误可能出在 hadoop，于是重新启动 hadoop。启动前，将三台虚拟机中的 hadoop data 相应目录下数据清空，在 master 上重新使用 namenode -format 初始化，然后再启动 hadoop。Hadoop 启动成功后，再启动 hbase，等待一段时间后 HMstaer 进程正常运行，说明启动成功。

10、登录 hadoop-master:16010 查看 hbase 基本信息，发现 server 节点只有一个 hadoop-slave2，缺少了 hadoop-slave1 节点。即 slave1 没有正常连接上 master。

ServerName
hadoop-slave2,16020,1647925878528
Total:1

11、于是进入 slave1 主机，查看主机 ip 配置，发现并没有问题。进入 slave1 安装的 habse 目录下查看相应 log 文件。Log 提示问题所在：无法告诉 master 已经启动成功。

```
1836 2022-03-22 13:25:08,251 WARN  [regionserver/hadoop-slave1:16020] regionserver.HRegionServer: error telling master we are up
1837 org.apache.hbase.thirdparty.com.google.protobuf.ServiceException: org.apache.hadoop.hbase.HBaseIOException: Call to address=hadoop-ma
```

12、由于 zookeeper 框架是负责这一块工作的，slave 机器上报 master 出现问题很有可能是 zookeeper 发生了问题。于是查看 zookeeper 启动状态，发现没有问题。最后，在 opt 目录下使用 ll 命令发现，zookeeper 所在文件夹所属用户为 user 而非 prestyan，猜测问题可能处在用户权限。

```
user@hadoop-slave1:/opt$ ll
total 20
drwxr-xr-x  5 root     root     4096 Mar 22 12:22 ./
drwxr-xr-x 23 root     root     4096 Mar 21 12:15 ../
drwxrwxr-x 10 prestyan prestyan 4096 Mar 21 13:02 hadoop/
drwxr-xr-x  8 prestyan user     4096 Mar 22 12:22 hbase/
drwxr-xr-x  3 user     user     4096 Mar 14 11:20 zk/
```

13、更改文件夹所属，然后关闭 zkserver 进程，用 prestyan 用户重启进程。

```
user@hadoop-slave1:/opt$ sudo chown -R prestyan:prestyan zk
[sudo] password for user:
user@hadoop-slave1:/opt$ ll
total 20
drwxr-xr-x  5 root     root     4096 Mar 22 12:22 ./
drwxr-xr-x 23 root     root     4096 Mar 21 12:15 ../
drwxrwxr-x 10 prestyan prestyan 4096 Mar 21 13:02 hadoop/
drwxr-xr-x  8 prestyan user     4096 Mar 22 12:22 hbase/
drwxr-xr-x  3 prestyan prestyan 4096 Mar 14 11:20 zk/
user@hadoop-slave1:/opt$
```

14、启动以后，登录网页刷新页面，查看 server 节点状态，发现 hadoop-slave1 出现在表当中，说明 hbase 部署成功。

ServerName	Start time
hadoop-slave1,16020,1647928453539	Tue Mar 22 13:54:13 CST 2022
hadoop-slave2,16020,1647928454331	Tue Mar 22 13:54:14 CST 2022

15、使用 hbase shell 命令，进入 shell 命令行，进行建表操作。建造 score、sales 两个表，查看数据库信息。

```
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.11, r7e672a0da0586e6b7449310815182695bc6ae193, Tue Mar 15 10:31:00 PDT 2022
Took 0.0010 seconds
hbase:001:0> scan 'sales'
ROW                                                 COLUMN+CELL
0 row(s)
Took 0.5129 seconds
hbase:002:0> list
TABLE
sales
score
2 row(s)
Took 0.1111 seconds
=> ["sales", "score"]
```

16、使用命令给表添加一些列与数据，扫描 sales 表，可以看到表信息，说明 hbase 正常运行。

```
hbase:001:0> scan 'sales'
ROW                                   COLUMN+CELL
 bread                                column=num:10, timestamp=2022-03-22T22:01:52.820, value=value:502
 waterbottle                          column=num:3, timestamp=2022-03-22T22:02:51.030, value=value:108
```

至此 HBase 集群安装部署成功。