注意要点：

1. Linux 操作系统 CentOS 7，下载地址：http://mirrors.aliyun.com/centos/7/isos/x86_64/

CentOS 7 安装教程：https://www.runoob.com/w3cnote/vmware-install-centos7.html

也可使用 Ubuntu 20.04，下载地址：https://ubuntu.com/download/desktop

（尝试在 Ubuntu 下配置 Hadoop 成功）

2. Hadoop 环境搭建教程：https://www.tutorialspoint.com/hadoop/hadoop_enviornment_setup.htm

a. **（optional）**创建一个新用户 hadoop，将 hadoop 文件系统与 Unix 文件系统区分开：

$ （用户），#（ root）

```
$ su
  password:
# useradd hadoop
# passwd hadoop
  New passwd:
  Retype new passwd
```

b. 配置 SSH 与 Key 生成

```
# su hadoop
$ ssh-keygen -t rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

http://codesfusion.blogspot.com/2013/10/setup-hadoop-2x-220-on-ubuntu.html

3. 使用 java-jdk-8:

```
$ sudo apt-get install openjdk-8-jdk
$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

```
$ vim ~/.bashrc

在.bashrc 文件末尾添加：
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME

export HADOOP_OPTS="$HADOOP_OPTS -Djava.library.path=$HADOOP_HOME/lib/native"

$ source ~/.bashrc
```

4. Hadoop3.3.1 下载安装：

```
# wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz --no-check-certificate
# tar zxf hadoop-3.3.1.tar.gz
# mkdir hadoop
# mv hadoop-3.3.1/* hadoop
```

检查 hadoop 是否安装成功

```
$ hadoop version
Hadoop 3.3.1
Source code repository https://github.com/apache/hadoop.git -r
a3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled by ubuntu on 2021-06-15T05:13Z
Compiled with protoc 3.7.1
From source with checksum 88a4ddb2299aca054416d6b7f81ca55
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.1.jar
```

5. Hadoop 配置

先修改文件读写权限，默认为只读

```
$ cd $HADOOP_HOME/etc/hadoop
$ su
  Password:
# chmod 777 hadoop-env.sh core-site.xml hdfs-site.xml yarn-site.xml mapred-site.xml
```

a. 修改 hadoop-env.sh：

覆盖 **JAVA_HOME**

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

b. 修改 core-site.xml:

在<configuration>中间添加：

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

c. 修改 hdfs-site.xml：

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/jianbo/hadoopinfra/hdfs/namenode </value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/jianbo/hadoopinfra/hdfs/datanode </value>
  </property>
</configuration>
```

d. 修改 yarn-site.xml：

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
   </property>
</configuration>
```

e. 修改 mapred-site.xml：

```
<configuration>
  <property>
     <name>yarn.app.mapreduce.am.env</name>
     <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
  </property>
 <property>
      <name>mapreduce.map.env</name>
      <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
 </property>
 <property>
     <name>mapreduce.reduce.env</name>
     <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
 </property>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
     <name>mapreduce.application.classpath</name>

<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
    </property>
</configuration>
```

6. 修改 hadoop 权限

```
$ su
# vim /etc/sudoers
(如果是只读权限，需修改文件的权限，chmod u+w /etc/sudoers，设置读写权限可另行查看相关命令)

找到 root ALL=(ALL) ALL 这一行，
在 root ALL=(ALL) ALL 下面一行增加
hadoop ALL=(ALL) ALL
```

当出现以下情况：

```
$ hdfs namenode -format
WARNING: /usr/local/hadoop/logs does not exist. Creating.
mkdir: cannot create directory ‘/usr/local/hadoop/logs’: Permission denied
ERROR: Unable to create /usr/local/hadoop/logs. Aborting.
```

解决办法：

```
[hadoop@centos7 ~]$ sudo chmod 777 -R /usr/local/hadoop/
```

7. 验证 hadoop 安装

a. Name Node 配置

```
[hadoop@centos7 ~]$ cd
[hadoop@centos7 ~]$ hdfs namenode -format
```

或者

```
[hadoop@centos7 ~]$ hadoop namenode -format
```

Expected results:

```
2021-12-08 06:19:43,594 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ubuntu/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.1……
……
2021-12-08 06:19:45,486 INFO common.Storage: Storage directory
/home/jianbo/hadoopinfra/hdfs/namenode has been successfully formatted.
2021-12-08 06:19:45,508 INFO namenode.FSImageFormatProtobuf: Saving image file
/home/jianbo/hadoopinfra/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 using no
compression
2021-12-08 06:19:45,610 INFO namenode.FSImageFormatProtobuf: Image file
/home/jianbo/hadoopinfra/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 of size
401 bytes saved in 0 seconds .
2021-12-08 06:19:45,616 INFO namenode.NNStorageRetentionManager: Going to retain 1 images
with txid >= 0
2021-12-08 06:19:45,635 INFO namenode.FSNamesystem: Stopping services started for active state
2021-12-08 06:19:45,635 INFO namenode.FSNamesystem: Stopping services started for standby state
2021-12-08 06:19:45,638 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when
meet shutdown.
2021-12-08 06:19:45,639 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
************************************************************/
```

b. 验证 hadoop dfs

```
$ start-dfs.sh
```

Expected results：

```
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
```

当出现 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable 时：

~/.bashrc： `export HADOOP_OPTS="$HADOOP_OPTS -Djava.library.path=$HADOOP_HOME/lib/native"`

c. 验证 yarn script

```
$ start-yarn.sh
```

Expected results：

```
Starting resourcemanager
Starting nodemanagers
```

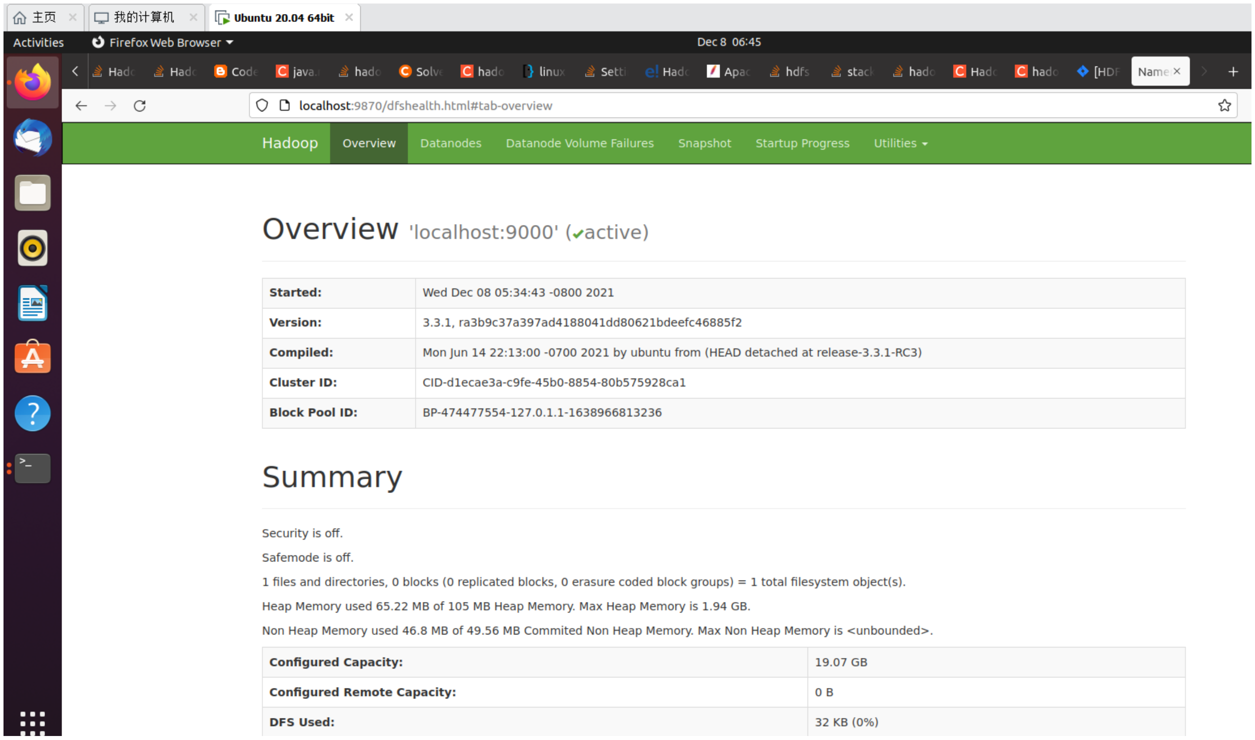
*可以使用 jps 查看当前 job

```
$ jps
41744 NameNode
40642 NodeManager
42117 SecondaryNameNode
40488 ResourceManager
42270 Jps
```

d. 在浏览器进入 hadoop

```
http://localhost:9870/
```

Expected results：



Overview 'localhost:9000' (✓active)

Started:	Wed Dec 08 05:34:43 -0800 2021
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Mon Jun 14 22:13:00 -0700 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-d1ecae3a-c9fe-45b0-8854-80b575928ca1
Block Pool ID:	BP-474477554-127.0.1.1-1638966813236

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 65.22 MB of 105 MB Heap Memory. Max Heap Memory is 1.94 GB.

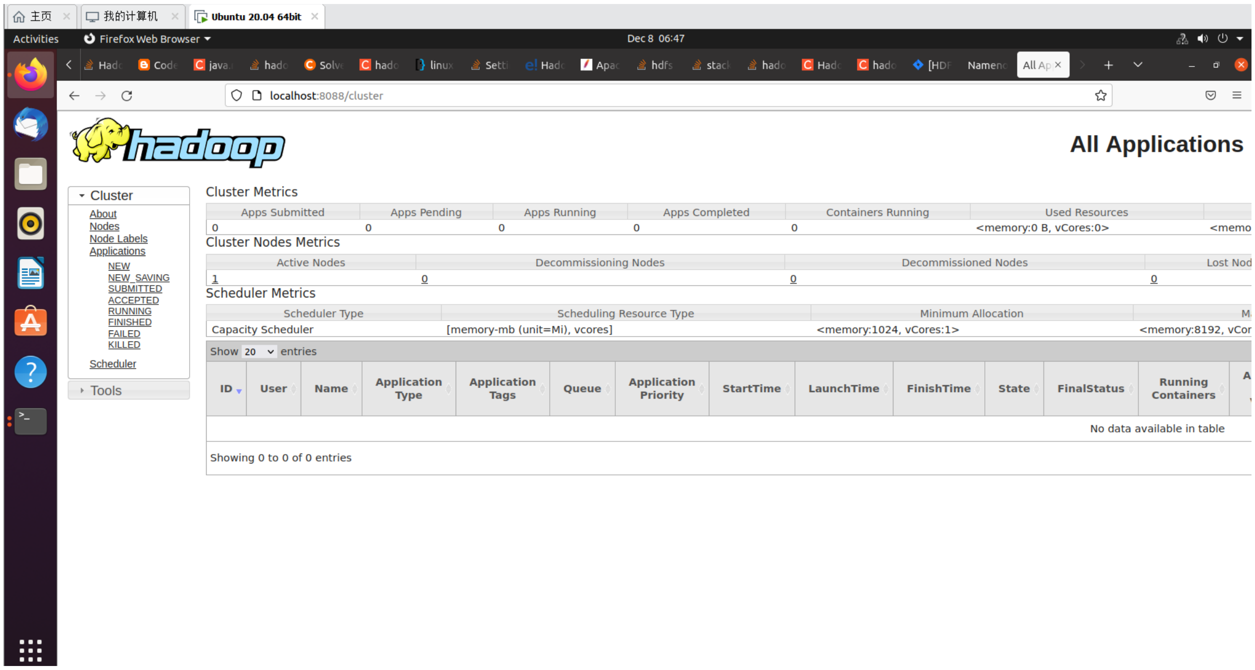
Non Heap Memory used 46.8 MB of 49.56 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	19.07 GB
Configured Remote Capacity:	0 B
DFS Used:	32 KB (0%)

e. 访问 All Applications on Cluster

```
http://localhost:8088/
```

Expected results：



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	
0	0	0	0	0	<memory:0 B, vCores:0>	<memo

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nod
1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	M
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCor

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	A
No data available in table													

Showing 0 to 0 of 0 entries

Run Example： https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

1. run a MapReduce job locally.

a. Format the filesystem:

```
$ hdfs namenode -format
```

b. Start NameNode daemon and DataNode daemon:

```
$ start-dfs.sh
```

The hadoop daemon log output is written to the $HADOOP_LOG_DIR directory (defaults to $HADOOP_HOME/logs).

c. Browse the web interface for the NameNode; by default it is available at:

```
http://localhost:9870/
```

d. Make the HDFS directories required to execute MapReduce jobs:

```
$ hdfs dfs -mkdir hdfs://localhost:9000/user/jianbo
```

e. Copy the input files into the distributed filesystem:

```
$ hdfs dfs -mkdir hdfs://localhost:9000/user/jianbo/input
$ hdfs dfs -put $HADOOP_HOME/etc/hadoop/*.xml input
```

f. Run some of the examples provided:

```
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep
input output 'dfs[a-z.]+'
```

g. Examine the output files: Copy the output files from the distributed filesystem to the local filesystem and examine them:

```
$ hdfs dfs -get output output
$ cat output/*
```

or

View the output files on the distributed filesystem:

```
$ hdfs dfs -cat output/*
```

h. When you’re done, stop the daemons with:

```
$ stop-dfs.sh
```