

## AWS Monitoring using Cloud watch

### Overview

1. Creating custom dashboard
2. Setting up cloud Alarm
3. Configure cloudwatch log

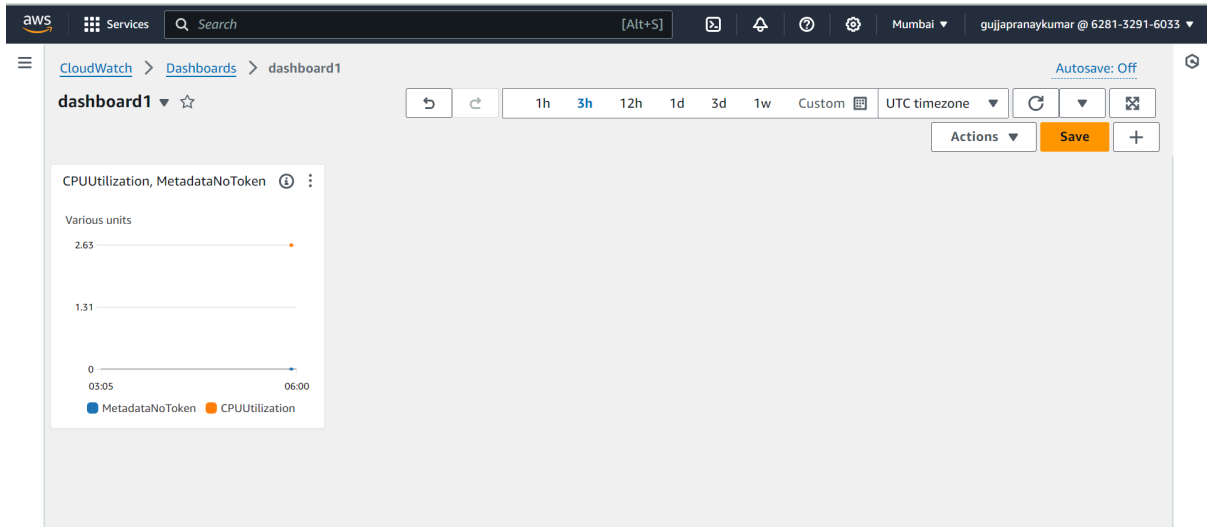
### Create a Custom Dashboard

1. Navigate to cloudwatch service
2. Click Dashboard -> Create Dashboard
3. Provide the name and proceed next and select any type of widget ex: Line
4. Select the data source as Metrics and then select EC2 in the Metrics
- A:
5. Now select Per-Instance metrics
6. Select any one of your nginx server and metric name as CPUUtilization
- A:

The screenshot shows the AWS CloudWatch 'Add metric graph' interface. At the top, there's a graph with a y-axis ranging from 0 to 1.31 and an x-axis showing time from 03:00 to 05:45. Below the graph, there's a table with columns: 'firstinstance', 'i-068ad9d293ad79166', and 'CPUUtilization'. The 'CPUUtilization' row is selected. To the right of the table, there are buttons for 'Add math' and 'Add query'. At the bottom right, there are 'Cancel' and 'Create widget' buttons.

7. Click on create widget
8. Your dashboard should now be created
9. You can analyze your instance CPUUtilization using the graph

A:



## Set a Cloud Alarm for CPUUtilization

1. Navigate to Alarm dashboard -> create alarm
2. Select Metric
3. Select the data source as Metrics and then select EC2 in the Metrics
4. Now select Per-Instance metrics
5. Select any one of your nginx server and metric name as CPUUtilization
6. In the period you can provide the minutes, say if 1 minute

A:

The screenshot shows the AWS CloudWatch Alarm configuration page. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information. The breadcrumb trail is 'CloudWatch > Alarms > Create new alarm'. The page is titled 'Metric' and contains a 'Graph' section and a 'Configuration' section. The 'Graph' section shows a line graph for 'CPUUtilization' with a y-axis ranging from 0 to 0.667 and an x-axis showing a time range from 09:30 to 11:30. The 'Configuration' section contains the following fields:

- Namespace: AWS/EC2
- Metric name: CPUUtilization
- InstanceId: i-068ad9d293ad79166
- Instance name: firstinstance
- Statistic: Average
- Period: 1 minute

7. Provide CPU utilization condition say if it is 5%, Click 'Next'

A:

### Conditions

Threshold type

☒ **Static**  
Use a value as a threshold

☐ **Anomaly detection**  
Use a band as a threshold

Whenever CPUUtilization is...  
Define the alarm condition.

☒ **Greater**  
> threshold

☐ **Greater/Equal**  
>= threshold

☐ **Lower/Equal**  
<= threshold

☐ **Lower**  
< threshold

than...  
Define the threshold value.

Must be a number

► **Additional configuration**

8. For sending email notification you can click create new topic by providing your email address

A:

Send a notification to the following SNS topic  
Define the SNS (Simple Notification Service) topic that will receive the notification.

☐ Select an existing SNS topic

☒ **Create new topic**

☐ Use topic ARN to notify other accounts

Create a new topic...  
The topic name must be unique.

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (\_).

Email endpoints that will receive the notification...  
Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

user1@example.com, user2@example.com

9. You can choose any EC2 action if your CPU utilization goes beyond 5 % you can select Stop this instance and click "Next"

A:

**EC2 action**

**Alarm state trigger**  
Define the alarm state that will trigger this action. Remove

☒ **In alarm**  
The metric or expression is outside of the defined threshold.

☐ **OK**  
The metric or expression is within the defined threshold.

☐ **Insufficient data**  
The alarm has just started or not enough data is available.

**Take the following action...**  
Define what will happen to the EC2 instance with the Instance ID i-068ad9d293ad79166 when this alarm is triggered.

☐ **Recover this instance**  
You can only recover certain EC2 instance types. [See documentation](#)

☒ **Stop this instance**  
You can only stop an instance if it is backed by an EBS volume. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

☐ **Terminate this instance**  
You will not be able to terminate this instance if termination protection is enabled. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

☐ **Reboot this instance**  
An instance reboot is equivalent to an operating system reboot. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

Add EC2 action

10. Provide the Alarm name / description click 'Next' , preview the settings and click 'create alarm'

A:

**Name and description**

**Alarm name**  
ec2-instancealarm

**Alarm description - optional** [View formatting guidelines](#)

**Edit** **Preview**

# This is an H1  
\*\*double asterisks will produce strong character\*\*  
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

11. Now you can see the alarm created

12. You can run the load test using jmeter to cross the Threshold your Instance will be stopped automatically and a mail will be triggered as below

A:

**ALARM: "ec2-instancealarm" in Asia Pacific (Mumbai)** Inbox x

**AWS Notifications**  
to me ▾

10:43 PM (1 minute ago) ☆ 😊 ↶ ⋮

You are receiving this email because your Amazon CloudWatch Alarm "ec2-instancealarm" in the Asia Pacific (Mumbai) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [6.380445185391014 (16/02/24 17:07:00)] was greater than the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Friday 16 February, 2024 17:13:37 UTC".

View this alarm in the AWS Management Console:  
<https://ap-south-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-south-1#alarmsV2:alarm/ec2-instancealarm>

Alarm Details:

- Name: ec2-instancealarm
- Description:
- State Change: INSUFFICIENT\_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [6.380445185391014 (16/02/24 17:07:00)] was greater than the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Friday 16 February, 2024 17:13:37 UTC
- AWS Account: 628132916033
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:628132916033:alarm:ec2-instancealarm

## Create IAM roles and policies for cloud watch logs

1.Navigate to IAM dashboard

2.Click on policies -> create policy

A:

**Policy details**

**Policy name**  
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+=,.,@-\_' characters.

**Description - optional**  
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+=,.,@-\_' characters.

3.Navigate to JSON tab

4.Modify the json as below

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {
```

```

"Effect": "Allow",

"Action": [

"logs:CreateLogGroup",

"logs:CreateLogStream",

"logs:PutLogEvents",

"logs:DescribeLogStreams"

],

"Resource": [

"*"

]

}

]

}

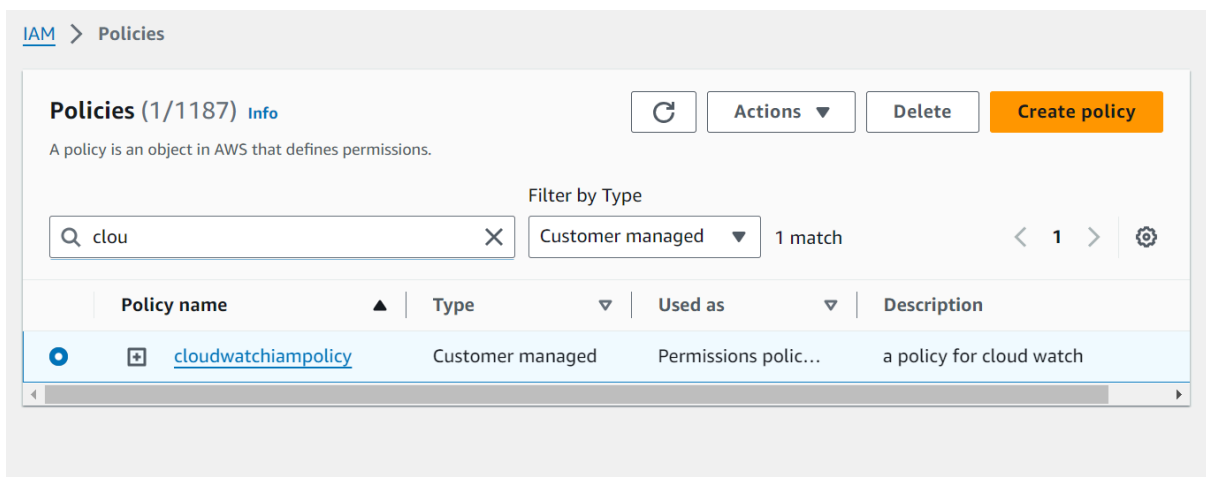
```

5. Click on Add tags, you can provide the tags and click on review

6. Provide the name and the description

7. Review once and click on create policy, your policy will be created

A:



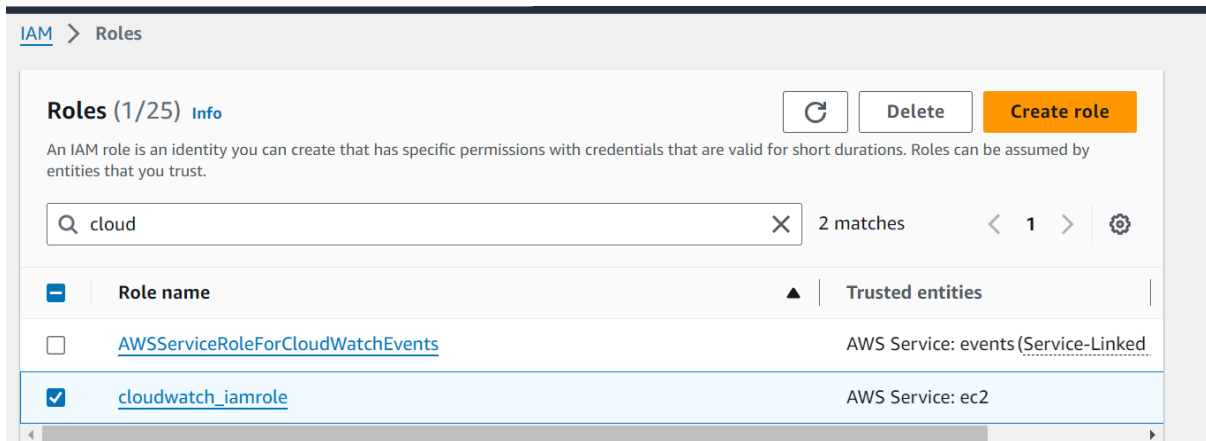
8. Navigate to IAM roles tab -> click create role

9. Select service as ec2 and click on Next

10. Select the policy that you have created and then click Next

11. Provide the role name, review once and then create role your role should be created

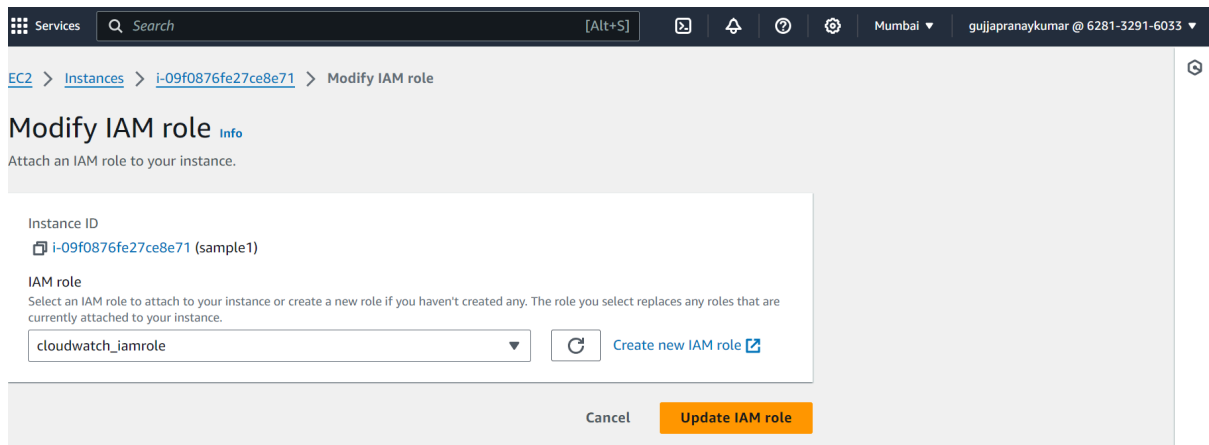
A:



12. You need to attach this Role to the ec2 instance (nginx server which you configured in cloudwatch)

13. Select the instance -> Actions -> security -> Modify IAM rule, you need to select the IAM role and update the IAM role

A:



## Configure the cloudwatch logs

1. Navigate to your ec2 instance (nginx server which you configured in cloudwatch)

2. Run `sudo yum install awslogs -y`

3. Now open the `awscli.conf` using `sudo vi /etc/awslogs/awscli.conf` and add modify your region

4. Open `awslogs.conf` file using `sudo vi /etc/awslogs/awslogs.conf`

```
[plugins]
cwlogs = cwlogs
[default]
region = ap-south-1
```

A:

5. Modify the lines as below

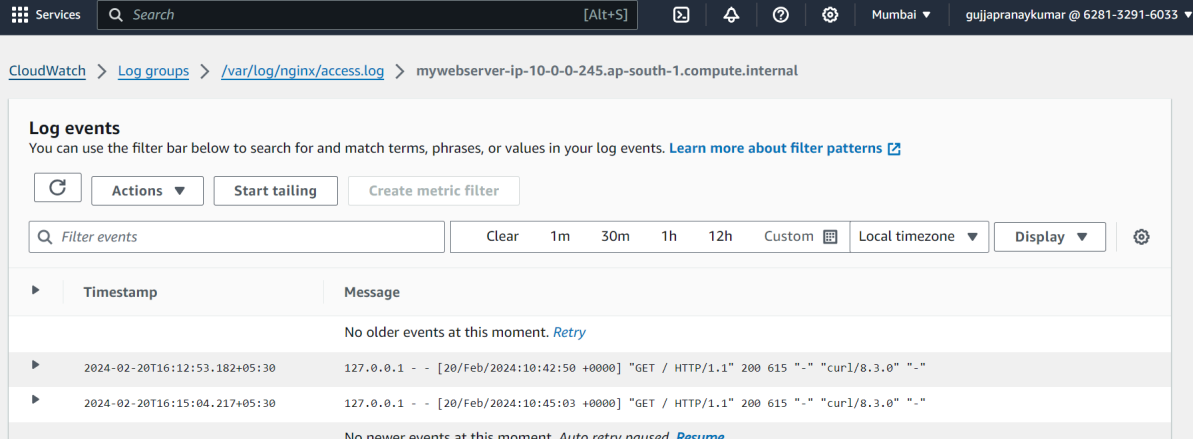
6. Now start your aws logs service using the following command `sudo systemctl start awslogs`

A:

```
[/var/log/nginx/error.log]
datetime_format = %b %d %H:%M:%S
file = /var/log/nginx/error.log
buffer_duration = 5000
log_stream_name = mywebserver-{instance_id}
initial_position = start_of_file
log_group_name = /var/log/nginx/error.log
```

7. Now navigate to your cloudwatch console and navigate to the log group and you will see the logs attached as below

A:



The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation at the top reads: CloudWatch > Log groups > /var/log/nginx/access.log > mywebserver-ip-10-0-0-245.ap-south-1.compute.internal. Below the navigation, there's a section titled "Log events" with a subtext: "You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)". There are buttons for "Actions", "Start tailing", and "Create metric filter". A search bar labeled "Filter events" is present, along with filters for "Clear", "1m", "30m", "1h", "12h", "Custom", "Local timezone", and a "Display" dropdown. The log events are displayed in a table with two columns: "Timestamp" and "Message". The table shows two log entries:

Timestamp	Message
2024-02-20T16:12:53.182+05:30	127.0.0.1 - - [20/Feb/2024:10:42:50 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.3.0" "-"
2024-02-20T16:15:04.217+05:30	127.0.0.1 - - [20/Feb/2024:10:45:03 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.3.0" "-"

Below the table, there are messages: "No older events at this moment. [Retry](#)" and "No newer events at this moment. Auto retry paused. [Resume](#)".

END