

# Scaling the EC2 Instance Based on Monitored CloudWatch Metrics

## Project Agenda

Use Auto Scaling to manage the EC2 instances and capture the metrics in the CloudWatch.

## Description

Let's take the case of Hotstar — a platform that provides on-demand video streaming services. The more the users join the streaming service platform, the more the resources in terms of servers (EC2 in AWS) Hotstar needs to invest in. This way, the load is distributed across different servers and leads to a jitter-free experience for the customers while watching the videos. Another example is Amazon Prime Day, where a bevy of customers access the amazon.com site. Depending on the number of customers logging into the amazon.com site, Amazon would like to add more servers for better customer experience.

Both the above actions lead to increased customer satisfaction, which will eventually boost profits for the companies. This feature of adding and removing servers is called Dynamic Scaling and is a unique feature of the Cloud. Simply put, the users of the Cloud can scale to thousands of servers and scale down when appropriate and pay for what they use. However, that flexibility to add/subtract servers does not come with the on-premise servers, which is why the cost is always fixed. Also, during the slack time, many resources remain under-utilized which is a wastage of CAPEX. One way of adding and deleting the EC2 instances is to do it manually which may lead to extra manual effort, increase in costs, and inaccurate results.

Another approach is to use Auto Scaling to manage the EC2 instances automatically. As Auto Scaling adds more EC2 instances, the software/application installation and configuration can be automated using the AMI (Amazon Machine Images). In the previous use case, we have seen how to capture custom metrics (number of users logged in) in the CloudWatch. Here, we would need the same metric to manage (add/delete) the EC2 instances depending on the number of users logged into the website.

## Tools Required

AWS Services - CloudWatch, Auto Scaling, EC2 and ELB.

## Expected Deliverables

- Use Auto Scaling to manage the EC2 instances
- Use EC2 instance and capture the metrics in the CloudWatch

## Steps

Following are the steps to be performed to complete this hands-on exercise:

1. Deploy a demo web application on an Amazon EC2 instance
2. Create a custom AMI
3. Create a Target Group
4. Create an Application Load Balancer
5. Create a launch template
6. Create an Auto Scaling Group
7. Send multiple requests to the load balancer's DNS name

8. View Activity Logs to check for newly launched instances
9. Clean Up

## Deploy a Demo Web Application on an Amazon EC2 Instance

First of all, we shall create a custom AMI encapsulating the OS, scripts, software, applications and services in a single package.

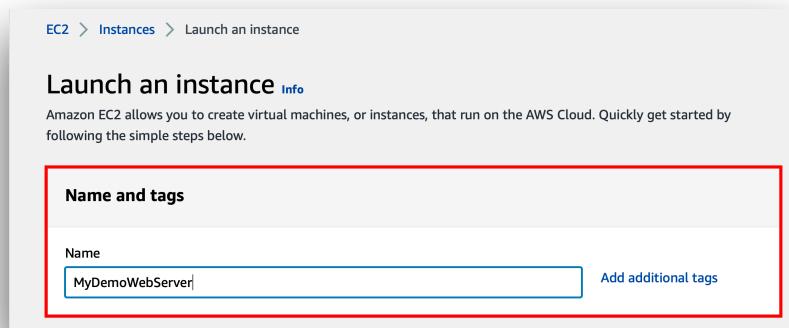
1. Launch an EC2 instance using a bootstrap script. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main area has sections for 'Resources' (listing 0 instances, 0 key pairs, etc.) and 'Service health' (status: 'This service is operating normally'). A prominent 'Launch instance' button is located in the 'Launch instance' section. To the right, there's an 'Account attributes' panel and an 'Explore AWS' panel with various promotional links.

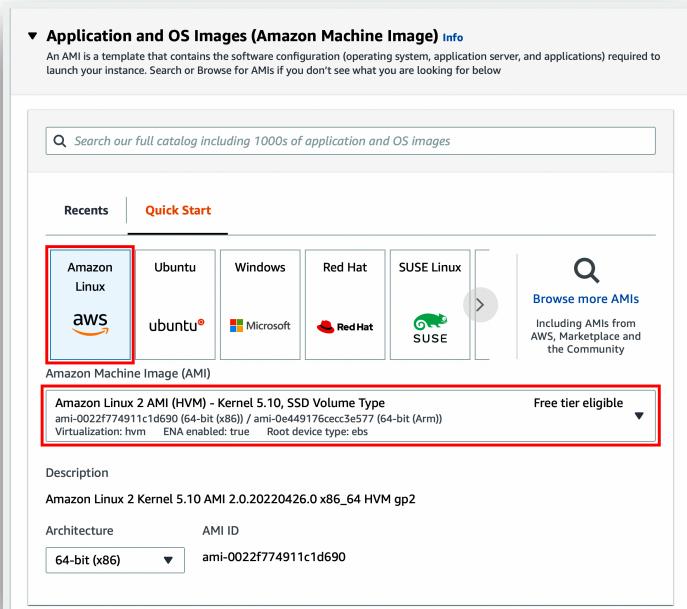
2. From the Amazon EC2 console dashboard, choose **Launch instance**.

This screenshot is similar to the previous one but with a red box highlighting the 'Launch instance' button in the 'Launch instance' section of the main dashboard. The rest of the interface, including the resource statistics and service health, remains the same.

3. For **Name**, enter a descriptive name for the instance. If you don't specify a name, the instance can be identified by its ID, which is automatically generated when you launch the instance.

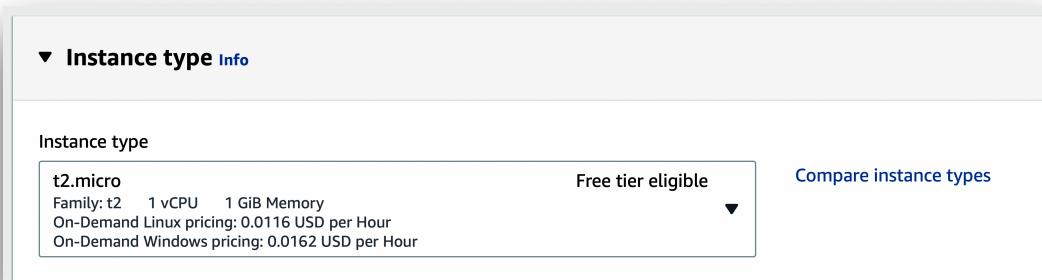


4. Under **Application and OS Images (Amazon Machine Image)**, choose **Quick Start**, and then choose the **Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type** for your instance.



5. For **Instance type**, select the instance type for the instance.

If your AWS account is less than 12 months old, you can use Amazon EC2 under the Free Tier by selecting the **t2.micro** instance type (or the **t3.micro** instance type in Regions where **t2.micro** is unavailable).



6. For **Key pair name**, choose **Proceed without a key pair (Not recommended)**.

▼ 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 - *required*

Proceed without a key pair (Not recommended) Default value ▾ [Create new key pair](#)

- As you will not be connecting to this instance, it will be safe to launch it without a key pair.

7. Configure the **networking settings** as following:

- **Network:** Here the default VPC is selected automatically. You will be launching this EC2 instance in the same default VPC of the region you're working in.
- **Subnet:** You can launch an instance in a subnet associated with an Availability Zone, Local Zone, Wavelength Zone, or Outpost. For this hands-on, keep Subnet to **No preference (Default subnet in any availability zone)**.
- **Auto-assign Public IP:** Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address, and instances in a non-default subnet don't. You can select **Enable** or **Disable** to override the subnet's default setting. For this hands-on, keep **Auto-assign public IP** to **Enable**.

▼ Network settings [Edit](#)

Network  
vpc-0be3ab4f4d995cb4a | my\_default\_vpc

Subnet  
No preference (Default subnet in any availability zone)

Auto-assign public IP  
Enable

8. The launch instance wizard automatically defines the **launch-wizard-x** security group and creates an inbound rule to allow you to connect to your instance over SSH (port 22). Include the inbound rules as following:

- Since this instance is launched just for extracting an image, therefore there will be no need for you to connect to this instance via SSH. Uncheck **Allow SSH traffic from** option.
- Select **Allow HTTPs traffic from the internet** and **Allow HTTP traffic from the internet** rules to allow internet traffic. This will allow you to access the website to be launched upon this very EC2 instance.

### Security groups (Firewall) [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.

We'll create a new security group called '**launch-wizard-2**' with the following rules:

**Allow SSH traffic from**

Helps you connect to your instance

**Allow HTTPS traffic from the internet**

To set up an endpoint, for example when creating a web server

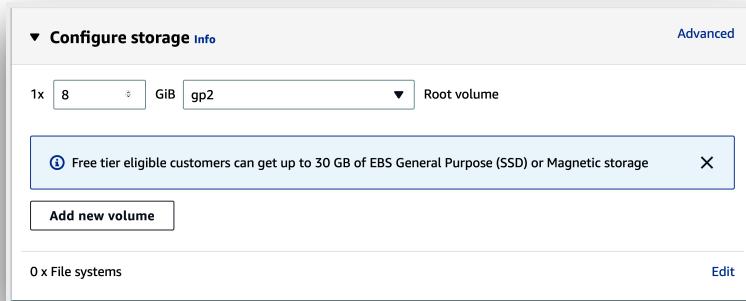
**Allow HTTP traffic from the internet**

To set up an endpoint, for example when creating a web server

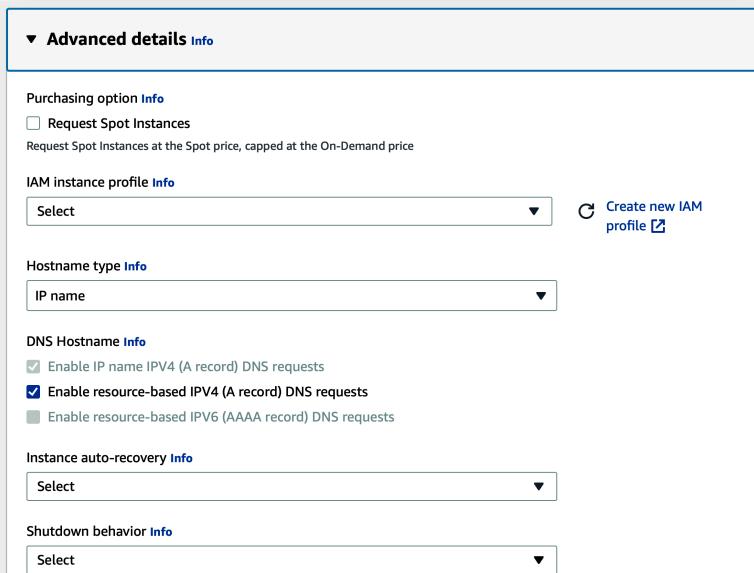
**⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

X

9. For **Configure storage**, accept default values. The AMI you selected includes one or more volumes of storage, including the root volume. By default, an 8 GiB of General Purpose SSD volume is attached to an Amazon Linux instance.



10. For **Advanced details**, expand the section to view the fields and specify any additional parameters for the instance.



- Accept all default parameters here except for **User data**. You can specify user data to configure an instance during launch, or to run a configuration script.
- Click [here](#) to download the bash script which will be used launch an Apache PHP web application on this EC2 instance.
- Copy and paste the contents of the download script within User data field.

User data [Info](#)

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
cd /var/www/html
echo "<html><h1>Let's understand the implementation of ELB and Auto Scaling services</h1></html>" > index.html
```

User data has already been base64 encoded

11. Use the **Summary** panel to specify the number of instances to launch, to review your instance configuration, and to launch your instances.

- Keep all settings to default and choose **Launch instance**.

Metadata response hop limit [Info](#)  
Select

Allow tags in metadata [Info](#)  
Select

User data [Info](#)

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
cd /var/www/html
echo "<html><h1>Let's understand the implementation of ELB and Auto Scaling services</h1></html>" > index.html
```

User data has already been base64 encoded

**▼ Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-079b5e5b3971bd10d

Virtual server type (instance type)  
t2.micro

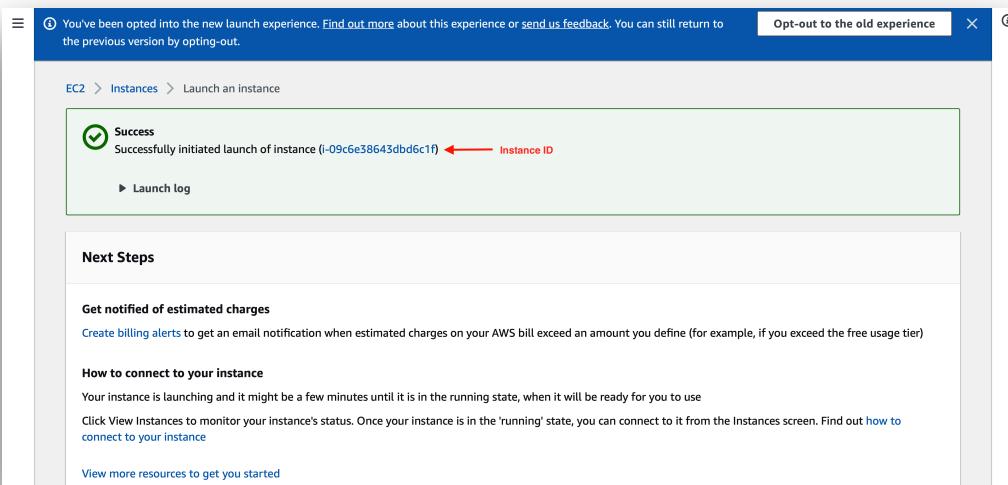
Firewall (security group)  
New security group

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

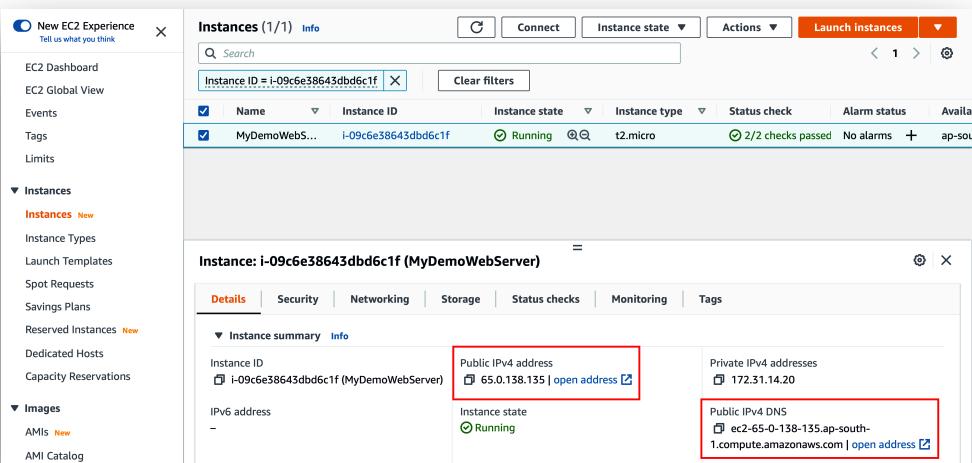
ⓘ Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the US East (N. Virginia) region) at no charge. [Learn more](#)

[Cancel](#) [Launch instance](#)

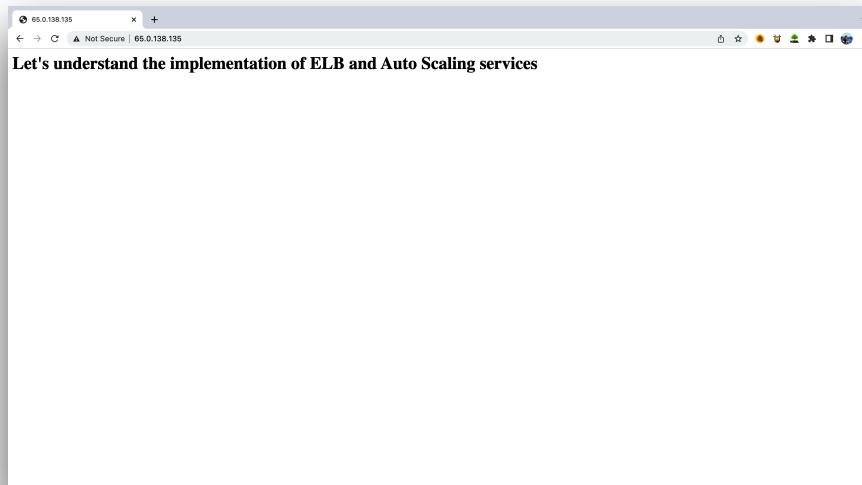
12. Click the Instance ID of the launched instance.



13. You will be straightaway sent to the **Instances** dashboard. Select the launched instance and look for **Public IPv4 address** and **Public IPv4 DNS** within **Details**.



14. Now, access the website using either the public IPv4 address or public IPv4 DNS name and you should get the following output.

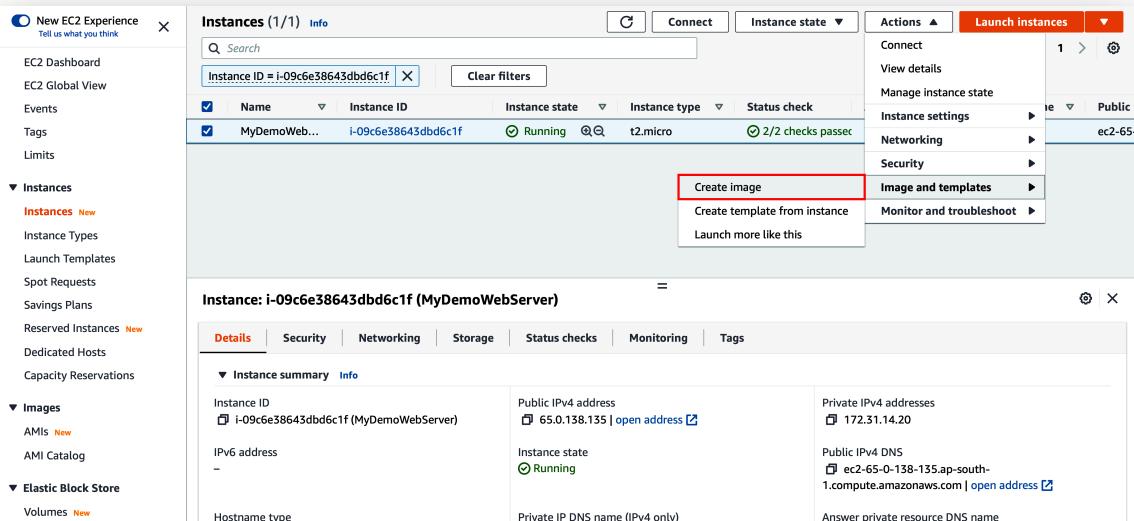


# Create a Custom AMI

15. In this step, you'll be creating a custom AMI from the instance launched in the previous step. After you customize the instance to suit your needs, create and register a new AMI, which you can use to launch new instances with these customizations. This custom image will be included as a part of the launch template to be used with Auto Scaling.

## To create an AMI from the instance

- A. In the navigation pane, choose **Instances**, select your instance, and then choose **Actions**, **Image and templates**, **Create image**.

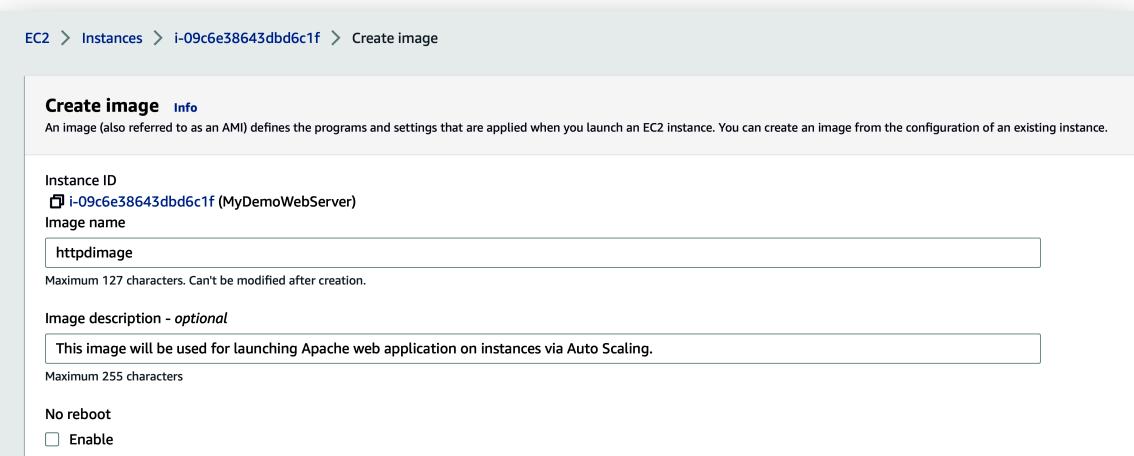


The screenshot shows the AWS EC2 Instances page. On the left, the navigation pane includes sections like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-options like Instances, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main area displays a table with one instance: MyDemoWeb... (Instance ID: i-09c6e38643dbd6c1f, State: Running, Type: t2.micro). A red box highlights the 'Create image' button in the Actions menu on the right. Below the table, a detailed view for the selected instance is shown, including its ID, public IP (65.0.138.135), private IP (172.31.14.20), and public DNS (ec2-65-0-138-135.ap-south-1.compute.amazonaws.com).

- B. On the **Create image** page, specify the following information, and then choose **Create image**.

- Image name** – A unique name for the image.
- Image description** – An optional description of the image, up to 255 characters.
- No reboot** – By default, when Amazon EC2 creates the new AMI, it reboots the instance so that it can take snapshots of the attached volumes while data is at rest, in order to ensure a consistent state. For the **No reboot** setting, you can select the **Enable** check box to prevent Amazon EC2 from shutting down and rebooting the instance.

For this hands-on exercise, make sure No reboot option is unchecked.



The screenshot shows the 'Create image' configuration dialog. It includes fields for Instance ID (selected: i-09c6e38643dbd6c1f), Image name (httpdimage), and Image description (This image will be used for launching Apache web application on instances via Auto Scaling.). A note states: 'An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.' At the bottom, there are two options for 'No reboot': 'Enable' (unchecked by default) and 'Disable'.

C. Accept default **Instance volumes** settings and skip adding **Tags**. Choose **Create image**.

The screenshot shows the 'Create Image' step in the AWS EC2 wizard. It displays the 'Instance volumes' configuration. Under 'Volume type', 'EBS' is selected. The 'Device' dropdown shows '/dev/x...'. The 'Snapshot' dropdown says 'Create new snapshot from...'. The 'Size' is set to 8. The 'Volume type' is 'EBS General Purpose S...'. 'IOPS' is 100, 'Throughput' is 0, and 'Delete on termination' is checked. 'Encrypted' is also checked. Below this, there's a note: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.' There are two radio button options for 'Tags - optional': 'Tag image and snapshots together' (selected) and 'Tag image and snapshots separately'. A note under the first option says 'Tag the image and the snapshots with the same tag.' A note under the second option says 'Tag the image and the snapshots with different tags.' Below these, it says 'No tags associated with the resource.' and 'Add tag' (with a note 'You can add up to 50 more tags.') At the bottom right are 'Cancel' and 'Create Image' buttons.

D. To view the status of your AMI while it is being created, in the navigation pane, choose **AMIs**. Initially, the status is pending but should change to available after a few minutes.

The screenshot shows the 'Amazon Machine Images (AMIs)' page in the AWS EC2 console. The left sidebar shows 'Images' with 'AMIs New' selected. The main table lists one AMI: 'Name' is 'ami-060c4064b506dfce0', 'AMI ID' is 'httpdimage', 'Source' is '149327762283/httpdimage', 'Owner' is '149327762283', 'Visibility' is 'Private', and 'Status' is 'Pending'. Below the table is a 'Select an AMI' section.

E. (Optional) To view the snapshot that was created for the new AMI, choose **Snapshots**. When you launch an instance from this AMI, we use this snapshot to create its root device volume.

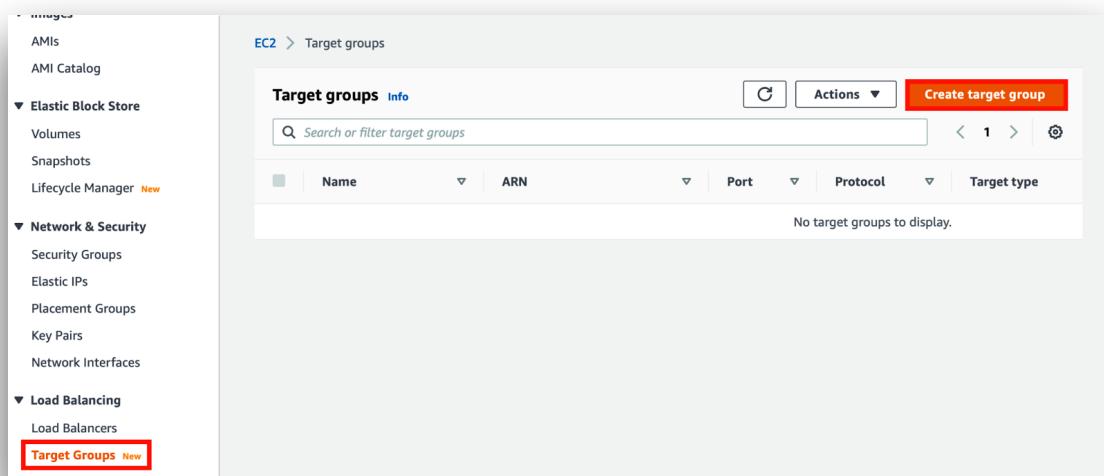
The screenshot shows the 'Snapshots' page in the AWS EC2 console. The left sidebar shows 'Elastic Block Store' with 'Snapshots New' selected. The main table lists one snapshot: 'Name' is 'snap-0f0432c327398edb1', 'Snapshot ID' is 'snap-0f0432c327398edb1', 'Size' is '8 GiB', 'Description' is 'Created by CreateImage(i....)', 'Storage...' is 'Standard', 'Snapshot status' is 'Pending', 'Started' is '2022/05/29 13:44 GMT+5...', and 'Progress' is 'Unavailable (0%)'. Below the table is a 'Select a snapshot above.' section.

## Create a Target Group and an Application Load Balancer

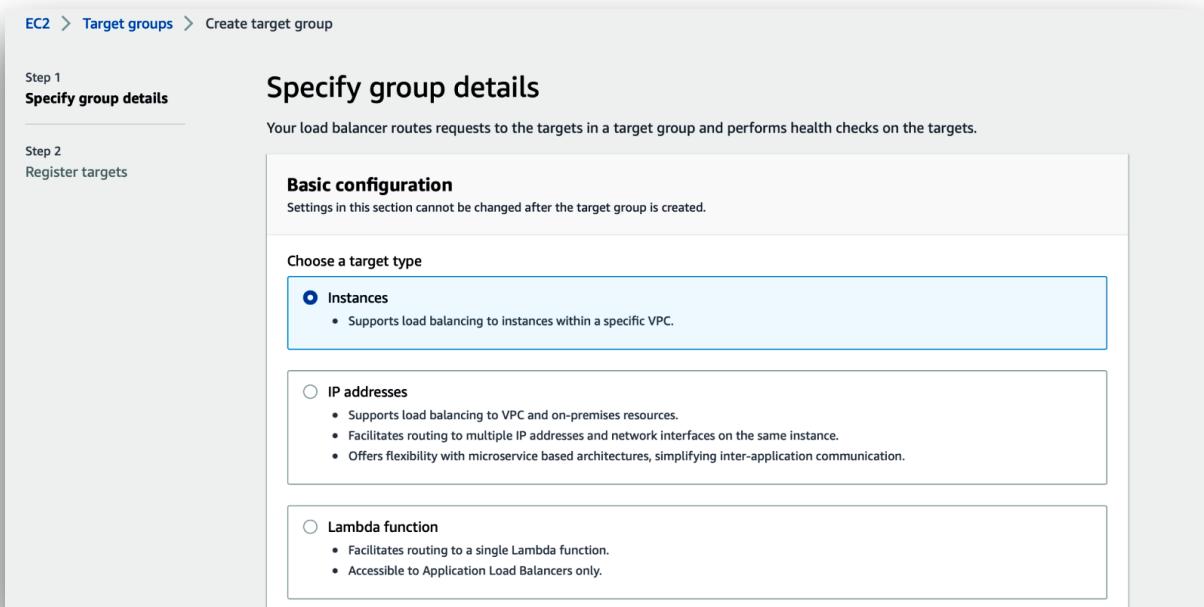
Each *target group* is used to route requests to one or more registered targets. When you create each listener rule, you specify a target group and conditions. When a rule condition is met, traffic is forwarded to the corresponding target group. You can create different target groups for different types of requests. For example, create one target group for general requests and other target groups for requests to the microservices for your application.

You define health check settings for your load balancer on a per target group basis. Each target group uses the default health check settings, unless you override them when you create the target group or modify them later on. After you specify a target group in a rule for a listener, the load balancer continually monitors the health of all targets registered with the target group that are in an Availability Zone enabled for the load balancer. The load balancer routes requests to the registered targets that are healthy.

16. On the navigation pane, under **Load Balancing**, choose **Target Groups**, and then choose **Create target group**.



17. For **Choose a target type**, select **Instances** to register targets by instance ID.



18. For **Target group name**, type a name for the target group. This name must be unique per region per account, can have a maximum of 32 characters, must contain only alphanumeric characters or hyphens, and must not begin or end with a hyphen.

- For **Protocol** and **Port**, accept the default values (Protocol as **HTTP** and Port as **80**).
- For **VPC**, select the default virtual private cloud (VPC).
- For **Protocol version**, accept the default value as **HTTP1**.

Target group name  
stockalb

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol      Port  
HTTP : 80

VPC  
Select the VPC with the instances that you want to include in the target group.  
my-default-vpc  
vpc-a1c94fc8  
IPv4: 172.31.0.0/16

Protocol version  
 **HTTP1**  
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.  
 **HTTP2**  
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.  
 **gRPC**  
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

19. Type **index.html** in ‘Health check path’ and click **Next**.

**Health checks**  
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol  
HTTP

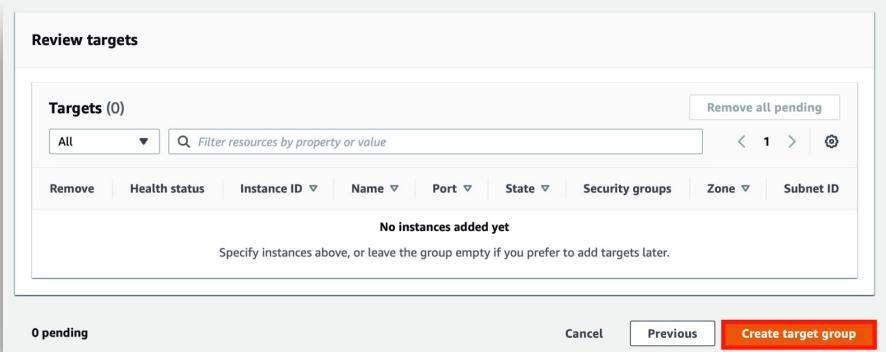
Health check path  
Use the default path of “/” to ping the root, or specify a custom path if preferred.  
/index.html  
Up to 1024 characters allowed.

► Advanced health check settings

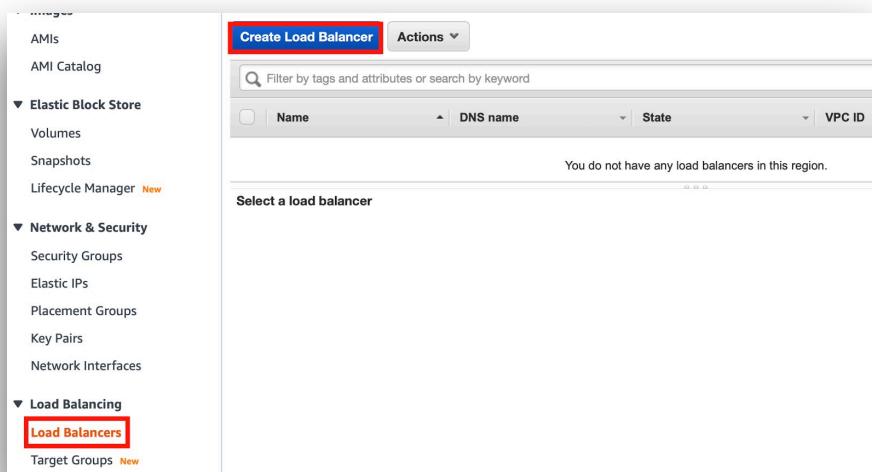
► Tags - optional  
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel      Next

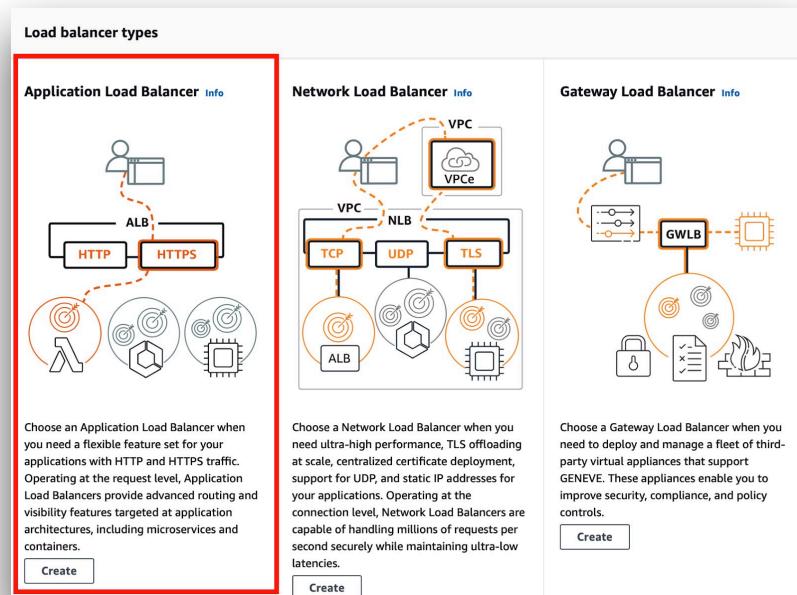
**20. Choose **Create target group**.**



**21. In the navigation pane, under **Load Balancing**, choose **Load Balancers**, and choose **Create Load Balancer**.**



**22. Under **Application Load Balancer**, choose **Create**.**



## 23. Basic configuration

- For **Load balancer name**, enter a name for your load balancer. For example, `my-alb`. The name of your Application Load Balancer must be unique within your set of Application Load Balancers and Network Load Balancers for the Region. Names can have a maximum of 32 characters, and can contain only alphanumeric characters and hyphens. They can not begin or end with a hyphen, or with `internal-`.
- For **Scheme**, choose **Internet-facing**. An internet-facing load balancer routes requests from clients to targets over the internet. An internal load balancer routes requests to targets using private IP addresses.
- For **IP address type**, choose **IPv4** or **Dualstack**. Use **IPv4** if your clients use IPv4 addresses to communicate with the load balancer. Choose **Dualstack** if your clients use both IPv4 and IPv6 addresses to communicate with the load balancer.

The screenshot shows the 'Create Application Load Balancer' wizard. The current step is 'Basic configuration'. It includes fields for 'Load balancer name' (set to 'stockalb'), 'Scheme' (set to 'Internet-facing'), and 'IP address type' (set to 'IPv4'). A note indicates that 'IPv4' is recommended for internal load balancers.

## 24. Network mapping

- For **VPC**, select the default VPC.
- For **Mappings**, select all the Availability Zones and corresponding subnets. Enabling multiple Availability Zones increases the fault tolerance of your applications.

The screenshot shows the 'Network mapping' configuration screen. It lists a single VPC ('my-default-vpc') and two Availability Zones ('eu-north-1a' and 'eu-north-1b'). Under each AZ, a subnet ('subnet-3677f35f') is selected. The 'IPv4 settings' section indicates that subnets are assigned by AWS.

25. For **Security Groups**, choose **Create a new security group** and this will open the Security Groups page.

The screenshot shows the 'Security groups' page. At the top, there's a header with 'Security groups' and a link to 'Info'. Below the header, a note says 'A security group is a set of firewall rules that control the traffic to your load balancer.' Under the heading 'Security groups', there's a dropdown menu labeled 'Select security groups' and a button labeled 'C'. A red box highlights the 'Create new security group' button, which is also labeled with a small icon. Below this, a list shows one item: 'default sg-7483ba16 X'. To the right of the list is a small icon with a minus sign.

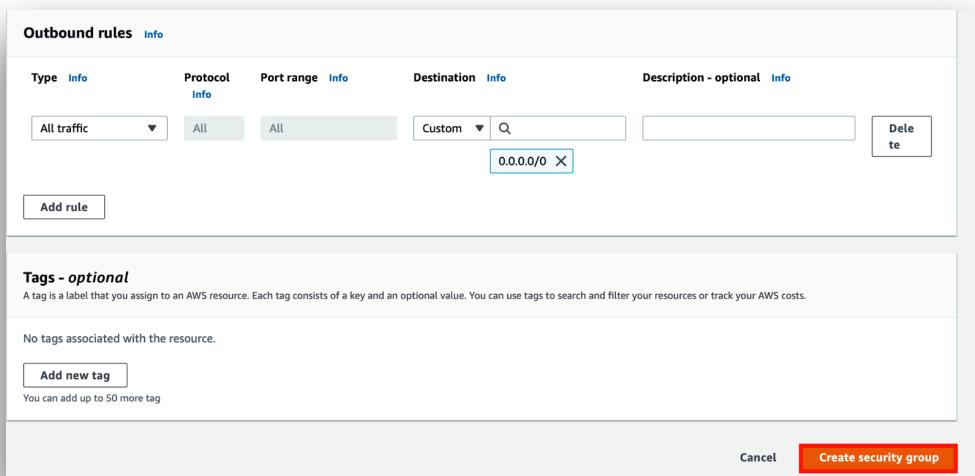
26. Assign a name and description to this security group, and keep the VPC to default.

The screenshot shows the 'Create security group' wizard. At the top, it shows the path 'EC2 > Security Groups > Create security group'. Below that, the title 'Create security group' has a link to 'Info'. A note below the title says 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.' The 'Basic details' section contains three fields: 'Security group name' with 'DemoELBASGSecurityGroup' entered, 'Description' with 'DemoELBASGSecurityGroup' entered, and 'VPC' with 'vpc-a1c94fc8' selected. The 'VPC' field has a search bar and a delete icon.

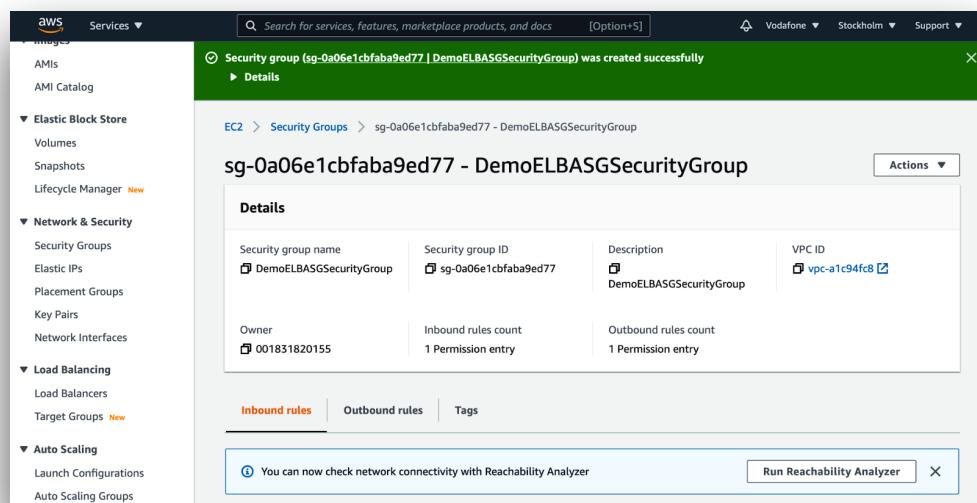
27. For **Inbound rules**, select **HTTP** protocol from anywhere (**0.0.0.0/0**).

The screenshot shows the 'Inbound rules' configuration page. At the top, it shows the title 'Inbound rules' with a link to 'Info'. Below the title, there are columns for 'Type' (with 'HTTP' selected), 'Protocol' (with 'TCP' selected), 'Port range' (with '80' entered), 'Source' (with 'Anywh...' and a search icon), 'Description - optional' (an empty text input), and a 'Delete' button. A single rule is listed at the bottom: '0.0.0.0/0 X'. Below the rule list is a button labeled 'Add rule'.

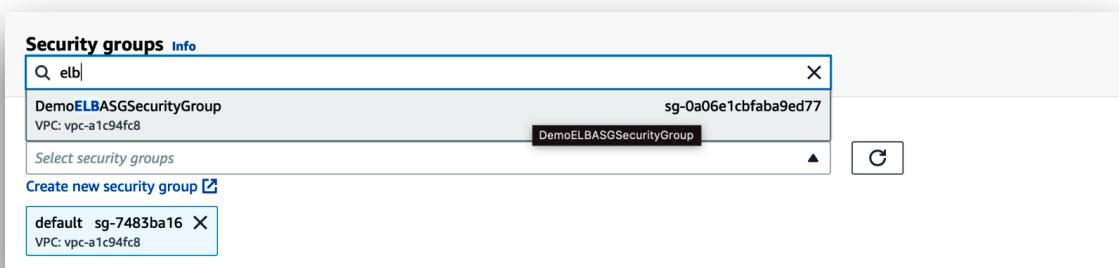
28. Keep the outbound rules to default and choose **Create Security Group**.



29. The security group is now created successfully. The same one will be assigned to the load balancer and EC2 instances.



30. Switch back to the load balancer dashboard, hit refresh, remove the default security group and then select the one created and configured above.



31. For **Listeners and routing**, the default listener accepts HTTP traffic on port 80. For **Default action**, choose the target group that you created.

**Listeners and routing** [Info](#)  
A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener **HTTP:80** [Remove](#)

Protocol	Port	Default action	Info
HTTP	80 1-65535	Forward to	<b>stockalb</b> Target type: Instance, IPv4
		Create target	<input type="text" value="Q"/>
			<b>stockalb</b> Target type: Instance, IPv4

[Add listener](#)

32. Review your configuration, and choose **Create load balancer**.

**Summary**  
Review and confirm your configurations. [Estimate cost](#)

**Basic configuration** [Edit](#)  
**stockalb**

- Internet-facing
- IPv4

**Security groups** [Edit](#)  
• DemoELBASGSecurityGroup [sg-0a06e1cbfaba9ed77](#)

**Network mapping** [Edit](#)  
VPC [vpc-a1c94fc8](#) [my-default-vpc](#)

- eu-north-1a [subnet-3677f55f](#)
- eu-north-1b [subnet-0e1eb975](#)
- eu-north-1c [subnet-f5fe3fb8](#)

**Listeners and routing** [Edit](#)  
• HTTP:80 defaults to [stockalb](#)

**Tags** [Edit](#)  
None

**Attributes**  
ⓘ Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

[Cancel](#) [Create load balancer](#)

33. After the load balancer is created, choose **View Load Balancer**.

ⓘ Successfully created load balancer: **stockalb**  
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers

**Suggested next steps**

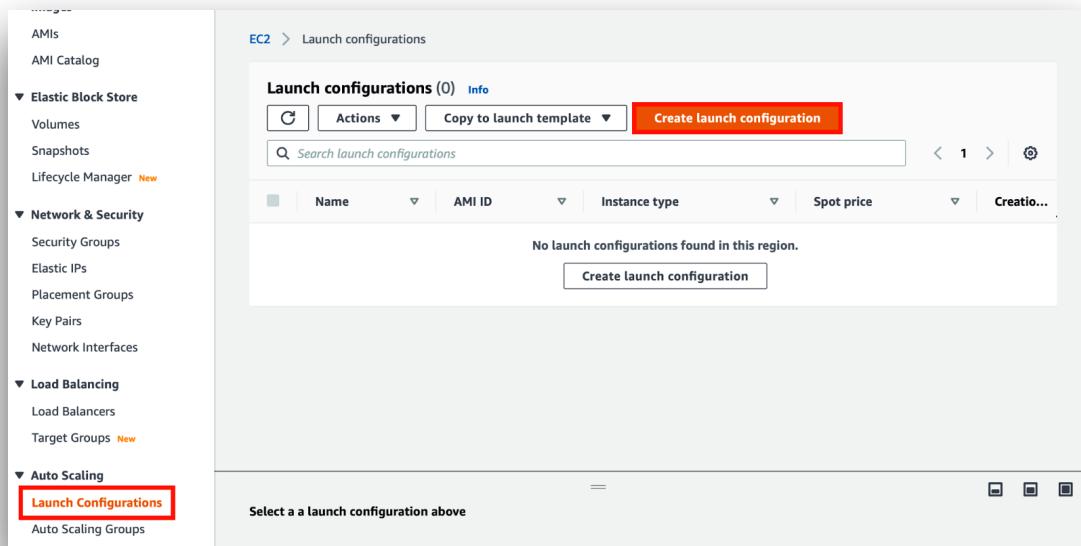
- Review, customize, or enable attributes for your load balancer and listeners using the **Description** and **Listeners** tabs within **stockalb**.
- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within **stockalb**.

[View load balancer](#)

## Create a Launch Configuration

A *launch configuration* is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.

34. On the navigation pane, under **Auto Scaling**, choose **Launch Configurations** and then choose **Create launch configuration**.



35. Enter a name for your launch configuration. For **Amazon machine image (AMI)**, choose the custom AMI you've created in the second step. For **Instance type**, choose either t2.micro or t3.micro (as they're free tier eligible).

The screenshot shows the 'Create launch configuration' wizard. The first step, 'Launch configuration name', has a 'Name' field containing 'stocklaunchconfig'. The second step, 'Amazon machine image (AMI)', has an 'AMI' dropdown set to 'httpdimage'. The third step, 'Instance type', has an 'Instance type' dropdown set to 't3.micro (2 vCPUs, 1 GiB, EBS Only)' with a 'Choose instance type' button next to it.

36. For **Security groups**, select the security group created and configured in steps 25-30.

The screenshot shows the AWS Security Groups interface. At the top, there are two options: 'Create a new security group' (radio button) and 'Select an existing security group' (radio button, which is selected). Below this is a search bar containing 'elb'. A table lists a single security group:

Security group ID	Name	VPC ID	Description
sg-0a06e1cbfaba9ed77	DemoELBASGSecurityGroup	vpc-a1c94fc8	DemoELBASGSecurityGroup

Below the table are two warning boxes:

- A red triangle icon: 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.
- A red triangle icon: You will not be able to connect to this instance as the AMI requires port(s) 22 to be open in order to have access. Your current security group doesn't have port(s) 22 open.

37. For **Key pair (login)**, choose **Proceed without a key pair** as you will not be connecting to the instances.

The screenshot shows the AWS Key pair (login) configuration page. At the top, it says 'Key pair options' and has a dropdown menu set to 'Proceed without a key pair'. Below the dropdown is a checkbox labeled 'I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.' At the bottom right are two buttons: 'Cancel' and 'Create launch configuration' (which is highlighted in orange).

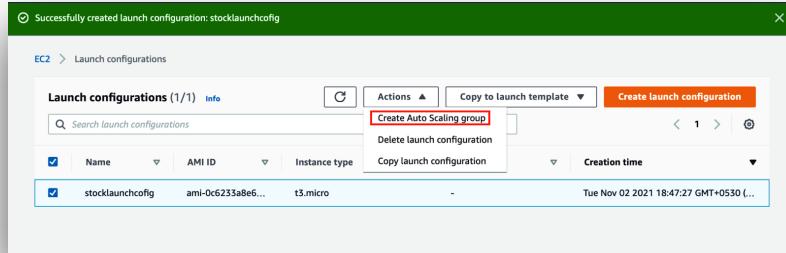
## Create an Auto Scaling Group

An *Auto Scaling group* contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service.

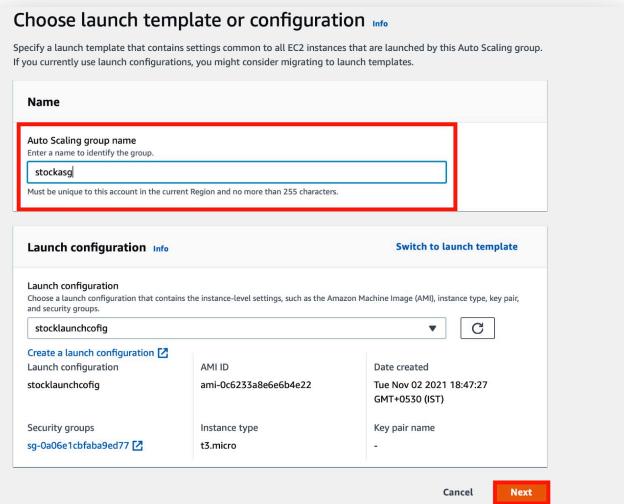
The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

An Auto Scaling group starts by launching enough instances to meet its desired capacity. It maintains this number of instances by performing periodic health checks on the instances in the group. The Auto Scaling group continues to maintain a fixed number of instances even if an instance becomes unhealthy. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.

38. Select the launch configuration created above, and choose **Create Auto Scaling group**.

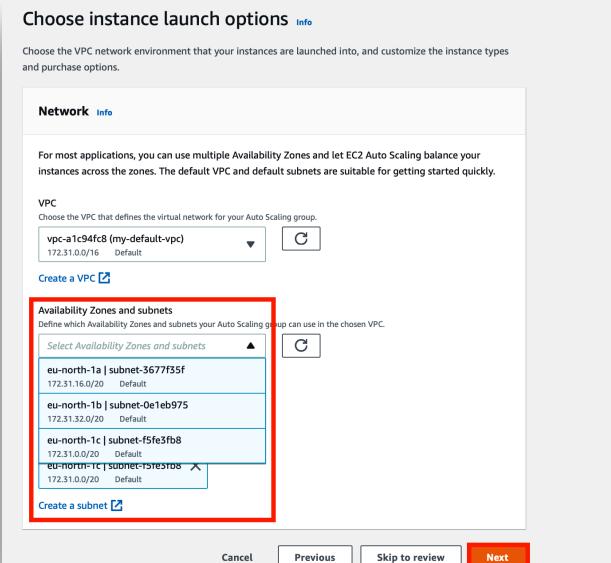


39. On the **Choose launch template or configuration** page, for **Auto Scaling group name**, enter a name for your Auto Scaling group. Verify that your launch configuration supports all of the options that you are planning to use, and then choose **Next**.



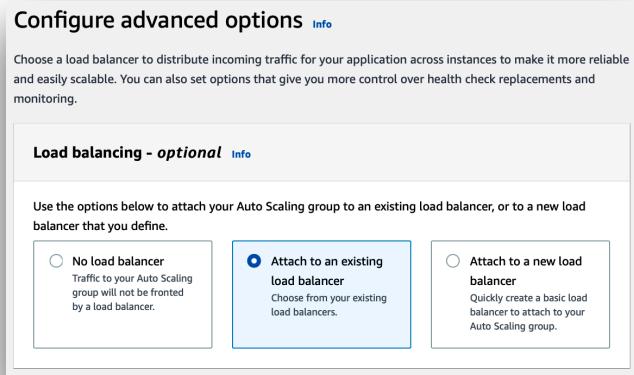
40. On the **Configure instance launch options** page, under **Network**, for **VPC**, choose the default VPC.

For **Availability Zones and subnets**, choose all the subnets in the specified VPC, and choose **Next**.



41. On the **Configure advanced options** page, configure the following options:

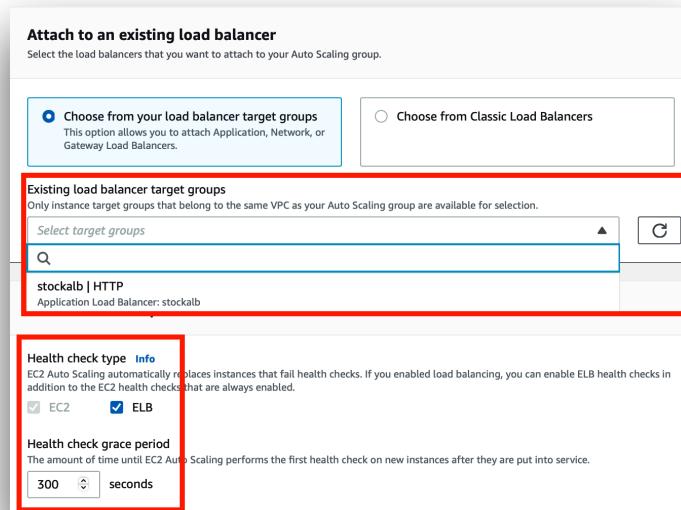
A. For **Load balancing**, choose **Attach to an existing load balancer**.



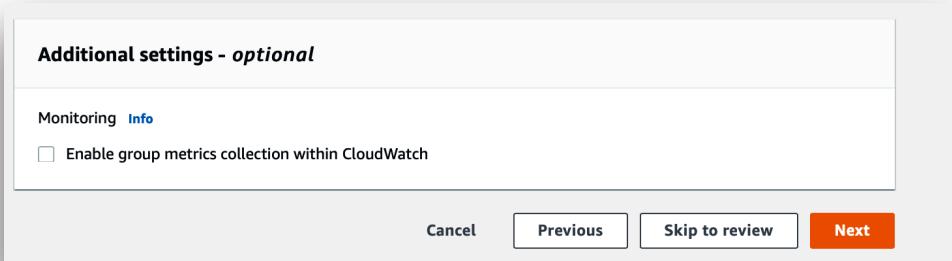
B. Under **Attach to an existing load balancer**, choose **Choose from your load balancer target groups**, and then choose a target group in the **Existing load balancer target groups** field.

C. To enable your Elastic Load Balancing (ELB) health checks, for **Health checks**, choose **ELB** under **Health check type**. These health checks are optional when you enable load balancing.

D. Under **Health check grace period**, enter the amount of time until Amazon EC2 Auto Scaling checks the Elastic Load Balancing health status of an instance after it enters the **InService** state.



E. Under **Additional settings, Monitoring**, choose whether to enable CloudWatch group metrics collection. Skip this option and choose **Next**.



## Create an Target Tracking Policy

You can choose to configure a target tracking scaling policy on an Auto Scaling group as you create it or after the Auto Scaling group is created.

### To create an Auto Scaling group with a target tracking scaling policy

42. Under **Group size**, specify the range that you want to scale between by updating the minimum capacity and maximum capacity. These two settings allow your Auto Scaling group to scale dynamically. Amazon EC2 Auto Scaling scales your group in the range of values specified by the minimum capacity and maximum capacity.

For this hands-on, set the Desired and Minimum capacity as 2, and the Maximum capacity as 4.

Configure group size and scaling policies Info

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.

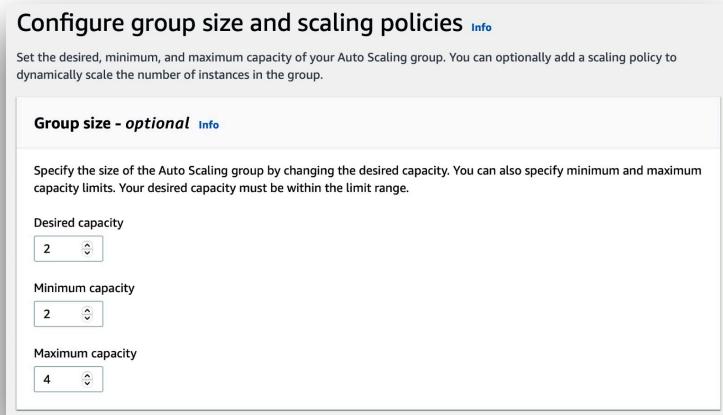
**Group size - optional** Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity  
2

Minimum capacity  
2

Maximum capacity  
4



43. Under Scaling policies, choose **Target tracking scaling policy**. To define a policy, do the following:

- Specify a name for the policy.
- For **Metric type**, choose **Application Load Balancer request count per target**

**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

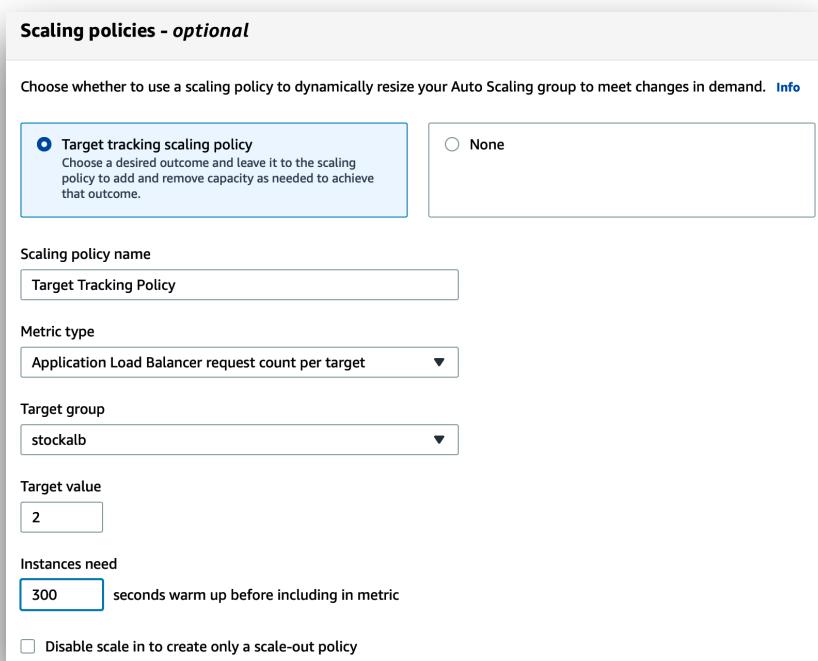
Metric type  
Application Load Balancer request count per target ▾

Target group  
stockalb ▾

Target value  
2

Instances need  
300 seconds warm up before including in metric

Disable scale in to create only a scale-out policy



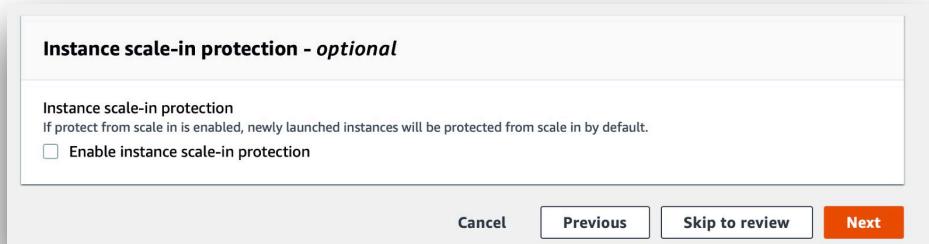
- C. For **Target group**, select the one created in steps 16-20.
- D. Specify a **Target value** for the metric. Enter a lower value such as **2** so that we can get faster results.
- E. Specify an instance warm-up value for **Instances need**. This allows you to control the time until a newly launched instance can contribute to the CloudWatch metrics. Enter **300** as the value here.
- F. Keep **Disable scale in to create only a scale-out policy** option as unchecked.

44. Choose **Next** to proceed.

**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

Cancel Previous Skip to review Next

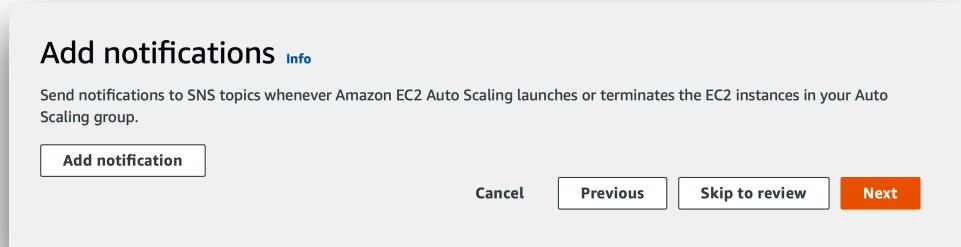


45. (Optional) To receive notifications, for **Add notification**, configure the notification, and then choose **Next**.

**Add notifications** Info

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

Add notification Cancel Previous Skip to review Next



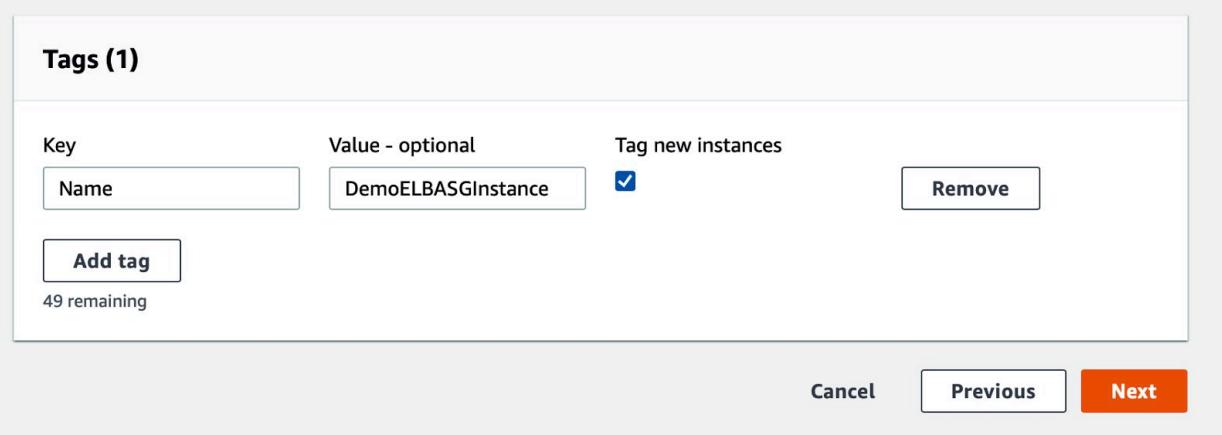
46. (Optional) To add tags, choose **Add tag**, provide a tag key and value for each tag, and then choose **Next**.

**Tags (1)**

Key	Value - optional	Tag new instances
Name	DemoELBASGInstance	<input checked="" type="checkbox"/>

Add tag Remove  
49 remaining

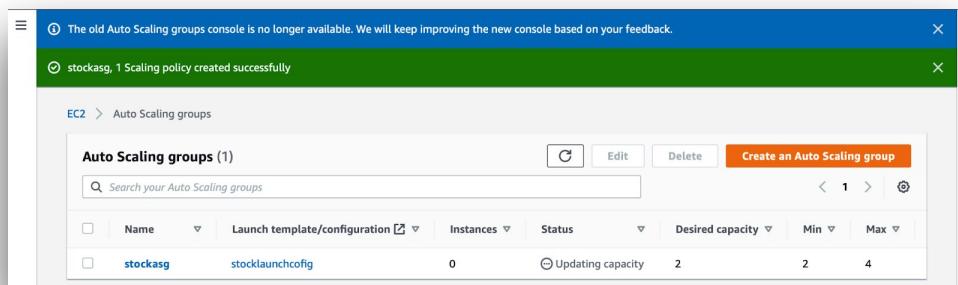
Cancel Previous Next



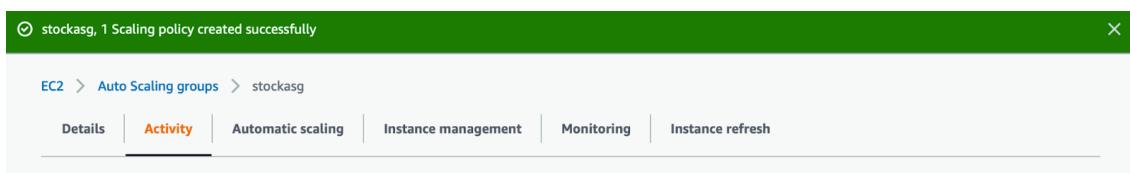
47. On the **Review** page, choose **Create Auto Scaling group**.

The screenshot shows the 'Review' step of the Auto Scaling group creation wizard. It consists of two stacked sections: 'Step 5: Add notifications' and 'Step 6: Add tags'. Both sections have an 'Edit' button in the top right corner. The 'Notifications' section has a sub-section titled 'Tags (1)' which contains one tag: 'Name' with value 'DemoELBASGInstance'. Below the tags is a 'Tag new instances' checkbox. At the bottom of each section is a 'Cancel' button and a red 'Create Auto Scaling group' button.

48. This marks the completion of the process of creating and configuring an Auto Scaling group.



49. Click on this newly created Auto Scaling group and go to **Activity** menu option.



50. Within **Activity history**, you can see the list of instances launched in this Auto Scaling group.

The screenshot shows the 'Activity history' table for the 'stockasg' Auto Scaling group. The table has columns: Status, Description, Cause, and Start time. There are two entries:

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-04bb224ce79e495a9	At 2021-11-02T14:02:45Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2021-11-02T14:02:48Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2021 November 02, 07:32:57 PM +05:30
Successful	Launching a new EC2 instance: i-07dd71e66f46b3aa6	At 2021-11-02T14:02:45Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2021-11-02T14:02:48Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2021 November 02, 07:32:57 PM +05:30

## Test your Deployment

51. In the navigation pane, under **Load Balancing**, choose **Load Balancers**, and select the newly created load balancer.

Choose **Description** and copy the DNS name of the load balancer (for example, my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com).

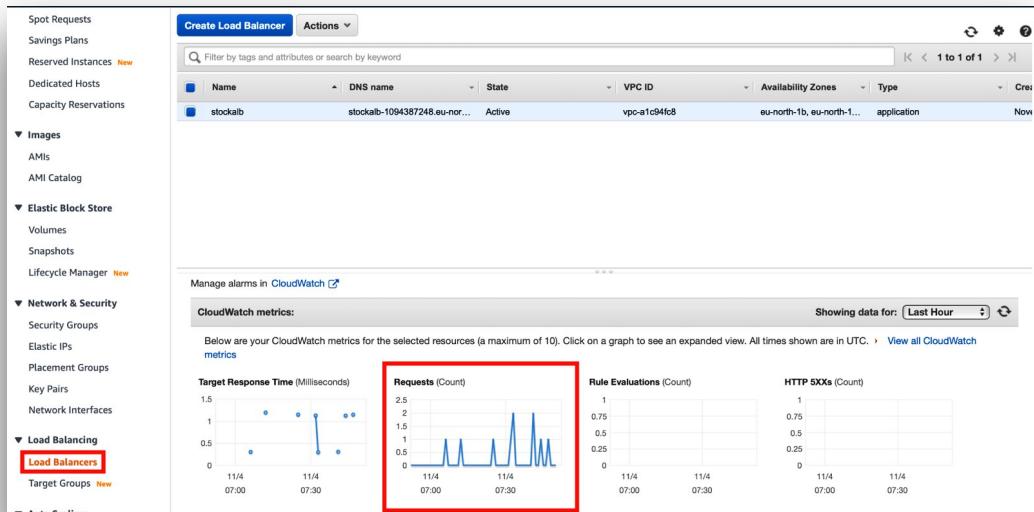
The screenshot shows the AWS CloudFormation console with the navigation pane on the left. Under the 'Load Balancing' section, the 'Load Balancers' tab is selected. The main area displays a table with one row for 'stockalb'. The 'DNS name' column shows 'stockalb-130186338.eu-north-1.elb.amazonaws.com (A Record)'. A red box highlights this specific field.

52. Paste the DNS name into the address field of an internet-connected web browser. If everything is working, the browser displays the default page of your server.



- Refresh the browser several times so that the Application Load Balancer request per count per target exceeds the pre-defined limit.

53. You can also check the request count by clicking on **Load Balancer** in the navigation menu, select the load balancer you've deployed, go to **Monitoring** and check **Requests (count)**.



54. In the navigation pane, select **Auto Scaling Groups** and click on the Auto Scaling group.

The screenshot shows the AWS EC2 Auto Scaling Groups console. On the left, the navigation pane includes sections for AMIs, ELASTIC BLOCK STORE (Volumes, Snapshots, Lifecycle Manager), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups), LOAD BALANCING (Load Balancers, Target Groups), and AUTO SCALING (Launch Configurations, Auto Scaling Groups). The 'Auto Scaling Groups' link is highlighted with a red box. The main content area displays a table titled 'Auto Scaling groups (1)'. The table has columns for Name, Launch template/configuration, Instances, Status, and Desired capacity. A single row is shown for 'stockasg', which has a status of '4' instances and a desired capacity of '4'. A search bar at the top of the table is also highlighted with a red box.

55. Choose **Activity**.

The screenshot shows the 'Auto Scaling groups' details page for 'stockasg'. The top navigation bar shows 'EC2 > Auto Scaling groups > stockasg'. Below the navigation bar, there are tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The 'Activity' tab is highlighted with a red box.

56. Scroll down and view the activity logs.

Activity history (4)		
<input type="text"/> Filter activity history		
Status	Description	Cause
WaitingForLaunch	Launching a new EC2 instance: i-06a6bf6a96004755e	At 2021-11-02T16:00:00Z a monitor alarm TargetTracking-stockasg-AlarmHigh-e875d78a-af8e-0624bd46c982 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 4. At 2021-11-02T16:00:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 4.
WaitingForLaunch	Launching a new EC2 instance: i-01718ef53e1359c12	At 2021-11-02T16:00:00Z a monitor alarm TargetTracking-stockasg-AlarmHigh-e875d78a-af8e-0624bd46c982 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 4. At 2021-11-02T16:00:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 4.
Successful	Launching a new EC2 instance: i-04bb224ce79e495a9	At 2021-11-02T14:02:45Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2021-11-02T14:02:48Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.
Successful	Launching a new EC2 instance: i-07dd71e66f46b3aa6	At 2021-11-02T14:02:45Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2021-11-02T14:02:48Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.

- As you can see, two new EC2 instances have been deployed in response to exceeding requests sent to previously launched EC2 instances running behind the load balancer.

## Clean Up

To avoid ongoing charges for resources you created to complete this tutorial, you should delete or terminate:

- Auto Scaling Group
- Auto Scaling Launch Configuration
- Load Balancer
- Load Balancer Target Group
- Amazon Machine Image (AMI)
- Snapshot