Application Load Balancer & Target Groups

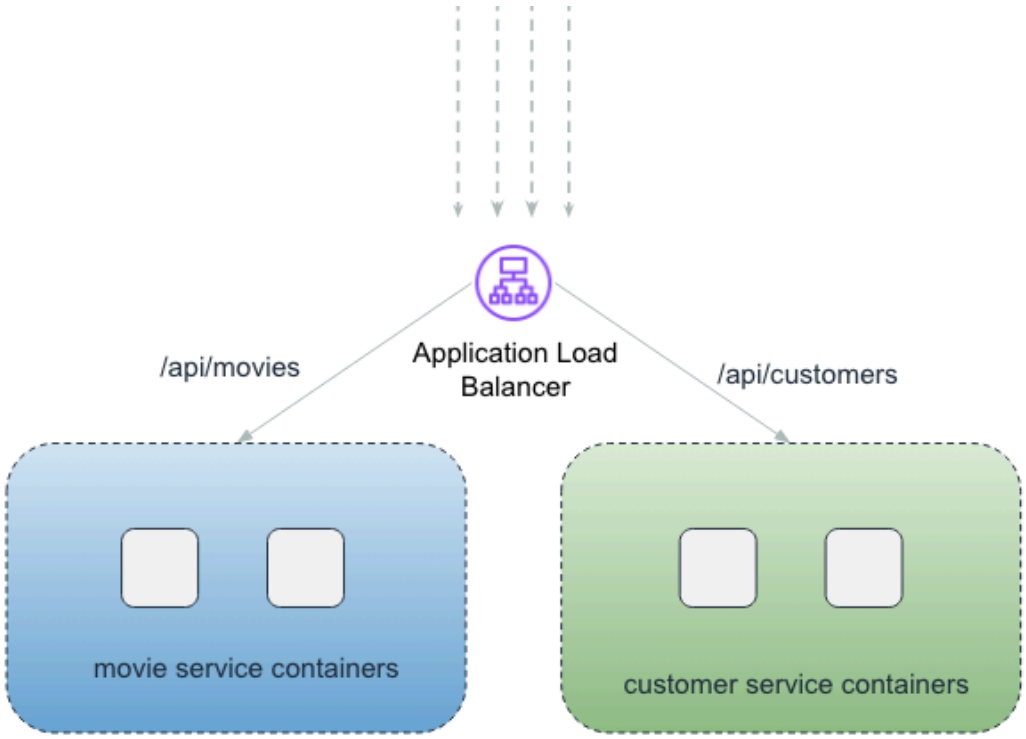
Target Groups **1**
Movie Service Targets / Containers 2
Customer Service Targets / Containers 5
Application Load Balancer **7**

Target Groups

We have 2 backend applications!

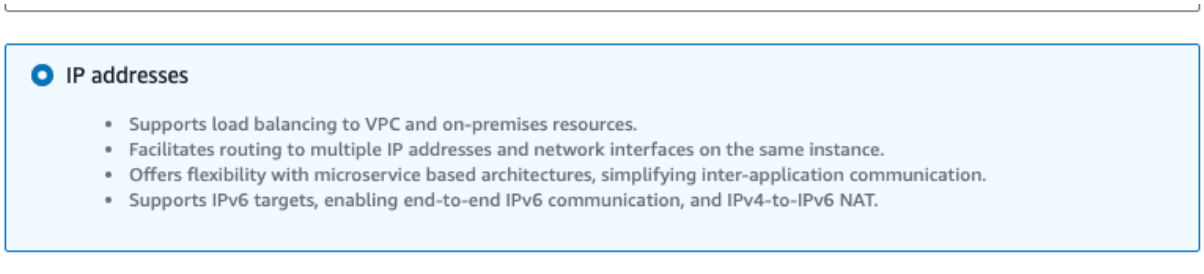
- customer-service
- movie-service

We will receive all the traffic via our Application Load Balancer. Based on the Path, we would route the requests to appropriate applications!



Movie Service Targets / Containers

- Select the target type as **IP addresses**



- Listening port will be 8080

Target group name

movie-service-containers

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 | 8080

1-65535

- Select the VPC and the application protocol

VPC

Select the VPC that hosts the load balancer. Only VPCs that support the IP address type selected above are available in this list. On the **Register targets** page, you can register IP addresses from this VPC, or from private IP addresses located outside of this load balancer's VPC (such as a peered VPC, EC2-Classic, or on-premises targets that are reachable over Direct Connect or VPN).

netflux-vpc
vpc-057e4b12c96c3791e
IPv4 VPC CIDR: 10.0.0.0/16

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.
- ( ) gRPC
  Send requests to targets using gRPC. Supported when the request protocol is gRPC.

- Enter the health check details

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.

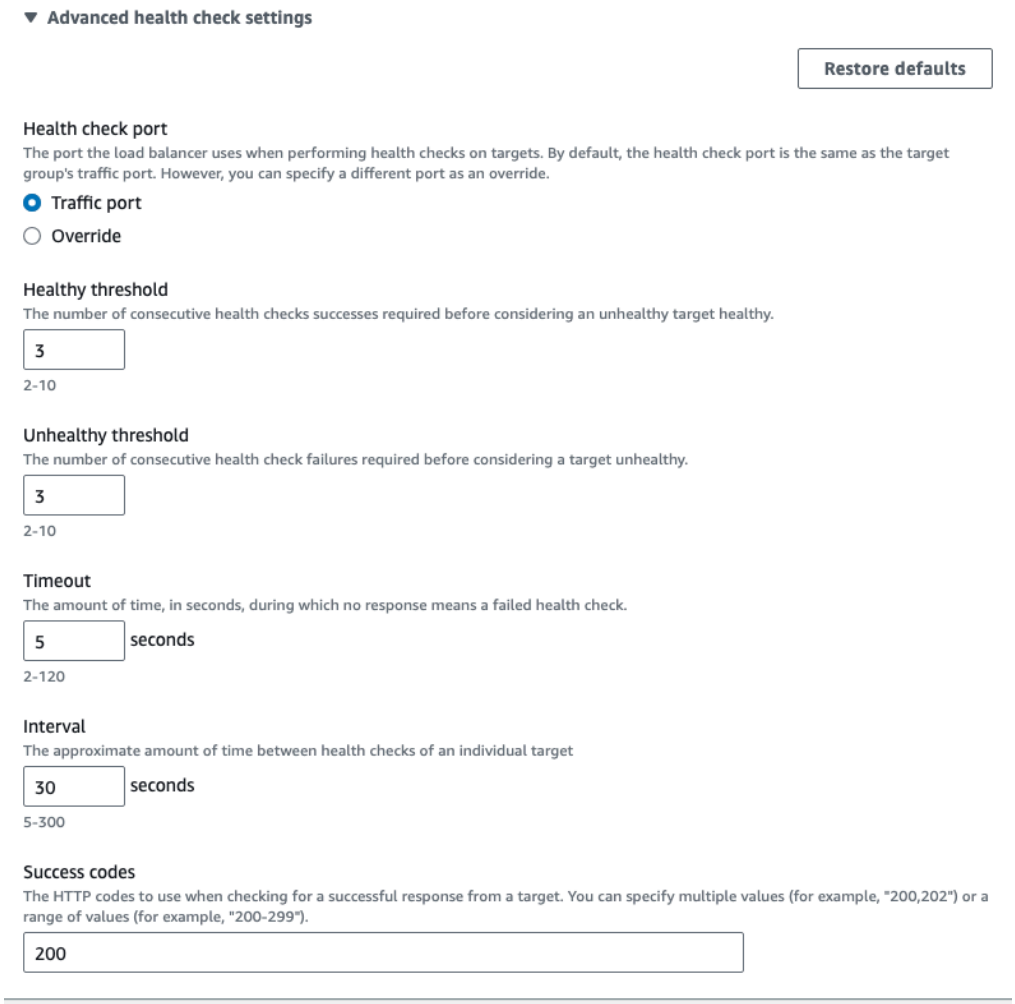
/actuator/health

Up to 1024 characters allowed.

- advanced health check settings

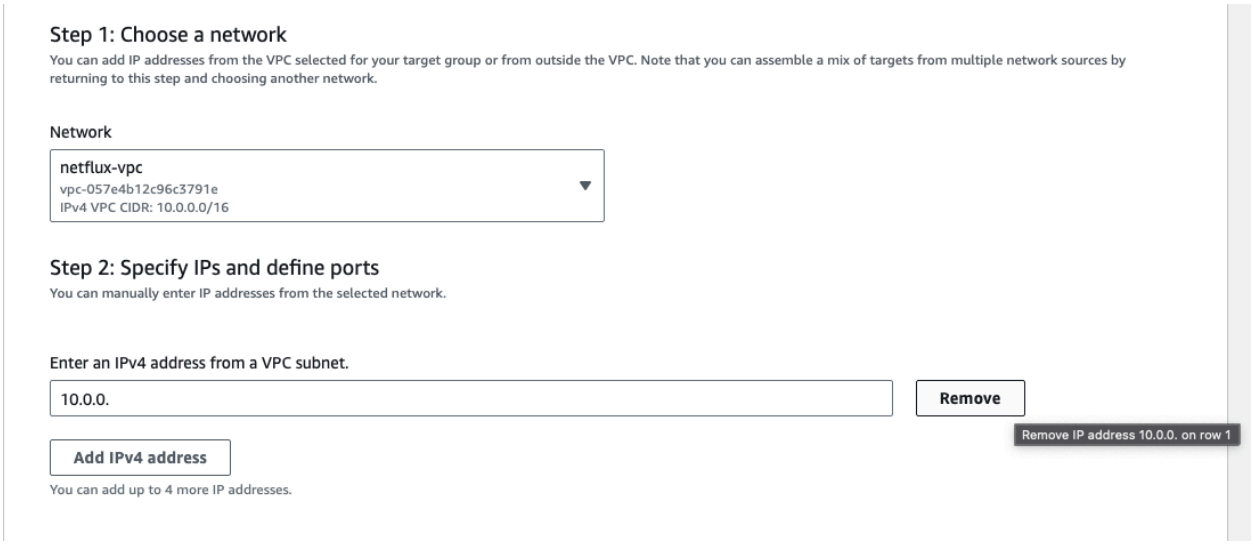
Healthy Threshold	3

Unhealthy Threshold	3
Timeout	5 seconds
Interval	30 seconds
Success codes	200

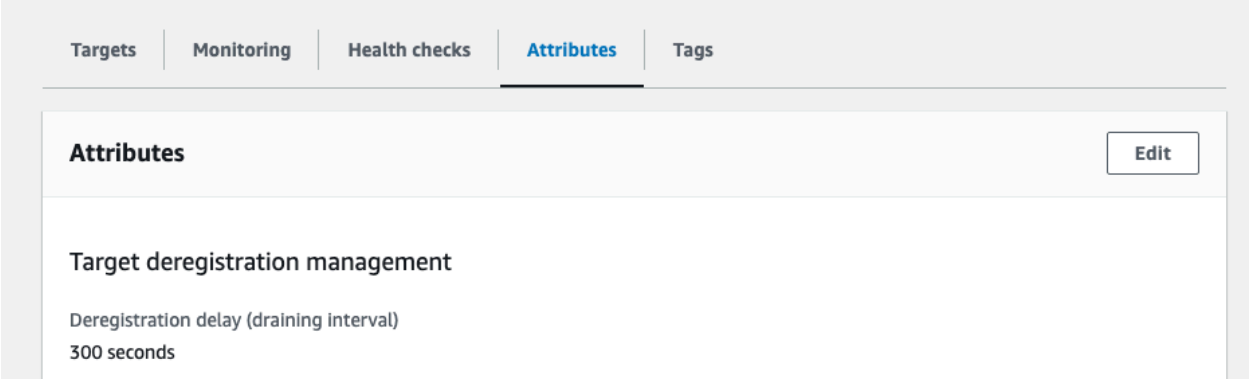


- Click “Next”

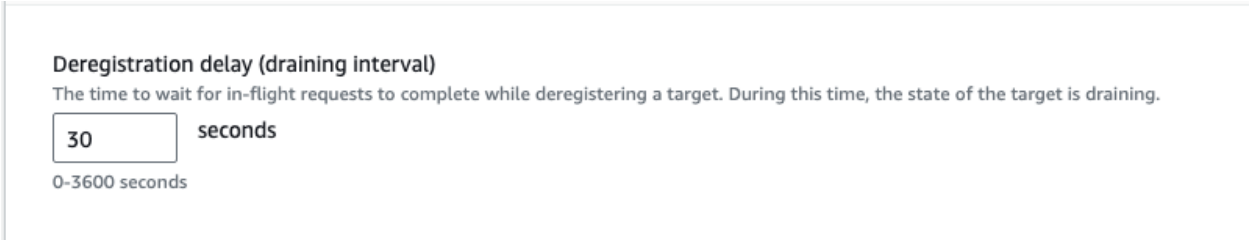
- We can remove the IP addresses as we do not know them. When the app starts, they will register themselves!



- Click on “Create Target Group”.
- Once created, Go to “Attributes” and “Edit”



- **Reduce the time to 30 seconds** as 5 mins would be too much!



Customer Service Targets / Containers

- Repeat the above steps for “customer-service-containers”

Target group name

customer-service-containers

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 | 8080

1-65535

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.

/actuator/health

Up to 1024 characters allowed.

Healthy Threshold	3
Unhealthy Threshold	3
Timeout	5 seconds
Interval	30 seconds
Success codes	200

Targets | Monitoring | Health checks | Attributes | Tags

Attributes

Edit

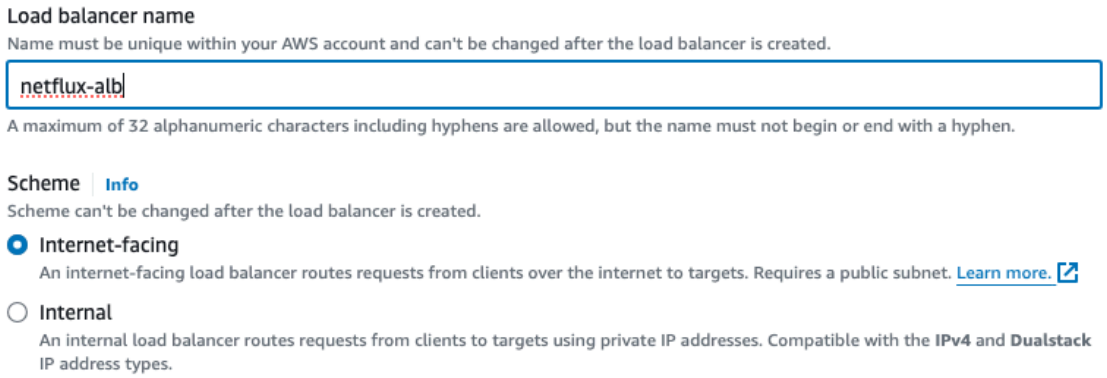
Target deregistration management

Deregistration delay (draining interval)

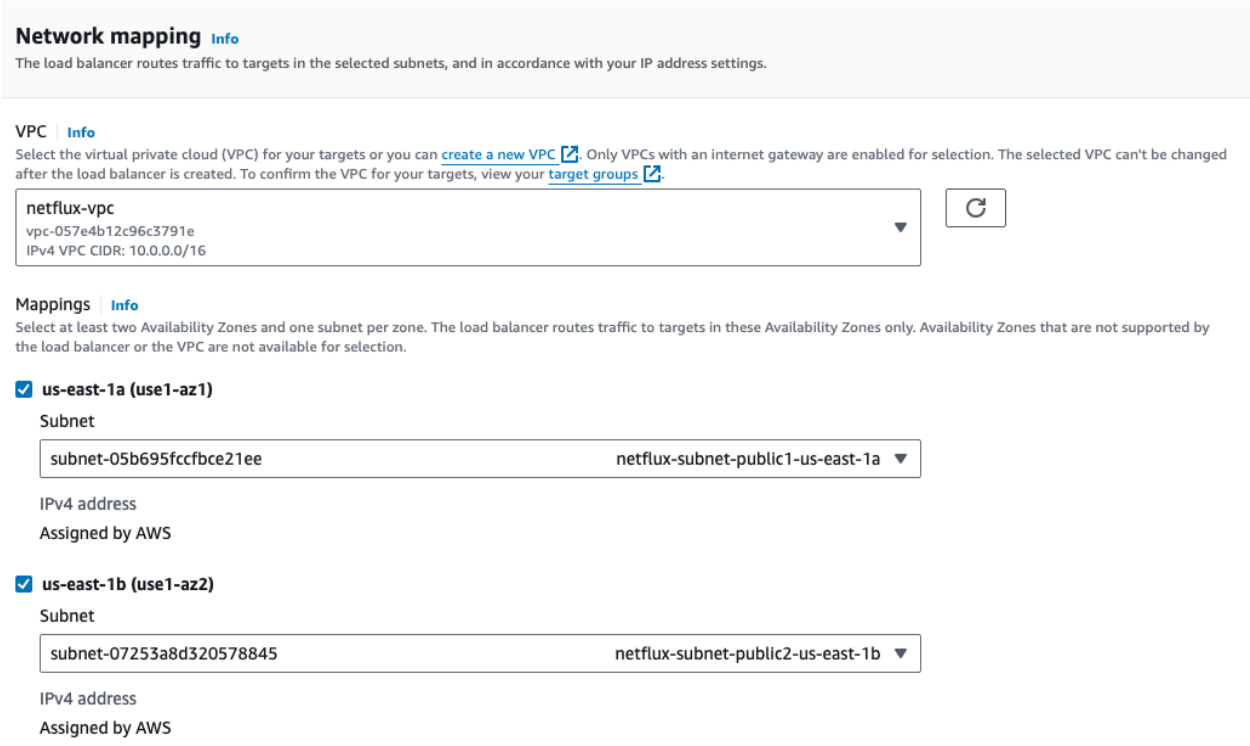
30 seconds

Application Load Balancer

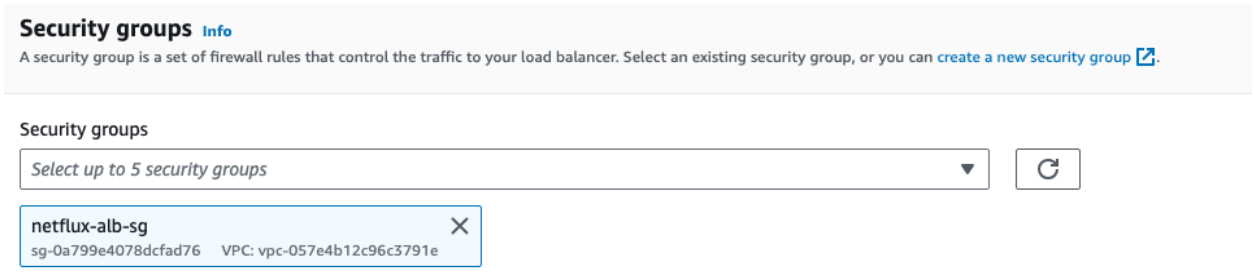
- Let’s create an application load balancer. It is **internet facing**!



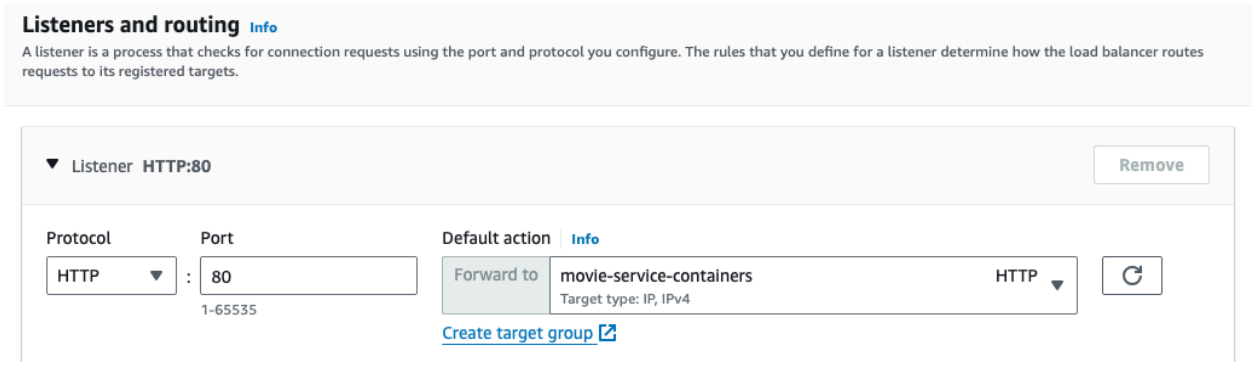
- Select VPC and Subnets. Our ALB will be placed under the public subnets!



- Attach the “**netflux-alb-sg**”



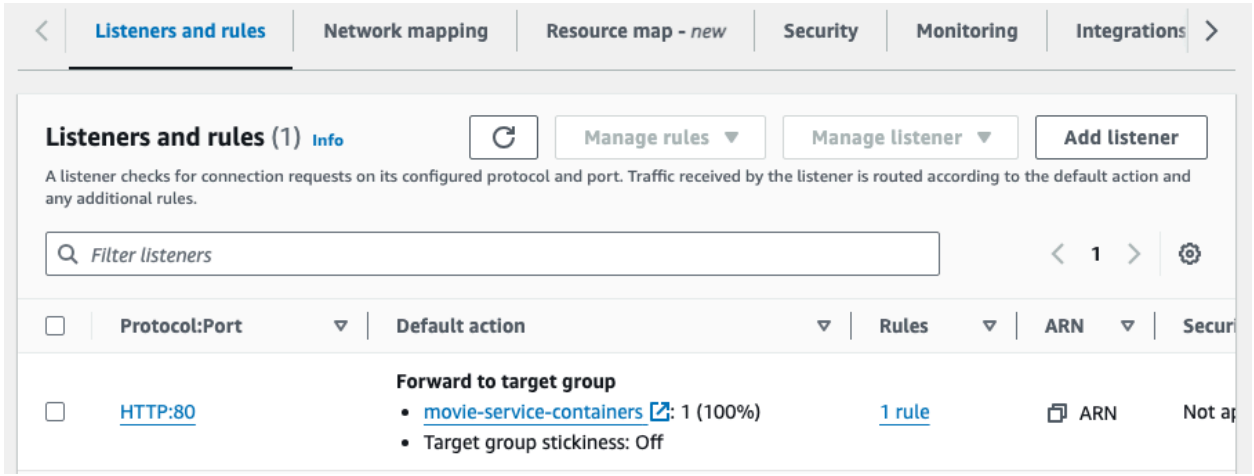
- Our ALB will listen on port 80. We need to provide the default target group. select “movie-service-containers”. We can update the rules later.



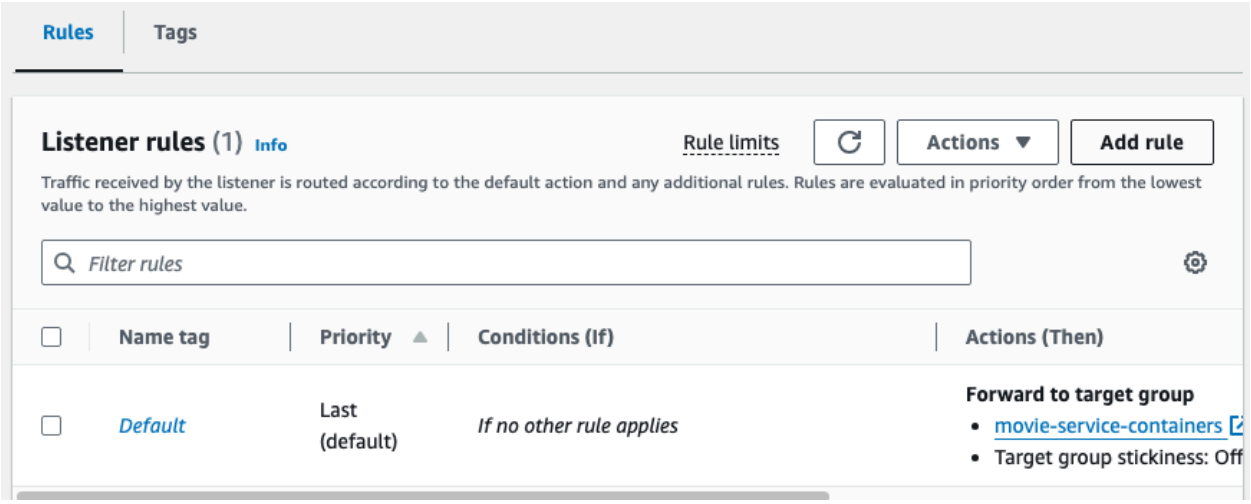
- Click on “Create Load Balancer”

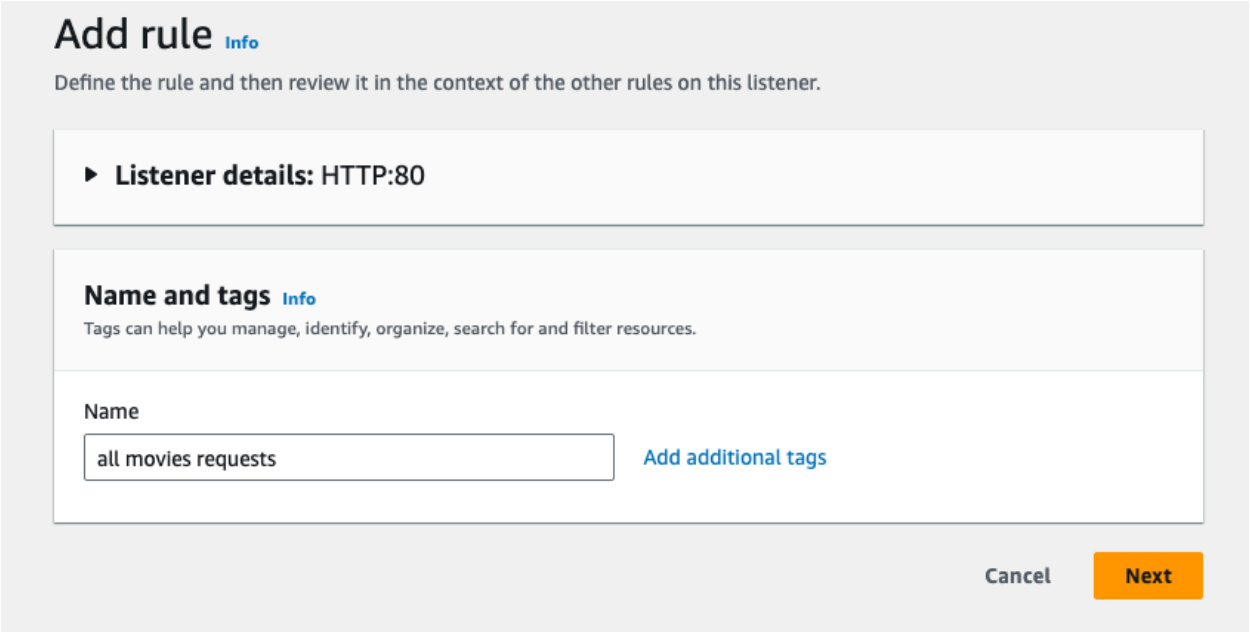
Listener Rules

- Once ALB is created, go to “Listeners and rules”

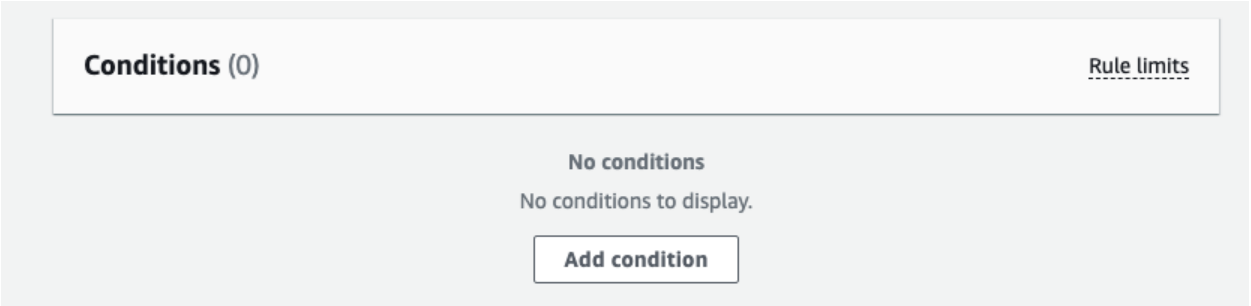


- Click on “**1 rule**” to update the rules for this listener.
- Click on “**Add rule**”. Let’s add 2 rules based on the Path.

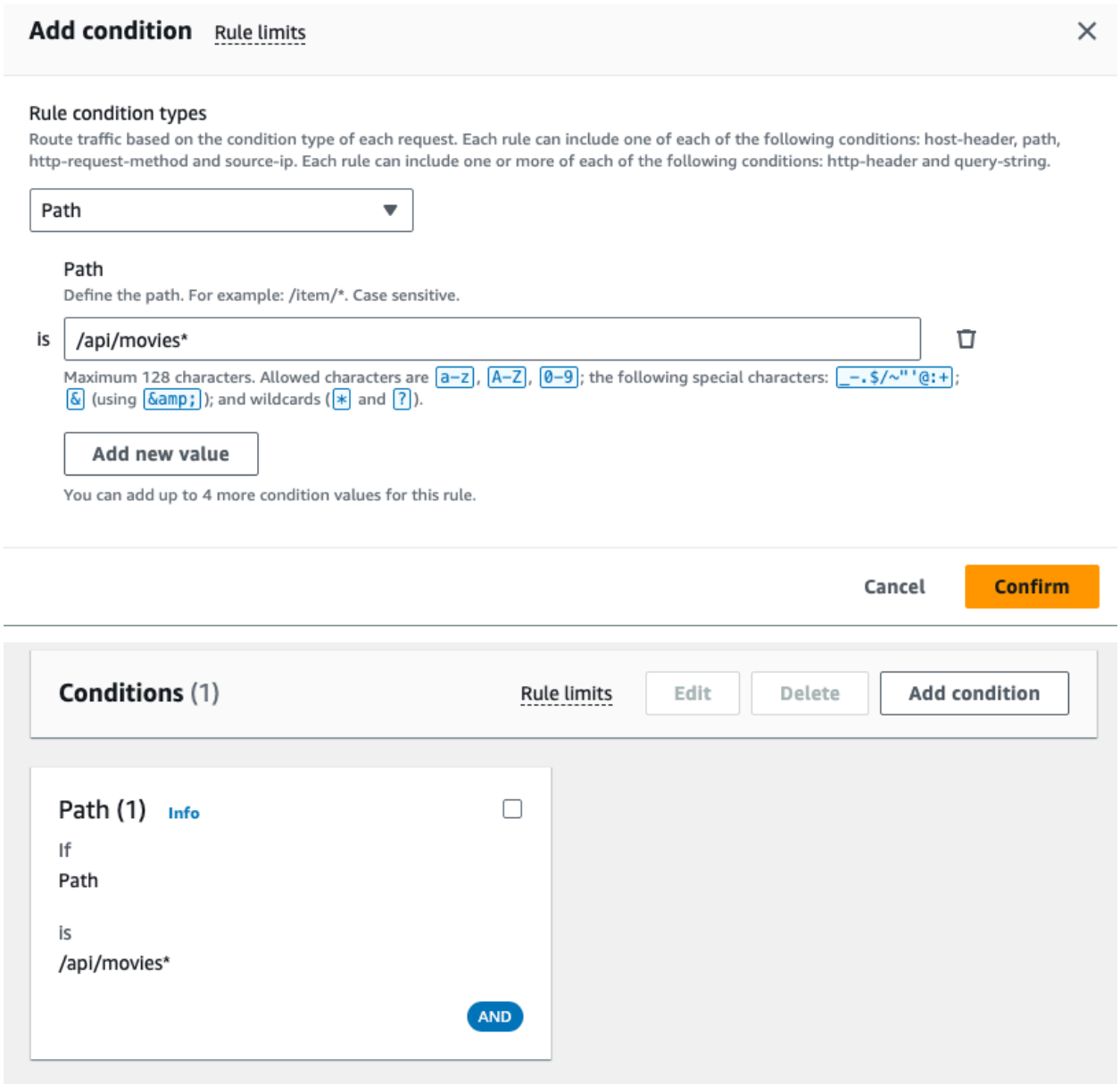




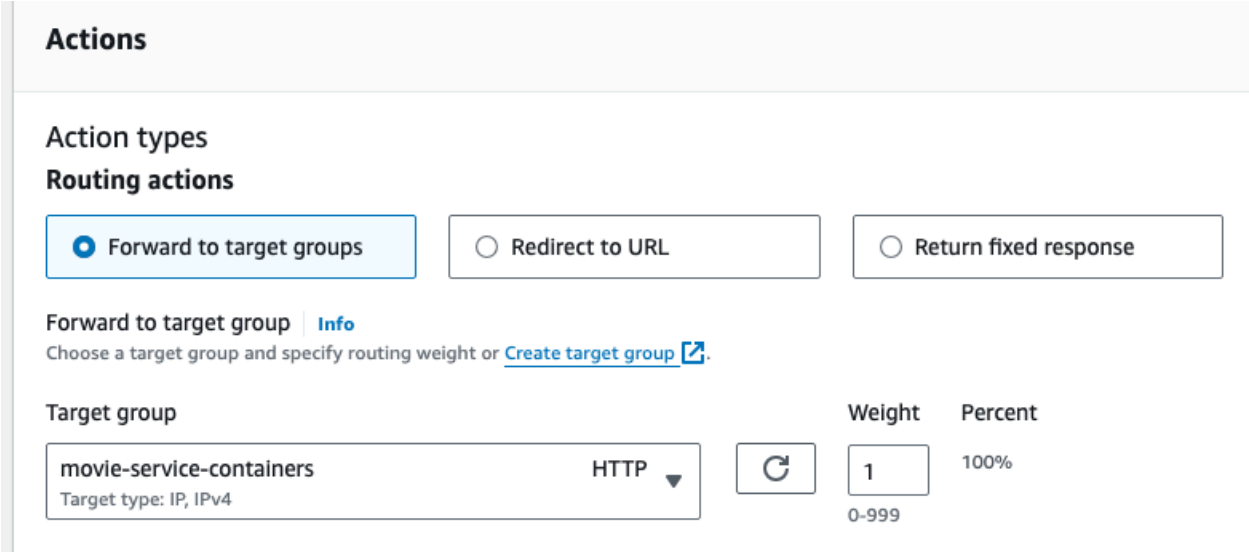
- Add condition



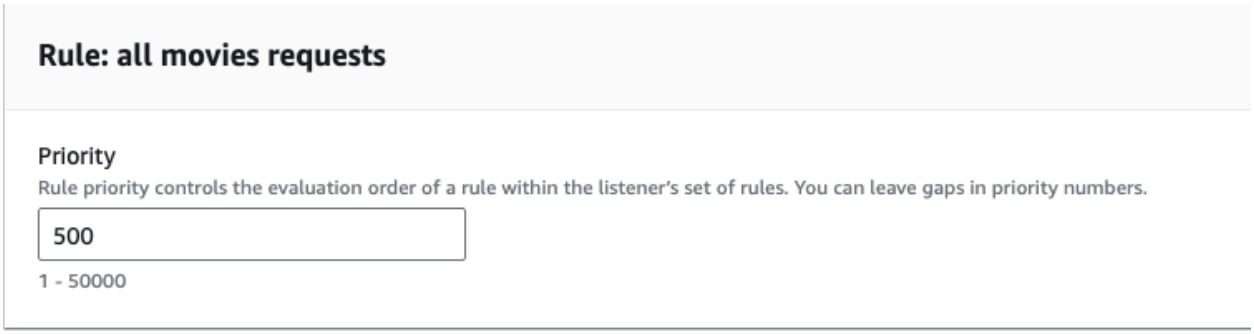
- We need the “Path” based routing.
    - Any “**/api/movies***” should go to **movie-service-containers**



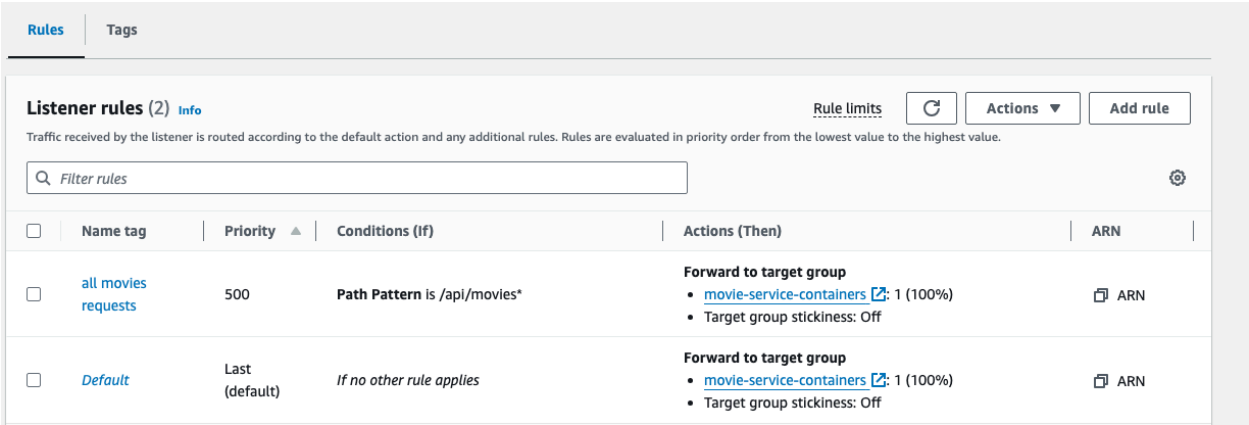
- Click on “Next”
- Select the appropriate target. In this case, It is as shown below.



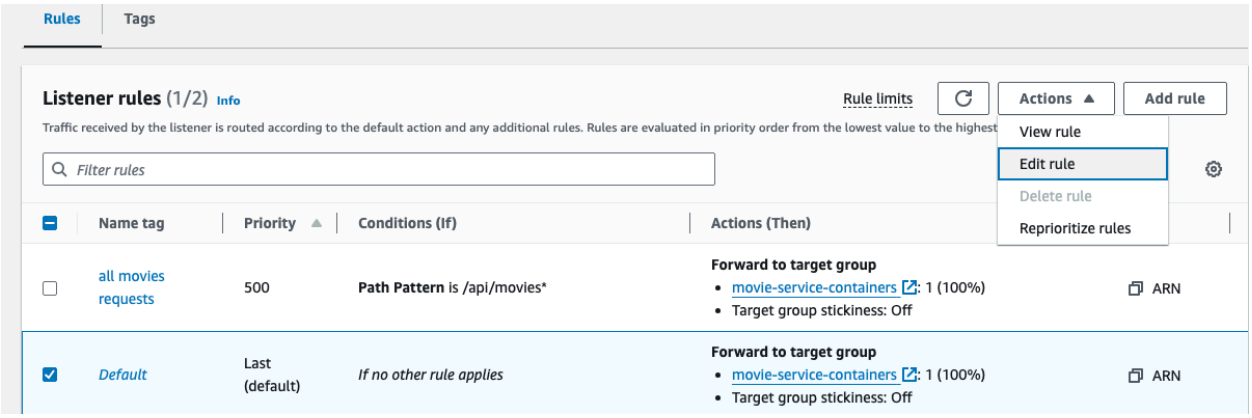
- We have to set the priority. I give ***500***.



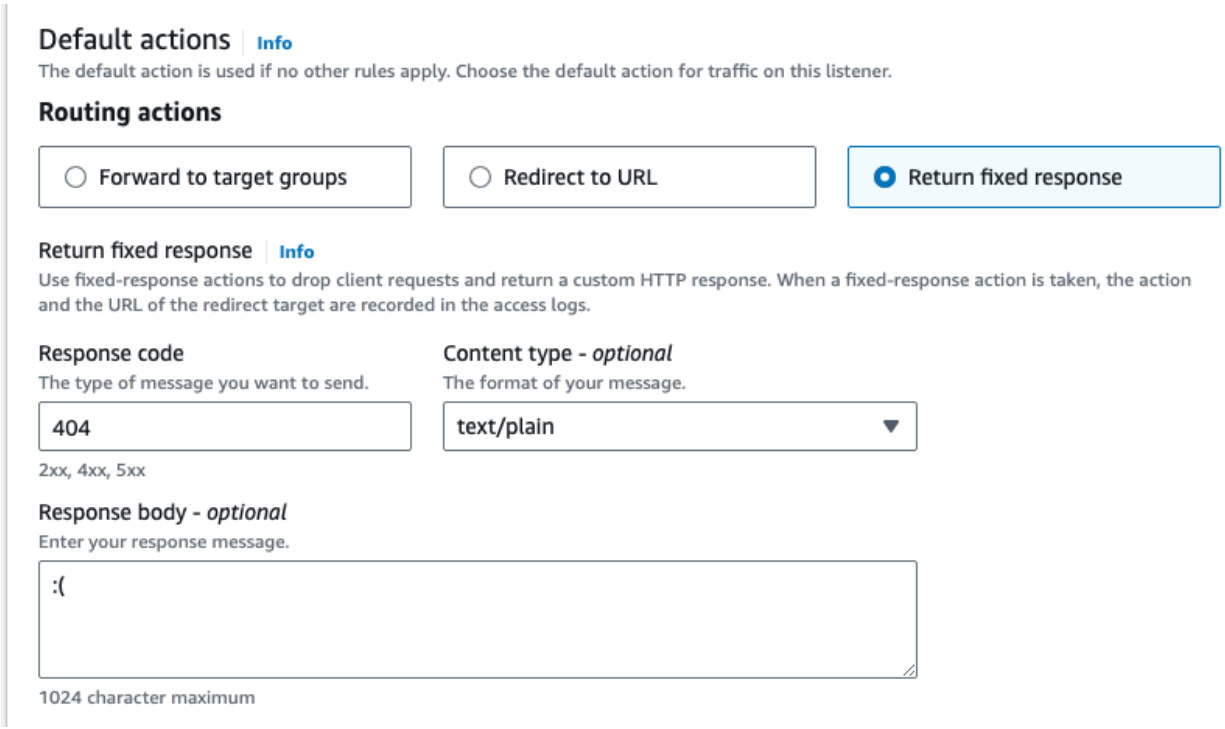
- Create the rule. We should see 2 rules as shown below.



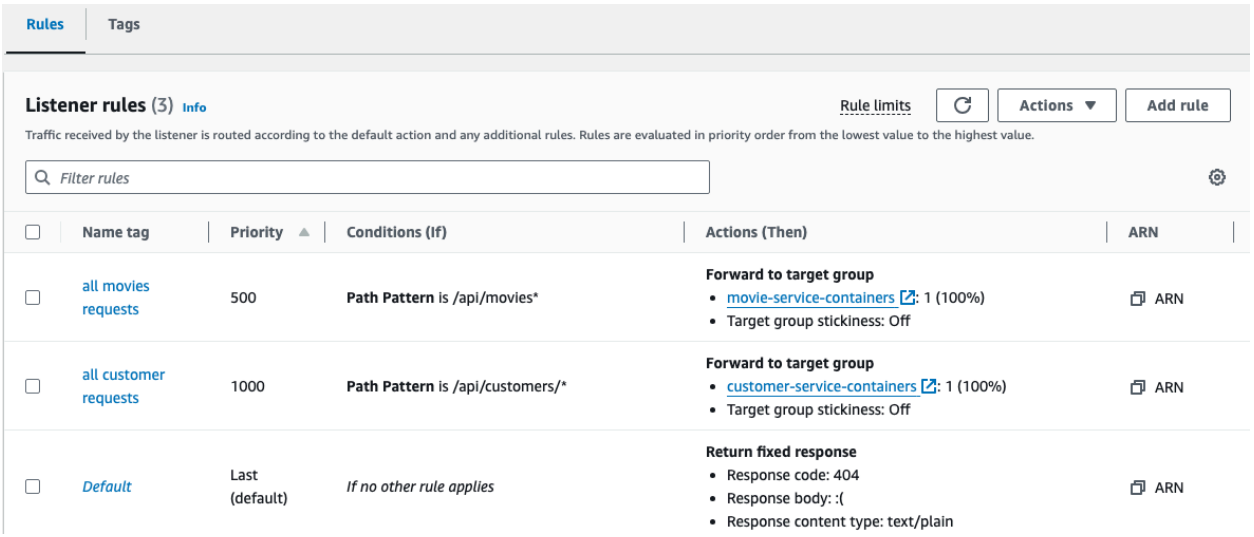
- We can edit the “Default” rule.



- We can change the default response as shown below or anything you prefer!



- Repeat the same for customer-service requests. Finally we should have 3 rules as shown below.


Rules
Tags
Listener rules (3) Info
Rule limits
Actions
Add rule
Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.
Filter rules
Name tag
Priority
Conditions (If)
Actions (Then)
ARN
all movies requests
500
Path Pattern is /api/movies*
Forward to target group
movie-service-containers: 1 (100%)
Target group stickiness: Off
ARN
all customer requests
1000
Path Pattern is /api/customers/*
Forward to target group
customer-service-containers: 1 (100%)
Target group stickiness: Off
ARN
Default
Last (default)
If no other rule applies
Return fixed response
Response code: 404
Response body: :(
Response content type: text/plain
ARN