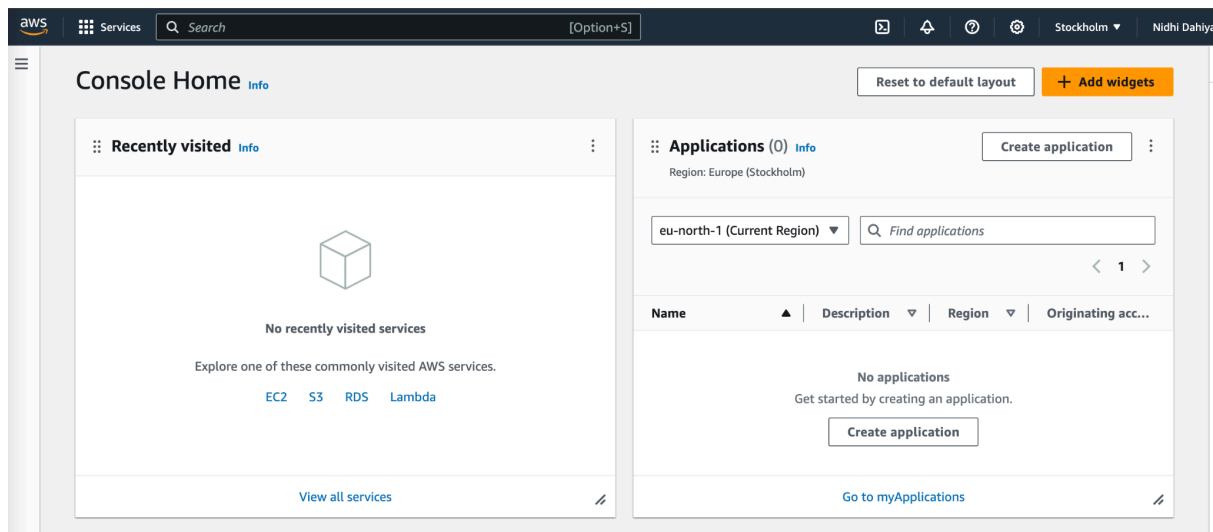


Cloud Computing

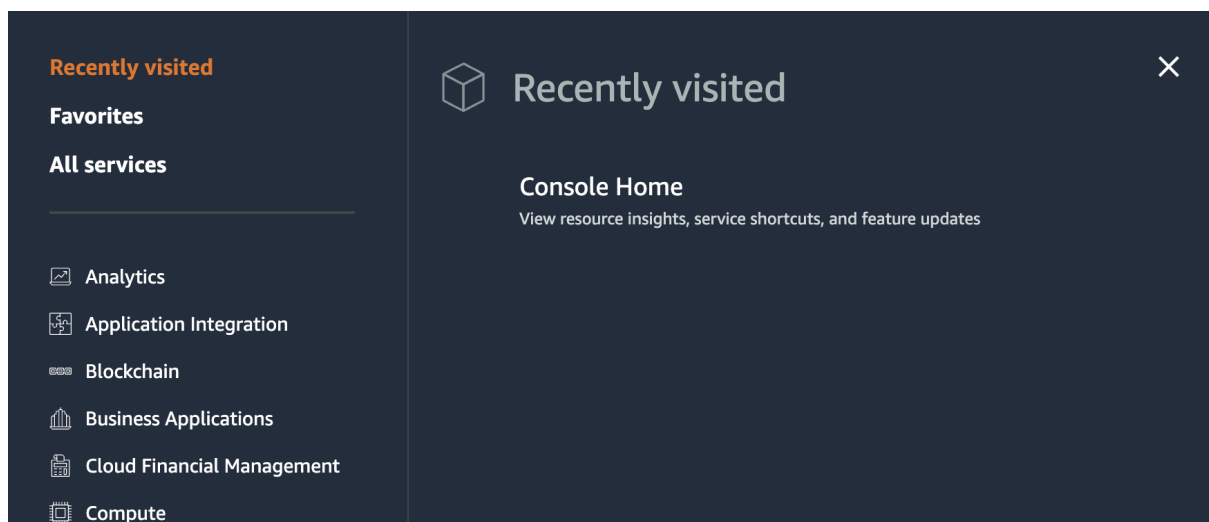
Prac 1: Infrastructure as a service using AWS.

Implement the windows machine using AWS EC2

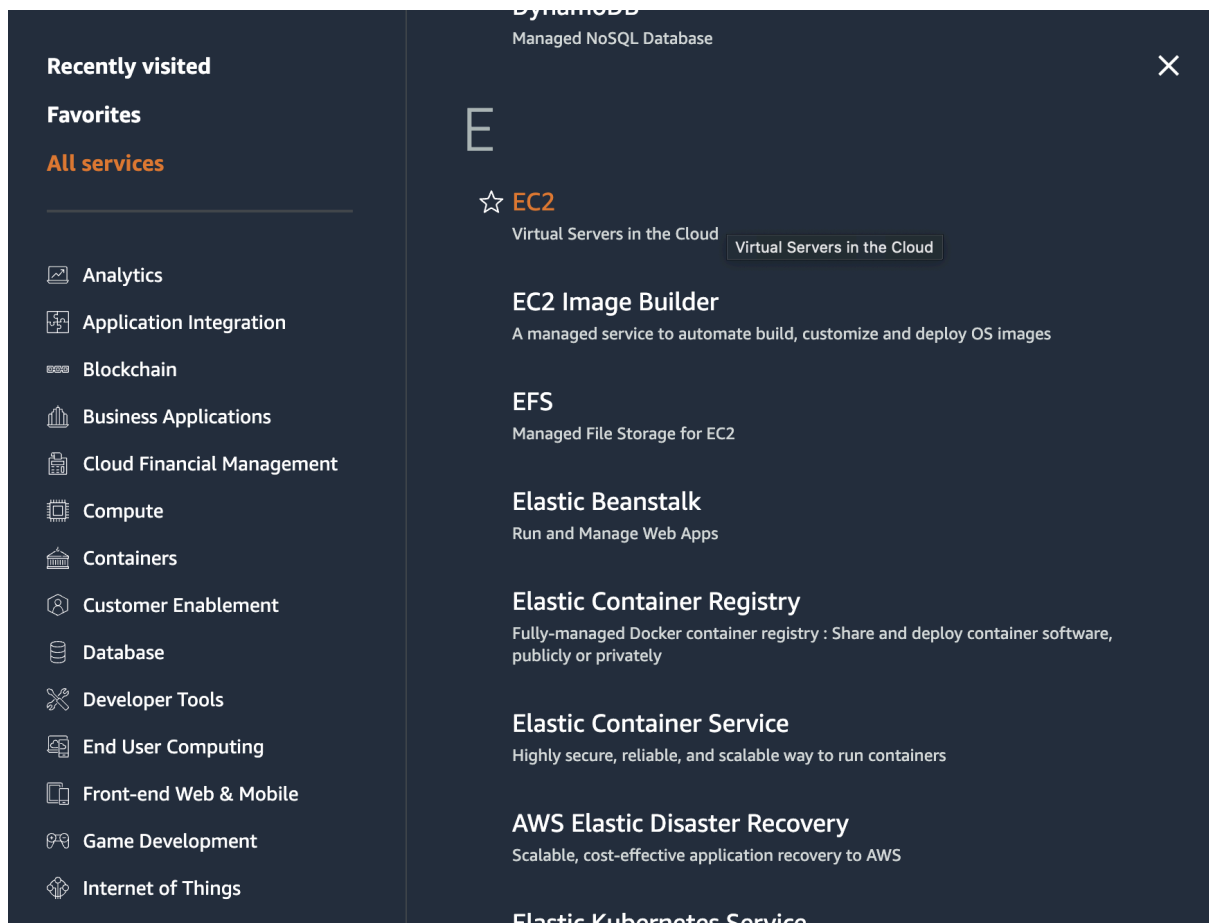
1. Sign in to your AWS account.



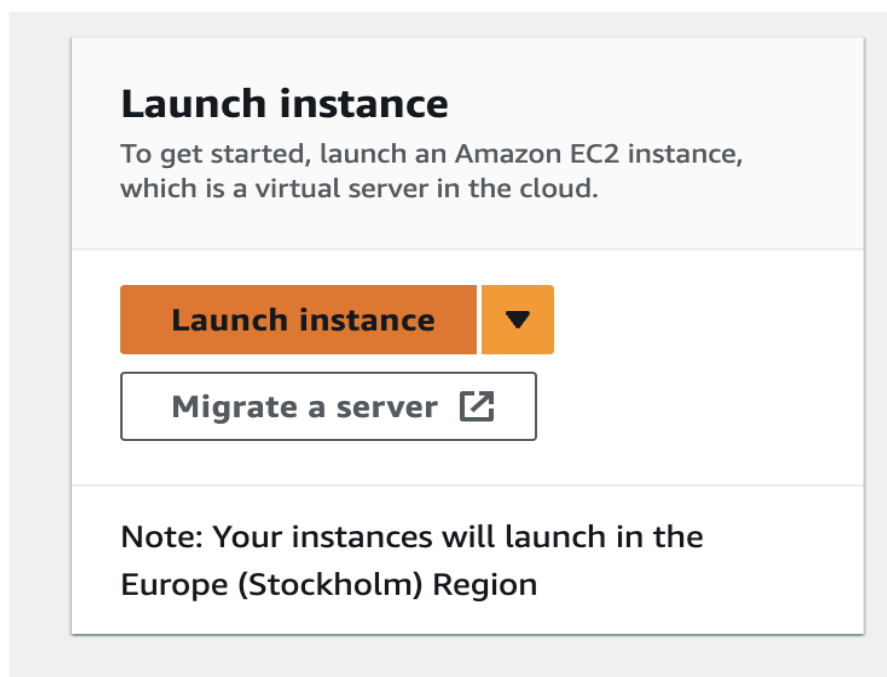
2. Select All Services.



3. Select EC2.



4. Launch Instance



5. Put the name and select Windows.

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)


Name

[Add additional tags](#)


▼ Application and OS Images (Amazon Machine Image) [Info](#)

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


Quick Start




Amazon Linux




macOS




Ubuntu




Windows



Red Hat



SUSE Linux



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

6. Create key value pair.

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

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

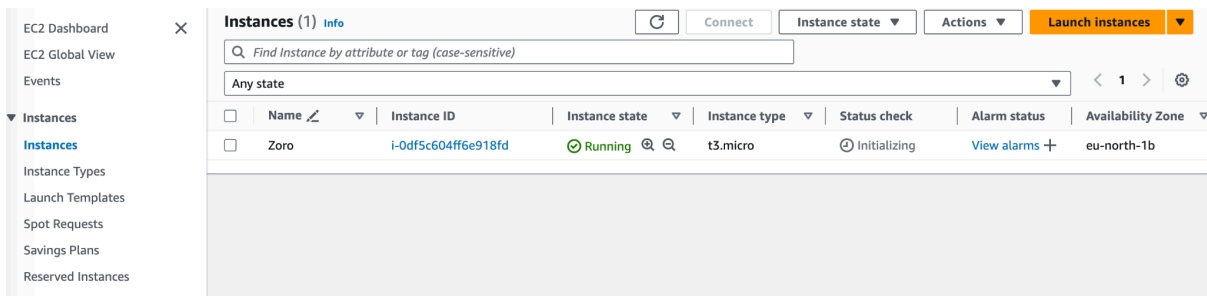
☐ .ppk
For use with PuTTY

 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](#)

Cancel Create key pair

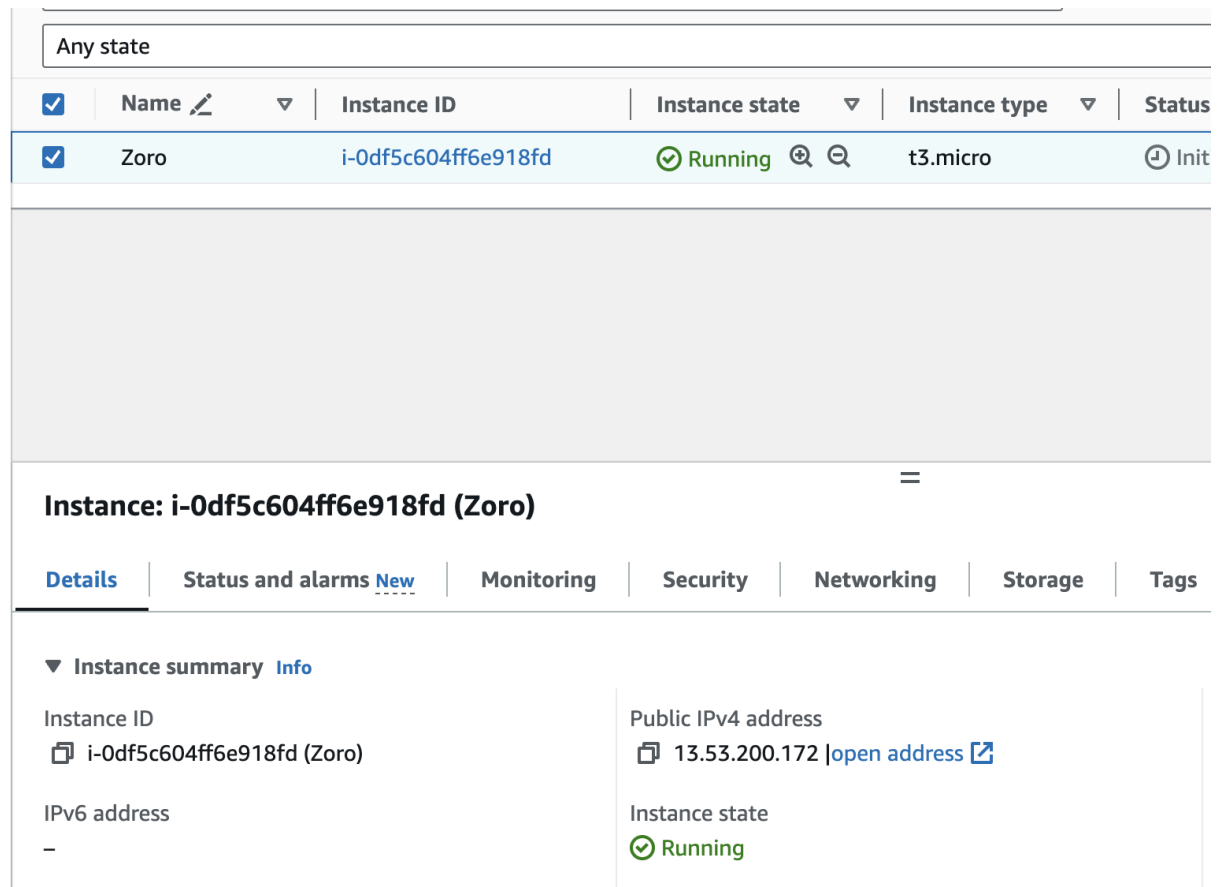
7. Launch the Instance.

8. Go to the instances tab to check the initialization.



9. It will initialise and then start running.

10. Select the instance.



11. Select "Connect" to establish a connection with the RDP client.

12. Click on the get password.

Connect to instance Info

Connect to your instance i-0df5c604ff6e918fd (Zoro) using any of these options

Session Manager
RDP client
EC2 serial console

Instance ID
i-0df5c604ff6e918fd (Zoro)

Connection Type

☒ **Connect using RDP client**
Download a file to use with your RDP client and retrieve your password.

☐ **Connect using Fleet Manager**
To connect to the instance using Fleet Manager Remote Desktop, the SSM Agent must be installed and running on the instance. For more information, see [Working with SSM Agent](#)

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

Download remote desktop file

When prompted, connect to your instance using the following details:

| | |
|--|---------------------------|
| Public DNS ec2-13-53-200-172.eu-north-1.compute.amazonaws.com | Username Administrator |
|--|---------------------------|

13. Upload the key value file and Decrypt password.

Get Windows password Info

Use your private key to retrieve and decrypt the initial Windows administrator password for this instance.

Instance ID
i-0df5c604ff6e918fd (Zoro)

Key pair associated with this instance
maya

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

Upload private key file

☒ maya.pem
1.678KB

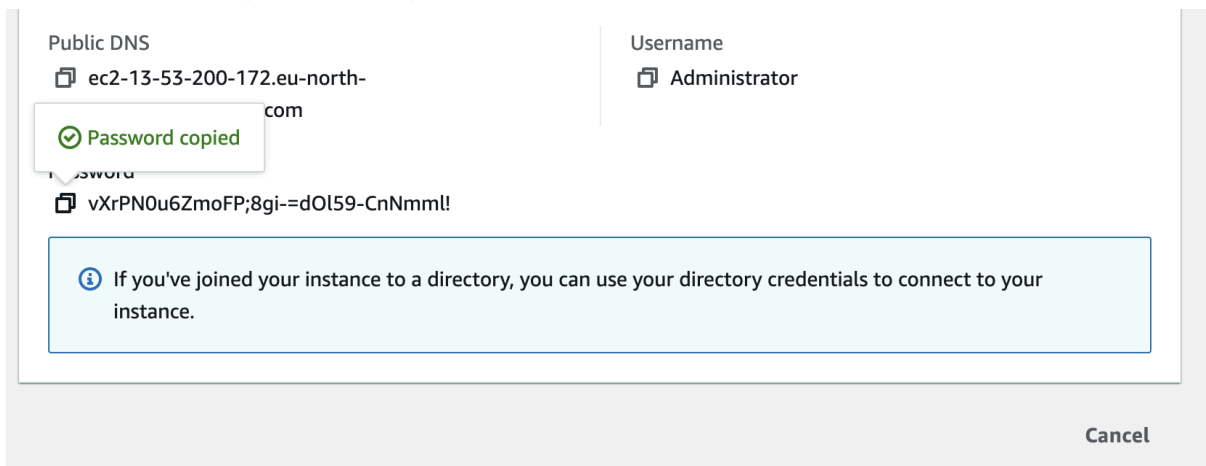
Private key contents - *optional*

```

-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAww9vhKGFyPZ429wbFgzSzhXeGo49BI2RevEahcbIHklqPtFH
p080vMD8QS/27H0d6PaMeUHa2JZxRRReV3FnbTpBkSCYDQmpX20UHlu+n+IH9bTHT
vA/lqFEmB00LalGo3Q3eaSQ4oYcKXPSr7ILnSD8qC5LGrbL6zsW9H4uhy3+/upu
uG1KsLjZCqElrZ0Ab690PcuWWFOx3y1KBqZ+RXrt4dfh7HR+/zpp2d8dh6E25A54
LQnly6BsOUhFbrCEfepvgk5W77+IEPBjql72iqPnx38QXYfuW0U3aQOen1J2At9G
pPHz/TYcukkPWewx4t2olbMNL1erLQQ6dcjmnQIDAQABAolBAHpTkqGiUrgHo1Qk
zYKKzYaxxkzX0YaX6CFER6aHZ6QX4epFnjdpuxhkGmg11zgUU76raP92ygR8c9wt

```

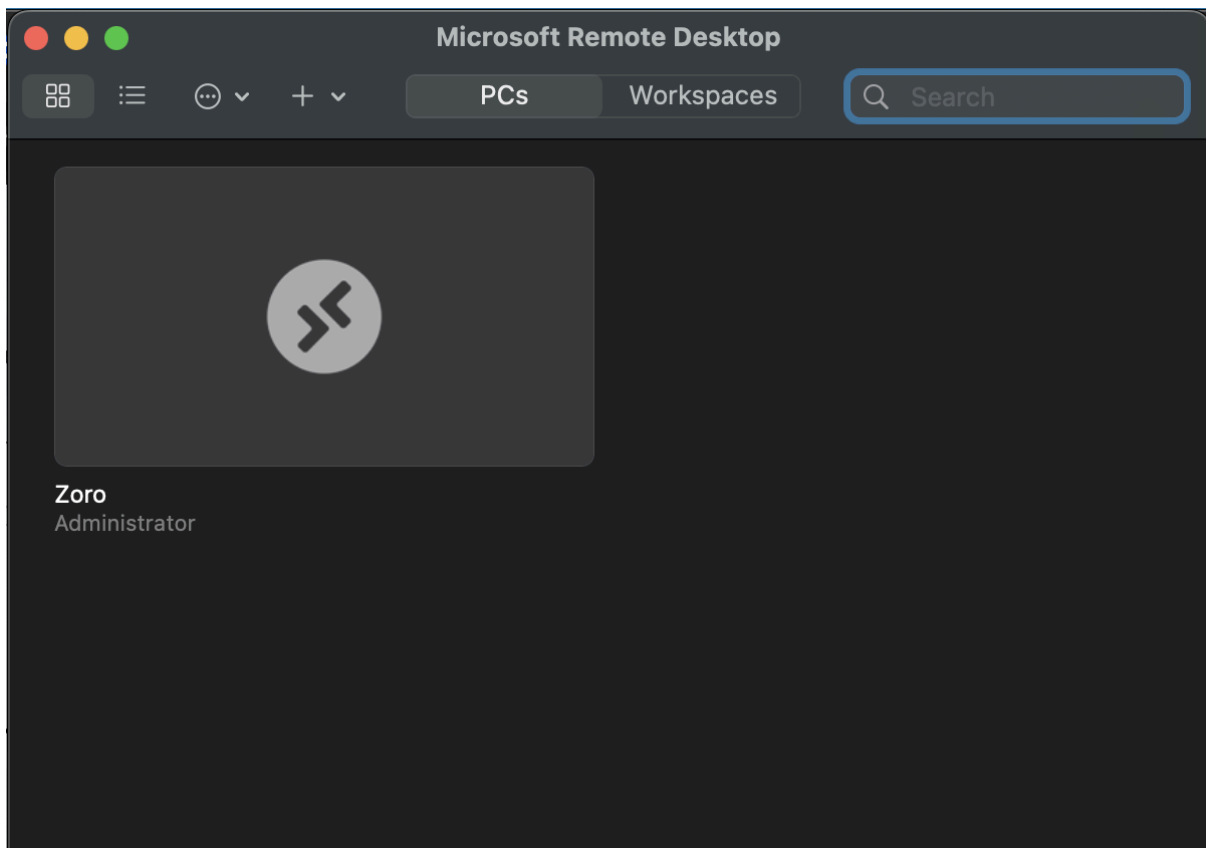
14. Save the password (vXrPN0u6ZmoFP;8gi==dOl59-CnNmml!)



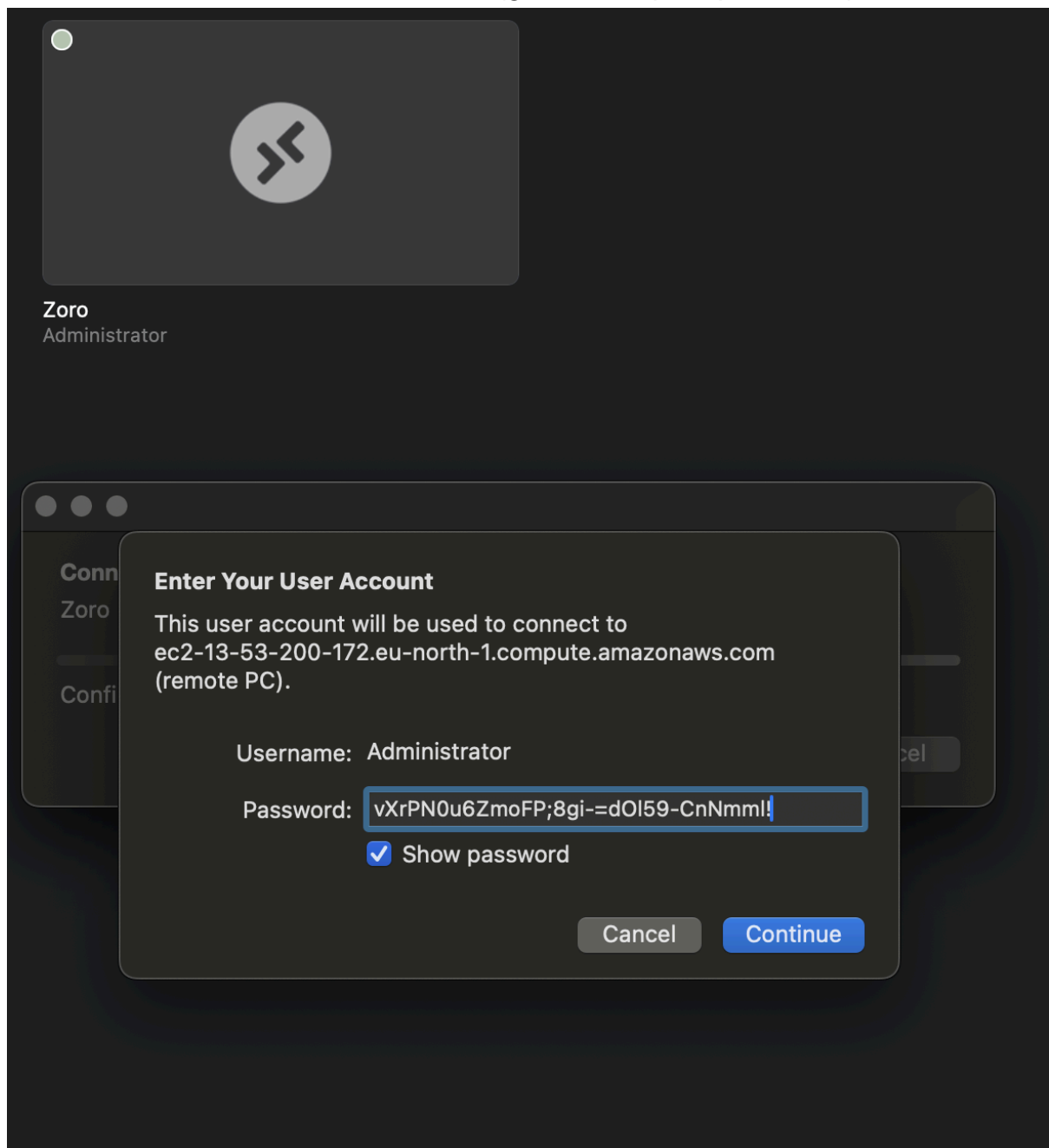
The screenshot shows the 'Connect to Instance' dialog box in the AWS Management Console. It has two main sections: 'Public DNS' and 'Username'. The 'Public DNS' section shows the instance ID 'ec2-13-53-200-172.eu-north-1' and a dropdown menu for the 'Password' field, which is currently set to 'vXrPN0u6ZmoFP;8gi==dOl59-CnNmml!'. A green tooltip with a checkmark and the text 'Password copied' is visible over the password field. The 'Username' section shows a dropdown menu set to 'Administrator'. Below these fields is a light blue informational box with an information icon and the text: 'If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.' At the bottom right of the dialog is a 'Cancel' button.

15. Go back to the instance, select the instance, connect, and download the RDP file.

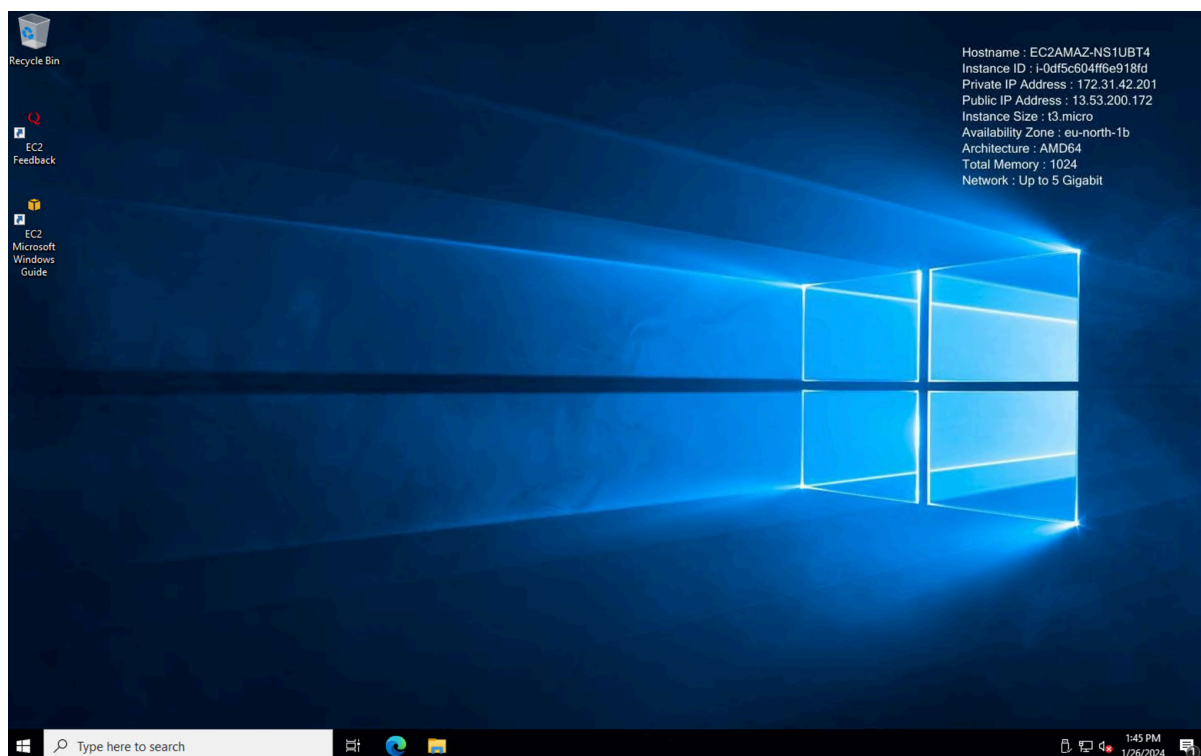
16. Open downloaded RDP file.



17. Connect, enter the credentials (give the copied password) and continue.

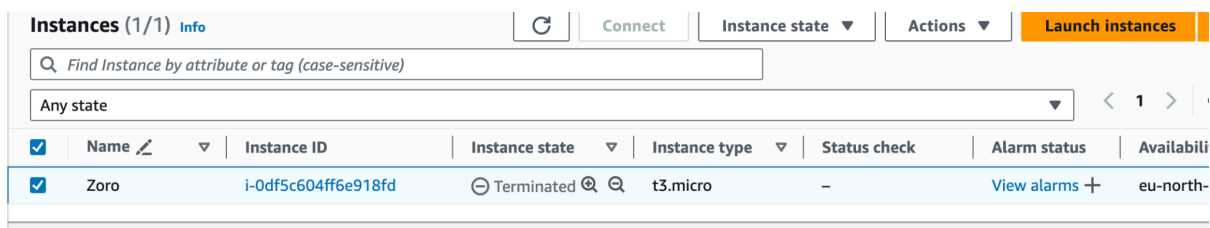
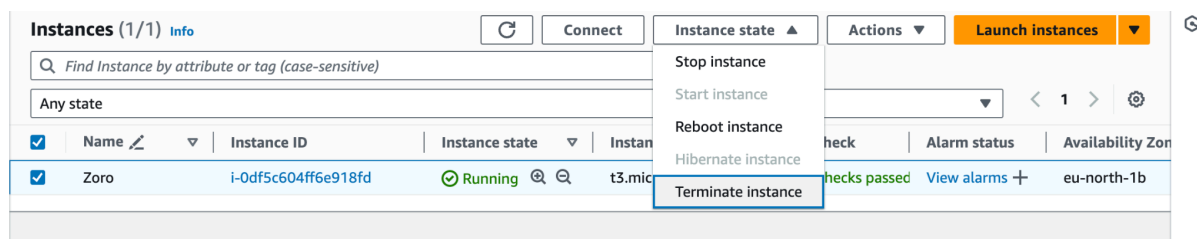


18. Windows.



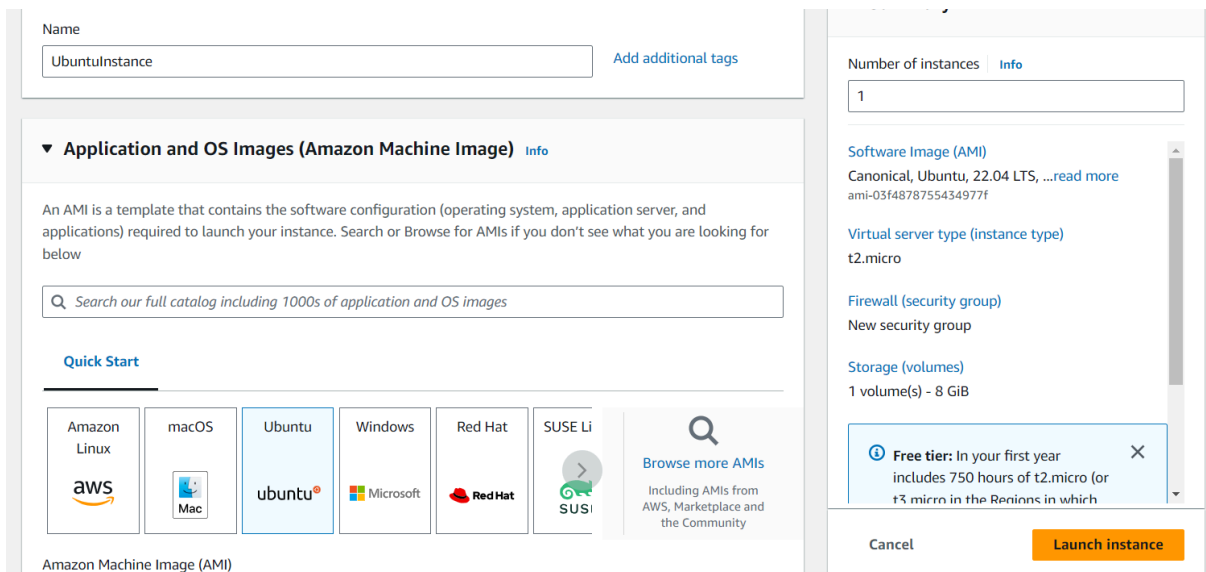
19. Close RDP.

20. Go back to Instances and terminate.

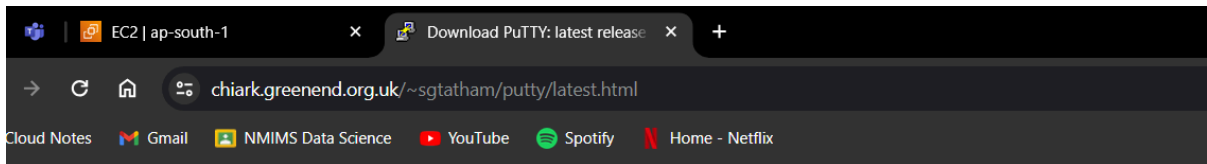


Implement Ubuntu machine using AWS EC2 and execute the Linux commands

1. Launch a new instance for Linux.
2. Repeat the same steps as for Windows but this time, choose Ubuntu as the OS Image.



3. Download putty.exe from the link below.



Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel and pterm), but you can download standalone binaries (Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)

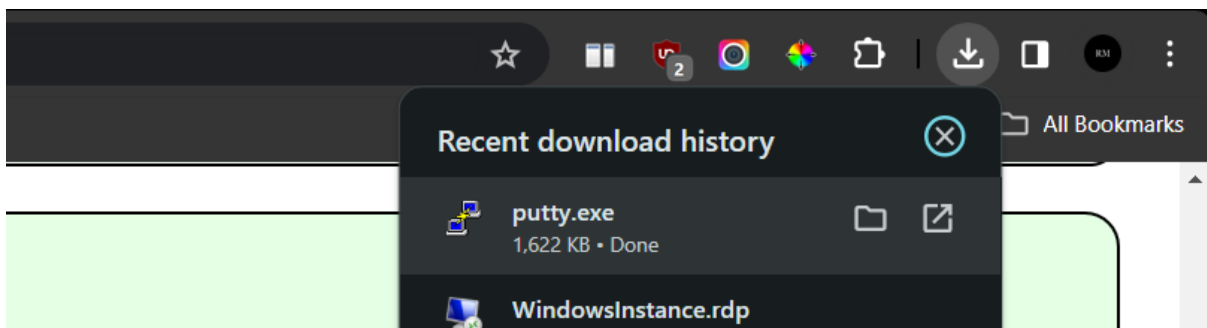
64-bit x86: [putty.exe](#) ([signature](#))
 64-bit Arm: [putty.exe](#) ([signature](#))
 32-bit x86: [putty.exe](#) ([signature](#))

pscp.exe (an SCP client, i.e. command-line secure file copy)

64-bit x86: [pscp.exe](#) ([signature](#))
 64-bit Arm: [pscp.exe](#) ([signature](#))
 32-bit x86: [pscp.exe](#) ([signature](#))

psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)

64-bit x86: [psftp.exe](#) ([signature](#))
 64-bit Arm: [psftp.exe](#) ([signature](#))
 32-bit x86: [psftp.exe](#) ([signature](#))



4. Create a new key pair for this instance and download the .ppk file.

5. Tick all the boxes for the traffic permissions in network settings.

▼ Network settings

Info

Edit

Network

Info

vpc-01cbdd6d2e35b52c0

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

Helps you connect to your instance

Anywhere
0.0.0.0/0

☒ 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

▼ Summary

Number of instances

Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

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 includes 750 hours of t2.micro (or t3.micro in the Regions in which

Cancel

Launch instance

- Launch the instance and in the instance summary, copy the public IPv4 address.

Instance: i-0f78f6b4231a898a3 (UbuntuInstance)

Details | Status and alarms [New](#) | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary [Info](#)

Instance ID
i-0f78f6b4231a898a3 (UbuntuInstance)

IPv6 address
-

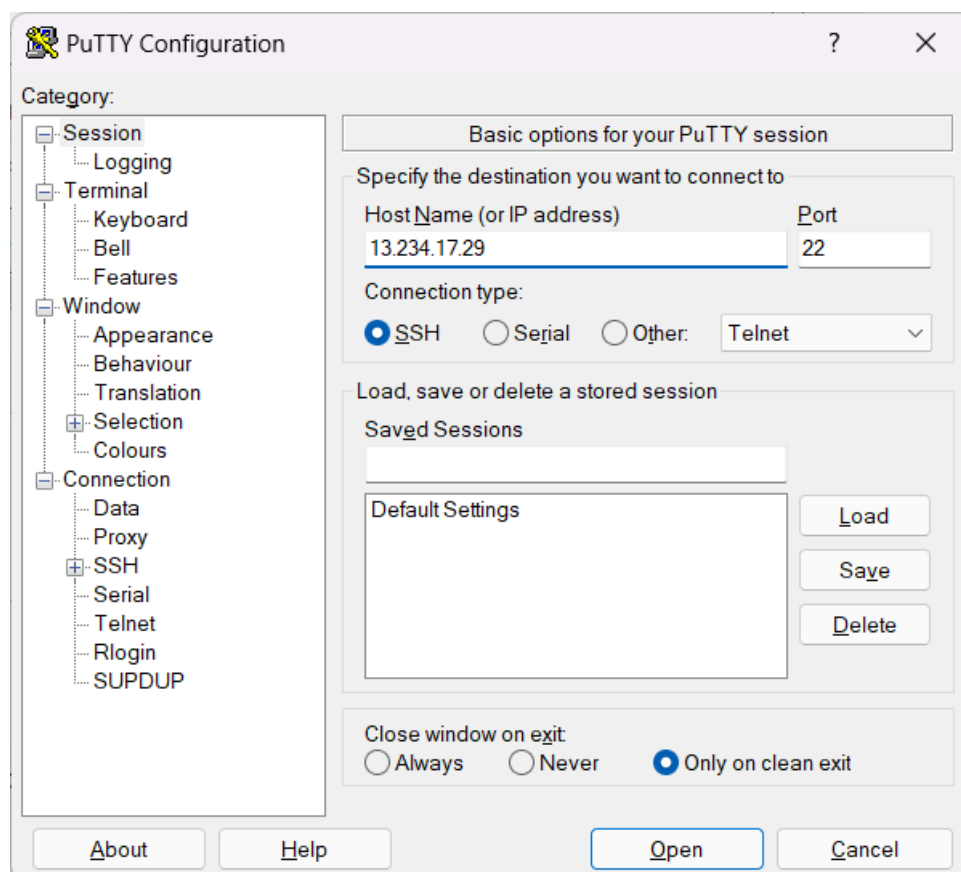
Public IPv4 address copied

13.234.17.29 [open address](#)

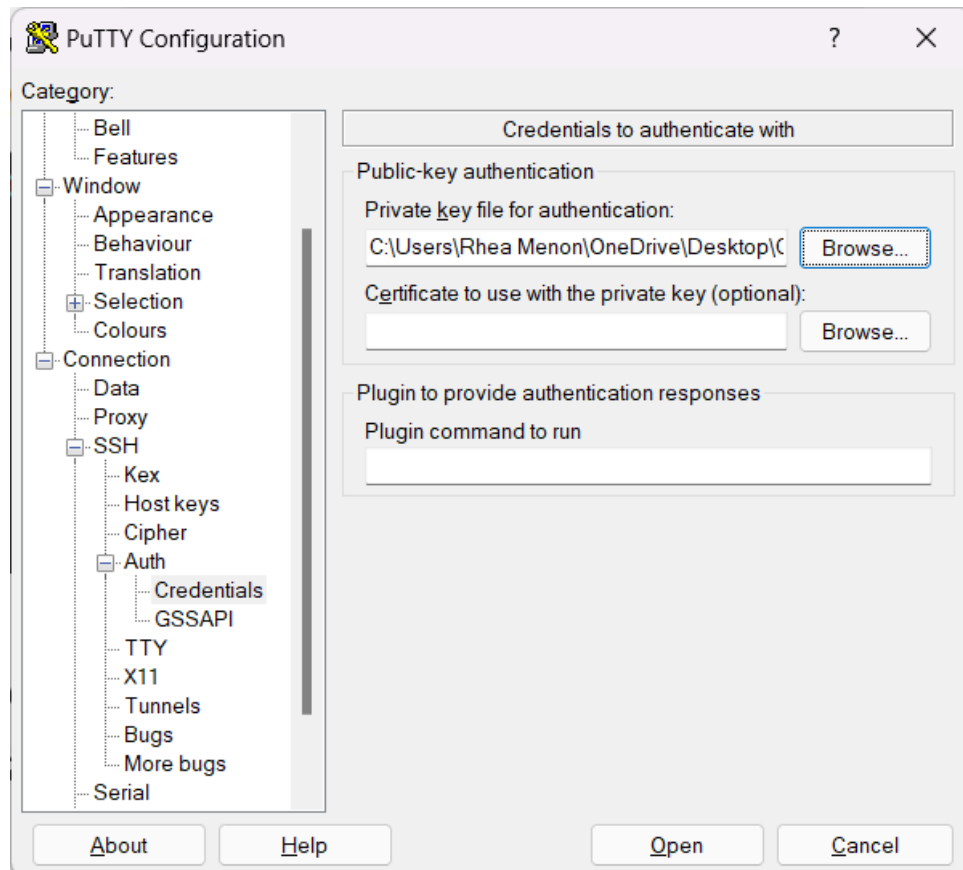
Instance state
Running

Priv
Pub
1.cc

- Open PuTTY and add the copied IP address to the Host IP address text box.



- Go to Connection > SSH > Auth > Credentials and upload the .ppk key pair file for the private key.



- Click open and the ubuntu console will be launched. Login as ubuntu.

```
ubuntu@ip-172-31-32-198: ~
login as: ubuntu
Authenticating with public key "LinuxKeyPair"
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Mon Jan 29 16:20:22 UTC 2024

System load:  0.30908203125   Processes:            102
Usage of /:   20.6% of 7.57GB   Users logged in:     0
Memory usage: 21%            IPv4 address for eth0: 172.31.32.198
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
```

Commands

- `mkdir [directory name]`: makes a directory of the same name.
- `ls`: lists all the files and directories in the current working directory.

```
ubuntu@ip-172-31-32-198:~$ mkdir directory1
ubuntu@ip-172-31-32-198:~$ ls
directory1
ubuntu@ip-172-31-32-198:~$ mkdir directory2
ubuntu@ip-172-31-32-198:~$ ls
directory1 directory2
ubuntu@ip-172-31-32-198:~$
```

- `cat > [file name]`: append text to a file.
- `cat [file name]`: view the contents of a text file.

```
ubuntu@ip-172-31-32-198:~$ cat > testtext.txt
Say the name, seventeen
ubuntu@ip-172-31-32-198:~$ cat testtext.txt
Say the name, seventeen
ubuntu@ip-172-31-32-198:~$
```

- `sudo apt install [program name]`: installs the named program.

```
ubuntu@ip-172-31-32-198:~$ 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.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-32-198:~$
```

```
ubuntu@ip-172-31-32-198:~$ sudo apt install firefox
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 1315 (apt)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 1315 (apt)
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.
ubuntu@ip-172-31-32-198:~$
```

`exit`: closes the console.

- Terminate this instance as well using the same steps as before.

The screenshot shows the AWS Management Console 'Instances' page. The 'UbuntuInstance' (i-0f78f6b4231a898a3) is selected. The 'Instance state' dropdown menu is open, showing options: Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate instance. The 'Terminate instance' option is highlighted.

| Name | Instance ID | Instance state | Instance type |
|-----------------|---------------------|----------------|---------------|
| WindowsInstance | i-00bf94df158a3441d | Terminated | t2.micro |
| UbuntuInstance | i-0f78f6b4231a898a3 | Running | t2.micro |

The screenshot shows the AWS Management Console 'Instances' page after the 'UbuntuInstance' (i-0f78f6b4231a898a3) has been successfully terminated. A green banner at the top indicates 'Successfully terminated i-0f78f6b4231a898a3'. The instance state is now 'Shutting-down'.

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|-----------------|---------------------|----------------|---------------|--------------|-----------------------------|
| WindowsInstance | i-00bf94df158a3441d | Terminated | t2.micro | - | View alarms |
| UbuntuInstance | i-0f78f6b4231a898a3 | Shutting-down | t2.micro | - | View alarms |

Writeup

- Cloud Computing architecture:

The cloud architecture is divided into 2 parts i.e.

1. Frontend

Frontend of the cloud architecture refers to the client side of the 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.

- ☐ Application –
Application in backend refers to a software or platform to which a client accesses. Means it provides the service in the backend as per the client requirement.
- ☐ Service –
Service in backend refers to the major three types of cloud based services like SaaS, PaaS, IaaS. Also manages which type of service the user accesses.
- ☐ Runtime Cloud-
Runtime cloud in backend provides the execution and Runtime platform/environment to the Virtual machine.
- ☐ Storage –
Storage in the backend provides flexible and scalable storage service and management of stored data.
- ☐ Infrastructure –
Cloud Infrastructure in backend refers to the hardware and software

components of cloud like it includes servers, storage, network devices, virtualization software etc.

☐ Management –

Management in backend refers to management of backend components like application, service, runtime cloud, storage, infrastructure, and other security mechanisms etc.

☐ Security –

Security in the backend refers to implementation of different security mechanisms in the backend for secure cloud resources, systems, files, and infrastructure to end-users.

☐ Internet –

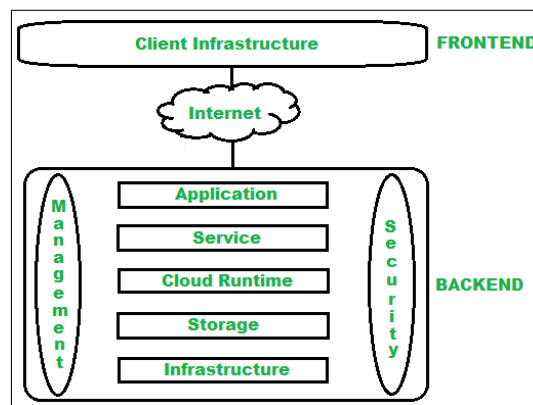
Internet connection acts as the medium or a bridge between frontend and backend and establishes the interaction and communication between frontend and backend.

☐ Database– Database in backend refers to a database for storing structured data, such as SQL and NOSQL databases. Examples of Database services include Amazon RDS, Microsoft Azure SQL database and Google CCloud SQL.

☐ Networking– Networking in backend services that provide networking infrastructure for applications in the cloud, such as load balancing, DNS and virtual private networks.

☐ Analytics– Analytics in backend service that provides analytics capabilities for data in the cloud, such as warehousing, business intelligence and machine learning.

The below figure represents an internal architectural view of cloud computing.



- IAAS:

Infrastructure as a service (IaaS) is a service model that delivers computer infrastructure on an outsourced basis to support various operations. Typically IaaS is a service where infrastructure is provided as outsourcing to enterprises such as networking equipment, devices, database, and web servers.

It is also known as Hardware as a Service (HaaS). IaaS customers pay on a per-user basis, typically by the hour, week, or month. Some providers also charge customers based on the amount of virtual machine space they use. It simply provides the underlying operating systems, security, networking, and servers for developing such applications, and services, and deploying development tools, databases, etc.

- AWS:

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and packaged-software-as-a-service (SaaS) offerings. AWS services can offer organisation tools such as compute power, database storage and content delivery services.

Amazon.com Web Services launched its first web services in 2002 from the internal infrastructure that Amazon.com built to handle its online retail operations. In 2006, it began offering its defining IaaS services. AWS was one of the first companies to introduce a pay-as-you-go cloud computing model that scales to provide users with compute, storage or throughput as needed. AWS offers many different tools and solutions for enterprises and software developers that can be used in data centres in up to 190 countries. Groups such as government agencies, education institutions, nonprofits and private organisations can use AWS services.

- AWS services:

1. Amazon EC2 (Elastic Cloud Compute)
2. Amazon RDS (Relational Database Services)
3. Amazon S3 (Simple Storage Service)
4. Amazon IAM (Identity and Access Management)
5. Amazon EBS (Elastic Block Store)
6. Amazon Lambda
7. Amazon EFS (Elastic File System)
8. Amazon CloudFront

9. Amazon SNS (Simple Notification Service)
10. Amazon VPC (Virtual Private Cloud)Amazon Auto-Scaling
11. Amazon Auto-Scaling

- EC2:

Amazon EC2 is the fastest cloud computing service provided by AWS. It offers virtual, secure, reliable, and resizable servers for any workload. Through this service, it becomes easy for developers to access resources and also facilitates web-scale cloud computing. This comes with the best suitable processors, networking facilities, and storage systems. Developers can quickly and dynamically scale capacities as per business needs. It has over 500 instances and you can also choose the latest processor, operating system, storage, and networking to help you choose according to the needs of the business. Also, with Amazon EC2, you only have to pay for what you use, and also as per the time period, scale with amazon EC2 auto-scaling has optimal storage and can optimise CPU configurations.