

# Deployment of Streamlit-Based Tomato Price Prediction Application on AWS EC2 with CI/CD Integration

Prepared by: [Your Name]

November 3, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Creating an EC2 Instance on AWS</b>	<b>3</b>
2.1	Step 1 — Access EC2 Console . . . . .	3
2.2	Step 2 — Name and Tags . . . . .	3
2.3	Step 3 — Choose AMI (Application and OS Image) . . . . .	3
2.4	Step 4 — Choose Instance Type . . . . .	3
2.5	Step 5 — Configure Key Pair (Login) . . . . .	3
2.6	Step 6 — Configure Network Settings . . . . .	4
2.7	Step 7 — Configure Storage . . . . .	4
2.8	Step 8 — Advanced Details (User Data) . . . . .	4
2.9	Step 9 — Review and Launch . . . . .	4
2.10	Step 10 — Connect to the Instance . . . . .	4
2.11	Step 11 — Setup Python Environment . . . . .	5
<b>3</b>	<b>CI/CD Pipeline Setup (GitHub Actions + EC2)</b>	<b>6</b>
3.1	Step 1 — Generate EC2 SSH Key for GitHub . . . . .	6
3.2	Step 2 — Add Secrets to GitHub . . . . .	6
3.3	Step 3 — Create Workflow File . . . . .	6
3.4	Step 4 — Create Streamlit Service . . . . .	7
3.5	Step 5 — Verify Deployment . . . . .	7
3.6	Step 6 — CI/CD Workflow Summary . . . . .	8
<b>4</b>	<b>Conclusion</b>	<b>8</b>

# 1. Introduction

This document provides a comprehensive, step-by-step guide for deploying and automating a Streamlit-based machine learning application — in this case, the **Tomato Price Prediction App** — using Amazon Web Services (AWS) EC2 and GitHub Actions (CI/CD).

The process includes:

- Creating and configuring an AWS EC2 instance
- Setting up Python and Streamlit environment
- Creating a systemd service for automatic uptime
- Implementing a CI/CD pipeline with GitHub Actions

## 2. Creating an EC2 Instance on AWS

This section outlines the detailed process for launching and configuring an Amazon EC2 instance to host the Streamlit app.

### 2.1. Step 1 — Access EC2 Console

1. Log in to <https://aws.amazon.com/console>.
2. Search and open **EC2**.
3. Choose the region: **Asia Pacific (Sydney)** (ap-southeast-2).
4. Click **Instances** → **Launch Instances**.

### 2.2. Step 2 — Name and Tags

- Name: `streamlit-tomato-app`
- Add tags:
  - Key: `Environment`, Value: `Production`
  - Key: `Project`, Value: `TomatoPricePrediction`

### 2.3. Step 3 — Choose AMI (Application and OS Image)

- Select **Ubuntu Server 24.04 LTS (HVM), SSD Volume Type**.
- Architecture: **64-bit (x86)**.
- Ensure it is marked as **Free Tier eligible**.

### 2.4. Step 4 — Choose Instance Type

Instance Type	vCPUs	Memory (GiB)	Network Performance
t2.micro	1	1	Up to 5 Gbps (burstable)

Suitable for small ML/Streamlit projects.

### 2.5. Step 5 — Configure Key Pair (Login)

- Create new key pair:
  - Name: `streamlit-key`
  - Type: `RSA`
  - Format: `.pem`
- Save to a secure location:  
`C:\Users\User\Desktop\Tomato_Deploy\streamlit-key.pem`

## 2.6. Step 6 — Configure Network Settings

Type	Protocol	Port	Source	Purpose
SSH	TCP	22	0.0.0.0/0	Secure SSH access
HTTP	TCP	80	0.0.0.0/0	Web access
Custom TCP	TCP	8501	0.0.0.0/0	Streamlit App Port

Ensure **Auto-assign Public IP** is enabled.

## 2.7. Step 7 — Configure Storage

Property	Value
Volume Type	gp3
Size	8 GiB
Encryption	Enabled
Delete on Termination	Yes

## 2.8. Step 8 — Advanced Details (User Data)

```
#!/bin/bash
apt update -y
apt install python3-pip python3-venv -y
pip install streamlit plotly pandas
```

## 2.9. Step 9 — Review and Launch

Check:

- AMI: Ubuntu 24.04 LTS
- Instance Type: t3.micro
- Key Pair: streamlit-key
- Security Group: streamlit-sg

Then click **Launch Instance**.

## 2.10. Step 10 — Connect to the Instance

**For Windows:**

```
cd C:\Users\User\Desktop\Tomato_Deploy
icaccls "streamlit-key.pem" /inheritance:r
icaccls "streamlit-key.pem" /grant:r "$($env:USERNAME):(R)"
ssh -i "streamlit-key.pem" ubuntu@<EC2_PUBLIC_IP>
```

**For Linux/Mac:**

```
chmod 400 streamlit-key.pem
ssh -i "streamlit-key.pem" ubuntu@<EC2_PUBLIC_IP>
```

## 2.11. Step 11 — Setup Python Environment

```
sudo apt update && sudo apt upgrade -y
sudo apt install python3 python3-pip python3-venv git -y
mkdir Tomato_Deploy && cd Tomato_Deploy
python3 -m venv venv
source venv/bin/activate
```

### 3. CI/CD Pipeline Setup (GitHub Actions + EC2)

This section describes how to automate deployment using GitHub Actions.

#### 3.1. Step 1 — Generate EC2 SSH Key for GitHub

```
cd ~/.ssh
ssh-keygen -t rsa -b 4096 -C "github-deploy-key" -f github_actions
```

Copy public key:

```
cat ~/.ssh/github_actions.pub
```

Add it in GitHub under:

Repository → Settings → Deploy Keys → Add Deploy Key

Enable “Allow write access”.

#### 3.2. Step 2 — Add Secrets to GitHub

Secret Name	Value
EC2_HOST	Public IPv4 of EC2 (e.g., 3.26.66.140)
EC2_SSH_KEY	Content of private key (streamlit-key.pem)
EC2_USER	ubuntu

#### 3.3. Step 3 — Create Workflow File

Create:

```
.github/workflows/deploy.yml
```

Paste:

```
name: Deploy Streamlit App to EC2

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest

    env:
      GIT_TERMINAL_PROMPT: 0
      GIT_SSH_COMMAND: "ssh -o StrictHostKeyChecking=no"

    steps:
      - name: Checkout code
        uses: actions/checkout@v3
        with:
          persist-credentials: false
          fetch-depth: 0
```

```
- name: Configure Git
  run: |
    git config --global user.email "github-actions@github.com"
    git config --global user.name "GitHub Actions Bot"

- name: Deploy to EC2
  uses: appleboy/ssh-action@v1.0.3
  with:
    host: ${ secrets.EC2_HOST }
    username: ubuntu
    key: ${ secrets.EC2_SSH_KEY }
    port: 22
    script: |
      set -e
      cd ~/Tomato_Deploy || exit 1
      git fetch origin main || git clone https://github.com/
        lamanabin2046/tomato_price_prediction_app.git .
      git reset --hard origin/main
      source venv/bin/activate
      pip install -r requirements.txt
      sudo systemctl daemon-reload
      sudo systemctl restart streamlit
      echo "    Deployment successful!"
```

### 3.4. Step 4 — Create Streamlit Service

```
sudo nano /etc/systemd/system/streamlit.service
```

Paste:

```
[Unit]
Description=Streamlit App
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/Tomato_Deploy
ExecStart=/home/ubuntu/Tomato_Deploy/venv/bin/streamlit run app.py --
  server.port 8501 --server.address 0.0.0.0
Restart=always

[Install]
WantedBy=multi-user.target
```

Activate service:

```
sudo systemctl daemon-reload
sudo systemctl enable streamlit
sudo systemctl start streamlit
sudo systemctl status streamlit
```

### 3.5. Step 5 — Verify Deployment

```
http://<EC2_PUBLIC_IP>:8501
```

Example:

`http://3.26.66.140:8501`

### 3.6. Step 6 — CI/CD Workflow Summary

Step	Description
1	Generate SSH key pair for GitHub Actions
2	Add Deploy Key and Secrets in GitHub
3	Create <code>deploy.yml</code> GitHub Workflow
4	Push code to main branch
5	GitHub auto-connects to EC2 and redeploys code
6	EC2 restarts Streamlit via systemd
7	Access app at <code>http://&lt;EC2-IP&gt;:8501</code>

## 4. Conclusion

By following this documentation, a fully automated deployment pipeline is achieved. Any updates pushed to the GitHub repository automatically redeploy the Streamlit app on AWS EC2, ensuring continuous integration and deployment with zero manual intervention.