

## ▪ Protocol: Choose HTTP.

The screenshot shows the AWS Lambda function configuration page. The 'Target group name' field is set to 'customer-tg-one'. Under 'Protocol : Port', 'HTTP' is selected as the protocol and port 80 is specified. The 'IP address type' is set to 'IPv4'. A note at the bottom indicates that VPCs supporting the selected IP address type must be chosen. The page includes standard AWS navigation and footer links.

## ▪ Port: Enter 8080

The screenshot shows the AWS Lambda function configuration page. The 'Target group name' field is set to 'customer-tg-one'. Under 'Protocol : Port', 'HTTP' is selected as the protocol and port 8080 is specified. The 'IP address type' is set to 'IPv4'. A note at the bottom indicates that VPCs supporting the selected IP address type must be chosen. The page includes standard AWS navigation and footer links.

## ▪ VPC: Choose LabVPC.

**customer-tg-one**

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	

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

IPv4  
 IPv6

**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).

LabVPC vpc-043e60ce741fd0a03 IPv4 VPC CIDR: 10.16.0.0/16
--

**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.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## ■ Health check path: Enter /

**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.

/

Up to 1024 characters allowed.

► Advanced health check settings

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► Tags - optional  
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## ○ Choose Next.

**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.

/

Up to 1024 characters allowed.

► Advanced health check settings

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► Tags - optional  
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- On the Register targets page, accept all defaults (don't register any targets), and choose Create target group.

The screenshots show the 'Register targets' step of the 'Create target group' wizard. The first screenshot shows the initial configuration with a single IP address (10.16.0.) added. The second screenshot shows the 'Ports' section where port 8080 is specified. The third screenshot shows the 'Review targets' step, which lists the target (IP 10.16.0.0, Port 8080) and provides a summary message.

**Step 1: Choose a network**  
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

**Network**

LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**Step 2: Specify IPs and define ports**  
You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.  
10.16.0.0 Remove

**Ports**  
Ports for routing to this target.  
8080  
1-65535 (separate multiple ports with commas)  
Include as pending below

**Review targets**

**Step 3: Review IP targets to include in your group**  
Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

**Targets (0)**  
Q Filter targets Show only pending Remove all pending

8080  
1-65535 (separate multiple ports with commas)  
Include as pending below

**Review targets**

**Step 3: Review IP targets to include in your group**  
Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

**Targets (0)**  
Q Filter targets Show only pending Remove all pending

Remove IPv4 address Health status IP address Port Zone

No IP addresses included yet  
Specify IP addresses above and add to list.

0 pending Cancel Previous Create target group

The screenshot shows the AWS EC2 Target Groups page. On the left, a sidebar lists various EC2 services like Instances, Images, and Elastic Block Store. The main content area shows a success message: "Successfully created the target group: customer-tg-one. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." Below this, the "customer-tg-one" target group is displayed with its details: Target type (IP), Protocol (HTTP: 8080), IP address type (IPv4), Load balancer (None associated), and VPC (vpc-045e6f0ce741fda03). A summary table shows 0 total targets, 0 healthy, 0 unhealthy, 0 unused, 0 initial, and 0 draining.

The screenshot shows the AWS EC2 Target Groups page. The sidebar is identical to the previous screenshot. The main content area shows a table titled "Target groups (1) info" with one entry: "customer-tg-one". The table columns include Name, ARN, Port, Protocol, Target type, and Load balancer. The "customer-tg-one" row has the ARN arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-one/0b608841b3b9d944, Port 8080, Protocol HTTP, Target type IP, and Load balancer None associated. Below the table, a message says "0 target groups selected" and "Select a target group above."

## 2. Create a second target group for the *customer* microservice. Use the same settings as the first target group except use *customer-tg-two* as the target group name.

The screenshot shows the "Create target group" wizard, Step 1: Specify group details. The sidebar shows "Step 1: Specify group details" and "Step 2: Register targets". The main content area is titled "Specify group details" and contains a "Basic configuration" section with the note "Settings in this section can't be changed after the target group is created." It includes a "Choose a target type" section with three options: "Instances" (radio button not selected), "IP addresses" (radio button selected, highlighted in blue), and "Lambda function" (radio button not selected). The "IP addresses" section lists benefits such as supporting load balancing to VPC and on-premises resources, facilitating routing to multiple IP addresses and network interfaces on the same instance, offering flexibility with microservice-based architectures, and supporting IPv6 targets.

**[Alt+S]**

N. Virginia v  
vocabs/user\$223197=Gupta\_Nancy @ 0824-5190-8674 ▾

**Lambda function**

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

**Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

**Target group name**

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  1-65535

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

IPv4

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**[Alt+S]**

N. Virginia v  
vocabs/user\$223197=Gupta\_Nancy @ 0824-5190-8674 ▾

**Lambda function**

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

**Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

**Target group name**

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  1-65535

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

IPv4

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**[Alt+S]**

N. Virginia v  
vocabs/user\$223197=Gupta\_Nancy @ 0824-5190-8674 ▾

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  1-65535

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

IPv4

IPv6

**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).

LabVPC  
vpc-0436ef0ce741fda05  
IPv4 VPC CIDR: 10.16.0.0/16

**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.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Health check protocol: HTTP

Health check path: /

[Advanced health check settings](#)

### Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

### Tags - optional

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

[Cancel](#)

[Next](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2 > Target groups > Create target group

Step 1: Specify group details

Step 2: Register targets

**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.

**IP addresses**

**Step 1: Choose a network**  
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

Network: LabVPC  
vpc-045e6f0ce741fda05  
IPv4 VPC CIDR: 10.16.0.0/16

**Step 2: Specify IPs and define ports**  
You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet:  
10.16.0. [Remove](#)

[Include as pending below](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

8080  
1-65535 (separate multiple ports with commas)

[Include as pending below](#)

**Review targets**

**Step 3: Review IP targets to include in your group**  
Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

Targets (0)				
<a href="#">Filter targets</a> <input type="checkbox"/> Show only pending				
Remove IPv4 address	Health status	IP address	Port	Zone
No IP addresses included yet Specify IP addresses above and add to list.				

0 pending [Cancel](#) [Previous](#) [Create target group](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully created the target group: customer-tg-two. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.

**customer-tg-two**

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
0	0	0	0	0	0

**Target groups (2)**

Name	ARN	Port	Protocol	Target type	Load balancer
customer-tg-two	arn:aws:elasticloadbalancing...	8080	HTTP	IP	None associated
customer-tg-one	arn:aws:elasticloadbalancing...	8080	HTTP	IP	None associated

**3. Create a target group for the *employee* microservice. Use the same settings as the other target groups with the following exceptions:**

- **Target group name: Enter employee-tg-one**
- **Health check path: Enter /admin/suppliers**

aws Services Search [Alt+S] N. Virginia vocabs/user5223197=Gupta\_Nancy @ 0824-5190-8674

EC2 > Target groups > Create target group

Step 1 Specify group details

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

Step 2 Register 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 communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vocabs/user5223197=Gupta\_Nancy @ 0824-5190-8674

EC2 > Target groups > Create target group

Target group name

employee-tg-one

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

IPv6

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).

vpc-000e564836085ddaa

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vocabs/user5223197=Gupta\_Nancy @ 0824-5190-8674

EC2 > Target groups > Create target group

Target group name

employee-tg-one

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

IP address type

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

IPv4

IPv6

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).

vpc-000e564836085ddaa

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**IP address type**

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

IPv4

IPv6

**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).

LabVPC  
vpc-043e6f0ce7411da03  
IPv4 VPC CIDR: 10.16.0.0/16

**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.

**Health checks**

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

**Health check protocol**

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

**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.  
/admin/suppliers

Up to 1024 characters allowed.

► Advanced health check settings

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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.  
/admin/suppliers

Up to 1024 characters allowed.

► Advanced health check settings

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► Tags - optional

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

Cancel Next

Screenshot of the AWS EC2 Target Groups creation process, Step 1: Specify group details.

**Step 1: Specify group details**

**Step 2: Register targets**

**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.

**IP addresses**

**Step 1: Choose a network**  
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

**Network**  
LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**Step 2: Specify IPs and define ports**  
You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.  
10.16.0.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Target Groups creation process, Step 2: Register targets.

**Review targets**

**Step 3: Review IP targets to include in your group**  
Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

**Targets (0)**  
 1-65535 (separate multiple ports with commas)

Remove IPv4 address	Health status	IP address	Port	Zone
No IP addresses included yet Specify IP addresses above and add to list.				

**0 pending**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Target Groups creation process, Step 3: Create target group.

**employee-tg-one**

**Details**

arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-one/b526f5dd78815302

Target type	Protocol : Port	Protocol version	VPC
IP	HTTP: 8080	HTTP1	vpc-043e6f0ce741fda03
IP address type	Load balancer	None associated	
IPv4			

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

**Targets**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS EC2 Target Groups page. On the left, there's a navigation sidebar with links like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area is titled "Target groups (3) Info". It has a table with columns: Name, ARN, Port, Protocol, Target type, and Load balancer. The table contains three rows:

Name	ARN	Port	Protocol	Target type	Load balancer
employee-tg-one	arn:aws:elasticloadbalanci...	8080	HTTP	IP	None associated
customer-tg-two	arn:aws:elasticloadbalanci...	8080	HTTP	IP	None associated
customer-tg-one	arn:aws:elasticloadbalanci...	8080	HTTP	IP	None associated

Below the table, it says "0 target groups selected" and "Select a target group above."

**4. Create a second target group for the *employee* microservice. Use the same settings as the other target groups with the following exceptions:**

- **Target group name:** Enter **employee-tg-two**
- **Health check path:** Enter **/admin/suppliers**

This screenshot is identical to the one above, showing the AWS EC2 Target Groups page with three target groups: employee-tg-one, customer-tg-two, and customer-tg-one. The addition of the new target group 'employee-tg-two' is not visually apparent in this specific screenshot, but it is mentioned in the accompanying text.

aws Services Search [Alt+S] N. Virginia vocabs/user\$223197=Gupta,\_Nancy @ 0824-5190-8674

EC2 > Target groups > Create target group

Step 1 Specify group details

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

Step 2 Register 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 communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vocabs/user\$223197=Gupta,\_Nancy @ 0824-5190-8674

EC2 > Target groups > Create target group

Instances

- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

**Target group name**

employee-tg-two

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

IPv6

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vocabs/user\$223197=Gupta,\_Nancy @ 0824-5190-8674

EC2 > Target groups > Create target group

Instances

Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

**Target group name**

employee-tg-two

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

**IP address type**

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

IPv4

IPv6

**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).

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Step 1: Target group details**

HTTP port: 8080  
HTTP port range: 1-65535

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

IPv4  
 IPv6

**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).

LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**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.

**Health checks**

CloudShell Feedback

Health check protocol: HTTP

Health check path: /admin/suppliers  
Up to 1024 characters allowed.

Advanced health check settings

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

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

Cancel Next

**Step 2: Register targets**

EC2 > Target groups > Create target group

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.

**IP addresses**

**Step 1: Choose a network**  
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

Network: LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**Step 2: Specify IPs and define ports**  
You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.  
10.16.0. Remove

CloudShell Feedback

**Review targets**

**Step 3: Review IP targets to include in your group**

Confirm the IP targets to include in your target group. Add more IP targets by repeating steps 1 and 2 on this page. You can also register additional targets after your target group is created.

Targets (0)				
Remove IPv4 address	Health status	IP address	Port	Zone
No IP addresses included yet Specify IP addresses above and add to list.				

0 pending

Cancel Previous Create target group

Successfully created the target group: employee-tg-two. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.

**employee-tg-two**

**Details**

arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-two/963c5e2cc0a65bb2

Target type	Protocol : Port	Protocol version	VPC
IP	HTTP: 8080	HTTP1	vpc-043e6f0ce741fda03
IP address type	Load balancer		
IPv4	None associated		

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

Targets Monitoring Health checks Attributes Tags

**Important: Carefully confirm the name and port number of each target group. The following image provides an example:**

**Target groups (4) [Info](#)**

Name	ARN	Port	Protocol	Target type	Load balancer
employee-tg-two	arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-two/963c5e2cc0a65bb2	8080	HTTP	IP	None associated
employee-tg-one	arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-one/963c5e2cc0a65bb2	8080	HTTP	IP	None associated
customer-tg-two	arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-two/963c5e2cc0a65bb2	8080	HTTP	IP	None associated
customer-tg-one	arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-one/963c5e2cc0a65bb2	8080	HTTP	IP	None associated

**0 target groups selected**

Select a target group above.

## Task 6.2: Create a security group and an Application Load Balancer, and configure rules to route traffic

In this task, you will create an Application Load Balancer. You will also define two listeners for the load balancer: one on port 80 and another on port 8080. For each listener, you will then define path-based routing rules so that traffic is routed to the correct target group depending on the URL that a user attempts to load.

The screenshot shows the AWS CloudSearch interface. The search bar at the top contains the query 'EC@'. The left sidebar has sections for 'AWS Cloud9', 'Services (111)', 'Features (291)', 'Resources (New)', 'Documentation (623,873)', 'Knowledge Articles (1,917)', 'Blogs (22,232)', 'Events (631)', and 'Tutorials (100)'. The main search results are displayed under 'Services' and 'Features'. Under 'Services', the 'EC2' card is selected, showing it's a 'Virtual Servers in the Cloud'. Under 'Features', cards for 'Security Hub', 'Security Lake', and 'Direct Connect' are shown. A right-hand panel titled 'Open in Cloud9' contains a code editor with AWS Lambda code and a 'Create environment' button. The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, and Copyright information.

1. Create a new EC2 security group named **microservices-sg** to use in *LabVPC*. Add inbound rules that allow TCP traffic from any IPv4 address on ports 80 and 8080.

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. The left sidebar includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Console-to-Code Preview', 'Instances' (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations New), 'Images' (AMIs, AMI Catalog), and 'Elastic Block Store' (Volumes). The main area displays a table for 'Security Groups (5)'. The table columns are Name, Security group ID, Security group name, VPC ID, and Description. One row is selected, showing 'sg-0d3dc7a32965ee9fa' as the Security group ID, 'c110323a2605598l6548437tw0824...' as the Security group name, 'vpc-043e6f0fce741fda05' as the VPC ID, and 'Enable inbound' as the Description. A 'Create security group' button is located at the top right of the table area. The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, and Copyright information.

AWS Services Search [Alt+S] N. Virginia v vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674 ⓘ

EC2 > Security Groups > Create security group

### 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  
microservices-sg  
Name cannot be edited after creation.

Description Info  
Allows SSH access to developers

VPC Info  
vpc-043e6f0ce741fda03 (LabVPC)

**Inbound rules Info**

This security group has no inbound rules.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] N. Virginia v vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674 ⓘ

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  
microservices-sg  
Name cannot be edited after creation.

Description Info  
Security group for microservices application

VPC Info  
vpc-043e6f0ce741fda03 (LabVPC)

**Inbound rules Info**

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
Custom TCP	TCP	8080	Any... ▾	<input type="text"/> 0.0.0.0/0 X

Add rule

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] N. Virginia v vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674 ⓘ

Name cannot be edited after creation.

Description Info  
Allows SSH access to developers

VPC Info  
vpc-043e6f0ce741fda03 (LabVPC)

**Inbound rules Info**

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
HTTP	TCP	80	Any... ▾	<input type="text"/> 0.0.0.0/0 X

Add rule

⚠ 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. X

**Outbound rules Info**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] N. Virginia voclabs/user\$223197=Gupta\_Nancy @ 0824-5190-8674

microservices-sg  
Name cannot be edited after creation.

Description Info  
Allows SSH access to developers

VPC Info  
vpc-043e6f0ce741fda03 (LabVPC)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anyw... <input type="text" value="0.0.0.0/0"/> X	
Custom TCP	TCP	8080	Anyw... <input type="text" value="0.0.0.0/0"/> X	

Add rule

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] N. Virginia voclabs/user\$223197=Gupta\_Nancy @ 0824-5190-8674

Outbound rules Info

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom <input type="text" value="0.0.0.0/0"/> X	

Add rule

⚠ 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. X

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 tags

Create security group

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] N. Virginia voclabs/user\$223197=Gupta\_Nancy @ 0824-5190-8674

EC2 Dashboard EC2 Global View Events Console-to-Code Preview Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations New

Security group (sg-091ac242028763d75 | microservices-sg) was created successfully

Details

EC2 > Security Groups > sg-091ac242028763d75 - microservices-sg

sg-091ac242028763d75 - microservices-sg Actions ▾

Details

Security group name <input type="text" value="microservices-sg"/>	Security group ID <input type="text" value="sg-091ac242028763d75"/>	Description <input type="text" value="Security group for microservices application"/>	VPC ID <input type="text" value="vpc-043e6f0ce741fda03"/>
Owner <input type="text" value="082451908674"/>	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Inbound rules (2)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**EC2 Dashboard**

**Details**

Security group name sg-091ac242028763d75	Security group ID sg-091ac242028763d75	Description Security group for microservices application	VPC ID vpc-043e6f0ce741fda03
Owner 082451908674	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

**Inbound rules (2)**

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0c1f5115578628941	IPv4	Custom TCP	TCP	8080
-	sgr-015a6a09830eac0fb	IPv4	HTTP	TCP	80

**EC2 Dashboard**

**Security Groups (1/6)**

Name	Security group ID	Security group name	VPC ID	Description
sg-0d3dc7a32965ee9fa	c110323a2605598165484371w0824...	vpc-043e6f0ce741fda03	Enable inb...	
DBSecurityGroup	sg-012f4abba21b3b5d8	DBSecurityGroup	vpc-043e6f0ce741fda03	Enable acc...
<b>sg-091ac242028763d75</b>	<b>microservices-sg</b>	<b>vpc-043e6f0ce741fda03</b>	<b>Security gr...</b>	

**sg-091ac242028763d75 - microservices-sg**

**Inbound rules (2)**

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0c1f5115578628941	IPv4	Custom TCP	TCP	8080
-	sgr-015a6a09830eac0fb	IPv4	HTTP	TCP	80

## 2. In the Amazon EC2 console, create an Application Load Balancer named **microservicesLB**.

**EC2 Dashboard**

**Resources**

Instances (running)	2	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	2	Key pairs	1
Load balancers	0	Placement groups	0	Security groups	5
Snapshots	0	Volumes	2		

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Service health**

AWS Health Dashboard

**Explore AWS**

Get Up to 40% Better Price Performance

T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

Amazon GuardDuty Malware Protection

GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. Learn more

**Introducing resource map for Network Load Balancers**

Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancer. Resource map can be viewed on the load balancers detail page. Share feedback to help us improve your experience.

**Load balancers**

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

**0 load balancers selected**

Select a load balancer above.

**Create load balancer**

CloudShell Feedback

**Compare and select load balancer type**

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

Load balancer types		
<b>Application Load Balancer Info</b>	<b>Network Load Balancer Info</b>	<b>Gateway Load Balancer Info</b>
Choose an Application Load	Choose a Network Load Balancer	Choose a Gateway Load Balancer

CloudShell Feedback

<b>Application Load Balancer Info</b>	<b>Network Load Balancer Info</b>	<b>Gateway Load Balancer Info</b>
<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.</p> <p><a href="#">Create</a></p>	<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.</p>	<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.</p> <p><a href="#">Create</a></p>

CloudShell Feedback

The screenshot shows the 'Create Application Load Balancer' wizard on the 'Basic configuration' step. The 'Load balancer name' field is set to 'microservicesLB'. The 'Scheme' section is set to 'Internet-facing', which is described as routing requests from clients over the internet to targets. The 'IP address type' is set to 'IPv4'.

- **Make it internet facing for IPv4 addresses.**

The screenshot shows the 'Create Application Load Balancer' wizard on the 'Basic configuration' step. The 'Load balancer name' field is set to 'microservicesLB'. The 'Scheme' section is set to 'Internet-facing', which is described as routing requests from clients over the internet to targets. The 'IP address type' is set to 'IPv4'.

- **Use *LabVPC*, *Public Subnet1*, *Public Subnet2*, and the *microservices-sg* security group.**

Screenshot of the AWS CloudFront configuration page for creating a new distribution.

**IP Address Type**

- IPv4
  - Includes only IPv4 addresses.
- Dualstack
  - Includes IPv4 and IPv6 addresses.

**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](#).

LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**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-az4)  
 us-east-1b (use1-az6)

**Security groups** Info

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](#).

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**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](#).

LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**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-az4)  
Subnet  
subnet-09458d72d02ae53ea Public Subnet1  
IPv4 address Assigned by AWS

us-east-1b (use1-az6)  
Subnet  
subnet-0e5c0293004d9e1c8 Public Subnet2  
IPv4 address Assigned by AWS

**Security groups** Info

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](#).

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

LabVPC  
vpc-043e6f0ce741fda03  
IPv4 VPC CIDR: 10.16.0.0/16

**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-az4)  
 us-east-1b (use1-az6)

**Security groups** Info

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](#).

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Security groups

Select up to 5 security groups

microservices-sg  
sg-091ac242028763d75 VPC: vpc-043e6f0ce741fda03

**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.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- **Configure two listeners on it. The first should listen on *HTTP:80* and forward traffic to *customer-tg-two* by default. The second should listen on *HTTP:8080* and forward traffic to *customer-tg-one* by default.**

AWS Services Search [Alt+S] N. Virginia vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674

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**

Protocol: HTTP	Port: 80	Default action: Info
Forward to: customer-tg-two Target type: IP, 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.

Add listener tag

You can add up to 50 more tags.

**Listener HTTP:8080**

Protocol: HTTP	Port: 8080	Default action: Info
Forward to: customer-tg-one Target type: IP, IPv4		
Create target group		

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Summary**  
Review and confirm your configurations. Estimate cost

<b>Basic configuration</b> Edit	<b>Security groups</b> Edit	<b>Network mapping</b> Edit	<b>Listeners and routing</b> Edit
microservicesLB <ul style="list-style-type: none"> <li>Internet-facing</li> <li>IPv4</li> </ul>	<ul style="list-style-type: none"> <li>microservices-sg sg-091ac242028763d75</li> </ul>	VPC vpc-043e6f0ce741fda03 LabVPC <ul style="list-style-type: none"> <li>us-east-1a subnet-09458bd72d02ae53ea</li> <li>us-east-1b subnet-0e5c0293004d9e1c8</li> </ul>	<ul style="list-style-type: none"> <li>HTTP:80 defaults to customer-tg-two</li> <li>HTTP:8080 defaults to customer-tg-one</li> </ul>
<b>Service integrations</b> Edit	<b>Tags</b> Edit	None	
AWS WAF: None AWS Global Accelerator: None			
<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>			

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Creation workflow and status**

**Server-side tasks and status**  
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Create load balancer

The screenshot shows the AWS EC2 Dashboard with the 'Load balancers' section selected. A success message at the top states: "Successfully created load balancer: microservicesLB". Below it, the 'microservicesLB' load balancer is listed with its details:

- Load balancer type:** Application
- Status:** Provisioning
- VPC:** vpc-045e6f0ce741fda03
- IP address type:** IPv4
- Scheme:** Internet-facing
- Hosted zone:** Z355XDOTRQ7X7K
- Availability Zones:** subnet-0e5c0295004d9e1c8 (us-east-1b (use1-az2)), subnet-09458d72d02ae53ea (us-east-1a (use1-az4))
- Date created:** April 28, 2024, 13:09 (UTC-04:00)
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:082451908674:loadbalancer/app/microservicesLB/489d84f8743cd112
- DNS name:** microservicesLB-1590830780.us-east-1.elb.amazonaws.com (A Record)

The screenshot shows the 'Listeners and rules' tab for the 'microservicesLB' load balancer. It displays two listeners:

- HTTP:80**: Forward to target group 'customer-tg-two' (1 rule, 100% weight, Group-level stickiness: Off).
- HTTP:8080**: Forward to target group 'customer-tg-one' (1 rule, 100% weight, Group-level stickiness: Off).

### 3. Add a second rule for the **HTTP:80** listener. Define the following logic for this new rule:

- **IF Path is /admin/\***
- **THEN Forward to... the employee-tg-two target group.**

The settings should be the same as shown in the following image:

Name tag	Priority	Conditions (If)	Actions (Then)
-	1	Path Pattern is /admin/*	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>employee-tg-two: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>
<b>Default</b>	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>customer-tg-two: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>

aws Services Search [Alt+S] N. Virginia vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674

EC2 Dashboard EC2 Global View Events Console-to-Code [Preview](#)

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations [New](#)

Images AMIs AMI Catalog

Elastic Block Store Volumes

CloudShell Feedback

**EC2 > Load balancers > microservicesLB > HTTP:80 listener**

### HTTP:80 [Info](#)

**Details**  
A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol:Port HTTP:80	Load balancer <a href="#">microservicesLB</a>	Default actions <b>Forward to target group</b> <ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>
Listener ARN <a href="#">arn:aws:elasticloadbalancing:us-east-1:082451908674:listener/app/microservicesLB/489d84f8743cd112/87c4d1a445adb929</a>		

**Rules** **Tags**

**Listener rules (1) [Info](#)**  
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.

Rule limits				Actions	Add rule										
<input type="text"/> Filter rules				<a href="#">Edit</a>	<a href="#">Actions</a>										
<b>Listener rules (1) <a href="#">Info</a></b> 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.															
<table border="1"><thead><tr><th>Name tag</th><th>Priority</th><th>Conditions (If)</th><th>Actions (Then)</th><th>ARN</th></tr></thead><tbody><tr><td><input type="checkbox"/> Default</td><td>Last (default)</td><td>If no other rule applies</td><td><b>Forward to target group</b><ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul></td><td><a href="#">Edit</a></td></tr></tbody></table>						Name tag	Priority	Conditions (If)	Actions (Then)	ARN	<input type="checkbox"/> Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	<a href="#">Edit</a>
Name tag	Priority	Conditions (If)	Actions (Then)	ARN											
<input type="checkbox"/> Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	<a href="#">Edit</a>											

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674

EC2 Dashboard EC2 Global View Events Console-to-Code [Preview](#)

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations [New](#)

Images AMIs AMI Catalog

Elastic Block Store Volumes

CloudShell Feedback

**EC2 > Load balancers > microservicesLB > HTTP:80 listener**

### HTTP:80 [Info](#)

**Details**  
A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol:Port HTTP:80	Load balancer <a href="#">microservicesLB</a>	Default actions <b>Forward to target group</b> <ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>
Listener ARN <a href="#">arn:aws:elasticloadbalancing:us-east-1:082451908674:listener/app/microservicesLB/489d84f8743cd112/87c4d1a445adb929</a>		

**Rules** **Tags**

**Listener rules (1) [Info](#)**  
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.

Rule limits				Actions	Add rule										
<input type="text"/> Filter rules				<a href="#">Edit</a>	<a href="#">Actions</a>										
<b>Listener rules (1) <a href="#">Info</a></b> 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.															
<table border="1"><thead><tr><th>Name tag</th><th>Priority</th><th>Conditions (If)</th><th>Actions (Then)</th><th>ARN</th></tr></thead><tbody><tr><td><input type="checkbox"/> Default</td><td>Last (default)</td><td>If no other rule applies</td><td><b>Forward to target group</b><ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul></td><td><a href="#">Edit</a></td></tr></tbody></table>						Name tag	Priority	Conditions (If)	Actions (Then)	ARN	<input type="checkbox"/> Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	<a href="#">Edit</a>
Name tag	Priority	Conditions (If)	Actions (Then)	ARN											
<input type="checkbox"/> Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"><li><a href="#">customer-tg-two</a>: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>	<a href="#">Edit</a>											

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vocabs/user3223197=Gupta\_Nancy @ 0824-5190-8674

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 **Add rule**

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

**Add rule [Info](#)**  
Define the rule and then review it in the context of the other rules on this listener.

**Listener details: HTTP:80**

**Name and tags [Info](#)**  
Tags can help you manage, identify, organize, search for and filter resources.

Name	<input type="text"/> Example, My workflow	<a href="#">Add additional tags</a>
------	---	-------------------------------------

[Cancel](#) **Next**

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia v vocabs/user3223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 Add rule

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

Define rule conditions Info

Requests reaching this rule must match all specified conditions for the rule to apply. At least 1 condition is required.

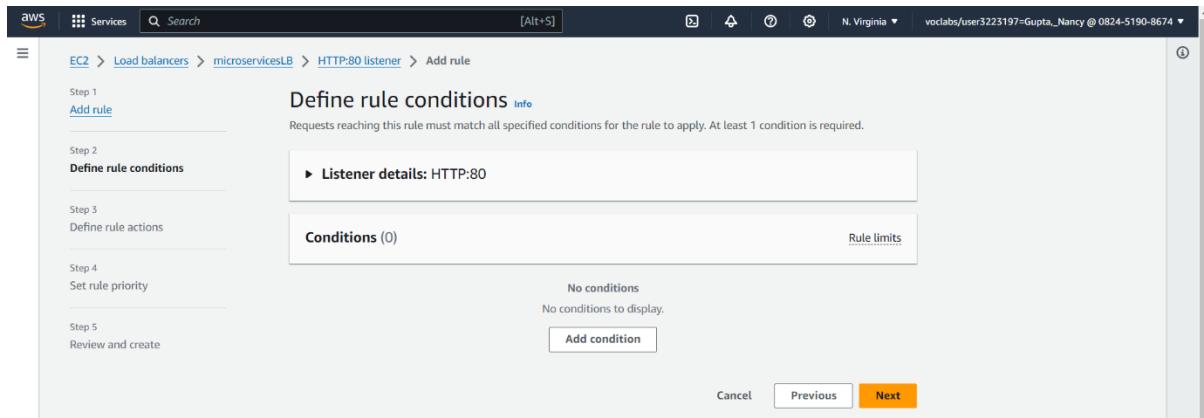
▶ Listener details: HTTP:80

Conditions (0) Rule limits

No conditions  
No conditions to display.

Add condition

Cancel Previous Next



CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia v vocabs/user3223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 Add rule

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

Define rule conditions

Add condition Rule limits

Rule condition types

Route traffic based on the condition type of each request. Each rule can include one or 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

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

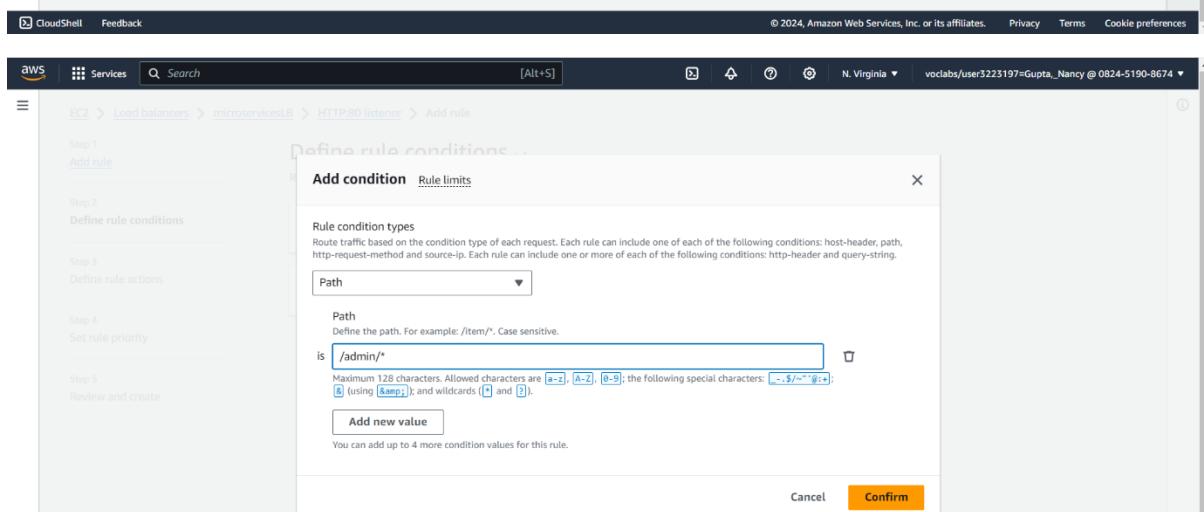
is /admin/\*

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



CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia v vocabs/user3223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 Add rule

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

Define rule conditions Info

Requests reaching this rule must match all specified conditions for the rule to apply. At least 1 condition is required.

▶ Listener details: HTTP:80

Conditions (1) Rule limits Edit Delete Add condition

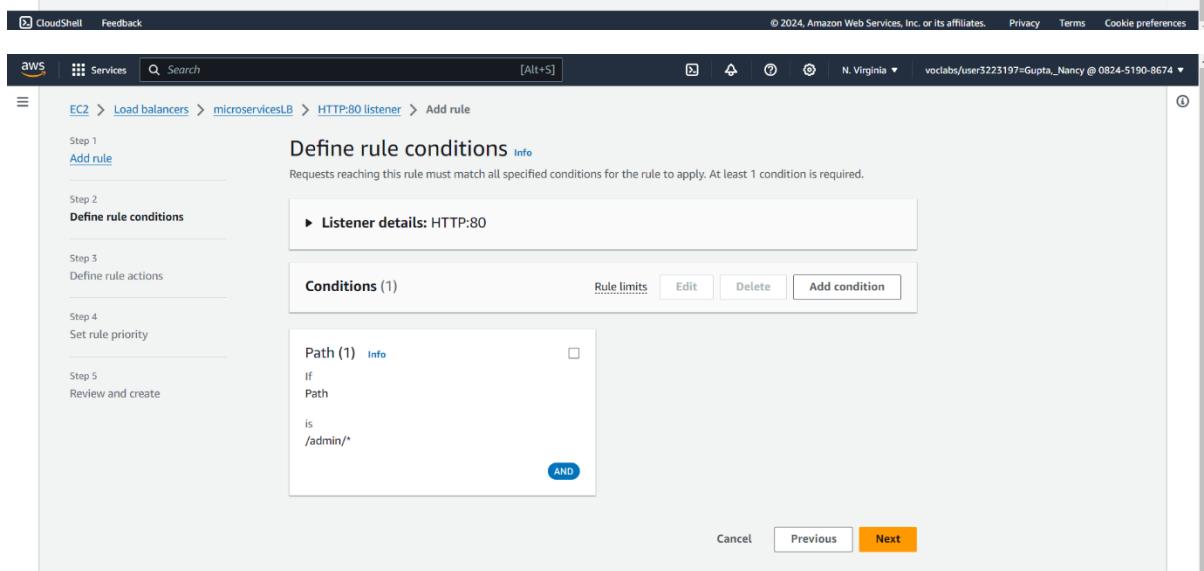
Path (1) Info

If Path

is /admin/\*

AND

Cancel Previous Next



aws Services Search [Alt+S] N. Virginia vclabs/user3223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 Add rule

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

### 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

employee-tg-two	HTTP	Weight	Percent
		1	100%
0-999			

Add target group

You can add up to 4 more target groups.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vclabs/user3223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 Add rule

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

### 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

employee-tg-two	HTTP	Weight	Percent
		1	100%
0-999			

Add target group

You can add up to 4 more target groups.

Group-level stickiness Info  
If a target group is sticky, requests routed to it remain in that target group for the duration of the session. Individual target stickiness is a configuration of the target group.

Turn on group-level stickiness

Cancel Previous Next

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia vclabs/user3223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:80 listener > Add rule

Step 1 Add rule

Step 2 Define rule conditions

Step 3 Define rule actions

Step 4 Set rule priority

Step 5 Review and create

### Set rule priority Info

Each rule has a priority. Rules are evaluated in priority order from the lowest value to the highest value. The default rule is evaluated last. You can change the priority of a non-default rule at any time. You can't change the priority of the default rule.

► Listener details: HTTP:80

#### Rule

Priority  
Rule priority controls the evaluation order of a rule within the listener's set of rules. You can leave gaps in priority numbers.

1	▼
---	---

1 - 50000

#### Listener rules (2) Info

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

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Set rule priority**

Step 5  
Review and create

**Priority**  
Rule priority controls the evaluation order of a rule within the listener's set of rules. You can leave gaps in priority numbers.

1
---

1 - 50000

**Listener rules (2) Info**

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.

Name tag	Priority	Conditions (If)	Actions (Then)	ARN
-	1	Path Pattern is /admin/*	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>employee-tg-two: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	Pending
Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>customer-tg-two: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	ARN

Cancel Previous Next

**CloudShell Feedback**

**Review and create**

Step 1 [Add rule](#)

Step 2 [Define rule conditions](#)

Step 3 [Define rule actions](#)

Step 4 [Set rule priority](#)

Step 5 [Review and create](#)

**Listener details: HTTP:80**

**Rule details**

Priority 1	Conditions (If) If request matches all: Path Pattern is /admin/*	Actions (Then) <b>Forward to target group</b> <ul style="list-style-type: none"><li>employee-tg-two: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>
---------------	--	--

Rule ARN  
Pending

**Rule tags (0)**

Tags can help you manage, identify, organize, search for and filter resources.

Key	Value
-----	-------

No tags found

Cancel Previous Create

**CloudShell Feedback**

**EC2 Dashboard**

EC2 Global View  
Events  
Console-to-Code [Preview](#)

**Instances**

Instances  
Instance Types  
Launch Templates  
Spot Requests  
Savings Plans  
Reserved Instances  
Dedicated Hosts  
Capacity  
Reservations [New](#)

**Images**

AMIs  
AMI Catalog

**Elastic Block Store**

Volumes

**Services**

Search [Alt+S]

**Successfully created rule on listener HTTP:80.**

EC2 > Load balancers > microservicesLB > HTTP:80 listener

**HTTP:80 Info**

**Details**

A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol:Port HTTP:80	Load balancer <a href="#">microservicesLB</a>	Default actions <b>Forward to target group</b> <ul style="list-style-type: none"><li>customer-tg-two: 1 (100%)</li><li>Group-level stickiness: Off</li></ul>
--------------------------	--	---

Listener ARN  
[arn:aws:elasticloadbalancing:us-east-1:082451908674:listener/app/microservicesLB/489d84f8743cd112/87c4d1a445adb929](#)

**Rules** **Tags**

**Listener rules (2) Info**

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.

Name tag	Priority	Conditions (If)	Actions (Then)	ARN
-	1	Path Pattern is /admin/*	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>employee-tg-two: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	Pending
Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>customer-tg-two: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	ARN

Rule limits Actions Add rule

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Protocol/Port: HTTP:80

Load balancer: microservicesLB

Default actions:

- Forward to target group
  - customer-tg-one: 1 (100%)
  - Group-level stickiness: Off

Listener rules (2)

Name tag	Priority	Conditions (If)	Actions (Then)	ARN
-	1	Path Pattern is /admin/*	Forward to target group <ul style="list-style-type: none"> <li>employee-tg-one: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	ARN
Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none"> <li>customer-tg-one: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>	ARN

#### 4. Add a second rule for the **HTTP:8080** listener. Define the following logic for this new rule:

- **IF Path is /admin/\***
- **THEN Forward to the employee-tg-one target group.**

The settings should be the same as shown in the following image:

Name tag	Priority	Conditions (If)	Actions (Then)
-	1	Path Pattern is /admin/*	Forward to target group <ul style="list-style-type: none"> <li>employee-tg-one: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>
Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none"> <li>customer-tg-one: 1 (100%)</li> <li>Group-level stickiness: Off</li> </ul>

Protocol/Port: HTTP:8080

Load balancer: microservicesLB

Default actions:

- Forward to target group
  - customer-tg-one: 1 (100%)
  - Group-level stickiness: Off

Listener rules (1)

Filter rules	Rule limits	Actions	Add rule
--------------	-------------	---------	----------

AWS Services Search [Alt+S] N. Virginia voclabs/user\$223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 Dashboard EC2 Global View Events Console-to-Code [Preview](#)

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations [New](#)

Images AMIs AMI Catalog

Elastic Block Store Volumes

CloudShell Feedback

routes requests to its registered targets.

Protocol:Port Load balancer Default actions  
HTTP:8080 microservicesLB Forward to target group  
[customer-tg-one](#) 1 (100%)  
Group-level stickiness: Off

Listener ARN arn:aws:elasticloadbalancing:us-east-1:082451908674:listener/app/microservicesLB/489d84f8743cd112/87ea893a809ff0aa

Rules Tags

Listener rules (1) [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
<input type="checkbox"/> Default	Last (default)	If no other rule applies	Forward to target group <a href="#">customer-tg-one</a> 1 (100%) Group-level stickiness: Off	<a href="#">Edit</a> ARN

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia voclabs/user\$223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:8080 listener > Add rule

Step 1 Add rule Step 2 Define rule conditions Step 3 Define rule actions Step 4 Set rule priority Step 5 Review and create

Add rule [Info](#)

Define the rule and then review it in the context of the other rules on this listener.

▶ Listener details: HTTP:8080

Name and tags [Info](#)

Tags can help you manage, identify, organize, search for and filter resources.

Name  Add additional tags

Cancel Next

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia voclabs/user\$223197=Gupta,\_Nancy @ 0824-5190-8674 ▾

EC2 > Load balancers > microservicesLB > HTTP:8080 listener > Add rule

Step 1 Add rule Step 2 Define rule conditions Step 3 Define rule actions Step 4 Set rule priority Step 5 Review and create

Define rule conditions [Info](#)

Requests reaching this rule must match all specified conditions for the rule to apply. At least 1 condition is required.

▶ Listener details: HTTP:8080

Conditions (0) Rule limits

No conditions  
No conditions to display.

Add condition

Cancel Previous Next

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS CloudFront Define rule conditions - Add condition step. The path condition is set to '/admin/\*'.

Step 1: Add rule

Step 2: Define rule conditions

Step 3: Define rule actions

Step 4: Set rule priority

Step 5: Review and create

Add condition Rule limits

Rule condition types

Path

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

is `/admin/*`

Maximum 128 characters. Allowed characters are `a-z`, `A-Z`, `0-9`; the following special characters: `- . _ / ~ ^ @ : &`; and wildcards `*` and `?`.

Add new value

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

Cancel Confirm

Screenshot of the AWS CloudFront Define rule conditions step, showing the condition added.

Step 1: Add rule

Step 2: Define rule conditions

Step 3: Define rule actions

Step 4: Set rule priority

Step 5: Review and create

Define rule conditions Info

Requests reaching this rule must match all specified conditions for the rule to apply. At least 1 condition is required.

Listener details: HTTP:8080

Conditions (1)

Path (1) info

If Path

is `/admin/*`

Cancel Previous Next

Screenshot of the AWS CloudFront Define rule actions step, showing the target group configuration.

Step 1: Add rule

Step 2: Define rule conditions

Step 3: Define rule actions

Step 4: Set rule priority

Step 5: Review and create

Define rule actions Info

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

Listener details: HTTP:8080

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

Weight	Percent		
employee-tg-one	HTTP	1	100%
Target type: IP, IPv4			

Add target group

You can add up to 4 more target groups.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Step 3 Define rule actions**

**Step 4 Set rule priority**

**Step 5 Review and create**

**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	Weight	Percent
employee-tg-one	HTTP	1 100%

[Add target group](#)  
You can add up to 4 more target groups.

**Group-level stickiness** [Info](#)  
If a target group is sticky, requests routed to it remain in that target group for the duration of the session. Individual target stickiness is a configuration of the target group.

Turn on group-level stickiness

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

**CloudShell Feedback**

**EC2 > Load balancers > microservicesLB > HTTP:8080 listener > Add rule**

**Step 1 Add rule**

**Step 2 Define rule conditions**

**Step 3 Define rule actions**

**Step 4 Set rule priority**

**Step 5 Review and create**

**Set rule priority** [Info](#)

Each rule has a priority. Rules are evaluated in priority order from the lowest value to the highest value. The default rule is evaluated last. You can change the priority of a non-default rule at any time. You can't change the priority of the default rule.

**Listener details: HTTP:8080**

**Rule**

**Priority**  
Rule priority controls the evaluation order of a rule within the listener's set of rules. You can leave gaps in priority numbers.

1
---

1 - 50000

**Listener rules (2) [Info](#)** [Rule limits](#)

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.

Name tag	Priority	Conditions (If)	Actions (Then)	ARN
	1			

[Filter rules](#)

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

**CloudShell Feedback**

**aws Services Search**

**Set rule priority**

**Step 5 Review and create**

**Priority**  
Rule priority controls the evaluation order of a rule within the listener's set of rules. You can leave gaps in priority numbers.

1
---

1 - 50000

**Listener rules (2) [Info](#)** [Rule limits](#)

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.

Name tag	Priority	Conditions (If)	Actions (Then)	ARN
-	1	Path Pattern is /admin/*	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>• <a href="#">employee-tg-one</a>: 1 (100%)</li> <li>• Group-level stickiness: Off</li> </ul>	Pending
Default	Last (default)	If no other rule applies	<b>Forward to target group</b> <ul style="list-style-type: none"> <li>• <a href="#">customer-tg-one</a>: 1 (100%)</li> <li>• Group-level stickiness: Off</li> </ul>	<a href="#">ARN</a>

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

**Review and create**

**Step 2: Define rule conditions**

**Step 3: Define rule actions**

**Step 4: Set rule priority**

**Step 5: Review and create**

**Listener details: HTTP:8080**

**Rule details**

Priority 1	Conditions (If) If request matches all: <b>Path Pattern</b> is /admin/*	Actions (Then) <b>Forward to target group</b> • <a href="#">employee-tg-one</a> 1 (100%) • Group-level stickiness: Off
---------------	---	---

Rule ARN  
*Pending*

**Rule tags (0)**

Tags can help you manage, identify, organize, search for and filter resources.

Key	Value
No tags found	

**Create**

**Successfully created rule on listener HTTP:8080.**

**HTTP:8080**

**Details**

A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol:Port HTTP:8080	Load balancer <a href="#">microservicesLB</a>	Default actions <b>Forward to target group</b> • <a href="#">customer-tg-one</a> 1 (100%) • Group-level stickiness: Off
----------------------------	--	--

Listener ARN  
[arn:aws:elasticloadbalancing:us-east-1:082451908674:listener/app/microservicesLB/489d84f8743cd112/87ea893a809ff0aa](#)

**Rules** **Tags**

**Listener rules (2)**

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.

**Add rule**

**Protocol:Port**  
HTTP:8080

**Load balancer**  
[microservicesLB](#)

**Default actions**  
**Forward to target group**  
• [customer-tg-one](#) 1 (100%)  
• Group-level stickiness: Off

**Listener ARN**  
[arn:aws:elasticloadbalancing:us-east-1:082451908674:listener/app/microservicesLB/489d84f8743cd112/87ea893a809ff0aa](#)

**Rules** **Tags**

**Listener rules (2)**

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.

**Add rule**

<b>Filter rules</b>			
	Name tag	Priority	Conditions (If)
<input type="checkbox"/>	-	1	<b>Path Pattern</b> is /admin/*
<input type="checkbox"/>	<b>Default</b>	Last (default)	<i>If no other rule applies</i>

**Actions (Then)**

	ARN
<b>Forward to target group</b>	<a href="#">employee-tg-one</a> 1 (100%)
	• Group-level stickiness: Off
<b>Forward to target group</b>	<a href="#">customer-tg-one</a> 1 (100%)
	• Group-level stickiness: Off

The screenshot shows the AWS Cloud9 interface with a code editor open. The file being edited is named 'create-customer-microservice-tg-two.json'. The code contains JSON configuration for an AWS Lambda function, specifying a role, memory, and timeout.

```

{
  "version": "0.1",
  "function_name": "customer-tg-two",
  "role": "arn:aws:iam::123456789012:lambda-role",
  "handler": "index.handler",
  "memory_size": 128,
  "timeout": 10
}

```

The screenshot shows the AWS Cloud9 interface with a code editor open. The file being edited is named 'create-customer-microservice-tg-one.json'. The code contains JSON configuration for an AWS Lambda function, specifying a role, memory, and timeout.

```

{
  "version": "0.1",
  "function_name": "customer-tg-one",
  "role": "arn:aws:iam::123456789012:lambda-role",
  "handler": "index.handler",
  "memory_size": 128,
  "timeout": 10
}

```

## Phase 7: Creating two Amazon ECS services

In this phase, you will create a service in Amazon ECS for each microservice. Although you could deploy both microservices to a single ECS service, for this project, it will be easier to manage the microservices independently if each is deployed to its own ECS service.

### Task 7.1: Create the ECS service for the *customer* microservice

1. In AWS Cloud9, create a new file named **create-customer-microservice-tg-two.json** in the deployment directory.

The screenshot shows a terminal window with the following details:

- File Path:** /home/ec2-user/.aws/lambda/functions/labuser-pem
- Logs:** The terminal displays deployment logs for a function named "labuser-pem". The logs include the beginning of an RSA private key, deployment configuration, and various AWS Lambda logs.
- Environment Variables:** The logs show environment variables being set for deployment, such as `VCS\_URL` and `VCS\_COMMIT\_ID`.
- Deployment Status:** The logs indicate successful deployment steps like "Deployment succeeded" and "Function deployed successfully".
- Metrics:** Metrics related to the deployment process are also present in the logs.

## **2. Paste the following JSON code into the file:**

{

```
"taskDefinition": "customer-microservice:REVISION-NUMBER",
"cluster": "microservices-serverlesscluster",
"loadBalancers": [
  {
    "targetGroupArn": "MICROSERVICE-TG-TWO-ARN",
    "containerName": "customer",
    "containerPort": 8080
  }
],
"desiredCount": 1,
"launchType": "FARGATE",
"schedulingStrategy": "REPLICA",
"deploymentController": {
  "type": "CODE_DEPLOY"
},
"networkConfiguration": {
  "awsvpcConfiguration": {
    "subnets": [
      "PUBLIC-SUBNET-1-ID"
    ],
    "securityGroups": [
      "SECURITY-GROUP-ID"
    ],
    "assignPublicIp": "ENABLED"
  }
}
```

```

    "PUBLIC-SUBNET-2-ID"

],
"securityGroups": [
    "SECURITY-GROUP-ID"
],
"assignPublicIp": "ENABLED"
}

}

```

```

{
    "taskDefinition": "customer-microservice:REVISION-NUMBER",
    "cluster": "microservices-serverlesscluster",
    "loadBalancers": [
        {
            "targetGroupArn": "MICROSERVICE-TG-TWO-ARN",
            "containerName": "customer",
            "containerPort": 8080
        }
    ],
    "desiredCount": 1,
    "launchType": "FARGATE",
    "schedulingStrategy": "REPLICAS",
    "deploymentController": {
        "type": "CODE_DEPLOY"
    },
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": [
                "PUBLIC-SUBNET-1-ID",
                "PUBLIC-SUBNET-2-ID"
            ],
            "securityGroups": [
                "SECURITY-GROUP-ID"
            ],
            "assignPublicIp": "ENABLED"
        }
    }
}

bash -c ip=10-16-10-153 echo
veclabs:/environment $ cd deployment
veclabs:/environment $ cd deployment
veclabs:/environment/deployment (dev) $ touch create-customer-microservice-tg-two.json
veclabs:/environment/deployment (dev) $ 

```

### 3. Edit the `create-customer-microservice-tg-two.json` file:

- Replace **REVISION-NUMBER** with the number of the latest revision of the *customer-microservice* task definition that is registered with Amazon ECS.
  - If this is the first time that you are completing this step, the revision number should be 1.
  - If you are repeating this step, find the latest revision number in the Amazon ECS console by choosing Task definitions, and then choosing *customer-microservice*.

Screenshot of the AWS Cloud Console showing the ECS service page. The search bar at the top contains 'ECS'. The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, and Instances (with sub-links like Instances, Instance Types, Launch Templates, etc.). The main content area displays a list of services under 'Services (26)'. The first item is 'Elastic Container Service' with a brief description: 'Highly secure, reliable, and scalable way to run containers'. Below it are 'Batch', 'AWS FIS', and 'EC2'. A 'Features' section follows with a 'Get started' button.

Screenshot of the AWS Cloud Console showing the 'Task definitions' page for the Amazon Elastic Container Service. The left sidebar includes links for Clusters, Namespaces, Task definitions (which is selected), and Account settings. The main content area shows a table of task definitions. Two entries are listed: 'customer-microservice' and 'employee-microservice', both marked as 'ACTIVE'.

Screenshot of the AWS CloudShell interface. The terminal window shows the command 'aws lambda invoke --function labuser-pem --payload file://create-customer-microservice.json > customer-tg-two.log'. The output log file 'customer-tg-two.log' is displayed below, showing the deployment process for the 'customer-microservice' task definition.

```

{
    "taskDefinitionArn": "customer-microservice:1",
    "cluster": "microservices-serverlessCluster",
    "loadBalancers": [
        {
            "targetGroupArn": "MICROSERVICE-TG-TWO-ARN",
            "containerName": "customer",
            "containerPort": 8080
        }
    ],
    "desiredCount": 1,
    "launchType": "FARGATE",
    "schedulingStrategy": "REPLICAS",
    "deploymentController": {
        "type": "CODE_DEPLOY"
    },
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": [
                "PUBLIC-SUBNET-1-ID",
                "PUBLIC-SUBNET-2-ID"
            ],
            "securityGroups": [
                "SECURITY-GROUP-ID"
            ],
            "assignPublicIp": "ENABLED"
        }
    }
}

```

- Replace **MICROSERVICE-TG-TWO-ARN** with the actual ARN of the **customer-tg-two** target group.

Screenshot of the AWS EC2 Home page in the US East (N. Virginia) Region.

**Resources**

- Instances (running): 2
- Auto Scaling Groups: 0
- Dedicated Hosts: 0
- Elastic IPs: 0
- Instances: 2
- Key pairs: 1
- Load balancers: 1
- Placement groups: 0
- Security groups: 6
- Snapshots: 0
- Volumes: 2

**Account attributes**

- Default VPC**: vpc-000e564836083dd8
- Settings**: Data protection and security, Zones, EC2 Serial Console, Default credit specification, Console experiments

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Service health**

AWS Health Dashboard

**Explore AWS**

**Amazon GuardDuty Malware Protection**

GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. [Learn more](#)

**10 Things You Can Do Today to Reduce AWS Costs**

Explore how to effectively manage your AWS costs without compromising on performance or capacity. [Learn more](#)

<https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroups>

Screenshot of the AWS Target Groups page for the 'customer-tg-two' target group.

**customer-tg-two**

**Details**

Copy ARN of target group to clipboard	Target type: IP	Protocol: Port: HTTP: 8080	Protocol version: HTTP1	VPC: vpc-043e6f0ce741fda03
IP address type: IPv4	Load balancer: microservicesLB			
0 Total targets	0 Healthy	0 Unhealthy	0 Unused	0 Initial
	0 Anomalous			0 Draining

**Targets** | **Monitoring** | **Health checks** | **Attributes** | **Tags**

**Registered targets (0)**

Anomaly mitigation: Not applicable

<https://us-east-1.console.aws.amazon.com/ec2/target-groups/customer-tg-two?region=us-east-1>

Screenshot of the AWS Cloud9 IDE showing the 'create-customer-microservice' file.

```

1  {
2    "taskDefinition": "customer-microservice:1",
3    "cluster": "microservices-serverlesscluster",
4    "loadBalancers": [
5      {
6        "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-two/f31332458bb3be4",
7        "containerName": "customer",
8        "containerPort": "8080"
9      }
10    ],
11    "desiredCount": 1,
12    "launchType": "FARGATE",
13    "schedulingStrategy": "REPLICAS",
14    "deploymentController": {
15      "type": "CODE_DEPLOY"
16    },
17    "networkConfiguration": {
18      "awsvpcConfiguration": {
19        "subnets": [
20          "PUBLIC-SUBNET-1-ID",
21          "PUBLIC-SUBNET-2-ID"
22        ],
23        "securityGroups": [
24          "SECURITY-GROUP-ID"
25        ],
26        "assignPublicIp": "ENABLED"
27      }
28    }
29  }

```

bash - ip-10-16-10-153.ex

```

veclabs:/environment $ cd deployment
veclabs:/environment $ cd deployment
veclabs:/environment/deployment (dev) $ touch create-customer-microservice-tg-two.json
veclabs:/environment/deployment (dev) $ 

```

- Replace PUBLIC-SUBNET-1-ID with the actual subnet ID of Public Subnet1.

**VPC dashboard**

Create VPC Launch EC2 Instances

Note: Your Instances will launch in the US East region.

### Resources by Region

You are using the following Amazon VPC resources

VPCs	US East 2	NAT Gateways	US East 0
Subnets	US East 10	VPC Peering Connections	US East 0
Route Tables	US East 5	Network ACLs	US East 2
Internet Gateways	US East 2	Security Groups	US East 6
Egress-only Internet Gateways	US East 0	Customer Gateways	US East 0
Carrier gateways			
DHCP option sets			
Elastic IPs			
Managed prefix lists			
Endpoints			
Endpoint services			
NAT gateways			
Peering connections			

**Service Health**

View complete service health details

**Settings**

Zones

Console Experiments

**Additional Information**

VPC Documentation All VPC Resources Forums Report an Issue

**AWS Network Manager**

AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity management, network monitoring and troubleshooting, IP management, and network security and governance.

Get started with Network Manager

**VPC dashboard**

Create Subnet

**Subnets (1/10) info**

Find resources by attribute or tag

Name	Subnet ID	State	VPC	IPv4 CIDR
Private Subnet 2	subnet-0c5349c587bed3954	Available	vpc-043e6f0ce741fda03   LabVPC	10.16.40.0/24
Public Subnet2	subnet-0e50293004d9e1c8	Available	vpc-043e6f0ce741fda03   LabVPC	10.16.20.0/24
-	subnet-0924e843100df916e	Available	vpc-000e564836083ddaa8	172.31.16.0/20
-	subnet-02429e95e434a6af	Available	vpc-000e564836083ddaa8	172.31.64.0/20
-	subnet-0d45103709ca680e1	Available	vpc-000e564836083ddaa8	172.31.48.0/20
<b>Public Subnet1</b>	<b>subnet-09458d72d02ae53ea</b>	<b>Available</b>	<b>vpc-043e6f0ce741fda03   LabVPC</b>	<b>10.16.10.0/24</b>

**subnet-09458d72d02ae53ea / Public Subnet1**

Details Flow logs Route table Network ACL CIDR reservations Sharing Tags

**Details**

Subnet ID subnet-09458d72d02ae53ea	Subnet ARN arn:aws:ec2:us-east-1:082451908674:subnet:subnet-09458d72d02ae53ea	State Available	IPv4 CIDR 10.16.10.0/24
Available IPv4 addresses 248		Availability Zone us-east-1a	Availability Zone ID use1-az4

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

File Edit Find Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

MicroservicesIDE - home

- deployment
  - appspec-customer.yaml
  - appspec-employee.yaml
- create-customer-microservice
- taskdef-customer.json
- taskdef-employee.json

aws

- microservices
  - customer
  - employee
  - app
  - public
  - css
  - img
  - js
  - views
- Dockerfile
- index.js
- package-lock.json
- package.json
- labuser.pem
- README.md

bash - ip-10-16-10-153.e.x

```

1  "taskDefinition": "customer-microservice:1",
2   "cluster": "microservices-serverlesscluster",
3   "loadBalancers": [
4     {
5       "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-two/f31332458bb3be42",
6       "containerName": "customer",
7       "containerPort": "8080"
8     }
9   ],
10  "desiredCount": 1,
11  "launchType": "FARGATE",
12  "schedulingStrategy": "REPLICAS",
13  "deploymentController": {
14    "type": "CODE_DEPLOY"
15  },
16  "networkConfiguration": {
17    "awspcConfiguration": {
18      "subnets": [
19        "subnet-09458d72d02ae53ea",
20        "PUBLIC-SUBNET-2-ID"
21      ],
22      "securityGroups": [
23        "SECURITY-GROUP-ID"
24      ],
25      "assignPublicIp": "ENABLED"
26    }
27  }
28}
29

```

bash - ip-10-16-10-153.e.x

```

veclabs:/environment $ cd deployment
veclabs:/environment $ cd deployment
veclabs:/environment/deployment (dev) $ touch create-customer-microservice-tg-two.json
veclabs:/environment/deployment (dev) $ []

```

- Replace PUBLIC-SUBNET-2-ID with the actual subnet ID of Public Subnet2.

The screenshot shows the AWS Subnets (1/10) page. The left sidebar includes sections for VPC dashboard, EC2 Global View, Filter by VPC (with a dropdown for 'Select a VPC'), Virtual private cloud, Your VPCs, Subnets (selected), Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main content area displays a table of subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR
Private Subnet 2	subnet-0c5349c587bed3954	Available	vpc-043e6f0ce741fda03   LabVPC	10.16.40.0/24
<input checked="" type="checkbox"/> Public Subnet2	subnet-0e5c0293004d9e1c8	Available	vpc-043e6f0ce741fda03   LabVPC	10.16.20.0/24
-	subnet-0934e8431004f916e	Available	vpc-000e564836083ddaa8	172.31.16.0/20
-	subnet-02429e95e45a46a6f	Available	vpc-000e564836083ddaa8	172.31.64.0/20
-	subnet-0d45103709ca680e1	Available	vpc-000e564836083ddaa8	172.31.48.0/20
Public Subnet1	subnet-09458d72d02ae53ea	Available	vpc-043e6f0ce741fda03   LabVPC	10.16.10.0/24

Below the table, a specific subnet is selected: **subnet-0e5c0293004d9e1c8 / Public Subnet2**. The Details tab is active, showing the following information:

Subnet ID	Subnet ARN	State	IPv4 CIDR
subnet-0e5c0293004d9e1c8	arn:aws:ec2:us-east-1:082451908674:subnet/subnet-0e5c0293004d9e1c8	Available	10.16.20.0/24
Available IPv4 addresses	251	Availability Zone	Availability Zone ID
		us-east-1b	use1-az6



The screenshot shows the AWS CodeCommit interface with the following details:

- File Explorer:** On the left, there's a tree view of the repository structure:
  - MicroservicesIDE - /home/ec2-user
  - deployment
  - appspec-customer.yaml
  - appspec-employee.yaml
  - create-customer-microservice (selected)
  - taskdef-customer.json
  - taskdef-employee.json
  - microservices
  - customer
  - employee
  - app
  - node\_modules
  - public
  - css
  - img
  - js
  - views
  - Dockerfile
  - JS index.js
  - package-lock.json
  - package.json
  - labuser.pem
  - README.md
- Code Editor:** The main area displays the content of the selected file, `create-customer-microservice`. The code defines a task definition for a Lambda function deployed to an AWS Lambda layer named `customer-lambda-layer`. It uses an AWS Lambda layer named `customer-lambda-layer` and an AWS Lambda layer named `customer-lambda-layer`. The task definition includes a container name (`customer`), port (`8080`), and a target group (`arn:aws:elasticloadbalancing:us-east-1:882451988574:targetgroup/customer-tg-two/f31332458bb3be42`). The deployment strategy is set to `FARGATE` with `REPLICAS` and `CODE_DEPLOY` as the deployment controller. Network configuration specifies a subnet (`subnet-0458d72d02ae53ea` and `subnet-0e5c0293004d9e1c8`) and a security group (`SECURITY-GROUP-ID`). Public IP assignment is enabled.
- Terminal:** At the bottom, a terminal window titled "bash - \*ip-10-16-10-153.ec2.internal" shows the command `cd deployment` being run in the `create-customer-microservice` directory.
- Header:** The top bar includes the AWS logo, navigation links (File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, Run), and user information (Share, Settings).

- Replace **SECURITY-GROUP-ID** with the actual security group ID of *microservices-sg*.
  - Save the changes.

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. The main pane displays a table of security groups, and a detailed view of the 'microservices-sg' group is shown in the bottom right.

**Security Groups (1/6) Info**

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0d3dc7a32965ee9fa	c110323a260559816548437t1w0824...	vpc-043e6f0ce741fda03	Enable inb...
DBSecurityGroup	sg-012f4abba21b3b3d8	DBSecurityGroup	vpc-043e6f0ce741fda03	Enable acc...
<input checked="" type="checkbox"/>	sg-091ac242028763d75	microservices-sg	vpc-043e6f0ce741fda03	Security gr...
-	sg-0ce210fe840097bec	default	vpc-043e6f0ce741fda03	default VP...
aws-rnldnq-Mirrns	sg-063haaa137h779h83	aws-rnldnq-Mirrnservices1NF-8d17a06	vnr-043e6f0ce741fda03	Security or...

**Details**

Security group name microservices-sg	Security group ID sg-091ac242028763d75	Description Security group for microservices application	VPC ID vpc-043e6f0ce741fda03
Owner 82451908674	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

```

1  {
2    "taskDefinition": "customer-microservice:1",
3    "cluster": "microservices-serverlesscluster",
4    "loadBalancers": [
5      {
6        "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:882451908674:targetgroup/customer-tg-two/f31332458bb3be42",
7        "containerName": "customer",
8        "containerPort": 8080
9      }
10   ],
11   "desiredCount": 1,
12   "launchType": "FARGATE",
13   "schedulingStrategy": "REPLICA",
14   "deploymentController": {
15     "type": "CODE_DEPLOY"
16   },
17   "networkConfiguration": {
18     "awsvpcConfiguration": {
19       "subnets": [
20         "subnet-09458072d0aae53ea",
21         "subnet-0e5c029300ad9e1c8"
22       ],
23       "securityGroups": [
24         "-sg-091ac242028763d75"
25       ],
26       "assignPublicIp": "ENABLED"
27     }
28   }
29 }

bash - ip-10-16-10-153 ~ x

veclabs:/environment $ cd deployment
veclabs:/environment $ cd deployment
veclabs:/environment/deployment (dev) $ touch create-customer-microservice-tg-two.json
veclabs:/environment/deployment (dev) $ ]

```

#### 4. To create the Amazon ECS service for the *customer* microservice, run the following commands:

cd ~/environment/deployment

```

1  {
2    "taskDefinition": "customer-microservice:1",
3    "cluster": "microservices-serverlesscluster",
4    "loadBalancers": [
5      {
6        "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:882451908674:targetgroup/customer-tg-two/f31332458bb3be42",
7        "containerName": "customer",
8        "containerPort": 8080
9      }
10   ],
11   "desiredCount": 1,
12   "launchType": "FARGATE",
13   "schedulingStrategy": "REPLICA",
14   "deploymentController": {
15     "type": "CODE_DEPLOY"
16   },
17   "networkConfiguration": {
18     "awsvpcConfiguration": {
19       "subnets": [
20         "subnet-09458072d0aae53ea",
21         "subnet-0e5c029300ad9e1c8"
22       ],
23       "securityGroups": [
24         "-sg-091ac242028763d75"
25       ],
26       "assignPublicIp": "ENABLED"
27     }
28   }
29 }

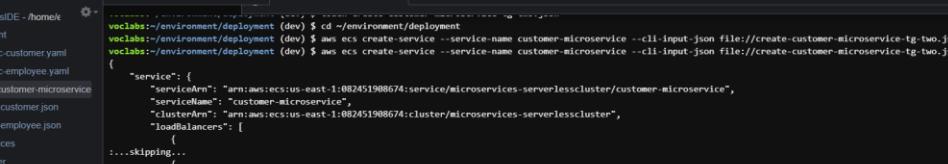
bash - ip-10-16-10-153 ~ x

veclabs:/environment $ cd deployment
veclabs:/environment $ cd deployment
veclabs:/environment/deployment (dev) $ touch create-customer-microservice-tg-two.json
veclabs:/environment/deployment (dev) $ cd ~/environment/deployment
veclabs:/environment/deployment (dev) $ ]

```

aws ecs create-service --service-name customer-microservice --cli-input-json file://create-customer-microservice-tg-two.json

**Troubleshooting tip:** If you are repeating this step and previously created the ECS service, you might receive an error about the creation of the service not being idempotent. To resolve this error, force delete the service from the Amazon ECS console, wait for it to drain, and then run the commands again.



The screenshot shows the AWS Lambda function editor with the following details:

- File Path:** /home/e/lambda/functions/create-customer-microservice
- Function Name:** create-customer-microservice
- Runtime:** Node.js 14.x
- Handler:** index.handler
- Code Content:**

```
const AWS = require('aws-sdk');
const https = require('https');

const region = 'us-east-1';
const endpoint = 'https://lambda.' + region + '.amazonaws.com';

AWS.config.update({region: region});

exports.handler = async (event) => {
    const { name } = event;
    const params = {
        service: {
            "serviceName": "customer-microservice",
            "clusterArn": "arn:aws:ecs:us-east-1:082451908674:cluster/microservices-serverlesscluster",
            "loadBalancers": [
                {
                    "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-two/f31332458bb3be42",
                    "containerName": "customer"
                }
            ]
        }
    };

    const response = await https.get({
        url: endpoint + '/2015-03-31/functions/' + process.env.FUNCTION_NAME + '/invocations',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(params)
    });

    const data = response.data;
    console.log(data);
}
```

The screenshot shows the AWS Lambda CodeWhisperer interface. On the left, there's a sidebar with project navigation and AWS services like CloudWatch Metrics and Lambda. The main area has tabs for 'labuser.pem' and 'create-customer-microservice'. The code editor displays the following function code:

```
    "stabilityStatusAt": "2024-04-28T17:49:09.862880+00:00",
    "tags": []
  },
  "roleArn": "arn:aws:iam::082451908674:role/aws-service-role/ecs.amazonaws.com/AWSServiceRoleForECs",
  "events": [],
  "createdAt": "2024-04-28T17:49:09.862800+00:00",
  "placementConstraints": [],
  "placementStrategy": [],
  "networkConfiguration": {
    "asyncConfiguration": {
      "subnets": [
        "subnet_0e5c029300d4e1c8",
        "subnet_09458d7d02ae53ea"
      ],
      "securityGroups": [
        "sg_091ac242028763d75"
      ],
      "assignPublicIp": "ENABLED"
    }
  },
  "healthCheckGracePeriodSeconds": 0,
  "schedulingStrategy": "REPLICAS",
  "deploymentController": {
    "type": "CODE_DEPLOY"
  },
  "createdBy": "arn:aws:iam::082451908674:role/voclabs",
  "enableCSManagedTags": false,
  "propagateTags": "NONE",
  "enableExecuteCommand": false
}
}

END

```

At the bottom, the terminal window shows the command: `voclabs-->/environment/deployment (dev) $`.

## Task 7.2: Create the Amazon ECS service for the *employee* microservice

## **1. Create an Amazon ECS service for the *employee* microservice.**

```
----BEGIN RSA PRIVATE KEY-----  
MIIEpQIBAAQEAjwvA5mPw+Ht...  
-----END RSA PRIVATE KEY-----  
bash -*p-10-16-10-153 x  
vocabs:-/environment $ cd ~/environment/deployment/  
vocabs:/environment/deployment (dev) $ touch create-employee-microservice-tg-two.json  
vocabs:/environment/deployment [1]
```

- Copy the JSON file that you created for the *customer* microservice, and name it `create-employee-microservice-tg-two.json`. Save it in the same directory.

The screenshot shows the AWS Lambda Code Editor interface. The left sidebar displays a file tree for a project named 'MicroservicesIDE'. The main editor area contains a file named 'labsuser.pem' with the following content:

```
1  {
2     "taskDefinition": "customer-microservice:1",
3     "cluster": "microservices-serverlesscluster",
4     "loadBalancers": [
5         {
6             "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/customer-tg-two/f31332458bb3be42",
7             "containerName": "customer",
8             "containerPort": 8080
9         }
10    ],
11    "desiredCount": 1,
12    "launchType": "FARGATE",
13    "schedulingStrategy": "REPLICA",
14    "deploymentController": {
15        "type": "CODE_DEPLOY"
16    },
17    "networkConfiguration": {
18        "awsvpcConfiguration": {
19            "subnets": [
20                "subnet-09458d7d402ae53ea",
21                "subnet-0e5c029304d9e1cb"
22            ],
23            "securityGroups": [
24                "sg-091ac242028763d75"
25            ],
26            "assignPublicIp": "ENABLED"
27        }
28    }
29 }
```

Below the editor, a terminal window titled 'bash - "p-10-16-10-153 ex'" shows the command:

```
volcabls:-/environment $ cd ~/environment/deployment/
volcabls:-/environment/deployment (dev) $ touch create-employee-microservice-tg-two.json
volcabls:-/environment/deployment (dev) $
```

- **Modify the `create-employee-microservice-tg-two.json` file:**

- On line 2, change customer-microservice to employee-microservice and also update the revision number.

- On line 6, enter the ARN of the employee-tg-two target group.

**Tip:** Don't just change customer to employee on this line. The ARN is unique in other ways.

```

1  {
2    "taskDefinition": "employee-microservice:1",
3    "cluster": "microservices-serverlesscluster",
4    "loadBalancers": [
5      {
6        "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-two/963c5e2cc8a65bb2",
7        "containerName": "customer",
8        "containerPort": 8080
9      }
10   ],
11   "desiredCount": 1,
12   "launchType": "FARGATE",
13   "schedulingStrategy": "REPLICAS",
14   "deploymentController": {
15     "type": "CODE_DEPLOY"
16   },
17   "networkConfiguration": {
18     "awsvpcConfiguration": {
19       "subnets": [
20         "subnet-09458d72d02ae53e8",
21         "subnet-0e5c029304ad9e1c8"
22       ],
23       "securityGroups": [
24         "sg-091ac242028763d75"
25       ],
26       "assignPublicIp": "ENABLED"
27     }
28   }
29 }

```

bash - \*ip-10-16-10-153.ec2.internal\*  
`veclabs:-/environment \$ cd ~/environment/deployment/  
`veclabs:-/environment/deployment (dev) \$ touch create-employee-microservice-tg-two.json  
`veclabs:-/environment/deployment (dev) \$`

- On line 7, change customer to employee

- Save the changes.

```

1  {
2    "taskDefinition": "employee-microservice:1",
3    "cluster": "microservices-serverlesscluster",
4    "loadBalancers": [
5      {
6        "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-two/963c5e2cc8a65bb2",
7        "containerName": "employee",
8        "containerPort": 8080
9      }
10   ],
11   "desiredCount": 1,
12   "launchType": "FARGATE",
13   "schedulingStrategy": "REPLICAS",
14   "deploymentController": {
15     "type": "CODE_DEPLOY"
16   },
17   "networkConfiguration": {
18     "awsvpcConfiguration": {
19       "subnets": [
20         "subnet-09458d72d02ae53e8",
21         "subnet-0e5c029304ad9e1c8"
22       ],
23       "securityGroups": [
24         "sg-091ac242028763d75"
25       ],
26       "assignPublicIp": "ENABLED"
27     }
28   }
29 }

```

bash - \*ip-10-16-10-153.ec2.internal\*  
`veclabs:-/environment \$ cd ~/environment/deployment/  
`veclabs:-/environment/deployment (dev) \$ touch create-employee-microservice-tg-two.json  
`veclabs:-/environment/deployment (dev) \$`

## 2. Run the appropriate AWS CLI command to create the service in Amazon ECS.

**Note:** If you go to the Amazon ECS console and look at the services in the cluster, you might see *0/1 Task running*, as shown in the following image. This is expected for now because you haven't launched task sets for these services yet.

```

Go to Anything (Ctrl-P) labuser.pem (1) create-employee-microservice.x
File Edit Find View Go Run Tools Window Support Preview Run
aws
MicroservicesIDE - /home/le
└── deployment
    ├── appspec-customer.yaml
    └── appspec-employee.yaml
    └── create-employee-microservice
    └── create-employee-microservice.json
    └── taskdef-employee.json
    └── taskdef-employee.json
└── microservices
    ├── customer
    └── employee
        ├── app
        └── node_modules
            └── public
                ├── css
                ├── img
                ├── js
                └── views
                    └── Dockerfile
            └── JS index.js
            └── package-lock.json
            └── package.json
        └── labsuser.pem
    └── README.md
bash - *ip-10-16-10-153.e x
vclabs:-/environment $ cd ~/environment/deployment/
vclabs:-/environment/deployment (dev) $ touch create-employee-microservice-tg-two.json
vclabs:-/environment/deployment (dev) $ cd ~/environment/deployment/
vclabs:-/environment/deployment (dev) $ aws ecs create-service --service-name employee-microservice --cli-input-json file://create-employee-microservice-tg-two.json
vclabs:-/environment/deployment (dev) $ aws ecs create-service --service-name employee-microservice --cli-input-json file://create-employee-microservice-tg-two.json
{
    "service": {
        "serviceName": "arn:aws:ecs:us-east-1:082451908674:service/microservices-serverlesscluster/employee-microservice",
        "serviceName": "employee-microservice",
        "clusterArn": "arn:aws:ecs:us-east-1:082451908674:cluster/microservices-serverlesscluster",
        "loadBalancers": [
            {
                "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:082451908674:targetgroup/employee-tg-two/963c5e2cc8a65bb2",
                "containerName": "employee",
                "containerPort": 8080
            }
        ],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "1.4.0",
        "platformVersion": "1.4.0"
    },
    "skipPlatformVersion": true
}
vclabs:-/environment/deployment (dev) $

```

```

Go to Anything (Ctrl-P) labuser.pem (1) create-employee-microservice.x
File Edit Find View Go Run Tools Window Support Preview Run
aws
MicroservicesIDE - /home/le
└── deployment
    ├── appspec-customer.yaml
    └── appspec-employee.yaml
    └── create-employee-microservice
    └── create-employee-microservice.json
    └── taskdef-employee.json
    └── taskdef-employee.json
└── microservices
    ├── customer
    └── employee
        ├── app
        └── node_modules
            └── public
                ├── css
                ├── img
                ├── js
                └── views
                    └── Dockerfile
            └── JS index.js
            └── package-lock.json
            └── package.json
        └── labsuser.pem
    └── README.md
bash - *ip-10-16-10-153.e x
vclabs:-/environment $ cd ~/environment/deployment/
vclabs:-/environment/deployment (dev) $ touch create-employee-microservice-tg-two.json
vclabs:-/environment/deployment (dev) $ cd ~/environment/deployment/
vclabs:-/environment/deployment (dev) $ aws ecs create-service --service-name employee-microservice --cli-input-json file://create-employee-microservice-tg-two.json
vclabs:-/environment/deployment (dev) $ aws ecs create-service --service-name employee-microservice --cli-input-json file://create-employee-microservice-tg-two.json
{
    "service": {
        "awsVpcConfiguration": {
            "subnets": [
                "subnet-e5c029300d9e1c8",
                "subnet-09458d72d02ae5ea"
            ],
            "securityGroups": [
                "sg-091a242028763d75"
            ],
            "assignPublicIp": "ENABLED"
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "CODE_DEPLOY"
        },
        "createdBy": "arn:aws:lambda:082451908674:role:vclabs",
        "enableECSManagedTags": false,
        "propagateTags": "NONE",
        "enableExecuteCommand": false
    },
    "skipPlatformVersion": true
}
vclabs:-/environment/deployment (dev) $

```

Tell us what you think X

Amazon Elastic Container Service Search

Amazon Elastic Container Service > Clusters > microservices-serverlesscluster > Services

**microservices-serverlesscluster**

Update cluster Delete cluster

Cluster overview			
ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:us-east-1:082451908674:cluster/microservices-serverlesscluster	Active	CloudWatch monitoring	-
Services		Tasks	
Draining	Active	Pending	Running
-	2	-	-

**Services (2) Info**

Create

Filter launch type	Filter service type
Any launch type	Any service type

1 < > (1)

Documentation Feedback

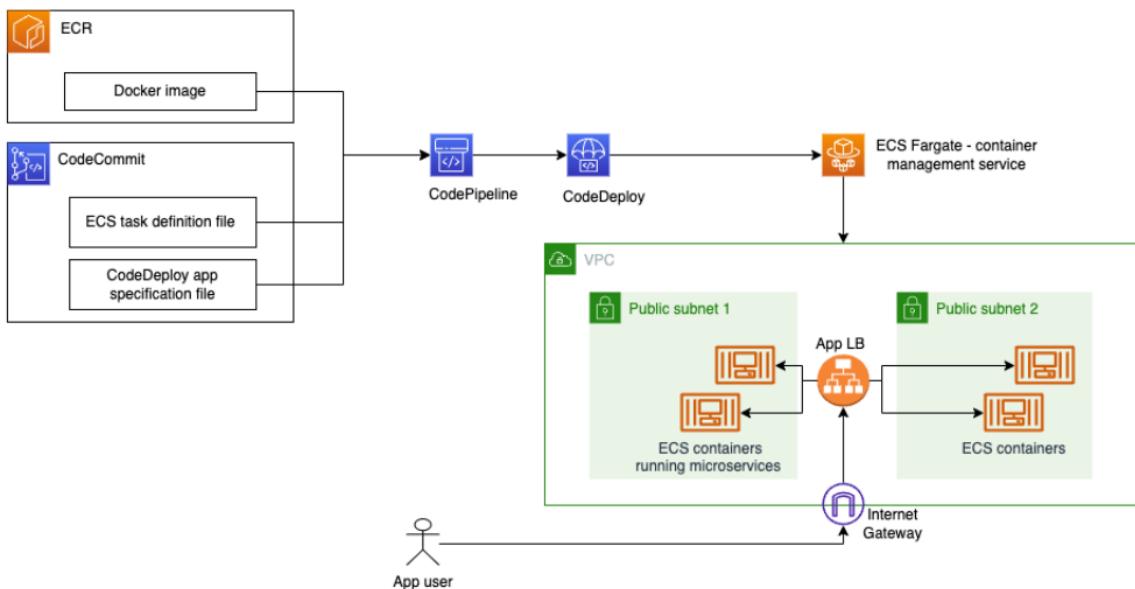
Discover products CloudShell

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Phase 8: Configuring CodeDeploy and CodePipeline

Now that you have defined the Application Load Balancer, target groups, and the Amazon ECS services that comprise the infrastructure that you will deploy your microservices to, the next step is to define the CI/CD pipeline to deploy the application.

The following diagram illustrates the role of the pipeline in the solution that you are building.



*Diagram description:* The pipeline will be invoked by updates to CodeCommit, where you have stored the ECS task definition files and the CodeDeploy AppSpec files. The pipeline can also be invoked by updates to one of the Docker image files that you have stored in Amazon ECR. When invoked, the pipeline will call the CodeDeploy service to deploy the updates. CodeDeploy will take the necessary actions to deploy the updates to the green environment. Assuming that no errors occur, the new task set will replace the existing task set.

### Task 8.1: Create a CodeDeploy application and deployment groups

A CodeDeploy application is a collection of deployment groups and revisions. A deployment group specifies an Amazon ECS service, load balancer, optional test listener, and two target groups. A group specifies when to reroute traffic to the replacement task set, and when to terminate the original task set and Amazon ECS application after a successful deployment.

1. Use the CodeDeploy console to create a CodeDeploy application with the name microservices that uses Amazon ECS as the compute platform.

**Tip:** See [Create an Application for an Amazon ECS Service Deployment \(Console\)](#) in the *AWS CodeDeploy User Guide*.

**Important:** *DON'T* create a deployment group yet. You will do that in the next step.

The top screenshot shows the AWS Services search bar with 'codedeploy' entered. The search results show 'Services (20)' and 'Features (19)'. Under 'Services', 'CodeDeploy' is listed as 'Automate Code Deployments'. The bottom screenshot shows the AWS CodeDeploy landing page with the heading 'AWS CodeDeploy' and a large call-to-action button 'Create AWS CodeDeploy deployment'.

Screenshot of the AWS CodeDeploy 'Create application' form. The application name is 'microservices' and the compute platform is 'Amazon ECS'. A 'Create application' button is visible.

Screenshot of the AWS CodeDeploy application details page for 'microservices'. It shows the application details and a deployment groups section with a 'Create deployment group' button.

## 2. Create a CodeDeploy deployment group for the *customer* microservice.

- On the *microservices* application detail page, choose the **Deployment groups** tab.

Screenshot of the AWS CodeDeploy application details page for 'microservices'. The 'Deployment groups' tab is selected, showing a table with one row: Name (Customer), Status (Unknown), Last attempted deploy... (N/A), Last successful deploy... (N/A), and Trigger count (0). A 'Create deployment group' button is visible.

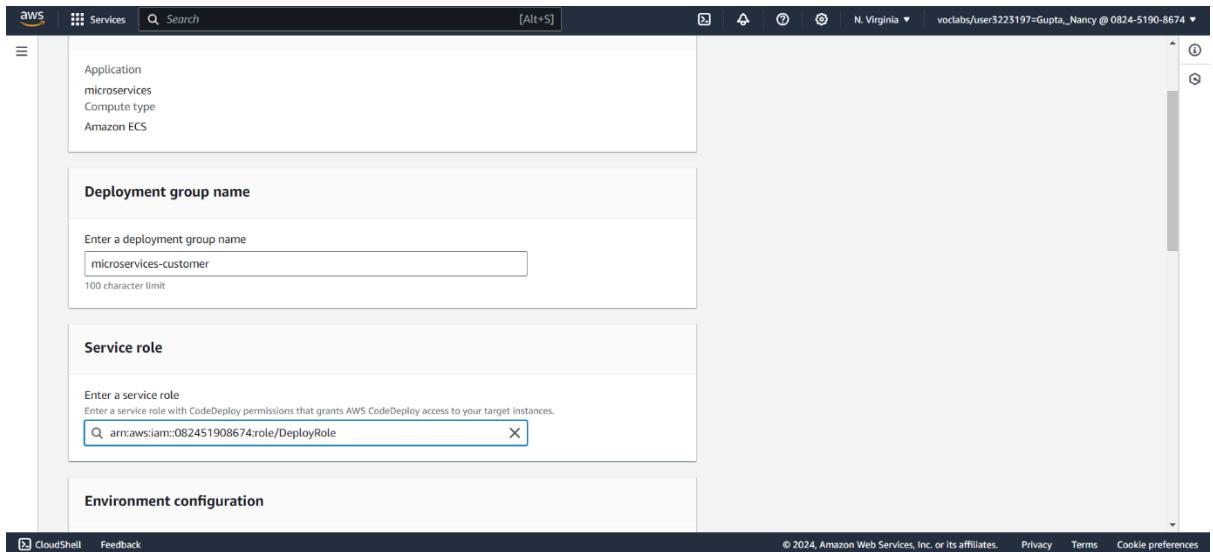
- Choose **Create deployment group** and configure the following:

The screenshot shows the 'Create deployment group' interface in the AWS CodeDeploy console. The 'Deployment group name' field is populated with 'microservices-customer'. The 'Service role' field is empty with a placeholder 'Enter a service role'.

- **Deployment group name:** Enter microservices-customer

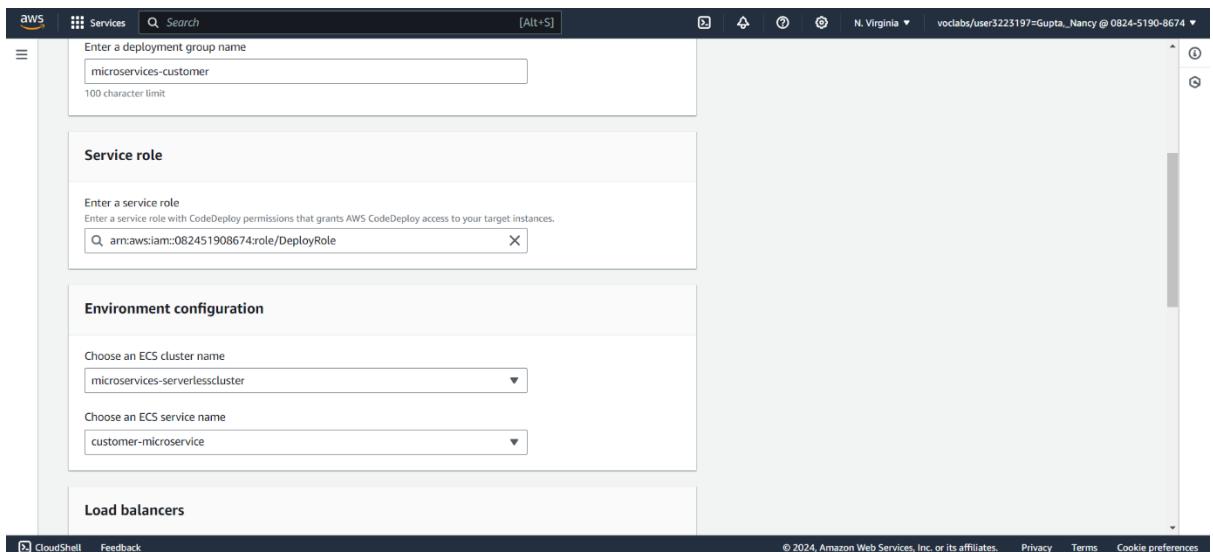
The screenshot shows the 'Create deployment group' interface in the AWS CodeDeploy console. The 'Deployment group name' field is populated with 'microservices-customer'. The 'Service role' field is empty with a placeholder 'Enter a service role'.

- **Service role:** Place your cursor in the search box, and choose the ARN for DeployRole.



- In the **Environment configuration** section:

- **ECS cluster name:** Choose **microservices-serverlesscluster**.
- **ECS service name:** Choose **customer-microservice**.



- In the **Load balancers** section:

- **Load balancer:** Choose **microservicesLB**.
- **Production listener port:** Choose **HTTP:80**.
- **Test listener port:** Choose **HTTP:8080**.
- **Target group 1 name:** Choose **customer-tg-two**.
- **Target group 2 name:** Choose **customer-tg-one**.