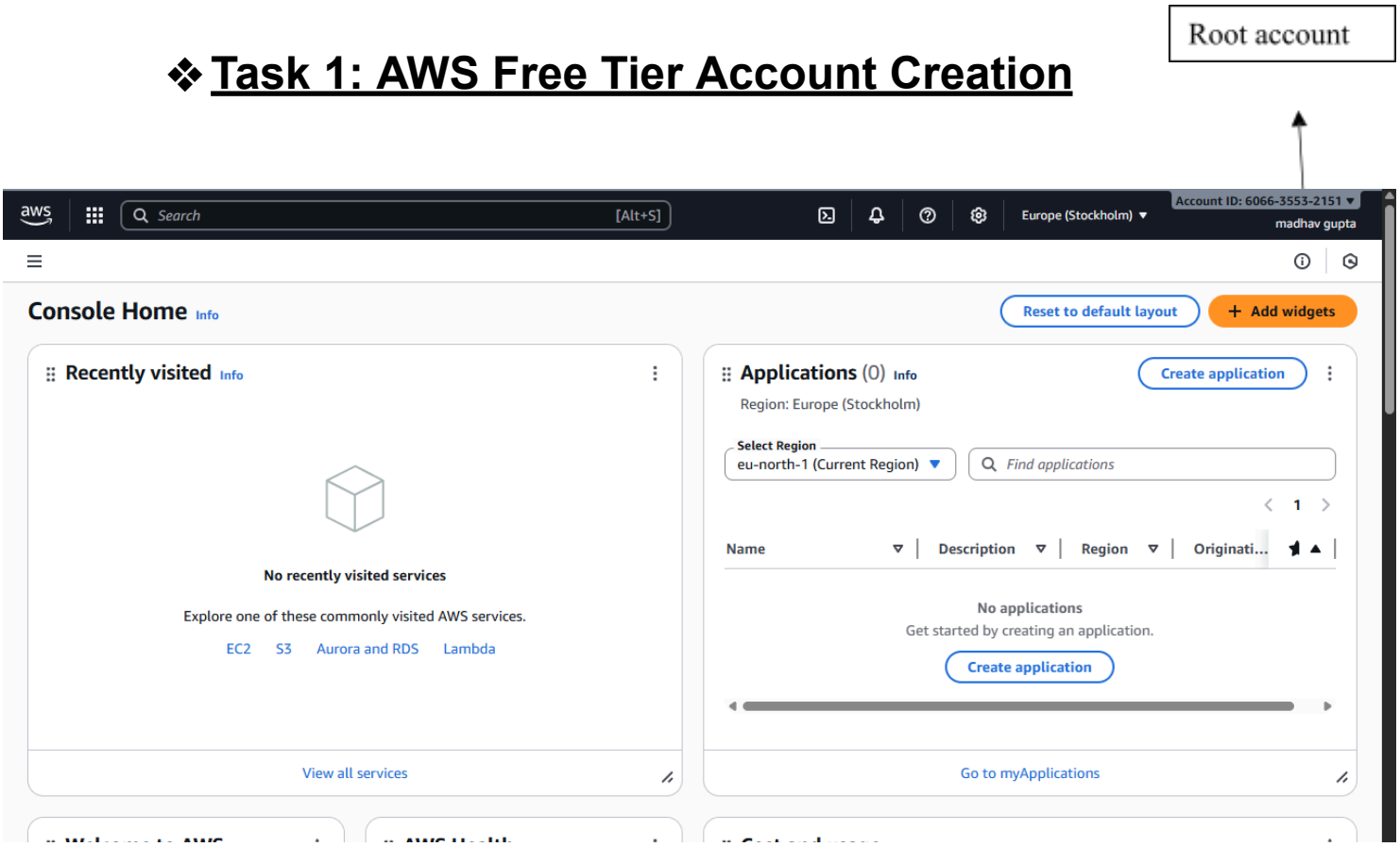
AWS Assignment-1

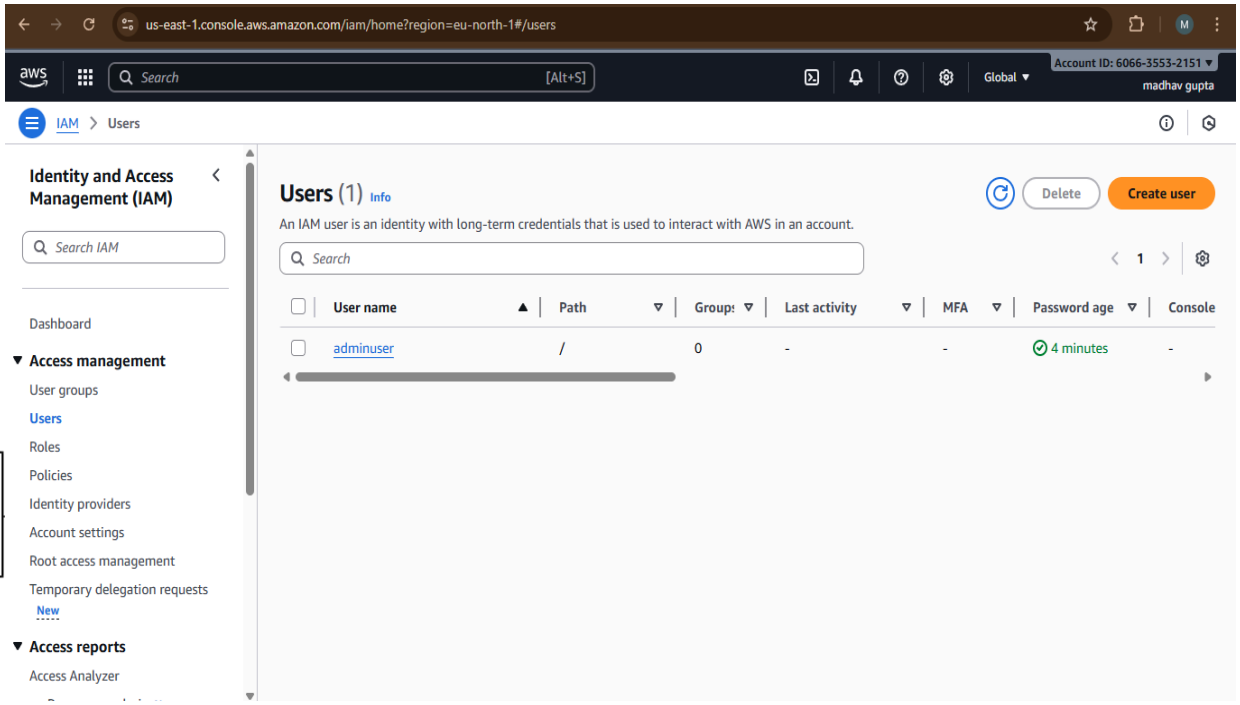
❖ Task 1: AWS Free Tier Account Creation



Steps:

1. Go to the **AWS Management Console →** https://aws.amazon.com/console.
2. Click **Sign in as root user**.
3. Enter the **email address** (used when creating the account).
4. Enter the **root password**.
5. (If MFA is enabled) Enter the **MFA code**.
6. After entering the **MFA code,** root user will be created

❖ Task 2: Root-Equivalent IAM User Creation

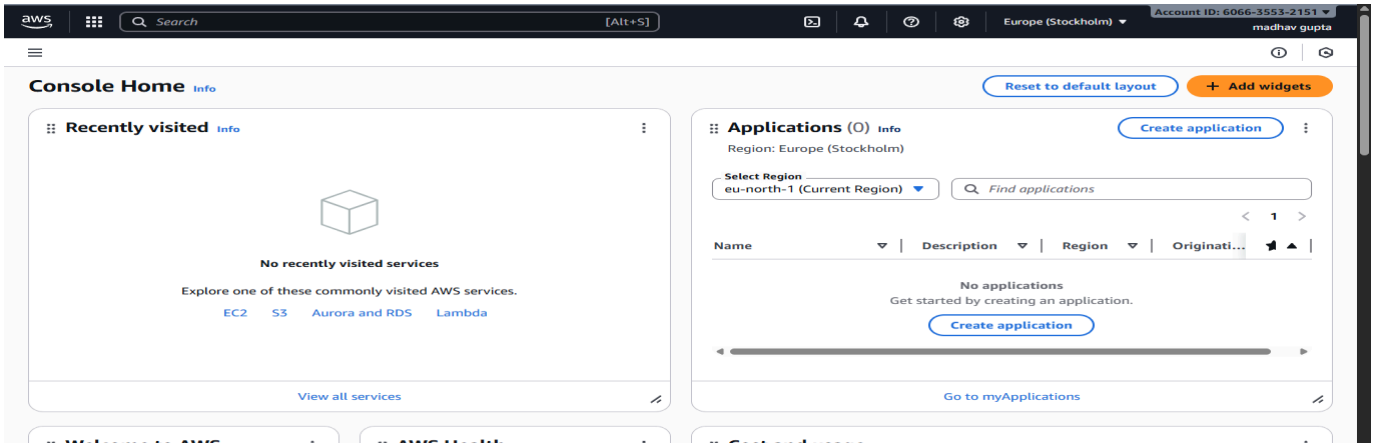


IAM user created as:--**adminuser**

Steps:

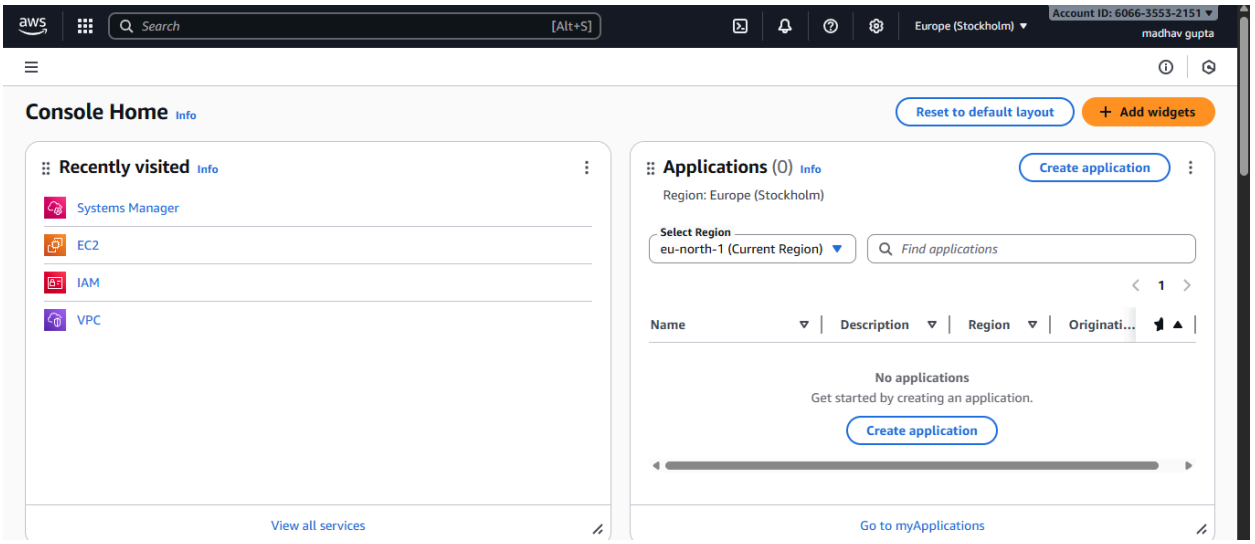
Step 1: Sign in as the Root User

- Go to the AWS Management Console.
- Click **"Sign in as root user"**.
- Enter the **root email** and **password**.
- (If MFA is enabled, enter the MFA code.)



Step 2: Open the IAM Console

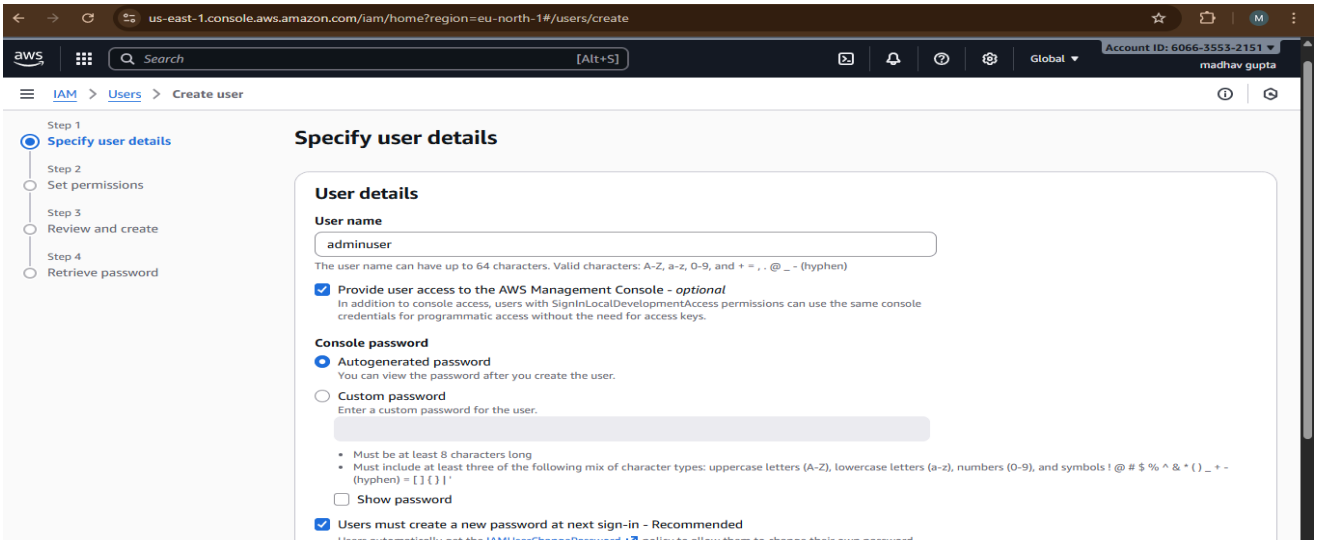
- In the AWS console search bar, type **"IAM"**.
- Click **IAM (Identity and Access Management)** to open it.



Step 3: Create a New IAM User

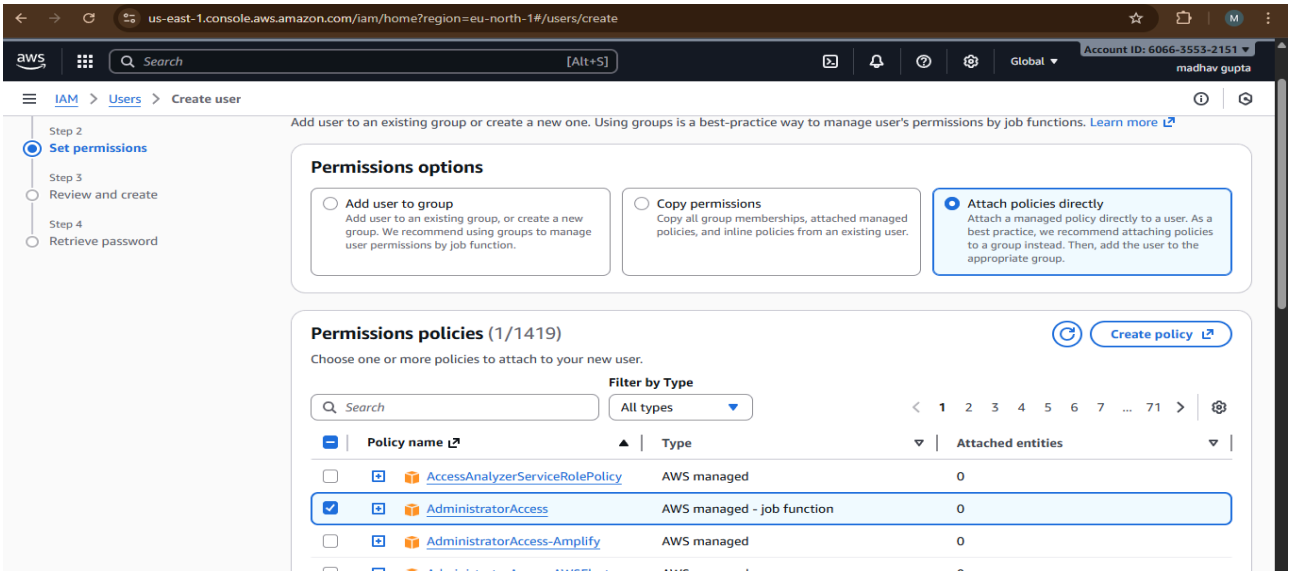
1. In the left sidebar, click **Users → Add users**.
2. Enter a **username** (e.g., admin-user).
3. Under **Select AWS access type**, choose:
    - ⬛ **Password - AWS Management Console access** (if user needs console access)
    - ⬛ **Access key - Programmatic access** (if user needs CLI or API access)

Click **Next**.



Step 4: Attach Administrator Permissions

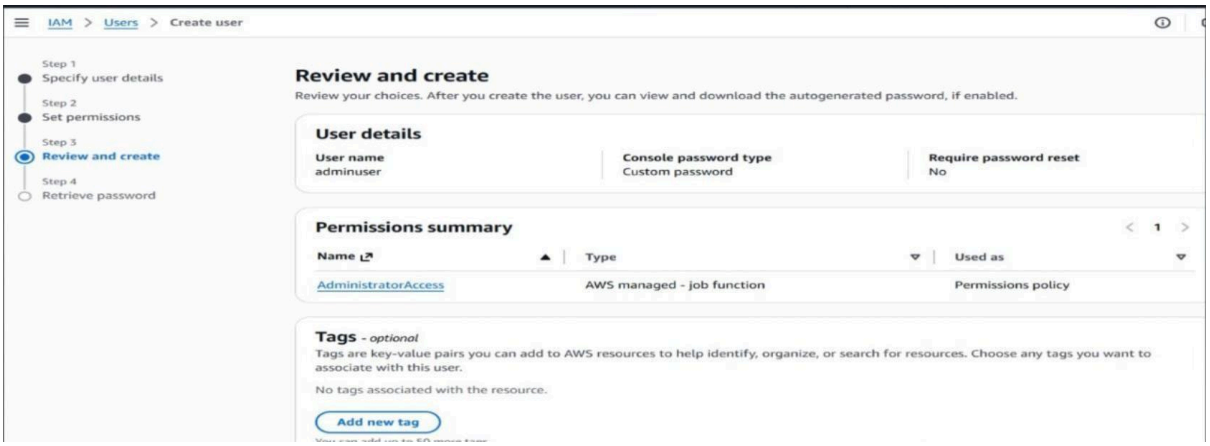
1. Choose **Attach policies directly**.
2. Search for and select **AdministratorAccess**.
    - This gives the user full permissions for all AWS services (like the root user, but still restricted by IAM).
3. Click **Next**.



Step 5: Review and Create

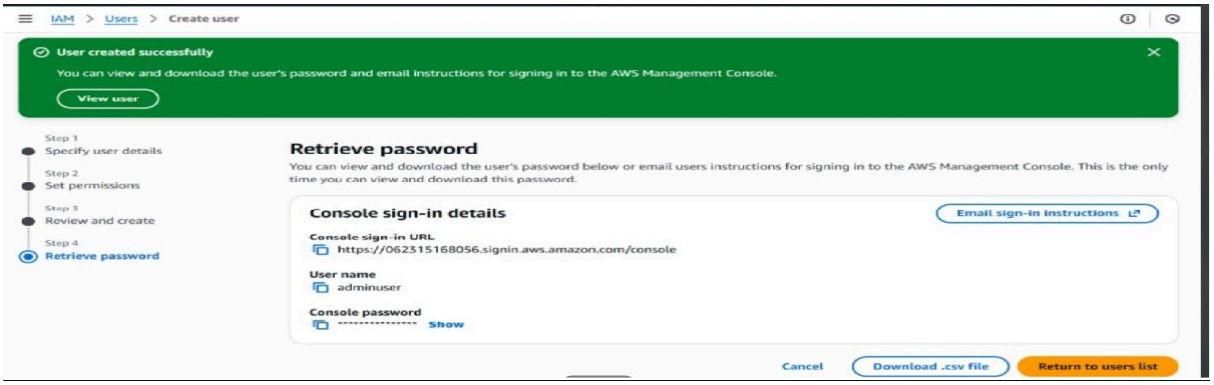
- Review all details.
- Click **Create user**.

AWS will show the **login URL**, **username**, and (if applicable) the **password** or **access keys**.

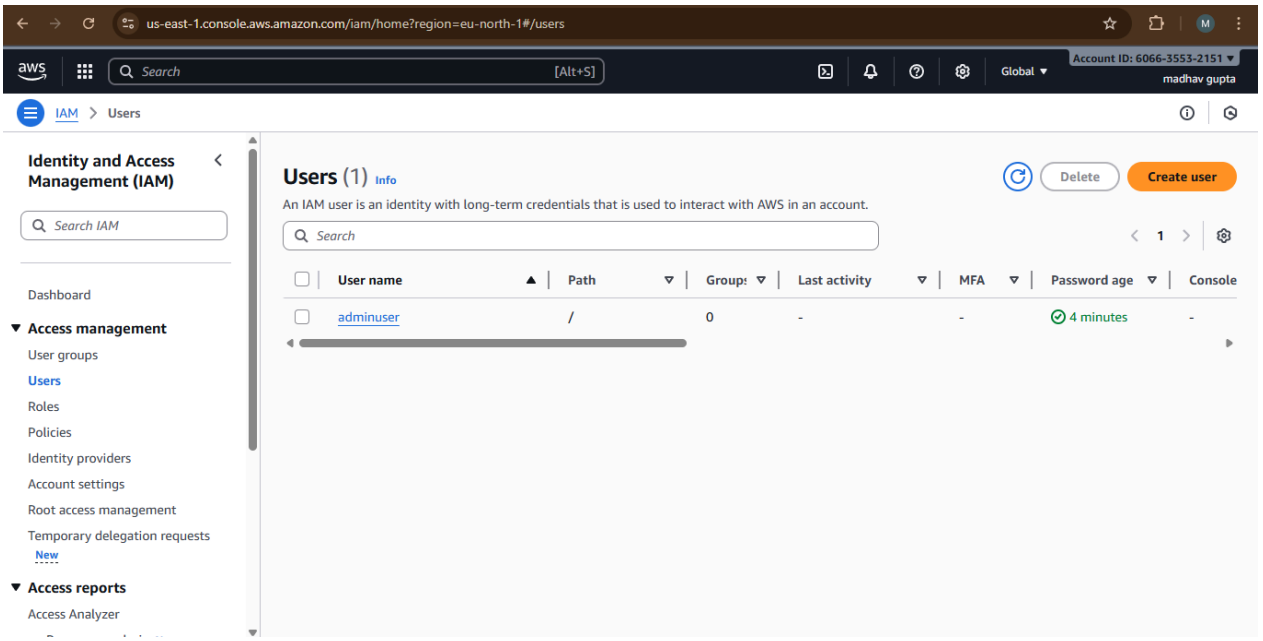


Step 6: Secure the Account

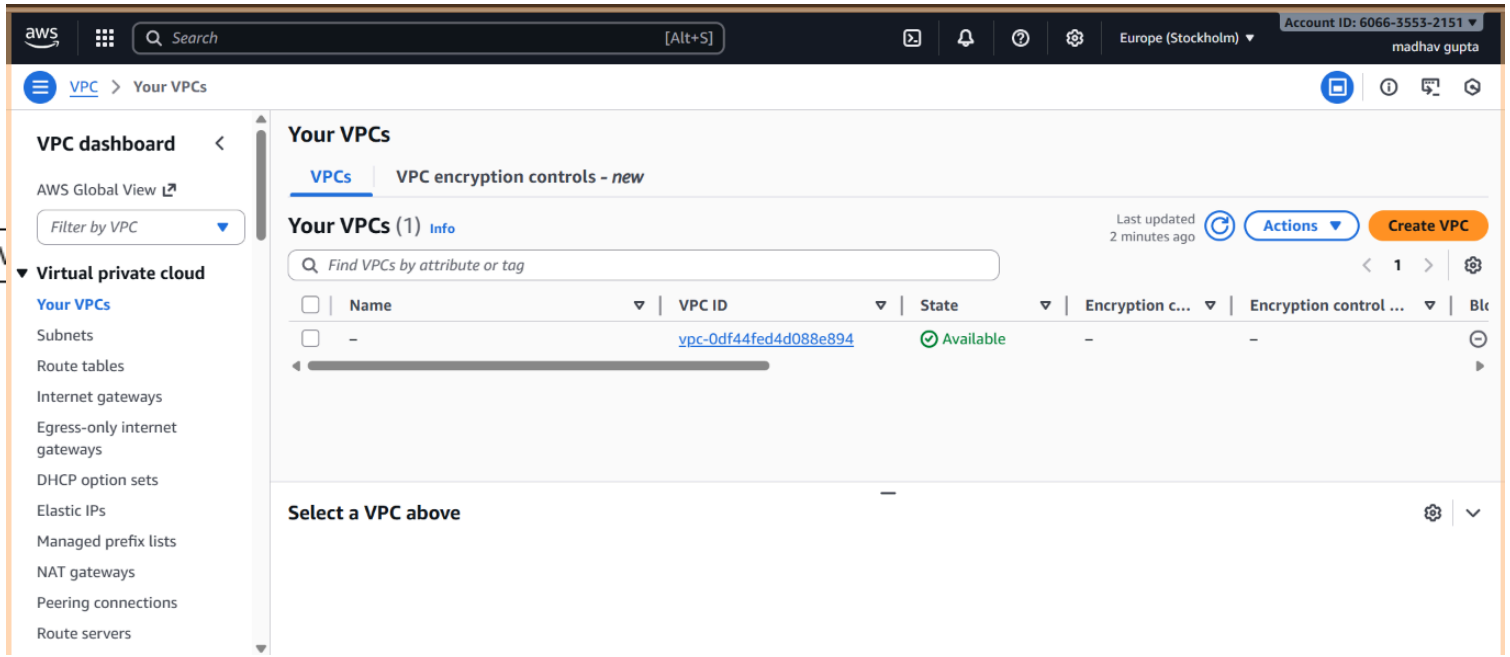
1. **Sign in as the new IAM user** to confirm it works.
2. **Enable MFA** (Multi-Factor Authentication):
    - Go to **IAM → Users → [your user] → Security credentials**.
    - Under **Multi-factor authentication (MFA)**, choose **Assign MFA device**.
    - Use an **authenticator app** (like Google Authenticator).
3. Store credentials securely (not in plaintext).



- Hence, IAM user created as **adminuser**



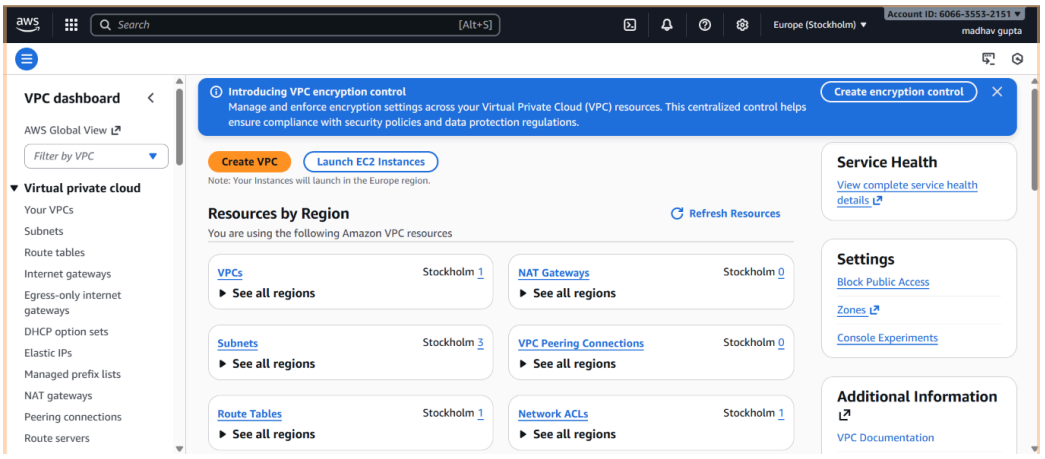
❖ Task 3: VPC Selection & Region Choice



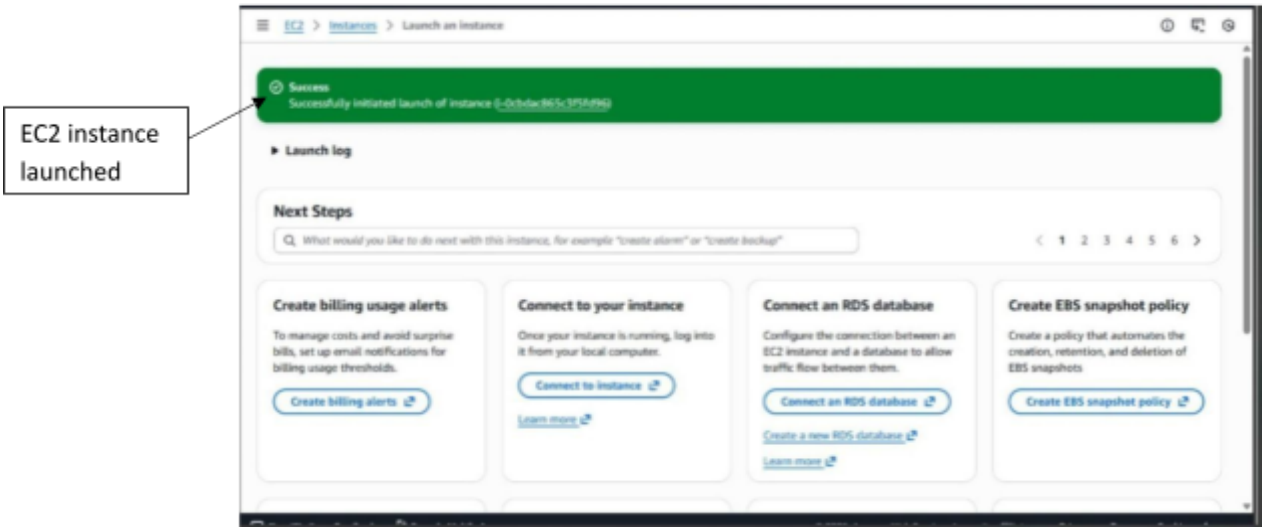
Steps:

Steps to Select or Create a VPC

1. In the AWS console, go to the **VPC Dashboard** (search for "VPC").
2. Click **Your VPCs** in the left sidebar.
3. You'll see a **Default VPC** (it usually has "Default" in the "Name" column).
4. You can use this VPC directly for EC2, RDS, or other resources — it already has:
    - Public subnets in each AZ
    - Internet Gateway attached
    - Route tables set up



❖ Task 4: Windows Web Server EC2 Instance



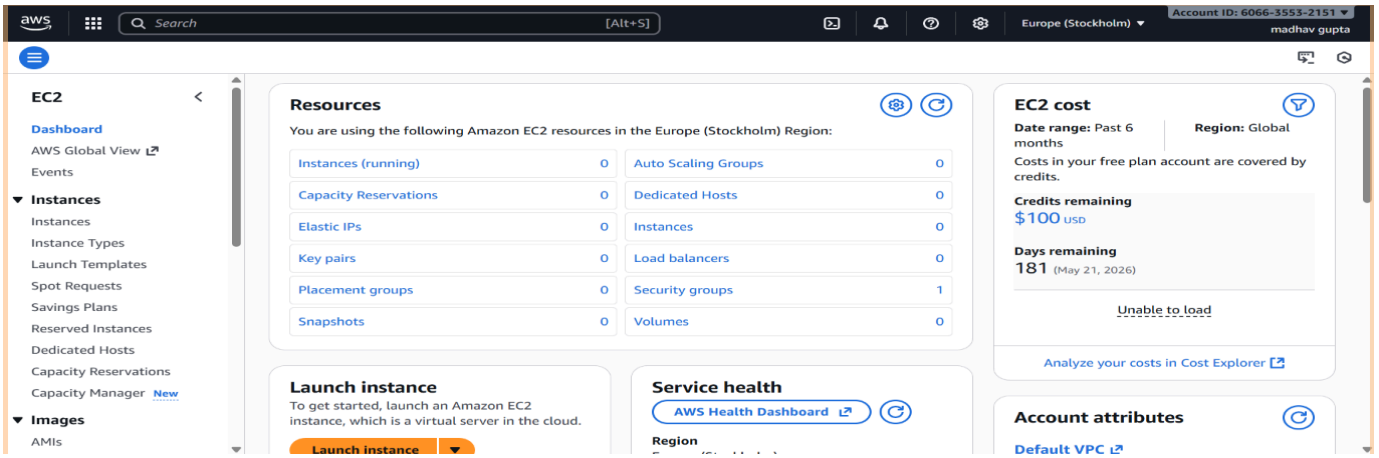
Steps:

Step 1: Sign in and Select a Region

1. Go to the AWS Management Console.

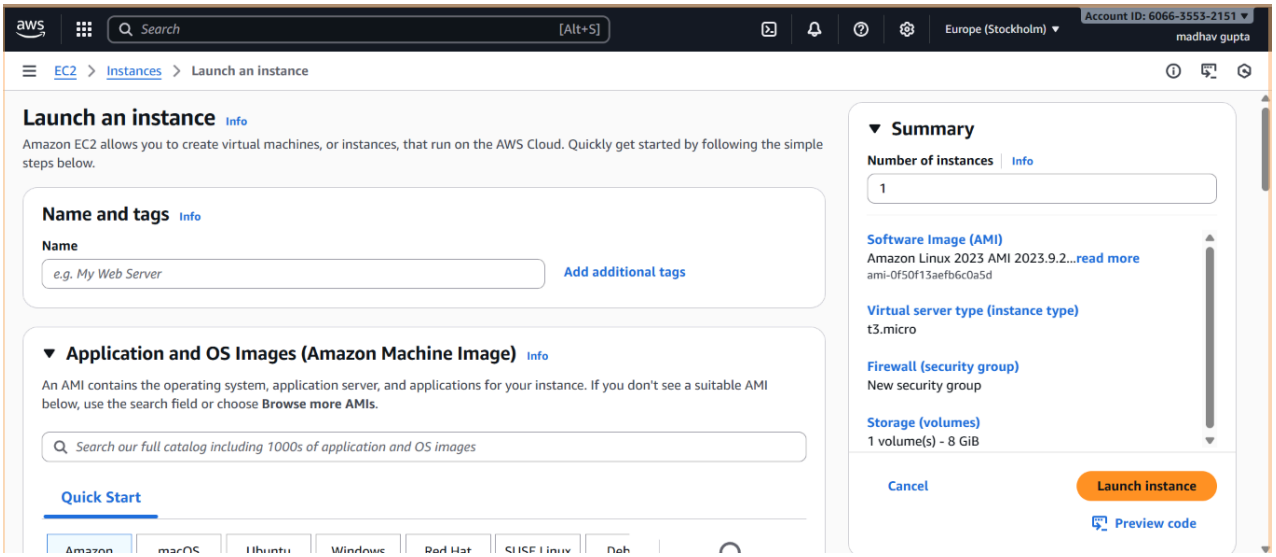
In the top-right corner, choose your **AWS Region** (e.g., Asia Pacific (Mumbai) ap-south-1).

- All resources you create will reside in this region.



Step 2: Open the EC2 Dashboard

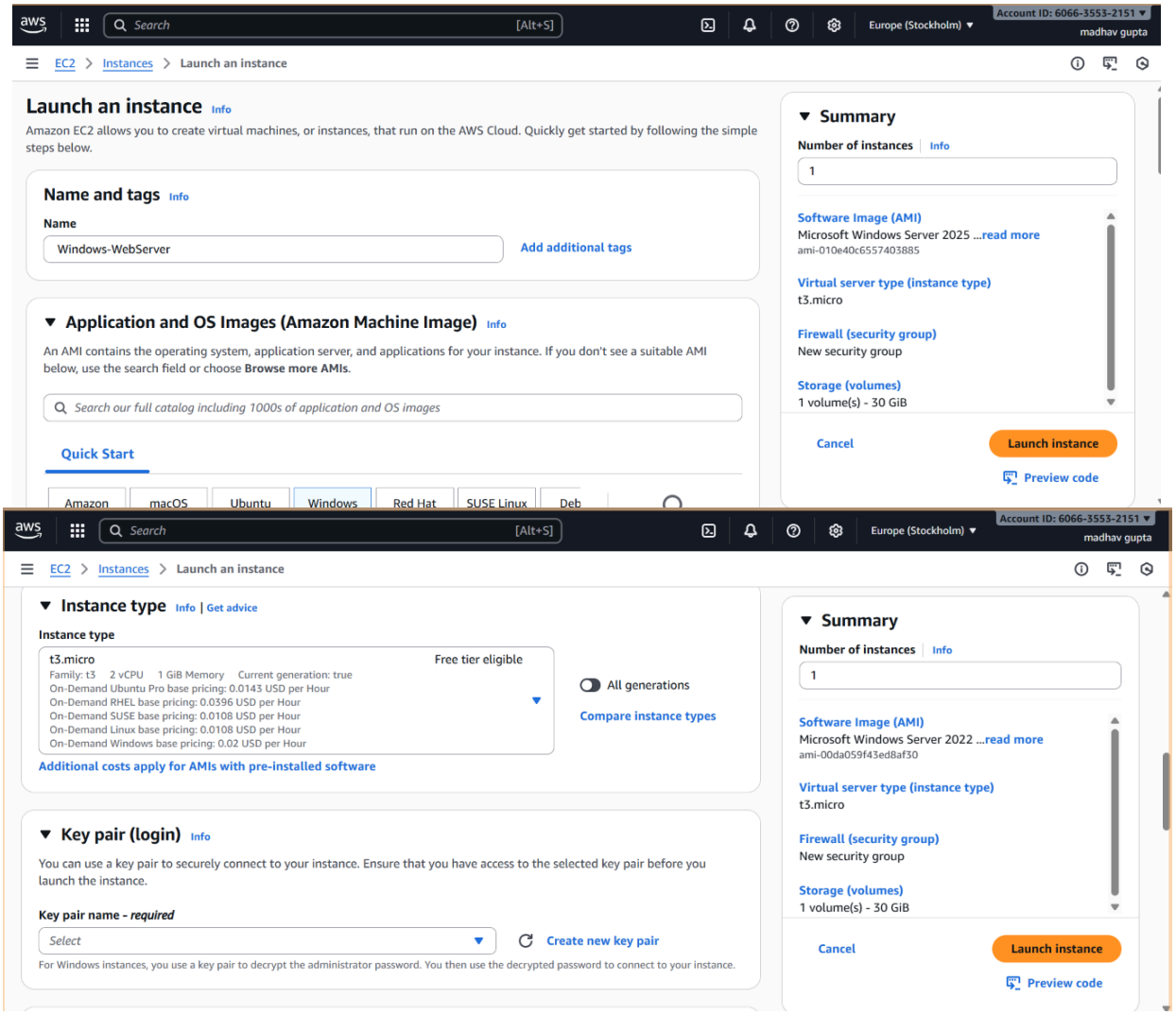
1. In the AWS Console search bar, type **"EC2"** and select it.
2. Click **Instances → Launch instances**.



Step 3: Configure Basic Details

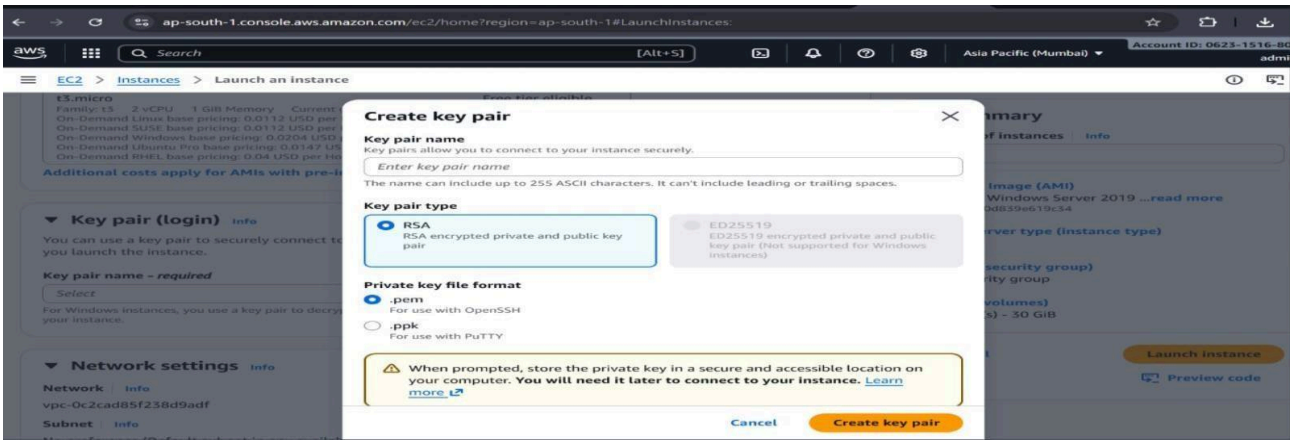
1. **Name your instance** — e.g., Windows-WebServer.
2. **Choose an Amazon Machine Image (AMI):**
   - Scroll to find **Microsoft Windows Server 2022 Base** (or 2019/2016 if needed).
   - These AMIs come with Windows preinstalled.
3. **Choose Instance Type:**

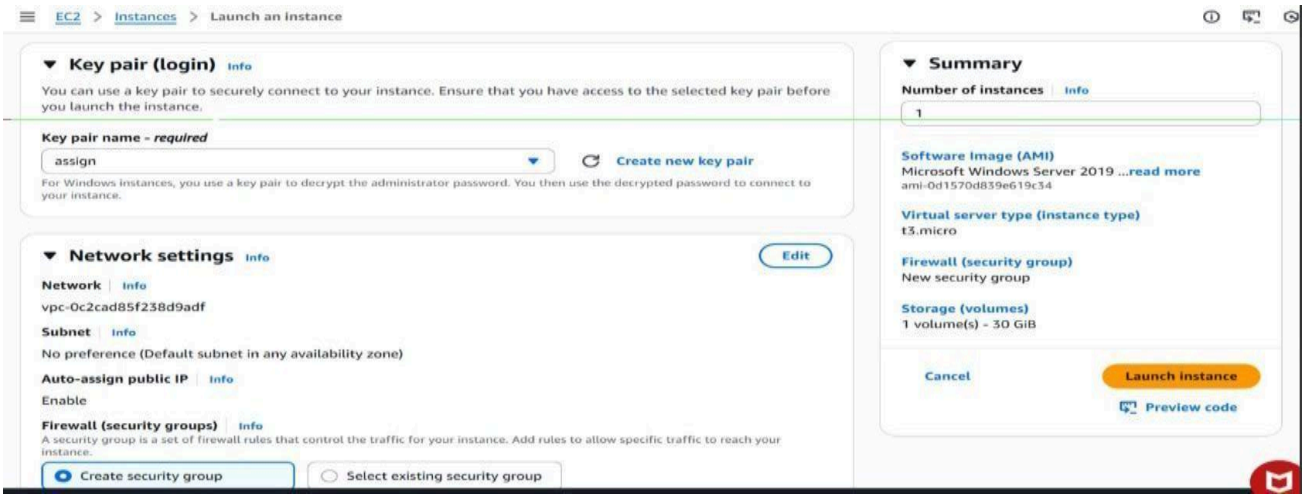
- For basic setups, select t3.micro or t2.micro (Free Tier eligible).



Step 4: Configure Key Pair (Login)

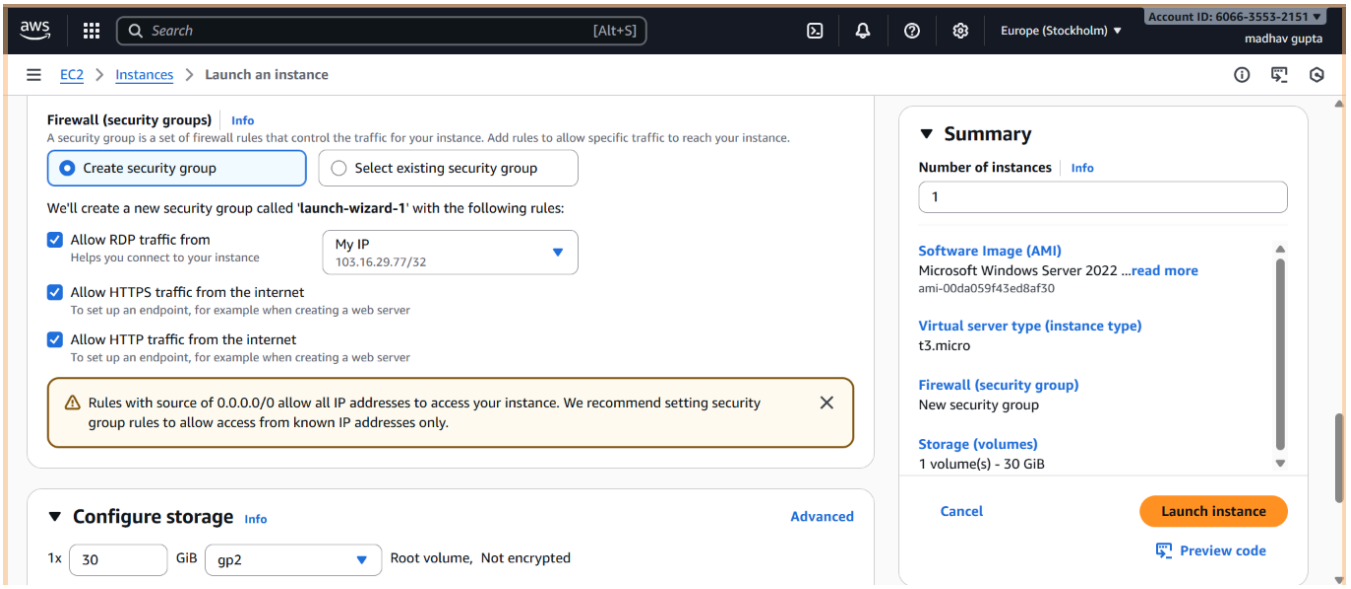
1. Under **Key pair (login)**, choose:
   - **Create new key pair** (if you don't have one yet).
   - Give it a name (e.g., WindowsKeyPair).
   - Choose **Key pair type: RSA**, and **File format: .pem** (for RDP decryption).
   - Click **Create key pair** — it will automatically download the .pem file to your computer.
     . ı Keep it safe — you'll need it to connect later.





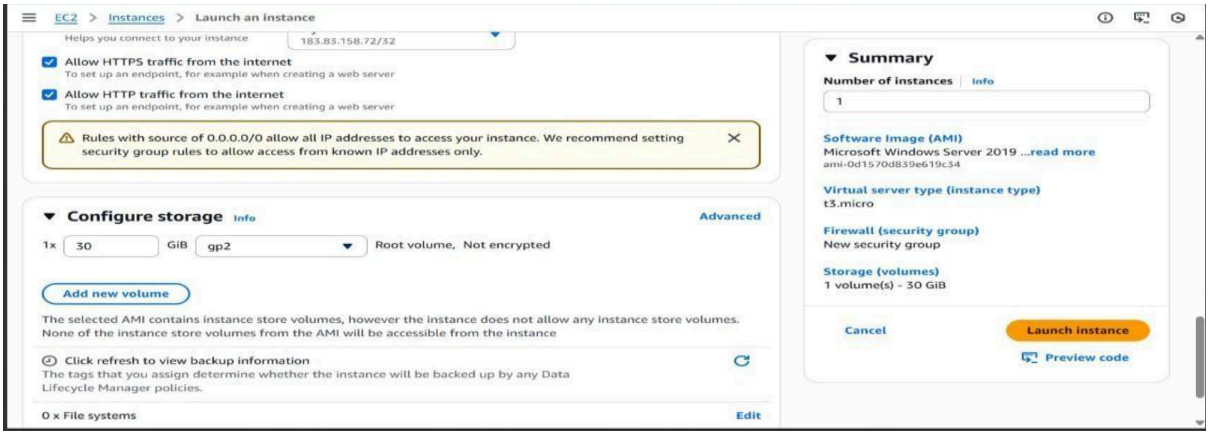
Step 5: Configure Network Settings

1. Under **Network settings**, select your **VPC** and **Subnet** (default VPC is fine).
2. Check **Auto-assign public IP** → *Enable*.
3. Under **Firewall (security group)**:
    - Choose **Create security group**.
    - Allow:
        - ⬛ **RDP (Port 3389)** → Source: My IP (for remote access)
        - ⬛ **HTTP (Port 80)** → Source: Anywhere (for web traffic)



Step 6: Configure Storage

- Default 30 GB is fine, or increase if needed.
- Ensure volume type is **gp3** (default).



Step 7: Launch Instance

- Review your settings and click **Launch instance**.
- Wait a few moments until the status shows **running**.

Task 5: Systems Manager Fleet Manager Setup

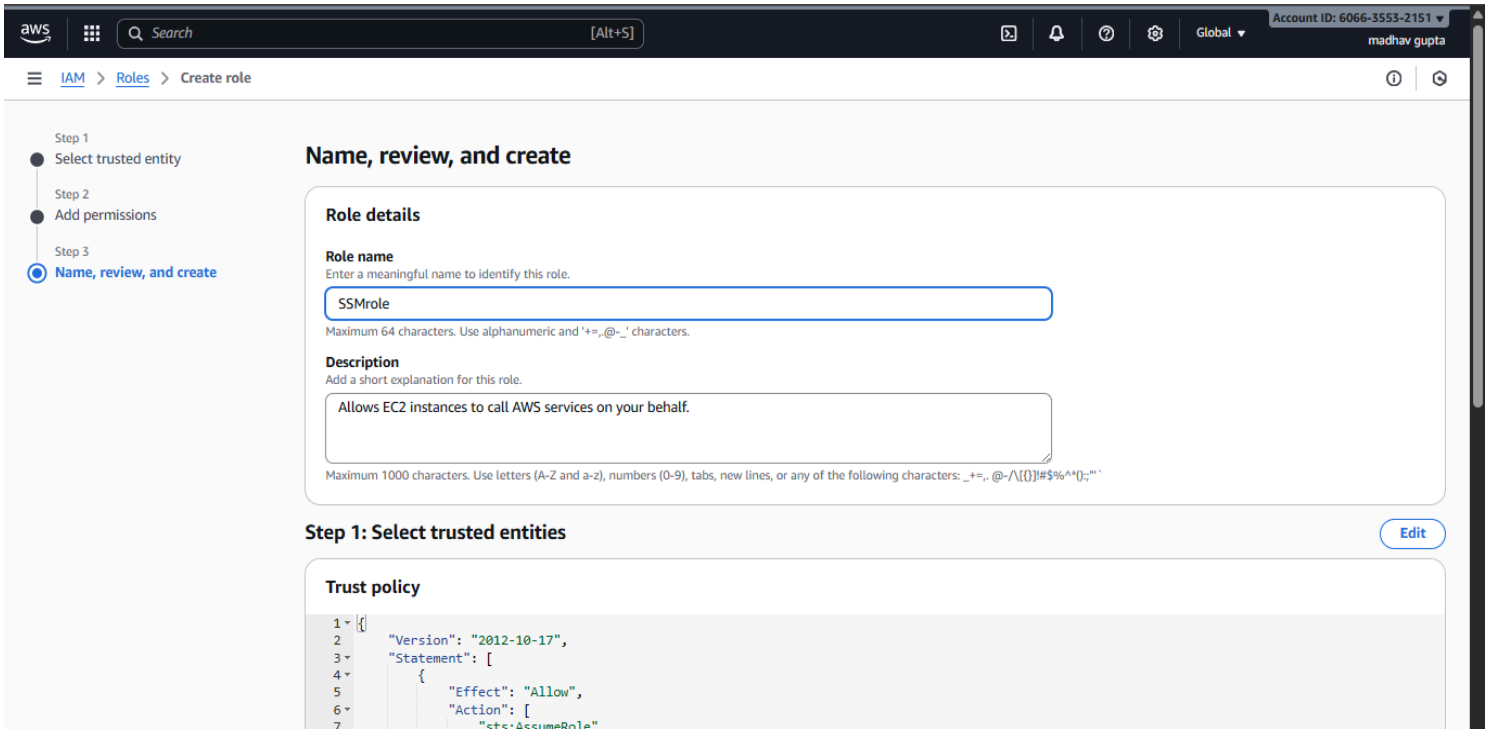
Steps:

Step 1: Create IAM Role for Systems Manager

Fleet Manager requires the instance to have permission to communicate with SSM.

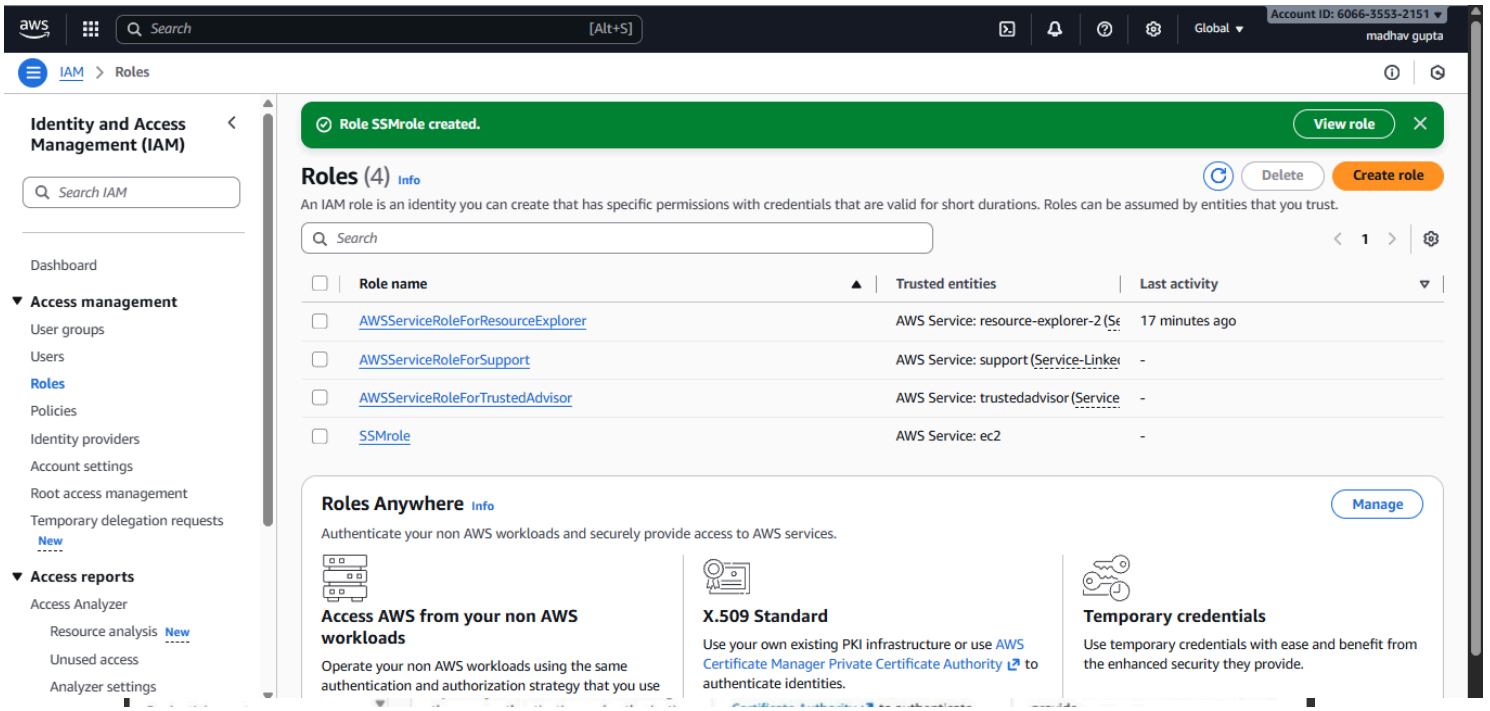
Steps:

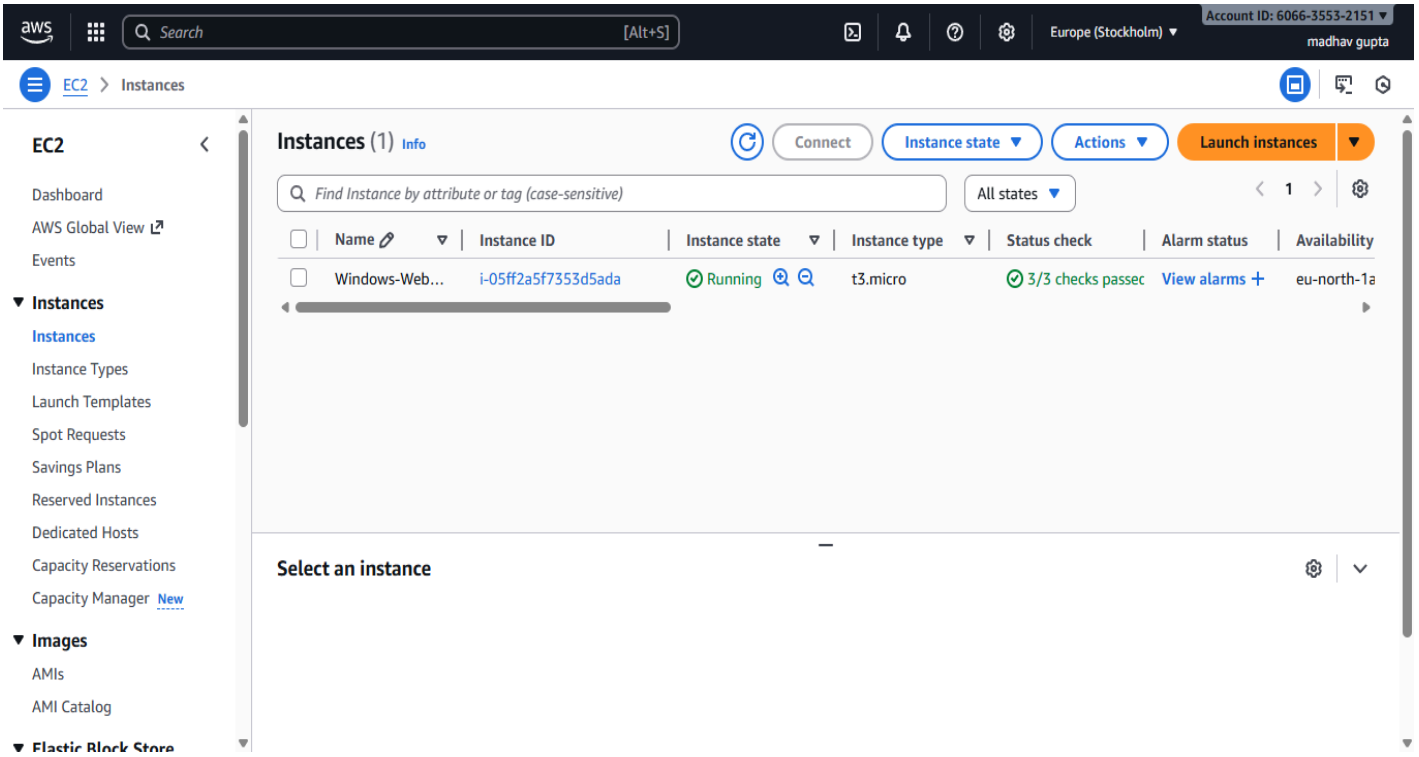
1. Go to **IAM → Roles → Create role**.
2. Under **Trusted entity type**, choose **AWS service**.
3. Choose **EC2** → Click **Next**.
4. In **Permissions policies**, search and select:
    - AmazonSSMManagedInstanceCore
5. Click **Next**, name the role:
6. SSM-Managed-Instance-Role
7. Click **Create role**.



Step 2: Attach IAM Role to EC2 Instance

1. Go to **EC2 → Instances**.
2. Select your instance → **Actions → Security → Modify IAM role**.
3. Select SSM-Managed-Instance-Role → click **Update IAM role**.





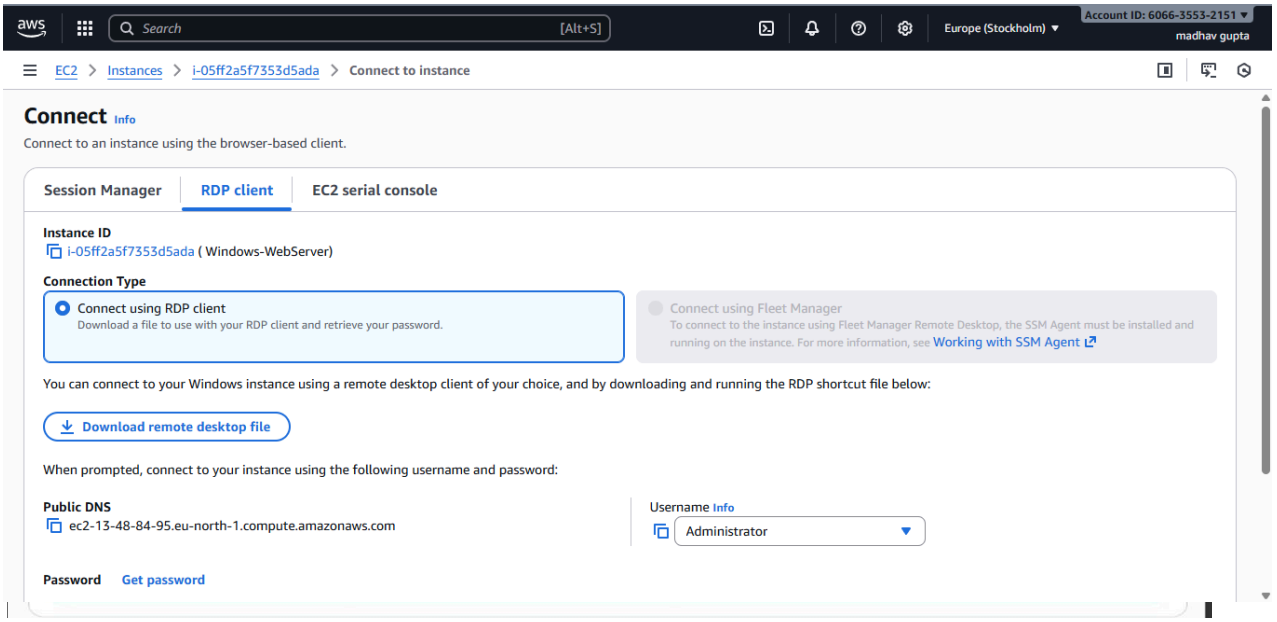
Step 3 Verify SSM Agent is Installed

For Amazon Linux 2 or Windows:

- The **SSM Agent** is preinstalled.

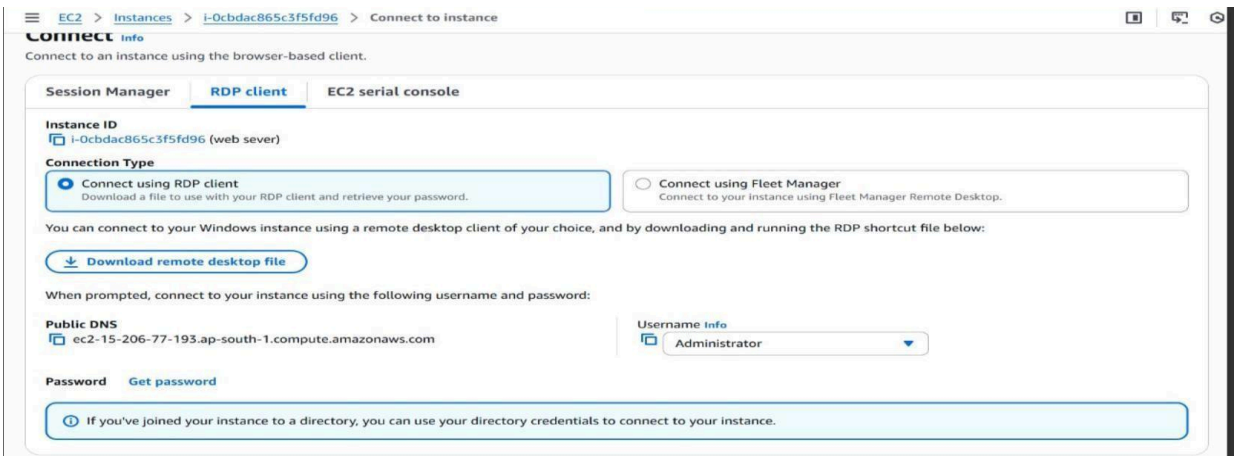
For Other OS (e.g., Ubuntu, CentOS):

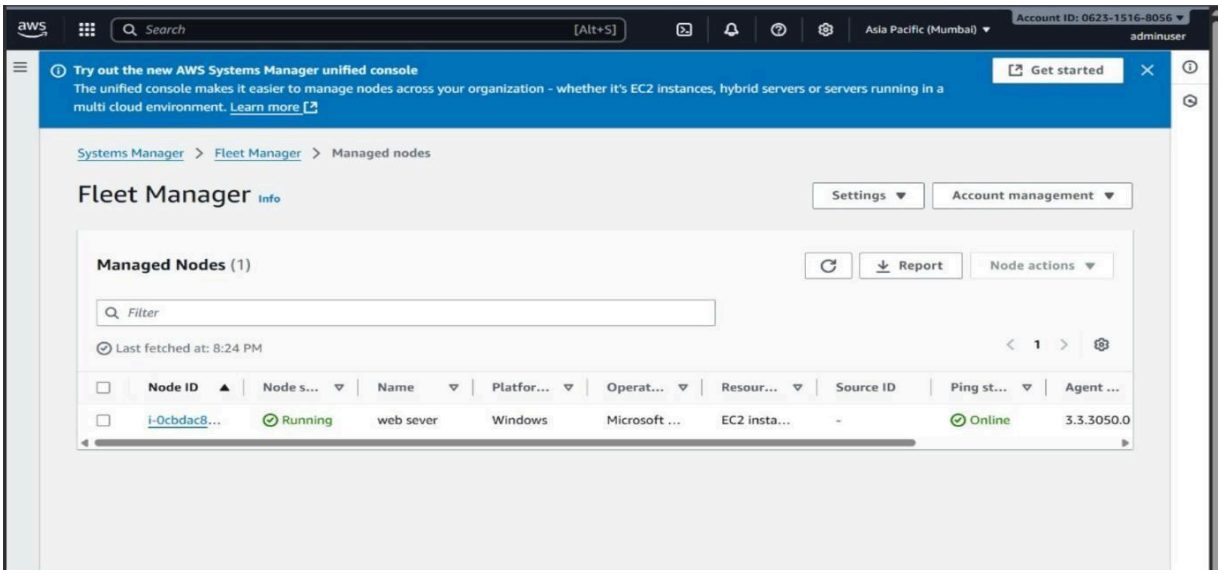
1. Connect via SSH or RDP.
2. Run these commands:



Step 4: Open Fleet Manager

1. In the AWS Console, go to **Systems Manager**.
2. In the left menu, under **Node Management**, click **Fleet Manager**.
3. You should now see your EC2 instance **listed**.
    - If it doesn't appear yet, wait a few minutes for SSM to register it.



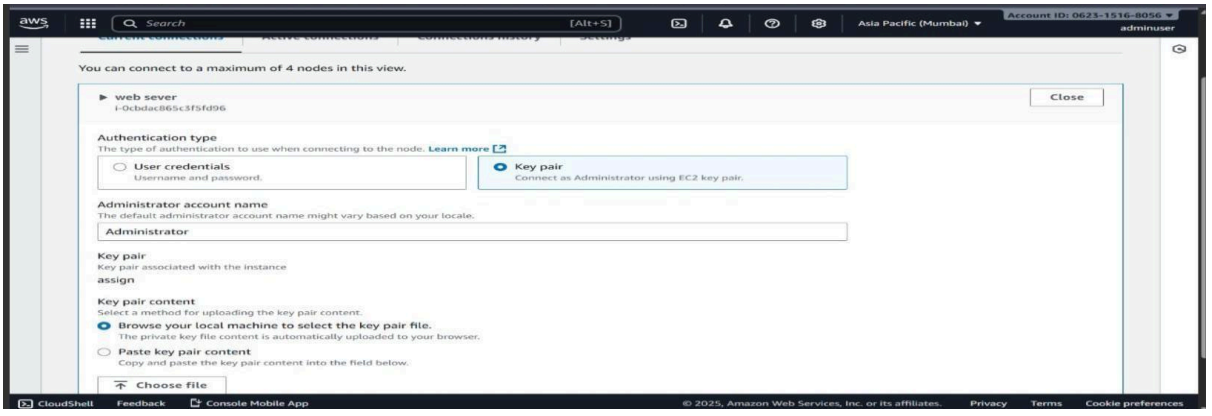


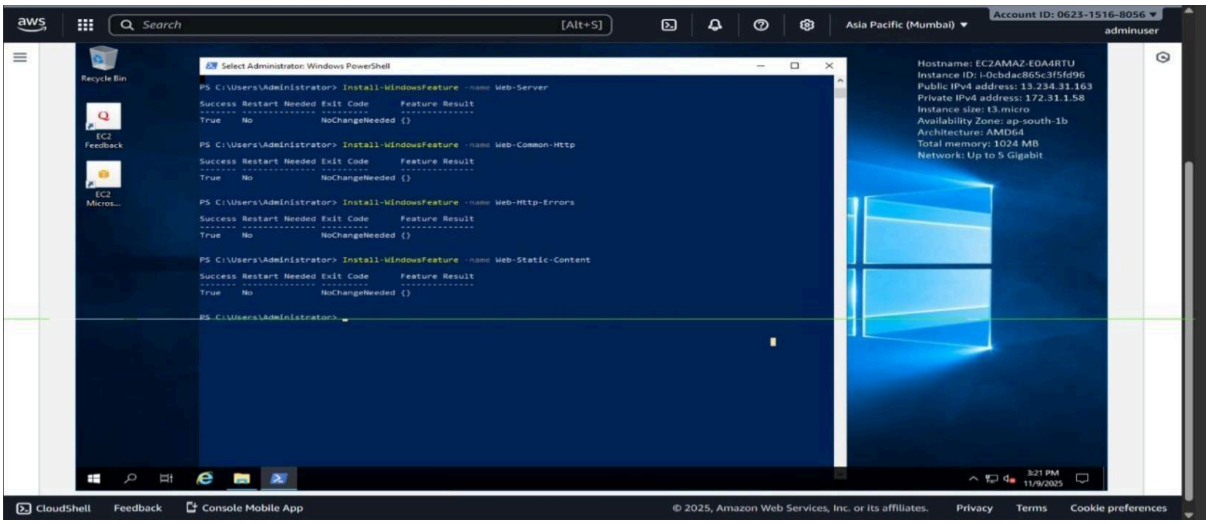
Step 5: Manage Instance via Fleet Manager

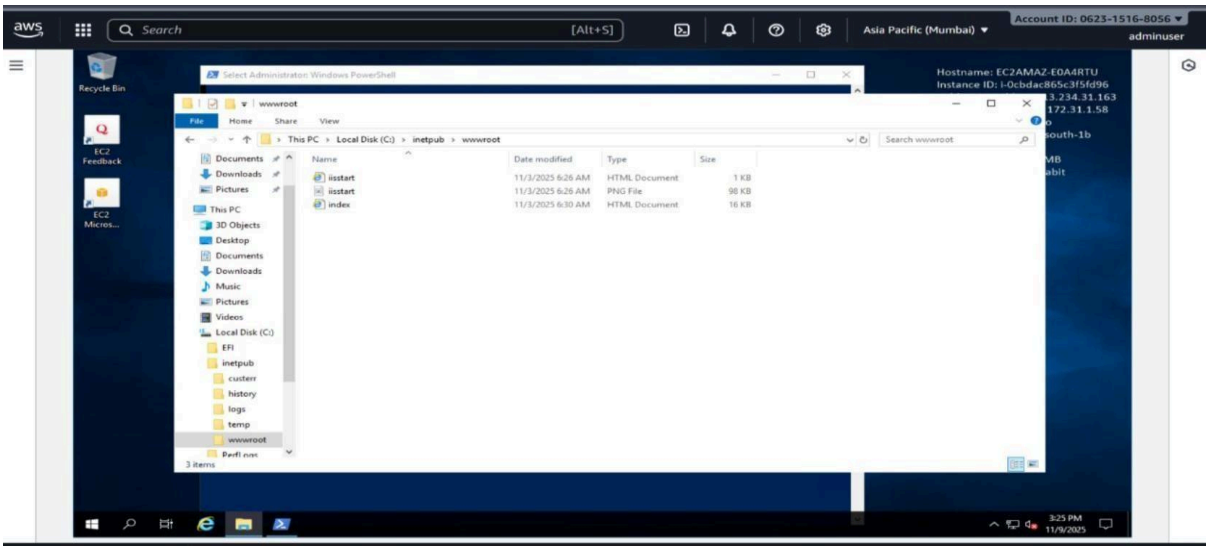
Now you can manage your EC2 instance **without RDP or SSH**:

From Fleet Manager Dashboard:

- Select your instance → click **Node actions** or instance name. You can now:
- View system info (CPU, memory, network, OS details)
- Browse and edit files
- View event logs
- Manage users
- Run PowerShell/Command Line directly in console
- Access the Windows registry (for Windows)
- Reboot or stop instance remotely







Final Output: