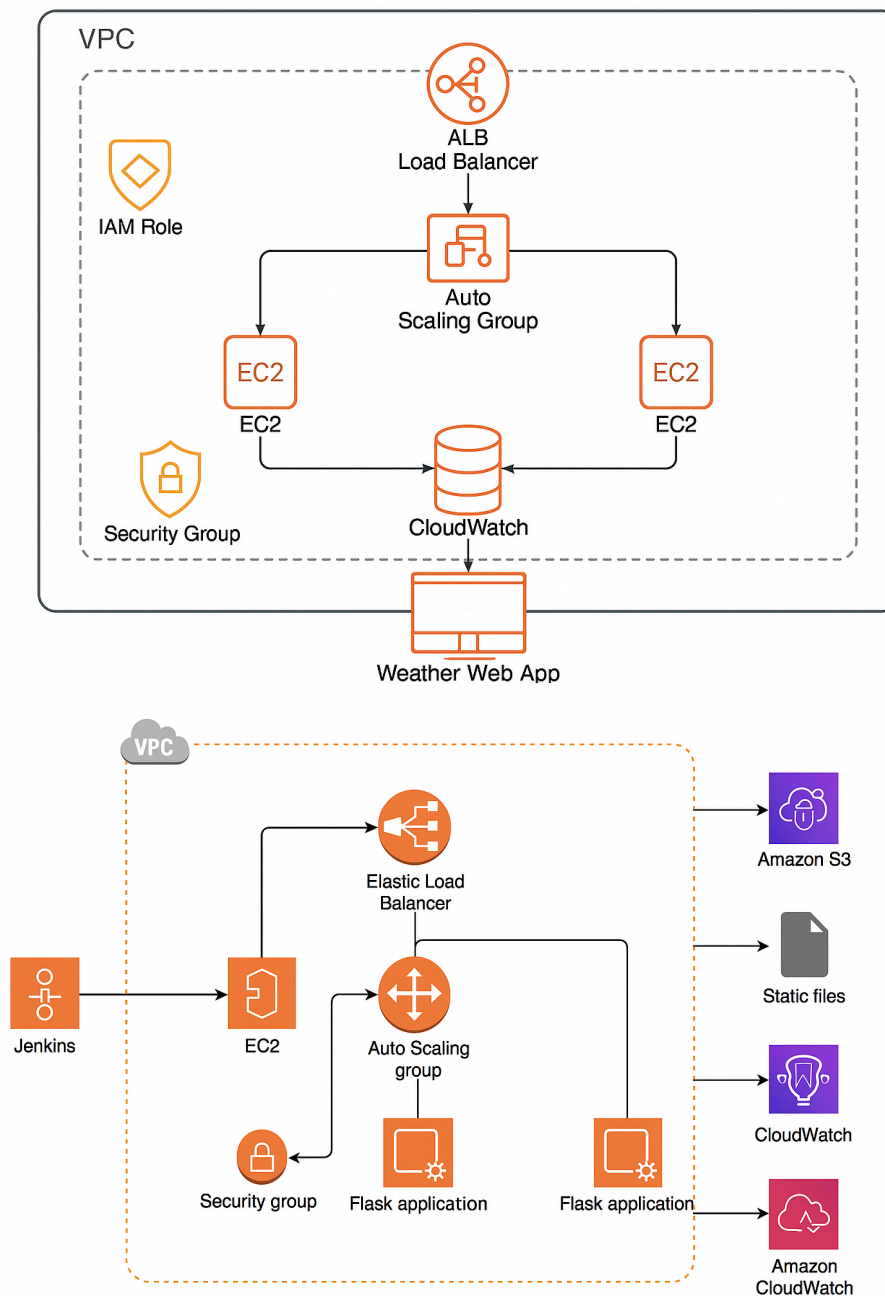


# Cloud-Based Web App Deployment & Resilience Simulation

## Cloud Infrastructure Project - Angeline CSE 'A'

### 1. Architecture Diagram

A visual representation of the deployed architecture, including:



## 2. Code & Scripts

### Terraform Scripts

- `main.tf`: Defines EC2 instance, security group, IAM role, and user data.

```
provider "aws" {
  region = "eu-north-1"
}

resource "aws_instance" "weather_ec2" {
  ami           = "ami-006b4a3ad5f56fbd6"
  instance_type = "t3.micro"
  key_name      = "WeatherAppKey"
  vpc_security_group_ids = ["sg-09328a13b7916ba38"]
  subnet_id    = "subnet-05f4980e60d37121e"
  tags = {
    Name = "WEATHERMAN"
  }

  user_data = <<-EOF

    yum update -y
    yum install -y python3 git
    cd /home/ec2-user
    git clone https://github.com/23f-3004447/WeatherApp.git
    cd WeatherApp
    pip3 install -r requirements.txt
    nohup python3 app.py > flask.log 2>&1 &
    EOF
}
```

## Jenkinsfile

- Pipeline script to automate deployment using CI/CD.

```
pipeline {
  agent any

  stages {
    stage('Clone Repo') {
      steps {
        git branch: 'main',
            url: 'https://github.com/23f-3004447/WeatherApp.git',
            credentialsId: 'a5308c41-1e1b-4839-af30-fe85ccdef2fa'
      }
    }

    stage('Install Dependencies') {
      steps {
        sh 'pip3 install -r requirements.txt'
      }
    }

    stage('Run Flask App') {
      steps {
        sh '''
            fuser -k 5000/tcp || true
            nohup python3 app.py > flask.log 2>&1 &
          '''
      }
    }
  }
}
```

**GitHub Repository:** <https://github.com/23f-3004447/WeatherApp.git>

---

### 3. Screenshots

- IAM Role: `EC2WebAppRole` with permissions for EC2, CloudWatch.

EC2WebAppRole

Info

Delete

Allows EC2 instances to call AWS services on your behalf.

Summary

Edit

Creation date

May 28, 2025, 19:36 (UTC+05:30)

ARN

arn:aws:iam::789665426725:role/EC2WebAppRole

Instance profile ARN

arn:aws:iam::789665426725:instance-profile/EC2WebAppRole

Last activity

4 hours ago

Maximum session duration

1 hour

Permissions

Trust relationships

Tags

Last Accessed

Revoke sessions

Permissions policies (7)

Info

Simulate

Remove

Add permissions

You can attach up to 10 managed policies.

Search

Filter by Type

All types

< 1 >

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	<div><div></div>AmazonEC2FullAccess</div>	AWS managed	1
<input type="checkbox"/>	<div><div></div>AmazonEC2FullAccess1</div>	Customer inline	0
<input type="checkbox"/>	<div><div></div>AmazonEC2ReadOnlyAccess</div>	AWS managed	1
<input type="checkbox"/>	<div><div></div>AmazonS3ReadOnlyAccess</div>	AWS managed	1
<input type="checkbox"/>	<div><div></div>CloudWatchAgentAdminPolicy</div>	AWS managed	1
<input type="checkbox"/>	<div><div></div>CloudWatchAgentServerPolicy</div>	AWS managed	1
<input type="checkbox"/>	<div><div></div>logpolicy</div>	Customer inline	0

- Security Group: `sg-09328a13b7916ba38` with necessary inbound/outbound rules.

sg-09328a13b7916ba38 - launch-wizard-2

Actions

Details

Security group name

launch-wizard-2

Security group ID

sg-09328a13b7916ba38

Description

launch-wizard-2 created 2025-05-28T13:43:25.486Z

VPC ID

yvc-04ecf5575e409dbcd

Owner

789665426725

Inbound rules count

6 Permission entries

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (6)

Manage tags

Edit inbound rules

Search

< 1 >

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
<input type="checkbox"/>	-	sgr-076189450118f2441	IPv4	Custom TCP	TCP	8080	0.0.0.0/0
<input type="checkbox"/>	-	sgr-03cf926a013de57db	IPv4	HTTPS	TCP	443	0.0.0.0/0
<input type="checkbox"/>	-	sgr-017babbded393a213	IPv4	Custom TCP	TCP	5000	0.0.0.0/0
<input type="checkbox"/>	-	sgr-00301a39194042e9e	IPv4	SSH	TCP	22	0.0.0.0/0
<input type="checkbox"/>	-	sgr-0b52c927632496a98	IPv4	Custom TCP	TCP	8000	0.0.0.0/0
<input type="checkbox"/>	-	sgr-0768854c9982f5ed9	IPv4	HTTP	TCP	80	0.0.0.0/0

- EC2 Instance Tags: `Name=WeatherAppServer, Role=WebApp`
- CloudWatch Agent Configuration & Logs

Instances (1/5) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input checked="" type="checkbox"/>	WEATHERMAN	i-03311636aef4fa7a4	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-16-171-232-70.eu-...	16.171.2
<input type="checkbox"/>	WEATHERMAN	i-0ebf1d08f4c89b9a2	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-51-21-255-217.eu-...	51.21.25
<input type="checkbox"/>	Webserver	i-0e8d728d7b2483ebb	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1b	ec2-51-21-180-223.eu-...	51.21.18
<input type="checkbox"/>	WeatherAppS...	i-046300da4754b3101	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1b	ec2-16-171-176-218.eu...	16.171.1

i-03311636aef4fa7a4 (WEATHERMAN)

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-23-118.eu-north-1.compute.internal

Instance type

t3.micro

VPC ID

vpc-04ecf5575e409dbcd

Subnet ID

subnet-05f4980e60d37121e

Public DNS

ec2-16-171-232-70.eu-north-1.compute.amazonaws.com | open address

Elastic IP addresses

AWS Compute Optimizer finding

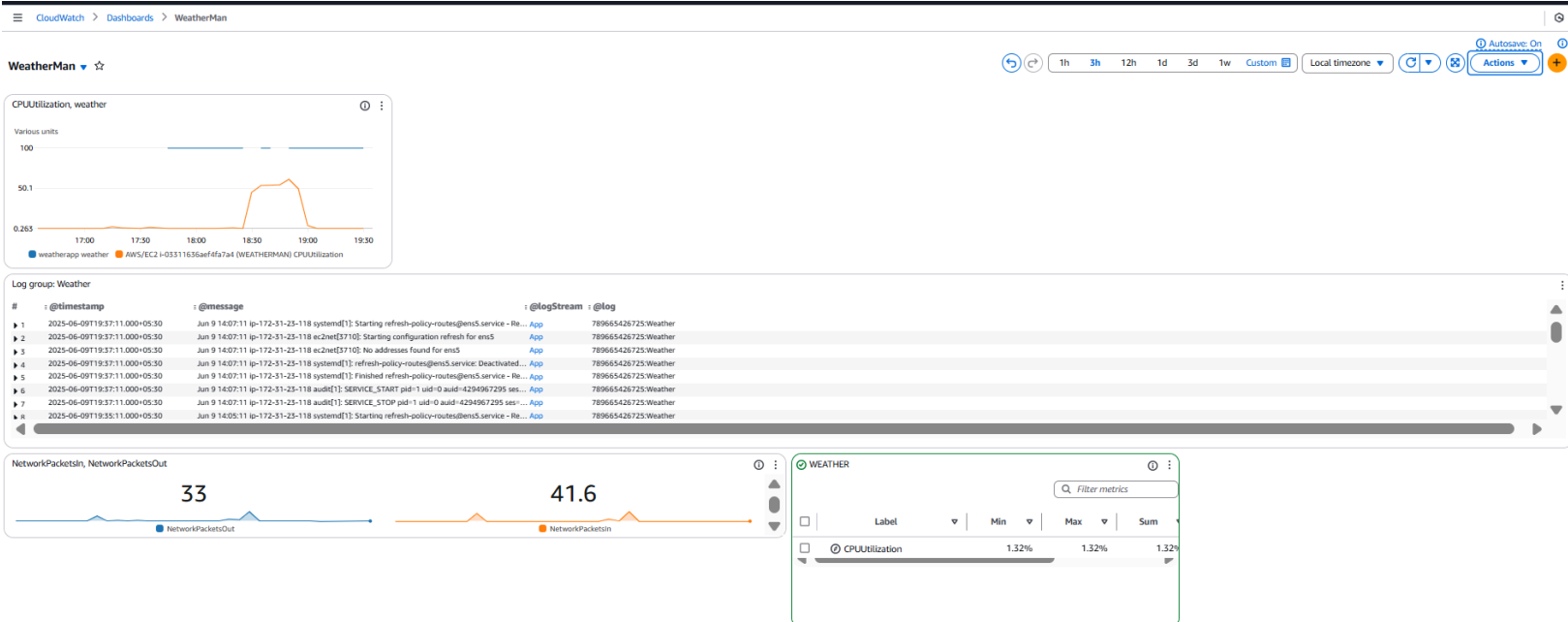
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name

WeatherAppASG

## 4. Monitoring Dashboard Snapshots

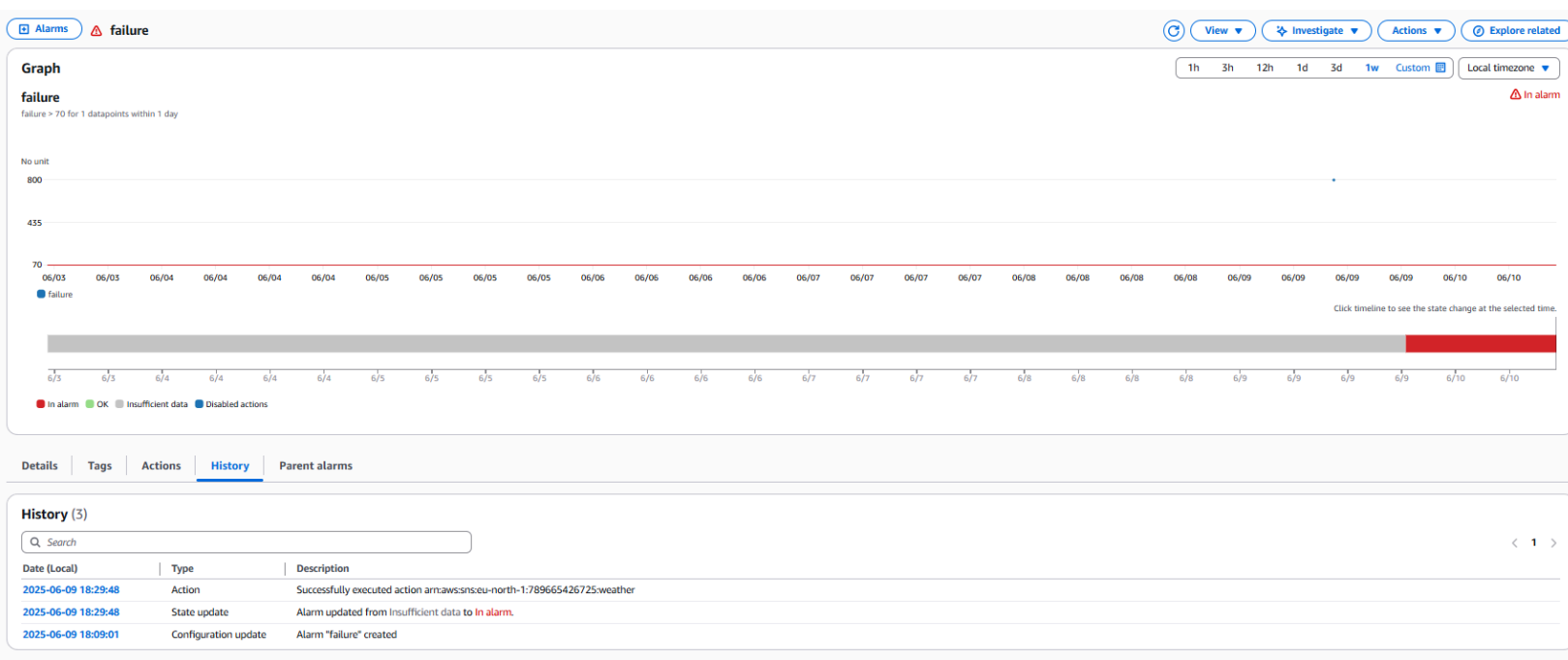
- **Service:** Amazon CloudWatch
- Dashboards monitoring:
  - CPU Utilization
  - Network In/Out
  - Custom Logs (e.g., application logs)



## 5. Simulation Report

### System Failure – Custom Alarm (**failure**)

- **Alarm Name:** `failure`
- **Metric Tracked:** Custom metric – `failure`
- **Condition:** `failure > 70` for 1 datapoint within 1 day
- **State:** In alarm
- **Trigger Time:** `2025-06-09 18:29:48` (Local Time)
- **Alarm History:**
  - `2025-06-09 18:09:01` – Alarm created.
  - `2025-06-09 18:29:48` – Alarm transitioned to In alarm.
  - SNS Action executed:  
`arn:aws:sns:eu-north-1:789665426725:weather.`




- **Response:**
  - Notification sent via SNS to the monitoring topic.

- No automatic remediation was configured for this custom metric.
- Manual inspection initiated.
- Admin verified application logs for anomalies.
- Instance showed abnormally high values; restarted Flask service.

---

## 6. Final Presentation / Video

 Capstone.mp4

<http://weatherman-alb-787651582.eu-north-1.elb.amazonaws.com/>

---

## 7. Conclusion

This project showcases end-to-end deployment of a Flask app using Terraform, Jenkins, and AWS services.

It demonstrates cloud-native practices like infrastructure as code, scalable architecture, proper security, real-time monitoring, CI/CD automation, and a successful resilience simulation.