



1. Creating VPC

  Services

[VPC](#) > [Your VPCs](#) > Create VPC

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☐ VPC only ☒ VPC and more

Name tag auto-generation [Info](#)
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.

65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block ☐ Amazon-provided IPv6 CIDR block

Tenancy [Info](#)

Default ▼

Number of Availability Zones (AZs) [Info](#)

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

► **Customize AZs**

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 2 4

► **Customize subnets CIDR blocks**

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None In 1 AZ 1 per AZ

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None

S3 Gateway


DNS options [Info](#)

☒ Enable DNS hostnames

☒ Enable DNS resolution

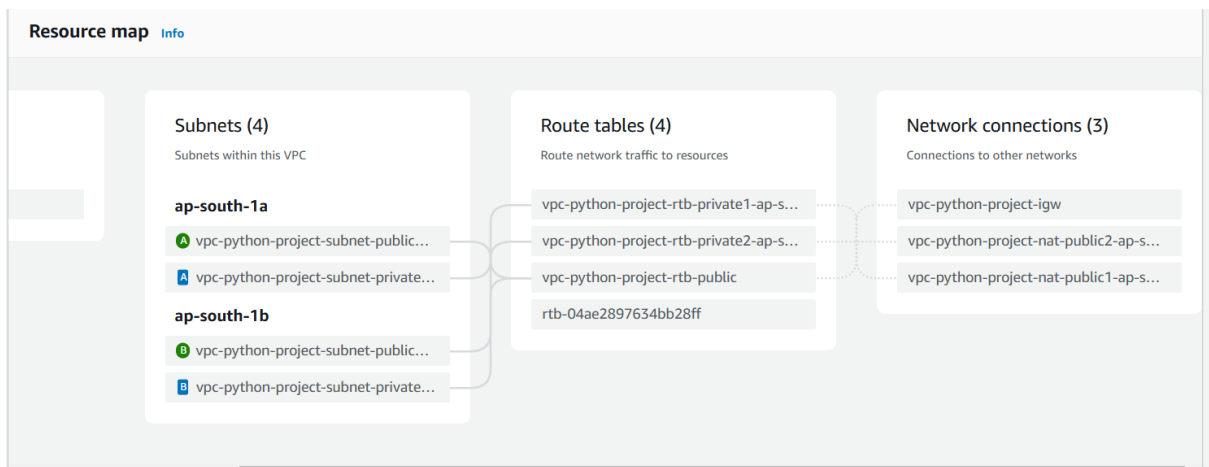
▶ Additional tags

Cancel

 Preview code

Create VPC

- Resource map will be look like below post creation of VPC.



2. Creating launch template for auto scaling group

- You need to create launch template first before creating the auto scaling group.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1

Choose launch template

Step 2

Choose instance launch options

Step 3 - optional

Integrate with other services

Step 4 - optional

Configure group size and scaling

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name
Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

① 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.

Select a launch template

Create a launch template [?](#)

Cancel

Next

- Creating launch template

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description
Launch template name - required

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

Template version description

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

Recents

My AMIs

Quick Start

🕒 Recently launched



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20240801

ami-0522ab6e1ddcc7055

2024-08-01T15:04:23.000Z Architecture: 64-bit (x86) Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

x86_64

AMI ID

ami-0522ab6e1ddcc7055

Verified provider

▼ **Instance type** [Info](#) | [Get advice](#)

[Advanced](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

Free tier eligible

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ **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

git_key

[Create new key pair](#)

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template ▼

↻ [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

Security group name - *required*

vpc-python-project

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#,@[]+=&{}!\$*

Description - *required* [Info](#)

Allows SSH access to developers

VPC [Info](#)

vpc-0540f7d6fdb0f220f (vpc-python-project-vpc)
10.0.0.0/16 ▼

↻

- **Creating inbound security group rule for ssh and application custom port 8000 which we are going to deploy on these instances.**

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

ssh ▼

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Anywhere ▼

Source [Info](#)

🔍 Add CIDR, prefix list or security group

0.0.0.0/0 ✕

Description - *optional* [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 8000, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

Custom TCP ▼

Protocol [Info](#)

TCP

Port range [Info](#)

8000

Source type [Info](#)

Anywhere ▼

Source [Info](#)

🔍 Add CIDR, prefix list or security group

0.0.0.0/0 ✕

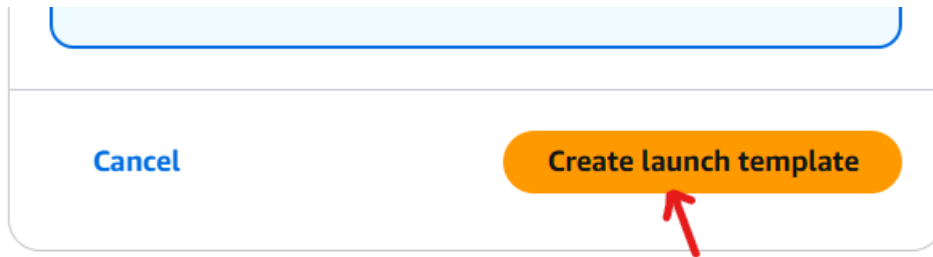
Description - *optional* [Info](#)

e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

[Add security group rule](#)

- Click on create launch template



3. Creating auto scaling group

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1: Choose launch template (selected)
Step 2: Choose instance launch options
Step 3 - optional: Integrate with other services
Step 4 - optional: Configure group size and scaling
Step 5 - optional: Add notifications
Step 6 - optional: Add tags
Step 7: Review

Choose launch template info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name
Auto Scaling group name
Enter a name to identify the group.
vpc-python-project
Must be unique to this account in the current Region and no more than 255 characters.

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.
vpc-python-project
[Create a launch template](#)

Version
Default (1)
[Create a launch template version](#)

- Select your VPC with private subnet since we have to keep our application secure do not want to expose it over the internet.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 5 - optional: Add notifications
Step 6 - optional: Add tags
Step 7: Review

Instance type
t2.micro

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-0540f7d6fbd0f220f (vpc-python-project-vpc)
[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](#)

ap-south-1a subnet-00019a63c6bc6a1b9 (vpc-python-project-subnet-private1-ap-south-1a) 10.0.128.0/20	X
ap-south-1b subnet-0eb310f70876a82e7 (vpc-python-project-subnet-private2-ap-south-1b) 10.0.144.0/20	X

[Create a subnet](#)

- Step 1 Choose launch template
- Step 2 Choose instance launch options
- Step 3 - optional
- Step 3 - optional Integrate with other services**
- Step 4 - optional
- Step 4 - optional Configure group size and scaling
- Step 5 - optional
- Step 5 - optional Add notifications
- Step 6 - optional
- Step 6 - optional Add tags
- Step 7
- Step 7 Review

Integrate with other services - optional [Info](#)

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

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.

VPC Lattice integration options [Info](#)

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

- ☒ No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.
- ☐ Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#)

- Step 4 - optional
- Step 4 - optional Configure group size and scaling**
- Step 5 - optional
- Step 5 - optional Add notifications
- Step 6 - optional
- Step 6 - optional Add tags
- Step 7
- Step 7 Review

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

2

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity 1
Equal or less than desired capacity

Max desired capacity 4
Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

- ☒ No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
- ☐ Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

- Step 1 Choose launch template
- Step 2 Choose instance launch options
- Step 3 - optional
- Step 3 - optional Integrate with other services
- Step 4 - optional
- Step 4 - optional Configure group size and scaling
- Step 5 - optional
- Step 5 - optional Add notifications**
- Step 6 - optional
- Step 6 - optional Add tags
- Step 7
- Step 7 Review

Add notifications - optional [Info](#)

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

[Add notification](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1: Choose launch template
 Step 2: Choose instance launch options
 Step 3 - optional: Integrate with other services
 Step 4 - optional: Configure group size and scaling
 Step 5 - optional: Add notifications
 Step 6 - optional: **Add tags**
 Step 7: Review

Add tags - optional Info
 Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

1 You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

Tags (0)
 Add tag
 50 remaining

Cancel Previous **Next**

EC2 > Auto Scaling groups > Create Auto Scaling group

Disabled Disabled Disabled

Capacity Reservation preference
 Preference: Default
 Capacity Reservation IDs: -
 Resource Groups: -

Step 5: Add notifications Edit
Notifications
 No notifications

Step 6: Add tags Edit
Tags (0)

Key	Value	Tag new instances
No tags		

Preview code Cancel Previous **Create Auto Scaling group**

Auto Scaling groups (1) Info Refresh Launch configurations Launch templates Actions **Create Auto Scaling group**

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
<input type="checkbox"/>	vpc-python-project	vpc-python-project Version Default	2	-	2	1	4

➤ Now auto scaling group will create the instances as per desired capacity mentioned with private ips.

EC2 > Auto Scaling groups

Instances (2) Info Last updated less than a minute ago Connect Instance state Actions **Launch instances**

All states

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>		i-01370156e420828a5	Running	t2.micro	Initializing	View alarms +	ap-south-1a	-
<input type="checkbox"/>		i-07643dbd6ca16a71c	Running	t2.micro	Initializing	View alarms +	ap-south-1b	-

4. Creating Bastion or jump host to access these instances since it do not have public ips. Bastion host is act as a mediator between private subnet and public subnet.

☰ [EC2](#) > [Instances](#) > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [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

Recents

My AMIs

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu®

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

...

>

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

▼ **Storage**

Number of volumes: 1

Software: Canonical Ubuntu Server 20.04 LTS (Focal Fossa) ami-05...

Virtualization: t2.micro

Firewall: New security group

Storage: 1 volume

Car

➤ Selecting our VPC with public subnet.

▼ Network settings [Info](#)

VPC - required [Info](#)

10.0.0.0/16

Subnet [Info](#)

vpc-python-project-subnet-public2-ap-south-1b
VPC: vpc-0540f7d6fbd0f220f Owner: 326156133222
Availability Zone: ap-south-1b Zone type: Availability Zone
IP addresses available: 4089 CIDR: 10.0.16.0/20

Auto-assign public IP [Info](#)

[Additional charges apply](#) when outside of [free tier allowance](#)

EC2 > Instances > Launch an instance

Security group name - *required*

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./!@#%&'()*+<=>:;[]{}*~

Description - *required* [Info](#)

Inbound Security Group Rules
 ▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range	Source type	Source	Description - optional
ssh	TCP	22	Anywhere	<input type="text" value="0.0.0.0/0"/>	e.g. SSH for admin desktop

[Add security group rule](#)

Summary

Number of instances [Info](#)

Software Image (AMI)
 Canonical, Ubuntu, 24.04, amd64...[read more](#)
 ami-053b12d5152c0cc71

Virtual server type (instance type)
 t2.micro

Firewall (security group)
 New security group

Storage (volumes)
 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage per free tier

[Cancel](#) [Launch instance](#)

- Copying pem key from local machine to bastion host to access application instance from bastion host.

```
C:\Users\swapnil\Downloads>scp -i git_key.pem git_key.pem ubuntu@65.2.57.9:/home/ubuntu/
```

- Successfully logged in to one of the application instance from bastion host.

```
ubuntu@ip-10-0-18-164:~$ ssh -i git_key.pem ubuntu@10.0.143.109
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
```

- Running python server with sample index.html file on application instance.

```
ubuntu@ip-10-0-143-109:~$ vim index.html
ubuntu@ip-10-0-143-109:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

5. Creating Application load balancer to incoming requests to your application

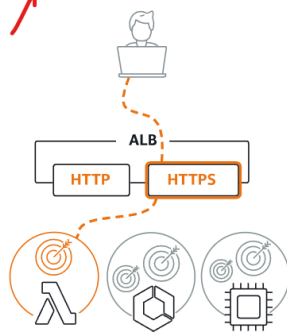
[EC2](#) > [Load balancers](#) > Compare and select load balancer type

Compare and select load balancer type

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

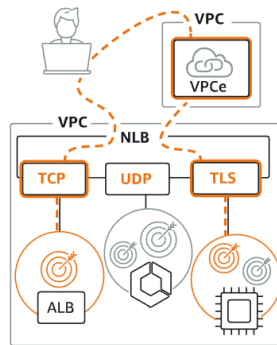
Load balancer types

Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your

Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading

Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-

➤ LB should be internet facing.

[EC2](#) > [Load balancers](#) > Create Application Load Balancer

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule as

► How Application Load Balancers work

Basic configuration

Load balancer name

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

vpc-python-project

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

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name is publicly resolvable.
- Requires a public subnet.

☐ Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name is not publicly resolvable.
- Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type [Info](#)

Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

☒ IPv4

- Includes only IPv4 addresses.

☐ Dualstack

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises target VPC for your targets, view [target groups](#). For a new VPC, [create a VPC](#).

vpc-python-project-vpc
vpc-0540f7d6fdb0f220f
IPv4 VPC CIDR: 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 available for selection.

Availability Zones

☒ ap-south-1a (aps1-az1)

Subnet

subnet-06d17766cffb060c0
IPv4 subnet CIDR: 10.0.0.0/20

vpc-python-project-subnet-public1-ap-south-1a

IPv4 address

Assigned by AWS

☒ ap-south-1b (aps1-az3)

Subnet

subnet-0b58999244a2a738e
IPv4 subnet CIDR: 10.0.16.0/20

vpc-python-project-subnet-public2-ap-south-1b

IPv4 address

Assigned by AWS

➤ Creating target group

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](#).

Security groups

Select up to 5 security groups

vpc-python-project
sg-05cc5c5a13b6d6b67 VPC: vpc-0540f7d6fdb0f220f

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 route registered targets.

▼ Listener HTTP:80

Protocol

HTTP

Port

80

1-65535

Default action [Info](#)

Forward to

Select a target group

[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](#)

➤ Selecting port 8000 since our application is running on same port.

EC2 > Target groups > Create target group

- 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

vpc-python-project

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 8000

1-65535

IP address type

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

☒ IPv4

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

☐ IPv6

Each instance you register must have an assigned address IPv6 address. This is configured on the instance's

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.

Available instances (2/3)

Filter instances

	Instance ID	Name	State	Security groups	Zone
<input type="checkbox"/>	i-0252e0941ccb72932	bastion-host	Running	launch-wizard-11	ap-south-1b
<input checked="" type="checkbox"/>	i-01370156e420828a5		Running	vpc-python-project	ap-south-1a
<input checked="" type="checkbox"/>	i-07643dbd6ca16a71c		Running	vpc-python-project	ap-south-1b

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

8000

1-65535 (separate multiple ports with commas)

Include as pending below

vpc-python-project Actions

Details

arn:aws:elasticloadbalancing:ap-south-1:326156133222:targetgroup/vpc-python-project/d74b8f39d98d8ab4

Target type Instance	Protocol : Port HTTP: 8000	Protocol version HTTP1	VPC vpc-0540f7d6fbd0f220f
IP address type IPv4	Load balancer None associated		

2 Total targets	0 Healthy	0 Unhealthy	2 Unused	0 Initial	0 Draining
0 Anomalous					

► **Distribution of targets by Availability Zone (AZ)**

Select values in this table to see corresponding filters applied to the Registered targets table below.

➤ Back to the ALB configuration.

EC2 > Load balancers > Create Application Load Balancer

registered targets.

▼ Listener HTTP:80 Remove

Protocol: HTTP Port: 80

Default action: Forward to vpc-python-project (Target type: Instance, IPv4) HTTP ⓘ

[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.

Service integrations [Edit](#)

Amazon CloudFront + AWS Web Application Firewall (WAF): None
AWS WAF: None
AWS Global Accelerator: None

Tags [Edit](#)

None

Attributes

ⓘ Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

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.

[Cancel](#) [Create load balancer](#)

➤ Try to access your page by using DNS name present in LB.

EC2 > Load balancers > vpc-python-project

AMIs
AMI Catalog

▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

▼ Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

▼ Load Balancing

vpc-python-project

[ⓘ](#) [Actions](#)

▼ Details

Load balancer type Application	Status ⏸ Provisioning	VPC vpc-0540f7d6fdb0f220f	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones subnet-0b58999244a2a738e ap-south-1b (aps1-az3) subnet-06d17766c9fb060c0 ap-south-1a (aps1-az1)	Date created December 3, 2024, 12:48 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:326156133222:loadbalancer/app/vpc-python-project/6a0fb37e9e87352d		DNS name Info vpc-python-project-1482817177.ap-south-1.elb.amazonaws.com (A Record)	

← → ↻ ⚠ Not secure vpc-python-project-1482817177.ap-south-1.elb.amazonaws.com

This is a first VPC python project

Click me