General Immersion Day

Lab 2

Auto Scaling on AWS

EC2 Auto Scaling

EC2 Auto Scaling Lab Overview

An overview of auto scaling on AWS can be found here. This lab will first walk you through the process of creating an Amazon Machine Image (AMI) from a web host created with CloudFormation. Then it will walk you through creating a launch template and setting up the web host within an auto scaling group behind an Appliction Load Balancer (ALB). The end result will be an auto scaling group behind a load balancer that scales based on CPU utilization of the hosts.

Services and concepts covered in this lab:

- Primary: Auto Scaling, Launch Templates, Creation and configuration of Security Groups
- Secondary: AMIs, Application Load Balancers

This lab will walk you through the following:

- Lab Prerequisites
- Creating a Launch Template
- Setup an Auto Scaling Group
- Configuring Security Groups
- Testing the Auto Scaling Group

2-1 Lab Prerequisites

To create an AMI (Amazon Machine Image) for our Auto-Scaling group we will first need to setup a web host. We will generate an AMI from the instance and then auto-scale the instance behind a load balancer. Click on the telmplate link button below to build the web host in EC2 using **CloudFormation.**

Download and launch the CloudFormation template

1. Download the "EC2-Auto-Scaling-Lab.yaml" CloudFormation template by **right-clicking** on this link and save it to your local hard drive.
2. In the AWS Console search for CloudFormation or select the **Services** menu and click on **CloudFormation** under "Management & Governance".
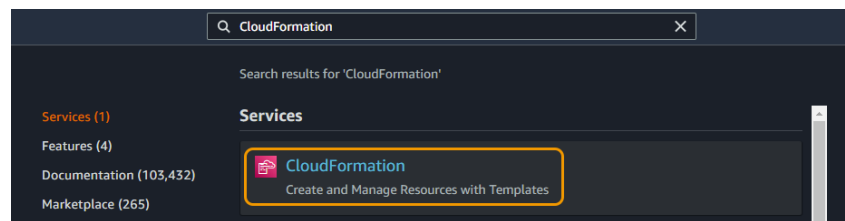
3. In the CloudFormation console select the **Create stack** button and then select **With new resources (standard)**.
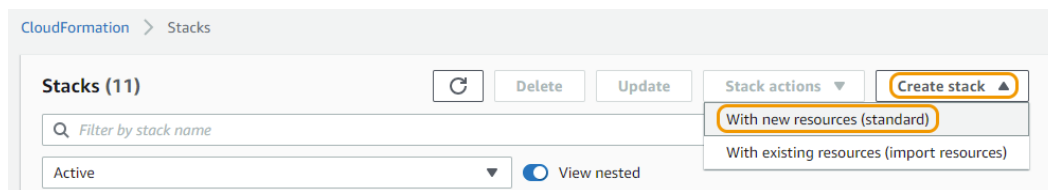
4. Under "Template source" select **Upload a template file** and then select the **Choose file** button. Select the "EC2-Auto-Scaling-Lab.yaml" template file you downloaded in the first step. Once you have selected the template file click on the **Next** button.
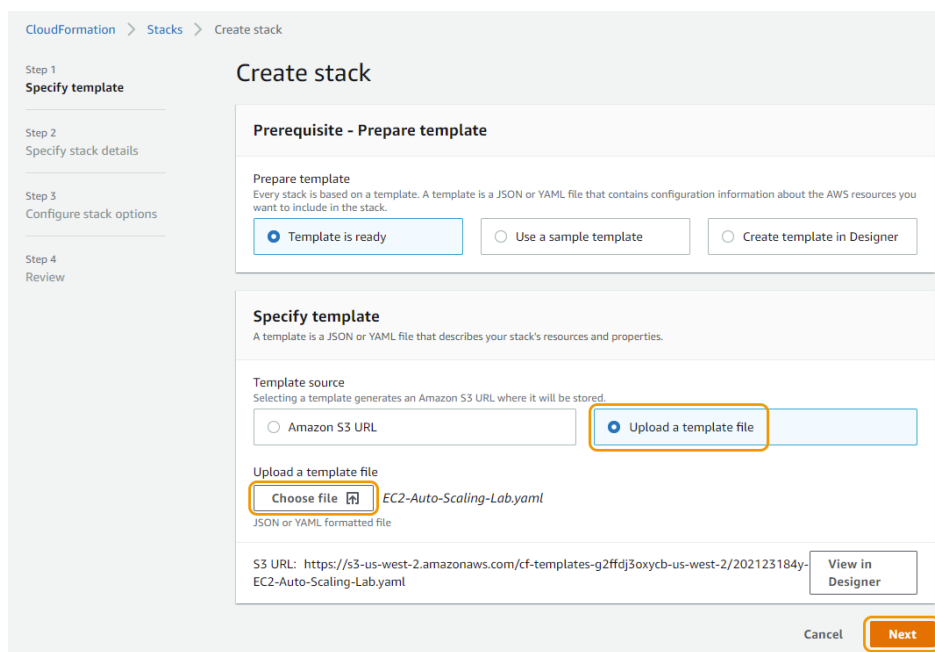
5. On the Specify stack details page, fill in the following fields:
   a. Under "Stack name" name your stack [Your Initials]-EC2-Web-Host.
   b. You can leave "AmiID" as the default, resulting in the use the most recent version of this AMI.
   c. Under "InstanceType" **select** the **m5.large** or the t2.micro. It is recommended that you use m5.large size instances to demonstrate real world performance as t2 types instances are not recommended for production workloads. If you have an issue with the m5 instances for any reason, load the Cloudformation template again and select the t2.micro, it is sufficient enough for this lab.
   d. Under "MyIP" **input** the IP address of your local machine followed by a /32. This will lock down HTTP port 80 to your machine. You can find you local IP by searching What is my IP.
   e. Under "MyVPC" **select** the VPC you want to use to setup the instance. In most accounts the default VPC will be a good choice and in new AWS accounts it will be the only choice.
   f. Under "PublicSubnet" **select** a subnet within your VPC that has internet access. A public subnet is defined by a subnet having a route to the internet gateway within it's route table. By default, the Default VPC subnets are all public.

Stack name

Stack name

```
ABC-EC2-Web-Host
```

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AmiID
The AMI ID - Leave as Default

```
/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
```

InstanceType
Web Host EC2 instance type

```
m5.large                                                        ▼
```

MyIP
Please enter your local IP address followed by a /32 to restrict HTTP(80) access. To find your IP use an internet search phrase "What is my IP".

```
25.555.555.245/32
```

MyVPC
Select Your VPC (Most Likely the Default VPC)

```
vpc-84252dfc (172.31.0.0/16)                                    ▼
```

PublicSubnet
Select a Public Subnet from your VPC that has access to the internet

```
subnet-25e59178 (172.31.0.0/20)                                 ▼
```

Cancel   Previous   **Next**

6. Once you are done entering the details above, click on **Next**. On the next page, "Configure stack options", you can leave "Tags", "Permissions", and "Advanced options" as default and select **Next**.
7. On the Review [Your Initials]-EC2-Web-Host page, review your settings and click on **Create stack** to start building your web server.
8. Wait till the "Logical ID" "[Your Initials]-EC2-Web-Host" shows a status of "CREATE_COMPLETE".

Note: The CloudFormation stack creation should be completed in about 3 minutes.

Confirm the successful setup of your instance:

1. Navigate to the EC2 service page by selecting **Services** and then **EC2** or search for EC2 in the search bar.
2. Select **Instances** from the left hand menu. On the "Instances" page, select your instance "[Your Initials]-S3-Web-Host" and copy the "Public IPv4 DNS" address into your clipboard by clicking on the image of two overlapping squares to the left of the Public IPv4 DNS Address. Paste this address into a new tab on your web browser.

Note: Clicking on the open address link under "Public IPv4 DNS" heading may result in you not being able to see your website. The "open address" link uses https:// instead of http://, which will result with an error because our web host has not been setup with an SSL cert.

3. You should now see the from page titled "EC2 Instance Metadata".

Note: If the page does not load it is recommended you wait for your instance "Status check" to show "2/2 checks passed" and then try again. Your metadata will look similar but **will not match** the image above.

Generate a custom AMI of the web server created in the EC2 - Linux Lab:

Now that we have our instance setup to host our website, we will generate a custom machine image for our auto scaling group. This will create an image of our web host that will be used by our Auto Scaling group to spin up multiple instances based on server load.
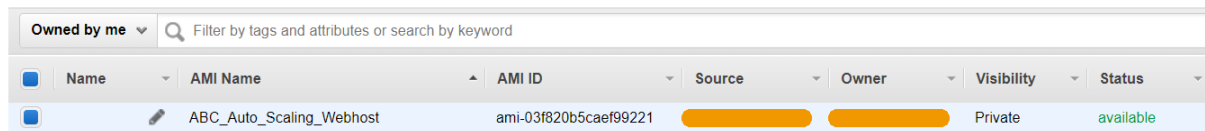
1. In the EC2 Console under Instances, you can create Amazon Machine Images (AMIs) from either running or stopped instances. Return to or open the EC2 console.
2. **Right-click** your webhost instance named "[Your Initials] - Web Server" and under "Image and templates" choose **Create image** from the context menu. (This can also be done by selecting the instance and clicking on the Actions menu in the upper right hand corner)
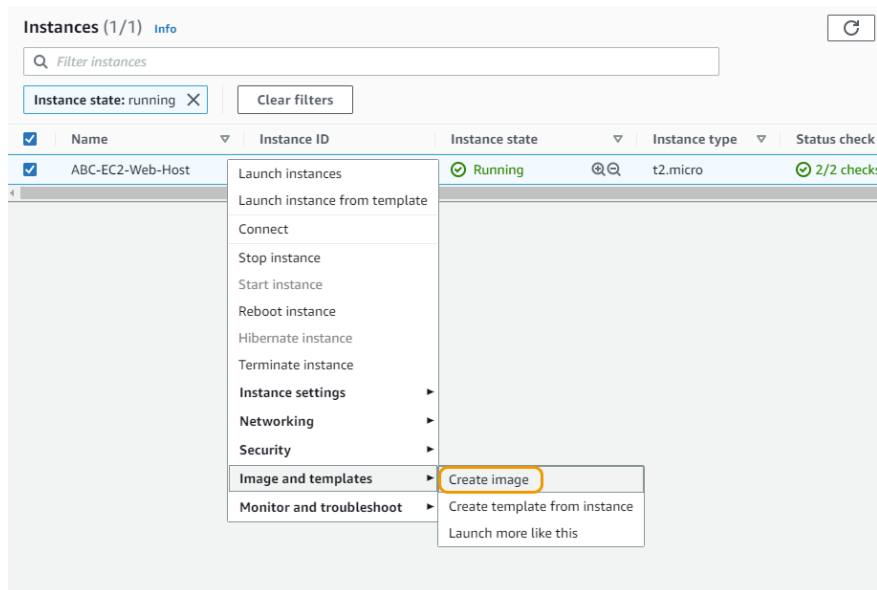
3. On the "Create Image" page, put in the Image name [Your Initials]_Auto_Scaling_Webhost and a description. You can leave the Instance volumes as default and then choose **Create Image**.

4. It may take a few minutes for the AMI to be created. In the EC2 console under "Images" in the left hand menu select **AMIs**. You should see the AMI you just created, it may be in a pending state, but after a few moments it will transition to an available state.
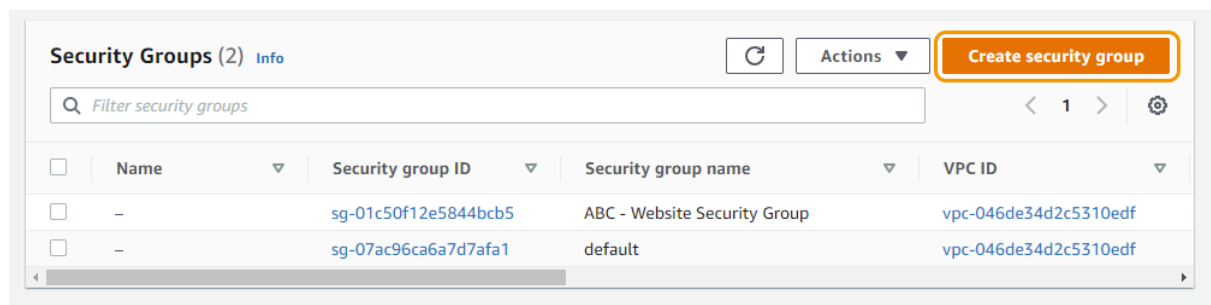
	Name	▼	AMI Name	▲	AMI ID	▼	Source	▼	Owner	▼	Visibility	▼	Status	▼
☑		✏	ABC_Auto_Scaling_Webhost		ami-03f820b5caef99221						Private		available	

We are done with our new Amazon Machine Image for now, we can now move on to setting up our auto scaling security group.

Create a new Security Group for our Auto Scaling Group:

Before we get into setting up our Launch Template we will need to setup a special Security Group for our Auto Scaling Group. A security group provides instance (virtual machine) level protection. If you want to know more about security groups, check out this page.

1. Within the console Under "Services" select **EC2** or search EC2 in the search bar. On the EC2 page under the "Network & Security" heading in the left-hand menu select **Security Groups**. You should see other security groups, including the security group for your web server named [Your Initials]-EC2-WEb-Host - Website Security Group. To start the creation of a new security group, click on the **Create security group** button.

Security Groups (2) Info — Actions ▼ — **Create security group**

	Name	▽	Security group ID	▽	Security group name	▽	VPC ID	▽
☐	–		sg-01c50f12e5844bcb5		ABC - Website Security Group		vpc-046de34d2c5310edf	
☐	–		sg-07ac96ca6a7d7afa1		default		vpc-046de34d2c5310edf	

2. Name your security group [Your Initials] - Auto Scaling SG and you can use the same name for the description as well and make sure you have the correct VPC select. (Most likely the **Default** VPC unless you setup a new one for this lab)

3. Under "Inbound rules" there currently are not any rules created, we will leave it empty for now. We will be creating a rule later in this lab but we need our load balancer security group to exist first.

4. "Outbound rules" currently allow all traffic out so there is no need for any additional rules.

Now click on **Create security group**.

2-2 Creating a Launch Template

Auto Scaling Lab Architecture:

Below is a diagram of our end state architecture. Let's get building!
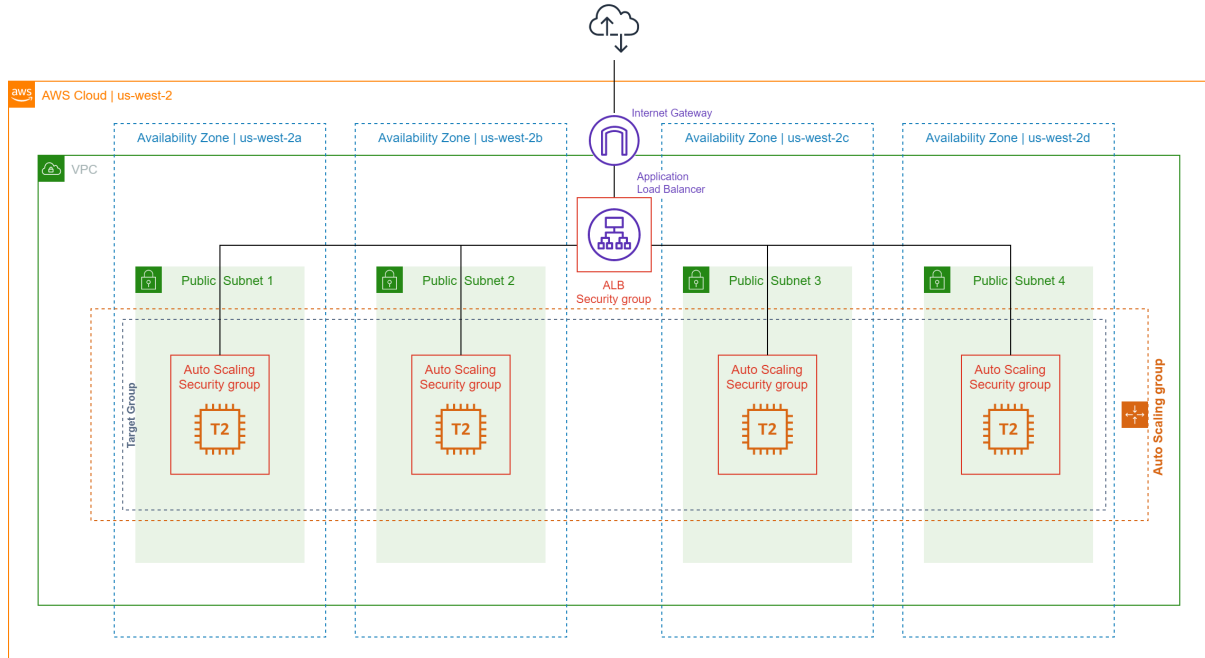
There are three main components to EC2 Auto Scaling on AWS

1. Launch Template: A Launch Template is a feature of EC2 Auto Scaling that allows a way to templatize your launch requests. It enables you to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain a specific Amazon Machine Image, instance type, storage, and networking settings that you typically use to launch instances. For each Launch Template, you can create one or more numbered Launch Template Versions. Each version can have different launch parameters.

2. Auto Scaling Groups: For auto scaling your EC2 instances are organized into groups so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and desired number of EC2 instances.

3. Scaling Policies: A Scaling Policy tells Auto Scaling when and how to scale. Scaling can occur manually, on a schedule, on demand, or you can use Auto Scaling to maintain a specific number of instances.

Auto Scaling is well suited for applications that have unpredictable demand patterns that can experience hourly, daily, or weekly variability in usage. This helps you to manage your cost and eliminate over-provisioning of capacity during times when it is not needed. Auto Scaling can also find an unhealthy instance, terminate that instance, and launch a new one based on the scaling plan.

The number of EC2 instances can be scaled in or out as Auto Scaling responds to the metrics you define when creating these groups.

- You can specify the minimum number of instances in each Auto Scaling Group, so that your group never goes below this size. (Even if the instances are determined to be unhealthy)

- You can specify the maximum number of instances in each Auto Scaling Group, so that your group never goes above this size.
- You can specify a desired capacity to specify the number of healthy instances your auto scaling group should have at all times. (More information can be found here)
- You can specify scaling policies so that Auto Scaling will modify the desired target capacity mentioned in the previous point. It will launch or terminate instances as demand on your application increases or decreases.

Creating a Launch Template

When you create an Auto Scaling Group, you must specify a Launch Template. The first step in this lab is to create the Launch Template for an EC2 Auto Scaling Group.

1. Under the "Services" select **EC2**.
2. In the left navigation pane, find "Instances" and select **Launch Templates**.
3. Now select **Create launch template**.
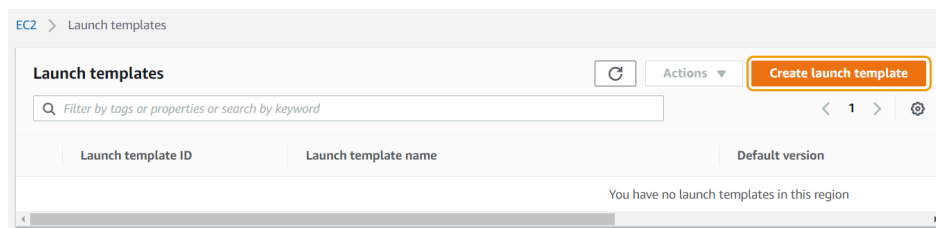
4. This takes you to the "Create launch template" page, starting with the "Launch template name and description":
   a. *Launch template name*: [Your Initials]-scaling-template
   b. *Template version description*: This is optional
   c. *Auto scaling guidance*: **Check the box** to provide guidance

5. "Launch Template Contents" defines the parameters for the instances in the Auto Scaling group:
   a. *Amazon machine image (AMI)*: Select "My AMIs" and "Owned by me". In the drop down search by typing your initials and select the custom AMI you just created **[Your Initials]_Auto_Scaling_Webhost**. (The new AMI you just created may already be selected)
   b. *Instance type*: **t2.micro**
   c. *Key pair (Login)*: **Select** the Key Pair you created in the first lab, it is most likely call [Your Name]-ImmersionDay
   d. *Networking Settings*:
   e. Subnet: **Don't include in launch template**
   f. Firewall (security groups): Select "Select existing security group" and then select the security group you created in the first part of this lab named **[Your Initials] - Auto Scaling SG**
   g. e. *Configure storage*: Leave as default
   g. *Resource tags*: None
   h. *Advanced Details*: **IMPORTANT**: Select the arrow to expand "Advanced details" and under "Detailed Cloudwatch monitoring" select **Enable**. Leave everything else as the default.

Here, you are enabling CloudWatch Detailed monitoring. By default, your instance has basic monitoring in 5-minute intervals for the instances. After you enable detailed monitoring CloudWatch will monitor the instances in your auto scaling group in 1-minute intervals. This will allow the auto scaling group to respond quicker to changes in the group.

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*

ABC-scaling-template

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

Template version description

V1

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**

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon machine Image) - required **Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 *Search our full catalog including 1000s of application and OS images*

Recents	**My AMIs**	Quick Start

○ ● Owned by me   ○ Shared with me

🔍 **Browse more AMIs**

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

ABC_Auto_Scaling_Webhost
ami-05d6c9992b719eb30
2022-03-05T23:23:16.000Z   Virtualization: hvm   ENA enabled: true   Root device type: ebs   ▼

Description

ABC - Auto Scaling Webhost

Architecture   AMI ID

x86_64   ami-05d6c9992b719eb30

▼ Instance type Info

Advanced

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour ▼

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

ABC-Keypair ▼   ⟳   Create new key pair

▼ Network settings

Subnet Info

Don't include in launch template ▼   ⟳   Create new subnet ↗

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

◉ Select existing security group   ○ Create security group

Security groups Info

Select security groups ▼   ⟳   Compare security group rules

ABC - Auto Scaling SG sg-042328f58c6a90756 ✕
VPC: vpc-84252dfc

▶ Advanced network configuration

▼ Configure storage Info

Advanced

1x 8 GiB gp2 ▼ Root volume

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage   ✕

Add new volume

▼ Resource tags Info

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

Add tag

50 remaining (Up to 50 tags maximum)

6. When you are sure the configurations are correct, click **Create launch template**, then **View launch templates**. You are now finished creating your Launch Template.

You are now ready to move onto the next step: Setup an Auto Scaling Group

2-3 Setup an Auto Scaling Group

Create Auto Scaling Group

You have created a Launch Template, which defines the parameters of the instances launched. Now we will create an Auto Scaling Group so that you can define how many EC2 instances should be launched and where to launch them.

1. Make sure you are on the EC2 Service page.
2. In the left navigation pane, find "Auto Scaling" and Select **Auto Scaling Groups**.
3. Click **Create an Auto Scaling group**.
4. Give the Auto Scaling group a name: [Your Initials]-Lab-AutoScaling-Group
5. From the Launch Template drop down choose the launch template named **[Your Initials]-scaling-template** you created in the previous section and select **Next**.

6. Configure settings page, configure the following and select **Next**:
   a. *Purchase options and instance types*: Select **Adhere to launch template**
   b. *Network*:
      o VPC: **Select** your VPC (most likely Default)
      o Subnets: **Select** the subnets where you would like the auto scaling group to use when spinning up the hosts. (If you are using the default VPC, this will most likely be four subnets, as shown below)

A best practice for your Auto Scaling Group would be to select only private subnets. The instances will be sitting behind a load balancer and will not need public IP addresses. For the sake of this lab, they could be either private or public subnets.

7. Specify load balancing and health checks:

   a. *Load balancing*: **Attach to a new load balancer**
   b. *Load balancer type*: **Application Load Balancer**
   c. *Load balancer name*: [Your Initials]-Application-Load-Balancer
   d. *Load balancer scheme*: **Internet-facing**
   e. *Networking mapping*: You should see all the Availability Zones and subnets you selected in the previous step. (If you had multiple subnets per AZ, this were it would let you choose between them)
   f. *Listeners and routing*: Keep the Port as 80 and select **Create a target group** from the "Default routing (forward to)" dropdown.
   g. *New target group name*: [Your Initials]-Target-Group
      - The target group is where your load balancer is going to look for instances to distribute traffic. We are setting our auto scaling group to automatically register instances into this group and it will also be associated to our load balancer.

   h. *Health checks & Additional settings: Leave as default and select **Next**.*

Configure advanced options Info

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - *optional* Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

| ○ **No load balancer** Traffic to your Auto Scaling group will not be fronted by a load balancer. | ○ **Attach to an existing load balancer** Choose from your existing load balancers. | ● **Attach to a new load balancer** Quickly create a basic load balancer to attach to your Auto Scaling group. |

Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, **visit the Load Balancing console.** 🗗

| ● **Application Load Balancer** HTTP, HTTPS | ○ **Network Load Balancer** TCP, UDP, TLS |

Load balancer name
Name cannot be changed after the load balancer is created.

`ABC-Application-Load-Balancer`

Load balancer scheme
Scheme cannot be changed after the load balancer is created.

| ○ Internal | ● Internet-facing |

Network mapping
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
vpc-0c138a8f51f4940c3 🗗   Default VPC

Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☑ us-west-2c	subnet-0b4795e7f7631df4f ▼
☑ us-west-2a	subnet-04265983a546e735f ▼
☑ us-west-2b	subnet-061caf8714af918ed ▼
☑ us-west-2d	subnet-0fe9816052c14f483 ▼

Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the **Load Balancing console** 🗗 after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group ▼

New target group name
An instance target group with default settings will be created.

`ABC-Target-Group`

Tags - *optional*
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add tag]

50 remaining

Health checks - *optional*

Health check type Info
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.

☑ EC2   ☐ ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

`300` seconds

Additional settings - *optional*

Monitoring Info
☐ Enable group metrics collection within CloudWatch

Cancel   Previous   Skip to review   **Next**

8. Configure the group size and scaling policies below and then select **Next**.
   a. *Group Size*: The settings below will keep our group size to one EC2 instance unless a scaling policy is triggered.

      o Desired capacity: 1
      o Minimum capacity: 1
      o Maximum Capacity: 5

   b. *Scaling policies*: Select **Target tracking scaling policy**

      o Metric type: **Average CPU utilization**
      o Target Value: 25

We are going to set our target CPU utilization low to speed up the lab.

9. Add Notifications:
You can configure your Auto Scaling Group to send notifications to an endpoint that you choose, such as an email address. You can receive notifications whenever a specified event takes place, including the successful launch of an instance, failed instance launch, instance termination, and failed instance termination.
For now, you are going to skip this step, select **Next**.

10. Add Tags: Add a single tag and then select **Next**.
   a. Select the **Add tag** button and configure the following:
       o   Key: Name
       o   Value: [Your Initials] - Auto Scaling Group

11. Review you settings and then select **Create Auto Scaling group**. You have now created your Auto Scaling Group, target group and load balancer.
    a. You will soon see a new instance created by the Auto Scaling group in the EC2 console with the name tag "[Your Initials] - Auto Scaling Group". (You may need to refresh the screen to see the instance)
    b. If you select **Load Balancers** under "Load Balancing" in the left hand menu, you will see your load balancer provisioning.

In the next step we will create an additional security group and update the security settings to allow traffic to flow between the ALB and our web hosts.
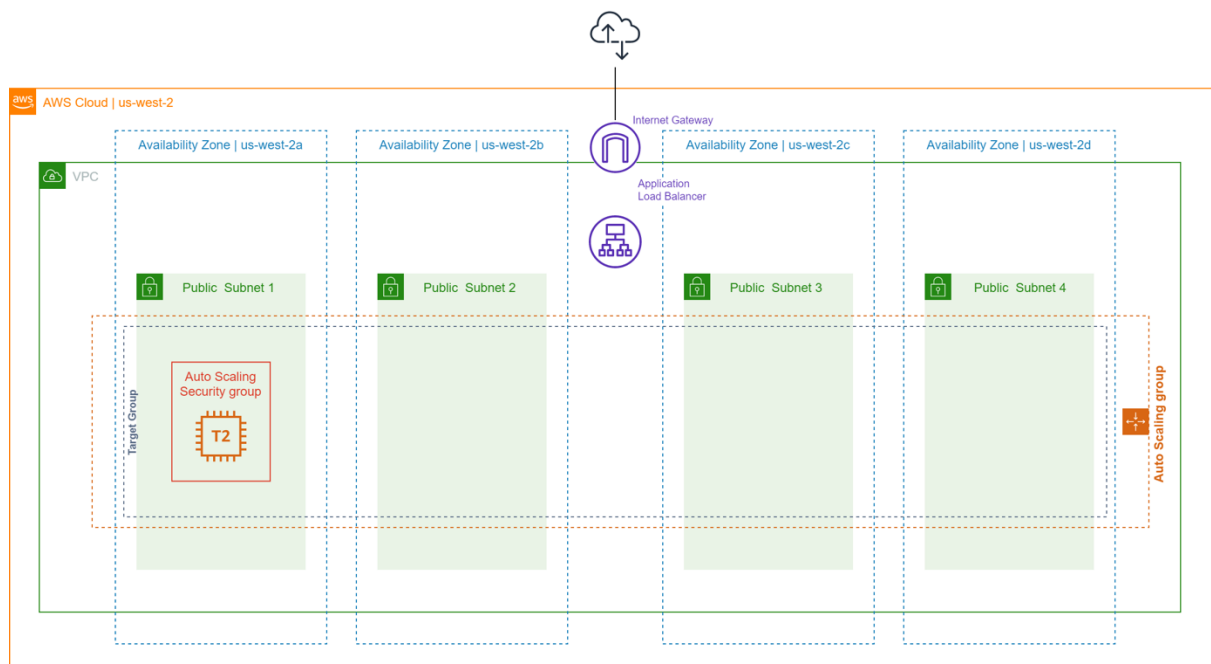
You are now ready to move onto the next step: **Configuring Security Groups**

2-4 Configuring Security Groups

Creating a Load Balancer Security Group

When our load balancer was provisioned it was setup with the default security group in our VPC. To allow access to the load balancer via the public DNS, we will need to create and attach a security group to allow inbound traffic on port 80 from the internet.

We will also create an outbound rule that allows outgoing traffic from the load balancer to only be sent to hosts within the Auto Scaling Security Group.

Within the console Under "Services" search and select **EC2**. On the EC2 page under the "Network & Security" heading in the left-hand menu select **Security Groups**. You should see other security groups, including the security group for your web server named [Your Initials]-EC2-Web-Host - Website Security Group. Click on the **Create security group** button.

1. Basic details:
   a. *Security group name*: [Your Initials]-SG-Load-Balancer
   b. *Description*: [Your Initials]-SG-Load-Balancer
   c. *VPC*: Select your VPC (Most likely the Default VPC)

2. Inbound rules:
   a. Click on the **Add rule** button
   b. *Type*: HTTP
   c. *Source*: Custom: [Input your private IP address followed by a /32] (You can find you local IP by searching <u>What is my IP. </u>)

3. Outbound rules:
   a. Find the "All traffic" rule and click on **Delete** to remove the rule. (All Outbound rules should now be removed)
   b. Click on the **Add rule** button
   c. *Type*: HTTP
   d. Under "Destination" select **Custom** and in the field select your **[Your Initials]-Auto Scaling SG** as the "Destination". Hint: start by typing sg to get the Security Group list.
   e. You security group configuration should look similar to the image below. Select **Create security group** when finished.

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name   Info
ABC-SG-Load-Balancer
Name cannot be edited after creation.

Description   Info
ABC-SG-Load-Balancer

VPC   Info
vpc-0c138a8f51f4940c3 (Default VPC)

Inbound rules   Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
HTTP	TCP	80	Custom Q		Delete
			25.555.555.245/32 X		

Add rule

Outbound rules   Info

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
HTTP	TCP	80	Custom Q		Delete
			sg-0f0fdef732325b855 X		

Add rule

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag
You can add up to 50 more tag

Cancel   Create security group

4.   Attach your new Load Balancer Security group to your Load Balancer:

a.   On the EC2 service page left side menu find "Load Balancing" and select **Load Balancers**. Select the load balancer you created. Make sure the State is "Active".
b.   Under the "Description" tab scroll down to the "Security" section and click on **Edit security groups**.
c.   Select the box to the left of your new load balancer sg named [Your Initials]-SG-Load-Balancer.
d.   Make sure you also **un-select** any other security group and then click on the **Save** button.

Task 4: Add Inbound Rule to the Auto Scaling Security Group

1. We will need to setup a rule to only allow traffic from the new Load Balancer Security Group to the Auto Scaling Security Group. This will be one of the layers of protection that will prevent our webhosts from being directly accessed from the internet.

   a. On the EC2 service page left side menu under "Network & Security" select **Security Groups**.
   b. Select your Auto Scaling Security Group: **[Your Initials] - Auto Scaling SG**
   c. Select the **Inbound Rules** tab and click on the **Edit inbound rules** button and then the **Add rule** button.
   d. From the "Type" drop down select **HTTP**. Under "Source" select **Custom** and in the field specify your **[Your Initials]-SG-Load-Balancer** as the "Source". Hint: start by typing sg to get the Security Group list. Now click on **Save rules**.

Your rule should now look similar to the image below.

2. We will now test to make sure your load balancer is working. There is currently only one instance (or target) running in the auto scaling group, but you should be able to access the website.

   Return to your the load balancers page by selecting **Load Balancers** from the left hand menu. Under the "Description" tab **copy** the DNS name and **paste** it into a web browser. You should now see the website being loaded from your auto scaling group. Leave this page open, you will need it in the next step.

Load balancer: ▌ ABC-Application-Load-Balancer

| Description | Listeners | Monitoring | Integrated services | Tags |

Basic Configuration

Name	ABC-Application-Load-Balancer
ARN	arn:aws:elasticloadbalancing:us-west-2:650564822972:loadbalancer/app/ABC-Application-Load-Balancer/676990d81c9a1d70
DNS name	ABC-Application-Load-Balancer-1550330346.us-west-2.elb.amazonaws.com
	(A Record)
State	Active
Type	application
Scheme	internet-facing

WELCOME!

powered by aws

EC2 Instance Metadata

EC2 Instance ID: i-059b0d74bbea12a6c

Availability Zone: us-west-2c

Private IP: 172.31.12.207

★

Start CPU Load Generation

Current CPU Load: 0%

You are now ready to move onto the next step: Testing the Auto Scaling Group

2-5 Testing the Auto Scaling Group

Testing the Auto Scaling Group

Now that you have created your Auto Scaling Group and load balancer, let's test it to ensure that everything is working correctly.

1. Make sure you are on the website accessed through the Load Balancer DNS address in the previous step.
2. At the bottom of the front page click on the **Start CPU Load Generation** link: Once the CPU load goes above 25% for a sustained period the Auto Scaling policy will begin spinning up the instances specified in the launch template to meet demand. *(You may have to do this twice if the first time doesn't generate enough load)*

Start CPU Load Generation

Current CPU Load: 0%

3. In the "Instances" section of the EC2 Console you can watch for the new instances created by Auto Scaling, this might take a couple of minutes. Refresh the EC2 instances page and you should soon see a new instance spinning up automatically. You can select the instance named [Your Initials] - Auto Scaling Group and click on the **Monitoring** tab below to keep an eye on the "CPU Utilization".

	Name	Instance ID	Instance state		Instance type	Status check	Alarm status		Availability Zone
☐	ABC - Auto Scaling Group	i-07ff59ae556077ecd	⏱ Pending	⊕⊖	t2.micro	–	No alarms	+	us-west-2a
☐	ABC - Auto Scaling Group	i-021b3a03152ca2169	⏱ Pending	⊕⊖	t2.micro	–	No alarms	+	us-west-2b
☐	ABC-EC2-Web-Host	i-0b6cab06d0d0934e1	⊘ Running	⊕⊖	m5.large	⊘ 2/2 checks passed.	No alarms	+	us-west-2c
☐	ABC - Auto Scaling Group	i-09abbc381b972202e	⊘ Running	⊕⊖	t2.micro	⊘ 2/2 checks passed.	No alarms	+	us-west-2c

4. You can also see this by going to the Auto Scaling Groups page. https://console.aws.amazon.com/ec2autoscaling

   Then select your auto scaling group **[Your Initials]-Lab-AutoScaling-Group**. If you look at the details under the Instance management tab, you can see if new instances are spinning up. You can look at the Instance management tab to see how many instances there are in your group currently. The monitoring tab shows you different metrics like group size, pending instances, total instances, and much more.

Details   Activity   Automatic scaling   **Instance management**   Monitoring

Instances (4)

Filter instances

	Instance ID ▽	Lifecycle ▽	Instance type ▽	Launch template/configuration ▽	Availability Zone ▽	Health status ▽
☐	i-0781a01ba47ed709e [↗]	InService	t2.micro	ABC-scaling-template [↗] | Version 1	us-east-1b	Healthy
☐	i-0b1a71f4fc2dbe25a [↗]	InService	t2.micro	ABC-scaling-template [↗] | Version 1	us-east-1a	Healthy
☐	i-0f1ed9f65dddfe7d5 [↗]	InService	t2.micro	ABC-scaling-template [↗] | Version 1	us-east-1c	Healthy
☐	i-0f74cb477f276f0ce [↗]	InService	t2.micro	ABC-scaling-template [↗] | Version 1	us-east-1f	Healthy

Your architecture now looks like this:
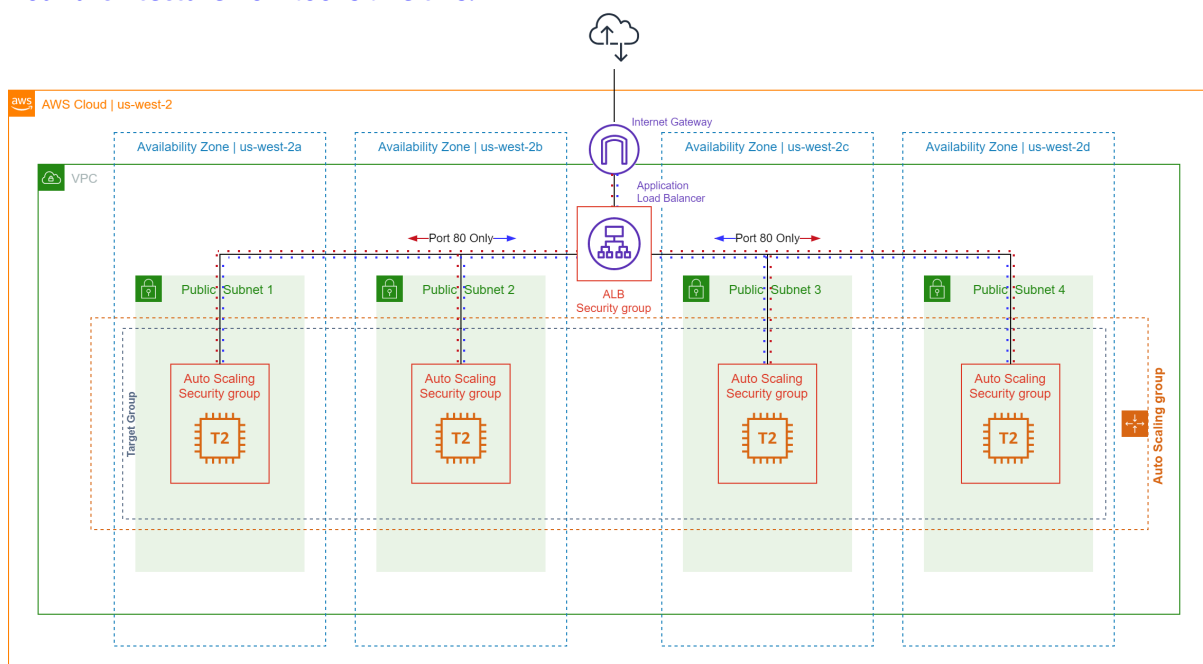
5. Once a number of new instances have successfully started (probably 3 or 4), repeatedly **refresh** your web-browser on you web host. You should now see the Instance ID, Availability Zone and Private IP change as the load balancer distributes the requests across the Auto Scaling group.

EC2 Instance Metadata	**EC2 Instance Metadata**	**EC2 Instance Metadata**
EC2 Instance ID: i-0dce1d899b5866510	EC2 Instance ID: i-0f6801f4e4b8ee4e8	EC2 Instance ID: i-0df21b362176d43fd
Availability Zone: us-west-2a	Availability Zone: us-west-2b	Availability Zone: us-west-2c
Private IP: 172.31.37.120	Private IP: 172.31.21.204	Private IP: 172.31.13.72

Congratulations! You have successfully created an EC2 Auto Scaling Group behind an Application Load Balancer.