Student: Love Shah
UCID: Lks9
Course: CS 643 852
Instructor: Manoop Talasila
Date: 27 April, 2022
Subject: Cloud Computing

New Jersey Institute of Technology

Module 07 Assignment 01:
Programming Assignment 2
Programming Language: Java

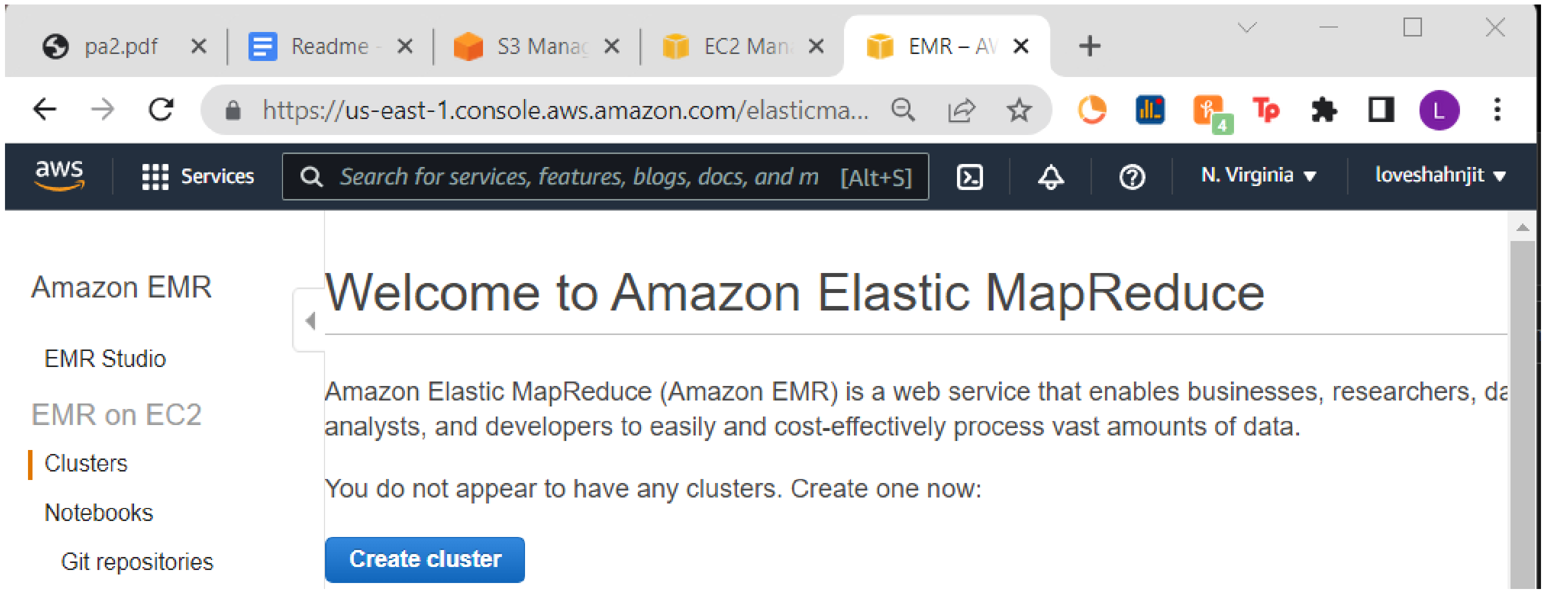
Background:

- The goal of this programming assignment is to (1) use Apache Spark to train a Machine Learning model in parallel on multiple EC2 instances and (2) use Spark's MLib to develop and use a Machine Learning model in the cloud and (3) how to use docker to create a container for your machine learning model to simplify model deployment.
- Implementation: build a wine quality prediction ML model in Spark over AWS.
- Model is trained in parallel using 4 EC2 instances. Then saved and loaded into a Spark application that will perform wine quality prediction.
  - (1) This application will run on one Ec2 instance.
  - (2) Assignment must be implemented in Java on Ubuntu Linux.
- Input for model training: 2 datasets given for your ML model
  - TrainingDataset.csv: Use this dataset to train the model in parallel on multiple EC2 instances.
  - ValidationDataset.csv: Use this dataset to validate the model and optimize its performance (i.e., select the best values for the model parameters). •
- Input for prediction testing: TestDataset.csv.
  - We will use this file, which has a similar structure with the two datasets above, to test the functionality and performance of your prediction application.
  - Your prediction application should take such a file as input.
    This file is not shared with you, but you can use the validation dataset to make sure your application works.
  - Output: The output of your application will be a measure of the prediction performance, specifically the F1 score, which is available in MLlib. •
  - Model Implementation: You have to develop a Spark application that uses MLlib to train for wine quality prediction using the training dataset. You will use the validation dataset to check the performance of your trained model and to potentially tune your ML model parameters for best performance. You should start with a simple linear regression or logistic regression model from MLlib, but you can try multiple ML models to see which one leads to better performance. For classification models, you can use 10 classes (the wine scores are from 1 to 10).
- A Docker container will be created for the Docker application so a prediction model can be quickly deployed across multiple different environments.
- The model training is done in parallel on 4 EC2 instances.
- The prediction with or without Docker is done on a single EC2 instance.

Overview:

- The application is automatically parallelized. Spark DataFrames, along with MLib, are used to make this application which runs on an Amazon Web Services EMR cluster. The task execution is taken care of, and HDFS is used for all file inquiries (locating files, storing training models.

Step 1: Creating an EMR cluster

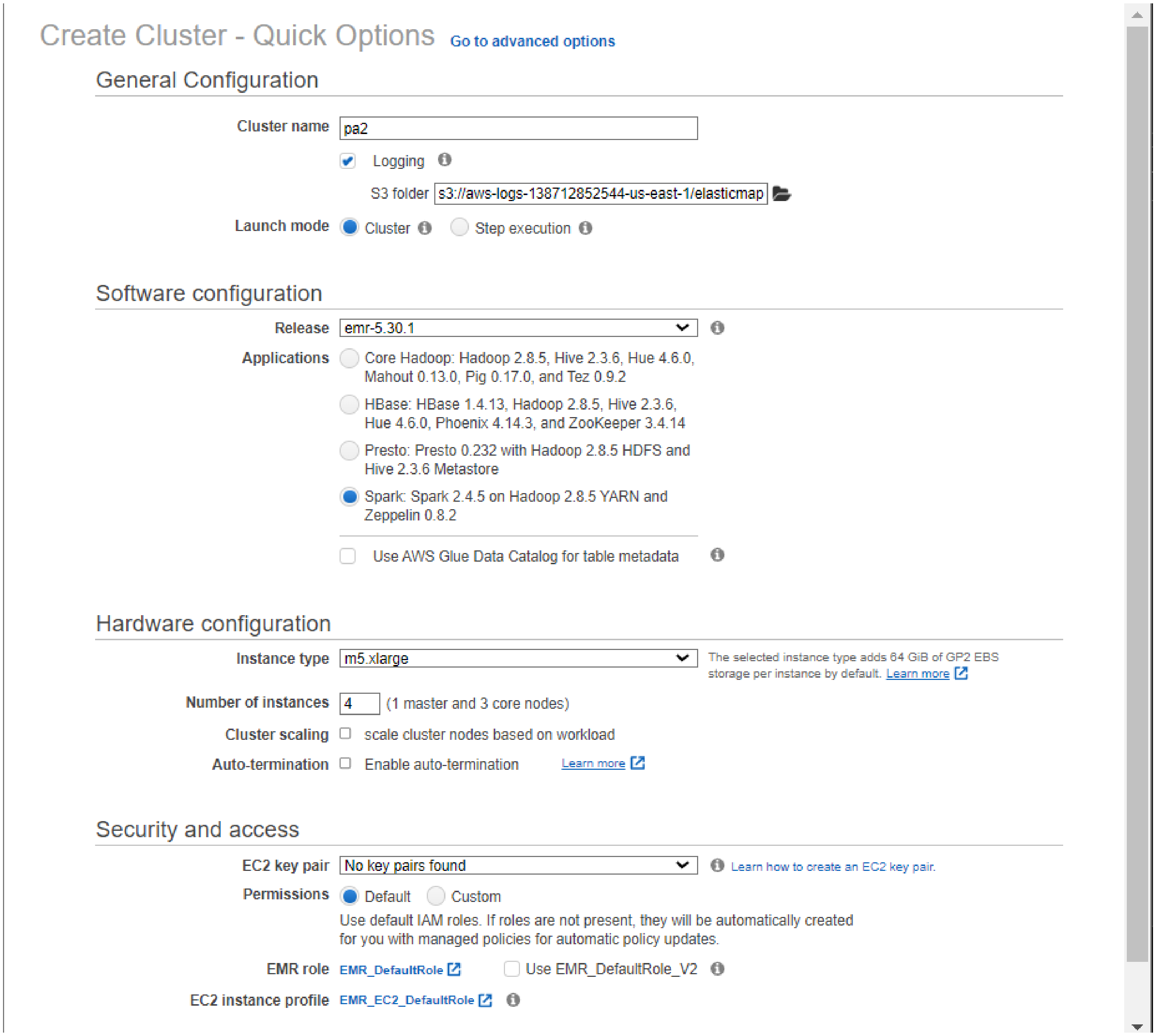


Welcome to Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, da
analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now:

Log into AWS console => Click EMR Service => Click Create Cluster => Launch Cluster => Set configurations such as launch mode, vendor, release (emr-5.30.1), spark version (2.4.5), hardware configurations (instance type and number of instances). We will be using 4 instances where 1 is a master and 3 are slaves.



Create Cluster - Quick Options Go to advanced options

General Configuration

Cluster name: pa2

[✓] Logging

S3 folder: s3://aws-logs-138712852544-us-east-1/elasticmap

Launch mode: [✓ Cluster] Step execution

Software configuration

Release: emr-5.30.1

Applications:
- Core Hadoop: Hadoop 2.8.5, Hive 2.3.6, Hue 4.6.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- HBase: HBase 1.4.13, Hadoop 2.8.5, Hive 2.3.6, Hue 4.6.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
- Presto: Presto 0.232 with Hadoop 2.8.5 HDFS and Hive 2.3.6 Metastore
- [✓ Spark: Spark 2.4.5 on Hadoop 2.8.5 YARN and Zeppelin 0.8.2]

☐ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type: m5.xlarge — The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. Learn more

Number of instances: 4 (1 master and 3 core nodes)

Cluster scaling: ☐ scale cluster nodes based on workload

Auto-termination: ☐ Enable auto-termination Learn more

Security and access

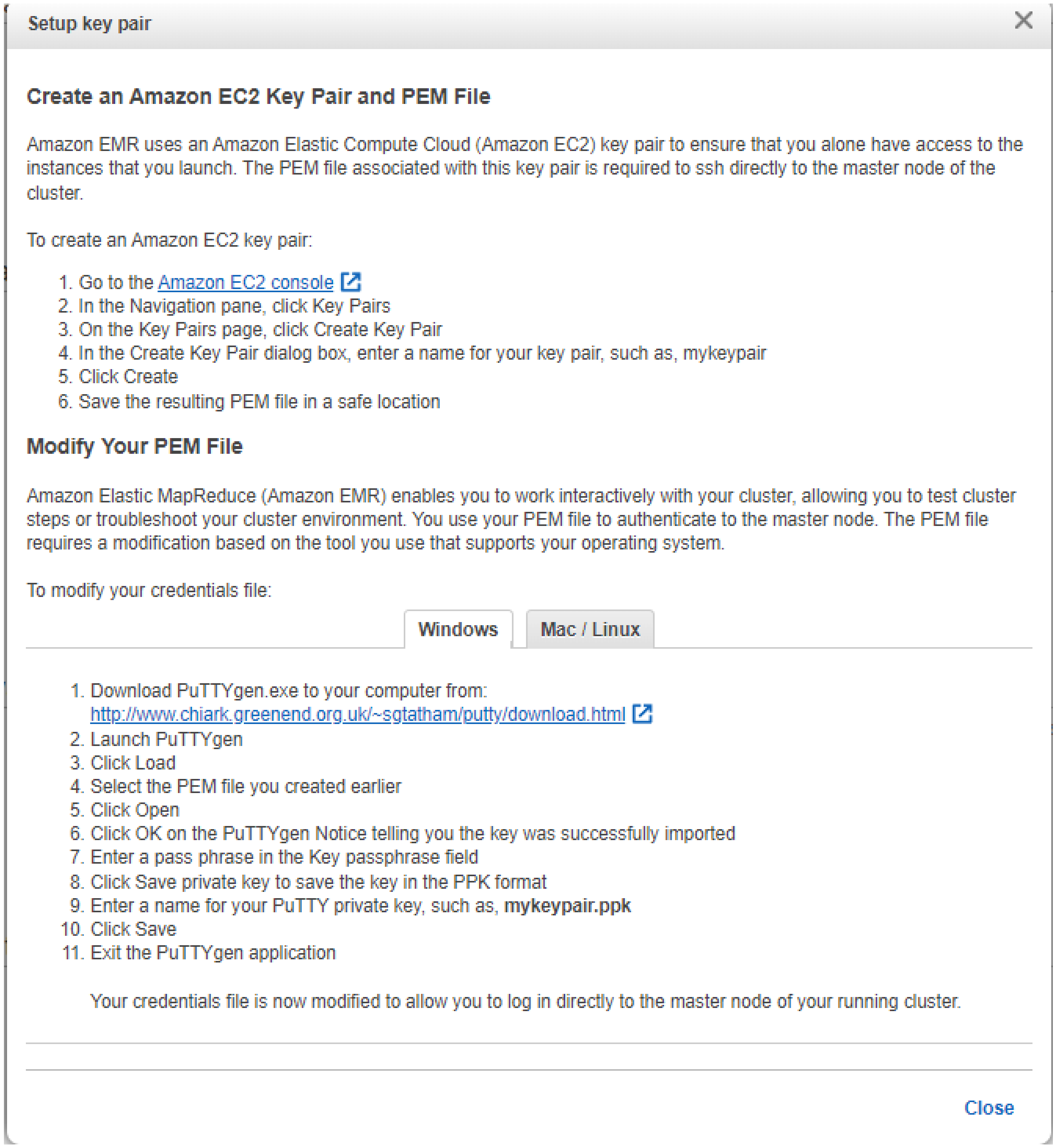
EC2 key pair: No key pairs found — Learn how to create an EC2 key pair.

Permissions: [✓ Default] Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole ☐ Use EMR_DefaultRole_V2

EC2 instance profile: EMR_EC2_DefaultRole



Setup key pair

Create an Amazon EC2 Key Pair and PEM File

Amazon EMR uses an Amazon Elastic Compute Cloud (Amazon EC2) key pair to ensure that you alone have access to the instances that you launch. The PEM file associated with this key pair is required to ssh directly to the master node of the cluster.

To create an Amazon EC2 key pair:

1. Go to the Amazon EC2 console
2. In the Navigation pane, click Key Pairs
3. On the Key Pairs page, click Create Key Pair
4. In the Create Key Pair dialog box, enter a name for your key pair, such as, mykeypair
5. Click Create
6. Save the resulting PEM file in a safe location

Modify Your PEM File

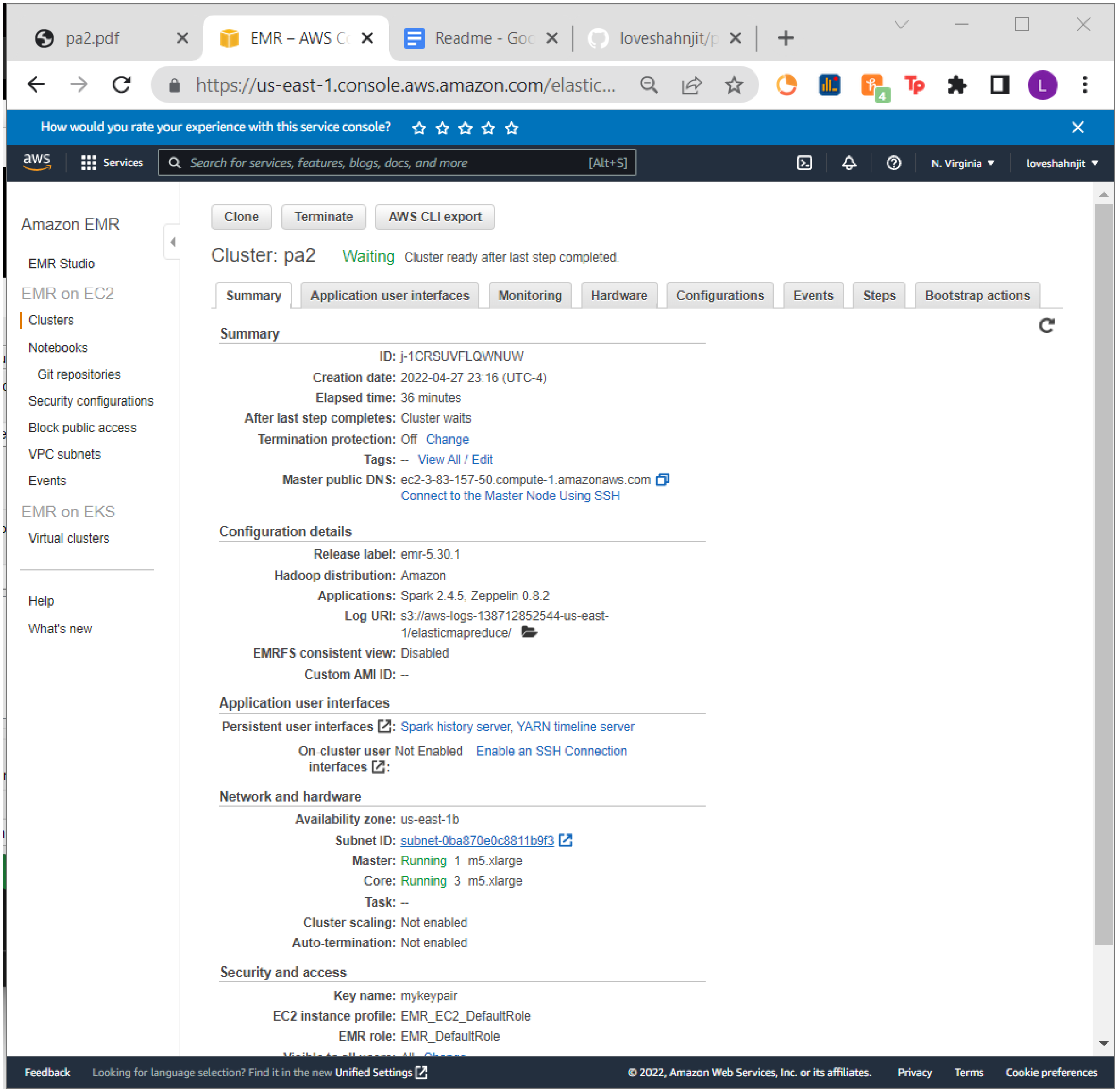
Amazon Elastic MapReduce (Amazon EMR) enables you to work interactively with your cluster, allowing you to test cluster steps or troubleshoot your cluster environment. You use your PEM file to authenticate to the master node. The PEM file requires a modification based on the tool you use that supports your operating system.

To modify your credentials file:

1. Download PuTTYgen.exe to your computer from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Launch PuTTYgen
3. Click Load
4. Select the PEM file you created earlier
5. Click Open
6. Click OK on the PuTTYgen Notice telling you the key was successfully imported
7. Enter a pass phrase in the Key passphrase field
8. Click Save private key to save the key in the PPK format
9. Enter a name for your PuTTY private key, such as, **mykeypair.ppk**
10. Click Save
11. Exit the PuTTYgen application

Your credentials file is now modified to allow you to log in directly to the master node of your running cluster.

Grab the Ec2 key pair if there, if not create one as needed



Cluster: pa2 Waiting Cluster ready after last step completed.

Summary

ID: j-1CRSUVFLQWNUW
Creation date: 2022-04-27 23:16 (UTC-4)
Elapsed time: 36 minutes
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS: ec2-3-83-157-50.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-5.30.1
Hadoop distribution: Amazon
Applications: Spark 2.4.5, Zeppelin 0.8.2
Log URI: s3://aws-logs-138712852544-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

Persistent user interfaces: Spark history server, YARN timeline server
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Network and hardware

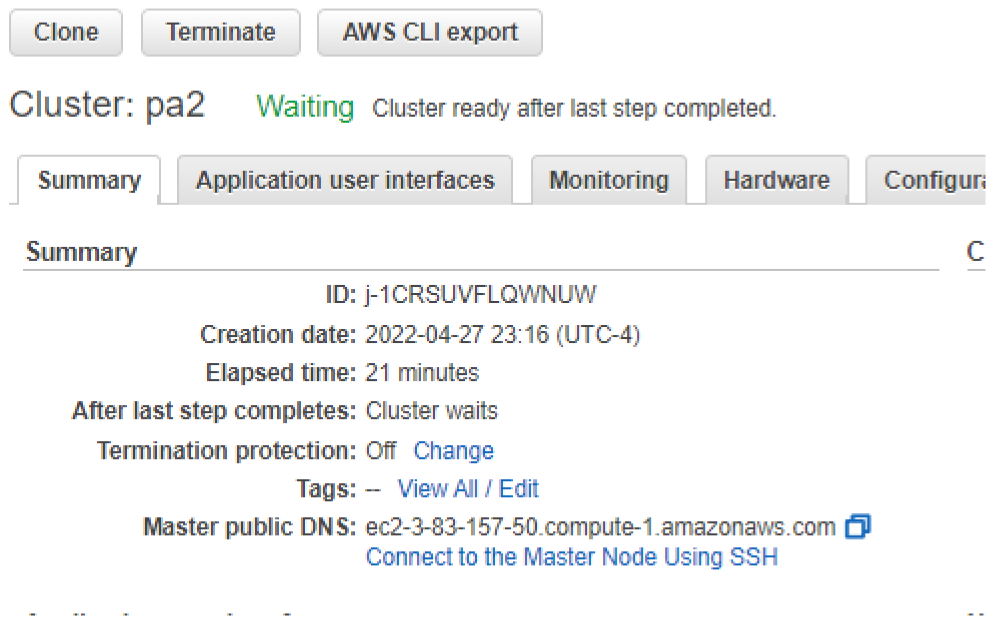
Availability zone: us-east-1b
Subnet ID: subnet-0ba870e0c8811b9f3
Master: Running 1 m5.xlarge
Core: Running 3 m5.xlarge
Task: --
Cluster scaling: Not enabled
Auto-termination: Not enabled

Security and access

Key name: mykeypair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole

Finish with creating the cluster

Step 2: SFTP connection to Master Node for uploads

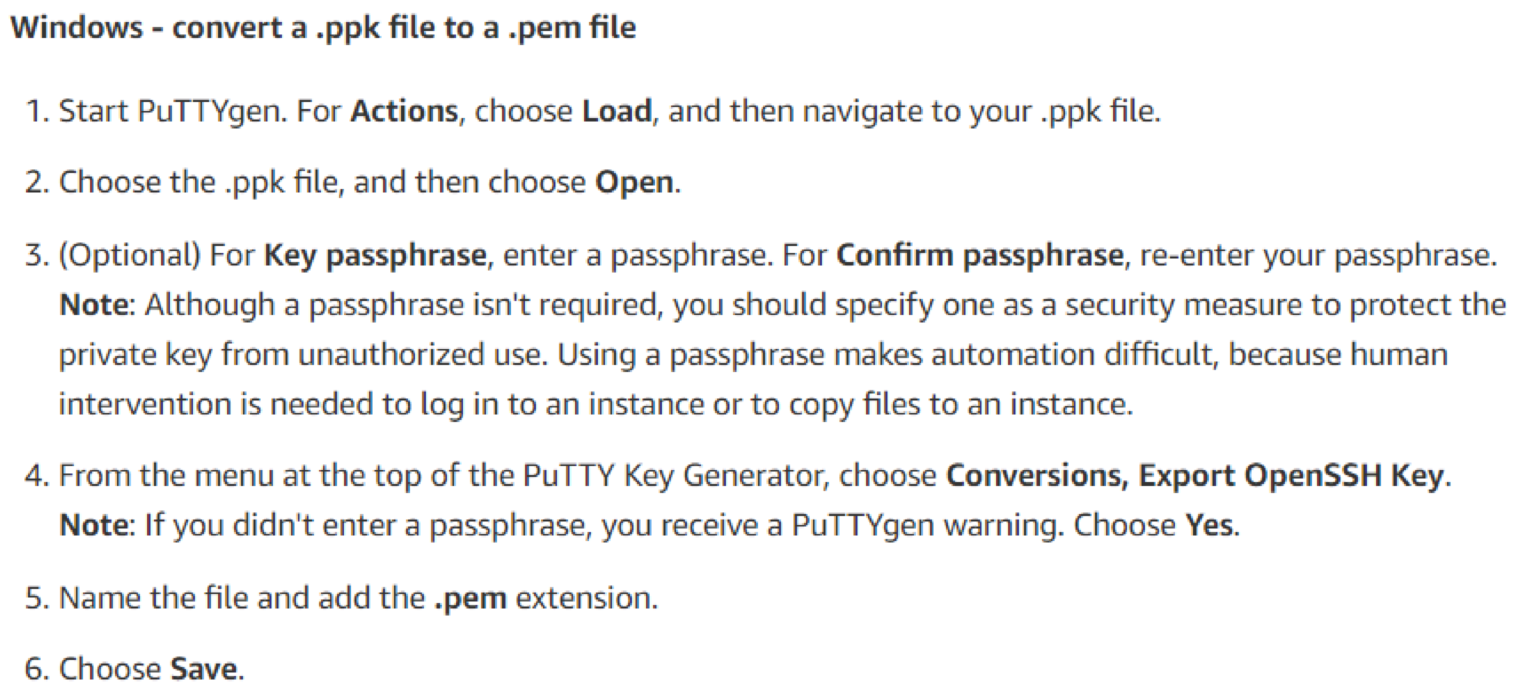


Cluster: pa2 Waiting Cluster ready after last step completed.

Summary

ID: j-1CRSUVFLQWNUW
Creation date: 2022-04-27 23:16 (UTC-4)
Elapsed time: 21 minutes
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS: ec2-3-83-157-50.compute-1.amazonaws.com
Connect to the Master Node Using SSH

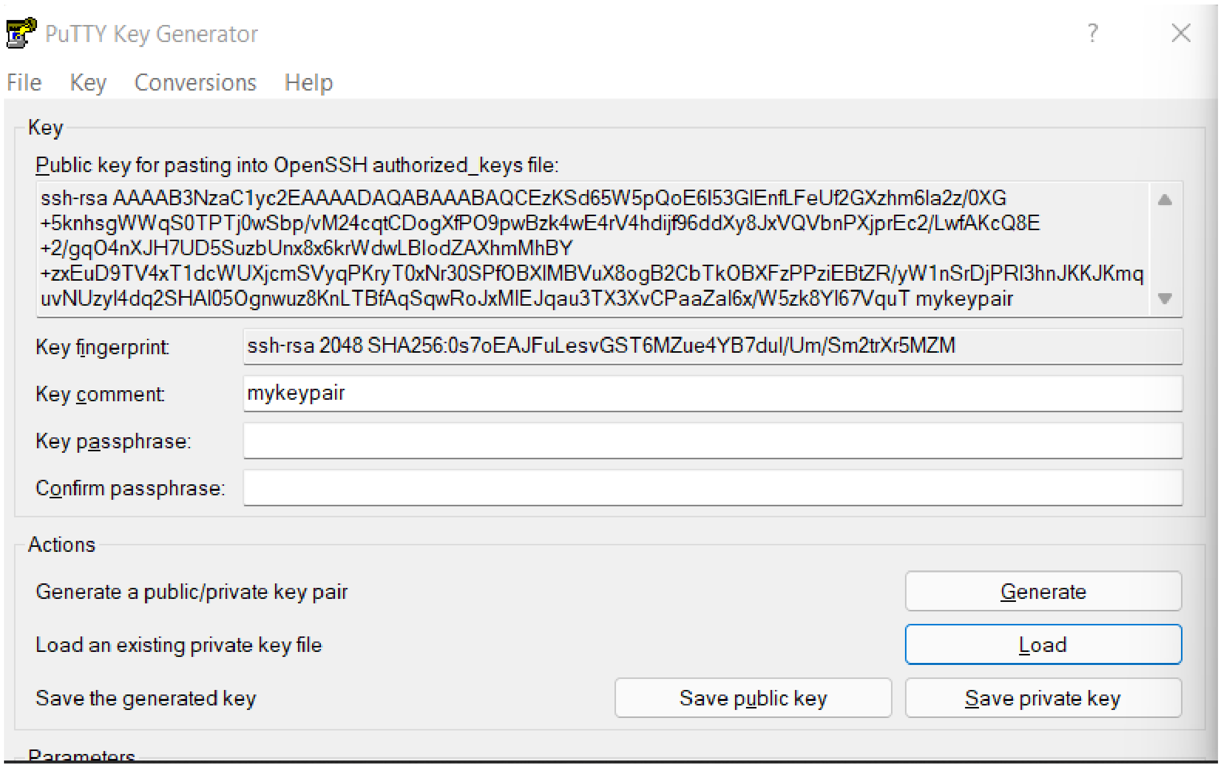
Grab the master node public dns address as shown above, open cmd on a local pc.



Windows - convert a .ppk file to a .pem file

1. Start PuTTYgen. For **Actions**, choose **Load**, and then navigate to your .ppk file.
2. Choose the .ppk file, and then choose **Open**.
3. (Optional) For **Key passphrase**, enter a passphrase. For **Confirm passphrase**, re-enter your passphrase. **Note**: Although a passphrase isn't required, you should specify one as a security measure to protect the private key from unauthorized use. Using a passphrase makes automation difficult, because human intervention is needed to log in to an instance or to copy files to an instance.
4. From the menu at the top of the PuTTY Key Generator, choose **Conversions, Export OpenSSH Key**. **Note**: If you didn't enter a passphrase, you receive a PuTTYgen warning. Choose **Yes**.
5. Name the file and add the **.pem** extension.
6. Choose **Save**.

Optional Step: Your .pem file might be downloaded in .ppk format, in which case it must be converted to .pem to work. The full directions are here: https://aws.amazon.com/premiumsupport/knowledge-center/ec2-ppk-pem-conversion/



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCEzKSd65W5pQoE6l53GlEnfLFeUf2GXzhm6la2z/0XG
+5knhsgWWqS0TPTj0wSbp/vM24cqtCDogXfPO9pwBzk4wE4rV4hdijf96ddXy8JxVQVbnPXjprEc2/LwfAKcQ8E
+2/gqO4nXJH7UD5SuzbUnx8x6krWdwLBlodZAXhmMhBY
+zxEuD9TV4xT1dcWUXjcmSVyqPKryT0xNr30SPfOBXlMBVuX8ogB2CbTkOBXFzPPziEBtZR/yW1nSrDjPRl3hnJKKJKmq
uvNUzyl4dq2SHAl05Ognwuz8KnLTBfAqSqwRoJxMlEJqau3TX3XvCPaaZal6x/W5zk8Yl67VquT mykeypair
```

Key fingerprint:	ssh-rsa 2048 SHA256:0s7oEAJFuLesvGST6MZue4YB7dul/Um/Sm2trXr5MZM
Key comment:	mykeypair
Key passphrase:	
Confirm passphrase:	

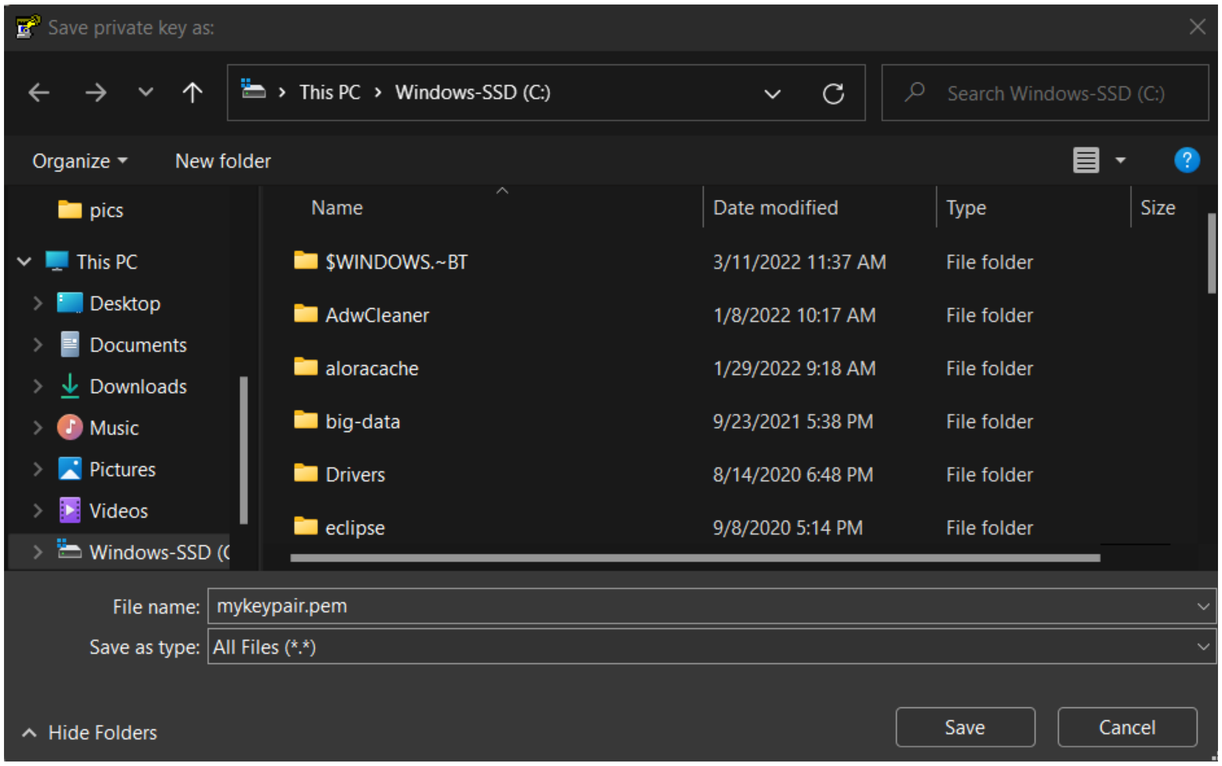
Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Parameters



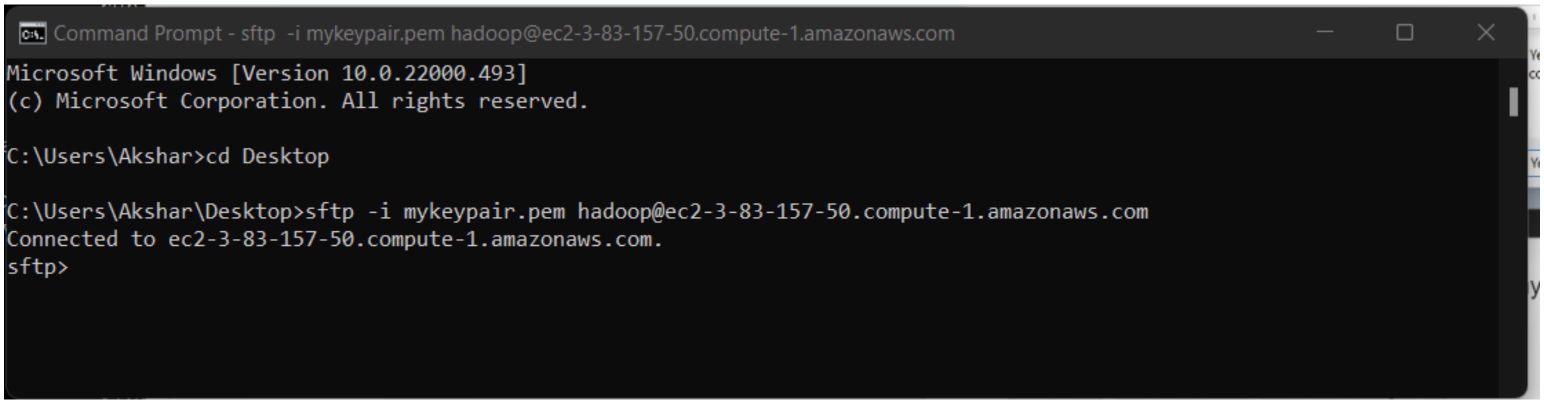
Name	Date modified	Type	Size
$WINDOWS.~BT	3/11/2022 11:37 AM	File folder	
AdwCleaner	1/8/2022 10:17 AM	File folder	
aloracache	1/29/2022 9:18 AM	File folder	
big-data	9/23/2021 5:38 PM	File folder	
Drivers	8/14/2020 6:48 PM	File folder	
eclipse	9/8/2020 5:14 PM	File folder	

File name: mykeypair.pem

Save as type: All Files (*.*)

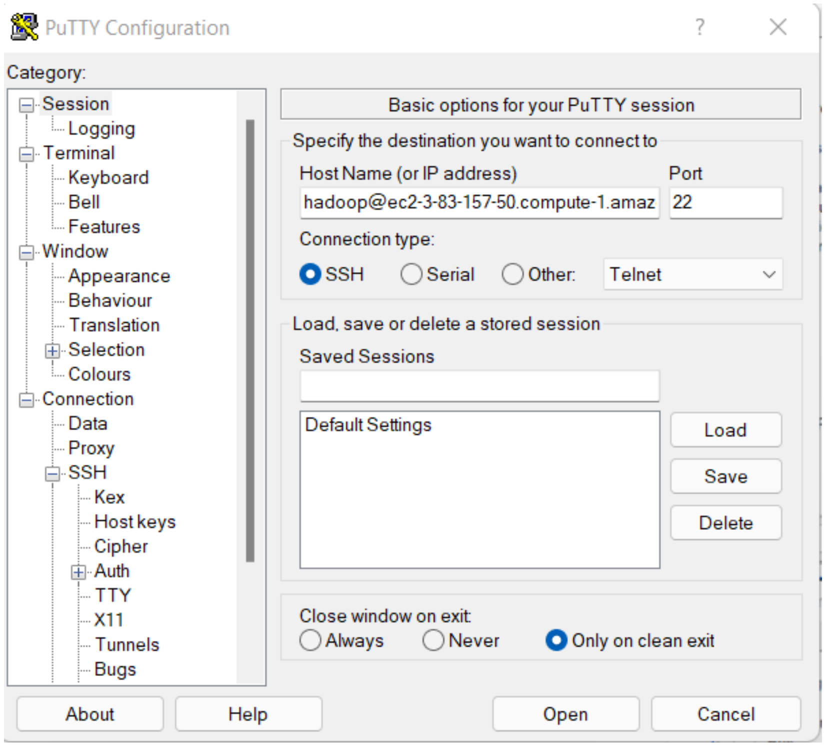
Shown above is using PuttyGen to convert the .ppk file that AWS gives to a .pem file to proceed.

Enter the following command (shown below) to authenticate yourself to your cluster with the pem key created when generating an ec2 key pair. The format for the master node dns address is hadoop@"master node dns address goes here".
My master node address is "***ec2-3-83-157-50.compute-1.amazonaws.com*** ".

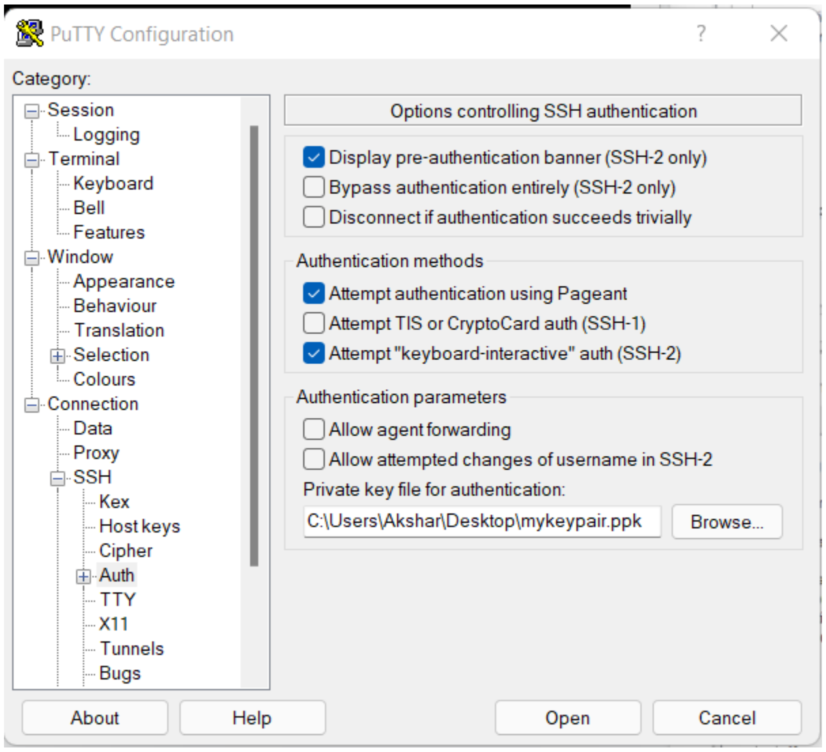

Command Prompt - sftp -i mykeypair.pem hadoop@ec2-3-83-157-50.compute-1.amazonaws.com

```
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Akshar>cd Desktop

C:\Users\Akshar\Desktop>sftp -i mykeypair.pem hadoop@ec2-3-83-157-50.compute-1.amazonaws.com
Connected to ec2-3-83-157-50.compute-1.amazonaws.com.
sftp>
```

$sftp -i mykeypair.pem hadoop@ec2-3-83-157-50.compute-1.amazonaws.com



Specify the destination you want to connect to
Host Name (or IP address): hadoop@ec2-3-83-157-50.compute-1.amaz
Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Load, save or delete a stored session
Saved Sessions:
Default Settings
Close window on exit: Always Never [✓ Only on clean exit]



Options controlling SSH authentication

- [✓] Display pre-authentication banner (SSH-2 only)
- [ ] Bypass authentication entirely (SSH-2 only)
- [ ] Disconnect if authentication succeeds trivially

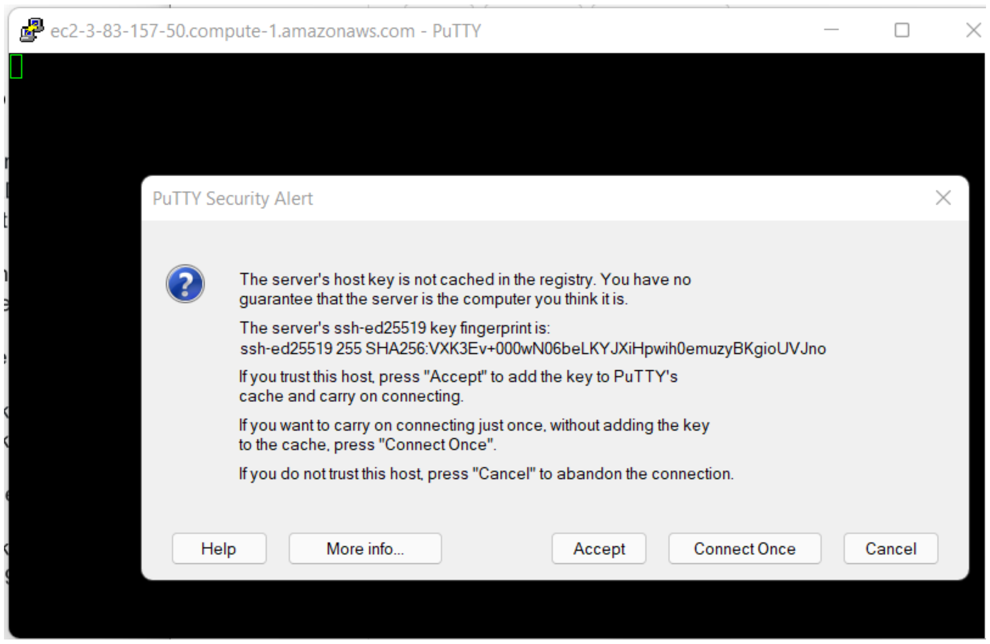
Authentication methods

- [✓] Attempt authentication using Pageant
- [ ] Attempt TIS or CryptoCard auth (SSH-1)
- [✓] Attempt "keyboard-interactive" auth (SSH-2)

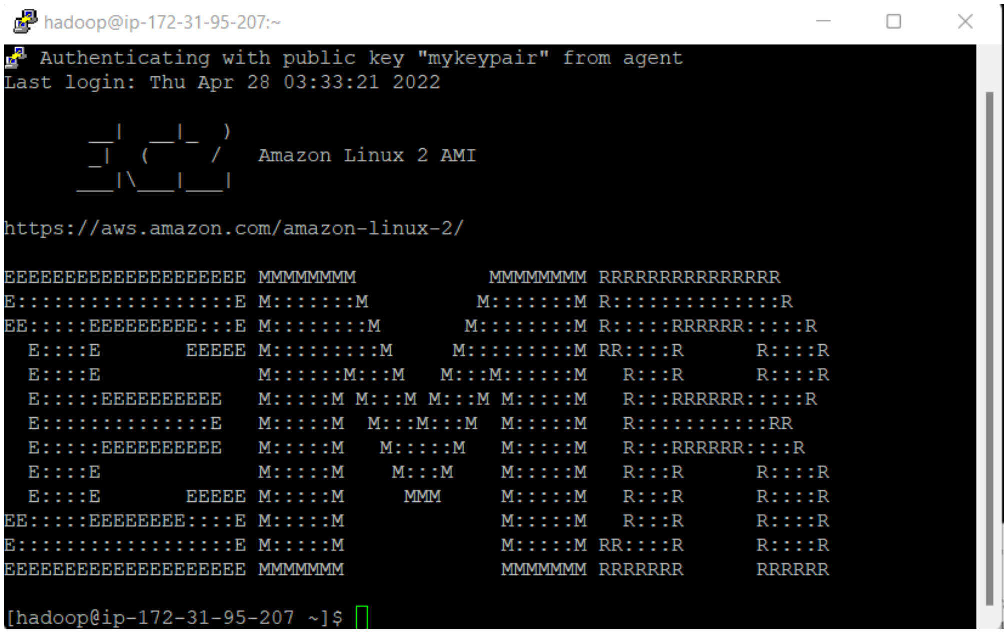
Authentication parameters

- [ ] Allow agent forwarding
- [ ] Allow attempted changes of username in SSH-2

Private key file for authentication:
C:\Users\Akshar\Desktop\mykeypair.ppk

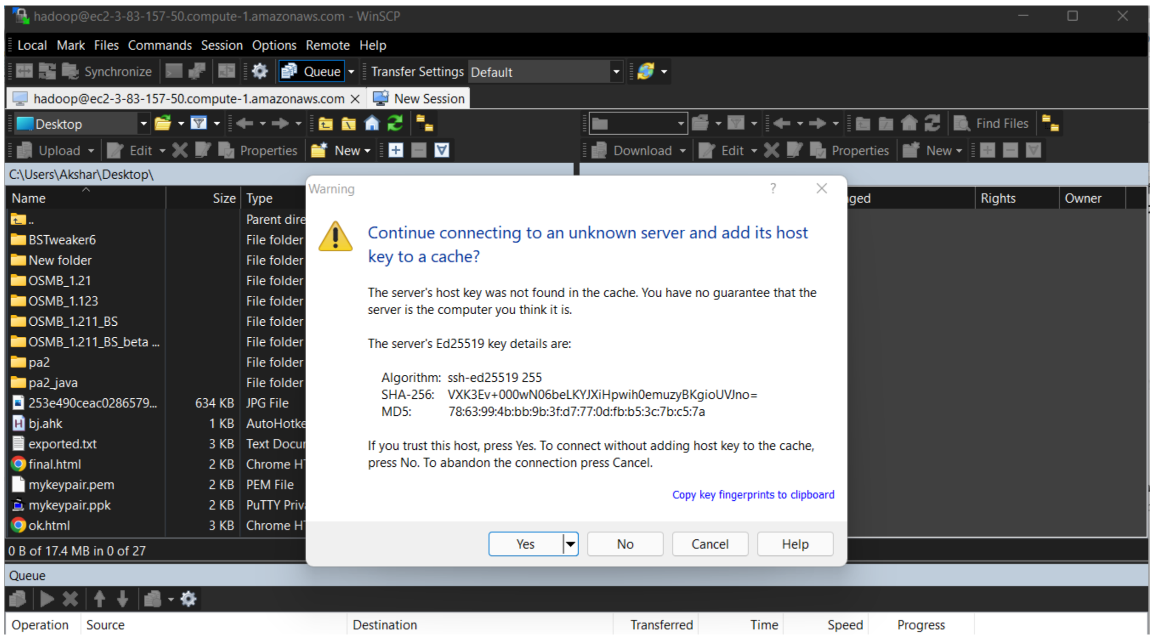


The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:VXK3Ev+000wN06beLKYJXiHpwih0emuzyBKgioUVJno
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".
If you do not trust this host, press "Cancel" to abandon the connection.



```
Authenticating with public key "mykeypair" from agent
Last login: Thu Apr 28 03:33:21 2022

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-95-207 ~]$
```

Alternatively, you can use Putty as shown above (with .ppk file authentication and accepting prompt).



Warning

Continue connecting to an unknown server and add its host key to a cache?

The server's host key was not found in the cache. You have no guarantee that the server is the computer you think it is.

The server's Ed25519 key details are:

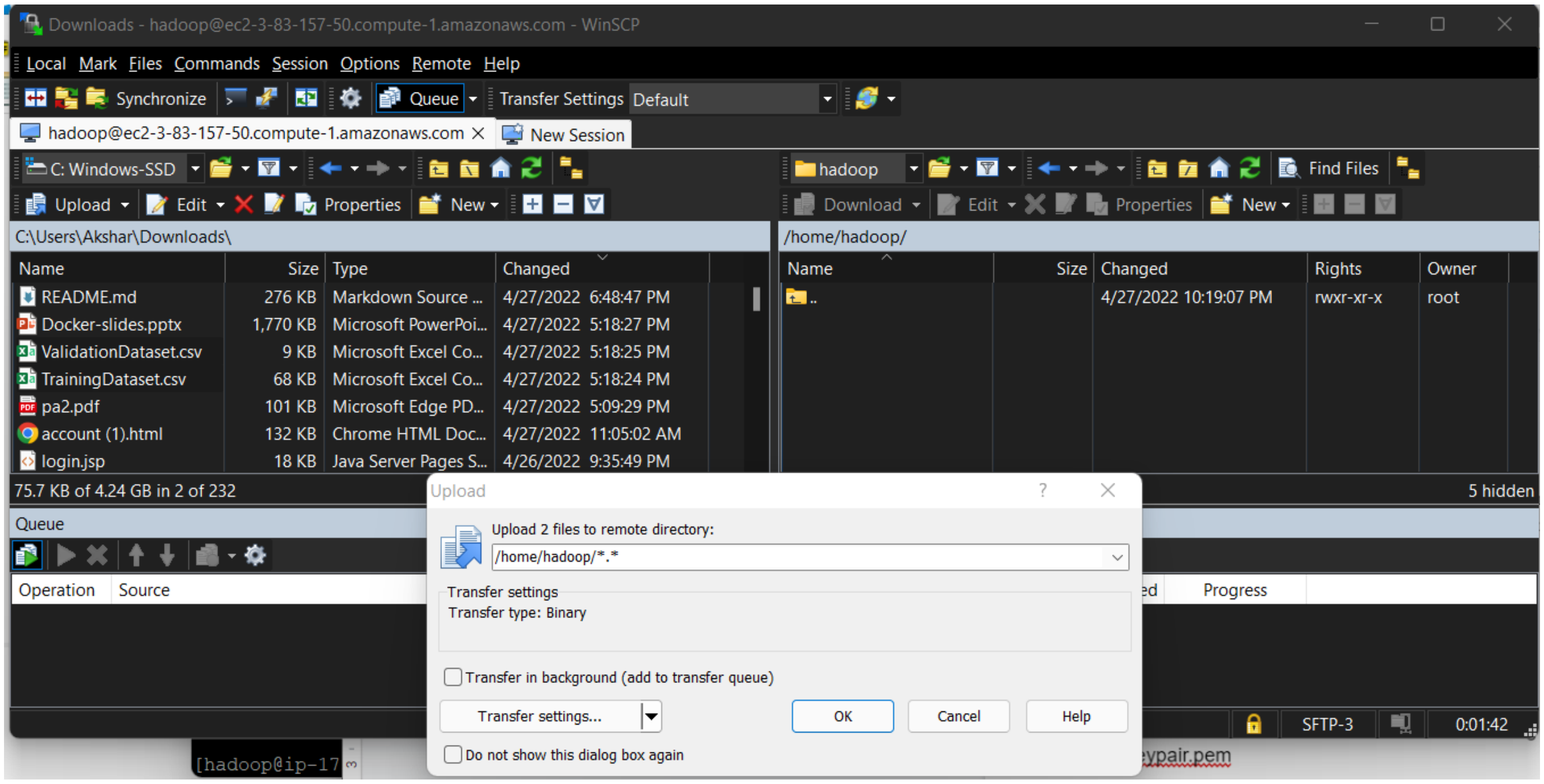
```
Algorithm:  ssh-ed25519 255
SHA-256:    VXK3Ev+000wN06beLKYJXiHpwih0emuzyBKgioUVJno=
MD5:        78:63:99:4b:bb:9b:3f:d7:77:0d:fb:b5:3c:7b:c5:7a
```

If you trust this host, press Yes. To connect without adding host key to the cache, press No. To abandon the connection press Cancel.

Copy key fingerprints to clipboard

Name	Size	Type
..		Parent dire
BSTweaker6		File folder
New folder		File folder
OSMB_1.21		File folder
OSMB_1.123		File folder
OSMB_1.211_BS		File folder
OSMB_1.211_BS_beta ...		File folder
pa2		File folder
pa2_java		File folder
253e490ceac0286579...	634 KB	JPG File
bj.ahk	1 KB	AutoHotke
exported.txt	3 KB	Text Docu
final.html	2 KB	Chrome H
mykeypair.pem	2 KB	PEM File
mykeypair.ppk	2 KB	PuTTY Priv
ok.html	3 KB	Chrome H

Alternatively, you can also use WinSCP as shown above (Easiest in my opinion)



Downloads - hadoop@ec2-3-83-157-50.compute-1.amazonaws.com - WinSCP

hadoop@ec2-3-83-157-50.compute-1.amazonaws.com

C:\Users\Akshar\Downloads\

Name	Size	Type	Changed
README.md	276 KB	Markdown Source ...	4/27/2022 6:48:47 PM
Docker-slides.pptx	1,770 KB	Microsoft PowerPoi...	4/27/2022 5:18:27 PM
ValidationDataset.csv	9 KB	Microsoft Excel Co...	4/27/2022 5:18:25 PM
TrainingDataset.csv	68 KB	Microsoft Excel Co...	4/27/2022 5:18:24 PM
pa2.pdf	101 KB	Microsoft Edge PD...	4/27/2022 5:09:29 PM
account (1).html	132 KB	Chrome HTML Doc...	4/27/2022 11:05:02 AM
login.jsp	18 KB	Java Server Pages S...	4/26/2022 9:35:49 PM

75.7 KB of 4.24 GB in 2 of 232

/home/hadoop/

Name	Size	Changed	Rights	Owner
..		4/27/2022 10:19:07 PM	rwxr-xr-x	root

5 hidden

Upload

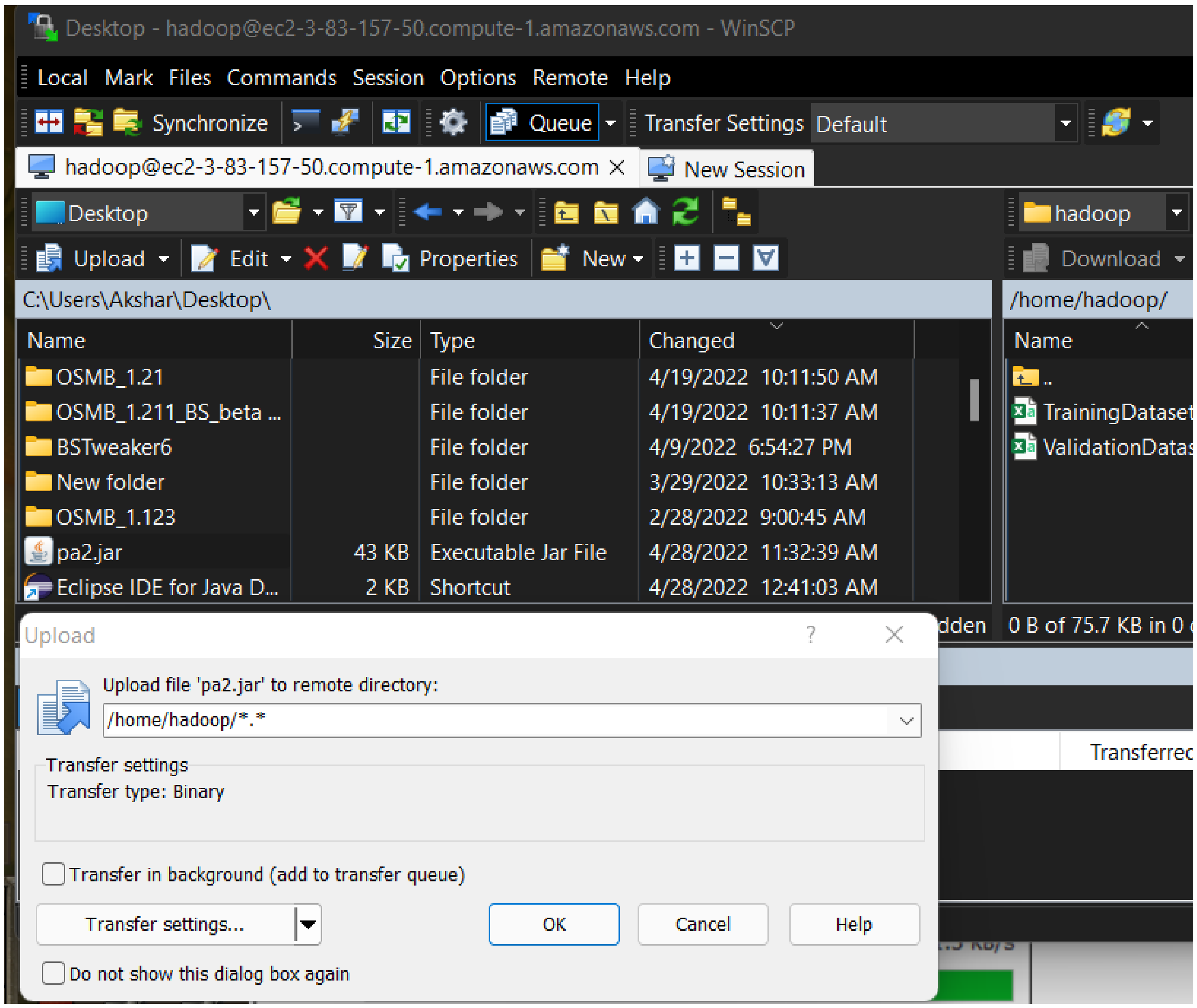
Upload 2 files to remote directory:
/home/hadoop/*.*

Transfer settings
Transfer type: Binary

☐ Transfer in background (add to transfer queue)

☐ Do not show this dialog box again

SFTP-3 0:01:42



Desktop - hadoop@ec2-3-83-157-50.compute-1.amazonaws.com - WinSCP

hadoop@ec2-3-83-157-50.compute-1.amazonaws.com

C:\Users\Akshar\Desktop\

Name	Size	Type	Changed
OSMB_1.21		File folder	4/19/2022 10:11:50 AM
OSMB_1.211_BS_beta ...		File folder	4/19/2022 10:11:37 AM
BSTweaker6		File folder	4/9/2022 6:54:27 PM
New folder		File folder	3/29/2022 10:33:13 AM
OSMB_1.123		File folder	2/28/2022 9:00:45 AM
pa2.jar	43 KB	Executable Jar File	4/28/2022 11:32:39 AM
Eclipse IDE for Java D...	2 KB	Shortcut	4/28/2022 12:41:03 AM

/home/hadoop/

Name
..
TrainingDataset
ValidationDatas

0 B of 75.7 KB in 0

Upload

Upload file 'pa2.jar' to remote directory:
/home/hadoop/*.*

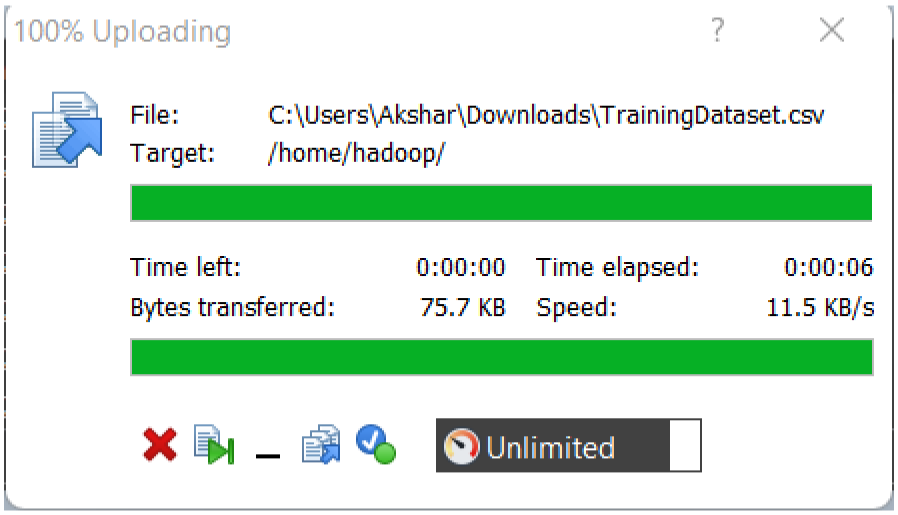
Transfer settings
Transfer type: Binary

☐ Transfer in background (add to transfer queue)

☐ Do not show this dialog box again

Upload the TrainingDataSet.csv, the ValidationDataSet.csv, and the .jar from the github page.

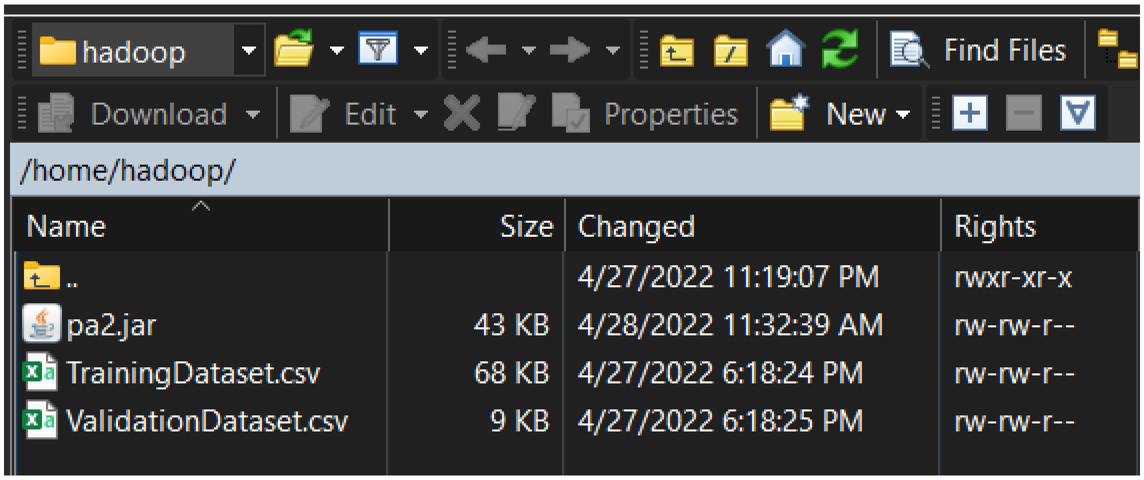
Github page: https://github.com/loveshahnjit/programmingassignment2_java



100% Uploading

```
File:                C:\Users\Akshar\Downloads\TrainingDataset.csv
Target:              /home/hadoop/

Time left:           0:00:00     Time elapsed:     0:00:06
Bytes transferred:   75.7 KB     Speed:            11.5 KB/s
```

If you're using Winscp, or even Putty its very similar to WinScp(shown here):

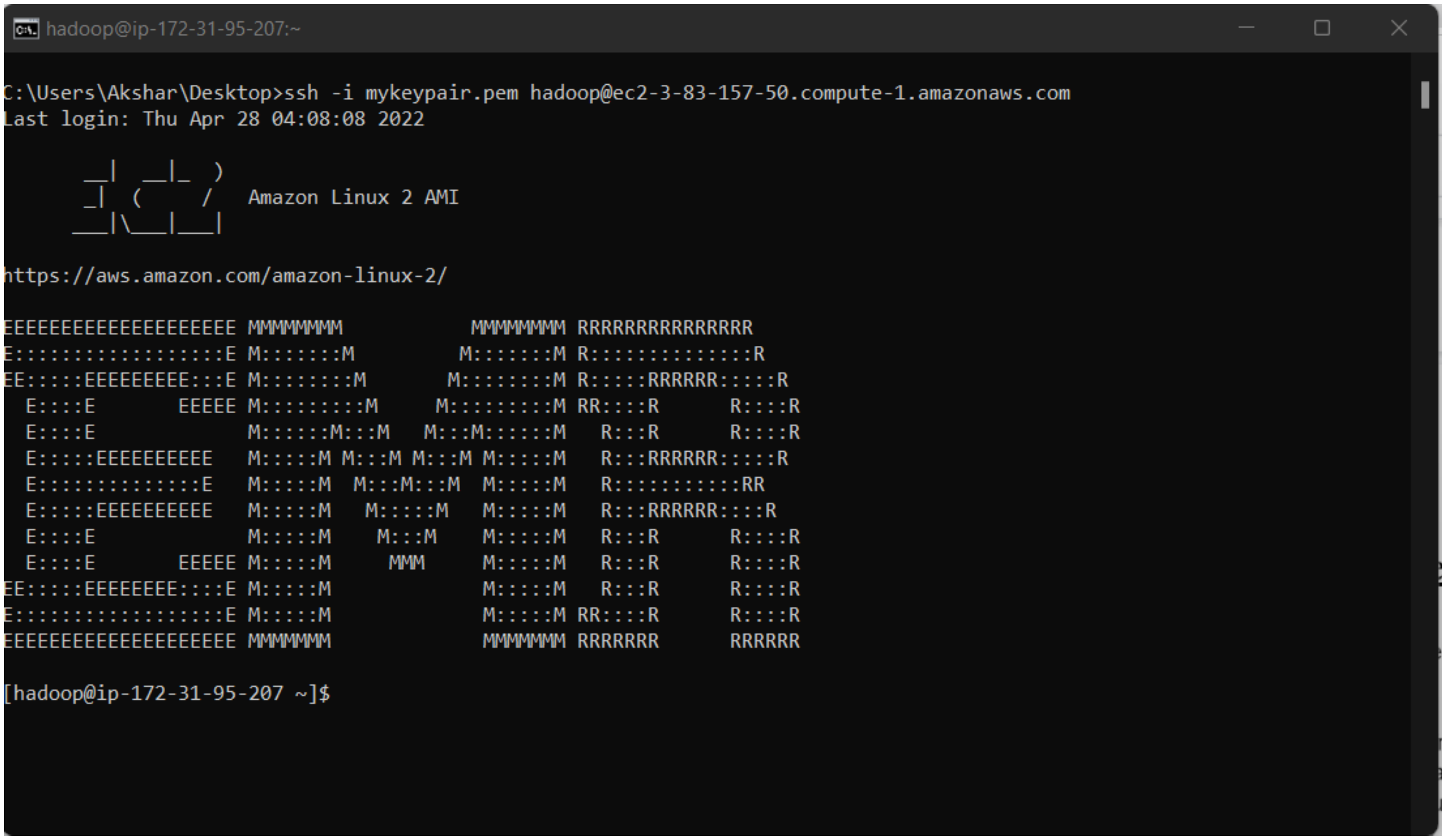


/home/hadoop/

Name	Size	Changed	Rights
..		4/27/2022 11:19:07 PM	rwxr-xr-x
pa2.jar	43 KB	4/28/2022 11:32:39 AM	rw-rw-r--
TrainingDataset.csv	68 KB	4/27/2022 6:18:24 PM	rw-rw-r--
ValidationDataset.csv	9 KB	4/27/2022 6:18:25 PM	rw-rw-r--

Step 3: HDFS file transfer

First we need to ssh into the master node with the following command:

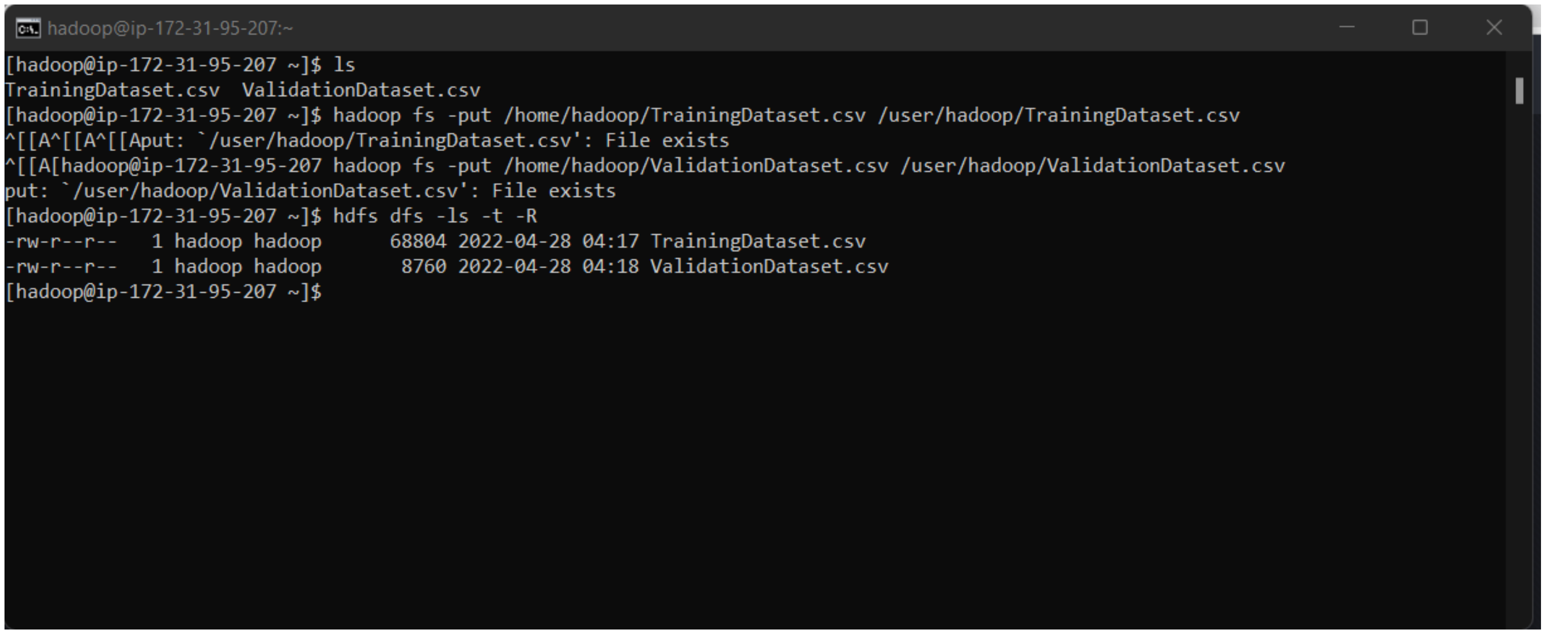


```
C:\Users\Akshar\Desktop>ssh -i mykeypair.pem hadoop@ec2-3-83-157-50.compute-1.amazonaws.com
Last login: Thu Apr 28 04:08:08 2022

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-95-207 ~]$
```

$ssh -i mykeypair.pem hadoop@ec2-3-83-157-50.compute-1.amazonaws.com

NOTE: If it asks you to update, do it! $sudo yum update.

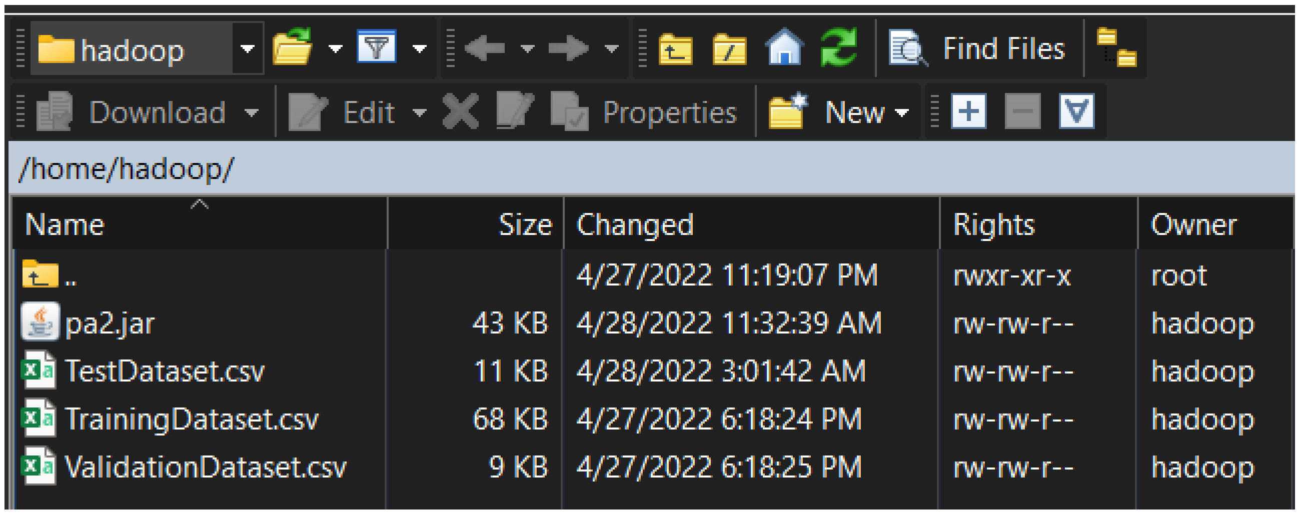
Then we transfer files over to the HDFS from the master node. This is because the slave nodes also must have access to them. Use the following commands shown below to put the files and verify their successful transfer over.



```
[hadoop@ip-172-31-95-207 ~]$ ls
TrainingDataset.csv  ValidationDataset.csv
[hadoop@ip-172-31-95-207 ~]$ hadoop fs -put /home/hadoop/TrainingDataset.csv /user/hadoop/TrainingDataset.csv
^[[A^[[A^[[Aput: `/user/hadoop/TrainingDataset.csv': File exists
^[[A[hadoop@ip-172-31-95-207 hadoop fs -put /home/hadoop/ValidationDataset.csv /user/hadoop/ValidationDataset.csv
put: `/user/hadoop/ValidationDataset.csv': File exists
[hadoop@ip-172-31-95-207 ~]$ hdfs dfs -ls -t -R
-rw-r--r--   1 hadoop hadoop      68804 2022-04-28 04:17 TrainingDataset.csv
-rw-r--r--   1 hadoop hadoop       8760 2022-04-28 04:18 ValidationDataset.csv
[hadoop@ip-172-31-95-207 ~]$
```

1. $ls
2. $hadoop fs -put /home/hadoop/TrainingDataset.csv /user/hadoop/TrainingDataset.csv
3. $hadoop fs -put /home/hadoop/ValidationDataset.csv /user/hadoop/ValidationDataset.csv
4. $hdfs dfs -ls -t -R

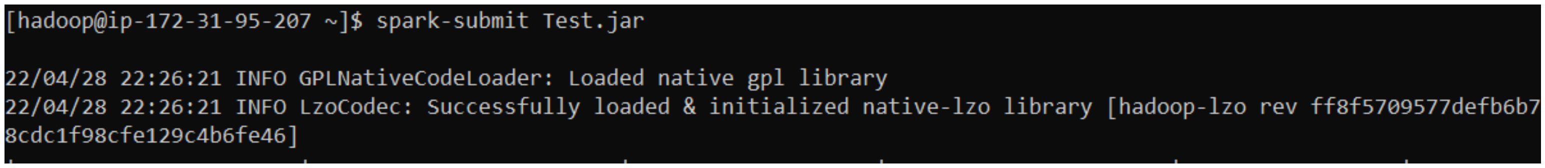
Step 4: Launch the application



/home/hadoop/

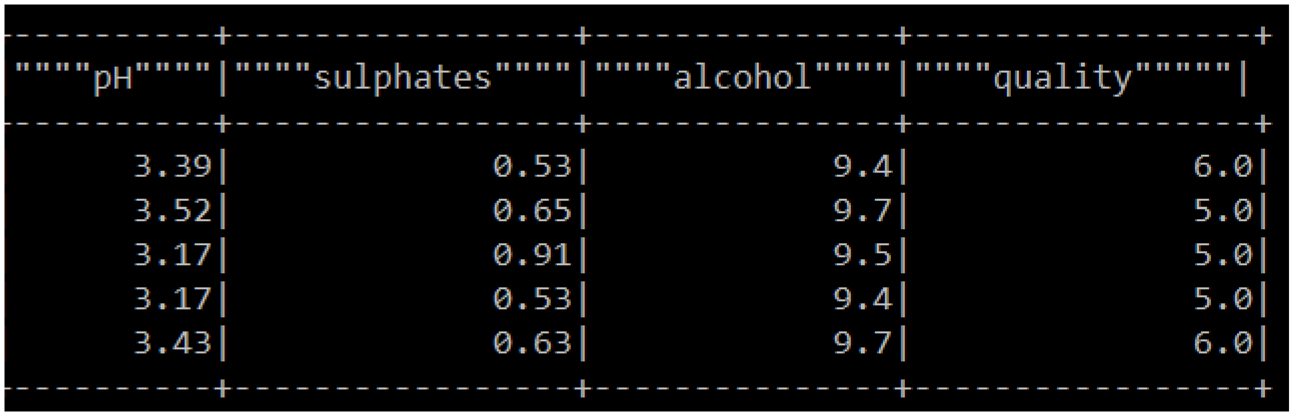
Name	Size	Changed	Rights	Owner
..		4/27/2022 11:19:07 PM	rwxr-xr-x	root
pa2.jar	43 KB	4/28/2022 11:32:39 AM	rw-rw-r--	hadoop
TestDataset.csv	11 KB	4/28/2022 3:01:42 AM	rw-rw-r--	hadoop
TrainingDataset.csv	68 KB	4/27/2022 6:18:24 PM	rw-rw-r--	hadoop
ValidationDataset.csv	9 KB	4/27/2022 6:18:25 PM	rw-rw-r--	hadoop

Launch the Apache-Spark application on the EMR cluster by executing the following command:



```
[hadoop@ip-172-31-95-207 ~]$ spark-submit Test.jar

22/04/28 22:26:21 INFO GPLNativeCodeLoader: Loaded native gpl library
22/04/28 22:26:21 INFO LzoCodec: Successfully loaded & initialized native-lzo library [hadoop-lzo rev ff8f5709577defb6b7
8cdc1f98cfe129c4b6fe46]
```

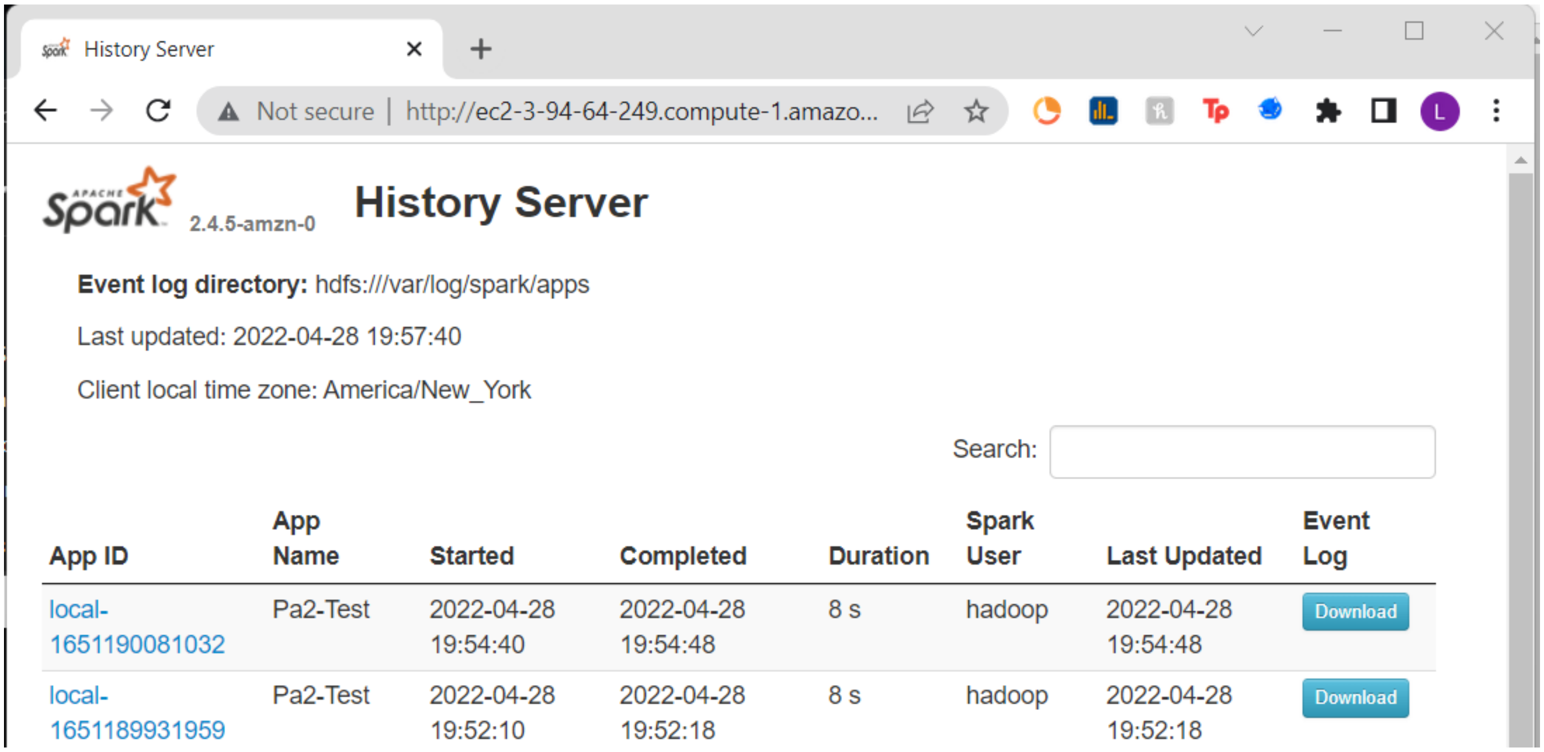
$spark-submit Test.jar.



```
----------+----------------+--------------+----------------+
""""pH""""|""""sulphates""""|""""alcohol""""|""""quality""""|
----------+----------------+--------------+----------------+
      3.39|            0.53|           9.4|             6.0|
      3.52|            0.65|           9.7|             5.0|
      3.17|            0.91|           9.5|             5.0|
      3.17|            0.53|           9.4|             5.0|
      3.43|            0.63|           9.7|             6.0|
----------+----------------+--------------+----------------+
```

Part 2 - right side from above image

As you can see above, if you navigate to the monitor section, then to the Spark dashboard.



History Server

Event log directory: hdfs:///var/log/spark/apps

Last updated: 2022-04-28 19:57:40

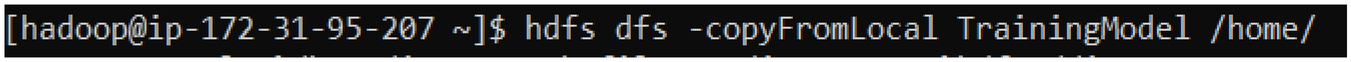
Client local time zone: America/New_York

Search:

App ID	App Name	Started	Completed	Duration	Spark User	Last Updated	Event Log
local-1651190081032	Pa2-Test	2022-04-28 19:54:40	2022-04-28 19:54:48	8 s	hadoop	2022-04-28 19:54:48	Download
local-1651189931959	Pa2-Test	2022-04-28 19:52:10	2022-04-28 19:52:18	8 s	hadoop	2022-04-28 19:52:18	Download

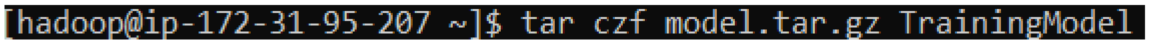
1st Try is at: http://ec2-3-83-157-50.compute-1.amazonaws.com:18080/
2nd Try is at: http://ec2-3-94-64-249.compute-1.amazonaws.com:18080/

You can see the following files have been created: folder with trained models stored, we can verify this using the following command:
$hdfs dfs -ls -t -R.



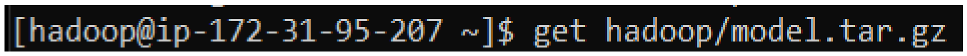
```
[hadoop@ip-172-31-95-207 ~]$ hdfs dfs -copyFromLocal TrainingModel /home/
```

We need the file back on our master node: $hdfs dfs -copyFromLocal TrainingModel /home/hadoop/



```
[hadoop@ip-172-31-95-207 ~]$ tar czf model.tar.gz TrainingModel
```

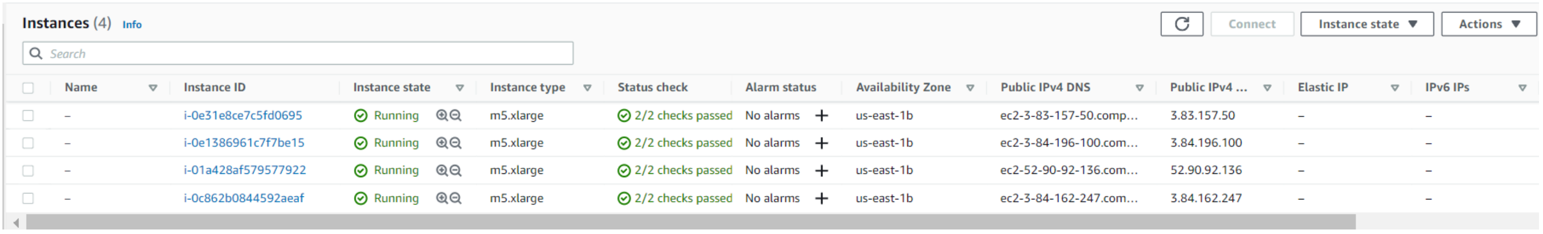
Compress the file: $tar czf model.tar.gz TrainingModel



```
[hadoop@ip-172-31-95-207 ~]$ get hadoop/model.tar.gz
```

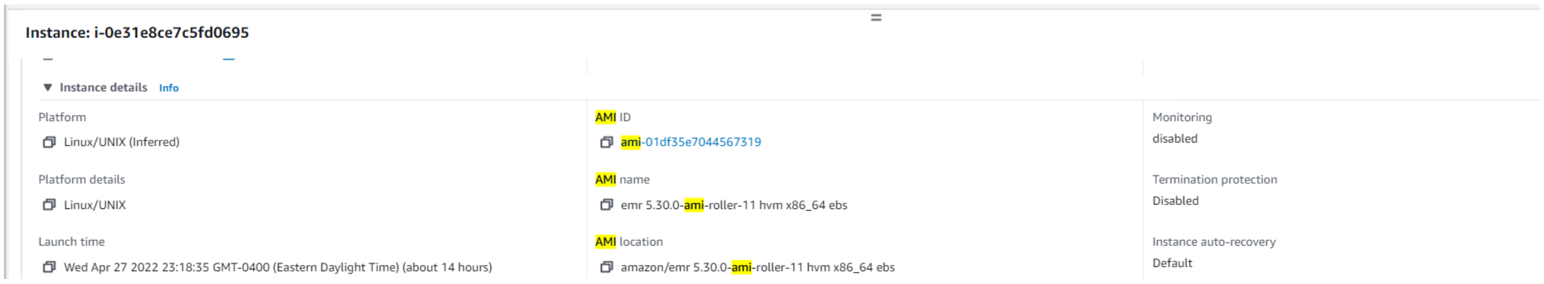
Go back to the cmd/putty session, and download it onto the local machine using: get hadoop/model.tar.gz

Step 5: Create an Ec2 Instance



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
–	i-0e31e8ce7c5fd0695	Running	m5.xlarge	2/2 checks passed	No alarms	us-east-1b	ec2-3-83-157-50.comp...	3.83.157.50	–	–
–	i-0e1386961c7f7be15	Running	m5.xlarge	2/2 checks passed	No alarms	us-east-1b	ec2-3-84-196-100.com...	3.84.196.100	–	–
–	i-01a428af579577922	Running	m5.xlarge	2/2 checks passed	No alarms	us-east-1b	ec2-52-90-92-136.com...	52.90.92.136	–	–
–	i-0c862b0844592aeaf	Running	m5.xlarge	2/2 checks passed	No alarms	us-east-1b	ec2-3-84-162-247.com...	3.84.162.247	–	–

First we need to create an Ec2 instance, as shown above.

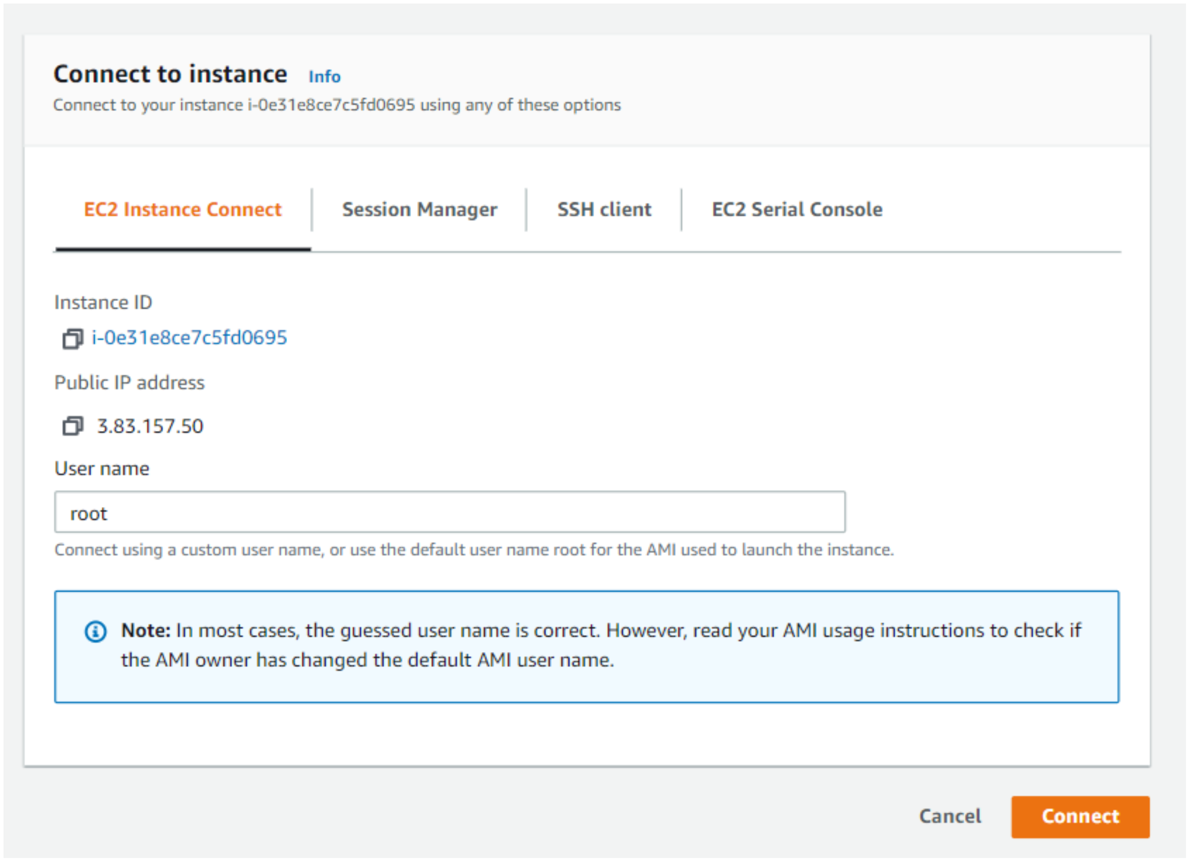


Instance: i-0e31e8ce7c5fd0695

▼ Instance details Info

Platform	AMI ID	Monitoring
Linux/UNIX (Inferred)	ami-01df35e7044567319	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	emr 5.30.0-ami-roller-11 hvm x86_64 ebs	Disabled
Launch time	AMI location	Instance auto-recovery
Wed Apr 27 2022 23:18:35 GMT-0400 (Eastern Daylight Time) (about 14 hours)	amazon/emr 5.30.0-ami-roller-11 hvm x86_64 ebs	Default

Log into the AWS console and go to EC2 => Launch instance => Select AMI as shown above.



Connect to instance Info

Connect to your instance i-0e31e8ce7c5fd0695 using any of these options

Instance ID

i-0e31e8ce7c5fd0695

Public IP address

3.83.157.50

User name

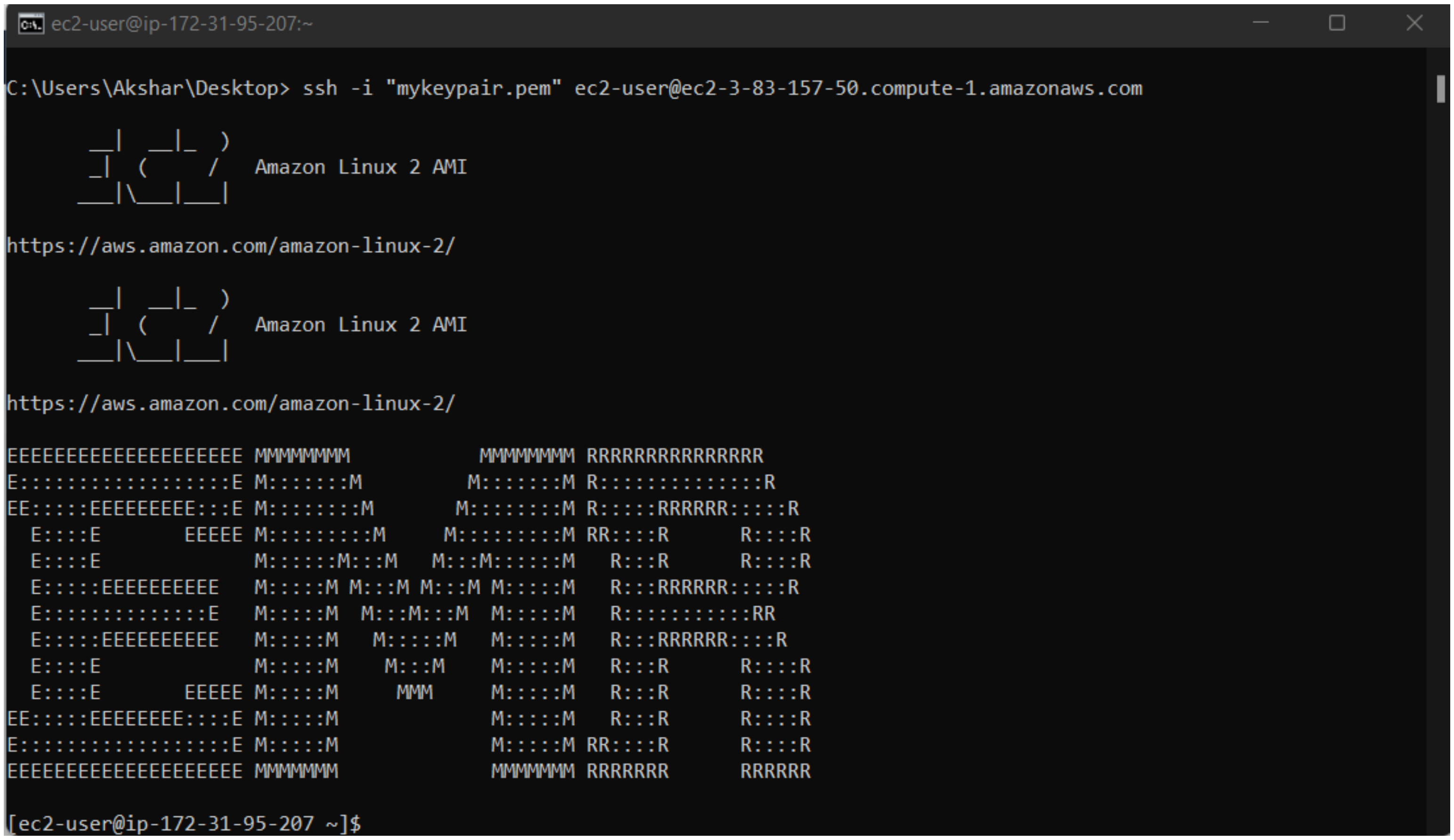
root

Connect using a custom user name, or use the default user name root for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Launch with correct authentication keypair.

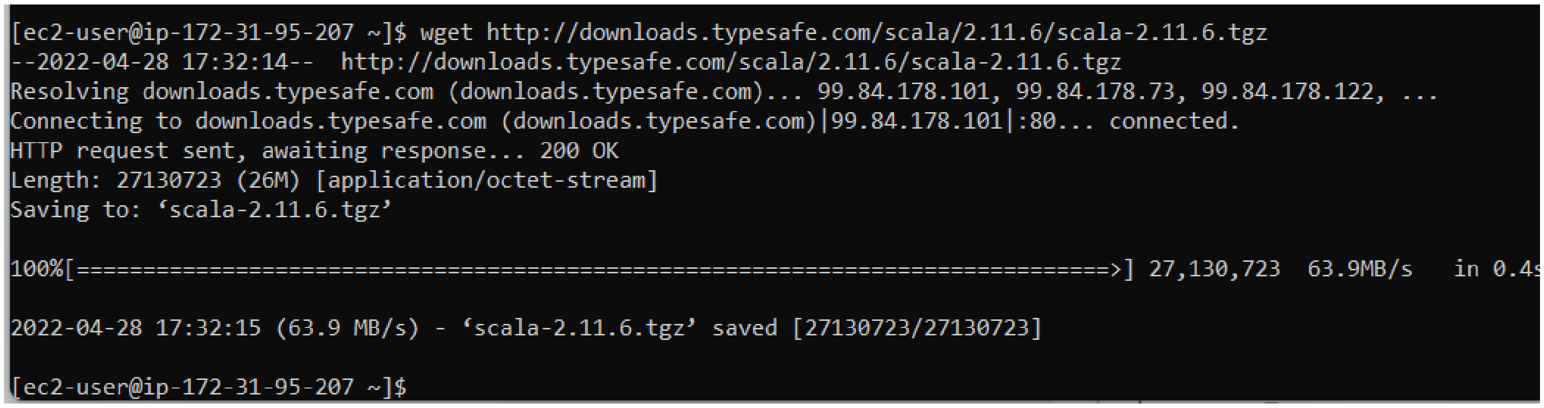
Step 6: Configure Ec2 Instance



```
ec2-user@ip-172-31-95-207:~

C:\Users\Akshar\Desktop> ssh -i "mykeypair.pem" ec2-user@ec2-3-83-157-50.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[ec2-user@ip-172-31-95-207 ~]$
```

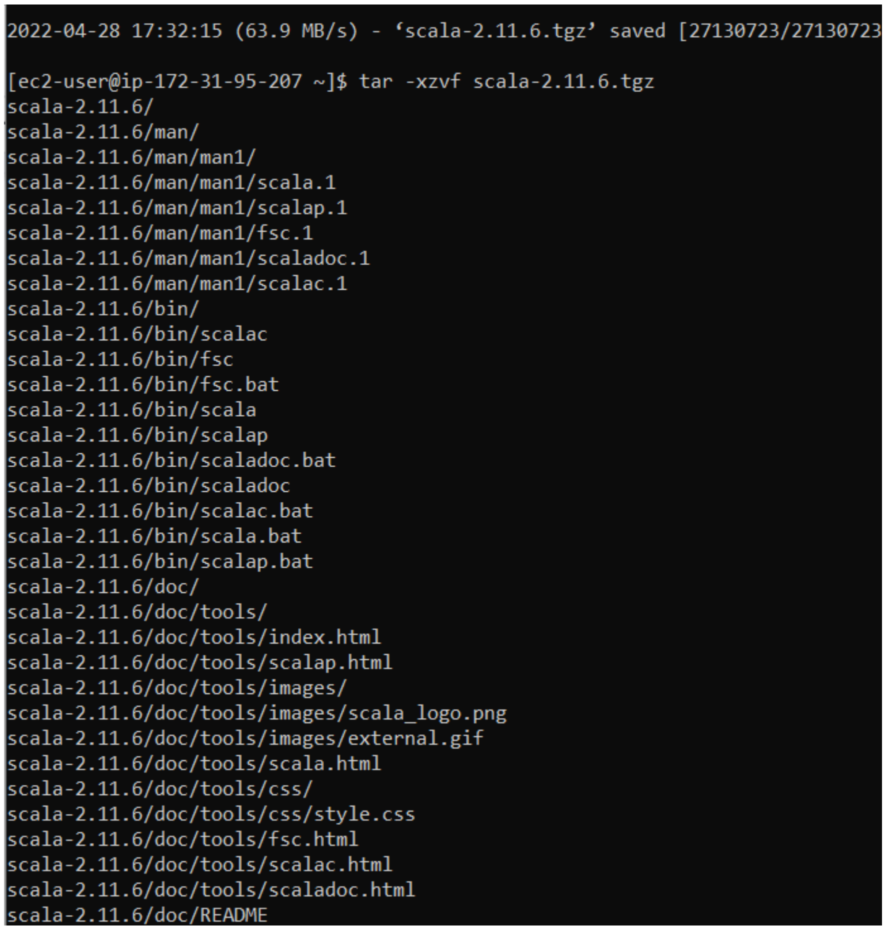
Ssh into the Ec2 instance public dns:
Mine is: $ssh -i "mykeypair.pem" root@ec2-3-83-157-50.compute-1.amazonaws.com

We require Scala and Spark, so run the following commands sequentially:



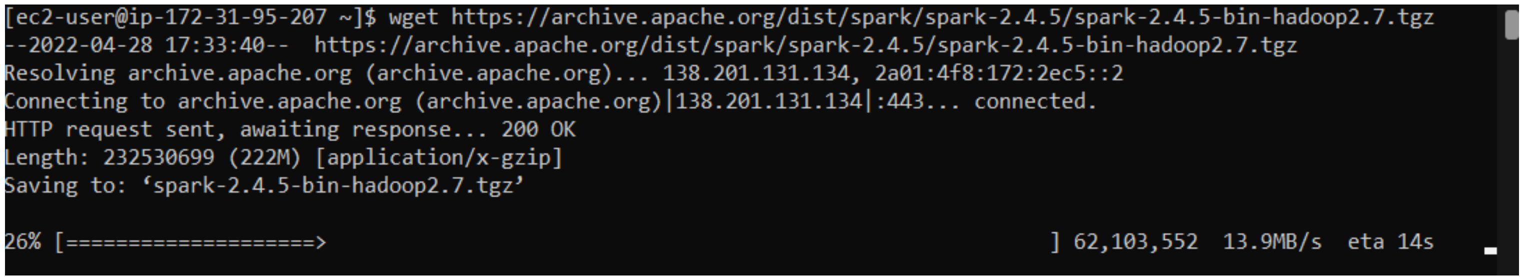
```
[ec2-user@ip-172-31-95-207 ~]$ wget http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz
--2022-04-28 17:32:14--  http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz
Resolving downloads.typesafe.com (downloads.typesafe.com)... 99.84.178.101, 99.84.178.73, 99.84.178.122, ...
Connecting to downloads.typesafe.com (downloads.typesafe.com)|99.84.178.101|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27130723 (26M) [application/octet-stream]
Saving to: ‘scala-2.11.6.tgz’

100%[==============================================================================>] 27,130,723  63.9MB/s   in 0.4

2022-04-28 17:32:15 (63.9 MB/s) - ‘scala-2.11.6.tgz’ saved [27130723/27130723]

[ec2-user@ip-172-31-95-207 ~]$
```

$wget http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz



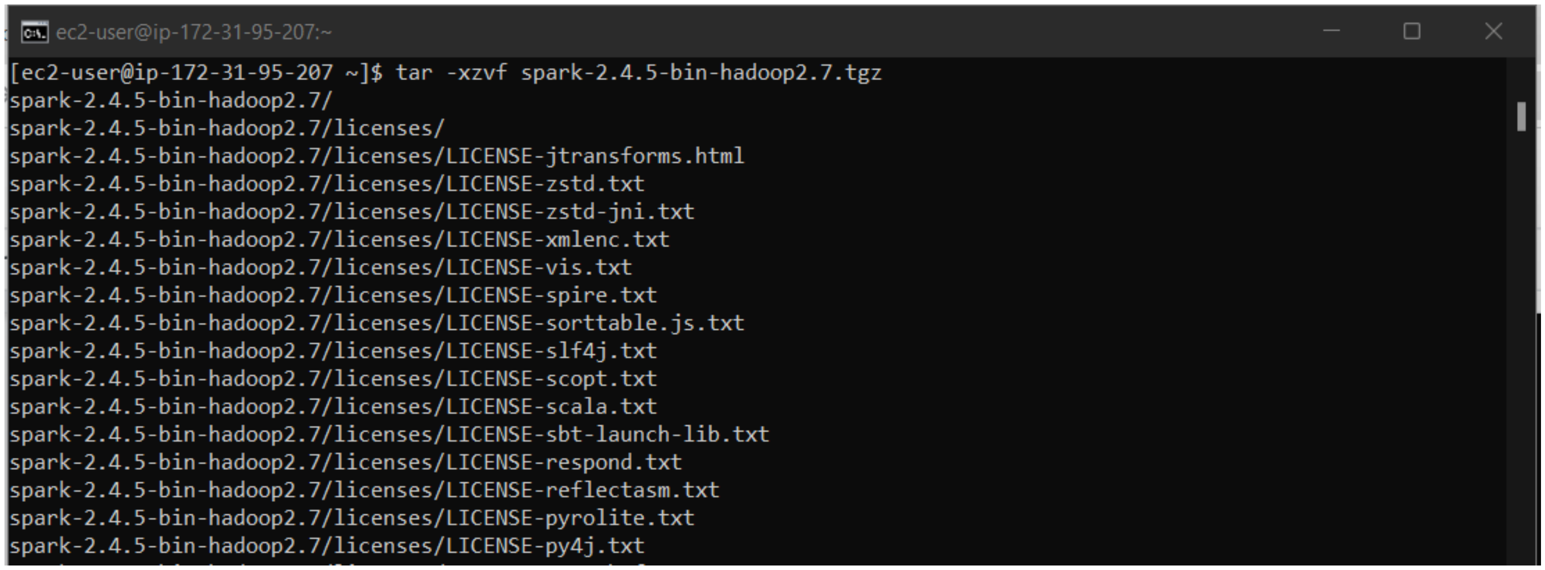
```
2022-04-28 17:32:15 (63.9 MB/s) - ‘scala-2.11.6.tgz’ saved [27130723/27130723

[ec2-user@ip-172-31-95-207 ~]$ tar -xzvf scala-2.11.6.tgz
scala-2.11.6/
scala-2.11.6/man/
scala-2.11.6/man/man1/
scala-2.11.6/man/man1/scala.1
scala-2.11.6/man/man1/scalap.1
scala-2.11.6/man/man1/fsc.1
scala-2.11.6/man/man1/scaladoc.1
scala-2.11.6/man/man1/scalac.1
scala-2.11.6/bin/
scala-2.11.6/bin/scalac
scala-2.11.6/bin/fsc
scala-2.11.6/bin/fsc.bat
scala-2.11.6/bin/scala
scala-2.11.6/bin/scalap
scala-2.11.6/bin/scaladoc.bat
scala-2.11.6/bin/scaladoc
scala-2.11.6/bin/scalac.bat
scala-2.11.6/bin/scala.bat
scala-2.11.6/bin/scalap.bat
scala-2.11.6/doc/
scala-2.11.6/doc/tools/
scala-2.11.6/doc/tools/index.html
scala-2.11.6/doc/tools/scalap.html
scala-2.11.6/doc/tools/images/
scala-2.11.6/doc/tools/images/scala_logo.png
scala-2.11.6/doc/tools/images/external.gif
scala-2.11.6/doc/tools/scala.html
scala-2.11.6/doc/tools/css/
scala-2.11.6/doc/tools/css/style.css
scala-2.11.6/doc/tools/fsc.html
scala-2.11.6/doc/tools/scalac.html
scala-2.11.6/doc/tools/scaladoc.html
scala-2.11.6/doc/README
```

$tar -xzvf scala-2.11.6.tgz



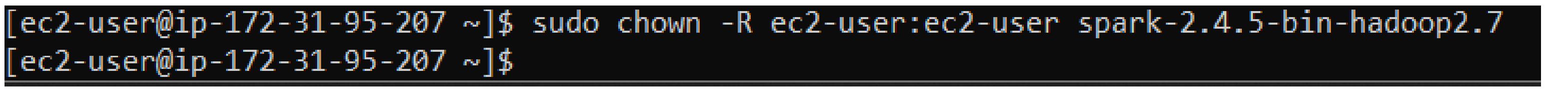
```
[ec2-user@ip-172-31-95-207 ~]$ wget https://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz
--2022-04-28 17:33:40--  https://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 232530699 (222M) [application/x-gzip]
Saving to: ‘spark-2.4.5-bin-hadoop2.7.tgz’

26% [===================>                                          ] 62,103,552  13.9MB/s  eta 14s
```

wget https://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz



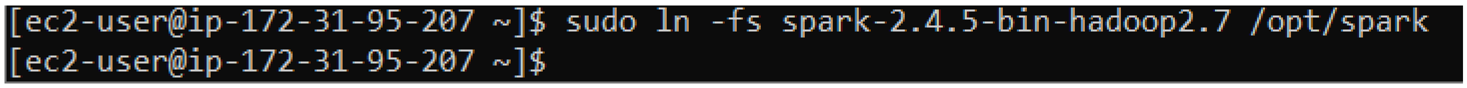
```
ec2-user@ip-172-31-95-207:~
[ec2-user@ip-172-31-95-207 ~]$ tar -xzvf spark-2.4.5-bin-hadoop2.7.tgz
spark-2.4.5-bin-hadoop2.7/
spark-2.4.5-bin-hadoop2.7/licenses/
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-jtransforms.html
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-zstd.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-zstd-jni.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-xmlenc.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-vis.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-spire.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-sorttable.js.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-slf4j.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-scopt.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-scala.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-sbt-launch-lib.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-respond.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-reflectasm.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-pyrolite.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-py4j.txt
```

tar xzvf spark-2.4.5-bin.hadoop2.7.tgz



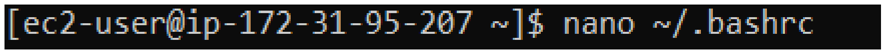
```
[ec2-user@ip-172-31-95-207 ~]$ sudo chown -R ec2-user:ec2-user spark-2.4.5-bin-hadoop2.7
[ec2-user@ip-172-31-95-207 ~]$
```

sudo chown -R ec2-user:ec2-user spark-2.4.5-bin-hadoop2.7



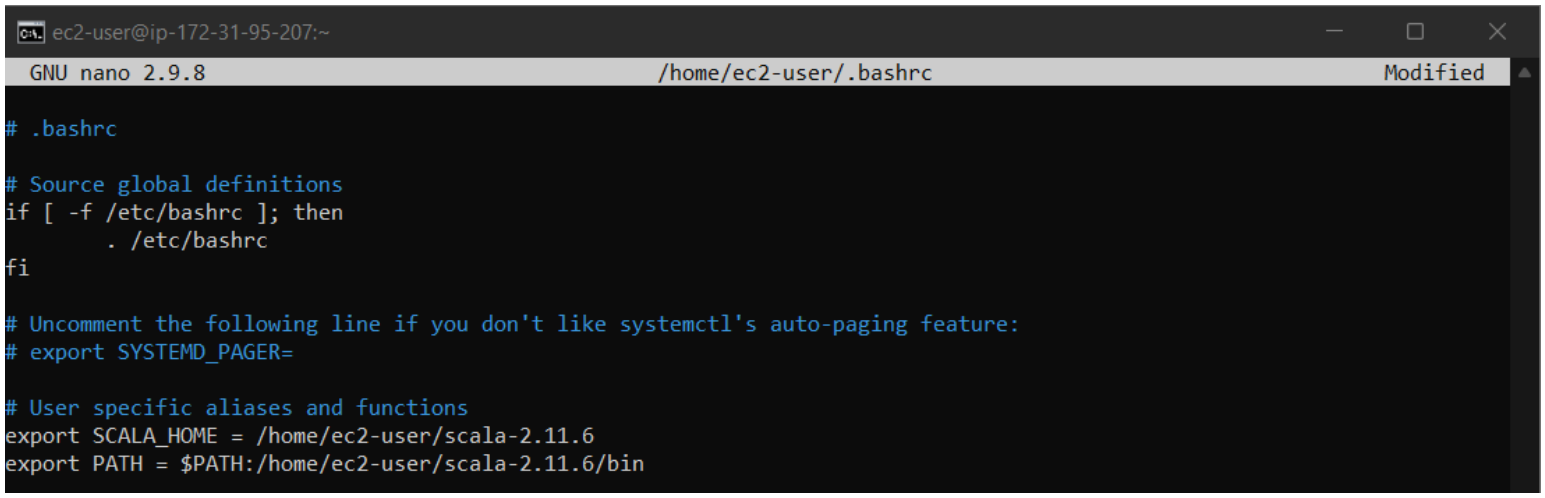
```
[ec2-user@ip-172-31-95-207 ~]$ sudo ln -fs spark-2.4.5-bin-hadoop2.7 /opt/spark
[ec2-user@ip-172-31-95-207 ~]$
```

sudo ln -fs spark-2.4.5-bin-hadoop2.7 /opt/spark



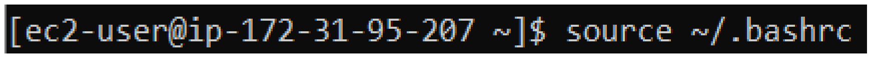
```
[ec2-user@ip-172-31-95-207 ~]$ nano ~/.bashrc
```

Open the ~/.bashrc file using this command and add the lines below: $nano ~/.bashrc



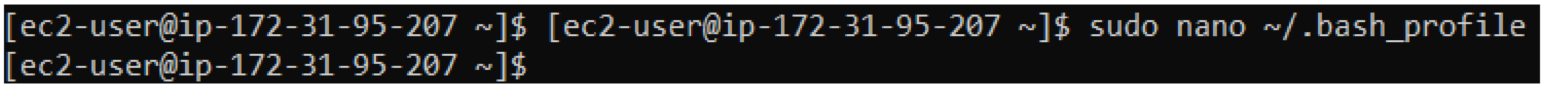
```
ec2-user@ip-172-31-95-207:~
  GNU nano 2.9.8                     /home/ec2-user/.bashrc                     Modified

# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export SCALA_HOME = /home/ec2-user/scala-2.11.6
export PATH = $PATH:/home/ec2-user/scala-2.11.6/bin
```

$ export SCALA_HOME = /home/ec2-user/scala-2.11.6
$ export PATH = $PATH:/home/ec2-user/scala-2.11.6/bin



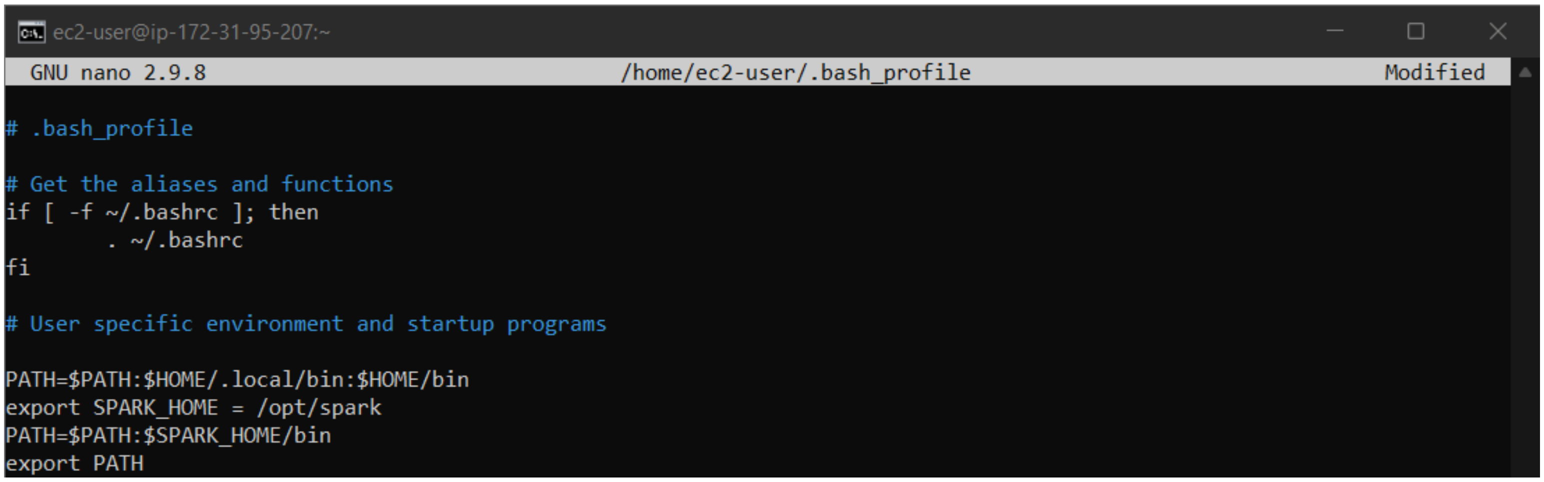
```
[ec2-user@ip-172-31-95-207 ~]$ source ~/.bashrc
```

Save it: $source ~/.bashrc



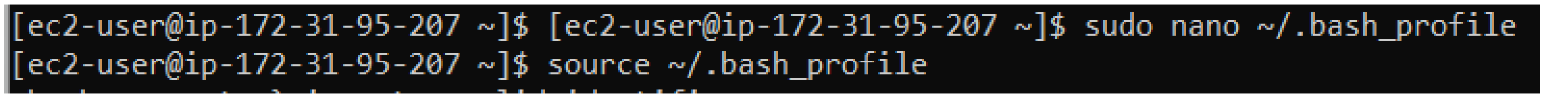
```
[ec2-user@ip-172-31-95-207 ~]$ [ec2-user@ip-172-31-95-207 ~]$ sudo nano ~/.bash_profile
[ec2-user@ip-172-31-95-207 ~]$
```

Open the ~/.bash_profile file using this command and add the lines below: $nano ~/.bash_profile



```
ec2-user@ip-172-31-95-207:~
  GNU nano 2.9.8                    /home/ec2-user/.bash_profile                    Modified

# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin
export SPARK_HOME = /opt/spark
PATH=$PATH:$SPARK_HOME/bin
export PATH
```

$ export SPARK_HOME = /opt/spark
$ PATH=$PATH:$SPARK_HOME/bin
$ export PATH

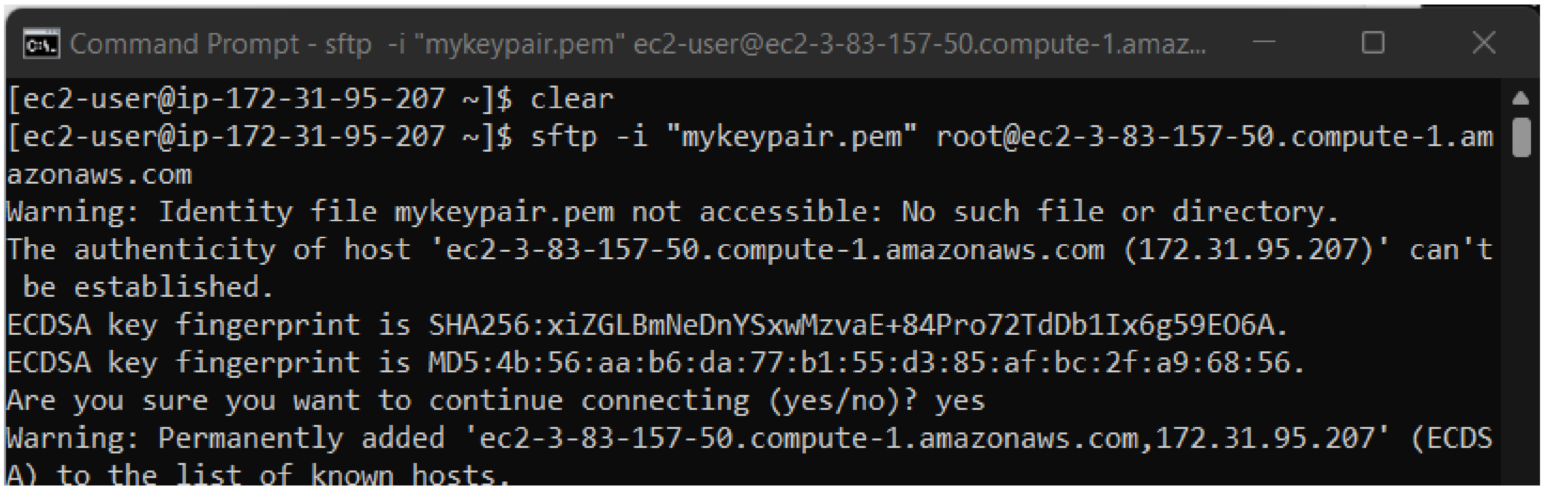


```
[ec2-user@ip-172-31-95-207 ~]$ [ec2-user@ip-172-31-95-207 ~]$ sudo nano ~/.bash_profile
[ec2-user@ip-172-31-95-207 ~]$ source ~/.bash_profile
```

Save it: $source ~/.bash_profile

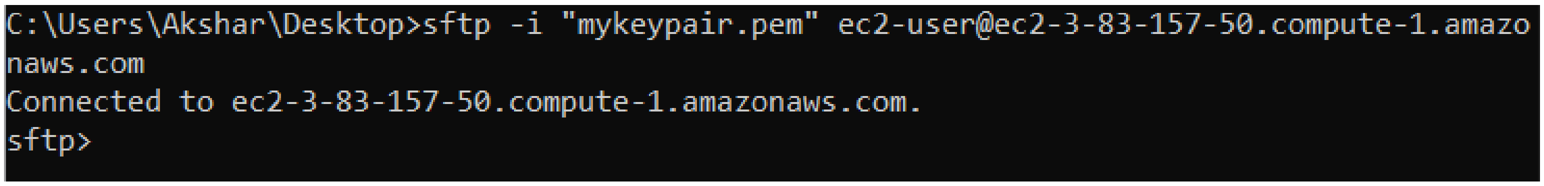
Step 7: Upload the trained model and necessary files

SFTP into the EC2 instance created in Step 5 using the following commands sequentially shown below:

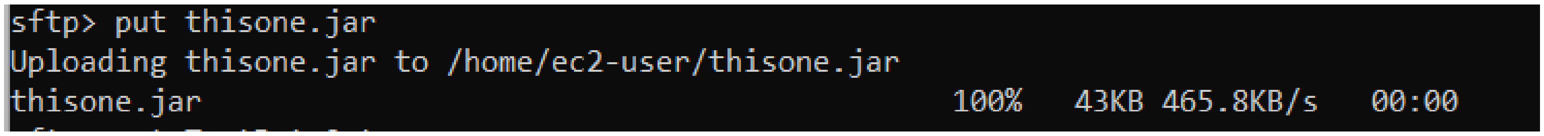


```
Command Prompt - sftp  -i "mykeypair.pem" ec2-user@ec2-3-83-157-50.compute-1.amaz...
[ec2-user@ip-172-31-95-207 ~]$ clear
[ec2-user@ip-172-31-95-207 ~]$ sftp -i "mykeypair.pem" root@ec2-3-83-157-50.compute-1.am
azonaws.com
Warning: Identity file mykeypair.pem not accessible: No such file or directory.
The authenticity of host 'ec2-3-83-157-50.compute-1.amazonaws.com (172.31.95.207)' can't
 be established.
ECDSA key fingerprint is SHA256:xiZGLBmNeDnYSxwMzvaE+84Pro72TdDb1Ix6g59EO6A.
ECDSA key fingerprint is MD5:4b:56:aa:b6:da:77:b1:55:d3:85:af:bc:2f:a9:68:56.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-83-157-50.compute-1.amazonaws.com,172.31.95.207' (ECDS
A) to the list of known hosts.
```

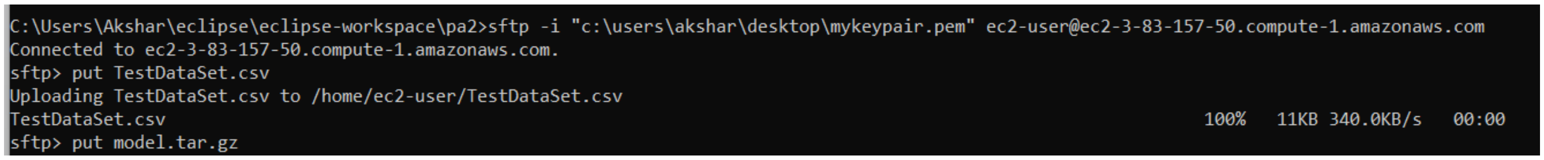
Mine is: $sftp -i ec2-A.pem root@ec2-3-83-157-50.compute-1.amazonaws.com



```
C:\Users\Akshar\Desktop>sftp -i "mykeypair.pem" ec2-user@ec2-3-83-157-50.compute-1.amazo
naws.com
Connected to ec2-3-83-157-50.compute-1.amazonaws.com.
sftp>
```



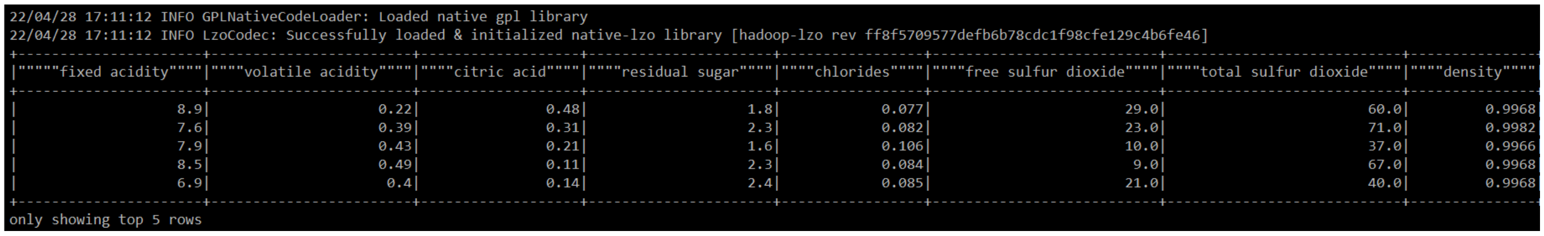
```
sftp> put thisone.jar
Uploading thisone.jar to /home/ec2-user/thisone.jar
thisone.jar                                    100%   43KB 465.8KB/s   00:00
```



```
C:\Users\Akshar\eclipse\eclipse-workspace\pa2>sftp -i "c:\users\akshar\desktop\mykeypair.pem" ec2-user@ec2-3-83-157-50.compute-1.amazonaws.com
Connected to ec2-3-83-157-50.compute-1.amazonaws.com.
sftp> put TestDataSet.csv
Uploading TestDataSet.csv to /home/ec2-user/TestDataSet.csv
TestDataSet.csv                                                                   100%   11KB 340.0KB/s   00:00
sftp> put model.tar.gz
```

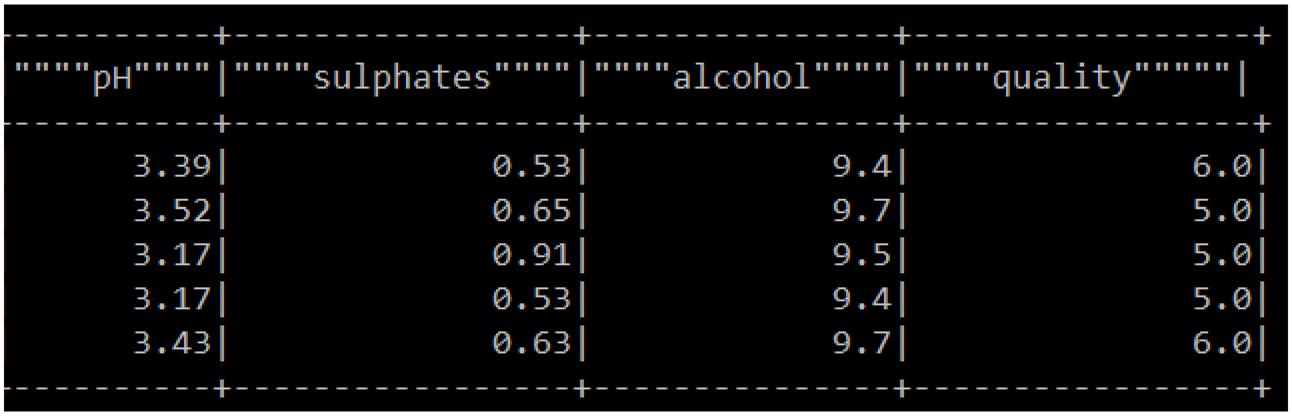
$put thisone.jar, $put TestDataSet.csv, $put model.tar.gz, $tar -xzvf model.tar.gz

Step 8: Run the application to predict



```
22/04/28 17:11:12 INFO GPLNativeCodeLoader: Loaded native gpl library
22/04/28 17:11:12 INFO LzoCodec: Successfully loaded & initialized native-lzo library [hadoop-lzo rev ff8f5709577defb6b78cdc1f98cfe129c4b6fe46]
+----------------------+------------------------+--------------------+-----------------------+-----------------+---------------------------+----------------------------+----------------
|"""""fixed acidity""""|""""volatile acidity""""|""""citric acid""""|""""residual sugar""""|""""chlorides""""|""""free sulfur dioxide""""|""""total sulfur dioxide""""|""""density""""|
+----------------------+------------------------+--------------------+-----------------------+-----------------+---------------------------+----------------------------+----------------
|                   8.9|                    0.22|                0.48|                    1.8|            0.077|                       29.0|                        60.0|          0.9968|
|                   7.6|                    0.39|                0.31|                    2.3|            0.082|                       23.0|                        71.0|          0.9982|
|                   7.9|                    0.43|                0.21|                    1.6|            0.106|                       10.0|                        37.0|          0.9966|
|                   8.5|                    0.49|                0.11|                    2.3|            0.084|                        9.0|                        67.0|          0.9968|
|                   6.9|                     0.4|                0.14|                    2.4|            0.085|                       21.0|                        40.0|          0.9968|
+----------------------+------------------------+--------------------+-----------------------+-----------------+---------------------------+----------------------------+----------------
only showing top 5 rows
```

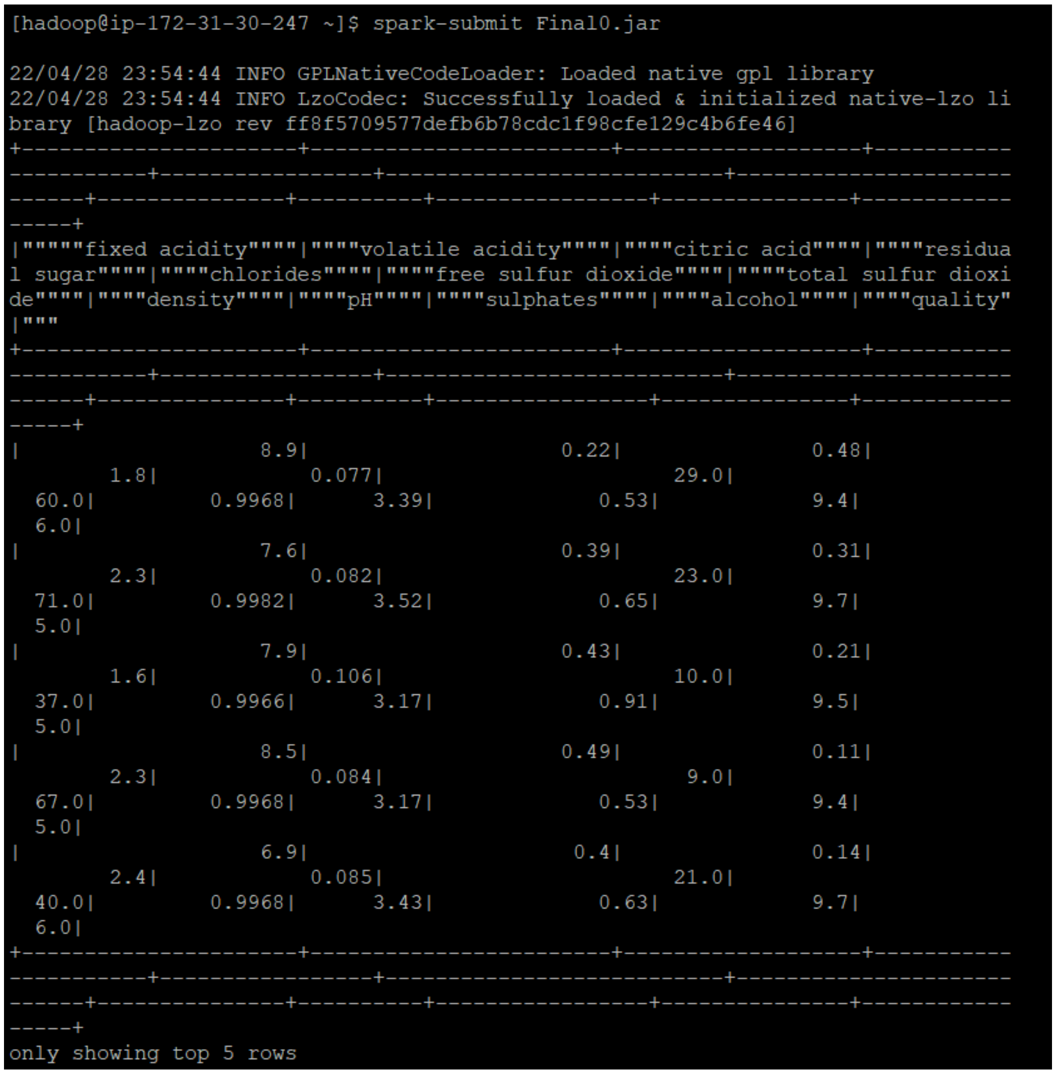
$spark-submit new.jar.



```
----------+----------------+--------------+----------------+
""""pH""""|""""sulphates""""|""""alcohol""""|""""quality""""|
----------+----------------+--------------+----------------+
      3.39|            0.53|           9.4|             6.0|
      3.52|            0.65|           9.7|             5.0|
      3.17|            0.91|           9.5|             5.0|
      3.17|            0.53|           9.4|             5.0|
      3.43|            0.63|           9.7|             6.0|
----------+----------------+--------------+----------------+
```

Part 2 - right side from above image

Verify run: Same result (below)



```
[hadoop@ip-172-31-30-247 ~]$ spark-submit Final0.jar

22/04/28 23:54:44 INFO GPLNativeCodeLoader: Loaded native gpl library
22/04/28 23:54:44 INFO LzoCodec: Successfully loaded & initialized native-lzo li
brary [hadoop-lzo rev ff8f5709577defb6b78cdc1f98cfe129c4b6fe46]
+----------------------+------------------------+-------------------+-----------
-----------+-----------------+---------------------------+----------------------
------+---------------+----------+-----------------+---------------+------------
-----+
|"""""fixed acidity""""|""""volatile acidity""""|""""citric acid""""|""""residua
l sugar""""|""""chlorides""""|""""free sulfur dioxide""""|""""total sulfur dioxi
de""""|""""density""""|""""pH""""|""""sulphates""""|""""alcohol""""|""""quality"
|"""
+----------------------+------------------------+-------------------+-----------
-----------+-----------------+---------------------------+----------------------
------+---------------+----------+-----------------+---------------+------------
-----+
|                   8.9|                    0.22|               0.48|
        1.8|            0.077|                       29.0|
  60.0|         0.9968|      3.39|             0.53|            9.4|
  6.0|
|                   7.6|                    0.39|               0.31|
        2.3|            0.082|                       23.0|
  71.0|         0.9982|      3.52|             0.65|            9.7|
  5.0|
|                   7.9|                    0.43|               0.21|
        1.6|            0.106|                       10.0|
  37.0|         0.9966|      3.17|             0.91|            9.5|
  5.0|
|                   8.5|                    0.49|               0.11|
        2.3|            0.084|                        9.0|
  67.0|         0.9968|      3.17|             0.53|            9.4|
  5.0|
|                   6.9|                     0.4|               0.14|
        2.4|            0.085|                       21.0|
  40.0|         0.9968|      3.43|             0.63|            9.7|
  6.0|
+----------------------+------------------------+-------------------+-----------
-----------+-----------------+---------------------------+----------------------
------+---------------+----------+-----------------+---------------+------------
-----+
only showing top 5 rows
```

Step 9: Using docker to predict/verify (easier way)

- $docker pull loveshahnjit/pa2
- $docker run -v pa2/TestDataSet.csv
- $docker run -v loveshahnjit/pa2/TestDataSet.csv