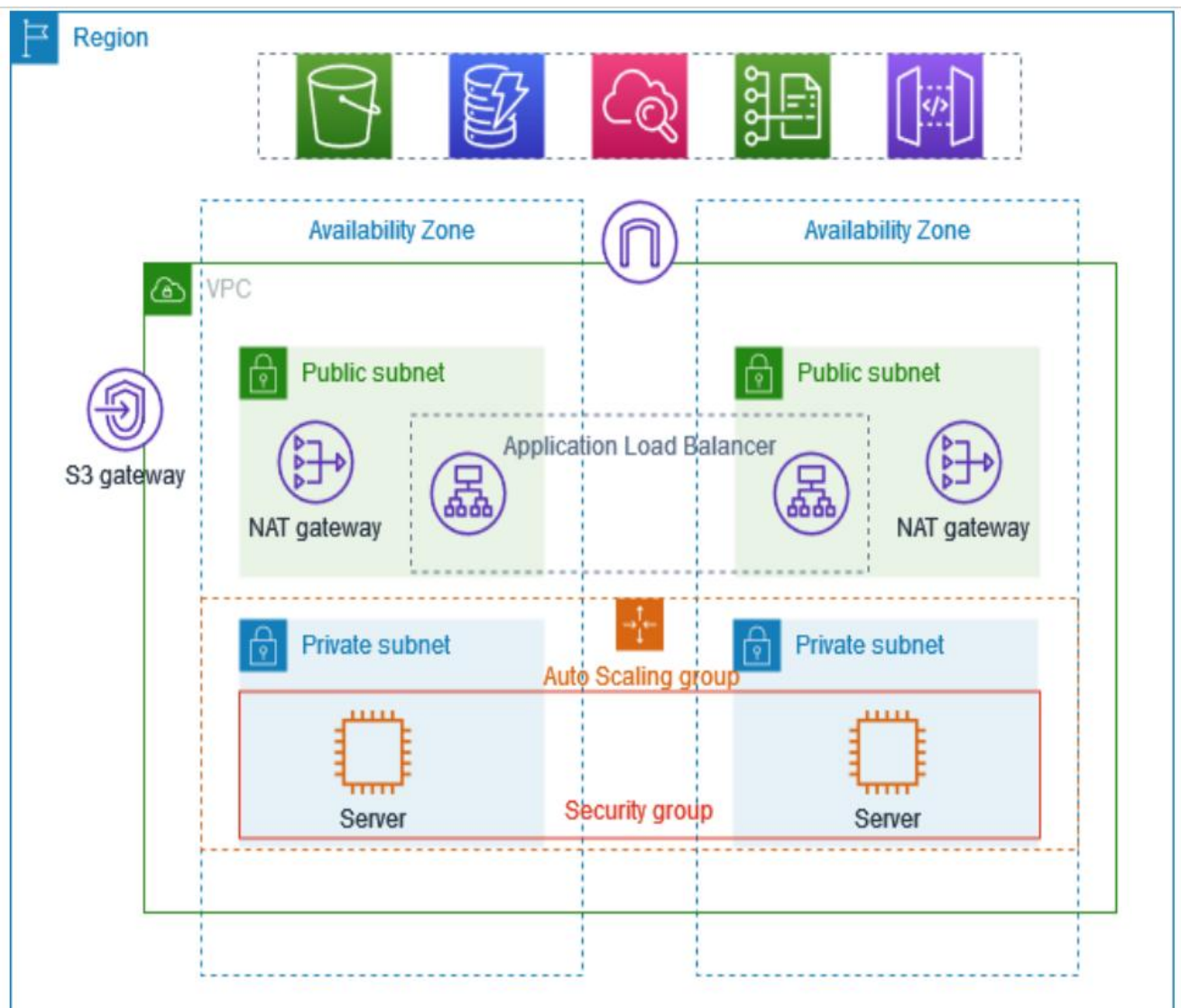


VPC WITH PUBLIC PRIVATE SUBNET IN PRODUCTION

This example demonstrates how to create a VPC that you can use for servers in a production environment. To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both Availability Zones.

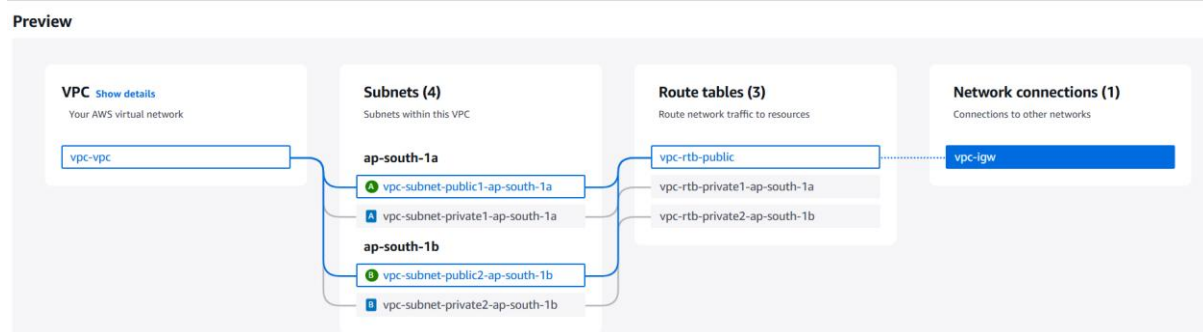
The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway.



>Goto vpc > create vpc

>Create vpc and more

The screenshot shows the AWS VPC dashboard. On the left is a sidebar with navigation links: VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Your VPCs, Subnets, Route tables, and Internet gateways. The main content area has buttons for 'Create VPC' and 'Launch EC2 Instances'. Below these is a 'Resources by Region' section with a 'Refresh Resources' link. It displays four resource categories: VPCs (1 in Asia Pacific), NAT Gateways (0 in Asia Pacific), Subnets (3 in Asia Pacific), and VPC Peering Connections (0 in Asia Pacific). Each category has a link to 'See all regions'.



>name > number of AZ >2

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](#)

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

>Number of public subnets > 2 > Number of private subnets > 2

> Number of NAT Gateway > 1 per AZ

► 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

>Create

Create VPC workflow

◀ Wait for NAT Gateways to activate

69%

▼ Details

- ✓ Create VPC: [vpc-09ed45b735e34ac4f](#)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: [vpc-09ed45b735e34ac4f](#)
- ✓ Create subnet: [subnet-0cbd33f38bb862a36](#)
- ✓ Create subnet: [subnet-083cd7581977a391a](#)
- ✓ Create subnet: [subnet-09c92b358e8ca230d](#)
- ✓ Create subnet: [subnet-02250245a5b7375da](#)
- ✓ Create internet gateway: [igw-0fdd67a36eddf320](#)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: [rtb-0a3e491d13b0eb75c](#)
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Allocate elastic IP: [eipalloc-05b6f71d617121035](#)
- ✓ Allocate elastic IP: [eipalloc-0230665970540f094](#)
- ✓ Create NAT gateway: [nat-046ccd5abeba391b0](#)
- ✓ Create NAT gateway: [nat-0b5762b59e25ee6be](#)
- ⌛ Wait for NAT Gateways to activate
- ⌚ Create route table

You can see all components are create automatically.

>Create 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*

temp

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

Template version description

for production

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

>select ubuntu image

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-00bb6a80f01f03502 (64-bit (x86)) / ami-09773b29dffbef1f2 (64-bit (Arm))

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

Free tier eligible

>Create new security group in production vpc

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*

new-sg

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](#)

new-sg

VPC | [Info](#)

vpc-09ed45b735e34ac4f (aws-production-vpc)

10.0.0.0/16

>in security group allow SSH and allow 8000 port

Inbound Security Group Rules

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

Type Info	Protocol Info	Port range Info
ssh	TCP	22
Source type Info	Source Info	Description - <i>optional</i> Info
Anywhere	<input type="text" value="Add CIDR, prefix list or security group"/> <input type="text" value="0.0.0.0/0"/>	e.g. SSH for admin desktop

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

Type Info	Protocol Info	Port range Info
Custom TCP	TCP	8000
Source type Info	Source Info	Description - <i>optional</i> Info
Anywhere	<input type="text" value="Add CIDR, prefix list or security group"/> <input type="text" value="0.0.0.0/0"/>	e.g. SSH for admin desktop

>done

>Create Auto-scaling

EC2 > Auto Scaling groups > Create Auto Scaling group Info Help

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](#)

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.

[Create a launch template](#)

>name > select temp

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.

[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-09c92b358e8ca230d (aws-production-subnet-private1-ap-south-1a) ×

10.0.128.0/20

ap-south-1b | subnet-02250245a5b7375da (aws-production-subnet-private2-ap-south-1b) ×

10.0.144.0/20

>select production vpc >select 2 private subnet

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

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

>desired capacity=2 >min desired capacity=1 >max desired capacity=4

Auto Scaling groups (1) [Info](#)

Search your Auto Scaling groups



Launch configurations

Launch templates [↗](#)

Actions ▼

Create Auto Scaling group

< 1 > ⚙

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	production-auto-scale	temp Version Default	2	-	2	1	4	ap-south-1b, ap-south...

You can see auto scaling group are create

Instances (1/2) [Info](#)

Last updated
1 minute ago

Connect

Instance state ▼

Actions ▼

Launch instances ▼

Find Instance by attribute or tag (case-sensitive)

All states ▼

< 1 > ⚙

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Pul
<input checked="" type="checkbox"/>		i-064c3fc2b03...	Running	t2.micro	2/2 checks p	View alarms +	ap-south-1b	-	-
<input type="checkbox"/>		i-0db3427528e...	Running	t2.micro	2/2 checks p	View alarms +	ap-south-1a	-	-

You can see 2 instance are deploy automatically.

Name and tags [Info](#)

Name

vm1

Add additional tags

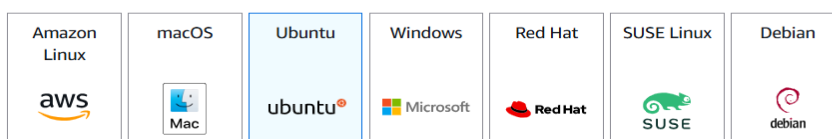
▼ Application and OS Images (Amazon Machine Image) [Info](#)

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

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

Recents

Quick Start



Browse more AMIs

Including AMIs from
AWS, Marketplace and
the Community

>Also create 1 instance for jump server

▼ 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 - *required*

mew

↕

Create new key pair

▼ Network settings Info

VPC - *required* Info

vpc-09ed45b735e34ac4f (aws-production-vpc)

↕

Subnet Info

subnet-083cd7581977a391a aws-production-subnet-public2-ap-south-1b

VPC: vpc-09ed45b735e34ac4f Owner: 533267265588 Availability Zone: ap-south-1b

Zone type: Availability Zone IP addresses available: 4090 CIDR: 10.0.16.0/20

↕ Create new subnet ↗

Auto-assign public IP Info

Enable

↕

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

>With same production Vpc with public subnet and enable public Ip

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.

Create security group

Select existing security group

Security group name - *required*

launch-wizard-1

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

launch-wizard-1 created 2025-02-21T12:13:20.578Z

Inbound Security Group Rules

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

Type Info

Protocol Info

Port range Info

Source type Info

Source Info

Description - *optional* Info

ssh

TCP

22

Anywhere

Add CIDR, prefix list or security group

e.g. SSH for admin desktop

Remove

>create security group with SSH rule

Instances (3/3) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

✓	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
✓		i-064c3fc2b03...	Running	t2.micro	2/2 checks pass	View alarms	ap-south-1b	-	-
✓	vm1	i-0c930c13105...	Running	t2.micro	Initializing	View alarms	ap-south-1b	ec2-3-109-3-174.a...	3.1
✓		i-0db3427528e...	Running	t2.micro	2/2 checks pass	View alarms	ap-south-1a	-	-

you can see new instances are created


```
>scp -i /users/downloads\abc.pem c:\users\downloads\abc.pem ubuntu@1.24.23.5:/home/ubuntu
```

```
C:\Users\kharv\Documents>scp -i /Users\kharv\Downloads\pem C:\Users\kharv\Downloads\pem ubuntu@3.109.3.174
1 file(s) copied.
```

Copy in ubuntu machine

```
C:\Users\kharv\Downloads>ssh -i pem ubuntu@3.109.3.174
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)
```

```
>ssh -i abc.pem ubuntu@1.24.23.5
```

```
ubuntu@ip-10-0-29-140:~$ ssh -i pem ubuntu@10.0.152.146
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)
```

you can see you access the machine

```
ubuntu@ip-10-0-152-146:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
```

```
>sudo apt install apache2
```

```
ubuntu@ip-10-0-139-110:/var/www/html$ cd /var/www/html
ubuntu@ip-10-0-139-110:/var/www/html$ sudo rm -rf *
ubuntu@ip-10-0-139-110:/var/www/html$ ls
ubuntu@ip-10-0-139-110:/var/www/html$ sudo nano index.html
ubuntu@ip-10-0-139-110:/var/www/html$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.25.174 - - [21/Feb/2025 13:53:10] "GET / HTTP/1.1" 200 -
10.0.25.174 - - [21/Feb/2025 13:53:12] "GET / HTTP/1.1" 304 -
10.0.25.174 - - [21/Feb/2025 13:53:12] "GET / HTTP/1.1" 304 -
10.0.25.174 - - [21/Feb/2025 13:53:13] "GET / HTTP/1.1" 304 -
10.0.25.174 - - [21/Feb/2025 13:53:14] "GET / HTTP/1.1" 304 -
10.0.25.174 - - [21/Feb/2025 13:53:14] "GET / HTTP/1.1" 304 -
10.0.25.174 - - [21/Feb/2025 13:53:16] "GET / HTTP/1.1" 304 -
```

```
>cd /var/www/html/
```

```
>sudo rm -rf *
```

```
>sudo nano index.html link - krishna-20802/-VPC-with-Public-Private-Subnet-in-Production
```

```
>python3 -m http.server 8000
```

```
>get ssh in another instance
```

```
ubuntu@ip-10-0-29-140:~$ ssh -i pem ubuntu@10.0.152.146
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)
```

```
>cd /var/www/html/
```



```
>sudo rm -rf *
```

```
>sudo nano index.html
```

```
>python3 -m http.server 8000
```

```
ubuntu@ip-10-0-152-146:/var/www/html$ sudo rm -rf *
ubuntu@ip-10-0-152-146:/var/www/html$ ls
ubuntu@ip-10-0-152-146:/var/www/html$ sudo nano index.html
ubuntu@ip-10-0-152-146:/var/www/html$ ubuntu@ip-10-0-152-146:/var/www/html$
ubuntu@ip-10-0-152-146:/var/www/html$ ls
index.html
ubuntu@ip-10-0-152-146:/var/www/html$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.25.174 - - [21/Feb/2025 13:51:47] "GET / HTTP/1.1" 200 -
10.0.3.180 - - [21/Feb/2025 13:51:57] "GET / HTTP/1.1" 200 -
10.0.25.174 - - [21/Feb/2025 13:52:17] "GET / HTTP/1.1" 200 -
10.0.3.180 - - [21/Feb/2025 13:52:27] "GET / HTTP/1.1" 200 -
10.0.25.174 - - [21/Feb/2025 13:52:47] "GET / HTTP/1.1" 200 -
10.0.3.180 - - [21/Feb/2025 13:52:57] "GET / HTTP/1.1" 200 -
```

>Create Application Load Balancer with internet facing

Basic configuration

Load balancer name

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

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 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

Includes IPv4 and IPv6 addresses.

☐ Dualstack without public IPv4

Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

>select production vpc and public subnet

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 targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#) [↗](#). For a new VPC, [create a VPC](#) [↗](#).

aws-production-vpc
vpc-09ed45b735e34ac4f
IPv4 VPC CIDR: 10.0.0.0/16



Availability Zones and subnets [Info](#)

Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

☒ ap-south-1a (aps1-az1)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.

subnet-0cbd33f38bb862a36
IPv4 subnet CIDR: 10.0.0.0/20

aws-production-subnet-public1-ap-south-1a

☒ ap-south-1b (aps1-az3)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.

subnet-083cd7581977a391a
IPv4 subnet CIDR: 10.0.16.0/20

aws-production-subnet-public2-ap-south-1b

>Select Security group

Security groups

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

new-sg

sg-03bee61565e6236c9 VPC: vpc-09ed45b735e34ac4f

Listeners and routing

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

Remove

Protocol

HTTP

Port

80

1-65535

Default action

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

You can add up to 50 more tags.

>create target group with port 8000

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.

Target group name

TG1

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 primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

aws-production-vpc

vpc-09ed45b735e34ac4f

IPv4 VPC CIDR: 10.0.0.0/16

>select 2 and include as pending below

<input checked="" type="checkbox"/>	i-064c3fc2b032cfc1c	Running	new-sg	ap-south-1b
<input checked="" type="checkbox"/>	i-0db3427528eb37793	Running	new-sg	ap-south-1a

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

[Include as pending below](#)

>copy the alb dns and paste it on chrome

You don't see any website

>go to alb security click on that

Load balancer: alb

Listeners and rules | Network mapping | Resource map | **Security** | Monitoring | Integrations | Attributes | Capacity | Tags

Security groups (1) [Edit](#)

A security group is a set of firewall rules that control the traffic to your load balancer.

Security Group ID	Name	Description
sg-03bee61565e6236c9	new-sg	new-sg

>add inbound rule 8000 port

Inbound rules (3) [Manage tags](#) [Edit inbound rules](#)

Search

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-07e914d19508e6e41	IPv4	Custom TCP	TCP	8000
-	sgr-02b8aaef05fd75818	IPv4	SSH	TCP	22
-	sgr-05f3c72953d9fbd3a	IPv4	HTTP	TCP	80

>copy the alb DNS

Load balancers (1) [Actions](#) [Create load balancer](#)

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

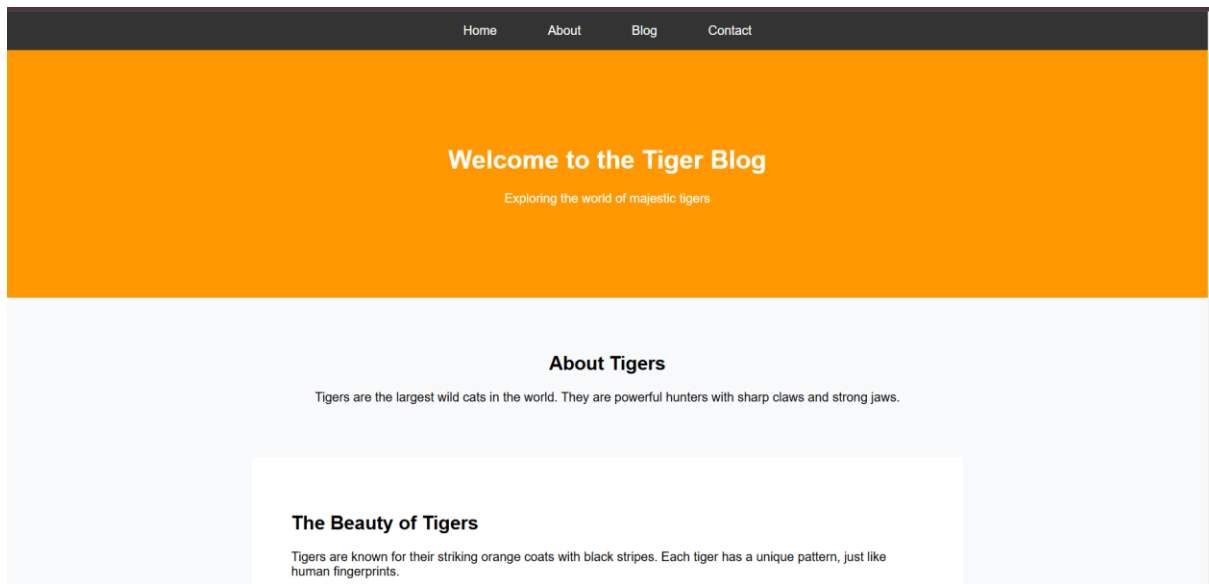
Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
alb	alb-889365918.ap-south-1.elb.amazonaws.com	Active	vpc-09ed45b735e34ac4f	2 Availability Zones	application	February 21, 2025, 15

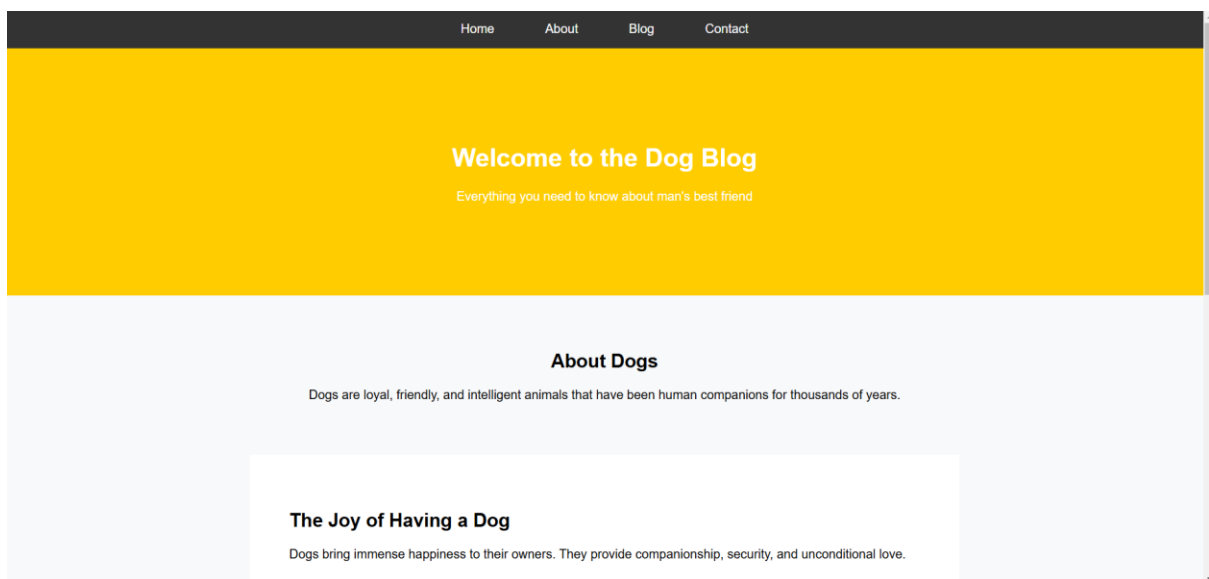
← → ↻ Not secure alb-889365918.ap-south-1.elb.amazonaws.com ☆

>paste it on chrome

You see your web-site is LIVE



>Refresh the website you can see your another website



Well done!!!!

Our project is done!!!!!!

Your web site work on private subnet.

We deploy our instance in private subnet. The private subnet does not face internet directly.

In This Project I Use EC2, VPC, Internet Gateway, Route Table, Nat Gateway, Elastic IP, HTML Code, Auto-Scaling Group, Target Group, Application Load Balancer, Launch Templates.

