

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.

1 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:

1 Navigate to the AWS IAM Console

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

2 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/>

3 Attach the Required Permissions

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

4 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.

2 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:

1 Navigate to the EC2 Console

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

2 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!

3 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.



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

1 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.

2 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
```

3 Install the CloudWatch Agent

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

4 Verify the Installation

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

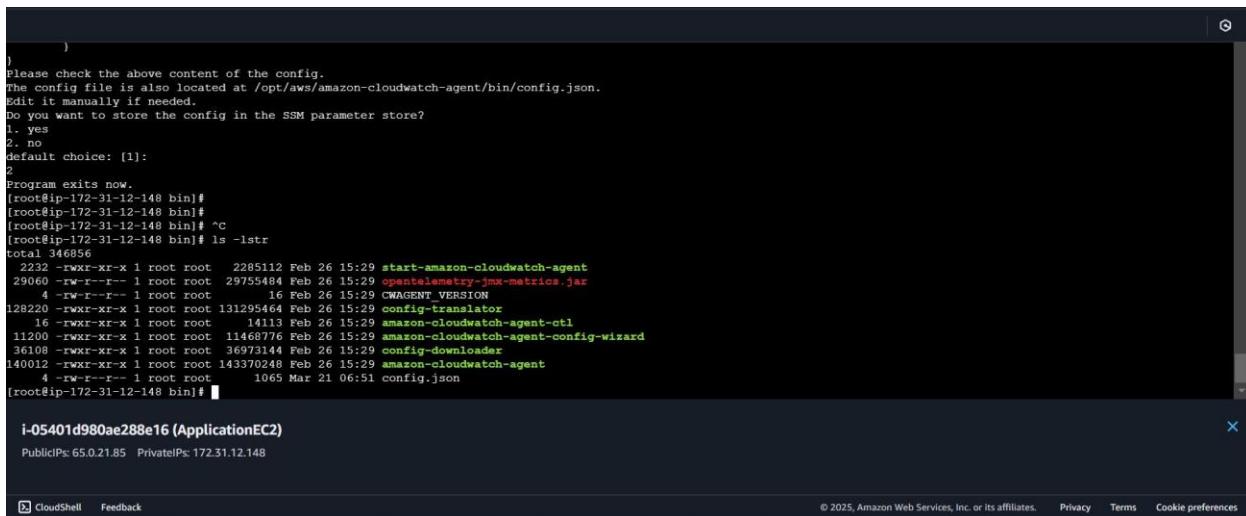
5 Run the CloudWatch Agent Configuration Wizard

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-  
config-wizard  
LinkedIn :- https://www.linkedin.com/in/vaibhav-chaudhari-14016b22a/
```

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.



The screenshot shows a terminal window in AWS CloudShell. The user has completed the CloudWatch Agent configuration wizard. The terminal output includes the configuration JSON file generated by the wizard:

```
        }
Please check the above content of the config.
The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.
Edit it manually if needed.
Do you want to store the config in the SSM parameter store?
1. yes
2. no
default choice: [1]: 2
Program exits now.
[root@ip-172-31-12-148 bin]# ls -lstr
total 346856
2232 -rwxr-xr-x 1 root root 2285112 Feb 26 15:29 start-amazon-cloudwatch-agent
29060 -rw-r--r-- 1 root root 29755484 Feb 26 15:29 opentelemetry-jmx-metrics.jar
128220 -rwxr-xr-x 1 root root 131295464 Feb 26 15:29 CWAGENT_VERSION
11200 -rwxr-xr-x 1 root root 14113 Feb 26 15:29 config-transformer
36108 -rwxr-xr-x 1 root root 11468776 Feb 26 15:29 amazon-cloudwatch-agent-ctl
36108 -rwxr-xr-x 1 root root 36973144 Feb 26 15:29 amazon-cloudwatch-agent-config-wizard
140012 -rwxr-xr-x 1 root root 143370348 Feb 26 15:29 config-downloader
4 -rw-r--r-- 1 root root 1065 Mar 21 06:51 config.json
[root@ip-172-31-12-148 bin]#
```

The terminal also displays the instance ID and public IP address:

i-05401d980ae288e16 (ApplicationEC2)
PublicIPs: 65.0.21.85 PrivateIPs: 172.31.12.148

5 Start CloudWatch Agent and Monitor Logs

1 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-
```

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

agent/bin/config.json -s

2 Monitor Logs in AWS CloudWatch

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

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar navigation includes CloudWatch, Favorites and recents, Dashboards, Alarms, Logs (with Log groups, Log Anomalies, Live Tail, and Logs Insights), and Metrics. The main content area is titled "Log events" and displays a table of log entries. The columns are "Timestamp" and "Message". The table shows several log entries from March 21, 2025, at 14:19:156+05:30, 14:23:887+05:30, 14:32:188+05:30, 14:32:439+05:30, 14:32:689+05:30, 14:32:689+05:30, and 14:32:939+05:30. Each entry includes a timestamp, IP address (106.215.181.110), and a detailed message about a GET request to /HTTP/1.1 from Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36. The interface also features a filter bar at the top with search, time range, and display options.

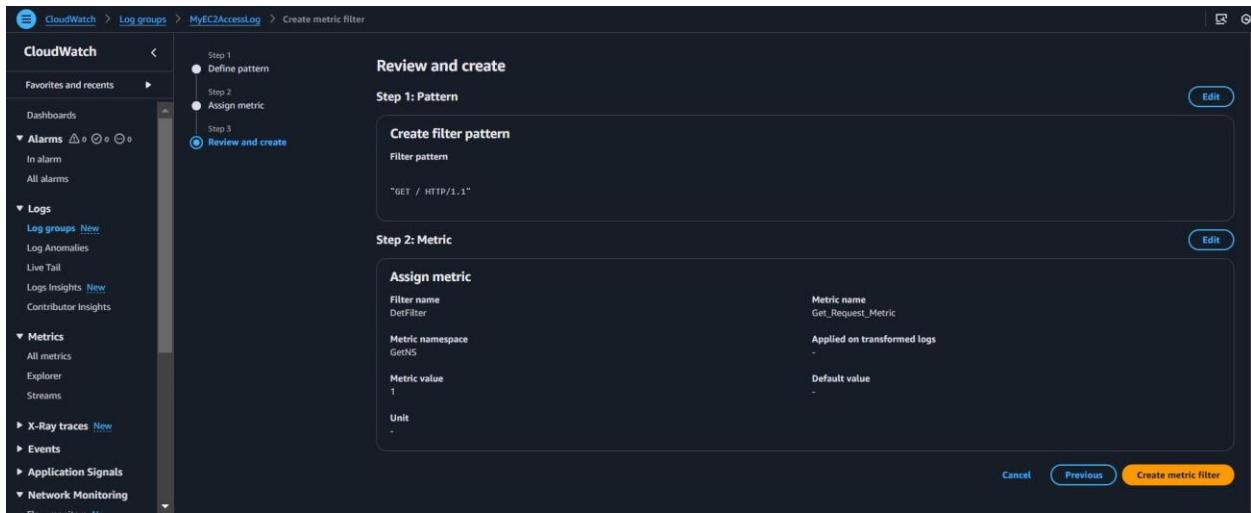
6 Create a CloudWatch Metric Filter

1 Select Log Stream and create a metric filter

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

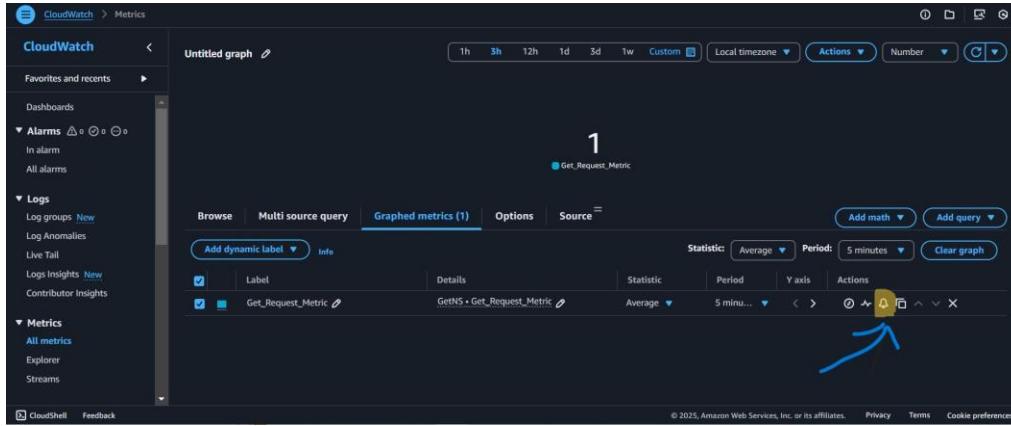
LinkedIn :- <https://www.linkedin.com/in/vaibhav-chaudhari-14016b22a/>

- Set Metric Details:
 - **Namespace:** GetNS
 - **Metric Name:** Get_Request_Metric
 - **Metric Value:** 1
- Click **Create Metric Filter.**



2 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!