Deploy 3 Tier Architecture, Configure Security Group and Autoscaling – Assignment

1. Introduction

1.1. Problem Description:

One specific problem within the cloud environment is the lack of a structured architecture to efficiently handle the deployment and management of applications. Without a well-defined architecture, organizations face challenges related to scalability, security, and maintainability. Specifically, the absence of a structured approach can lead to monolithic applications, making it difficult to scale components independently, secure sensitive data, and maintain agility in development and deployment processes. Additionally, without a clear separation of concerns, troubleshooting and debugging become more complex, impacting operational efficiency and hindering innovation.

1.2. Consequences:

Scalability becomes a major issue as applications grow and user demands increase.

Security vulnerabilities arise due to the lack of isolation between different components of the application. Without clear boundaries between layers, sensitive data stored within the application may be more susceptible to unauthorized access or data breaches.

The absence of a structured architecture inhibits agility and innovation. Monolithic applications are harder to update and maintain, slowing down the development and deployment cycles.

1.3. Solution using 3-tier structure with AWS:

By adopting a three-tier architecture with AWS services, organizations can address scalability, security, and maintainability challenges effectively. The structured separation of concerns enables independent scaling of components, improves security by isolating sensitive data, and enhances agility through automation and deployment flexibility. This approach empowers organizations to leverage the full potential of cloud technology while ensuring optimal performance, security, and cost efficiency.

Here are some specific aspects of this challenge:

Aspects	Challenges	Solution
Cost Visibility	Many organizations struggle to gain visibility into their cloud spending across multiple services, regions, and accounts. This lack of visibility makes it difficult to understand where costs are coming from and how they can be optimized.	Use AWS Cost Explorer to gain visibility into your AWS spending by service, region, instance type, and more.
		Enable detailed billing to get granular insights into resource usage and costs.
		Set up AWS Budgets to create custom cost and usage budgets with alerts when thresholds are exceeded.
Resource Utilization	Cloud resources, such as virtual machines and storage, are often provisioned with excess capacity to accommodate potential spikes	Utilize AWS Trusted Advisor to identify underutilized resources and recommend rightsizing opportunities.
		Implement AWS Auto Scaling to dynamically scale resources based on

	in demand. However, this can lead to underutilization and unnecessary costs if resources are not rightsized or scaled dynamically based on actual usage.	demand, optimizing resource utilization and cost.
		Consider using AWS Savings Plans or Reserved Instances to commit to usage in exchange for significant discounts
Complex Pricing Models	Cloud resources, such as virtual machines and storage, are often provisioned with excess capacity to accommodate potential spikes in demand. However, this can lead to underutilization and unnecessary costs if resources are not rightsized or scaled dynamically based on actual usage.	Leverage AWS Pricing Calculator to estimate costs for various AWS services and configurations before deployment.
		Explore AWS Cost Anomaly Detection to identify unexpected cost changes and investigate root causes.
		Utilize Cost Allocation Tags to categorize resources for more granular cost analysis and tracking.
Lack of Governance	In multi-cloud or hybrid cloud environments, organizations may lack centralized governance and controls over resource provisioning and spending. This can result in instances of shadow IT, where departments or teams independently procure cloud resources without considering cost implications.	Implement AWS Organizations to centrally manage and govern multiple AWS accounts within your organization.
		Use AWS Control Tower to set up and govern a secure, multi-account AWS environment based on AWS best practices.
		Utilize AWS Service Catalog to create and manage approved catalogs of resources for standardized deployment.
Optimization Strategies	Implementing effective cost optimization strategies requires expertise in areas such as rightsizing, instance reservations, spot instances, storage lifecycle management, and container orchestration. Organizations may struggle to develop and implement these strategies without dedicated resources or tools.	Leverage AWS Compute Optimizer to analyze your EC2 instances and provide recommendations for right-sizing and instance type optimization.
		Use AWS Cost Explorer's Cost Anomaly Detection and recommendations to identify areas for optimization.
		Implement AWS Spot Instances or Spot Fleets for cost-effective, short-lived workloads with flexible pricing options.
Security and Compliance	Cost optimization efforts must also consider security and compliance requirements to ensure that cost-saving measures do not compromise data integrity, confidentiality, or regulatory compliance.	Implement AWS Identity and Access Management (IAM) to control access to AWS services and resources based on security policies.
		Utilize AWS Config to assess, audit, and evaluate the configurations of your AWS resources for compliance

		with industry standards and best practices.
		Leverage AWS CloudTrail for visibility into API activity and changes made to your AWS resources, aiding in security analysis, resource change tracking, and compliance auditing.

2. Architecture Overview

2.1. The 3-tier architecture

To address the above said challenges, organizations can implement a three-tier architecture using AWS services:

a) **Presentation Tier (Frontend)**:Deploy frontend components
b) **Application Tier (Backend)**:Deploy application logic and business processes
c) **Data Tier (Database)**:Deploy data storage and management services

2.2. Components involved (frontend, application tier, data tier)

A three-tier architecture utilises the following components. For ease of use and isolation we will launch the entire 3 tier architecture within a separate Virtual Private Cloud.

a) Presentation Tier (Frontend):

i. Web servers using Ec2 and S3
ii. Utilize AWS Identity and Access Management (IAM) to manage user access and permissions, ensuring secure access to frontend resources.

b) Application Tier (Backend):

i. Deploy application logic and business processes using EC2 for virtual server hosting.
ii. Utilize AWS Application Load Balancer (ALB) to distribute traffic across multiple backend instances, improving availability and scalability.

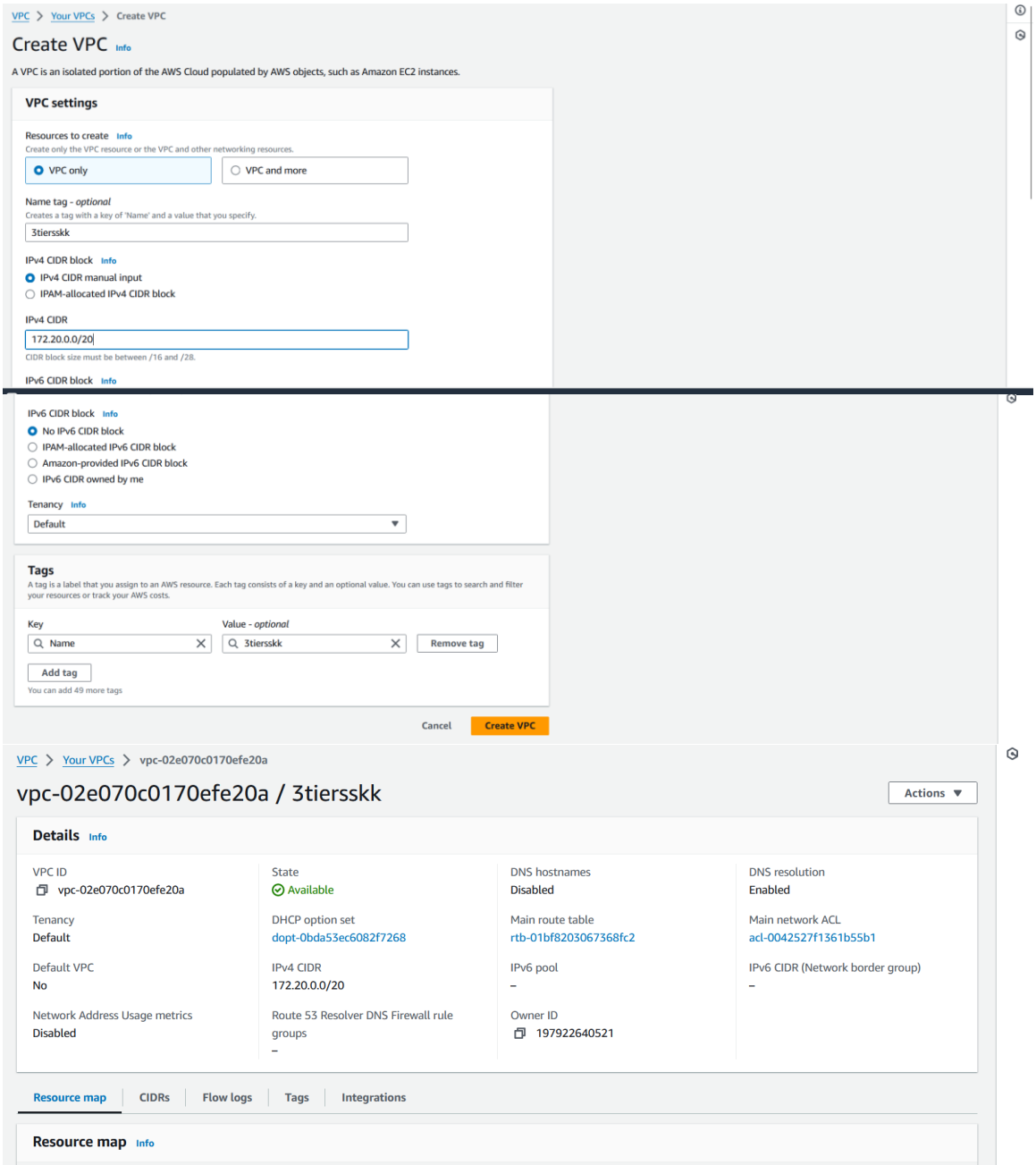
c) Data Tier (Database):

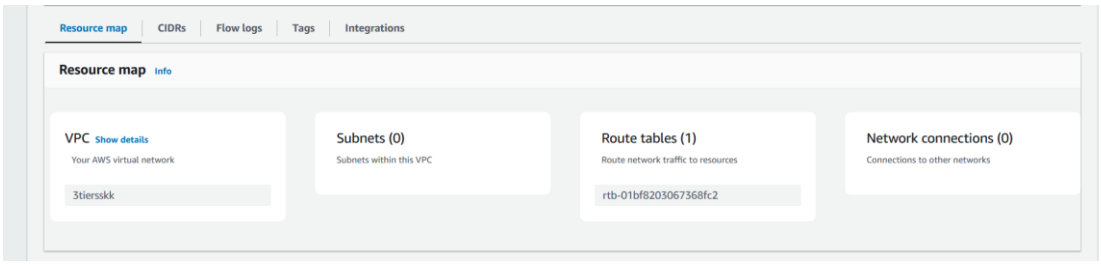
i. Deploy data storage and management services using Amazon RDS for relational databases-MySQL

3. Networking

3.1. Overview of the network setup using VPC (Virtual Private Cloud)

When setting up a network in AWS, the first step is to create a VPC. We define the IP address range (CIDR block) for the VPC, which determines the range of IP addresses available for instances launched within the VPC. Here we have used the CIDR block as 172.20.0.0/20





3.2. Configuration of public and private subnets

3.2.1. Create Public Subnet

Navigate to the VPC dashboard in your cloud provider's console.

- Create a subnet with a CIDR block that falls within your VPC's CIDR range.
- Tag the subnet appropriately to identify it as a public subnet

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-02e070c0170efe20a (3tiersskk)

Associated VPC CIDRs

IPv4 CIDRs
172.20.0.0/20

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_public1
The name can be up to 256 characters long.

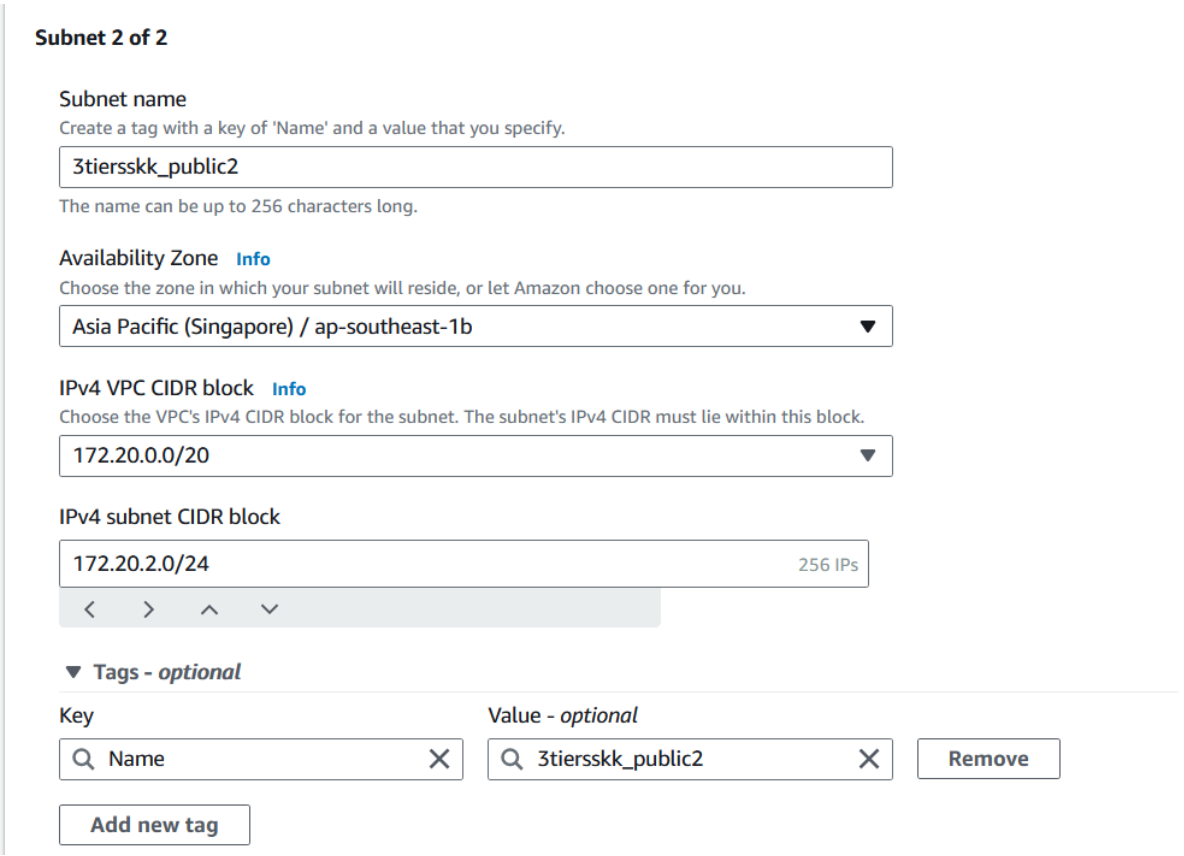
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Singapore) / ap-southeast-1a

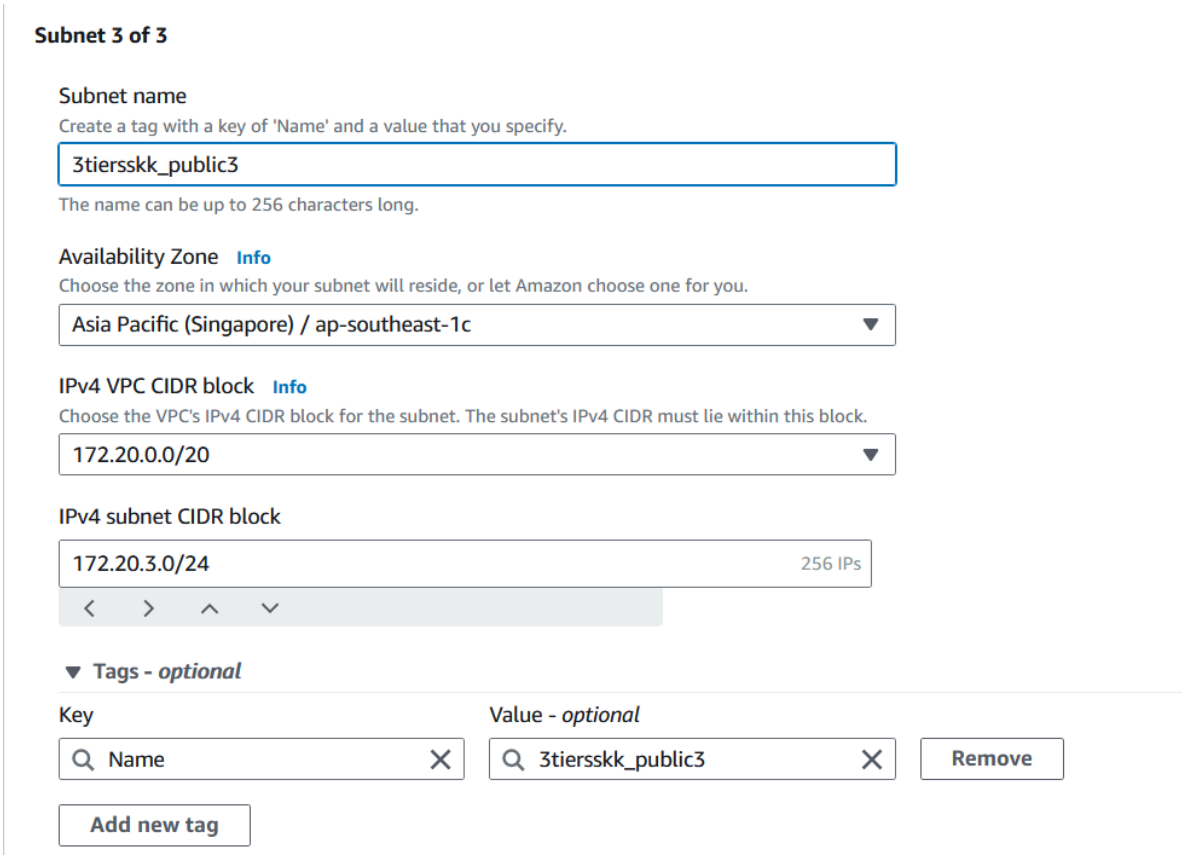
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.1.0/24 256 IPs

▼ Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_public1	Remove





3.2.2. Create Private Subnet:

- Similarly, create another subnet for private resources with a different CIDR block.
- Tag this subnet to identify it as a private subnet

Subnet 4 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_private1
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.
Asia Pacific (Singapore) / ap-southeast-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.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.4.0/24 256 IPs

▼ Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_private1	Remove

Subnet 5 of 5

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_private2
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.
Asia Pacific (Singapore) / ap-southeast-1b

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.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.5.0/24 256 IPs

▼ Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_private2	Remove

Subnet 6 of 6

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_private3
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.
No preference

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.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.6.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_private3	Remove

Add new tag

Subnet 7 of 7

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_db1
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.
Asia Pacific (Singapore) / ap-southeast-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.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.7.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_db1	Remove

Add new tag

Subnet 8 of 9

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_db2
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.
Asia Pacific (Singapore) / ap-southeast-1b

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.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.8.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_db2	Remove

Add new tag
You can add 49 more tags.

Subnet 9 of 9

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
3tiersskk_db3
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.
Asia Pacific (Singapore) / ap-southeast-1c

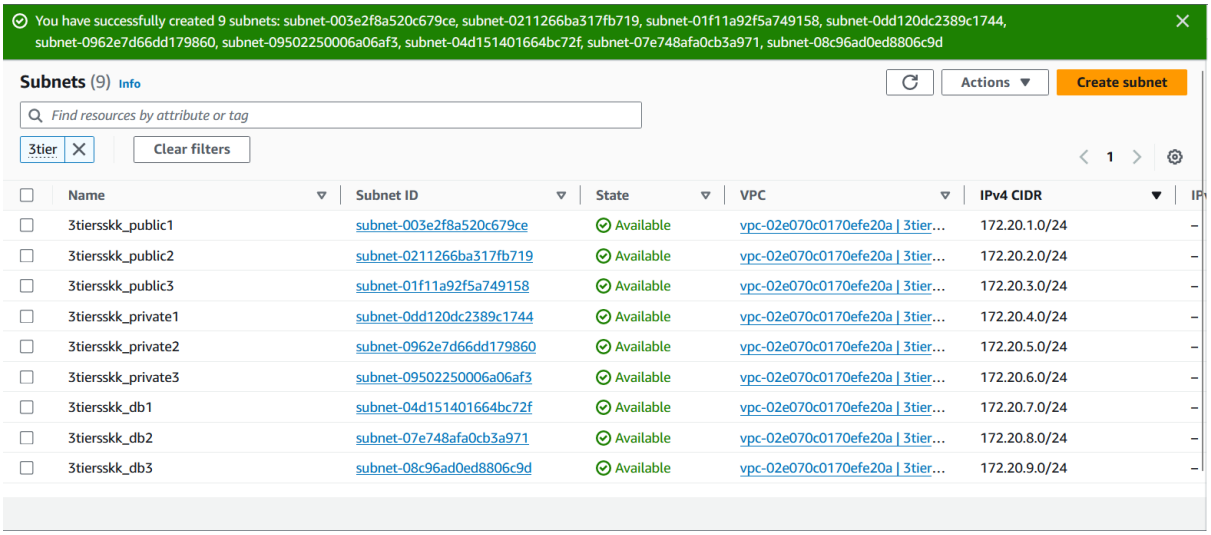
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
172.20.0.0/20

IPv4 subnet CIDR block
172.20.9.0/24 256 IPs

Tags - *optional*

Key	Value - *optional*	
Name	3tiersskk_db3	Remove

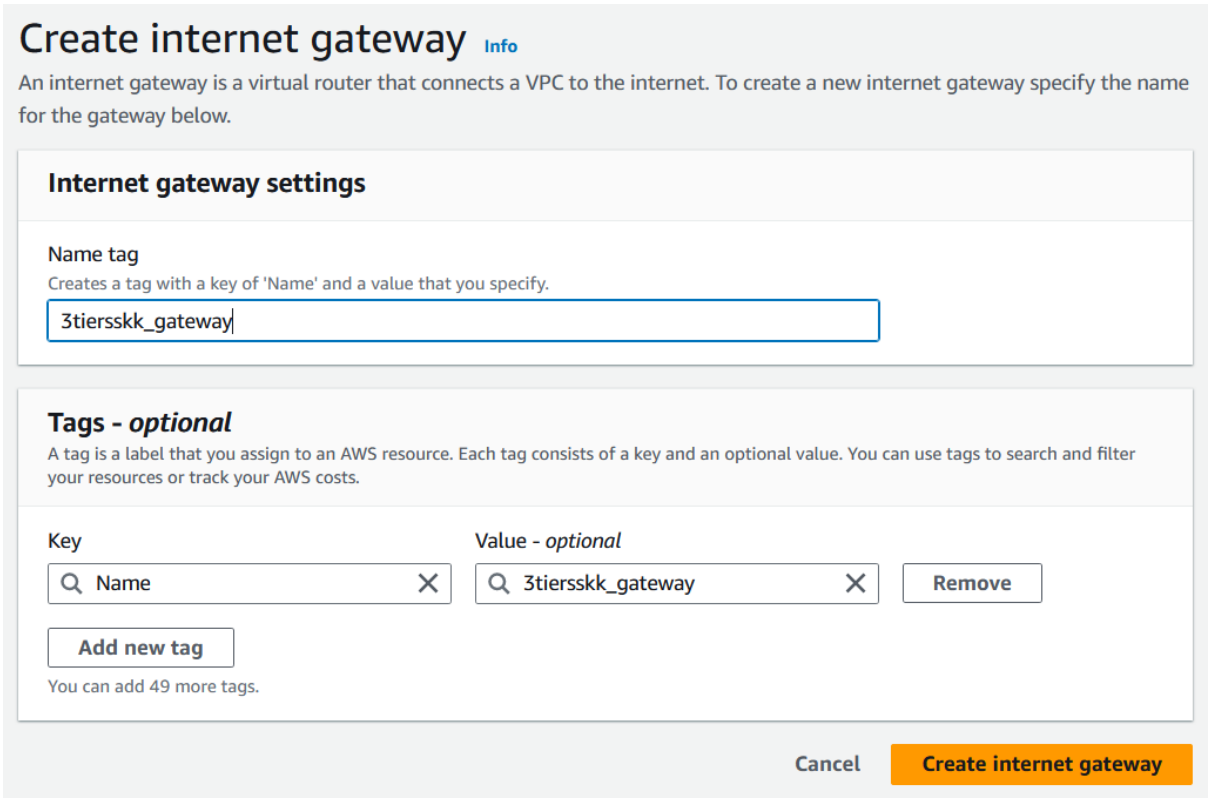
Add new tag

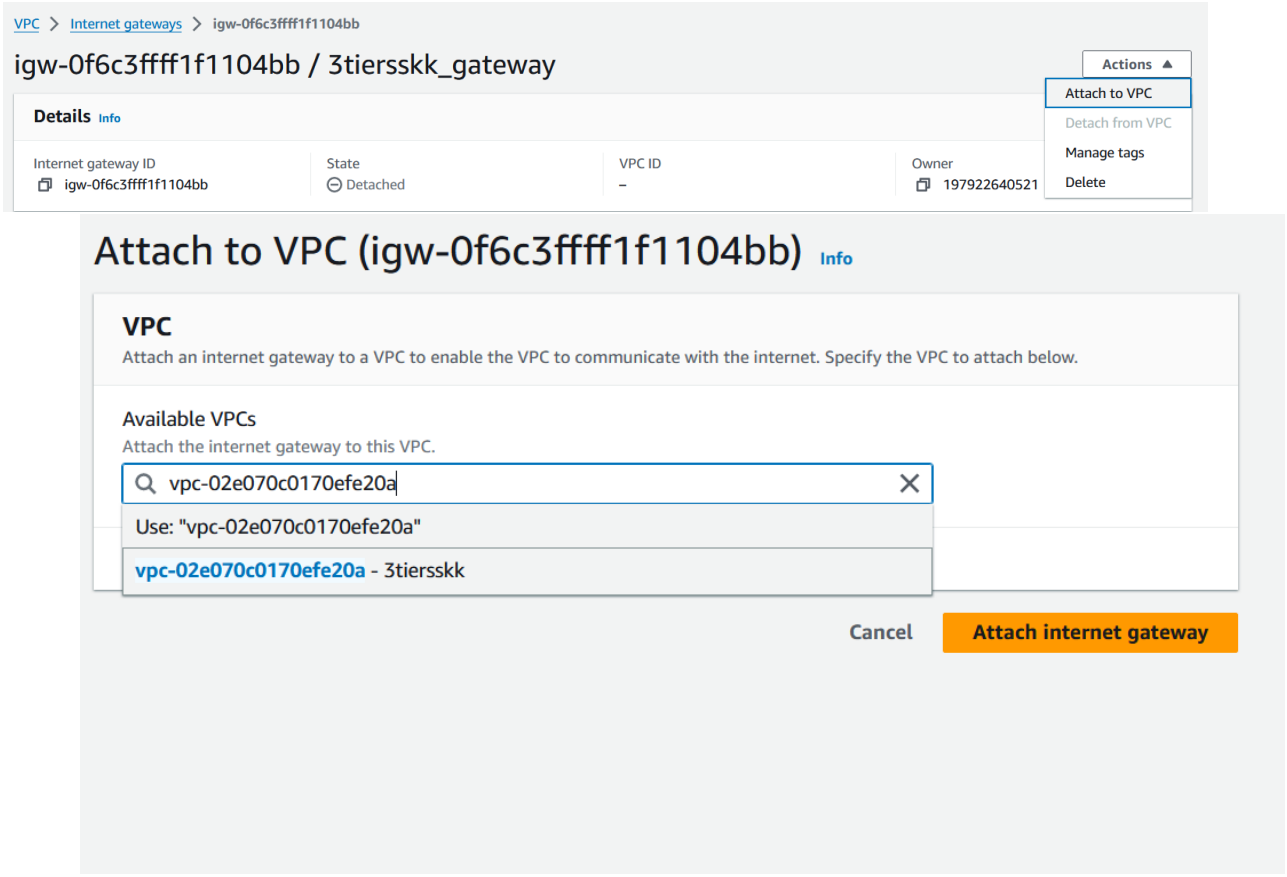


Configuration of Subnets	
Number of Public subnets	03
Number of Private subnets (For private instances)	03
Number of Private subnets (For database)	03

3.3. Internet Gateway setup for public subnet

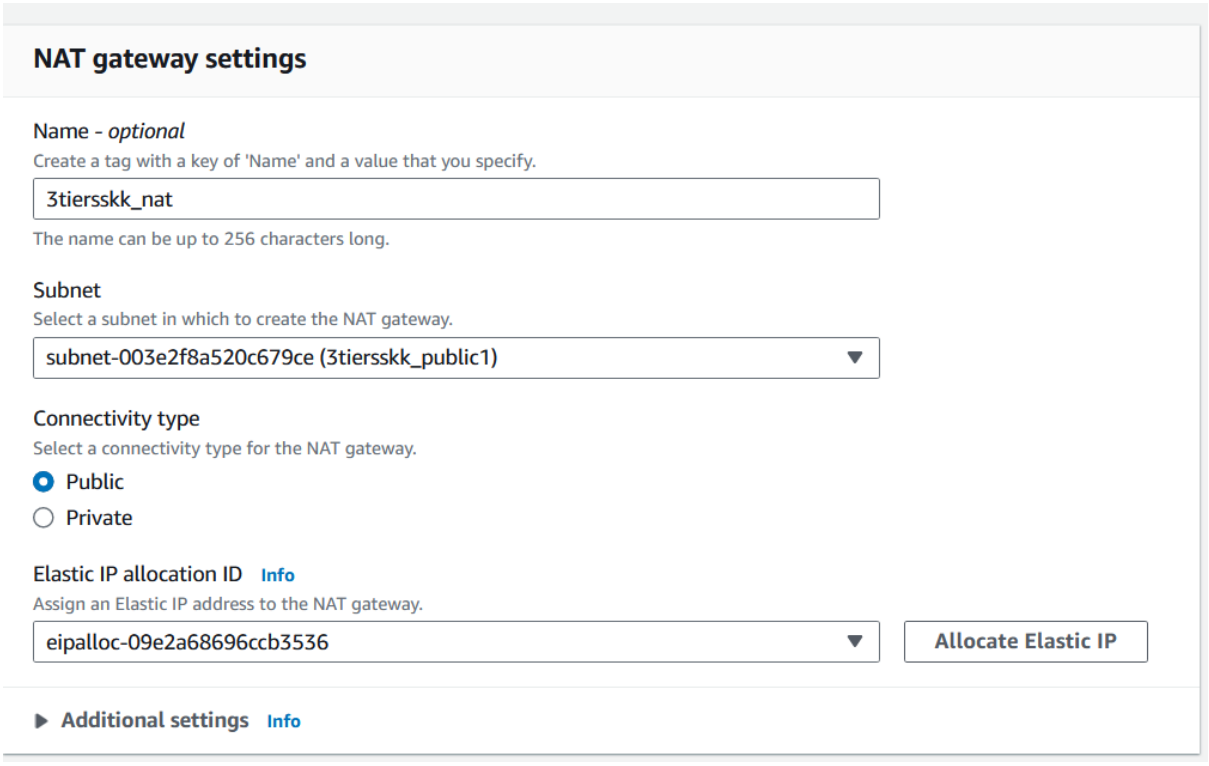
- Go to the internet gateway section in the VPC dashboard.
- Create a new internet gateway and attach it to your VPC.

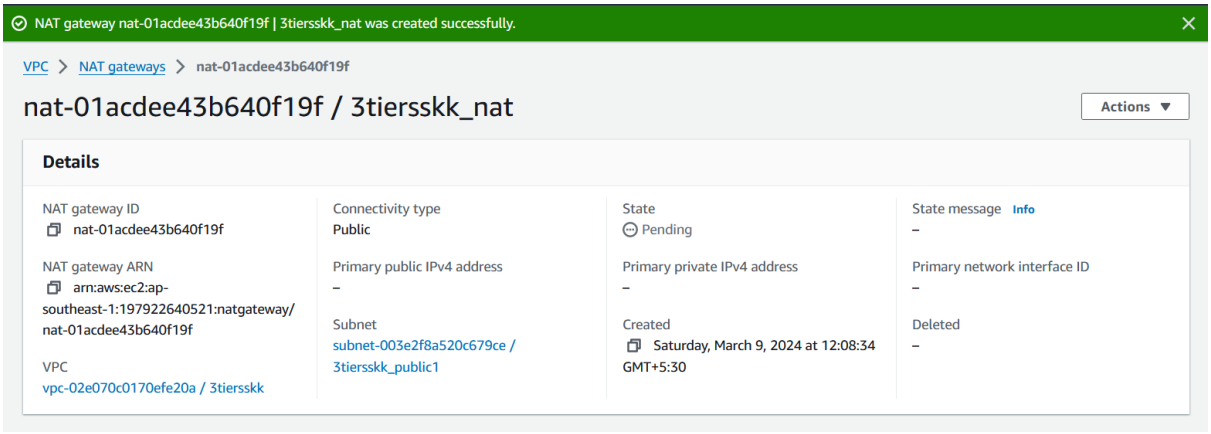


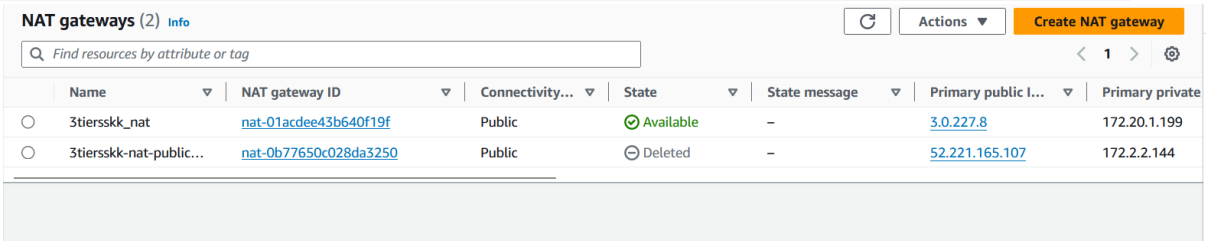


3.4. NAT Gateway setup for private subnet

- In the NAT gateway section of the VPC dashboard, create a new NAT gateway.
- Choose a subnet (preferably public) and allocate an Elastic IP address for the NAT gateway.



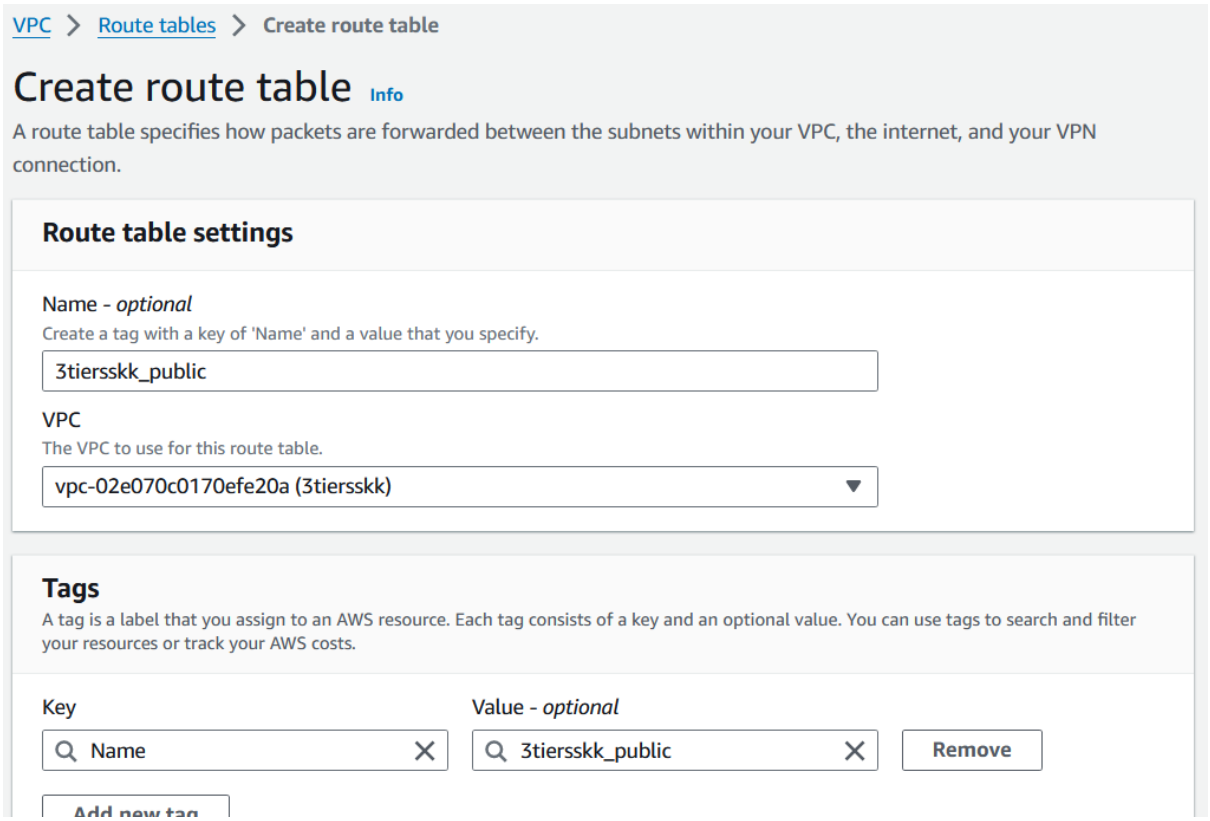




3.5. Routing tables configuration

3.5.1. Create route tables (for public and private subnets)

- Select the VPC that you created or want to associate this route table with.
- Click on "Create" to create the route table



Route table settings

Name - *optional*

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

3tiersskk_private

VPC

The VPC to use for this route table.

vpc-02e070c0170efe20a (3tiersskk)

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	3tiersskk_private	Remove

Add new tag

You can add 49 more tags.

Cancel | Create route table

Route table settings

Name - *optional*

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

3tiersskk_db

VPC

The VPC to use for this route table.

vpc-02e070c0170efe20a (3tiersskk)

Tags

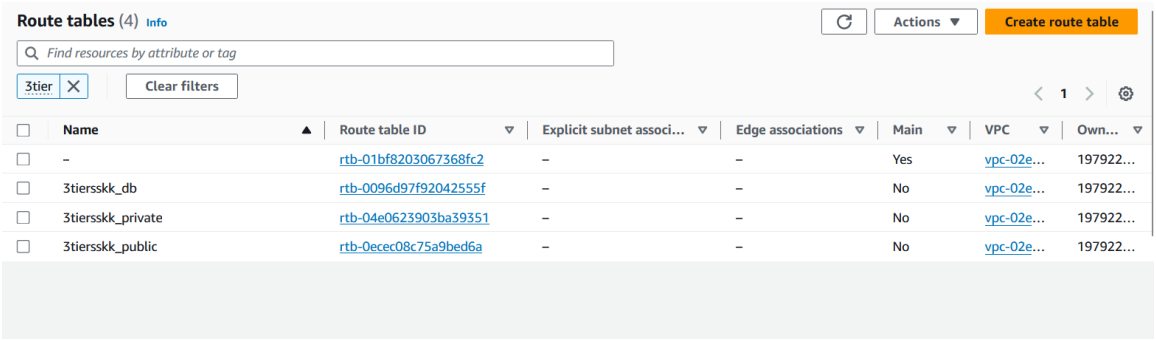
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	3tiersskk_db	Remove

Add new tag

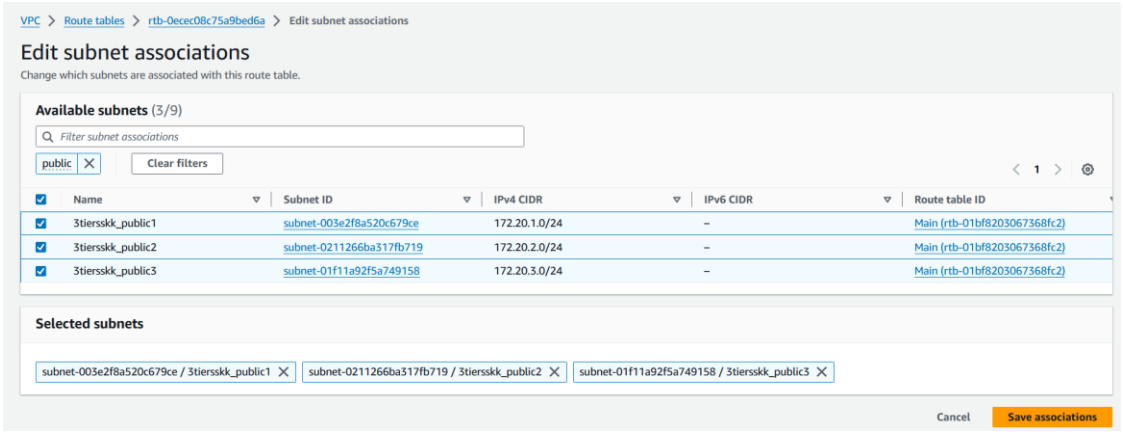
You can add 49 more tags.

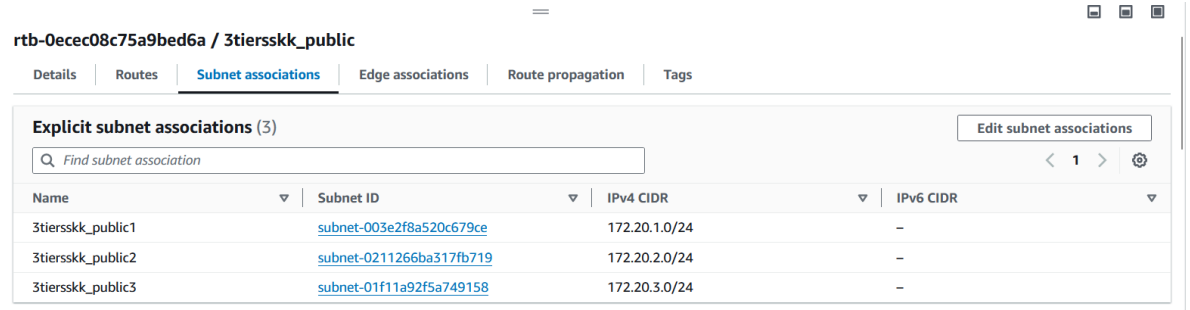
Cancel | Create route table

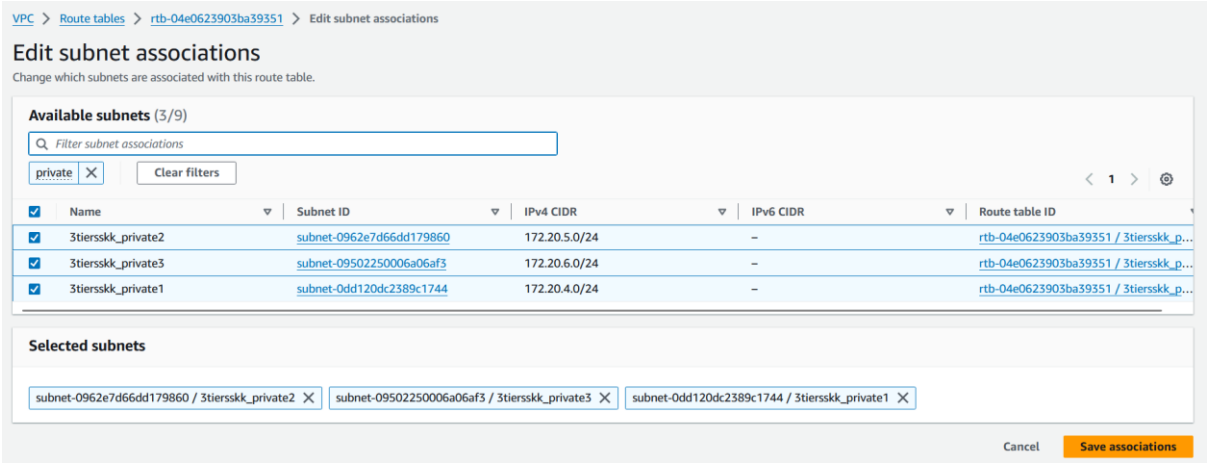


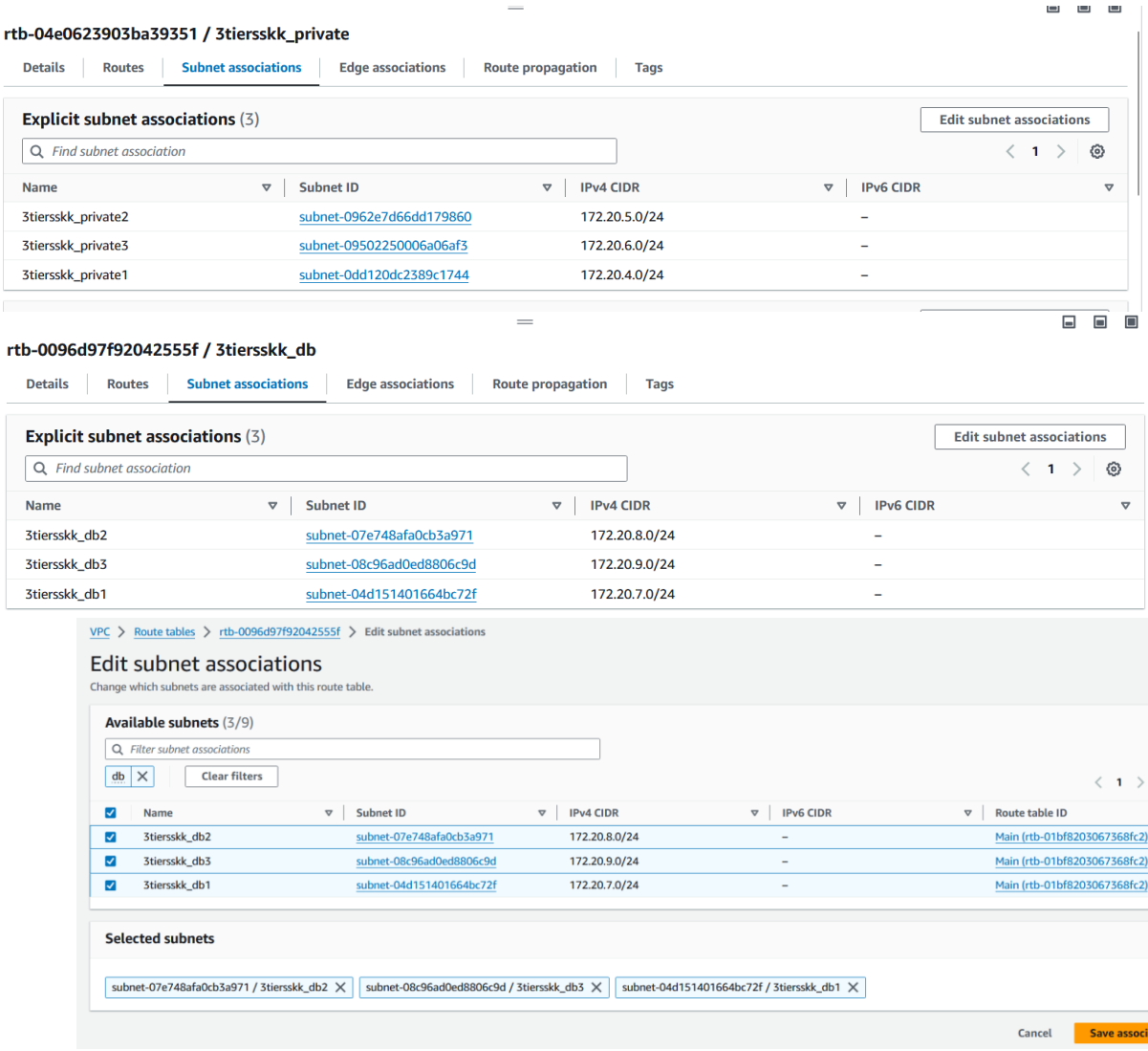
3.5.2. Associate route tables with subnets

- After creating the route table, select it from the list of route tables.
- Click on the "Subnet Associations" tab.
- Click on the "Edit subnet associations" button.
- Select the public subnet(s) that you want to associate with this route table.
- Click on "Save"



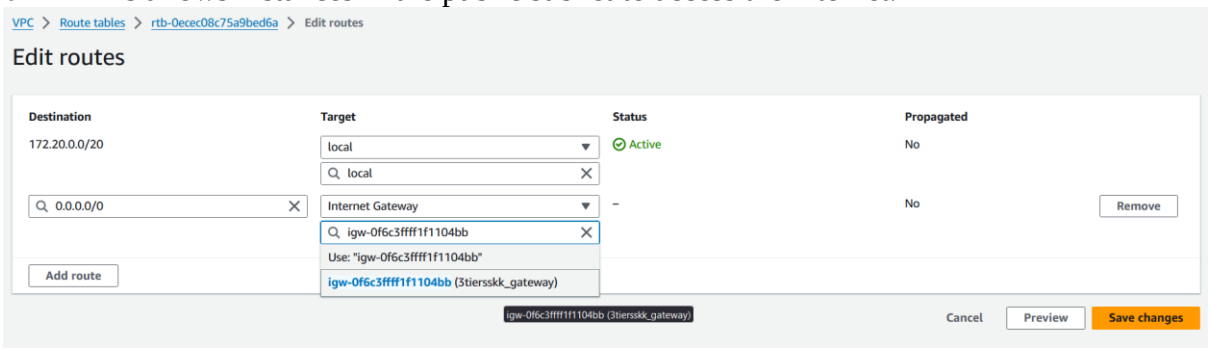




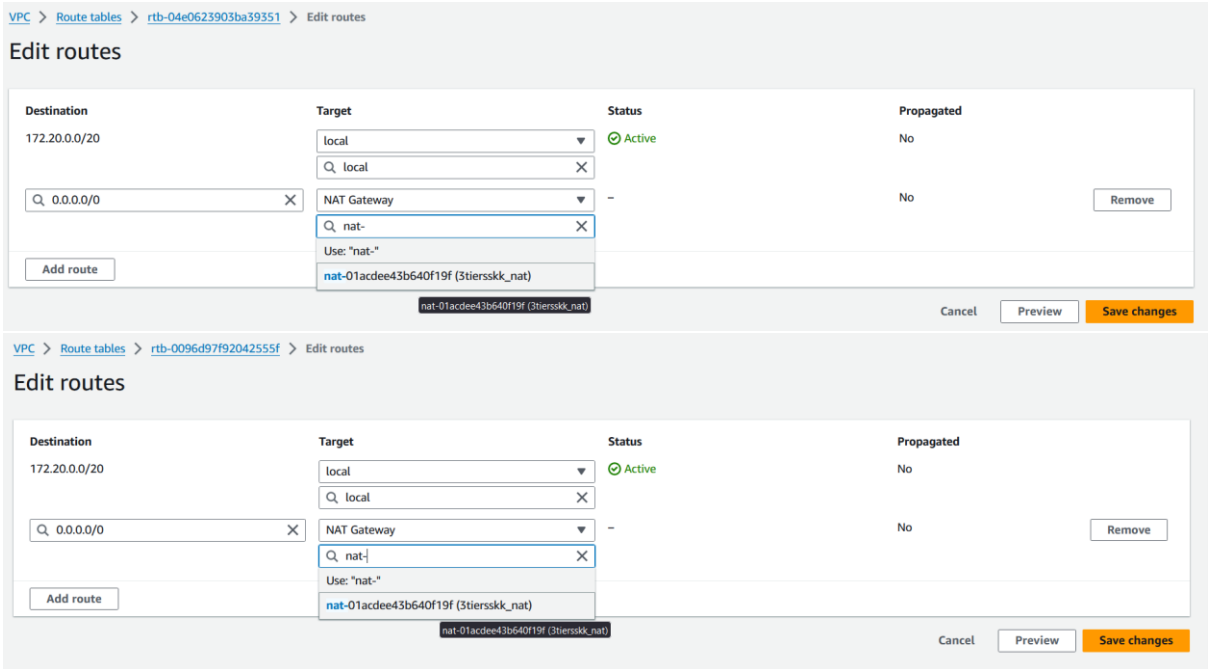


3.5.3. Edit route tables

- **Public Subnet Route Table:**
  - In the route table associated with the public subnet:
  - Add a route for internet-bound traffic (0.0.0.0/0) pointing to the internet gateway.
  - This allows instances in the public subnet to access the internet.



- **Private Subnet Route Table:**
  - In the route table associated with the private subnet:
  - Add a route for internet-bound traffic (0.0.0.0/0) pointing to the NAT gateway.
  - This enables instances in the private subnet to access the internet through the NAT gateway.

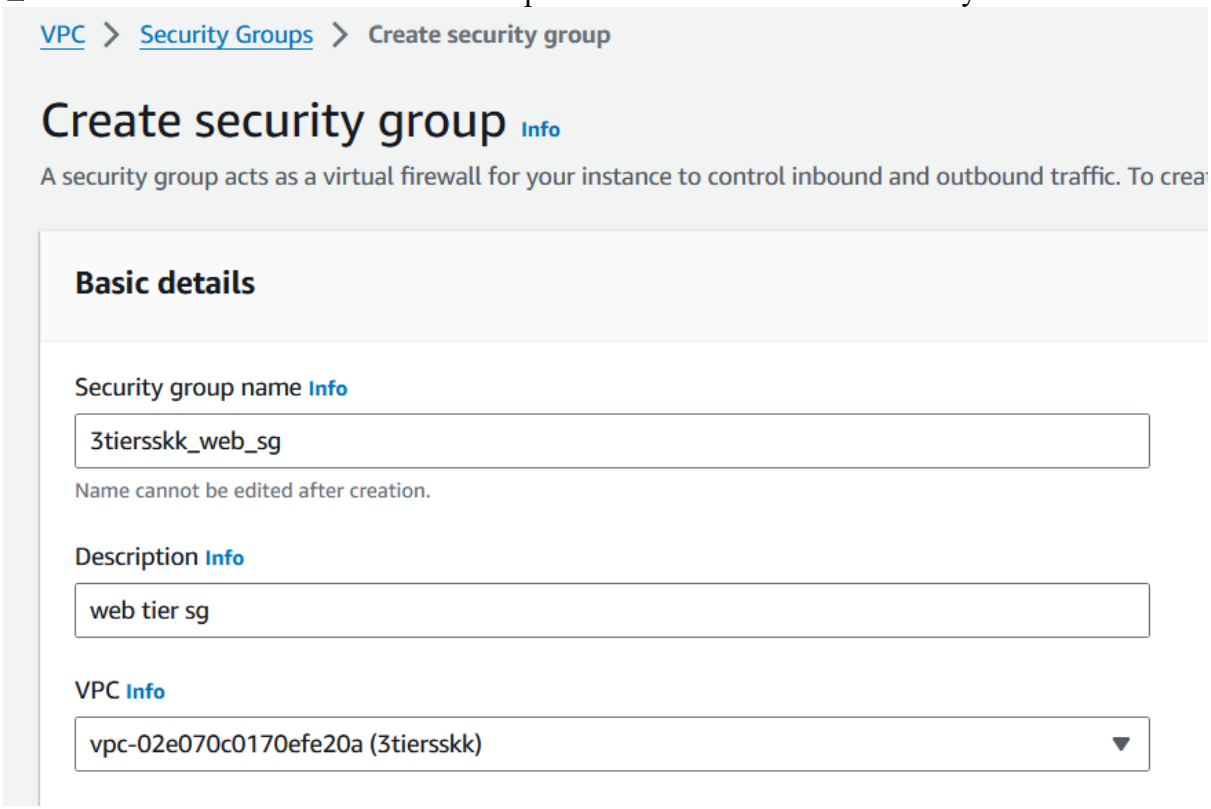


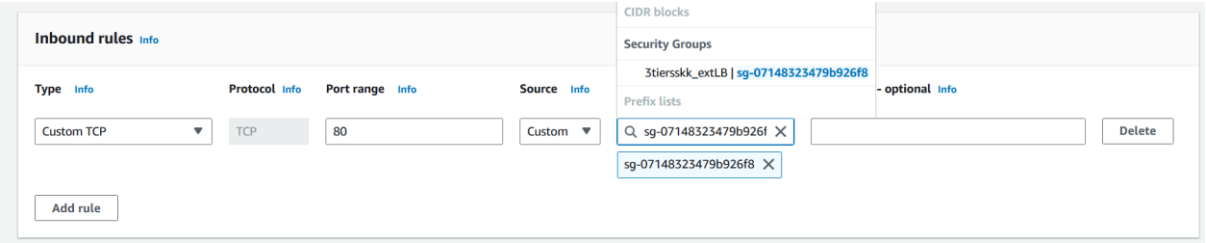
3.6. Security measures (Security Groups)

3.6.1. Create security groups

Security Group for Public Subnet

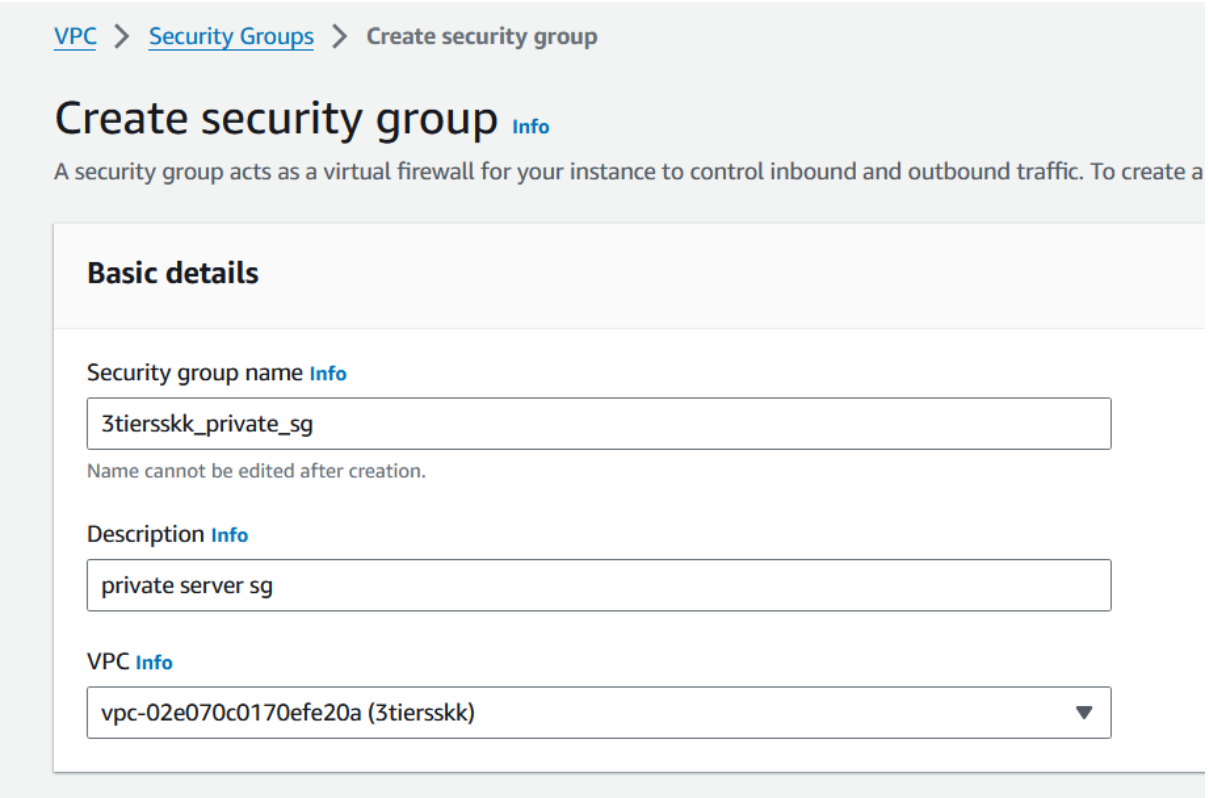
- Select the VPC that your public subnet is part of.
- In the "Inbound rules" section, define rules to allow inbound traffic to your public resources.
- Leave the "Outbound rules" section open to allow all outbound traffic by default.

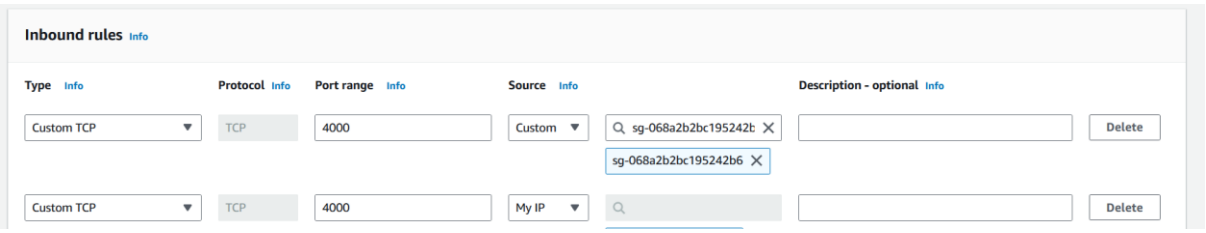




Security Group for Private Subnet

- Select the same VPC that your private subnet is part of.
- Define inbound rules to allow the public subnet security group to communicate with private subnet, avoid internet traffic for security purposes.
- Leave outbound rules open to allow all outbound traffic by default.





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 crea

Basic details

Security group name Info

3tiersskk_DB_sg

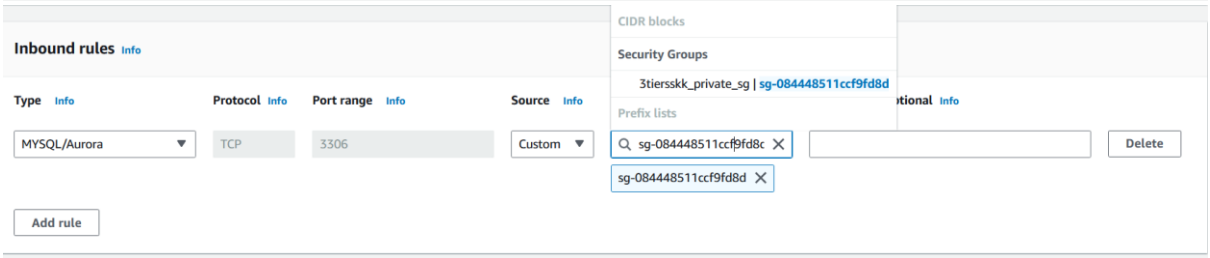
Name cannot be edited after creation.

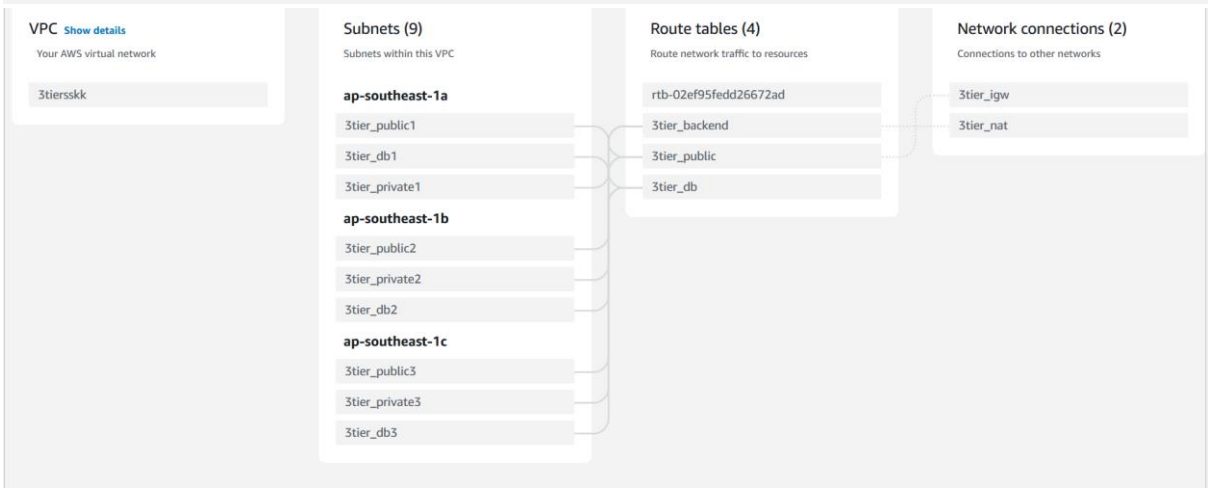
Description Info

sg for database

VPC Info

vpc-02e070c0170efe20a (3tiersskk)

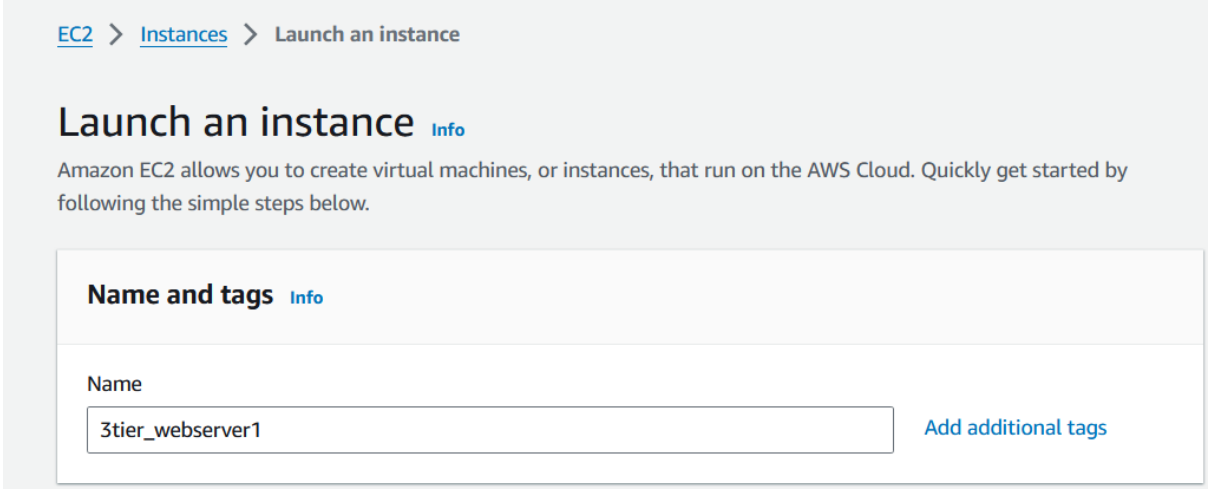




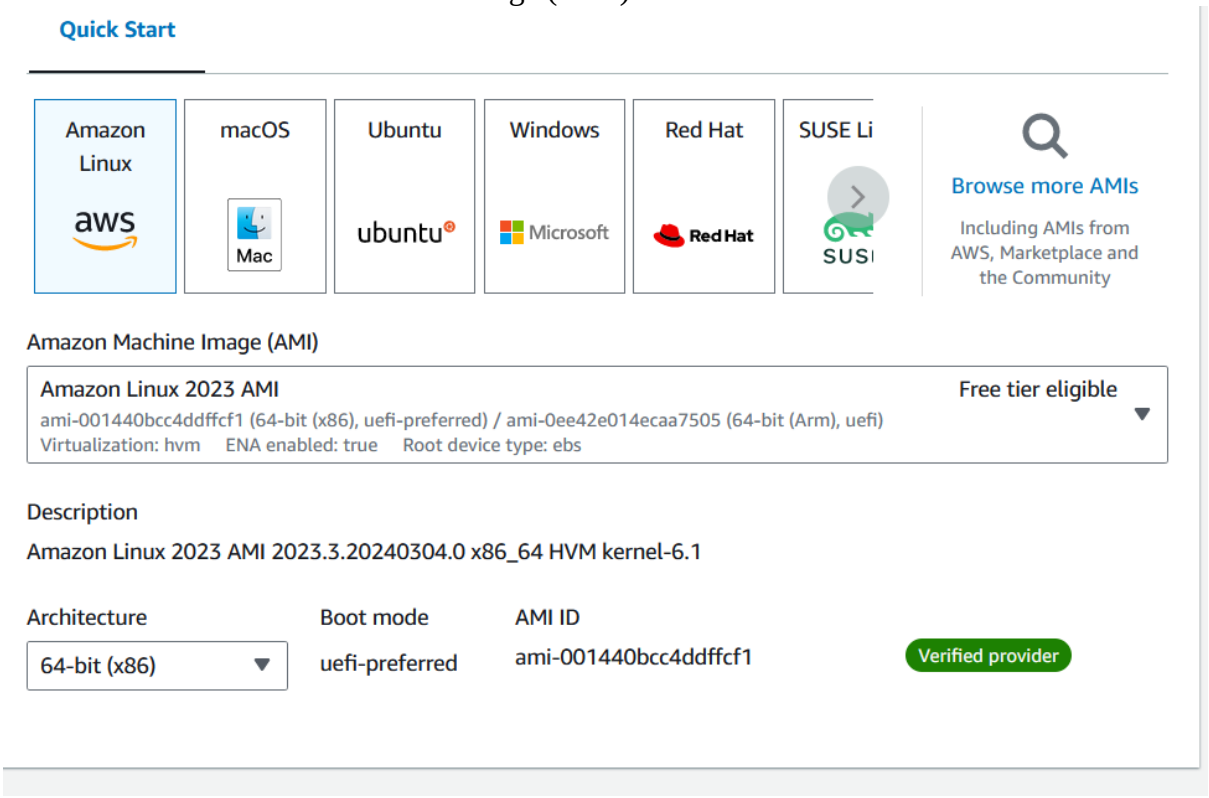
4. Frontend Tier

4.1. Creation of the EC2 instance in the public subnet

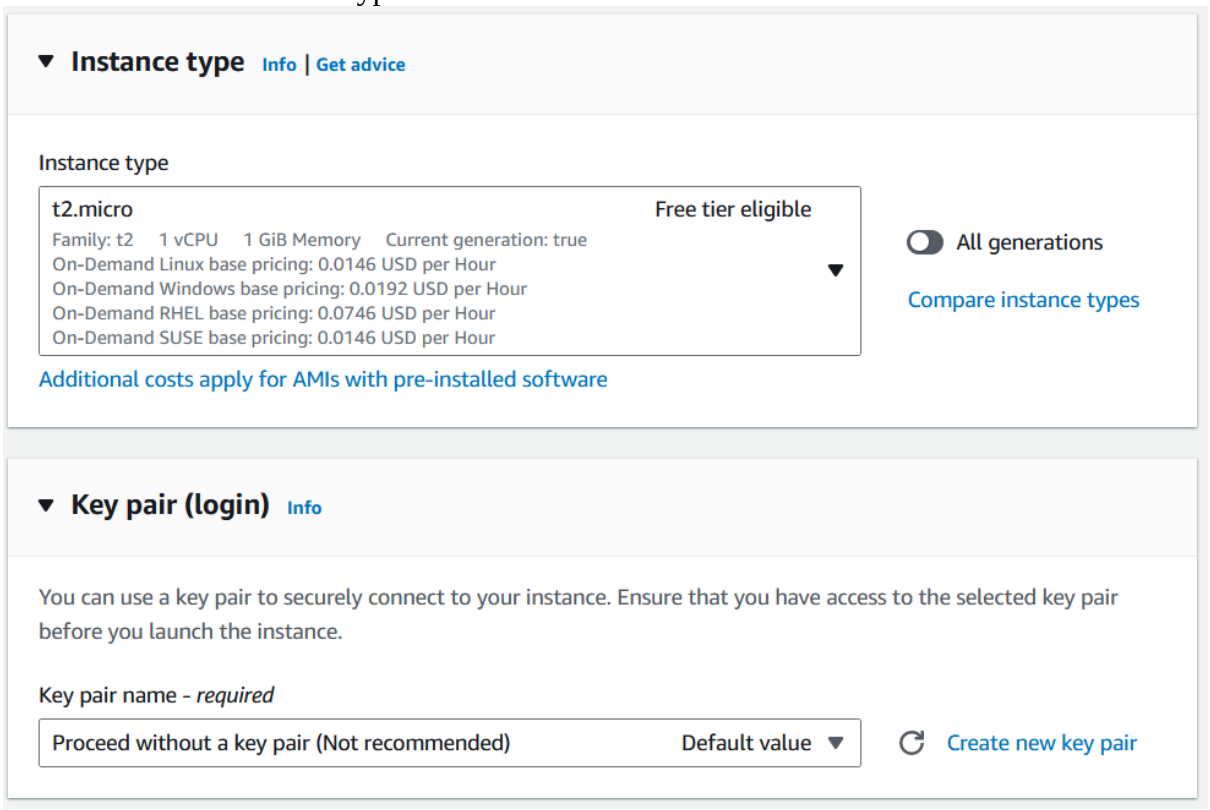
- Go to the EC2 Dashboard
- Launch Instance



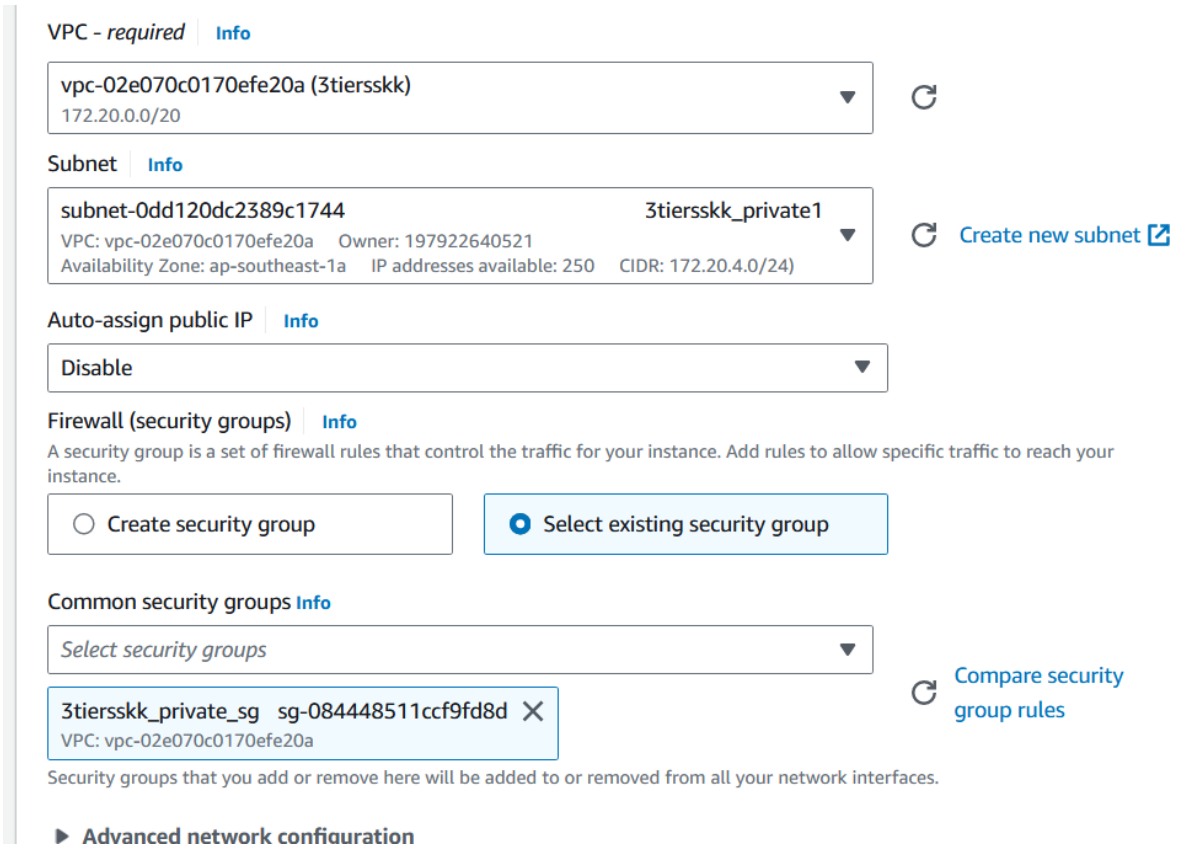
- Choose an Amazon Machine Image (AMI)

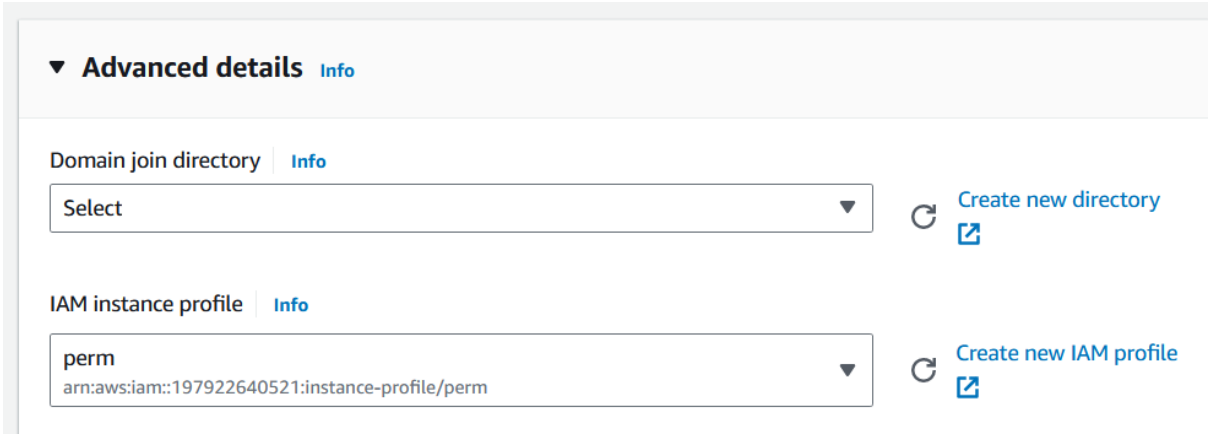


- Choose an Instance Type



- Configure Instance Details , Security Group and IAM roles

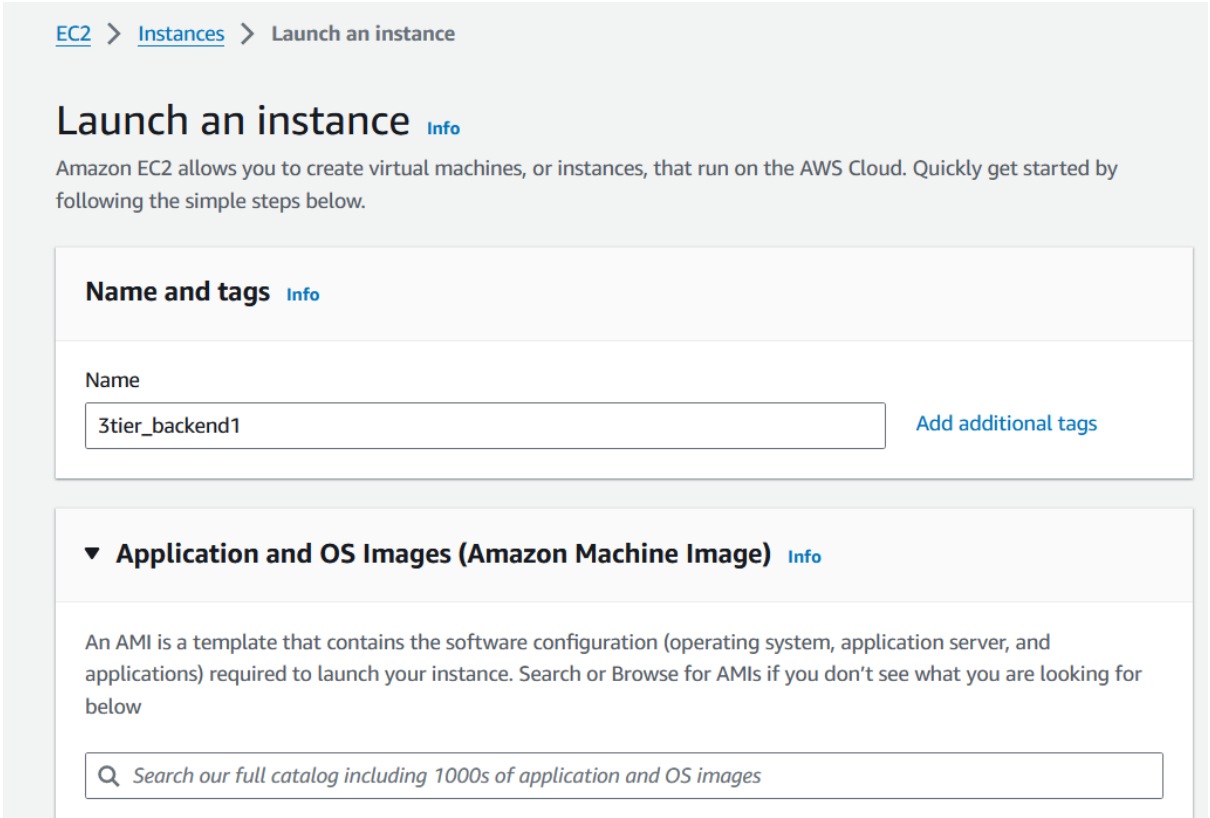




5. Application Tier

5.1. Creation of the two EC2 instances in private subnets

Follow the same steps as above and the keypair should be the same as the keypair for public subnet, this step allows the public subnet to connect with the private instance through ssh.



Key pair (login) Info

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

Key pair name - *required*

3tier_key

Create new key pair

VPC - *required* Info

vpc-02e070c0170efe20a (3tiersskk)
172.20.0.0/20

Subnet Info

subnet-0dd120dc2389c1744 3tiersskk_private1
VPC: vpc-02e070c0170efe20a Owner: 197922640521
Availability Zone: ap-southeast-1a IP addresses available: 251 CIDR: 172.20.4.0/24)

Create new subnet

Auto-assign public IP Info

Disable

Firewall (security groups) Info

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

(•) Create security group ( ) Select existing security group

Security group name - *required*

3tier_backend1

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

Description - *required* Info

3tier_backend1 created 2024-03-09T08:12:54.330Z

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Source Info	Description - *optional* Info
Custom	Add CIDR, prefix list or security	allow only from server
	sg-06fccb9717e6da459	

Add security group rule

▶ Advanced network configuration

EC2 > Instances > Launch an instance

Launch an instance Info

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

Name and tags Info

Name

3tier_backend2

Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

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

Key pair (login) Info

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

Key pair name - *required*

3tier_key

Create new key pair

VPC - *required* Info

vpc-02e070c0170efe20a (3tiersskk)
172.20.0.0/20

Subnet Info

subnet-0962e7d66dd179860 3tiersskk_private2
VPC: vpc-02e070c0170efe20a Owner: 197922640521
Availability Zone: ap-southeast-1b IP addresses available: 251 CIDR: 172.20.5.0/24)

Create new subnet

Auto-assign public IP Info

Disable

Firewall (security groups) Info

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

Create security group

Select existing security group

Common security groups Info

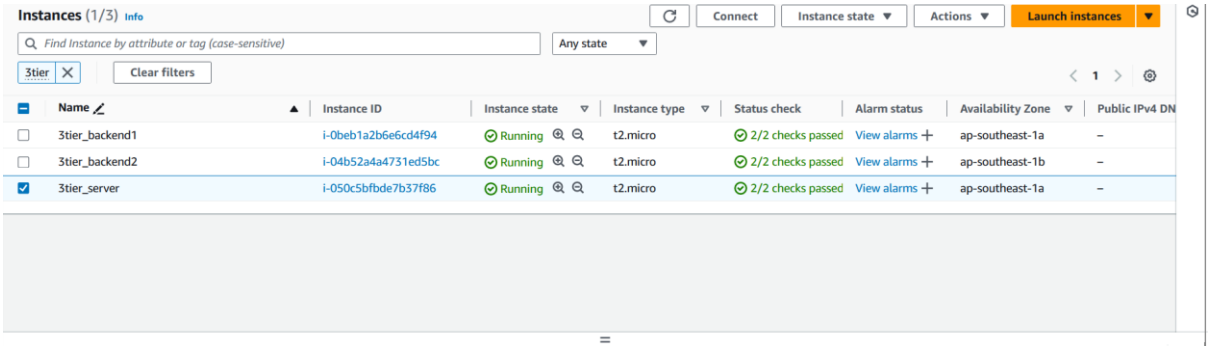
Select security groups

3tier_backend1 sg-0abb41151fda30566
VPC: vpc-02e070c0170efe20a

Compare security group rules

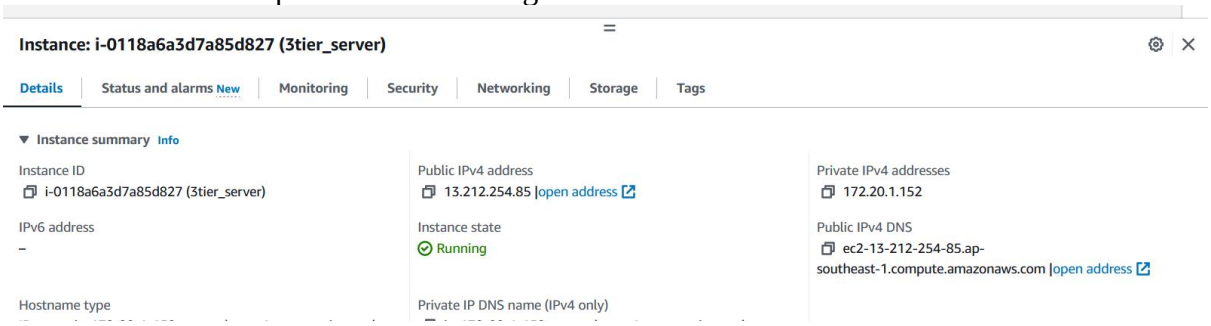
Security groups that you add or remove here will be added to or removed from all your network interfaces.

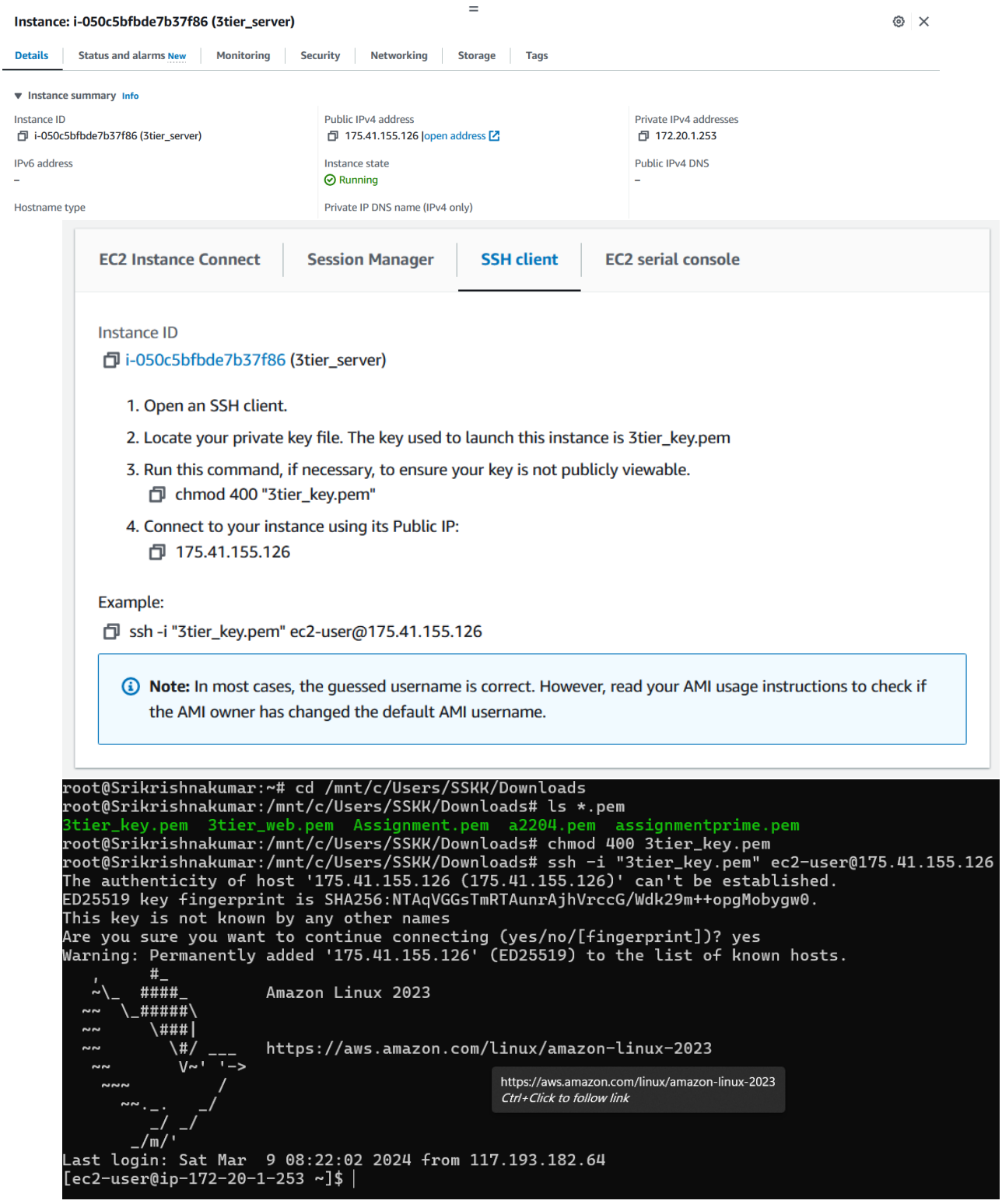
▶ Advanced network configuration



5.2. Configuring the application tier (business logic, processing)

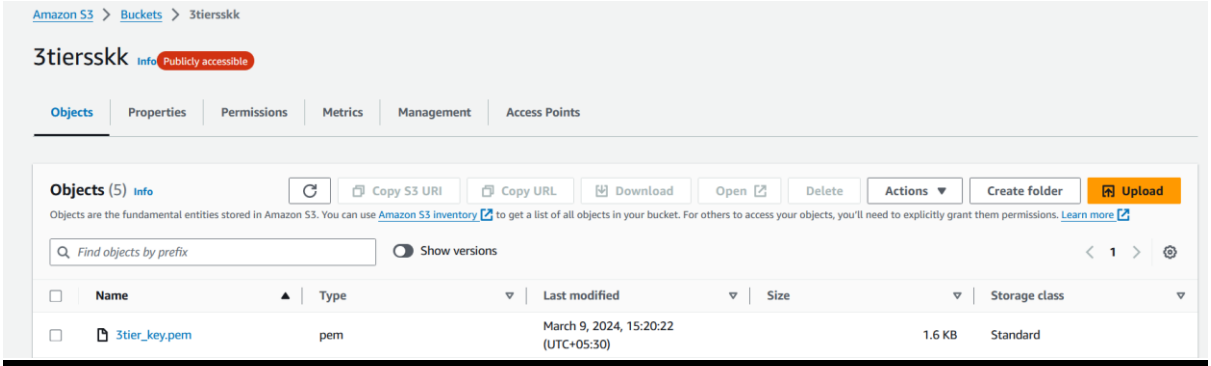
- Connect to the public subnet through ssh or Instance connect




Instance: i-050c5bfbde7b37f86 (3tier_server)
Details
Status and alarms New
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-050c5bfbde7b37f86 (3tier_server)
Public IPv4 address
175.41.155.126 |open address
Private IPv4 addresses
172.20.1.253
IPv6 address
–
Instance state
Running
Public IPv4 DNS
–
Hostname type
Private IP DNS name (IPv4 only)
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-050c5bfbde7b37f86 (3tier_server)
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is 3tier_key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "3tier_key.pem"
4. Connect to your instance using its Public IP:
175.41.155.126
Example:
ssh -i "3tier_key.pem" ec2-user@175.41.155.126
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
root@Srikrishnakumar:~# cd /mnt/c/Users/SSKK/Downloads
root@Srikrishnakumar:/mnt/c/Users/SSKK/Downloads# ls *.pem
3tier_key.pem 3tier_web.pem Assignment.pem a2204.pem assignmentprime.pem
root@Srikrishnakumar:/mnt/c/Users/SSKK/Downloads# chmod 400 3tier_key.pem
root@Srikrishnakumar:/mnt/c/Users/SSKK/Downloads# ssh -i "3tier_key.pem" ec2-user@175.41.155.126
The authenticity of host '175.41.155.126 (175.41.155.126)' can't be established.
ED25519 key fingerprint is SHA256:NTAqVGGsTmRTAunrAjhVrccG/Wdk29m++opgMobygw0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '175.41.155.126' (ED25519) to the list of known hosts.
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
https://aws.amazon.com/linux/amazon-linux-2023
Ctrl+Click to follow link
Last login: Sat Mar 9 08:22:02 2024 from 117.193.182.64
[ec2-user@ip-172-20-1-253 ~]$

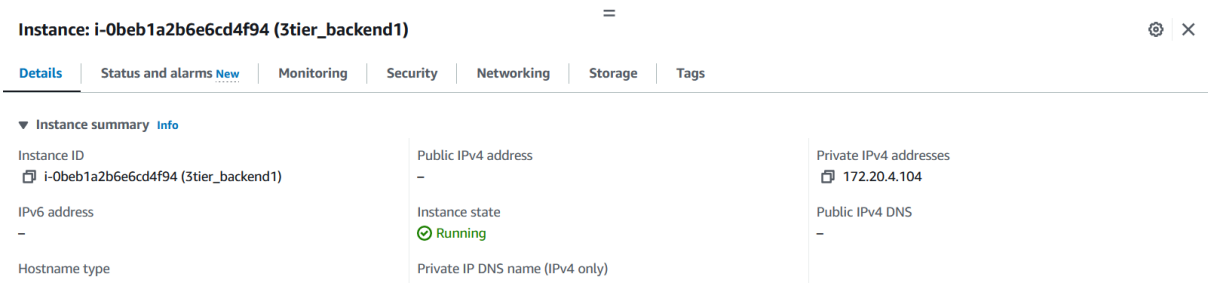
```
root@Srikrishnakumar:~# cd /mnt/c/Users/SSKK/Downloads
root@Srikrishnakumar:/mnt/c/Users/SSKK/Downloads# ls *.pem
3tier_key.pem  3tier_web.pem  Assignment.pem  a2204.pem  assignmentprime.pem
root@Srikrishnakumar:/mnt/c/Users/SSKK/Downloads# chmod 400 3tier_key.pem
root@Srikrishnakumar:/mnt/c/Users/SSKK/Downloads# ssh -i "3tier_key.pem" ec2-user@175.41.155.126
The authenticity of host '175.41.155.126 (175.41.155.126)' can't be established.
ED25519 key fingerprint is SHA256:NTAqVGGsTmRTAunrAjhVrccG/Wdk29m++opgMobygw0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '175.41.155.126' (ED25519) to the list of known hosts.
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sat Mar  9 08:22:02 2024 from 117.193.182.64
[ec2-user@ip-172-20-1-253 ~]$
```

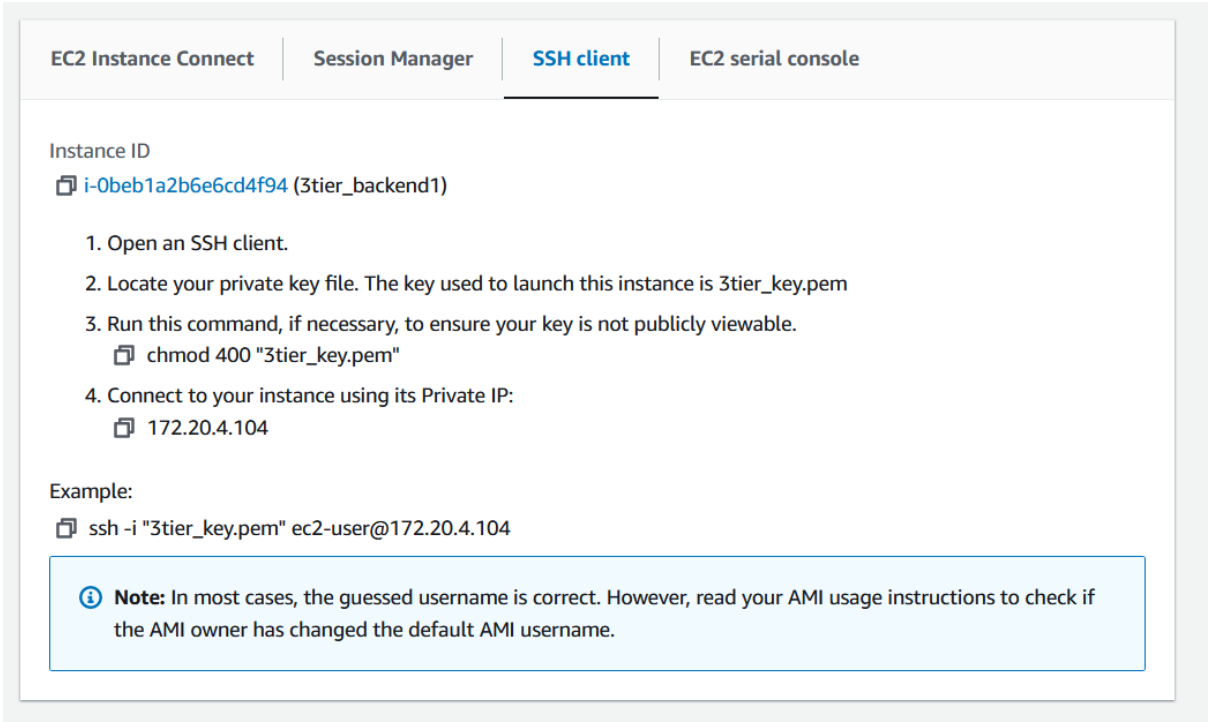
- The keypair is stored in a separate location (S3 bucket) and downloaded to the public subnet. Storing the key in an internet accessible source enables using the key from any location.

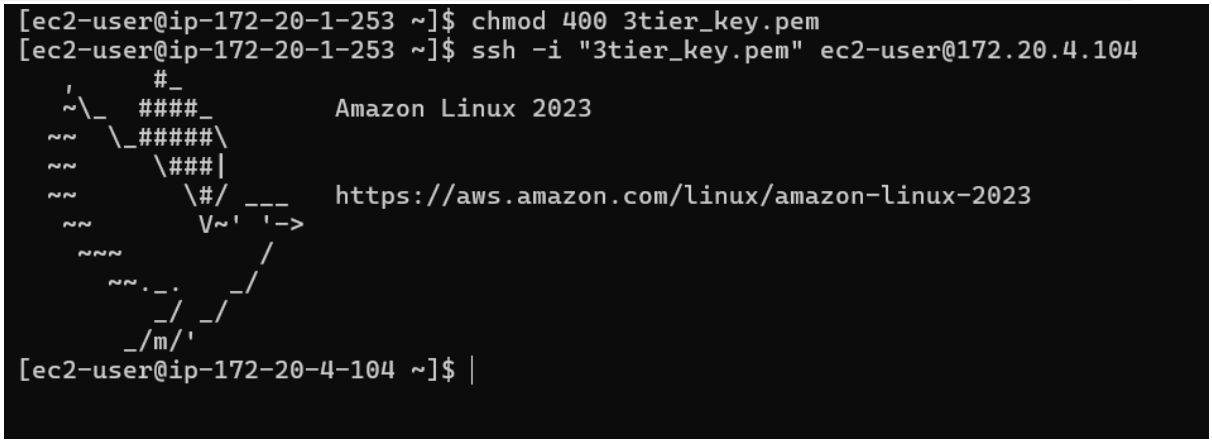



```
[ec2-user@ip-172-20-1-253 ~]$ aws s3 cp s3://3tiersskk/3tier_key.pem /home/ec2-user
download: s3://3tiersskk/3tier_key.pem to ./3tier_key.pem
[ec2-user@ip-172-20-1-253 ~]$ ls
3tier_key.pem
[ec2-user@ip-172-20-1-253 ~]$
```

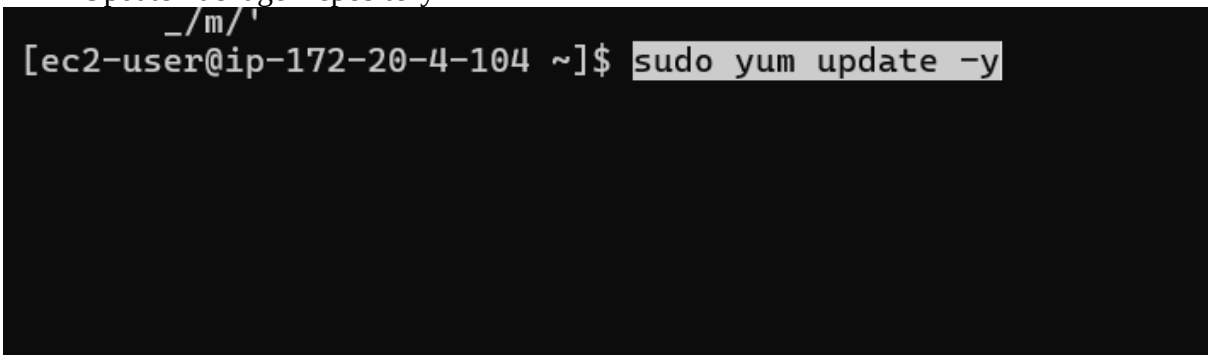
- Connect to the private subnet from public subnet through ssh or Instance connect







- Update Package Repository



```
sskk1987@Srikrishnakumar:~/Downloads$ ssh -i "3tier_key.pem" ec2-user@175.41.155.126
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sat Mar  9 09:52:20 2024 from 3.0.5.36
[ec2-user@ip-172-20-1-253 ~]$ ls
3tier_key.pem
[ec2-user@ip-172-20-1-253 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-20-1-253 ~]$ chmod 400 3tier_key.pem
[ec2-user@ip-172-20-1-253 ~]$ ssh -i "3tier_key.pem" ec2-user@172.20.4.104
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sat Mar  9 09:55:19 2024 from 172.20.1.253
[ec2-user@ip-172-20-4-104 ~]$
```

- Install Apache Web Server

```
[ec2-user@ip-172-20-4-104 ~]$ sudo yum update -y
Last metadata expiration check: 1:56:20 ago on Sat Mar  9 08:18:19 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-20-4-104 ~]$ sudo yum install httpd -y
Last metadata expiration check: 1:56:26 ago on Sat Mar  9 08:18:19 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture            Version                 Repository              Size
================================================================================
Installing:
 httpd                   x86_64                  2.4.58-1.amzn2023       amazonlinux             47 k
Installing dependencies:
 apr                     x86_64                  1.7.2-2.amzn2023.0.2    amazonlinux            129 k
```

```
Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64                  apr-util-1.6.3-1.amzn2023.0.1.x86_64   apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.58-1.amzn2023.x86_64       httpd-core-2.4.58-1.amzn2023.x86_64
  httpd-filesystem-2.4.58-1.amzn2023.noarch        httpd-tools-2.4.58-1.amzn2023.x86_64   libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch             mod_http2-2.0.11-2.amzn2023.x86_64     mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-20-4-104 ~]$
```

- After installation, start the Apache service and enable it to start on boot:

```
[ec2-user@ip-172-20-4-104 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-20-4-104 ~]$ sudo systemctl enable httpd
```

```
[ec2-user@ip-172-20-4-104 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sat 2024-03-09 10:18:01 UTC; 3min 48s ago
       Docs: man:httpd.service(8)
   Main PID: 28655 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1114)
     Memory: 13.2M
        CPU: 197ms
     CGroup: /system.slice/httpd.service
             ├─28655 /usr/sbin/httpd -DFOREGROUND
             ├─28656 /usr/sbin/httpd -DFOREGROUND
             ├─28658 /usr/sbin/httpd -DFOREGROUND
             ├─28659 /usr/sbin/httpd -DFOREGROUND
             └─28660 /usr/sbin/httpd -DFOREGROUND

Mar 09 10:18:01 ip-172-20-4-104.ap-southeast-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Mar 09 10:18:01 ip-172-20-4-104.ap-southeast-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Mar 09 10:18:01 ip-172-20-4-104.ap-southeast-1.compute.internal httpd[28655]: Server configured, listening on: port 80
[ec2-user@ip-172-20-4-104 ~]$
```

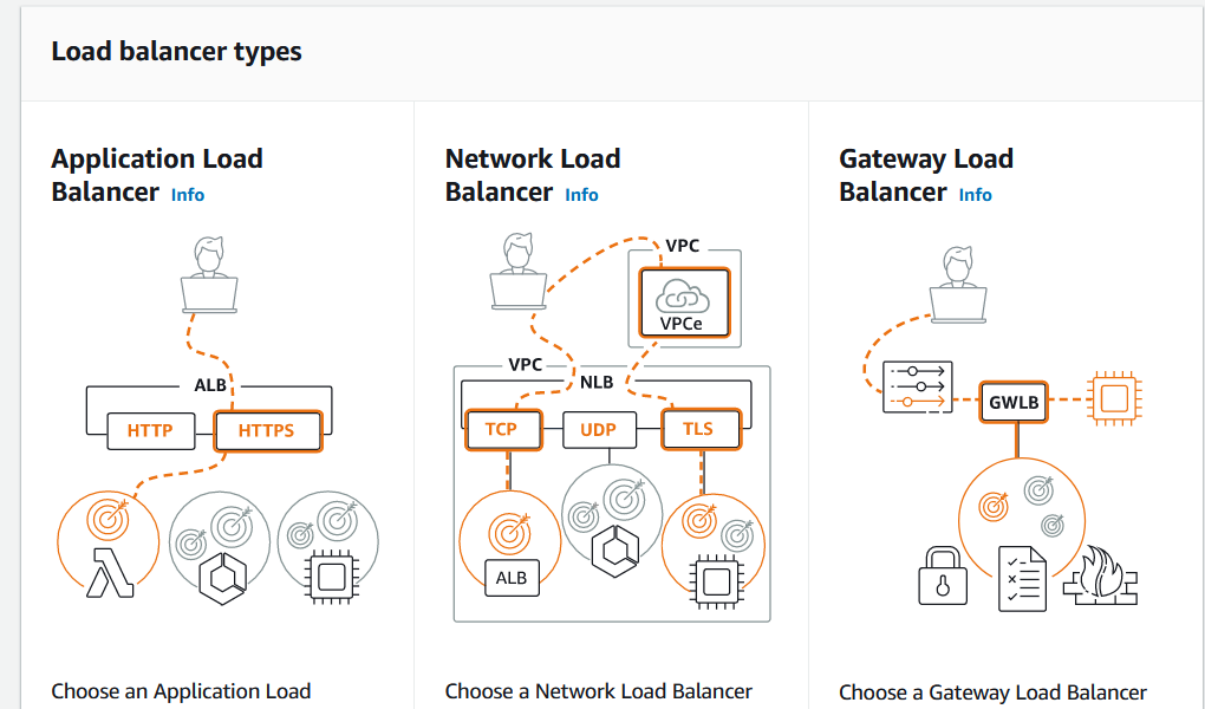
```
Installed:
  libsodium-1.0.18-13.amzn2023.0.1.x86_64     libxslt-1.1.34-5.amzn2023.0.2.x86_64
  php8.2-8.2.15-1.amzn2023.0.1.x86_64         php8.2-cli-8.2.15-1.amzn2023.0.1.x86_64
  php8.2-fpm-8.2.15-1.amzn2023.0.1.x86_64     php8.2-mbstring-8.2.15-1.amzn2023.0.1.x86_6
  php8.2-pdo-8.2.15-1.amzn2023.0.1.x86_64     php8.2-process-8.2.15-1.amzn2023.0.1.x86_64
  php8.2-xml-8.2.15-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-20-4-104 ~]$ sudo nano /etc/httpd/conf.d/phpMyAdmin.conf
[ec2-user@ip-172-20-4-104 ~]$ sudo systemctl restart httpd
[ec2-user@ip-172-20-4-104 ~]$
```

5.3. Load balancing setup using Application Load Balancers (ALB)

5.3.1. Creating the Load balancer

- Provide a name for your load balancer.
- Select the VPC (Virtual Private Cloud) where you want to deploy the load balancer.
- Choose the availability zones where you want the load balancer to distribute traffic.



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.

Create

Create Application Load Balancer

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.

Create

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.

Create

▶ Classic Load Balancer - *previous generation*

Close

Basic configuration

Load balancer name

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

3tier

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

Scheme | Info

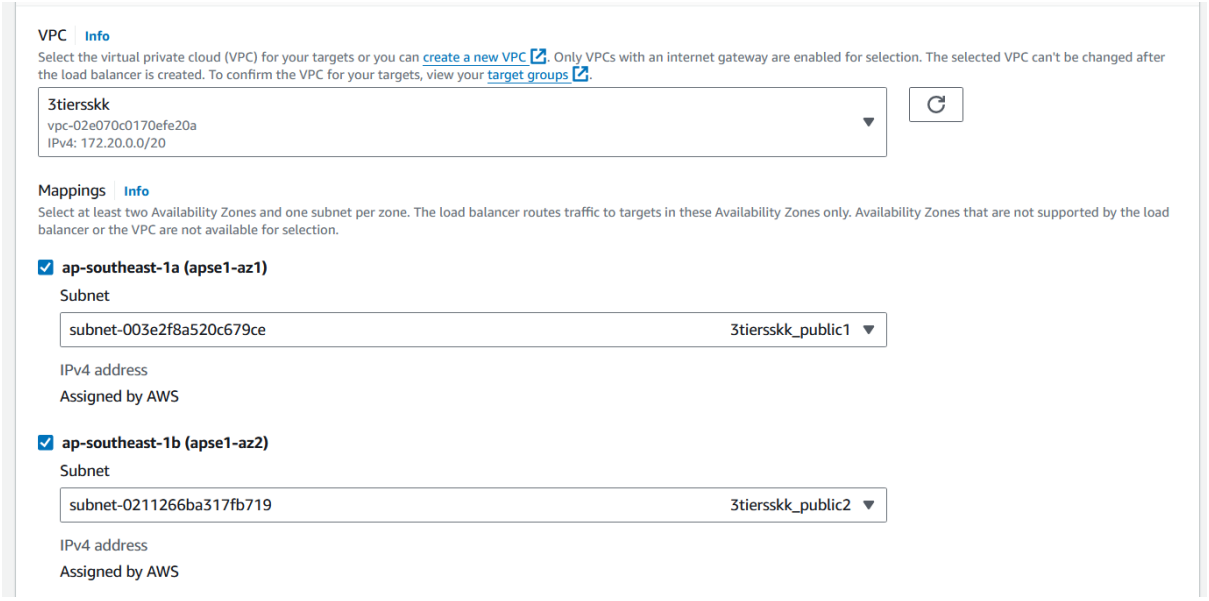
Scheme can't be changed after the load balancer is created.

- (●) Internet-facing
  An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
- ( ) Internal
  An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type | Info

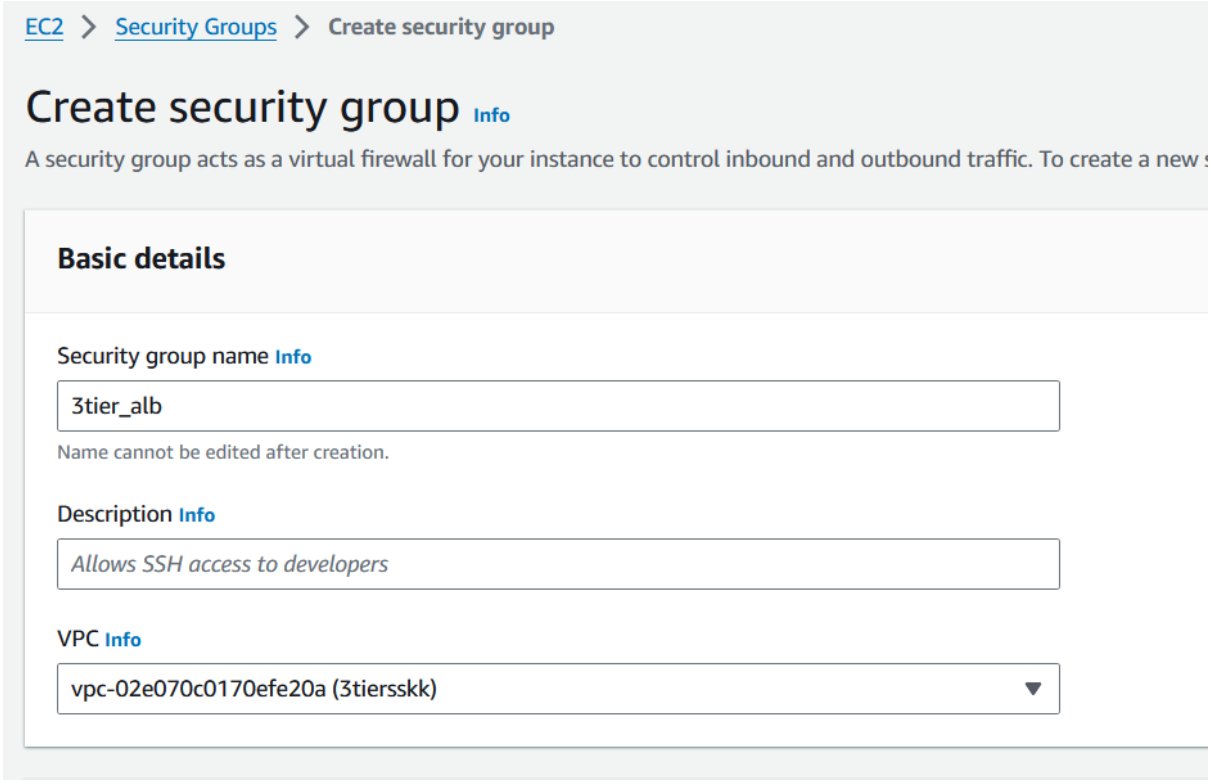
Select the type of IP addresses that your subnets use.

- (●) IPv4
  Recommended for internal load balancers.
- ( ) Dualstack
  Includes IPv4 and IPv6 addresses.



5.3.2. Security group of backend to allow alb

- Configure security settings for your load balancer, including security groups and whether to enable deletion protection.
- Define target groups and specify the target instances (e.g., EC2 instances) that the load balancer will distribute traffic to.



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

3tier_alb
sg-0a1993d5e838b724a VPC: vpc-02e070c0170efe20a

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

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.

3tier_tg

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

Protocol : Port

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

HTTP | 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

Register targets

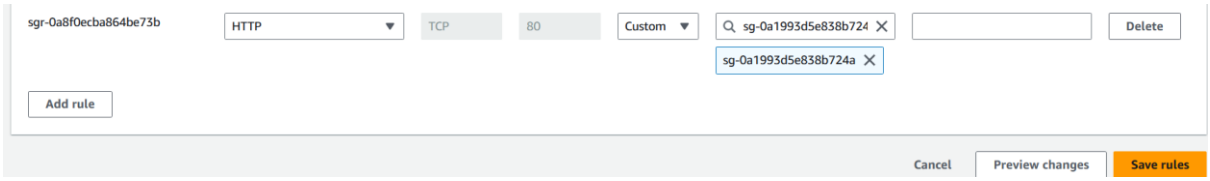
This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/3)

Filter instances

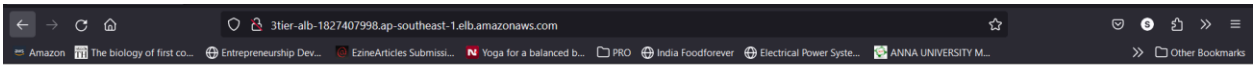
1

Instance ID	Name	State	Security groups	Zor
i-04b52a4a4731ed5bc	3tier_backend2	Running	3tier_backend1	ap-
i-0beb1a2b6e6cd4f94	3tier_backend1	Running	3tier_backend1	ap-
i-050c5bfbde7b37f86	3tier_server	Running	3tier_server	ap-

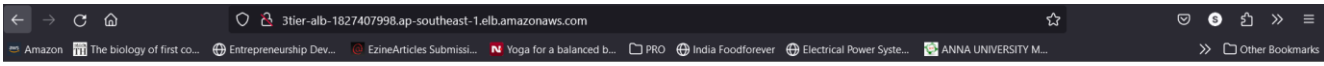


5.3.3. Load balancing between two servers

- The two private instances has been configured with web servers, now we have created html files to display different text to observe the load balancing between the two instances.



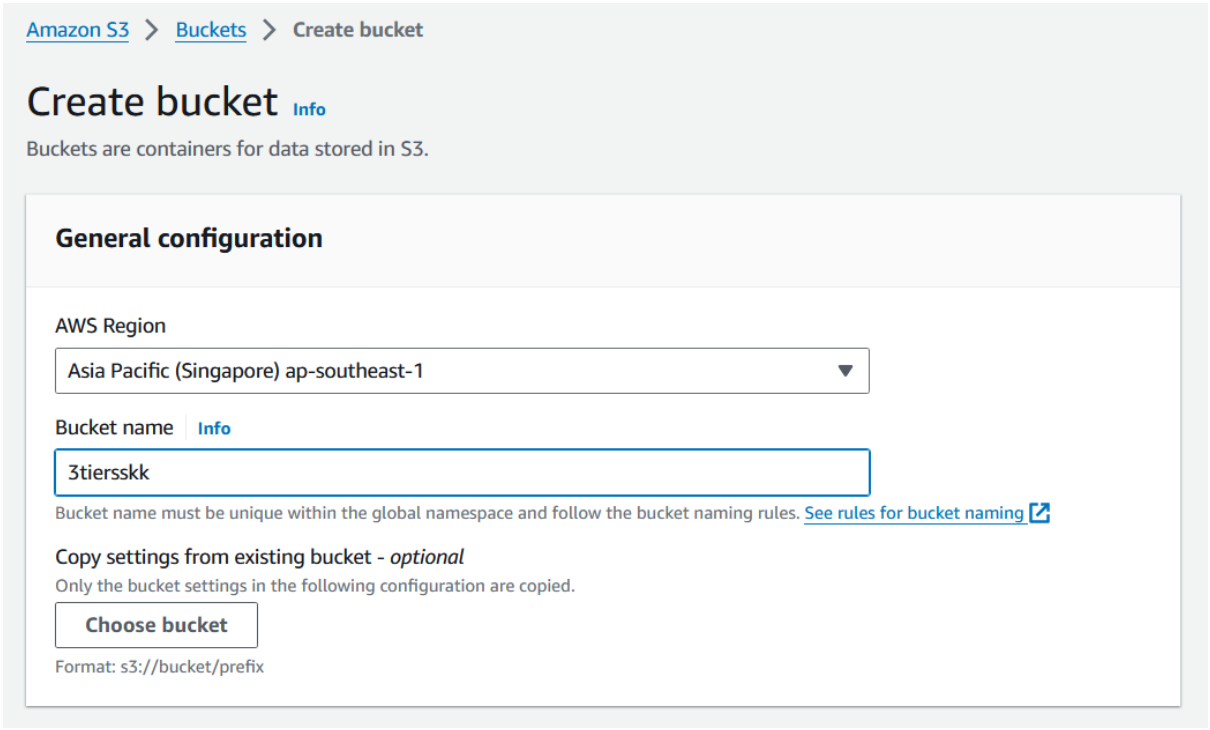
123456789



123456

5.3.4. Description of data storage in S3 bucket

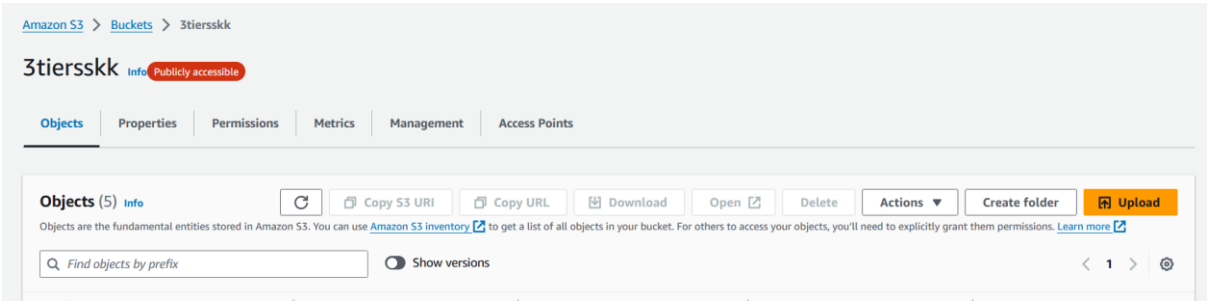
- S3 bucket offers a storage solution, and we use it to store the necessary html, css and javascript needed for websites.



Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- ☐ **Block *all* public access**
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
  - ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - ☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - ☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Trusted entity type

- ◉ **AWS service**
  Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ○ **AWS account**
  Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ○ **Web identity**
  Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ○ **SAML 2.0 federation**
  Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ○ **Custom trust policy**
  Create a custom trust policy to enable others to perform actions in this account.

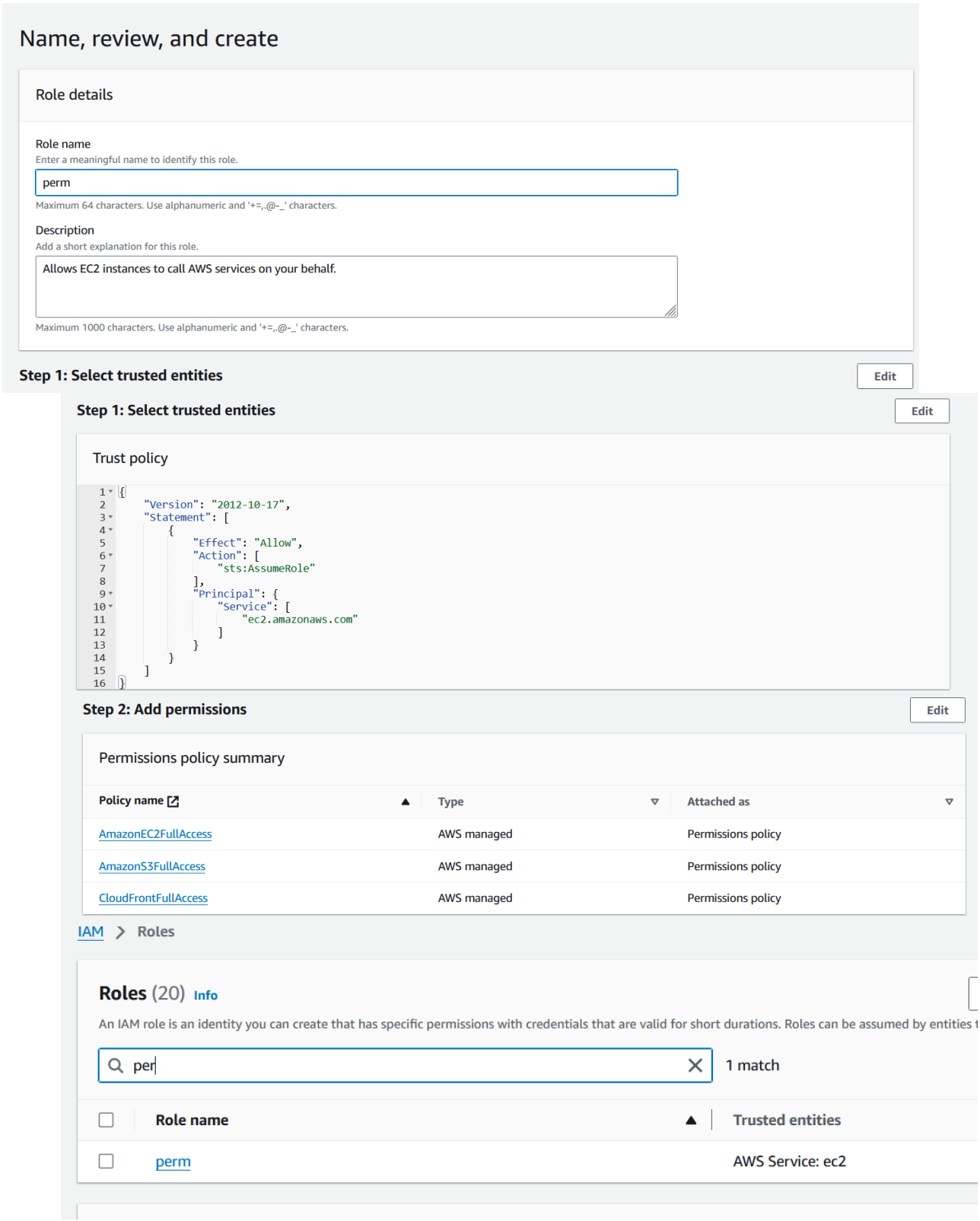
Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

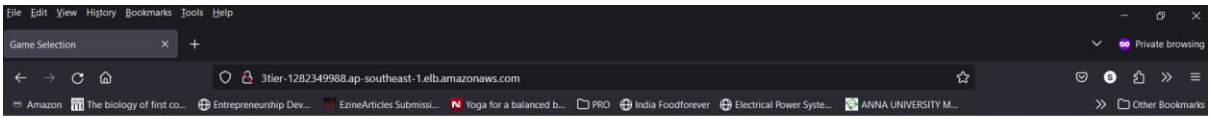
EC2

Choose a use case for the specified service.


Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
perm
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.
Step 1: Select trusted entities
Edit
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"sts:AssumeRole"
],
"Principal": {
"Service": [
"ec2.amazonaws.com"
]
}
}
]
}
Step 2: Add permissions
Edit
Permissions policy summary
Policy name
Type
Attached as
AmazonEC2FullAccess
AWS managed
Permissions policy
AmazonS3FullAccess
AWS managed
Permissions policy
CloudFrontFullAccess
AWS managed
Permissions policy
IAM
Roles
Roles (20) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities
per
1 match
Role name
Trusted entities
perm
AWS Service: ec2

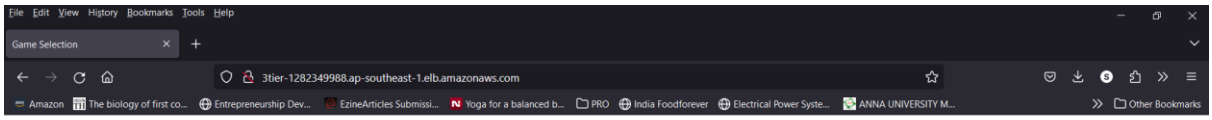
5.3.5. Load balancing between two servers fetching from S3 bucket

- To observe difference between the servers, the appearance of text in the landing page has been slightly modified.



Choose a Game

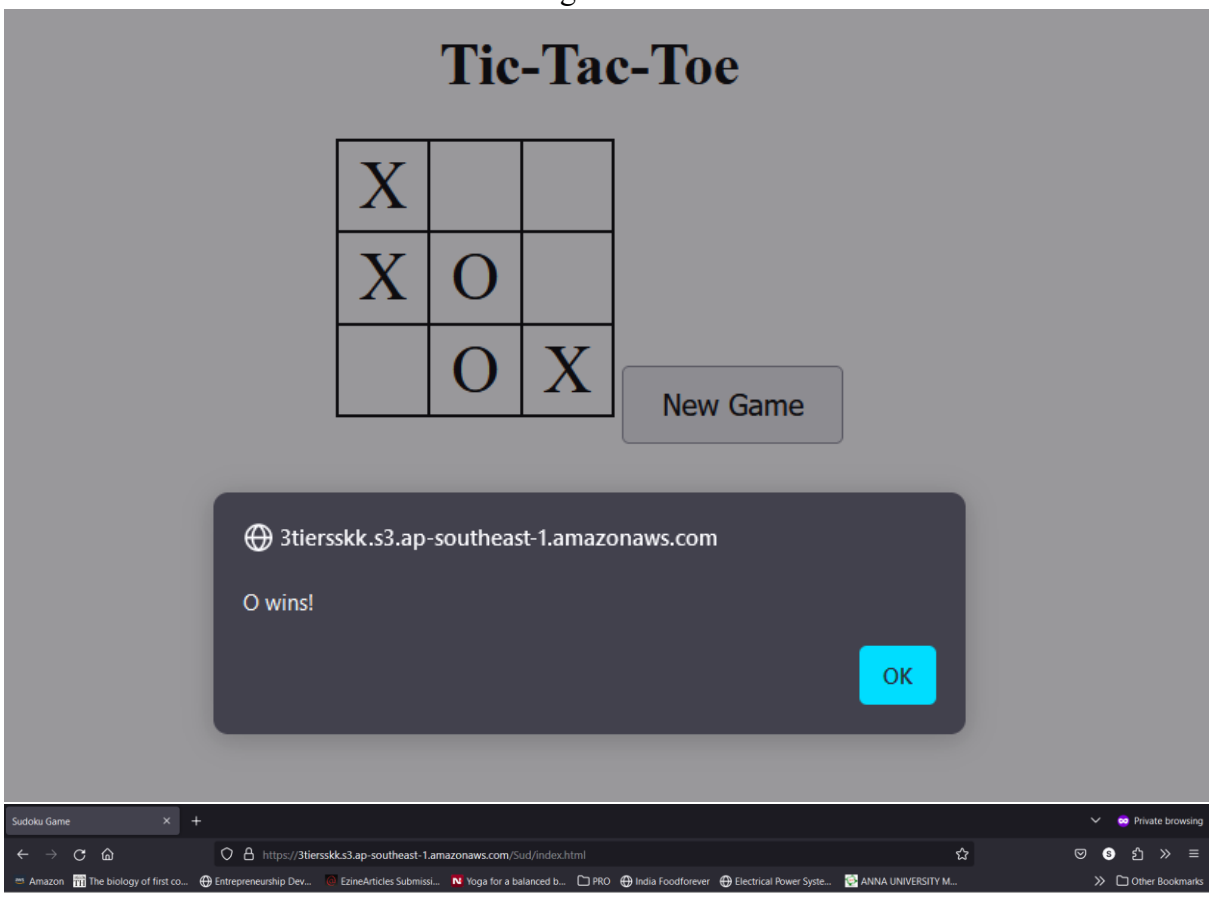
Tic-Tac-Toe Sudoku Fun Sudoku



CHOOSE A GAME

Tic-Tac-Toe Sudoku Fun Sudoku

- Both servers lead to the same game files served from S3 bucket



Sudoku Game

.	.	.	.	.	.	.	8	.
.	.	6	7	.	.	1	2	.
.	.	9	1	2	3	4	5	.
.	1	4	3	6	.	.	9	.
3	.	5	8	.	7	2	.	4
.	9	.	.	1	4	3	.	.
5	3	1	.	.	.	9	.	8
6	4	2	9	.	8	.	.	.
.	7	8	5	.	.	6	.	2

New Game Check Solution

Sudoku Game

New Game Solutions

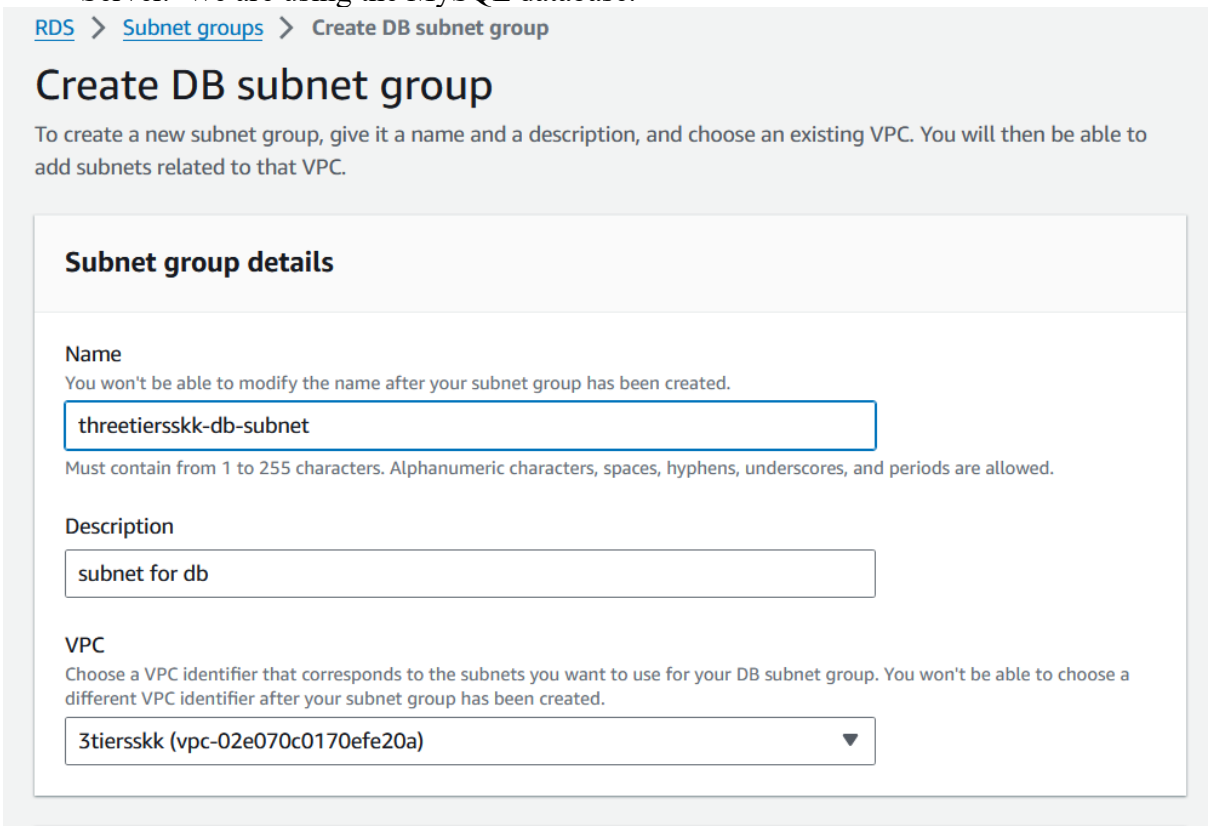
.	.	C	.	E	F	G	.	I
D	E	F	.	.	.	A	.	.
.	H	I	.	B	.	D	E	F
B	A	.	.	.	.	H	I	G
.	.	.	H	I	.	B	.	D
H	.	.	B	A	D	C	.	.
.	C	A	F	.	.	.	.	.
F	.	B	.	.	.	E	.	A
I	.	.	.	C	A	.	D	B

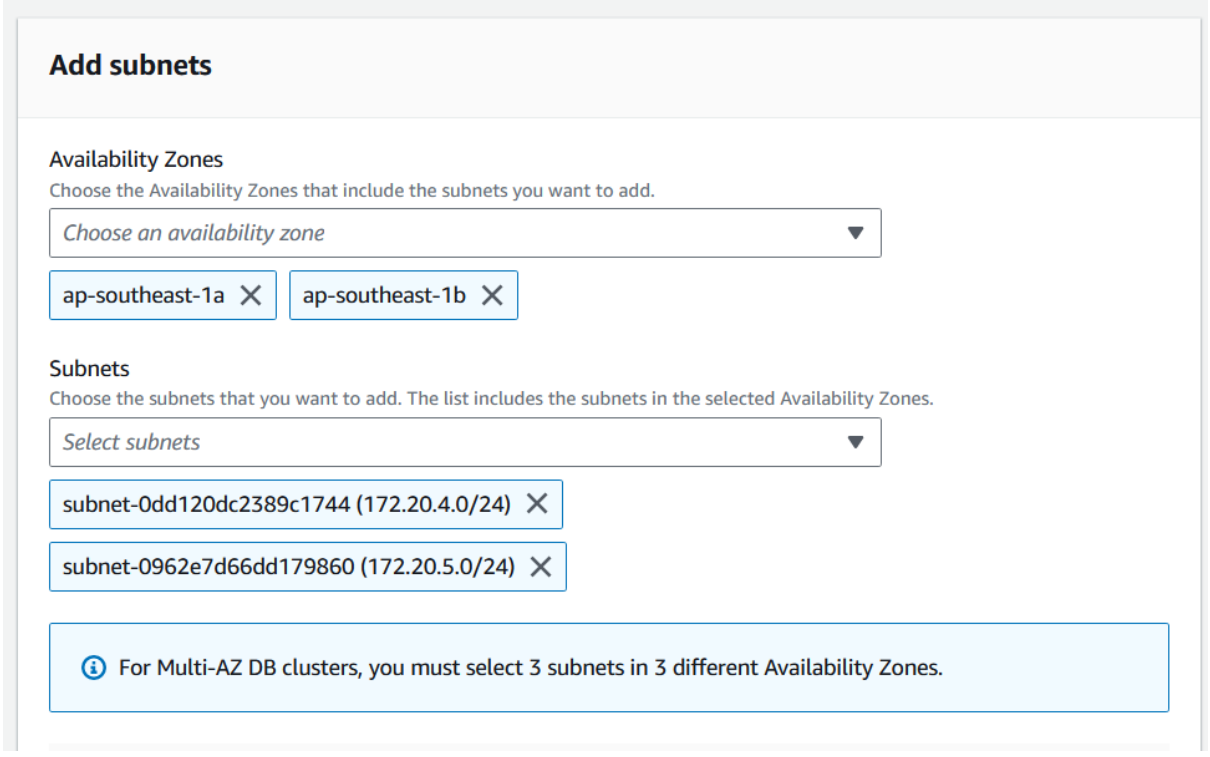
-

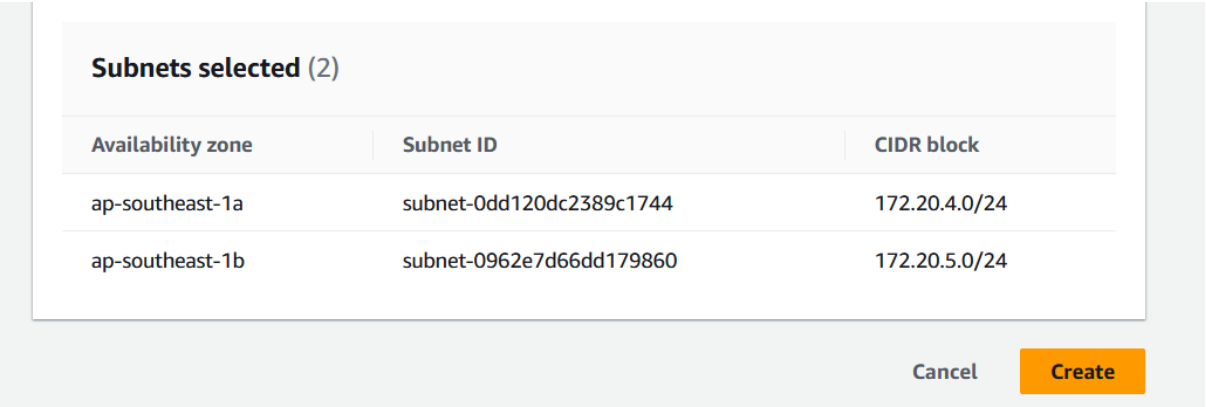
6. Adding database to create third tier

6.1. RDS Deployment

- Choose the database engine that you want to use for your database instance. AWS RDS supports various engines such as MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. We are using the MySQL database.

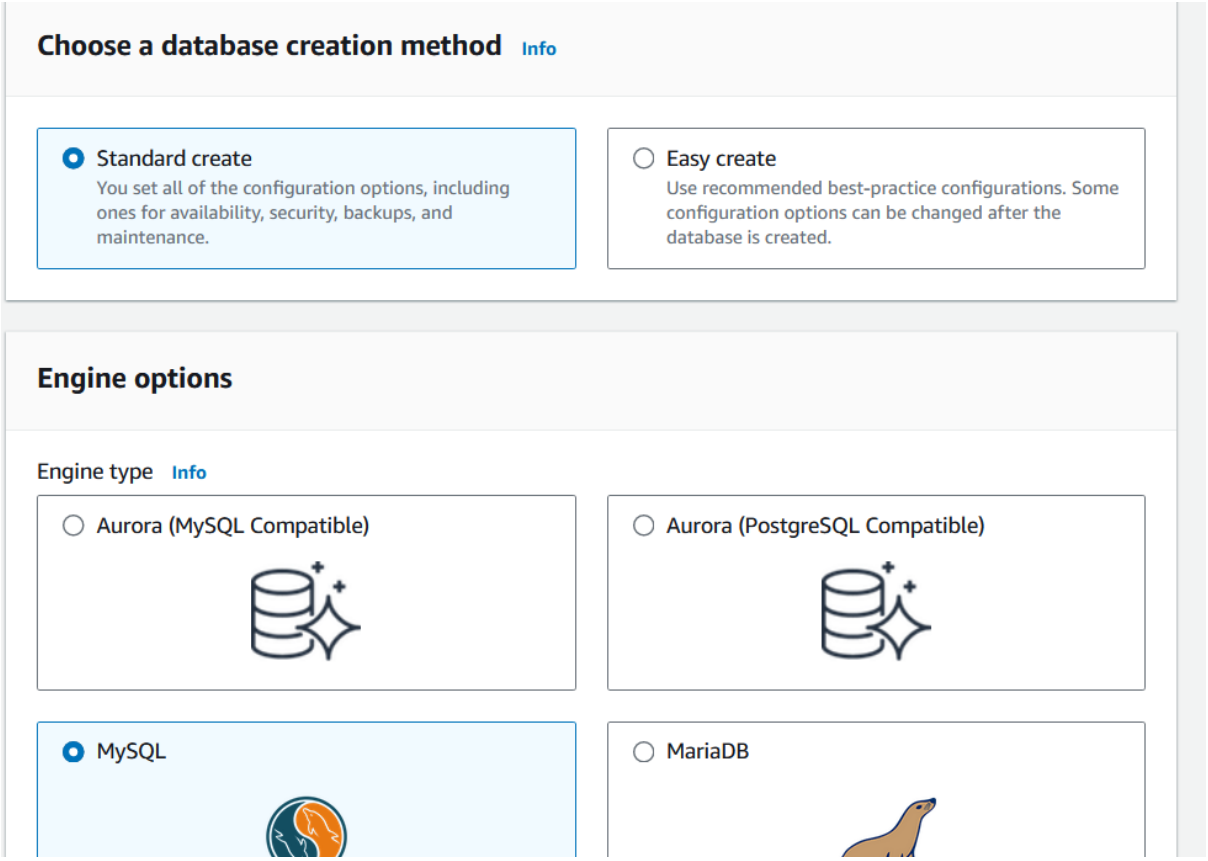






- Choose the "Standard Create" option and choose MySQL engine to create a new database instance

.



Engine Version

MySQL 8.0.35

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

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

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.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized

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

20 GiB

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

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.
Learn more

▼ 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.

Master username Info

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

admin

1 to 32 alphanumeric characters. The first character must be a letter.

☐ Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
Learn more

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

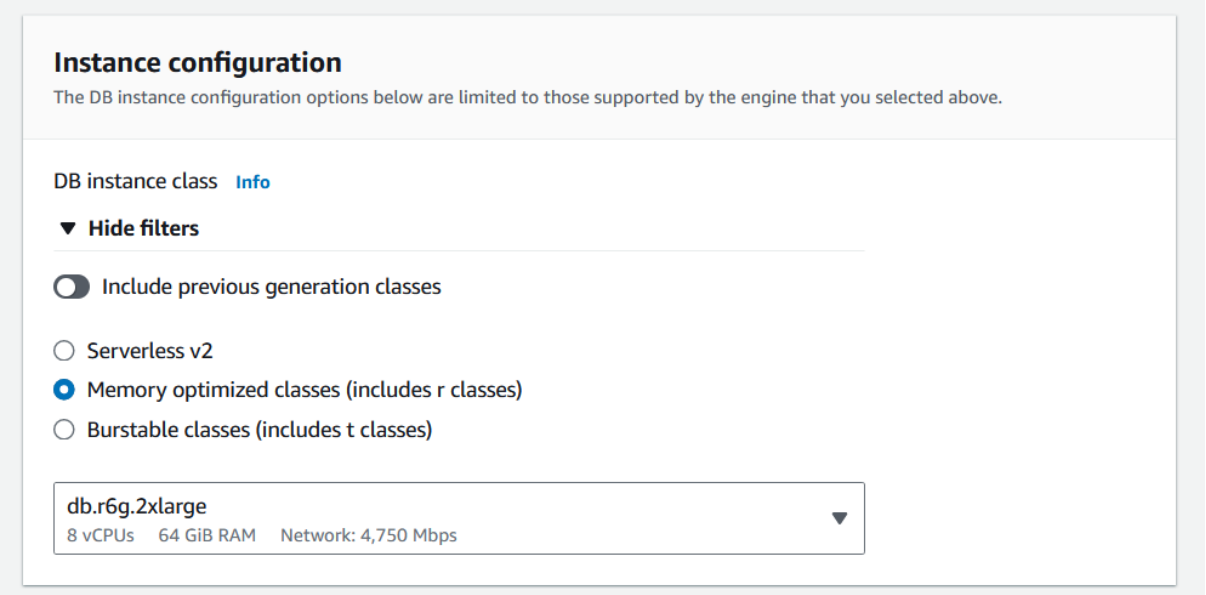
Master password Info

••••••••••

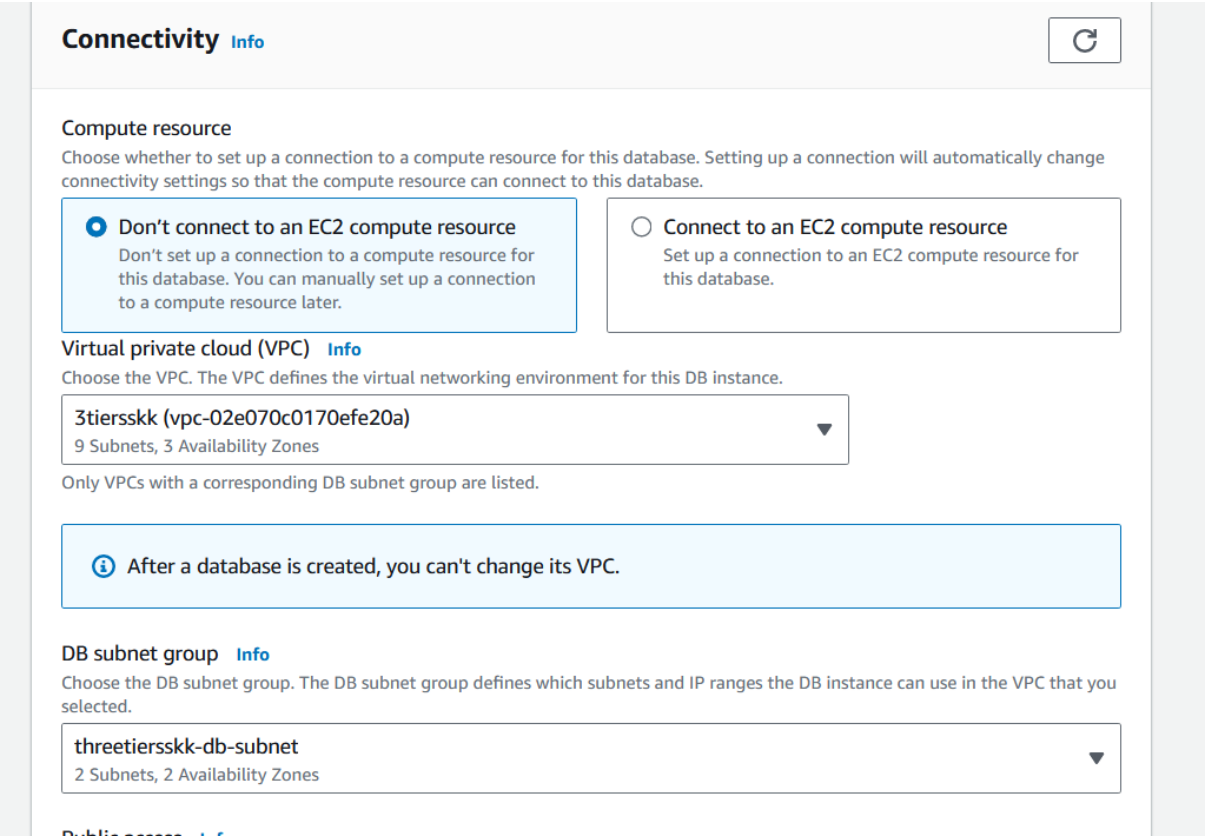
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

••••••••••



- Configure the VPC and subnet group to ensure the database is deployed within your desired network configuration.
- Specify the security group settings to control inbound and outbound traffic to the database instance. Allow inbound traffic from your application tier (e.g., EC2 instances) to access the database.



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

3tiersskk_DB_sg

Availability Zone Info

ap-southeast-1a

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.

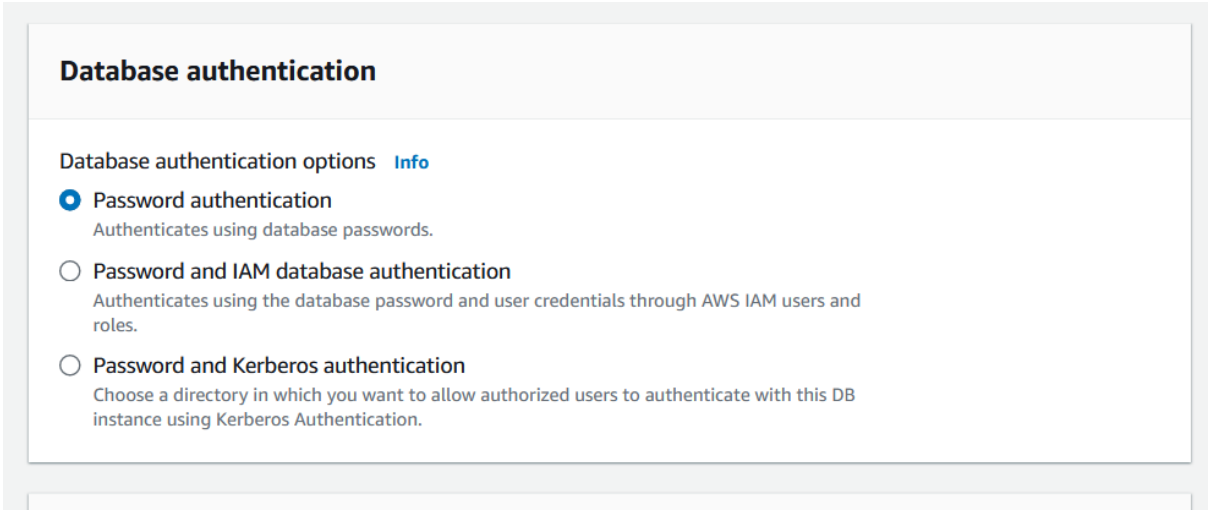
Certificate authority - *optional* Info

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiry: May 22, 2061

If you don't select a certificate authority, RDS chooses one for you.

▶ **Additional configuration**



- Use the package management tool (apt for Ubuntu) to install the MySQL server package.
- After MySQL server is installed, start the MySQL service
- MySQL installation comes with a default security script that you can run to secure your MySQL server installation

```
ubuntu@ip-172-20-4-157:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-l
  libhtml-tagset-perl libhtml-template-perl libhttp-date-pe
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils my
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isb
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-l
  libhtml-tagset-perl libhtml-template-perl libhttp-date-pe
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils my
0 upgraded, 25 newly installed, 0 to remove and 37 not upgr
Need to get 26.8 MB of archives.
After this operation, 181 MB of additional disk space will
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu j
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu j
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /li
Setting up libcgi-pm-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mysql-server (8.0.36-0ubuntu0.22.04.1) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-20-4-157:~$
```

```
ubuntu@ip-172-20-4-157:~$ sudo service mysql status
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:
     Active: active (running) since Mon 2024-03-11 05:07:05 UTC; 1min 33s ago
    Process: 1578 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=e
   Main PID: 1594 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 1121)
     Memory: 356.8M
        CPU: 1.405s
     CGroup: /system.slice/mysql.service
             └─1594 /usr/sbin/mysqld

Mar 11 05:07:04 ip-172-20-4-157 systemd[1]: Starting MySQL Community Server...
Mar 11 05:07:05 ip-172-20-4-157 systemd[1]: Started MySQL Community Server.
ubuntu@ip-172-20-4-157:~$
```

```
ubuntu@ip-172-20-4-157:~$ sudo ss -tap | grep mysql
LISTEN    0      151                127.0.0.1:mysql              0.0.0.0:*     users:(("mysqld",pid=1594,fd=23))
LISTEN    0      70                 127.0.0.1:33060              0.0.0.0:*     users:(("mysqld",pid=1594,fd=21))
ubuntu@ip-172-20-4-157:~$
```

- If you haven't already created a database, you can create one
- After creating the database, switch to it

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW TABLES;
ERROR 1046 (3D000): No database selected
mysql> CREATE DATABASE mydatabase;
Query OK, 1 row affected (0.00 sec)

mysql> use mydatabase;
Database changed
```

- Now, you can create tables within the database using the CREATE TABLE SQL command
- We are creating two tables. One with 5 columns and other with 3 columns.

```
mysql> CREATE TABLE Groceries (
    ->     item_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     item_name VARCHAR(255) NOT NULL,
    ->     quantity INT,
    ->     unit_price DECIMAL(10, 2),
    -> payment_mode VARCHAR(255) NOT NULL
    -> )
    -> ;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE Expenses (
    ->     expense_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     description VARCHAR(255),
    ->     amount DECIMAL(10, 2)
    -> )
    -> ;
Query OK, 0 rows affected (0.01 sec)
```

- After creating the table, you can view it by using the SHOW TABLES command

```
mysql> show tables
    -> ;
+----------------------+
| Tables_in_mydatabase |
+----------------------+
| Expenses             |
| Groceries            |
+----------------------+
2 rows in set (0.00 sec)

mysql>
```

- To view the structure of a specific table, you can use the DESCRIBE command

```
mysql> describe Expenses;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| expense_id  | int           | NO   | PRI | NULL    | auto_increment |
| description | varchar(255)  | YES  |     | NULL    |                |
| amount      | decimal(10,2) | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

- Use the INSERT INTO SQL command to insert data into a specific table and verify the data using SELECT command.

```
mysql> INSERT INTO Expenses (description, amount) VALUES
    -> ('Petrol', 250);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Expenses (description, amount) VALUES
    -> ('Shopping',2521);
Query OK, 1 row affected (0.00 sec)
```

```
mysql>  SELECT * FROM Expenses;
+------------+-------------+---------+
| expense_id | description | amount  |
+------------+-------------+---------+
|          1 | Petrol      |  250.00 |
|          2 | Shopping    | 2521.00 |
+------------+-------------+---------+
2 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Groceries(item_name,quantity,unit_price,payment_mode) VALUES ('Coffee',1,120,'Gpay'), ('Veggies',10,239,'Cash');
Query OK, 2 rows affected (0.00 sec)
```

```
mysql>  SELECT * FROM Groceries;
+---------+-----------+----------+------------+--------------+
| item_id | item_name | quantity | unit_price | payment_mode |
+---------+-----------+----------+------------+--------------+
|       1 | Coffee    |        1 |     120.00 | Gpay         |
|       2 | Veggies   |       10 |     239.00 | Cash         |
+---------+-----------+----------+------------+--------------+
2 rows in set (0.00 sec)
```

7. Best practices and Points to remember

7.1. Performance Standards:

- **Response Time**: Aim for an average response time of less than 500 milliseconds for web requests.
- **Throughput**: Serve a minimum of 1000 requests per second during peak traffic.
- **Database Performance**: Ensure that critical database operations execute within predefined time limits.

7.2. Reliability Standards:

- **Uptime**: Maintain a system uptime of at least 99.9% over a defined period.
- **MTBF (Mean Time Between Failures)**: Increase the MTBF to minimize system failures and downtime.
- **MTTR (Mean Time to Recover)**: Strive to reduce MTTR to quickly recover from incidents and minimize service disruption.

7.3. Scalability and Elasticity Standards:

- **Auto Scaling Responsiveness**: Ensure that the Autoscaling Group responds promptly to changes in load, scaling instances as needed.
- **Elasticity Ratio**: Achieve an elasticity ratio that efficiently matches capacity to demand fluctuations.

7.4. Security and Compliance Standards:

- **Zero Security Incidents**: Aim for zero security incidents or breaches by implementing robust security measures.
- **Compliance Adherence**: Ensure compliance with relevant regulations and internal security policies.

7.5. Cost Optimization Standards:

- **Cost Efficiency**: Optimize infrastructure costs while maintaining performance and reliability, achieving a predefined cost reduction target.
- **Resource Utilization**: Ensure efficient resource utilization to avoid over-provisioning and unnecessary expenses.

7.6. User Experience Standards:

- **High User Satisfaction**: Maintain high user satisfaction scores through regular feedback and improvements.
- **Low Error Rates**: Strive for a low error rate in the application to enhance user experience and reliability.
- **Improvement in Conversion Rates**: Measure and improve conversion rates to demonstrate positive user engagement.

7.7. Monitoring and Incident Response Standards:

- **Prompt Alert Response**: Acknowledge and respond to alerts promptly, ensuring swift incident resolution.
- **Efficient Incident Resolution**: Resolve incidents quickly to minimize service disruption and impact on users.
- **Minimize False Positives**: Reduce false positive alerts to improve incident response efficiency.

7.8. Continuous Improvement Standards:

- **Iterative Development**: Embrace iterative development practices to continuously deliver new features and enhancements.
- **Feedback Loop Integration**: Establish a feedback loop to gather user and stakeholder feedback for continuous improvement.

- **Technical Debt Reduction**: Continuously address and reduce technical debt to maintain system agility and sustainability.

8. Additional elements and features followed as per current trend

To enhance the quality and value of the project, we can incorporate additional elements or features that improve various aspects such as performance, scalability, security, user experience, and maintainability. Here are some suggestions:

8.1. Performance Optimization:

- Implement caching mechanisms (e.g., Redis, Memcached) to reduce latency and improve response times.
- Use Content Delivery Networks (CDNs) to deliver static content closer to users and reduce load on origin servers.
- Optimize database queries and indexes to improve database performance.

8.2. Scalability Enhancements:

- Implement asynchronous processing using message queues (e.g., Amazon SQS, RabbitMQ) to decouple components and handle spikes in traffic.
- Explore serverless computing (e.g., AWS Lambda, Azure Functions) for handling intermittent or bursty workloads.
- Horizontal scaling of components using containerization (e.g., Docker) and orchestration tools (e.g., Kubernetes) for better resource utilization.

8.3. Security Features:

- Implement multi-factor authentication (MFA) for user accounts to enhance security.
- Integrate intrusion detection and prevention systems (IDS/IPS) to monitor and protect against security threats.
- Conduct regular security audits and penetration testing to identify and mitigate vulnerabilities.

8.4. User Experience Enhancements:

- Implement responsive web design to ensure optimal viewing experience across different devices and screen sizes.
- Enhance accessibility features to make the application usable by a wider range of users, including those with disabilities.
- Implement personalization features based on user preferences and behavior to improve engagement.

8.5. Monitoring and Analytics:

- Implement comprehensive monitoring and logging solutions (e.g., ELK stack, Splunk) to track application performance and diagnose issues.
- Use A/B testing and user analytics tools to gather insights into user behavior and preferences for data-driven decision making.
- Implement anomaly detection algorithms to automatically detect and alert on unusual behavior or performance degradation.

8.6. Documentation and Knowledge Sharing:

- Create thorough documentation covering architecture, deployment procedures, troubleshooting guides, and best practices.
- Establish a knowledge sharing platform or internal wiki to facilitate collaboration and knowledge transfer among team members.

- Conduct regular code reviews and provide feedback to improve code quality and maintainability.

8.7. Integration with Third-party Services:

- Integrate with external services (e.g., payment gateways, social media platforms) to extend functionality and provide additional value to users.
- Implement APIs for third-party developers to extend the platform and encourage ecosystem growth.
- Explore partnerships with complementary services or platforms to enhance the overall offering.

8.8. Continuous Improvement Processes:

- Implement continuous integration and continuous deployment (CI/CD) pipelines to automate software delivery and deployment processes.
- Establish regular retrospective meetings to reflect on past iterations and identify areas for improvement.
- Foster a culture of innovation and experimentation, encouraging team members to propose and explore new ideas and technologies.

Incorporating these additional elements or features can significantly enhance the quality, value, and overall success of the project while providing learners with valuable experience in building robust and scalable software systems.