**AWS Services**

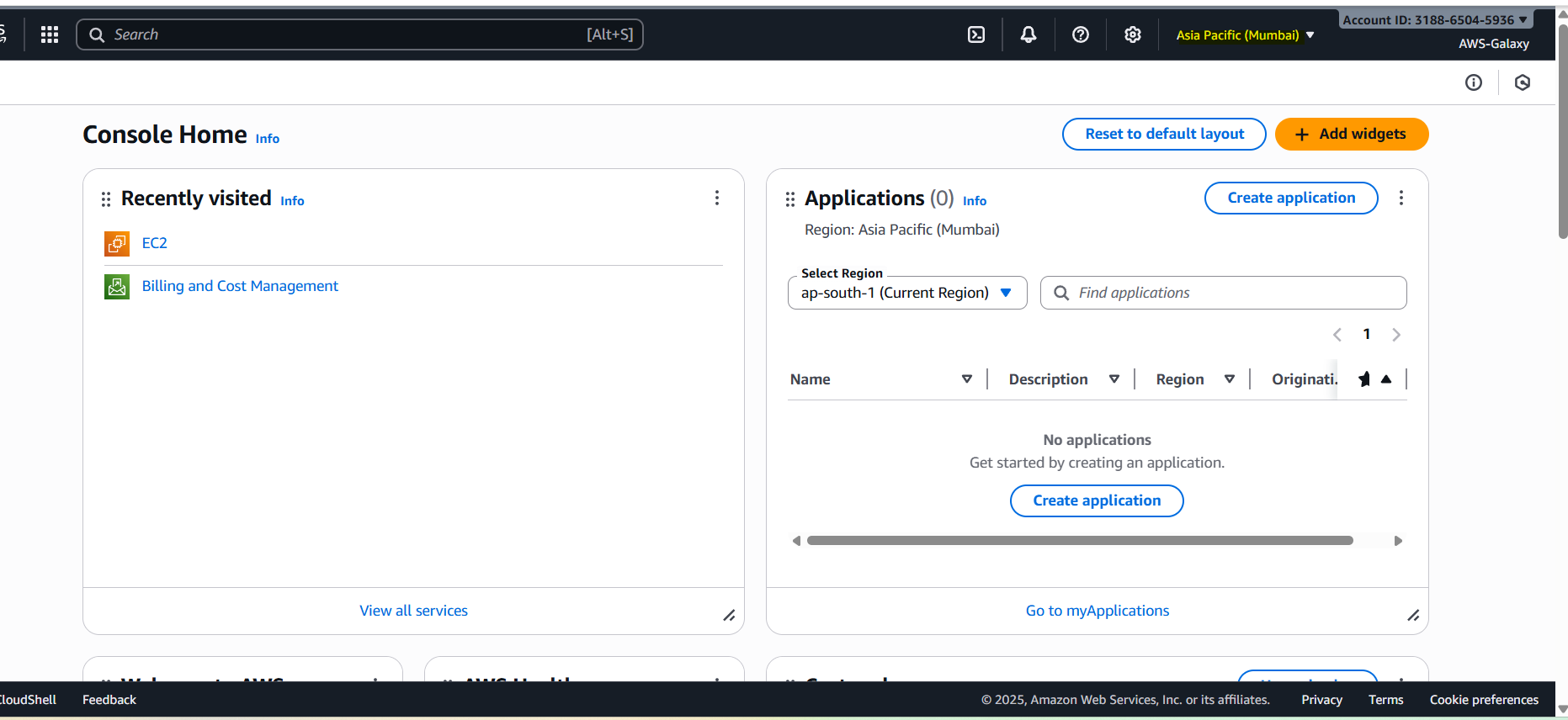
**Amazon EC2 (Elastic Compute Cloud)**

It is **a core AWS service that provides scalable virtual servers in the cloud, known as EC2 instances.** These instances allow users to run applications without investing in physical hardware, offering flexibility, scalability, and cost-efficiency.

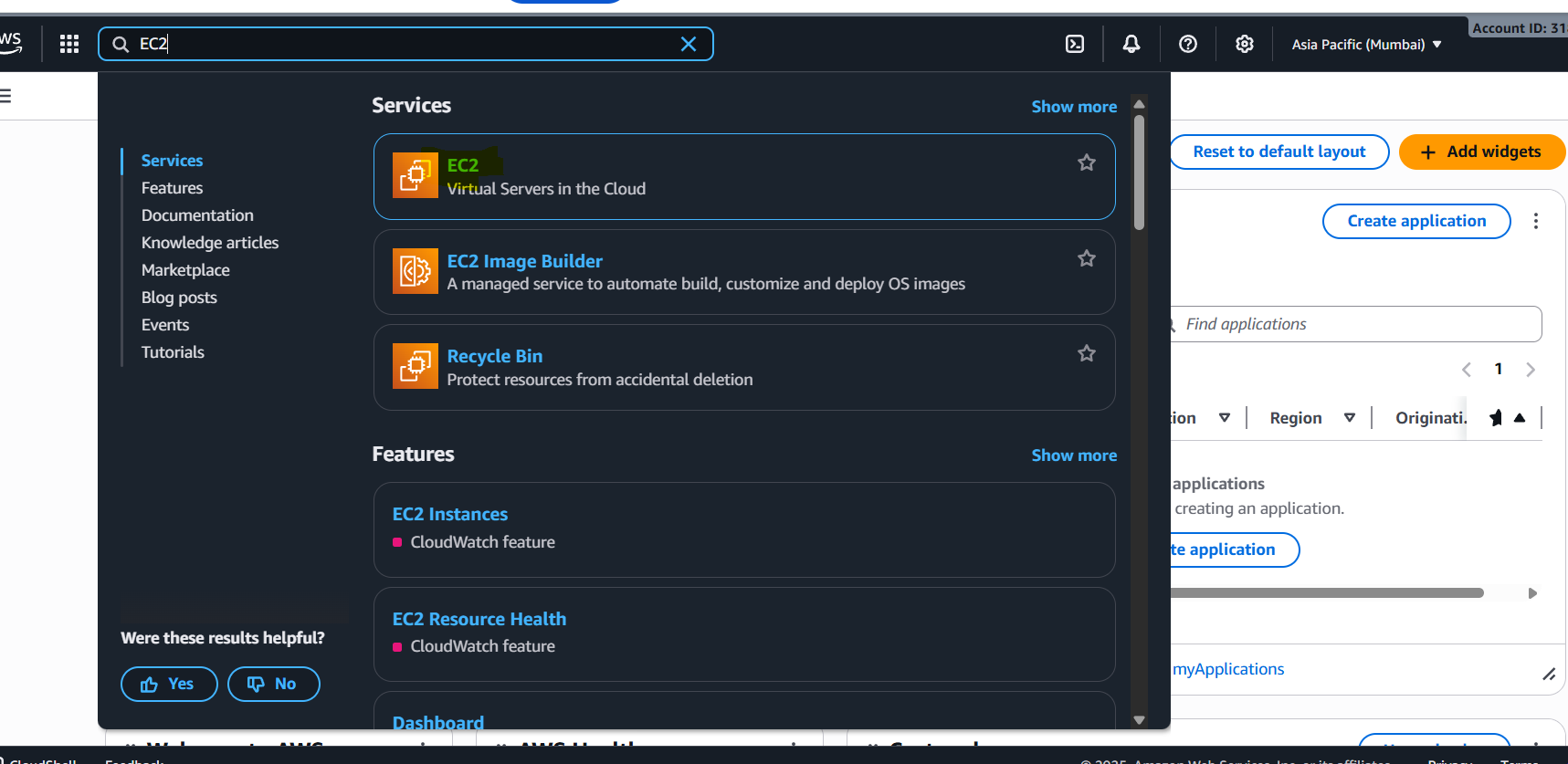
**Rec 3-**

**How to create windows EC2 machine?**

1. Login to AWS console 🡪 [Amazon Web Services Sign-In](https://signin.aws.amazon.com/signin?client_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3FhashArgs%3D%2523%26isauthcode%3Dtrue%26nc2%3Dh_si%26src%3Dheader-signin%26state%3DhashArgsFromTB_eu-north-1_7e5c9f71b3d0a0ec&page=resolve&code_challenge=3842HMtBFa38f8b8nikuR4XHFIfJO82RwiDVtOqVRmA&code_challenge_method=SHA-256&backwards_compatible=true)
2. Login as root user.
3. Select near by **region** where you want to work

****

1. Search for EC2 and select it.

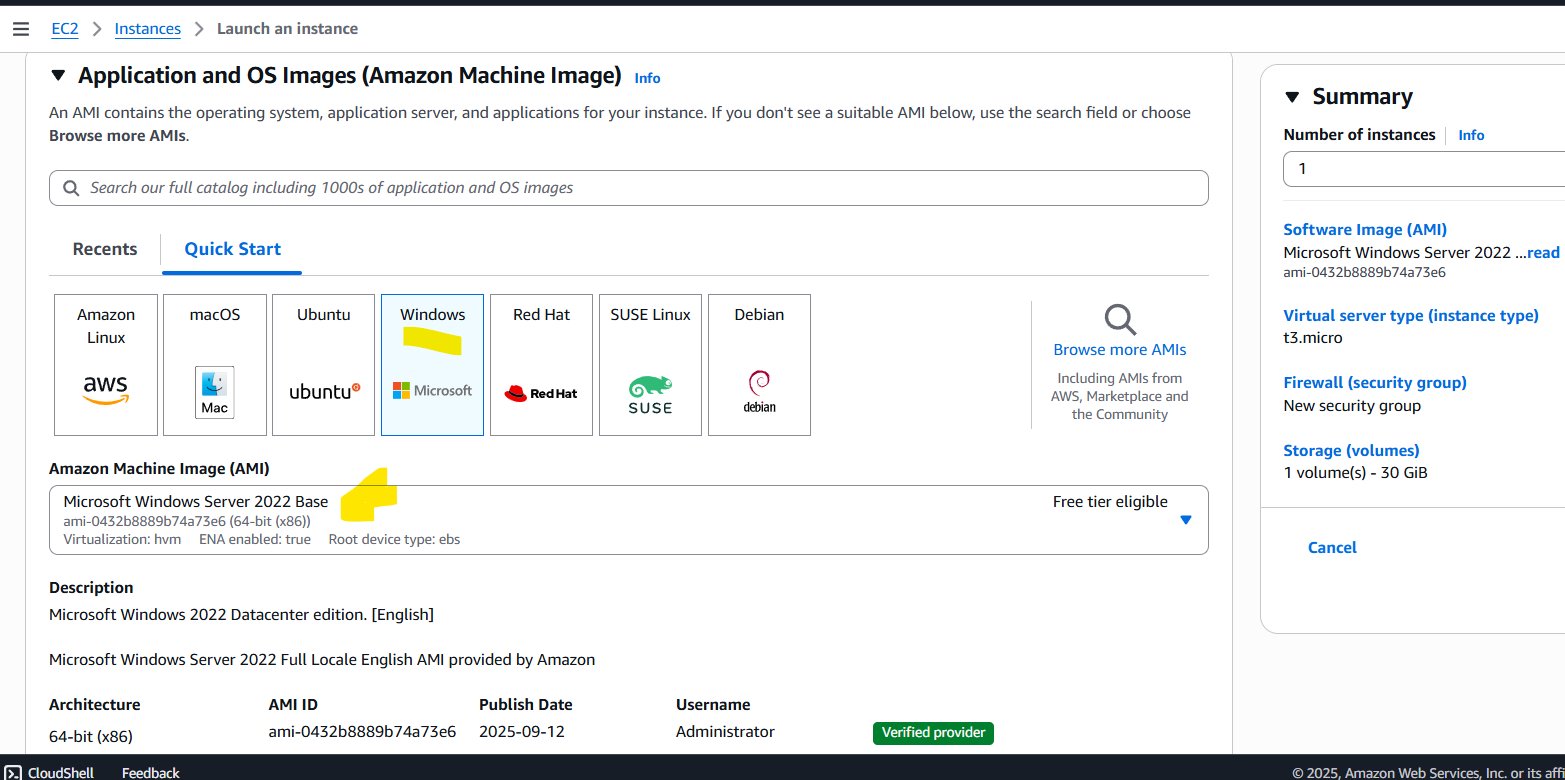


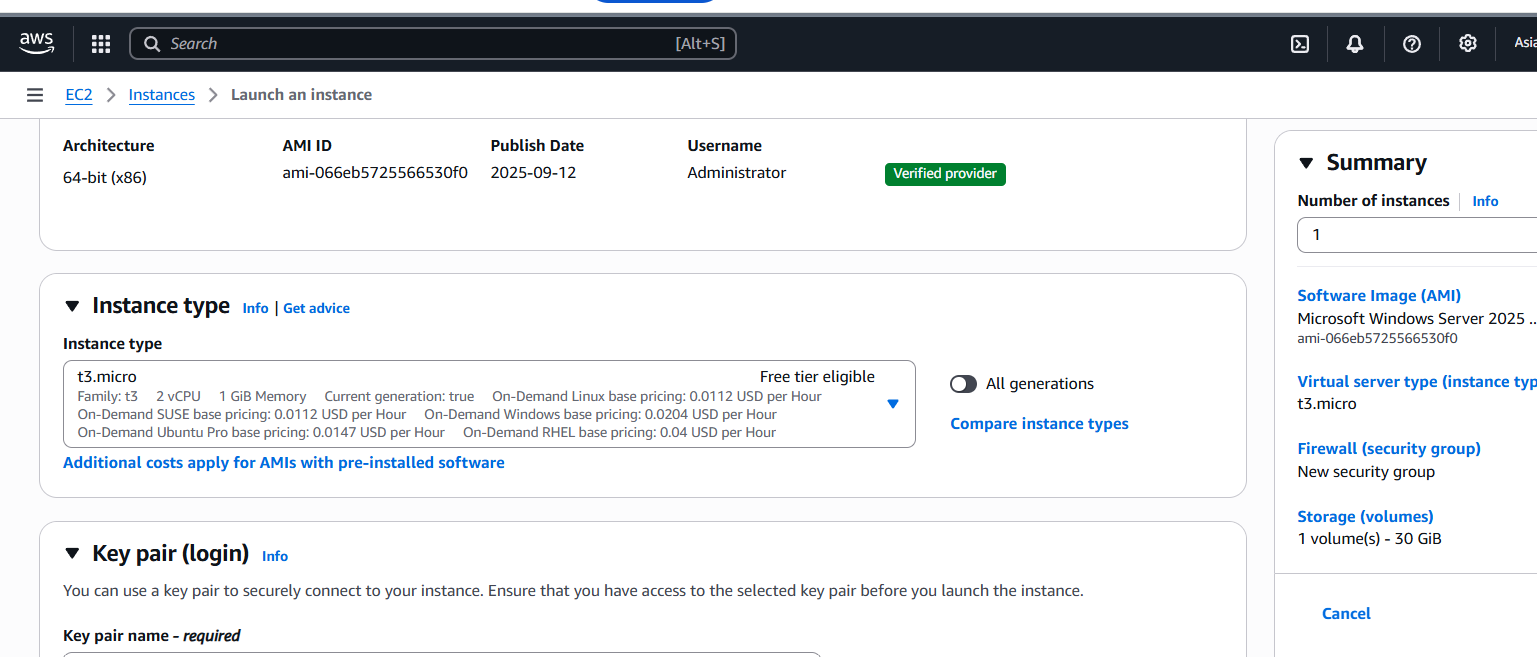
1. Click on launch instanceA screenshot of a computer

   AI-generated content may be incorrect.
2. Provide the unique name to your EC2 machine.

A screenshot of a computer

AI-generated content may be incorrect.

1. Select Operating system. Operating system also known as AMI (Amazon Machine Image). From the dropdown you can select any AMI as per you requirement. 
2. Select instance type. Instance type is group where we will get cpu and ram together. And every instance type have name.



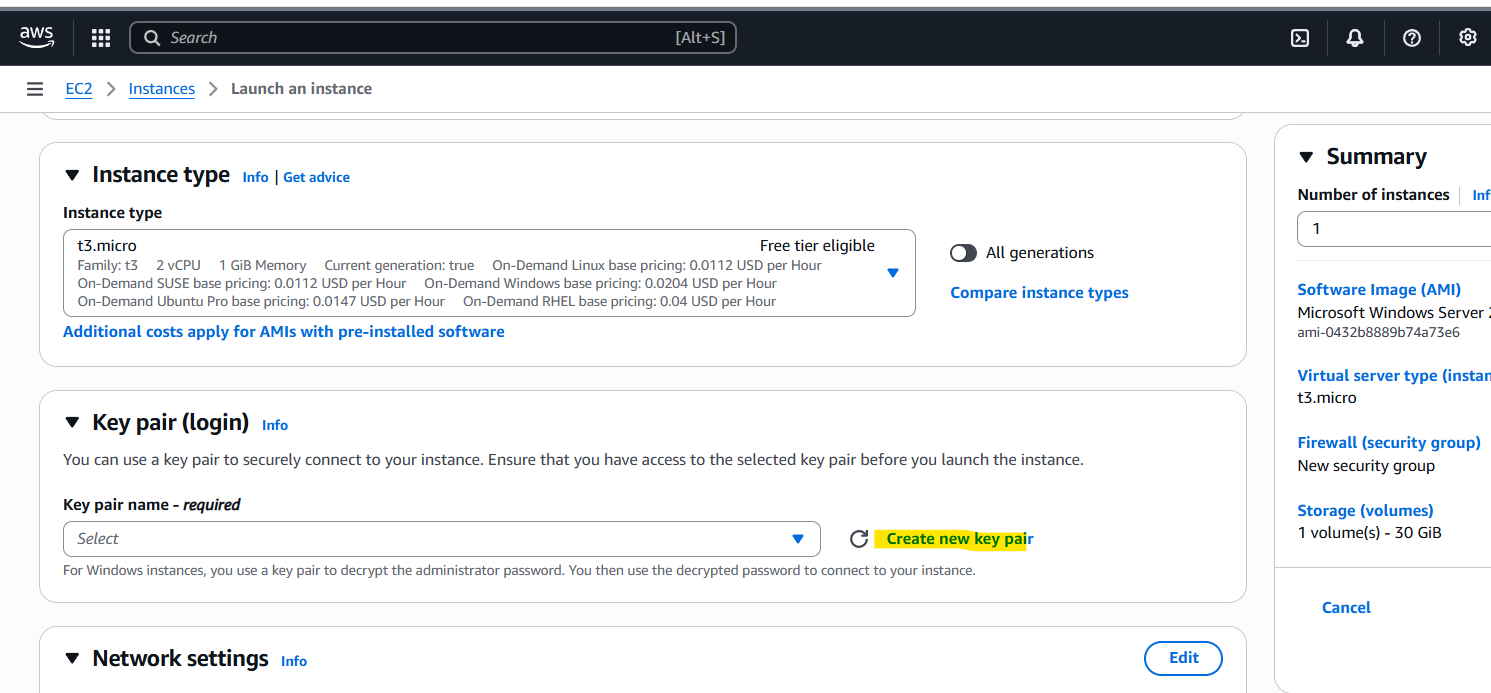
1. Key pair:

Key pair help us to login our EC2 machine. Combination of 2keys.

**Public key**🡪stored by AWS

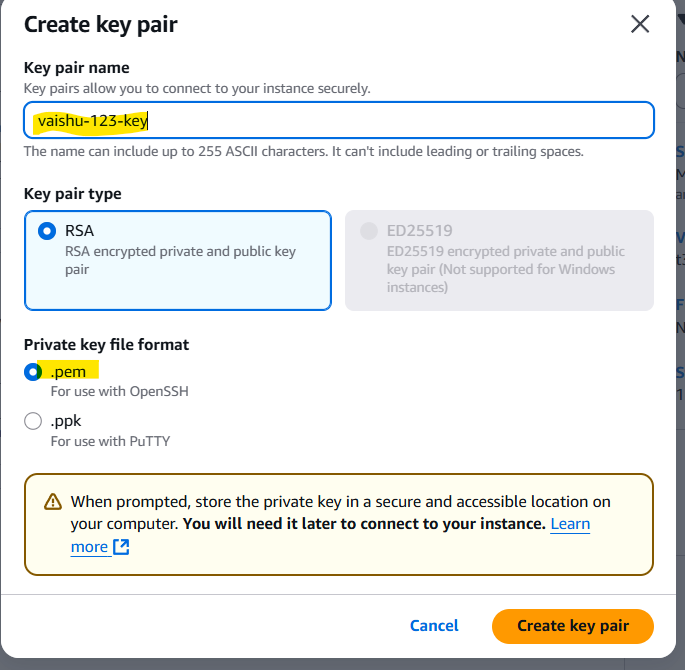
**Private key**🡪 stored by Local

To create key pair click on Create new key pair



Provide name and select file format as **.pem (Privacy Enhanced Mail).** Click on Create key pair.

**Note:** Remember you will get single chance to download it.



1. Network setting section.

* **Security group**-- basically a virtual firewall that controls **inbound** and **outbound** traffic to your EC2 instance.

| **Type** | **Direction** | **Example Use Case** |
| --- | --- | --- |
| **Inbound** | Traffic **coming into** your instance | Someone connecting via SSH (port 22) or accessing your website (port 80) |
| **Outbound** | Traffic **going out from** your instance | Your server downloading updates or calling an external API |

* Under security group we are going to enable ports which help us to connect to our EC2 machine. To connect to your windows machine, you have to enable one port. As we selected windows os by default AWS enabled RDP port.
* Check Allow RDP(Remote Desktop Protocol) Traffic from.
* Port number of RDP🡪 **3389**
* Select source as Anywhere. I can connect to EC2 machine from any part of location

A screenshot of a computer

AI-generated content may be incorrect.

1. Configure storage

By default it is 30gb. We can set it as per our requirement. Min 30gb is required for windows EC2 machine.

**A screenshot of a computer

AI-generated content may be incorrect.**

1. Summary

Here we can set how many machines we want to create. And we can see the details of EC2 machine.

Once you click on launch instance EC2 machine will get created.

A screenshot of a computer

AI-generated content may be incorrect.

1. Here you can see our EC2 machine is created and running

A screenshot of a computer

AI-generated content may be incorrect.

**Generate/Decrypting password:**

1. Select the EC2 Machine & Click on Connect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Go to RDP Client.

Note down public DNS and Username.

Public DNS: ec2-65-0-97-183.ap-south-1.compute.amazonaws.com

Username : Administrator

A screenshot of a computer

AI-generated content may be incorrect.

1. Click on get passwordA screenshot of a computer

   AI-generated content may be incorrect.
2. Upload .pem file generated during key pair creation

A screenshot of a computer

AI-generated content may be incorrect.

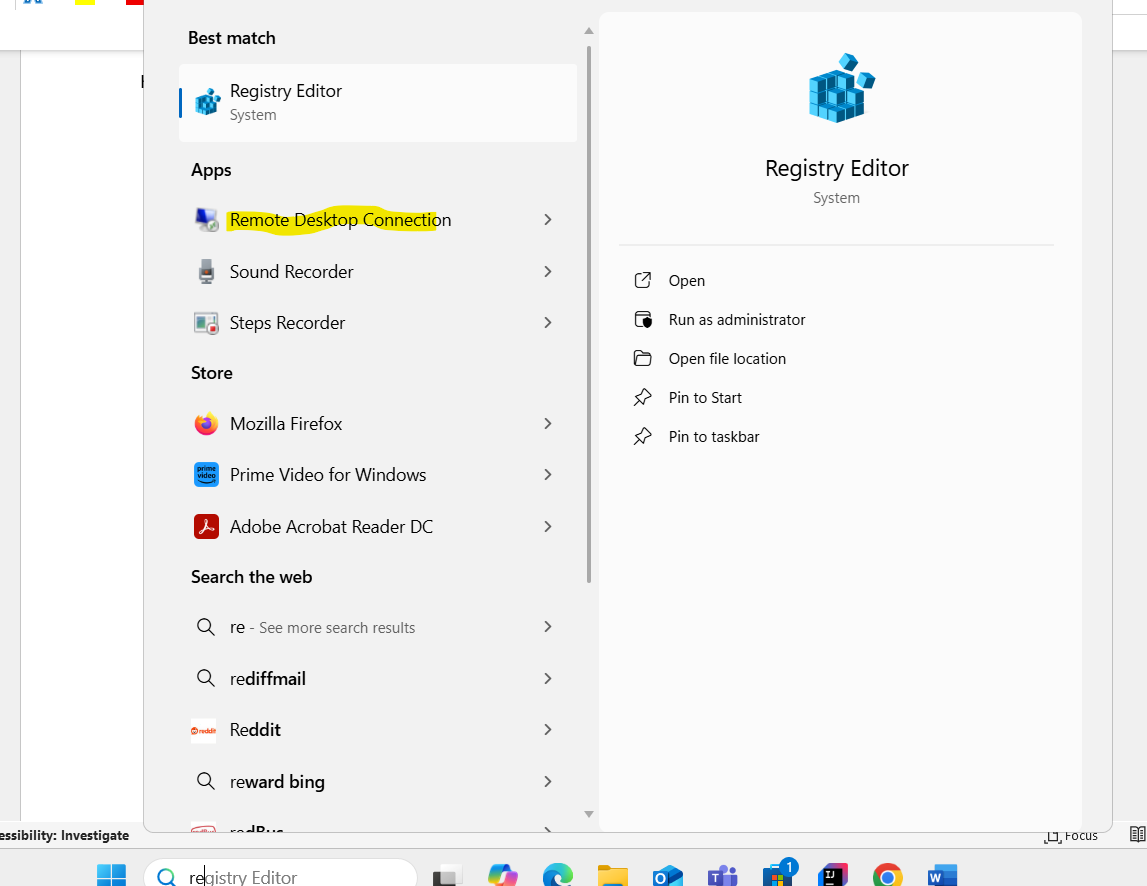
1. Click on Decrypt Password. Password will get generated.

Password: NK=!Qi(k)APjR4CU&=2IBGOfS%D=x59Y

A screenshot of a computer

AI-generated content may be incorrect.

**How to connect to our windows EC2 machine?**

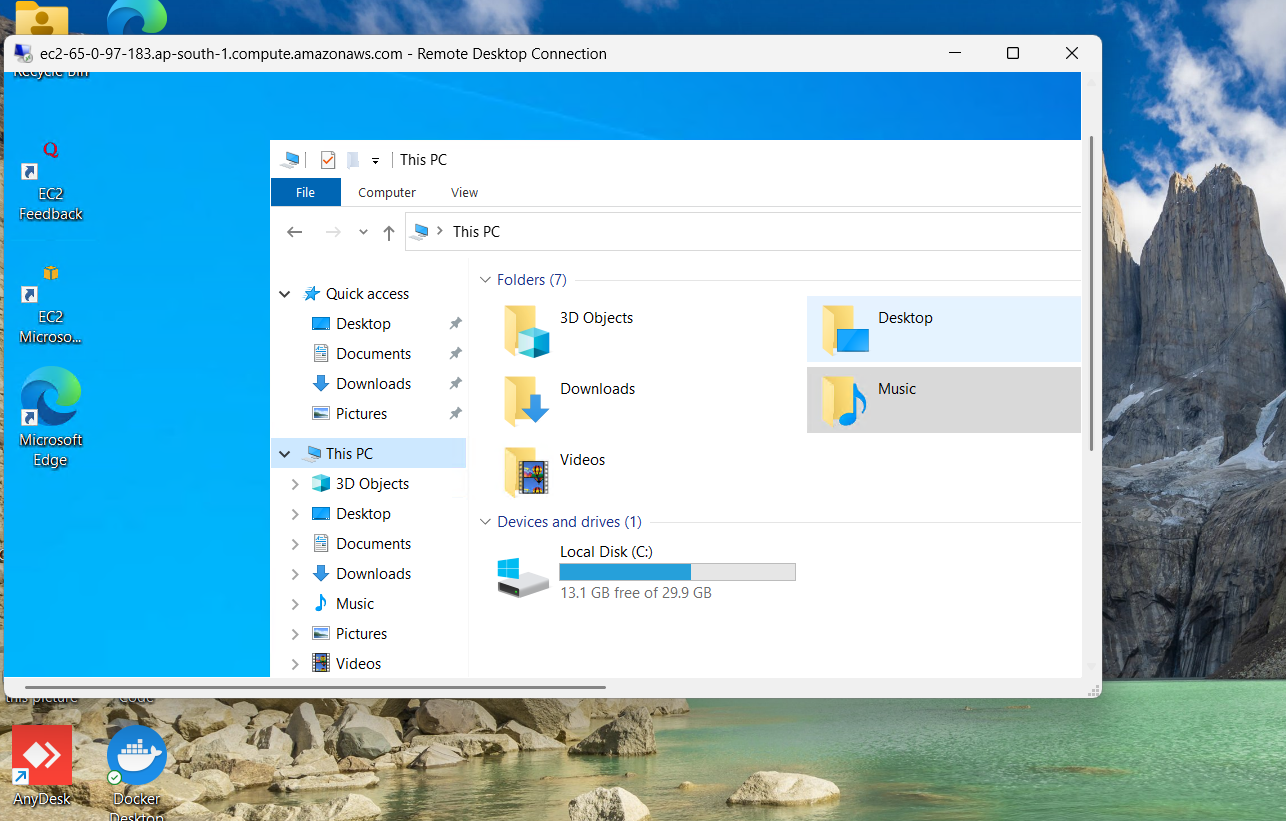
1. Open Remote Desktop Connection
2. Enter the Public DNS click on connect A screenshot of a computer

   AI-generated content may be incorrect.
3. Enter Username and Generated Password

A screenshot of a computer

AI-generated content may be incorrect.

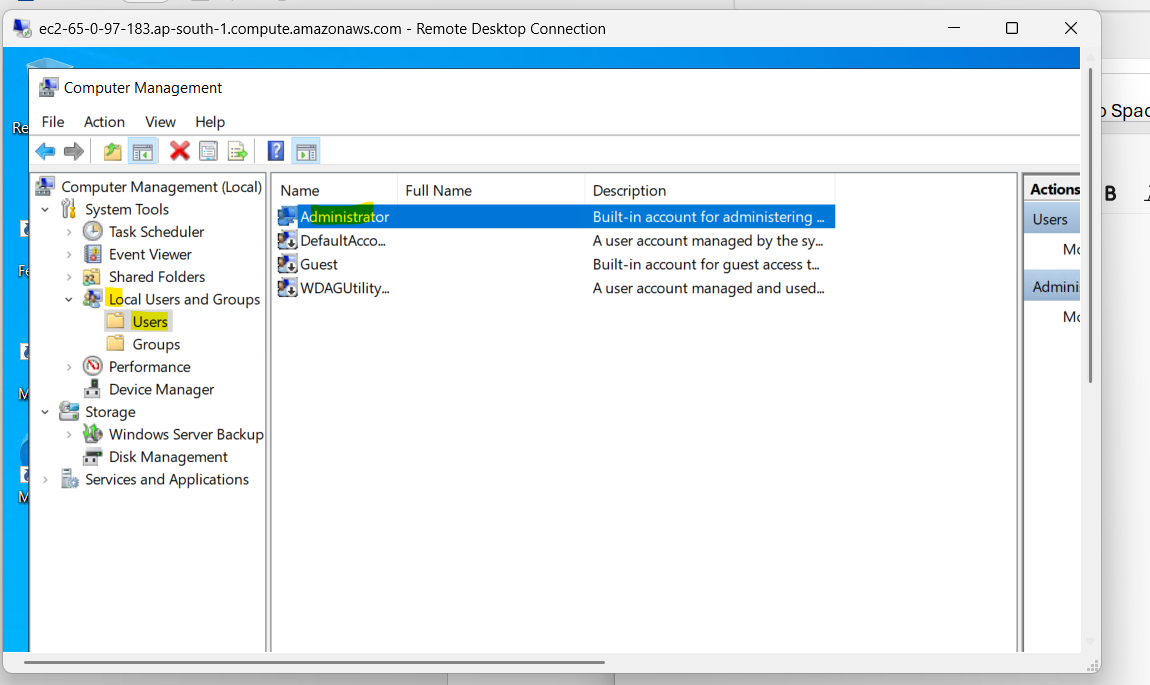
1. You are successfully connected to EC2 machine



**How to set our own password?**

1. Login to your EC2 machine.
2. Seach type run
3. There run below command

**compmgmt.msc**

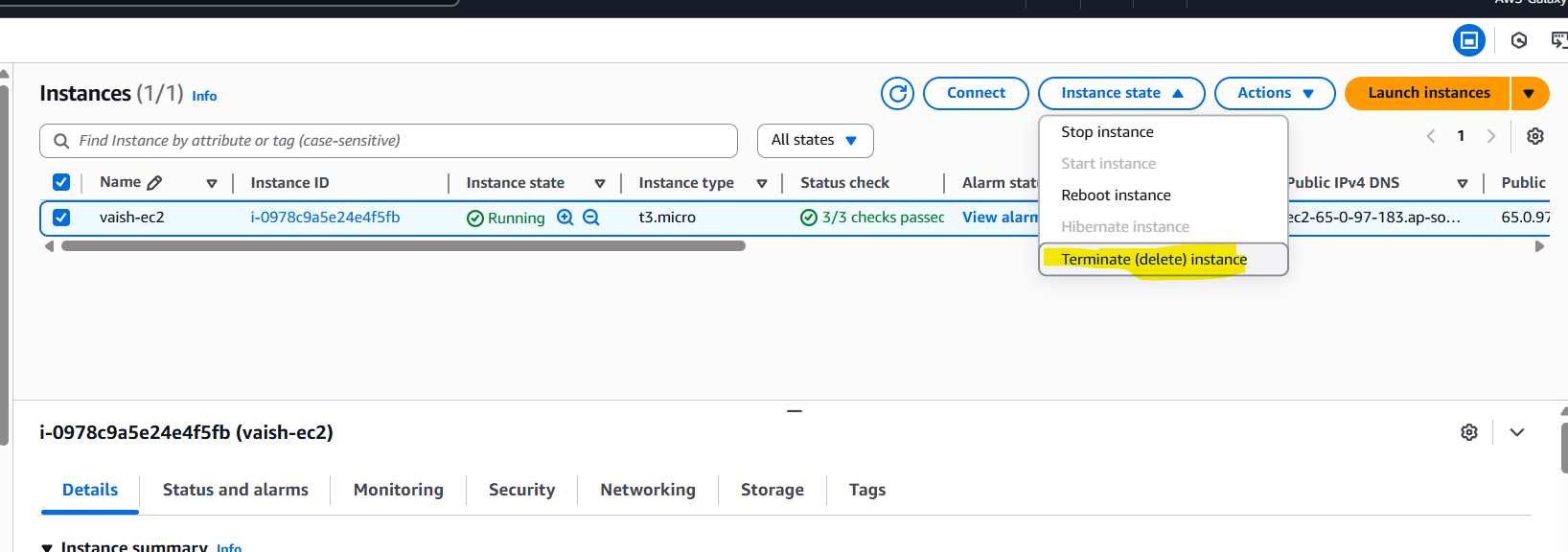
1. Click on local users 
2. Right click on username and set password

A screenshot of a computer

AI-generated content may be incorrect.

1. Provide the password which you want to set.
2. Try to login your password will be successfully set.

**How to terminate EC2 machine?**



A screenshot of a computer

AI-generated content may be incorrect.

**Questions**

**What Is a .pem File?**

**PEM** stands for **Privacy Enhanced Mail**, but today it's mostly used for **cryptographic data** like certificates and keys—not email.

A .pem file is a **text-based format** that stores cryptographic elements in a **Base64-encoded** structure, wrapped with specific header and footer lines.

**Structure of a PEM File:**

|  |
| --- |
| -----BEGIN CERTIFICATE-----  (base64 encoded data)  -----END CERTIFICATE----- |

Other types of PEM blocks include:

* -----BEGIN RSA PRIVATE KEY-----
* -----BEGIN PUBLIC KEY-----
* -----BEGIN CERTIFICATE REQUEST-----

Each block tells you what kind of data is inside.

**Commonly used:**  
PEM files are widely used in:

* **SSL/TLS certificates** for securing websites
* **SSH keys** for secure server access
* **Certificate chains** that include root, intermediate, and leaf certificates
* **Web servers** like Apache and Nginx to configure HTTPS

**Why They Matter?**

PEM files are:

* **Human-readable** (you can open them in Notepad or any text editor)
* **Flexible** (can contain multiple certificates or keys in one file)
* **Standardized** (used across many platforms and tools like OpenSSL)

**Rec 4-**

**Linux EC2 machine**

**How to create Linux EC2 machine?**

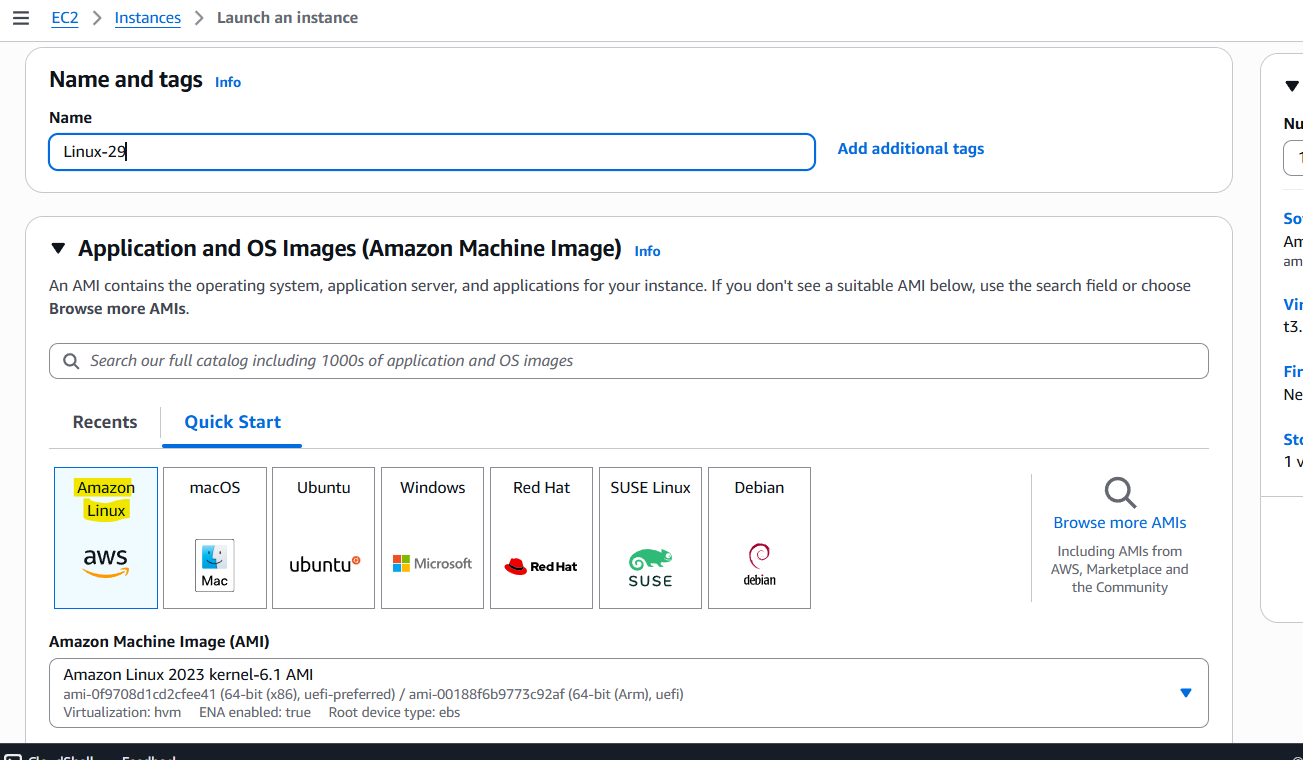
For creating Linux ec2 machine process is same as Windows ec2 machine just difference is selecting OS.

Steps:

1. Login to console. Search EC2.
2. Click on launch instance
3. Give unique name
4. Select Amazon Linux OS (or any flavour of Linux os as per your requirement)

There are different linux flavours are available. Each one has its ows functionality. Liux is open source and free operating system so other organinizations used there code and build there own operating systems. AWS console defaultly it shows 5 flavours. Refere below table for deciding which one is better for you according to functionality. 1st five are **officially maintained and supported** by AWS or their vendors. They’re **frequently updated** and tested for compatibility with EC2 infrastructure.

| **Distribution** | **Description** | **Best For** |
| --- | --- | --- |
| **Amazon Linux** | AWS's own distro, optimized for EC2 with tight integration and regular updates. | General-purpose workloads, AWS-native apps |
| **Ubuntu** | Popular and user-friendly, with strong community support. | Web servers, development environments |
| **Red Hat Enterprise Linux (RHEL)** | Enterprise-grade, stable, and secure with paid support. | Corporate apps, compliance-heavy environments |
| **Debian** | Stable and minimal, with a strong focus on free software. | Lightweight servers, custom builds |
| **SUSE Linux Enterprise Server (SLES)** | Enterprise-focused with strong support for SAP workloads. | SAP, enterprise-grade systems |
| **Fedora** | Cutting-edge features, upstream of RHEL. | Testing, development |
| **Arch Linux** | Minimal and rolling release, not officially supported but available via community AMIs. | Advanced users, custom setups |
| **CentOS / Rocky Linux / AlmaLinux** | Free alternatives to RHEL, community-supported. | Cost-effective enterprise workloads |

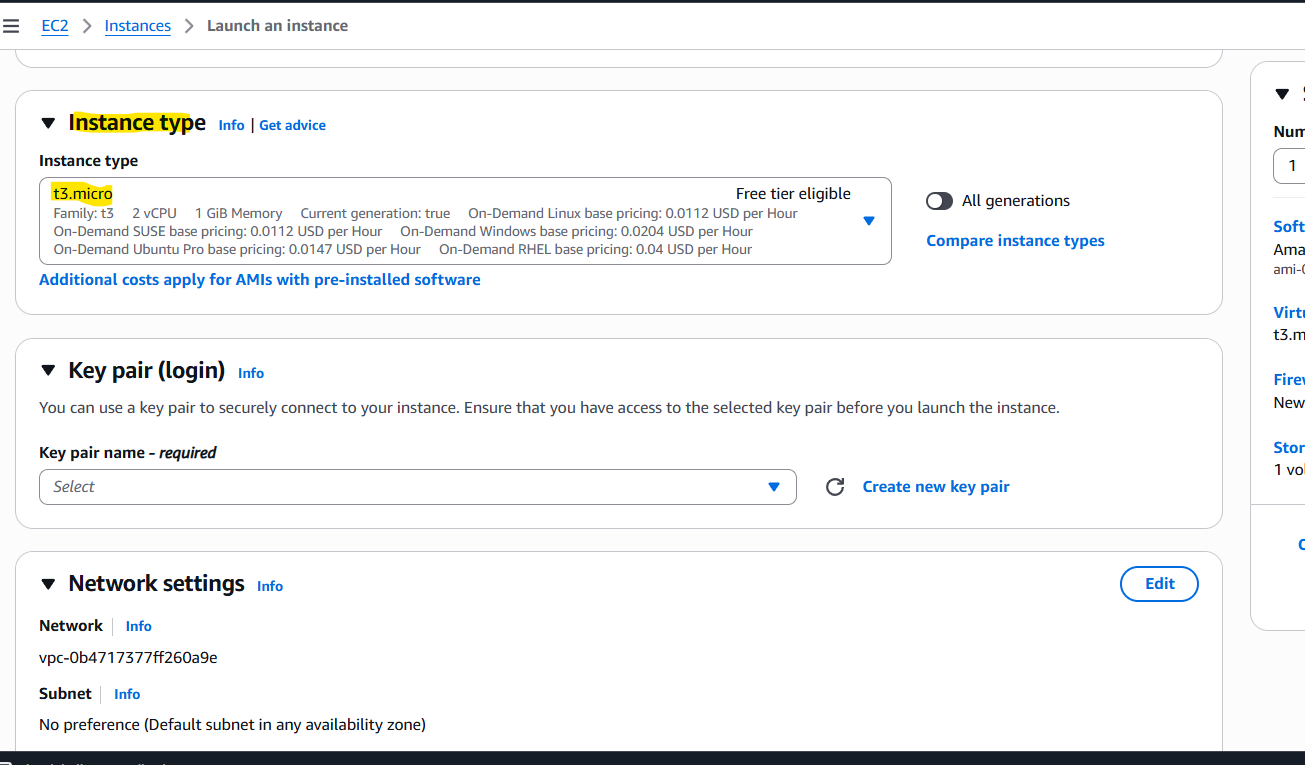


1. Select appropriate image from dropdown

A screenshot of a computer

AI-generated content may be incorrect.

1. Select instance type.



1. Key pair you can use existing one or you can create new.

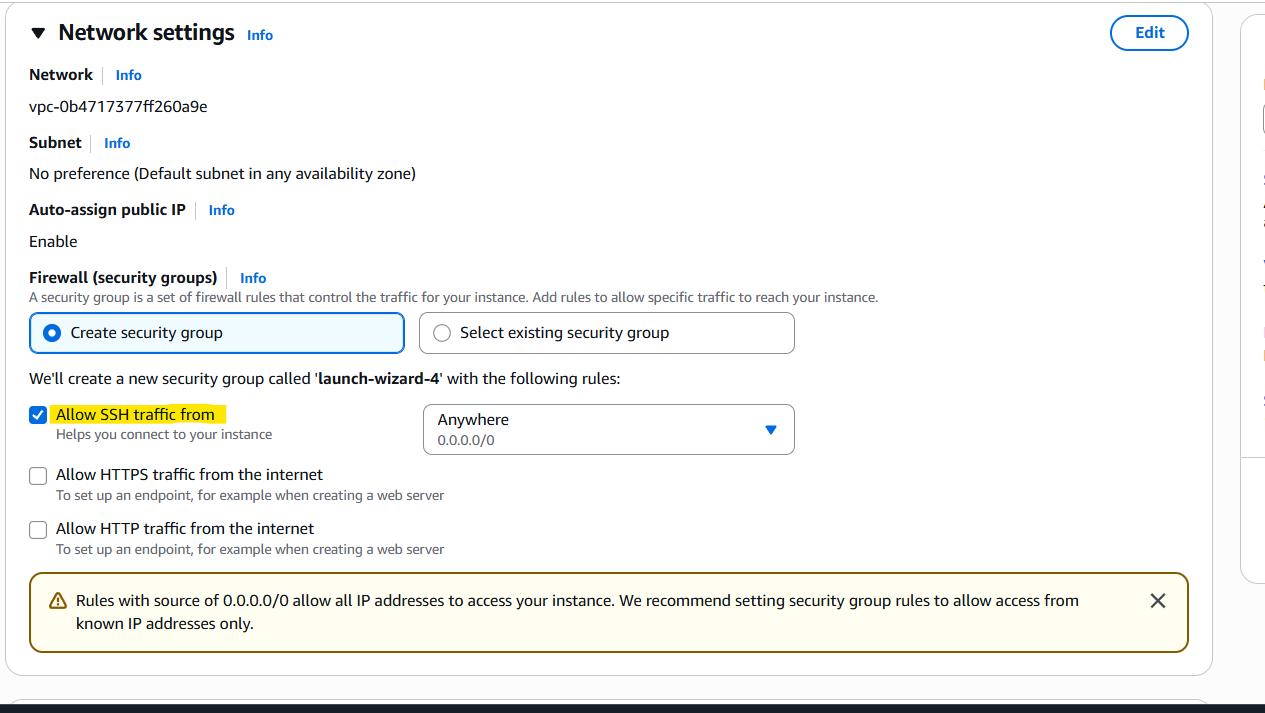
A screenshot of a computer

AI-generated content may be incorrect.

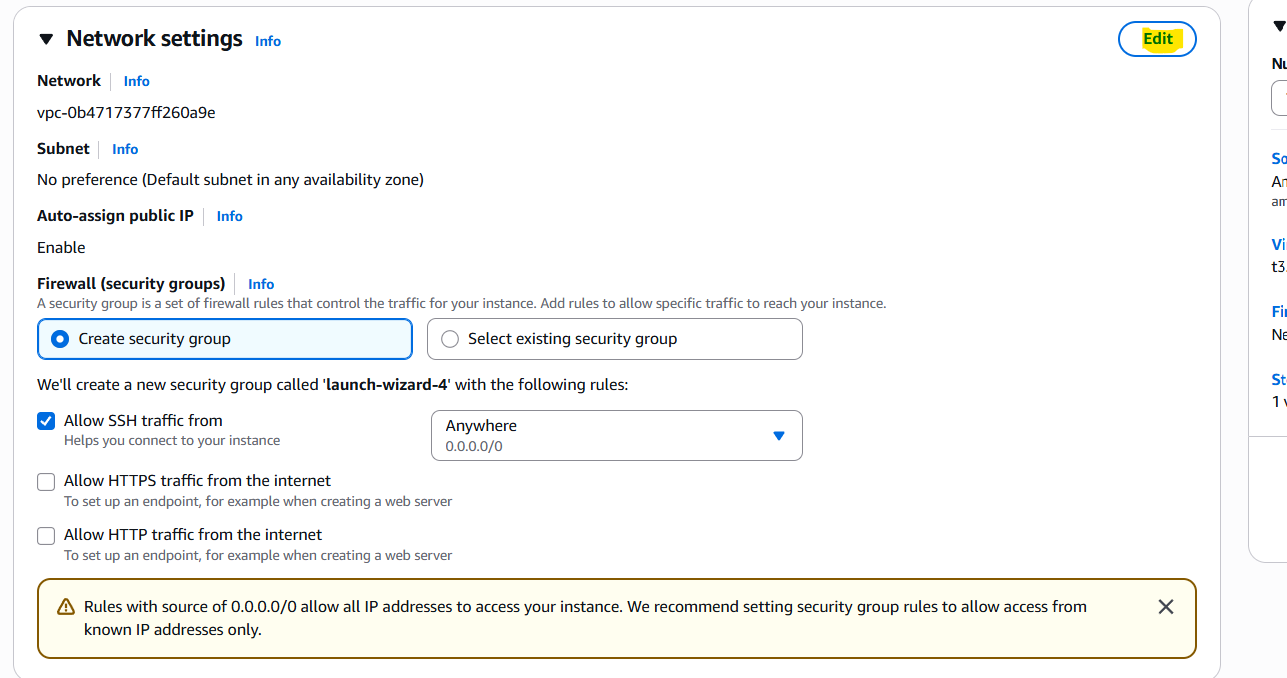
1. Security group

As we selected Linux OS by default AWS enable SSH port.

**Port number of SSH- 22.**



1. For creating rules who can access your EC2 machine you can edit security group. And add your configurations.



Ignore red box as of now

A screenshot of a computer

AI-generated content may be incorrect.

- You can create new rule or you can use existing one as well.

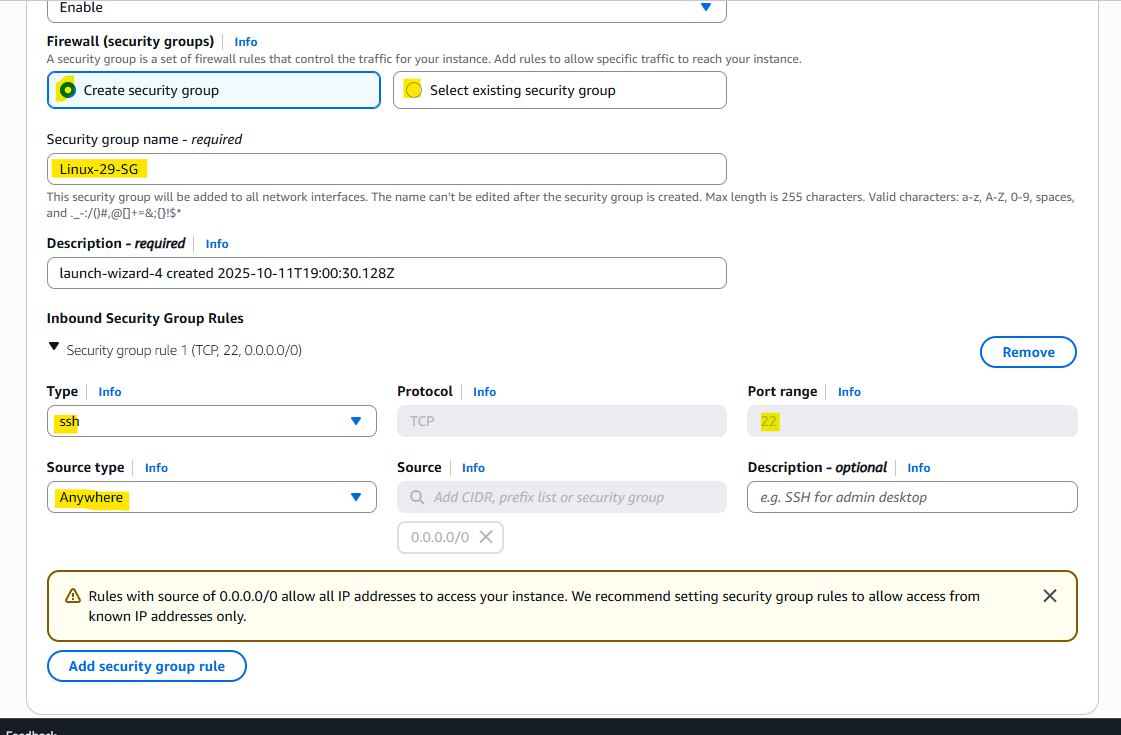
- Give name to your security group as per your choice. By default, AWS use to give name like “launch-wizard-4”.

- The name launch-wizard-4 means it was the **fourth time** you used the EC2 launch wizard to create an instance. AWS auto-increments this number each time you use the wizard without manually naming the security group.

- Default 1 security group rule is already there where AWS enabled SSH port and selected source type as anywhere which means anyone can access our EC2 machine.

- To avoid unwanted access, you can set rule using CIDR notation to control access of your EC2 machine.

- In one security group you can configure multiple rules.



1. Configure Storage

For Linux min 8gb is required.

A screenshot of a computer

AI-generated content may be incorrect.

1. Verify Summary

As we created new security group so its displaying there as New Security Group.

If we use existing one then it will display that name there.

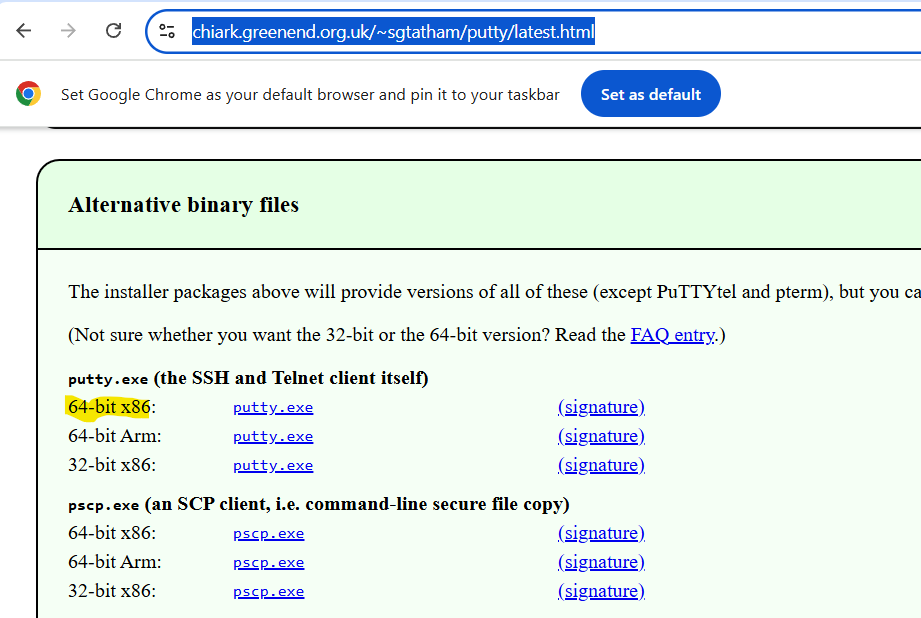
A screenshot of a computer

AI-generated content may be incorrect.

1. Click on Launch instance to create it.

**How to connect Linux EC2 machine**

* To connect windows EC2 machine we were having “Remote Desktop Connection”. Similarly
* To connect Linx EC2 machine we require “PUTYY” and “PUTTYGEN” application.
* PUTTY 🡪 To connect Linux EC2 machine
* PUTTYGEN 🡪 To Covert .pem to .ppk file.

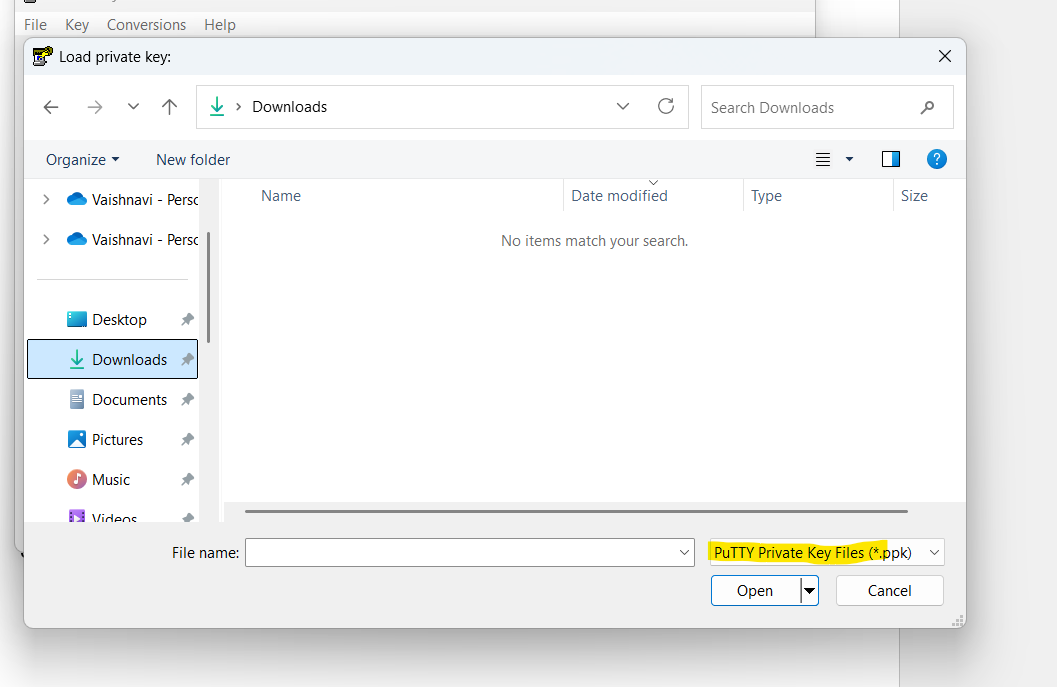
To download PUTTY and PUTTGEN use this link🡪 <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> A screenshot of a computer

AI-generated content may be incorrect.

**How to conver .pem into .ppk using PUTTYGEN.**

Steps:

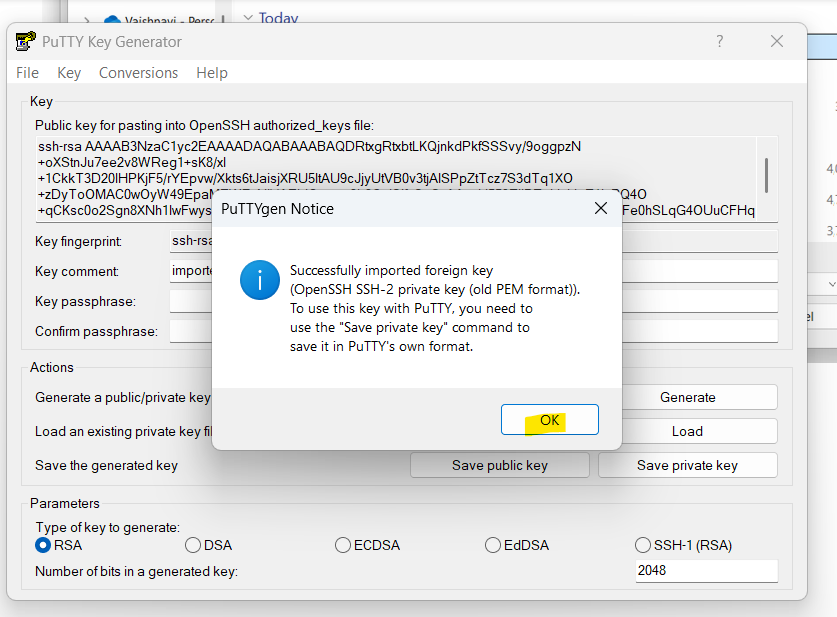
1. Double click on Puttygen.exe file to open it. A screenshot of a computer

   AI-generated content may be incorrect.
2. Click on Load
3. Select the folder where you stored .pem file. Here extension is selected as .ppk so its searching for .ppk files. So we are not able to see .pem files. Change this extension to All files.

-Select key open.

A screenshot of a computer

AI-generated content may be incorrect.

1. Click on Ok
2. Click on Save private keyA screenshot of a computer

   AI-generated content may be incorrect.
3. Give name for your PPK file ad save it. A screenshot of a computer

   AI-generated content may be incorrect.
4. This will get Downloaded. You can use this key now to connect to Linux EC2 machine.

**What is PPK? And Why we are converting .pem to .ppk?**

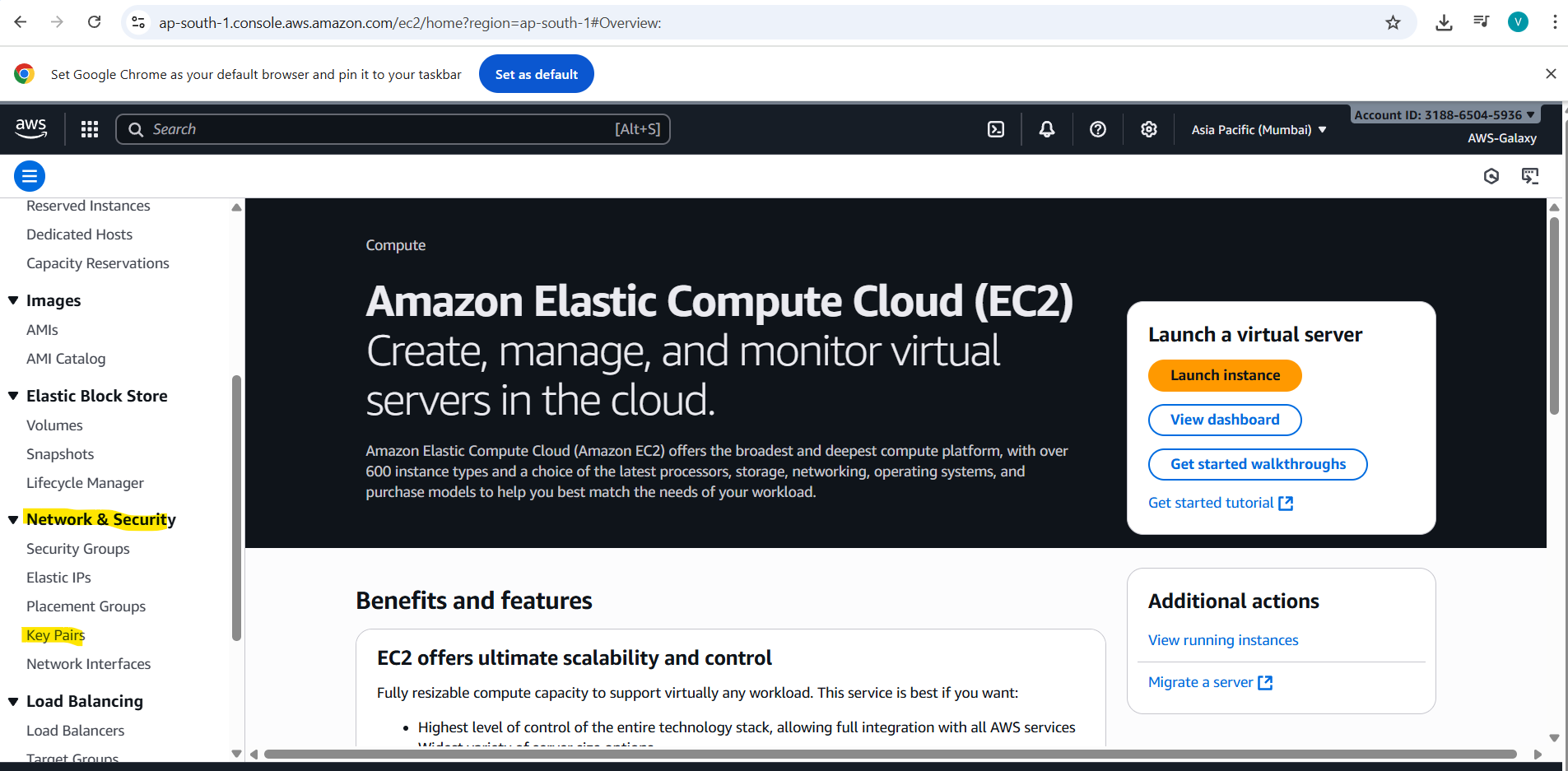
**PPK (PuTTY Private Key)** is the format used by **PuTTY**, a popular SSH client on **Windows**.

To connect Linux EC2 machine in windows we require Putty and Putty does not support .pem file. So we have conver .pem file into .ppk. This ensures compatibility with PuTTY.

**Note:** We can create ppk file using AWS as well but that is not used even in organization as well. Because only putty supports .ppk Other terminals like git, powershell, command prompt. Just for knowledge purpose see steps.

**Create PPK file using AWS**

Steps:

1. Go to Network & Security
2. Click on key pair. Here it will display all created key pairs. A screenshot of a computer

   AI-generated content may be incorrect.
3. Click on Create key pair.

Give name to key. Select file format as .ppk and click on Create key pair. It will generate .ppk file.

A screenshot of a computer

AI-generated content may be incorrect.

**How to connect to your Linux EC2 machine?**

There are different ways to connect to Linux EC2 machine.

1. **Using Putty-**

Steps:

1. Go to console select EC2 machine. A screenshot of a computer

   AI-generated content may be incorrect.
2. Go to SSH client. You will get Hostname there. Copy Hostname.A screenshot of a computer

   AI-generated content may be incorrect.
3. Open Putty. Enter hostname. A computer screen shot of a computer

   AI-generated content may be incorrect.
4. Add .ppk file

- Click on Connect

- Click on SSH

- Credentials

- Auth

A screenshot of a computer

AI-generated content may be incorrect.

1. Browse her PPK file. Click on Open

A screenshot of a computer

AI-generated content may be incorrect.

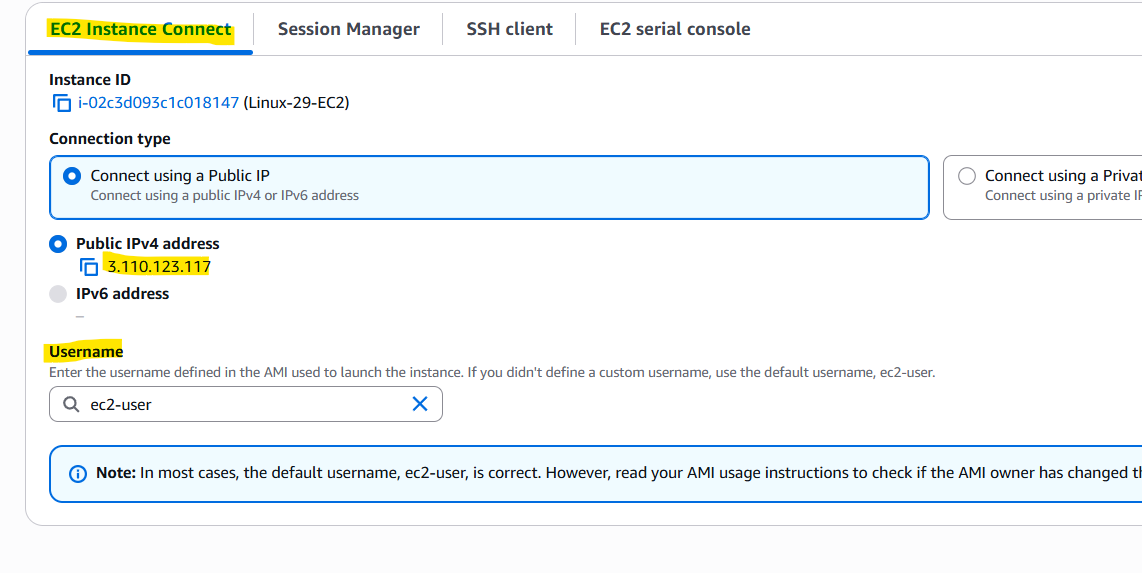
1. You will get connected to your EC2 machineA computer screen shot of a black screen

   AI-generated content may be incorrect.

Note: Here we have used hostname. You can also use public ip address to connect ec2 machine.

When we use hostname, it will not ask for username it will take it from hostname itself. As hostname is starting with username. But when you try to connect your ec2 machine using ip address it will ask username rest process is same only.

Ip address you will get it from description of machine in main console or from EC2 instance connect console as well. There you will get username also.



1. Copy the Ip address
2. Open putty paste it there
3. Browse ppk file and click on open
4. In console it will ask username. Click enter. A screenshot of a computer

   AI-generated content may be incorrect.
5. **Using GitBash🡪** [**Link**](youtube.com/watch?v=rIwiz_Zjz3E&feature=youtu.be)
6. Open GitBash in folder where your .pem file is available.
7. Copy SSH command from SSh client and paste in git bashA screen shot of a computer

   AI-generated content may be incorrect.

Remember to Connect ec2 maching using Git, Command prompt, Powershell we required .pem file.

Run same command to connect in different terminal. Refer below videos

* Using **Command prompt**🡪 [Link](youtube.com/watch?v=quxIQ19ycPw&feature=youtu.be)
* Connect Using **PowerShell**🡪 [Link](youtube.com/watch?v=rIwiz_Zjz3E&feature=youtu.be)

1. **Using Browser🡪** [**Link**](https://www.youtube.com/watch?v=_3wzq96HSqw&feature=youtu.be)

**Questions:**

**What is Linux?**

|  |
| --- |
| In 1991, Linus Torvalds created the Linux kernel, inspired by [UNIX operating system](https://www.geeksforgeeks.org/linux-unix/introduction-to-unix-system/) and the MINIX operating system, to make a flexible and affordable system for personal computers   * **Operating System**: Like Windows or macOS, Linux is software that manages your computer's hardware and lets you interact with it. * **Based on UNIX**: It’s a Unix-like system, meaning it inherits the stability, security, and multitasking capabilities of UNIX. * **Open Source**: Anyone can view, modify, and distribute its source code. This has led to a massive global community that constantly improves it   How to create Linux ?  **1. Local Installation on Physical Hardware**   * **Use a bootable USB drive** with a Linux ISO (e.g., Ubuntu, Fedora). * Boot your PC from the USB and install Linux directly. * You can dual-boot with Windows or replace it entirely.   ✅ Best for: Full control, learning Linux deeply, using it as your daily OS.  **2. Virtual Machine (VM) on Your Computer**   * Use tools like **VirtualBox**, **VMware**, or **Hyper-V**. * Download a Linux ISO and create a VM with allocated RAM, CPU, and disk. * Install Linux inside the VM.   ✅ Best for: Safe experimentation, running multiple OSes, testing scripts.  **3. Cloud-Based Linux Instances**   * **AWS EC2**: Launch a Linux instance (e.g., Ubuntu, Amazon Linux) and connect via SSH. * **Azure Virtual Machines**: Choose a Linux image and deploy. * **Google Cloud Compute Engine**: Same idea—select a Linux distro and spin up a VM.   ✅ Best for: Real-world server management, DevOps, scalable deployments.  **4. Containers (Lightweight Linux Environments)**   * Use **Docker** to run Linux containers.   bash  docker run -it ubuntu /bin/bash   * Containers are isolated environments that share the host OS kernel.   ✅ Best for: Microservices, testing apps, CI/CD pipelines.  **5. WSL (Windows Subsystem for Linux)**   * On Windows 10/11, enable WSL and install a Linux distro from the Microsoft Store. * You get a full Linux CLI inside Windows.   ✅ Best for: Developers who want Linux tools without leaving Windows.  For more info use this [Link](https://www.geeksforgeeks.org/linux-unix/introduction-to-linux-operating-system/) |

**Why are preferring to create Linux machine instead?**

|  |
| --- |
| **1. Open Source & Cost-Effective**   * Linux is **free and open source**, which means no licensing fees. * Organizations can customize it deeply without vendor lock-in.   **2. Dominance in Cloud Infrastructure**   * Over **90% of public cloud workloads** run on Linux. * Major cloud providers like AWS, Azure, and Google Cloud offer better support and tooling for Linux-based instances.   **3. DevOps & Automation Friendly**   * Tools like **Docker, Kubernetes, Jenkins, Ansible**, and **Terraform** are built with Linux in mind. * Linux’s command-line interface (CLI) is powerful and scriptable, making automation seamless.   **4. Security & Stability**   * Linux is known for its **robust security model** and **stability**, making it ideal for servers and production environments. * Frequent updates and a strong global community help patch vulnerabilities quickly.   **5. Lightweight & Efficient**   * Linux can run on minimal hardware, which is perfect for **microservices**, **containers**, and **virtual machines**. * It consumes fewer resources compared to Windows, which is crucial in cloud environments where you pay for compute usage.   **6. Community & Ecosystem**   * A massive global community means better support, documentation, and faster innovation. * Thousands of distributions (like Ubuntu, CentOS, Debian) cater to different needs. |

**What SSH?**

|  |
| --- |
| SSH stands for **Secure Shell**, and it’s like a secret tunnel between two computers that lets you communicate safely over an insecure network—like the internet.  **What Does SSH Do?**   * **Remote Access**: It allows you to log into another computer (usually a server) and control it as if you were sitting right in front of it. * **Encrypted Communication**: Everything sent through SSH is encrypted, so hackers can’t snoop on your commands or data. * **File Transfers**: You can securely upload or download files using tools like scp or sftp.   **How It Works**  SSH uses **key pairs** or **passwords** for authentication:   * **Public Key**: Shared with the server. * **Private Key**: Stays with you and must be kept secure. |

**Rec 4-**

**Linux Commands**

**File or directory creation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Task** | **Command** | **Description** | | --- | --- | --- | | Create file with content | cat filename | Crete file with given name and you add the content and enter. Ctr+d for save and exit. It will create file with .file extension by default. To create file with desired extension provide etension at the end. | | Create empty file | touch filename | Creates a blank file | | Create directory | mkdir dirname | Makes a new folder | | Create nested directories | mkdir -p dir1/dir2 | Creates parent and child folders in one go | | Create hidden file | Cat >.filename | You need to give dot (.) before filename to create hidden file. | | Create hidden directory | mkdir .foldername | Add dot (.) before foldername | | Create copy of file | cp <source file> <destination file> | Source file🡪 file which we are copying  Dest means final copy of file. | |

**Viewing File**

| Command | Description |
| --- | --- |
| cat filename | Displays entire file content |
| less filename | Opens file in scrollable view (quits with q) |
| more filename | Similar to less, but older |
| head filename | Shows first 10 lines |
| tail filename | Shows last 10 lines |
| tail -f filename | Live view of file updates (great for logs) |
| nl filename | Shows file with line numbers |
| wc -l filename | Counts number of lines in file |

**Removing files and directory**

| Command | Description |
| --- | --- |
| rm filename | Deletes a file. |
| rm -i filename | Prompts before deleting |
| rm -r dirname | Deletes a directory and its contents recursively. Delete everything. |
| rm -rf dirname | Force deletes directory and contents without prompt. Can delete entire system. |
| rm -ri dirname | Delete folder and confirm each file |
| rmdir dirname | Deletes empty directory only |

**Listing files**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Command | Description | | --- | --- | | ls | List all files | | ls -l | Long format: shows permissions, owner, size, date. Also you can find which one is file or dir.  “-“ file  “d” folder | | ls -lh | Human-readable sizes (e.g., KB, MB) | | ls -lt | Sort by modification time (newest first) | | ls -ltr | Sort by time, reverse order (oldest first) | | Ls -a | Display hidden files and folder. | | ls -la / ls -al | Display hidden files and folder. Also you can find which one is file or dir. | |