

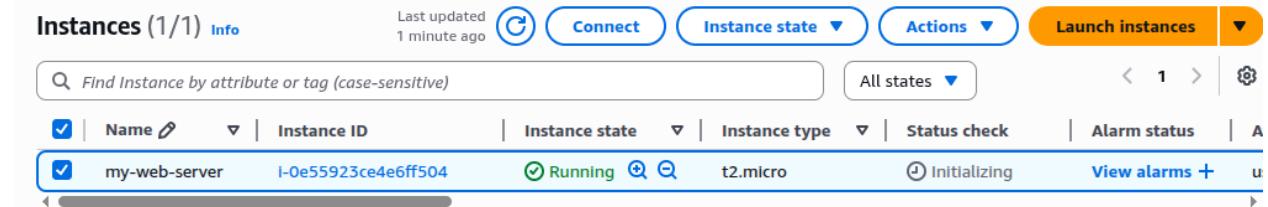
Implementation of CloudWatch CPU Alarm and SNS Email Notification

Overview

Amazon CloudWatch alarm that monitors the CPU usage of an EC2 instance. When the CPU exceeds a threshold (80%), the alarm triggers an SNS email notification so the support team can take action.

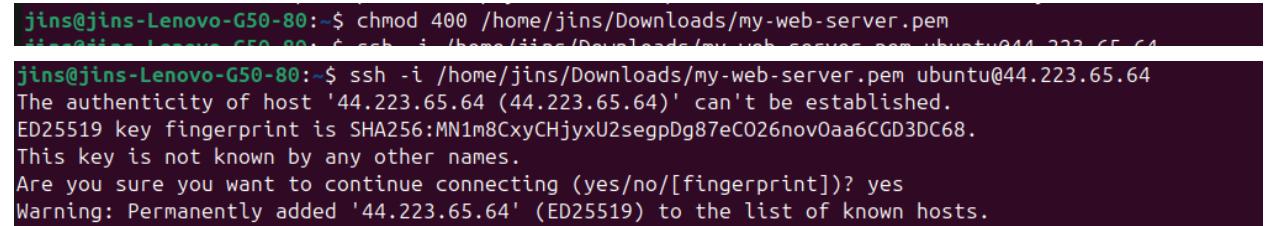
Prerequisites

- 1.AWS EC2 instance (t2.micro) with port 22 allowed (allow only your IP) for ssh access and port 80 allowed (for http traffic from the internet).
2. Select OS ubuntu 24.04
- 3.Create key value pair for ssh access change permission to 400 for maintaining best security practice.



The screenshot shows the AWS CloudWatch Instances console. At the top, there's a header with 'Instances (1/1)' and a 'Info' button. Below the header are several filters: 'Name' (with a search icon), 'Instance ID' (showing 'i-0e55923ce4e6ff504'), 'Instance state' (showing 'Running' with a green checkmark), 'Instance type' (showing 't2.micro'), 'Status check' (showing 'Initializing'), and 'Alarm status' (showing 'View alarms +'). A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and a dropdown says 'All states'. On the right, there are buttons for 'Launch instances' and other navigation controls. Below the filters, a table lists the instance details.

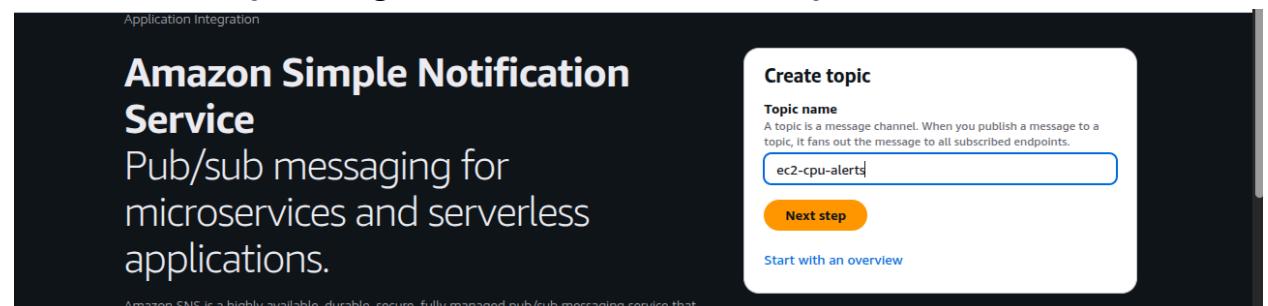
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
my-web-server	i-0e55923ce4e6ff504	Running	t2.micro	Initializing	View alarms +



The terminal session starts with the command 'ssh -i /home/jins/Downloads/my-web-server.pem ubuntu@44.223.65.64'. It then displays a warning about the host fingerprint and asks if the user wants to continue connecting. The user responds with 'yes' and a warning message is shown about adding the host to the list of known hosts.

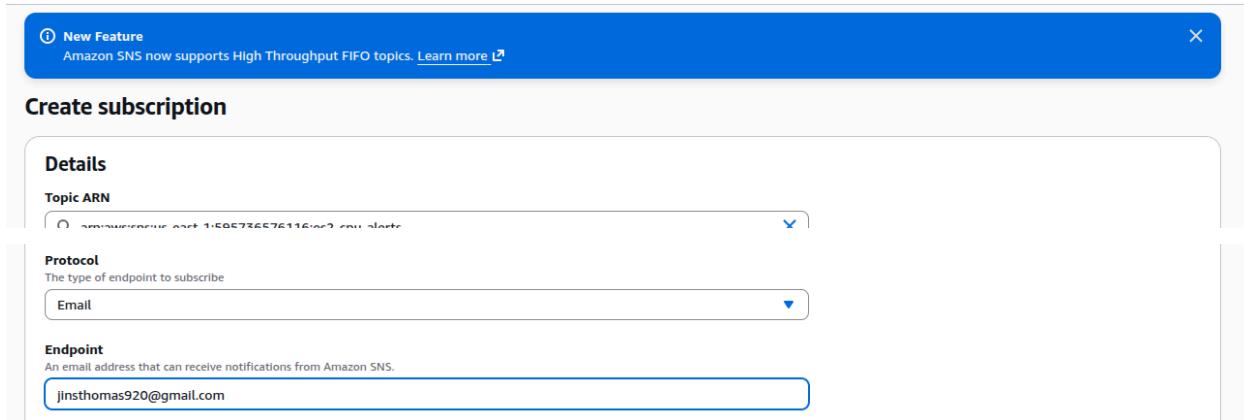
```
jins@jins-Lenovo-G50-80:~$ chmod 400 /home/jins/Downloads/my-web-server.pem
jins@jins-Lenovo-G50-80:~$ ssh -i /home/jins/Downloads/my-web-server.pem ubuntu@44.223.65.64
The authenticity of host '44.223.65.64 (44.223.65.64)' can't be established.
ED25519 key fingerprint is SHA256:MN1m8CxyCHjyxU2segpDg87eC026nov0aa6CGD3DC68.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.223.65.64' (ED25519) to the list of known hosts.
```

1.Create SNS topic using console and create subscription

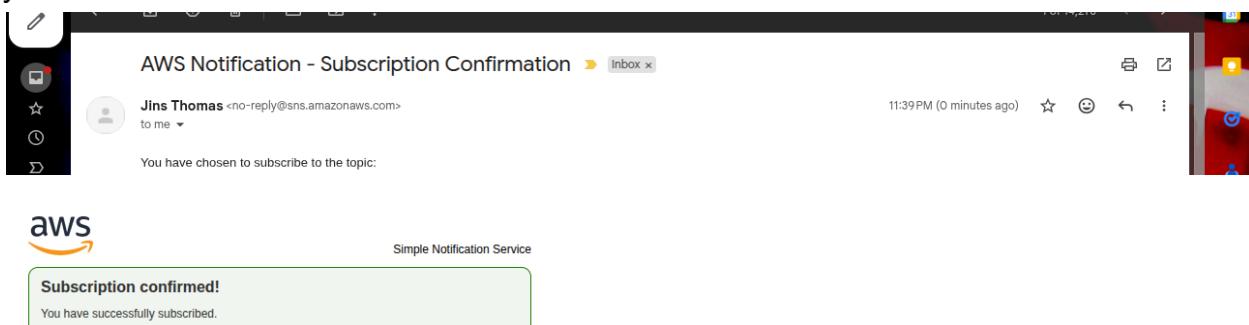


The screenshot shows the 'Create topic' step of the AWS SNS wizard. The left side has a title 'Amazon Simple Notification Service' and a subtitle 'Pub/sub messaging for microservices and serverless applications.' The right side is a form with a 'Topic name' field containing 'ec2-cpu-alerts'. Below the field is a description: 'A topic is a message channel. When you publish a message to a topic, it fans out the message to all subscribed endpoints.' There are two buttons: 'Next step' and 'Start with an overview'.

While creating a subscription select protocol i.e the communication method. I have selected email and provided my email as the endpoint.

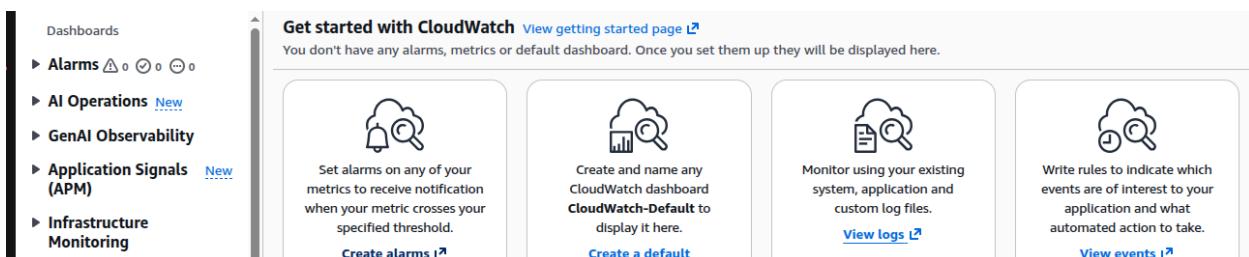


Once the subscription is created make sure you confirm the subscription received in your mail. Else AWS will not send the alert notification.

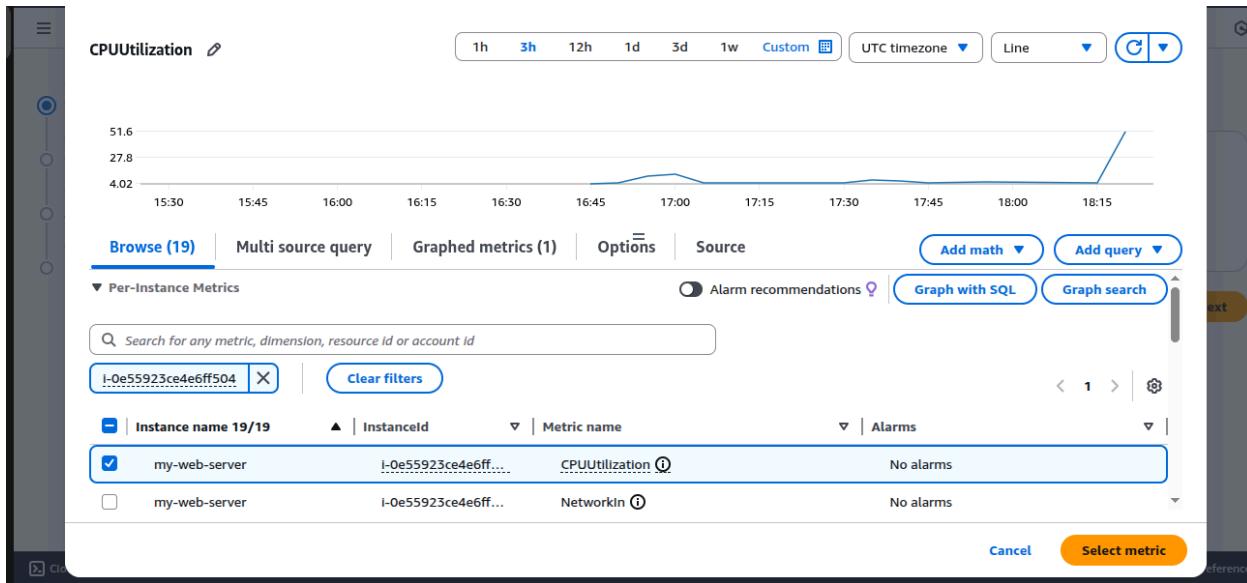


2.Creating the cloudwatch alarm

Select create alarm



Select the EC2 instance by identifying the instance id and select the desired metric. In this case i have selected cpu utilization of my EC2 instance(my-web-server).



Next we will specify the metric condition. The metric name in our case will be CPU utilization select statistics as average.

The screenshot shows the "Create alarm" wizard in the AWS CloudWatch Metrics console, currently on Step 2: "Specify metric and conditions". The left sidebar shows steps: Step 2 (selected), Step 3, Step 4, and Preview and create. The main area is titled "Specify metric and conditions". It includes a "Metric" section with a graph showing CPUUtilization for instance "i-0e55923ce4e6ff504". The graph has a red threshold line at approximately 51.6. To the right of the graph, there are fields for "Namespace" (set to "AWS/EC2"), "Metric name" (set to "CPUUtilization"), "InstanceId" (set to "i-0e55923ce4e6ff504"), "Instance name" (set to "my-web-server"), and "Statistic" (set to "Average"). There are "Edit" and "Next Step" buttons at the bottom right. The footer includes links for CloudShell, Feedback, Console Mobile App, and navigation links like Privacy, Terms, and Cookie preferences.

Now we will define the conditions for firing our alarm to receive an email notification on our endpoint which is my email w.r.t high CPU utilization for our configured instance. So in our case it will be fired when the cpu utilization is ≥ 50 (this condition is just for hands-on practice and exploring the service).

The screenshot shows the 'Conditions' step of creating an alarm. Under 'Threshold type', 'Static' is selected. Below it, 'Whenever CPUUtilization is...' defines the condition as 'Greater/Equal >= threshold' with the value set to '50'. Other options like 'Anomaly detection' and 'Lower/Equal <= threshold' are shown but not selected.

Once we are done with alarm creation we will now configure the actions for our alarm. So in the notification part in alarm we select an SNS topic that we have already created by the name ec2-cpu-alerts.

The screenshot shows the 'Configure actions' step. The left sidebar shows steps 2 through 4: 'Specify metric and conditions', 'Configure actions' (selected), 'Add alarm details', and 'Preview and create'. In the main area, under 'Configure actions', the 'Notification' section is active. It shows an 'Alarm state trigger' with 'In alarm' selected. Below it, 'Send a notification to the following SNS topic' lists 'ec2-cpu-alerts' as the selected topic. Other options like 'OK' and 'Insufficient data' are shown but not selected.

Next we will provide the name and description of the alarm. I have provided High CPU Utilization and a short description that we can see in the below screenshot. After this we have successfully created the alarm.

Add alarm details

Name and description

Alarm name
High CPU Utilization

Alarm description - optional View formatting guidelines

Edit | Preview
you insatce has reached the threshold value of maximum CPU usage that is 50

Up to 1024 characters (74/1024)

CloudWatch > Alarms

Alarms (1) Hide Auto Scaling alarms **Create alarm**

Name	State	Last state update (UTC)	Conditions
High CPU Utilization	<small>Insufficient data</small>	2025-12-02 18:31:56	CPUUtilization >= 50 for 1 datapoint within 5 minutes

3. Testing the alarm by creating stress

```
ubuntu@ip-172-31-29-208: $ sudo apt install -y stress
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  stress
```

Final output

Recent alarms [Info](#)

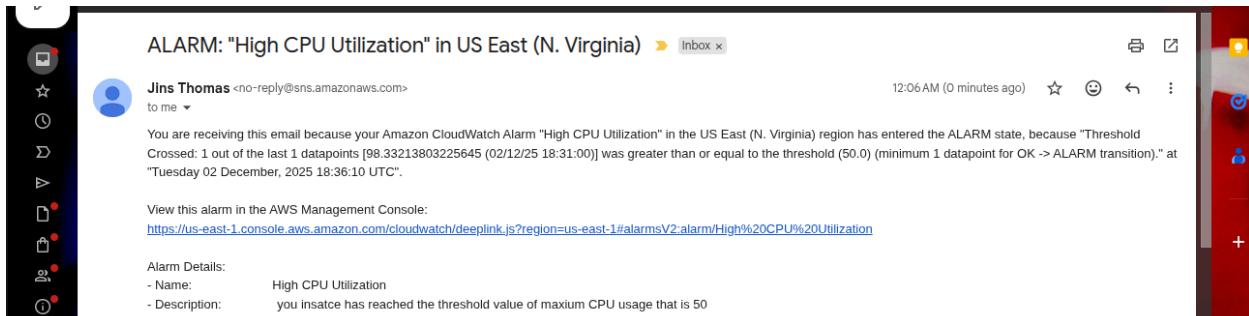
High CPU Utilization [Info](#)

Percent

51.6 CPUUtilization >= 50 for 1 datapoint within 5 minutes

27.8

4.02



Note

For faster results and testing purposes keep the threshold value less and the time period less.

