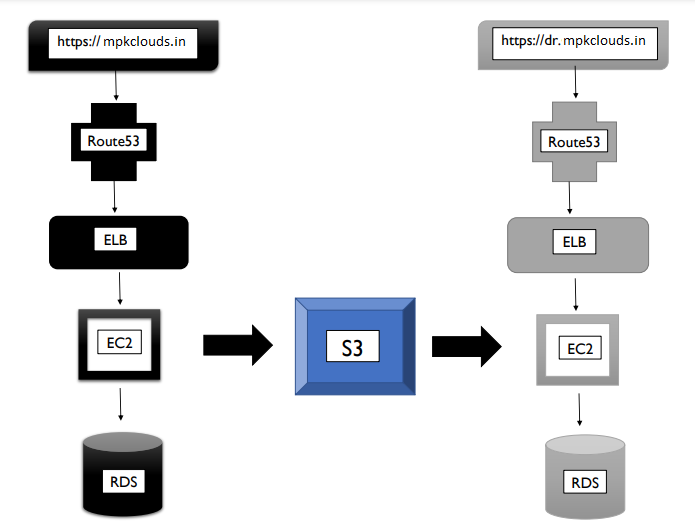
**IMPLEMENTION OF PRODUCTION AND DISASTER RECOVERY ENVIRONMENT FOR A TWO-TIERED APPLICATION DEPLOYMENT USING ACM, ROUTE53, EC2, ELB, RDS, IAM, S3, CLOUD WATCH**

**ARCHITECTURE**

This project done by using AWS - ACM , ROUTE 53, ELB, EC2, S3, RDS and IAM and CLOUD WATCH



**ACM:**

AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources.

**ROUTE 53:**

It is a highly available and scalable Domain Name System (DNS) web service It is designed for developers and corporates to route the end users to Internet applications by translating human readable names like www.mydomain.com, into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other.

**EC2:**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

**ELB:**

Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more Availability Zones (AZs).

**IAM:**

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

**S3:**

Amazon Simple Storage Service is a scalable, high-speed, web-based cloud storage service. The service is designed for online backup and archiving of data and applications on Amazon Web Services. Amazon S3 was designed with a minimal feature set created to make web-scale computing easier for developers.

**CLOUD WATCH:**

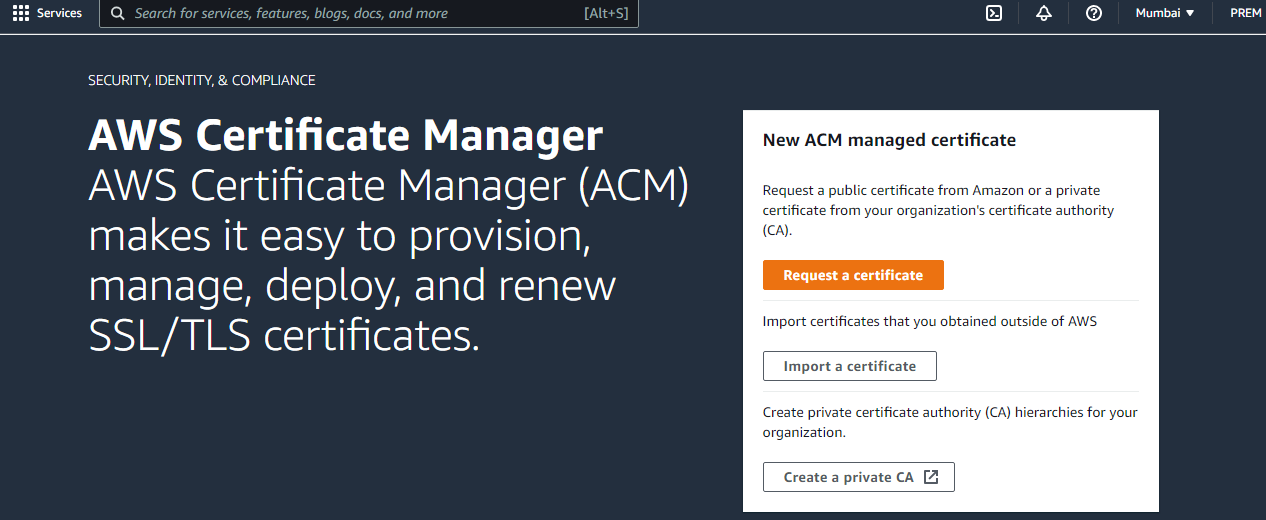
CloudWatch enables you to monitor your complete stack (applications, infrastructure, and services) and use alarms, logs, and events data to take automated actions and reduce mean time to resolution (MTTR).

**WORKING PROCEDURE:**

**1. AWS CERTIFICATE MANAGER CREATION:**

Request public certificates as per the registered domains. ACM certifications can be used to establish secure access to communications over the Internet or within an internal network.

# Go to aws certificate manager – request a certificate – request of public certificate.



# Domain name : mpkclouds.in

# Subdomain name: \*.mpkclouds.in

**Validation method:**

There are two types of validations exist.

1. DNS validation

2. Email validation

Choose DNS validation. DNS validation is the SSL/TSL certificates in Route 53 service. Route 53 requires validation of this certificate.

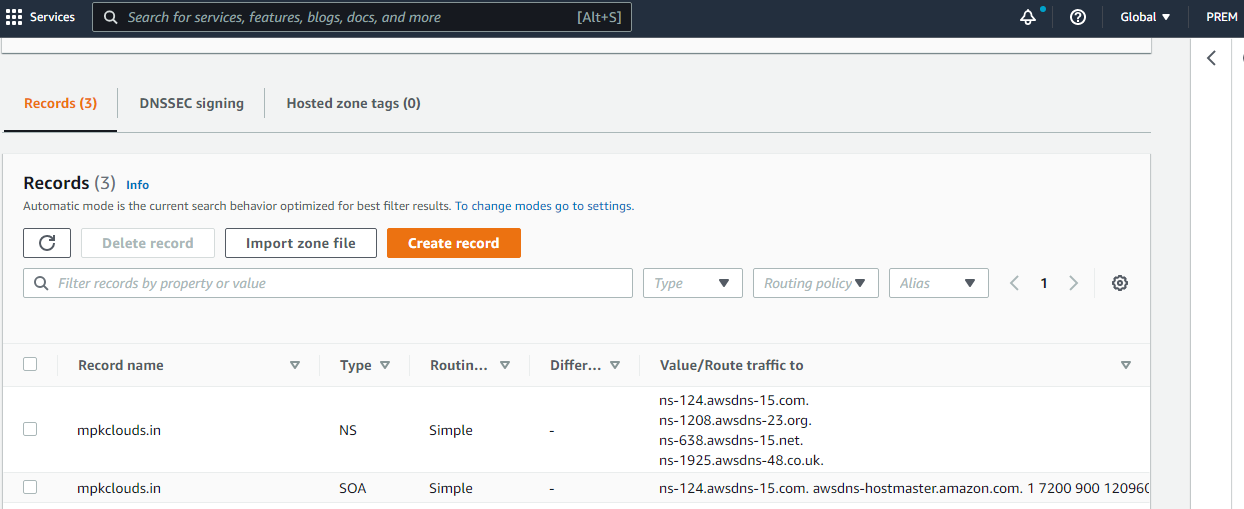
**2. AWS ROUTE53 SETUP & LINK TO GODADDY DOMAIN DNS MGMT:**

**A. Create Hosted Zone:**

# Go to route53 – create hosted zone.

# Purchase a domain from godaddy .

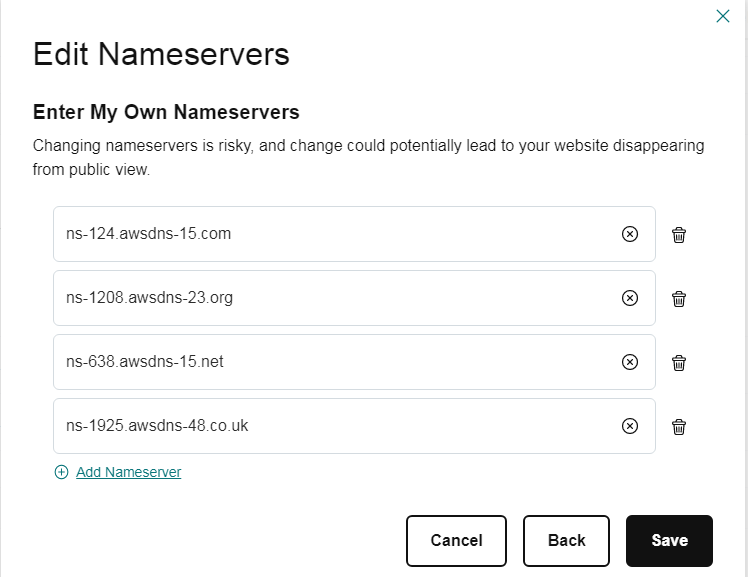
# Domain name: mpkclouds.in (Default two record sets will be created )



**2.Map The N-S Under Godaddy Domain:**

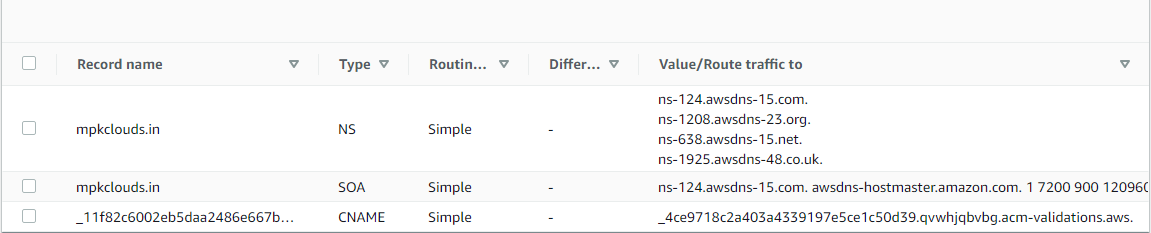
# Copy these name servers and paste it under godaddy’s name server .

* ns-124.awsdns-15.com
* ns-1208.awsdns-23.org
* ns-638.awsdns-15.net
* ns-1925.awsdns-48.co.uk

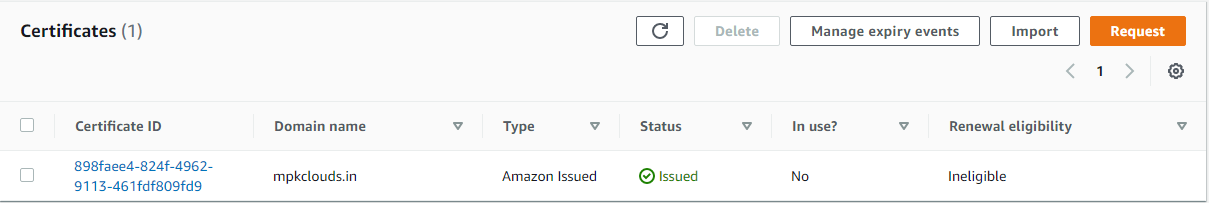


**3.CREATE RECORD SET FROM ACM**

# Go to ACM – select existing certificate ID – domains – create records in Route53 (Now three number of record set is available in route 53)



# Go to ACM – It shows Certificates validation is success. Status shows issued.



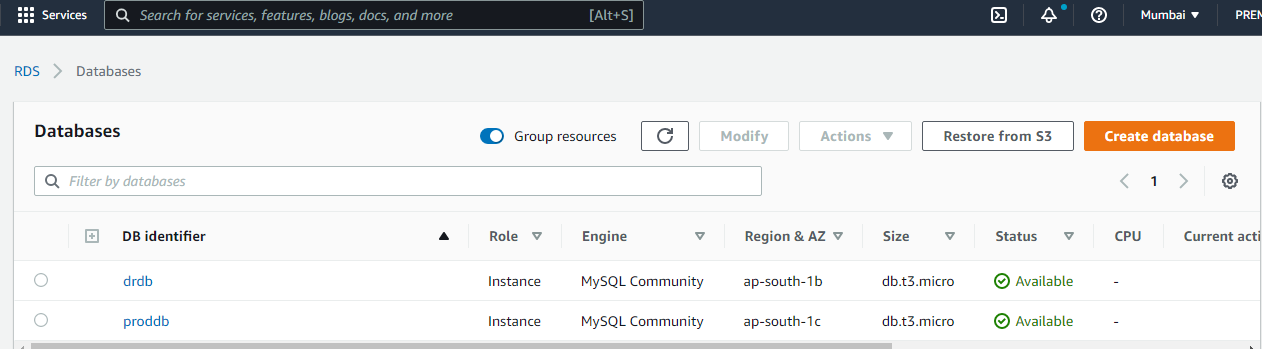
**3. AWS RDS SETUP:**

# create two number of data base.

i) production db - In this data base security group make as all traffic

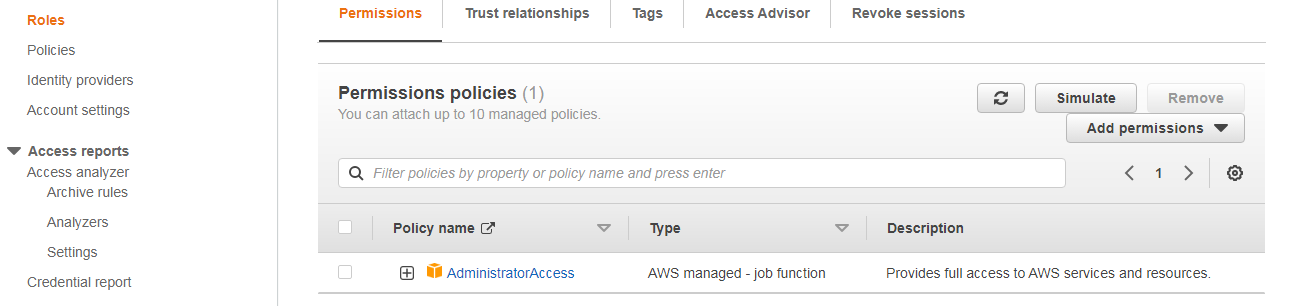
ii) dr db – In this data base security group make as all traffic

# MySQL data base engine is used for both data bases.



**4. AWS IAM ROLE**

# Create an iam role with full access.



**5. AWS EC2 CREATION**

* create two no of ec2 server with iam roles enabled & add user data under advanced section
* add the below details in the advanced section,

**USER DATA:**

#!/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

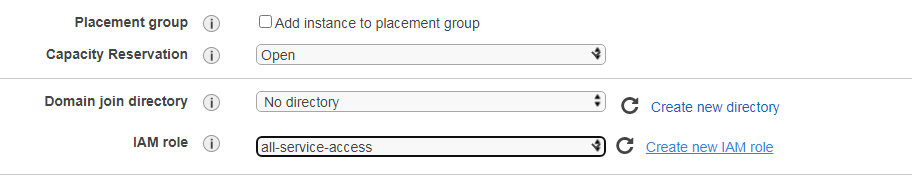
chown -R apache:apache wp-content

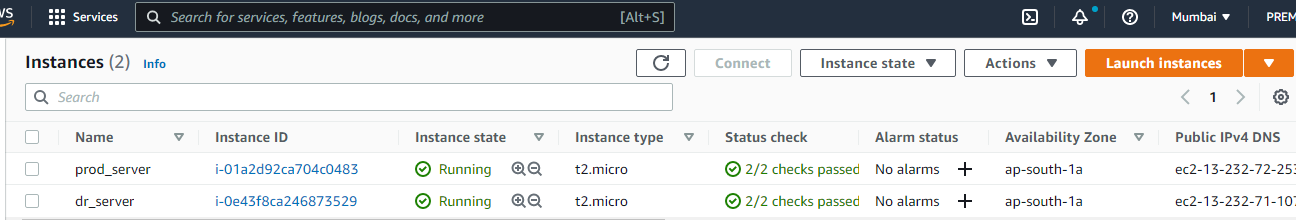
wget https://s3.amazonaws.com/bucketforwordpresslab-donotdelete/htaccess.txt

mv htaccess.txt .htaccess

chkconfig httpd on

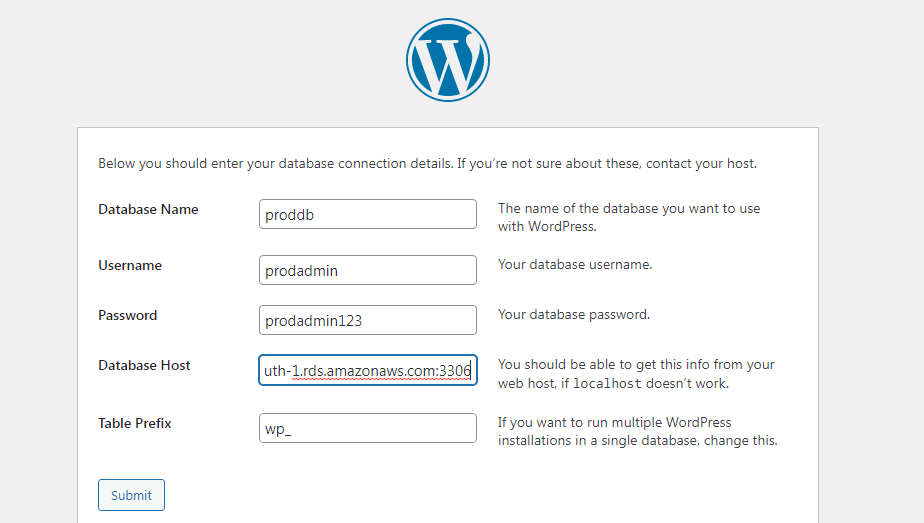
service httpd start

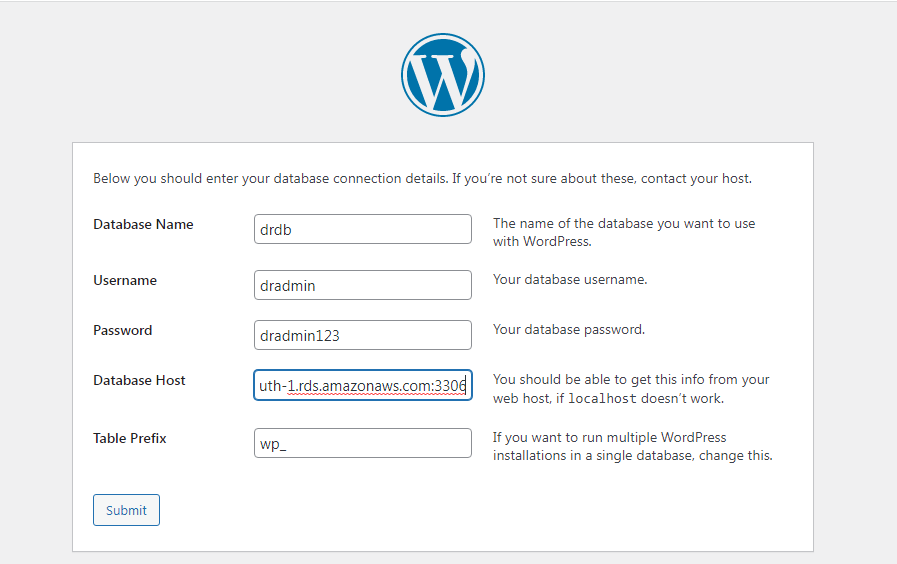




**7. ADD THE DB DETAILS UNDER WORDPRESS APPLICATION FOR BOTH DR & PRODUCTION**

# Hit the browser by production ip and dr server ip to access word press web page.



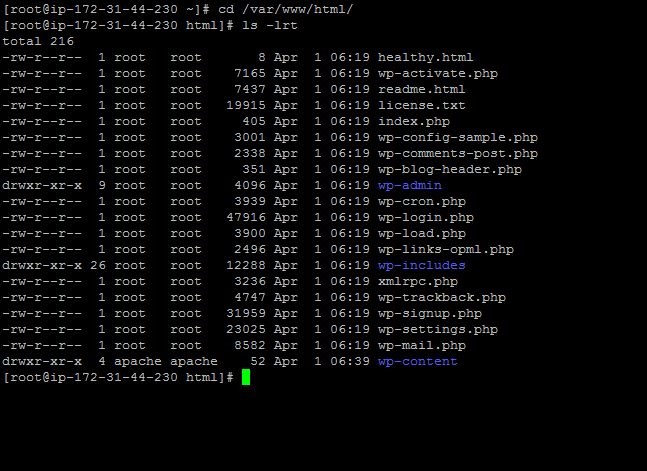


# cd /var/www/html

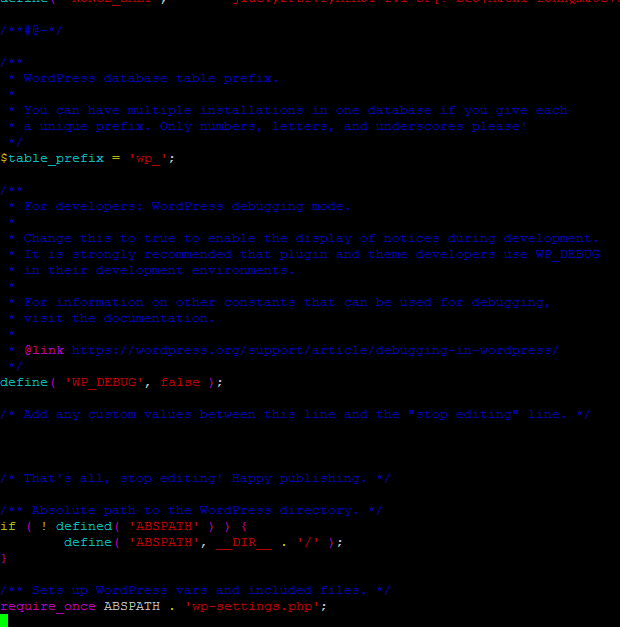
# create & edit the wp-config.php file in both the ec2s

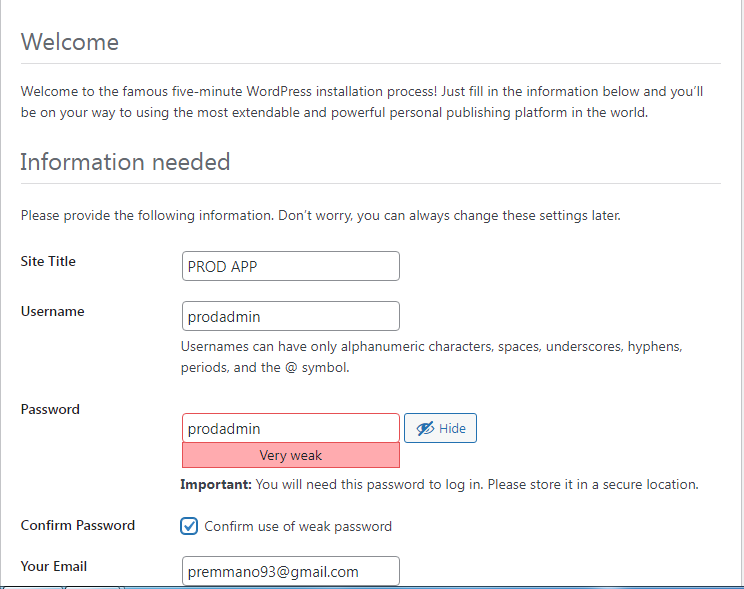
# php file is missing. So you have to create wp-config.php file.

# Copy the content from hosted application page and paste it inside the wp-config.php file and run the installation



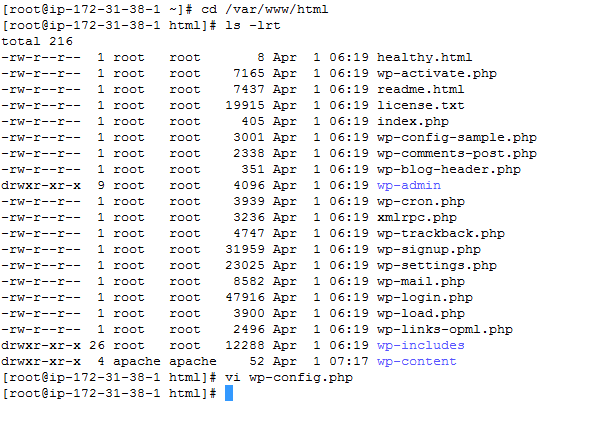
# vi wp-config.php

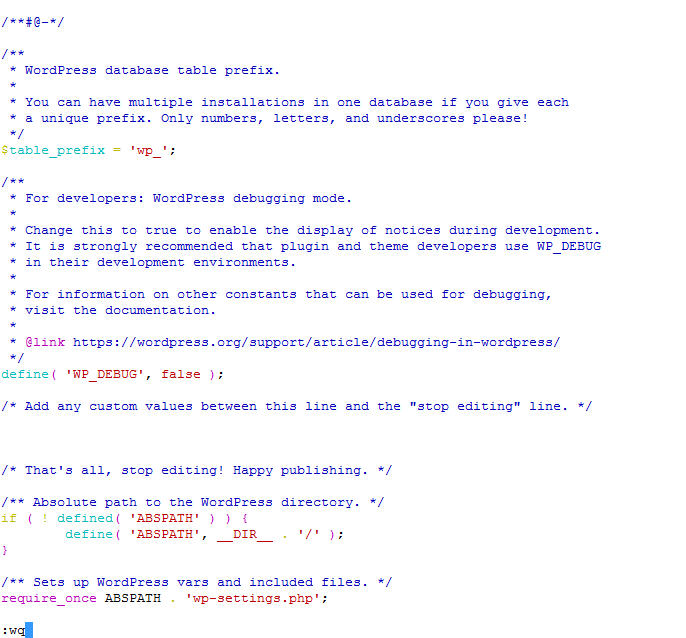


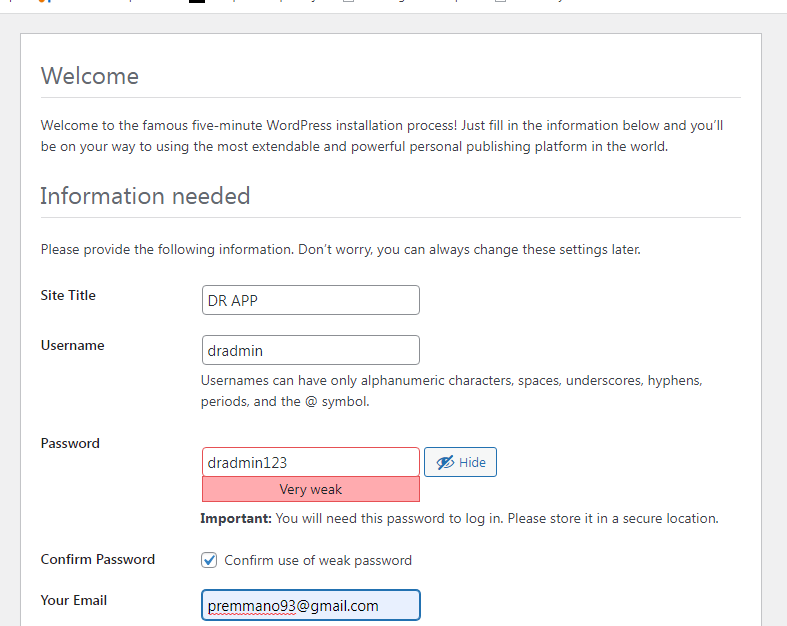


Now the application page will be opened and used by the administrator rather than the end user.







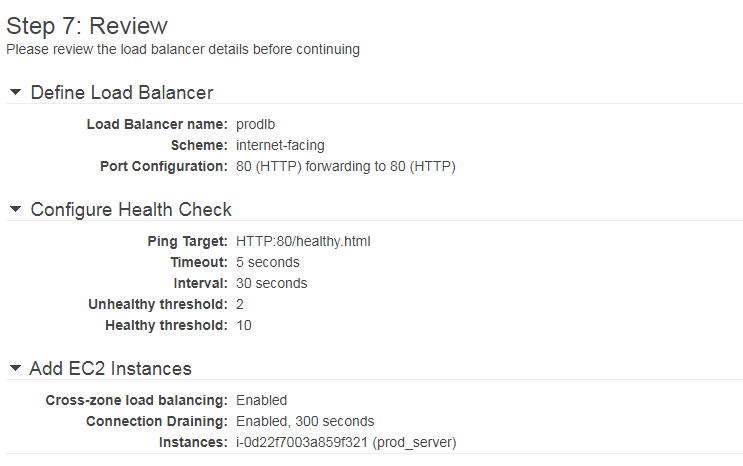


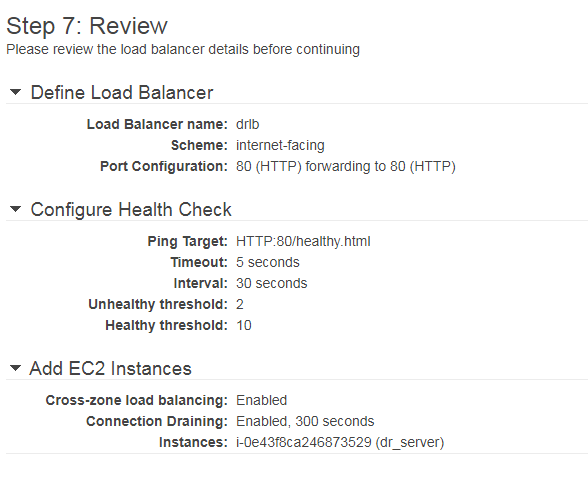
Now the application page will be opened and used by the administrator rather than the end user.



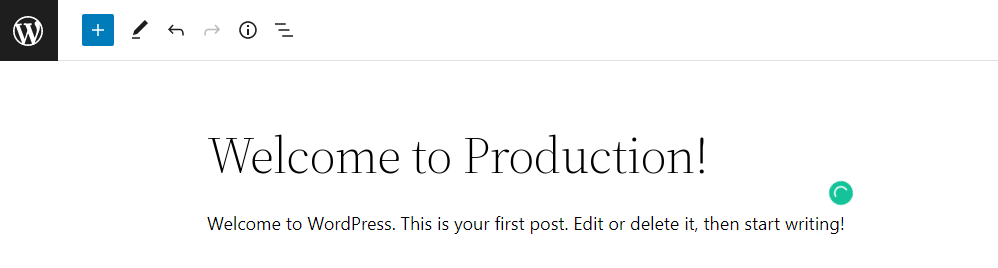
**6. AWS ELB CREATION & SETUP**

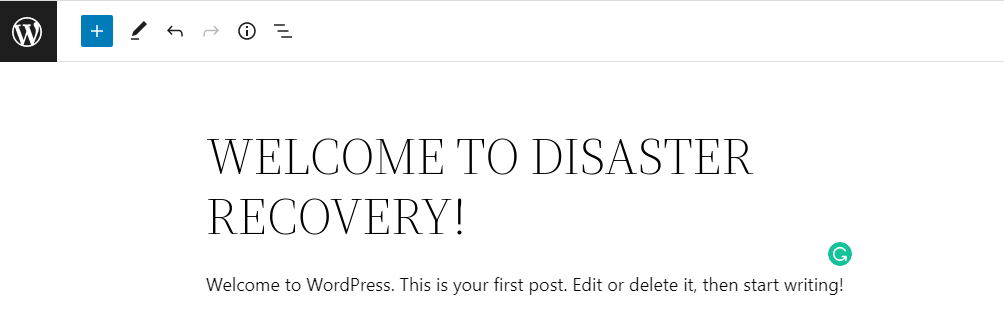
# create two number of classic elb & map the appropirate ec2 to it.





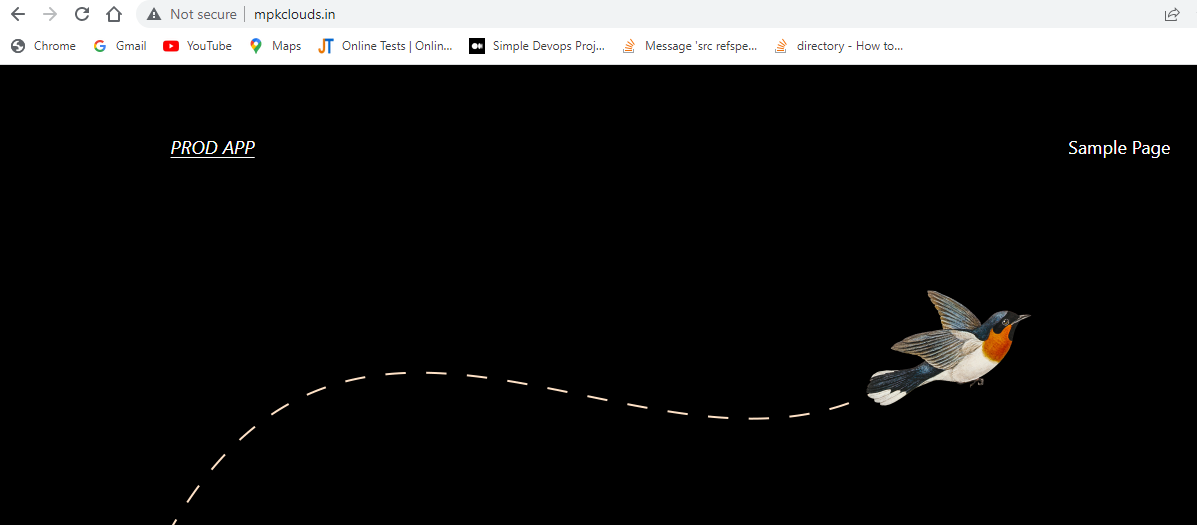
**8. LAUNCH THE WORDPRESS PAGE & POST IT WITH APPROPIRATE DR & PROD IMAGES TO GET DIFFERENCE IN THE APPLICATION PAGE.**

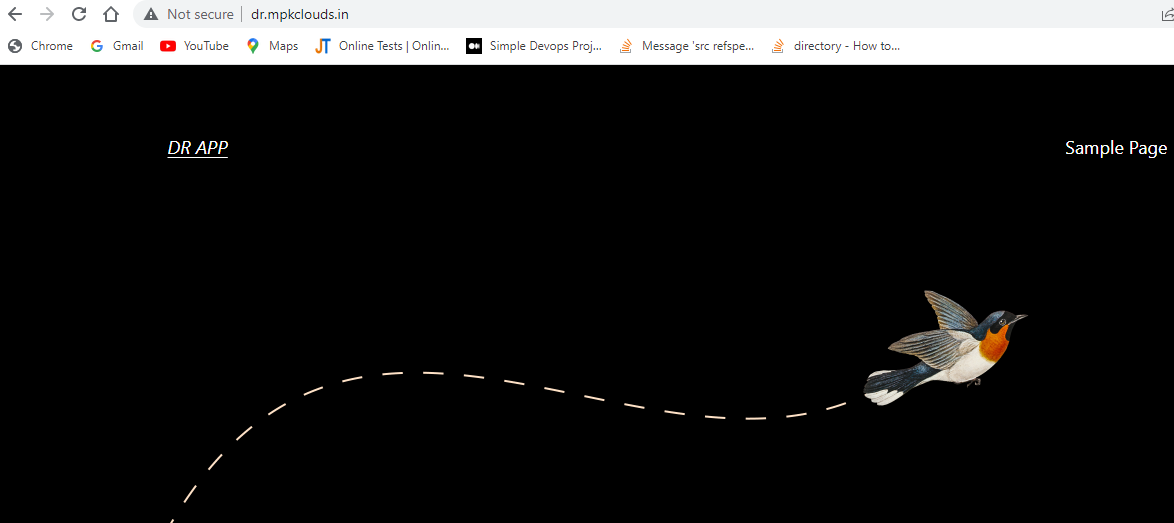




**9. AWS R53 & ELB SYNC:**

# Hit the browser by following urls <http://mpkclouds.in> and <http://dr.mpkclouds.in> . It shows app is not secured.





# refresh the r53

# create record set and map the appropirate elb under the alias

# Integtrating Route 53 with Load Balancer . Now we have to Create Record two domains

1.Record name : mpkclouds.in

# Record type – A – Route traffic to an IPV4 address and some AWS resources

# Route traffic to Alias is Enable

-Alias to Application and classic Load Balancer

-Asia Pacific (Mumbai) [ap-south-1]

-dualstack.prodlb-125612298.ap-south-1.elb.amazonaws.com

# Route policy - simple routing.

2.Record name : dr.mpkclouds.in

# Record type – A – Route traffic to an IPV4 address and some AWS resources

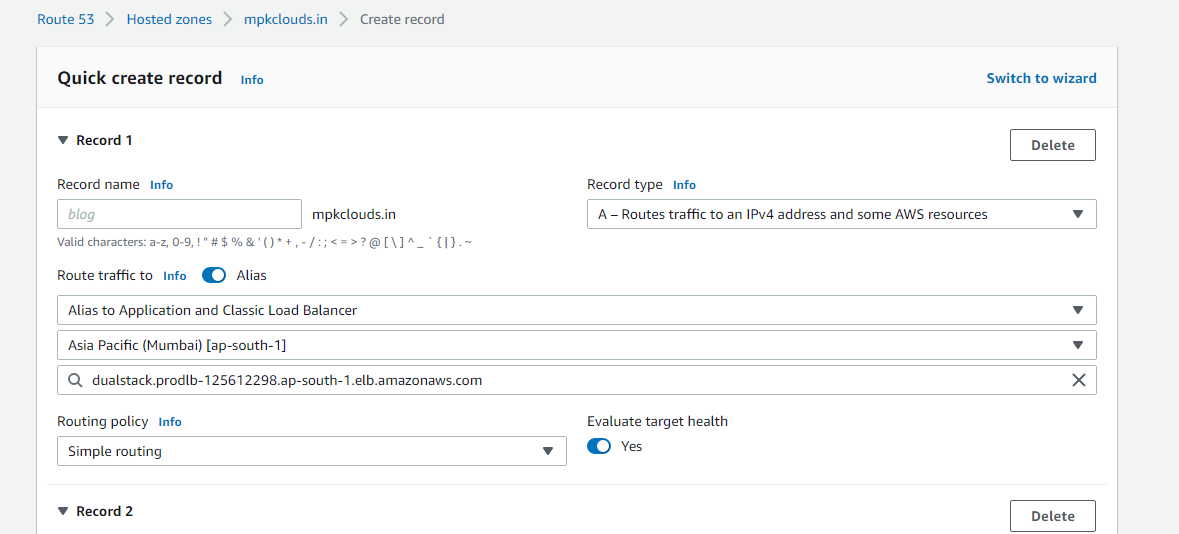
-Route traffic to Alias is Enable

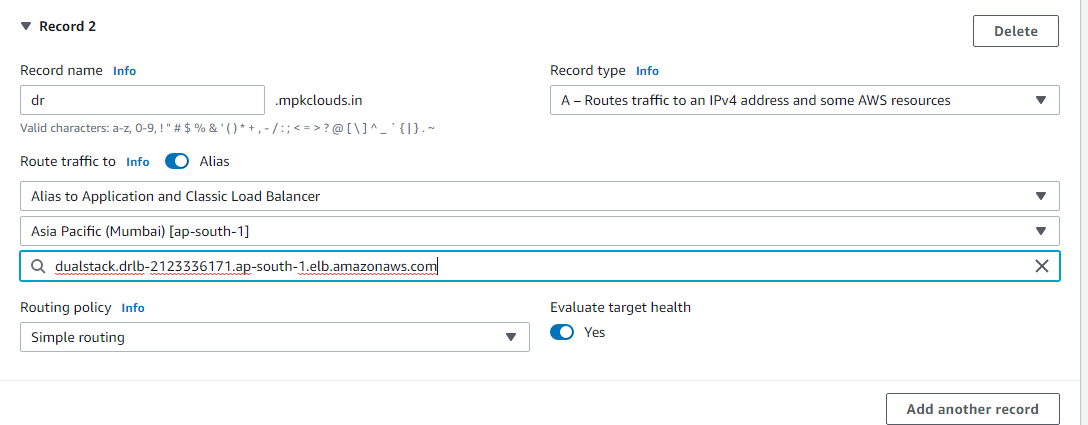
-Alias to Application and classic Load Balancer

-Asia Pacific (Mumbai) [ap-south-1]

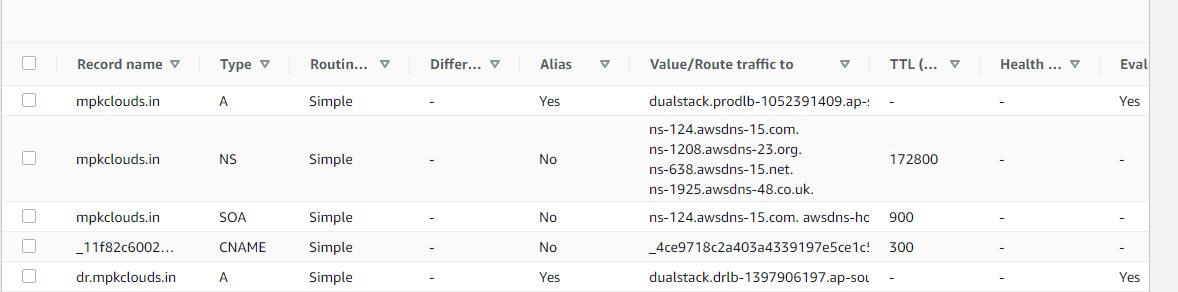
-dualstack.drlb-2123336171.ap-south-1.elb.amazonaws.com

# Route policy - simple routing.





Now five number of record set has been created.



**INSTALLING THE SSL CERTIFICATE USING ACM AND MAKING THE APP SECURED:**

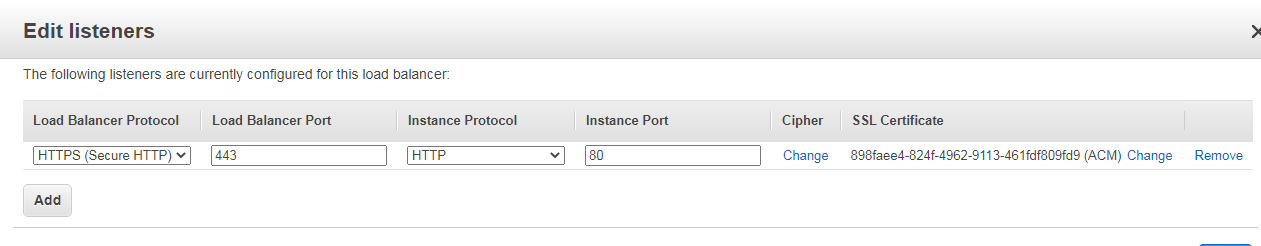
# Security will be provided in ELB. Because ELB is the entry point of the architect.

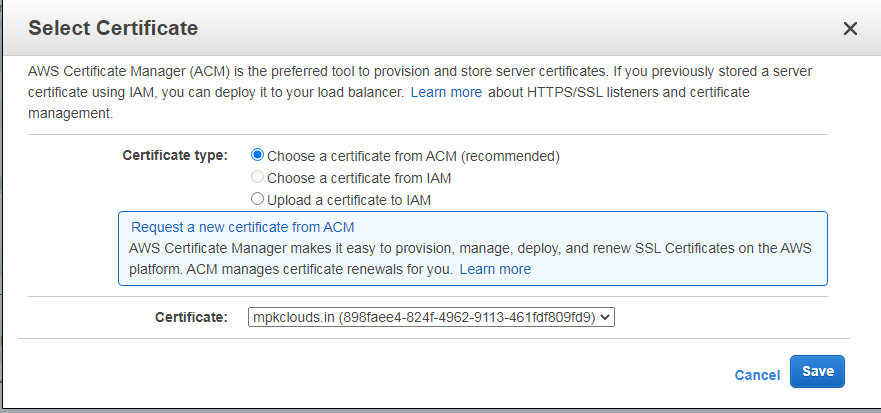
Go to production ELB - listeners – edit – add

# Load balancer protocol – https

# Load balancer – 443

# SSL certificate – Choose a certificate from ACM



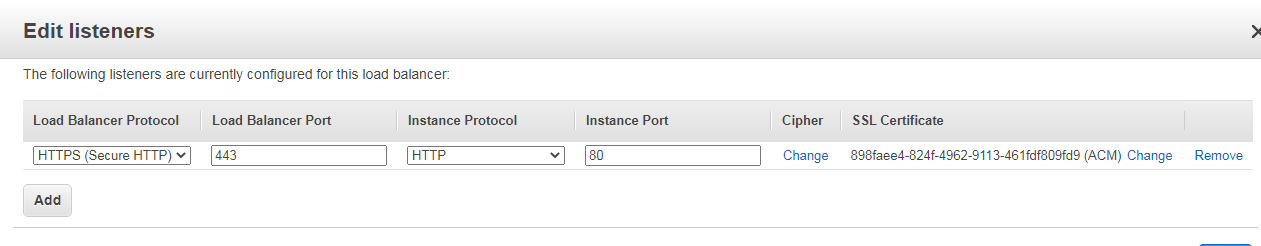


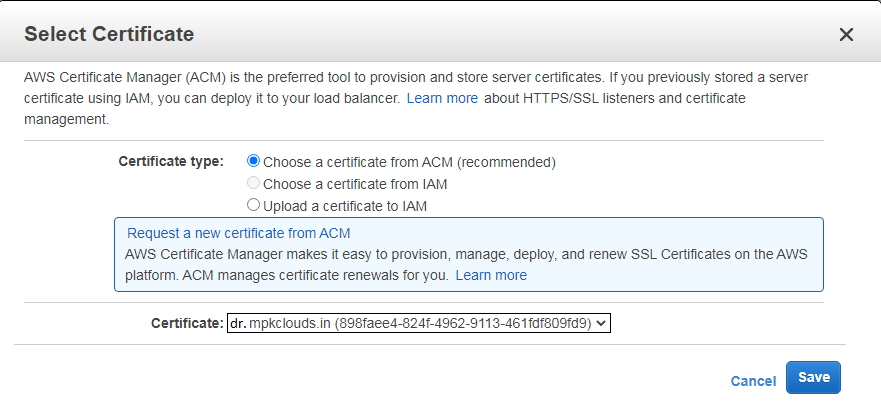
Go to dr ELB - listeners – edit – add

# Load balancer protocol – https

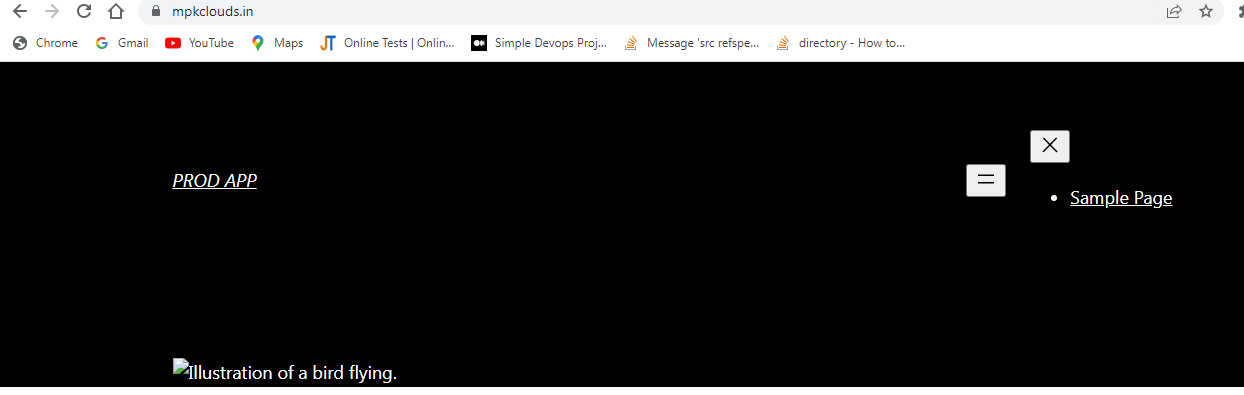
# Load balancer – 443

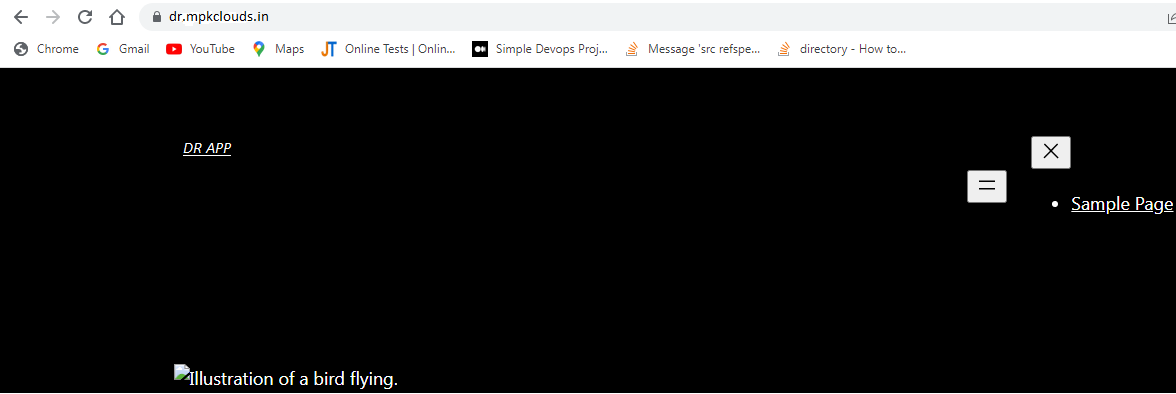
# SSL certificate – Choose a certificate from ACM





Hit the browser by following urls https://mpkclouds.in and https://dr.mpkclouds.in . It shows app is secured



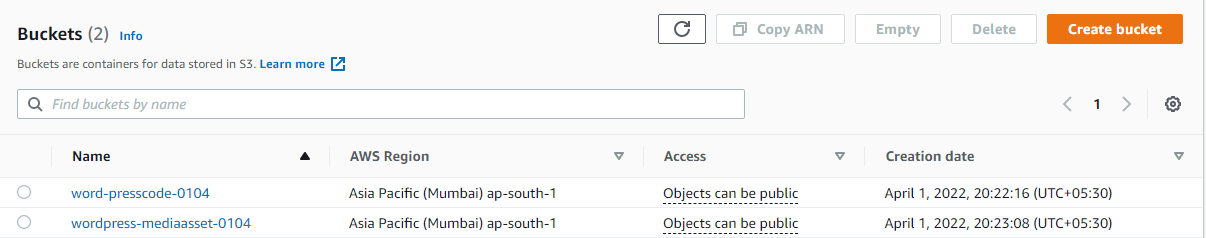


**10. AWS S3 BUCKET CREATION AND SYNCHORONIZATION OF ENVIRONMENTS (PROD&DR)**

# create two number of s3 buckets.

- 1.word-presscode-0104

- 2.wordpress-mediaasset-0104

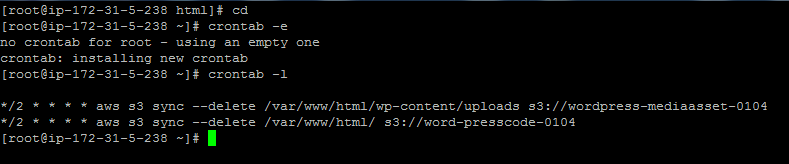


# set crontab jobs in the ec2 in some small intervals to copy the content from prod to s3 & s3 to dr,

# crontab -e

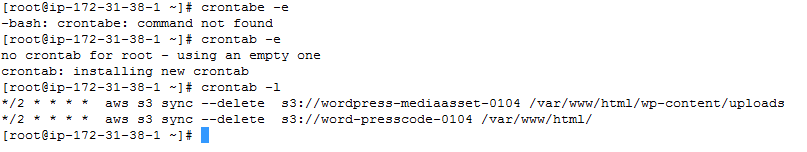
**PRODUCTION:**

* \*/2 \* \* \* \* aws s3 sync --delete /var/www/html/wp-content/uploads s3://wordpress-mediaasset-greens0811
* \*/2 \* \* \* \* aws s3 sync --delete /var/www/html/ s3://wordpress-code-greens0811



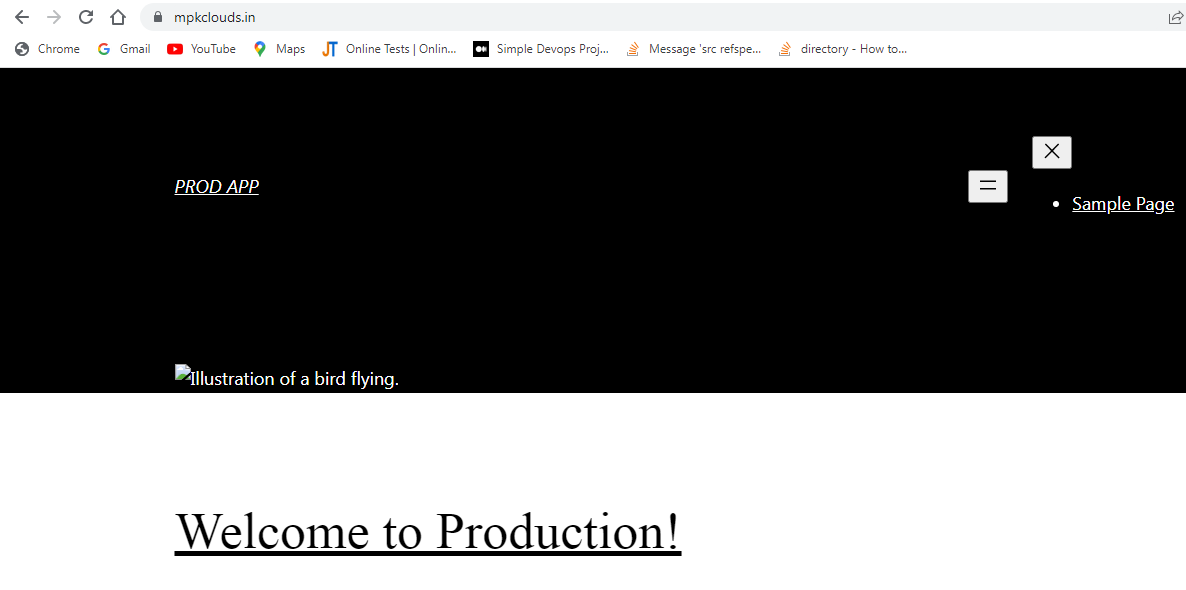
**DR:**

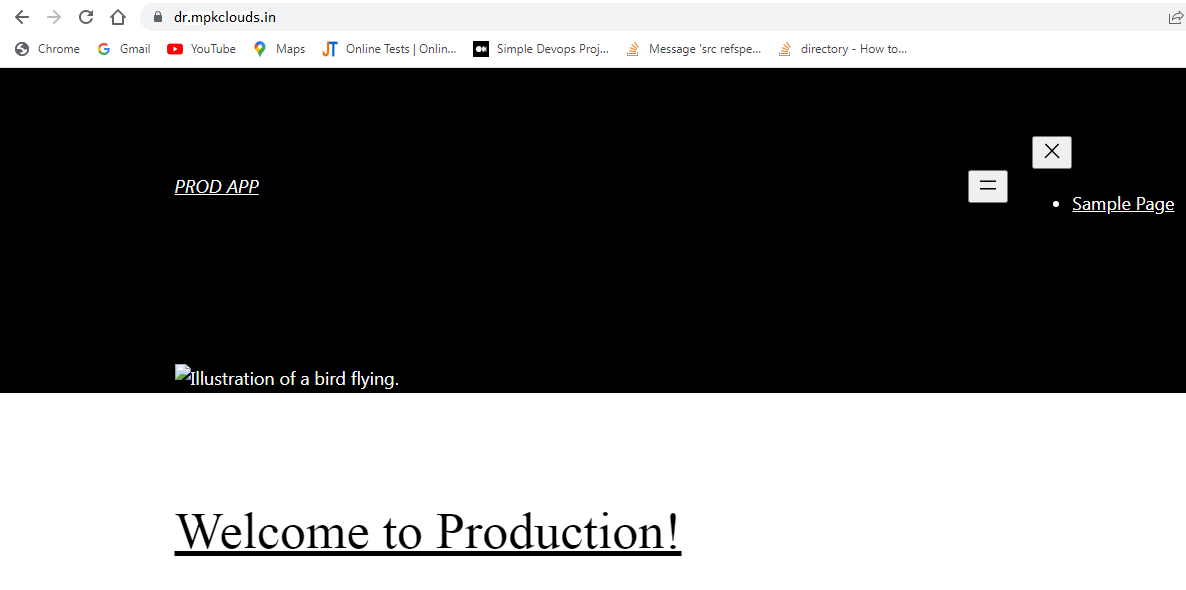
* \*/2 \* \* \* \* aws s3 sync --delete s3://wordpress-mediaasset-greens0811 /var/www/html/wp-content/uploads
* \*/2 \* \* \* \* aws s3 sync --delete s3://wordpress-code-greens0811 /var/www/html/



# Do some changes in prod server it will reflect in dr server.

# wait for few mins and check the below things,





CLOUD WATCH:

# sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86\_64

# Curl https://awscloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip -O

# unzip CloudWatchMonitoringScripts-1.2.2.zip && \rm CloudWatchMonitoringScripts-1.2.2.zip && \

# cd aws-scripts-mon

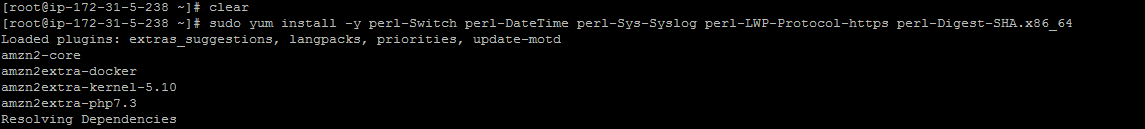
# mon-put-instance-data.pl

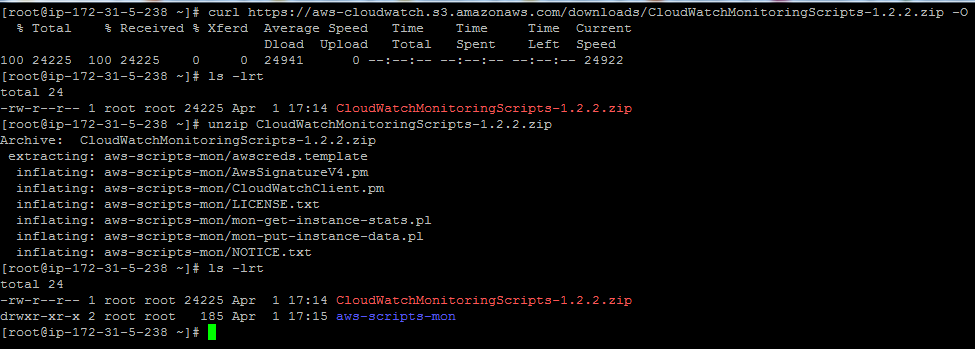
# ./mon-put-instance-data.pl --mem-util --verify –verbose

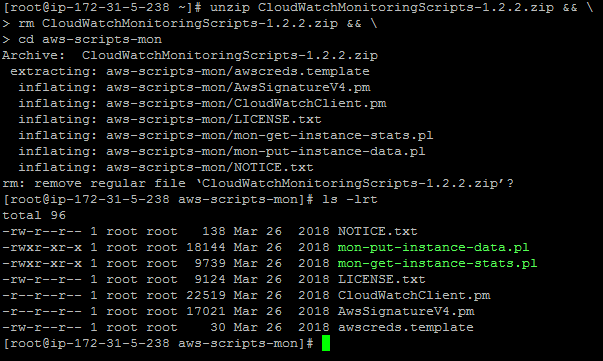
The above command is used show the memory utilization of servers. And we have to export this date from ec2 server to cloudwatch using IAM roles.

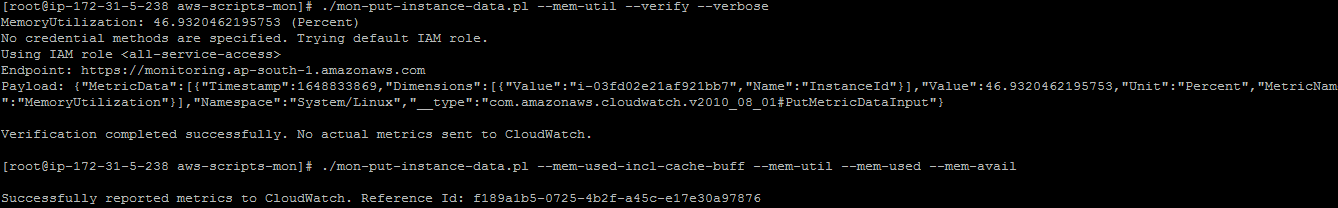
# ./mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail

The above command is used to send the data from ec2 server to cloud watch.

****







# Go to cloud watch - dashboard – create dash board – name (prod server) – Number – instance id – linux-system papmeter

# We can view memory utilization , memory available and memory used by the prod server.

