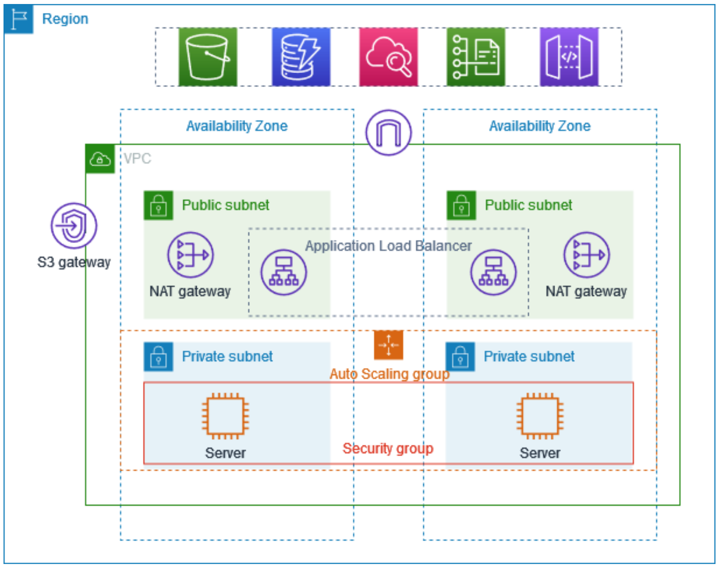
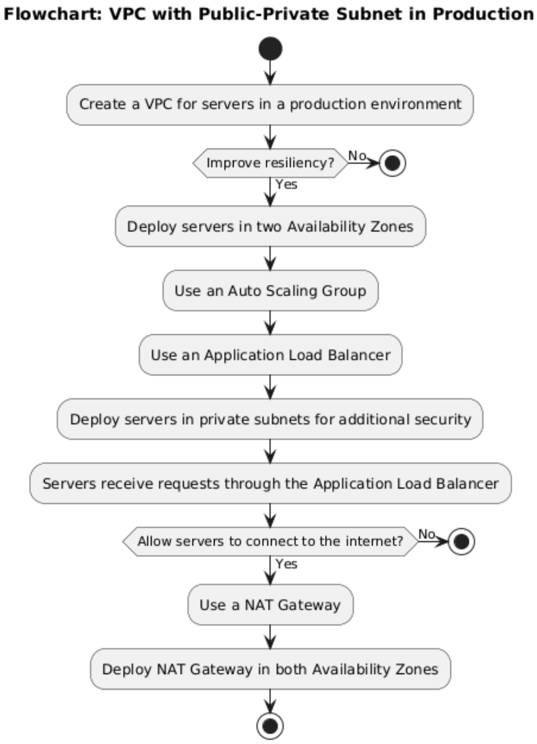
AWS Project Used in Production

VPC With Public Private Subnet in Production



About the Project:


Flowchart: VPC with Public-Private Subnet in Production
Create a VPC for servers in a production environment
Improve resiliency?
No
Yes
Deploy servers in two Availability Zones
Use an Auto Scaling Group
Use an Application Load Balancer
Deploy servers in private subnets for additional security
Servers receive requests through the Application Load Balancer
Allow servers to connect to the internet?
No
Yes
Use a NAT Gateway
Deploy NAT Gateway in both Availability Zones

Key Concepts:

1. Auto Scaling Group:

Let's say you want to deploy your application across two availability zones. Instead of manually creating EC2 instances twice, you can configure the Auto Scaling Group to maintain a minimum of two replicas. If your application starts receiving more requests and two servers are insufficient to handle the traffic, the Auto Scaling Group will automatically make a decision to scale the number of servers dynamically, based on the demand.

2. Load Balancer:

A Load Balancer is used to distribute incoming traffic across multiple servers to ensure efficient utilization and prevent overloading any single server. Additionally, it supports path-based and host-based routing, allowing requests to be directed to specific targets based on the URL path or hostname.

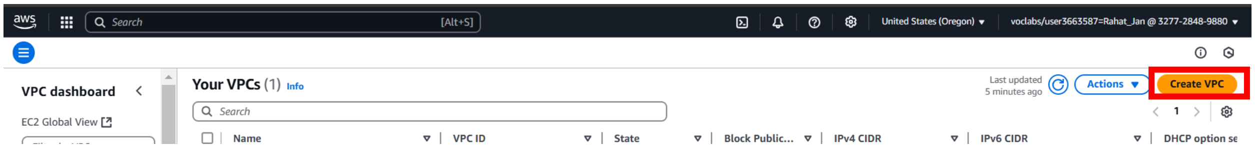
3. Bastion Host or Jump Server:

When EC2 instances are created in a private subnet, they do not have public IP addresses, meaning you cannot SSH into these instances directly. This is done to enhance security by avoiding public exposure. Instead, a Bastion Host (or Jump Host) is created in the public subnet. Using the Bastion Host, you can securely connect to EC2 instances in the private subnet. This approach allows for proper auditing of who is accessing the private subnet. Additionally, you can configure a set of rules on the Bastion Host to control and monitor the traffic that flows to the private subnet.

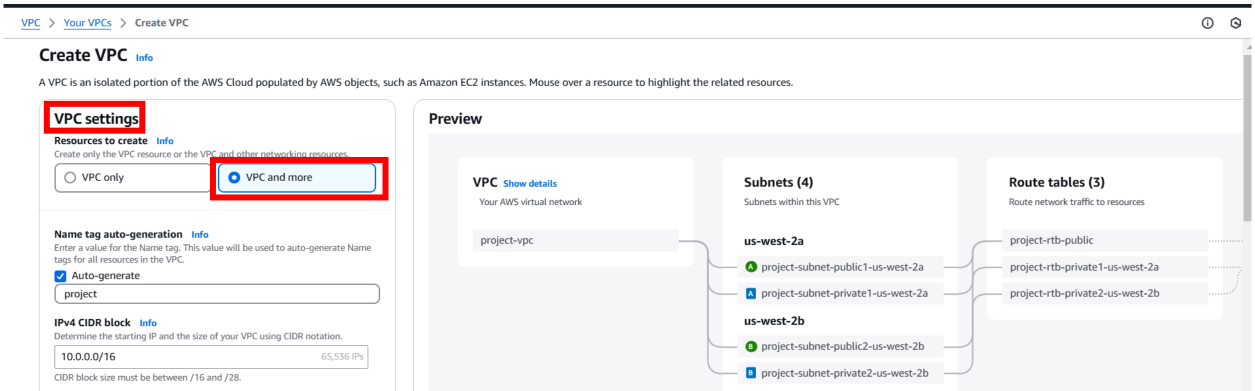
Project Implementation

Here are Following list of steps that are used to implement this project.

1. Click on create VPC



2. On VPC settings click on VPC and more


VPC > Your VPCs > Create VPC
Create VPC Info
VPC settings
Resources to create Info
○ VPC only
● VPC and more
Preview
VPC Show details
Your AWS virtual network
project-vpc
Subnets (4)
Subnets within this VPC
us-west-2a
project-subnet-public1-us-west-2a
project-subnet-private1-us-west-2a
us-west-2b
project-subnet-public2-us-west-2b
project-subnet-private2-us-west-2b
Route tables (3)
Route network traffic to resources
project-rtb-public
project-rtb-private1-us-west-2a
project-rtb-private2-us-west-2b

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

Create only the VPC resource or the VPC and other networking resources.

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

project

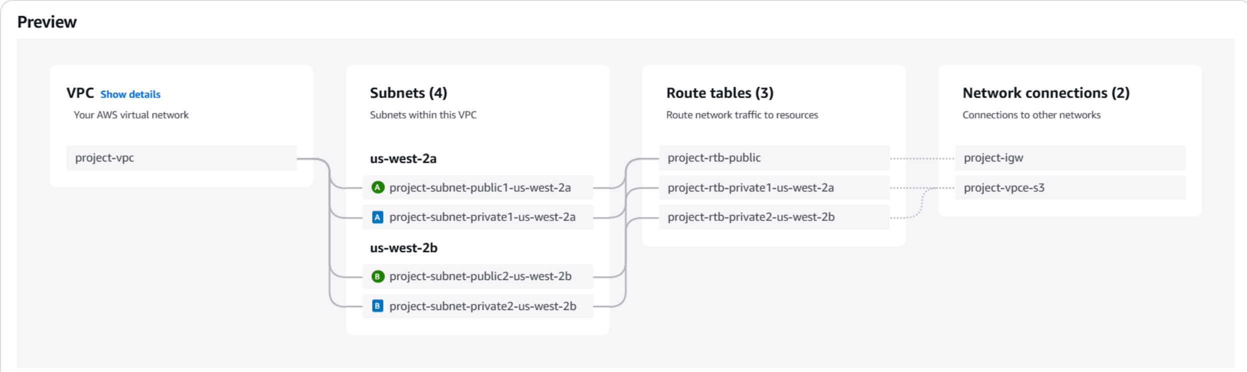
IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

AWS creates public private subnets in US-west 2a and US-west 2b. Subnets have to be attached to the route table. Route table is the one which defines how to route the traffic within the subnet. Route table has a destination and an internet gateway.



Preview

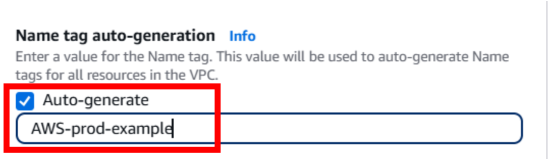
VPC Show details
Your AWS virtual network
project-vpc

Subnets (4)
Subnets within this VPC
us-west-2a
A project-subnet-public1-us-west-2a
A project-subnet-private1-us-west-2a
us-west-2b
B project-subnet-public2-us-west-2b
B project-subnet-private2-us-west-2b

Route tables (3)
Route network traffic to resources
project-rtb-public
project-rtb-private1-us-west-2a
project-rtb-private2-us-west-2b

Network connections (2)
Connections to other networks
project-igw
project-vpce-s3

3. Now select the name of the project.



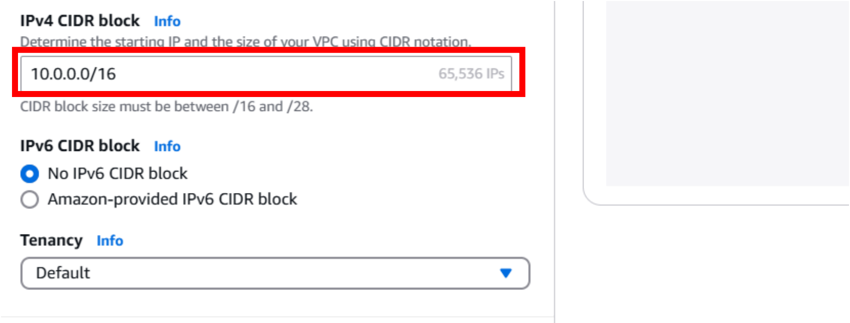
Name tag auto-generation Info

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

AWS-prod-example

4. Choose the subnet block. In my case I chose by default which is giving me 65536 IP addresses.



IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

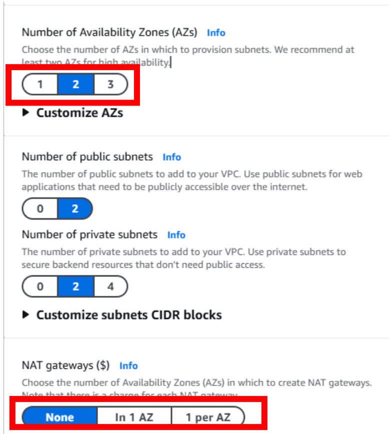
● No IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

Tenancy Info

Default ▼

5. Set the number of availability zone to 2 that we require in our project and one NAT gateway in 1 availability zone



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

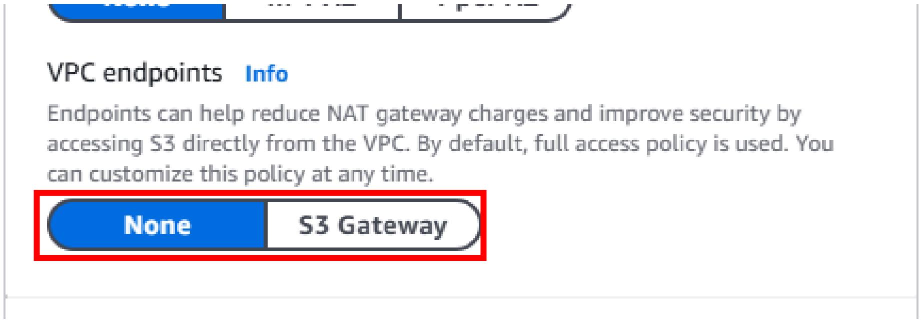
▸ **Customize subnets CIDR blocks**

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.

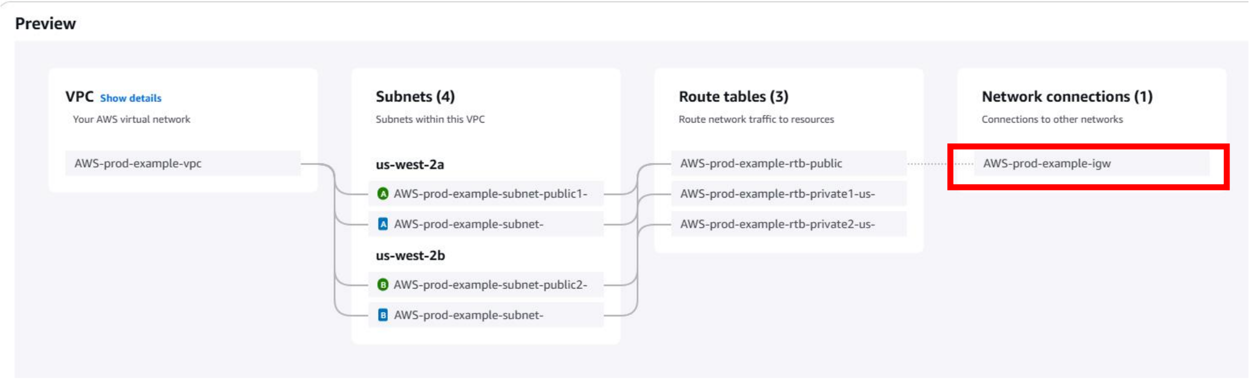
[✓ None] In 1 AZ 1 per AZ

6. For VPC endpoint click on none, You will notice in the diagram that there is no endpoint For S3 bucket.



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.



Preview

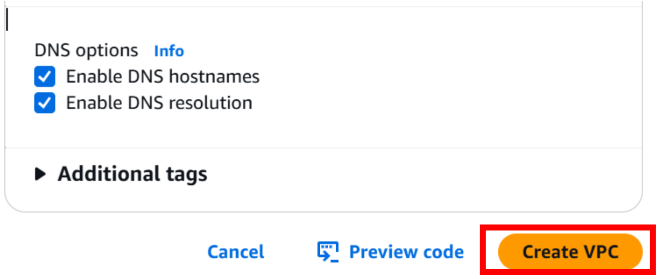
VPC Show details
Your AWS virtual network
AWS-prod-example-vpc

Subnets (4)
Subnets within this VPC
us-west-2a
AWS-prod-example-subnet-public1-
AWS-prod-example-subnet-
us-west-2b
AWS-prod-example-subnet-public2-
AWS-prod-example-subnet-

Route tables (3)
Route network traffic to resources
AWS-prod-example-rtb-public
AWS-prod-example-rtb-private1-us-
AWS-prod-example-rtb-private2-us-

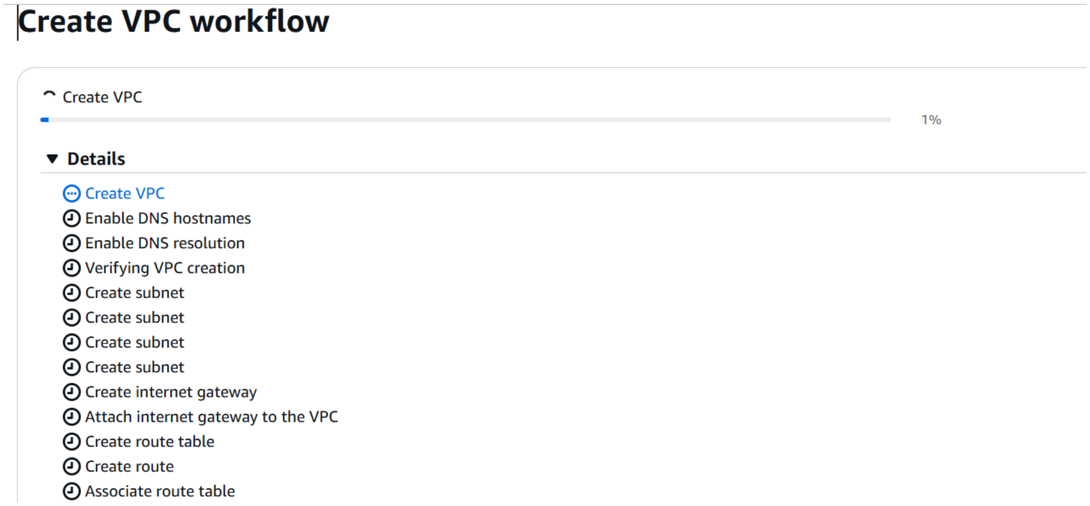
Network connections (1)
Connections to other networks
AWS-prod-example-igw

7. Now click on Create VPC.



DNS options Info
☑ Enable DNS hostnames
☑ Enable DNS resolution

▶ Additional tags



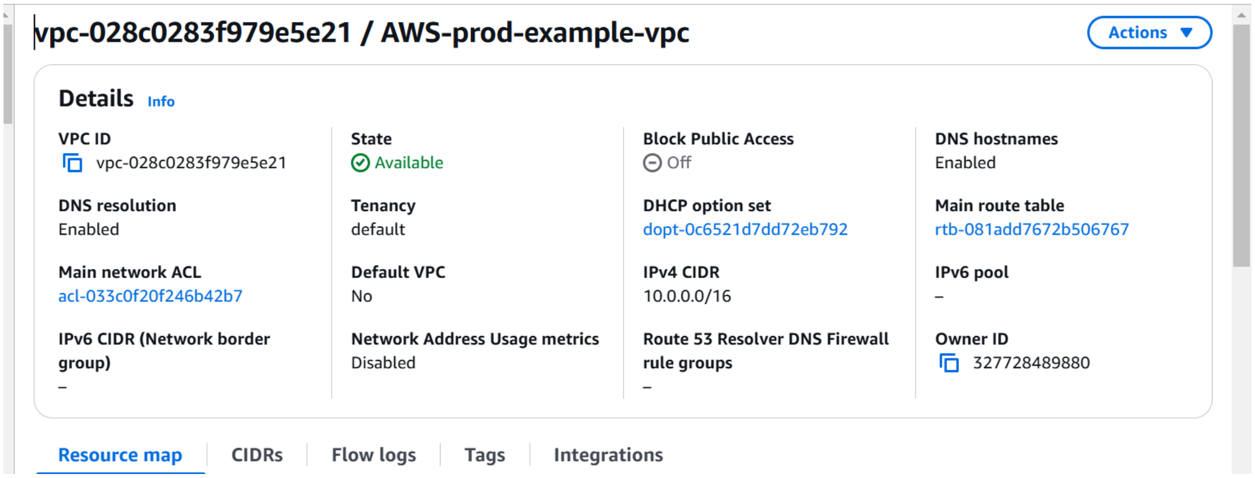
Create VPC workflow

Create VPC 1%

▼ Details

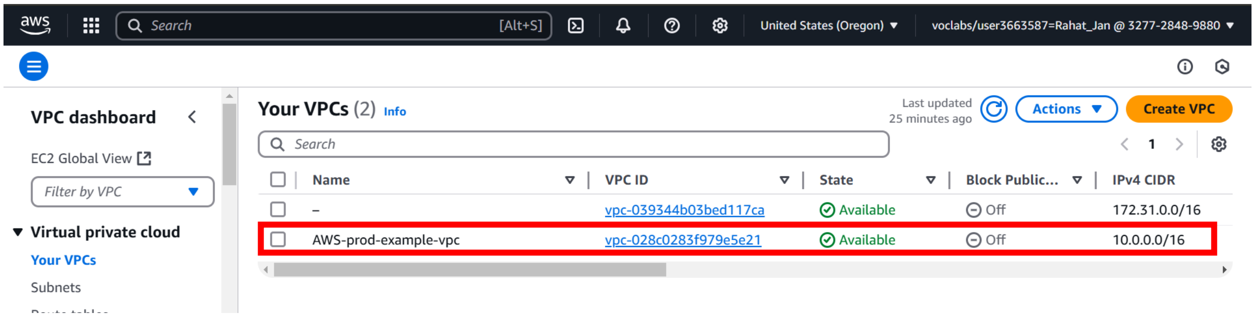
- Create VPC
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation
- Create subnet
- Create subnet
- Create subnet
- Create subnet
- Create internet gateway
- Attach internet gateway to the VPC
- Create route table
- Create route
- Associate route table

8. Now VPC has been created successfully



vpc-028c0283f979e5e21 / AWS-prod-example-vpc

VPC ID vpc-028c0283f979e5e21	**State** Available	**Block Public Access** Off	**DNS hostnames** Enabled
DNS resolution Enabled	**Tenancy** default	**DHCP option set** dopt-0c6521d7dd72eb792	**Main route table** rtb-081add7672b506767
Main network ACL acl-033c0f20f246b42b7	**Default VPC** No	**IPv4 CIDR** 10.0.0.0/16	**IPv6 pool** –
IPv6 CIDR (Network border group) –	**Network Address Usage metrics** Disabled	**Route 53 Resolver DNS Firewall rule groups** –	**Owner ID** 327728489880

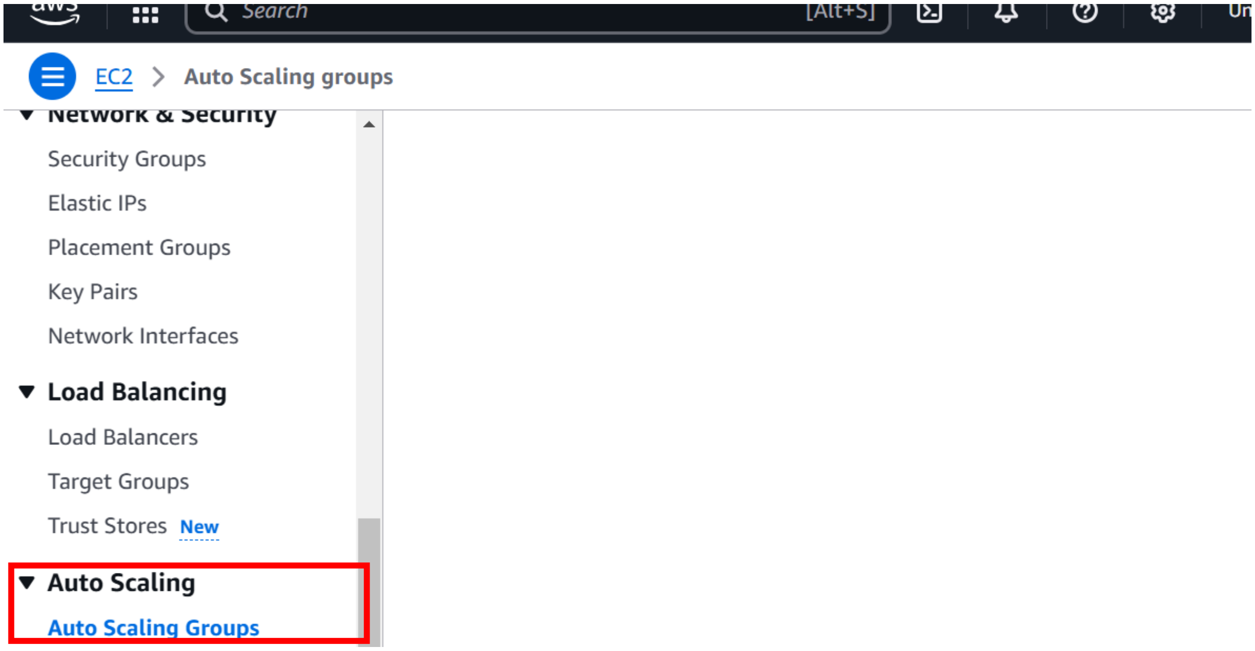


Your VPCs (2) Info

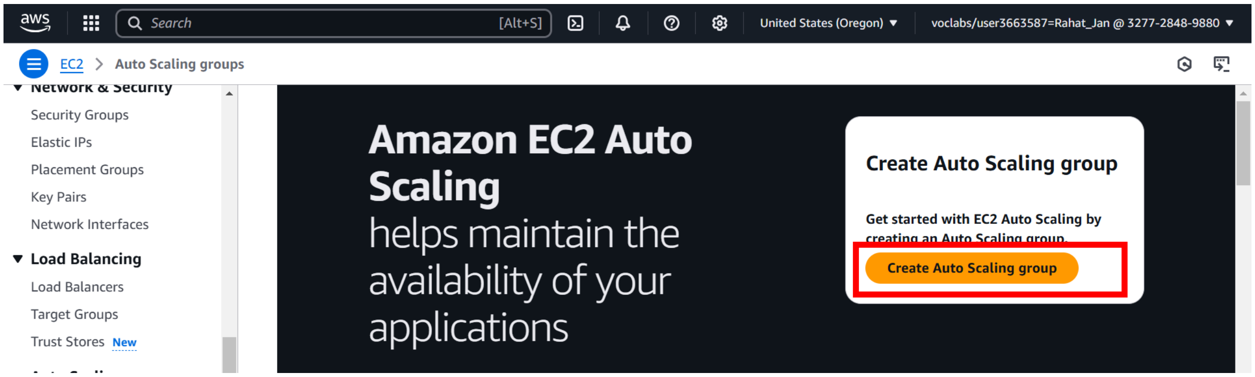
□	Name	VPC ID	State	Block Public...	IPv4 CIDR
□	–	vpc-039344b03bed117ca	Available	Off	172.31.0.0/16
□	AWS-prod-example-vpc	vpc-028c0283f979e5e21	Available	Off	10.0.0.0/16

Creation of Auto-scaling group:

9. Click on EC2 and select the autoscaling option



10. Click on creating Auto-scaling groups

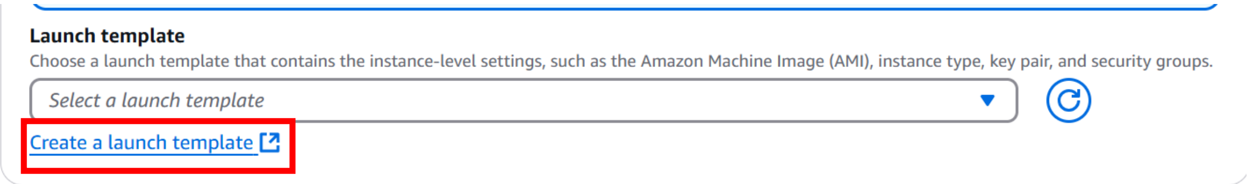


Amazon EC2 Auto Scaling helps maintain the availability of your applications

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

11. click on launch template. Autoscaling in AWS cannot be created directly. For this you can use the launch template. You can use that template in multiple autoscaling groups and acts as a reference

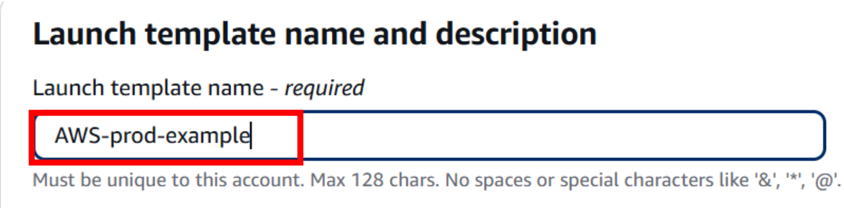


Launch template

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 ▾

12. Give the name to template



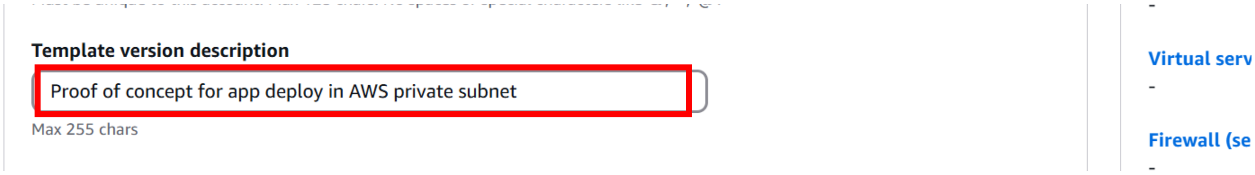
Launch template name and description

Launch template name - *required*

AWS-prod-example

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

13. Give the description of your template

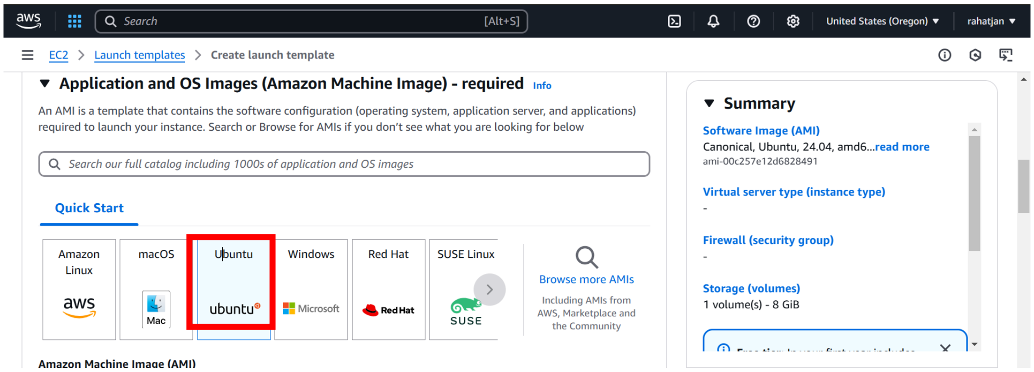


Template version description

Proof of concept for app deploy in AWS private subnet

Max 255 chars

14. Select the OS, For this project I am selecting ubuntu



▾ **Application and OS Images (Amazon Machine Image) - required**

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)

▾ **Summary**

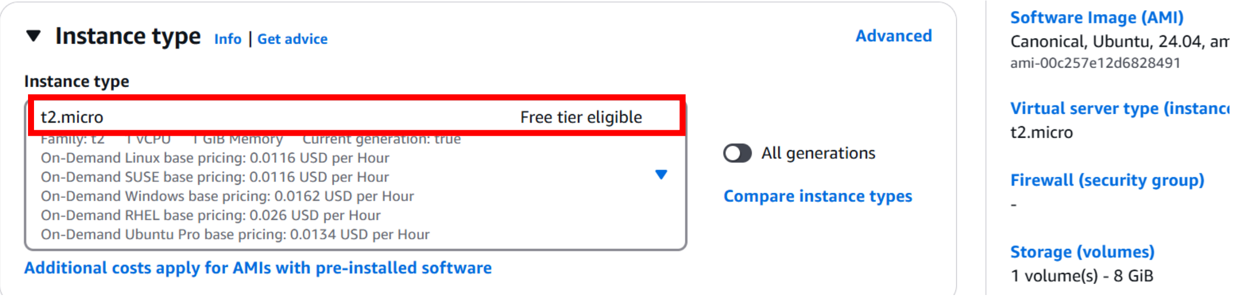
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00c257e12d6828491

Virtual server type (instance type)
-

Firewall (security group)
-

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

15. Choose the Instance type



▼ **Instance type** Info | Get advice — Advanced

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

Additional costs apply for AMIs with pre-installed software

All generations

Compare instance types

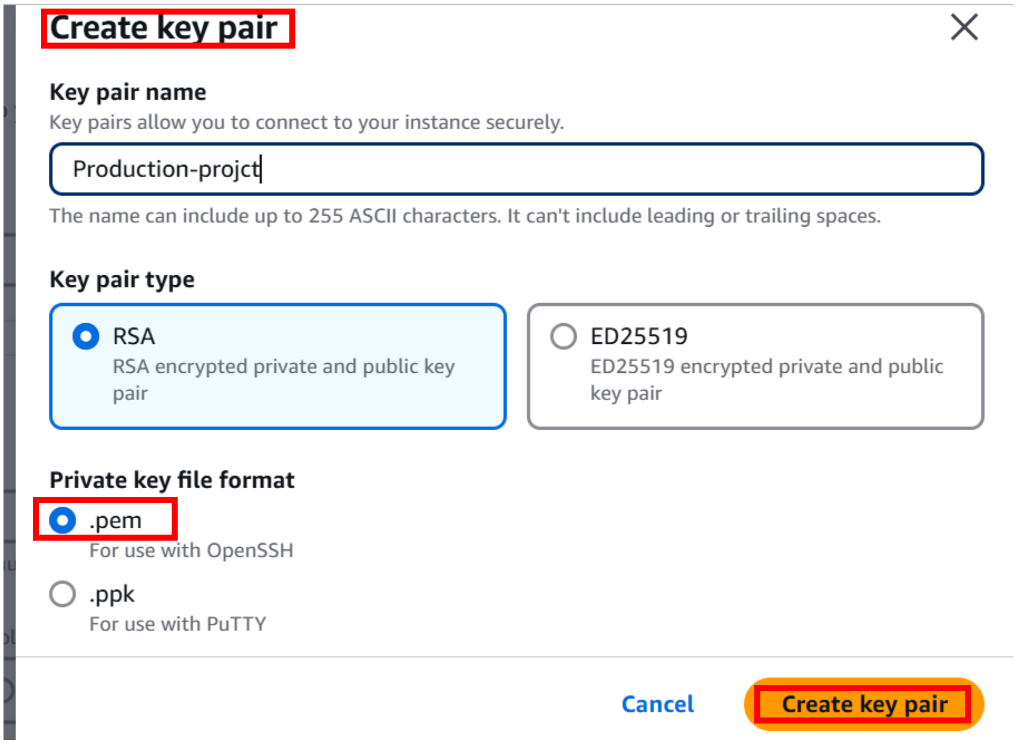
Software Image (AMI)
Canonical, Ubuntu, 24.04, am
ami-00c257e12d6828491

Virtual server type (instanc
t2.micro

Firewall (security group)
-

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

16. Create the Key value pair and save it as .pem file



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

Production-projct

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA — RSA encrypted private and public key pair

○ ED25519 — ED25519 encrypted private and public key pair

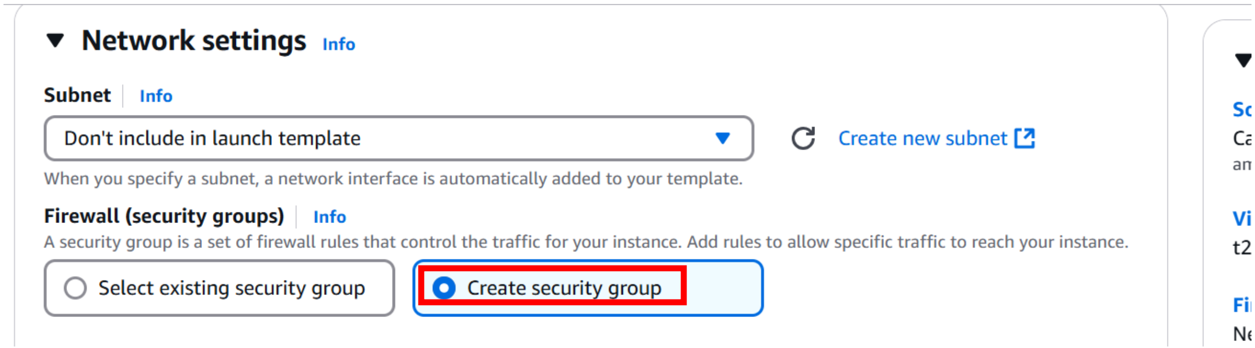
Private key file format

● .pem — For use with OpenSSH

○ .ppk — For use with PuTTY

Cancel | Create key pair

17. Click to create a new security group



▼ **Network settings** Info

Subnet Info

Don't include in launch template ▼ — Create new subnet

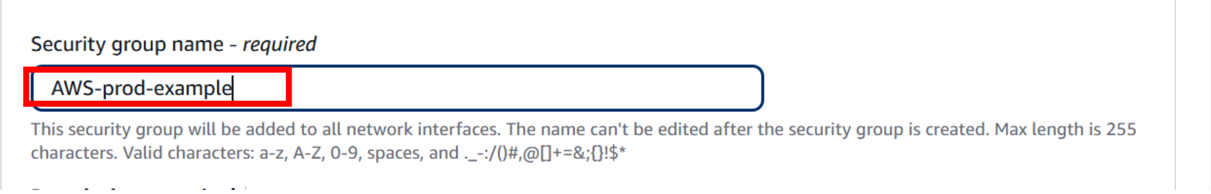
When you specify a subnet, a network interface is automatically added to your template.

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

18. Select the name of the group

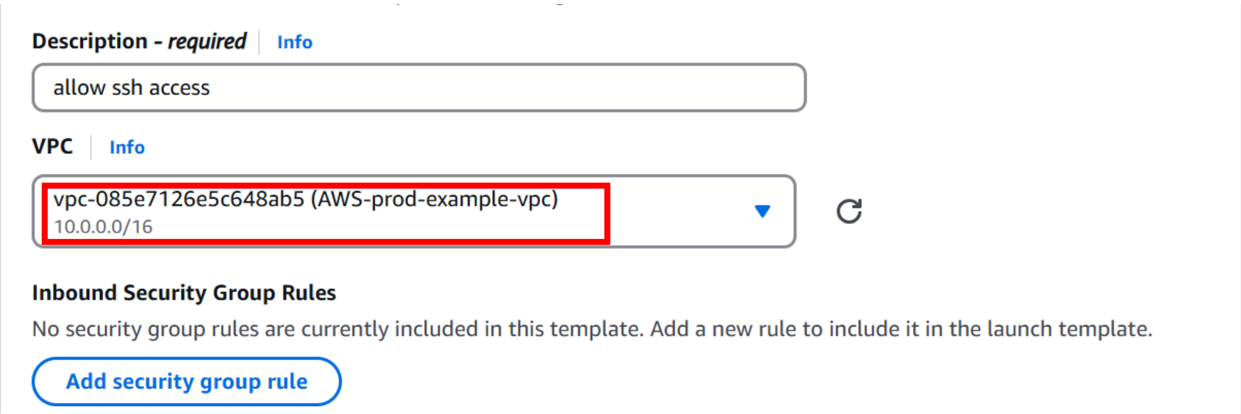


Security group name - *required*

AWS-prod-example

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 ._-:/()#,@[]+=&;{}!$*

19. Select the VPC that we just created



Description - *required* Info

allow ssh access

VPC Info

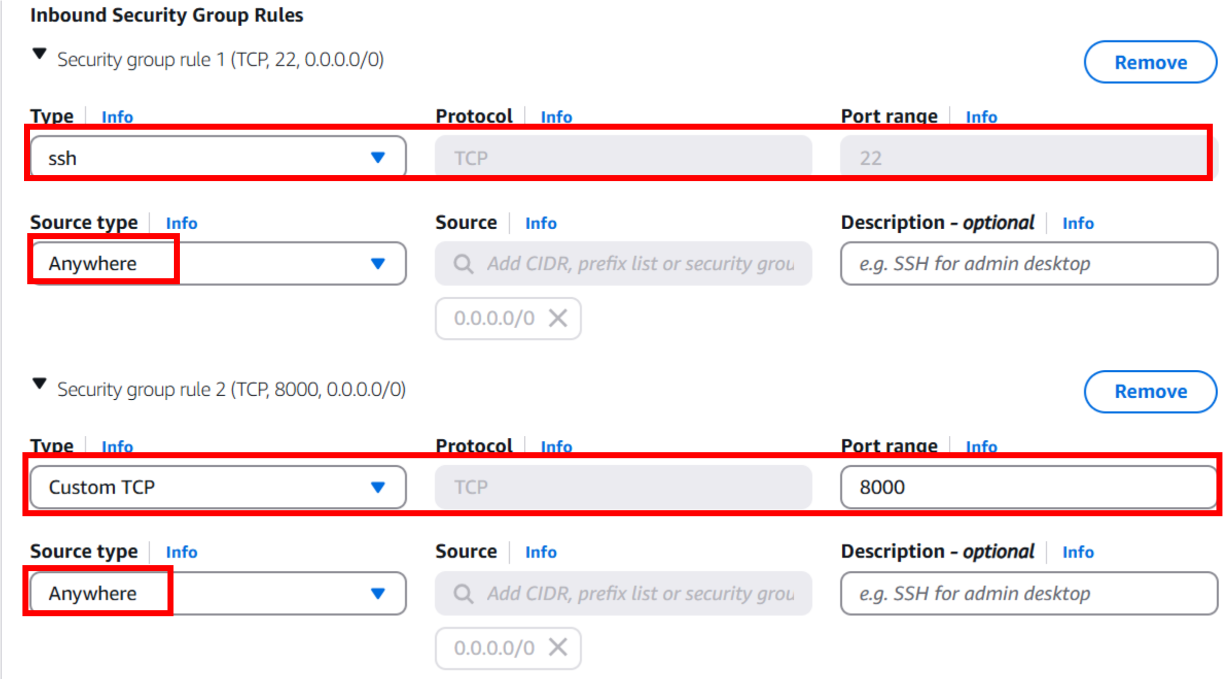
vpc-085e7126e5c648ab5 (AWS-prod-example-vpc)
10.0.0.0/16

Inbound Security Group Rules

No security group rules are currently included in this template. Add a new rule to include it in the launch template.

Setting the Inbound rules for the application:

20. The port that we are using for accessing the application is port 8000 and adding the SSH rule on port 22 and setting the source type to Anywhere



Inbound Security Group Rules

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

Type: ssh | Protocol: TCP | Port range: 22

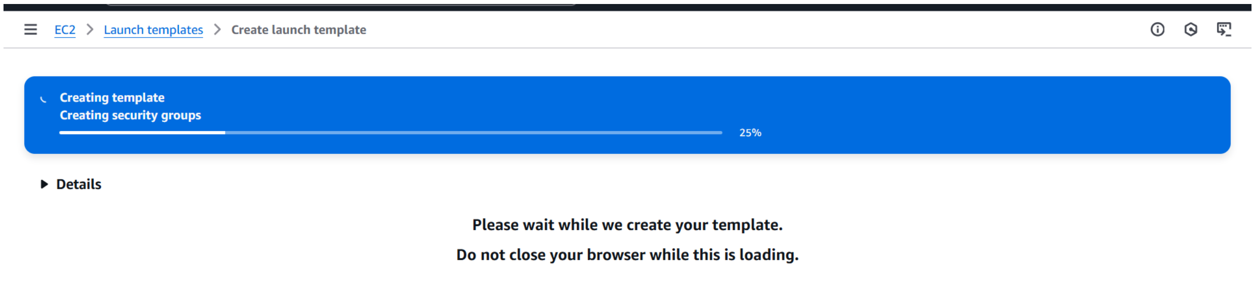
Source type: Anywhere | Source: 0.0.0.0/0 | Description - *optional*: e.g. SSH for admin desktop

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

Type: Custom TCP | Protocol: TCP | Port range: 8000

Source type: Anywhere | Source: 0.0.0.0/0 | Description - *optional*: e.g. SSH for admin desktop

21. Now click on Launch template

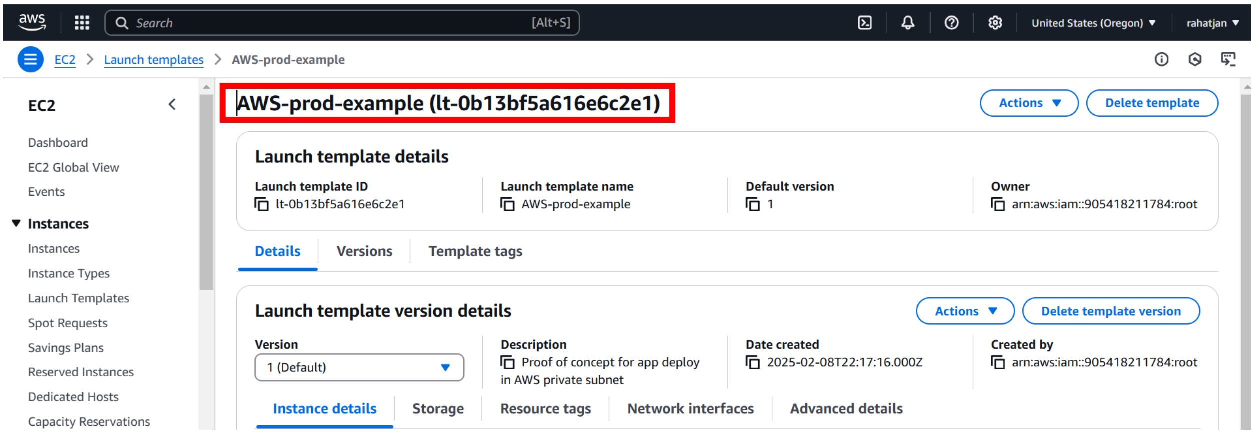


EC2 > Launch templates > Create launch template

Creating template
Creating security groups
25%

Details

Please wait while we create your template.
Do not close your browser while this is loading.



EC2 > Launch templates > AWS-prod-example

AWS-prod-example (lt-0b13bf5a616e6c2e1)

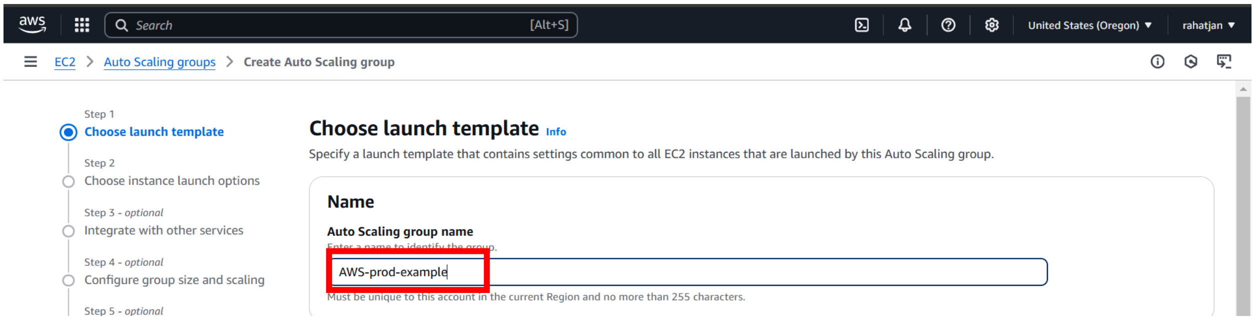
Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-0b13bf5a616e6c2e1	AWS-prod-example	1	arn:aws:iam::905418211784:root

Launch template version details

Version	Description	Date created	Created by
1 (Default)	Proof of concept for app deploy in AWS private subnet	2025-02-08T22:17:16.000Z	arn:aws:iam::905418211784:root

22. Now **Choose launch template**



EC2 > Auto Scaling groups > Create Auto Scaling group

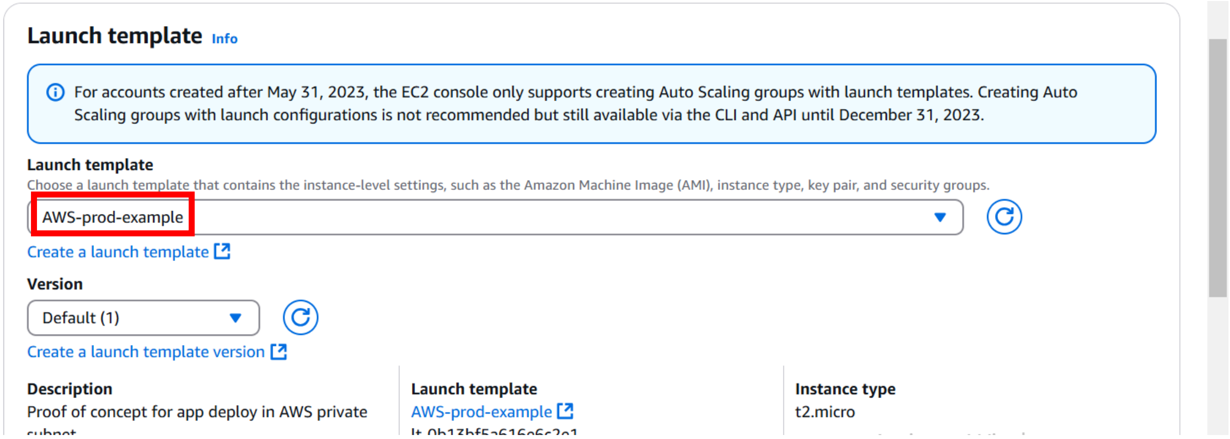
Choose launch template Info

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

23. Now select the template that we just created



Launch template Info

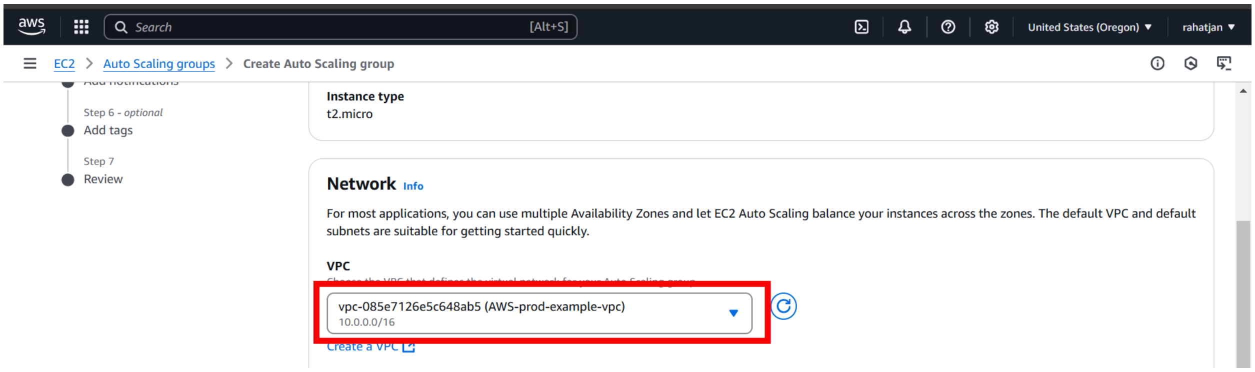
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.
AWS-prod-example ▼
Create a launch template

Version
Default (1) ▼
Create a launch template version

Description	Launch template	Instance type
Proof of concept for app deploy in AWS private subnet	AWS-prod-example lt-0b13bf5a616e6c2e1	t2.micro

24. Choose the VPC that we have created



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.

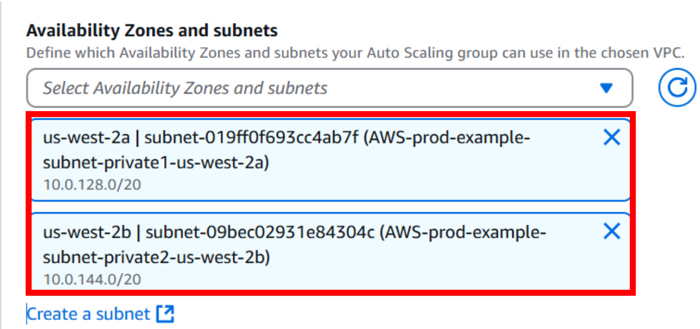
VPC

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

vpc-085e7126e5c648ab5 (AWS-prod-example-vpc)
10.0.0.0/16

Create a VPC

25. Choose the availability zones as private subnet



Availability Zones and subnets

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

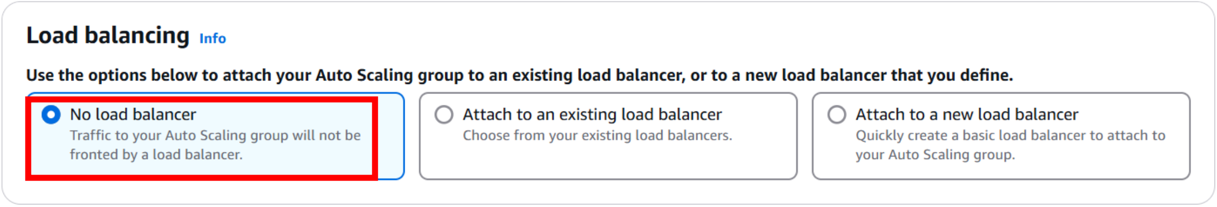
Select Availability Zones and subnets ▼

us-west-2a | subnet-019ff0f693cc4ab7f (AWS-prod-example-subnet-private1-us-west-2a)
10.0.128.0/20

us-west-2b | subnet-09bec02931e84304c (AWS-prod-example-subnet-private2-us-west-2b)
10.0.144.0/20

Create a subnet

26. Click on no load balancer, we will create the load balancer in the public subnet



Load balancing Info

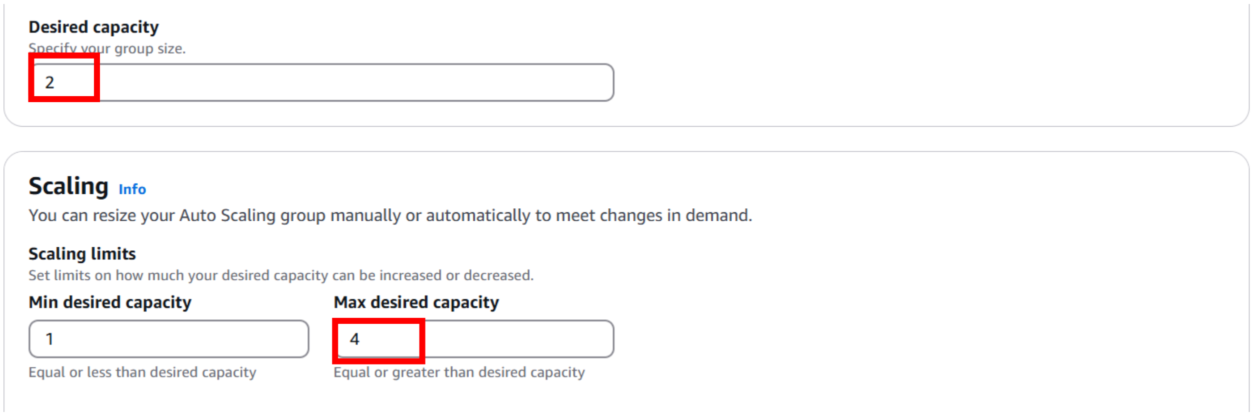
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

● 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.

27. Select the group capacity



Desired capacity

Specify your group size.

2

Scaling Info

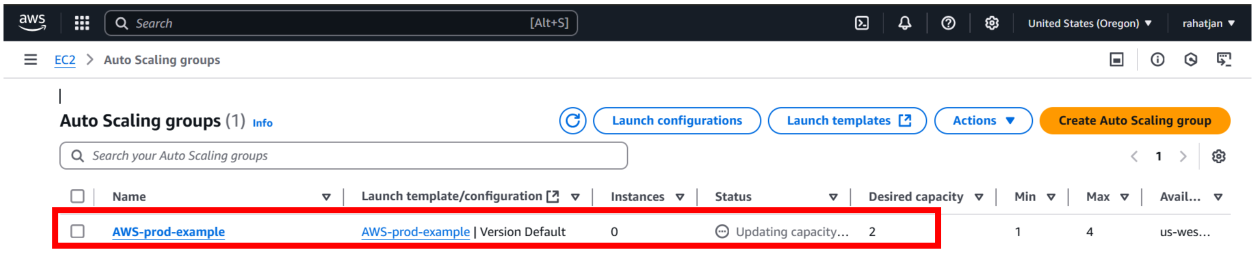
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

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

Min desired capacity	Max desired capacity
1	4
Equal or less than desired capacity	Equal or greater than desired capacity

28. Click on create button

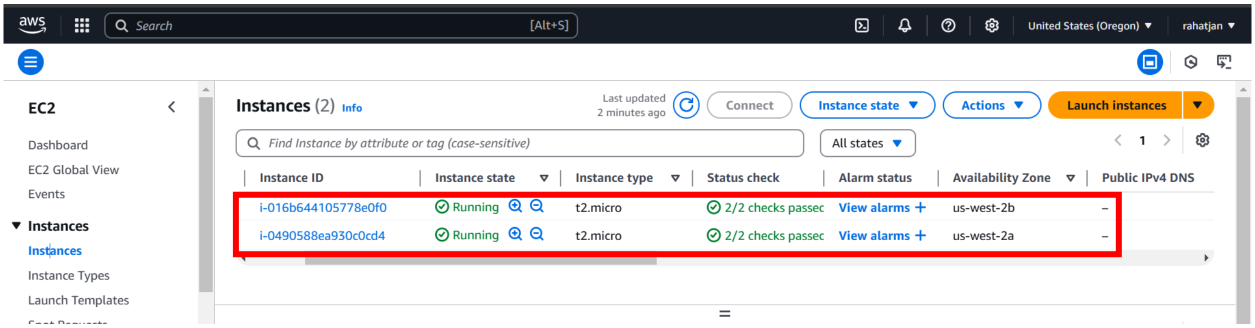


Auto Scaling groups (1)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Avail...
AWS-prod-example	AWS-prod-example | Version Default	0	Updating capacity...	2	1	4	us-wes...

Now the scaling group has been launched successfully

Two instances of ec2 were created successfully and 2 instances are created in different availability zones



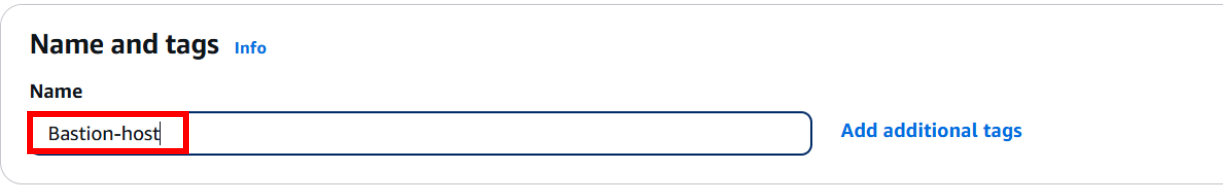
Instances (2)

Find Instance by attribute or tag (case-sensitive)

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
i-016b644105778e0f0	Running	t2.micro	2/2 checks passec	View alarms +	us-west-2b	–
i-0490588ea930c0cd4	Running	t2.micro	2/2 checks passec	View alarms +	us-west-2a	–

Creating the Bastion Host:

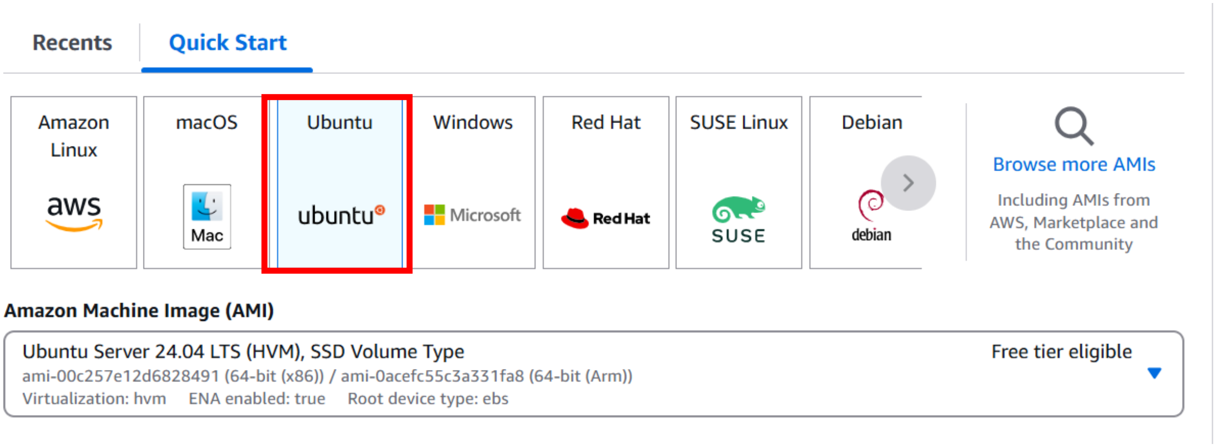
29. In ec2 click on launch instance and give it a name



Name and tags

Name: Bastion-host

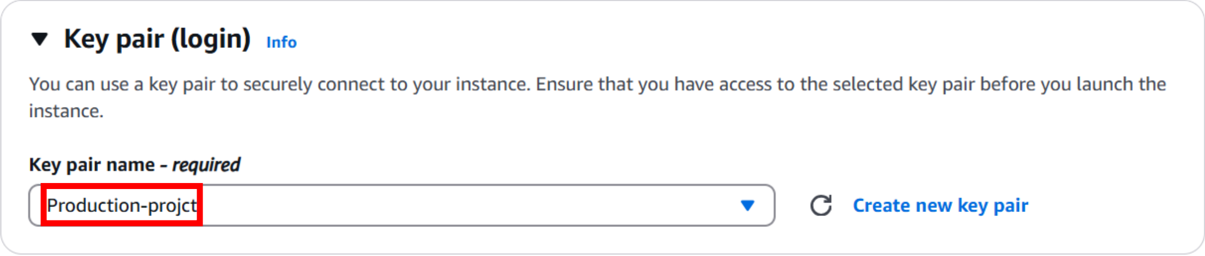
30. Choose ubuntu as an image



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00c257e12d6828491 (64-bit (x86)) / ami-0acefc55c3a331fa8 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

31. Provide the key pair



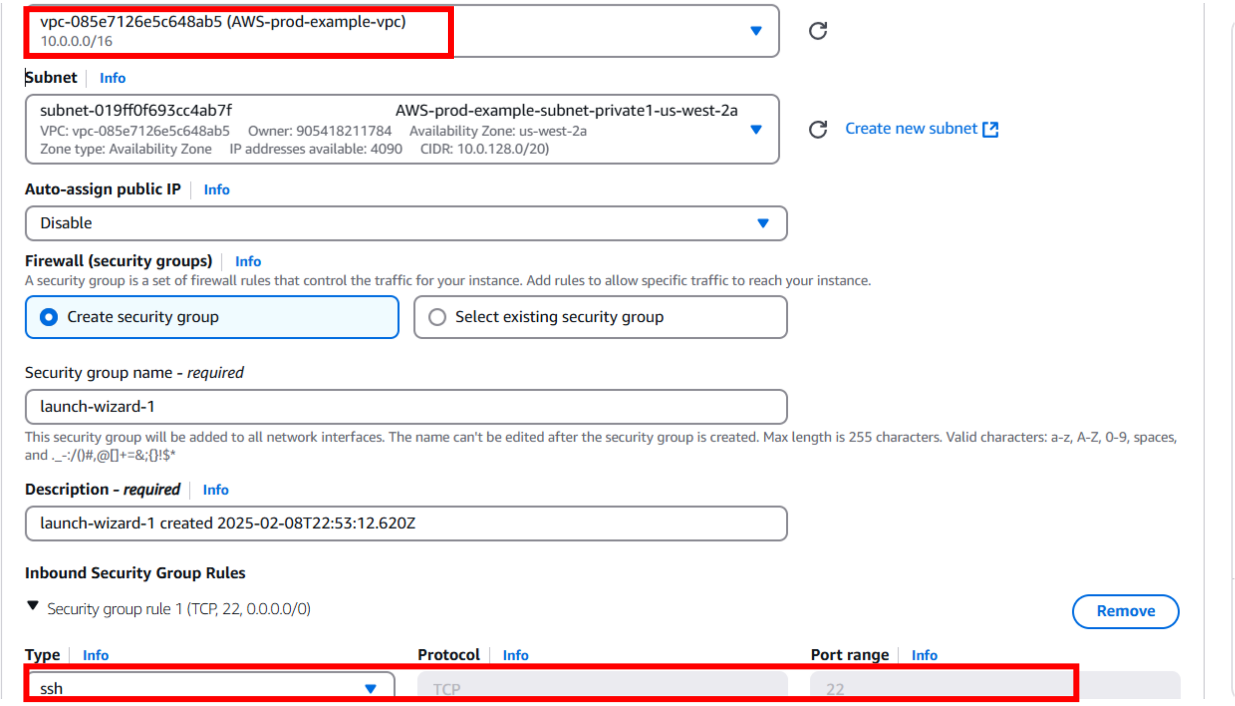
Key pair (login) Info

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*

Production-projct

32. Make sure to add a security group that has access to ssh and make sure that bastion host is created in the same VPC



vpc-085e7126e5c648ab5 (AWS-prod-example-vpc)
10.0.0.0/16

Subnet Info

subnet-019ff0f693cc4ab7f AWS-prod-example-subnet-private1-us-west-2a
VPC: vpc-085e7126e5c648ab5 Owner: 905418211784 Availability Zone: us-west-2a
Zone type: Availability Zone IP addresses available: 4090 CIDR: 10.0.128.0/20)

Auto-assign public IP Info

Disable

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

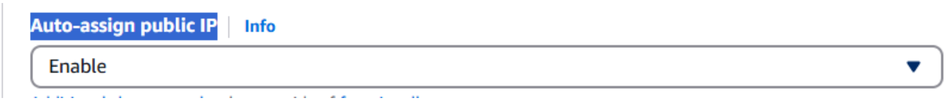
launch-wizard-1 created 2025-02-08T22:53:12.620Z

Inbound Security Group Rules

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

Type Info	Protocol Info	Port range Info
ssh	TCP	22

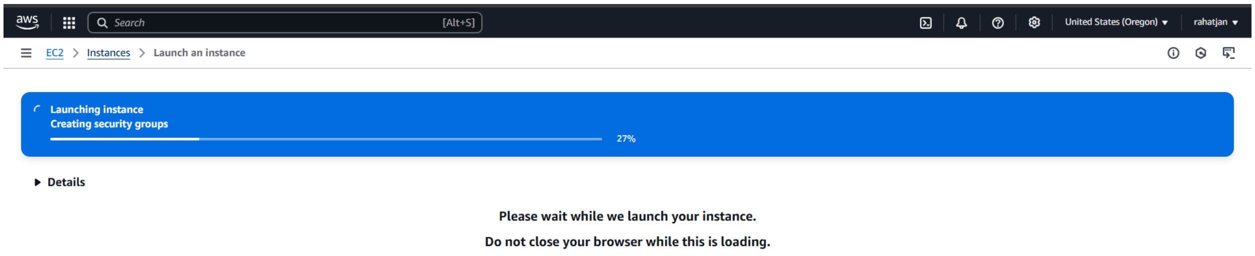
33. Enable the **Auto-assign public IP**



Auto-assign public IP | Info

Enable

34. Now launch the instance

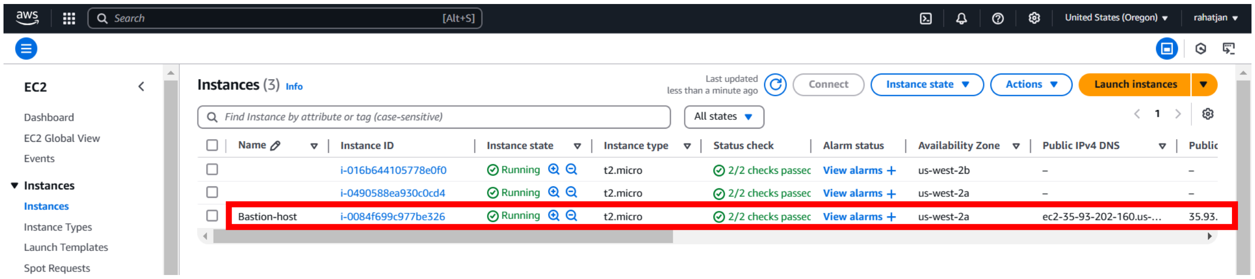


Launching instance
Creating security groups
27%

▸ Details

Please wait while we launch your instance.
Do not close your browser while this is loading.

35. Now bastion-host has been created successfully



Instances (3) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

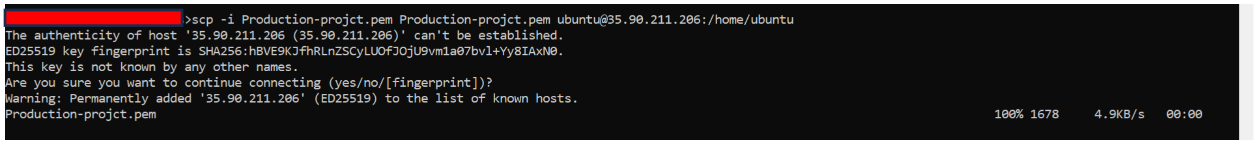
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
□		i-016b644105778e0f0	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-west-2b	–	–
□		i-0490588ea930c0cd4	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-west-2a	–	–
□	Bastion-host	i-0084f699c977be326	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-west-2a	ec2-35-93-202-160.us-...	35.93.

SSH into private subnet using bastion host:

36. First lets do the secure copy, The command transfers the file rahatjan.pem (likely a private key file) from the local system to the /home/ubuntu directory on the remote server at 35.93.189.166. This is commonly done to:

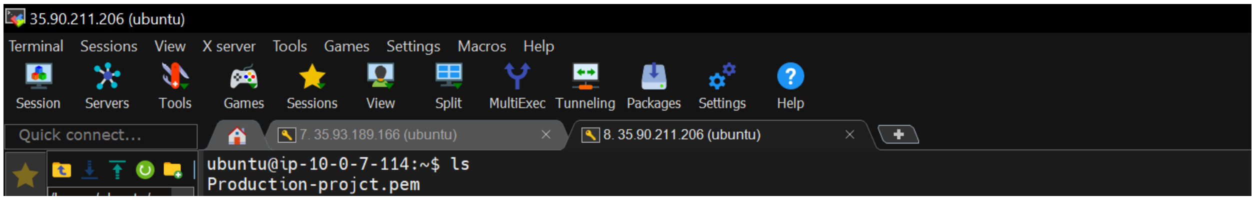
- Share the file for further use on the remote server.

- Prepare the remote environment for SSH-based operations.



```
[redacted]>scp -i Production-projct.pem Production-projct.pem ubuntu@35.90.211.206:/home/ubuntu
The authenticity of host '35.90.211.206 (35.90.211.206)' can't be established.
ED25519 key fingerprint is SHA256:hBVE9KJfhRLnZSCyLUOfJOjU9vm1a07bvl+Yy8IAxN0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '35.90.211.206' (ED25519) to the list of known hosts.
Production-projct.pem                                                                        100% 1678     4.9KB/s   00:00
```

37. File is successfully uploaded
Now ssh into the ubuntu machine



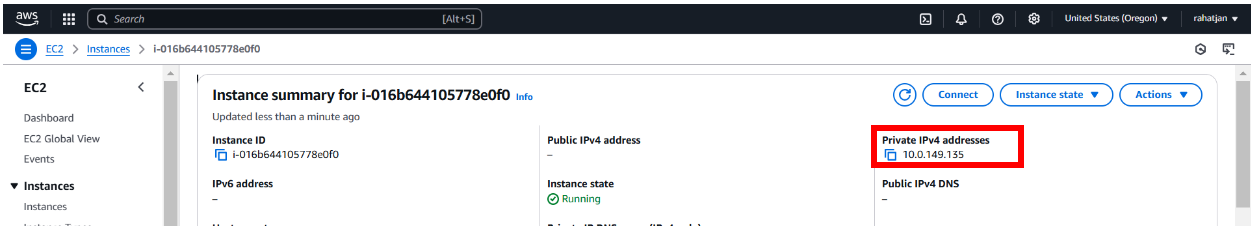
35.90.211.206 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
7. 35.93.189.166 (ubuntu)
8. 35.90.211.206 (ubuntu)

```
ubuntu@ip-10-0-7-114:~$ ls
Production-projct.pem
```

Now you can see that the .pem file is available in the bastion host.

With this setup, I will proceed to deploy a straightforward application on one of the instances we created within the private subnet. To accomplish this, we will establish an SSH connection to one of the instances, utilizing the same terminal environment for seamless access and control.

38. Take the private ip address of any of instances and ssh into it, In my case I am taking 10.0.149.135



Instance summary for i-016b644105778e0f0 Info
Updated less than a minute ago

Instance ID
i-016b644105778e0f0

Public IPv4 address
–

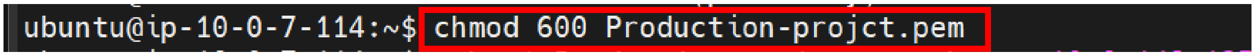
Private IPv4 addresses
10.0.149.135

IPv6 address
–

Instance state
Running

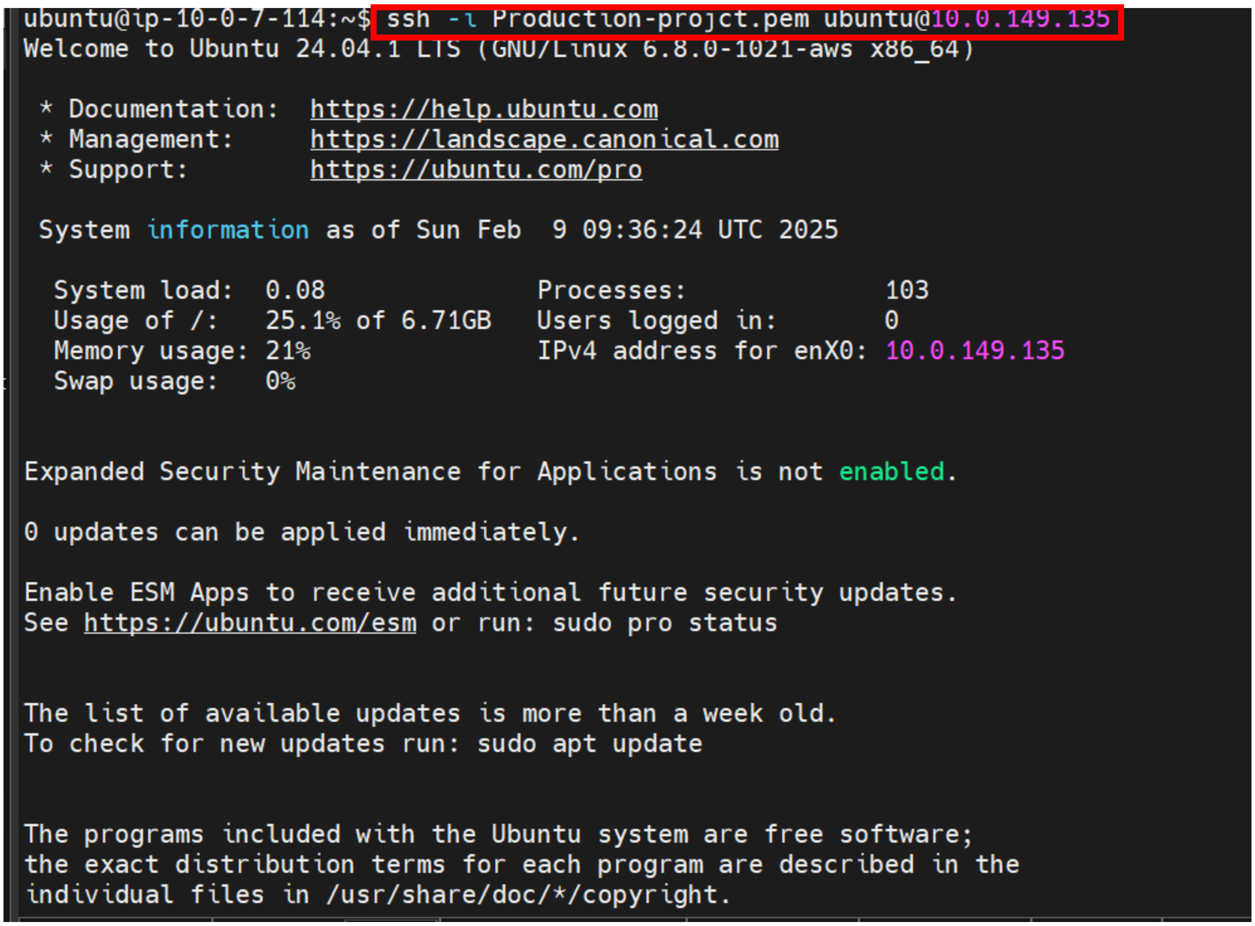
Public IPv4 DNS
–

39. Give 600 permissions to .pem file



```
ubuntu@ip-10-0-7-114:~$ chmod 600 Production-projct.pem
```

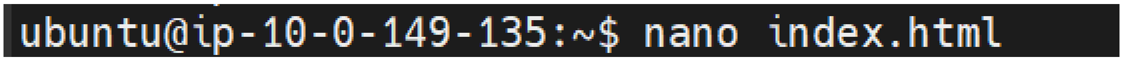
40. Now ssh into one of the private subnet



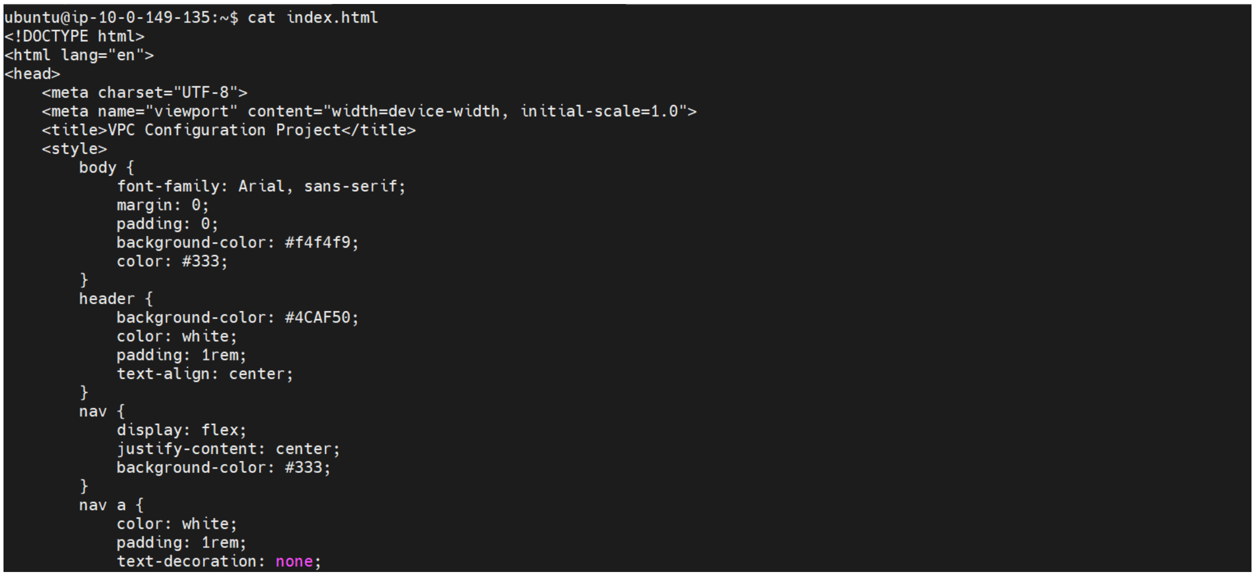
```
ubuntu@ip-10-0-7-114:~$ ssh -i Production-projct.pem ubuntu@10.0.149.135
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

 System information as of Sun Feb  9 09:36:24 UTC 2025

  System load:  0.08              Processes:             103
  Usage of /:   25.1% of 6.71GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 10.0.149.135
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

Now we are able to login to the private instance as well

41. Now we will install a simple python application in it
42. First create a html file

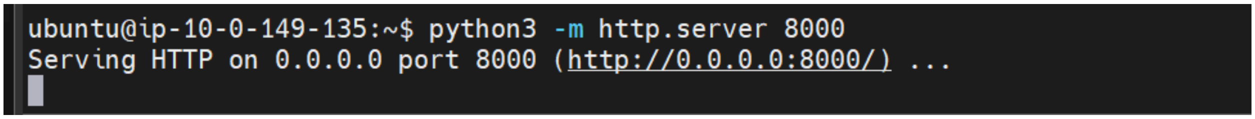


```
ubuntu@ip-10-0-149-135:~$ nano index.html
```



```
ubuntu@ip-10-0-149-135:~$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>VPC Configuration Project</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f9;
            color: #333;
        }
        header {
            background-color: #4CAF50;
            color: white;
            padding: 1rem;
            text-align: center;
        }
        nav {
            display: flex;
            justify-content: center;
            background-color: #333;
        }
        nav a {
            color: white;
            padding: 1rem;
            text-decoration: none;
```

43. Now run the python server by using the following command python3 -m http.server 8000

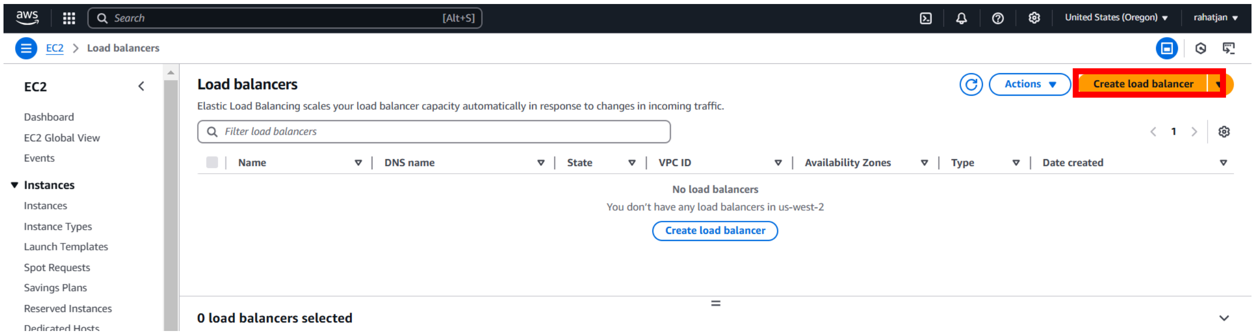


```
ubuntu@ip-10-0-149-135:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Application Load Balancer:

Now we will try to do is create a load balancer and attach private subnet instances as target group that will be our final stage.

44. Go to AWS and search for load balancer and click on create load balancer



Load balancers

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

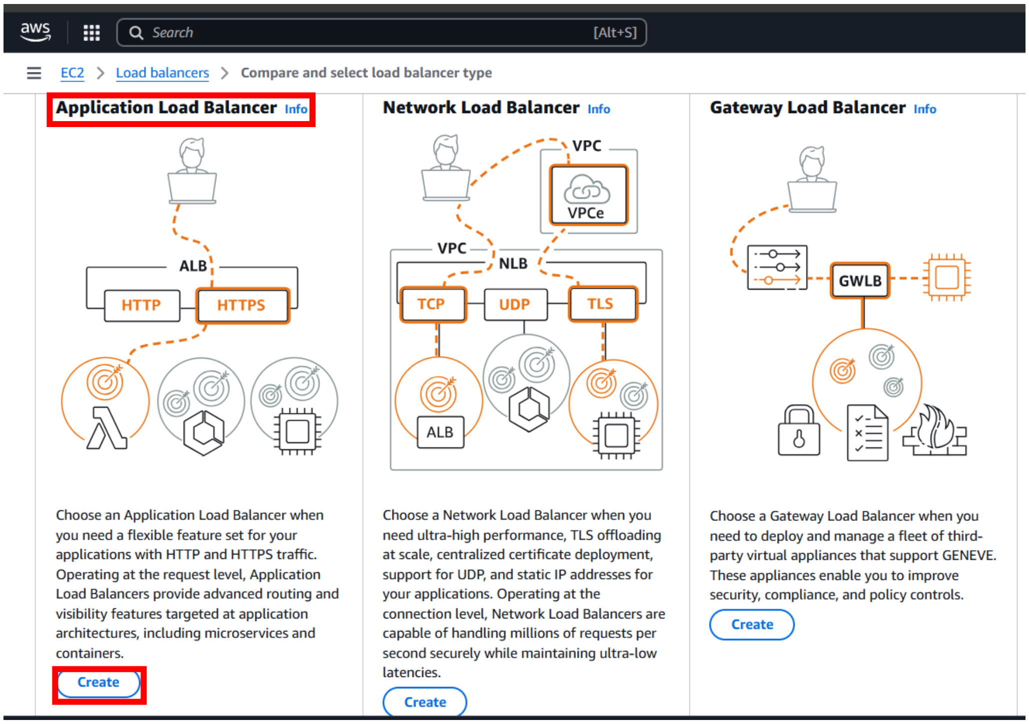
No load balancers

You don't have any load balancers in us-west-2

0 load balancers selected

45. In our used case, go with application load balancer which is L7 load balancer.

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 the 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 action.



EC2 > Load balancers > Compare and select load balancer type

Application Load Balancer Info

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.

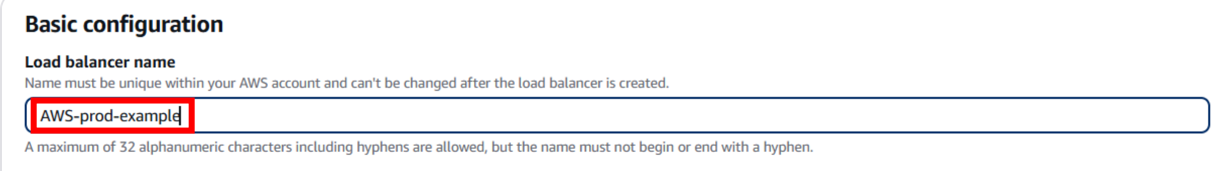
Network Load Balancer Info

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.

Gateway Load Balancer Info

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.

46. Give a name to your load balancer



Basic configuration

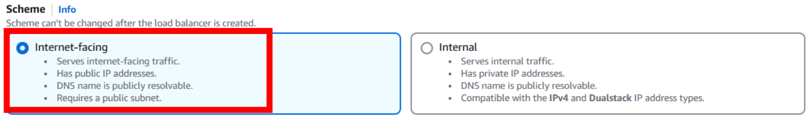
Load balancer name

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

AWS-prod-example

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

47. The scheme should be in internet-facing



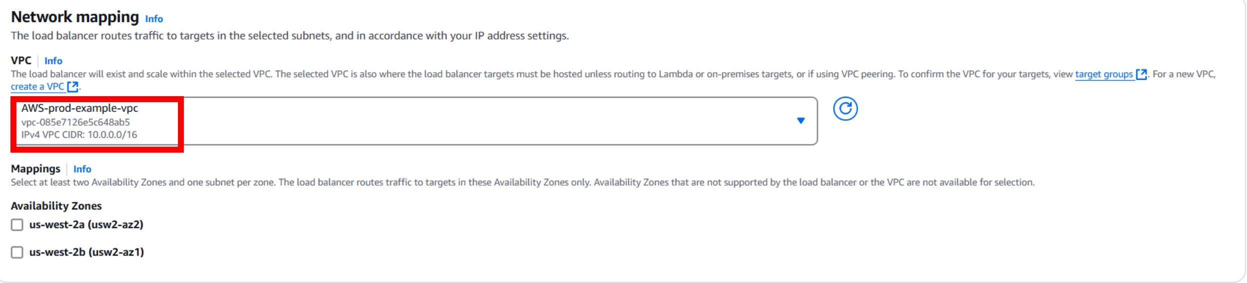
Scheme | Info

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.

48. Select the VPC that we created



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-prod-example-vpc
vpc-085e7126e5c648ab5
IPv4 VPC CIDR: 10.0.0.0/16

Mappings | Info

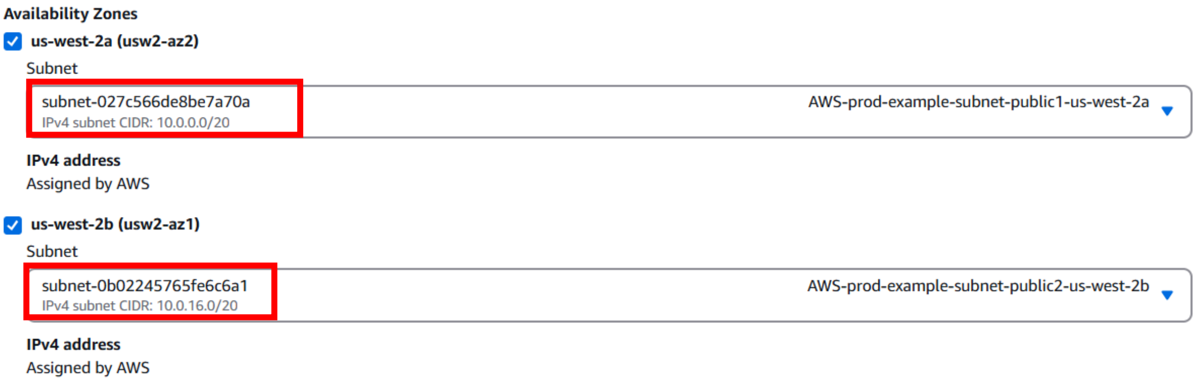
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

Availability Zones

□ us-west-2a (usw2-az2)

□ us-west-2b (usw2-az1)

49. Select both availability zones and it should be the public subnet



Availability Zones

☑ us-west-2a (usw2-az2)

Subnet

subnet-027c566de8be7a70a — AWS-prod-example-subnet-public1-us-west-2a
IPv4 subnet CIDR: 10.0.0.0/20

IPv4 address
Assigned by AWS

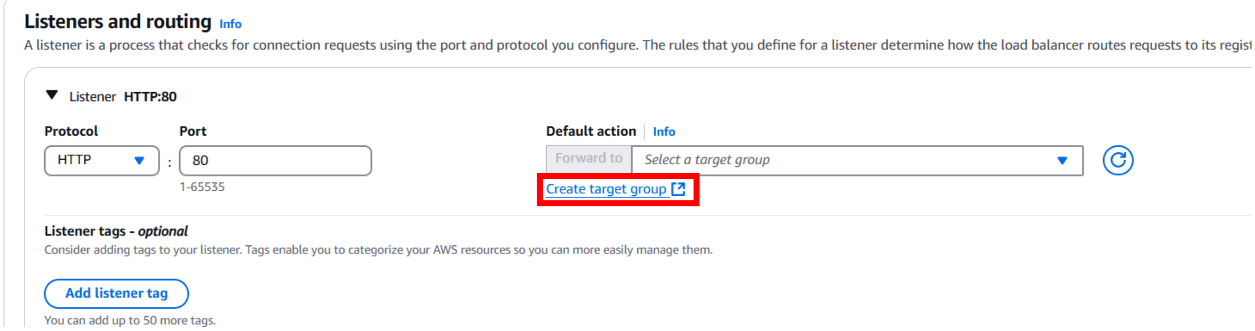
☑ us-west-2b (usw2-az1)

Subnet

subnet-0b02245765fe6c6a1 — AWS-prod-example-subnet-public2-us-west-2b
IPv4 subnet CIDR: 10.0.16.0/20

IPv4 address
Assigned by AWS

50. Create a target group which instances should be accessible. For this, click on target group



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 regist

▼ Listener **HTTP:80**

Protocol: HTTP

Port: 80
1-65535

Default action Info

Forward to — *Select a target group*

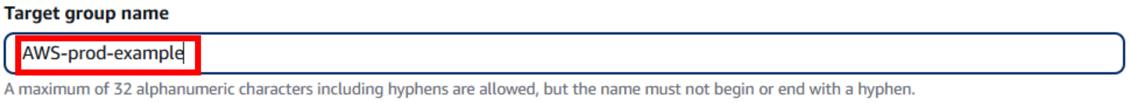
Create target group

Listener tags - *optional*

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

You can add up to 50 more tags.

51. Give a name to target group

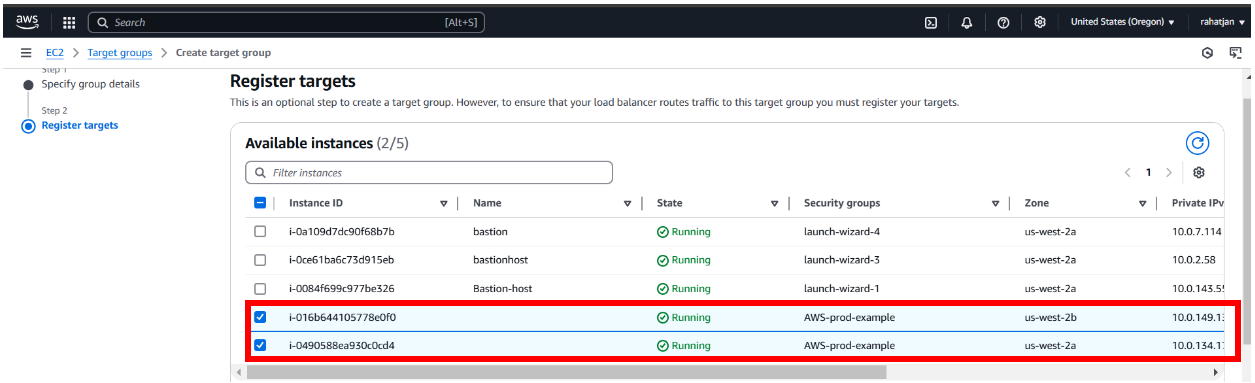


Target group name

AWS-prod-example

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

52. Select the instances of private subnet. One has the application and other doesn't and click on create target group



EC2 > Target groups > Create target group

Step 1
● Specify group details

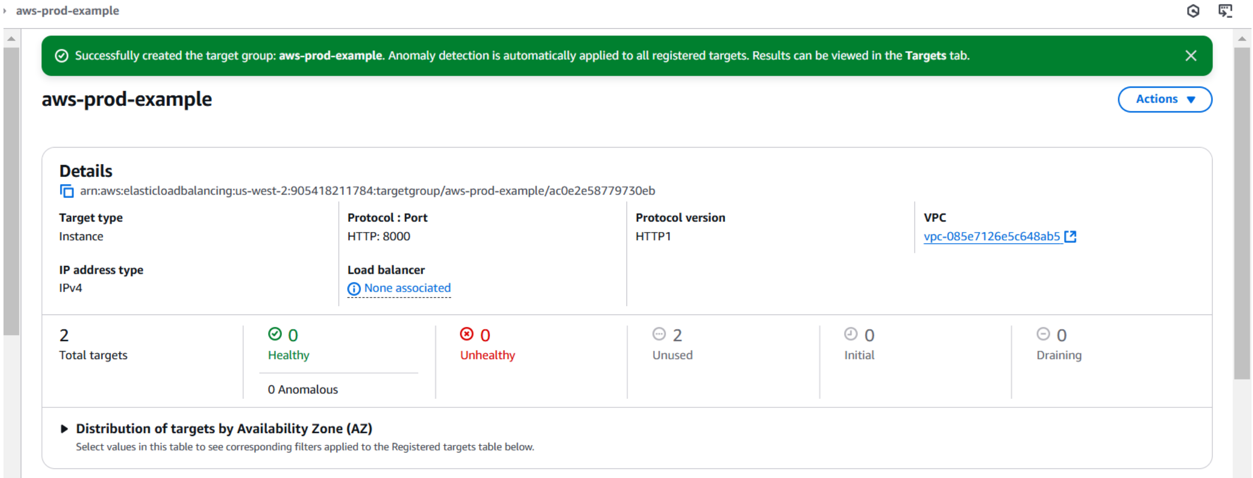
Step 2
◉ Register targets

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 (2/5)

	Instance ID	Name	State	Security groups	Zone	Private IPv
☐	i-0a109d7dc90f68b7b	bastion	✓ Running	launch-wizard-4	us-west-2a	10.0.7.114
☐	i-0ce61ba6c73d915eb	bastionhost	✓ Running	launch-wizard-3	us-west-2a	10.0.2.58
☐	i-0084f699c977be326	Bastion-host	✓ Running	launch-wizard-1	us-west-2a	10.0.143.5
☑	i-016b644105778e0f0		✓ Running	AWS-prod-example	us-west-2b	10.0.149.1
☑	i-0490588ea930c0cd4		✓ Running	AWS-prod-example	us-west-2a	10.0.134.1



aws-prod-example

Successfully created the target group: **aws-prod-example**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

aws-prod-example

Details

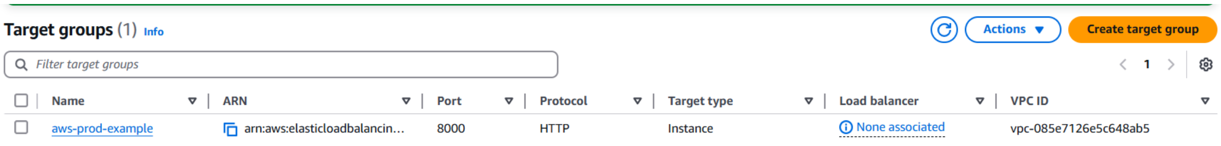
arn:aws:elasticloadbalancing:us-west-2:905418211784:targetgroup/aws-prod-example/ac0e2e58779730eb

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 8000	HTTP1	vpc-085e7126e5c648ab5
IP address type	**Load balancer**		
IPv4	None associated		

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

Distribution of targets by Availability Zone (AZ)

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



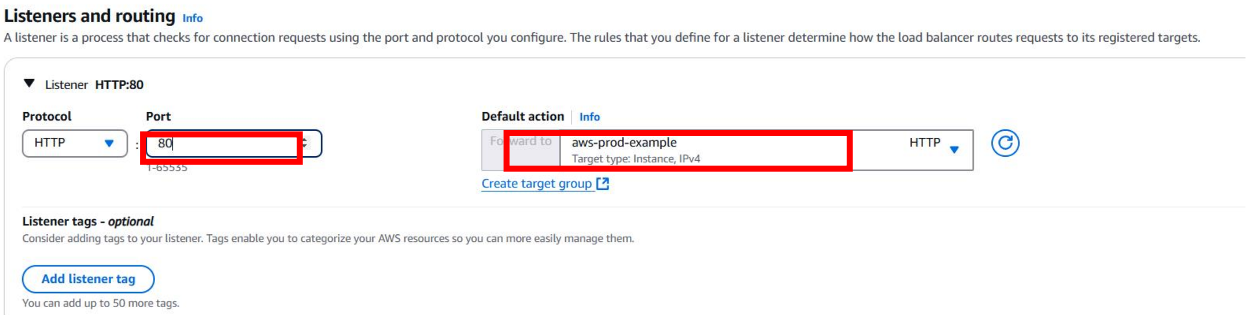
Target groups (1) Info

Filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
aws-prod-example	arn:aws:elasticloadbalancin...	8000	HTTP	Instance	None associated	vpc-085e7126e5c648ab5

Adding target group to the load balancer:

53. Click on create load balancer and add the target group and port 80 which is default



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.

Listener HTTP:80

Protocol: HTTP

Port: 80

1-65535

Default action Info

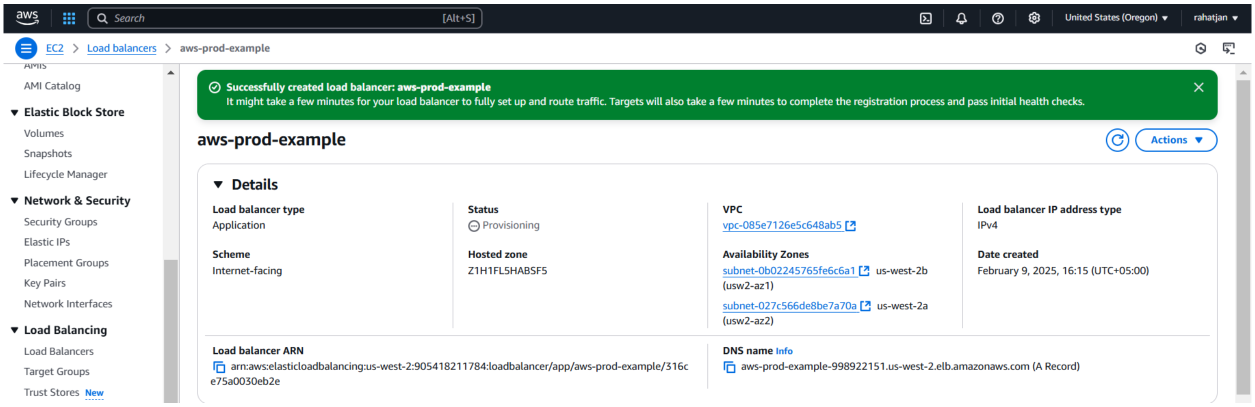
Forward to: aws-prod-example
Target type: Instance, IPv4
HTTP

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.

You can add up to 50 more tags.



Successfully created load balancer: aws-prod-example
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

aws-prod-example

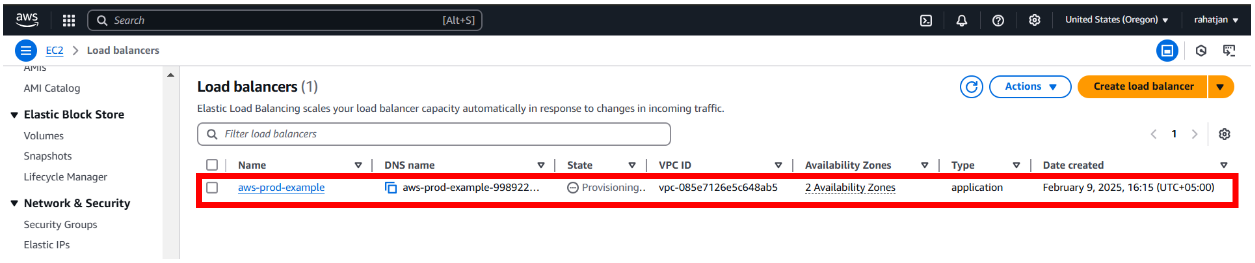
Details

Load balancer type	Status	VPC	Load balancer IP address type
Application	Provisioning	vpc-085e7126e5c648ab5	IPv4
Scheme	**Hosted zone**	**Availability Zones**	**Date created**
Internet-facing	Z1H1FL5HABSF5	subnet-0b02245765fe6c6a1 us-west-2b (usw2-az1) subnet-027c566de8be7a70a us-west-2a (usw2-az2)	February 9, 2025, 16:15 (UTC+05:00)

Load balancer ARN: arn:aws:elasticloadbalancing:us-west-2:905418211784:loadbalancer/app/aws-prod-example/316ce75a0030eb2e

DNS name Info: aws-prod-example-998922151.us-west-2.elb.amazonaws.com (A Record)

Now load balancer with the attached target group has been created successfully



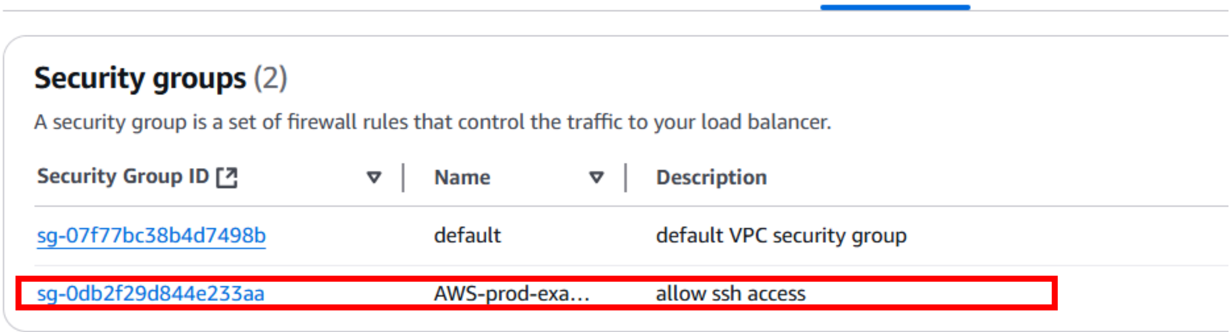
Load balancers (1)

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

□	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
□	aws-prod-example	aws-prod-example-998922…	Provisioning..	vpc-085e7126e5c648ab5	2 Availability Zones	application	February 9, 2025, 16:15 (UTC+05:00)

Accessing the application from outside world:

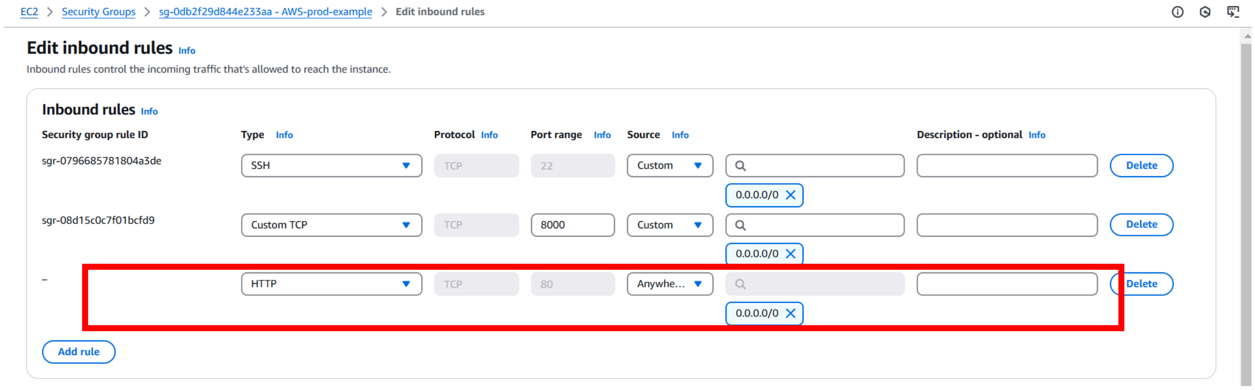
Let's try to access the load balancer and you will see that the load balancer is not accessible because the subnet that you attached to the load balancer does not expose port 80 so we have to explicitly add the rule



Security groups (2)

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

Security Group ID	Name	Description
sg-07f77bc38b4d7498b	default	default VPC security group
sg-0db2f29d844e233aa	AWS-prod-exa…	allow ssh access



EC2 > Security Groups > sg-0db2f29d844e233aa - AWS-prod-example > Edit inbound rules
Add rule

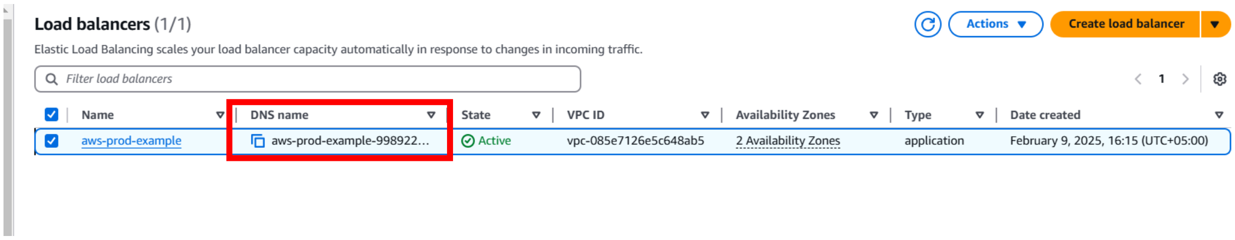
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0796685781804a3de	SSH	TCP	22	Custom	0.0.0.0/0 ✕		Delete
sgr-08d15c0c7f01bcfd9	Custom TCP	TCP	8000	Custom	0.0.0.0/0 ✕		Delete
–	HTTP	TCP	80	Anywhe…	0.0.0.0/0 ✕		Delete

54. Now to access the application, click on DNS name

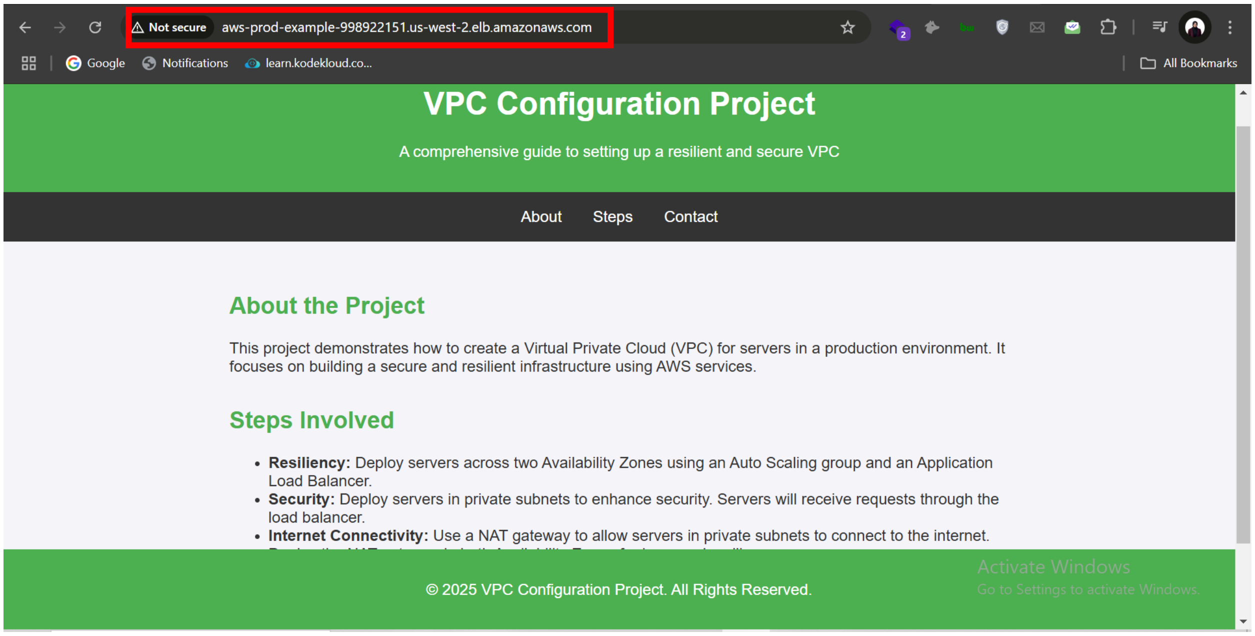


Load balancers (1/1)

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

☑	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
☑	aws-prod-example	aws-prod-example-998922…	Active	vpc-085e7126e5c648ab5	2 Availability Zones	application	February 9, 2025, 16:15 (UTC+05:00)

Now our application is accessible successfully



VPC Configuration Project

A comprehensive guide to setting up a resilient and secure VPC

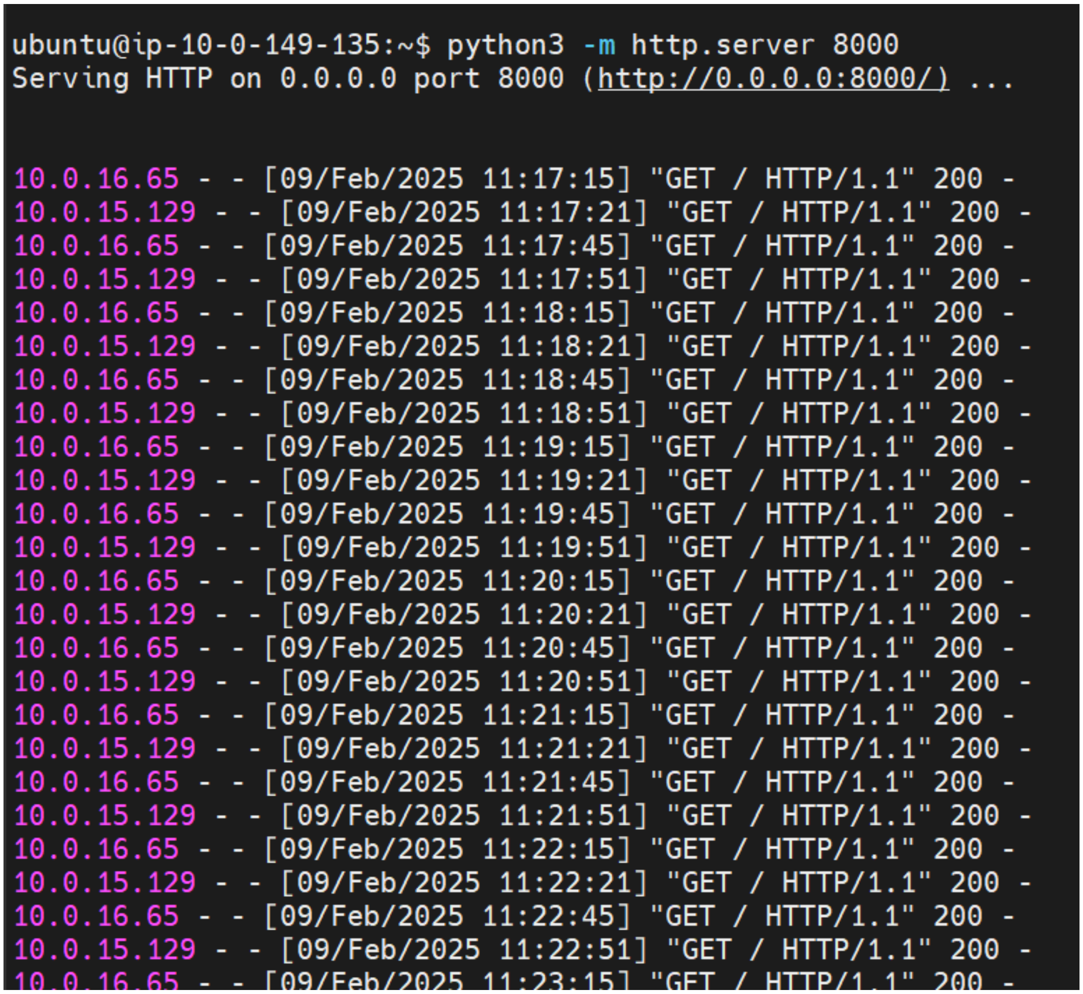
About the Project

This project demonstrates how to create a Virtual Private Cloud (VPC) for servers in a production environment. It focuses on building a secure and resilient infrastructure using AWS services.

Steps Involved

- **Resiliency:** Deploy servers across two Availability Zones using an Auto Scaling group and an Application Load Balancer.
- **Security:** Deploy servers in private subnets to enhance security. Servers will receive requests through the load balancer.
- **Internet Connectivity:** Use a NAT gateway to allow servers in private subnets to connect to the internet.

© 2025 VPC Configuration Project. All Rights Reserved.



```
ubuntu@ip-10-0-149-135:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...


10.0.16.65 - - [09/Feb/2025 11:17:15] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:17:21] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:17:45] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:17:51] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:18:15] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:18:21] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:18:45] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:18:51] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:19:15] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:19:21] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:19:45] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:19:51] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:20:15] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:20:21] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:20:45] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:20:51] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:21:15] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:21:21] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:21:45] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:21:51] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:22:15] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:22:21] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:22:45] "GET / HTTP/1.1" 200 -
10.0.15.129 - - [09/Feb/2025 11:22:51] "GET / HTTP/1.1" 200 -
10.0.16.65 - - [09/Feb/2025 11:23:15] "GET / HTTP/1.1" 200 -
```