Edge Computing Observability

Overview

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real-time. You can use CloudWatch to collect and track metrics, which are the variables you want to measure for your resources and applications. CloudWatch alarms send notifications or automatically make changes to the resources you are monitoring based on rules that you define. For example, you can monitor the CPU usage and disk reads and writes of your Amazon Elastic Compute Cloud (Amazon EC2) instances and then use this data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop under-used instances to save money. In addition to monitoring the built-in metrics that come with AWS, you can monitor your own custom metrics. With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.

In this lab, you will utilize CloudWatch to track EC2 CPU utilization and set up Alarm based on a configured threshold. The Alarm will trigger a Simple Notification Service (SNS) notification. As an optional exercise, you will utilize CloudWatch to monitor Billing and send a notification if estimated charges are above a defined threshold.

Note: You will need an AWS account and the associated administrative login credentials used in our experiments.

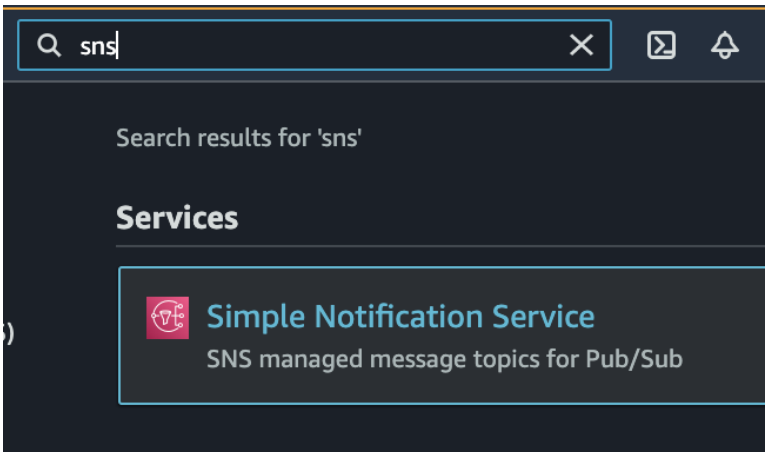
Sign in to AWS

Open the console link provided for your AWS account in us-east-1.console.aws.amazon.com

Create Simple Notification Service (SNS) Topic

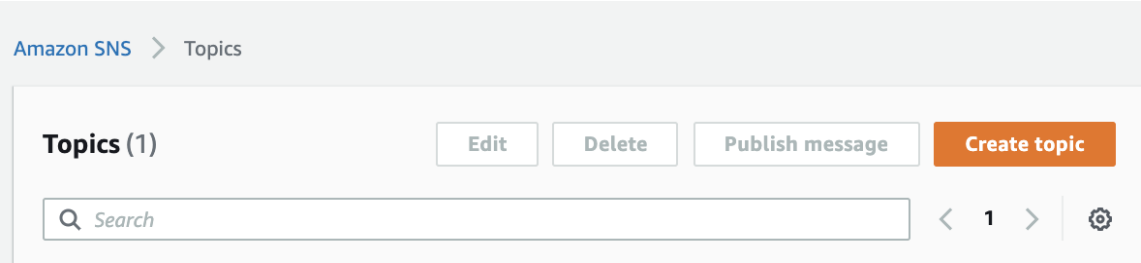
First, we will set up a topic for notifying our email address that we will then be attaching to our alarm.

1. Navigate to the SNS service.

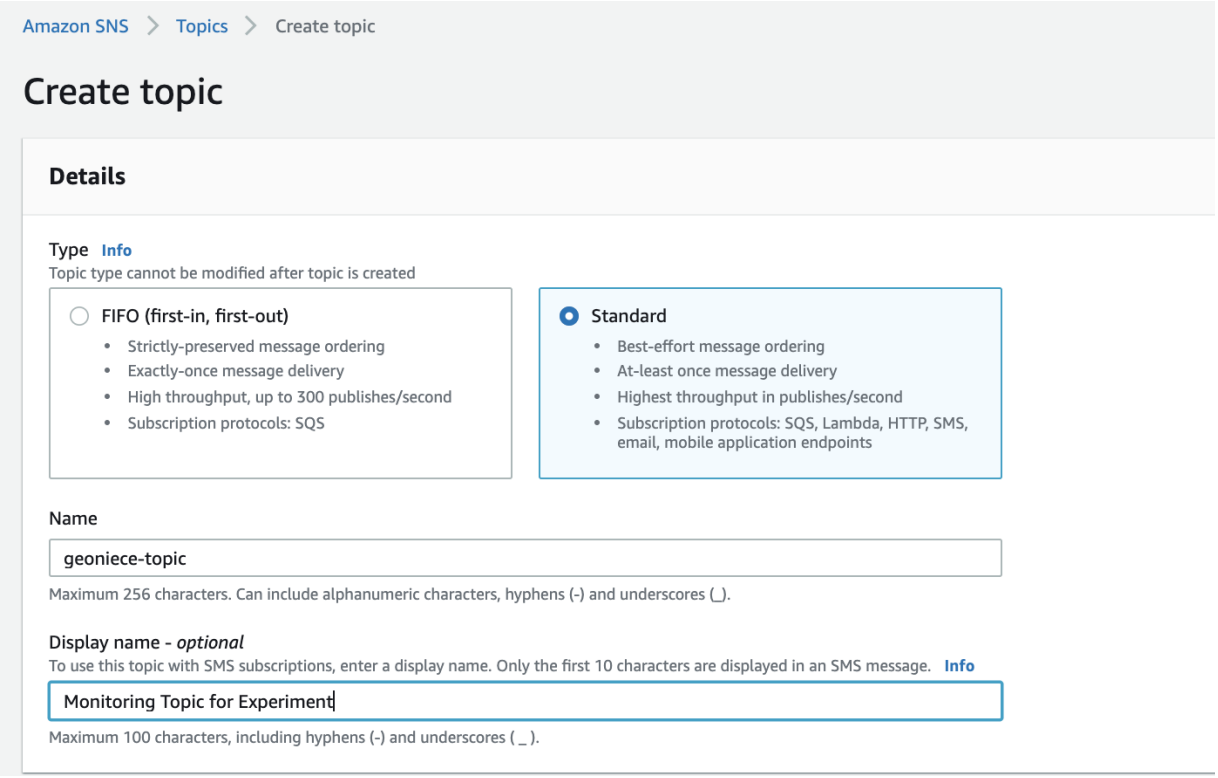


2. On the left side of the screen, select Topics.

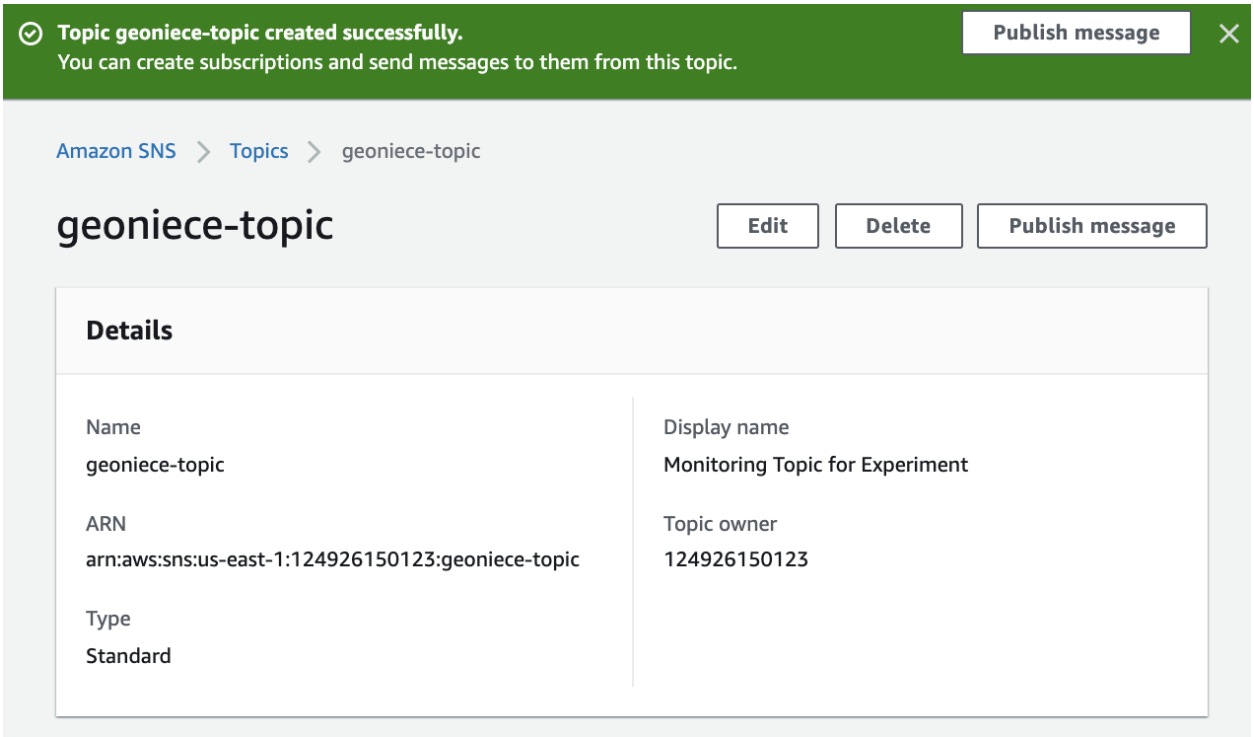
3. Click “Create topic”



4. Choose Standard for the Topic Type. In the Name field , type a name for your topic that includes your name and optionally a Display Name of Monitoring Topic for Experiment. Scroll to the bottom of the screen and click “Create topic”

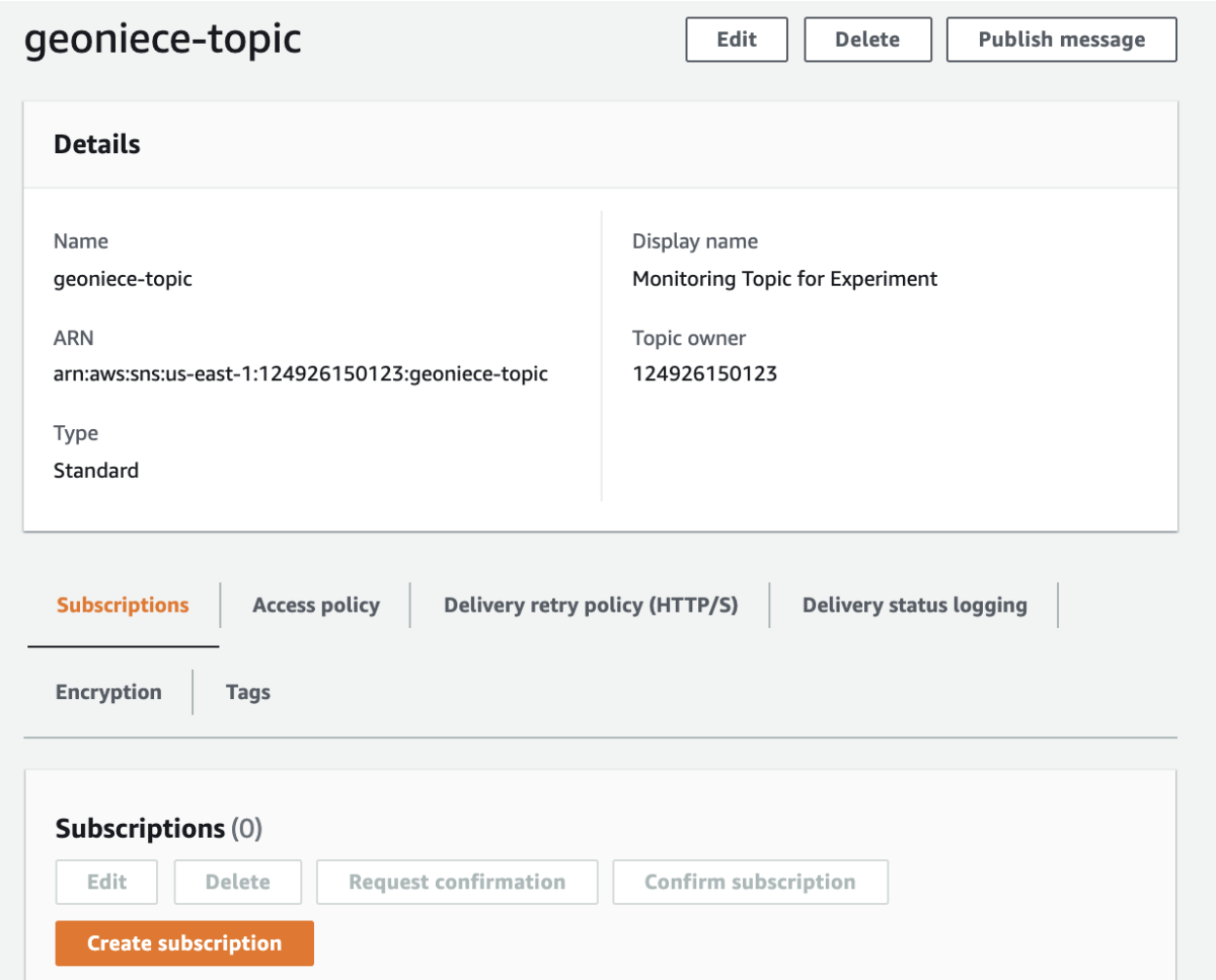


5. Creating the topic will bring you to the topic's specific dashboard.

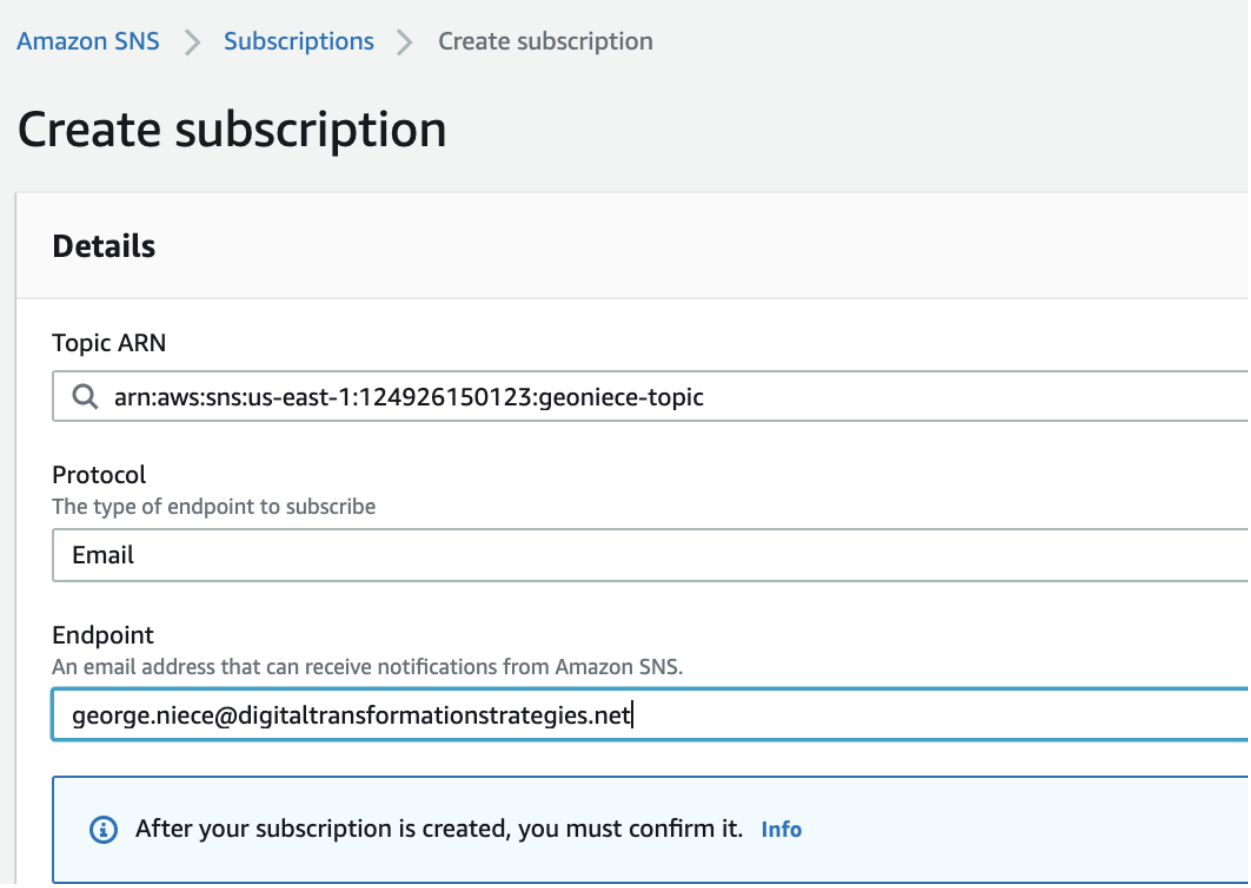


Create Subscription

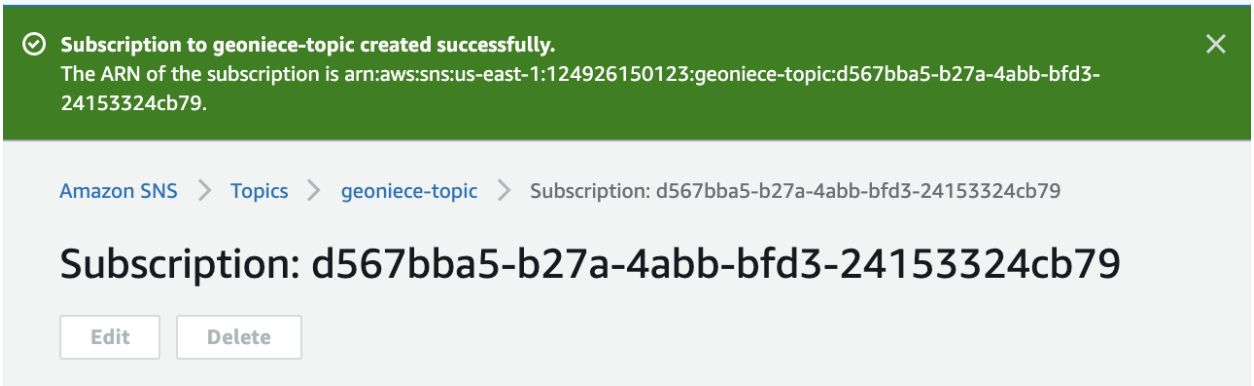
1. Click “Create subscription” on the right side of the screen.



2. In the Protocol drop down select Email and enter a working email address you are able to access. Utilize a non-business email if there may potentially be a spam filter that will block the SNS messages. Click Create Subscription.

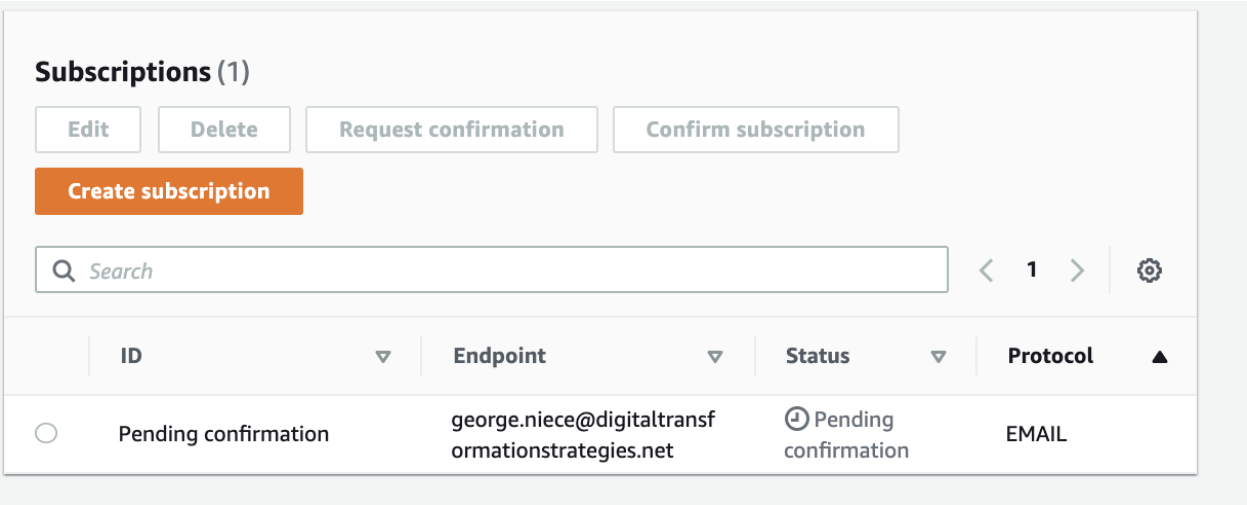


3. The subscription should show the created messaging.

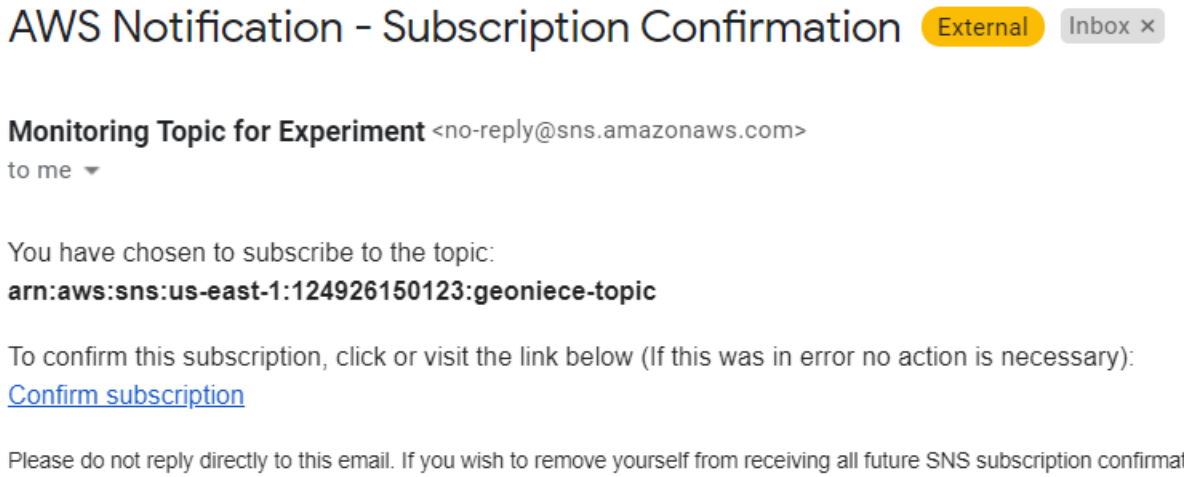


4. Click on the topic name in the breadcrumb navigation.

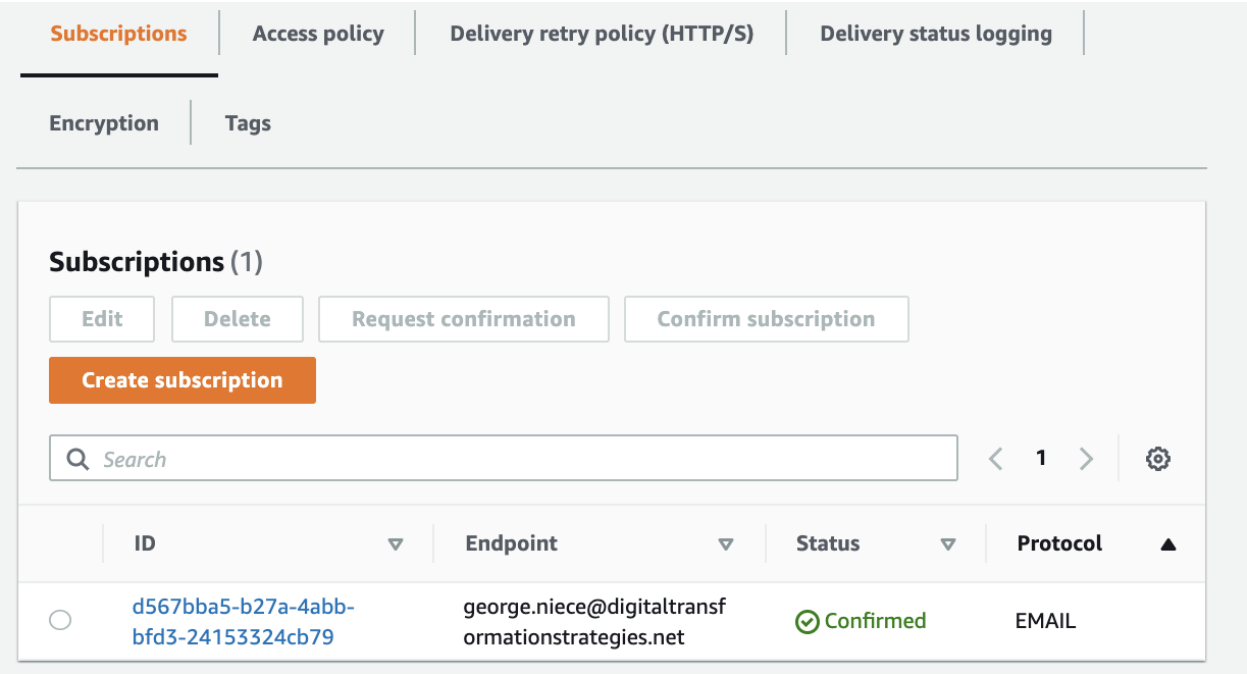
5. Scroll to the subscription section, and our new subscription will be displayed with the status Pending



6. A verification email will be sent to your address with the subject “AWS Notification – Subscription Confirmation”. Open the email and click the Confirm Subscription link.



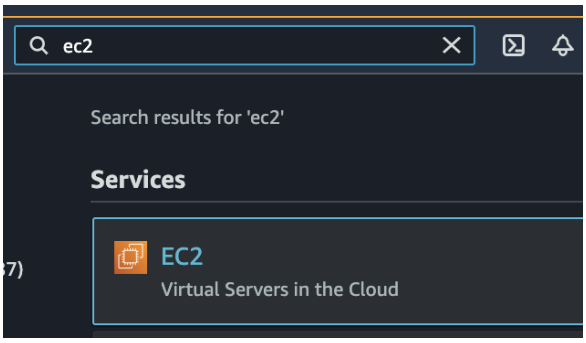
7. Your subscription should now be Status “Confirmed” and not “PendingConfirmation” under the Subscriptions section in the SNS console.



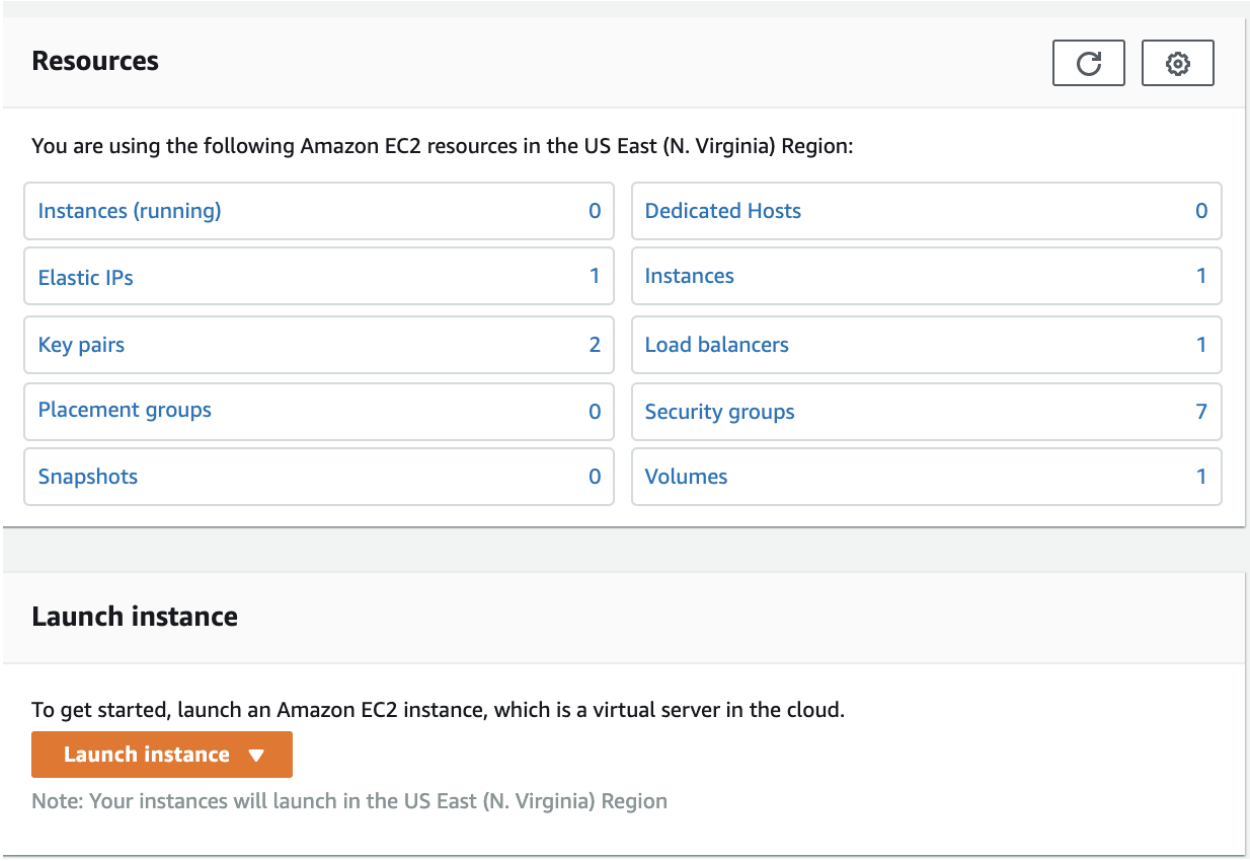
Launch Elastic Compute Cloud (EC2) Instance/VM

In this step you will launch an EC2 instance and configure the User Data to install and launch the stress tool. The stress tool will begin simulating CPU load 5 minutes after the instance launches to allow you time to configure the CloudWatch Alarm.

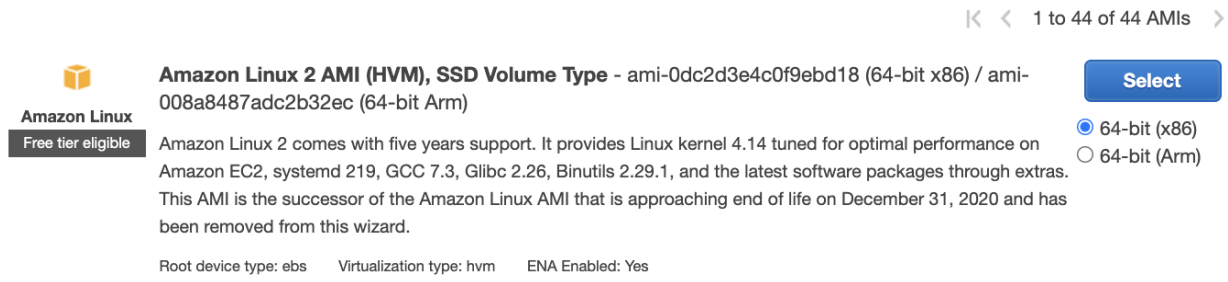
1. Navigate to the EC2 Service through the service menu.



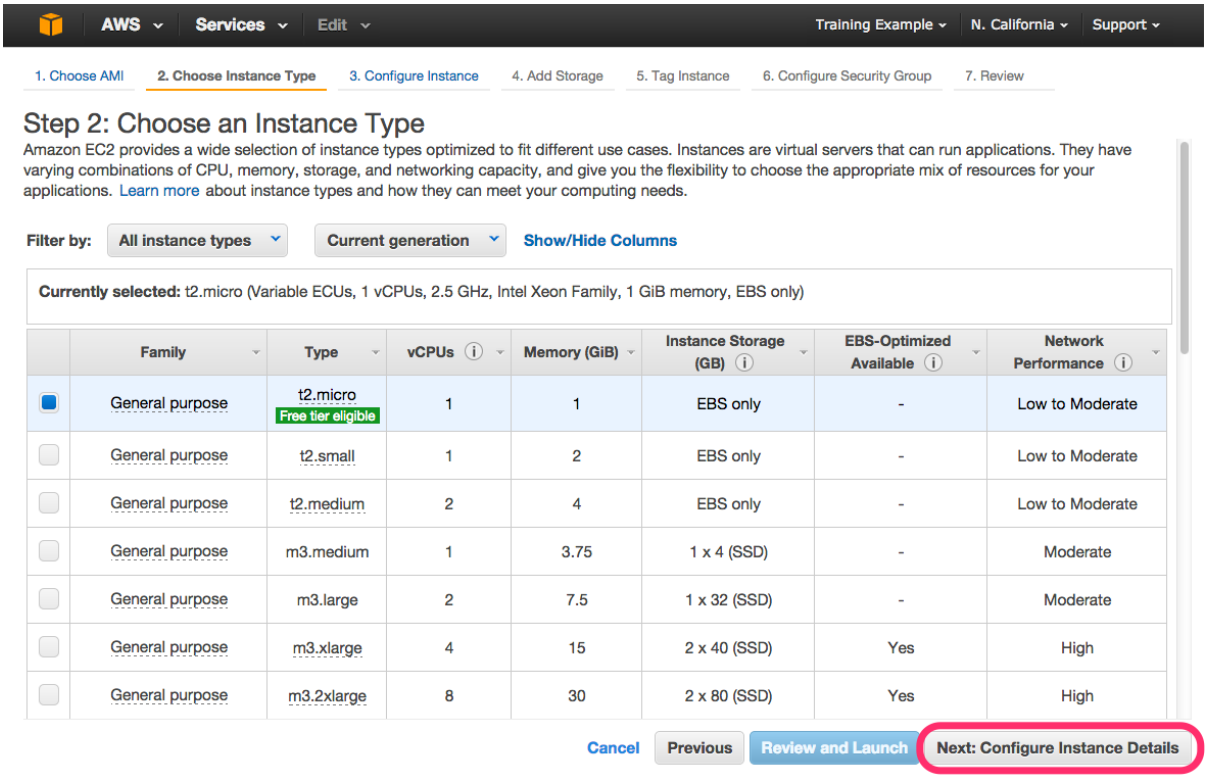
2. Click on Launch Instance



3. In the Choose AMI section, select the “Amazon Linux 2 AMI” and (x86) radio input and click Select



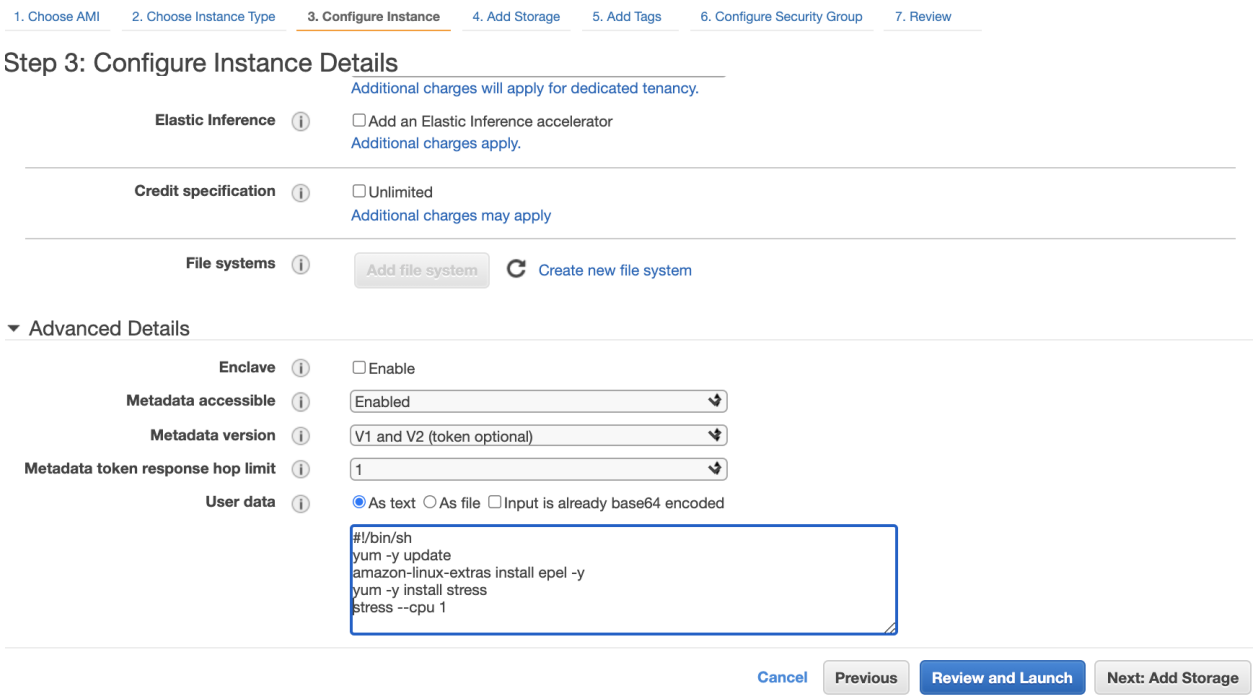
4. Select the General purpose t2.micro instance type and click Next: Configure Instance Details



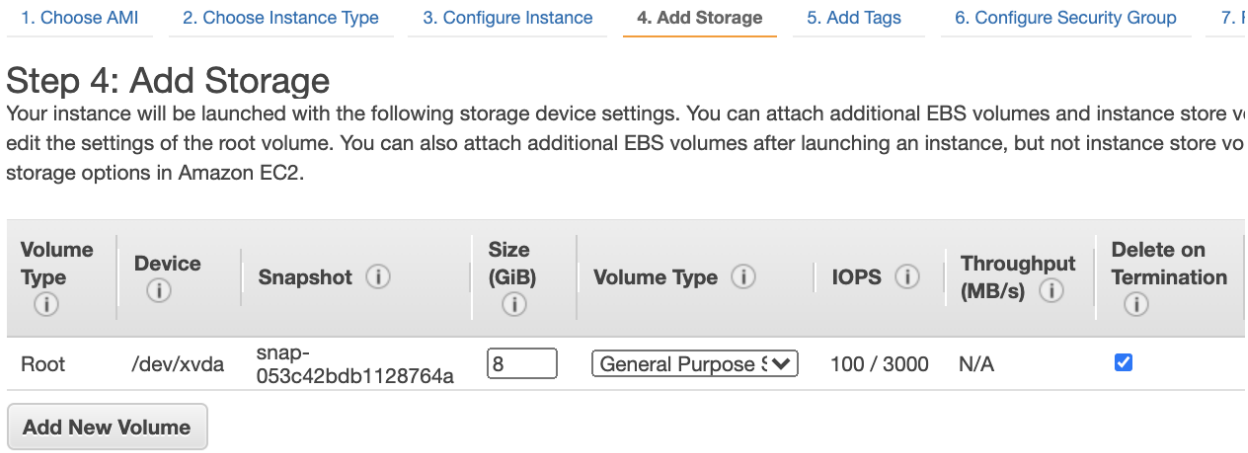
Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
General purpose	t2.small	1	2	EBS only	-	Low to Moderate
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

5. Now we will add a script that will create a test stress script to simulate hits on your instance. Still on the Configure Instance Details page, expand the Advanced Details section at the bottom of the page, and type the following initialization script information into the User Data field (this will automatically install and start the stress tool):

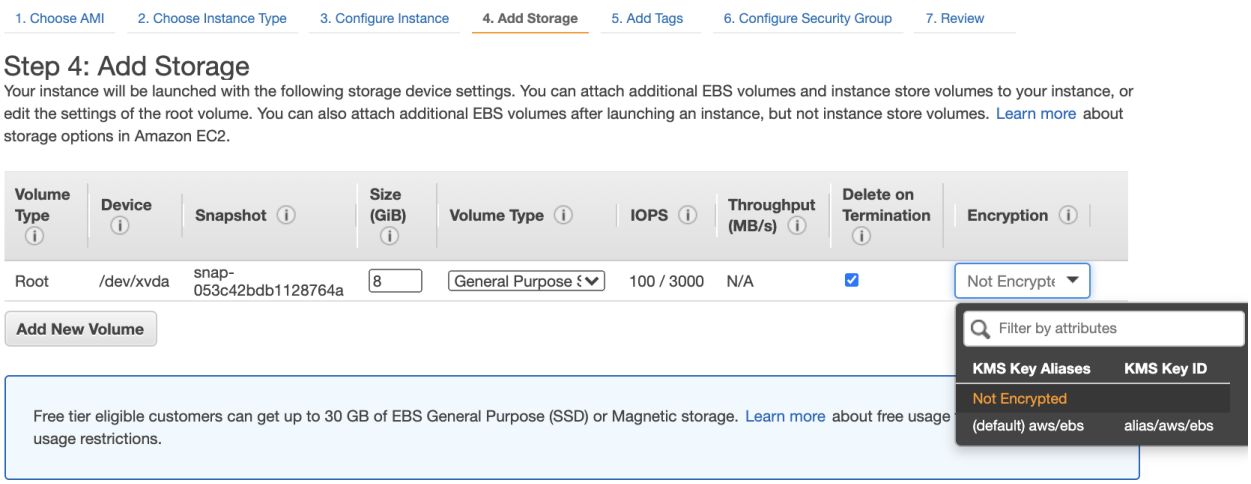
```
#!/bin/sh
yum -y update
amazon-linux-extras install epel -y
yum -y install stress
 stress --cpu 1
```



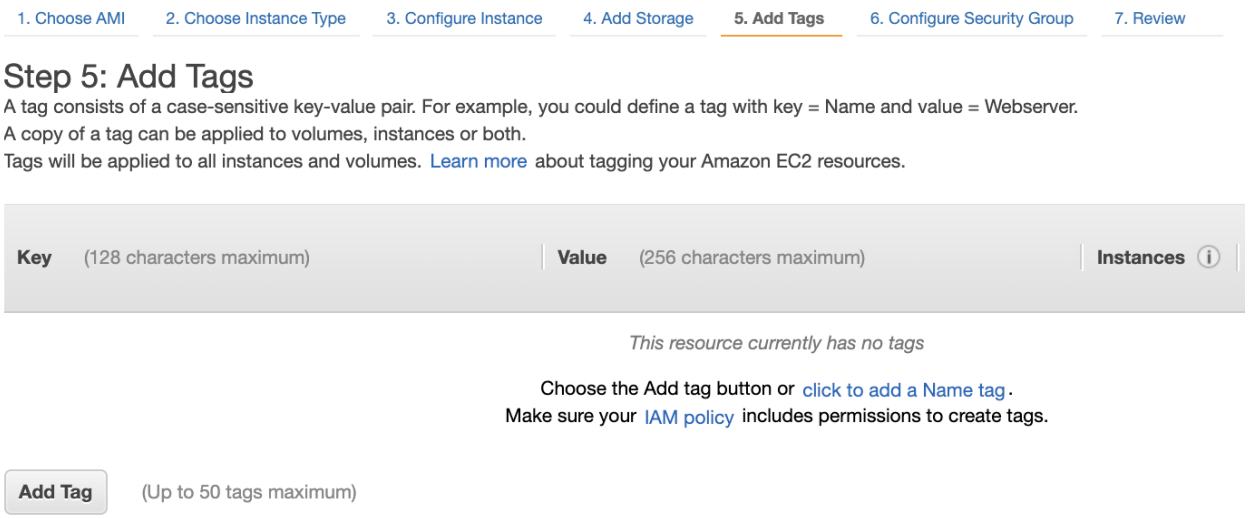
6. Click Next: Add Storage



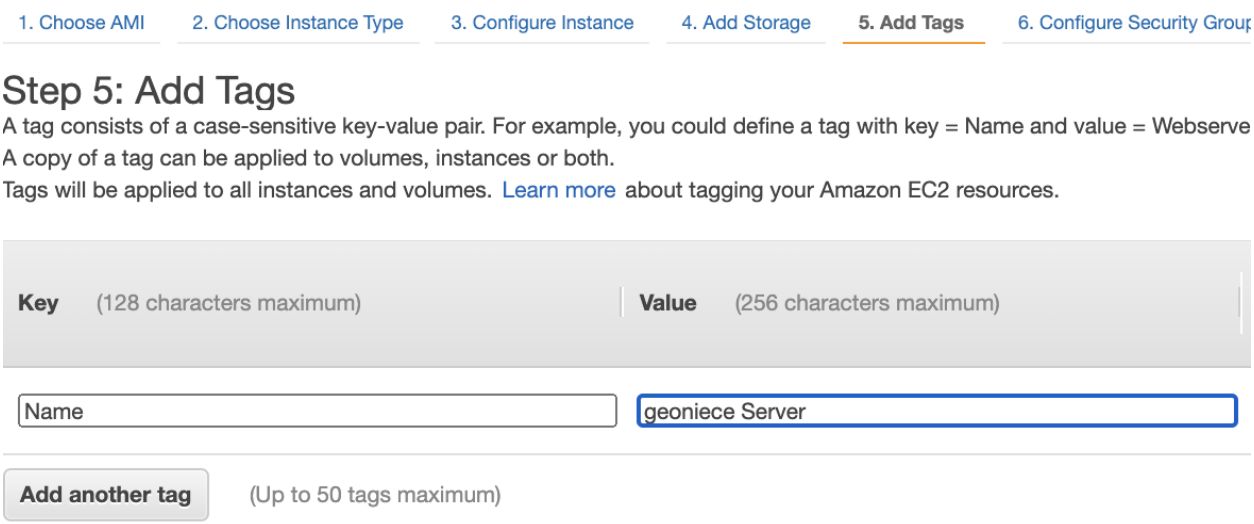
7. On the Add Storage we note options that are very important for data considerations. Note that Delete on Termination is specified. That means that this instance will have the disk deleted if it were terminated. In the case, where an instance backing store needed to be saved when the instance attached to it is deleted we'd uncheck this option. We'll leave it default for our experiment. The other option noted is the Encryption. We won't encrypt this disk since it will be ephemeral and have nothing of note on it for our experiment.



8. Click Next: Add Tags to accept the default Storage Device Configuration.



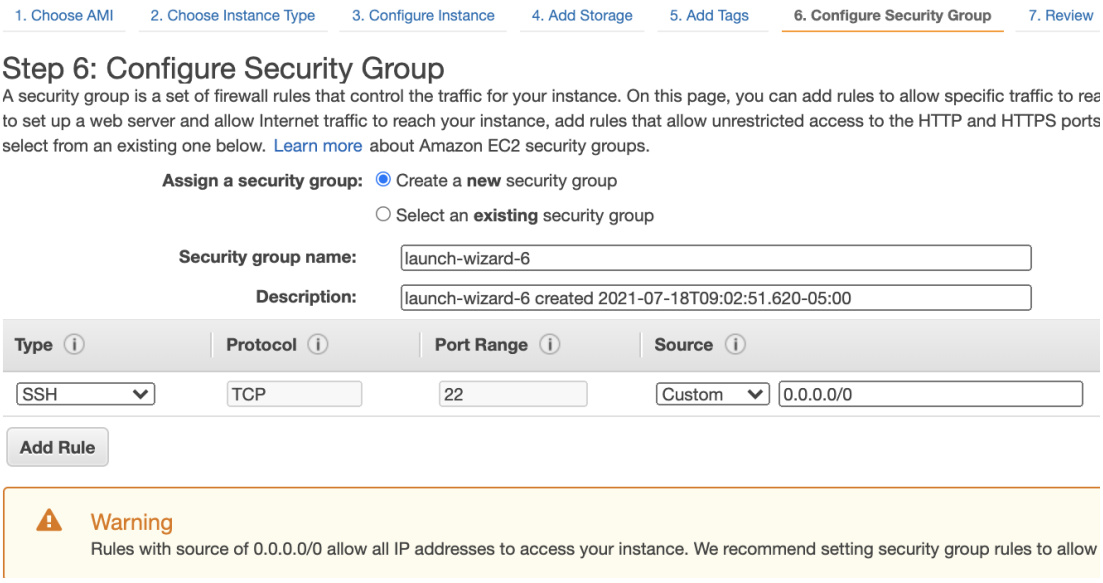
9. Click Add Tag.



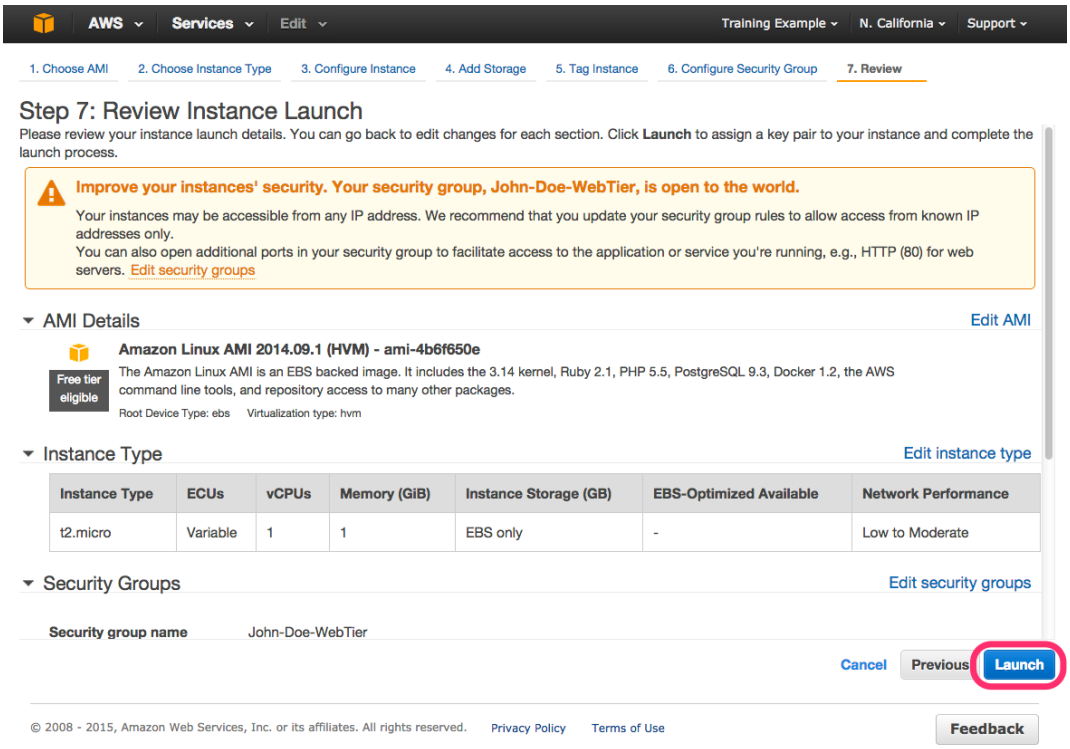
10. Write “Name” in the Key input. Then choose a reasonable value for your instance. This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment. Note that this action will tag instances, volumes, and network interfaces for this virtual machine.

    For this lab, you can name yours in this format: “[Your Name] Server”. Then click Next: Configure Security Group

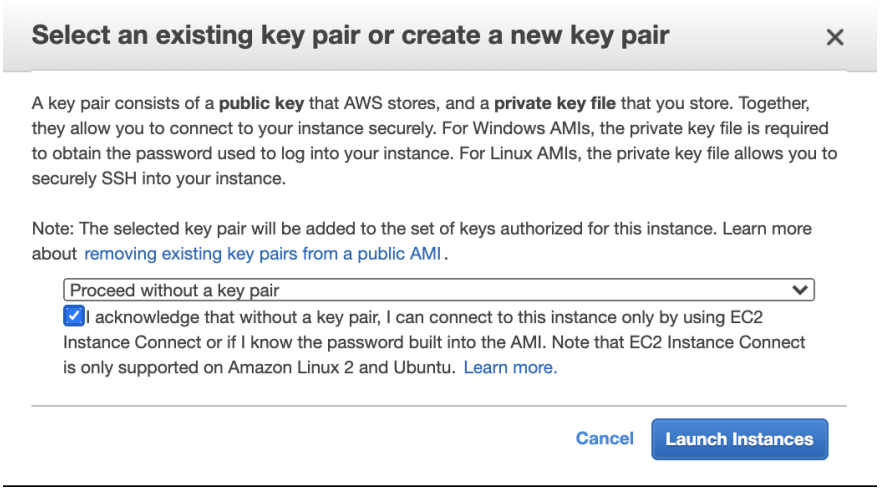
11. Leave the security group as noted. We’ll launch without a keypair which will only allow this instance to be connected with EC2 Connect through the console and not directly from outside via SSH. Then click Review and Launch.



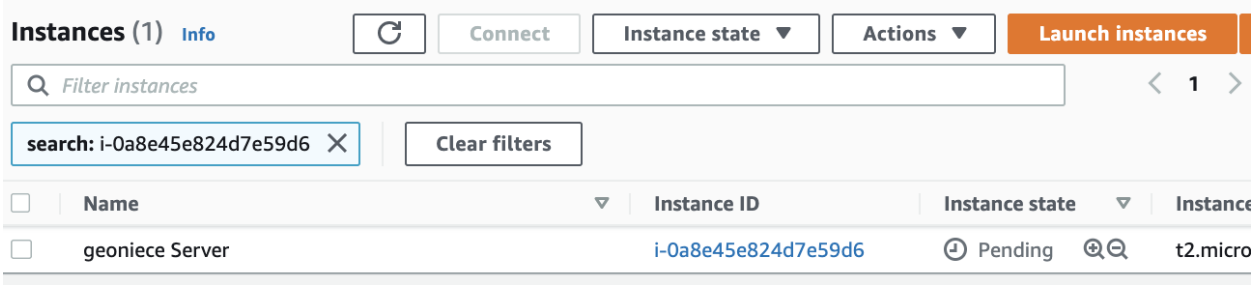
12. Review your Instance Launch Configuration, and then click Review Launch.



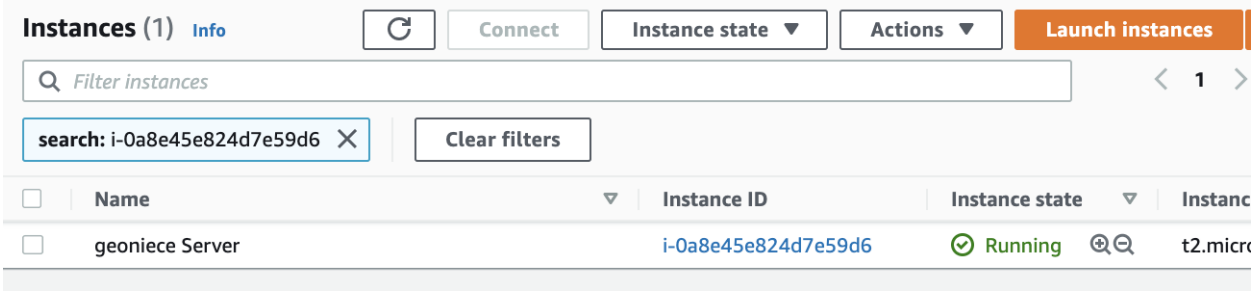
13. In the drop down choose “Proceed without a keypair” and click Launch Instances.



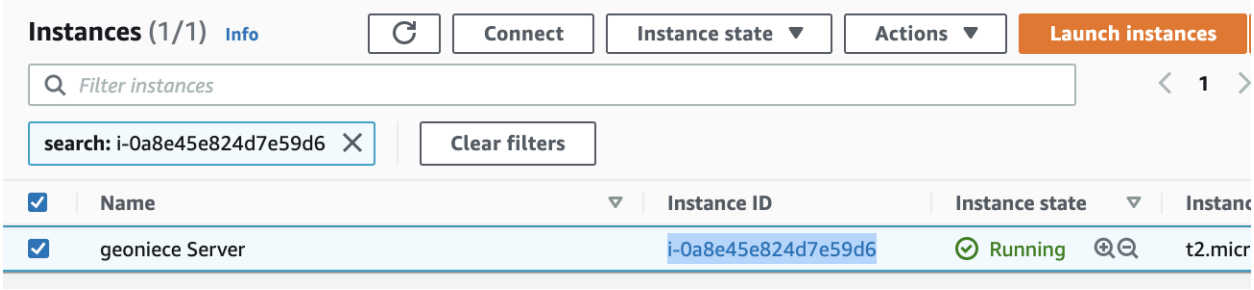
14. Click View Instances button in the lower right-hand portion of the screen to view the list of EC2 instances. Once your instance has launched, you will see your server as well as the Availability Zone the instance is in.



15. If the instance is Pending, wait a minute or two and click the refresh until you see the instance state as Running

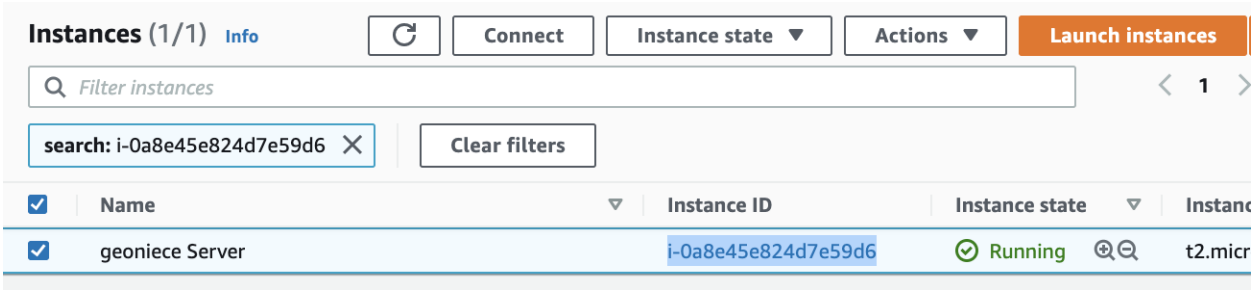


16. Copy the Instance ID and save that in a text file or other location, as we'll need that later in the experiment to configure our CloudWatch Alarm.

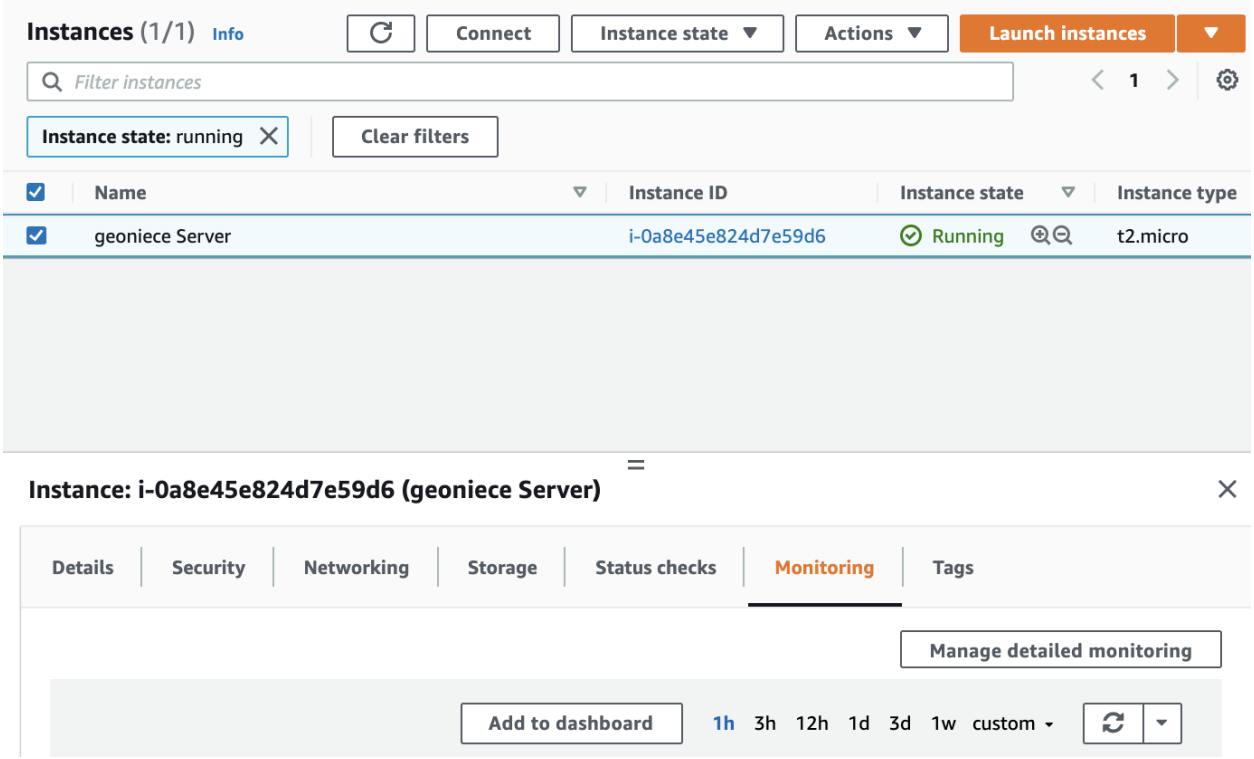


Configure a CloudWatch Alarm

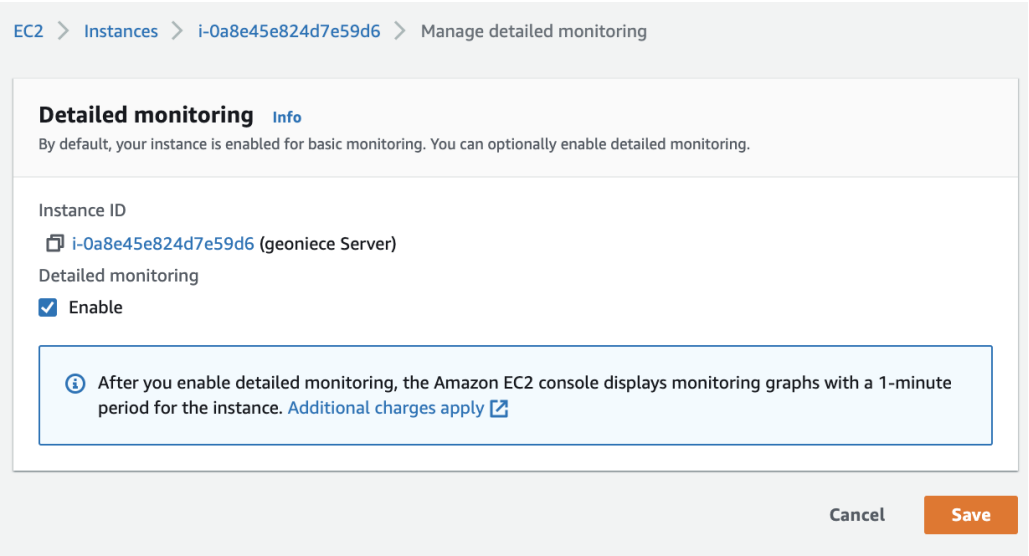
1. In the EC2 Console, click the checkbox next to your server name to view details about this EC2 instance.



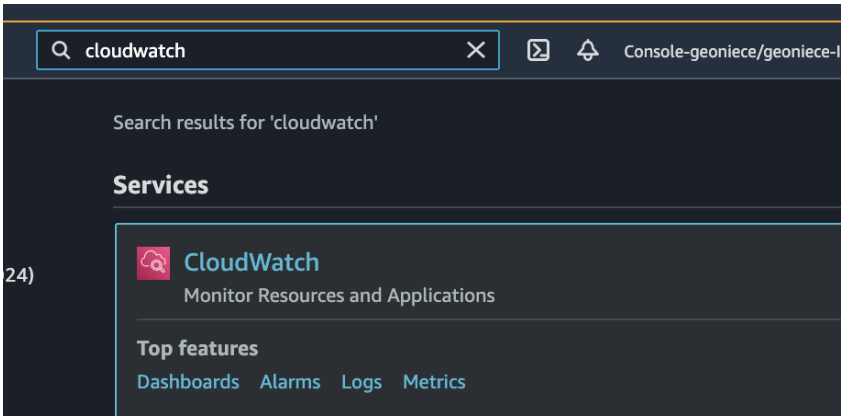
2. Click the Monitoring tab and then click Manage detailed monitoring.



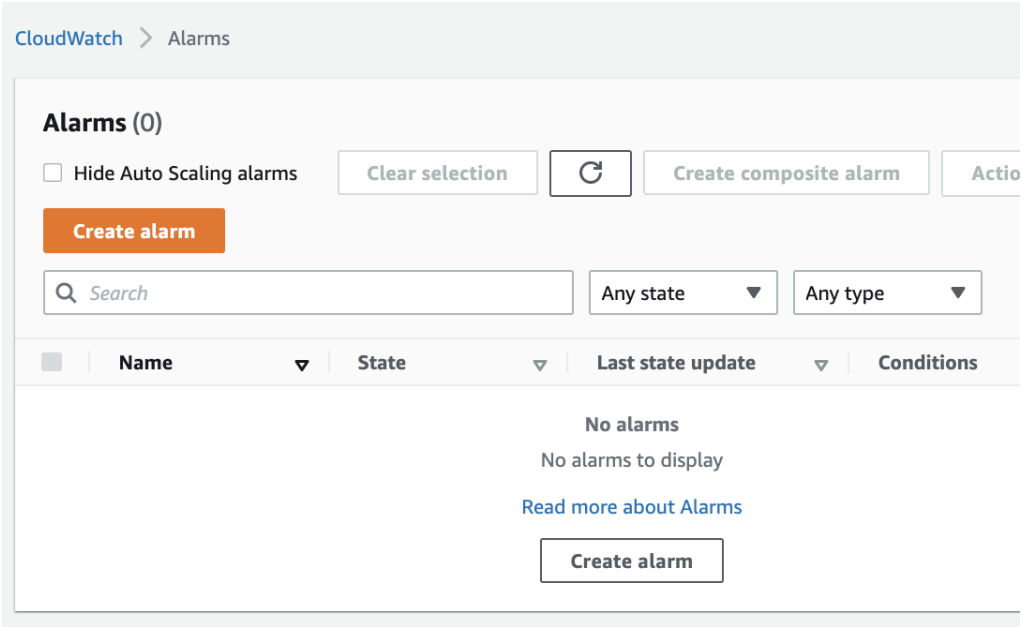
3. Enable Detailed Monitoring by checking Enable checkbox and choosing Save. This will update the CloudWatch configuration to provide monitoring data at a 1 minute interval vs. the default of 5 minutes.



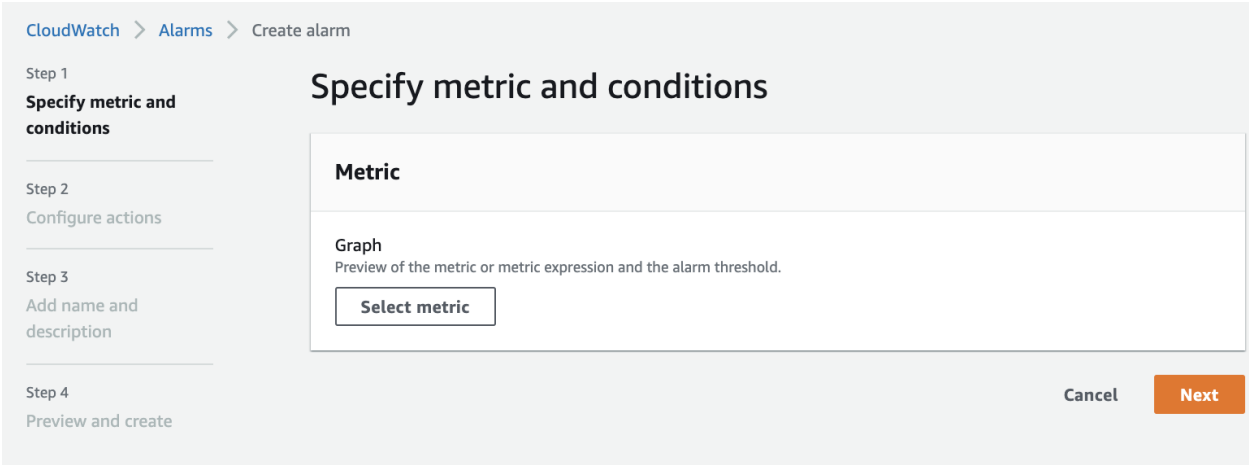
4. Navigate to the Cloud Alarms console



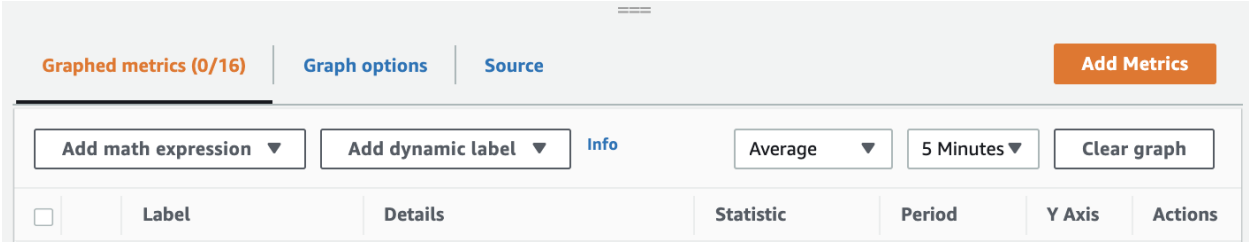
5. Click Create Alarm. If you don't see Create Alarm you need to choose Alarms in the left navigation pane for CloudWatch.



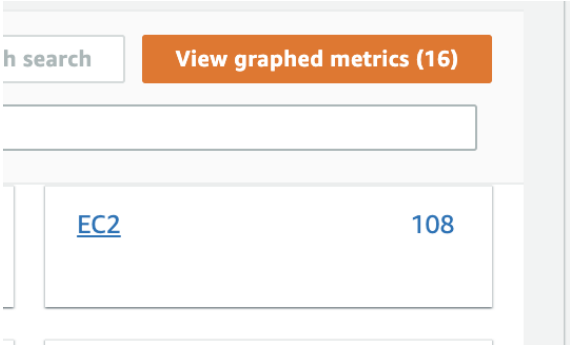
6. Choose Select metric.



7. Choose Add metrics

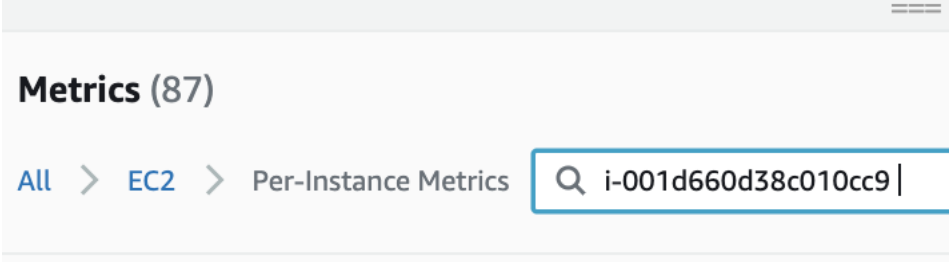


8. Choose EC2 for the metric type

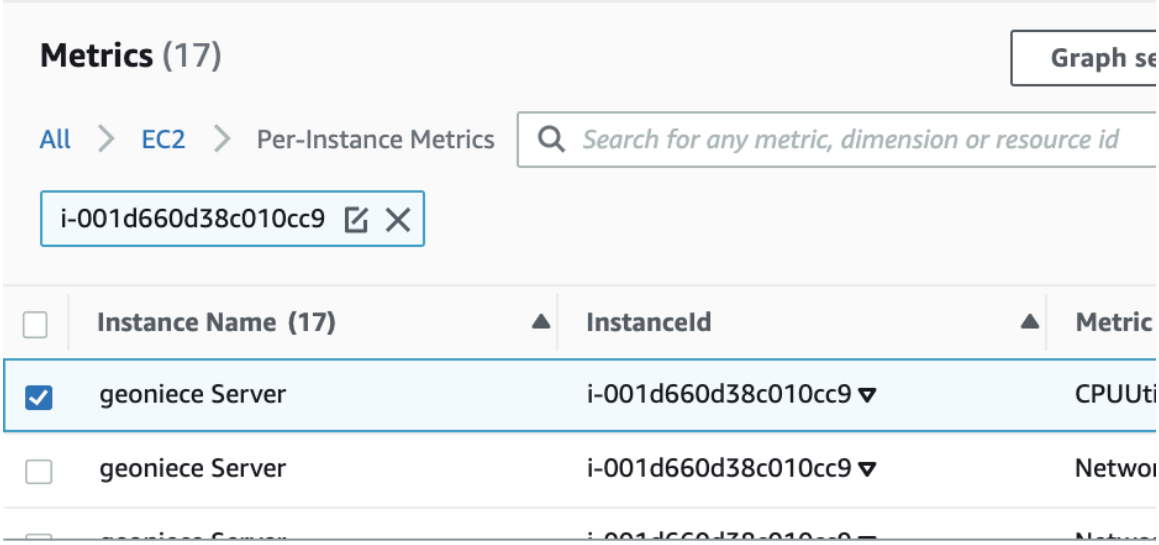


9. Choose Per-instance Metrics

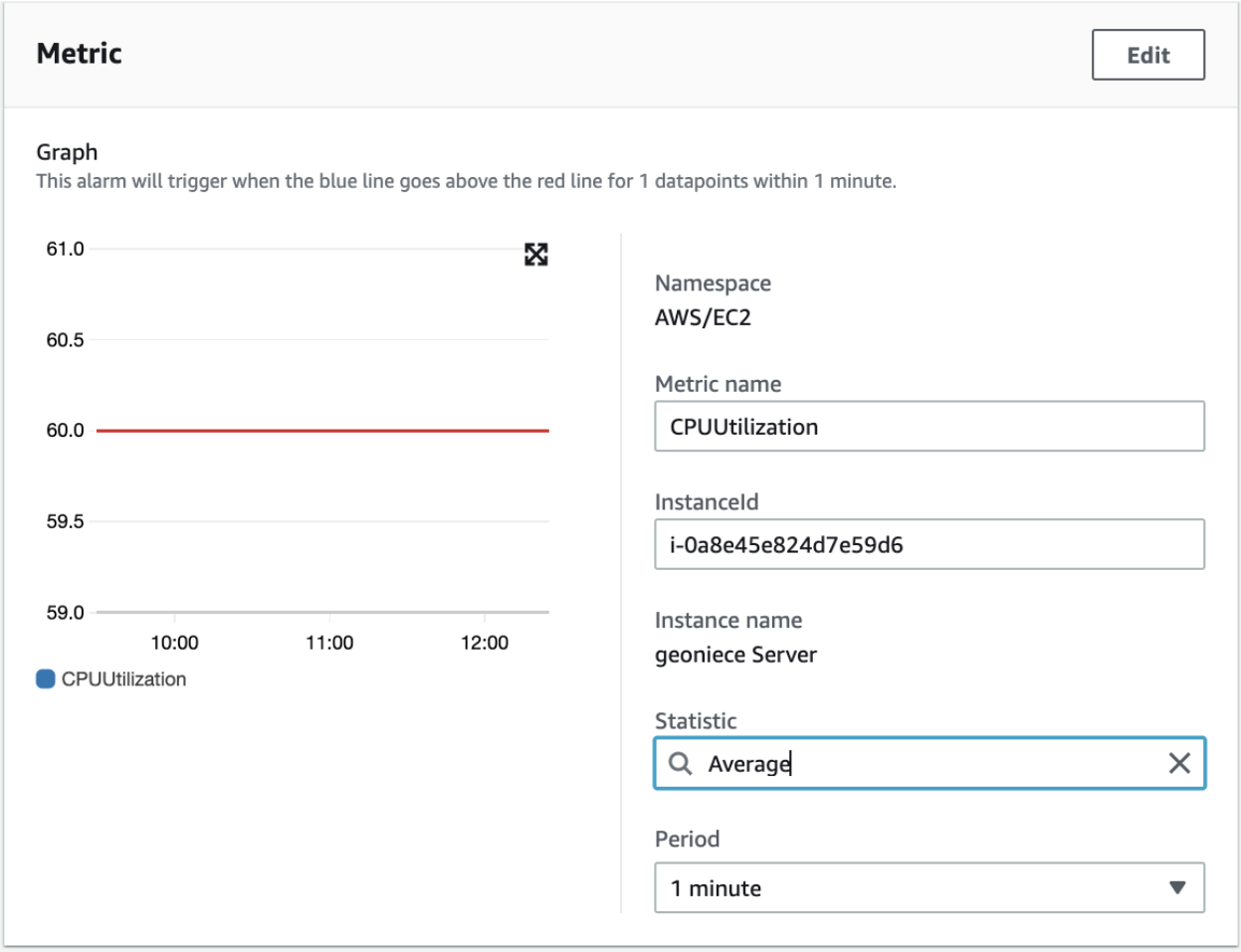
10. Paste the instance ID name we saved earlier.



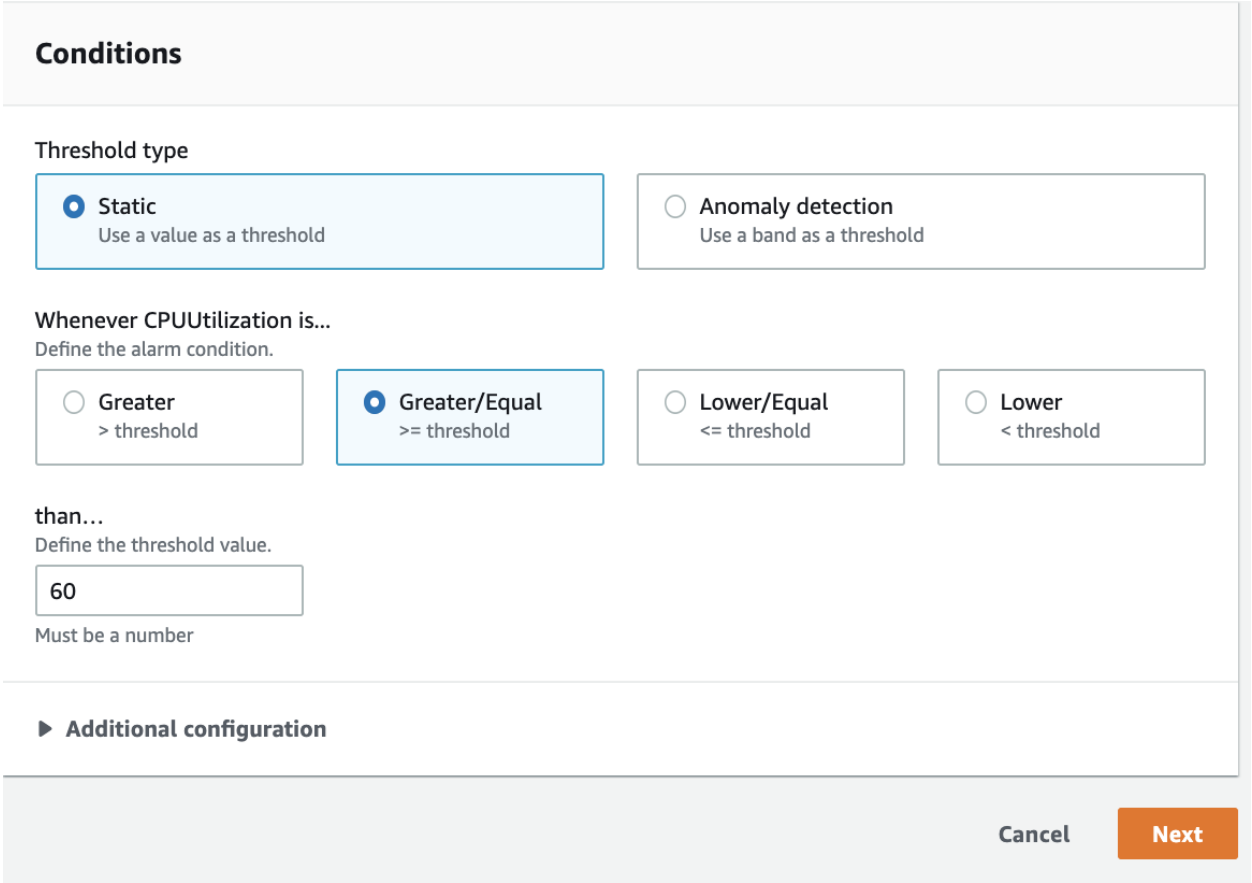
11. Select the checkbox for CPU Utilization as our metric to monitor for this alarm. Choose Select metric.



12. Change the Statistic to Average. For Period select 1 minute.



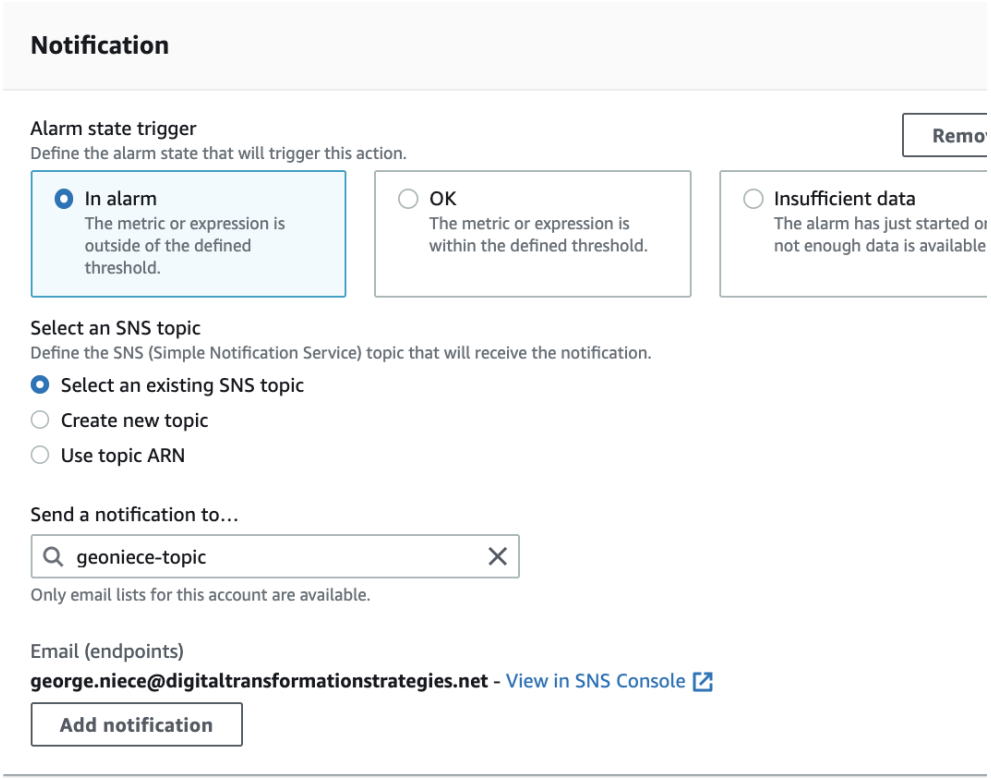
13. Scroll to the Conditions section. Choose Static for the Threshold type, Greater/Equal for the alarm condition for our CPUUtilization metric, and make the threshold value 60.



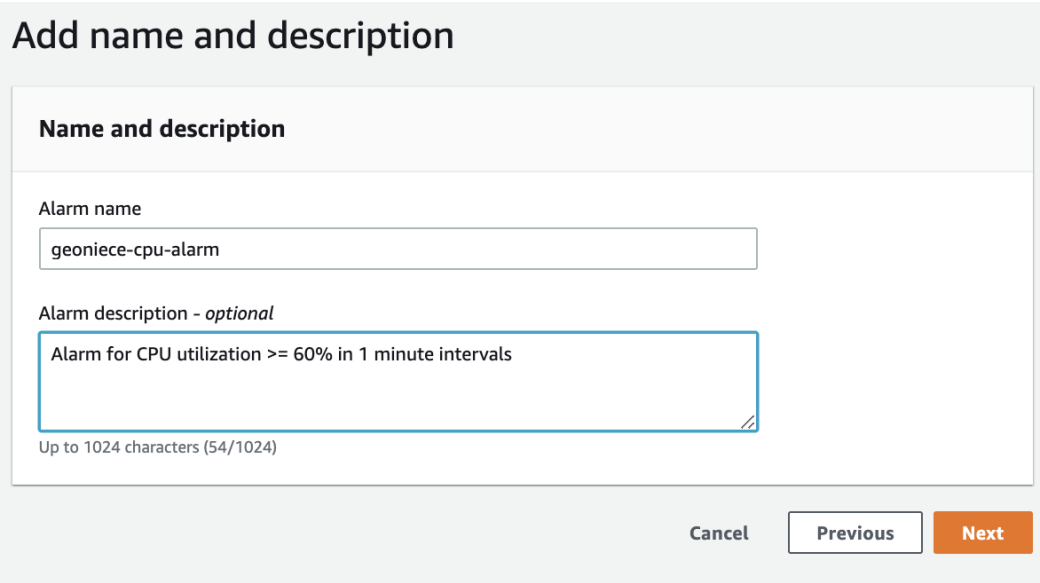
14. Now our Metric configuration will reflect the 60% CPU configuration on our metric graph.

15. On the Configure actions screen we update the Notification to use our SNS Topic that we created in an earlier section of this experiment. For notification chose “In alarm”. Choose the “Select and existing SNS Topic”. To the right of the “Send a notification to:” drop down, select the SNS Topic you created in the earlier section. Choose Next.

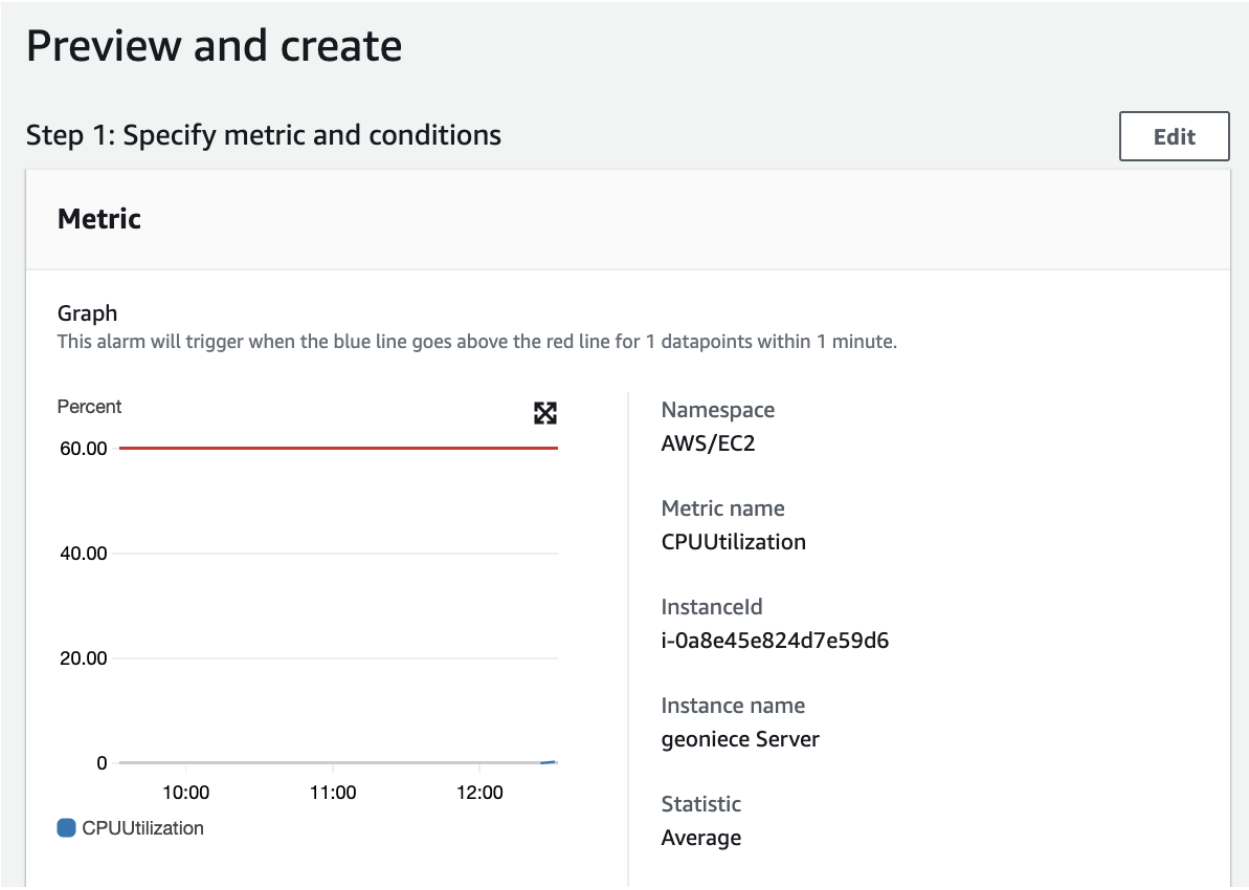
16. Add you're Alarm name, for our experiment we should name it in the format name-cpu-alarm and enter an Alarm description. Choose Next

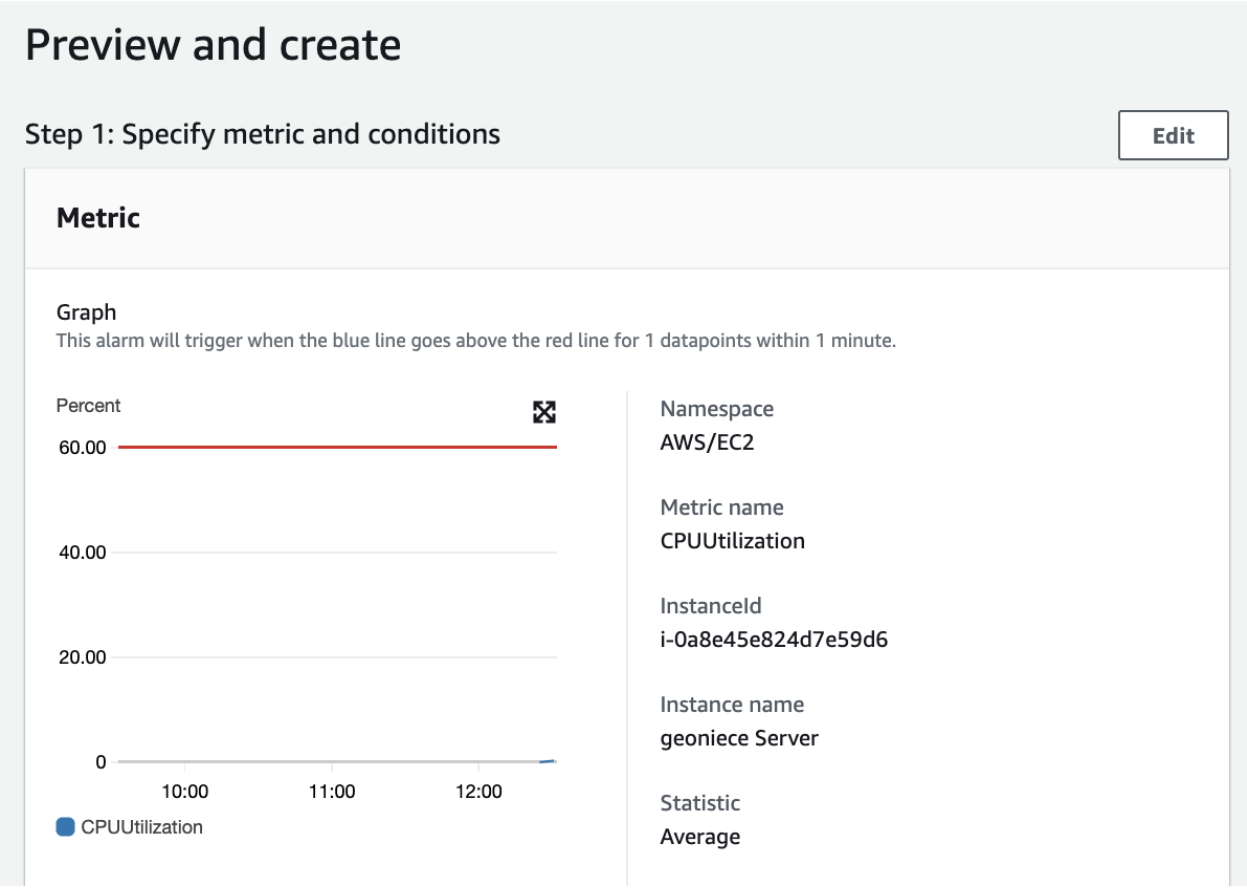


17. Add you're Alarm name, for our experiment we should name it in the format ***name-cpu-alarm*** field and enter an Alarm description. Choose Next

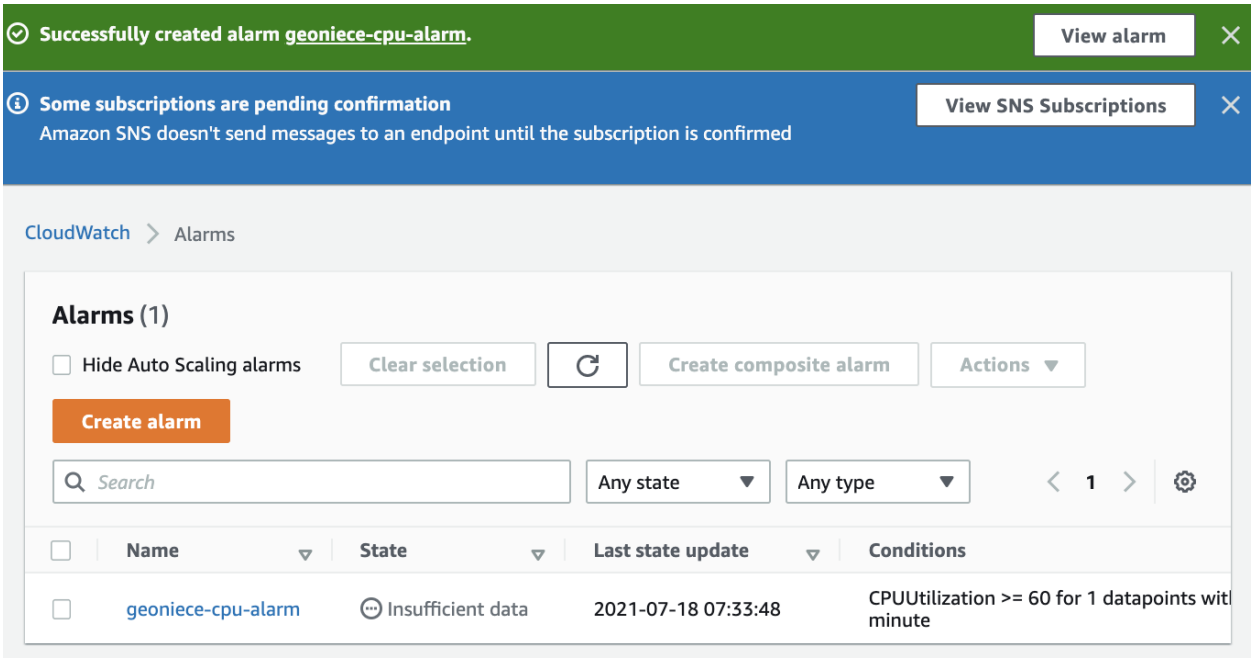


18. On Preview and create screen choose Create alarm





19. Our Alarm should be created.

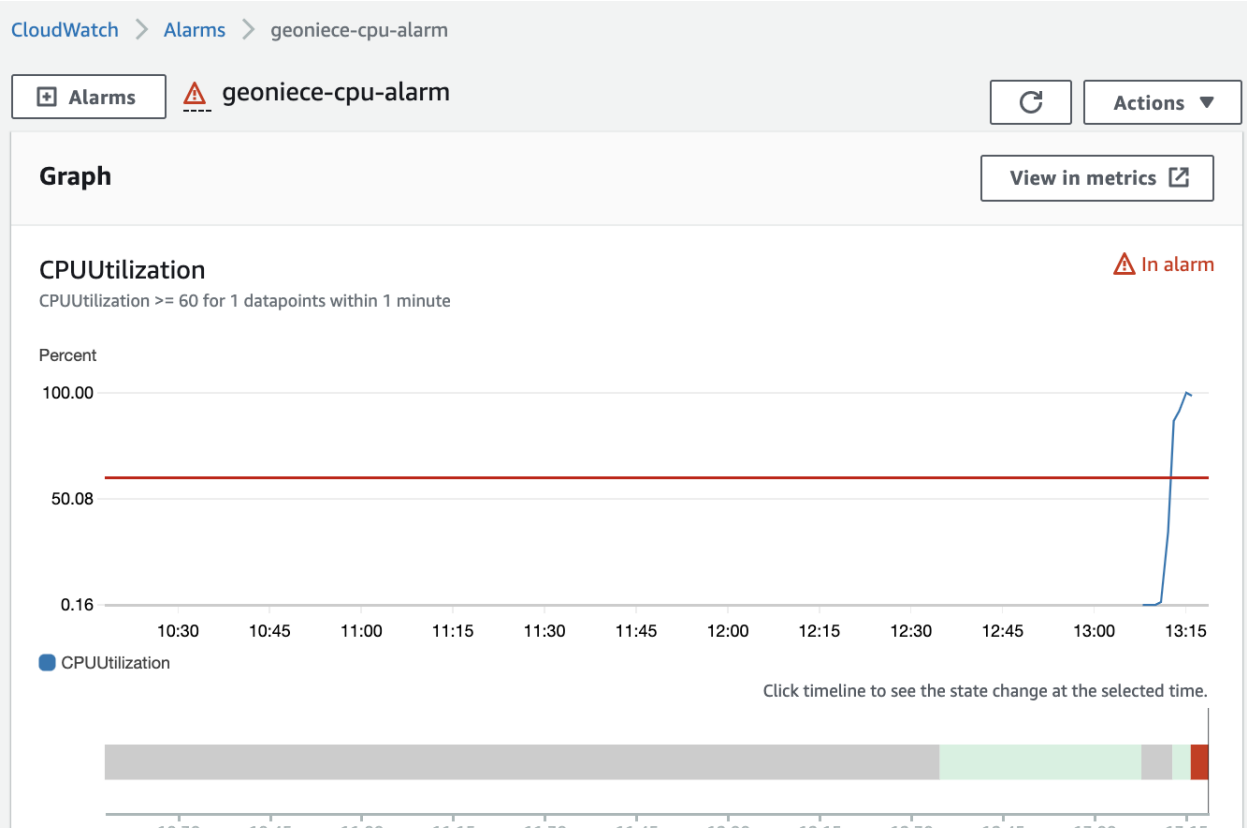


Validate our CloudWatch Alarm

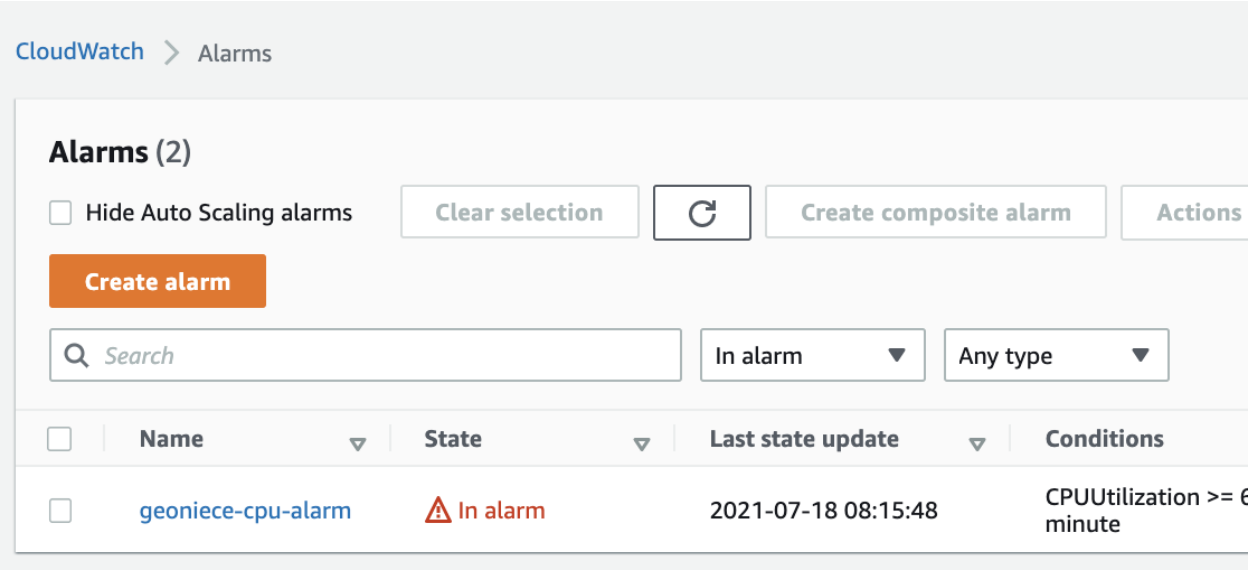
1. Navigate back to the main screen for CloudWatch Alarms.

2. Click Alarms in the left pane of the Console and check the State of your alarm. It most likely says INSUFFICIENT_DATA because you just created it.

3. In the CloudWatch Console. Select Metrics. Select the Graphed metrics tab and change the Period to 1 Minute. Change the graph interval to a custom value of 30m and select Auto refresh of 1min.



4. After 5 minutes, the stress tool will begin to simulate CPU workload and trigger the Alarm once the threshold is reached.

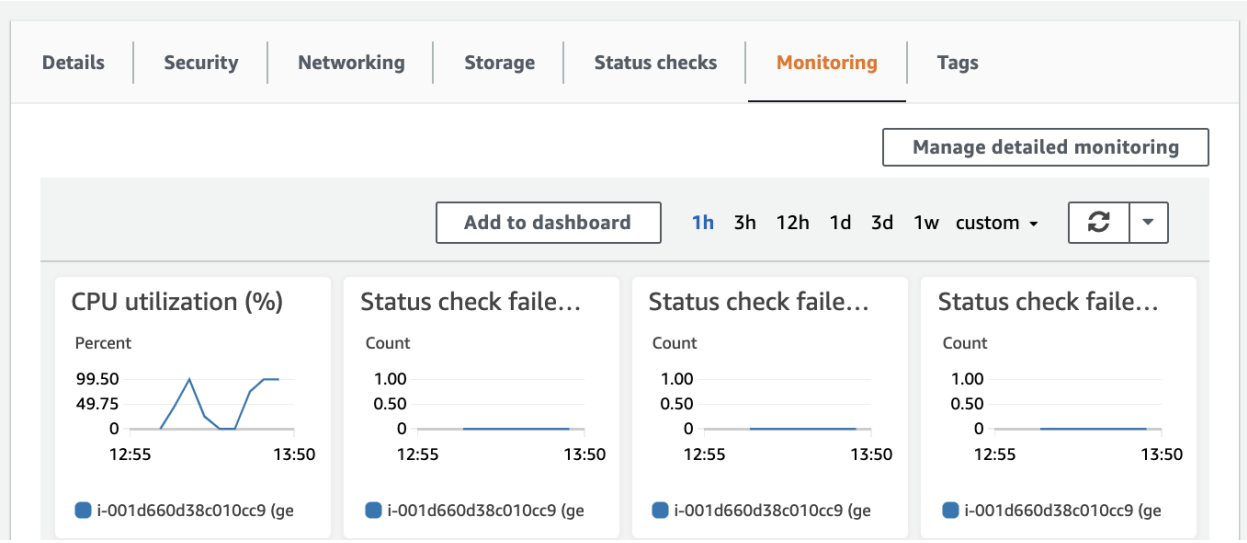


5. You can view the Alarm state in the CloudWatch console under Alarms.

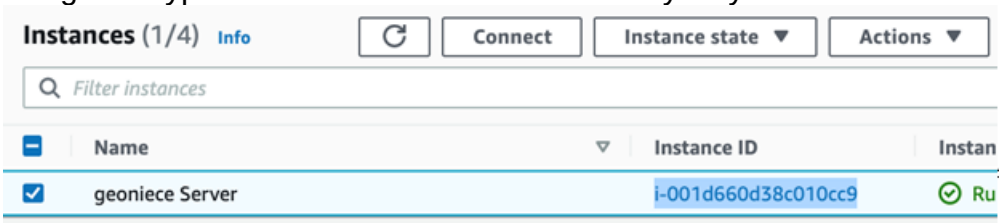
6. We setup an email notification and you will receive an email alert when the Alarm is triggered.

Use EC2 Connect

1. Navigate back to our EC2 Console
2. Select our instance and the Monitoring Tab, we'll see a graph showing that our instance is getting crushed.



3. Select our instance and click Connect. This will invoke the EC2 Connect Tool and we'll see a new browser window with a Terminal session on our instance. Note that since we didn't assign a keypair for this instance this is the only way to connect.



4. Enter the top command to view our instance performance details.

```
$ top
```

5. This will show us a screen similar to the one shown with the stress tool beating the heck our of our instance


```
top - 13:27:05 up 3 min,  1 user,  load average: 1.06, 0.52, 0.21
Tasks:  95 total,   3 running,  52 sleeping,   0 stopped,   0 zombie
%Cpu(s):100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  1006892 total,   377988 free,   116820 used,   512084 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   745224 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 7391 root      20   0    7572     96      0 R 99.7  0.0   1:58.81 stress
 3145 root      20   0  729116  27756  13376 S  0.3  2.8   0:00.10 ssm-agent-worke
 7836 ec2-user  20   0  170896   4376   3828 R  0.3  0.4   0:00.01 top
    1 root      20   0  125640   5444   3820 S  0.0  0.5   0:01.87 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0
```

6. Noted PID for the stress tool that is running in this instance. For example, “7391”
7. Quit the top tool

   Press the letter **q** on your keyboard
8. Kill the stress job

   $ sudo kill -9 7391
9. End our EC2 Connect session

   $ exit

You have successfully configured a SNS Topic, EC2 Instance, Stress test, CloudWatch Alarm, validated our alarm on the instance and notification, and connected to your instance being monitored!

View Notification

1. Browse to your email used for the notification and view your alarm.

ALARM: "geoniece-cpu-alarm" in US East (N. Virginia)

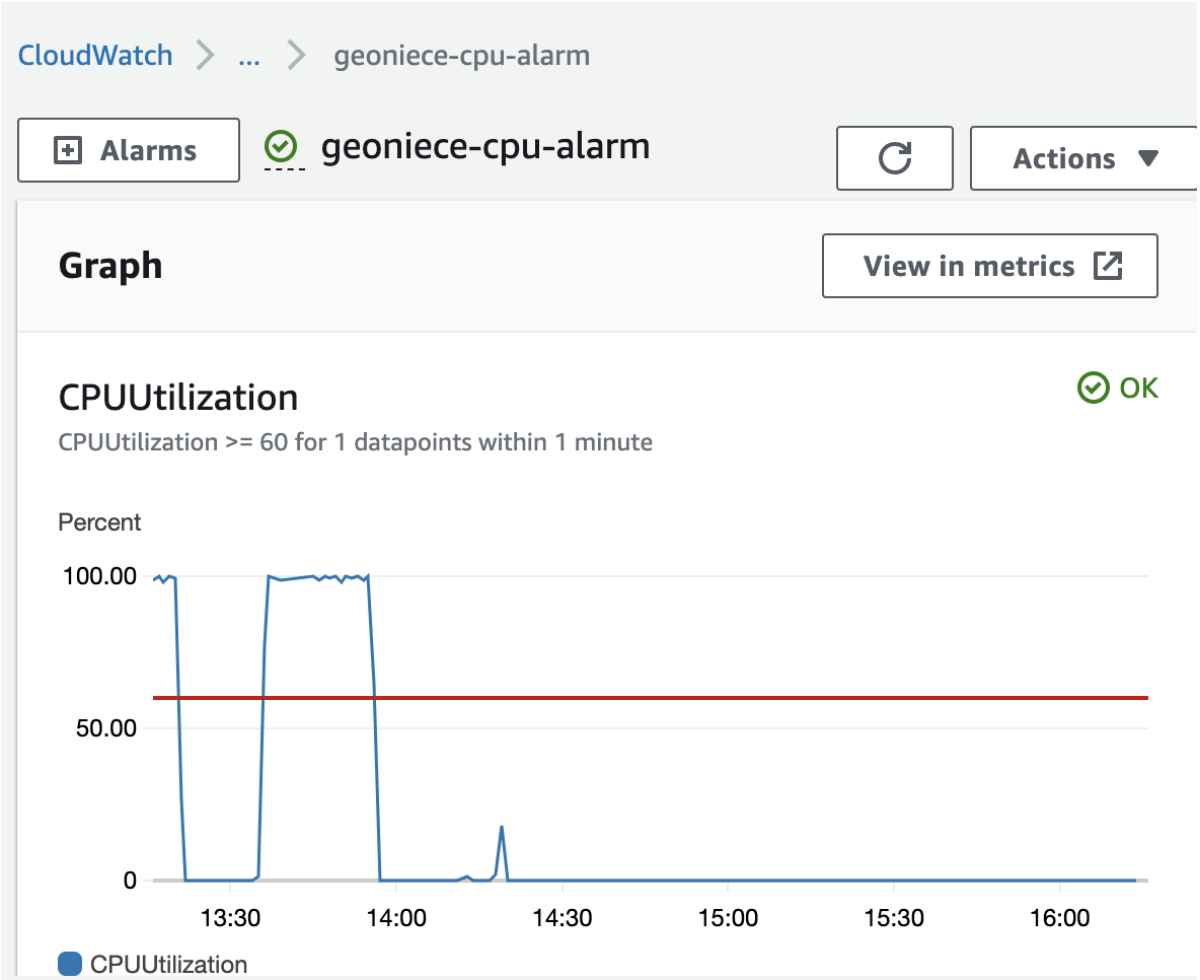
Monitoring Topic for Experiment <no-reply@sns.amazonaws.com>
to me

You are receiving this email because your Amazon CloudWatch Alarm "geoniece-cpu-alarm"
"Threshold Crossed: 1 out of the last 1 datapoints [77.3333333333335 (18/07/21 13:36:00)]
> ALARM transition)." at "Sunday 18 July, 2021 13:38:48 UTC".

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

Alarm Details:
- Name: geoniece-cpu-alarm
- Description: Alarm for CPU >= 60 in 1 minute intervals
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [77.3333333
(minimum 1 datapoint for OK -> ALARM transition)

2. Click on the deep link to be redirected to the alarm view in the AWS Management Console



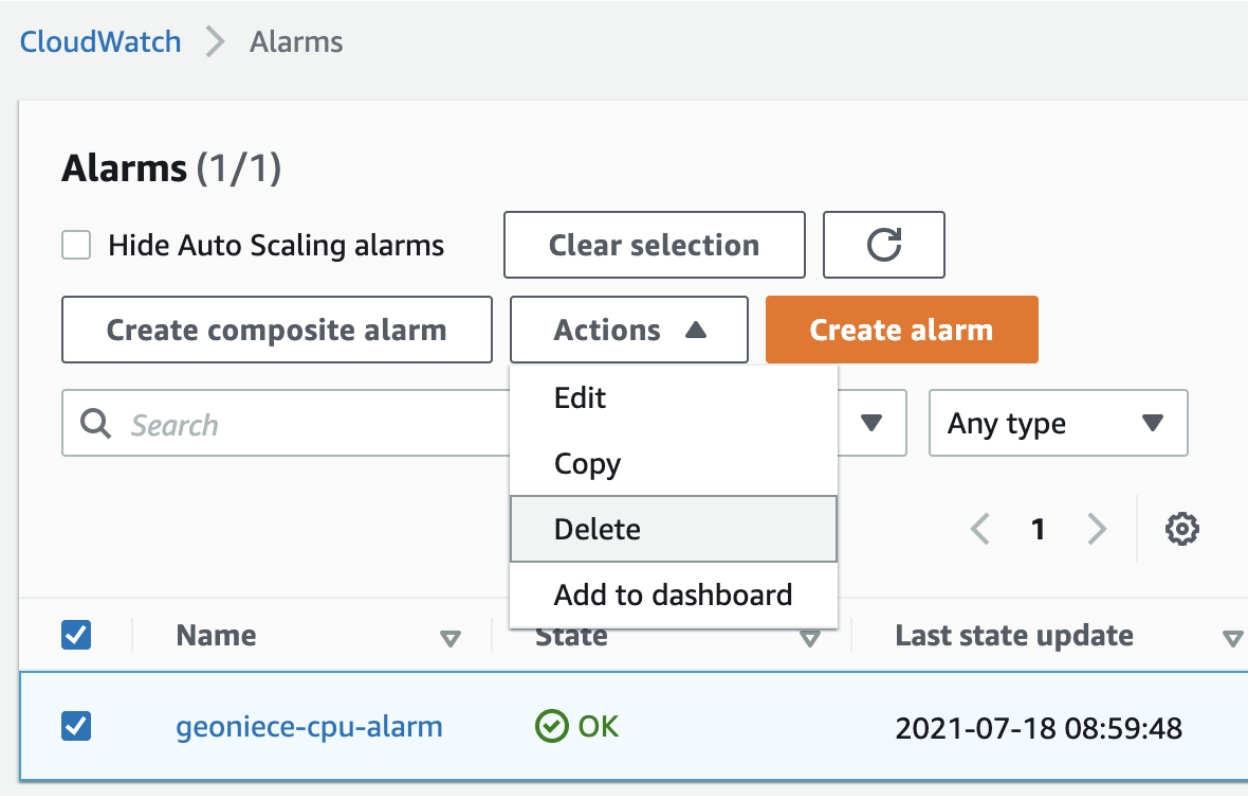
3. Note that after we killed the stress test that our alarm will come back to the OK state.

Clean Up

Be sure to delete the following resources after you are finished:

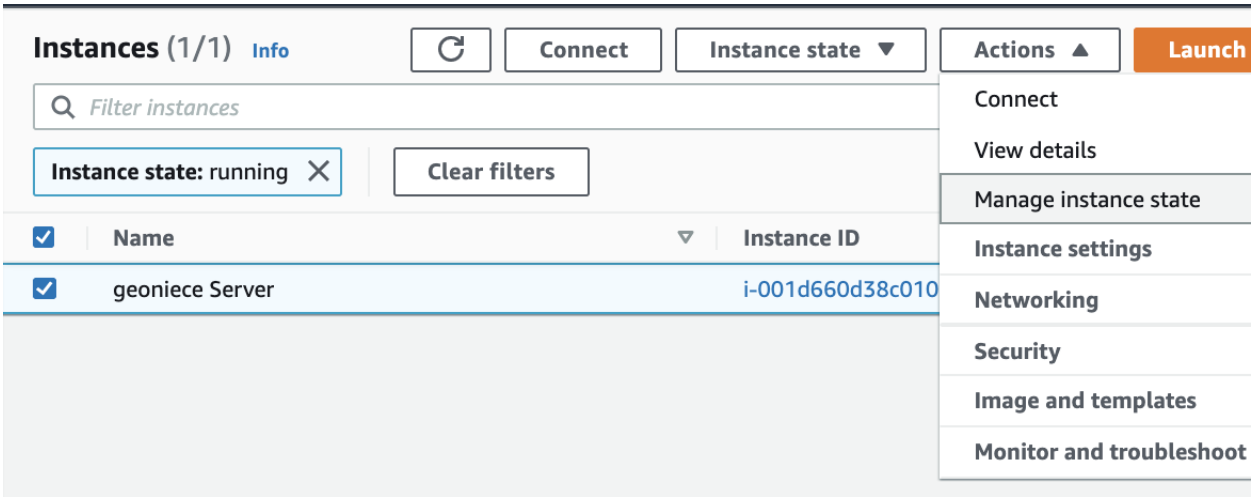
1. Select Delete on your alarm after you are finished.

   Navigate to the Cloud Alarms console, select the checkbox for our alarm, from the Actions drop down select Delete. Choose Delete to permanently delete our alarm.

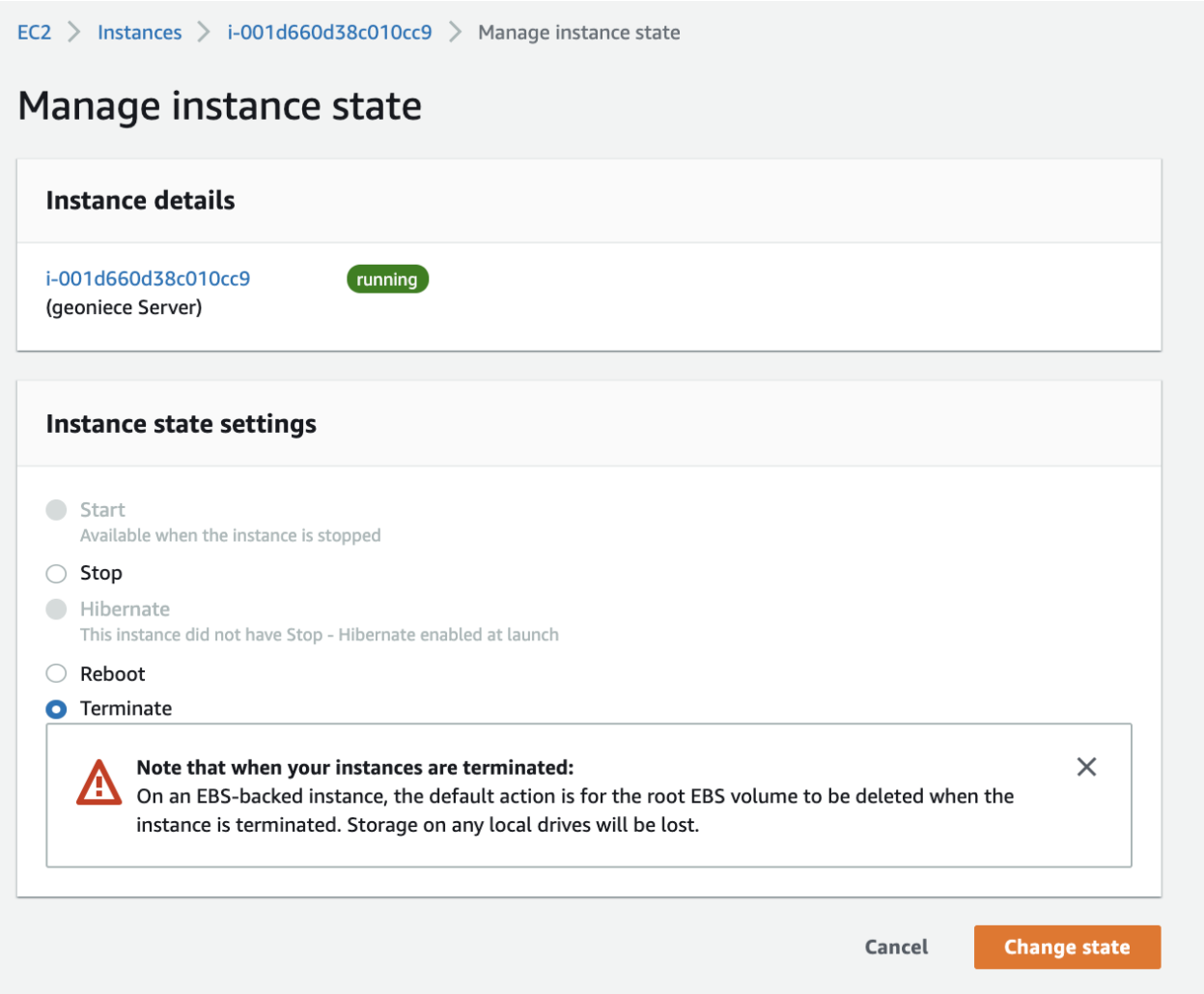


2. Stop and terminate your EC2 instance.

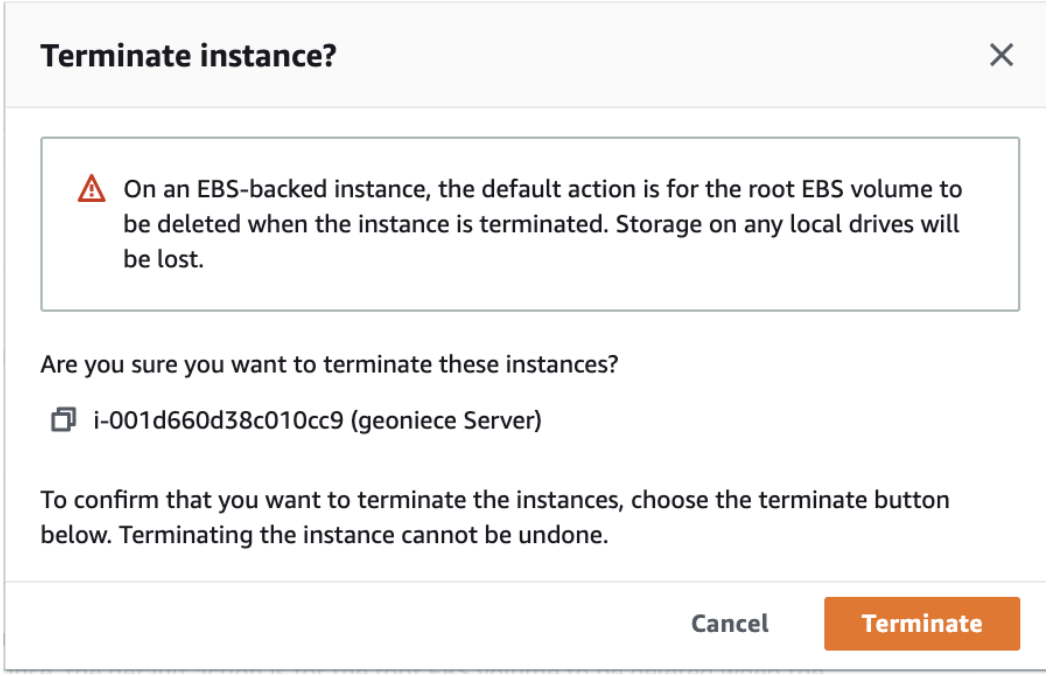
   Navigate to the EC2 console, choose Instances, select the checkbox for our instance, from the Actions menu choose Manage State



   Choose the Terminate radio and choose Change state

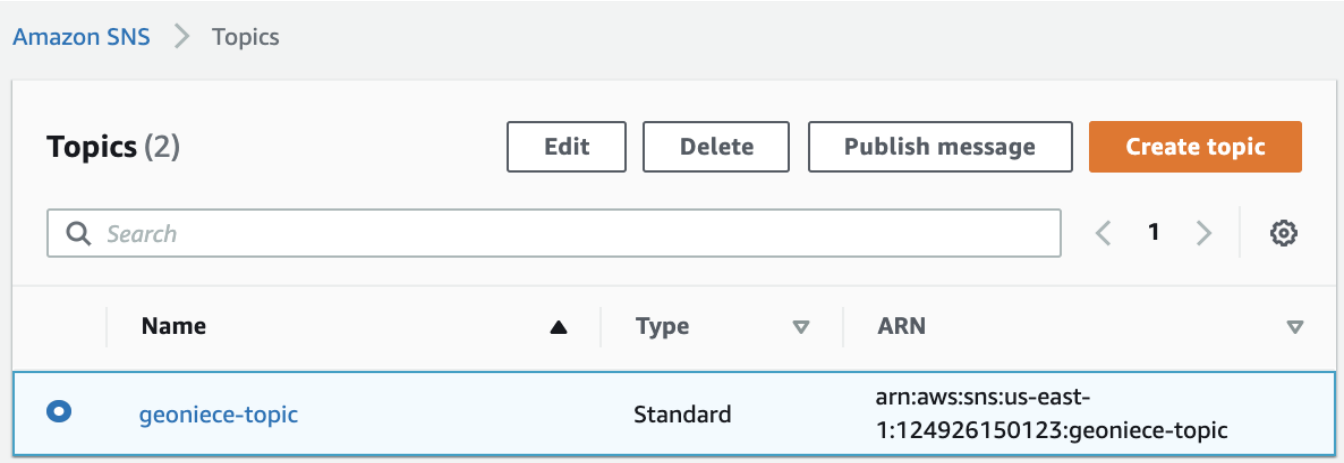


Choose Terminate

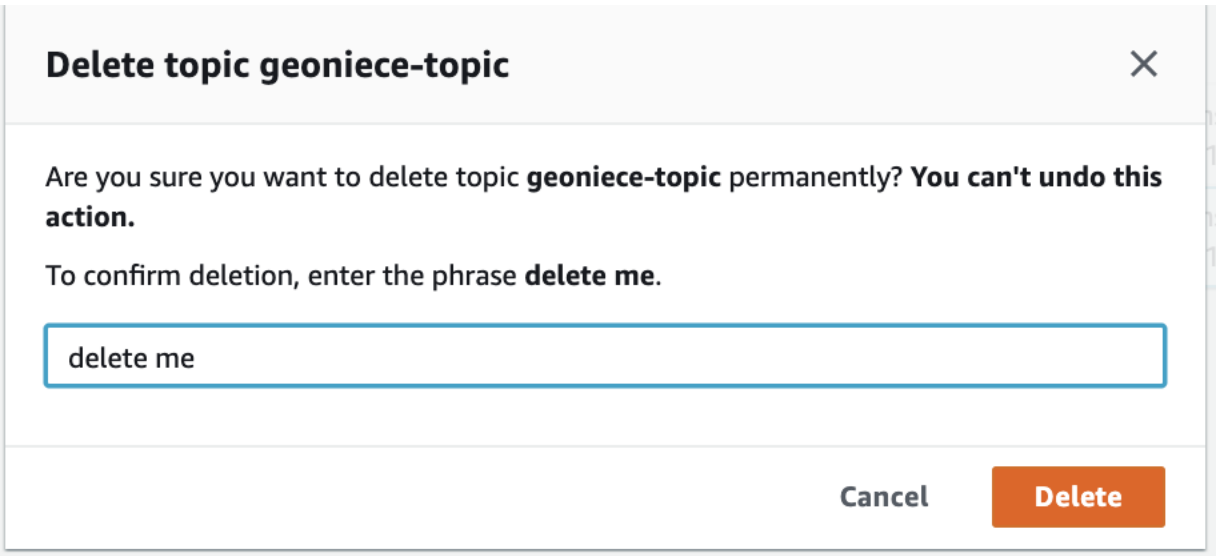


3. Delete your SNS topic.

   Navigate to the Simple Notification Service (SNS) console, choose Topics, select the checkbox for our topic, choose Delete



   Enter “delete me” in the input and Choose Delete



Congratulations