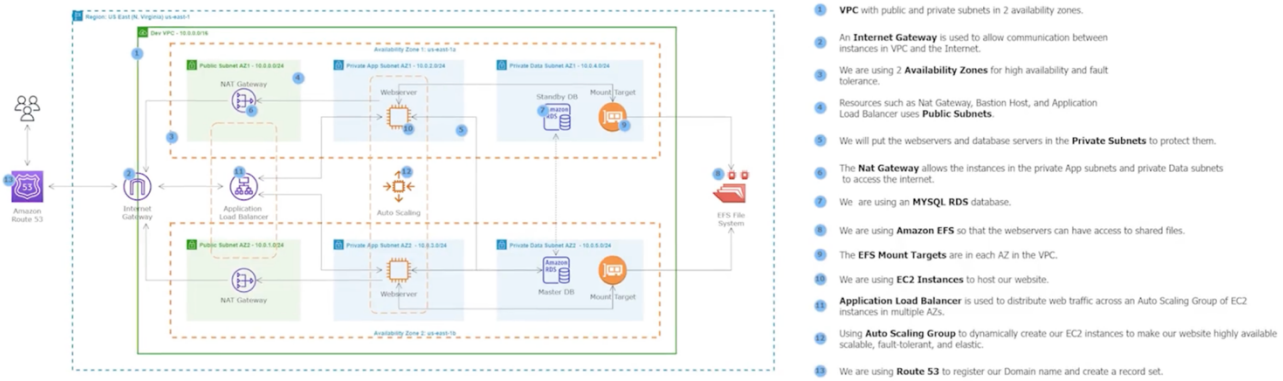
Deployed a dynamic WordPress website on AWS.

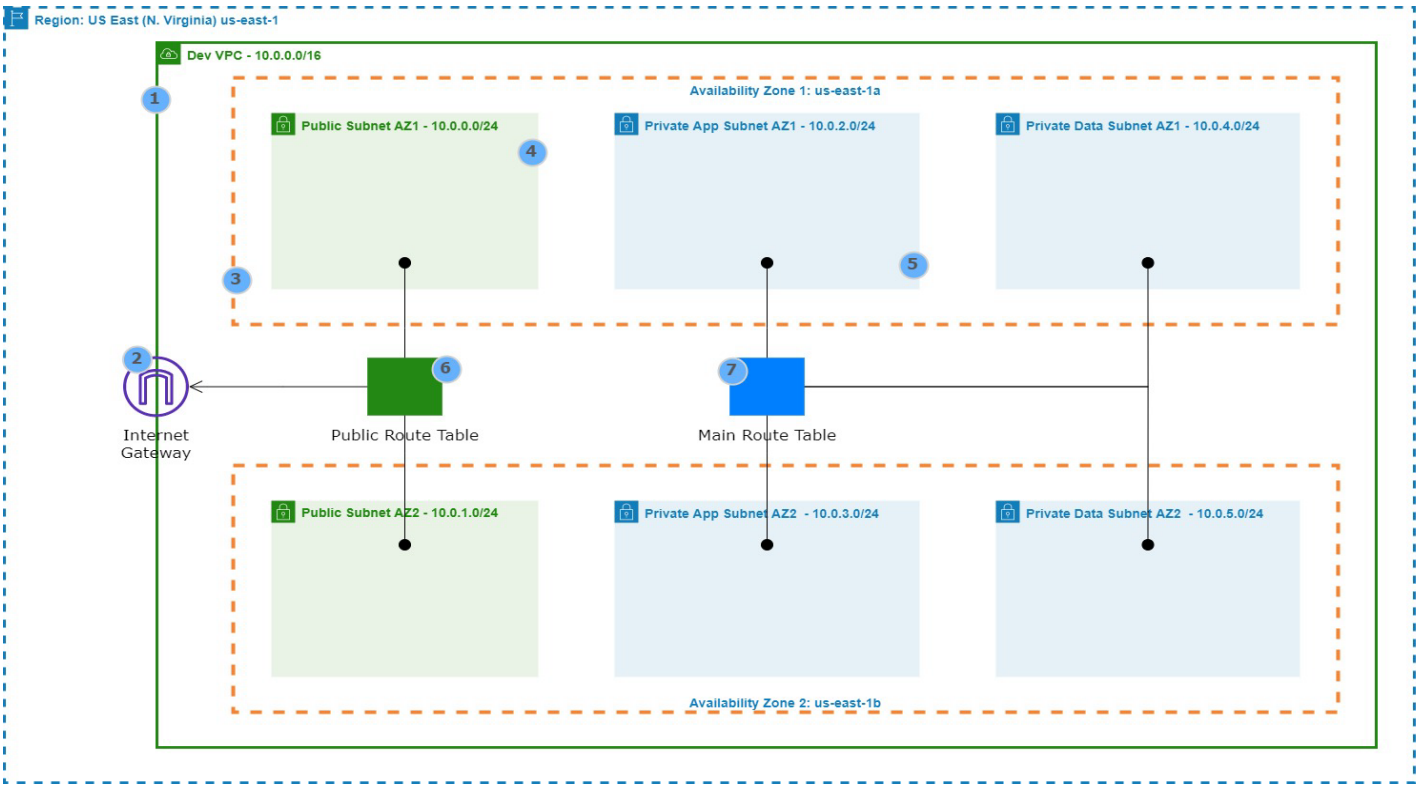


1. **VPC** with public and private subnets in 2 availability zones.
2. An **Internet Gateway** is used to allow communication between instances in VPC and the Internet.
3. We are using 2 **Availability Zones** for high availability and fault tolerance.
4. Resources such as Nat Gateway, Bastion Host, and Application Load Balancer uses **Public Subnets**.
5. We will put the webservers and database servers in the **Private Subnets** to protect them.
6. The **Nat Gateway** allows the instances in the private App subnets and private Data subnets to access the internet.
7. We are using an **MYSQL RDS** database.
8. We are using **Amazon EFS** so that the webservers can have access to shared files.
9. The **EFS Mount Targets** are in each AZ in the VPC.
10. We are using **EC2 Instances** to host our website.
11. **Application Load Balancer** is used to distribute web traffic across an Auto Scaling Group of EC2 instances in multiple AZs.
12. Using **Auto Scaling Group** to dynamically create our EC2 instances to make our website highly available scalable, fault-tolerant, and elastic.
13. We are using **Route 53** to register our Domain name and create a record set.

Services Used

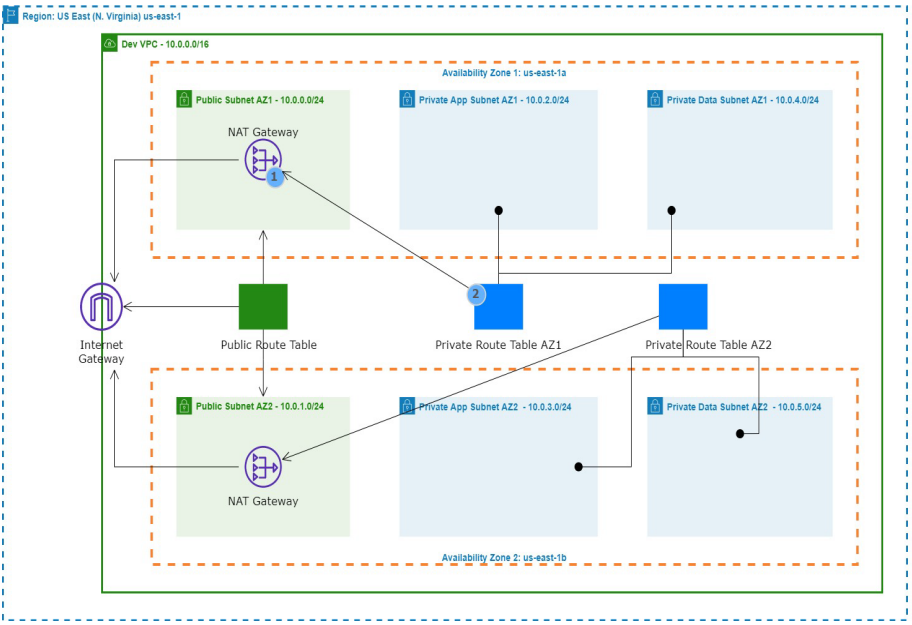
- ☐ VPC
- ☐ Security Groups
- ☐ EC2
- ☐ NAT Gateways
- ☐ RDS
- ☐ (ALB)
- ☐ Route 53
- ☐ Auto Scaling groups
- ☐ Certificate Manager
- ☐ EFS

1. BUILD A THREE-TIER AWS NETWORK VPC



Build a three-tier AWS network VPC as described in the reference, entails creating distinct layers of networking, such as public, private, and database subnets.

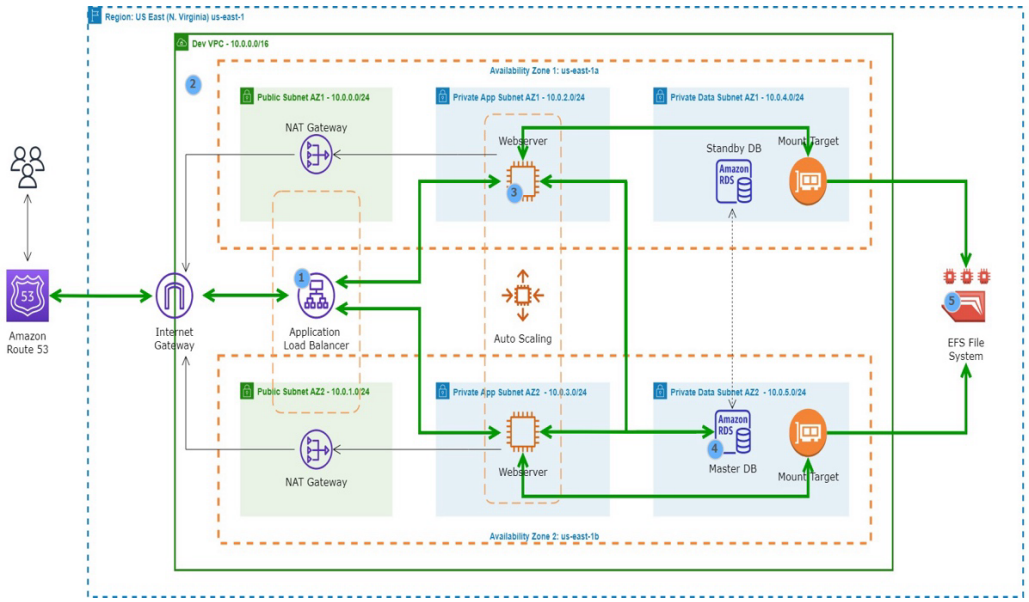
2. CREATE NAT GATEWAYS



1. The **Nat Gateway** allows the instances in the private App subnets and private Data subnets to access the internet.
2. The **Private Route Table** is associated with the private subnets and routes traffic to the internet through the nat gateway.

To establish NAT gateways, it is essential to assign Elastic IP (EIP) addresses to each of the NAT gateways to ensure they have static, publicly routable IP addresses, enabling seamless communication between private instances in a Virtual Private Cloud (VPC) and the external internet resources while maintaining a reliable and consistent network configuration.

3. CRETATE THE SECURITY GROUPS



1 **ALB Security Group**
Port = 80 and 443 **Source** = 0.0.0.0/0

2 **SSH Security Group**
Port = 22 **Source** = Your IP Address

3 **Webserver Security Group**
Port = 80 and 443 **Source** = ALB Security Group
Port = 22 **Source** = SSH Security Group

4 **Database Security Group**
Port = 3306 **Source** = Webserver Security Group

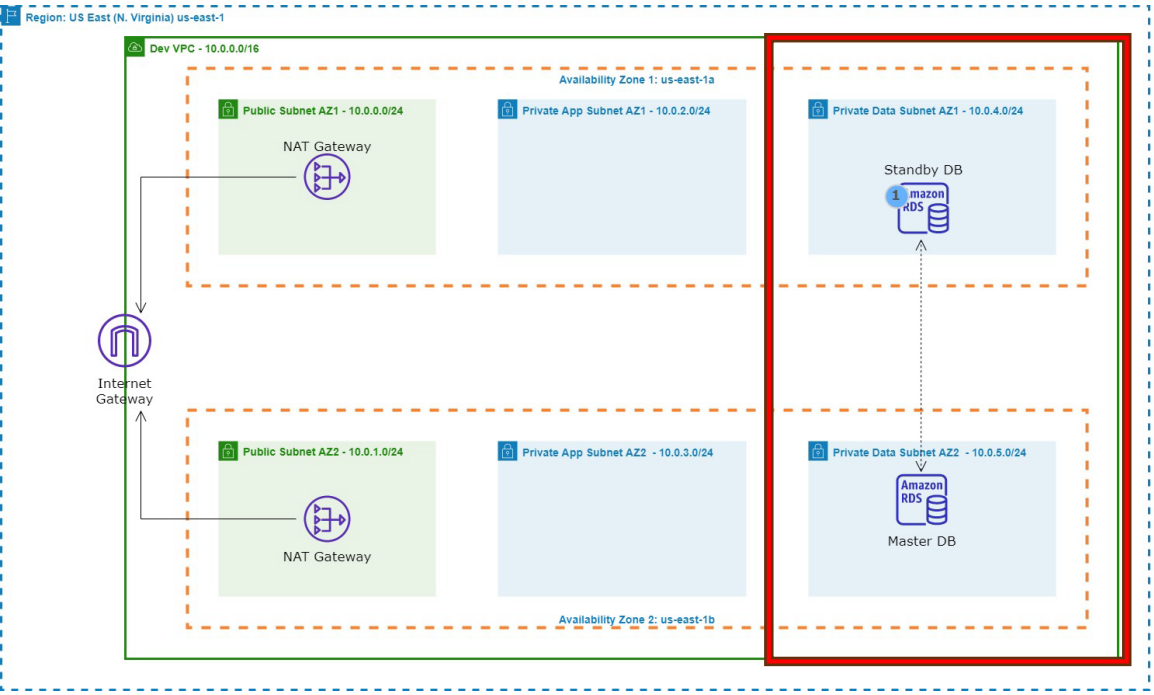
5 **EFS Security Group**
Port = 2049 **Source** = Webserver Security Group, EFS Security Group
Port = 22 **Source** = SSH Security Group

Key Notes/Security group port names

- **Port 80 – HTTP**
- **Port 443 – HTTPS**
- **Port 22 – SSH**
- **Port 3306 – MySQL/Aurora**
- **Port 2049 – NFS**

To create a security group, specify inbound and outbound rules for required ports, such as port 80 for HTTP, port 443 for HTTPS, port 22 for SSH, port 3306 for MySQL/Aurora, and port 2049 for NFS, to control the network traffic to and from your AWS resources effectively.

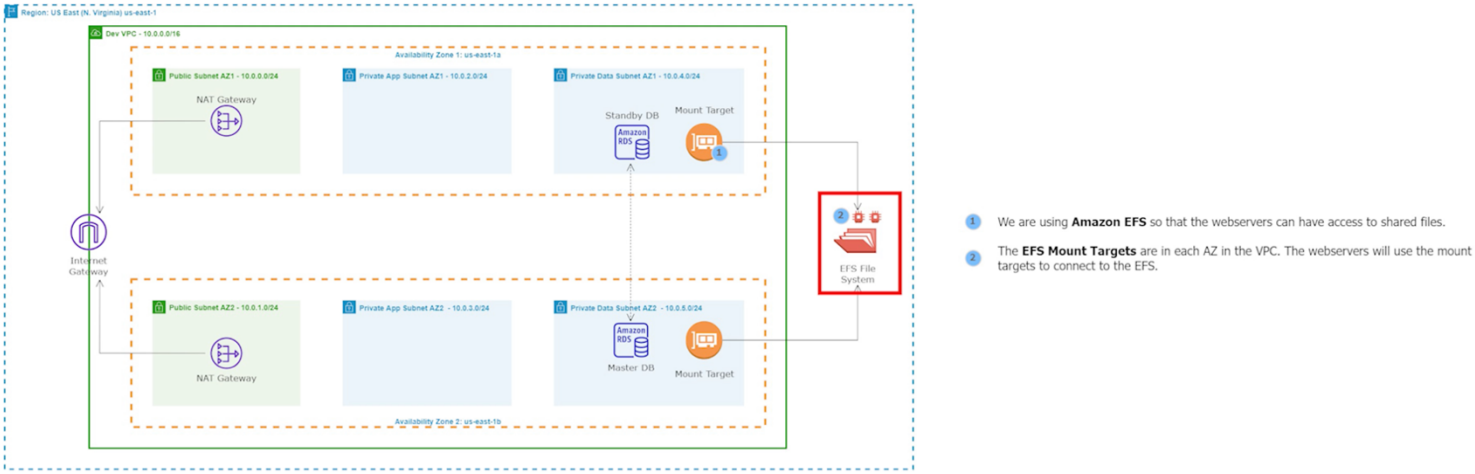
4. CREATE THE RDS INSTANCE



For the database, start by selecting Engine Version 5.7 and opting for the Dev/Test template. When it comes to Availability and durability, you have a choice: you can either go for a 'multi-AZ DB instance,' which creates a standby database in a different Availability Zone (AZ), though it's chargeable for this project, or you can select 'Single DB Instance,' which creates a standalone database without a standby instance, and it won't incur charges. Moving on to settings, create a unique DB username and password, ensuring to store them separately for security.

For the DB Instance Class, choose 'Burstable Classes' with 'include previous generation classes' toggled on and select a free 't2.micro' instance. In terms of connectivity, choose our desired VPC, ensure that the previously created 'subnet group' is selected, opt for the 'Database security Group' in the security group section, and set our master database location to 'us-east-1b.' Lastly, in Additional Configuration, give our database a name to ensure that Amazon RDS successfully creates it, and then proceed to create the database.

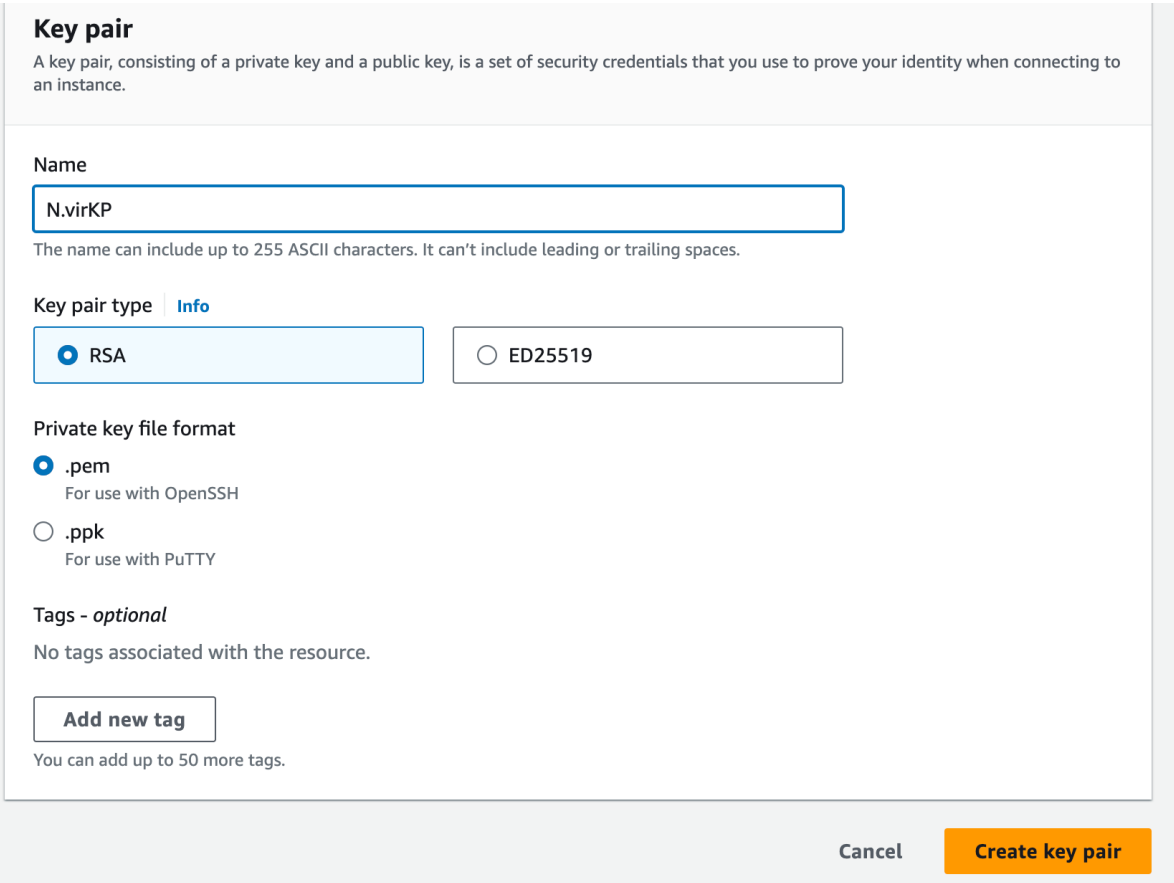
5. CREATE ELASTIC FILE SYSTEM (EFS)



To set up an Amazon Elastic File System (EFS), start by clicking "Create File System," choose the "Customize" option with default settings while disabling encryption to avoid additional charges, proceed to configure tags by providing a tag and value, then, under Network Settings, select your VPC, the tier 3 subnet, and the designated 'EFS security group,' before finally clicking "Create" to establish the EFS resource.

6. CREATE A KEY-PAIR TO SSH INTO THE EC2 INSTANCE |

I'm a Mac user so I gonna use this setting for my key-pair.

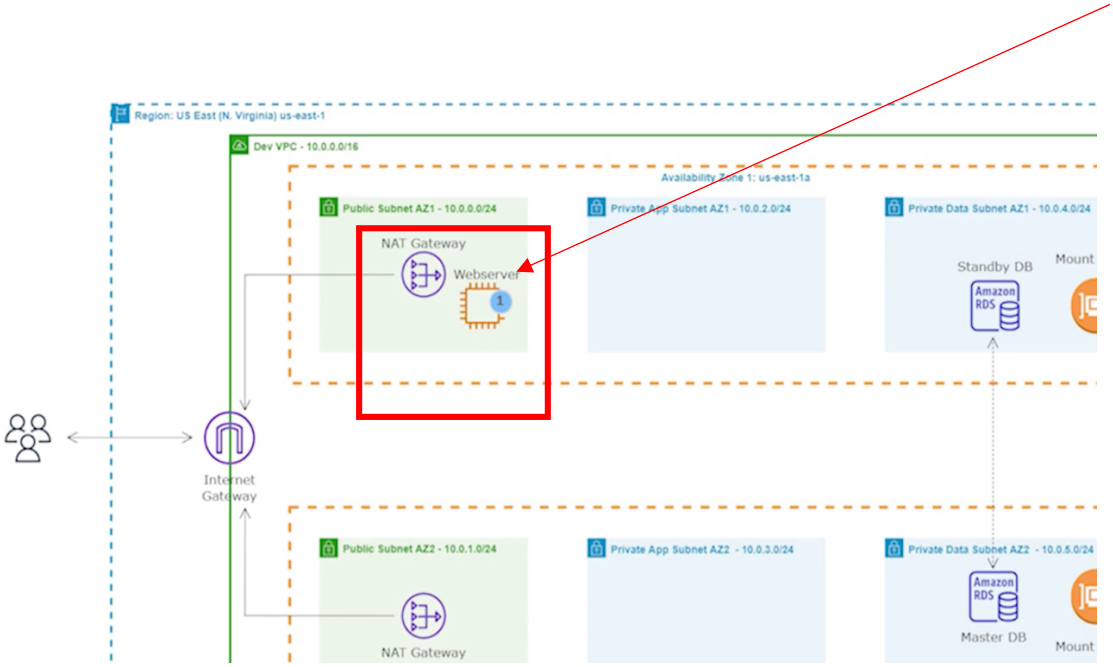


To SSH into an EC2 instance located in the Public Subnet, begin by creating your key pairs and ensuring you choose the correct private key file format, then download the key pair to your home directory if you're a Mac user. Open Terminal and set the key pair permissions using **'chmod 400 NameOfYourKP.pem'**. If you're using Windows, utilize Putty for SSH. To initiate the SSH connection, copy the Public IP address of the EC2 instance, open your Terminal, and type **'ssh -I <NameOftheKP.pem> ec2user@<publicIPAddress>'**, followed by pressing Enter. **Type 'yes'.**

'chmod 400 NameOfYourKP.pem'

'ssh -I <NameOftheKP.pem> ec2user@<publicIPAddress>'

7. LAUNCH A SETUP SERVER/ PUBLIC SUBNET AZ1



To launch an EC2 instance in the 'Public Subnet AZ1' for the purpose of setting up a WordPress website, name the EC2 'Setup Server,' choose the 'Amazon Linux 2 AMI' as the AMI, select 't2.micro' as the instance type, pick your key pair (KP), you previously created for 'your Network,' specifically selecting 'Public Subnet AZ1' and ensure that 'public IP is enabled.' Additionally, assign the 'SSH,' 'ALB,' and 'Web-Server SG' security groups, and finally, proceed to launch the instance.

8. COMMANDS TO INSTALL WORDPRESS SITE INTO THE SETUP SERVER

Commands were gonna run to install the WordPress site.

Keynotes

- Before executing any commands, it's crucial to ensure that you've mounted your EFS DNS name, to your EC2 instance. This step is essential for establishing the necessary file system connection between your EC2 instance and the Elastic File System (EFS) you previously created.

- Mine was this - fs-03975a6ee0e99e4bf.efs.us-east-1.amazonaws.com

#1. create the html directory and mount the EFS to it.

sudo su

yum update -y

mkdir -p /var/www/html

sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-03975a6ee0e99e4bf.efs.us-east-1.amazonaws.com:/ /var/www/html

check whether you EFS is Mounted – **df -h**

#2. install apache

sudo yum install -y httpd httpd-tools mod_ssl

sudo systemctl enable httpd

sudo systemctl start httpd

#3. 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,gettext,json,xml,fpm,intl,zip} -y

#4. install mysql5.7

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

sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

sudo yum install mysql-community-server -y

sudo systemctl enable mysqld

sudo systemctl start mysqld

#5. set 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 chmod 0664 {} \;

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

#6. download wordpress and moves the files to the WordPress Directories.

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

tar -xzf latest.tar.gz

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

#7. create the wp-config.php file

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

#8**. edit the wp-config.php file (Put your RDS instance information. You can find all this information in the (Configuration and Connectivity & Security section of your DB Instance)**

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

#9. restart the webserver

service httpd restart

#10 Copy the Public IP address of the 'Start-up Web-Server' paste it in the URL.

Create your WordPress account.

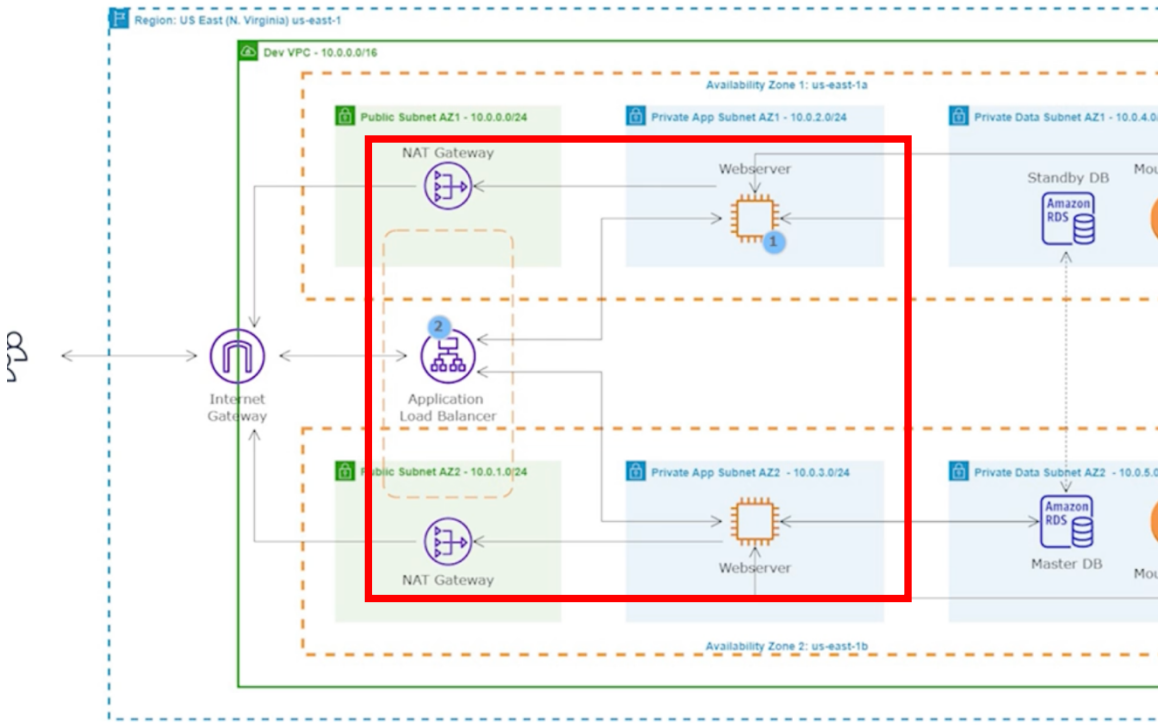
Before

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );

/** Database username */
define( 'DB_USER', 'username_here' );

/** Database password */
define( 'DB_PASSWORD', 'password_here' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justi
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Spe
```

After (Yours should look like this)

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'ApplicationDB' );

/** Database username */
define( 'DB_USER', 'vijith' );

/** Database password */
define( 'DB_PASSWORD', 'vijith2002!' );

/** Database hostname */
define( 'DB_HOST', 'dev-rds-db.cvqsdk05cjd9.us-east-1.rds.amazonaws.com' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
                                [ Read 96 lines (Converted from DOS
```

9. CREATE AN APPLICATION LOAD BALANCER



To set up the network configuration, start by selecting the VPC and choosing the 'webserver security group' as the security group. Next, create a Target Group (TG) and add the 'webserver' EC2 instances to it to enable the Application Load Balancer (ALB) to route traffic. Create an 'Application Load Balancer,' ensuring it's mapped to both Availability Zones (AZs) and selecting one public subnet from each AZ. Choose the ALB security group, set the listener to the previously created target group. Once you've completed these steps, you can use the DNS name of the ALB to access your WordPress website.

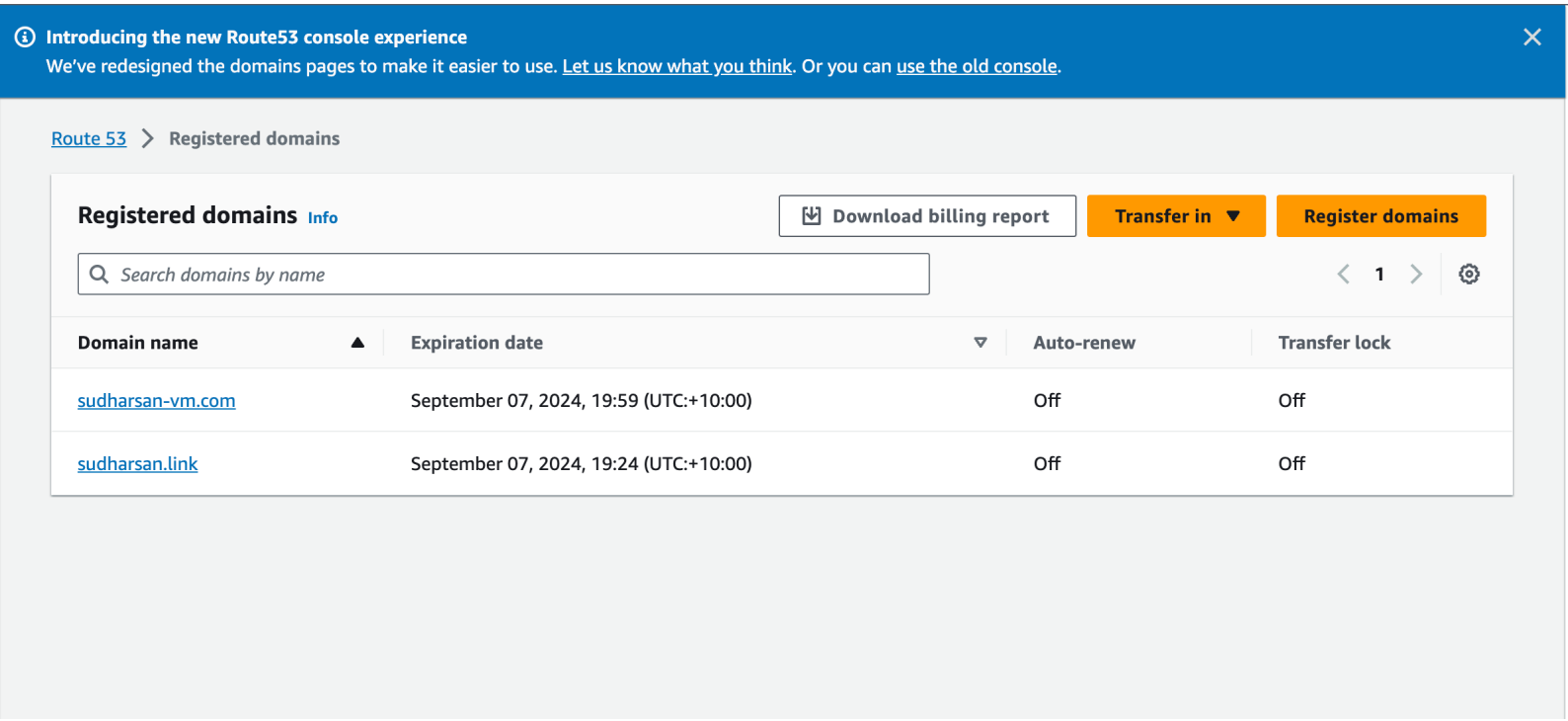
Now that you've have launched two EC2 instances in the private app subnets and can access the websites via the ALB DNS.

Remember to update your WordPress settings if you ever change your domain address. we can safely terminate our 'Setup Server' instance.

Paste these commands in the user data when you create your web server in Private App Subnet AZ1/2 and make sure to attach your EFS DNS name in the highlighted section.

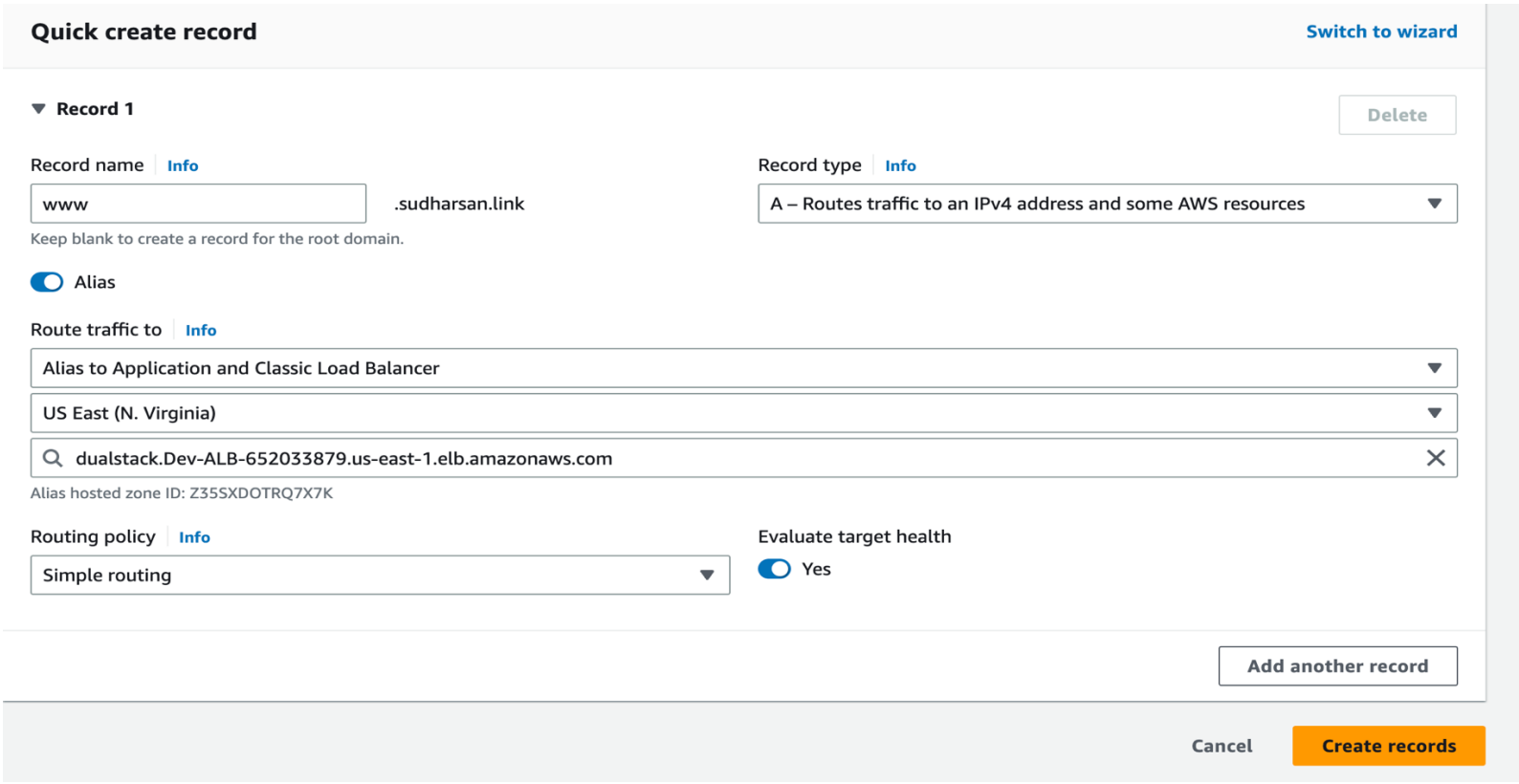
```
#!/bin/bash

yum update -y

sudo yum install -y httpd httpd-tools mod_ssl

sudo systemctl enable httpd

sudo systemctl start 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,gettext,json,xml,fpm,intl,zip} -y

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

sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

sudo yum install mysql-community-server -y

sudo systemctl enable mysqld

sudo systemctl start mysqld

echo "fs-03975a6ee0e99e4bf.efs.us-east-1.amazonaws.com:/ /var/www/html nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0" >> /etc/fstab

mount -a

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

sudo service httpd restart
```

10. REGISTER A DOMAIN & CREATE A RECORD SET IN ROUTE 53



To register a domain on Route 53, navigate to the 'Registered Domains' section and proceed with the domain registration process. Based on my personal experience, it's advisable to choose domain extensions like .com, .net, or .org, as they typically become available more quickly compared to other extensions.

This recommendation stems from the fact that some domain extensions may involve a longer waiting period before they are ready for use, so opting for one of the mentioned extensions can expedite the domain registration process and get your website up and running more swiftly.

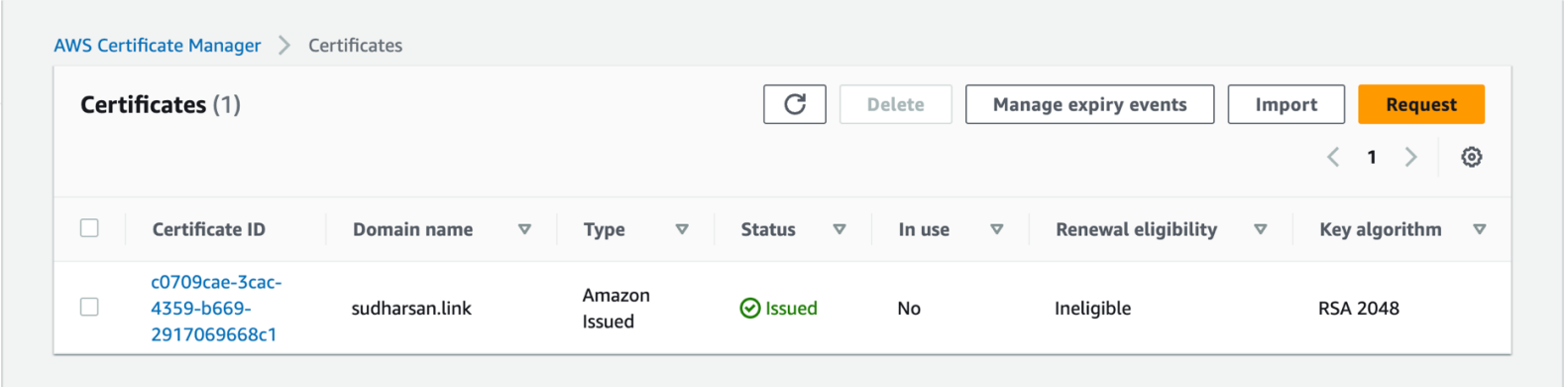


To set up the website's domain with Route 53, create a record set by specifying the 'record name' and configuring it to point to the ALB (Application Load Balancer) using an alias. This setup ensures that when users access your domain name, they are directed to your website hosted on the ALB.

However, it's crucial to remember that if you ever change your domain name, you must also update the domain configuration within your WordPress settings to reflect the change.

This synchronization between your Route 53 record set and your WordPress settings is essential to ensure a seamless and uninterrupted online experience for your visitors when accessing your website via its domain name.

11. CREATE SSL CERTIFICATE USING CERTIFICATE MANAGER

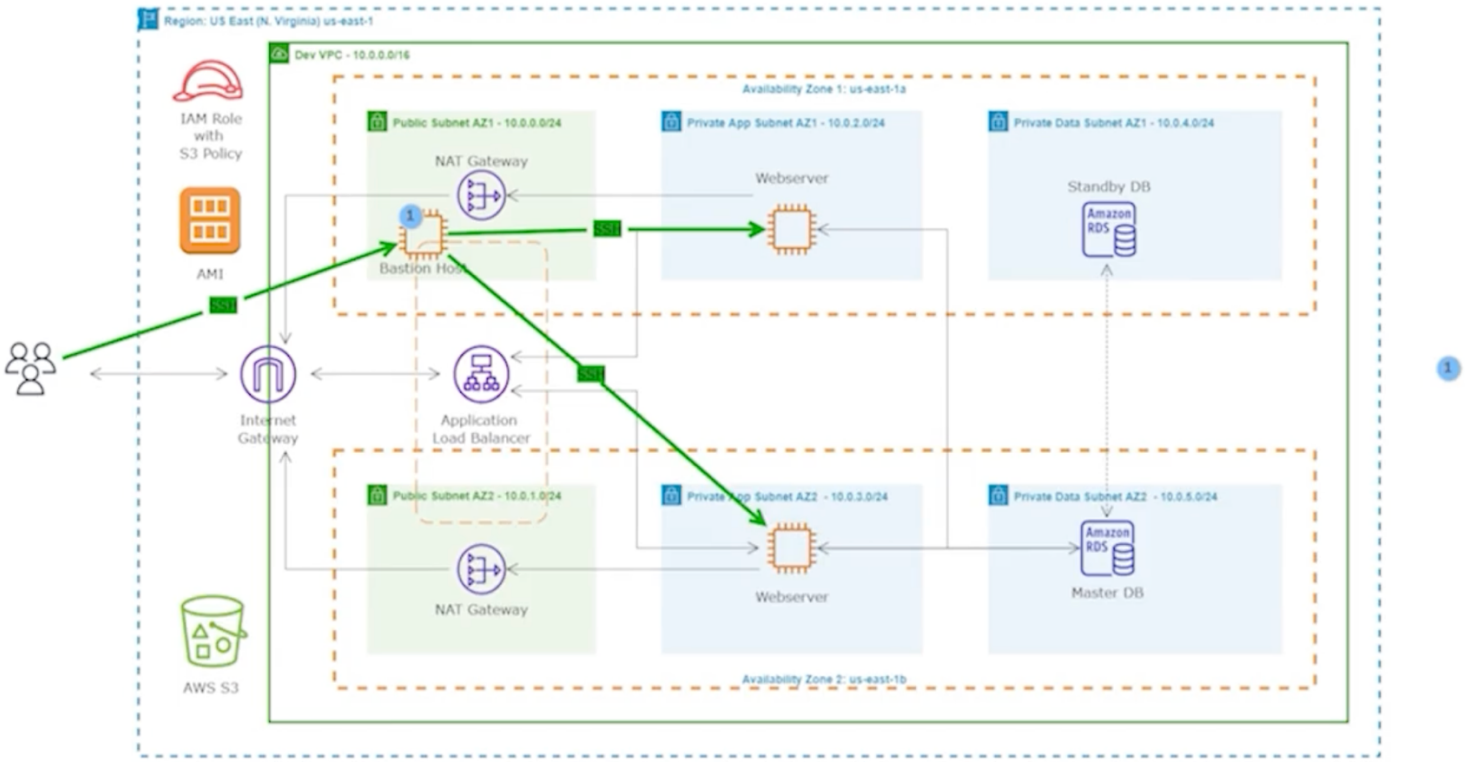


To secure the website, I utilized the Certificate Manager to generate an SSL certificate, and to prove ownership of the domain, I established a record set for validation.

After successfully completing this verification process, return to the Application Load Balancer (ALB) configuration. Within the ALB settings, add an HTTPS listener to ensure secure communication.

Additionally, configure the HTTP listener to automatically redirect incoming HTTP requests to the newly created HTTPS listener. This comprehensive setup guarantees that the website benefits from SSL encryption, enhancing security and providing a secure browsing experience for visitors.

12. LAUNCH A BASTION HOST



To enhance the security of the webserver and facilitate SSL certificate configuration, a bastion host is set up in Public Subnet AZ1. This bastion host serves as a secure gateway to establish connections with the webserver situated in Private App Subnet AZ1/2.

By utilizing the bastion host, one can gain access to and modify the 'WordPress Config' file, allowing for the updating of the domain name on the WordPress site and ensuring the website's security through SSL certificate encryption.

Commands for Mac users

Commands to SSH into an EC2 instance in the private subnet on a Mac computer

These commands are for Amazon Linux instance.

Command 1:

ssh-add --apple-use-keychain <the-name-of-your-private-key.pem>

Example:

ssh-add --apple-use-keychain myec2key.pem

Command 2: SSH into the Bastion host

ssh -A ec2-user@<the-public-ipv4-ip-of-your-bastion-host>

Example:

ssh -A ec2-user@54.162.137.241

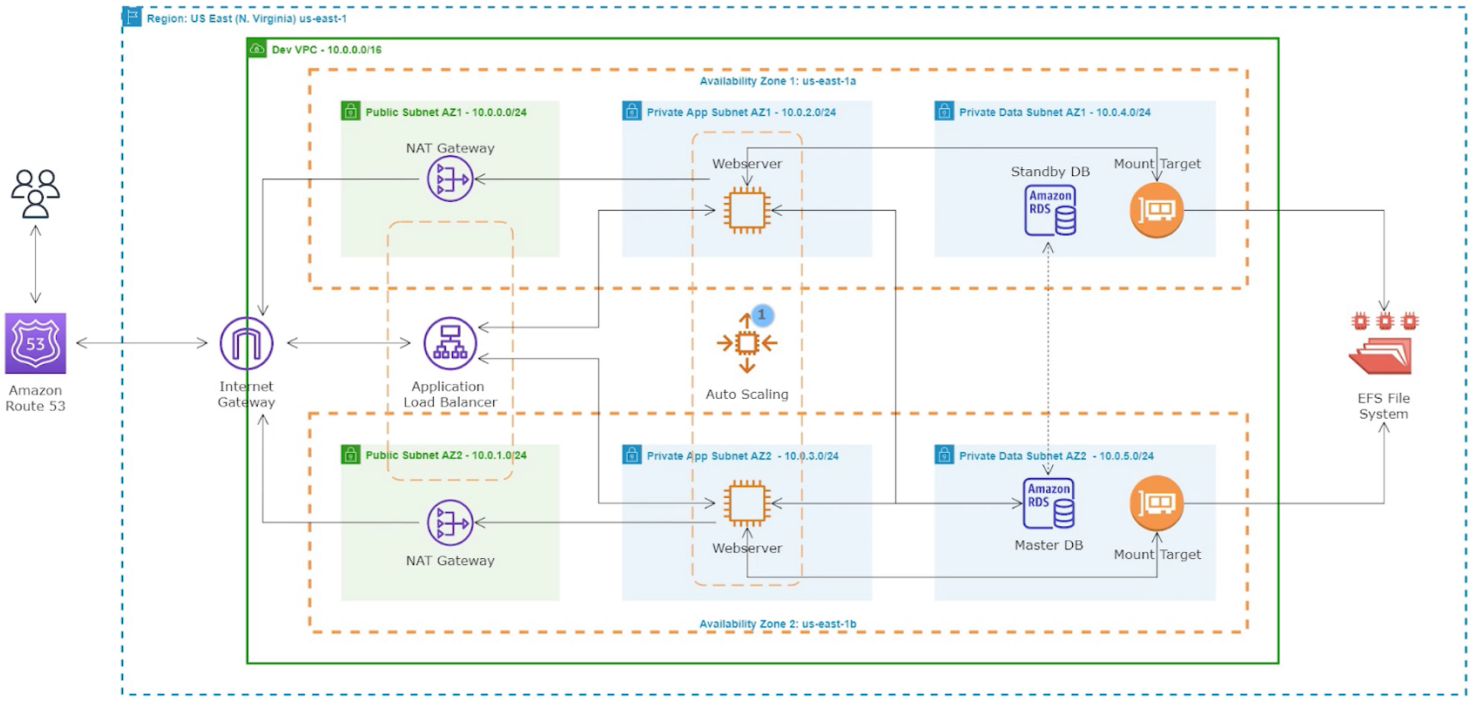
Command 3: SSH into the web server in private subnet

ssh ec2-user@<the-private-ipv4-ip-of-the-instance-in-the-private-subnet>

Example:

ssh ec2-user@18.232.135.220

13. CREATE AN AUTO SCALING GROUP & LAUNCH TEMPLATES



In this step, an Auto Scaling Group (ASG) is established, along with the implementation of launch templates to facilitate the dynamic scaling of webserver resources situated within the Private App Subnet. A specialized Launch Template that encompasses the precise configurations for the EC2 instances is employed, enabling the ASG to seamlessly initiate new instances whenever the need arises.

As a result, the previously manually deployed webserver instances can be safely terminated. To ensure the new instances are correctly configured, the same set of commands employed during the initial webserver creation are incorporated into the user data section when crafting the launch templates. Subsequently, the ASG is created using this Launch Template, streamlining, and automating the scaling process for optimal efficiency.

After the completion of the preceding steps, the final phase entails making essential edits to the website template to ensure it aligns with the updated configurations and settings. Once these adjustments are completed, the website is fully prepared and secured with SSL certification, ensuring a seamless and secure online experience for the visitors.