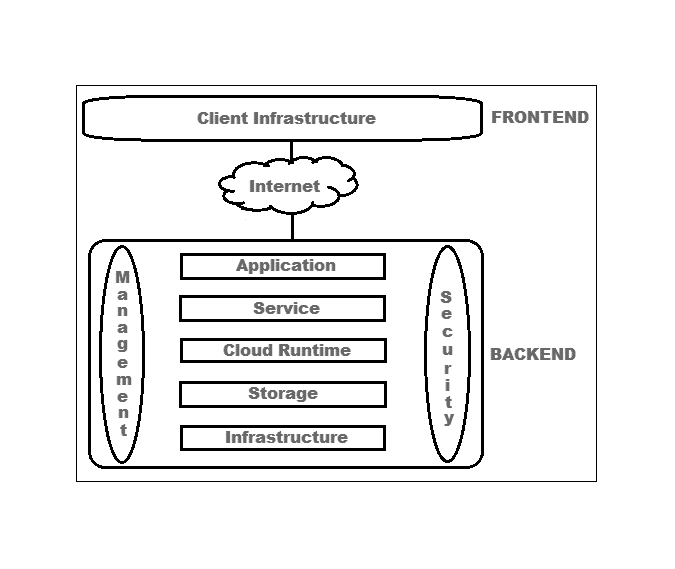
**CLOUD COMPUTING PRACTICAL 01 – IaaS (Infrastructure as a Service)**

**RHEA MENON | A029**

**Cloud Computing Architecture :**  
The cloud architecture is divided into 2 parts i.e.

1. Frontend
2. Backend



Architecture of cloud computing is the combination of both [SOA (Service Oriented Architecture)](https://www.geeksforgeeks.org/service-oriented-architecture/) and EDA (Event Driven Architecture). Client infrastructure, application, service, runtime cloud, storage, infrastructure, management and security all these are the components of cloud computing architecture.

**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](https://www.geeksforgeeks.org/cloud-based-services/). 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.
6. **Management –**  
   Management in backend refers to management of backend components like application, service, runtime cloud, storage, infrastructure, and other security mechanisms etc.
7. **Security –**  
   Security in backend refers to implementation of different security mechanisms in the backend for secure cloud resources, systems, files, and infrastructure to end-users.
8. **Internet –**  
   Internet connection acts as the medium or a bridge between frontend and backend and establishes the interaction and communication between frontend and backend.
9. **Database**– Database in backend refers to provide database for storing structured data, such as SQL and NOSQL databases. Example of Databases services include Amazon RDS, Microsoft Azure SQL database and Google CLoud SQL.
10. **Networking**– Networking in backend services that provide networking infrastructure for application in the cloud, such as load balancing, DNS and virtual private networks.
11. **Analytics**– Analytics in backend service that provides analytics capabillities for data in the cloud, such as warehousing, bussness intellegence and machine learning.

**IaaS**

Infrastructure as a Service (IaaS) is a fundamental component of cloud computing that provides virtualized computing resources over the internet. In an IaaS model, users can rent virtualized hardware resources, such as virtual machines, storage, and networking, instead of investing in and maintaining physical infrastructure.

Key characteristics of IaaS include:

1. On-Demand Resources: IaaS allows users to access and provision resources on-demand, enabling flexibility and scalability. Users can quickly scale up or down based on their current needs.

2. Virtualization: IaaS relies heavily on virtualization technologies to abstract physical hardware and create virtual instances of computing resources. This enables efficient resource utilization and isolation between different users.

3. Self-Service: Users have control over their computing infrastructure, and they can provision and manage resources through a web-based interface or API without requiring direct human intervention from the service provider.

4. Resource Pooling: IaaS providers pool and share computing resources among multiple users, optimizing resource utilization and achieving economies of scale. This allows users to benefit from cost savings compared to traditional on-premises infrastructure

5. Pay-As-You-Go Model: Users are billed based on their actual usage of resources, typically on a metered or pay-as-you-go basis. This cost model provides cost efficiency by eliminating the need for upfront capital investment and allowing users to pay only for the resources they consume.

6. Scalability: IaaS platforms are designed to easily scale resources horizontally or vertically, allowing businesses to adapt to changing workloads and requirements.

Popular examples of IaaS providers include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Organizations leverage IaaS to build, deploy, and manage their applications without the complexities and costs associated with maintaining physical hardware.

**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 cloud services, enabling individuals, businesses, and organizations to build, deploy, and manage applications and infrastructure in a scalable, flexible, and cost-effective manner. Here are some key points about AWS:

1. Service Offering: AWS provides a diverse range of services, including computing power, storage, databases, machine learning, analytics, content delivery, and more. These services are grouped into categories like Compute, Storage, Database, Machine Learning, Analytics, and Networking, among others.

2. Global Infrastructure: AWS has a global network of data centers, known as Availability Zones, located in multiple geographic regions around the world. This allows users to deploy applications in proximity to their end-users for lower latency and increased reliability.

3. Elasticity and Scalability: AWS allows users to scale resources up or down based on demand. This elasticity ensures that businesses can adapt to varying workloads, optimizing costs and performance.

4. Pay-as-You-Go Model: AWS follows a pay-as-you-go pricing model, where users only pay for the computing resources and services they consume. This eliminates the need for upfront capital investment and provides cost efficiency.

5. Security and Compliance: AWS places a strong emphasis on security, offering a wide range of tools and features to help users secure their data and applications. AWS also complies with various industry-specific and regional compliance standards.

6. Developer-Friendly Environment: AWS provides tools and resources for developers, including software development kits (SDKs), application programming interfaces (APIs), and a variety of programming languages to build and deploy applications seamlessly.

7. Ecosystem and Marketplace: AWS has a vast ecosystem of partners and a marketplace where users can find and deploy third-party solutions, further enhancing the capabilities of their applications.

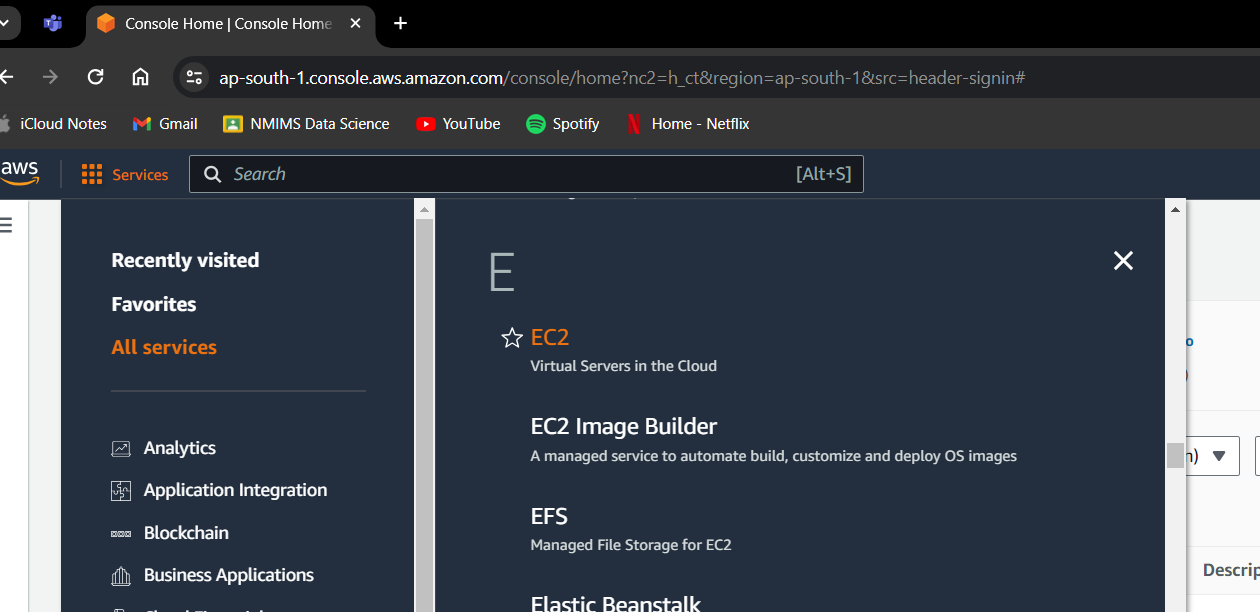
8. Innovation: AWS is known for continuous innovation, regularly introducing new services and features to address emerging technologies and industry trends.

**EC2**

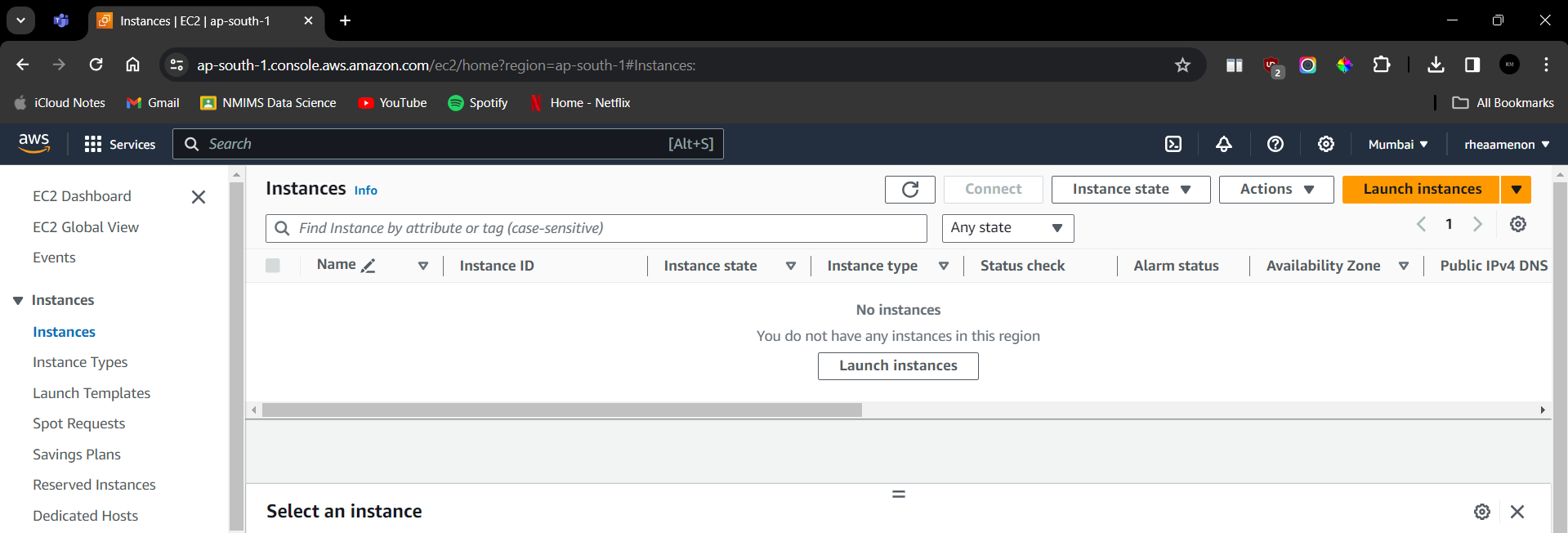
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate them from common failure scenarios.

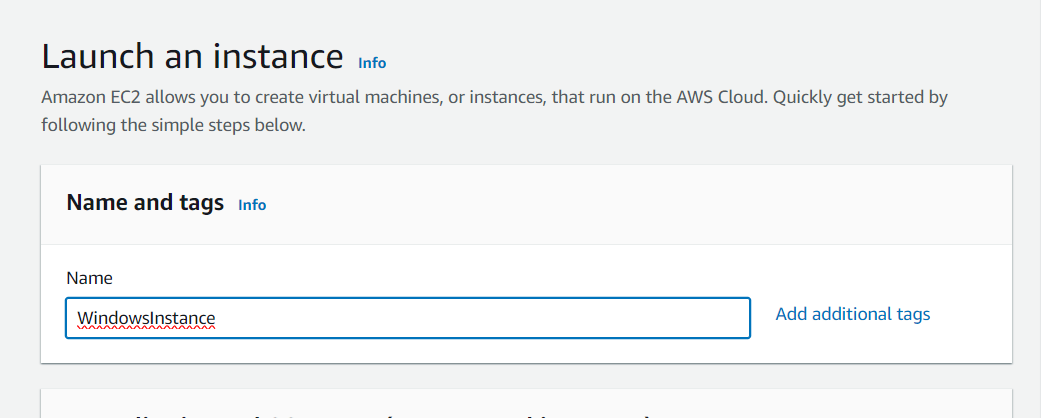
Go to the AWS console home, select All Services and select EC2.

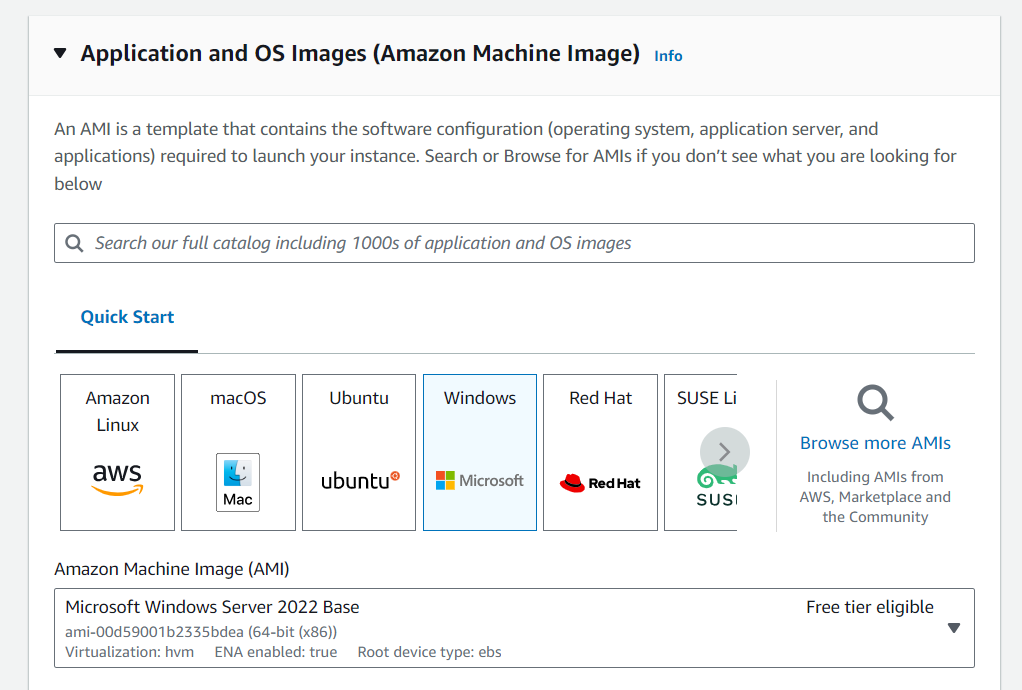


Go to Instances and launch an instance.

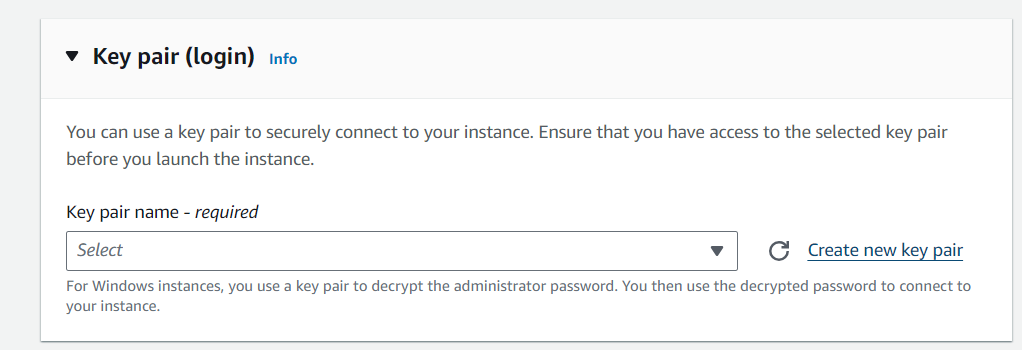


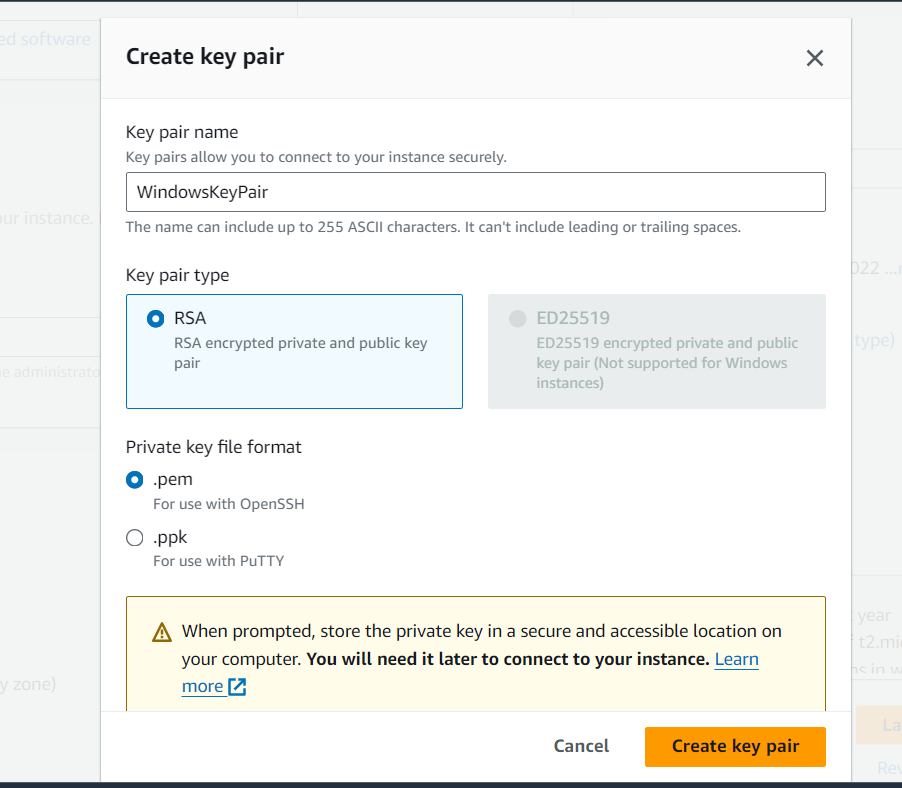
Give a name to the instance and choose Windows as the OS Image.



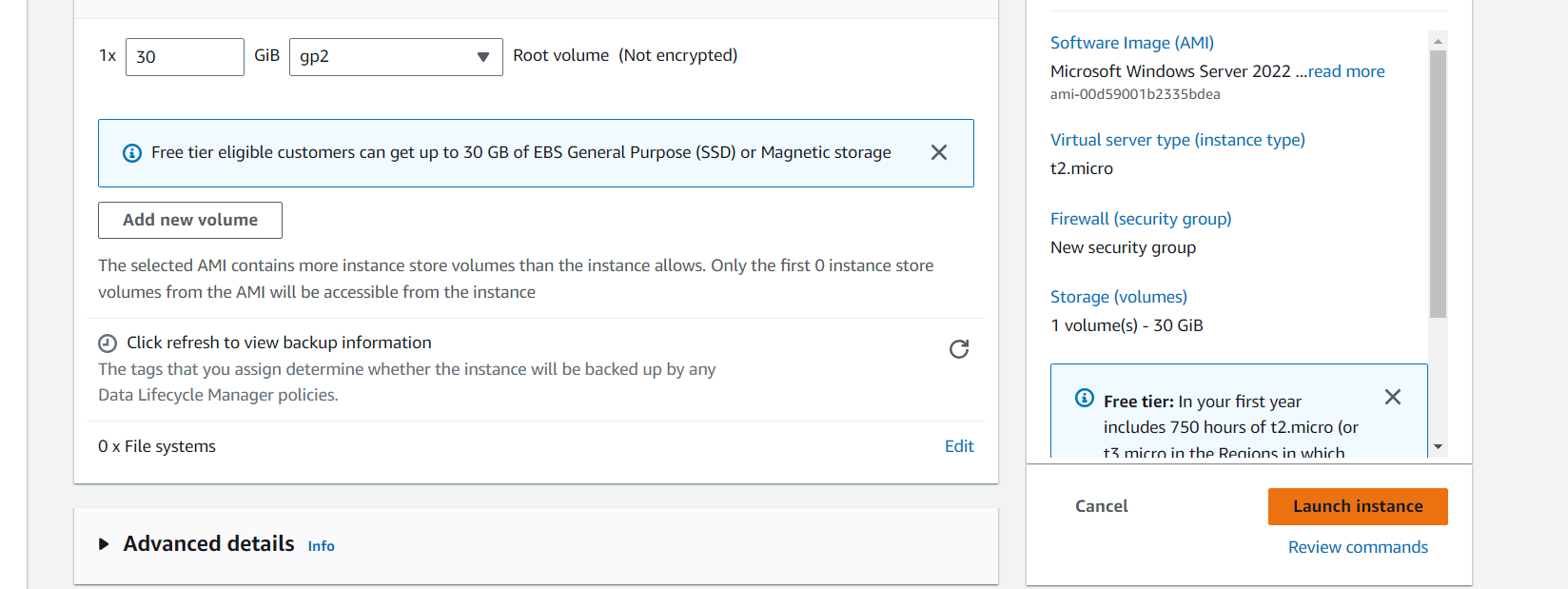


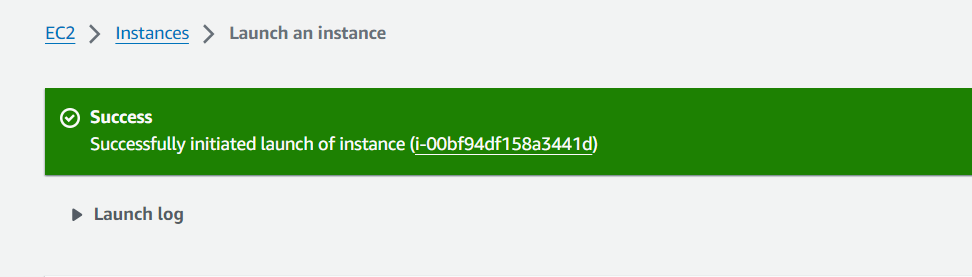
Create a new key pair and save the .pem file in your local device.



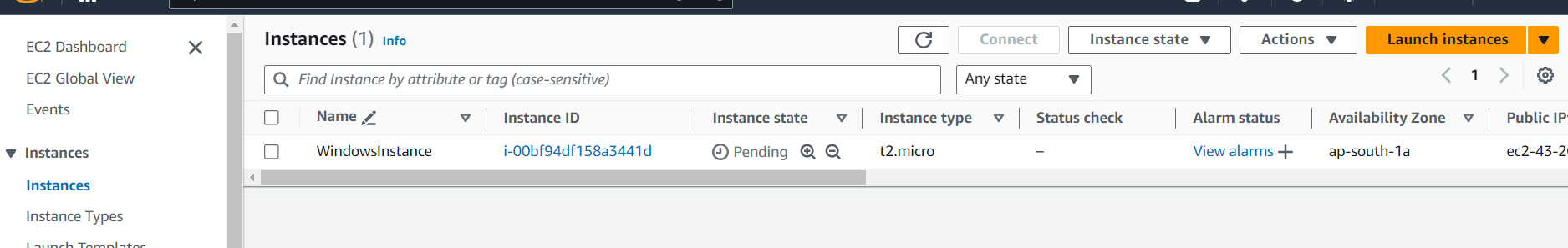


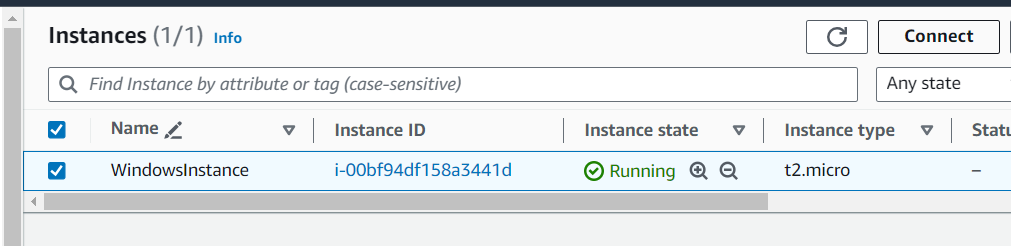
Leave all other default settings and launch the instance.



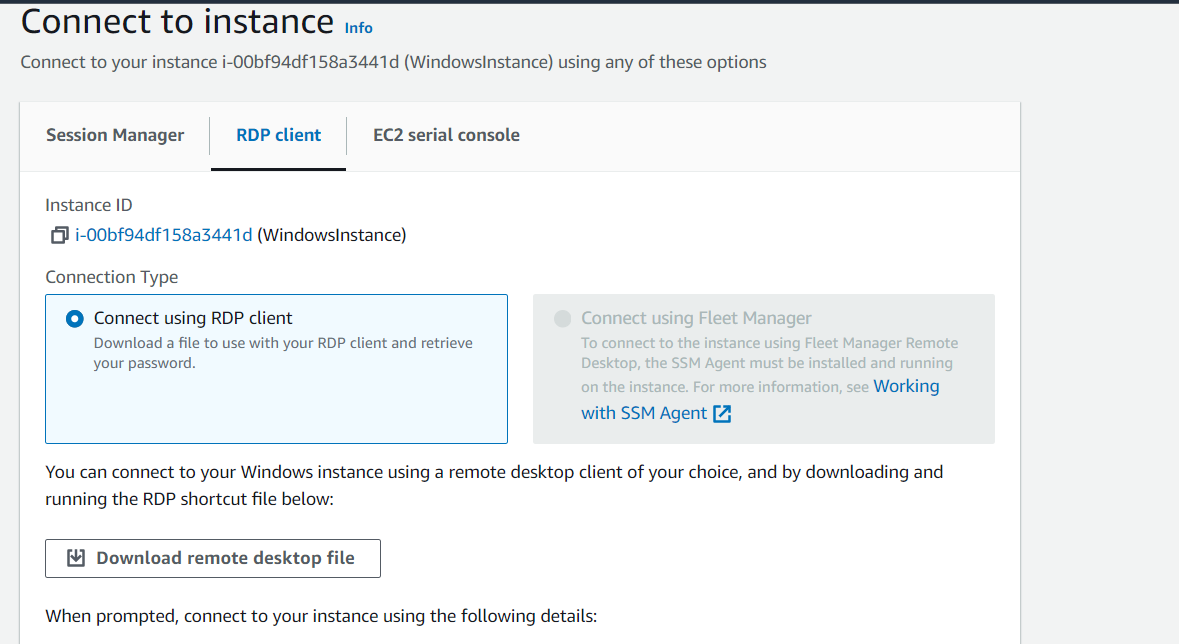


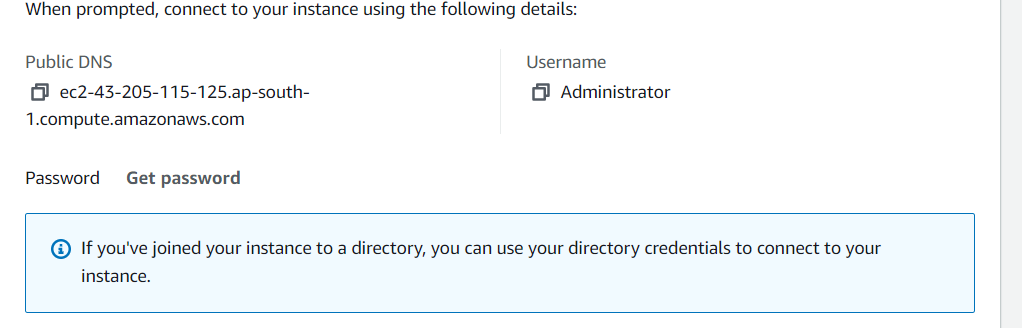
Go to the Instances menu, tick the created instance and click the Connect button.



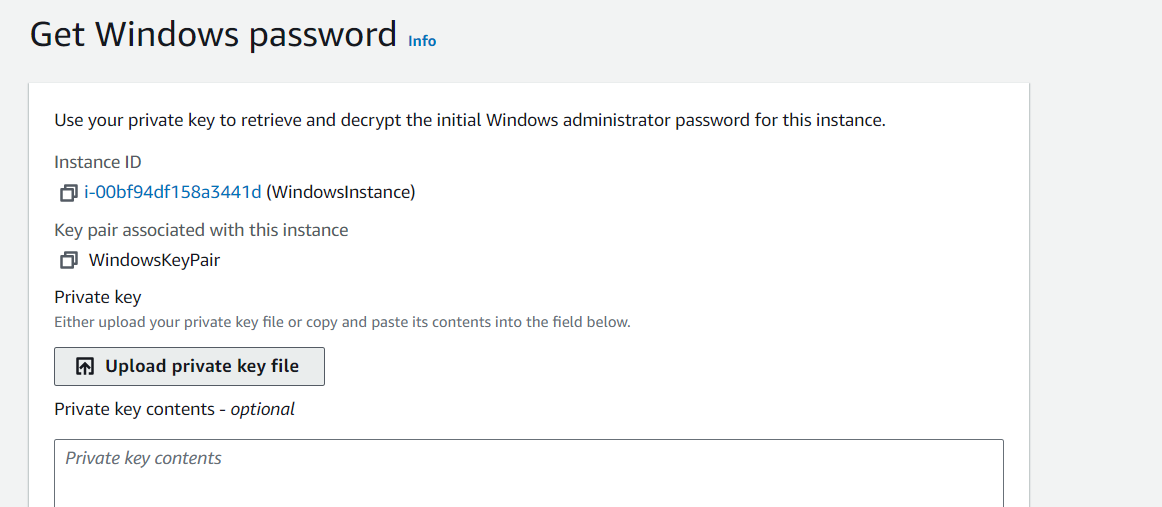


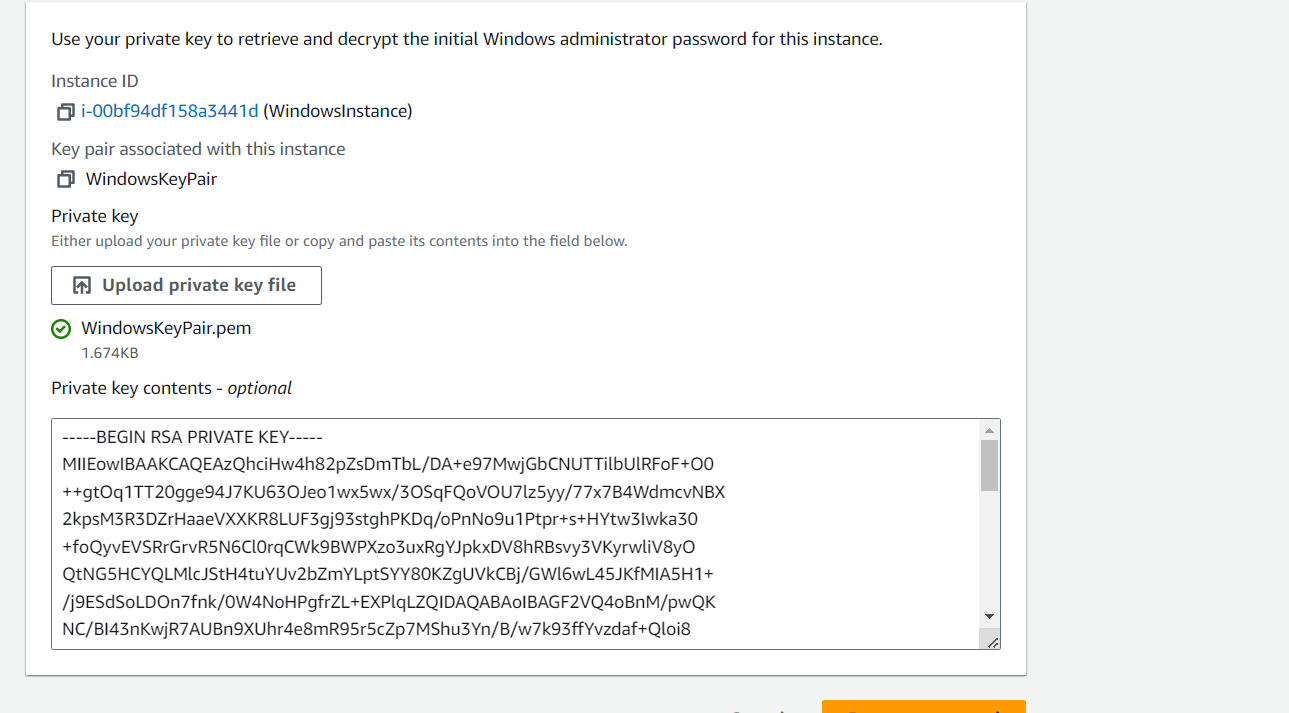
Click on the RDP Client tab and select ‘get password’.



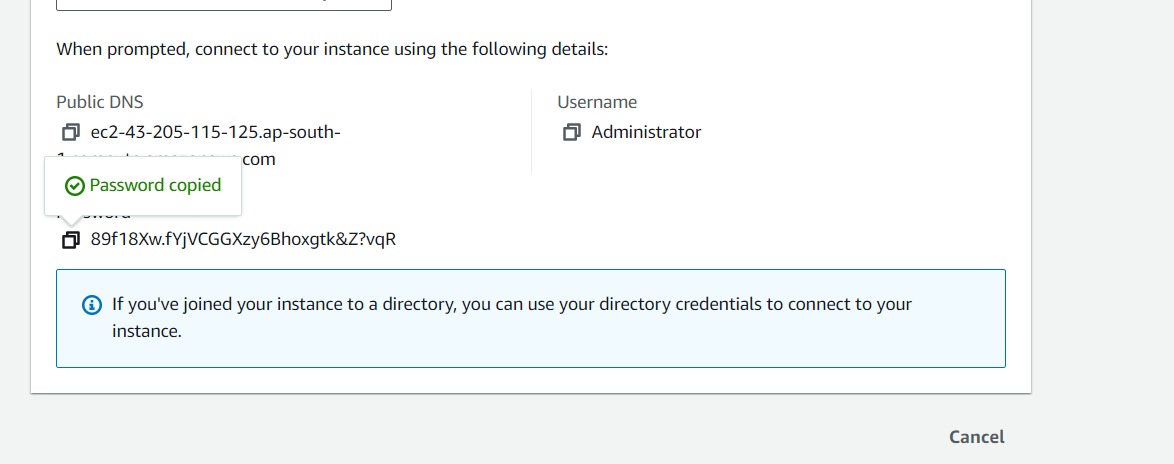


Upload the previously saved .pem file and decrypt the password.



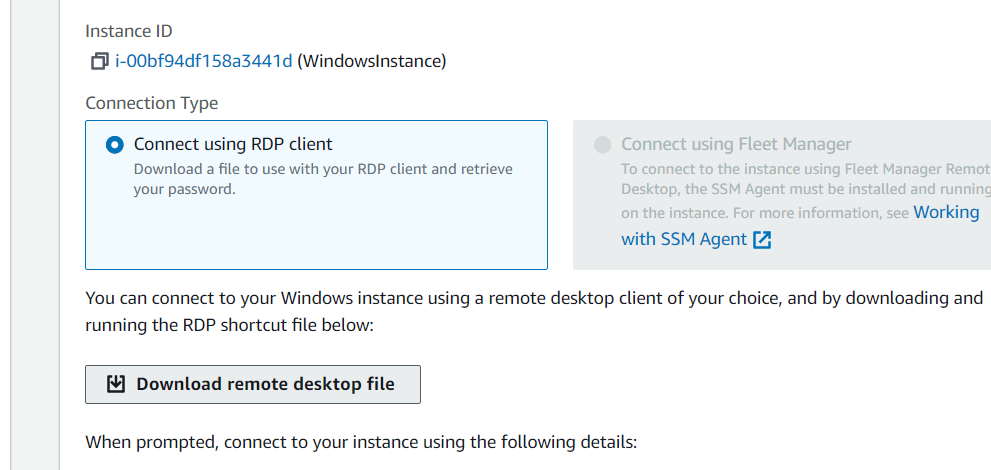


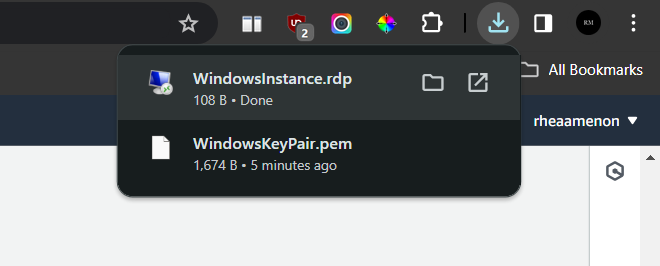
Copy the decrypted password.

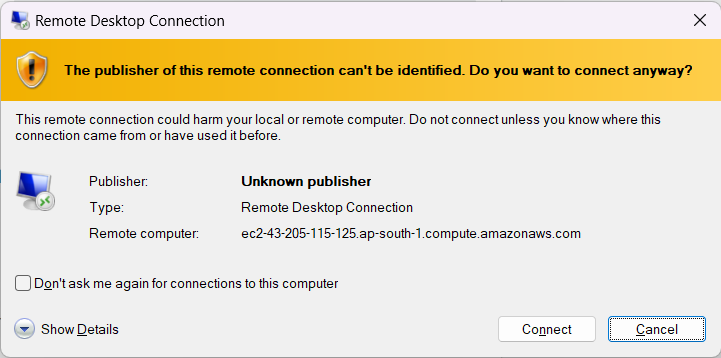


*Password: 89f18Xw.fYjVCGGXzy6Bhoxgtk&Z?vqR*

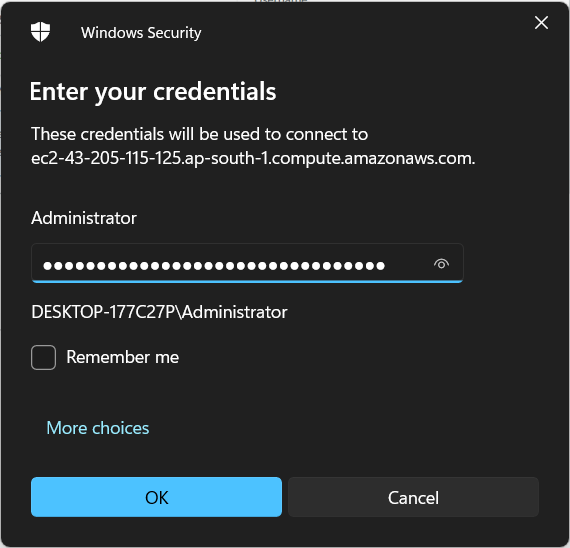
Now download the RDP File and open it.



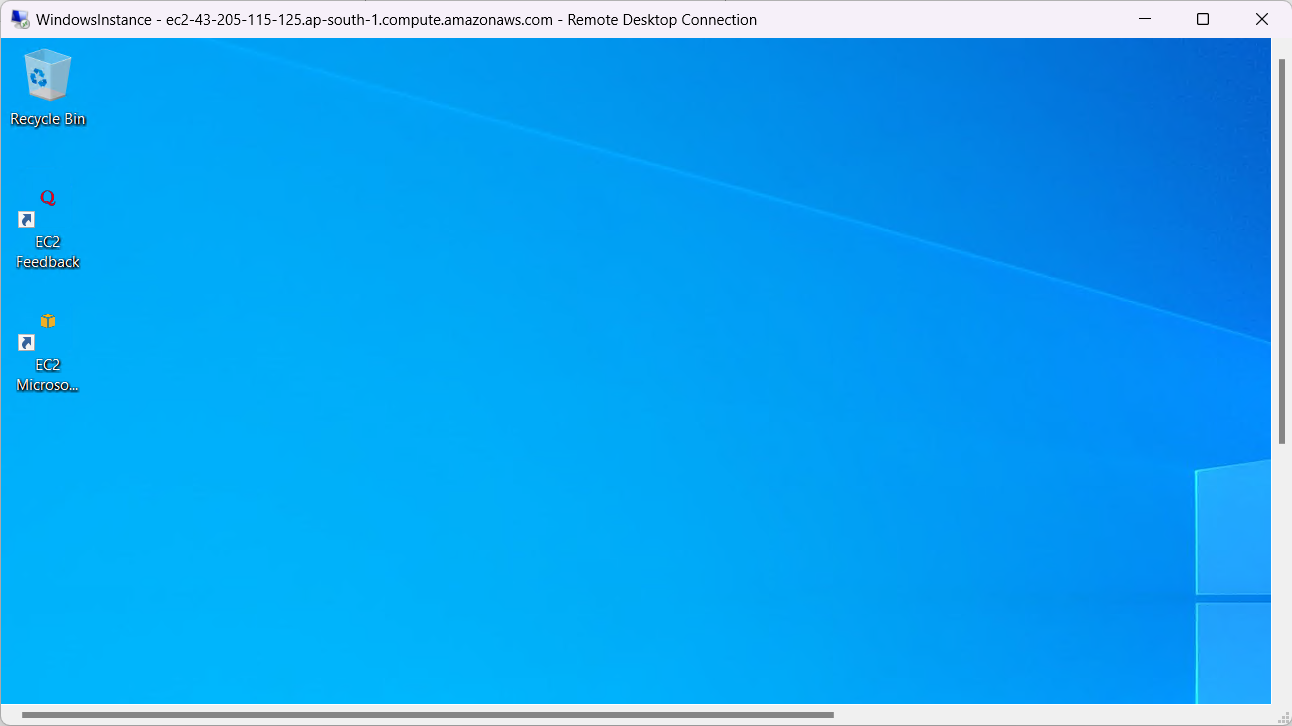




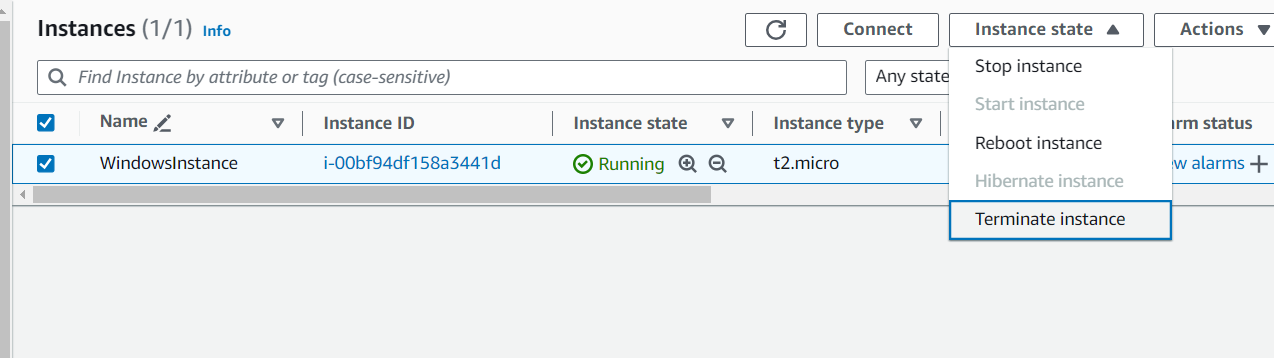
A dialog box will appear to add the password, paste the previously copied password.



The RDP Connection tab will now open with a virtual instance of the Windows OS.

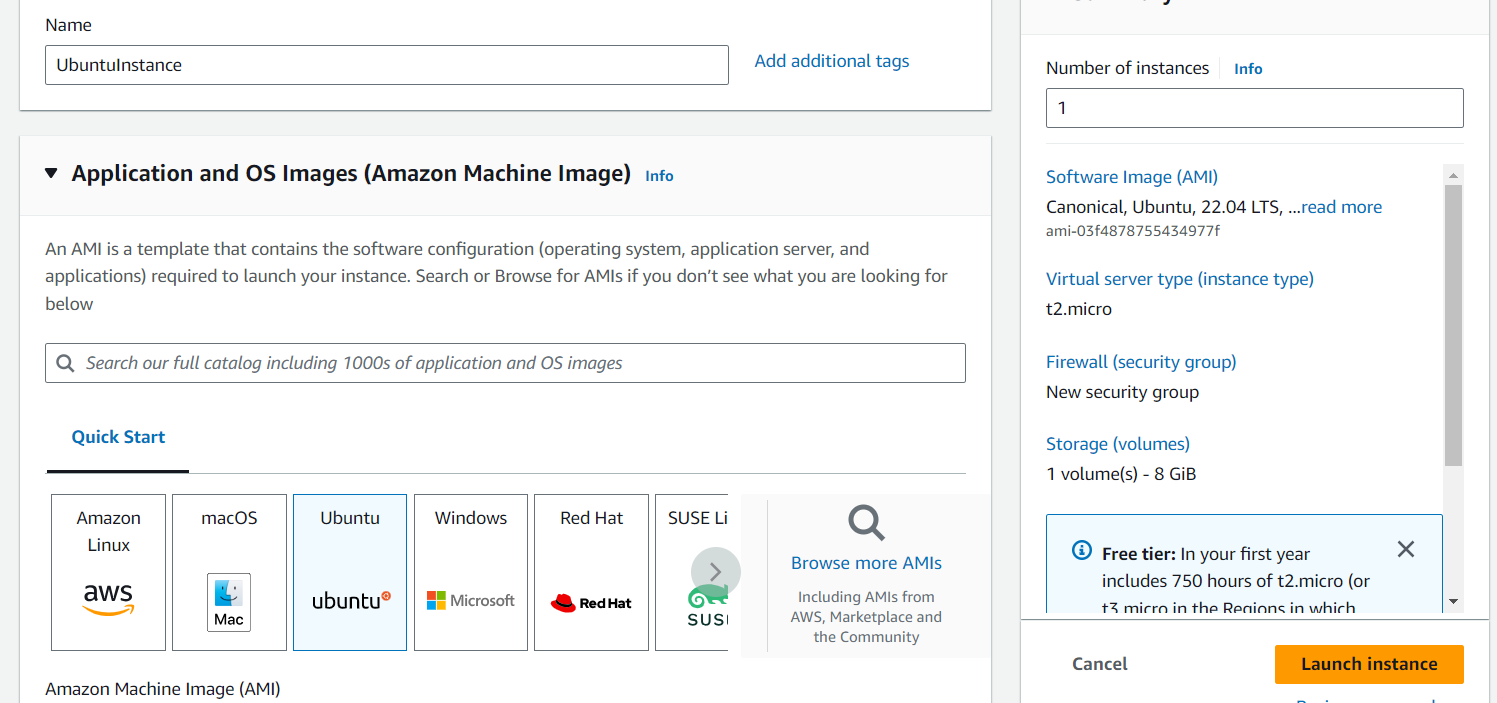


Now terminate the instance by clicking on it > Instance State > Terminate Instance.

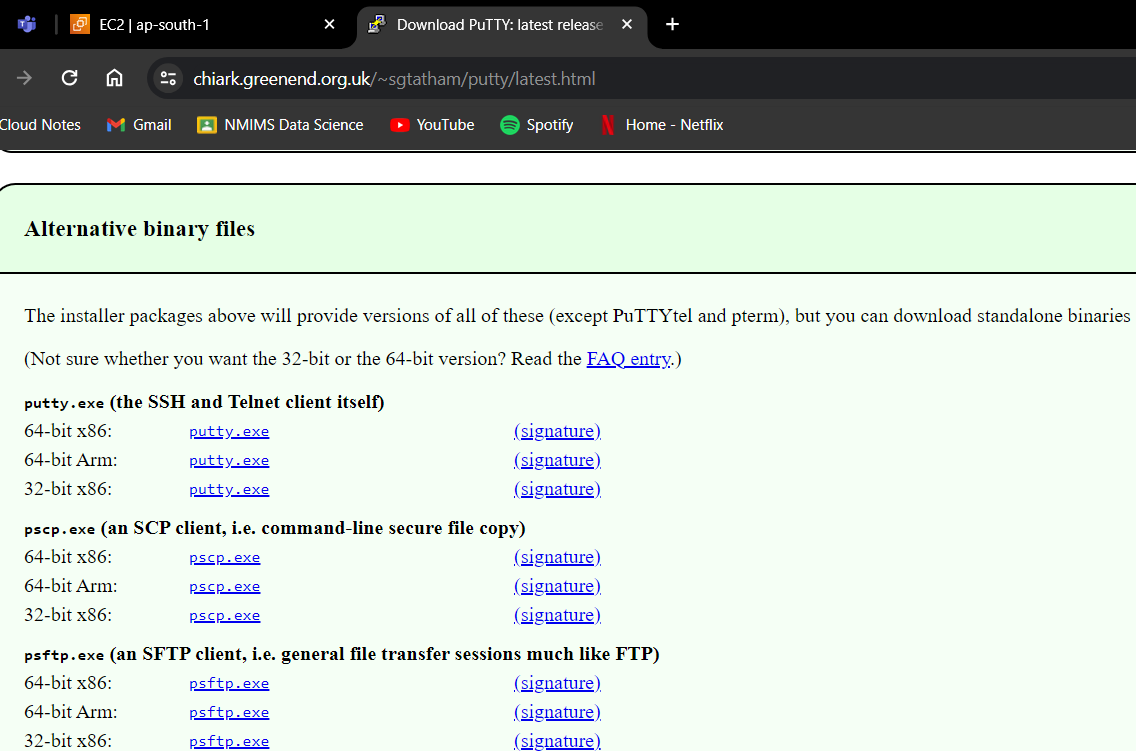


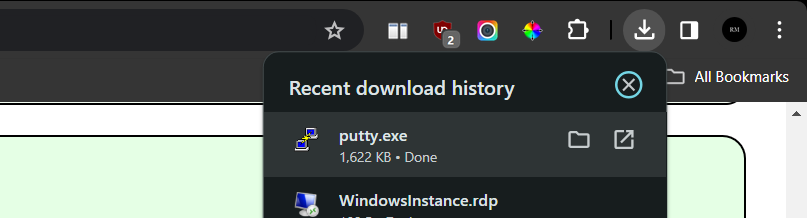
**Linux OS**

Repeat the same steps as for Windows but this time, choose Ubuntu as the OS Image.

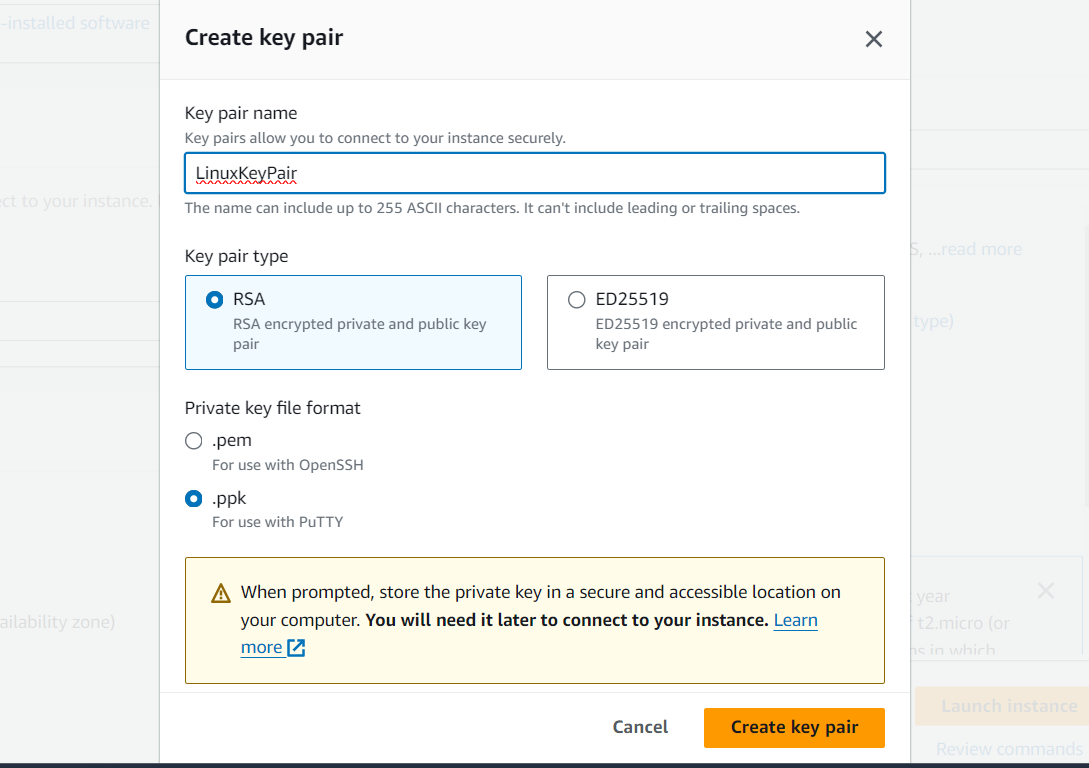


Download putty.exe from the link below.

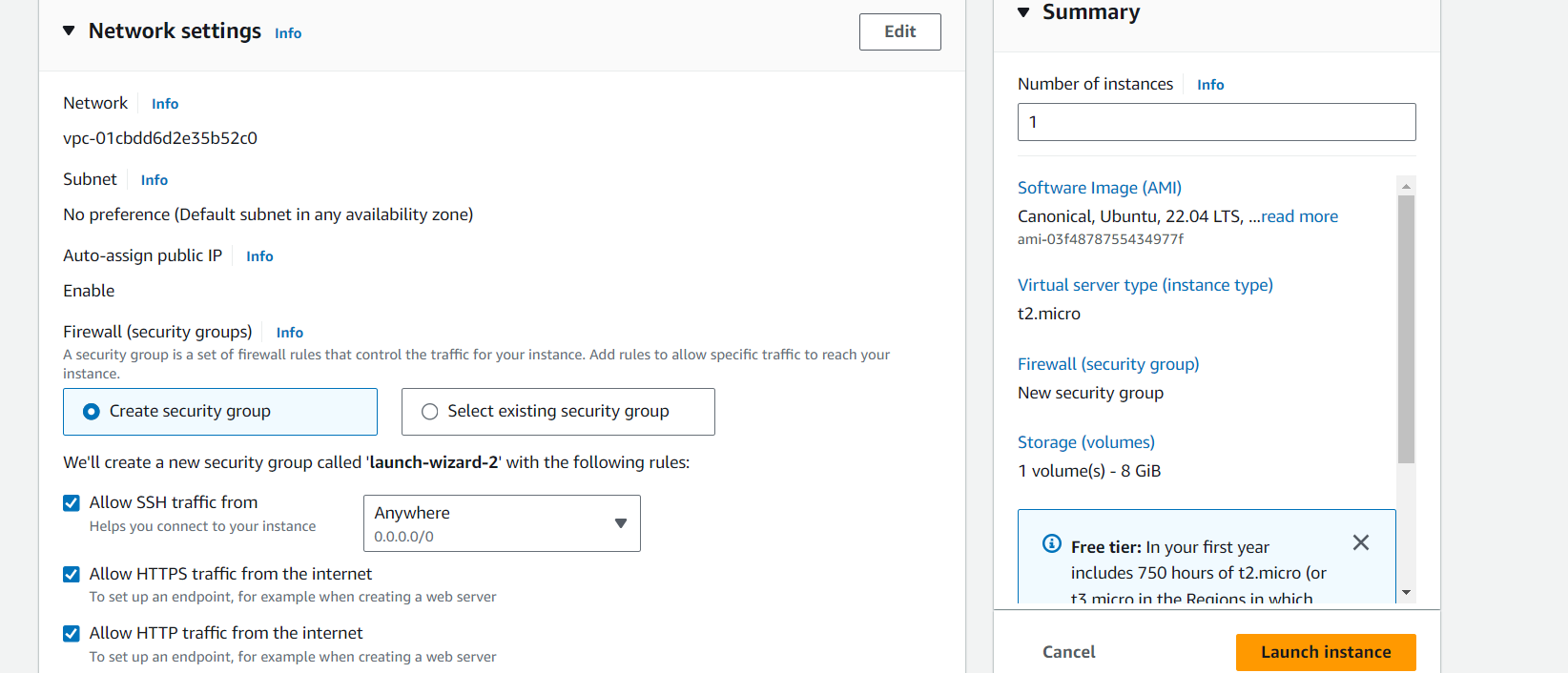




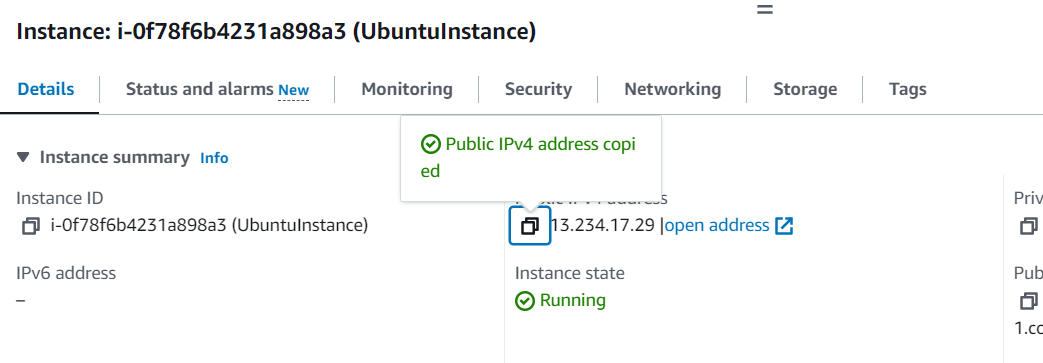
Create a new key pair for this instance and download the .ppk file.



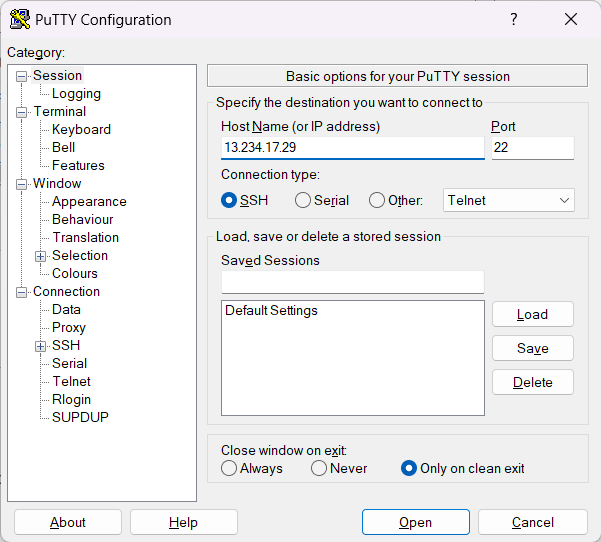
Tick all the boxes for the traffic permissions in network settings.



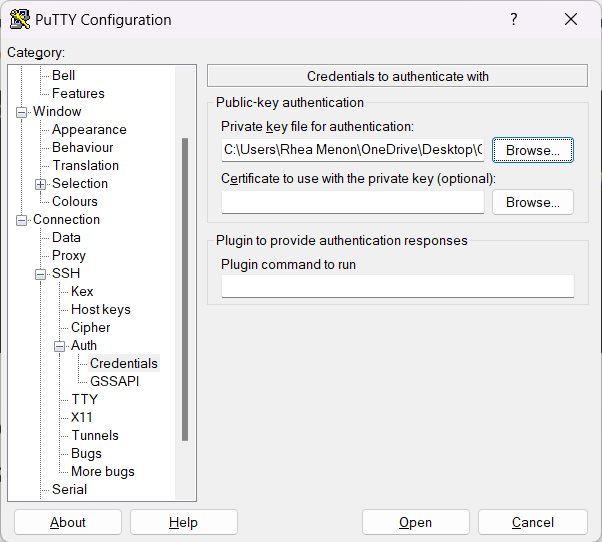
Launch the instance and in the instance summary, copy the public IPv4 address.



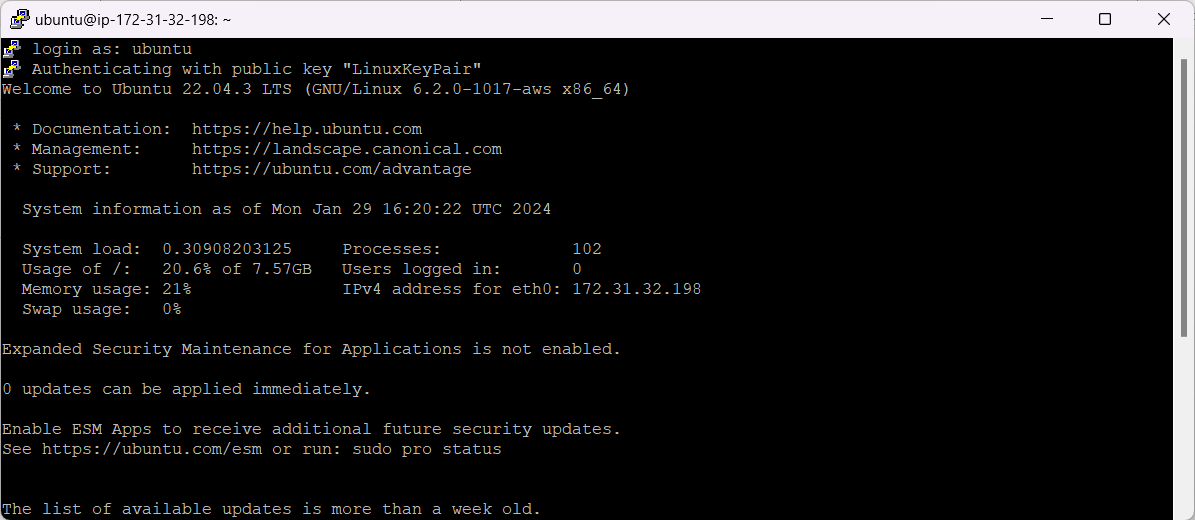
Open PuTTY and add the copied IP address to the Host IP address text box.



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



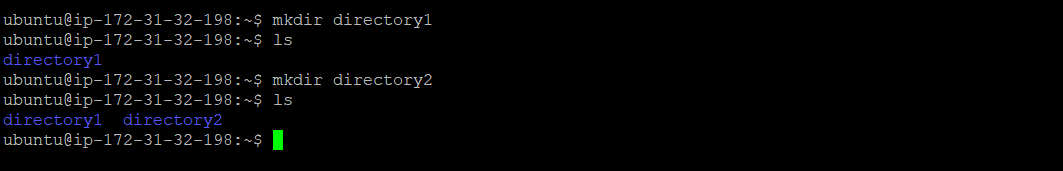
Click open and ubuntu console will be launched. Login as *ubuntu*.



**Commands:**

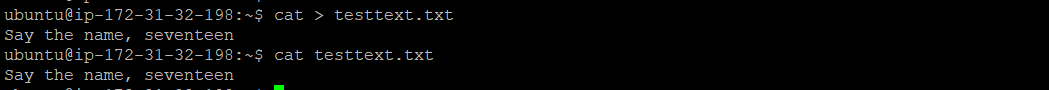
mkdir [directory name]: makes a directory of the same name.

ls: lists all the files and directories in the current working directory.

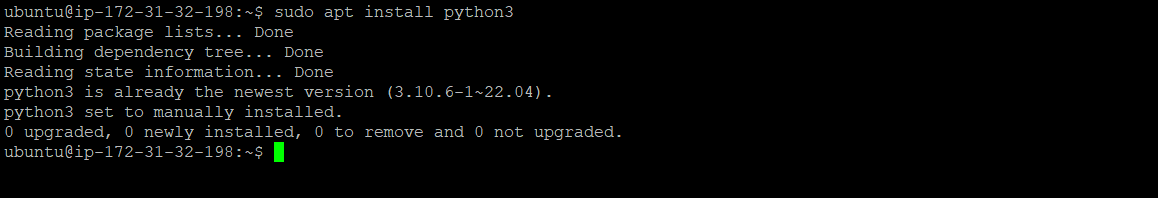


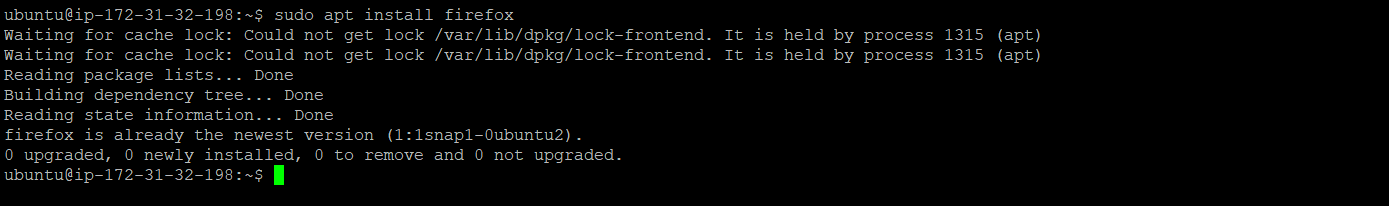
cat > [file name]: append text to a file.

cat [file name]: view the contents of a text file.



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





exit: closes the console.

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

