Cloudwatch service

What is Metrics??

CloudWatch can collect metrics data and provide charts and graphs to visualize that data. Metrics can be collected for AWS resources like EC2 instances, RDS databases, Lambda functions, etc.

What is Alarms??

CloudWatch lets you set alarms that notify you about a certain threshold for a metric. This allows you to monitor your application and resources proactively.

What is Logs??

CloudWatch can collect log files generated by your resources and applications that are running on AWS. It provides storage and dashboards to visualize the log data.

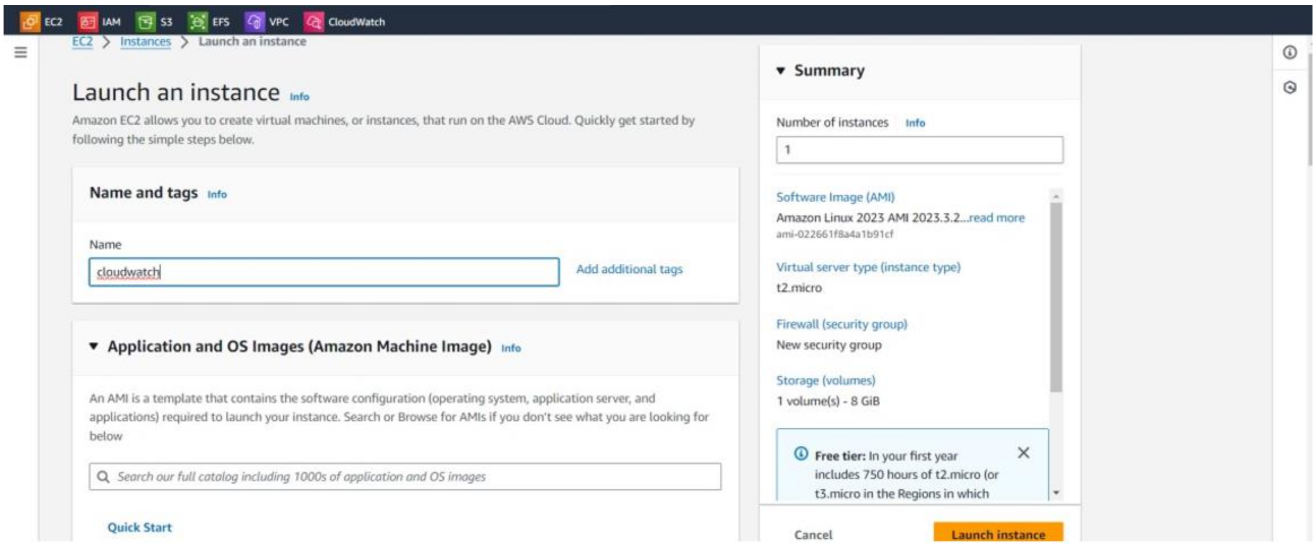
What is Events ??

CloudWatch Events allow you to trigger actions in reaction to changes in your resources or applications.

What is Dashboards??

CloudWatch provides fully customizable dashboards where you can add widgets with metrics and log data. This gives you a single view of the health and performance of your applications.

1. For performing this practical we need one public instance



EC2 IAM S3 EFS VPC CloudWatch

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: cloudwatch

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Summary

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2...read more
ami-022661f8a4a1b91cf

Virtual server type (instance type): t2.micro

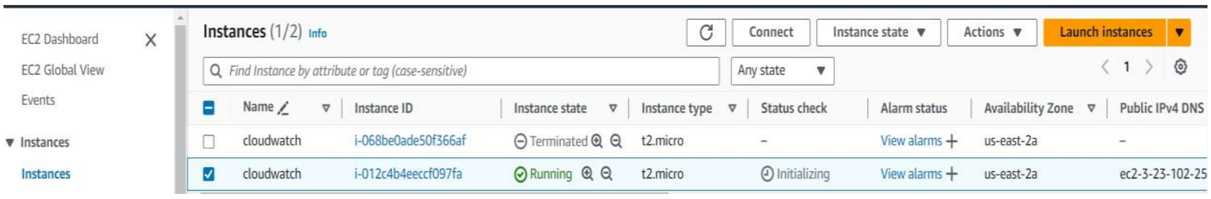
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

Cancel   Launch instance

2. **Instance Created successfully….**



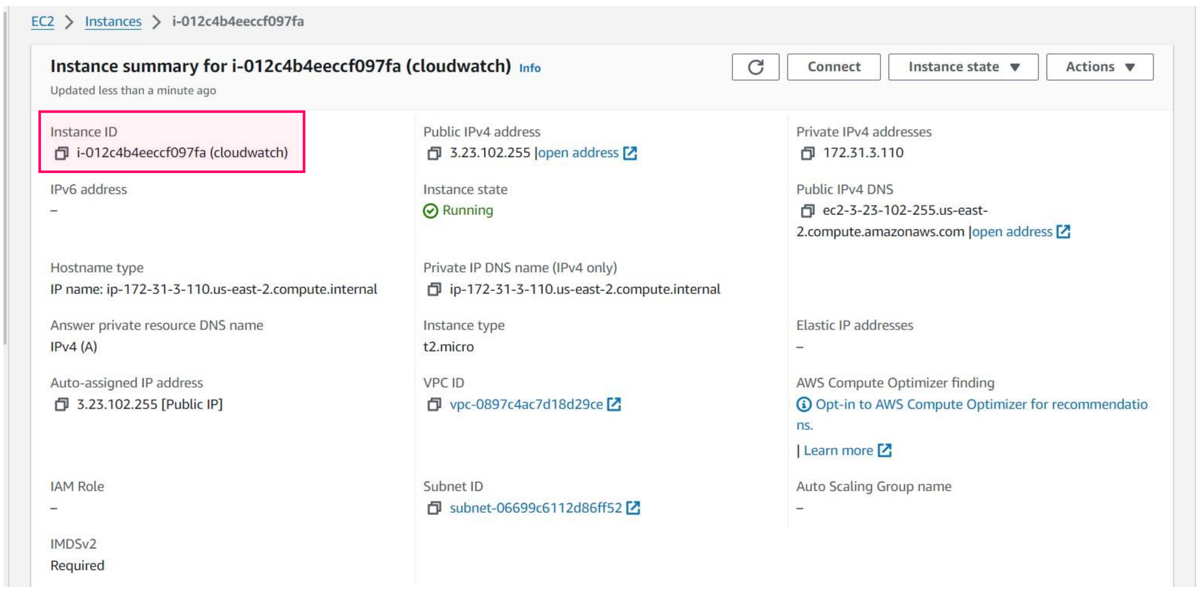
EC2 Dashboard
EC2 Global View
Events
Instances
Instances

Instances (1/2) Info   Connect   Instance state   Actions   Launch instances

Find Instance by attribute or tag (case-sensitive)   Any state

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	cloudwatch	i-068be0ade50f366af	Terminated	t2.micro	–	View alarms +	us-east-2a	–
☑	cloudwatch	i-012c4b4eeccf097fa	Running	t2.micro	Initializing	View alarms +	us-east-2a	ec2-3-23-102-25

3. Copy instance ID



EC2 > Instances > i-012c4b4eeccf097fa

Instance summary for i-012c4b4eeccf097fa (cloudwatch) Info
Updated less than a minute ago

Connect   Instance state   Actions

Instance ID: i-012c4b4eeccf097fa (cloudwatch)

Public IPv4 address: 3.23.102.255 |open address

Private IPv4 addresses: 172.31.3.110

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-23-102-255.us-east-2.compute.amazonaws.com |open address

Hostname type: IP name: ip-172-31-3-110.us-east-2.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-3-110.us-east-2.compute.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 3.23.102.255 [Public IP]

VPC ID: vpc-0897c4ac7d18d29ce

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

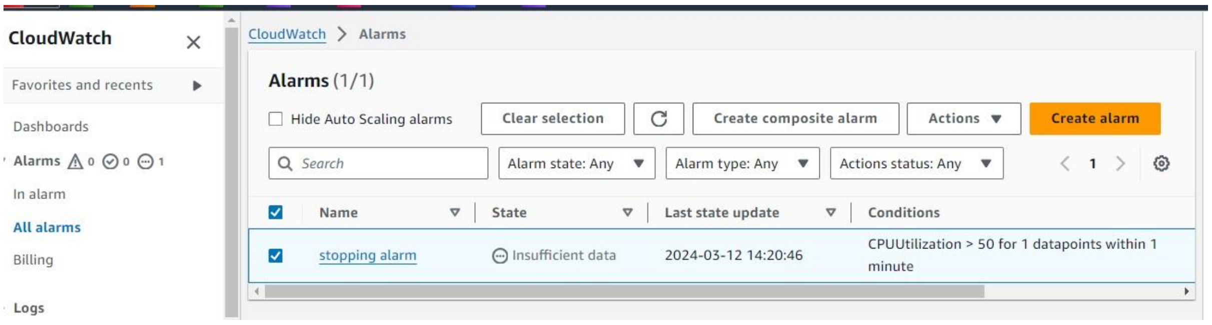
IAM Role: –

Subnet ID: subnet-06699c6112d86ff52

Auto Scaling Group name: –

IMDSv2: Required

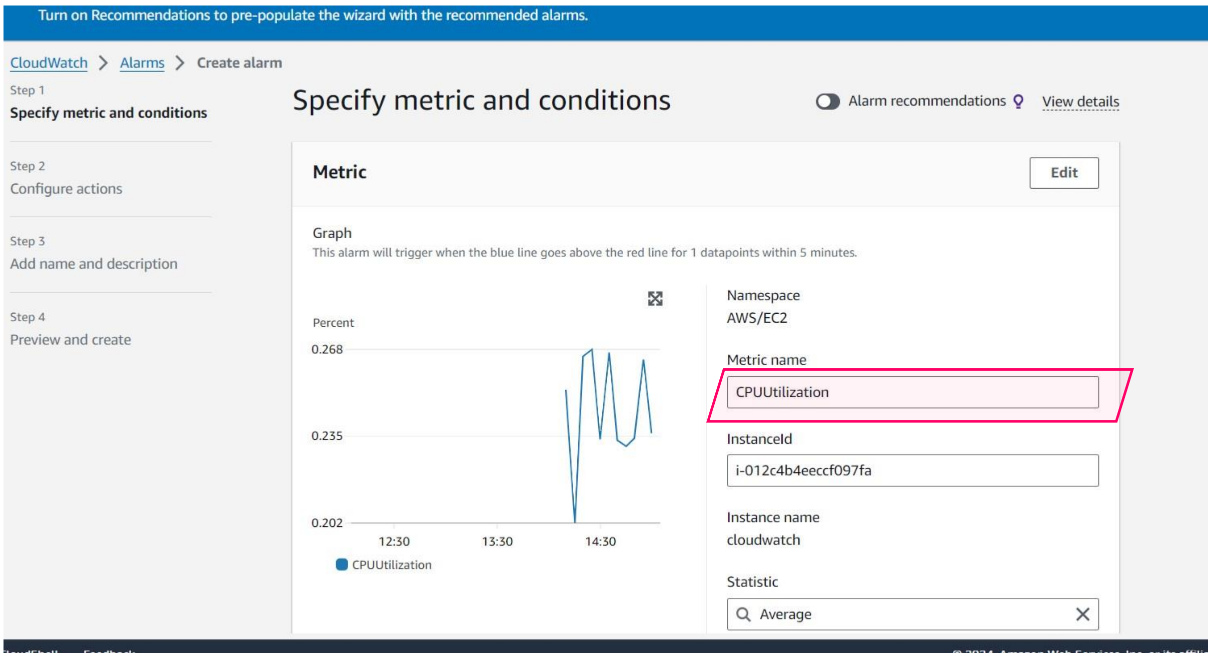
4. go to cloudwatch service and Under alarm option click on **Create alarm**



Alarms (1/1)

	Name	State	Last state update	Conditions
✓	stopping alarm	Insufficient data	2024-03-12 14:20:46	CPUUtilization > 50 for 1 datapoints within 1 minute

5. select the **CPUUtilization** Option



Turn on Recommendations to pre-populate the wizard with the recommended alarms.

Specify metric and conditions

Metric

Graph

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

Namespace: AWS/EC2

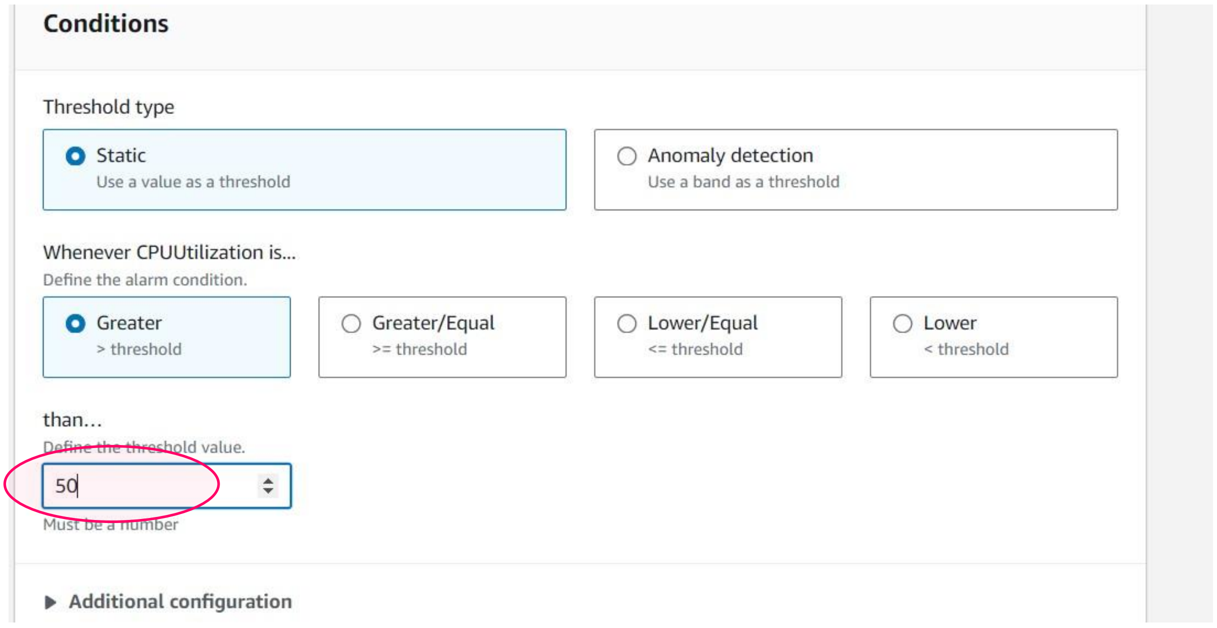
Metric name: CPUUtilization

InstanceId: i-012c4b4eeccf097fa

Instance name: cloudwatch

Statistic: Average

6. Add condition as per your requirement….



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…

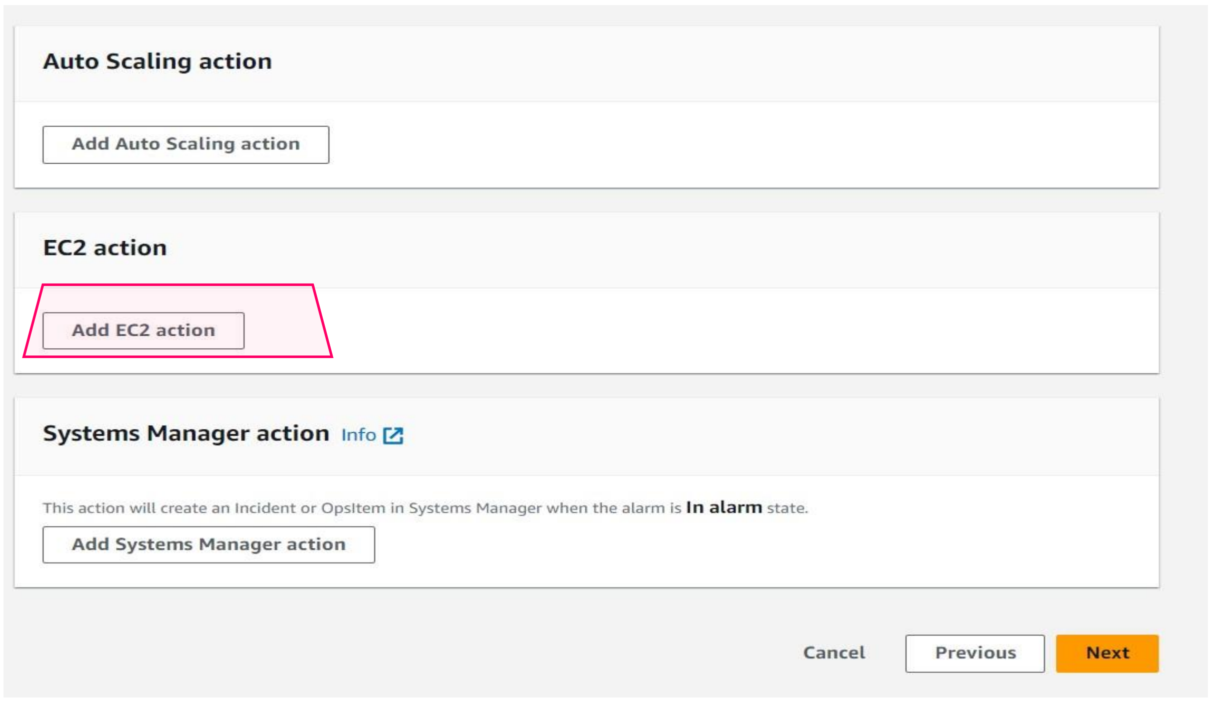
Define the threshold value.

50

Must be a number

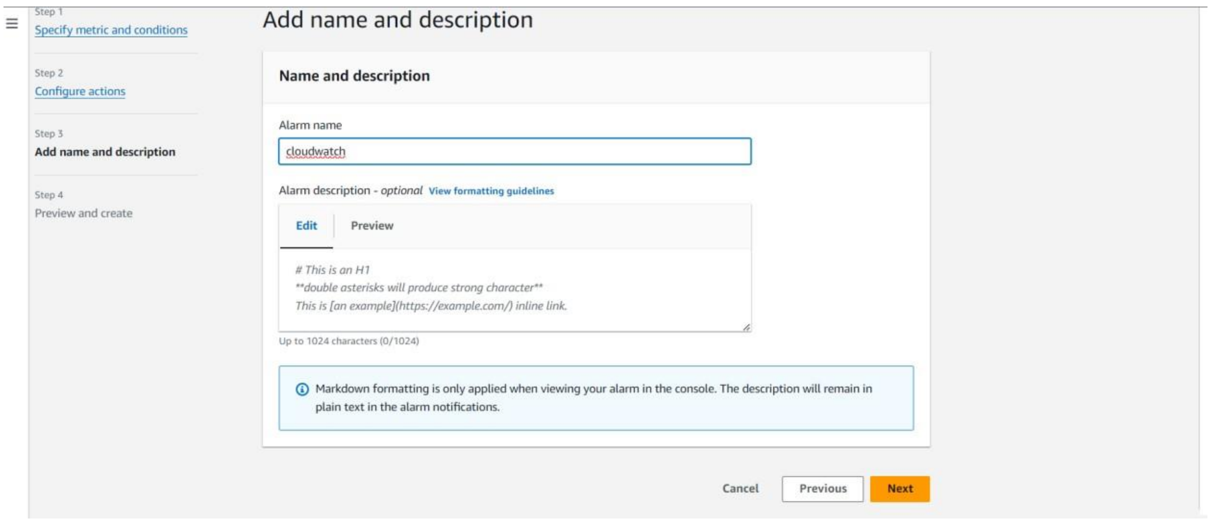
Additional configuration

7. Click on **Add Ec2 Action**



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

8. Give alarm as per your choice and click on next….



Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Add name and description

Name and description

Alarm name

cloudwatch

Alarm description - *optional* View formatting guidelines

Edit Preview

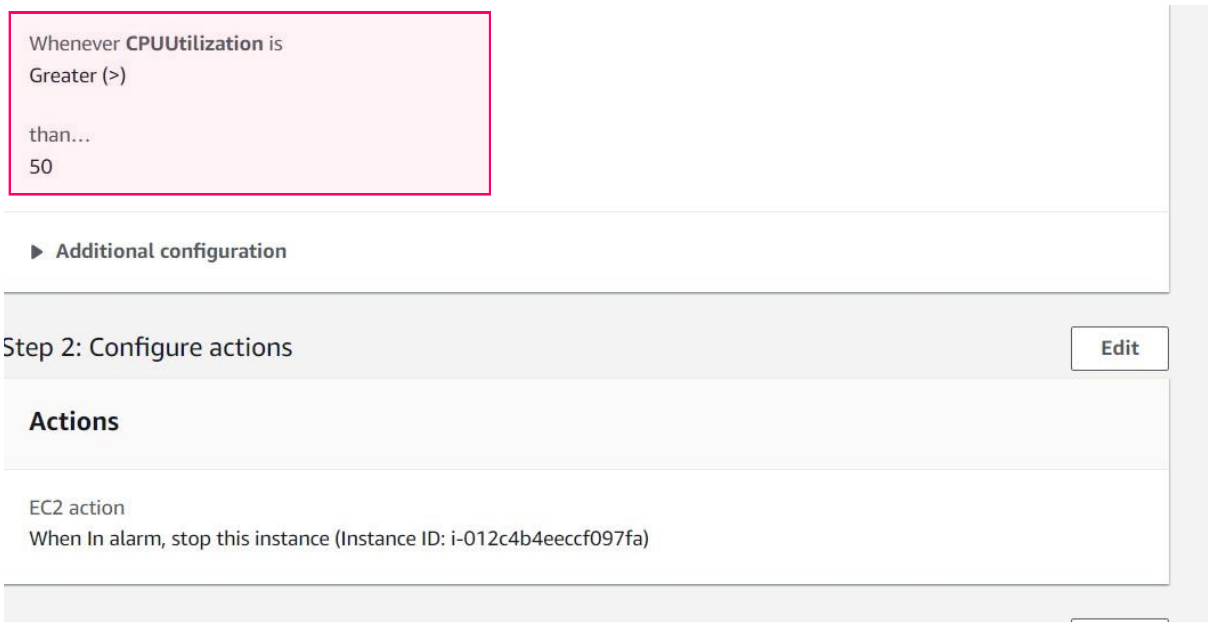
# This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.

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

9. Summary….



Whenever **CPUUtilization** is
Greater (>)

than…
50

Additional configuration

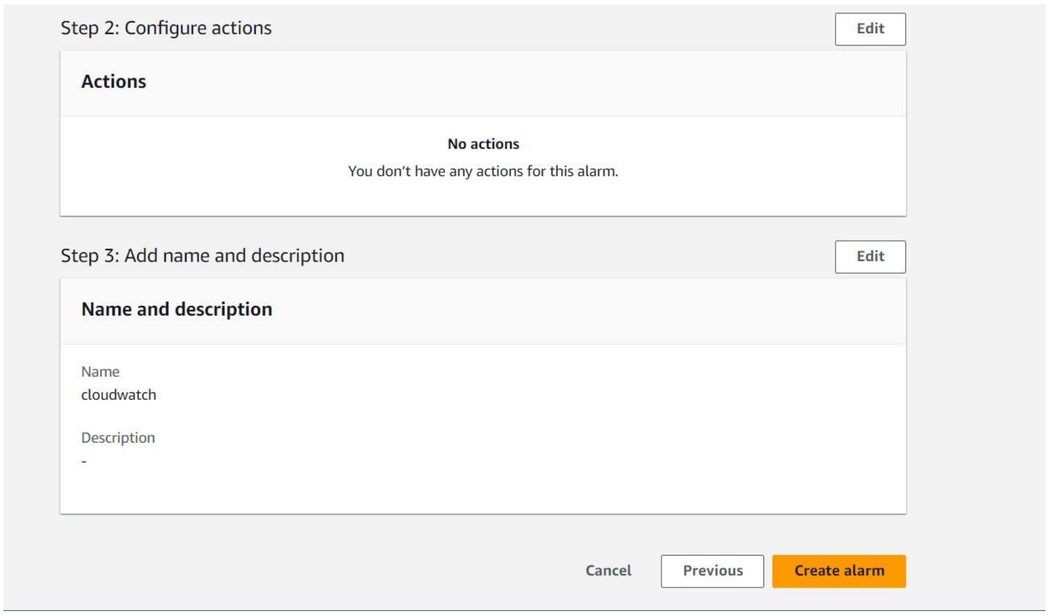
Step 2: Configure actions

Edit

Actions

EC2 action
When In alarm, stop this instance (Instance ID: i-012c4b4eeccf097fa)

10.Click on Create alarm


Step 2: Configure actions

Actions

No actions
You don't have any actions for this alarm.

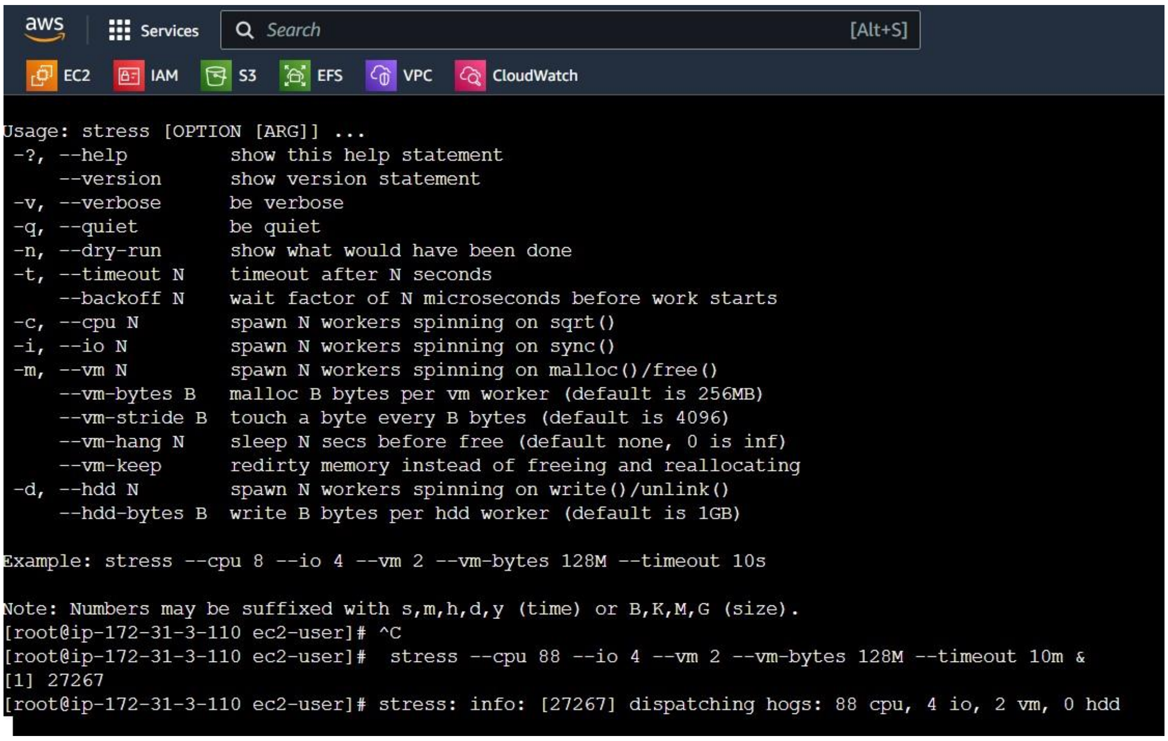
Step 3: Add name and description

Name and description

Name
cloudwatch

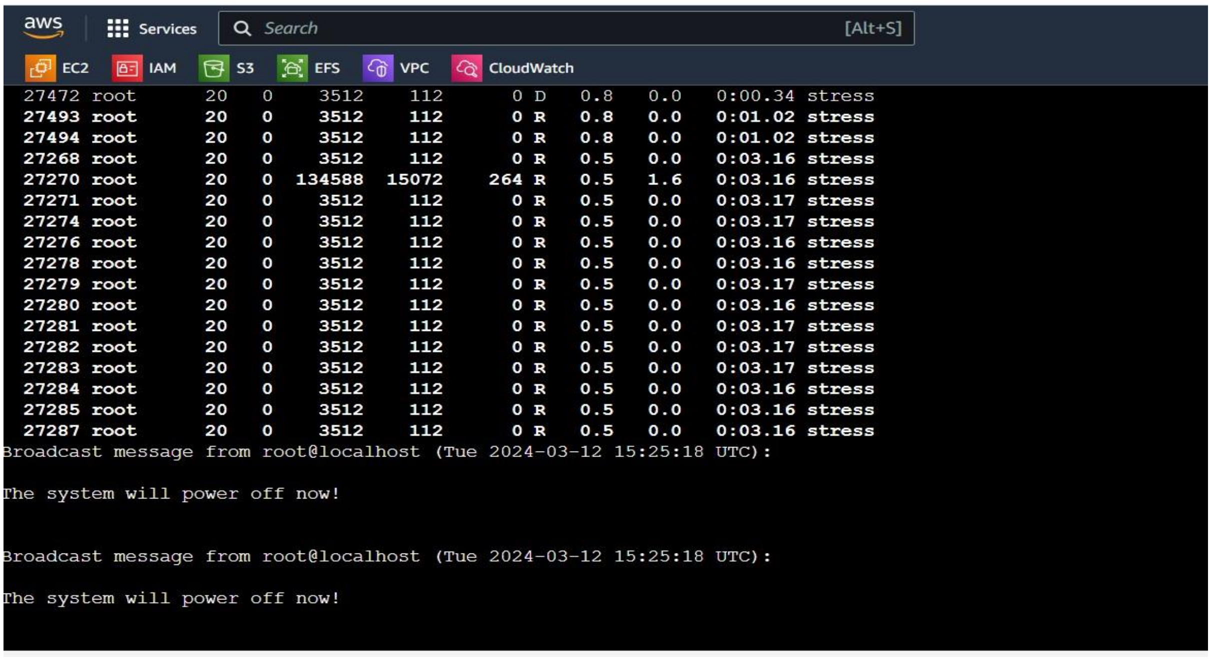
Description
-

11.After Performing configuration connect to the instance and give Load using **stress command**….



```
Usage: stress [OPTION [ARG]] ...
 -?, --help         show this help statement
     --version      show version statement
 -v, --verbose      be verbose
 -q, --quiet        be quiet
 -n, --dry-run      show what would have been done
 -t, --timeout N    timeout after N seconds
     --backoff N    wait factor of N microseconds before work starts
 -c, --cpu N        spawn N workers spinning on sqrt()
 -i, --io N         spawn N workers spinning on sync()
 -m, --vm N         spawn N workers spinning on malloc()/free()
     --vm-bytes B   malloc B bytes per vm worker (default is 256MB)
     --vm-stride B  touch a byte every B bytes (default is 4096)
     --vm-hang N    sleep N secs before free (default none, 0 is inf)
     --vm-keep      redirty memory instead of freeing and reallocating
 -d, --hdd N        spawn N workers spinning on write()/unlink()
     --hdd-bytes B  write B bytes per hdd worker (default is 1GB)

Example: stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 10s

Note: Numbers may be suffixed with s,m,h,d,y (time) or B,K,M,G (size).
[root@ip-172-31-3-110 ec2-user]# ^C
[root@ip-172-31-3-110 ec2-user]#  stress --cpu 88 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[1] 27267
[root@ip-172-31-3-110 ec2-user]# stress: info: [27267] dispatching hogs: 88 cpu, 4 io, 2 vm, 0 hdd
```

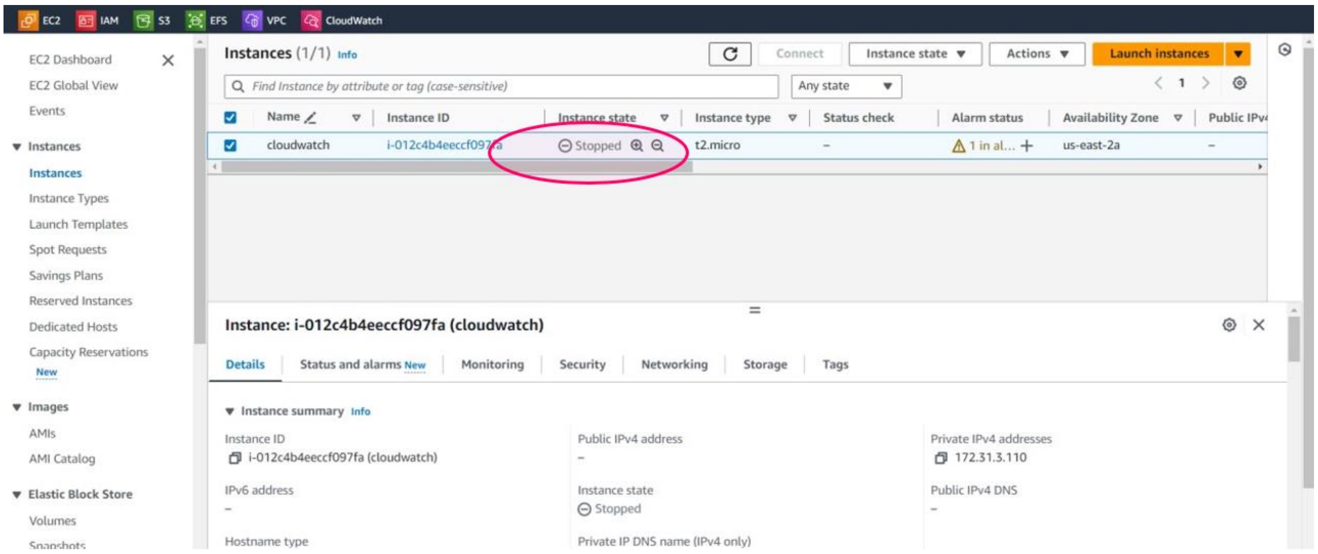
12.Instance is terminated because we apply apply greater that 50% load……



```
 27472 root      20   0    3512    112      0 D   0.8   0.0   0:00.34 stress
 27493 root      20   0    3512    112      0 R   0.8   0.0   0:01.02 stress
 27494 root      20   0    3512    112      0 R   0.8   0.0   0:01.02 stress
 27268 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
 27270 root      20   0  134588  15072    264 R   0.5   1.6   0:03.16 stress
 27271 root      20   0    3512    112      0 R   0.5   0.0   0:03.17 stress
 27274 root      20   0    3512    112      0 R   0.5   0.0   0:03.17 stress
 27276 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
 27278 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
 27279 root      20   0    3512    112      0 R   0.5   0.0   0:03.17 stress
 27280 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
 27281 root      20   0    3512    112      0 R   0.5   0.0   0:03.17 stress
 27282 root      20   0    3512    112      0 R   0.5   0.0   0:03.17 stress
 27283 root      20   0    3512    112      0 R   0.5   0.0   0:03.17 stress
 27284 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
 27285 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
 27287 root      20   0    3512    112      0 R   0.5   0.0   0:03.16 stress
Broadcast message from root@localhost (Tue 2024-03-12 15:25:18 UTC):

The system will power off now!


Broadcast message from root@localhost (Tue 2024-03-12 15:25:18 UTC):

The system will power off now!
```

#Instance is terminated because we apply condition…

(Above 50% cpu =stop instance)



Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
cloudwatch	i-012c4b4eeccf097fa	Stopped	t2.micro	–	1 in al...	us-east-2a	–

Instance: i-012c4b4eeccf097fa (cloudwatch)

Instance summary Info

Instance ID
i-012c4b4eeccf097fa (cloudwatch)

Public IPv4 address
–

Private IPv4 addresses
172.31.3.110

IPv6 address
–

Instance state
Stopped

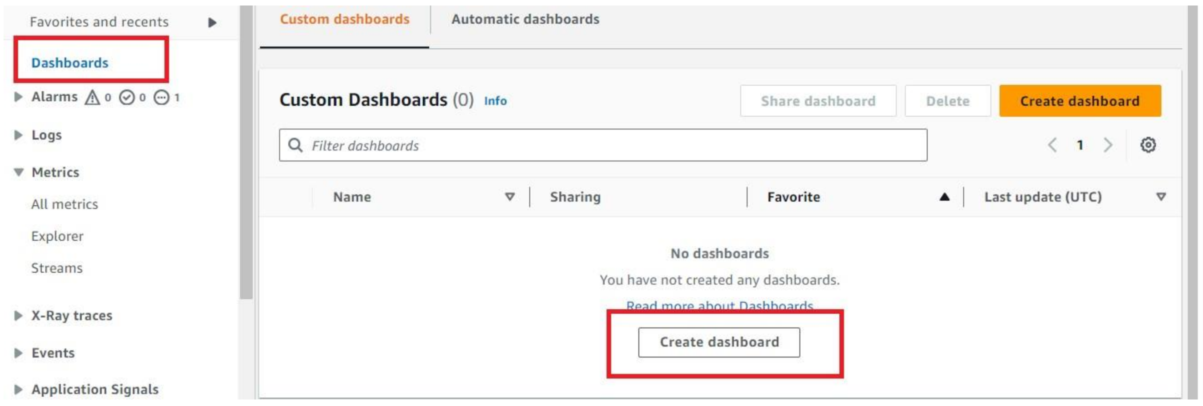
Public IPv4 DNS
–

Hostname type

Private IP DNS name (IPv4 only)

<u>Creating Custom dashboard:-</u>

1. Click on **Create dashboard** Option



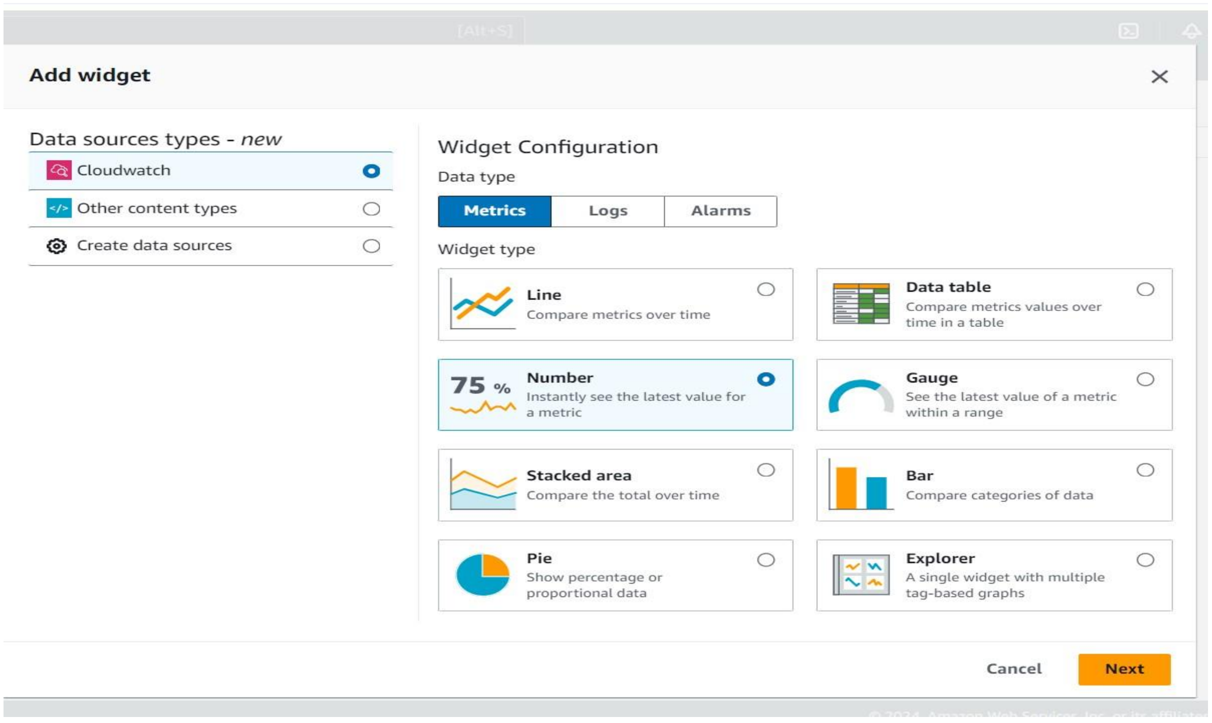
Custom Dashboards (0) Info

No dashboards

You have not created any dashboards.

Read more about Dashboards

2. Select **Widget type == Numbers**



Data sources types - *new*

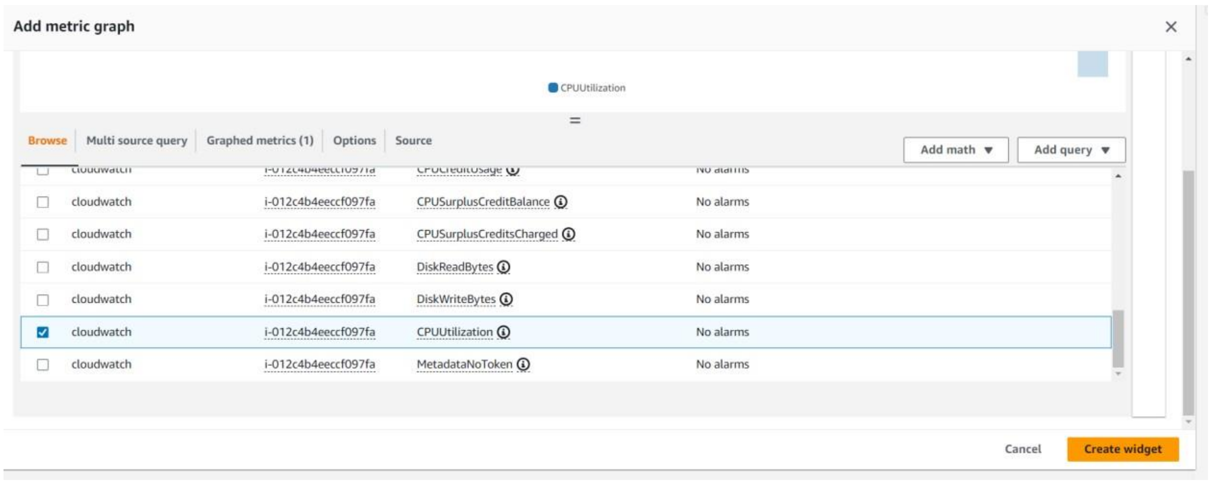
Widget Configuration

Data type: Metrics

Widget type

- Line — Compare metrics over time
- Data table — Compare metrics values over time in a table
- Number — Instantly see the latest value for a metric
- Gauge — See the latest value of a metric within a range
- Stacked area — Compare the total over time
- Bar — Compare categories of data
- Pie — Show percentage or proportional data
- Explorer — A single widget with multiple tag-based graphs

3. Select the Matrix as per your requirement and click on **Create widget** option



Add metric graph

🟦 CPUUtilization

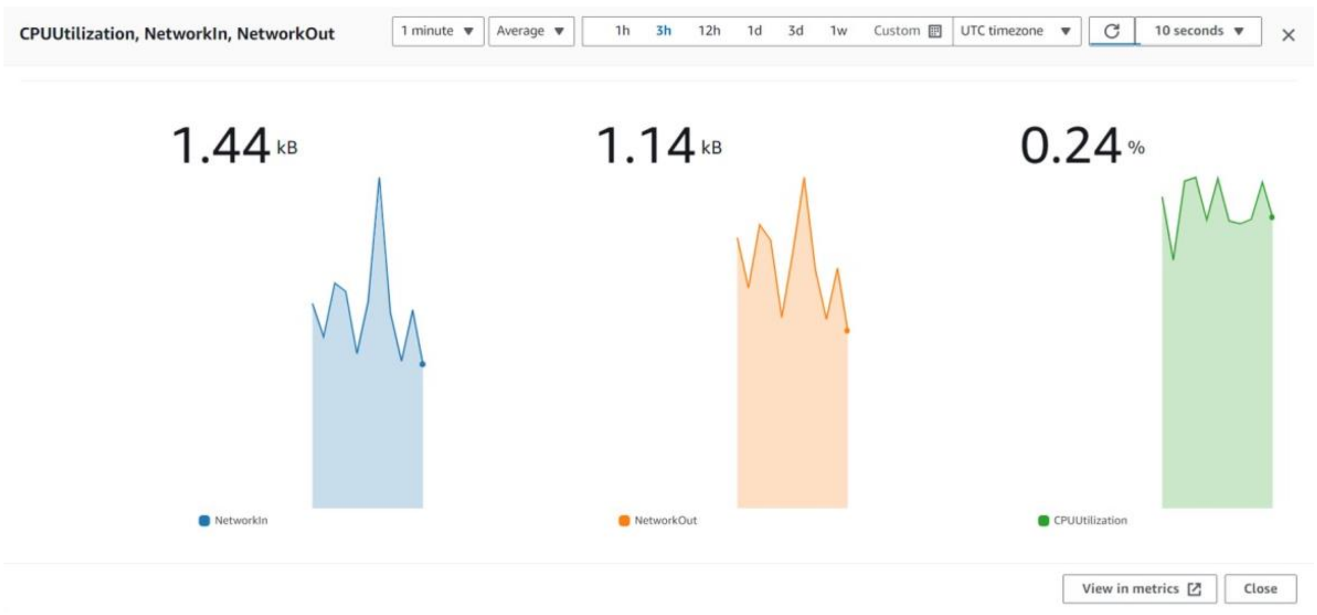
Browse | Multi source query | Graphed metrics (1) | Options | Source

Add math ▼ | Add query ▼

☐	cloudwatch	i-012c4b4eeccf097fa	CPUSurplusCreditBalance	No alarms
☐	cloudwatch	i-012c4b4eeccf097fa	CPUSurplusCreditsCharged	No alarms
☐	cloudwatch	i-012c4b4eeccf097fa	DiskReadBytes	No alarms
☐	cloudwatch	i-012c4b4eeccf097fa	DiskWriteBytes	No alarms
✓	cloudwatch	i-012c4b4eeccf097fa	CPUUtilization	No alarms
☐	cloudwatch	i-012c4b4eeccf097fa	MetadataNoToken	No alarms

Cancel | Create widget

4. **Dashboard Created successfully….**



CPUUtilization, NetworkIn, NetworkOut

1 minute ▼ | Average ▼ | 1h | 3h | 12h | 1d | 3d | 1w | Custom | UTC timezone ▼ | 10 seconds ▼