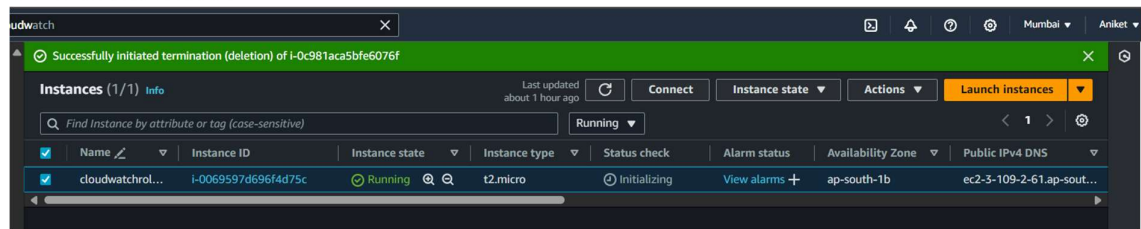


CLOUDWATCH PRATICAL

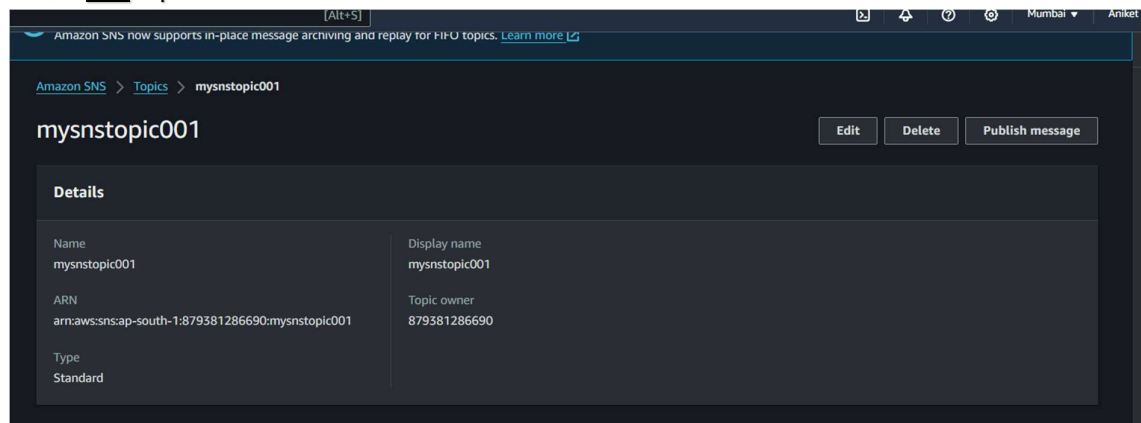
ANIKET THORAT..

- Monitor the CPU utilization of Instance and set threshold of utilization greater than 80% and send alert via SNS.

1. CREATE A INSTANCE



2. create SNS topic



3. SUBSCRIBE THE SNS..

AWS Notification - Subscription Confirmation Inbox x



mysnstopic001 <no-reply@sns.amazonaws.com>
to me ▾

Sat, Sep 14, 2:11PM (7 days ago)

You have chosen to subscribe to the topic:
arn:aws:sns:ap-south-1:879381286690:mysnstopic001

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

4. SUBSCRIPTION CONFIRM



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

```
arn:aws:sns:ap-south-1:879381286690:mysnstopic001:4f296cb1-d4dd-432c-b2fa-06e752162212
```

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

5. AFTER SUBSCRIPTION DONE THE STATUS WILL CHANGE TO CONFIRM

Subscriptions

Access policy

Data protection policy

Delivery policy (HTTP/S)

Delivery status logging

Encryption

Tags

Integrations

Subscriptions (1)

Edit

Delete

Request confirmation

Confirm subscription

Create subscription

<

1

>

⚙

	ID	Endpoint	Status	Protocol
<input type="radio"/>	4f296cb1-d4dd-452c-b2fa-06e75216...	aniketthoratl402@gmail.com	Confirmed	EMAIL

6. CREATE A CLOUD WATCH ALARM FOR CPU UTILIZATION

a. Specify PERIOD and conditions

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 2 minutes.

Percent

2.48

2.21

1.94

06:30 07:30 08:30

CPUUtilization

Namespace
AWS/EC2

Metric name
CPUUtilization

InstanceId
i-0c981aca5bfe6076f

Instance name
cloudwatch001

Statistic
Average

Period
2 minutes

Conditions

Threshold type

☒ Static
Use a value as a threshold

☐ Anomaly detection
Use a band as a threshold

B . Configure THE ALARM CONDITION TO STATIC AND THRESHOLD VALUE ≥ 50

1.96

06:30 07:30 08:30

CPUUtilization

Instance name
cloudwatch001

Statistic
Average

Period
2 minutes

Conditions

Threshold type

☒ Static
Use a value as a threshold

☐ Anomaly detection
Use a band as a threshold

Whenever CPUUtilization is...
Define the alarm condition.

☐ Greater
> threshold

☒ Greater/Equal
 \geq threshold

☐ Lower/Equal
 \leq threshold

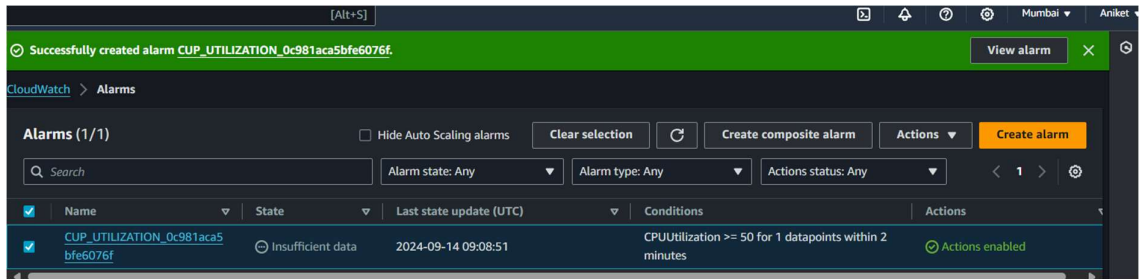
☐ Lower
< threshold

than...
Define the threshold value.

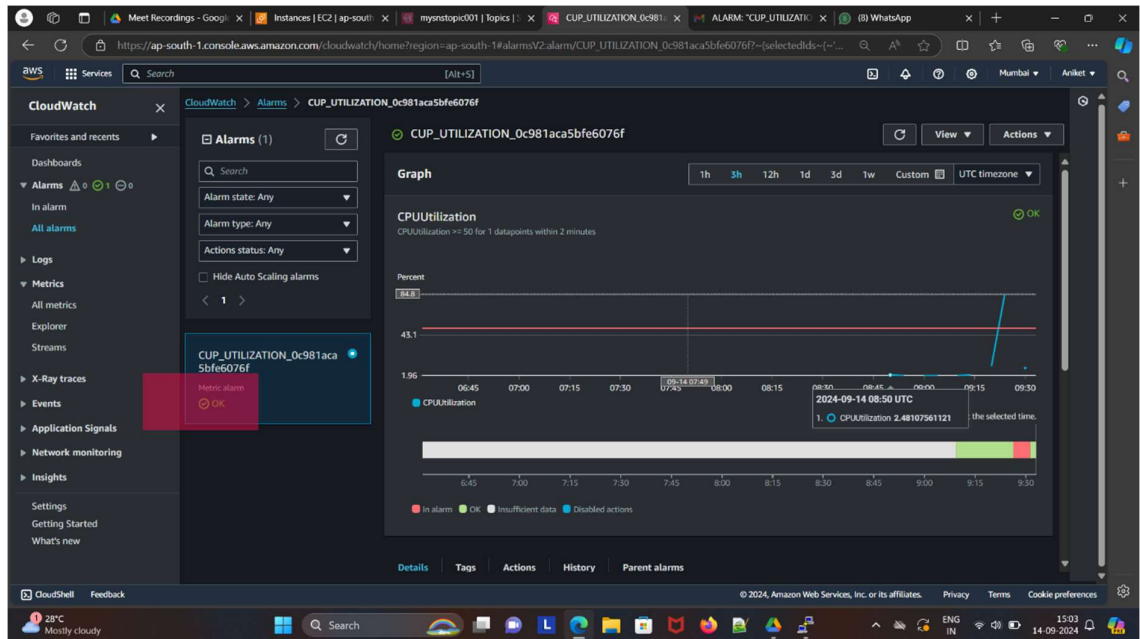
50

Must be a number

C. CLOUD WATCH ALARM CREATED .



7. WE CAN CE THE CPU UTILIZATION IS IN OK STATE



8. TO INCREASE THE CPU UTILIZATION WE HAVE TO STRESS PAKAGE

STEPS

A. `sudo amazon-linux-extras install epel -y`

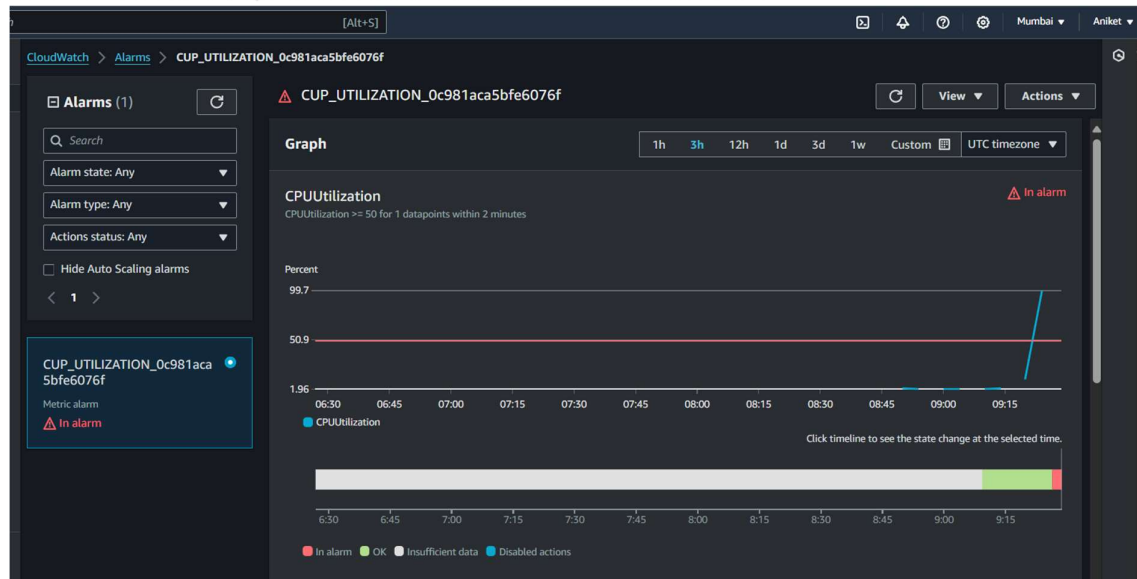
B. sudo yum install stress -y

```
root@ip-172-31-5-4:~  
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 | 39 kB 00:00  
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 (@amzn2extra-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
```

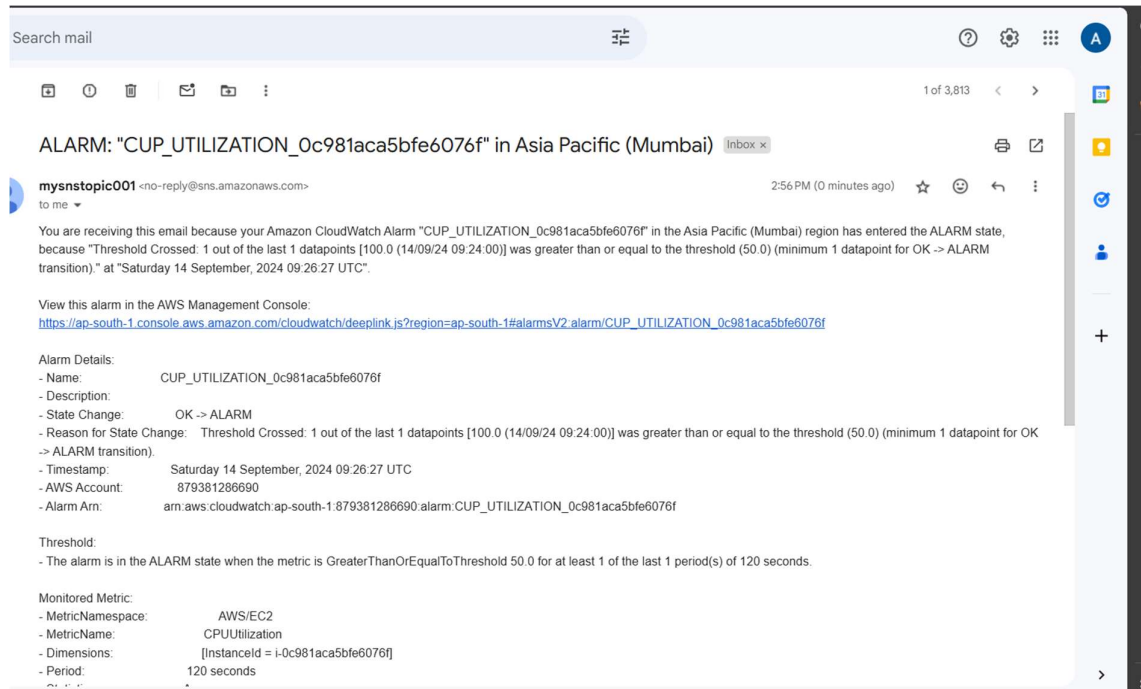
9. NOW TO INCREASE CPU UTILIZATION USE COMMAND (**stress --CPU 20 --timeout 1000**)

```
[root@ip-172-31-5-4 ~]# stress --cpu 20 --timeout 1000  
stress: info: [3579] dispatching hogs: 20 cpu, 0 io, 0 vm, 0 hdd
```

10. Now we can see the cpu utilization is increased now it is alarm state

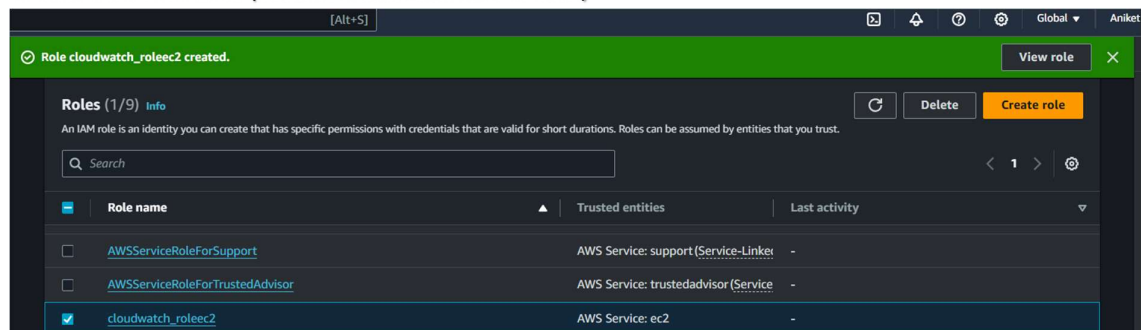


11. We have received an email

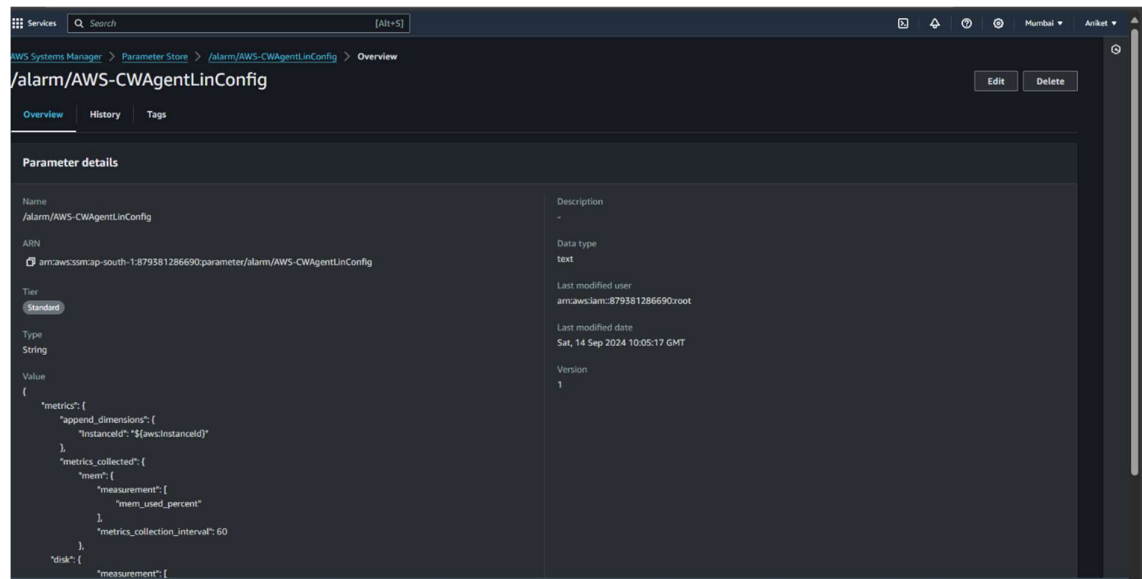


Install CloudAgent using bootstrapping and create dashboard of Utilizations.

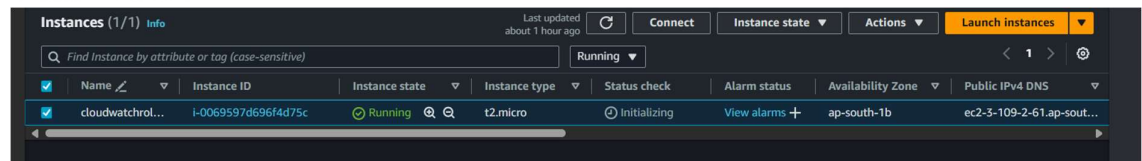
A. Create IAM role with (CloudWatchfullaccess & SSM)



- B. Create a Parameter in Systems Manager with the name `"/alarm/AWS-CWAgentLinConfig"` and store the value.



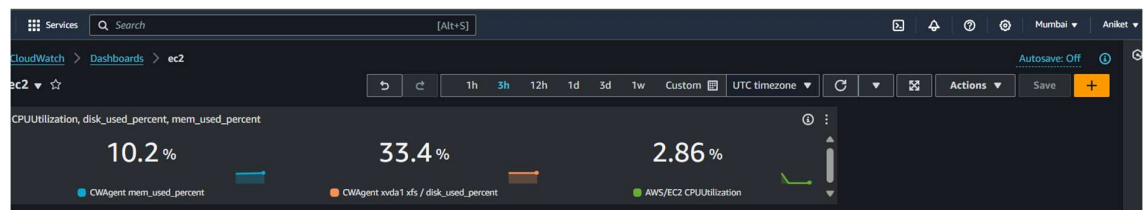
- C. Create a instance



D. CloudAgent have been installed successfully

```
root@ip-172-31-0-26:~  
login as: ec2-user  
Authenticating with public key "aniket-2024"  
#  
#####  
~\#####\ Amazon Linux 2  
~\#####\  
~\#####\ AL2 End of Life is 2025-06-30.  
~\#####\  
~\#####\ A newer version of Amazon Linux is available!  
~\#####\  
~\#####\ Amazon Linux 2023, GA and supported until 2028-03-15.  
~\#####\ https://aws.amazon.com/linux/amazon-linux-2023/  
~\#####\  
ec2-user@ip-172-31-0-26 ~]$ sudo su -  
[root@ip-172-31-0-26 ~]# sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudw  
atch-agent-ctl -m ec2 -a status  
  
"status": "running",  
"starttime": "2024-09-14T10:08:35+0000",  
"configstatus": "configured",  
"version": "1.300044.0b793"  
  
[root@ip-172-31-0-26 ~]#
```

E. Created one Dashboard as "EC2" and add metrix



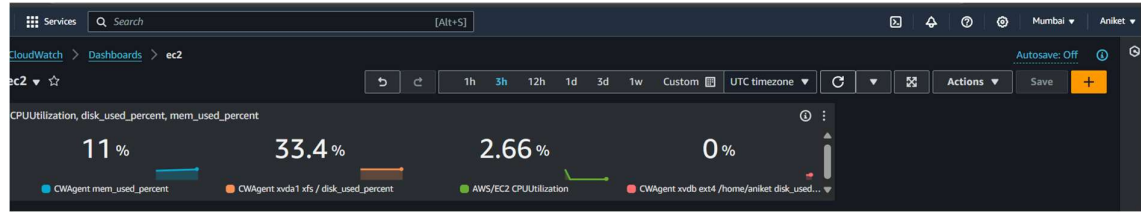
F. Create one EBS volume and mount it

```
[ec2-user@ip-172-31-0-26 ~]$ sudo su -
Last login: Sat Sep 14 10:11:42 UTC 2024 on pts/0
[root@ip-172-31-0-26 ~]# 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-172-31-0-26 ~]# 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-172-31-0-26 ~]# mkdir /home/aniket
[root@ip-172-31-0-26 ~]# mount /dev/xvdb /home/aniket
[root@ip-172-31-0-26 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        467M   0  467M   0% /dev
tmpfs           477M   0  477M   0% /dev/shm
tmpfs           477M 448K  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
/dev/xvdb       9.7G  24K  9.2G   1% /home/aniket
[root@ip-172-31-0-26 ~]#
```

G. The dashboard created with four parameters (CPU-utilization,Memory-utilization,root-disk-utilization & EBS-volume utilization)



Thanyou....