

Hosting Web-App on AWS

Author: Tomisin Abimbola Adeniyi

Submission Date: 5/04/2024

Batch Code: LISUM31

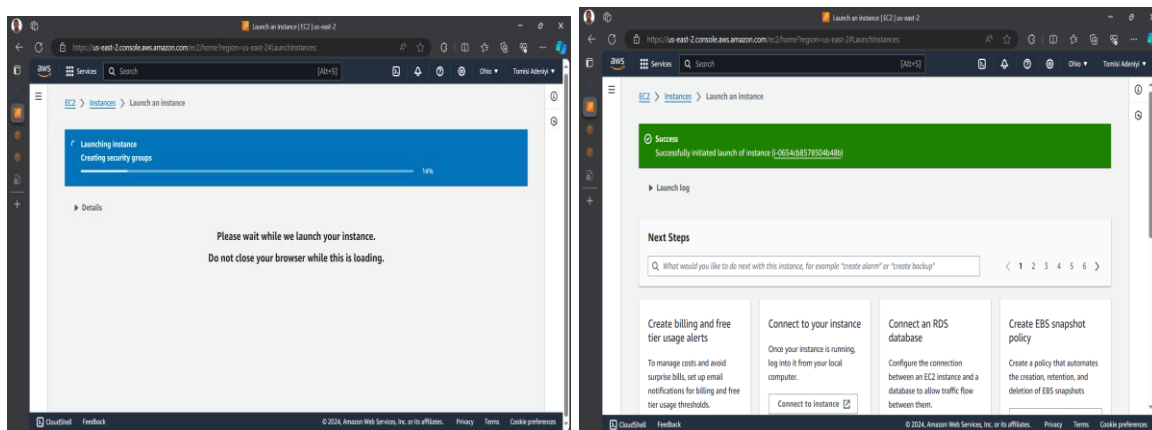
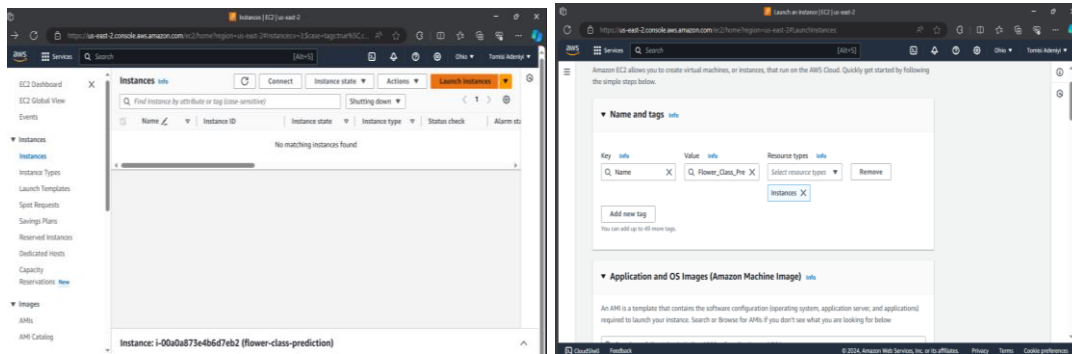
Submitted To: GitHub

Introduction

The objective of this project is to deploy a machine learning (ML) model developed as part of the Week 4 task onto the cloud.

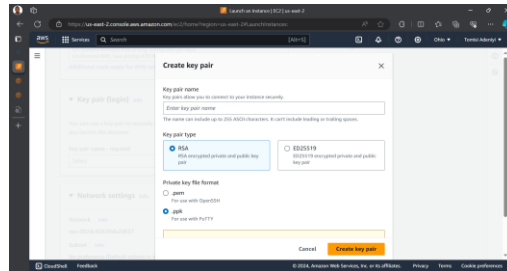
Here's a description of the steps take to host an app on AWS:

1. **Amazon Web Services (AWS) EC2 Instance Management:** This involves launching an instance, which is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

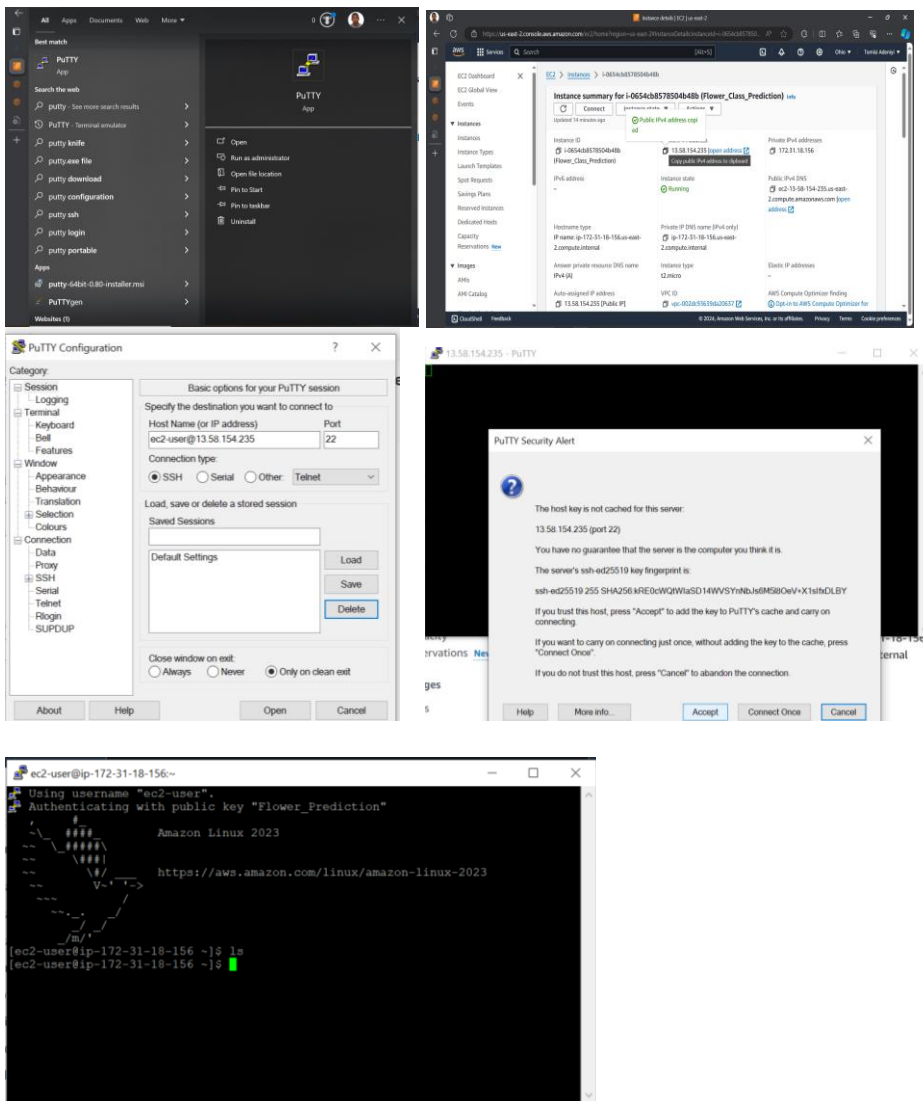


Hosting Web-App on AWS

2. **SSH Key Pair Creation:** For secure login to the instance, an SSH key pair is created. This is crucial for authentication purposes.



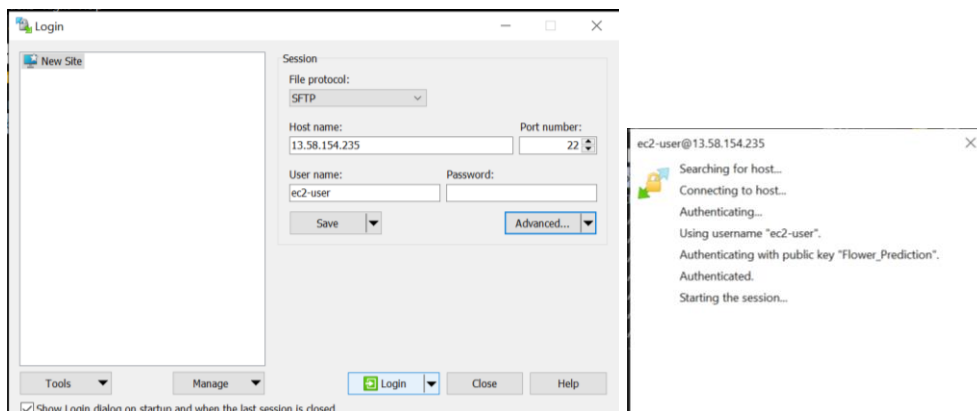
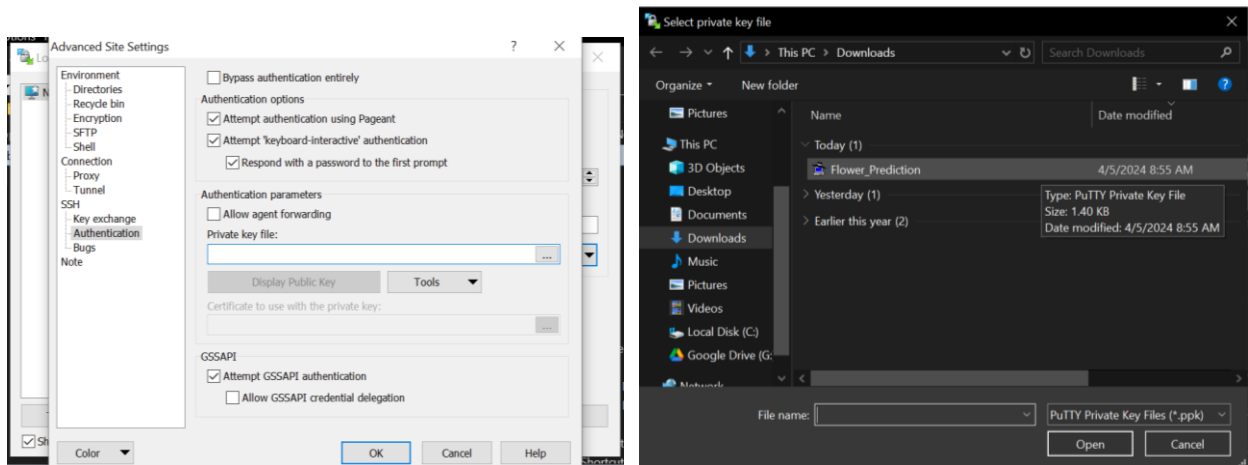
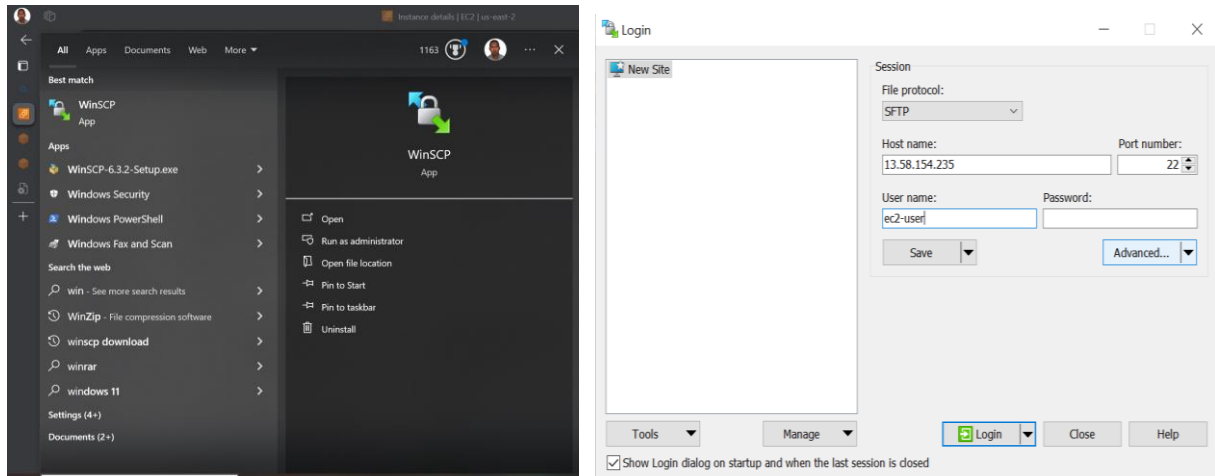
3. **Connecting to an Instance:** Connecting to the instance after setting up the instance and the SSH keys, you connected to the instance. This was done through SSH client.



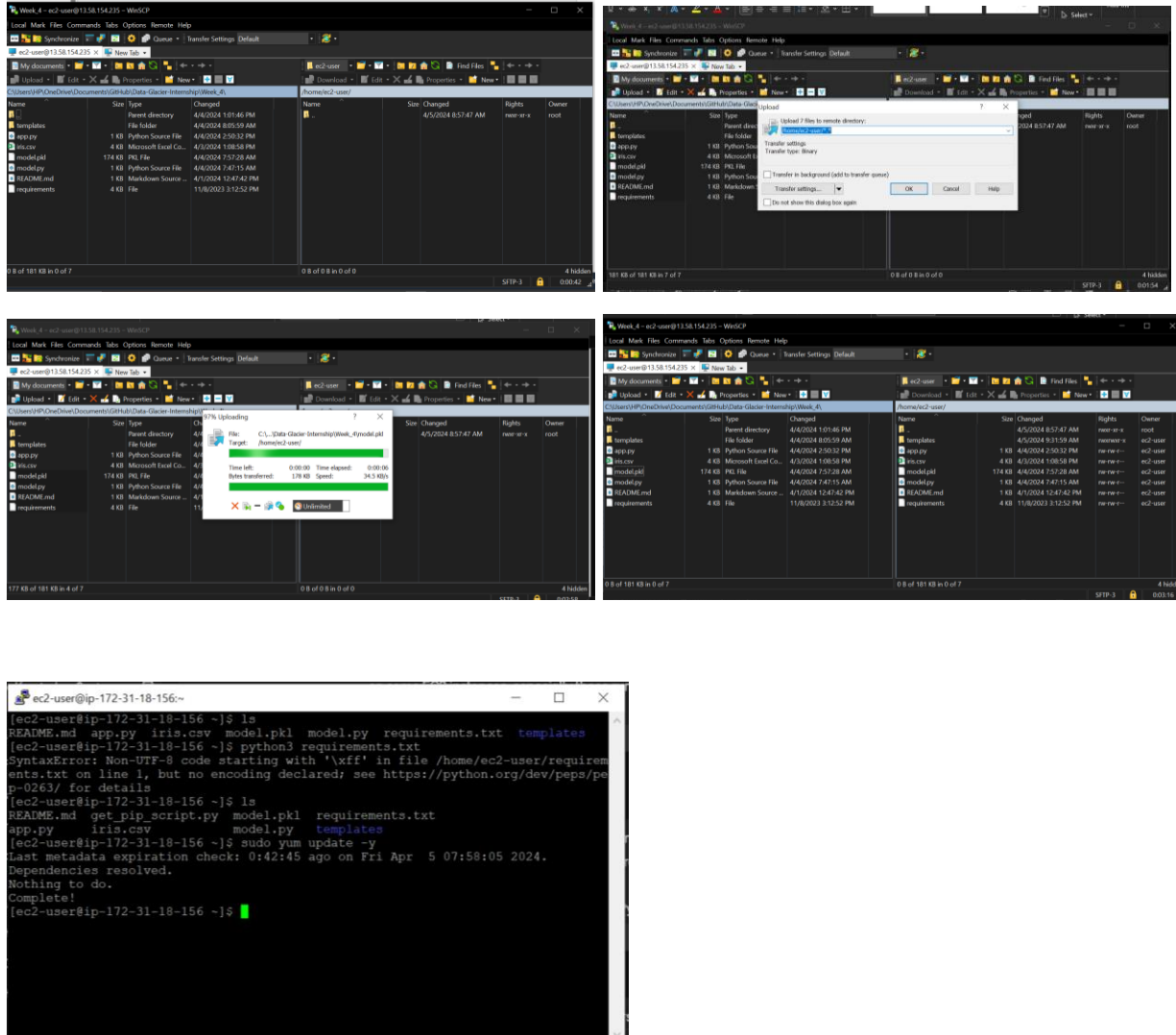
Instance successfully connected

Hosting Web-App on AWS

4. **File Transfers with WinSCP:** For managing files on the instance, WinSCP was used. These tools allow files transfer to/from the instance and also execute commands on the instance.



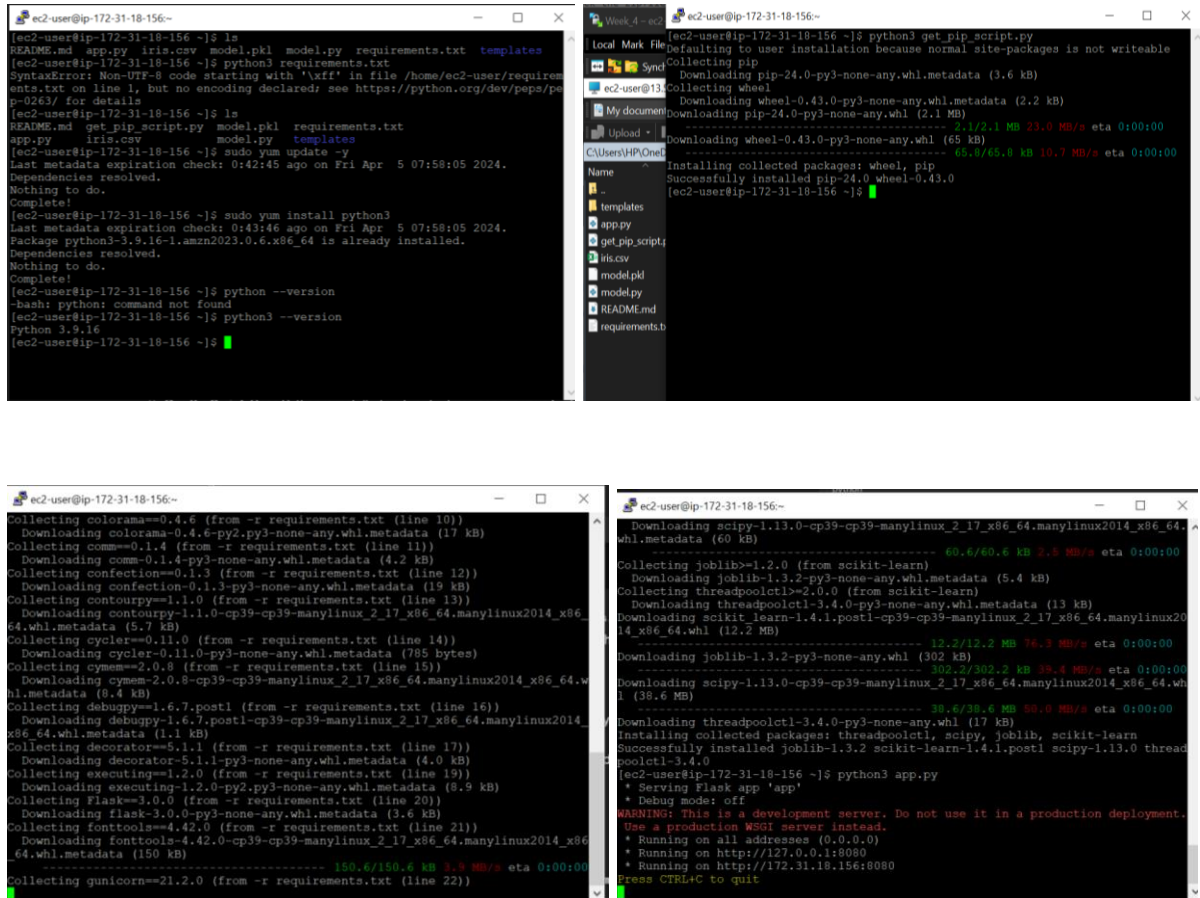
Hosting Web-App on AWS



Files successfully copied to the instance

Hosting Web-App on AWS

5. **Installing Packages:** After connecting to the instance, necessary packages need to be installed for the application to run. This included Python, pip, numpy, scikit-learn and other requirements.



```
ec2-user@ip-172-31-18-156:~$ ls
README.md app.py iris.csv model.pkl model.py requirements.txt templates
ec2-user@ip-172-31-18-156:~$ python3 requirements.txt
SyntaxError: Non-UTF-8 code starting with '\xff' in file /home/ec2-user/requir
ments.txt on line 1, but no encoding declared: see https://python.org/dev/peps/pe
p-0263/ for details
ec2-user@ip-172-31-18-156:~$ ls
README.md get_pip_script.py model.pkl requirements.txt
app.py iris.csv model.py templates
ec2-user@ip-172-31-18-156:~$ sudo yum update -y
Last metadata expiration check: 0:42:45 ago on Fri Apr 5 07:58:05 2024.
Dependencies resolved.
Nothing to do.
Complete!
ec2-user@ip-172-31-18-156:~$ sudo yum install python3
Last metadata expiration check: 0:43:46 ago on Fri Apr 5 07:58:05 2024.
Package python3-3.9.16-1.amzn2023.0.6.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
ec2-user@ip-172-31-18-156:~$ python --version
-bash: python: command not found
ec2-user@ip-172-31-18-156:~$ python3 --version
Python 3.9.16
ec2-user@ip-172-31-18-156:~$

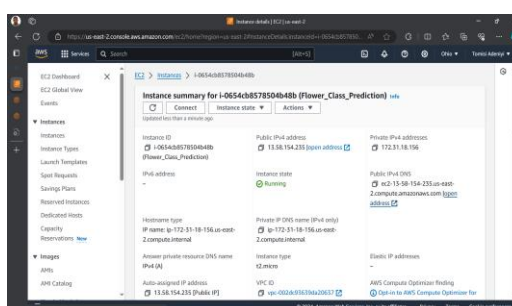
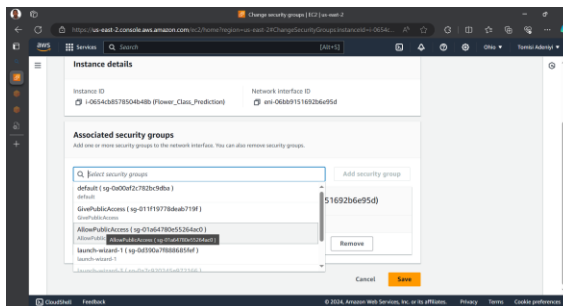
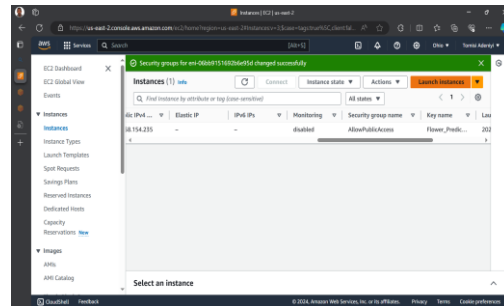
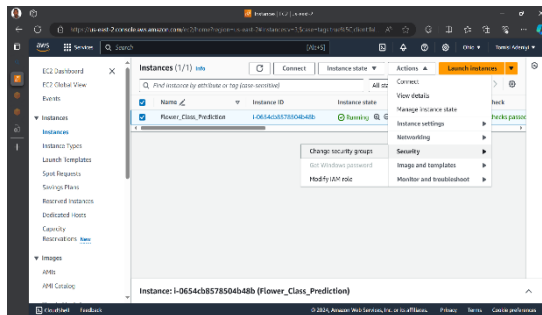
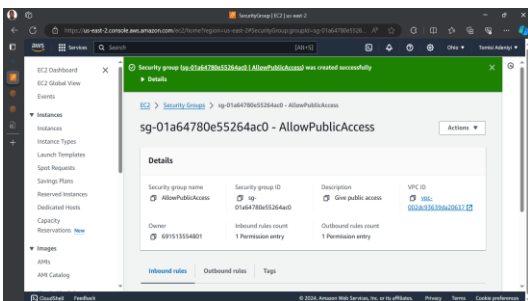
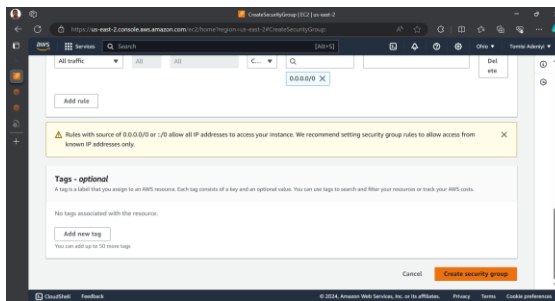
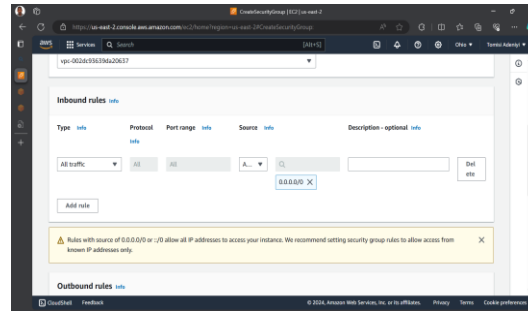
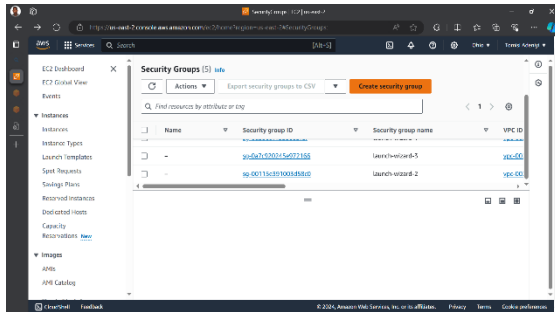
Local Mark File ec2-user@ip-172-31-18-156:~$ python3 get_pip_script.py
Defaulting to user installation because normal site-packages is not writeable
Collecting pip
  Downloading pip-24.0-py3-none-any.whl.metadata (3.6 kB)
Collecting wheel
  Downloading wheel-0.43.0-py3-none-any.whl.metadata (2.2 kB)
  Downloading pip-24.0-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 23.0 MB/s eta 0:00:00
  Downloading wheel-0.43.0-py3-none-any.whl (65 kB)
----- 65.0/65.0 kB 10.7 MB/s eta 0:00:00
Installing collected packages: wheel, pip
Successfully installed pip-24.0 wheel-0.43.0
ec2-user@ip-172-31-18-156:~$

Name
-
templates
app.py
get_pip_script.py
iris.csv
model.pkl
model.py
README.md
requirements.txt

ec2-user@ip-172-31-18-156:~$
Collecting colorama==0.4.6 (from -r requirements.txt (line 10))
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting comm==0.1.4 (from -r requirements.txt (line 11))
  Downloading comm-0.1.4-py3-none-any.whl.metadata (4.2 kB)
Collecting confection==0.1.3 (from -r requirements.txt (line 12))
  Downloading confection-0.1.3-py3-none-any.whl.metadata (19 kB)
Collecting contourpy==1.1.0 (from -r requirements.txt (line 13))
  Downloading contourpy-1.1.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.7 kB)
Collecting cycycler==0.11.0 (from -r requirements.txt (line 14))
  Downloading cycycler-0.11.0-py3-none-any.whl.metadata (785 bytes)
Collecting cymem==2.0.8 (from -r requirements.txt (line 15))
  Downloading cymem-2.0.8-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (0.4 kB)
Collecting debugpy==1.6.7.post1 (from -r requirements.txt (line 16))
  Downloading debugpy-1.6.7.post1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.1 kB)
Collecting decorator==5.1.1 (from -r requirements.txt (line 17))
  Downloading decorator-5.1.1-py3-none-any.whl.metadata (4.0 kB)
Collecting executing==1.2.0 (from -r requirements.txt (line 19))
  Downloading executing-1.2.0-py2.py3-none-any.whl.metadata (8.9 kB)
Collecting Flask==3.0.0 (from -r requirements.txt (line 20))
  Downloading flask-3.0.0-py3-none-any.whl.metadata (3.6 kB)
Collecting fonttools==4.42.0 (from -r requirements.txt (line 21))
  Downloading fonttools-4.42.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (150 kB)
----- 150.6/150.6 kB 3.9 MB/s eta 0:00:00
Collecting gunicorn==21.2.0 (from -r requirements.txt (line 22))
  Downloading gunicorn-21.2.0-py3-none-any.whl.metadata (4.9 kB)
----- 4.9/4.9 kB 10.7 MB/s eta 0:00:00
Collecting joblib==1.2.0 (from scikit-learn)
  Downloading joblib-1.2.0-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl==2.0.0 (from scikit-learn)
  Downloading threadpoolctl-2.0.0-py3-none-any.whl.metadata (13 kB)
Downloaded threadpoolctl-3.4.0-py3-none-any.whl.metadata (13 kB)
Downloaded scikit_learn-1.4.1.post1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.2 MB)
----- 12.2/12.2 MB 76.3 MB/s eta 0:00:00
Downloaded joblib-1.3.2-py3-none-any.whl (302 kB)
----- 302.2/302.2 kB 39.4 MB/s eta 0:00:00
Downloaded scipy-1.13.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (38.6 MB)
----- 38.6/38.6 MB 59.0 MB/s eta 0:00:00
Downloaded threadpoolctl-3.4.0-py3-none-any.whl (17 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.3.2 scikit-learn-1.4.1.post1 scipy-1.13.0 threadpoolctl-3.4.0
ec2-user@ip-172-31-18-156:~$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://172.31.18.156:8080
Press CTRL+C to quit
```

Hosting Web-App on AWS

6. **Creating a New Security Group:** To allow public access to the application, created a new security group was created in AWS. A security group acts as a virtual firewall that controls the traffic for one or more instances. A security group add rules that control the inbound traffic to instances, and outbound traffic from instances. In this case, the security group to open the access gate to the public.



Hosting Web-App on AWS

7. **Running the App on AWS:** After setting up your instance, installing necessary packages, and configuring security groups, you're ready to deploy your app.

