Guided Lab: Scaling EC2 instances using Auto-Scale Group (ASG)

Description

EC2 Auto Scaling ensures that you maintain application availability and lets you scale Amazon EC2 capacity up or down automatically according to the conditions you define. This represents horizontal scaling, as opposed to increasing the specifications of a single instance (vertical scaling).

Prerequisites

This lab assumes you have experience creating EC2 instances and are familiar with its basic components.

If you find any gaps in your knowledge, consider taking the following labs:

- Creating an Amazon EC2 instance (Linux)

Objectives

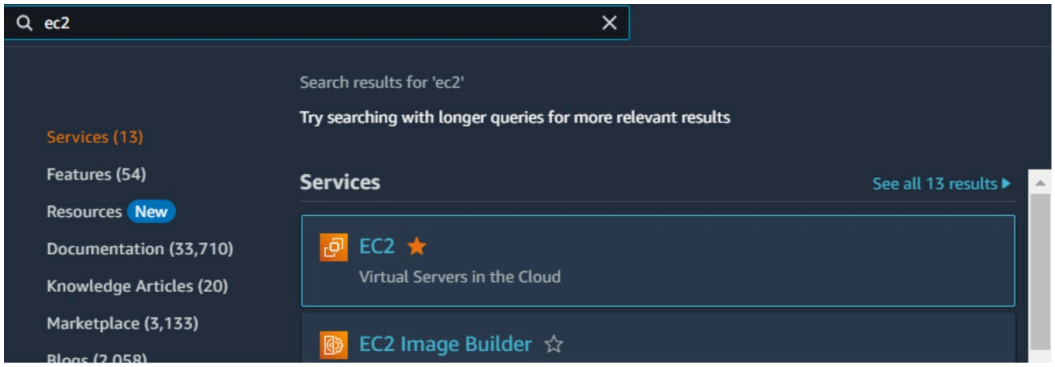
In this lab, you will:

- Understand the importance of EC2 Auto Scaling.
- Learn to set up Launch Templates.
- Experience the dynamic adjustment of EC2 capacity with Auto Scaling Groups.

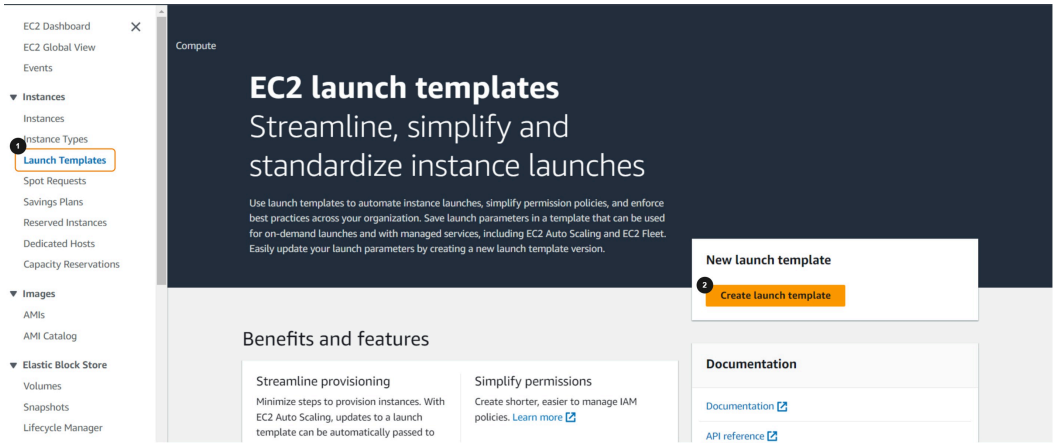
Lab Steps

Creating a Launch template

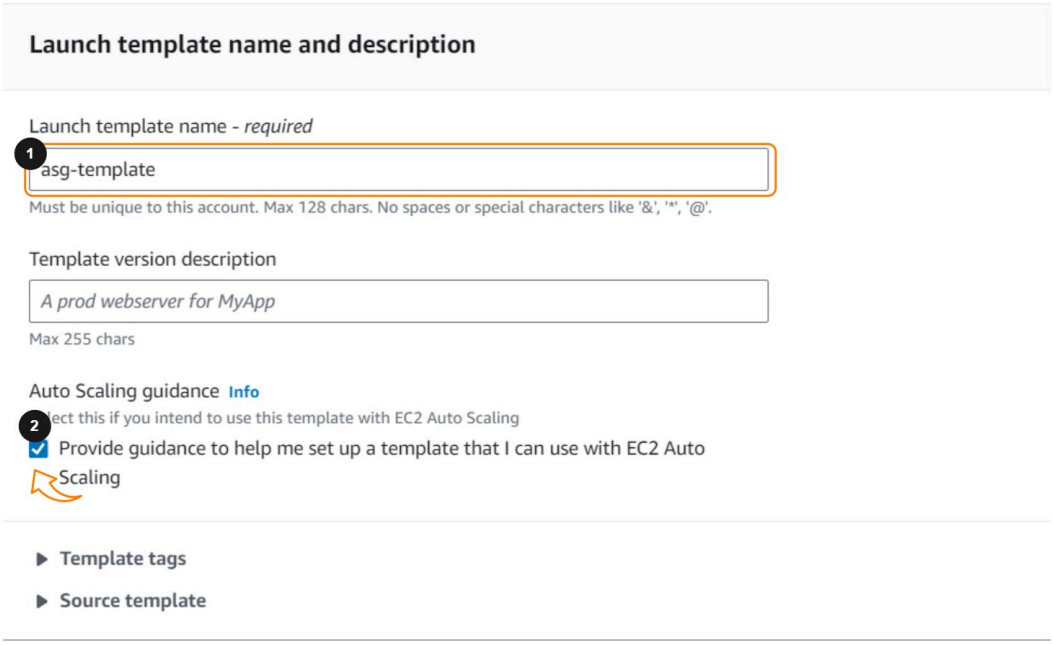
1. Search '*ec2*' in the AWS Management Console search bar. Click **EC2** on the search results.



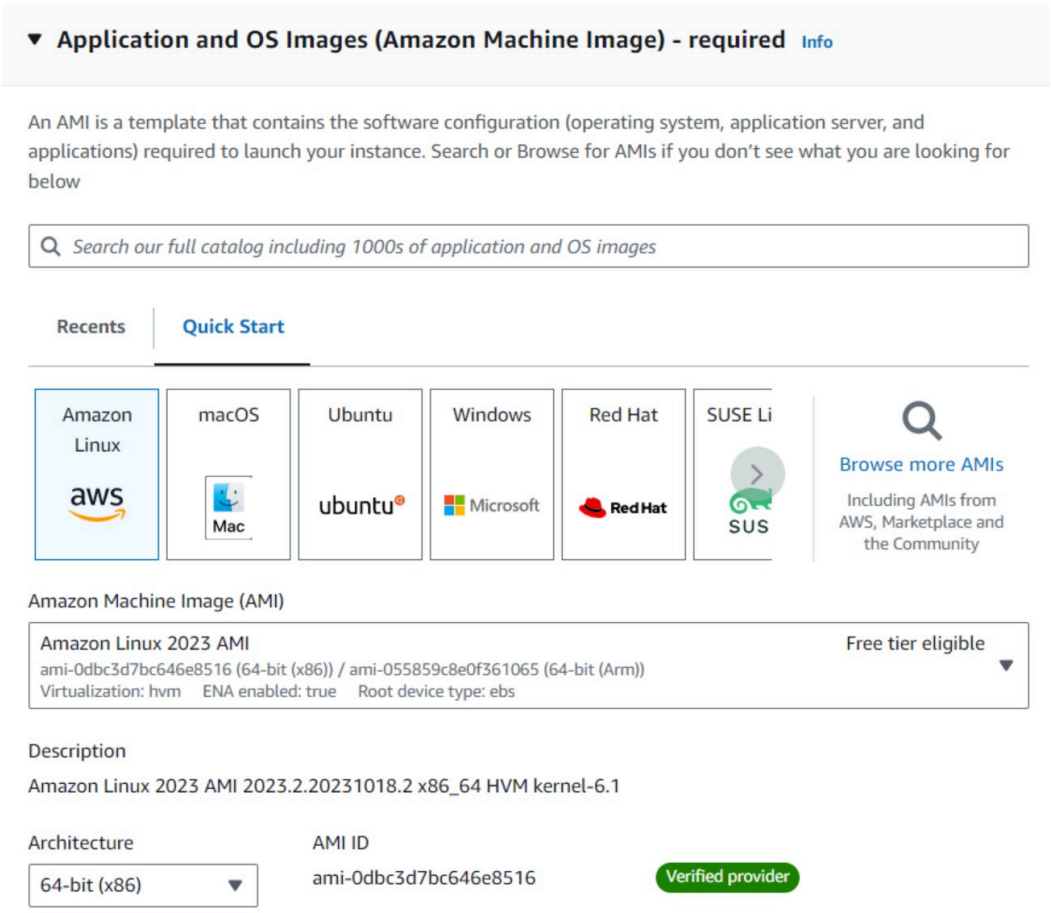
2. On the left window, under **Instances**, select **Launch Templates**, then click **Create launch template**.



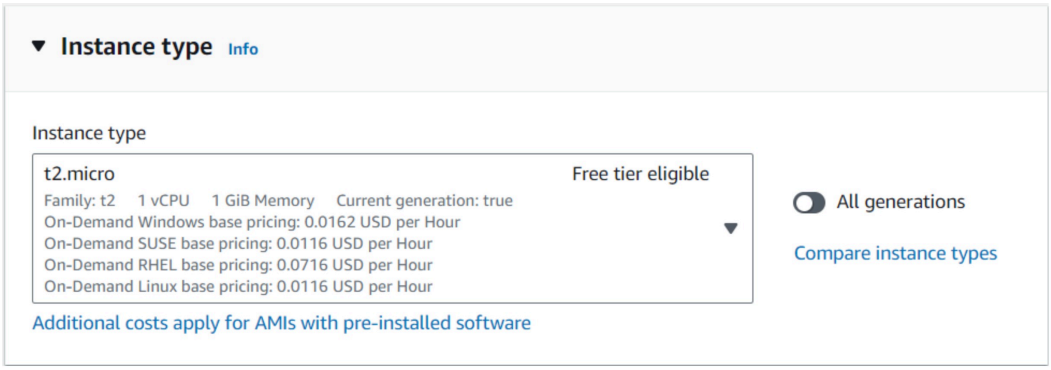
3. Enter '*asg-template*' for the template name and tick the **Auto Scaling guidance** checkbox.



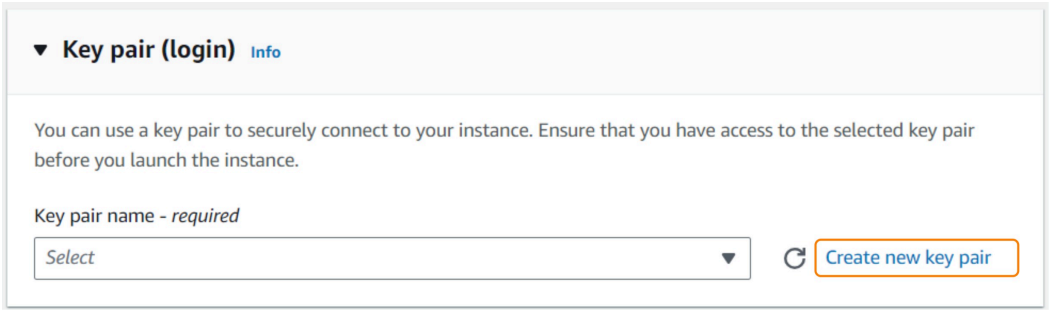
4. Under **Application and OS Images**, click the default **Amazon Linux** AMI.



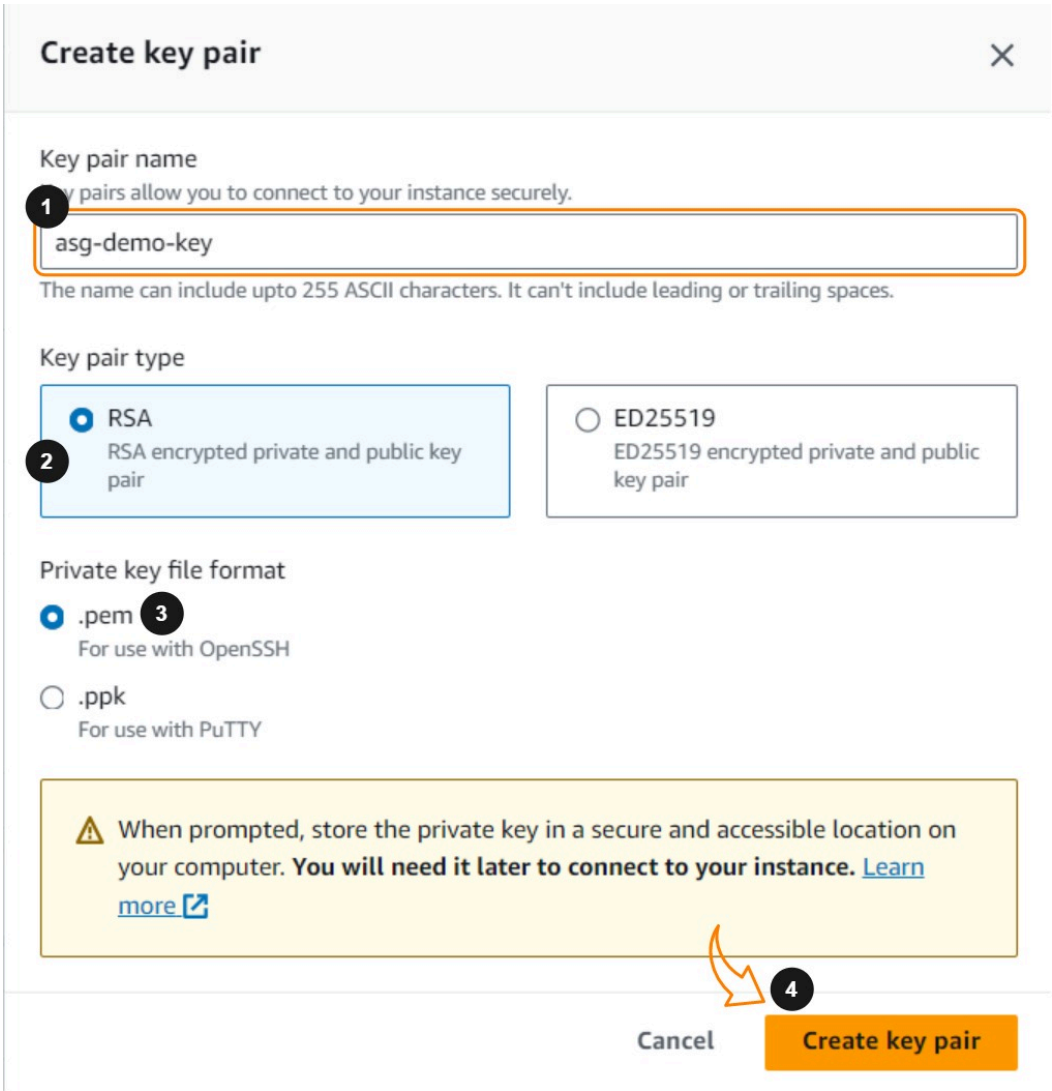
5. Under the **Instance Type** section, select **t2.micro**.



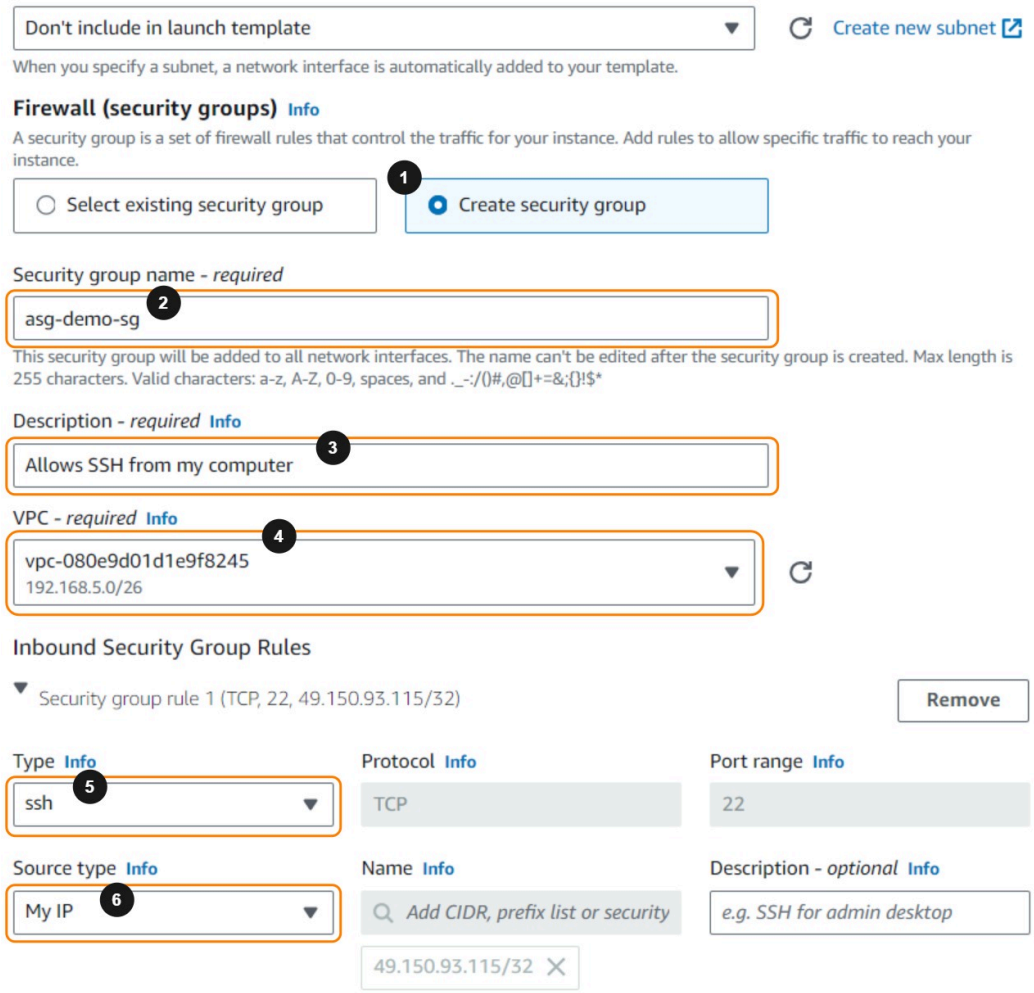
6. Under the **Key Pair** section, click **Create new key pair**.



7. Enter '*asg-demo-key*' for the key pair name and follow the configurations below. Finally, click **Create key pair**.

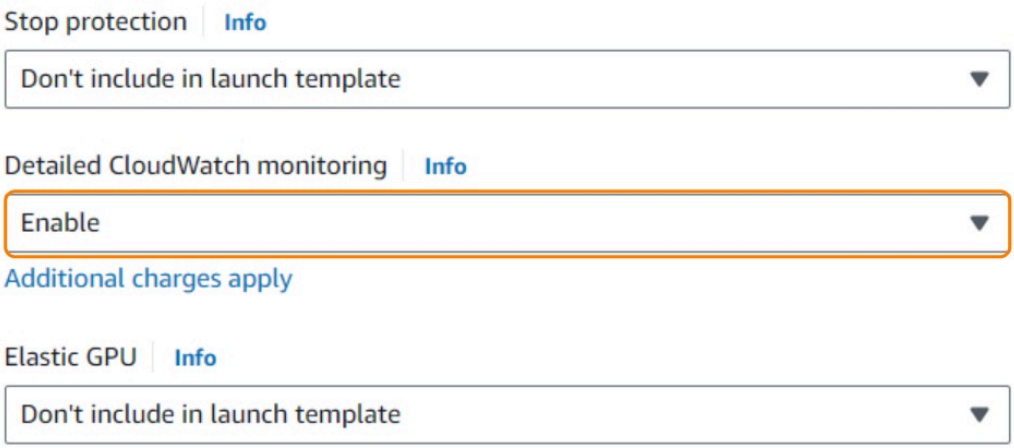


8. Under **Network settings**, follow the configurations shown in the screenshot below.



9. Scroll down to the bottom page and click the **Advanced details** dropdown menu.

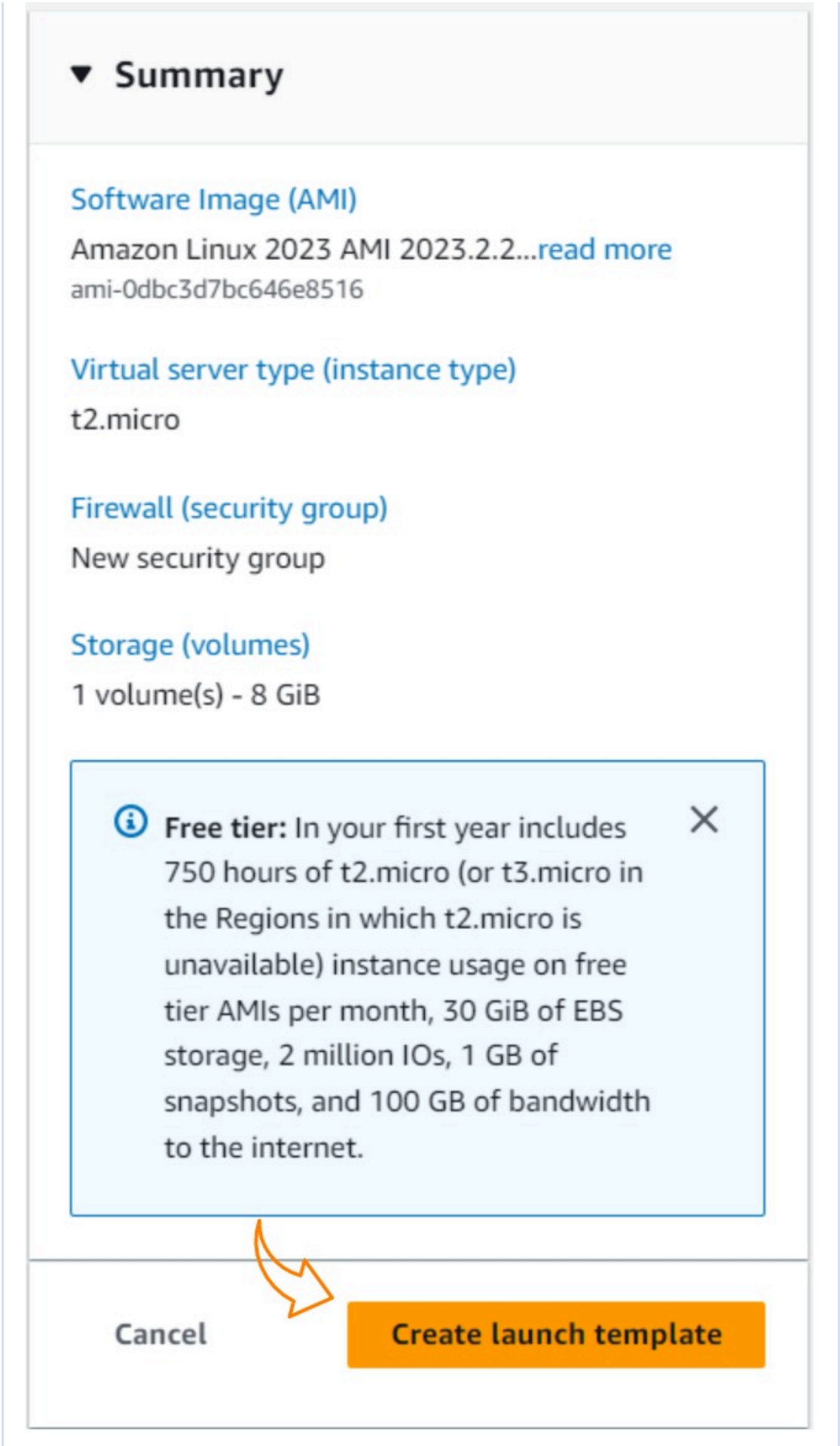
10. Select **Enable** on the dropdown menu of the **Detailed CloudWatch monitoring** option.



By default, Amazon EC2 sends metrics to CloudWatch every 5 minutes. This includes metrics like CPU utilization, disk reads/writes, and network packets in/out. For more granular insight, you can opt for detailed monitoring. This captures the same metrics but updates

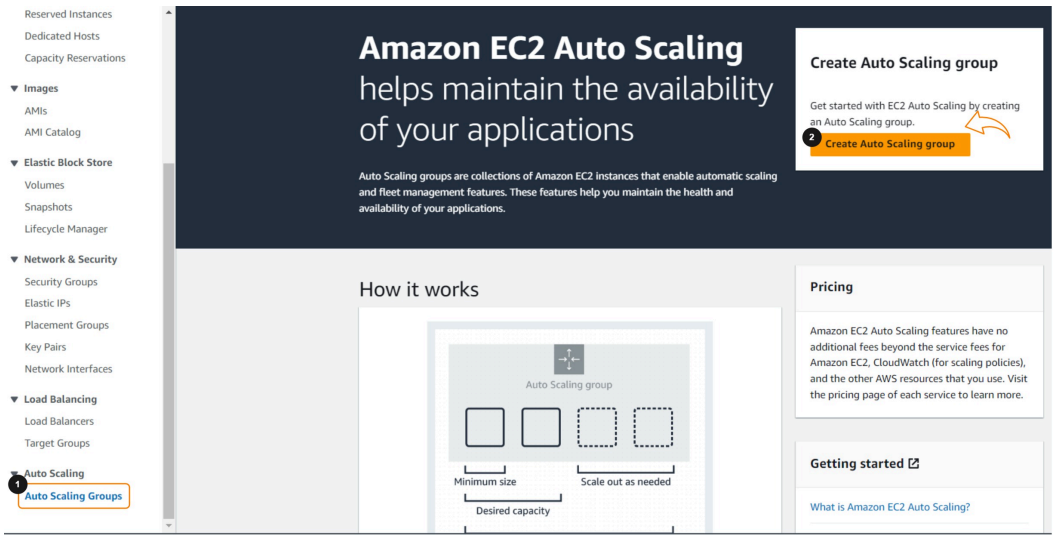
them every 1 minute. In this lab, we'll use Detailed Monitoring to get quicker feedback, enabling our Auto Scaling Group to scale in/out faster during our tests.

11. On the right window, click the **Create launch template** button.

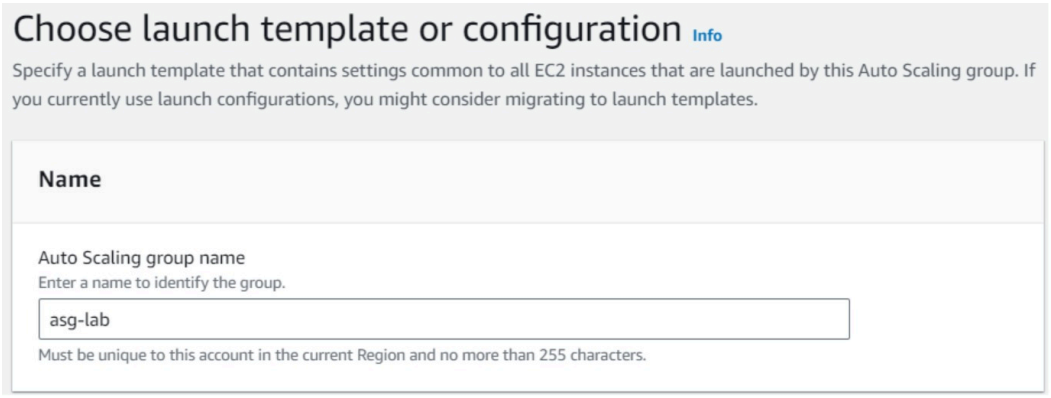


Creating an Auto Scaling Group

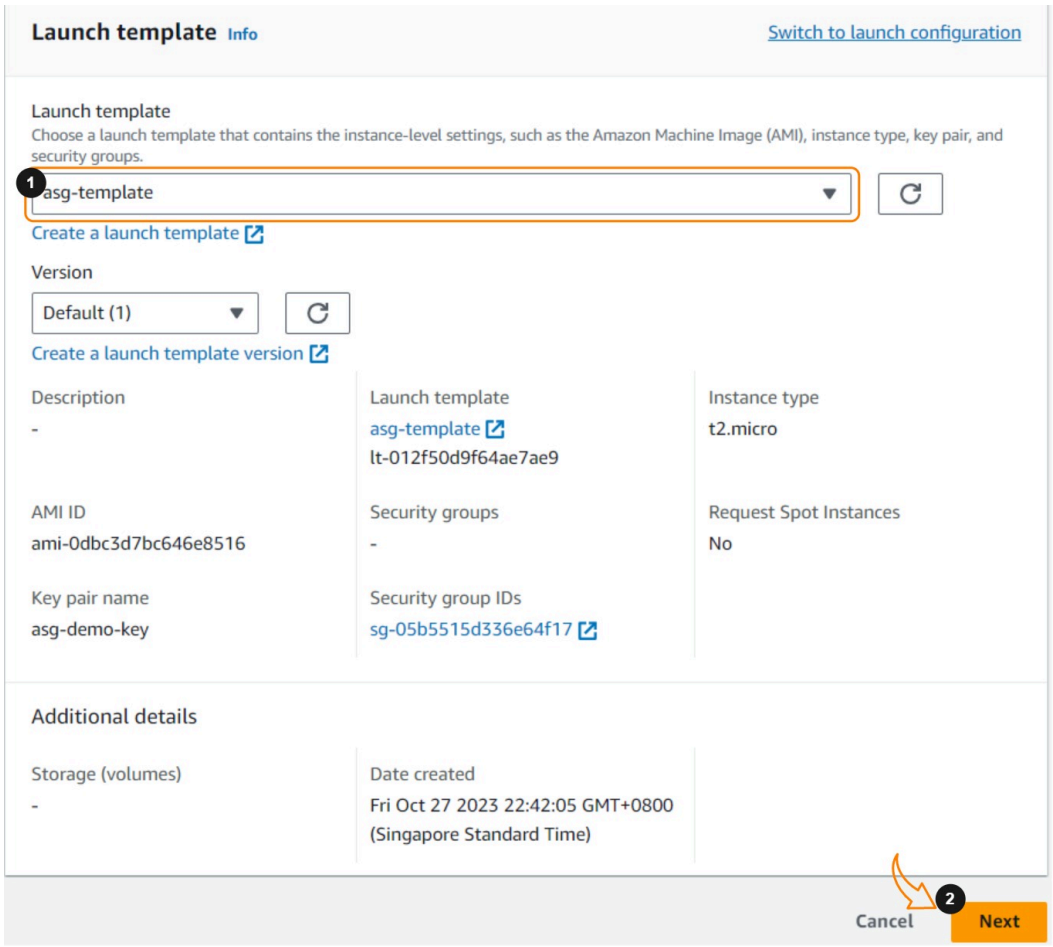
12. In the Amazon EC2 Console, under the **Auto Scaling** menu, click **Auto Scaling Groups**, then click the **Create Auto Scaling group** button.



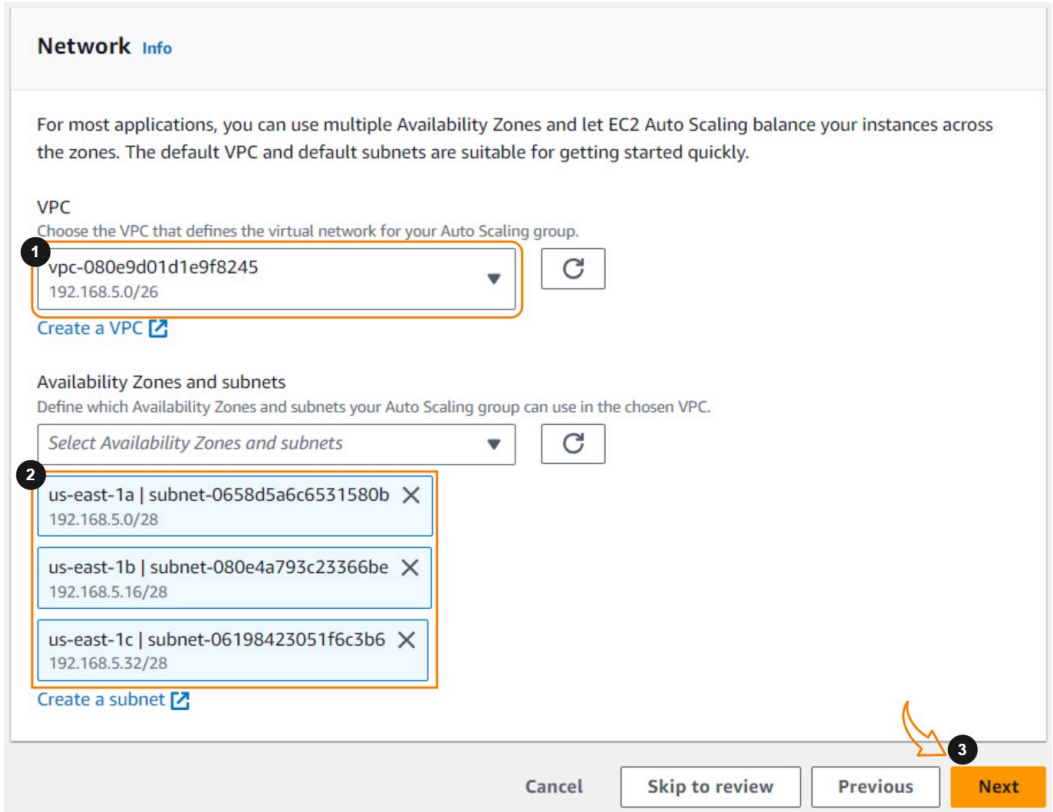
13. Enter '*asg-lab*' into the **Auto Scaling group** name field.



14. In the Launch template section, choose the '*asg-template*' that we created in the **Creating a Launch template** section.



15. In the **Network** section, select the default VPC in the VPC dropdown menu. From the **Availability Zones and subnets** dropdown menu, select all the default subnets. Click **Next**.

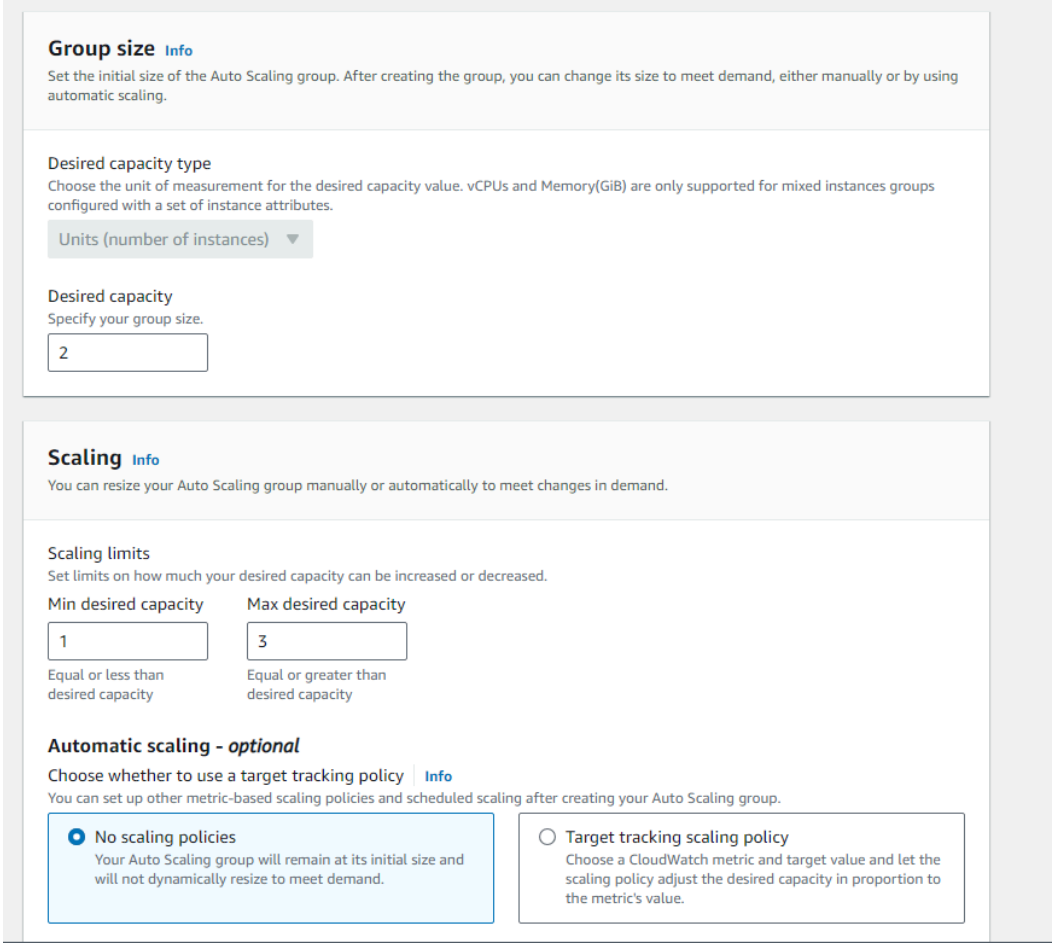


16. Scroll down to the bottom of **Configure advanced options – optional** page, and click **Next**.

In this lab, we'll focus on the basics of setting up an Auto Scaling Group. Therefore, we'll skip the "Configure advanced options" step. Our primary goal is to demonstrate how scaling works without diving into the more intricate configurations.

17. For the Group Size, set the following configurations:

- Desired Capacity: 2
- Minimum Capacity: 1
- Maximum Capacity: 3



We start with a **Desired Capacity** of 2, meaning that initially, we want two instances up and running. However, as demands shift and the need for resources changes, the ASG can adjust the number of instances. It will always keep at least 1 instance active, which is our **Minimum Capacity**. However, even during high demand, it won't spin up more than 3 instances, which is set by **Maximum Capacity**.

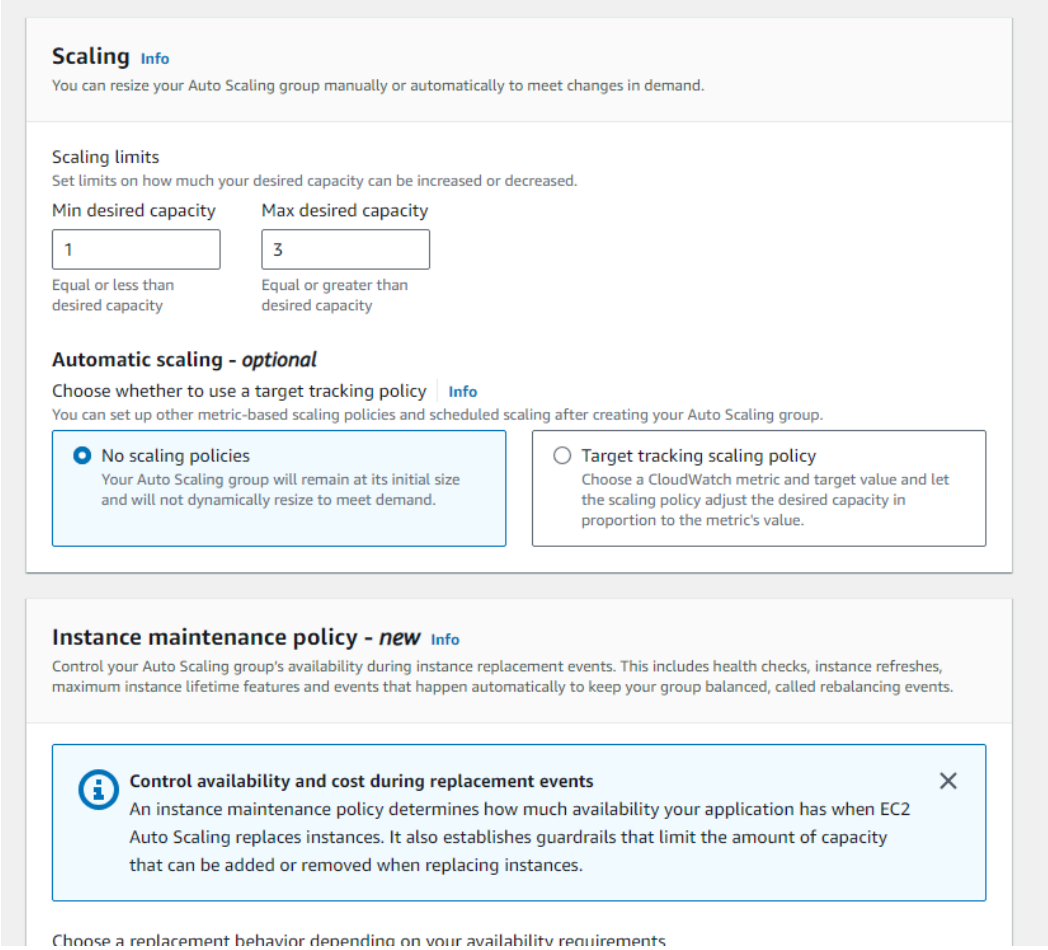
Amazon's Auto Scaling Groups (ASG) offers the following scaling policies to accommodate a variety of loads:

- Target tracking scaling – Increase and decrease the current capacity of the group based on an Amazon CloudWatch metric

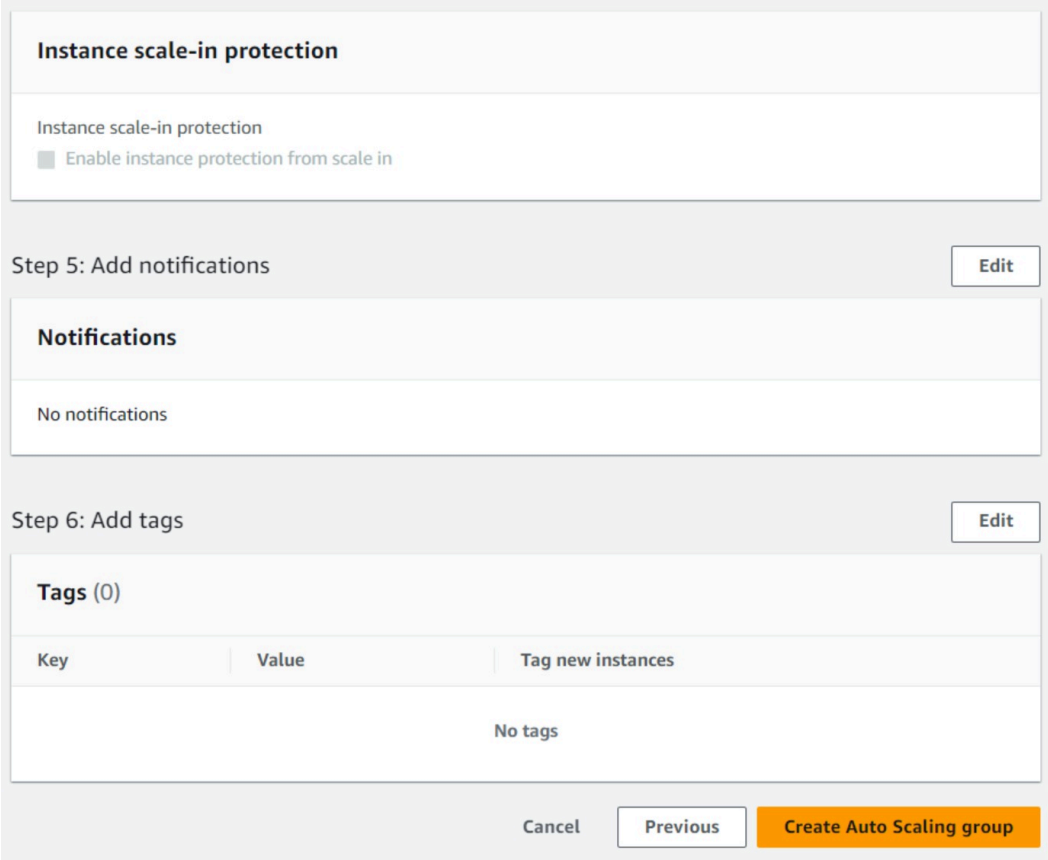
and a target value. It works similarly to the way that your thermostat maintains the temperature of your home—you select a temperature, and the thermostat does the rest.

- Step scaling – Increase and decrease the current capacity of the group based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
- Simple scaling – Increase and decrease the current capacity of the group based on a single scaling adjustment, with a cooldown period between each scaling activity.

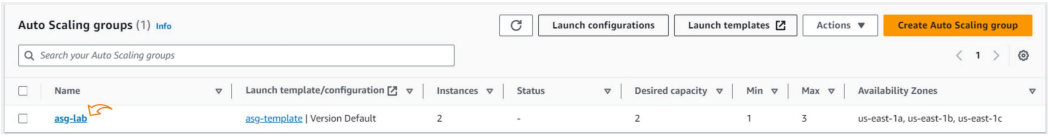
18. In the **Scaling policies** section, we'll set our choice to **None** for now and configure a Simple scaling policy later. To skip the succeeding optional steps, click the **Skip to review** button.



19. Click **Create Auto Scaling Group**.

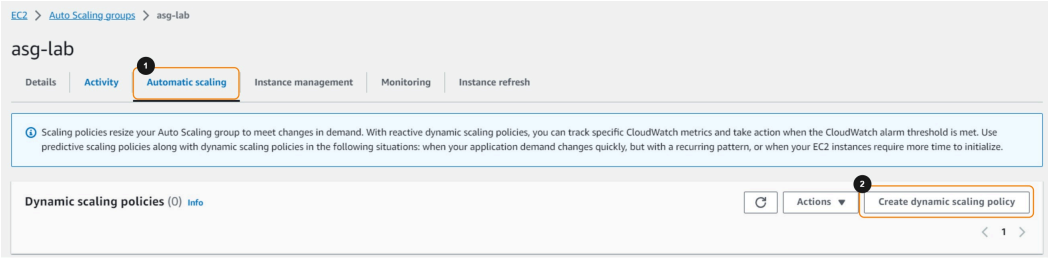


20. On the ASG dashboard. Click on the name of your ASG. This will take you to the **details page** for your Auto Scaling Group.



Configuring the Auto Scaling Group with a Simple Scaling Policy

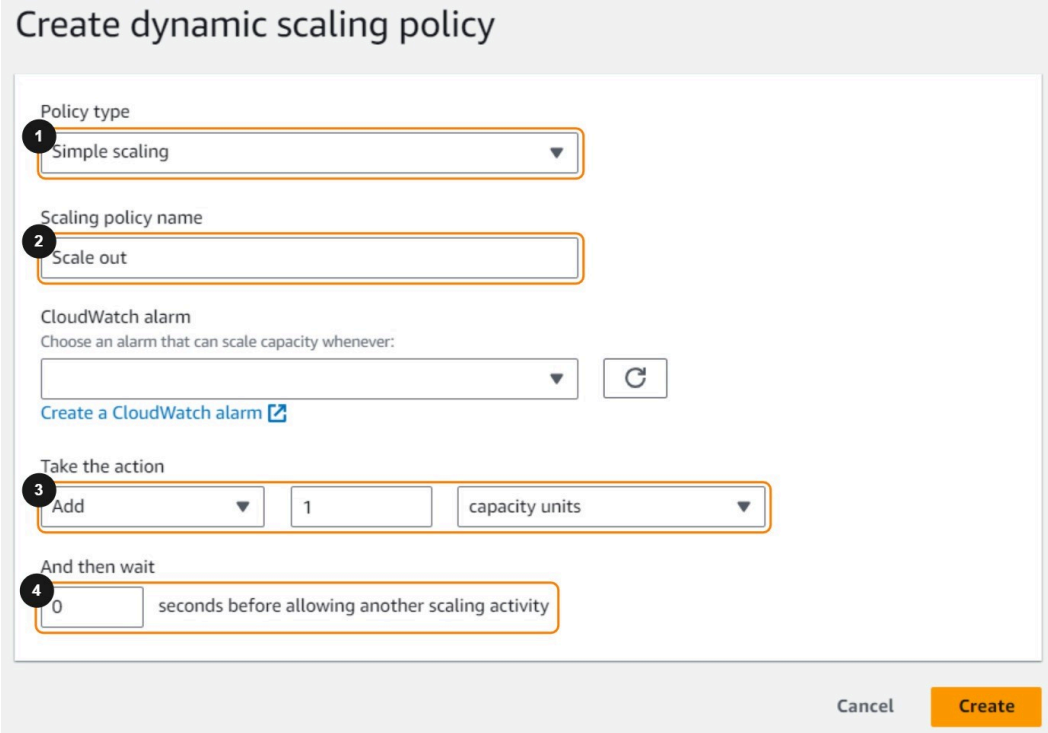
21. Click on the **Automatic scaling** tab, then click **Create dynamic scaling policy** button.



We will set up two distinct scaling policies: one for scaling out and another for scaling in. The scale-out policy is what prompts ASG to add more instances when demands surge, such as during high CPU utilization. Conversely, the scale-in policy reduces the number of instances when demand subsides.

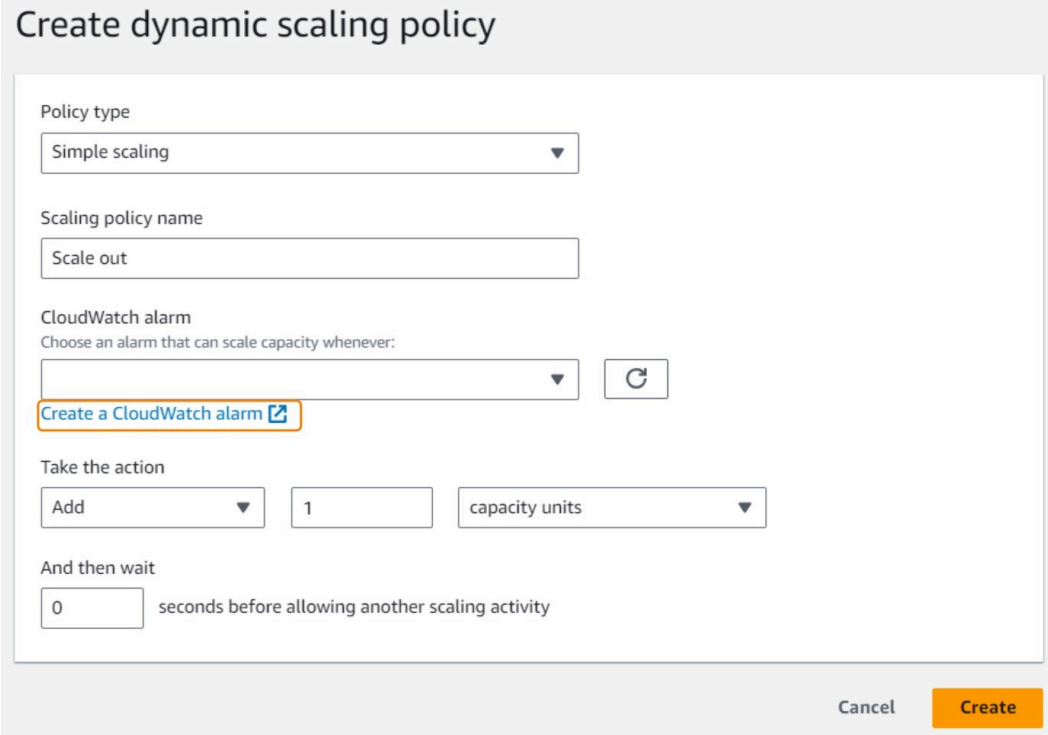
Let’s start by setting up the scale-out policy.

22. Create a scale-out policy by filling out the Scaling Policy fields with the following values.



Note: We’re setting the **seconds before allowing another scaling activity to **0** so ASG can respond immediately for demonstration purposes. In a real-world scenario, you’d typically use a longer cooldown to account for the time it takes an instance to start up and begin handling traffic.*

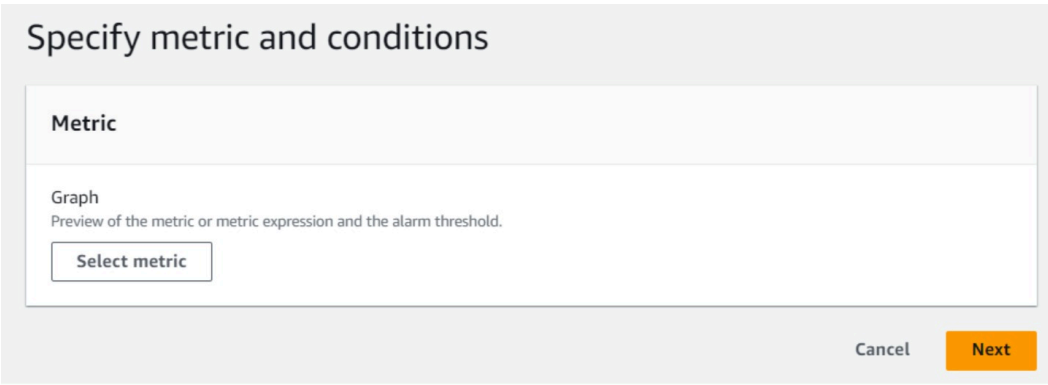
23. Click **Create a CloudWatch alarm**. This will open a new tab on the CloudWatch Console.



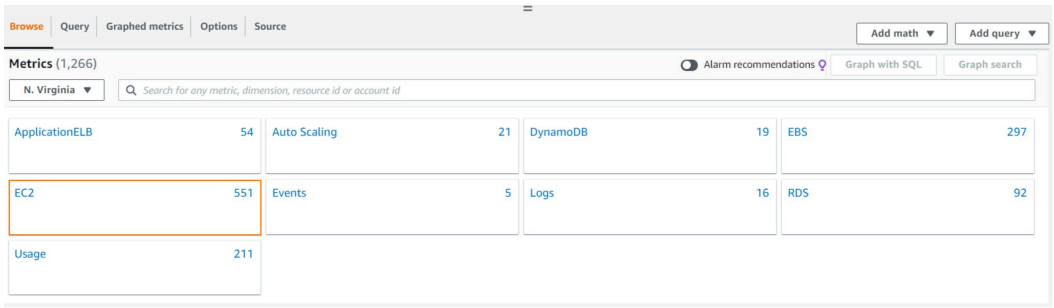
CloudWatch alarms play a pivotal role in the operations of ASGs. They monitor specific metrics, like CPU utilization, and when thresholds are breached, they signal ASG to scale in or out.

For this lab, we'll be creating two alarms based on CPU utilization: The first alarm will signal ASG to scale out if the CPU usage exceeds 30% over a 1-minute span. Conversely, the second alarm will signal ASG to scale in if the CPU usage dips below 30% for the same duration.

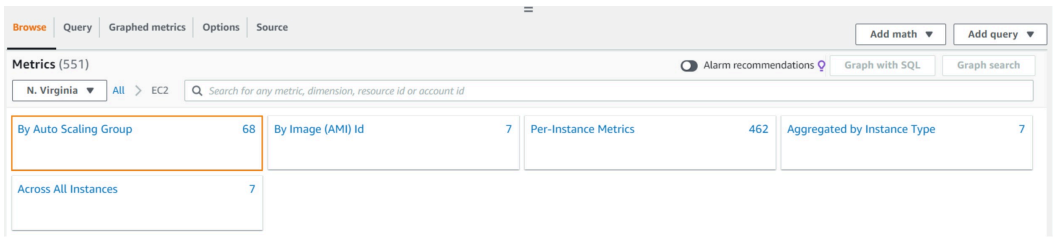
24. On **Specify metric and conditions**, click the **Select metric** button.



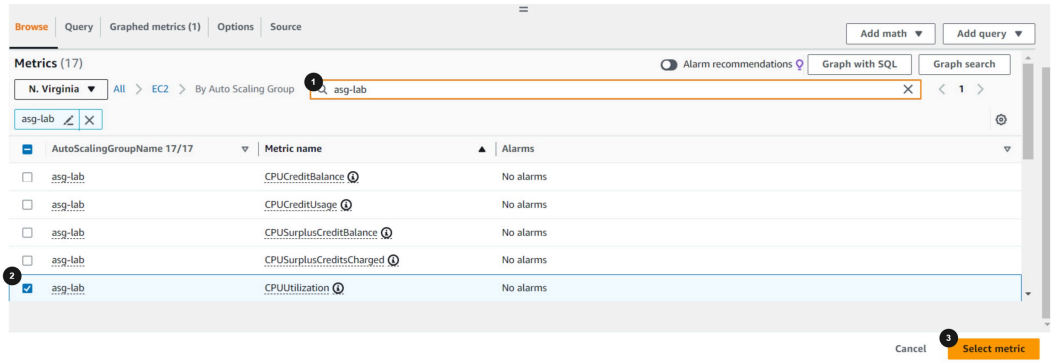
25. Select **EC2**.



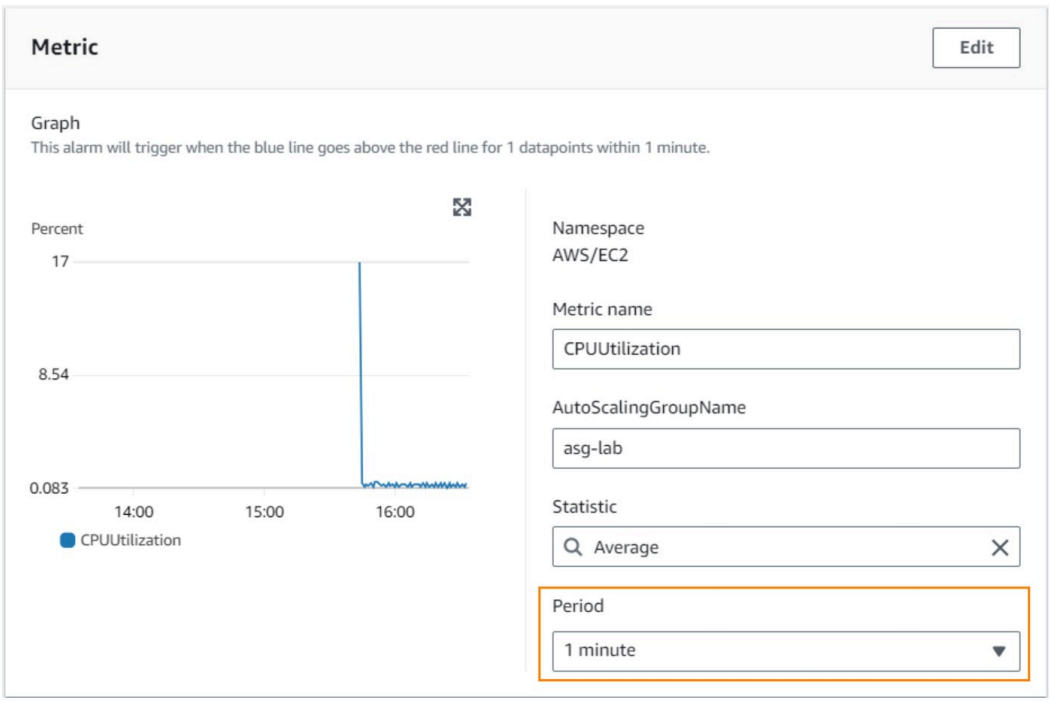
26. Select **By Auto Scaling Group.**



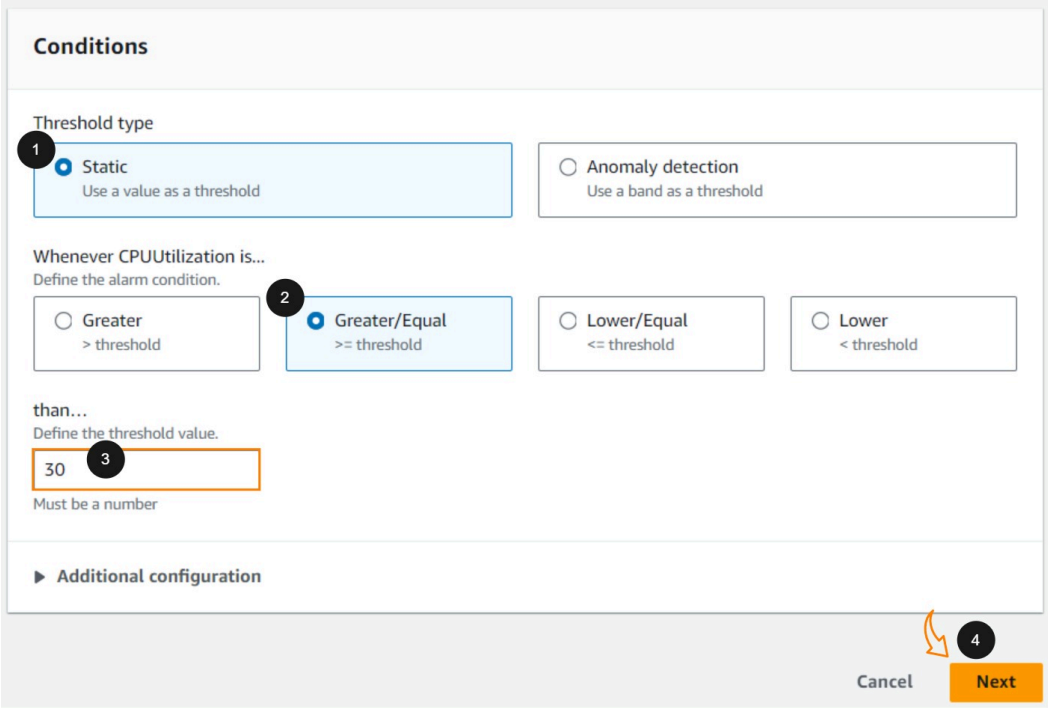
27. Enter ‘asg-lab’ into the search field. Toggle the checkbox for **CPUUtilization metric**. Then, click **Select Metric**.



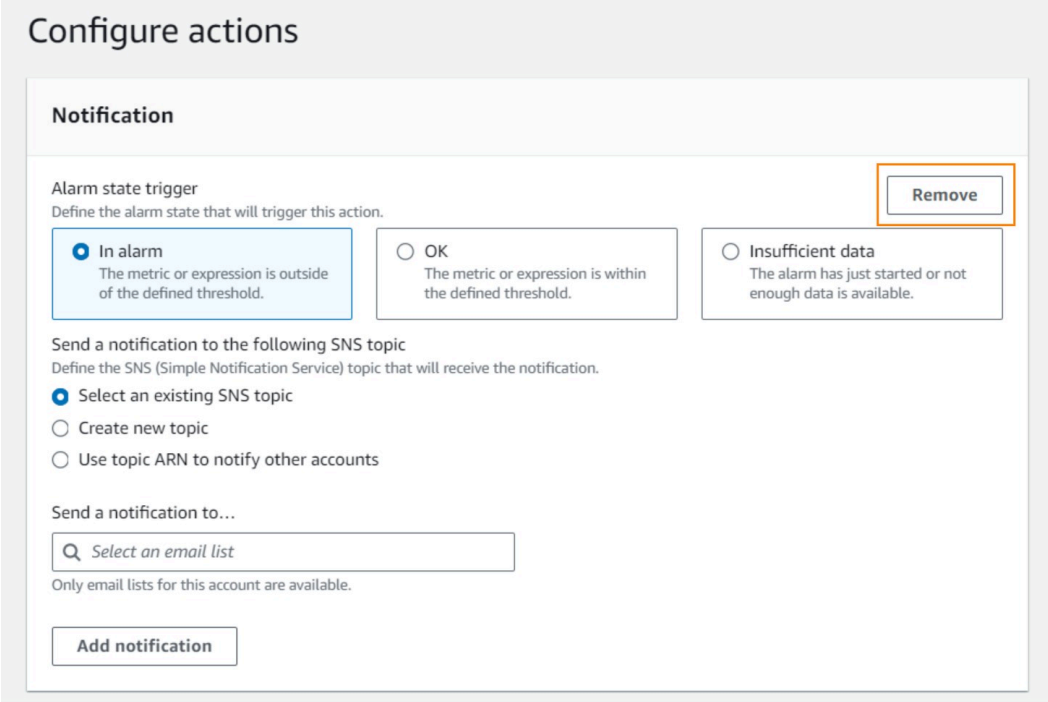
28. In the **Metric** menu, change the **Period** to **1 minute**.



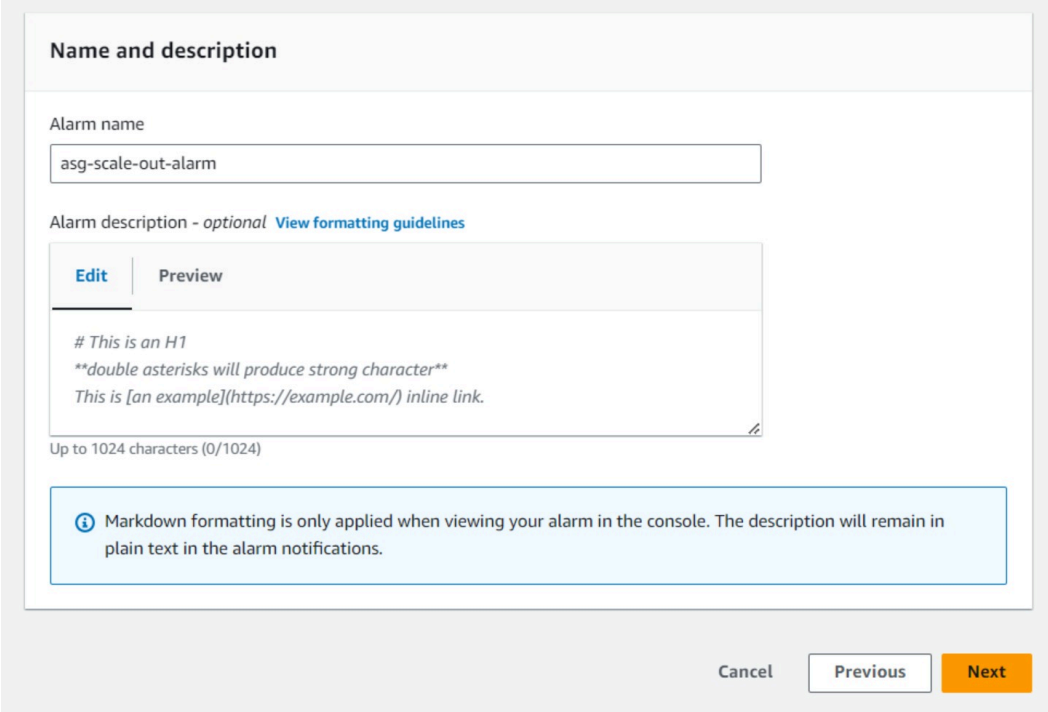
29. In the **Conditions** menu, fill out the fields with the following configuration values. Then, click **Next**.



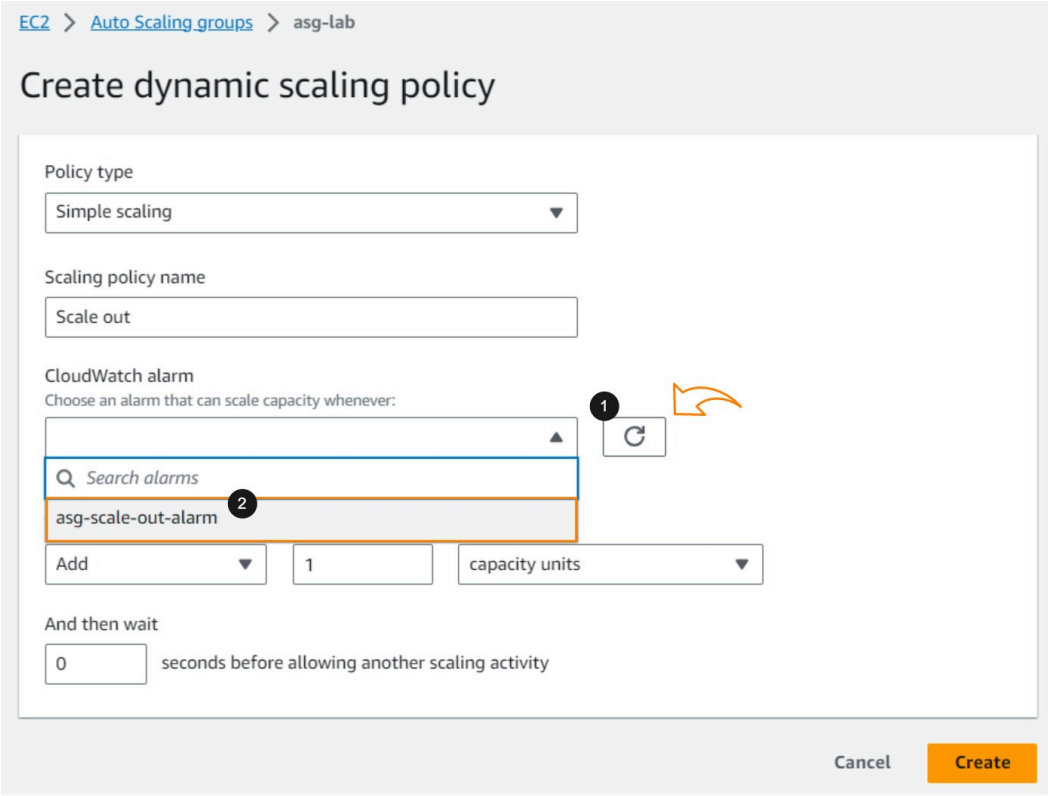
30. Remove **Notifications** from **Configure actions**. Then, scroll down and click **Next**.



31. Enter '*asg-scale-out-alarm*' for the Alarm name. Click "Next", then, click "Create alarm".

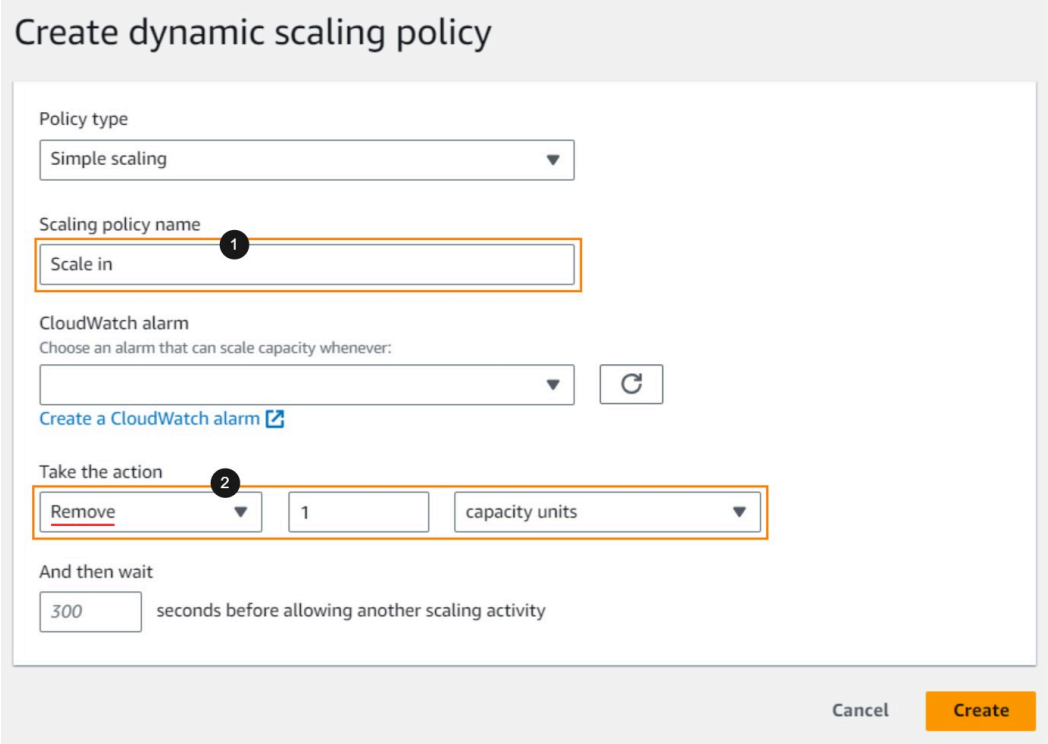


32. Go back to the ASG tab. Click the reload icon and select the newly created '*asg-scale-out-alarm*' from the dropdown. Then, click "Create".

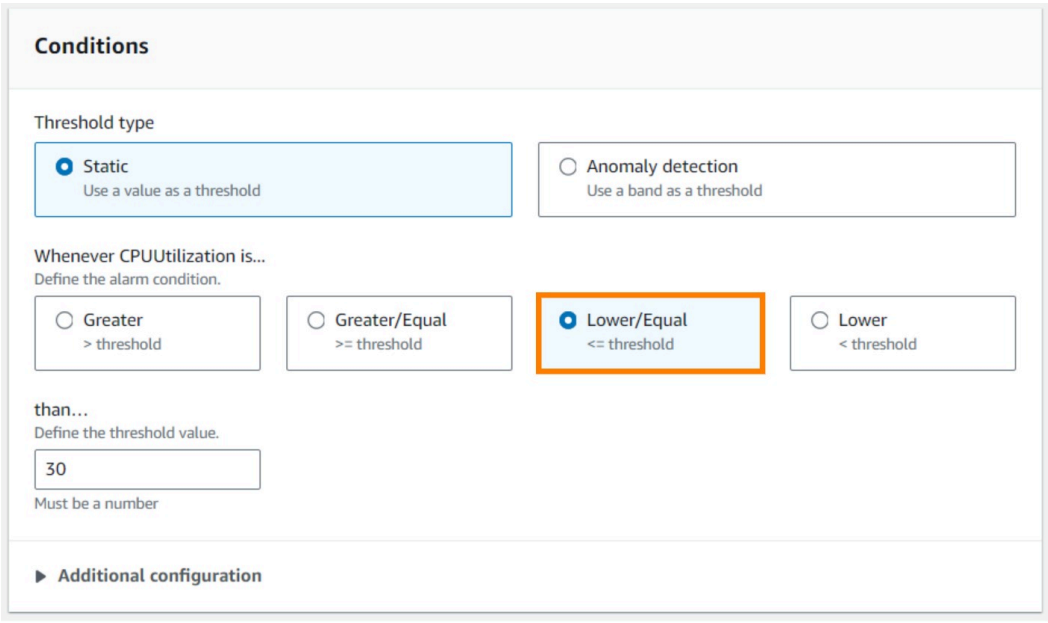


To create the scale in policy, follow the instructions between **Steps 22 – 32**, but **with the following changes**:

Scale in policy:



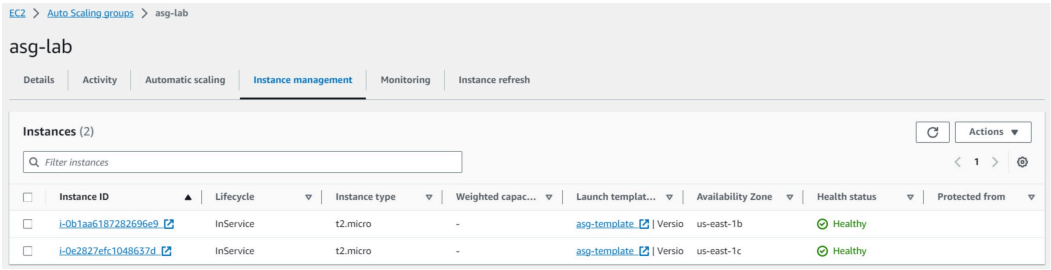
CloudWatch alarm condition:



For alarm name, enter ‘asg-scale-in-alarm’

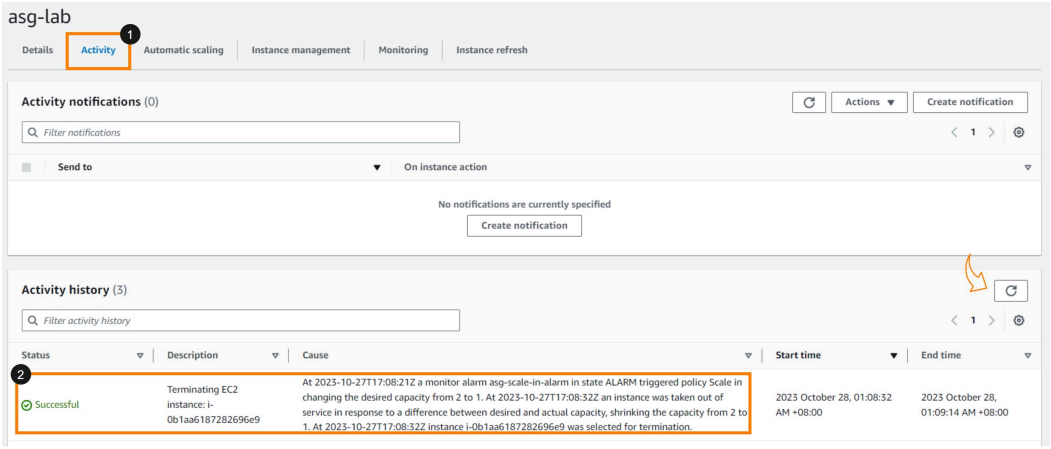
Testing the Scale in Policy

33. Go back to the ASG details page. Click on the **Instance Management** tab. This area provides insights into the status and health of your instances being managed by ASG. You’ll notice that ASG initially creates two instances. This is because we set the desired capacity to 2.



Since we've configured a scale-in policy that removes 1 instance when CPU utilization is below 30 percent within 1 minute, expect the instances to be reduced to 1 after a minute or so.

34. In the **Activity** tab, you'll see a notification indicating that the scale-in policy was triggered. Please be patient and wait for this notification to appear. To ensure you're seeing the most recent updates, consider clicking the refresh icon once in a while. You may also monitor the number of instances in your ASG under the **Instance Management** tab.



Now let's test our scale-out policy by simulating a high CPU load using a tool called '**stress**'.

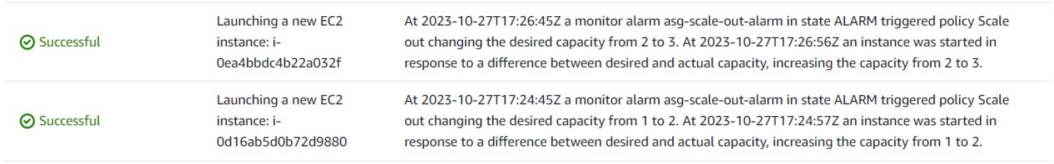
35. SSH into your EC2 instance and run the following commands.

```
#This updates the package list in your system
sudo yum update -y
```

```
#This installs stress
sudo yum install stress -y
```

```
#Spawns 50 workers for 5 minutes
stress --cpu 50 --timeout 5m
```

36. Monitor the **Activity** for scaling notifications or check the **Instance Management** tab to observe changes in the number of instances. Please be patient to wait, as the scaling won't happen instantaneously.



Successful	Launching a new EC2 instance: i-0ea4bbdc4b22a032f	At 2023-10-27T17:26:45Z a monitor alarm asg-scale-out-alarm in state ALARM triggered policy Scale out changing the desired capacity from 2 to 3. At 2023-10-27T17:26:56Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.
Successful	Launching a new EC2 instance: i-0d16ab5d0b72d9880	At 2023-10-27T17:24:45Z a monitor alarm asg-scale-out-alarm in state ALARM triggered policy Scale out changing the desired capacity from 1 to 2. At 2023-10-27T17:24:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.

Congratulations! You've successfully set up and experienced Auto Scaling in action. While we tackled a straightforward scaling scenario, remember there are more advanced scaling policies that cater to dynamic and complex workloads. This was just the starting point, and there's so much more to explore and learn. Keep up the momentum, and happy scaling!