CLOUD WATCH - AWS

CLOUD WATCH:

- It is used for various purposes like MONITORING, ALARMS, DASHBOARDS, LOGS AND SCALING
- It is used to monitor various AWS services.
- It allows us to record metrics for aws services like EC2, EBS, ELB AND Amazon S3.
- We can setup alarms for our EC2 Instances

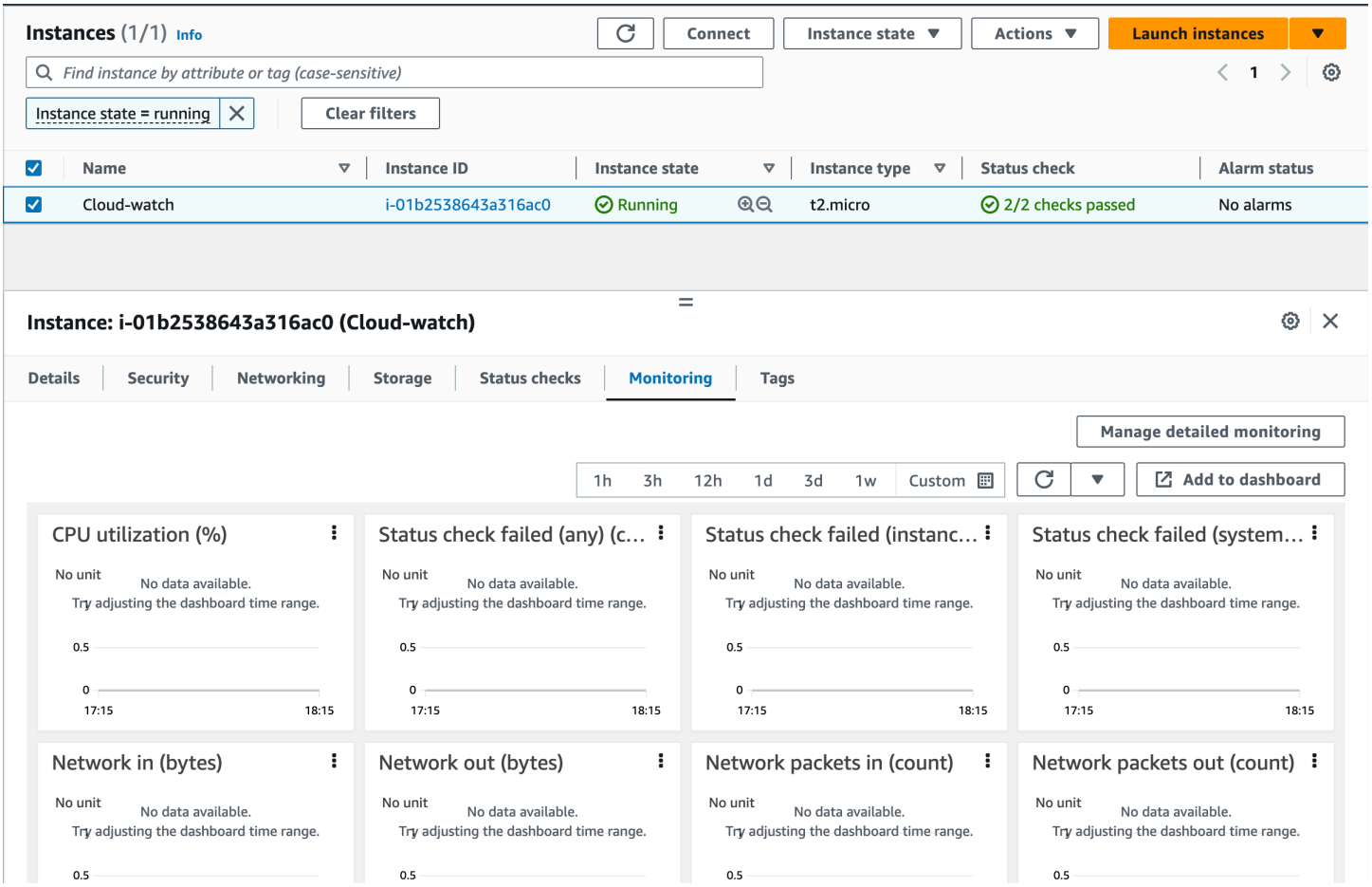
PROJECT-1: CREATE ALARM FOR AN EC2-INSTANCE

HOW IT WORKS:

- Once we launch an instance if the CPU utilisation of instance is above 80% then alarm will be triggered
- Once alarm is triggered you will get notified by SNS service

HOW TO SETUP:

STEP-1: LAUNCH AN EC2-INSTANCE

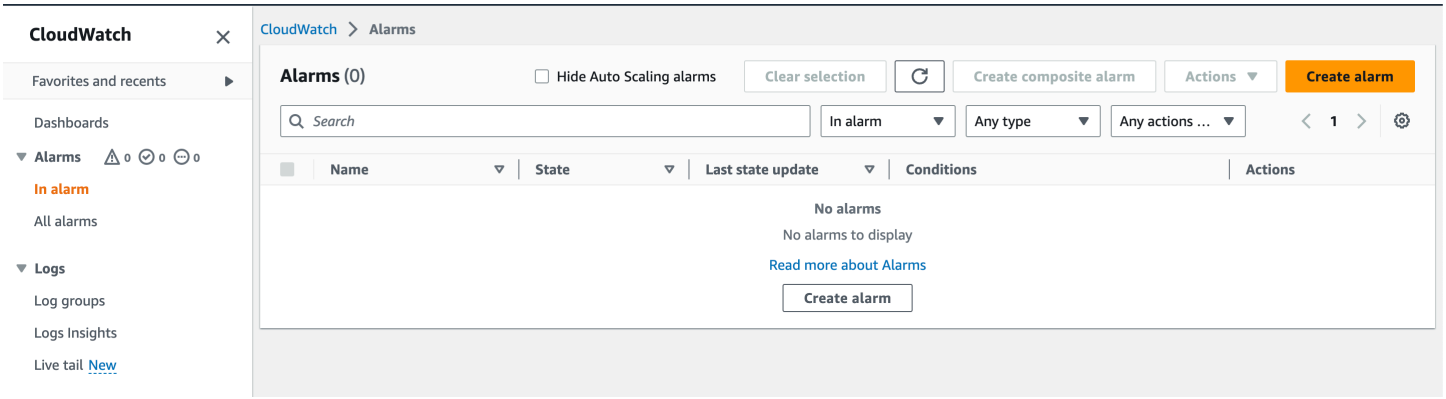


After launching the instance, check in monitoring tab no data is available. because we just created this instance

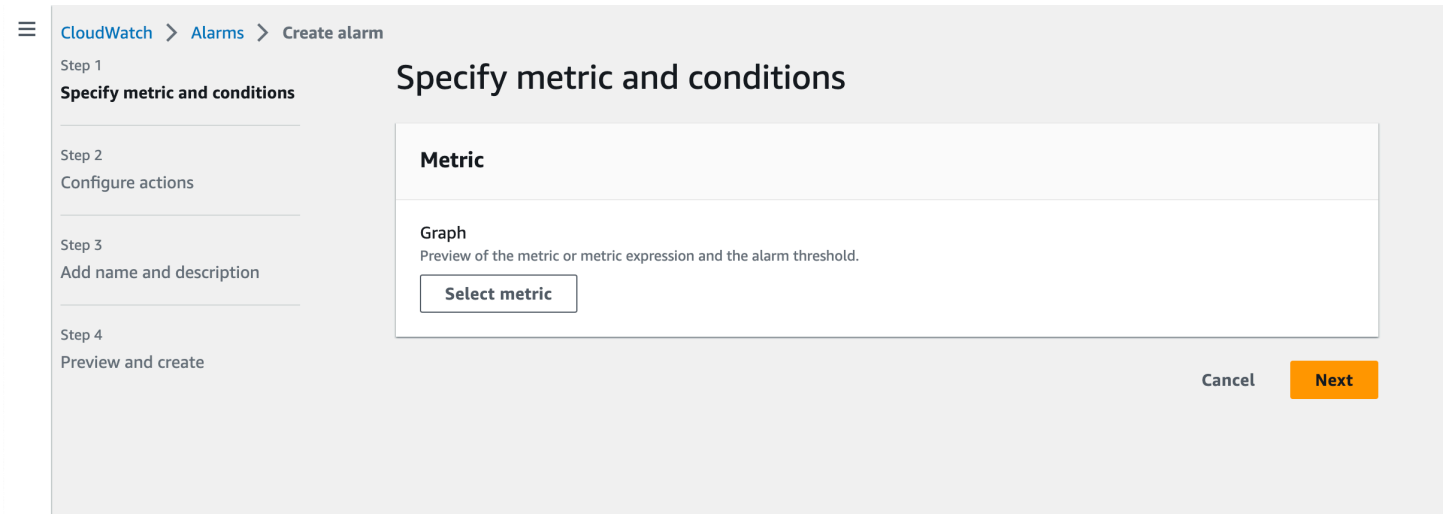
HERE OUR ULTIMATE AIM IS TO MONITOR THE INSTANCE, WHEN CPU UTILISATION IS MORE THAN 50% THEN WE HAVE TO GET A MAIL. SO WE CAN PERFORM THE ACTION AS PER THE REQUIREMENT.

STEP-2: OPEN CLOUD WATCH AND SET AN ALARM

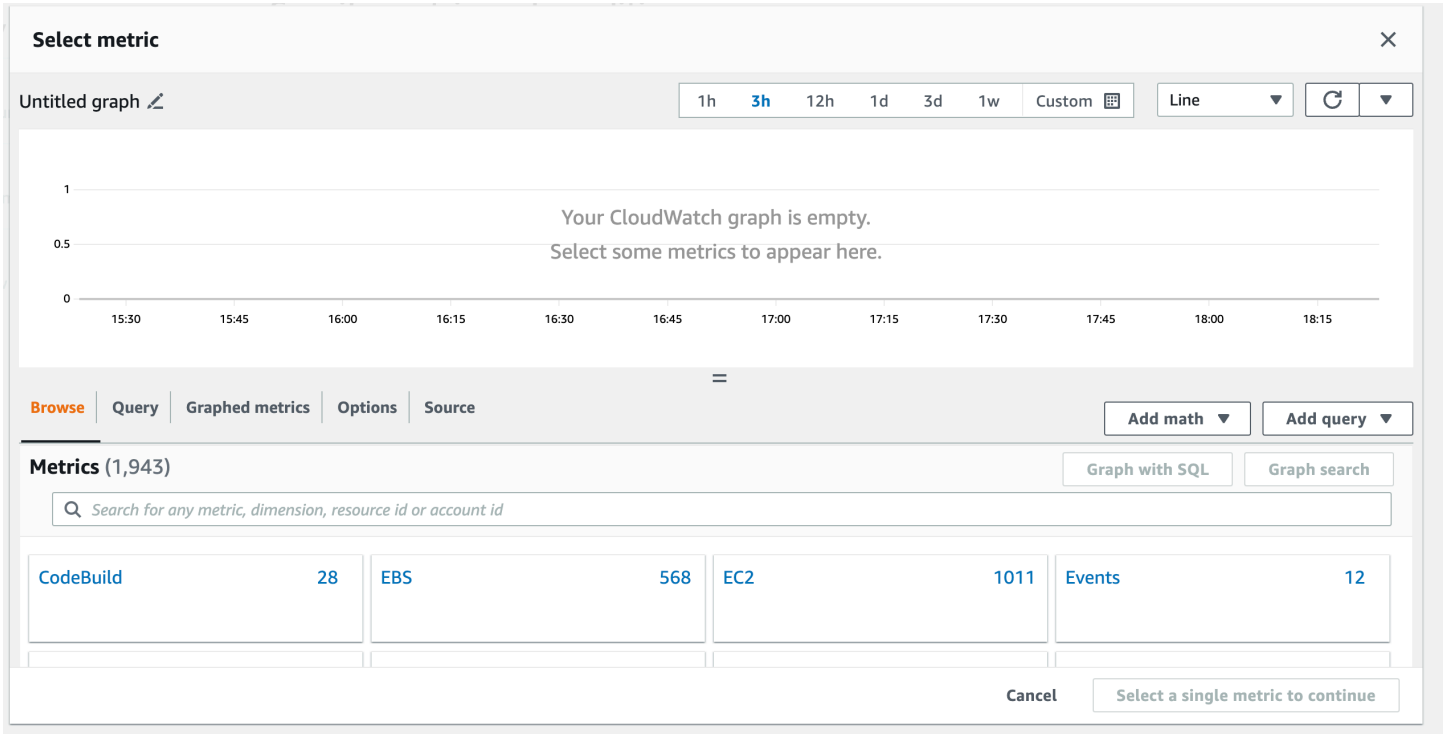
Open cloud watch service in console and select alarm



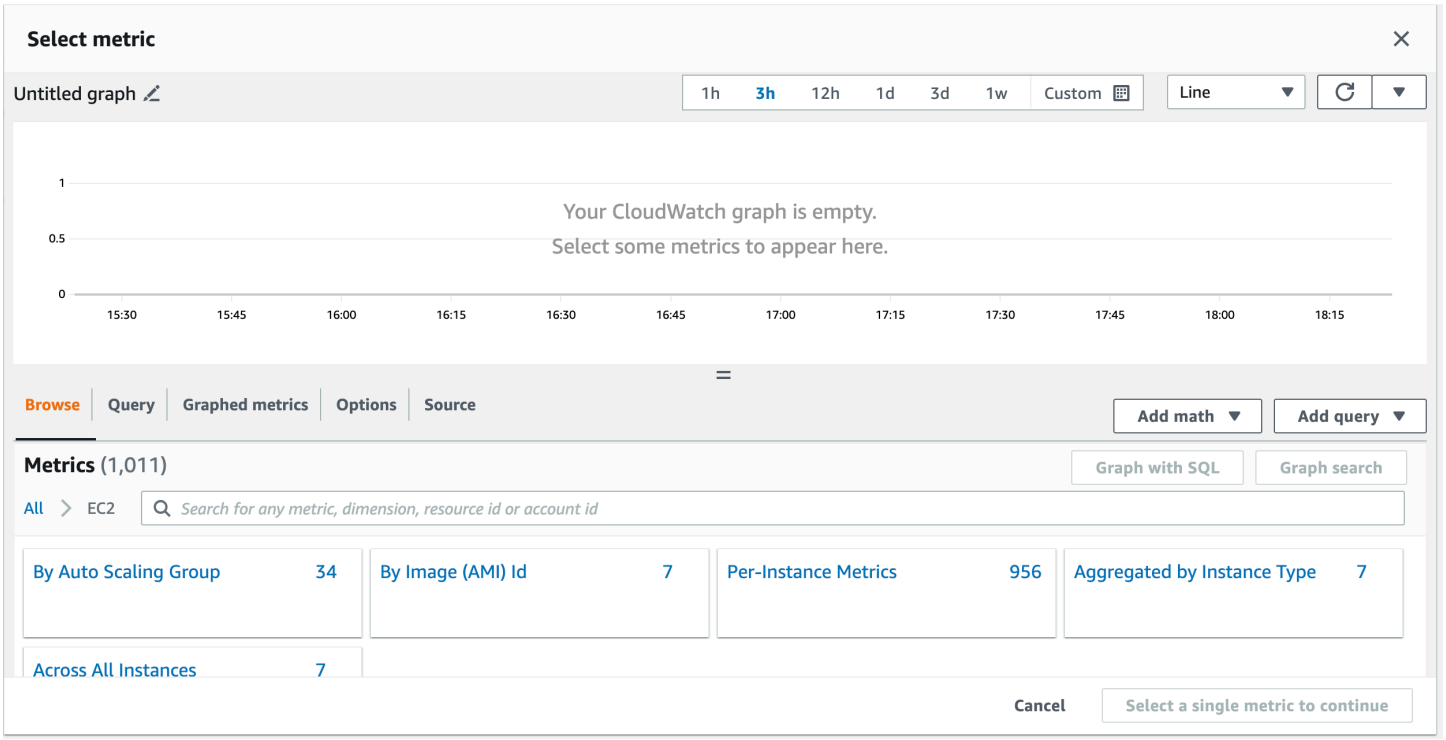
Click on create alarm



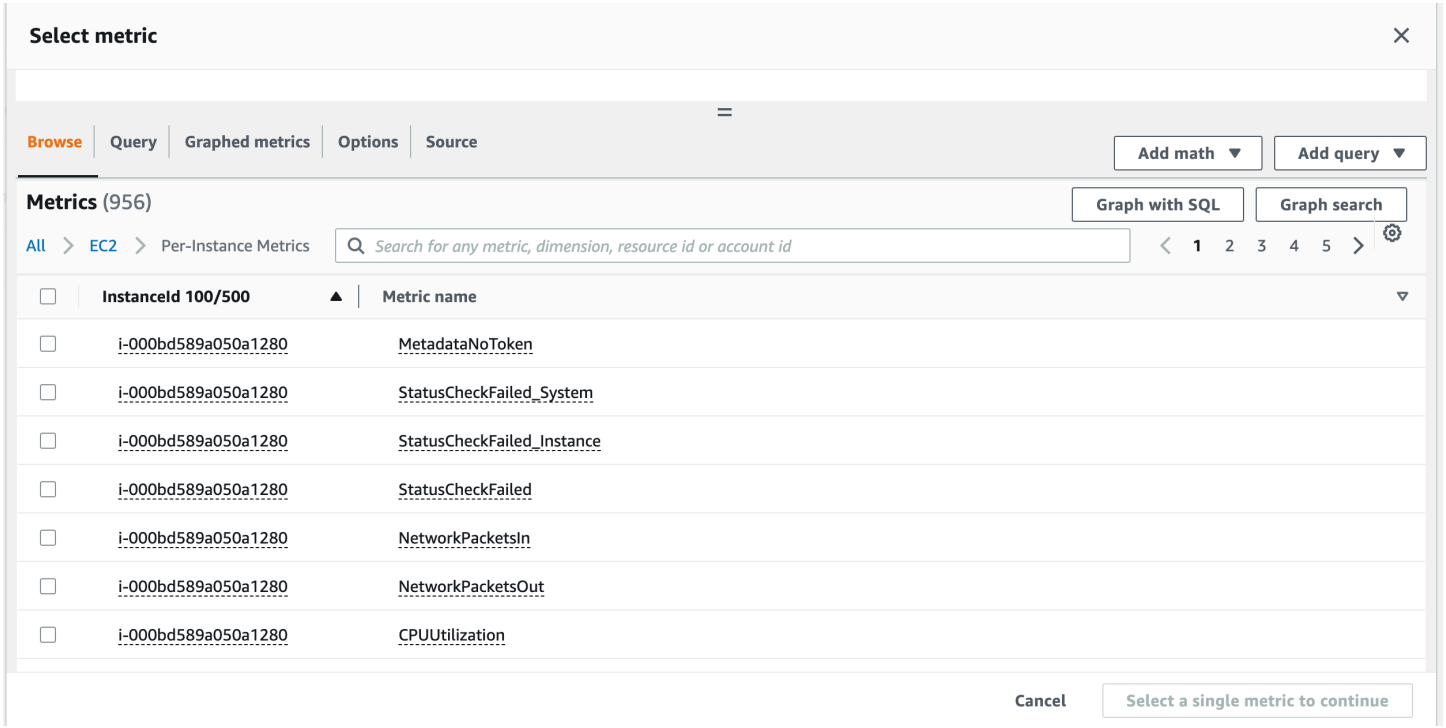
click on select metrics



select EC2

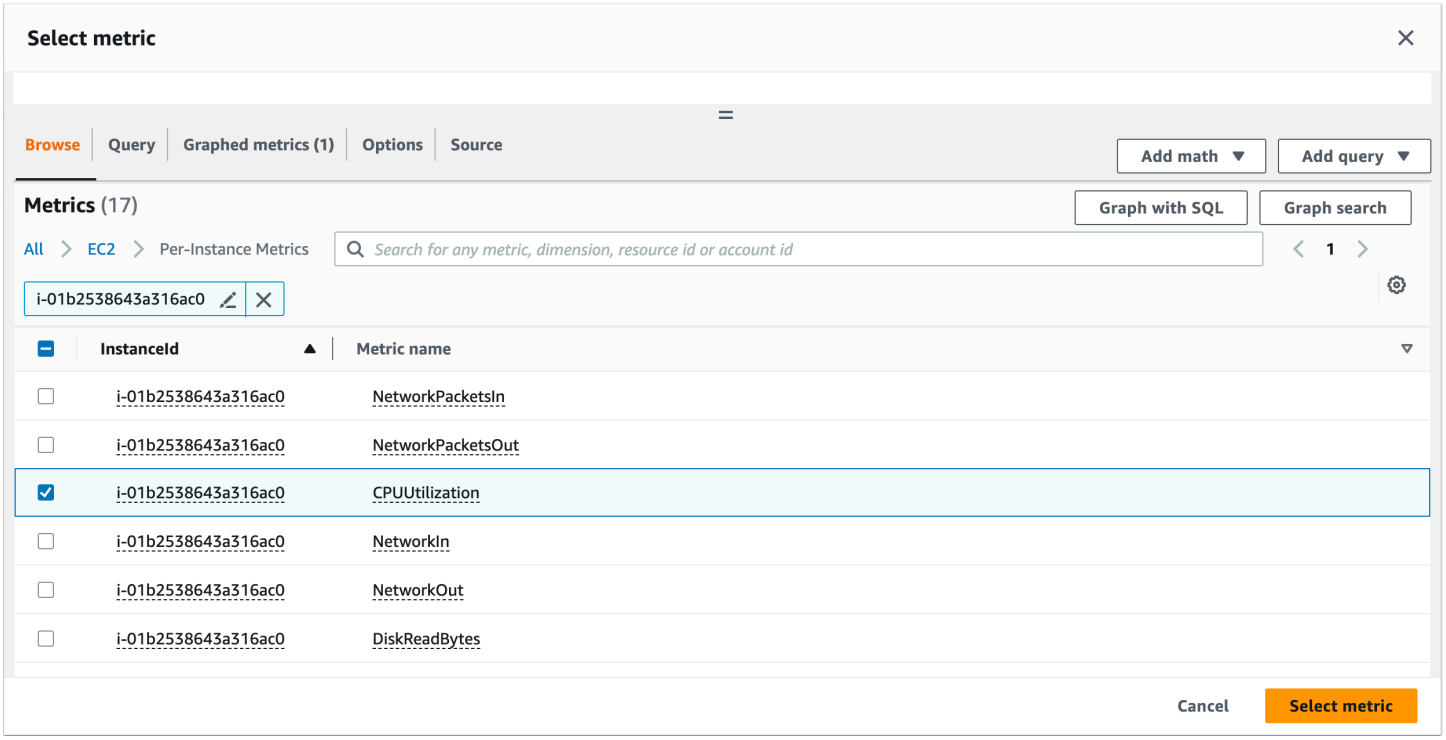


select Pre-Instance Metrics and then you will get a lot of instances like this

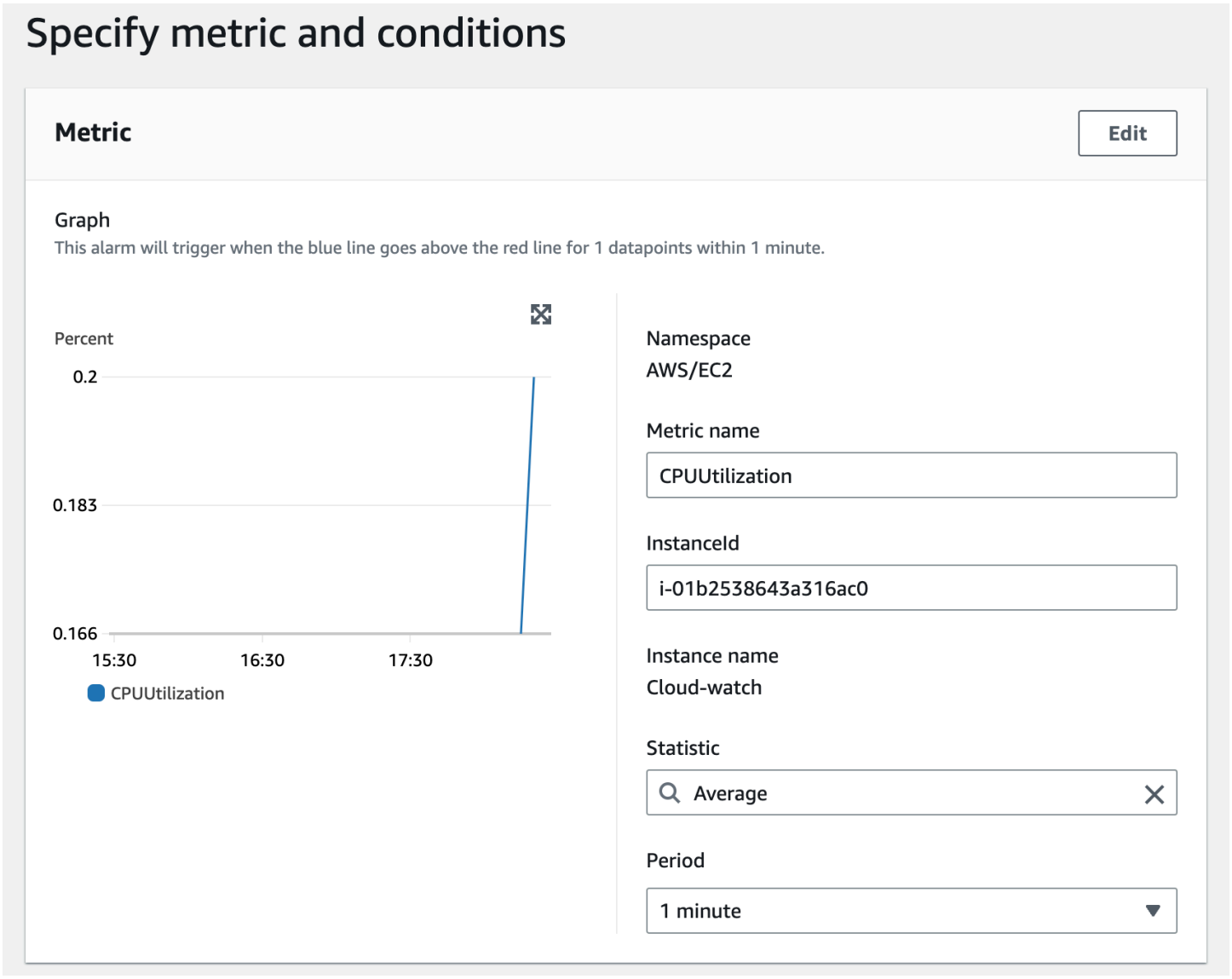


Here we have to select a single metrics for our server,

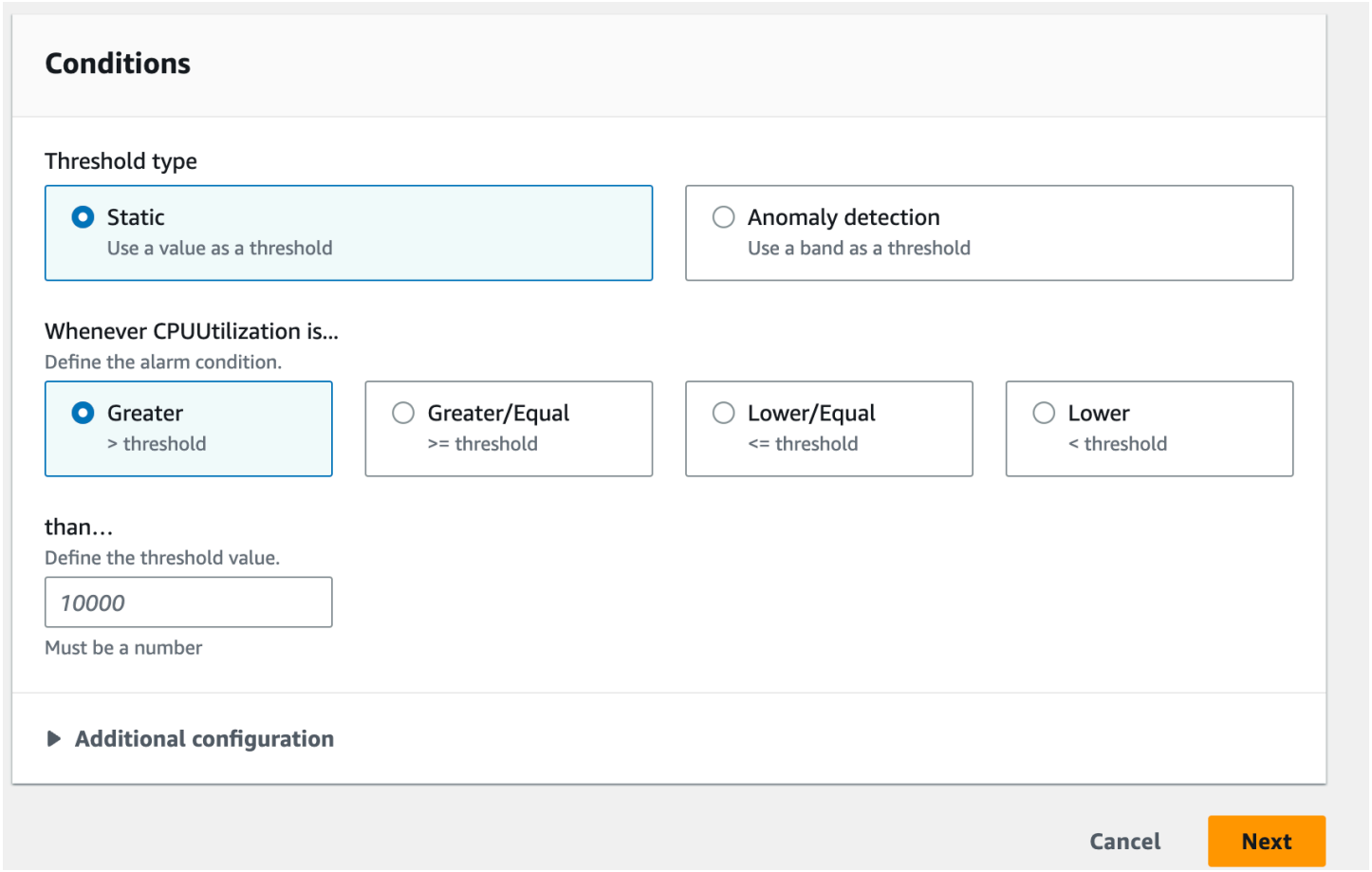
Search our instance with instance-id



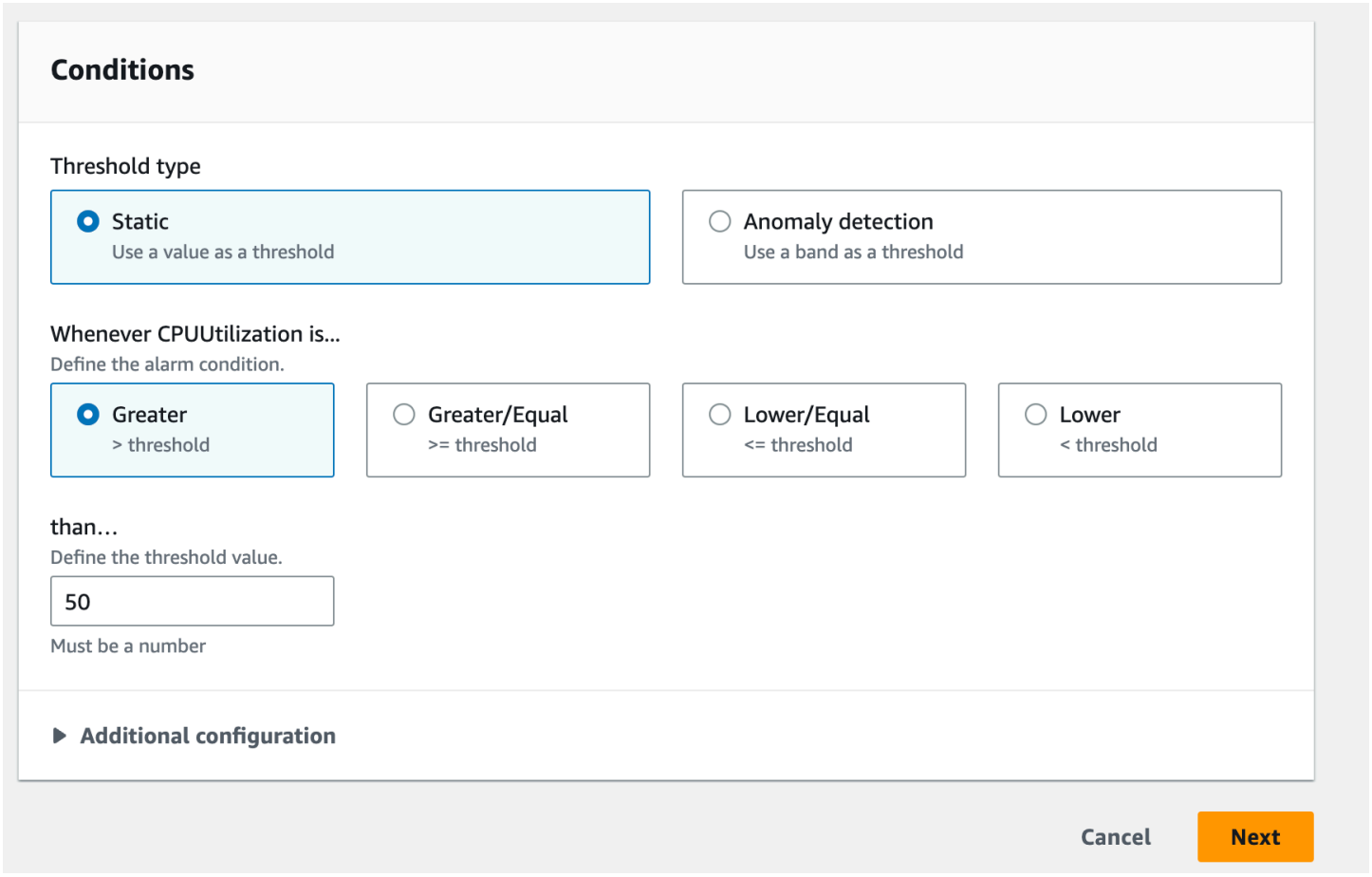
select our instance id with the CPUUtilization and click on select metrics



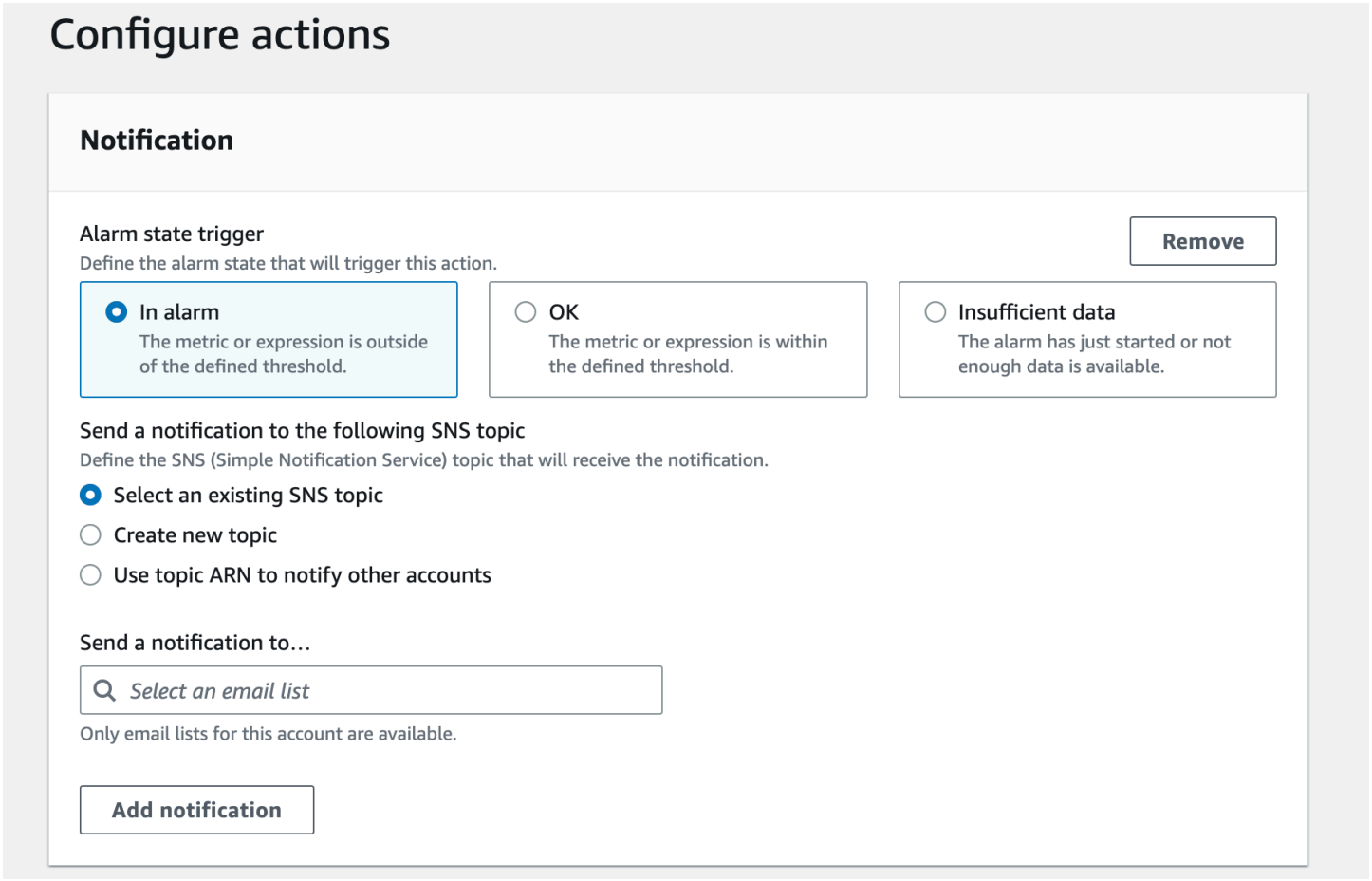
In this section we have to select the period, by default it will be 5 minutes but i have changes to 1 minute. And continue the second section which is conditions.



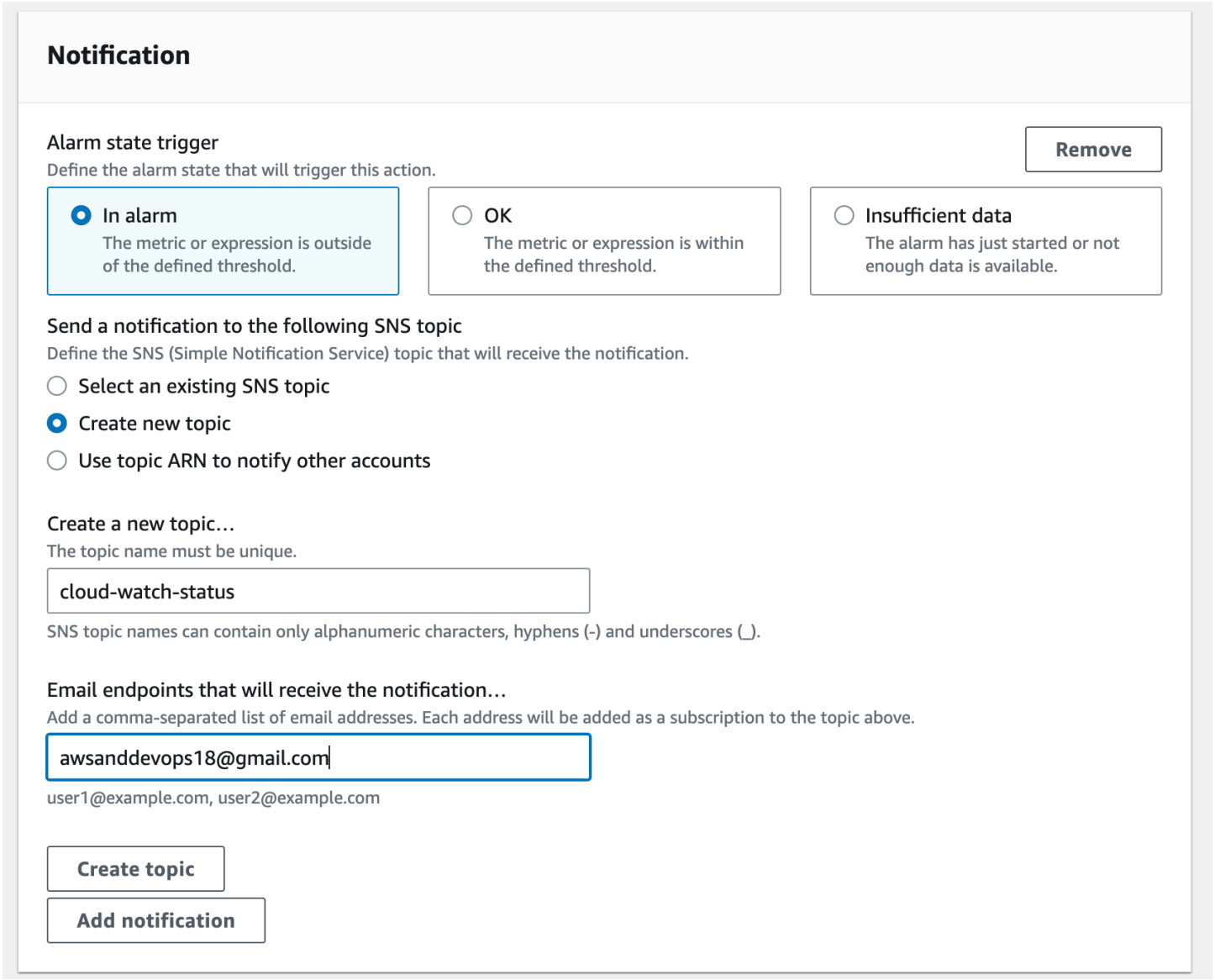
here we have to specify the CPU Utilization, i preferred to take 50%



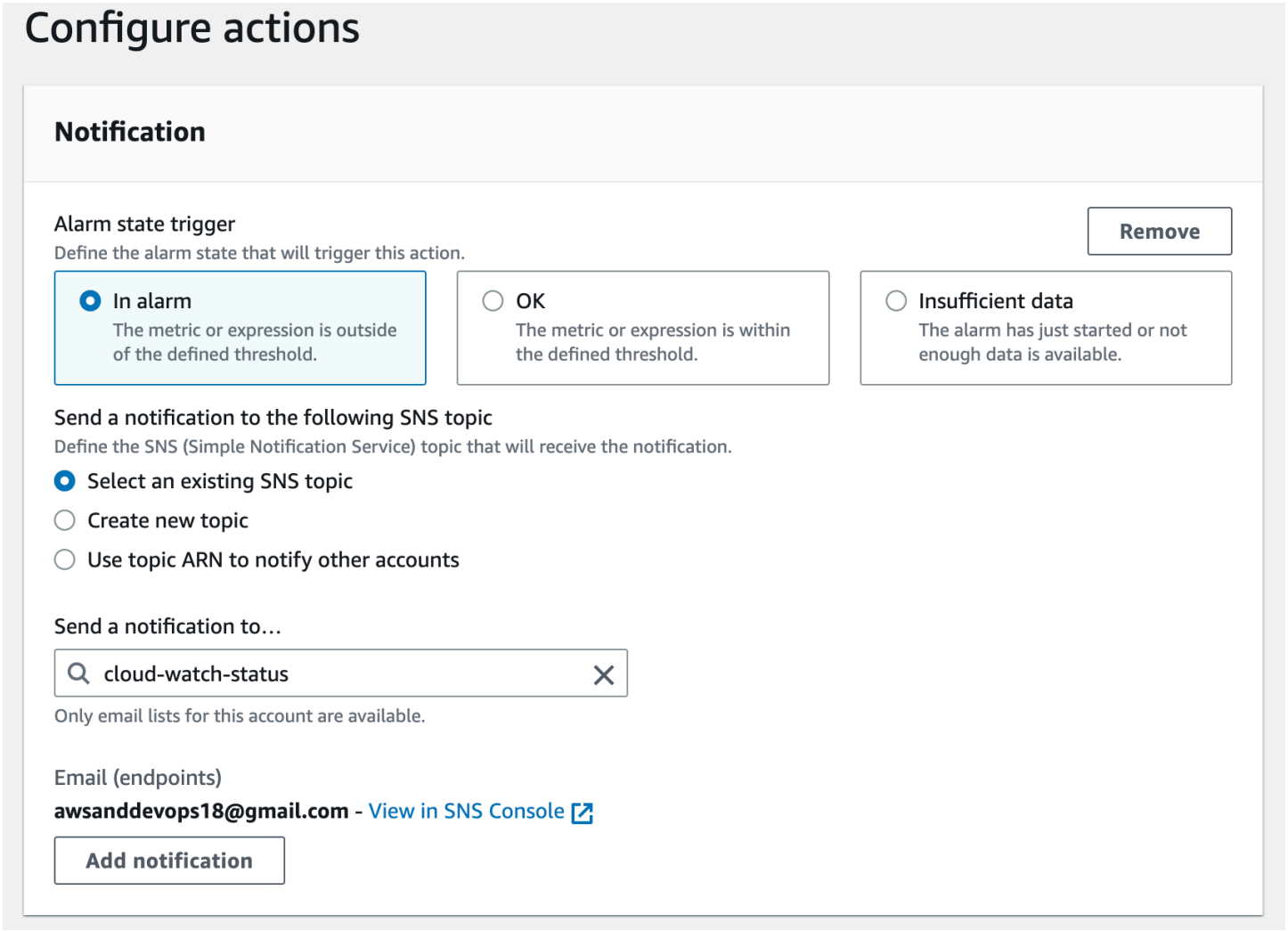
click on Next



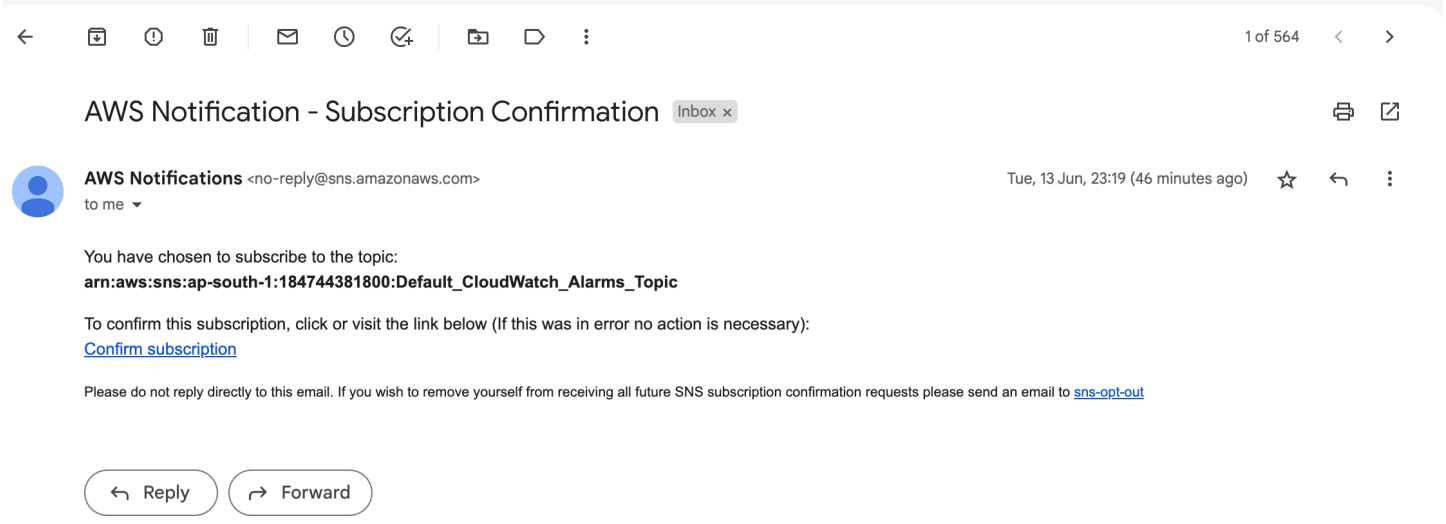
In this step we have to integrate SNS to get notified through our GMAIL. If the SNS topic is already created then you can select but in my case i haven't created so i am creating here.



i have entered the topic name and my email id and click on create topic



This will send a notification to our mail, we have to confirm the subscription on our mail.



click on confirm subscription and you will get like this

Simple Notification Service

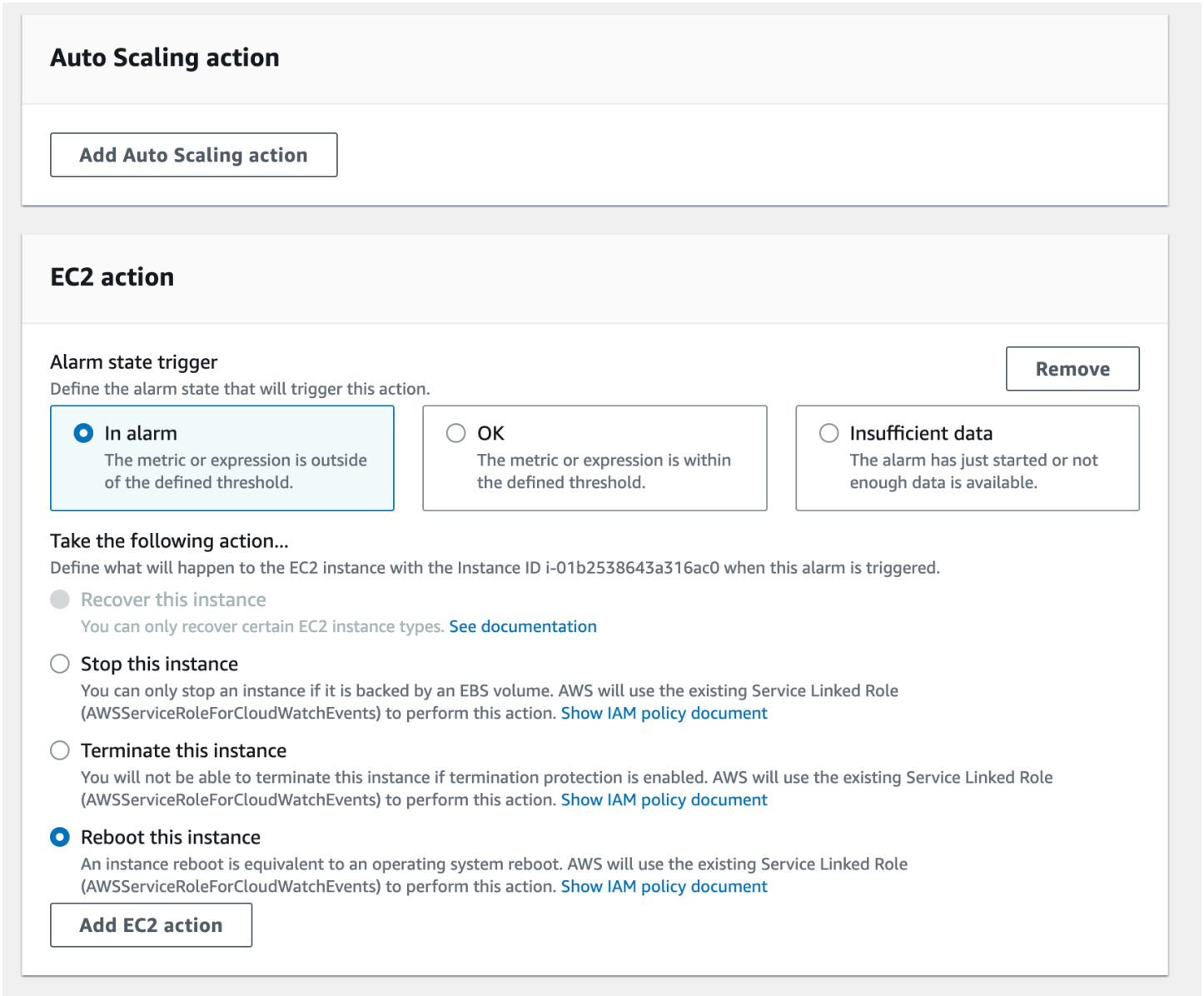
Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
`arn:aws:sns:ap-south-1:184744381800:Default_CloudWatch_Alarms_Topic:6c27ee2a-da0d-4e80-925a-7be3f435bbc4`

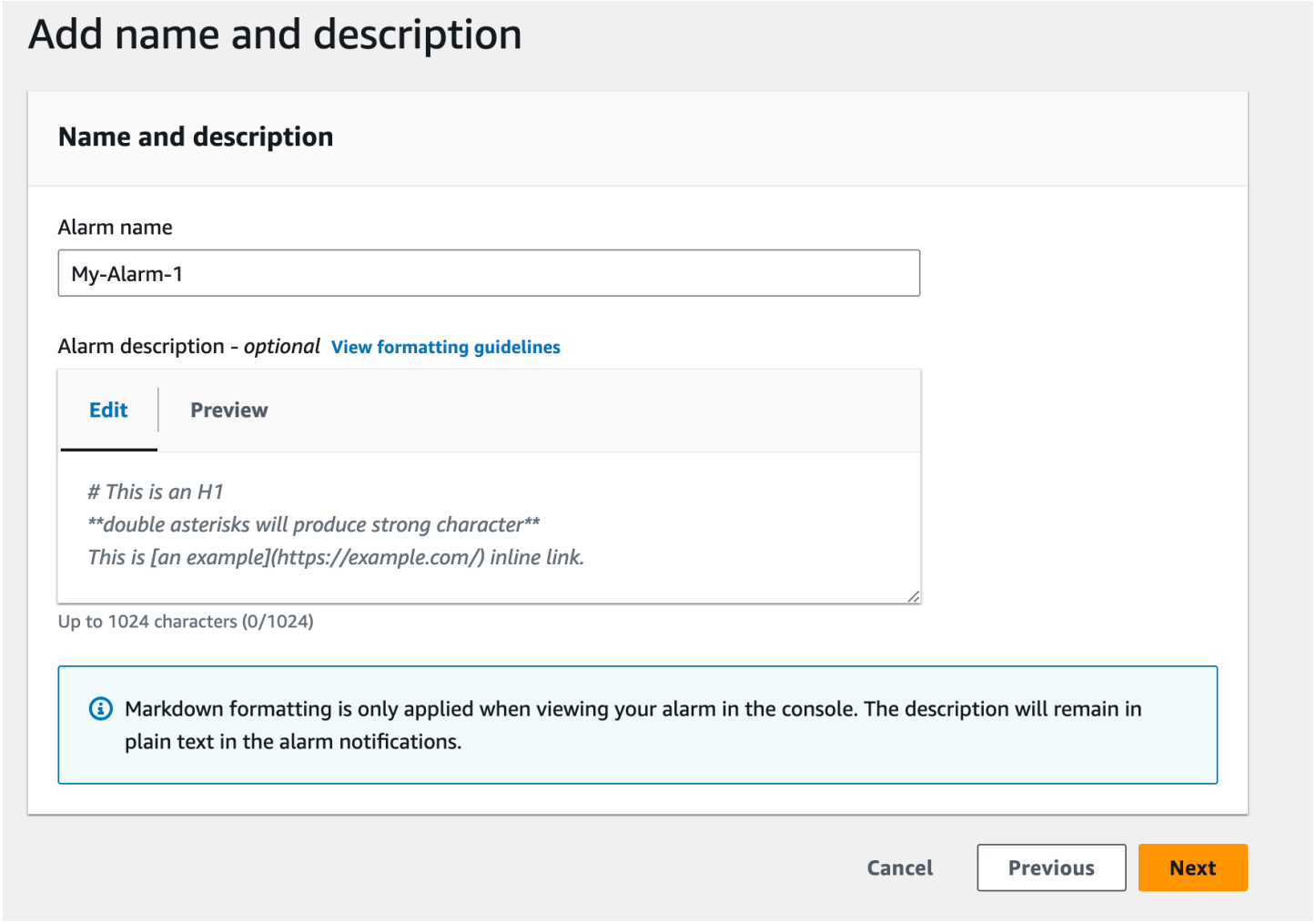
If it was not your intention to subscribe, click here to unsubscribe.

After that go back to cloud watch,



In this EC2 action section select Reboot the instance, because whenever the cpu utilization is more than 50% the our server has to be reboot, thats the best practice.

click on next



and set an alarm name and click on next

review all the steps and click on create alarm.

Till now we configured our instance to cloud watch.

STEP-3: INCREASE THE CPU UTILIZATION MORE THAN 50%

install stress in our instance by following the commands:

- amazon-linux-extras install epel -y

- yum install stress -y

take the duplicate session of your server,

one is to monitor the cpu utilization

another one is to increase the cpu utilization

in session one use **top** command to get cpu utilization

```
top - 18:44:19 up 29 min,  2 users,  load average: 0.16, 0.10, 0.04
Tasks:  99 total,   1 running,  55 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   975600 total,   416140 free,    89100 used,   470360 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   746164 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0  123620   5508   3928 S   0.0   0.6   0:02.01 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-ev
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
```

By default my CPU ULITIZATION of my server is 0%

open the second session and use following command to increase cpu utilization

stress -c 40 -t 500 -v

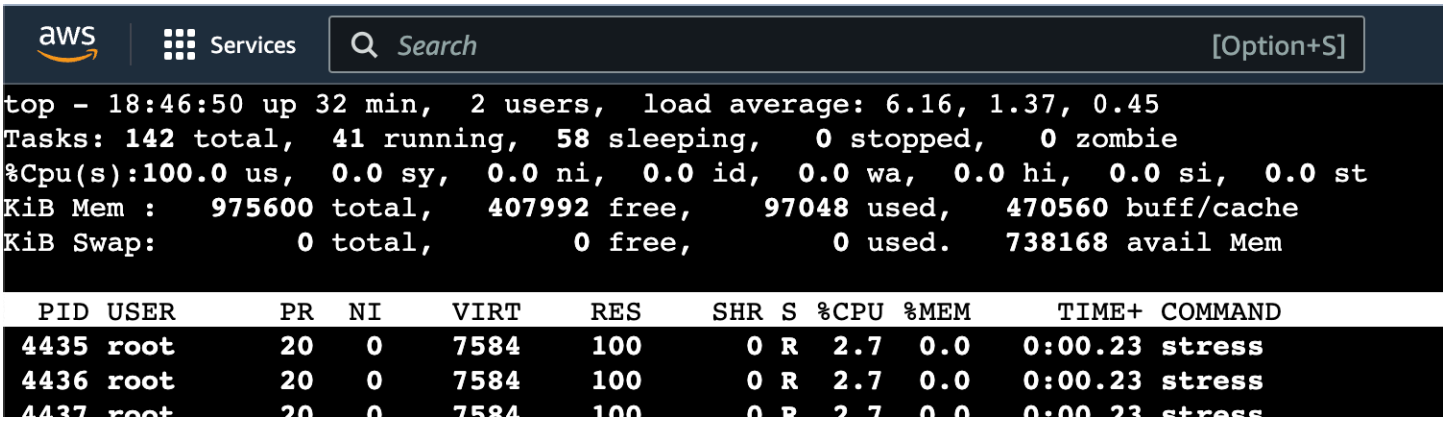
-c : cpu

-t : time

-v : verbose

by this command we are giving some load to cpu, after performing the command

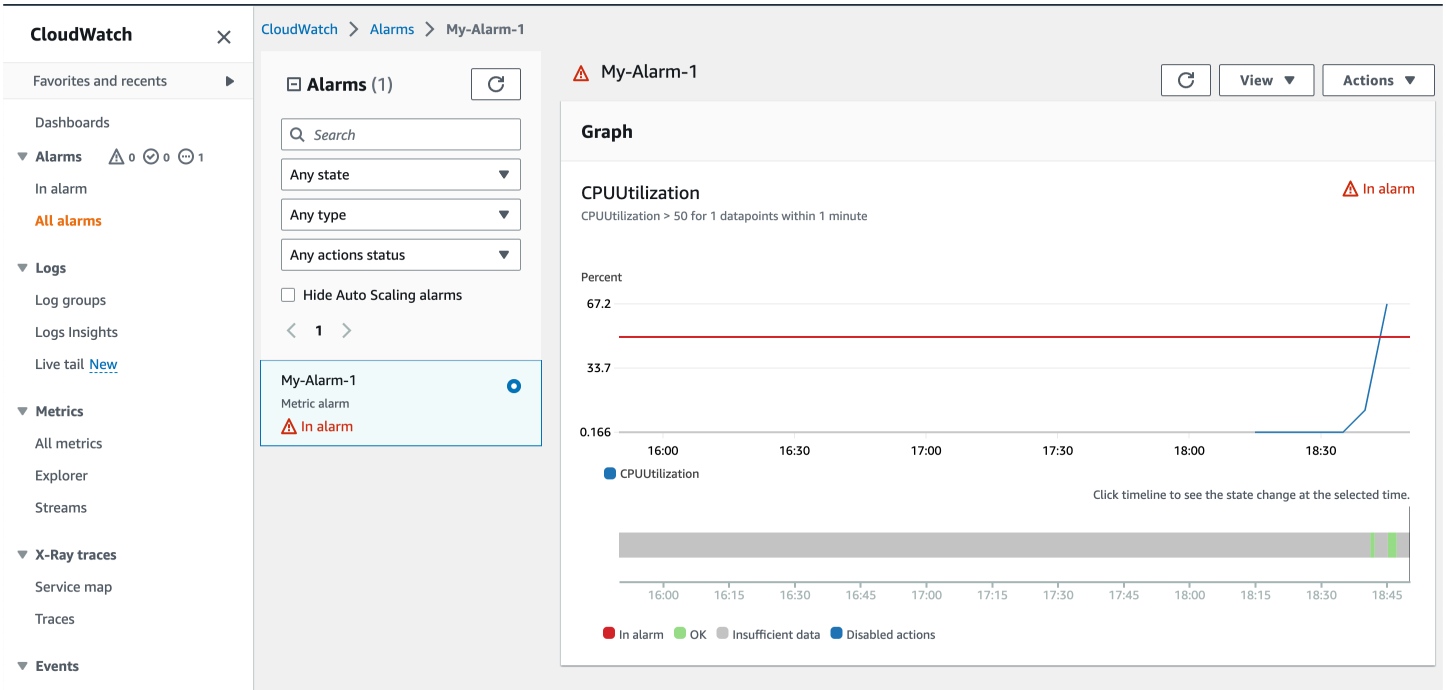
check the cpu utilization on 1st session again



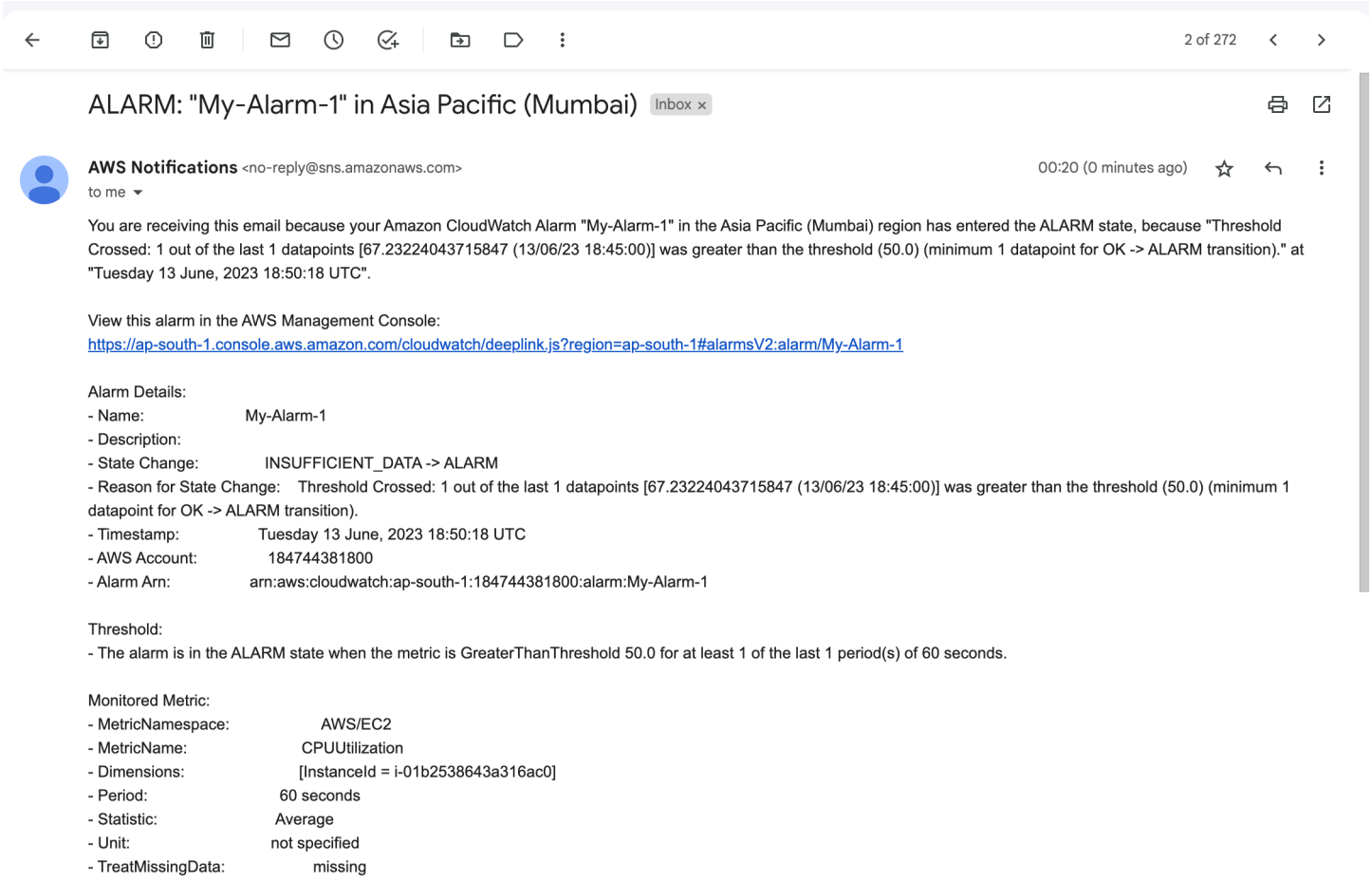
as you can observe here my cpu reached to 100%, if it stays like this for 1 minute, then you will get mail.

meanwhile you can watch this in cloud watch also, open cloud watch and open your alarm

After waiting few minutes, by graph reached to 62% above in cloud watch



So finally i get a mail from CLOUD WATCH SERVICE like this



2 of 272

ALARM: "My-Alarm-1" in Asia Pacific (Mumbai) Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 00:20 (0 minutes ago)
to me

You are receiving this email because your Amazon CloudWatch Alarm "My-Alarm-1" in the Asia Pacific (Mumbai) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [67.23224043715847 (13/06/23 18:45:00)] was greater than the threshold (50.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 13 June, 2023 18:50:18 UTC".

View this alarm in the AWS Management Console:
https://ap-south-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-south-1#alarmsV2:alarm/My-Alarm-1

Alarm Details:
- Name: My-Alarm-1
- Description:
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [67.23224043715847 (13/06/23 18:45:00)] was greater than the threshold (50.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 13 June, 2023 18:50:18 UTC
- AWS Account: 184744381800
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:184744381800:alarm:My-Alarm-1

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 50.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [InstanceId = i-01b2538643a316ac0]
- Period: 60 seconds
- Statistic: Average
- Unit: not specified
- TreatMissingData: missing

PROJECT-2: UPLOAD LOG FILES IN CLOUD WATCH

REFERENCE
(https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/QuickStartEC2Instance.html)

STEP-1: LAUNCH EC2 INSTANCE (Ubuntu)

STEP-2: INSTALL CLOUD LOGS

DOWNLOAD FILE : curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O

INSTALL PYTHON : apt update && apt install python2.7 -y

RUN PYTHON FILE : python2.7 awslogs-agent-setup.py —region ap-south-1

Enter

acces_key : enter

secret_key : enter

default_region : enter

O/p_format : enter

Path to log file to upload : /var/log/apache2/access.log

Destination log group name : syslog_ec2

Log stream name: 1 (EC2 instance ID)

Log event time stamp format: 2

Initial position to upload a file : 1

More log files for configure : N

Check the service : systemctl status awslogs

vim /var/awslogs/etc/awslogs.conf ——> **this is the path where we can store all the log files paths**

First lets check these files are storing in cloud watch or not, if its working fine then we can start store our app log files.

CREATE IAM ROLES:

IAM —> ROLES ——> CREATE ROLE

SELECT EC2 AND ADD CLOUD WATCH PERMISSIONS

ATTACH THAT ROLE TO EC2 INSTANCE

RESTART AWS LOGS AGAIN : systemctl restart awslogs

Go to CLOUD WATCH AND SEE THE LOGS

CHANGE THE PATH TO OUR APP LOG FILES

To do that we have to install web server and deploy a web application

apt install apache2 -y

Add some files in (/var/www/html/)

AFTER DEPLOYED THE APPLICATION, WE HAVE TO CHECK THE APP LOGS IN (vim /var/log/apache2/access.log file

There you found all log info

We have to configure this path to aws cloud watch logs

TO DO THAT

Go to the path : vim /var/awslogs/etc/awslogs.conf

Go to the last line of the file and copy the data as it is (change path)

RESTART CLOUDWATCH LOGS : systemctl restart awslogs

RESTART WEBSERVER : systemctl restart apache2

RELATIONAL DATABASE SERVICE

Usually we have 2 types of databases

Relational databases : Oracle, MYSQL, PostgresSQL etc...

No-SQL databases : Mongo DB, Dynamo DB etc..

IN RD —> we store the data in table format

In early days we have to purchase the databases like

Buy database license

Setu machines to install db server

Set database server

setup network, power and AC connections

Setup security resources

Setup data backups

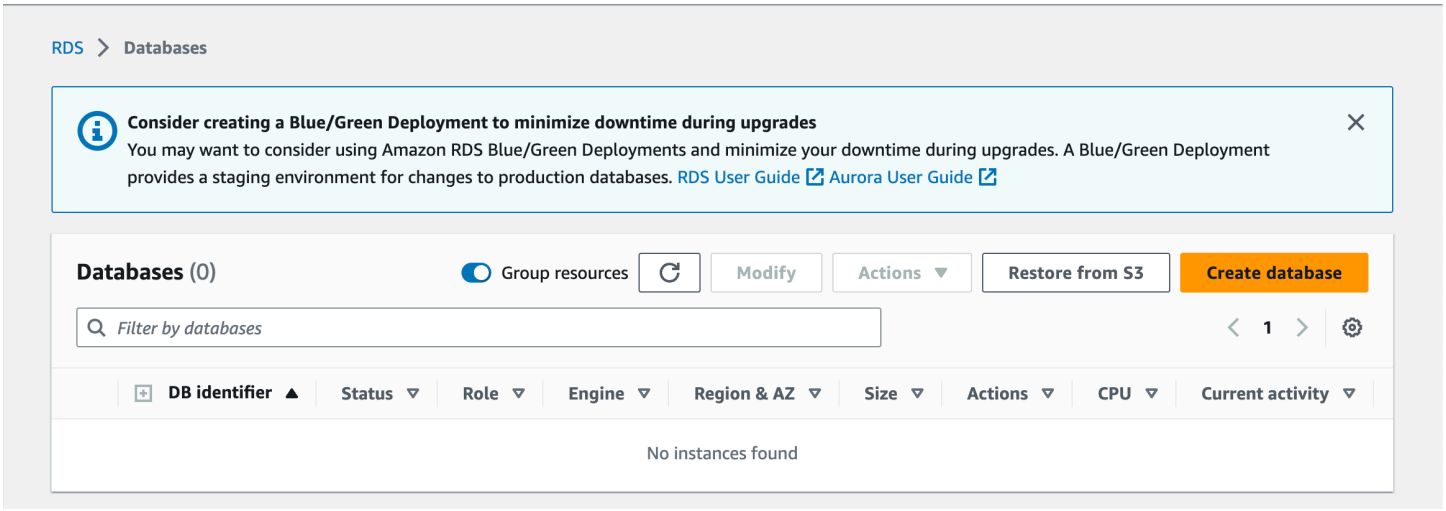
But now a days most of the companies are moving to cloud, if your choice is AWS

They will take care each and everything about these databases and maintenance

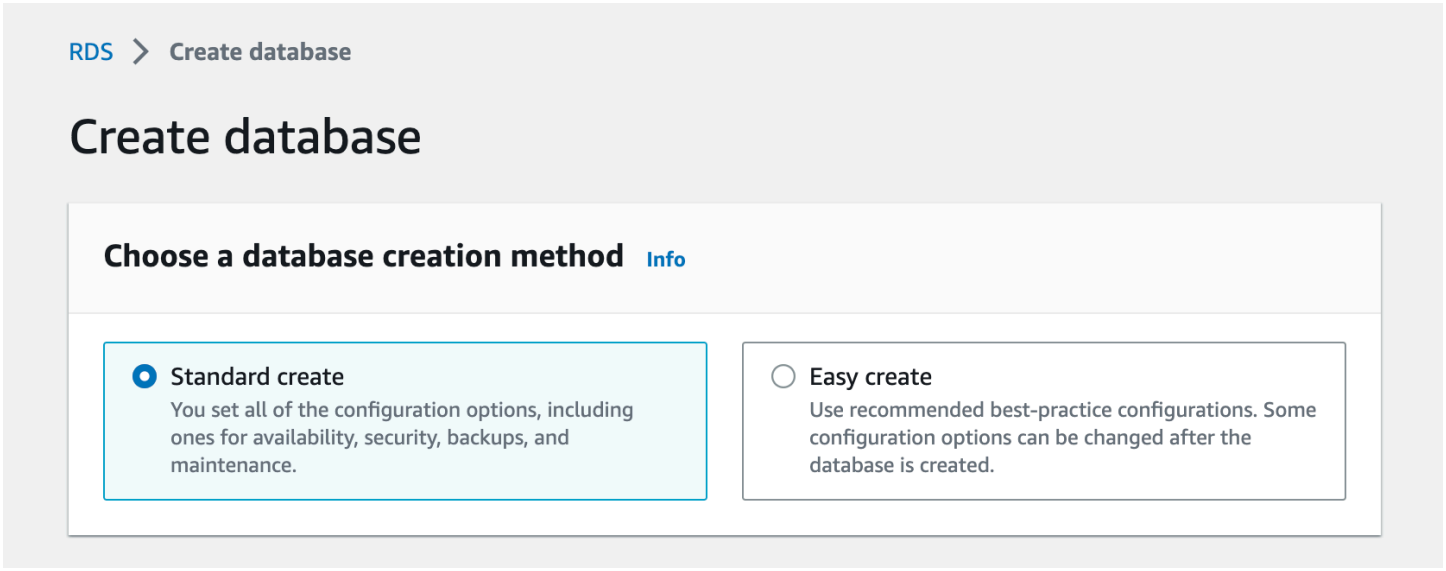
We just need to create a database and relax

HOW TO CREATE RDS :

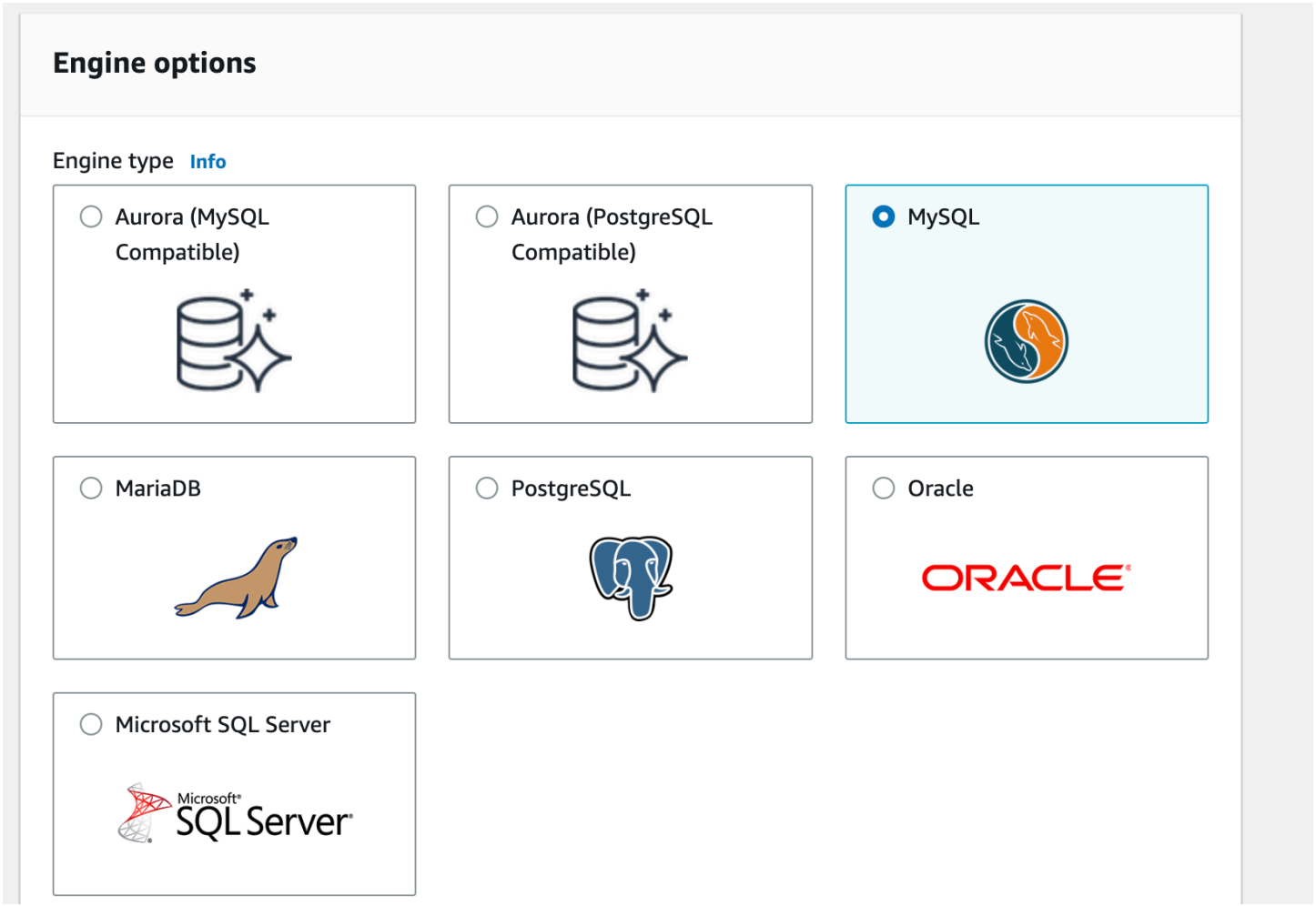
Go to RDS in AWS Console



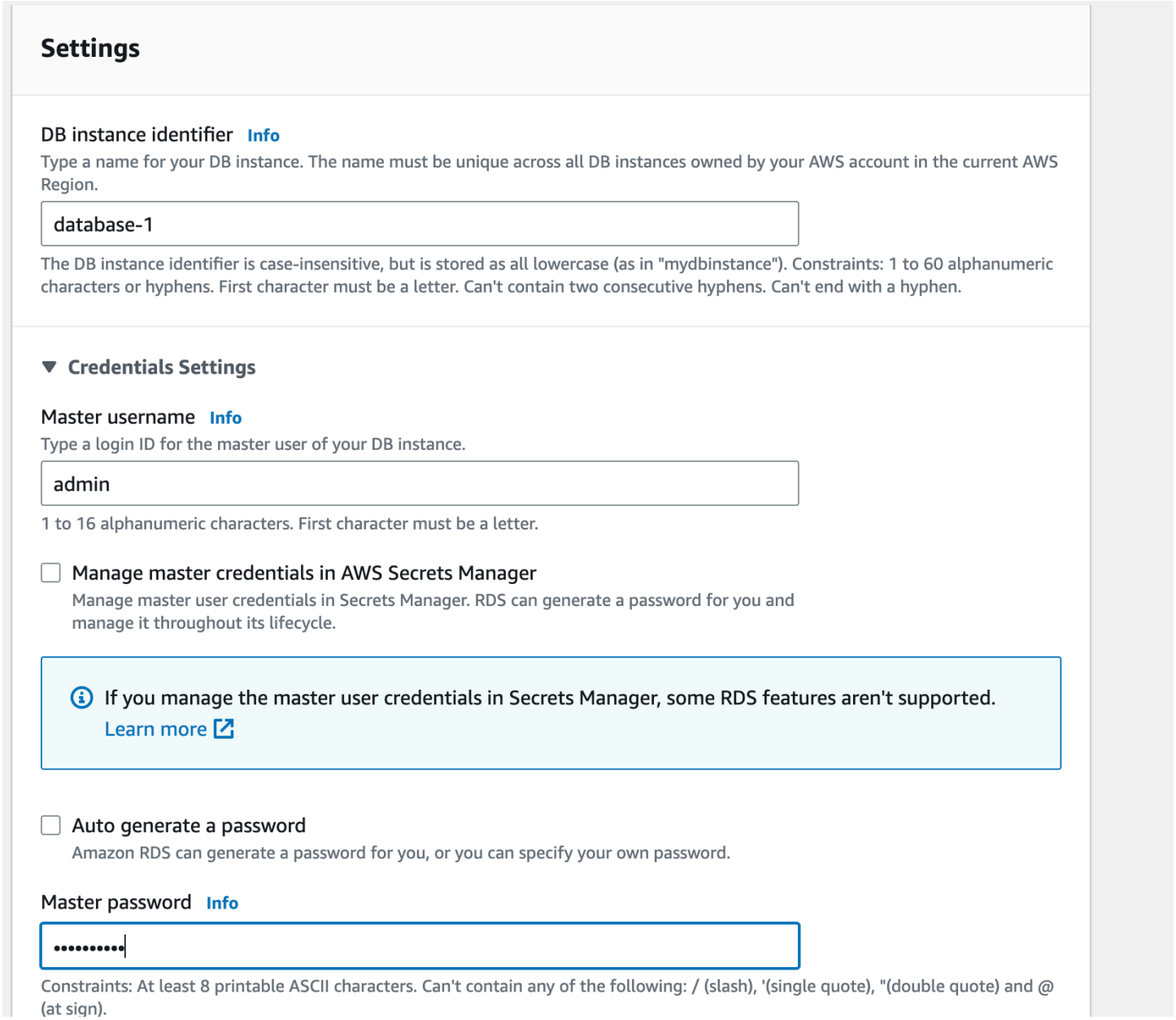
Click on Crete database



Select Standard Create



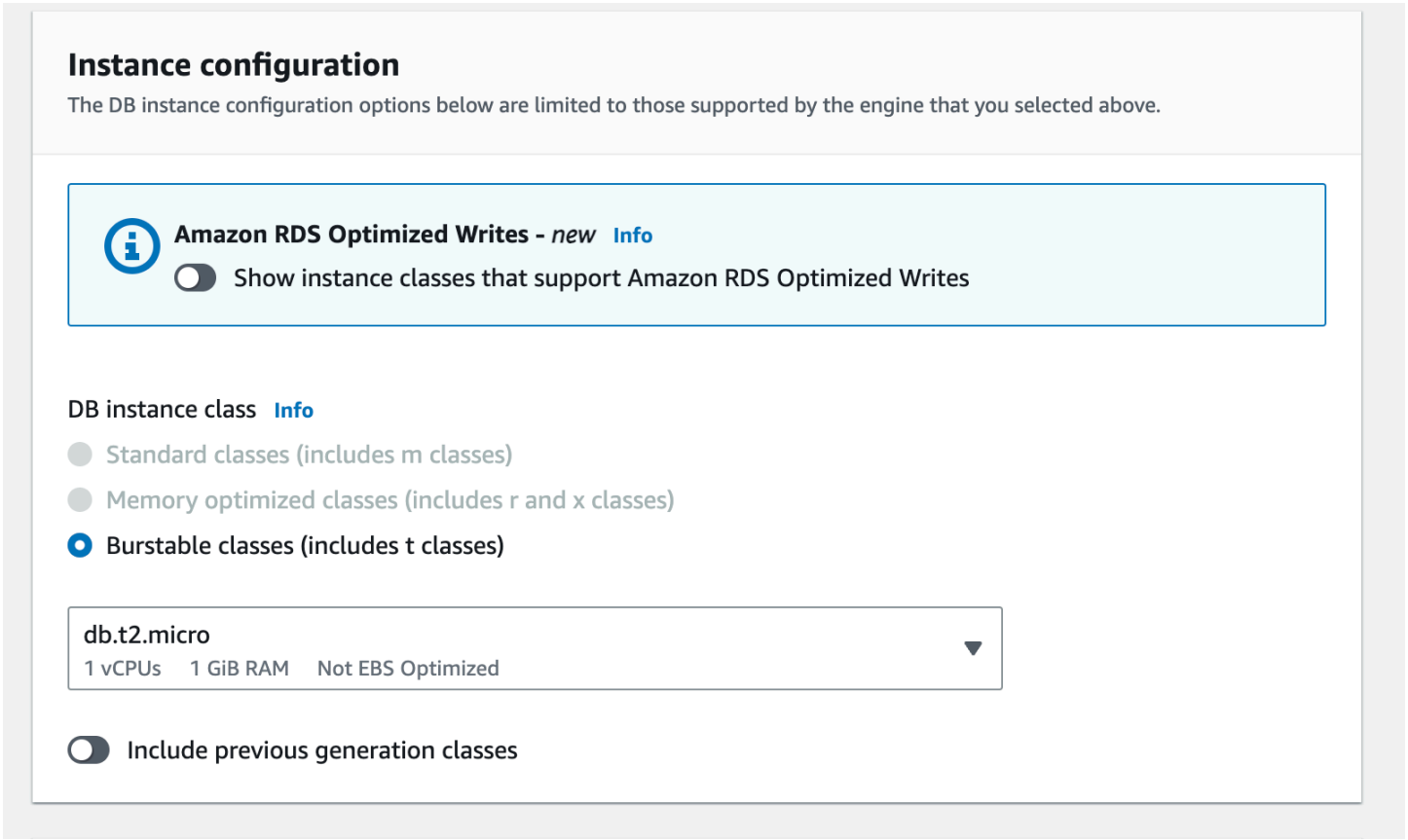
Select MySql Engine



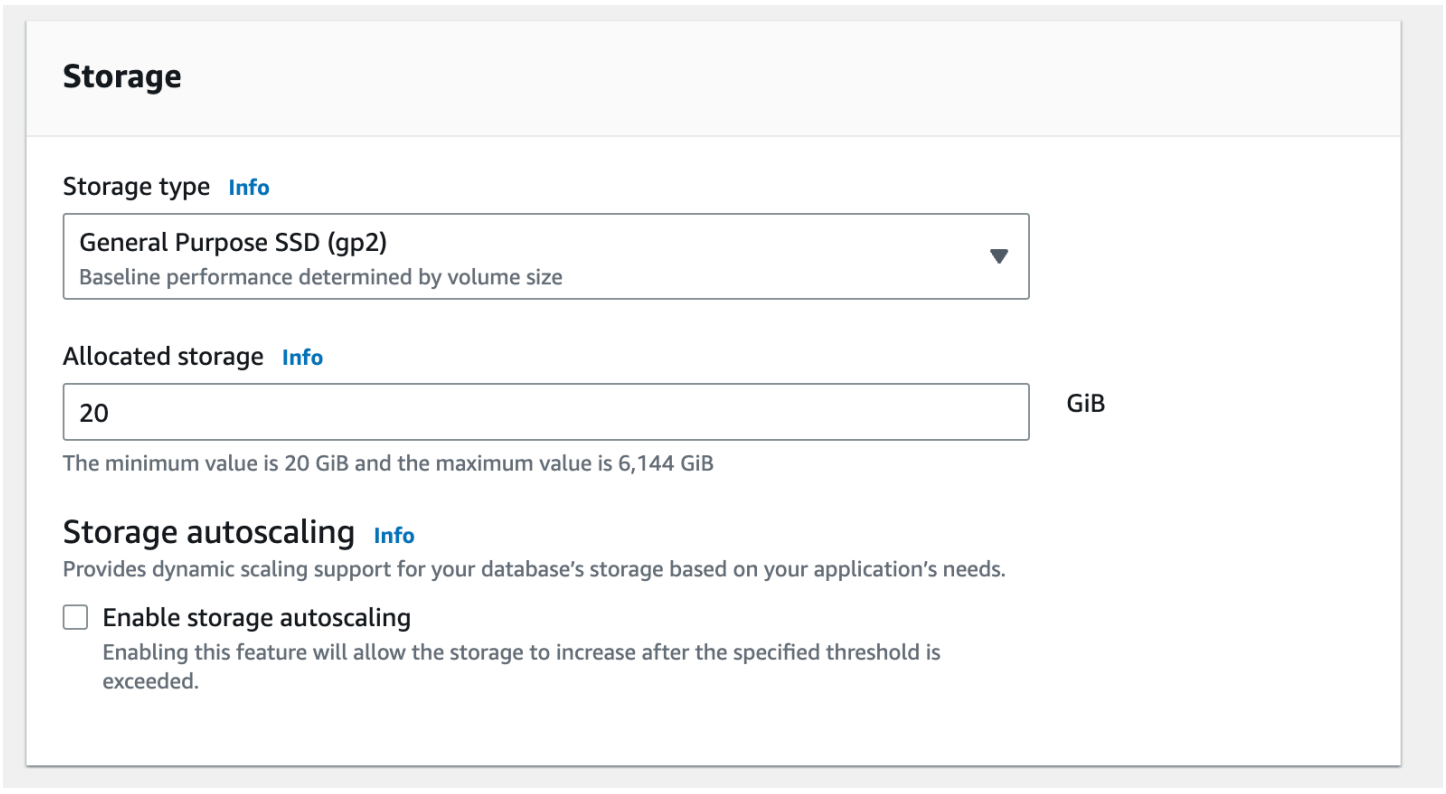
In this settings give database name and set database password

username: admin

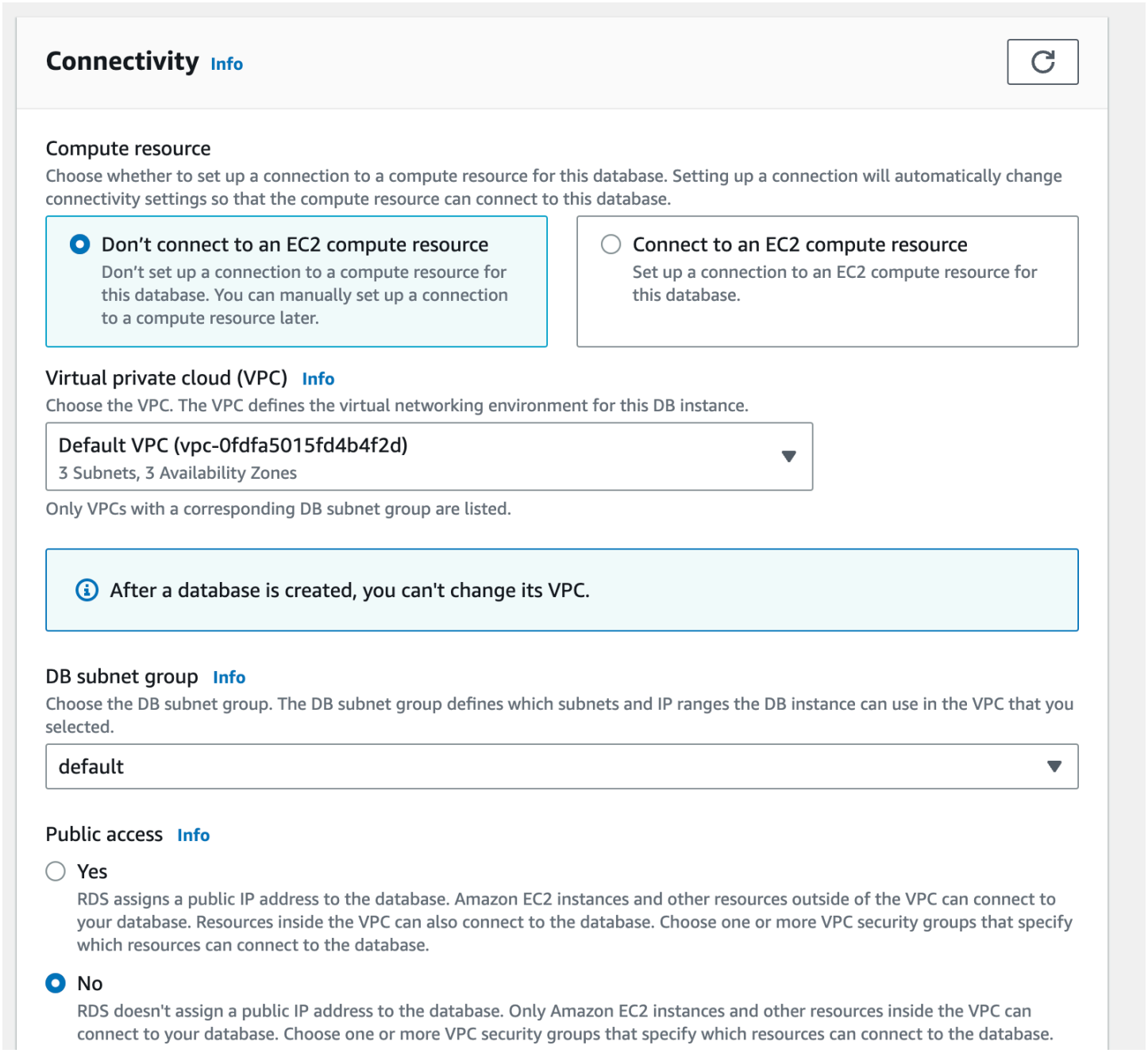
password: mypassword



In this instance configuration step select db.t2.micro which is completely free tier



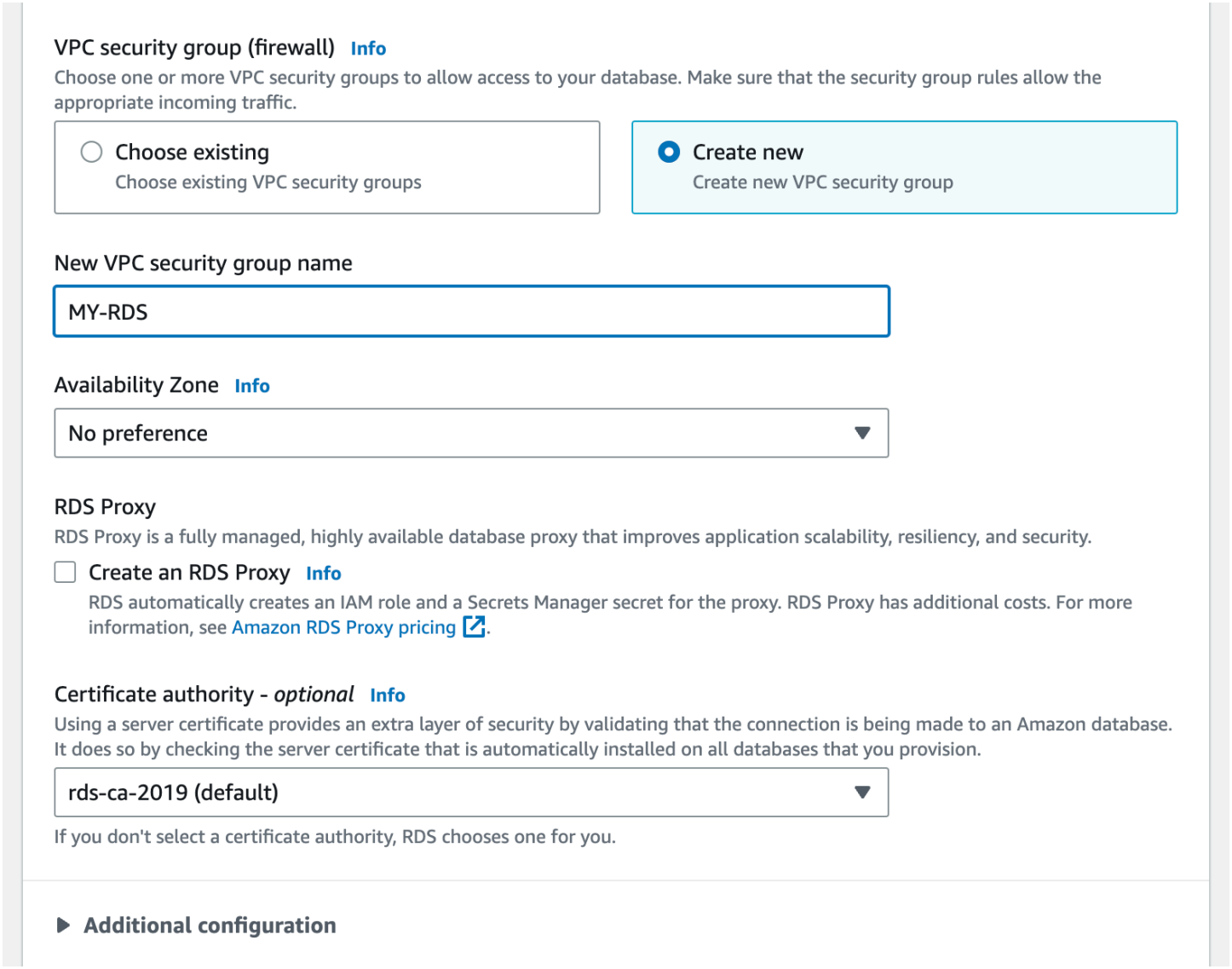
In this storage section give 20 GB of SSD and disable autoscaling



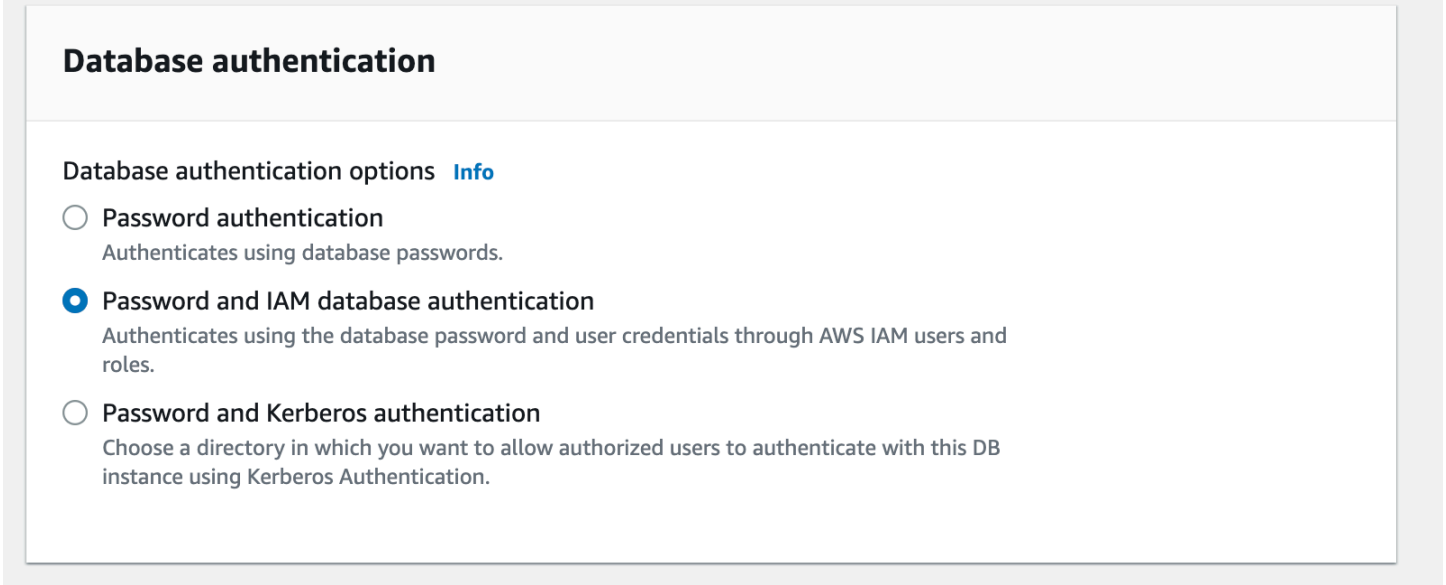
In this connectivity part, Since i don't have any EC2 instances so i am not going to connect my servers,

VPC and Subnets will be default

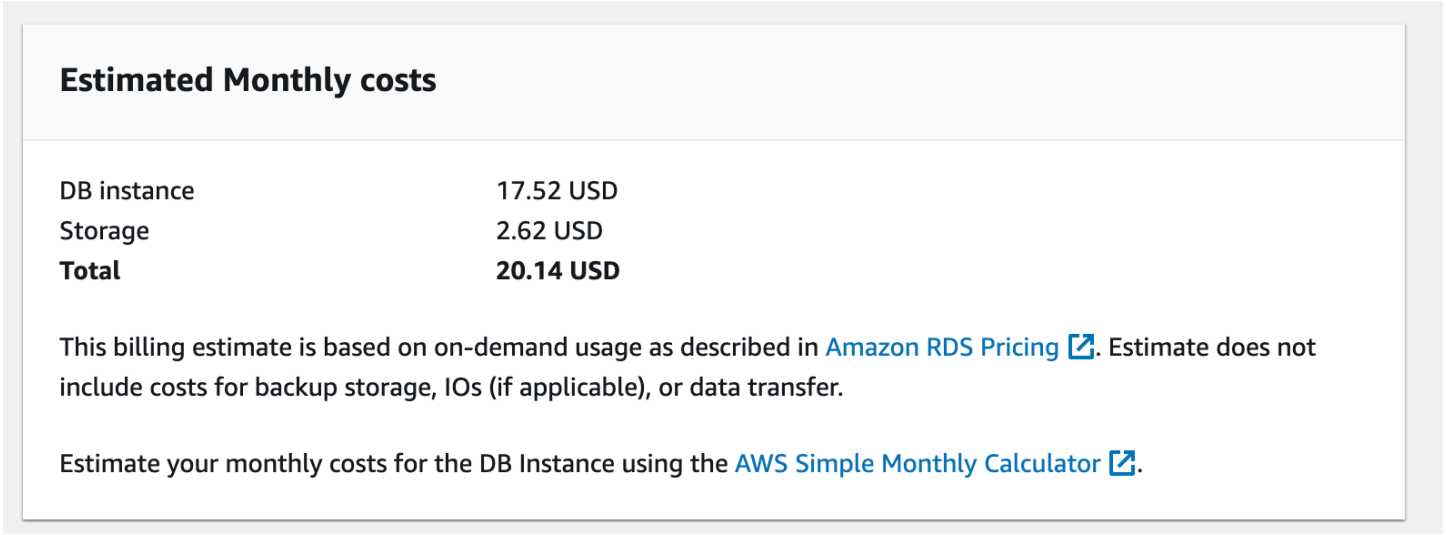
Public access will be NO



Security groups : it will create a new Sg for me and then i will change it to later as per my requirement.

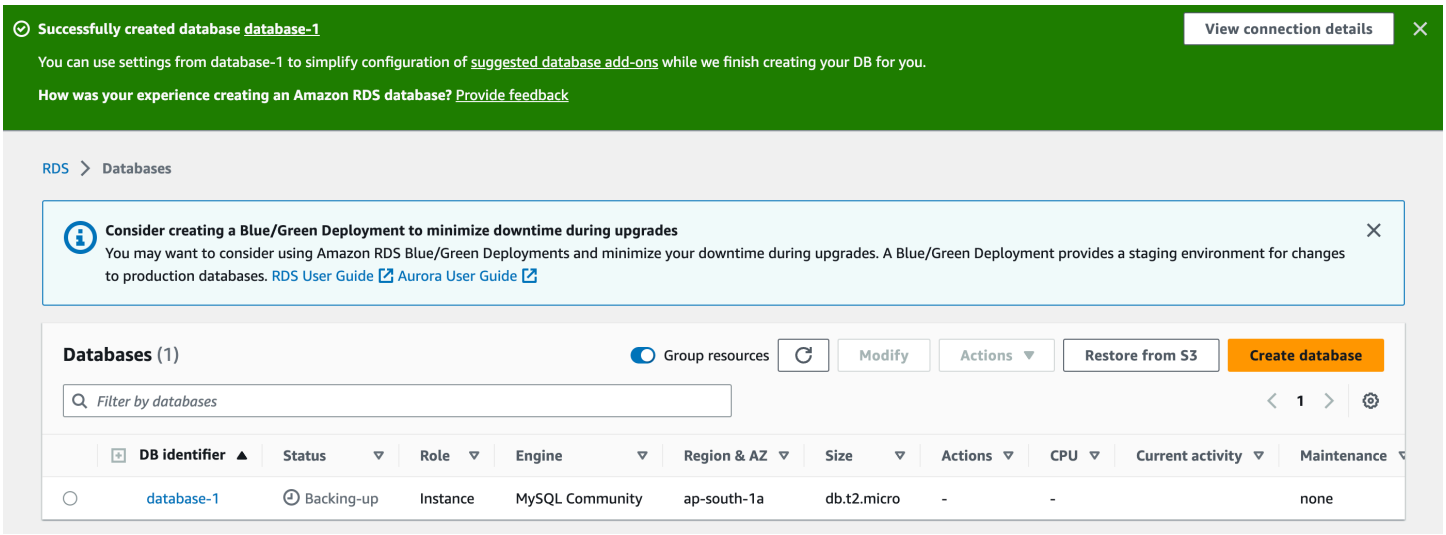


Here we have to select Password and IAM database authentication

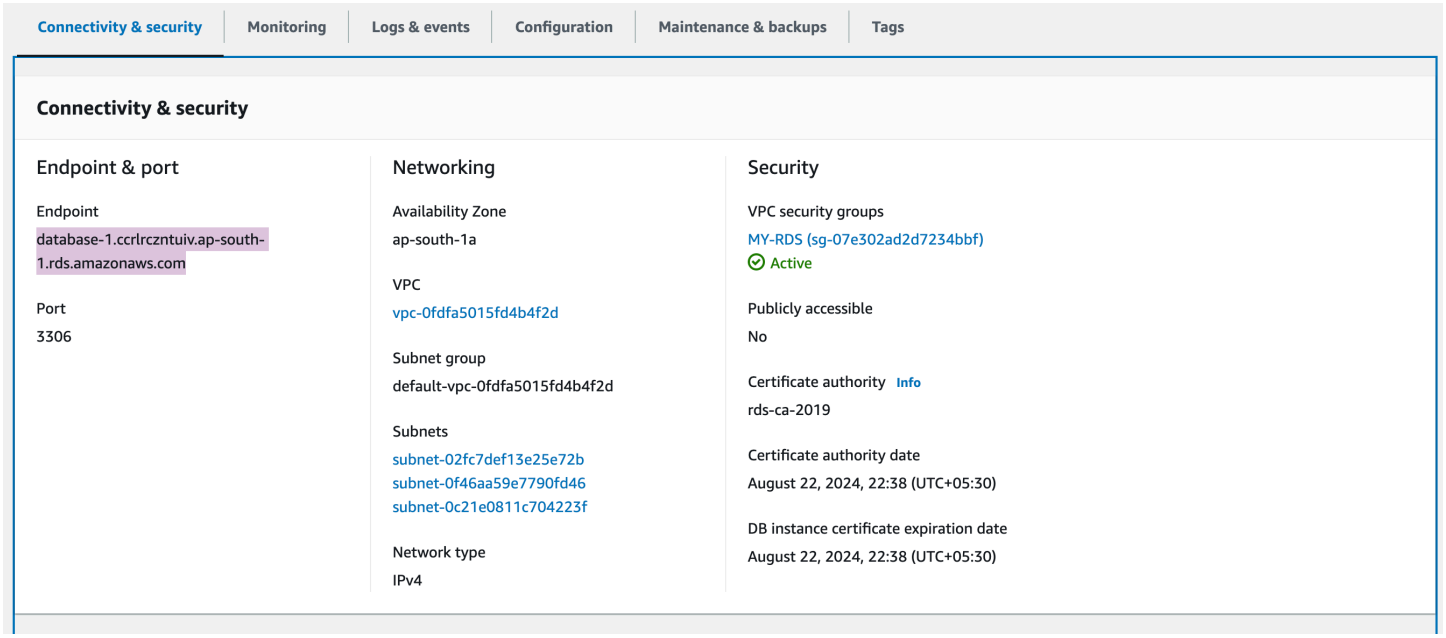


And this is the monthly billing, but dont worry about this because we are using only less than one hour, so we will not get billing much may be 5-6 rs

Finally click on create database it will take atleast 5 minutes to create our database.



Finally my Database is created.



This is details of my database

LAUNCH AN INSTANCE IN SAME VPC (where our DB is created)

INSTALL MYSQL:

- sudo rpm -Uvh https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
- rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
- sudo yum install mysql-community-server -y
- sudo systemctl enable mysqld
- sudo systemctl start mysqld
- sudo grep 'temporary password' /var/log/mysqld.log
- sudo mysql_secure_installation

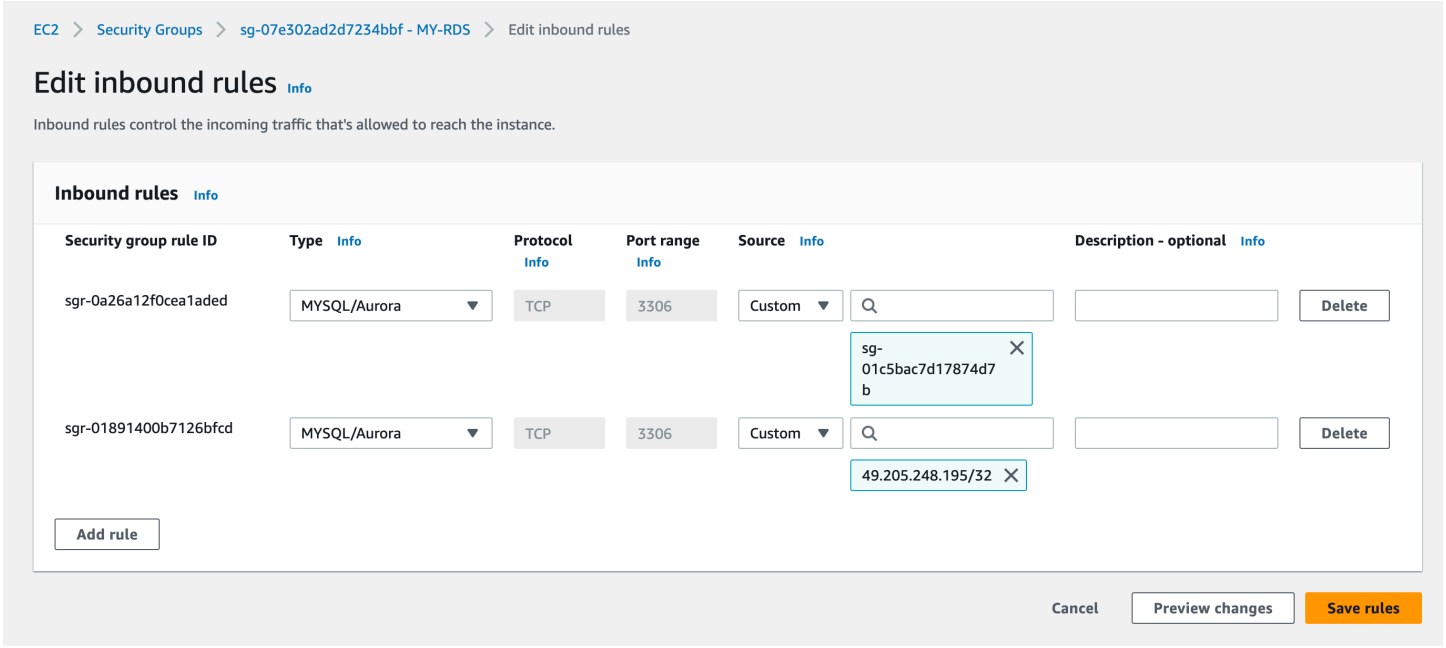
TO CONNECT WITH DATABASE

- Modify the security groups :

Go to security groups >> select MY-RDS security groups and click on edit inbound rules >>>

Add rule >> MySql/Aurora

source : our-instance-sg (MY-SG)



click on save now

This means we are adding our SG to DATABASE SG

- perform command to connect with database : **mysql -h endpoint -u username -p**
- It will asks you password enter it and you will connect with database.
- after connecting with mysql, perform some database commands
  - **show databases;** ---> to show the list of databases
  - **CREATE DATABASE accounts**; -----> this command is used to create a database

INSTALL GIT AND GET A SOURCE CODE

go to /**root/docker-webapp/src/main/resources**

You will find a database schema/query (db_backup.sql) that you have to deploy on database.

command: **mysql -h endpoint -u user -p database_name < db_backup.sql**

enter the password

Login into database: **mysql -h endpoint -u user -p**

use accounts; --------> To change to accounts database

show tables; -----> To see list of tables in accounts database

exit from the database

INSTALL MAVEN & STEUP TOMCAT

To build and deploy the source code

give mvn clean package to get war file

copy the war file to webapps folder in tomcat

THE APPLICATION IS DEPLOYED INTO TOMCAT

TRY TO CREATE AN ACCOUNT INTO THE APPLICATION YOU WILL GET HTTP 500 ERROR

TO RESOLVE THOS WE NEED TO CONNECT DATABASE TO OUR APPLICATION.

To do that : vim apache-tomcat-9.0.76/webapps/vprofile-v2/WEB-INF/classes/application.properties

change the details as

username : admin

pasword : mypassword

endpoint : database-1.ccrlrczntuiv.ap-south-1.rds.amazonaws.com

jdbc.url=jdbc:mysql://**database-1.ccrlrczntuiv.ap-south-1.rds.amazonaws.com**:3306/**accounts**?useUnicode=true&characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull

jdbc.username=**admin**

jdbc.password=**mypassword**

RESTART TOMCAT AND USE APPLICATION