11.ABOUT PROJECT

11.1 –Project Introduction:

Auto Scaling with EC2:

The project "Auto Scaling with EC2" focuses on building a highly available and scalable web infrastructure using AWS services. The system dynamically adjusts computing resources based on traffic demands through **Auto Scaling** and ensures traffic distribution across instances with **Elastic Load Balancing (ELB)**. Additionally, **SNS (Simple Notification Service)** is integrated to send notifications about scaling events. To assess the system's robustness, a **stress test** is conducted to evaluate its ability to handle high traffic loads efficiently.

11.2 – Project Requirements:

Amazon EC2 (Elastic Compute Cloud):

- **Set up EC2 instances with custom Amazon Machine Images (AMIs).**
- **Configure Launch Configurations or Launch Templates for automated instance provisioning.**

Auto Scaling:

- **Define an Auto Scaling Group (ASG) with appropriate scaling policies.**
- **Set up triggers to automatically scale out/in based on CPU utilization or traffic load.**

Elastic Load Balancer (ELB):

- **Implement Application Load Balancer (ALB) or Network Load Balancer (NLB) to distribute traffic evenly across EC2 instances.**
- **Ensure fault tolerance by directing traffic to healthy instances only.**

Amazon SNS (Simple Notification Service):

- **Configure SNS for real-time notifications when scaling actions occur (e.g., instance launch or termination).**
- **Subscribe email or SMS endpoints to receive alerts.**

Stress Testing:

- **Perform stress tests using tools like Mobaxterm to simulate high loads on the infrastructure.**
- **Monitor system performance and ensure Auto Scaling effectively handles the increase in traffic.**

Monitoring and Logging:

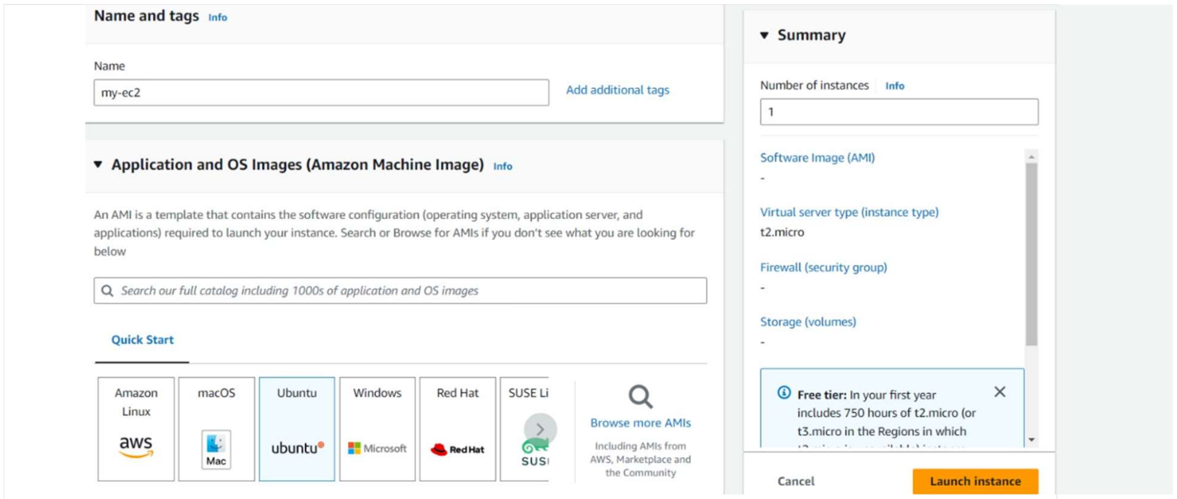
- **Use CloudWatch to monitor system performance (CPU, network, etc.) and trigger scaling events.**
- **Enable logging to track scaling events and load balancer activity.**

Security:

- **Ensure security using Security Groups and IAM roles for EC2 instances.**
- **Use VPC for network isolation and configure subnets for different availability zones.**

We use Azure Bot Service to create, configure, and host the chatbot, making it accessible via various
channels like web, mobile apps, and messaging platforms. Azure Bot Service simplifies
bot deployment, scalability, and maintenance.

Step 1: Go to the **AWS portal** and login to your account then create a AMI and launch a ec2 instance



Name and tags Info

Name

my-ec2

Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances Info

1

Software Image (AMI)

-

Virtual server type (instance type)

t2.micro

Firewall (security group)

-

Storage (volumes)

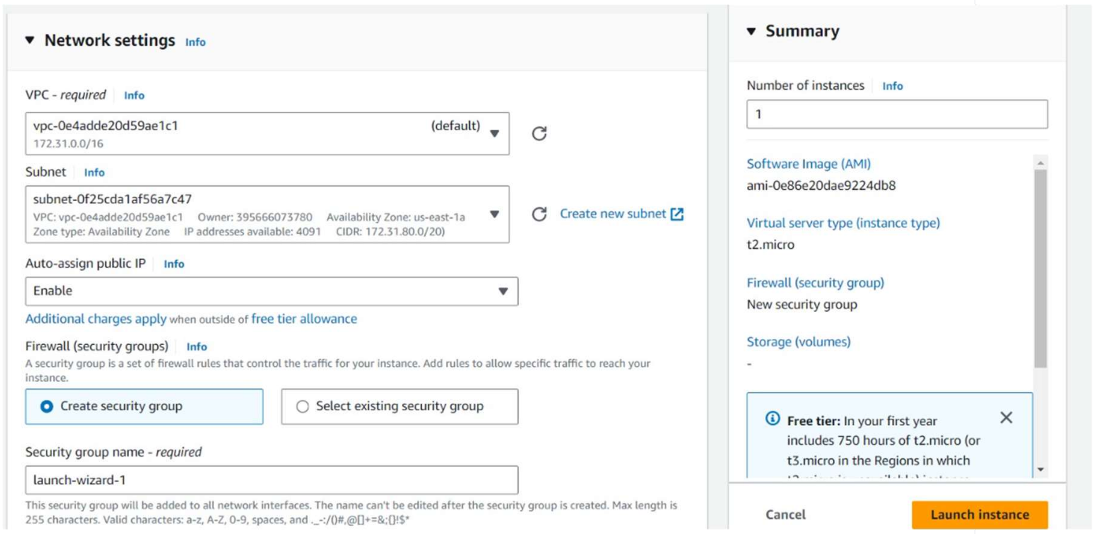
-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

Cancel

Launch instance

Fig-1



Network settings Info

VPC - *required* Info

vpc-0e4adde20d59ae1c1 (default)
172.31.0.0/16

Subnet Info

subnet-0f25cda1af56a7c47
VPC: vpc-0e4adde20d59ae1c1 Owner: 395666073780 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.80.0/20)

Create new subnet

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[✓ Create security group] Select existing security group

Security group name - *required*

launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Summary

Number of instances Info

1

Software Image (AMI)

ami-0e86e20dae9224db8

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

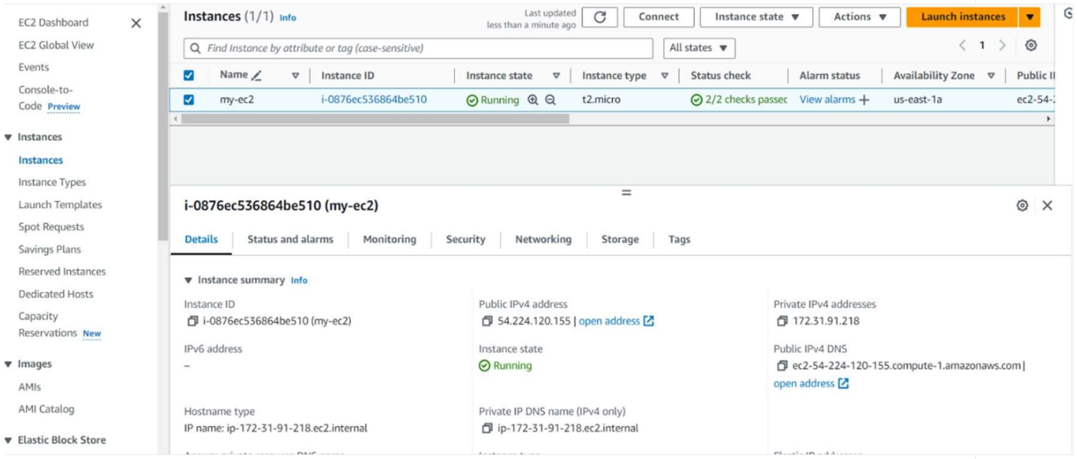
-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

Cancel

Launch instance

Fig-2



EC2 Dashboard
EC2 Global View
Events
Console-to-Code Preview

▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New

▼ Images
AMIs
AMI Catalog

▼ Elastic Block Store

Instances (1/1) Info — Last updated less than a minute ago — Connect | Instance state ▼ | Actions ▼ | Launch instances ▼

Find Instance by attribute or tag (case-sensitive) — All states ▼

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☑	my-ec2	i-0876ec536864be510	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1a	ec2-54-

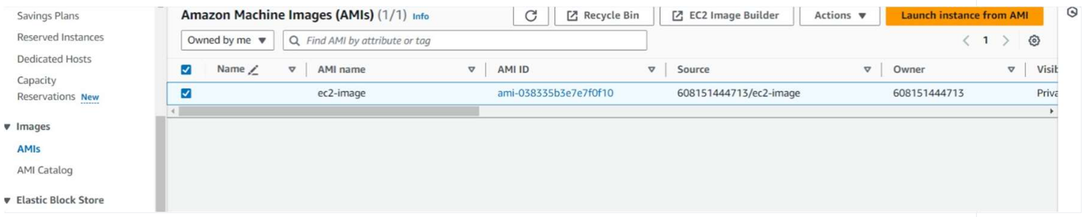
i-0876ec536864be510 (my-ec2)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary Info

Instance ID i-0876ec536864be510 (my-ec2)	Public IPv4 address 54.224.120.155 | open address	Private IPv4 addresses 172.31.91.218
IPv6 address –	Instance state ✓ Running	Public IPv4 DNS ec2-54-224-120-155.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-91-218.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-91-218.ec2.internal	

Fig-3



Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New

▼ Images
AMIs
AMI Catalog

▼ Elastic Block Store

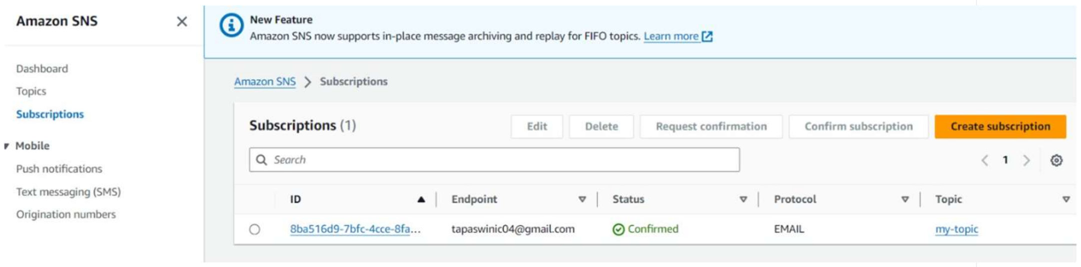
Amazon Machine Images (AMIs) (1/1) Info — Recycle Bin | EC2 Image Builder | Actions ▼ | Launch instance from AMI

Owned by me ▼ — Find AMI by attribute or tag

	Name	AMI name	AMI ID	Source	Owner	Visib
☑		ec2-image	ami-038335b3e7e7f0f10	608151444713/ec2-image	608151444713	Priva

Fig-4

Step2: Create SNS topic and subscribe to your mail



Amazon SNS

Dashboard
Topics
Subscriptions

▼ Mobile
Push notifications
Text messaging (SMS)
Origination numbers

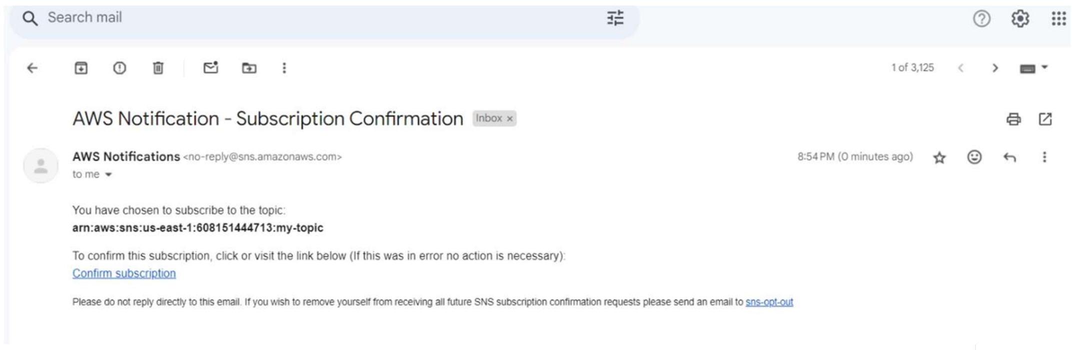
New Feature
Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

Amazon SNS > Subscriptions

Subscriptions (1) — Edit | Delete | Request confirmation | Confirm subscription | Create subscription

Search

	ID	Endpoint	Status	Protocol	Topic
○	8ba516d9-7bfc-4cce-8fa...	tapaswinic04@gmail.com	✓ Confirmed	EMAIL	my-topic



AWS Notification - Subscription Confirmation

AWS Notifications <no-reply@sns.amazonaws.com>
to me
8:54 PM (0 minutes ago)

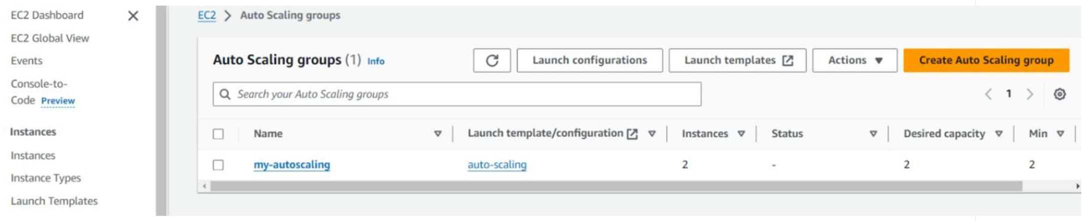
You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:608151444713:my-topic

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

Fig-6

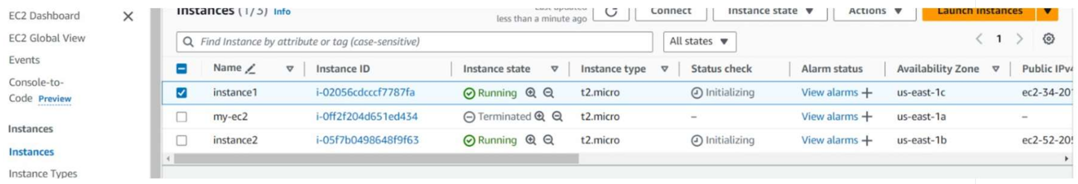
Step 3: Now configure the auto- scaling groups and load-balancing, health checks and also capacity



Name	Launch template/configuration	Instances	Status	Desired capacity	Min
my-autoscaling	auto-scaling	2	-	2	2

Fig-7

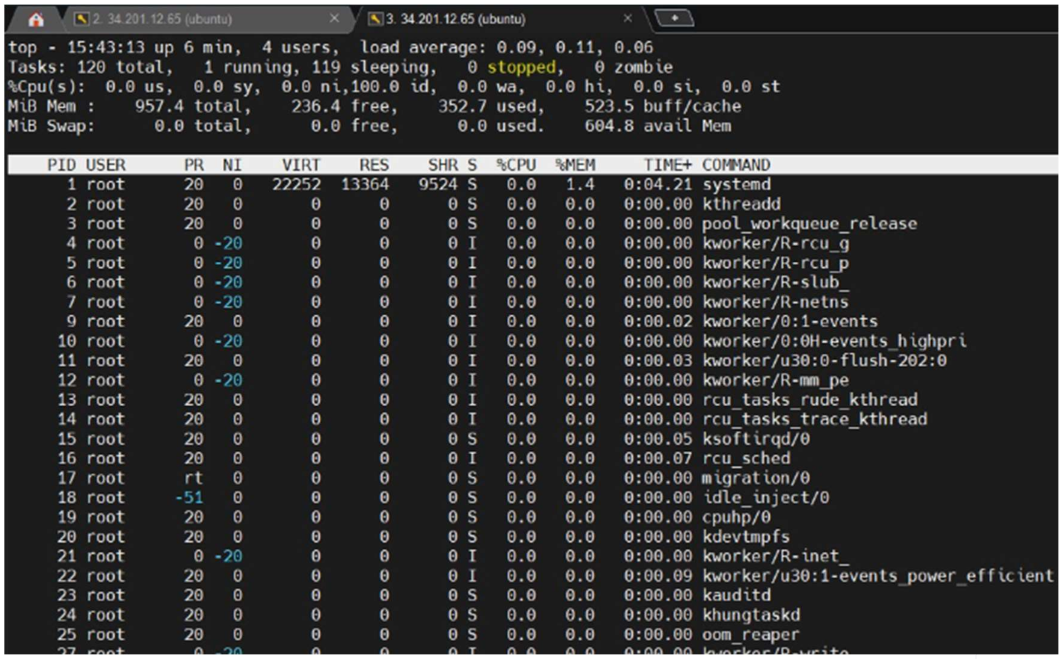
Step 4: As the result of Auto scaling the instances are increased according to capacity



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
instance1	i-02056cdcccf7787fa	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-34-20
my-ec2	i-0ff2f204d651ed434	Terminated	t2.micro	–	View alarms +	us-east-1a	–
instance2	i-05f7b0498648f9f63	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-52-20

Fig-8

Step 5: Connect to Mobaxterm and check the CPU utilization.



```
top - 15:43:13 up 6 min,  4 users,  load average: 0.09, 0.11, 0.06
Tasks: 120 total,   1 running, 119 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
MiB Mem :    957.4 total,    236.4 free,    352.7 used,    523.5 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    604.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0   22252  13364   9524 S   0.0   1.4   0:04.21 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
      9 root      20   0       0      0      0 I   0.0   0.0   0:00.02 kworker/0:1-events
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.03 kworker/u30:0-flush-202:0
     12 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_pe
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.05 ksoftirqd/0
     16 root      20   0       0      0      0 I   0.0   0.0   0:00.07 rcu_sched
     17 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     18 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-inet_
     22 root      20   0       0      0      0 I   0.0   0.0   0:00.09 kworker/u30:1-events_power_efficient
     23 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
     24 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
     25 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-write
```

Fig-9

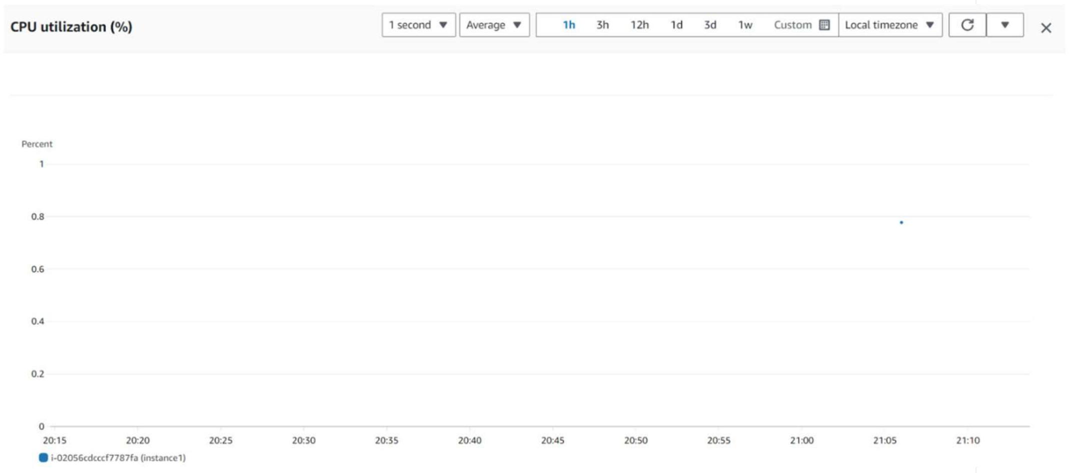
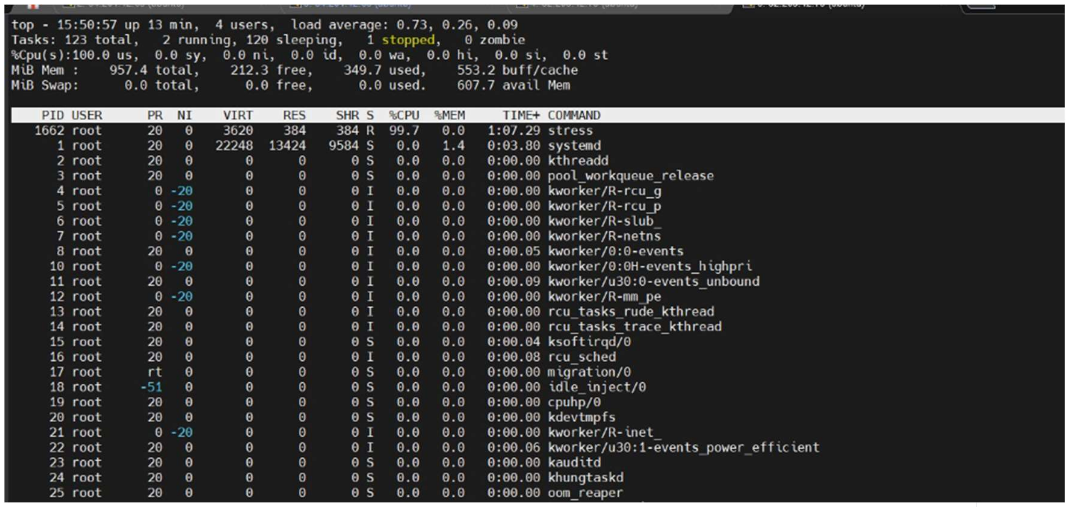


Fig 10

Step 6: After the stress test, the CPU utilization increased.



```
top - 15:50:57 up 13 min,  4 users,  load average: 0.73, 0.26, 0.09
Tasks: 123 total,   2 running, 120 sleeping,   1 stopped,   0 zombie
%Cpu(s):100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    957.4 total,    212.3 free,    349.7 used,    553.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    607.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1662 root      20   0    3620    384    384 R  99.7   0.0   1:07.29 stress
      1 root      20   0   22248  13424   9584 S   0.0   1.4   0:03.80 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
      8 root      20   0       0      0      0 I   0.0   0.0   0:00.05 kworker/0:0-events
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.09 kworker/u30:0-events_unbound
     12 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_pe
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.04 ksoftirqd/0
     16 root      20   0       0      0      0 I   0.0   0.0   0:00.08 rcu_sched
     17 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     18 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-inet_
     22 root      20   0       0      0      0 I   0.0   0.0   0:00.06 kworker/u30:1-events_power_efficient
     23 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
     24 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
     25 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
```

Fig 11

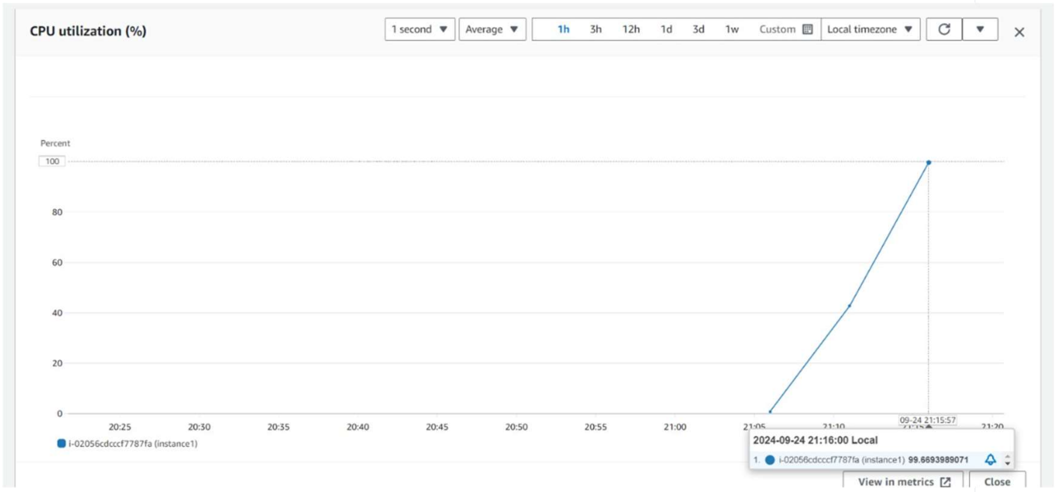
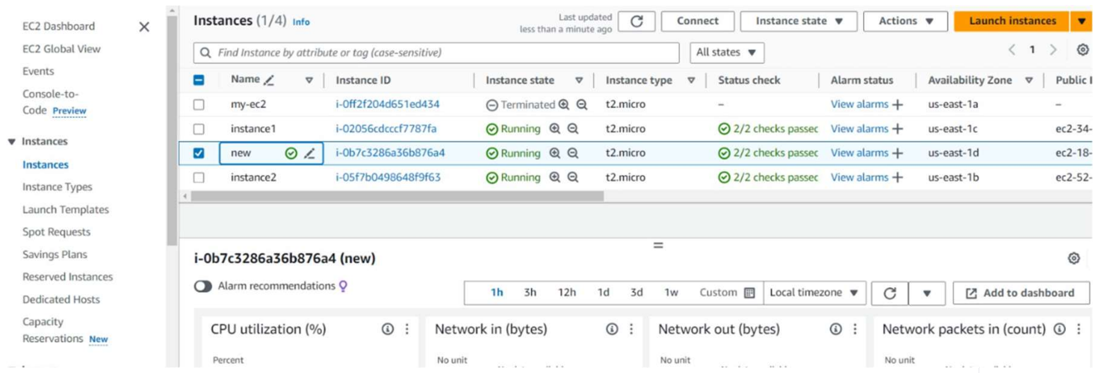


Fig-12

Step 7: As the result one more instance increased according to the Auto scaling capacity.



EC2 Dashboard
EC2 Global View
Events
Console-to-Code Preview
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New

Instances (1/4) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☐	my-ec2	i-0ff2f204d651ed434	Terminated	t2.micro	–	View alarms +	us-east-1a	–
☐	instance1	i-02056cdcccf7787fa	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-34-
☑	new	i-0b7c3286a36b876a4	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1d	ec2-18-
☐	instance2	i-05f7b0498648f9f63	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-52-

i-0b7c3286a36b876a4 (new)

Alarm recommendations

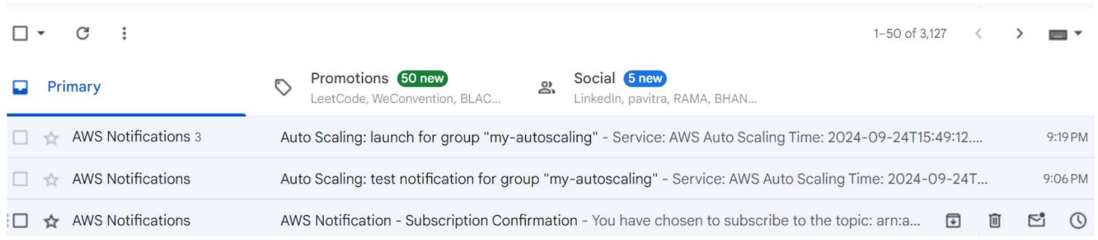
1h 3h 12h 1d 3d 1w Custom Local timezone

Add to dashboard

CPU utilization (%) — Percent
Network in (bytes) — No unit
Network out (bytes) — No unit
Network packets in (count) — No unit

Fig-13

Step 9: These are the AWS notifications we get through the process.



1–50 of 3,127

Primary | Promotions 50 new — LeetCode, WeConvention, BLAC... | Social 5 new — LinkedIn, pavitra, RAMA, BHAN...

AWS Notifications 3	Auto Scaling: launch for group "my-autoscaling" - Service: AWS Auto Scaling Time: 2024-09-24T15:49:12....	9:19 PM
AWS Notifications	Auto Scaling: test notification for group "my-autoscaling" - Service: AWS Auto Scaling Time: 2024-09-24T...	9:06 PM
AWS Notifications	AWS Notification - Subscription Confirmation - You have chosen to subscribe to the topic: arn:a...	

Fig-14