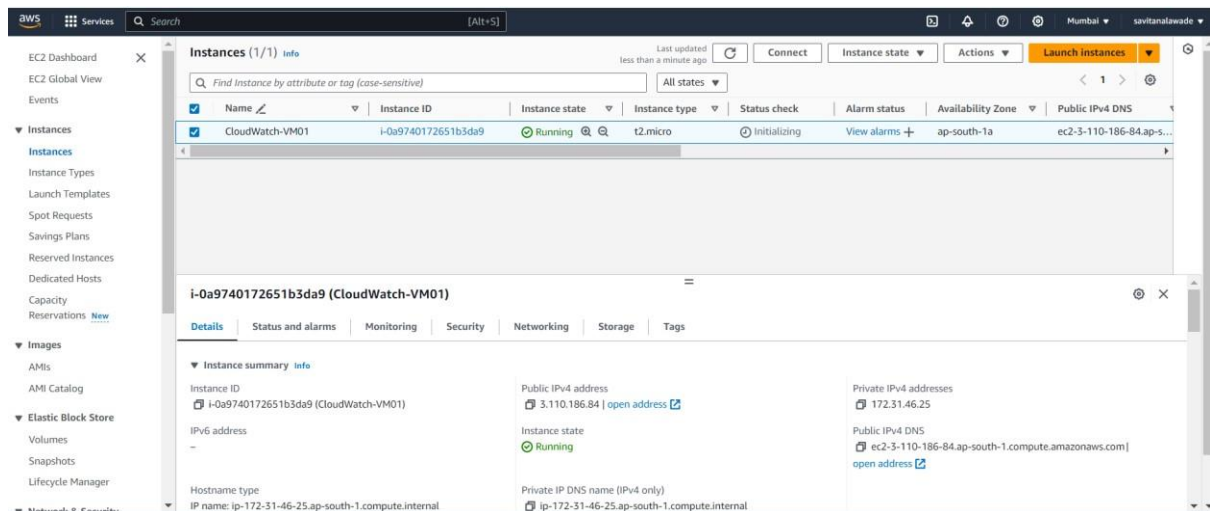


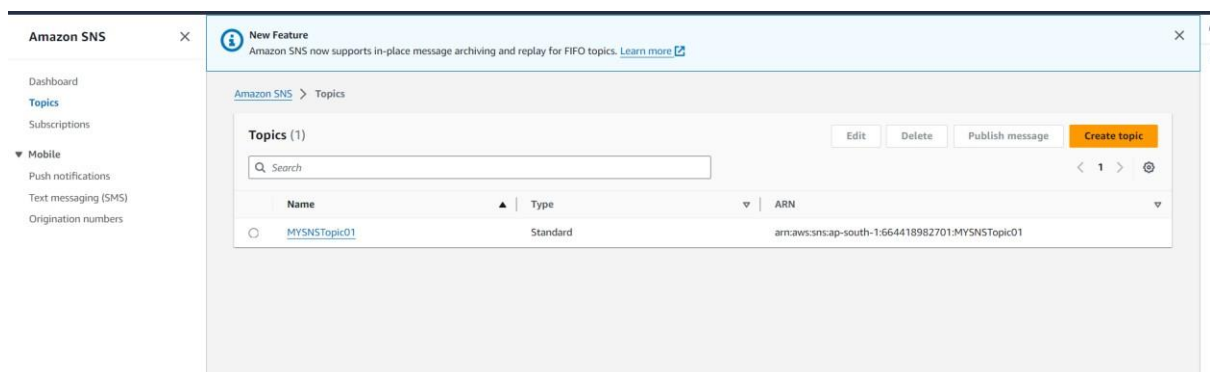
Cloud Watch

Ritesh Deshmukh

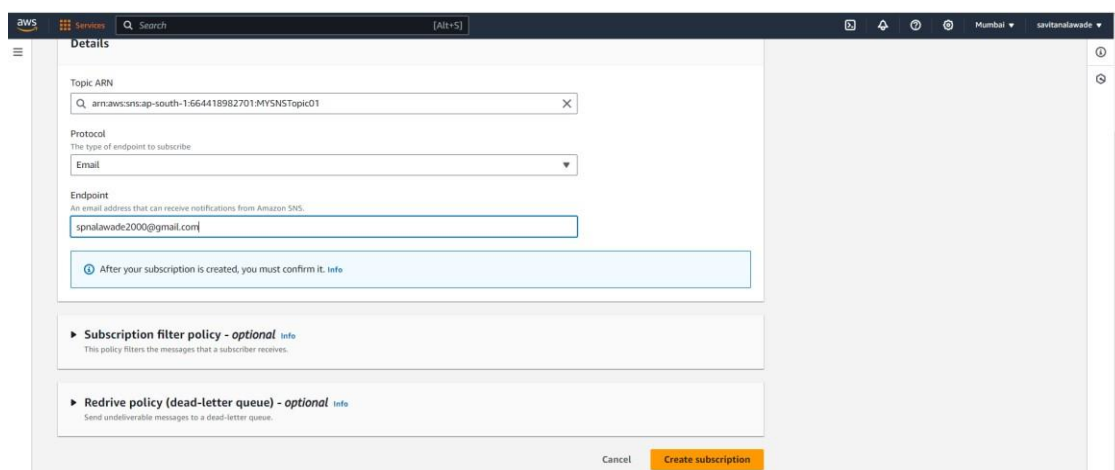
1) Create Instance



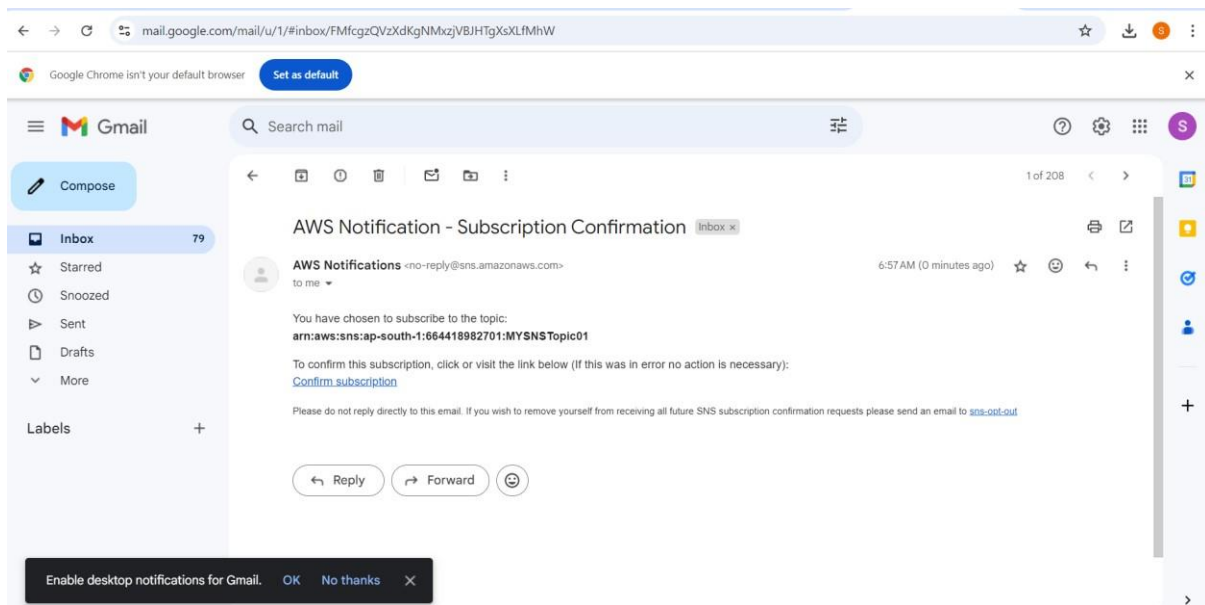
2) By Default, there will be no Alarm to set alarm go to SNS service and create topic first



3) Create Subscription



4) After creating subscription, got email for confirmation



5) Subscription confirmed



Simple Notification Service

Subscription confirmed!

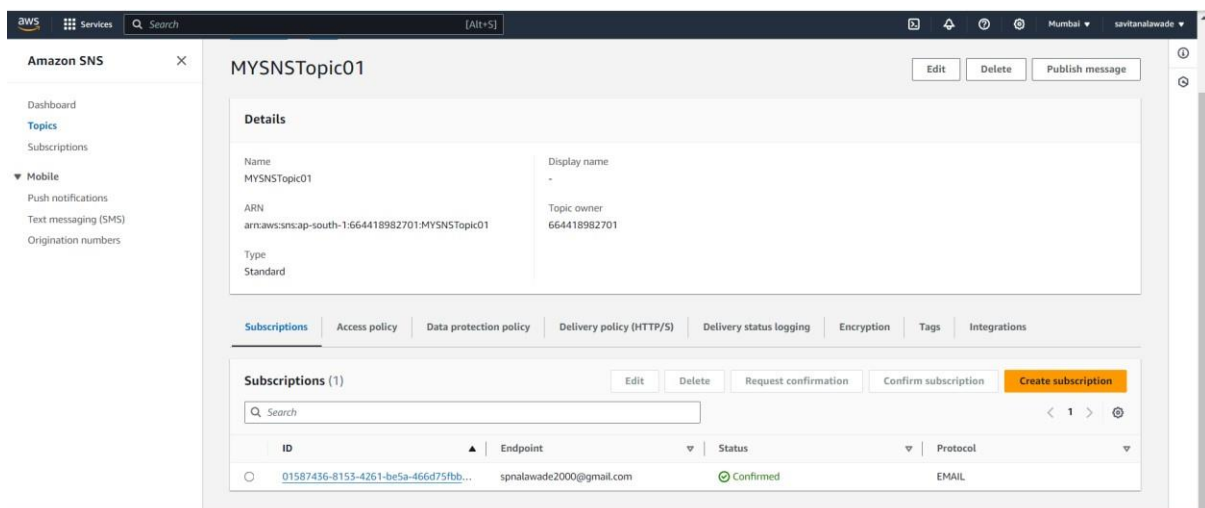
You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:ap-south-1:664418982701:MYSNSTopic01:01587436-8153-4261-be5a-466d75fbb7e9

If it was not your intention to subscribe, [click here to unsubscribe](#).

6) After conformation, status will changed to "confirmed"



7) Now, GO to “CloudWatch” service and create alarm

a. Specify metric and conditions

The screenshot shows the 'Specify metric and conditions' step of the AWS CloudWatch 'Create alarm' wizard. The left sidebar indicates the current step is Step 1. The main area is titled 'Specify metric and conditions' and contains a 'Metric' section with a graph and a configuration table.

Metric Configuration:

Namespace	Metric name	Instanceid	Instance name	Statistic	Period
AWS/EC2	CPUUtilization	i-0a9740172651b3da9	CloudWatch-VM01	Average	2 minutes

The graph shows a blue line representing CPU utilization over time, with a red threshold line at 80%.

b. Configure actions

The screenshot shows the 'Configure actions' step of the AWS CloudWatch 'Create alarm' wizard. The left sidebar indicates the current step is Step 2. The main area is titled 'Configure actions' and contains a 'Notification' section and a 'Lambda action' section.

Notification Configuration:

Alarm state trigger: Define the alarm state that will trigger this action.

- ☒ In alarm: The metric or expression is outside of the defined threshold.
- ☐ OK: The metric or expression is within the defined threshold.
- ☐ Insufficient data: The alarm has just started or not enough data is available.

Send a notification to the following SNS topic:

- ☒ Select an existing SNS topic
- ☐ Create new topic
- ☐ Use topic ARN to notify other accounts

Send a notification to...

am-aws-sns-ap-south-1-664418982701-MY5NSTo
arn:aws:sns:ap-south-1-664418982701-MY5NS:Topic01

Email (endpoints):
spnawade2000@gmail.com - View in SNS Console

Lambda action:

c. Add name and description

The screenshot shows the 'Add name and description' step of the AWS CloudWatch 'Create alarm' wizard. The left sidebar indicates the current step is Step 3. The main area is titled 'Add name and description' and contains a 'Name and description' section.

Name and description:

Alarm name: CPU_Utilization_i-0a9740172651b3da9

Alarm description - optional: View formatting guidelines

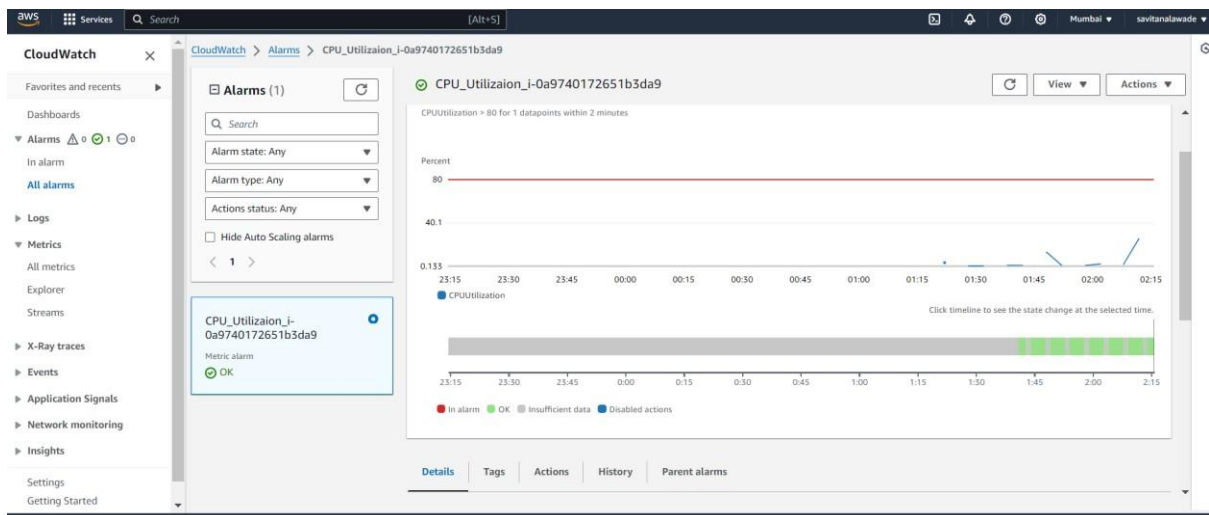
Markdown formatting: # This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/3) inline link.
Up to 1024 characters (57/1024)

Buttons: Cancel, Previous, Next

The screenshot shows the AWS CloudWatch console displaying the 'Successfully created alarm CPU_Utilization_i-0a9740172651b3da9' message. The 'Alarms (1/1)' table shows the details of the created alarm.

Name	State	Last state update (UTC)	Conditions	Actions
CPU_Utilization_i-0a9740172651b3da9	Insufficient data	2024-09-10 01:39:35	CPUUtilization > 80 for 1 datapoints within 2 minutes	Actions enabled

e. CPU utilization



8) To increase cpu utilization installed stress package with below steps:

- ```
{
1. sudo amazon-linux-extras install epel -y
2. sudo yum install stress -y
}
```

```
install 1 Package
total download size: 39 k
installed size: 94 k
downloading packages:
warning: /var/cache/yum/x86_64/2/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
public key for stress-1.0.4-16.el7.x86_64.rpm is not installed
stress-1.0.4-16.el7.x86_64.rpm
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x352c64e5:
Userid : "Fedora EPEL (7) <epel@fedoraproject.org>"
Fingerprint: 91e9 7d7c 4a5e 96f1 7f3e 888f 6a2f aea2 352c 64e5
Package : epel-release-7-11.noarch (@amazon2extra-epel)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : stress-1.0.4-16.el7.x86_64 1/1
Verifying : stress-1.0.4-16.el7.x86_64 1/1
Installed:
stress.x86_64 0:1.0.4-16.el7
Complete!
root@ip-172-31-46-25 ~# sudo stress --cpu 20 --timeout 1000
stress: info: [15191] dispatching hogs: 20 cpu, 0 io, 0 vm, 0 hdd
```

9) Received confirmation over the email as cpu got high

ALARM: "CPU\_Utilizaion\_i-0a9740172651b3da9" in Asia Pacific (Mumbai) Inbox x

**AWS Notifications** <no-reply@sns.amazonaws.com> 7:53 AM (0 minutes ago)  
to me

You are receiving this email because your Amazon CloudWatch Alarm "CPU\_Utilizaion\_i-0a9740172651b3da9" in the Asia Pacific (Mumbai) ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [99.7720430107527 (10/09/24 02:17:00)] was greater than the threshold (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 10 September, 2024 02:23:57 UTC".

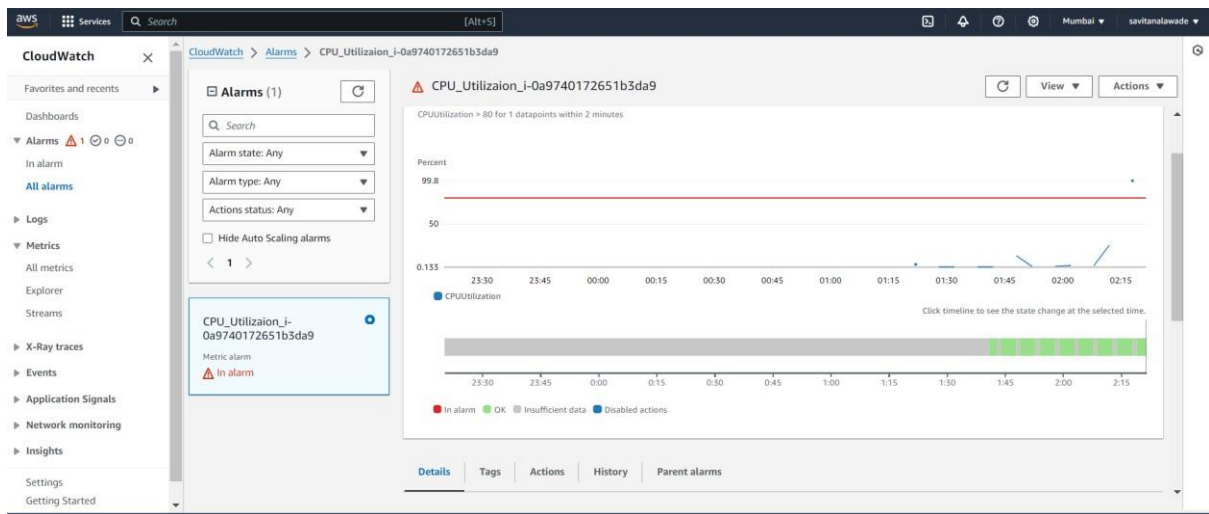
View this alarm in the AWS Management Console:  
[https://ap-south-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-south-1#alarmsV2:alarm/CPU\\_Utilizaion\\_i-0a9740172651b3da9](https://ap-south-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-south-1#alarmsV2:alarm/CPU_Utilizaion_i-0a9740172651b3da9)

Alarm Details:

- Name: CPU\_Utilizaion\_i-0a9740172651b3da9
- Description:
- State Change: INSUFFICIENT\_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [99.7720430107527 (10/09/24 02:17:00)] was greater than the threshold (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 10 September, 2024 02:23:57 UTC
- AWS Account: 664418982701
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:664418982701:alarm:CPU\_Utilizaion\_i-0a9740172651b3da9

Threshold:

As we can see utilization is in Alarm state



1) Create IAM role with (CloudWatchfullaccess & SSM)

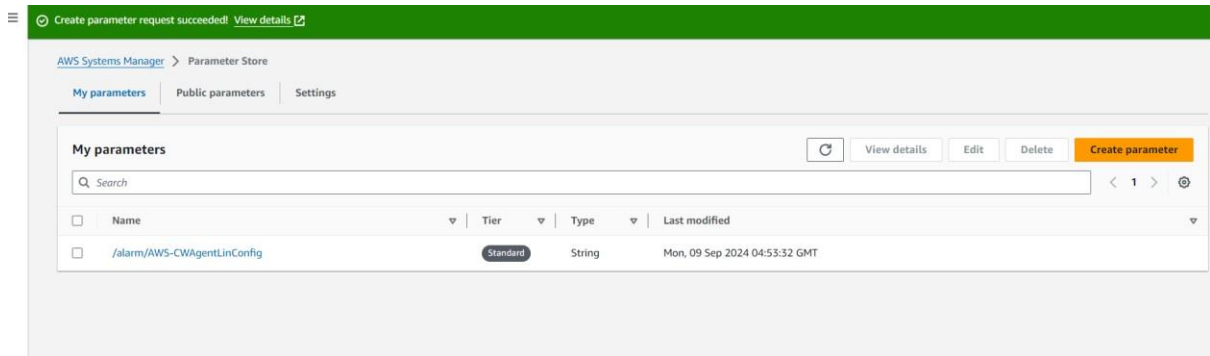
The screenshot shows the AWS IAM console. A green banner at the top indicates that the 'Role EC2-CloudWatch-Role' has been created. Below the banner, the 'Roles (15)' section is displayed. A table lists the roles, with the 'EC2-CloudWatch-Role' highlighted. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'.

| Role name                                           | Trusted entities    | Last activity |
|-----------------------------------------------------|---------------------|---------------|
| <a href="#">EC2-CloudWatch-Role</a>                 | AWS Service: ec2    | -             |
| <a href="#">iamadminrole</a>                        | AWS Service: ec2    | 14 days ago   |
| <a href="#">lambdafunction2role</a>                 | AWS Service: lambda | -             |
| <a href="#">s3crr_role_for_s3lifecyclebucket001</a> | AWS Service: s3     | 19 days ago   |

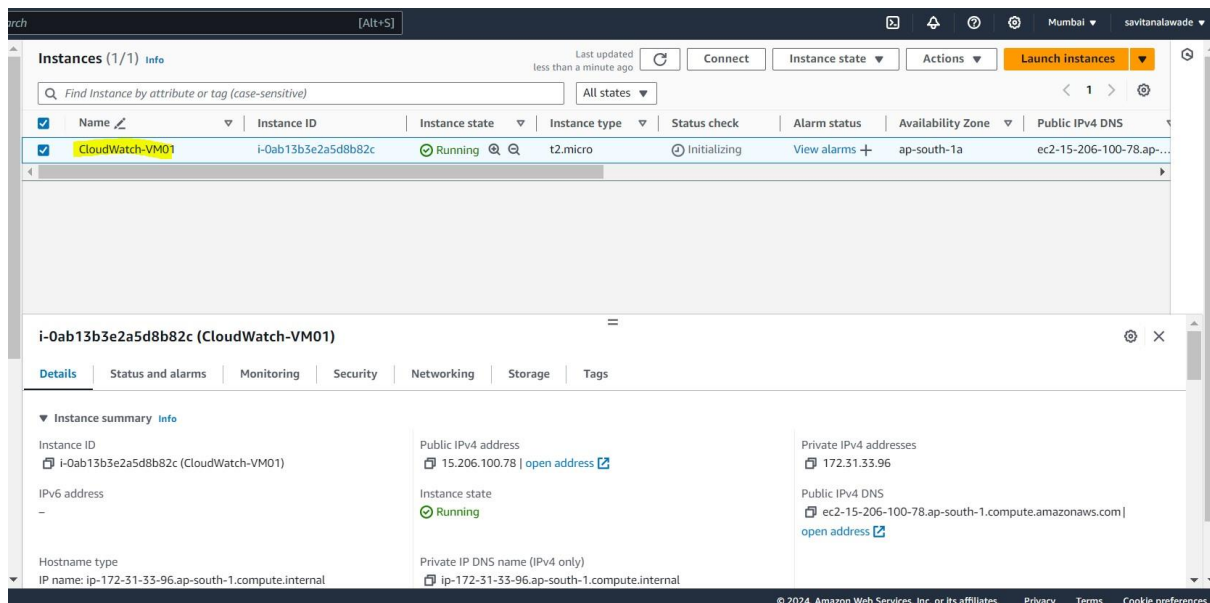
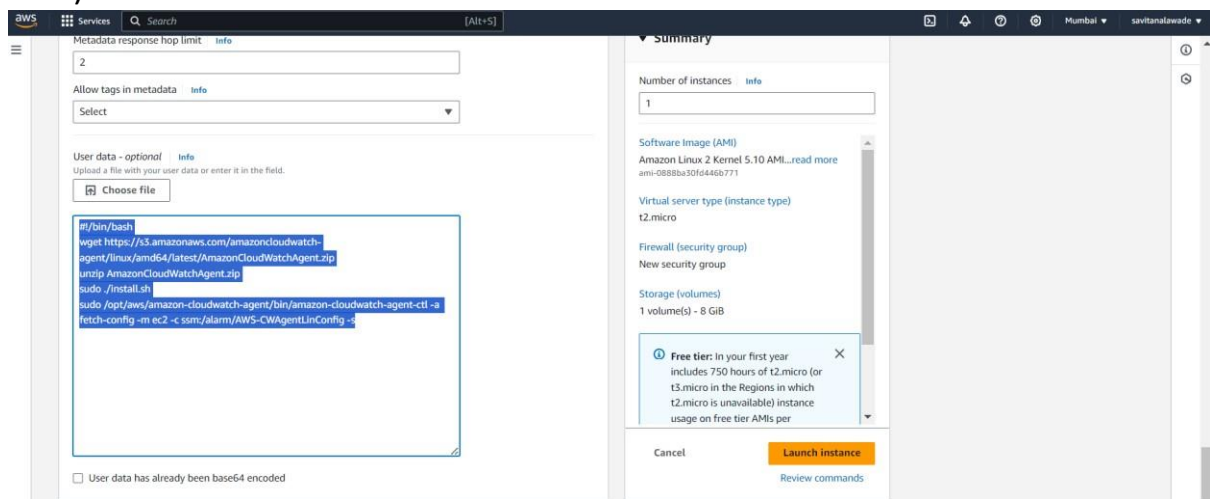
2) Create a Parameter in Systems Manger with the name "/alarm/AWSCWAgentLinConfig" and store the value.

The screenshot shows the AWS Systems Manager console, specifically the 'Create parameter' page. The 'Name' field is filled with '/alarm/AWSCWAgentLinConfig'. The 'Description' field is also filled with '/alarm/AWSCWAgentLinConfig'. The 'Tier' section shows 'Standard' selected, with a note that standard parameters cannot be shared with other AWS accounts. The 'Type' section shows 'String' selected. The footer of the console shows the copyright notice for Amazon Web Services, Inc. or its affiliates.

Created Successfully.



### 3) Create Instance



### 4) Cloud Agent have been installed





```
[root@ip-172-31-33-96 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda 202:0 0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
xvdb 202:16 0 10G 0 disk

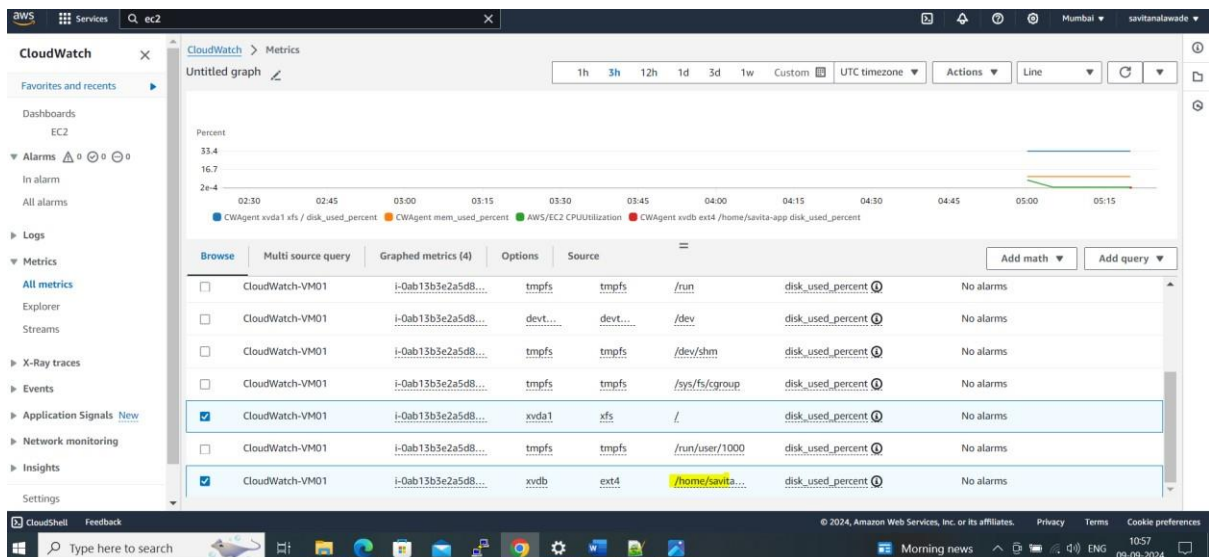
[root@ip-172-31-33-96 ~]# mkfs -t ext4 /dev/xvdb
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-33-96 ~]# mkdir /home/savita-app
[root@ip-172-31-33-96 ~]# mount /dev/xvdb /home/savita-app
```

```
[root@ip-172-31-33-96 ~]# mkdir /home/savita-app
[root@ip-172-31-33-96 ~]# mount /dev/xvdb /home/savita-app
[root@ip-172-31-33-96 ~]# df -h
Filesystem Size Used Avail Use% Mounted on
devtmpfs 467M 0 467M 0% /dev
tmpfs 477M 0 477M 0% /dev/shm
tmpfs 477M 468K 476M 1% /run
tmpfs 477M 0 477M 0% /sys/fs/cgroup
/dev/xvda1 8.0G 2.7G 5.4G 34% /
tmpfs 96M 0 96M 0% /run/user/1000
tmpfs 96M 0 96M 0% /run/user/0
/dev/xvdb 9.7G 24K 9.2G 1% /home/savita-app
[root@ip-172-31-33-96 ~]#
```

## 6) Add Newly created volume in graph



## 7) We can see dashboard now with four parameters

(CPU-utilization, Memory-utilization, root-disk-utilization  
volume utilization)

& EBS-



