









Docker Compose and AWS SES - Assignment



AWS EC2 Setup

Create EC2 instance (t3.medium)


Instance summary for i-07693ecb6e7024f5b (ws3) [Info](#)
Updated less than a minute ago

Instance ID  i-07693ecb6e7024f5b	Public IPv4 address  54.210.76.154 open address 
IPv6 address -	Instance state  Running
Hostname type IP name: ip-172-31-24-96.ec2.internal	Private IP DNS name (IPv4 only)  ip-172-31-24-96.ec2.internal
Answer private resource DNS name IPv4 (A)	Instance type t3.medium
Auto-assigned IP address  54.210.76.154 [Public IP]	VPC ID  vpc-04225d76b207fbd2d 

Configure SES



Identities
The **Identities** pane lists your domains, subdomains, and email address identities. All identities must be verified before you use them to send email in Amazon SES. [Learn more](#) . To impact email authentication issues found for the identities you select and check for recommendations. [Learn more](#) 

Identities (2) [Info](#)

Last updated 30 minutes ago 

[Check for recommendations](#)

[Send test email](#)

<input type="checkbox"/>	Identity	▼	Identity type	▼	Identity status
<input type="checkbox"/>	rajeshchandran007@gmail.com		Email address		 Verified
<input type="checkbox"/>	rajeshchandranaaws3@gmail.com		Email address		 Verified

Create IAM Role

ses-role-for-ec2 [Info](#)

Allows EC2 instances to call AWS services on your behalf.

Summary


Creation date

September 17, 2025, 11:12 (UTC+05:30)

Last activity

-

ARN

 arn:aws:iam::307946636515:role/ses-role-for-ec2

Maximum session duration

1 hour

[Permissions](#)

[Trust relationships](#)

[Tags](#)

[Last Accessed](#)

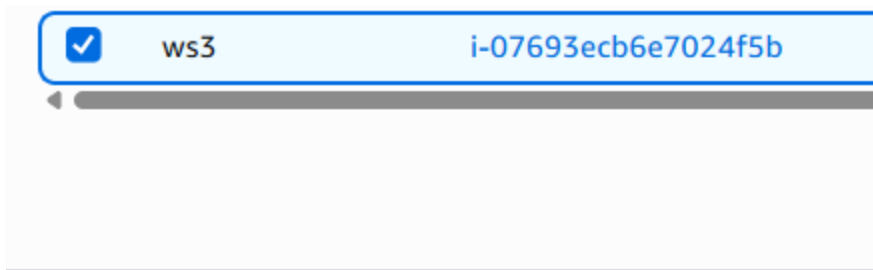
[Revoke sessions](#)

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type		
<div><div><div><div></div><div>Search</div></div></div></div> <div>All types</div>		
<input type="checkbox"/>	Policy name ↗	▲ Type
<input type="checkbox"/>	  AmazonSESFullAccess	AWS managed

Attach Role to EC2



i-07693ecb6e7024f5b (ws3)

Details

Status and alarms

Monitoring

▼ Security details

IAM Role

 ses-role-for-ec2 

Source Code Setup

Clone the code

Git clone <https://github.com/rajeshchandranaws3/bootcamp-august-rajesh.git>

Amend the code

Update .env file

```
# Email Configuration
SENDER_EMAIL=rajeshchandranaws3@gmail.com
RECIPIENT_EMAILS=rajeshchandran007@gmail.com
```

Remove AWS Keys from the environment variables in the docker-compose

```

alert-service:
  build:
    context: alert-service
    dockerfile: Dockerfile.alert
  container_name: alert-service
  volumes:
    - ./logs:/var/log:ro # Read-only access to log files
  environment:
    - CONTAINER_NAME=flask-app
    - ALERT_LOG=/var/log/container_alerts.log
    - AWS_REGION=${AWS_REGION:-us-east-1}
    - SENDER_EMAIL=${SENDER_EMAIL}
    - RECIPIENT_EMAILS=${RECIPIENT_EMAILS}
    - CHECK_INTERVAL=${CHECK_INTERVAL:-30}
    - ALERT_COOLDOWN=${ALERT_COOLDOWN:-300}
    - BUFFER_TIMEOUT=${BUFFER_TIMEOUT:-60}
  depends_on:
    - monitor
  restart: unless-stopped
  networks:
    - app-network

```

Remove AWS Keys from the boto3 ses client api call in the alert_service.py

```

class AlertService:
    def __init__(self):
        # AWS SES Configuration
        self.ses_client = boto3.client(
            'ses',
            region_name=os.getenv('AWS_REGION', 'us-east-1'),
        )

```

Replace "required_vars" in alert_service.py

from:

```

required_vars = ['AWS_ACCESS_KEY_ID', 'AWS_SECRET_ACCESS_KEY',
'SENDER_EMAIL', 'RECIPIENT_EMAILS']

```

to:

```

required_vars = ['SENDER_EMAIL', 'RECIPIENT_EMAILS']

```

```

if __name__ == "__main__":
    # Check required environment variables
    required_vars = ['SENDER_EMAIL', 'RECIPIENT_EMAILS']
    missing_vars = [var for var in required_vars if not os.getenv(var)]

```

Program Execution

docker compose up -d - --build

```

[+] Running 10/10
✔ project_container_monitoring-monitor Built
✔ project_container_monitoring-alert-service Built
✔ project_container_monitoring-webapp Built
✔ project_container_monitoring-stress-generator Built
✔ Network project_container_monitoring_app-network Created
✔ Container db Healthy
✔ Container flask-app Healthy
✔ Container app-monitor Started
✔ Container project_container_monitoring-stress-generator-1 Started
✔ Container alert-service Started

```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
93a9a141f02c		project_container_monitoring-alert-service	"python3 alert_servi..."	About a minute ago	Up 43 seconds (healthy)	
887a9d09c569		project_container_monitoring-monitor	"/app/start_monitor..."	About a minute ago	Up 43 seconds	0.0.0.0:8001->8001/tcp, [::]:8001->8001/tcp
149947e42f06		project_container_monitoring-stress-generator	"python stress_app.py"	About a minute ago	Up 43 seconds	
2771e42ef951		project_container_monitoring-stress-generator-1	"python app.py"	About a minute ago	Up 9 seconds (health: starting)	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
6a61d0a572c		flask-app	"docker-entrypoint.s..."	About a minute ago	Up About a minute (healthy)	0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp
5432->5432/tcp		postgres:15-alpine				
		db				

Program Output

Flask App Output (Port: 8080)

High-Performance Monitored Application

Memory Cache Size 0 entries	Computation Results 6 stored	Background Tasks 3 running	Database Records 97176 total
---------------------------------------	--	--------------------------------------	--

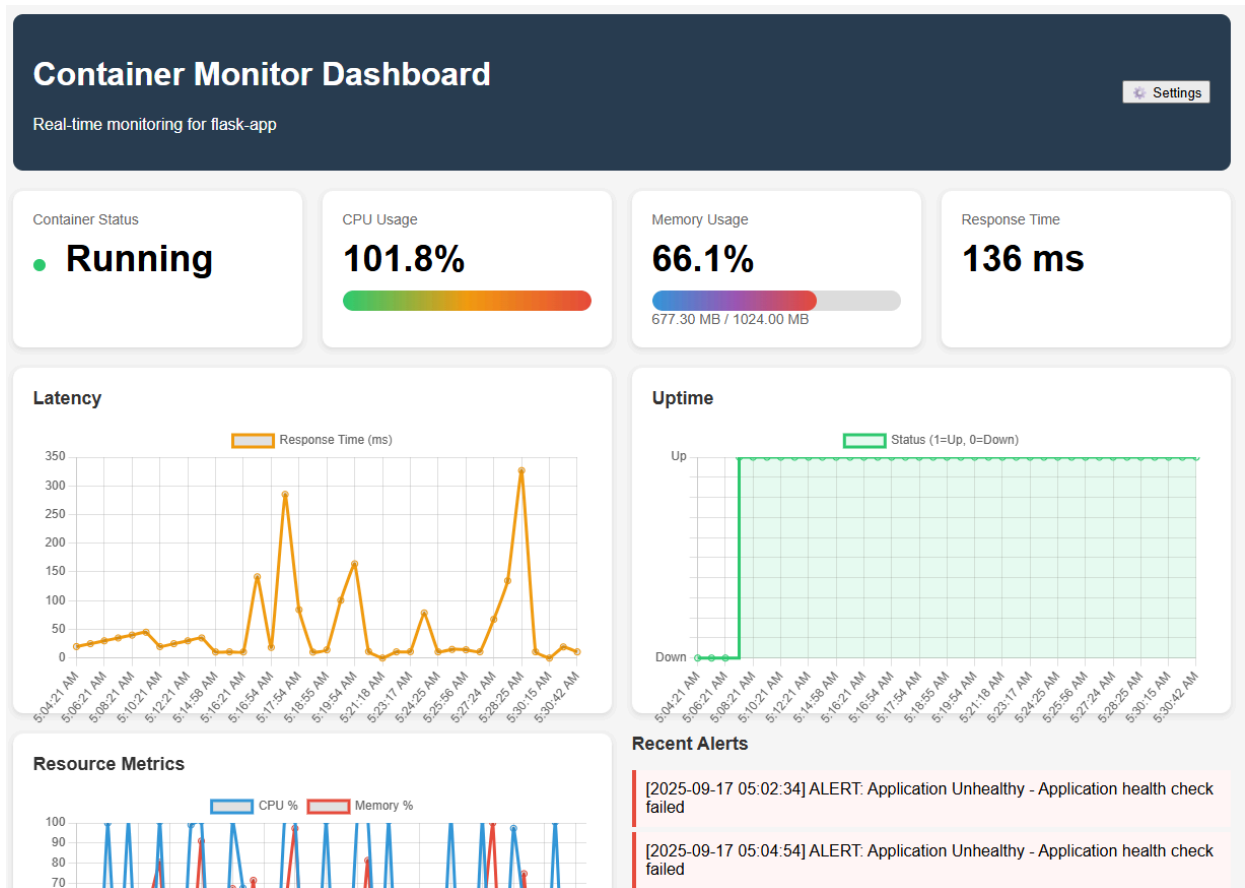
Stress Test Controls

Run CPU Test	Run Memory Test	Run Database Test	Run Combined Test
--------------	-----------------	-------------------	-------------------

Available Endpoints:

- / - Main dashboard
- /health - Health check
- /api/stats - System statistics
- /api/cpu-intensive - CPU intensive task
- /api/memory-intensive - Memory intensive task
- /api/database-intensive - Database intensive task

Monitoring App Output (Port: 8001)



Email Output



rajeshchandranaws3@gmail.com via amazonses.com
to me ▾

Container Monitoring Alert Summary

=====

Container: flask-app

Alerts Detected:

Application Unhealthy (5 occurrences):

- [2025-09-17 05:02:34] Application health check failed
- [2025-09-17 05:04:54] Application health check failed
- [2025-09-17 05:07:14] Application health check failed
- [2025-09-17 05:08:26] Application health check failed
- [2025-09-17 05:11:10] Application health check failed

Container Down (1 occurrences):

- [2025-09-17 05:07:46] Container flask-app is not running

Action Required:

Please check the container status and take appropriate action.

Dashboard: <http://localhost:8000>

Application: <http://localhost:8080>