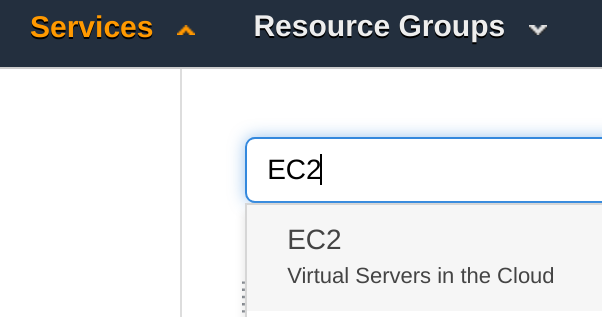
**Instructions**

1.  In the AWS Management Console, click **Services**, enter *EC2* in the search box, and click the **EC2** result:

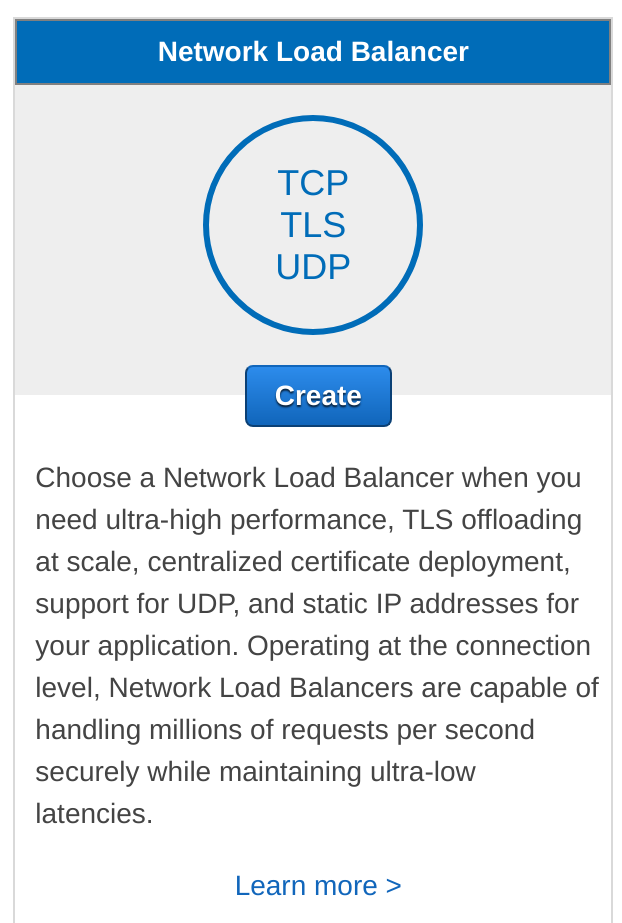


2. In the left-hand menu, click **Load Balancers**:

alt

3.  Click **Create Load Balancer**.

4.  Take a moment to read the information for the load balancer types before clicking **Create** in the **Network Load Balancer**tile:

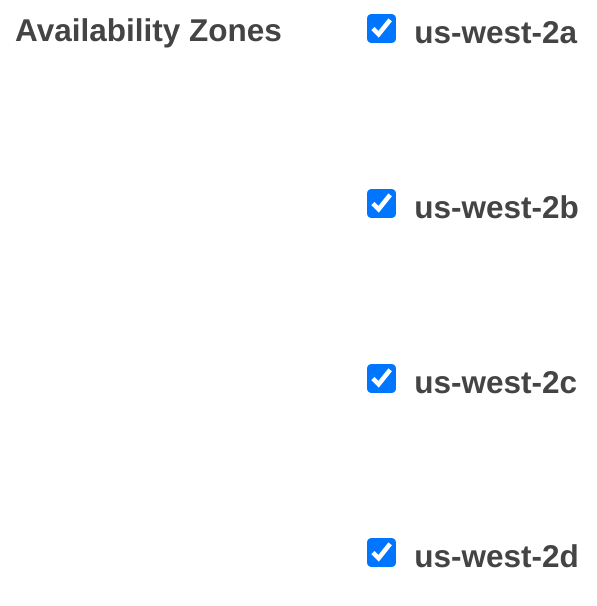


A multi-step wizard starts for creating a load balancer.

5. On the **Configure Load Balancer** step, set the following values leaving the others at their defaults:

* **Basic Configuration:**
  + **Name**: *Web*
* **Availability Zones**:
  + **Availability Zones**: Check all subnets





Notice that the default listener is TCP port 80 which is used for serving HTTP traffic.

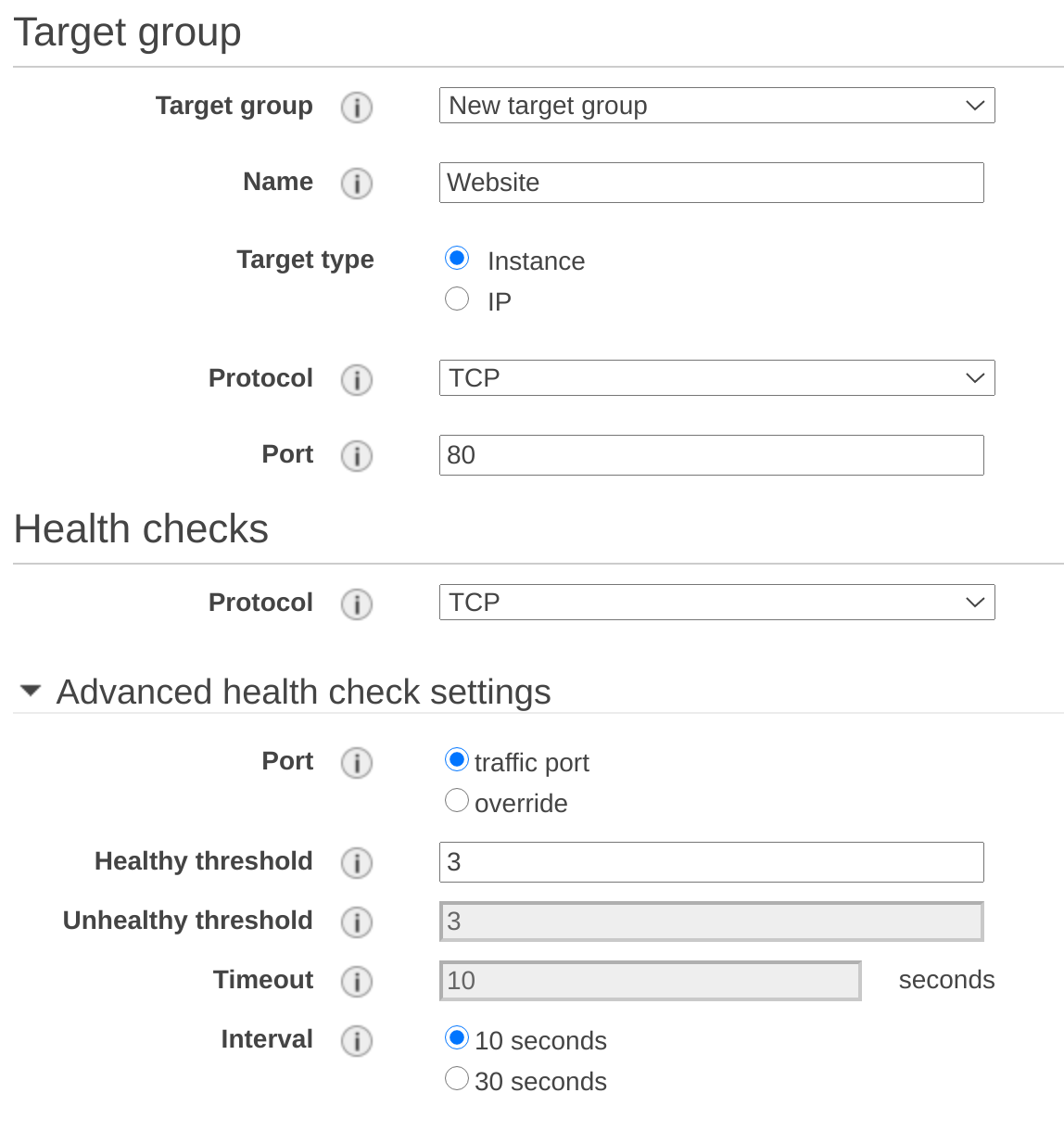
6. Click **Next: Configure Security Settings** when ready.

7. On the **Configure Security Settings** step, click**Next: Configure Routing**.

The warning message informs you that the listener isn't secure (not using TLS). You should carefully consider if you do not need TLS. For this Lab, it is not going to be an issue because there will be no sensitive information being transmitted.

8. On the **Configure Routing** step, set the following values leaving the others at their defaults:

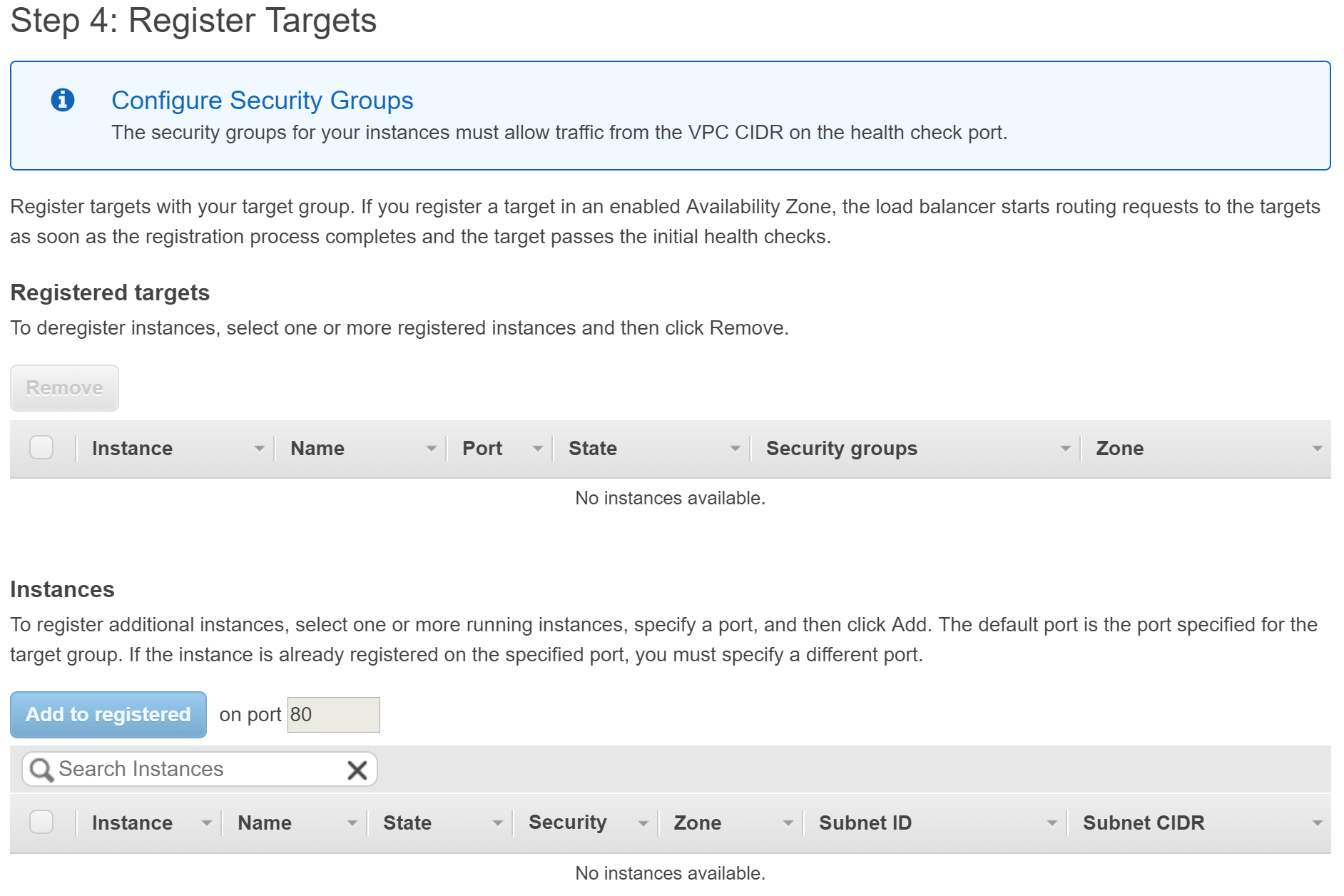
* **Target group:**
  + **Name**: *Website*
* **Health checks**:
  + **Advanced health check settings**: (Click the triangle to expand the section)
    - **Interval**: **10 seconds** (This will cause instances to reach a healthy state faster for this Lab, but may be too fast for certain applications)



**Target type**allows you to target **IP**addresses rather than instances. This gives you the ability to use the Network Load Balancer with instances outside of AWS.

9. Click **Next: Register Targets**.

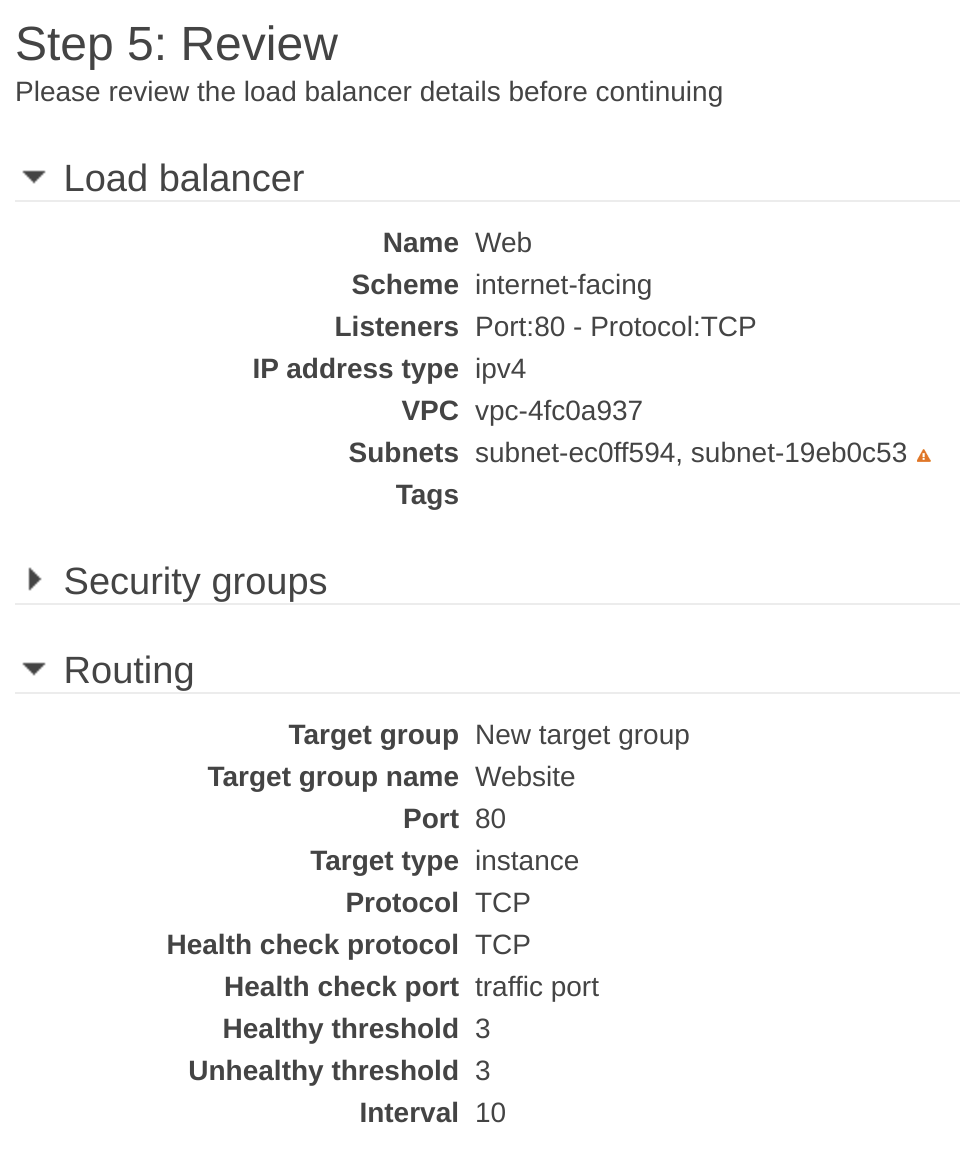
On the **Register Targets** step, notice there are **No instances available**:



The message is because you have not created an Auto Scaling group or launched any EC2 instances yet. That is not a problem. You will configure your Auto Scaling group to register its EC2 instances in the Network Load Balancer's target group.

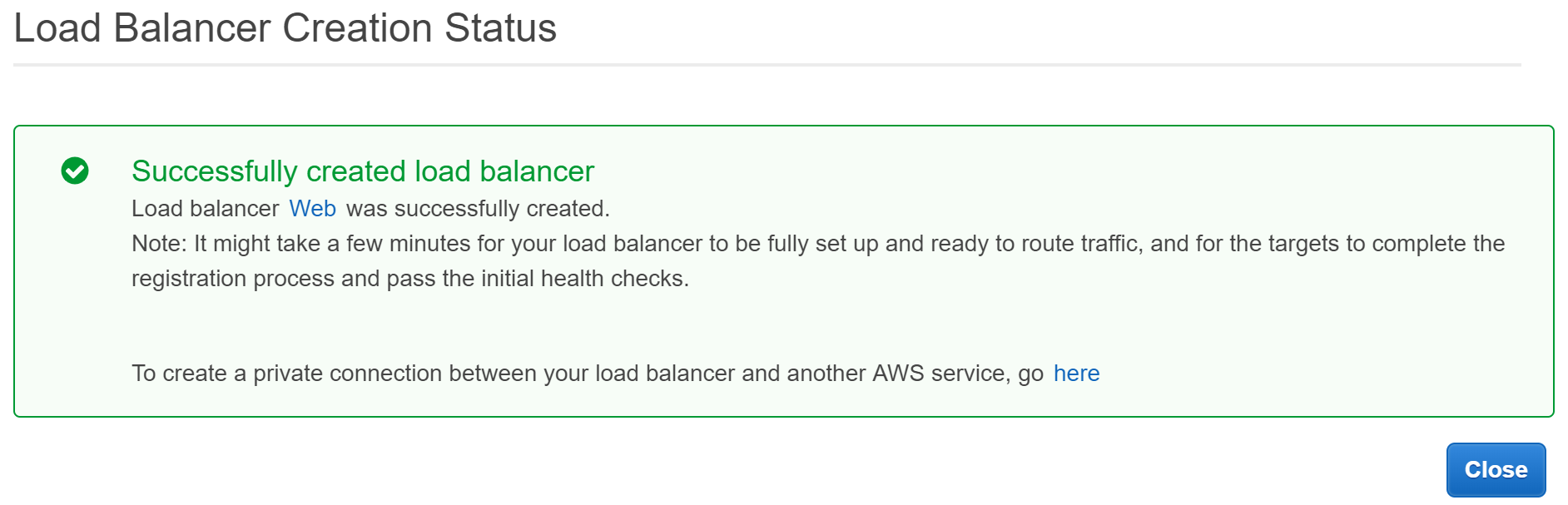
10. Click **Next: Review**.

11. Review your settings for correctness and then click **Create** when ready:

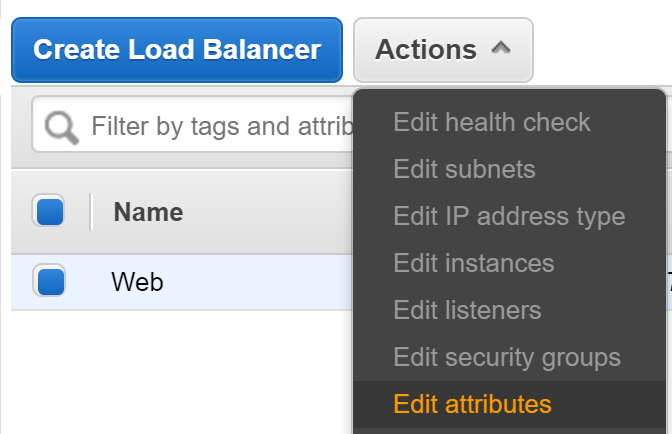


There is a warning icon by the **Subnets** value to inform you that there are no instances added from either Availabiliity Zone associated with the subnets. That is expected since no instances are registered as targets yet. When everything has been set up, it is a best practice to have instances in all availability zones for high availability.

12. Wait for the **Load Balancer Creation Status** success message to display before clicking **Close**:



13. In the load balancers table, ensure the **Web**load balancer is selected and click **Actions**>**Edit attributes**:



14. In the **Edit load balancer attributes** form, set the following value before clicking **Save**:

* **Cross-Zone Load Balancing**: **Enable**

alt

You must enable **Cross-Zone Load Balancing** to achieve the highest level of availability. Without enabling this feature, clients could cache the DNS address of the load balancer node in one availability zone and that node would only distribute requests to instances within the availability zone. Cross-Zone Load Balancing allows every load balancer node to distribute requests across all availability zones, although for the Network Load Balancer there are data transfer charges when this feature is enabled. (There are no data charges for other types of load balancers)

15. In the left-hand menu, click **Target Groups**:



16. Click on the **Website** target group.

17. Scroll to the **Attributes** section, and click on **Edit**:



17. Change the **Deregistration delay**to *30* seconds and click **Save changes**:

alt

The deregistration delay specifies how long the load balancer should wait before removing an instance from the target group. The default value of 300 seconds gives connections to the instance five minutes to drain before they are forcefully closed. Depending on your application, you may be able to reduce to delay to remove instances more quickly. Thirty seconds is enough for this Lab.

# Creating a Launch Template

**Instructions**

1. [Navigate to EC2 in the AWS Management Console](https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Home:).

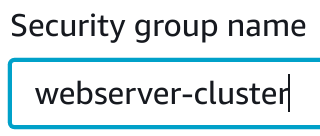
2. In the left-hand menu, under **Network & Security**, click **SecurityGroups**:



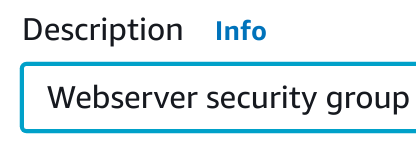
3. To start creating a new security group, click **Create security group**:



4. Under **Basic details**, in the **Security group name** field, enter w*ebserver-cluster*:



5. In the **Description** field, enter *Webserver security group*:

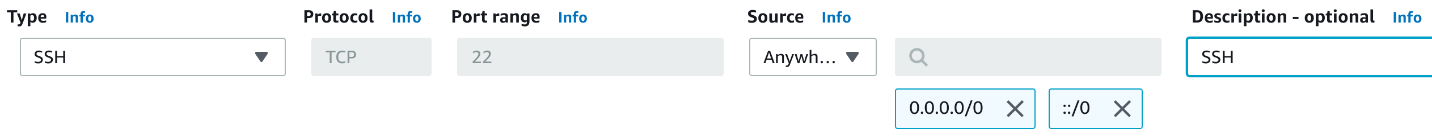


6. In the **Inbound rules** section, click **Add rule**:

New fields allowing you to specify a rule will appear.

7. To configure a rule allowing SSH traffic, enter the following values:

* **Type**: Enter *SSH* and select **SSH**
* **Source**: Select **Anywhere**
* **Description**: Enter *SSH*

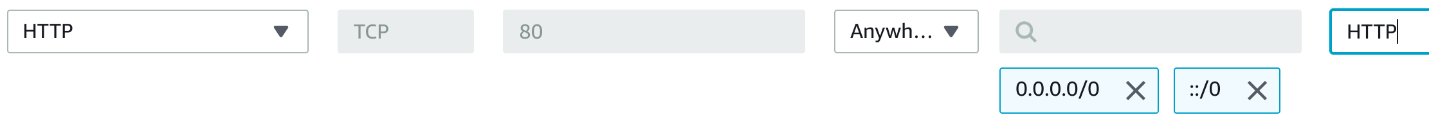


You have added this rule so that later you can access instances using SSH.

8. Click **Add rule** again.

9. To configure a rule allowing HTTP traffic, enter the following values:

* **Type**: Enter *HTTP* and select **HTTP**
* **Source**: Select **Anywhere**
* **Description**: Enter *HTTP*



You have added two inbound rules.

In this lab, you are allowing access from anywhere (**0.0.0.0/0**). In a non-lab environment, you are likely to be required to have a much more restrictive **Source**. For example, you may be required to specify a corporate IP range to reduce the likelihood of unauthorized access.

10. To finish creating your security group, scroll to the bottom of the page and click **Create security group**:



You will see a notification that your security group has been created:



You have created a security group which you can specify in the launch template you are about to create.

11. In the left-hand menu, under **Instances**, click **Launch Templates**:



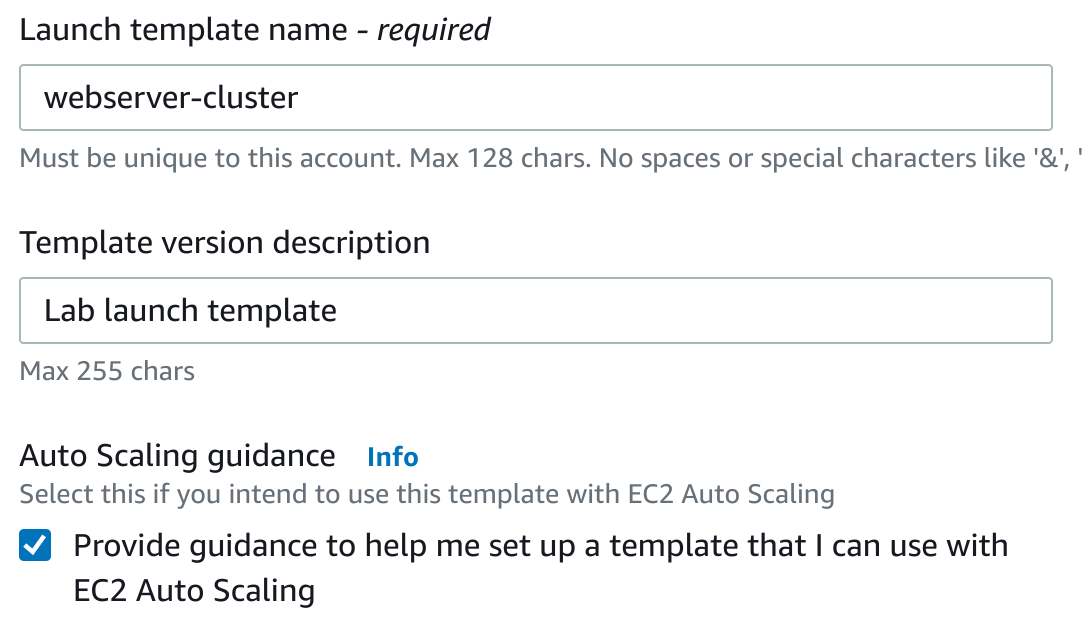
12. To open the **Launch Templates** page and click the **Create launch template** button:



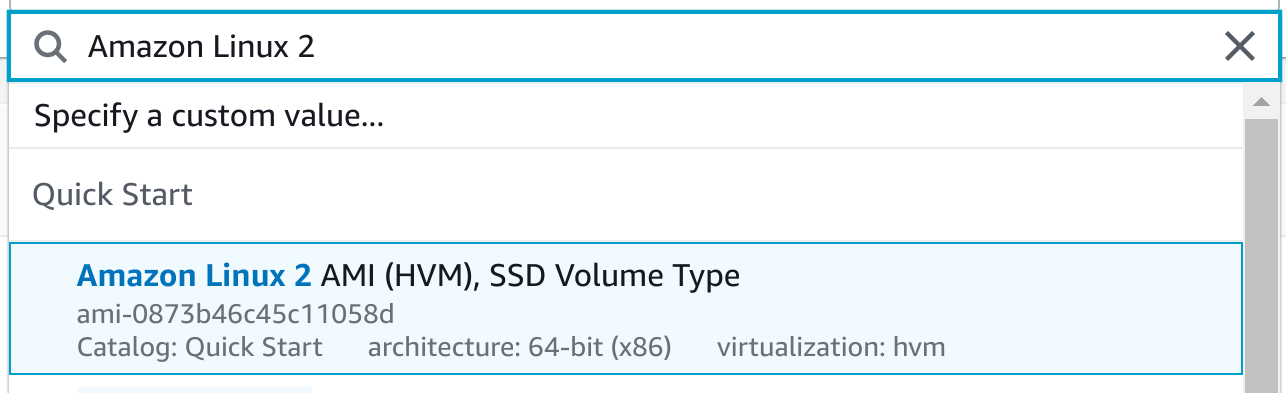
The create launch template page will load.

13. In the **Launch template name and description** section, enter the following values accepting the defaults for fields not specified:

* **Launch template name***: webserver-cluster*
* **Template version description**: *Lab launch template*
* **Provide guidance to help me set up a template that I can use with EC2 Auto Scaling**: *checked* (this setting makes some fields required)

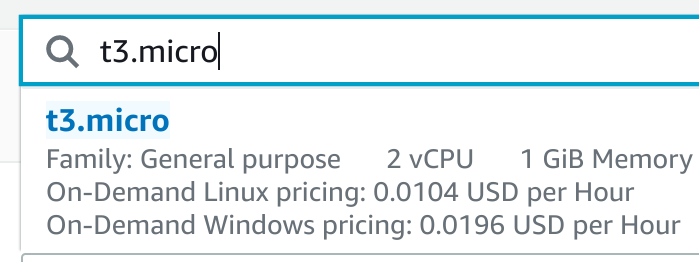


14. Scroll down to the **AMI** field, enter *Amazon Linux 2* and click the top result:



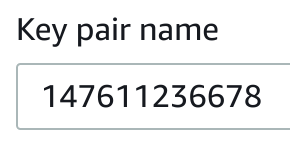
You have selected an Amazon Linux 2 AMI using its id.

15. In the **Instance type** field, enter *t3.micro*and click the **t3.micro** result:



Be aware that in this lab you are restricted from using larger instances, trying to launch other types of instance may result in your CloudAcademy account being temporarily banned.

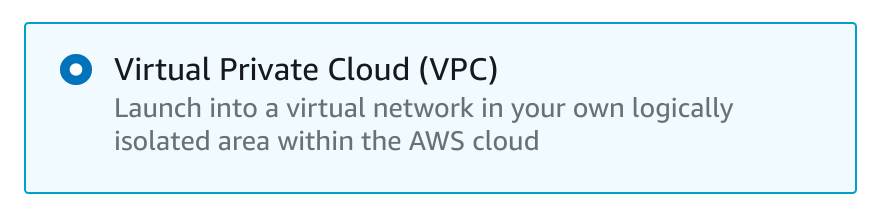
16. In the **Key pair (login)** section, under **Key pair name**, select the numeric option:



You will use the key pair to access EC2 instances launched with this configuration later in the lab.

In situations where you don't need SSH, you don't have to configure a key pair, this is often preferred in terms of security.

17. Scroll down to **Network Settings** and ensure **Virtual Private Cloud (VPC)** is selected:



18. In the **Security groups** drop-down, select **webserver-cluster**:



This is the security group you created earlier.

Notice you have created separate security groups for the instances and the load balancer, in a non-lab environment it is often the case that the instances are listening on a different port than the load balancer, and the load balancer re-directs the traffic. Assigning separate security groups allows you to more flexibly control and restrict traffic at the network level.

19. Take a look at the **Storage (volumes)** section.

This part of the form allows you to add or increment the size of any EBS volume attached to each EC2 instance started by the Auto Scaling group. Leave the defaults and do not add any EBS volumes.

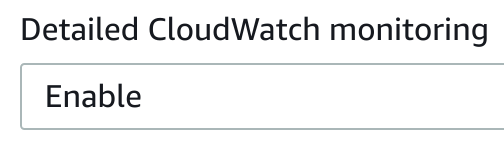
Typically large EBS volumes are only needed if your software requires storage space to process the application data. Many applications store raw or processed data with Amazon S3, Redshift, DynamoDB, or another storage/database service provided by Amazon. When that is the use-case, large EBS volumes are usually not required. This lab environment does not need extra disk space.

20. Scroll down to the bottom and click **Advanced details**:



More configuration options will appear.

21. Scroll down to the **Detailed CloudWatch monitoring** option, and click **Enable**:



By default, CloudWatch monitors EC2 instances approximately every 5 minutes. Detailed monitoring enables monitoring more often (each minute).

*Note*: Enabling detailed monitoring incurs an extra cost.

22. In the **User data** text-box at the bottom of the page, enter the following script:

1

2

3

4

5

6

7

8

9

10

11

*#!/bin/bash*

*#Enable the epel-release*

sudo amazon-linux-extras install epel

*#Install and start Apache web server*

sudo yum install -y httpd php

service httpd start

*#Install CPU stress test tool*

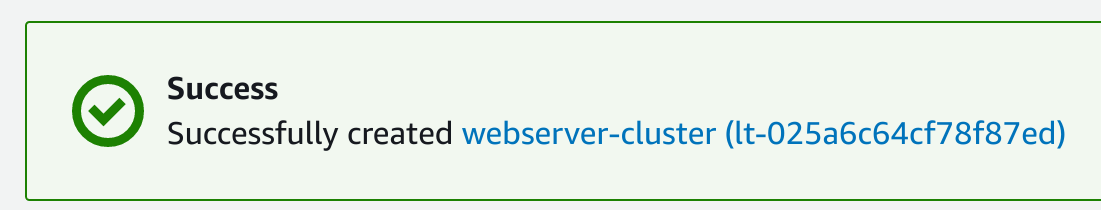
sudo yum install -y stress

This bash script installs PHP, an Apache web server (httpd), and a tool for stress testing called Stress.

23. To create your Launch Template, click **Create launch template**:



You will see a notification that your launch template has been created:



24. To return to the EC2 management console, click **View launch templates**:



# Creating an Auto Scaling Group from a Launch Template

**Instructions**

1. [Navigate to the EC2 service in the AWS Management Console](https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Home:).

2. In the left-hand menu, click **Auto Scaling Groups**:

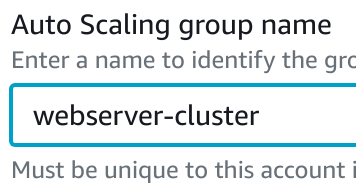
alt

3. To start creating an auto-scaling group, click **Create an Auto Scaling group**:

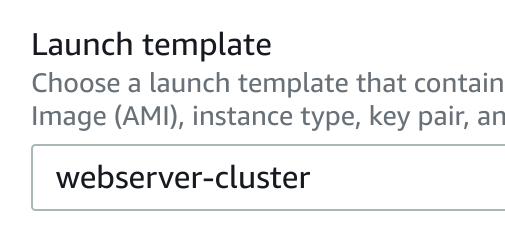


A multi-step wizard will start.

4. In the **Auto Scaling group name** text-box, enter *webserver-cluster*:



5. In the **Launch template** field, select **webserver-cluster**:



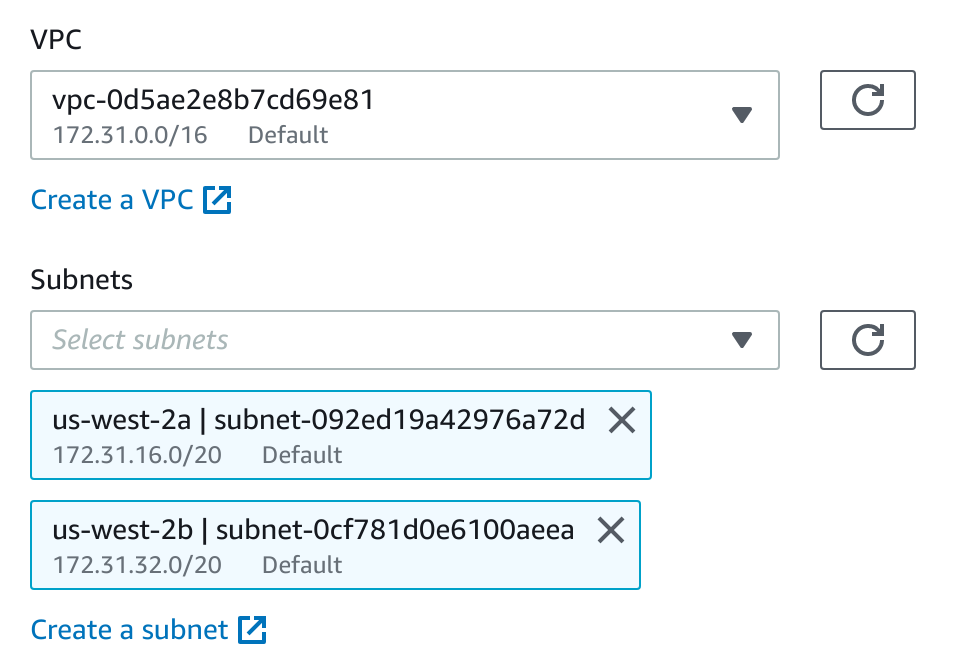
This is the launch template you created previously.

6. To advance to the next page of the wizard, click **Next**:



7. To configure your auto-scaling group, enter the following values:

* **VPC**: Select the only option available
* **Subnets**: Select all available

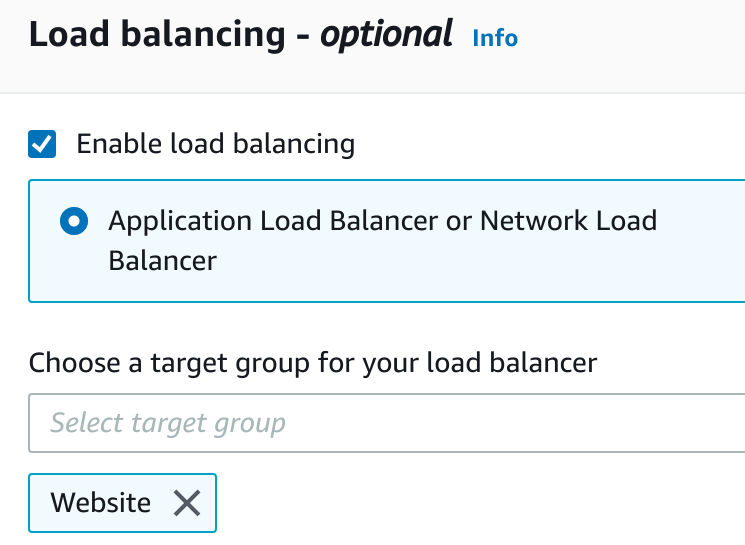


8. To advance to the next page of the wizard, click **Next**:

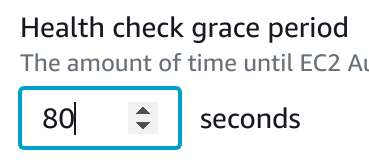


9. Under the **Load balancing** section, enter the following leaving unspecified fields at their defaults:

* **Enable load balancing**: Checked
* **Choose a target group for your load balancer**: Select **Website**



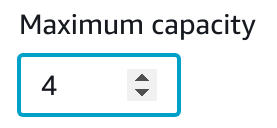
10. Under **Health check grace period**, enter *80*:



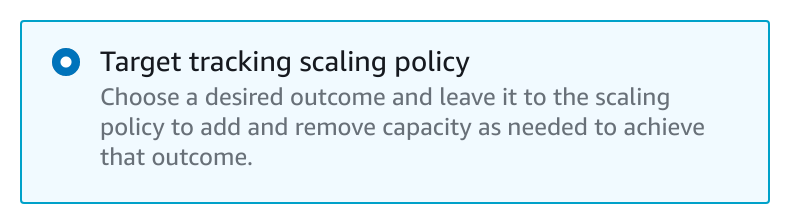
11. To advance to the next page of the wizard, click **Next**:



12. In the **Group size** section, in the **Maximum capacity** field, enter *4*:



13. Under **Scaling policies**, select **Target tracking scaling policy**:



Take a look at the available options.

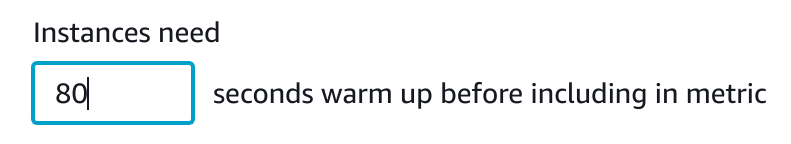
Auto-scaling groups allow you to scale out (add more instances) or scale in (remove instances) based on metrics such as:

* Average CPU utilization
* Network traffic (ingress and egress)
* Application Load Balancer request counts

It's possible to create your own metrics and use these to decide when to scale, such metrics can be populated in CloudWatch directly from the application rather than inspecting the instance's resource usage. This is useful when the point that you need to scale at is determined by something unrelated to the instance, such as reaching a database connection limit or some other application-specific bottleneck.

By default, the **Metric type** is **Average CPU utilization**, and the **Target value** is 50 percent. Leave these values unchanged.

14. In the **Instances need** text-box, enter *80*:



 15. To advance to the next page of the wizard, click **Next**:



You will see a wizard step that allows you to configure notifications when scaling events occur. Notifications aren't used in this lab.

16. To advance to the **Add tags** page of the wizard, click **Next**.

In a non-lab environment, it is best practice to tag resources when you create them so they can be easily filtered and discovered. Tags are not required in this lab.

17. To advance to the **Review** page of the wizard, click **Next**.

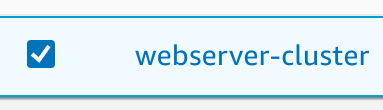
18. Check the details for accuracy and when ready, click **Create Auto Scaling group**:



You will be taken to the auto-scaling group list page and you will see a notification that your group has been created:



19. To see details about your auto-scaling group, check the box next to **webserver-cluster**:

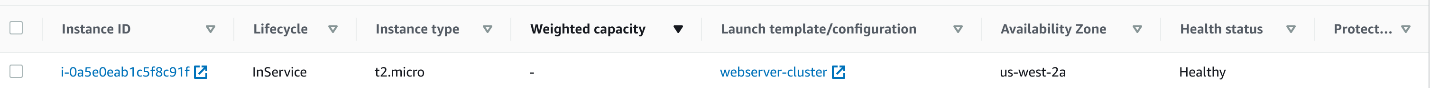


You will see some tabs appear under the list of groups.

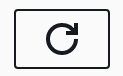
20. To view instance information, click the **Instance management** tab:



You will see an instance:



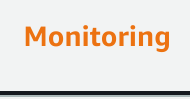
If you don't see an instance that has a **LifeCycle**value of **InService**, wait a minute or two, and click the refresh button:



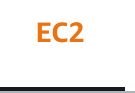
The auto-scaling group has started an instance, this is because the minimum capacity of the group is 1.

Because the webserver is not CPU-intensive and there is no load on the webserver, the high CPU alarm won't trigger. The number of instances will stay at 1 unless CPU utilization on the existing instance increases above 50 percent.

21. To see monitoring information about your auto-scaling group, click the **Monitoring** tab:

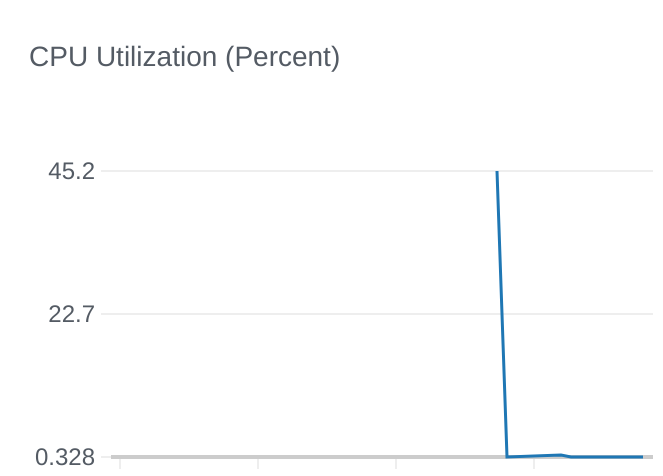


22. Under **CloudWatch monitoring details**, click the **EC2** tab:



Take a look at the different metrics recorded.

Focus on the **CPU Utilization (Percent)** graph:



Please note it may take several minutes for data points to display. Click the refresh button periodically to update the chart.

You will see the CPU utilization is currently near zero. You may see a higher value in the past, this spike in CPU utilization occurred when the instance was started and executed the commands you specified in the user data of your launch template.

You configured the auto-scaling group to have a minimum number of instances of one. This means that even though the CPU utilization is currently below 50 percent, the auto-scaling group is keeping this instance running instead of terminating it.

23. In the left-hand side menu, under **LOAD BALANCING**, click **Target Groups**:



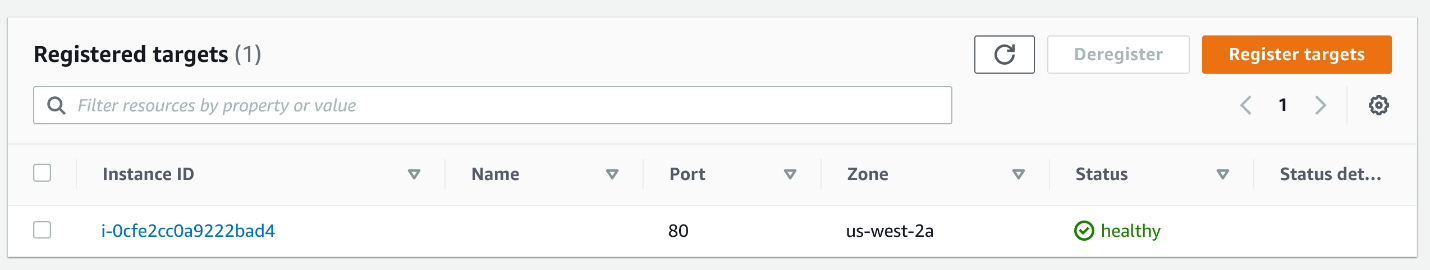
You will see the target group list page with one item, the **Website** target group you created earlier.

24. Select the **Website** target group, and then click the **Targets** tab:



Note you may need to click the refresh button to update the table if the instance is in the **initial** status while the load balancer waits for three successful health checks before assigning a healthy status.

Observe there is an instance added to the **Registered targets** and it is the same instance created by the Auto Scaling group. Also, notice the **Status** is **healthy** meaning the instance is reachable on TCP port 80 (HTTP). That means the launch template's user data script successfully completed to start the Apache webserver on the instance. Everything appears to be working. You will perform more thorough tests in the next Lab Step.



If the instance doesn't become healthy after five minutes, a likely reason is the user data being absent or incorrect. Check the user data field on your Lanuch Template, if necessary, you can delete and re-create the Launch Template and Auto Scaling group.

# Testing the Auto Scaling Group from End-to-End

**Instructions**

1. In the left-hand menu, under **Load Balancing**, click **Load Balancers**:

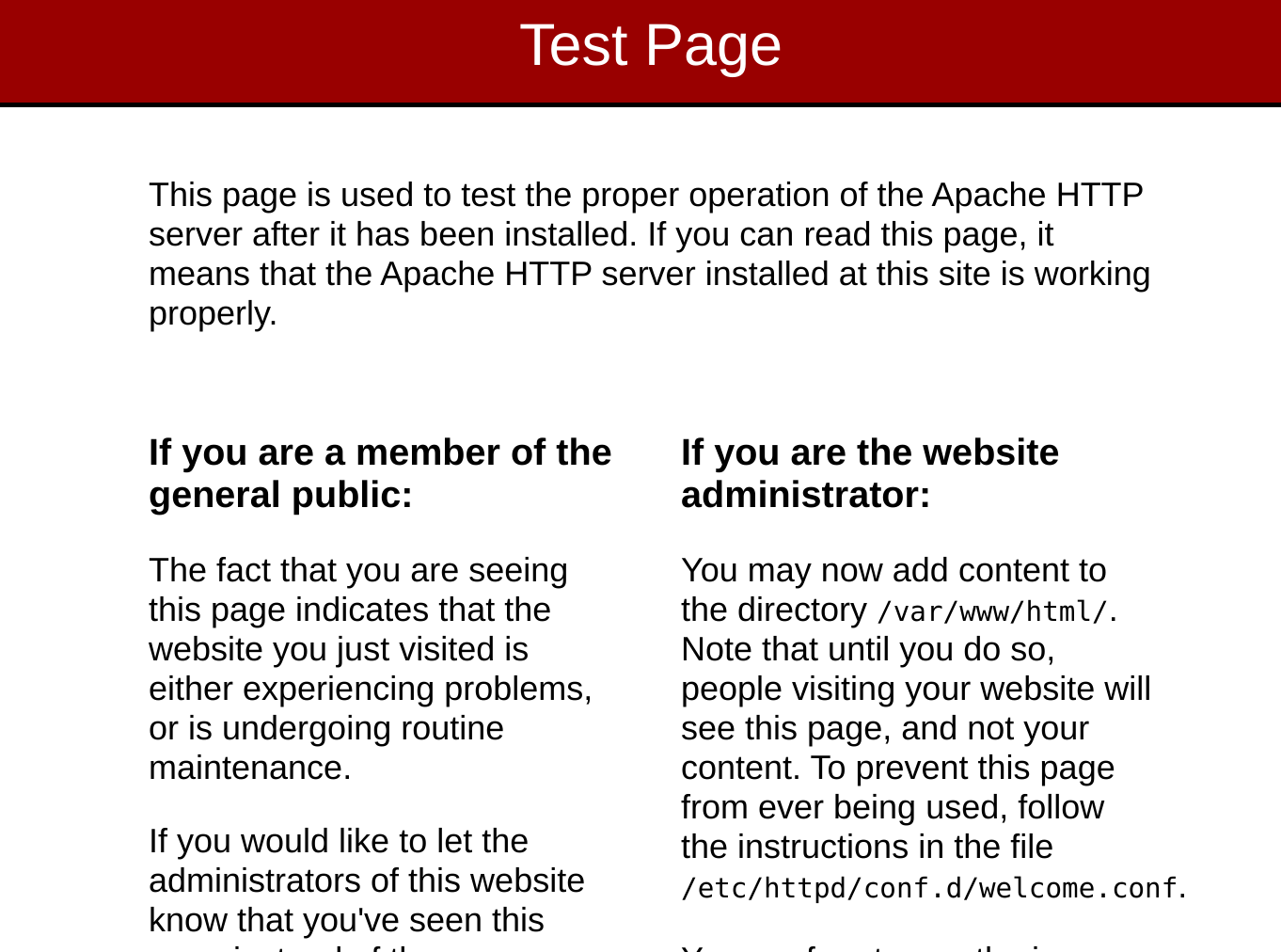


You will see a list of load balancers with one item selected, the load balancer you created previously.

2. In the **Description** tab, look for the **DNS name** field, and copy the value (you can click the copy button at the end of the field).

3. Open a new browser tab and paste the DNS name you just copied into the address bar.

You will see the Apache webserver test page that is served by default upon a fresh unconfigured web server installation:



If the page is not displayed, there are several places you can check to troubleshoot the issue starting with the following:

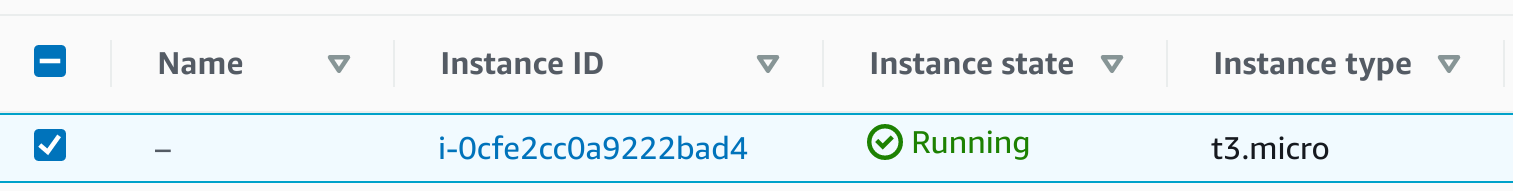
* Ensure the security groups of the load balancer and the instances allows HTTP ingress traffic
* Ensure the user data script in the launch template correctly installs and runs the Apache webserver
* Ensure the Auto Scaling group is configured to add its instances to the load balancer's target group
* Ensure the health checks are configured for TCP port 80 otherwise, the instances will never reach a healthy status and will be terminated and then replaced with a new instance by the Auto Scaling group. The new instance will subsequently never reach a healthy status and be replaced, and the process repeats.
  + To allow for you to debug the instances without having them be replaced, you can block instance termination by performing the following steps:
    - Navigate to **Auto Scaling Groups**> **Actions** >**Edit**
    - Set the **Suspended Processes** to **Terminate** (This will prevent instances in your group from getting terminated.Don't forget to remove the configuration once the issue is resolved.)

4. To list EC2 instances, in the left-hand menu, click **Instances**:

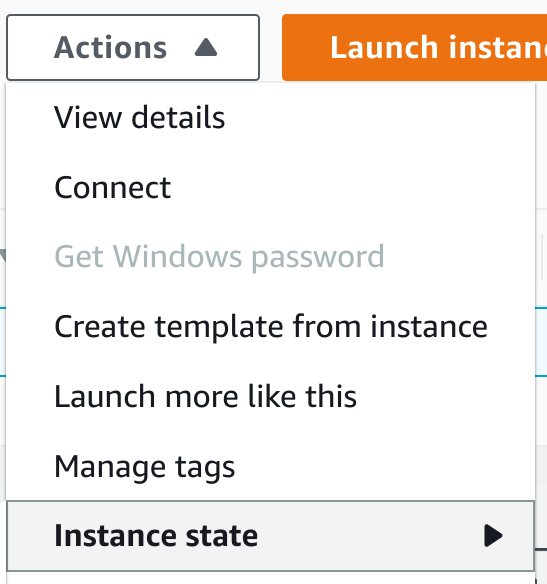


You will see a list of instances with one running instance, this instance is created by your auto-scaling group.

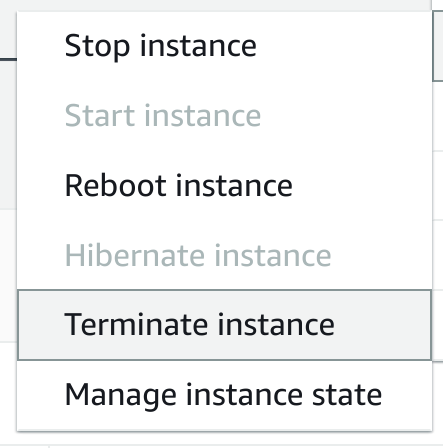
5. To select your instance, click the check-box on the left-hand side:



6.  To terminate the running instance, click **Actions**>**Instance State**

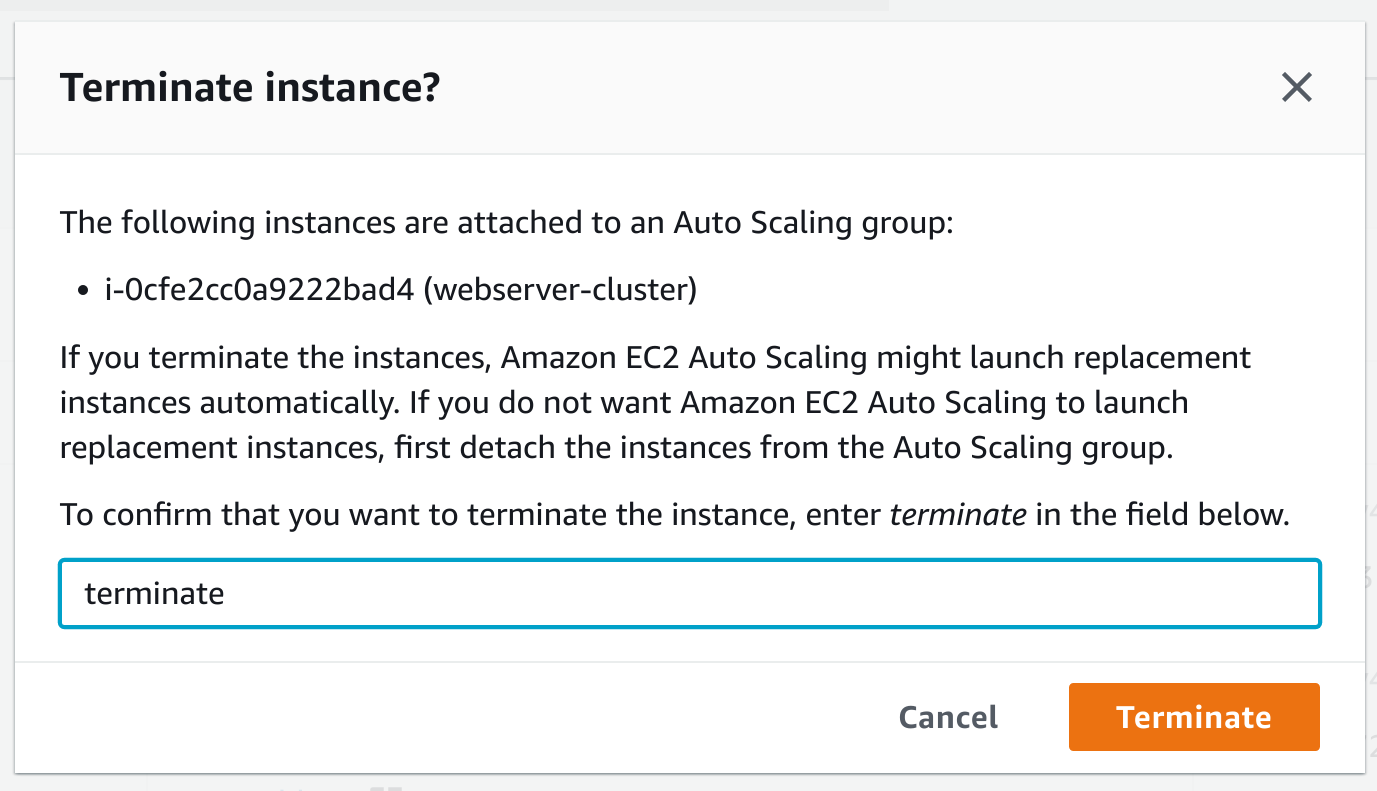


7. In the menu that opens, select **Terminate instance**:



The **Terminate instance?** dialog will open.

8. Enter *terminate* in the confirmation dialog text-box, and click **Terminate**:

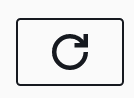


In the instances list, you will see the **Instance state** change from **Running** to **Shutting-down** and then **Terminated**.

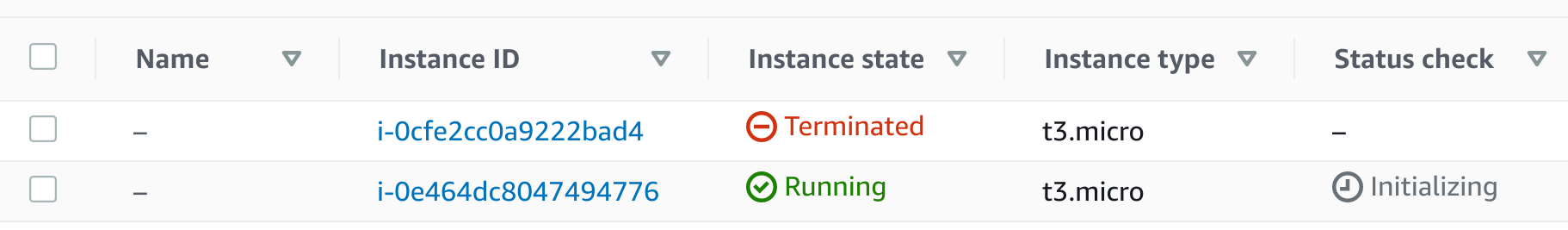
You have terminated the running instance.

The Auto Scaling group will detect this change and relaunch an instance automatically to meet the minimum desired capacity of one.

9. Wait two or three minutes and click the refresh icon:

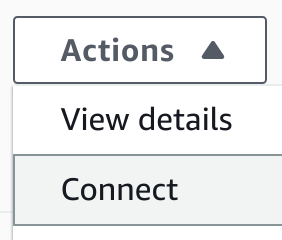


You will see the new instance launch and settle into a running state:



10. Connect to the running instance using the PEM (macOS/Linux) or PPK (Windows) key file in the **Your lab data** of this Lab.

*Tip*: To remind yourself how to connect, select the running instance, click **Actions**, and click **Connect**:



CloudAcademy recommends connecting using EC2 Instance Connect. You can also connect using an SSH client and PEM file in the credentials section of this lab. Connecting using session manager is not supported in this lab.

11. Enter the following command at the command line to run stress causing the CPU utilization to increase for five minutes:

1

*stress* --*cpu* *2* --*io* *1* --*vm* *1* --*vm*-*bytes* *128M* --*timeout* *5m*

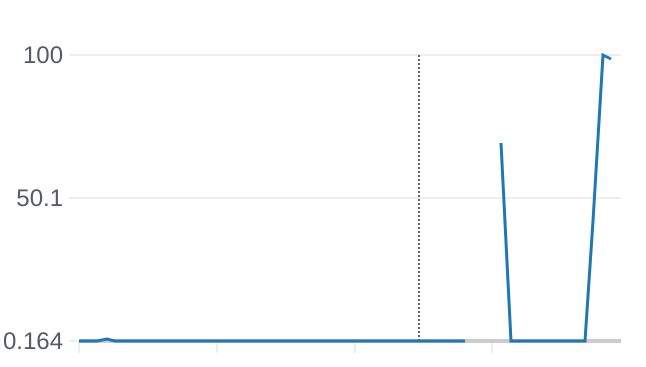
alt

 Enter man stress for more information about stress.

12. [Navigate to the Auto Scaling group's Monitoring tab in the EC2 Console](https://us-west-2.console.aws.amazon.com/ec2autoscaling/home?region=us-west-2#/details?id=webserver-cluster&view=monitoring) and click the **EC2** tab.

Wait a minute or two and click the refresh button.

You will see the CPU utilization metric increase to near one hundred percent:

[](https://us-west-2.console.aws.amazon.com/ec2autoscaling/home?region=us-west-2#/details?id=webserver-cluster&view=monitoring)

The auto-scaling group will detect the increase in utilization and launch a second instance in response.

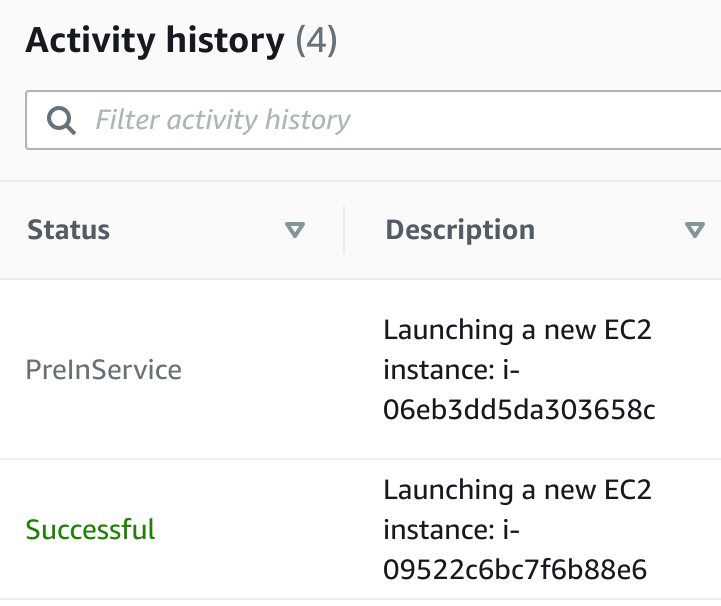
Note that it will take a few minutes the metrics and graph to update.

13. To see auto-scaling events, click the **Activity** tab:



Look at the **Activity history** section.

The exact events you will see may differ depending on how long the auto-scaling group and stress command have been running:



You will see events detailing your earlier deletion of the initial instance.

If you don't see a new instance launching wait a minute or two and click the refresh button.

Eventually, because the stress command stops after five minutes, the number of instances will drop down to the minimum of one.

14. [Navigate to the Instances section of the EC2 Console](https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:sort=instanceId).

You will see two running instances, or if your auto-scaling group has already initiated a scale-in (because the stress command finished and CPU utilization dropped), you will see two terminated instances including the one you manually terminated, and one running instance.