# Monitoring AWS Resources using CloudWatch Dashboards and Widgets

Created by[Stephane Maarek | AWS Certified Cloud Practitioner,Solutions Architect,Developer](https://deloittedevelopment.udemy.com/user/stephane-maarek/)

**AWS Workspace**

**90-120 minutes**

In this lab, you will be a SysOps Administrator at a company that wants to build a resource monitoring and reporting solution to understand the root cause of performance bottlenecks. The company wants to be able to assess the health of its AWS resources and applications across multiple AWS Regions. The company also wants to incorporate dashboards and widgets into the solution so that it can provide guidance for team members during operational events about how to respond to specific incidents.

Our SysOps Administration team has been mandated to build a resource monitoring and reporting solution for our company. We need you to implement a monitoring solution that helps the company to monitor key resource metrics and configure automated notifications in case any performance bottlenecks are detected.

To build a monitoring and reporting layer for the entire end-to-end solution, you will leverage CloudWatch dashboards and widgets to get aggregated views of the health and performance statistics for the EC2 instance. We want you to report any performance bottlenecks via SNS notifications as soon as these are detected.

TASK

1-Provision EC2 in custom vpc and configure SG

2- Create cloudwatch dashboard and add widgets to monitor EC2

3-Create cloudwatch dashboard alarm to send sns when instance metric breache the threshold

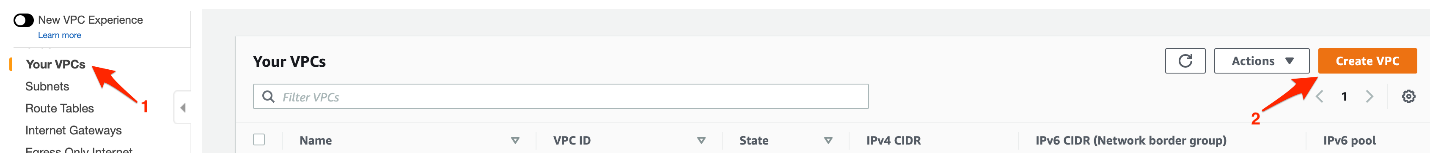
4-generate synthetic CPU load on EC2 by using stress utility

5-Clean Up

Provision EC2 in custom vpc and configure SG

To implement this step, you need to set up a new VPC and then create a public subnet in this VPC. You will provision an Internet Gateway and make route entries for the Internet Gateway in the main Route Table for the given VPC. Provision an EC2 instance in a custom VPC and create a new security group to allow SSH access on port 22.

1. Make sure that you are in the N.Virginia AWS Region on the AWS Management Console. Enter **VPC** in the search bar and select **VPC**service.
2. Let's create a new VPC that will be used to provision the EC2 instance
   1. Click on the **Your VPCs** link in the left sidebar and then click **Create VPC** button



* 1. Enter the following values to create the VPC:

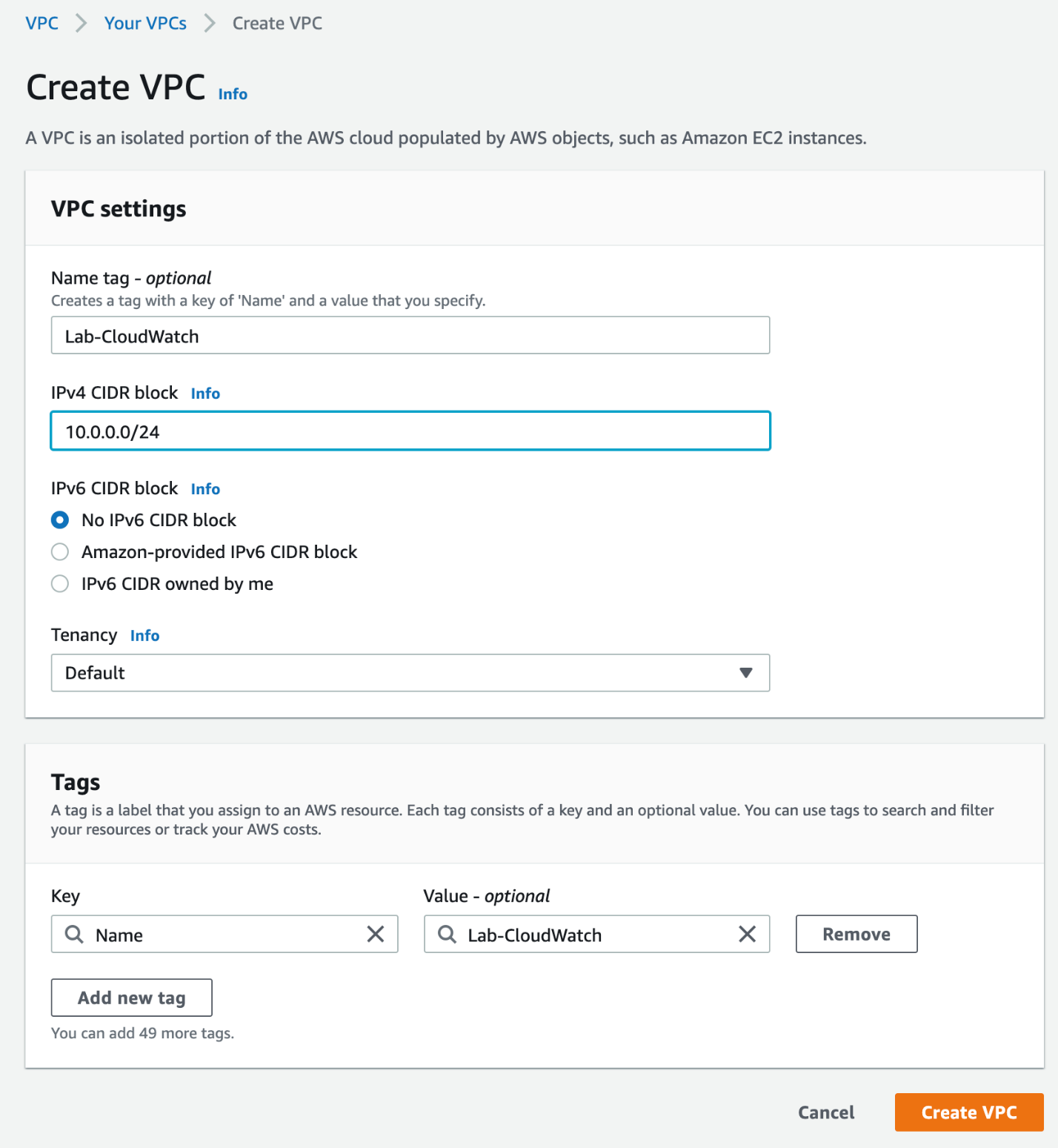
**Name-tag:** Lab-CloudWatch

**IPv4 CIDR Block:** 10.0.0.0/24

For **IPv6 CIDR Block**, select **No IPv6 CIDR Block**

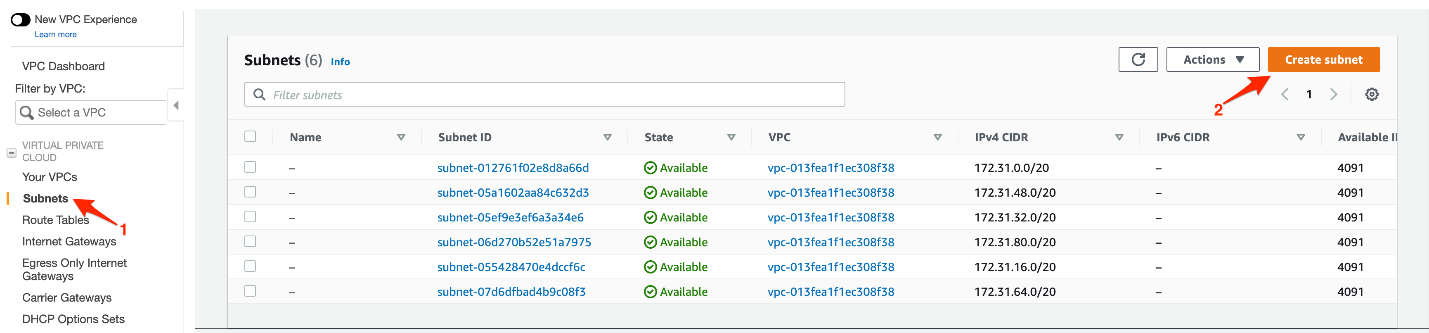
For **Tenancy**, select **Default** as the value

Click **Create VPC** to complete the configuration.



* 1. You should see that your VPC has been created successfully.

1. Now, we shall create a new subnet in this VPC.
   1. Click on the **Subnets** link in the left sidebar and then click **Create subnet** button.



* 1. Enter the following values to create the subnet:

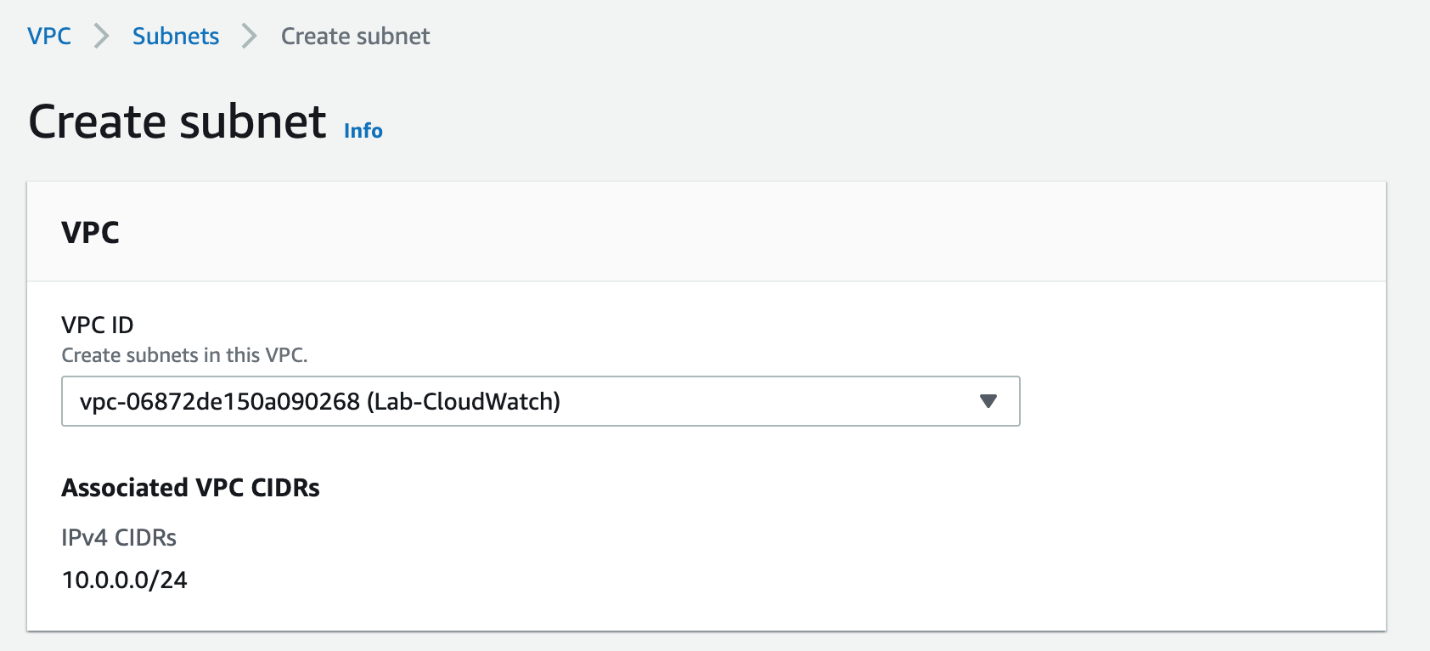
For **VPC ID**, select the VPC - **Lab-CloudWatch**

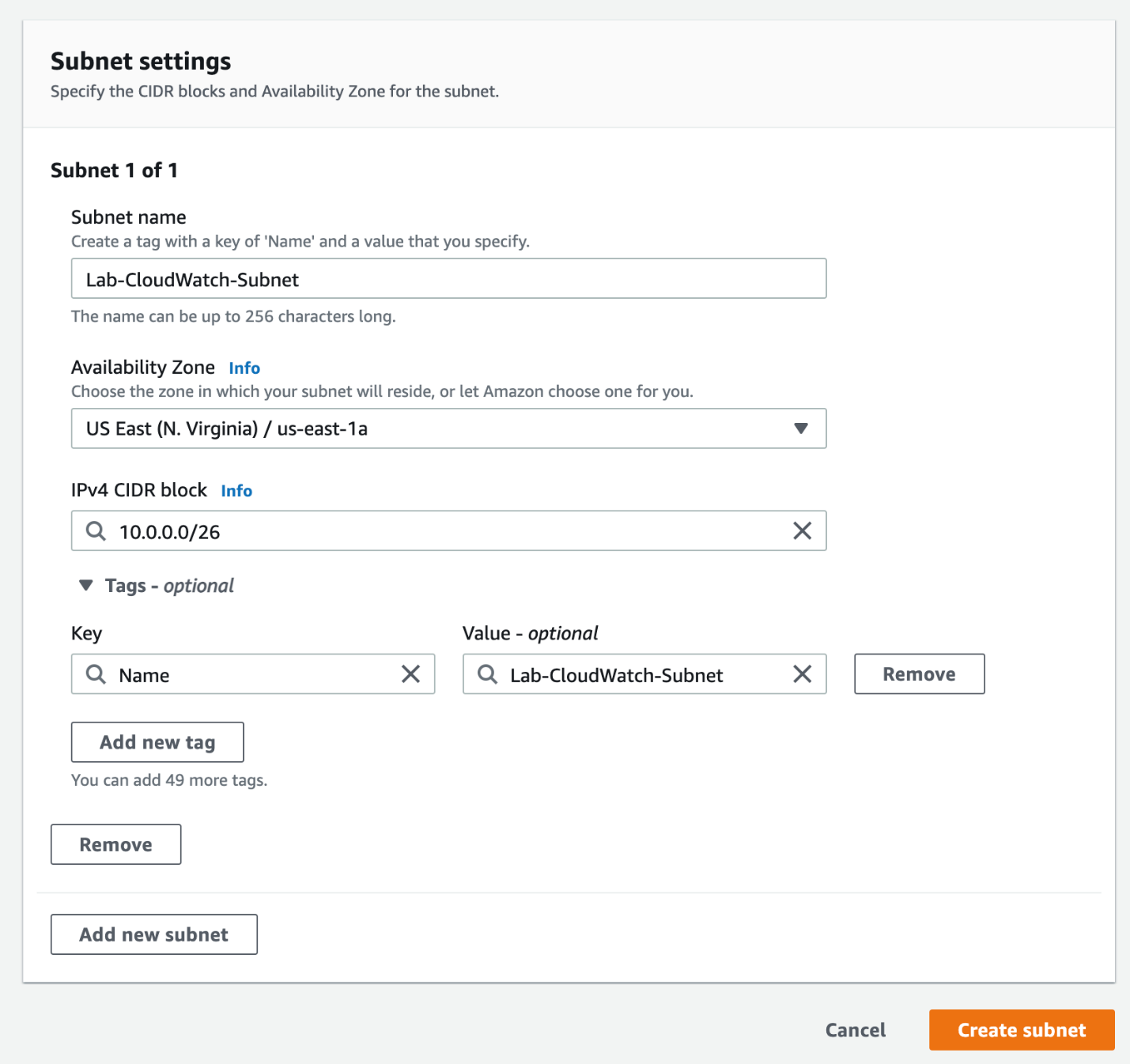
**Subnet name:**Lab-CloudWatch-Subnet

For **Availability Zone**, select **us-east-1a**

**IPv4 CIDR block:**10.0.0.0/26

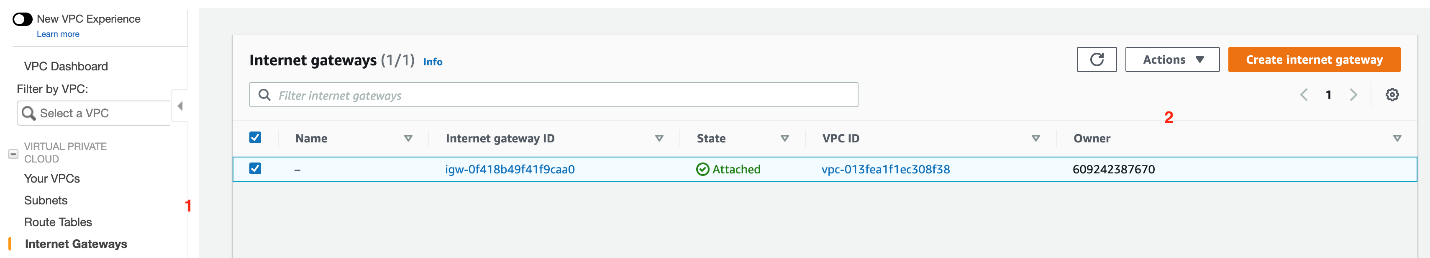
Click **Create subnet** to complete the configuration.





* 1. You should see that your subnet has been created successfully.

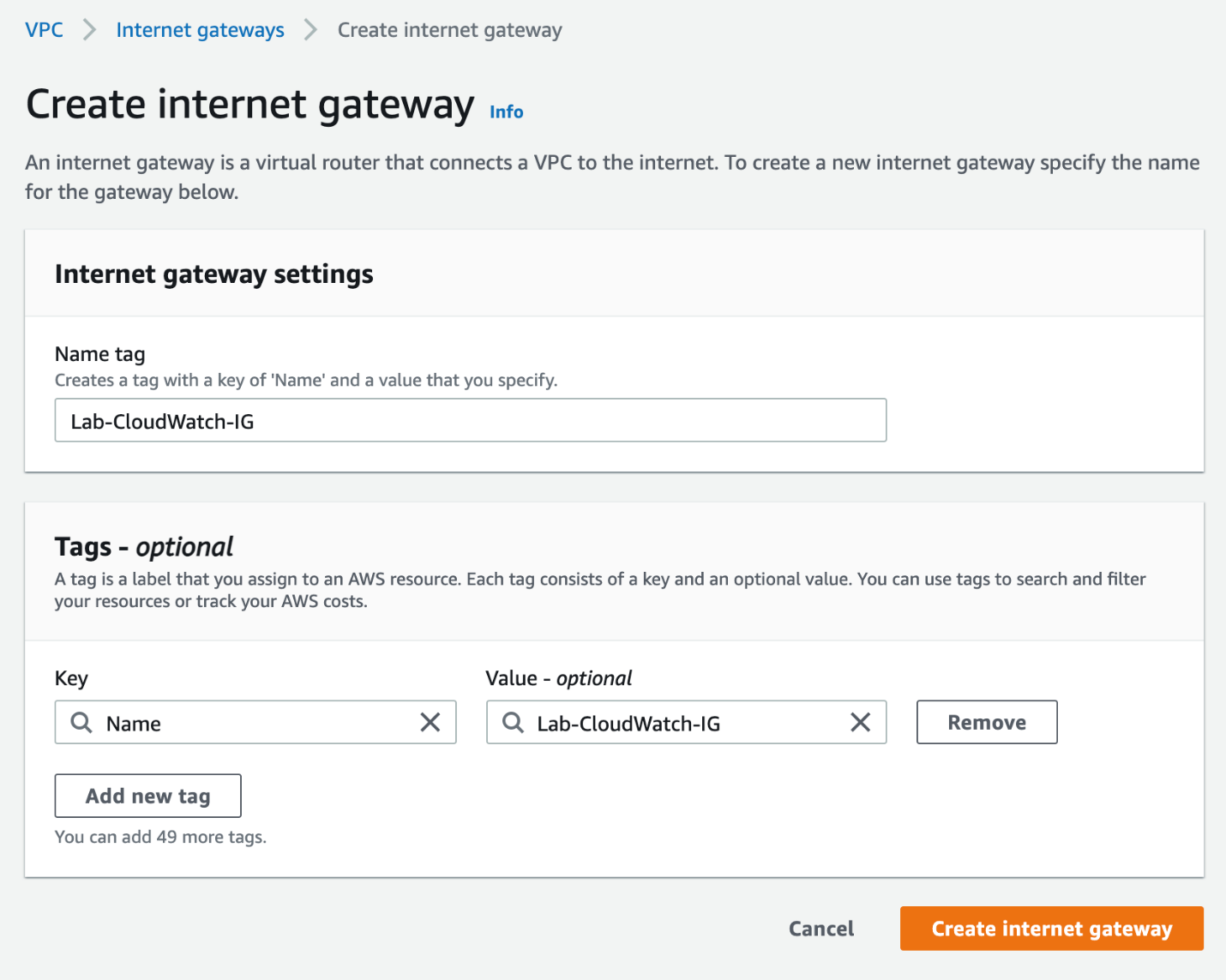
1. Let's create an Internet Gateway that will be attached to the VPC.
   1. Click on the **Internet Gateways** link in the left sidebar and then click **Create internet gateway** button.



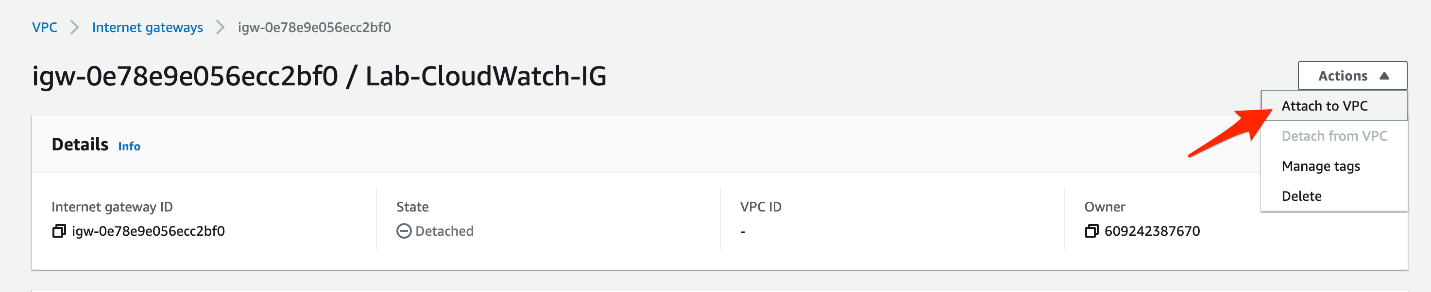
* 1. Enter the following values to create the internet gateway:

**Name tag**: Lab-VPC-Flow-Log-IG

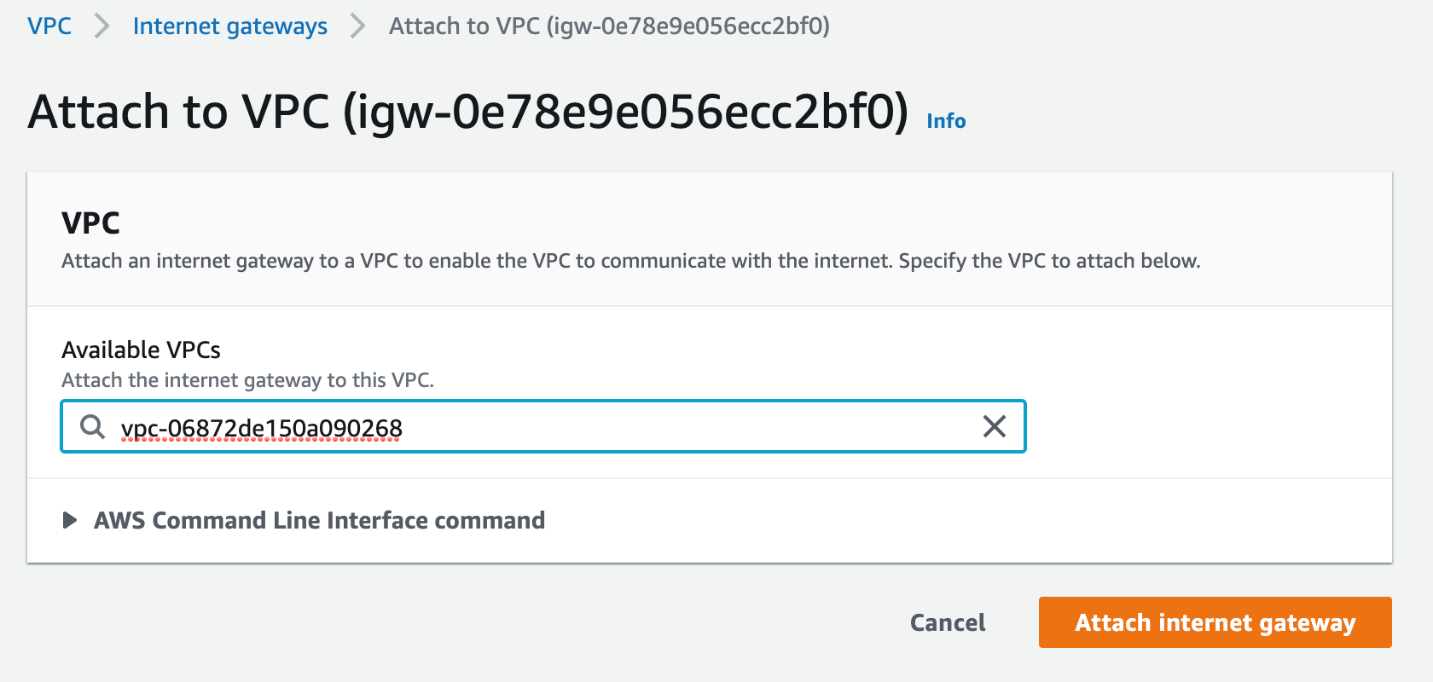
Click **Create internet gateway** to complete the configuration.



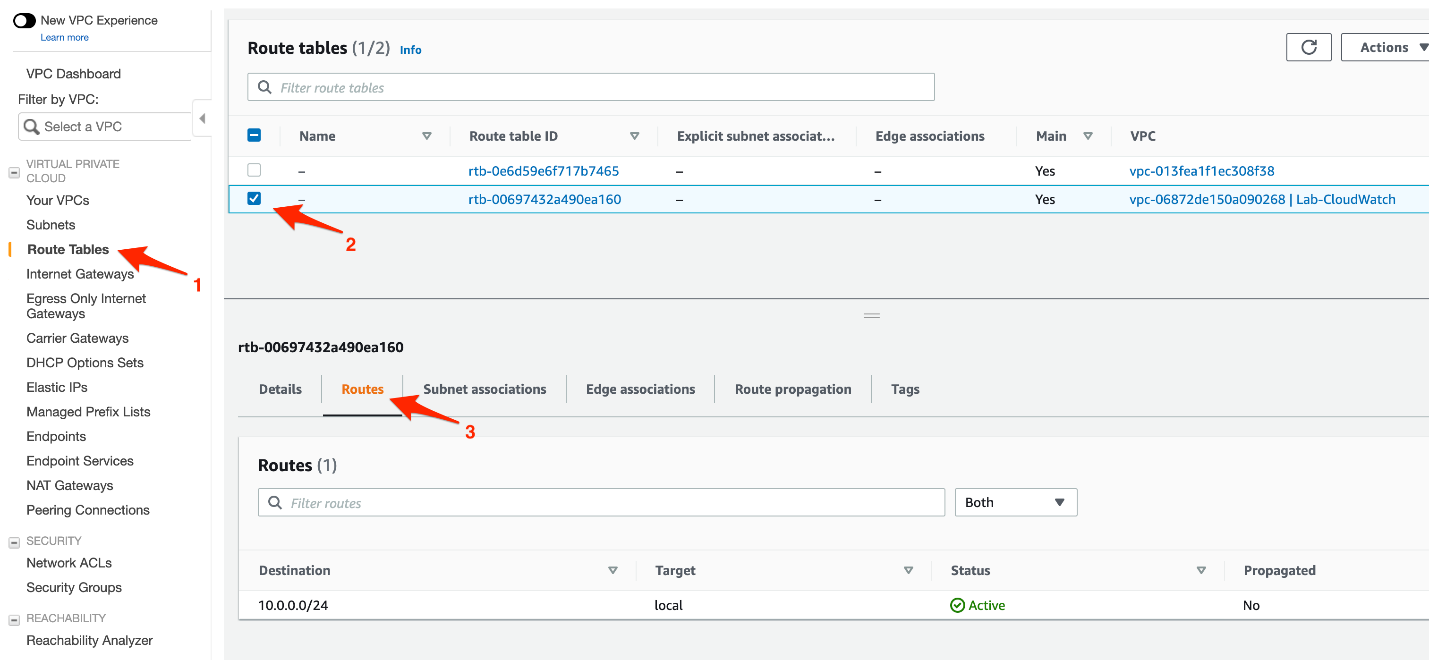
* 1. You should see that your internet gateway has been created successfully. Click on **Attach to VPC** from the **Actions** menu.



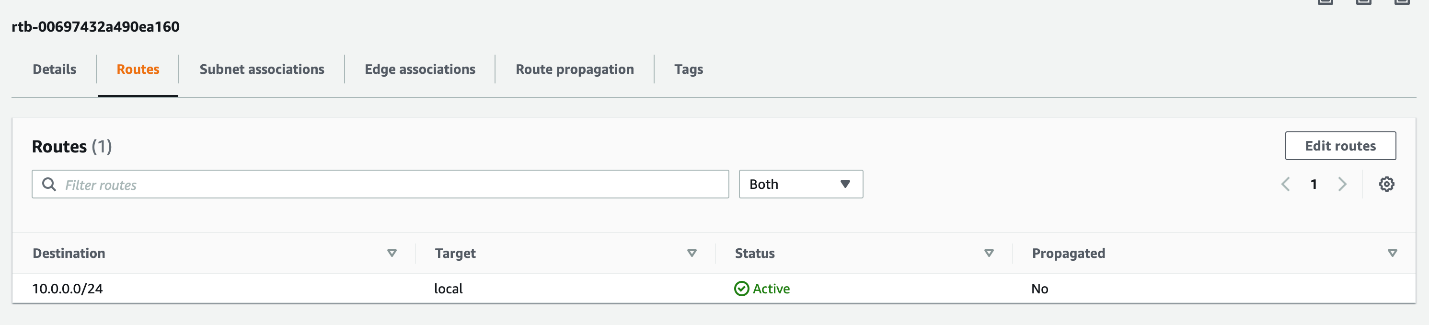
* 1. Select the VPC - **Lab-CloudWatch**and click on the **Attach internet gateway** button.



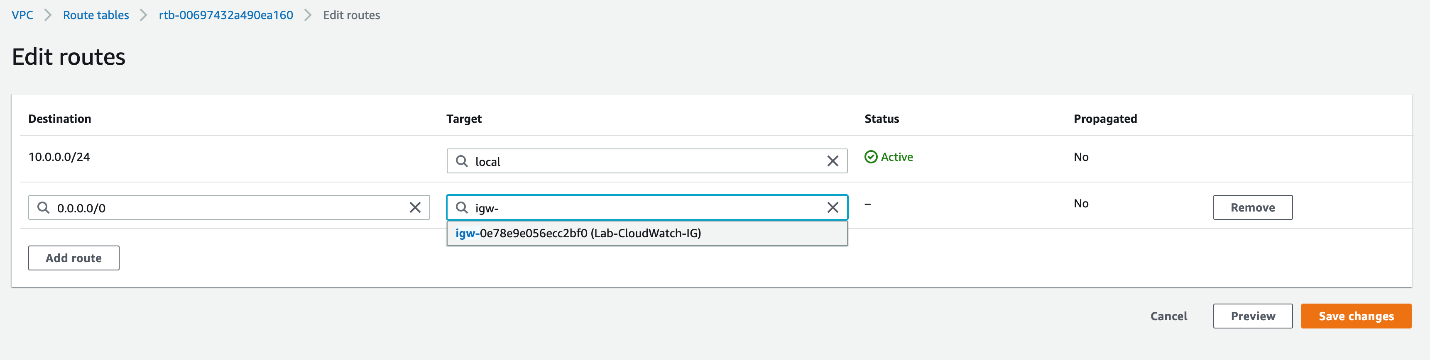
1. Now, we shall create a route for this internet gateway in the main route table of the VPC.
   1. Click on the **Route Tables** link in the left sidebar and then select the main route table for your VPC - **Lab-VPC-Flow-Logs**. Select the **Routes** tab.



* 1. Click on the **Edit routes** button.

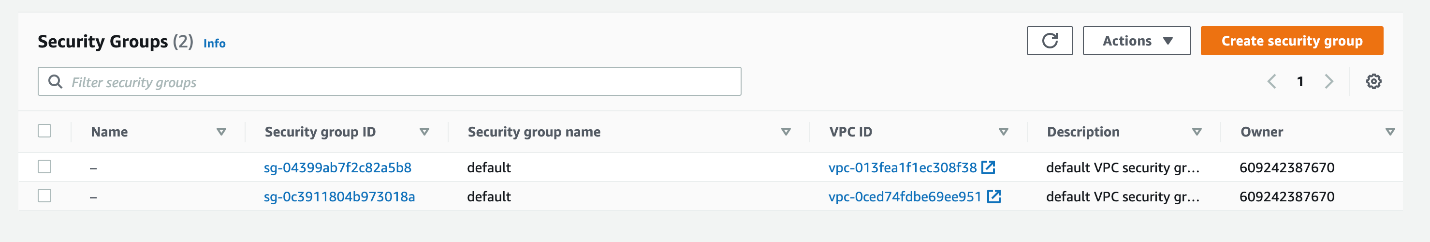


* 1. Click on the **Add route** button and enter **destination** as 0.0.0.0/0, select the internet gateway - **Lab-CloudWatch-IG**- as the **target**. Click on the **Save routes** button.



You should see that the route to the internet gateway has been added successfully.

1. Let's create a security group that will be used while provisioning the EC2 instance.
   1. Make sure that you are in the N.Virginia AWS Region on the AWS Management Console. Enter **EC2** in the search bar and select **EC2**service.
   2. Let's create a security group to allow SSH into the EC2 instance. Select **Security Groups**from the left sidebar and click on **Create security group** button

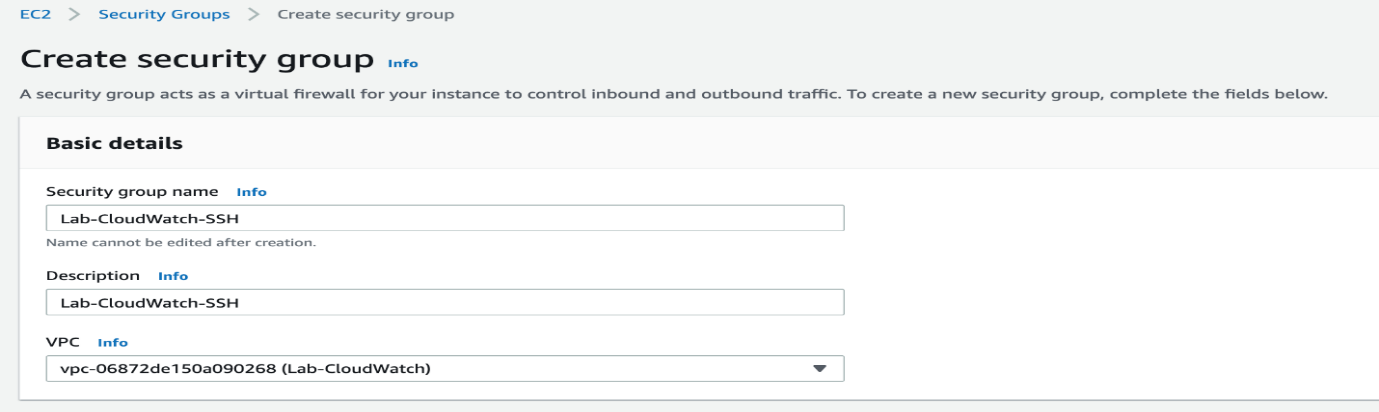


* 1. Enter the following values to create the security group:

**Security group name**: Lab-CloudWatch-SSH

**Description**: Lab-CloudWatch-SSH

For **VPC**, select **Lab-CloudWatch**



* 1. Click on **Add rule** button in the **Inbound rules** section and configure values for the inbound rule for SSH access as shown below:

Type: SSH

Protocol: TCP

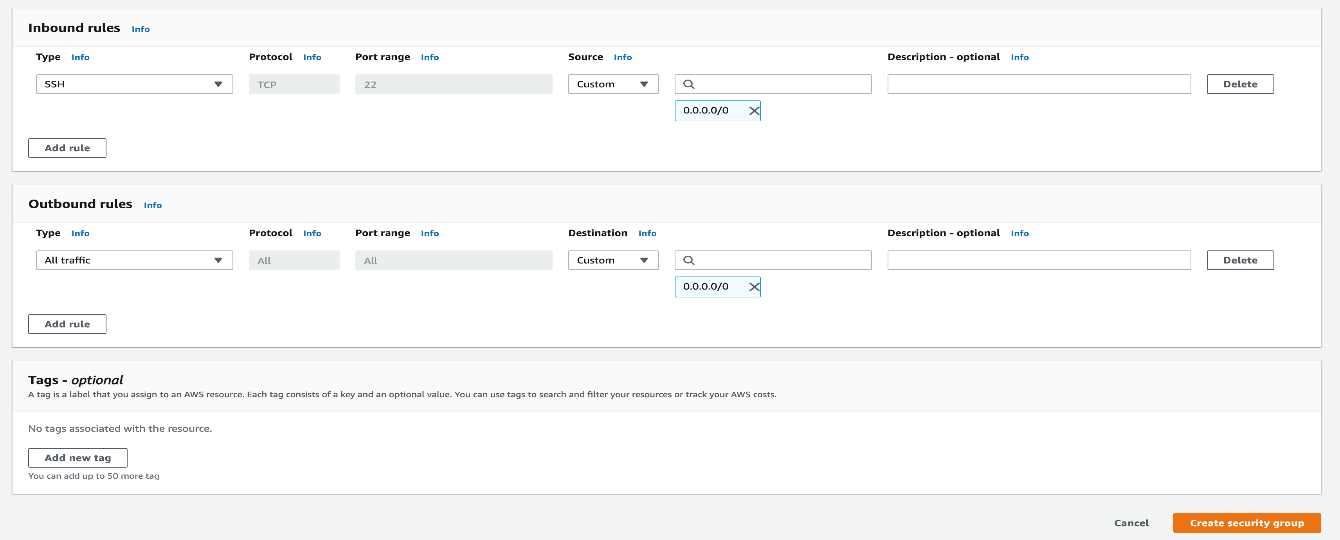
Port Range: 22

Source: Custom

Put the source IP value as 0.0.0.0/0

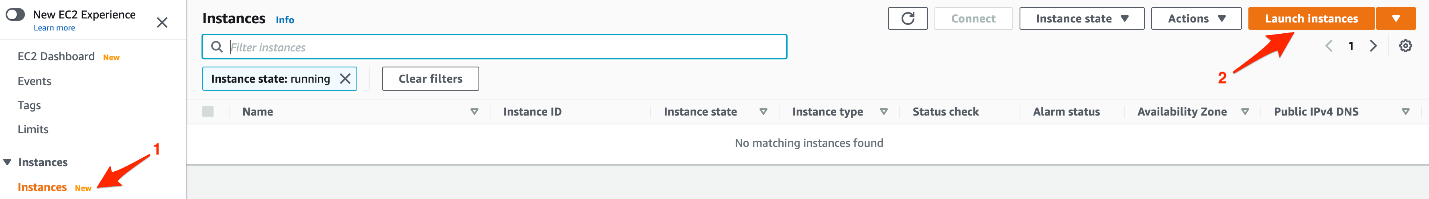
*Note - For production systems, you should not have such open SSH access from any source. You should allow SSH only from your own public IP.*

Click on the **Create security group** button.



You should see that your security group has been created successfully.

1. Now, we shall provision the EC2 instance
   1. Click on the **Instances** link on the left sidebar and click on the **Launch Instances** button.



* 1. Choose **Amazon Linux 2 AMI (HVM), SSD Volume Type** and click on Select.
  2. Choose **t2.micro** as the instance type and click on Next.
  3. Configure instance details as follows:

Select **Lab-CloudWatch** as the **Network**

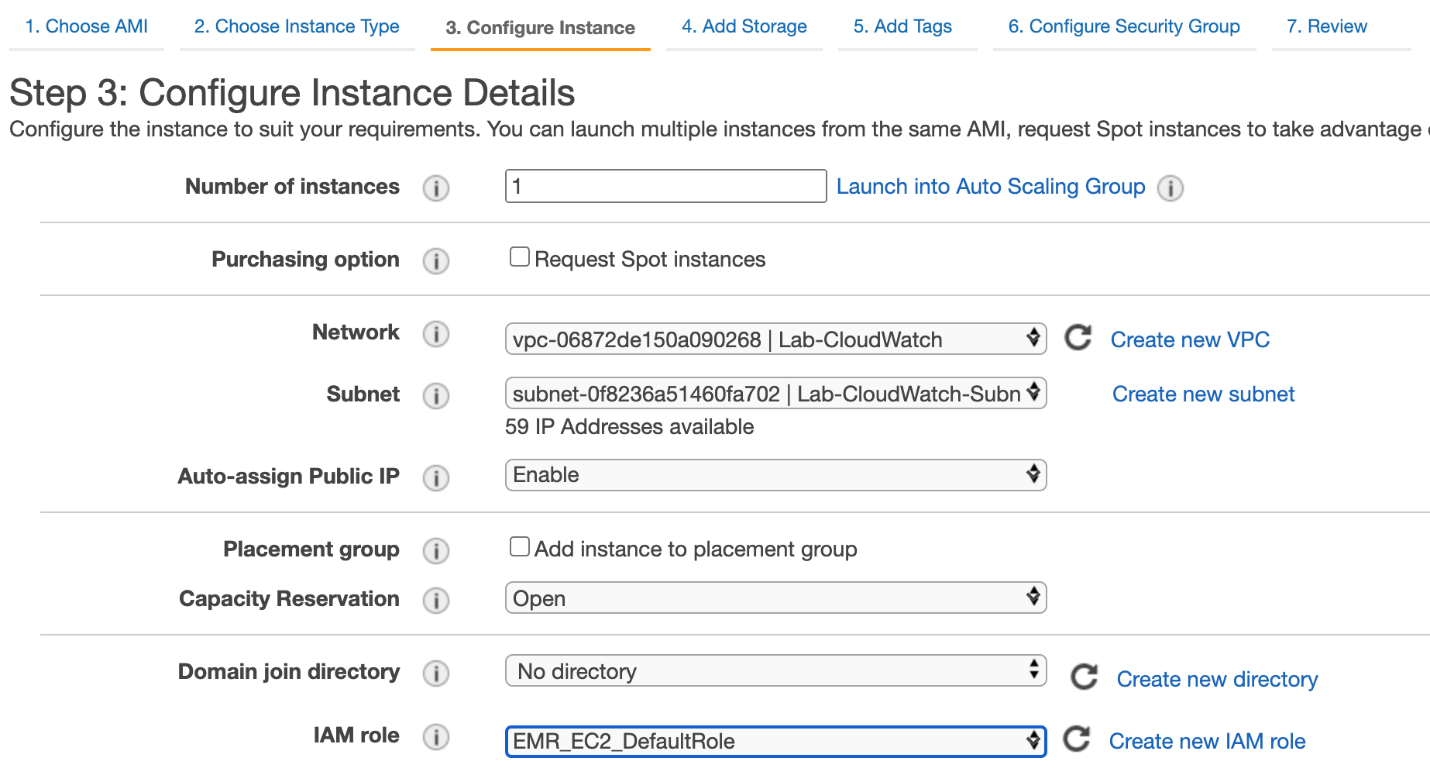
Select**Lab-CloudWatch-Subnet**as the**Subnet**

For**Auto-assign Public IP,**select**Enable**

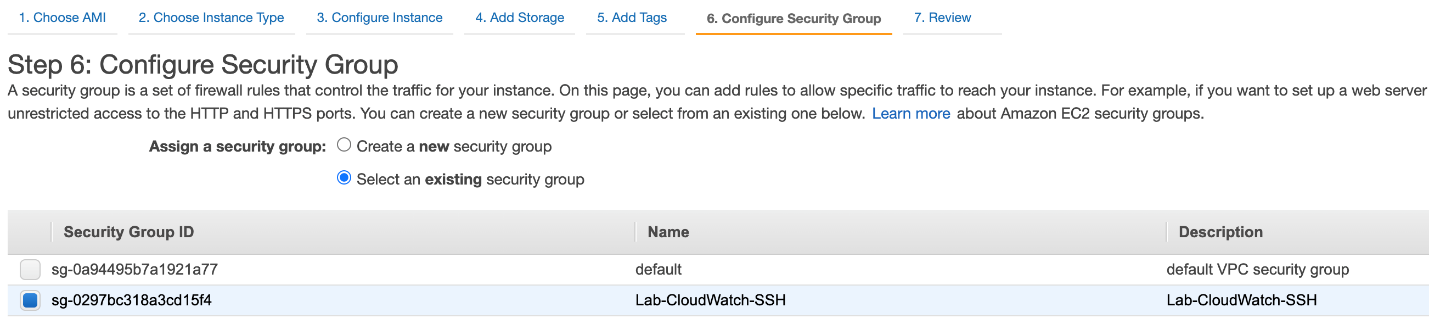
Select**IAM role**as**EMR\_EC2\_default**

Leave all the other values as unchanged

Click on Next



* 1. Leave the default values for **Add Storage** and click on Next.
  2. Leave the default values for **Add Tags** and click on Next.
  3. Configure the security group. Click on **Select an existing security group** and then select **Lab-CloudWatch-SSH**security group



* 1. Review the instance launch details and then click on **Launch**button.
  2. For the key pair, select **Proceed without a key pair** as we shall use EC2 Instance Connect to SSH into the instance. Select the checkbox for acknowledgement. Click on Launch Instance.

<https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html>

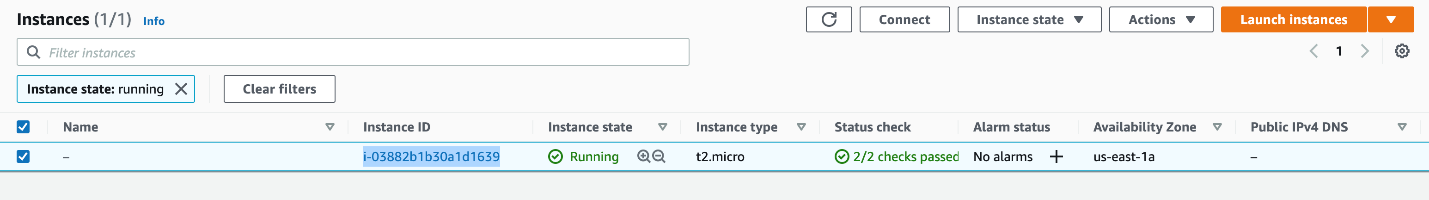
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

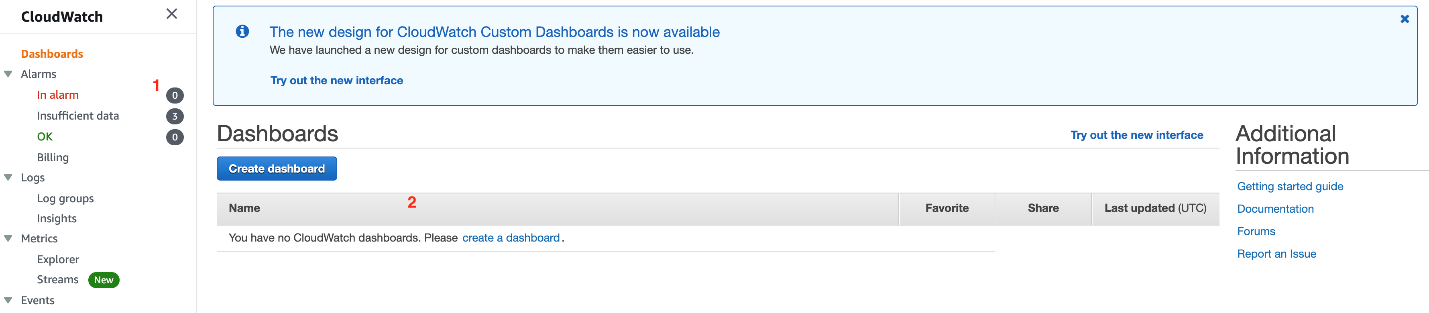
Create cloudwatch dashboard and add widgets to monitor EC2

Create a CloudWatch dashboard and add widgets to track CPU utilization, status check and network packets in/out to get aggregated views of the health and performance information for that instance.

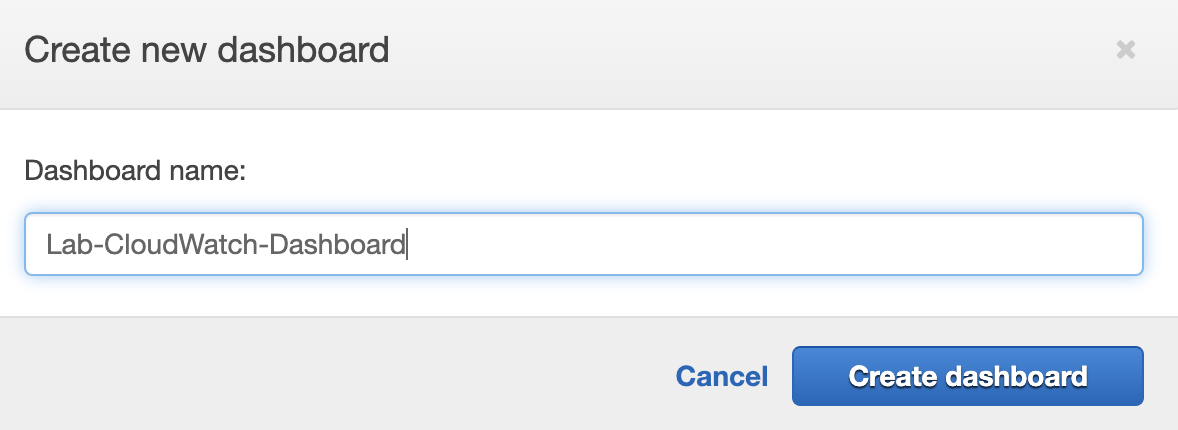
1. Make sure that you are in the N.Virginia AWS Region on the AWS Management Console.
   1. Navigate to the EC2 service and copy the **Instance ID** of the EC2 instance you have already created. You shall use this instance ID while creating the CloudWatch widgets.



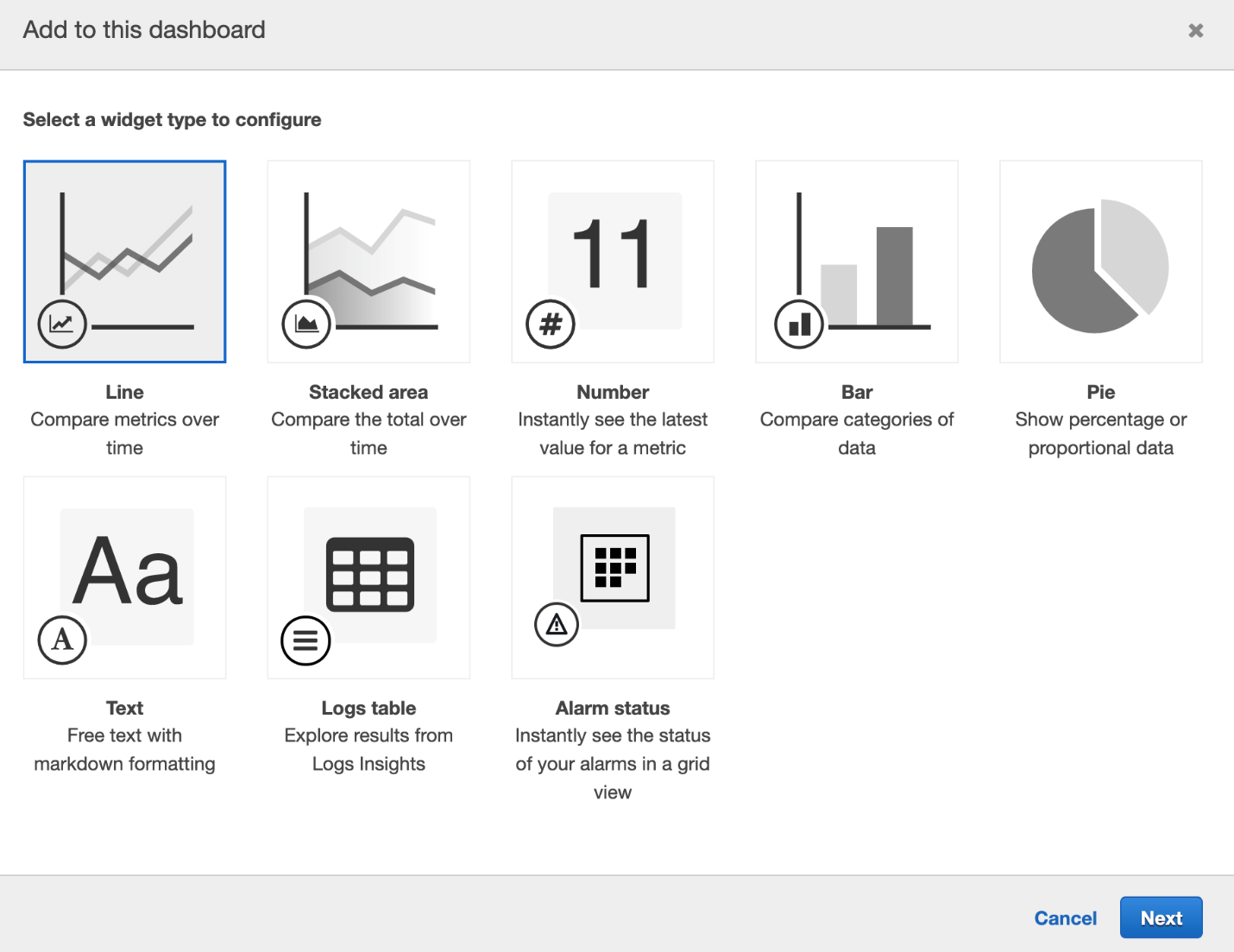
* 1. Navigate to the **CloudWatch**service. Click on the **Dashboards** link in the left sidebar and then click on the **Create dashboard** button.



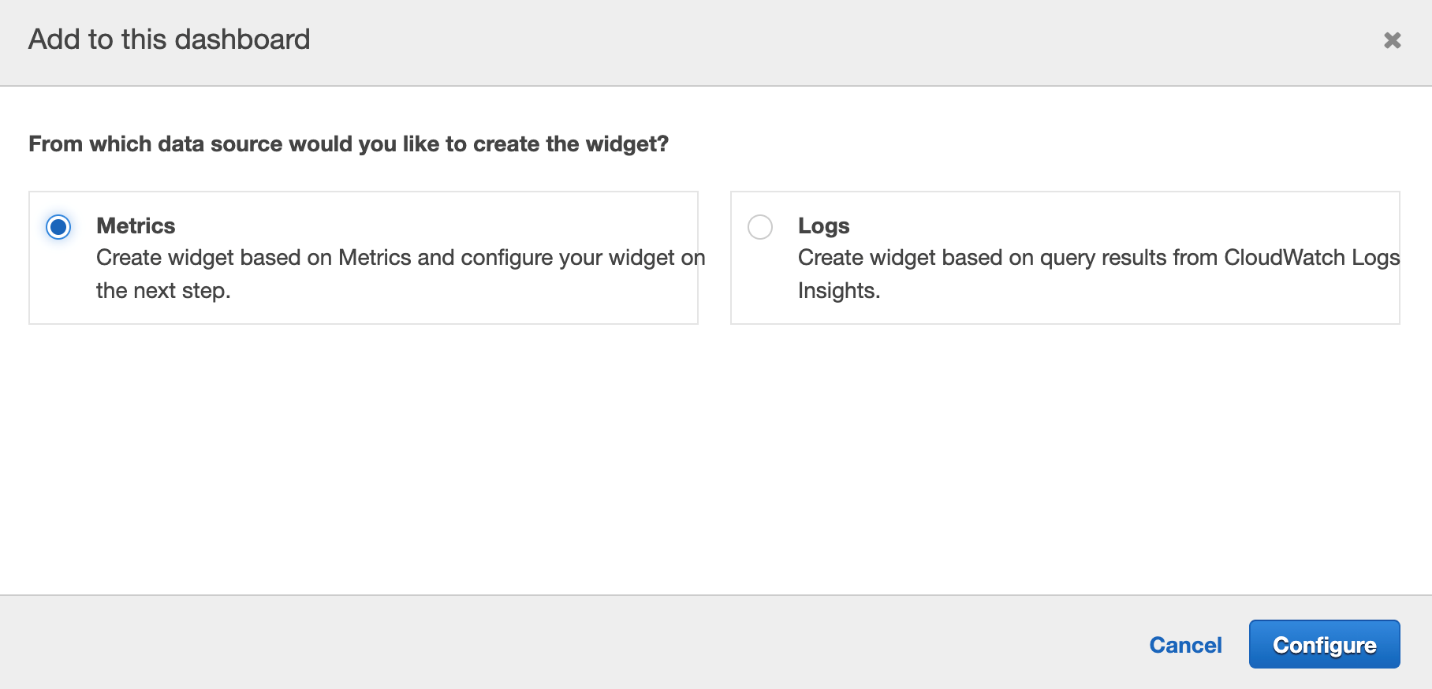
* 1. Enter the Dashboard name as **Lab-CloudWatch-Dashboard** and click on the **Create dashboard** button.



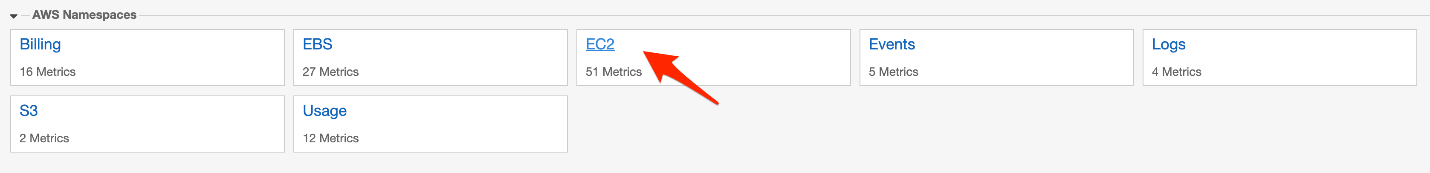
* 1. We shall now create a **line** type CloudWatch widget for monitoring the **CPUUtilization** metric of our EC2 instance. Select **Line** as **Widget Type**and click on Next.



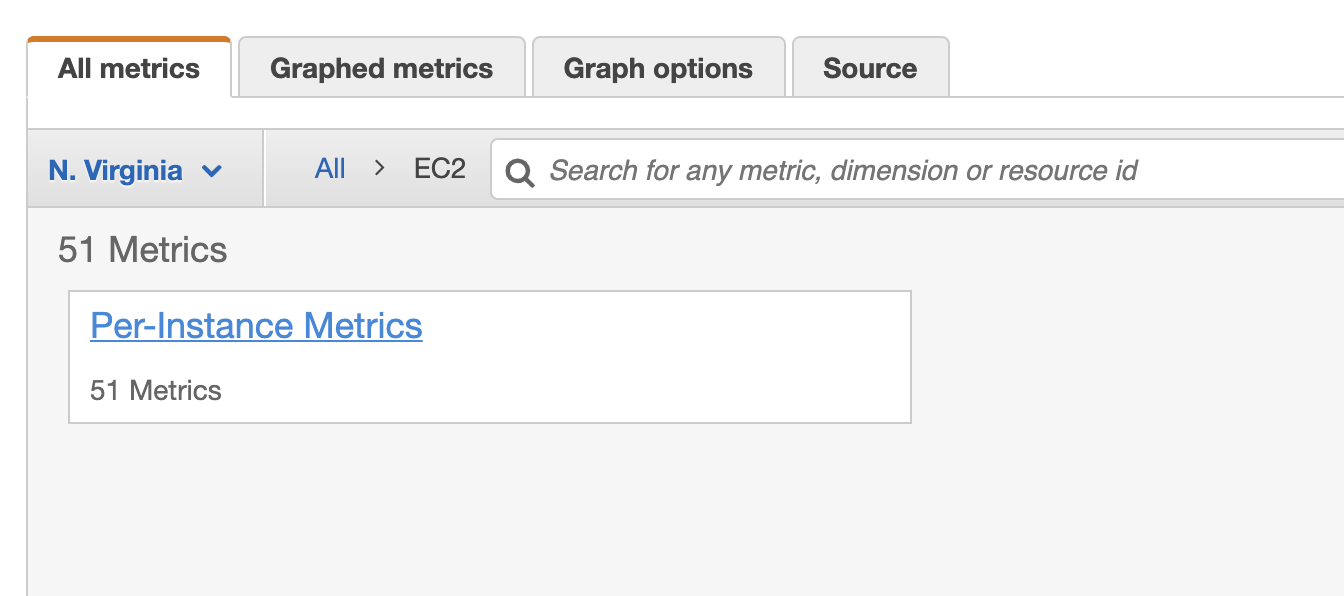
* 1. Select **Metrics** as the data source for the widget and click on Configure.



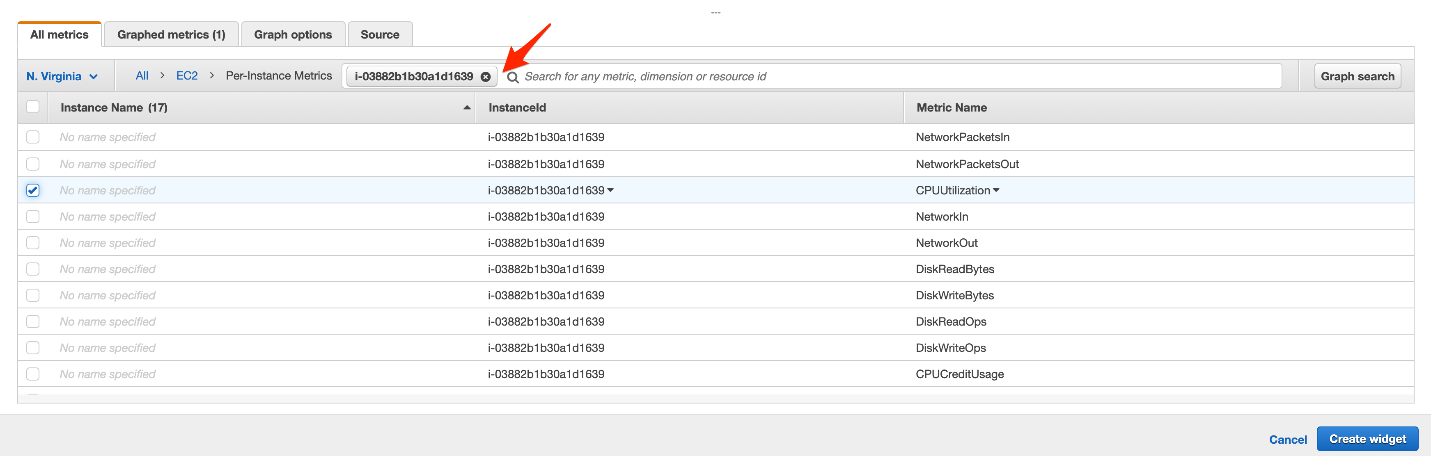
* 1. Select **EC2** from the AWS namespace.



* 1. Select **Per-Instance Metrics**.

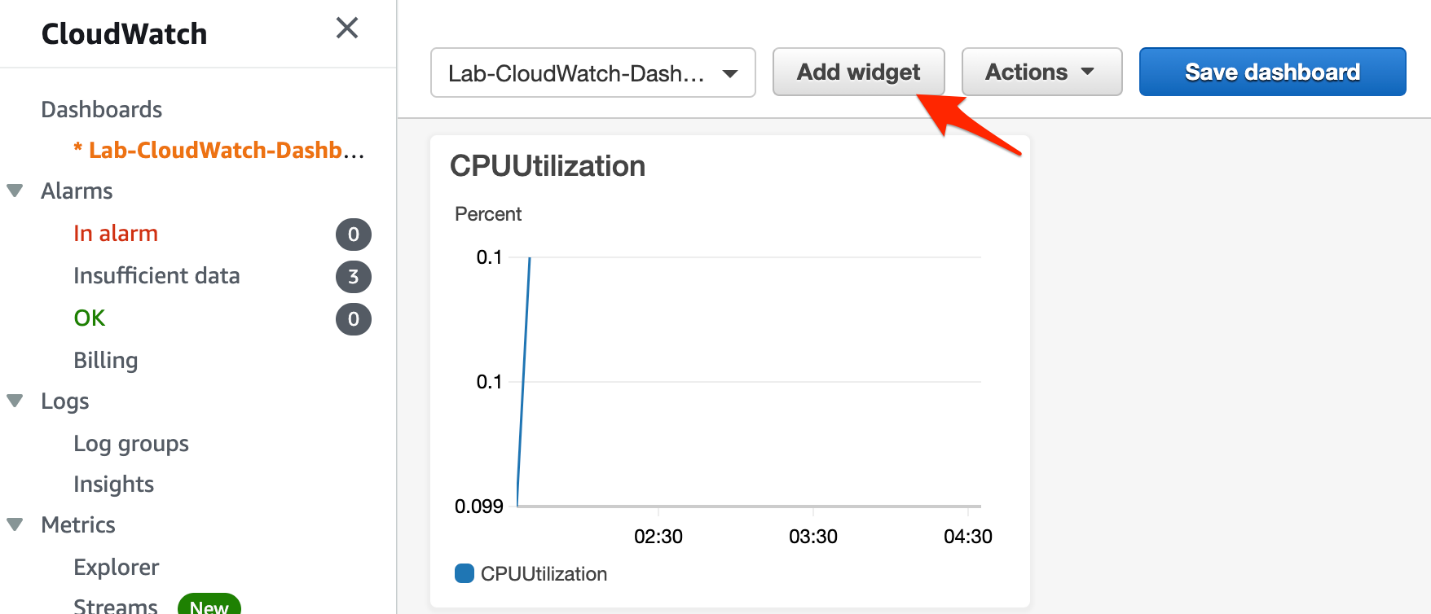


* 1. Paste the instance ID (for your EC2 instance that you had copied earlier) in the search bar and hit enter. Select the metric - **CPUUtilization** and click on **Create widget**.

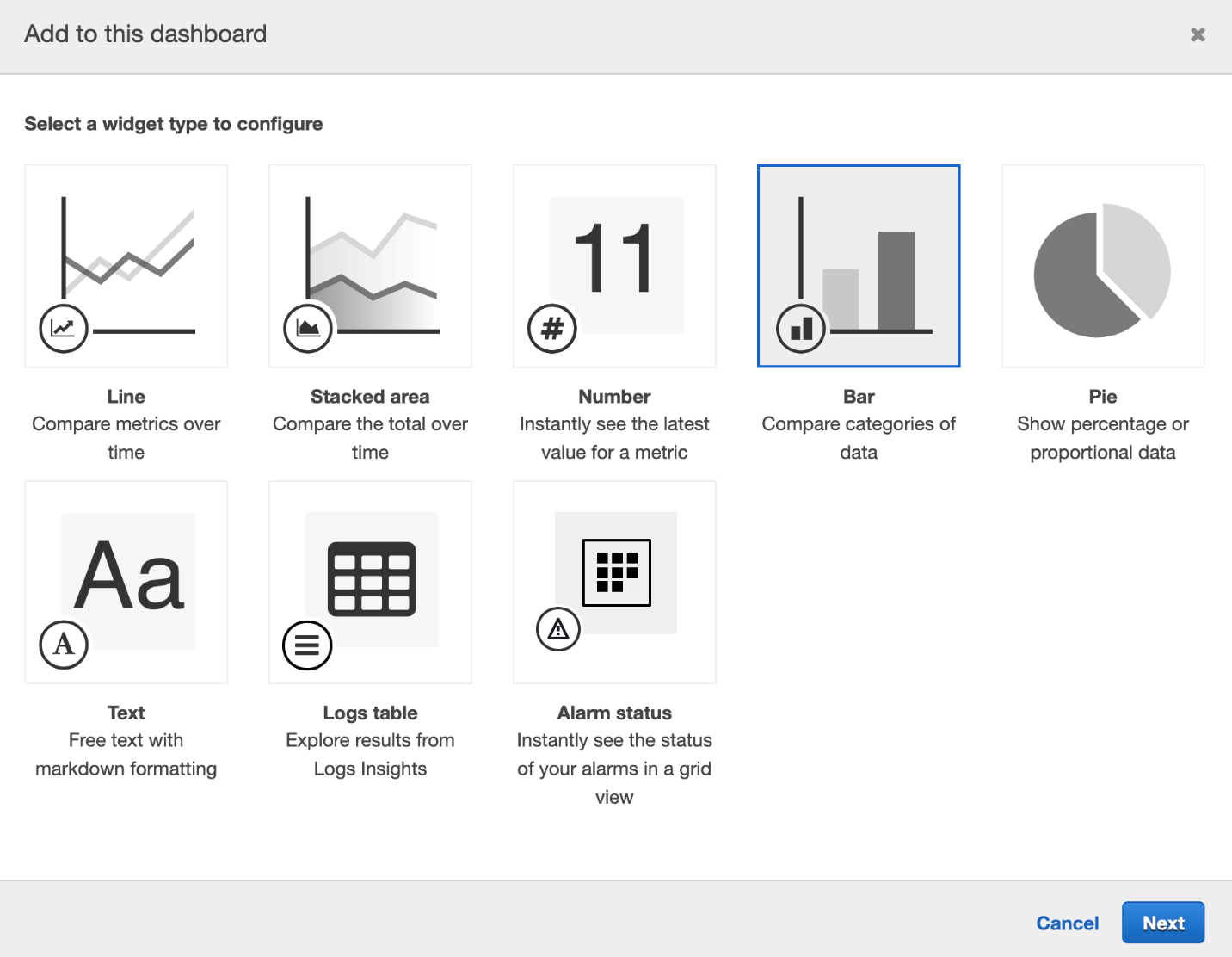


You should see that your CloudWatch widget has been created successfully.

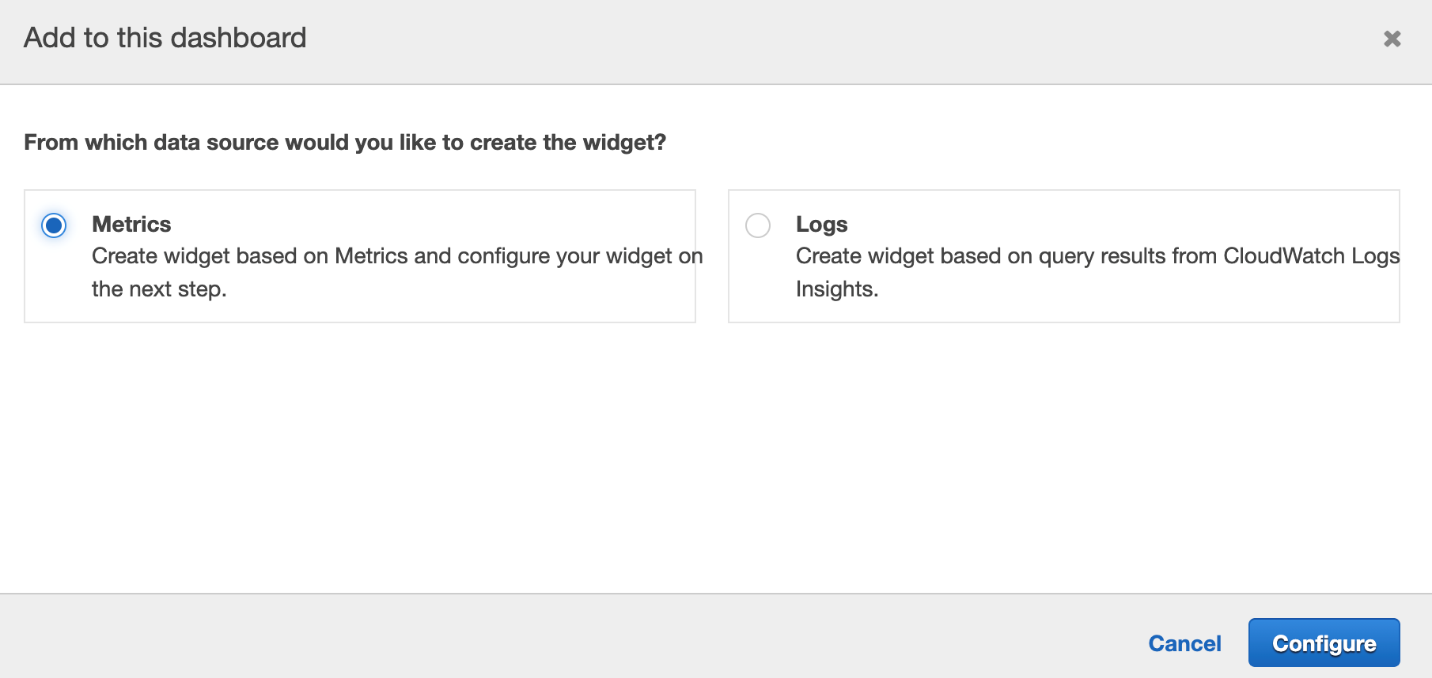
1. Let's create another CloudWatch widget for monitoring the **StatusCheckFailed\_System**and**StatusCheckFailed\_Instance** metrics for our EC2 instance.
   1. Click on the **Add widget** button



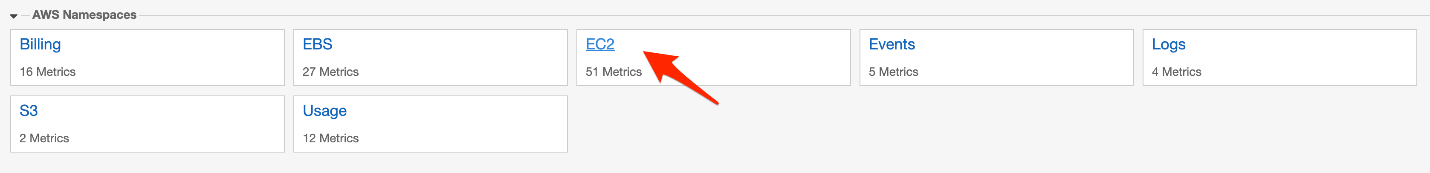
* 1. Select the **Bar** widget as the type for the CloudWatch widget and click on Next.



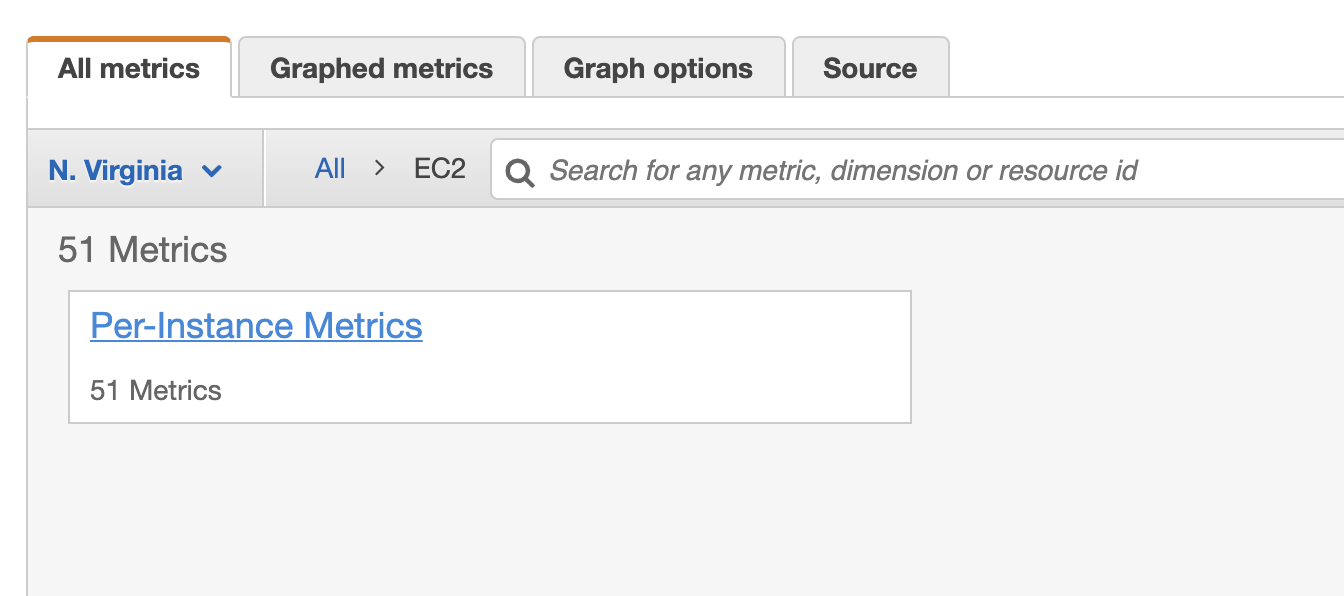
* 1. Select **Metrics** as the data source for the widget and click on Configure.



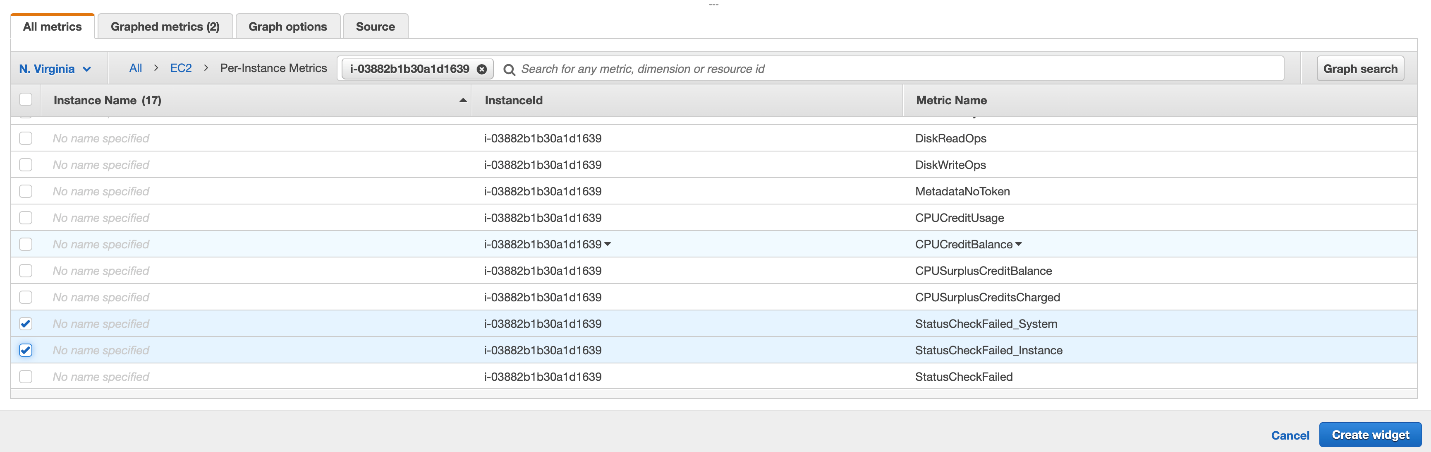
* 1. Select **EC2** from the AWS namespace.



* 1. Select **Per-Instance Metrics**.

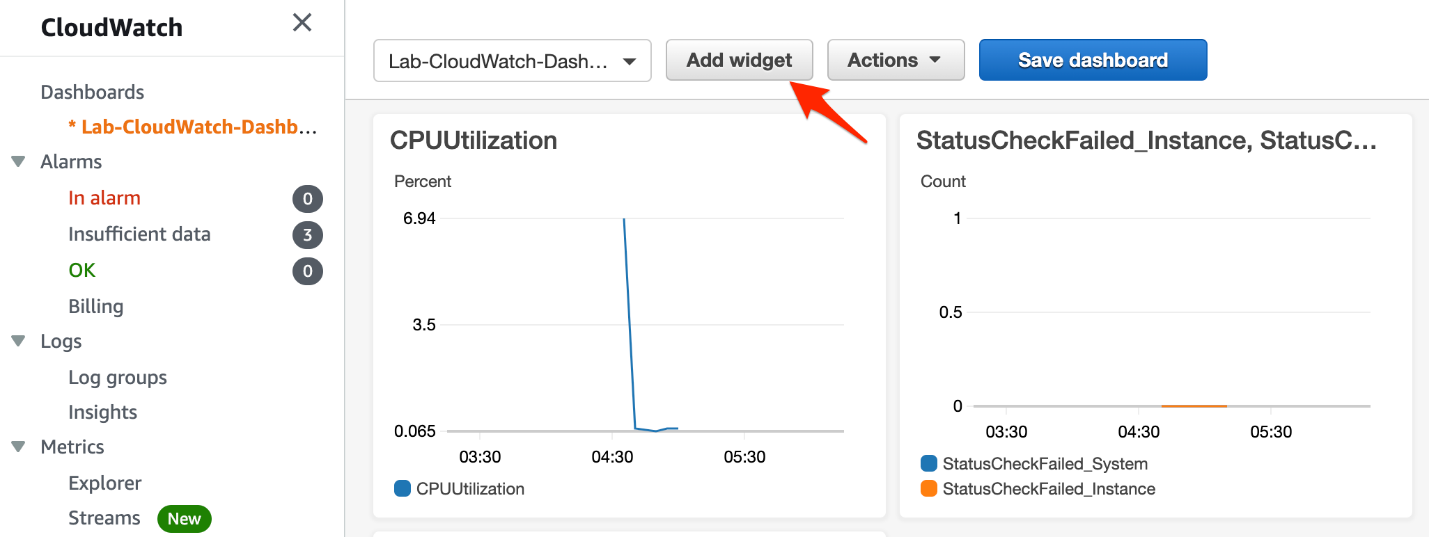


* 1. Paste the instance ID (for your EC2 instance that you had copied earlier) in the search bar and hit enter. Select the metrics - **StatusCheckFailed\_System**and**StatusCheckFailed\_Instance**. Click on **Create widget**.

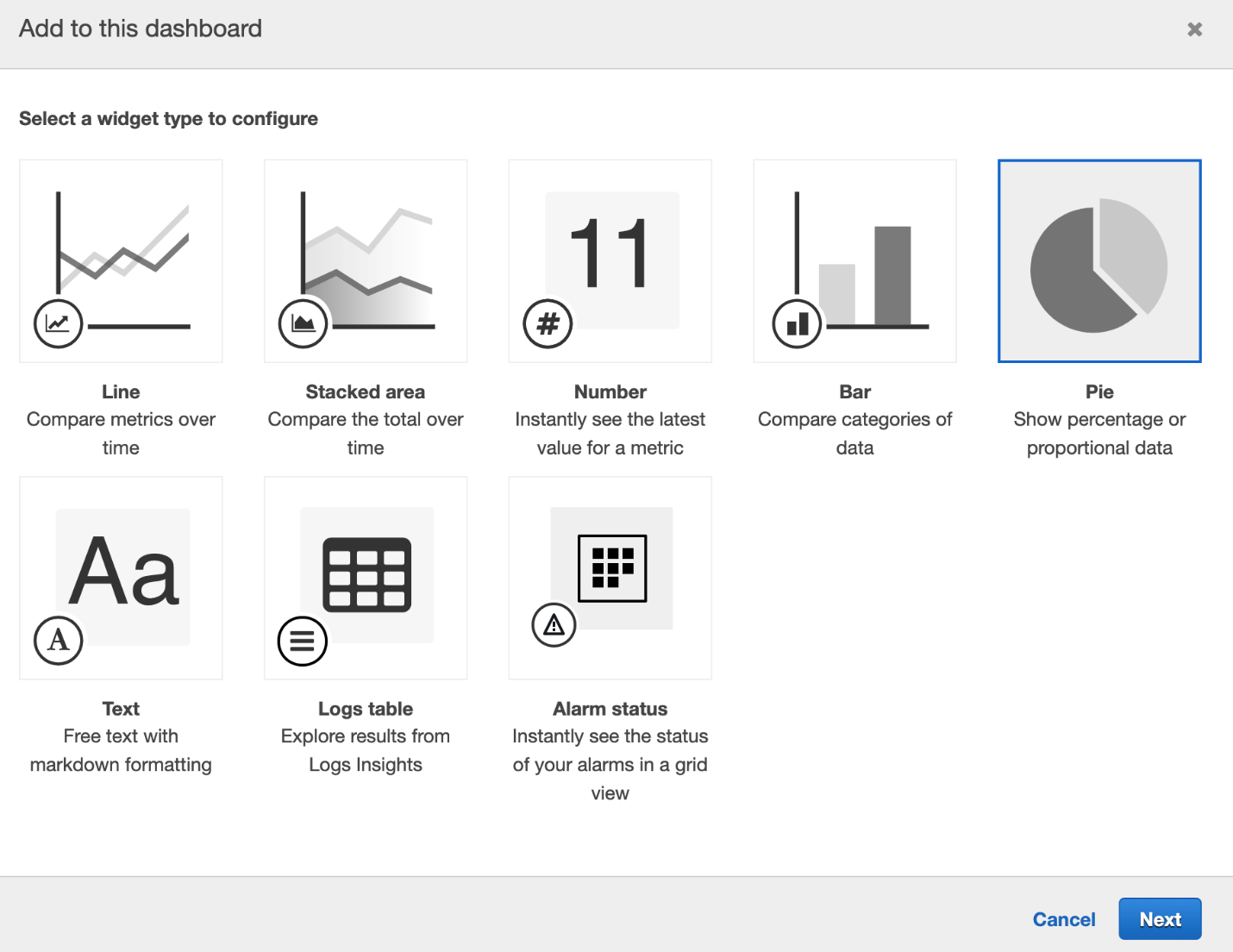


* 1. You should see that the widget has been created successfully.

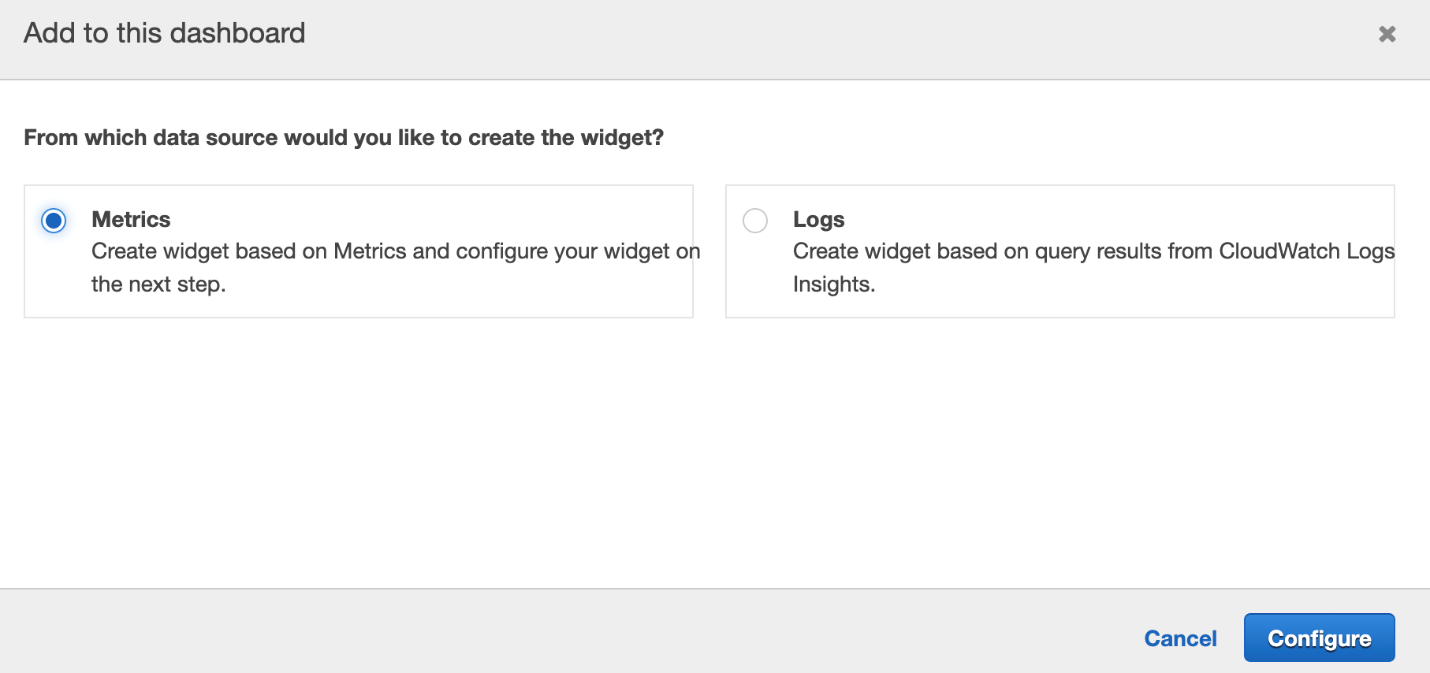
1. Let's add another CloudWatch widget for monitoring the **NetworkPacketsIn**and**NetworkPacketsOut** metrics for our EC2 instance.
   1. Click on the **Add widget** button



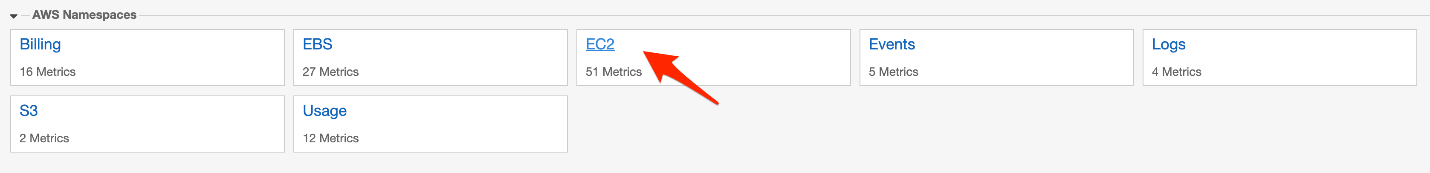
* 1. Select the **Pie** widget as the type for the CloudWatch widget and click on Next.



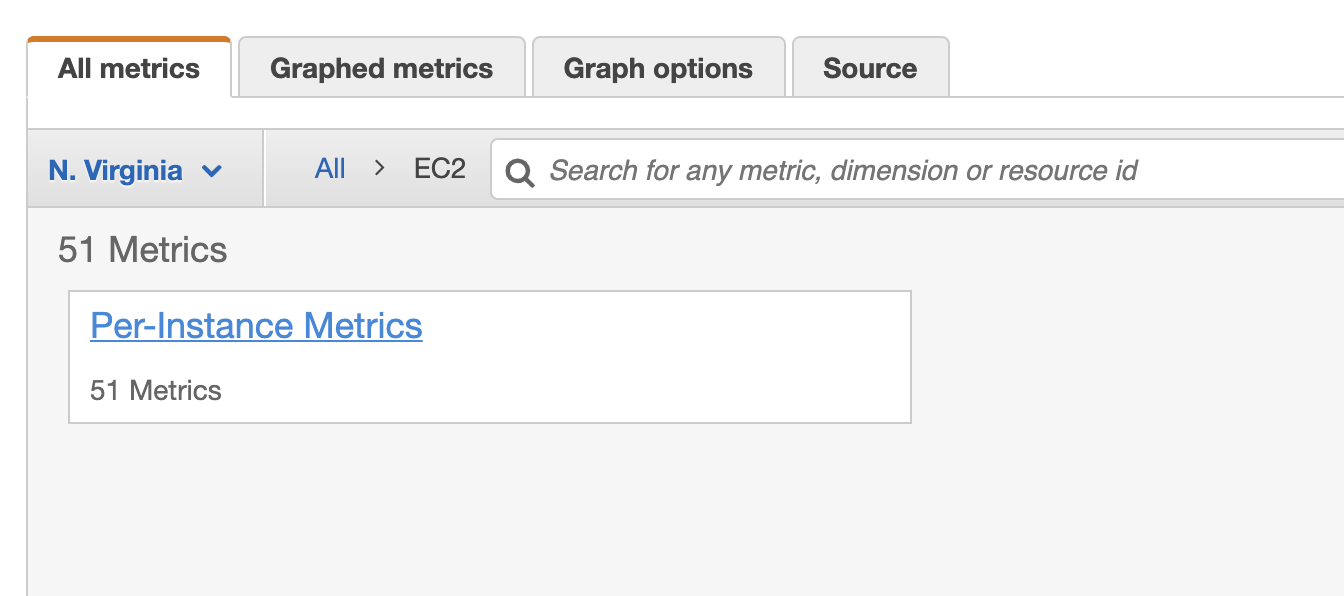
* 1. Select **Metrics** as the data source for the widget and click on Configure.



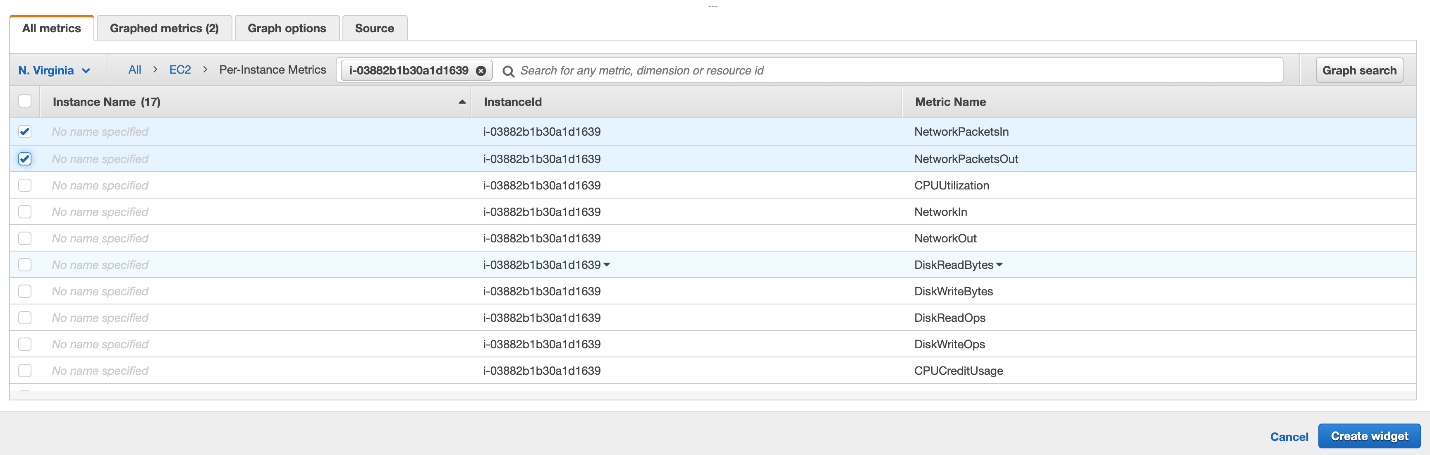
* 1. Select **EC2** from the AWS namespace.



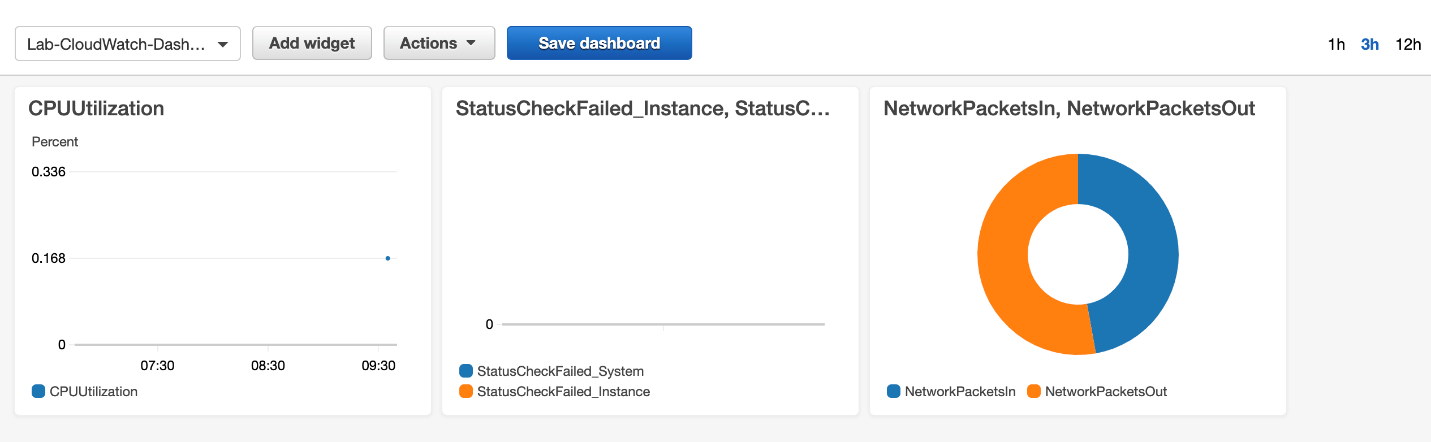
* 1. Select **Per-Instance Metrics**.



* 1. Paste the instance ID (for your EC2 instance that you had copied earlier) in the search bar and hit enter. Select the metrics - **NetworkPacketsIn**and**NetworkPacketsOut**. Click on **Create widget**.



* 1. You should see that the widget has been created successfully.



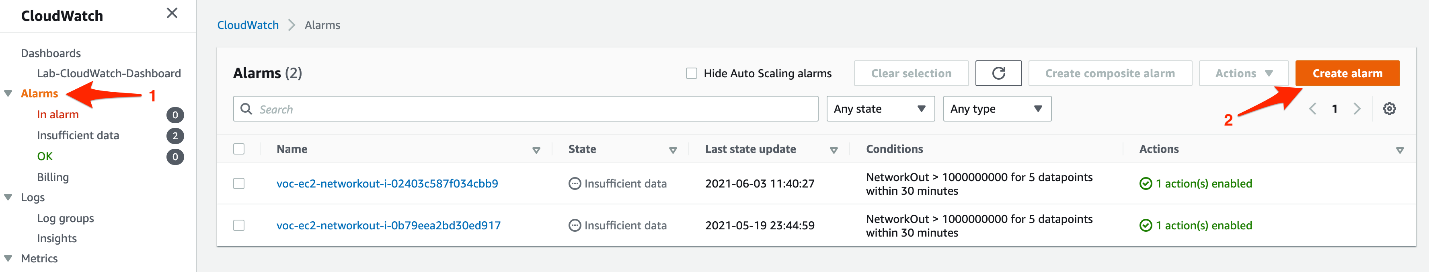
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-and-work-with-widgets.html>

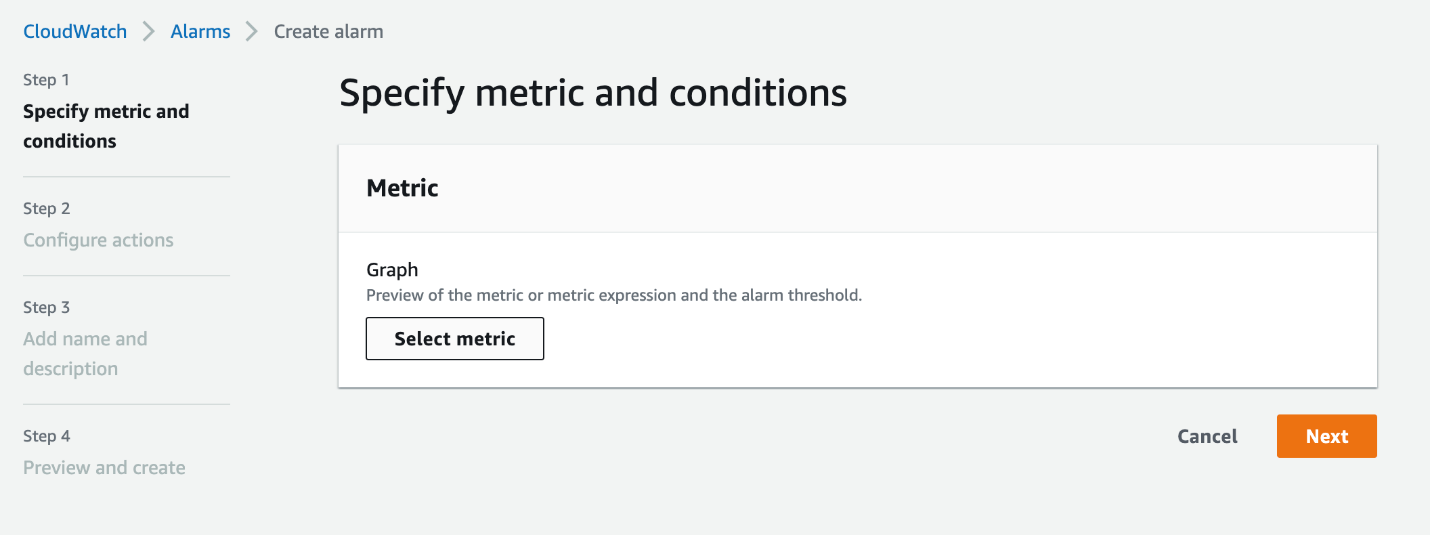
Create cloudwatch dashboard alarm to send sns when instance metric breache the threshold

Configure CloudWatch alarm to send SNS notifications. This will be used to send you an email when the CPUUtilization for the EC2 instance breaches the threshold value.

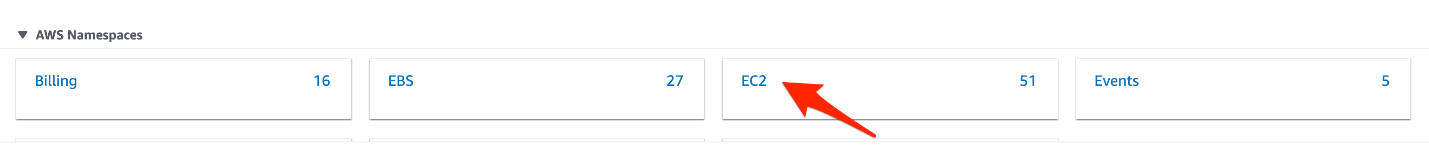
1. Let's create the CloudWatch alarm.
   1. Select the **Alarms** link on the left sidebar and click on the **Create alarm** button.



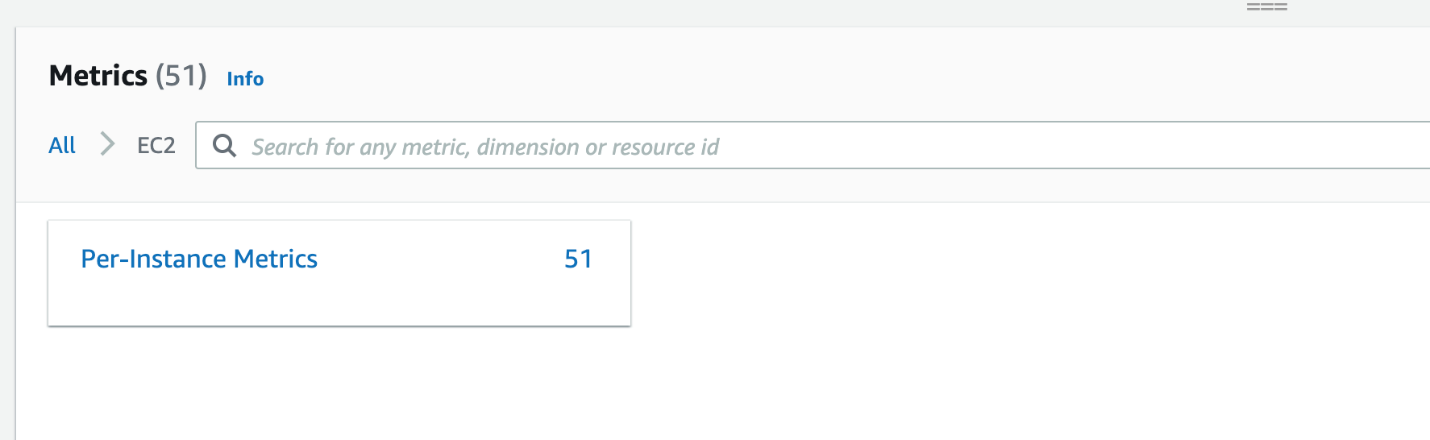
* 1. Click on the **Select metric** button.



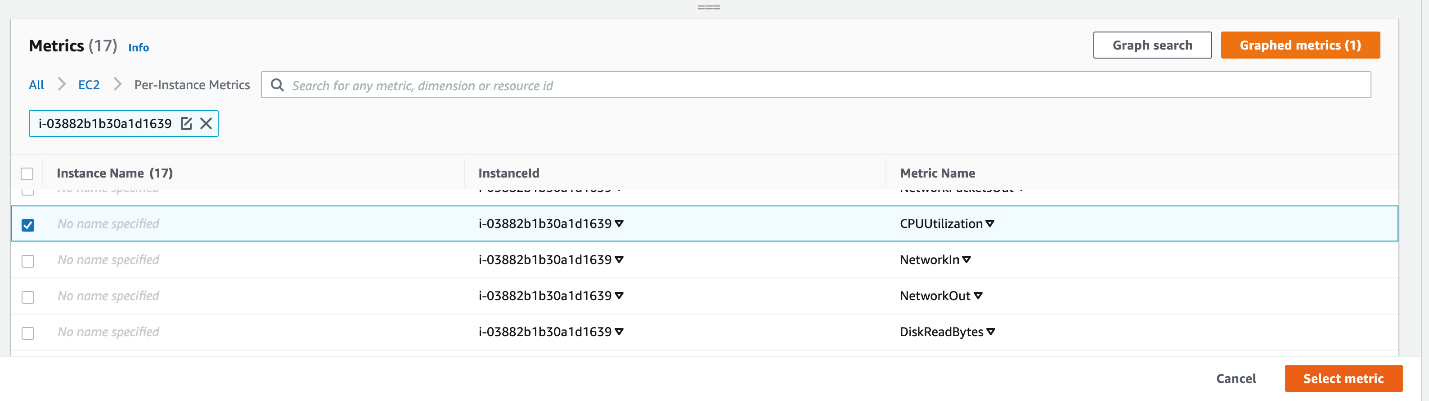
* 1. Select **EC2** from the AWS namespaces.



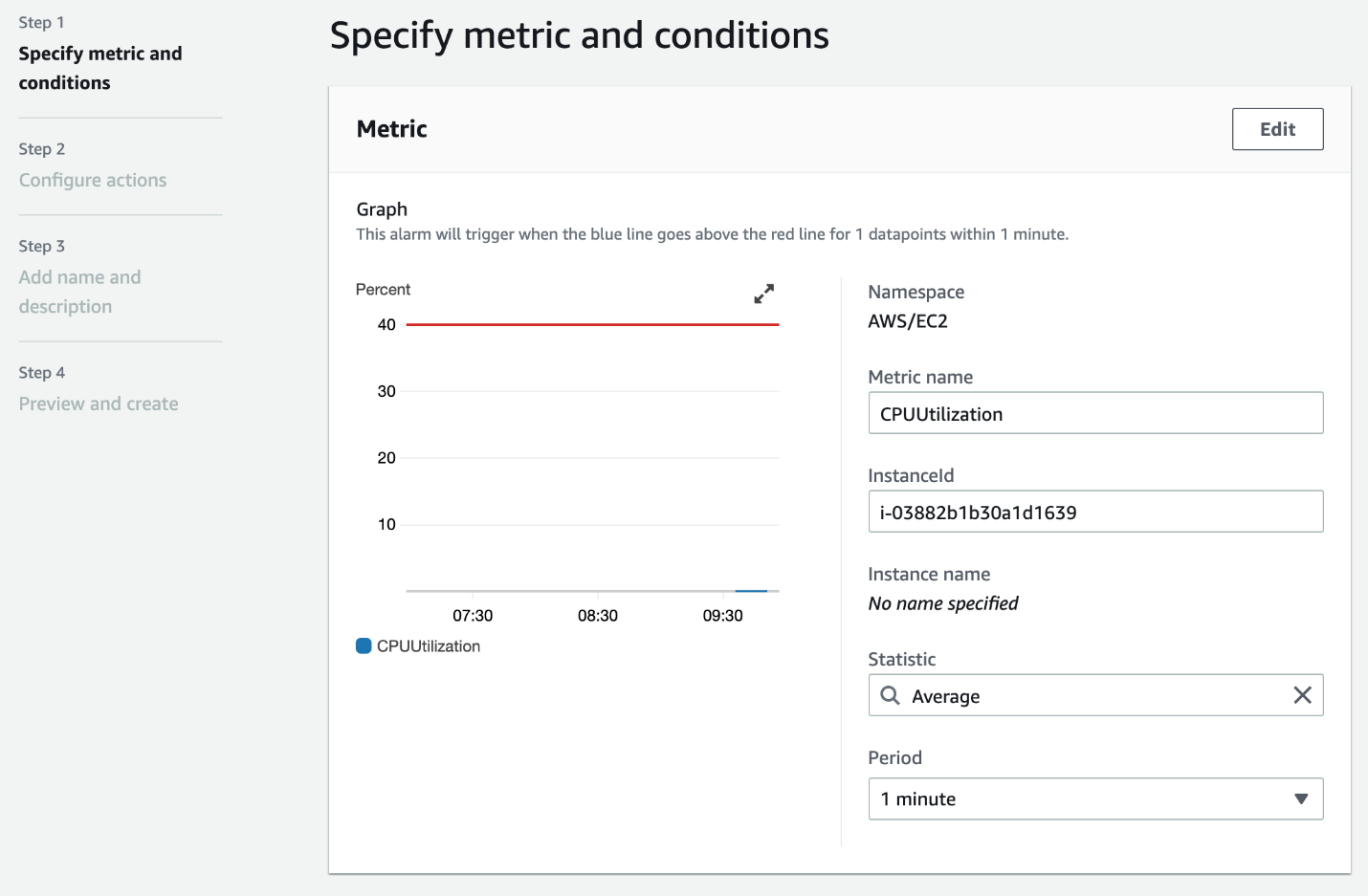
* 1. Select **Per-Instance Metrics**.



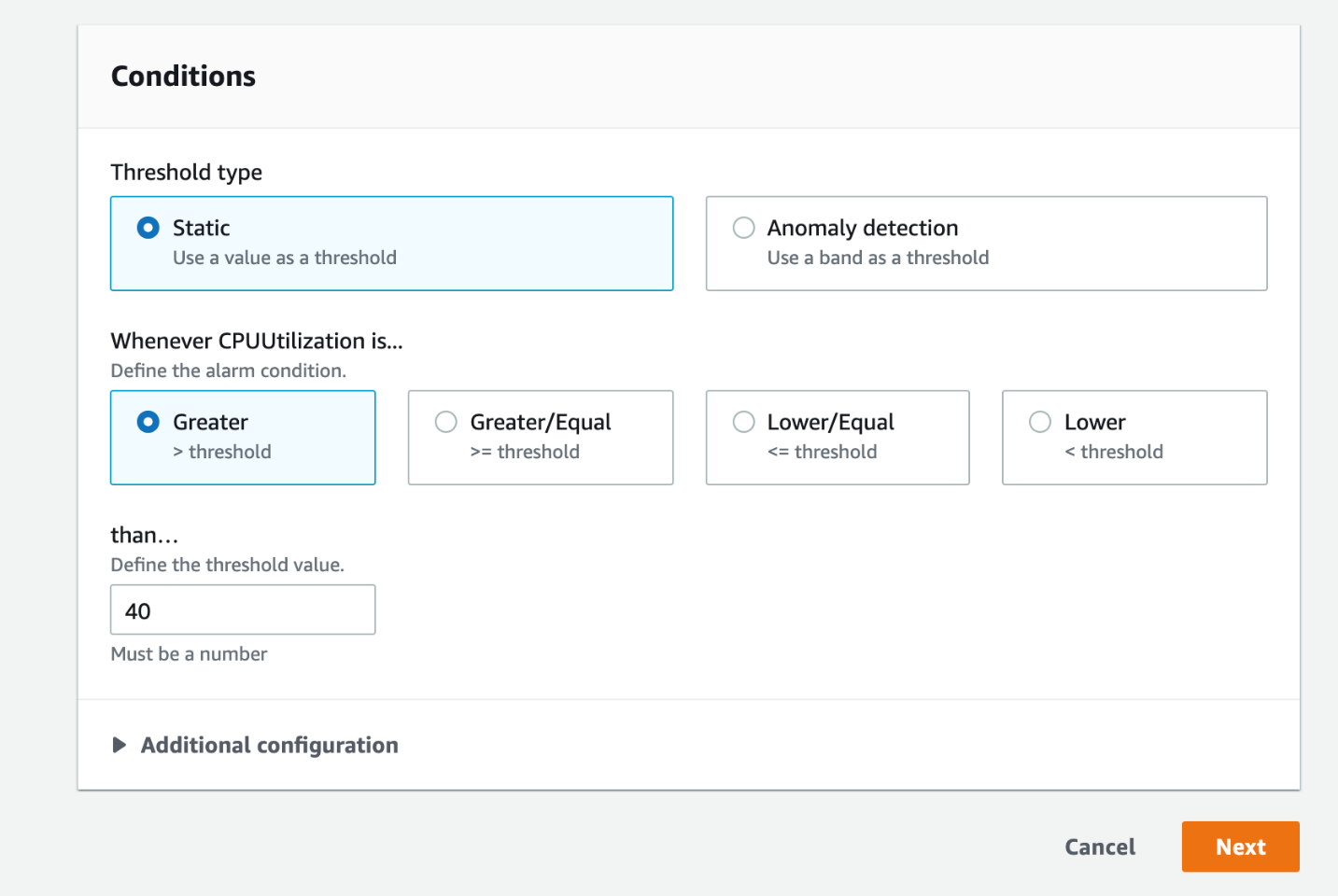
* 1. Paste the instance ID (for your EC2 instance that you had copied earlier) in the search bar and hit enter. Select the metric - **CPUUtilization**. Click on **Select metric**.



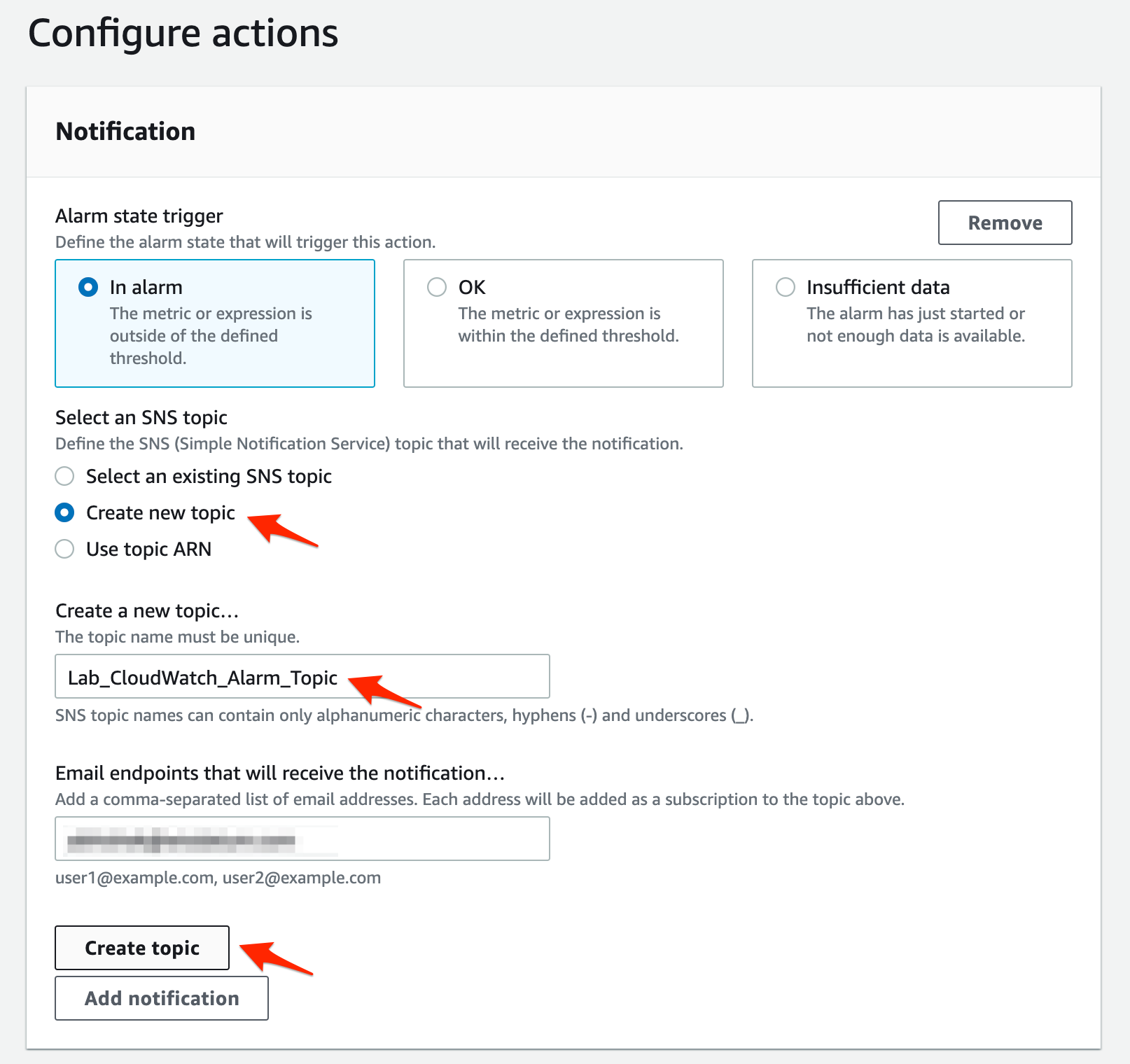
* 1. Specify metric specific configurations. Select **Average** as **Statistic** and select **1 minute** as **Period**.



* 1. Specify condition specific configurations. Select threshold type, alarm condition and threshold value like so:

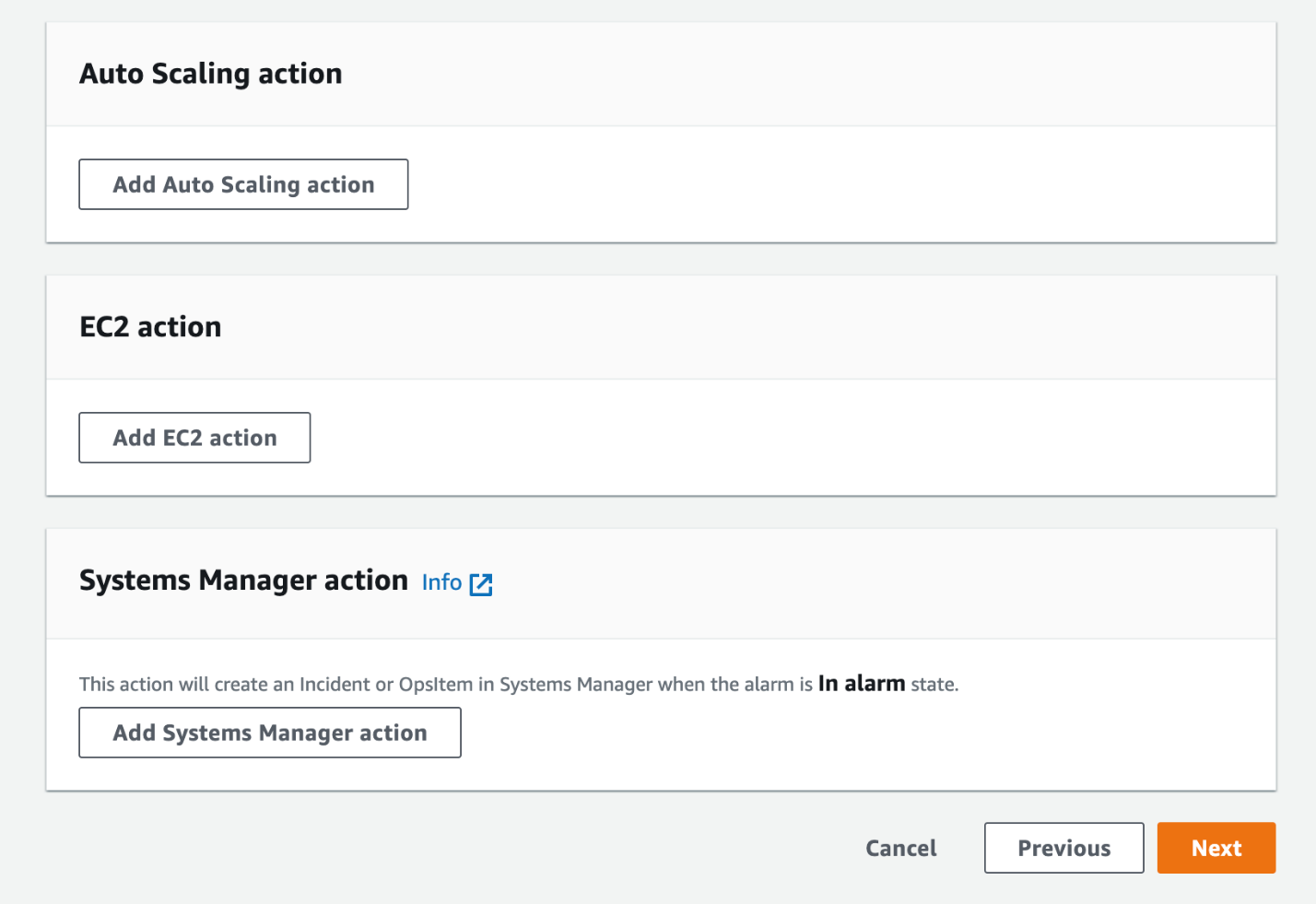


* 1. Now let's configure an action for the alarm. We shall create a **notification action.**This will be used to send you an email when the CPUUtilization for the EC2 instance breaches the threshold value. In the **Notification** section, Select 'In alarm'as the option for **Alarm State Trigger**, select **Create new topic**for SNS topic. Provide a **name** for the topic - **Lab\_CloudWatch\_Alarm\_topic**. Then enter your email address as the **endpoint**. Click on the**Create Topic** button.

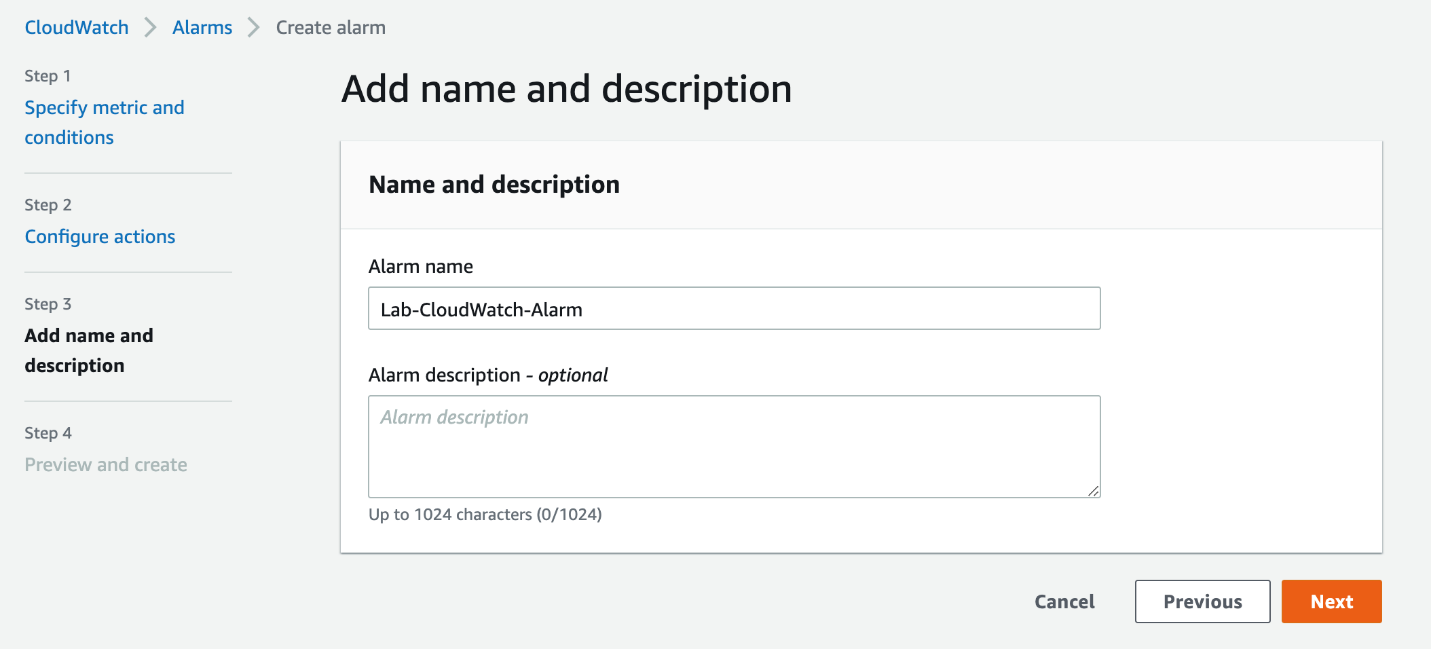


You should see that your topic has been created successfully.

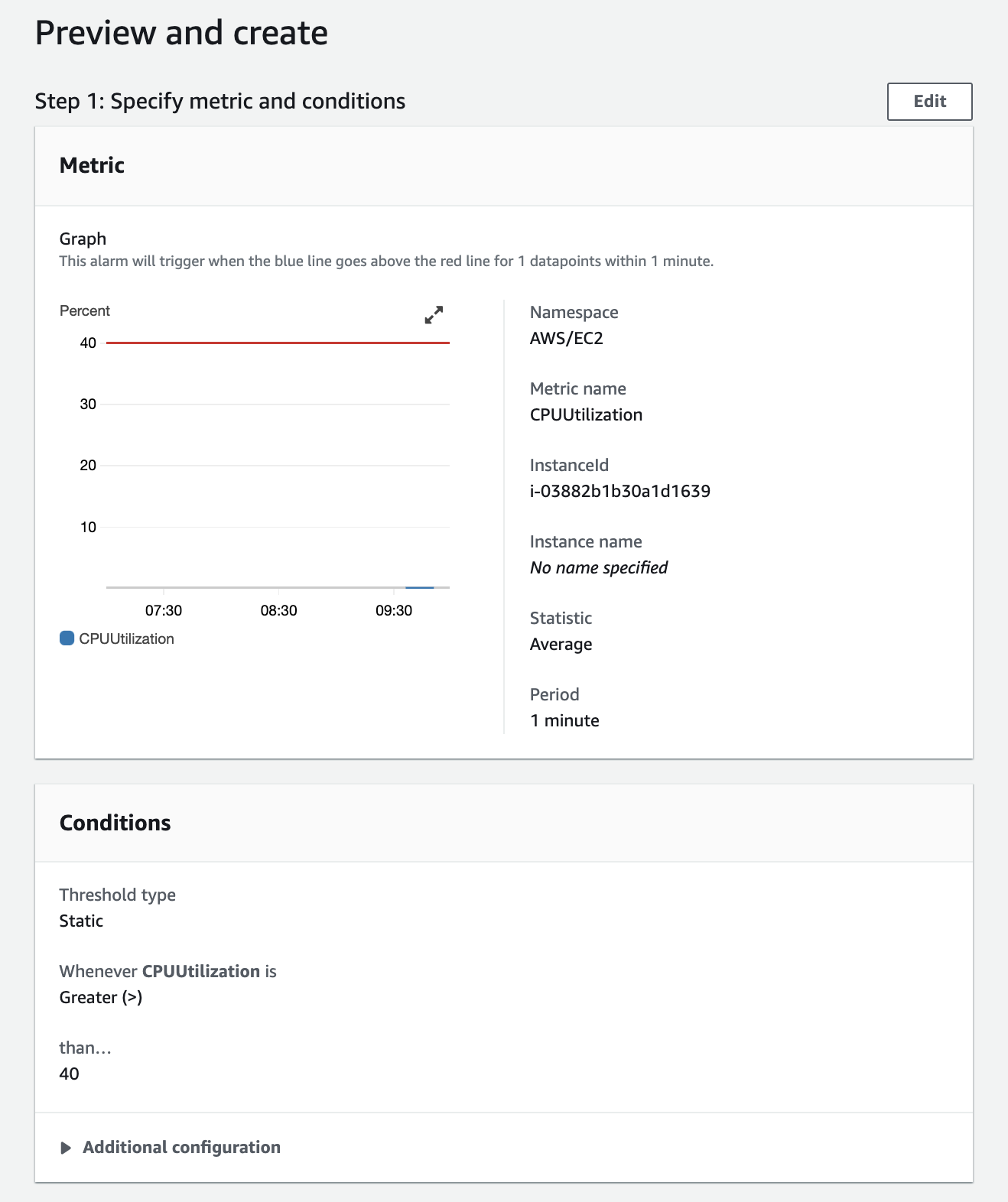
* 1. Leave the other sections unchanged and click on Next.

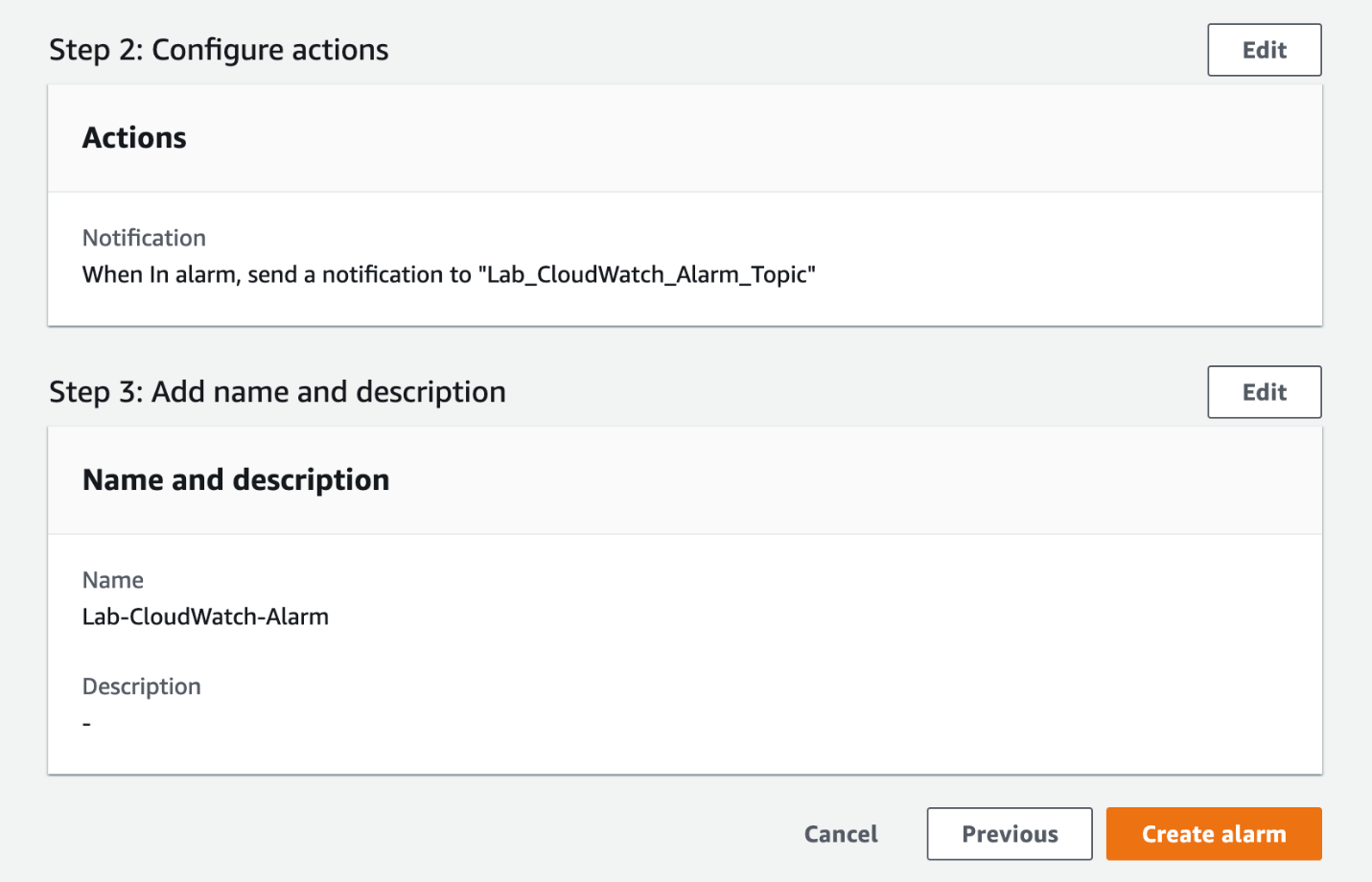


* 1. Enter the alarm name as Lab-CloudWatch-Alarm and click on Next.



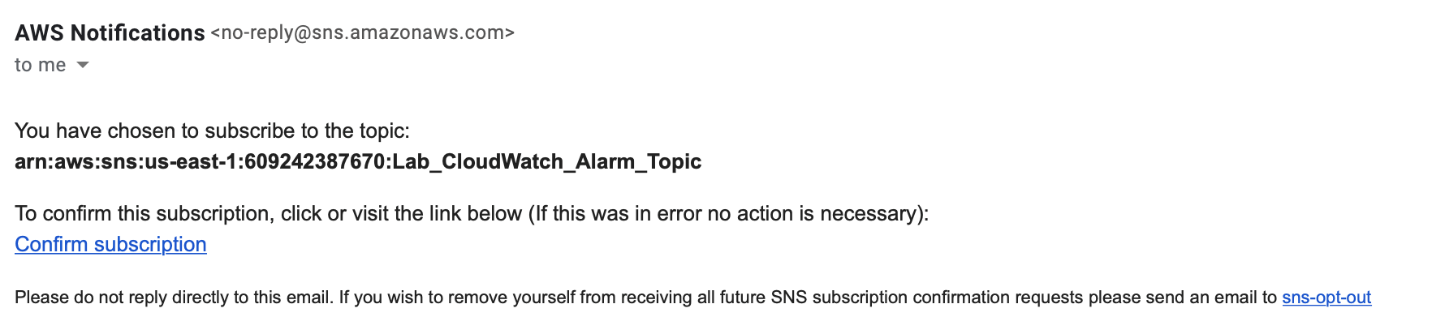
* 1. Preview all sections for your alarm and click on the **Create alarm** button to complete the set up for the alarm.



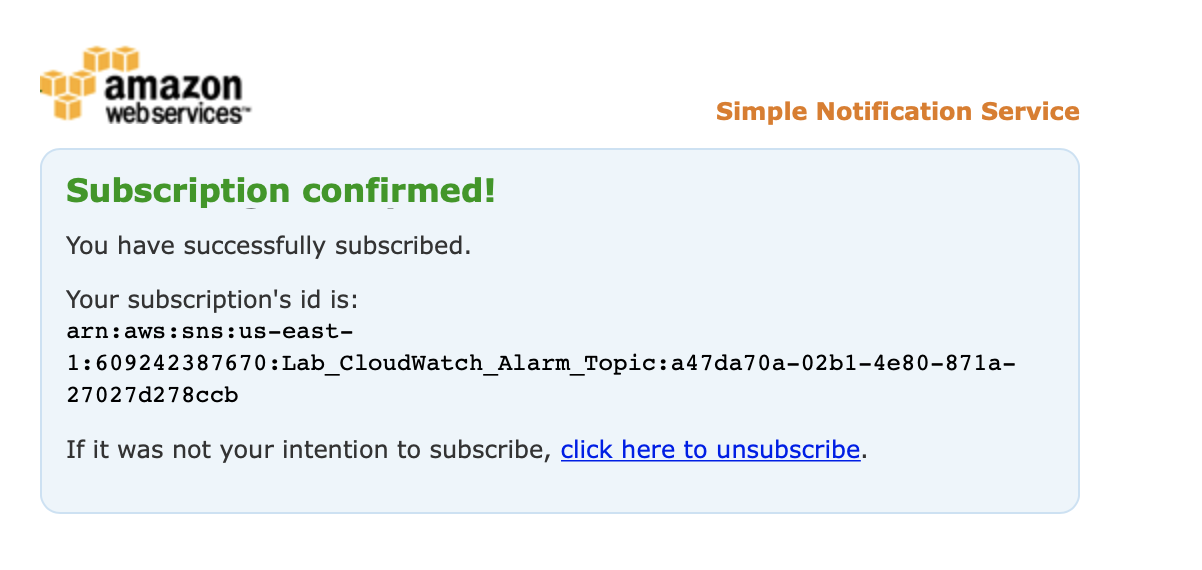


You should see that the alarm has been created successfully. However, the alarm shows up in the pending confirmation status. This is because you need to confirm the subscription for the SNS notifications that will be sent to your email.

* 1. Go to your inbox for the email address you have used as an endpoint for the SNS topic. Click on the **Confirm subscription** link in the email that you have received from AWS Notifications.



You should see a message that you've successfully subscribed to SNS Notifications.



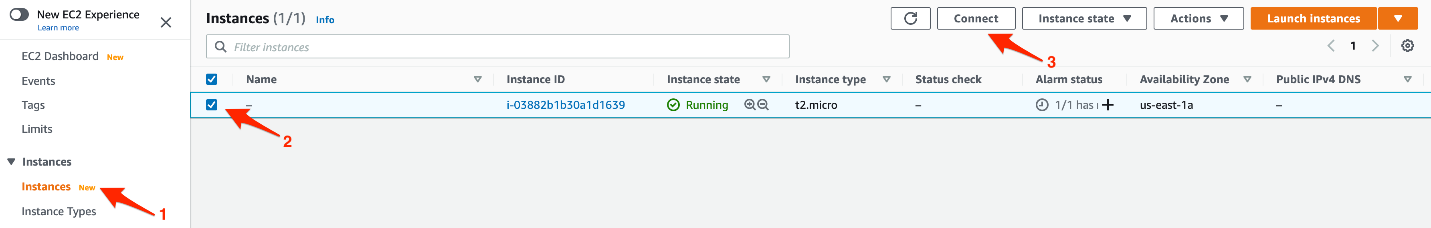
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/US_SetupSNS.html>

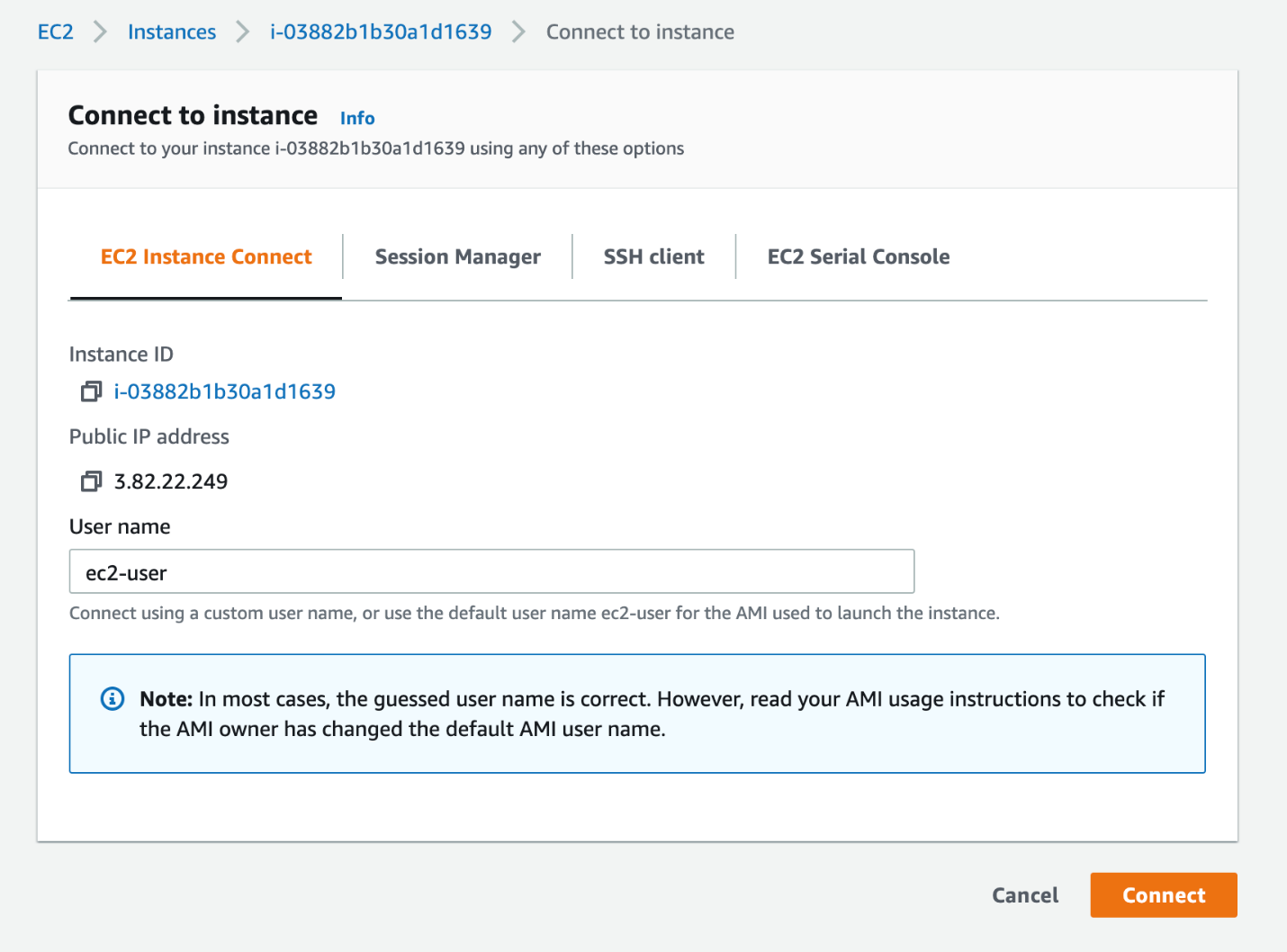
generate synthetic CPU load on EC2 by using stress utility

You should use the **stress** utility to generate synthetic CPU load on the EC2 instance. The increased CPU utilization would trigger the CloudWatch alarm, that in turn would send out an email via SNS notifications. In addition, you would also be able to observe the spike in CPU usage via the CloudWatch dashboard.

1. Make sure that you are in the N.Virginia AWS Region on the AWS Management Console.
   1. Navigate to the EC2 service. Click on **Instances** from the left sidebar. Select the instance you have already provisioned. Click on the **Connect** button.

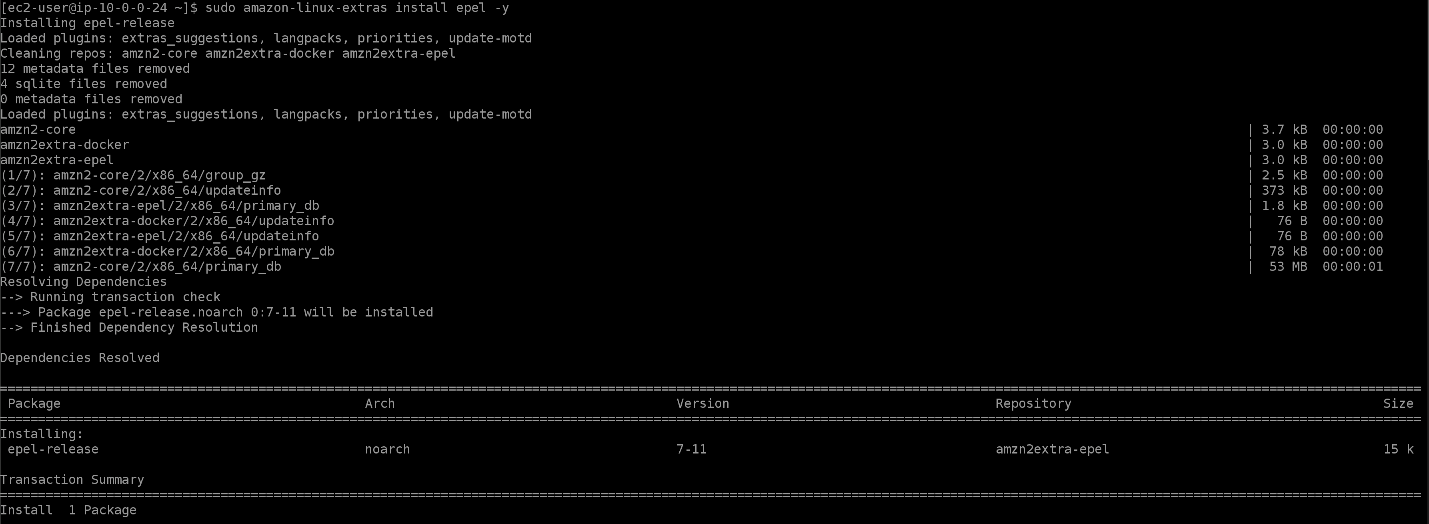


* 1. Select the **EC2 Instance Connect** tab and click on the **Connect** button.



* 1. Run the following commands to install the **stress** utility on the EC2 instance

sudo amazon-linux-extras install epel -y

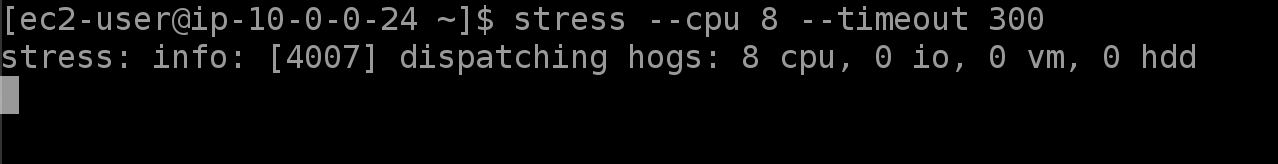


sudo yum install stress -y

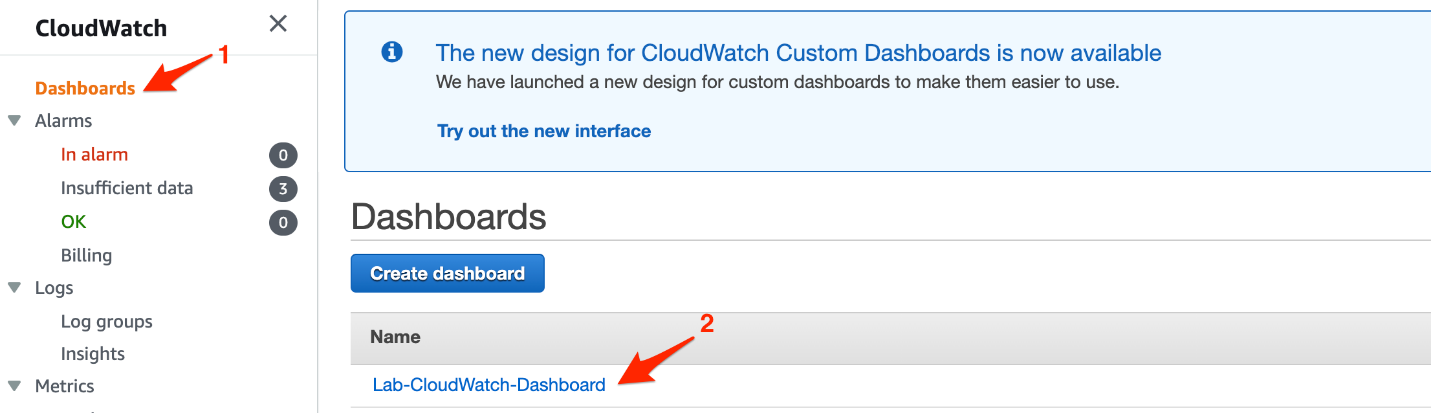


* 1. Invoke the stress test on the instance using the following command. The stress utility generates synthetic CPU load on the EC2 instance. The increased CPU utilization would trigger the CloudWatch alarm.

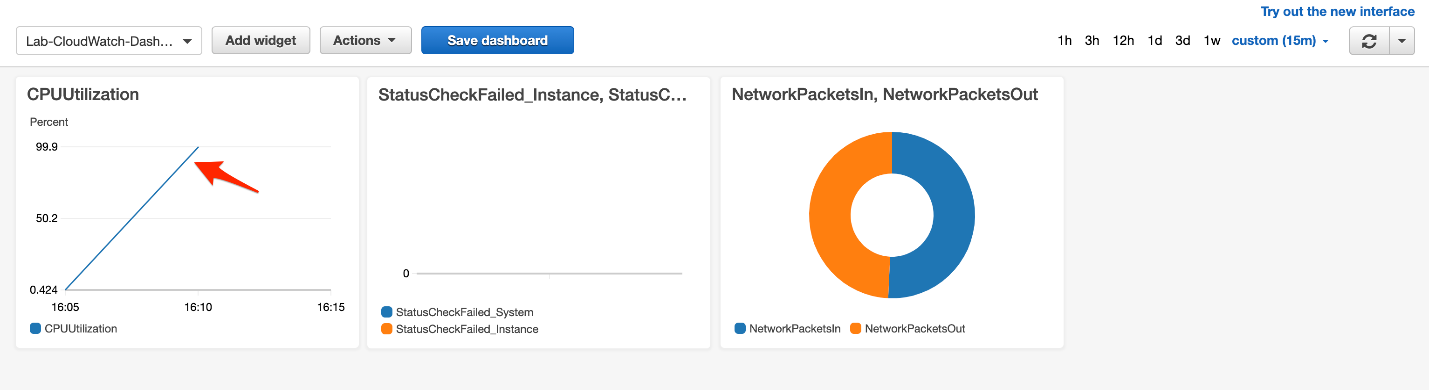
stress --cpu 8 --timeout 300



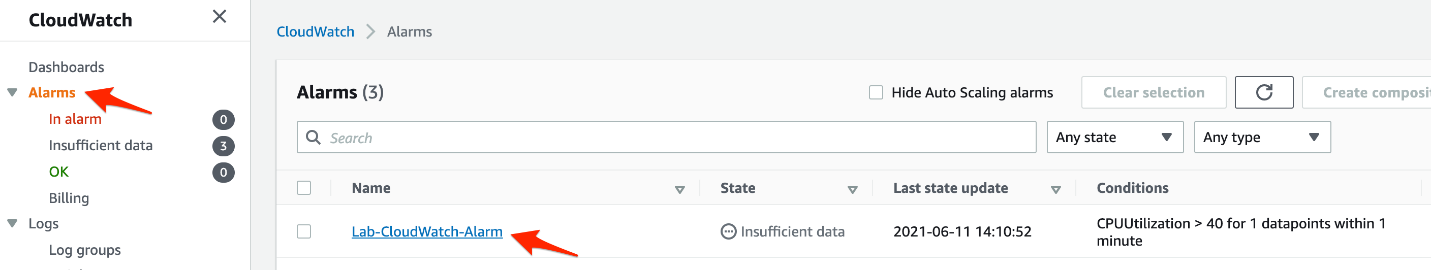
1. Navigate to the CloudWatch service.
   1. Select **Dashboards** from the left sidebar and click on the **Lab-CloudWatch-Dashboard**.



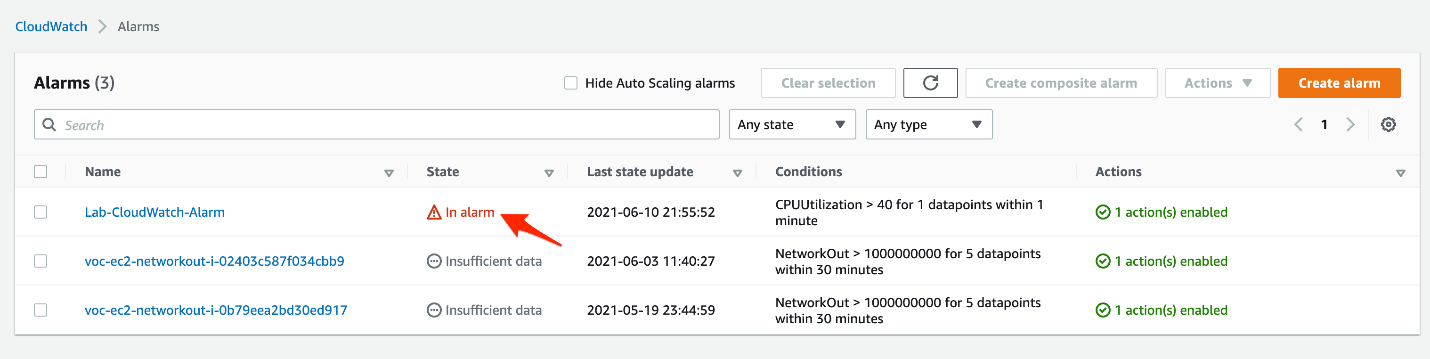
* 1. You should see a spike in the CPU utilization for your EC2 instance within 5 to 10 minutes of running the **stress** test.



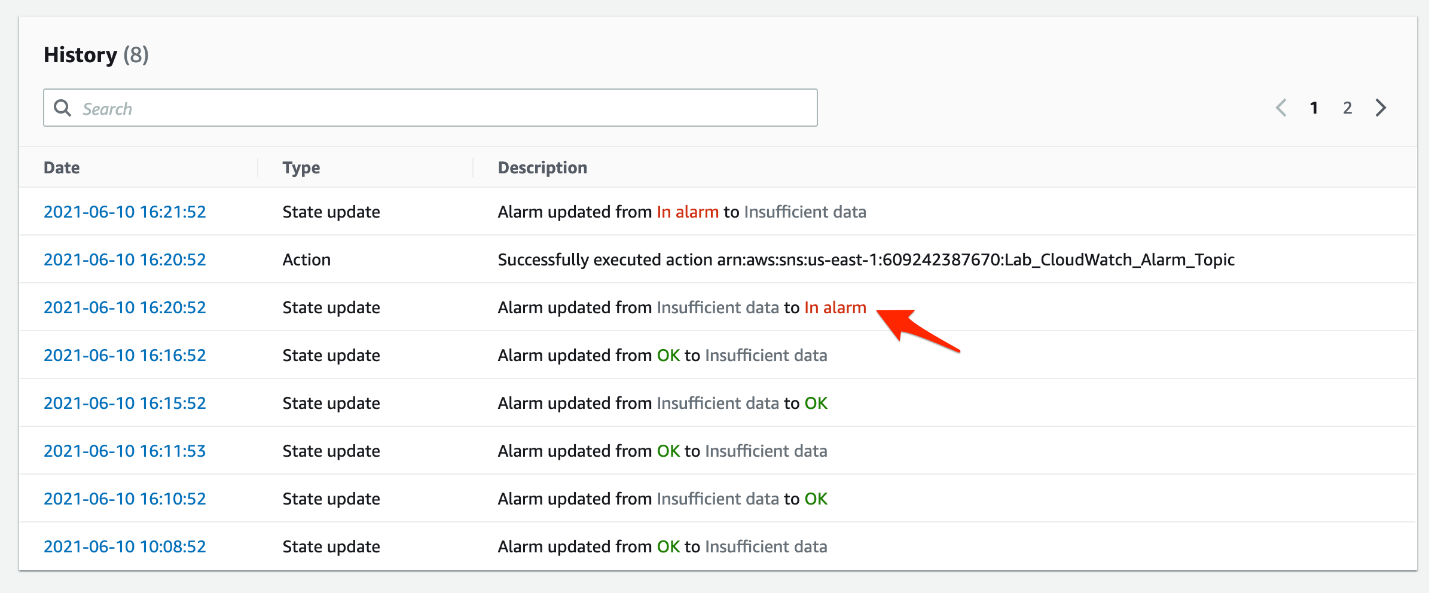
* 1. Navigate to the Alarms section of CloudWatch. The alarm **Lab-CloudWatch-Alarm**will initially be in the**Insufficient data** or **OK** state.



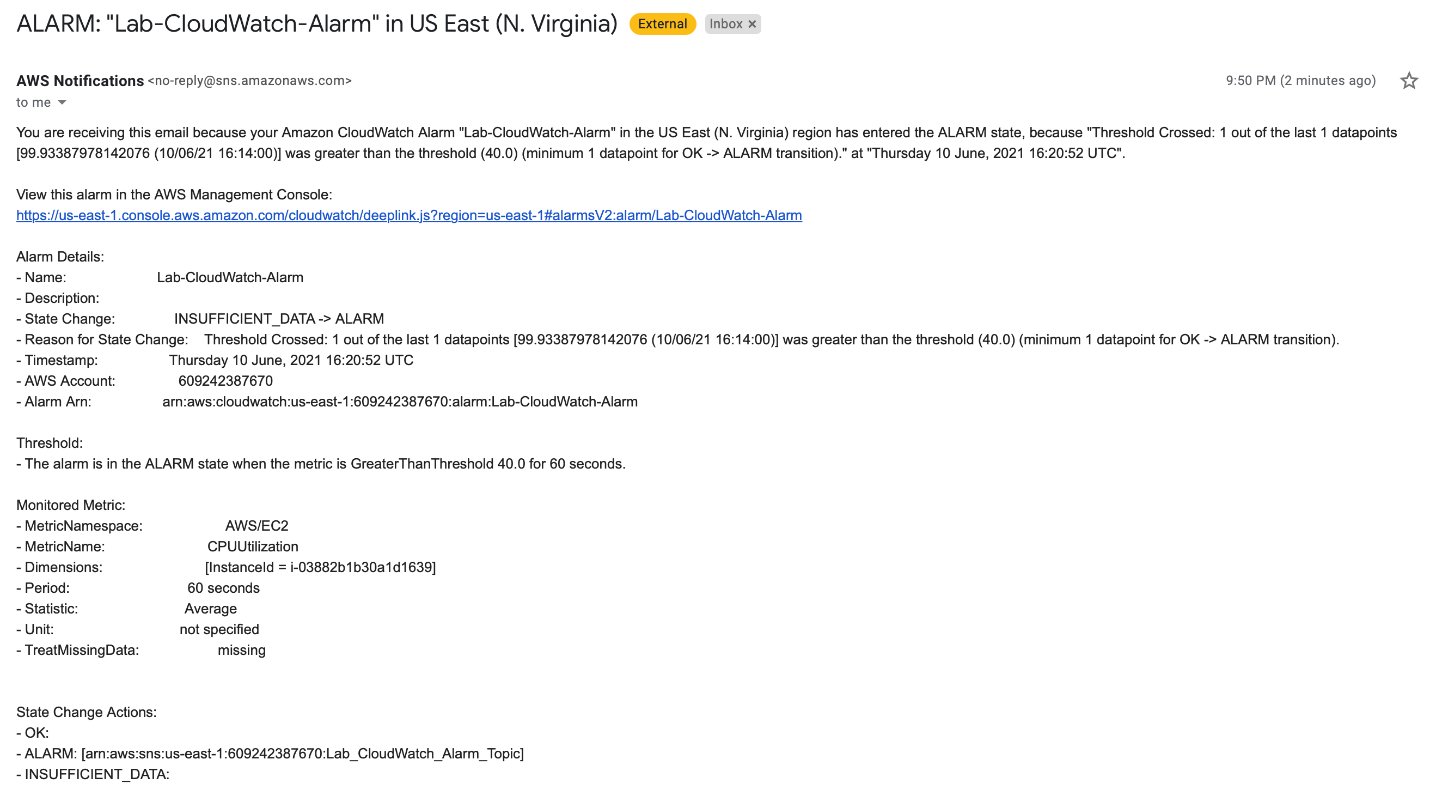
* 1. Within 5 to 10 minutes of invoking the stress test, you should see that the alarm changes the state to **In alarm**.



* 1. Click on the alarm - **Lab-CloudWatch-Alarm**. In the **History** section of the alarm, you should see the history of events where the state changes to **In alarm**.



* 1. You should also receive an email via SNS Notifications with the details of the alarm.



<https://www.tecmint.com/linux-cpu-load-stress-test-with-stress-ng-tool/>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-and-work-with-widgets.html>

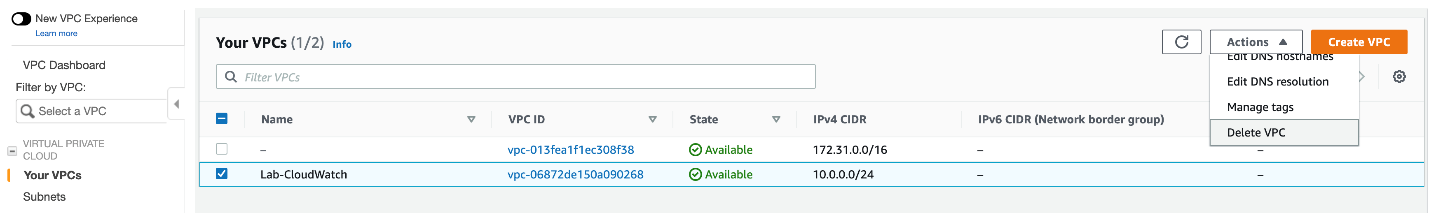
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/US_SetupSNS.html>

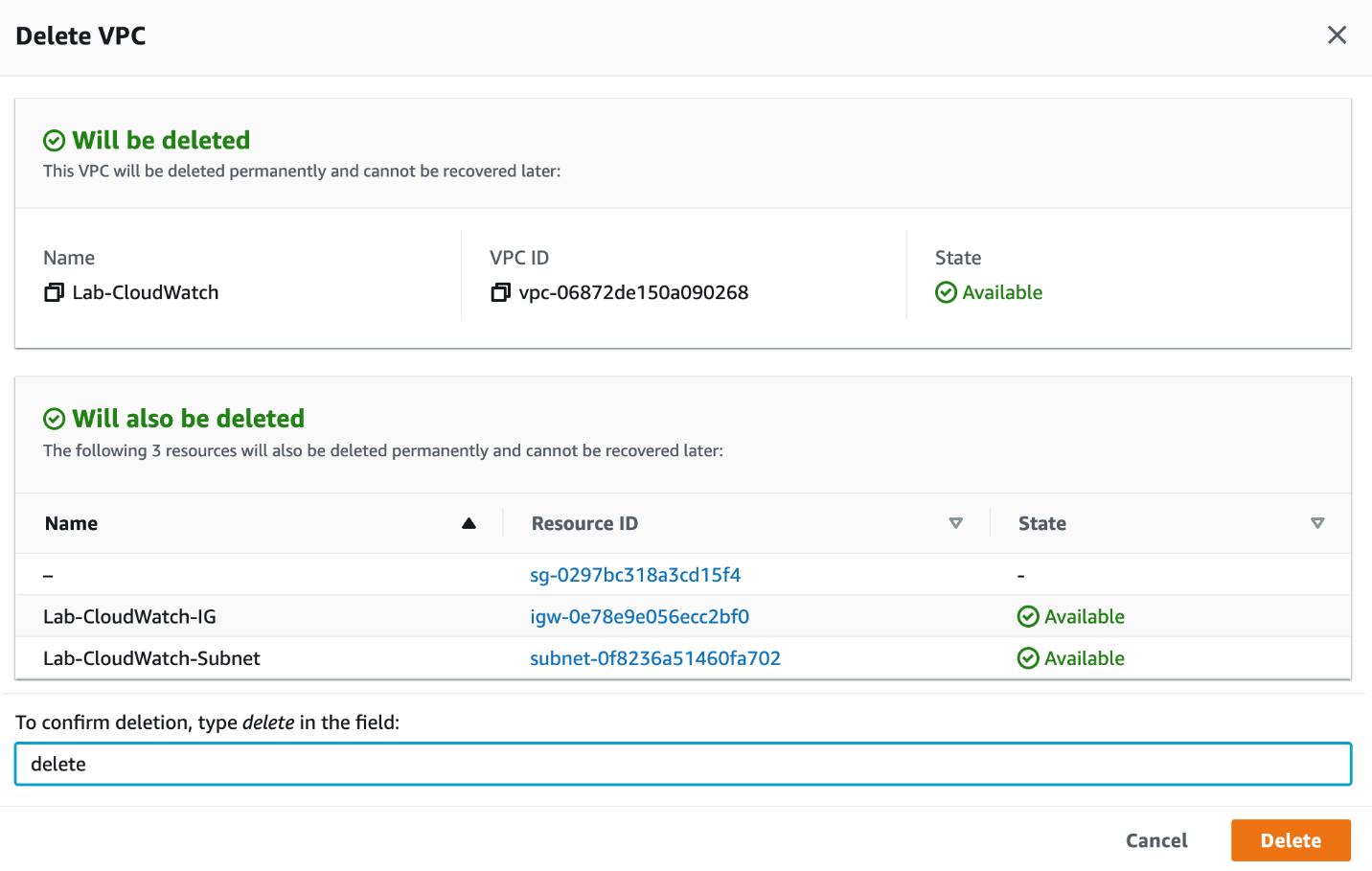
Clean Up

You created a custom VPC, an EC2 instance, a CloudWatch Dashboard along with widgets and a CloudWatch alarm . As this is a proof of concept, you should now delete these resources.

1. Navigate to the EC2 service on the AWS management console. Select the **Instances**link on the left sidebar. Select the EC2 instance that you created. Click the **Instance state**button, select **Terminate instance**and then click on **Terminate**.
2. Navigate to the VPC service on the AWS management console. Let's delete the VPC you created.
   1. Click on the **Your VPCs** link on the left sidebar, then select your VPC **Lab-CloudWatch**, click on the Actions menu and select Delete VPC.



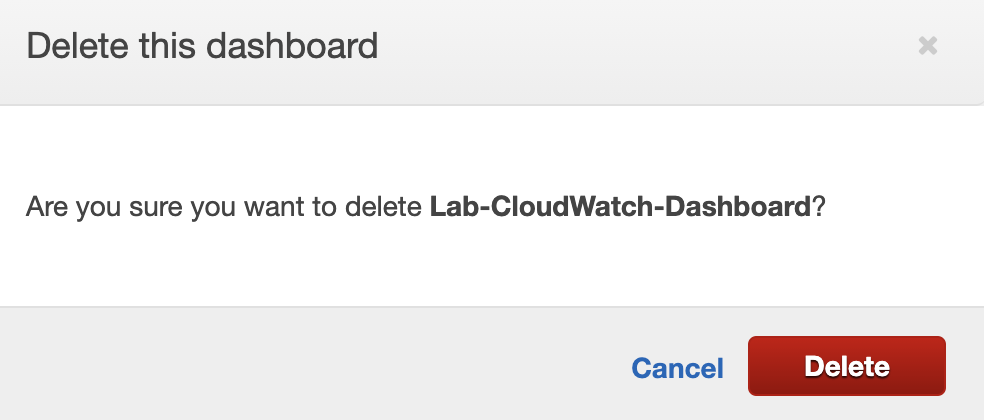
* 1. Type delete and click on the **Delete** button.



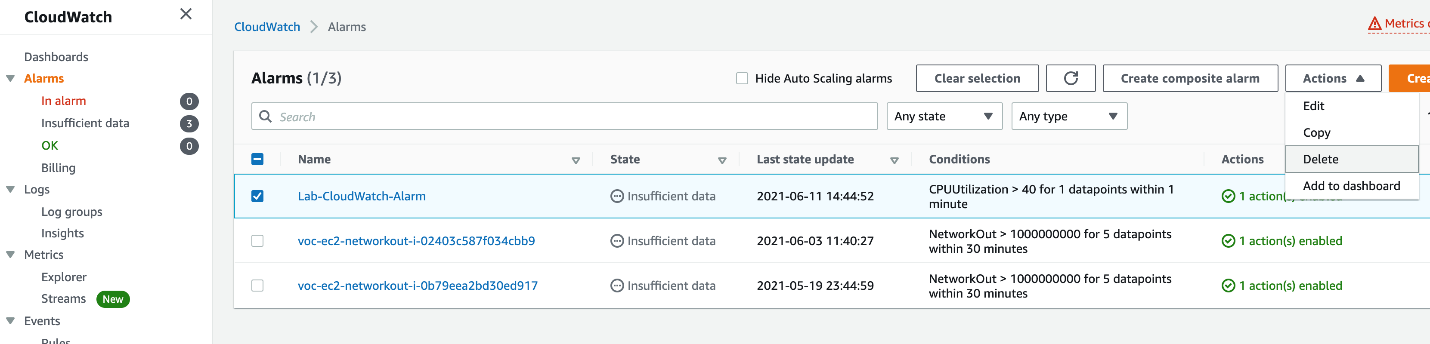
1. Navigate to the CloudWatch service on the AWS management console. Let's delete the CloudWatch dashboard and the CloudWatch alarm.
   1. Click on the Dashboards link on the left sidebar. Click on your dashboard - **Lab-CloudWatch-Dashboard**.
   2. Click on the **Actions** button and then click on the **Delete dashboard** option.



Click on the **Delete** button.



* 1. Click on the **Alarms** link on the left sidebar. Select the alarm - **Lab-CloudWatch-Alarm**and click on the **Action** button. Click on the **Delete** option.



Click on the **delete** button on the popup to confirm.