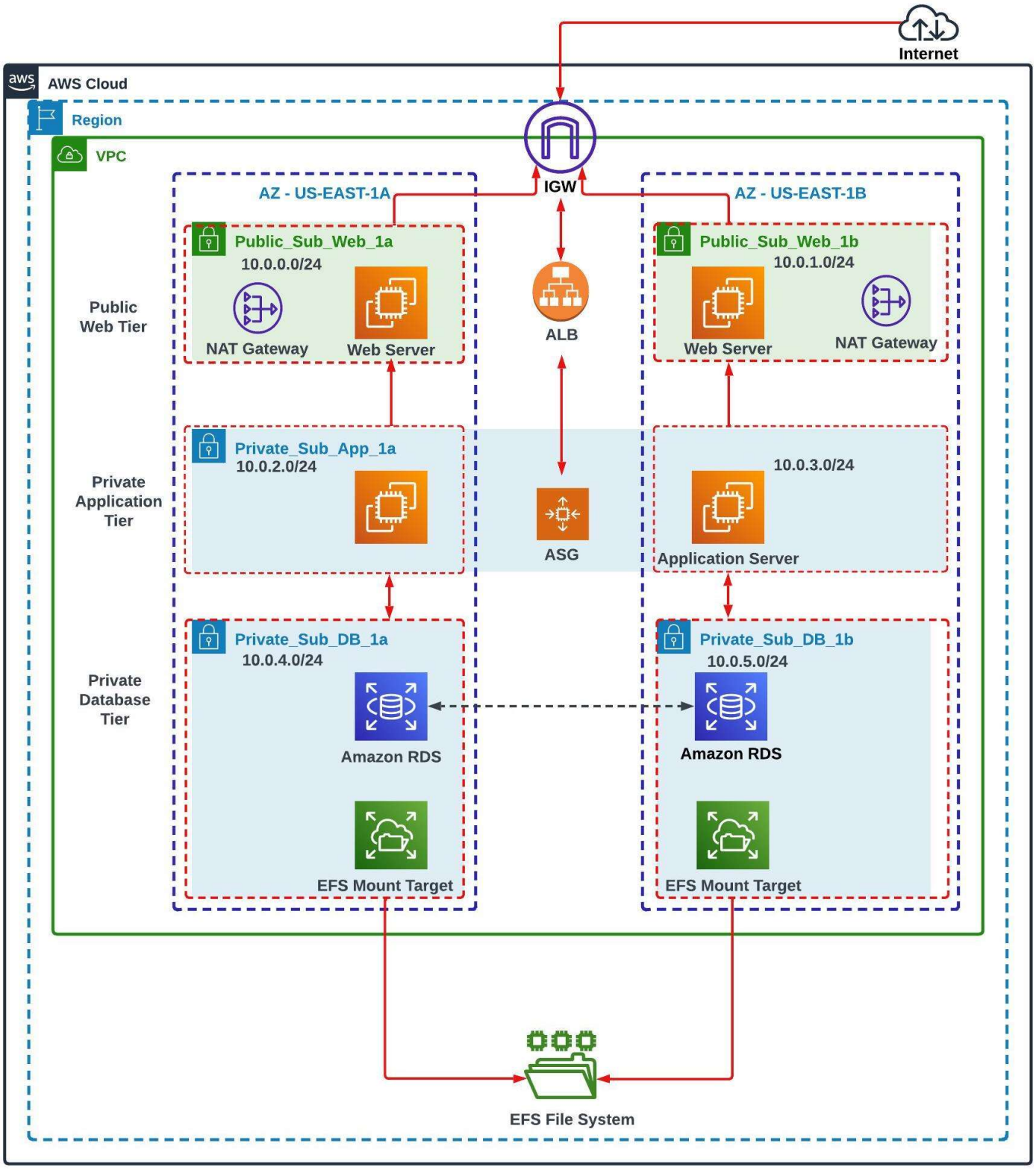
DEPLOYMENT OF A WORDPRESS WEBSITE ON AWS



Deploying a WordPress website on AWS involves several steps setting up the necessary infrastructure and configuring the environment for the various AWS services to host and run the WordPress application.

The specific steps and tools used in the configuration may vary depending on the hosting option chosen and your specific requirements. Detailed documentation and tutorials are available from AWS and the WordPress community to assist you in each stage of the deployment process.

With AWS as our launchpad, let’s navigate the vast AWS cosmos together to successfully deploy our WordPress website and have it running seamlessly.

REQUIRED AWS SERVICES

1. VPC WITH PRIVATE AND PUBLIC SUBNETS 10.0.0.0/16
2. EC2
3. SECURITY GROUPS
4. RDS
5. NAT GATEWAY
6. ELASTIC FILE SYSTEM (EFS)
7. APPLICATION LOAD BALANCER (ALB)
8. AUTO SCALING GROUP (ASG)
9. AMAZON CEERTIFICATE MANAGER
10. ROUTE 53
11. CLOUDFRONT

However, here is a brief introduction to the process step by step:

STEP 1:

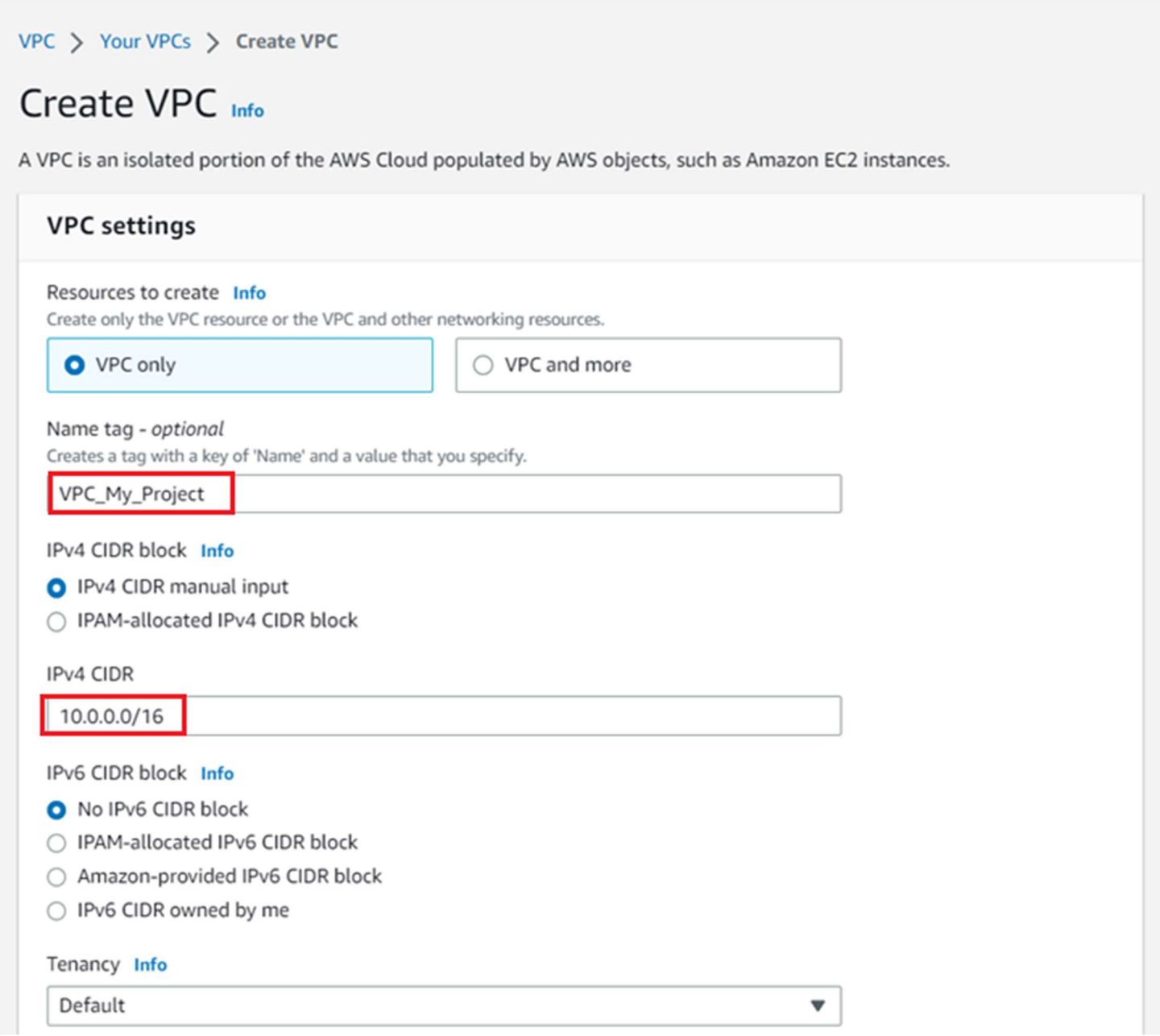
CREATE AN AWS ACCOUNT: If you don't have an AWS account, you'll need to sign up for one at aws.amazon.com. This will provide you with access to the AWS Management Console.

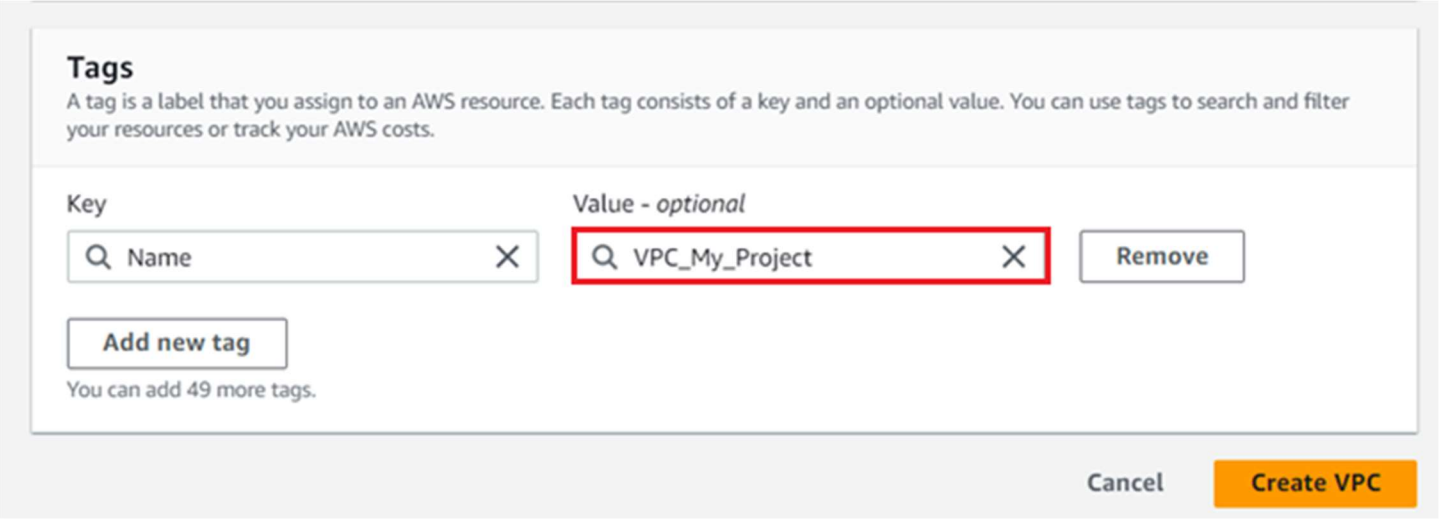
STEP 2:

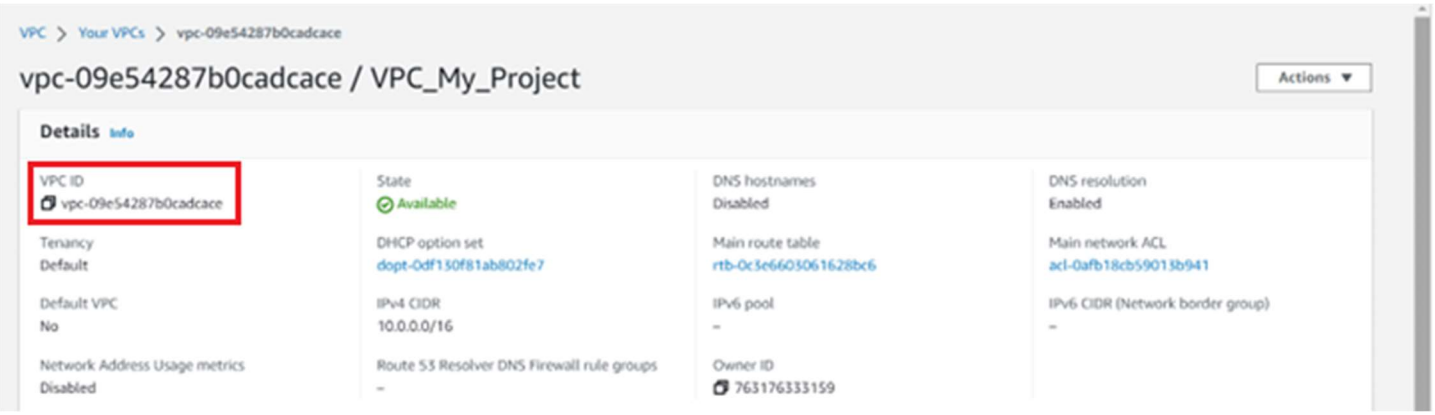
WE HAVE TO CREATE A VIRTUAL PRIVATE CLOUD (VPC)

First, let's create a VPC.

We also must specify the range of IPv4 CIDR block address for the VPC, and my specified CIDR range will be **10/0.0.0/16.**







My VPC for this project **(VPC_My_Project)** has been successfully created.

STEP 3:

NOW, WE HAVE TO CREATE AN INTERNET GATEWAY (IGW)

Let's create an Internet Gateway **(IGW)**, after that, we must select my VPC and then attach the internet gateway to the VPC.

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.

IGW_My_Project

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	IGW_My_Project	Remove

Add new tag

You can add 49 more tags.

Cancel | Create internet gateway

Attach to VPC (igw-0e1e2f6ca9c065b82) Info

VPC

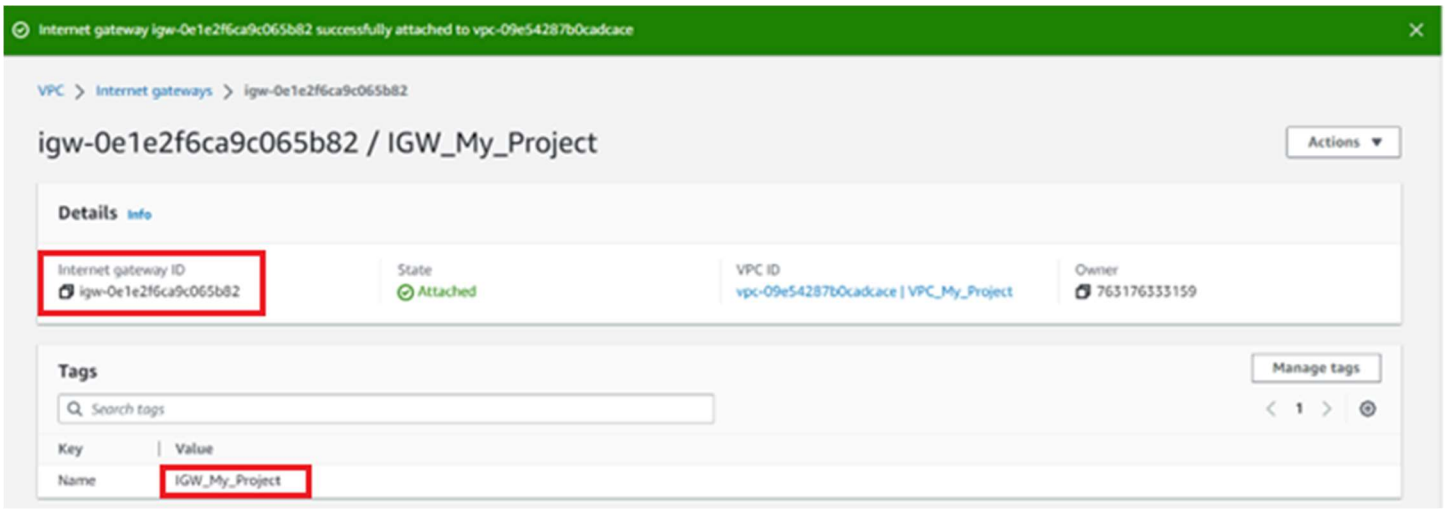
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

vpc-09e54287b0cadcace

▶ AWS Command Line Interface command

Cancel | Attach internet gateway



The Internet gateway **(IGW_My_Project)** was successfully created and attached to our VPC.

STEP 4:

NEXT WE HAVE TO CONFIGURE TWO (2) NAT GATEWAYS

Let’s create two (2) NAT Gateways for redundancy.

The main function of the NAT Gateway is to allow our EC2 instances in the private subnets to have access to the internet for software updates, package installation, downloads etc.

Navigate to VPC, scroll down to select NAT gateways, and click on create NAT gateway.

Let’s specify the public subnets where we want the NAT gateways to reside when created. Let’s select our first public subnet **"Public_Sub_Web_1a"** for this configuration, and then select connectivity type as public.

We must click on "Allocate Elastic IP" to allocate an Elastic IP address to the NAT gateway.

Click on create NAT gateway and repeat same process to create the second NAT gateway in the second public subnet **"Public_Sub_Web_1b".**

VPC > NAT gateways > Create NAT gateway

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

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

NATGW1_My_Project

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

subnet-0552f97133c2ed30b (Public_Sub_Web_1a)

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-0094a388a534106c7

Allocate Elastic IP

▶ Additional settings Info

Tags

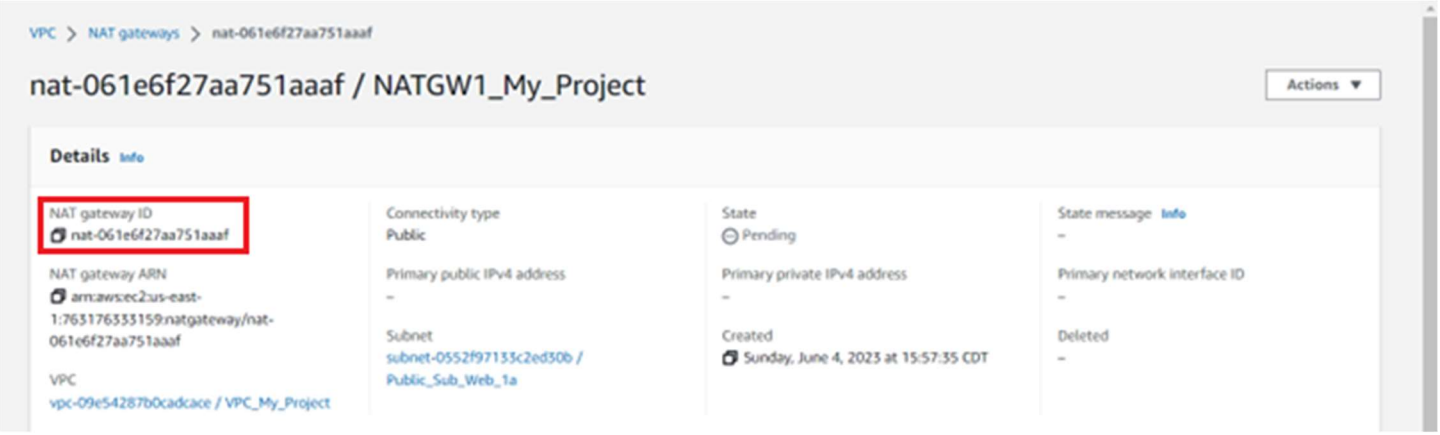
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	NATGW1_My_Project	Remove

Add new tag

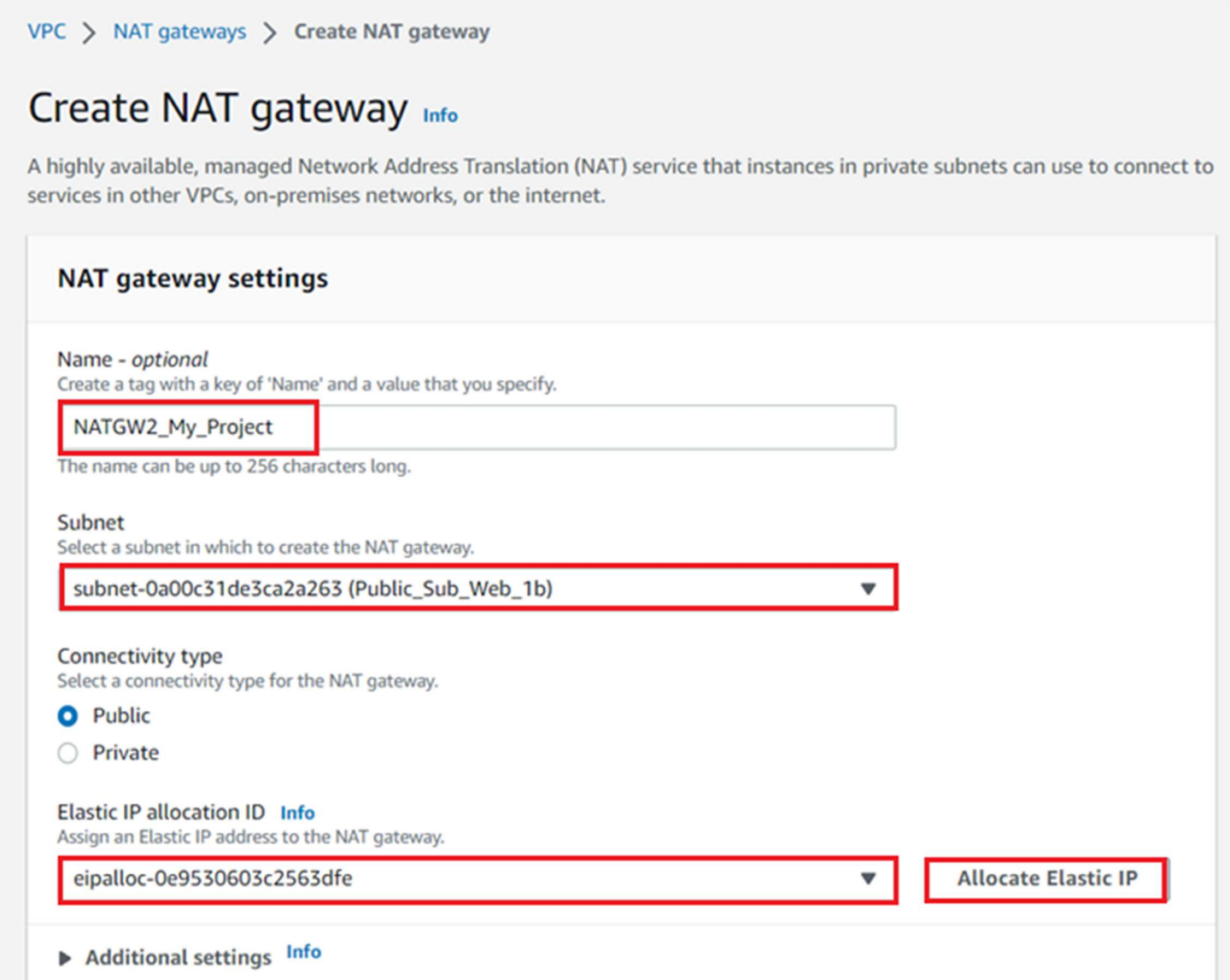
You can add 49 more tags.

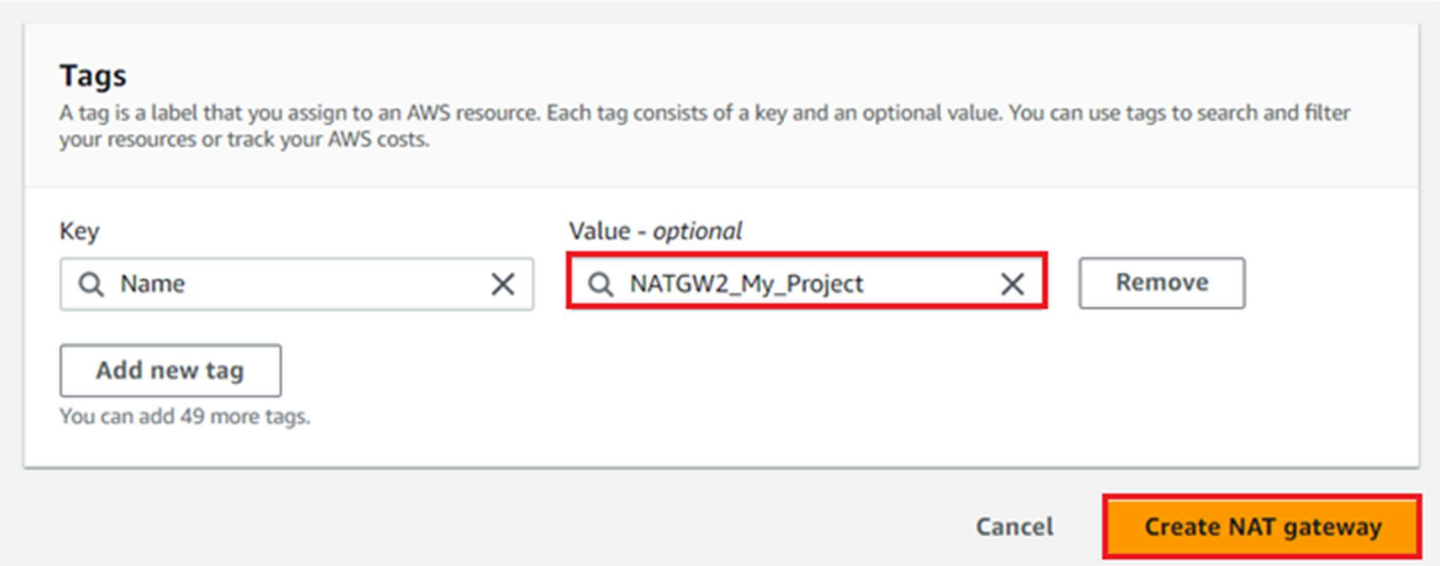
Cancel | Create NAT gateway

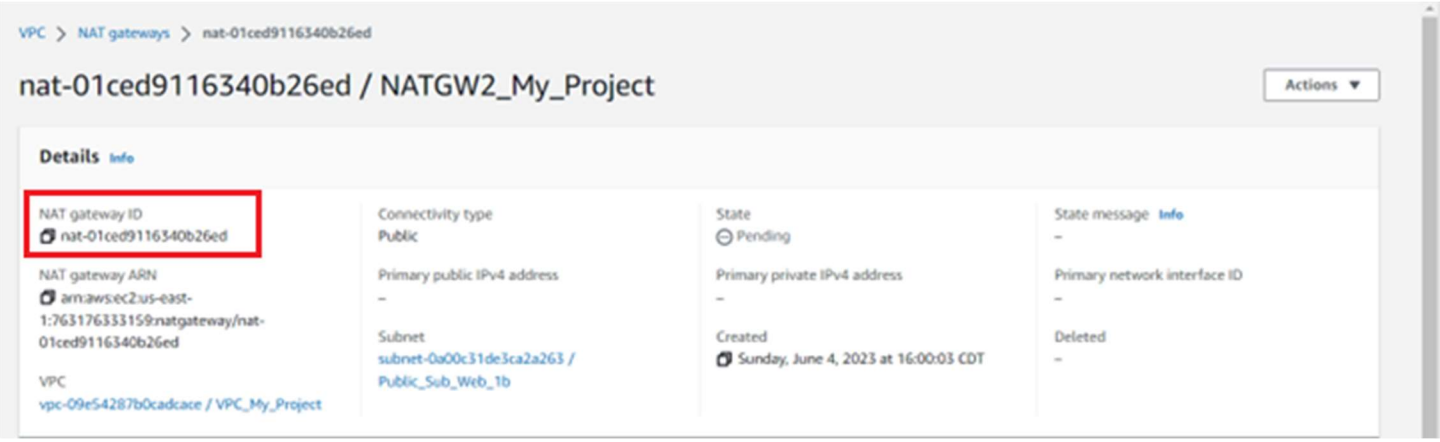


The first NAT Gateway **(NATGW1_My_Project)** was successfully created.

Now let’s create the second NAT Gateway.







The second NAT Gateway **(NAT2_My_Project)** has been successfully created.

STEP 5:

NEXT WE NEED TO CREATE TWO PUBLIC SUBNETS

Next, we need to create 2 Subnets, we can do this by specifying a subnet name, the availability zone and the IPv4 CIDR block for each subnet.

The CIDR blocks assigned to the subnets must be derived from my main VPC CIDR block **10.0.0.0/16.**

Create subnet Info

VPC

VPC ID

Create subnets in this VPC.

vpc-09e54287b0cadcace (VPC_My_Project)

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 2

Subnet name

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

Public_Sub_Web_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 CIDR block Info

10.0.0.0/24

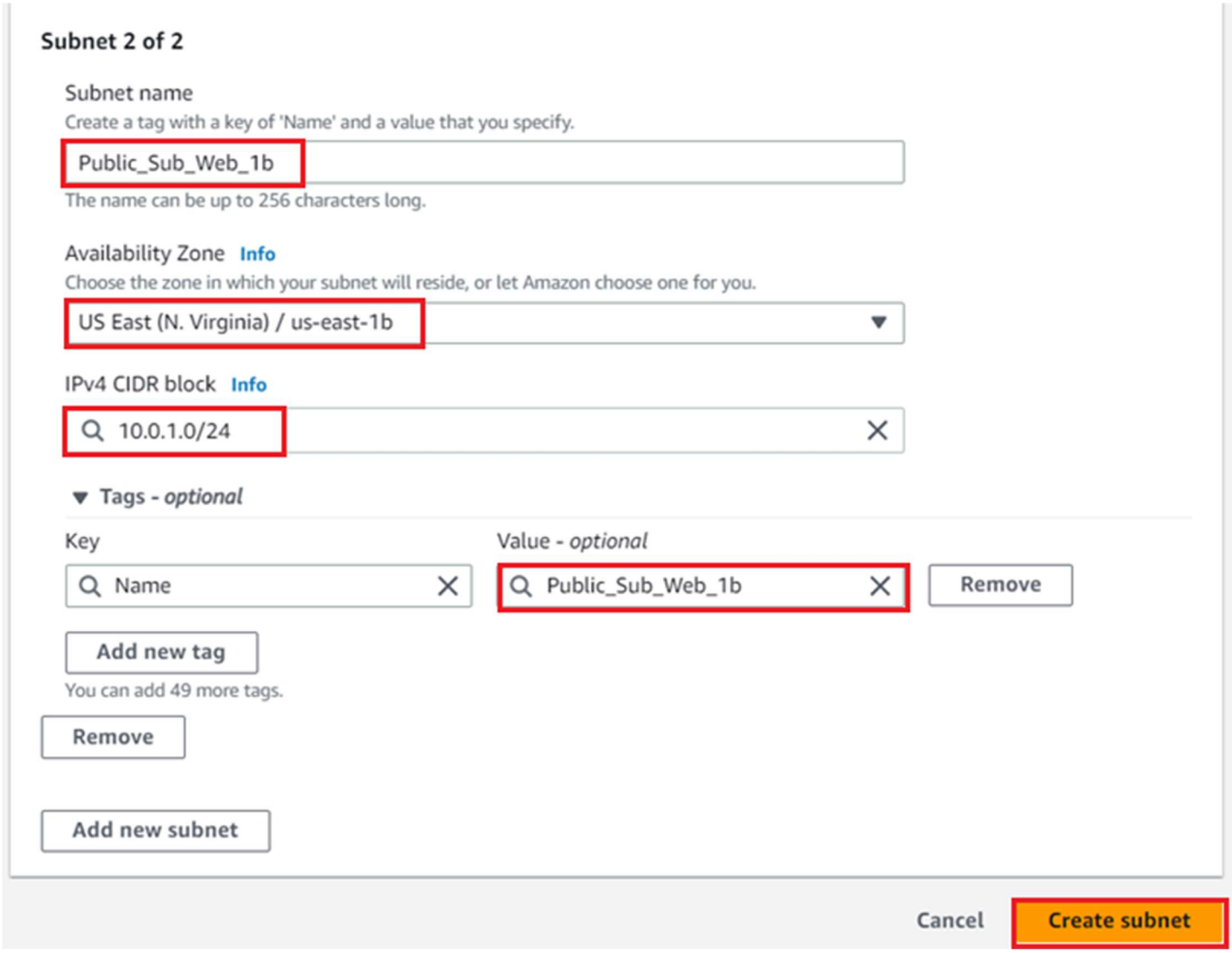
▼ Tags - *optional*

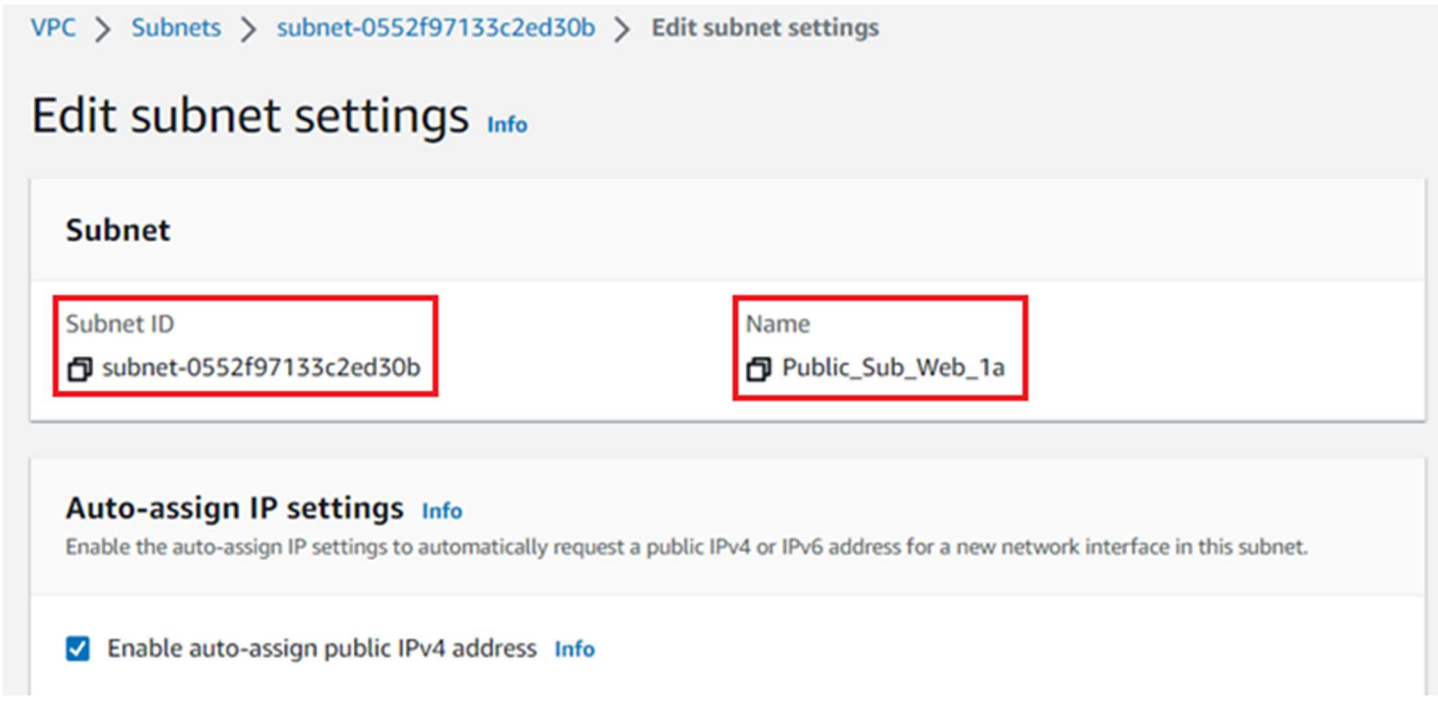
Key	Value - *optional*	
Name	Public_Sub_Web_1a	Remove

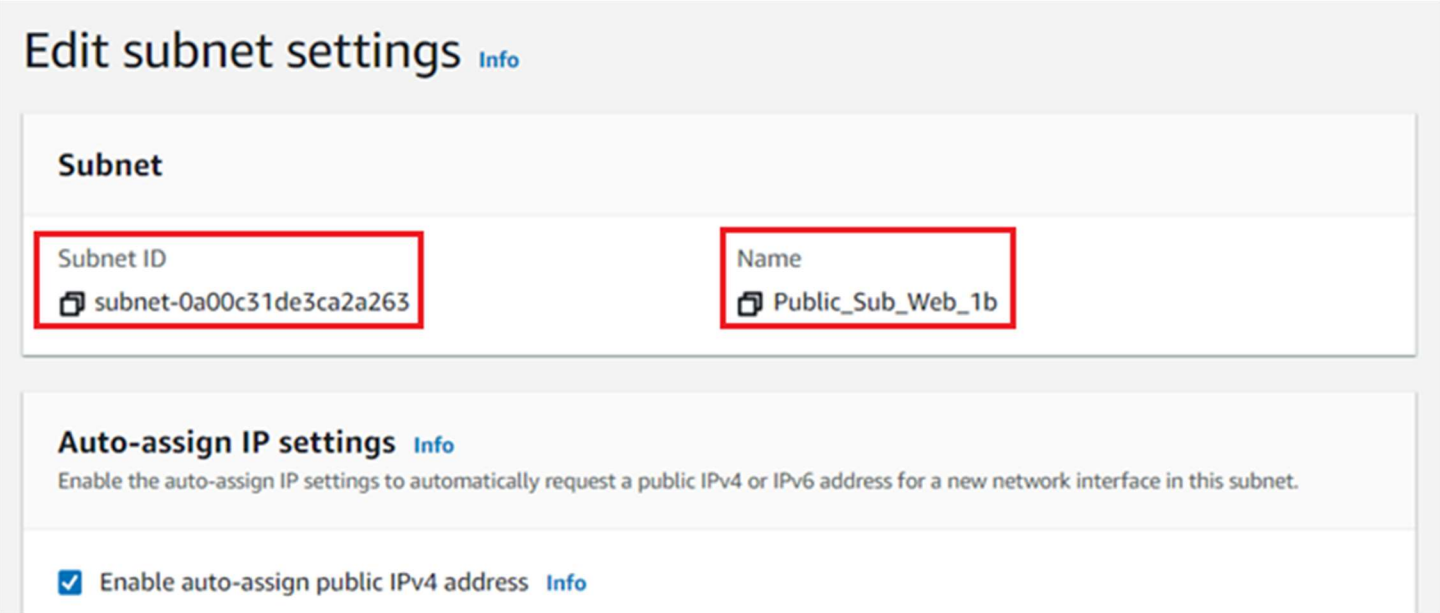
Add new tag

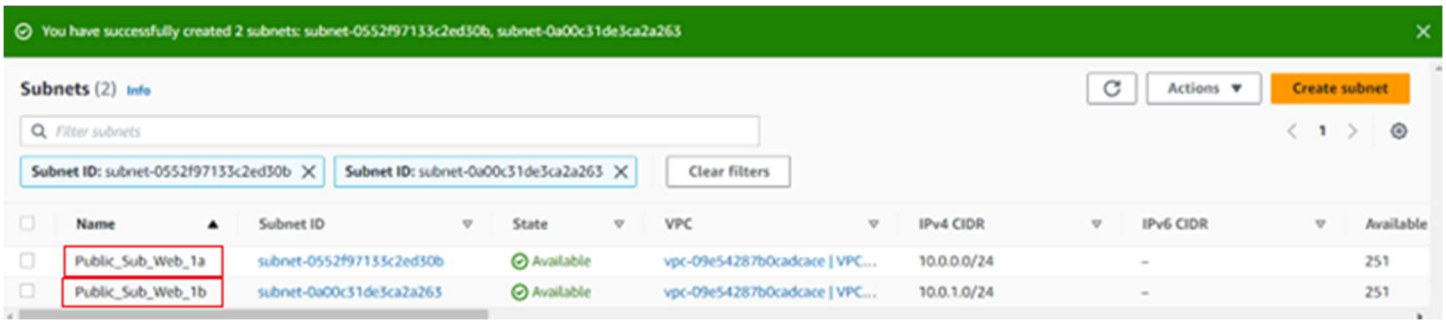
You can add 49 more tags.

Remove


Subnet 2 of 2
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Public_Sub_Web_1b
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-1b
IPv4 CIDR block Info
10.0.1.0/24
Tags - optional
Key
Value - optional
Name
Public_Sub_Web_1b
Remove
Add new tag
You can add 49 more tags.
Remove
Add new subnet
Cancel
Create subnet


VPC > Subnets > subnet-0552f97133c2ed30b > Edit subnet settings
Edit subnet settings Info
Subnet
Subnet ID
subnet-0552f97133c2ed30b
Name
Public_Sub_Web_1a
Auto-assign IP settings Info
Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.
Enable auto-assign public IPv4 address Info





The first two public subnets have been successfully created as we can see from the above screenshots.

It is important to note that the public subnets **(Public_Sub_Web_1a)** and **(Public_Sub_Web_1b)** will be used in the Web Tier.

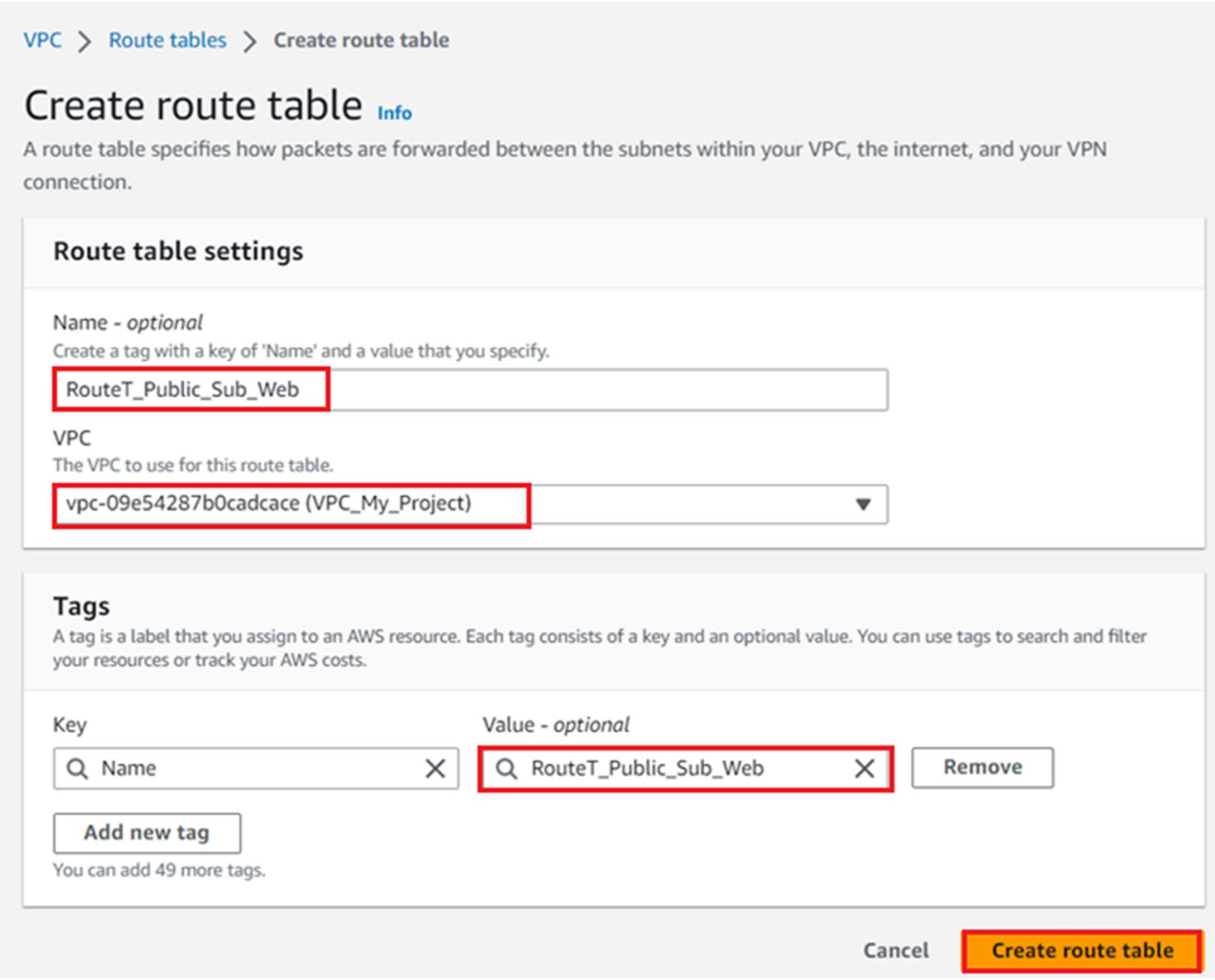
Now let's modify auto-assign IP settings of the newly created public subnets so that when we create EC2 instances, public IPV4 addresses will automatically be assigned to the EC2 instances.

The auto assign IPV4 address has been successfully modified and saved.

STEP 6:

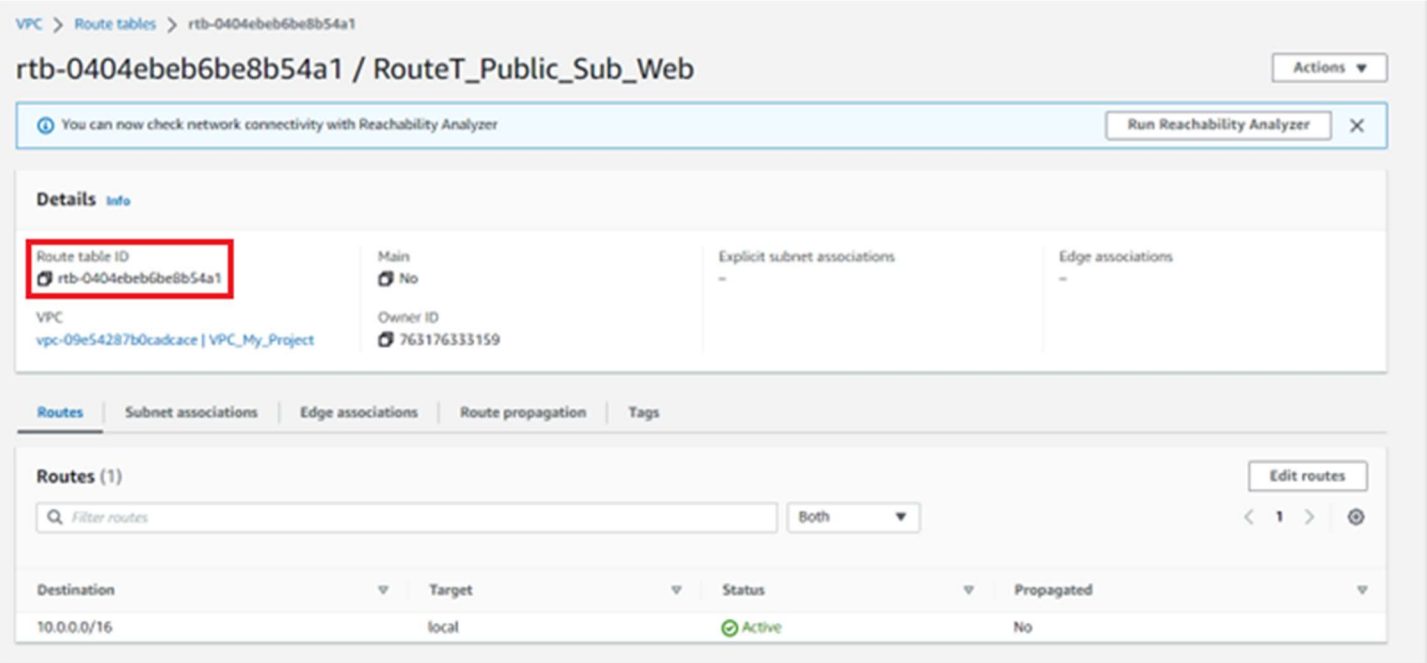
CREATION OF PUBLIC ROUTE TABLES

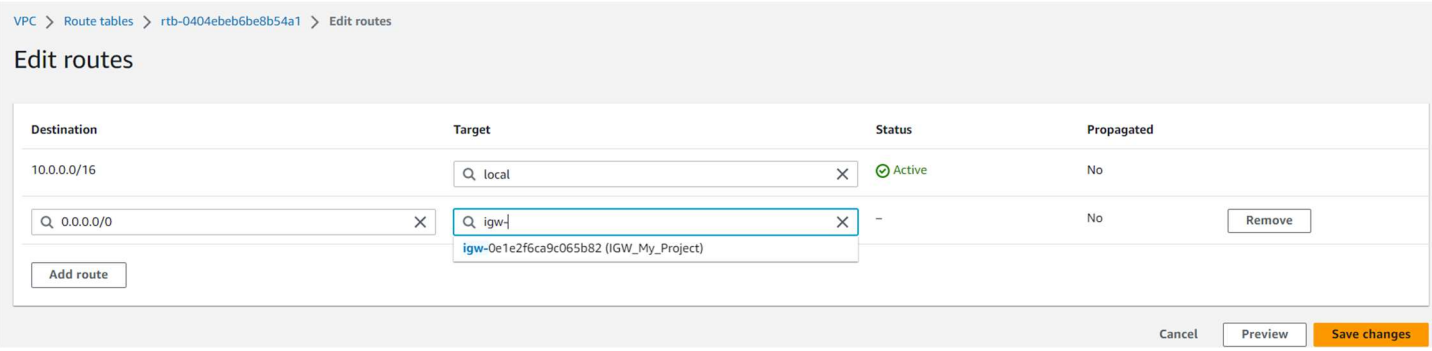
(a) Create Public Route Table **(RouteT_Public_Sub_Web)**



(b) Edit the route table **(RouteT_Public_Sub_Web)** to route traffic to Internet Gateway **(IGW_My_Project)**

For this configuration, simply click on the Route table **(RouteT_Public_Sub_Web),** select edit routes, select add route, in the blank field, type in **0.0.0.0/0** and then scroll down to select Internet gateway **(IGW_My_Project)** as target, then click "Save changes".

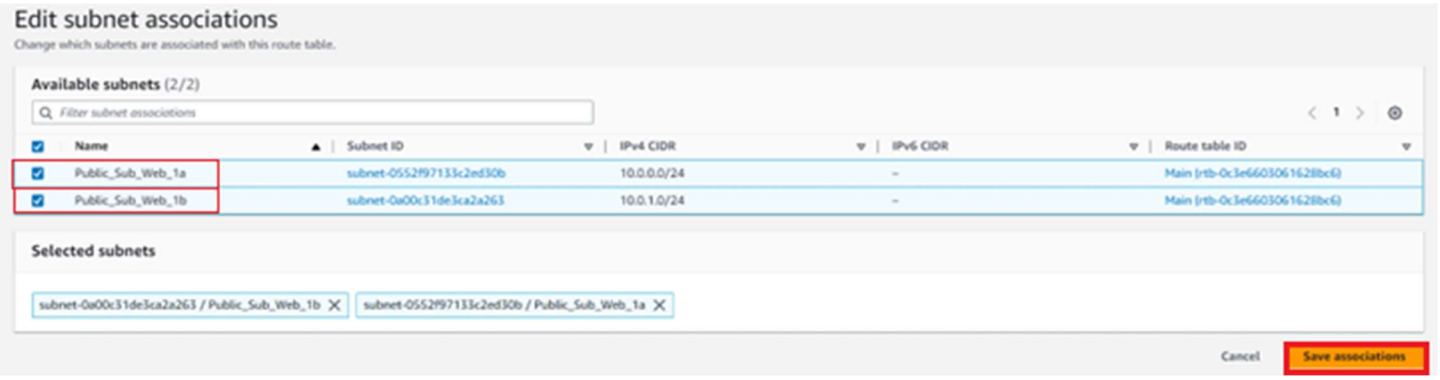


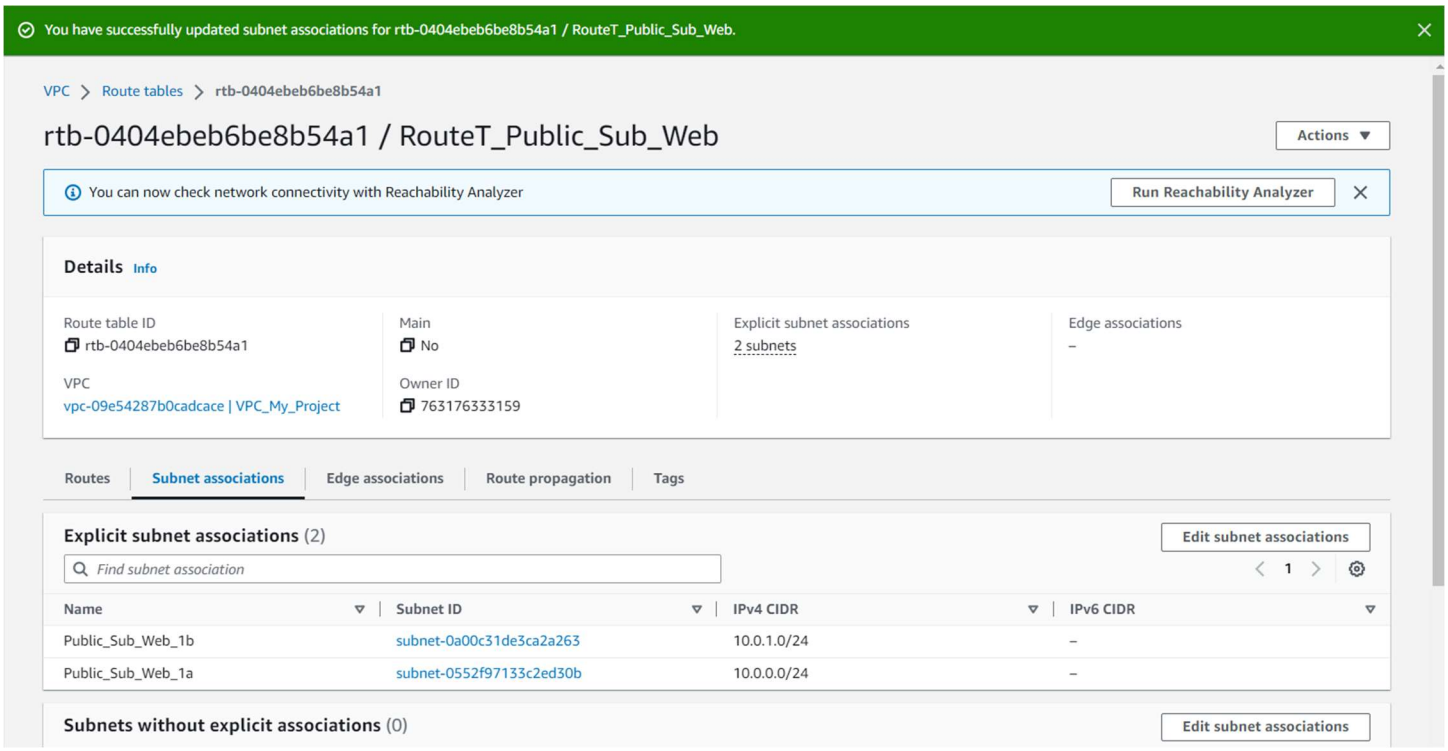


(c) Edit subnet association

To edit the subnet association of the route table (**RouteT_Public_Sub_Web)** associated with the two public subnets **(Public_Sub_Web_1a)** and **(Public_Sub_Web_1b),** simply click on "Subnet associations", proceed to click on "Edit subnet associations", select the public subnets **(Public_Sub_Web_1a)** and **(Public_Sub_Web_1b),** then click on "save associations"

NOTE: The Public Subnets **(Public_Sub_Web_1a)** and **(Public_Sub_Web_1b)** that are linked with my route table **(RouteT_Public_Sub_Web)** will now have access to the internet.





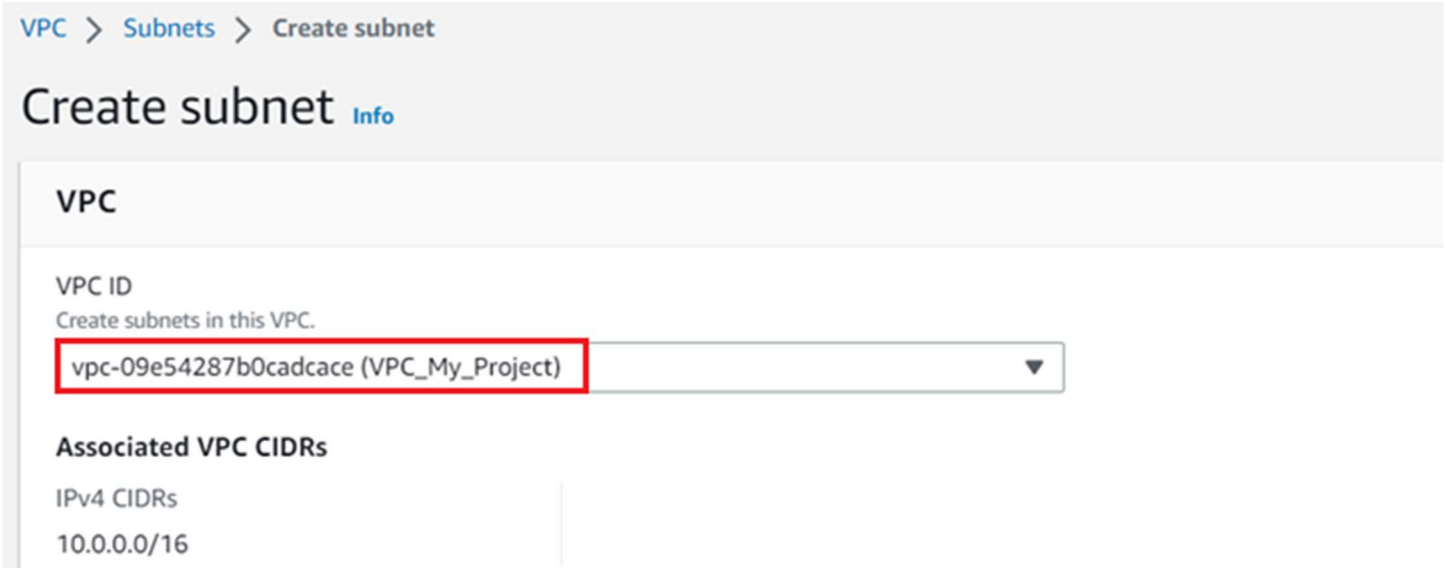
Subnets were successfully associated with route table.

STEP 7:

NEXT WE NEED TO CREATE FOUR (4) PRIVATE SUBNETS

Next, we need to create 4 private Subnets, we can do this by specifying a subnet name, the availability zone and the IPv4 CIDR block for each subnet.

The CIDR blocks assigned to the subnets must be derived from my main VPC CIDR block **10.0.0.0/16.**



Subnet 1 of 4

Subnet name

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

Private_Sub_App_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 CIDR block **Info**

10.0.2.0/24

▼ Tags - *optional*

Key	Value - *optional*	
Name	Private_Sub_App_1a	Remove

Add new tag

You can add 49 more tags.

Remove

Subnet 2 of 4

Subnet name

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

Private_Sub_App_1b

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-1b

IPv4 CIDR block **Info**

10.0.3.0/24

▼ Tags - *optional*

Key	Value - *optional*	
Name	Private_Sub_App_1b	Remove

Add new tag

You can add 49 more tags.

Remove

Subnet 3 of 4

Subnet name

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

Private_Sub_DB_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 CIDR block Info

10.0.4.0/24

▼ Tags - *optional*

Key	Value - *optional*	
Name	Private_Sub_DB_1a	Remove

Add new tag

You can add 49 more tags.

Remove

Subnet 4 of 4

Subnet name

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

Private_Sub_DB_1b

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-1b

IPv4 CIDR block Info

10.0.5.0/24

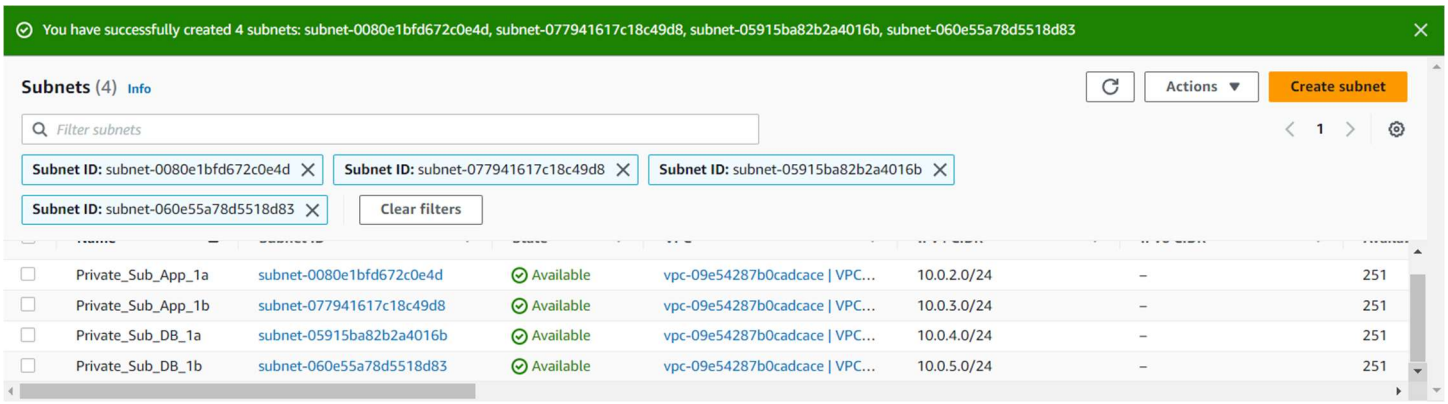
▼ Tags - *optional*

Key	Value - *optional*	
Name	Private_Sub_DB_1b	Remove

Add new tag

You can add 49 more tags.

Remove

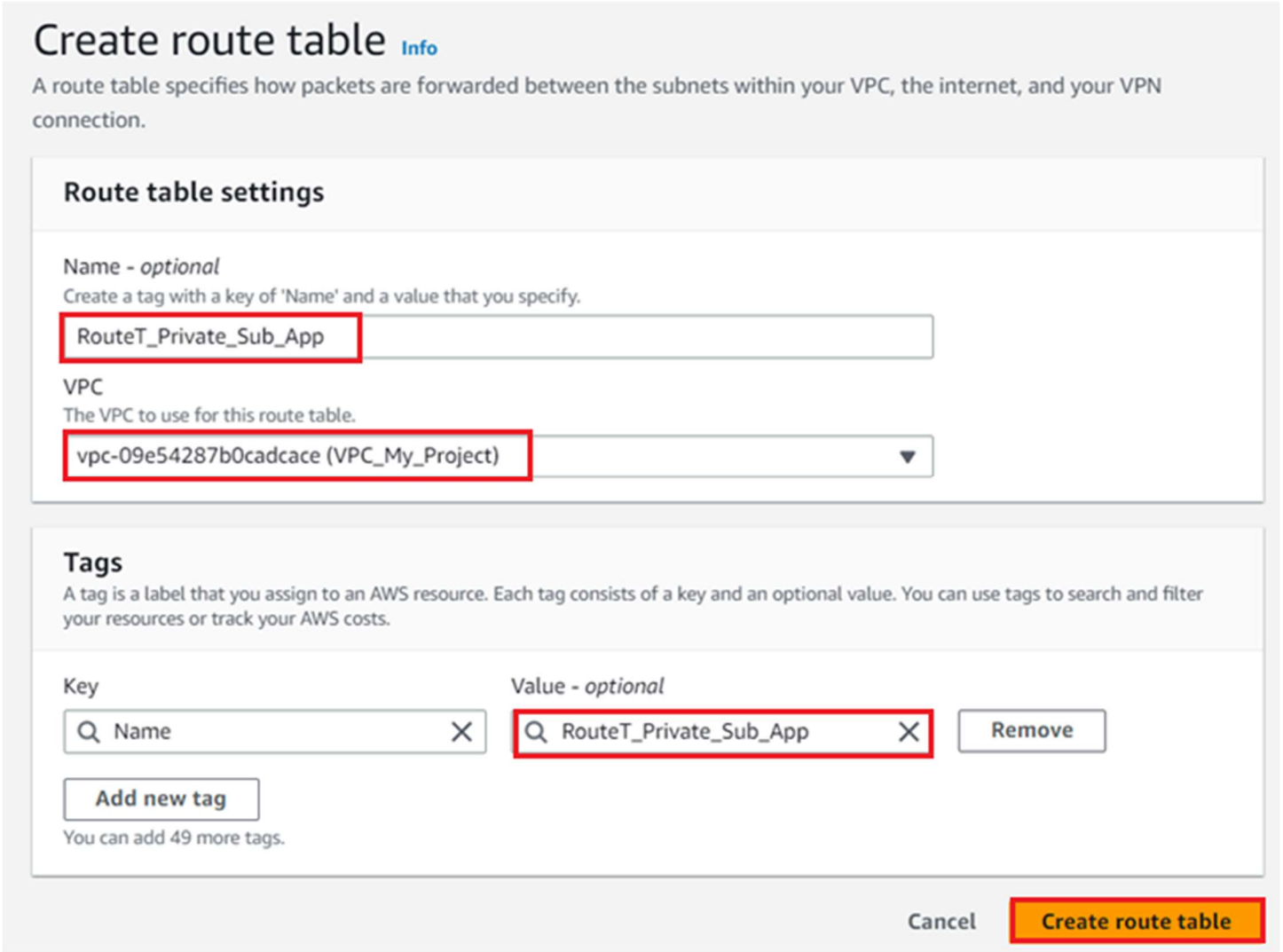


All 4 private subnets have been successfully created.

STEP 8:

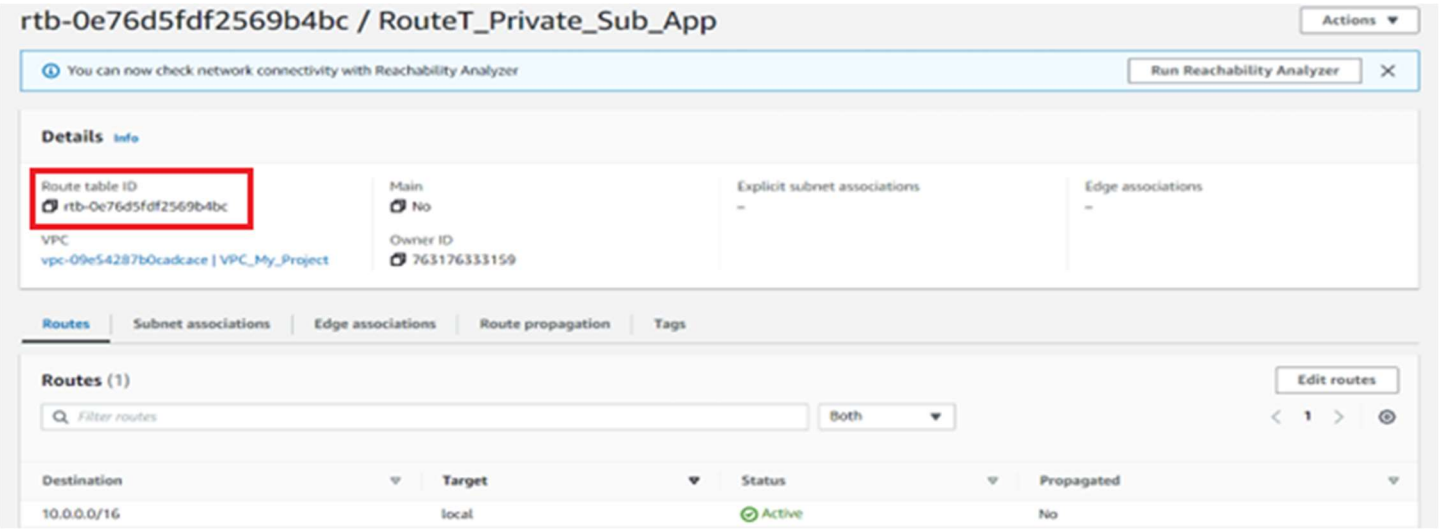
CREATION OF PRIVATE ROUTE TABLES THE APP TIER SUBNETS

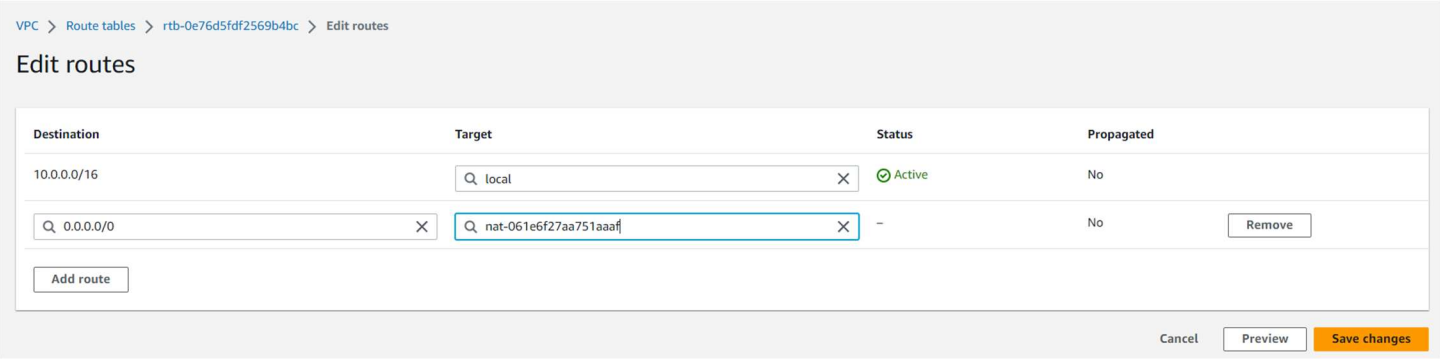
(a) Create the private Route Table **(RouteT_Private_Sub_App)**



(b) Edit the route table **(RouteT_Private_Sub_App)** to route traffic to the first NAT gateway **(NATGW1_My_Project**) that resides in the public subnet **(Public_Sub_Web_1a).**

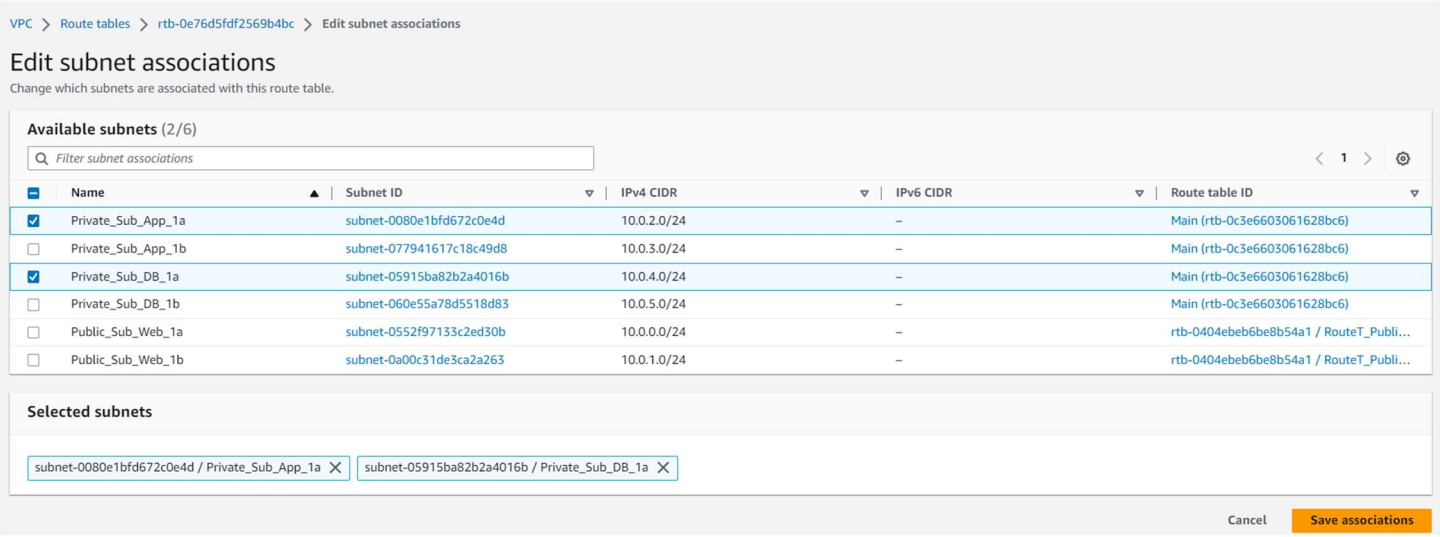
For this configuration, simply click on the Route table **(RouteT_Private_Sub_App),** select edit routes, select add route, in the blank field, type in **0.0.0.0/0** and then scroll down to select NAT gateway **(NATGW1_My_Project)** as target, then click save changes.





(c) Edit subnet association

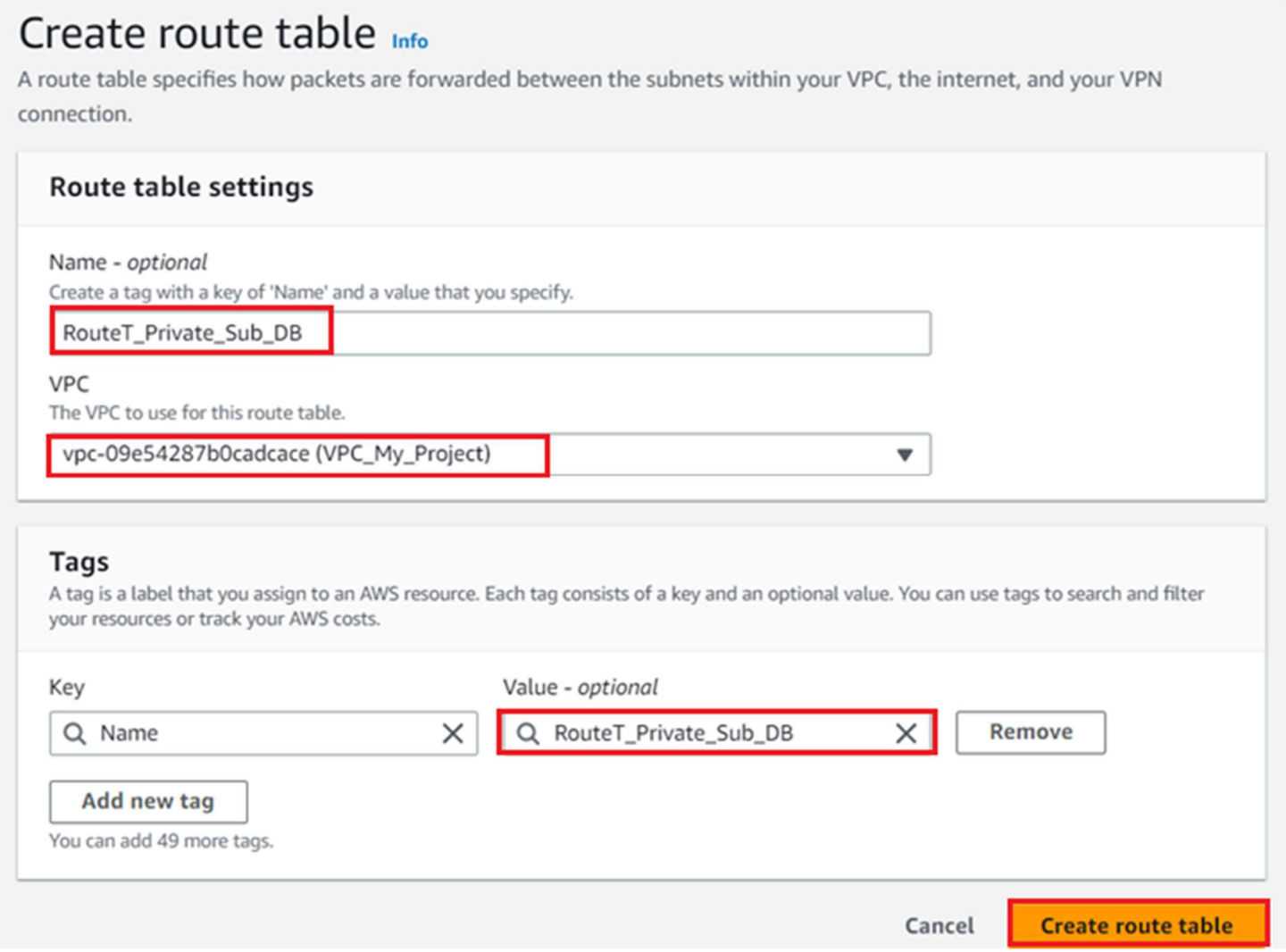
To edit the subnet association of the route table **(RouteT_Private_Sub_App)** associated to the 2 private subnets **(Private_Sub_App_1a)** and **(Private_Sub_DB_1a),** simply click on "Subnet associations", proceed to click on **Edit subnet associations, select** the private subnets **(Private_Sub_App_1a)** and **(Private_Sub_DB_1a)** then click on **save associations.**



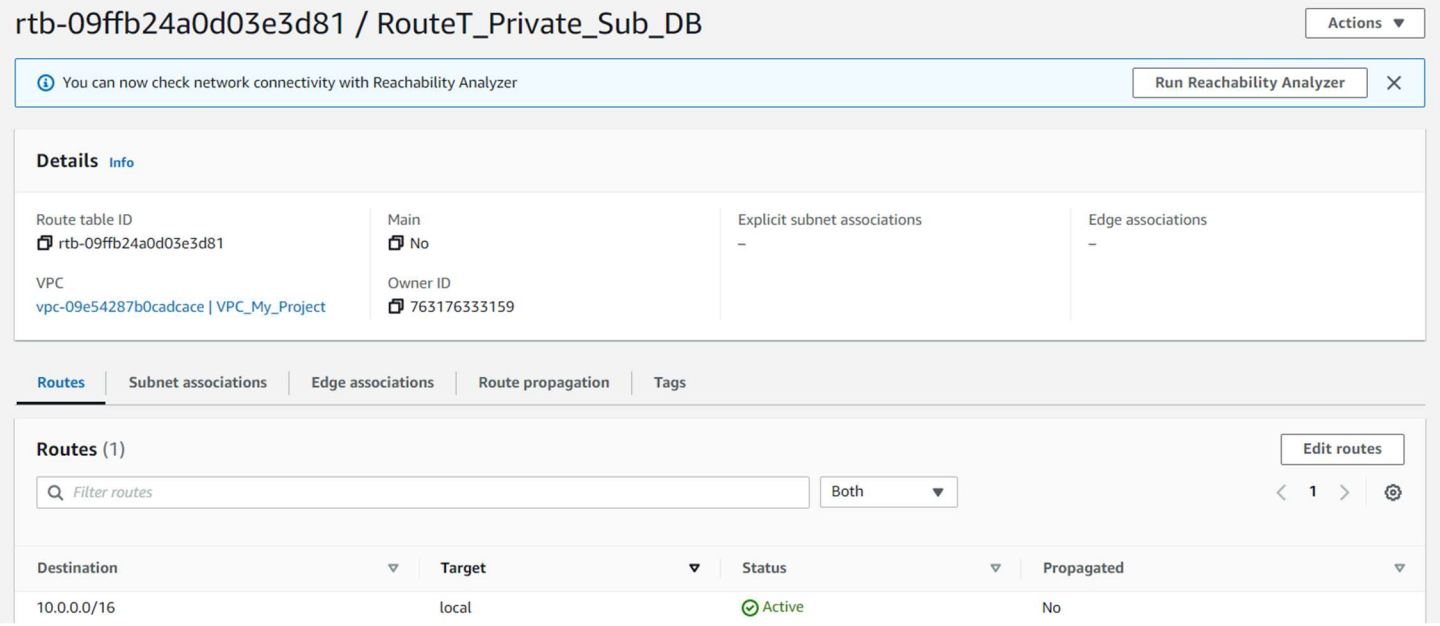
STEP 9:

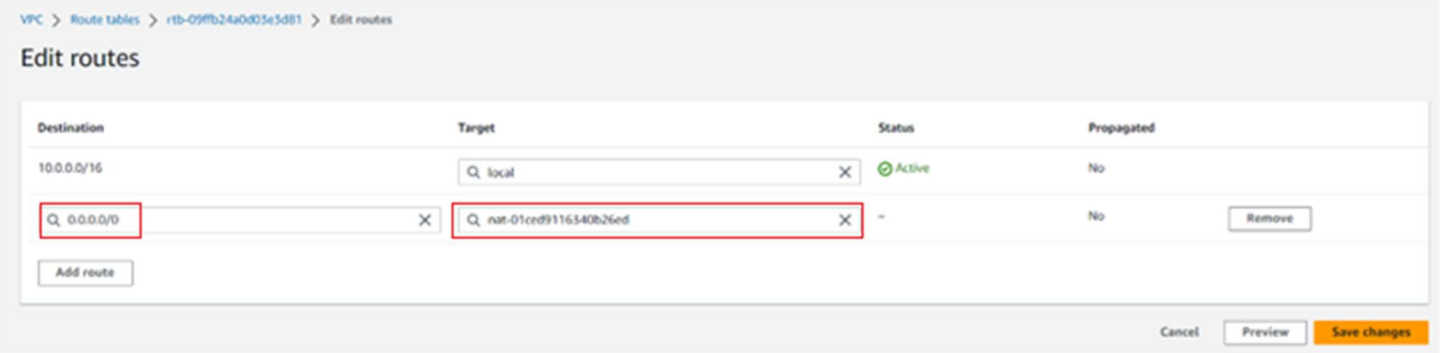
CREATION OF PRIVATE ROUTE TABLES FOR THE DATABASE TIER SUBNETS

(d) Create the private Route Table **(RouteT_Private_Sub_DB)**



(e) Edit the route table **(RouteT_Private_Sub_DB)** to route traffic to the second NAT gateway **(NATGW2_My_Project)** that resides in the public subnet **(Public_Sub_Web_1b).** For this configuration, simply click on the Route table **(RouteT_Private_Sub_DB),** select edit routes, select add route, in the blank field, type in **0.0.0.0/0** and then scroll down to select NAT gateway **(NATGW2_My_project)** as target, then click **save changes.**

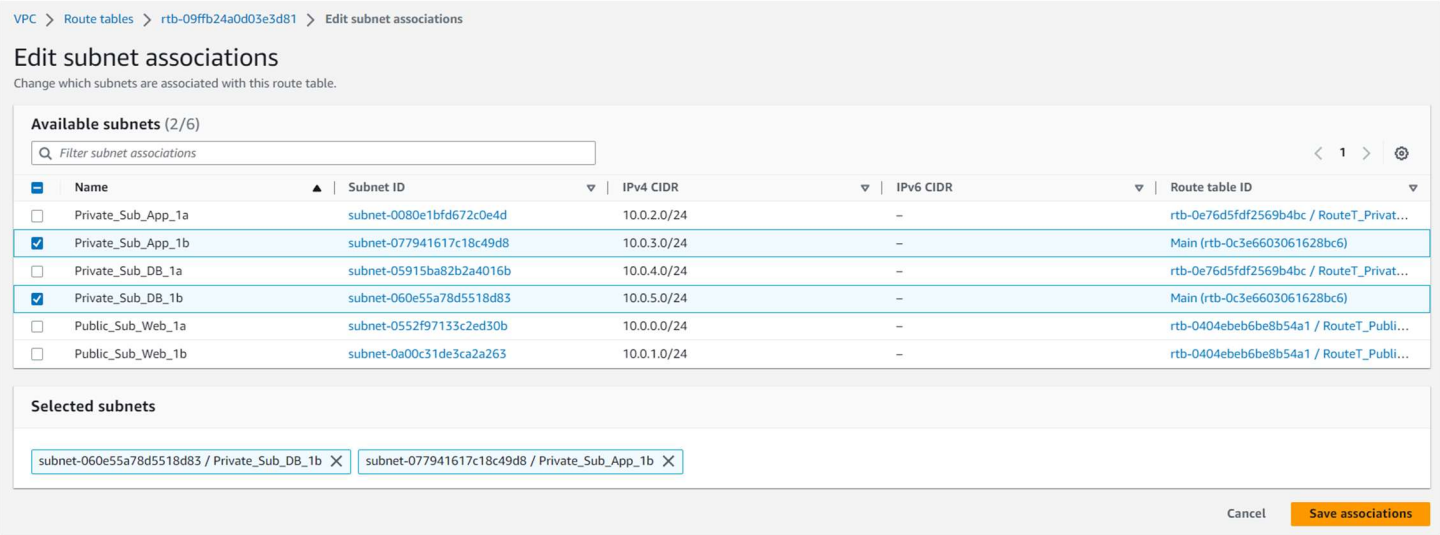




(f) Edit subnet association

To edit the subnet association of the route table **(RouteT_Private_Sub_DB)** linked to the 2 private subnets **(Private_Sub_App_1b)** and **(Private_Sub_DB_1b),** simply click on "Subnet

associations", proceed to click on "Edit subnet associations", select the private subnets **Private_Sub_App_1b** and **Private_Sub_DB_1b** then click on **save associations.**



Private Subnets (**Private_Sub_App_1b** and **Private_Sub_DB_1b)** successfully associated to the Route table **(RouteT_Private_Sub_DB)** as shown above.

STEP 10:

CREATION OF SECURITY GROUPS

Let’s confirm that we are in the right region **(us-east-1).** Now let's navigate to the EC2 dashboard, scroll down to select Security Groups.

Click on create security group, give it a name, add description, and specify your VPC.

Choose the security group rule that will control our web server inbound and outbound traffic.

When traffic flows in from the internet, it must flow through the Application load balancer (ALB) before entering the web servers.

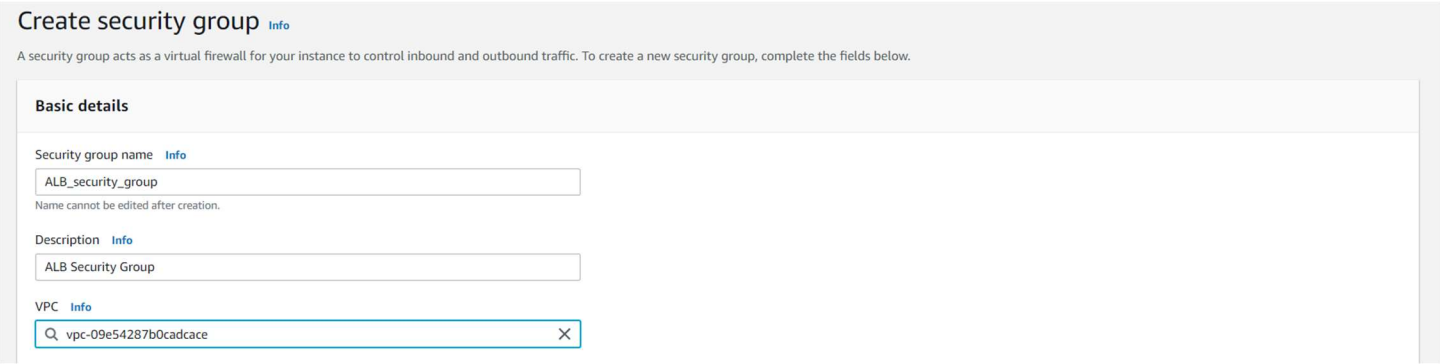
IMPORTANT: It is important to note that the Web Servers will only accepts traffic that flows in from the internet through the Application Load Balancer and to the Web Server.

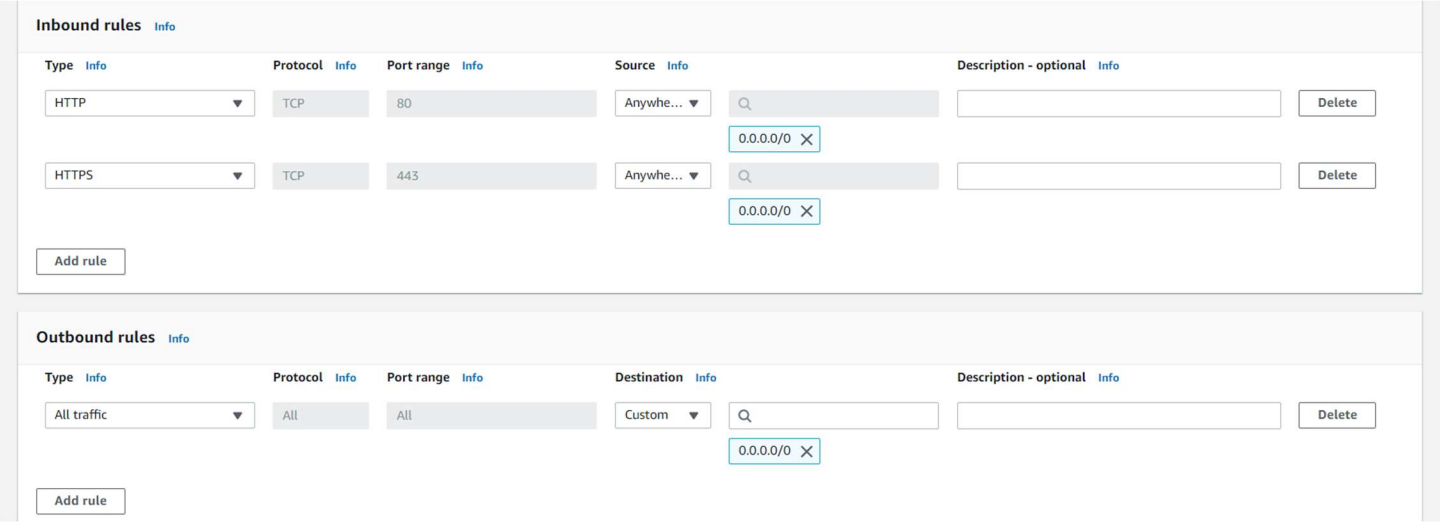
IMPORTANT: It also very important to know that the Database servers will only accept traffic that flows in from the Web Servers.

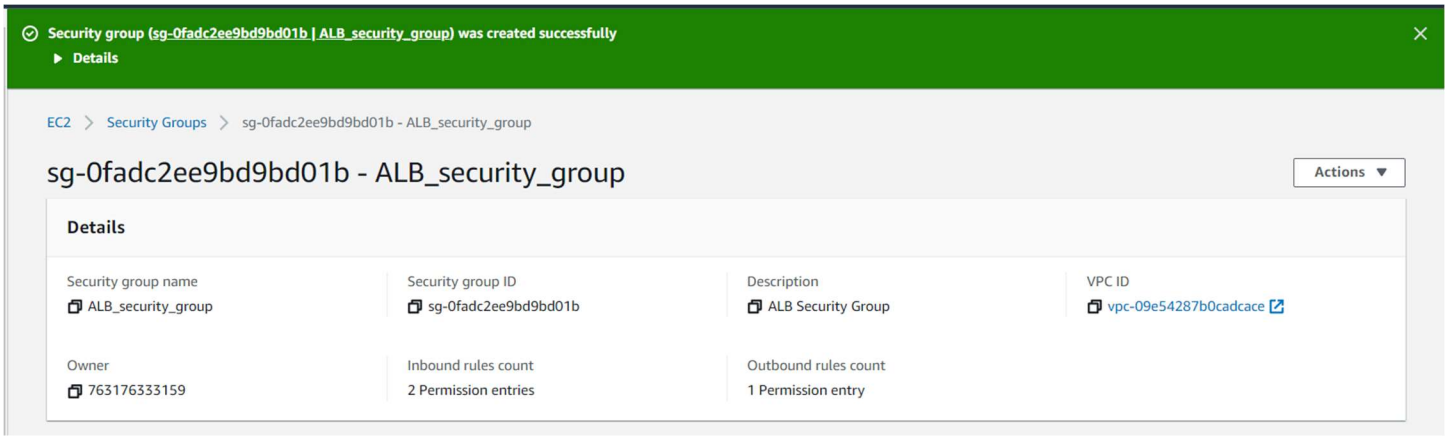
IMPORTANT: Finally, it is important to know that the EFS file system servers will only accept traffic that flows in from the Web Servers as well.

Now, let's dive in to create our security groups.

(a)Lets create the security group for our Application Load balancer **(ALB_security_group)**

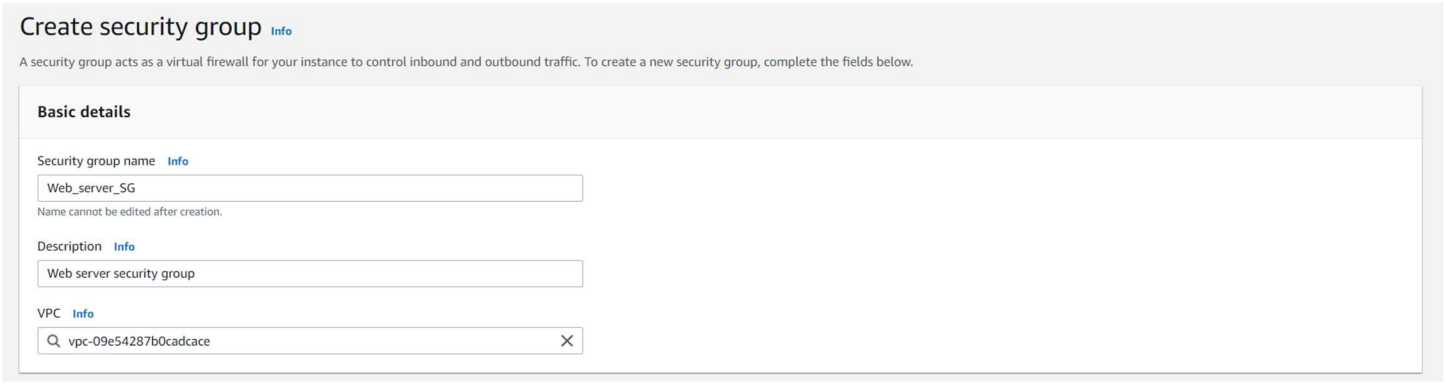


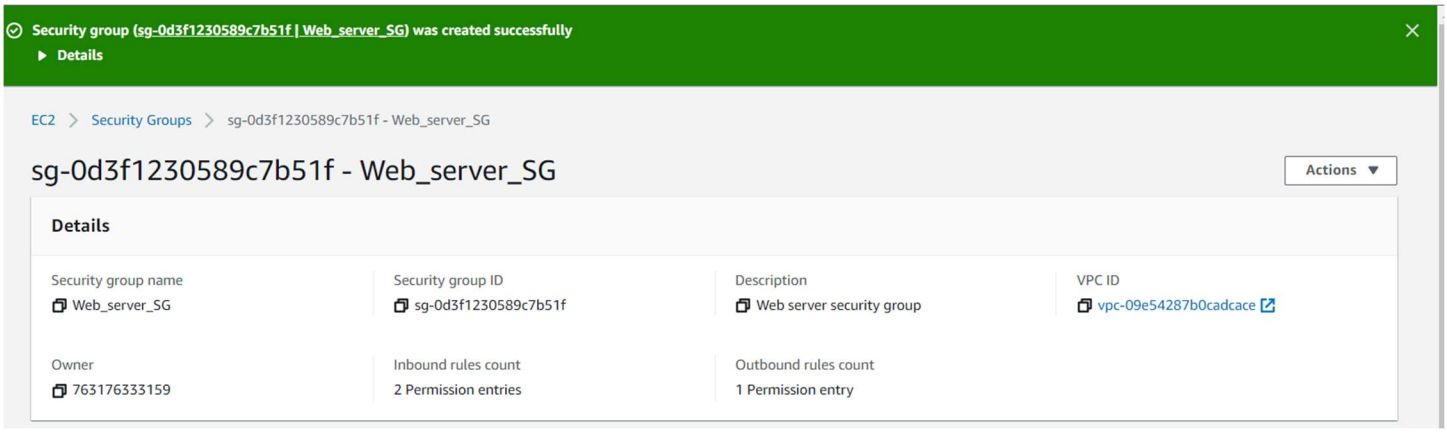




ALB_Security_group was successfully created.

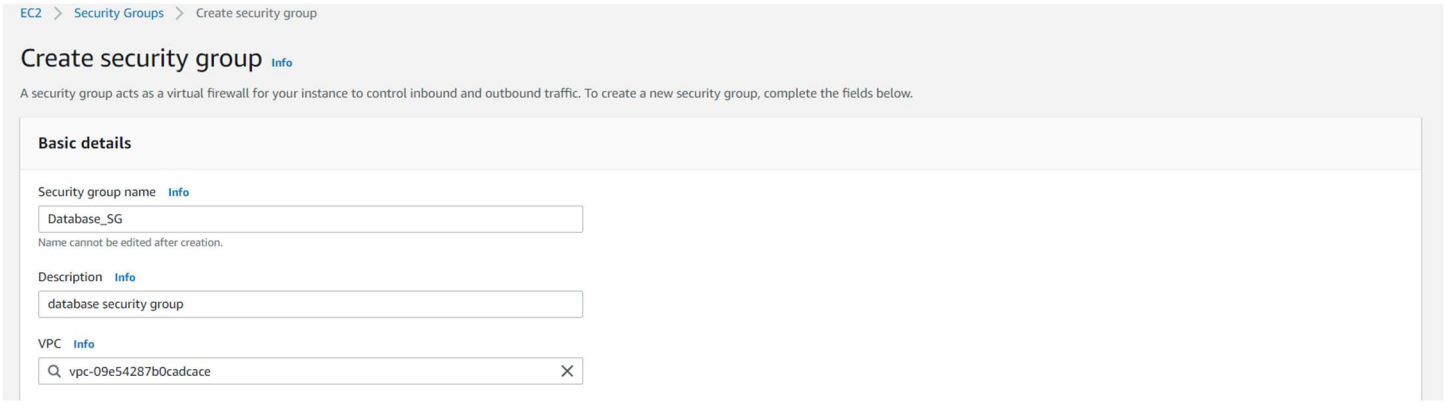
(b)Lets create the security group for our web servers **(Web_server_SG)**

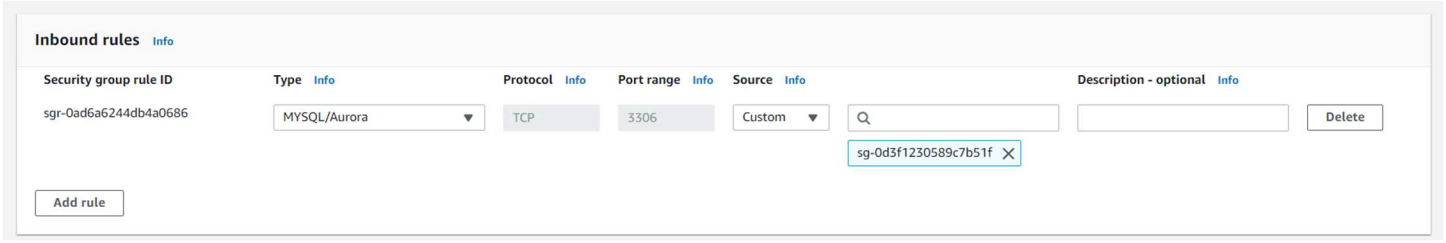


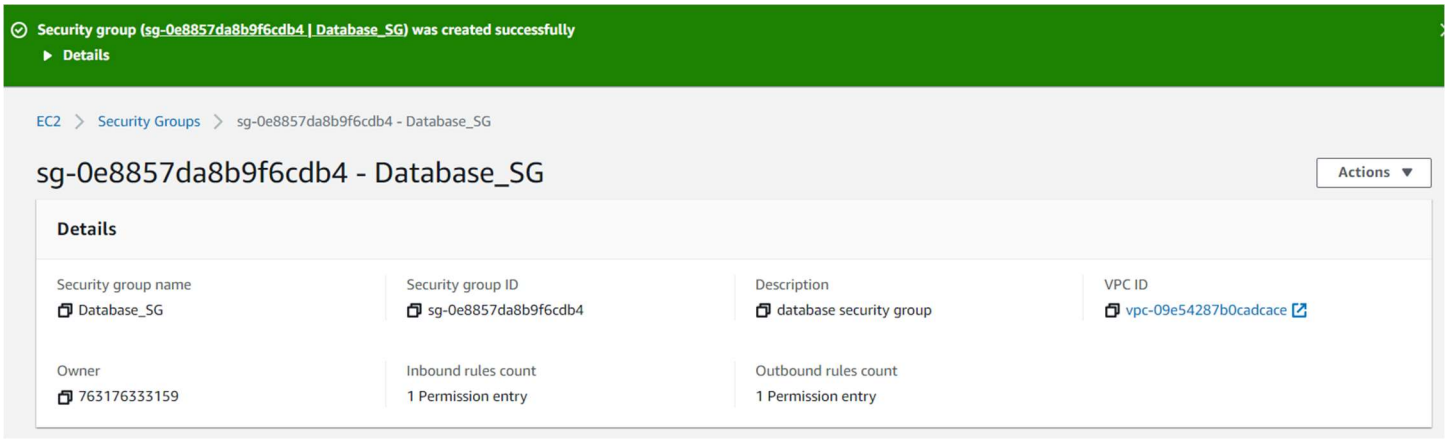


Web_Security_group was successfully created.

(c)Lets create security group for our database **(Database_SG)**

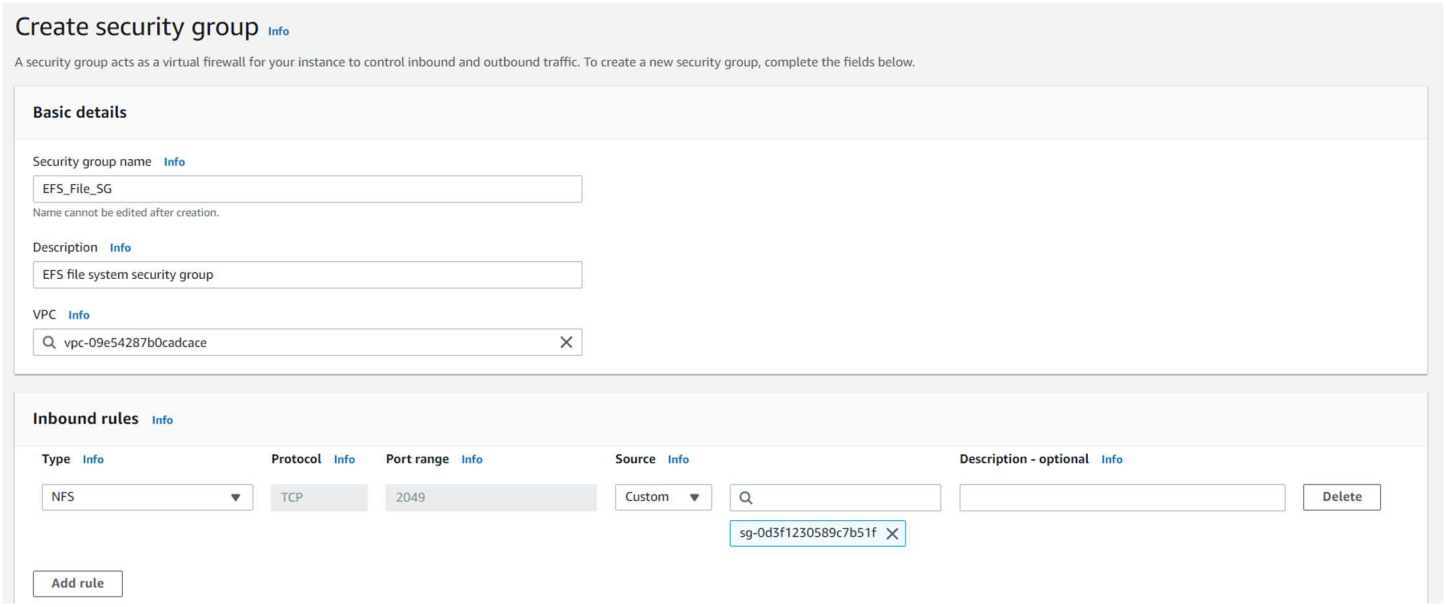


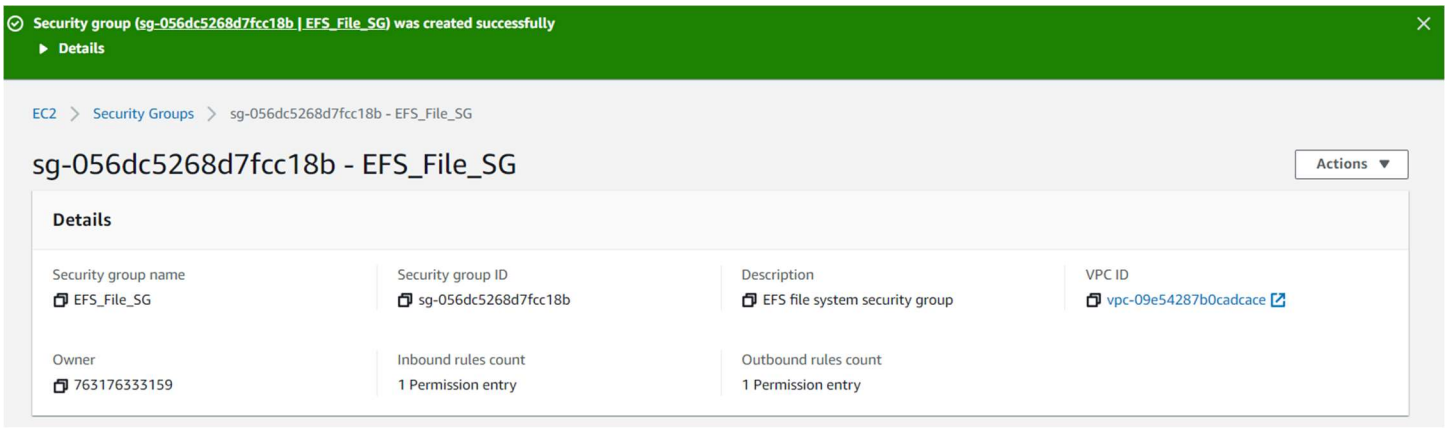




Database_SG was successfully created.

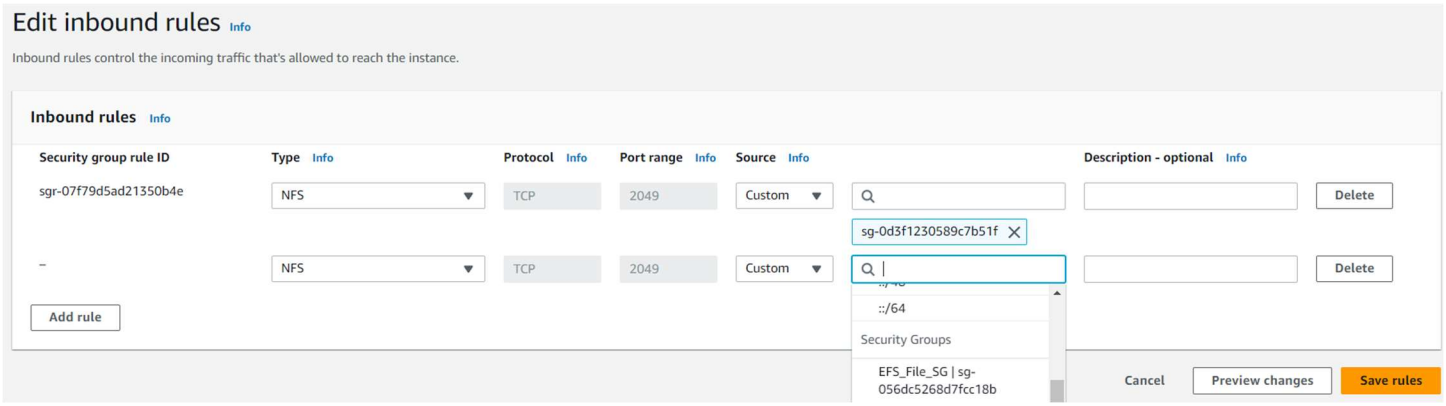
(d)Lets create security group for our EFS File system **(EFS_File_SG)**

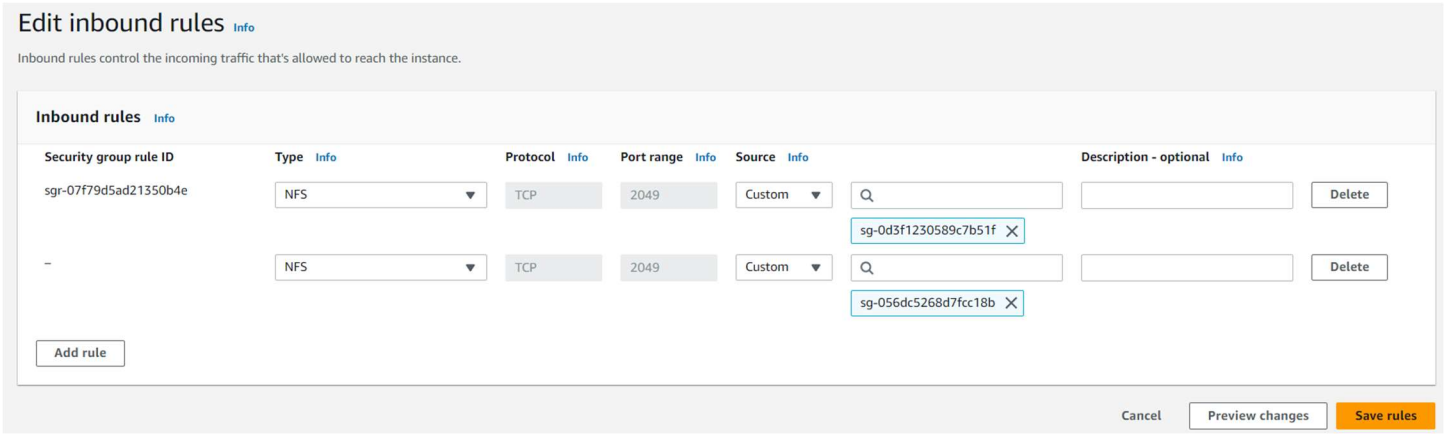


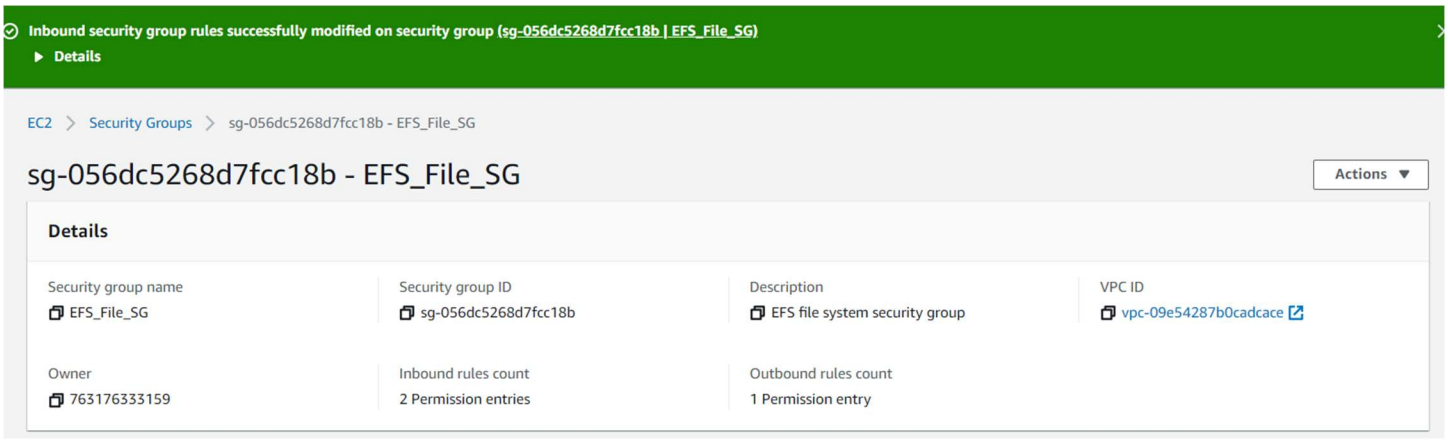


EFS_File_SG has been successfully created.

Let's edit the inbound rule of the NFS security group to add EFS security group **(EFS_File_SG)** as source for the NFS security group.







Finally, the inbound rule for the NFS security group has been edited and added to EFS security group **(EFS_File_SG)** as traffic source for the NFS security group.

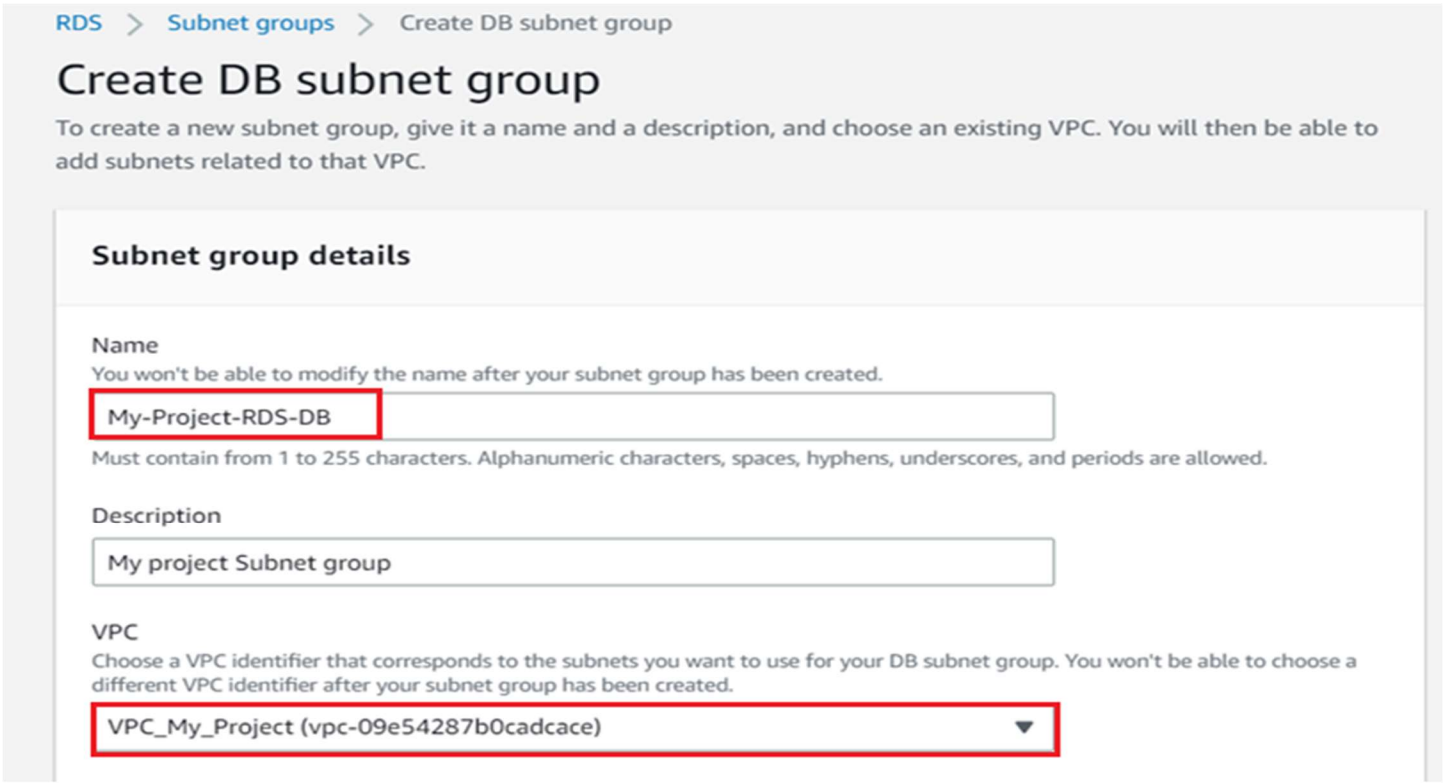
STEP 11:

CREATE THE DATABASE SUBNET GROUP THAT WOULD BE USED TO CREATE THE RDS DB.

First, we must confirm that we are in the right **region (N. Virginia us-east-1)** where our RDS databases will be launched. Let's carefully follow the steps below to create our RDS Databases.

RDS Engine version 5.7.42 is what we are going to use for this project.

IMPORTANT: First thing to do is to RDS > Subnet groups > Create DB subnet group



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-060e55a78d5518d83 (10.0.5.0/24) ✕

subnet-05915ba82b2a4016b (10.0.4.0/24) ✕

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-060e55a78d5518d83	10.0.5.0/24
us-east-1a	subnet-05915ba82b2a4016b	10.0.4.0/24

Cancel **Create**

Subnet groups (3)

Edit | Delete | **Create DB subnet group**

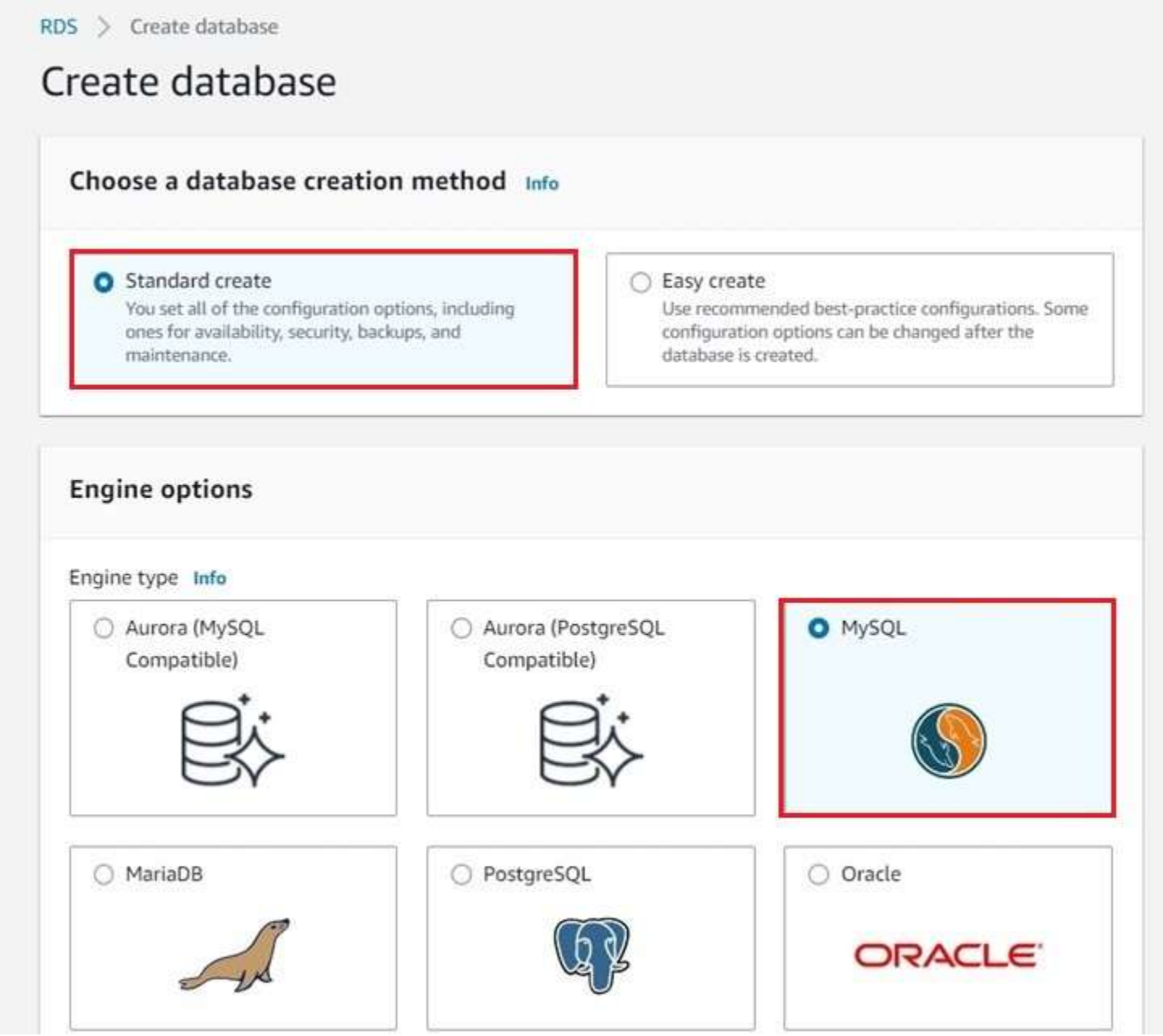
Filter by subnet group

1

Name	Description	Status	VPC
default-vpc-08328352666abbbb8	Created from the RDS Management Console	Complete	vpc-08328352666abbbb8
default-vpc-09e54287b0cadcace	Created from the RDS Management Console	Complete	vpc-09e54287b0cadcace
my-project-rds-db	My project Subnet group	Complete	vpc-09e54287b0cadcace

Successfully created Database subnet groups.

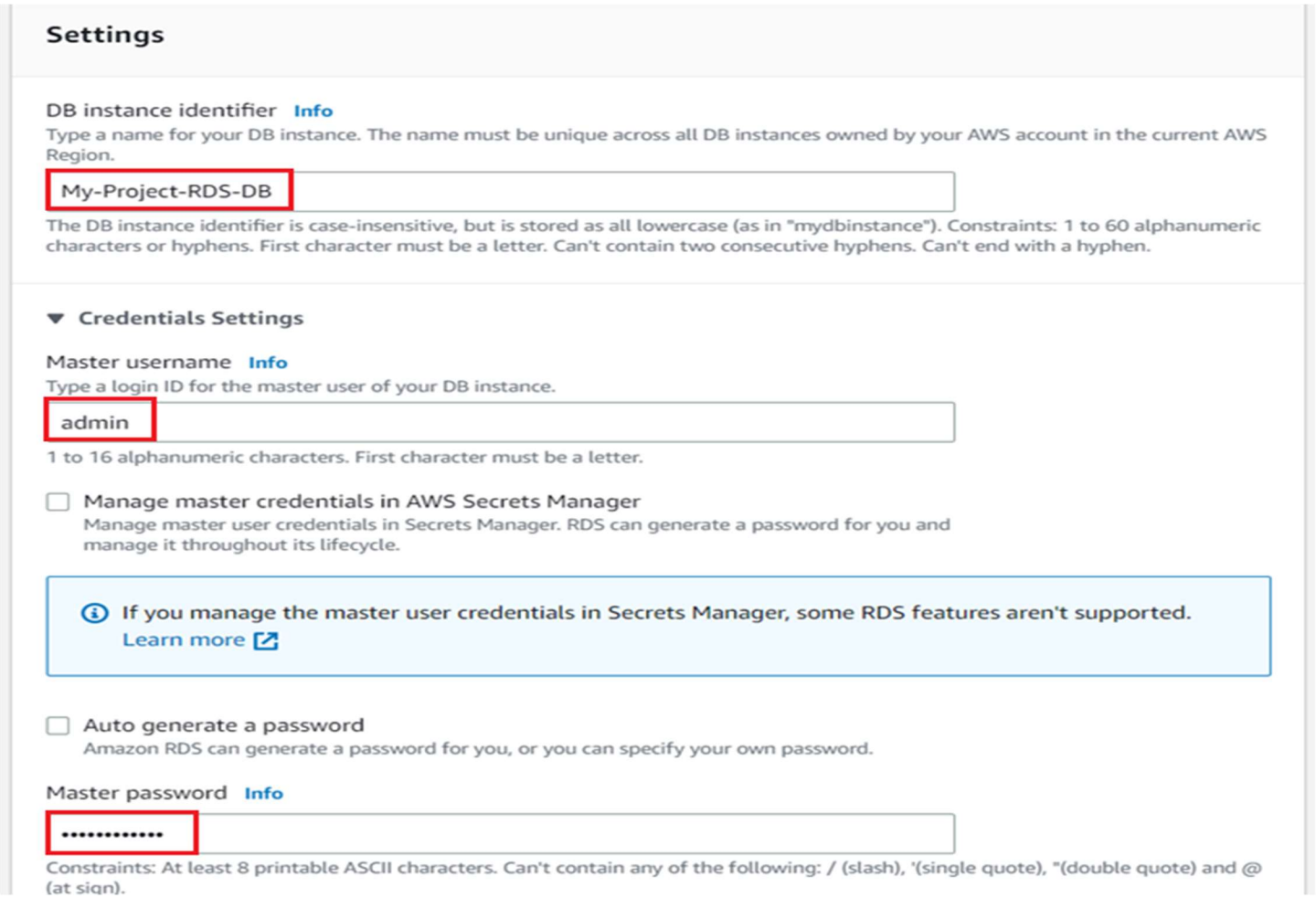
Now, let's navigate to **RDS,** click on create database, select "**Standard create"** and Choose **MySQL.**



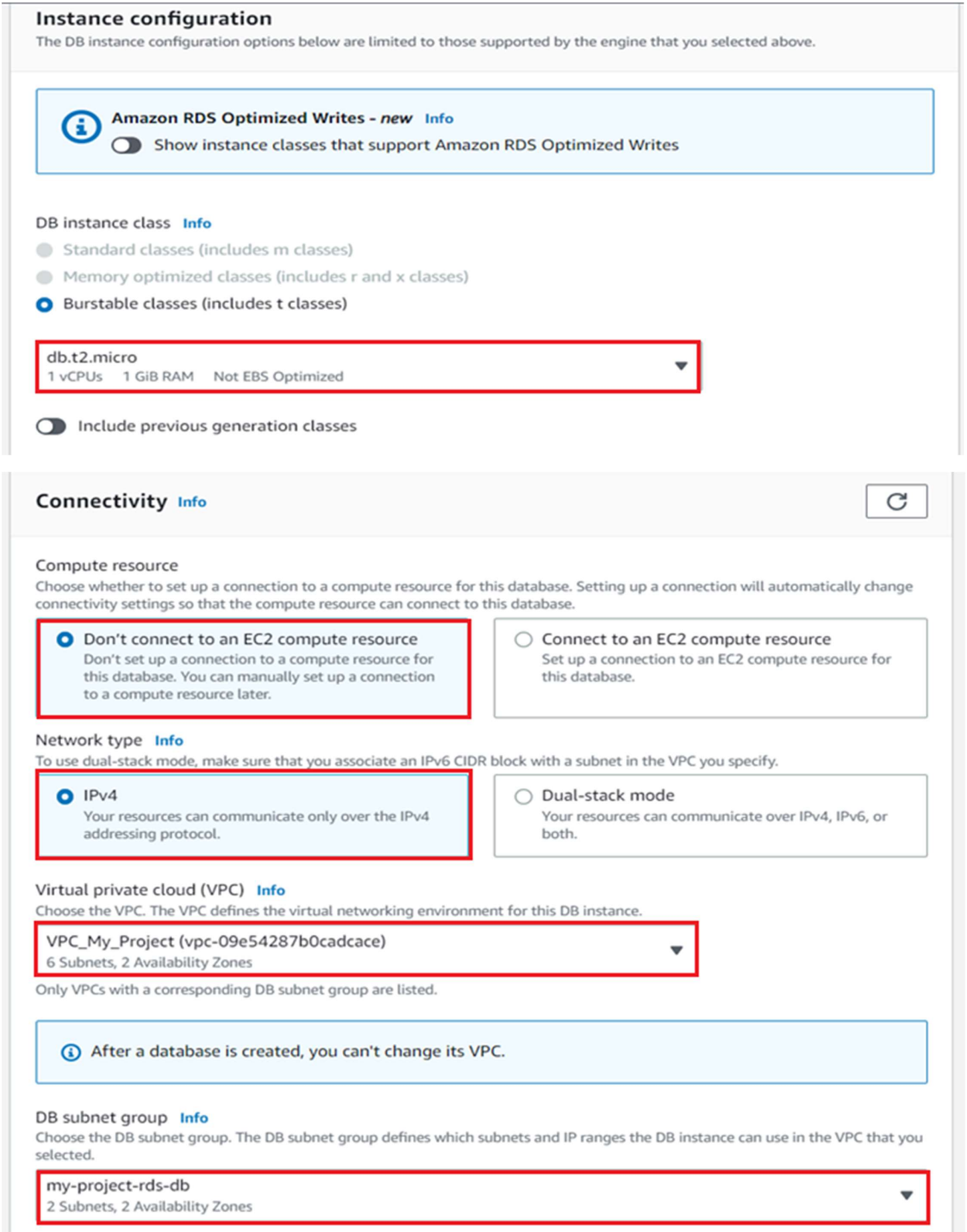
We want to leave Engine Version as default MySQL 5.7.42

Then entering the name (It is important to note that First character must be a letter. Can't have two consecutive hyphens (- -), it can't contain underscores (_), and it can't end with a hyphen.)

Next step will be to scroll down to settings, enter DB identifier and master password.



Scroll down to networking, select Burstable classes, and select db.t2.micro

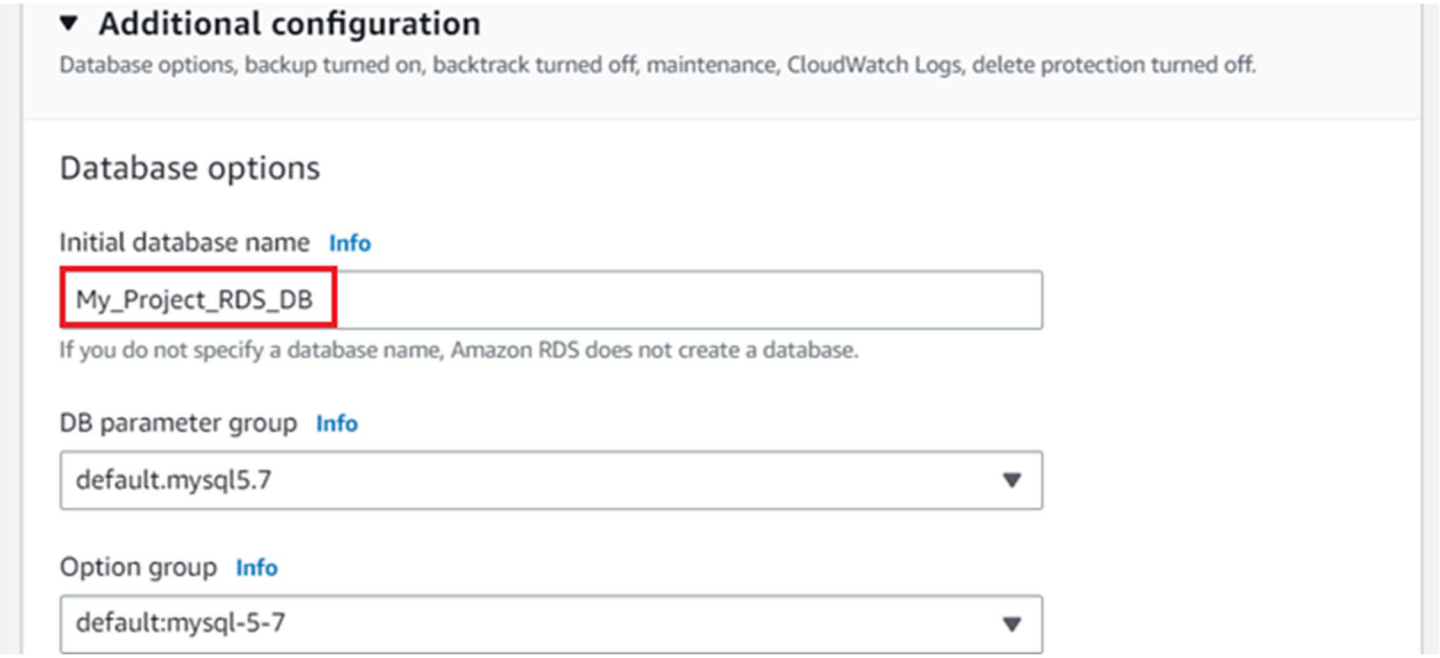


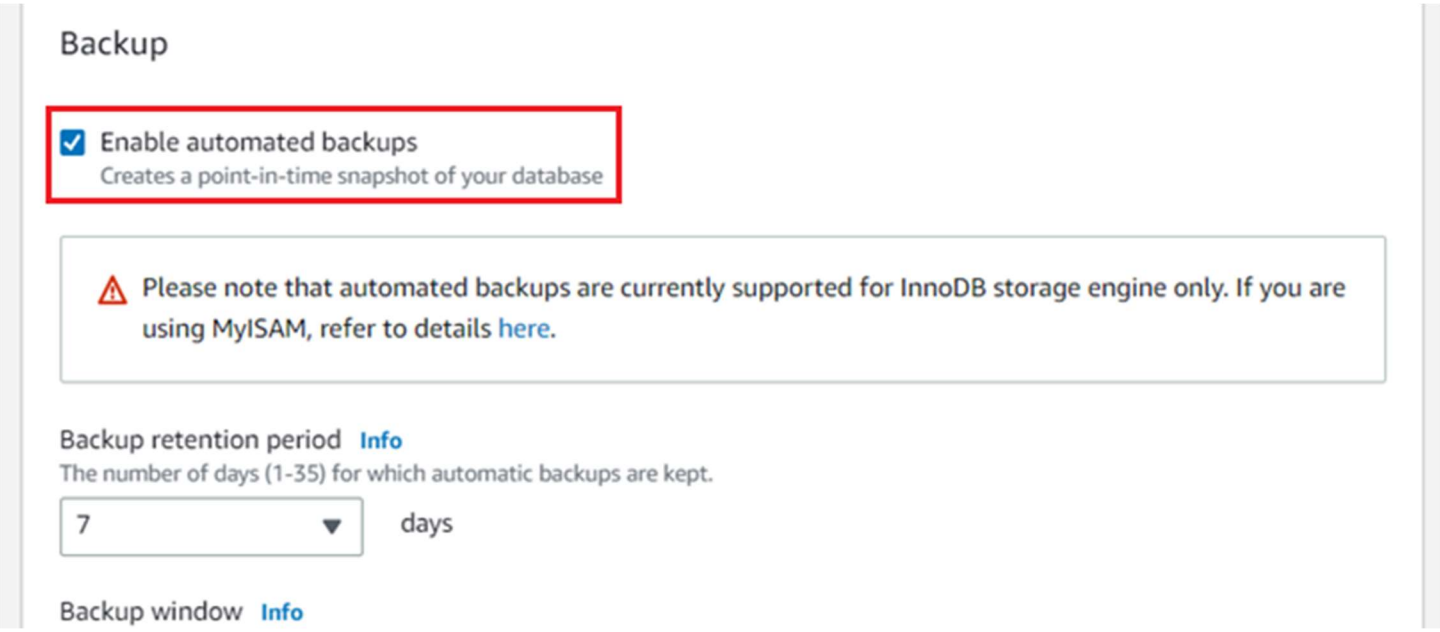
Scroll down to Additional configuration > Database options.

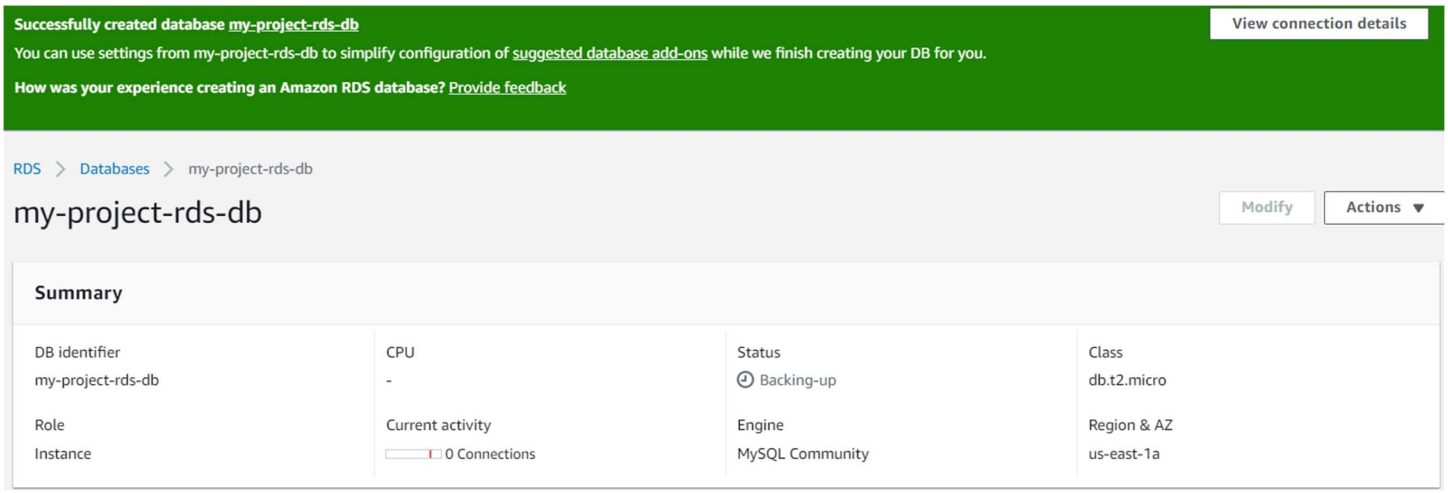
All we need to do, is give a database name and maintain all other default values as they are.

IMPORTANT: Note that the "Initial database name" field accepts letters, numbers, underscores (_)

Navigate to the end of the page and then **create** a database.







Database successfully created.

STEP 12:

CREATE EFS FILE SYSTEM

We are going to create an Elastic File System (EFS) and create mount targets in each of the Database subnets **(Private_Sub_DB_1a)** and **(Private_Sub_DB_1b)** in each availability zone.

Then, we will put our application code in the created File System so that our Web Servers can pull our application code from the same location.

To create our EFS file system, navigate to:

services > storage > EFS > create file system > select VPC > customize.

Create file system

Create an EFS file system with recommended settings, including Elastic Throughput, Lifecycle Management, and Automatic Backups. These settings are designed to optimize the price-performance of your file system. Learn more

Name - *optional*

Name your file system.

Optional. Apply a name to your file system

Name can include letters, numbers, and +-=._:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)

Choose the VPC where you want EC2 instances to connect to your file system.

vpc-09e54287b0cadcace
VPC_My_Project

Cancel | Customize | Create

File system settings

General

Name - *optional*

Name your file system.

My_Project_EFS

Storage class | Learn more

- (•) Standard — Stores data redundantly across multiple AZs
- ( ) One Zone — Stores data redundantly within a single AZ

Automatic backups

Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. Learn more

[x] Enable automatic backups

Lifecycle management

EFS Intelligent-Tiering uses Lifecycle Management to automatically achieve the right price and performance blend for your application by moving your files between the Standard and Standard-Infrequent Access storage classes. Learn more

Transition into IA
Transition files from Standard to Standard-Infrequent Access.
30 day(s) since last access

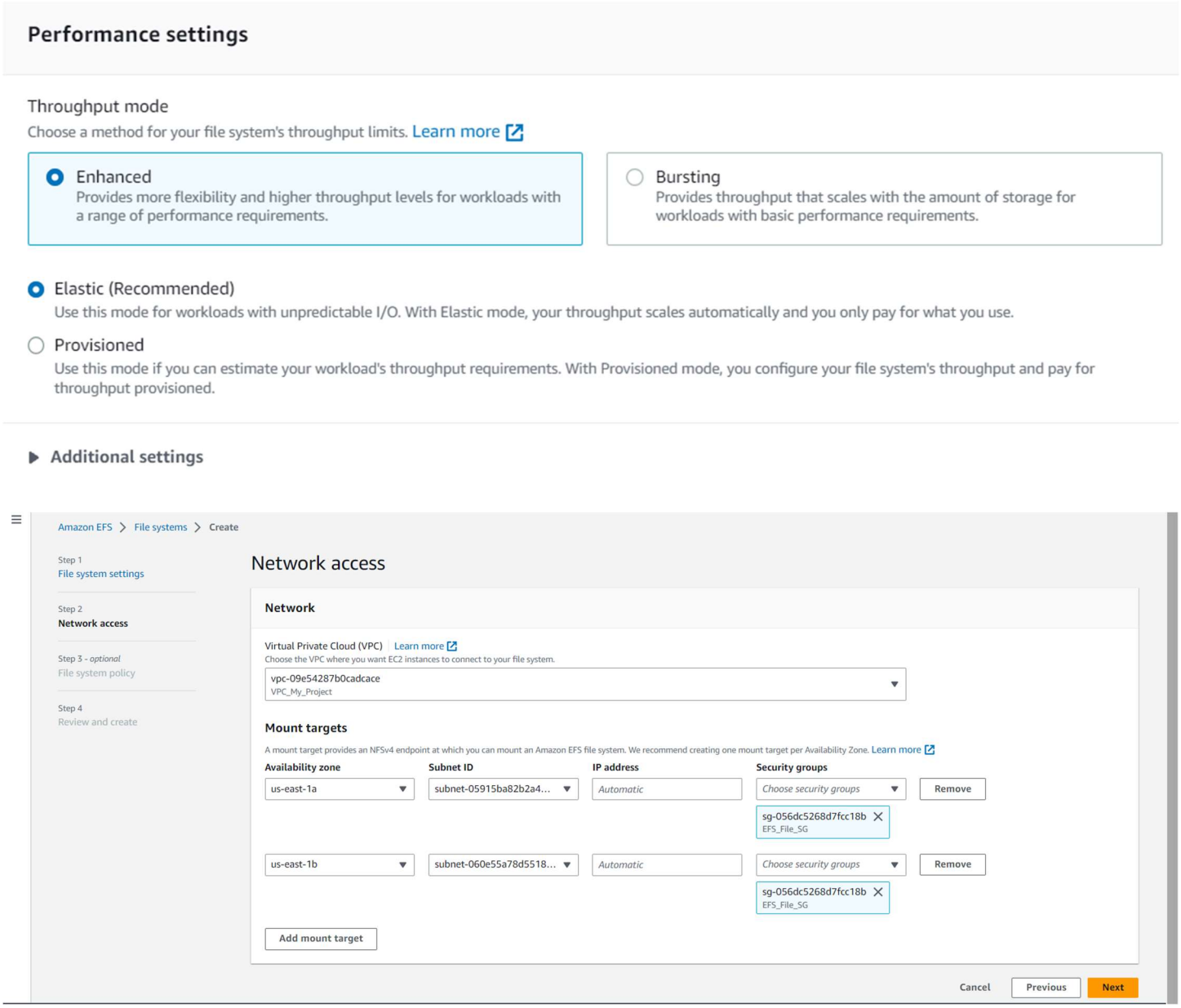
Transition out of IA
Transition files from Standard-Infrequent Access to Standard.
None

Encryption

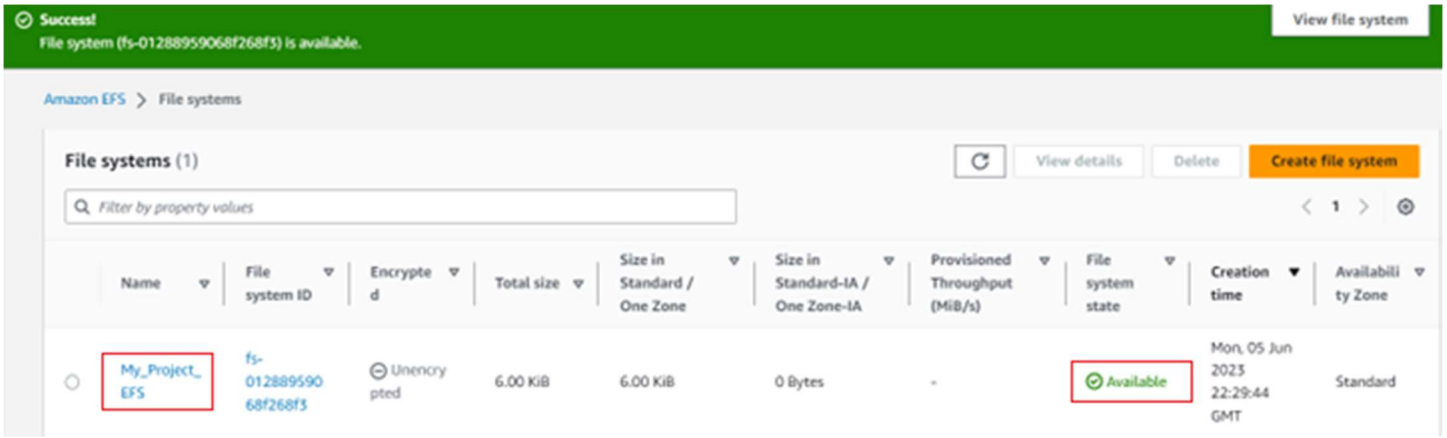
Choose to enable encryption of your file system's data at rest. Uses the AWS KMS service key (aws/elasticfilesystem) by default. Learn more

[ ] Enable encryption of data at rest

At this point, the file system name is optional, so we are not going to include any at this point.



Review and create file system.

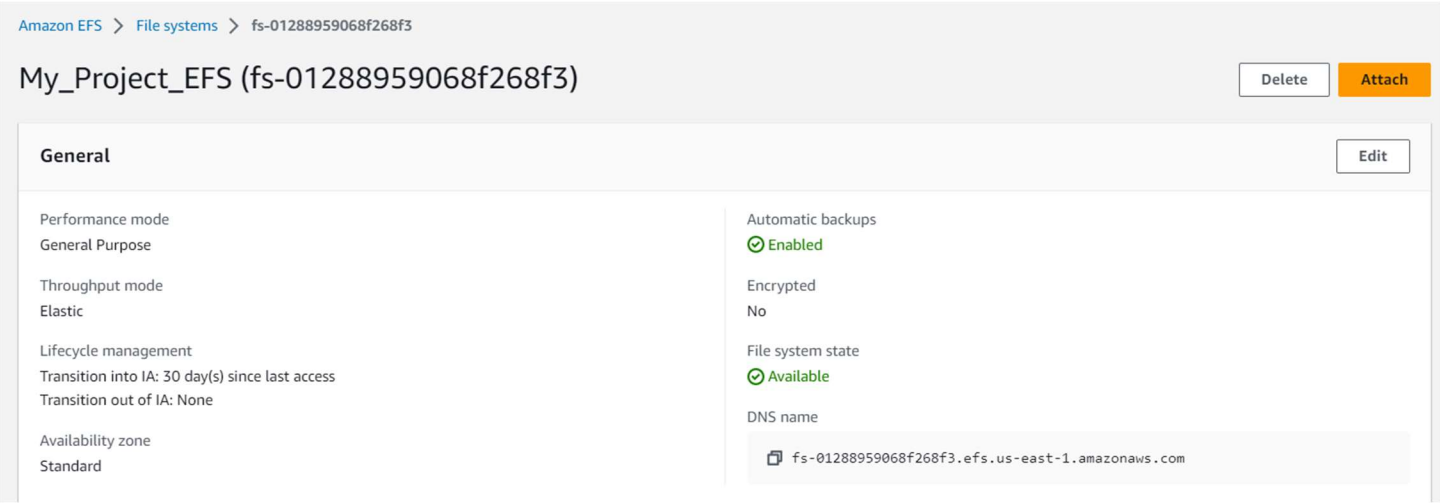


File system successfully created.

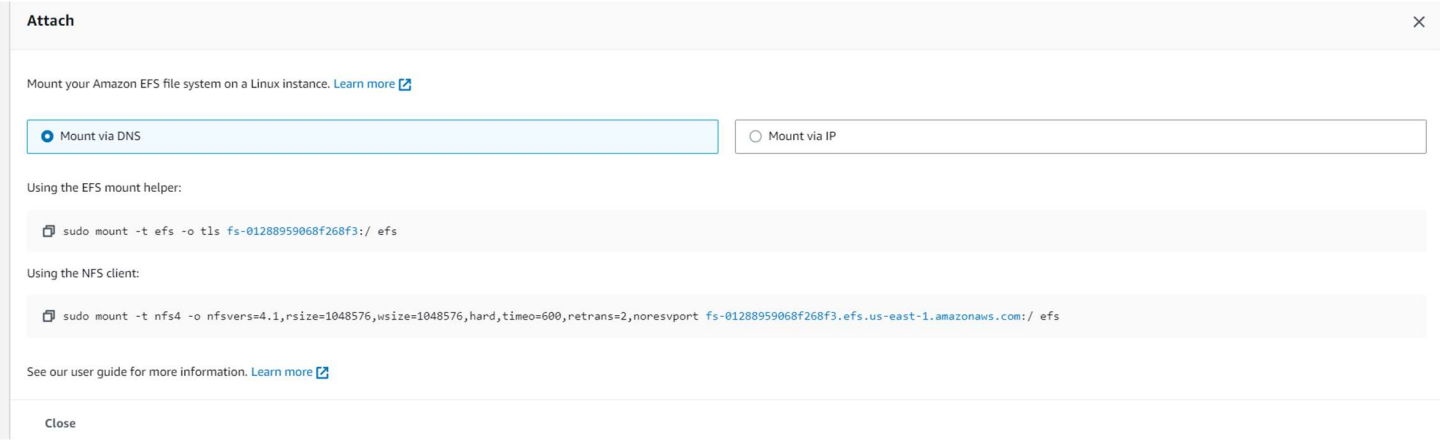
The file system state is **available** as shown in the above screenshot.

Click on **My_Project_EFS**

Click on **Attach**



This will display the code that will be used to attach the file system to the web servers

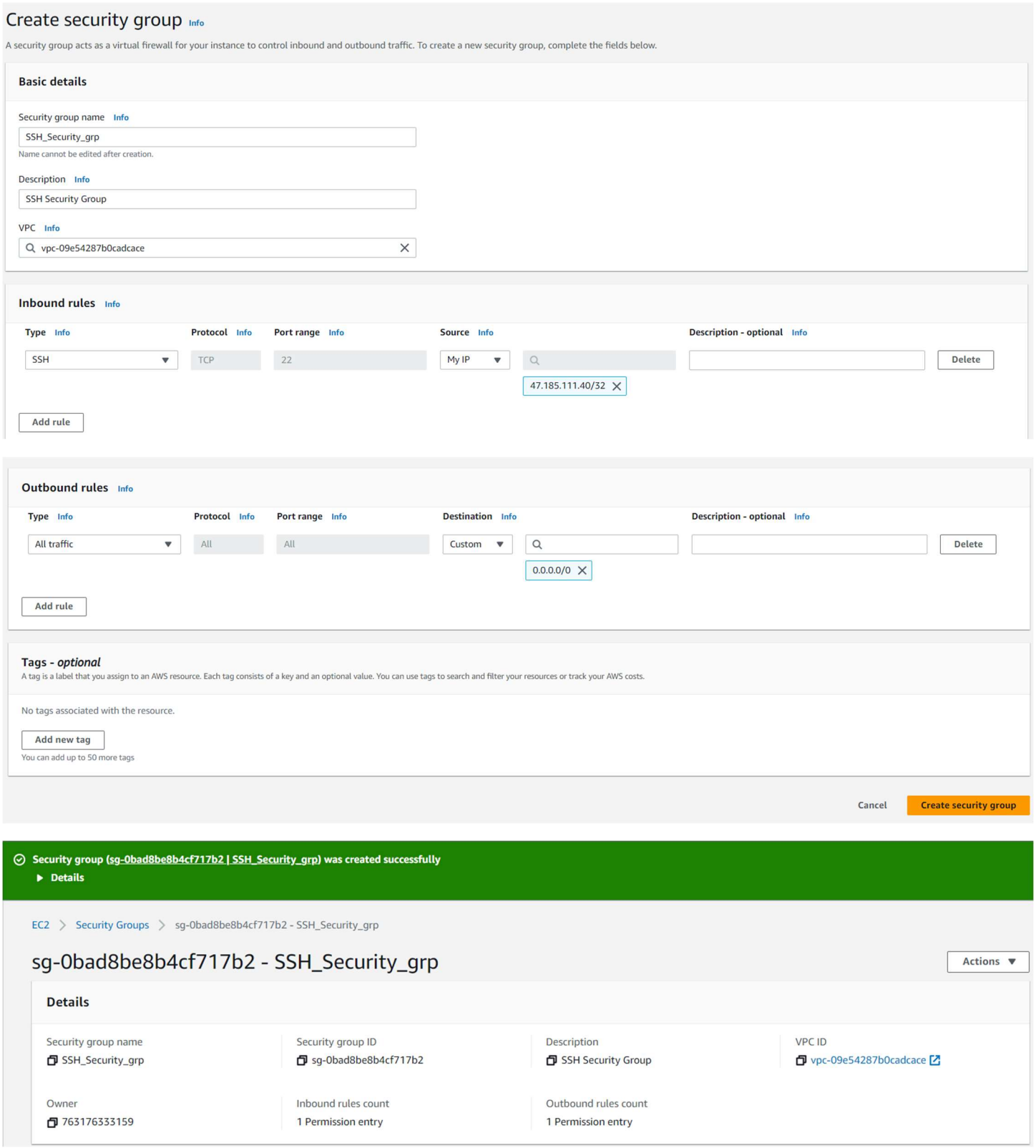


Now, the next step will be to mount the NFS file system.

STEP 13: HOW TO USE AMAZON LINUX 2 TO INSTALL WORDPRESS AND MOVE FILES TO EFS

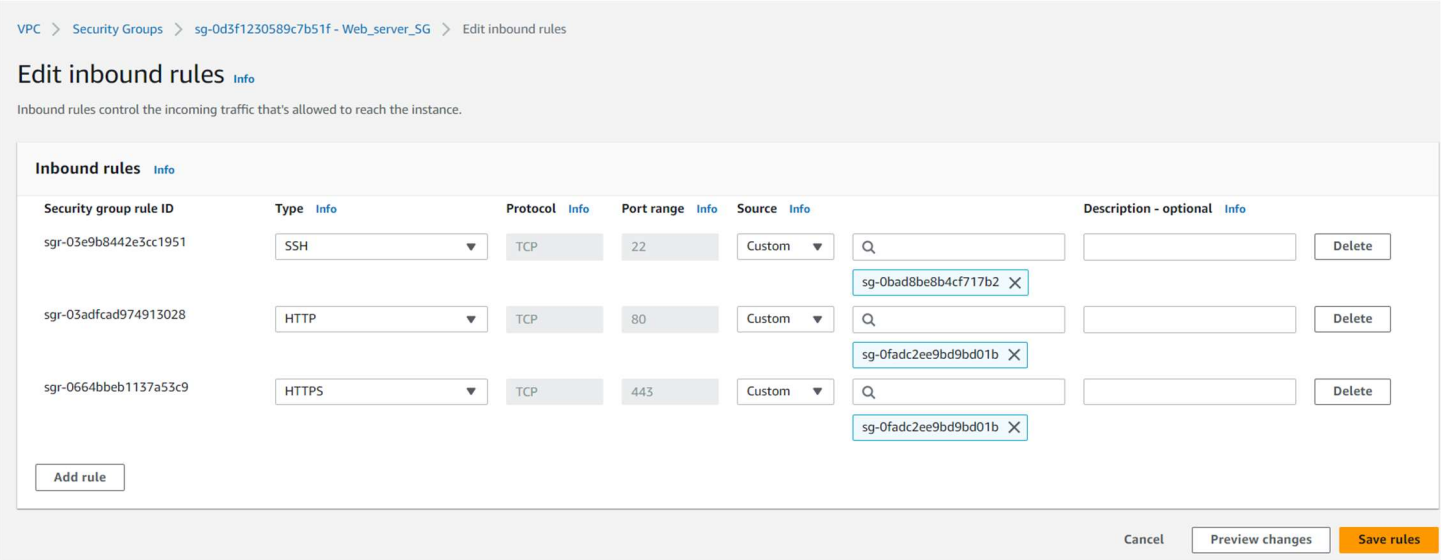
Let’s navigate to our EC2 console to create a new SECURITY GROUP to enable us SSH into our EC2 web servers.

EC2 Dashboard > Security group > Create Security group.



SSH Security Group was successfully created.

Now let's edit the inbound rules of our Web Servers **(Web_server_SG)** security group.



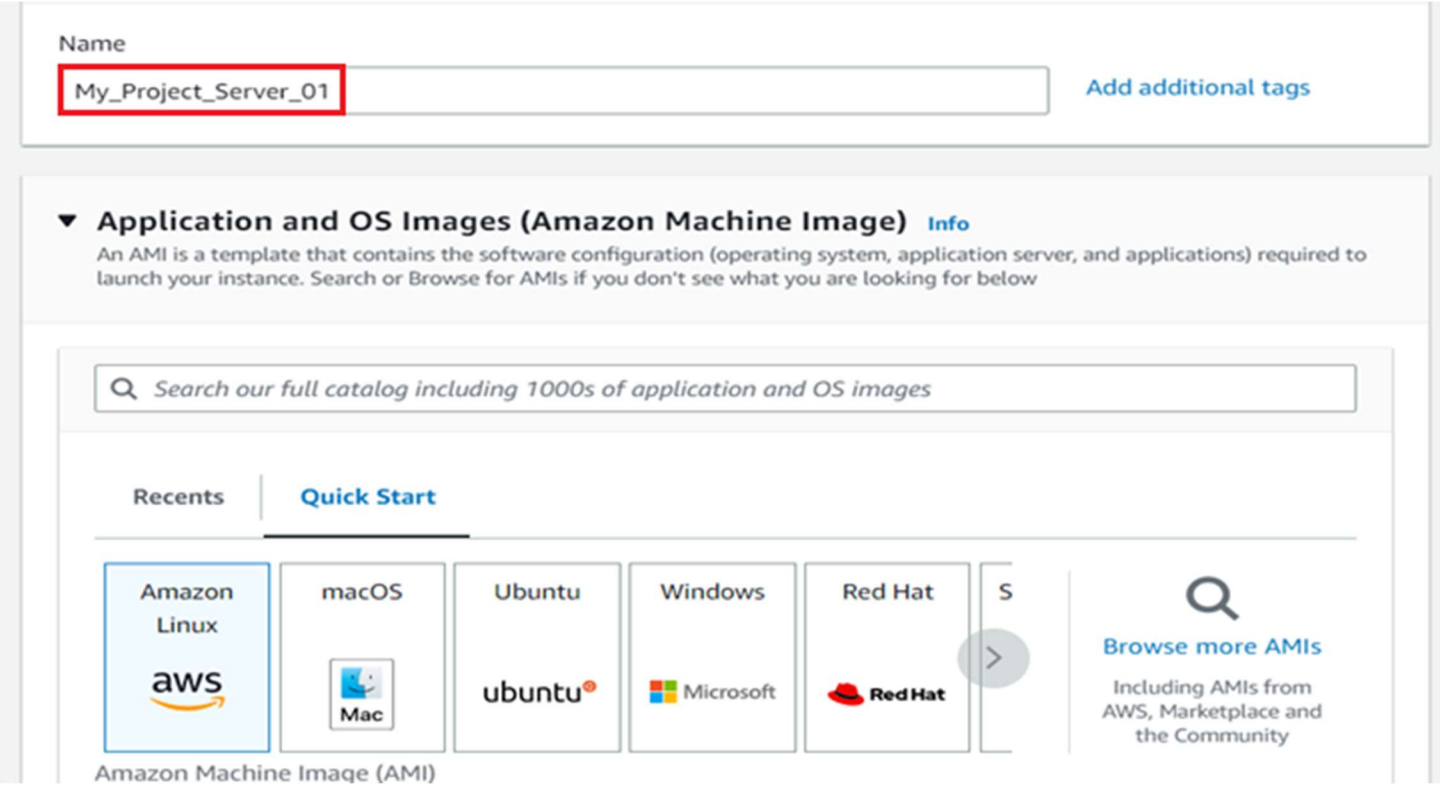
Now let's edit the inbound rules of our **(EFS_File_SG)** security group.

SSH was added to the inbound rules of **(Web_server_SG)** and EFS **(EFS_File_SG)** security groups respectively.

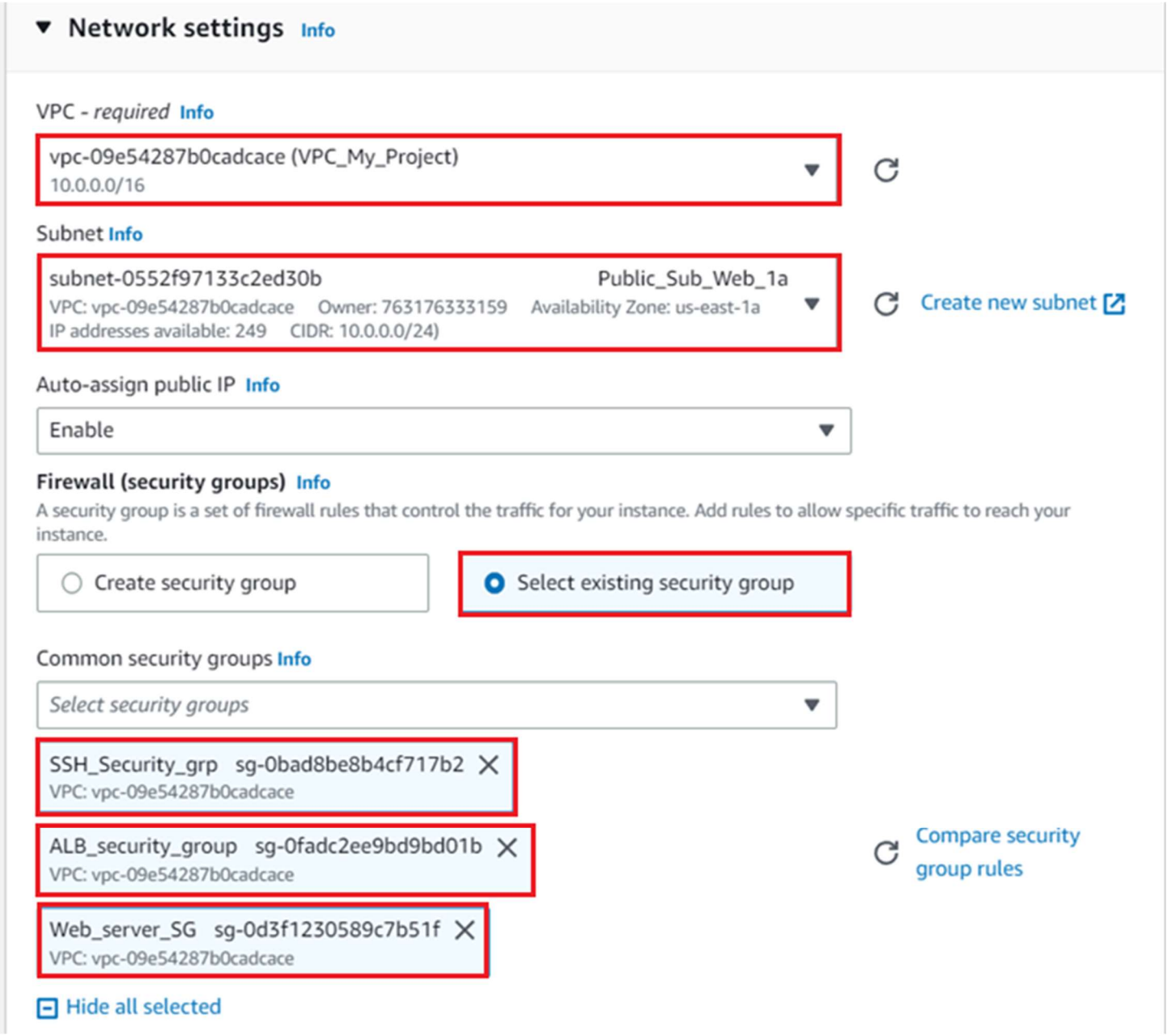
STEP 14: CREATION OF TWO EC2 WEB SERVERS

Now, let's create two (2) EC2 Linux Servers in our Public Web Tier Subnets:

(Public_Sub_Web_1a) and **(Public_Sub_Web_1b).**



Network Settings for Web Server **(My_Project_Server_01)** launched in Public Web Subnet. **(Public_Sub_Web_1a)**



Network Settings for Web Server **(My_Project_Server_02)** is launched in Public Web Subnet. **(Public_Sub_Web_1b)**

Name

My_Project_Server_02

Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

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

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Network settings Info

VPC - *required* Info

vpc-09e54287b0cadcace (VPC_My_Project)
10.0.0.0/16

Subnet Info

subnet-0a00c31de3ca2a263 Public_Sub_Web_1b
VPC: vpc-09e54287b0cadcace Owner: 763176333159 Availability Zone: us-east-1b
IP addresses available: 250 CIDR: 10.0.1.0/24)

Create new subnet

Auto-assign public IP Info

Enable

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

SSH_Security_grp sg-0bad8be8b4cf717b2
VPC: vpc-09e54287b0cadcace

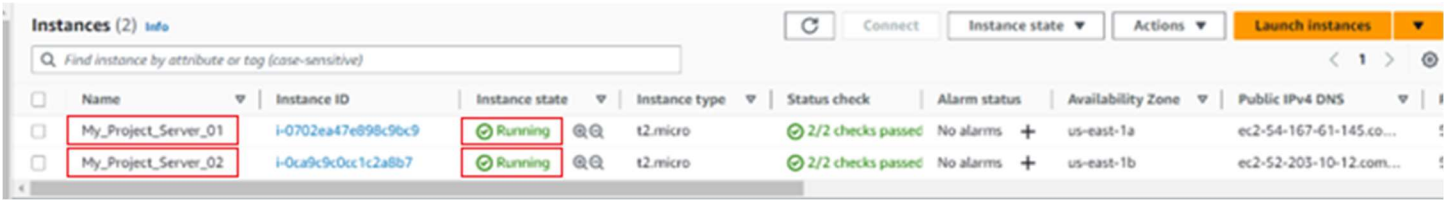
Web_server_SG sg-0d3f1230589c7b51f
VPC: vpc-09e54287b0cadcace

ALB_security_group sg-0fadc2ee9bd9bd01b
VPC: vpc-09e54287b0cadcace

Compare security group rules

Hide all selected

The 2 Web servers **(My_Project_Server_01)** and **(My_Project_Server_02)** have been successfully launched and they are running as you can see below.



STEP 15:

CONNECTION TO OUR EC2 WEB SERVERS VIA SSH

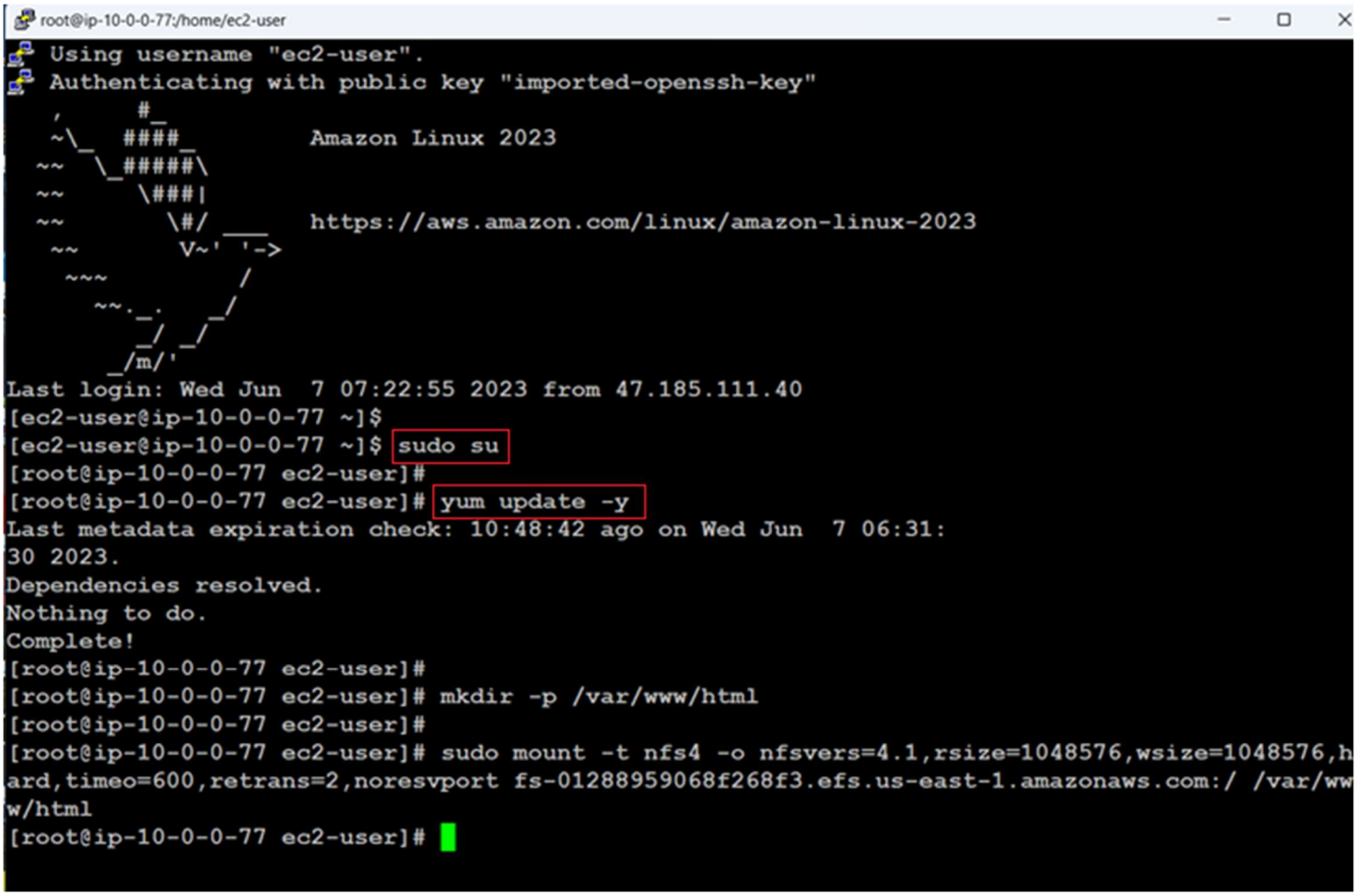
I'm using a Windows 11 machine for this Project, so I will be connecting to my EC2 Web Servers using putty.

Established a connection to the Server using Putty.

Create the html directory and mount the efs to it

Ran the **sudo su** to switch user to root and then ran **yum update -y** to update the Web Server.

mkdir -p /var/www/html to create a new directory.



sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-01288959068f268f3.efs.us-east-1.amazonaws.com:/ /var/www/html

Installed Apache

sudo yum install httpd -y httpd-tools mod_ssl.

sudo systemctl enable httpd to enable the httpd service.

sudo systemctl start httpd to start the service.

sudo systemctl status httpd to check the status of the httpd daemon if it is running or not.

```
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# yum install httpd -y httpd-tools mod_ss
Last metadata expiration check: 11:05:21 ago on Wed Jun  7 06:31:30 2023.
Package httpd-2.4.56-1.amzn2023.x86_64 is already installed.
Package httpd-tools-2.4.56-1.amzn2023.x86_64 is already installed.
No match for argument: mod_ss
Error: Unable to find a match: mod_ss
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# systemctl enable http
Failed to enable unit: Unit file http.service does not exist.
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# systemctl enable httpd
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# systemctl start httpd
[root@ip-10-0-0-77 ec2-user]#
[root@ip-10-0-0-77 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Wed 2023-06-07 07:56:58 UTC; 9h ago
       Docs: man:httpd.service(8)
   Main PID: 27811 (httpd)
     Status: "Total requests: 269; Idle/Busy workers 100/0;Requests/sec: 0.0077; Bytes serve>
      Tasks: 230 (limit: 1108)
     Memory: 24.1M
        CPU: 24.555s
     CGroup: /system.slice/httpd.service
             ├─27811 /usr/sbin/httpd -DFOREGROUND
             ├─27823 /usr/sbin/httpd -DFOREGROUND
             ├─27825 /usr/sbin/httpd -DFOREGROUND
             ├─27826 /usr/sbin/httpd -DFOREGROUND
             ├─27827 /usr/sbin/httpd -DFOREGROUND
             └─30502 /usr/sbin/httpd -DFOREGROUND
```

STEP 16: INSTALLATION OF WORDPRESS

Now to need to run the following commands step by step for the installation of WordPress.

Install php 7.4

sudo Amazon-Linux-extras enable php7.4

sudo yum clean metadata

sudo yum install php php-common php-pear -y

sudo yum install php-{cgi,curl,mbstring,gd,mysqlnd,gettextjson,xml,fpm,intl,zip} -y

```
Installed:
  libxslt-1.1.34-5.amzn2023.0.2.x86_64
  nginx-filesystem-1:1.22.1-1.amzn2023.0.3.noarch
  php-pear-1:1.10.13-2.amzn2023.0.4.noarch
  php8.1-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-cli-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-common-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-fpm-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-mbstring-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-opcache-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-pdo-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-process-8.1.16-1.amzn2023.0.1.x86_64
  php8.1-xml-8.1.16-1.amzn2023.0.1.x86_64

Complete!
Last metadata expiration check: 0:00:05 ago on Wed Jun  7 17:56:54 2023.
Package php8.1-cli-8.1.16-1.amzn2023.0.1.x86_64 is already installed.
Package php8.1-common-8.1.16-1.amzn2023.0.1.x86_64 is already installed.
Package php8.1-mbstring-8.1.16-1.amzn2023.0.1.x86_64 is already installed.
No match for argument: php-gettextjson
Package php8.1-xml-8.1.16-1.amzn2023.0.1.x86_64 is already installed.
Package php8.1-fpm-8.1.16-1.amzn2023.0.1.x86_64 is already installed.
No match for argument: php-zip
Error: Unable to find a match: php-gettextjson php-zip
[root@ip-10-0-0-77 ec2-user]#
```

Install mysql 5.7

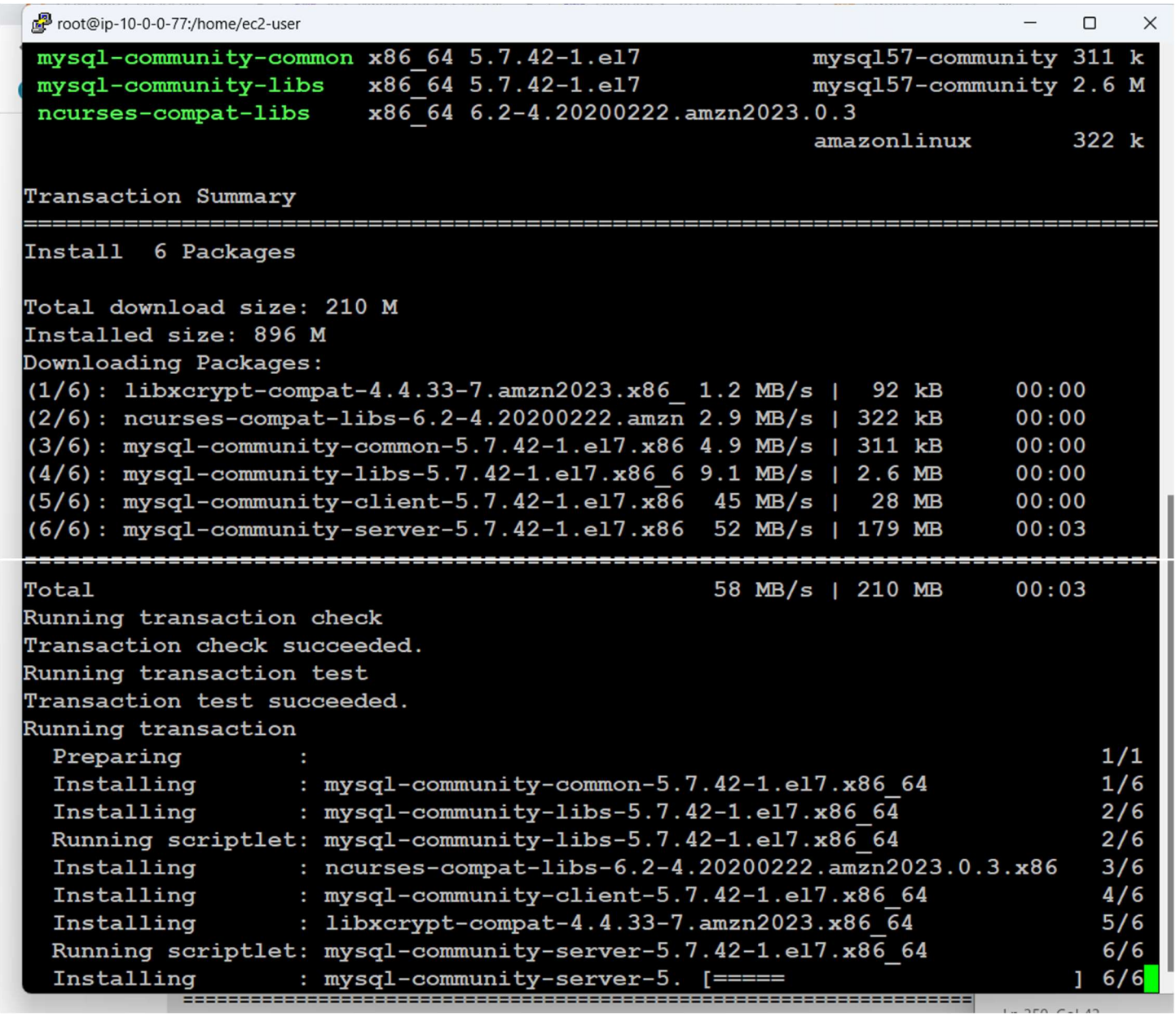
sudo rpm -Uvh https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

sudo yum install mysql-community-server -y

sudo systemctl enable mysqld

sudo systemctl start mysqld

sudo systemctl status mysqld



Set the right permissions

sudo usermod -a -G apache ec2-user

sudo chown -R ec2-user:apache /var/www

sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;

sudo find /var/www -type f -exec sudo chmod0664 {} \;

chown apache:apache -R /var/www/html

Download of wordpresss files

wget https://wordpress.org/latest.tar.gz

tar -xzf latest.tar.gz

cp -r wordpress/* /var/www/html/

Create the wp-config.php file

cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php

Edit the wp-config.php file

vi /var/www/html/wp-config.php

Navigate to your database, click on configuration to get information to edit the file

We need to configure 4 parameters:

DB_NAME

DB_USER

DB_PASSWORD

DB_HOST

Now after the configuration lets save by :wq! + Enter

Restart the Webserver

sudo systemctl restart httpd or service httpd restart

Navigate to the EC2 console

Select the EC2, select (My_Project_Server_01)

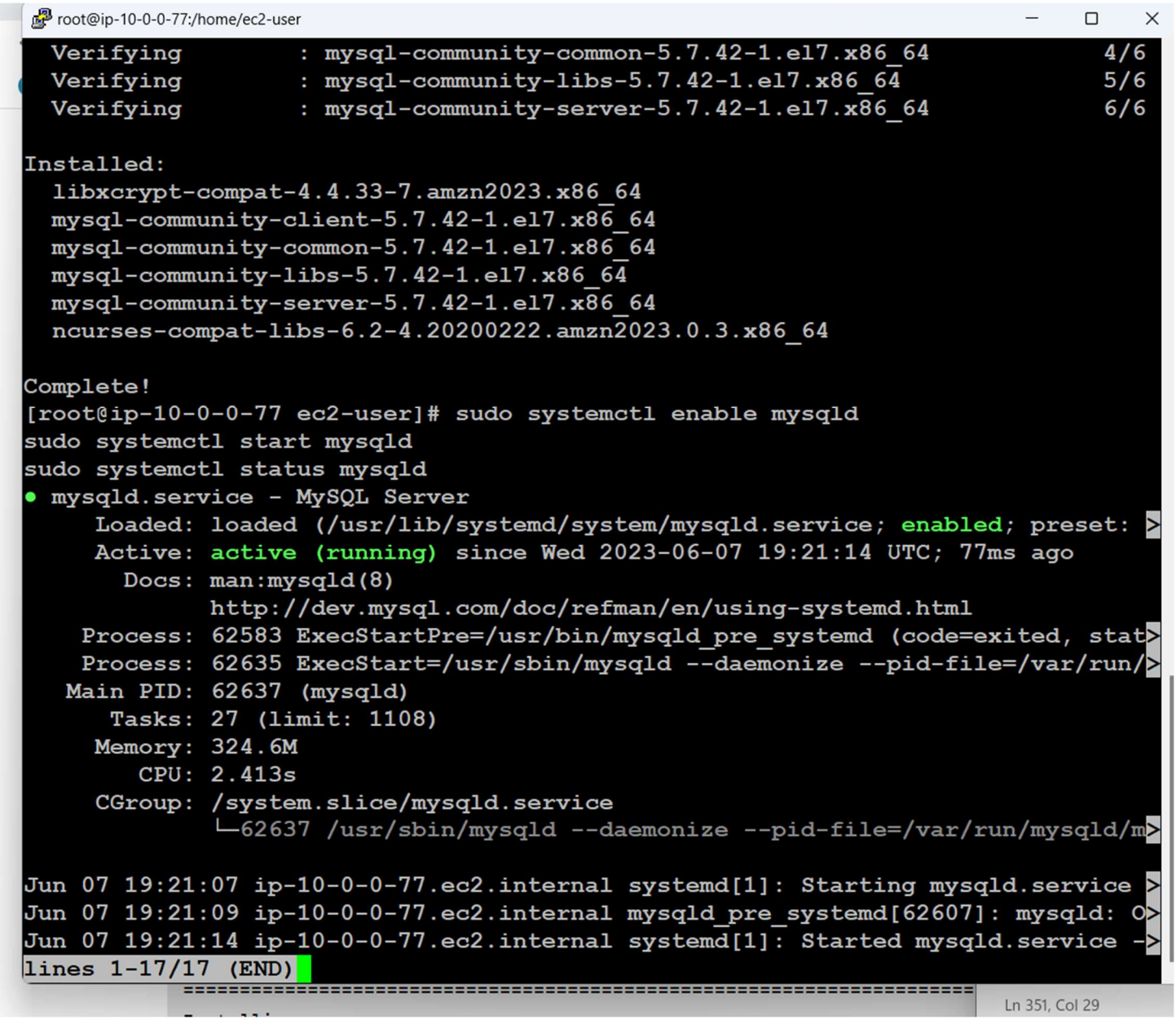
Copy the public IPV4 address

Paste the copied IP address on a new tab and booooom

WordPress User Interface is ready!

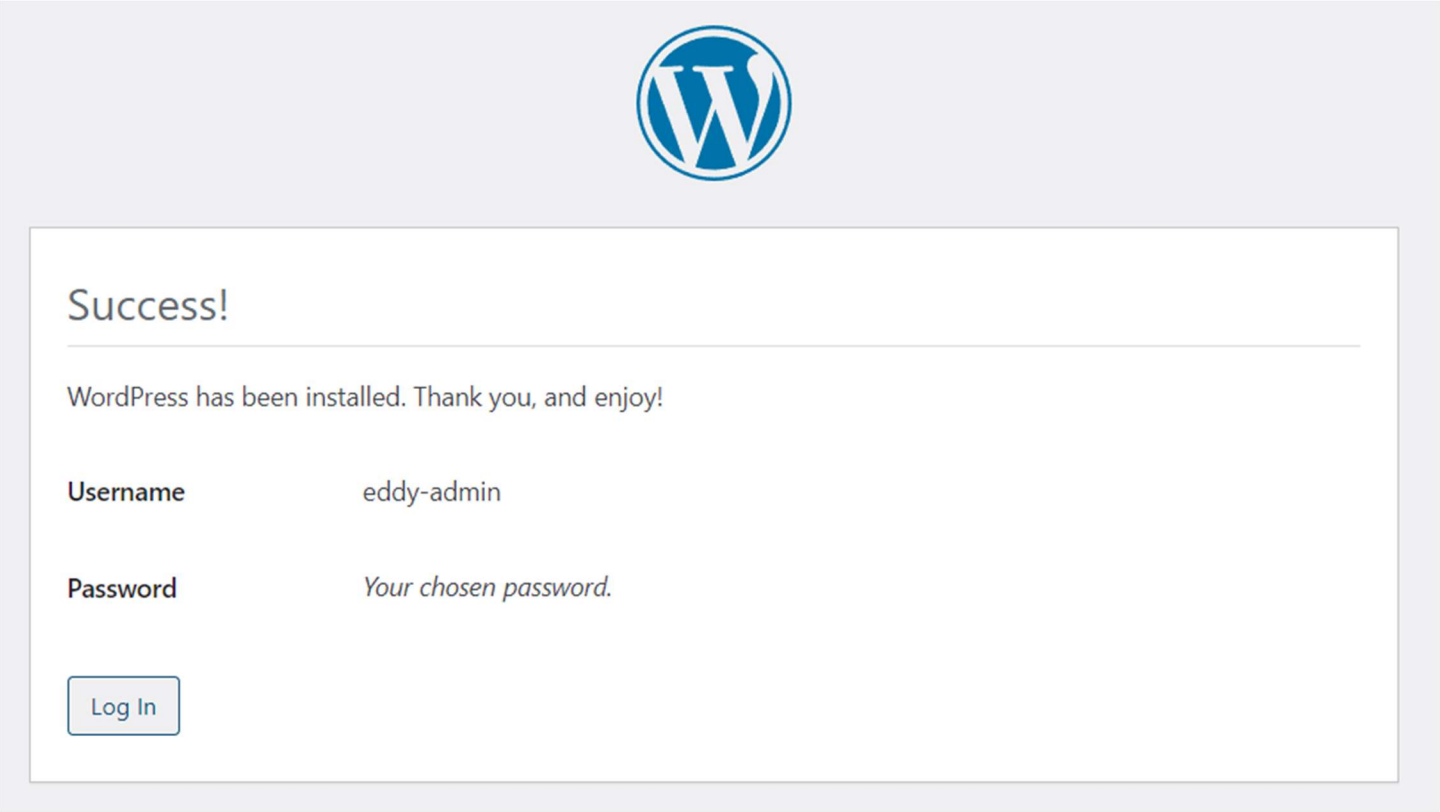
In case of any extension error message, run the following command to install extension:

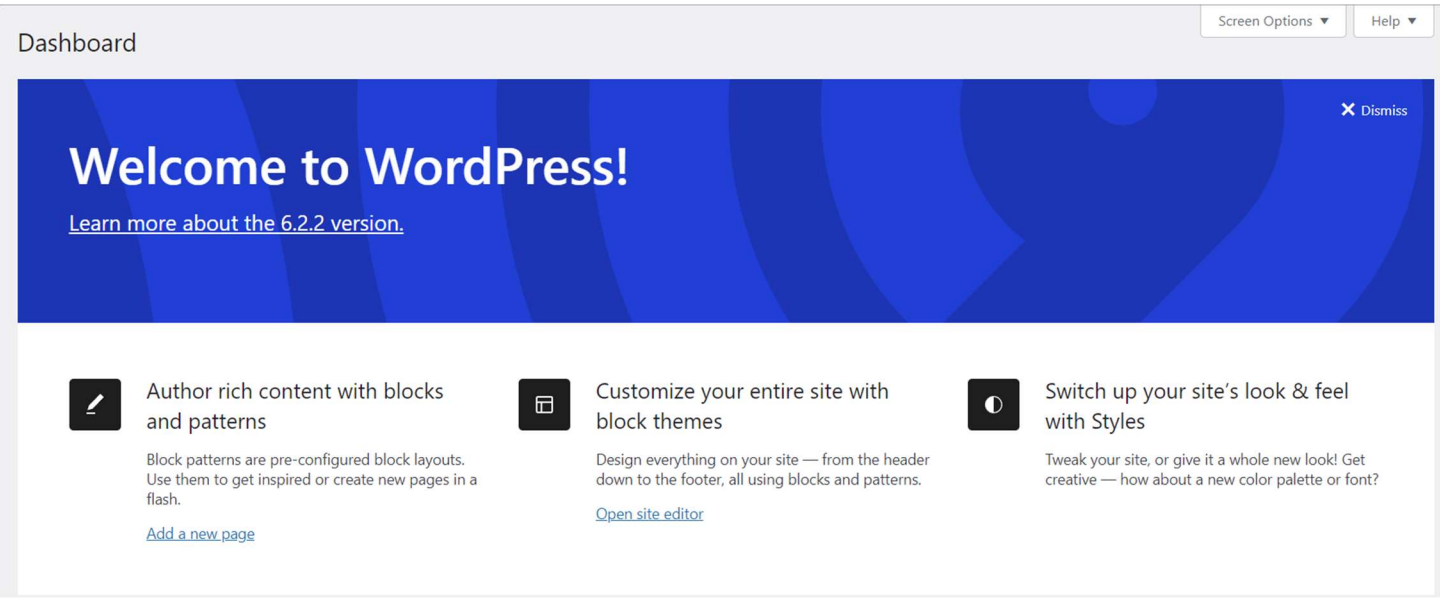
sudo yum install php-mysqli or sudo yum install php-pdo-mysql



In case of any extension error message, run the following command to install extension:

sudo yum install php-mysqli or sudo yum install php-pdo-mysql





We have successfully installed WordPress application on Amazon Linux 2 AMI, and we have moved the file to EFS.

STEP 17:

CREATION OF 2 EC2 INSTANCES FOR TARGET GROUPS

Launch EC2 instances

IMPORTANT: Here is the user data script for the installation of the EC2 instances

```
#! /bin/bash
yum update -y
sudo yum install httpd -y httpd-tools mod_ssl
sudo systemctl enable httpd
sudo systemctl start httpd
sudo systemctl status httpd
sudo amazon-linux-extras enable php7.4
sudo yum clean metadata
sudo yum install php php-common php-pear -y
sudo yum install php-{cgi,curl,mbstring,gd,mysqlnd,gettextjson,xml,fpm,intl,zip} -y
sudo rpm -Uvh https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
sudo yum install mysql-community-server -y
sudo systemctl enable mysqld
sudo systemctl start mysqld
echo "fs-01288959068f268f3.efs.us-east-1.amazonaws.com:/ /var/www/html nfs4
nfsvers=4.1,rsize=1048576 mount -a
service httpd restart
```

My two EC2 instances (EC2-For-ALB-App-Sub-1a) and (EC2-For-ALB-App-Sub-1b) were successfully created and running.

STEP 18:

CREATION OF ALB TARGET GROUPS

Go to the EC2 service.

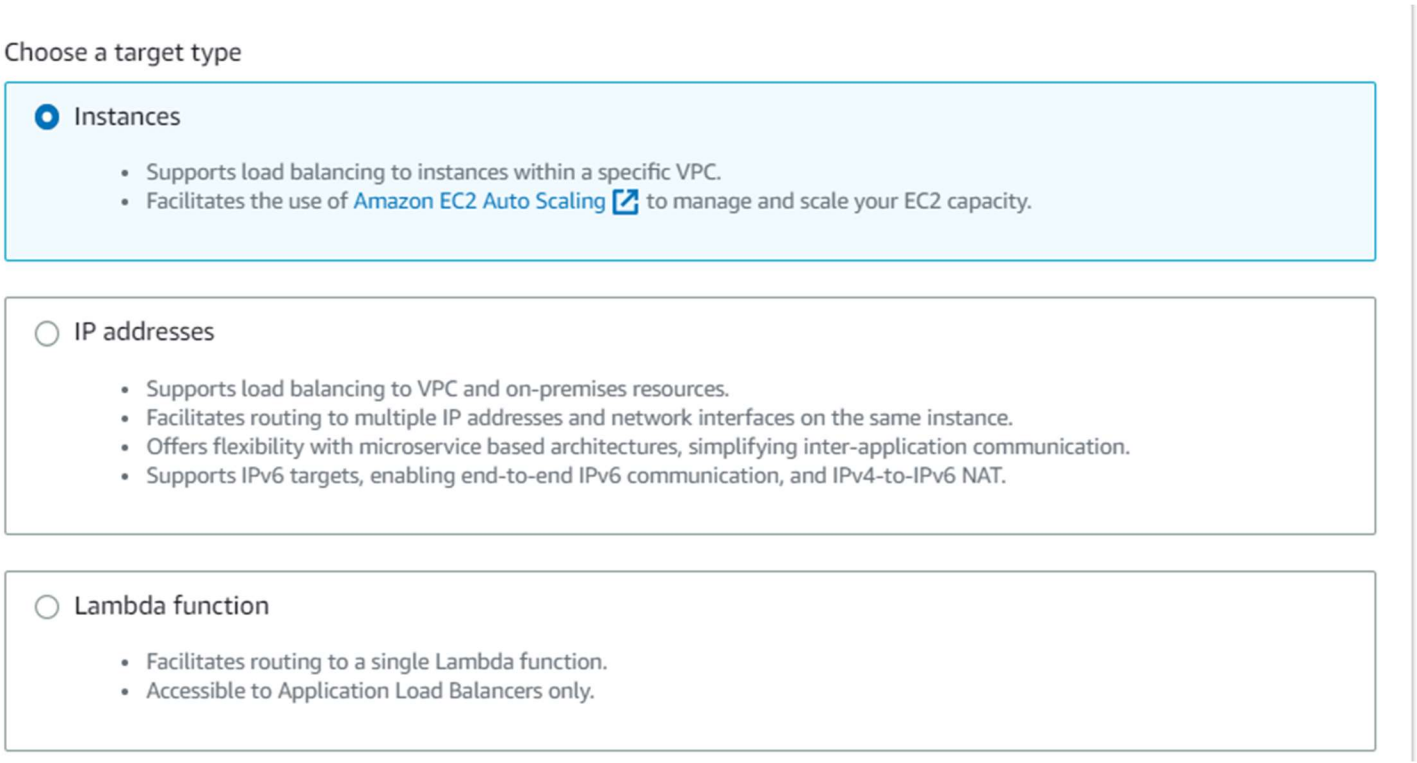
In the navigation pane, click on "Target Groups" under "Load Balancing."

Click on the "Create target group" and specify a name for the target group,

Configure the health checks for your target group. Health checks help the load balancer determine the availability of targets. Set the protocol, path, and interval for the health checks.

Specify the targets for your target group. This depends on the target type you selected earlier. If you choose "Instance," you can select one or more instances to include in the target group.

Once the target group is created, you can associate it with your Application Load Balancer to route traffic to the targets based on the specified rules.



Target group name

TargetGrp-My-Project

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

Protocol

HTTP

Port

80

1-65535

VPC

Select the VPC with the instances that you want to include in the target group.

VPC_My_Project
vpc-09e54287b0cadcace
IPv4: 10.0.0.0/16

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 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

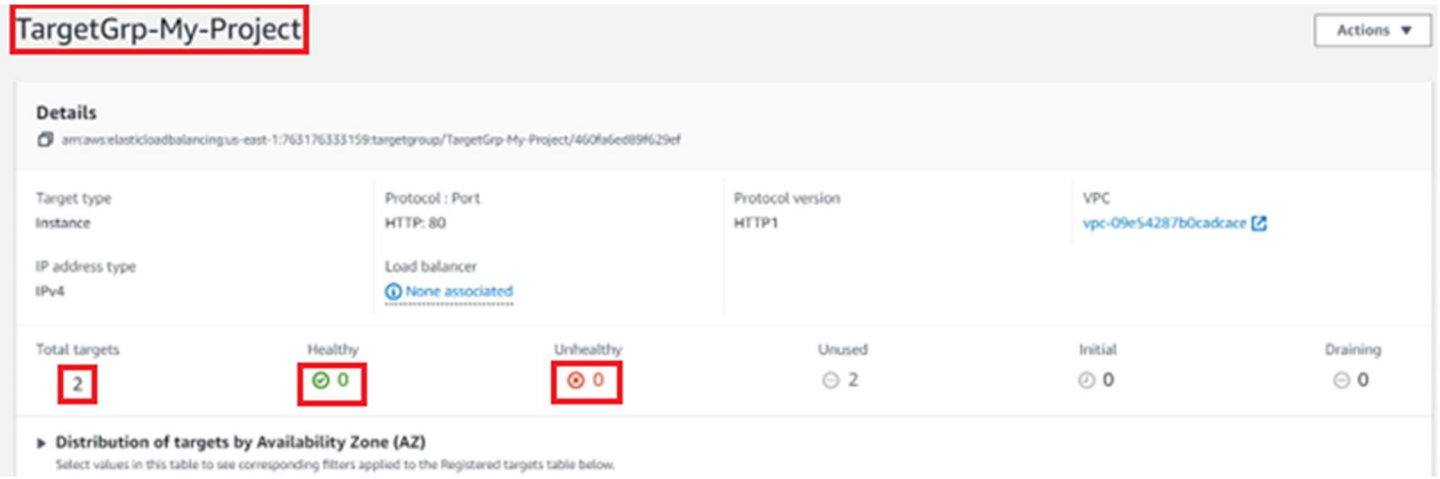
30 seconds

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



Target group successfully created and registered with the ALB.

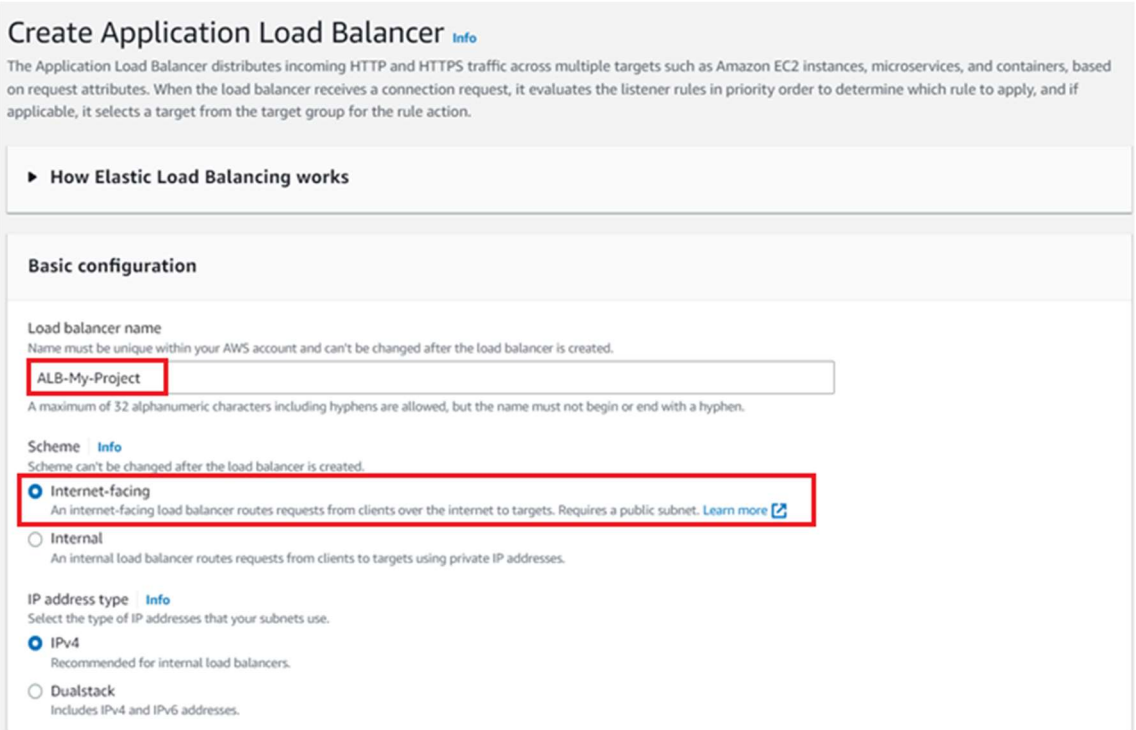
STEP 19:

CREATION OF APPLICATION LOAD BALANCER (ALB)

Now, let's proceed to create our Application Load Balancer

IMPORTANT: Very important that the Application Load balancer (ALB) should be associated with Public Subnets **(Public_Sub_Web-1a)** and **(Public_Sub_Web_1b).**

The reason for this is simple, this is because Internet traffic flows into both subnets through the internet gateway **(IGW),** the ALB being associated with the public subnets, with then be contaminated with the traffic. The ALB will then wire this traffic from both Public Subnet straight to the Private Application Subnets and to the Private Database Subnets as well.



Network mapping Info

For internet-facing load balancers, the IPv4 addresses of the nodes are assigned by AWS. For internal load balancers, the IPv4 addresses are assigned from the subnet CIDR.

VPC
vpc-09e54287b0cadcace

IP address type
IPv4

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.

us-east-1a (use1-az4)

Subnet
subnet-0552f97133c2ed30b Public_Sub_Web_1a

IPv4 address
Assigned by AWS

us-east-1b (use1-az6)

Subnet
subnet-0a00c31de3ca2a263 Public_Sub_Web_1b

IPv4 address
Assigned by AWS

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

ALB_security_group sg-0fadc2ee9bd9bd01b
VPC: vpc-09e54287b0cadcace

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: TargetGrp-My-Project HTTP
Target type: Instance, IPv4

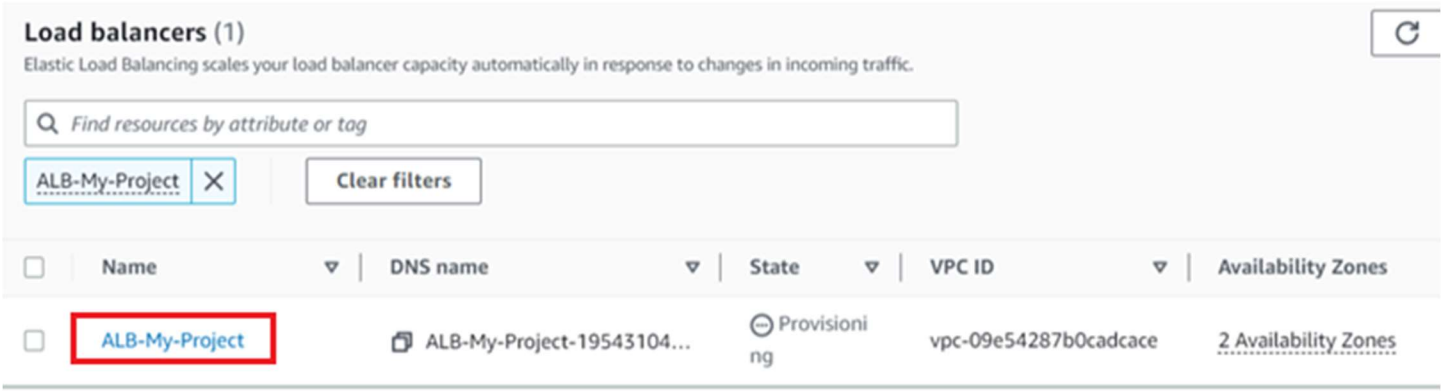
Create target group

Listener tags - *optional*

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

Add listener tag

You can add up to 50 more tags.



Load balancer successfully created.

PROJECT SUMMARY

Deployed a WordPress Website on AWS using some of the AWS core services such as VPC, Public and Private Subnets, EC2, Security Groups, Route Tables, Internet Gateway, NAT Gateway, MySQL, Elastic File System (EFS), Target groups, Application Load Balancer, Auto Scaling Group and Route 53.

Installed and configured WordPress, Apache Web Server, PHP, and MySQL on my EC2 instance and provided some necessary details, such as database name, username, and password.

Set up a domain name for my WordPress website with Route 53 and mapped my domain DNS record to the IP address of my EC2 instance.

Tested the WordPress website by accessing the domain in a web browser and was able to see the WordPress installation default page and WordPress ran perfectly.

Launch an EC2 Instance: In the AWS Management Console, navigate to EC2 (Elastic Compute Cloud) and launch a new EC2 instance. Select an appropriate Amazon Machine Image (AMI) based on your requirements. It's common to choose an AMI with a pre-installed LAMP (Linux, Apache, MySQL, PHP) stack.

For this project, we would be using our desired Amazon Linux 2 (AMI) type t2.micro, and configuring the necessary settings, as well as installing Apache, MySQL and PHP based on the WordPress requirements.