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:

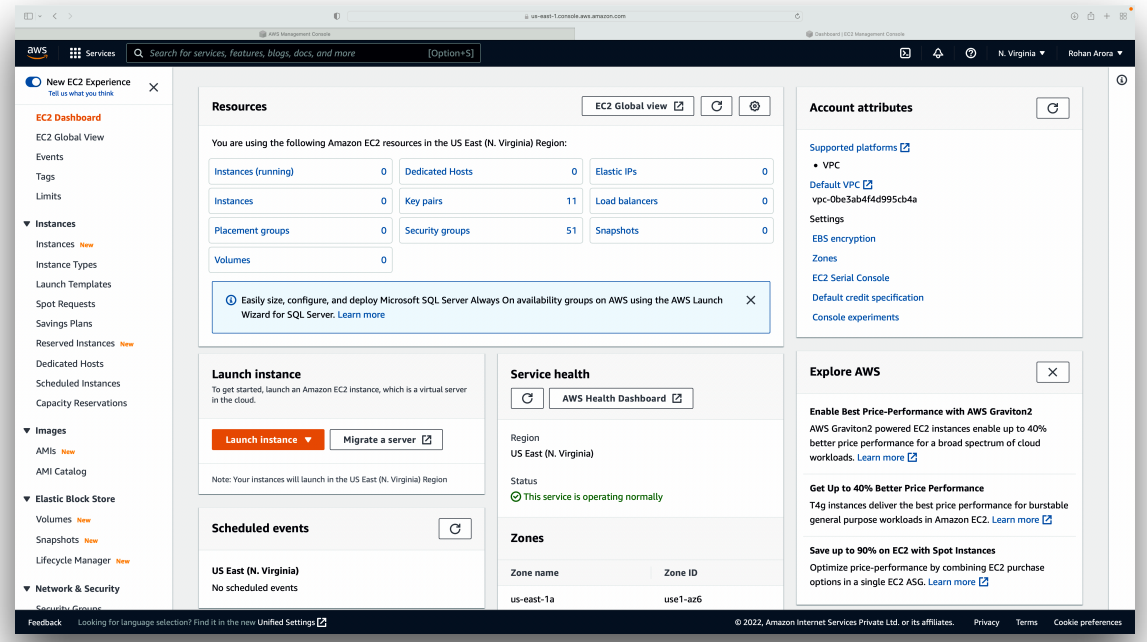
1. Deploy a demo web application on an Amazon EC2 instance
2. Create a custom AMI
3. Create a Target Group
4. Create an Application Load Balancer
5. Create a launch template
6. Create an Auto Scaling Group
7. Send multiple requests to the load balancer's DNS name

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

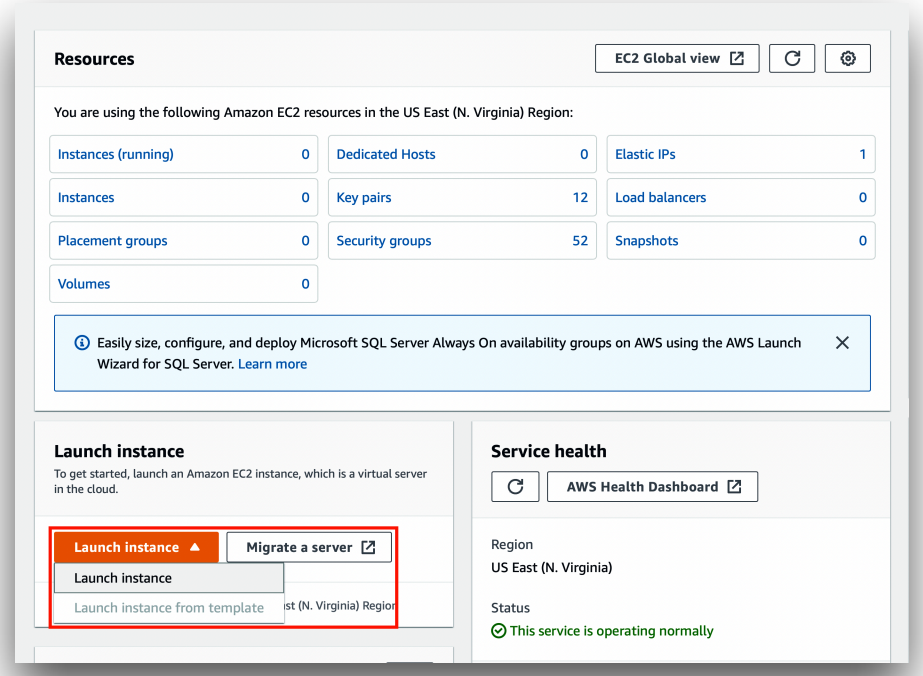
Deploy a Demo Web Application on an Amazon EC2 Instance

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

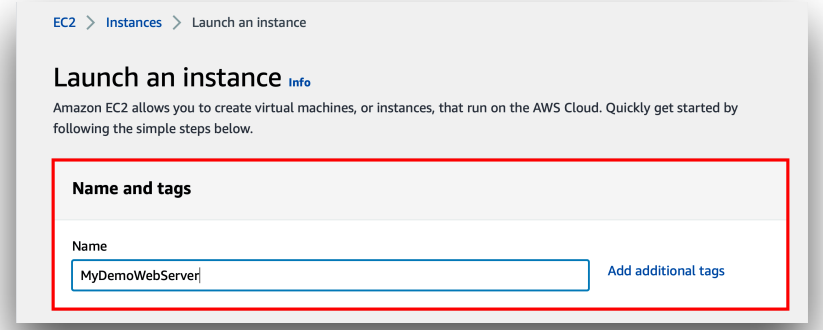
1. Launch an EC2 instance using a bootstrap script. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.



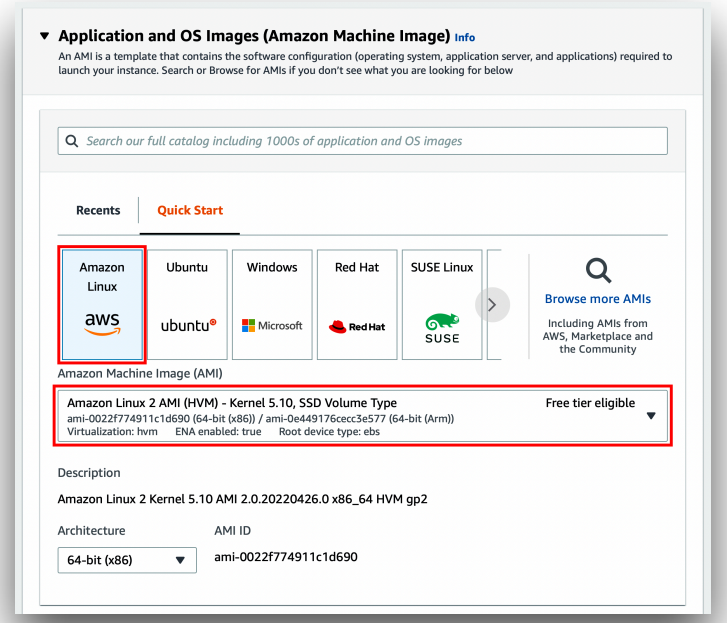
2. From the Amazon EC2 console dashboard, choose **Launch instance**.



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.

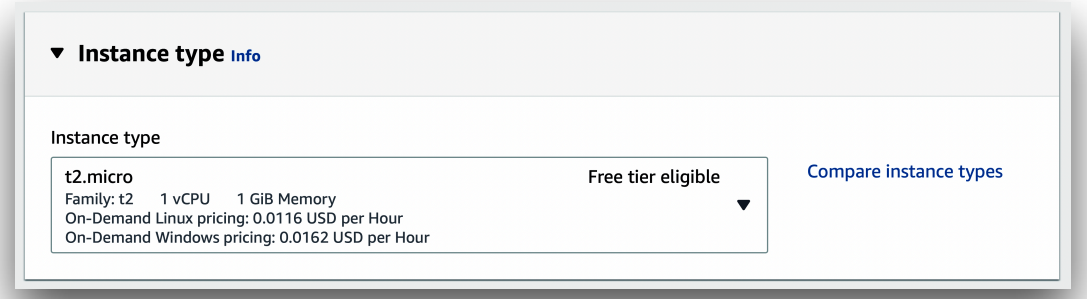


4. Under **Application and OS Images (Amazon Machine Image)**, choose **Quick Start**, and then choose the **Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type** for your instance.

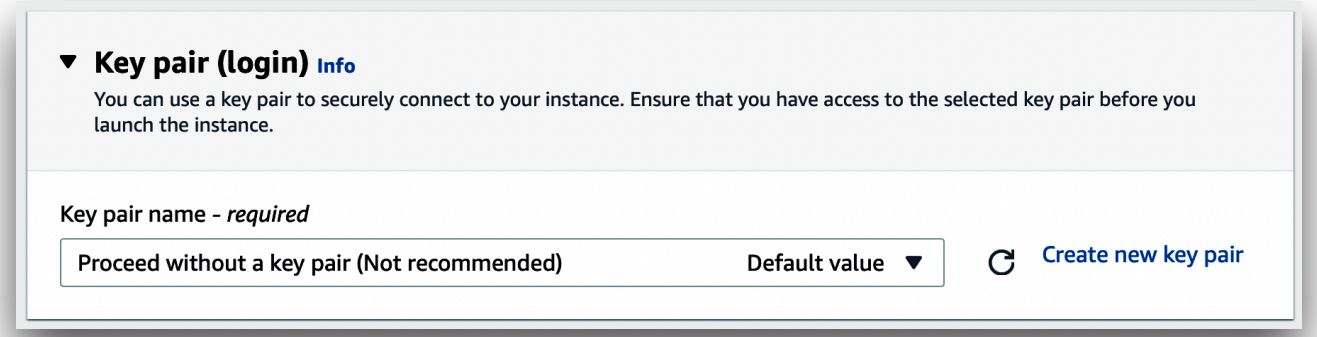


5. For **Instance type**, select the instance type for the instance.

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).



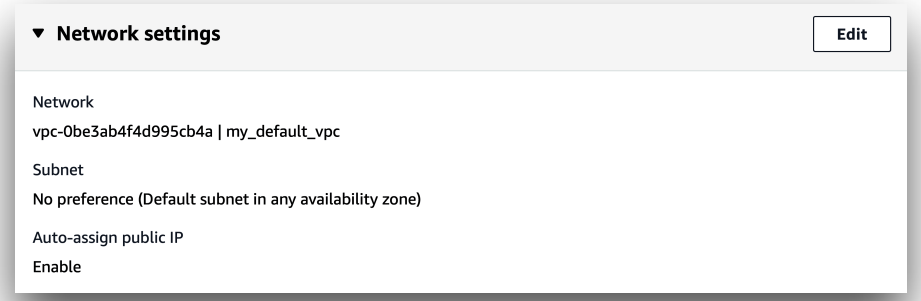
6. For **Key pair name**, choose **Proceed without a key pair (Not recommended)**.



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

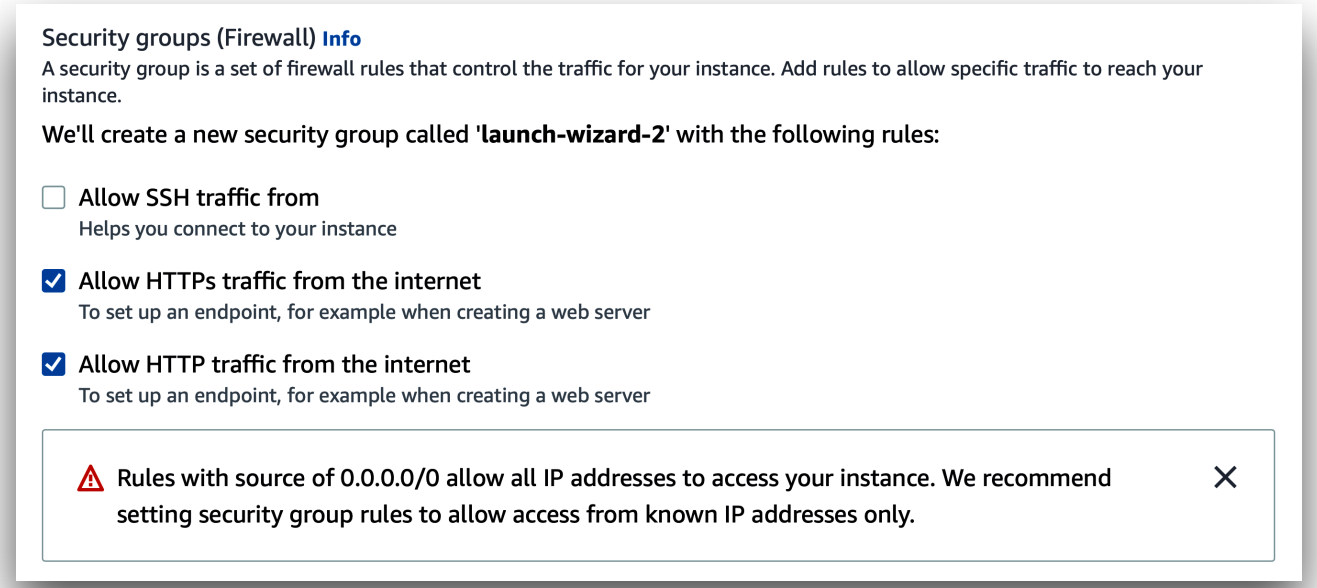
7. Configure the **networking settings** as following:

- **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**.

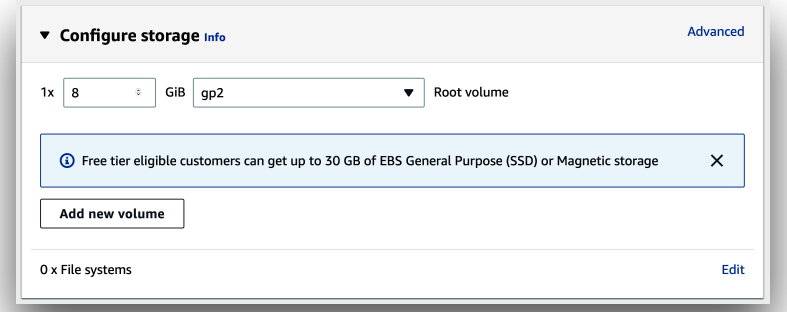


8. The launch instance wizard automatically defines the **launch-wizard-*x*** security group and creates an inbound rule to allow you to connect to your instance over SSH (port 22). Include the inbound rules as following:

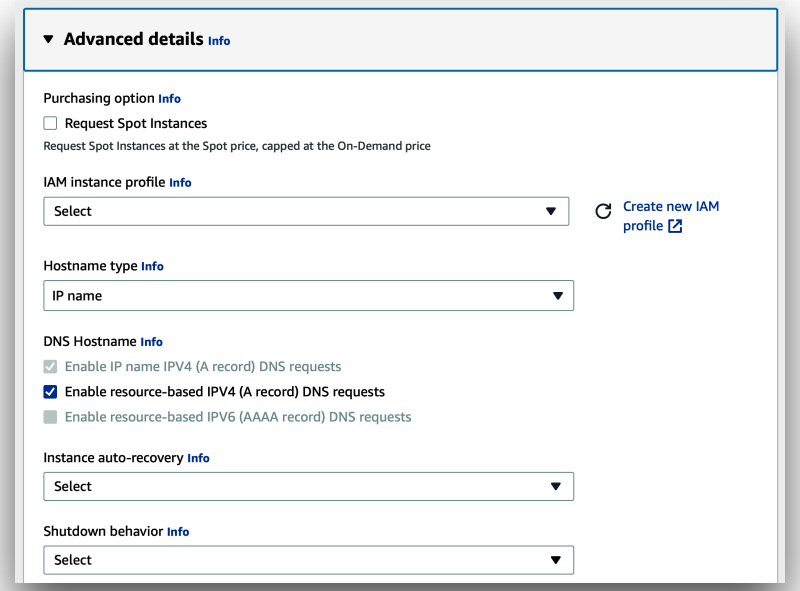
- Since this instance is launched just for extracting an image, therefore there will be no need for you to connect to this instance via SSH. Uncheck **Allow SSH traffic from** option.
- Select **Allow HTTPs traffic from the internet** and **Allow HTTP traffic from the internet** rules to allow internet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



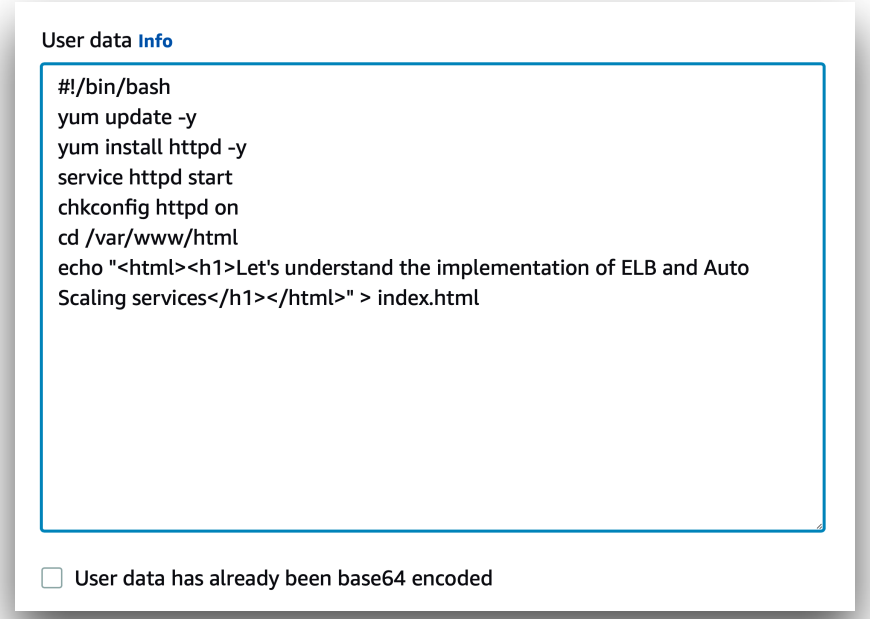
9. For **Configure storage**, accept default values. The AMI you selected includes one or more volumes of storage, including the root volume. By default, an 8 GiB of General Purpose SSD volume is attached to an Amazon Linux instance.



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

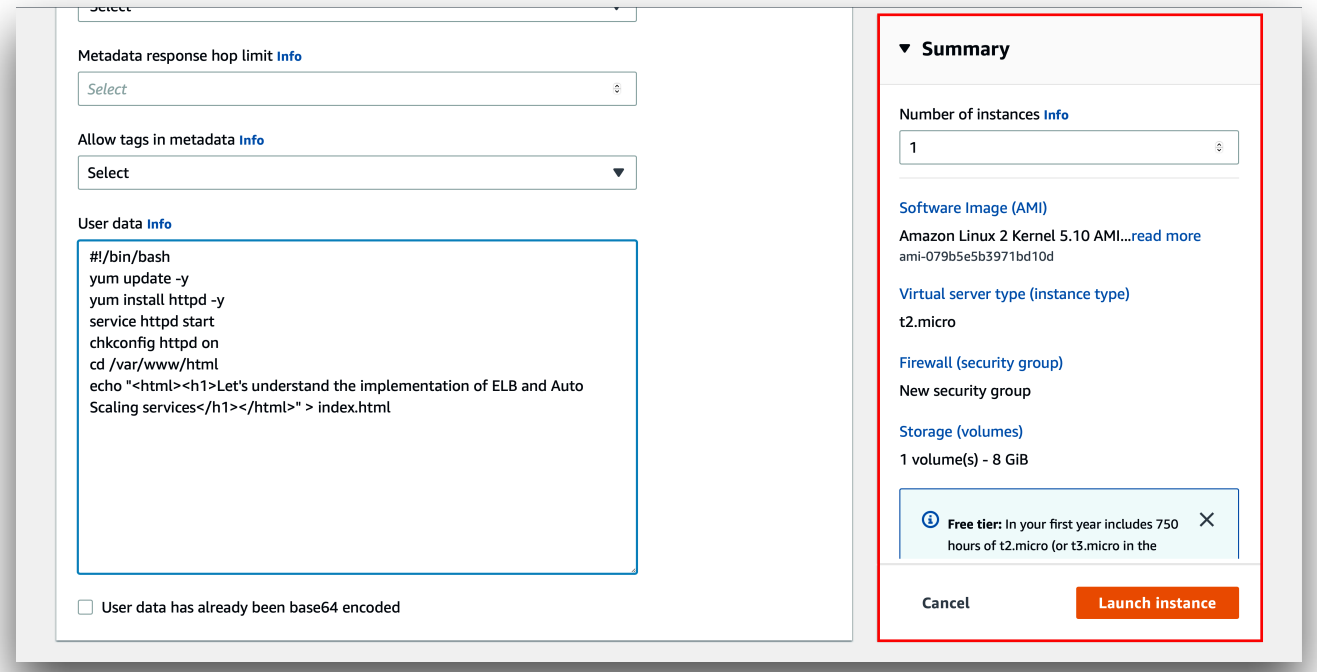


- Accept all default parameters here except for **User data**. You can specify user data to configure an instance during launch, or to run a configuration script.
- Click here to download the bash script which will be used launch an Apache PHP web application on this EC2 instance.
- Copy and paste the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

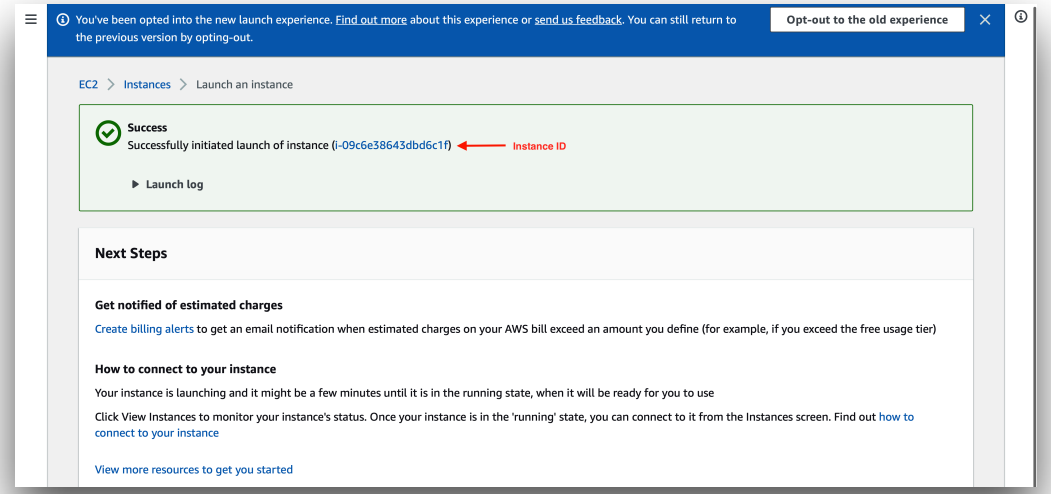


11. Use the **Summary** panel to specify the number of instances to launch, to review your instance configuration, and to launch your instances.

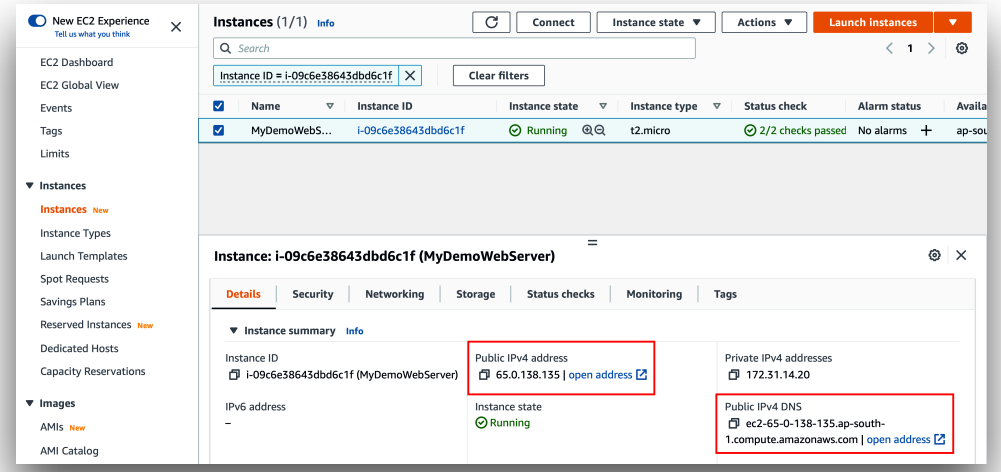
- Keep all settings to default and choose **Launch instance**.



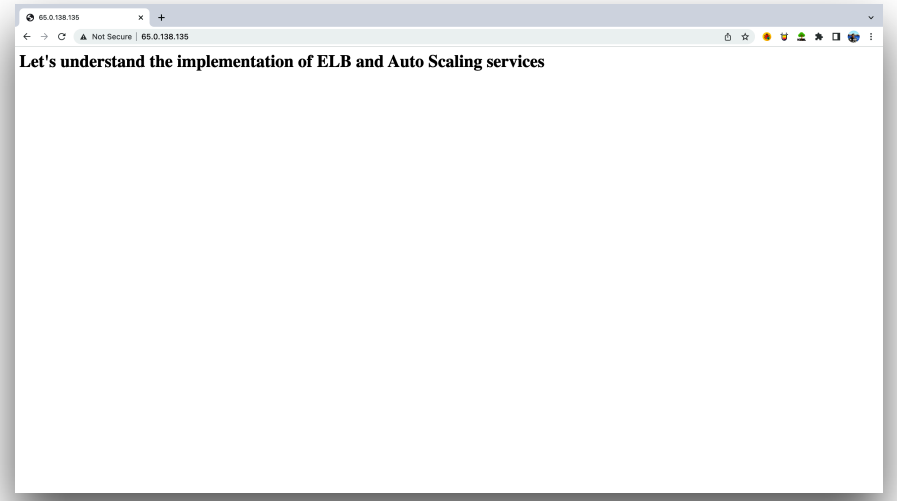
12. Click the Instance ID of the launched instance.



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.

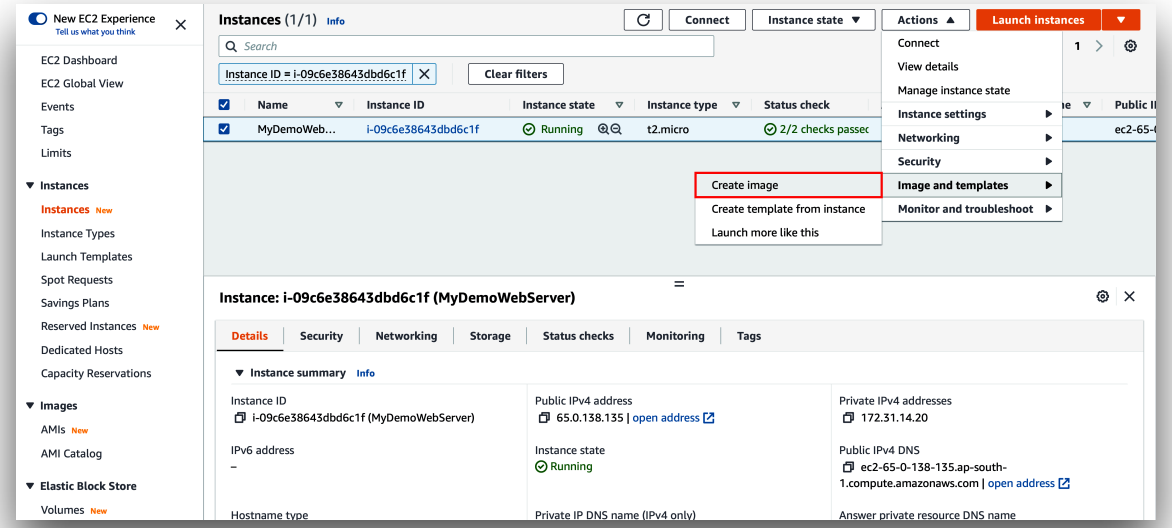


Create a Custom AMI

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

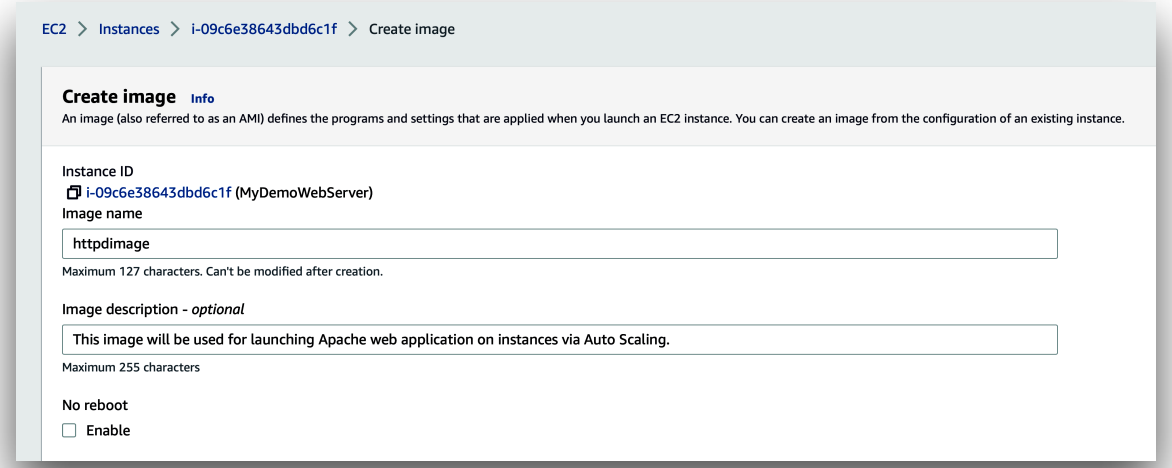
A. In the navigation pane, choose **Instances**, select your instance, and then choose **Actions**, **Image and templates**, **Create image**.



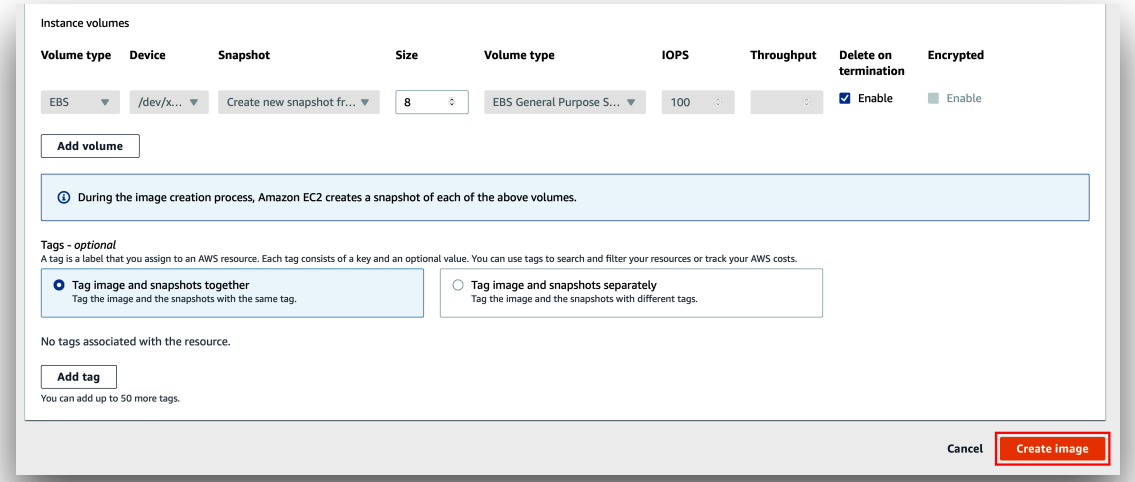
B. On the **Create image** page, specify the following information, and then choose **Create image**.

- **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.

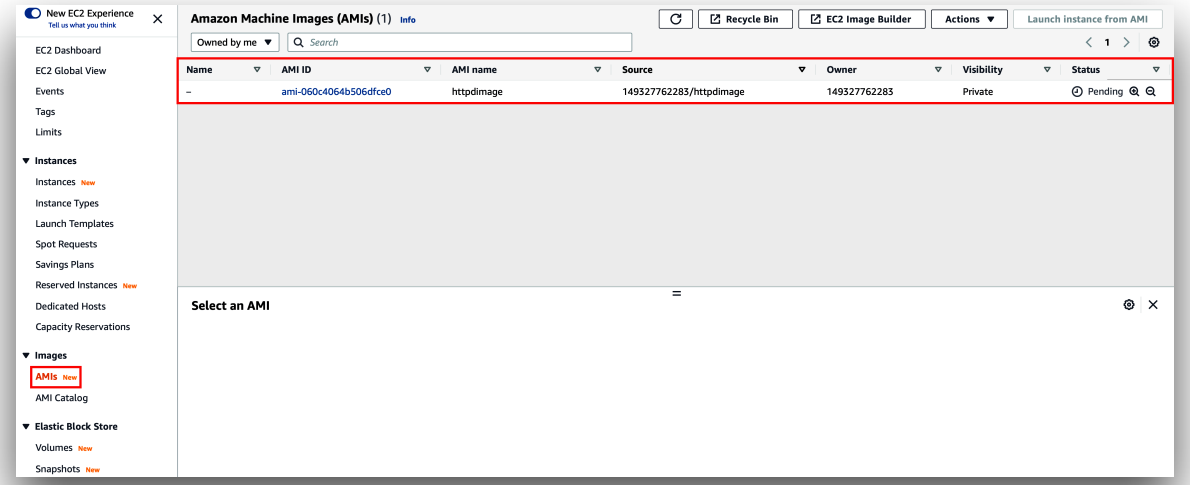
For this hands-on exercise, make sure No reboot option is unchecked.



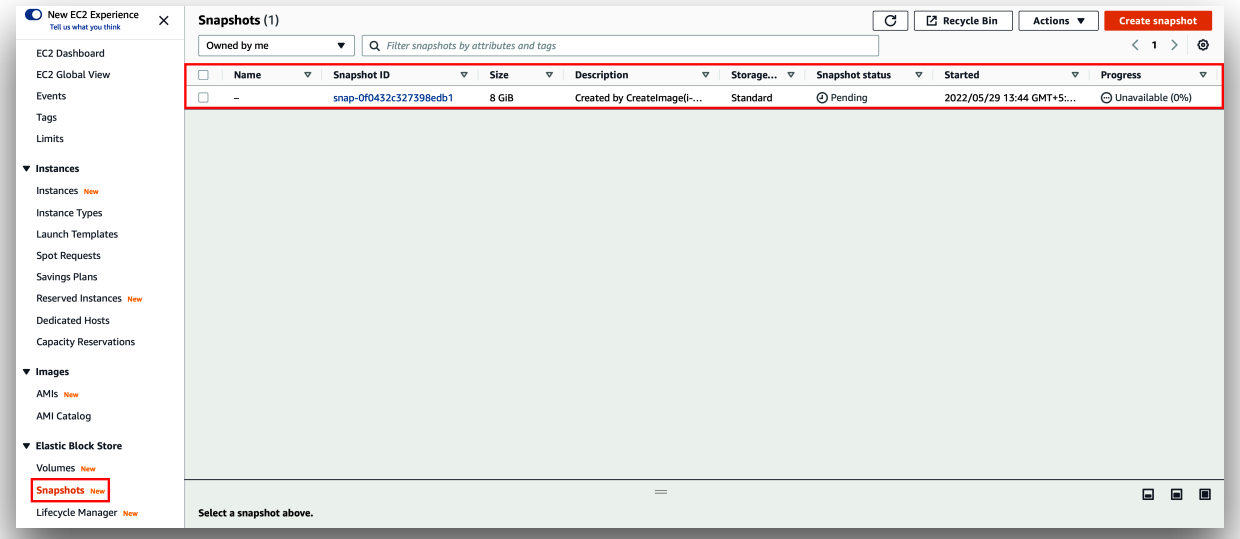
C. Accept default **Instance volumes** settings and skip adding **Tags**. Choose **Create image**.



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.



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.

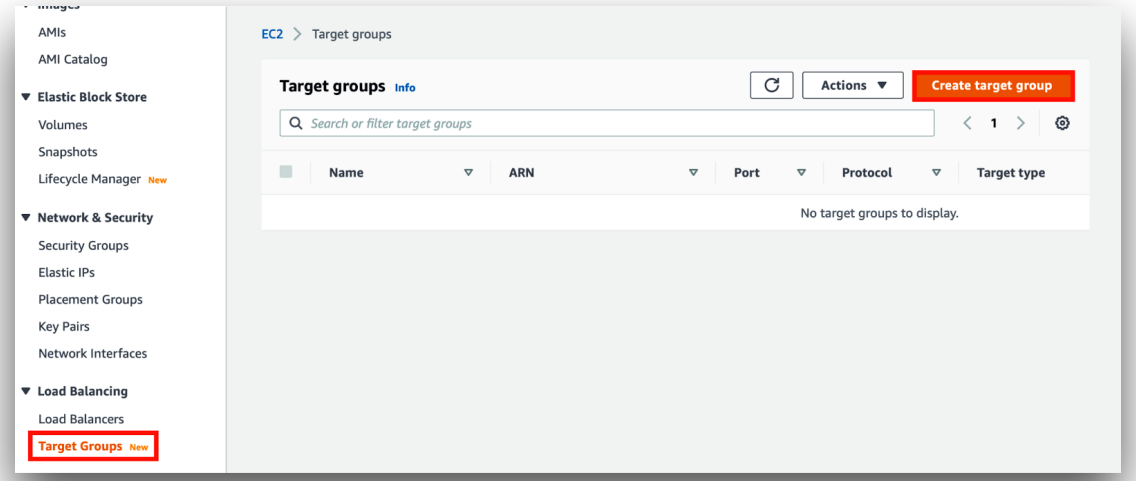


Create a Target Group and an Application Load Balancer

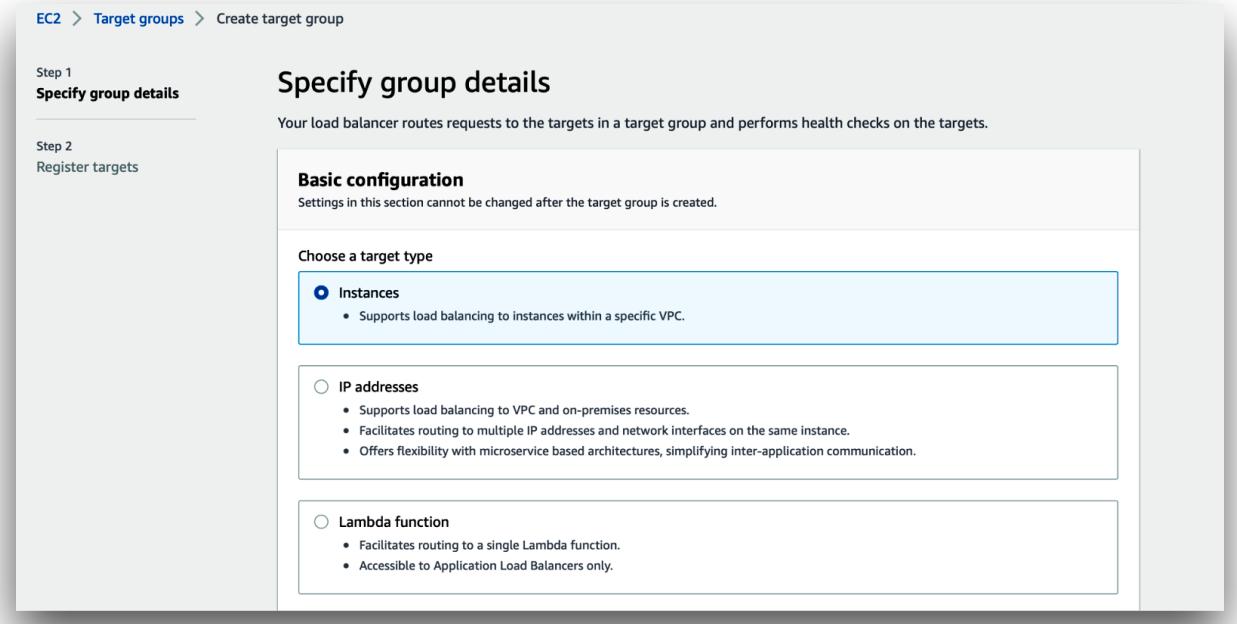
Each *target group* is used to route requests to one or more registered targets. When you create each listener rule, you specify a target group and conditions. When a rule condition is met, traffic is forwarded to the corresponding target group. You can create different target groups for different types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

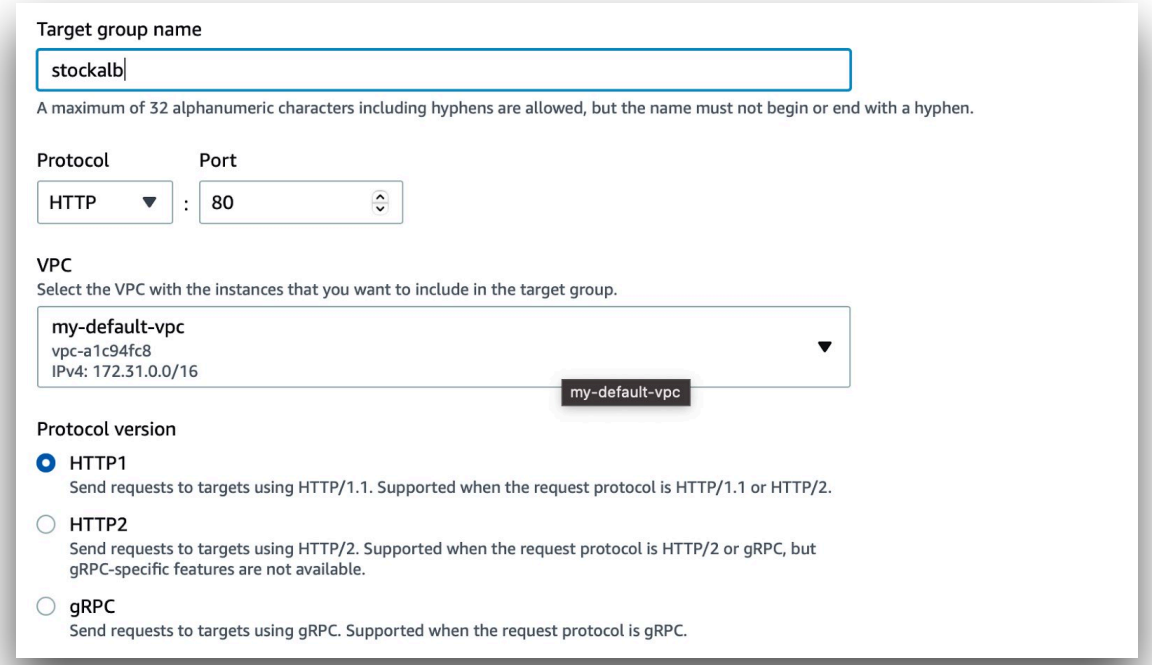


18. For **Target group name**, type a name for the target group. This name must be unique per region per account, can have a maximum of 32 characters, must contain only alphanumeric characters or hyphens, and must not begin or end with a hyphen.

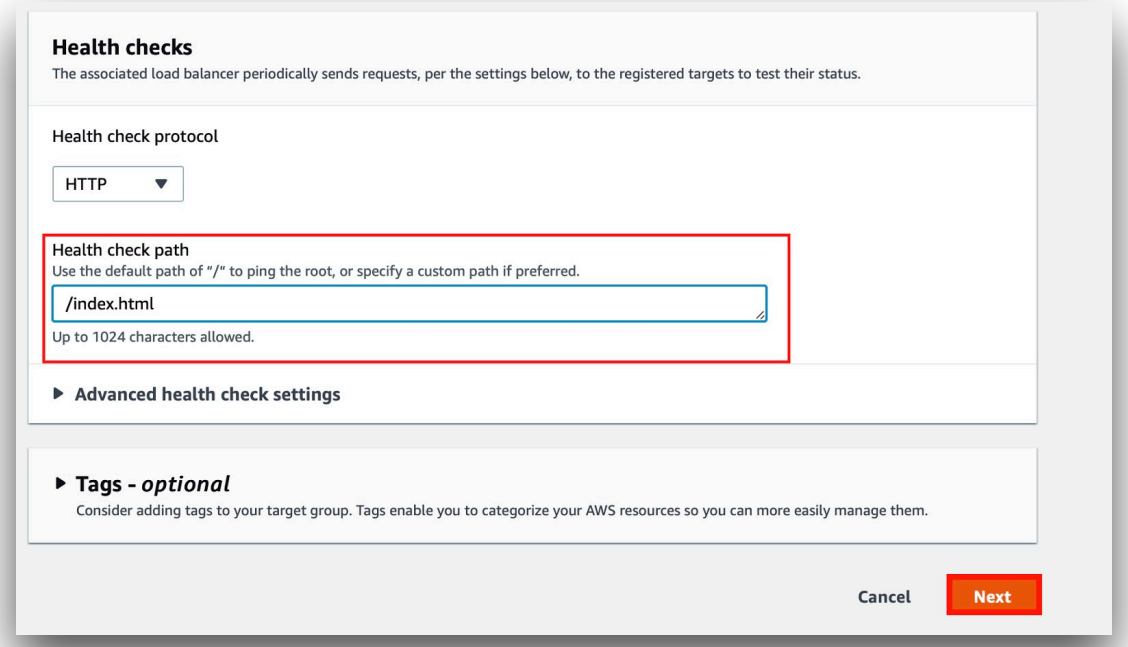
- For **Protocol** and **Port**, accept the default values (Protocol as **HTTP** and Port as **80**).

- For **VPC**, select the default virtual private cloud (VPC).

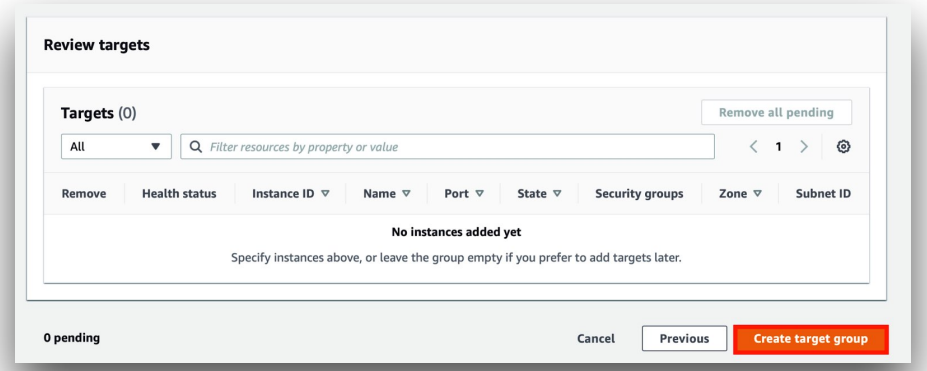
- For **Protocol version**, accept the default value as **HTTP1**.



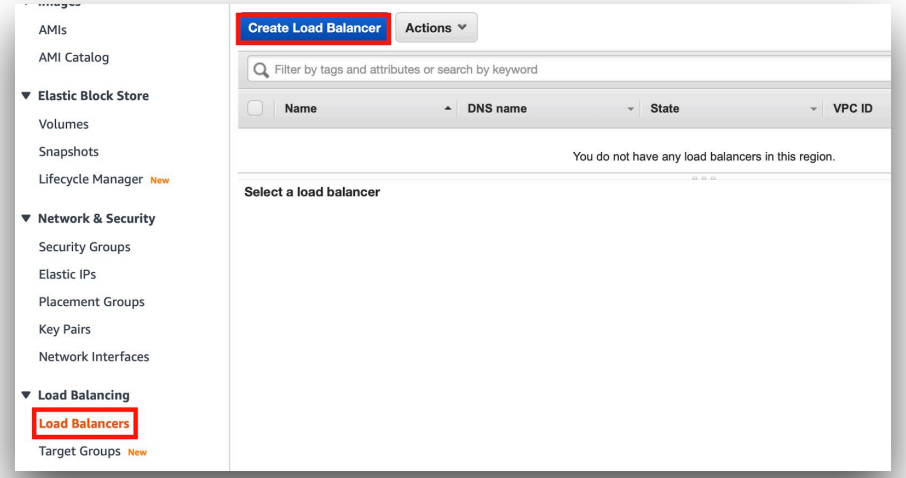
19. Type **index.html** in 'Health check path' and click **Next**.



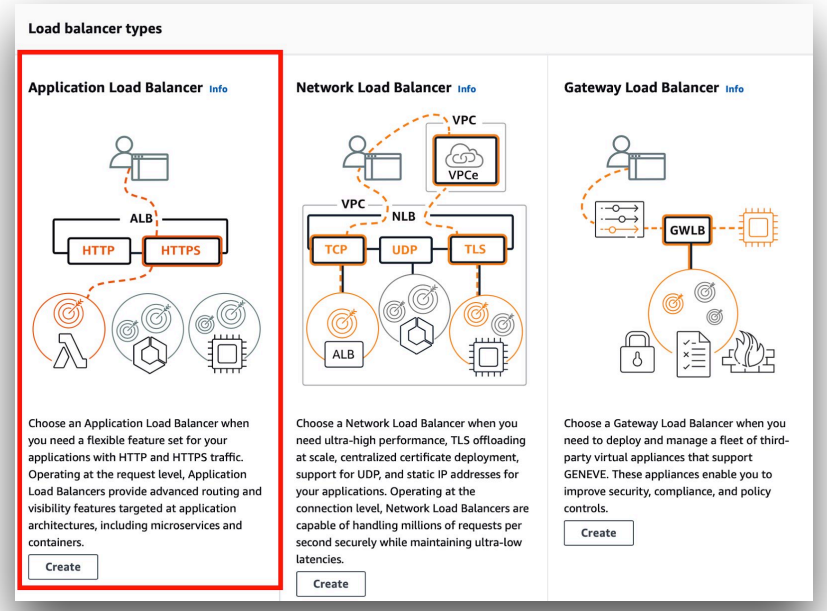
20. Choose **Create target group**.



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



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

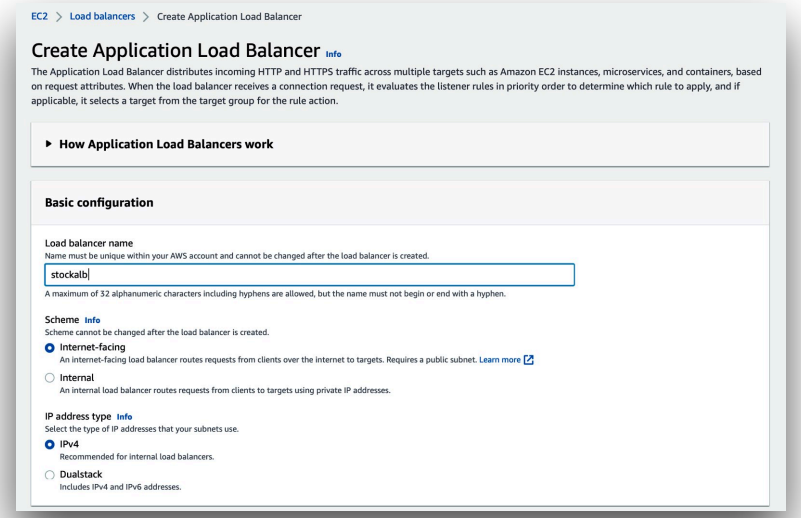


23. **Basic configuration**

A. 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-`.

B. 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.

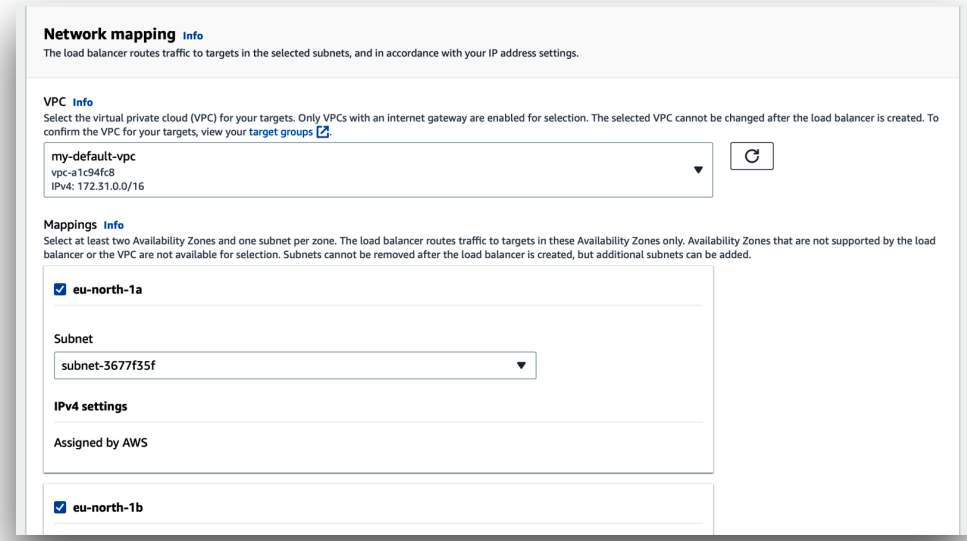
C. For **IP address type**, choose **IPv4** or **Dualstack**. Use **IPv4** if your clients use IPv4 addresses to communicate with the load balancer. Choose **Dualstack** if your clients use both IPv4 and IPv6 addresses to communicate with the load balancer.



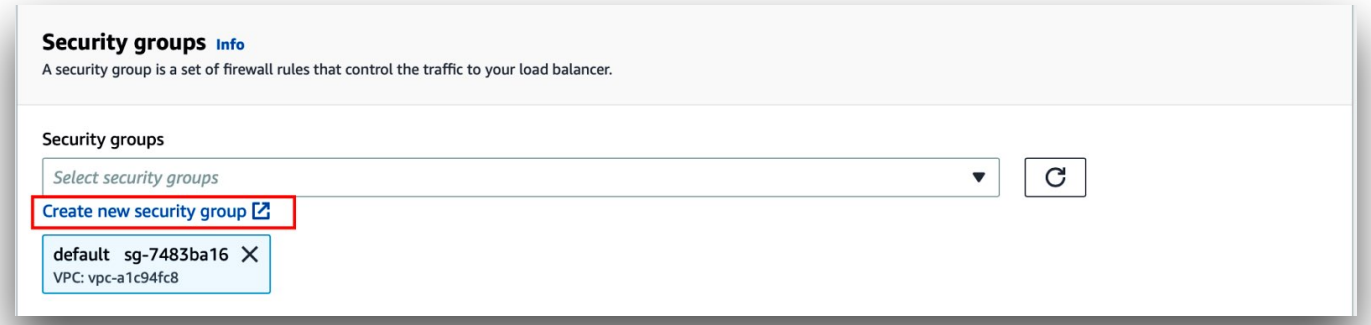
24. **Network mapping**

A. For **VPC**, select the default VPC.

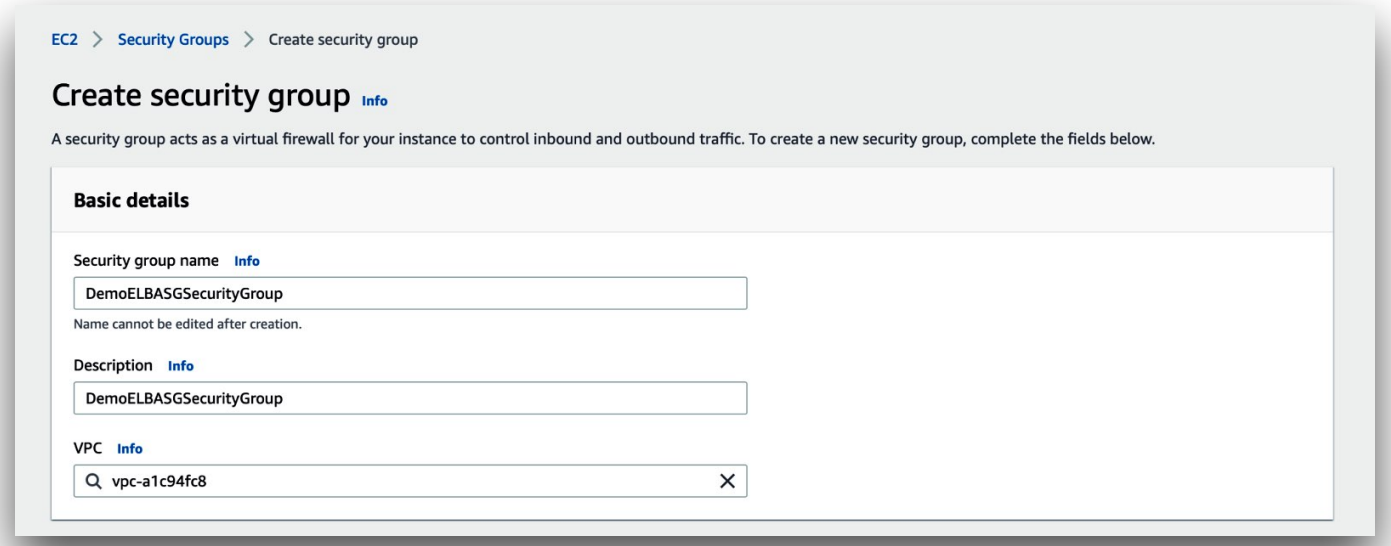
B. For **Mappings**, select all the Availability Zones and corresponding subnets. Enabling multiple Availability Zones increases the fault tolerance of your applications.



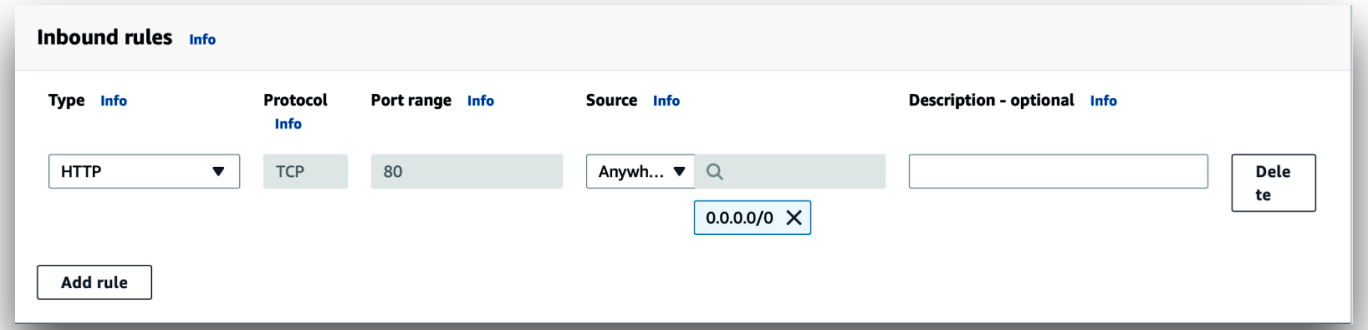
25. For **Security Groups**, choose **Create a new security group** and this will open the Security Groups page.



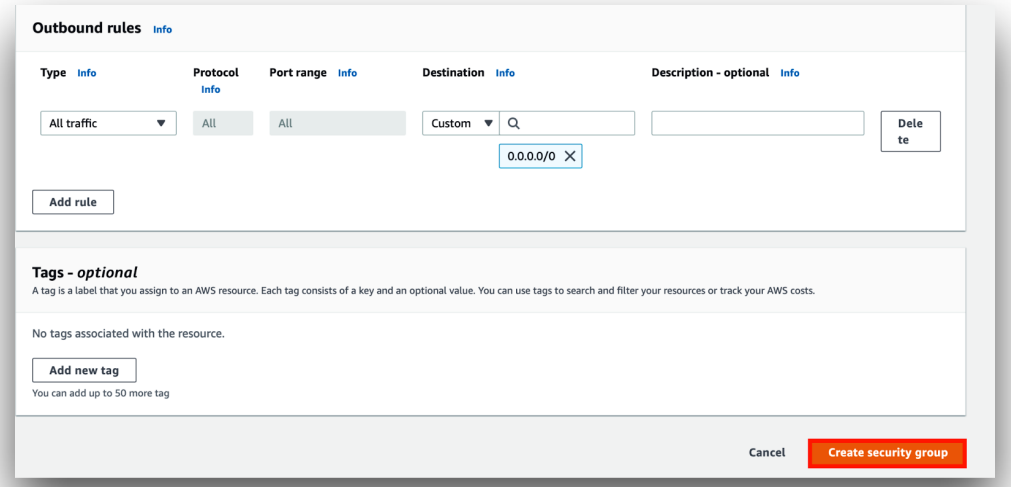
26. Assign a name and description to this security group, and keep the VPC to default.



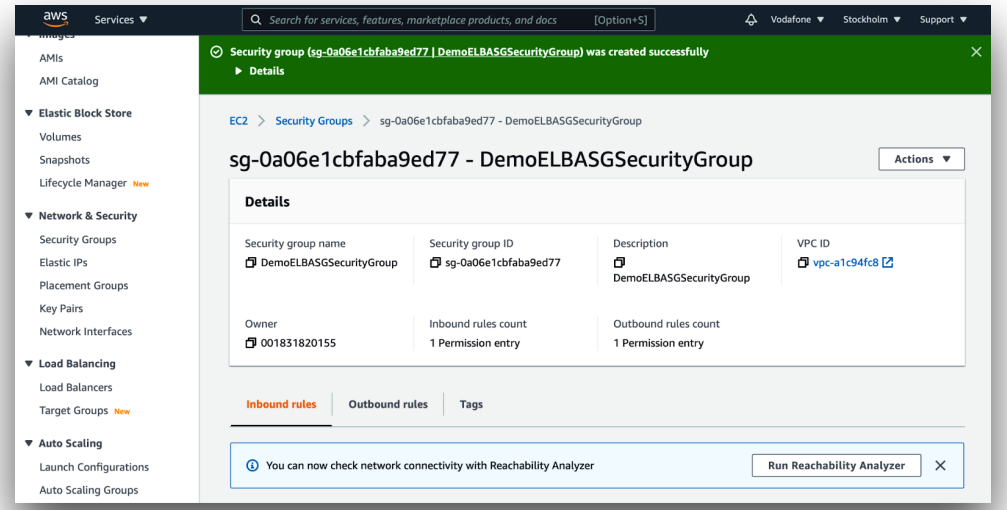
27. For **Inbound rules**, select **HTTP** protocol from anywhere **(0.0.0.0/0)**.



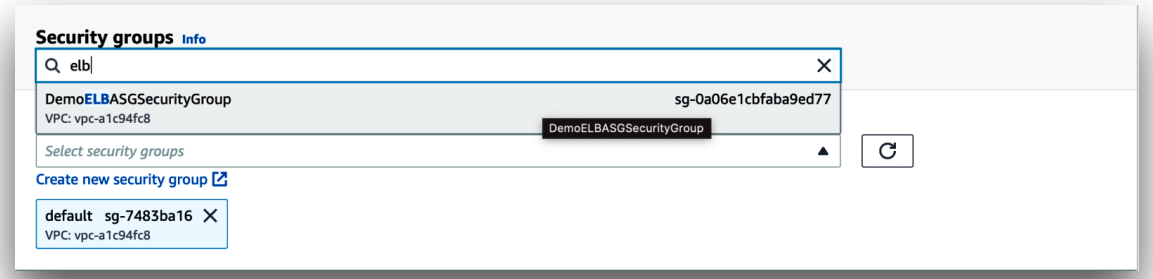
28. Keep the outbound rules to default and choose **Create Security Group**.



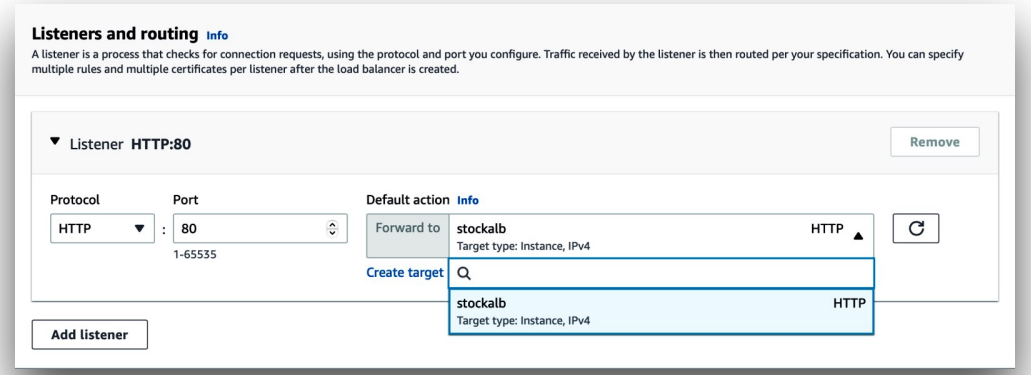
29. The security group is now created successfully. The same one will be assigned to the load balancer and EC2 instances.



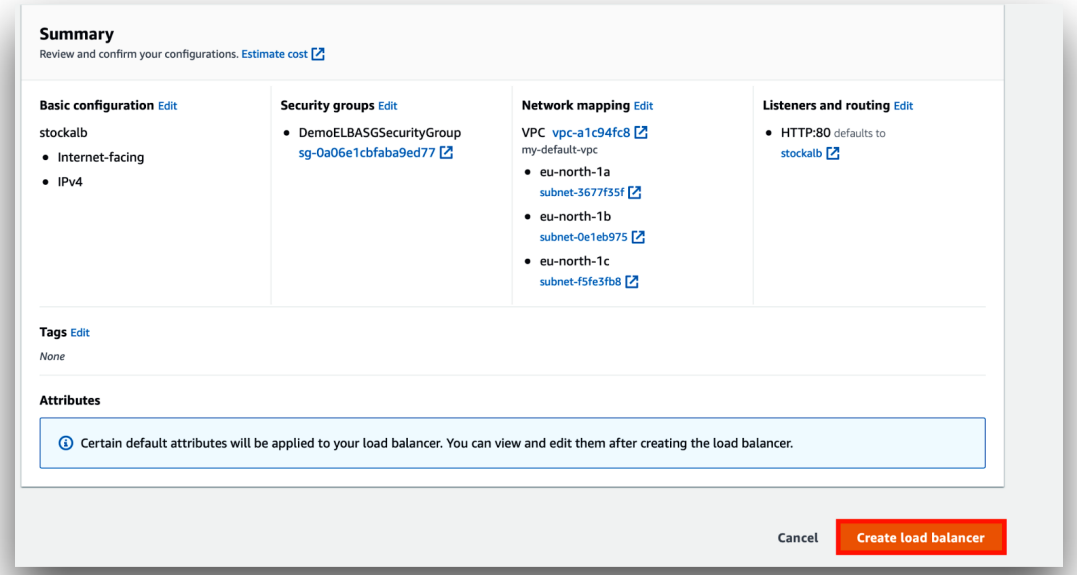
30. Switch back to the load balancer dashboard, hit refresh, remove the default security group and then select the one created and configured above.



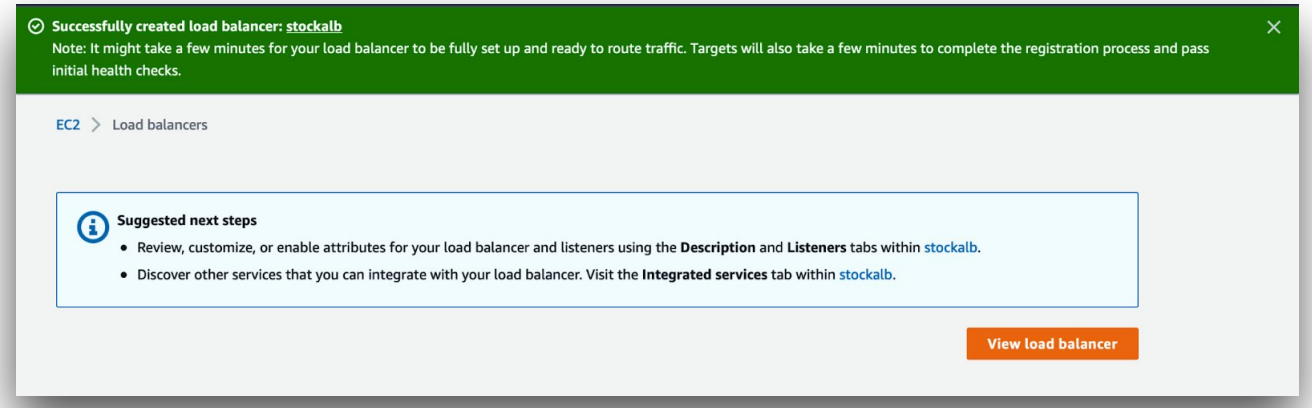
31. For **Listeners and routing**, the default listener accepts HTTP traffic on port 80. For **Default action**, choose the target group that you created.



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



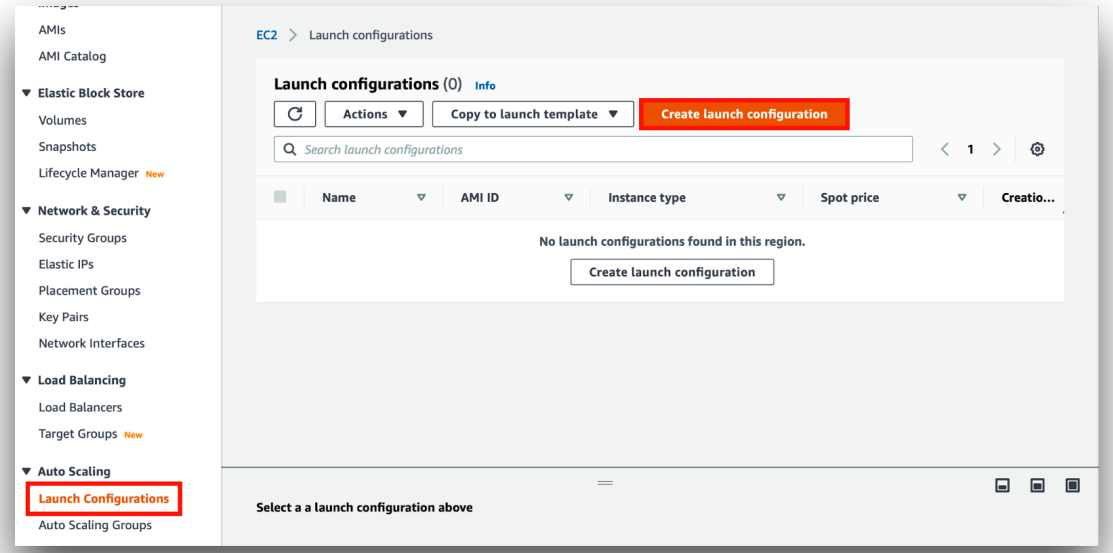
33. After the load balancer is created, choose **View Load Balancer**.



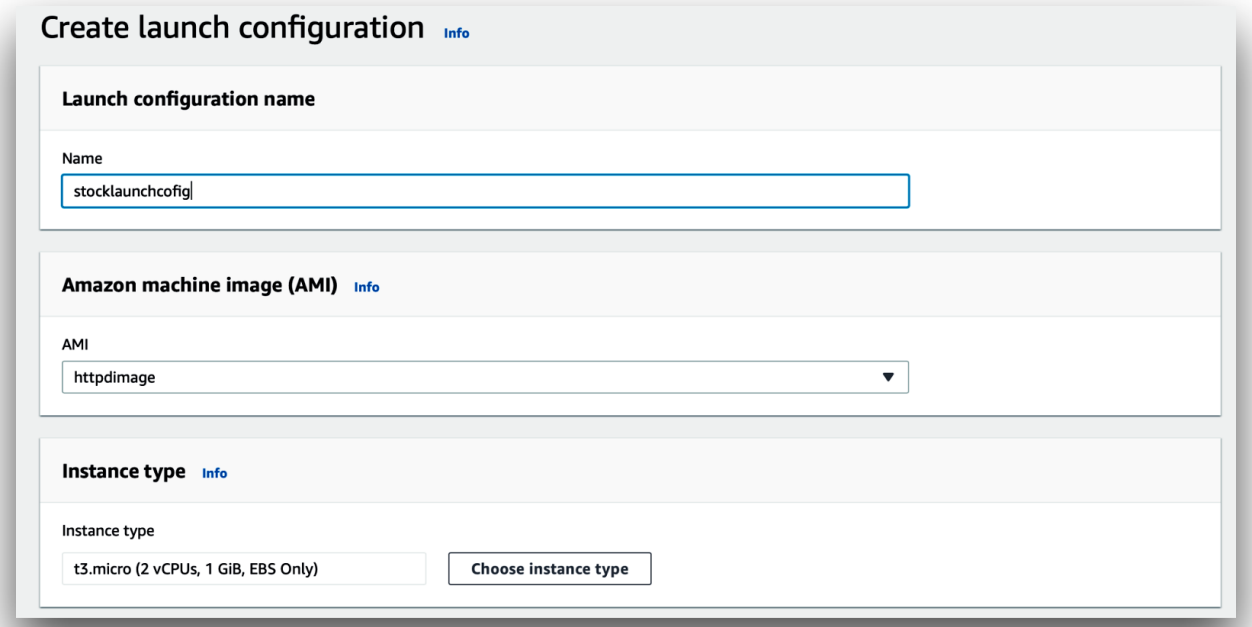
Create a Launch Configuration

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.

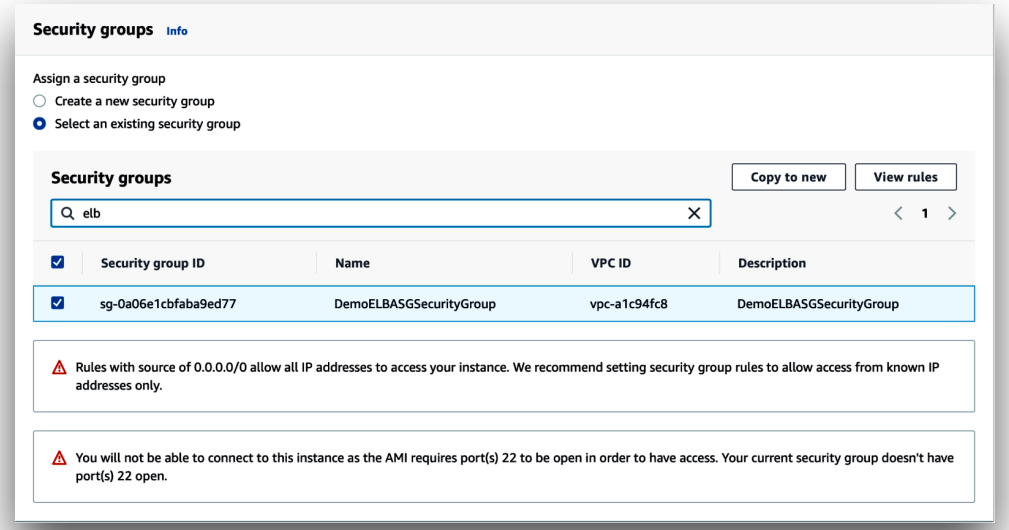
34. On the navigation pane, under **Auto Scaling**, choose **Launch Configurations** and then choose **Create launch configuration**.



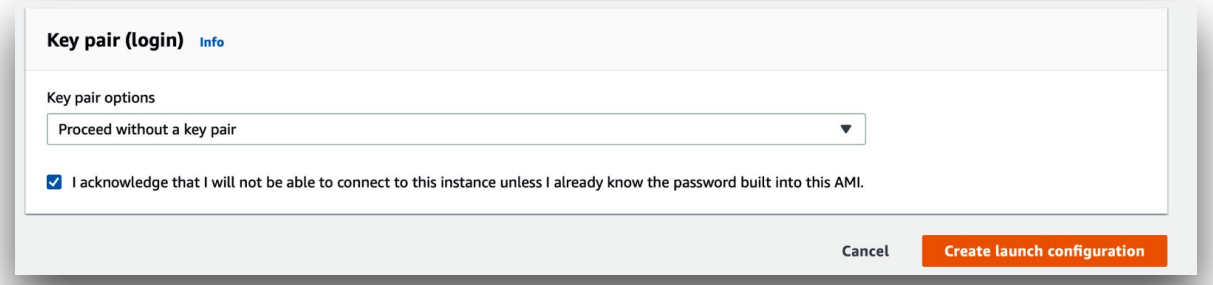
35. Enter a name for your launch configuration. For **Amazon machine image (AMI)**, choose the custom AMI you've created in the second step. For **Instance type**, choose either t2.micro or t3.micro (as they're free tier eligible).



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



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



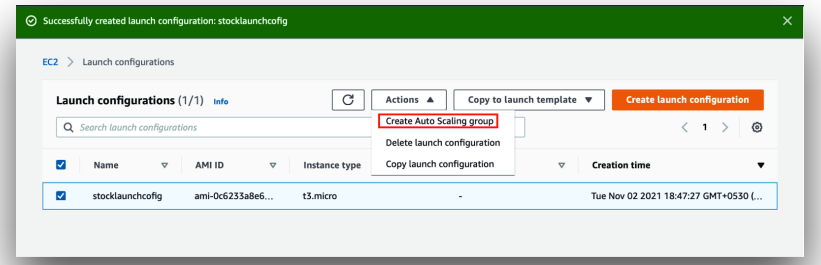
Create an Auto Scaling Group

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.

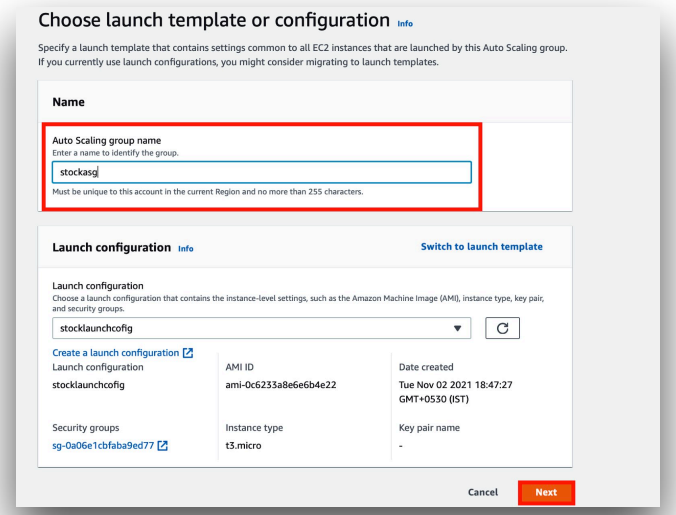
The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

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**.

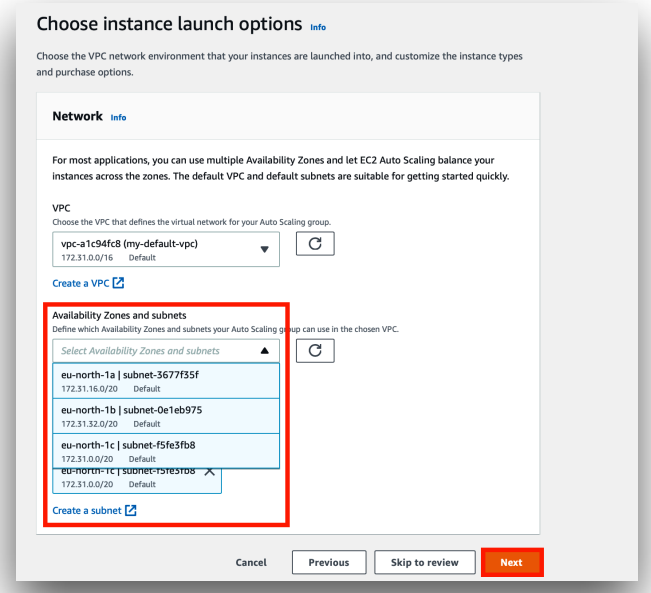


39. On the **Choose launch template or configuration** page, for **Auto Scaling group name**, enter a name for your Auto Scaling group. Verify that your launch configuration supports all of the options that you are planning to use, and then choose **Next**.



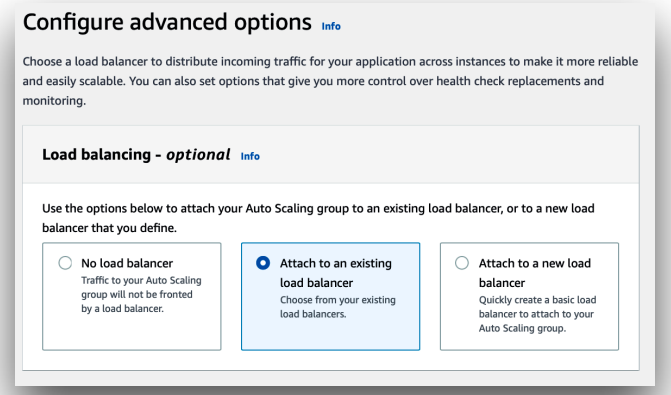
40. On the **Configure instance launch options** page, under **Network**, for **VPC**, choose the default VPC.

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:

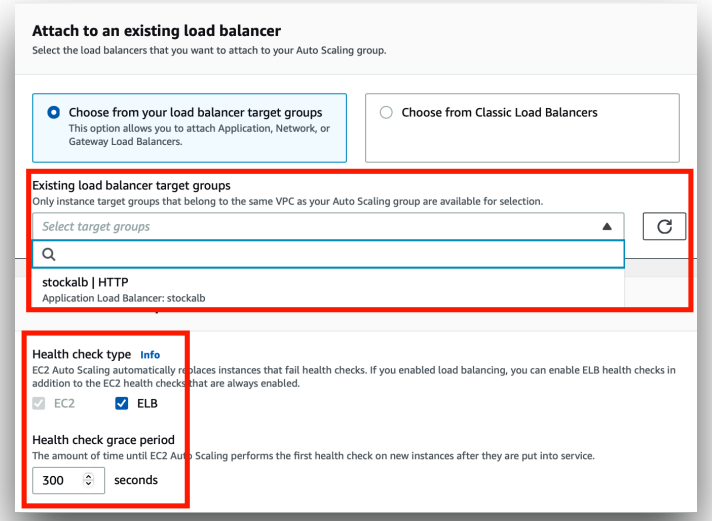
A. For **Load balancing**, choose **Attach to an existing load balancer**.



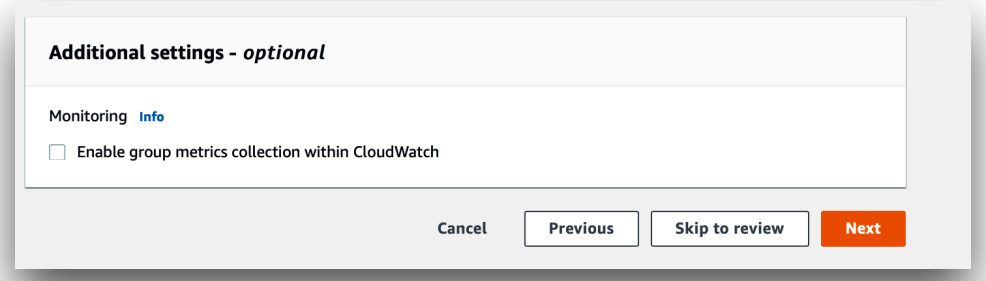
B. Under **Attach to an existing load balancer**, choose **Choose from your load balancer target groups**, and then choose a target group in the **Existing load balancer target groups** field.

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**.



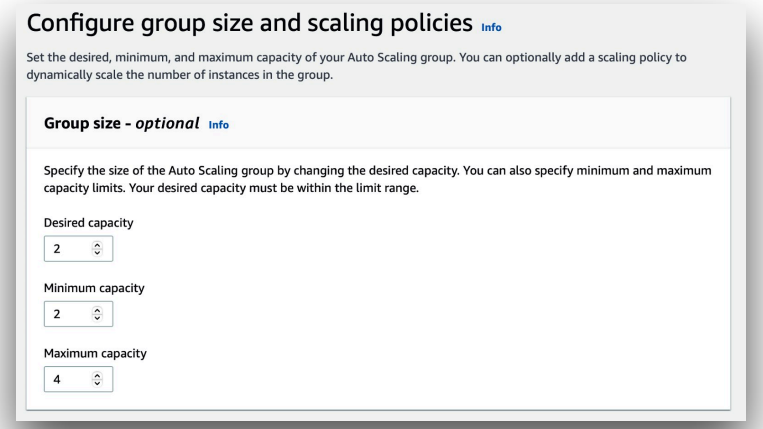
Create an Target Tracking Policy

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.

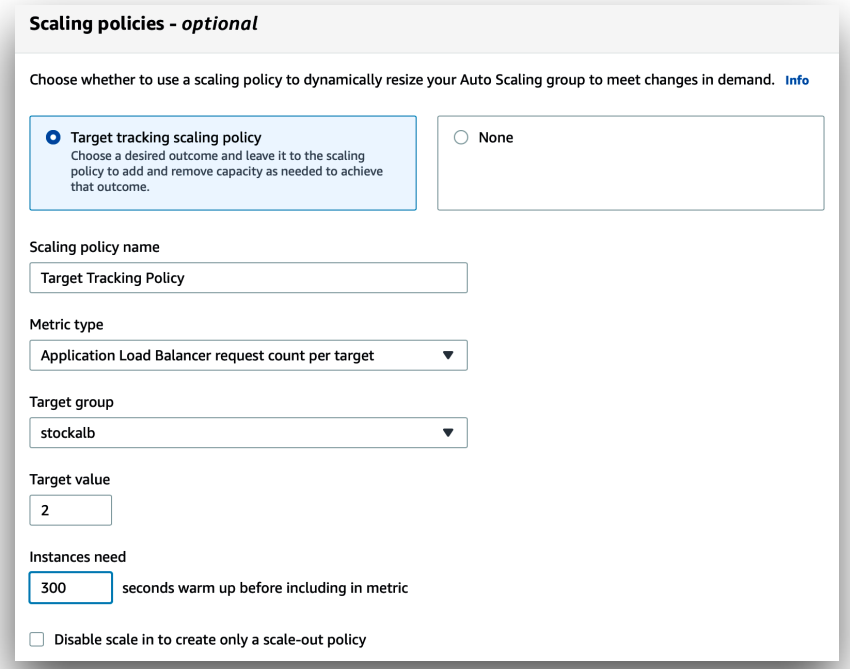
For this hands-on, set the Desired and Minimum capacity as 2, and the Maximum capacity as 4.



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

A. Specify a name for the policy.

B. For **Metric type**, choose **Application Load Balancer request count per target**



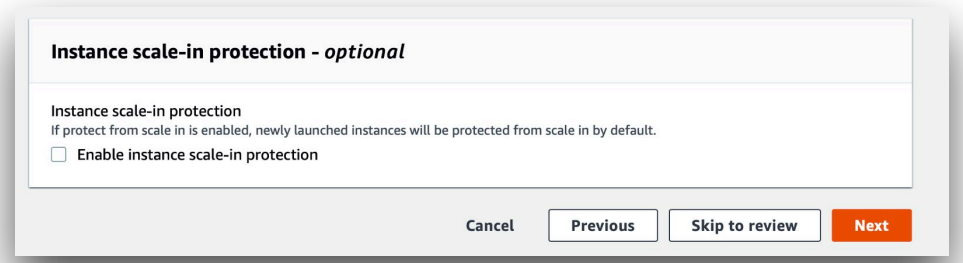
C. For **Target group**, select the one created in steps 16-20.

D. Specify a **Target value** for the metric. Enter a lower value such as **2** so that we can get faster results.

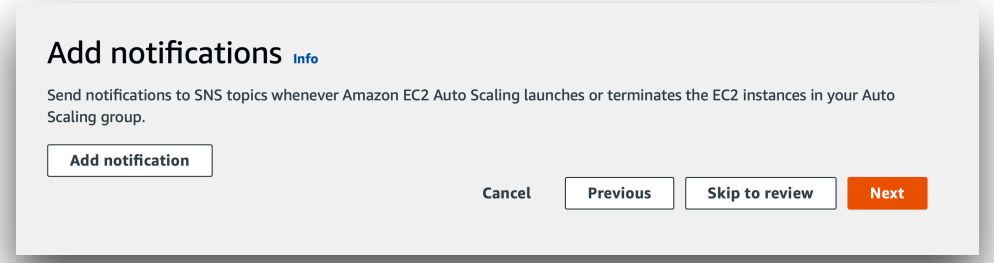
E. Specify an instance warm-up value for **Instances need**. This allows you to control the time until a newly launched instance can contribute to the CloudWatch metrics. Enter **300** as the value here.

F. Keep **Disable scale in to create only a scale-out policy** option as unchecked.

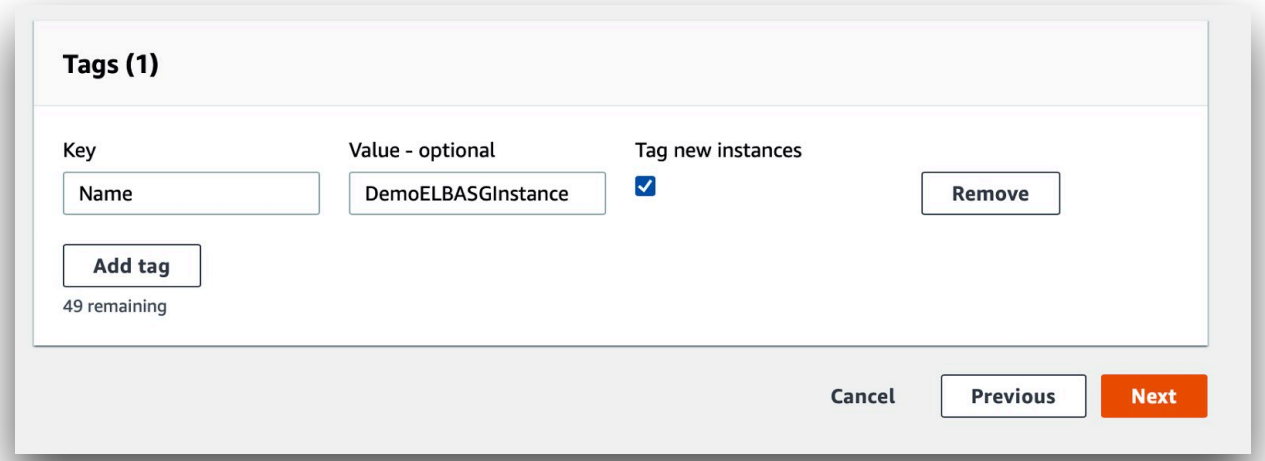
44. Choose **Next** to proceed.



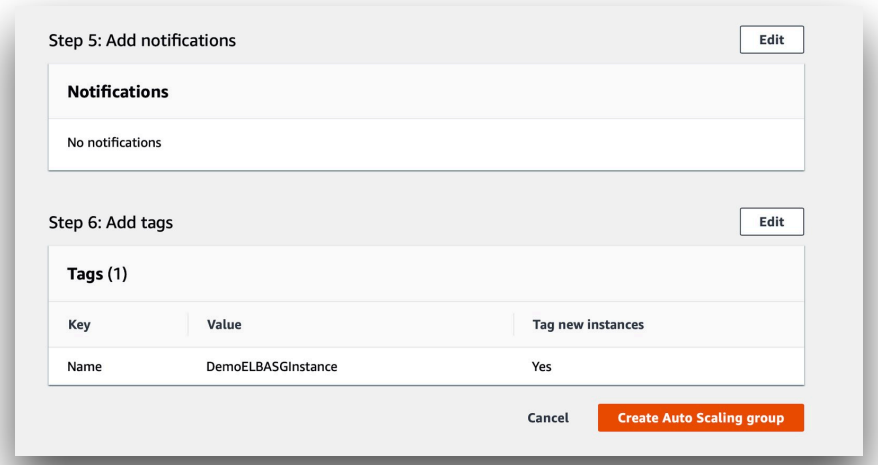
45. (Optional) To receive notifications, for **Add notification**, configure the notification, and then choose **Next**.



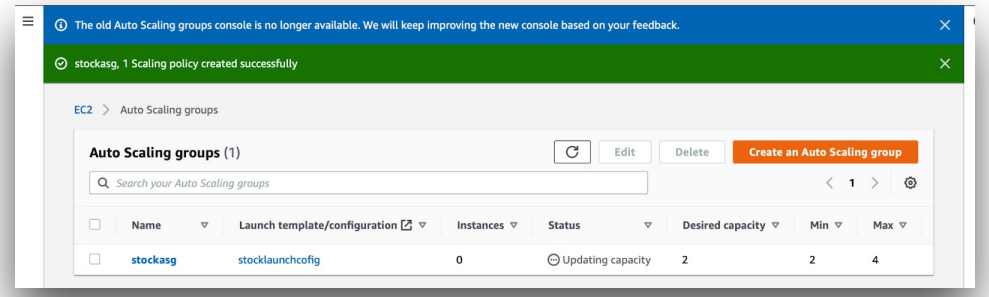
46. (Optional) To add tags, choose **Add tag**, provide a tag key and value for each tag, and then choose **Next**.



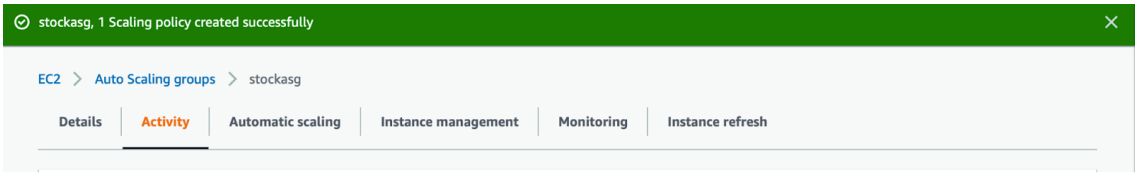
47. On the **Review** page, choose **Create Auto Scaling group**.



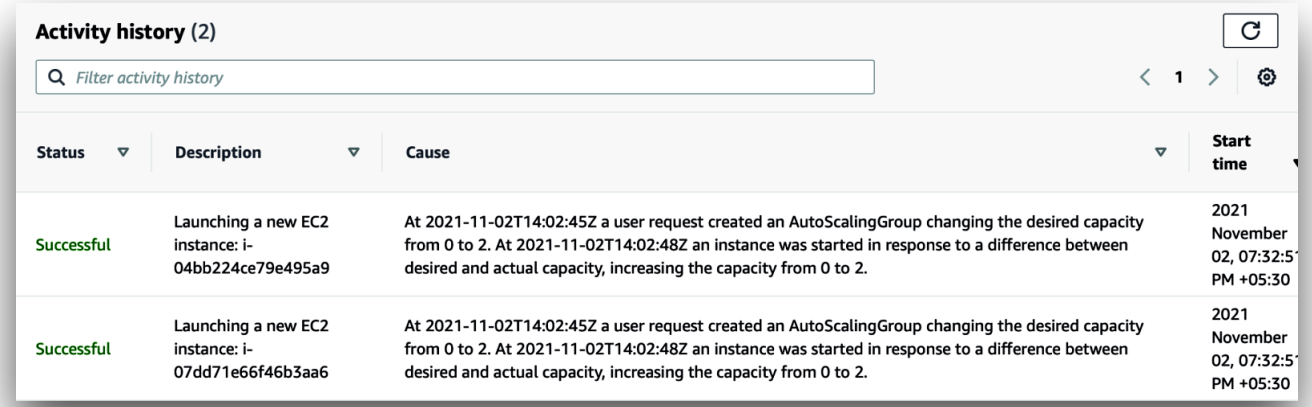
48. This marks the completion of the process of creating and configuring an Auto Scaling group.



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



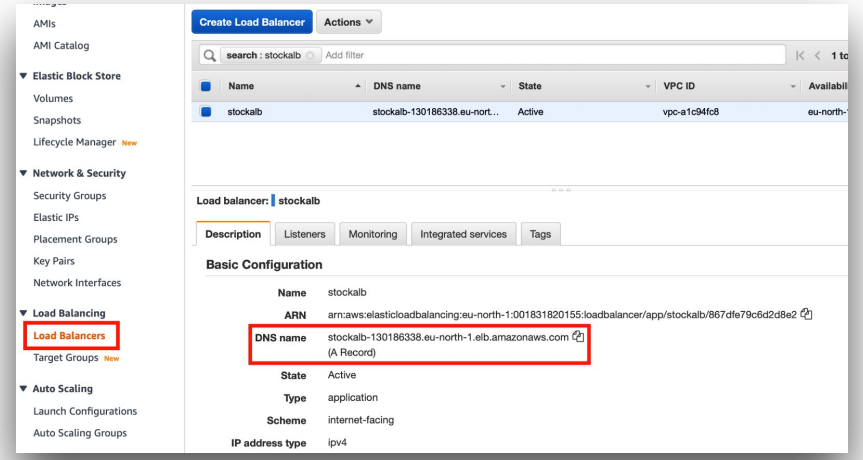
50. Within **Activity history**, you can see the list of instances launched in this Auto Scaling group.



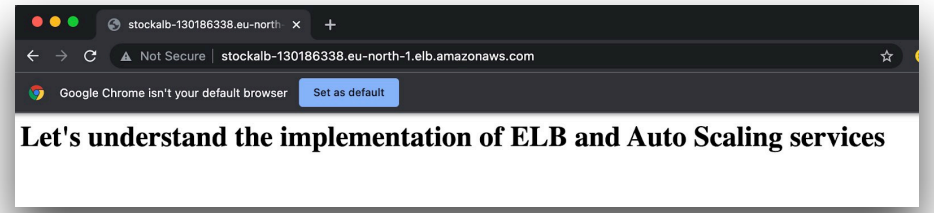
Test your Deployment

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).

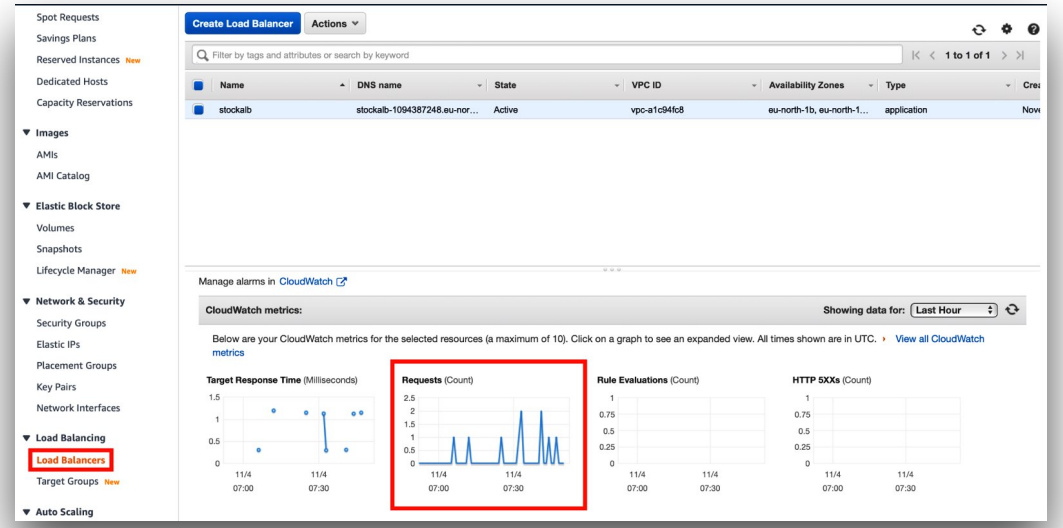


52. Paste the DNS name into the address field of an internet-connected web browser. If everything is working, the browser displays the default page of your server.

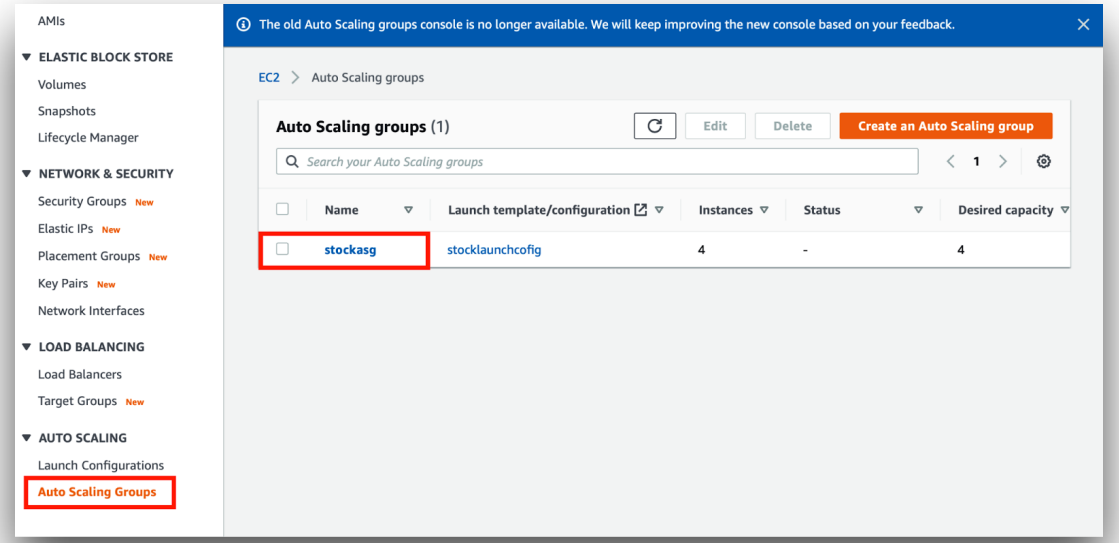


- *Refresh the browser several times so that the Application Load Balancer request per count per target exceeds the pre-defined limit.*

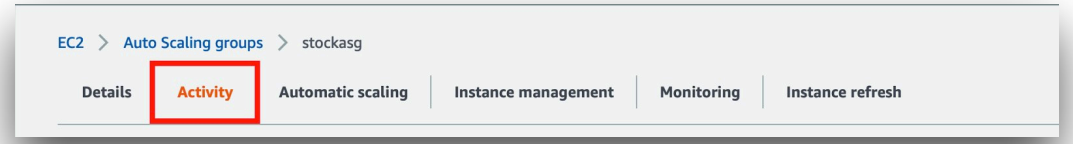
53. You can also check the request count by clicking on **Load Balancer** in the navigation menu, select the load balancer you've deployed, go to **Monitoring** and check **Requests (count)**.



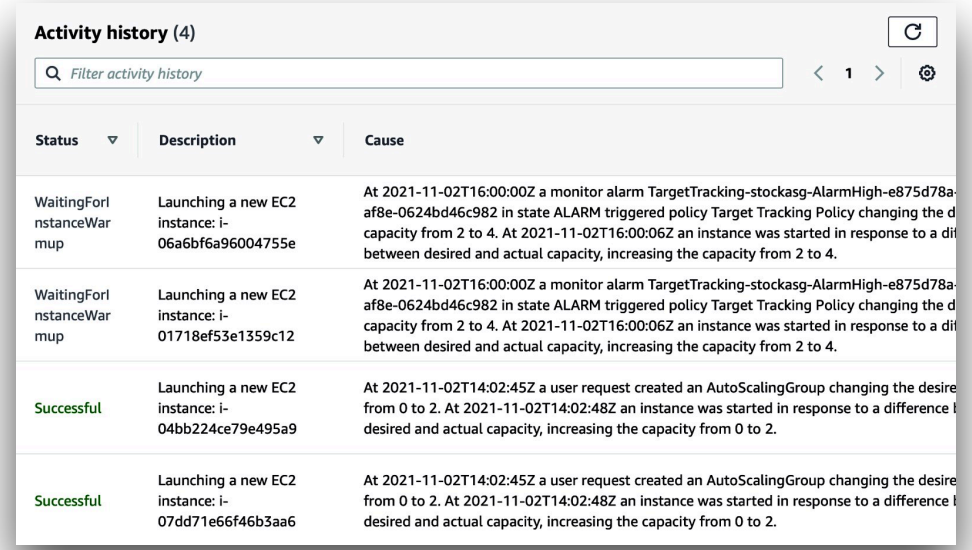
54. In the navigation pane, select **Auto Scaling Groups** and click on the Auto Scaling group.



55. Choose **Activity**.



56. Scroll down and view the activity logs.



- *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