

# Atria Institute of Technology



**Department of Information Science and Engineering**

**Big Data Analytics (18CS72)**

**Assignment-1**

**SUBMITTED BY**

Name: RICHA H S

USN:1AT20IS076

Section: B

Submission Date: 27-11-2023

**COURSE HANDLING FACULTY NAME:**

Dr. K S Ananda Kumar

Associate Professor

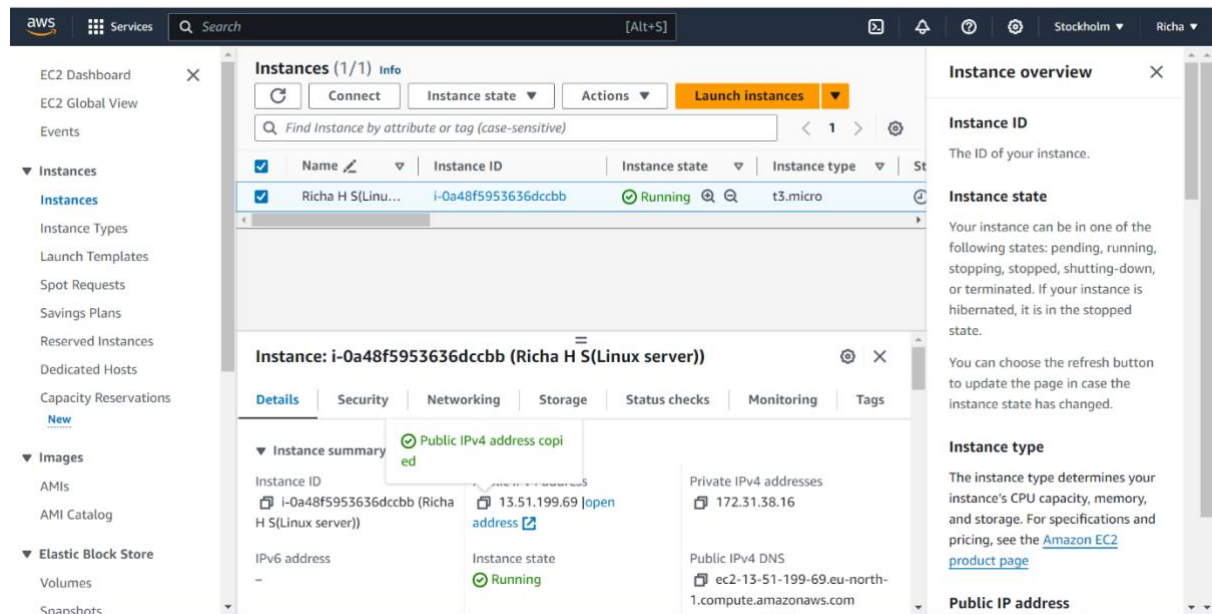
Dept of ISE, Atria IT.

## Table of contents

Sl. No	Description
1	1. create an <b>EC2 Linux</b> instance in AWS Cloud /Any cloud INSTANCE NAME - <b>YOUR NAME</b> INSTANCE TYPE - t2.micro/any other also. key pair name- your name storage - 10 GB Take the screenshot of instance running status Mention the private IP address and Public IP address. (Execute this program/concept and take a screenshot of the output)
2	Execute the basic Linux commands/ simple program on the instance (Execute this program and take a screenshot of the output)
3	Create the <b>GitHub</b> Account with your credentials, Same things stored in public repository in Github. Share the assignment in github link.

# Instance Creation-01

## SCREENSHOTS OF AWS INSTANCE



- Instance:  
i-0a48f5953636dccbb (Richa H S(Linux server))
- Instance ID:  
i-0a48f5953636dccbb (Richa H S(Linux server))
- Public IPv4 address:  
13.51.199.69
- Private IPv4 addresses:  
172.31.38.16
- Instance state:  
Running

# SCREENSHOTS OF AWS INSTANCE

Instance: i-0a48f5953636dccbb (Richa H S(Linux server))

▼ Instance summary Info

Instance ID

i-0a48f5953636dccbb (Richa H S(Linux server))

IPv6 address

-

Hostname type

IP name: ip-172-31-38-16.eu-north-1.compute.internal

Public IPv4 address

13.51.199.69 [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-38-16.eu-north-1.compute.internal

Private IPv4 addresses

172.31.38.16

Public IPv4 DNS

ec2-13-51-199-69.eu-north-1.compute.amazonaws.com [open address](#)

Instance: i-0a48f5953636dccbb (Richa H S(Linux server))

Answer private resource DNS name IPv4 (A)

Auto-assigned IP address

13.51.199.69 [Public IP]

Instance type

t3.micro

VPC ID

vpc-0ee72b16b745d7c62

Subnet ID

subnet-0de5ecdc5f112fc6b

Elastic IP addresses

-

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)

Auto Scaling Group name

-

### Instance: i-0a48f5953636dccbb (Richa H S(Linux server))



IMDSv2  
Required

#### ▼ Instance details [Info](#)

Platform  
 Amazon Linux (Inferred)

AMI ID  
 [ami-0416c18e75bd69567](#)

Monitoring  
disabled

Platform details  
 Linux/UNIX

AMI name  
 al2023-ami-2023.2.20231113.0-kernel-6.1-x86\_64

Termination protection  
Disabled

Stop protection  
Disabled

Launch time  
 Sun Nov 26 2023 20:21:07

AMI location  
 amazon/al2023-ami-

### Instance: i-0a48f5953636dccbb (Richa H S(Linux server))



Stop protection  
Disabled

Launch time  
 Sun Nov 26 2023 20:21:07 GMT+0530 (India Standard Time) (3 minutes)

AMI location  
 amazon/al2023-ami-2023.2.20231113.0-kernel-6.1-x86\_64

Instance auto-recovery  
Default

Lifecycle  
normal

Stop-hibernate behavior  
Disabled

AMI Launch index  
0

Key pair assigned at launch  
 [richahs-linux](#)

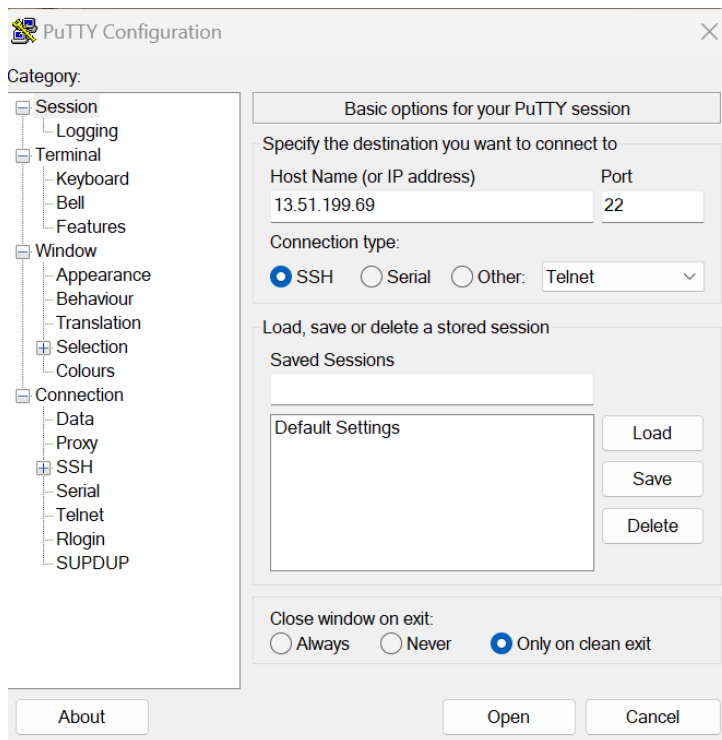
State transition reason  
-

Credit specification  
unlimited

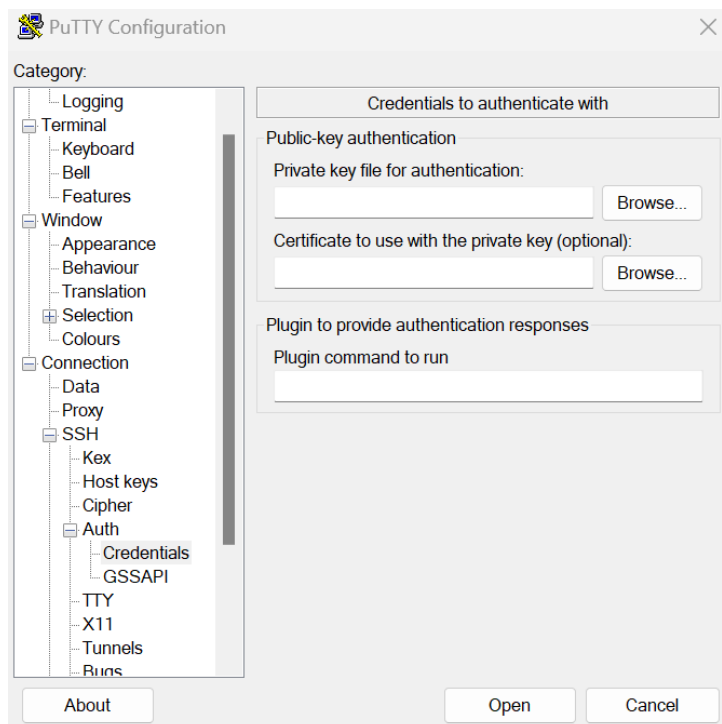
Kernel ID  
-

State transition message  
-

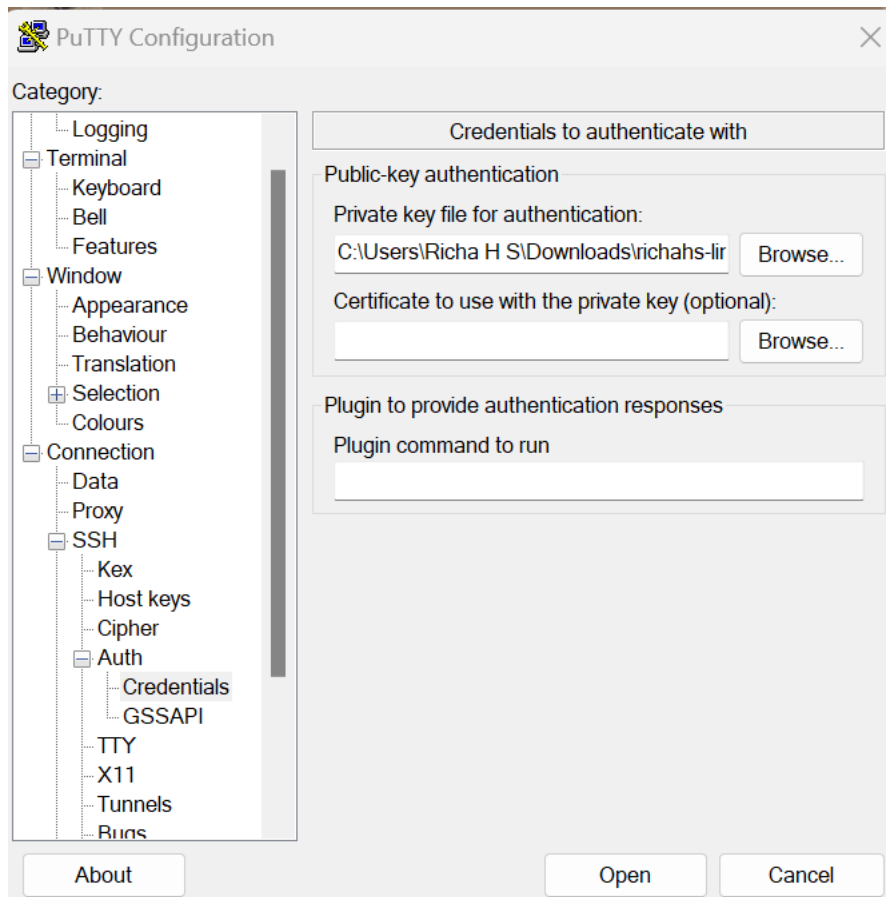
## SCREENSHOTS FROM puTTY-Setup



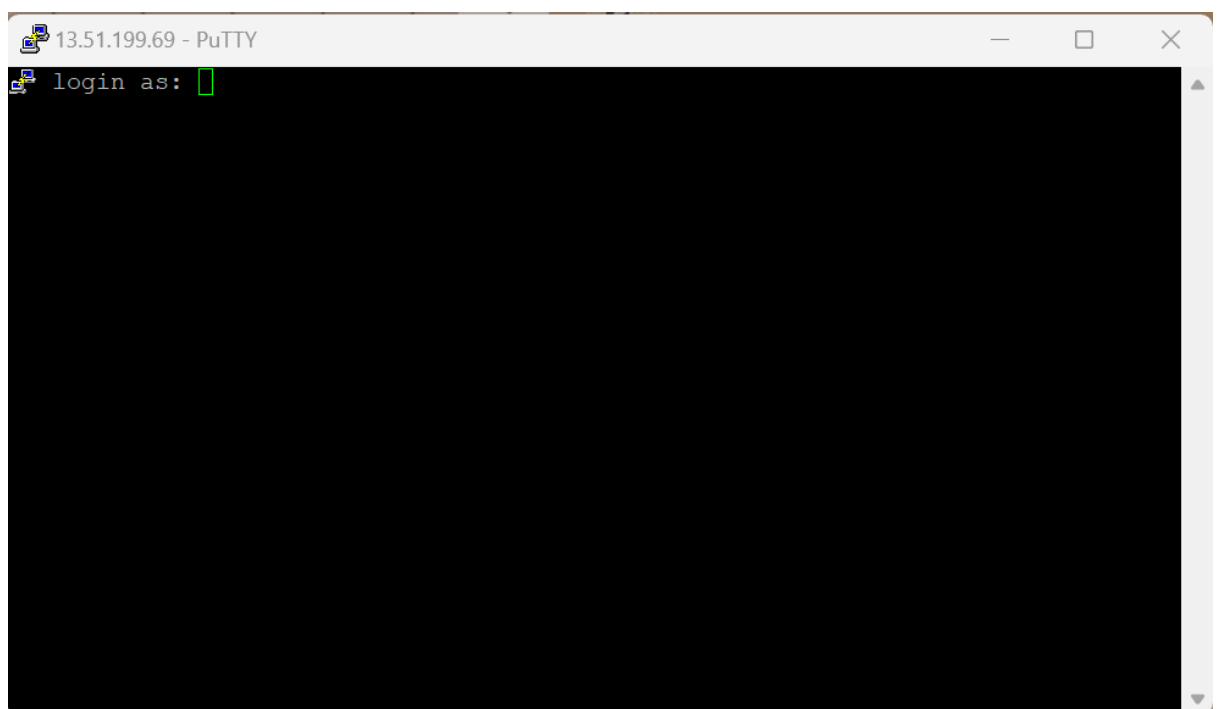
In putty under host name enter the Public IPv4 address.



Under Auth in private key file for authentication browse and add the PuTTY Private Key File (.ppk) named richahs-linux.



After browsing click on Open button



This directs us to this screen now enter the User name that is ec2-user





## SCREENSHOTS FROM puTTY-commands

```
login as: ec2-user
Authenticating with public key "richahs-linux"

#_
~\  #####_      Amazon Linux 2023
~~\  #####\
~~  \####|
~~   \#/      https://aws.amazon.com/linux/amazon-linux-2023
~~    V~' '->
~~~
~~~.~.~
~~~/_/
~~~/_m/'

Last login: Sun Nov 26 15:40:40 2023 from 49.37.250.114
[ec2-user@ip-172-31-38-16 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-38-16 ~]$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
[ec2-user@ip-172-31-38-16 ~]$ mkdir new_directory
[ec2-user@ip-172-31-38-16 ~]$ ls
new_directory
[ec2-user@ip-172-31-38-16 ~]$
```

- pwd:  
pwd is used to present working directory, this gave the output /home/ec2-user
- mkdir:  
The mkdir (**make directory**) command creates a new directory in the provided location. I have created a directory called new\_directory .
- ls:  
The ls command (**list**) prints a list of the current directory's contents. Therefore we got the directory created display as output.

```
[ec2-user@ip-172-31-38-16 ~]$ ls
new_directory
[ec2-user@ip-172-31-38-16 ~]$ touch new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ ls
new_directory  new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ touch file.txt
[ec2-user@ip-172-31-38-16 ~]$ ls
file.txt  new_directory  new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ cat new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ echo "hello this is the content in new_file.txt" >> new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ cat new_file.txt
hello this is the content in new_file.txt
[ec2-user@ip-172-31-38-16 ~]$
```

- touch:
  - The touch command's primary purpose is to modify an existing file's timestamp. The command creates an empty file if it does not exist. Due to this effect, touch is also a quick way to make a new file (or a batch of files).
  - Here I have created a txt file called new\_file first then a second txt file called file\_txt.
  - Using ls command we can find where these files have been created .
- cat:
  - The cat command (concatenate) displays the contents of a file in the terminal (standard output or stdout).
  - To use the command, provide a file name from the current directory.
  - Here I provide the txt file called new\_file.txt.
- echo:
  - The echo command to print arguments to the terminal.
  - Here I have used echo “hello this is the content in new\_file.txt”.
  - The >> operator redirects output to a file.

Later I use cat to find the content in new\_file.txt. Therefore we can see that “hello this is the content in new\_file.txt” has been added to new\_file.txt

```
[ec2-user@ip-172-31-38-16 ~]$ cat new_file.txt
hello this is the content in new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ cat file.txt
[ec2-user@ip-172-31-38-16 ~]$ cp new_file.txt file.txt
[ec2-user@ip-172-31-38-16 ~]$ cat file.txt
hello this is the content in new_file.txt
[ec2-user@ip-172-31-38-16 ~]$ echo "after copying contents from new_file i am adding a new line into file .txt" >> file.txt
[ec2-user@ip-172-31-38-16 ~]$ cat file.txt
hello this is the content in new_file.txt
after copying contents from new_file i am adding a new line into file .txt
[ec2-user@ip-172-31-38-16 ~]$
```

cat file.txt is executed to show that there is no content in file.txt.

- cp:
  - The main way to copy files and directories in Linux is through the cp (**copy**) command. cp <source file> <target file>.
  - The source and target files must have different names since the command copies in the same directory. Provide a path before the file name to copy to another location.
  - Here we are copying the content of new\_file.txt into file.txt using cp [cp new\_file.txt file.txt]
  - Then when we use cat on file.txt it shows “hello this is the content in new\_file.txt” so content is successfully copied.
  - Now we make use of echo and >> to add a new line in file.txt i.e “after copying contents from new\_file I am adding a new line into file.txt”.
  - Now when cat is used on file.txt both are lines are given as output.

**Assignment GitHub Link:**

<https://github.com/richasalian/BDA-assignment-1>

