Sherin Ouseph
A017

Practical-1 Infrastructure as a service using AWS.

Writeup:-

- **Cloud Computing architecture**

  Cloud computing architecture comprises the following key components:

  1. Front End:
     - User Interface (UI)
     - Client (device)
  2. Back End:
     - Servers
     - Storage
     - Virtualization
     - Networking
  3. Middleware:
     - APIs
     - Runtime
  4. Cloud Services:
     - Infrastructure as a Service (IaaS)
     - Platform as a Service (PaaS)
     - Software as a Service (SaaS)
  5. Deployment Models:
     - Public Cloud
     - Private Cloud
     - Hybrid Cloud

- **IAAS**

  Infrastructure as a Service (IaaS) is a cloud computing model that provides virtualized computing resources over the internet. In an IaaS environment, users can rent virtual machines, storage, and networking components on a pay-as-you-go basis. This eliminates the need for organizations to invest in and maintain physical hardware, as they can scale their infrastructure up or down based on demand. IaaS offers flexibility, allowing users to deploy and manage their own operating systems, applications, and development frameworks. This model is particularly beneficial for businesses with dynamic or fluctuating workloads, as it provides a cost-effective and scalable solution without the burden of managing and maintaining physical infrastructure. Popular IaaS providers include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

- **AWS**
Amazon Web Services (AWS) is a comprehensive and widely used cloud computing platform provided by Amazon. Launched in 2006, AWS offers a vast array of computing services, including computing power, storage options, and databases, delivered over the internet. AWS allows businesses and individuals to access scalable and cost-effective cloud solutions without the need for significant upfront investments in physical infrastructure. It provides a range of services, from Infrastructure as a Service (IaaS) with offerings like Amazon EC2 virtual servers, to Platform as a Service (PaaS) with tools like AWS Elastic Beanstalk for application deployment and management. AWS has a global presence with data centers in various regions, ensuring low-latency access to services. Its flexibility, reliability, and extensive service catalog make AWS a go-to choice for organizations of all sizes, from startups to large enterprises, seeking to leverage cloud computing for their diverse computing needs.
- **AWS services**
Amazon Web Services (AWS) offers a wide range of cloud computing services, covering computing power, storage, databases, machine learning, analytics, networking, security, and more. Here's an overview of some key AWS services:
1. Compute Services:
   - Amazon EC2 (Elastic Compute Cloud): Provides resizable compute capacity in the cloud, allowing users to run virtual servers for various applications.
   - AWS Lambda: Enables serverless computing, where you can run code without provisioning or managing servers.
2. Storage Services:
   - Amazon S3 (Simple Storage Service): Offers scalable object storage for data backup, archiving, and content delivery.
   - Amazon EBS (Elastic Block Store): Provides block-level storage volumes for use with EC2 instances.
3. Database Services:
   - Amazon RDS (Relational Database Service): Manages relational databases such as MySQL, PostgreSQL, and Oracle.
   - Amazon DynamoDB: A fully managed NoSQL database service for applications requiring low-latency and high-performance data access.
4. Networking:
   - Amazon VPC (Virtual Private Cloud): Enables users to launch Amazon Web Services resources into a virtual network.
   - Amazon Route 53: A scalable domain name system (DNS) web service.

5. Machine Learning and AI:
   - Amazon SageMaker: A fully managed service that enables developers to build, train, and deploy machine learning models.
   - Amazon Comprehend: A natural language processing service that extracts insights and relationships from text.
6. Analytics:
   - Amazon Redshift: A fully managed data warehouse service for running complex queries on large datasets.
   - Amazon EMR (Elastic MapReduce): A cloud-based big data platform for processing vast amounts of data quickly.
7. Management and Monitoring:
   - Amazon CloudWatch: Monitors AWS resources and applications, collecting and tracking metrics.
   - AWS CloudTrail: Records AWS API calls for account activity tracking and security analysis.
8. Security:
   - AWS Identity and Access Management (IAM): Manages access to AWS services securely.
   - AWS Key Management Service (KMS): Creates and controls encryption keys used to encrypt data.
9. Serverless Computing:
   - AWS Step Functions: Coordinates the components of distributed applications using visual workflows.
   - Amazon API Gateway: Creates, publishes, and manages APIs.
10. Internet of Things (IoT):
   - AWS IoT Core: Connects devices to the cloud and enables secure communication between them.

- **EC2**
  Amazon Elastic Compute Cloud (Amazon EC2) is a central component of Amazon Web Services (AWS), providing scalable and resizable compute capacity in the cloud. With EC2, users can easily launch virtual servers, known as instances, to run applications, host websites, and perform various computing tasks. Users have flexibility in choosing the instance type, operating system, storage, and network configurations, allowing for tailored computing environments. EC2 instances can be provisioned and scaled up or down based on demand, providing cost efficiency by paying only for the resources used. This on-demand nature makes EC2 suitable for a wide range of applications, from simple web hosting to complex, high-performance computing tasks. Additionally, EC2 instances can be
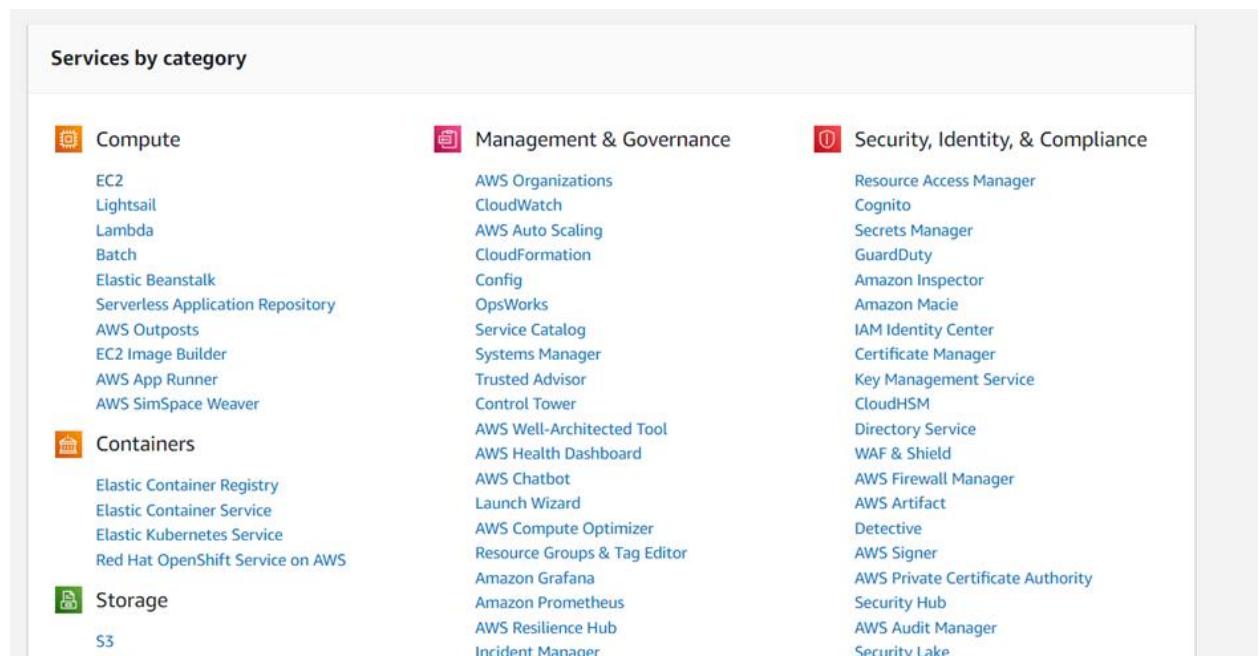
integrated with other AWS services, offering a comprehensive and customizable cloud computing solution for businesses and developers.

1. **Implement the windows machine using AWS ec2.**

   STEPS:

   Step 1: Sign into your AWS account
   Step 2: Select All Services, Select EC2

   Step 3: Launch Instance, create key value pair, pem and save
   Step 4: Select Windows and launch the instance

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

Windows

Add additional tags

Create key pair   ✕

Key pair name

Key pairs allow you to connect to your instance securely.

windows123

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**

Step 5: Go to instances and initialize and then start running
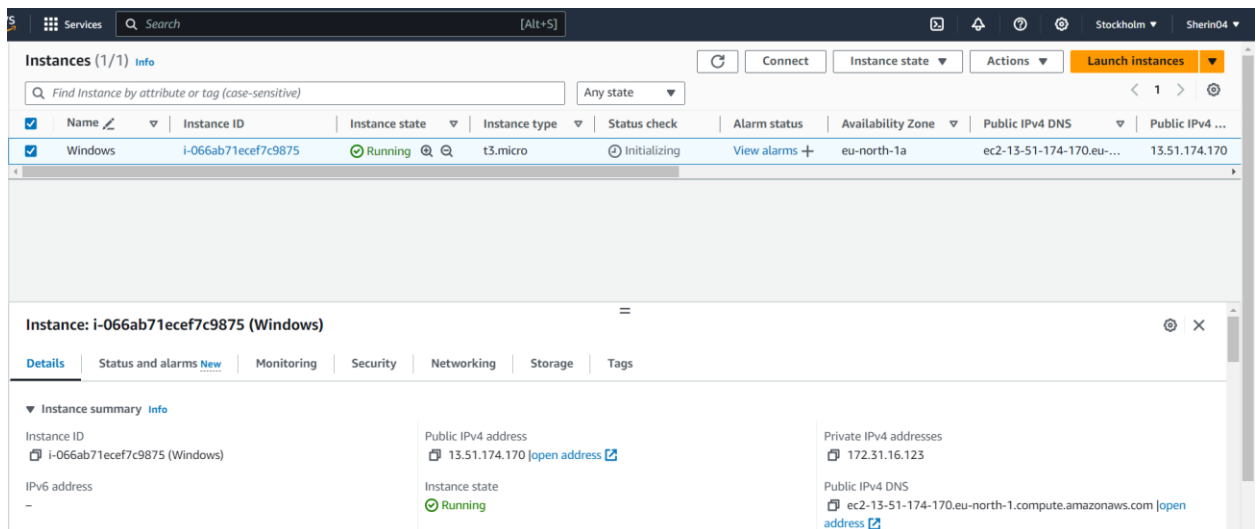
Step 6: Select the instance, click on connect for connecting the RDP client

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

Instance ID

📋 i-066ab71ecef7c9875 (Windows)

Key pair associated with this instance

📋 windows123

Private key

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

🔲 Upload private key file

⊘ windows123.pem
  1.674KB

Private key contents - *optional*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA0rl7l0KPg6+2LkxzI751/rPTlVFLyYHHeAnBqc0VQL9wefgX
MOZKIrH8Gr4rQqmtPCOfkYnkBXXv4TrwpUACrIdp56ogr+hgWxW0wXehGdc5crsV
V/jTzlGypTCmkwNT487PRAJKJpgbV622wyBxYmJzFbZKPxUohiMDbZw9Bh1IycRW
G7bu69Fto9mjDIBfMeg/IBVAhuybxxV1DSuG474x7av1oPq+4osXhXyYyORrqZQs
SFXIsgdO4IPujJzUSPveolTUZamt5eaTKCr3Fy585rFjr0pwjQpam5GrmNbBrSSO
p0JU3UQb+GMhlSYbbr6593rQKgTyEut19gkjTQIDAQABAoIBAFpSjMoiCpfcag9g
T34DtlntRITdKsn9qb0nGW6p7DErnY96Kp6YJzDF75Sa7LoPNqTvdAcS/rzkd2So
```

Step 7: Decrypt the password. Copy the password.

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	Username
▢ ec2-13-51-174-170.eu-north- .com	▢ Administrator

✓ Password copied

▢ 7zyU&inM&O47bgQh4c7;lz8plyoRI1gc

ⓘ If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

Step 8: Open remote desktop connection

🖥 Remote Desktop Connection   ✕

⚠ **The publisher of this remote connection can't be identified. Do you want to connect anyway?**
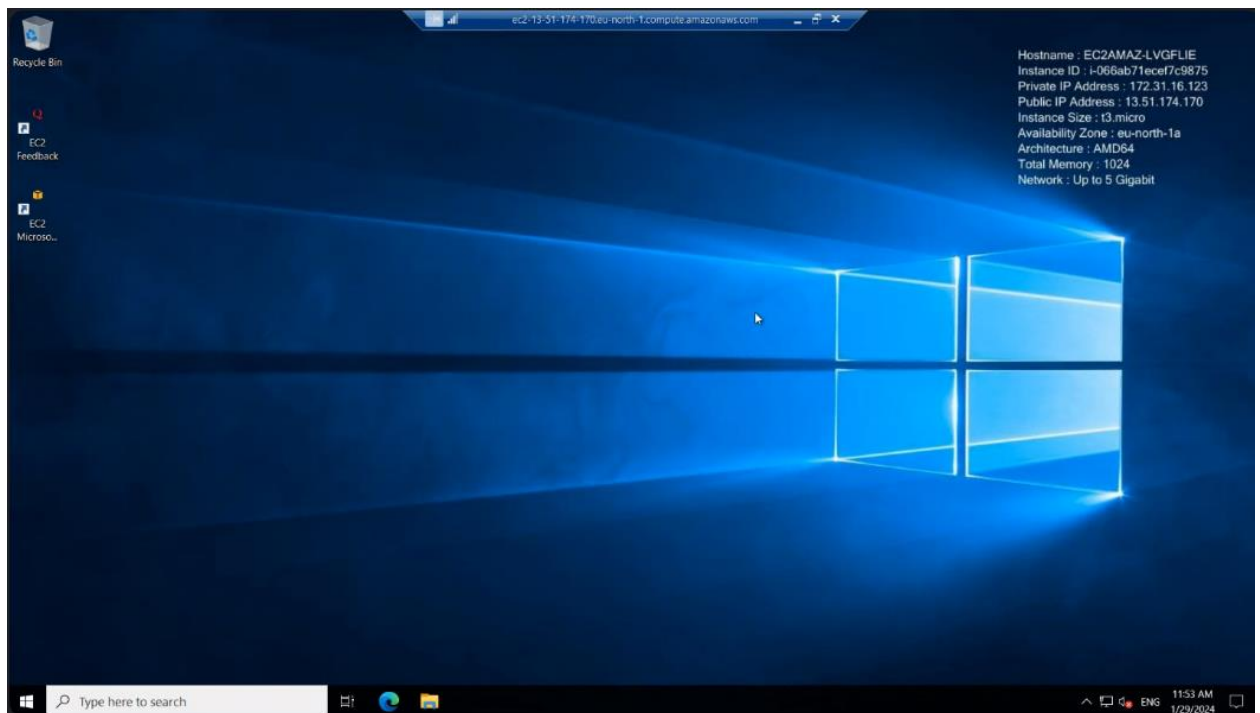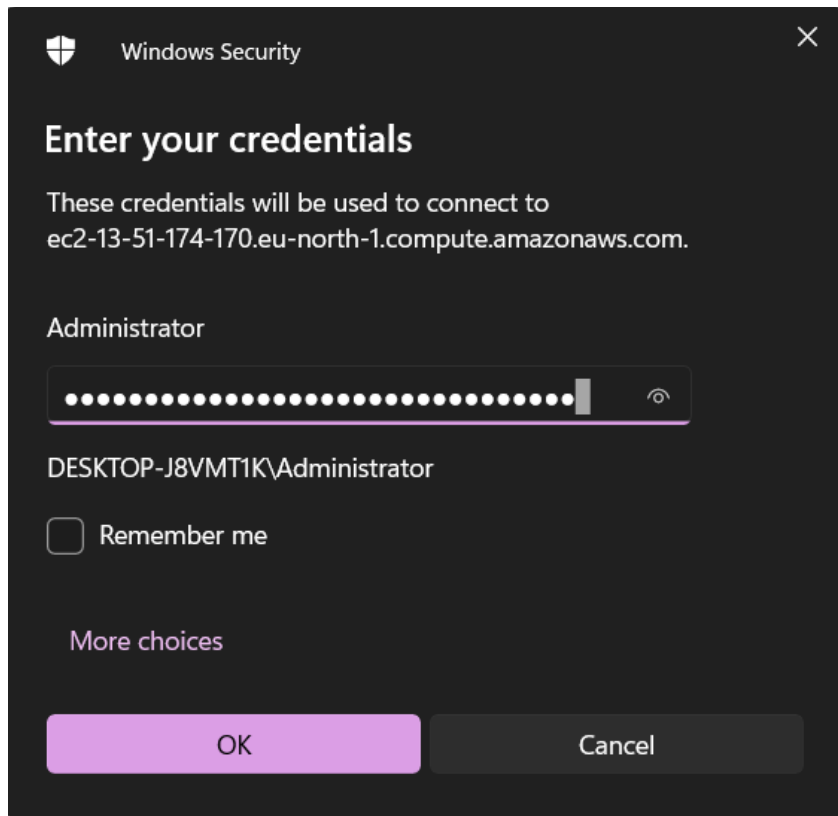
This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

Publisher: **Unknown publisher**
Type: Remote Desktop Connection
Remote computer: ec2-13-51-174-170.eu-north-1.compute.amazonaws.com

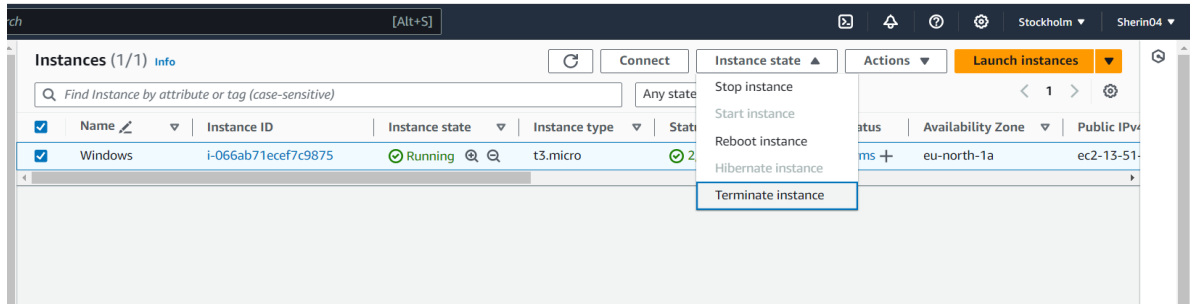☐ Don't ask me again for connections to this computer

⊘ Show Details   [Connect]   [Cancel]

Step 9: Close RDP and go back to instances
Step 10: Terminate the instance

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

STEPS:

Step 1: Launch a new instance for Linux

Step 2: Write a new web server name and select Ubuntu server
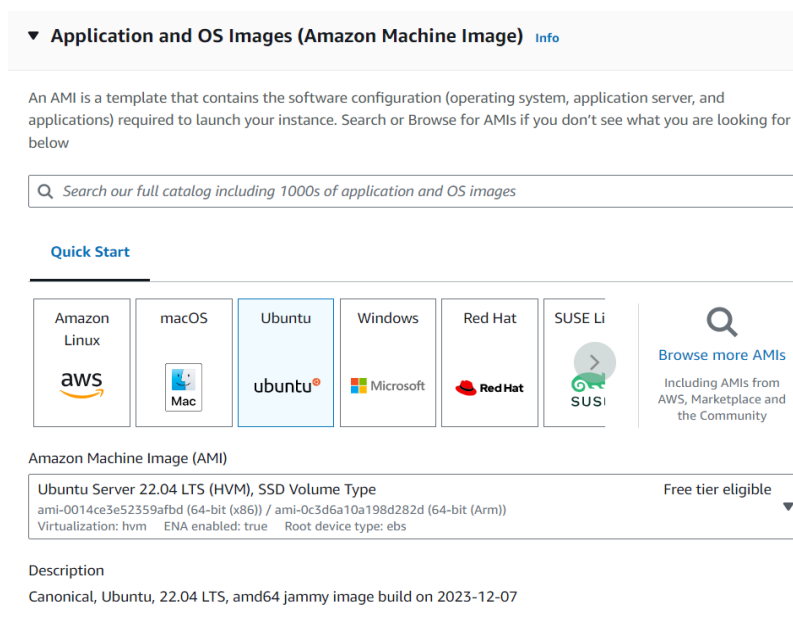
Step 3: Create a new key value pair and select ppk

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

tiger

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 ↗

Cancel   **Create key pair**

Network   Info
vpc-0e60bd18c6915ece3

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-6'** with the following rules:

☑ Allow SSH traffic from   Anywhere
Helps you connect to your instance   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

⚠ 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.

Step 4: Download putty.exe file from Google

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)

Step 5: Launch the instance

Step 6: Select and copy public IPV4 address

Step 5: Go to putty and paste the IP address
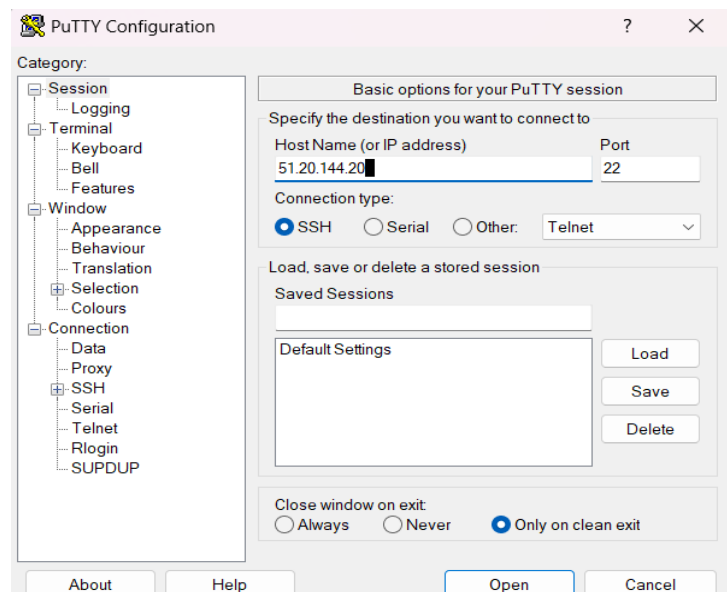
Step 6: Go to category → SSH → auth → credentials → select the ppk file
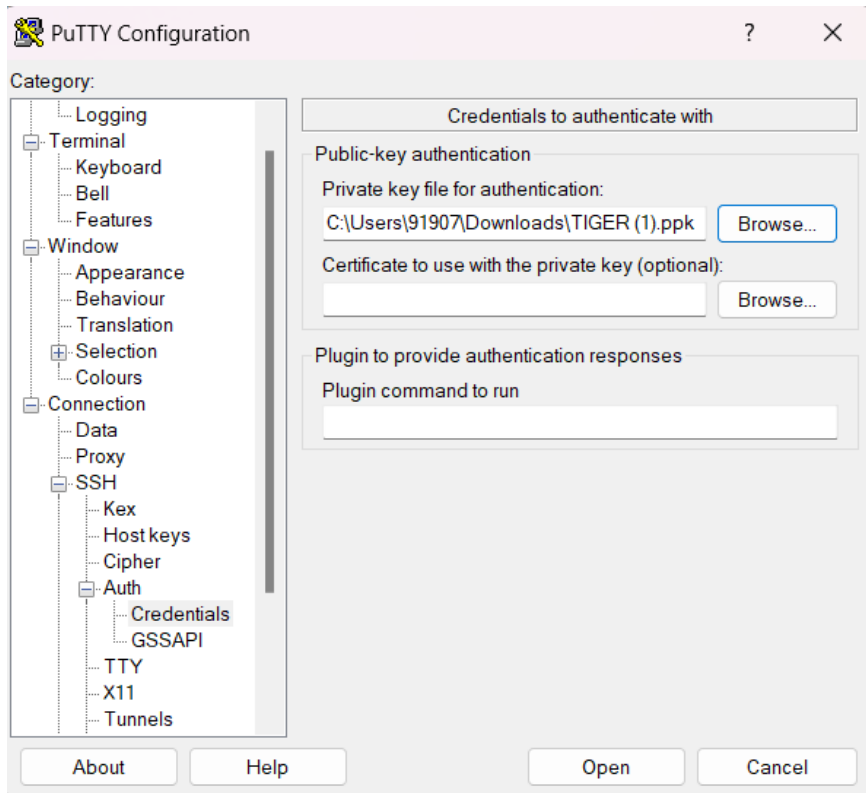
Step 7: Putty will launch
Step 8: Login as ubuntu
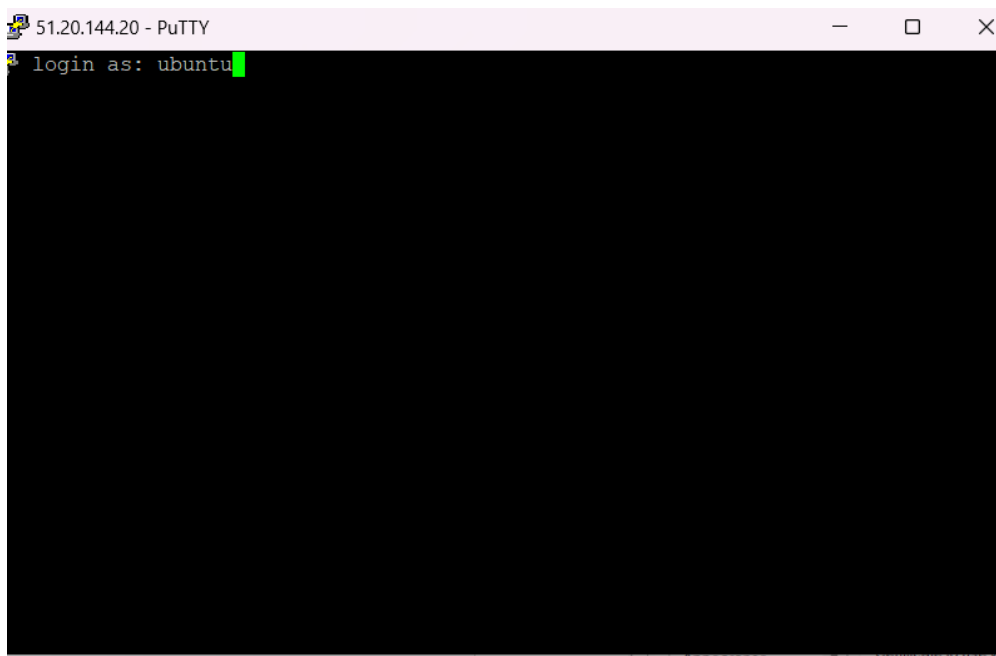

Step 9: Install python

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-19-183:~$ ls
ubuntu@ip-172-31-19-183:~$ mkdir msc
ubuntu@ip-172-31-19-183:~$ ls
msc
ubuntu@ip-172-31-19-183:~$ cd msc
ubuntu@ip-172-31-19-183:~/msc$ touch cloud.txt
ubuntu@ip-172-31-19-183:~/msc$ ls
cloud.txt
ubuntu@ip-172-31-19-183:~/msc$ cloud.txt
cloud.txt: command not found
ubuntu@ip-172-31-19-183:~/msc$ cat



^Z
[1]+  Stopped                        cat
```

```
ubuntu@ip-172-31-19-183:~/msc$  cat> cloud.txt
I am working on linux
^Z
[2]+  Stopped                        cat > cloud.txt
ubuntu@ip-172-31-19-183:~/msc$ cat cloud.txt
I am working on linux
ubuntu@ip-172-31-19-183:~/msc$ nano cloud.txt
ubuntu@ip-172-31-19-183:~/msc$ cat cloud.txt
^T^I am working on linux
ubuntu@ip-172-31-19-183:~/msc$ python 3
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
ubuntu@ip-172-31-19-183:~/msc$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(Hello World)
  File "<stdin>", line 1
    print(Hello World)
          ^^^^^^^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
>>> print('Hello World')
Hello World
>>>
```

Step 10: Install Firefox

```
ubuntu@ip-172-31-19-183: ~/msc                                  —    □    ✕

ubuntu@ip-172-31-19-183:~/msc$ nano cloud.txt
ubuntu@ip-172-31-19-183:~/msc$ cat cloud.txt
^T^I am working on linux
ubuntu@ip-172-31-19-183:~/msc$ python 3
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
ubuntu@ip-172-31-19-183:~/msc$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(Hello World)
  File "<stdin>", line 1
    print(Hello World)
          ^^^^^^^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
>>> print('Hello World')
Hello World
>>>
>>>
>>>
[3]+  Stopped                 python3
ubuntu@ip-172-31-19-183:~/msc$ sudo snap install firefox
firefox 122.0-2.1 from Mozilla✓ installed
ubuntu@ip-172-31-19-183:~/msc$ ▮
```