|  |  |
| --- | --- |
| **Project Title** | **Designing an Automatic Data Collection and Storage System with AWS Lambda and Slack Integration for Server Availability Monitoring and Slack Notification** |
| **Technologies** | **AWS Lambda, Amazon RDS, CloudWatch, Slack API** |

**Problem Statement:**

You are tasked with creating an AWS Lambda function that will periodically fetch data from an [API](http://api.open-notify.org/iss-now.json) and store it in an Amazon RDS instance. The function should be triggered by an Amazon CloudWatch Event that occurs every 15 seconds.

To fetch the data from the API, the function should use the requests library (or a similar library) to make a GET request to the API. The function should then use a library such as psycopg2 to connect to the Amazon RDS instance and store the data in the database.

In addition to fetching and storing the data, the function should also use Amazon CloudWatch to monitor the server and send an alert to a Slack community if the server goes down. This can be done using the Slack API.

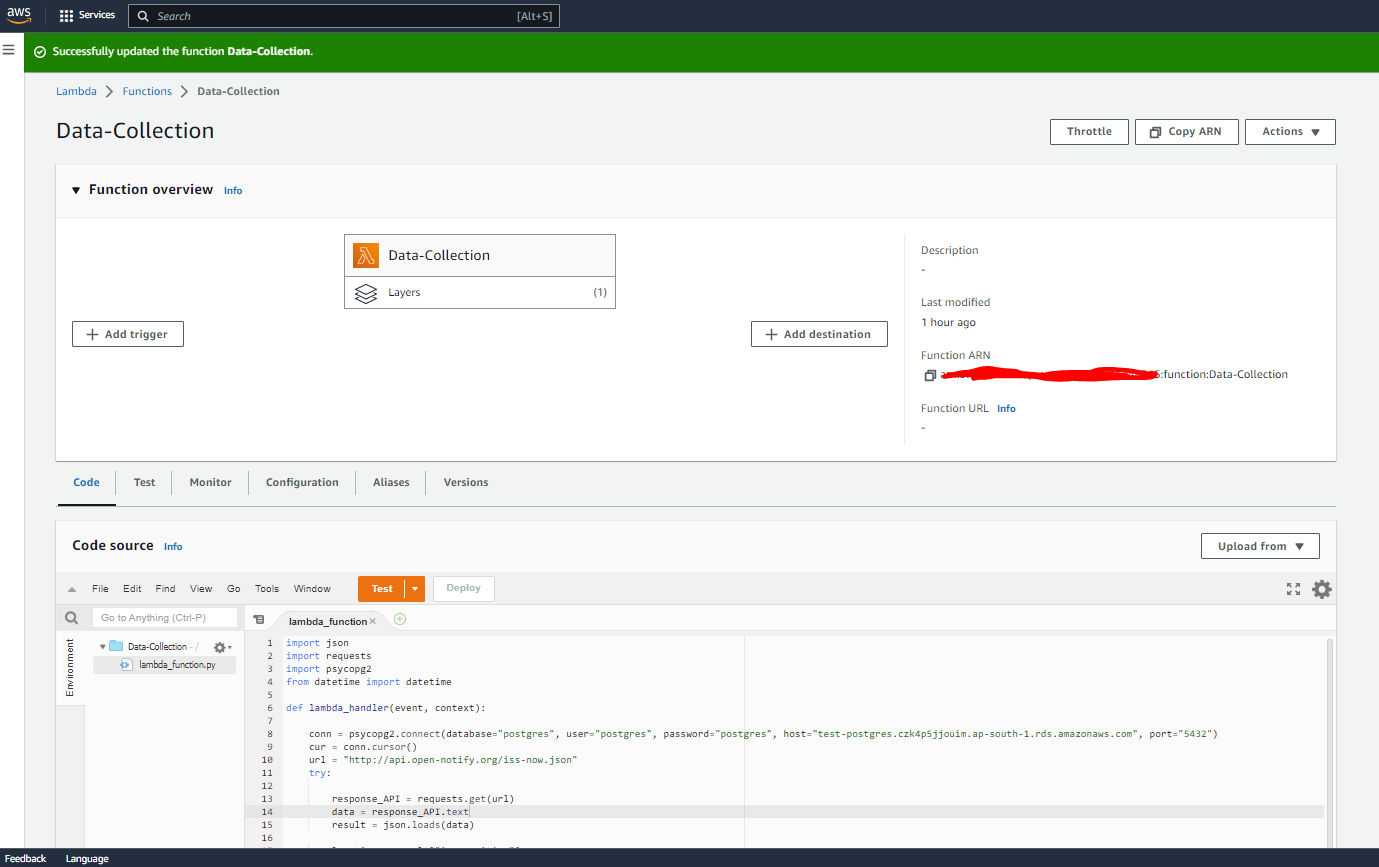
Overall, the function should be able to run indefinitely and continue to fetch and store the data on a regular basis.

**Approach:**

1. Create an AWS Lambda function and configure it to be triggered by an Amazon CloudWatch Event that occurs every 15 seconds.
2. In the function's code, use the requests library to make a GET request to the API to fetch the data.
3. Use a library such as psycopg2 to connect to the Amazon RDS instance and store the data in the database.
4. Use Amazon CloudWatch to set up a monitoring alarm that will trigger when the server is unavailable.
5. Use the Slack API to send a message to your Slack community when the alarm is triggered.
6. Test the function to ensure that it is able to fetch and store the data correctly, and that the monitoring and alerting functionality is working as expected.
7. Deploy the function to run indefinitely, continuing to fetch and store the data on a regular basis.

**SOLUTION:**

**1.CREATE a LAMBDA function to fetch the data from an open source API-**  <http://api.open-notify.org/iss-now.json>.



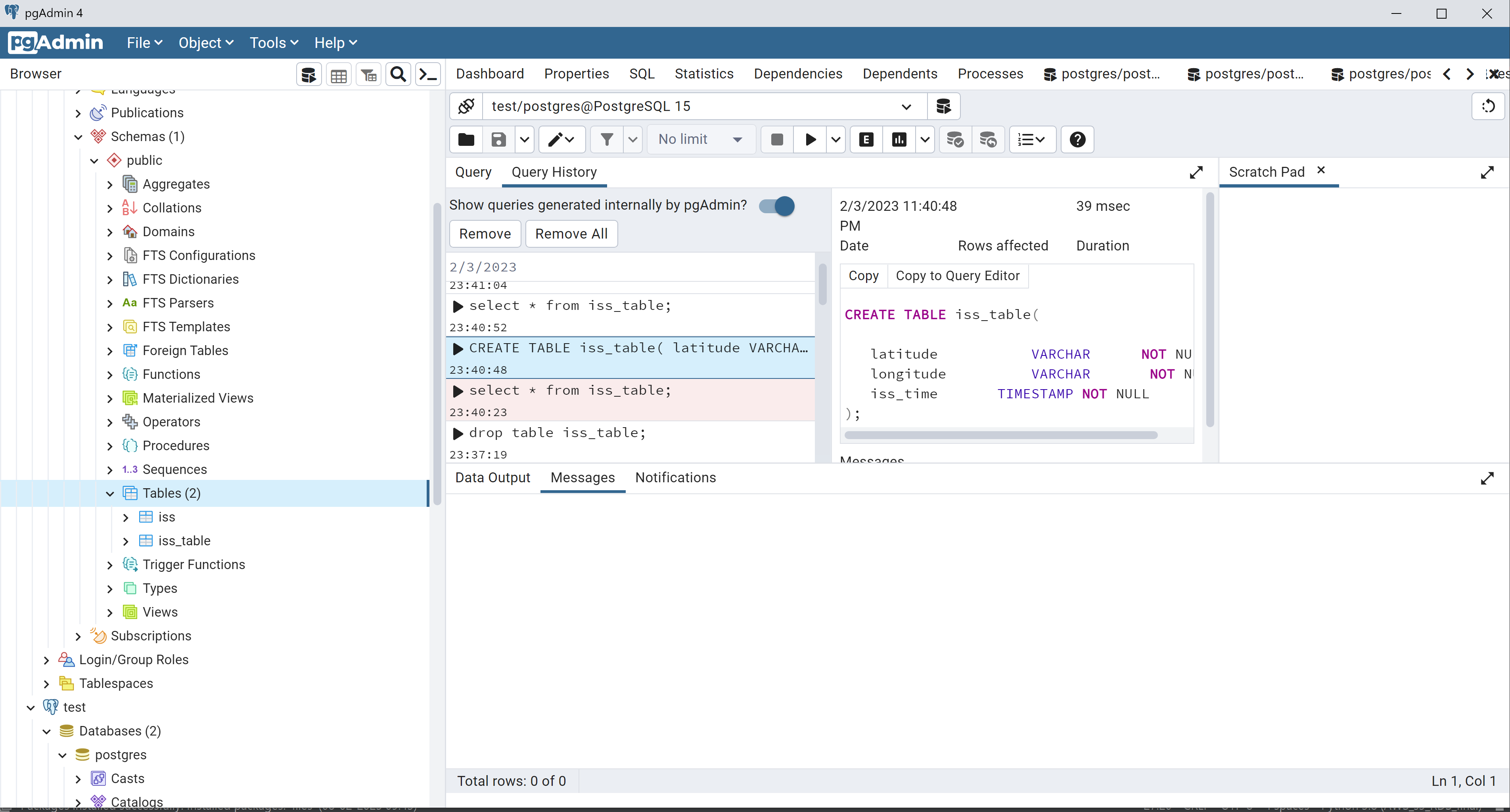


2.Create the postgres DB in AWS RDS

Graphical user interface, application, Word

Description automatically generated

3.Connect to PgAdmin4



* Connect the Amazon RDS postgres to pgAdmin4 and create the database and table to entry the the data into the db.
* Create table postgres
* CREATE TABLE iss\_table(
* latitude VARCHAR NOT NULL,
* longitude VARCHAR NOT NULL,
* iss\_time TIMESTAMP NOT NULL
* );
* select \* from iss\_table;
* drop table iss\_table;

**4.Create a connection to postgres db and store the data to Amazon RDS postgres db whichever read from API .**

****

**5. Create a trigger for the above lambda function by an Amazon CloudWatch Event that occurs every 15 seconds.**

**Schedule your lambda function from cloudwatch:**

**🡪create a cloudwatch event rule:**

Go to cloudwatch console->Events->Rules->Create Rule

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

This trigger will Schedule the lambda function run for every 1 minute and u can watch logs in cloudwatchlogs

Graphical user interface, text, application

Description automatically generated

Creating Lambda layers for

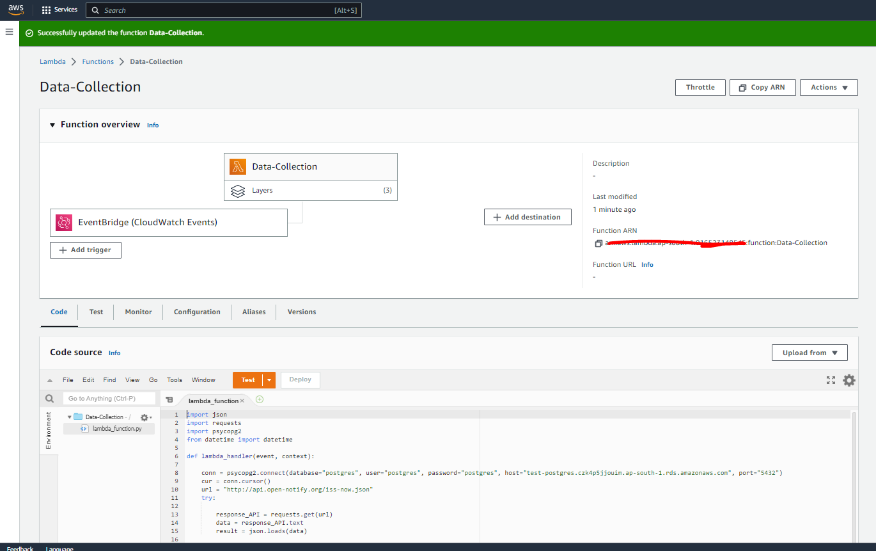
Graphical user interface, application

Description automatically generated

Graphical user interface, application

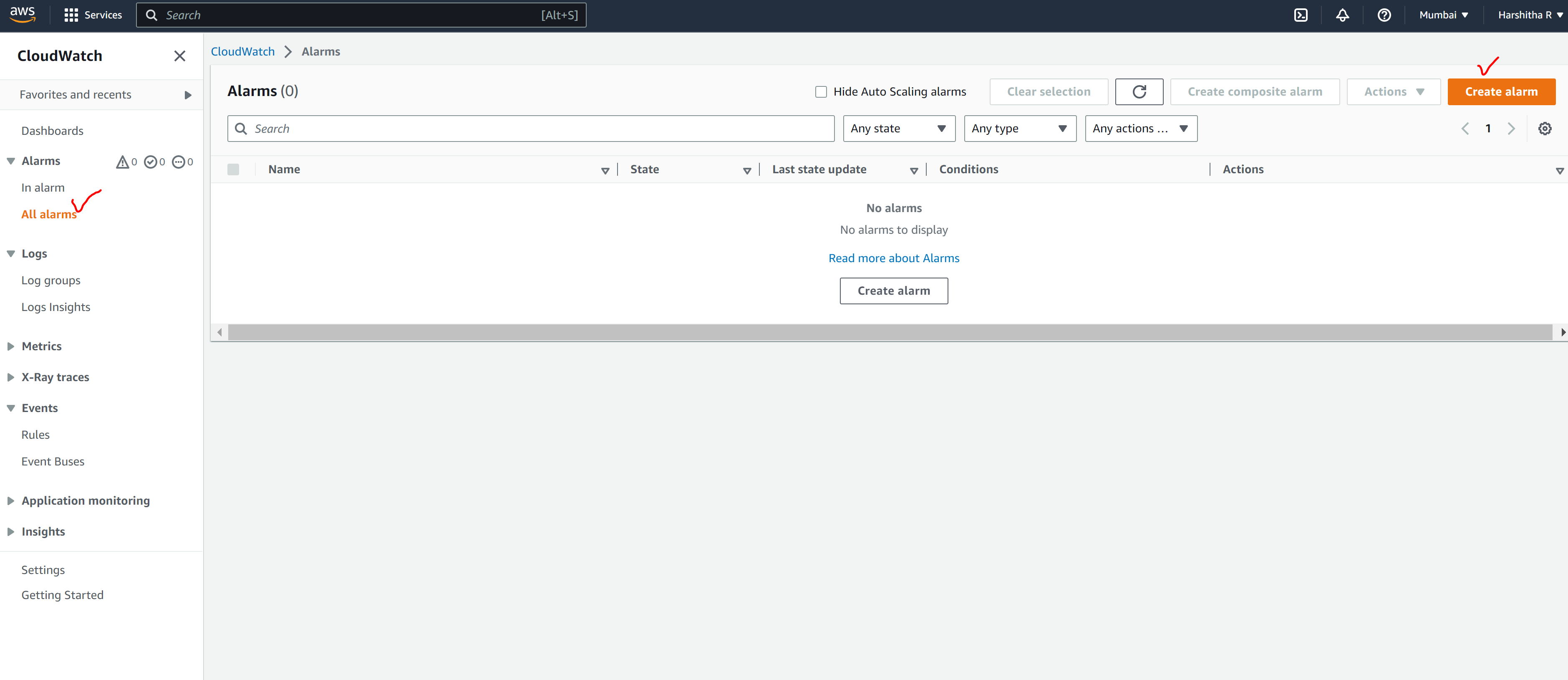
Description automatically generated

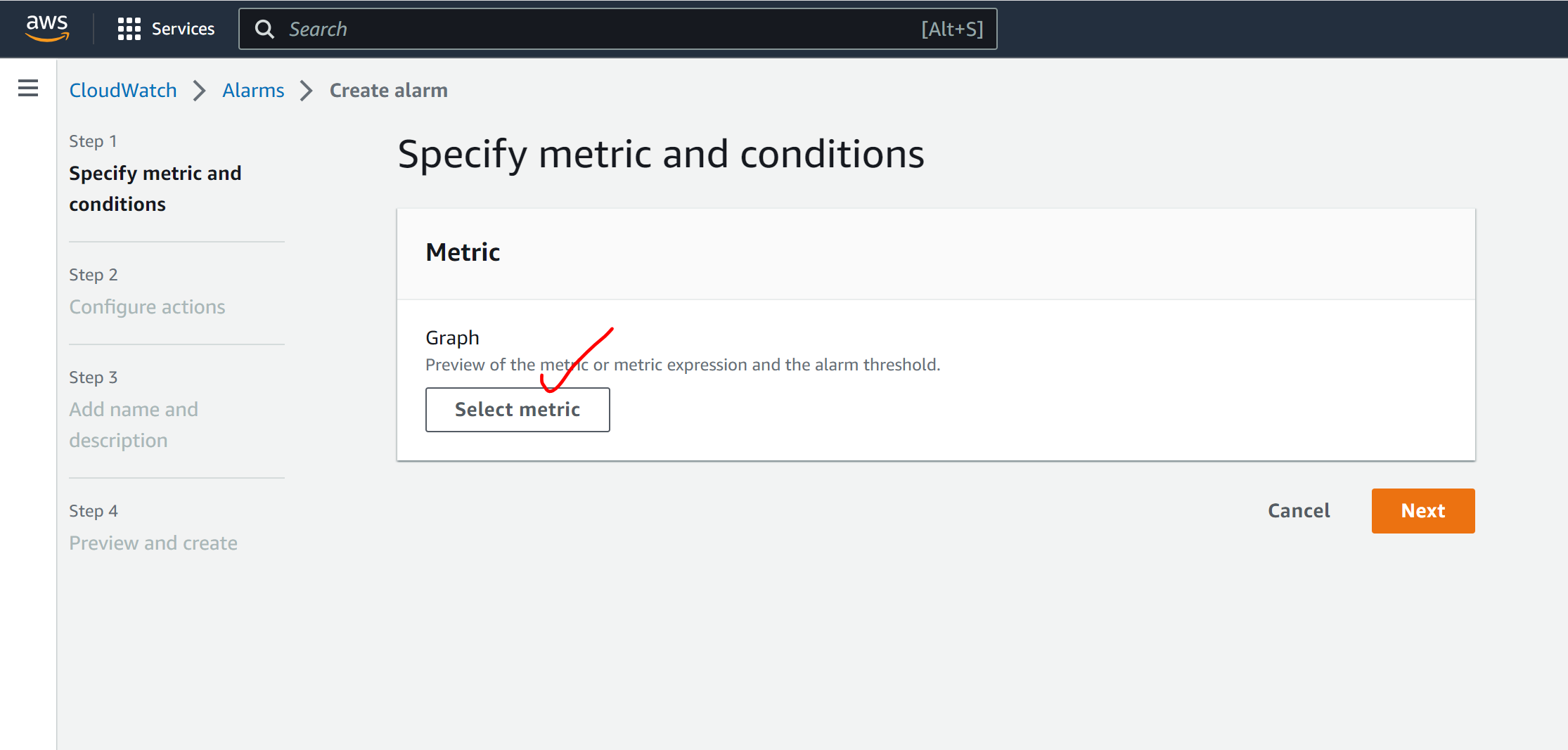
Create a trigger for scheduling the lambda function for every 1 minute.

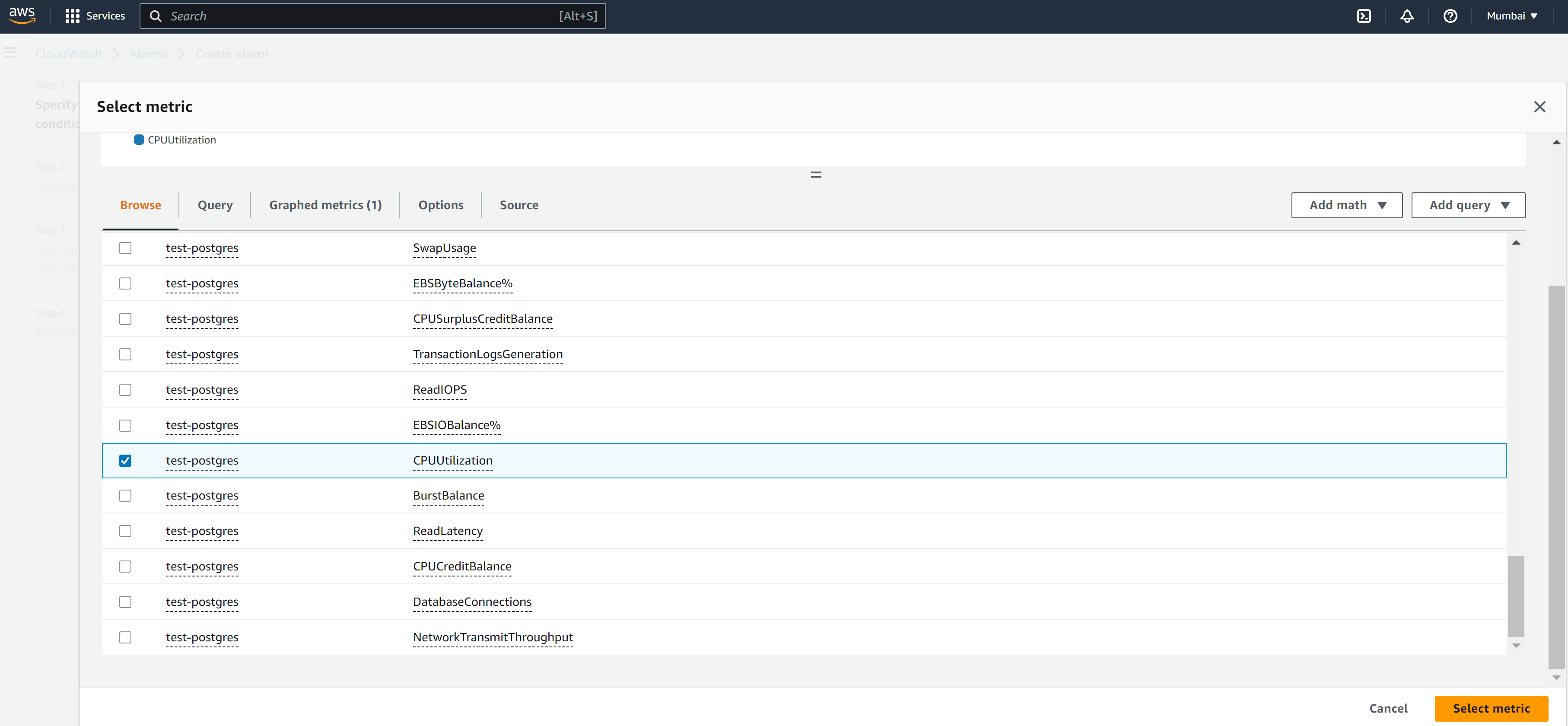


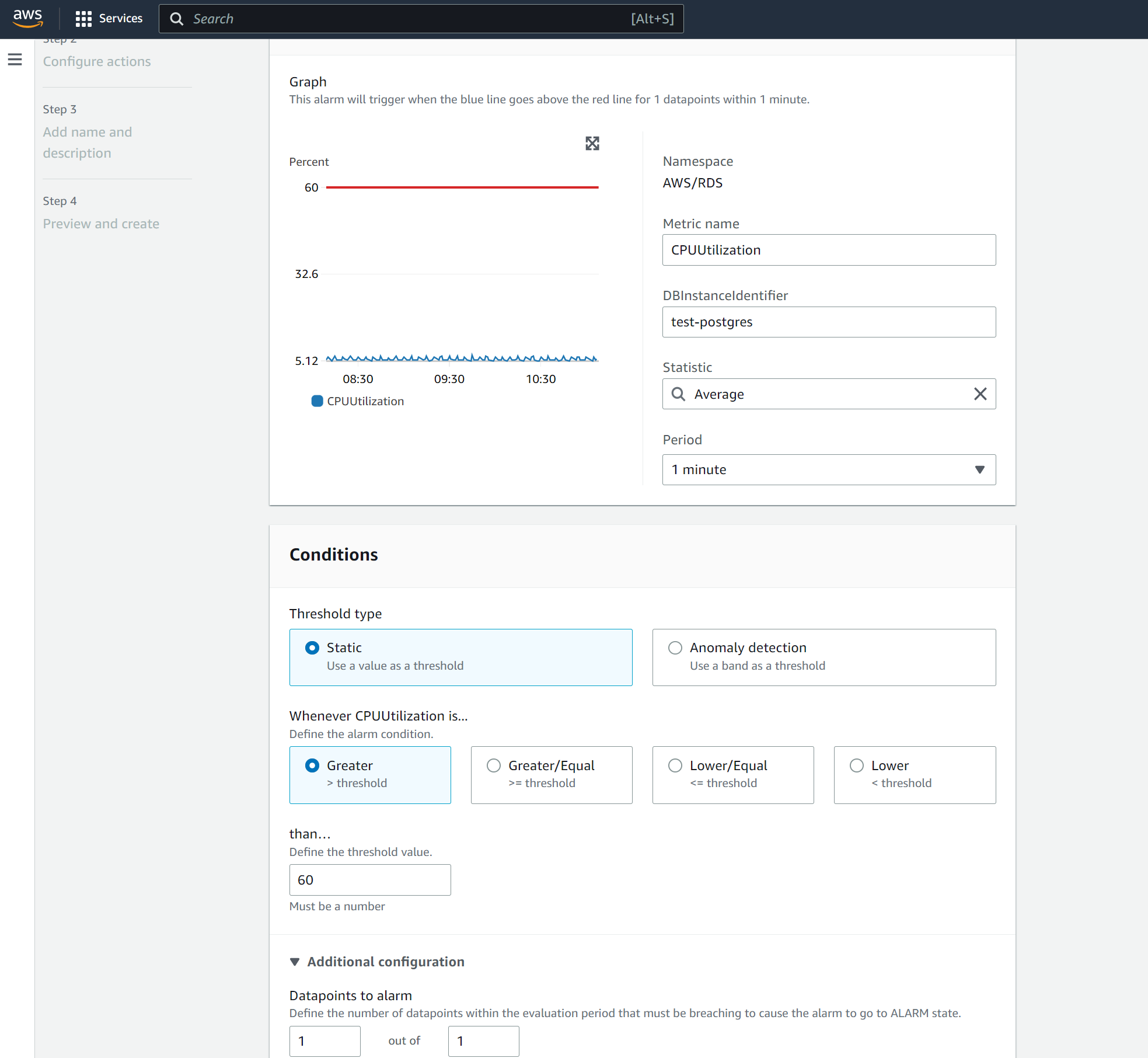
Create a trigger for monitoring the RDS server failure via cloudwatch

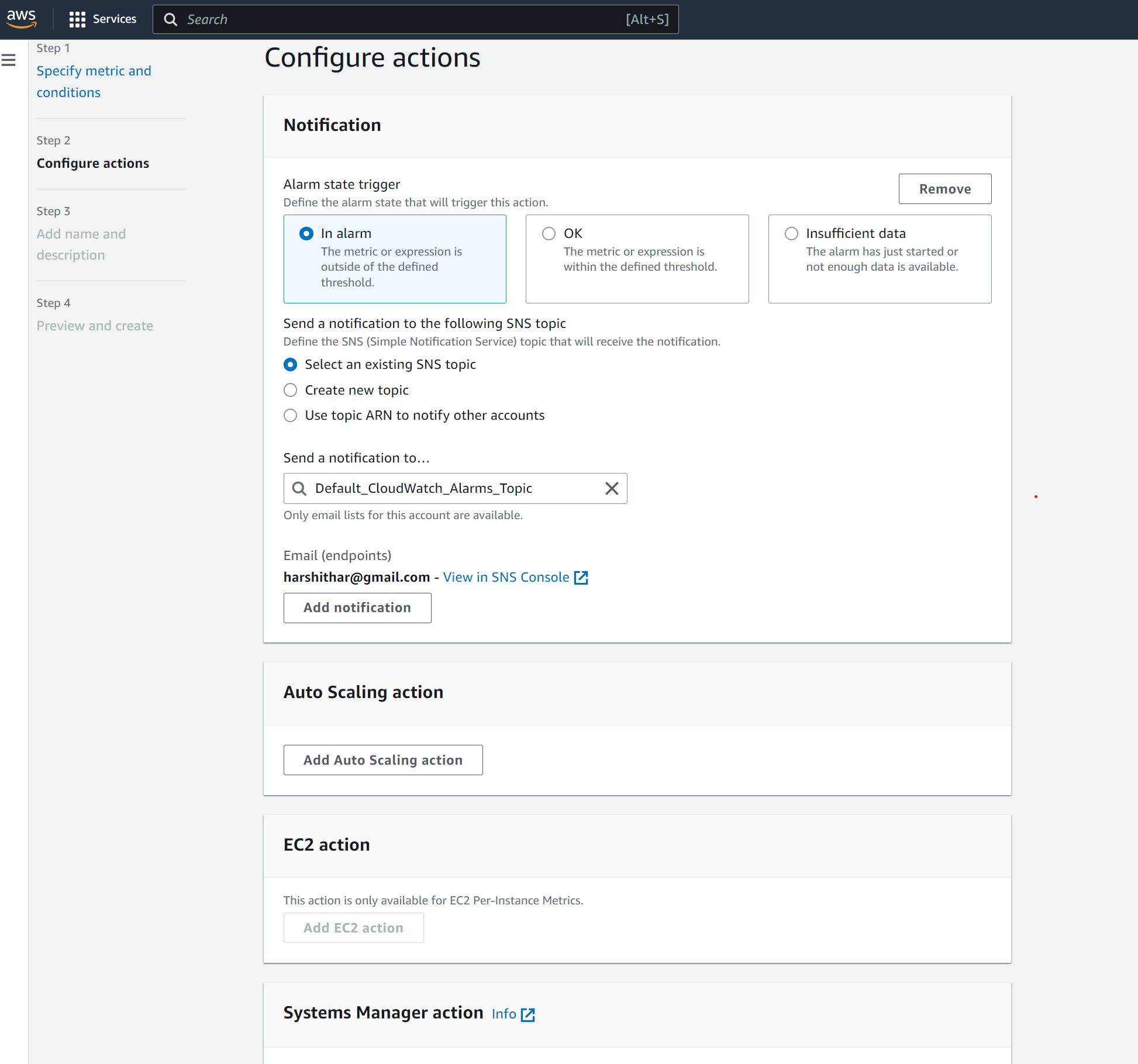
1.Create a cloudwatch alarm setup

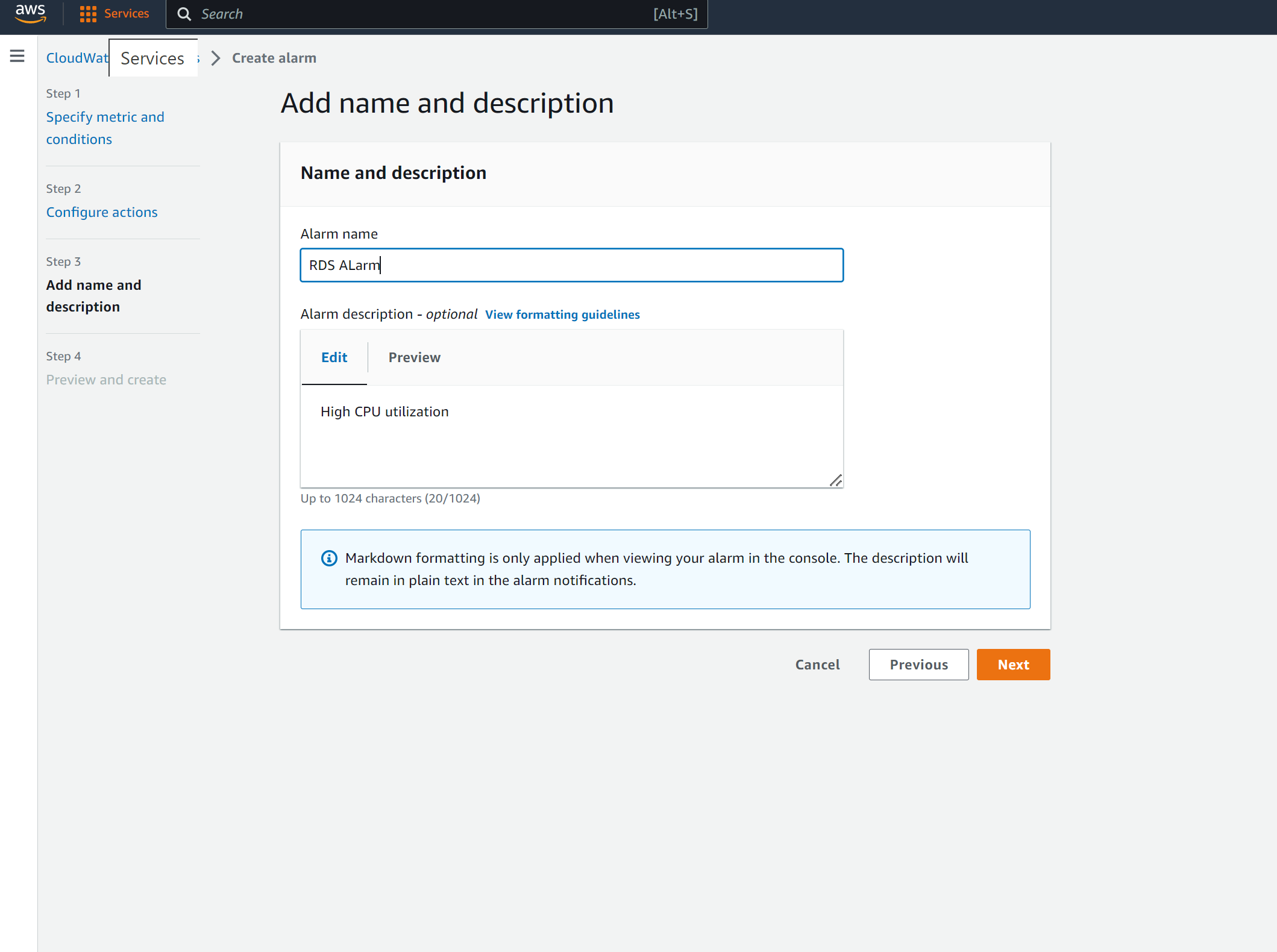












**Create a Slack channel to integrate the alarms**

