**Deployed a dynamic WordPress website on AWS.**

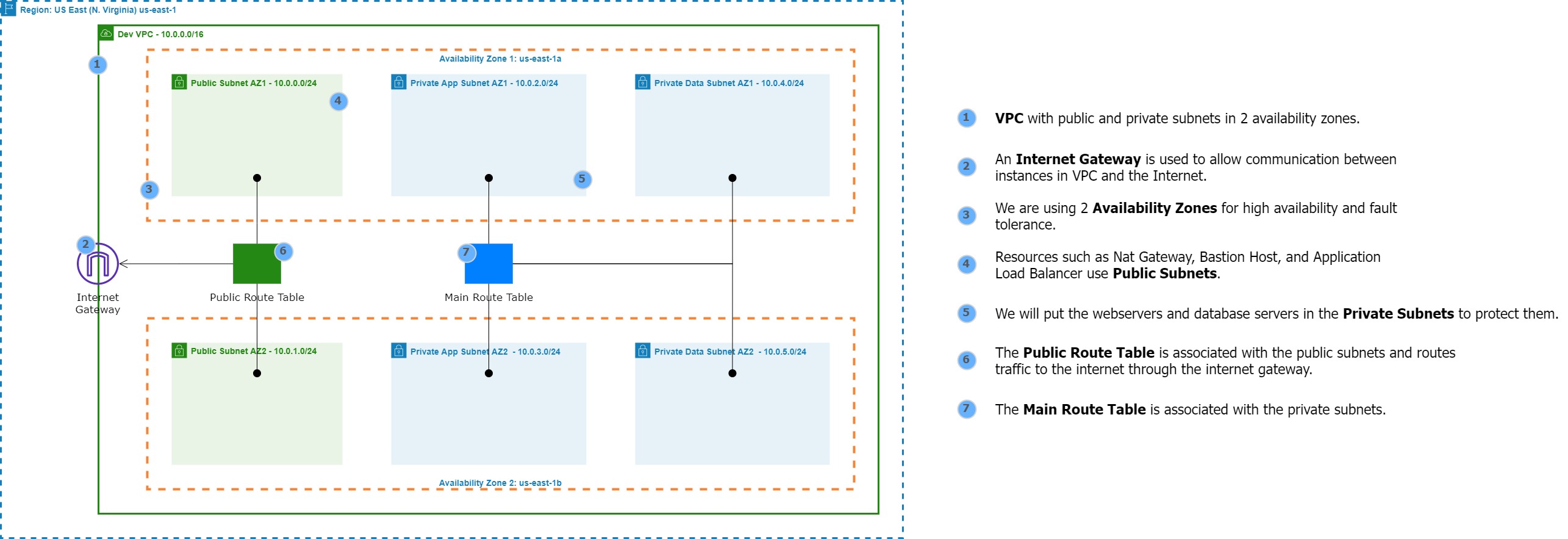
A diagram of a computer network

Description automatically generated with medium confidence

**Services Used**

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

## Build a Three-Tier AWS Network VPC



**Tier 1 or Public-subnet-AZ1/2**

- Nat -Gateway

- ALB

- Bastion Host

**Tier 2 or Private-App-Subnet AZ1/2**

-Web servers (EC2)

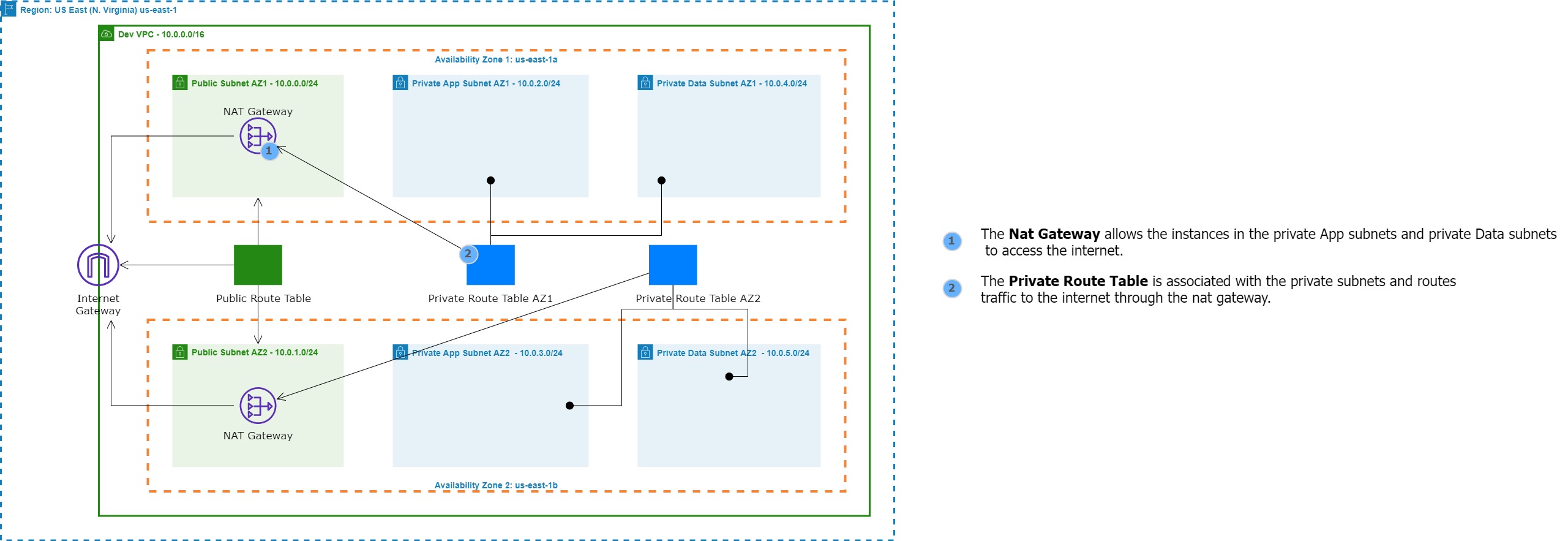
**Tier 3 0r Private-Data-Subnet AZ1/2**

-Database

**Key Notes**

-Enable DNS Hostnames

## Create Nat gateways



**Key Notes**

* **Make sure to assign Elastic IP (EIP) to the Nat Gateways.**

## Cretate the Security Groups

A diagram of a computer

Description automatically generated

**Key Notes/Security group port names**

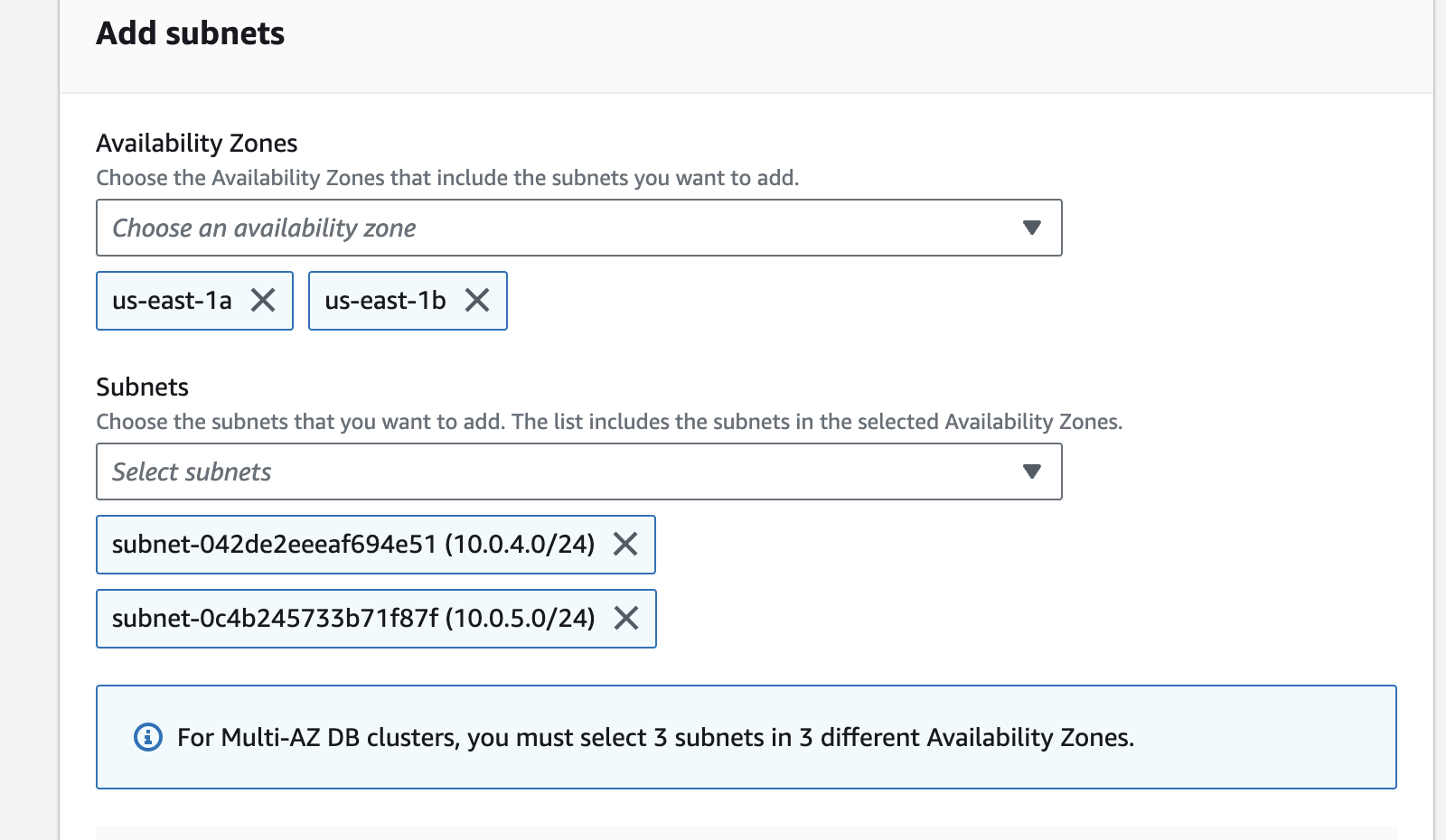
* **Port 80 – HTTP**
* **Port 443 – HTTPS**
* **Port 22 – SSH**
* **Port 3306 – MySQL/Aurora**
* **Port 2049 – NFS**

## Create The rds Instance

A screenshot of a computer

Description automatically generated

## RDS Settings | Create subnet groups

A screenshot of a subnet group

Description automatically generated

**Select the database subnet.**

-Private Data Subnet AZ1

-Private Data Subnet AZ2

Click Create

## RDS Settings | Create The rds database

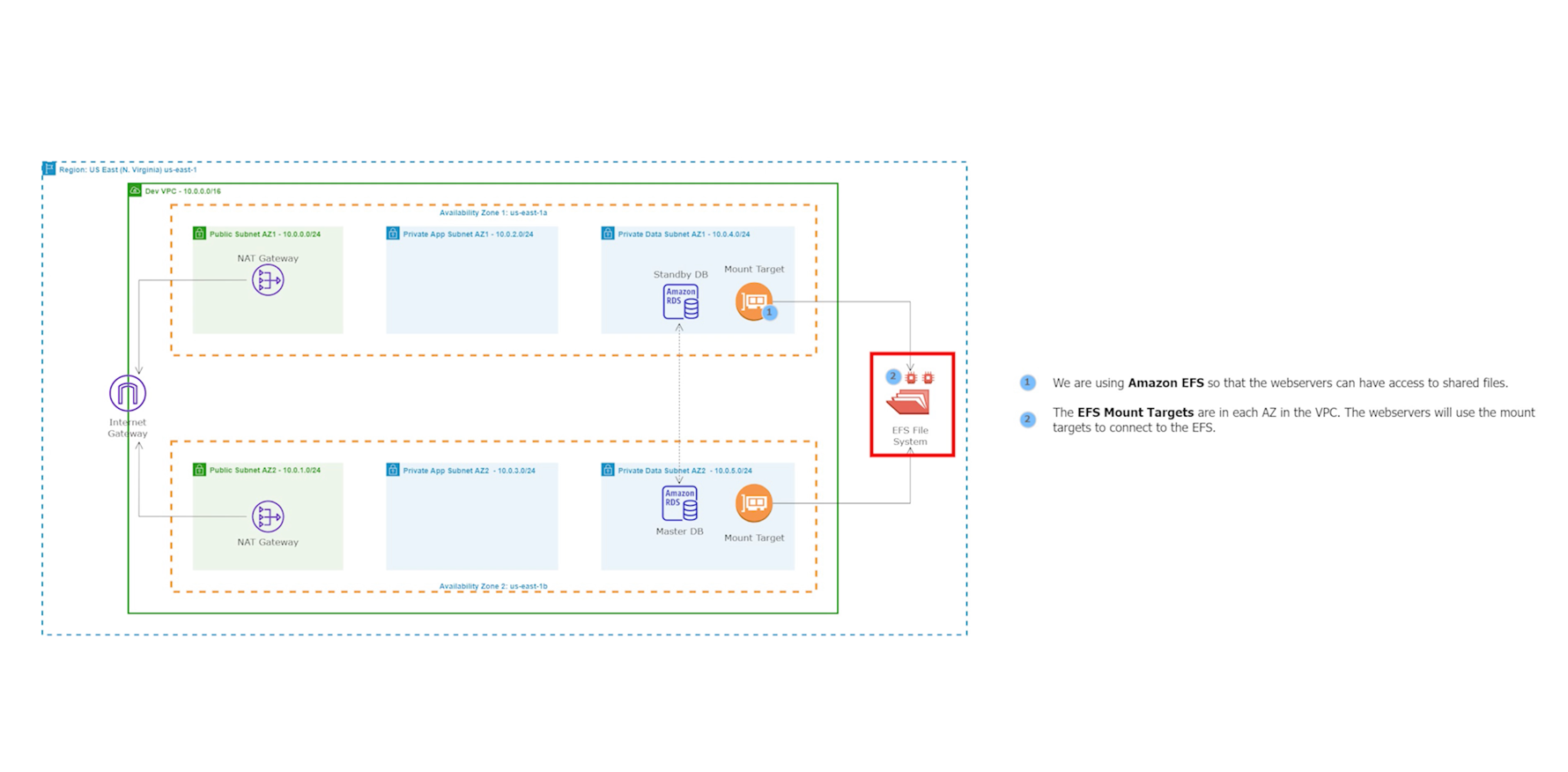
A screenshot of a computer

Description automatically generated

**Key Notes**

* For this project choose **Engine Version** 5.7
* On **Templates** choose Dev/Test
* **On Availability and durability** choose either ‘multi-AZ DB instance’ to create standby database in a different AZ which is chargeable for this project or choose ‘Single DB Instance’ which just create a single DB instance with not standby DB Instance and it’s not chargeable.
* **In Setting** Create your DB username and password and make sure to store it separately.
* **In DB Instance Class** choose ‘Burstable Classes’ toggle ‘include previous generation classes’ and your DB instance class should be ‘t2.micro’ which is free.
* **In connectivity** choose your VPC and in subnet group make sure the ‘subnet group’ you created before is selected. In the security group choose the ‘Database security Group’ and choose us-east-1b because that where we want to create our master database in.
* **In Additional Configuration** give your database a **name** otherwise Amazon RDS does not create a database.
* **Click create Database**

## Create Elastic File System (EFS)



**Key Notes**

* Go on EFS and click ‘Create File System’ and click ‘Customise’ and leave everything default and disable ‘Encryption’ so we don’t get charged.
* Click on tags and give it tag and Value.
* In the Network Setting, choose your VPC and select your tier 3 subnet and select the ‘EFS security group.’
* Create

## 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.

**A screenshot of a computer

Description automatically generatedKeynotes**

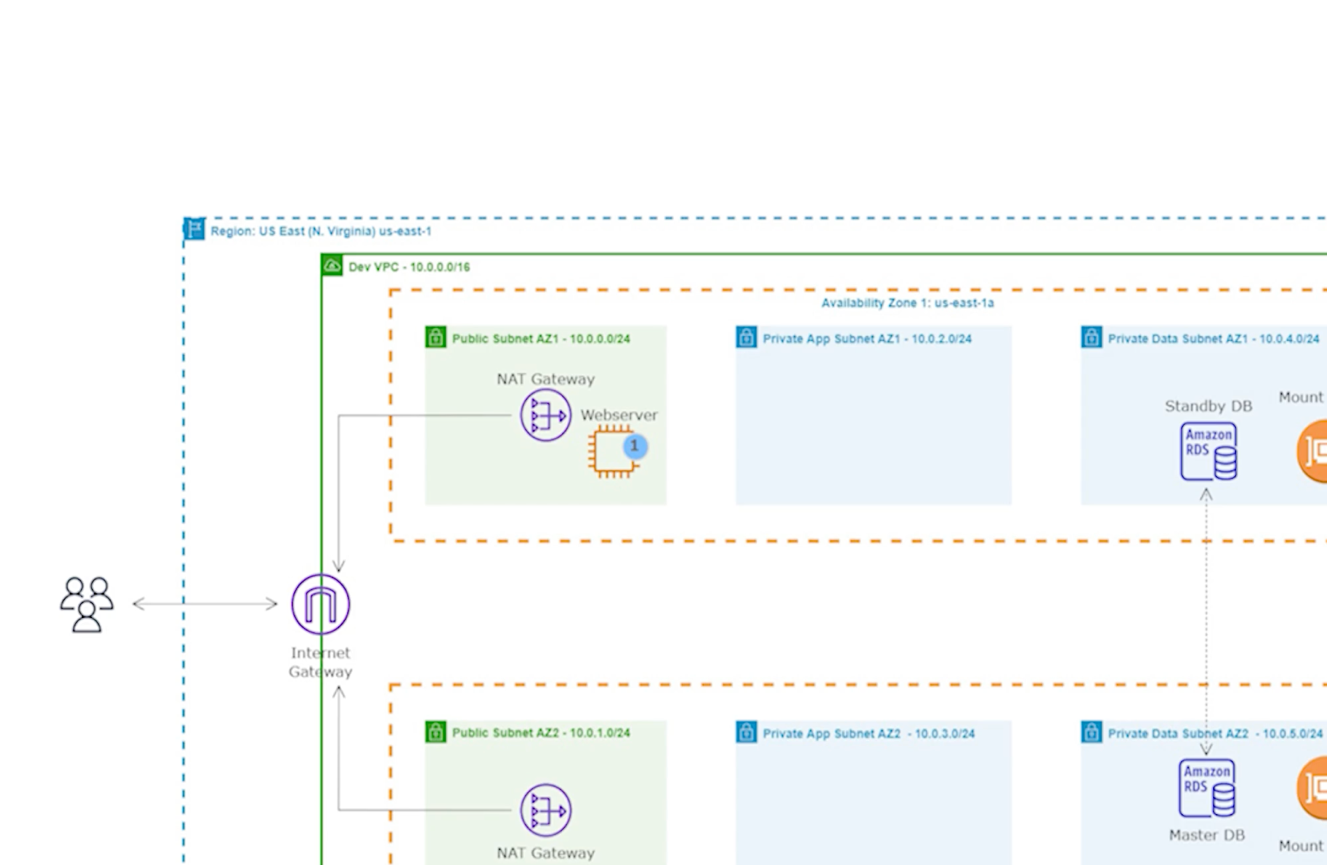
* Once your KP is downloaded, Move it to home directories.
  + - * Open up Terminal and run ‘chmod 400’ to set the permission on the KP.
* Chmod 400 NameOfYourKP.pem

-if you want to SSH on Windows use Putty.

**How to SSH in an EC2 Instance in the Public Subnet**

* Copy the Public IP address of the EC2 Instance.
* Open your Terminal.
* Type – ***ssh -I <NameOftheKP.pem> ec2user@<publicIPAddress>*** (press enter)
* **Type ‘yes’**

## Launch a Setup Server/ Public Subnet AZ1



**Lunch EC2 Instance**

* **Name** – Setup Server
* **AMI** – Must choose ‘Amazon Linux 2 AMI’ other it
* would not work.
* **Instance** **type** – t2micro
* **KP** – select your KP
* **Network** – choose ‘your Network’ and select ‘Public Subnet AZ1’ and make sure ‘public IP is enabled.’
* **Security Group** – SSH, ALB and Web-Server SG
* **Launch Instance.**
* Launch the EC2 instance in the ‘Public subnet AZ1’ to install our WordPress website.

## COmmands to Install WordPress site into the Setup Server

Commands were gonna run to install the WordPress site.

Keynotes

* Before you run the command make sure to mount your EFS DNS name to it, this can be found in the EFS you’ve created before.
* My 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 of this information in the **Configuration and Connectivity & Security section of your DB Instance)**

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

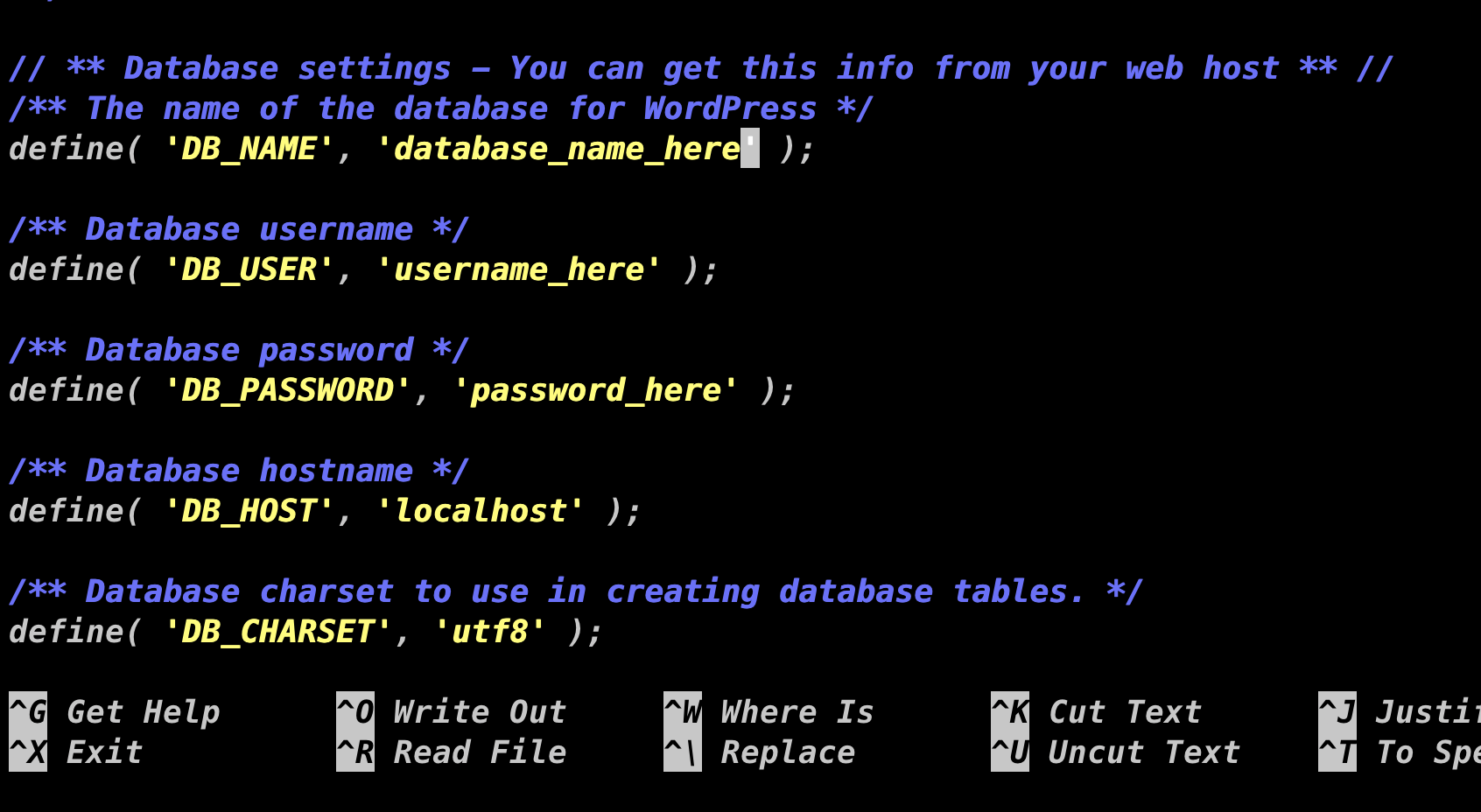
#9. restart the webserver

service httpd restart

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

Create your WordPress account.

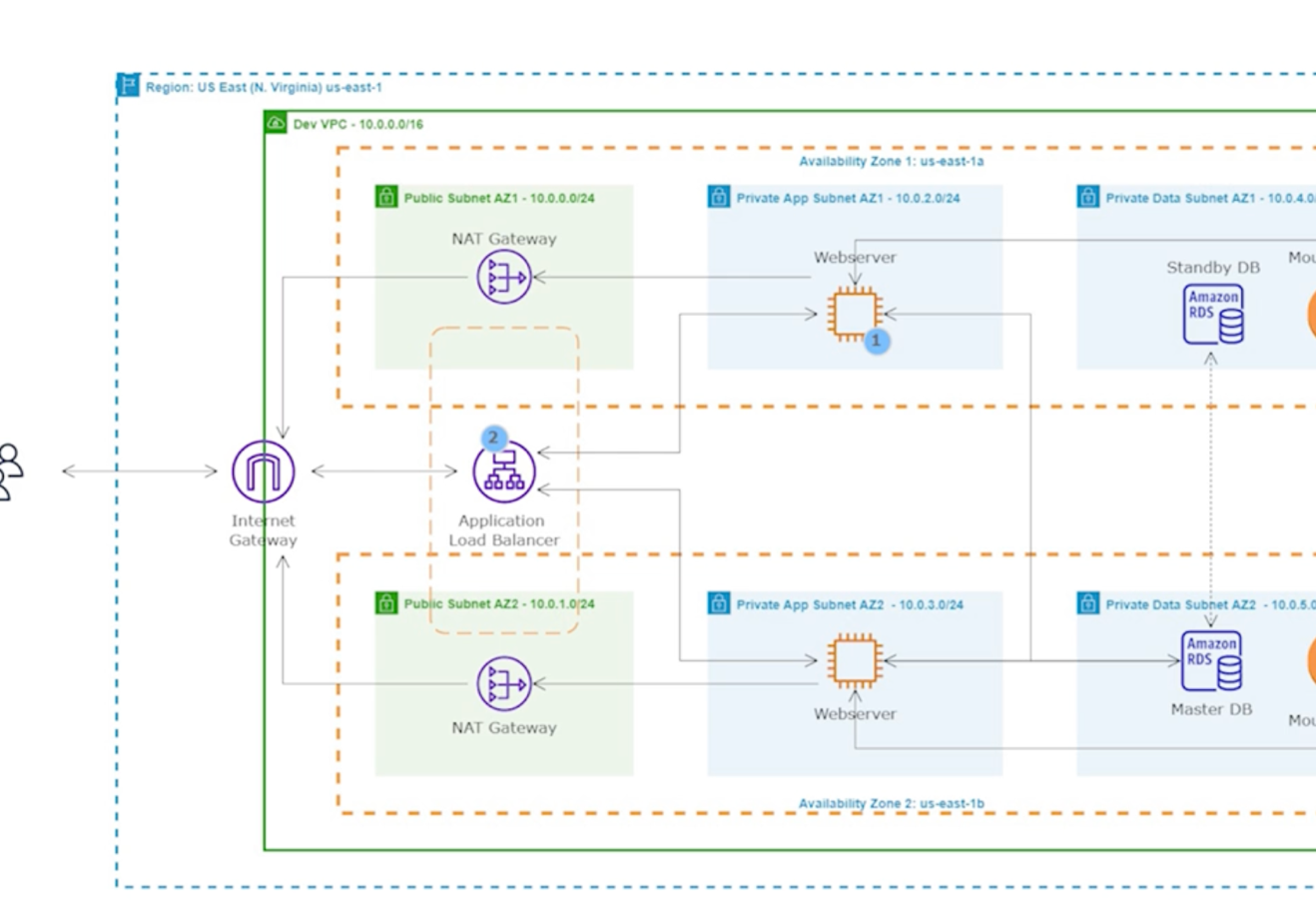
**Before**



**After (Yours should look like this)**



## Create an APPlication Load Balancer



**Key Notes**

* Create an application Load Balancer to route traffic to the EC2 instances in Private subnet.
* 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

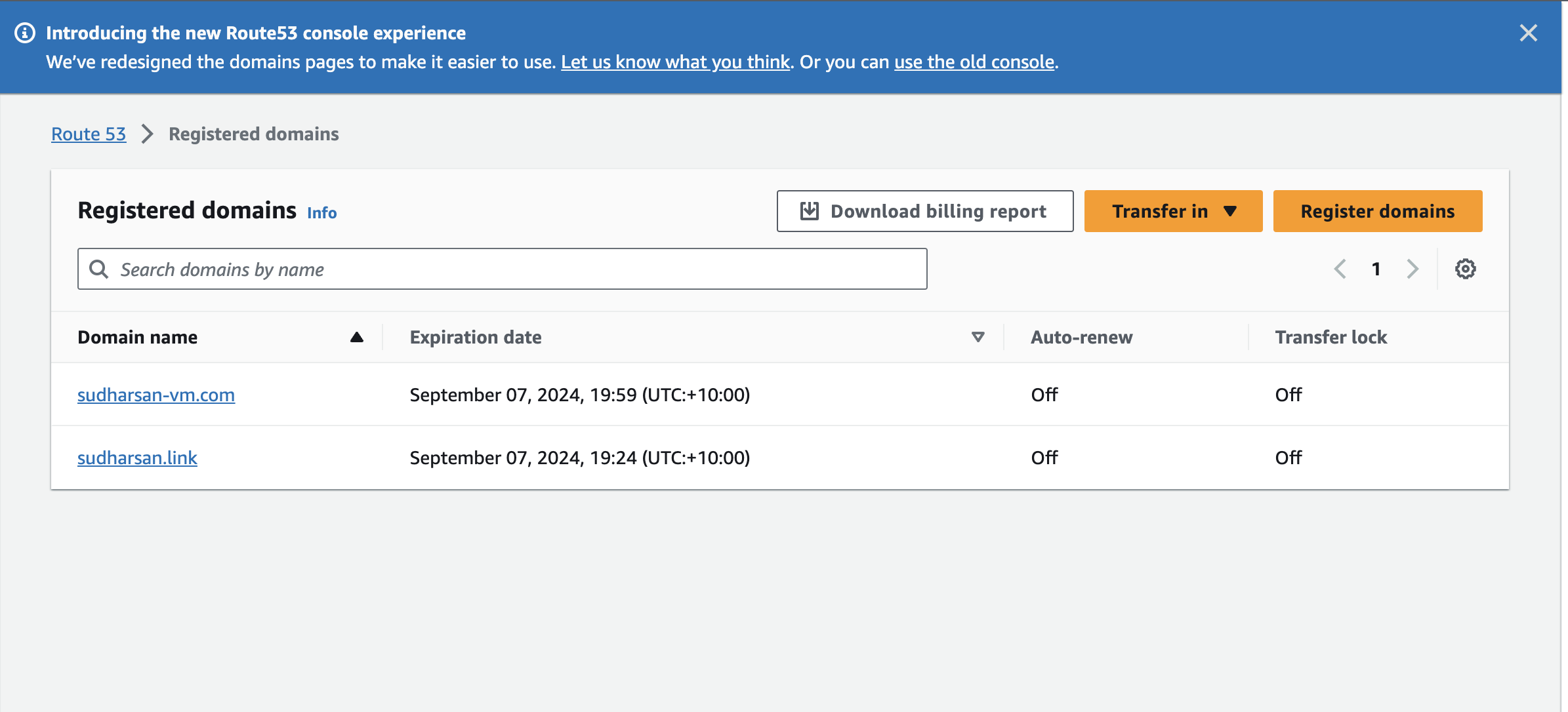
mount -a

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

sudo service httpd restart

* Network – Select your ‘VPC.’
* **Security group –** choose ‘webserver security group.’
* **TG –** Create a Target Group then put these ‘webserver’ EC2 instances in the Target group to allow the ALB to route there.
* **Create ALB –** create an ‘Application Load Balancer’ and map it to both of your AZs and you must choose one public subnet from each of the AZs.
* **Security group –** choose your **ALB** security group.
* **On listener –** choose your target group that you’ve created.
* **Once you’ve completed all this steps you can use your DNS name of your ALB to access your WP website.**
* **Remember anytime you change your Domain address, you have to go your WordPress settings and change you u Domain address there as well.**
* **Now that we have launched two EC2 instances in the private app subnets and we can access our websites using the DNS of our ALB. We can terminate our set up server.**

## Register a DOMAIN & Create a record set in route 53





Go on route 53, click on registered domains and register a domain there. Personally, from my experience choose .com, .net or .org because it might take longer for your domain to be available if you choose any other domain. Again, this is just from my personal experience.

A screenshot of a computer

Description automatically generated

Create a record set in route 53 with your ‘record name’ and point the alias to the ALB to access your website using your Domain name. remember, anytime you change your Domain name you must go into your WordPress setting and change your domain there as well.

## create SSL certificate using Certificate manager

A screenshot of a computer

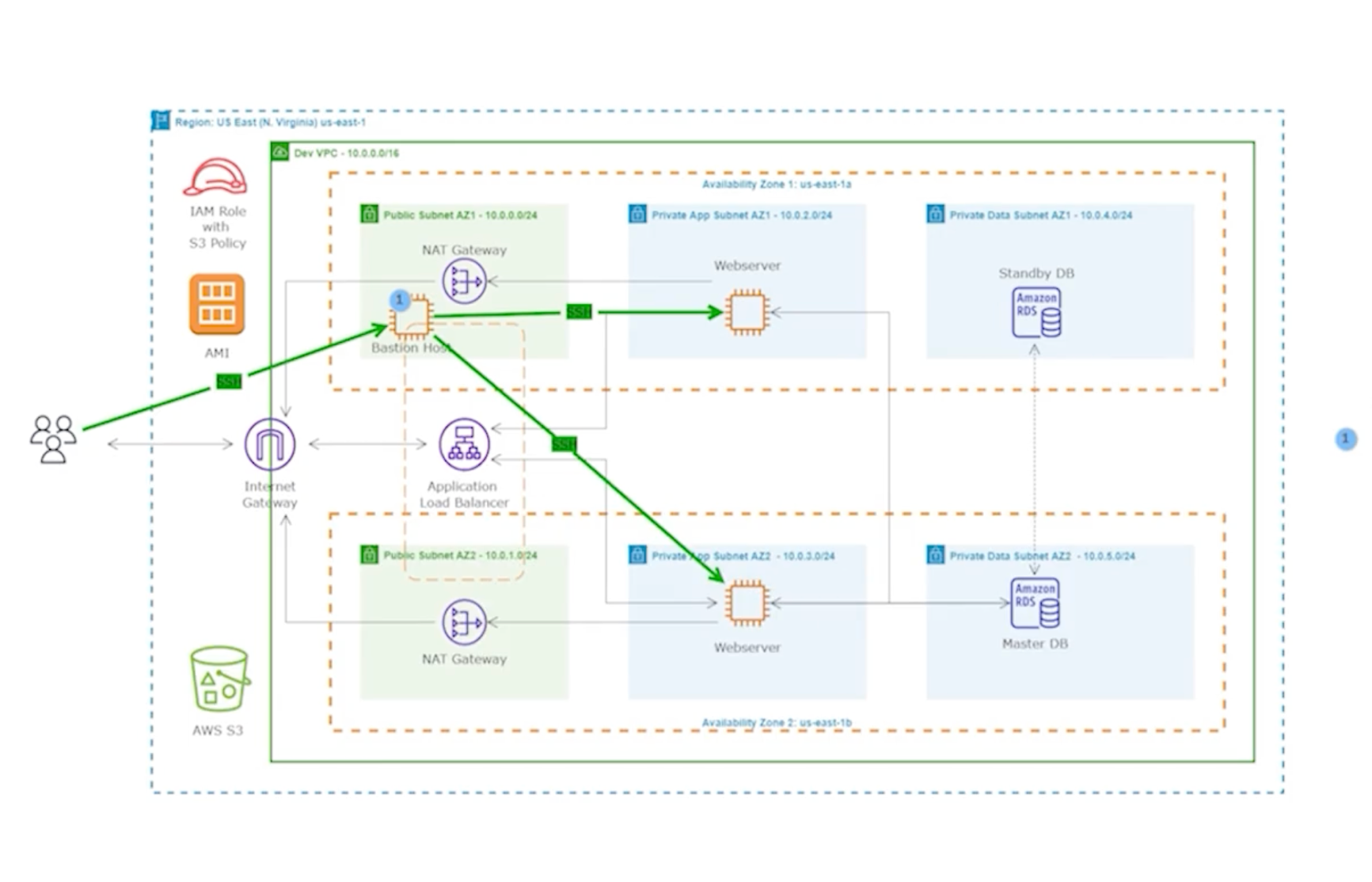
Description automatically generated



I used Certificate Manager create an SSL certificate and I created a record set to validate that this domain name belongs to me.

Once you finish this go back to your ALB and add a HTTPS Listener and go into your HTTP listener and redirect the HTTP request to HTTPS listener you just created.

## Launch a BAstion host



Create a bastion host in Public Subnet AZ1 to connect to webserver in Private App Subnet AZ1/2. We need to do this to edit the ‘WordPress Config’ file of our webserver for SSL certificate and the website should be secure with SSL certificate. Change you domain name in WP site.

**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

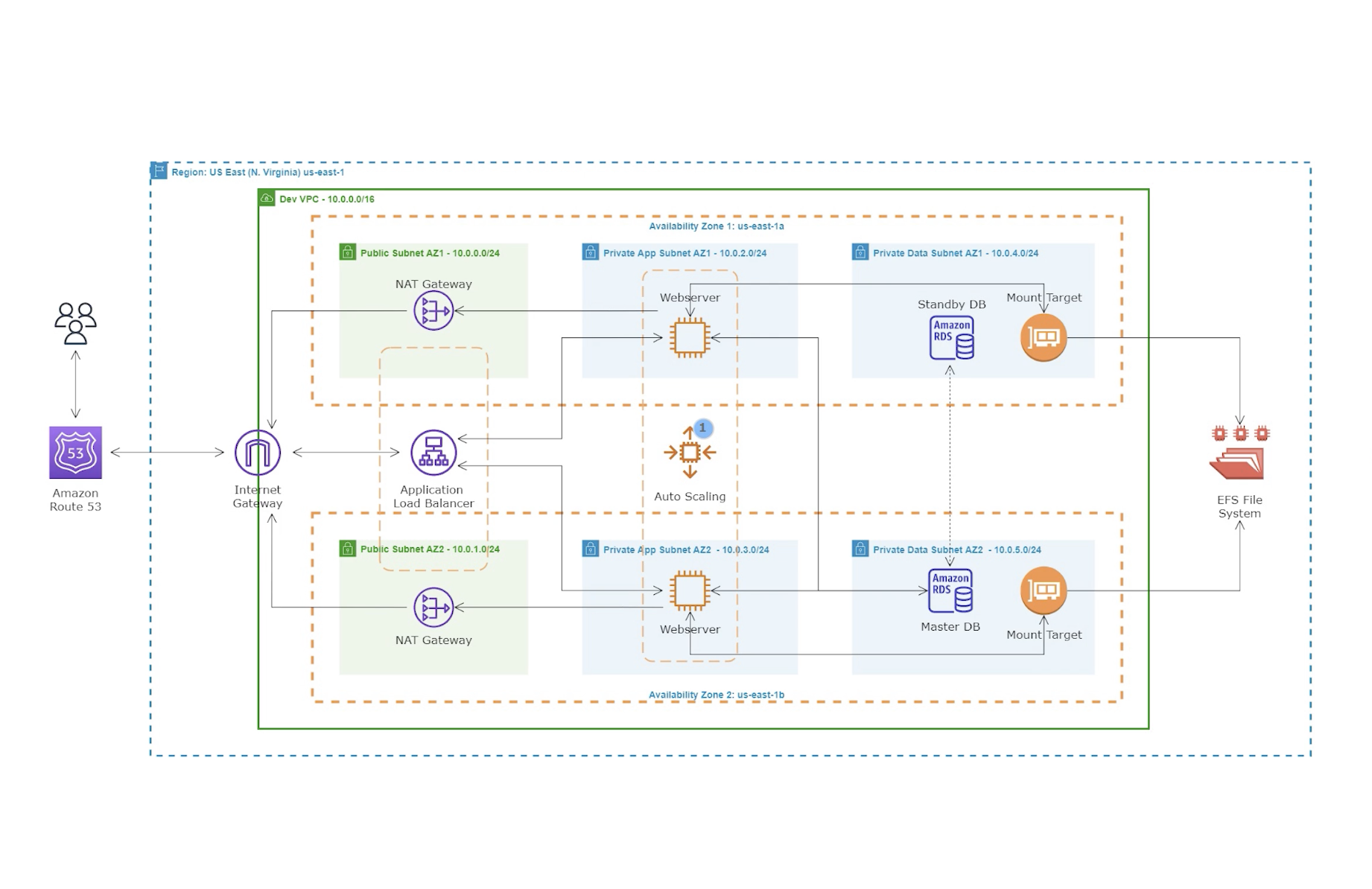
Command 3: SSH into the web server in private subnet

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

Example:

ssh ec2-user@18.232.135.220

## Create an Auto Scaling Group & LAunch templates



Here we gonna create an ASG and launch templates to dynamically scale the webserver in Private App Subnet. Since we’re gonna use Launch Template which contains the configuration about our EC2 instance that ASG will use to launch new EC2 instances, we can terminate the webserver instances we’ve manually created.

When you create your launch templates paste the commands that you used to when you created your webserver in the Private App Subnet. Paste it in user data when you create it. Then create a ASG from the Launch Template.

A screenshot of a computer

Description automatically generated

After this just edit your website template and you’ll be good to go.