

# Create and Configure the Auto Scaling Group in EC2

**Auto Scaling** is an Amazon Web Service it allows instances to scale when traffic or CPU load increases.

Auto-scaling is a service that monitors all instances that are configured into the Auto Scaling group and ensures that loads are balanced in all instances.

Depending on the load scaling group, increase the instance according to the configuration. When we created the auto-scaling group, we configured the Desired capacity, Minimum capacity, maximum capacity, and CPU utilization. If CPU utilization increases by 60% in all instances, one more instance is created, and if CPU utilization decreases by 30% in all instances, one instance is terminated. These are totally up to us; what is our requirement. If any Instance fails due to any reason, then the Scaling group maintains the Desired capacity and starts another instance.

The auto-scaling group follows Horizontal Scaling. This service is very important for us nowadays because we do not need to create new instances manually and do not require manual monitoring.

## Benefits of Auto Scaling

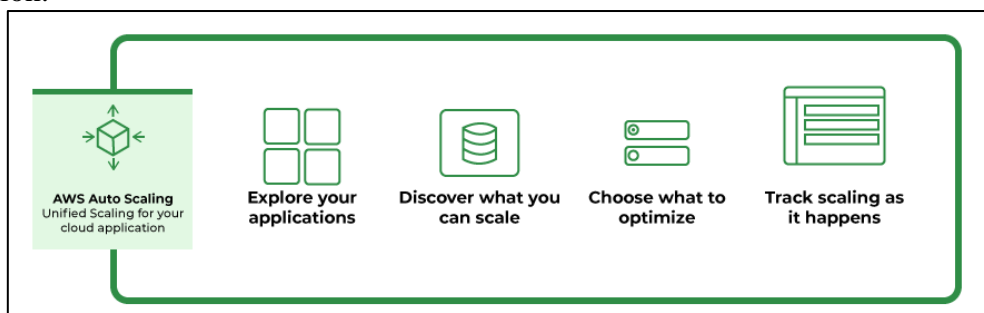
**Dynamical scaling:** AWS auto-scaling service doesn't required any type of manual intervention it will automatically scale the application down and up by depending up on the incoming traffic.

**Pay For You Use:** Because of auto scaling the resource will be utilised in the optimised way where the demand is low the resource utilisation will be low and the demand will high the resource utilisation will increase so the AWS is going to charge you only for the amount of resources you really used.

**Automatic Performance Maintenance:** AWS autoscaling maintains the optimal application performance with considering the workloads it will ensures that the application is running to desired level which will decrease the latency and also the capacity will be increased by based on your application

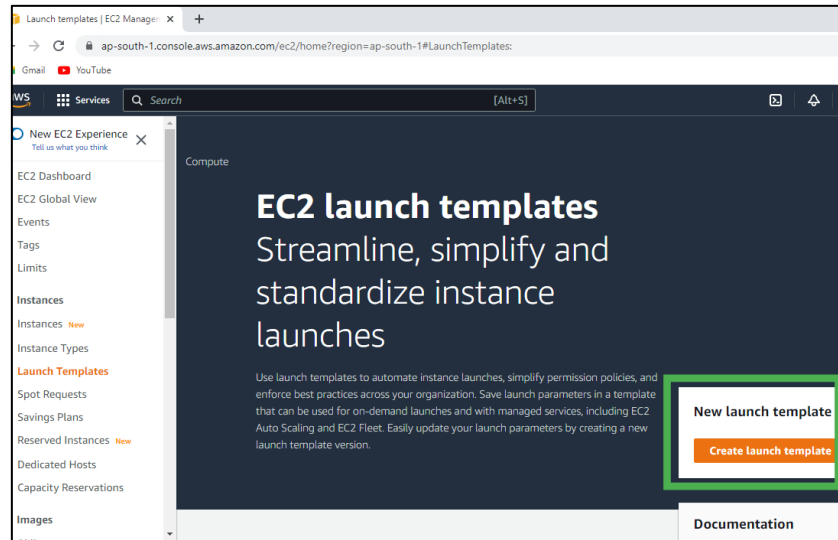
## How AWS Auto Scaling Works?

AWS autoscaling will scale the application based on the load of application. Instead of scaling manually AWS auto scaling will scale the application automatically when the incoming traffic is high it will scale up the application and when the traffic is low it will scale down the application.



## Steps To create Auto Scaling Launch Template

- Login to your AWS Account.
- Click on the EC2(Elastic Cloud Computing) in the homepage search bar.
- Scroll Down and click on the **Launch Templates** and click on the **Create launch template**



- Type the Template name.

**Create launch template**

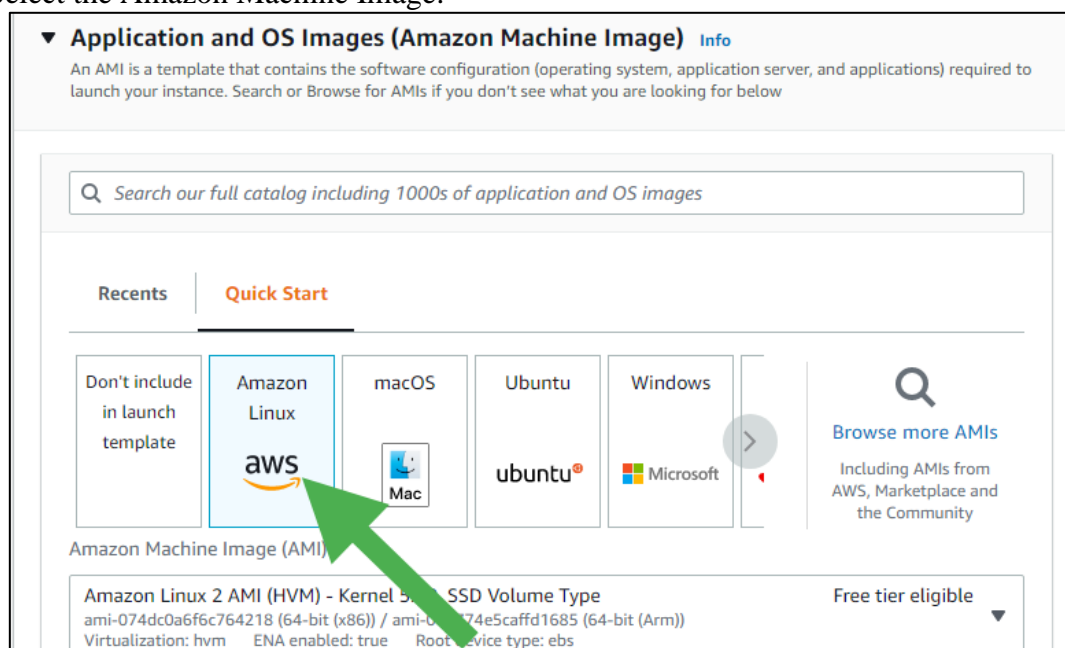
Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

**Launch template name and description**

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '%', '@'.

- Select the Amazon Machine Image.



- Select the Instance Type and Key pair.

▼ Instance type [Info](#) | [Get advice](#)

Advanced

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0124 USD per Hour On-Demand Windows base pricing: 0.017 USD per Hour  
On-Demand RHEL base pricing: 0.0268 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour  
On-Demand SUSE base pricing: 0.0124 USD per Hour

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

linux114

[Create new key pair](#)

- In Network Settings, Provide Your Subnet or Create a new one & Select the Security Group or Create the new one.

▼ Network settings [Info](#)

Subnet | [Info](#)

subnet-006b586c0d566eaa Subnet1-MyVPC

VPC: vpc-0343bfe016dcf2454 Owner: 245712304097 Availability Zone: ap-south-1a  
Zone type: Availability Zone IP addresses available: 59 CIDR: 192.168.0.0/26

[Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

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.

☒ Select existing security group

☐ Create security group

Common security groups | [Info](#)

Select security groups

MySG sg-077a397e58dc270bd ×

VPC: vpc-0343bfe016dcf2454

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

- In Advanced Details, provide your User Data it you have any and click Create launch template on the right-hand side.

User data - optional | [Info](#)

Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash
# Install Apache HTTP server
yum install httpd -y

# Enable Apache to start on boot
systemctl enable httpd

# Write content to the index.html file
echo "<h1>Welcome Vivek, $(hostname)</h1>" > /var/www/html/index.html

# Start the Apache HTTP server
systemctl start httpd
```

☐ User data has already been base64 encoded

Amazon Linux 2023 AMI 2023.6.2...[read more](#)

ami-07b69f62c1d38b012

Virtual server type (instance type)

t2.micro

Firewall (security group)

MySG

Storage (volumes)

1 volume(s) - 8 GiB

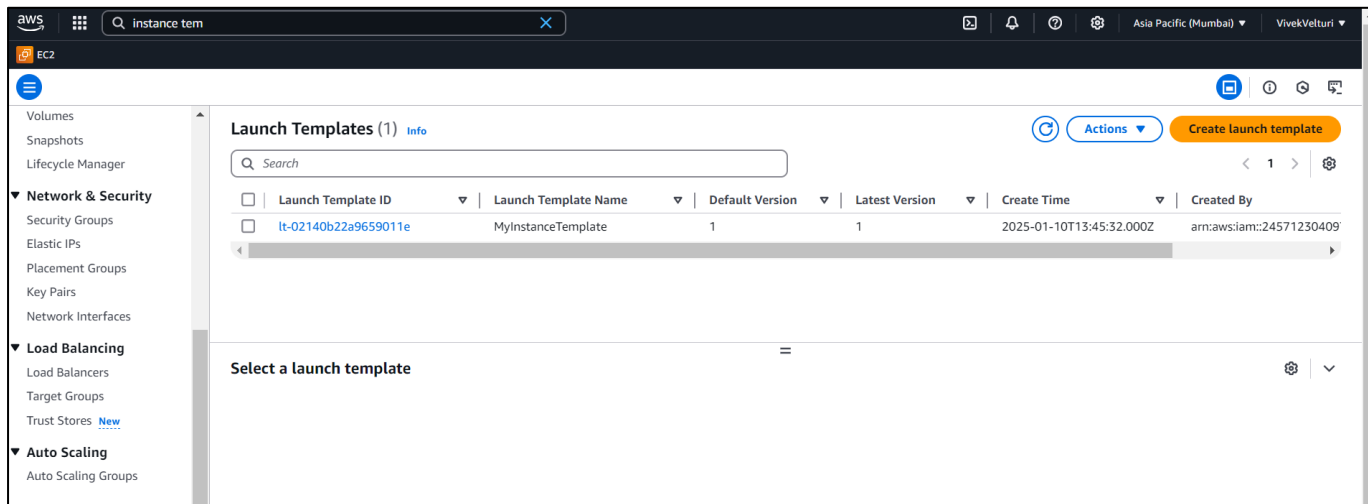
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IP addresses per month, and 100 GB of local disk space.

×

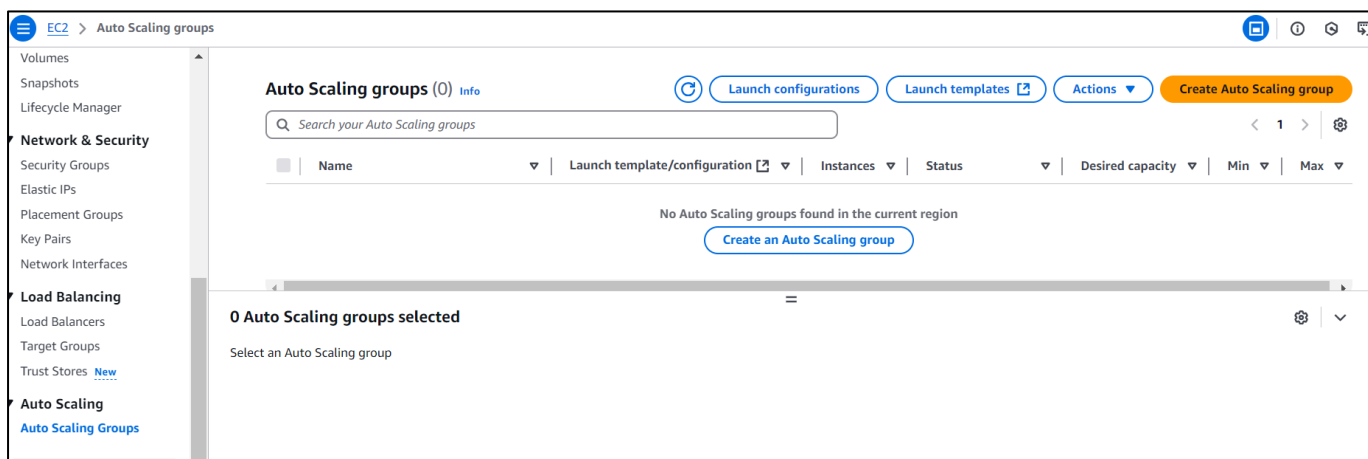
Cancel

Create launch template

- Your Instance template has been created, Now, scroll down and click on the Auto Scaling Groups.



## Create An Auto Scaling Group Using a Launch Template



- Type the Auto Scaling group name and select the launch template we have created.

### Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

#### 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 Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

**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.

- Click Next
- Select the VPC or go with the default VPC and also select the Availability zone.

### Network Info

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

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0343bfe016dcf2454 (MyVPC)  
192.168.0.0/24

Create a VPC

#### Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

ap-south-1a | subnet-006b586c0d566eeaa (Subnet1-MyVPC) X  
192.168.0.0/26

ap-south-1b | subnet-0b802820851e97d6e (Subnet2-MyVPC) X  
192.168.0.64/26

Create a subnet

- Select Availability Zone distribution to Balance best effort as default and click Next.

### Availability Zone distribution - new

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

☒ **Balanced best effort**  
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

☐ **Balanced only**  
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Cancel
Skip to review
Previous
Next

- In Integrate with other services, leave the settings as default and click Next or edit as per you requirement if your project demands for the use of load balancer, health check, etc.
- Configure the Group size and Scaling policies.
  - Select as per your requirement:
  - Desired: 2
  - Minimum: 1
  - Maximum: 3

### Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size.

#### Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported.

Units (number of instances)

#### Desired capacity

Specify your group size.

2

### Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

#### Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity  
1  
Equal or less than desired capacity

Max desired capacity  
3  
Equal or greater than desired capacity

- Select the Target tracking scaling policy and provide the values as per your requirement, in my scenario I have assigned the Metric type to be Average CPU utilization, and the Target value to be 50 while the instance warmup time to be 60 seconds.

**Choose whether to use a target tracking policy** | [Info](#)  
 You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☐ No scaling policies  
 Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☒ Target tracking scaling policy  
 Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

**Scaling policy name**

**Metric type** | [Info](#)  
 Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

**Target value**

**Instance warmup** | [Info](#)  
 seconds

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

- Click on the Create Auto Scaling Group.

**Step 6: Add tags** [Edit](#)

**Tags (0)**

Key	Value	Tag new instances
No tags		

[Preview code](#) [Cancel](#) [Previous](#) [Create Auto Scaling group](#)

- Now you can see the Auto Scaling is creating and it is also creating the desired state of the EC2 Instance

EC2

EC2 > Auto Scaling groups

**Auto Scaling groups (1)** [Info](#) [Launch configurations](#) [Launch templates](#) [Actions](#) [Create Auto Scaling group](#)

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	MyASG	MyInstanceTemplate   Version Default	0	Updating capacity...	2	1	3	ap-south-1b, ap-south-1a

- We selected the Desired state equal to 2 and we can see observe 2 Instances running in the EC2 panel.

Instances (2) <a href="#">Info</a>							
Find Instance by attribute or tag (case-sensitive)				Last updated 2 minutes ago	<a href="#">Connect</a>	<a href="#">Instance state</a>	<a href="#">Actions</a>
<a href="#">Instance state = running</a> <a href="#">Clear filters</a>				All states			
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>		i-0868e7dbeea43ed5c	<span>Running</span>	t2.micro	<span>2/2 checks passec</span>	<a href="#">View alarms +</a>	ap-south-1b
<input type="checkbox"/>		i-09577037faaeeb5aa	<span>Running</span>	t2.micro	<span>2/2 checks passec</span>	<a href="#">View alarms +</a>	ap-south-1a

- Now if we manually Terminate the Instance created, The ASG which is monitoring will again start launching instances as per our requirement given, in our case we have requested for a minimum capacity of 1 and desirable capacity of 2 and maximum capacity of 3.
- Upon termination of the instances manually, if we look into the activity of our ASG created we can observe that a new instance has been launched.

Activity history (2)				
Filter activity history				
Status	Description	Cause	Start time	End time
<span>Successful</span>	Launching a new EC2 instance: i-09577037faaeeb5aa	At 2025-01-10T14:26:57Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-01-10T14:27:02Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 January 10, 07:57:04 PM +05:30	2025 January 10, 07:57:35 PM +05:30

Instances (3) <a href="#">Info</a>							
Find Instance by attribute or tag (case-sensitive)				Last updated less than a minute ago	<a href="#">Connect</a>	<a href="#">Instance state</a>	<a href="#">Actions</a>
<a href="#">Instance state = running</a> <a href="#">Clear filters</a>				All states			
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>		i-04d7bcb6b96d5a012	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	ap-south-1b
<input type="checkbox"/>		i-0868e7dbeea43ed5c	<span>Terminated</span>	t2.micro	-	<a href="#">View alarms +</a>	ap-south-1b
<input type="checkbox"/>		i-09577037faaeeb5aa	<span>Terminated</span>	t2.micro	-	<a href="#">View alarms +</a>	ap-south-1a

- After a while the launch of a second instance by our ASG has also been observed

Instances (2) <a href="#">Info</a>							
Find Instance by attribute or tag (case-sensitive)				Last updated less than a minute ago	<a href="#">Connect</a>	<a href="#">Instance state</a>	<a href="#">Actions</a>
<a href="#">Instance state = running</a> <a href="#">Clear filters</a>				All states			
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>		i-04d7bcb6b96d5a012	<span>Running</span>	t2.micro	<span>2/2 checks passec</span>	<a href="#">View alarms +</a>	ap-south-1b
<input type="checkbox"/>		i-02e2887740bcc63ed	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	ap-south-1a

- I have knowingly given 2 subnets, one private and one public, the instance which was launched with the private subnet was shutdown automatically as it was not able to connect to internet due to lack of an IGW not being assigned to it. While the instance launched with public subnet was observed to be healthy and running.





```

CPU[|||||100.0%] Tasks: 45, 186 thr, 68 kthr; 1 running
Mem[|||||142M/949M] Load average: 2.18 0.58 0.21
Swp[|||||0K/0K] Uptime: 00:20:09

  CPU 170
  PID USER      PR  NI  VIRT  RES  SHR  S  CPU% MEM%   TIME+  Command
27493 ec2-user    20   0  3516  108   0  R   25.0  0.0  0:10.44 stress --cpu 4 --timeout 300s
27496 ec2-user    20   0  3516  108   0  R   25.0  0.0  0:10.44 stress --cpu 4 --timeout 300s
27495 ec2-user    20   0  3516  108   0  R   23.7  0.0  0:10.42 stress --cpu 4 --timeout 300s
1      root      20   0  103M  17408 10600  S  0.0  1.8  0:01.11 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
1093   root      20   0  52548 14900 13788  S  0.0  1.5  0:00.29 /usr/lib/systemd/systemd-journald
1767   root      20   0  31312 11328  8228  S  0.0  1.2  0:00.06 /usr/lib/systemd/systemd-udevd
1770   systemd-re 20   0  21236 14584 11096  S  0.0  1.5  0:00.05 /usr/lib/systemd/systemd-resolved
1789   root      16  -4  20228  2344  1616  S  0.0  0.2  0:00.00 /sbin/auditd
1790   root      16  -4  20228  2344  1616  S  0.0  0.2  0:00.00 /sbin/auditd
1961   root      20   0  15300  6560  5712  S  0.0  0.7  0:00.00 /usr/bin/systemd-inhibit --what=handle-suspend-key:handle-hibernate-key --who=noah --why=acpid instead --m
1964   libstorage 20   0  2760  1980  1820  S  0.0  0.2  0:00.01 /usr/bin/lsm -d
1966   root      20   0  89008  5944  4952  S  0.0  0.6  0:10.02 /usr/sbin/rngd -f -x pkcs11 -x nist
1968   root      20   0  15784  7716  6752  S  0.0  0.8  0:00.01 /usr/lib/systemd/systemd-homed
1969   root      20   0  17792  9844  7640  S  0.0  1.0  0:00.06 /usr/lib/systemd/systemd-logind
1973   dbus      20   0  8352  3852  3248  S  0.0  0.4  0:00.01 /usr/bin/dbus-broker-launch --scope system --audit
1974   systemd-ne 20   0  2308  9736  8452  S  0.0  1.0  0:00.03 /usr/lib/systemd/systemd-networkd
1992   dbus      20   0  5400  3044  2328  S  0.0  0.3  0:00.04 dbus-broker --log 4 --controller 9 --machine-id ec241e1543d30b4be1b7ef3a715ee600 --max-bytes 536870912 --m
1993   root      20   0  2672  1140  1056  S  0.0  0.1  0:00.00 /usr/sbin/acpid -f

```

Find Instance by attribute or tag (case-sensitive)		All states		Launch instances	
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	i-04d7bcb6b96d5a012	Terminated	t2.micro	-	View alarms +
<input type="checkbox"/>	i-0868e7dbaea43ed5c	Terminated	t2.micro	-	View alarms +
<input type="checkbox"/>	i-0c7874d1fdf48b197	Running	t2.micro	2/2 checks passed	View alarms +
<input checked="" type="checkbox"/>	i-02e2887740bcc63ed	Running	t2.micro	2/2 checks passed	View alarms +
<input type="checkbox"/>	i-09577037faaeb5aa	Terminated	t2.micro	-	View alarms +

- A new instance has been created to balance out the load we have applied on the old instance.

## Delete Your Resources

### Delete your ASG:

- In your AWS Management Console, head to **Amazon EC2 and Auto Scaling Groups**
- Select Actions on the right-hand side, Click **Delete**.
- Type delete in the pop up window and Click **Delete**.
- The Instances launched by the ASG will also be deleted automatically as we can go to the EC2 Dashboard and observe if any instances are still Running.

Find Instance by attribute or tag (case-sensitive)		All states		Launch instances	
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
No matching instances found					

### Delete your instance template:

- In your AWS Management Console, head to **Amazon EC2 and Launch Templates**.
- Select Actions on the right-hand side, Click **Delete template**.

- Type delete in the pop up window and Click **Delete**.