## 1. What is EC2?

- **EC2 =** Elastic Cloud Compute
- Scalable virtual machines in the cloud
- Pay-as-you-go pricing
- Choose from many instance types and OS options (Linux, Windows, etc.)

## 2. EC2 Features

- **Elasticity**: Scale up/down instances as needed
- Use **Amazon Machine Images (AMIs)** for preconfigured environments
- Easily integrates with services like **S3**, **RDS**, etc.

## 3. EC2 Lifecycle

- **Launch** – First-time start
- **Start/Stop** – Pause and resume usage
- **Terminate** – Permanently delete
- **Reboot** – Restart without deleting root data

## 4. EC2 Storage Options

- **Instance Store** – Fast but temporary storage
- **EFS** – Shared file storage across instances
- **EBS** – Persistent block-level storage
- **S3** – Object storage; useful for backups or large datasets

## 5. Common EC2 Use Cases

- Web Hosting (web servers, app servers)
- Data Processing (ETL pipelines, batch jobs)
- Machine Learning (GPU-enabled training)
- Disaster Recovery (backups, redundancy)

## 6. Ubuntu VM Essentials (on EC2)

- Default user: `ubuntu`
- Use `sudo` for admin commands
- Package manager: `apt`
  (like Homebrew for macOS or Choco for Windows)

# Update packages

sudo apt update

sudo apt upgrade

## 7. Installing Miniconda on EC2

curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

bash ./Miniconda3-latest-Linux-x86_64.sh

## 8. Installing & Running Streamlit

# After reconnecting to EC2:

pip --version  # check pip

pip install streamlit scikit-learn

# Create a project directory

mkdir web

cd web

# Create app file

nano test.py

**Sample `test.py` Streamlit App**:

```
import streamlit as st

def main():
    st.title("Welcome to my Streamlit App")
    st.write("## Data Sets")
    st.write("""
        - data set 01
        - data set 02
        - data set 03
    """)
    st.write("\n")
    st.write("## Goodbye!")
if __name__ == "__main__":
    main()
```

Run the app:

```
streamlit run test.py
```

# Be sure to open the correct port in your EC2 Security Group to access the Streamlit app from a browser.

## 9. AWS Lambda Overview

- **Serverless computing**: No need to manage servers
- Only pay for execution time (unlike EC2, which runs 24/7)
- Automatically runs code in response to **events**

## 10. Lambda Features

- Triggered by events from other AWS services
- Supports multiple runtimes: **Python, Java, Node.js**, etc.
- Fully managed & highly scalable
- Integrated with services like **S3**, **DynamoDB**, **API Gateway**, etc.

## 11. How Lambda Works

1. Upload your code using the AWS Management Console
2. Define event source (e.g., S3 upload, DynamoDB change)
3. Lambda runs automatically when the event is triggered