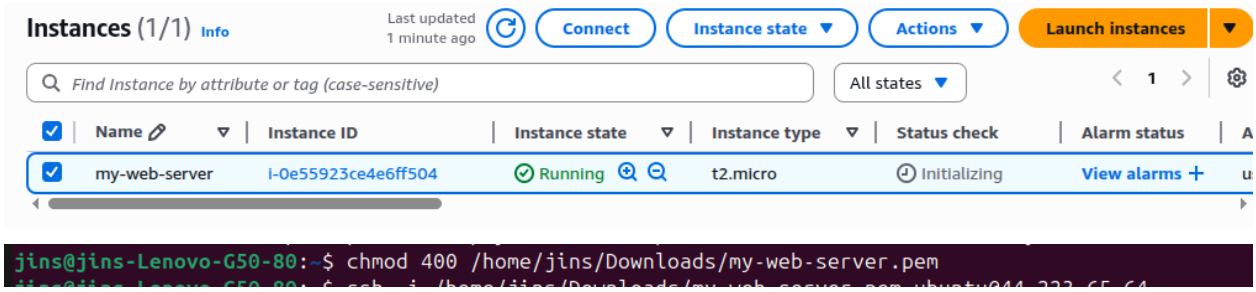
Implementation of CloudWatch CPU Alarm and SNS Email Notification

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

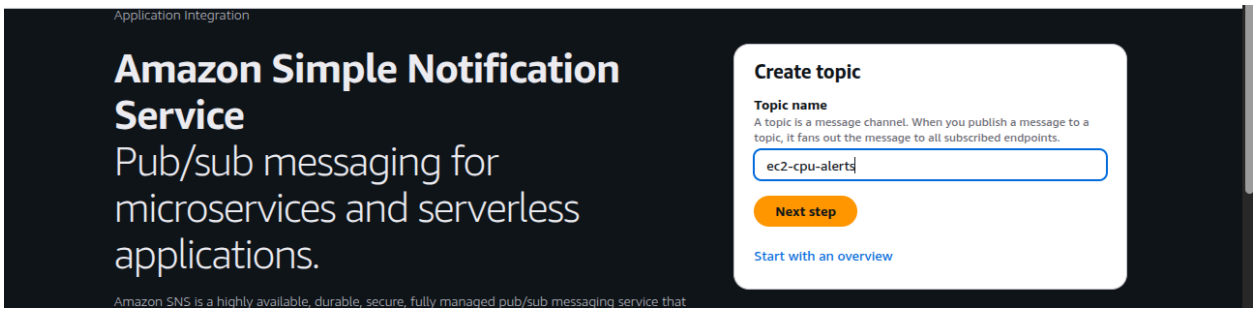
1.AWS EC2 instance (t2.micro) with port 22 allowed (allow only your IP) for ssh access and port 80 allowed (for http traffic from the internet).
2. Select OS ubuntu 24.04
3.Create key value pair for ssh access change permission to 400 for maintaining best security practice.



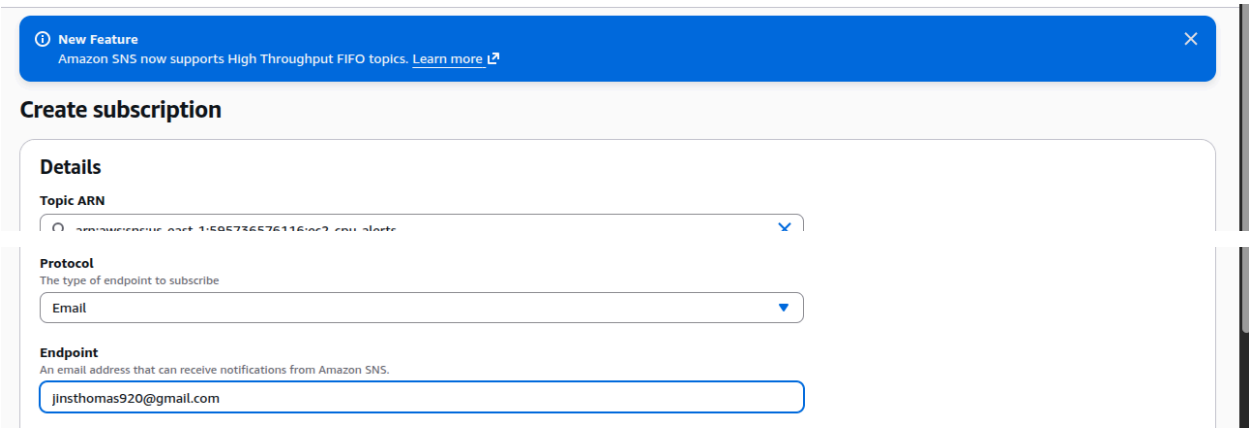

```
jins@jins-Lenovo-G50-80:~$ chmod 400 /home/jins/Downloads/my-web-server.pem
```

```
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:MN1m8CxyCHjyxU2segpDg87eCO26novOaa6CGD3DC68.
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.
```

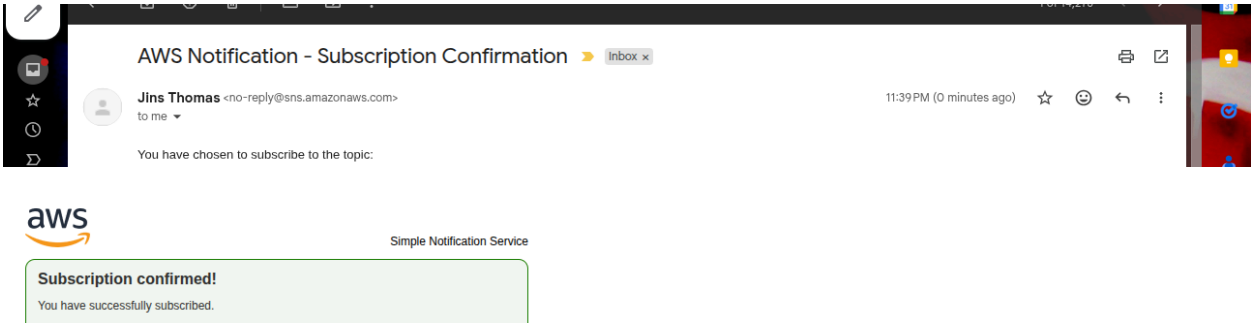
1.Create SNS topic using console and create subscription



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

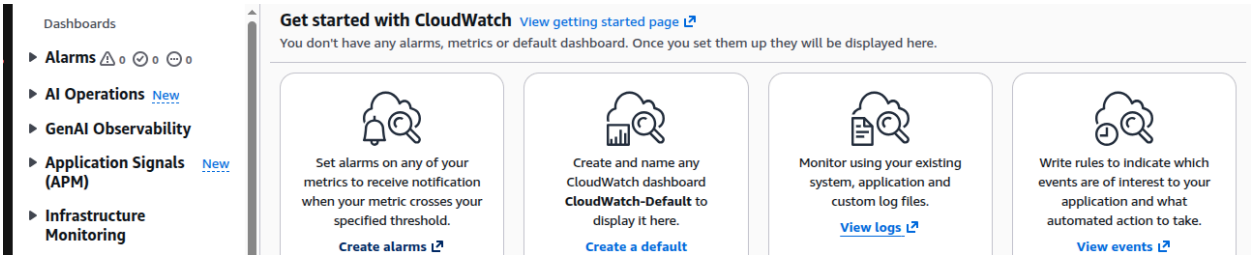


Once the subscription is created make sure you confirm the subscription received in your mail. Else AWS will not send the alert notification.

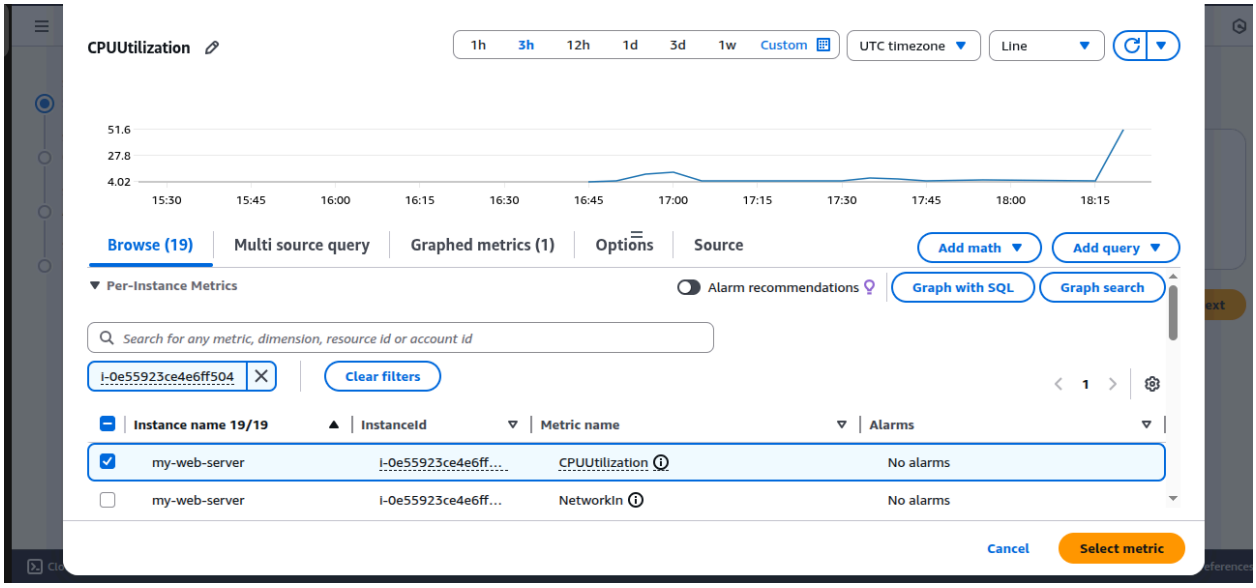


2.Creating the cloudwatch alarm

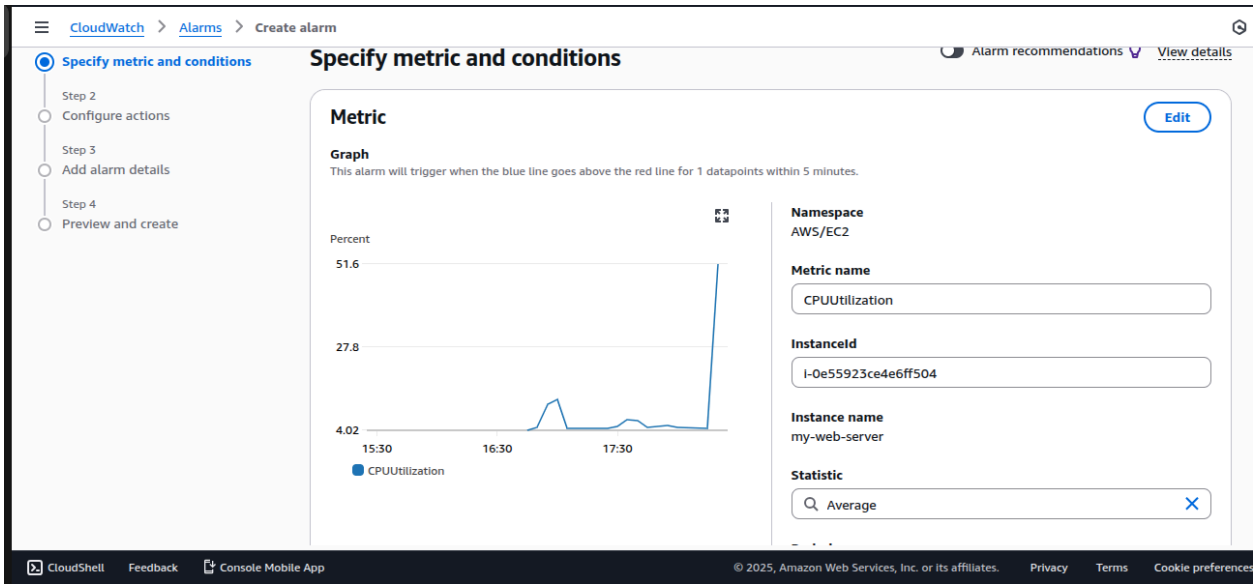
Select create alarm



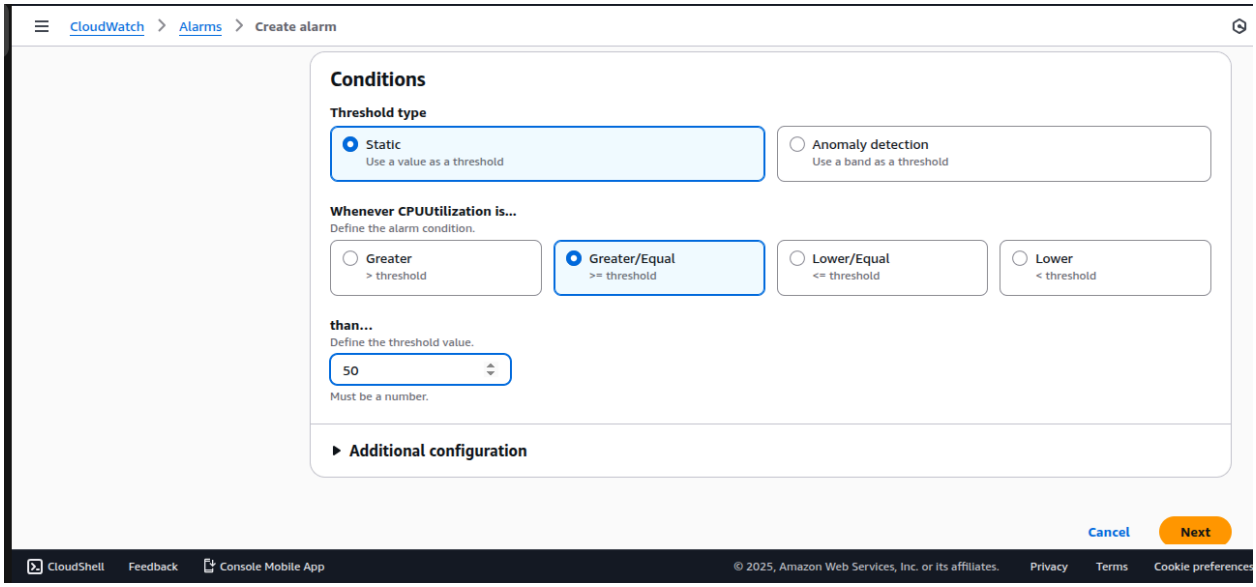
Select the EC2 instance by identifying the instance id and select the desired metric. In this case i have selected cpu utilization of my EC2 instance(my-web-server).



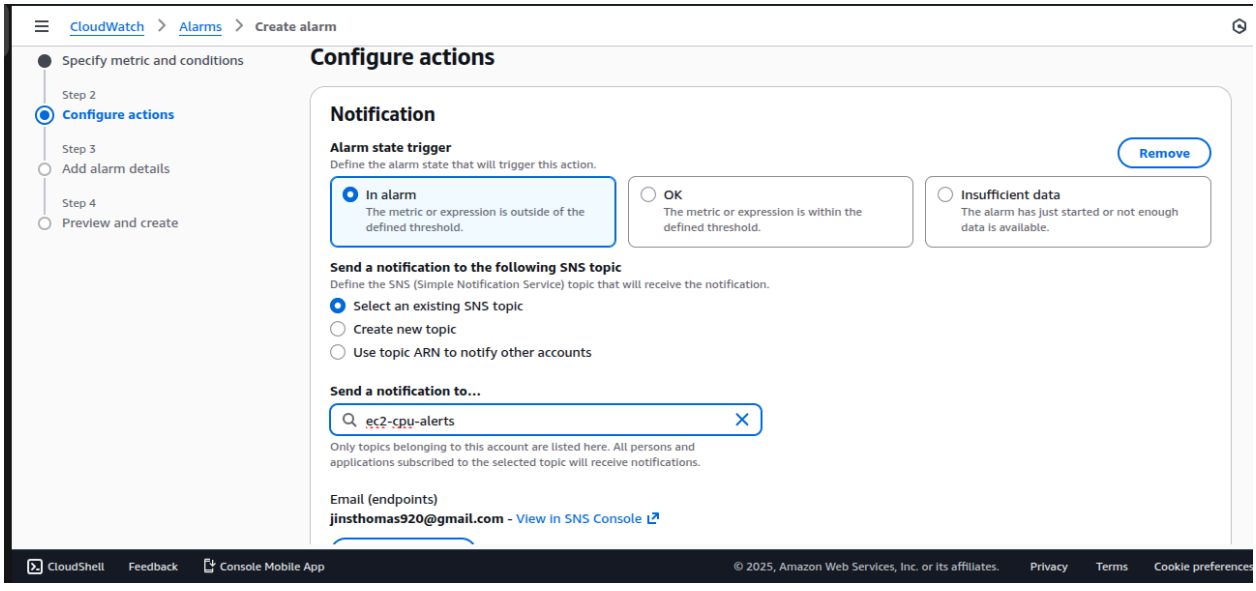
Next we will specify the metric condition. The metric name in our case will be CPU utilization select statistics as average.



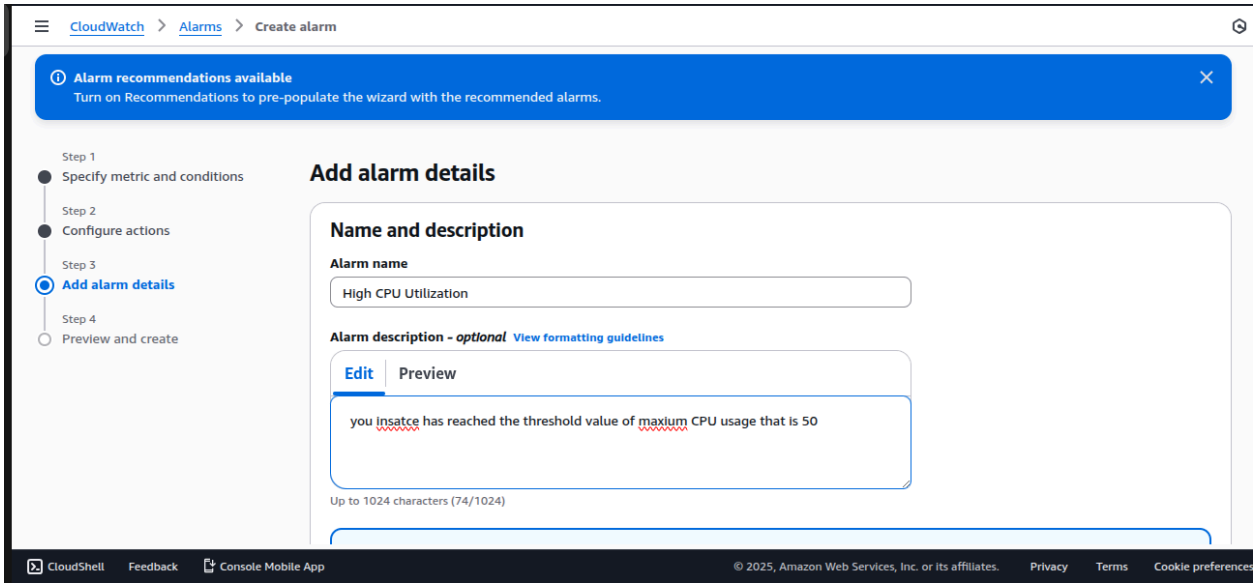
Now we will define the conditions for firing our alarm to receive an email notification on our endpoint which is my email w.r.t high CPU utilization for our configured instance. So in our case it will be fired when the cpu utilization is >= 50(this condition is just for hands-on practice and exploring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

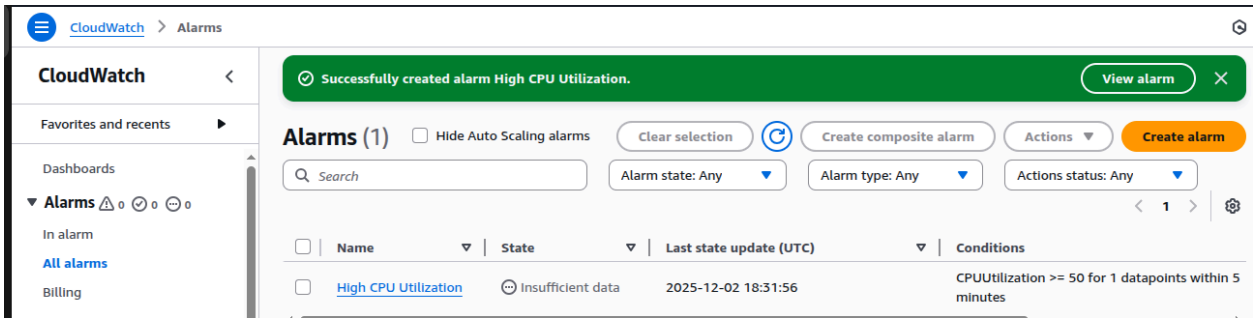


Once we are done with alarm creation we will now configure the actions for our alarm. So in the notification part in alarm we select an SNS topic that we have already created by the name ec2-cpu-alerts.



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.

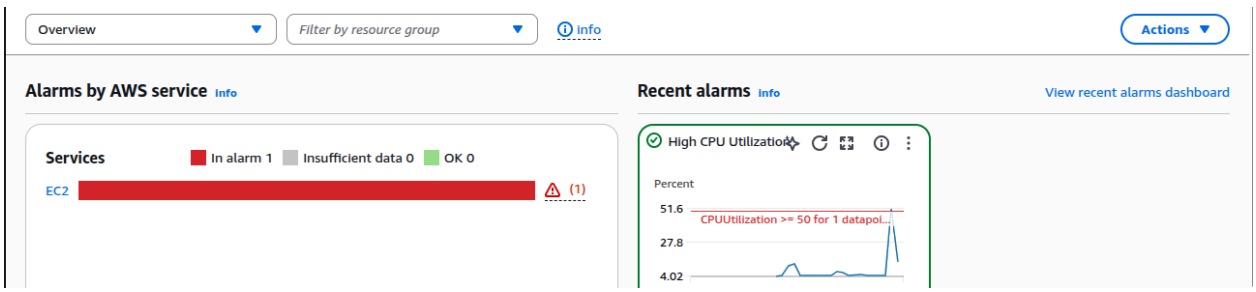


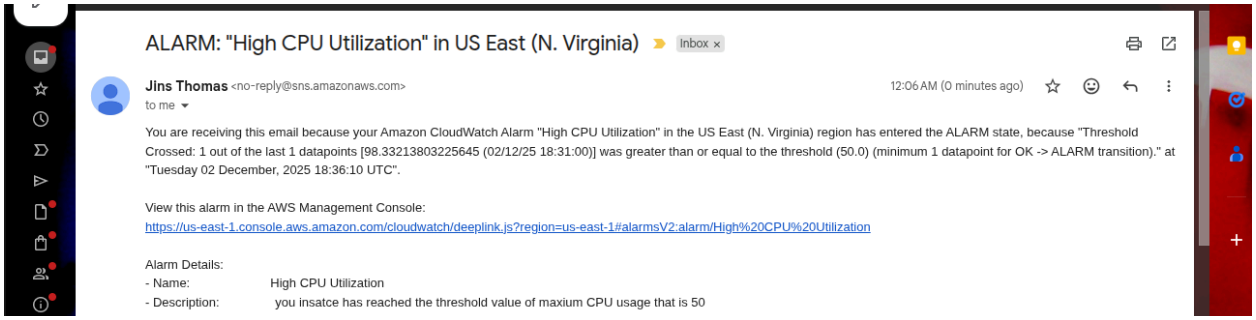


3.Testing the alarm by creating stress

```
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





Note

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