Creating EC2 Instances

Within the scope of the project, a virtual machine will be created to install nginx web server. As virtual machine features:

1. Amazon AMI 2 will be used as it is free and resource sufficient.

2. t3.micro instance type will be used. It will provide a sufficient and free configuration for the server.

3. Vpc and security groups created previously in the Network tab will be used. Also Auto-Assign public IP option is selected as enable. This option will automatically assign an IP address to the EC2 server.

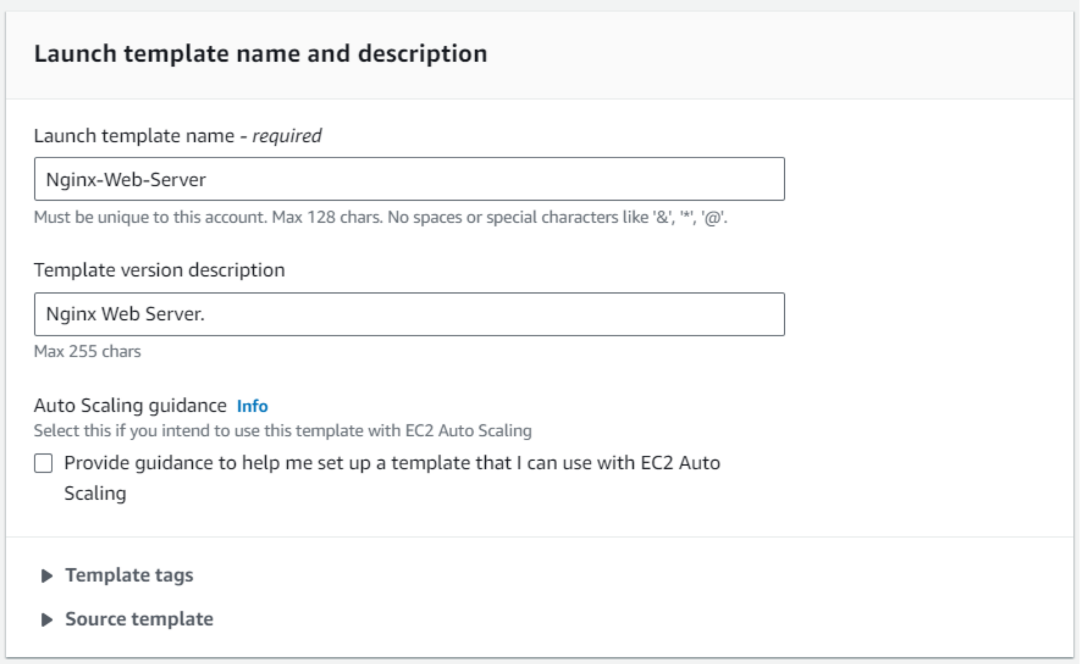
4. In the Advanced Options tab, the following code line will be written in "User Data".

```
#!/bin/bash
yum update -y # Update operating system packages


# Install Nginx web server
yum install nginx -y


yum upgrade -y # Update all operating system packages
yum autoremove -y # Remove unused packages
yum clean all # Clear cache


systemctl enable nginx # Start the Nginx service automatically
systemctl start nginx # Start the Nginx service
```



Launch template name and description

Launch template name - *required*

Nginx-Web-Server

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

Template version description

Nginx Web Server.

Max 255 chars

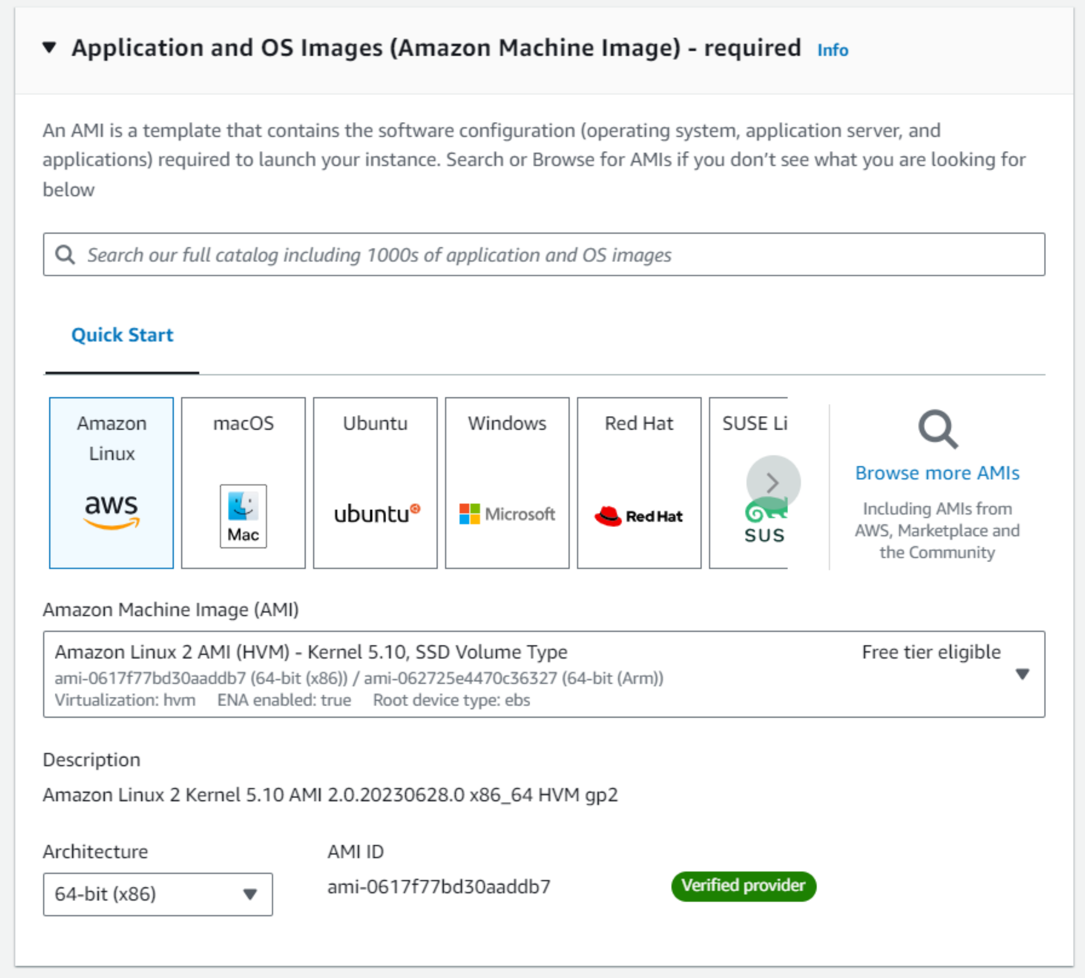
Auto Scaling guidance Info

Select this if you intend to use this template with EC2 Auto Scaling

□ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags

▶ Source template



▼ Application and OS Images (Amazon Machine Image) - required Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible

ami-0617f77bd30aaddb7 (64-bit (x86)) / ami-062725e4470c36327 (64-bit (Arm))

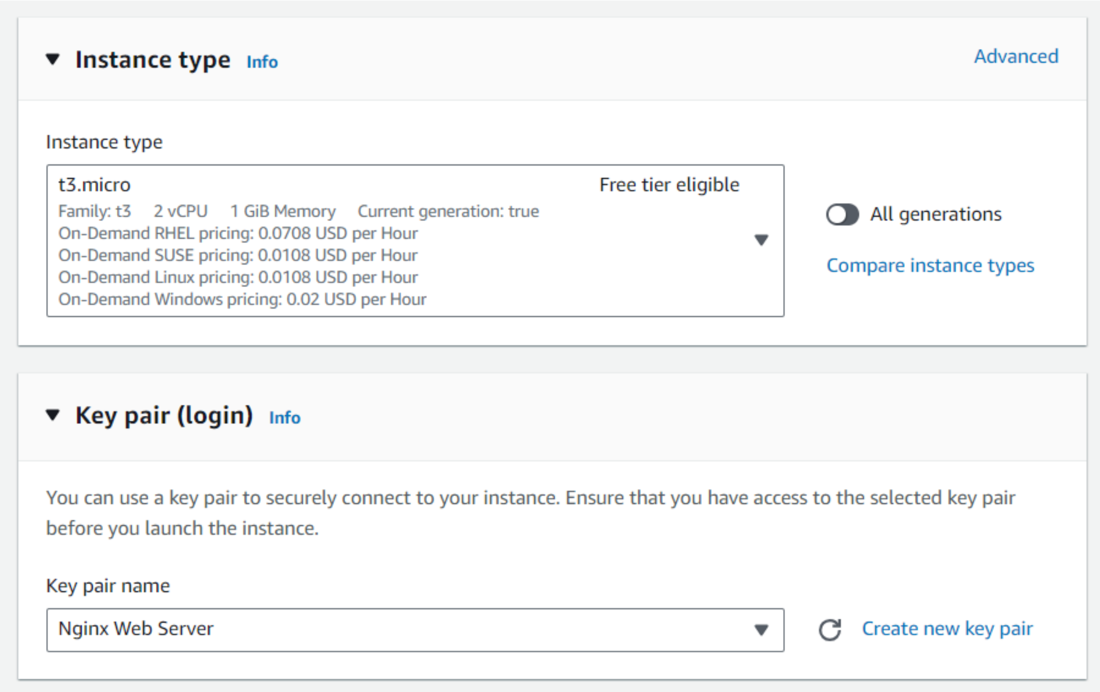
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230628.0 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-0617f77bd30aaddb7 Verified provider



▼ Instance type Info   Advanced

Instance type

t3.micro   Free tier eligible
Family: t3   2 vCPU   1 GiB Memory   Current generation: true
On-Demand RHEL pricing: 0.0708 USD per Hour
On-Demand SUSE pricing: 0.0108 USD per Hour
On-Demand Linux pricing: 0.0108 USD per Hour
On-Demand Windows pricing: 0.02 USD per Hour

All generations

Compare instance types

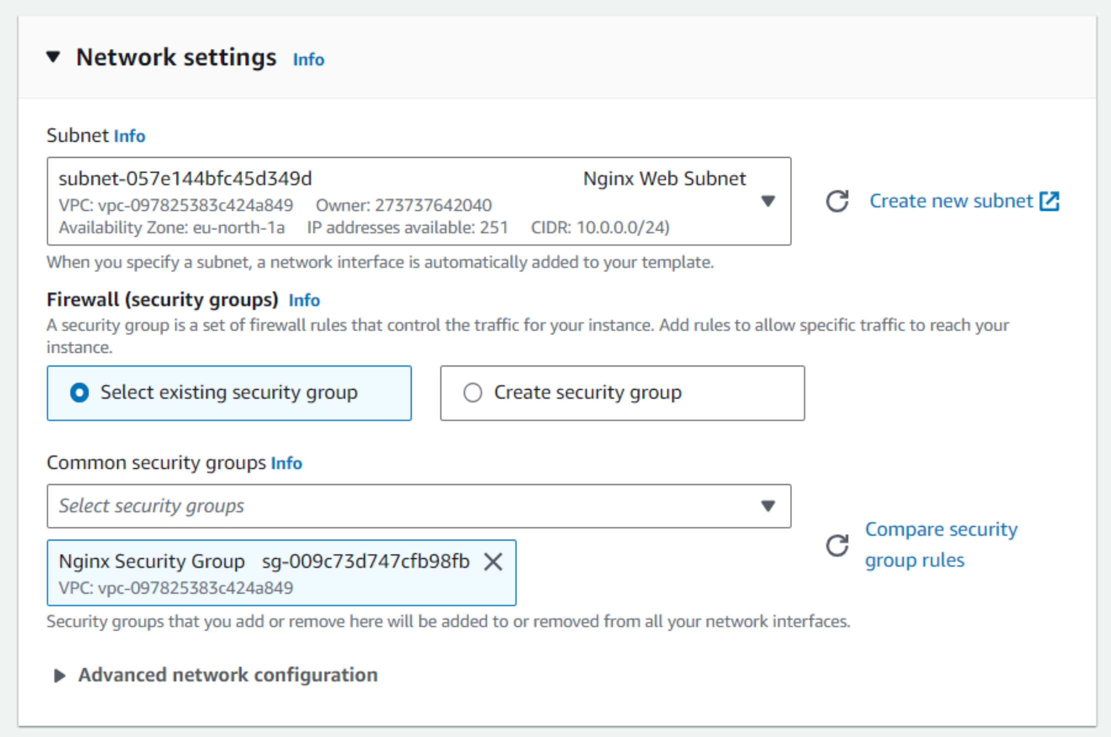
▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

Nginx Web Server ▼

Create new key pair



▼ Network settings Info

Subnet Info

subnet-057e144bfc45d349d   Nginx Web Subnet
VPC: vpc-097825383c424a849   Owner: 273737642040
Availability Zone: eu-north-1a   IP addresses available: 251   CIDR: 10.0.0.0/24)

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Select existing security group   ○ Create security group

Common security groups Info

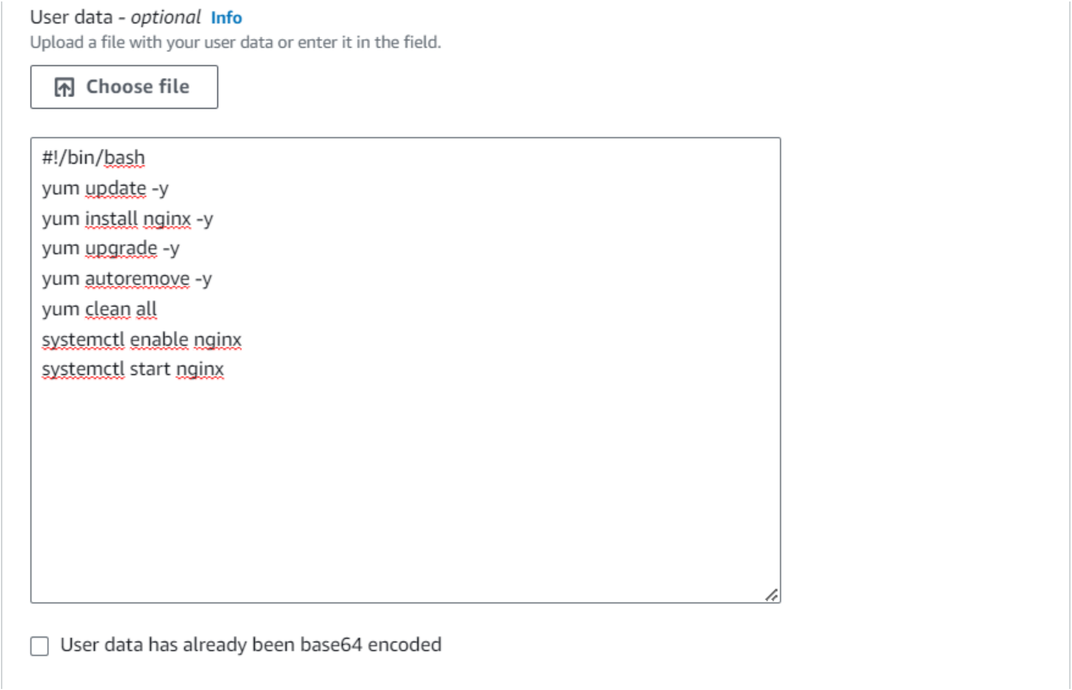
Select security groups ▼

Nginx Security Group   sg-009c73d747cfb98fb ✕
VPC: vpc-097825383c424a849

Compare security group rules

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

▶ Advanced network configuration



User data - *optional* Info

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

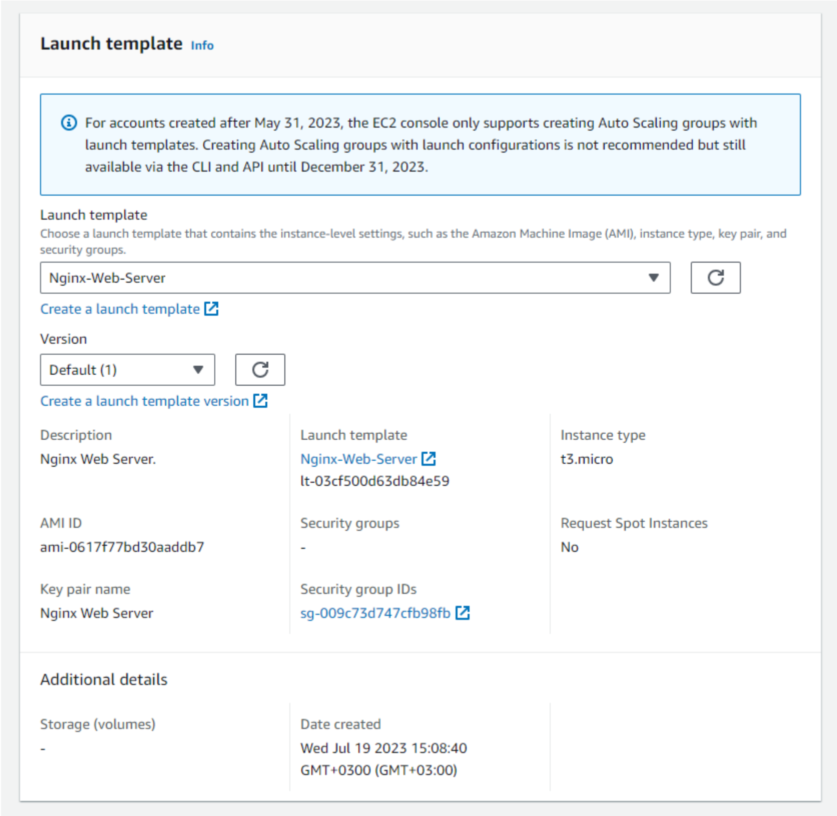
```
#!/bin/bash
yum update -y
yum install nginx -y
yum upgrade -y
yum autoremove -y
yum clean all
systemctl enable nginx
systemctl start nginx
```

□ User data has already been base64 encoded

Creating Auto Scaling Group

Auto scaling means that aws services can be scaled automatically. This means that the memory of EC2 servers can be increased or decreased as needed. Auto scaling can also automatically deploy EC2 instances running in multiple Availability Zones to ensure high availability. This ensures that your application runs without interruption and is always available to users.

In auto scaling, the ec2 launch template that we previously created as launch template will be used. In the Network tab, vpc and subnets created for the project will be used. While creating the network in the project, only 1 subnet was created. To make an application running in more than one availability zone, one more subnet is created here.



Launch template Info

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

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Nginx-Web-Server ▼

Create a launch template

Version

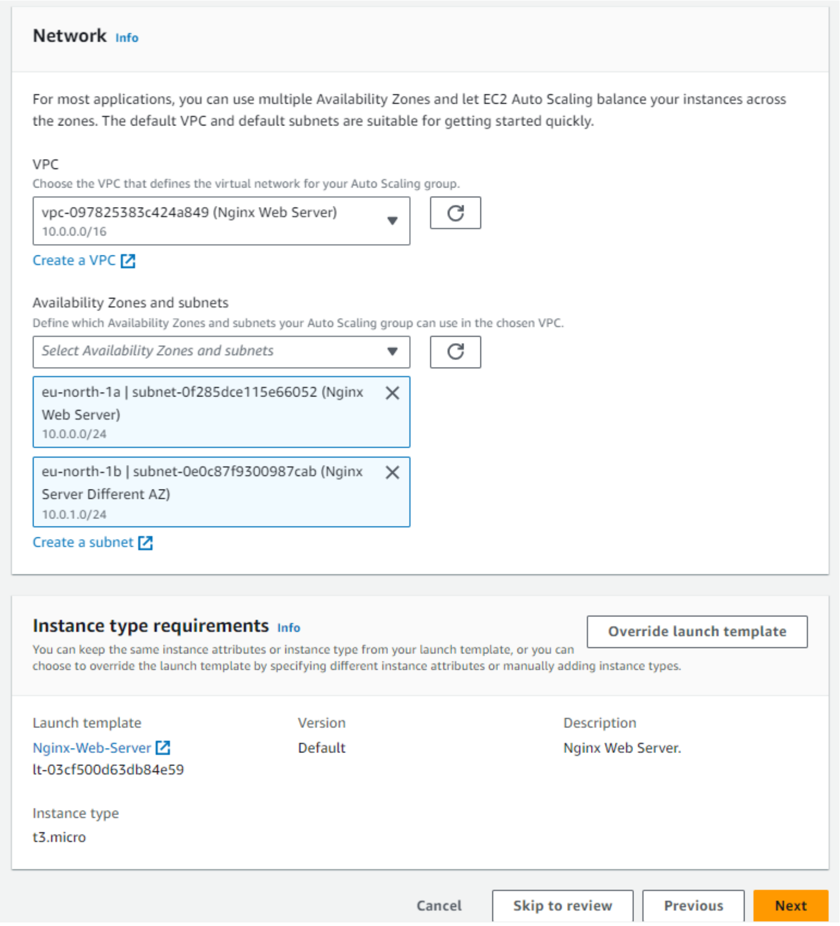
Default (1) ▼

Create a launch template version

Description Nginx Web Server.	Launch template Nginx-Web-Server lt-03cf500d63db84e59	Instance type t3.micro
AMI ID ami-0617f77bd30aaddb7	Security groups -	Request Spot Instances No
Key pair name Nginx Web Server	Security group IDs sg-009c73d747cfb98fb	

Additional details

Storage (volumes) -	Date created Wed Jul 19 2023 15:08:40 GMT+0300 (GMT+03:00)



Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-097825383c424a849 (Nginx Web Server)
10.0.0.0/16 ▼

Create a VPC

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets ▼

eu-north-1a | subnet-0f285dce115e66052 (Nginx Web Server)
10.0.0.0/24 ×

eu-north-1b | subnet-0e0c87f9300987cab (Nginx Server Different AZ)
10.0.1.0/24 ×

Create a subnet

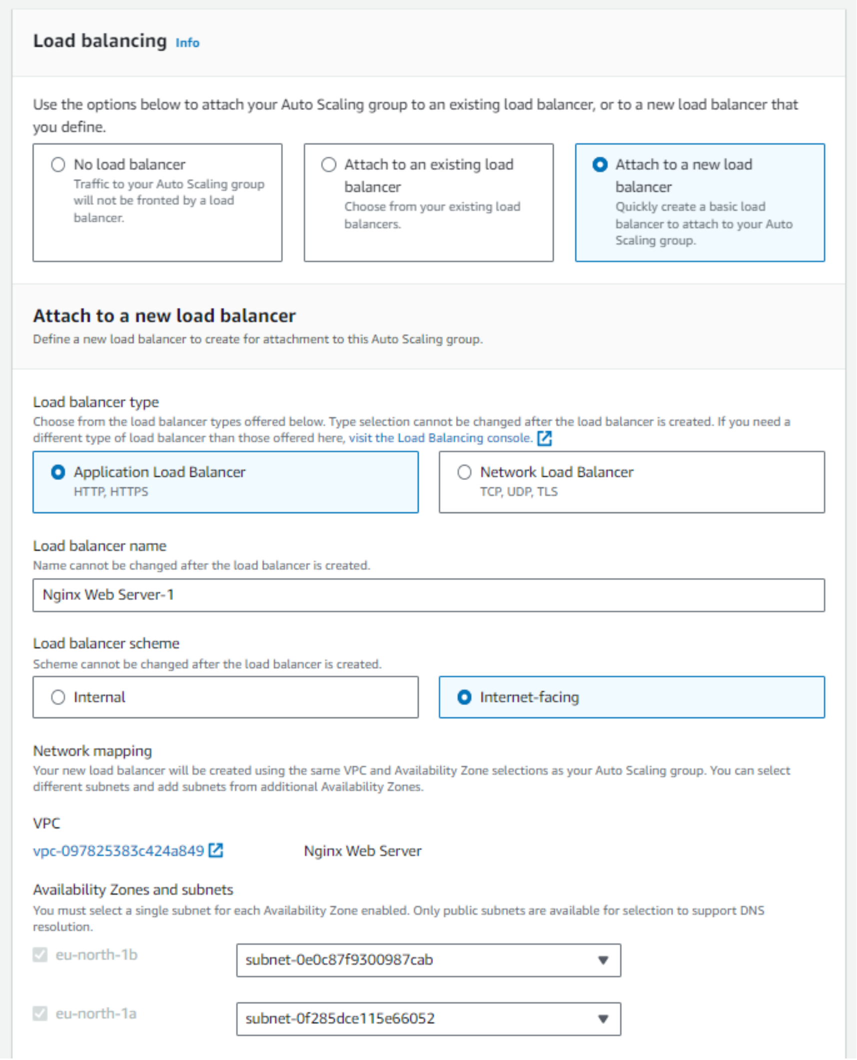
Instance type requirements Info

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template Nginx-Web-Server lt-03cf500d63db84e59	Version Default	Description Nginx Web Server.
Instance type t3.micro		

The load balancer receives incoming requests and distributes these requests to multiple servers or resources according to a specific algorithm or method. In this way, each server and resource are loaded evenly, performance is increased, and load balancing is performed.

In the Load balancer tab, the "Attach to a new Load Balancer" option was selected because we want to add a new load balancer. In the Attach to a new load balancer tab, the "Application Load Balancer" (ALB) option was selected. ALB is a layer 7 (application layer) load balancer used to route HTTP and HTTPS traffic. It has features such as application-level routing, dynamic scaling, multi-protocol support and advanced monitoring. "Internet-facing": This option indicates that your load balancer will be open to the internet. That is, it accepts requests from outside and directs these requests to the correct destinations. It is a preferred option for services open to the internet such as web applications.



Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

○ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

○ Attach to an existing load balancer
Choose from your existing load balancers.

● Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to a new load balancer
Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.

● Application Load Balancer
HTTP, HTTPS

○ Network Load Balancer
TCP, UDP, TLS

Load balancer name
Name cannot be changed after the load balancer is created.
Nginx Web Server-1

Load balancer scheme
Scheme cannot be changed after the load balancer is created.

○ Internal

● Internet-facing

Network mapping
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
vpc-097825383c424a849 Nginx Web Server

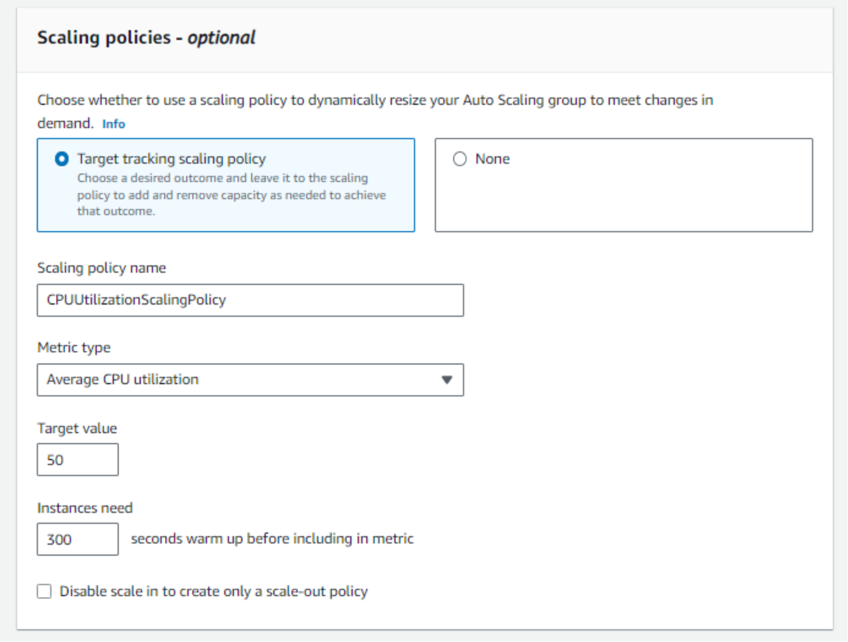
Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☑ eu-north-1b subnet-0e0c87f9300987cab ▼

☑ eu-north-1a subnet-0f285dce115e66052 ▼

Scaling Policies are directives that enable AWS Auto Scaling to automatically perform scaling operations based on a specific rule. These policies allow you to set your scaling goals and ensure that your application scales automatically based on demand.

By selecting the "Target tracking scaling policy" option in the scaling policies tab, it will automatically add and remove capacity to reach a specific target. Since the rule will be created according to CPU usage in the scaling policy name option, the name "CPUUtilisationScalingPolicy" is used as the name (Different name can be used if desired). In the metric type option, it is specified that the rule will be created according to the CPU usage of the EC2 instance by selecting the "Avarage CPU Utilisation" option. Selecting Target Value: 50% and instances need: 300; It is stated that a new instance will be created when the EC2 instance exceeds 50% utilisation for 300 seconds.



Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

● Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

○ None

Scaling policy name
CPUUtilizationScalingPolicy

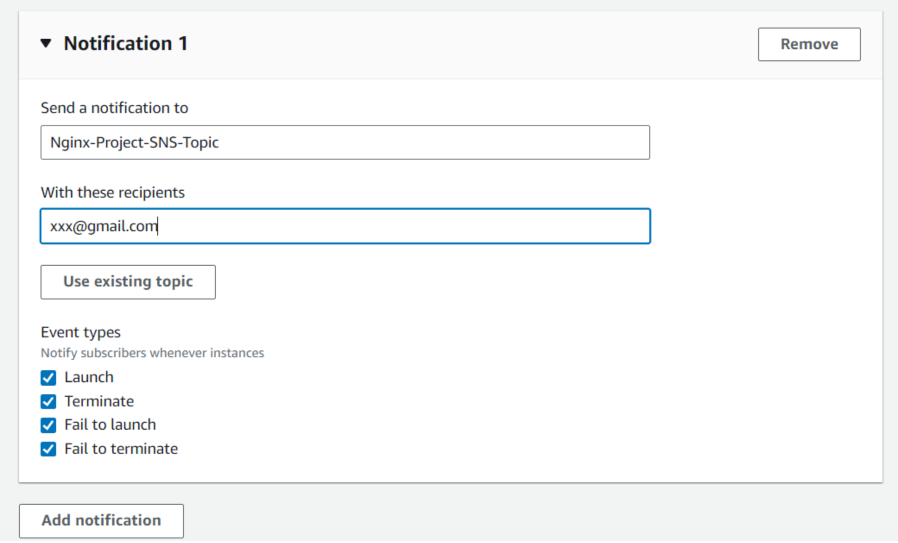
Metric type
Average CPU utilization

Target value
50

Instances need
300 seconds warm up before including in metric

□ Disable scale in to create only a scale-out policy

A new sns has been created in the Add notification tab. AWS SNS (Simple Notification Service) is a service used to provide instant communication and notifications between your applications and systems. SNS allows you to send push notifications to your subscribers (message queues, email, SMS, HTTP endpoints, AWS Lambda functions, etc.) when an event occurs, or a certain condition is met.



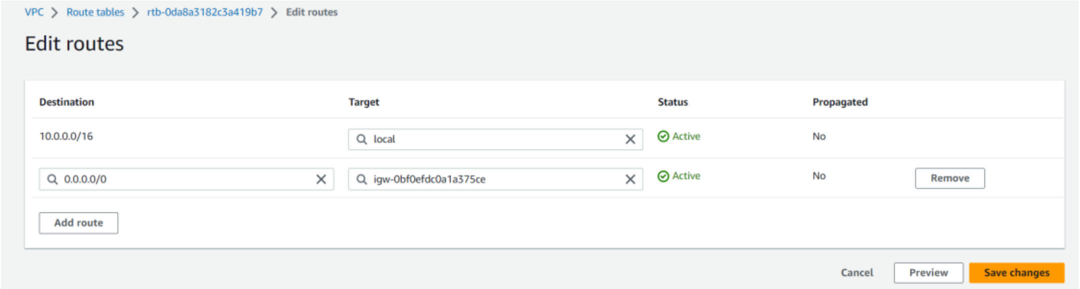
Send a notification to
Nginx-Project-SNS-Topic

With these recipients
xxx@gmail.com

Event types
Notify subscribers whenever instances
☑ Launch
☑ Terminate
☑ Fail to launch
☑ Fail to terminate

TROUBLESHOOTING

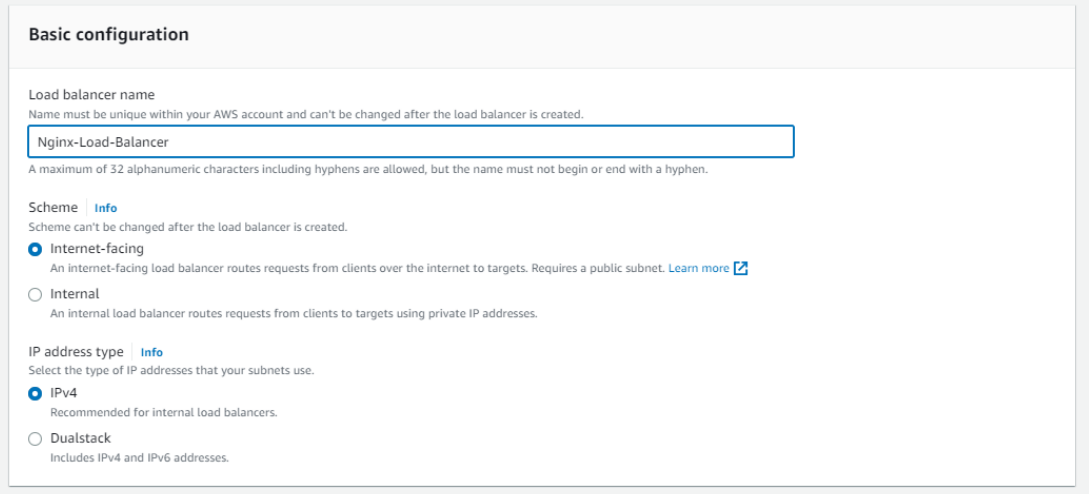
1. SSH connection was not available. I thought that the reason for this was that the public IP address was not assigned, so I first tried to get an Elastic IP address and associate this address for the required instance. I thought it would be fixed when I rebooted the instance, but it was not. For this I ran network track analysis (also known as run reachability analysis). Here it was realized that the problem was in the route table. It seemed that no route was created for the internet gateway in the route table. For this, a new route was created in the route table:
   Destination: 0.0.0.0.0/0 and target: igw-igw_ID. In this configuration, all traffic with the destination 0.0.0.0.0/0 is routed to the Internet Gateway specified as "igw-igw_ID". In this way, private subnets in the VPC can access the internet and, for example, EC2 instances can go to the internet or respond to requests from the internet.
2. The commands written in the User Data field were not executed because the server is not rooted. For this, the commands were executed again after connecting with ssh. The nginx configuration command was rewritten as: ***sudo amazon-linux-extras install nginx1.12 -y***.
3. While creating auto scaling, an error was encountered on the load balancer side, so the load balancer was not created. For this, a new load balancer was created from the load balancer tab and the load balancer was configured by connecting to the required vpc and subnets.



VPC > Route tables > rtb-0da8a3182c3a419b7 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	Active	No	
0.0.0.0/0	igw-0bf0efdc0a1a375ce	Active	No	Remove

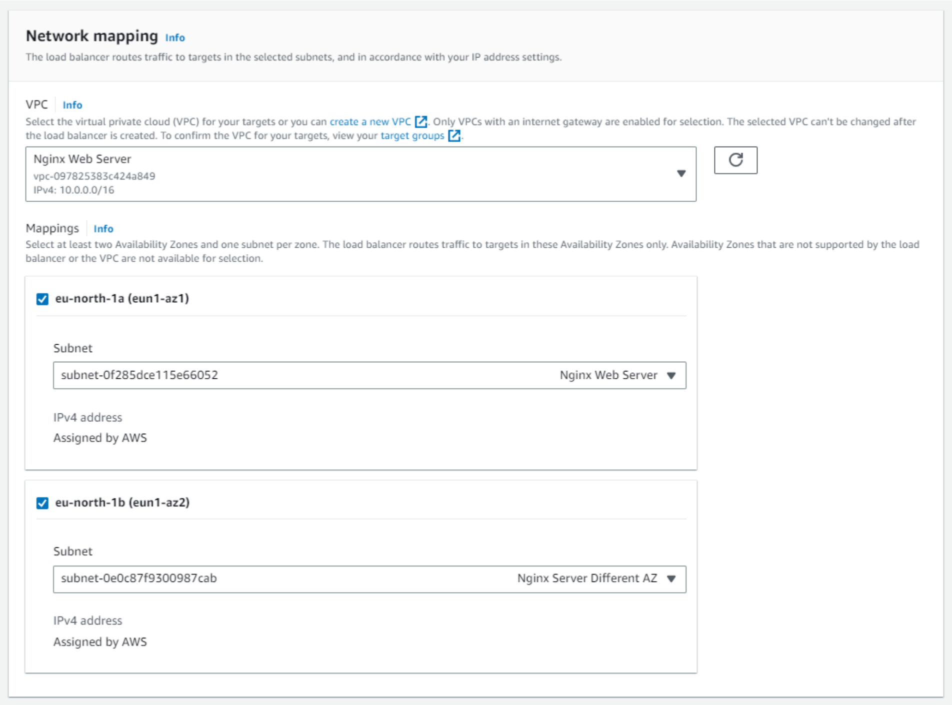


Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
Nginx-Load-Balancer
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.
● Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
○ Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info
Select the type of IP addresses that your subnets use.
● IPv4
Recommended for internal load balancers.
○ Dualstack
Includes IPv4 and IPv6 addresses.



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.

Nginx Web Server
vpc-097825383c424a849
IPv4: 10.0.0.0/16

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.

☑ eu-north-1a (eun1-az1)

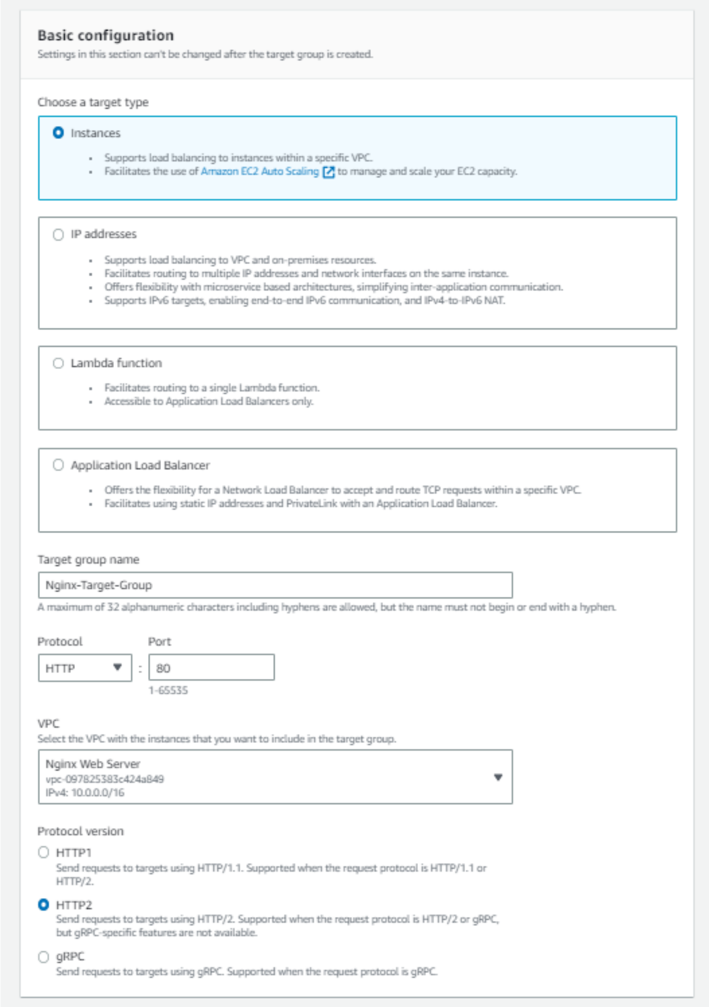
Subnet: subnet-0f285dce115e66052 — Nginx Web Server

IPv4 address: Assigned by AWS

☑ eu-north-1b (eun1-az2)

Subnet: subnet-0e0c87f9300987cab — Nginx Server Different AZ

IPv4 address: Assigned by AWS



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.

○ 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: Nginx-Target-Group

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

Protocol: HTTP : Port: 80

1-65535

VPC

Select the VPC with the instances that you want to include in the target group.

Nginx Web Server
vpc-097825383c424a849
IPv4: 10.0.0.0/16

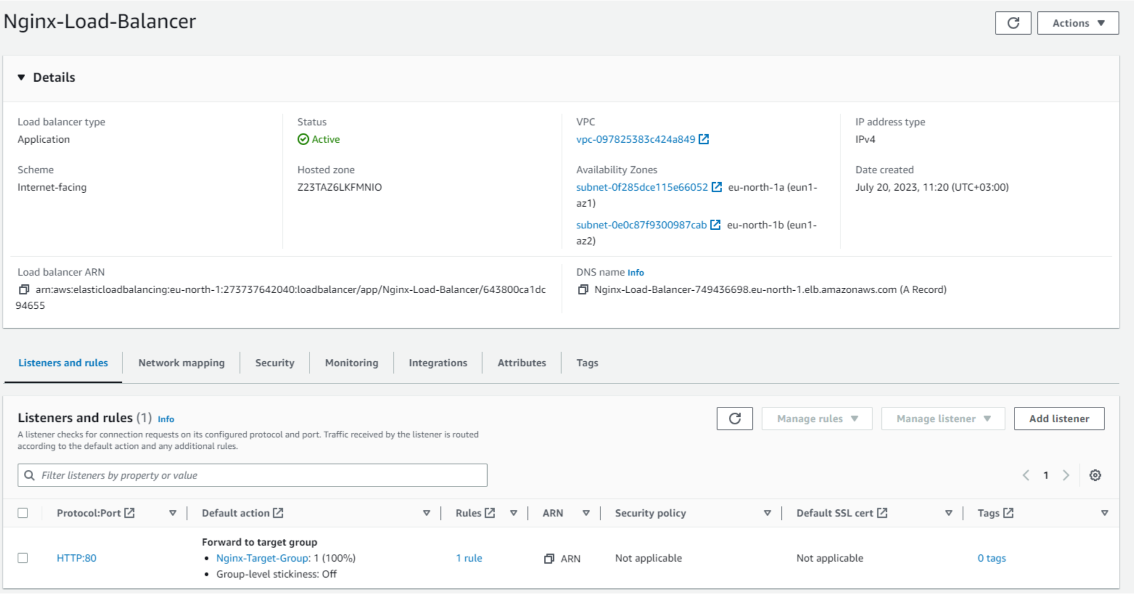
Protocol version

○ HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

● HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

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

Created target group for Elastic load balancer.



Nginx-Load-Balancer

▼ Details

Load balancer type
Application

Status
Active

VPC
vpc-097825383c424a849

IP address type
IPv4

Scheme
Internet-facing

Hosted zone
Z23TAZ6LKFMNIO

Availability Zones
subnet-0f285dce115e66052 eu-north-1a (eun1-az1)
subnet-0e0c87f9300987cab eu-north-1b (eun1-az2)

Date created
July 20, 2023, 11:20 (UTC+03:00)

Load balancer ARN
arn:aws:elasticloadbalancing:eu-north-1:273737642040:loadbalancer/app/Nginx-Load-Balancer/643800ca1dc94655

DNS name Info
Nginx-Load-Balancer-749436698.eu-north-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

□	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL cert	Tags
□	HTTP:80	Forward to target group • Nginx-Target-Group: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	Not applicable	Not applicable	0 tags