Practical 6

Cloud Computing

2CSDE67

Mistry Unnat

20BCE515

Date

April 5, 2022

Department of Computer Science and Engineering

Institute of Technology

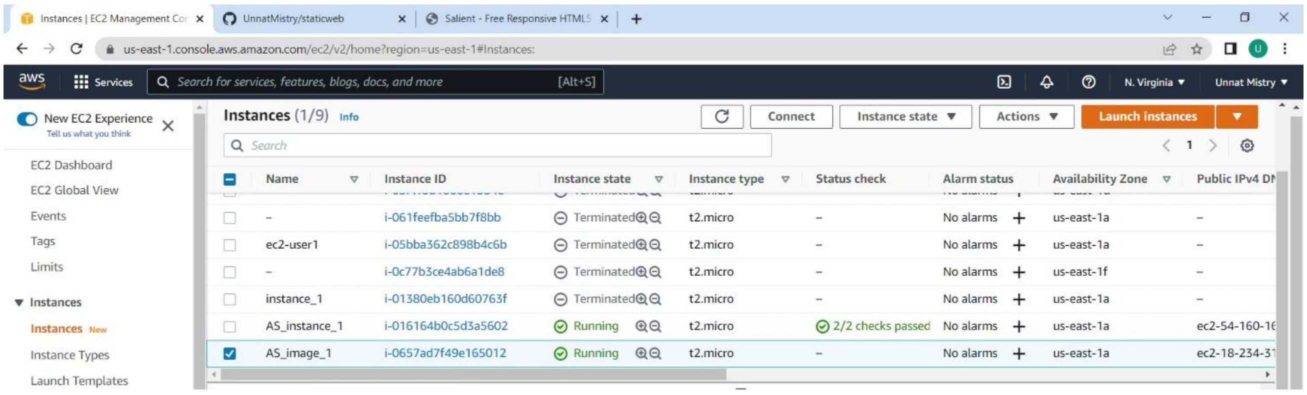
Nirma University

Ahmedabad

AIM: Working with an IaaS Cloud Computing: Using AWS (Amazon Web Services) to understating Auto Scaling Concept.

Steps:

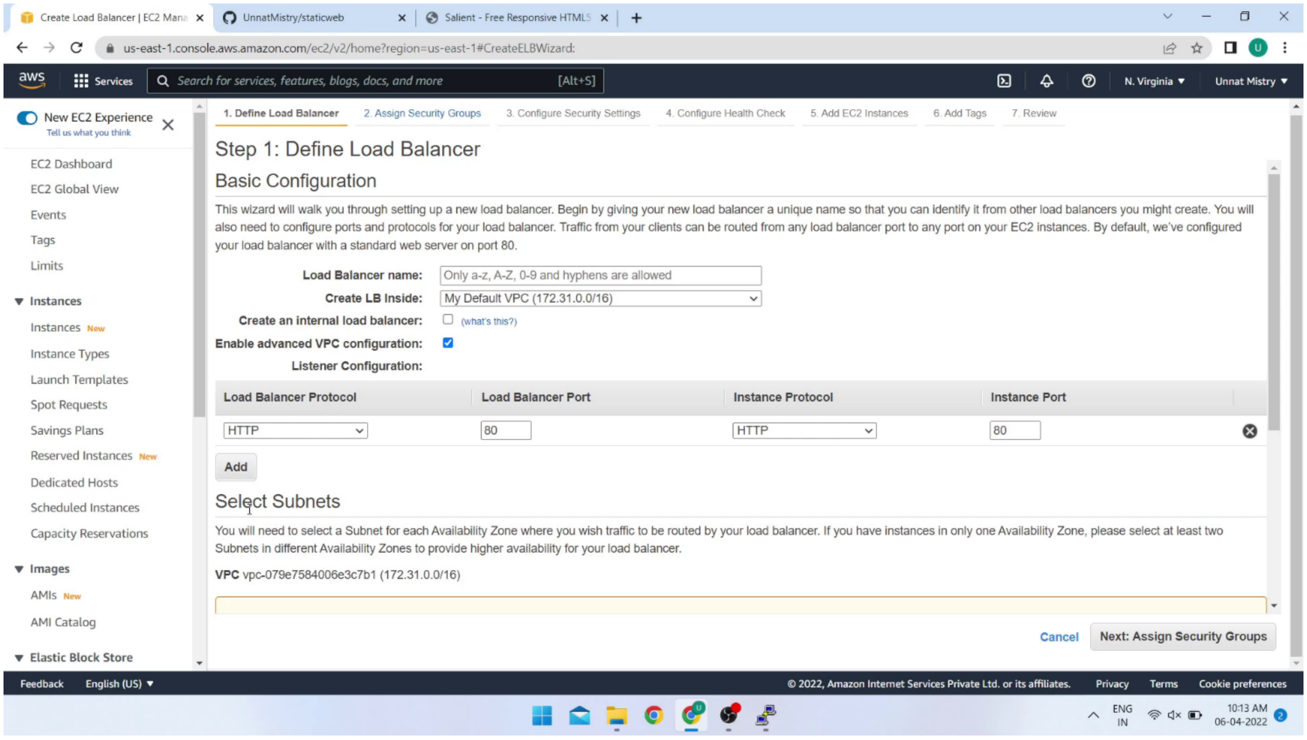
We will use already created instances.



Instances (1/9) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
–	i-061feefba5bb7f8bb	Terminated	t2.micro	–	No alarms	us-east-1a	–
ec2-user1	i-05bba362c898b4c6b	Terminated	t2.micro	–	No alarms	us-east-1a	–
–	i-0c77b3ce4ab6a1de8	Terminated	t2.micro	–	No alarms	us-east-1f	–
instance_1	i-01380eb160d60763f	Terminated	t2.micro	–	No alarms	us-east-1a	–
AS_instance_1	i-016164b0c5d3a5602	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-160-1
AS_image_1	i-0657ad7f49e165012	Running	t2.micro	–	No alarms	us-east-1a	ec2-18-234-3

Creating classic load balancer:



1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

Step 1: Define Load Balancer

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name:	Only a-z, A-Z, 0-9 and hyphens are allowed
Create LB Inside:	My Default VPC (172.31.0.0/16)
Create an internal load balancer:	☐ (what's this?)
Enable advanced VPC configuration:	✓
Listener Configuration:	

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Add

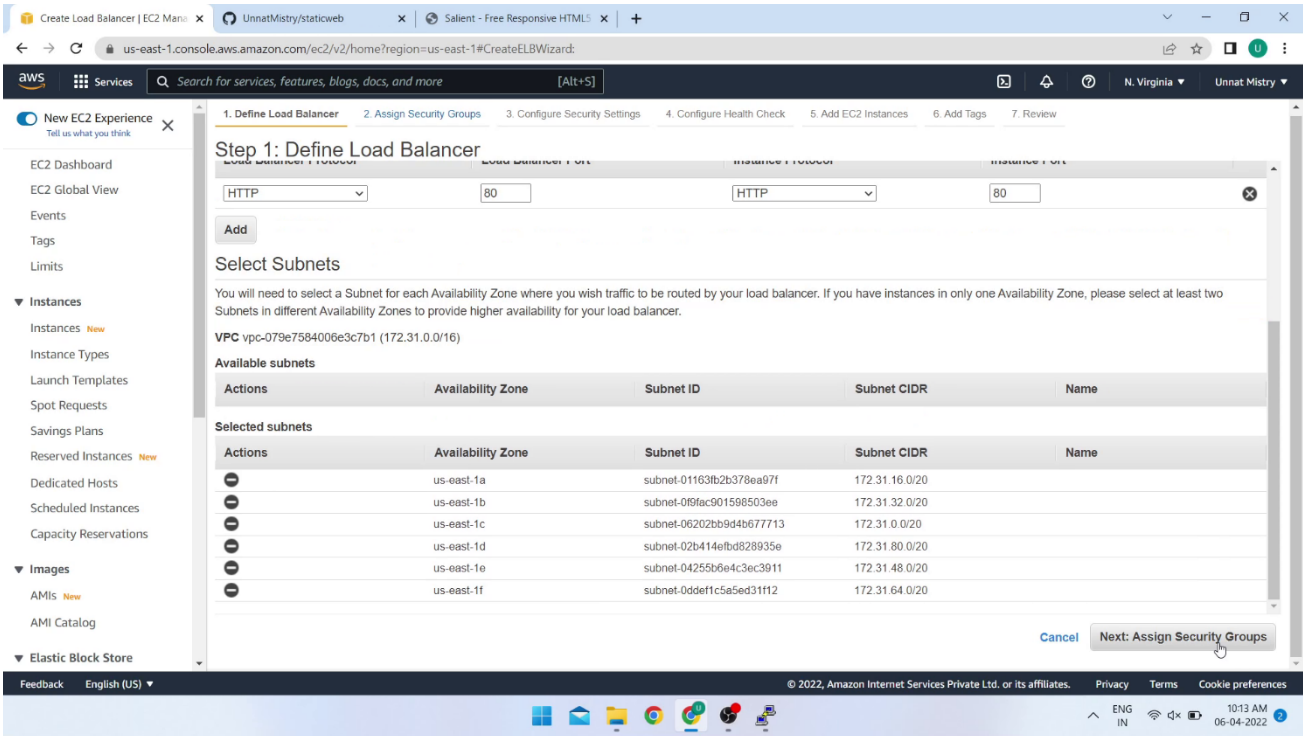
Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-079e7584006e3c7b1 (172.31.0.0/16)

Cancel Next: Assign Security Groups

Tick enable vpc configuration and select all subnets:



Step 1: Define Load Balancer

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-079e7584006e3c7b1 (172.31.0.0/16)

Available subnets

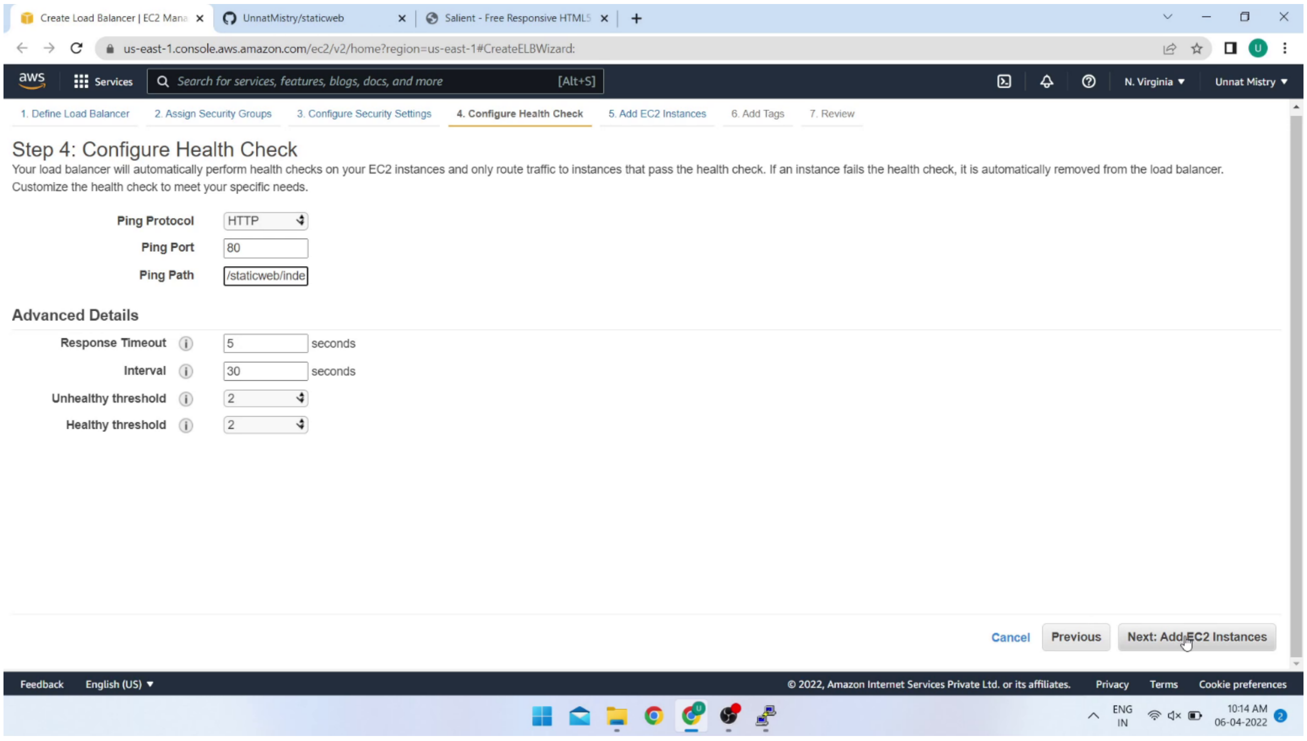
Actions	Availability Zone	Subnet ID	Subnet CIDR	Name

Selected subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊖	us-east-1a	subnet-01163fb2b378ea97f	172.31.16.0/20	
⊖	us-east-1b	subnet-0f9fac901598503ee	172.31.32.0/20	
⊖	us-east-1c	subnet-06202bb9d4b677713	172.31.0.0/20	
⊖	us-east-1d	subnet-02b414efbd828935e	172.31.80.0/20	
⊖	us-east-1e	subnet-04255b6e4c3ec3911	172.31.48.0/20	
⊖	us-east-1f	subnet-0ddef1c5a5ed31f12	172.31.64.0/20	

1. Define Load Balancer
2. Assign Security Groups
3. Configure Security Settings
4. Configure Health Check
5. Add EC2 Instances
6. Add Tags
7. Review
Add
Cancel
Next: Assign Security Groups

In configure health check update ping path and set healthy threshold to 2:



Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

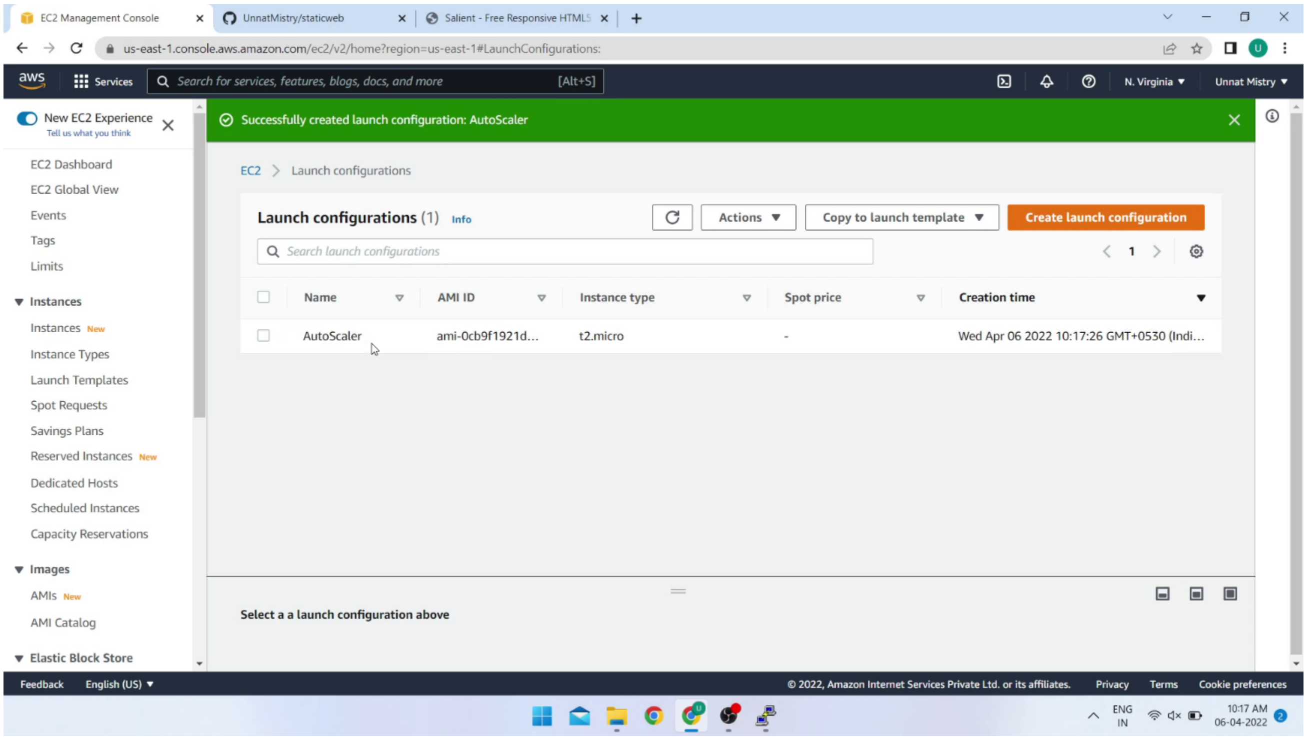
Ping Protocol	HTTP
Ping Port	80
Ping Path	/staticweb/inde

Advanced Details

Response Timeout	5 seconds
Interval	30 seconds
Unhealthy threshold	2
Healthy threshold	2

To create Auto Scaling Groups, we first need to create a Launch Configuration.

In the Left pane select Launch configurations and select Create launch configurations and create one by selecting all appropriate options:



us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchConfigurations:

Successfully created launch configuration: AutoScaler

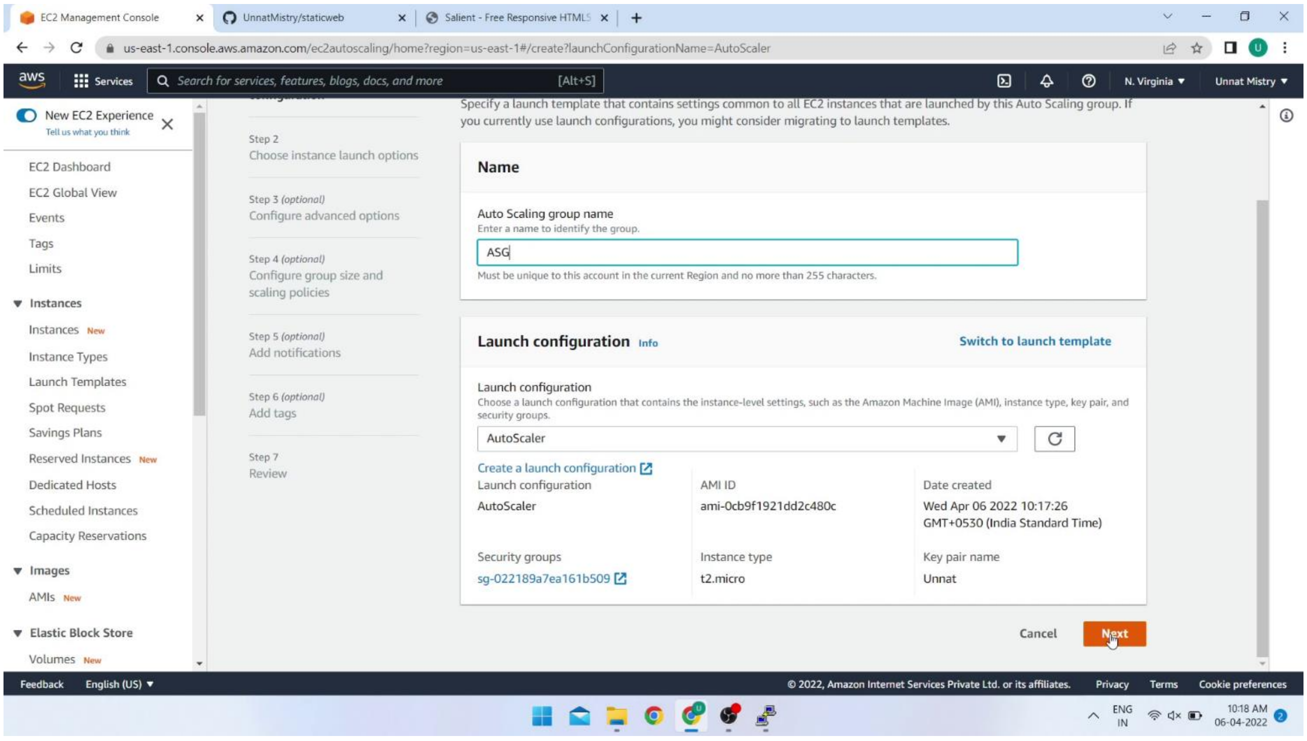
Launch configurations (1)

	Name	AMI ID	Instance type	Spot price	Creation time
☐	AutoScaler	ami-0cb9f1921d...	t2.micro	-	Wed Apr 06 2022 10:17:26 GMT+0530 (Indi...

Select a a launch configuration above

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Now creating auto scaling group enter name and select the launch configuration we created in previous step:



us-east-1.console.aws.amazon.com/ec2autoscaling/home?region=us-east-1#/create?launchConfigurationName=AutoScaler

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.
ASG
Must be unique to this account in the current Region and no more than 255 characters.

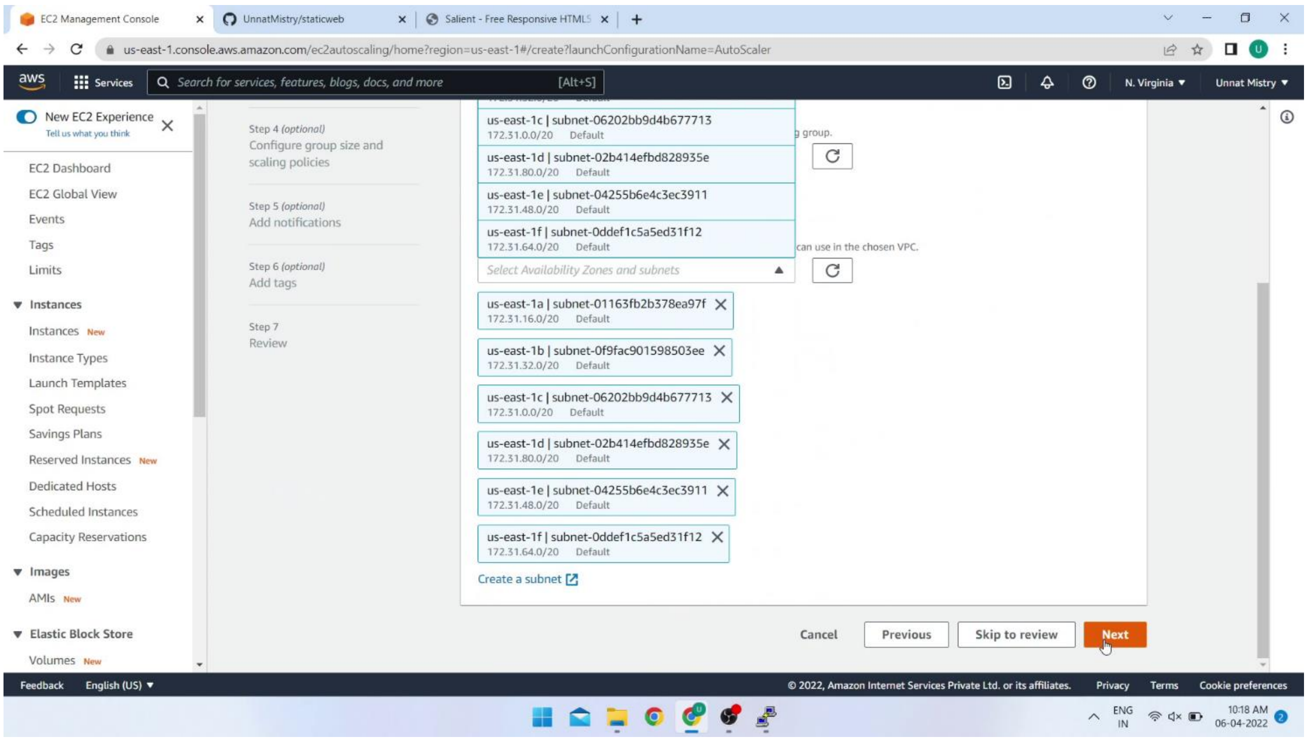
Launch configuration

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.
AutoScaler

Launch configuration	AMI ID	Date created
AutoScaler	ami-0cb9f1921dd2c480c	Wed Apr 06 2022 10:17:26 GMT+0530 (India Standard Time)
Security groups	**Instance type**	**Key pair name**
sg-022189a7ea161b509	t2.micro	Unnat

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

In availability zones and subnets select all the subnets available:



Step 4 *(optional)*
Configure group size and scaling policies

Step 5 *(optional)*
Add notifications

Step 6 *(optional)*
Add tags

Step 7
Review

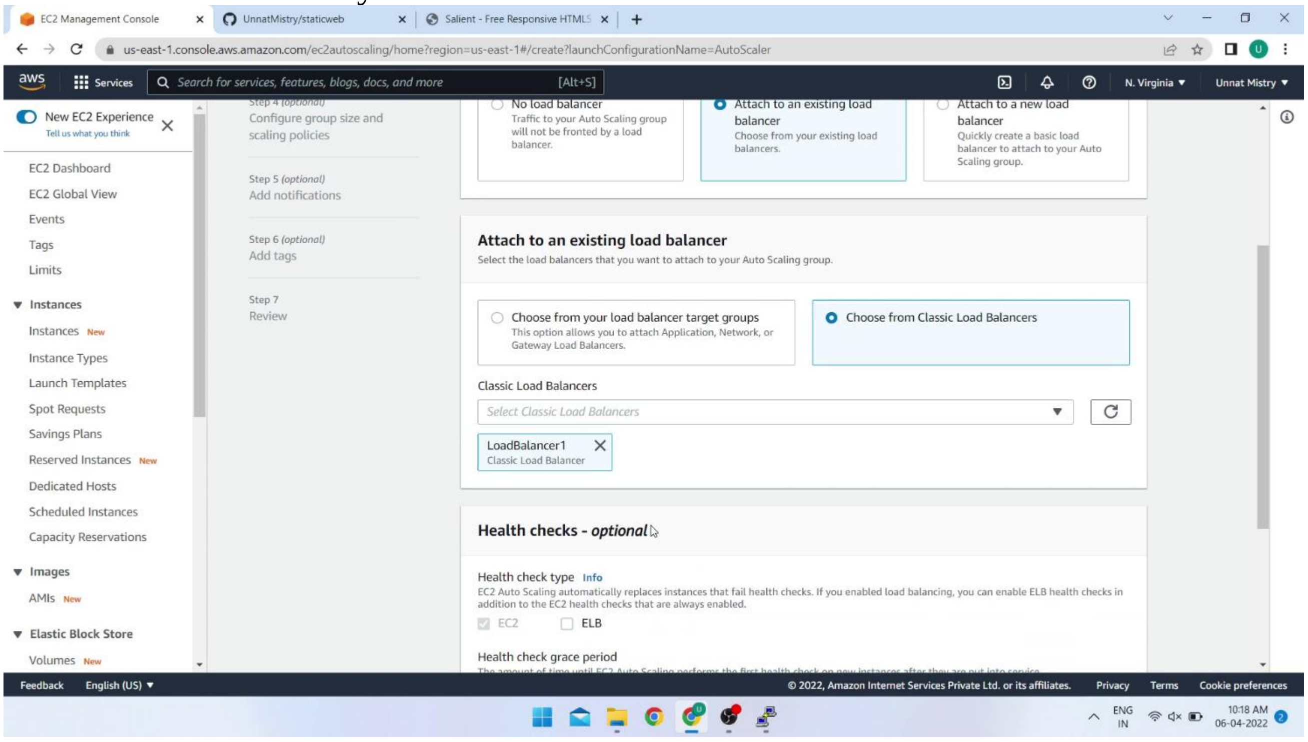
us-east-1c | subnet-06202bb9d4b677713
172.31.0.0/20 Default
us-east-1d | subnet-02b414efbd828935e
172.31.80.0/20 Default
us-east-1e | subnet-04255b6e4c3ec3911
172.31.48.0/20 Default
us-east-1f | subnet-0ddef1c5a5ed31f12
172.31.64.0/20 Default

Select Availability Zones and subnets

us-east-1a | subnet-01163fb2b378ea97f
172.31.16.0/20 Default
us-east-1b | subnet-0f9fac901598503ee
172.31.32.0/20 Default
us-east-1c | subnet-06202bb9d4b677713
172.31.0.0/20 Default
us-east-1d | subnet-02b414efbd828935e
172.31.80.0/20 Default
us-east-1e | subnet-04255b6e4c3ec3911
172.31.48.0/20 Default
us-east-1f | subnet-0ddef1c5a5ed31f12
172.31.64.0/20 Default

Create a subnet

Select and attach already created load balancer:



No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Classic Load Balancers
Select Classic Load Balancers
LoadBalancer1
Classic Load Balancer

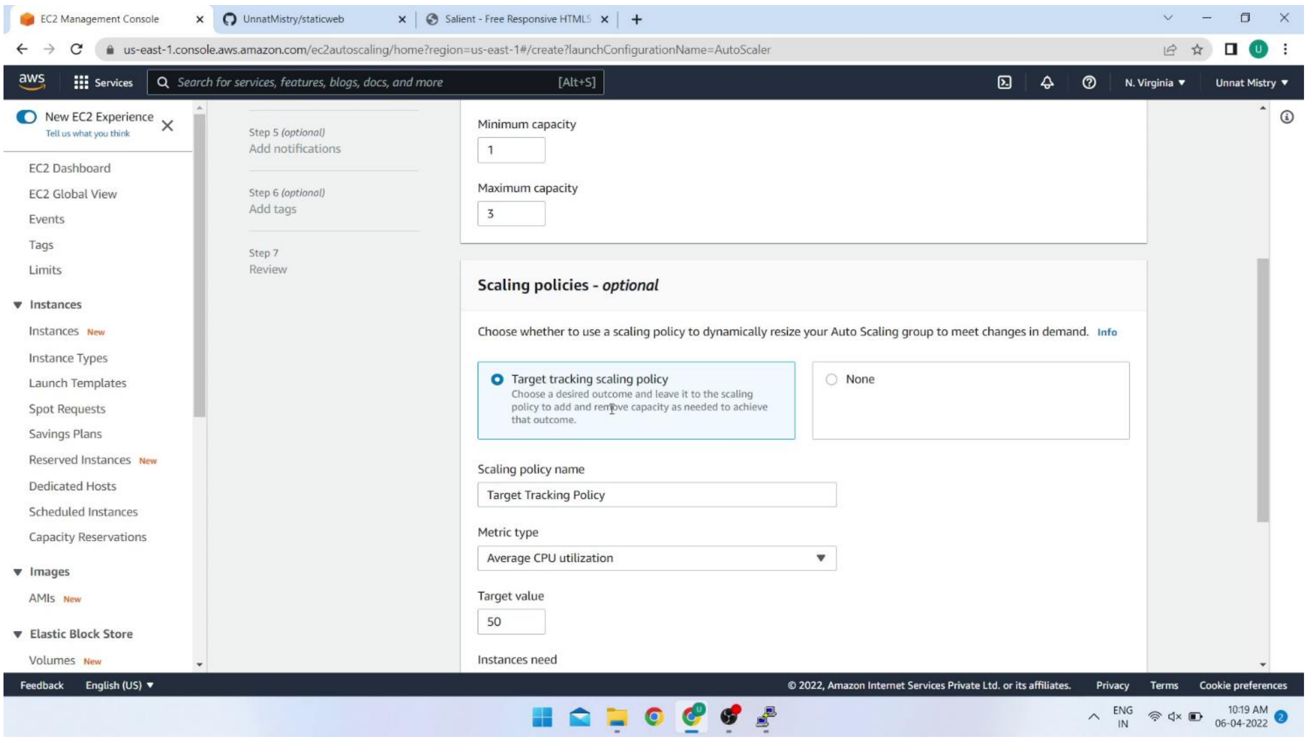
Health checks - *optional*

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

EC2 Management Console
UnnatMistry/staticweb
Salient - Free Responsive HTML5
us-east-1.console.aws.amazon.com/ec2autoscaling/home?region=us-east-1#/create?launchConfigurationName=AutoScaler
Services
N. Virginia
Unnat Mistry
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
10:18 AM
06-04-2022

In configure group size set max capacity to 3 and select target tracking scaling policy in scaling policies option:



Minimum capacity: 1

Maximum capacity: 3

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

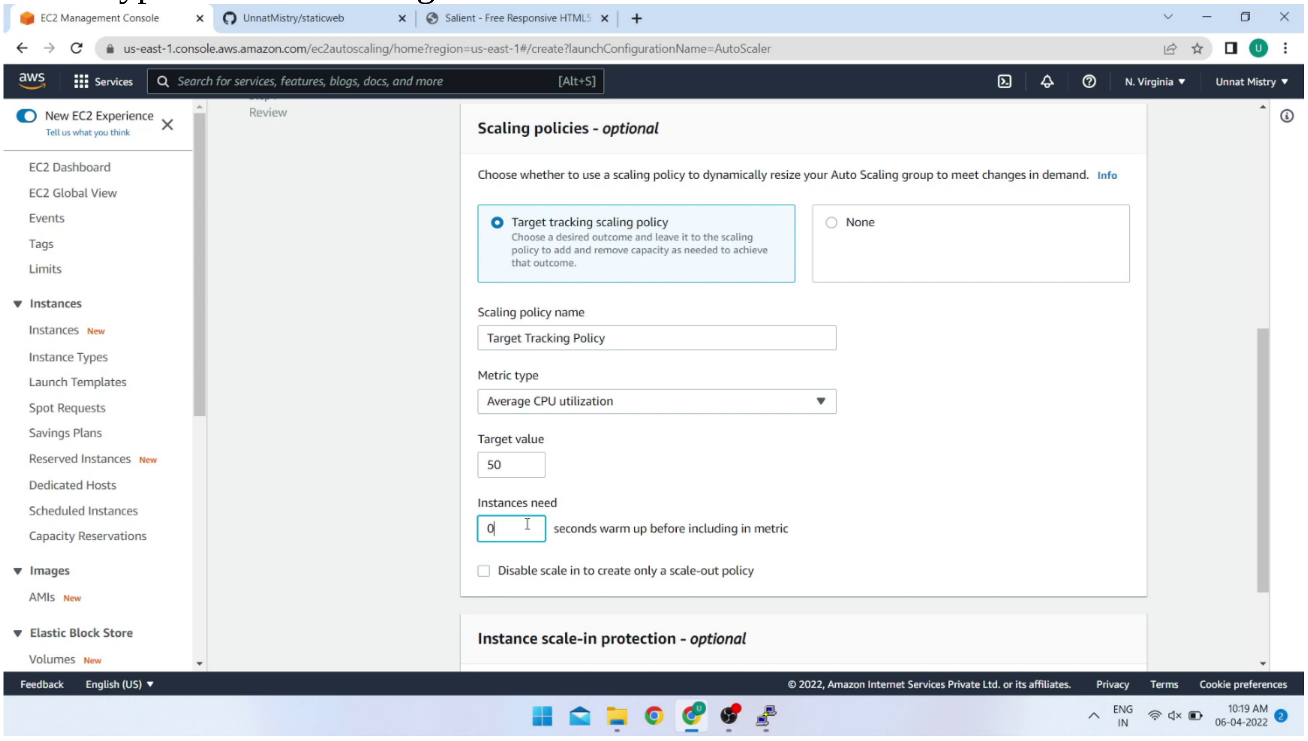
Scaling policy name: Target Tracking Policy

Metric type: Average CPU utilization

Target value: 50

Instances need

Set the instances need to 0 seconds and the target value should be 50 and the metric type should be average CPU utilization.



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

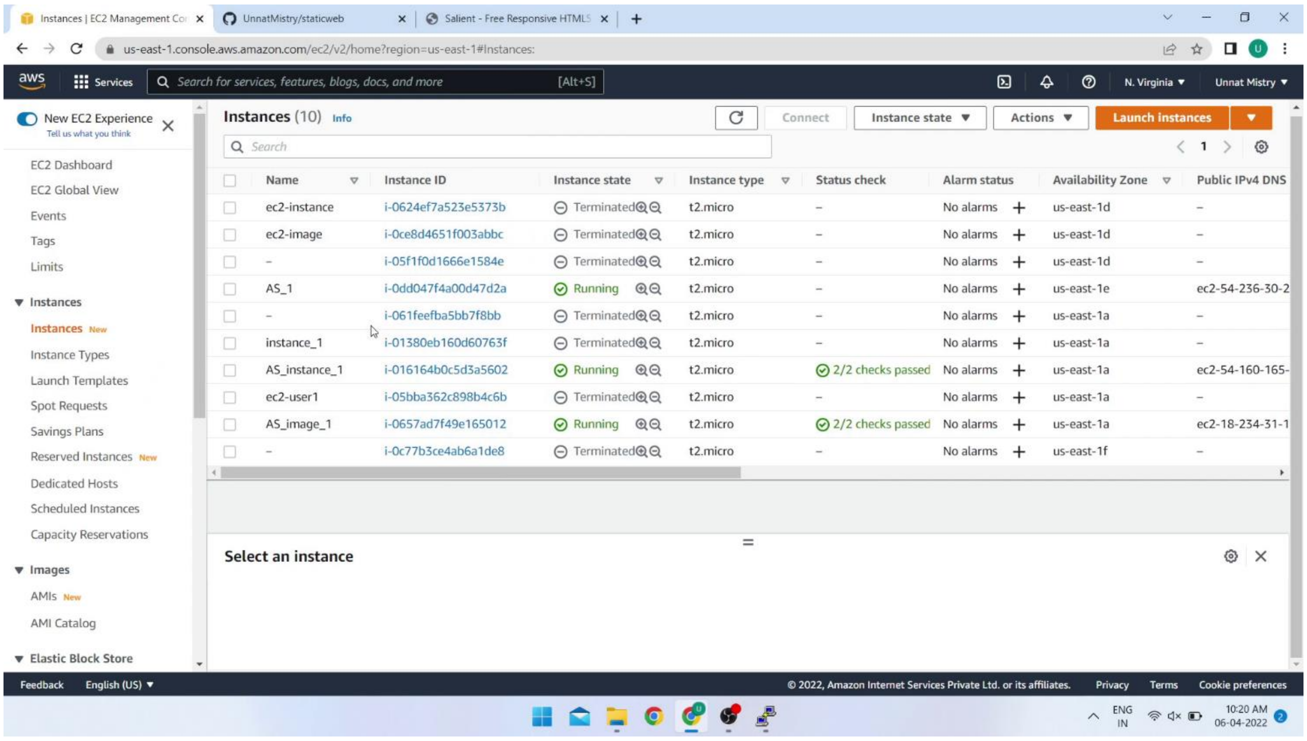
Instances need: 0 seconds warm up before including in metric

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

Instance scale-in protection - *optional*

An auto scaling group will be created. In the instances the AS_1 will create a new instance.

Connect to the new instance using putty.

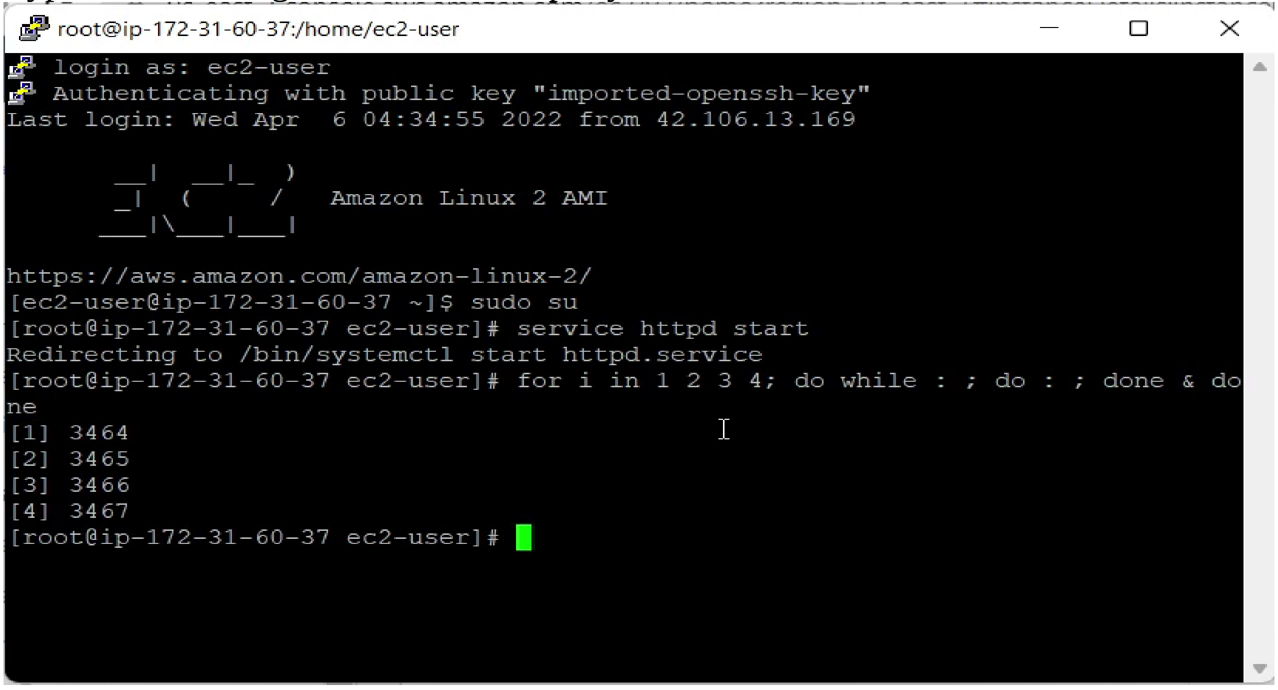


Instances (10) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
ec2-instance	i-0624ef7a523e5373b	Terminated	t2.micro	–	No alarms	us-east-1d	–
ec2-image	i-0ce8d4651f003abbc	Terminated	t2.micro	–	No alarms	us-east-1d	–
–	i-05f1f0d1666e1584e	Terminated	t2.micro	–	No alarms	us-east-1d	–
AS_1	i-0dd047f4a00d47d2a	Running	t2.micro	–	No alarms	us-east-1e	ec2-54-236-30-2
–	i-061feefba5bb7f8bb	Terminated	t2.micro	–	No alarms	us-east-1a	–
instance_1	i-01380eb160d60763f	Terminated	t2.micro	–	No alarms	us-east-1a	–
AS_instance_1	i-016164b0c5d3a5602	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-160-165-
ec2-user1	i-05bba362c898b4c6b	Terminated	t2.micro	–	No alarms	us-east-1a	–
AS_image_1	i-0657ad7f49e165012	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-18-234-31-1
–	i-0c77b3ce4ab6a1de8	Terminated	t2.micro	–	No alarms	us-east-1f	–

Select an instance

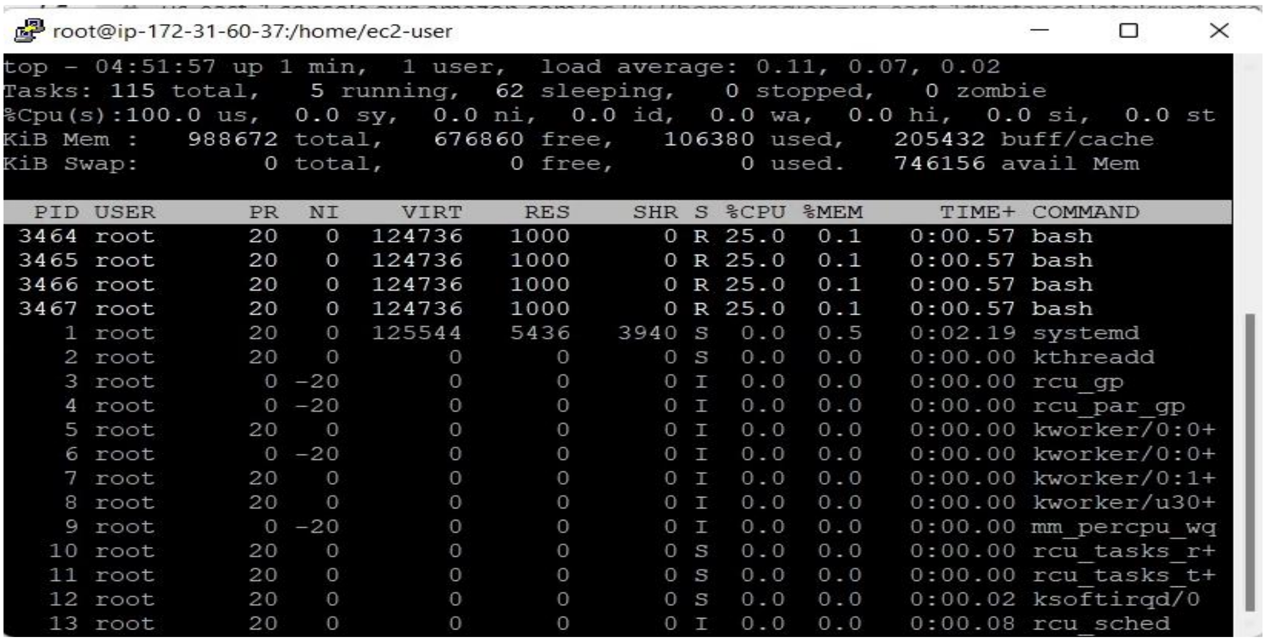
Type the following command in the putty console:



root@ip-172-31-60-37:/home/ec2-user

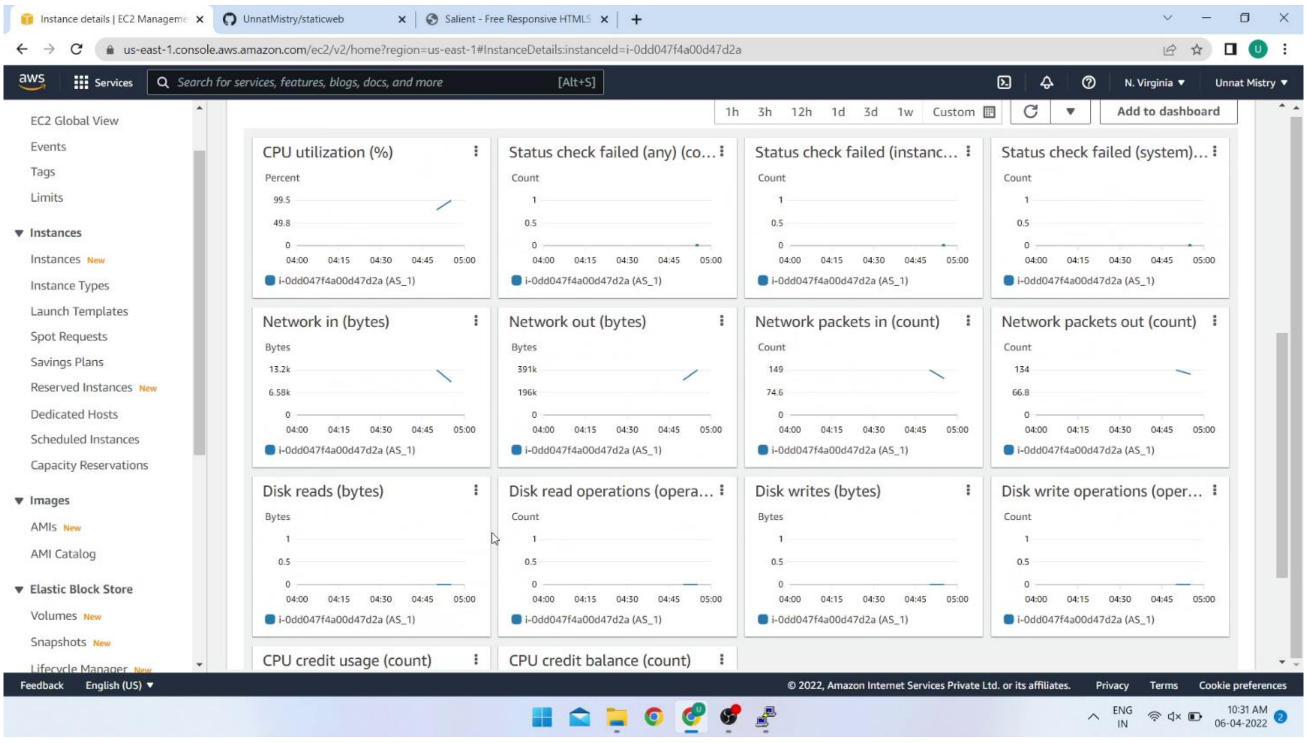
```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Wed Apr  6 04:34:55 2022 from 42.106.13.169

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-60-37 ~]$ sudo su
[root@ip-172-31-60-37 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-60-37 ec2-user]# for i in 1 2 3 4; do while : ; do : ; done & do
ne
[1] 3464
[2] 3465
[3] 3466
[4] 3467
[root@ip-172-31-60-37 ec2-user]#
```

Then enter top command to see the processes:



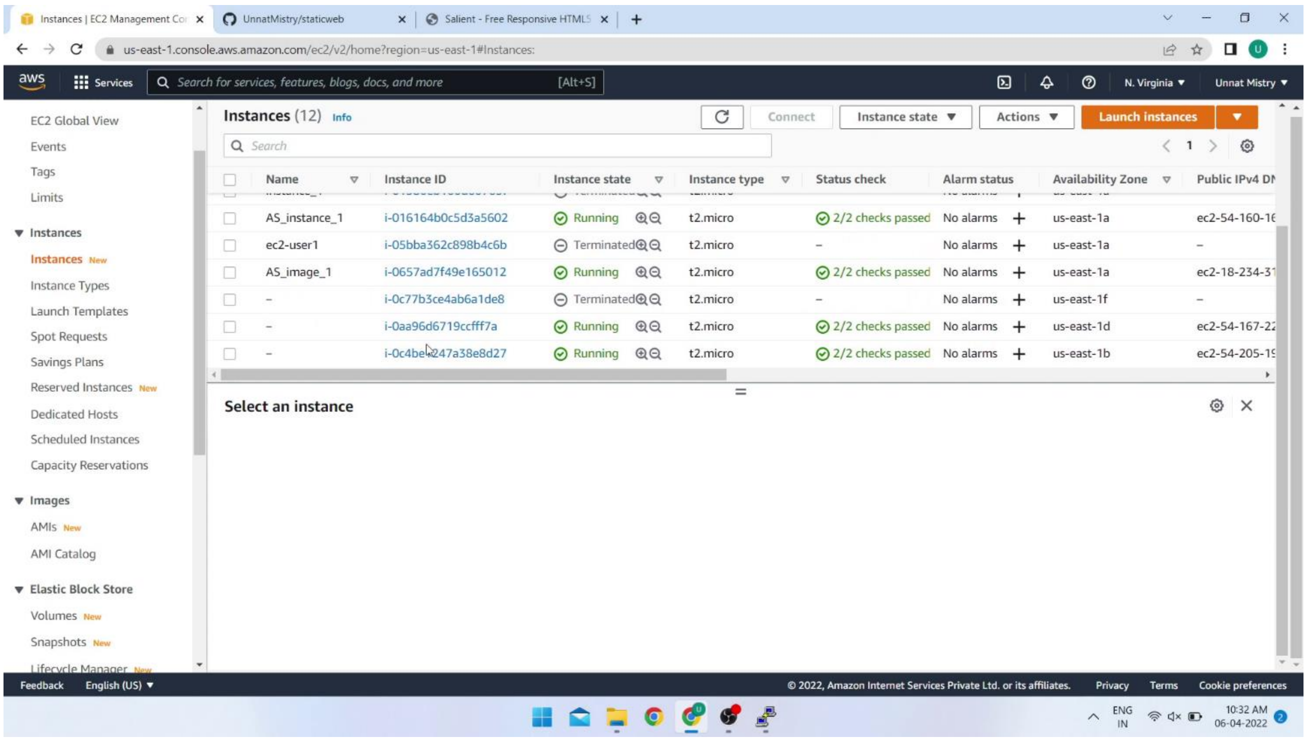
```
root@ip-172-31-60-37:/home/ec2-user
top - 04:51:57 up 1 min,  1 user,  load average: 0.11, 0.07, 0.02
Tasks: 115 total,   5 running,  62 sleeping,   0 stopped,   0 zombie
%Cpu(s):100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   988672 total,   676860 free,   106380 used,   205432 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   746156 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 3464 root      20   0  124736   1000      0 R  25.0  0.1   0:00.57 bash
 3465 root      20   0  124736   1000      0 R  25.0  0.1   0:00.57 bash
 3466 root      20   0  124736   1000      0 R  25.0  0.1   0:00.57 bash
 3467 root      20   0  124736   1000      0 R  25.0  0.1   0:00.57 bash
    1 root      20   0  125544   5436   3940 S   0.0  0.5   0:02.19 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_par_gp
    5 root      20   0       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0+
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0+
    7 root      20   0       0      0      0 I   0.0  0.0   0:00.00 kworker/0:1+
    8 root      20   0       0      0      0 I   0.0  0.0   0:00.00 kworker/u30+
    9 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_wq
   10 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_tasks_r+
   11 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_tasks_t+
   12 root      20   0       0      0      0 S   0.0  0.0   0:00.02 ksoftirqd/0
   13 root      20   0       0      0      0 I   0.0  0.0   0:00.08 rcu_sched
```

Monitoring CPU utilization:



Instance details | EC2 Manageme
UnnatMistry/staticweb
Salient - Free Responsive HTML5
us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-0dd047f4a00d47d2a
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
Unnat Mistry
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes New
Snapshots New
Lifecycle Manager New
1h 3h 12h 1d 3d 1w Custom
Add to dashboard
CPU utilization (%)
Status check failed (any) (co...
Status check failed (instanc...
Status check failed (system)...
Network in (bytes)
Network out (bytes)
Network packets in (count)
Network packets out (count)
Disk reads (bytes)
Disk read operations (opera...
Disk writes (bytes)
Disk write operations (oper...
CPU credit usage (count)
CPU credit balance (count)
i-0dd047f4a00d47d2a (AS_1)
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
10:31 AM
06-04-2022

New instances will be created automatically as the CPU utilization of created instance exceeds 100:

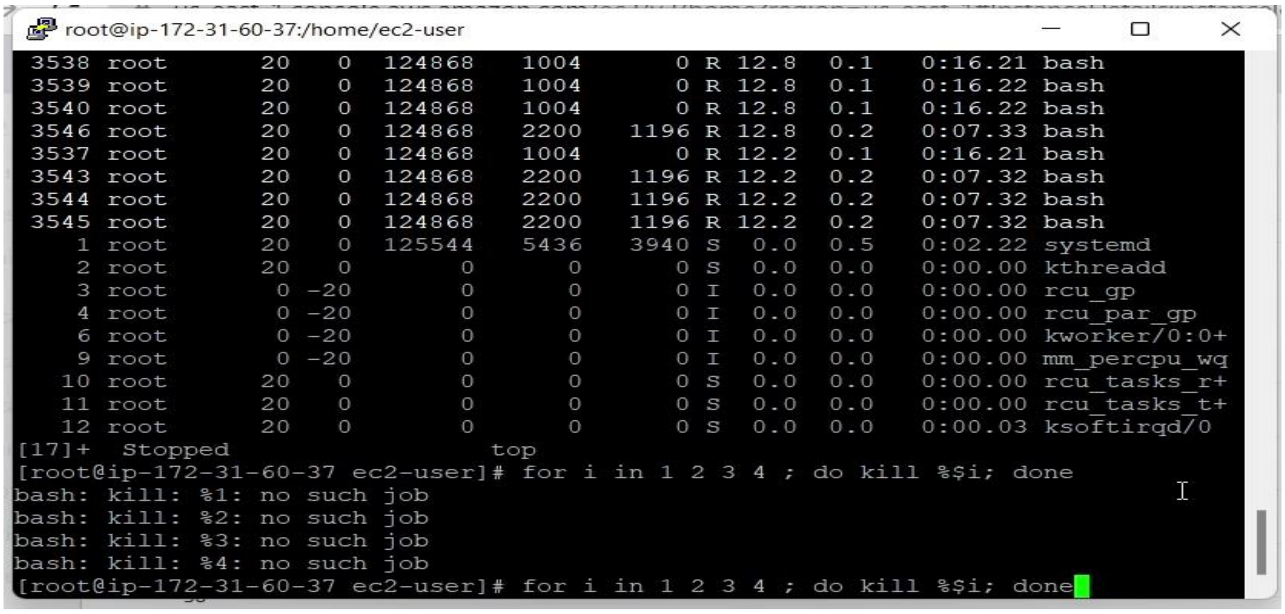


Instances (12) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
AS_instance_1	i-016164b0c5d3a5602	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-160-16
ec2-user1	i-05bba362c898b4c6b	Terminated	t2.micro	–	No alarms	us-east-1a	–
AS_image_1	i-0657ad7f49e165012	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-18-234-31
–	i-0c77b3ce4ab6a1de8	Terminated	t2.micro	–	No alarms	us-east-1f	–
–	i-0aa96d6719ccfff7a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-54-167-22
–	i-0c4be247a38e8d27	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-205-19

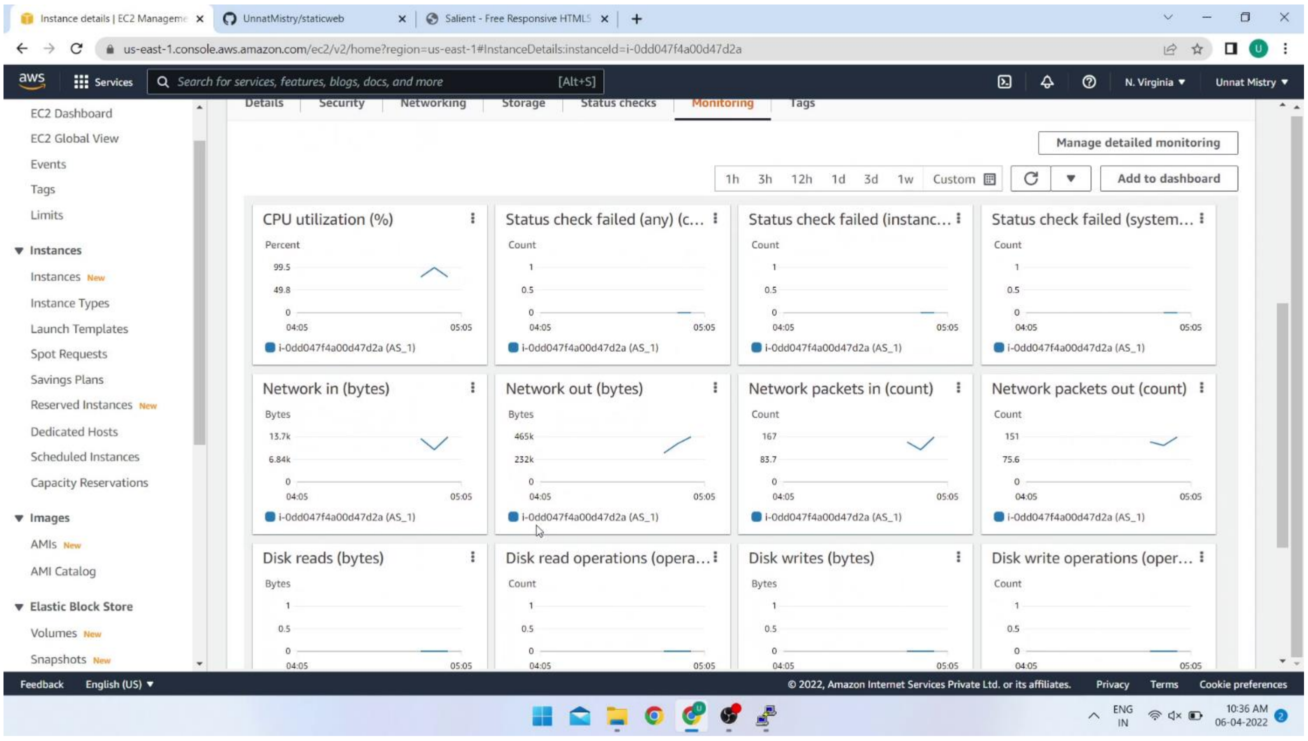
Select an instance

To decrease the load enter following command in putty:

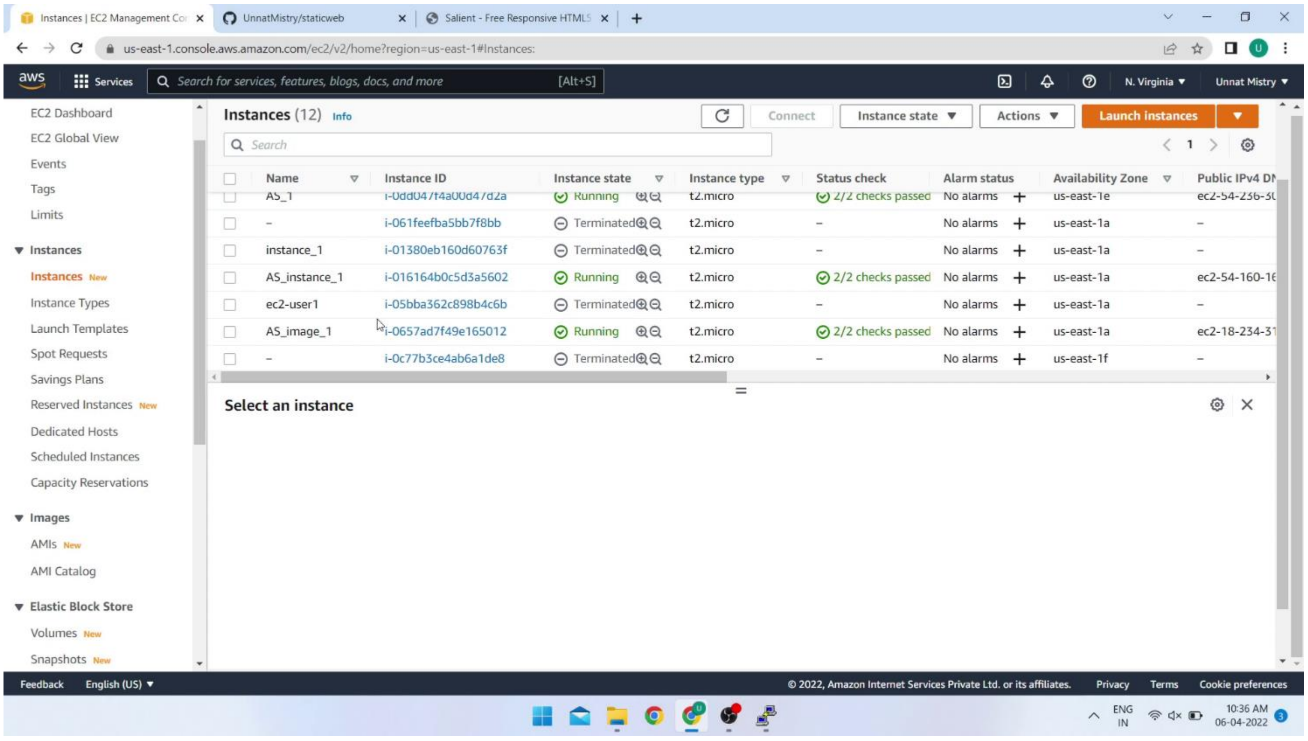


```
 3538 root      20   0  124868   1004      0 R 12.8  0.1   0:16.21 bash
 3539 root      20   0  124868   1004      0 R 12.8  0.1   0:16.22 bash
 3540 root      20   0  124868   1004      0 R 12.8  0.1   0:16.22 bash
 3546 root      20   0  124868   2200   1196 R 12.8  0.2   0:07.33 bash
 3537 root      20   0  124868   1004      0 R 12.8  0.1   0:16.21 bash
 3543 root      20   0  124868   2200   1196 R 12.2  0.2   0:07.32 bash
 3544 root      20   0  124868   2200   1196 R 12.2  0.2   0:07.32 bash
 3545 root      20   0  124868   2200   1196 R 12.2  0.2   0:07.32 bash
    1 root      20   0  125544   5436   3940 S  0.0  0.5   0:02.22 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0+
    9 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
   10 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_tasks_r+
   11 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_tasks_t+
   12 root      20   0       0      0      0 S  0.0  0.0   0:00.03 ksoftirqd/0
[17]+  Stopped                 top
[root@ip-172-31-60-37 ec2-user]# for i in 1 2 3 4 ; do kill %$i; done
bash: kill: %1: no such job
bash: kill: %2: no such job
bash: kill: %3: no such job
bash: kill: %4: no such job
[root@ip-172-31-60-37 ec2-user]# for i in 1 2 3 4 ; do kill %$i; done
```

Monitoring CPU utilization:


Monitoring
Manage detailed monitoring
1h 3h 12h 1d 3d 1w Custom
Add to dashboard
CPU utilization (%)
Status check failed (any)
Status check failed (instance)
Status check failed (system)
Network in (bytes)
Network out (bytes)
Network packets in (count)
Network packets out (count)
Disk reads (bytes)
Disk read operations
Disk writes (bytes)
Disk write operations
i-0dd047f4a00d47d2a (AS_1)

As the load decreases, the new created instances will be automatically terminated.



Instances (12)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
AS_1	i-0dd047f4a00d47d2a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1e	ec2-54-236-5
–	i-061feefba5bb7f8bb	Terminated	t2.micro	–	No alarms	us-east-1a	–
instance_1	i-01380eb160d60763f	Terminated	t2.micro	–	No alarms	us-east-1a	–
AS_instance_1	i-016164b0c5d3a5602	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-160-16
ec2-user1	i-05bba362c898b4c6b	Terminated	t2.micro	–	No alarms	us-east-1a	–
AS_image_1	i-0657ad7f49e165012	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-18-234-31
–	i-0c77b3ce4ab6a1de8	Terminated	t2.micro	–	No alarms	us-east-1f	–

Select an instance

END