

## **1. Part-1: EC2 with ELB and ASG**

**\*\*Objective\*\*:** Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

**\*\*Approach\*\*:**

**\*\*Launch EC2 Instances\*\*:** Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).

**\*\*Configure Load Balancer\*\*:** Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.

**\*\*Set Up Auto Scaling Group (ASG)\*\*:** Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.

**\*\*Test Your Setup\*\*:** Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.

**\*\*Verify Website Functionality\*\*:** Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations.

**\*\*Goal\*\*:** By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

## 1.1. Launch EC2 instance

Give name of instance and select application and OS Images

**Launch an instance** Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** Info

Name  
Web Server [Add additional tags](#)

**▼ Application and OS Images (Amazon Machine Image)** 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** **Quick Start**

**Amazon Linux**  **macOS**  **Ubuntu**  **Windows**  **Red Hat**  **SUSE Linux** 

 [Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

**Amazon Linux 2023 AMI** Free tier eligible 

ami-0e731c8a588258d0d (64-bit (x86), uefi-preferred) / ami-0bbebc09f0a12d4d9 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**  
Amazon Linux 2023 AMI 2023.3.20240205.2 x86\_64 HVM kernel-6.1

**Architecture** 64-bit (x86) **Boot mode** uefi-preferred **AMI ID** ami-0e731c8a588258d0d **Verified provider**

T2.micro instance type is selected.

The screenshot shows the 'Instance type' section of the AWS Lambda configuration interface. The 't2.micro' instance type is selected, indicated by a highlighted box. Key details shown include:

- Family: t2
- 1 vCPU
- 1 GiB Memory
- Current generation: true
- On-Demand Windows base pricing: 0.0162 USD per Hour
- On-Demand SUSE base pricing: 0.0116 USD per Hour
- On-Demand RHEL base pricing: 0.0716 USD per Hour
- On-Demand Linux base pricing: 0.0116 USD per Hour

To the right of the instance type details, there is a 'Free tier eligible' status indicator and a '▼' button. Below the instance type details, there is a note: 'Additional costs apply for AMIs with pre-installed software'. On the far right, there is a toggle switch labeled 'All generations' and a link to 'Compare instance types'.

Existing key pair is selected, which was created initially.

The screenshot shows the 'Key pair (login)' section of the AWS Lambda configuration interface. A key pair named 'key-pair' is selected, indicated by a highlighted box. The section includes a descriptive text: '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.' Below this, there is a 'Key pair name - required' label and a dropdown menu containing the value 'key-pair'. To the right of the dropdown, there is a 'Create new key pair' button with a circular arrow icon.

### ▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0d8fab51a5f972f19 (default) [Edit](#)

Subnet [Info](#)

No preference [Edit](#) [Create new subnet](#)

Auto-assign public IP [Info](#)

Enable [Edit](#)

Firewall (security groups) [Info](#)

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.

Create security group  Select existing security group

Security group name - required

web-ssh

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/()#,@[]+=&;!\$^\*

Description - required [Info](#)

web server ssh

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/8) [Remove](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>
ssh <a href="#">Edit</a>	TCP <a href="#">Edit</a>	22 <a href="#">Edit</a>
Source type <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Custom <a href="#">Edit</a>	<input type="text"/> Add CIDR, prefix list or security <a href="#">Edit</a>	e.g. SSH for admin desktop <a href="#">Edit</a>
	<a href="#">0.0.0.0/8</a> <a href="#">X</a>	

▼ Security group rule 2 (TCP, 80, 0.0.0.0/8) [Remove](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>
HTTP <a href="#">Edit</a>	TCP <a href="#">Edit</a>	80 <a href="#">Edit</a>
Source type <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Custom <a href="#">Edit</a>	<input type="text"/> Add CIDR, prefix list or security <a href="#">Edit</a>	e.g. SSH for admin desktop <a href="#">Edit</a>
	<a href="#">0.0.0.0/8</a> <a href="#">X</a>	

[Add security group rule](#)

At Advanced Detail section below script is attached and rest configuration are left as it is.

User data - *optional* | [Info](#)

Upload a file with your user data or enter it in the field.

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1> Hello from $(hostname -f)</h1>" > /var/www/html/index.html
```

User data has already been base64 encoded

**▼ Summary**

Number of instances | [Info](#)  
2

When launching more than 1 instance, [consider EC2 Auto Scaling](#)

**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.3.20240205.2 x86\_64  
HVM kernel-6.1  
ami-0e731c8a588258d0d

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
New security group

**Storage (volumes)**  
1 volume(s) - 8 GiB

**ⓘ Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Review commands](#)

Success  
Successfully initiated launch of instances [i-07856c4f5f3ea8e1, i-0626bcab4466a5eff]

Launch log

<input checked="" type="checkbox"/>	Web Server	i-0626bcab4466a5eff	<span>Running</span>	<span>Q</span>	<span>Q</span>	t2.micro	<span>2/2 checks passed</span>	<span>View alarms +</span>	us-east-1c	ec2-3-83-67-48.compute...	3.83.67.48	-	-	disabled	web-s
<input type="checkbox"/>	Web Server	i-07856c4f5f3ea8e1	<span>Running</span>	<span>Q</span>	<span>Q</span>	t2.micro	<span>2/2 checks passed</span>	<span>View alarms +</span>	us-east-1c	ec2-3-82-219-75.compute...	3.82.219.75	-	-	disabled	web-s

Ec2 instance with same configuration is created.

The created security group inbound rule is updated as the ec2 instance was not working while trying to run from Public IPv4 address of the created EC2 instance.

Edit inbound rules [info](#)  
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <a href="#">info</a>
Security group rule ID Type <a href="#">info</a> Protocol <a href="#">info</a> Port range <a href="#">info</a> Source <a href="#">info</a> Description - optional <a href="#">info</a>
sgr-0520e572668f19a80 HTTP TCP 80 Anywhere-IPv4 0.0.0.0/0

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel [Preview changes](#) [Save rules](#)

After updating inbound rule, the public IPv4 address of both ec2 instances worked in web browser.

Launch AWS Academy Learner | Instances | EC2 | us-east-1 | 52.90.51.229 | 54.161.84.181 | +

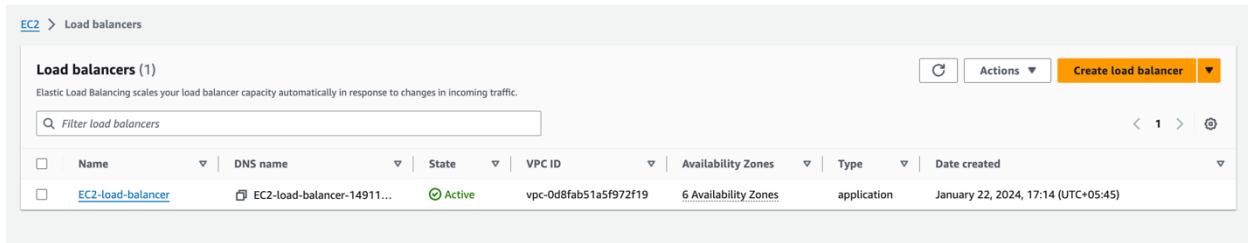
Hello from ip-172-31-83-149.ec2.internal

Launch AWS Academy Learner | Instances | EC2 | us-east-1 | 52.90.51.229 | 54.161.84.181 | +

Hello from ip-172-31-91-151.ec2.internal

## 1.2. Configure Load Balancer

Navigate to load balancers screen and click on “Create load balancer” and create a new load balancer.



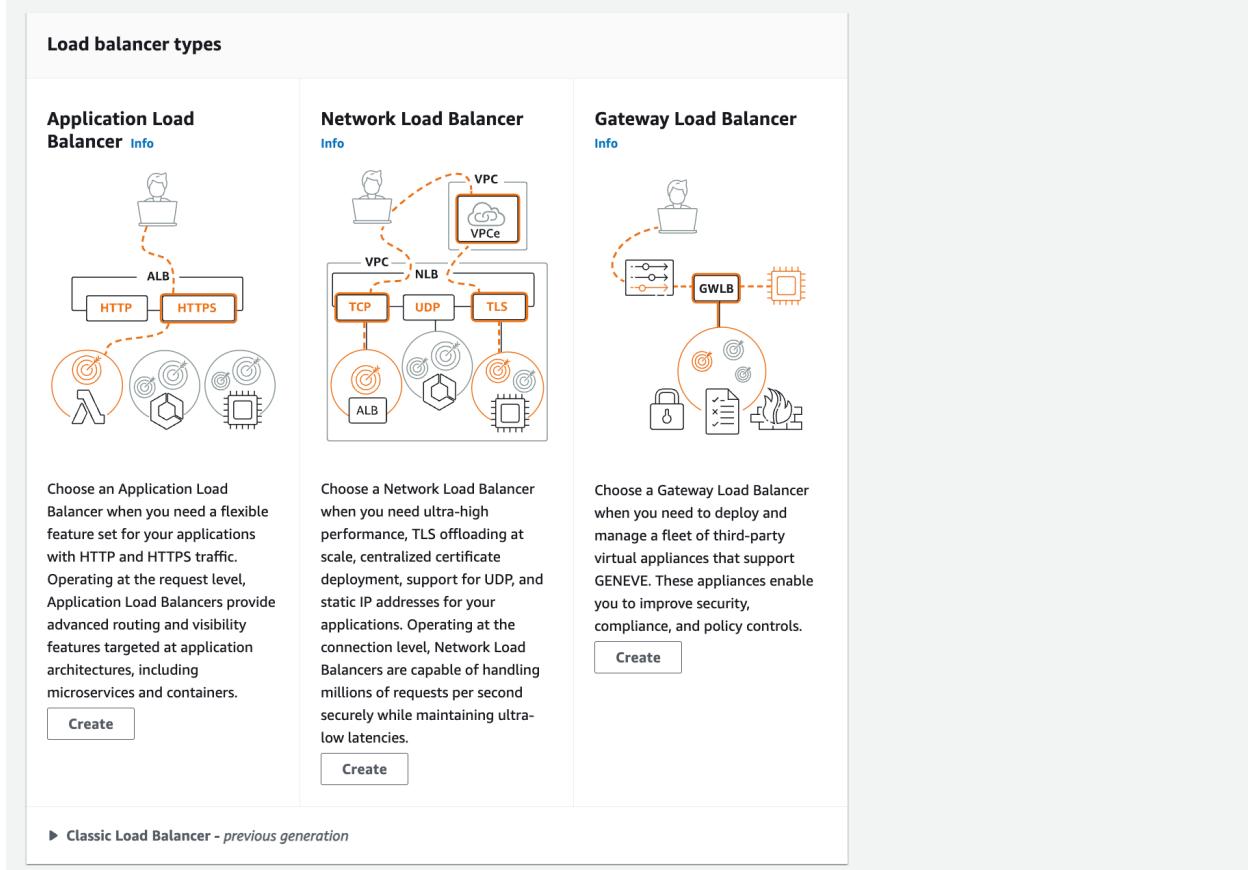
The screenshot shows the AWS EC2 Load Balancers page. At the top, there is a breadcrumb navigation: EC2 > Load balancers. Below the header, a table lists one load balancer entry:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
EC2-load-balancer	EC2-load-balancer-14911...	Active	vpc-0d8fab51a5f972f19	6 Availability Zones	application	January 22, 2024, 17:14 (UTC+05:45)

For this, lab Application Load balancer is created.

### Compare and select load balancer type

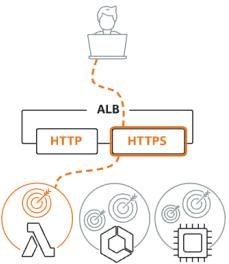
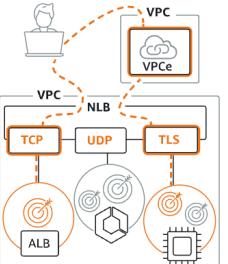
A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)



The chart compares three types of AWS Load Balancers:

- Application Load Balancer**: Handles HTTP and HTTPS traffic at the application layer. It routes requests to targets like Lambda functions, Amazon API Gateway, and Amazon EC2 instances.
- Network Load Balancer**: Handles TCP, UDP, and TLS traffic at the connection level. It supports VPC endpoints and can offload TLS to VPC endpoints.
- Gateway Load Balancer**: Manages a fleet of third-party virtual appliances supporting GENEVE. It handles security, compliance, and policy controls.

Below each section, there is a brief description and a "Create" button.

Load balancer types		
<b>Application Load Balancer</b> <a href="#">Info</a>  <p>Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers. <a href="#">Create</a></p>	<b>Network Load Balancer</b> <a href="#">Info</a>  <p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies. <a href="#">Create</a></p>	<b>Gateway Load Balancer</b> <a href="#">Info</a>  <p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls. <a href="#">Create</a></p>
<p>▶ <a href="#">Classic Load Balancer - previous generation</a></p>		

Required configuration are provided.

### Basic configuration

**Load balancer name**  
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** [Info](#)  
Scheme can't be changed after the load balancer is created.

**Internet-facing**  
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

**Internal**  
An internal load balancer routes requests from clients to targets using private IP addresses.

**IP address type** [Info](#)  
Select the type of IP addresses that your subnets use.

**IPv4**  
Recommended for internal load balancers.

**Dualstack**  
Includes IPv4 and IPv6 addresses.

Default VPC and all the available zone in mappings are selected.

### Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

**VPC** [Info](#)  
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

-  
vpc-0d8fab51a5f972f19  
IPv4: 172.31.0.0/16

**Mappings** [Info](#)  
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

**us-east-1a (use1-az6)**  
Subnet

IPv4 address  
Assigned by AWS

**us-east-1b (use1-az1)**  
Subnet

IPv4 address  
Assigned by AWS

**us-east-1c (use1-az2)**  
Subnet

IPv4 address  
Assigned by AWS

**us-east-1d (use1-az4)**  
Subnet

IPv4 address  
Assigned by AWS

Existing security group is selected.

The screenshot shows the 'Security groups' section of the AWS Lambda console. A message at the top states: 'A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group [Create new security group]'. Below this, a dropdown menu titled 'Select up to 5 security groups' contains one item: 'ec2-server-sg sg-0388661cf7577429c VPC: vpc-0d8fab51a5f972f19'. There is also a 'Clear' button next to the dropdown.

In Listeners and routing section, target group is to be selected. For this purpose, new target group is created.

### 1.2.1. Target group creation

Basic configuration for target group is provided. Here, target group instance is chosen and for rest default configuration are kept as it is.

The screenshot shows the 'Specify group details' step of creating a target group. The left sidebar shows 'Step 1 Specify group details' and 'Step 2 Register targets'. The main area is titled 'Specify group details' with the sub-section 'Basic configuration'. It says: 'Your load balancer routes requests to the targets in a target group and performs health checks on the targets.' Under 'Basic configuration', it says: 'Settings in this section can't be changed after the target group is created.' The 'Choose a target type' section has four options: 'Instances' (selected), 'IP addresses', 'Lambda function', and 'Application Load Balancer'. Each option has a list of benefits. Below this, the 'Target group name' field is filled with 'webserver-target-group'. A note says: 'A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.' At the bottom, the 'Protocol : Port' section shows 'HTTP' selected in a dropdown and '80' in a text input field. A note below says: 'Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.'

After that, target EC2 instances are registered in this newly created target group.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a navigation menu. Below it, a large orange button labeled 'Create Function' is prominent. To its right, there's a section for 'Lambda@Edge' with a 'Create' button. The main area is titled 'HelloWorld' and contains several tabs: 'Overview', 'Code', 'Logs', 'Metrics', 'Actions', and 'Configuration'. Under the 'Code' tab, there's a 'File' input field with a file icon, a 'Handler' dropdown set to 'index.handler', and a 'Runtime' dropdown set to 'Node.js 18.x'. A 'Create' button is located at the bottom right of this section.

Here, the created target group can be viewed in target group lists.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a navigation menu. Below it, a large orange button labeled 'Create Function' is prominent. To its right, there's a section for 'Lambda@Edge' with a 'Create' button. The main area is titled 'HelloWorld' and contains several tabs: 'Overview', 'Code', 'Logs', 'Metrics', 'Actions', and 'Configuration'. Under the 'Code' tab, there's a 'File' input field with a file icon, a 'Handler' dropdown set to 'index.handler', and a 'Runtime' dropdown set to 'Node.js 18.x'. A 'Create' button is located at the bottom right of this section.

Now, the load balancer configuration can be continued. In Listeners and routing section, recently created target group is selected.

**Listeners and routing** [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 [Remove](#)

Protocol	Port	Default action	Info
HTTP	: 80 1-65535	Forward to	webserver-target-group Target type: Instance, IPv4
<a href="#">Create target group</a>			

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

[Add listener tag](#)  
You can add up to 50 more tags.

[Add listener](#)

The summary of configuration is reviewed and after confirmation load balancer is created.

**Review**

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose [Create load balancer](#).

**Summary**  
Review and confirm your configurations. [Estimate cost](#)

<b>Basic configuration</b> <a href="#">Edit</a> load-balancer <ul style="list-style-type: none"> <li>Internet-facing</li> <li>IPv4</li> </ul>	<b>Security groups</b> <a href="#">Edit</a> <ul style="list-style-type: none"> <li>ec2-server-sg <a href="#">sg-0388661cf7577429c</a> </li> </ul>	<b>Network mapping</b> <a href="#">Edit</a> <ul style="list-style-type: none"> <li>VPC <a href="#">vpc-0d8fab51a5f972f19</a> <ul style="list-style-type: none"> <li>us-east-1a <a href="#">subnet-0bcf0a037de2b454c</a> </li> <li>us-east-1b <a href="#">subnet-08a6d516cb5ea1f23</a> </li> <li>us-east-1c <a href="#">subnet-0f7066271adf75ba</a> </li> <li>us-east-1d <a href="#">subnet-07dc58050f08fb06</a> </li> <li>us-east-1e <a href="#">subnet-04283ea4be42bf579</a> </li> <li>us-east-1f <a href="#">subnet-0ef6821697077fb03</a> </li> </ul> </li> </ul>	<b>Listeners and routing</b> <a href="#">Edit</a> <ul style="list-style-type: none"> <li>HTTP:80 defaults to <a href="#">webserver-target-group</a> </li> </ul>
<b>Service integrations</b> <a href="#">Edit</a> AWS WAF: <i>None</i> AWS Global Accelerator: <i>None</i>	<b>Tags</b> <a href="#">Edit</a> <i>None</i>		
<b>Attributes</b>			
<p> Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.</p>			

## Testing load balancer

After the created load balancer "load-balancer" is in active state,

Load balancers (2)								
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.								
<input type="text"/> Filter load balancers								
Name	DNS name	State	VPC ID	Availability Zones	Type	Date created		
<input type="checkbox"/> EC2-load-balancer	<input type="checkbox"/> EC2-load-balancer-14911...	<span>Active</span>	vpc-0d8fab51a5f972f19	6 Availability Zones	application	January 22, 2024, 17:14 (UTC+05:45)		
<input type="checkbox"/> load-balancer	<input type="checkbox"/> load-balancer-128156101...	<span>Active</span>	vpc-0d8fab51a5f972f19	6 Availability Zones	application	February 19, 2024, 20:02 (UTC+05:45)		

Security group was updated to another security group "web-ssh" which contains http inbound rule, so that the instance can accessed.

EC2 > Load balancers > load-balancer > Edit security groups

## Edit security groups

▶ Load balancer details: load-balancer

**Security groups**

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

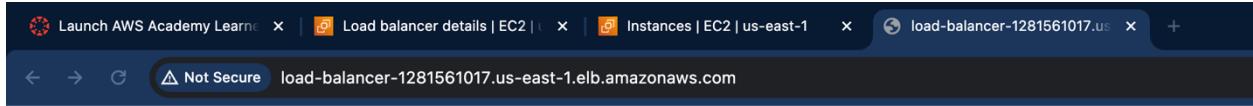
Security groups

Select up to 5 security groups

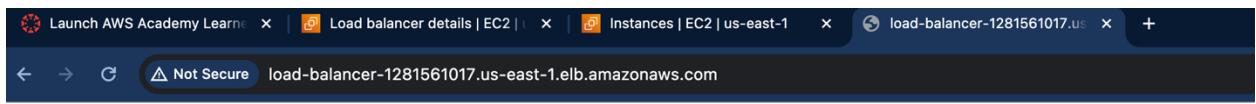
web-ssh  
sg-0b225ed36018a5d8f VPC: vpc-0d8fab51a5f972f19

Cancel Save changes

After that, the DNS name of load balancer is copied and accessed in web browser. When the URL is refreshed, multiple times it runs different instance of the specified target group to maintain the load.



**Hello from ip-172-31-91-151.ec2.internal**



**Hello from ip-172-31-83-149.ec2.internal**

### 1.3. Set up Auto Scaling Group

Create new auto scaling group.

The screenshot shows the 'Choose launch template' step of the EC2 Auto Scaling group creation wizard. On the left, there's a vertical sidebar with steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 - optional (Configure advanced options), and Step 4 - optional (Configure group size and scaling). The main area has a title 'Choose launch template' with an 'Info' link. It says 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.' Below this is a 'Name' input field containing 'WebserverASG'. Underneath it, a note says 'Auto Scaling group name' and 'Enter a name to identify the group.' A sub-note below the input field says 'Must be unique to this account in the current Region and no more than 255 characters.'

In launch template section, “Create a launch template” is selected and new template is created.

The screenshot shows the 'Launch template' step of the EC2 Auto Scaling group creation wizard. At the top, it says 'Launch template' with an 'Info' link. Below is a note: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' The main area has a 'Launch template' section with the sub-instruction 'Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.' It includes a dropdown menu labeled 'Select a launch template' with a downward arrow icon, a 'Create a launch template' button with a plus icon, and a 'C' icon. At the bottom right are 'Cancel' and 'Next' buttons.

### 1.3.1. Launch Template

Required configuration is given to create launch template.

[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*

WebServerTemplate

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

Template version description

A prod webserver for MyApp

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.

For Application and OS Images, currently in use is selected.

**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](#) | [Quick Start](#)

**Currently in use**

 [Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

al2023-ami-2023.3.20240117.0-kernel-6.1-x86\_64  
ami-0e9107ed11be76fde 2024-01-17T21:43:12.000Z Architecture: 64-bit (x86) Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

**Description**  
Amazon Linux 2023 AMI 2023.3.20240117.0 x86\_64 HVM kernel-6.1

Architecture x86\_64 AMI ID ami-0e9107ed11be76fde 

For instance type, t2.micro is selected and for key pair, existing key pair is selected.

▼ **Instance type** [Info](#) | [Get advice](#) [Advanced](#)

**Instance type**

**t2.micro** Family: t2 1 vCPU 1 GiB Memory Current generation: true Free tier eligible  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.0716 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations [Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

▼ **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  

In network setting section existing security group is selected and other configurations are kept as default.

The screenshot shows the 'Network settings' section of the AWS Lambda configuration interface. It includes fields for Subnet (set to 'Don't include in launch template'), Firewall (security groups) (with 'Select existing security group' selected), and Common security groups (listing 'web-ssh sg-0b225ed36018a5d8f'). A 'Create new subnet' button is also present.

**Network settings** [Info](#)

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) [Info](#)

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

Common security groups [Info](#)

Select security groups [▼](#)

web-ssh sg-0b225ed36018a5d8f [X](#) [Compare security group rules](#)

VPC: vpc-0d8fab51a5f972f19

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

Bash script is added in advanced detail section.

The screenshot shows the 'User data - optional' section of the AWS Lambda configuration interface. It contains a file upload field ('Choose file') with a sample bash script pasted into it. A checkbox at the bottom indicates that the user data has been base64 encoded.

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1> Hello from $(hostname -f)</h1>" > /var/www/html/index.html
```

User data has already been base64 encoded

After confirming the configuration, template is launched .

▼ **Summary**

**Software Image (AMI)**  
Amazon Linux 2023.3.2...[read more](#)  
ami-0e9107ed11be76fde

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
web-ssh

**Storage (volumes)**  
1 volume(s) - 8 GiB

**ⓘ Free tier:** In your first year includes X  
750 hours of t2.micro (or t3.micro in  
the Regions in which t2.micro is  
unavailable) instance usage on free  
tier AMIs per month, 30 GiB of EBS  
storage, 2 million IOs, 1 GB of  
snapshots, and 100 GB of bandwidth  
to the internet.

**Cancel** **Create launch template**

EC2 > [Launch templates](#) > Create launch template



Success

Successfully created WebServerTemplate(lt-0d66b8fd69de3d24e).

Now, the configuration of auto scaling is continued. Here newly created launch template is selected.

**Launch template** [Info](#)

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

**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.

WebServerTemplate [C](#)

[Create a launch template](#)

**Version**  
Default (1) [C](#)

[Create a launch template version](#)

Description	Launch template	Instance type
-	<a href="#">WebServerTemplate</a> lt-0d66b8fd69de3d24e	t2.micro
AMI ID ami-0e9107ed11be76fde	Security groups -	Request Spot Instances No
Key pair name key-pair	Security group IDs <a href="#">sg-0b225ed36018a5d8f</a>	

**Additional details**

Storage (volumes)	Date created
-	Mon Feb 19 2024 21:50:44 GMT+0545 (Nepal Time)

[Cancel](#) [Next](#)

In network section default VPC and all the subnets are selected

**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**  
Choose the VPC that defines the virtual network for your Auto Scaling group.  
vpc-0ddfbab51a5f972f19 [C](#)  
172.31.0.0/16 Default

[Create a VPC](#)

**Availability Zones and subnets**  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.  
[Select Availability Zones and subnets](#) [C](#)

us-east-1a   subnet-0bcf0a037de2b454d 172.31.32.0/20 Default
us-east-1b   subnet-08a6d516cb5ea1f23 172.31.0.0/20 Default
us-east-1c   subnet-0ff7066271adf75ba 172.31.80.0/20 Default
us-east-1d   subnet-07dc58050f08fba06 172.31.16.0/20 Default
us-east-1e   subnet-04283ea4be42bf579 172.31.48.0/20 Default
us-east-1f   subnet-0ef6821697077fb03 172.31.64.0/20 Default

[Create a subnet](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

Advanced options configurations is done. Firstly, in load balancing configuration, existing load balancer is chosen and then it's target group is selected. Rest advanced configuration is kept as default.

The screenshot shows the 'Configure advanced options - optional' step of the AWS Auto Scaling wizard. On the left, a vertical sidebar lists steps from 1 to 7. Step 1: 'Choose launch template'. Step 2: 'Choose instance launch options'. Step 3 (optional): 'Configure advanced options' (selected). Step 4 (optional): 'Configure group size and scaling'. Step 5 (optional): 'Add notifications'. Step 6 (optional): 'Add tags'. Step 7: 'Review'. The main content area is titled 'Load balancing' and contains the following information:

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

**Load balancing** 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 an existing load balancer**

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups  
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

**Existing load balancer target groups**  
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

webserver-target-group | HTTP   
Application Load Balancer: load-balancer

Now, group size and scaling are configured. Here, desired capacity of Group size is set as 2. In scaling, min desired capacity is set as 2 and max desired capacity as 4. After reviewing the configuration auto scaling group is created.

Instances (11) <a href="#">Info</a>										
<a href="#">Find Instance by attribute or tag (case-sensitive)</a> <span style="float: right;">Any state </span>										
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elast.
<input type="checkbox"/>		i-05c590a0937214db7	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1f	-	-	-
<input type="checkbox"/>	Ec2 server	i-060fd3a281f02e32c	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1d	ec2-44-196-3-52.comp...	44.196.3.52	44.1
<input type="checkbox"/>		i-09a79f4a213c103ef	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1a	ec2-3-80-166-29.comp...	3.80.166.29	-
<input type="checkbox"/>	EC2 Server	i-0d63dcae5732580d2	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-44-202-71-135.co...	44.202.71.135	-
<input type="checkbox"/>	EC2 Server	i-022827d5a2ae2738	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-3-94-61-79.comput...	3.94.61.79	-
<input type="checkbox"/>	Web Server	i-0626bcab4466a5eff	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-44-211-247-237.co...	44.211.247.237	-
<input type="checkbox"/>	Web Server	i-07856c4f5f3ee8e1	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-3-87-201-137.com...	3.87.201.137	-
<input type="checkbox"/>		i-062e881785d3e795d	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1c	ec2-3-95-165-56.comp...	3.95.165.56	-
<input type="checkbox"/>	my server	i-06bcfc1a46016a8eb	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1a	ec2-18-232-72-219.co...	18.232.72.219	-

Select an instance

As we have defined 2 desired capacity, 2 EC2 instance of same target group is always maintained based on the created launch template. In the activity tab of auto scaling, we can observe the activity history of the EC2 instances.

Connection draining in progress	At 2024-02-19T16:29:57Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2024 February 19, 10:14:57 PM +05:45
Successful	Launching a new EC2 instance: i-0007c56f0a9db5ca7 At 2024-02-19T16:28:52Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2024 February 19, 10:15:54 PM +05:45      2024 February 19, 10:14:26 PM +05:45
Cancelled	Launching a new EC2 instance: i-011cb2604f22e40ee. Status Reason: Instance became unhealthy while waiting for instance to be in InService state. Termination Reason: ClientUserInitiatedShutdown own: User initiated shutdown	At 2024-02-19T16:26:15Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2. 2024 February 19, 10:11:18 PM +05:45      2024 February 19, 10:11:49 PM +05:45
Cancelled	Launching a new EC2 instance: i-062e881785d3e795d. Status Reason: Instance became unhealthy while waiting for instance to be in InService state. Termination Reason: ClientUserInitiatedShutdown own: User initiated shutdown	At 2024-02-19T16:24:39Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2. 2024 February 19, 10:09:41 PM +05:45      2024 February 19, 10:10:12 PM +05:45
Cancelled	Launching a new EC2 instance: i-05c590a0937214db7. Status Reason: Instance became unhealthy while waiting for instance to be in InService state. Termination Reason: ClientUserInitiatedShutdown own: User initiated shutdown	At 2024-02-19T16:23:44Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2024-02-19T16:23:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2. 2024 February 19, 10:08:59 PM +05:45      2024 February 19, 10:09:30 PM +05:45
Successful	Launching a new EC2 instance: i-09a79f4a213c103ef At 2024-02-19T16:23:44Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2024-02-19T16:23:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2024 February 19, 10:08:58 PM +05:45      2024 February 19, 10:09:31 PM +05:45

When DNS of load balancer is refreshed even for multiple times the, it can be observed that EC2 instance is working.

← → ⚡ Not Secure load-balancer-1281561017.us-east-1.elb.amazonaws.com

## Hello from ip-172-31-91-151.ec2.internal

← → ⚡ Not Secure load-balancer-1281561017.us-east-1.elb.amazonaws.com

## Hello from ip-172-31-43-204.ec2.internal

← → ⚡ Not Secure load-balancer-1281561017.us-east-1.elb.amazonaws.com

## Hello from ip-172-31-83-149.ec2.internal

It can be observed that, how auto scaling adds and removes EC2 instances to manage the traffic.

Instances (17) Info												
Find Instance by attribute or tag (case-sensitive)				Actions								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring
Web Server	i-0626bcab4466a5eff	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-44-211-247-237.co...	44.211.247.237	-	-	-	disabled
	i-05e590a0937214db7	Terminated	t2.micro	-	View alarms +	us-east-1f	-	-	-	-	-	disabled
Ec2 server	i-060fd3a281f02e32c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-196-3-52.comp...	44.196.3.52	44.196.3.52	-	-	disabled
	i-03021fe25a105894a	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-	-	-	-	disabled
	i-06c5bf91a5ddff68e2	Terminated	t2.micro	-	View alarms +	us-east-1f	-	-	-	-	-	disabled
	i-09a79f4a213c103ef	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-80-166-29.comp...	3.80.166.29	-	-	-	disabled
	i-011cb2604722e40e6	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	-	disabled
EC2 Server	i-0d63dcae5732580d2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-44-202-71-135.co...	44.202.71.135	-	-	-	disabled
EC2 Server	i-022827d3a2aeef2738	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-3-94-61-79.comput...	3.94.61.79	-	-	-	disabled
Web Server	i-07856c4f5f5eeea8e1	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-3-87-201-137.com...	3.87.201.137	-	-	-	disabled
	i-000756f0a9fdb3c7	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	-	disabled
	i-0baeb15ccb6d356778	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	-	disabled
	i-062e81785d3e795d	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	-	disabled
my server	i-06bcfc1a46016a8eb	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-18-232-72-219.co...	18.232.72.219	-	-	-	disabled
my server	i-006f2305f86394850	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-161-13-240.co...	54.161.13.240	-	-	-	disabled

## **Part 2: Hosting a Static Portfolio Website on S3**

**Objective:** Learn to host a static website (such as a personal portfolio) on Amazon S3.

**Approach:**

1. **Create an S3 Bucket:** Start by creating a new S3 bucket. Configure the bucket for website hosting, which includes setting permissions to make the content publicly accessible.
2. **Upload Website Files:** Upload the static files of your portfolio website (HTML, CSS, JavaScript, images) to the S3 bucket.
3. **Configure DNS:** Use Amazon Route 53 or another DNS service to point a domain name to the S3 bucket. This makes the website accessible via a user-friendly URL.
4. **Enable Additional Features (Optional):** Implement features like HTTPS for secure access and CloudFront for content delivery optimization.

**Goal:** Students will understand how to use S3 for hosting static websites, manage bucket permissions, and integrate with other AWS services for a complete web hosting solution.

## Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#)

### General configuration

#### AWS Region

US East (N. Virginia) us-east-1 ▾

#### Bucket type Info

General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory - New

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

#### Bucket name Info

www.basic-s3bucket.com

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

#### Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

### Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership

## Edit static website hosting Info

### Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#) 

#### Static website hosting

- Disable
- Enable

#### Hosting type

- Host a static website  
Use the bucket endpoint as the web address. [Learn more](#) 
- Redirect requests for an object  
Redirect requests to another bucket or domain. [Learn more](#) 

#### Host name

Target bucket website address or personal domain

#### Protocol - *Optional*

- none
- http
- https

[Cancel](#)

[Save changes](#)

## Create hosted zone [Info](#)

### Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

#### Domain name [Info](#)

This is the name of the domain that you want to route traffic for.

basic-s3bucket

Valid characters: a-z, 0-9, !"#\$%&'()\*+,-/:;<=>?@[\]^\_`{|}.~

#### Description - *optional* [Info](#)

This value lets you distinguish hosted zones that have the same name.

*The hosted zone is used for...*

The description can have up to 256 characters. 0/256

#### Type [Info](#)

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

**Public hosted zone**

A public hosted zone determines how traffic is routed on the internet.

**Private hosted zone**

A private hosted zone determines how traffic is routed within an Amazon VPC.

### Tags [Info](#)

Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.

[Add tag](#)

You can add up to 50 more tags.

[Cancel](#)

[Create hosted zone](#)

## Get started Info

### Choose your starting point

#### Register a domain

Register the name, such as example.com, that your users use to access your application.



#### Transfer domain

You can transfer domain names to Route 53 that you registered with another domain registrar.



#### Create hosted zones

A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.



#### Configure health checks

Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.



#### Configure traffic flow

A visual tool that lets you easily create policies for multiple endpoints in complex configurations.



#### Configure resolvers

A regional service that lets you route DNS queries between your VPCs and your network.



Cancel

Get started