

Step 1: Launch an EC2 Instance (for Jupyter Notebook)

1. Go to EC2 Dashboard:

- Once logged in to the AWS console, search for **EC2** in the search bar and click on it to open the EC2 Dashboard.

2. Launch Instance:

- Click **Launch Instance** to start creating a new virtual machine (VM) on AWS.
- Select an **Amazon Machine Image (AMI)**. Choose **Ubuntu Server 20.04 LTS**.
- Choose an instance type. For a free-tier, select **t2.micro**.
- Configure the instance and leave the default options for most fields.

3. Create Key Pair (for SSH access):

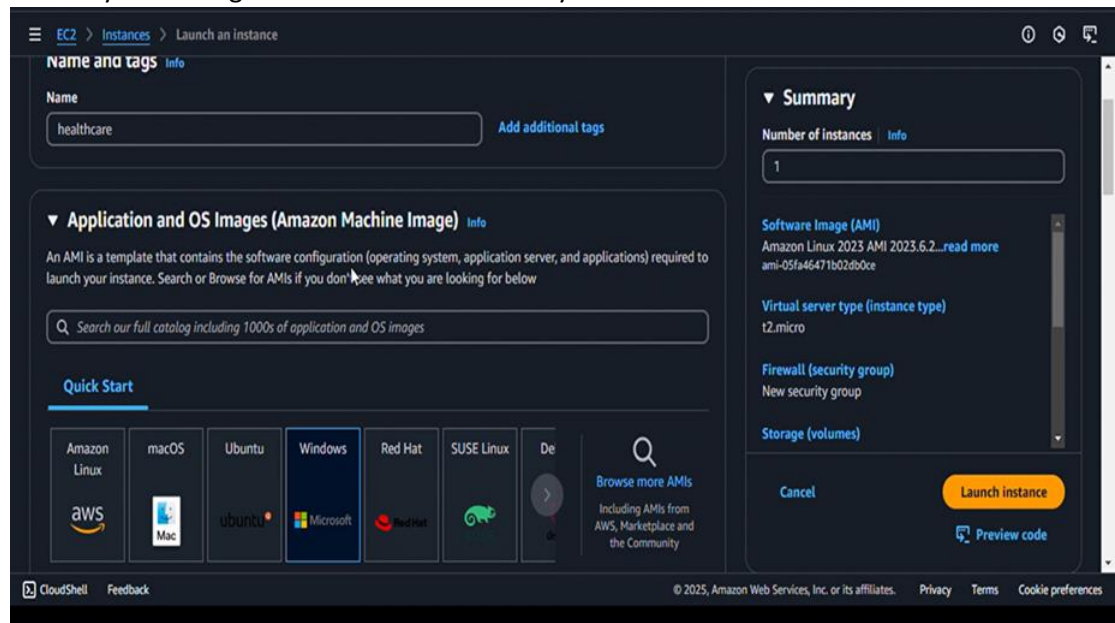
- Under **Key Pair**, click **Create a new key pair**.
- Give the key a name , then download the .pem file. This file is required to SSH into your EC2 instance.

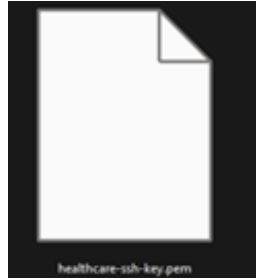
4. Security Group:

- In the **Configure Security Group** step, create a new security group that allows **SSH** (port 22) and **HTTP** (port 80) or **Custom TCP (port 8888)** for Jupyter access.

5. Launch Instance:

- Review your settings and click **Launch** to start your EC2 instance.





Step 2: SSH into EC2 Instance

1. Get EC2 Public IP:

- After the instance is running, go to the **Instances** page on the EC2 dashboard.
- Copy the **Public IP address** of your instance (e.g., 15.206.186.182).

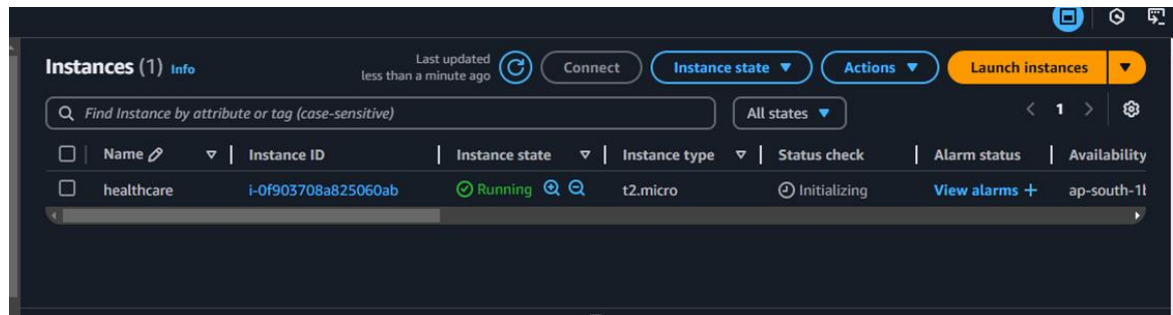
2. SSH Access:

- On your local machine, open a terminal (or PowerShell if on Windows) and navigate to the folder where your .pem file is located.
- Run the following SSH command to connect to your EC2 instance (replace the your-key.pem with your actual key file and your-public-ip with the public IP of your instance):

```
ssh -i "your-key.pem" ubuntu@your-public-ip
```

- Confirm the connection by typing **yes** when prompted.

The screenshot shows the 'Connect to instance' dialog in the AWS Management Console. The breadcrumb trail at the top reads: EC2 > Instances > i-0f903708a825060ab > Connect to instance. The dialog has four tabs: 'EC2 Instance Connect' (selected), 'Session Manager', 'SSH client', and 'EC2 serial console'. Under 'Instance ID', it shows 'i-0f903708a825060ab (healthcare)'. The 'Connection Type' section has two options: 'Connect using EC2 Instance Connect' (selected) and 'Connect using EC2 Instance Connect Endpoint'. Below this, the 'Public IPv4 address' is listed as '3.110.208.117', with 'IPv6 address' set to '-'. The 'Username' field contains 'ubuntu'. A note at the bottom states: 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right are 'Cancel' and 'Connect' buttons.



Step 3: Install Jupyter Notebook on EC2

1. **Update Packages:**

```
sudo apt update -y
```

2. **Install Python3 and pip:**

```
sudo apt install python3-pip -y
```

3. **Install Jupyter Notebook:**

```
pip3 install jupyter
```

Step 4: Configure Jupyter Notebook for External Access

1. **Generate Jupyter Config File:**

```
jupyter notebook --generate-config
```

2. **Allow Access from Any IP:**

- Open the Jupyter config file in a text editor:

```
nano ~/.jupyter/jupyter_notebook_config.py
```

- Add the following line to the config file:

```
c.NotebookApp.ip = '0.0.0.0'
```

- Save and close the editor (Ctrl + X, then Y to confirm changes).

Step 5: Start Jupyter Notebook on EC2

1. **Start Jupyter Notebook:**

```
jupyter notebook --port=8888 --no-browser --allow-root
```

2. **Copy the Token URL:**

- After running the above command, Jupyter will print a URL with a token like this:

```
http://127.0.0.1:8888/?token=your_token_here
```

Step 6: Access Jupyter Notebook in Browser

1. Open Your Browser:

- In the address bar, enter the following URL (replace your-public-ip with the public IP of your EC2 instance):

`http://your-public-ip:8888`

2. Enter the Token:

- Copy the token from the terminal and paste it into the browser to access your Jupyter Notebook interface.

Step 7: Create an S3 Bucket

1. Go to S3 Dashboard:

- In the AWS Console, search for **S3** and go to the S3 service.

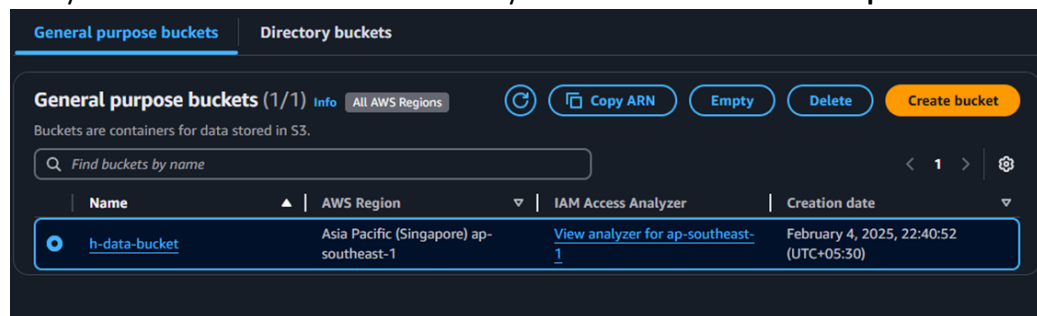
2. Create a New Bucket:

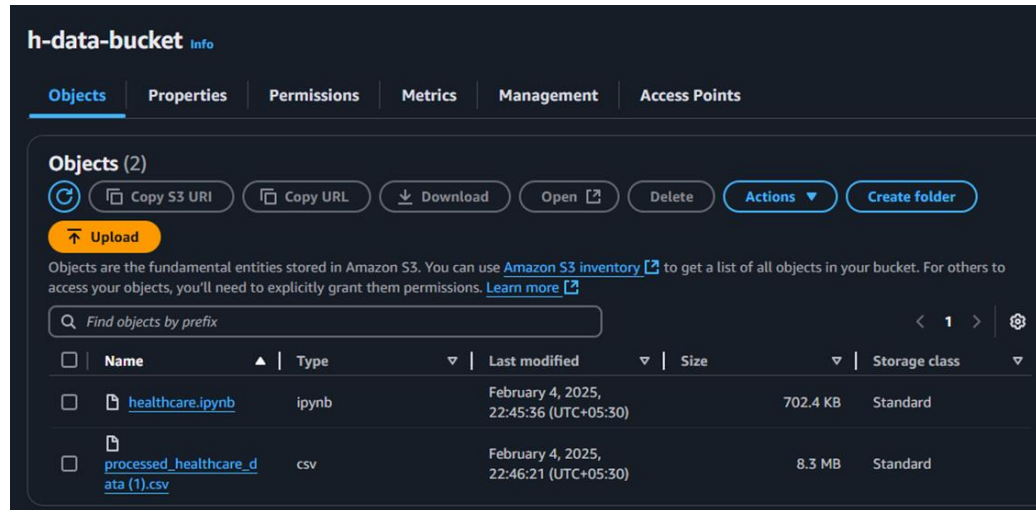
- Click **Create Bucket** and give it a unique name.
- Select the **region** where your EC2 instance is located.
- Leave the default options, or adjust as needed for your use case.
- Click **Create Bucket**.

Step 8: Upload Files to S3

1. Upload Data Files:

- In the S3 bucket dashboard, click **Upload**.
- Add your .csv files or other datasets from your local machine and click **Upload**.





Step 9: Access Files from S3 in Jupyter Notebook

1. Download Files from S3 Using Boto3:

- Example Python code to download your dataset (healthcare.csv) from S3 into your Jupyter Notebook:

```
import pandas as pd

s3 = boto3.client('s3')

bucket_name = 'your-bucket-name'

file_name = 'healthcare.csv'

# Download the file from S3 to local EC2 instance
s3.download_file(bucket_name, file_name, file_name)

# Load the CSV file into a pandas DataFrame
data = pd.read_csv(file_name)

print(data.head())
```

Step 10: Process Data and Upload Results Back to S3

1. Process Data in Jupyter:

- Run your analysis and data processing code in Jupyter.

2. Upload Results to S3:

- Save your processed data to a .csv file and upload it back to S3:

```
# Save DataFrame to CSV
```

```
data.to_csv('processed_data.csv', index=False)
```

```
# Upload the CSV file to S3
```

```
s3.upload_file('processed_data.csv', bucket_name, 'processed_data.csv')
```

```
aws [Alt+S] Search Asia Pacific (Mumbai) Account ID: 4182-7

Last login: Thu Feb  6 05:16:30 2025 from 13.233.177.3
ubuntu@ip-172-31-14-17:~$ jupyter notebook --generate-config
echo "c.NotebookApp.ip = '0.0.0.0'" >> ~/.jupyter/jupyter_notebook_config.py
jupyter notebook --port=8888 --no-browser --allow-root
Overwrite /home/ubuntu/.jupyter/jupyter_notebook_config.py with default config? [y/N]y
Writing default config to: /home/ubuntu/.jupyter/jupyter_notebook_config.py
[W 2025-02-10 02:49:46.884 ServerApp] ServerApp.token config is deprecated in 2.0. Use IdentityProvider.token.
[I 2025-02-10 02:49:47.391 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-02-10 02:49:47.395 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-02-10 02:49:47.400 ServerApp] jupyterlab | extension was successfully linked.
[W 2025-02-10 02:49:47.402 JupyterNotebookApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to
e your config before our next release.
[I 2025-02-10 02:49:47.404 ServerApp] notebook | extension was successfully linked.
[I 2025-02-10 02:49:48.028 ServerApp] notebook_shim | extension was successfully linked.
[W 2025-02-10 02:49:48.114 ServerApp] All authentication is disabled. Anyone who can connect to this server will be able to run code.
[I 2025-02-10 02:49:48.115 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-02-10 02:49:48.116 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-02-10 02:49:48.117 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-02-10 02:49:48.127 LabApp] JupyterLab extension loaded from /home/ubuntu/.local/lib/python3.12/site-packages/jupyterlab
[I 2025-02-10 02:49:48.128 LabApp] JupyterLab application directory is /home/ubuntu/.local/share/jupyter/lab
[I 2025-02-10 02:49:48.128 LabApp] Extension Manager is 'pypi'.
[I 2025-02-10 02:49:48.228 ServerApp] jupyterlab | extension was successfully loaded.

i-0f903708a825060ab (healthcare)
PublicIPs: 3.110.208.117 PrivateIPs: 172.31.14.17
```

