

What is CloudWatch?

- CloudWatch is a service used to monitor your AWS resources and applications that you run on AWS in real time. CloudWatch is used to collect and track metrics that measure your resources and applications.
- It displays the metrics automatically about every AWS service that you choose.
- You can create the dashboard to display the metrics about your custom application and also display the metrics of custom collections that you choose.
- You can also create an alarm to watch metrics. For example, you can monitor CPU usage, disk read and disk writes of Amazon EC2 instance to determine whether the additional EC2 instances are required to handle the load or not. It can also be used to stop the instance to save money.

Following are the terms associated with CloudWatch:

- **Dashboards:** CloudWatch is used to create dashboards to show what is happening with your AWS environment.
- **Alarms:** It allows you to set alarms to notify you whenever a particular threshold is hit.
- **Logs:** CloudWatch logs help you to aggregate, monitor, and store logs.
- **Events:** CloudWatch help you to respond to state changes to your AWS resources.

Creating a Dashboard

- Sign in to the AWS Management Console.
- I created an EC2 instance, and the name of an EC2 instance is **EC2instance**.

The screenshot displays the AWS Management Console interface. On the left sidebar, the 'INSTANCES' section is expanded, showing 'Instances' as the selected option. The main content area shows a table of EC2 instances. One instance is listed with the name 'EC2instance', ID 'i-0134dad99c6abab71', type 't2.micro', and state 'running'. Below the table, the details for the selected instance are shown, including its public DNS address 'ec2-34-204-72-128.compute-1.amazonaws.com' and its public IP address '34.204.72.128'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv6
EC2instance	i-0134dad99c6abab71	t2.micro	us-east-1c	running	Initializing	None	ec2-34-204-72-128.co...	34.2

Instance: i-0134dad99c6abab71 (EC2instance)		Public DNS: ec2-34-204-72-128.compute-1.amazonaws.com	
Description	Status Checks	Monitoring	Tags
Instance ID	i-0134dad99c6abab71	Public DNS (IPv4)	ec2-34-204-72-128.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	34.204.72.128
Instance type	t2.micro	IPv6 IPs	-

- Move to the **CloudWatch** service.

The screenshot shows the AWS CloudWatch Overview page. The left sidebar contains a navigation menu with options like Dashboards, Alarms, Billing, Events, Rules, Event Buses, Logs, Insights, Metrics, and Favorites. The 'Dashboards' option is highlighted. The main content area is titled 'CloudWatch: Overview' and shows a dropdown for 'All resources'. Below this, there are two sections: 'Alarms by AWS service' and 'Recent alarms'. The 'Alarms by AWS service' section contains a table with columns for Status, Alarm, Insufficient, and OK. The table lists three services: Classic ELB, EC2, and Elastic Block Store, all with a status of 'OK'. The 'Recent alarms' section is empty and contains a message: 'Recent alarms will appear here. Learn more about CloudWatch Alarms.'

- Click on the **Dashboards** appearing on the left side of the console.

The screenshot shows the AWS CloudWatch Dashboards page. The left sidebar contains a navigation menu with options like Dashboards, Alarms, Billing, Events, Rules, Event Buses, Logs, Insights, Metrics, and Favorites. The 'Dashboards' option is highlighted. The main content area is titled 'Dashboards' and shows a 'Create dashboard' button. Below this, there is a table with columns for Name, Favorite, and Last updated (UTC). The table is empty and contains a message: 'You have no CloudWatch dashboards. Please create a dashboard.'

- Click on the **Create dashboard** button. Enter the dashboard name. Suppose I write the dashboard name as **WebServer**.

Create new dashboard


Dashboard name:

CancelCreate dashboard


- Click on the **Create dashboard**.
- Select the **widget type** and add to the dashboard.
- I add the **Line widget** to our dashboard.

Add to this dashboard


Select a widget type to configure and add to this dashboard.




Line
Compare metrics over time




Stacked area
Compare the total over time



Number
Instantly see the latest value for a metric



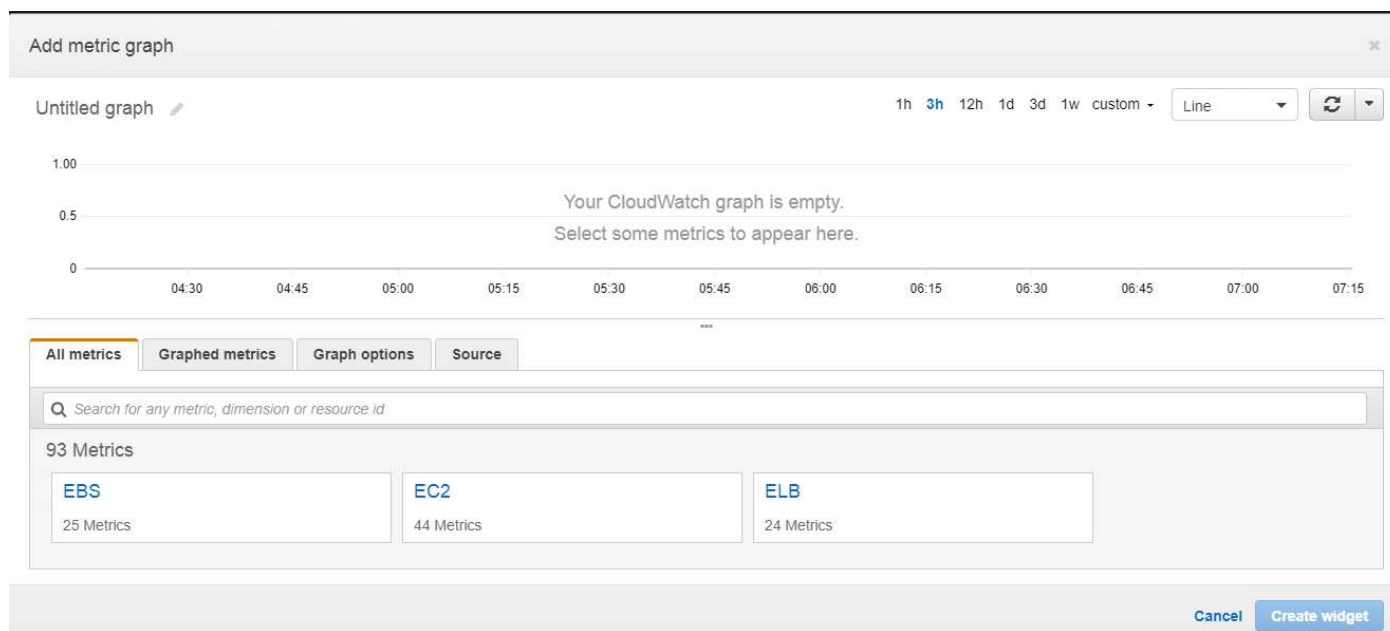
Text
Free text with markdown formatting



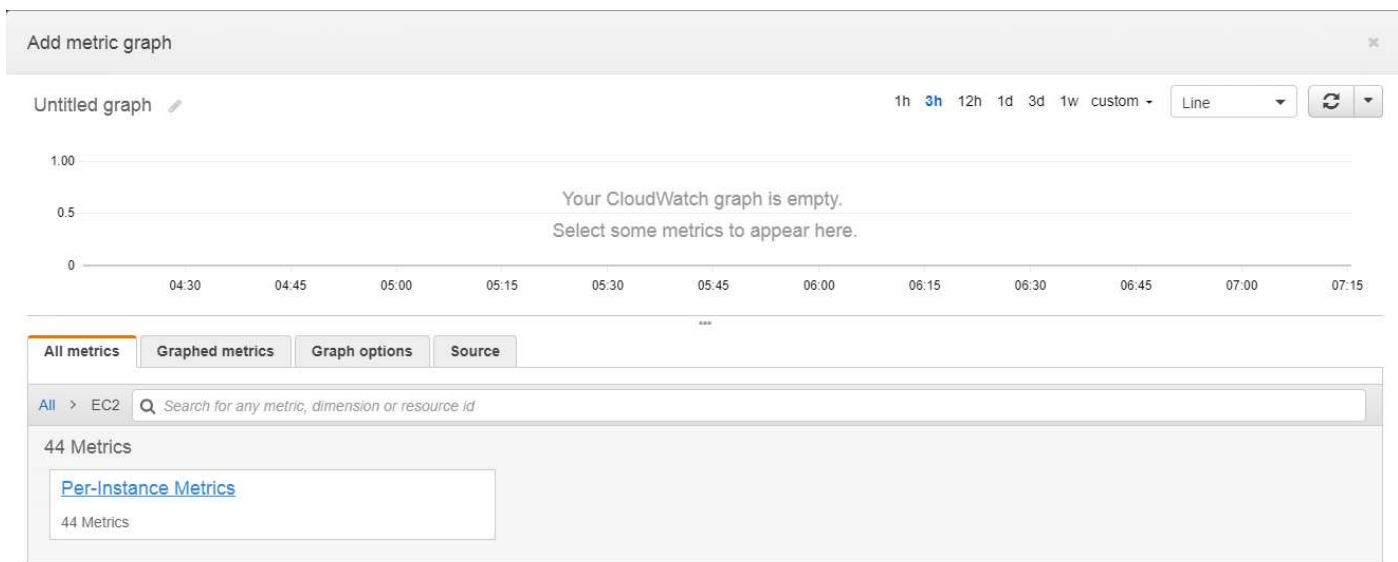
Query results
Explore results from Logs Insights

CancelConfigure

- After adding a line widget, you have to choose the service. I choose the EC2 service.



- Click on the **per-instance metrics**.



- Choose the **CPUUtilization** as a metric name and click on the create widget button.

Add metric graph

Untitled graph

1h 3h 12h 1d 3d 1w custom Line

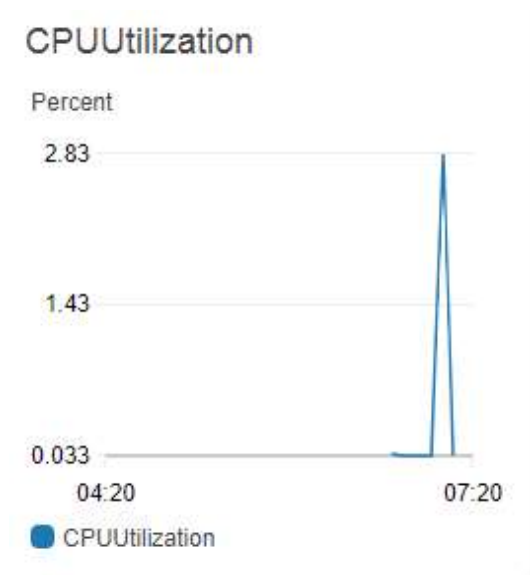
All metrics Graphed metrics (1) Graph options Source

All > EC2 > Per-Instance Metrics

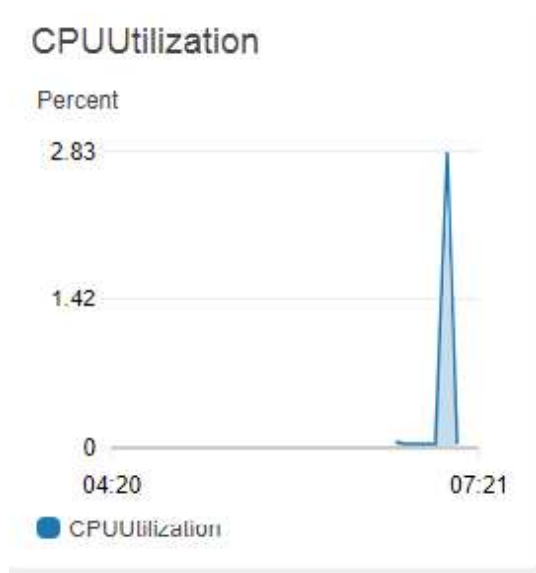
<input type="checkbox"/>	Instance Name (44)	InstanceId	Metric Name
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	DiskWriteBytes
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	NetworkOut
<input checked="" type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	CPUUtilization
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	NetworkPacketsIn
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	NetworkPacketsOut
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	DiskReadOps
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	CPUUtilization

Cancel Create widget

- The below screen shows the CPU utilization in the form of **line widget**.



- o I add another widget known as **stacked area**.



The above screen shows the CPU utilization in the form of a colored graph.

Creating an Alarm

- o Click on the **Alarms** appearing on the left side of the console.

The screenshot shows the AWS CloudWatch Overview page. The left sidebar contains navigation links for CloudWatch, Dashboards, Alarms, Billing, Events, Rules, Event Buses, Logs, Insights, Metrics, and Favorites. The main content area is titled 'CloudWatch: Overview' and includes a 'Time range' dropdown set to '3h' and an 'Action' button. Below the header, there's a section 'Alarms by AWS service' with a table showing the status of alarms for various services. The table has columns for 'Status', 'Alarm', 'Insufficient', and 'OK'. The services listed are Classic ELB, EC2, and Elastic Block Store. The 'Status' column shows 'Alarm' for Classic ELB and EC2, and 'Insufficient' for Elastic Block Store. The 'Insufficient' and 'OK' columns show '-' for all services. To the right of the table is a section 'Recent alarms' which is currently empty, displaying the message 'Recent alarms will appear here.' and a link 'Learn more about CloudWatch Alarms.'

Status	Alarm	Insufficient	OK
Alarm	Classic ELB	-	-
Alarm	EC2	-	-
Insufficient	Elastic Block Store	-	-

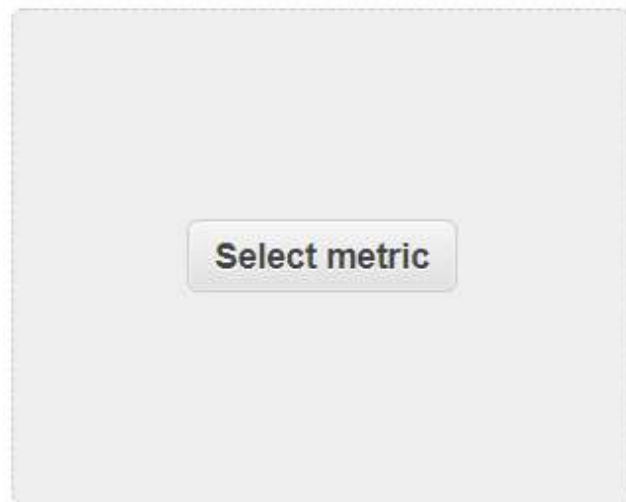
- Click on the **Create Alarm**.

The screenshot shows the AWS CloudWatch 'Create Alarm' page. The left sidebar is the same as the previous screenshot. The main content area has a 'Create Alarm' button and an 'Add to Dashboard' button. Below these buttons is a search bar with the text 'Search Alarms' and a 'Filter' dropdown set to 'All alarms'. There is also a checkbox for 'Hide all AutoScaling alarms'. The main content area is currently empty, displaying the message '0 Alarms selected' and 'Select an alarm above'.

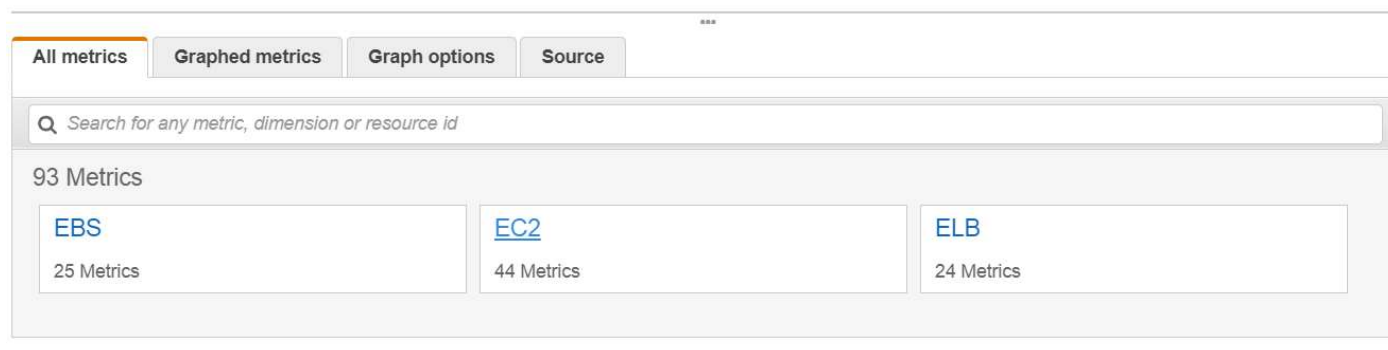
- **Select a metric.**

Metric

Select a metric to alarm on.



- o select an EC2.I



- o Click on the **Per-Instance Metrics**.

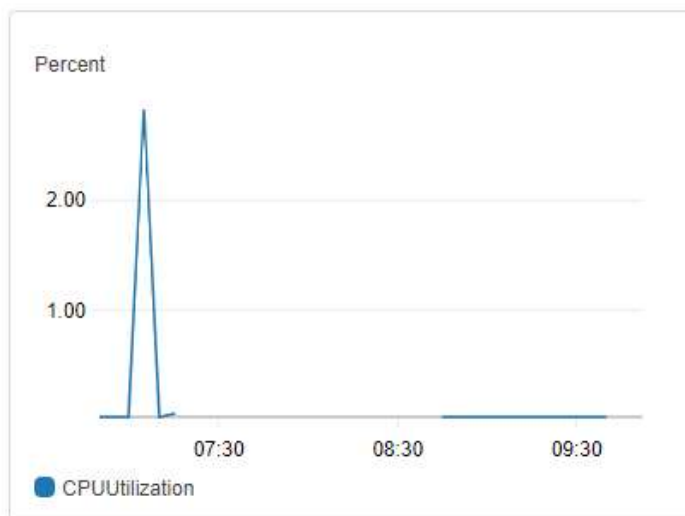


- o I select a **CPUUtilization** metric.

All metrics	Graphed metrics (1)	Graph options	Source
All > EC2 > Per-Instance Metrics <input type="text" value="Search for any metric, dimension or resource id"/>			
<input type="checkbox"/>	Instance Name (44)	InstanceId	Metric Name
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	NetworkOut
<input checked="" type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	CPUUtilization
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	NetworkPacketsIn
<input type="checkbox"/>	EC2Instance	i-0134dad99c6abab71	NetworkPacketsOut

Metric [Edit](#)

This alarm will trigger when the blue line goes up to or above the red line for 1 datapoints within 5 minutes



Namespace: AWS/EC2
Metric Name: CPUUtilization
InstanceId: i-0134dad99c6abab71
InstanceName: EC2instance
Period: 5 Minutes
Statistic: Average

- Enter the Alarm details.

Alarm details

Provide the details and threshold for your alarm. Use the graph to help set the appropriate threshold.

Name: CPU utilization

This field is mandatory and cannot be empty.

Description: My CPU utilization is 80%

Whenever: CPUUtilization

is:

for: 1 out of 1 datapoints

Additional settings

Provide additional configuration for your alarm.

Treat missing data as:



Actions

Define what actions are taken when your alarm changes state.

Notification

[Delete](#)

Whenever this alarm:

Send notification to:

[Select list](#)

Email list:

+ Notification

+ AutoScaling Action

+ EC2 Action

[Cancel](#)

[Create Alarm](#)


The above details show that when CPU utilization is greater than 80%, then an alarm is triggered and sent to the email address that you mentioned while filling the alarm details.

- Click on the link sent to your email address for confirmation and this confirmation shows that you are ready to receive the alerts.

Confirm new email addresses

Check your email inbox for a message with the subject "AWS Notification - Subscription Confirmation" and click the included link to confirm that you are willing to receive alerts to that address. AWS can only send notifications to confirmed addresses

Waiting for confirmation of 1 new email address

 [gakshita123@gmail.com](#) [Resend confirmation link](#)

Note: You have 72 hours to confirm these email addresses

[I will do it later](#)

[View Alarm](#)

[← Prev](#)

[Next →](#)



For Videos Join Our Youtube Channel: [Join Now](#)

Feedback

- Send your Feedback to feedback@javatpoint.com