

## Load Balancer and Path-Based Routing in AWS

Imagine you are working for JJTech e-commerce division, which has two primary applications: Orders and Payments. As the company grows, the traffic to these applications increases, and it becomes essential to ensure that the application can handle large volumes of traffic efficiently.

To achieve this, you decide to implement an Application Load Balancer (ALB) with path-based routing to distribute the traffic between the two applications, ensuring that requests are directed to the appropriate service.

### Goals:

**Efficient Traffic Distribution:** Direct traffic to the correct service based on the URL path.

This runbook provides detailed steps to simulate load balancing and path-based routing using an Application Load Balancer (ALB) in AWS. We will create two EC2 instances, set up target groups, create an ALB, and configure listener rules for path-based routing.

### 1. Create Instances with User Data

In this step, we will launch two EC2 instances with user data to serve different application paths. One instance will handle 'orders' and the other will handle 'payments'.

#### Instance for Orders

1. Open the Amazon EC2 console.
2. Choose 'Launch Instance'.
3. Select 'Amazon Linux 2 AMI'.
4. Choose an instance type and configure instance details.
  - name: order-instance
  - putting instances in VPC with subnets and connectivity to the internet
  - configure SG for instance ( allow http for internet)

5. In the 'Advanced Details' section, add the following user data:

```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
mkdir /var/www/html/orders/
echo "<h1>This page is to receive all orders</h1>" > /var/www/html/orders/index.html
```

### Instance for Payments

1. Open the Amazon EC2 console.
2. Choose 'Launch Instance'.
3. Select 'Amazon Linux 2 AMI'.
4. Choose an instance type and configure instance details.
  - name: payment-instance
  - putting instances in VPC with subnets and connectivity to the internet
  - configure SG for instance ( allow http for internet)
5. In the 'Advanced Details' section, add the following user data:

```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
mkdir /var/www/html/payments/
echo "<h1>This page is to receive all payments</h1>" >
/var/www/html/payments/index.html
```

## 2. Create Target Groups

Next, we will create two target groups. One for the orders instance and the other for the payments instance.

1. Open the Amazon EC2 console.
2. In the left navigation pane, choose '**Target Groups**'.
3. Choose '**Create target group**'.
4. Configure the target group for 'orders'
  - Target type: Instances
  - Target group name: '**orders**' for the order Target Group and '**payments**' for payment Target Groups

## Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

### Basic configuration

Settings in this section can't be changed after the target group is created.

#### Choose a target type

☒ Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

☐ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application

- Protocol: HTTP Port: 80
- IP address type: IPv4

#### Target group name

orders

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

#### Protocol : Port

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

HTTP

80

1-65535

#### IP address type

Only targets with the indicated IP address type can be registered to this target group.

☒ IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

#### VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

defaultVPC

vpc-d0c9adad

IPv4 VPC CIDR: 172.31.0.0/16

#### Protocol version

☒ HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

- Health check path: /orders/

### Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP ▼

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/orders/

Up to 1024 characters allowed.

► Advanced health check settings

5. Choose **'Next'** and register the orders instance with the target group.

- select the orders instance and
- click on **Include as pending below**

### Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (1/2)

Filter instances

	Instance ID	Name	State	Security groups
<input type="checkbox"/>	i-06a2978d2459873bf	payment-instance	Running	asg-template-sg
<input checked="" type="checkbox"/>	i-0c4eecf6decc7e4ef	Orders-instance	Running	asg-template-sg

1 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80


1-65535 (separate multiple ports with commas)



Include as pending below






6. Then click on **Create Target group**. Once the target is registered you should see a healthy target

EC2 > [Target groups](#) > orders

## orders

**Details**  
 arn:aws:elasticloadbalancing:us-east-1:945685952191:targetgroup/orders/ec76edfd20e475

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC <a href="#">vpc-d0c9adad</a> 
IP address type IPv4	Load balancer <a href="#">testALB</a> 		

1 Total targets	 1 Healthy 0 Anomalous	 0 Unhealthy	 0 Unused	 0 Initial	 0 Draining
--------------------	---	--	---	--	---

7. Repeat the process to create a target group for 'payments':

- Target group name: payments
- Health check path: /payments/

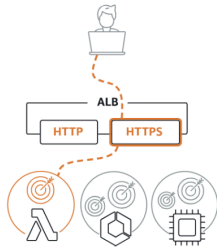
### 3. Create an Application Load Balancer (ALB)

Now we will create an ALB to distribute traffic to the target groups based on the path.

1. Open the Amazon EC2 console.
2. In the left navigation pane, choose '**Load Balancers**'.
3. Choose '**Create Load Balancer**'. This opens a new page with the different load balancer types
4. Select '**Application Load Balancer**' and click on create

## Load balancer types

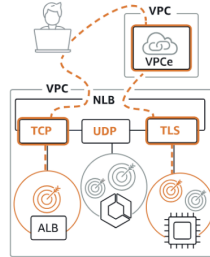
### Application Load Balancer [Info](#)



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.

Create

### Network Load Balancer [Info](#)



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.

Create

### Gateway Load Balancer [Info](#)



Choose a Gateway Load Balance 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.

Create

## 5. Configure the load balancer settings:

- Name: myALB
- Scheme: Internet-facing
- IP address type: IPv4

### Basic configuration

#### Load balancer name

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

ijtechALB

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. Compatible with the IPv4 and Dualstack IP address types.

#### Load balancer IP address type [Info](#)

Select the type of IP addresses that your subnets use. Public IPv4 addresses have an additional cost.

☒ IPv4

Includes only IPv4 addresses.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

☐ Dualstack without public IPv4

Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only.

- Listeners: HTTP
- VPC: Select your VPC and enable AZs for the loadbalancer by selecting subnets in the AZs
- Subnets: Select two or more subnets

### 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. To confirm the VPC for your targets, view your [target groups](#).

defaultVPC  
vpc-d0c9adad  
IPv4 VPC CIDR: 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 or the VPC are not available for selection.

##### ☒ us-east-1a (use1-az2)

Subnet

subnet-e0dc8ac1

IPv4 address

Assigned by AWS

##### ☒ us-east-1b (use1-az4)

Subnet

subnet-61b9a22c

IPv4 address

Assigned by AWS

##### ☒ us-east-1c (use1-az6)

Subnet

subnet-562f7b09

IPv4 address

## 6. Configure security groups and routing settings.

- open port 80 on security group.

#### Security groups

Select up to 5 security groups

testALB-sg  
sg-0f26873a5fcfaef9 VPC: vpc-d0c9adad

### 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 traffic to its registered targets.

#### ▼ Listener HTTP:80

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

[Create target group](#)

#### Listener tags - optional

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

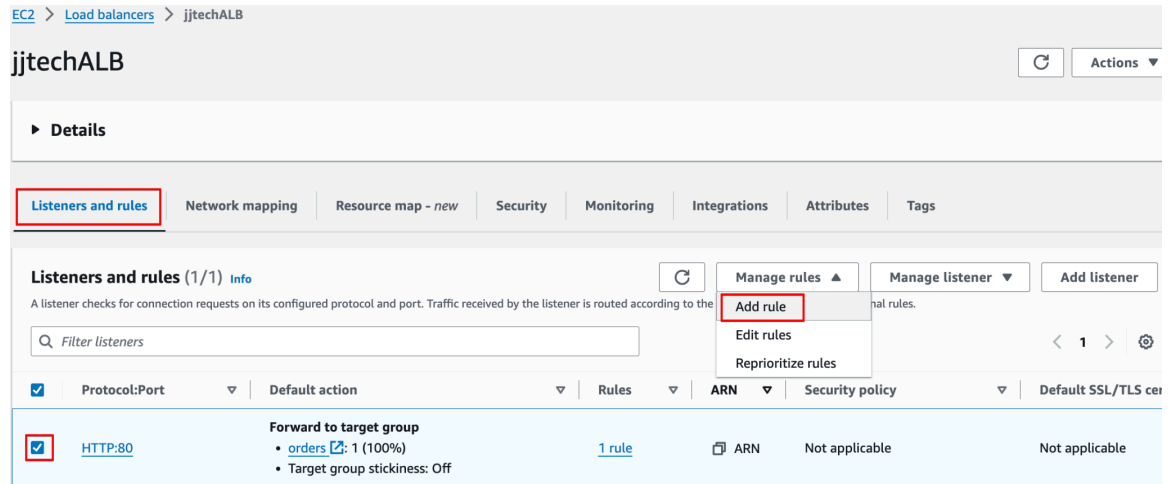
## 7. Leave other settings as default and click on **create load balancer**.

## 8. Skip the target registration and choose 'Create'.

## 4. Update Listener Rules

In this step, we will update the listener rules to route traffic based on the URL path.

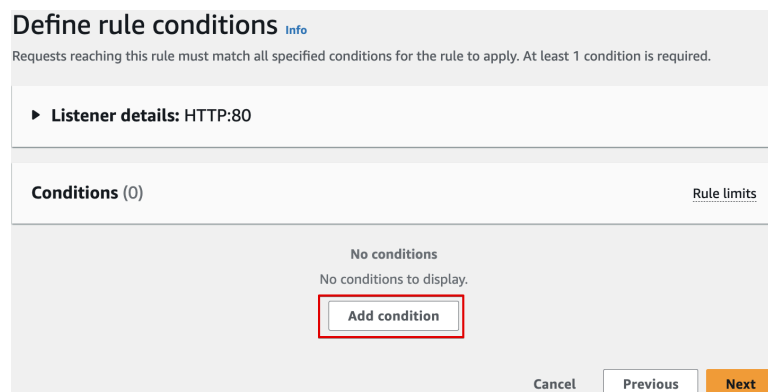
1. Open the Amazon EC2 console.
2. In the left navigation pane, choose 'Load Balancers'.
3. Select the ALB 'jjtechALB'.
4. Choose the '**Listeners and rules**' tab.
5. Select the HTTP listener and choose '**Manage rules**' and click on **Add rule**.



Update the listener rules to route traffic based on path:

A) If the URL path is /orders/, forward to the 'orders' target group.

- name the rule, **OrdersRule** (for the routing the the orders backend server) and click on Next
- Define rule conditions: click on Add condition



- choose the **Path** condition from the drop-down menu, and enter the path to the orders application **/orders/** and click on **confirm**



Add condition
Rule limits

Rule condition types

Route traffic based on the condition type of each request. Each rule can include one of each of the following conditions: host-header, path, http-request-method and source-ip. Each rule can include one or more of each of the following conditions: http-header and query-string.

Path

Path

Define the path. For example: /item/\*. Case sensitive.

is /orders/

Maximum 128 characters. Allowed characters are [a-z], [A-Z], [0-9]; the following special characters: [\_.\$/~\*'@:~]; [&] (using [amp;]); and wildcards ([\*] and [?]).

Add new value

You can add up to 4 more condition values for this rule.

Cancel Confirm

- click on **Next** and Define the rule actions

Define rule actions
Info

These actions will be applied to requests matching the rule conditions.

▶ Listener details: HTTP:80

Actions

Action types

Routing actions

☒ Forward to target groups
☐ Redirect to URL
☐ Return fixed response

Forward to target group Info

Choose a target group and specify routing weight or [Create target group](#).

Target group

orders
Target type: Instance, IPv4

HTTP

Weight
Percent

1
100%

0-999

Add target group

You can add up to 4 more target groups.

Target group stickiness Info

Stickiness enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies.

☐ Turn on target group stickiness

Cancel Previous Next

- set rule priority to 1
- review and create the rule.

B) If the URL path is **/payments/**, forward to the 'payments' target group.

repeat all the steps in (A) above using

- rule name: PaymentRule

- path: /payments/
- rule action: forward to target group payments (created above)

7. Save the changes.

## 5. Test the Load Balancer

Finally, we will test the ALB to ensure that traffic is routed correctly.

1. Obtain the DNS name of the ALB from the EC2 console.

The screenshot shows the AWS Management Console for the 'jjtechALB' load balancer. The 'Details' section is expanded, showing various attributes. The 'DNS name' is highlighted with a red box and a red arrow pointing to it, with a note 'will be different for each account'.

jjtechALB			
▼ Details			
Load balancer type Application	Status Active	VPC vpc-d0c9adad	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-61b9a22c us-east-1b (use1-az4) subnet-562f7b09 us-east-1c (use1-az6) subnet-e0dc8ac1 us-east-1a (use1-az2)	Date created July 12, 2024, 23:13 (UTC+02:00)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:945685952191:loadbalancer/app/jjtechALB/a2b32f348cb8b401		DNS name jjtechALB-1724398090.us-east-1.elb.amazonaws.com (A Record)	

2. Open a web browser and test the following URLs:

- http://<ALB-DNS>/orders/ (This should display the orders page)

e.g **http://jjtechALB-1724398090.us-east-1.elb.amazonaws.com/orders/**

- http://<ALB-DNS>/payments/ (This should display the payments page)

e.g **http://jjtechALB-1724398090.us-east-1.elb.amazonaws.com/payments/**

This shows that the traffic is routed via the ELB to the appropriate instances based on the path.