

AWS (Amazon Web Services)

CloudWatch



Introduction to CloudWatch:

Overview:

CloudWatch is system monitoring by AWS for cloud resources so monitoring is so important, and it will make sure that your applications are running the right way and It allows you to predict a problem before it occurs.

Concepts in this service:

- **Logs**

The place to store your application logs in AWS, we must first define log groups they are whatever name we want, then we define your log expiration policy. So, you can have the logs re being retained indefinitely to never expire, or you can choose to expire them from anywhere between one day to 10 years.

And we have a feature called Live Tail that Enables you to view and analyze log events as they occur.

- **Metrics**

a variable you need to monitor like CPU Utilization, the Network In .

- **Alarms**

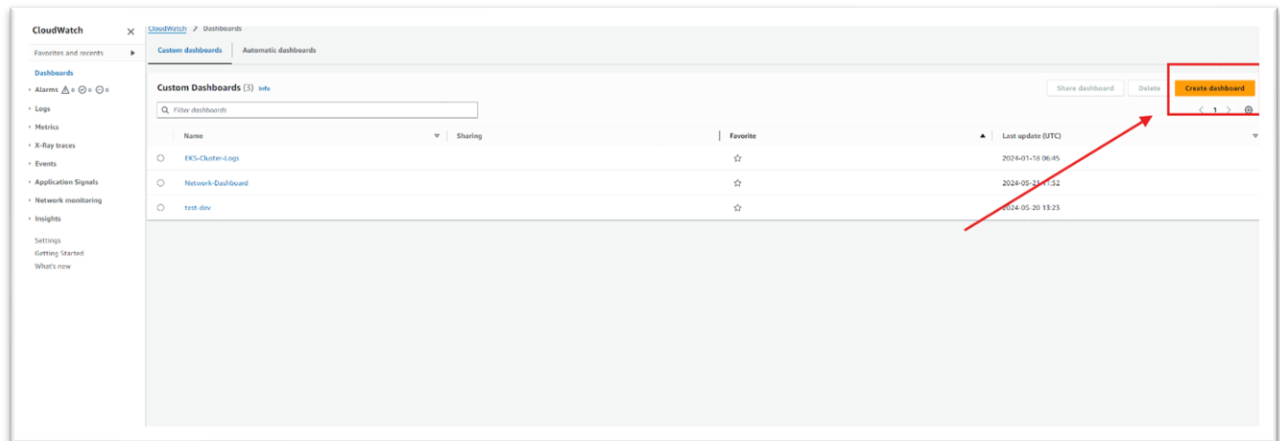
alarms as your system's early warning system. When a metric crosses a threshold, an alarm triggers an action.

- **Dashboard**

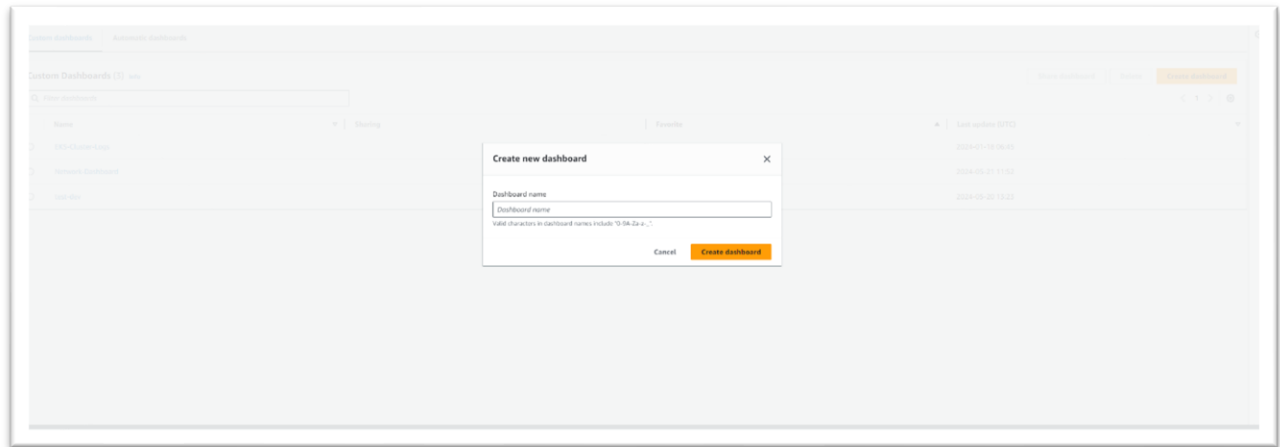
where you can create charts, graphs, and widgets to track specific metrics and keep an eye.

How can create a new Dashboard:

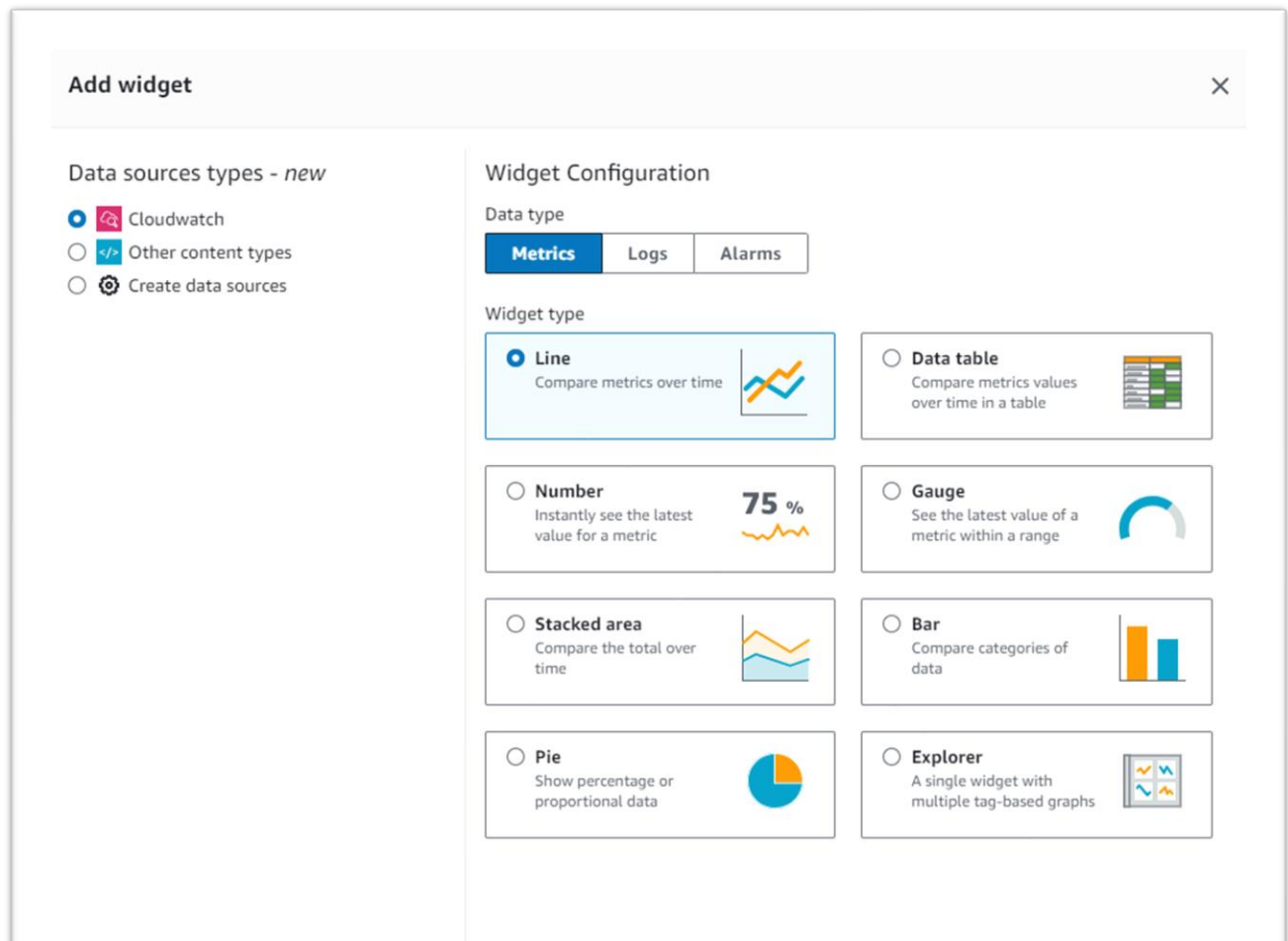
1) To create Dashboard we should have an account, log in to AWS, open the CloudWatch console, choose Dashboards from the navigation paned then select Create dashboard see picture



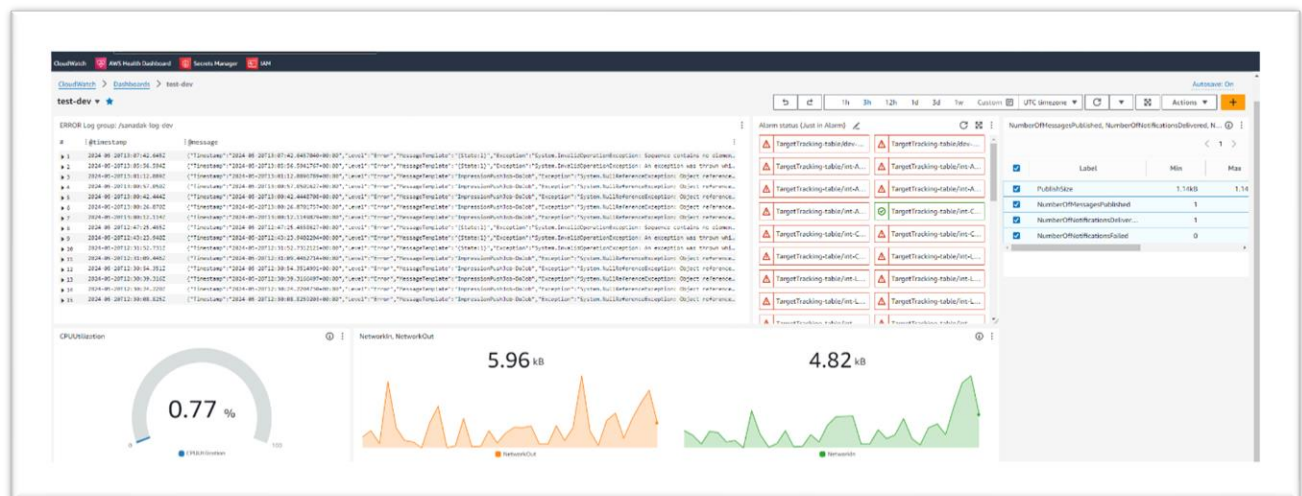
2) Choose the dashboard name



3) Add widget we need to monitor in dashboard



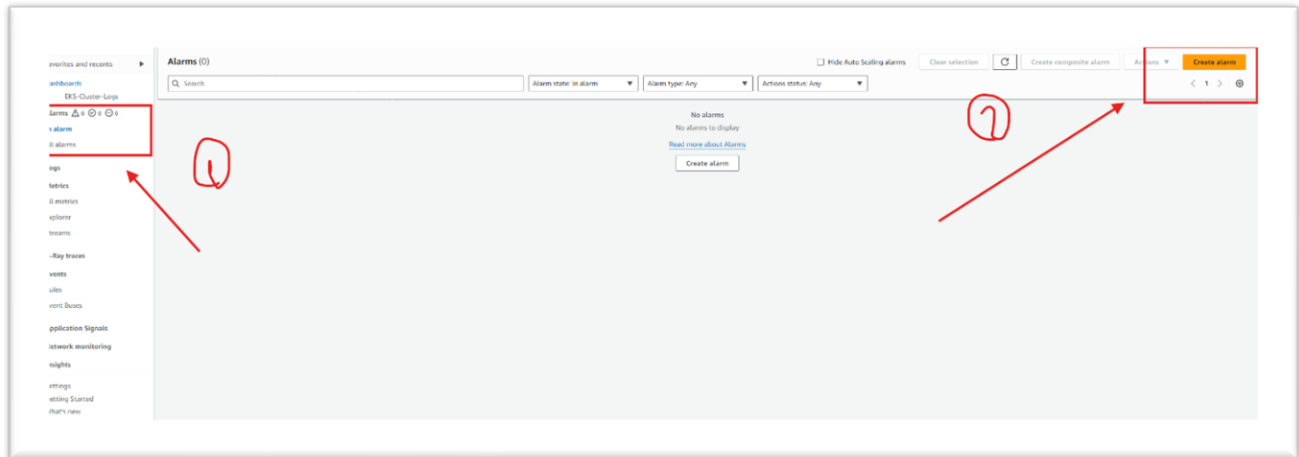
4) Finally, you can see what you added to the Dashboard for example:



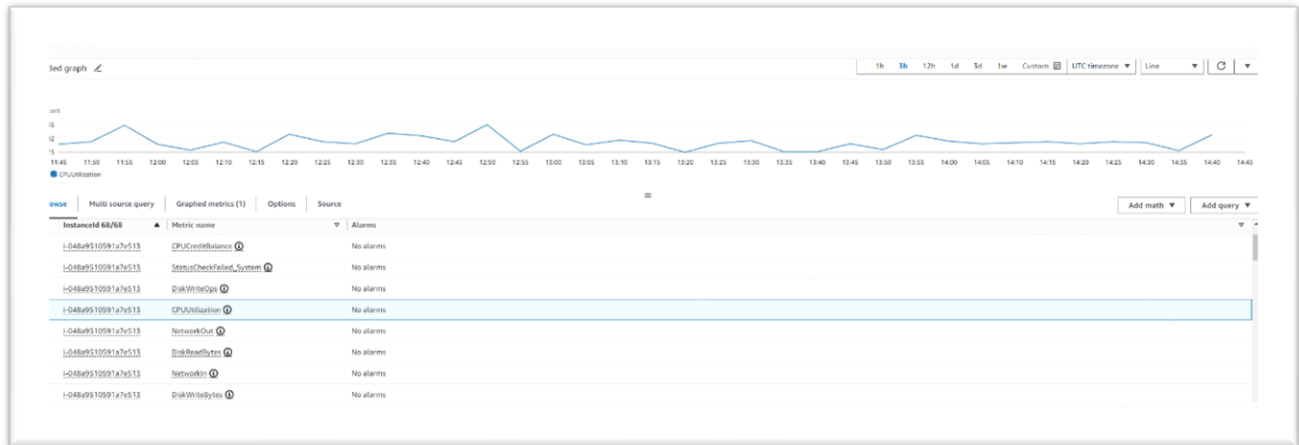
How can create alarm:

I want to create an alarm to alert me when the CPU up to 90% so let's go :

1) first we need log in to AWS, open the CloudWatch console choose **Alarms and choose Create Alarm**



2) Select Specify metric and conditions
In our example, we will use CPU utilization

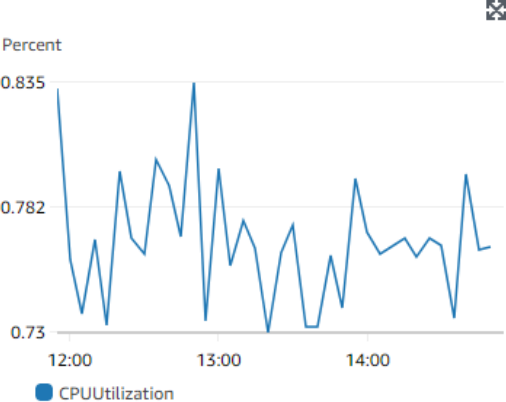


Specify metric and conditions

☐ Alarm recommendations [View details](#)

MetricEdit

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.



Namespace
AWS/EC2

Metric name
CPUUtilization

InstanceId
i-048a9510591a7e513

Instance name

Statistic
Average

Period
5 minutes

and then we choose the Conditions

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.

90

Must be a number

► Additional configuration

Cancel

Next

3) Configure actions:

Configure actions

Notification

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

☒ **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.

[Remove](#)

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

Send a notification to...

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

[Add notification](#)

Lambda action

[Add Lambda action](#)

Auto Scaling action

[Add Auto Scaling action](#)

EC2 action

[Add EC2 action](#)

Systems Manager action [info](#)

This action will create an Incident or OpsItem in Systems Manager when the alarm is **In alarm** state.

[Add Systems Manager action](#)

[Cancel](#) [Previous](#) [Next](#)

4) Add name and description:

Add name and description

Name and description

Alarm name

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)

 Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel

Previous

Next

5) finally Preview and create the alarm.

Create a Dashboard which targets all network issues:

- PacketsDropCount
- Active Connection Count
- Total Error Rate
- Network Receive/Transmit Throughput

The screenshot displays the AWS CloudWatch Network Dashboard. The top navigation bar includes links for 'CloudWatch', 'Dashboards', and 'Network Dashboard'. The main content area is divided into several sections:

- Summary Cards:**
 - Packets:** Shows a large '0' and a 'PacketsDropCount' metric.
 - Errors:** Shows a large '0' and a 'SnsTopicSizeError' metric.
 - ActiveConnections:** Shows a large '178' and an 'ActiveConnectionCount' metric.
- Graphs:**
 - Various units:** A line graph showing 'Clone from SnsTopic Integration Latency' (red), 'Clone from SnsTopic Count' (orange), and 'Clone from SnsTopic Latency' (green) over time.
 - TotalErrorRate:** A line graph showing error rates for various SNS topics over time.
- ActiveConnections, BytesInFromSource, BytesOutToDestination, ConnectionAttemptsCount, ConnectionEstablishedCount, ErrorPortAllocation, IdleTimeTolerance, PacketsDropCount:** A section with multiple metrics and their corresponding values (e.g., 29.5 k, 178, 37.2, 3.07 k, 29.5 k, 0, 1.74 k, 41.4 MB, 0.9, 41.4 MB, 37.1, 2.32 MB/s, 3.07 k, 0).
- Network Receiver/Transmit Throughput:** A section with two metrics: 'Network ReceiverThroughput' (5.12 kB/s) and 'Network TransmitterThroughput' (615 kB/s).