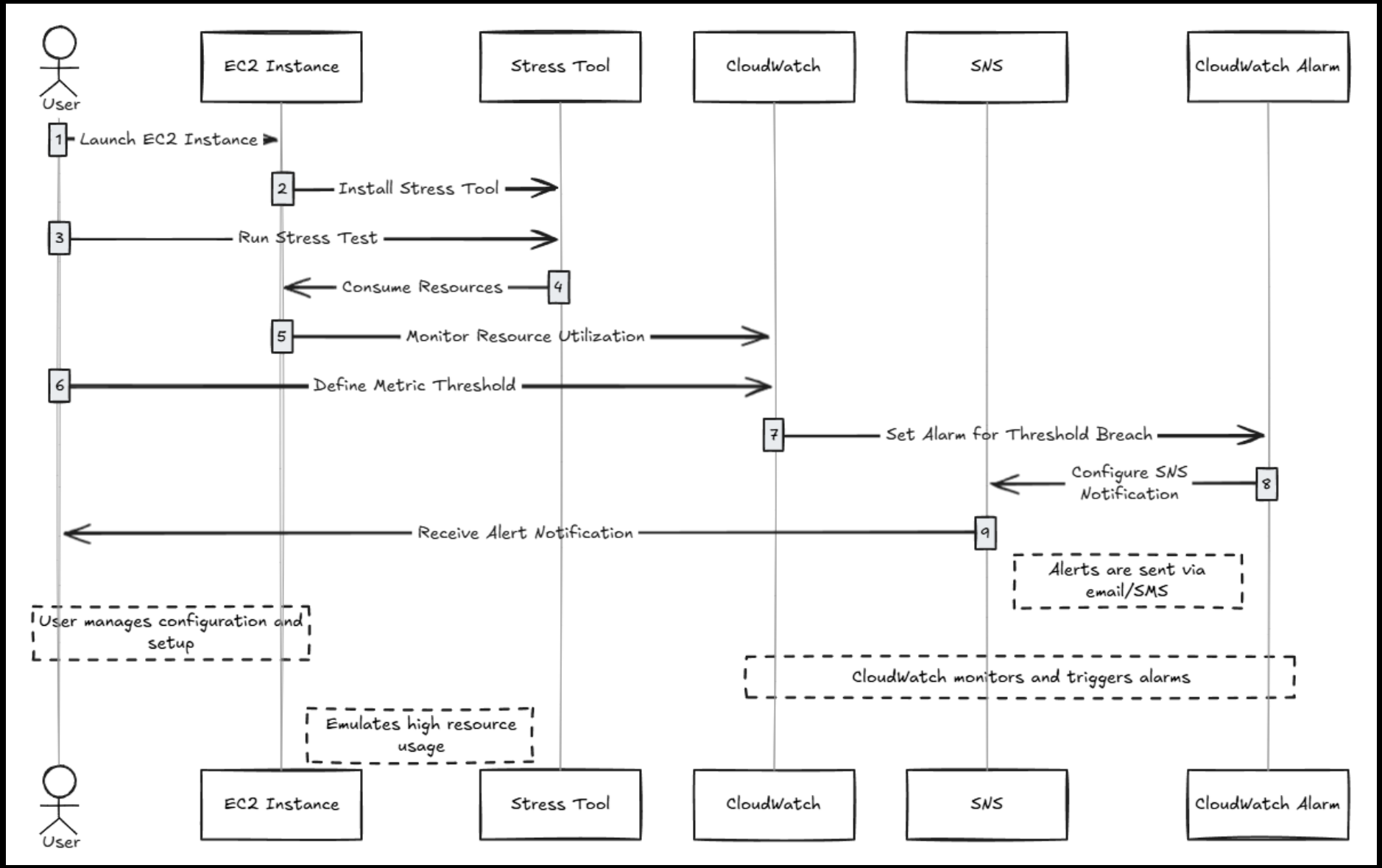

User
EC2 Instance
Stress Tool
CloudWatch
SNS
CloudWatch Alarm
1 Launch EC2 Instance
2 Install Stress Tool
3 Run Stress Test
Consume Resources 4
5 Monitor Resource Utilization
6 Define Metric Threshold
7 Set Alarm for Threshold Breach
Configure SNS Notification 8
Receive Alert Notification 9
Alerts are sent via email/SMS
User manages configuration and setup
CloudWatch monitors and triggers alarms
Emulates high resource usage
User
EC2 Instance
Stress Tool
CloudWatch
SNS
CloudWatch Alarm

PROJECT DOCUMENTATION: CREATING A STRESS ENVIRONMENT ON EC2 AND ENABLING CLOUDWATCH ALARM WITH SNS NOTIFICATION 🚀 🔥

BY KHUSHI NANDWANI

📢 INTRODUCTION

This project demonstrates how to create a stress environment on an AWS EC2 instance, set up CloudWatch alarms to monitor CPU utilization, and use Amazon SNS (Simple Notification Service) to trigger alerts. By following these steps, you will learn how to monitor EC2 performance, receive alerts when CPU usage exceeds a defined threshold, and effectively use AWS services for infrastructure monitoring.

🔧 PREREQUISITES

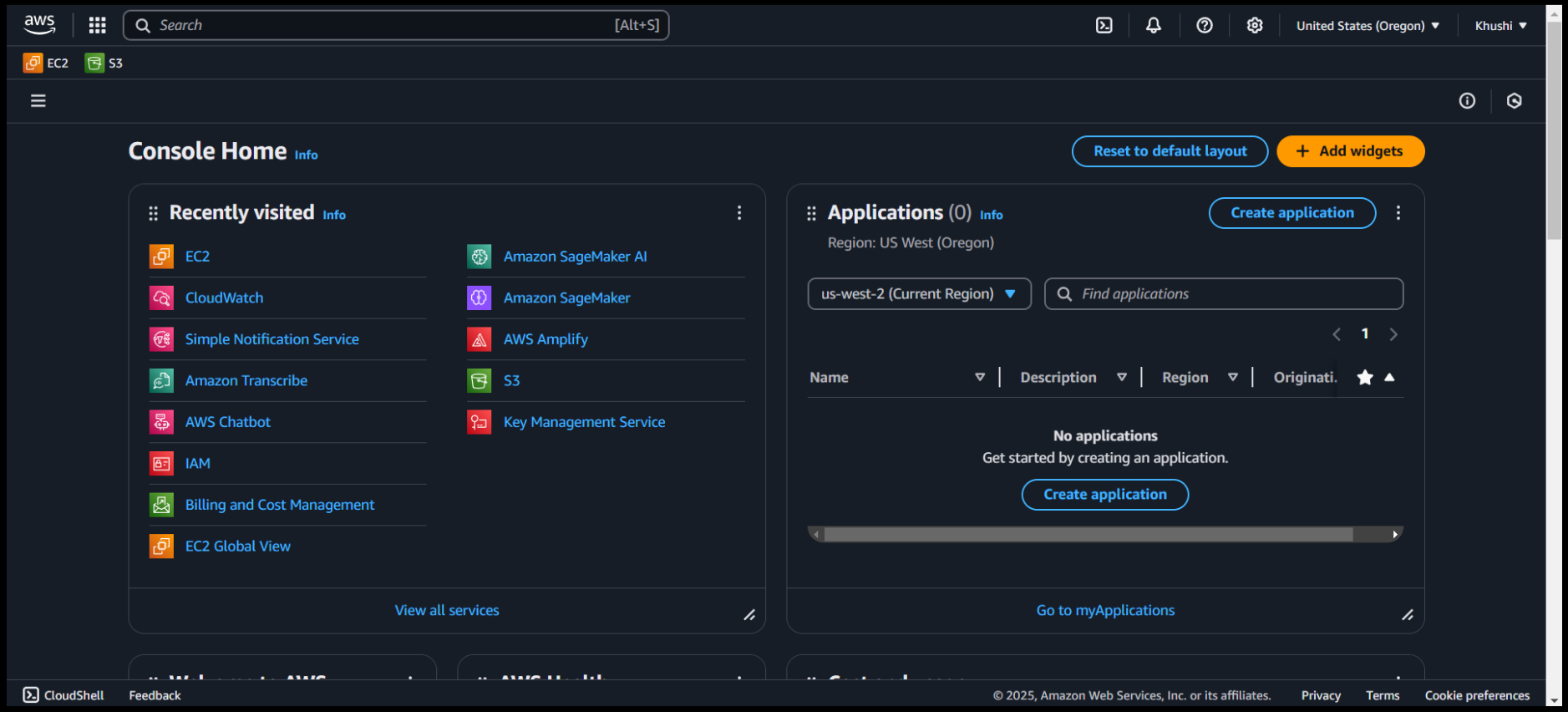
Before starting, ensure you have the following:

- 🏢 An AWS account
- 🌐 Access to the AWS Management Console
- ☁️ Basic understanding of AWS services (EC2, CloudWatch, SNS)
- 🐧 Familiarity with Linux commands

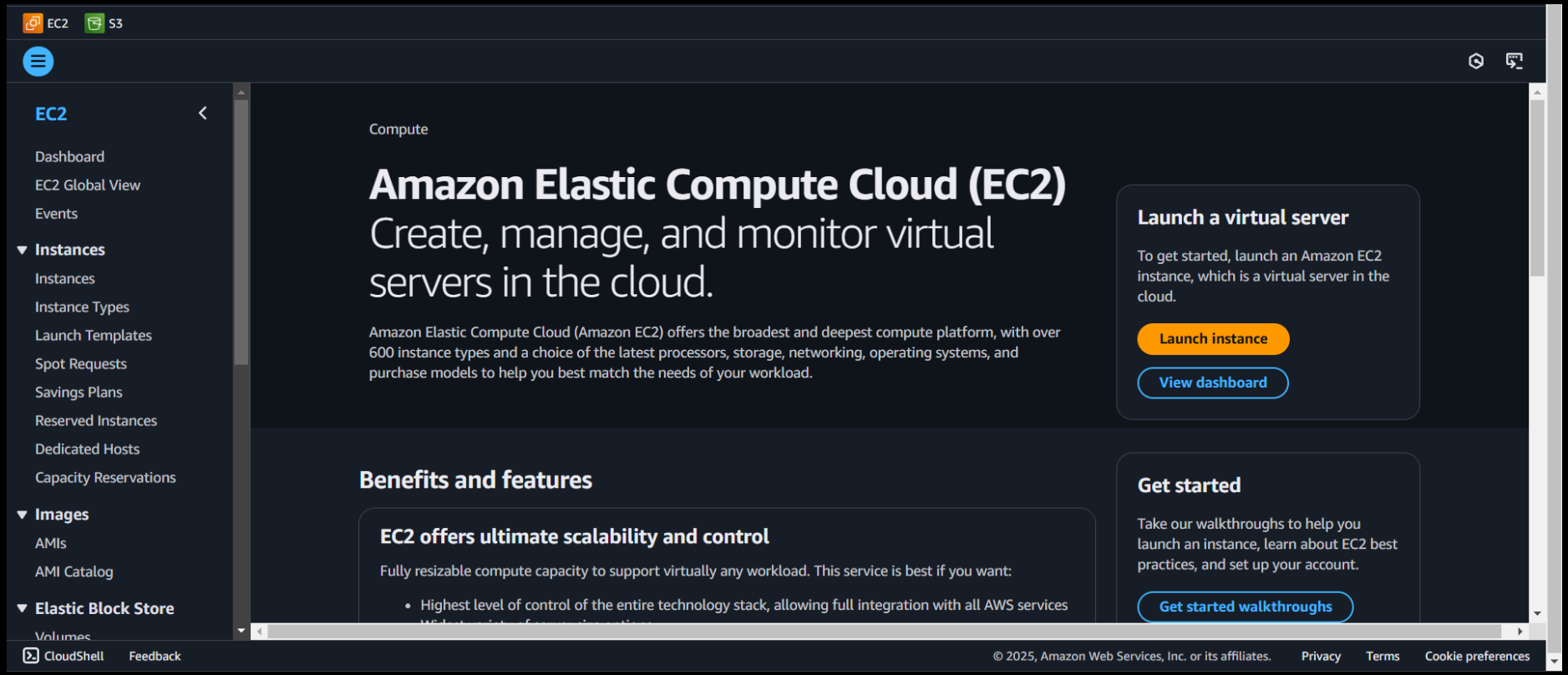
🚀 IMPLEMENTATION STEPS

I. SETTING UP THE EC2 INSTANCE 🖥️

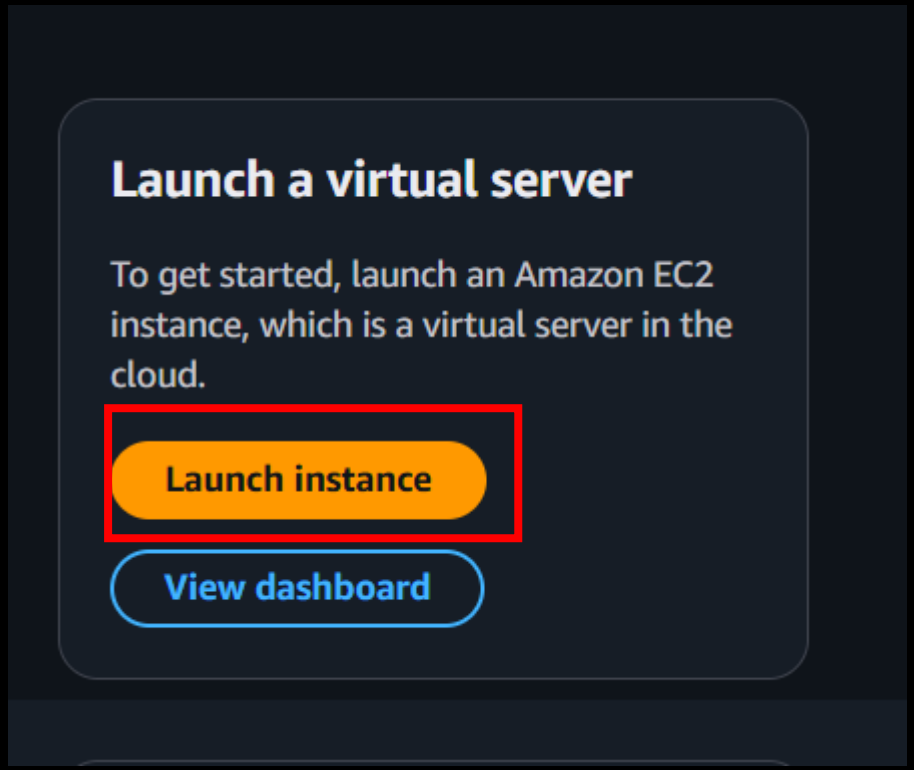
1. Open **AWS Management Console.**



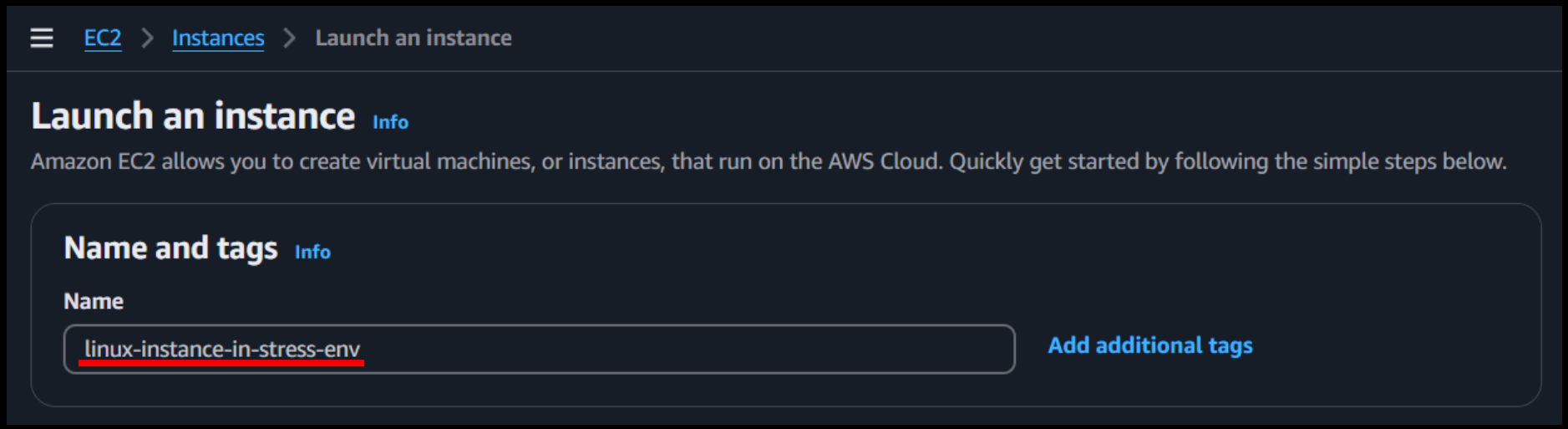
2. Search for **EC2** and navigate to the **EC2 dashboard.**



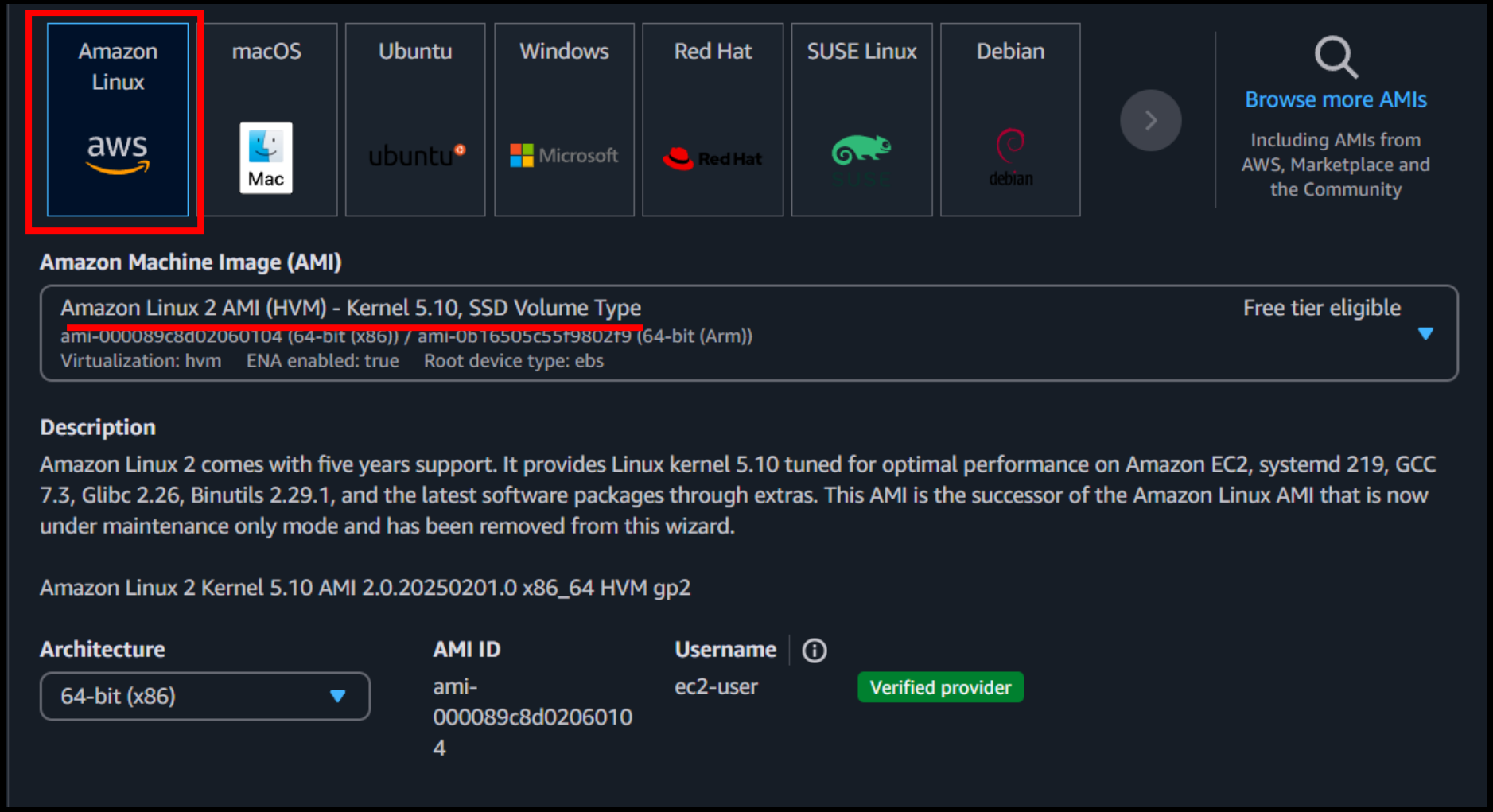
3. Click on **Launch Instance.**



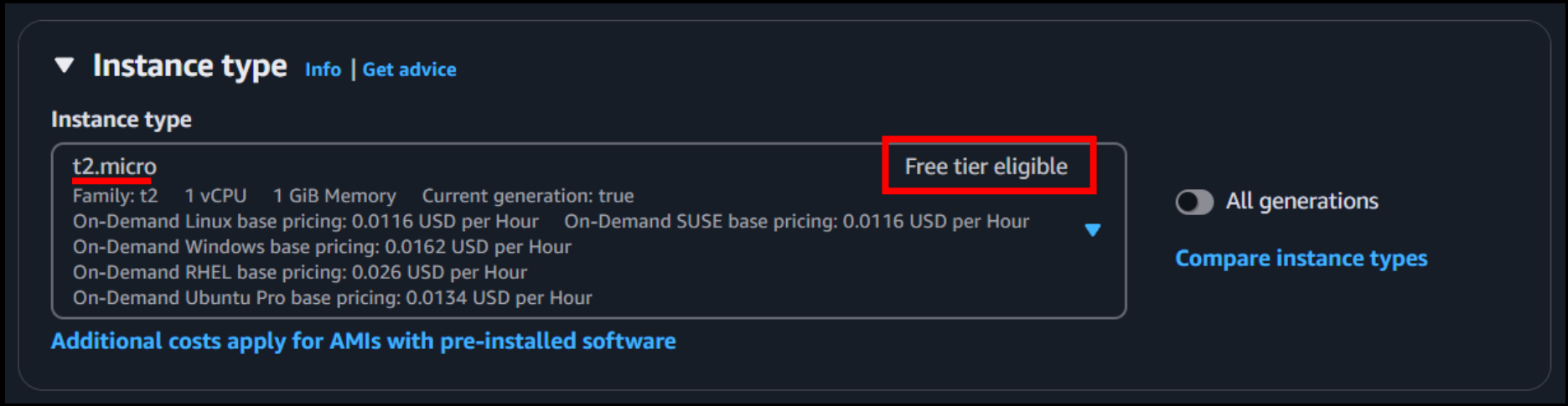
4. Assign a **name** to your instance.



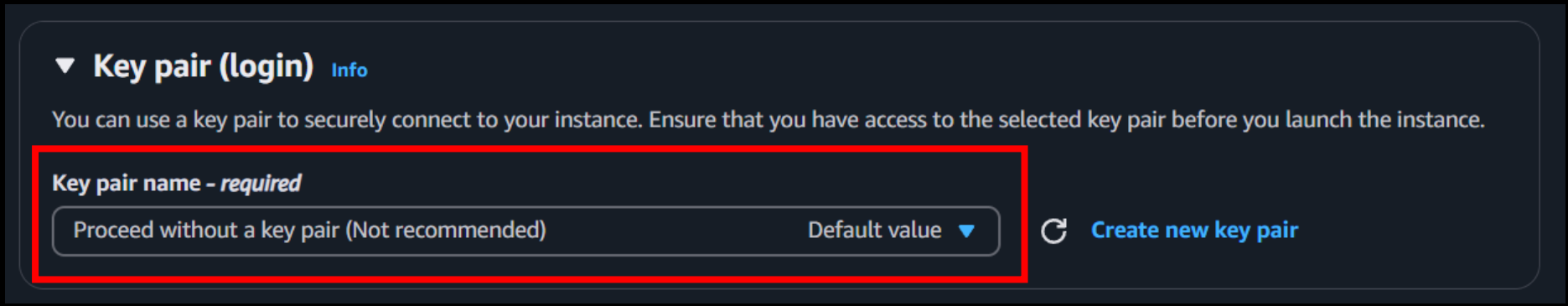
5. Choose **Amazon Linux 2 AMI (HVM)**



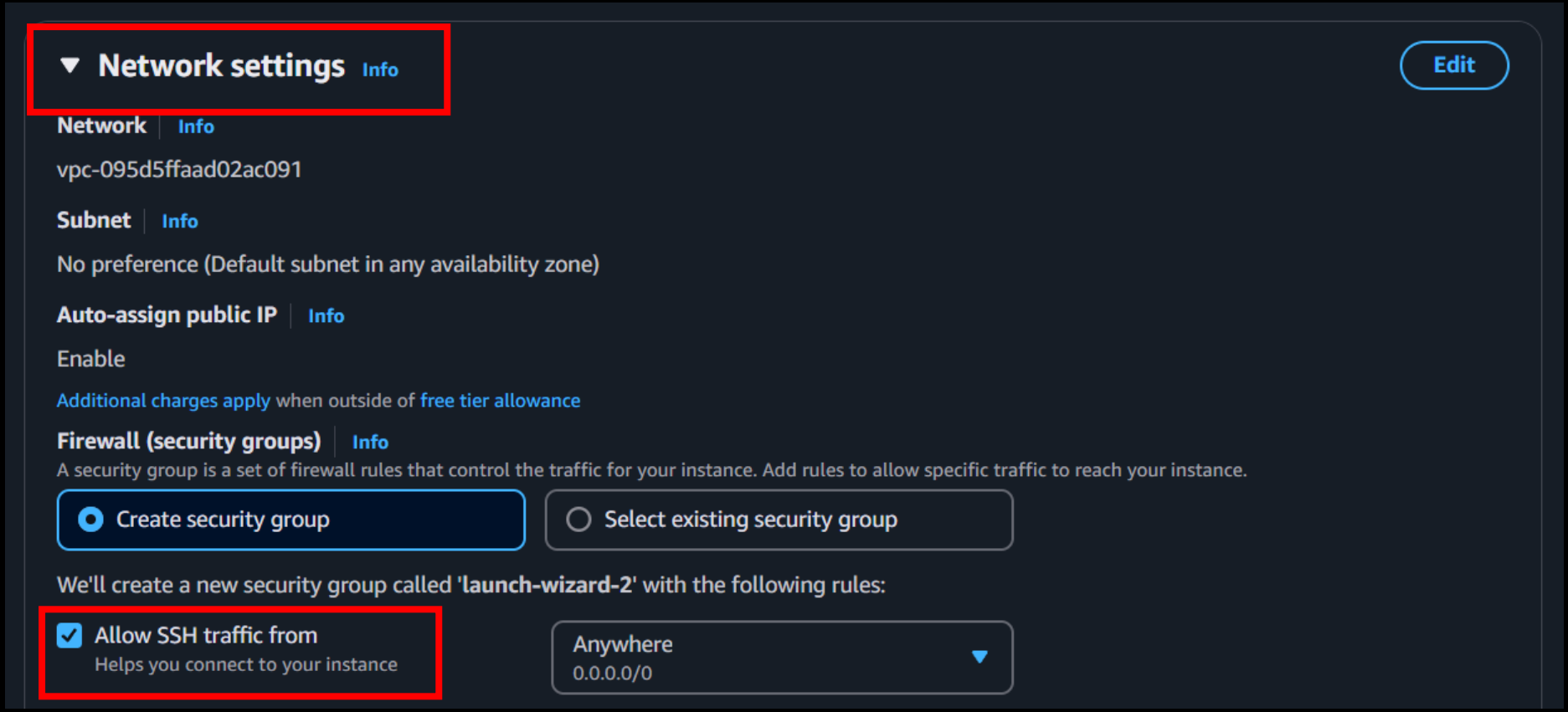
6. Select **t2.micro** or **t3.micro** as the instance type (whichever is available under free tier).



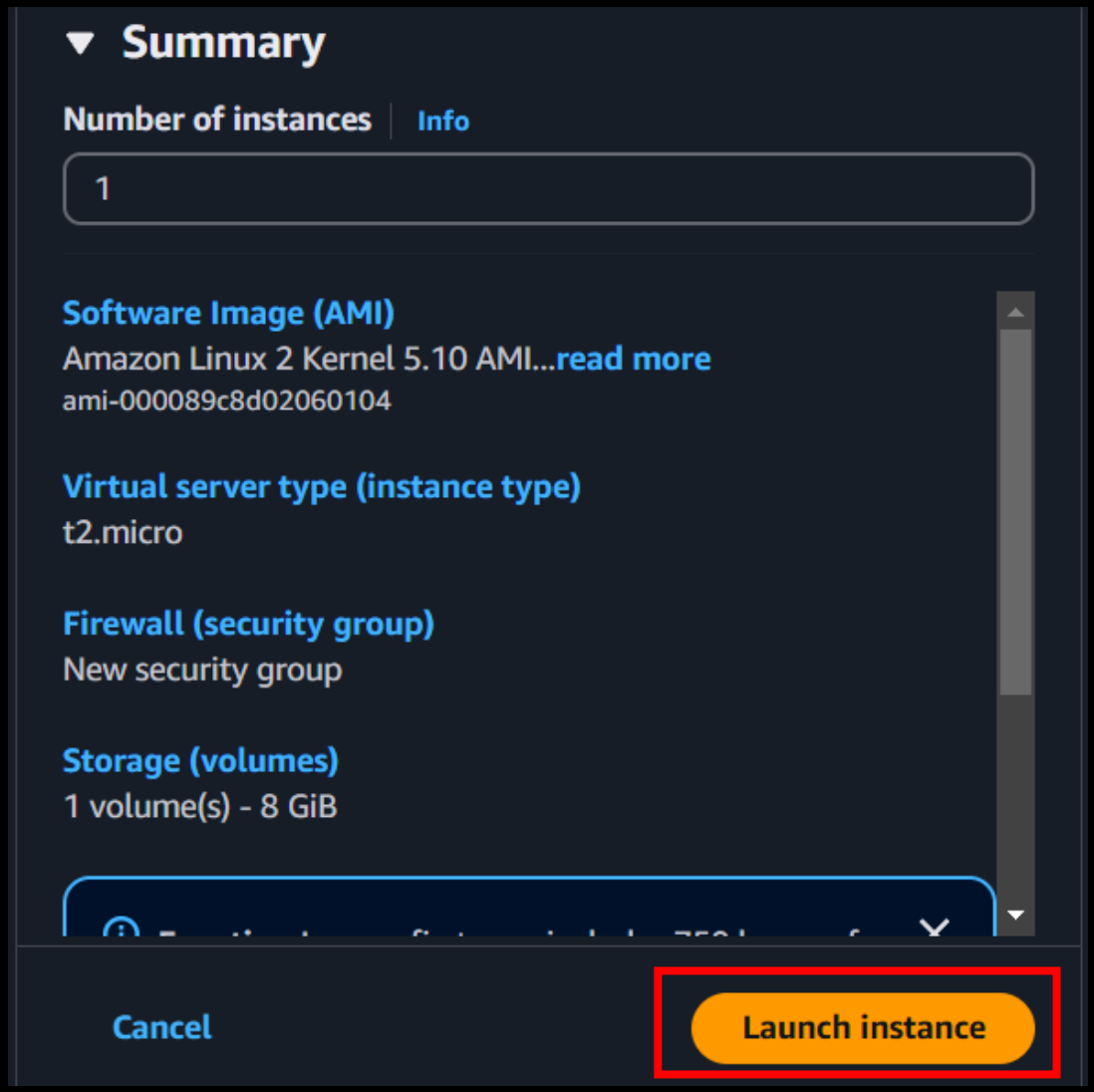
7. Proceed **without a key pair**.



8. In **Networking settings**, allow **SSH traffic.**

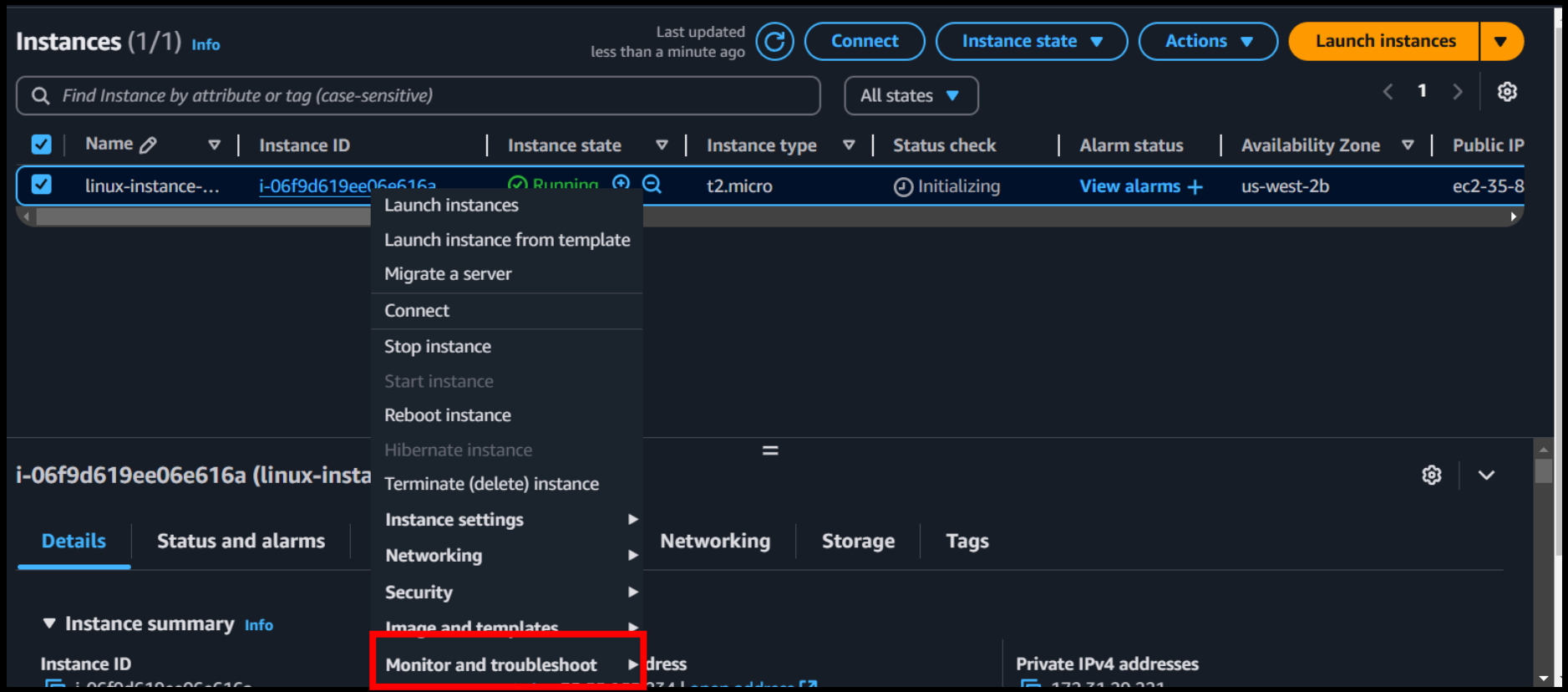


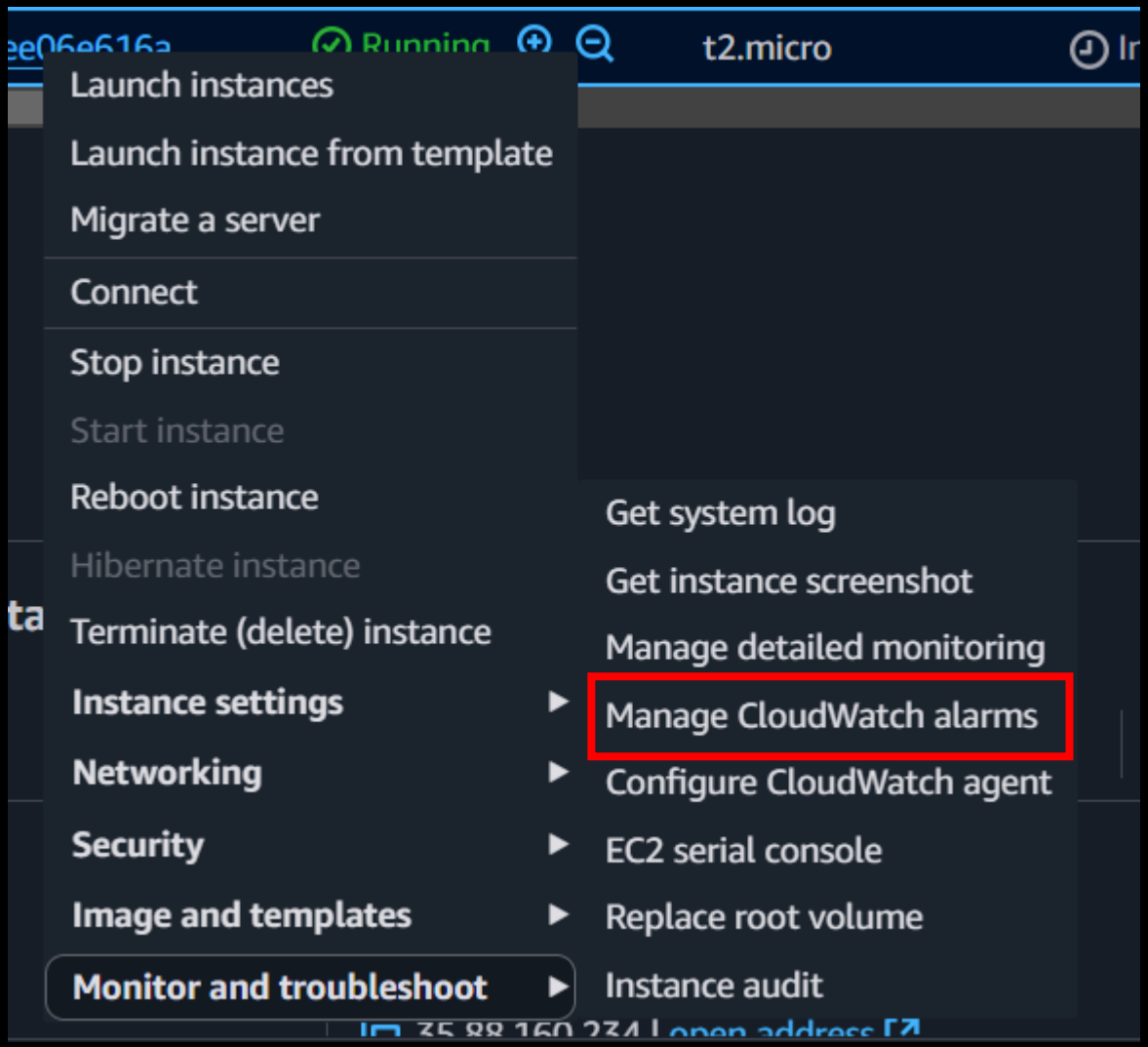
9. Click on **Launch Instance.**



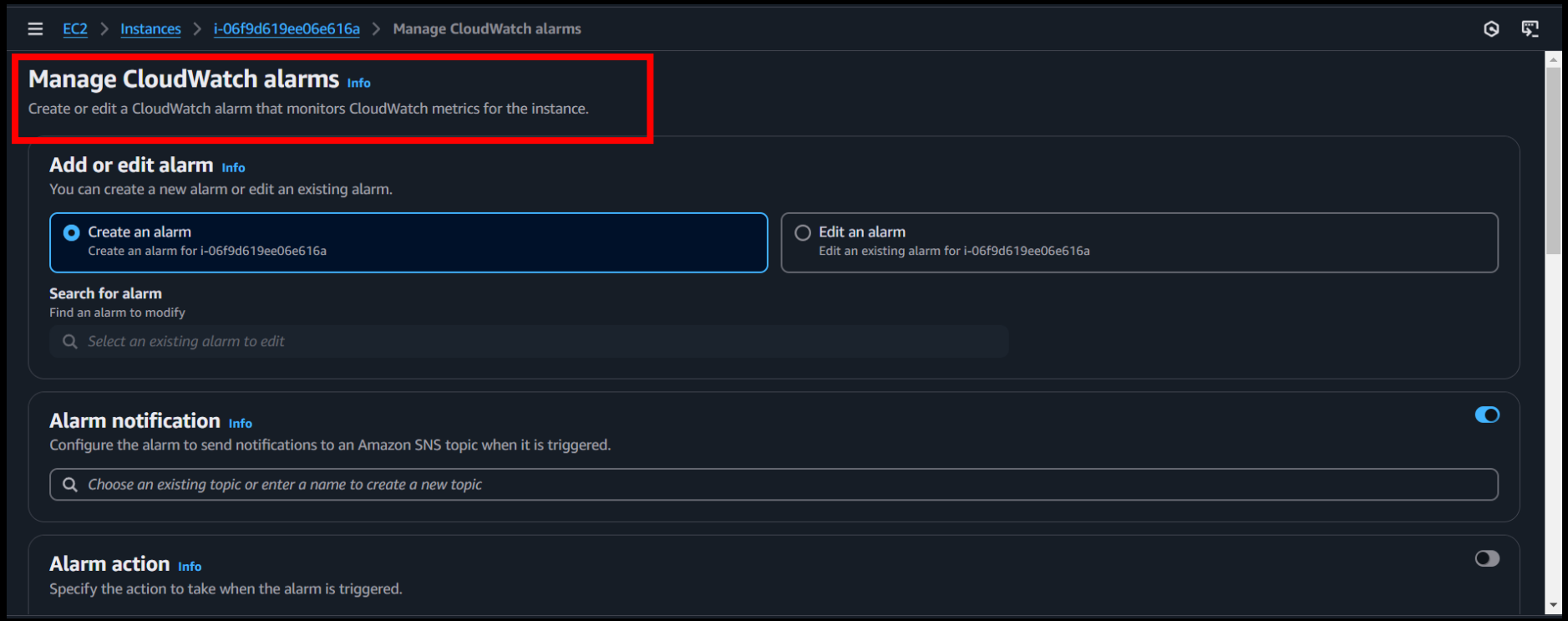
II. CONFIGURING CLOUDWATCH ALARM AND SNS 🔔

10. Click on the **Instance ID**, right-click, and navigate to **Monitor and Troubleshoot → Manage CloudWatch Alarms.**



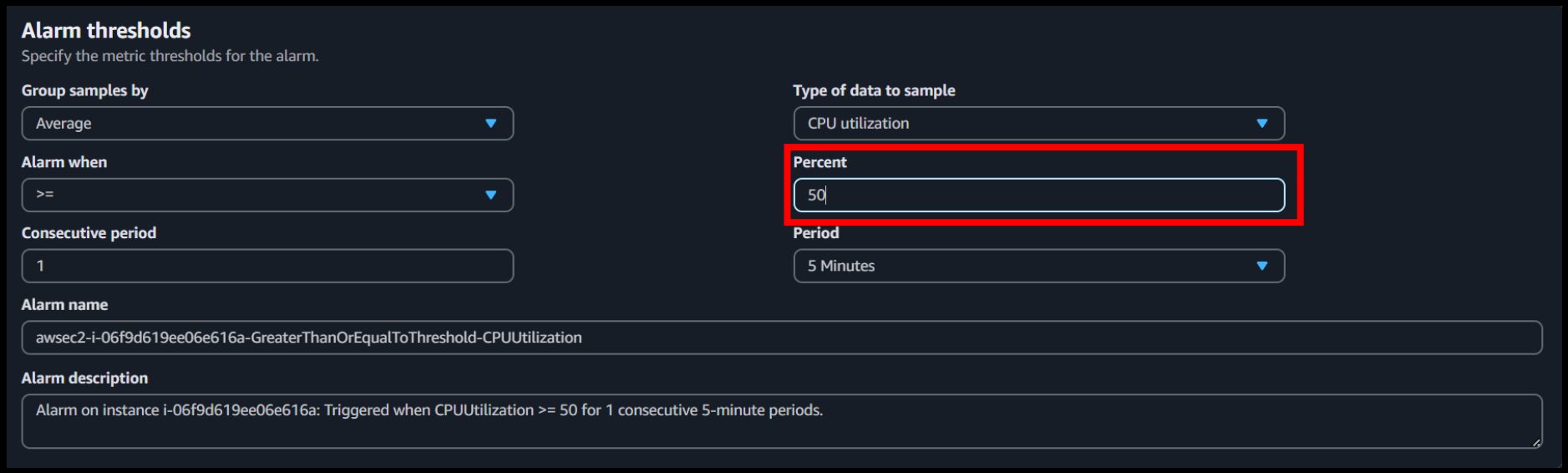

Running
t2.micro
Launch instances
Launch instance from template
Migrate a server
Connect
Stop instance
Start instance
Reboot instance
Hibernate instance
Terminate (delete) instance
Instance settings
Networking
Security
Image and templates
Monitor and troubleshoot
Get system log
Get instance screenshot
Manage detailed monitoring
Manage CloudWatch alarms
Configure CloudWatch agent
EC2 serial console
Replace root volume
Instance audit

11. You will be redirected to the **CloudWatch Alarms dashboard**.

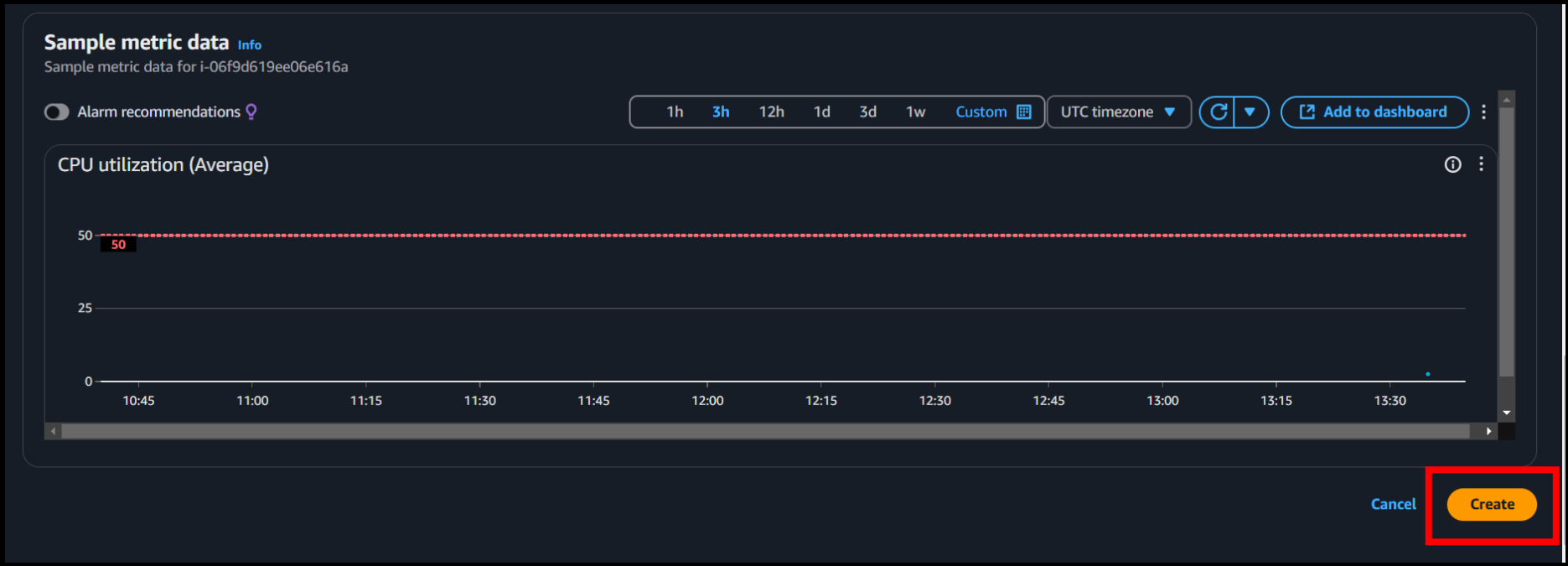


12. Click **Create Alarm** → In **Alarm Notification**, select or search for SNSTopicEC2.

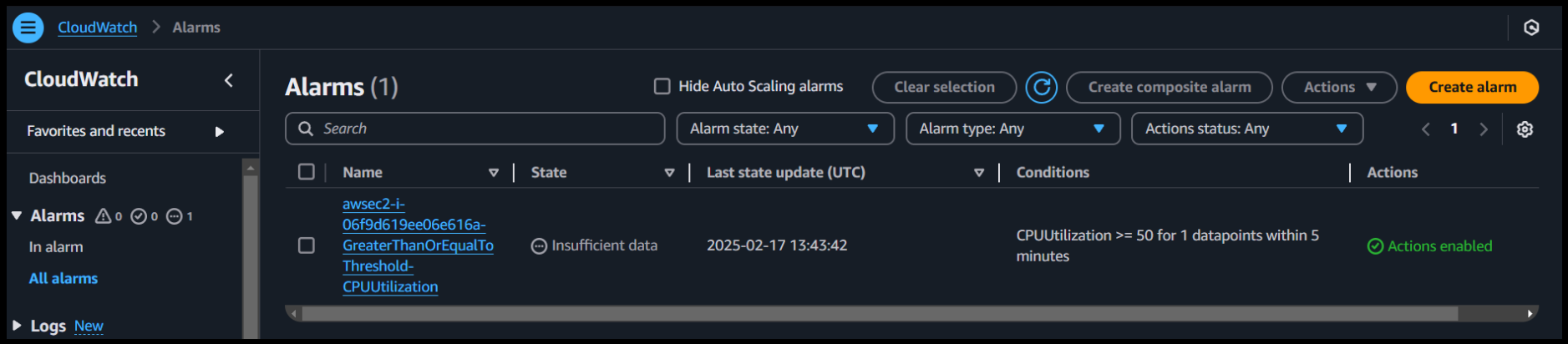
13. Under **Alarm Threshold**, change the CPU utilization percentage to **50%.**



14. Scroll down and click **Create Alarm.**

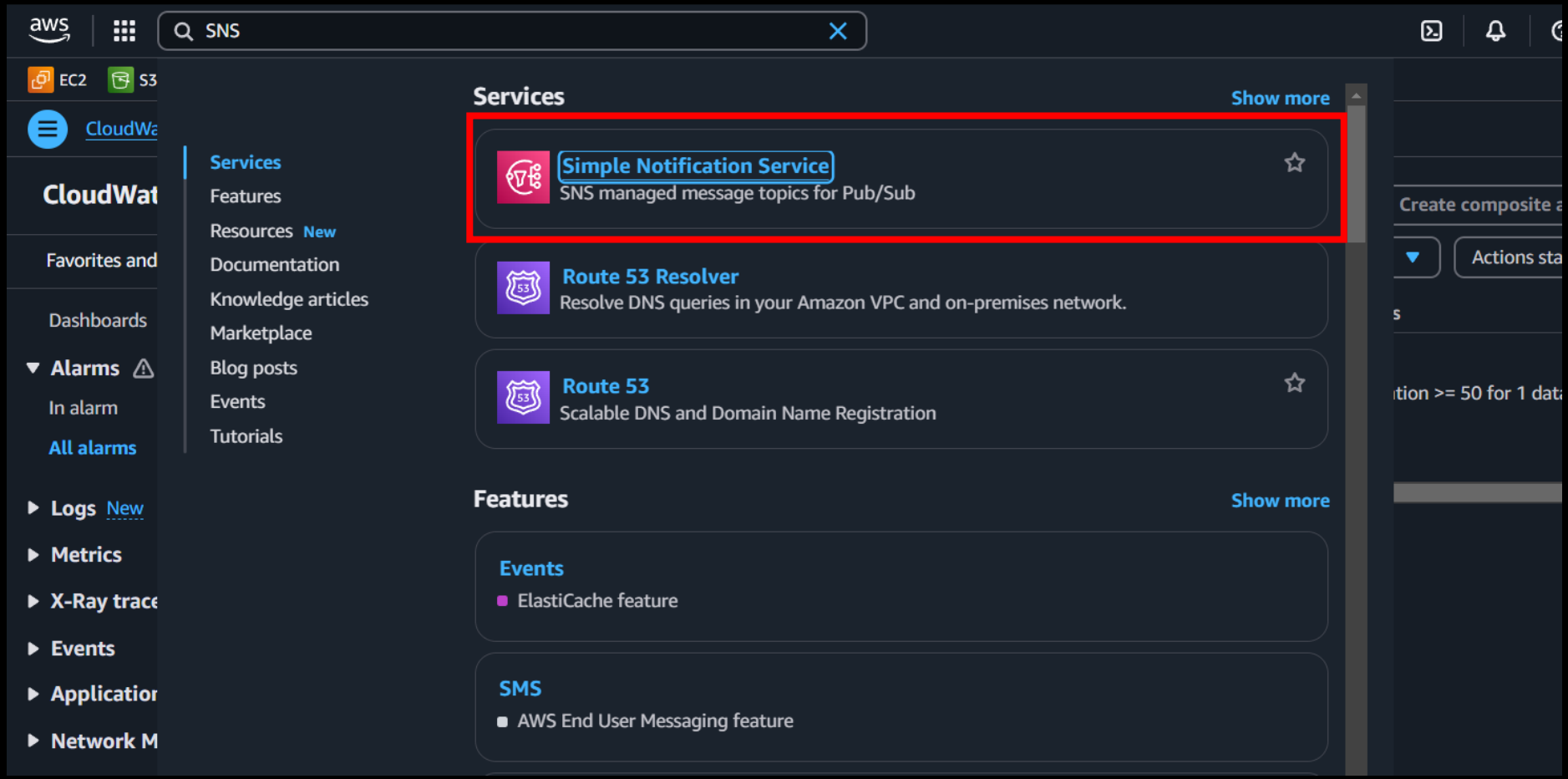


15. Verify that the alarm has been created successfully.

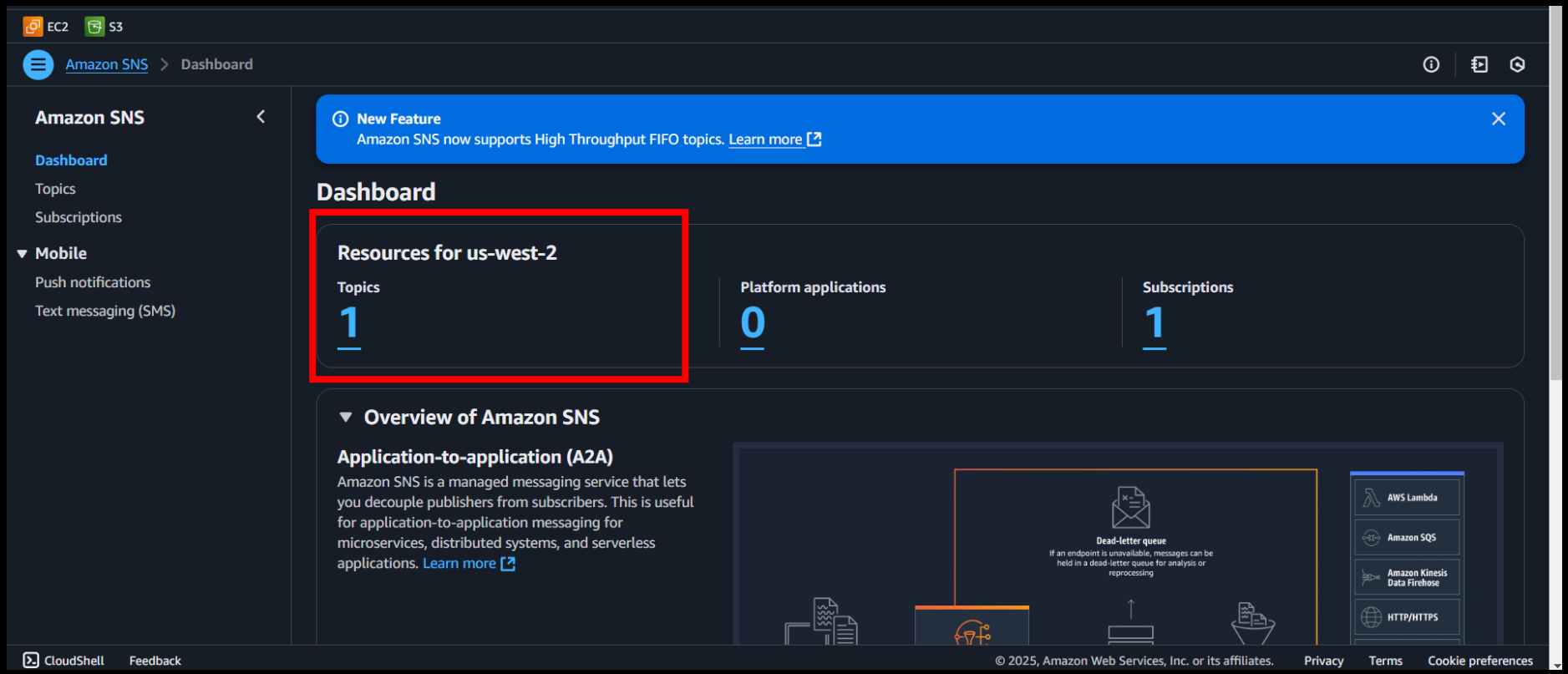


III. CONFIGURING SNS FOR NOTIFICATIONS 📧

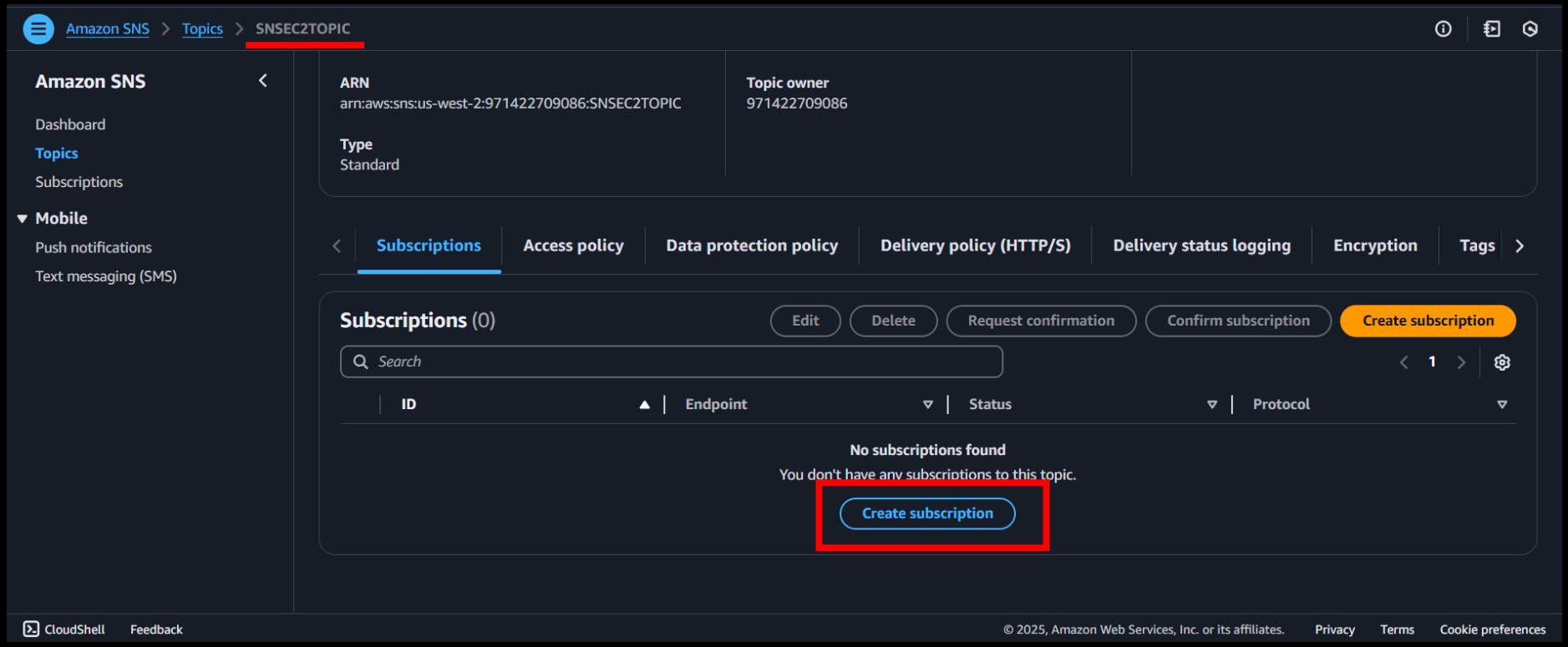
16. Search for **SNS service** and open it in a new tab.



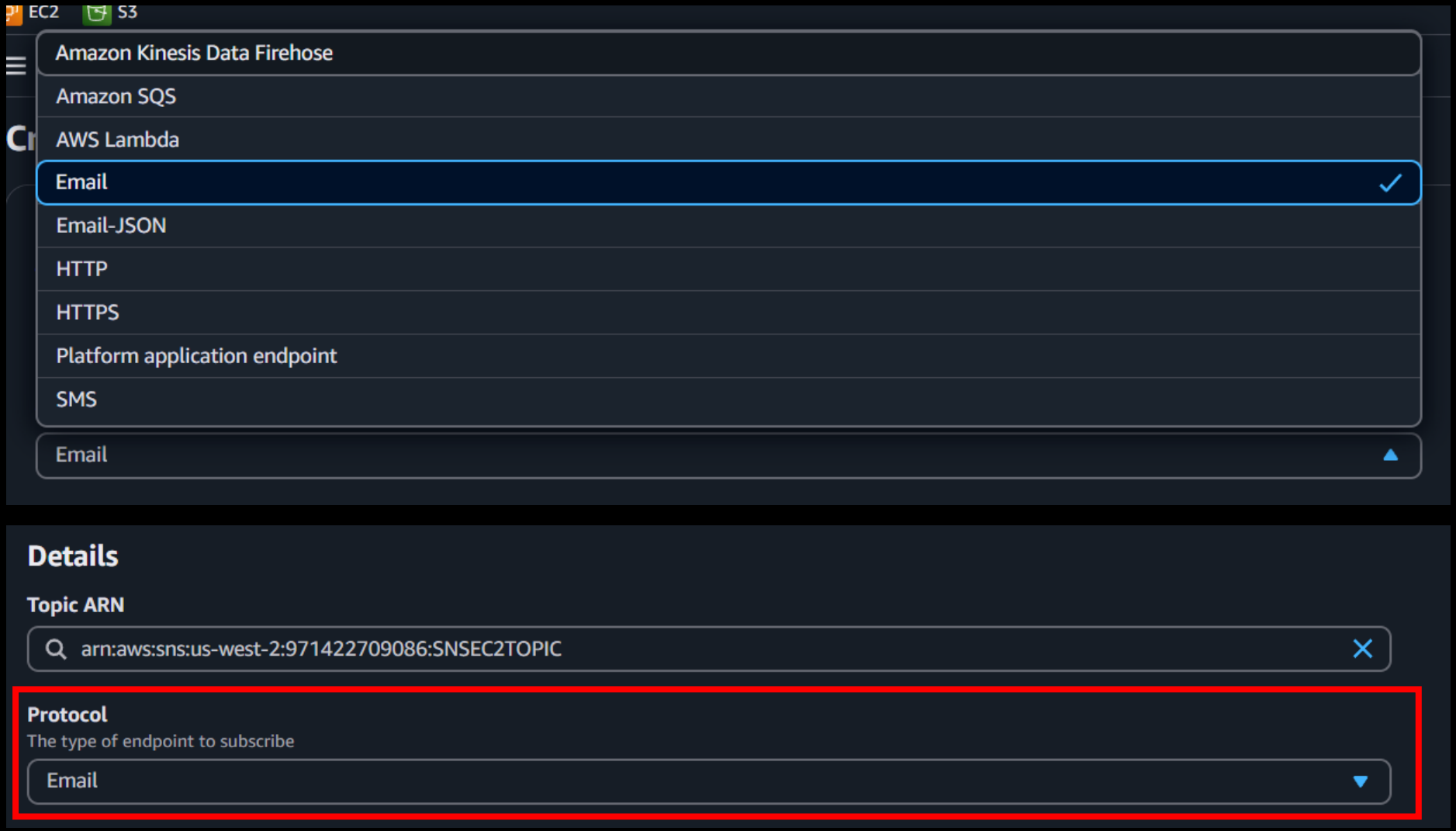
17. Navigate to the **SNS Dashboard**, where you will see one SNS topic.



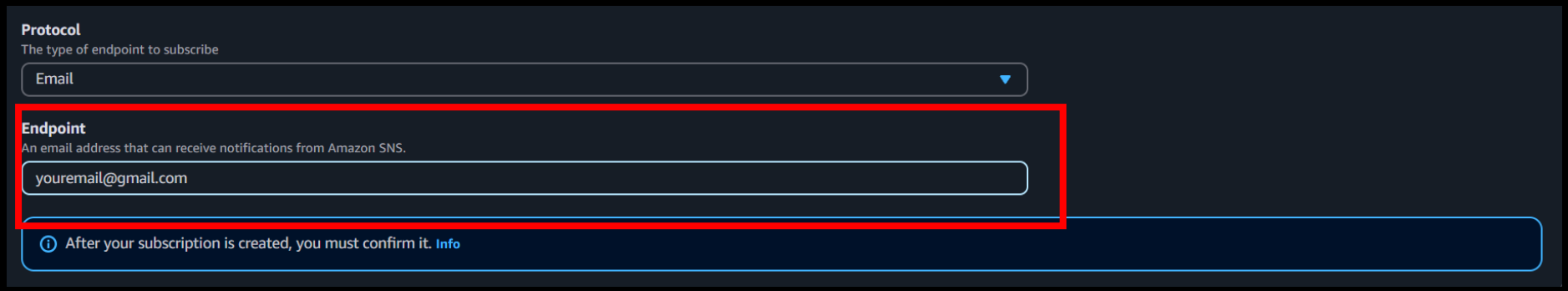
18. Click on the topic, scroll down, and click **Create Subscription.**



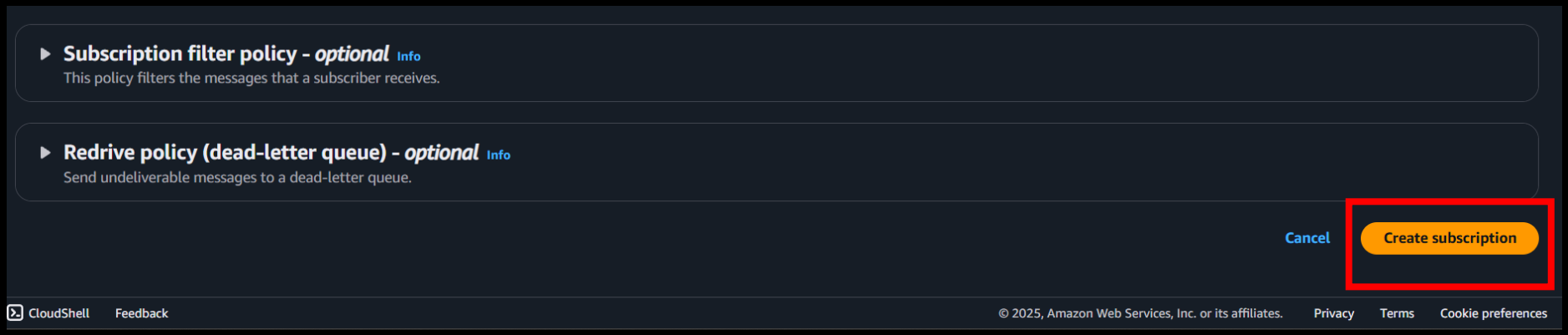
19. Choose **Email** as the protocol.



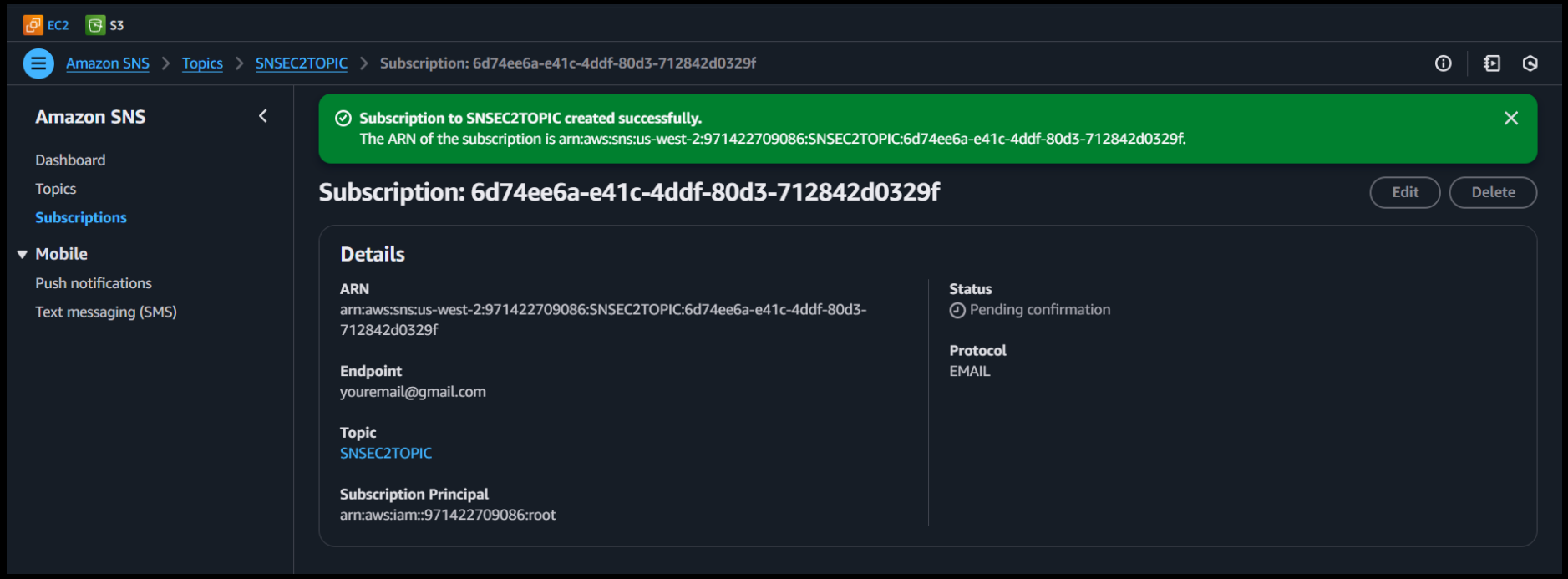
20. Enter your **email address** in the **Endpoint** field.



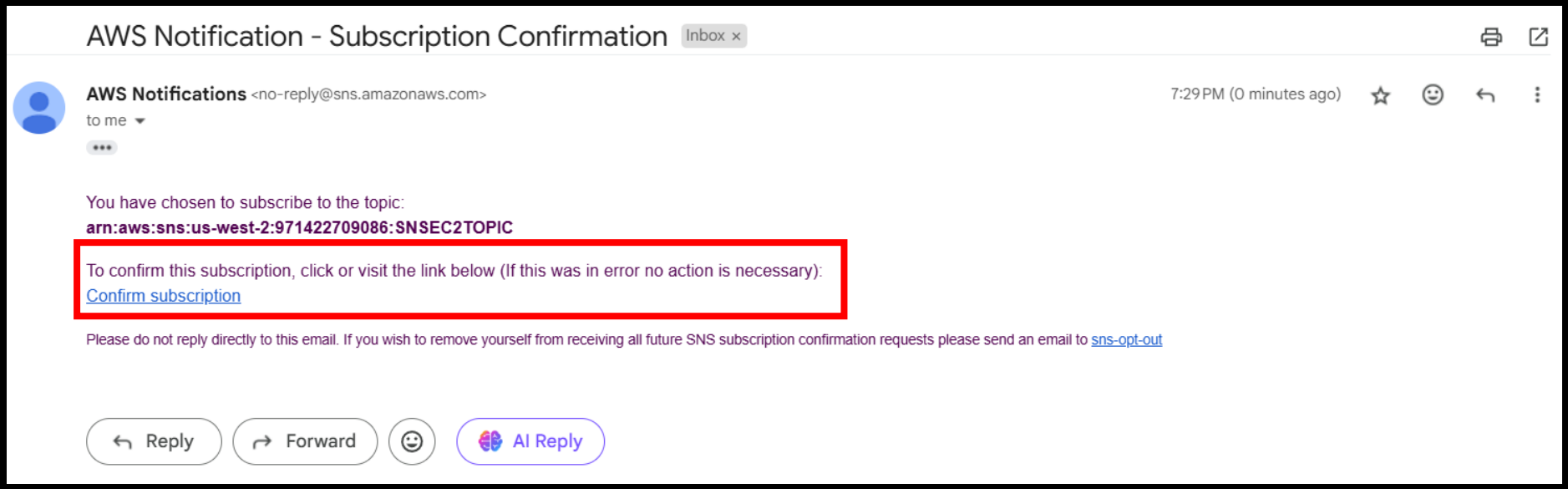
21. Scroll down and click **Create Subscription.**



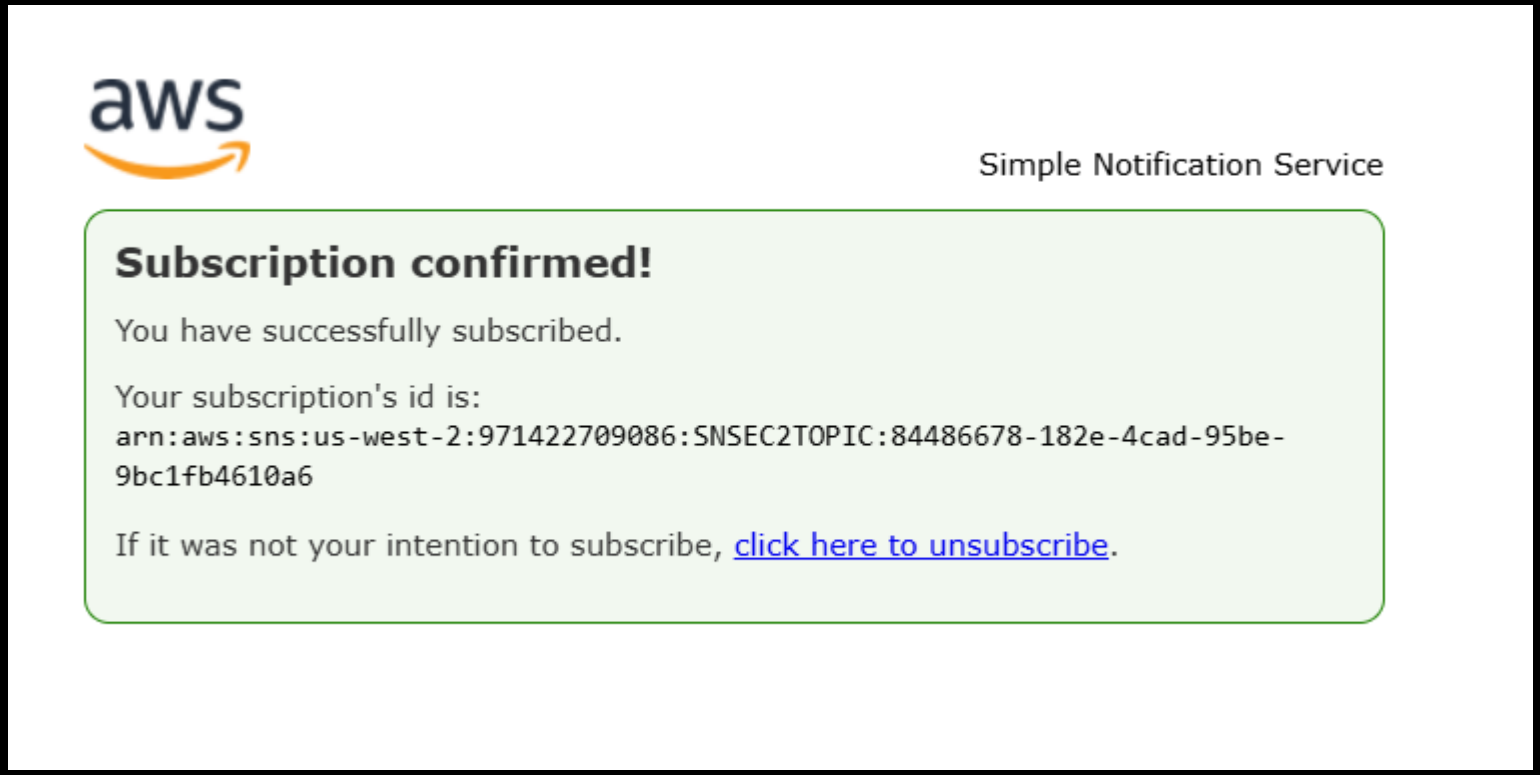
22. Verify that your subscription appears in the **SNS Dashboard.**



23. Open your email and confirm the **AWS Notification Subscription.**



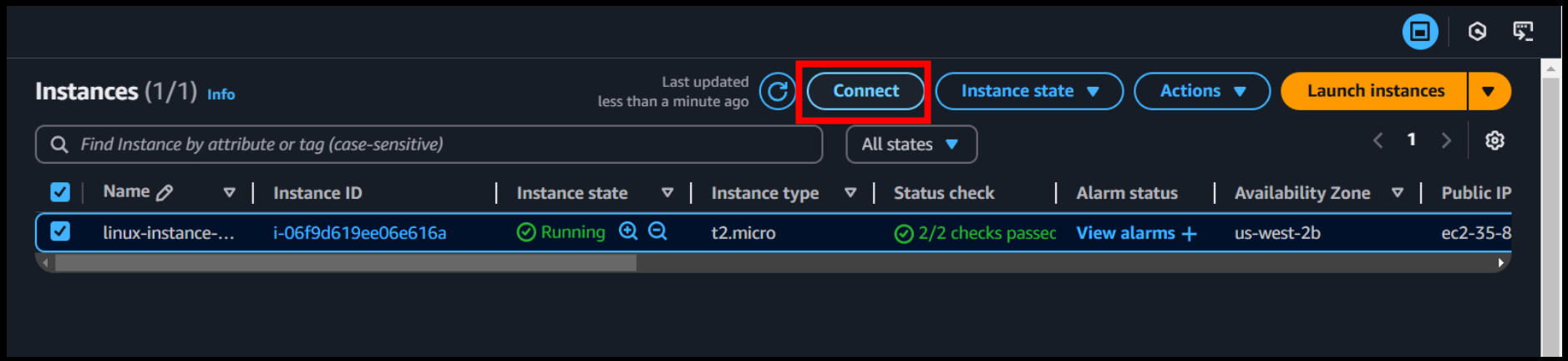
24. Upon confirmation, you will be redirected to a page that says **Subscription Confirmed.**

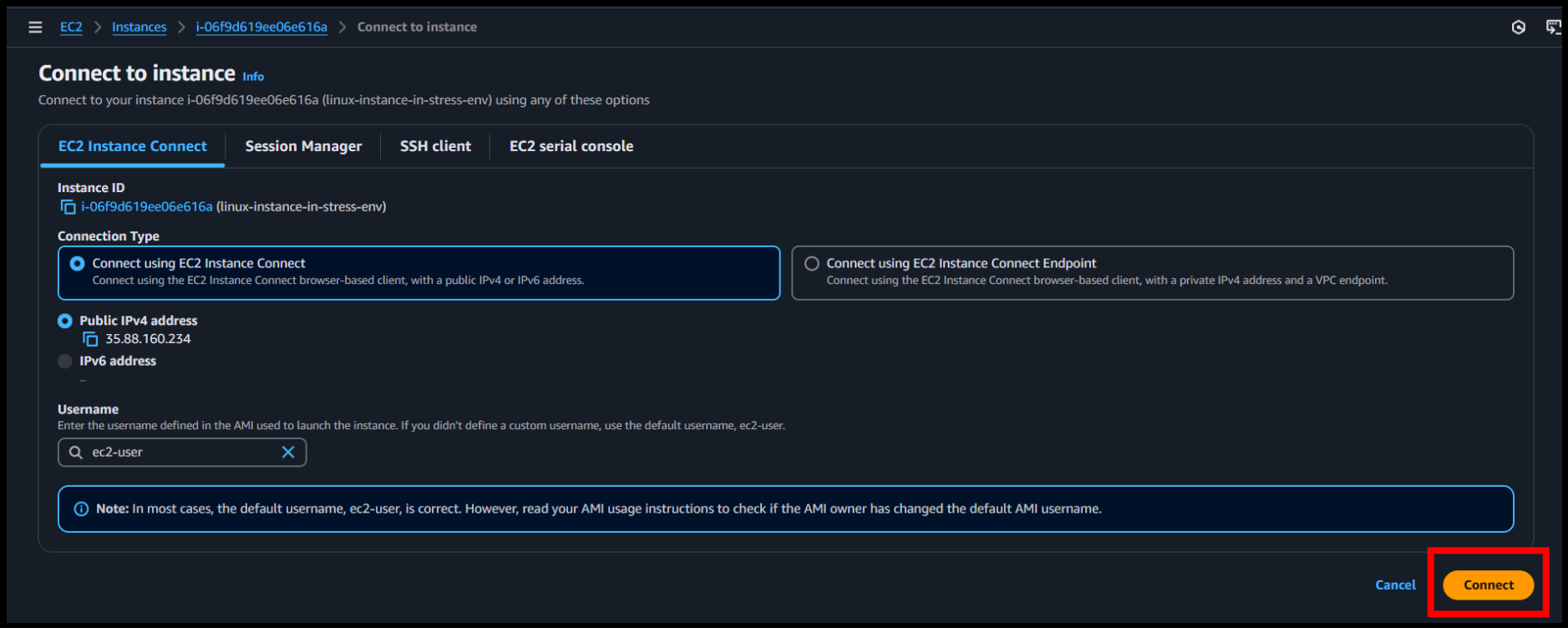


25. Refresh the SNS dashboard and verify that the subscription status is **confirmed.**

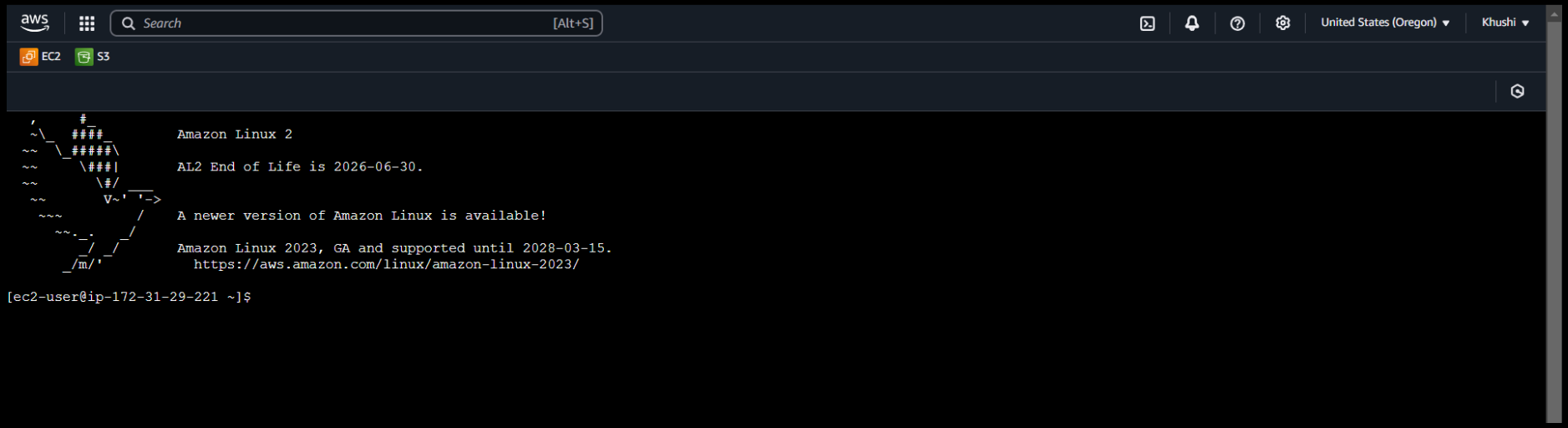
IV. CREATING A STRESS ENVIRONMENT ON EC2 💻

26. Navigate back to your **EC2 instance**, select it, and click **Connect.**

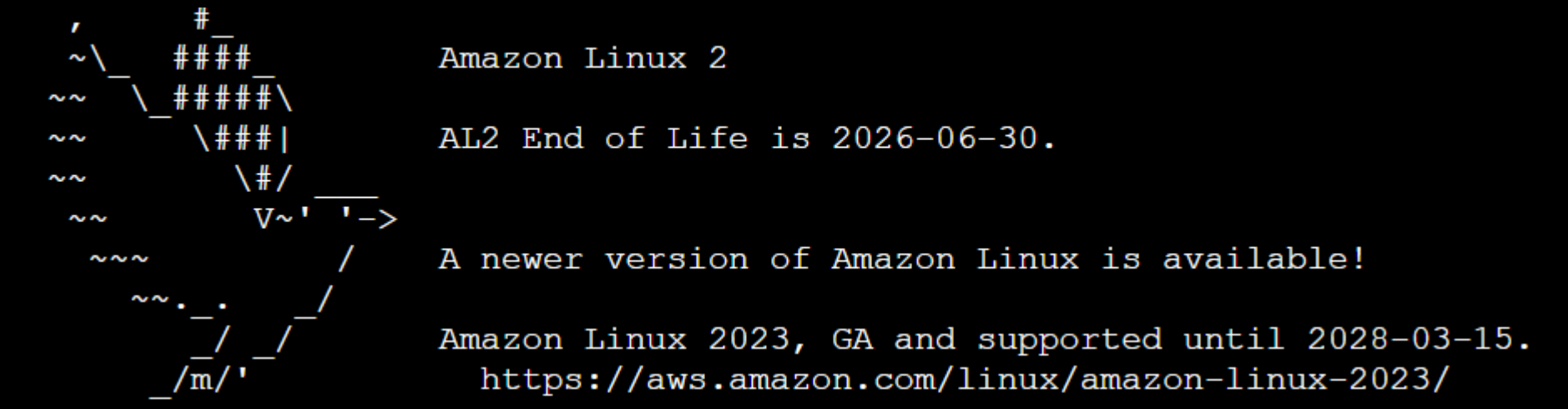



EC2 > Instances > i-06f9d619ee06e616a > Connect to instance
Connect to instance Info
Connect to your instance i-06f9d619ee06e616a (linux-instance-in-stress-env) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-06f9d619ee06e616a (linux-instance-in-stress-env)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IPv4 address
35.88.160.234
IPv6 address
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.
ec2-user
Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect

27. Once the Linux server is ready, execute the following commands to install the stress tool:



- **sudo yum update -y**: Updates all installed packages.



- **sudo amazon-linux-extras install epel -y**: Enables the Extra Packages for Enterprise Linux (EPEL) repository.

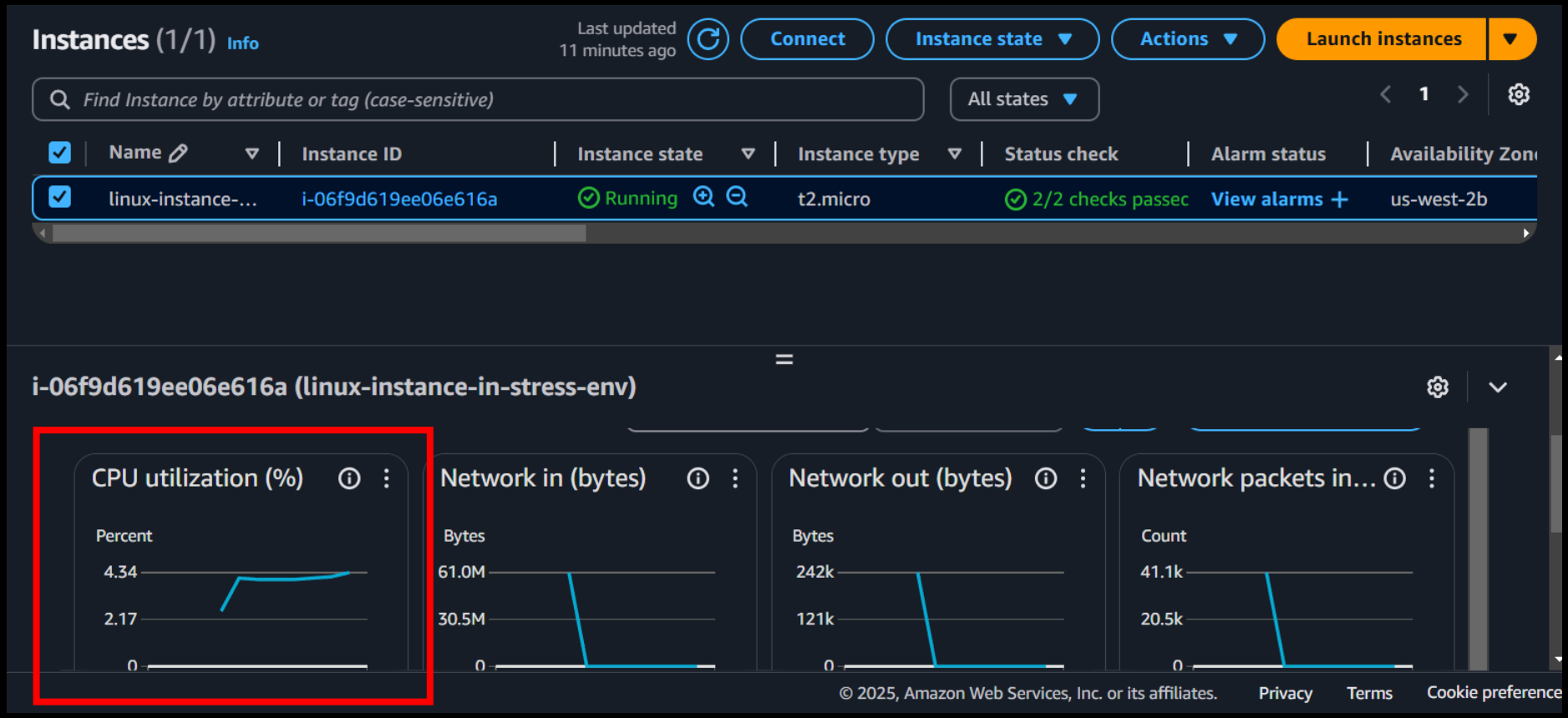
```
[ec2-user@ip-172-31-29-221 ~]$ sudo amazon-linux-extras install epel -y
Installing epel-release
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-epel amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
           | 3.6 kB  00:00:00
amzn2extra-docker
           | 2.9 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
           | 2.7 kB  00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo
           | 1.0 MB  00:00:00
(3/9): amzn2extra-epel/2/x86_64/primary_db
           | 1.8 kB  00:00:00
```

- **sudo yum install stress -y**: Installs the **stress** tool for generating high CPU usage.

```
[ec2-user@ip-172-31-29-221 ~]$ sudo yum install stress -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
230 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
Installed size: 94 k
```

V. GENERATING STRESS AND MONITORING CLOUDWATCH

28. Go to the **EC2 instance → Monitoring** tab, scroll down, and enlarge the **CPU Utilization** graph.

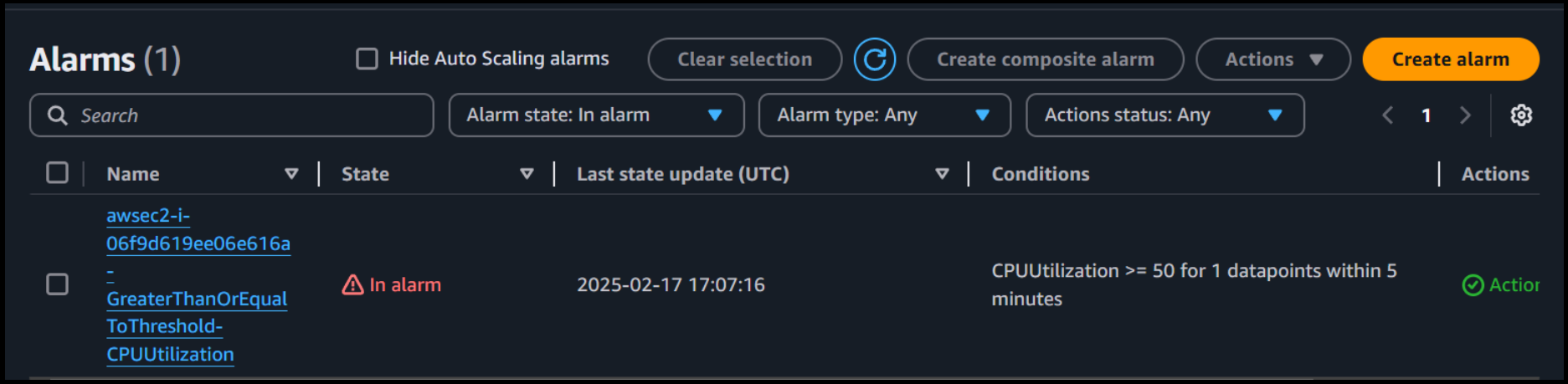


29. Run the following command in your Linux terminal to generate stress on the CPU:

stress --cpu 9

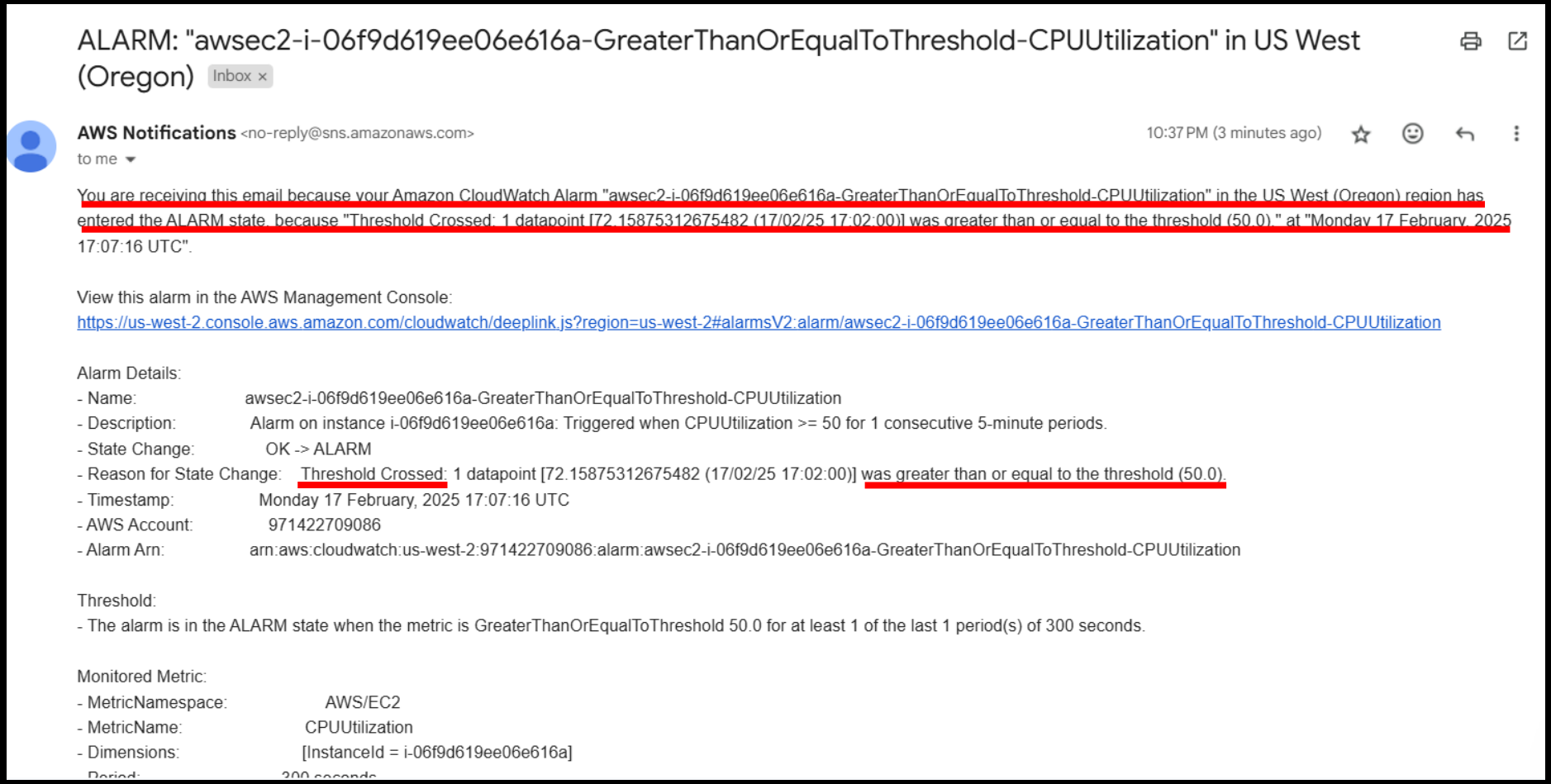
```
                                1/1
  Verifying  : stress-1.0.4-16.el7.x86_64
                                1/1

Installed:
  stress.x86_64 0:1.0.4-16.el7


Complete!
[ec2-user@ip-172-31-29-221 ~]$ stress --cpu 9
stress: info: [32665] dispatching hogs: 9 cpu, 0 io, 0 vm, 0 hdd
```

30. Go back to the **CloudWatch dashboard**, click **All Alarms**, and you will see your alarm



31. Wait a few minutes and observe the increase in CPU utilization.

32. You will receive an **email notification** when CPU utilization crosses the threshold.



ALARM: "awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization" in US West (Oregon) Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 10:37 PM (3 minutes ago)

to me

You are receiving this email because your Amazon CloudWatch Alarm "awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization" in the US West (Oregon) region has entered the ALARM state, because "Threshold Crossed: 1 datapoint [72.15875312675482 (17/02/25 17:02:00)] was greater than or equal to the threshold (50.0)." at "Monday 17 February, 2025 17:07:16 UTC".

View this alarm in the AWS Management Console:
https://us-west-2.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-west-2#alarmsV2:alarm/awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization

Alarm Details:
- Name: awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization
- Description: Alarm on instance i-06f9d619ee06e616a: Triggered when CPUUtilization >= 50 for 1 consecutive 5-minute periods.
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint [72.15875312675482 (17/02/25 17:02:00)] was greater than or equal to the threshold (50.0).
- Timestamp: Monday 17 February, 2025 17:07:16 UTC
- AWS Account: 971422709086
- Alarm Arn: arn:aws:cloudwatch:us-west-2:971422709086:alarm:awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization

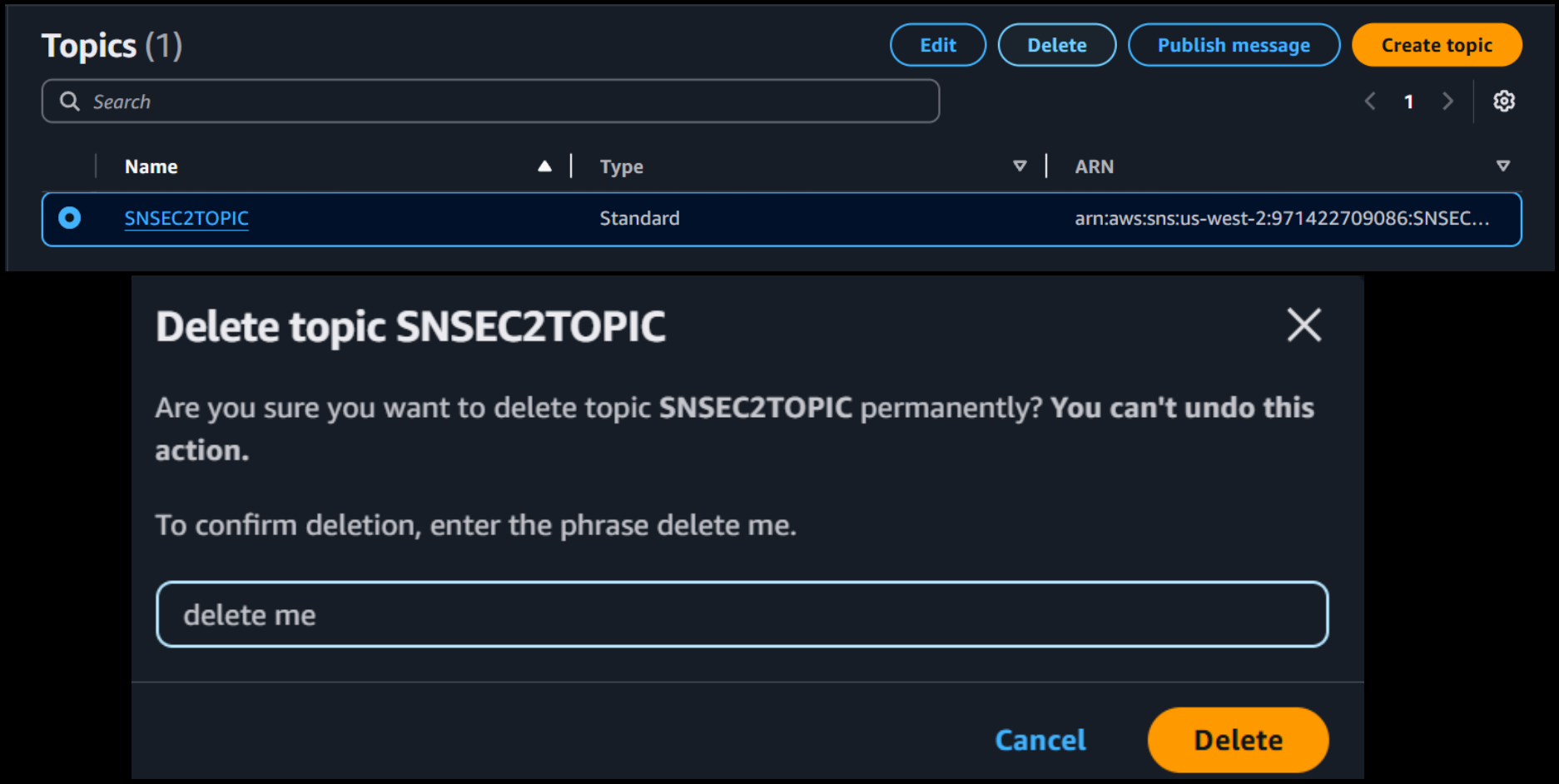
Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 50.0 for at least 1 of the last 1 period(s) of 300 seconds.

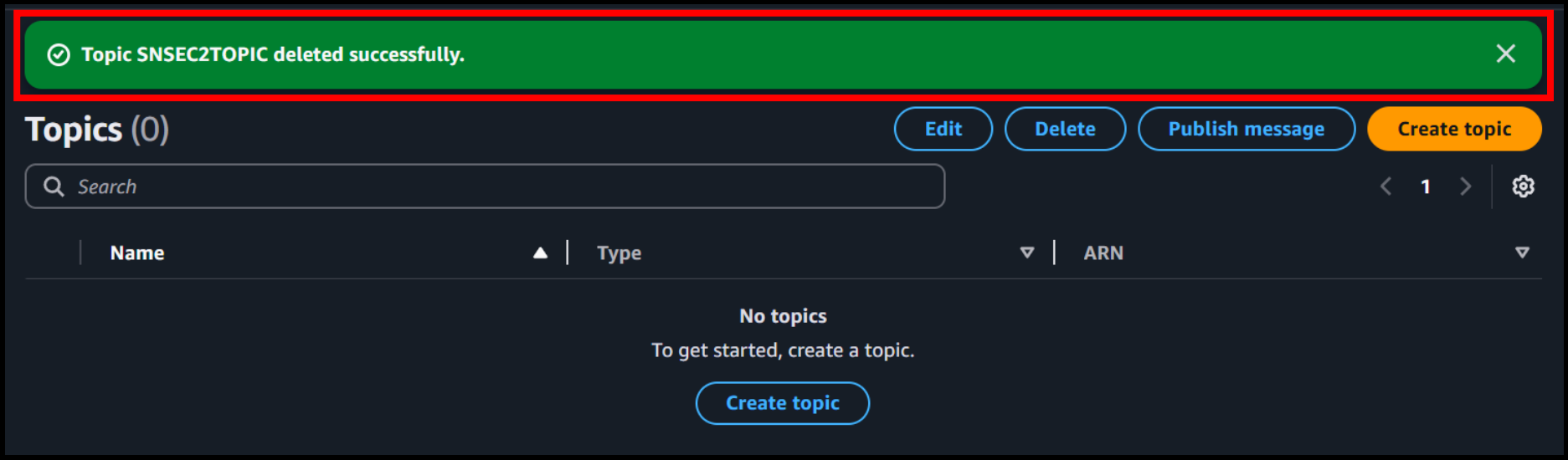
Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [InstanceId = i-06f9d619ee06e616a]
- Period: 300 seconds

VI. CLEANING UP RESOURCES 🗑️

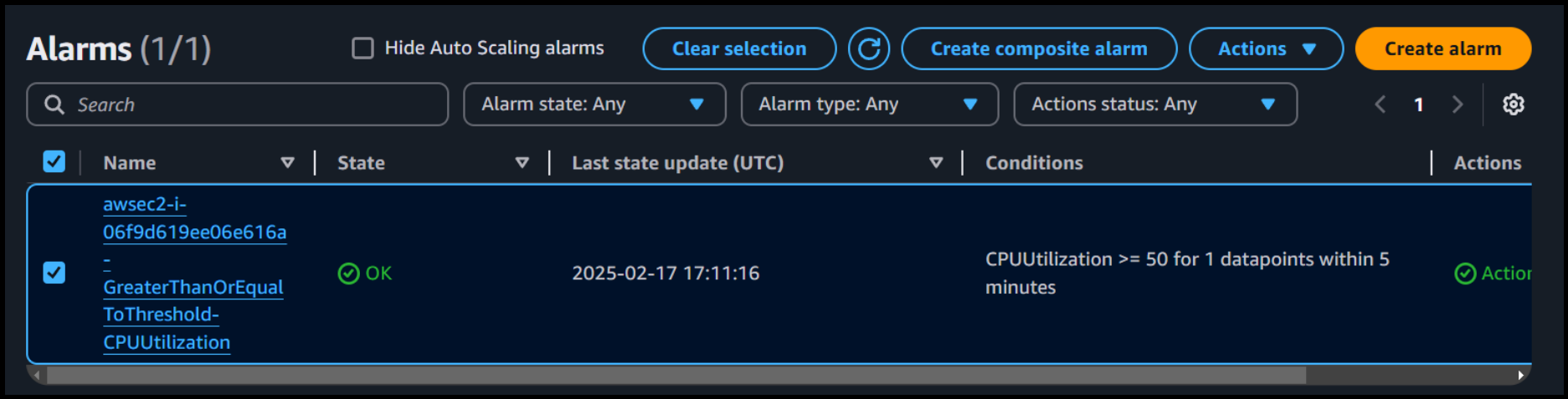
To avoid unnecessary charges, follow these steps to delete resources:

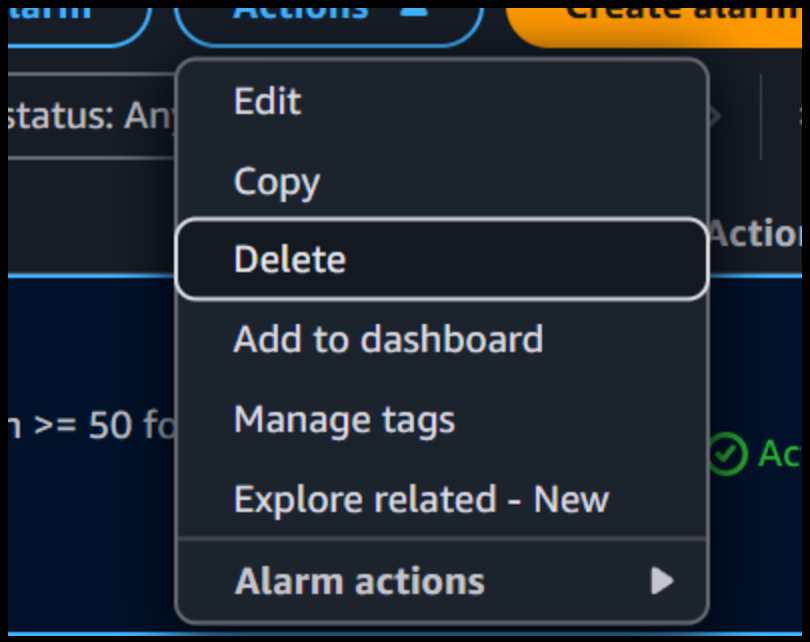
1. Delete the **SNS Topic.**

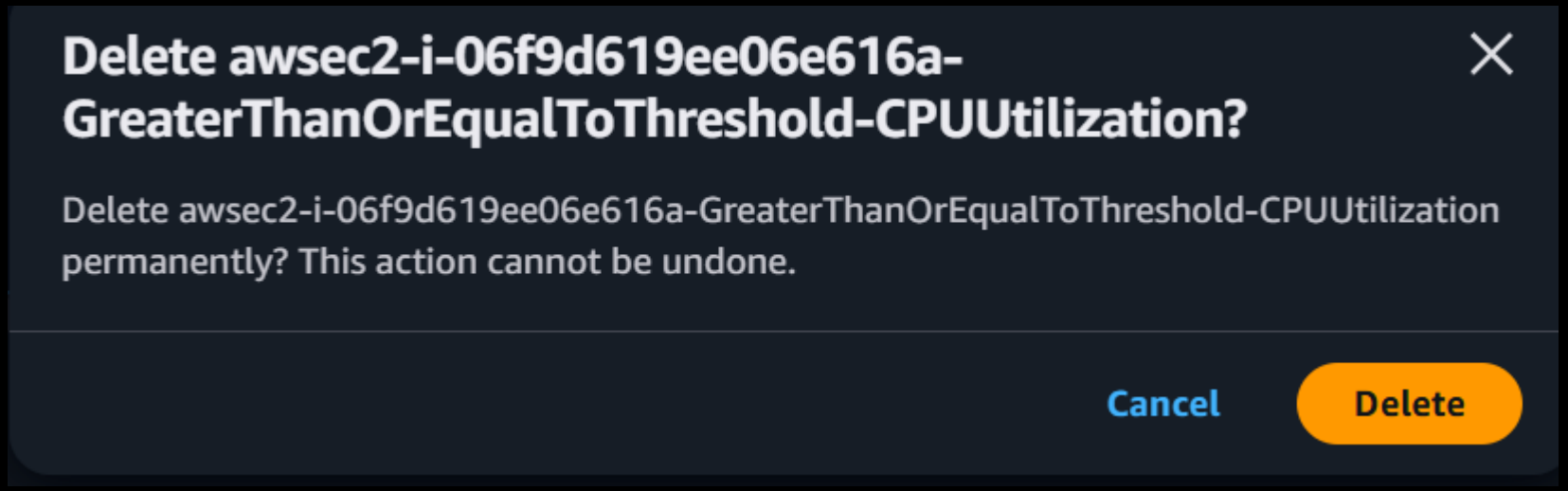


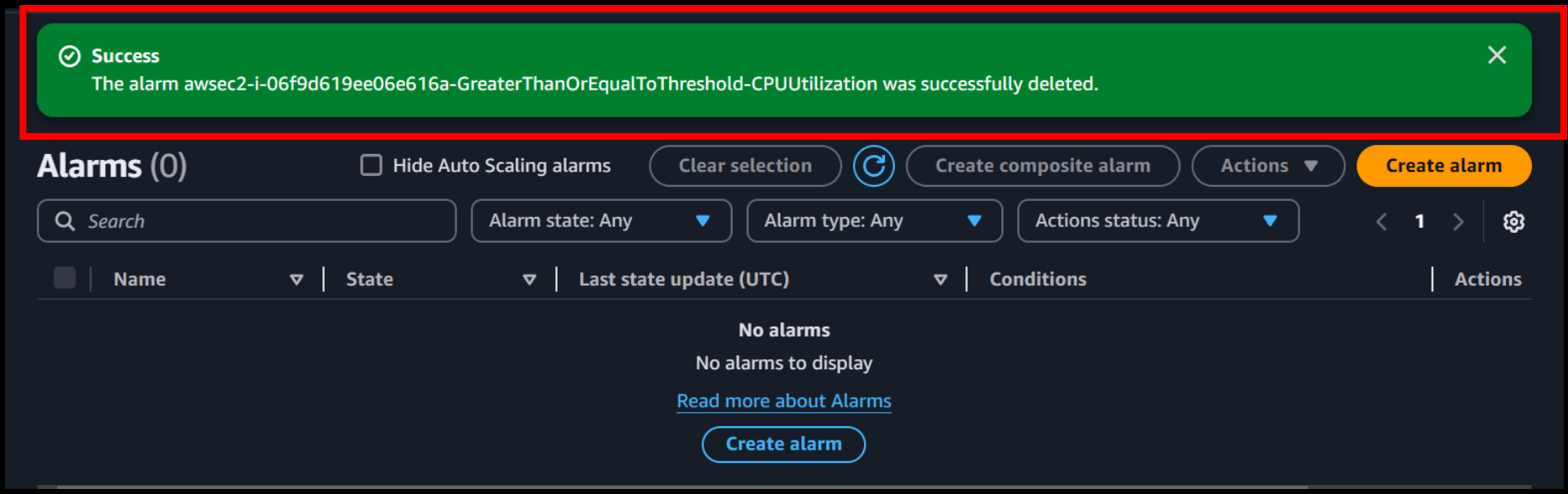


2. Delete the **CloudWatch Alarm.**

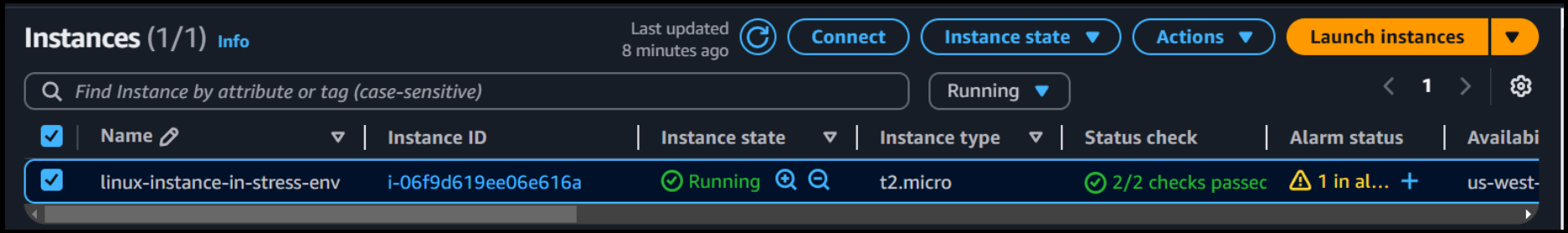


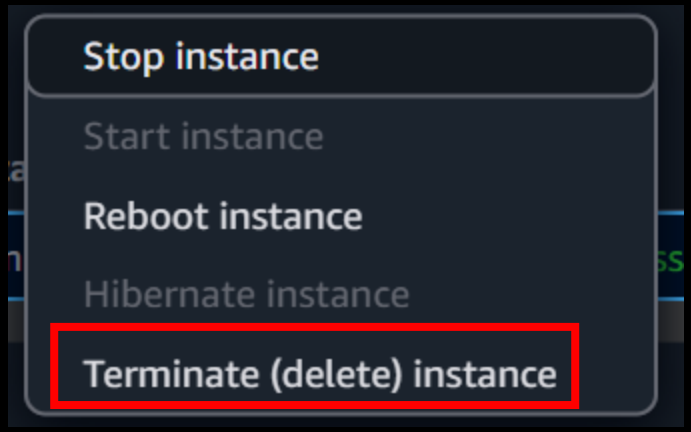

Edit
Copy
Delete
Add to dashboard
Manage tags
Explore related - New
Alarm actions

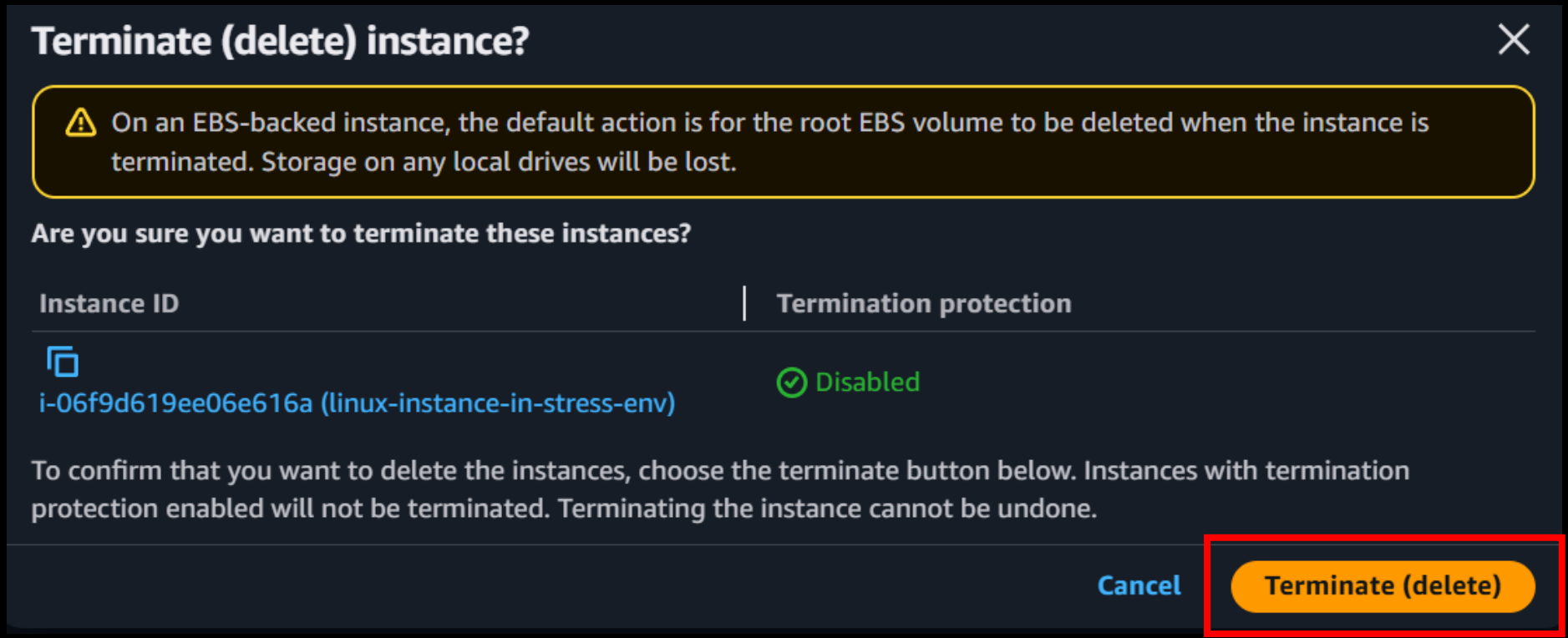

Delete awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization?
Delete awsec2-i-06f9d619ee06e616a-GreaterThanOrEqualToThreshold-CPUUtilization permanently? This action cannot be undone.
Cancel
Delete

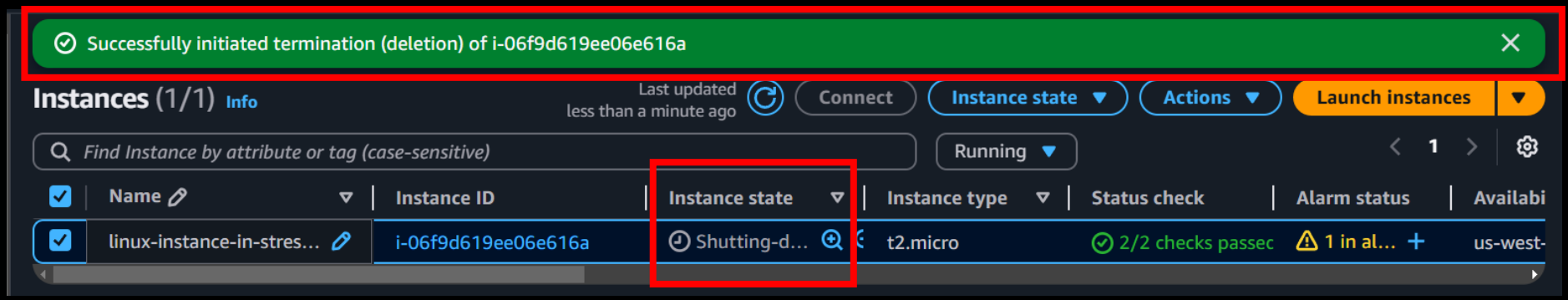


3. Terminate the **EC2 Instance.**




Stop instance
Start instance
Reboot instance
Hibernate instance
Terminate (delete) instance


Terminate (delete) instance?
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.
Are you sure you want to terminate these instances?
Instance ID
Termination protection
i-06f9d619ee06e616a (linux-instance-in-stress-env)
Disabled
To confirm that you want to delete the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.
Cancel
Terminate (delete)


Successfully initiated termination (deletion) of i-06f9d619ee06e616a
Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Running
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availabi
linux-instance-in-stres...
i-06f9d619ee06e616a
Shutting-d...
t2.micro
2/2 checks passec
1 in al...
us-west-

CONCLUSION

In this project, we successfully created an EC2 instance, configured CloudWatch alarms, and set up SNS notifications to monitor CPU utilization. By inducing artificial stress using the **stress** tool, we observed real-time CPU performance monitoring and received alert notifications via email. This setup is useful for performance testing, monitoring system health, and responding to potential issues in a cloud environment.

By following these steps, you have learned the fundamentals of setting up an automated alert system using **AWS EC2, CloudWatch, and SNS**, which can be extended for production-level monitoring and response automation.

'WHAT GETS MEASURED, GETS MANAGED'

- PETER DRUCKER