Three-Tier Architecture

Building a Three-Tier Architecture on AWS with Deploying Application

Achieving High Scalability, High Availability, and Fault Tolerance

➢ **Prerequisites**

- AWS Account
- Basic knowledge of Linux

➢ **List of AWS services**

- Amazon Route 53
- Amazon EC2
- Amazon Auto scaling
- Amazon Certificate Manager
- Amazon RDS
- Amazon VPC
- Amazon Cloud Watch

Plan of Execution

- What is three-tier architecture
- The architecture of the project
- A step-by-step guide with screenshots
- Testing
- Resource cleanup

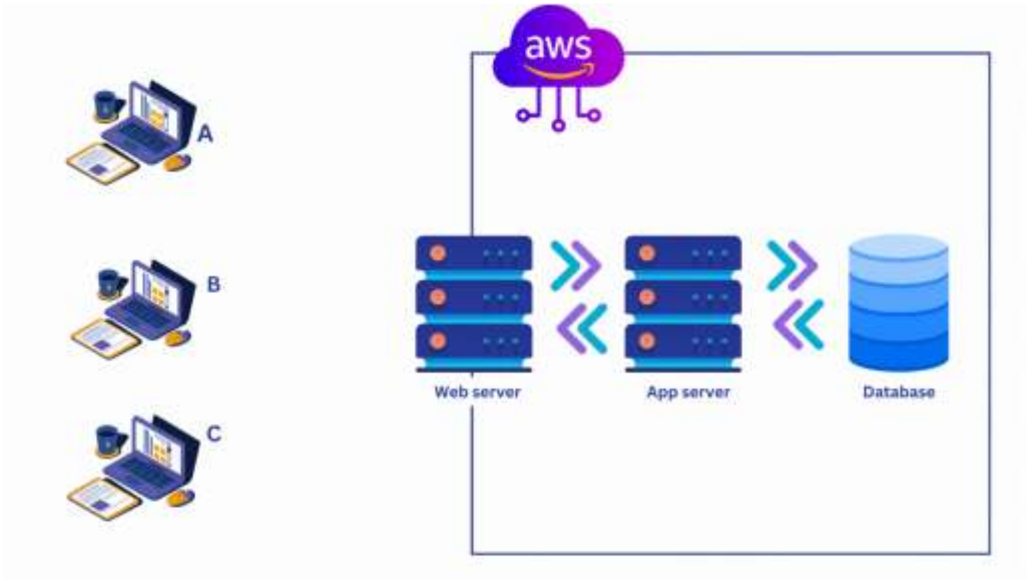
What is Three-tier architecture?

Three-tier architecture is a software architecture pattern that separates an application into three layers.

Presentation layer ➡ handles user interaction

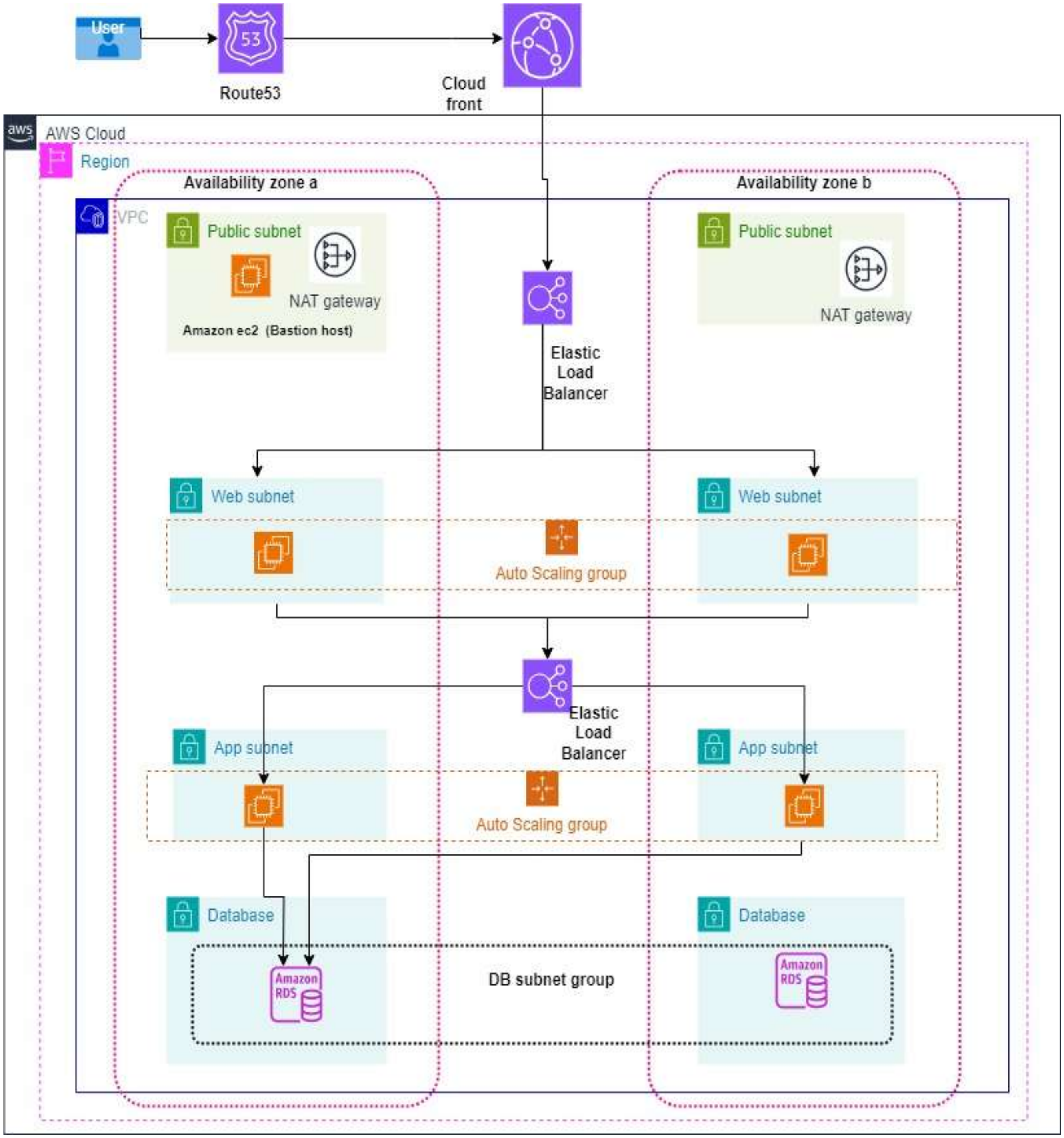
Application layer (backend logic) ➡ processes business logic and data processing

Data layer (database) ➡ manages data storage and retrieval



Each layer has distinct responsibilities, allowing for modularity, scalability, and maintainability. This architecture promotes the separation of concerns and facilitates easy updates or modifications to specific layers without impacting others.

Architecture of the Project



Architecture

The image describes three-tier architecture on AWS, which is commonly used to build highly available and scalable web applications.

1. Route 53 (DNS Service)

- **User Requests**: A user initiates a request, which is routed through AWS Route 53, a DNS service that translates domain names to IP addresses.
- **Load Balancing**: Route 53 directs the traffic to an Elastic Load Balancer (ELB) in the appropriate region.

2. CloudFront Distribution:

- CloudFront will cache the static content (images, CSS, JavaScript, etc.) and provide faster access to users from the edge locations around the world.
- CloudFront receives the user's request before passing it to the ELB.
- The CloudFront distribution is connected to the ELB for dynamic content delivery from the web and app servers.

3. Elastic Load Balancer (ELB)

- **Traffic Distribution**: The ELB distributes incoming application traffic across multiple targets, such as EC2 instances in different Availability Zones (AZs), ensuring high availability and fault tolerance.
- **Scaling**: ELBs also work with Auto Scaling groups to dynamically adjust the number of EC2 instances based on traffic demand.

4. VPC (Virtual Private Cloud)

- **Network Segmentation**: The architecture is hosted within a VPC, providing network isolation and segmentation. The VPC spans multiple Availability Zones for high availability.

5. Public Subnets

- **NAT Gateway and Bastion Host**: Each Availability Zone has a public subnet containing a NAT Gateway and possibly a Bastion Host (Amazon EC2 instance). The NAT Gateway allows instances in private subnets to connect to the internet for updates or other external communications without exposing them to inbound internet traffic.

6. Web Tier (Web Subnets)

- **Web Servers**: The web subnet in each Availability Zone hosts EC2 instances (web servers) running the front-end application. These instances are part of an Auto Scaling group that adjusts the number of servers based on demand.
- **ELB Connection**: The Elastic Load Balancer routes incoming traffic to these web servers.

7. Application Tier (App Subnets)

- **App Servers**: The app subnets contain EC2 instances that handle business logic and processing. Like the web tier, these instances are part of an Auto Scaling group for scalability.
- **Internal Load Balancing**: A second ELB could be used here to balance traffic among app servers.

8. Private Subnets

- **Database Tier (DB Subnet Group)**: The private subnet houses Amazon RDS instances (Relational Database Service) within a DB Subnet Group. This ensures the database is only accessible from within the VPC, enhancing security.
- **Network Isolation**: Since this subnet is private, it does not have direct internet access, further protecting sensitive data.

9. Security

- **Subnets and Security Groups**: Each tier (web, app, and database) is in separate subnets with different security groups, ensuring proper access control. Web servers might only accept traffic from the internet, app servers from the web servers, and the database only from the app servers.

Workflow Overview

1. **User Request:** A user sends a request, which is routed through **Route 53**.
2. **CloudFront Caching:** CloudFront checks if the requested content is cached at an edge location.
   - If cached, the content is served directly from the edge location.
   - If not cached, CloudFront forwards the request to the **Elastic Load Balancer**.
3. **Load Balancing:** The ELB distributes the request to an available **web server** in the web subnet.

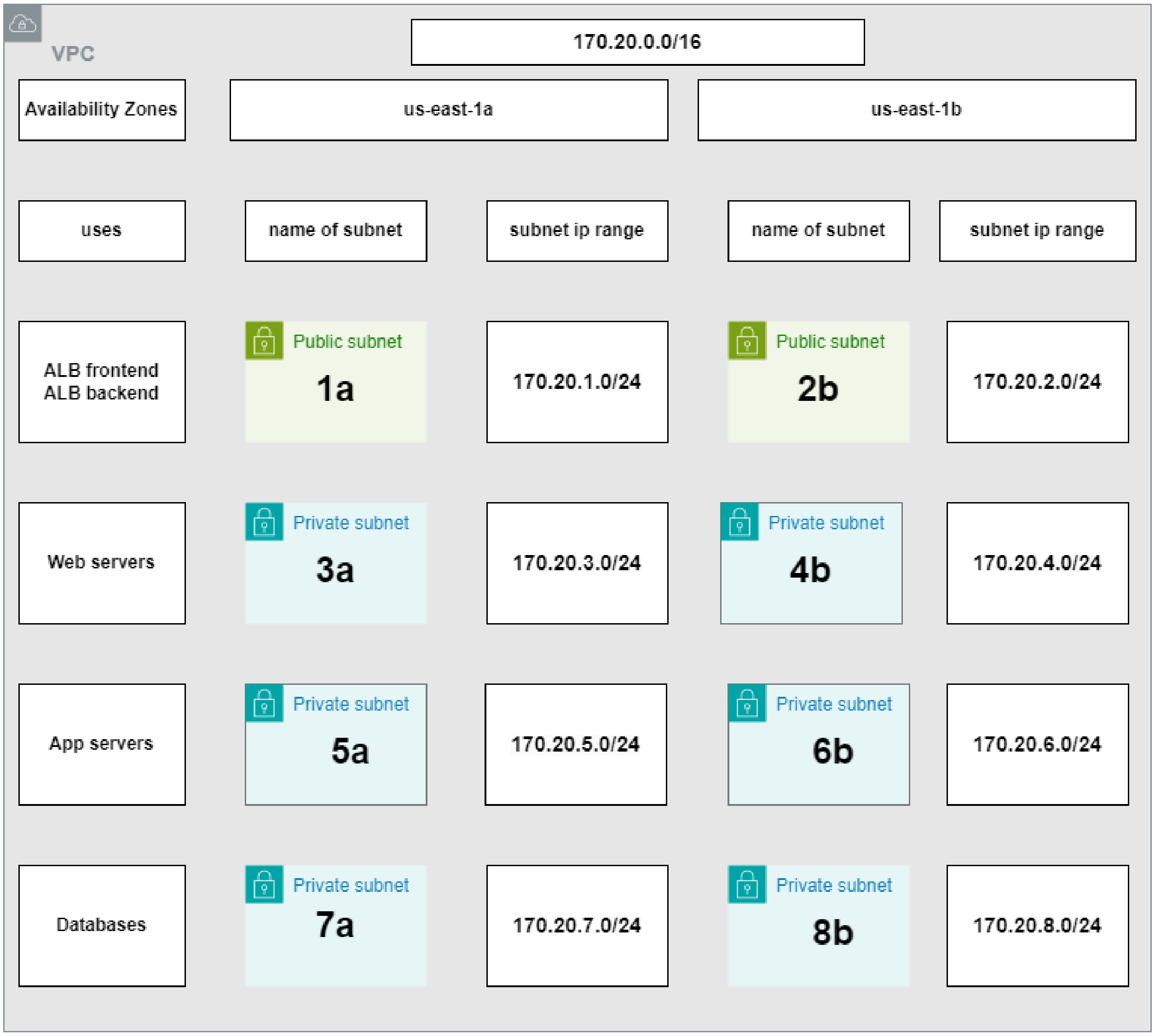
4. **Web Server Processing:** The web server processes the request or forwards it to the app server.

5. **App Server Logic:** The app server performs business logic, querying the **Amazon RDS database** if needed.

6. **Response Delivery:** The data flows back from the app server to the web server, through CloudFront, and then to the user.

Creation of VPC, subnets, Rouetables, IGW, Natgateway

Setting up VPC in us-east-1 region to isolate our resources from the internet. The below image contained all the subnets, their IP range, and their uses.



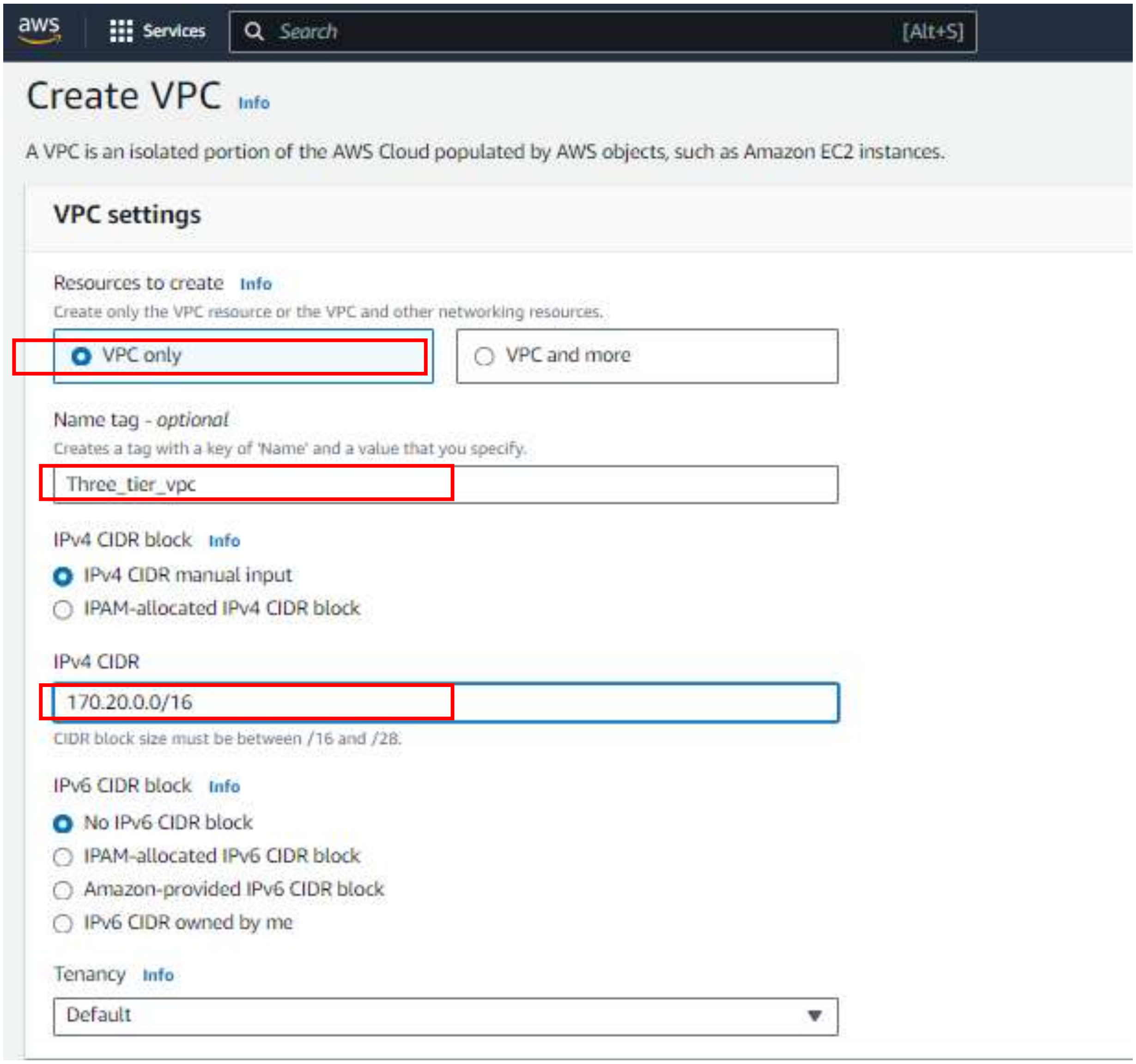
VPC

170.20.0.0/16

Availability Zones	us-east-1a		us-east-1b	
uses	name of subnet	subnet ip range	name of subnet	subnet ip range
ALB frontend ALB backend	Public subnet 1a	170.20.1.0/24	Public subnet 2b	170.20.2.0/24
Web servers	Private subnet 3a	170.20.3.0/24	Private subnet 4b	170.20.4.0/24
App servers	Private subnet 5a	170.20.5.0/24	Private subnet 6b	170.20.6.0/24
Databases	Private subnet 7a	170.20.7.0/24	Private subnet 8b	170.20.8.0/24

VPC:

1. Please log in to your AWS Account and type VPC in the AWS console. And click on VPC service.
2. Click on Your VPC's button on the left and then click on Create VPC the button on the top right corner of the page
3. Here we can see the form where we can fill the configuration of VPC. Please enter the name that you want to keep and the IPV4 CIDR block. in my case CIDE block is 170.20.0.0/16.



Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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 - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

Three_tier_vpc

IPv4 CIDR block Info

● IPv4 CIDR manual input

○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

170.20.0.0/16

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

IPv6 CIDR block Info

● No IPv6 CIDR block

○ IPAM-allocated IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

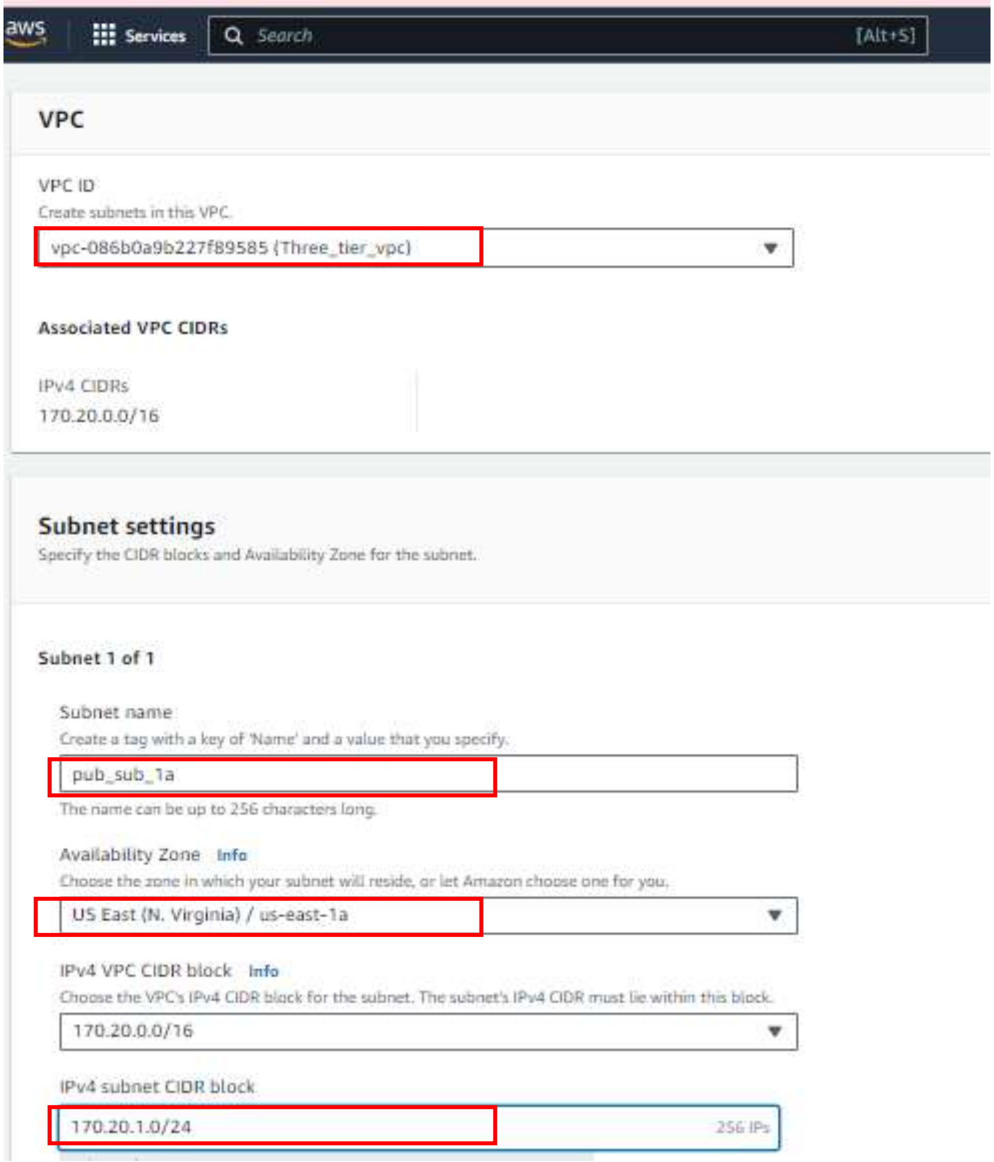
○ IPv6 CIDR owned by me

Tenancy Info

Default ▼

Subnets:

1. Now click on the subnet button which is located on the left side and then click on the Create subnet button on the top right corner of the page.
2. Please remove the default VPC ID and choose the VPC ID that we have just created in the VPC ID field. And click on the Add Subnet button at the bottom.
3. Now we need to configure our subnets. We are going to create a total of 8 subnets of which 2 of them are public and the rest of 6 subnets are private. After adding all the subnets click on Create subnet button.



VPC

VPC ID
Create subnets in this VPC.
vpc-086b0a9b227f89585 (Three_tier_vpc)

Associated VPC CIDRs

IPv4 CIDRs
170.20.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

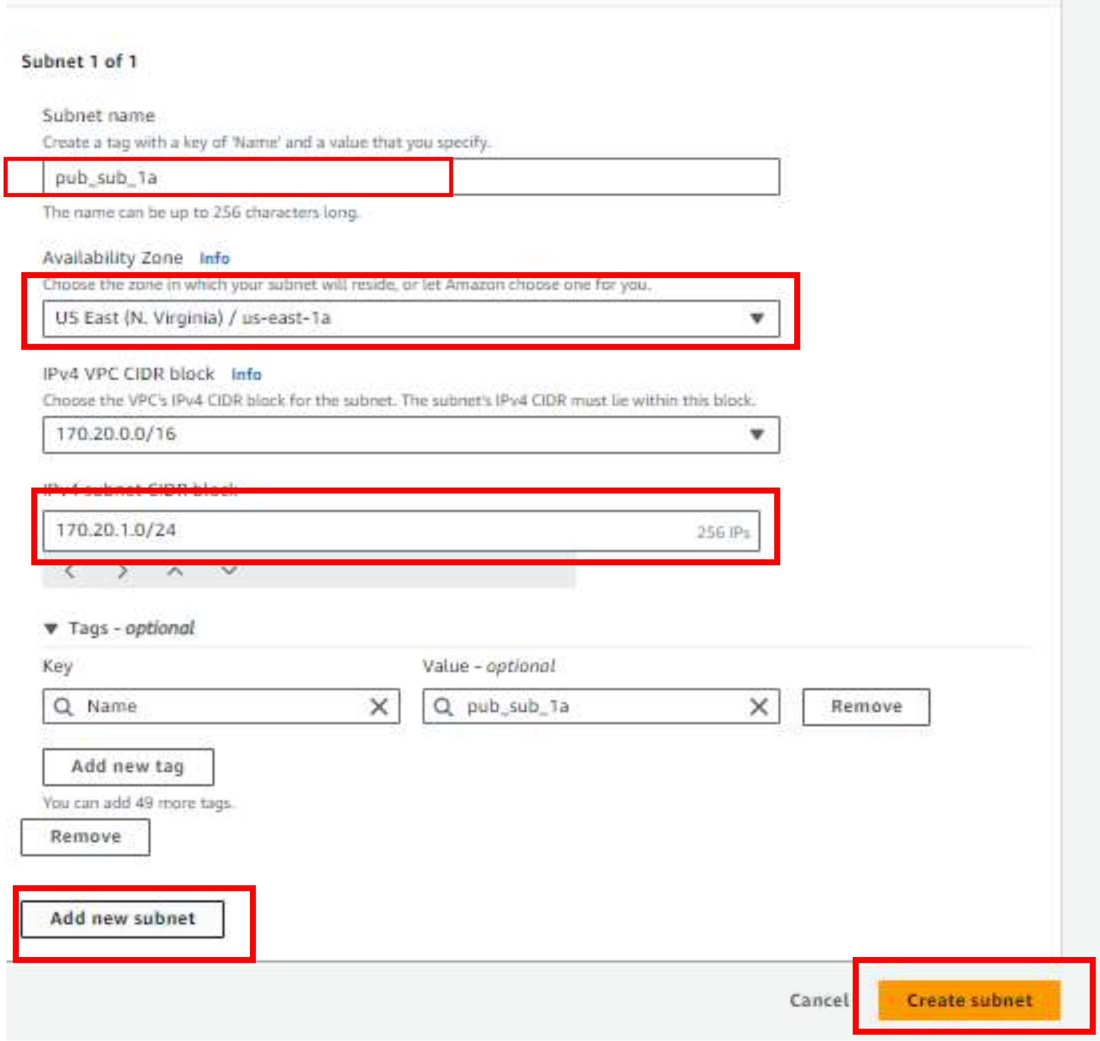
Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pub_sub_1a
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
170.20.0.0/16

IPv4 subnet CIDR block
170.20.1.0/24 256 IPs



Subnet 1 of 1

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

pub_sub_1a

The name can be up to 256 characters long.

Availability Zone Info

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

170.20.0.0/16

IPv4 subnet CIDR block

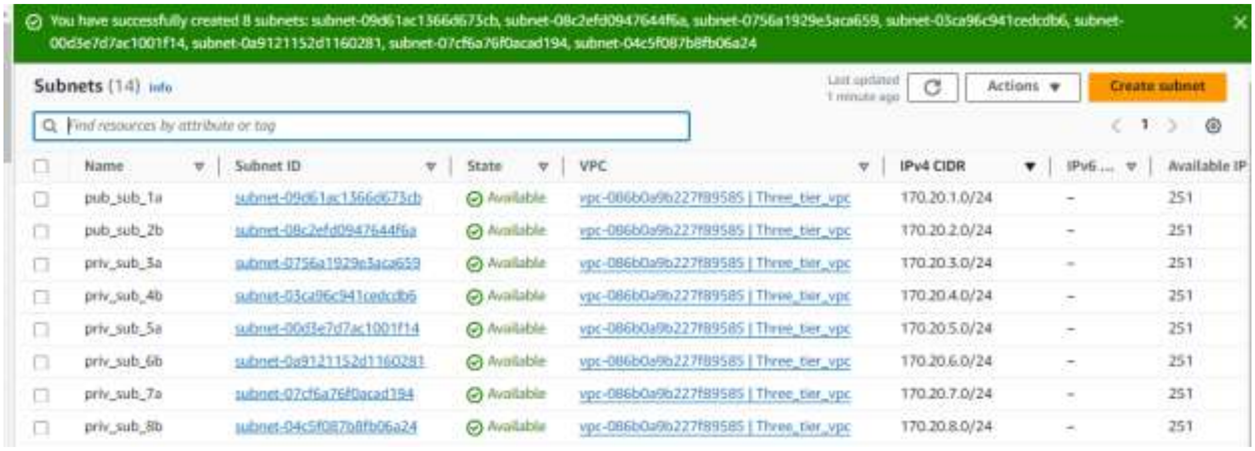
170.20.1.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*
Name	pub_sub_1a

You can add 49 more tags.

After the successful creation of all 8 subnets, they look like this. you can verify with my subnets.



You have successfully created 8 subnets: subnet-09d61ac1366d673cb, subnet-08c2efd0947644f6a, subnet-0756a1929e3aca659, subnet-03ca96c941cedcdb6, subnet-00d3e7d7ac1001f14, subnet-0a9121152d1160281, subnet-07cf6a76f0acad194, subnet-04c5f087b8fb06a24

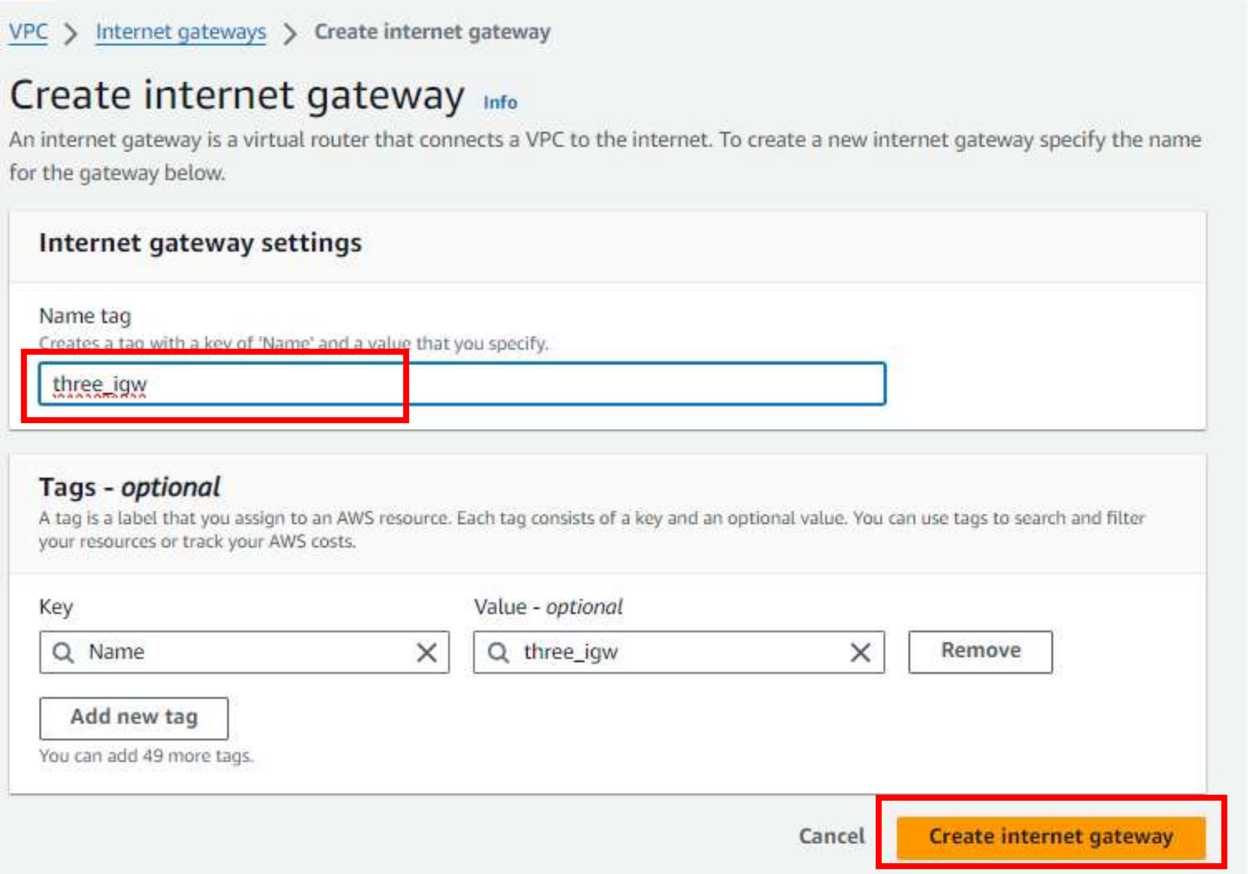
Subnets (14) Info

□	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 ...	Available IP
□	pub_sub_1a	subnet-09d61ac1366d673cb	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.1.0/24	–	251
□	pub_sub_2b	subnet-08c2efd0947644f6a	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.2.0/24	–	251
□	priv_sub_3a	subnet-0756a1929e3aca659	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.3.0/24	–	251
□	priv_sub_4b	subnet-03ca96c941cedcdb6	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.4.0/24	–	251
□	priv_sub_5a	subnet-00d3e7d7ac1001f14	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.5.0/24	–	251
□	priv_sub_6b	subnet-0a9121152d1160281	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.6.0/24	–	251
□	priv_sub_7a	subnet-07cf6a76f0acad194	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.7.0/24	–	251
□	priv_sub_8b	subnet-04c5f087b8fb06a24	✓ Available	vpc-086b0a9b227f89585 | Three_tier_vpc	170.20.8.0/24	–	251

Internet Gateway:

Now we are going to create Internet Gateway also known as **IGW.** It is responsible for communication between VPC, VPC's public subnet with the Internet. Without IGW we won't be able to communicate with the Internet. Click on the internet gateways button at the left panel. and then click on the Create Internet gateways button on the top right corner of the page.

Give any name to IGW. And click on Create Internet gateway button.



VPC > Internet gateways > Create internet gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag

Creates a tag with a key of 'Name' and a value that you specify.

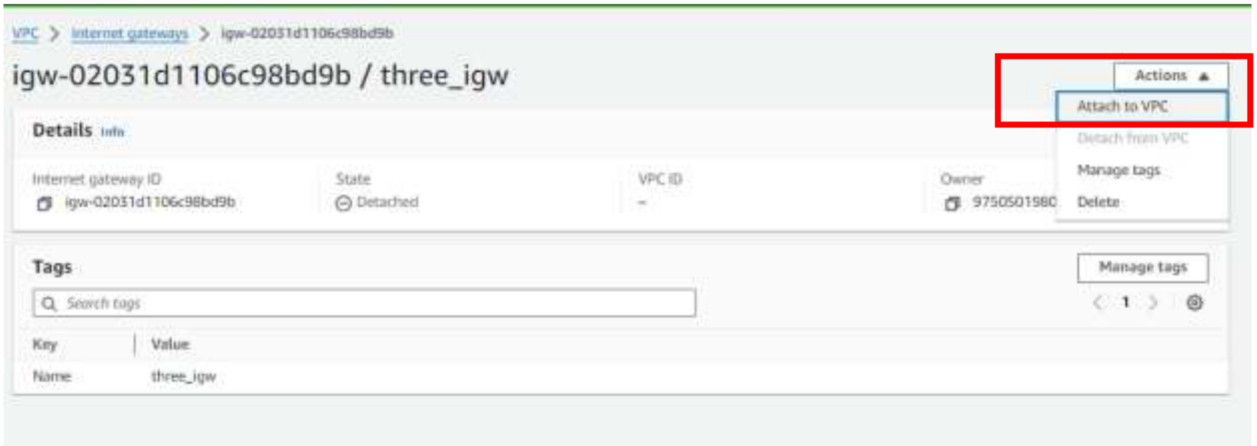
three_igw

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	three_igw

After creating an internet gateway we need to attach it with VPC to use it. For that click on the Action button. Here you can see the drop-down list. Please select the option Attach to VPC.



VPC > Internet gateways > igw-02031d1106c98bd9b

igw-02031d1106c98bd9b / three_igw

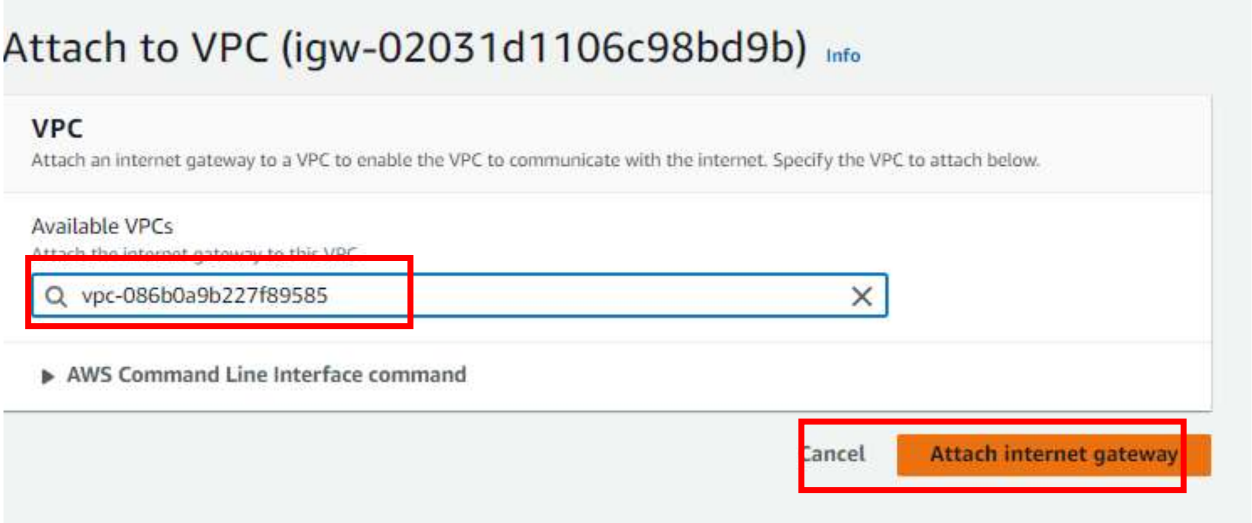
Details Info

Internet gateway ID	State	VPC ID	Owner
igw-02031d1106c98bd9b	Detached	-	9750501980

Tags

Key	Value
Name	three_igw

Please select VPC that we have created just now from the Available VPC list. And then click on the Attach Internet gateway button.



Attach to VPC (igw-02031d1106c98bd9b) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

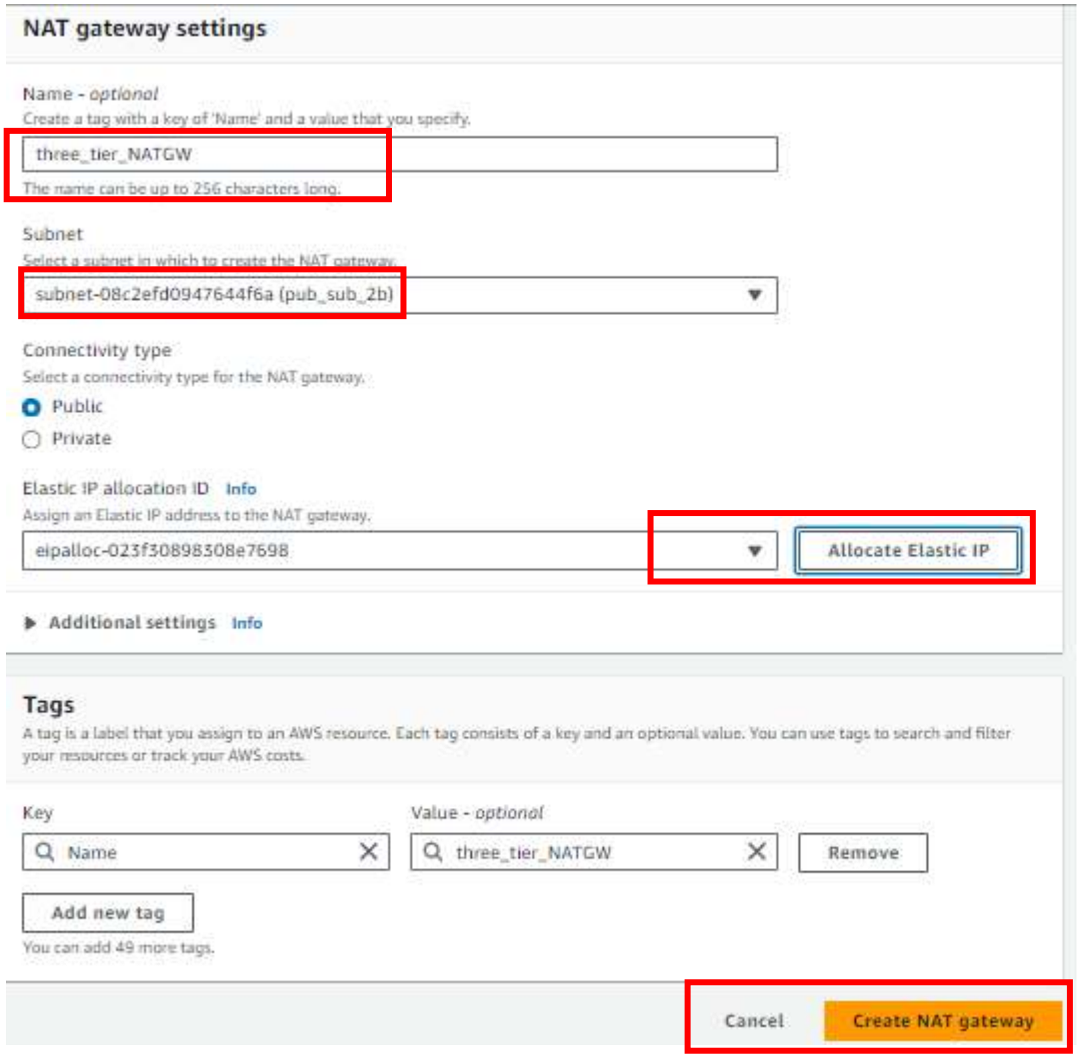
Attach the internet gateway to this VPC

vpc-086b0a9b227f89585

NAT gateway

NAT gateway is responsible to connect resources that are in the private subnet to communicate with the internet. All the resources which will be there in a private subnet will communicate to the internet through the NAT gateway. We will keep the NAT gateway in the public subnet so that it can access the internet. NAT gateway is a chargeable resource. So you will be charged by AWS as long as you keep it up. Now to create a NAT gateway click on the NAT gateways button on the left panel of the web page. and then click on the Create NAT gateways button in the top right corner of the page.

Give any name to the NAT gateway. But be cautious with selecting a subnet. ***You have to select one of the Public subnets among the two. either pub-sub-1a or pub-sub-2b***. Then click on the Allocate Elastic IP button to allocate Elastic IP. And then click on the Create NAT gateways button. NAT gateways creation takes 2-4 minutes.



NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
three_tier_NATGW
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-08c2efd0947644f6a (pub_sub_2b)

Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-023f30898308e7698

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

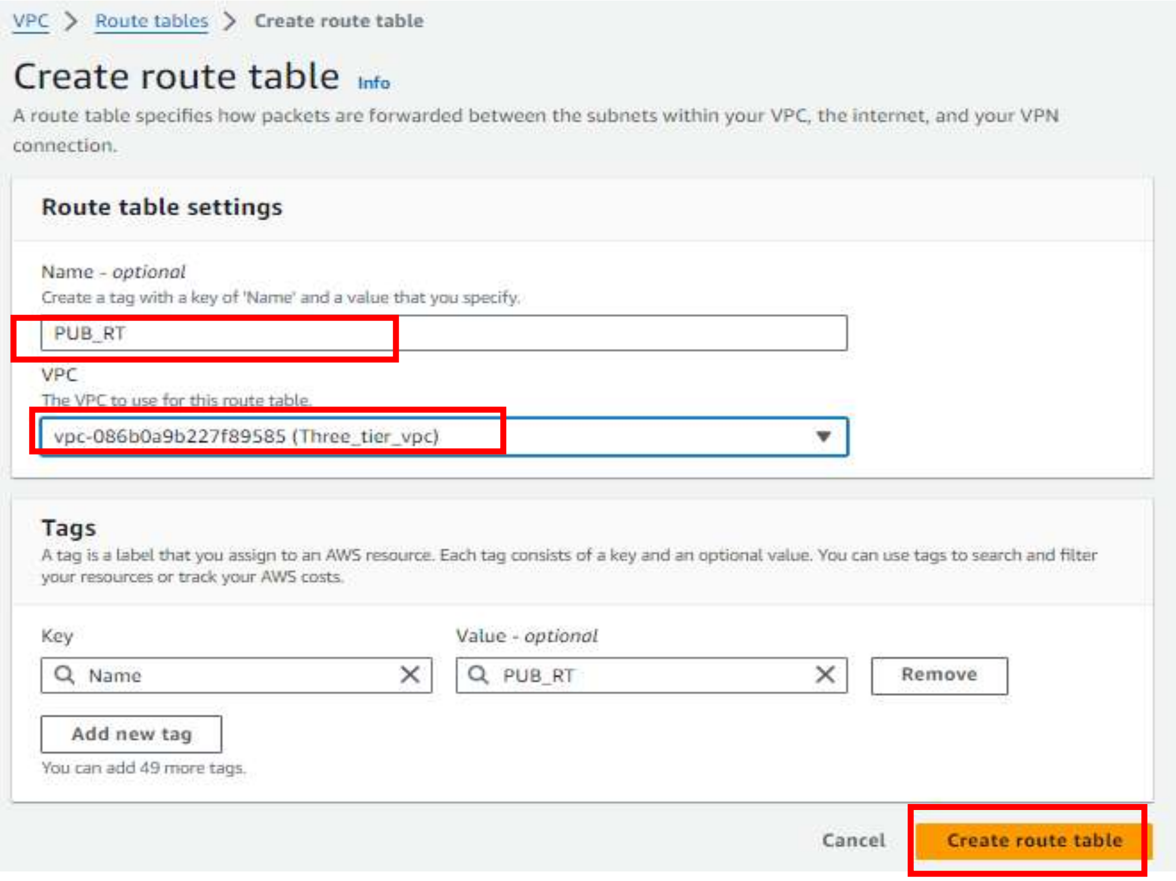
Key	Value - *optional*
Name	three_tier_NATGW

You can add 49 more tags.

Route table:

A Route Table to handle traffic for public subnet and private subnet and for that, we need to create a Route table. We are going to create two route tables one for the public subnet and another one for the private subnet. First, we are going to create RT for the public subnet. So click on the Route table button which you can see on the left panel. and click on the Create Route table button on the top corner of the page.

Give a name to your RT such as Pub-RT. Please give a name that is appropriate for resources then it will be easy to organize the things. Make sure you select the correct VPC. And then click on the create route table.



Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
PUB_RT

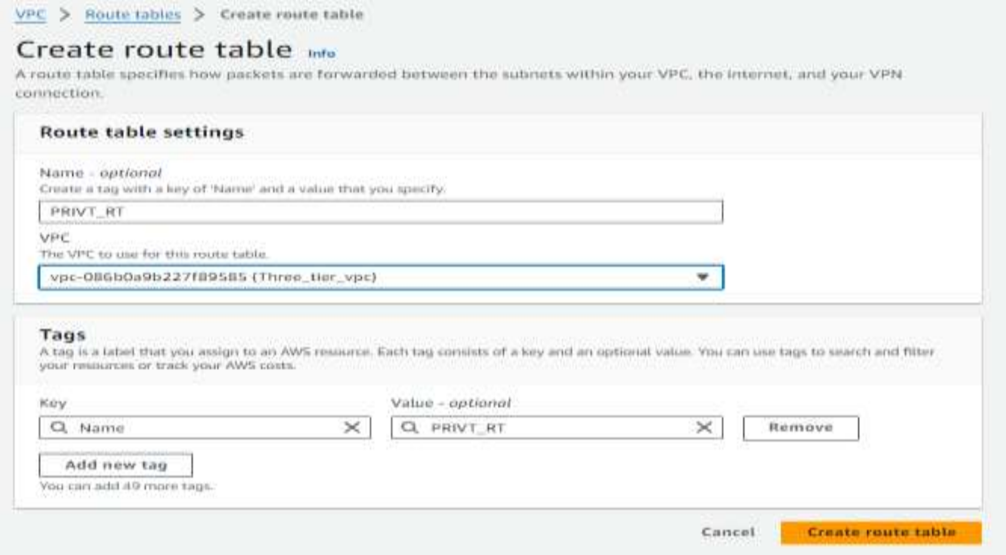
VPC
The VPC to use for this route table.
vpc-086b0a9b227f89585 (Three_tier_vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	PUB_RT

You can add 49 more tags.

Create RT for the private subnet.



Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
PRIVT_RT

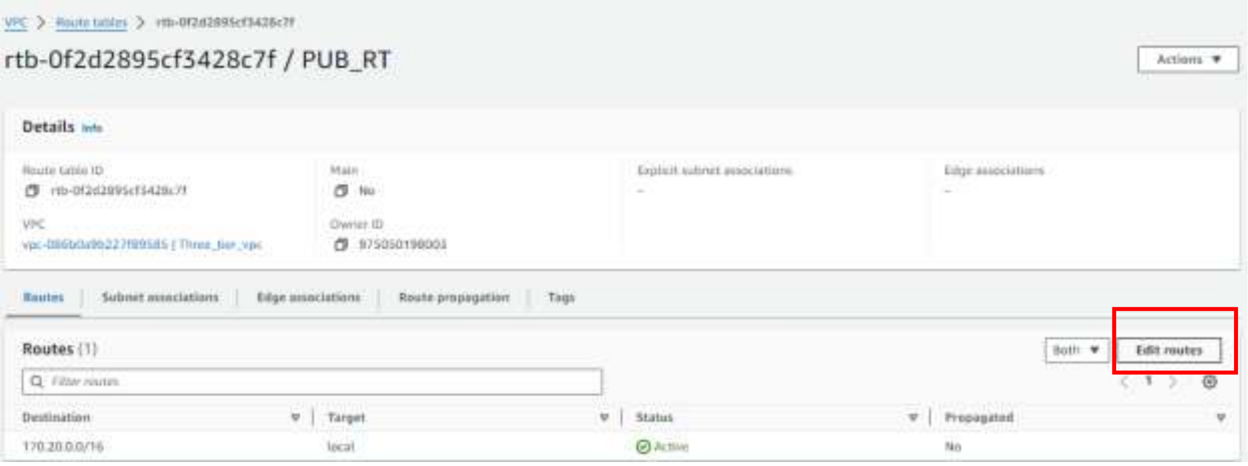
VPC
The VPC to use for this route table.
vpc-086b0a9b227f89585 (Three_tier_vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	PRIVT_RT

You can add 49 more tags.

Now, we need to do some association with both RTs so select **Pub-RT** and click on the Routes tab at the bottom and then click on the edit route button.



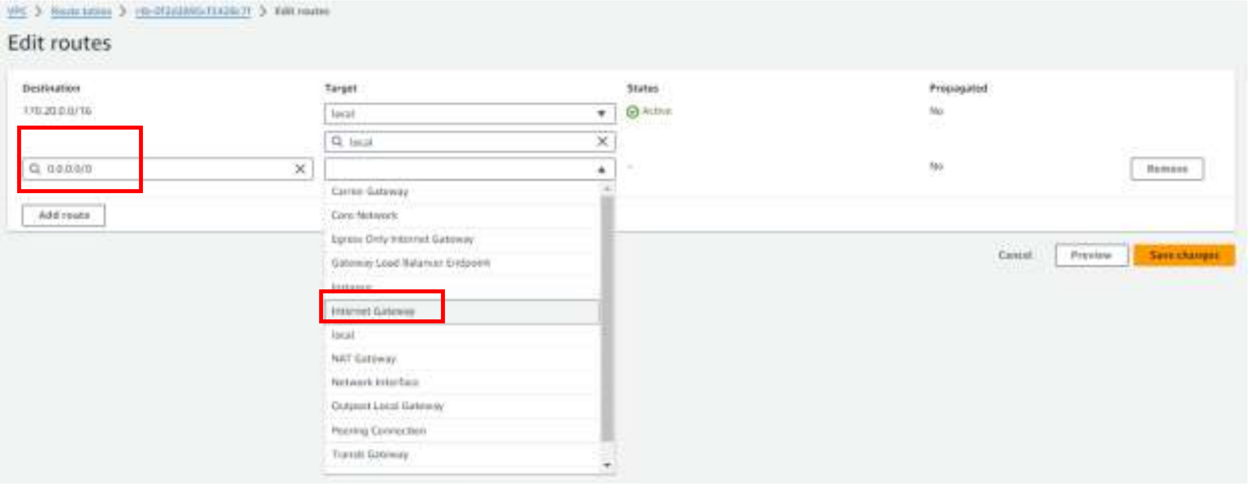
rtb-0f2d2895cf3428c7f / PUB_RT

Route table ID: rtb-0f2d2895cf3428c7f
Main: No
Explicit subnet associations: –
Edge associations: –
VPC: vpc-086b0a9b227f89585 | Three_tier_vpc
Owner ID: 975050198003

Routes (1)

Destination	Target	Status	Propagated
170.20.0.0/16	local	Active	No

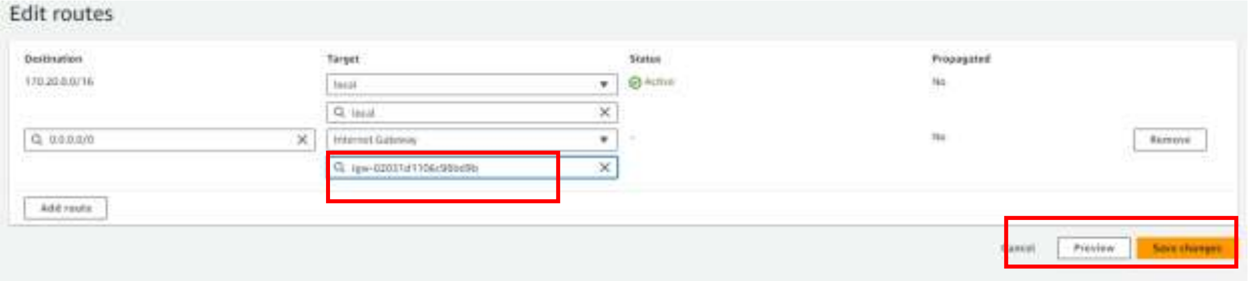
Click on the Add Route button. And select 0.0.0.0/0 in the destination field. And then click on the Target field. As soon as you click on the Target field one drop-down will open and here you have to select Internet gateway, shown in the below image.



Edit routes

Destination	Target	Status	Propagated
170.20.0.0/16	local	Active	No
0.0.0.0/0		-	No

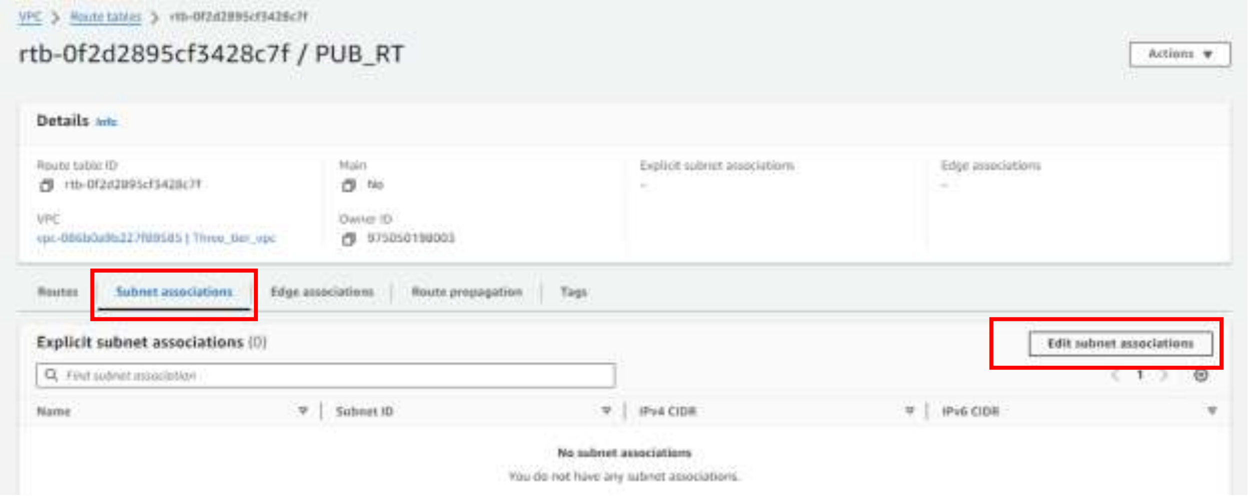
Here you can see the IGW that we created earlier. Select that IGW and click the save changes button.



Edit routes

Destination	Target	Status	Propagated	
170.20.0.0/16	local local	Active	No	
0.0.0.0/0	Internet Gateway igw-02031d1106c98bd9b	-	No	Remove

Keep Pub-RT selected and click on the Subnet associations tab next to the Routes tab. and then click on the Edit subnet associations.



VPC > Route tables > rtb-0f2d2895cf3428c7f

rtb-0f2d2895cf3428c7f / PUB_RT

Details

Route table ID: rtb-0f2d2895cf3428c7f

Main: No

Explicit subnet associations: –

Edge associations: –

VPC: vpc-086b0a9b227f89585 | Three_tier_vpc

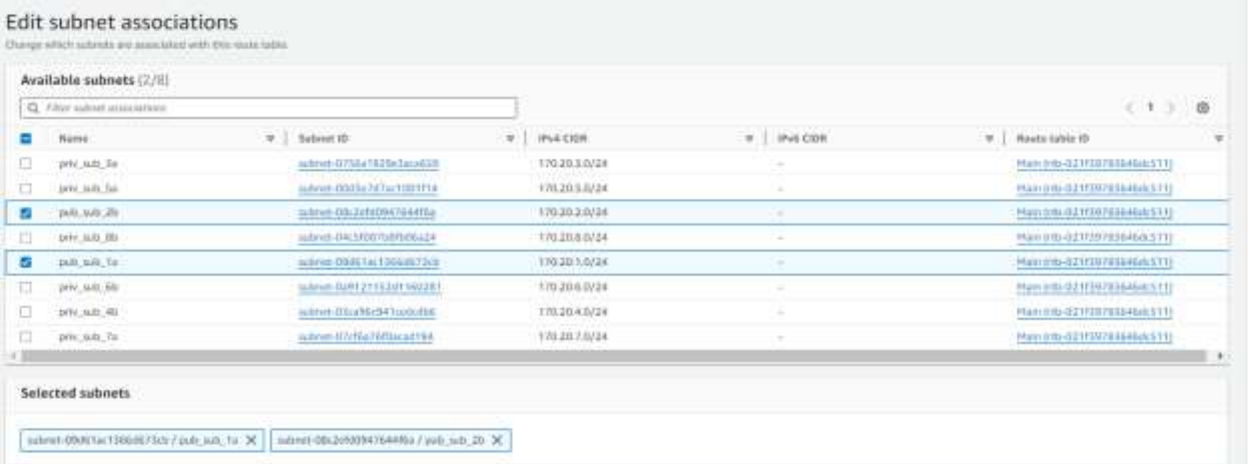
Owner ID: 975050198003

Explicit subnet associations (0)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR

No subnet associations
You do not have any subnet associations.

Now select both public subnets. **pub-sub-1a** and **pub-sub-2b** and click on the save associations button.



Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/8)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
□	priv_sub_3a	subnet-0758a1820e3aca638	170.20.3.0/24	–	Main (rtb-021f39783b46dc511)
□	priv_sub_5a	subnet-00d3e7d7ac1001f14	170.20.5.0/24	–	Main (rtb-021f39783b46dc511)
☑	pub_sub_2b	subnet-0b2efd0947644f6a	170.20.2.0/24	–	Main (rtb-021f39783b46dc511)
□	priv_sub_8b	subnet-04c5f0070d8fb06a24	170.20.8.0/24	–	Main (rtb-021f39783b46dc511)
☑	pub_sub_1a	subnet-09d61ac1366d673cb	170.20.1.0/24	–	Main (rtb-021f39783b46dc511)
□	priv_sub_6b	subnet-0a9127152d1992281	170.20.6.0/24	–	Main (rtb-021f39783b46dc511)
□	priv_sub_4b	subnet-03ca96c941ca0cdb6	170.20.4.0/24	–	Main (rtb-021f39783b46dc511)
□	priv_sub_7a	subnet-07cf6a76f8acad194	170.20.7.0/24	–	Main (rtb-021f39783b46dc511)

Selected subnets

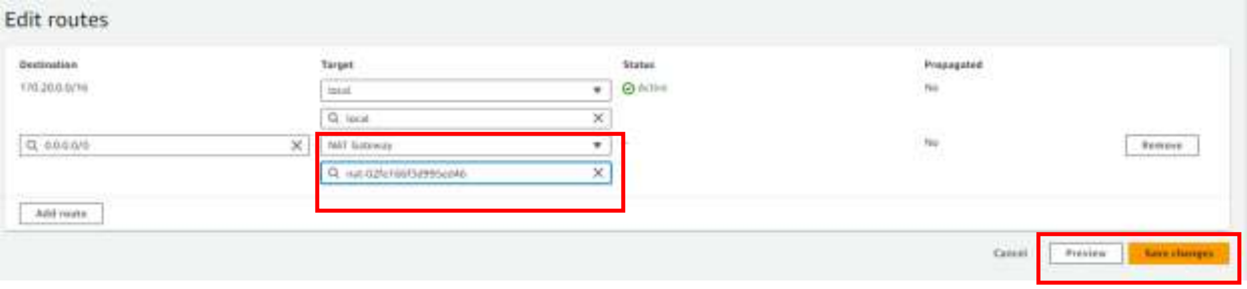
subnet-09d61ac1366d673cb / pub_sub_1a ✕
subnet-0b2efd0947644f6a / pub_sub_2b ✕

Now please select Pri-RT and click on the Routes tab at the bottom of the page.

Here please select 0.0.0.0/0 in the destination field and click on the target. As soon as you click on the target you will see the drop-down list.

Please select NAT gateway from the drop-down list. As shown in the below image.

Select the NAT gateway that we have just created. And click on the save changes button.

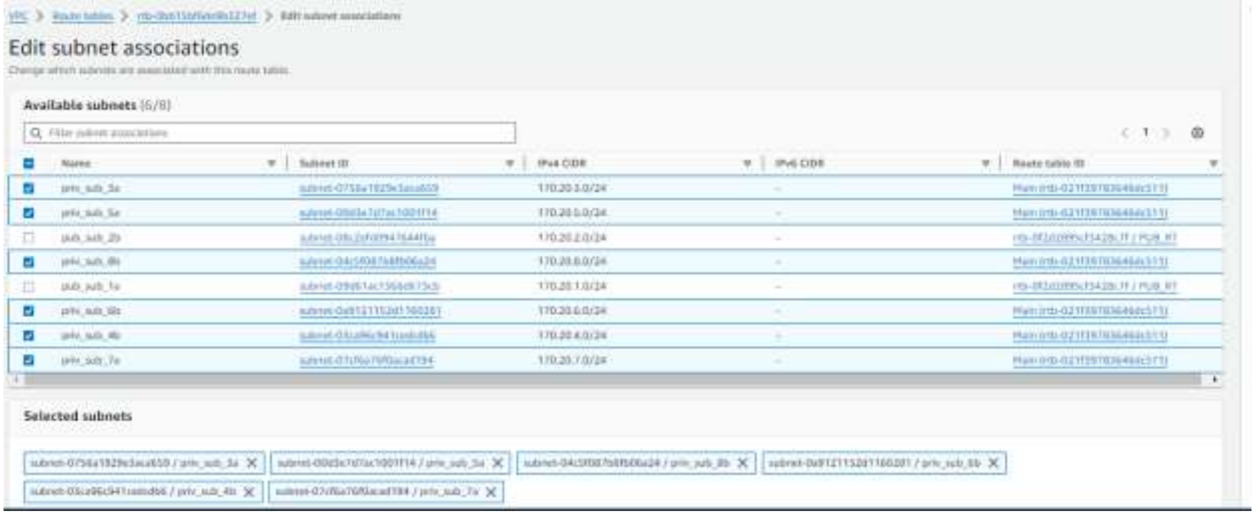


Edit routes

Destination	Target	Status	Propagated	
170.20.0.0/16	local local	Active	No	
0.0.0.0/0	NAT Gateway nat-02fe166f3d995cd46	-	No	Remove

Keep Pri-RT selected and click on the subnet associations tab at the bottom next to the Routes tab. And then click on the Edit route association’s button.

Select all the 6 private subnets. And then click on the save association button.



VPC > Route tables > rtb-0bb15bf6eb8b127ef > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

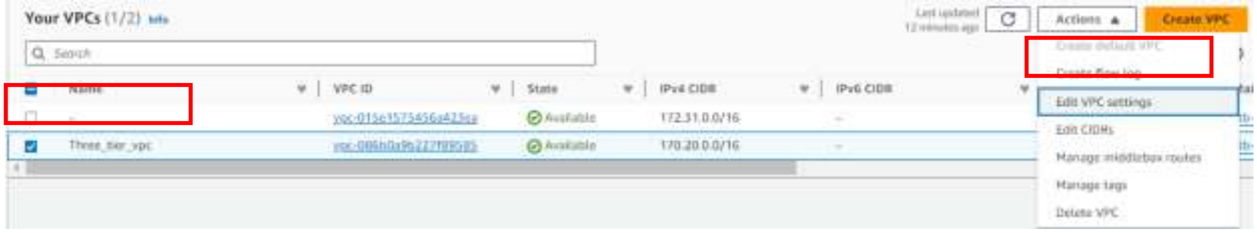
Available subnets (6/8)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	priv_sub_3a	subnet-0756a1929e3aca659	170.20.3.0/24	-	Main (rtb-021f39783646dc511)
☑	priv_sub_5a	subnet-00d3e7d7ac1001f14	170.20.5.0/24	-	Main (rtb-021f39783646dc511)
□	pub_sub_2b	subnet-08c2ef0f947644fba	170.20.2.0/24	-	rtb-0f2d0995cf3428c7f / PUB_RT
☑	priv_sub_8b	subnet-04c5f087b8fb06a24	170.20.8.0/24	-	Main (rtb-021f39783646dc511)
□	pub_sub_1a	subnet-09d61ac1566d675c5	170.20.1.0/24	-	rtb-0f2d0995cf3428c7f / PUB_RT
☑	priv_sub_6b	subnet-0a9121152d1160201	170.20.6.0/24	-	Main (rtb-021f39783646dc511)
☑	priv_sub_4b	subnet-03ca96c941ceddd66	170.20.4.0/24	-	Main (rtb-021f39783646dc511)
☑	priv_sub_7a	subnet-07cf6a70f0acad184	170.20.7.0/24	-	Main (rtb-021f39783646dc511)

Selected subnets

subnet-0756a1929e3aca659 / priv_sub_3a
subnet-00d3e7d7ac1001f14 / priv_sub_5a
subnet-04c5f087b8fb06a24 / priv_sub_8b
subnet-0a9121152d1160201 / priv_sub_6b
subnet-03ca96c941ceddd66 / priv_sub_4b
subnet-07cf6a70f0acad184 / priv_sub_7a

Change the settings of VPC and two public subnets. So just click on the VPC button on the left panel and select VPC that we have created and click on the action button and there you will see the drop-down menu. Select the Edit VPC setting button.

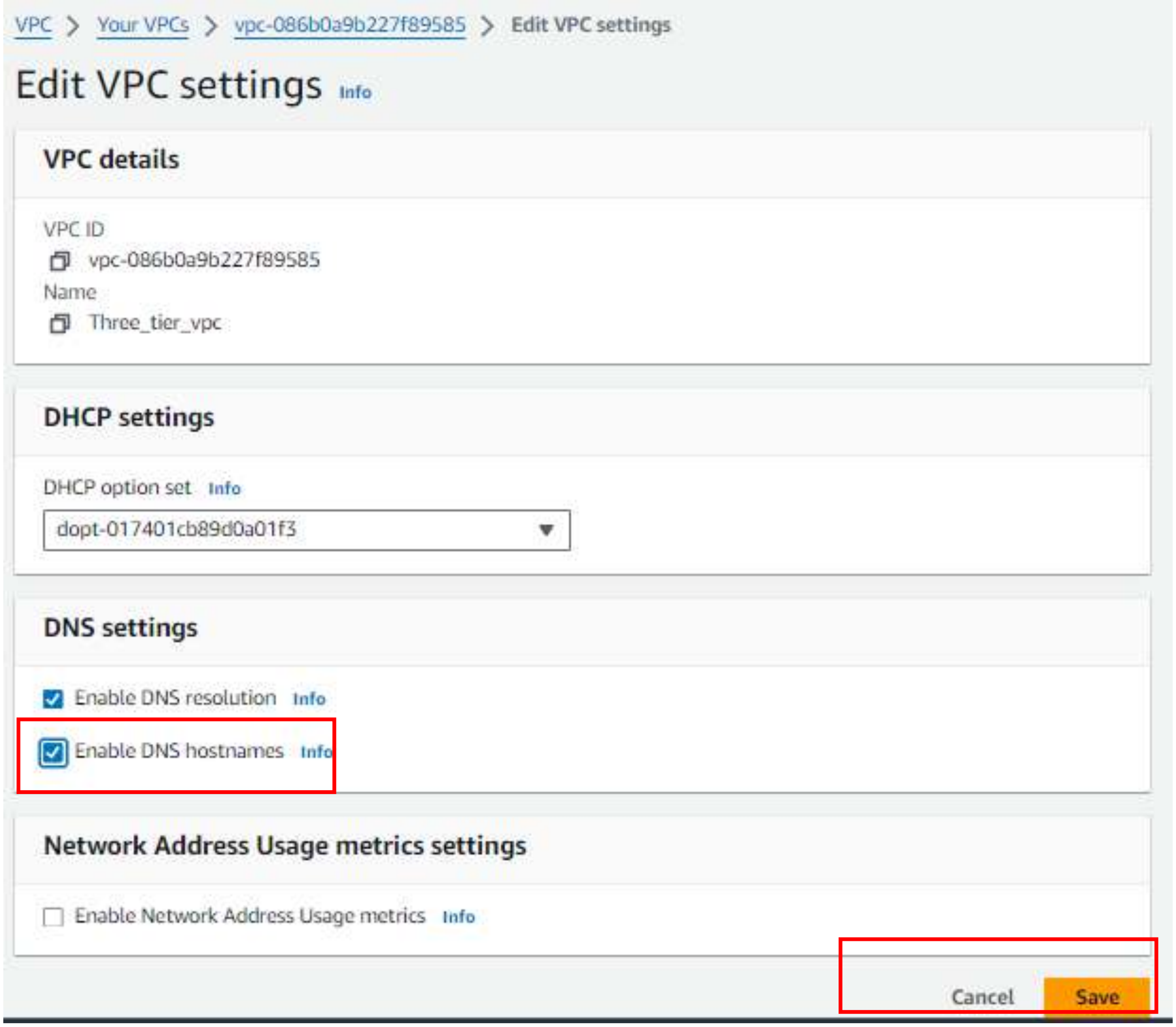


Your VPCs (1/2) Info

Last updated 12 minutes ago

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	–	vpc-015e1575456a423ea	Available	172.31.0.0/16	–
☑	Three_tier_vpc	vpc-086b0a9b227f89585	Available	170.20.0.0/16	–

And here please **Enable DNS hostname** checkbox by clicking on it and then click on the Save button



VPC > Your VPCs > vpc-086b0a9b227f89585 > Edit VPC settings

Edit VPC settings Info

VPC details

VPC ID
vpc-086b0a9b227f89585

Name
Three_tier_vpc

DHCP settings

DHCP option set Info
dopt-017401cb89d0a01f3 ▼

DNS settings

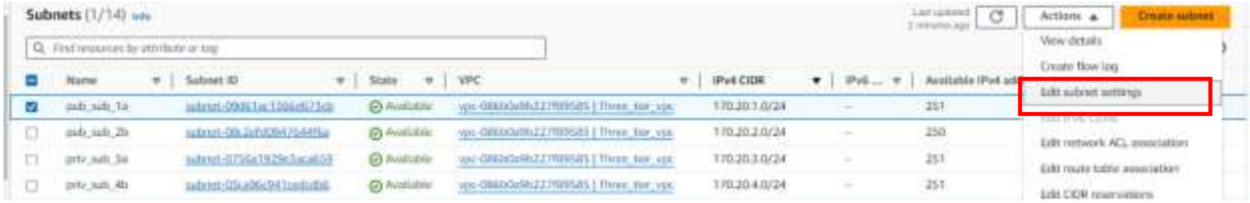
☑ Enable DNS resolution Info

☑ Enable DNS hostnames Info

Network Address Usage metrics settings

□ Enable Network Address Usage metrics Info

Please go to the subnet page and select the public subnet and click on the action button and then choose the Edit subnet setting button from the drop-down list.

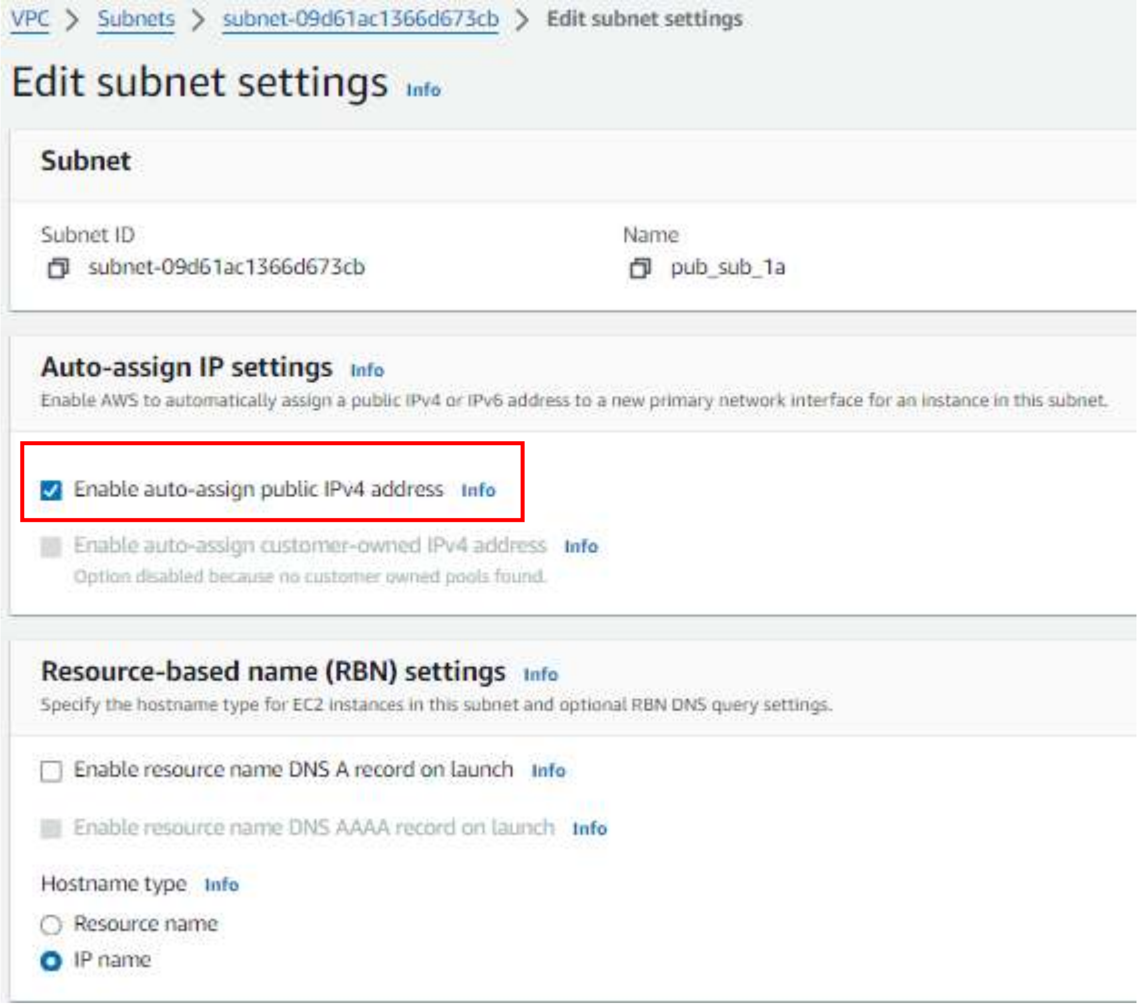


Subnets (1/14) Info

Find resources by attribute or tag

☑	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 ...	Available IPv4 ad
☑	pub_sub_1a	subnet-09d61ac1366d673cb	Available	vpc-088b0a9b227f89585 | three_tier_vpc	170.20.1.0/24	–	251
□	pub_sub_2b	subnet-0b2afd0947644ffa	Available	vpc-088b0a9b227f89585 | three_tier_vpc	170.20.2.0/24	–	250
□	prtv_sub_3a	subnet-0756a1929e3aca659	Available	vpc-088b0a9b227f89585 | three_tier_vpc	170.20.3.0/24	–	251
□	prtv_sub_4b	subnet-05ca06c941ceddb6	Available	vpc-088b0a9b227f89585 | three_tier_vpc	170.20.4.0/24	–	251

Here you have to mark right on **Enable public assign public IPV4 address**. And then click on the save button



VPC > Subnets > subnet-09d61ac1366d673cb > Edit subnet settings

Edit subnet settings Info

Subnet

Subnet ID: subnet-09d61ac1366d673cb

Name: pub_sub_1a

Auto-assign IP settings Info

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

☑ Enable auto-assign public IPv4 address Info

□ Enable auto-assign customer-owned IPv4 address Info
Option disabled because no customer owned pools found.

Resource-based name (RBN) settings Info

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

□ Enable resource name DNS A record on launch Info

□ Enable resource name DNS AAAA record on launch Info

Hostname type Info

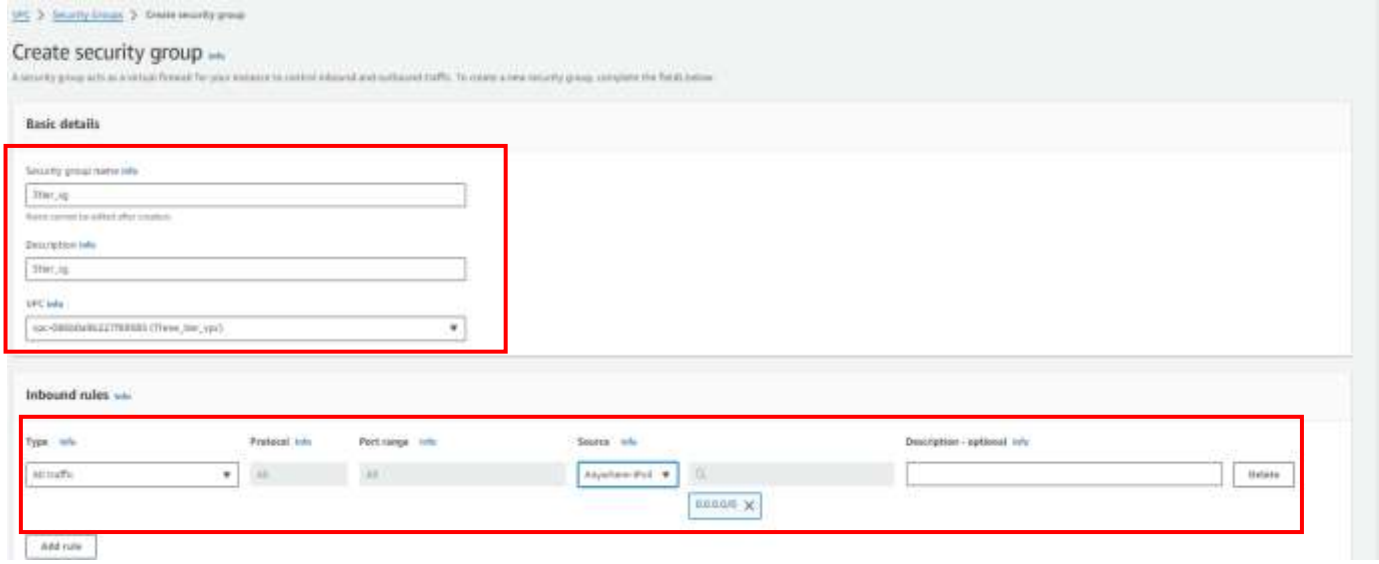
○ Resource name
● IP name

Same for pub_sub_2b, **Enable public assign public IPV4 address**.

Security Groups (SG):

Security groups are very essential part of the infrastructure. Because it can secure the resources in the cloud. SGs are a kind of firewall that allow or block incoming and outgoing traffic. SGs are applied to the resources like ALB, ec2, rds, etc. One resource can have more than one SG.

To create SG, click on the security groups tab on the left panel and here you will see the Security Groups button. Note that SGs are specific with VPC. So we can't use SG which is created in a different VPC. So when you create SG please make sure that you choose the right VPC. Click on the create security button on the top right corner.



VPC > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info: 3tier_sg

Name cannot be edited after creation.

Description Info: 3tier_sg

VPC Info: vpc-[illegible] (Three_tier_vpc)

Inbound rules Info

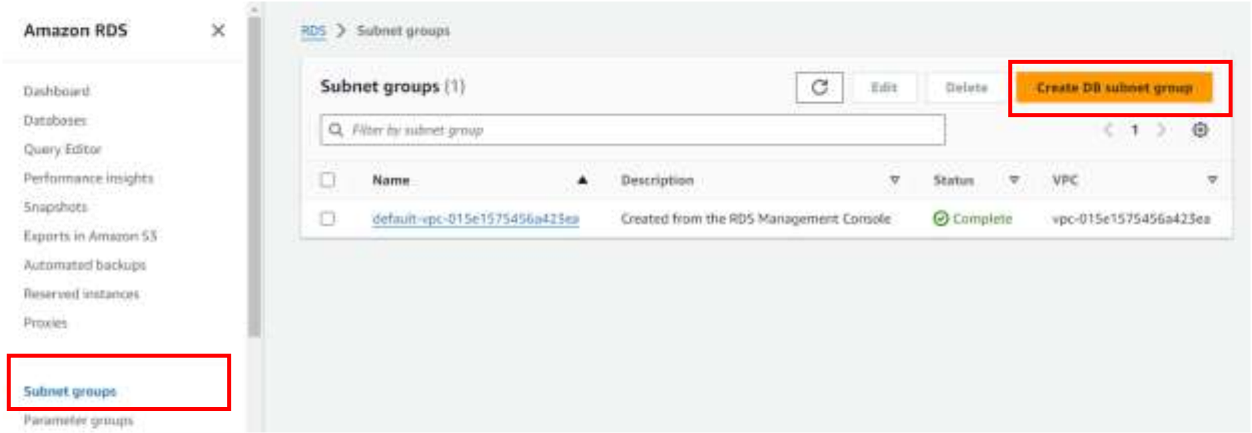
Type	Protocol	Port range	Source		Description - optional
All traffic	All	All	Anywhere-IPv4	0.0.0.0/0	

And here our SG setups are complete now.

RDS and Route 53:

Now we are going to set up a database for our application. And for that, we are going to utilize the RDS service of AWS. So let's head over to the RDS dashboard. Just search RDS in the AWS console. And click on the service.

Now first we need to set up a subnet group. It specifies in which subnet and Availability zone out database instance will be created. So click on the subnet group button on the left panel. And click on the button Create database subnet group which is in the middle of the web page.



□	Name ▲	Description	Status	VPC
□	default-vpc-015e1575456a423ea	Created from the RDS Management Console	Complete	vpc-015e1575456a423ea

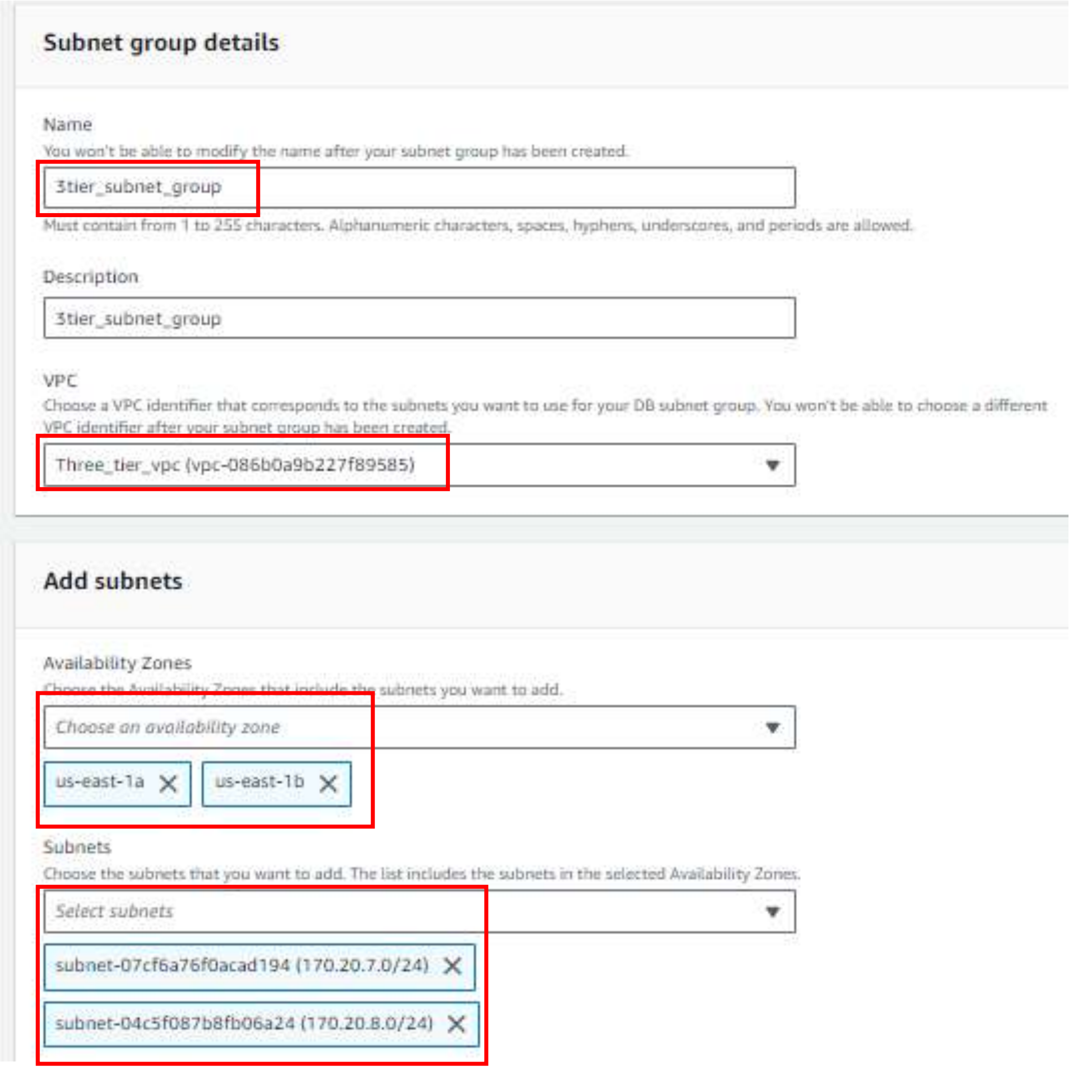
Here we can configure our VPC, subnet, and availability zone.

Give any name to your subnet but make sure you select the correct VPC.

And select Azs **us-east-1a** and **us-east-2b**.

According to the architecture that I have shown you, our database will be in private subnet **pri-sub-7a** and **pri-sub-8b**.

So please select as I have shown in the below figure. And then click on the create button.



Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

3tier_subnet_group

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

3tier_subnet_group

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Three_tier_vpc (vpc-086b0a9b227f89585)

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

us-east-1a

us-east-1b

Subnets

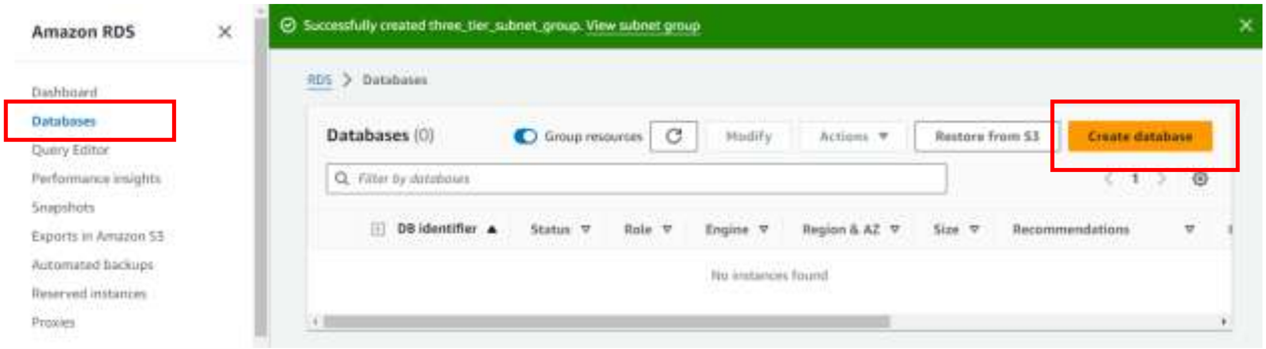
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

subnet-07cf6a76f0acad194 (170.20.7.0/24)

subnet-04c5f087b8fb06a24 (170.20.8.0/24)

Create a database. So click on the database button on the left panel and then click on the created database button.

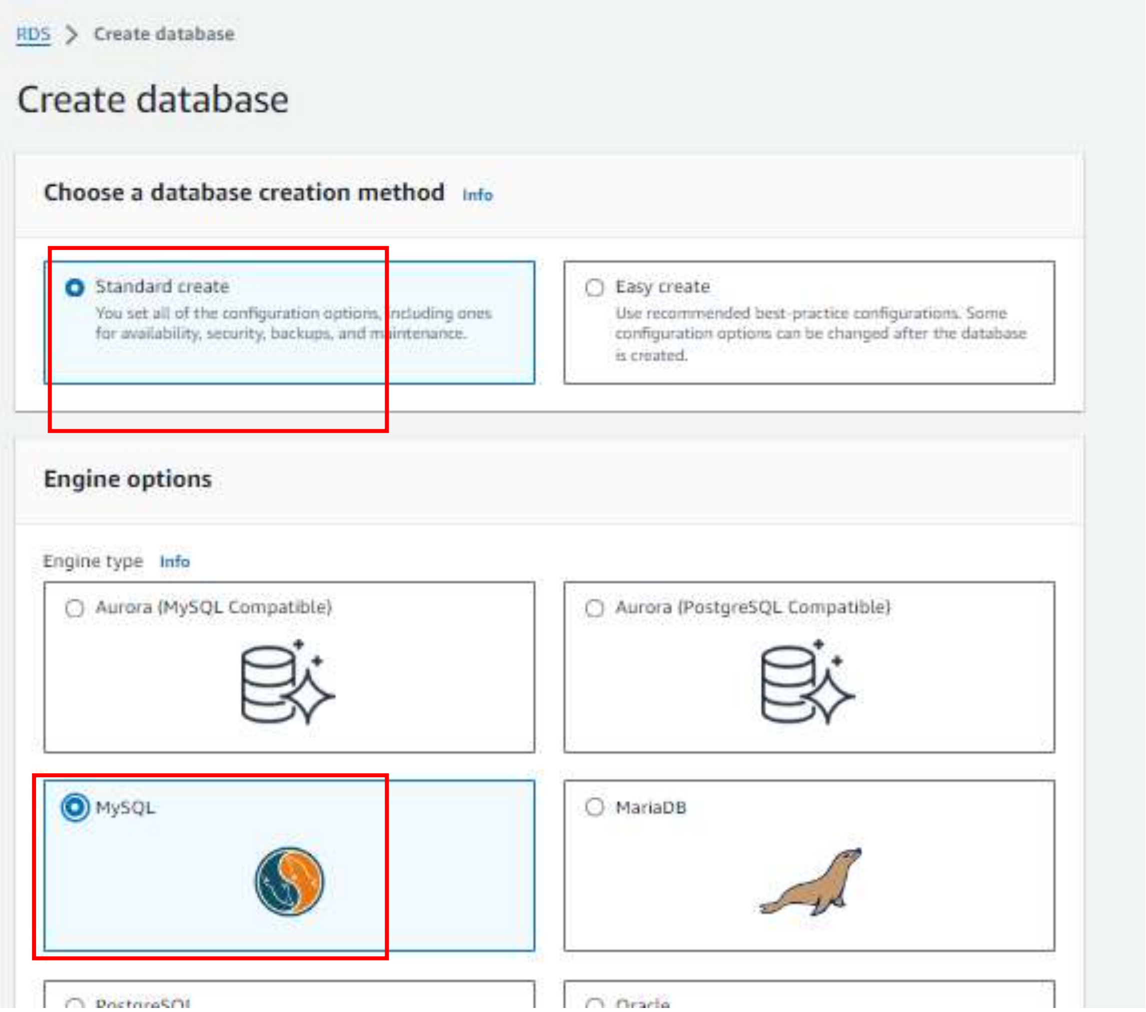


Successfully created three_tier_subnet_group. View subnet group

Databases (0)

No instances found

On this page, we can configure our database. Select standard create. Select MySQL in the engine option because our application runs on MySQL database Furthermore, you can select the engine version my application is compatible with MySQL version.



RDS > Create database

Create database

Choose a database creation method Info

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

Scroll down, and select Dev/test as template.

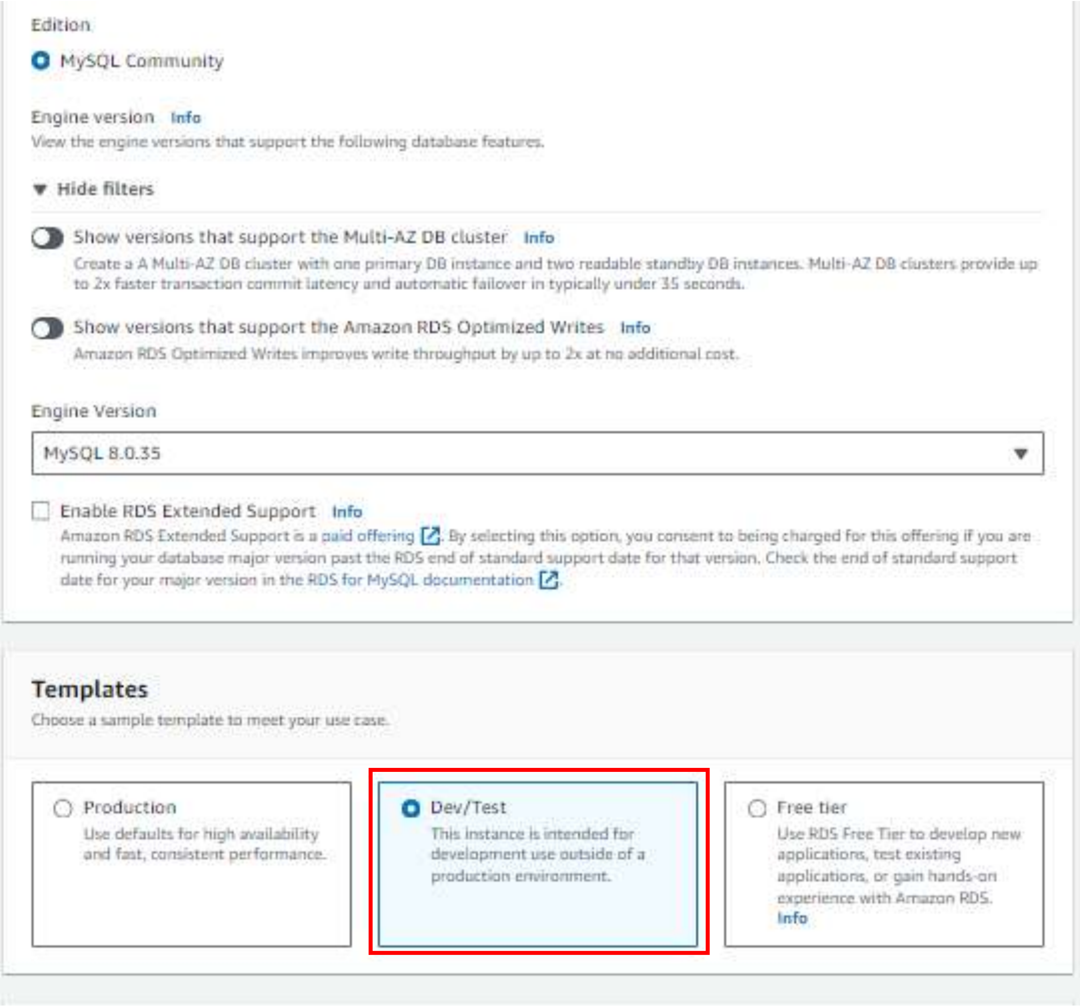
If you select the free tier then you won’t be able to deploy RDS in a multi-availability zone.

Select Multi-AZ DB instance from availability and durability option.

In settings give any name to your database.

In the credential setting give the username of the database in the Master Username field and give the password in the Master password field. And then confirm the password below.

Please do remember your username and password.



Edition

● MySQL Community

Engine version Info

View the engine versions that support the following database features.

▼ Hide filters

Show versions that support the Multi-AZ DB cluster Info

Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes Info

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

MySQL 8.0.35

□ Enable RDS Extended Support Info

Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL documentation.

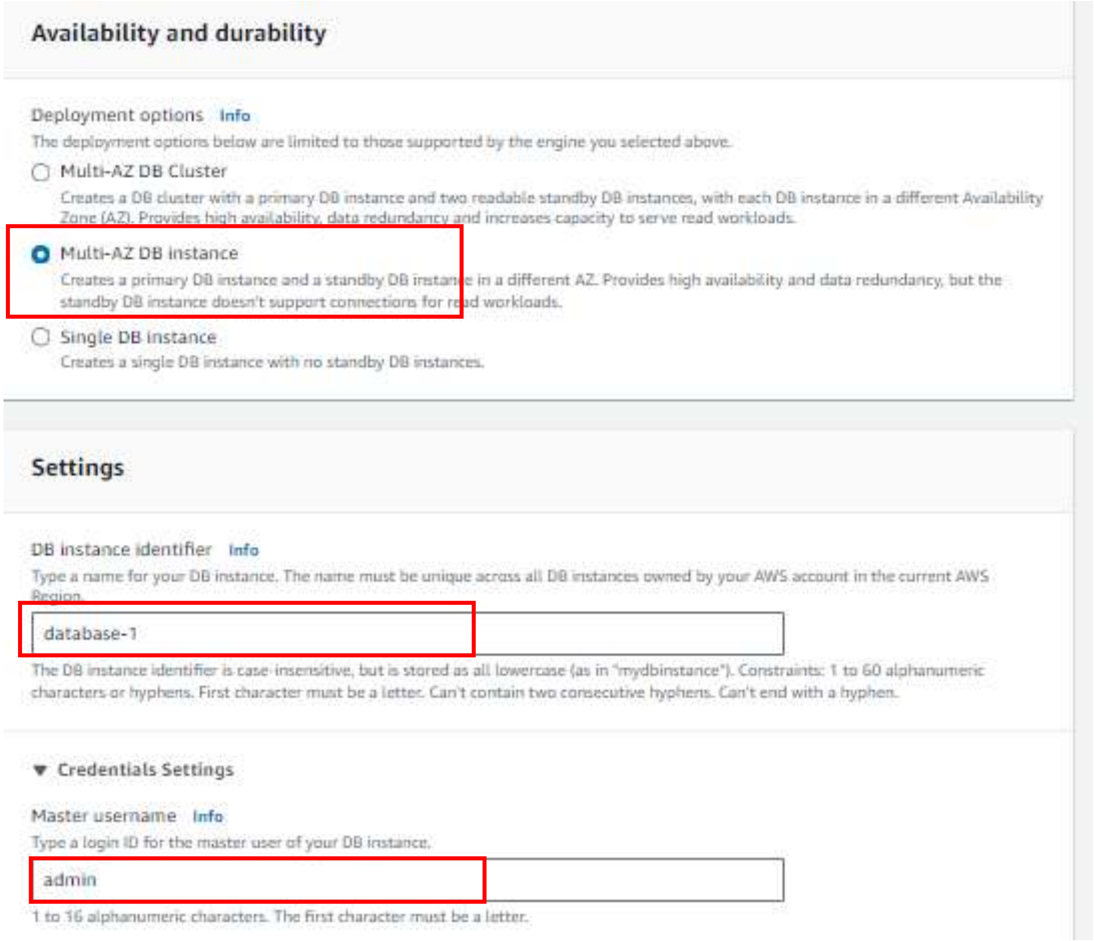
Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

● Dev/Test
This instance is intended for development use outside of a production environment.

○ Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info



Availability and durability

Deployment options

The deployment options below are limited to those supported by the engine you selected above.

○ Multi-AZ DB Cluster
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

● Multi-AZ DB instance
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

○ Single DB instance
Creates a single DB instance with no standby DB instances.

Settings

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

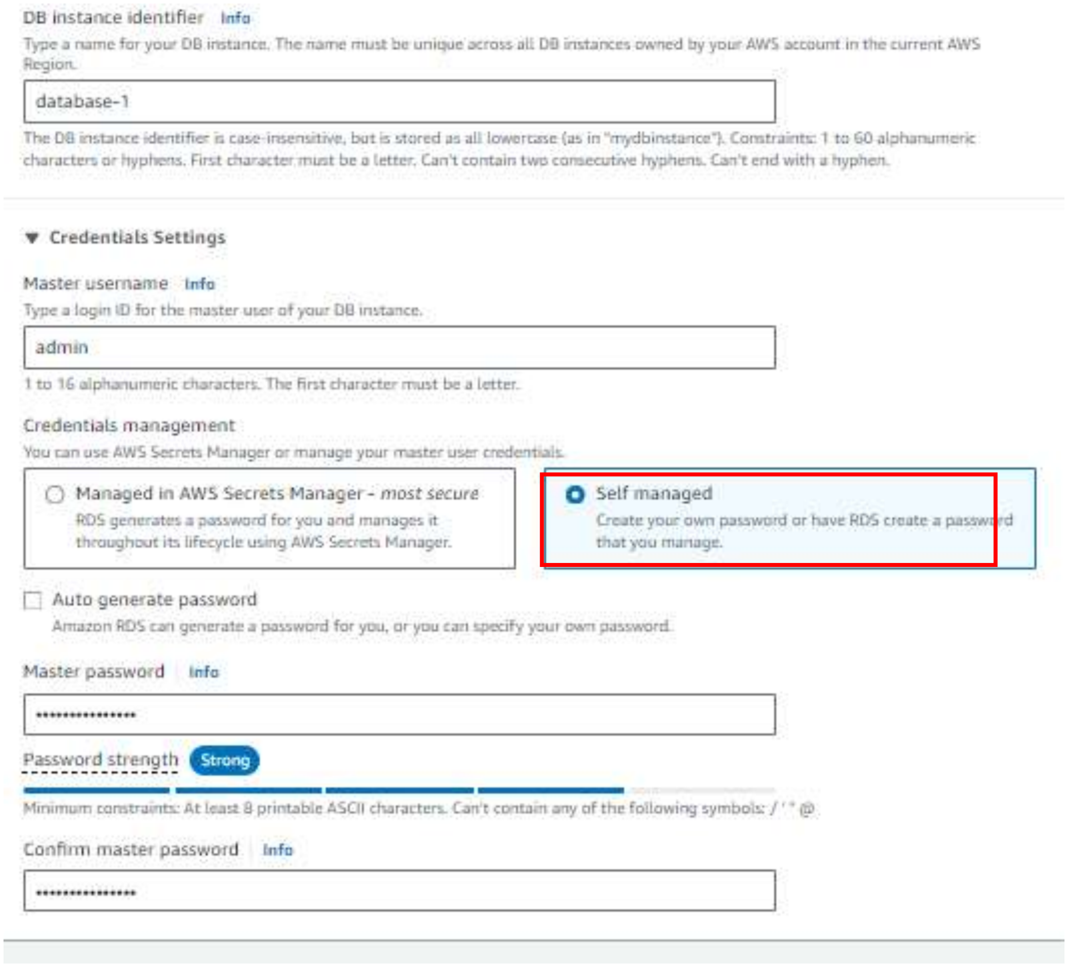
▼ Credentials Settings

Master username

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.



DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

● Self managed
Create your own password or have RDS create a password that you manage.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

•••••••••••••••

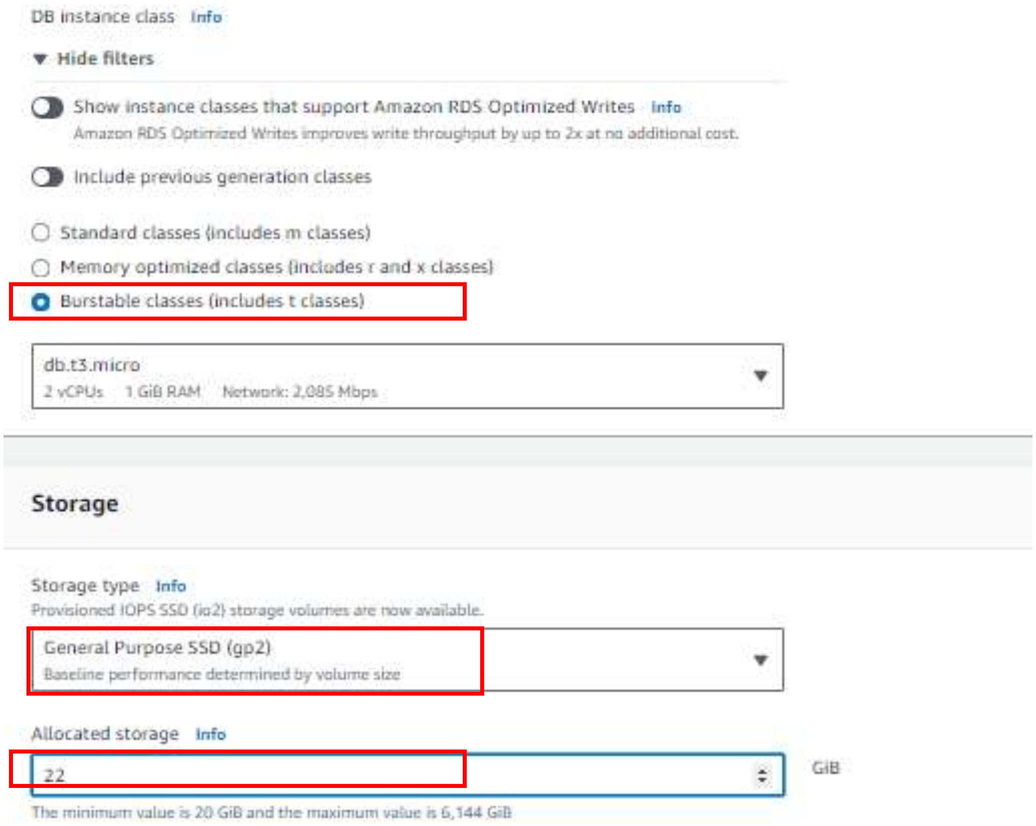
Password strength Strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password Info

•••••••••••••••

Again scroll down, select Brustable class in the instance setting and select the instance type. Actually, it depends on your application uses. But for learning purposes, I am selecting t3.micro. Now in storage type select General purpose (GP2) and allocate 22 GiB for database. Please uncheck the auto-scaling option to keep our costs low. And In the connectivity option please select the option according below screenshot.



DB instance class Info

▼ Hide filters

Show instance classes that support Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Include previous generation classes

○ Standard classes (includes m classes)
○ Memory optimized classes (includes r and x classes)
● Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Storage

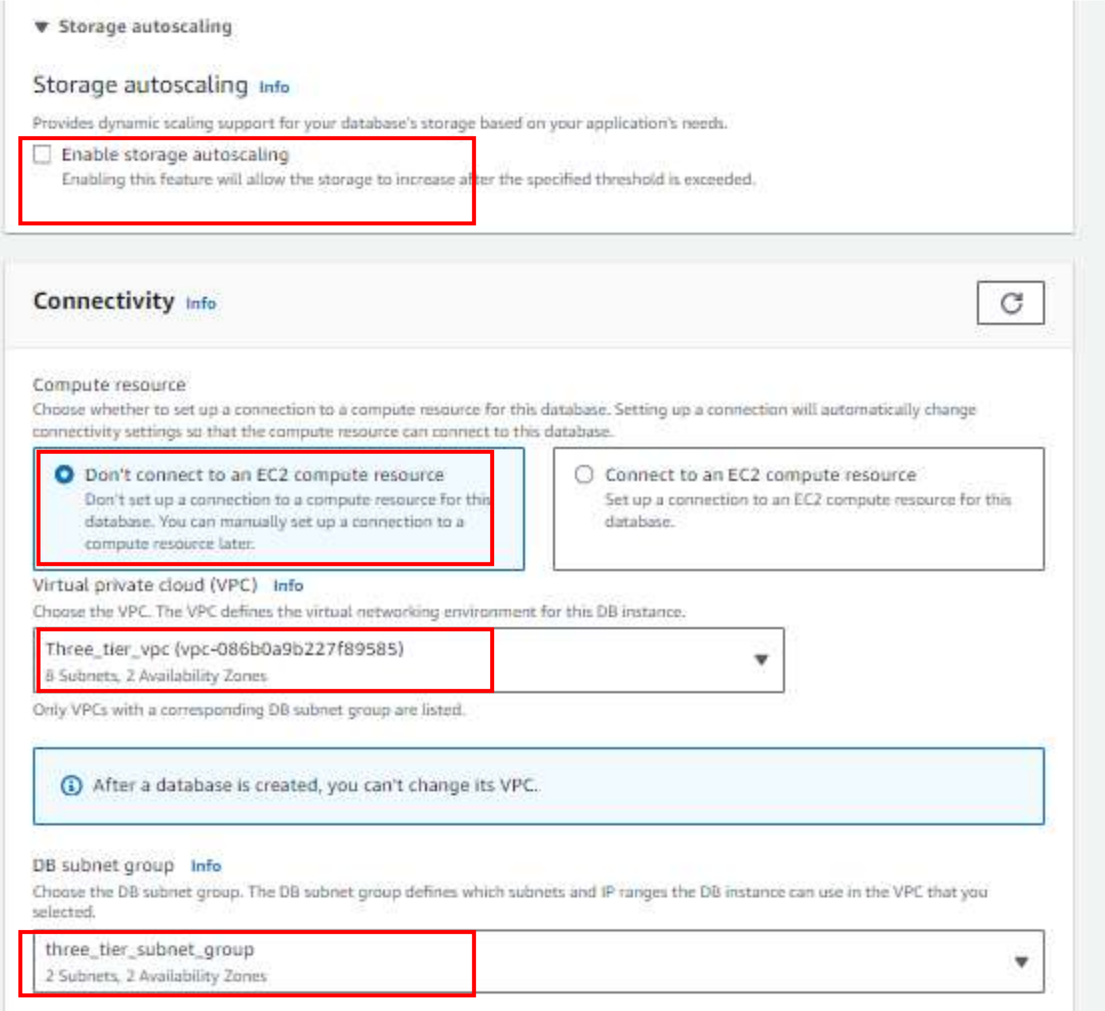
Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage Info

22 GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB



▼ Storage autoscaling

Storage autoscaling Info

Provides dynamic scaling support for your database's storage based on your application's needs.

☐ Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Connectivity Info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

● Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

○ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Three_tier_vpc (vpc-086b0a9b227f89585)
8 Subnets, 2 Availability Zones

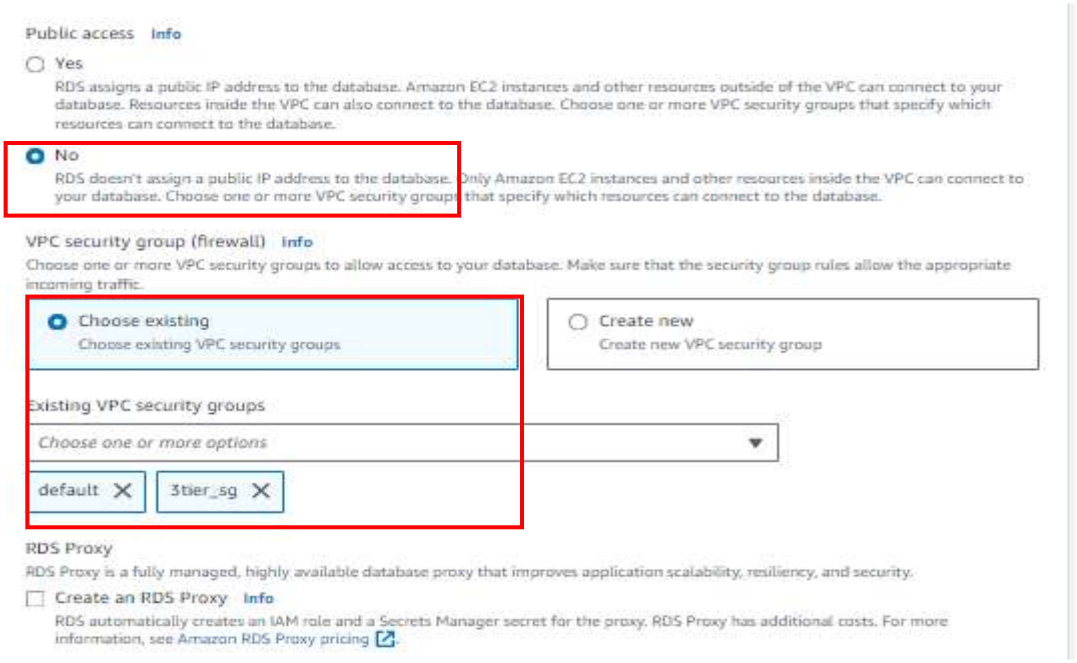
Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

three_tier_subnet_group
2 Subnets, 2 Availability Zones

In VPC, select VPC that we created earlier and in DB subnet group select the group that we just created, In the public access option please select No, choose existing security, and select security group **book-rds-db**.



Public access Info

○ Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

● No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

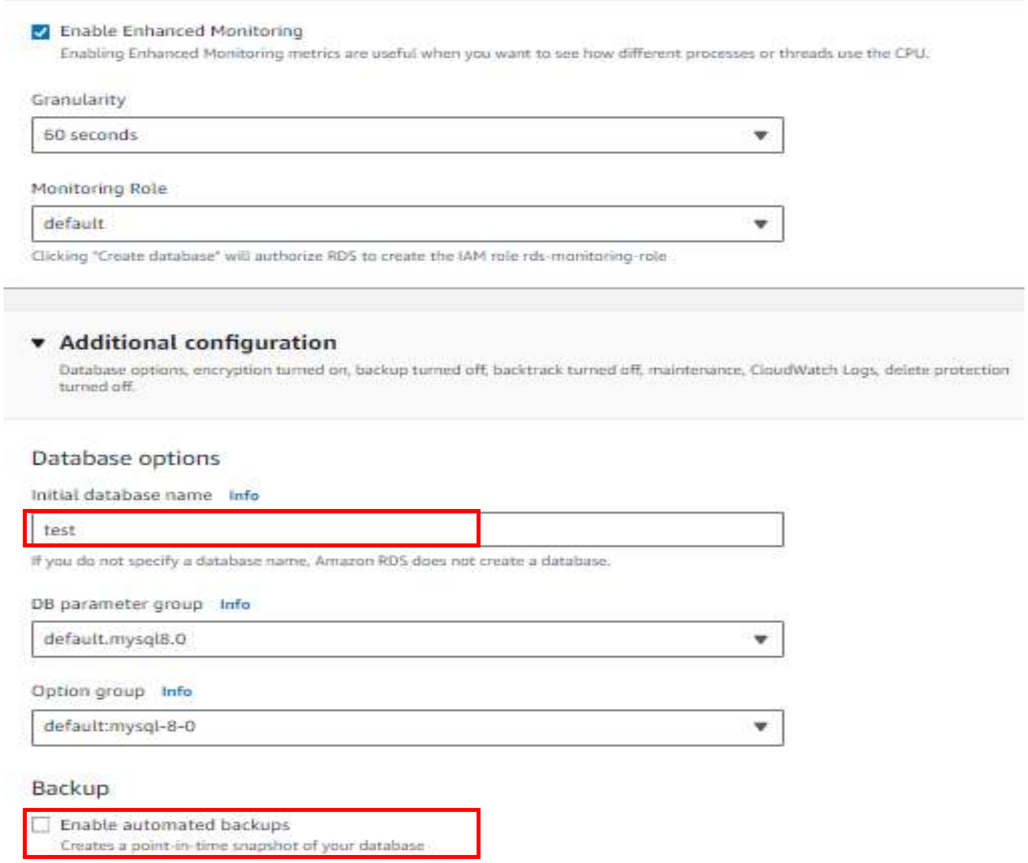
○ Create new
Create new VPC security group

Existing VPC security groups
Choose one or more options
default ✕ 3tier_sg ✕

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

□ Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.

Scroll down, click on Additional Configuration, and in the database option give the name **test** because we need a database with the name of the **test** in the application.



☑ Enable Enhanced Monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Granularity
60 seconds

Monitoring Role
default
Clicking "Create database" will authorize RDS to create the IAM role rds-monitoring-role

▼ Additional configuration
Database options, encryption turned on, backup turned off, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name Info
test
If you do not specify a database name, Amazon RDS does not create a database.

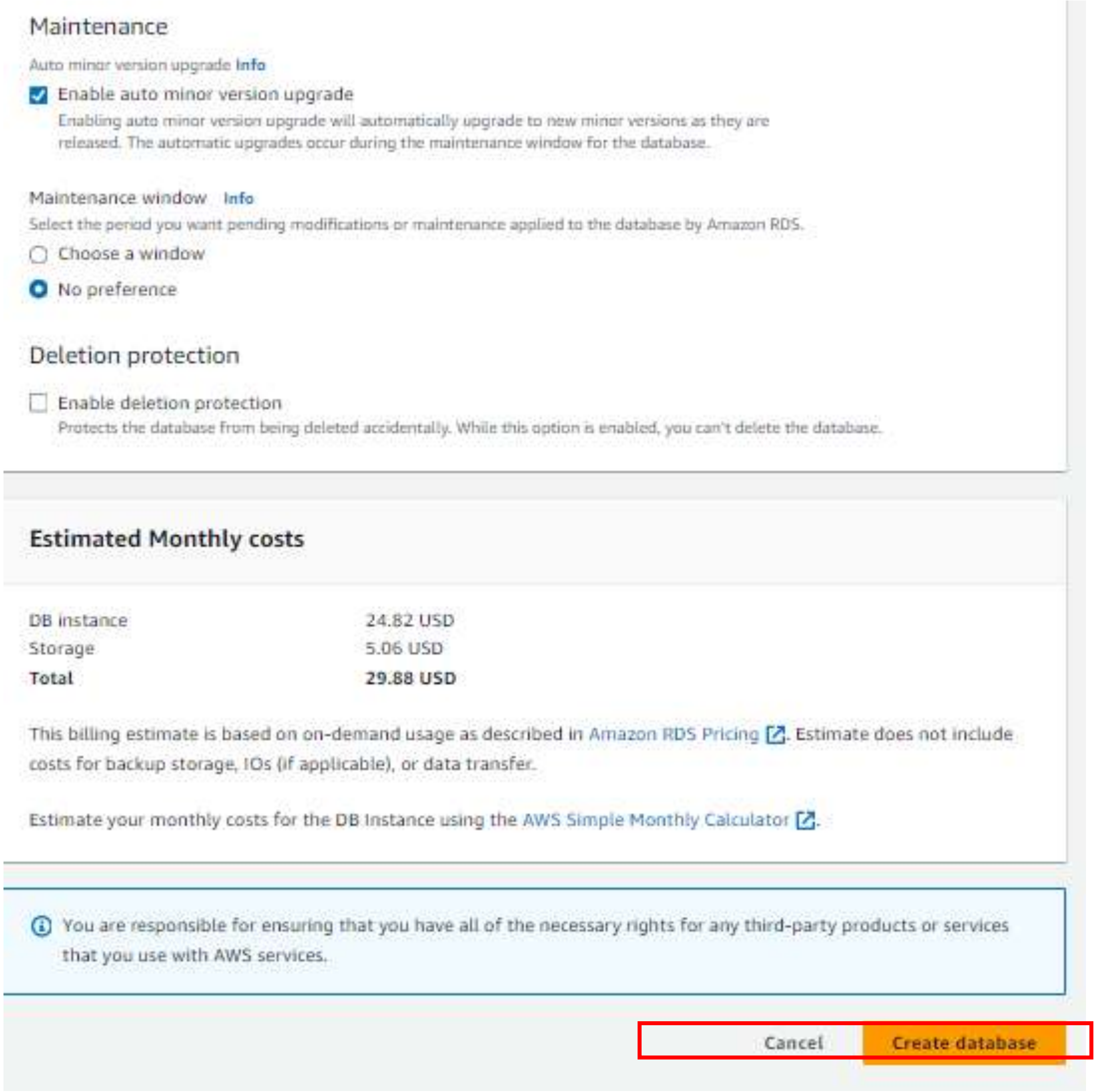
DB parameter group Info
default.mysql8.0

Option group Info
default:mysql-8-0

Backup

□ Enable automated backups
Creates a point-in-time snapshot of your database

Scroll down, mark on enable encryption checkbox to make the database bit more secure, and click on Create database button below.



Maintenance

Auto minor version upgrade Info

☑ Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window Info
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.
○ Choose a window
● No preference

Deletion protection

□ Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Estimated Monthly costs

DB instance	24.82 USD
Storage	5.06 USD
Total	**29.88 USD**

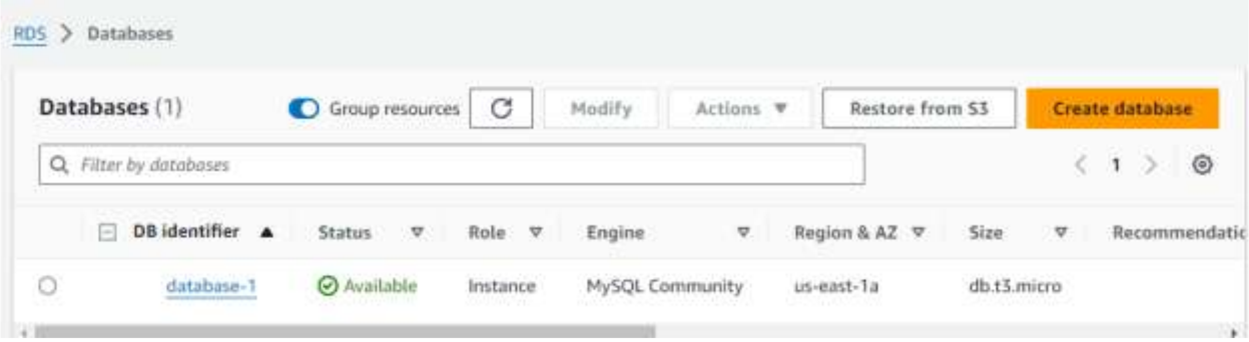
This billing estimate is based on on-demand usage as described in Amazon RDS Pricing. Estimate does not include costs for backup storage, IOs (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Note: *RDS take 15-20 minute because it creates a database*

After your database is completely ready and you see the status Available

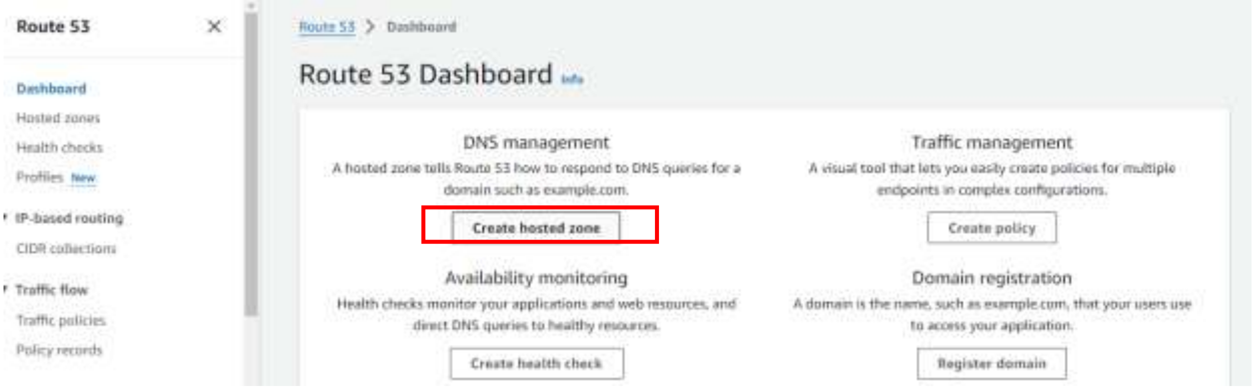


DB identifier ▲	Status	Role	Engine	Region & AZ	Size	Recommendatic
database-1	✓ Available	Instance	MySQL Community	us-east-1a	db.t3.micro	

Route 53

Now we are going to utilize route 53 services and create private hosted zone. So head over to Route 53. Type route 53 in the AWS console. And click on the service.

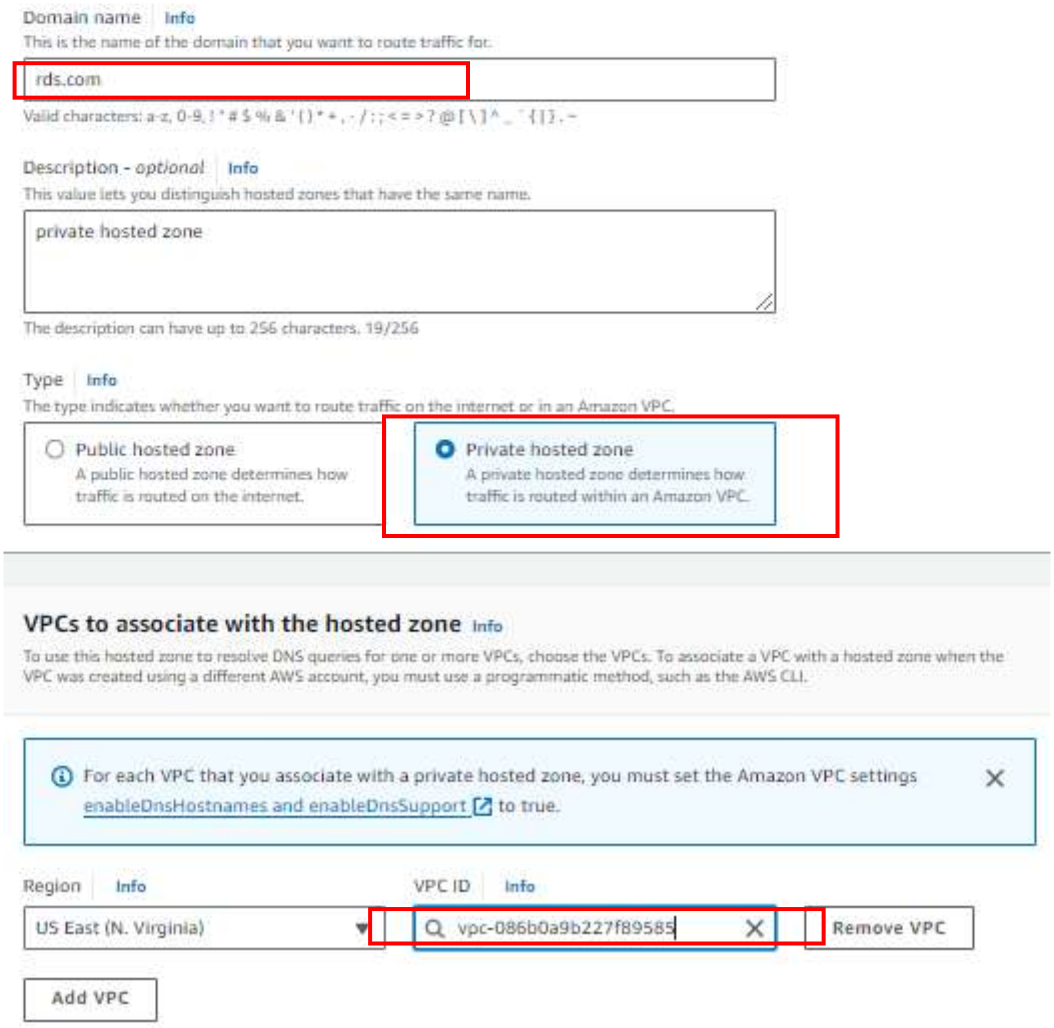
Firstly, we are going create a hosted zone. Click on the created hosted zone


Route 53
Dashboard
Hosted zones
Health checks
Profiles New
IP-based routing
CIDR collections
Traffic flow
Traffic policies
Policy records
Route 53 > Dashboard
Route 53 Dashboard Info
DNS management
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.
Create hosted zone
Traffic management
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.
Create policy
Availability monitoring
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.
Create health check
Domain registration
A domain is the name, such as example.com, that your users use to access your application.
Register domain

Give any domain name because anyhow it will be private hosted zone but it would be great if you give the name same as mine (**rds.com**).

Please select the private hosted zone and Select the region. In my case, it is **us-east-1**. And then select VPC ID.

Make sure you select VPC that we created earlier. Because this hosted zone will resolve the record only in specified VPC. And then click on the Create hosted zone.



Domain name | Info

This is the name of the domain that you want to route traffic for.

rds.com

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - *optional* | Info

This value lets you distinguish hosted zones that have the same name.

private hosted zone

The description can have up to 256 characters. 19/256

Type | Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

○ Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

● Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

VPCs to associate with the hosted zone Info

To use this hosted zone to resolve DNS queries for one or more VPCs, choose the VPCs. To associate a VPC with a hosted zone when the VPC was created using a different AWS account, you must use a programmatic method, such as the AWS CLI.

For each VPC that you associate with a private hosted zone, you must set the Amazon VPC settings enableDnsHostnames and enableDnsSupport to true.

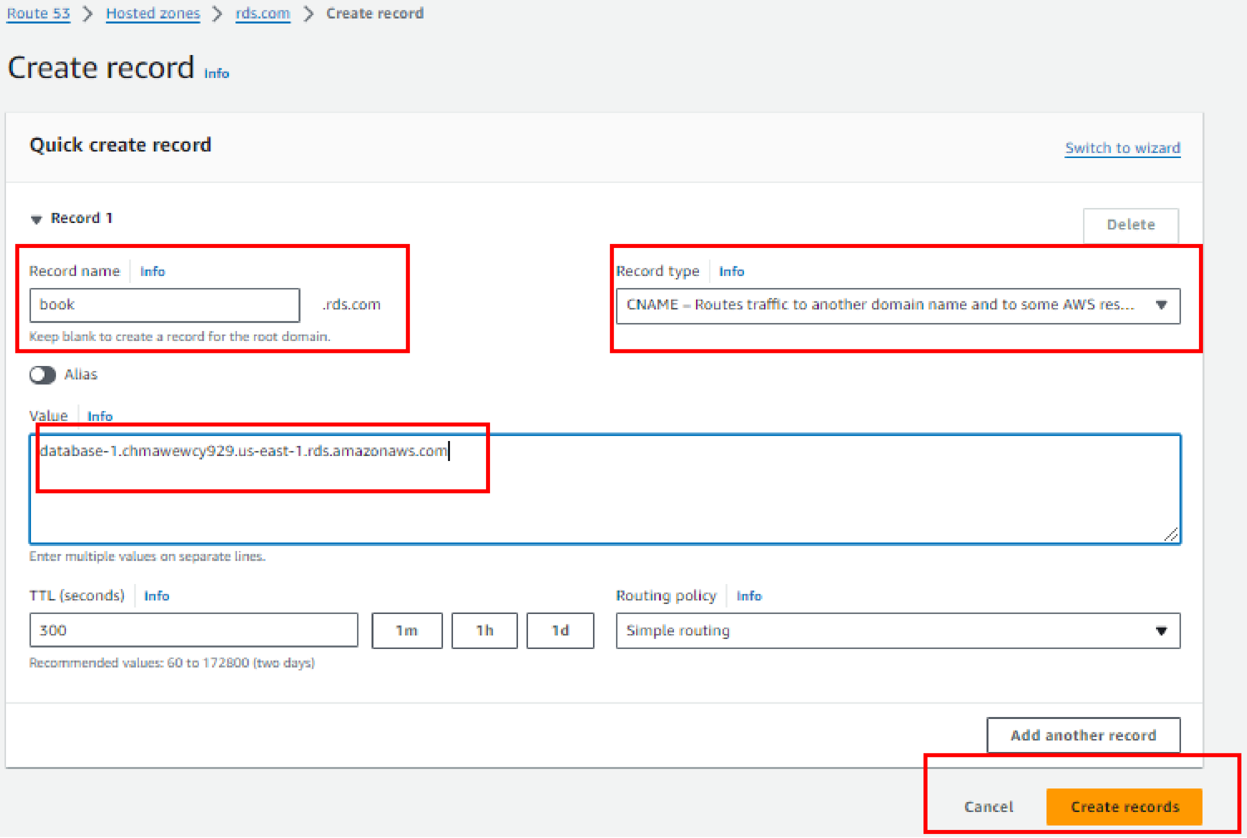
Region | Info: US East (N. Virginia)

VPC ID | Info: vpc-086b0a9b227f89585

Now we are going to create a Record that points to our RDS instance which is in **us-east-1**. So click on create record button on the top right corner.

Here type book in the record name field. In the record type select CNAME. In the value field paste **endpoint of the RDS which is in us-east-1**. Then click on the defined record button.

Click on create record button.



Route 53 > Hosted zones > rds.com > Create record

Create record Info

Quick create record

Record 1

Record name: book .rds.com

Keep blank to create a record for the root domain.

Record type: CNAME – Routes traffic to another domain name and to some AWS res...

Alias

Value: database-1.chmawewcy929.us-east-1.rds.amazonaws.com

Enter multiple values on separate lines.

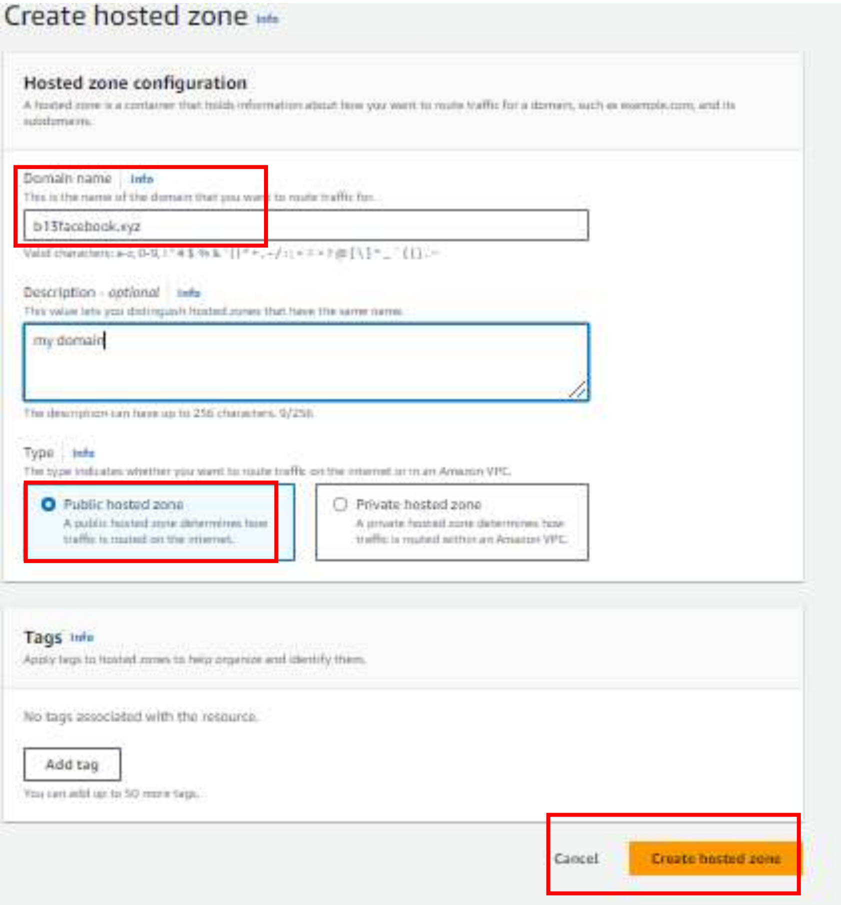
TTL (seconds): 300

Recommended values: 60 to 172800 (two days)

Routing policy: Simple routing

Creating our domain Host:

1. **Navigate to Hosted Zones** in the Route 53 dashboard.
2. **Click on "Create Hosted Zone"**.
3. **Enter the Domain Name** (e.g.: **b13facebook.xyz)** you want to associate with the hosted zone.
4. **Select Public Hosted Zone** as the type.
5. **Click on "Create"**.



Create hosted zone Info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name | Info

This is the name of the domain that you want to route traffic for.

b13facebook.xyz

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - optional | Info

This value lets you distinguish hosted zones that have the same name.

my domain

The description can have up to 256 characters. 0/256

Type | Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

● Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

○ Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

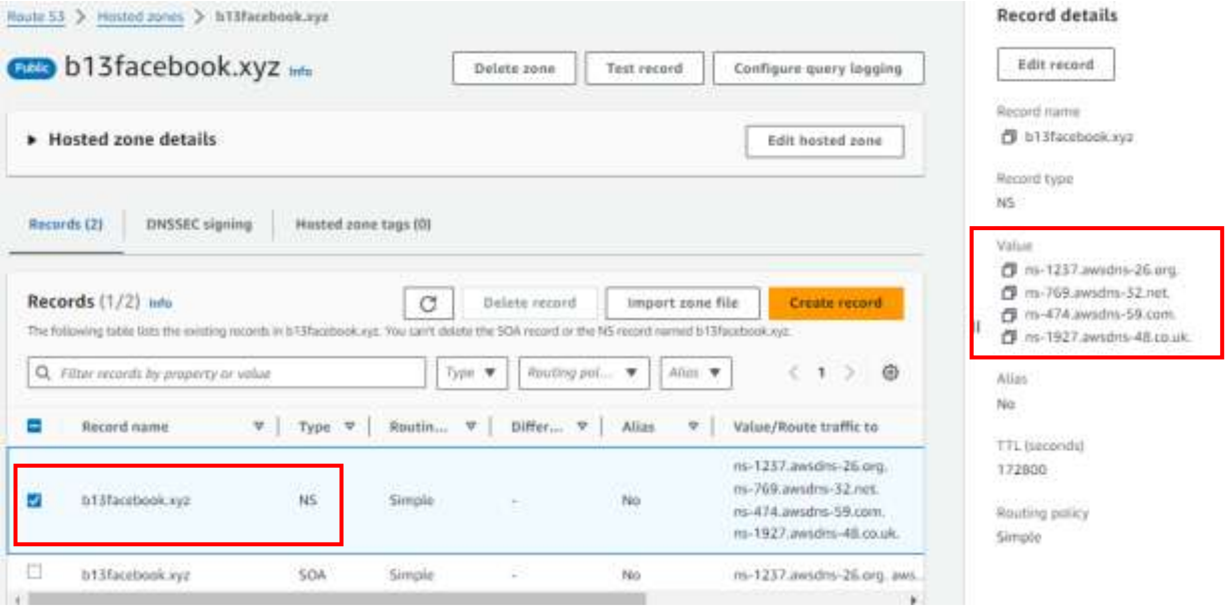
Tags Info

Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.

You can add up to 50 more tags.

After creating hosted zone, select the record b13facebook.xyz and copy the values



Public b13facebook.xyz Info

Records (1/2) Info

The following table lists the existing records in b13facebook.xyz. You can't delete the SOA record or the NS record named b13facebook.xyz.

☐	Record name	Type	Routin...	Differ...	Alias	Value/Route traffic to
☑	b13facebook.xyz	NS	Simple	-	No	ns-1237.awsdns-26.org. ns-769.awsdns-32.net. ns-474.awsdns-59.com. ns-1927.awsdns-48.co.uk.
☐	b13facebook.xyz	SOA	Simple	-	No	ns-1237.awsdns-26.org. aws...

Record details

Record name
b13facebook.xyz

Record type
NS

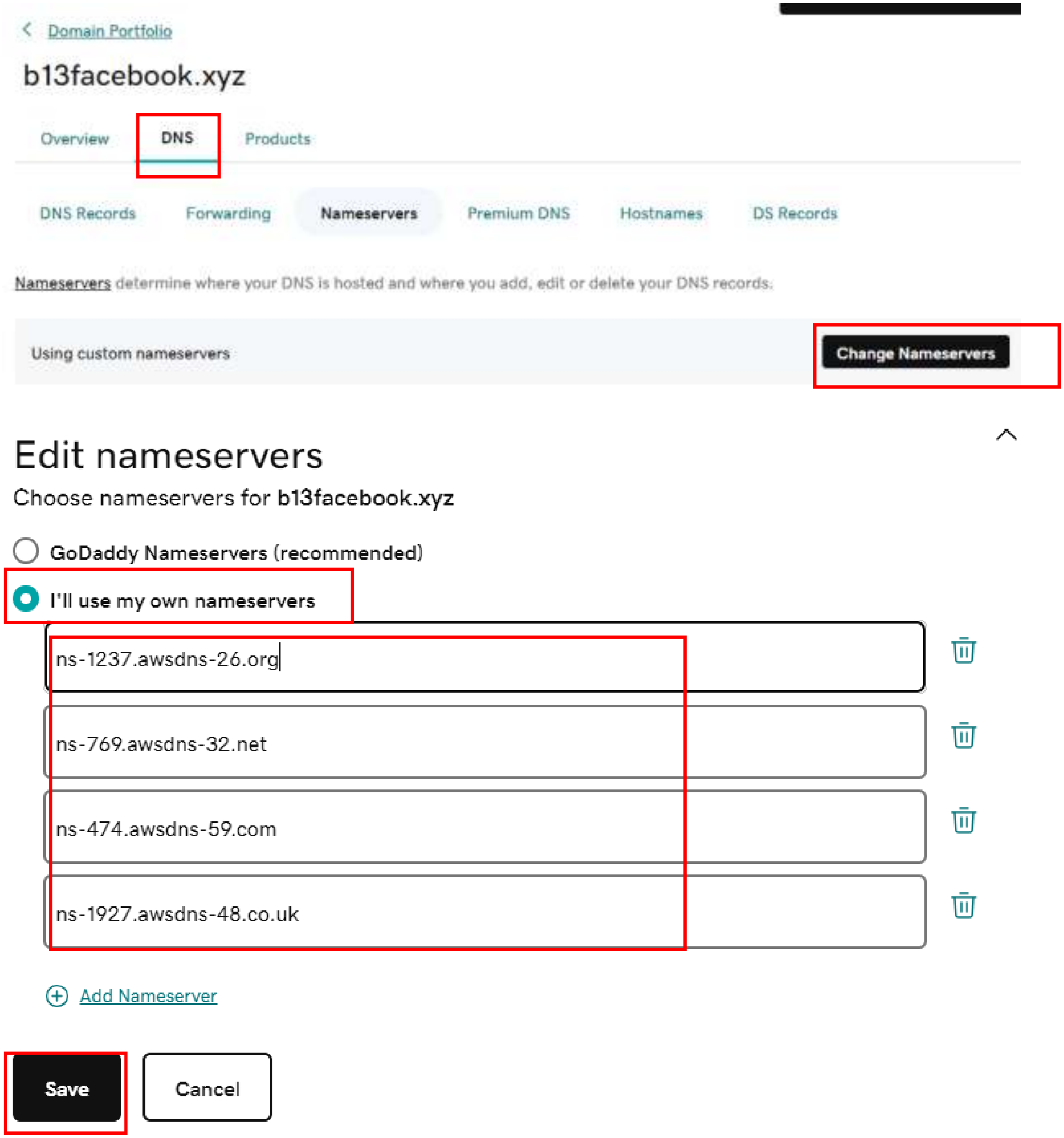
Value
ns-1237.awsdns-26.org.
ns-769.awsdns-32.net.
ns-474.awsdns-59.com.
ns-1927.awsdns-48.co.uk.

Alias
No

TTL (seconds)
172800

Routing policy
Simple

Add these values to “godaddy” name servers, as shown in below figure.


Domain Portfolio
b13facebook.xyz
Overview
DNS
Products
DNS Records
Forwarding
Nameservers
Premium DNS
Hostnames
DS Records
Nameservers determine where your DNS is hosted and where you add, edit or delete your DNS records.
Using custom nameservers
Change Nameservers
Edit nameservers
Choose nameservers for b13facebook.xyz
○ GoDaddy Nameservers (recommended)
● I'll use my own nameservers
Add Nameserver
Save
Cancel

ns-1237.awsdns-26.org
ns-769.awsdns-32.net
ns-474.awsdns-59.com
ns-1927.awsdns-48.co.uk

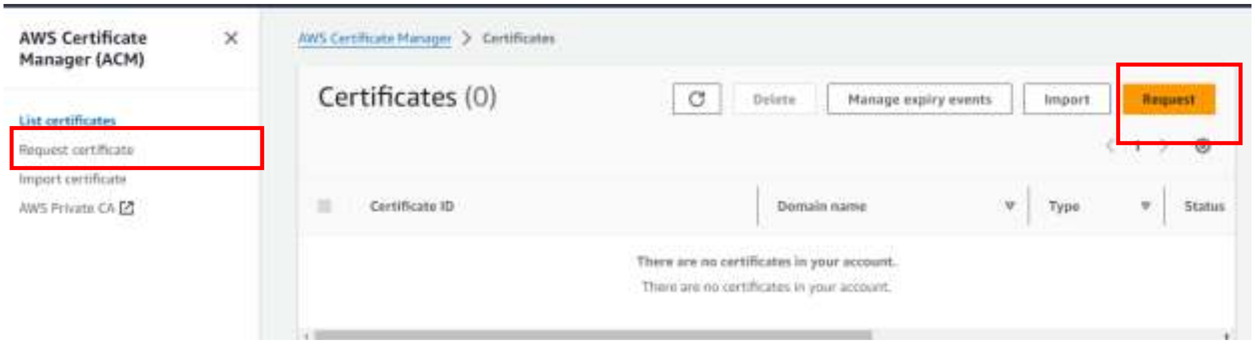
It Takes 5-10mins to create

Certificate Manager

I have the domain name b13facebook.xyz from godaddy in Route 53. Now I am going to use this domain name to create sub domains such as api.b13facebook.xyz and that will resolve **ALB-backend DNS**. Furthermore, we need an SSL certificate so that we can make the connection secure.

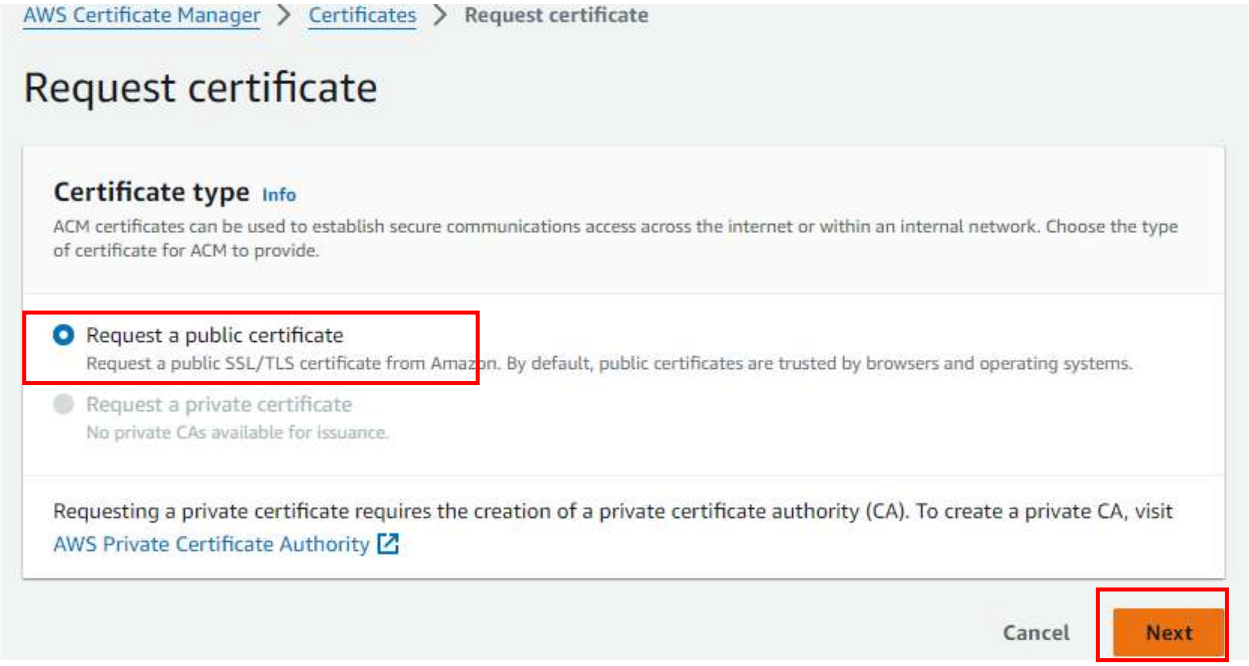
So let's head over to ACM (AWS certificate manager).Type certificate manager in the AWS console search bar. And click on the service.

Now click on the list certificates button on the left panel and then click on the request certificate on the top right corner.



There are no certificates in your account.
There are no certificates in your account.

Select the option Request the public certificate and click on the next button.


AWS Certificate Manager > Certificates > Request certificate
Cancel
Next

Request certificate

Certificate type Info

ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

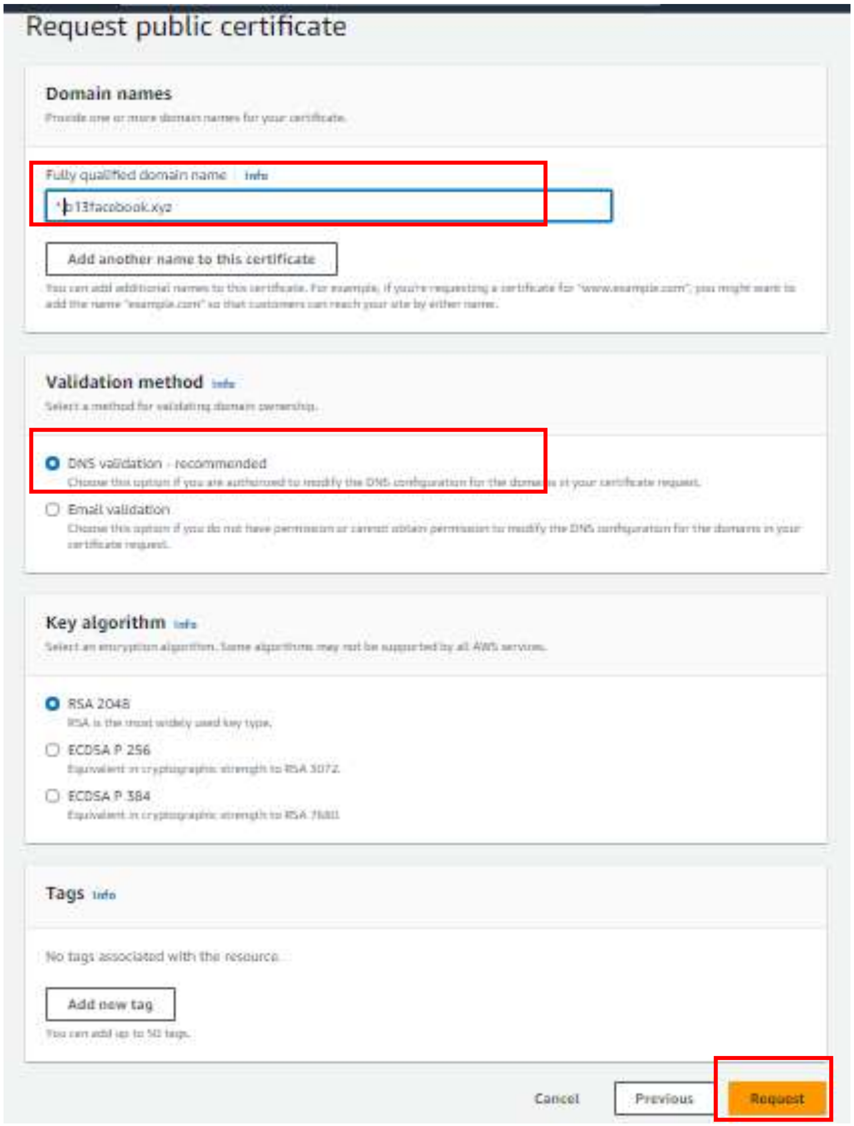
● Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

○ Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit AWS Private Certificate Authority

In the domain name field please type ***.Your_Domain_Name.xyz** in my case it is ***.b13facebook.xyz**.

In the validation methods select DNS validation and click on the request certificate.



Request public certificate

Domain names

Provide one or more domain names for your certificate.

Fully qualified domain name Info

*.b13facebook.xyz

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Validation method Info

Select a method for validating domain ownership.

● DNS validation - recommended

Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

○ Email validation

Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

Key algorithm Info

Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

● RSA 2048

RSA is the most widely used key type.

○ ECDSA P 256

Equivalent in cryptographic strength to RSA 3072.

○ ECDSA P 384

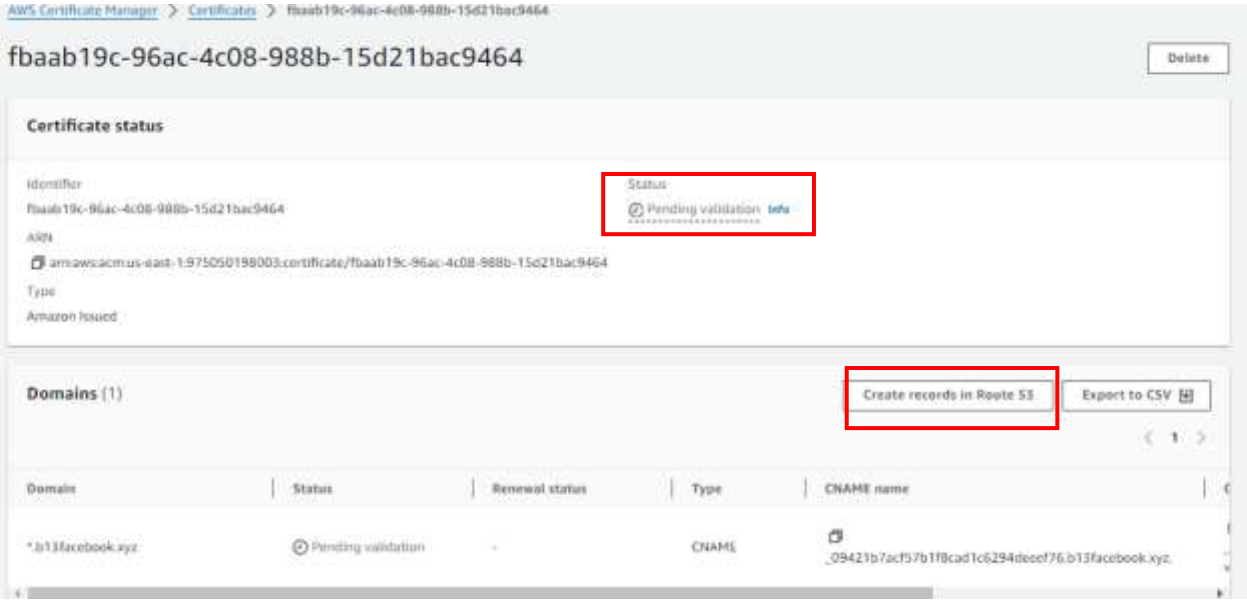
Equivalent in cryptographic strength to RSA 7680.

Tags Info

No tags associated with the resource.

You can add up to 50 tags.

Here you can see the status pending validation. Now we need to add a **CNAME record** in our domain. If you are not using route 53 then you need to add this CNAME record manually by going to your DOMAIN REREGISTER. And if you are using route 53 then click on the button create record in route 53 and click on the create record button



AWS Certificate Manager > Certificates > fbaab19c-96ac-4c08-988b-15d21bac9464

fbaab19c-96ac-4c08-988b-15d21bac9464

Certificate status

Identifier
fbaab19c-96ac-4c08-988b-15d21bac9464

ARN
arn:aws:acm:us-east-1:975050198003:certificate/fbaab19c-96ac-4c08-988b-15d21bac9464

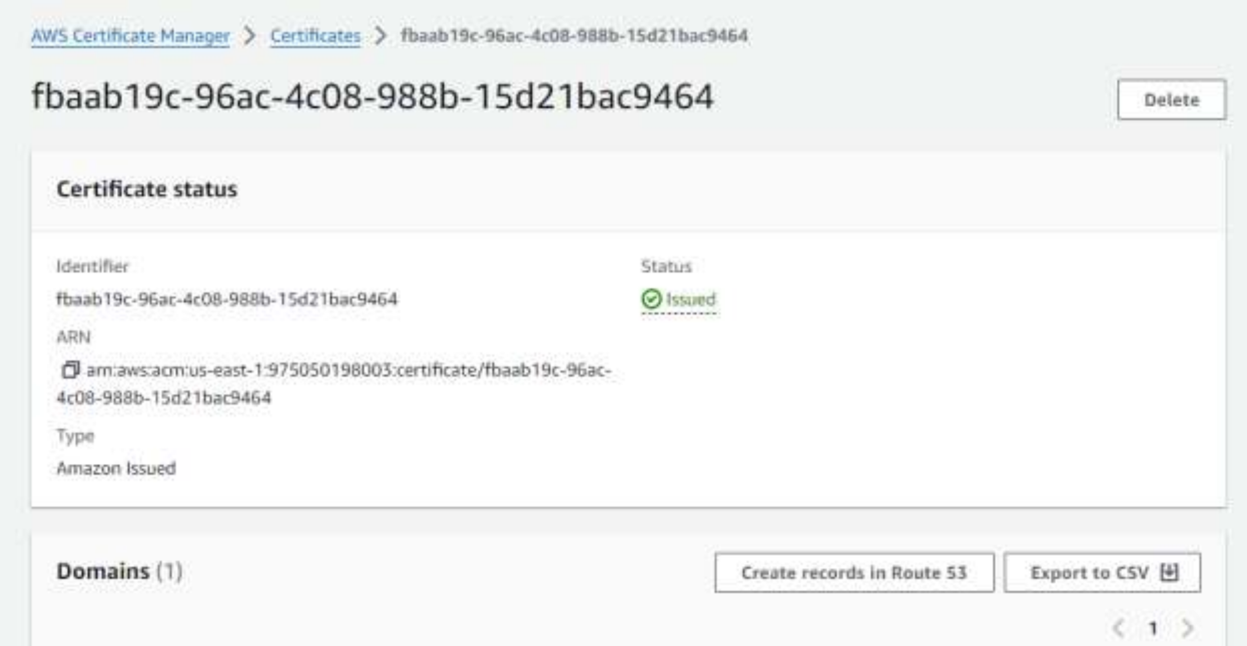
Type
Amazon Issued

Status
Pending validation Info

Domains (1)

Domain	Status	Renewal status	Type	CNAME name
*.b13facebook.xyz	Pending validation	-	CNAME	_09421b7acf57b1f8cad1c6294deeef76.b13facebook.xyz.

And in just a few minutes you will see the status issued.



AWS Certificate Manager > Certificates > fbaab19c-96ac-4c08-988b-15d21bac9464

fbaab19c-96ac-4c08-988b-15d21bac9464

Certificate status

Identifier
fbaab19c-96ac-4c08-988b-15d21bac9464

ARN
arn:aws:acm:us-east-1:975050198003:certificate/fbaab19c-96ac-4c08-988b-15d21bac9464

Type
Amazon Issued

Status
Issued

Domains (1)

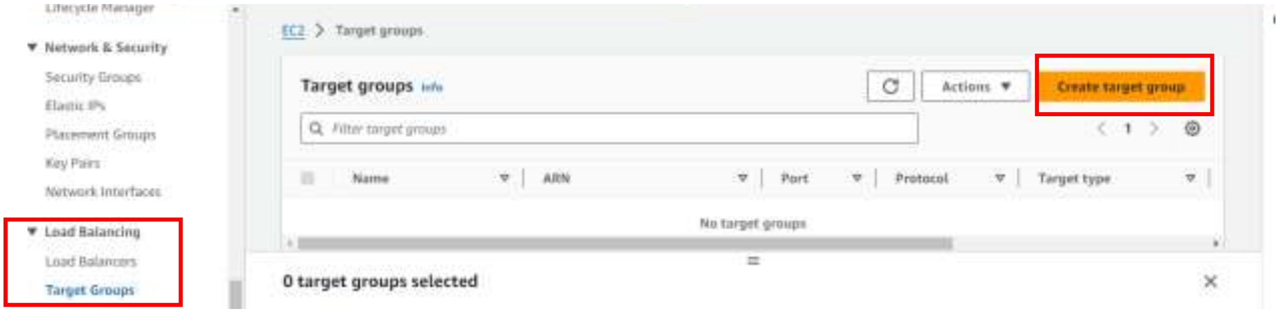
Application Load balancer (ALB) and Route 53

Now it's time to set up an Application load balancer. We need two load balancers, one point to the backend server, and another point to the frontend server.

Type ec2 in the AWS console. And click on the EC2 service.

Note: before we created ALB we need to create a Target group (TG). So first we will create TG for ALB-frontend and then create TG for ALB-backend.

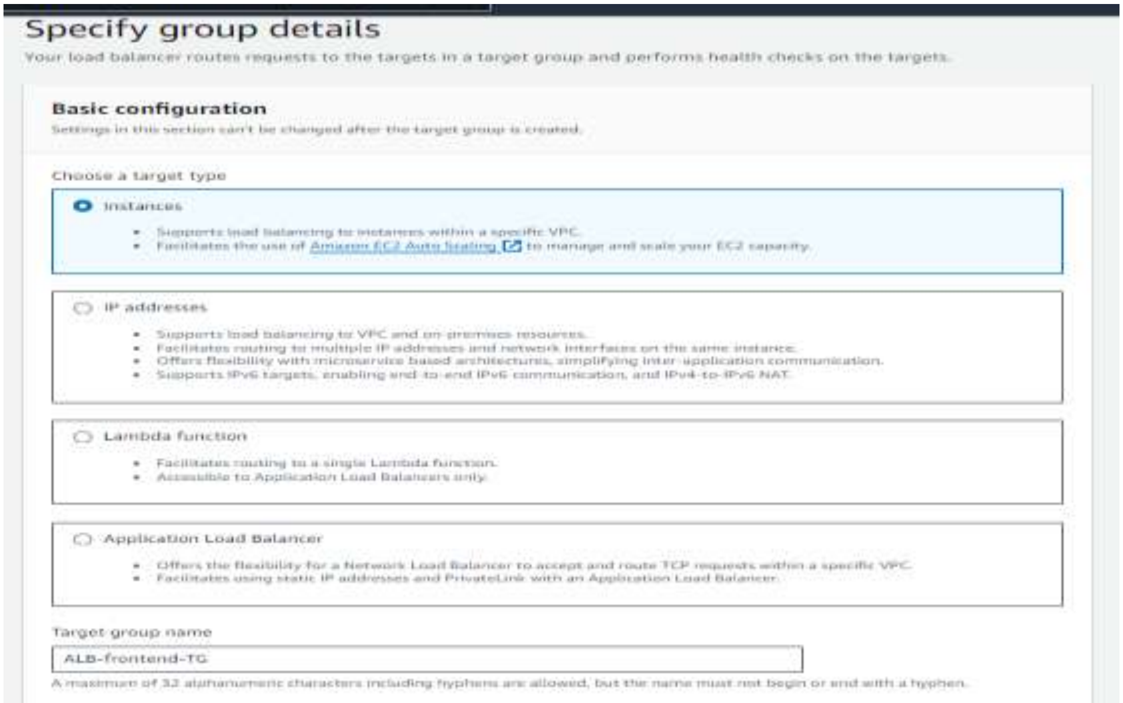
Click the target group button on the bottom of the left panel. And click on the create target group button.



No target groups

0 target groups selected

Here we can configure our TG. Select the instance in the target type. You can give any name to TG but try to give some relevant name such as **ALB-frontend-TG** because we are creating TG for ALB-frontend.



Specify group details

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

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

● Instances
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

○ IP addresses
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

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

○ 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.

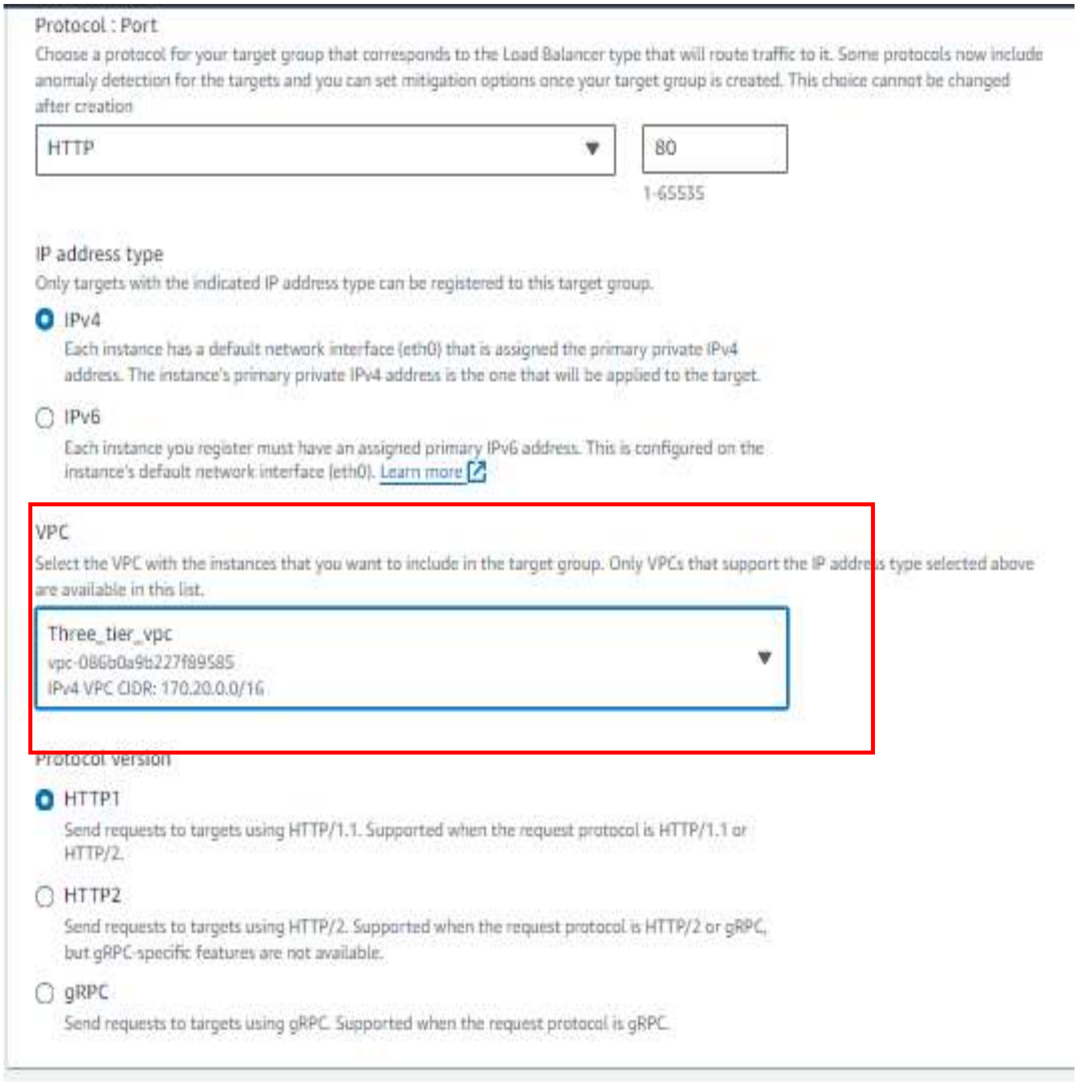
Target group name

ALB-frontend-TG

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

In the VPC section select VPC that we created earlier.

Keep everything as it is, scroll down, and click on the Next button.



Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP ▼ | 80

1-65535

IP address type

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

● IPv4

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.

Three_tier_vpc
vpc-086b0a9b227f89585
IPv4 VPC CIDR: 170.20.0.0/16 ▼

Protocol version

● HTTP1

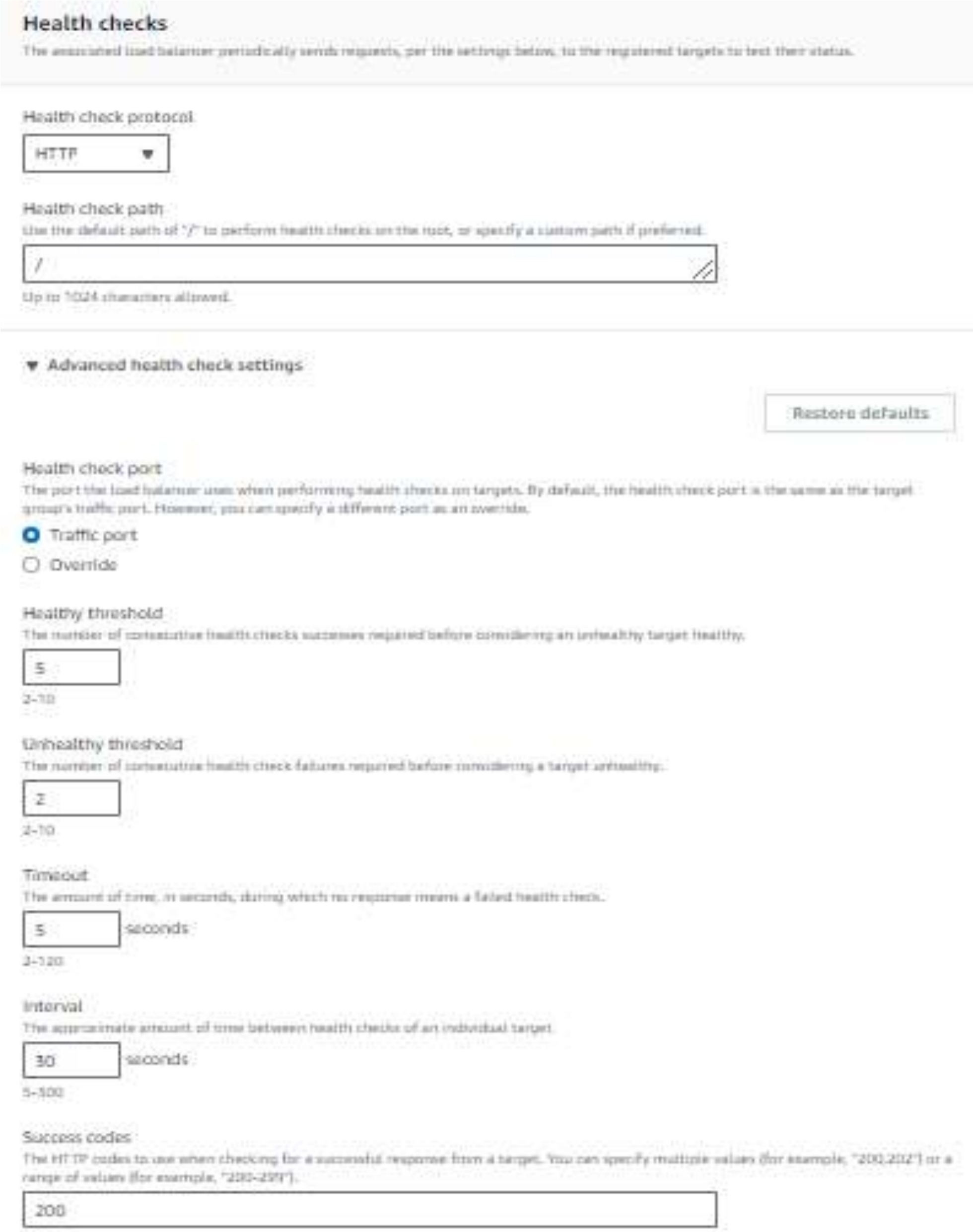
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

○ HTTP2

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

○ gRPC

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



Health checks

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

Health check protocol

HTTP ▼

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

▼ Advanced health check settings

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

● Traffic port

○ Override

Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

5

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

5 seconds

2-120

Interval

The approximate amount of time between health checks of an individual target.

30 seconds

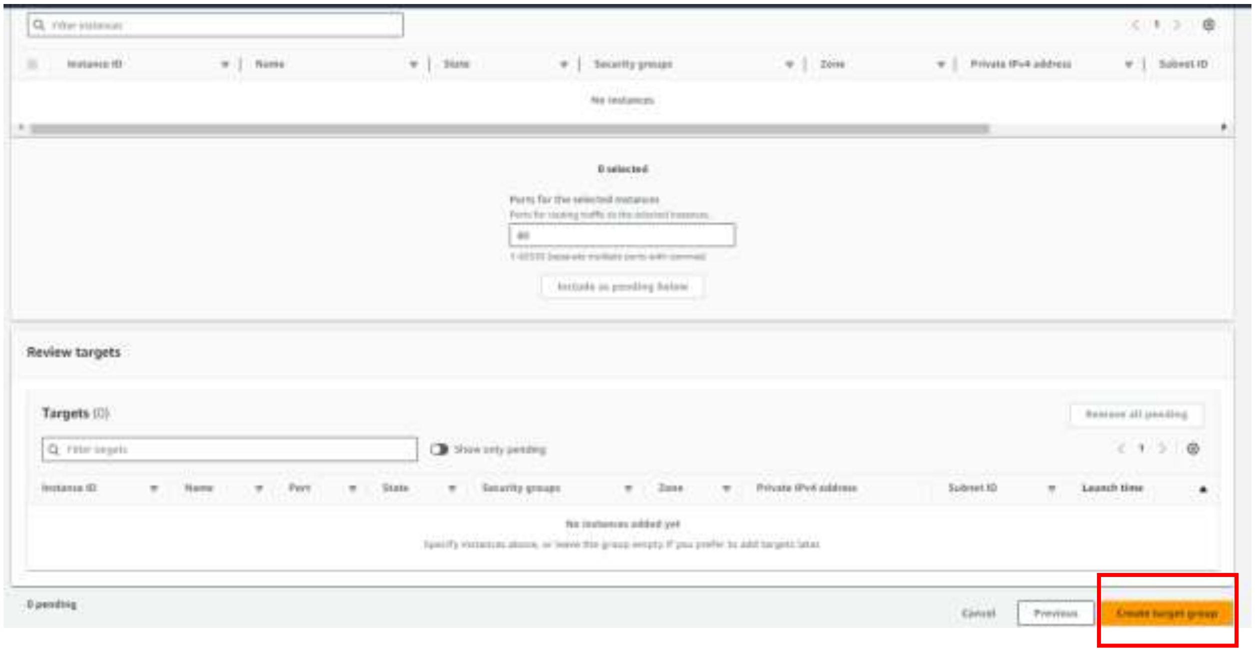
5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

200

Click on the create target group button.



Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID
No instances						

0 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Review targets

Targets (0)

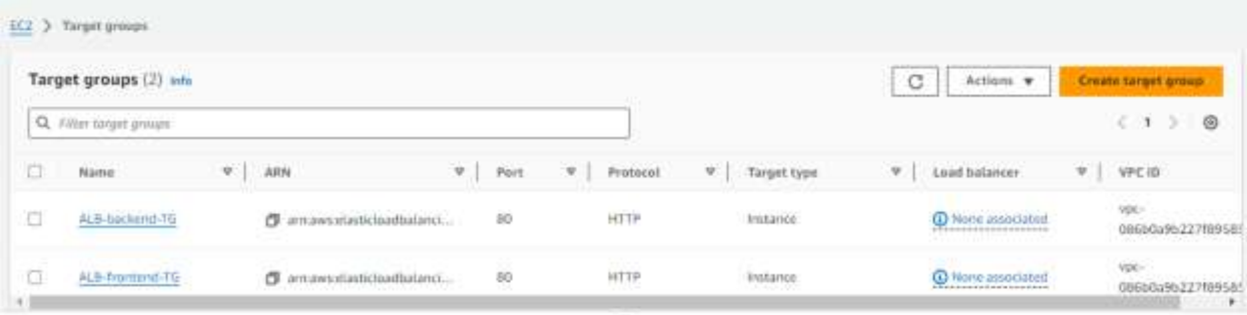
Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
No instances added yet								

Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending

Let's create TG for **ALB-backend**. Follow same as above steps.

So we have two TG. **ALB-frontend-TG and ALB-backend-TG.**

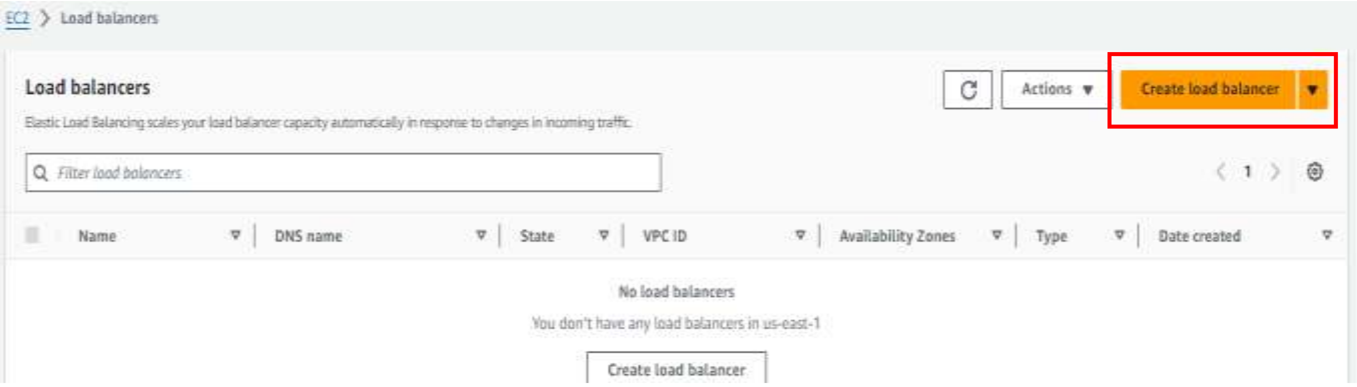


EC2 > Target groups

Target groups (2) Info

	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
□	ALB-backend-TG	arn:aws:elasticloadbalanci...	80	HTTP	Instance	None associated	vpc-086b0a9b227f8958
□	ALB-frontend-TG	arn:aws:elasticloadbalanci...	80	HTTP	Instance	None associated	vpc-086b0a9b227f8958

Now let's associate these TG with the load balancer. So click on the Load Balancer button at the bottom of the left panel and click on the create load balancer button. First, we will create ALB for frontend.



EC2 > Load balancers

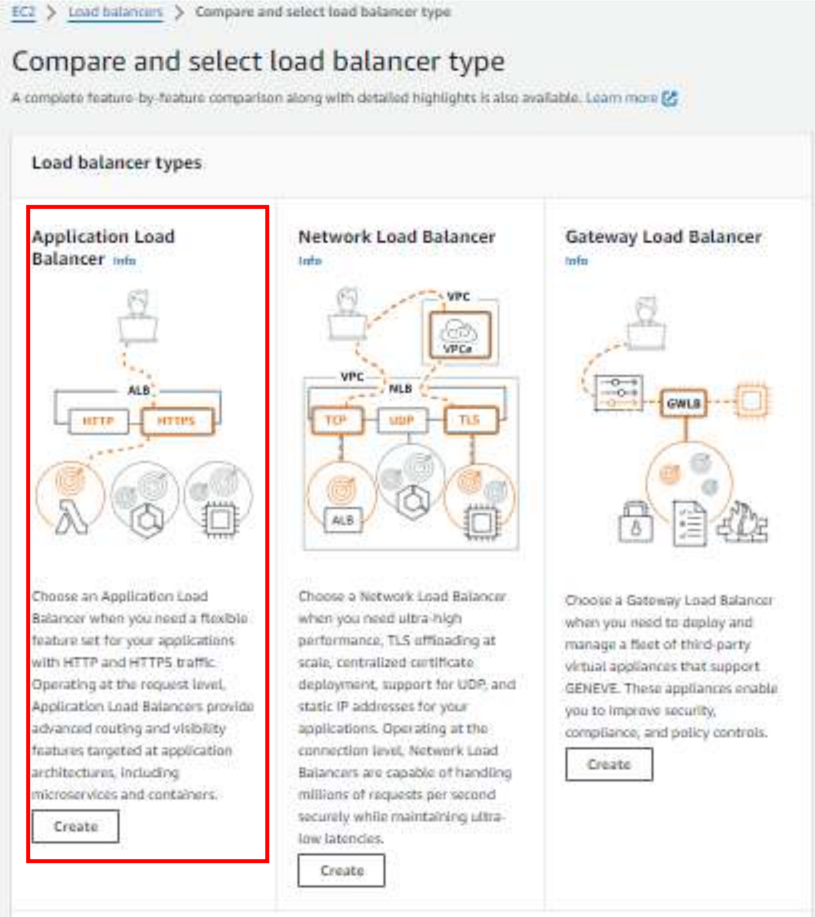
Load balancers

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-east-1

Choose Application load balancer and click on create button.



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 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.

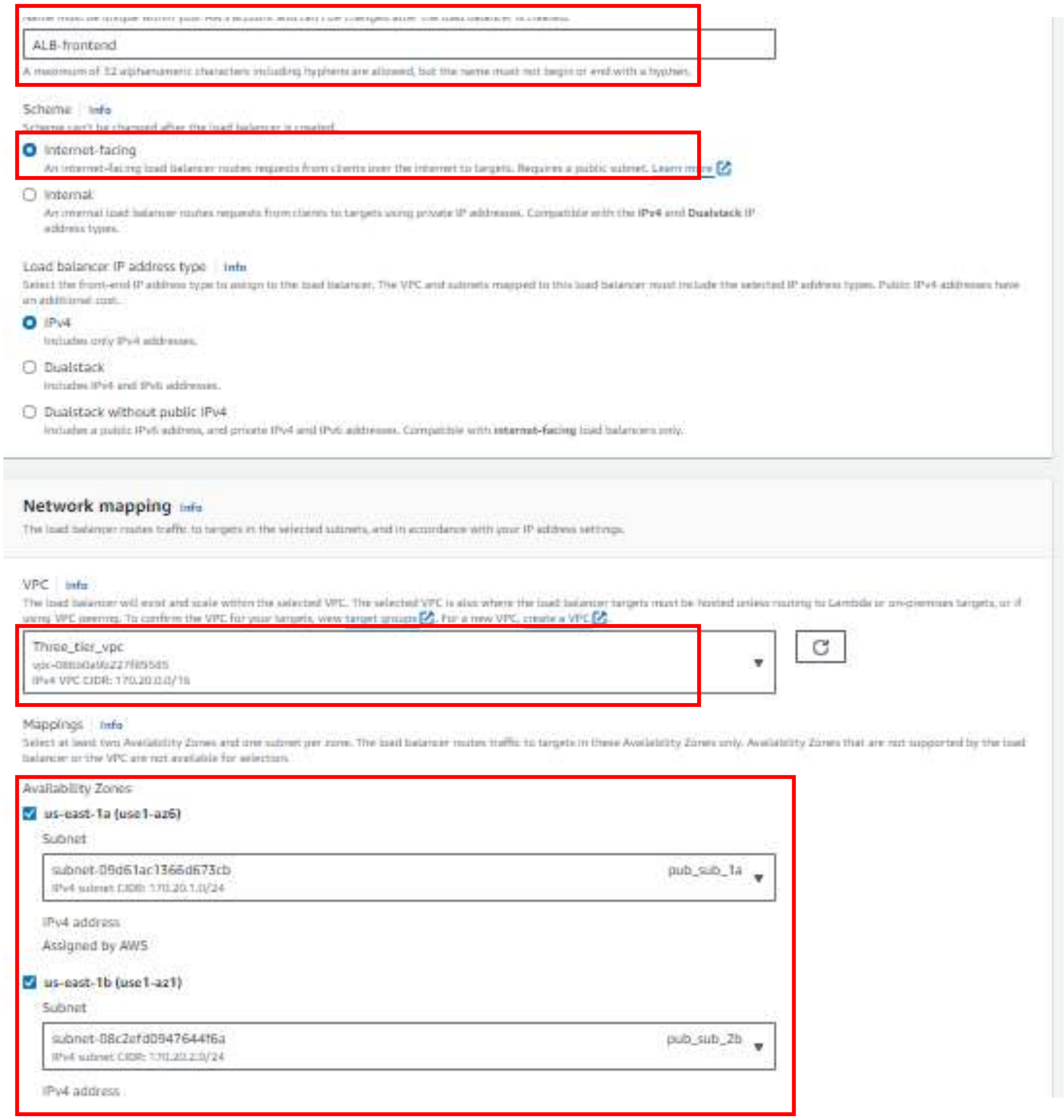
Network Load Balancer Info

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.

Here we can configure our ALB. First, give the relevant name to ALB such as **ALB-frontend**. Select the internet-facing option. In Network mapping select VPC that we have created. Select both availability zone **us-east-1a** and **us-east-2b**. And select subnet **pub-sub-1a** and **pub-sub-2b** respectively.



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

ALB-frontend

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

Scheme | Info

Scheme can't be changed after the load balancer is created.

● Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more

○ Internal
An internal load balancer routes requests from clients to targets using private IP addresses. 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.

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.

Three_tier_vpc
vpc-088b0a9b227f85585
IPv4 VPC CIDR: 170.20.0.0/16

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-east-1a (use1-az6)

Subnet

subnet-09d61ac1366d673cb pub_sub_1a
IPv4 subnet CIDR: 170.20.1.0/24

IPv4 address

Assigned by AWS

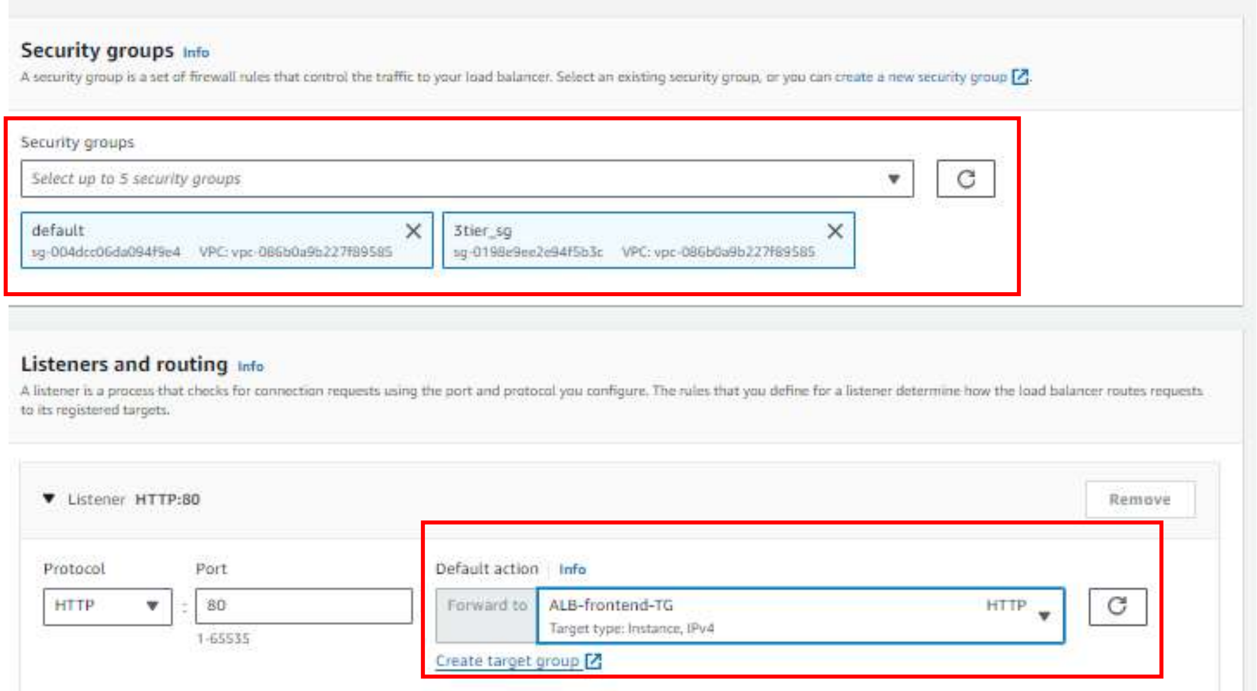
☑ us-east-1b (use1-az1)

Subnet

subnet-08c2efd0947644f6a pub_sub_2b
IPv4 subnet CIDR: 170.20.2.0/24

IPv4 address

Select security group. In the listener part select TG that we have just created **ALB-frontend-TG**.



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

default
sg-004dcc06da094f9e4 VPC: vpc-086b0a9b227f89585

3tier_sg
sg-0198e9ee2e94f5b3c VPC: vpc-086b0a9b227f89585

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

Remove

Protocol: HTTP

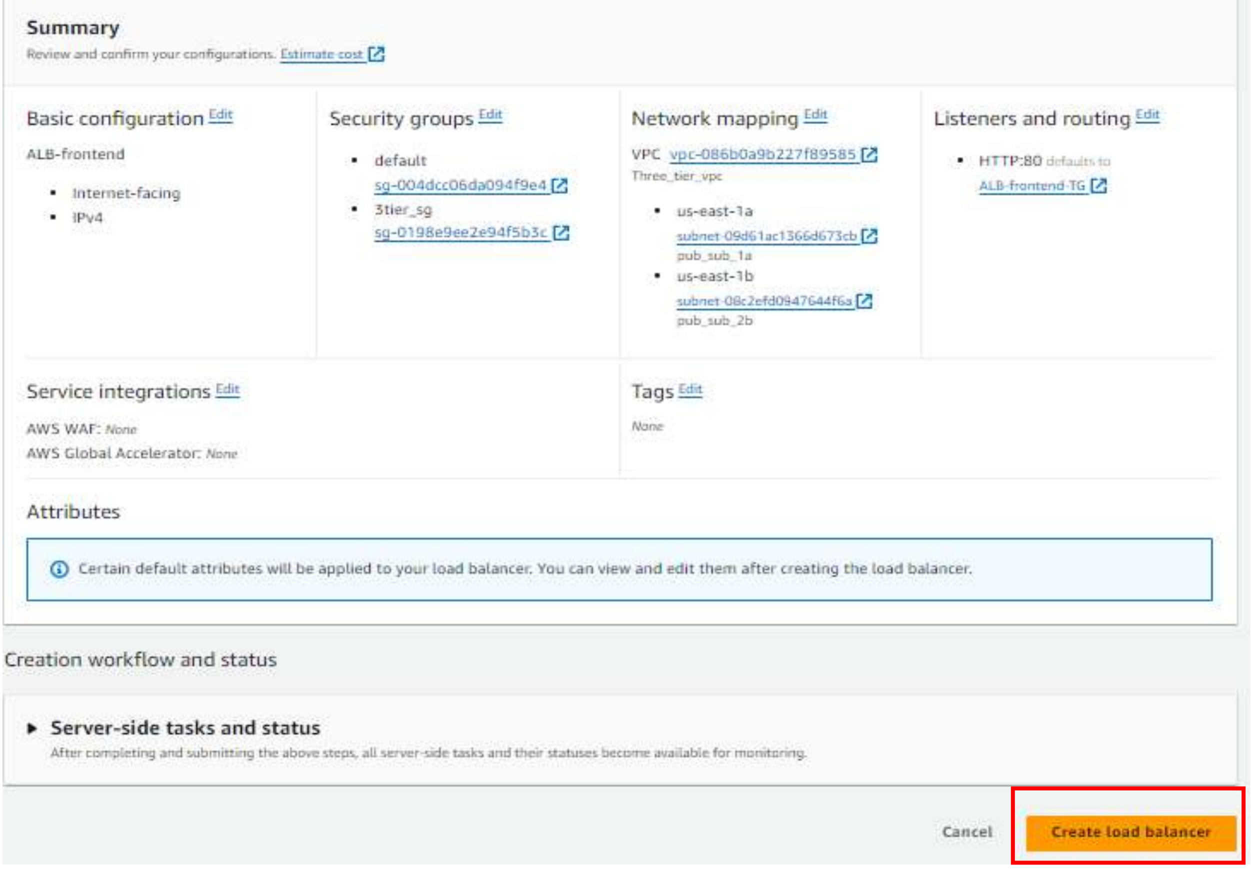
Port: 80
1-65535

Default action Info

Forward to: ALB-frontend-TG HTTP
Target type: Instance, IPv4

Create target group

Scroll down and click on the create load balancer button.



Summary

Review and confirm your configurations. Estimate cost

Basic configuration Edit

ALB-frontend

- Internet-facing
- IPv4

Security groups Edit

- default
  sg-004dcc06da094f9e4
- 3tier_sg
  sg-0198e9ee2e94f5b3c

Network mapping Edit

VPC vpc-086b0a9b227f89585
Three_tier_vpc

- us-east-1a
  subnet-09d61ac1366d673cb
  pub_sub_1a
- us-east-1b
  subnet-08c2efd0947644f6a
  pub_sub_2b

Listeners and routing Edit

- HTTP:80 defaults to
  ALB-frontend-TG

Service integrations Edit

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

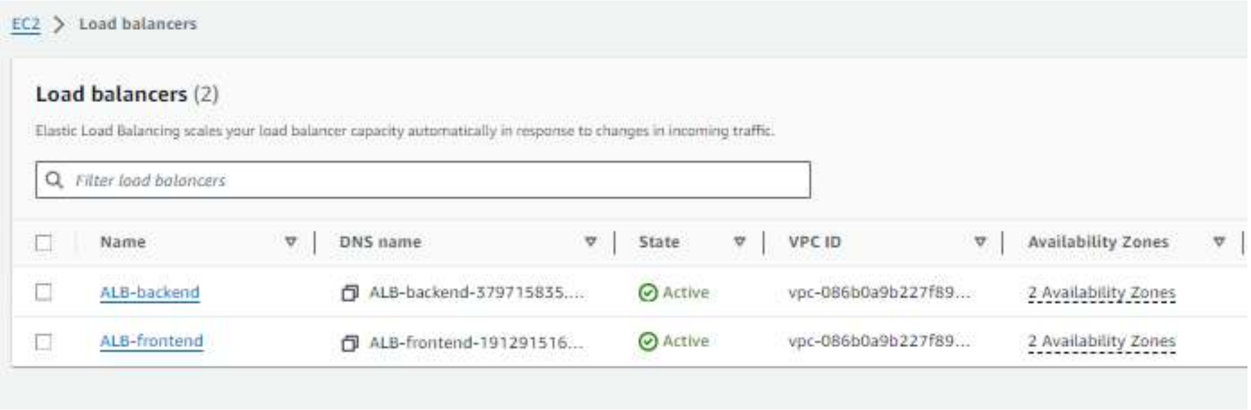
▶ Server-side tasks and status

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

Cancel

Create load balancer

Now, let's create ALB for backend. Follow same above steps, but in the listener part select TG that we just created **ALB-backend-TG**.



EC2 > Load balancers

Load balancers (2)

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

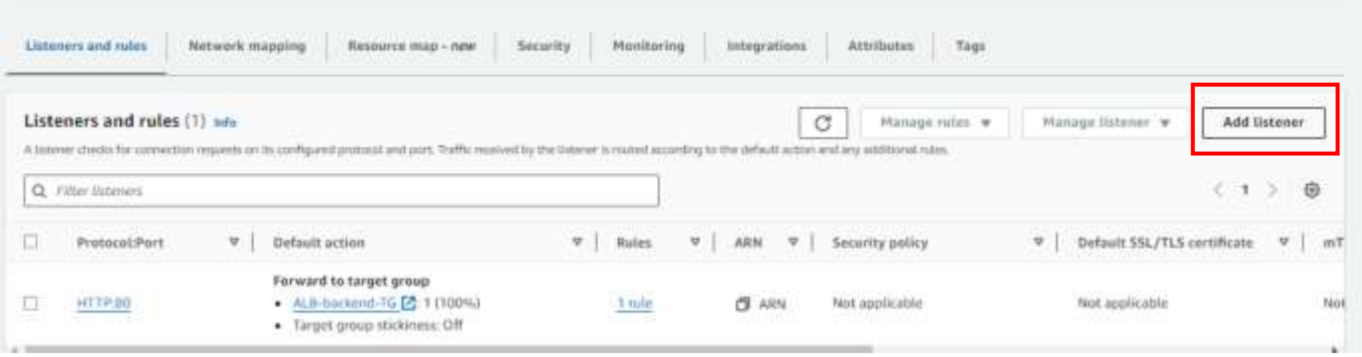
Filter load balancers

□	Name	DNS name	State	VPC ID	Availability Zones
□	ALB-backend	ALB-backend-379715835....	Active	vpc-086b0a9b227f89...	2 Availability Zones
□	ALB-frontend	ALB-frontend-191291516...	Active	vpc-086b0a9b227f89...	2 Availability Zones

Now we have two load balancers, **ALB-frontend and ALB-backend**. But we need to add one more listener in **ALB-frontend and ALB-backend**.

So click on ALB-backend.

Click on add listener the button that is located on the right side.



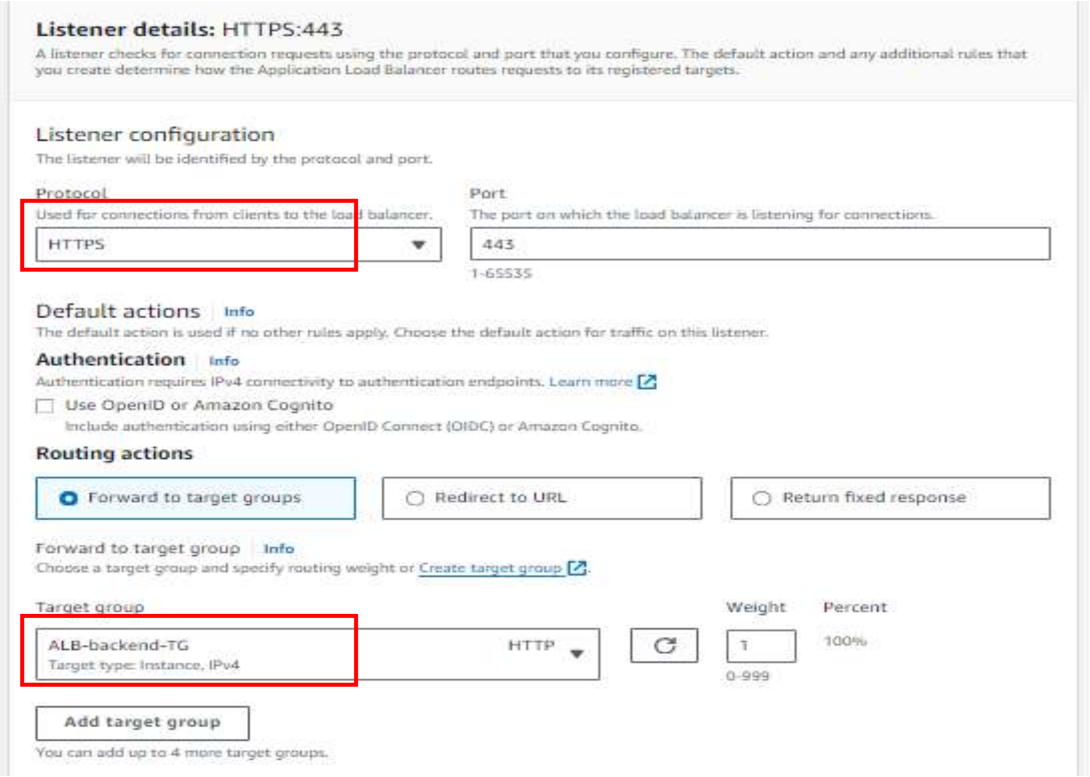
Listeners and rules (1) Info

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

Filter listeners

□	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mT
□	HTTP:80	Forward to target group • ALB-backend-TG: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not

Here In listener details select HTTPS. Default Action should be forward and select ALB-backend-TG. Now we need to select the certificate that we have created. So in the Secure Listener setting select the certificate. And click on the add button below.



Listener details: HTTPS:443

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

Listener configuration

The listener will be identified by the protocol and port.

Protocol

Used for connections from clients to the load balancer.

HTTPS

Port

The port on which the load balancer is listening for connections.

443

1-65535

Default actions Info

The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Authentication Info

Authentication requires IPv4 connectivity to authentication endpoints. Learn more

□ Use OpenID or Amazon Cognito

Include authentication using either OpenID Connect (OIDC) or Amazon Cognito.

Routing actions

● Forward to target groups ○ Redirect to URL ○ Return fixed response

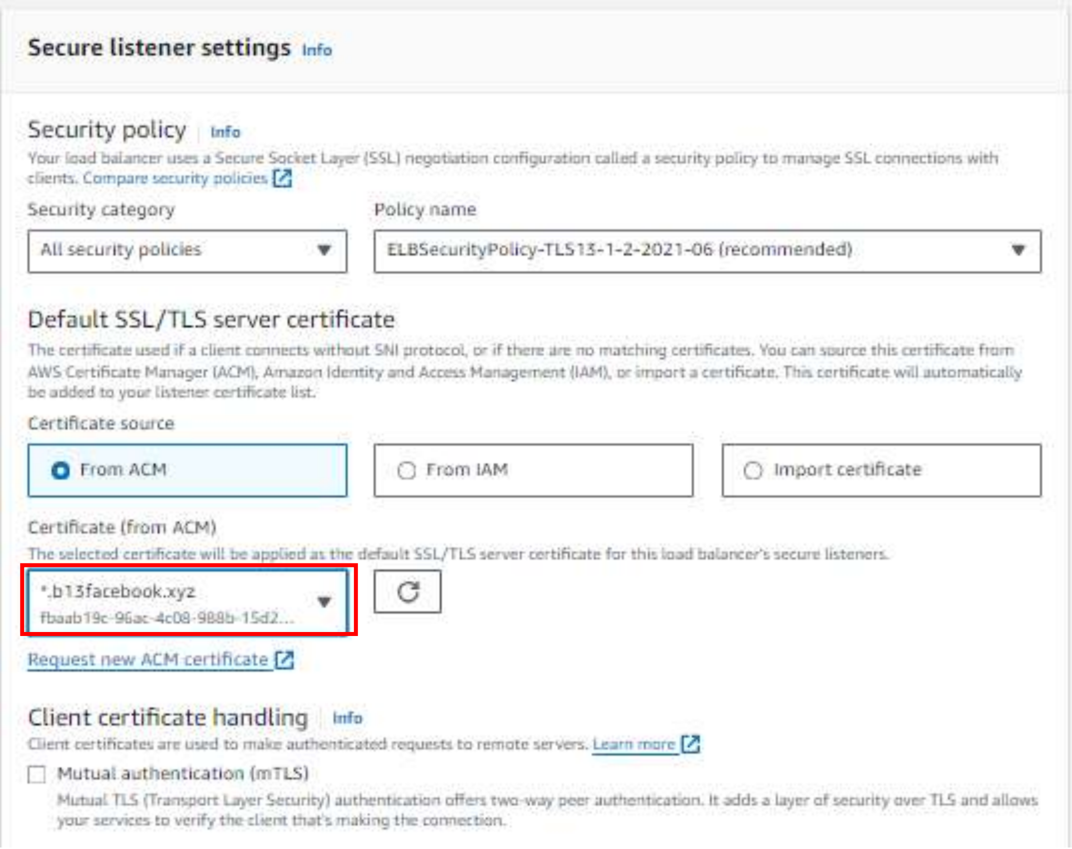
Forward to target group Info

Choose a target group and specify routing weight or Create target group.

Target group		Weight	Percent
ALB-backend-TG Target type: Instance, IPv4	HTTP	1 0-999	100%

Add target group

You can add up to 4 more target groups.



Secure listener settings Info

Security policy Info

Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration called a security policy to manage SSL connections with clients. Compare security policies

Security category

All security policies

Policy name

ELBSecurityPolicy-TLS13-1-2-2021-06 (recommended)

Default SSL/TLS server certificate

The certificate used if a client connects without SNI protocol, or if there are no matching certificates. You can source this certificate from AWS Certificate Manager (ACM), Amazon Identity and Access Management (IAM), or import a certificate. This certificate will automatically be added to your listener certificate list.

Certificate source

● From ACM ○ From IAM ○ Import certificate

Certificate (from ACM)

The selected certificate will be applied as the default SSL/TLS server certificate for this load balancer's secure listeners.

*.b13facebook.xyz

fbaab19c-96ac-4c08-988b-15d2...

Request new ACM certificate

Client certificate handling Info

Client certificates are used to make authenticated requests to remote servers. Learn more

□ Mutual authentication (mTLS)

Mutual TLS (Transport Layer Security) authentication offers two-way peer authentication. It adds a layer of security over TLS and allows your services to verify the client that's making the connection.

Follow same steps for **ALB-frontend** also.

So here we successfully completed the ALB setup

EC2

Now we are going to create a temporary frontend and backend server to do all the required setup, take snapshots and create Machine images from it. So that we can utilize it in the launch template.

1. First, click on the instance button and then click on the Launch Instance button on the top right corner.
2. First, we are going to set up a frontend server. Give a name to your instance **(temp-frontend-server)**. Select Ubuntu as the operating system. Choose the instance type as t2.micro. Click on Create key pair if you don't have it.
3. Here we are doing a temporary setup so we don't use our OWN VPC. We can use the default VPC given by AWS. In short, keep the Network setting as it is.

We have successfully launched **temp-frontend-server**. So now let's launch a temporary backend server. Give a name to your instance **(temp-backend-server)**.Follow above steps

Please wait for 5-8 minutes so that the instance comes in a running state. And then we will utilize instances for further steps.

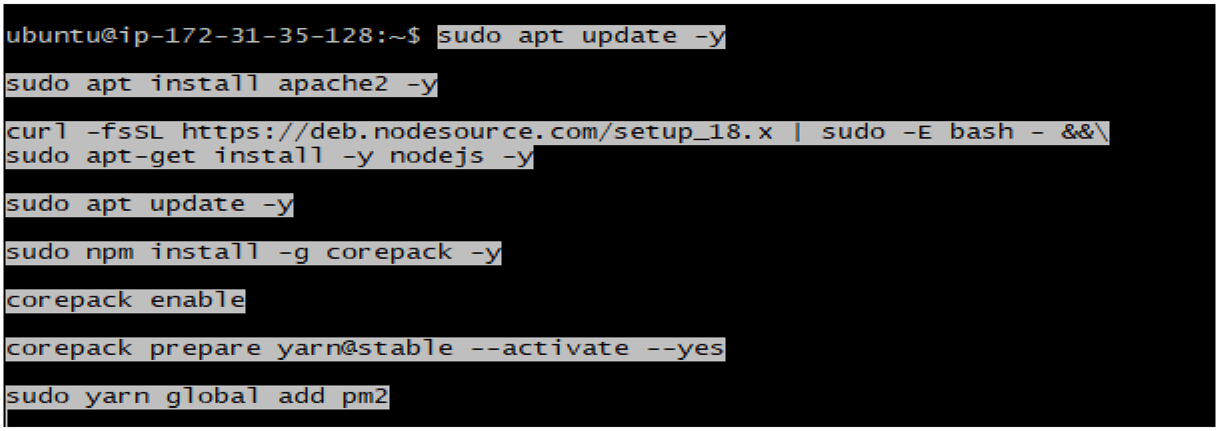
Temp-frontend-server:

Select **temp-frontend-server**. Now open Gitbash where you have downloaded your YOUR_KEY.pem file. And type the command.

ssh -i <name_of_key>.pem ubuntu@<Public_IP_add_of_Instance>

Now you are successfully logged your remote **temp-frontend-server**. Now our first task is to install some packages and second task is to clone git repo.

```
#packages for our frontend server
#!/bin/bash

sudo apt update -y

sudo apt install apache2 -y

curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
sudo apt-get install -y nodejs -y
sudo apt update -y

sudo npm install -g corepack -y

corepack enable

corepack prepare yarn@stable --activate

sudo yarn global add pm2
```



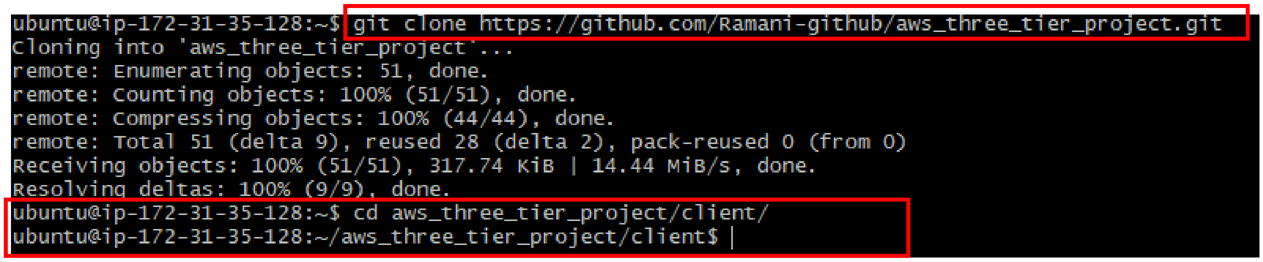
```
ubuntu@ip-172-31-35-128:~$ sudo apt update -y

sudo apt install apache2 -y

curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
sudo apt-get install -y nodejs -y

sudo apt update -y

sudo npm install -g corepack -y

corepack enable

corepack prepare yarn@stable --activate --yes

sudo yarn global add pm2
```

The Github repository link is https://github.com/Ramani-github/aws_three_tier_project.git

```
git clone https://github.com/Ramani-github/aws_three_tier_project.git
```

Go inside the directory.

```
cd aws_three_tier_project/client
```

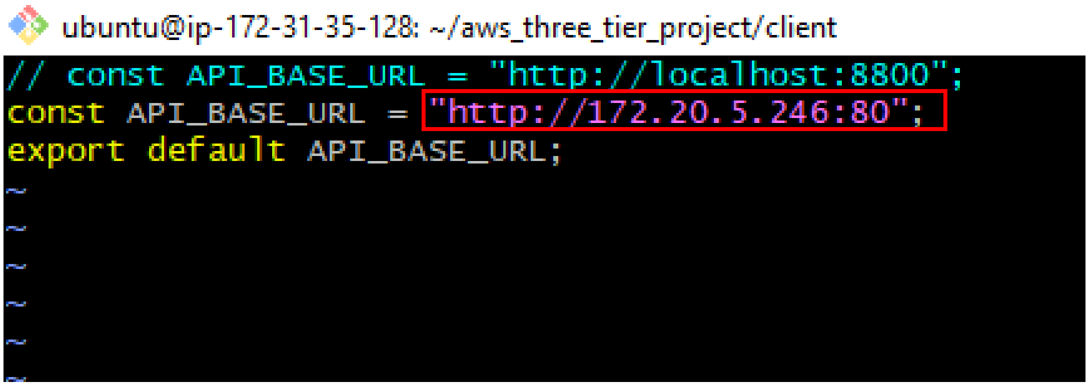


```
ubuntu@ip-172-31-35-128:~$ git clone https://github.com/Ramani-github/aws_three_tier_project.git
Cloning into 'aws_three_tier_project'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 51 (delta 9), reused 28 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (51/51), 317.74 KiB | 14.44 MiB/s, done.
Resolving deltas: 100% (9/9), done.
ubuntu@ip-172-31-35-128:~$ cd aws_three_tier_project/client/
ubuntu@ip-172-31-35-128:~/aws_three_tier_project/client$ |
```

Now, we need to change just one line in our frontend application that is built in React. So type the command

```
vim src/pages/config.js
```

The above command opens the file in a text editor. Now press esc + I the button on your keyboard to edit the file. In this file, we have to change API_BASE_URL. So remove whatever is present in the API_BASE_URL variable.

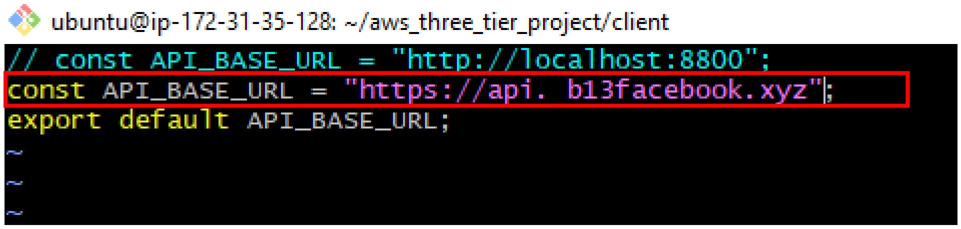


ubuntu@ip-172-31-35-128: ~/aws_three_tier_project/client

```
// const API_BASE_URL = "http://localhost:8800";
const API_BASE_URL = "http://172.20.5.246:80";
export default API_BASE_URL;
~
~
~
~
~
~
```

And add **https://api. b13facebook.xyz**, in my case I have added this URL but in your case it is different. This means you need to use your OWN domain name. So your API_BASE_URL should be like https://api.<YOUR_DOMAIN_NAME>.XYZ. After updating the variable press ESC key on your keyboard and then type: wq and hit the Enter button.

API_BASE_URL = https://api.b13facebook.xyz



ubuntu@ip-172-31-35-128: ~/aws_three_tier_project/client

```
// const API_BASE_URL = "http://localhost:8800";
const API_BASE_URL = "https://api. b13facebook.xyz";
export default API_BASE_URL;
~
~
~
```

After making these changes our frontend of the application will send all the API calls on the domain name https://api.b13facebook.xyz and lastly, that will point to our backend server.

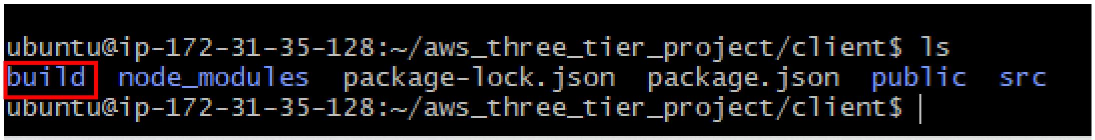
Now type the command npm install in the terminal to install all the required packages.

```
npm install
```

Type the command npm run build to create the optimize static pages.

```
npm run build
```

Now you have one more folder in the directory called build. You can verify that by tying ls command0



```
ubuntu@ip-172-31-35-128:~/aws_three_tier_project/client$ ls
build  node_modules  package-lock.json  package.json  public  src
ubuntu@ip-172-31-35-128:~/aws_three_tier_project/client$ |
```

Now type the very essential command sudo cp -r build/* /var/ww/html/

```
sudo cp -r build/* /var/www/html
```

The above command takes all the static files from the build folder and stores them in /var/www/html so that Apache can serve them.

Here our temp-frontend-server configuration is completed.

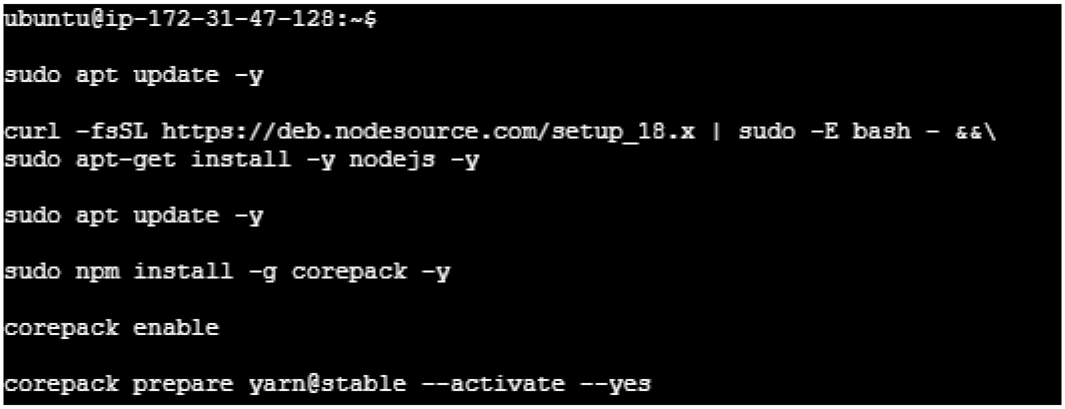
Temp-backend-server

Now let's set up the temp-backend-server. So select the temp-backend-server and copy the IP address of the instance. Again please open Git bash in the same directory where your stored key.pem file. And type the below command

ssh -i name_of_your_key>.pem ubuntu@<Public_IP_add>

We are successfully logged in inside the backend server. First, install packages and we will clone the repo.

```
#!/bin/bash

sudo apt update -y

curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
sudo apt-get install -y nodejs -y

sudo apt update -y

sudo npm install -g corepack -y

corepack enable

corepack prepare yarn@stable --activate

sudo yarn global add pm2
```

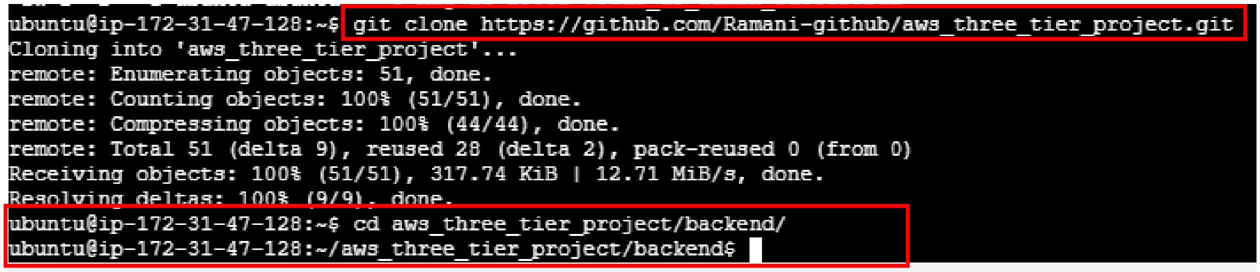


```
ubuntu@ip-172-31-47-128:~$

sudo apt update -y

curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
sudo apt-get install -y nodejs -y

sudo apt update -y

sudo npm install -g corepack -y

corepack enable

corepack prepare yarn@stable --activate --yes
```

git clone https://github.com/Ramani-github/aws_three_tier_project.git

go inside the aws_three_tier_project/backend

cd aws_three_tier_project/backend



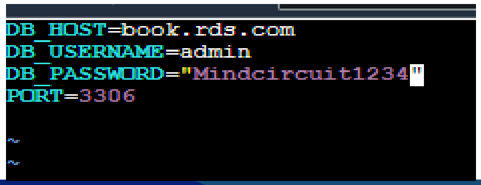
```
ubuntu@ip-172-31-47-128:~$ git clone https://github.com/Ramani-github/aws_three_tier_project.git
Cloning into 'aws_three_tier_project'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 51 (delta 9), reused 28 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (51/51), 317.74 KiB | 12.71 MiB/s, done.
Resolving deltas: 100% (9/9), done.
ubuntu@ip-172-31-47-128:~$ cd aws_three_tier_project/backend/
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$
```

Here we are going to create one file with the name .env

vim .env

Press the esc I button on your keyboard. And copy the code given below and paste the snippet into the code editor. This code contains information about the RDS instance. Please change your username and password according to whatever you kept while creating a database. And then click on the ESC button and type: wq and hit the enter button

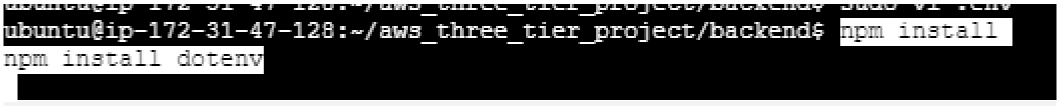
```
DB_HOST=book.rds.com

DB_USERNAME=admin

DB_PASSWORD="Mindcircuit1234"

PORT=3306
```



```
DB_HOST=book.rds.com
DB_USERNAME=admin
DB_PASSWORD="Mindcircuit1234"
PORT=3306

~
~
```

Now type the below commands in terminal

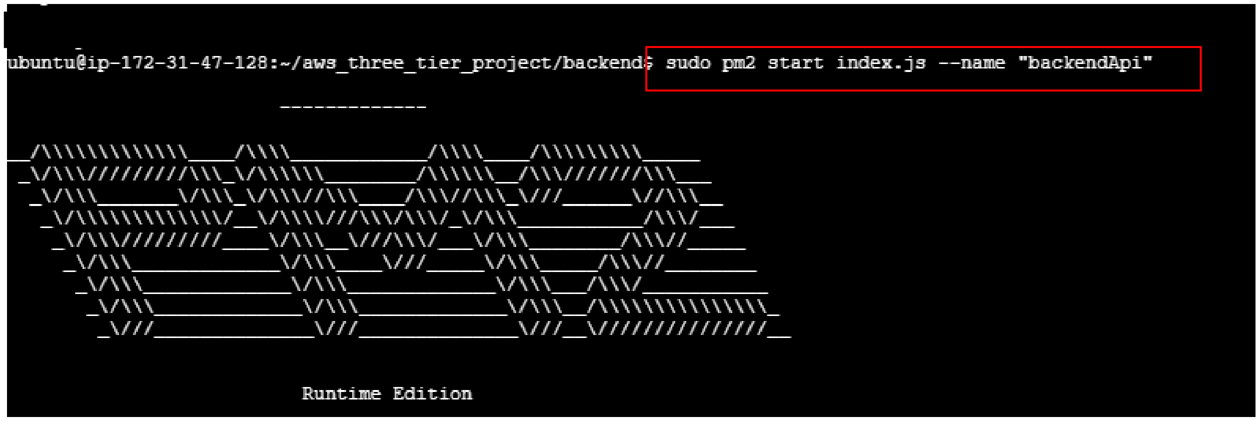
```
npm install
npm install dotenv
```



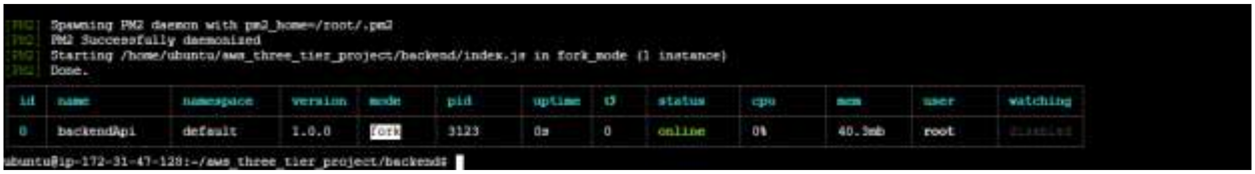
```
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$ sudo vi .env
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$ npm install
npm install dotenv
```

Now, let's start the backend server. (***Very IMP***)

```
sudo pm2 start index.js --name "backendApi"
```



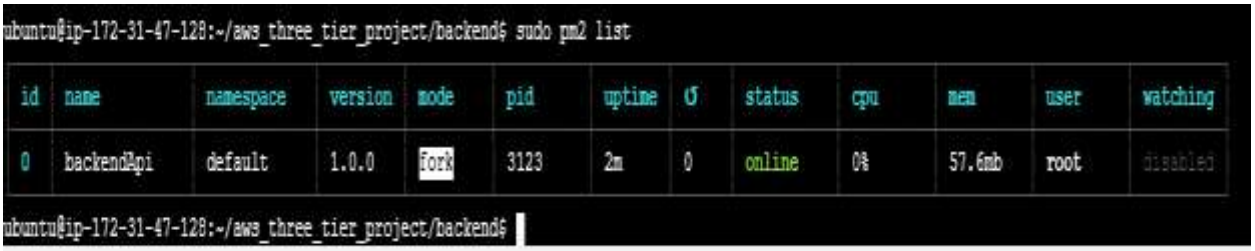
```
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$ sudo pm2 start index.js --name "backendApi"

                        -------------

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                          Runtime Edition
```



```
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/ubuntu/aws_three_tier_project/backend/index.js in fork_mode (1 instance)
[PM2] Done.
│ id │ name        │ namespace   │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
│ 0  │ backendApi  │ default     │ 1.0.0   │ fork    │ 3123     │ 0s     │ 0    │ online    │ 0%       │ 40.3mb   │ root     │ disabled │
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$
```

You can verify that by typing the command

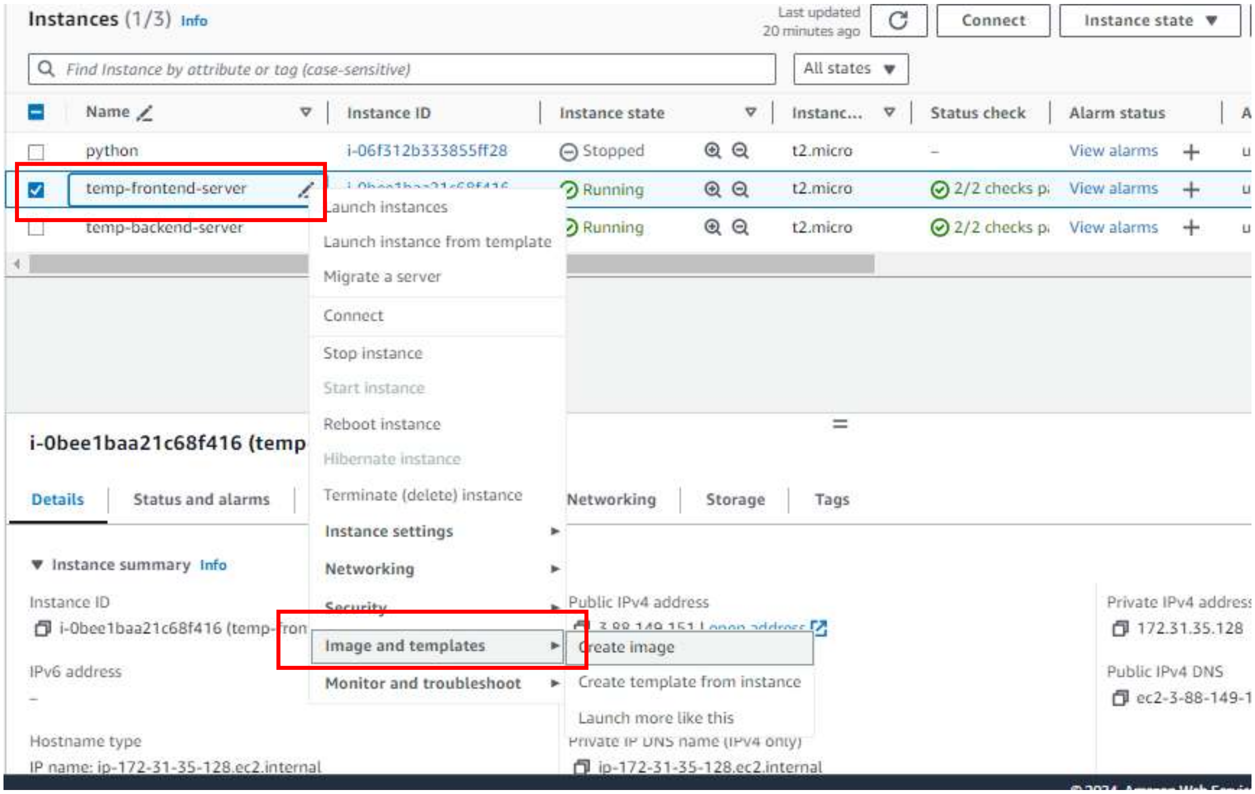
```
sudo pm2 list
```



```
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$ sudo pm2 list
│ id │ name        │ namespace   │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
│ 0  │ backendApi  │ default     │ 1.0.0   │ fork    │ 3123     │ 2m     │ 0    │ online    │ 0%       │ 57.6mb   │ root     │ disabled │
ubuntu@ip-172-31-47-128:~/aws_three_tier_project/backend$
```

Now we have to create Machine images of these servers so that we can create a launch template.

So please select temp-frontend-server and click on the Action button in the top right corner. One drop-down menu will open. You have to select the images and template option and that will give one more drop-down menu from which we need to click on create image button.



Instances (1/3) Info

Last updated 20 minutes ago

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	A
□	python	i-06f312b333855ff28	Stopped	t2.micro	–	View alarms	u
☑	temp-frontend-server	i-0bee1baa21c68f416	Running	t2.micro	2/2 checks pa	View alarms	u
□	temp-backend-server		Running	t2.micro	2/2 checks pa	View alarms	u

i-0bee1baa21c68f416 (temp

Instance summary Info

Instance ID
i-0bee1baa21c68f416 (temp-fron

IPv6 address
–

Hostname type
IP name: ip-172-31-35-128.ec2.internal

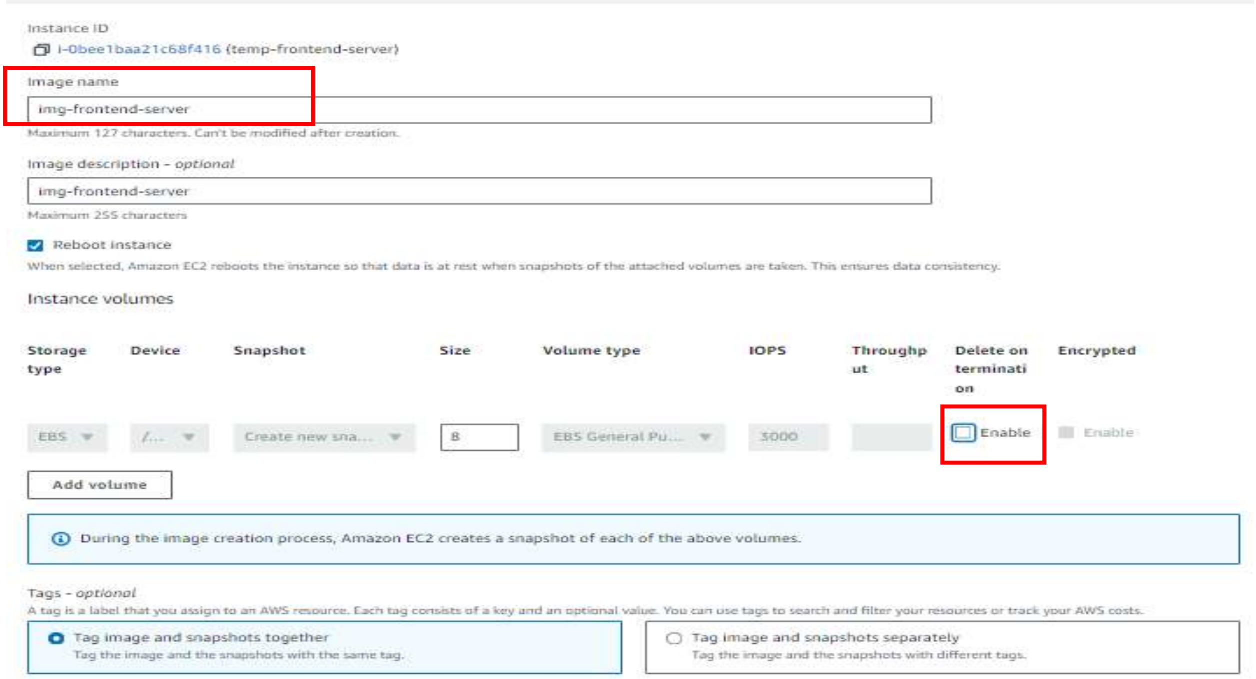
Public IPv4 address
3.88.149.151 | open address

Private IP DNS name (IPv4 only)
ip-172-31-35-128.ec2.internal

Private IPv4 address
172.31.35.128

Public IPv4 DNS
ec2-3-88-149-1

Give the name you your image (**img-frontend-server**). Just deselect that delete on the termination button and click on the create image button.



Instance ID
i-0bee1baa21c68f416 (temp-frontend-server)

Image name
img-frontend-server
Maximum 127 characters. Can't be modified after creation.

Image description - *optional*
img-frontend-server
Maximum 255 characters

☑ Reboot instance
When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/...	Create new sna...	8	EBS General Pu...	3000		□ Enable	□ Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

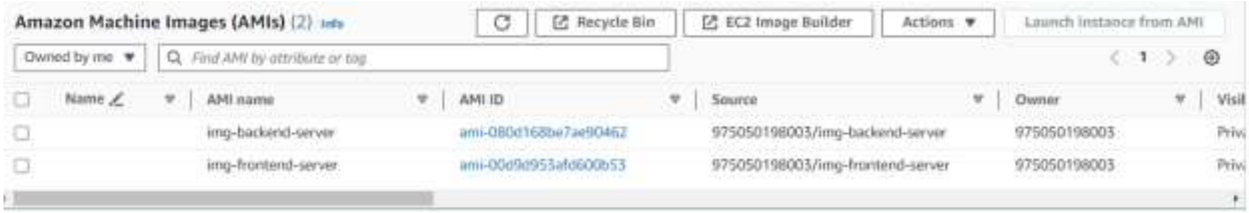
Tags - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

● Tag image and snapshots together
Tag the image and snapshots with the same tag.

○ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

You have to do the same thing for the temp-backend-server as well.

After a couple of minutes (10-15) you can see those images. Click on the AMIs button on the left panel and you can see both images here.



Amazon Machine Images (AMIs) (2) Info

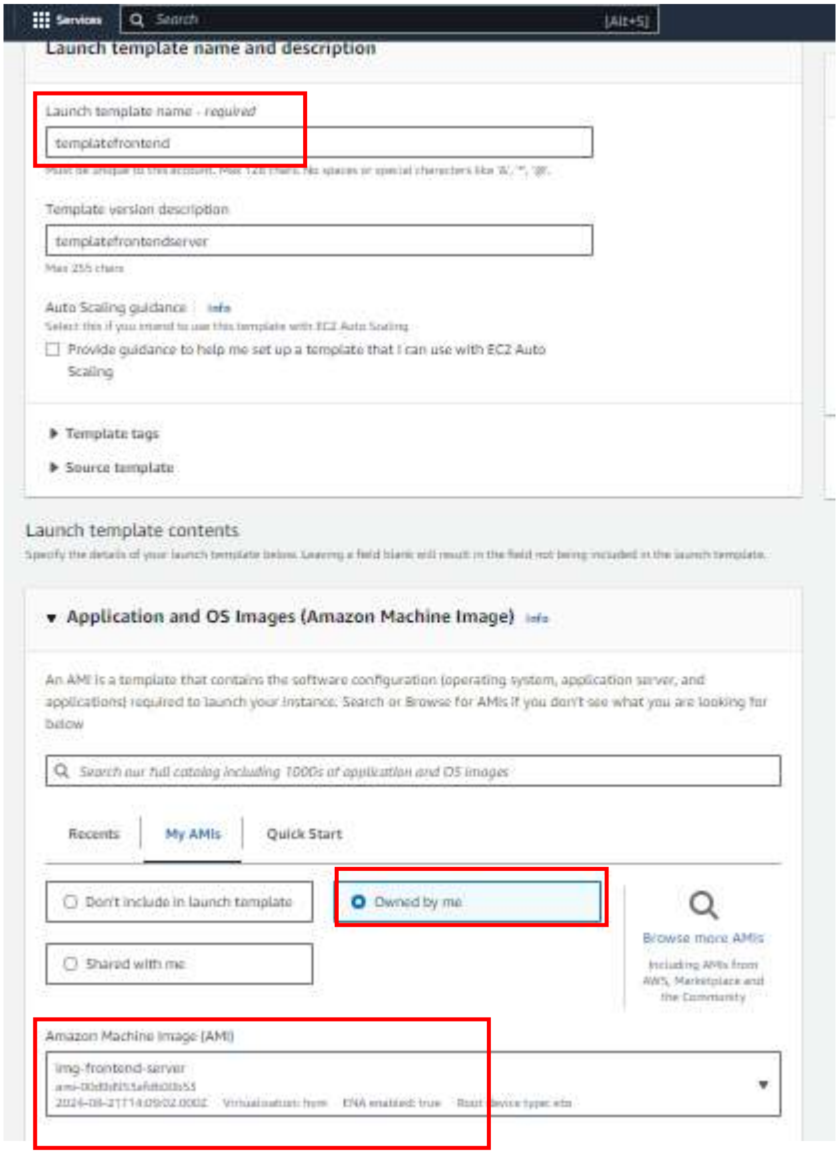
Owned by me ▼ | Find AMI by attribute or tag

☐	Name	AMI name	AMI ID	Source	Owner	Visil
☐		img-backend-server	ami-080d168be7ae90462	975050198003/img-backend-server	975050198003	Priv
☐		img-frontend-server	ami-00d9d953afd600b53	975050198003/img-frontend-server	975050198003	Priv

Launch Template

Create a launch template, so click on the launch template button on the left panel and click on the create launch template button.

Give the name to your launch template such as template-frontend-server as we are creating a launch template for frontend-server. Here we need to select AMI so click on My AMIs tab and select the option owned by me. So now it will show you all the images that are present in your current region. Here you have to select the image that contains the frontend application. Select instance type t2.micro



Launch template name and description

Launch template name - *required*

templatefrontend

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

Template version description

templatefrontendserver

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

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ 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

○ Don't include in launch template

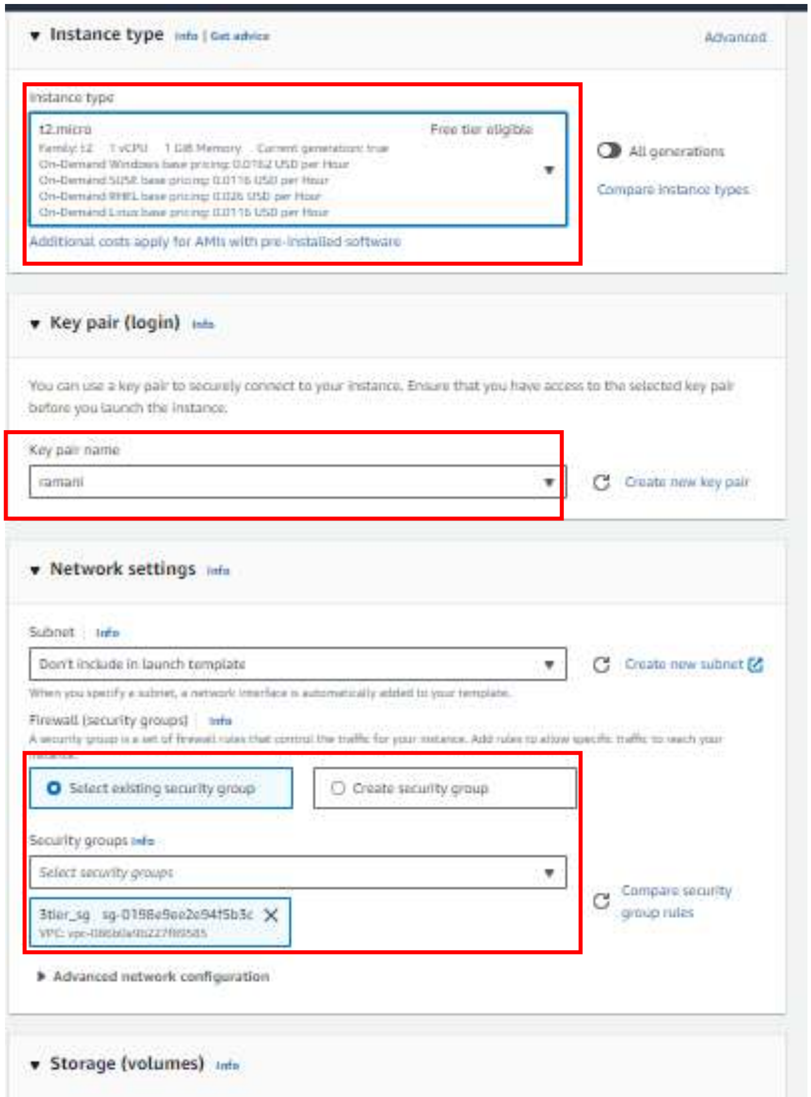
● Owned by me

○ Shared with me

Amazon Machine Image (AMI)

img-frontend-server
ami-00d3bf15fa8b00b53
2024-08-21T14:09:02.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Scroll down, attach the key pair, and in the network setting just select the security group that we created for the frontend server.



▾ 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 Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software

All generations
Compare instance types

▾ Key pair (login) Info

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

Key pair name: ramani

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 groups Info: Select security groups

3tier_sg sg-0198e9ee2e94f5b3c ✕
VPC: vpc-0865b0a9b227f69585

Compare security group rules

▸ Advanced network configuration

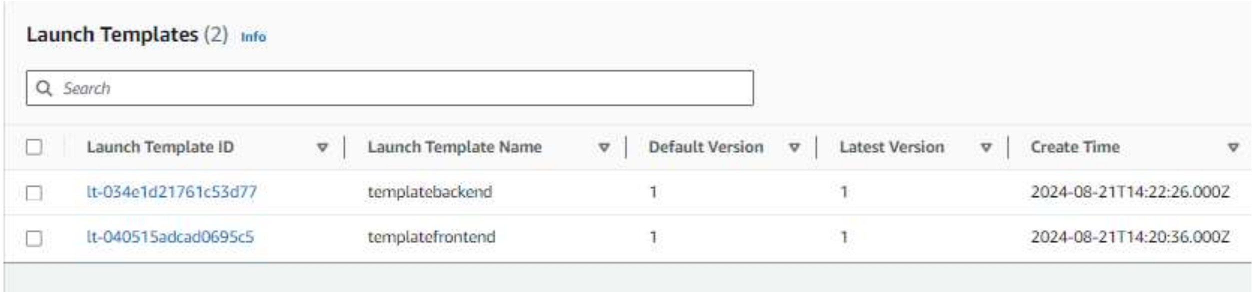
▾ Storage (volumes) Info

We successfully created a launch template for the frontend-server. Now let's create a launch template for the backend server.

Give a name to your launch template (template-backend-server). Give version 1 in the version field, but make you select the correct AMI that holding your backend application. And Select an instance type t2.micro

Select the key pair, and in the network setting just select the security group that we have created.

And then click on the Create launch template button.
We have created two launch templates, template-frontend-server and template-backend-server



Launch Templates (2) Info

Search

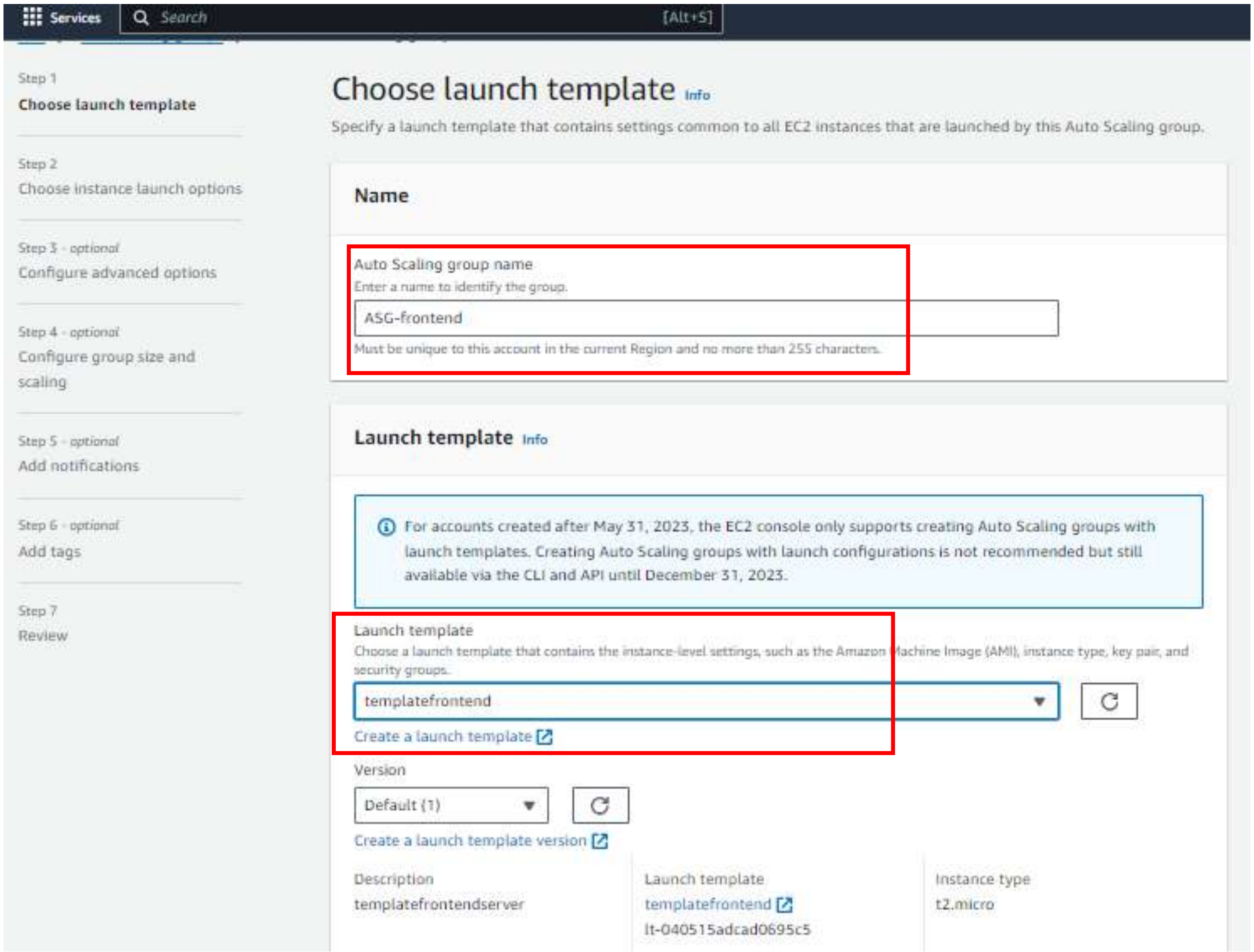
□	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time
□	lt-034e1d21761c53d77	templatebackend	1	1	2024-08-21T14:22:26.000Z
□	lt-040515adcad0695c5	templatefrontend	1	1	2024-08-21T14:20:36.000Z

Auto scaling group (ASG)

The auto-scaling group is the functionality of EC2 service that launches instances depending on your network traffic or CPU utilization or parameter that you set. It launches instances from the launch template.

Click on the Auto scaling group’s button which is located at the bottom of the left panel. And then click on the Create auto scaling group button.

Give a name to your ASG. E.g. ASG-frontend. And select the launch template that we have created for frontend (e.g. templatefrontend) in the launch template field. And click on the next button.



Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

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

Launch template Info

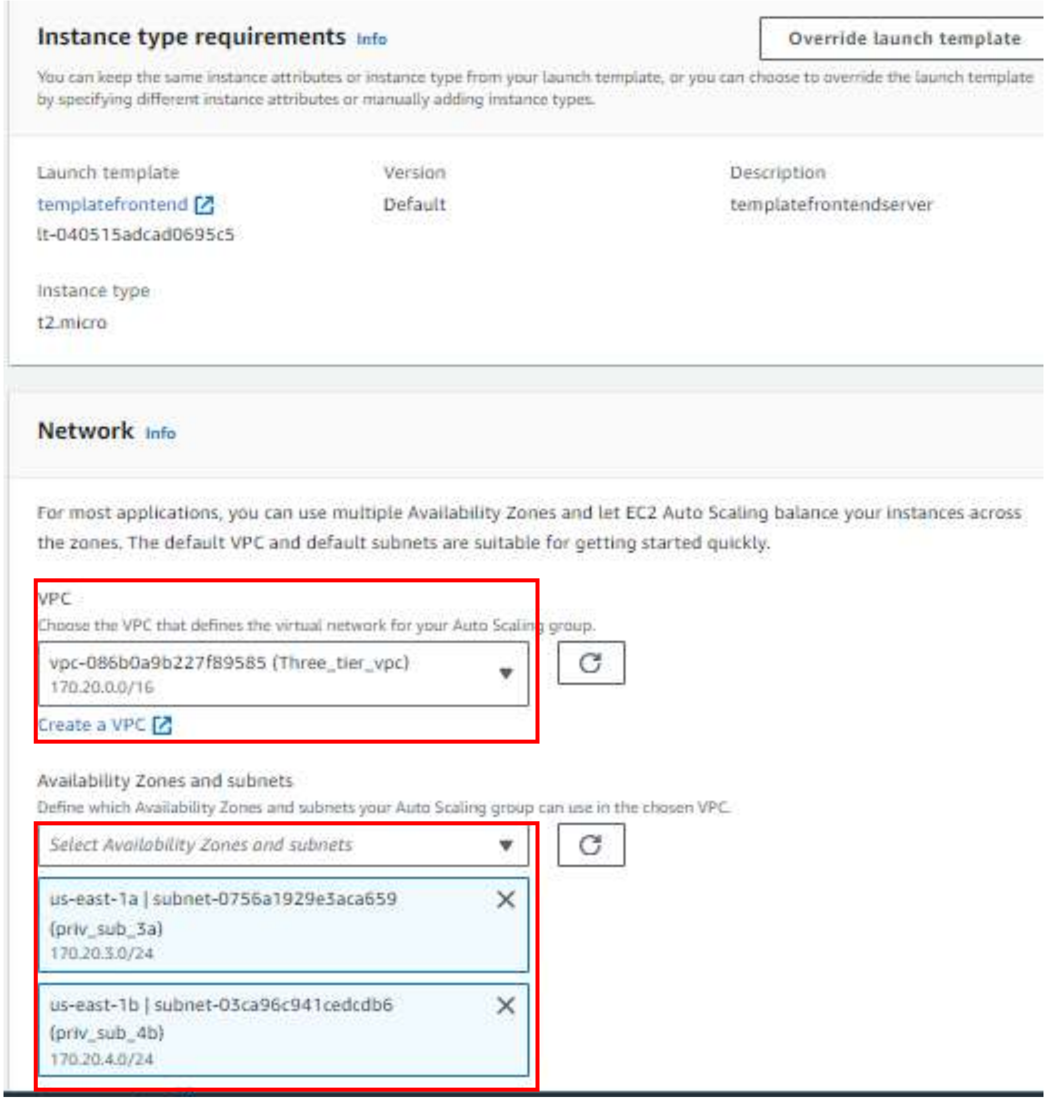
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.
templatefrontend
Create a launch template

Version
Default (1)
Create a launch template version

Description	Launch template	Instance type
templatefrontendserver	templatefrontend lt-040515adcad0695c5	t2.micro

In the network field, you have to choose VPC that we created earlier. And in AZs and subnet filed choose pri-sub-3a and pri-sub-4b. These subnets we have created for frontend servers. And click on the next button.



Instance type requirements Info

Override launch template

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

Launch template: templatefrontend
lt-040515adcad0695c5

Version: Default

Description: templatefrontendserver

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-086b0a9b227f89585 (Three_tier_vpc)
170.20.0.0/16

Create a VPC

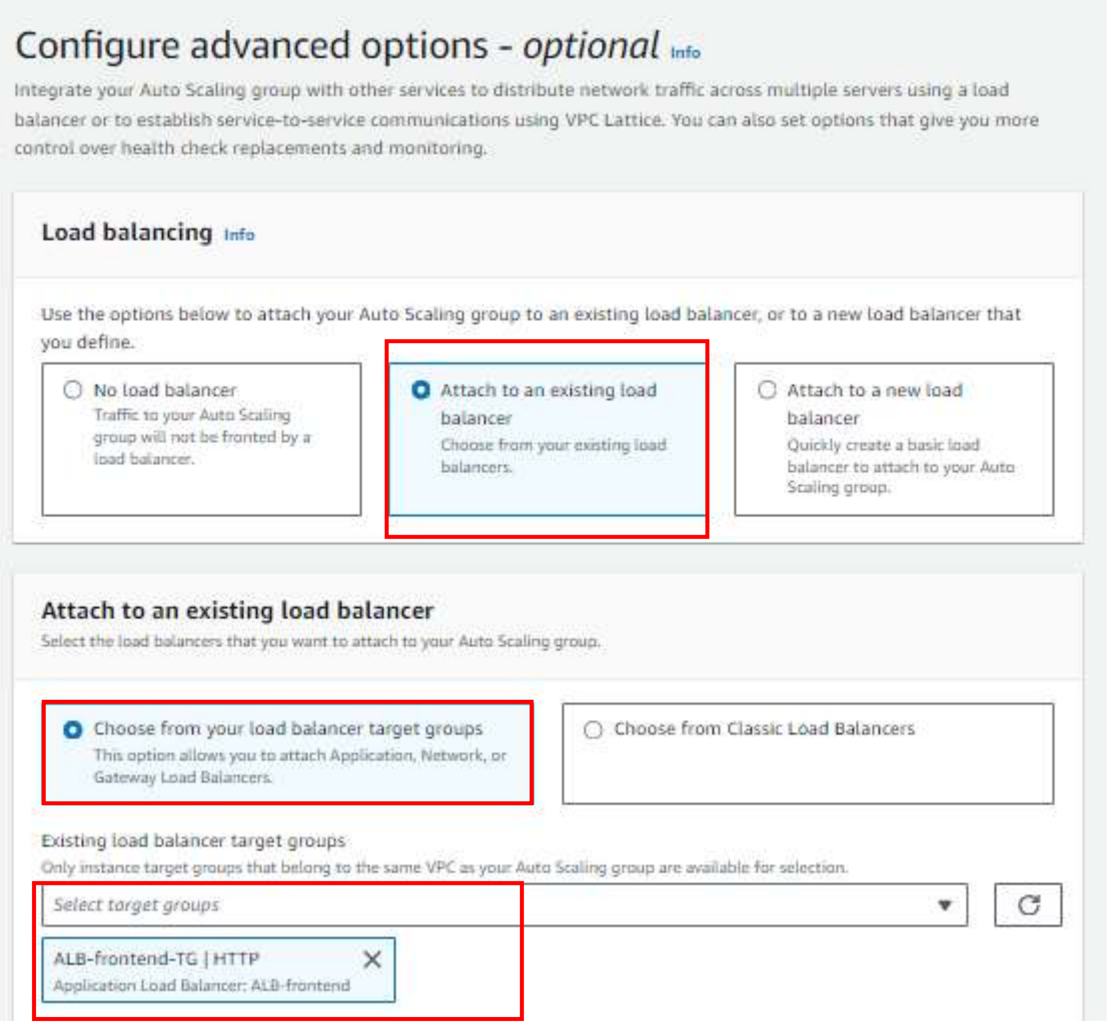
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-0756a1929e3aca659 (priv_sub_3a)
170.20.3.0/24

us-east-1b | subnet-03ca96c941cedcdb6 (priv_sub_4b)
170.20.4.0/24

On this page we need to attach ASG with ALB so select the Attach existing ALB option and select TG that we have created for frontend e.g. ALB-frontend-TG. And then scroll down and click on the NEXT button



Configure advanced options - *optional* Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over 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.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

● Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

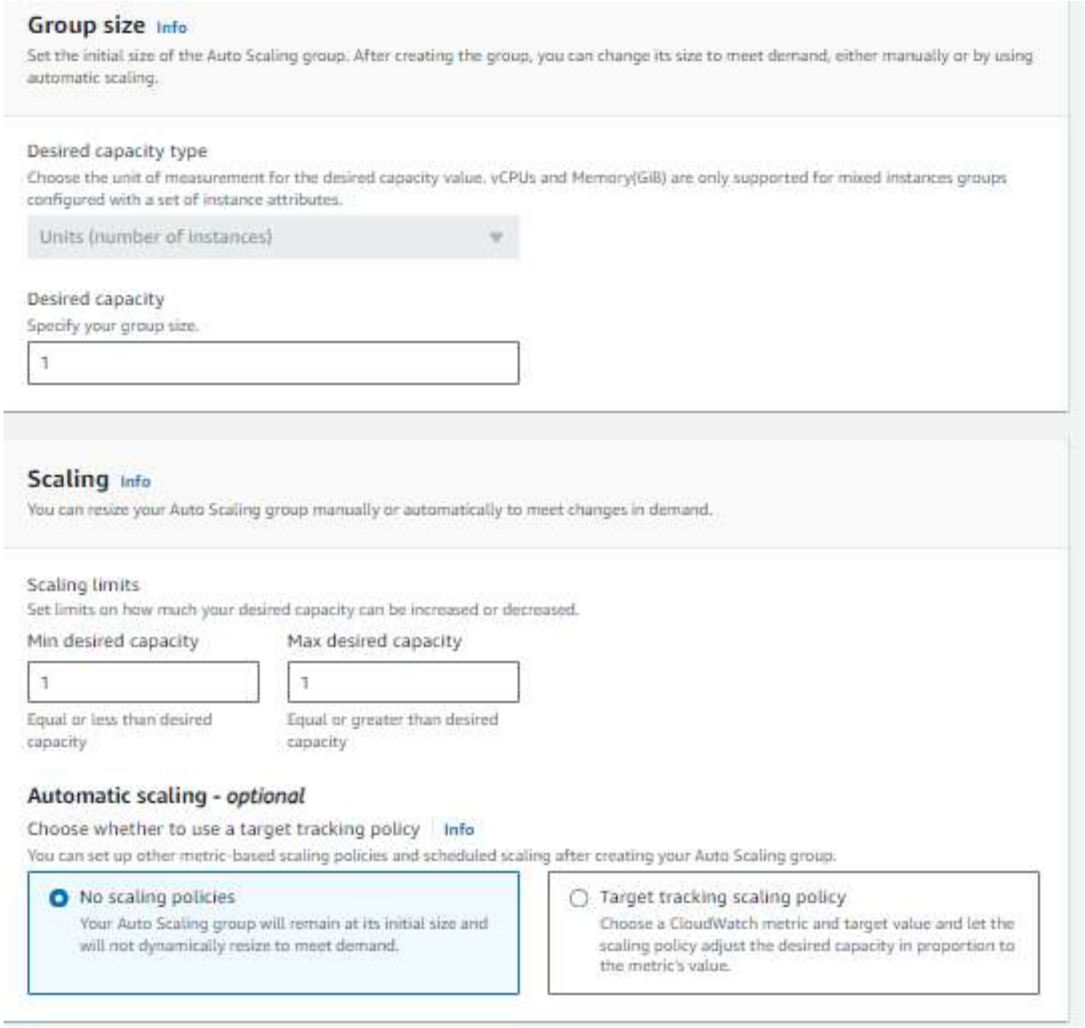
○ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

ALB-frontend-TG | HTTP
Application Load Balancer: ALB-frontend

Here you can set the capacity and scaling policy but now I am keeping 1, 1, and 1 to save cost but in real projects, it depends on the traffic. Click on the NEXT->next->next-> and create ASG button.



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.

1

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: 1 — 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.

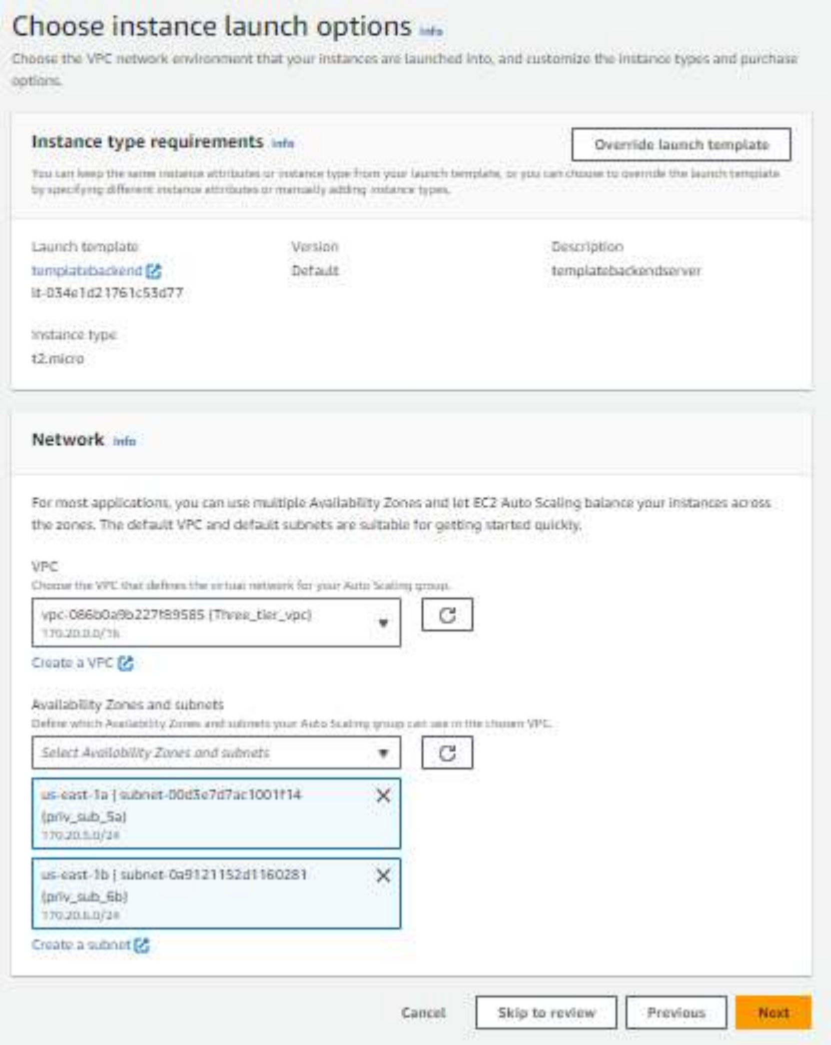
● No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

○ 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.

Let's set up ASG for the backend.

Give a name to your ASG. E.g. backendasg. And select the launch template that we have created for the backend (e.g. templatebackend) in the launch template field. And click on the next button.

In the network field, you have to choose VPC that we created earlier. And in AZ and subnet field choose pri-sub-5a and pri-sub-6b. These subnets we have created for backend servers. And click on the next button.



Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info

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

Launch template	Version	Description
templatebackend lt-034e1d21761c53d77	Default	templatebackendserver

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-086b0a9b227f89585 (Three_tier_vpc)
170.20.0.0/16
Create a VPC

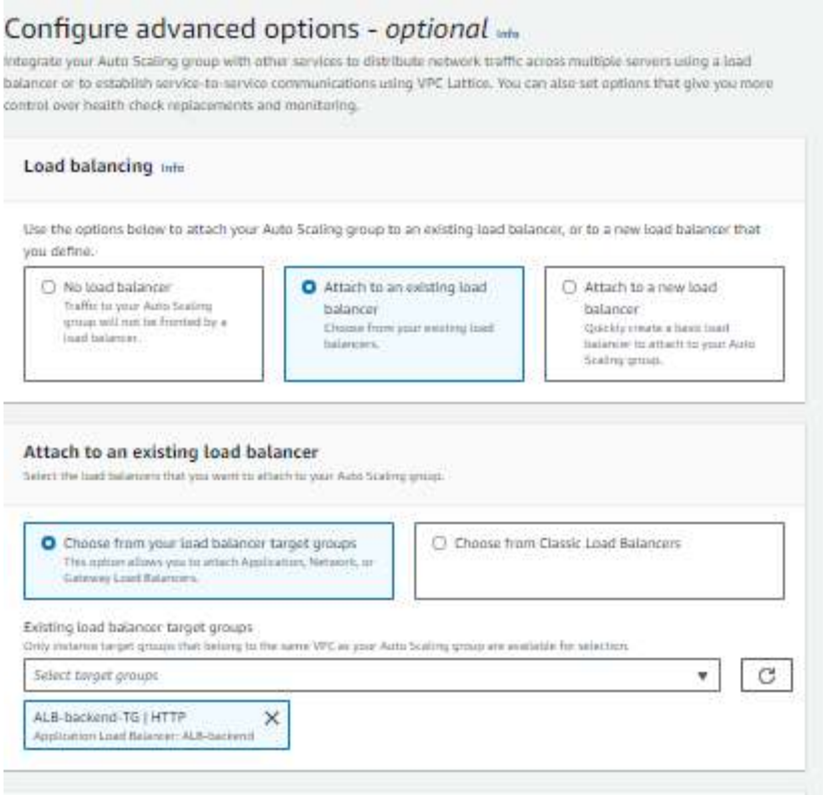
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets

us-east-1a | subnet-00d3e7d7ac1001f14 (priv_sub_5a)
170.20.5.0/24

us-east-1b | subnet-0a9121152d1160281 (priv_sub_6b)
170.20.6.0/24

Create a subnet

On this page we need to attach ASG with ALB so select the Attach existing ALB option and select TG that we have created for the backend e.g. ALB-backend-TG. And then scroll down and click on the NEXT button.



Configure advanced options - *optional* Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over 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.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

● Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

○ Choose from Classic Load Balancers

Existing load balancer target groups

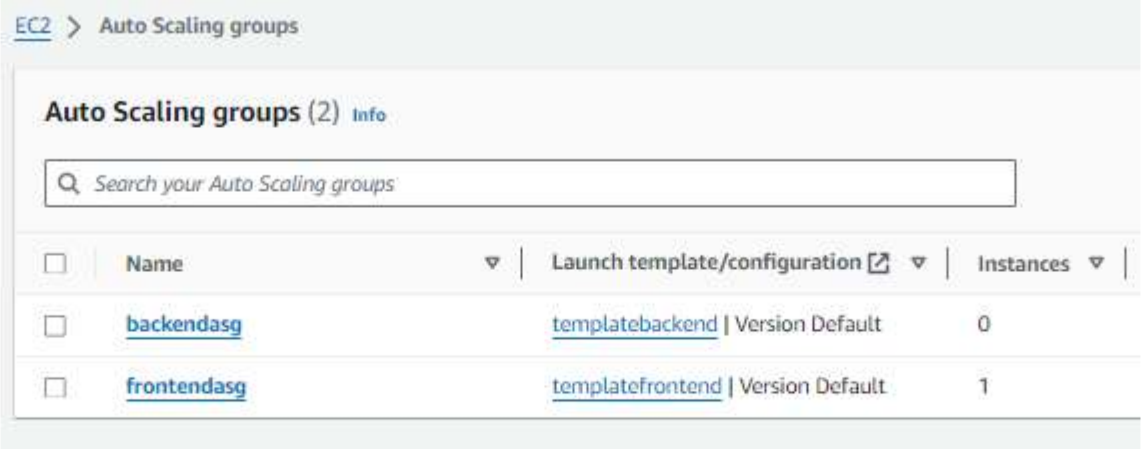
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

ALB-backend-TG | HTTP
Application Load Balancer: ALB-backend

Here you can set the capacity and scaling policy but I’m keeping 1, 1, 1 to save cost but in real projects, it depends on the traffic. Click on the NEXT->next->next-> and create ASG button.

Now, We have two ASGs, ASG-frontend will launch frontend servers and ASG-backend will launch backend servers.



EC2 > Auto Scaling groups

Auto Scaling groups (2) Info

Search your Auto Scaling groups

□	Name	Launch template/configuration	Instances
□	backendasg	templatebackend | Version Default	0
□	frontendasg	templatefrontend | Version Default	1

We need to initialize our database and need to create some tables. But we can’t access the RDS instance or backend server directly coz they are in a private subnet. So we need to launch an instance in the same VPC but in the public subnet that instance is called bastion host or jump-server. And through that instance, we will log in to the backend server, and from the backend server we will initialize our database.

Click on the instance button on the left panel and click on the launch instance button in the top right corner.

Give a name to the instance (bastion-jump-server). Select Ubuntu as OS, instance typet2.micro, and select Key pair. In all the instance and launch template we have used only **one key** so it will be easy to login in any instance. And then click on the Edit button of the Network setting.

In the network setting select VPC that we have created and in the subnet select pub-sub-1a, you can select any public subnet from the VPC. And then select security group. And click on the launch instance.

Once the instance becomes healthy, we can SSH into it. So select the instance and copy its public IP. Open Git bash or terminal in which folder your key.pem file is present and connect

Now copy the pem file to ubuntu server and give permission to connect the backend server which is in private subnet

Now type the below command to login into the Bastion host. And copy the public IP of the Bastion host.

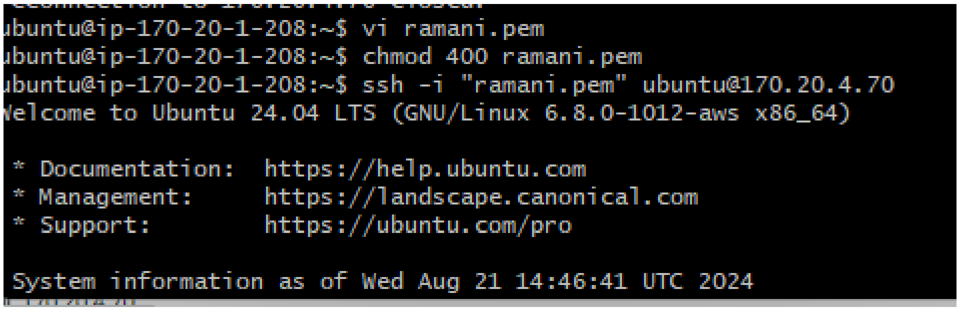
ssh -i <name_of_your_key>.pem ubuntu@<Public_IP_add_of_instance>

We are successfully logged in inside the bastion host.

Create a file of your pem

Give permissions: chmod 400 keypair.pem

Ssh keypair.pem ubuntu@<frontend/backend server private ip>



```
Connection to 170.20.4.70 closed.
ubuntu@ip-170-20-1-208:~$ vi ramani.pem
ubuntu@ip-170-20-1-208:~$ chmod 400 ramani.pem
ubuntu@ip-170-20-1-208:~$ ssh -i "ramani.pem" ubuntu@170.20.4.70
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

 System information as of Wed Aug 21 14:46:41 UTC 2024
```

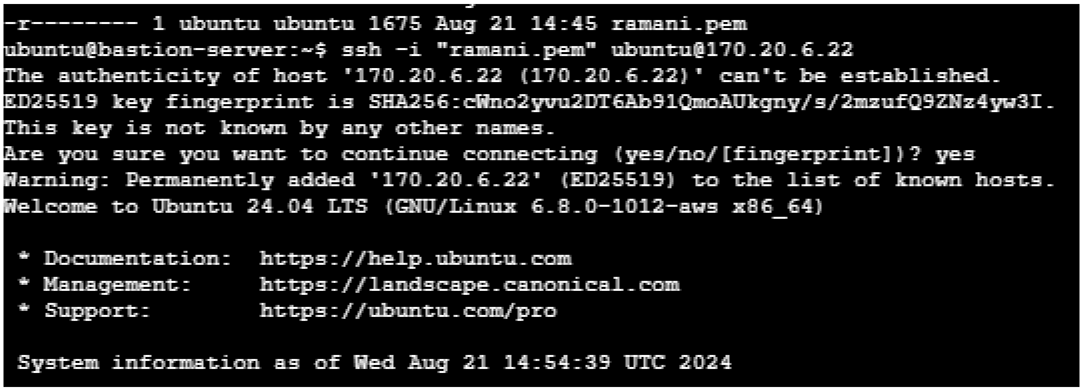
Now we you have login into the frontend server

Run below script:

```
#!/bin/bash

sudo apt update -y

sleep 90

sudo systemctl start apache2.service
```

Type the below command to log in to the backend server.

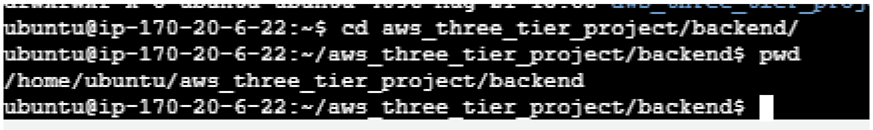
ssh -i key.pem ubuntu@<Private_IP_add_backend_server>



```
-r--------- 1 ubuntu ubuntu 1675 Aug 21 14:45 ramani.pem
ubuntu@bastion-server:~$ ssh -i "ramani.pem" ubuntu@170.20.6.22
The authenticity of host '170.20.6.22 (170.20.6.22)' can't be established.
ED25519 key fingerprint is SHA256:cWno2yvu2DT6Ab91QmoAUkgny/s/2mzufQ9ZNz4yw3I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '170.20.6.22' (ED25519) to the list of known hosts.
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

 System information as of Wed Aug 21 14:54:39 UTC 2024
```

Now we are logged in inside the backend server. Just go into aws_three_tier_project/backend directory

```
cd  aws_three_tier_project/backend
```

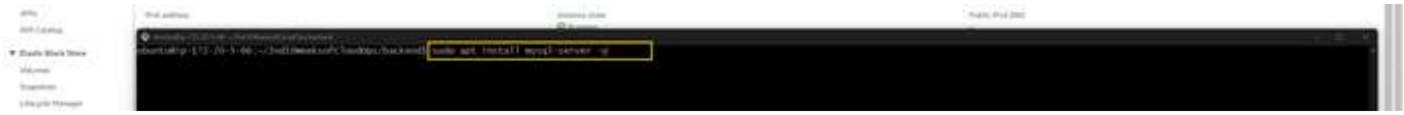


```
ubuntu@ip-170-20-6-22:~$ cd aws_three_tier_project/backend/
ubuntu@ip-170-20-6-22:~/aws_three_tier_project/backend$ pwd
/home/ubuntu/aws_three_tier_project/backend
ubuntu@ip-170-20-6-22:~/aws_three_tier_project/backend$ 
```

We need to install one package type below the command

```
#!/bin/bash

sudo apt update -y
```

```
sleep 150

sudo pm2 startup

sudo env PATH=$PATH:/usr/bin
/usr/local/share/.config/yarn/global/node_modules/pm2/bin/pm2 startup systemd -u ubuntu
--hp /home/ubuntu

sudo systemctl start pm2-root

sudo systemctl enable pm2-root

sudo apt install mysql-server -y
```



```
ubuntu@ip-172-20-5-66:~/2nd10WeeksofCloudOps/backend$ sudo apt install mysql-server -y
```

And type the below command to initialize the database.

mysql -h book.rds.com -u <user_name_of_rds> -p<password_of_rds> test < test.sql

```
ubuntu@ip-172-20-3-66:~/2nd10WeeksofCloudOps/backend$ ls
index.js  node_modules  package-lock.json  package.json  test.sql
ubuntu@ip-172-20-3-66:~/2nd10WeeksofCloudOps/backend$ mysql -h book.rds.com -u admin -p[illegible] test < test.sql
```

Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service.

If you try to access the web app using ALB-frontend DNS then you won’t see the website in functional mode because our frontend or loaded static pages try to call the API from your browser on the domain namehttps://api.<Your_Domain_name>.xyz in my case, https://api.b13facebook.xyz

And that record we didn’t add yet in our domain name.

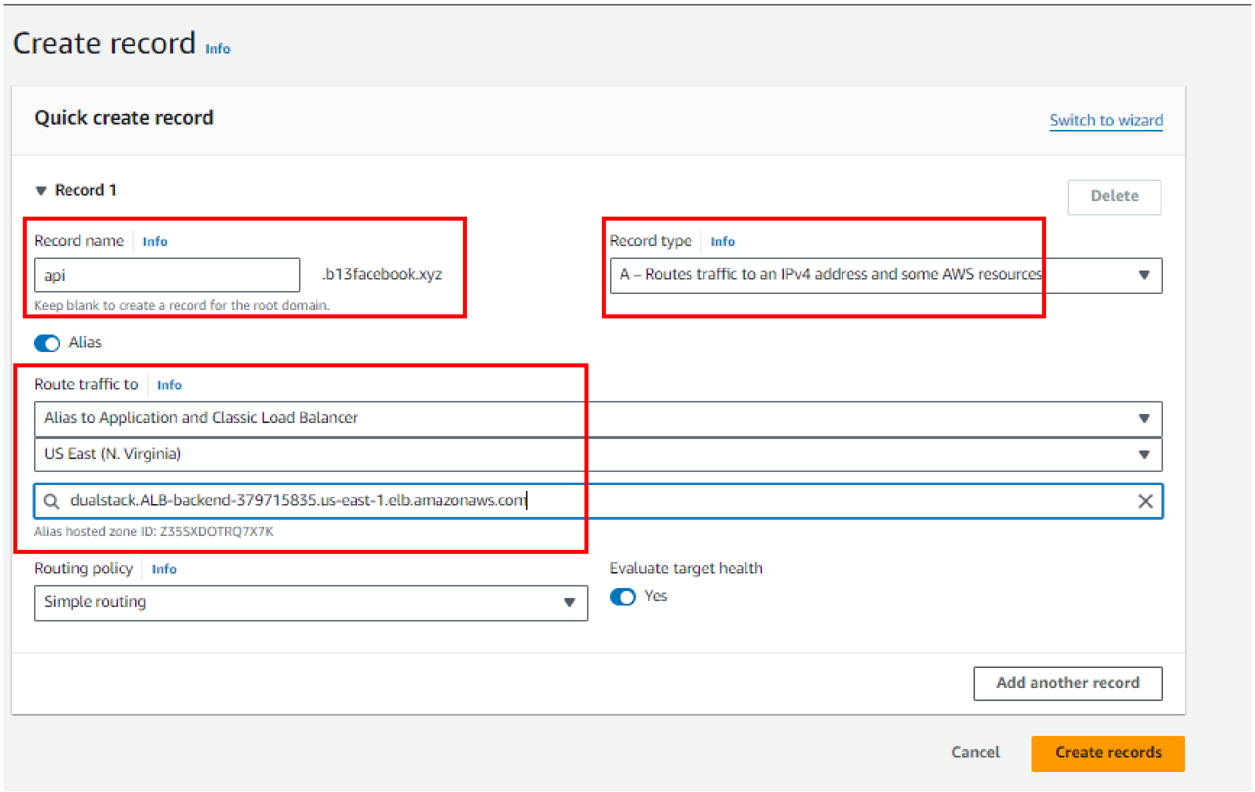
Head over to route 53 service.

Here in the record name field write api so that our record name becomes api.<Your_Domain_name>.xyz in my case, it is api. b13facebook.xyz

In the record type field select “A”

Firstly Select Alias to application and classic Load balancer from the drop-down list, secondly, select us-east-1 as a region. And in the below drop-down list select DNS of the ALB-backend.

Click on create record button.



Create record Info

Quick create record

Record 1

Record name: api .b13facebook.xyz
Keep blank to create a record for the root domain.

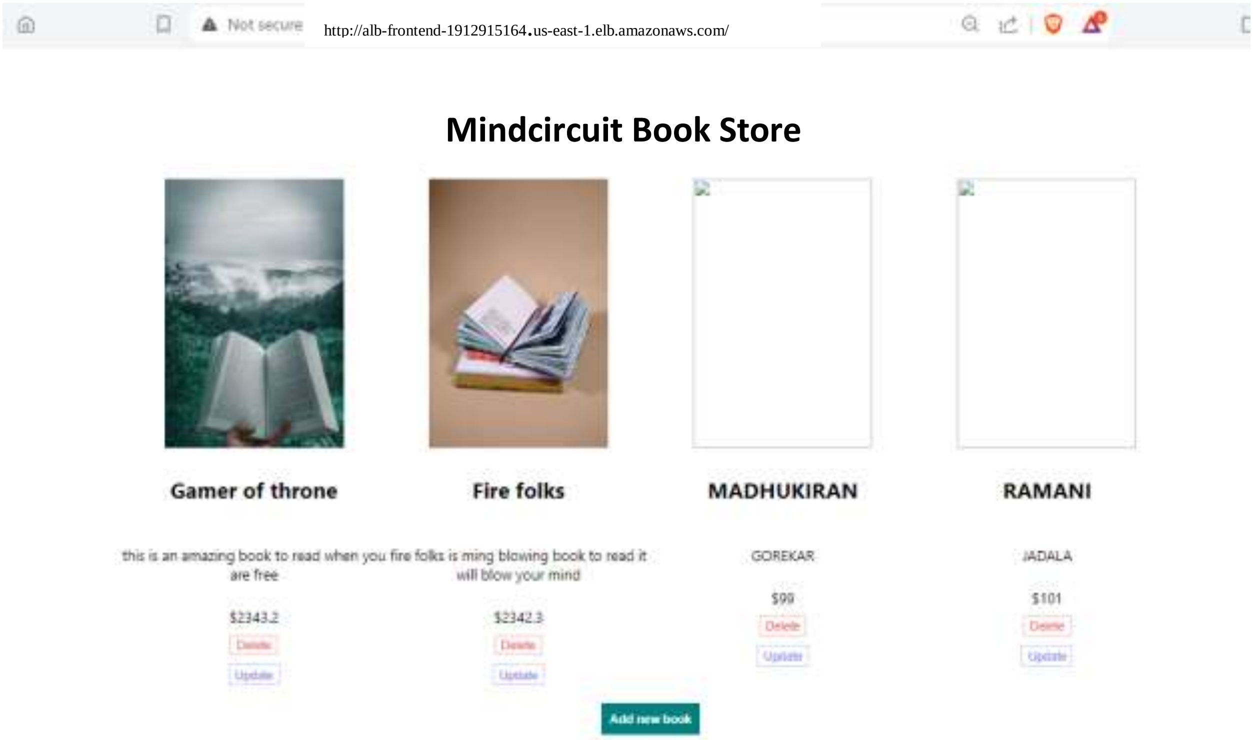
Record type: A – Routes traffic to an IPv4 address and some AWS resources

Alias: on

Route traffic to: Alias to Application and Classic Load Balancer
US East (N. Virginia)
dualstack.ALB-backend-379715835.us-east-1.elb.amazonaws.com
Alias hosted zone ID: Z35SXDOTRQ7X7K

Routing policy: Simple routing

Evaluate target health: Yes



Not secure http://alb-frontend-1912915164.us-east-1.elb.amazonaws.com/

Mindcircuit Book Store

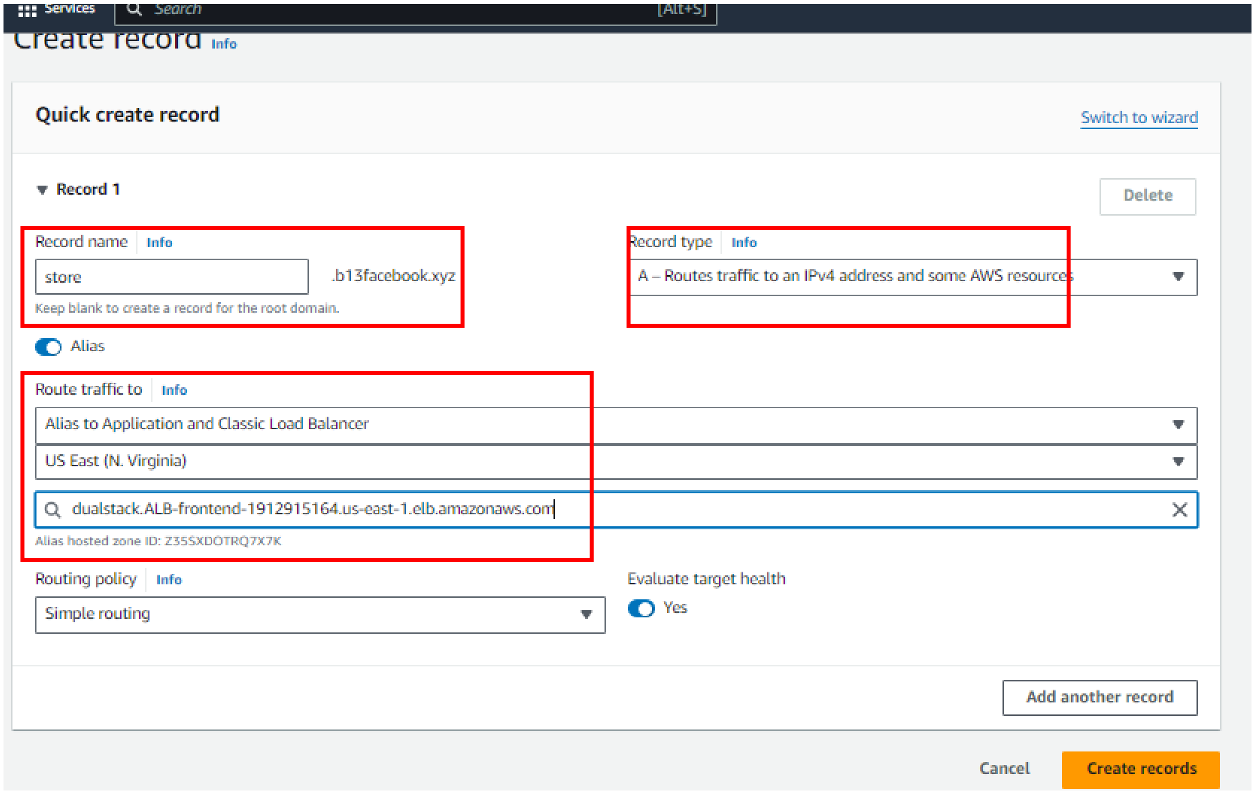
Gamer of throne
this is an amazing book to read when you are free
$2343.2

Fire folks
fire folks is ming blowing book to read it will blow your mind
$2342.3

MADHUKIRAN
GOREKAR
$99

RAMANI
JADALA
$101

Create the records for alb frontend dns to secure the server



Create record Info

Quick create record

Record 1

Record name: store .b13facebook.xyz
Keep blank to create a record for the root domain.

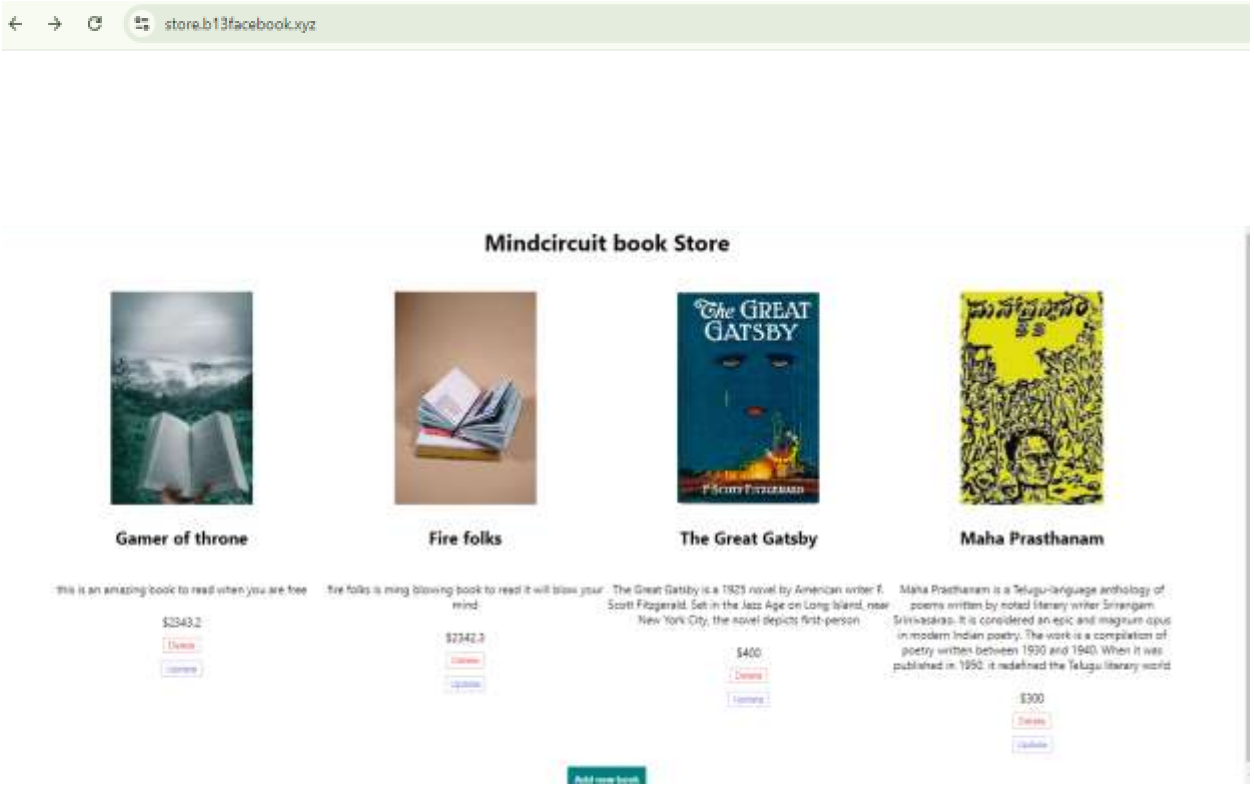
Record type: A – Routes traffic to an IPv4 address and some AWS resources

Alias: on

Route traffic to: Alias to Application and Classic Load Balancer
US East (N. Virginia)
dualstack.ALB-frontend-1912915164.us-east-1.elb.amazonaws.com
Alias hosted zone ID: Z35SXDOTRQ7X7K

Routing policy: Simple routing

Evaluate target health: Yes



Mindcircuit book Store

Gamer of throne

this is an amazing book to read when you are free

$2343.2

Fire folks

fire folks is ming blowing book to read it will blow your mind

$2342.3

The Great Gatsby

The Great Gatsby is a 1925 novel by American writer F. Scott Fitzgerald. Set in the Jazz Age on Long Island, near New York City, the novel depicts first-person

$400

Maha Prasthanam

Maha Prasthanam is a Telugu-language anthology of poems written by noted literary writer Srirangam Srinivasarao. It is considered an epic and magnum opus in modern Indian poetry. The work is a compilation of poetry written between 1930 and 1940. When it was published in 1950, it redefined the Telugu literary world

$300

Resource cleanup

✓ **RDS**

- RDS instance (takes a lot of time)

✓ **Route 53**

- Delete private hosted zone (rds.com)
- Delete all records in the public hosted zone

✓ **EC2**

- Delete ASG
- Terminate Bastion host
- Delete ALB
- Delete TG
- Delete the Launch template
- Deregister AMIs which are created manually

✓ **ACM**

- Delete the certificate

✓ **VPC**

- Delete NAT gateways (takes around 5 minutes)
- Release the Elastic IP
- Delete VPC in both regions (17 resources will be deleted on one click)