Assignment: 01

Practical Title: Case study on Amazon EC2 and learn about Amazon EC2 web services.

Objectives:

• To learn Amazon EC2 web services
• To study on Amazon EC2 and learn about Amazon EC2 web services.

Hardware Requirements:

• Pentium IV with latest configuration

Software Requirements:

• Ubuntu 20.04

Theory:

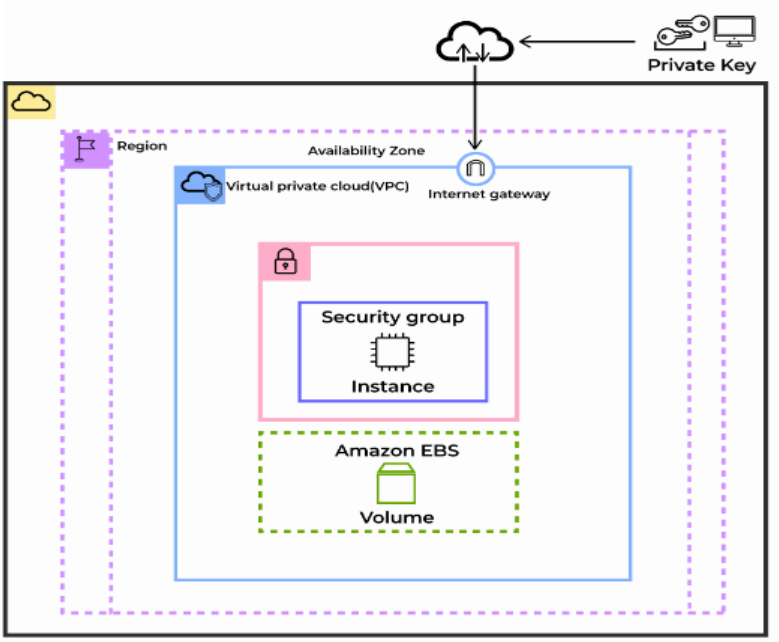
An EC2 instance is nothing but a virtual server in Amazon Web services terminology. It stands for Elastic Compute Cloud. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud. An on-demand EC2 instance is an offering from AWS where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications. The instance will be charged per hour with different rates based on the type of the instance chosen.

AWS provides multiple instance types for the respective business needs of the user. Thus, you can rent an instance based on your own CPU and memory requirements and use it as long as you want. You can terminate the instance when it's no more used and save on costs. This is the most striking advantage of an on-demand instance- you can drastically save on your CAPEX.

What is Amazon EC2 (Elastic Compute Cloud)?

Amazon Web service offers EC2 which is a short form of Elastic Compute Cloud (ECC) it is a cloud computing service offered by the Cloud Service Provider AWS. You can deploy your applications in EC2 servers without any worrying about the underlying infrastructure. You configure the EC2-Instance in a very secure manner by using the VPC, Subnets, and Security groups. You can scale the configuration of the EC2 instance you have configured based on the demand of the application by attaching the autoscaling group to the EC2 instance. You can scale up and scale down the instance based on the incoming traffic of the application.

following figure shows the EC2-Instance which is deployed in VPC (Virtual Private Cloud).

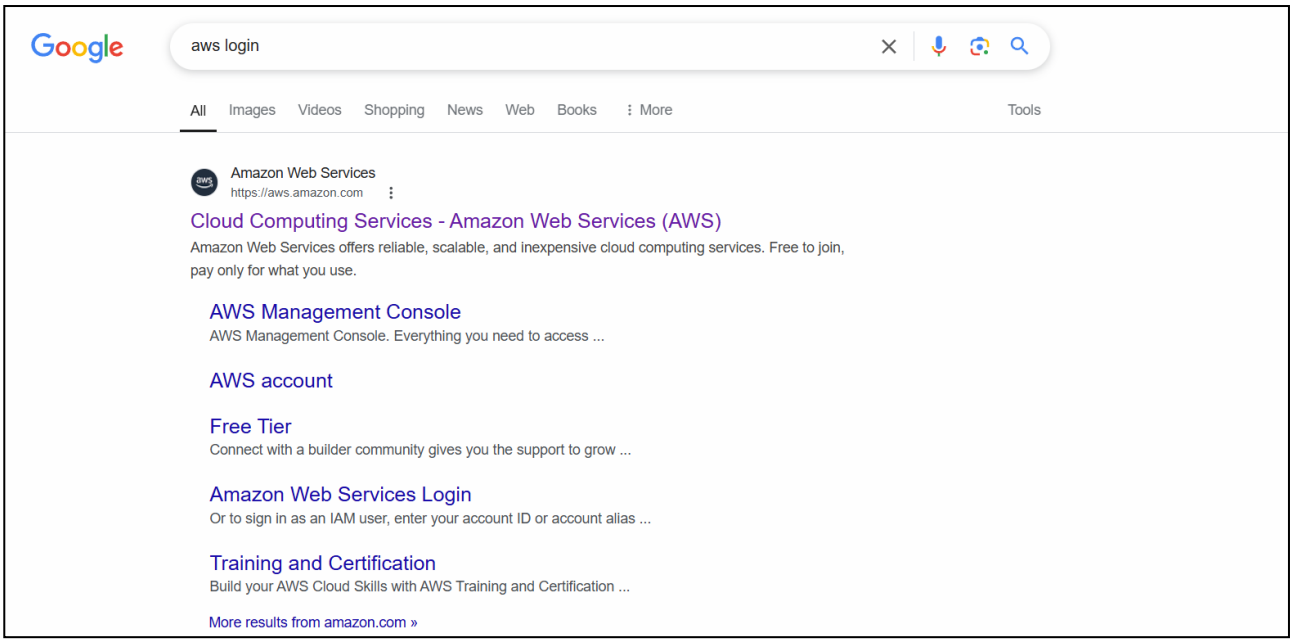


Use Cases of Amazon EC2 (Elastic Compute Cloud)

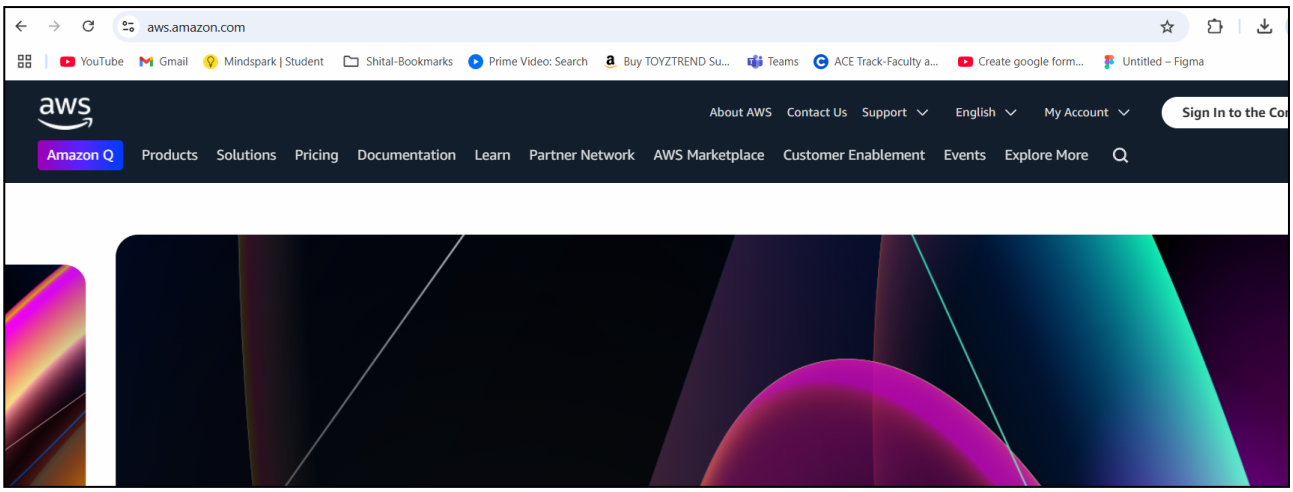
1. **Deploying Application:** In the AWS EC2 instance, you can deploy your application like **.jar,.war,** or **.ear** application without maintaining the underlying infrastructure.
1. **Scaling Application:** Once you deployed your web application in the EC2 instance know you can scale your application based upon the demand you are having by scaling the AWS EC2-Instance.
1. **Deploying The ML Models:** You can train and deploy your ML models in the EC2-instance because it offers up to 400 Gbps), and storage services purpose-built to optimize the price performance for ML projects.
1. **Hybrid Cloud Environment:** You can deploy your web application in EC2-Instance and you can connect to the database which is deployed in the on-premises servers.
1. **Cost-Effective:** Amazon EC2-instance is cost-effective so you can deploy your gaming application in the Amazon EC2-Instances

Steps For Creating EC2 Instance- Take a print from here till end

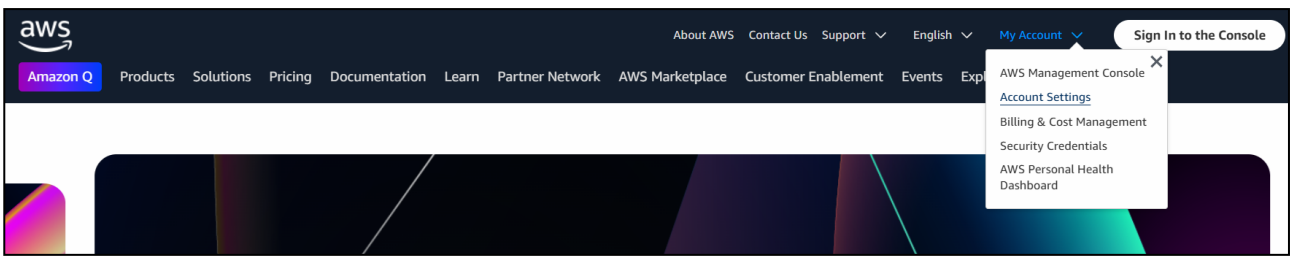
Login to AWS Account



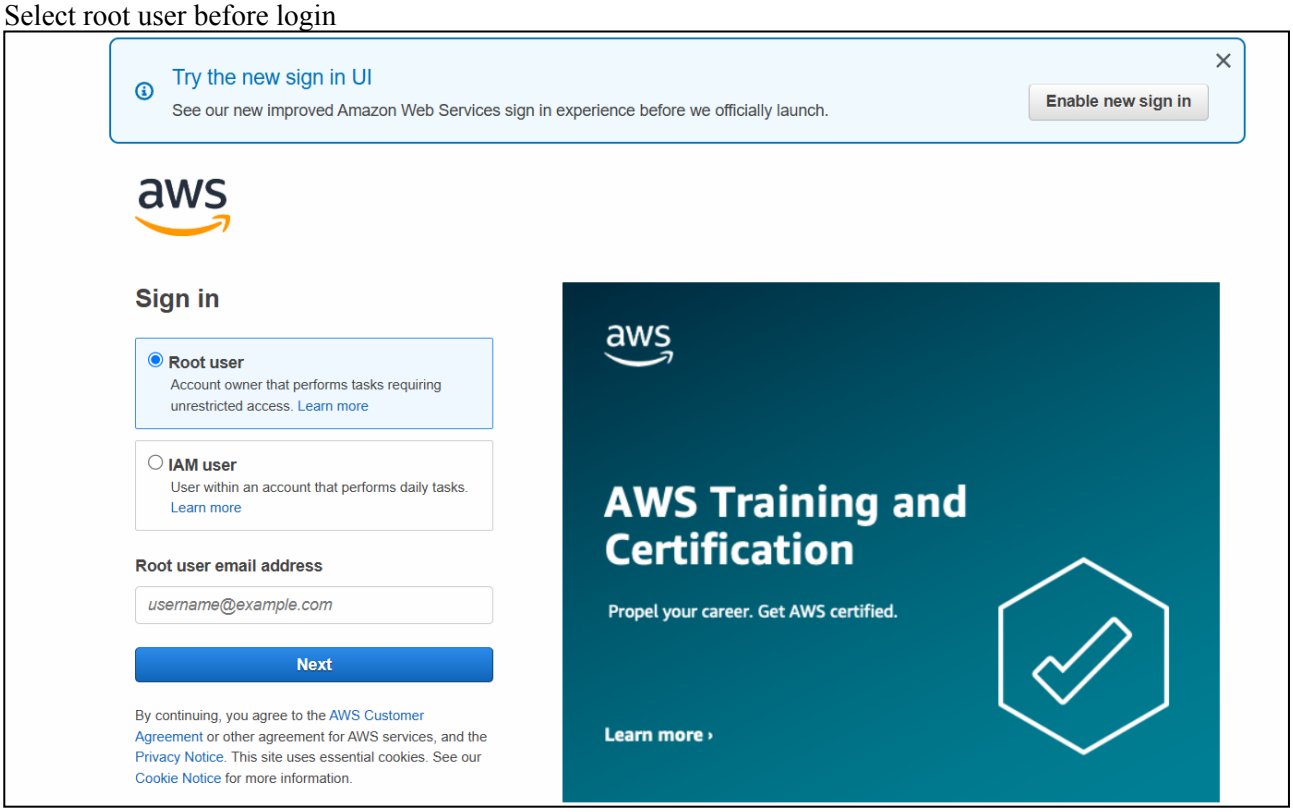
Click on Cloud Computing Service



Then go to My Account and select Account settings

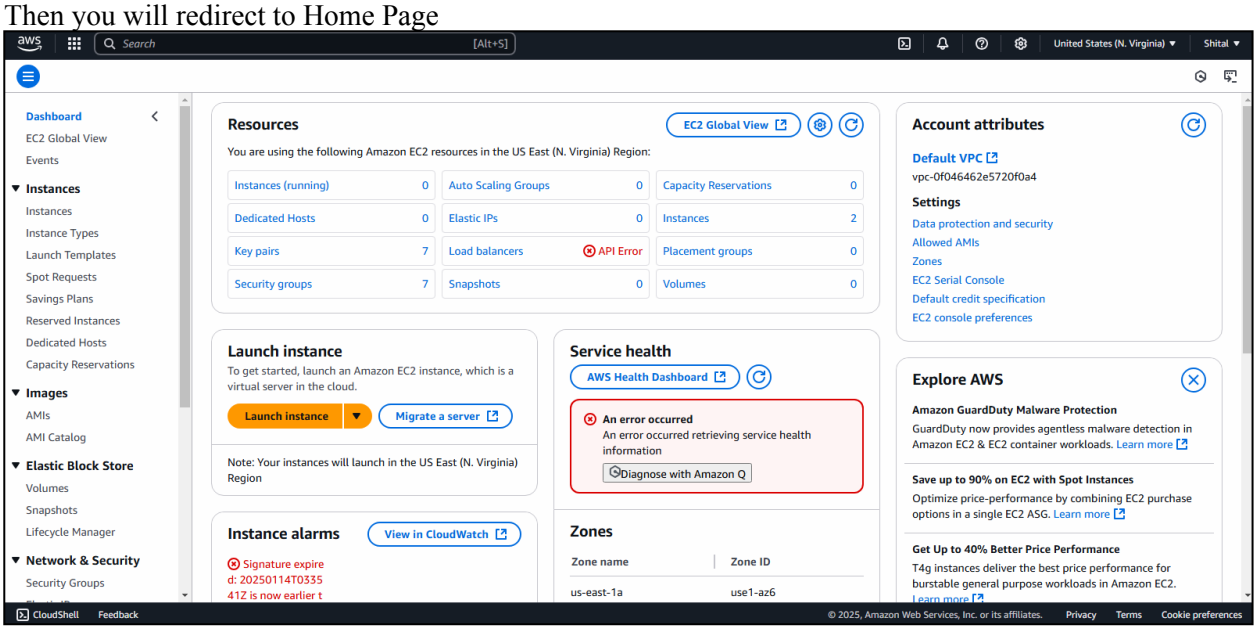


Select root user before login

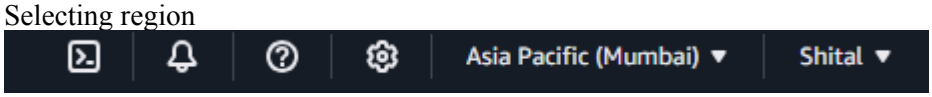


Provide your credential and click on next

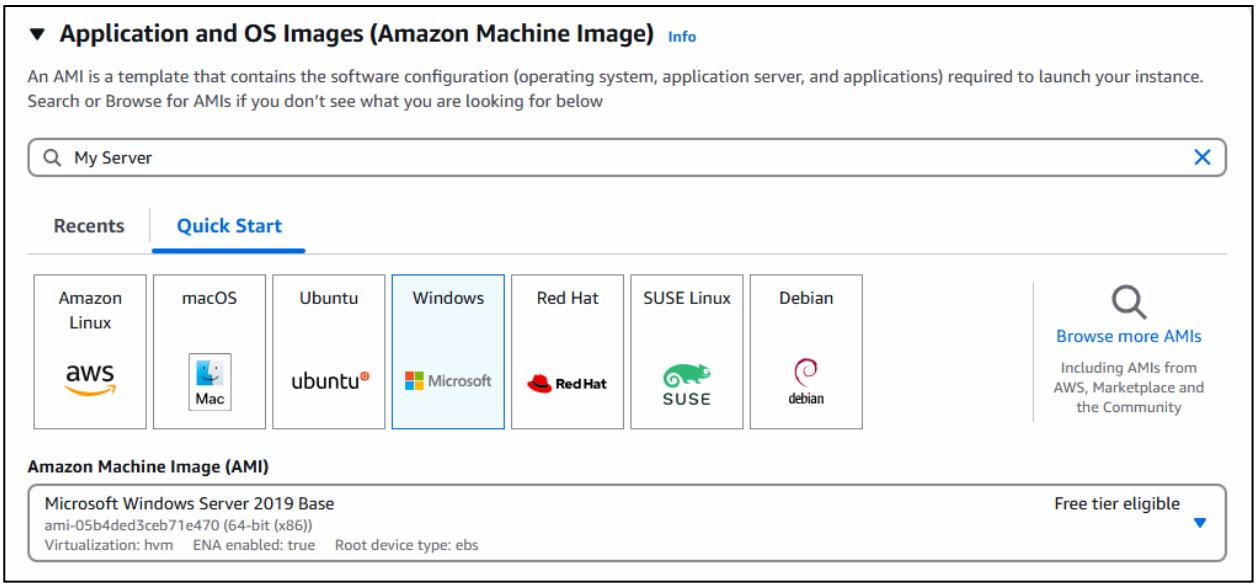
Then you will redirect to Home Page



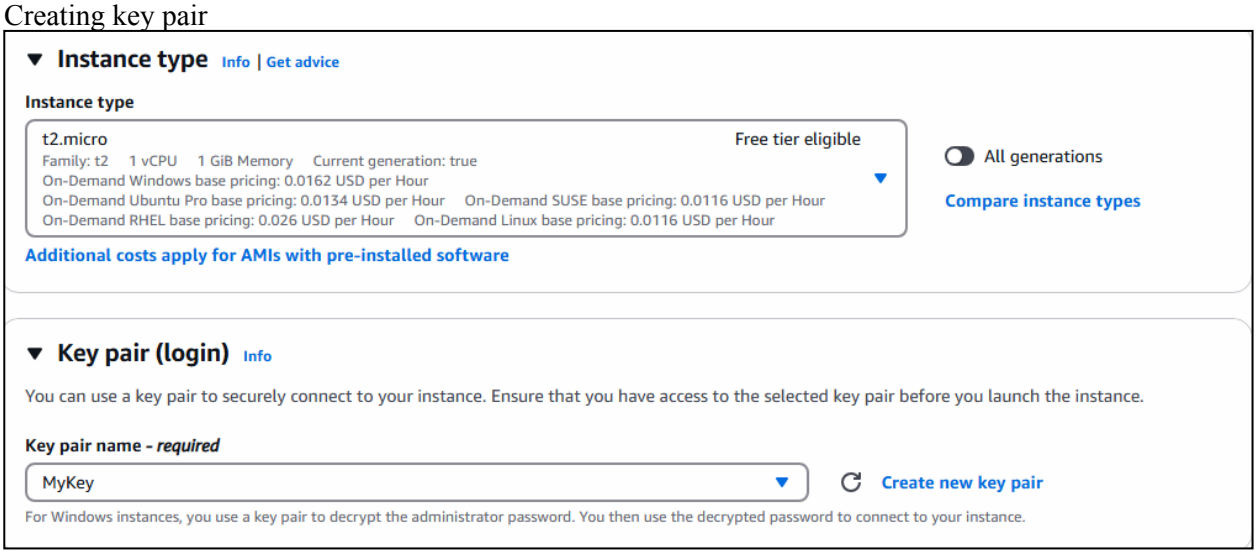
Selecting region



Then click on Launching an instance



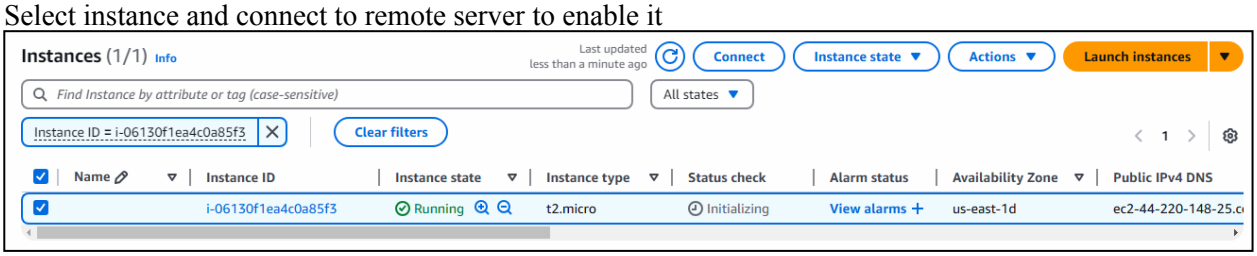
Creating key pair



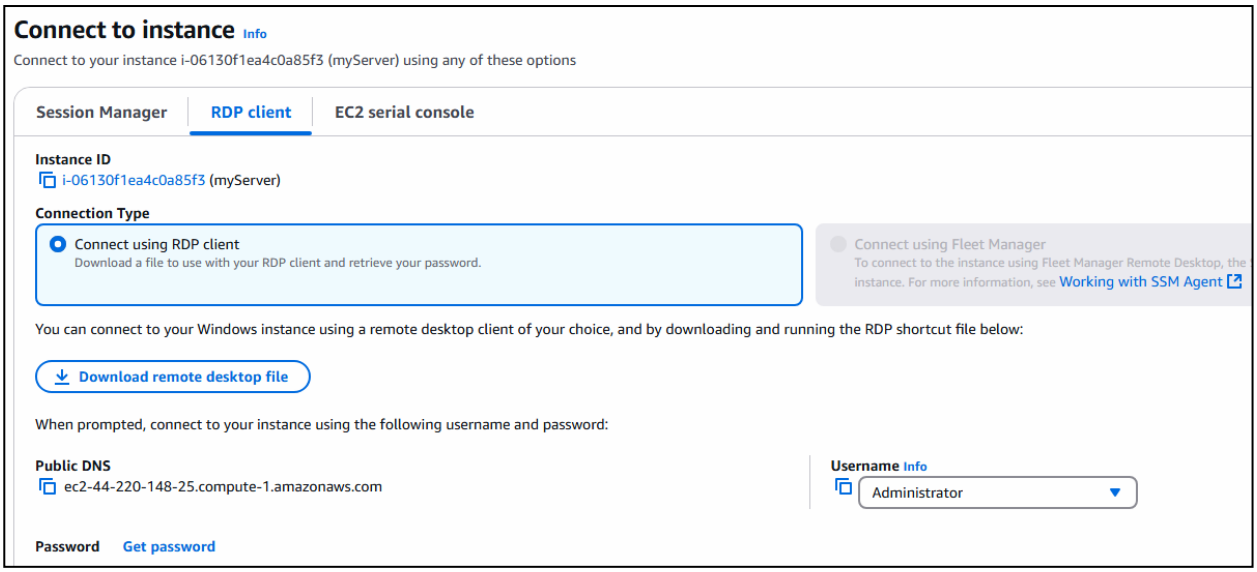
Click on create new key pair
Provide name for key pair(E.g S1.key)
Then select .pem option
Then key file downloaded

Launch the instance.
Instance is successfully created (follow the link)

Select instance and connect to remote server to enable it

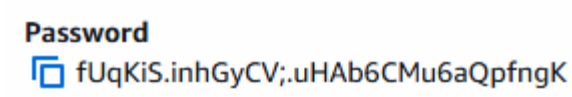


Then click on connect
Choose RDP agent and click on “download remote desktop file” and “Get password”

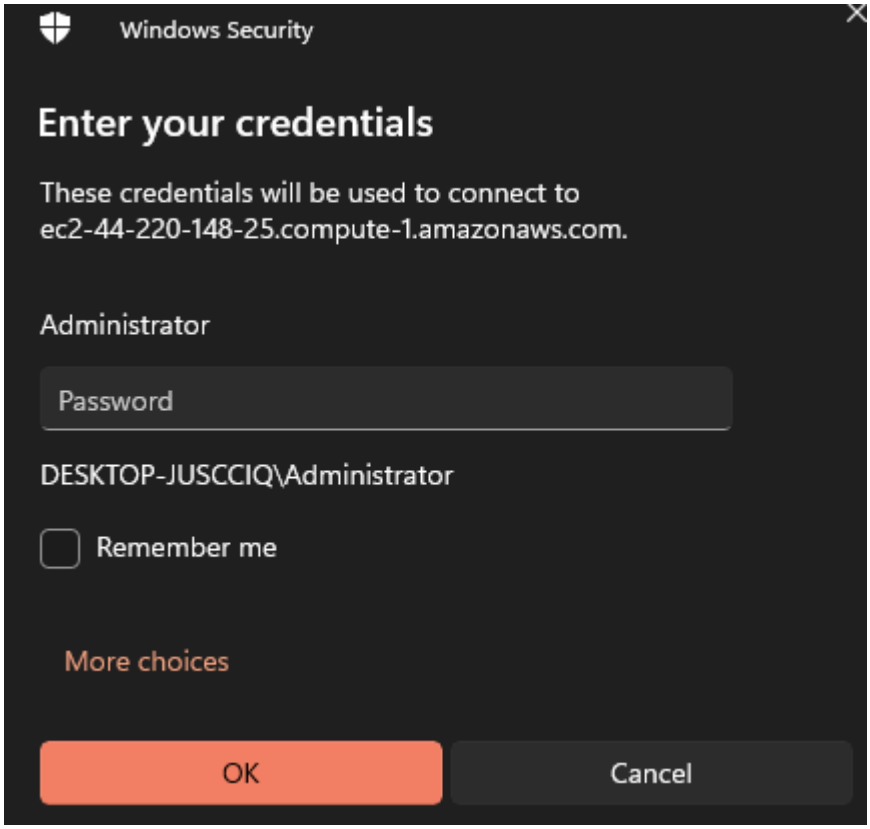


Upload downloaded key pair file

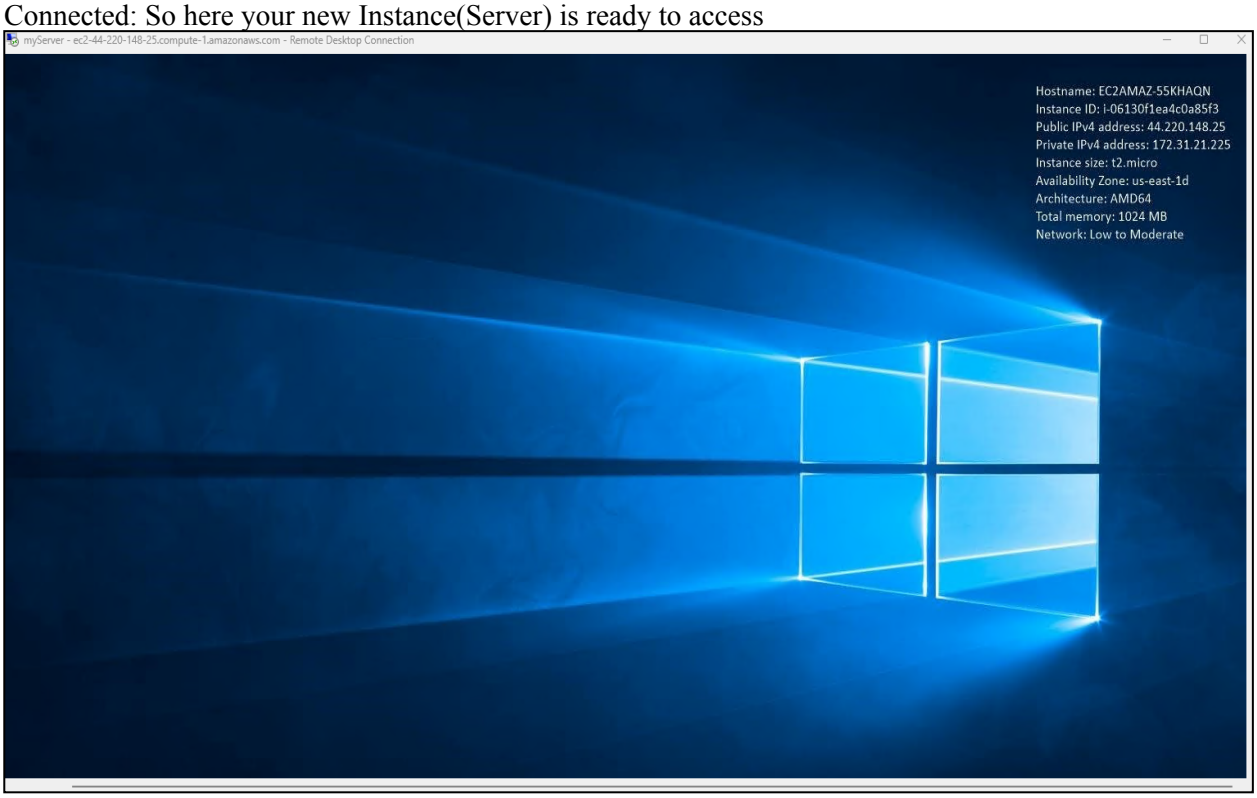
Save decrypted password



Open the downloaded server application Enter decrypted password



Connected: So here your new Instance(Server) is ready to access



Conclusion:

Thus, we saw in detail how to create an on-demand EC2 instance. Because it is an on demand server, you can keep it running when in use and 'Stop' it when it's unused to save on your costs