Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

AWS CLOUD COMPUTING

CONFIGURE AMAZON AUTO SCALING WITH LOAD BALANCER AND SNAPSHOT

Andhra Pradesh State Skill Development Corporation (APSSDC)

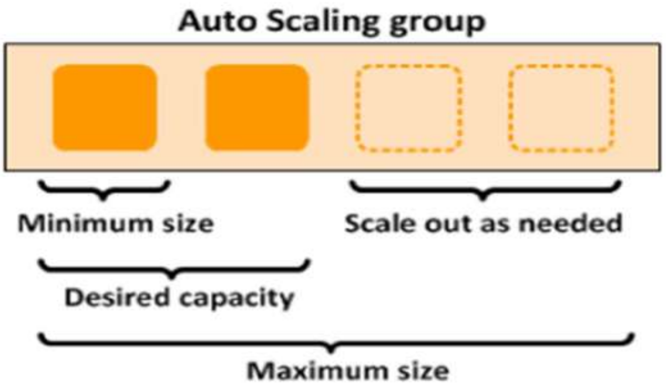
Configure Amazon Auto Scaling with load balancer and snapshot

Amazon EC2 Auto Scaling

To configure Amazon Auto Scaling with load balancer and snapshot

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of 1 instance, a desired capacity of 2 instances, and a maximum size of 4 instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



Auto Scaling Component

The following table describes the key components of Amazon EC2 Auto Scaling.

Groups

Your EC2 instances are organized into *groups* so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and desired number of EC2 instances.

Launch configurations

Your group uses a *launch configuration* as a template for its EC2 instances. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.

Scaling options

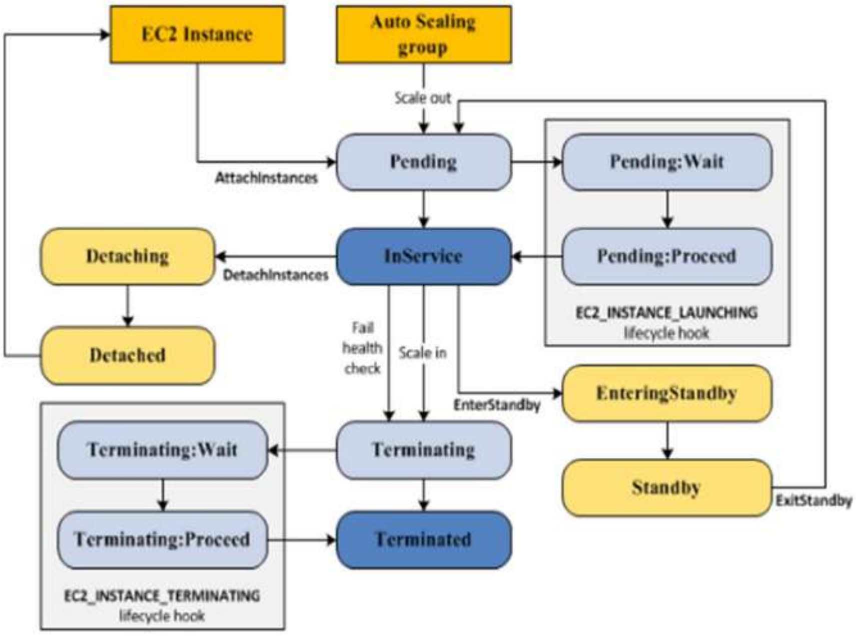
Amazon EC2 Auto Scaling provides several ways for you to scale your Auto Scaling groups. For example, you can configure a group to scale based on the occurrence of specified conditions (dynamic scaling) or on a schedule.

Auto Scaling Lifecycle

The EC2 instances in an Auto Scaling group have a path, or lifecycle, that differs from that of other EC2 instances. The lifecycle starts when the Auto Scaling group launches an instance and puts it into service. The lifecycle ends when you terminate the instance,or the Auto Scaling group takes the instance out of service and terminates it.

Note : You are billed for instances as soon as they are launched, including the time that they are not yet in service.

The following illustration shows the transitions between instance states in the Amazon EC2 Auto Scaling lifecycle.



PRE-REQUISITES

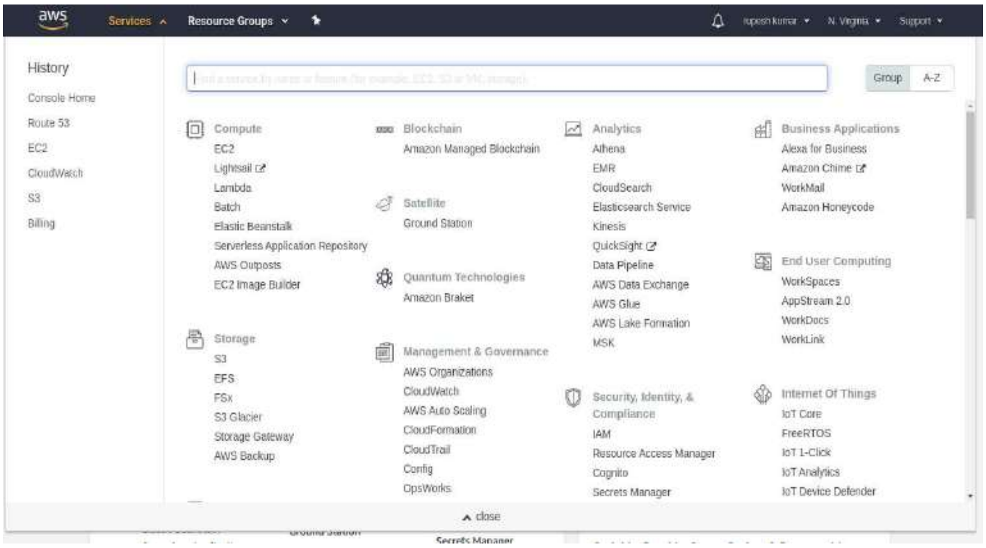
You should have AWS account, or IAM user with EC2 Full access

Task

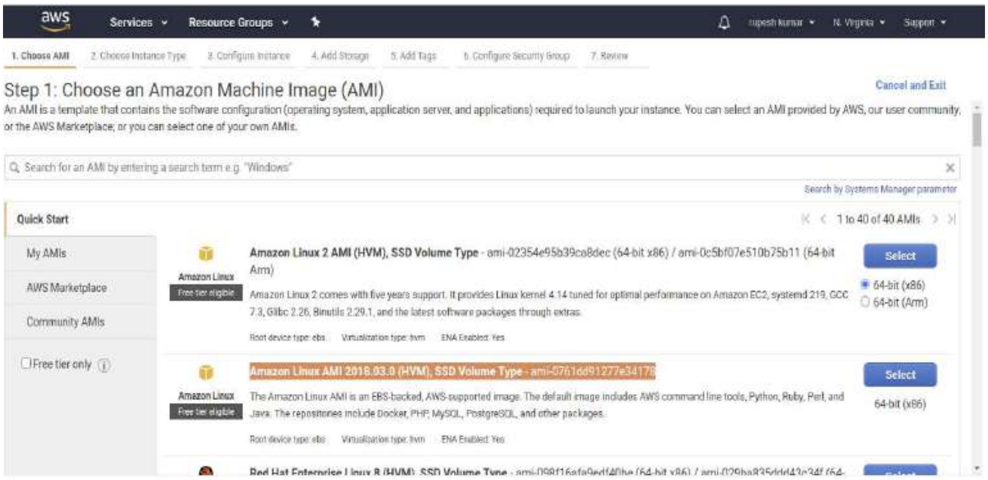
1. **Launch Amazon linux instance**
2. **Configure web server**
3. **Stop the instance**
4. **Create AMI image of above instance**
5. **Configure Auto Scaling launch configuration and Auto Scaling group**
6. **Configure load balancer auto scaling**

Steps :

From the aws management console, click on services and select ec2 service from the compute section.



Launch an Amazon linux instance and Configure web server like NGINX



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-02354e95b39ca8dec (64-bit x86) / ami-0c5bf07e510b75b11 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0761dd91277e34178

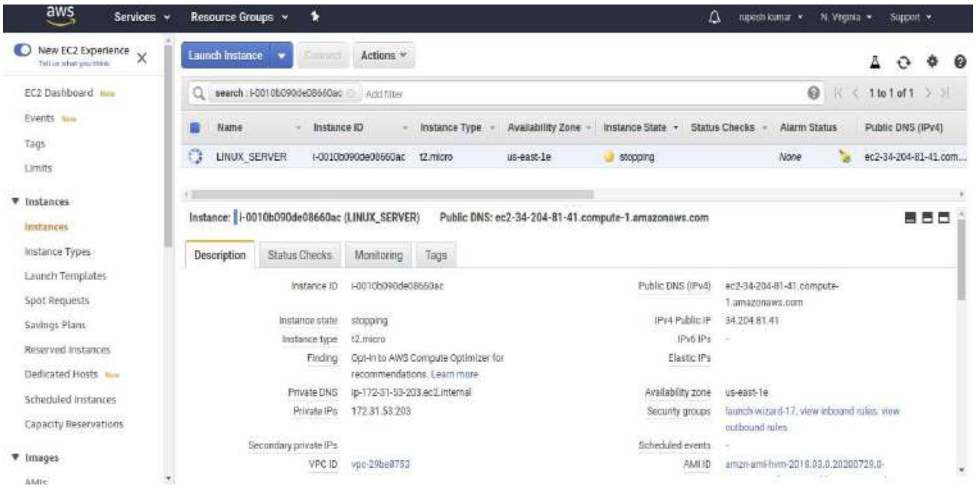
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (x86)

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-098f16afa9edf40be (64-bit x86) / ami-029ba835ddd43c34f (64-

Stop the instance : To create AMI from this instance on ‘EC2 Dashboard Panel”. select the instance and click on action and select the instance state then stop.



search : i-0010b090de08660ac

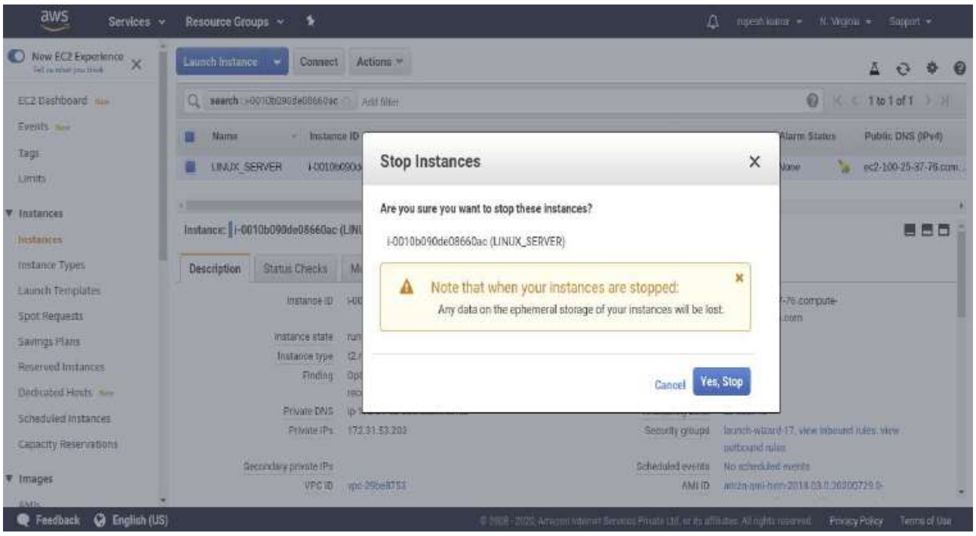
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
LINUX_SERVER	i-0010b090de08660ac	t2.micro	us-east-1e	stopping		None	ec2-34-204-81-41.com...

Instance: i-0010b090de08660ac (LINUX_SERVER) Public DNS: ec2-34-204-81-41.compute-1.amazonaws.com

Instance ID i-0010b090de08660ac
Instance state stopping
Instance type t2.micro
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-172-31-53-203.ec2.internal
Private IPs 172.31.53.203
Secondary private IPs
VPC ID vpc-29be8753

Public DNS (IPv4) ec2-34-204-81-41.compute-1.amazonaws.com
IPv4 Public IP 34.204.81.41
IPv6 IPs -
Elastic IPs
Availability zone us-east-1e
Security groups launch-wizard-17. view inbound rules. view outbound rules
Scheduled events -
AMI ID amzn-ami-hvm-2018.03.0.20200729.0-

Click on Yes, Stop.



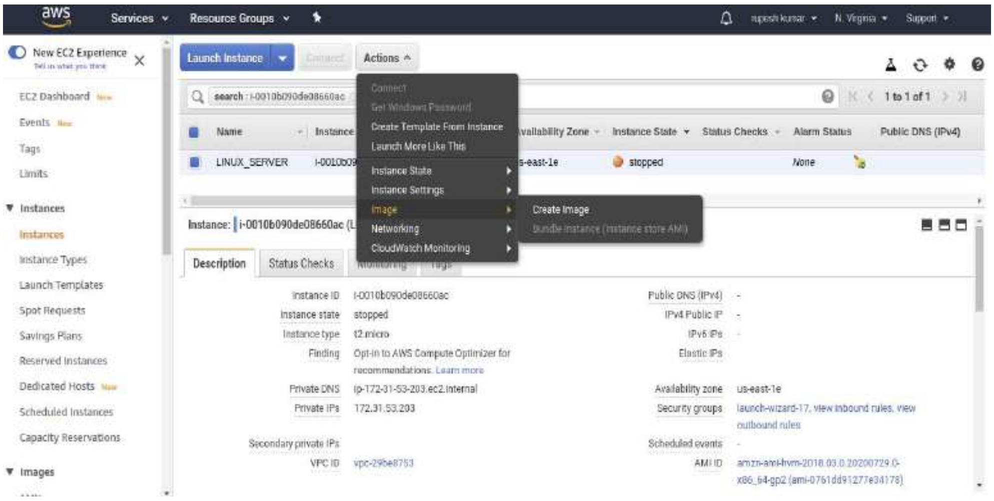
Stop Instances

Are you sure you want to stop these instances?

i-0010b090de08660ac (LINUX_SERVER)

Note that when your instances are stopped:
Any data on the ephemeral storage of your instances will be lost.

Select the stopped instance and click on action button
Select Image and then click on create image button



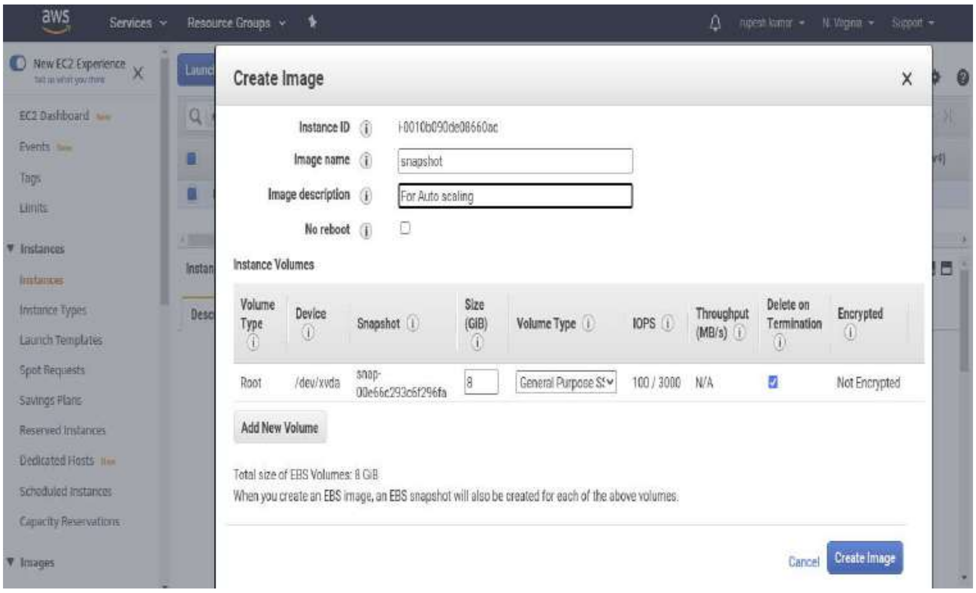
Name	Instance	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
LINUX_SERVER	i-0010b09	us-east-1e	stopped		None	

Instance: i-0010b090de08660ac

Instance ID i-0010b090de08660ac
Instance state stopped
Instance type t2.micro
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-172-31-53-203.ec2.internal
Private IPs 172.31.53.203
Secondary private IPs
VPC ID vpc-29be8753

Public DNS (IPv4) -
IPv4 Public IP -
IPv6 IPs -
Elastic IPs
Availability zone us-east-1e
Security groups launch-wizard-17, view inbound rules, view outbound rules
Scheduled events -
AMI ID amzn-ami-hvm-2018.03.0.20200729.0-x86_64-gp2 (ami-0761dd91277e34178)

Click on create Image



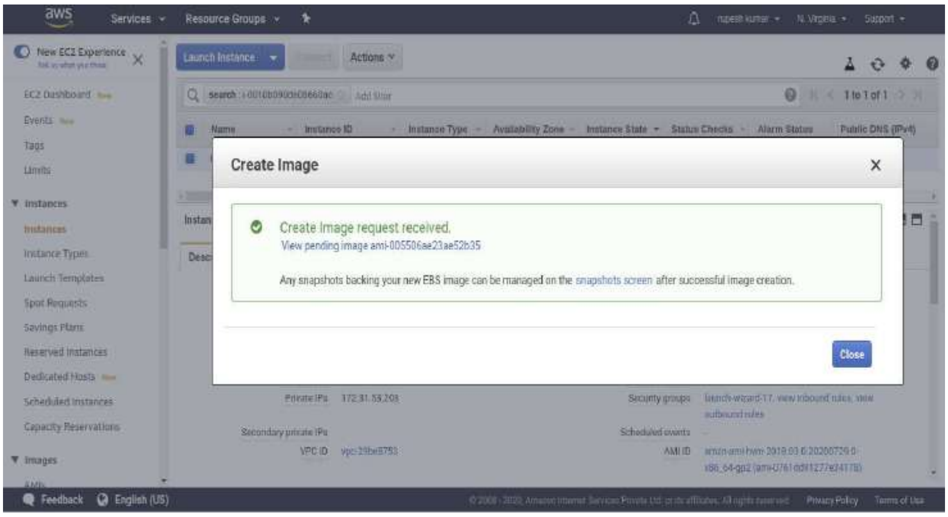
Instance ID i-0010b090de08660ac
Image name snapshot
Image description For Auto scaling
No reboot □

Instance Volumes

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-00e66c293c6f296fa	8	General Purpose SS	100 / 3000	N/A	☑	Not Encrypted

Total size of EBS Volumes: 8 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Now, click on Close

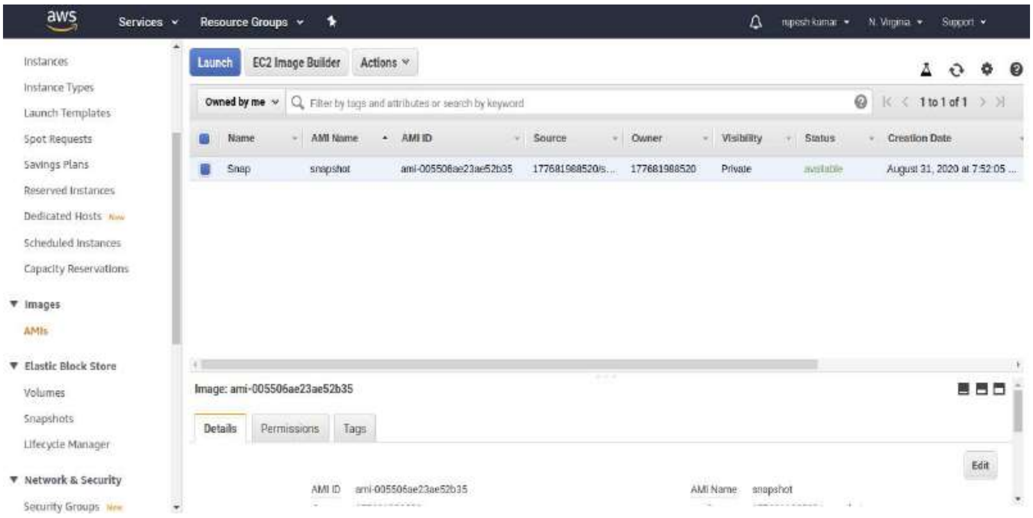


Create Image

Create Image request received.
View pending image ami-005506ae23ae52b35

Any snapshots backing your new EBS image can be managed on the snapshots screen after successful image creation.

Verify AMI is created or not in the EC2 Dashboard panel. On the EC2 Dashboard panel, Select Images and Click on AMI's then check status is on Available state or Not.



Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
Snap	snapshot	ami-005506ae23ae52b35	177681988520/s...	177681988520	Private	available	August 31, 2020 at 7:52:05 ...

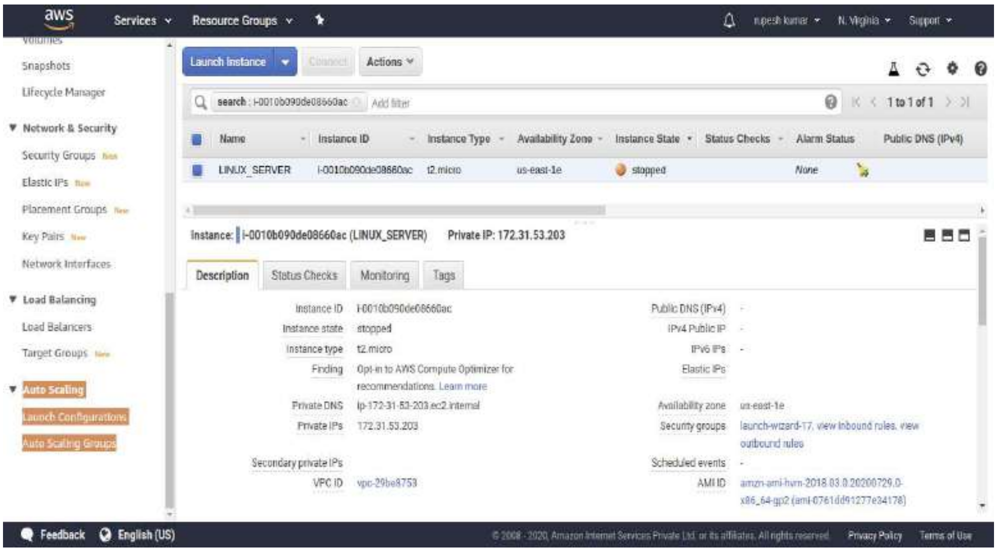
Image: ami-005506ae23ae52b35

AMI ID ami-005506ae23ae52b35 AMI Name snapshot

Configure Auto Scaling launch configuration and Auto Scaling group

To configure Auto Scaling Group on the Ec2 Dashboard Panel

Select Auto Scaling



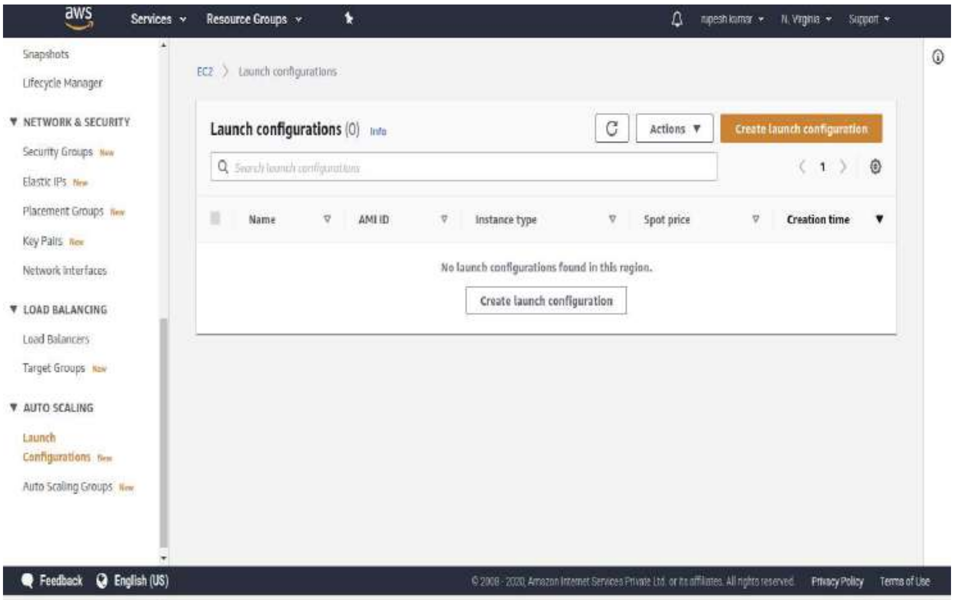
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
LINUX_SERVER	i-0010b090de08660ac	t2.micro	us-east-1e	stopped		None	

Instance: i-0010b090de08660ac (LINUX_SERVER) Private IP: 172.31.53.203

Instance ID i-0010b090de08660ac
Instance state stopped
Instance type t2.micro
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-172-31-53-203.ec2.internal
Private IPs 172.31.53.203
Secondary private IPs
VPC ID vpc-29be8753

Public DNS (IPv4) -
IPv4 Public IP -
IPv6 IPs -
Elastic IPs
Availability zone us-east-1e
Security groups launch-wizard-17. view inbound rules. view outbound rules
Scheduled events -
AMI ID amzn-ami-hvm-2018.03.0.20200729.0-x86_64-gp2 (ami-0761dd91277e34178)

Click on Launch Configurations

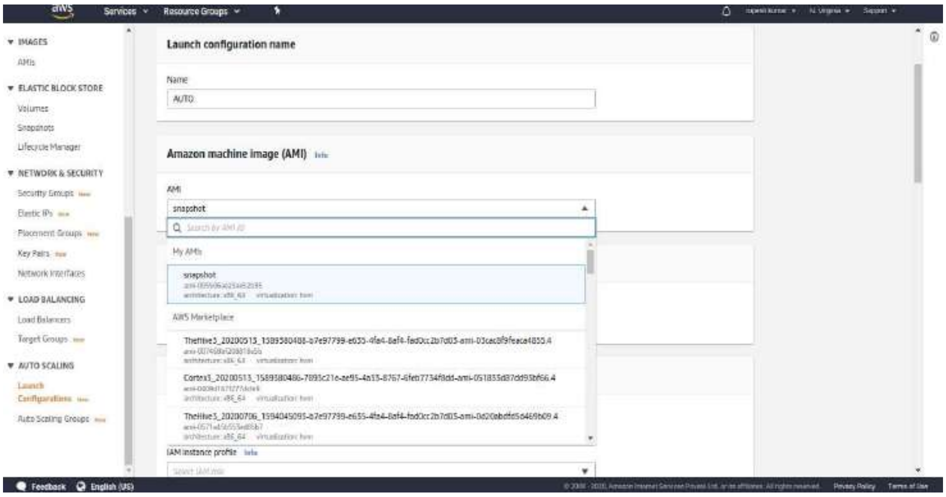


Launch configurations (0)

Name	AMI ID	Instance type	Spot price	Creation time

No launch configurations found in this region.

Assign a name for the Launch Configurations and select the AMI that you want to configure



Launch configuration name

Name: AUTO

Amazon machine image (AMI) Info

AMI: snapshot

My AMIs

snapshot

AWS Marketplace

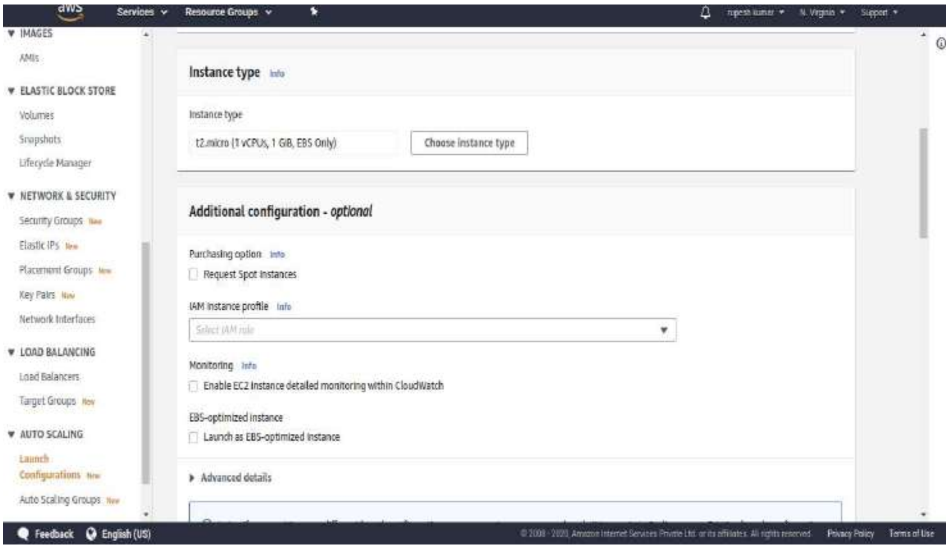
TheHive3_20200513_1589380488-b7e97799-e635-4fa4-8af4-fad0cc2b7d03-ami-03cac8f9feaca4855.4

Cortex3_20200513_1589380486-7893c21e-ae95-4a53-8767-6feb7734f8dd-ami-051835d87dd93bf66.4

TheHive3_20200706_1594045095-b7e97799-e635-4fa4-8af4-fad0cc2b7d03-ami-0d20abdfd5d469b09.4

IAM instance profile Info

Select the instance type as t2.micro ,because the original instance has the same instance type .



Instance type Info

Instance type: t2.micro (1 vCPUs, 1 GiB, EBS Only) Choose instance type

Additional configuration - *optional*

Purchasing option Info

□ Request Spot Instances

IAM instance profile Info

Select IAM role

Monitoring Info

□ Enable EC2 instance detailed monitoring within CloudWatch

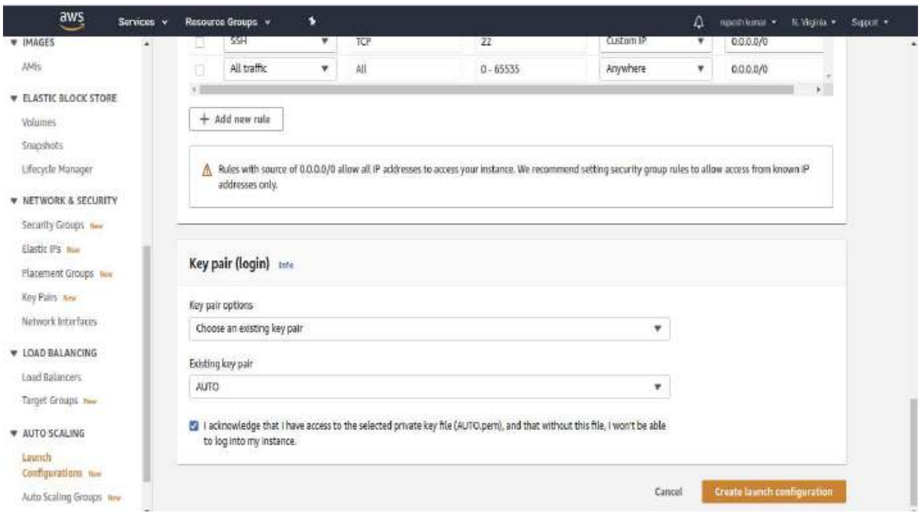
EBS-optimized instance

□ Launch as EBS-optimized instance

▸ Advanced details

Under security groups , click on the Create a new security group .Scroll down a little bit, under the rules section ,click on the “add new rule” button and add a rule.(All traffic or HTTP and HTTPS)

Select on the existing key pair, select your keypair(previously you're using in instance creation) click on i acknowledge checkbox and click on “create launch configuration” button.



SSH	TCP	22	Custom IP	0.0.0.0/0
All traffic	All	0 - 65535	Anywhere	0.0.0.0/0

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

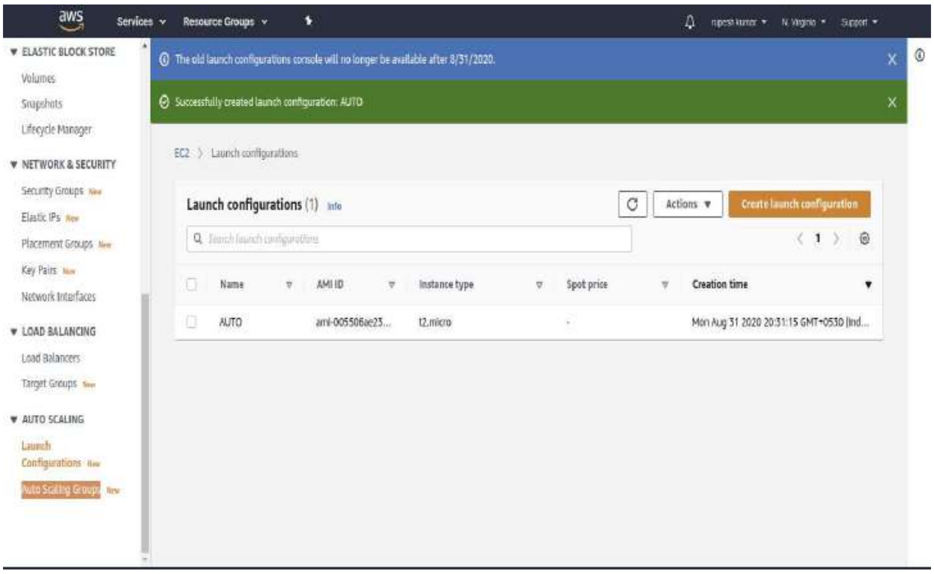
Key pair (login) Info

Key pair options: Choose an existing key pair

Existing key pair: AUTO

☑ I acknowledge that I have access to the selected private key file (AUTO.pem), and that without this file, I won't be able to log into my instance.

Launch configuration was successfully created. Now scroll down the left side panel and click on auto scaling groups under the auto scaling section.



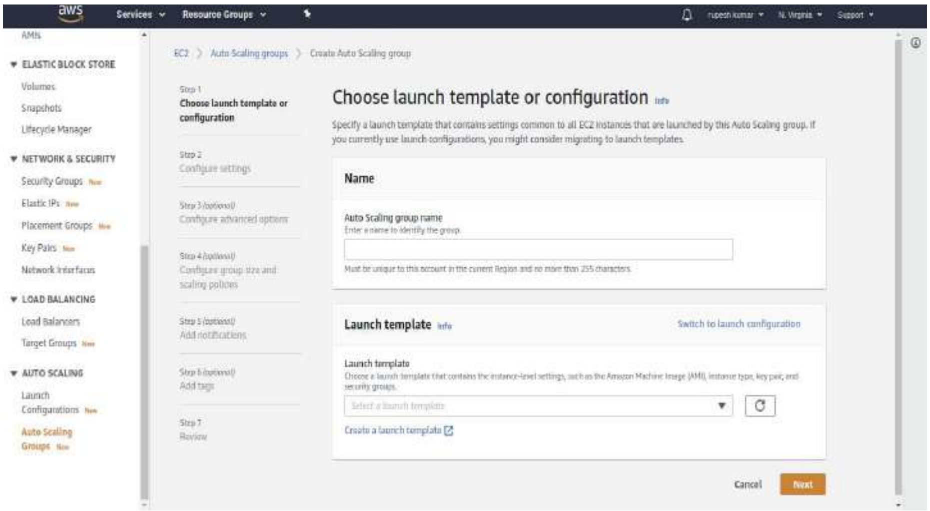
The old launch configurations console will no longer be available after 8/31/2020.

Successfully created launch configuration: AUTO

Launch configurations (1) Info

Name	AMI ID	Instance type	Spot price	Creation time
AUTO	ami-005506ae23...	t2.micro	-	Mon Aug 31 2020 20:31:15 GMT+0530 (Ind...

Click on Create Auto Scaling Group



Choose launch template or configuration

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

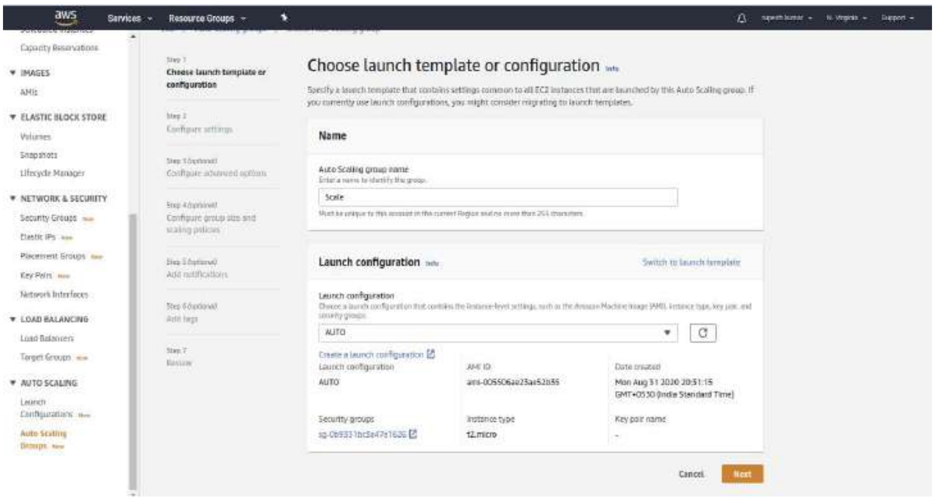
Auto Scaling group name
Enter a name to identify the group.
Must be unique to this account in the current Region and no more than 255 characters.

Launch template

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Choose launch template or configuration:

Enter Auto scaling group name and Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.(*Here i am using Launch configuration option.Already i launched launch configuration with the name AUTO*).Click on Next option.



Choose launch template or configuration

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.
Scale
Must be unique to this account in the current Region and no more than 255 characters.

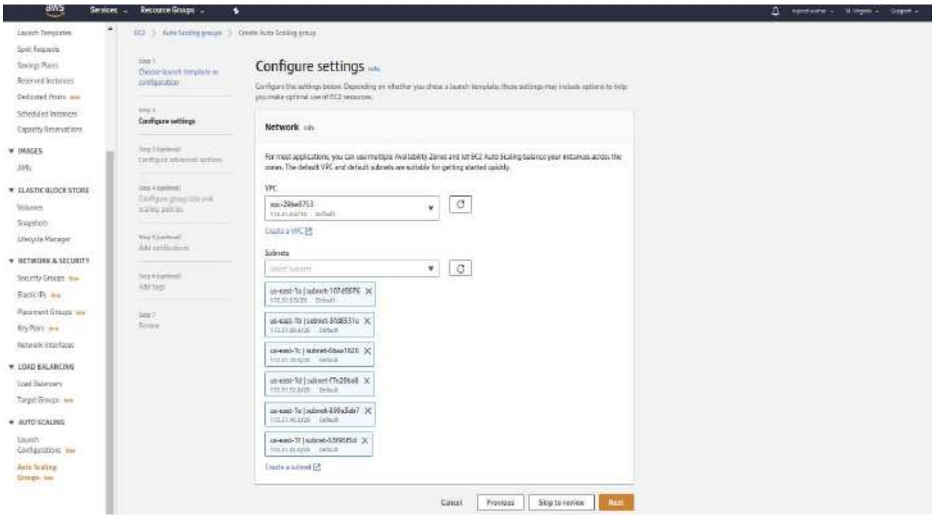
Launch configuration

Launch configuration
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
AUTO

Launch configuration	AMI ID	Date created
AUTO	ami-0055O6ae23ae52b55	Mon Aug 31 2020 20:51:15 GMT+0530 (India Standard Time)
Security groups	**Instance type**	**Key pair name**
sg-0b9331bc3e47e1626	t2.micro	-

Configure settings:

Configure the settings below. Depending on whether you chose a launch template, these settings may include options to help you make optimal use of EC2 resources.Select VPC and all subnets under that VPC.And click on Next.



Configure settings

Configure the settings below. Depending on whether you chose a launch template, these settings may include options to help you make optimal use of EC2 resources.

Network

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

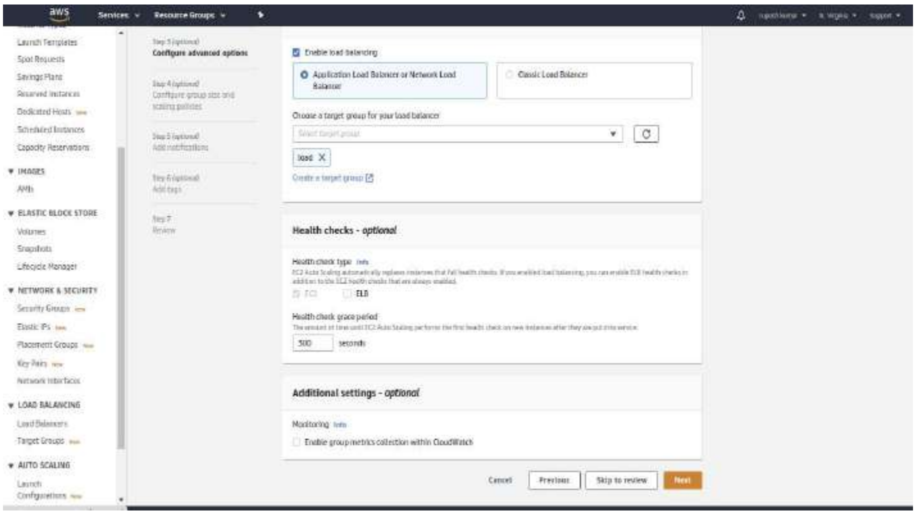
VPC: vpc-29ba8753 [illegible] Default

Subnets:
- us-east-1a | subnet-107d9076 [illegible] Default
- us-east-1b | subnet-3f0853[illegible] [illegible] Default
- us-east-1c | subnet-6baa1[illegible] [illegible] Default
- us-east-1d | subnet-f7c20ba8 [illegible] Default
- us-east-1e | subnet-890a3ab7 [illegible] Default
- us-east-1f | subnet-53f9[illegible] [illegible] Default

Configure advanced options :

Choose a load balancer to distribute incoming traffic for your application across instances. You can also set options that give you more control over checking the health of instances.

Under Load balancing please enable that *Enable* ***load balancing*** check box.After that select the Target group of your load balancer.If you are not created previously please create a new one, by click on create new target group.I already created the Target group name as ***load.*** After that Health checks grace period make it 300 seconds.After that click on Next button.



☑ Enable load balancing

● Application Load Balancer or Network Load Balancer ○ Classic Load Balancer

Choose a target group for your load balancer

Select target group ▼

load ✕

Create a target group

Health checks - *optional*

Health check type Info

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☑ EC2 □ ELB

Health check grace period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300 seconds

Additional settings - *optional*

Monitoring Info

□ Enable group metrics collection within CloudWatch

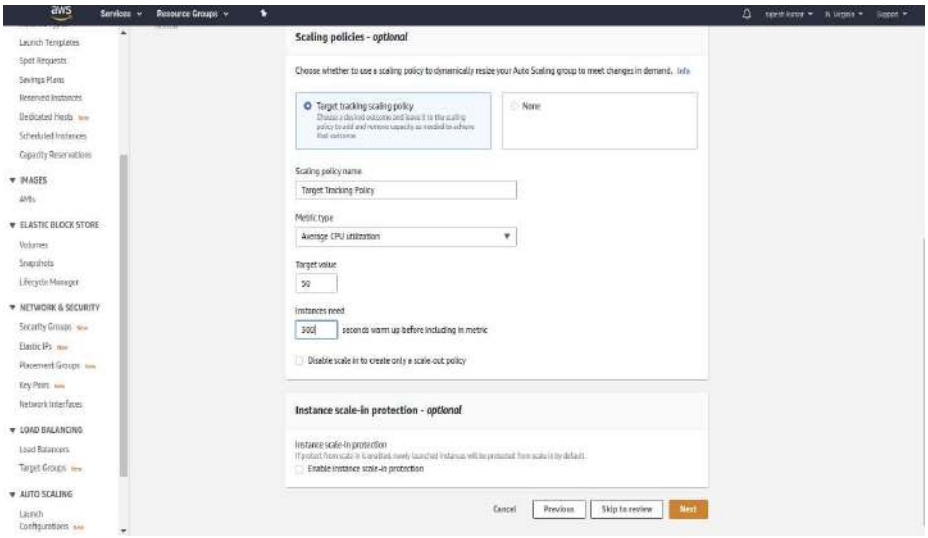
Configure group size and scaling policies:

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

In group size option select the Desired capacity as greater than or equal to Minimum capacity and less than or equal to Maximum capacity .

In Scaling policies , select Target tracking scaling policy then give the scaling policy name ,Metric type (select Metric as CPU utilization),and Target value 50 %.

Instances need time that are 300 seconds warm up before including in the metric.After that click on the Next button.



Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

● Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome

○ None

Scaling policy name: Target Tracking Policy

Metric type: Average CPU utilization

Target value: 50

Instances need: 300 seconds warm up before including in metric

□ Disable scale in to create only a scale-out policy

Instance scale-in protection - *optional*

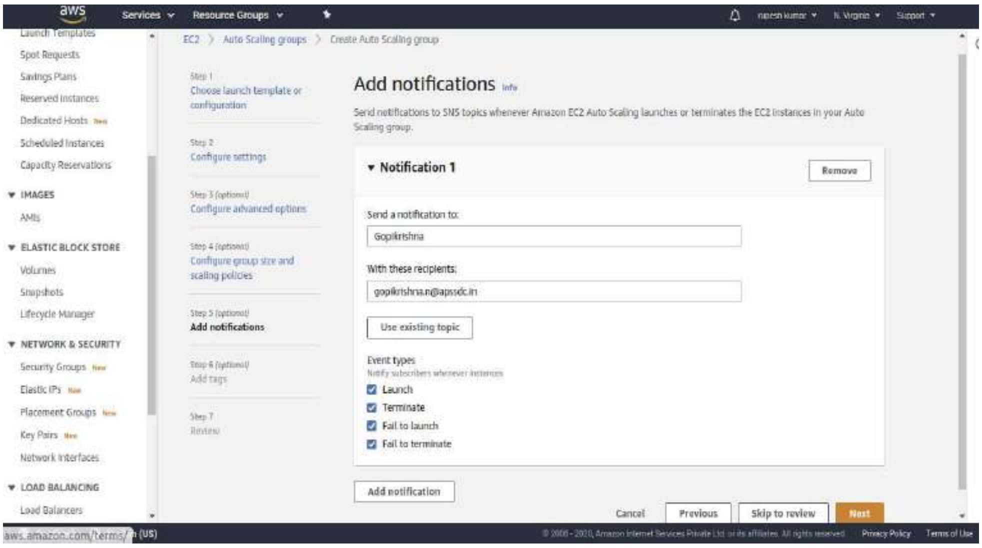
Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

□ Enable instance scale-in protection

Add notifications:

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Click on Add notification button and Enter the name of the person you want to send notification and mail id of the person.After that click on the next button.



Add notifications Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Notification 1

Send a notification to: Gopikrishna

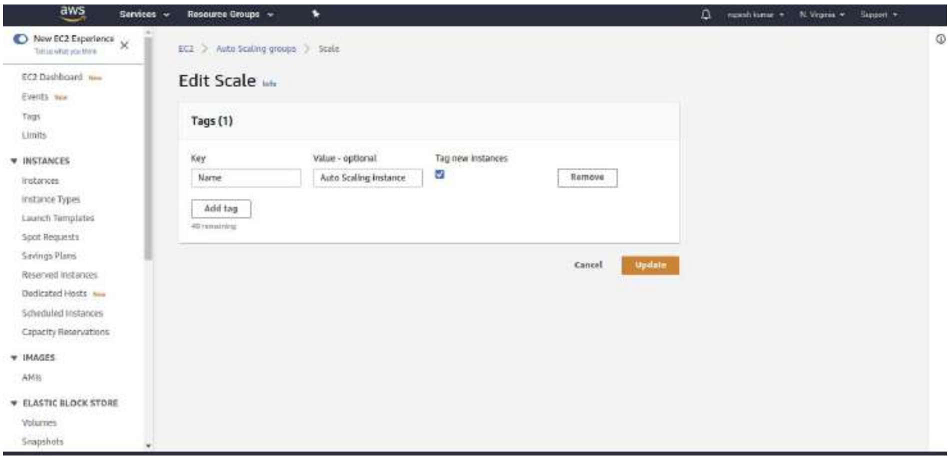
With these recipients: gopikrishna.n@apssdc.in

Event types
Notify subscribers whenever instances

☑ Launch
☑ Terminate
☑ Fail to launch
☑ Fail to terminate

Add tags :

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.After that click on the Next button.



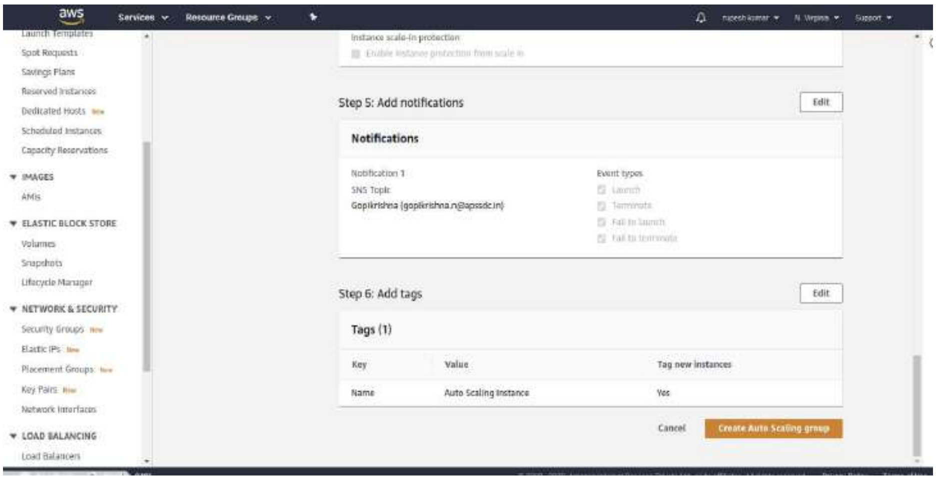
Edit Scale

Tags (1)

Key	Value - optional	Tag new instances
Name	Auto Scaling Instance	☑

Review :

In the Review part once again select all and click on Create Auto scaling group.



Instance scale-in protection

□ Enable instance protection from scale in

Step 5: Add notifications

Notifications

Notification 1
SNS Topic
Gopikrishna (gopikrishna.n@apssdc.in)

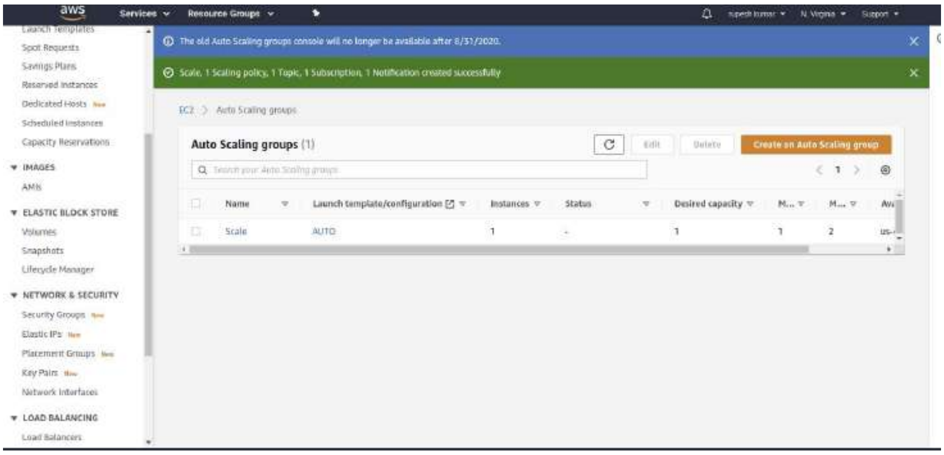
Event types
☑ Launch
☑ Terminate
☑ Fail to launch
☑ Fail to terminate

Step 6: Add tags

Tags (1)

Key	Value	Tag new instances
Name	Auto Scaling Instance	Yes

Check the status of the Auto scaling group.



The old Auto Scaling groups console will no longer be available after 8/31/2020.

Scale, 1 Scaling policy, 1 Topic, 1 Subscription, 1 Notification created successfully

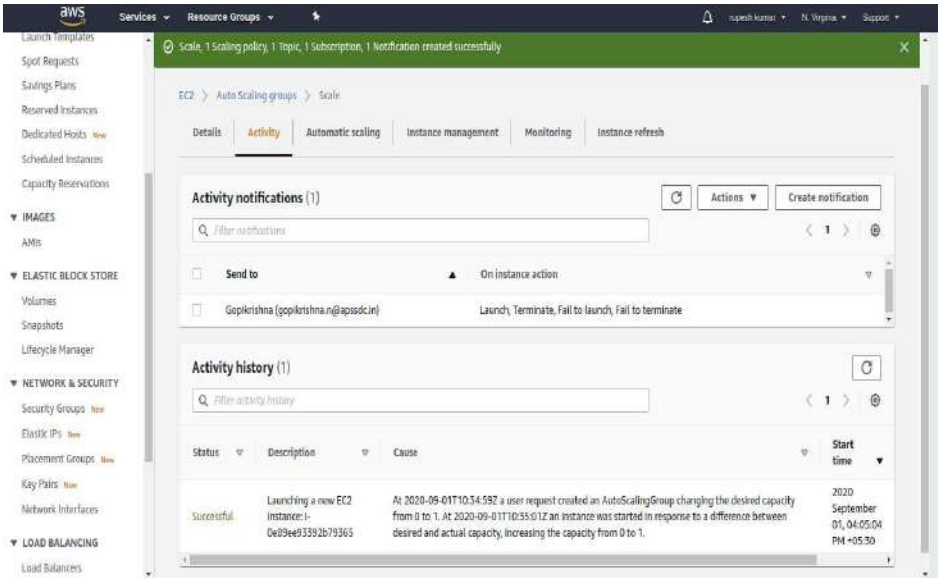
EC2 > Auto Scaling groups

Auto Scaling groups (1)

Name	Launch template/configuration	Instances	Status	Desired capacity	M...	M...	Av
Scale	AUTO	1	-	1	1	2	us-

Click on the Auto scaling group name ,after that click on Activity and check the Activity History status.

Launched on Instance successfully



Scale, 1 Scaling policy, 1 Topic, 1 Subscription, 1 Notification created successfully

EC2 > Auto Scaling groups > Scale

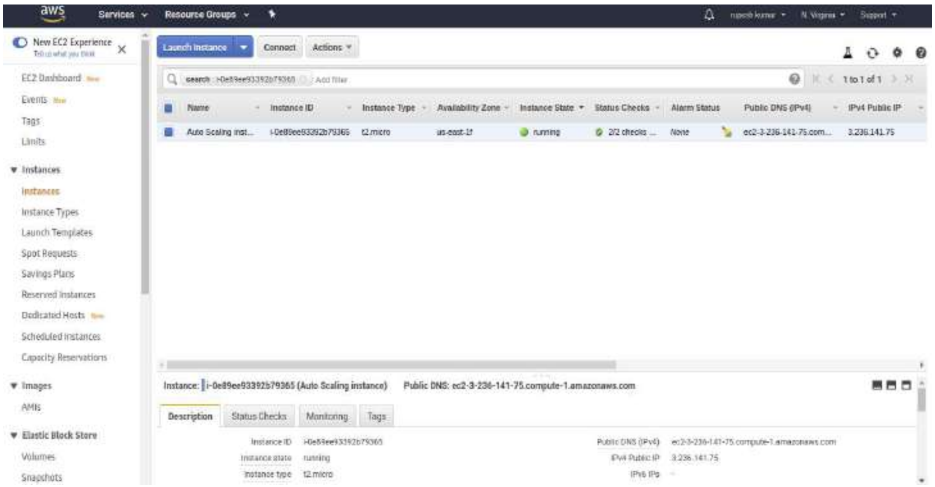
Activity notifications (1)

Send to	On instance action
Gopikrishna (gopikrishna.n@apssdc.in)	Launch, Terminate, Fail to launch, Fail to terminate

Activity history (1)

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-0e89ee93392b79365	At 2020-09-01T10:34:59Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2020-09-01T10:35:01Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2020 September 01, 04:05:04 PM +05:30

Open AWS Management console and click on running instances.You can find out the instance name as *Auto scaling instance*.



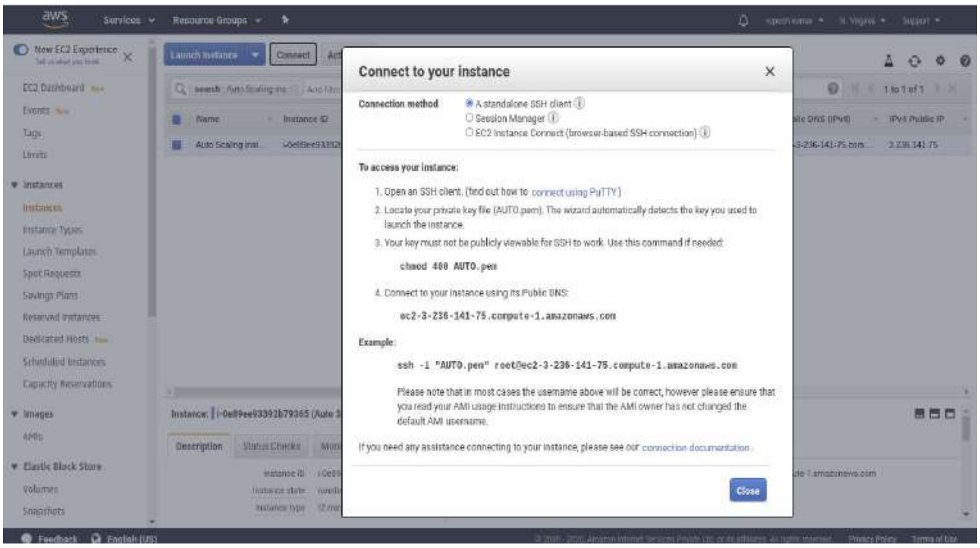
search : i-0e89ee93392b79365

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
Auto Scaling inst...	i-0e89ee93392b79365	t2.micro	us-east-1f	running	2/2 checks ...	None	ec2-3-236-141-75.com...	3.236.141.75

Instance: i-0e89ee93392b79365 (Auto Scaling instance) Public DNS: ec2-3-236-141-75.compute-1.amazonaws.com

Instance ID: i-0e89ee93392b79365
Instance state: running
Instance type: t2.micro
Public DNS (IPv4): ec2-3-236-141-75.compute-1.amazonaws.com
IPv4 Public IP: 3.236.141.75
IPv6 IPs: -

Now we can apply stress management on Auto scaling instance.Connecting the server through putty or Terminal or windows powershell.Now I am connecting through Terminal.



Connect to your instance

Connection method
● A standalone SSH client
○ Session Manager
○ EC2 Instance Connect (browser-based SSH connection)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (AUTO.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 AUTO.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-3-236-141-75.compute-1.amazonaws.com
```

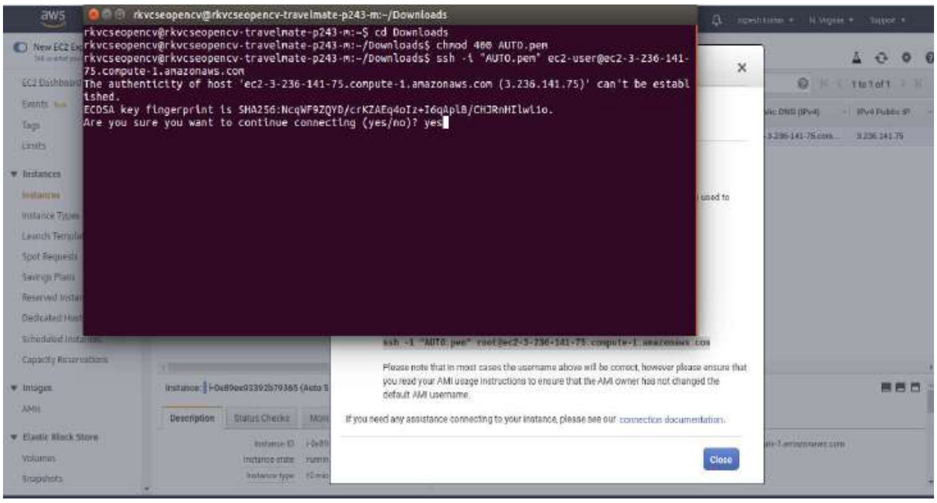
Example:

```
ssh -i "AUTO.pem" root@ec2-3-236-141-75.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

Now open Terminal and connect to your server /instance.



```
rkvcseopencv@rkvcseopencv-travelmate-p243-m:~$ cd Downloads
rkvcseopencv@rkvcseopencv-travelmate-p243-m:~/Downloads$ chmod 400 AUTO.pem
rkvcseopencv@rkvcseopencv-travelmate-p243-m:~/Downloads$ ssh -i "AUTO.pem" ec2-user@ec2-3-236-141-
75.compute-1.amazonaws.com
The authenticity of host 'ec2-3-236-141-75.compute-1.amazonaws.com (3.236.141.75)' can't be establ
ished.
ECDSA key fingerprint is SHA256:NcqWF9ZQYD/crKZAEq4oIz+I6qAplB/CHJRnHIlwl1o.
Are you sure you want to continue connecting (yes/no)? yes
```

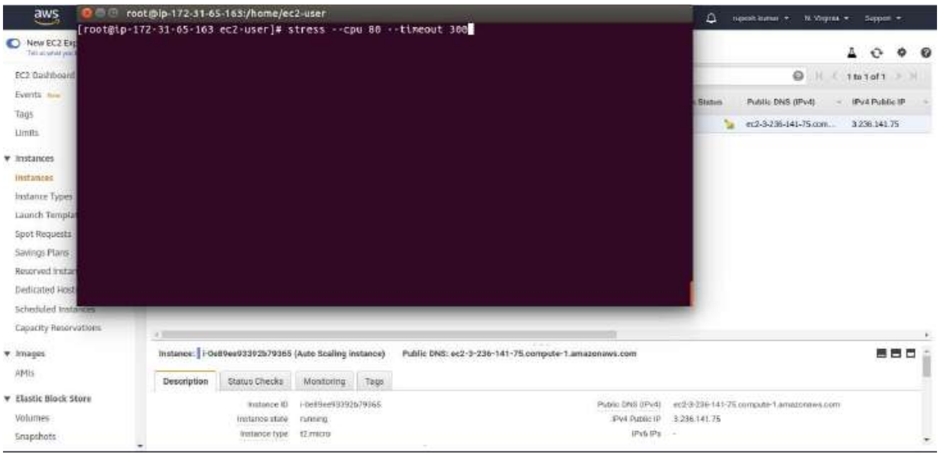
Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

Now install stress and apply stress Management on the server
Use this commands for installing and apply stress management

For Installation:
Sudo yum install stress (incase Linux server)
Sudo apt-get install stress (Incase of Ubuntu server)

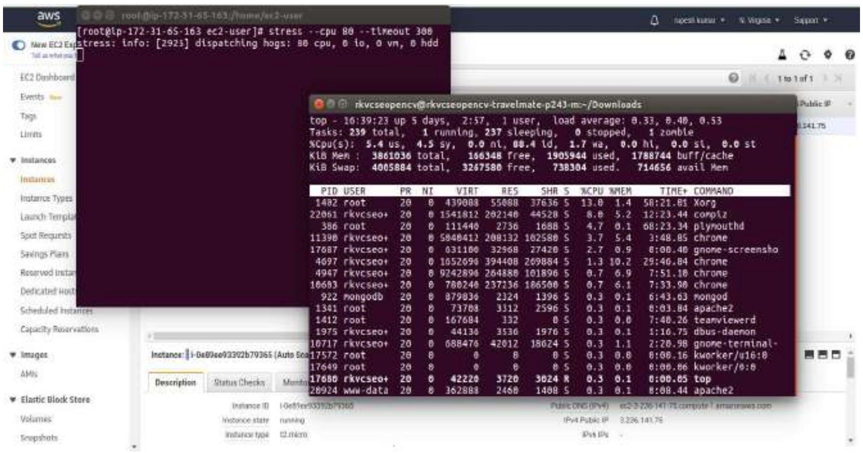
Apply stress by using :
stress --cpu 80 --timeout 300



```
[root@ip-172-31-65-163 ec2-user]# stress --cpu 80 --timeout 300
```

Now check the status of CPU utilization by using following command

Open new Terminal and use this command **: top**

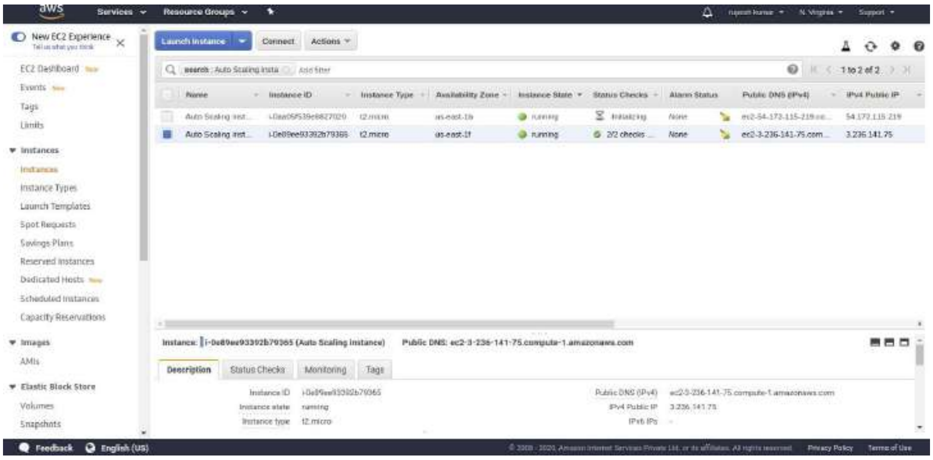


```
[root@ip-172-31-65-163 ec2-user]# stress --cpu 80 --timeout 300
stress: info: [2925] dispatching hogs: 80 cpu, 0 io, 0 vm, 0 hdd
```

```
top - 16:39:23 up 5 days,  2:57,  1 user,  load average: 0.33, 0.40, 0.53
Tasks: 239 total,   1 running, 237 sleeping,   0 stopped,   1 zombie
%Cpu(s):  5.4 us,  4.5 sy,  0.0 ni, 88.4 id,  1.7 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  3861036 total,   166348 free,  1905944 used,  1788744 buff/cache
KiB Swap:  4005884 total,  3267580 free,   738304 used.   714656 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1402 root      20   0  439088  55088  37636 S  13.0  1.4  58:21.81 Xorg
22061 rkvcseo+  20   0 1541812 202140  44528 S   8.0  5.2  12:23.44 compiz
  386 root      20   0  111440   2736   1688 S   4.7  0.1  68:23.34 plymouthd
11390 rkvcseo+  20   0 5840412 208132 102580 S   3.7  5.4   3:48.85 chrome
17687 rkvcseo+  20   0  631100  32968  27420 S   2.7  0.9   0:00.40 gnome-screensho
 4697 rkvcseo+  20   0 1652696 394408 209884 S   1.3 10.2  29:46.84 chrome
 4947 rkvcseo+  20   0 9242896 264888 101896 S   0.7  6.9   7:51.18 chrome
10683 rkvcseo+  20   0  780240 237236 186500 S   0.7  6.1   7:33.98 chrome
  922 mongodb   20   0  879836   2324   1396 S   0.3  0.1   6:43.63 mongod
 1341 root      20   0   73708   3312   2596 S   0.3  0.1   0:03.84 apache2
 1412 root      20   0  167684    332      0 S   0.3  0.0   7:40.26 teamviewerd
 1975 rkvcseo+  20   0   44136   3536   1976 S   0.3  0.1   1:16.75 dbus-daemon
10717 rkvcseo+  20   0  688476  42012  18624 S   0.3  1.1   2:20.98 gnome-terminal-
17572 root      20   0       0      0      0 S   0.3  0.0   0:00.16 kworker/u16:0
17649 root      20   0       0      0      0 S   0.3  0.0   0:00.06 kworker/0:0
17680 rkvcseo+  20   0   42220   3720   3024 R   0.3  0.1   0:00.05 top
20924 www-data  20   0  362888   2460   1408 S   0.3  0.1   0:08.44 apache2
```

When i am applying stress management on instance(Name of instance : **Auto scaling instance**).Automatically one more new instance launched .Check the status of the below screenshot.



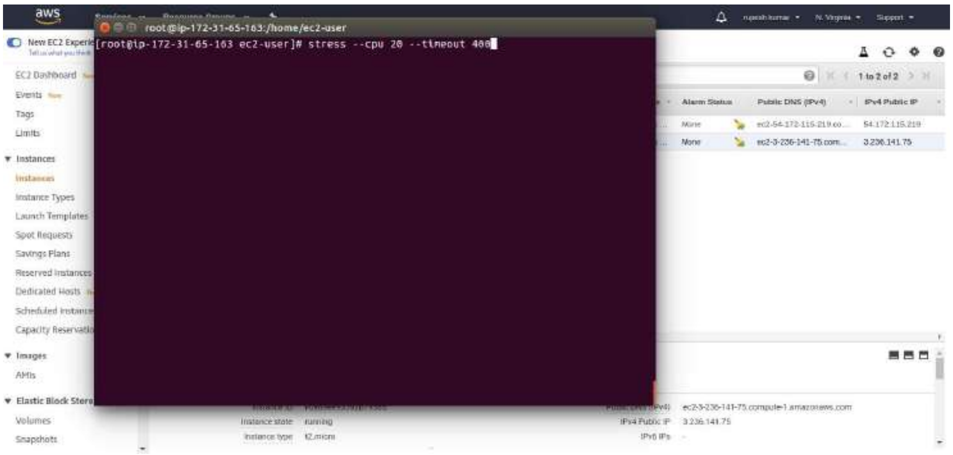
search : Auto Scaling insta

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
Auto Scaling inst...	i-0aa05f539e6827020	t2.micro	us-east-1b	running	Initializing	None	ec2-54-172-115-219.co...	54.172.115.219
Auto Scaling inst...	i-0e89ee93392b79365	t2.micro	us-east-1f	running	2/2 checks ...	None	ec2-3-236-141-75.com...	3.236.141.75

Instance: i-0e89ee93392b79365 (Auto Scaling Instance)   Public DNS: ec2-3-236-141-75.compute-1.amazonaws.com

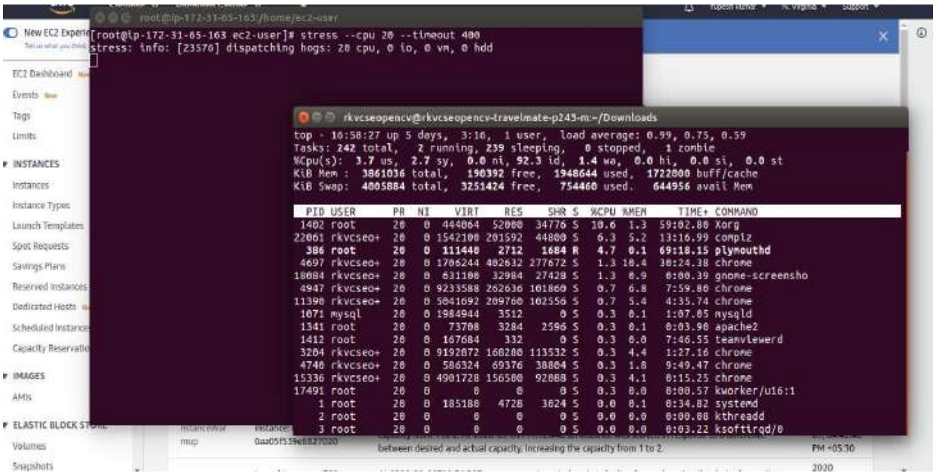
Instance ID	i-0e89ee93392b79365	Public DNS (IPv4)	ec2-3-236-141-75.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.236.141.75
Instance type	t2.micro	IPv6 IPs	-

Now you can Decrease the Stress management for instance by using the following command.



```
[root@ip-172-31-65-163 ec2-user]# stress --cpu 20 --timeout 400
```

Check the status of the CPU Utilization on the server in the below screenshot.

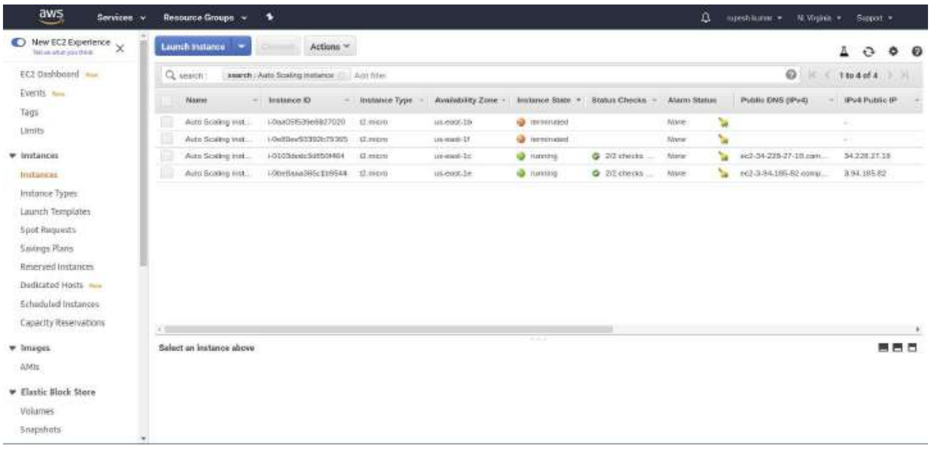


```
[root@ip-172-31-65-163 ec2-user]# stress --cpu 20 --timeout 400
stress: info: [23576] dispatching hogs: 20 cpu, 0 io, 0 vm, 0 hdd
```

```
top - 16:58:27 up 5 days,  3:16,  1 user,  load average: 0.99, 0.75, 0.59
Tasks: 242 total,   2 running, 239 sleeping,   0 stopped,   1 zombie
%Cpu(s):  3.7 us,  2.7 sy,  0.0 ni, 92.3 id,  1.4 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  3861036 total,   190392 free,  1948644 used,  1722000 buff/cache
KiB Swap:  4005884 total,  3251424 free,   754460 used.   644956 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1402 root      20   0  444064  52080  34776 S  10.6  1.3  59:02.80 Xorg
22061 rkvcseo+  20   0 1542100 201592  44800 S   6.3  5.2  13:16.99 compiz
  386 root      20   0  111440   2712   1684 R   4.7  0.1  69:18.15 plymouthd
 4697 rkvcseo+  20   0 1706244 402632 277672 S   1.3 10.4  30:24.38 chrome
18084 rkvcseo+  20   0  631100  32984  27428 S   1.3  0.9   0:00.39 gnome-screensho
 4947 rkvcseo+  20   0 9233588 262636 101860 S   0.7  6.8   7:59.80 chrome
11390 rkvcseo+  20   0 5041692 209760 102556 S   0.7  5.4   4:35.74 chrome
 1071 mysql     20   0 1984944   3512      0 S   0.3  0.1   1:07.05 mysqld
 1341 root      20   0   73708   3284   2596 S   0.3  0.1   0:03.90 apache2
 1412 root      20   0  167684    332      0 S   0.3  0.0   7:46.55 teamviewerd
 3204 rkvcseo+  20   0 9192072 160200 113532 S   0.3  4.4   1:27.16 chrome
 4740 rkvcseo+  20   0  586324  69376  38804 S   0.3  1.8   9:49.47 chrome
15336 rkvcseo+  20   0 4901728 156500  92088 S   0.3  4.1   0:15.25 chrome
17491 root      20   0       0      0      0 S   0.3  0.0   0:00.57 kworker/u16:1
    1 root      20   0  185188   4728   3024 S   0.0  0.1   0:34.82 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.08 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:03.22 ksoftirqd/0
```

Now goto Ec2 dashboard and check the status of the instances.The CPU utilization goes down so that instances will be terminated which were launched when CPU utilization high.Look at this below screenshot.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
Auto Scaling inst...	i-0aa05f539e8827020	t2.micro	us-east-1b	terminated		None		-
Auto Scaling inst...	i-0e8f0ee93392b79365	t2.micro	us-east-1f	terminated		None		-
Auto Scaling inst...	i-0103dedc3d850f484	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-34-228-27-18.com...	34.228.27.18
Auto Scaling inst...	i-0beBaaa385c1b9544	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-3-94-185-82.comp...	3.94.185.82

The lab regarding Auto scaling with Load Balancer and snapshot was completed………………...