# 🚀 Secure Service-to-Service Access in AWS: Securely Logging EC2 Web Application Logs to CloudWatch Using IAM Roles

## ◇ Overview

In this project, I implemented **secure service-to-service access** in AWS by allowing an **EC2 instance to send application logs to CloudWatch**. This is achieved using **IAM roles**, ensuring that the access is granted securely without hardcoding credentials.

Here's how I set up the **IAM role, attached it to an EC2 instance, and configured CloudWatch Agent** to capture logs from a web application.

## 1️⃣ Create an IAM Role for EC2

### ◇ Why is this needed?

By default, an EC2 instance does not have permission to send logs to CloudWatch. Instead of storing credentials on the instance, we use an **IAM role** to grant temporary, secure access to AWS services.

### ◇ Steps to create the IAM role:

**1️⃣ Navigate to the AWS IAM Console**

- Sign in to the **AWS Management Console**.
- Open the **IAM** service.
- Go to **Roles** → Click **Create Role**.

**2️⃣ Select a Trusted Entity**

- Choose **AWS Service** as the trusted entity type.
- Under **Use case**, select **EC2**.

LinkedIn :- https://www.linkedin.com/in/vaibhav-chaudhari-14016b22a/

### 3️⃣ Attach the Required Permissions

- In the **Permissions** section, search for **CloudWatchLogsFullAccess**.
- Select it to allow the EC2 instance to write logs to CloudWatch.

### 4️⃣ Name and Create the Role

- Provide a meaningful name, e.g., **CloudWatchLogAccess4EC2**.
- Click **Create Role**.

🌟 **At this point, we have successfully created an IAM role that allows EC2 instances to interact with CloudWatch Logs.**

## 2️⃣ Attach the IAM Role to an EC2 Instance

### 🔹 Why is this needed?

Now that we have created the IAM role, we need to attach it to an existing EC2 instance so it can assume the role and gain the required permissions.

### 🔹 Steps to attach the IAM role to EC2:

### 1️⃣ Navigate to the EC2 Console

- Open the **AWS EC2 Console**.
- Select the EC2 instance that needs access to CloudWatch Logs.

### 2️⃣ Modify the Instance IAM Role

- Click **Actions** → **Security** → **Modify IAM Role**.
- In the **IAM Role** dropdown, select **CloudWatchLogAccess4EC2**.
- Click **Update IAM Role** to apply the changes.

🌟 **Now, the EC2 instance has permission to push logs to CloudWatch securely!**

# 3️⃣ Deploy a Web Application on EC2

We will deploy a simple **Apache Web Server** on our EC2 instance and set up logging for it.

## 🔹 User Data for EC2 Instance (Advanced Configuration)

When launching an EC2 instance, use the following **User Data script** to automatically install and start Apache:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "Hello World from $(hostname -f)" > /var/www/html/index.html
```

🌟 **This script ensures that a basic web server is running on the instance upon startup.**

# 4️⃣ Install and Configure CloudWatch Agent

## ◇ Why Install CloudWatch Agent?

CloudWatch Agent enables us to **collect system metrics and logs** from the EC2 instance and send them to **AWS CloudWatch** for monitoring.

## ◇ Steps to Install and Configure CloudWatch Agent

### 1️⃣ Connect to the EC2 Instance

- Use SSH to log in: `ssh -i your-key.pem ec2-user@your-ec2-ip`

  Or
- You Can Use EC2 Instance Connect to connect using browser-based Client with a public Ipv4 or IPv6 address from AWS Console.

### 2️⃣ Download CloudWatch Agent Package

```
wget https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
```

### 3️⃣ Install the CloudWatch Agent

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

### 4️⃣ Verify the Installation

```
cd /opt/aws/amazon-cloudwatch-agent/
ls -lstr
```

### 5️⃣ Run the CloudWatch Agent Configuration Wizard

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

This wizard will prompt several configuration questions. Key selections:

✅ **OS:** Linux

✅ **EC2 or On-Premises:** EC2

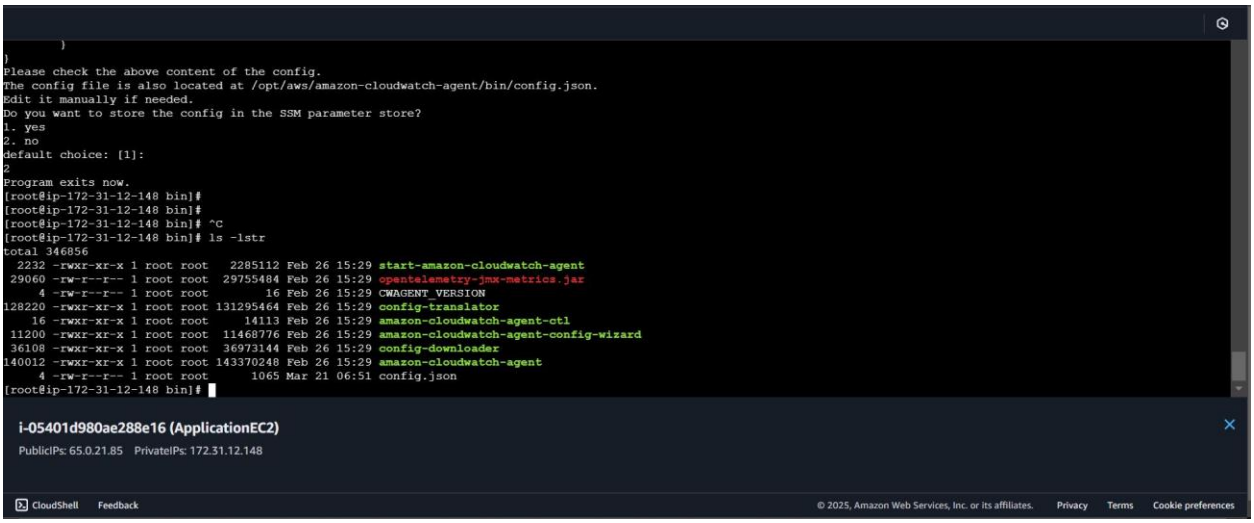
✅ **Enable Log Monitoring:** Yes

✅ **Log File Path:** `/var/log/httpd/access_log`

✅ **Log Group Name:** `MyEC2AccessLog`

✅ **Retention Period:** 1 day

🌟 **Once completed, the wizard generates a `config.json` file for CloudWatch Agent.**




## 5Start CloudWatch Agent and Monitor Logs

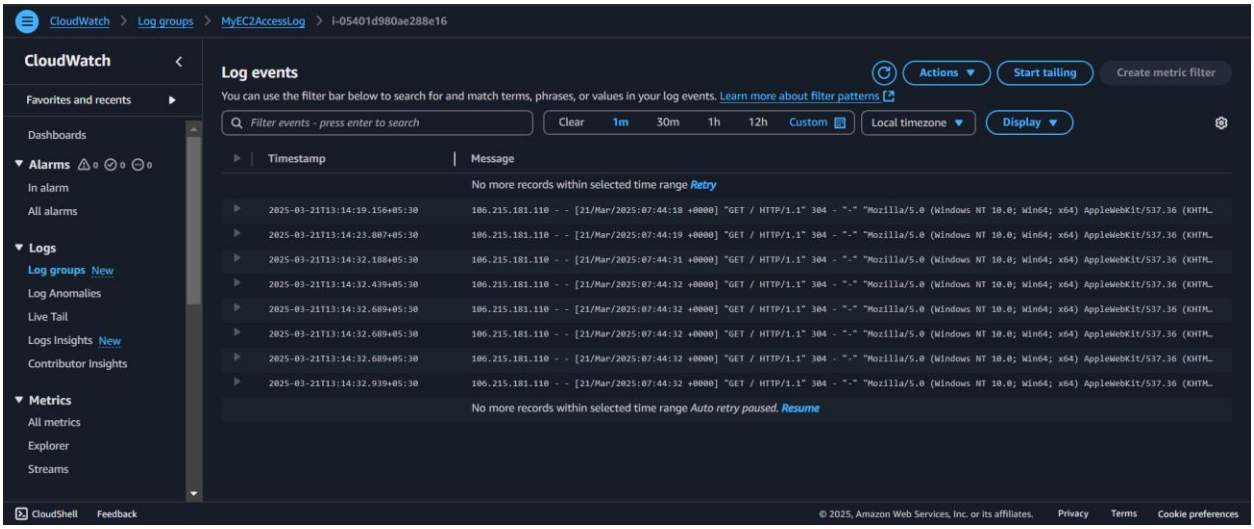
1️⃣ **Start the Agent with Config File**

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl
-a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-
```

```
agent/bin/config.json -s
```

### 2️⃣ Monitor Logs in AWS CloudWatch

- Open **AWS Console** → **CloudWatch**
- Go to **Logs** → **Log Groups** → `MyEC2AccessLog`
- Select **Log Streams** to monitor logs in real-time



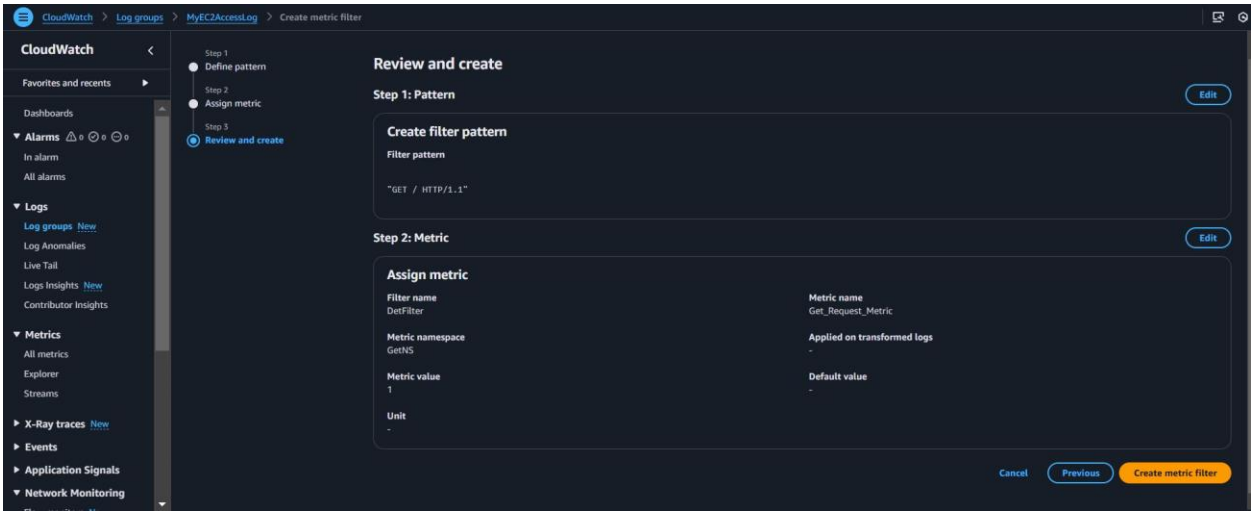

# 6️⃣ Create a CloudWatch Metric Filter

### 1️⃣ Select Log Stream and create a metric filter

- Add Filter Pattern: **"GET / HTTP/1.1"**
- Test with your log stream.
- Name the filter: **DetFilter**

- Set Metric Details:
  - **Namespace:** GetNS
  - **Metric Name:** Get_Request_Metric
  - **Metric Value:** 1

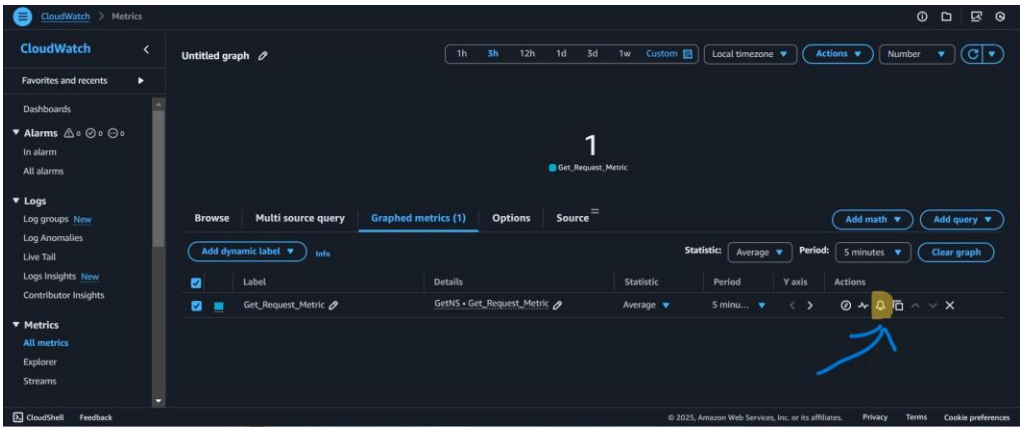
- Click **Create Metric Filter**.




## 2️⃣ Create an Alarm for the Metric

- Open the **Graphed Metric**.

- Set threshold and create an alarm.




🌟 **Now, CloudWatch will track GET requests and alert based on predefined conditions!**

## ◇ Key Takeaways

☑ **IAM Roles ensure secure, service-to-service communication** without hardcoding credentials.

☑ **CloudWatch Logs enable centralized logging**, making troubleshooting easier.

☑ **CloudWatch Agent provides detailed monitoring** for EC2 instances.

☑ **This setup ensures that web application logs are securely streamed to AWS CloudWatch for real-time monitoring.**

This project helped me **strengthen my AWS skills** in security, observability, and automation.
Hope it help you exploring AWS IAM, EC2 and CloudWatch!