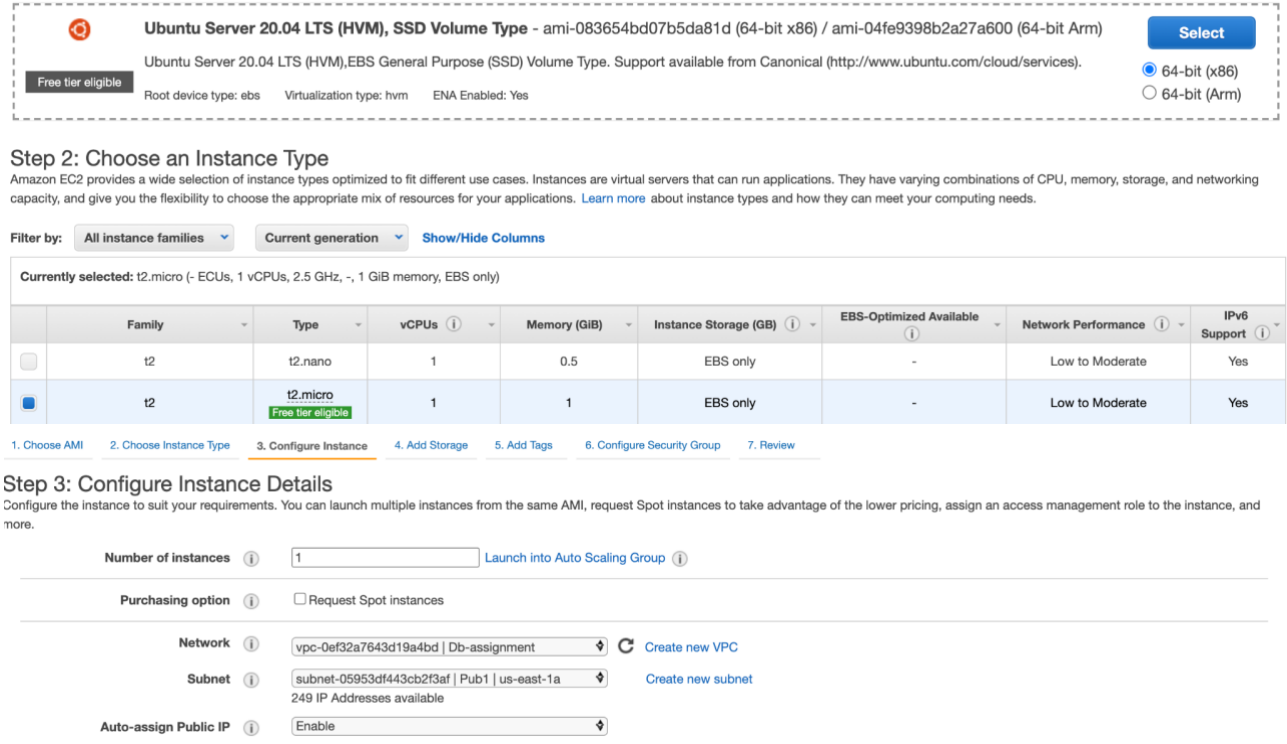
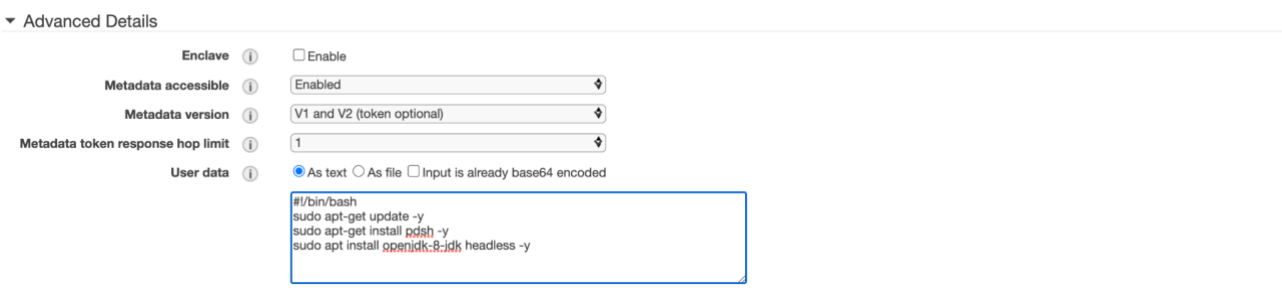
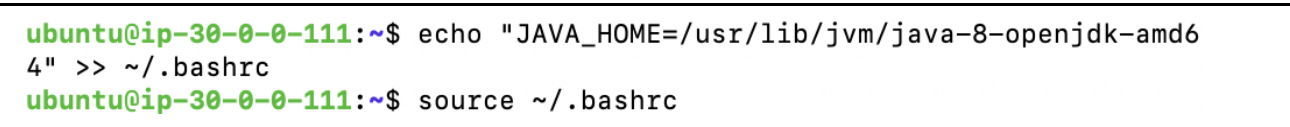
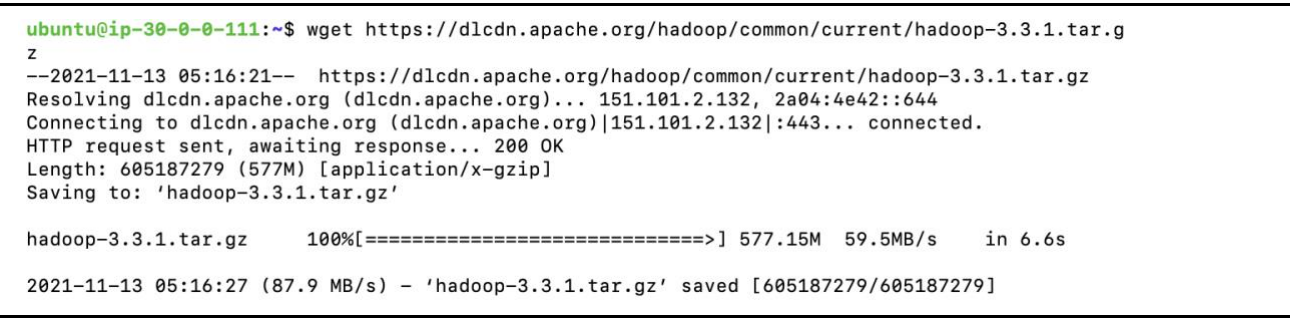
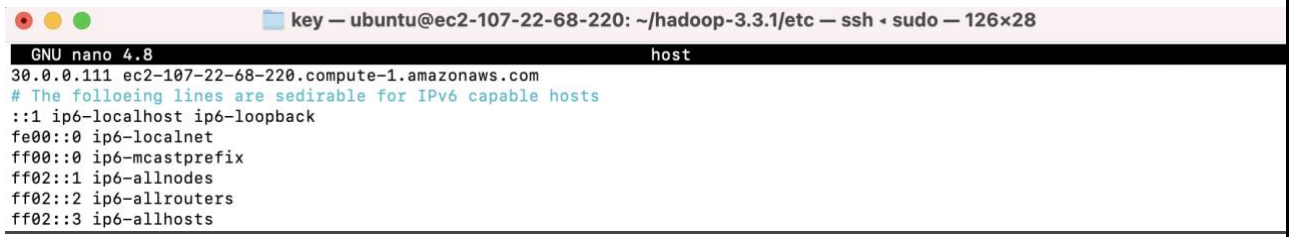
Group member	Sisi Zhang, Harish Ram
Launch a new EC2 Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-cd0f5cb6 (free tier) inside LabVpc, PublicSubnet1, with auto assigned public IP enabled	

As advanced Details, select user data as text and provide the following script #!/bin/bash sudo apt-get update -y sudo apt-get install pdsh -y sudo apt install openjdk-8-jdk headless -y	
Create a new security group called hadoop that would accept all traffic, from all protocols, all port rage, from anywhere	
Login in the ec2 using putty (Windows) or ssh (Lunix)	
Update ~/.bashrc with the java home	
Download Hadoop	

Uncompress hadoop	`[ubuntu@ip-30-0-0-111:~$ tar -xvf hadoop-3.3.1.tar.gz`
	`hadoop-3.3.1/`
	`hadoop-3.3.1/licenses-binary/`
	`hadoop-3.3.1/licenses-binary/LICENSE-hsql.txt`
	`hadoop-3.3.1/licenses-binary/LICENSE-zstd-jni.txt`
	`hadoop-3.3.1/licenses-binary/LICENSE-paranamer.txt`

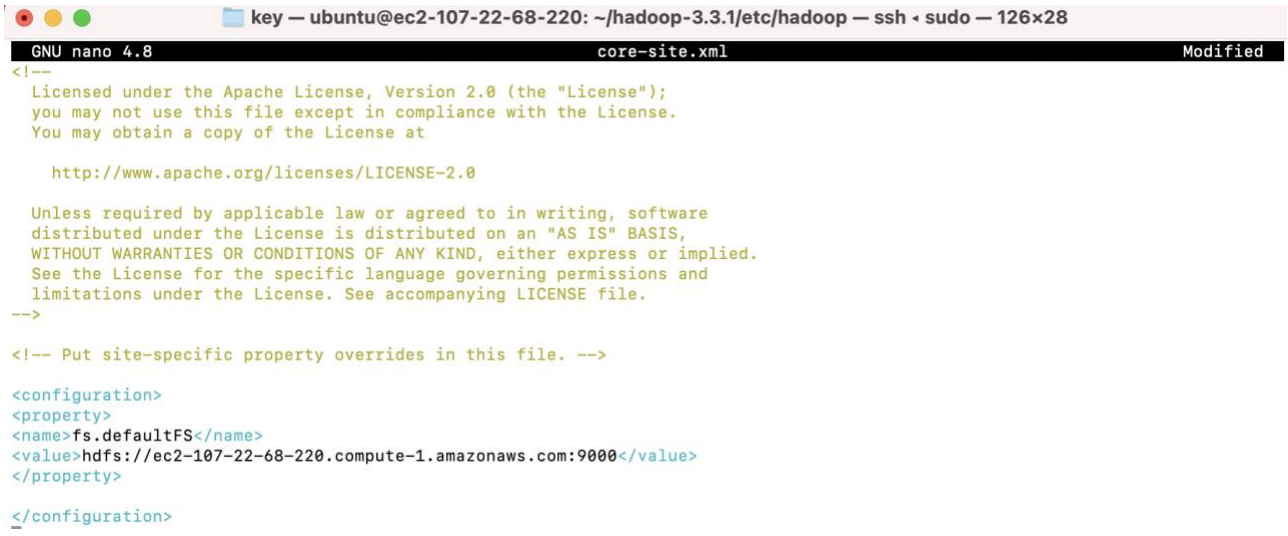
run ifconfig and make note of the ip address provided by ifconfig. Note if you stop and restart your EC2 instance again, this number will change and you will have to reconfigure /etc/hosts, /etc/hostname, and rsa	`[ubuntu@ip-30-0-0-111:~$ ifconfig`
	`eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001`
	`inet 30.0.0.111  netmask 255.255.255.0  broadcast 30.0.0.255`
	`inet6 fe80::8b0:29ff:fe05:ecaf  prefixlen 64  scopeid 0x20<link>`
	`ether 0a:b0:29:05:ec:af  txqueuelen 1000  (Ethernet)`
	`RX packets 462160  bytes 682076186 (682.0 MB)`
	`RX errors 0  dropped 0  overruns 0  frame 0`
	`TX packets 79270  bytes 11367712 (11.3 MB)`
	`TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0`
	
	`lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536`
	`inet 127.0.0.1  netmask 255.0.0.0`
	`inet6 ::1  prefixlen 128  scopeid 0x10<host>`
	`loop  txqueuelen 1000  (Local Loopback)`
	`RX packets 150  bytes 14246 (14.2 KB)`
	`RX errors 0  dropped 0  overruns 0  frame 0`
	`TX packets 150  bytes 14246 (14.2 KB)`
	`TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0`

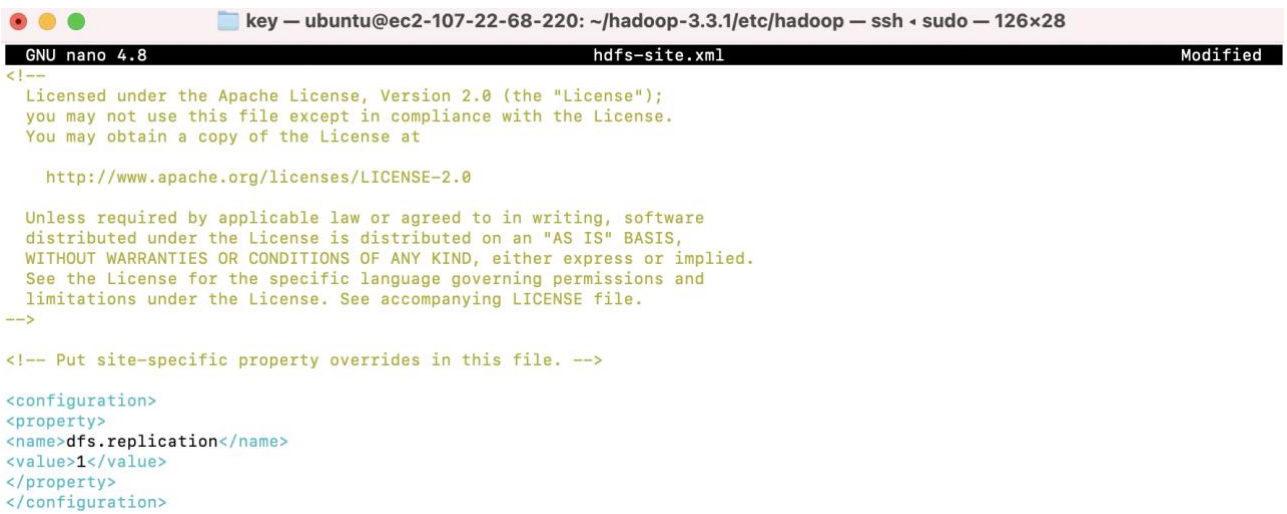
In my case, the ip address is 172.31.35.158 (notice that is this aws private IP address, which is ok)	30.0.0.111
Update /etc/hosts using this IP address and the EC2 instance **Public** DNS (IPv4). To do so, you can use sudo nano /etc/hosts **Make sure you delete the entry to 127.0.0.0 localhost**	

Update hostname with the EC2 instance Public DNS (IPv4). In my case, my EC2 DNS is 'ec2-34- 228-184-174.compute 1.amazonaws.com' , therefore to do so I used sudo hostnamectl set-hostname 'ec2-34-228-184-174.compute 1.amazonaws.com'	[ubuntu@ec2-54-226-110-108:~/hadoop-3.3.1/etc$ sudo hostnamectl set-hostname ec2-107-22-68-220.compute-1.amazonaws.com [ubuntu@ec2-54-226-110-108:~/hadoop-3.3.1/etc$ hostname ec2-107-22-68-220.compute-1.amazonaws.com
Create a ssh key	[ubuntu@ec2-54-226-110-108:~/hadoop-3.3.1/etc$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa Generating public/private rsa key pair. /home/ubuntu/.ssh/id_rsa already exists. [Overwrite (y/n)? y Your identification has been saved in /home/ubuntu/.ssh/id_rsa Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub The key fingerprint is: [redacted] ubuntu@ec2-107-22-68-220.compute-1.amazonaws.com The key's randomart image is: [redacted] [ubuntu@ec2-54-226-110-108:~/hadoop-3.3.1/etc$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Test ssh. If you are not able to ssh to localhost, you did something wrong. First time, ubuntu will ask you confirm: type Y	[ubuntu@ec2-54-226-110-108:~/hadoop-3.3.1/etc$ ssh localhost Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1021-aws x86_64) * Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage System information as of Sun Nov 14 05:41:07 UTC 2021 System load: 0.0 Processes: 106 Usage of /: 50.4% of 7.69GB Users logged in: 1 Memory usage: 22% IPv4 address for eth0: 30.0.0.111 Swap usage: 0% * Ubuntu Pro delivers the most comprehensive open source security and compliance features. https://ubuntu.com/aws/pro 11 updates can be applied immediately. To see these additional updates run: apt list --upgradable Last login: Sun Nov 14 05:26:27 2021 from 72.196.199.101 [ubuntu@ec2-107-22-68-220:~$ ls
If you successfully have been ssh to localhost, reboot the ec2 instance using AWS console and login again using putty	

Update Hadoop env with JAVA_HOME. To do so: cd hadoop-2.7.4 nano etc/hadoop/hadoop env.sh	GNU nano 4.8 hadoop-env.sh ### # Generic settings for HADOOP ### # Technically, the only required environment variable is JAVA_HOME. # All others are optional. However, the defaults are probably not # preferred. Many sites configure these options outside of Hadoop, # such as in /etc/profile.d # The java implementation to use. By default, this environment # variable is REQUIRED on ALL platforms except OS X! export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Test your configuration by using the stand-alone Hadoop installation	[ubuntu@ec2-107-22-68-220:~$ cd hadoop-3.3.1 [ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ ls LICENSE-binary NOTICE-binary README.txt etc include lib licenses-binary sbin unput LICENSE.txt NOTICE.txt bin hadoop-env.sh input libexec pp share [ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ cp etc/hadoop/*.xml input [ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep input output 'dfs[a-z.]+' 2021-11-14 05:57:29,800 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties 2021-11-14 05:57:30,029 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s). [ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ cat output/* cd haddop-2.7.4

	mkdir input cp etc/hadoop/*.xml input bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-haddop-2.7.4.jar grep input output 'dfs[a-z.]+' cat output/*
Check output: more output/part*	[ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ cat output/* 1 dfsadmin
Pseudo-Distributed Operation Hadoop can also be run on a single-node in a pseudo distributed mode where each Hadoop daemon runs in a separate Java process.	
Change etc/hadoop/core-site.xml by adding the following properties under configuration. Remember to use EC2 instance Public DNS (IPv4) you used previously. <configuration> <property> <name>fs.defaultFS</name>	

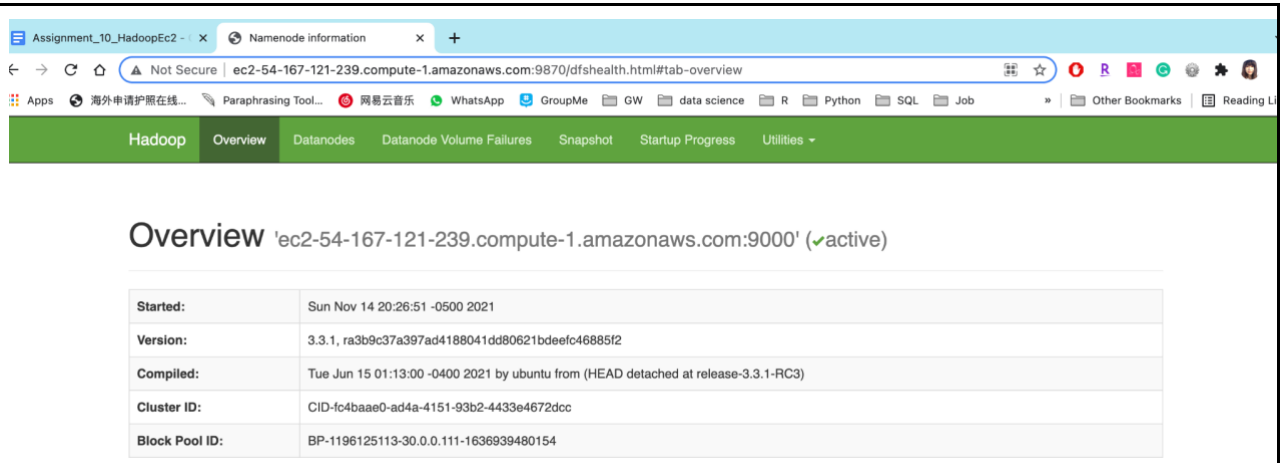
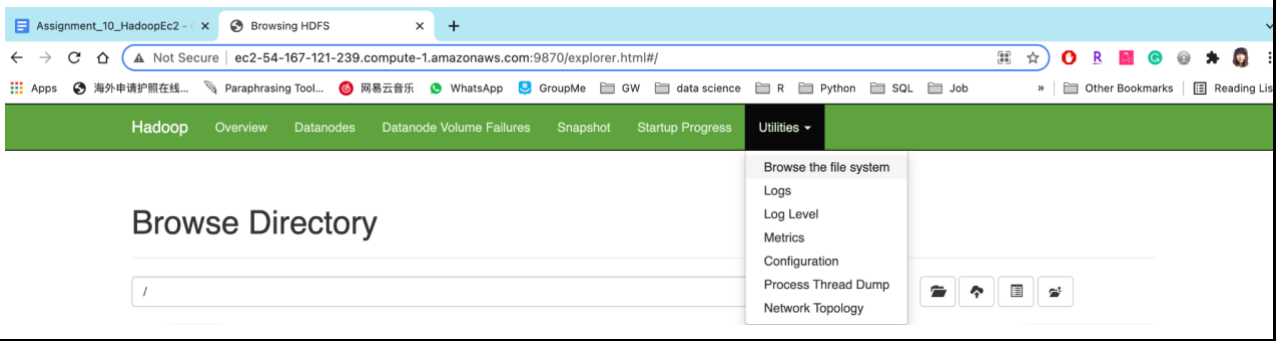
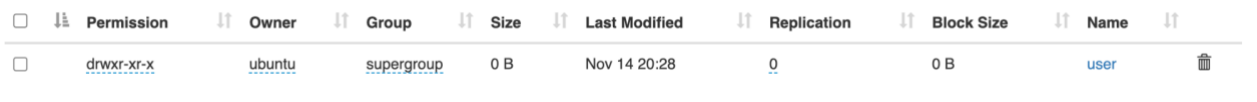
<value>hdfs://ec2-34-228-184-174.compute 1.amazonaws.com:9000</value > </property> </configuration>	
Change etc/hadoop/hdfs-site.xml by adding the following properties under configuration. <configuration> <property> <name>dfs.replication</name> <value>1</value> </property> </configuration>	
Now let us run a MapReduce job locally	
Format the filesystem:	$ bin/hdfs namenode -format ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hdfs namenode -format
Start NameNode daemon and DataNode daemon:	$ sbin/start-dfs.sh ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ sbin/start-dfs.sh

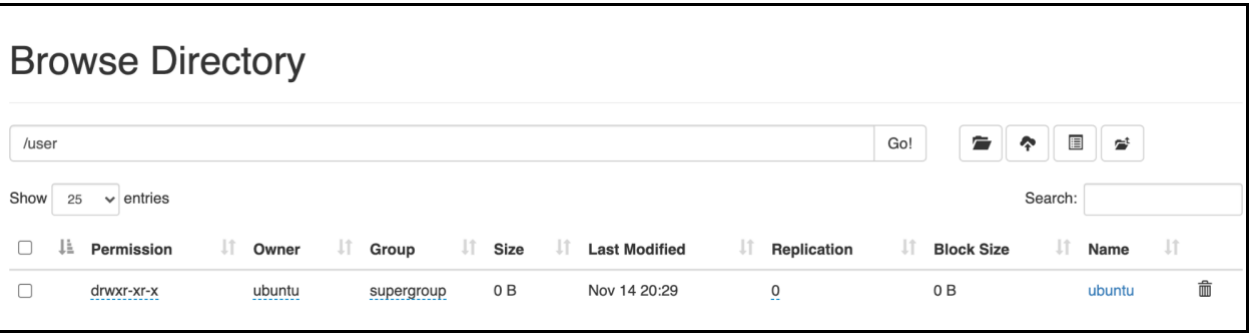
Make the HDFS directories required to execute MapReduce jobs:	$ bin/hdfs dfs -mkdir /user
	ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hdfs dfs -mkdir /user
	$ bin/hdfs dfs -mkdir /user/ubuntu
	ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hdfs dfs -mkdir /user/ubuntu

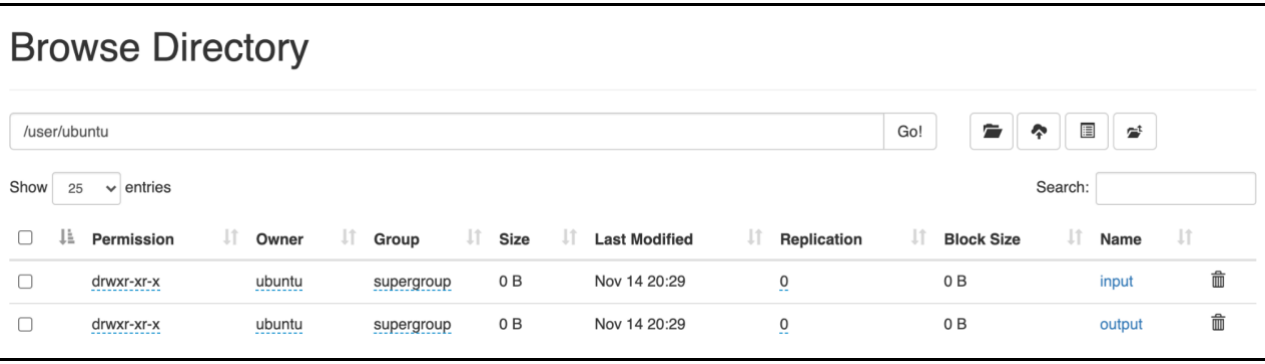
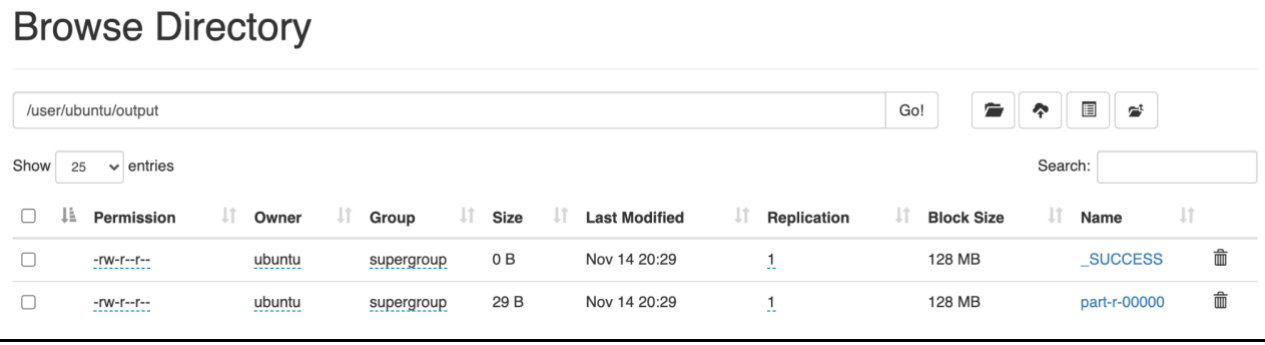
Copy the input files into the distributed filesystem (ignore the WARN exceptions)	$ bin/hdfs dfs -mkdir input
	ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hdfs dfs -mkdir input
	$ bin/hdfs dfs -put etc/hadoop/*.xml input
	ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hdfs dfs -put etc/hadoop/*.xml input

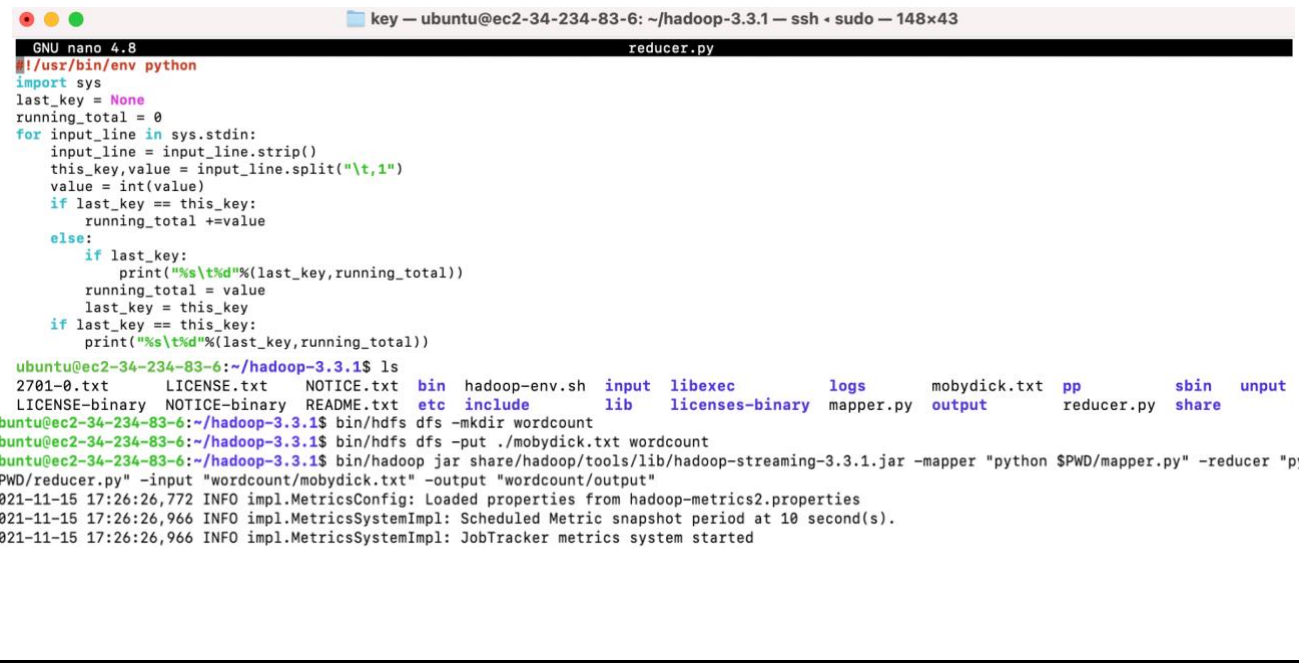
Run some of the examples provided:	`$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep input output 'dfs[a-z.]+'` `[ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar grep input output 'dfs[a-z.]+'`

Examine the output files: Copy the output files from the distributed filesystem to the local filesystem and examine them:	`$ bin/hdfs dfs -get output output`
	`[ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ bin/hdfs dfs -get output output`
	`$ cat output/*`
	`[ubuntu@ec2-107-22-68-220:~/hadoop-3.3.1$ cat output/*`
	`cat: output/output: Is a directory`
	`1 dfsadmin`

Open a browser and check results using Hadoop web UI	
Select Utilities/Browse the file system, then click Go!	
Click on user	

Click on ubuntu	

Click on output	
Click on part-r-0000 Download the file and check results	
Wordcount Test	


key — ubuntu@ec2-34-234-83-6: ~/hadoop-3.3.1 — ssh ‹ sudo — 148×43
GNU nano 4.8 reducer.py
#!/usr/bin/env python
import sys
last_key = None
running_total = 0
for input_line in sys.stdin:
input_line = input_line.strip()
this_key,value = input_line.split("\t,1")
value = int(value)
if last_key == this_key:
running_total +=value
else:
if last_key:
print("%s\t%d"%(last_key,running_total))
running_total = value
last_key = this_key
if last_key == this_key:
print("%s\t%d"%(last_key,running_total))
ubuntu@ec2-34-234-83-6:~/hadoop-3.3.1$ ls
2701-0.txt LICENSE.txt NOTICE.txt bin hadoop-env.sh input libexec logs mobydick.txt pp sbin unput
LICENSE-binary NOTICE-binary README.txt etc include lib licenses-binary mapper.py output reducer.py share
ubuntu@ec2-34-234-83-6:~/hadoop-3.3.1$ bin/hdfs dfs -mkdir wordcount
ubuntu@ec2-34-234-83-6:~/hadoop-3.3.1$ bin/hdfs dfs -put ./mobydick.txt wordcount
ubuntu@ec2-34-234-83-6:~/hadoop-3.3.1$ bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar -mapper "python $PWD/mapper.py" -reducer "python
$PWD/reducer.py" -input "wordcount/mobydick.txt" -output "wordcount/output"
2021-11-15 17:26:26,772 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2021-11-15 17:26:26,966 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-11-15 17:26:26,966 INFO impl.MetricsSystemImpl: JobTracker metrics system started