Creating a CPU usage alarm

You can create an CloudWatch alarm that sends a notification using Amazon SNS when the alarm changes state from OK to ALARM.

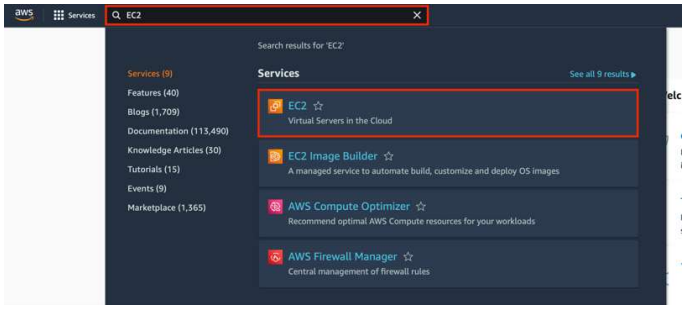
The alarm changes to the ALARM state when the average CPU use of an EC2 instance exceeds a specified threshold for consecutive specified periods.

Steps:

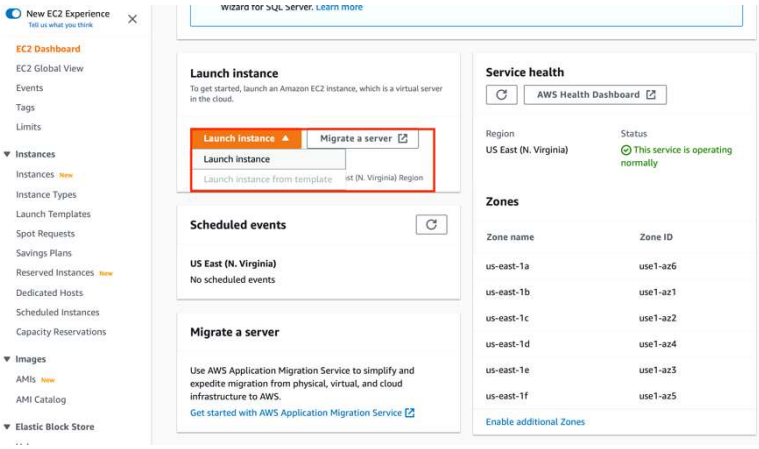
- Launch an EC2 Instance
- Set up a CPU usage alarm using the AWS Management Console
- Initiate an SSH connection to the EC2 instance
- Install the stress package and run the stress command
- Monitor the CPU Utilization of EC2 Instance via CloudWatch
- Terminate your instance

Step A: Launch an EC2 instance

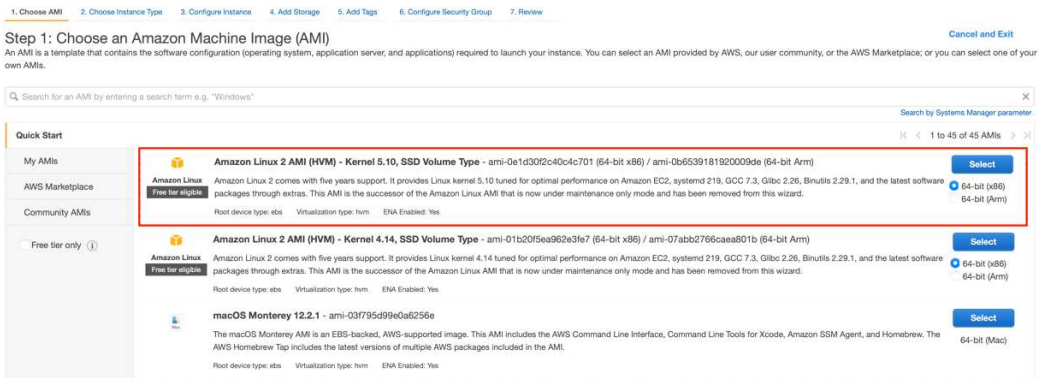
- Login to AWS Management Console and search for EC2 in the search bar. Once EC2 service shows up in the list, click and go to EC2 dashboard.



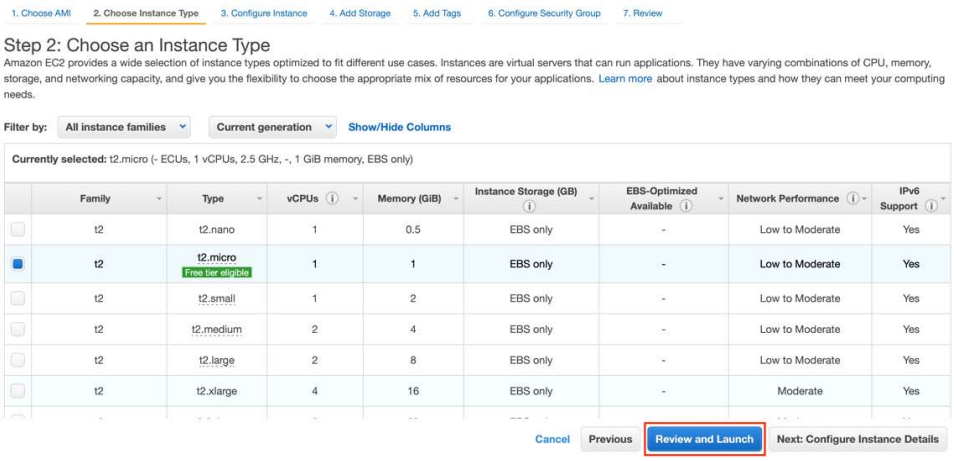
- Click Launch Instance.



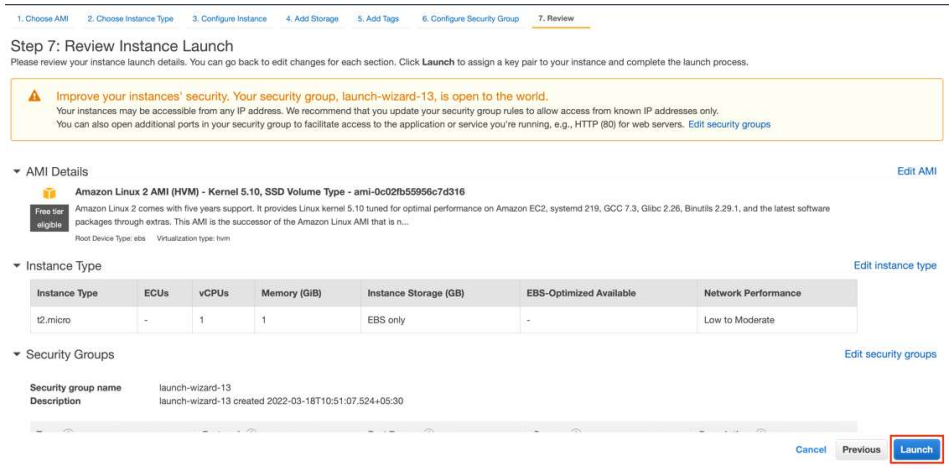
- <u>On Step 1: Choose an Amazon Machine Image (AMI), select the Amazon Linux 2 AMI.</u>



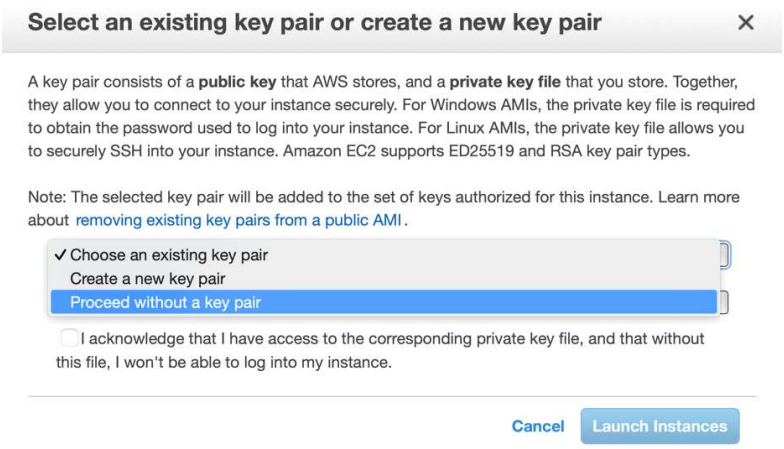
- <u>On Step 2: Choose an Instance Type, choose **t2.micro** as the instance type and click **Review and Launch**.</u>



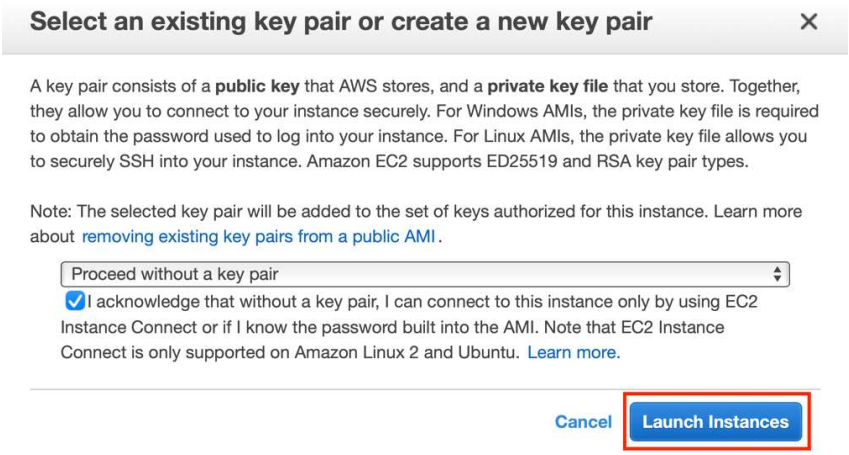
- <u>On **Step 7: Review Instance Launch**, click **Launch**.</u>



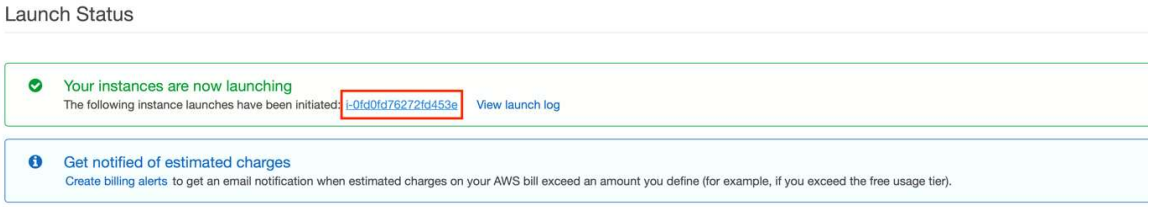
- <u>When prompted to select an existing key pair or create a new key pair, choose **Proceed without a key pair**. For this demo, access to instance via SSH is not required.</u>



- Check the acknowledgment option and click **Launch Instances** to launch the EC2 instance.

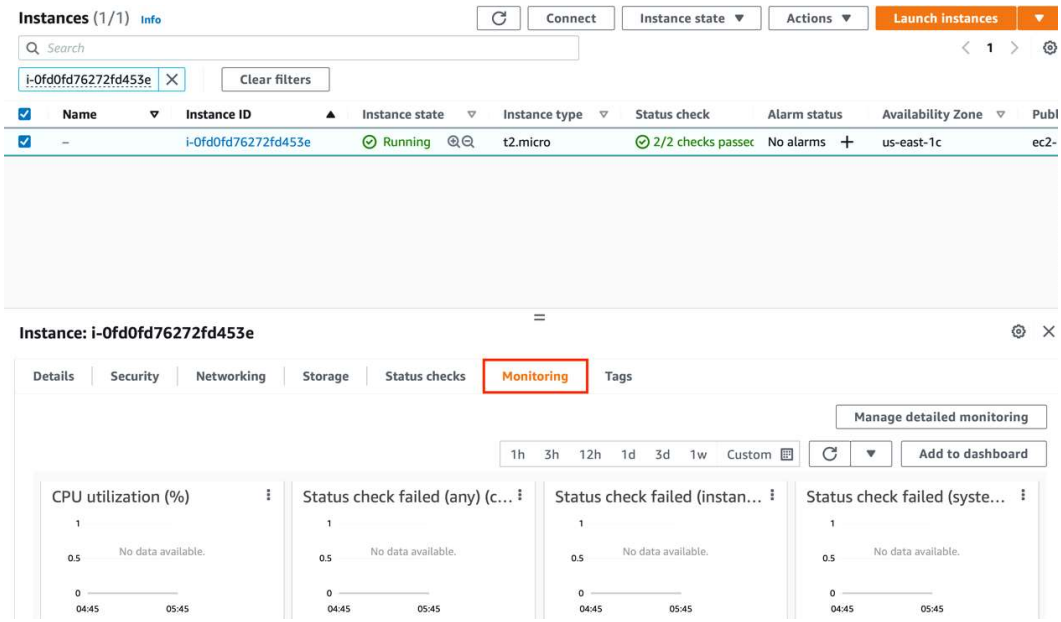


- Click on the instance ID to view the EC2 instance we just launched.

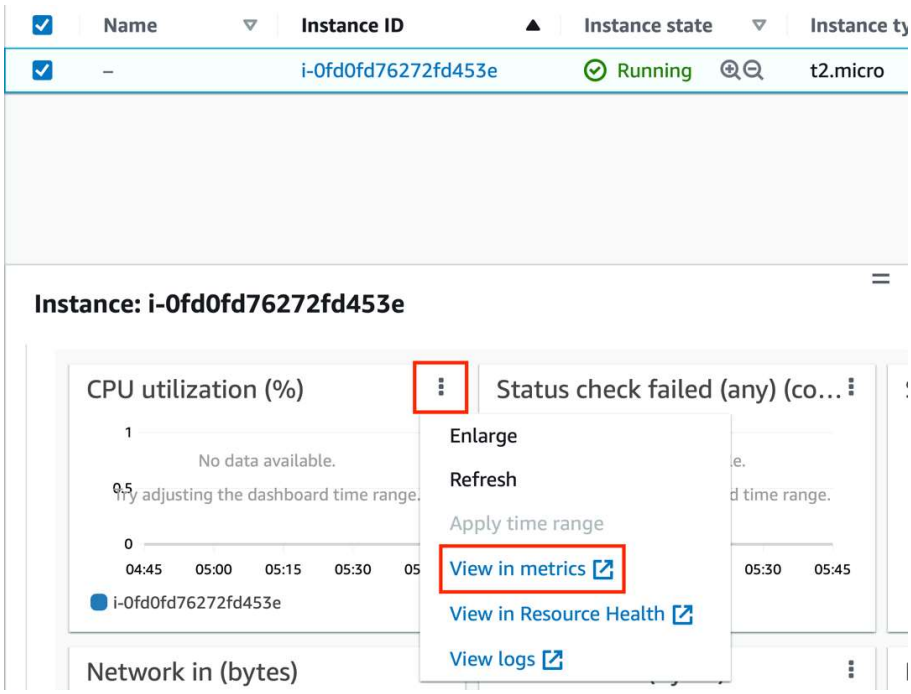


Step B: Set up a CPU usage alarm using the AWS Management Console

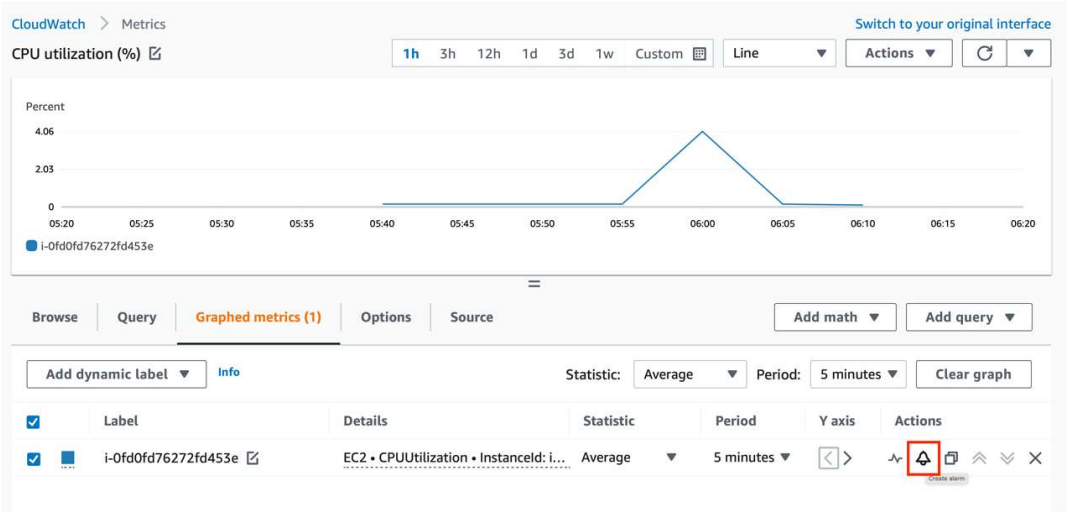
- Go to the Monitoring section to access and view all metrics based on which this instance's performance will be measured.



- Within **CPU utilization** metric, click three vertical dots and then click **View in metrics**.

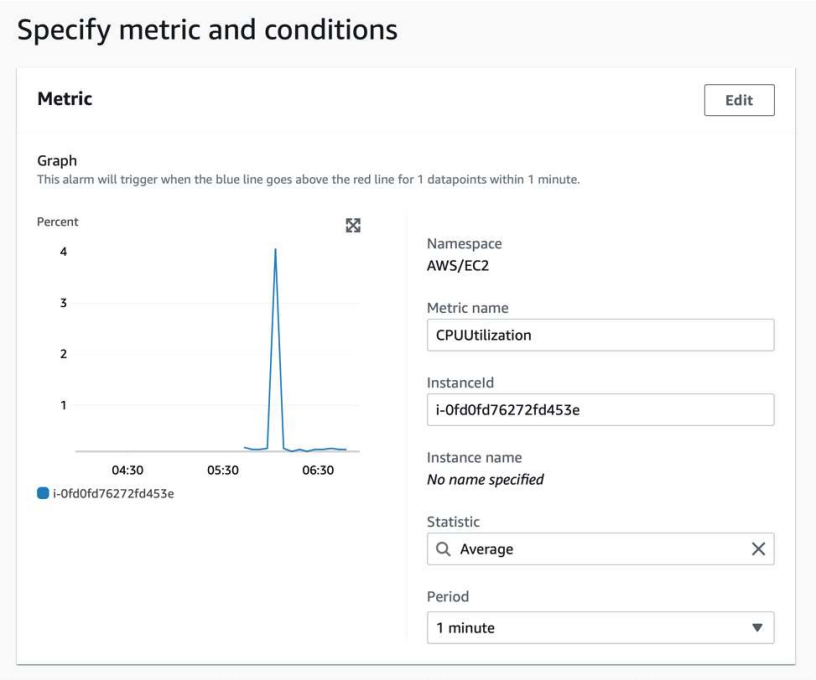


- <u>This will take you the CloudWatch dashboard. Under **Graphed metrics**, you can access and view the instance ID along with other relevant details. On this row, click on the bell icon under **Actions.**</u>



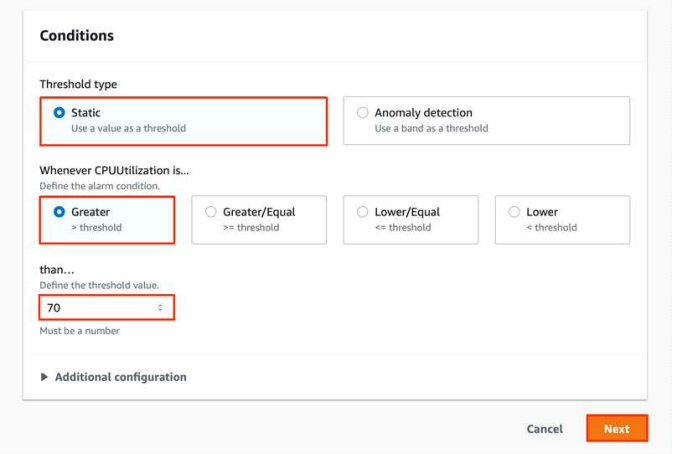
- <u>Under **Specify metric and conditions**, ensure that following values are assigned:</u>

- Metric Name = CPUUtilization
- InstanceID = Actual instance ID of the launched instance
- Statistic = Average
- Period = 1 minute



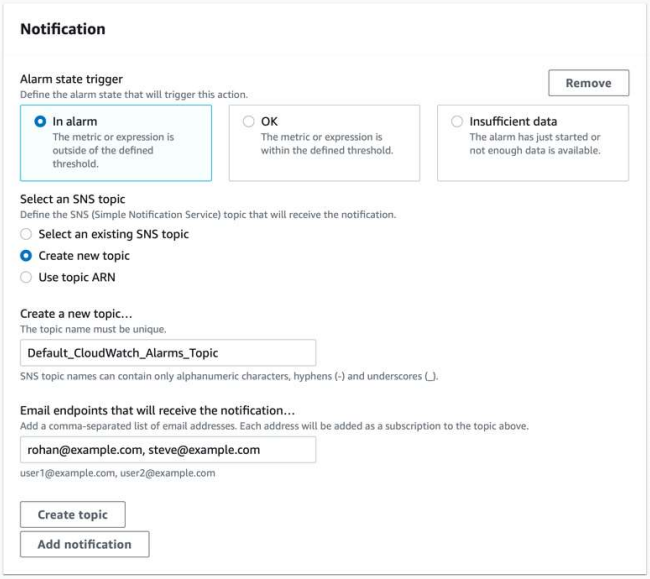
- Under **Conditions**, specify the following:

- For **Threshold type**, choose **Static**.
- For **Whenever CPUUtilization** is, specify **Greater**. Under than..., specify the threshold that is to trigger the alarm to go to ALARM state if the CPU utilization exceeds this percentage. For example, 70.
- Choose **Next**.



- Under **Notification**, choose **In alarm** and create an SNS topic to notify when the alarm is in *ALARM* state.

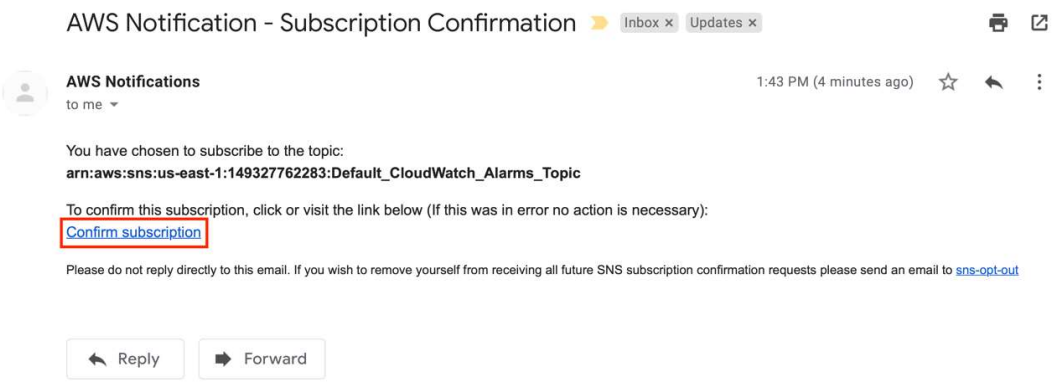
- For **Select an SNS topic**, choose **Create a new topic**.
- Within **Create a new topic…**, specify the SNS topic name or keep it to default.
- Within **Email endpoints that will receive the notification…**, add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.
- Click **Create Topic**.



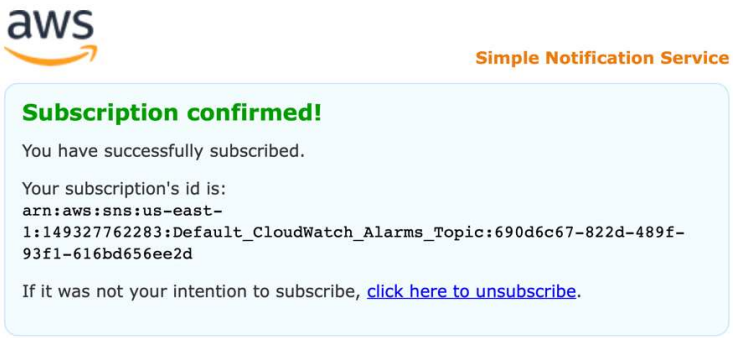
- You must confirm the subscription before notifications can be sent to an email address.

    - Check your inbox and look for subscription e-mail received.

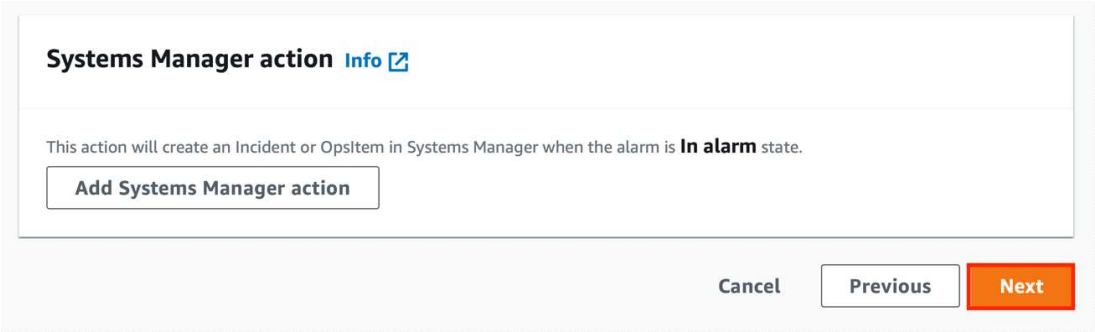
- Open message and click **Confirm Subscription**.



- Once successfully subscribed, you will be sent to subscription confirmation page.

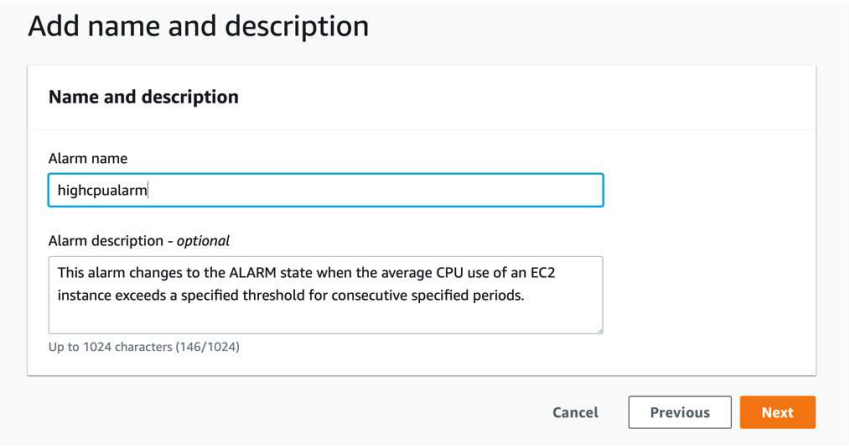


- Go back to CloudWatch Management Console and choose **Next**.

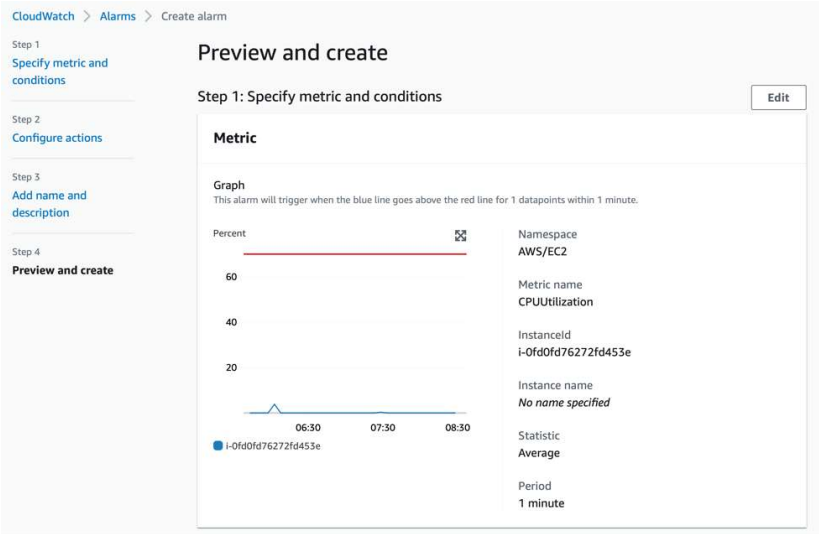


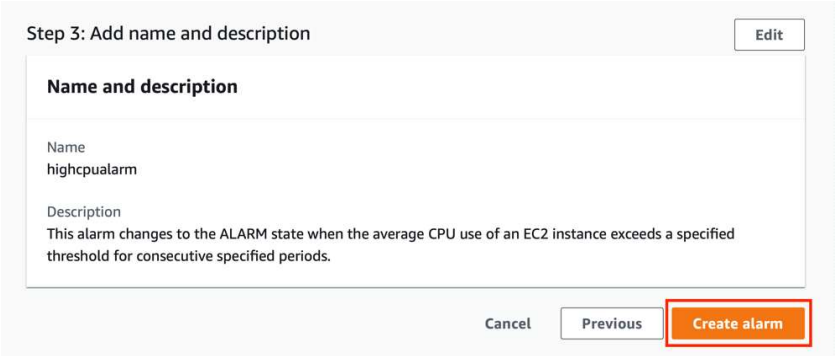
- Enter a name and description for the alarm. The name must contain only ASCII

characters. Then choose **Next**.

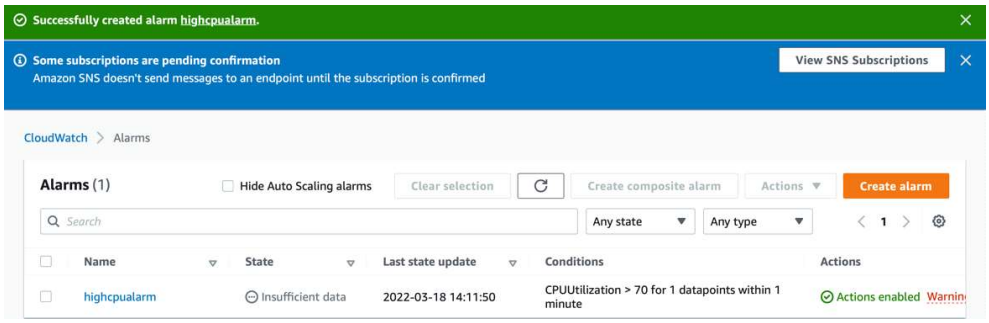


- Under **Preview and create**, confirm that the information and conditions are what you want, then choose **Create alarm**.



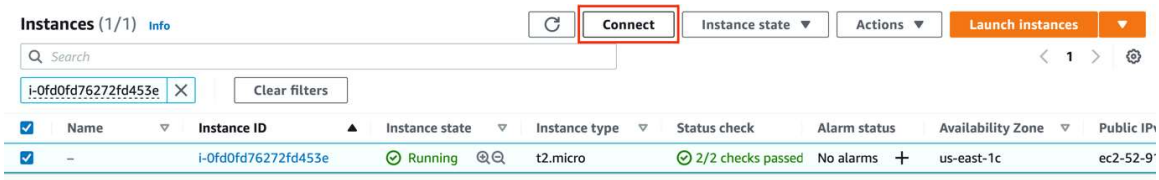


- This will complete the process of alarm creation on CloudWatch.

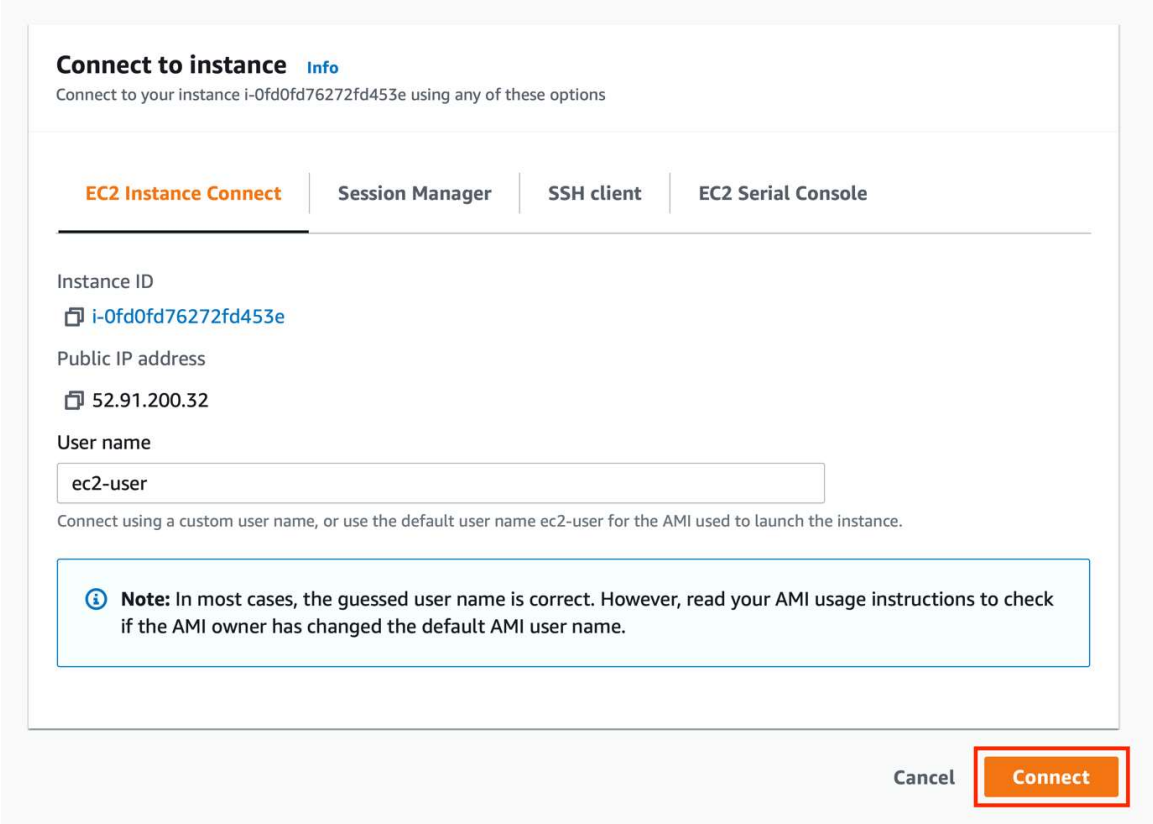


Step C: Initiate an SSH connection to EC2 instance

- Go back to instance dashboard, choose the instance, and click **Connect**.



- Under EC2 Instance Connect, click **Connect**.



Step D: Install the stress utility and run the stress command

- Run following commands to install stress utility:

- sudo amazon-linux-extras install epel -y
- sudo yum install stress -y

```
Last login: Fri Mar 18 12:12:05 2022 from ec2-18-206-107-25.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-80-36 ~]$ sudo amazon-linux-extras install epel -y
Installing epel-release
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-epel amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                              | 3.7 kB  00:00:00
amzn2extra-docker                                                       | 3.0 kB  00:00:00
amzn2extra-epel                                                         | 3.0 kB  00:00:00
amzn2extra-kernel-5.10                                                  | 3.0 kB  00:00:00
(1/9): amzn2-core/2/x86_64/group_gz                                     | 2.5 kB  00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo                                   | 452 kB  00:00:00
(3/9): amzn2extra-epel/2/x86_64/primary_db                              | 1.8 kB  00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo                       |  12 kB  00:00:00
(5/9): amzn2extra-docker/2/x86_64/updateinfo                            | 5.9 kB  00:00:00
```

i-0fd0fd76272fd453e

Public IPs: 52.91.200.32 Private IPs: 172.31.80.36

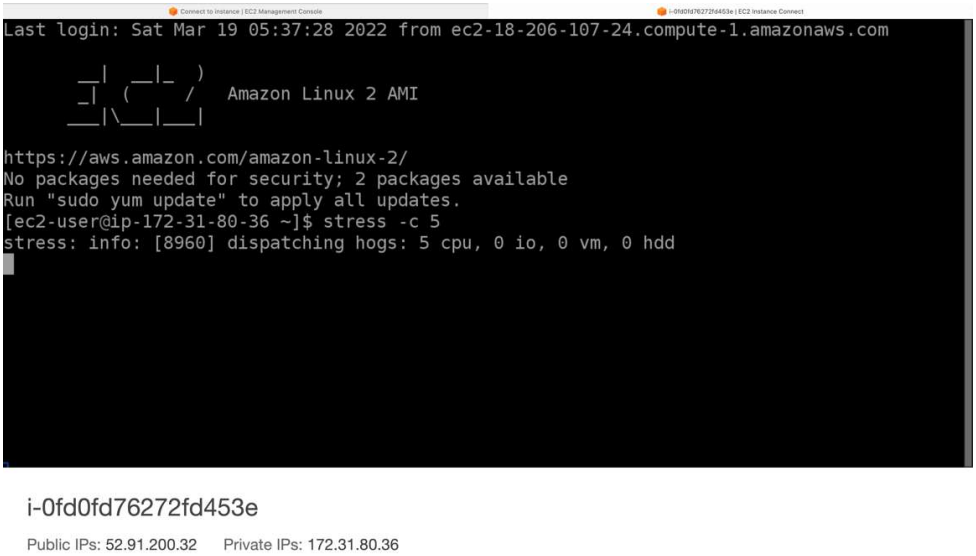
```
 59  postgresql13             available    [ =stable ]
 60  mock2                    available    [ =stable ]
 61  dnsmasq2.85              available    [ =stable ]
[ec2-user@ip-172-31-80-36 ~]$ sudo yum install stress -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
209 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package stress.x86_64 0:1.0.4-16.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package               Arch                Version                   Repository          Size
================================================================================
Installing:
 stress                x86_64              1.0.4-16.el7              epel                39 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 39 k
```

i-0fd0fd76272fd453e

Public IPs: 52.91.200.32 Private IPs: 172.31.80.36

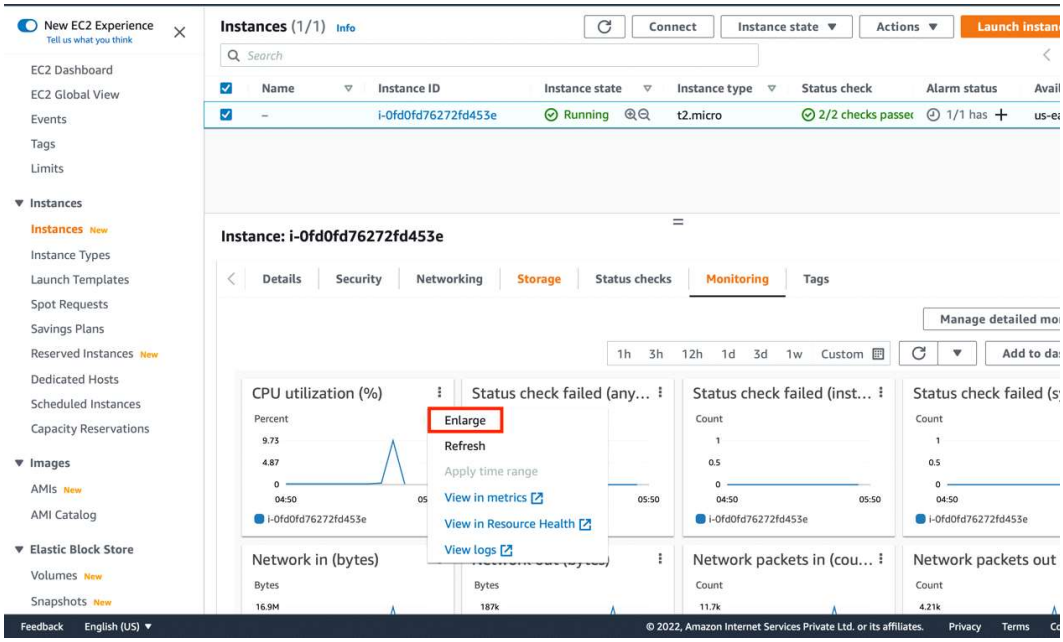
- Now, run the following command to impose stress to spike CPU utilization of the instance.

- stress -c 5

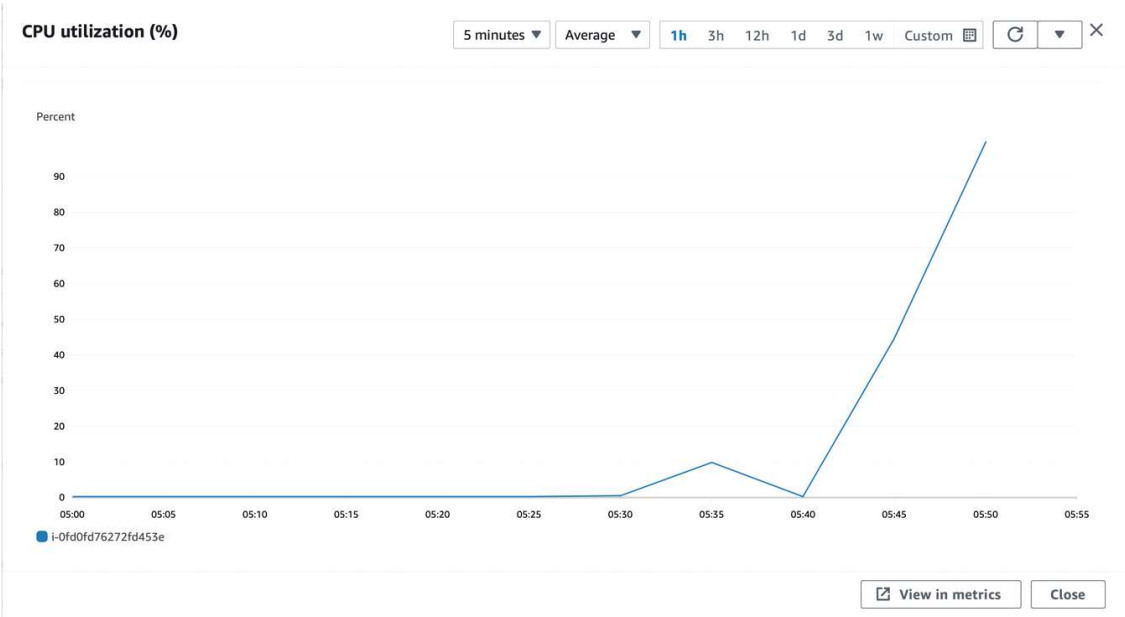


Step E: Monitor the CPU Utilization of EC2 Instance via CloudWatch

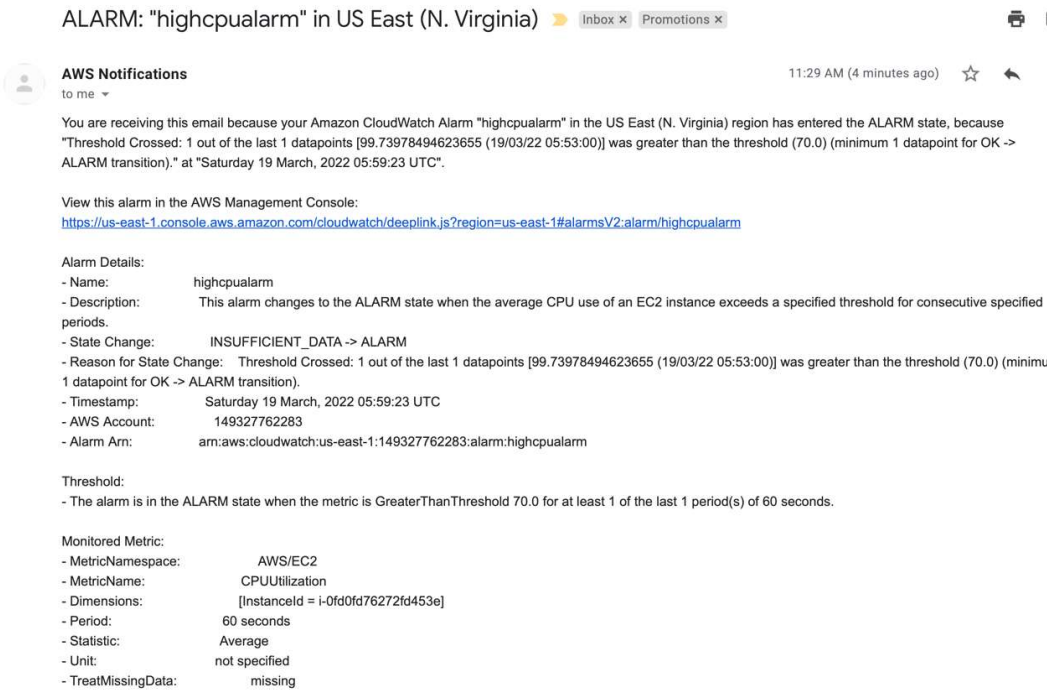
- Go back to EC2 Management Console, choose the EC2 instance, go to **Monitoring**, click three vertical dots on **CPU Utilization** and click **Enlarge**.



- In next few minutes you will see a spike in CPU utilization.



- Additionally, you can check your mailbox as you will receive an e-mail for this alarm generated via CloudWatch.



ALARM: "highcpualarm" in US East (N. Virginia) Inbox × Promotions ×

AWS Notifications 11:29 AM (4 minutes ago)
to me

You are receiving this email because your Amazon CloudWatch Alarm "highcpualarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [99.73978494623655 (19/03/22 05:53:00)] was greater than the threshold (70.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Saturday 19 March, 2022 05:59:23 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/highcpualarm

Alarm Details:
- Name: highcpualarm
- Description: This alarm changes to the ALARM state when the average CPU use of an EC2 instance exceeds a specified threshold for consecutive specified periods.
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [99.73978494623655 (19/03/22 05:53:00)] was greater than the threshold (70.0) (minimu 1 datapoint for OK -> ALARM transition).
- Timestamp: Saturday 19 March, 2022 05:59:23 UTC
- AWS Account: 149327762283
- Alarm Arn: arn:aws:cloudwatch:us-east-1:149327762283:alarm:highcpualarm

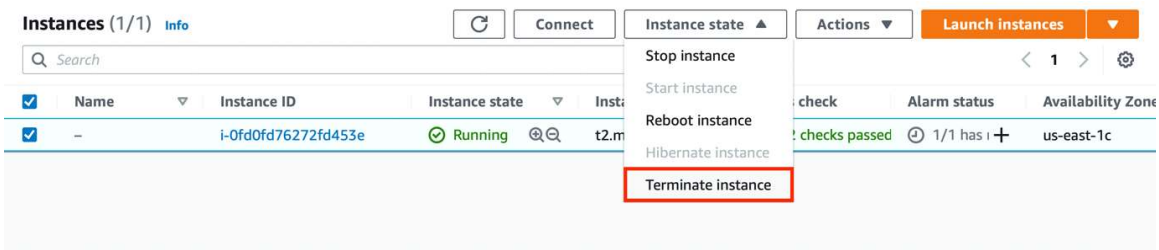
Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 70.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [InstanceId = i-0fd0fd76272fd453e]
- Period: 60 seconds
- Statistic: Average
- Unit: not specified
- TreatMissingData: missing

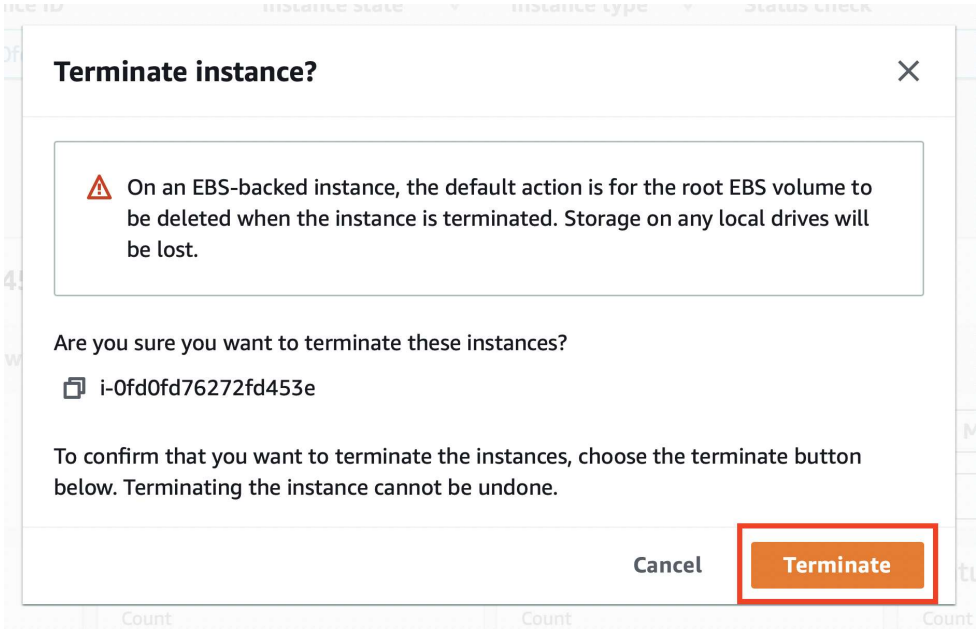
Step F: Terminate your instance

If you launched an instance that is not within the AWS Free Tier, you'll stop incurring charges for that instance as soon as the instance status changes to shutting down or terminated. To keep your instance for later, but not incur charges, you can stop the instance now and then start it again later.

- In the navigation pane, choose Instances. In the list of instances, select the instance. Choose **Instance state**, **Terminate instance**.



- Choose **Terminate** when prompted for confirmation.



Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is automatically deleted.

You cannot remove the terminated instance from the console display yourself.