Programming Assignment 5

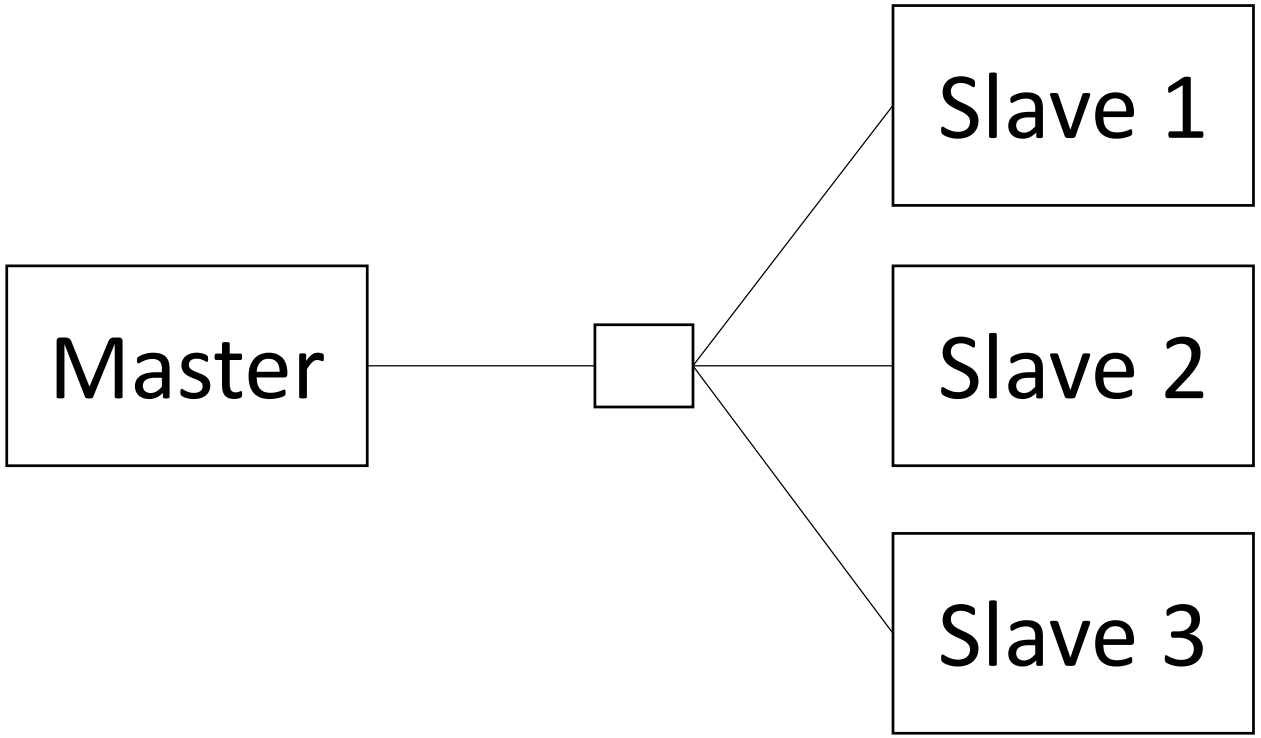
CS 4740 Cloud Computing

Department of Computer Science,
University of Virginia, USA

Goal

- Gain hands-on experience on how to configure a multi-node Hadoop cluster on AWS EC2 instances

Hadoop Architecture



Steps

- Tutorial videos

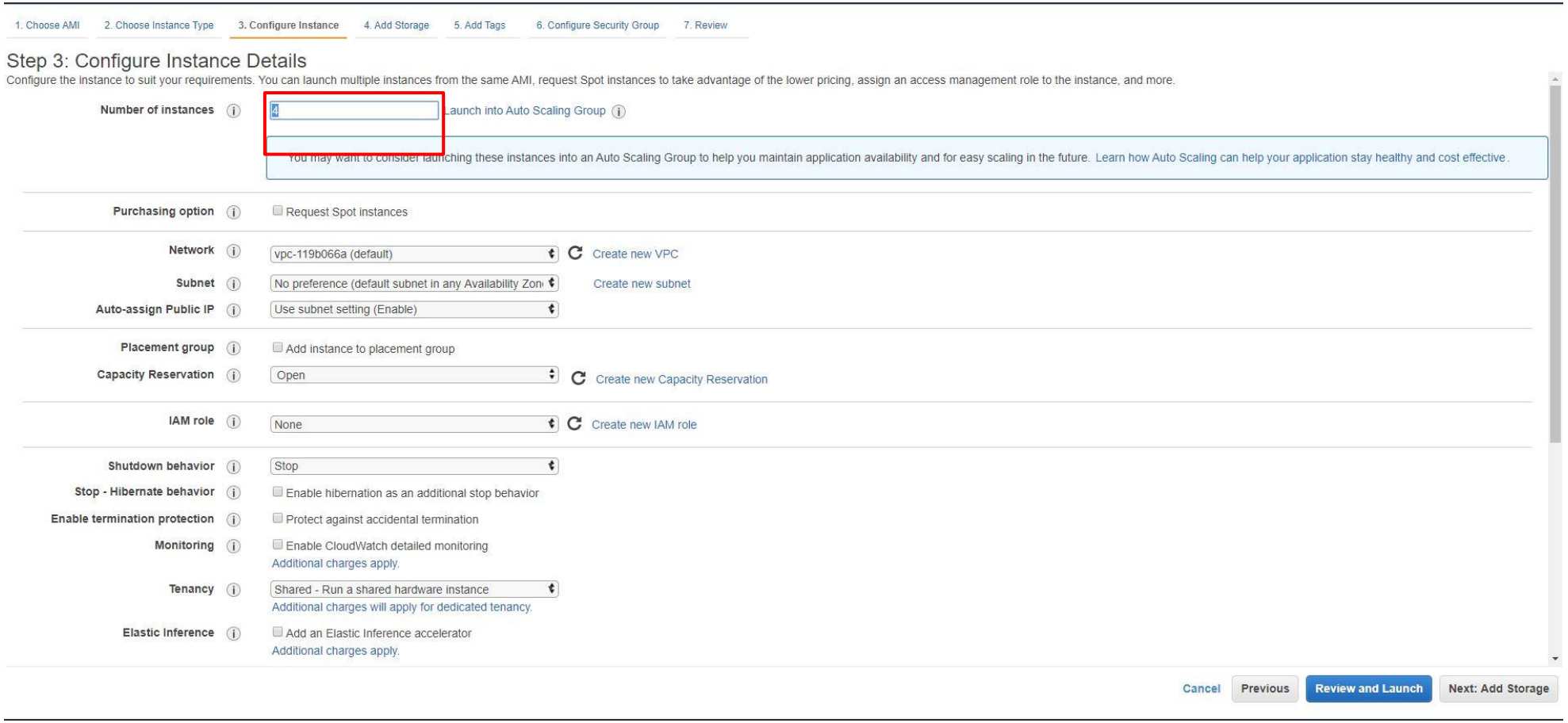
- Step 1: Request four t2.micro instances on EC2 – free tier
  - Remember to shut down all of them after the assignment
  - Pay attention to the Ubuntu OS version you request. It affects what JAVA version you need to install

  Video Link: https://www.youtube.com/watch?v=cr5RmnyWbYw

- Step 2: Login to the EC2 instances on Windows PC
  - Using Putty and WinSCP
  - Terminal on Macbook - search how to connect to EC2 instance on Macbook

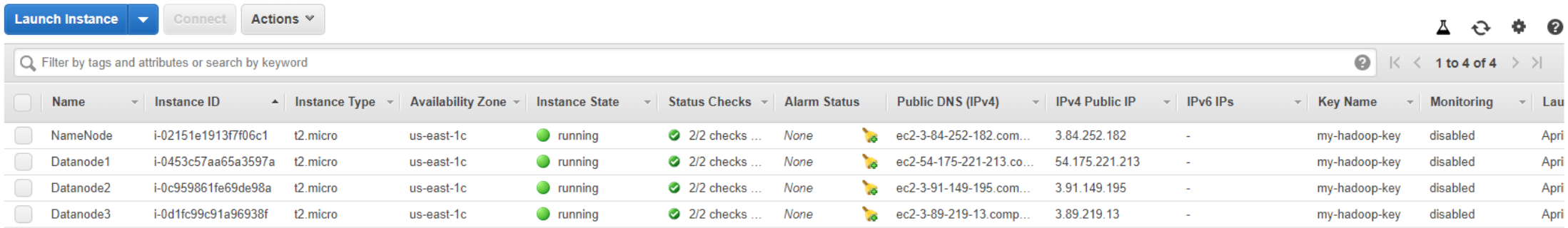
  Video Link: https://www.youtube.com/watch?v=lRQqR0Fm1oE

Step 1: Request four t2.micro instances on EC2 – free tier
Configuring the instances simultaneously



Step 1: Request four t2.micro instances on EC2 – free tier

Final Outcome



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Lau
NameNode	i-02151e1913f7f06c1	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-3-84-252-182.com...	3.84.252.182	-	my-hadoop-key	disabled	Apri
Datanode1	i-0453c57aa65a3597a	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-54-175-221-213.co...	54.175.221.213	-	my-hadoop-key	disabled	Apri
Datanode2	i-0c959861fe69de98a	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-3-91-149-195.com...	3.91.149.195	-	my-hadoop-key	disabled	Apri
Datanode3	i-0d1fc99c91a96938f	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-3-89-219-13.comp...	3.89.219.13	-	my-hadoop-key	disabled	Apri

Step 3: setting up Passwordless SSH

- Setup Passwordless SSH
  - Nodes communicate frequently with each other
  - Too slow to authenticate every time
  - Command: ssh-keygen -f ~/.ssh/sshkey_rsa -t rsa -P ""
  - Replace the name sshkey_rsa with id_rsa and also replace the name for the following steps that use sshkey_rsa

  Video Link: https://www.youtube.com/watch?v=u7fIf_R-gaM&t=3s

Step 3: setting up Passwordless SSH Commands

```
# set permission level on PEM on all nodes
sudo chmod 600 ~/.ssh/my-hadoop-key.pem
#copy config and pem file to all nodes
scp ~/.ssh/my-hadoop-key.pem ~/.ssh/config Datanode1:~/.ssh
scp ~/.ssh/my-hadoop-key.pem ~/.ssh/config Datanode2:~/.ssh
scp ~/.ssh/my-hadoop-key.pem ~/.ssh/config Datanode3:~/.ssh
#generate key files on Namenode
ssh-keygen -f ~/.ssh/id_rsa -t rsa -P ""
#copy the generated key from sshkey_rsa.pub to authorized keys of namenodes
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
#copy the file to all datanodes
cat ~/.ssh/id_rsa.pub | ssh Datanode1 'cat >> ~/.ssh/authorized_keys'
cat ~/.ssh/id_rsa.pub | ssh Datanode2 'cat >> ~/.ssh/authorized_keys'
cat ~/.ssh/id_rsa.pub | ssh Datanode3 'cat >> ~/.ssh/authorized_keys'
```

Step 4: Install Hadoop on the four nodes

- Install Hadoop on the four nodes
  - Install JAVA
  - Configure environment variables, e.g.,
    - JAVA_HOME
    - HADOOP_HOME

    Video Link: https://www.youtube.com/watch?v=Z0cOE2SRo5c

Step 4: Install Hadoop on the four nodes Commands-1

- #Install Hadoop - Following commands are for all four machines

sudo apt-get update

- #Installing Java on Ubuntu 16.04

sudo apt-get install openjdk-8-jdk-headless

- #check java version

java -version

- #Download Hadoop from Apache to Downloads folder in home

wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz -P ~/Downloads/Hadoop

- #Uncompress the Hadoop tar file into the /usr/local folder

sudo tar zxvf ~/Downloads/Hadoop/hadoop-2.7.3.tar.gz -C /usr/local

- #Move all Hadoop related file from /usr/local to /usr/local/hadoop

sudo mv /usr/local/hadoop-* /usr/local/hadoop

Step 4: Install Hadoop on the four nodes Commands-2

- #Set environment variables on all the nodes in /home/ubuntu/.profile

```
export JAVA_HOME=/usr
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
```

- #load variables

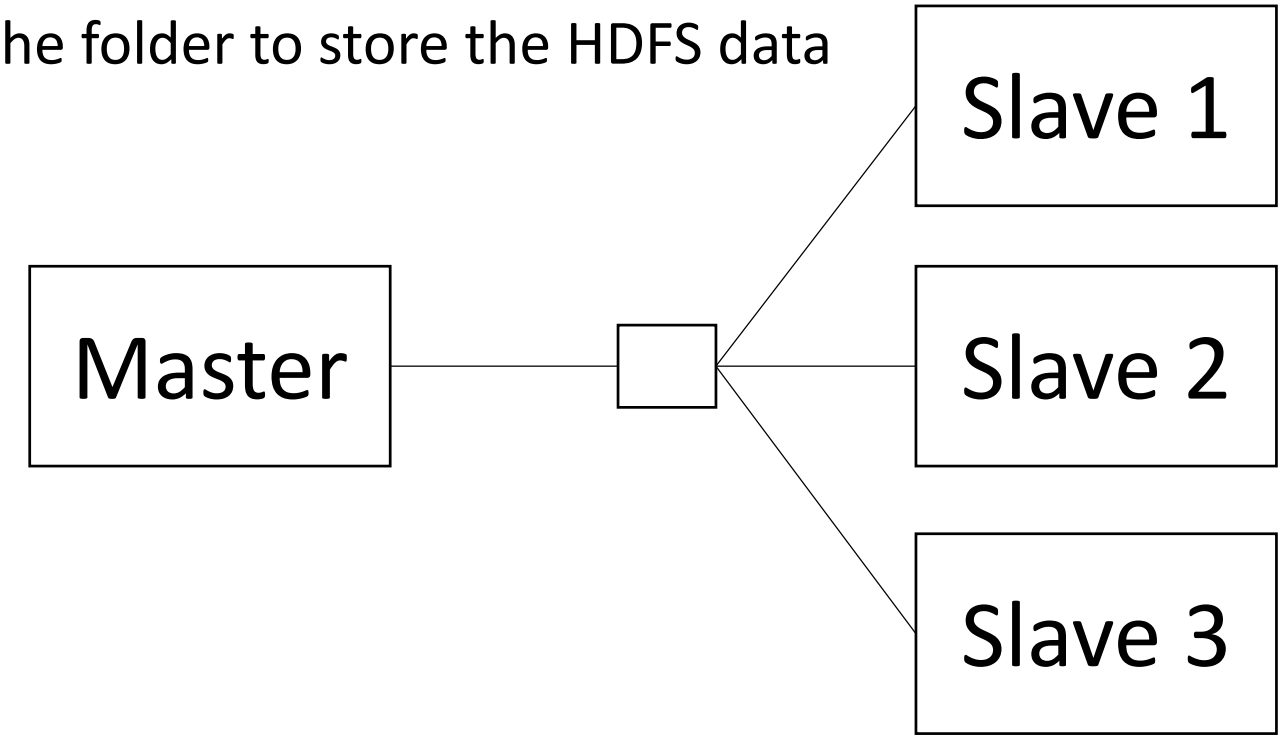
```
. ~/.profile
```

- #change ownership of hadoop folder

```
sudo chown -R ubuntu /usr/local/hadoop
```

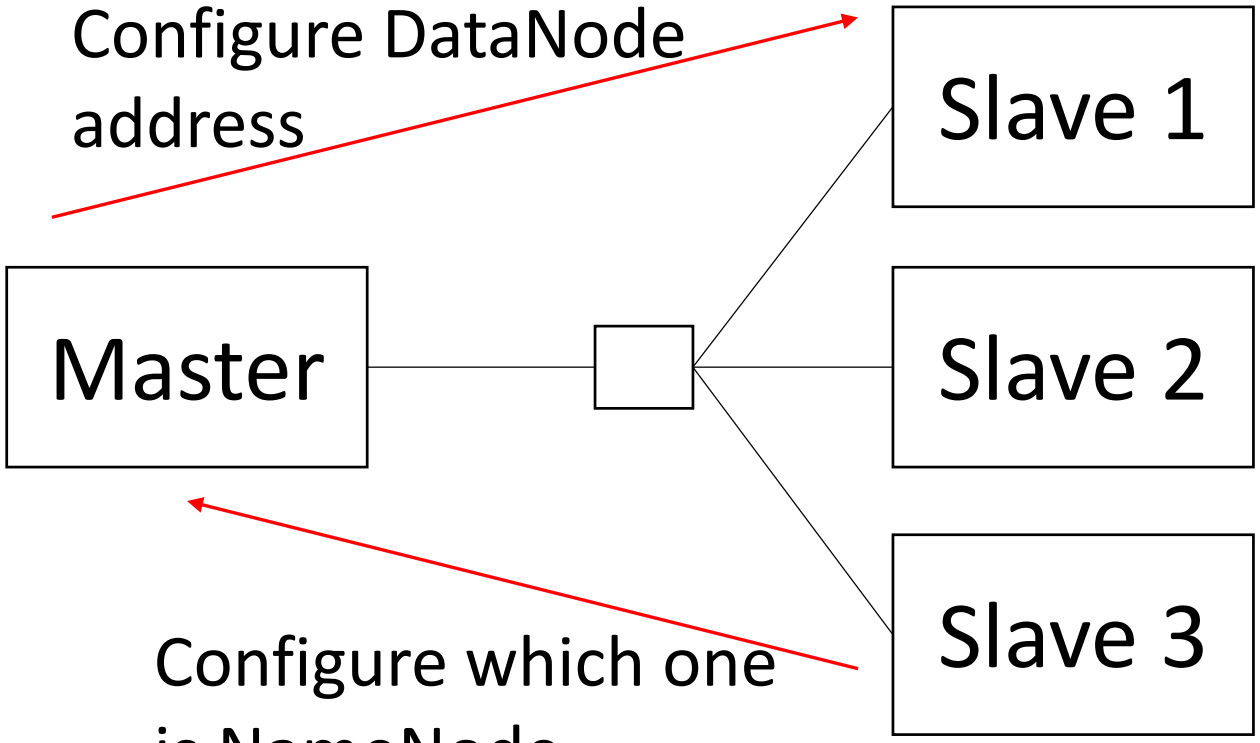
Step 5: Configure NameNode and DataNode

- Configure NameNode and DataNode
  - Hadoop distributed file system - HDFS
  - Configure the folder to store the HDFS data



Step 5: Configure NameNode and DataNode

- Configure NameNode and DataNode



Step 5: Configure NameNode and DataNode Sub-Steps

- 3 types of configuration
- Sub-step1: Applicable to both Namenode and Datanodes
- Sub-step2: Applicable to only Namenode
- Sub-step3: Applicable to only Datanodes

Video Link: https://www.youtube.com/watch?v=PF1dUAKs_bg

https://www.youtube.com/watch?v=k-DGPIwfitM

Step 5: Configure NameNode and DataNode sub-step1: Applicable to both type of nodes

- Change in core-site.xml (Excluded in the following command)
- Change in hadoop-env.sh
- Change in mapred-site.xml
- Change in yarn-site.xml

One way to do it easily: Only change in Namenode and scp the files datanodes.

E.g., scp $HADOOP_CONF_DIR/mapred-site.xml $HADOOP_CONF_DIR/yarn-site.xml $HADOOP_CONF_DIR/core-site.xml $HADOOP_CONF_DIR/hadoop-env.sh Datanode1:$HADOOP_CONF_DIR

Datanode1: As set in config (in slide 8: scp ~/.ssh/my-hadoop-key.pem ~/.ssh/config Datanode1:~/.ssh)

Replace Datanode1 with each of other Datanodes and run the above command

Step 5: Configure NameNode and DataNode Applicable to only namenode

- Change in /etc/hosts
- Change in hdfs-site.xml
- Creating Hadoop data directory
- Creating two files named masters and slaves

Step 5: Configure NameNode and DataNode Applicable to datanodes

- Change in hdfs-site.xml
- Creating Hadoop data directory

Step 6: Start the cluster

- Start the cluster

- Monitor cluster health by going to ec2-{instance-address}.amazonaws.com:50070
  - Showing all three datanodes running properly

  Video Link: https://www.youtube.com/watch?v=Z_5pJMCoeM8

Deliverables

- **What to submit:** You need to submit screenshots for some configuration files, screenshots for testing some simple Hadoop command, and screenshot showing Teragen, Terasort is running to show that you successfully configure the cluster and the benchmark runs successfully.
- You **MUST** submit one screenshot of your cluster health by going to ec2-{instance-address}.amazonaws.com:50070 when your cluster is running. See the example below.

Deliverables
1. screenshots for some configuration files

- hdfs-site.xml
- mapred-site.xml
- yarn-site.xml
- etc.

Deliverable
2. screenshots for testing some simple Hadoop command

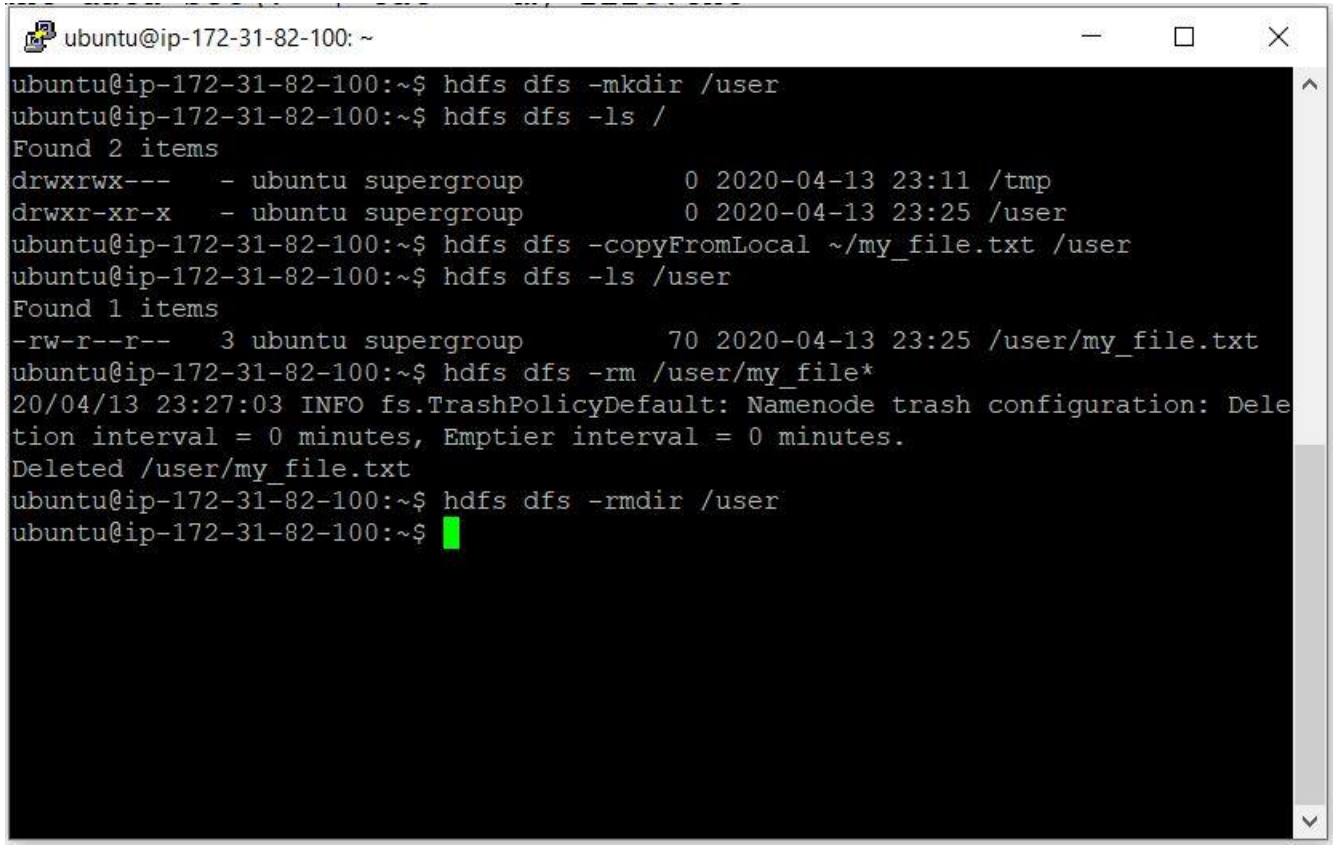
- Some Basic command on hdfs

Such as- creating directory, copy from local to hdfs, deleting directory, showing java process (using JPS), etc.

Video Link: https://www.youtube.com/watch?v=X8Tn3FETqXw

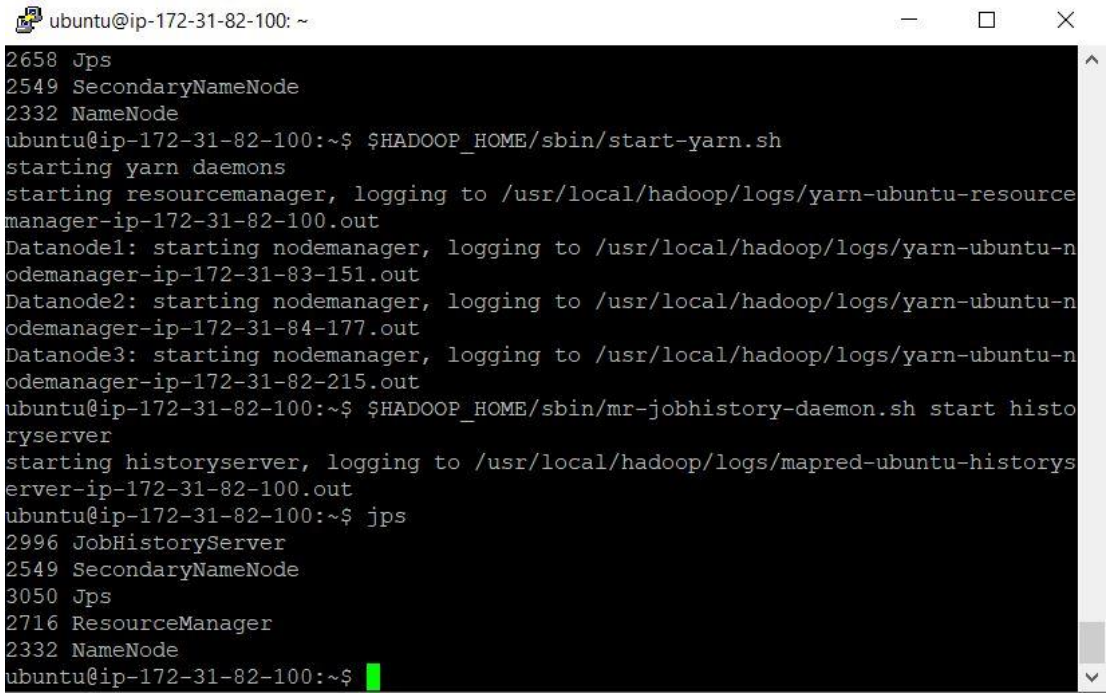
Deliverable
2. screenshots for testing some simple Hadoop command

deleting directory

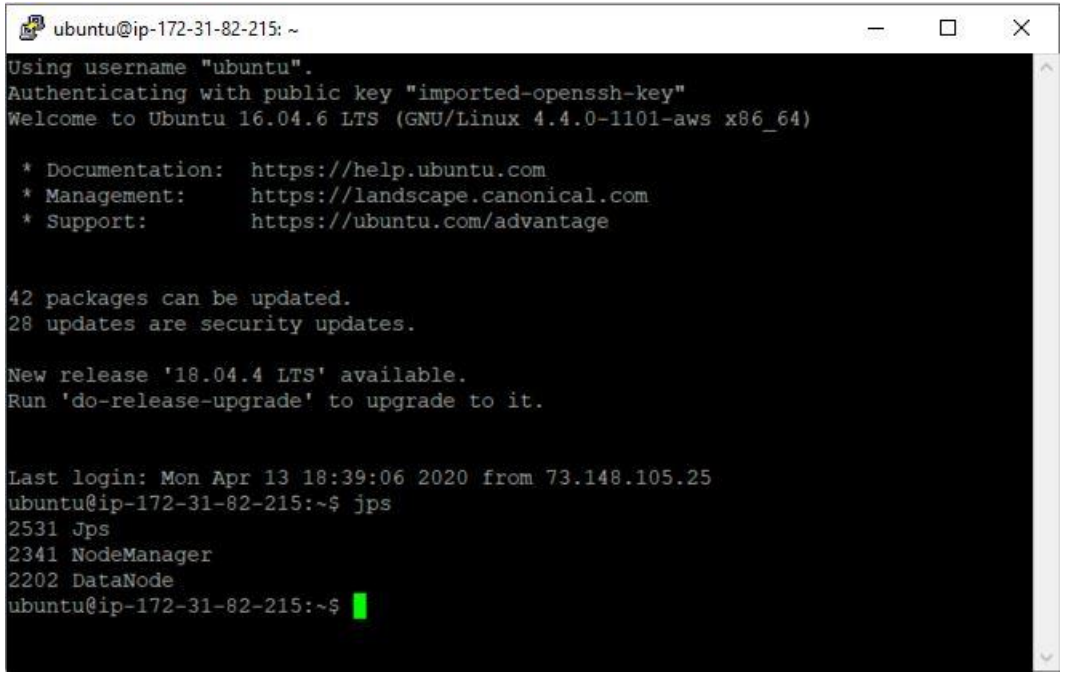


Deliverable
2. screenshots for testing some simple Hadoop command

Running JPS command for namenode



Running JPS command for datanode



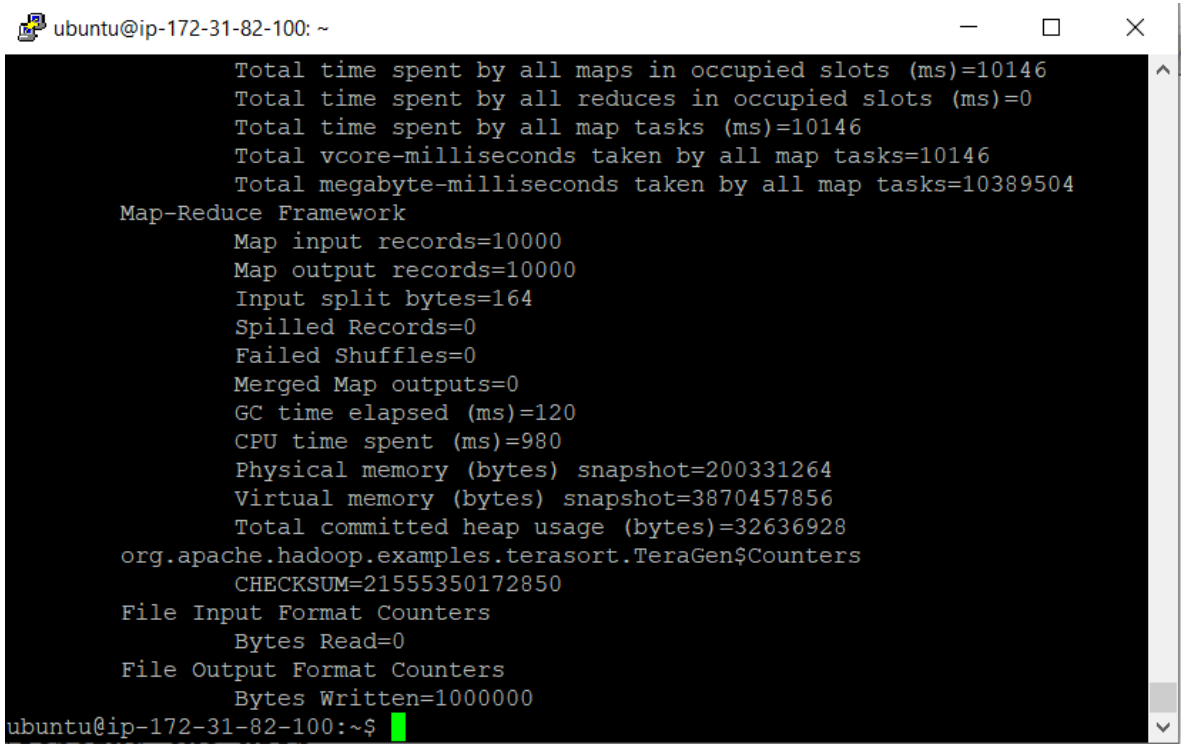
Deliverable 3: screenshot showing Teragen, Terasort is running

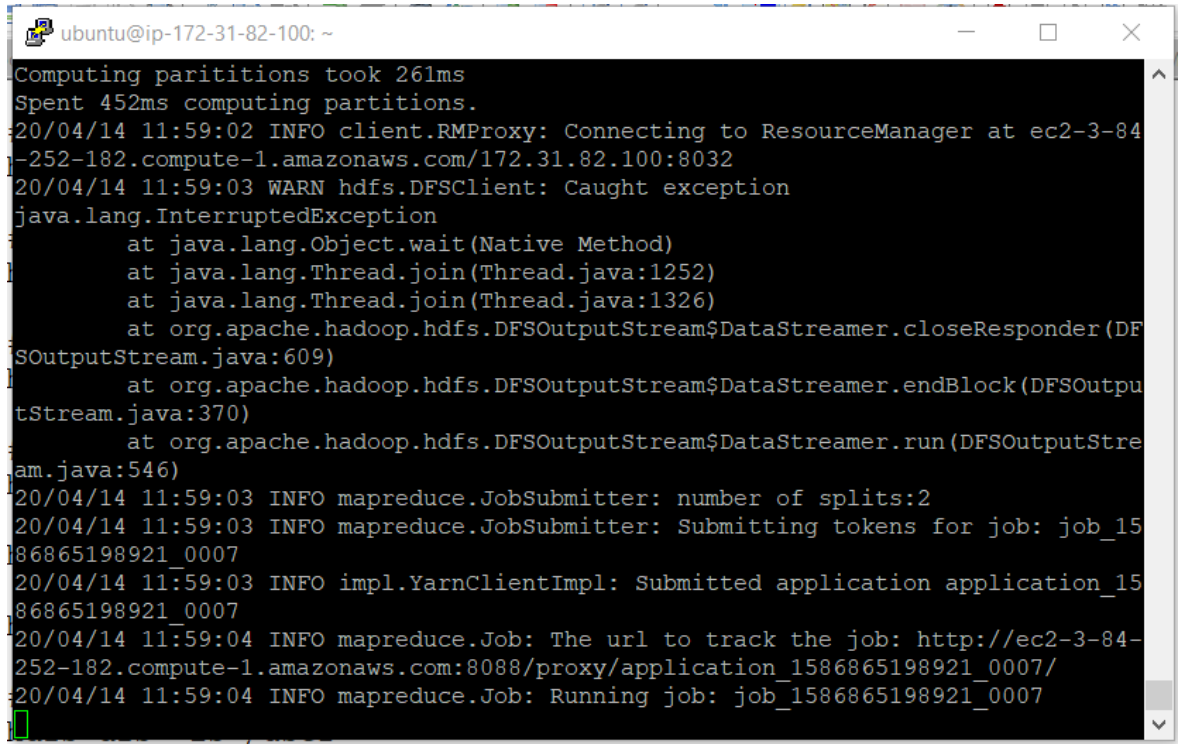
- Run Hadoop TeraGen and TeraSort benchmarks

- hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-*examples*.jar teragen 10000000 /terasort-input
- --generate unsorted data and store the data in /terasort-input

- hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-*examples*.jar terasort /terasort-input /terasort-output
- --sort the generated unsorted data and store the data in /terasort-output

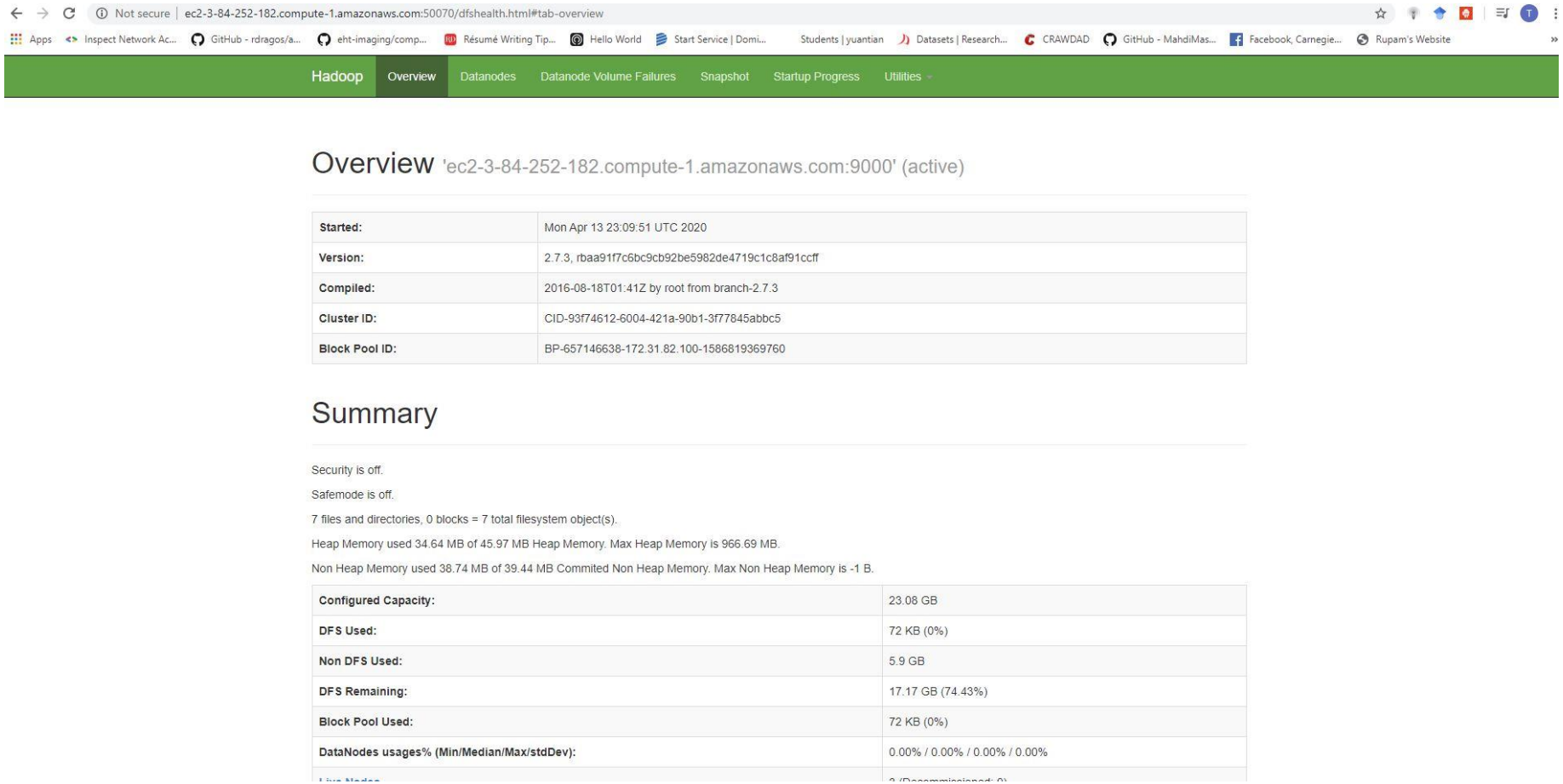
Deliverable
3: screenshot showing Teragen, Terasort is running





Deliverable

- You **MUST** submit one screenshot of your cluster health by going to ec2-{instance-address}.amazonaws.com:50070 when your cluster is running. See the example below.



Questions?