Roll NO. A041

Cloud Computing Practical No: 01

Writeup:

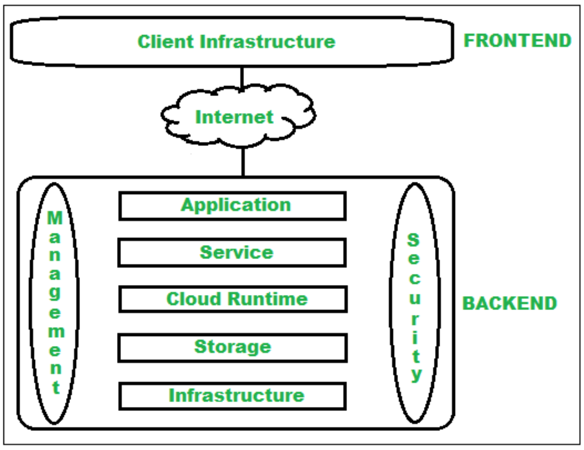
1.) Cloud Computing architecture

Cloud Computing means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server.

Cloud computing architecture is a combination of **service-oriented architecture** and **event-driven architecture**.

Cloud computing architecture is divided into the following two parts-

Front end and Back end



1. Frontend :

Frontend of the cloud architecture refers to the client side of cloud computing system. Means it contains all the user interfaces and applications which are used by the client to access the cloud computing services/resources. For example, use of a web browser to access the cloud platform.

- **Client Infrastructure –** Client Infrastructure is a part of the frontend component. It contains the applications and user interfaces which are required to access the cloud platform.
- In other words, it provides a GUI(Graphical User Interface) to interact with the cloud.

2. Backend :

Backend refers to the cloud itself which is used by the service provider. It contains the resources as well as manages the resources and provides security mechanisms. Along with this, it includes huge storage, virtual applications, virtual machines, traffic control mechanisms, deployment models, etc.

1.Application –Application in backend refers to a software or platform to which client accesses. Means it provides the service in backend as per the client requirement.

2.Service –Service in backend refers to the major three types of cloud based services like SaaS, PaaS and IaaS. Also manages which type of service the user accesses.

3. Runtime Cloud-Runtime cloud in backend provides the execution and Runtime platform/environment to the Virtual machine.
4.Storage –Storage in backend provides flexible and scalable storage service and management of stored data.
5.Infrastructure –Cloud Infrastructure in backend refers to the hardware and software components of cloud like it includes servers, storage, network devices, virtualization software etc.

2.) Infrastructure as a Service (IaaS):

Infrastructure-as-a-Service (IaaS) is a cloud computing service offering on-demand computing, storage, and networking resources. It usually works on a pay-as-you-go basis.Organizations can purchase resources on-demand and as needed instead of buying the hardware outright.

The IaaS cloud vendor hosts the infrastructure components, including the on-premises data centre, servers, storage, networking hardware, and the hypervisor (virtualization layer).

This Model contains the basic building blocks for your web application. It provides complete control over the hardware that runs your application (storage, servers, VMs, networks & operating systems). IaaS model gives you the best flexibility and management control over your IT resources.

Characteristics of IaaS

There are the following characteristics of IaaS:

- Resources are available as a service
- Services are highly scalable
- Dynamic and flexible Cloud Service Model
- GUI and API-based access
- Automate the administrative tasks

Advantages of IaaS

There are the following advantages of IaaS computing layer -

1. Shared infrastructure-IaaS allows multiple users to share the same physical infrastructure.

2. Web access to the resources-Iaas allows IT users to access resources over the internet.

3. Pay-as-per-use model-IaaS providers provide services based on the pay-as-per-use basis. The users are required to pay for what they have used.

4. Focus on the core business-IaaS providers focus on the organization's core business rather than on IT infrastructure.

5. On-demand scalability-On-demand scalability is one of the biggest advantages of IaaS. Using IaaS, users do not worry about to upgrade software and troubleshoot the issues related to hardware components.

Disadvantages of IaaS cloud computing layer

1. Security-Security is one of the biggest issues in IaaS. Most of the IaaS providers are not able to provide 100% security.

2. Maintenance & Upgrade-Although IaaS service providers maintain the software, but they do not upgrade the software for some organizations.

3. Interoperability issues-It is difficult to migrate VM from one IaaS provider to the other, so the customers might face problem related to vendor lock-in.

3.) Amazon Web Services(AWS)

Amazon Web Services (AWS) is a comprehensive, evolving cloud computing platform provided by Amazon. It offers a wide range of services, including computing power, storage, databases, machine learning, analytics, content delivery, Internet of Things (IoT), security, and more. AWS allows businesses and individuals to access computing resources on a pay-as-you-go basis, without the need to invest in and maintain physical infrastructure.

Advantages of AWS

1) Flexibility-We can get more time for core business tasks due to the instant availability of new features and services in AWS.It provides effortless hosting of legacy applications. AWS does not require learning new technologies and migration of applications to the AWS provides the advanced computing and efficient storage.

2) Cost-effectiveness-AWS requires no upfront investment, long-term commitment, and minimum expense when compared to traditional IT infrastructure that requires a huge investment.

3) Scalability/Elasticity-Through AWS, autoscaling and elastic load balancing techniques are automatically scaled up or down, when demand increases or decreases respectively. AWS techniques are ideal for handling unpredictable or very high loads. Due to this reason, organizations enjoy the benefits of reduced cost and increased user satisfaction.

4) Security-AWS provides end-to-end security and privacy to customers.AWS has a virtual infrastructure that offers optimum availability while managing full privacy and isolation of their operations.AWS ensures the three aspects of security, i.e., Confidentiality, integrity, and availability of user's data.

4.)AWS Services

Compute Services:

Amazon EC2 (Elastic Compute Cloud): Provides scalable virtual servers in the cloud.

AWS Lambda: Allows you to run code without provisioning or managing servers.

Storage Services:Amazon S3 (Simple Storage Service): Scalable object storage for data storage and retrieval.

Amazon EBS (Elastic Block Store): Provides persistent block-level storage volumes for use with EC2 instances.

Database Services:

Amazon RDS (Relational Database Service): Managed relational databases in the cloud.

Amazon DynamoDB: A fully managed NoSQL database service.

Networking:

Amazon VPC (Virtual Private Cloud): Allows you to provision a logically isolated section of the AWS Cloud.

Amazon Route 53: A scalable domain name system (DNS) web service.

Machine Learning and AI:

Amazon SageMaker: A fully managed service for building, training, and deploying machine learning models.

Amazon Comprehend: Natural language processing service for extracting insights and relationships from text.

Analytics:

Amazon Redshift: A fully managed data warehouse service.

Amazon EMR (Elastic MapReduce): A cloud-based big data platform for processing and analyzing large datasets.

Security:

AWS Identity and Access Management (IAM): Enables you to securely control access to AWS services.

Amazon GuardDuty: A managed threat detection service.

IoT (Internet of Things):

AWS IoT Core: Connects devices to the cloud and allows them to interact with AWS services.

DevOps:

AWS CodeDeploy: Automates code deployments to any instance, including EC2 instances and on-premises servers.

AWS CodePipeline: A continuous integration and continuous delivery (CI/CD) service.

Containers:

Amazon ECS (Elastic Container Service): A fully managed container orchestration service.

Amazon EKS (Elastic Kubernetes Service): Managed Kubernetes service.

5.)Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.

It allows organizations to obtain and configure virtual compute capacity in the cloud. You can select from a variety of operating systems and resource configurations like memory, CPU, and storage that are required for your application.

Amazon EC2 enables you to increase or decrease capacity within minutes. You can use one or hundreds or even thousands of server instances simultaneously. Because this is all controlled with web service APIs, your application can automatically scale itself up and down depending on its needs.

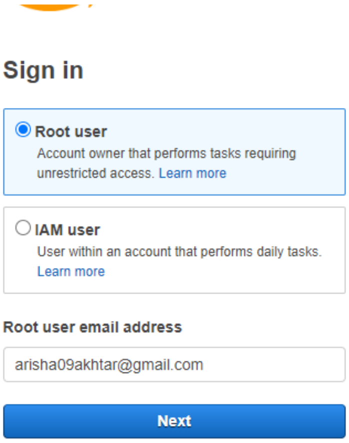
Amazon EC2 is integrated with most AWS services, such as Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), and Amazon Virtual Private Cloud (Amazon VPC) to provide a complete, secure solution for computing applications.

Amazon EC2 is an example of Infrastructure as a Service(IaaS). EC2 delivers secure, reliable, cost-effective compute and high-performance compute infrastructure so as to meet the needs of demanding businesses.

Amazon EC2 is one of the easiest ways of providing servers on AWS Cloud and also the access to Operating system.

1.Implement the windows machine using AWS ec2.

1. Sign In to your aws acc



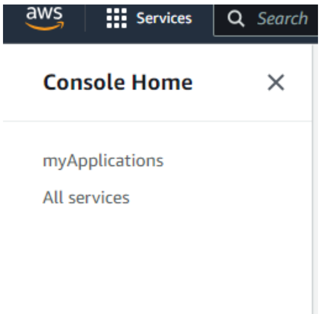
Sign in

● Root user
Account owner that performs tasks requiring unrestricted access. Learn more

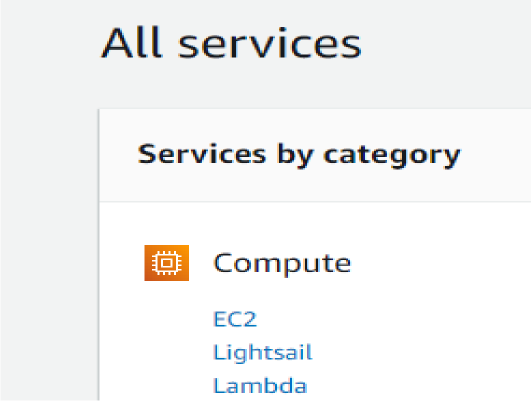
○ IAM user
User within an account that performs daily tasks. Learn more

Root user email address
arisha09akhtar@gmail.com

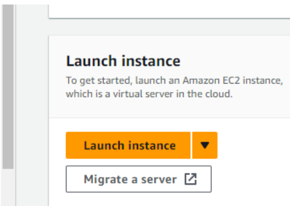
2.Select all services



3.Select EC2

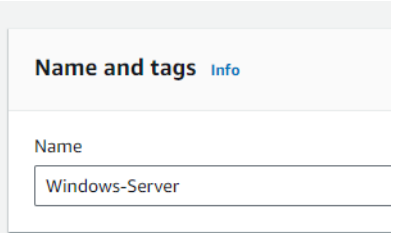


4.Launch Instance



Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

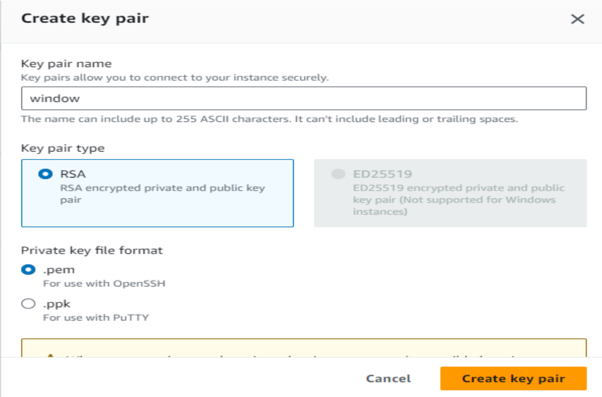
5.Put name , create key value pair in .pem extension and then one file will get downloaded open it with notepad



Name and tags Info

Name

Windows-Server



Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

window

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

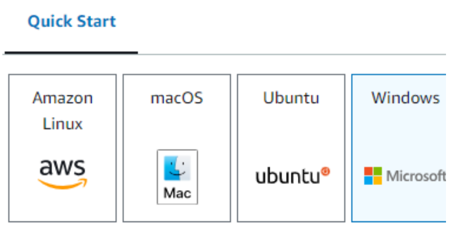
○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

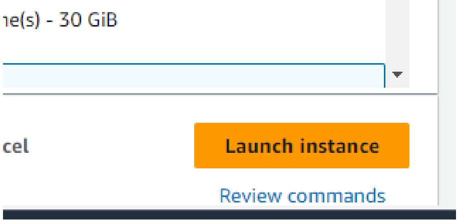
● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

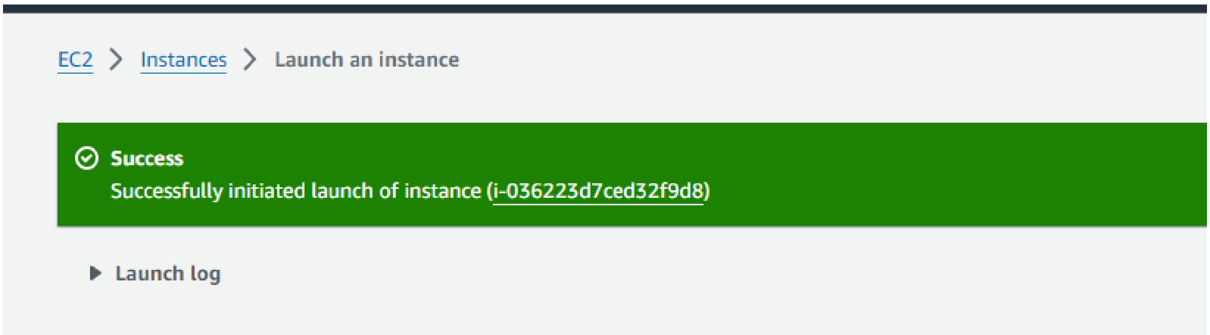
6.Select Windows



7.Launch the instance



ne(s) - 30 GiB

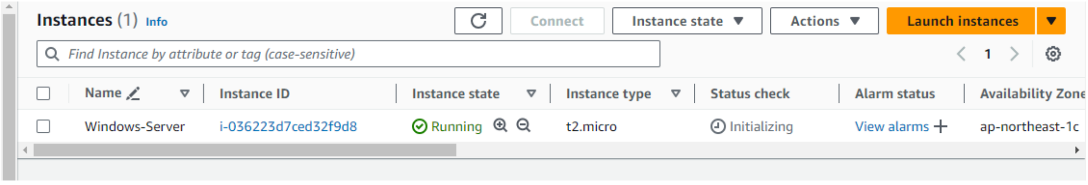


EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-036223d7ced32f9d8)

▶ Launch log

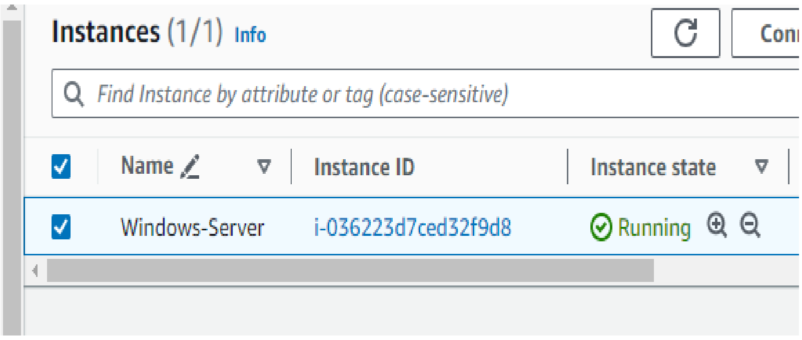
8.Go to instances and refresh It will initialize and then start running



Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	Windows-Server	i-036223d7ced32f9d8	✓ Running	t2.micro	Initializing	View alarms +	ap-northeast-1c

9.Select the instance

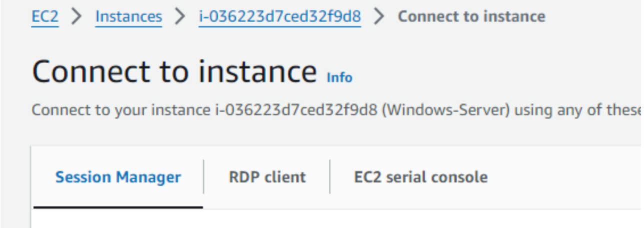


Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

☑	Name	Instance ID	Instance state
☑	Windows-Server	i-036223d7ced32f9d8	✓ Running

10.Click on connect, for connecting to RDP client

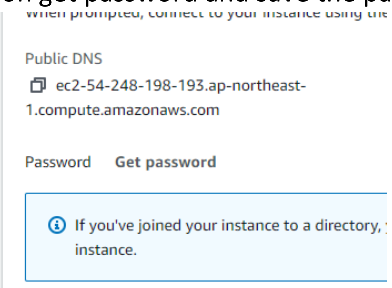


EC2 > Instances > i-036223d7ced32f9d8 > Connect to instance

Connect to instance Info

Connect to your instance i-036223d7ced32f9d8 (Windows-Server) using any of these

11.Click on get password and save the pass in your notepad



when prompted, connect to your instance using the

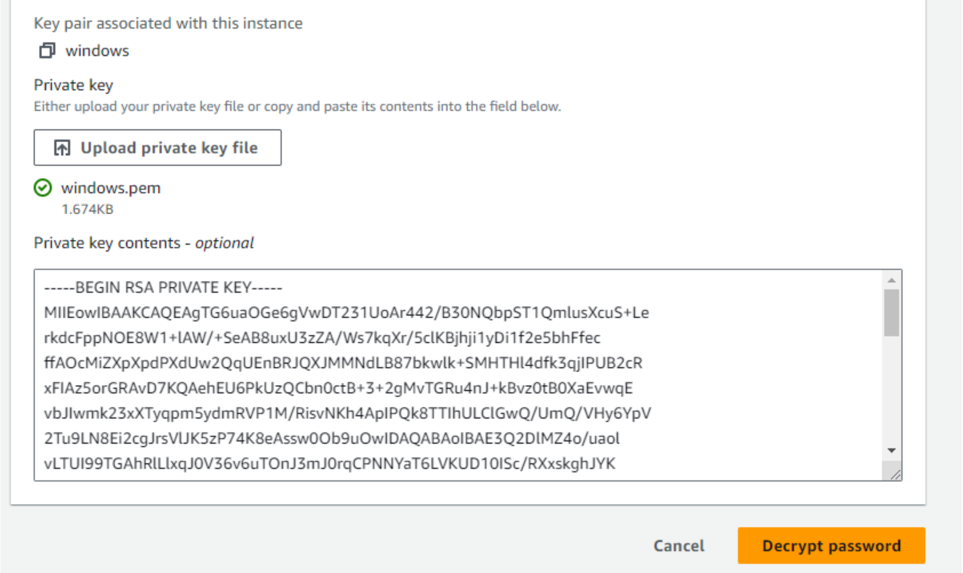
Public DNS

ec2-54-248-198-193.ap-northeast-1.compute.amazonaws.com

Password Get password

ⓘ If you've joined your instance to a directory, instance.

12.Upload key value file which you have downloaded before in .pem extension and decrypt it



Key pair associated with this instance

windows

Private key

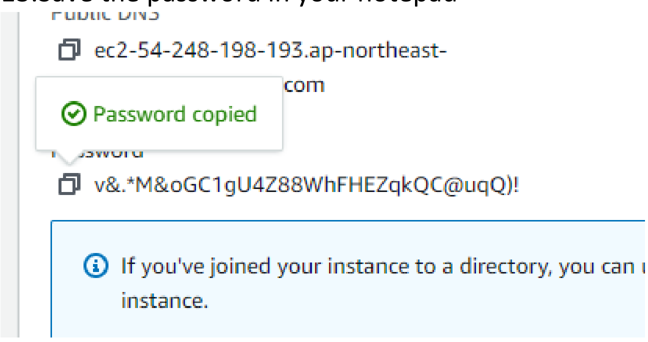
Either upload your private key file or copy and paste its contents into the field below.

✓ windows.pem
1.674KB

Private key contents - *optional*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAgTG6uaOGe6gVwDT231UoAr442/B30NQbpST1QmlusXcuS+Le
rkdcFppNOE8W1+lAW/+SeAB8uxU3zZA/Ws7kqXr/5clKBjhji1yDi1f2e5bhFfec
ffAOcMiZXpXpdPXdUw2QqUEnBRJQXJMMNdLB87bkwlk+SMHTHl4dfk3qjIPUB2cR
xFIAz5orGRAvD7KQAehEU6PkUzQCbn0ctB+3+2gMvTGRu4nJ+kBvz0tB0XaEvwqE
vbJlwmk23xXTyqpm5ydmRVP1M/RisvNKh4ApIPQk8TTIhULClGwQ/UmQ/VHy6YpV
2Tu9LN8Ei2cgJrsVlJK5zP74K8eAssw0Ob9uOwIDAQABAoIBAE3Q2DlMZ4o/uaol
vLTUI99TGAhRlLlxqJ0V36v6uTOnJ3mJ0rqCPNNYaT6LVKUD10ISc/RXxskghJYK
```

13.Save the password in your notepad



ec2-54-248-198-193.ap-northeast-

com

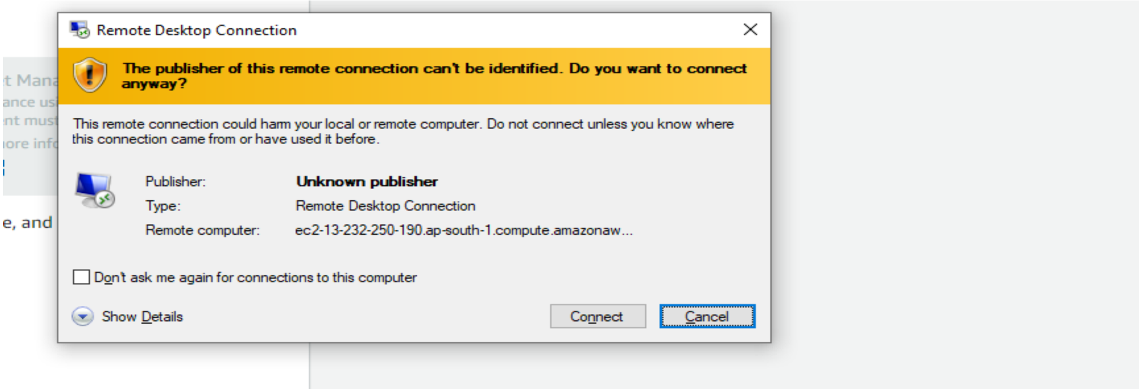
v&.*M&oGC1gU4Z88WhFHEZqkQC@uqQ)!

If you've joined your instance to a directory, you can

instance.

14.Go back to Instance , connect then download RDP file

15.Open RDP



Remote Desktop Connection

The publisher of this remote connection can't be identified. Do you want to connect anyway?

This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

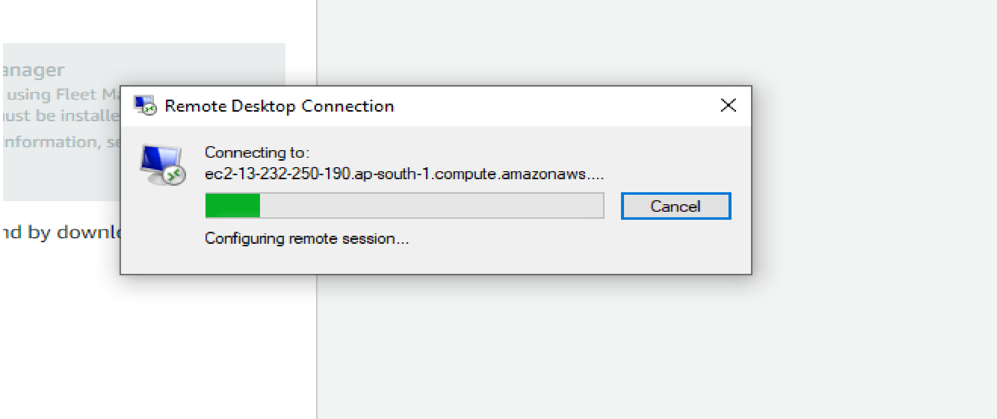
Publisher: **Unknown publisher**

Type: Remote Desktop Connection

Remote computer: ec2-13-232-250-190.ap-south-1.compute.amazonaw...

□ Don't ask me again for connections to this computer

16.Connect RDP :

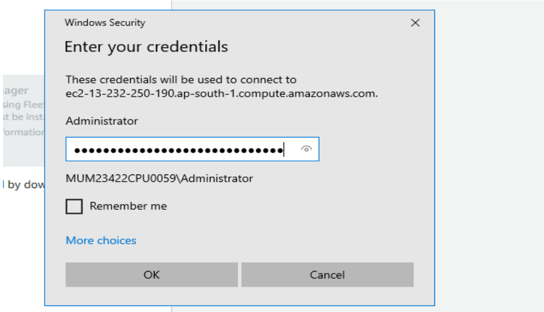


Remote Desktop Connection

Connecting to:
ec2-13-232-250-190.ap-south-1.compute.amazonaws....

Configuring remote session...

17.Enter Credentials that you have copied in step 13 and click on ok window machine will get open



Windows Security

Enter your credentials

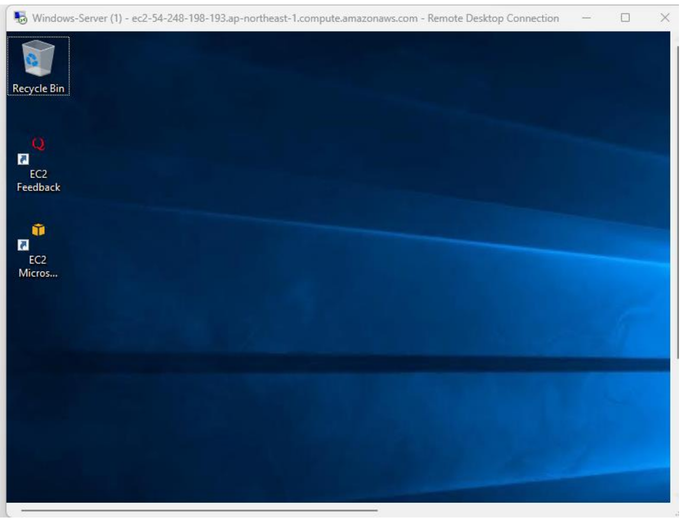
These credentials will be used to connect to ec2-13-232-250-190.ap-south-1.compute.amazonaws.com.

Administrator

••••••••••••••••••••••••••••••

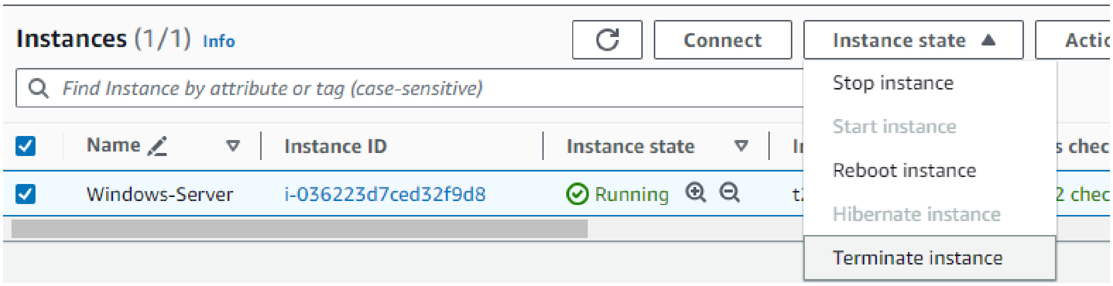
MUM23422CPU0059\Administrator

□ Remember me



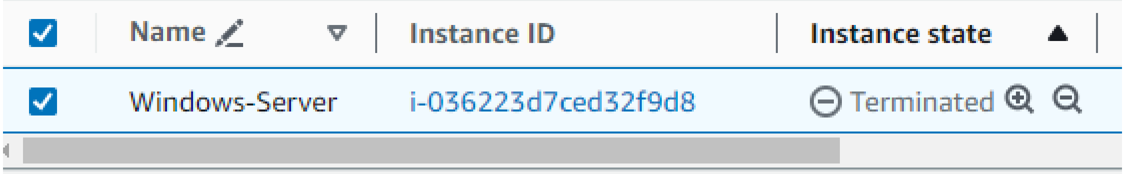
Windows-Server (1) - ec2-54-248-198-193.ap-northeast-1.compute.amazonaws.com - Remote Desktop Connection

18.Close RDP
19.Go back to instances
20.Terminate the instance



Instances (1/1) Info

☑	Name	Instance ID	Instance state	In	s chec
☑	Windows-Server	i-036223d7ced32f9d8	✓ Running	t2	2 chec

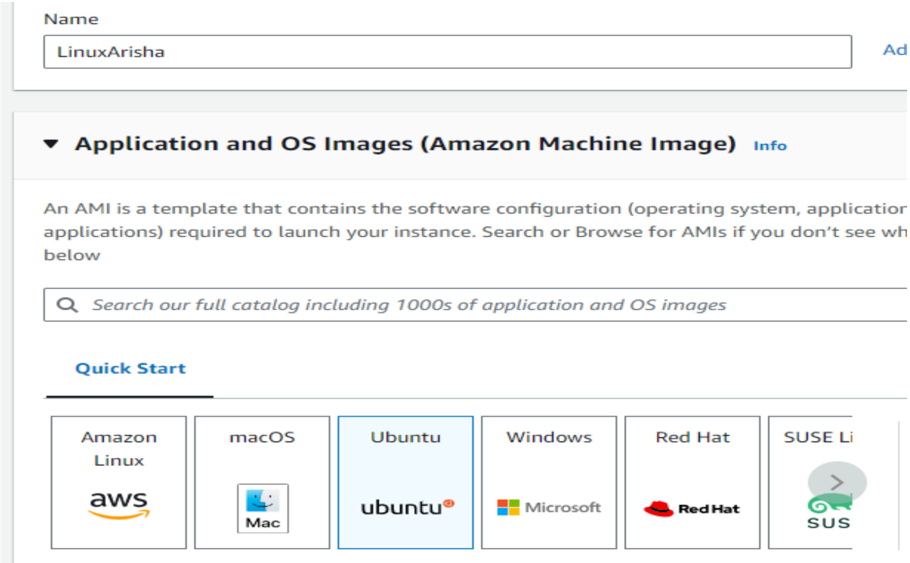


☑	Name	Instance ID	Instance state ▲
☑	Windows-Server	i-036223d7ced32f9d8	⊖ Terminated

2.Implement Ubuntu machine using AWS ec2 and execute the Linux commands.

- **Disk information in human readable format**
- **Create a folder with your name**
- **Create a file with your cityname and add your address in it**
- **Display the created file**
- **Copy the contents of the created file in other file and print it**
- **Install firefox/python 3**

1.Write a new web server name and Select Ubuntu server

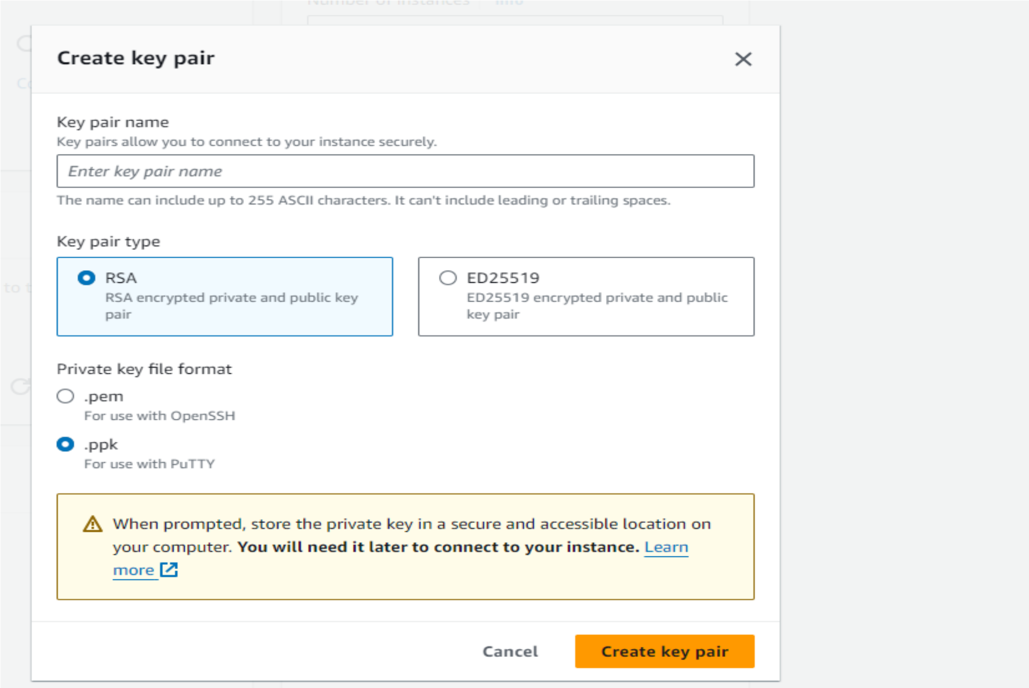


Name: LinuxArisha

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, applicatior applications) required to launch your instance. Search or Browse for AMIs if you don't see wh below

2.Create a new key value pair and select .ppk file extension for Ubuntu and save it



Create key pair ✕

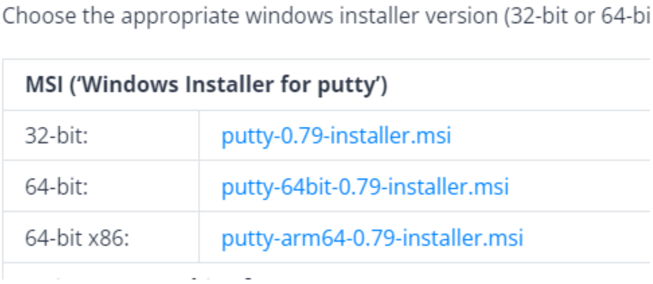
Key pair name
Key pairs allow you to connect to your instance securely.
Enter key pair name
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type
● RSA
RSA encrypted private and public key pair
○ ED25519
ED25519 encrypted private and public key pair

Private key file format
○ .pem
For use with OpenSSH
● .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

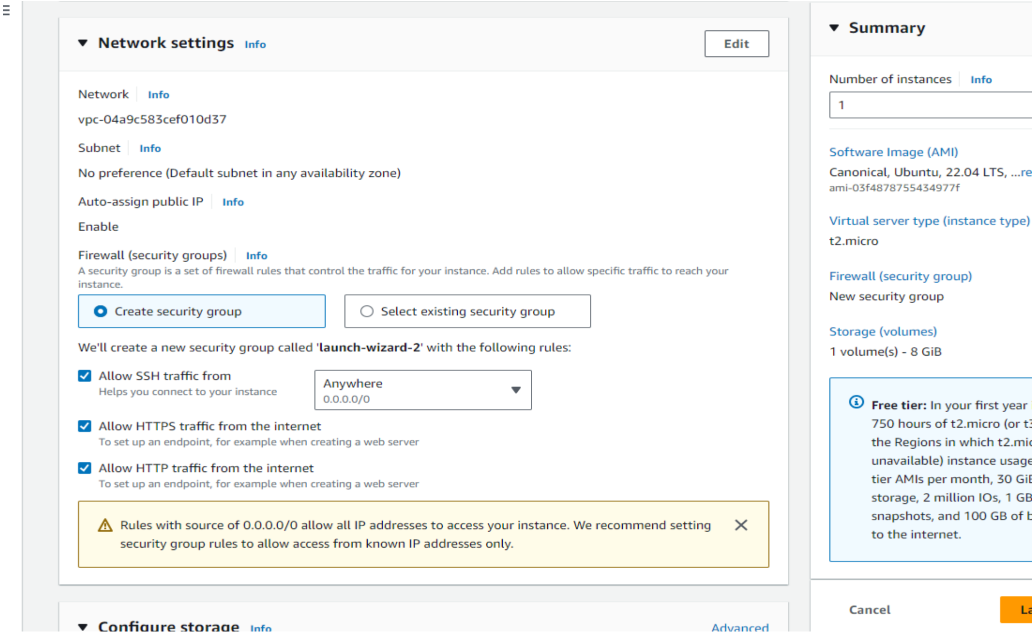
3.Download putty.exe from google [link to download]



Choose the appropriate windows installer version (32-bit or 64-bi

MSI ('Windows Installer for putty')	
32-bit:	putty-0.79-installer.msi
64-bit:	putty-64bit-0.79-installer.msi
64-bit x86:	putty-arm64-0.79-installer.msi

4.Allow all traffic



Network settings Info

Network Info
vpc-04a9c583cef010d37

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...re
ami-03f4878755434977f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

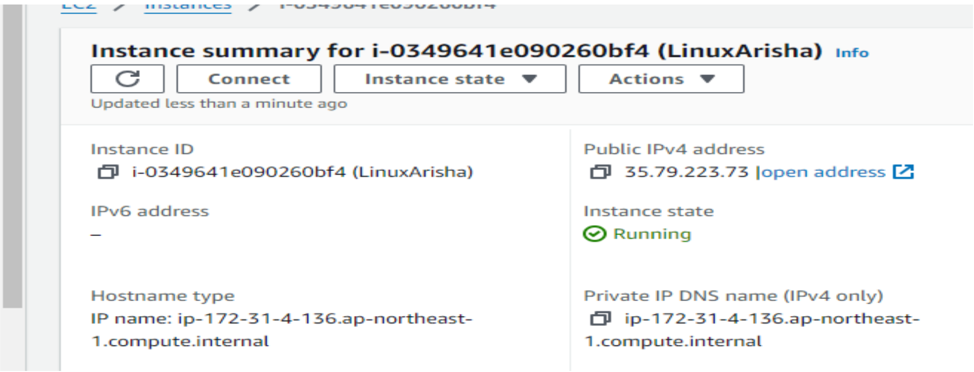
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year
750 hours of t2.micro (or t3
the Regions in which t2.mic
unavailable) instance usage
tier AMIs per month, 30 GiB
storage, 2 million IOs, 1 GB
snapshots, and 100 GB of b
to the internet.

5.Launch Instance

6.Go to Instances and refresh

7.Select and copy Public IPV4 address



Instance summary for i-0349641e090260bf4 (LinuxArisha)

Updated less than a minute ago

Instance ID: i-0349641e090260bf4 (LinuxArisha)

Public IPv4 address: 35.79.223.73 |open address

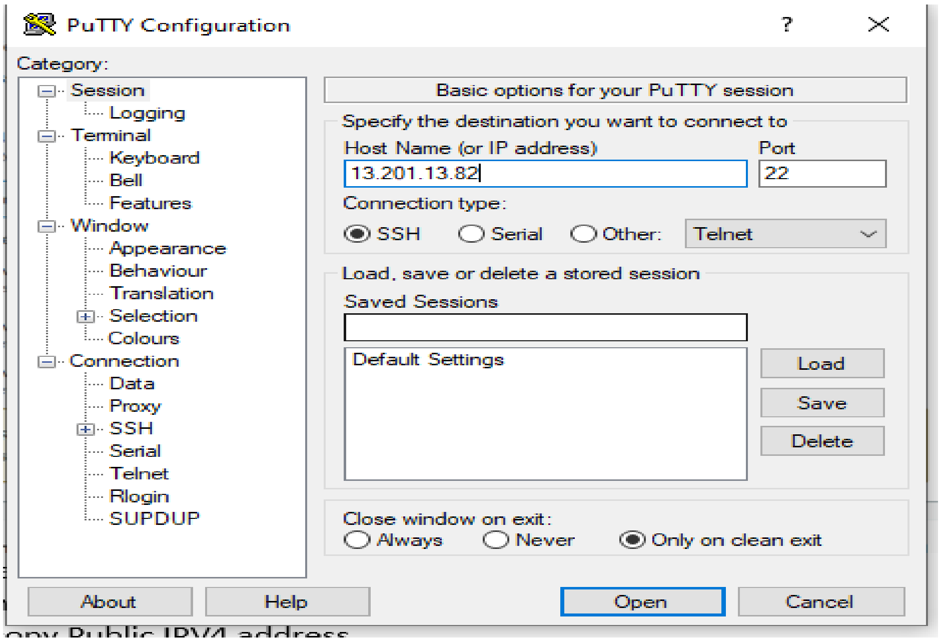
IPv6 address: –

Instance state: Running

Hostname type: IP name: ip-172-31-4-136.ap-northeast-1.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-4-136.ap-northeast-1.compute.internal

8.Go to putty, paste IP address



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 13.201.13.82

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

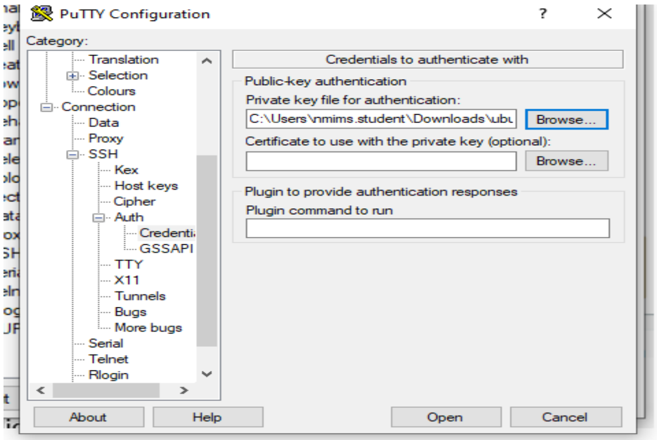
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

9.Category -> SSH -> Auth -> Credentials -> Browse and select ppk file



PuTTY Configuration

Credentials to authenticate with

Public-key authentication

Private key file for authentication: C:\Users\nmims.student\Downloads\ubu

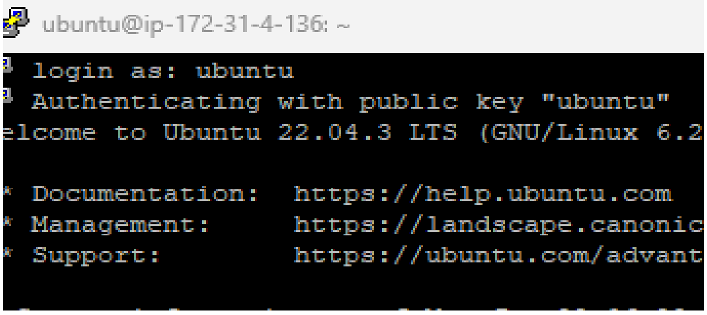
Certificate to use with the private key (optional):

Plugin to provide authentication responses

Plugin command to run

10.Putty will launch

11.Login as Ubuntu



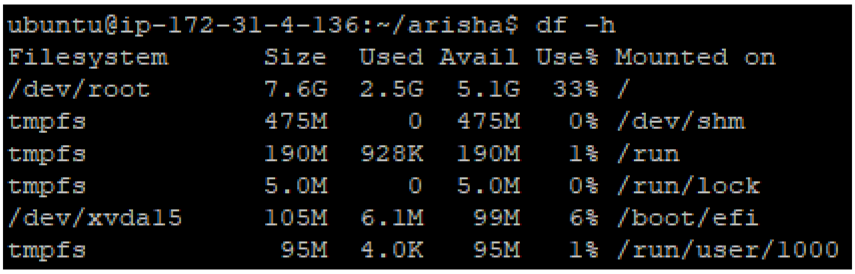
```
ubuntu@ip-172-31-4-136: ~

 login as: ubuntu
 Authenticating with public key "ubuntu"
elcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonic
* Support:        https://ubuntu.com/advant
```

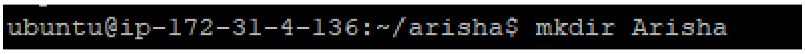
12.Type commands-

1.Disk information in human readable format



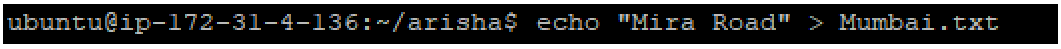
```
ubuntu@ip-172-31-4-136:~/arisha$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.6G  2.5G  5.1G  33% /
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  928K  190M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            95M  4.0K   95M   1% /run/user/1000
```

2.Create a folder with your name



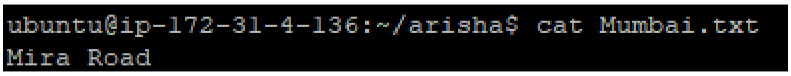
```
ubuntu@ip-172-31-4-136:~/arisha$ mkdir Arisha
```

3.Create a file with your cityname and add your address in it



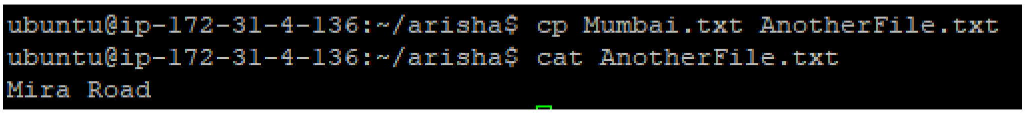
```
ubuntu@ip-172-31-4-136:~/arisha$ echo "Mira Road" > Mumbai.txt
```

4.Display the created file



```
ubuntu@ip-172-31-4-136:~/arisha$ cat Mumbai.txt
Mira Road
```

5.Copy the contents of the created file in other file and print it

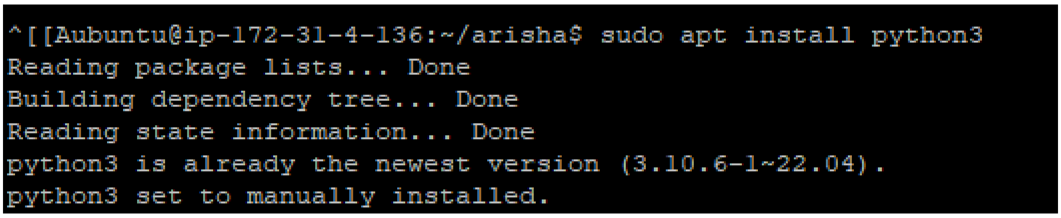


```
ubuntu@ip-172-31-4-136:~/arisha$ cp Mumbai.txt AnotherFile.txt
ubuntu@ip-172-31-4-136:~/arisha$ cat AnotherFile.txt
Mira Road
```

control + D for exit

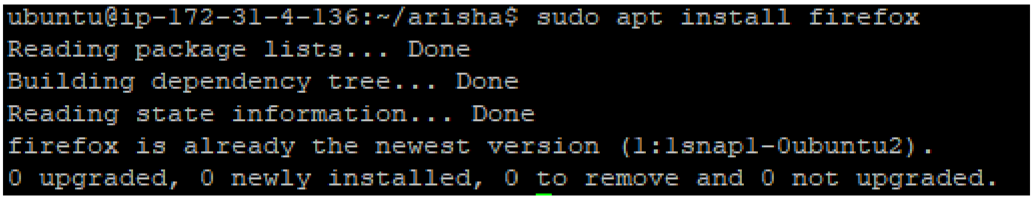
6.To Install Python3 and Firefox commands used are as follows:

1.sudo apt install python3



```
^[[Aubuntu@ip-172-31-4-136:~/arisha$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
```

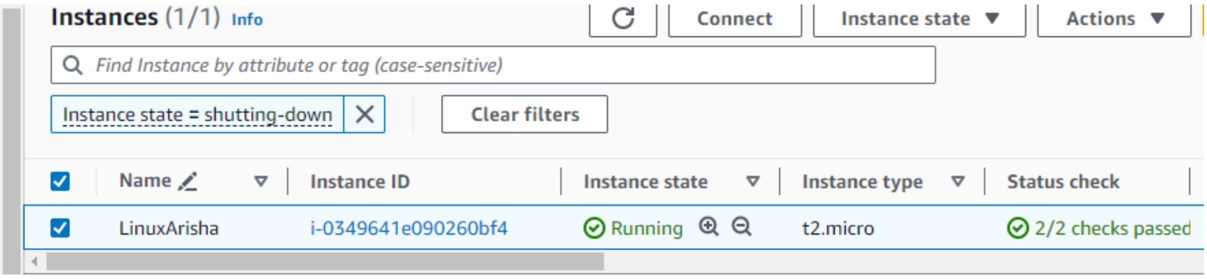
2. sudo apt install firefox



```
ubuntu@ip-172-31-4-136:~/arisha$ sudo apt install firefox
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
firefox is already the newest version (1:1snap1-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

13.Type exit command to exit putty

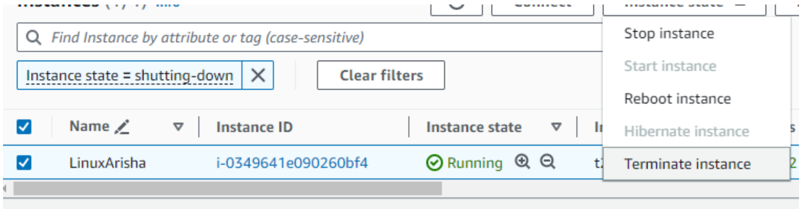
14.Go back to instances in AWS and select the current instance



Instance state = shutting-down

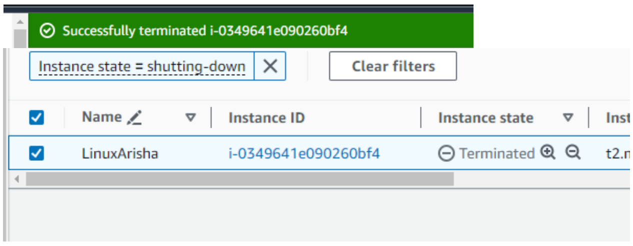
☑	Name	Instance ID	Instance state	Instance type	Status check
☑	LinuxArisha	i-0349641e090260bf4	Running	t2.micro	2/2 checks passed

15.TO TERMINATE THE CURRENT RUNNING INSTANCE:
Instances ->Select Instance -> Instance state -> Terminate the Instance



Instance state = shutting-down

☑	Name	Instance ID	Instance state
☑	LinuxArisha	i-0349641e090260bf4	Running



Successfully terminated i-0349641e090260bf4

Instance state = shutting-down

☑	Name	Instance ID	Instance state	Inst
☑	LinuxArisha	i-0349641e090260bf4	Terminated	t2.r