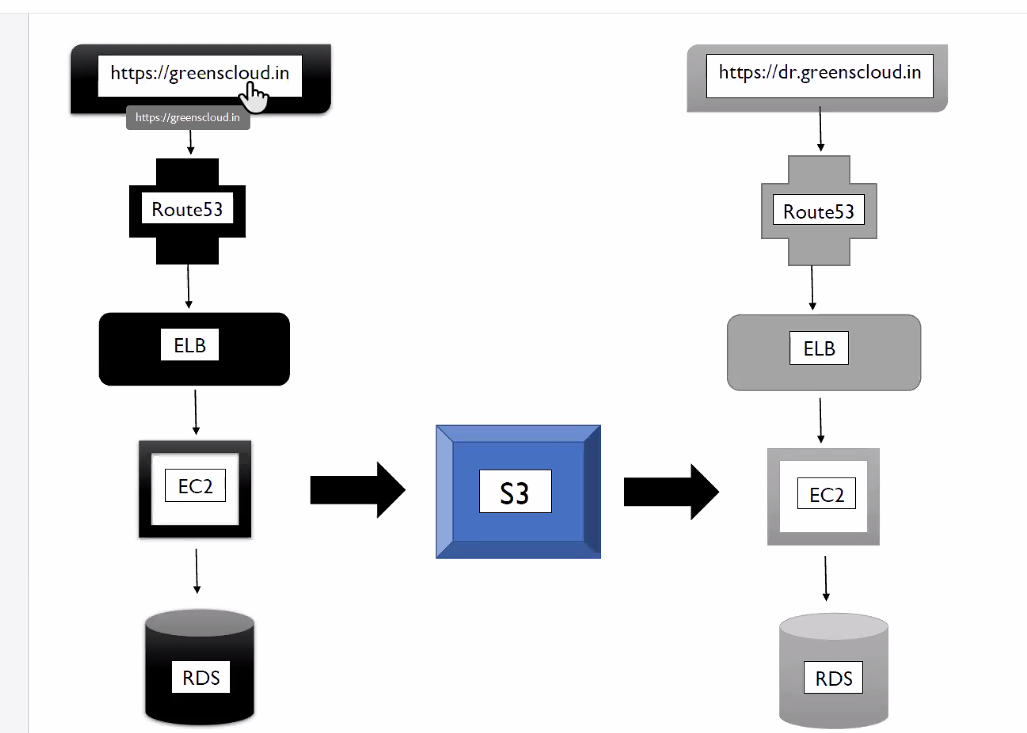
**AWS PROJECT**

**PRODUCTION AND DISASTER RECOVERY**

**Description**: To create the Production & Disaster Recovery environment in three -tier architecture by using Domain and also giving the SSL certification. The production server contain some application content, if suppose the production server may down or destroyed, we can recover the data from the disaster recovery server. DR server replicate the data in the production server.

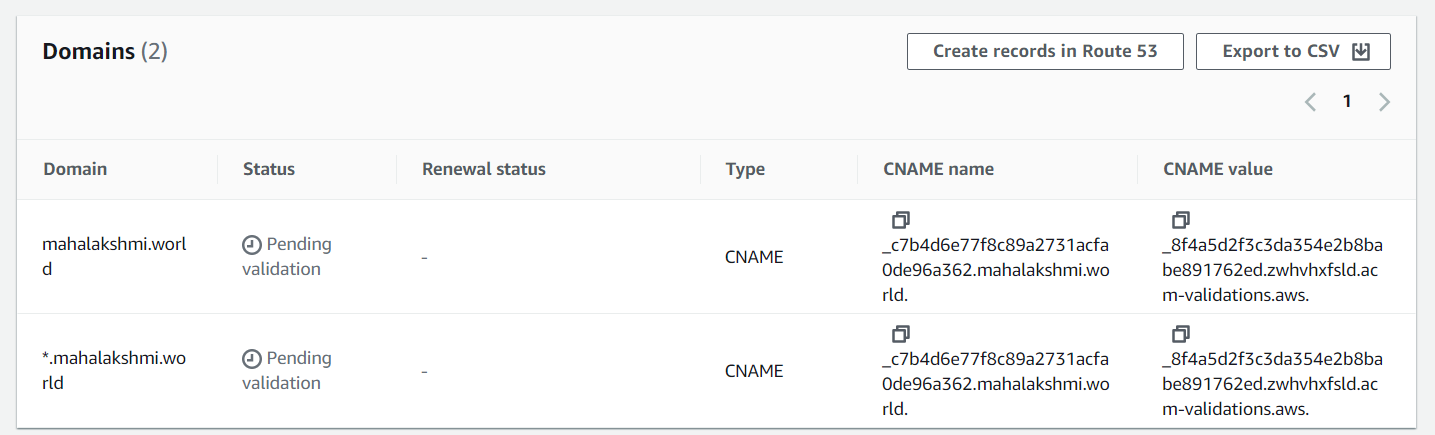
* The procedure to get server with configuration and replicate the database.
* Created ACM and R53 then integrate both.
* Security Groups to give access to traffic.
* Created the IAM roles.
* RDS setup with Subnet Group.
* Created an EC2 instance for production and DR server with user description (to install HTTPD, Linux extras and WORDPRESS).
* ELB setup with integration of those EC2 instance.
* Integrate R53 & ELB under the Alias.
* Created S3 bucket for production & Disaster Recovery.
* Transferring and made replicates between production and
* Disaster Recovery from Database to S3.
* Finally, attached the ACM certificate in ELB for HTTPS ports.

****

**STEPS:**

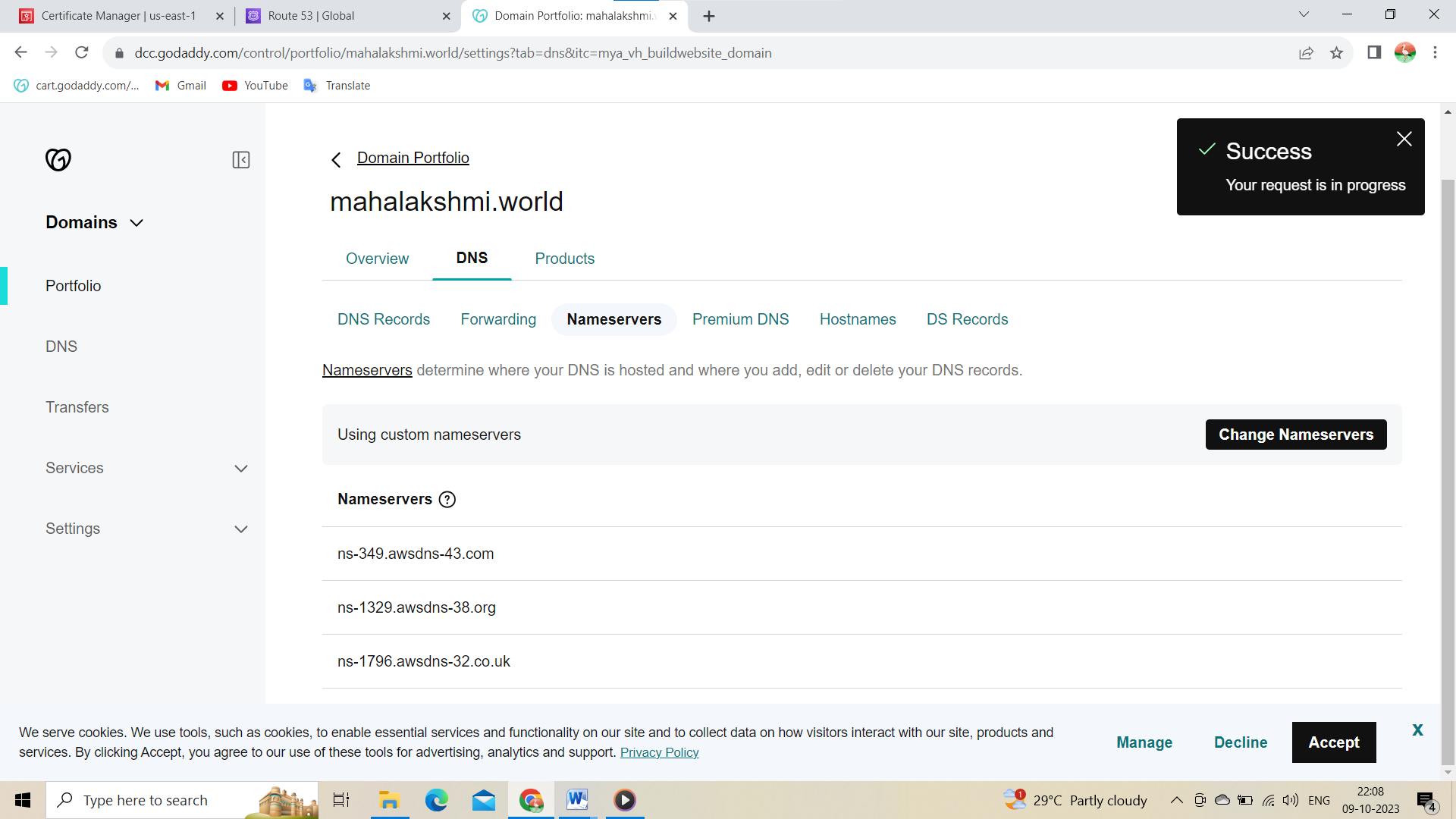
1. Purchase a domain name from go daddy
2. Get a ACM certificate

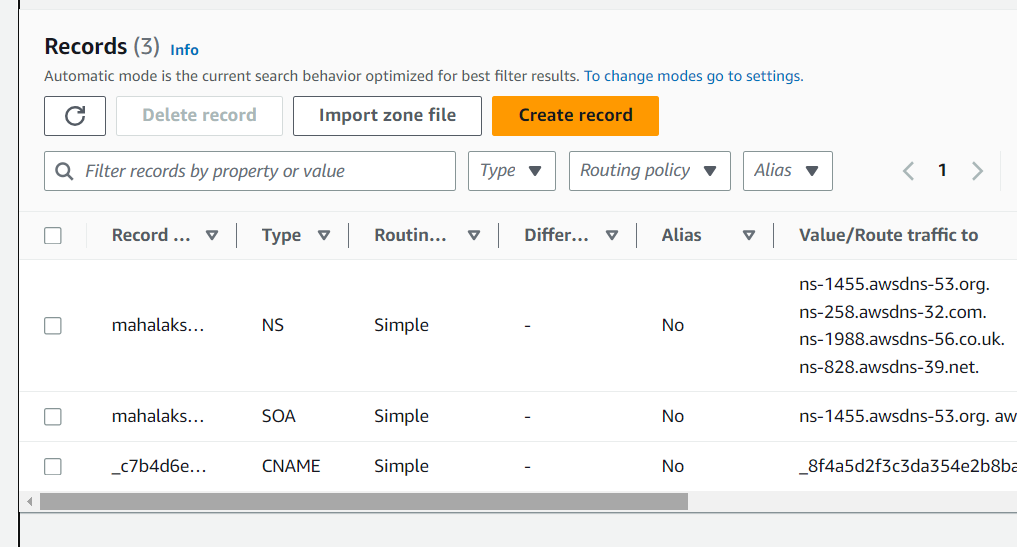
* Go to CERTIFICATE MANAGER in AWS console and request a public certificate.
* Request public certificate as per the registered domains which are 1. mahalakshmi.world 2. \*.mahalakshmi.world
* Select DNS validation and give a request.



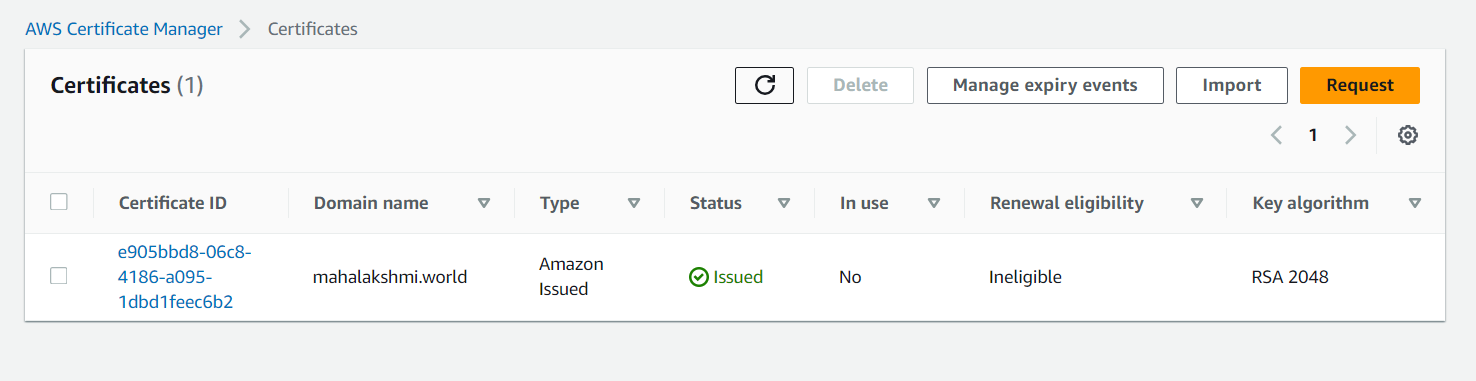
1. Create route53 and link to godaddy domain

* Create hosted zone with a registered domain
* Map the N-S under godaddy domain
* Create a record with route53 for registered domains

****

****

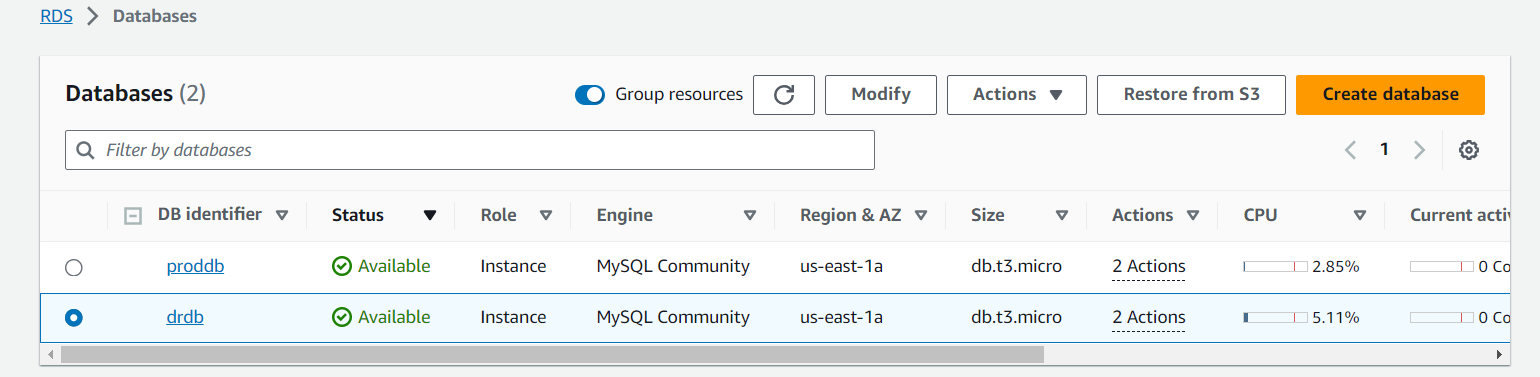
Certificate issued:



1. AWS RDS(Relational Database Service) setup

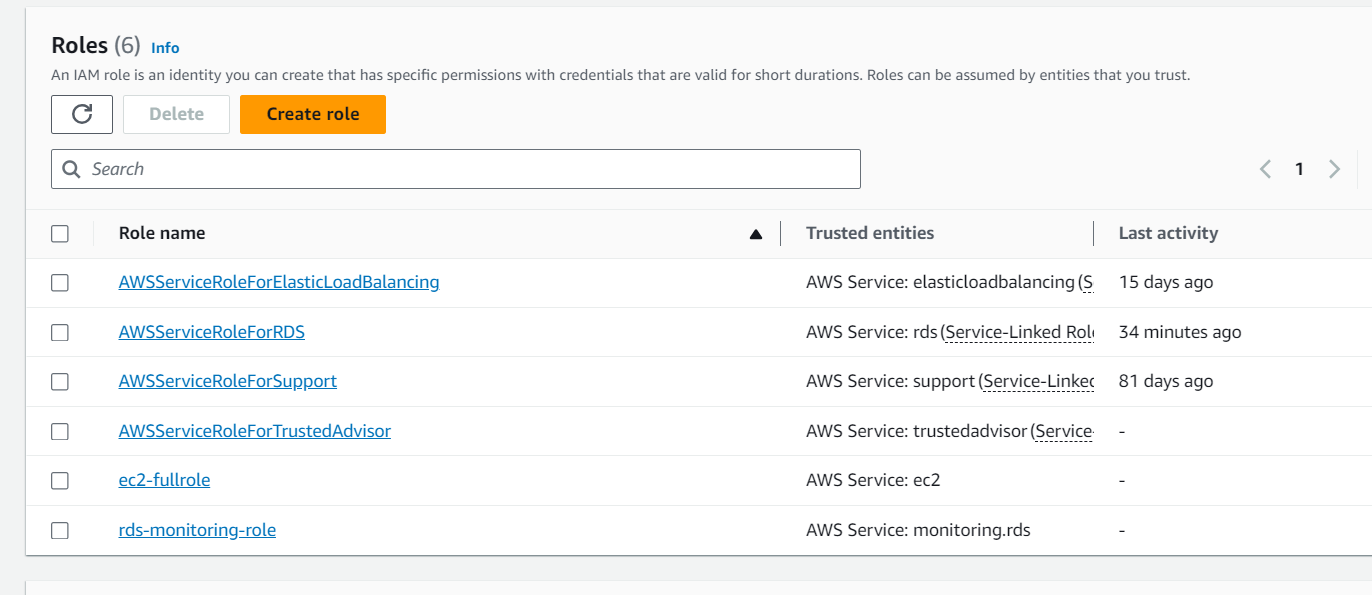
* Create DB subnet group with 3 availability zones
* Create 2 DBS

1. Production db
2. Dr db



1. Create IAM role

* Create an IAM role with full access
* Roll name – ec2-fullrole



1. AWS ec2 creation

* Create 2 ec2 instance with IAM roles enabled and add user data under advanced section
* Add the below details in the advanced section

#! /bin/bash

yum install httpd php-mysql –y

amazon-linux-extras install –y php7.3

cd /var/www/html

echo “healthy” > healthy.html

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

tar –xzf latest.tar.gz

cp –r wordpress/\* /var/www/html/

rm –rf wordpress

rm –rf latest.tar.gz

chmod –R 755 wp-content

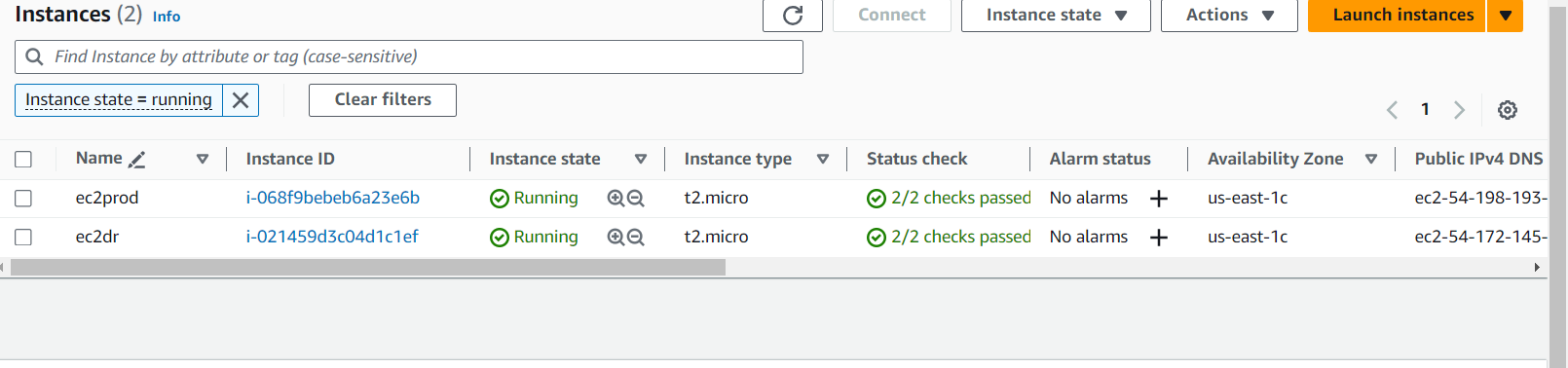
chown –R apache:apache wp-content

wget <https://s3.amazonaws.com/bucketforwordpresslab-donotdelete/htaccess.txt>

mv htaccess.txt .htaccess

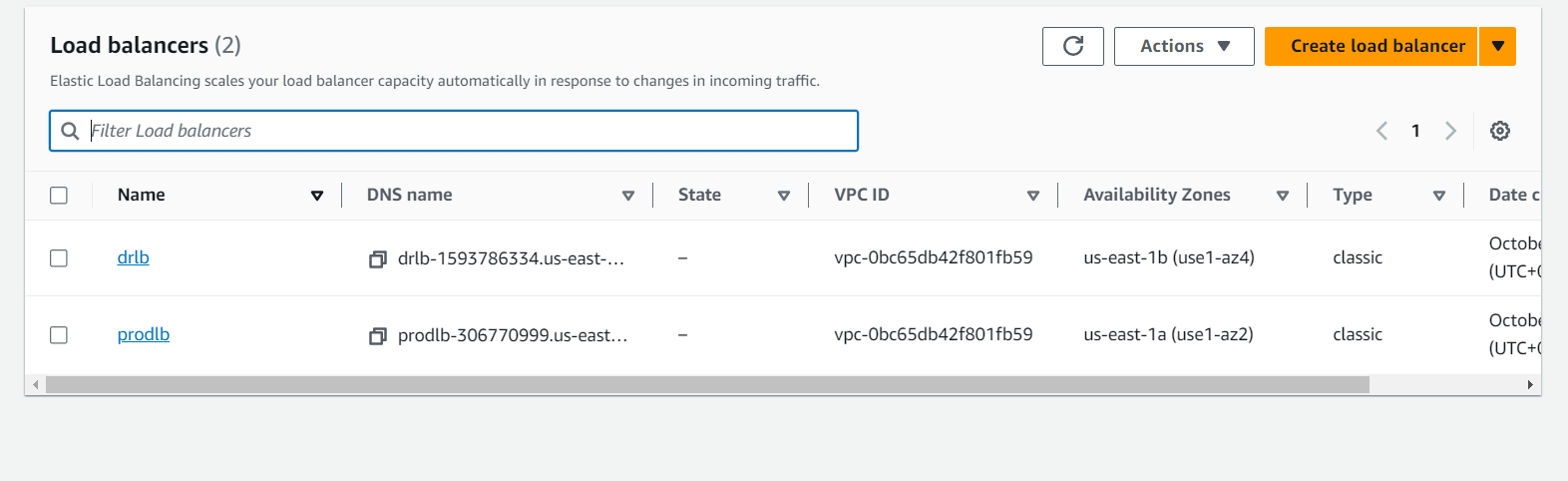
chkconfig httpd on

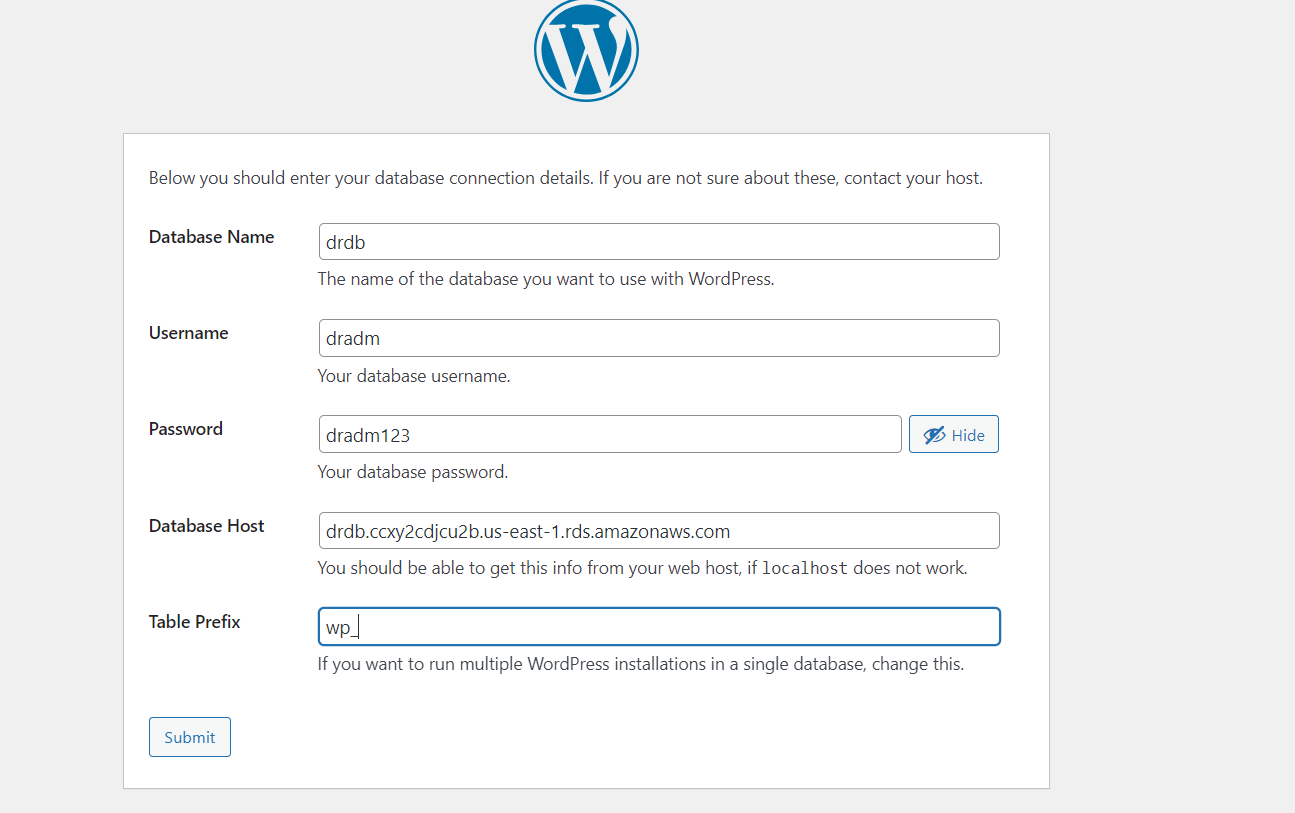
service httpd start



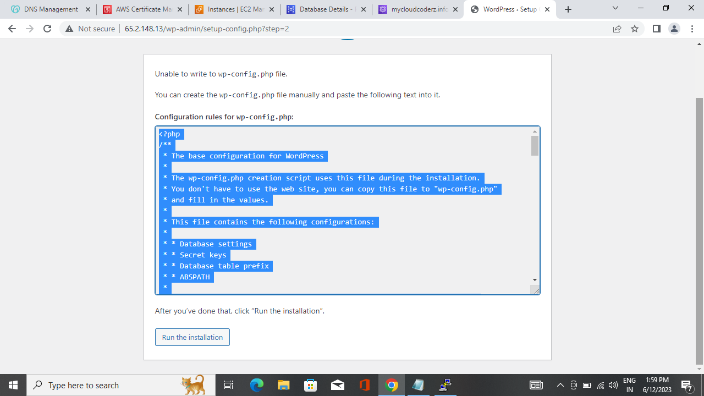
1. AWS ELB creation and setup

* Create 2 classic ELB and map the appropriate ec2 to it

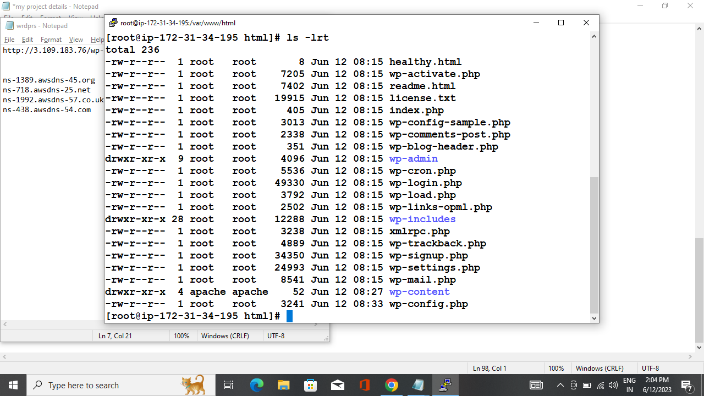




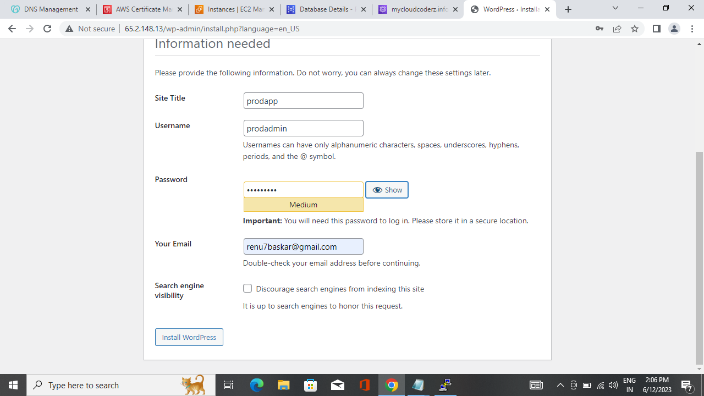
1. Copy wp-config.php text file and paste it on prod server by giving vi wp-config command



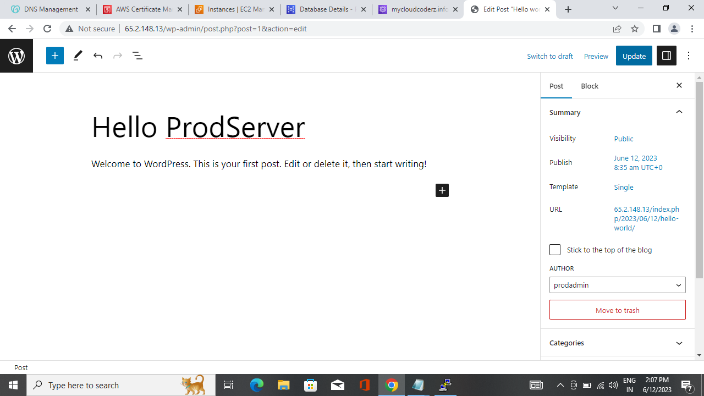
1. whether wp-config.php file has been added to server using ls -lrt command



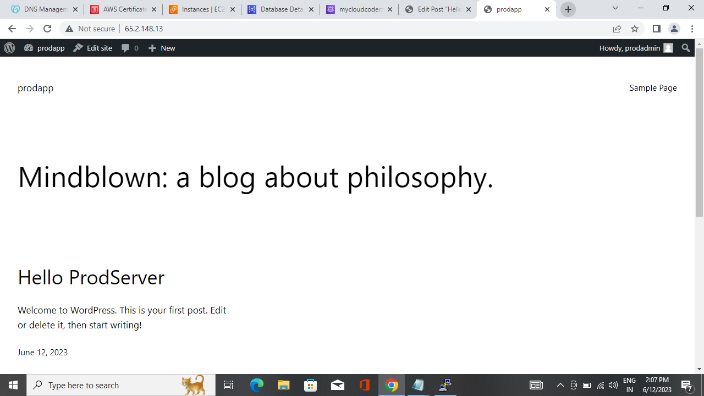
1. Provide the login details of wordpress



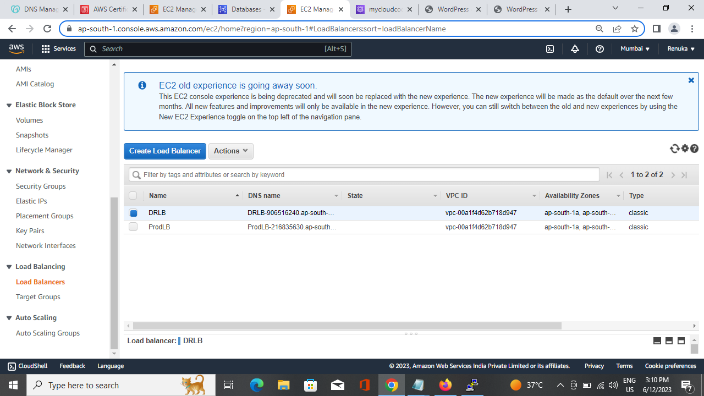
1. Edit the post on wordpress page. Change hello world to hello prodserver it Will immediately affect in prodserver.



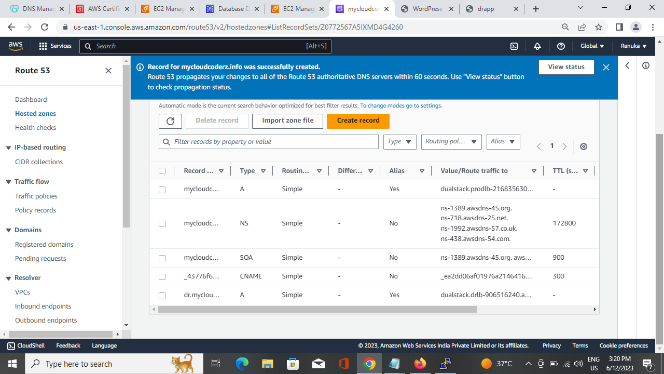
1. Copy the ip address of prod server and paste it on browser. It will show hello prodserver



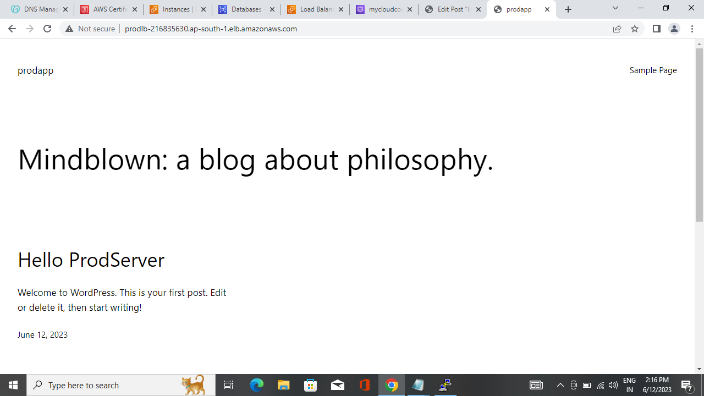
1. Create two classic load balancer. One for producion and another for DR



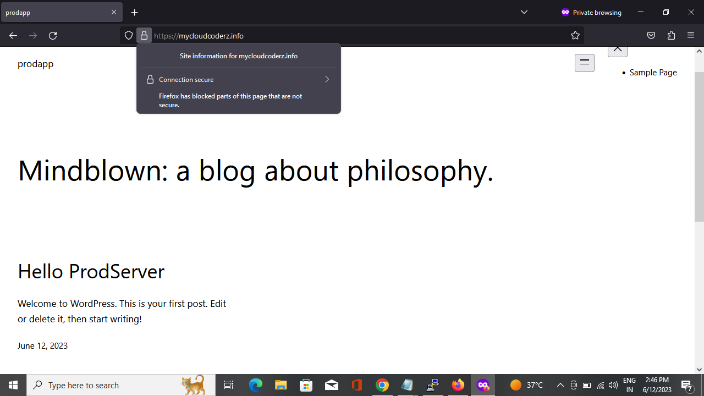
1. Create recordset for prodLB and DRLB



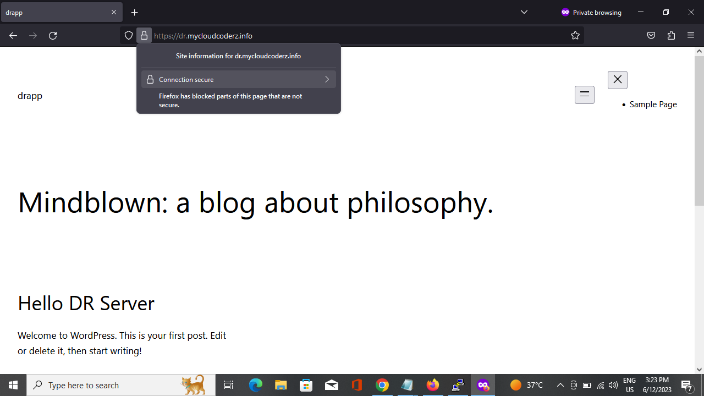
1. Copy the DNS name of prodLB and paste on browser. It will also Show hello prodserver



1. Give the domain name (https) on browser. It will show hello prodserver and check the connection is secured.



1. Do the same process for DR server what we have done for prod server And give the sub domain name



1. Create two buckets in s3. Give crontab command to upload files automatically from ec2 to s3 for production and s3 to ec2 for DR server

