

LAB: Automatically Scale Instances with Load Balancing

You need:

- An AWS Account

Duration of the Lab: 30 Minutes.

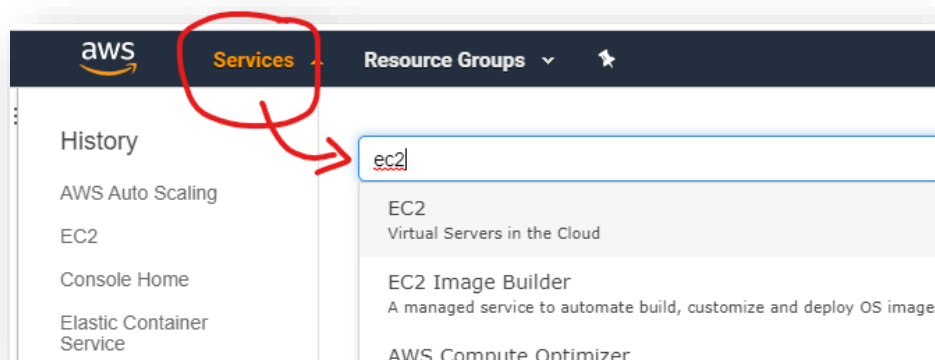
Difficulty: medium

Contents

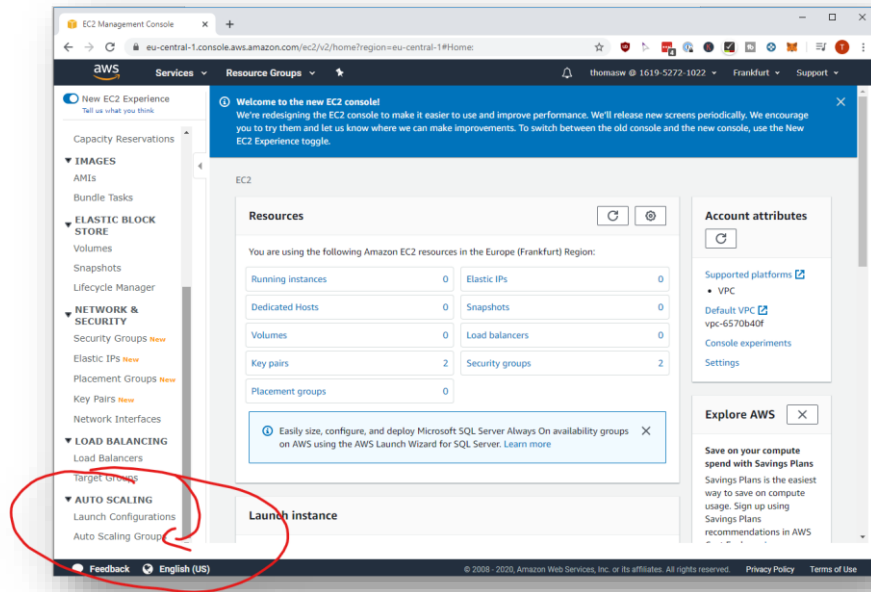
Create an Auto Scaling Group	1
Create an Instance Launch Template	3
Select Placement	8
Enable Load Balancing.....	9
Create a Target Group.....	9
Set the Group Size and Scaling Policies	11
Verify the Instances are Running	13
Troubleshooting	14
Create a Load Balancer.....	15
Testing the Load Balancer	17
Tear Down and Clean Up.....	18

Create an Auto Scaling Group

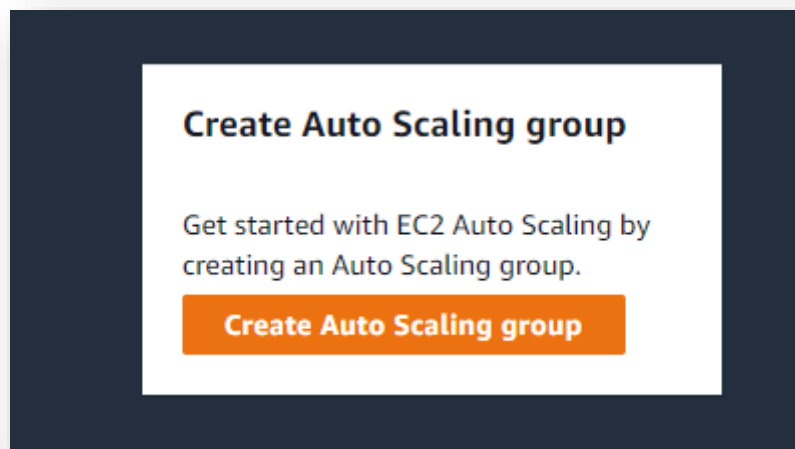
Open your EC2 Dashboard



On the bottom of the list, select “Auto Scaling Group”:



Click on Create Auto Scaling Group



To create an Autoscaling Group you need a launch configuration. This tells the Auto Scaling group how to launch a new instance when it needs to scale out. Give the Autoscaling group a name and then create a launch template:

Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.

My Autoscaling Group for Course

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info[Switch to launch configuration](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template ▼

[Create a launch template](#)

Cancel

Next

Launch template Info

[Switch to launch configuration](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template ▼

[Create a launch template](#)

Cancel

Next

Create an Instance Launch Template

A launch template is a blueprint for AWS how to start a new Instance. Give the Launch Template a name:

EC2 > Launch templates > Create launch template

Create launch template

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

Launch template name and description

Launch template name - *required*

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

Template version description

Max 255 chars

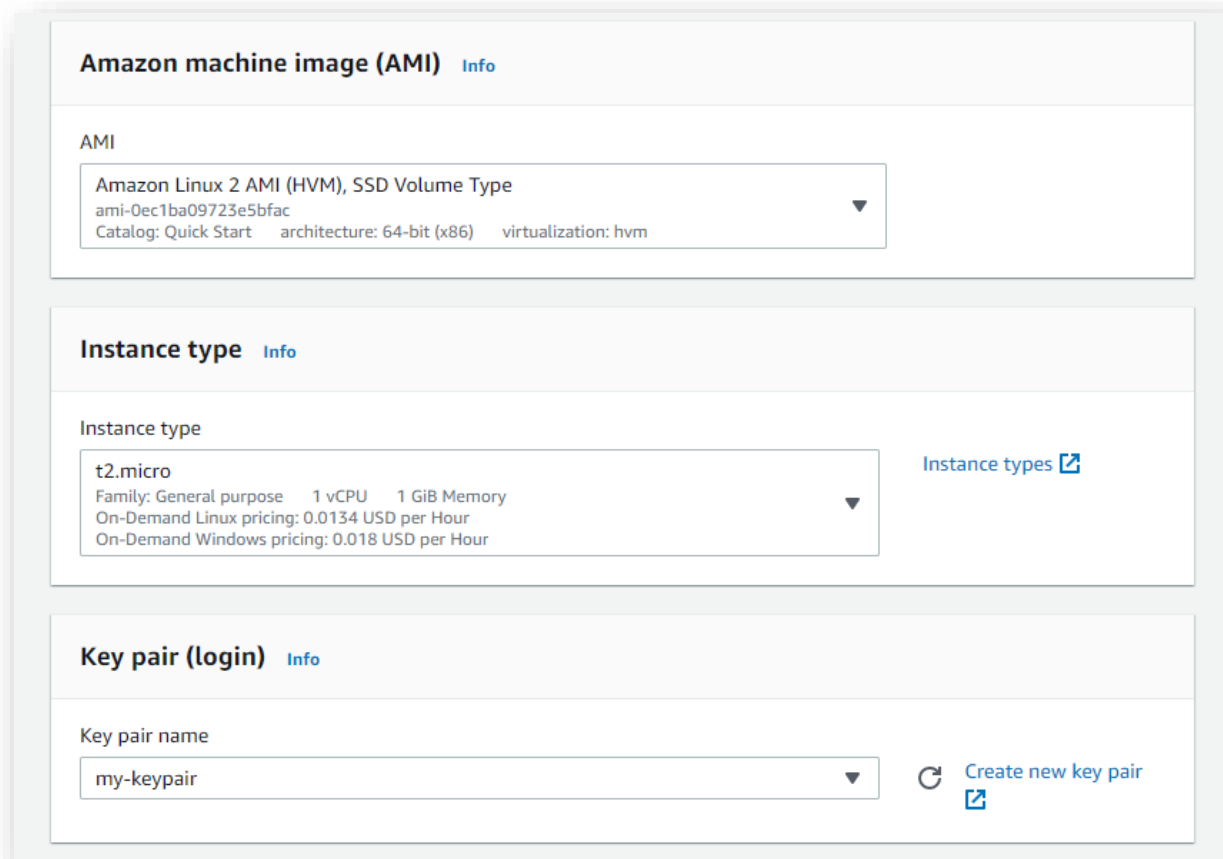
Auto Scaling guidance [Info](#)
Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags

► Source template

Select an AMI and an Instance Type. As well as a key-pair if you want to be able to login:



Amazon machine image (AMI) [Info](#)

AMI

Amazon Linux 2 AMI (HVM), SSD Volume Type
ami-0ec1ba09723e5bfac
Catalog: Quick Start architecture: 64-bit (x86) virtualization: hvm

Instance type [Info](#)

Instance type

t2.micro
Family: General purpose 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0134 USD per Hour
On-Demand Windows pricing: 0.018 USD per Hour

[Instance types](#)

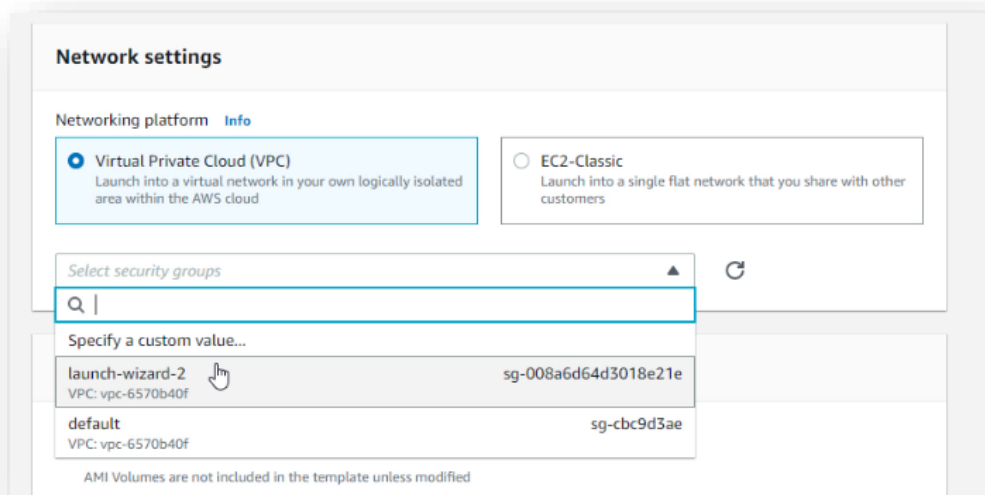
Key pair (login) [Info](#)

Key pair name

my-keypair

[Create new key pair](#)

Select the Security group from the previous lectures where port 80 is open:



Network settings

Networking platform [Info](#)

☒ Virtual Private Cloud (VPC)
Launch into a virtual network in your own logically isolated area within the AWS cloud

☐ EC2-Classic
Launch into a single flat network that you share with other customers

Select security groups

Q |

Specify a custom value...

launch-wizard-2 sg-008a6d64d3018e21e
VPC: vpc-6570b40f

default sg-cbc9d3ae
VPC: vpc-6570b40f

AMI Volumes are not included in the template unless modified

Under Advanced Setting add some user data to automatically install an apache + php upon launching the instance:

Advanced details Info

Cancel Create launch template

Metadata response hop limit Info

Don't include in launch template

User data Info

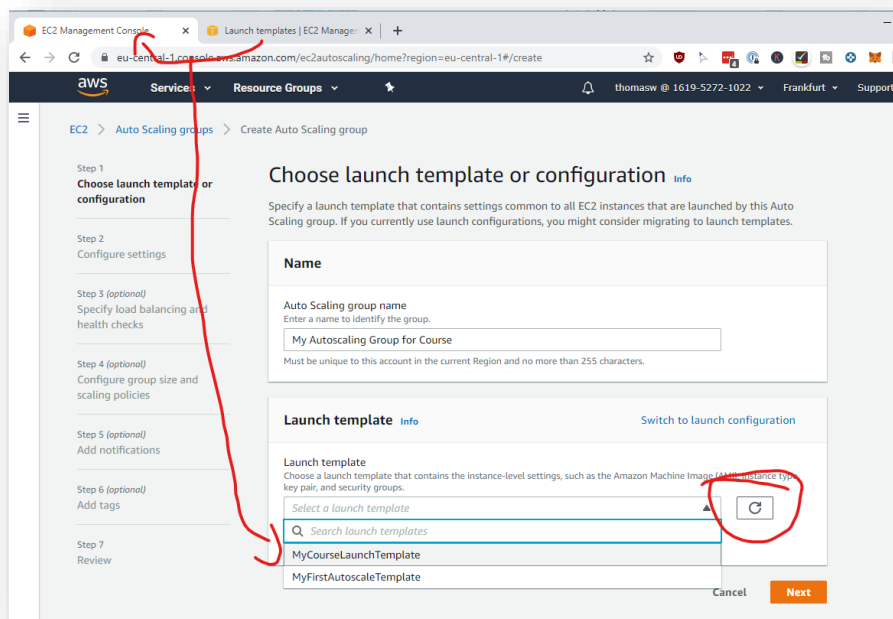
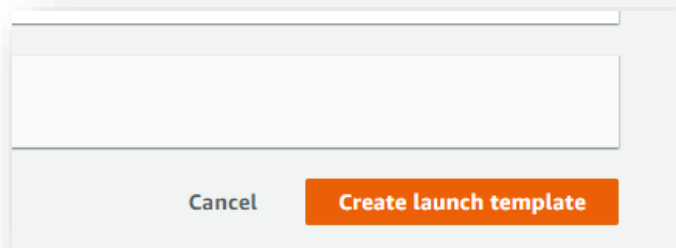
User data has already been base64 encoded

Cancel Create launch template

Paste the following script, it installs apache, sets some

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php echo
file_get_contents('http://169.254.169.254/latest/meta-data/instance-
id/');" > /var/www/html/index.php
```

Then create the launch template and go back to your Auto Scaling Group Wizard, reload the wizard and select your new Launch Template:



Launch template Info [Switch to launch configuration](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

MyCourseLaunchTemplate

[Create a launch template](#)

Version
Default (1)

Description
A simple launch template

AMI ID
ami-0ec1ba09723e5bfac

Key pair name
my-keypair

Launch template
MyCourseLaunchTemplate
lt-0882a8e74dd0eb0b8

Instance type
t2.micro

Security groups
-

Security group IDs
sg-cbc9d3ae

Additional details

Storage (volumes)
-

Date created
Thu Mar 19 2020 14:35:34
GMT+0100 (Central European
Standard Time)

Cancel Next

Select Placement

On the next page select the VPC (there should be only one) and all three subnets to place your instances across all three availability zones. If you have only two subnets then select only two.

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

eu-central-1a | subnet-cfd47ba5
172.31.16.0/20 Default

eu-central-1b | subnet-d5f6eca8
172.31.32.0/20 Default

eu-central-1c | subnet-bc21c8f0
172.31.0.0/20 Default

Select subnets

eu-central-1a | subnet-cfd47ba5 X
172.31.16.0/20 Default

eu-central-1b | subnet-d5f6eca8 X
172.31.32.0/20 Default

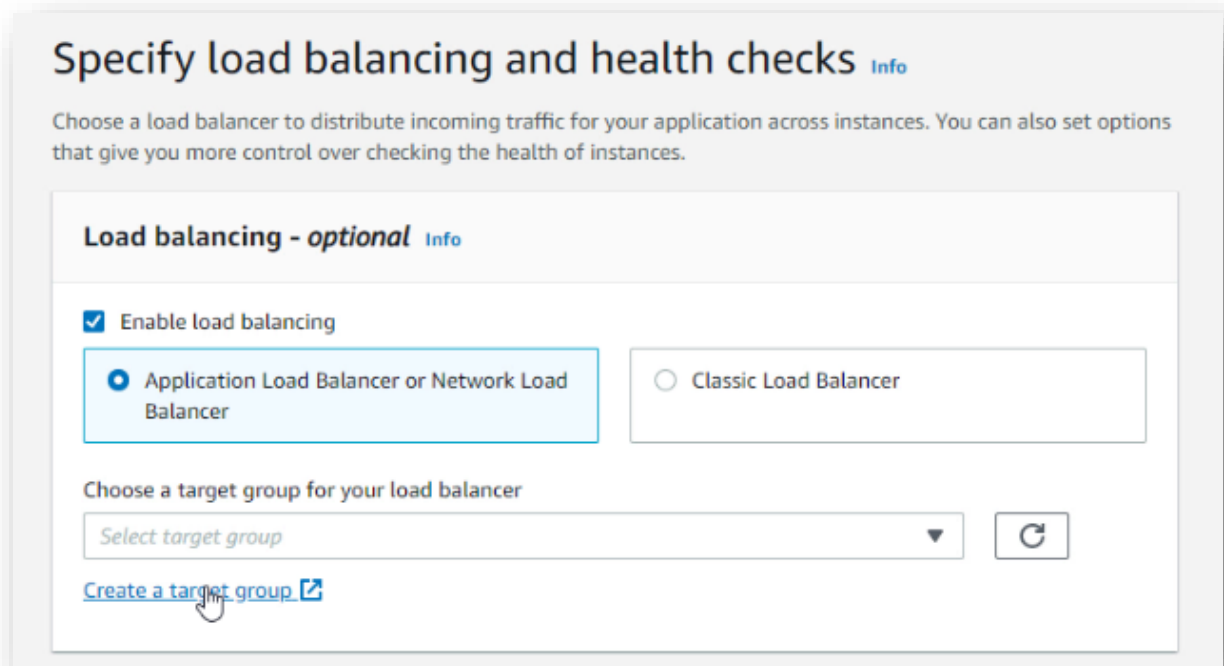
eu-central-1c | subnet-bc21c8f0 X
172.31.0.0/20 Default

[Create a subnet](#)

Cancel Next

Enable Load Balancing

On the next page select that to enable load balancing:



Specify load balancing and health checks [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances. You can also set options that give you more control over checking the health of instances.

Load balancing - optional [Info](#)

☒ **Enable load balancing**

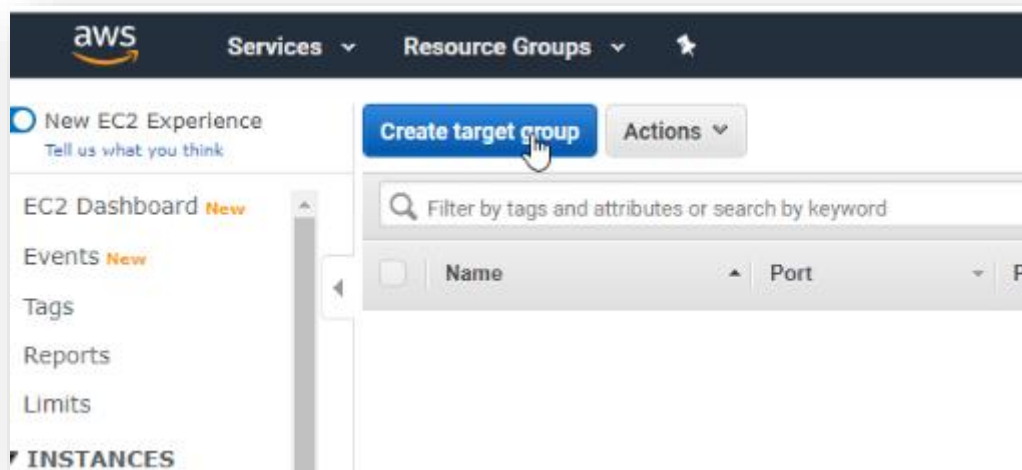
☒ **Application Load Balancer or Network Load Balancer** ☐ **Classic Load Balancer**

Choose a target group for your load balancer

Select target group ▼ [Create a target group](#)

Create a Target Group

As there is no target group yet, create one. Give your target group a name and select as type "Instance":



The screenshot shows the 'Create target group' dialog in the AWS Management Console. The dialog has a title bar with a close button (X) and a breadcrumb trail: Home > Elastic Load Balancing > Target Groups > Create target group. The main content area has a subtitle: 'Your load balancer routes requests to the targets in a target group using the target group settings that you specify, and performs health checks on the targets using the health check settings that you specify.'

The form contains the following fields and options:

- Target group name:** A text input field containing 'MyTargetGroup'.
- Target type:** Radio buttons for 'Instance' (selected), 'IP', and 'Lambda function'.
- Protocol:** A dropdown menu set to 'HTTP'.
- Port:** A text input field containing '80'.
- VPC:** A dropdown menu showing 'vpc-6570b40f (172.31.0.0/16) (My Default Vi...'.
- Health check settings:**
 - Protocol:** A dropdown menu set to 'HTTP'.
 - Path:** A text input field containing '/'.
- Advanced health check settings:** A section header with a right-pointing arrow.

At the bottom right, there are 'Cancel' and 'Create' buttons.

Go back to the other tab and reload the target groups.

The screenshot shows the 'Specify load balancing and health checks' step in the AWS Management Console. The title is 'Specify load balancing and health checks' with an 'Info' link. Below the title is a subtitle: 'Choose a load balancer to distribute incoming traffic for your application across instances. You can also set options that give you more control over checking the health of instances.'

The form is divided into two main sections:

- Load balancing - optional** (with an 'Info' link):
 - Enable load balancing:** A checked checkbox.
 - Load balancer type:** Two radio buttons: 'Application Load Balancer or Network Load Balancer' (selected) and 'Classic Load Balancer'.
 - Choose a target group for your load balancer:** A dropdown menu with a search icon and a refresh icon. The dropdown is open, showing 'MyTargetGroup' as the selected option.
- Health checks - optional**:
 - Health check type:** A link to 'Info'. Below it is a subtitle: 'EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.'
 - Health check type:** Two checkboxes: 'EC2' (checked) and 'ELB'.
 - Health check grace period:** A subtitle: 'The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.' Below it is a text input field containing '300' followed by the word 'seconds'.

At the bottom, there are four buttons: 'Cancel', 'Previous', 'Skip to review', and 'Next'.

Set the Group Size and Scaling Policies

Scaling for your Auto Scaling Group. Select desired instances 2, minimum 1 and maximum 4. For the scaling policy select target tracking scaling policy.

The screenshot shows the AWS Auto Scaling console configuration page. It is divided into two main sections: 'Group size - optional' and 'Scaling policies - optional'. In the 'Group size' section, three input fields are visible: 'Desired capacity' with the value '2', 'Minimum capacity' with the value '1', and 'Maximum capacity' with the value '4'. A red bracket on the left side of these fields indicates they are being set together. In the 'Scaling policies' section, there is a radio button selection for 'Target tracking scaling policy', which is highlighted with a red circle. Below this, a text box explains the policy: 'Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.' To the right of this text box is a radio button for 'None'. Below the policy selection, there are three more fields: 'Scaling policy name' with the value 'Target Tracking Policy', 'Metric type' with a dropdown menu showing 'Average CPU utilization', and 'Target value' with the value '50'. The 'Instances need' section is partially visible at the bottom.

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
1

Maximum capacity
4

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
Average CPU utilization ▼

Target value
50

Instances need

How to scale out or scale in?

Complete AWS DevOps Training for Beginners

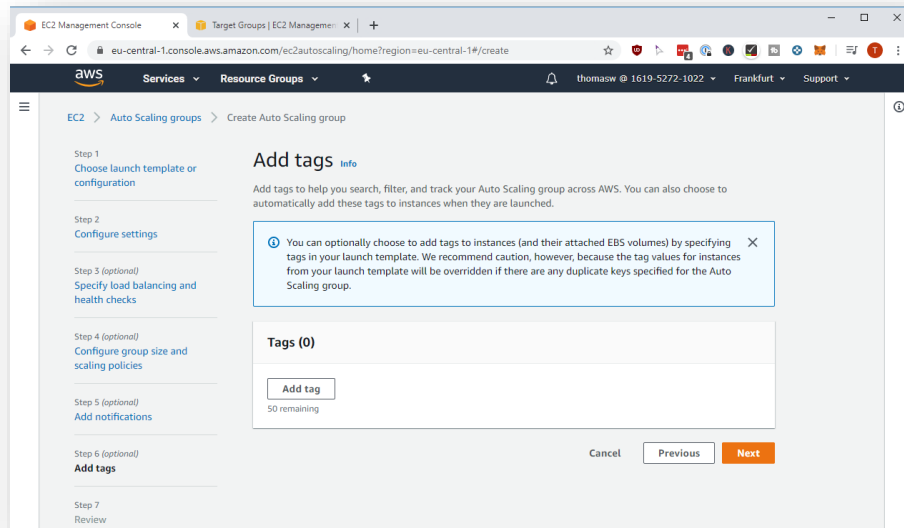
The screenshot shows the AWS Target Tracking scaling policy configuration page. The 'Target tracking scaling policy' section is selected. The 'Scaling policy name' is 'Target Tracking Policy'. The 'Metric type' is 'Average CPU utilization'. The 'Target value' is set to 50. The 'Instances need' is set to 120 seconds warm up before including in metric. The 'Disable scale in to create only a scale-out policy' checkbox is unchecked. The 'Instance scale-in protection - optional' section is shown with the 'Enable instance scale-in protection' checkbox unchecked. The 'Next' button is highlighted with a red circle.

No notifications needed at this point:

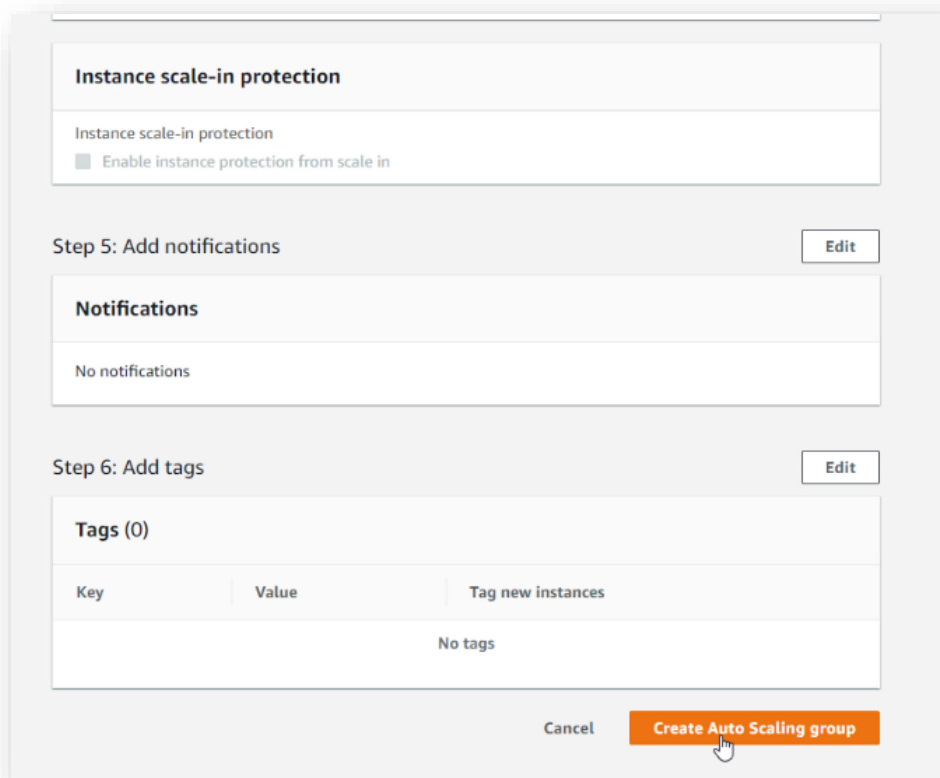
The screenshot shows the 'Add notifications' step in the 'Create Auto Scaling group' wizard. The 'Add notification' button is highlighted. The 'Add notification' button is highlighted.

No Tags needed at this point:

Complete AWS DevOps Training for Beginners

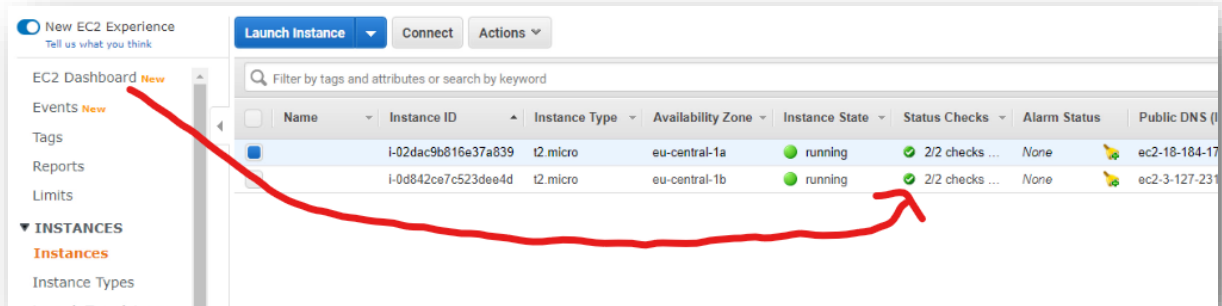


On the last page review everything and click “Create Auto Scaling Group”:

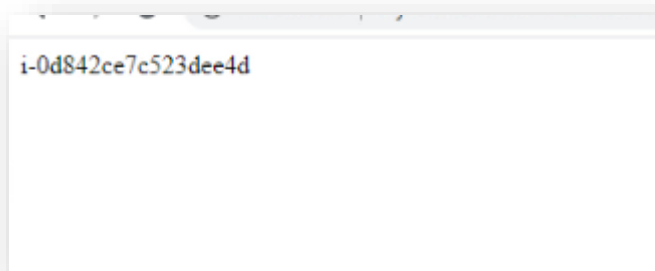
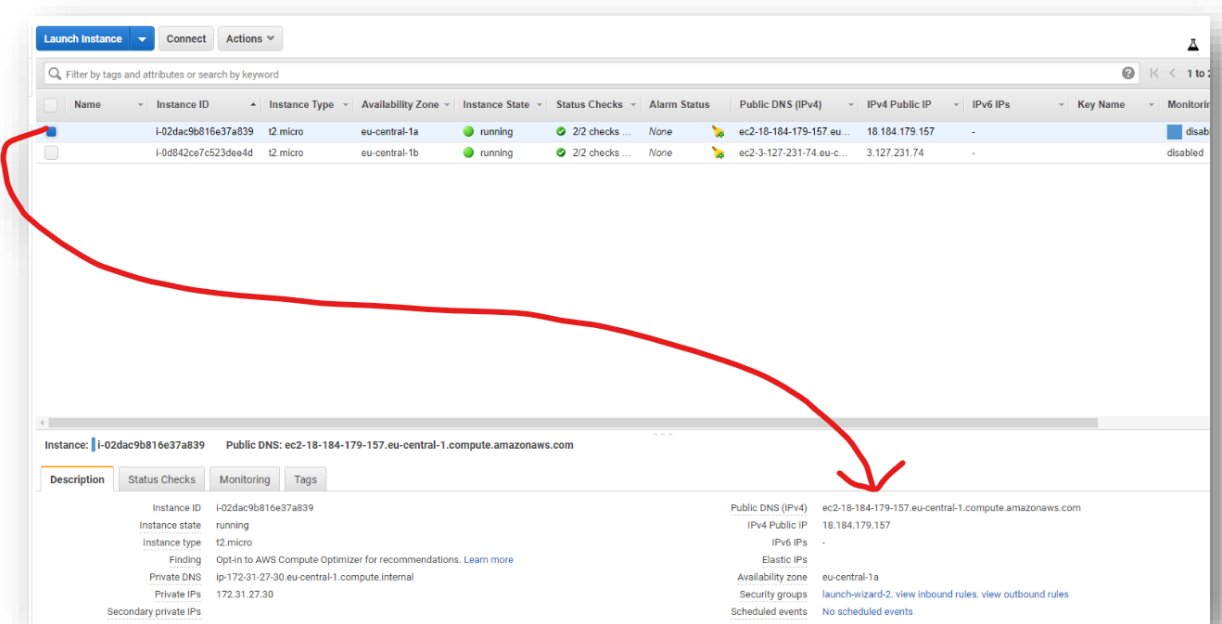


Verify the Instances are Running

Check your EC2 Dashboard and wait until the two instances are ready:



Check if you can reach the instances by opening their public DNS in a new tab:

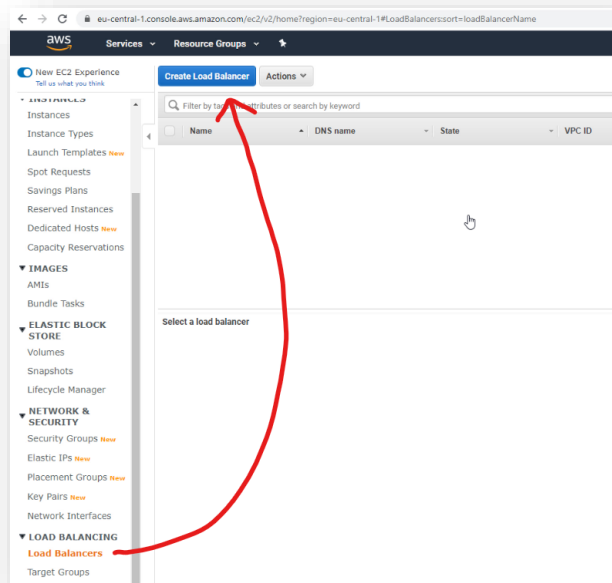


Troubleshooting

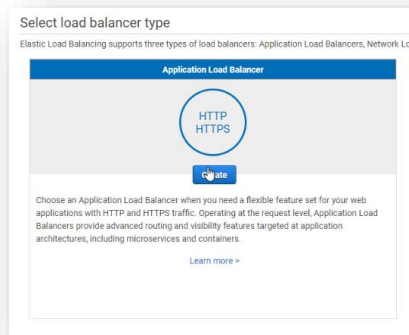
If your instances are up and running, but you can't open the page, make sure the port 80 in the security group is open. Click on the security group inbound rules and add port 80.

Create a Load Balancer

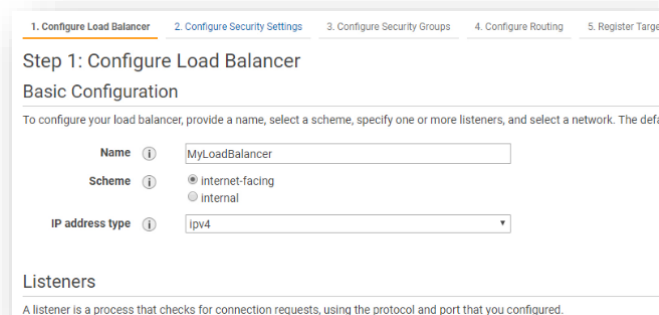
Next, we create a load balancer. You find them on the left side in your EC2 Dashboard under “Load Balancing”:



Select the Application Load Balancer



Give the Load Balancer a name:



And place the load balancer in all three subnets:

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones.

VPC vpc-6570b40f (172.31.0.0/16) (default)

Availability Zones

- ☒ **eu-central-1a** subnet-cfd47ba5
IPv4 address Assigned by AWS
- ☒ **eu-central-1b** subnet-d5f6eca8
IPv4 address Assigned by AWS
- ☒ **eu-central-1c** subnet-bc21c8f0
IPv4 address Assigned by AWS

Attach the same security group that has port 80 open:

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can assign a security group to your load balancer.

Assign a security group:

- ☐ Create a new security group
- ☒ Select an existing security group

Security Group ID	Name
sg-cbc9d3ae	default
sg-008a6d64d3018e21e	launch-wizard-2

Select the *existing* Target Group from your Auto Scaling Group:

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks to ensure that the targets are healthy.

Target group

Target group Existing target group

Name MyTargetGroup

Target type

- ☒ Instance
- ☐ IP
- ☐ Lambda function

Protocol HTTP

Port 80

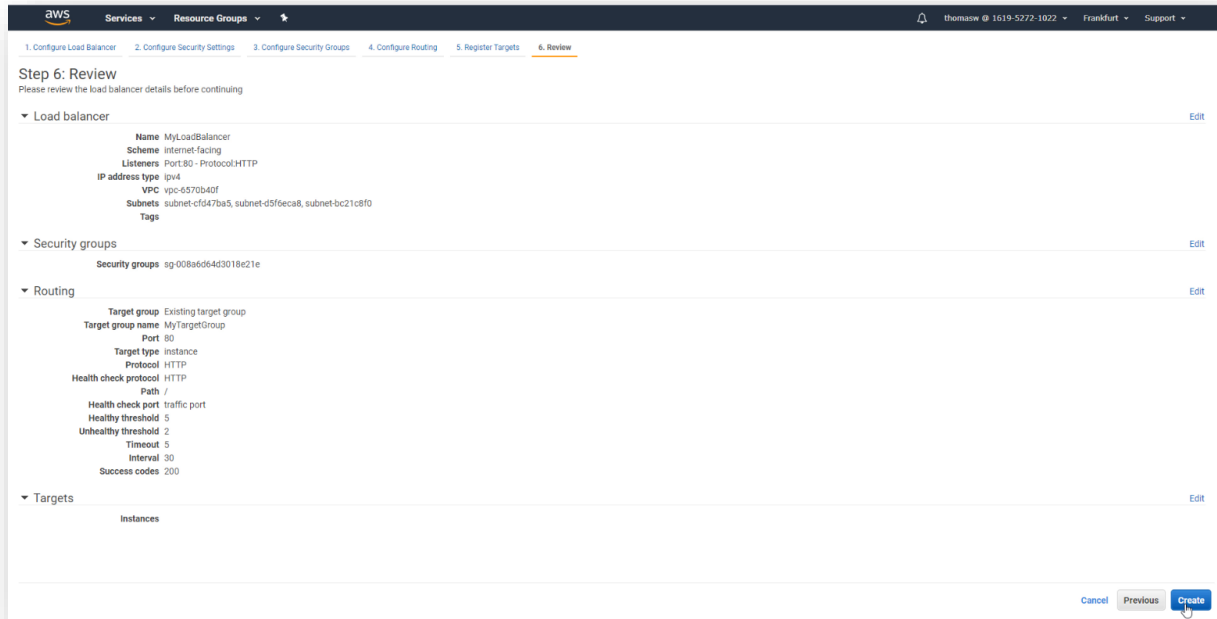
Health checks

Protocol HTTP

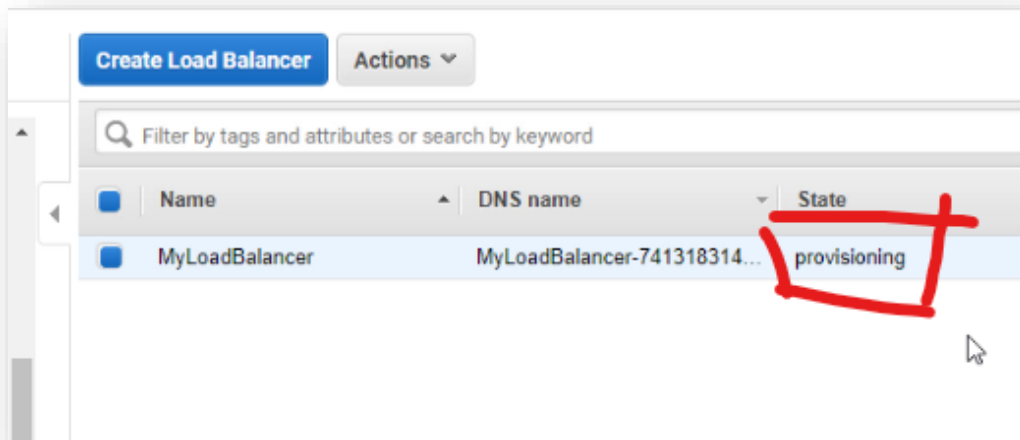
Path /

► Advanced health check settings

Review everything and click Create:

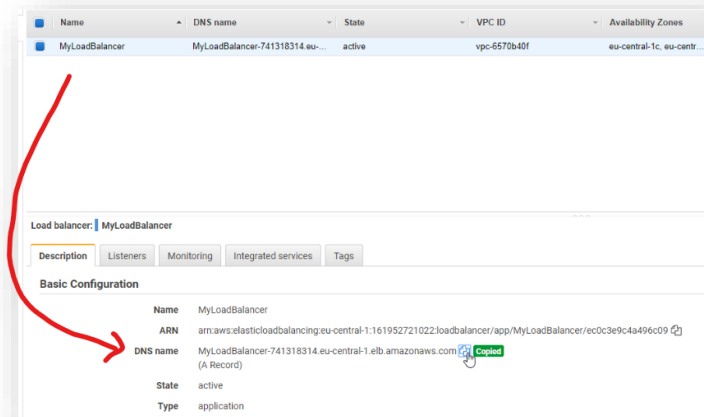


Wait until the Load Balancer is provisioned:

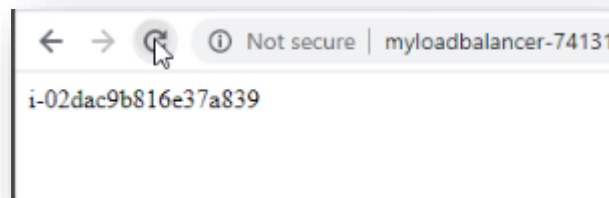
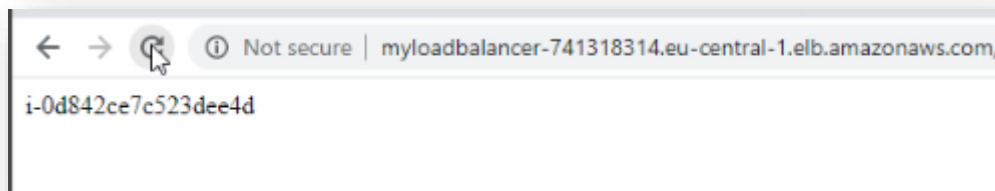


Testing the Load Balancer

Once it's ready copy the address of the load balancer and open it in a new tab:



If you reload the tab, see that the instance id changes, because the load balancer does round robin load balancing:



Tear Down and Clean Up

To save costs, let's remove everything again.

1. Remove the Auto Scaling Group
2. Remove the Load Balancer
3. Remove the Target Group
4. Remove the Instance Launch Template
5. Terminate the remaining EC2 Instances

Lab End
