AWS Sentiment Analysis Project

Daniel Mehta   n01753264

Overview

This project is a **sentiment analysis web application** deployed on an **AWS EC2 instance**. It uses a **Flask web server** to provide an interface for **users to input text and receive sentiment predictions** using AWS AI services.

This implementation returns only the **overall sentiment classification (e.g., Positive, Negative, Neutral),** rather than showing all individual sentiment scores together.

The application consists of:

- **Front-end UI (index.html)** for user input.
- **Flask back-end (main.py)** to process text and interact with AWS services.
- **Helper script (comprehend.py)** for sentiment analysis (AWS Comprehend for sentiment analysis).

Setting Up the Application in EC2

Step 1: Connect to the EC2 Instance

Use SSH to connect:

```
ssh -i your-key.pem ec2-user@your-ec2-ip
```

Step 2: Install Required Dependencies

Ensure your system is updated and install the necessary Python libraries:

```
sudo yum update -y
sudo yum install python3-pip -y
pip3 install flask boto3
```

Step 3: Transfer Project Files

You can either create the files manually using nano:

```
nano main.py  # Copy & paste the Python backend code
nano comprehend.py  # Copy & paste the AWS AI logic
nano index.html  # Copy & paste the front-end HTML
```

Or

Transfer them using SCP

```
scp -i your-key.pem main.py comprehend.py index.html ec2-user@your-ec2-ip:~/
```

Step 4: Run the Application

Start the Flask application:

```
sudo python3 main.py --host=0.0.0.0 --port=5000
```

Once the application is running, open your browser and enter:

http://your-ec2-public-ip/

Expected Output

If a user enters 'I love this product,' the application will return a ‘POSITIVE’ sentiment.

Link to Demo Video

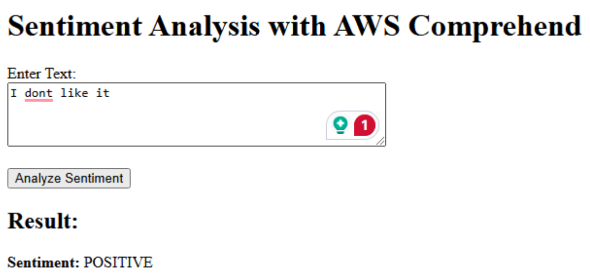
YouTube video of the working application: https://youtu.be/MHjkXWYV2_c

How the Application Works

1. **User Input**:
   - The user enters text into the input text box on the web page.
2. **Processing**:
   - When the user clicks the submit button, the application sends the text to the backend.
   - The backend uses AWS AI services to analyze the sentiment of the text.
3. **Output Display**:

- The sentiment result is **only updated after the button is clicked and the analysis is complete**.
- The webpage refreshes to show the determined sentiment (e.g., **Positive, Negative, or Neutral**).

Screenshots of the Application



Sentiment Analysis with AWS Comprehend

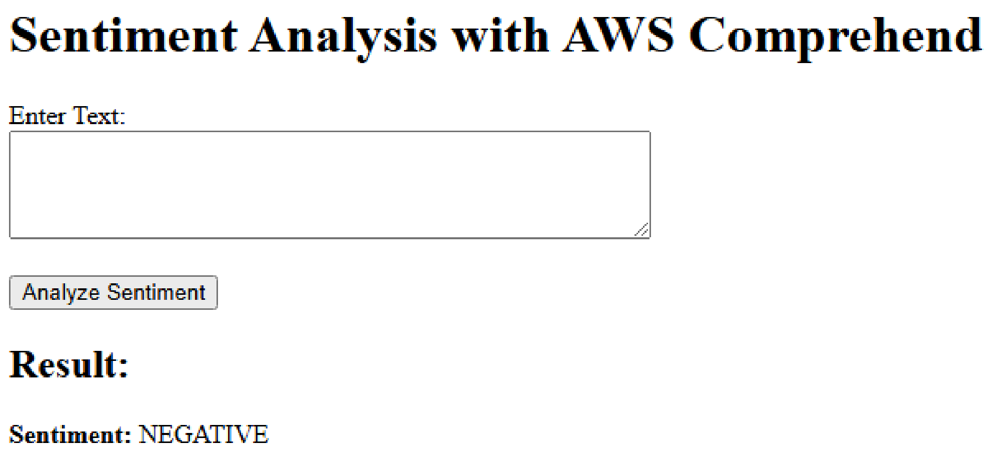
Enter Text:

I dont like it

Result:

Sentiment: POSITIVE

First, enter the text then click ‘Analyze Sentiment’. (Says Positive since the previous test was a positive commentt and it won’t be updated until the button is clicked)



Sentiment Analysis with AWS Comprehend

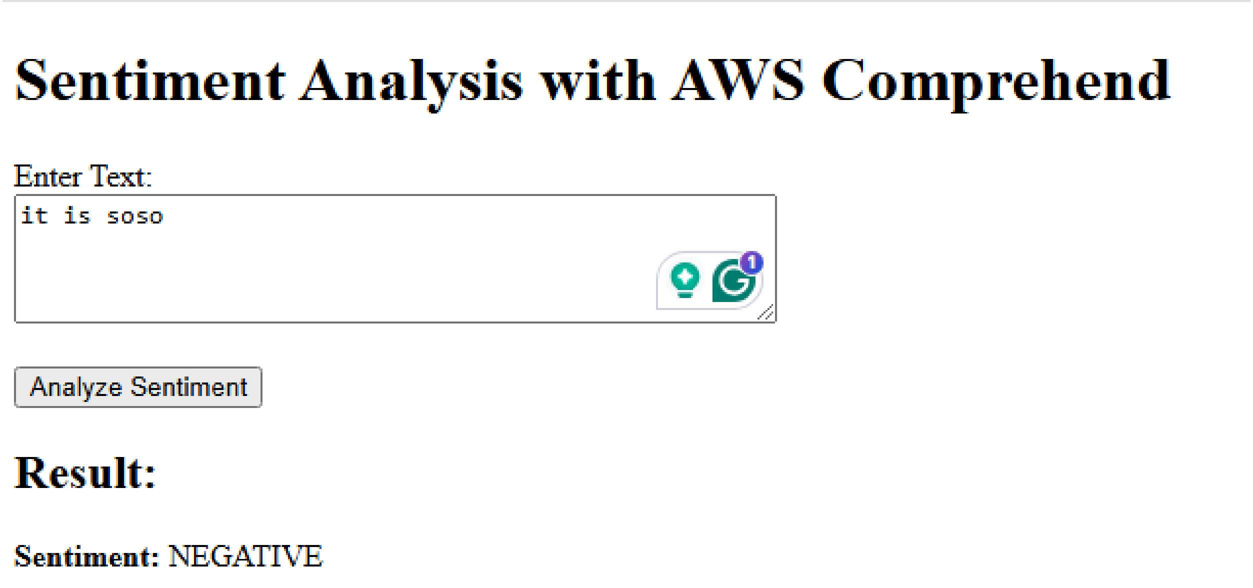
Enter Text:

Result:

Sentiment: NEGATIVE

The text box will clear and the sentiment will update at the bottom.

The following are other examples:



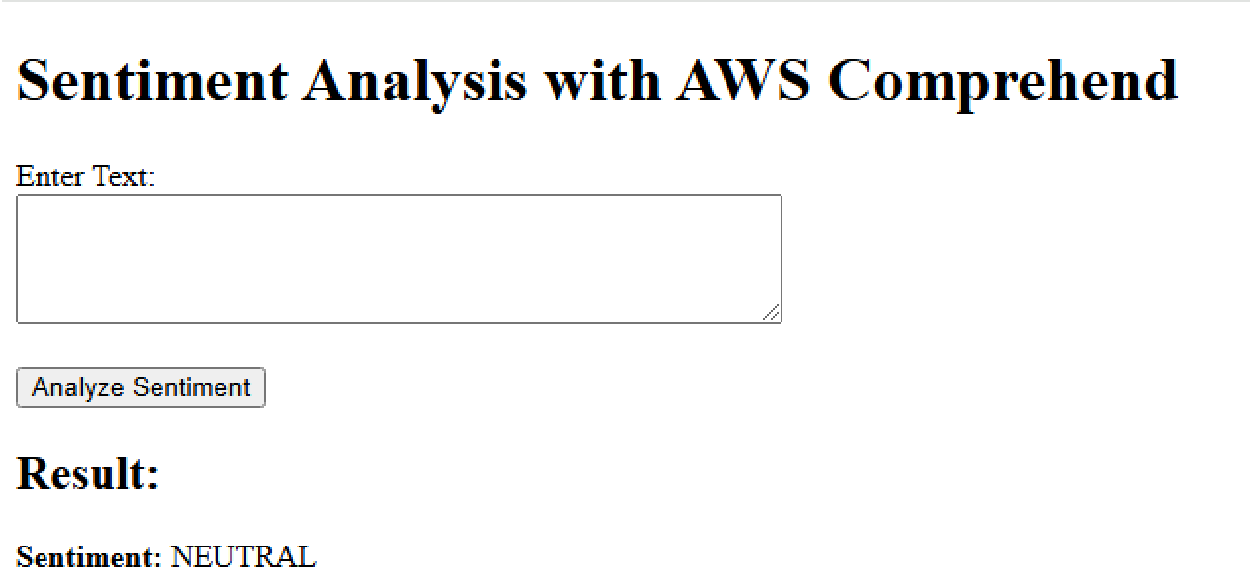
Sentiment Analysis with AWS Comprehend

Enter Text:

```
it is soso
```

Result:

Sentiment: NEGATIVE

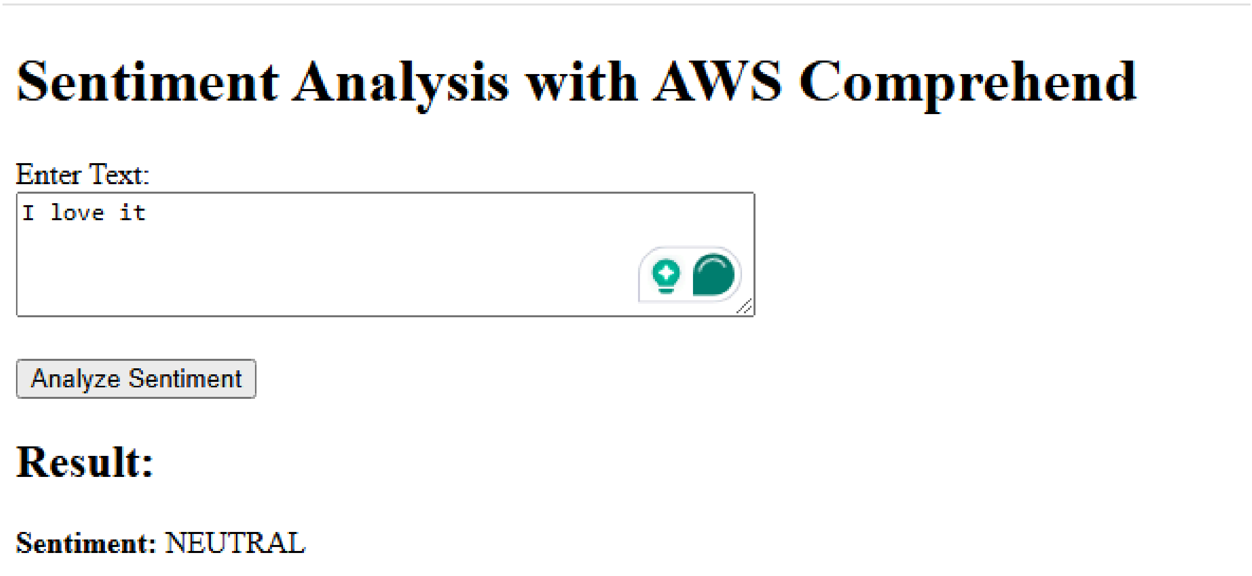


Sentiment Analysis with AWS Comprehend

Enter Text:

Result:

Sentiment: NEUTRAL



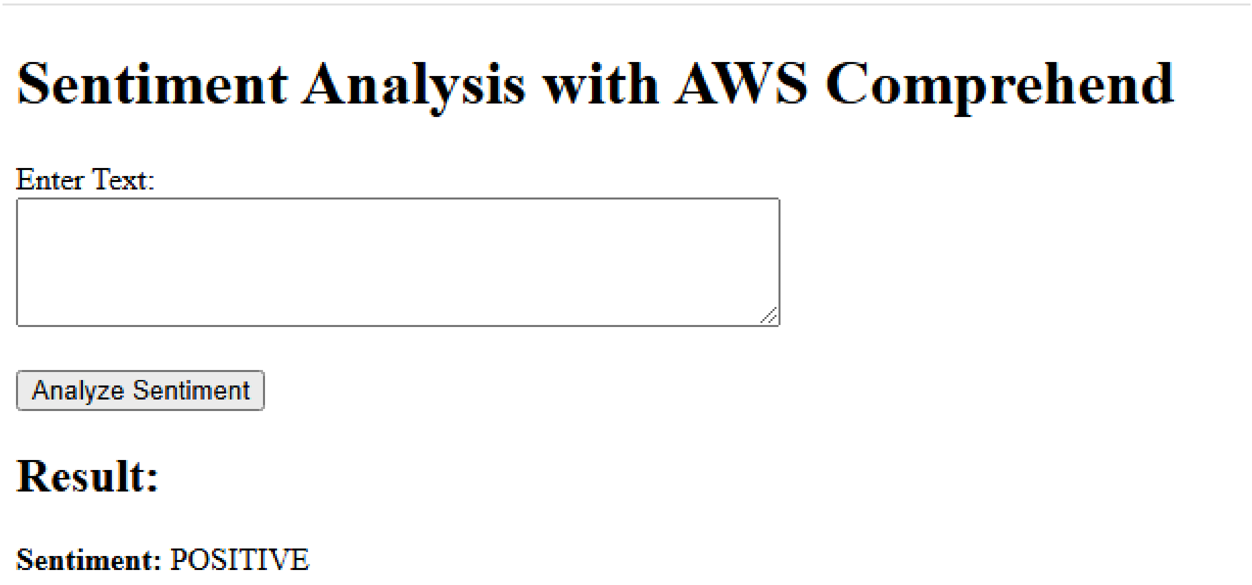
Sentiment Analysis with AWS Comprehend

Enter Text:

```
I love it
```

Result:

Sentiment: NEUTRAL

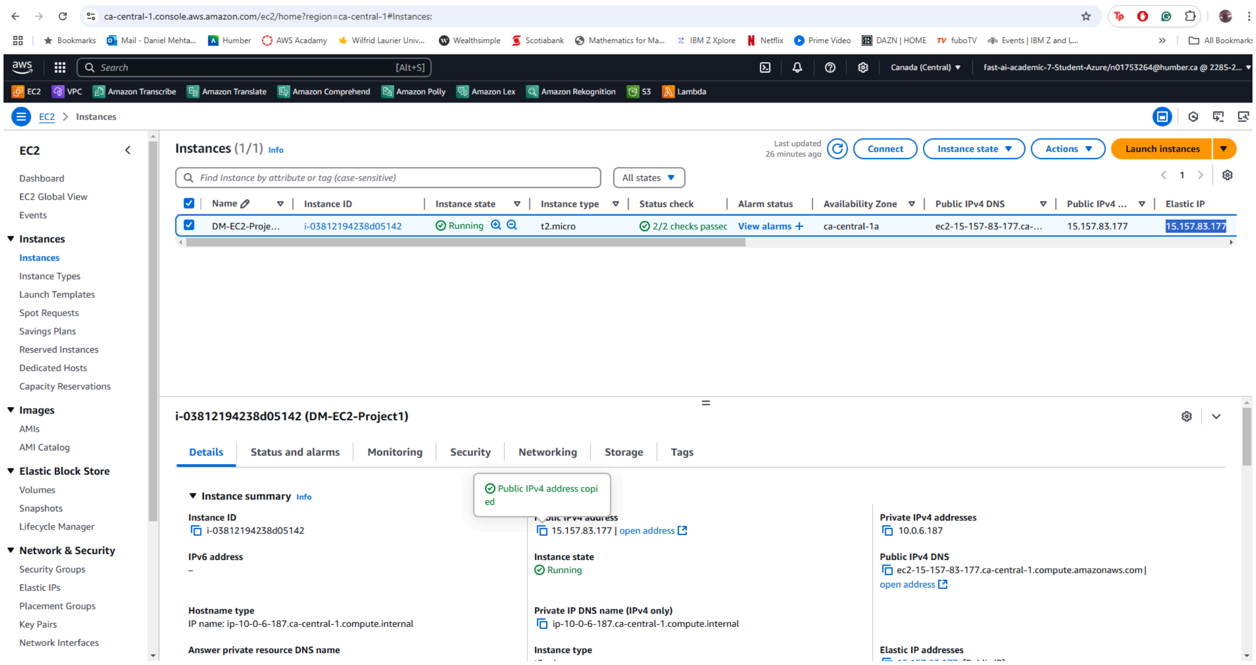


Sentiment Analysis with AWS Comprehend

Enter Text:

Result:

Sentiment: POSITIVE



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	DM-EC2-Proje...	i-03812194238d05142	Running	t2.micro	2/2 checks passec	View alarms +	ca-central-1a	ec2-15-157-83-177.ca-...	15.157.83.177	15.157.83.177

i-03812194238d05142 (DM-EC2-Project1)

Instance summary Info

Public IPv4 address copi ed

Instance ID
i-03812194238d05142

Public IPv4 address
15.157.83.177 | open address

Private IPv4 addresses
10.0.6.187

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-15-157-83-177.ca-central-1.compute.amazonaws.com | open address

Hostname type
IP name: ip-10-0-6-187.ca-central-1.compute.internal

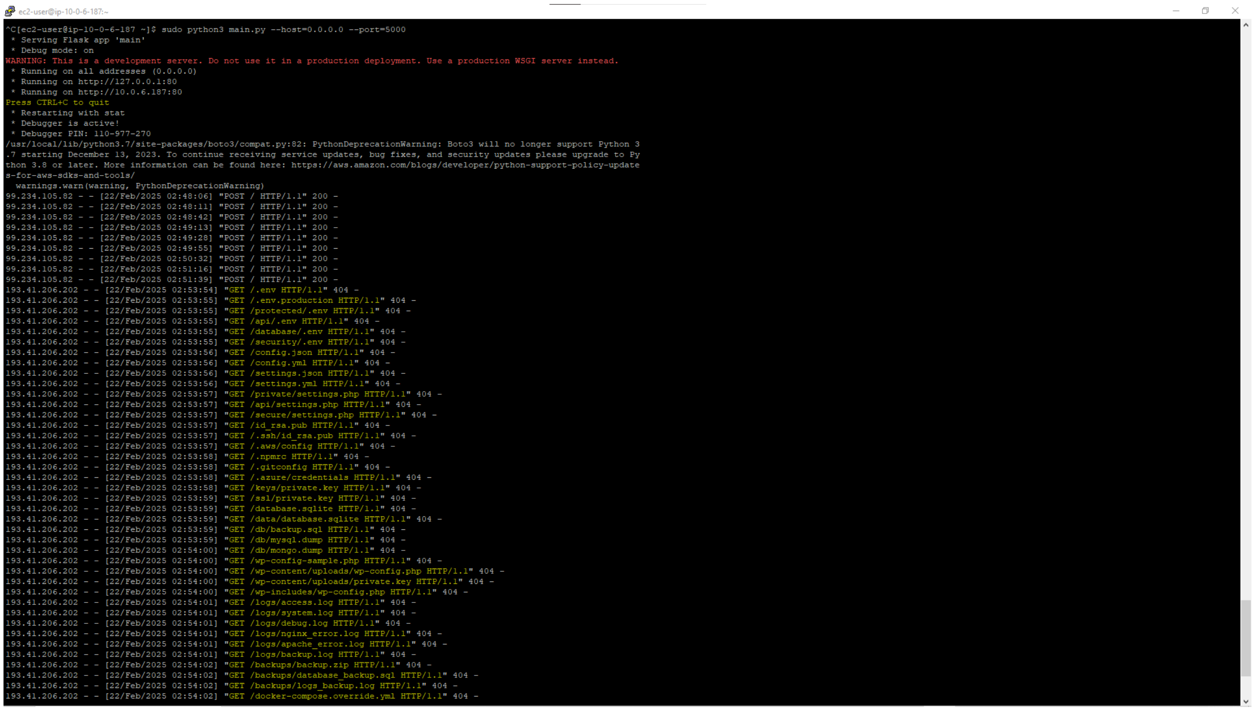
Private IP DNS name (IPv4 only)
ip-10-0-6-187.ca-central-1.compute.internal

Answer private resource DNS name

Instance type

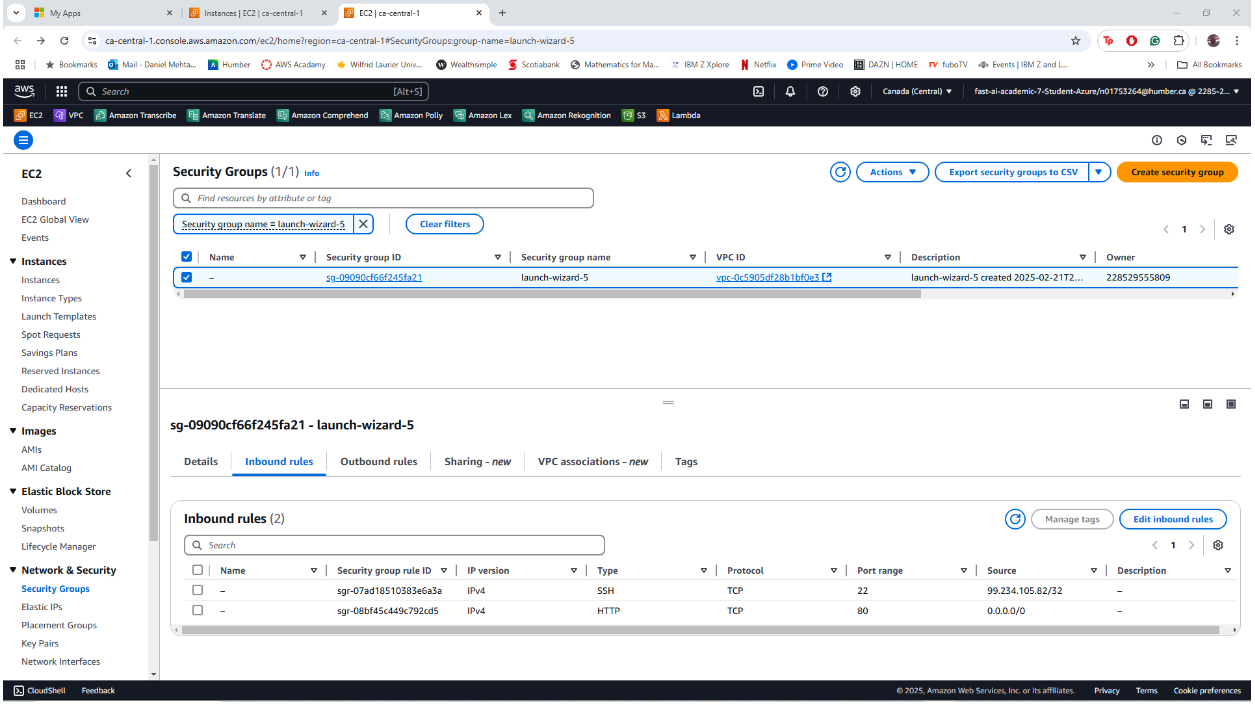
Elastic IP addresses

The EC2 instance



```
^C[ec2-user@ip-10-0-6-187 ~]$ sudo python3 main.py --host=0.0.0.0 --port=5000
 * Serving Flask app 'main'
 * Debug mode: on
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://127.0.0.1:80
 * Running on http://10.0.6.187:80
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 110-977-270
/usr/local/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer support Python 3
.7 starting December 13, 2023. To continue receiving service updates, bug fixes, and security updates please upgrade to Py
thon 3.8 or later. More information can be found here: https://aws.amazon.com/blogs/developer/python-support-policy-update
s-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
99.234.105.82 - - [22/Feb/2025 02:48:06] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:48:11] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:48:42] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:49:13] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:49:28] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:49:55] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:50:32] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:51:16] "POST / HTTP/1.1" 200 -
99.234.105.82 - - [22/Feb/2025 02:51:39] "POST / HTTP/1.1" 200 -
193.41.206.202 - - [22/Feb/2025 02:53:54] "GET /.env HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:55] "GET /.env.production HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:55] "GET /protected/.env HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:55] "GET /api/.env HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:55] "GET /database/.env HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:55] "GET /security/.env HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:56] "GET /config.json HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:56] "GET /config.yml HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:56] "GET /settings.json HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:56] "GET /settings.yml HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:57] "GET /private/settings.php HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:57] "GET /api/settings.php HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:57] "GET /secure/settings.php HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:57] "GET /id_rsa.pub HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:57] "GET /.ssh/id_rsa.pub HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:57] "GET /.aws/config HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:58] "GET /.npmrc HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:58] "GET /.gitconfig HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:58] "GET /.azure/credentials HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:58] "GET /keys/private.key HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:59] "GET /ssl/private.key HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:59] "GET /database.sqlite HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:59] "GET /data/database.sqlite HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:59] "GET /db/backup.sql HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:53:59] "GET /db/mysql.dump HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:00] "GET /db/mongo.dump HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:00] "GET /wp-config-sample.php HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:00] "GET /wp-content/uploads/wp-config.php HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:00] "GET /wp-content/uploads/private.key HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:00] "GET /wp-includes/wp-config.php HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:01] "GET /logs/access.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:01] "GET /logs/system.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:01] "GET /logs/debug.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:01] "GET /logs/nginx_error.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:01] "GET /logs/apache_error.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:01] "GET /logs/backup.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:02] "GET /backups/backup.zip HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:02] "GET /backups/database_backup.sql HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:02] "GET /backups/logs_backup.log HTTP/1.1" 404 -
193.41.206.202 - - [22/Feb/2025 02:54:02] "GET /docker-compose.override.yml HTTP/1.1" 404 -
```

The SSH while the application is running.



ca-central-1.console.aws.amazon.com/ec2/home?region=ca-central-1#SecurityGroups:group-name=launch-wizard-5

Security Groups (1/1)

Security group name = launch-wizard-5

Name	Security group ID	Security group name	VPC ID	Description	Owner
–	sg-09090cf66f245fa21	launch-wizard-5	vpc-0c5905df28b1bf0e3	launch-wizard-5 created 2025-02-21T2...	228529555809

sg-09090cf66f245fa21 - launch-wizard-5

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
–	sgr-07ad18510383e6a3a	IPv4	SSH	TCP	22	99.234.105.82/32	–
–	sgr-08bf45c449c792cd5	IPv4	HTTP	TCP	80	0.0.0.0/0	–

A security group rule was configured to allow inbound traffic on port 5000, ensuring external users can access the Flask application.