

# InNewsIN: News Summarizer App

## Deploying Python Web App to Aws Ec2 Instance

### Team Cloud 03:

- Tunga Dinesh ( TL )
- K Naveen kumar
- Tambabathina Sai krishna



## App using Python

### **Source Code:**

[https://github.com/Pavan91775/TECHPLEMENT\\_PROJECT\\_InNews\\_Deploying\\_a\\_Python\\_WebApp\\_ON\\_Ec2Instance.git](https://github.com/Pavan91775/TECHPLEMENT_PROJECT_InNews_Deploying_a_Python_WebApp_ON_Ec2Instance.git)

### **To create an Ubuntu EC2 instance in AWS, follow these steps:**

#### **1. Create an AWS account**

- Go to [AWS](#) and create an account.
- Sign in to the AWS Management Console.
- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>

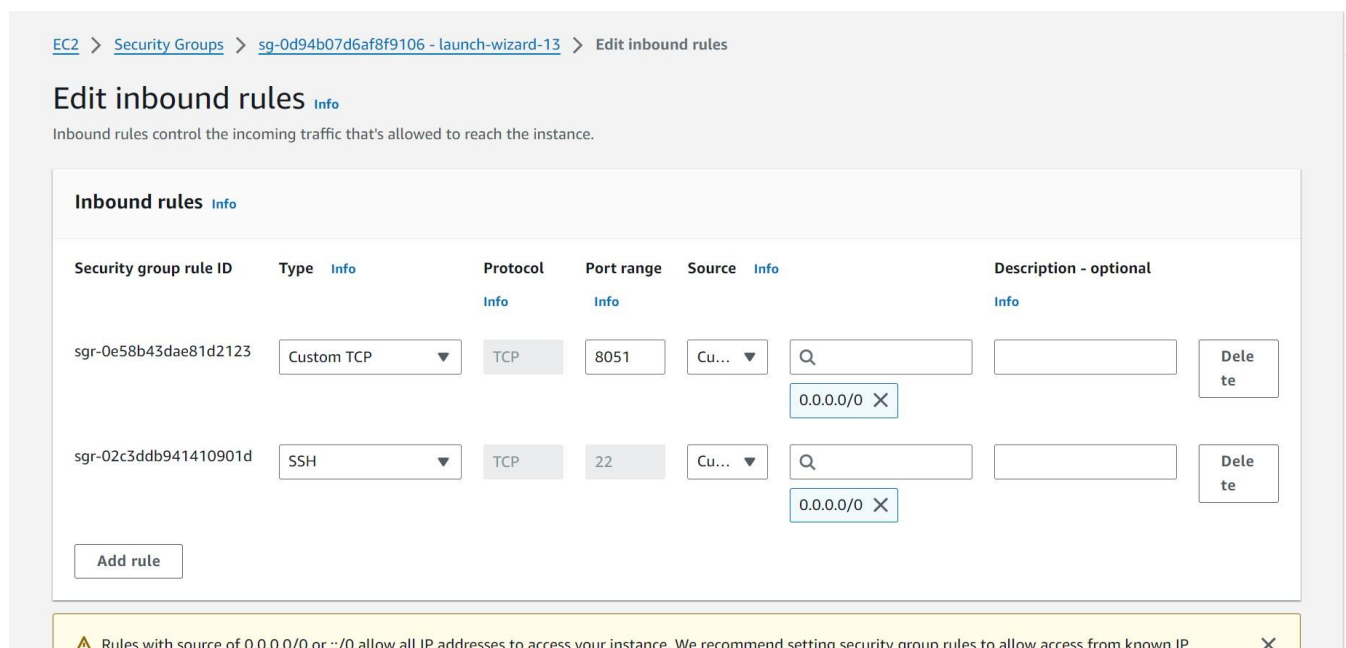
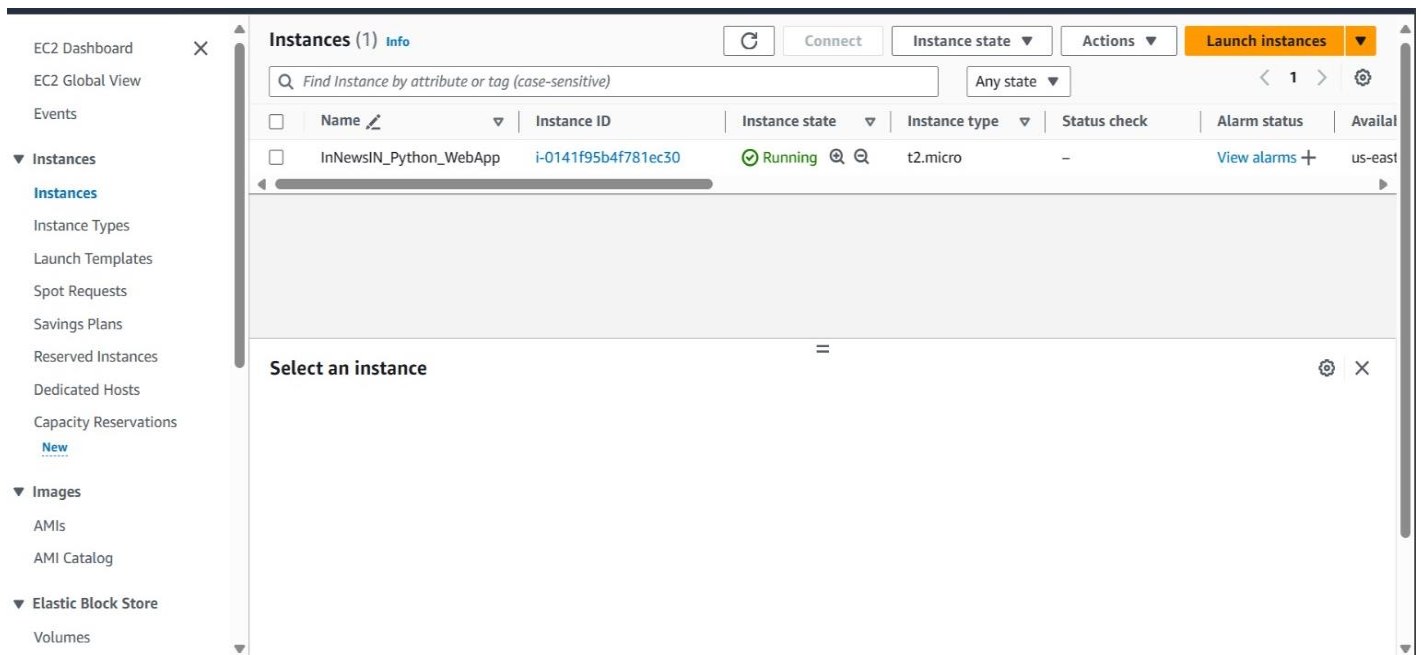
#### **2. Launch an EC2 instance**

- Create an EC2 instance by clicking on the Launch Instance button.
- Choose an Amazon Machine Image (AMI) - Ubuntu Server 20.04 LTS (HVM), SSD Volume Type.
- Choose an Instance Type - t2.micro (Free tier eligible) it may change based on your requirements.
- Configure Instance Details - Keep the default settings.
- Create a and save keypair for ssh access and do not share with anyone.
- Configure Security Group - Create a new security group and add rules to allow ○ HTTP and SSH traffic.
  - Pay attention to create **8501** port for flask and streamlit apps.
  - Add rule to access from anywhere.
- Add Storage - Keep the default settings.

- Add Tags - Keep the default settings.
- Review and Launch - Review the settings and click on the Launch button.

### 3. Connect to your instance

- Go to the EC2 dashboard and click on the Running Instances.
- Select the instance you just created and click on the Connect button.
- Follow the instructions to connect to your instance using SSH.
- Use the following command to connect to your instance: `chmod 400 your-key-pair.pem`  
`ssh -i "your-key-pair.pem" ubuntu@your-instance-public-ip`



### 4. Install the required software

- Update the package list and install the required software:

- `sudo -i` (To convert root user)
- `sudo apt update -y` (To update all packages)

# Install git to clone your app ○ `sudo apt install git` (To install git)

## 5. Clone the repository

- Clone the repository using the following command:
  - `git clone repository-url`
  - `cd (your Repository)`
- Install the required packages using pip:
  - `"python3"` (check python install or not) ○ `"python -m pip"` (check pip install or not)
  - `yum install python3 pip` ○ `python3 -m pip install -r requirements.txt`

## 6. Run the webapp

- Run the webapp using the following command by navigating to the directory where the filename.py file is located:
  - `python3 -m streamlit run filename.py`

## 7. Access the webapp

- Open a web browser and go to ``http://your-instance-public-ip:8501`` to access the webapp.

## 8. Run the webapp in the background

- Use the following command to run the webapp in the background:
  - `nohup python3 -m streamlit run filename.py`

## 9. Stop the webapp

- Use the following command to stop the webapp:
  - `ps -ef` (Get process ID from here) ○ `kill [process-id]`

## 10. Access the webapp using a specific weblink other than IP

- You can use a domain name to access the webapp by setting up a domain name and pointing it to the public IP address of your instance. You can use services like Route 53 to set up a domain name and associate it with your instance.
- You can also use a service like ngrok to create a secure tunnel to your instance and access the webapp using a specific weblink.
- **Note:** Make sure to secure your webapp by setting up SSL/TLS certificates and using HTTPS to encrypt the data transmitted between the client and the server.

- **Note:** Make sure to secure your instance by setting up a firewall, using strong passwords, and keeping the software up to date.
- **Note:** Make sure to monitor your instance and set up alerts to be notified of any issues or unusual activity.
- **Note:** Make sure to back up your data and set up automated backups to prevent data loss.
- **Note:** Make sure to follow best practices for security, performance, and cost optimization when deploying webapps on AWS.

## 11. Terminate the instance and save money

- Go to the EC2 dashboard and click on the Running Instances.
- Select the instance you want to terminate and click on the Actions button.
- Click on the Instance State option and then click on the Terminate option.
- Confirm that you want to terminate the instance.
- **Note:** Terminating an instance will delete all the data on the instance, so make sure to back up any data you want to keep.

## Output:

