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:

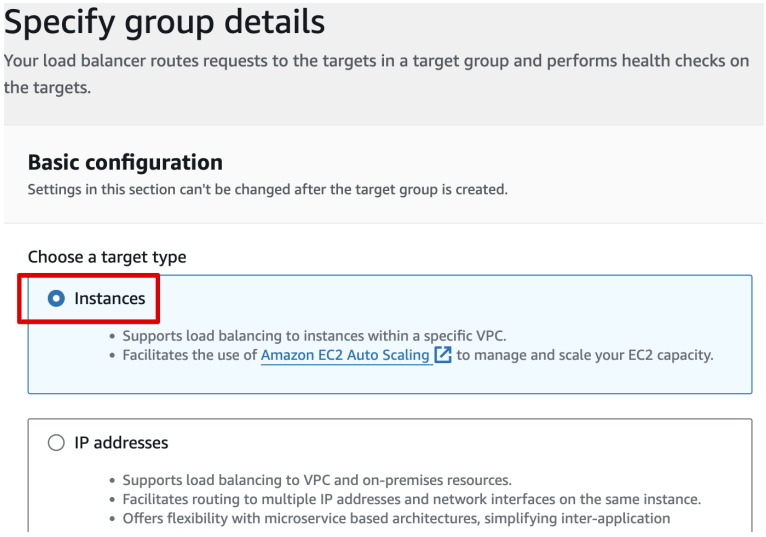
```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
mkdir /var/www/html/payments/
echo "<h1>This page is to receive all payments</h1>" >
/var/www/html/payments/index.html
```

2. Create Target Groups

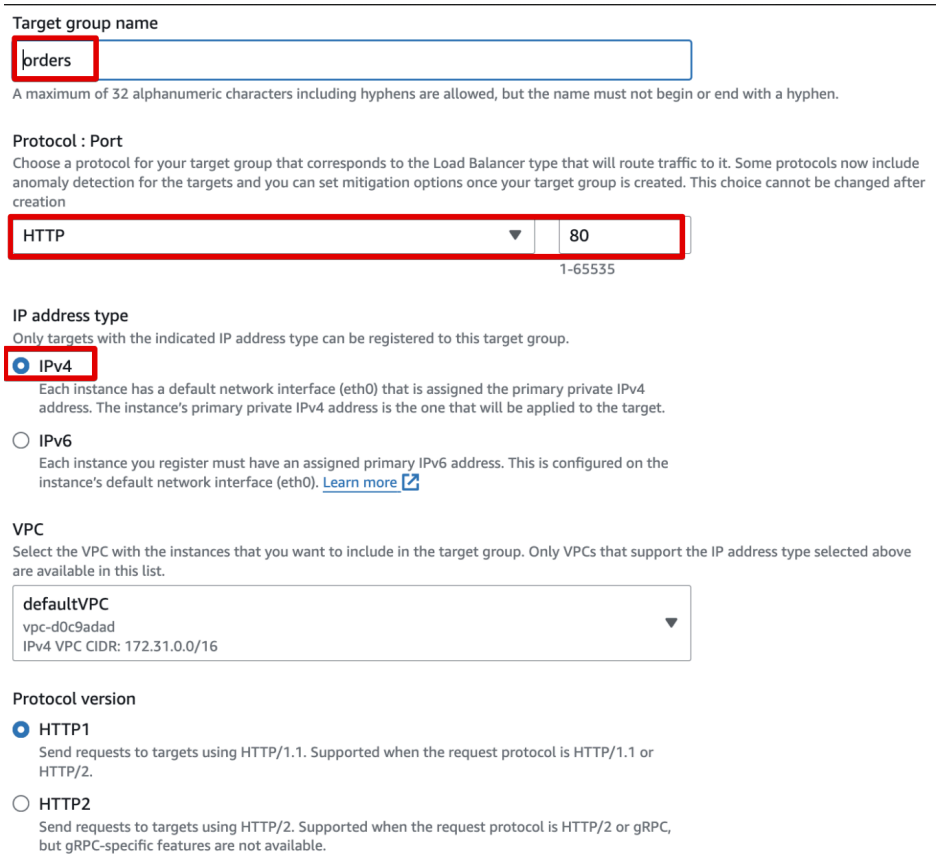
Next, we will create two target groups. One for the orders instance and the other for the payments instance.

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'

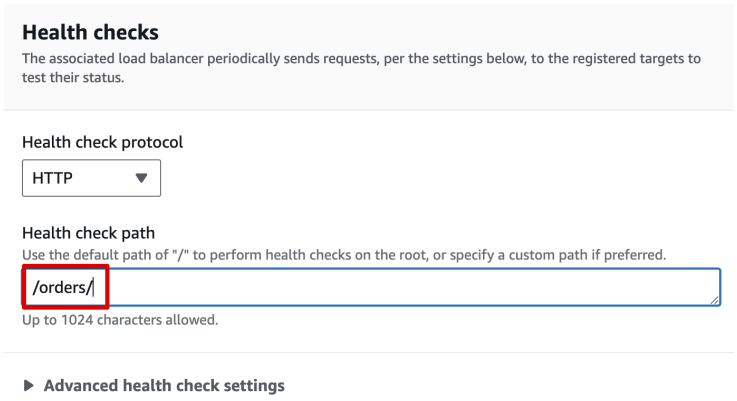
- Target type: Instances
- Target group name: '**orders'** for the order Target Group and '**payments'** for payment Target Groups



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

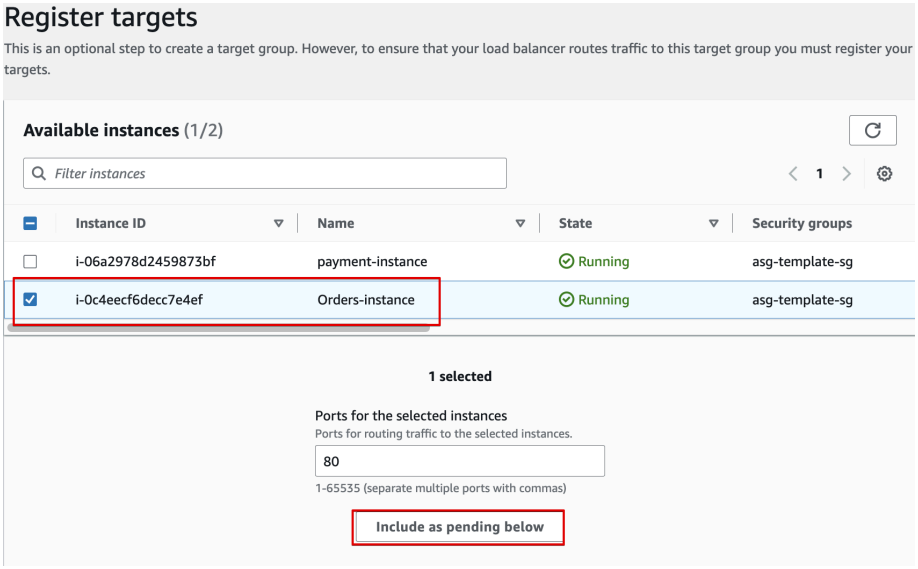


- Health check path: /orders/

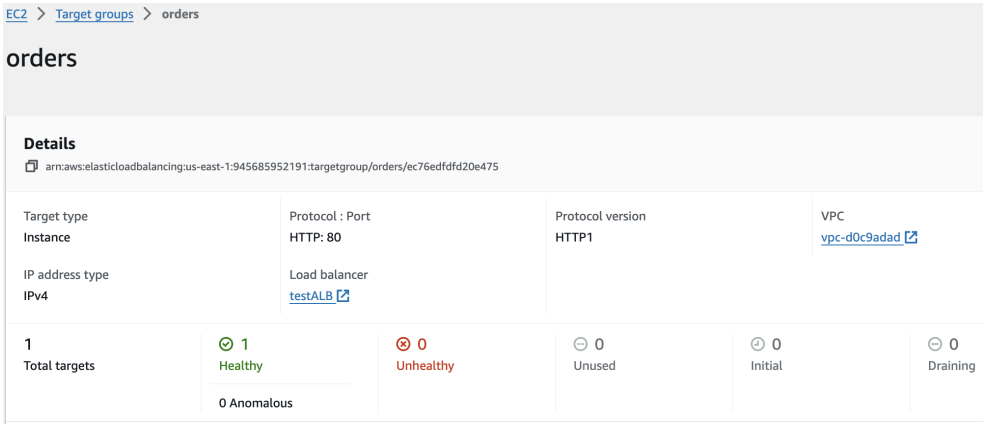


5. Choose '**Next**' and register the orders instance with the target group.

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



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



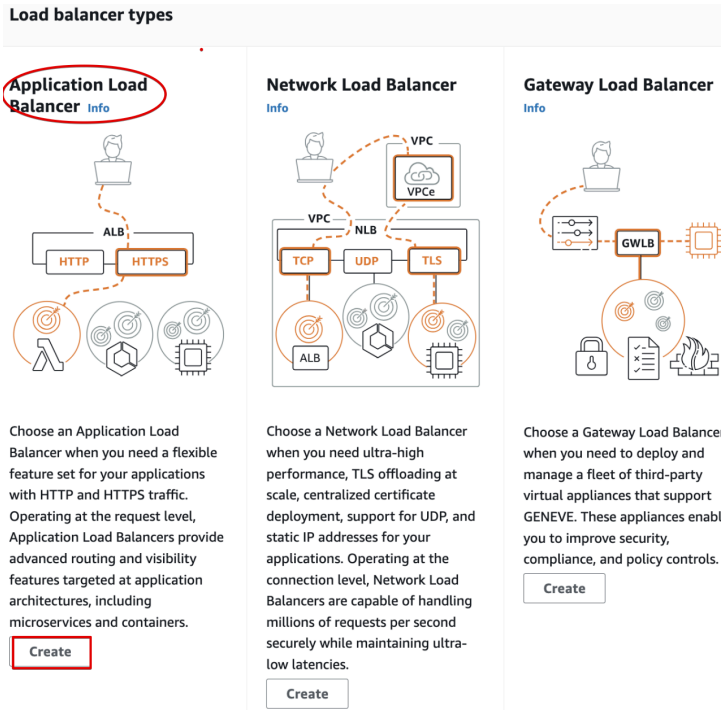
7. Repeat the process to create a target group for 'payments':

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

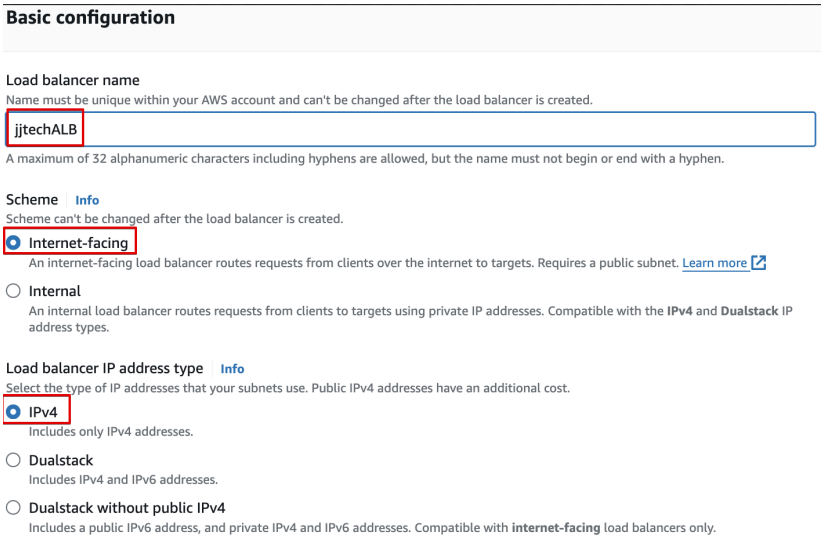
3. Create an Application Load Balancer (ALB)

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

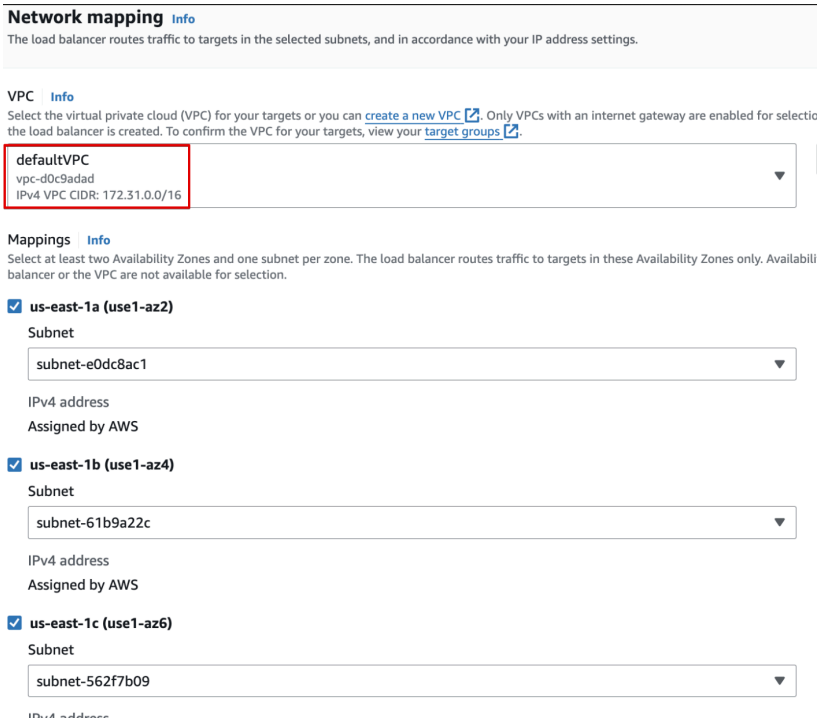
1. Open the Amazon EC2 console.
2. In the left navigation pane, choose '**Load Balancers**'.
3. Choose '**Create Load Balancer**'. This open a new page with the different load balancer types
4. Select '**Application Load Balancer**' and click on create



5. Configure the load balancer settings:
   - Name: myALB
   - Scheme: Internet-facing
   - IP address type: IPv4

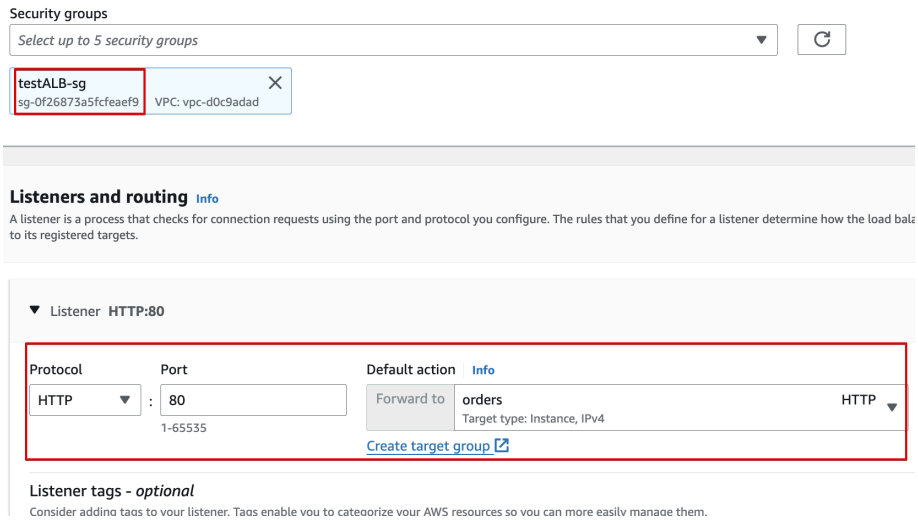


   - Listeners: HTTP
   - VPC: Select your VPC and enable AZs for the loadbalancer by selecting subnets in the AZs
   - Subnets: Select two or more subnets



6. Configure security groups and routing settings.

- open port 80 on security group.



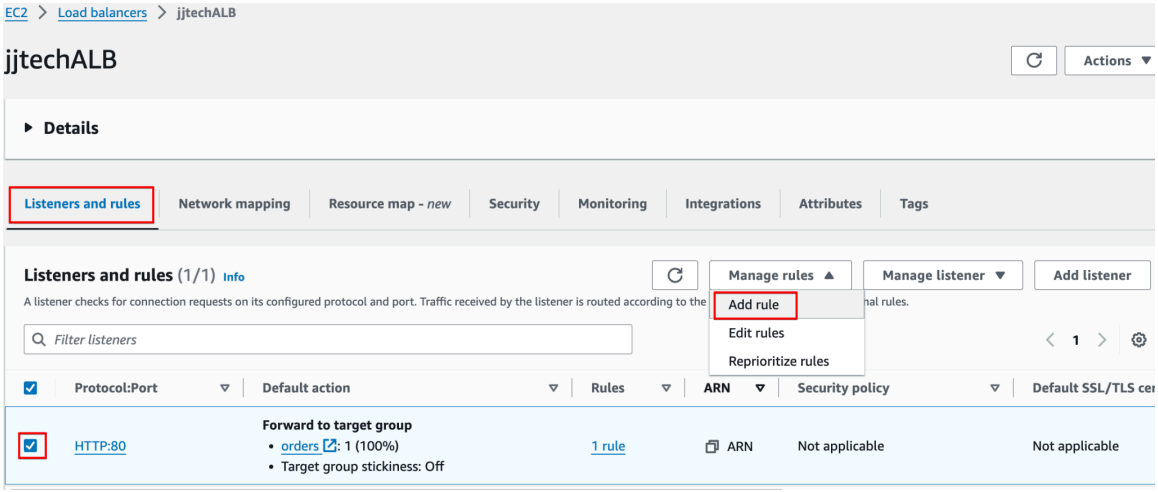
7. Leave other settings as default and click on **create load balancer**.

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.

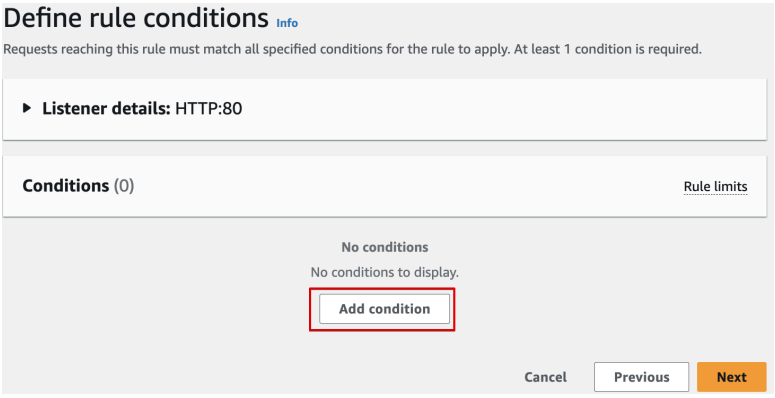
1. Open the Amazon EC2 console.
2. In the left navigation pane, choose 'Load Balancers'.
3. Select the ALB 'jjtechALB'.
4. Choose the '**Listeners and rules**' tab.
5. Select the HTTP listener and choose '**Manage rules**' and click on **Add rule**.



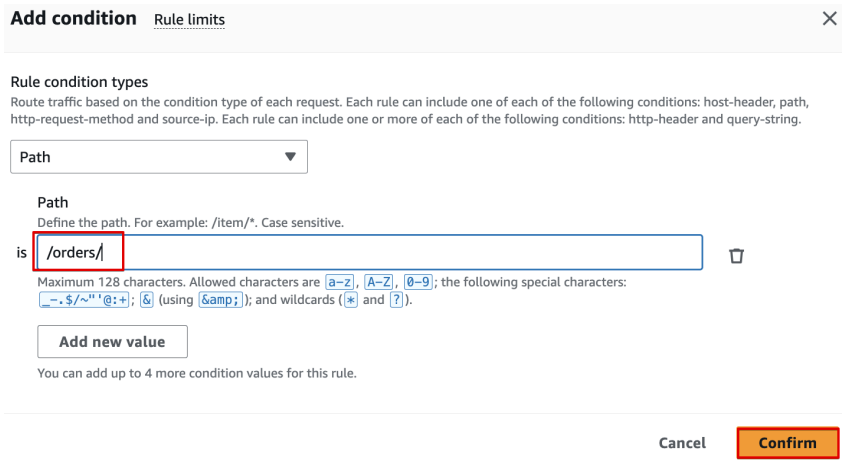
Update the listener rules to route traffic based on path:

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

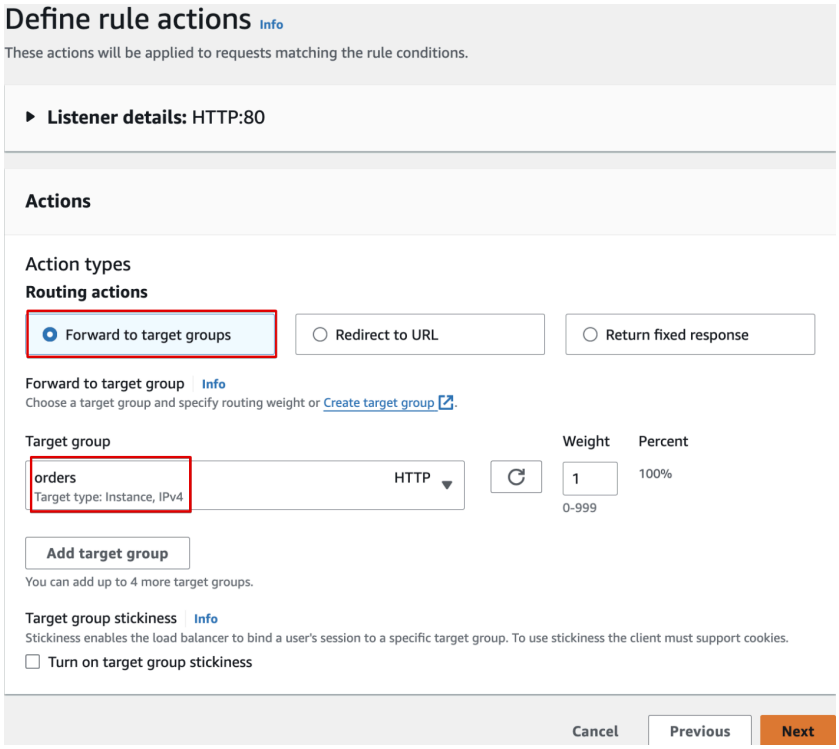
- name the rule, **OrdersRule** (for the routing the the orders backend server) and click on Next
- Define rule conditions: click on Add condition



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



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



- 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

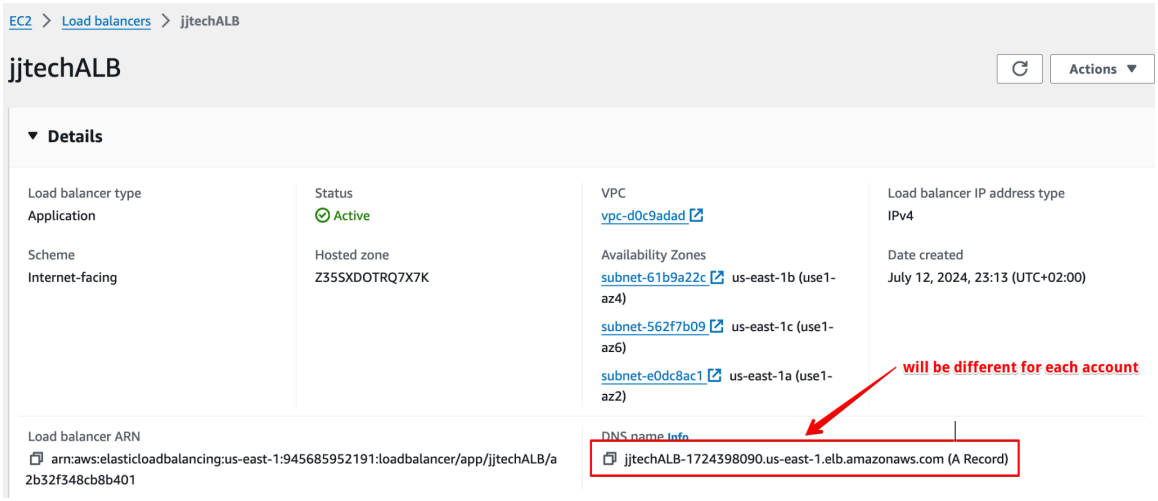
- path: /payments/
- ruel action: forward to target group payments (created above)

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.



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.