Task4- AWS VPC, EC2 Backup & Monitoring

1. VPC SETUP

VPC Name:

projecttask4-vpc

Subnets:

Public Subnets:

projecttask4-subnet-public1-us-east-1a
projecttask4-subnet-public2-us-east-1b

Private Subnets:

projecttask4-subnet-private1-us-east-1a
projecttask4-subnet-private2-us-east-1b

Route Tables

Public Route Table:

projecttask4-rtb-public

Private Route Tables:

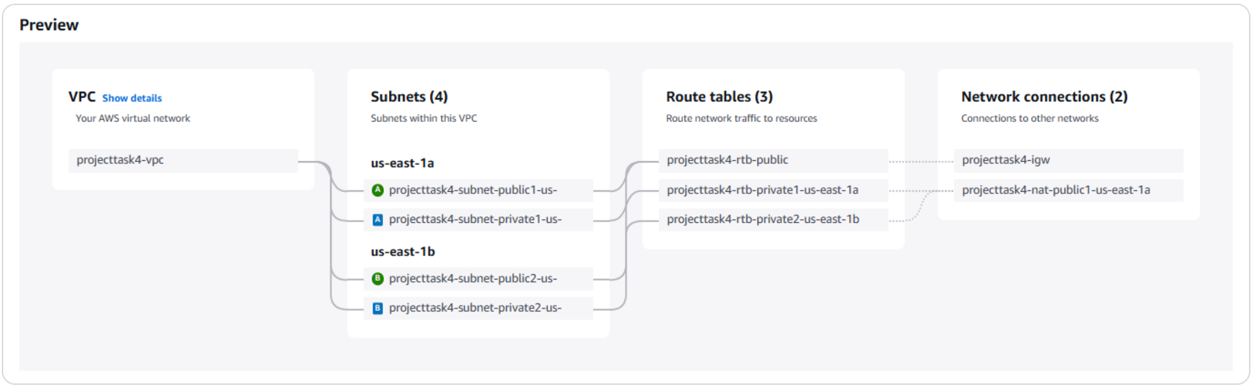
projecttask4-rtb-private1-us-east-1a

projecttask4-rtb-private2-us-east-1b

Network Connections

Internet Gateway: projecttask4-igw

NAT Gateway: projecttask4-nat-public1-us-east-1a



Preview

VPC Show details
Your AWS virtual network
projecttask4-vpc

Subnets (4)
Subnets within this VPC
us-east-1a
A projecttask4-subnet-public1-us-
A projecttask4-subnet-private1-us-
us-east-1b
B projecttask4-subnet-public2-us-
B projecttask4-subnet-private2-us-

Route tables (3)
Route network traffic to resources
projecttask4-rtb-public
projecttask4-rtb-private1-us-east-1a
projecttask4-rtb-private2-us-east-1b

Network connections (2)
Connections to other networks
projecttask4-igw
projecttask4-nat-public1-us-east-1a

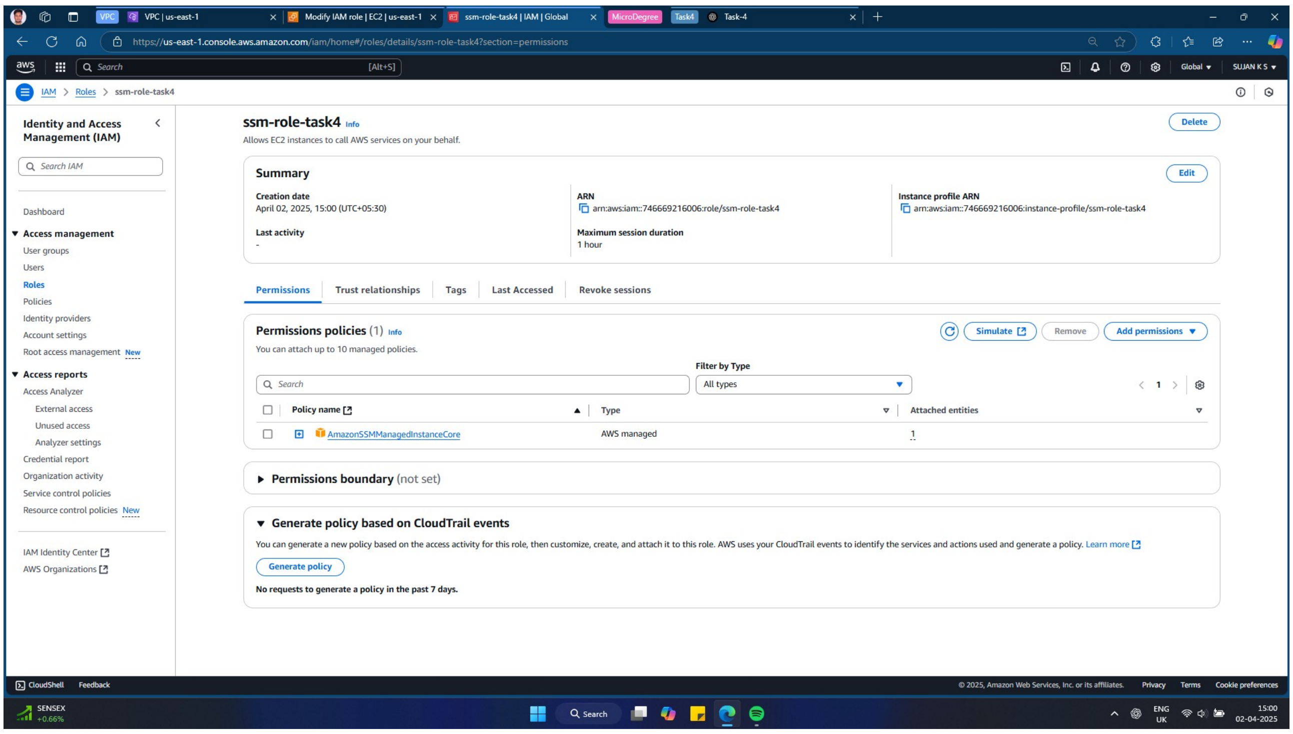
2. EC2 Deployment & Apache Setup

2.1 Launch EC2 Instance in Private Subnet

- Instance Type: t2.micro
- Key Pair: Select or create a key pair
- Subnet: Select a Private Subnet
- Security Group Rules:
    - Inbound: Allow HTTP (80), ICMP, and SSM (443)
    - Outbound: Allow all traffic
- User Data Script: (For automatic SSM Agent installation)

```
#!/bin/bash
sudo yum update -y
sudo yum install -y amazon-ssm-agent
sudo systemctl enable amazon-ssm-agent
sudo systemctl start amazon-ssm-agent
```

- IAM Role: Attach **AmazonSSMManagedInstanceCore >** Reboot instance



https://us-east-1.console.aws.amazon.com/iam/home#/roles/details/ssm-role-task4?section=permissions

IAM > Roles > ssm-role-task4

ssm-role-task4 Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date April 02, 2025, 15:00 (UTC+05:30)	**ARN** arn:aws:iam::746669216006:role/ssm-role-task4	**Instance profile ARN** arn:aws:iam::746669216006:instance-profile/ssm-role-task4
Last activity -	**Maximum session duration** 1 hour	

Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonSSMManagedInstanceCore	AWS managed	1

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

No requests to generate a policy in the past 7 days.

2.2 Connect to EC2 via SSM

1. AWS Console > Systems Manager > Session Manager
2. Select the instance > Click Start Session
3. Session starts

2.3 Install & Start Apache Web Server

sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd

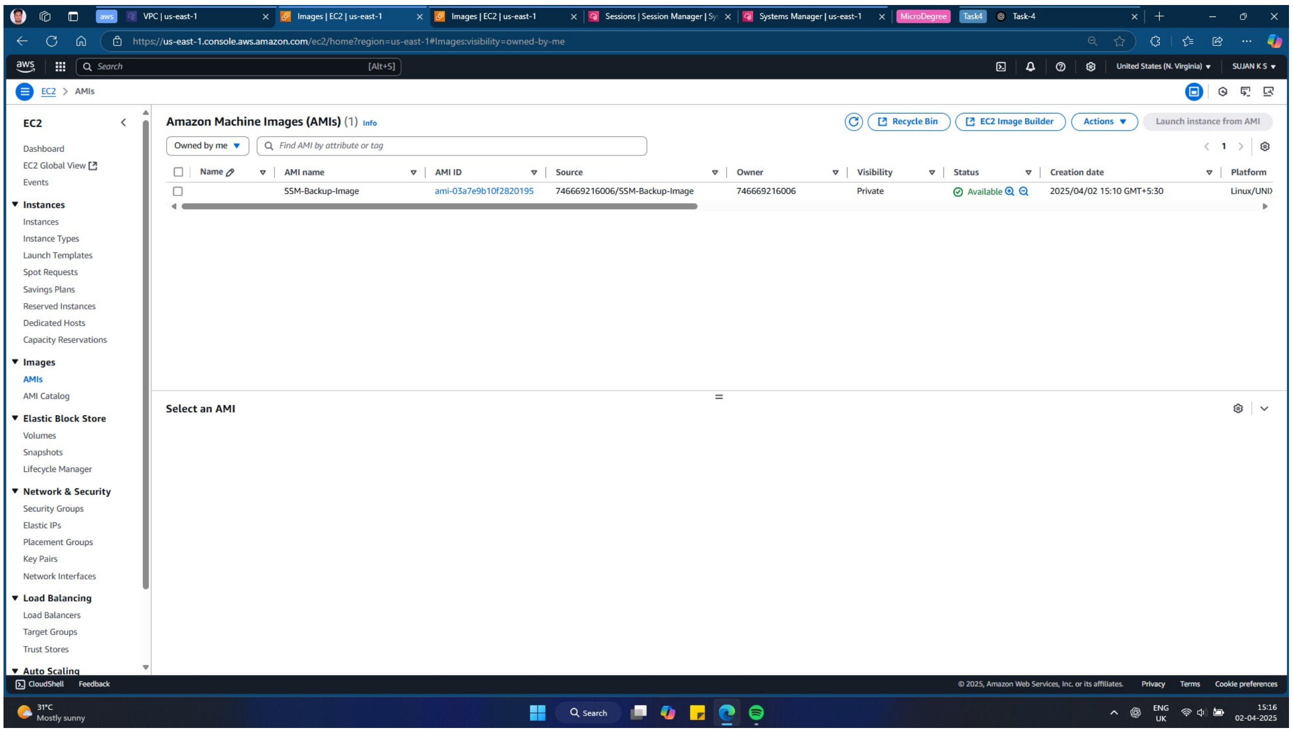
2.4 Verify Apache installation:

curl http://localhost
verified: The instance launched first with apache server and ssm agent will be connected using the session manager and checked with above cmd curl http://localhost
result: <html>...</html>.

3. Backup & Restore EC2

3.1 Create an AMI Backup

1. AWS Console >EC2 >Instances
2. Select the private EC2 instance
3. Actions >Image and templates > Create Image
4. Enter a name for the AMI and create it.



https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Images:visibility=owned-by-me

Amazon Machine Images (AMIs) (1)

Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date	Platform
	SSM-Backup-Image	ami-03a7e9b10f2820195	746669216006/SSM-Backup-Image	746669216006	Private	Available	2025/04/02 15:10 GMT+5:30	Linux/UNIX

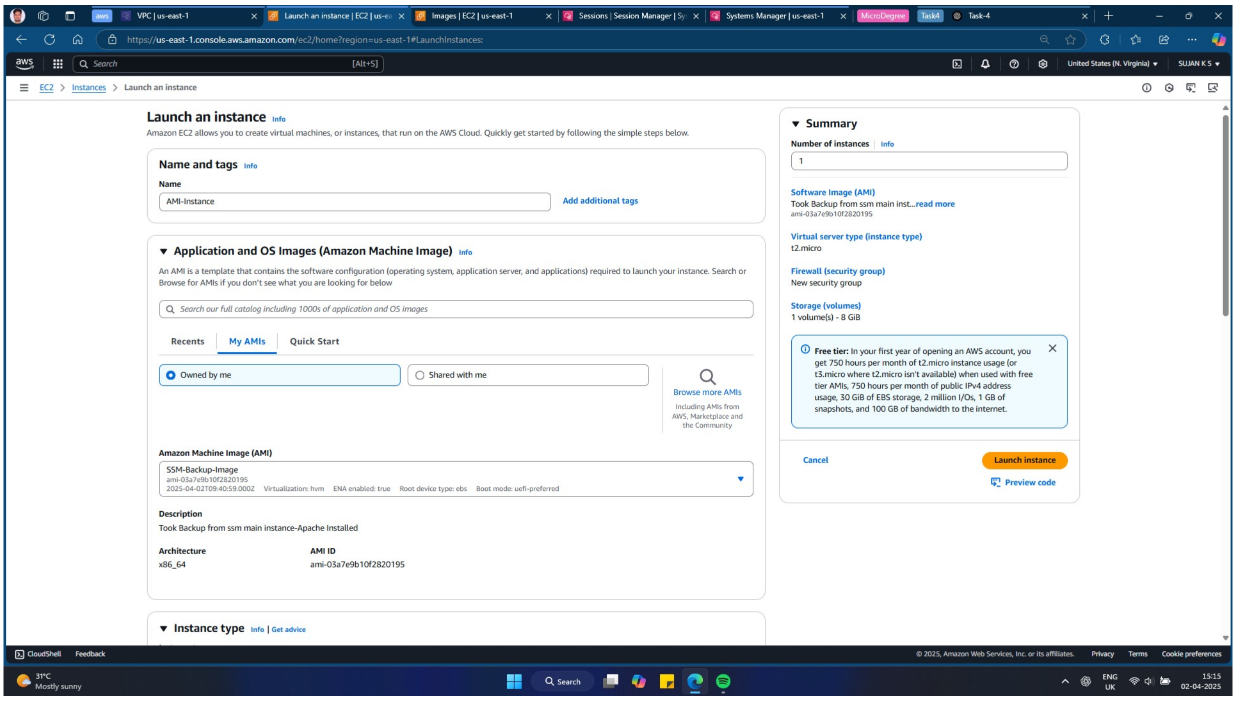
Select an AMI

3.2 Terminate EC2 Instance

1. Select the EC2 instance in the console
2. Actions > Instance State > Terminate

3.3 Restore EC2 from AMI

1. AWS Console > EC2 >AMIs
2. Select the AMI created in step 3.1
3. Launch Instance from Image



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: AMI-Instance

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

My AMIs: Owned by me

Amazon Machine Image (AMI)

SSM-Backup-Image
ami-03a7e9b10f2820195
2025-04-02T09:40:59.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description

Took Backup from ssm main instance-Apache Installed

Architecture	AMI ID
x86_64	ami-03a7e9b10f2820195

Summary

Number of instances: 1

Software Image (AMI): Took Backup from ssm main inst...read more
ami-03a7e9b10f2820195

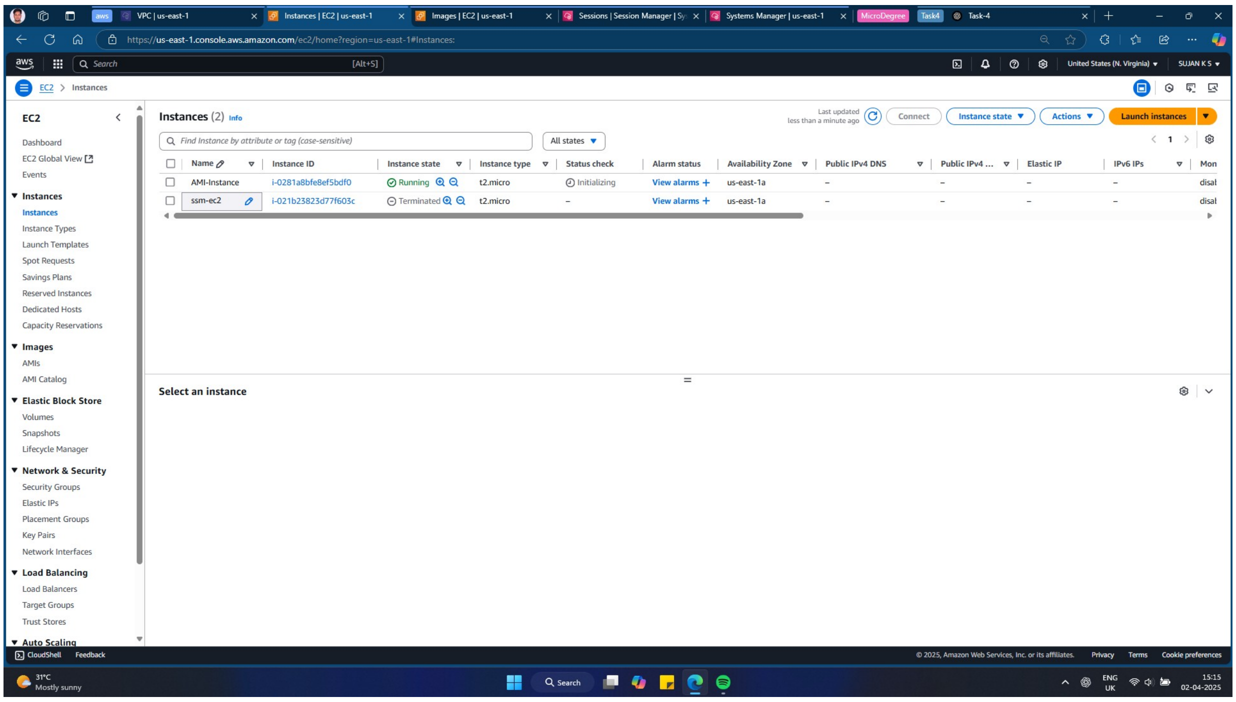
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

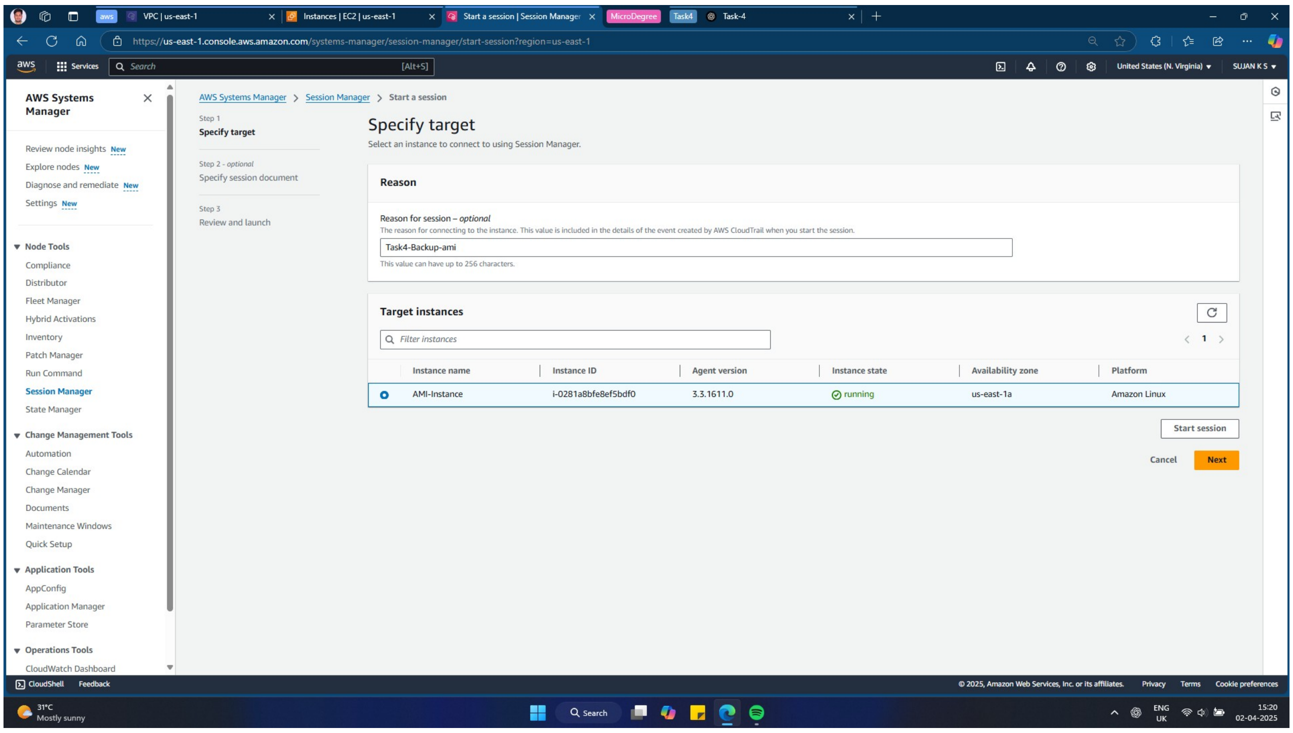
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
EC2 > Instances > Launch an instance
Recents
My AMIs
Quick Start
Shared with me
Browse more AMIs
Instance type
Cancel
Launch instance
Preview code



Instances (2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Mon
AMI-Instance	i-0281a8bfe8ef5bdf0	Running	t2.micro	Initializing	View alarms +	us-east-1a	–	–	–	–	disal
ssm-ec2	i-021b23823d77f603c	Terminated	t2.micro	–	View alarms +	us-east-1a	–	–	–	–	disal

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
EC2 > Instances
Connect
Instance state
Actions
Launch instances
All states
Select an instance



AWS Systems Manager > Session Manager > Start a session

Step 1
Specify target

Step 2 - *optional*
Specify session document

Step 3
Review and launch

Specify target

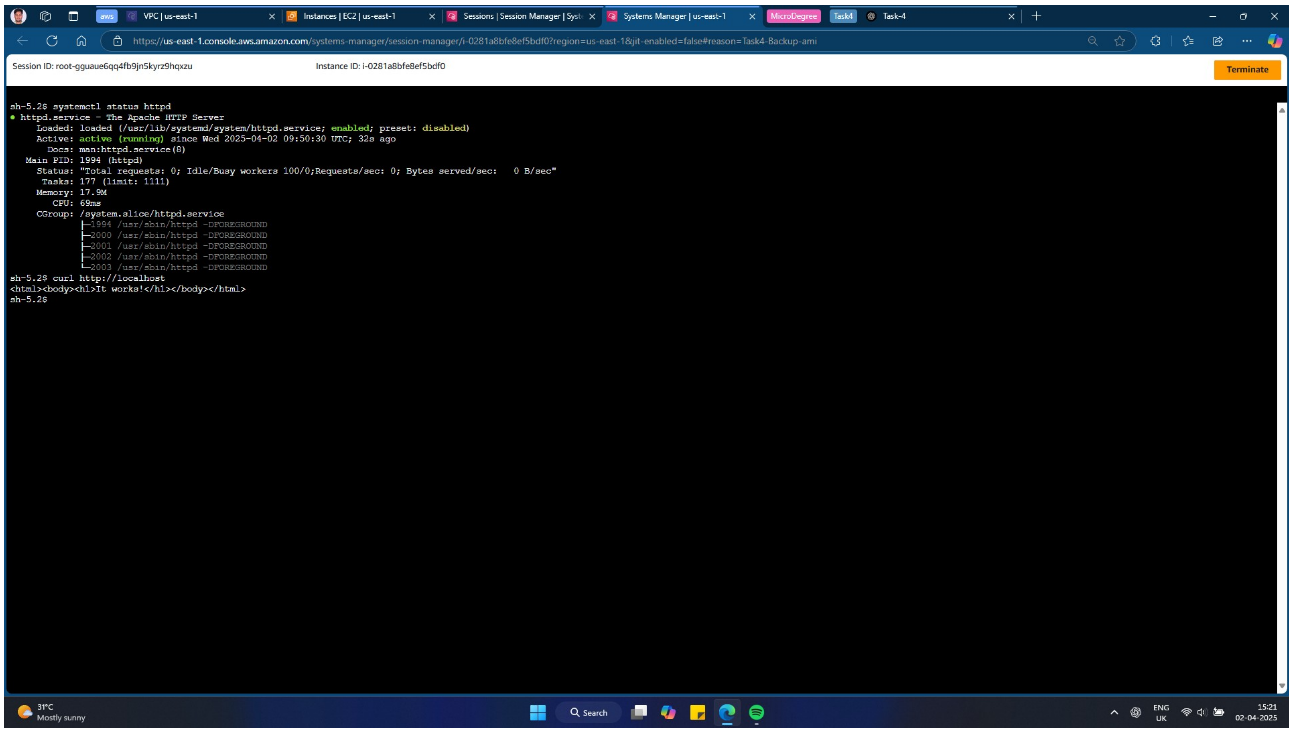
Select an instance to connect to using Session Manager.

Reason

Reason for session – *optional*
The reason for connecting to the instance. This value is included in the details of the event created by AWS CloudTrail when you start the session.
Task4-Backup-ami
This value can have up to 256 characters.

Target instances

	Instance name	Instance ID	Agent version	Instance state	Availability zone	Platform
●	AMI-Instance	i-0281a8bfe8ef5bdf0	3.3.1611.0	running	us-east-1a	Amazon Linux



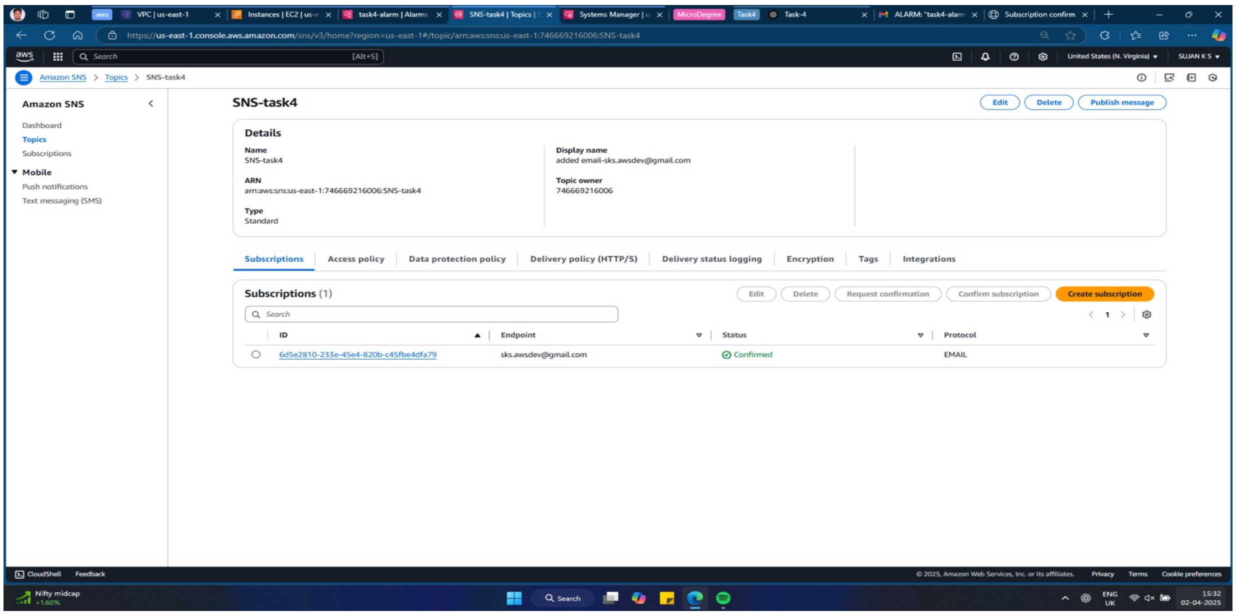
Session ID: root-gguaue6qq4fb9jn5kyrz9hqxzu   Instance ID: i-0281a8bfe8ef5bdf0

```
sh-5.2$ systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Wed 2025-04-02 09:50:30 UTC; 32s ago
       Docs: man:httpd.service(8)
   Main PID: 1994 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1111)
     Memory: 17.9M
        CPU: 69ms
     CGroup: /system.slice/httpd.service
             ├─1994 /usr/sbin/httpd -DFOREGROUND
             ├─2000 /usr/sbin/httpd -DFOREGROUND
             ├─2001 /usr/sbin/httpd -DFOREGROUND
             ├─2002 /usr/sbin/httpd -DFOREGROUND
             └─2003 /usr/sbin/httpd -DFOREGROUND
sh-5.2$ curl http://localhost
<html><body><h1>It works!</h1></body></html>
sh-5.2$
```

4. Monitoring with CloudWatch Alarm

4.1 Create SNS Topic & Subscribe for Alerts

1. AWS Console >SNS
2. Create a new SNS topic
3. Subscribe with an email address
4. Confirm subscription via email



SNS-task4

Details

Name
SNS-task4

Display name
added email-sks.awsdev@gmail.com

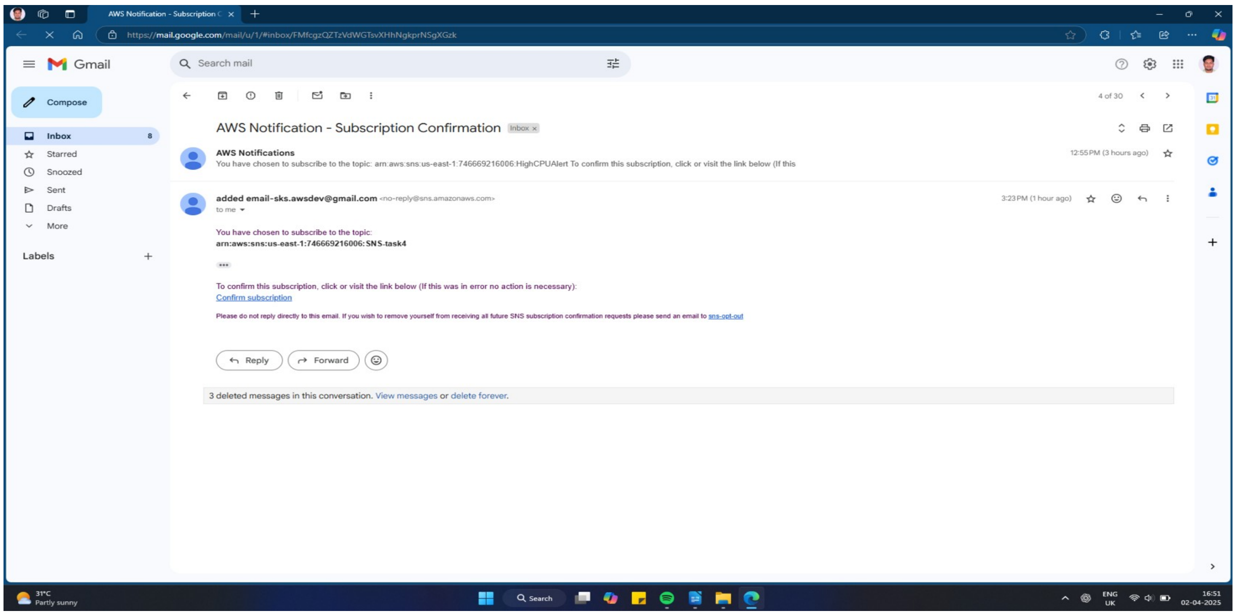
ARN
arn:aws:sns:us-east-1:746669216006:SNS-task4

Topic owner
746669216006

Type
Standard

Subscriptions (1)

ID	Endpoint	Status	Protocol
6d5e2810-233e-45e4-820b-c45fbe4dfa79	sks.awsdev@gmail.com	Confirmed	EMAIL



AWS Notification - Subscription Confirmation Inbox

AWS Notifications 12:55 PM (3 hours ago)
You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:746669216006:HighCPUAlert To confirm this subscription, click or visit the link below (If this

added email-sks.awsdev@gmail.com <no-reply@sns.amazonaws.com> 3:23 PM (1 hour ago)
to me

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:746669216006:SNS-task4

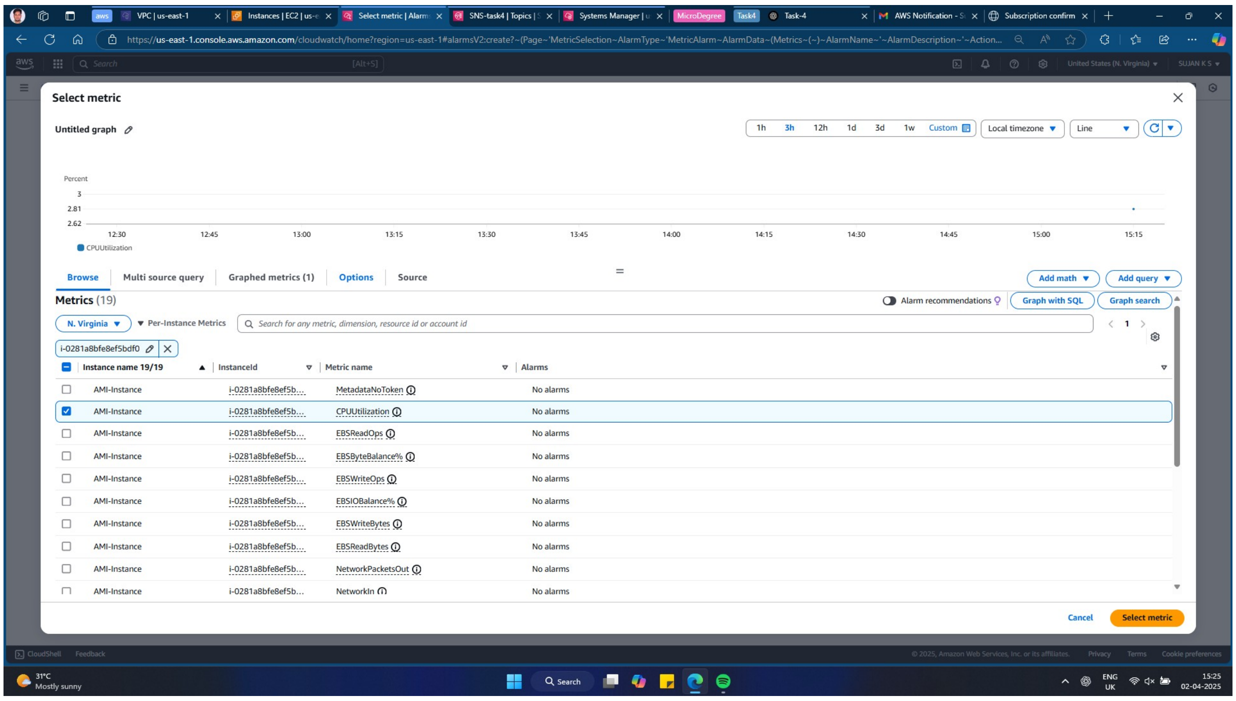
To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

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

3 deleted messages in this conversation. View messages or delete forever.

4.2 Set Up CloudWatch Alarm for CPU Utilization > 80%

1. AWS Console > CloudWatch > Alarms > Create Alarm
2. Select EC2 Instance and CPU Utilization metric (selected using my instance ID)
3. Set the threshold to 80%
4. Select SNS topic for notifications > create



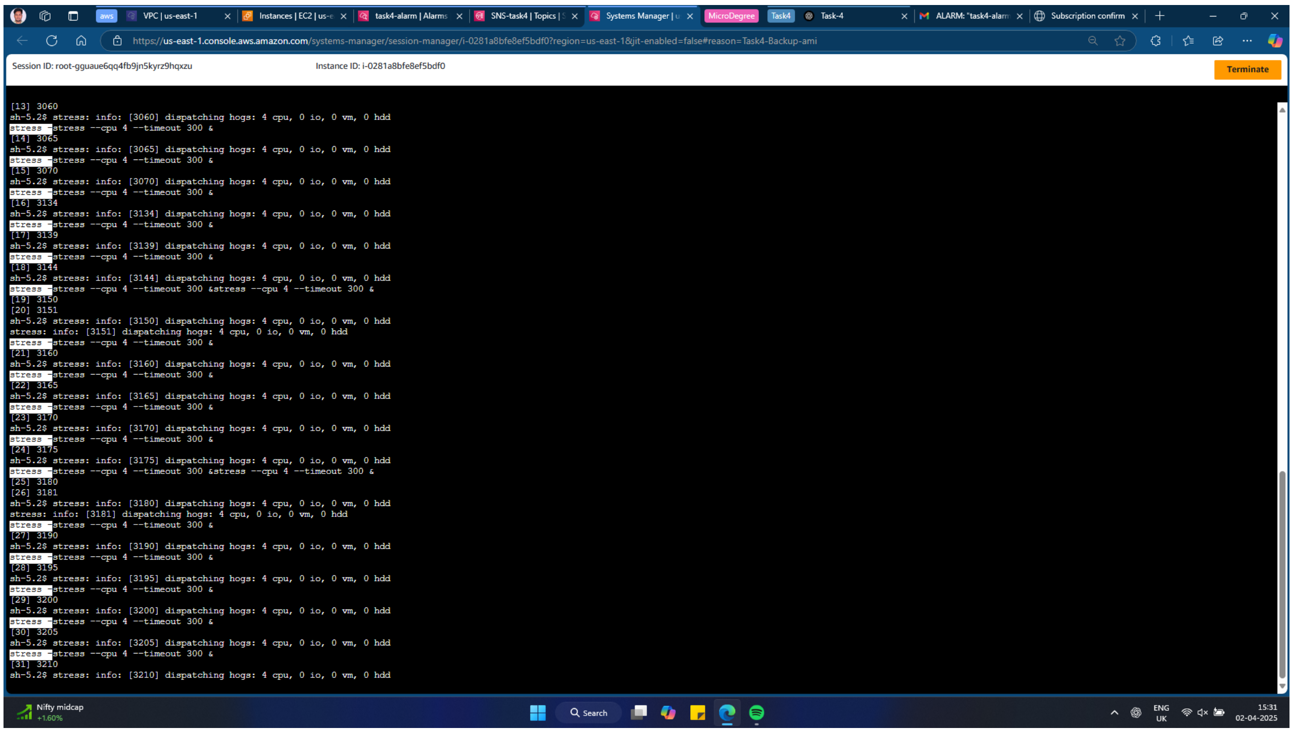
Select metric

Instance name 19/19	InstanceId	Metric name	Alarms
AMI-Instance	i-0281a8bfe8ef5b...	MetadataNoToken	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	CPUUtilization	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	EBSReadOps	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	EBSByteBalance%	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	EBSWriteOps	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	EBSIOBalance%	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	EBSWriteBytes	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	EBSReadBytes	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	NetworkPacketsOut	No alarms
AMI-Instance	i-0281a8bfe8ef5b...	NetworkIn	No alarms

5. Simulating high CPU load

5.1 Install and Run CPU Stress Test

```
sudo yum install-y epel-release
sudo yum install-y stress
stress--cpu 4--timeout 300 &
```

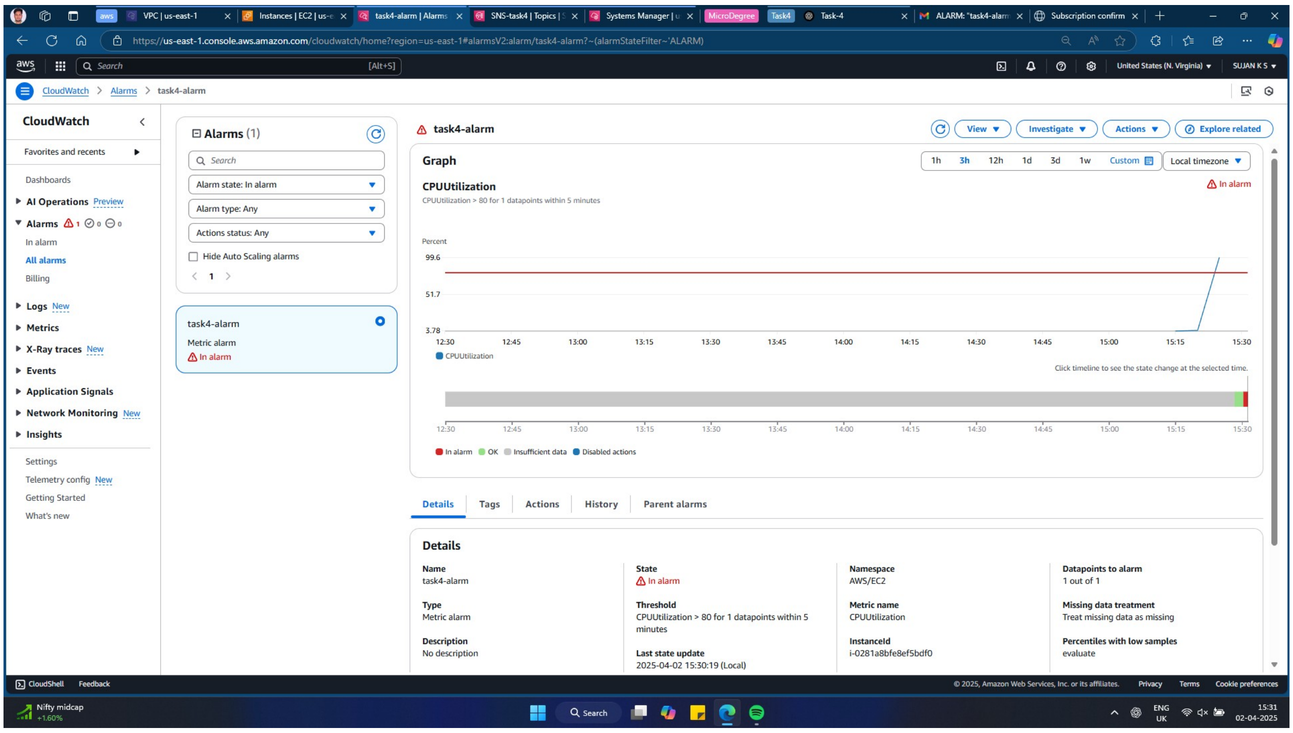


Session ID: root-gguaue6qq4fb9jn5kyrz9hqxzu   Instance ID: i-0281a8bfe8ef5bdf0

```
[13] 3060
sh-5.2$ stress: info: [3060] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[14] 3065
sh-5.2$ stress: info: [3065] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[15] 3070
sh-5.2$ stress: info: [3070] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[16] 3134
sh-5.2$ stress: info: [3134] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[17] 3139
sh-5.2$ stress: info: [3139] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[18] 3144
sh-5.2$ stress: info: [3144] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &stress --cpu 4 --timeout 300 &
[19] 3150
[20] 3151
sh-5.2$ stress: info: [3150] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress: info: [3151] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[21] 3160
sh-5.2$ stress: info: [3160] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[22] 3165
sh-5.2$ stress: info: [3165] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[23] 3170
sh-5.2$ stress: info: [3170] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[24] 3175
sh-5.2$ stress: info: [3175] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &stress --cpu 4 --timeout 300 &
[25] 3180
[26] 3181
sh-5.2$ stress: info: [3180] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress: info: [3181] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[27] 3190
sh-5.2$ stress: info: [3190] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[28] 3195
sh-5.2$ stress: info: [3195] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[29] 3200
sh-5.2$ stress: info: [3200] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[30] 3205
sh-5.2$ stress: info: [3205] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress -stress --cpu 4 --timeout 300 &
[31] 3210
sh-5.2$ stress: info: [3210] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
```

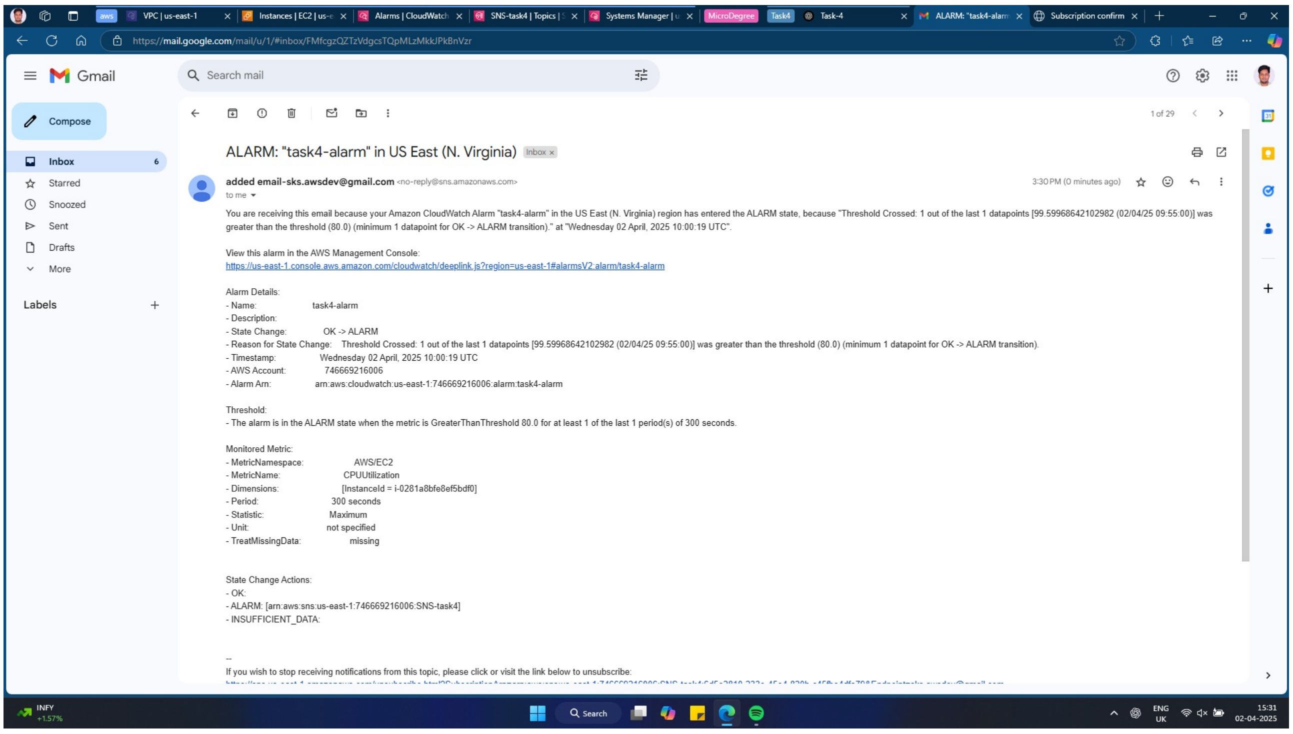
5.2 Verify CloudWatch Alarm

- AWS Console >CloudWatch >Alarms
- alarm triggers when CPU > 80%
- Check if SNS Notification is received



Details

Name	State	Namespace	Datapoints to alarm
task4-alarm	In alarm	AWS/EC2	1 out of 1
Type	**Threshold**	**Metric name**	**Missing data treatment**
Metric alarm	CPUUtilization > 80 for 1 datapoints within 5 minutes	CPUUtilization	Treat missing data as missing
Description	**Last state update**	**InstanceId**	**Percentiles with low samples**
No description	2025-04-02 15:30:19 (Local)	i-0281a8bfe8ef5bdf0	evaluate



ALARM: "task4-alarm" in US East (N. Virginia)

added email-sks.awsdev@gmail.com <no-reply@sns.amazonaws.com>
to me
3:30 PM (0 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "task4-alarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [99.59968642102982 (02/04/25 09:55:00)] was greater than the threshold (80.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Wednesday 02 April, 2025 10:00:19 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/task4-alarm

```
Alarm Details:
- Name:                  task4-alarm
- Description:
- State Change:          OK -> ALARM
- Reason for State Change:    Threshold Crossed: 1 out of the last 1 datapoints [99.59968642102982 (02/04/25 09:55:00)] was greater than the threshold (80.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp:             Wednesday 02 April, 2025 10:00:19 UTC
- AWS Account:           746669216006
- Alarm Arn:             arn:aws:cloudwatch:us-east-1:746669216006:alarm:task4-alarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 80.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:
- MetricNamespace:            AWS/EC2
- MetricName:                 CPUUtilization
- Dimensions:                 [InstanceId = i-0281a8bfe8ef5bdf0]
- Period:                     300 seconds
- Statistic:                  Maximum
- Unit:                       not specified
- TreatMissingData:           missing


State Change Actions:
- OK:
- ALARM: [arn:aws:sns:us-east-1:746669216006:SNS-task4]
- INSUFFICIENT_DATA:
```

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

Conclusion

In this task, we successfully set up a custom VPC with public and private subnets, configured networking components such as Internet Gateway, NAT Gateway, and Route Tables, and deployed an EC2 instance in a private subnet using SSM for secure access.

We installed and tested the Apache web server, created EC2 backups using AMI, and demonstrated the restoration process by launching a new instance from the backup. Additionally, we configured CloudWatch Alarms to monitor CPU utilization and tested the alert mechanism using a CPU stress test.

This hands-on implementation reinforced key AWS concepts such as networking, secure EC2 access, backup & recovery, and monitoring, ensuring a strong foundation in AWS infrastructure management.

Submitted by: Sujan K S

Submitted on: April 2, 2025