**Launching Web Application integrated with AWS RDS using Kubernetes | Terraform**

## Amazon Web Services (AWS)

Amazon Web Services offers reliable, scalable, and inexpensive cloud computing services. Free to join, pay only for what you use.

## Terraform

## Terraform enables you to safely and predictably create, change, and improve infrastructure. It is an open source tool that codifies APIs into declarative language.

## Minikube

Minikube is a tool that helps to run kubernetes locally. It helps us to create and run a single node Kubernetes cluster on top of a Virtual Machine on your own system.

## AWS RDS

**Amazon Relational Database Service** makes it easy to create, operate and also helps to scale a relational-db in the AWS cloud.This service provided by AWS is really easy to administer and with its availability and durability makes it even more reliable accompanying a high and fast performance. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups.

## Project:

 Deploy the Wordpress application on Kubernetes and AWS using terraform including the following steps;

1.  Write an Infrastructure as code using terraform, which automatically deploy the Wordpress application

2.  On AWS, use RDS service for the relational database for Wordpress application.

3. Deploy the Wordpress as a container either on top of Minikube or EKS or Fargate service on AWS

4. The Wordpress application should be accessible from the public world if deployed on AWS or through workstation if deployed on Minikube.

## Prerequisites:

To begin with the task, first we need an AWS account and we require to create an IAM account on the account. Also, we need to download the AWS CLI in our local system. With all that done we need to download Terraform as we are going to build our project using it.

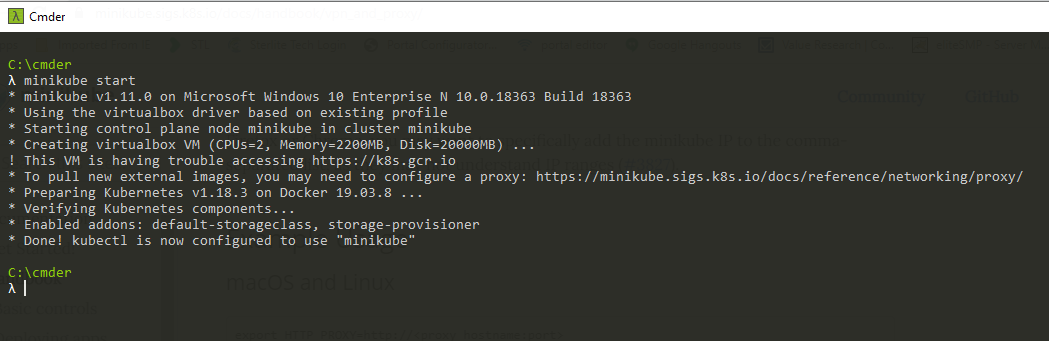
We also need to create a profile as it conceals the access and secret keys if we sometime share our code through a source code management tool. We use the following command to create a profile:

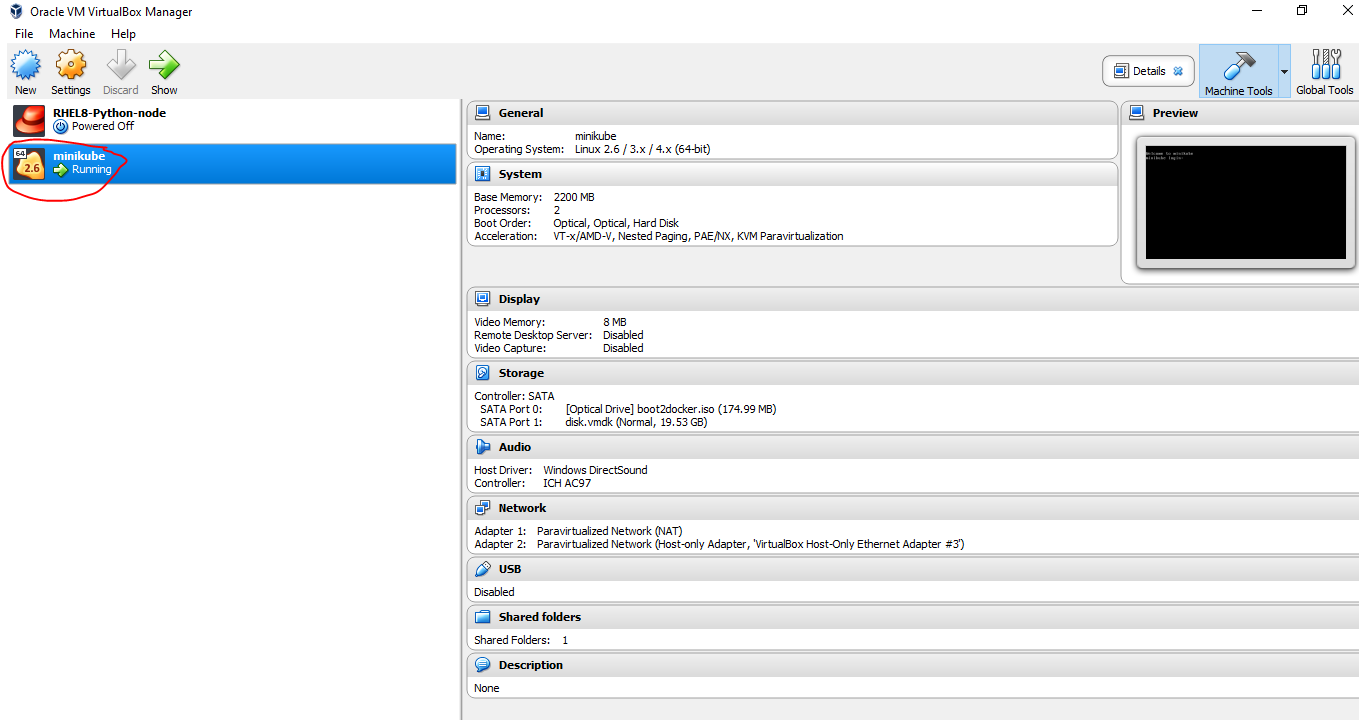
aws configure --profile 'profile-name'

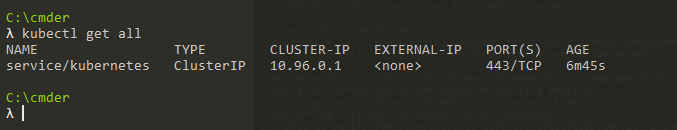
## Execution:

First we need to start the minikube by running the following command on the cmd:

minikube start







**Terraform code to deploy WordPress Application on top of Minikube Kubernetes Cluster:**

Now we need to create a code so that we can deploy our Wordpress Application on the minikube K8S cluster and for the same I’ve used the “wordpress:4.8-apache” image.I have made the use of the kubernetes deployment as Kubernetes automatically starts pods on the cluster based on our resource requirements and automatically restarts pods if they or the instances they are running on fail.The 3 replicas i have created will control the traffic and would automatically relaunch if any goes down.

provider "kubernetes" {

    config\_context\_cluster   = "minikube"

}

resource "kubernetes\_deployment" "WordPress" {

  metadata {

    name = "wordpress"

  }

  spec {

    replicas = 3

    selector{

    match\_labels = {

      dc = "IN"

      env = "Testing"

      App = "wordpress"

    }

    match\_expressions {

      key = "env"

      operator = "In"

      values = ["Testing" , "wordpress"]

    }

  }

   template {

        metadata {

         labels = {

      dc = "IN"

      env = "Testing"

      App = "wordpress"

    }

        }

      spec {

        container {

          image = "wordpress:4.8-apache"

          name  = "wpress"

        }

      }

    }

}

}

resource "kubernetes\_service" "service" {

  metadata {

    name = "loadbalancer"

  }

  spec {

    selector = {

      App = kubernetes\_deployment.WordPress.spec.0.template.0.metadata[0].labels.App

    }

    port {

      node\_port   = 30003

      port        = 80

      target\_port = 80

    }

    type = "NodePort"

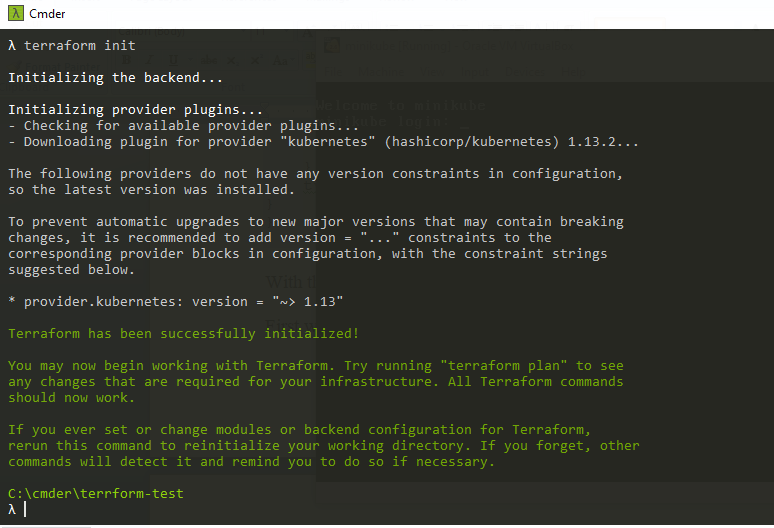
}

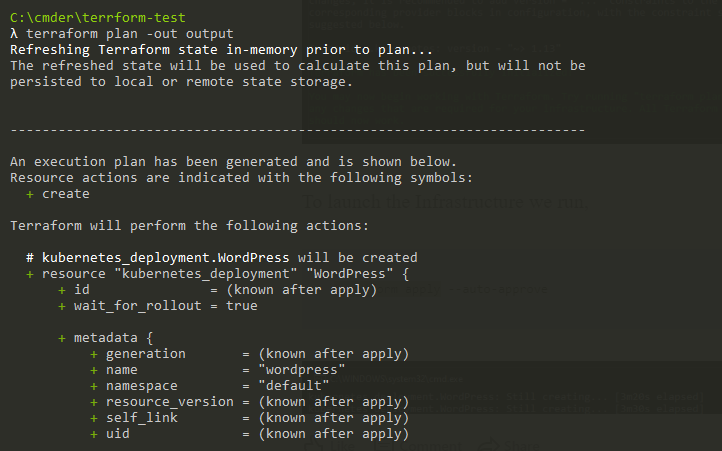
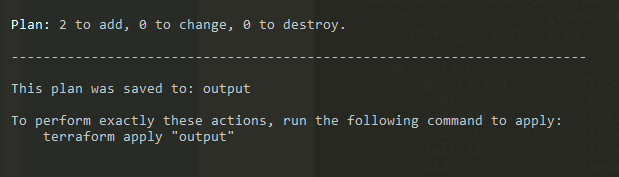
}

With the creation of the above now we will run the above code as follows:

First we run the following command for terraform to install the plugins:

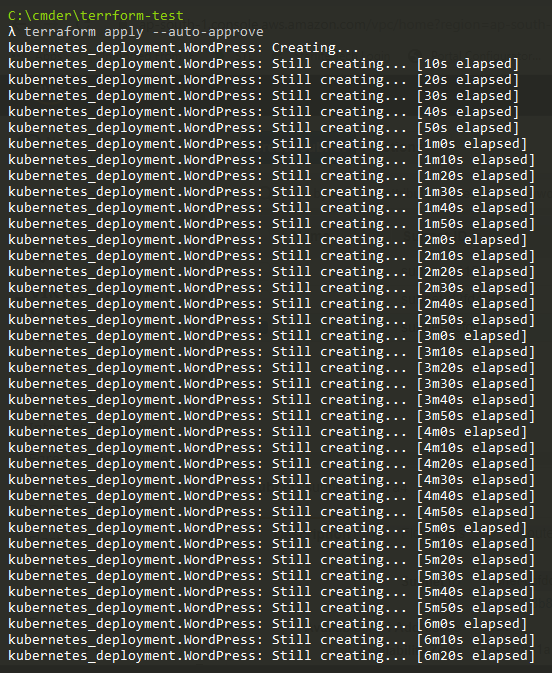
terraform init

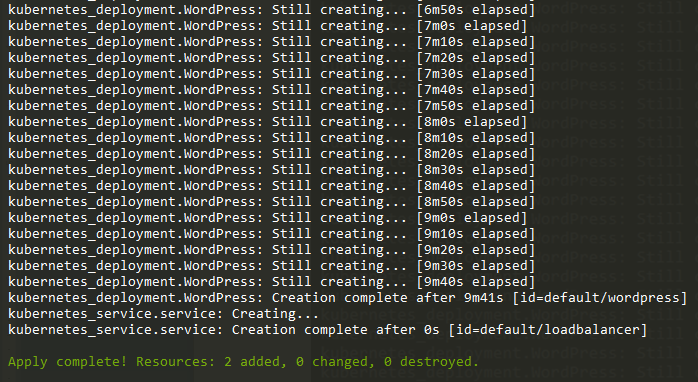


To launch the Infrastructure we run,

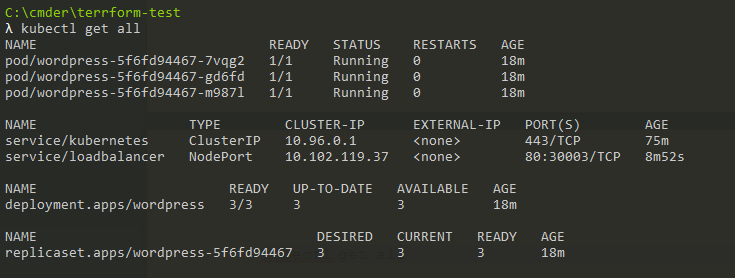
terraform apply --auto-approve





To check if the Kubernetes cluster we created has deployed we can run the following command:

kubectl get all

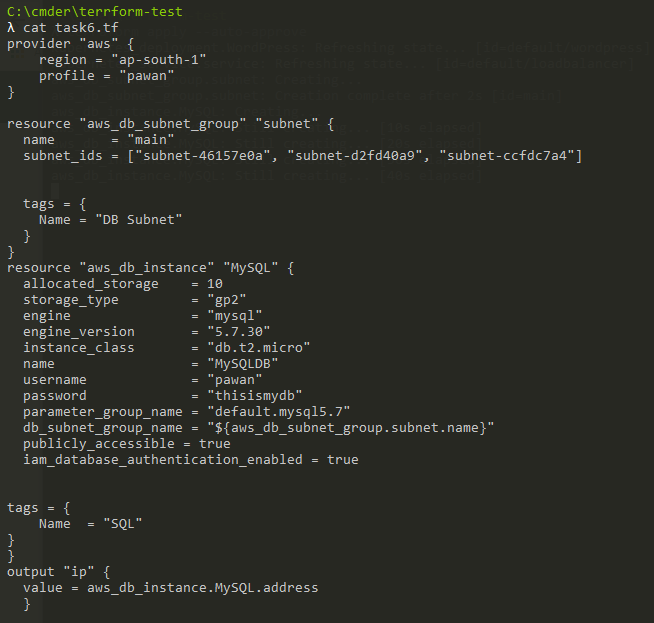


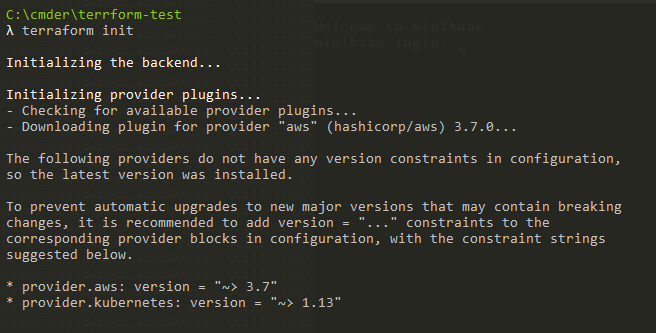
**Terraform code for AWS-RDS:**

**MySQL Database**

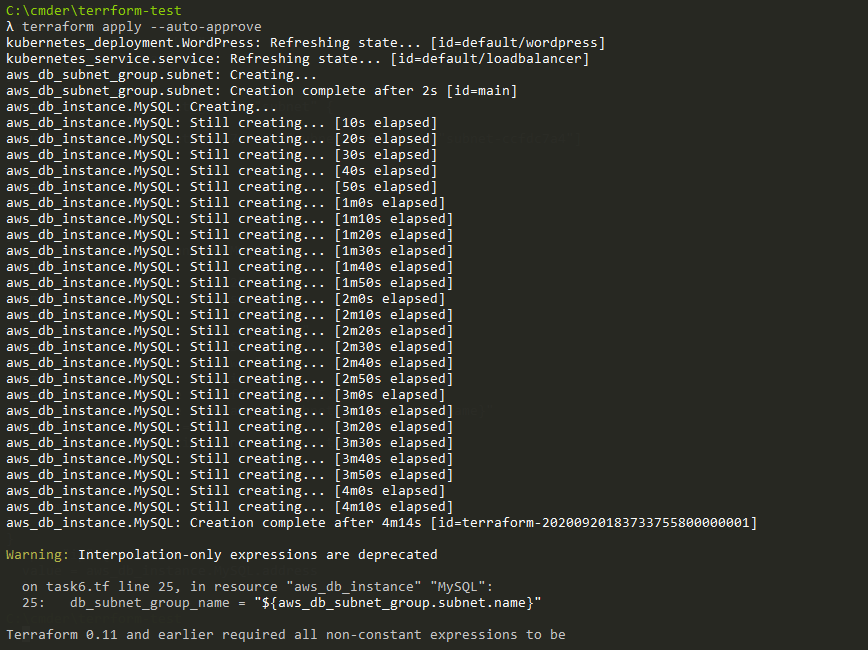
The Database As a service(DAAS) provided by AWS is AWS-RDS and which would hence help us to manage the database for our Web application.

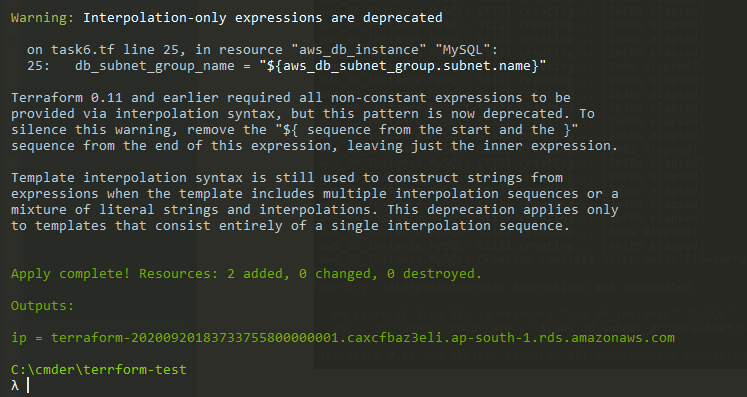
Now we need to run the with the same routine as above and the IP we get as the output would be the database host which would later help us to login into the WordPress.

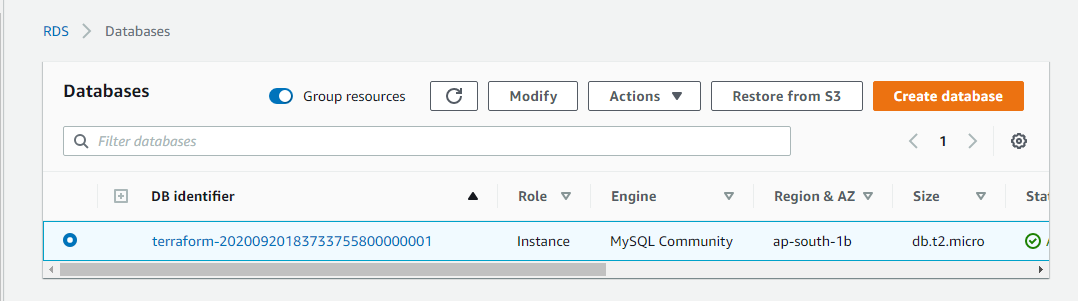


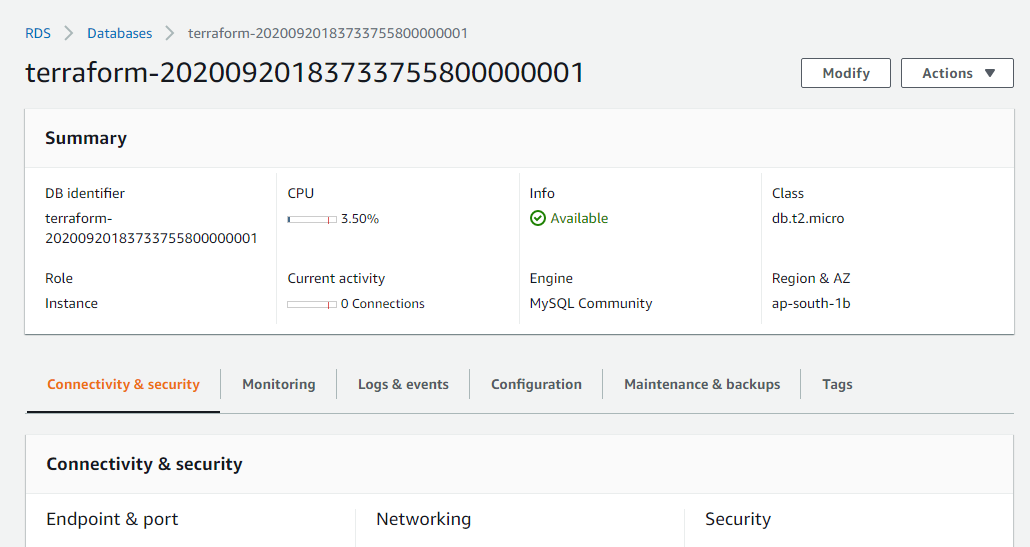


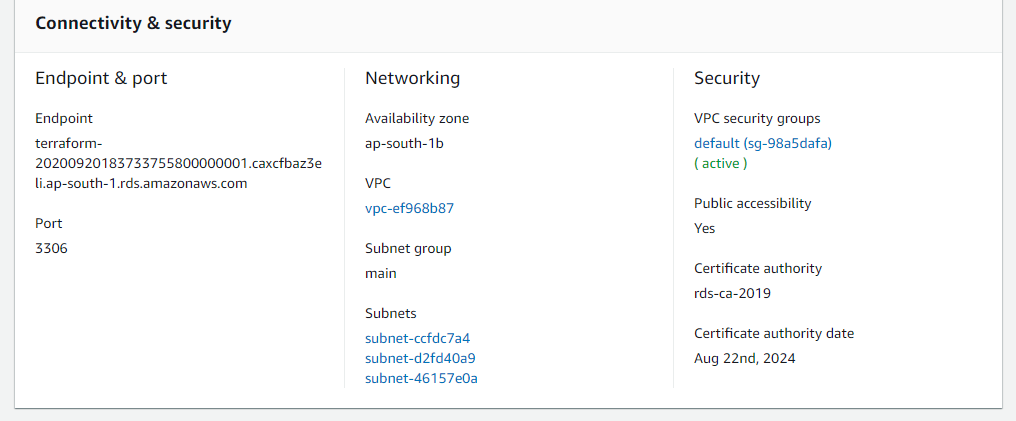
Now we need to run the with the same routine as above and the IP we get as the output would be the database host which would later help us to login into the WordPress.

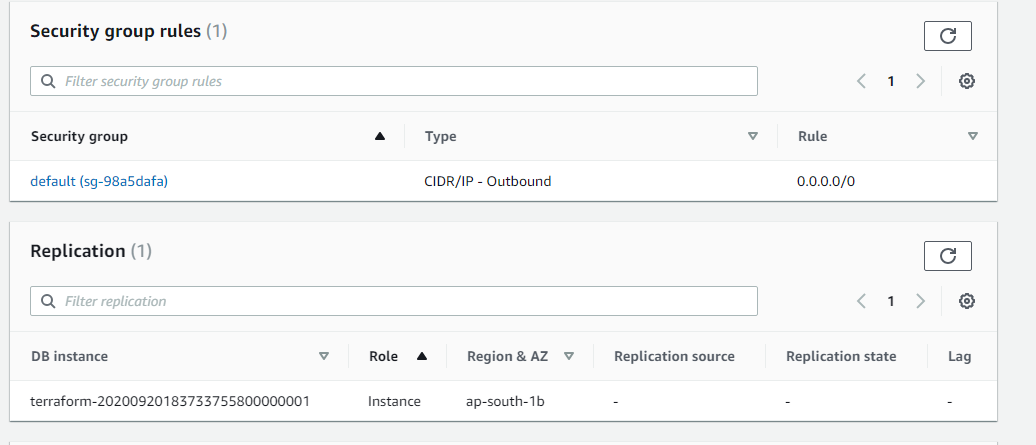






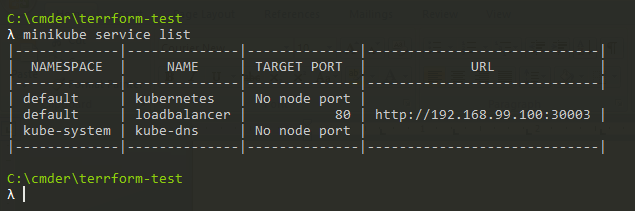




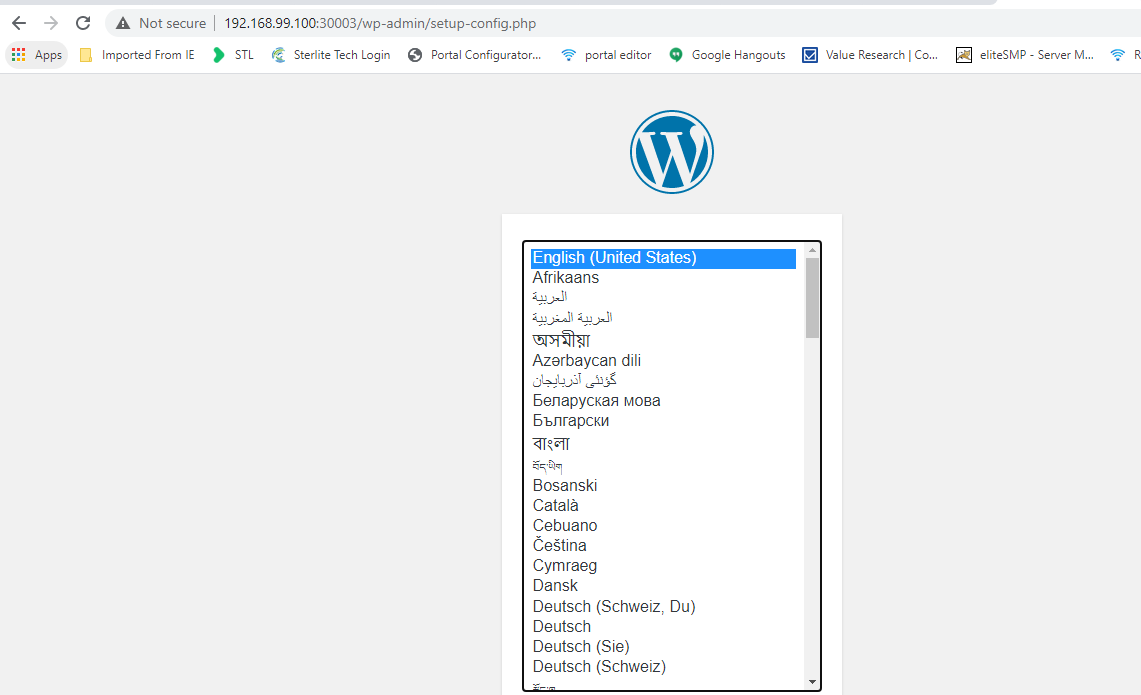


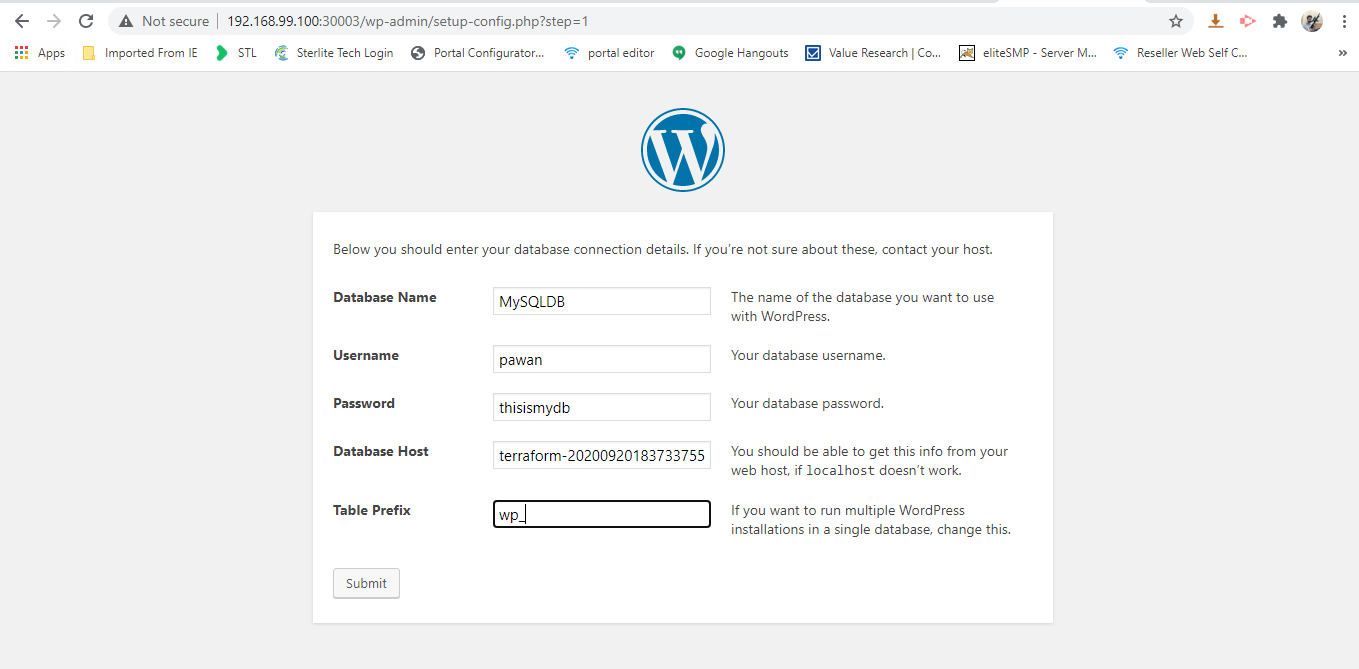
To get the Service URL from the minikube we can run the below command:

minikube service list



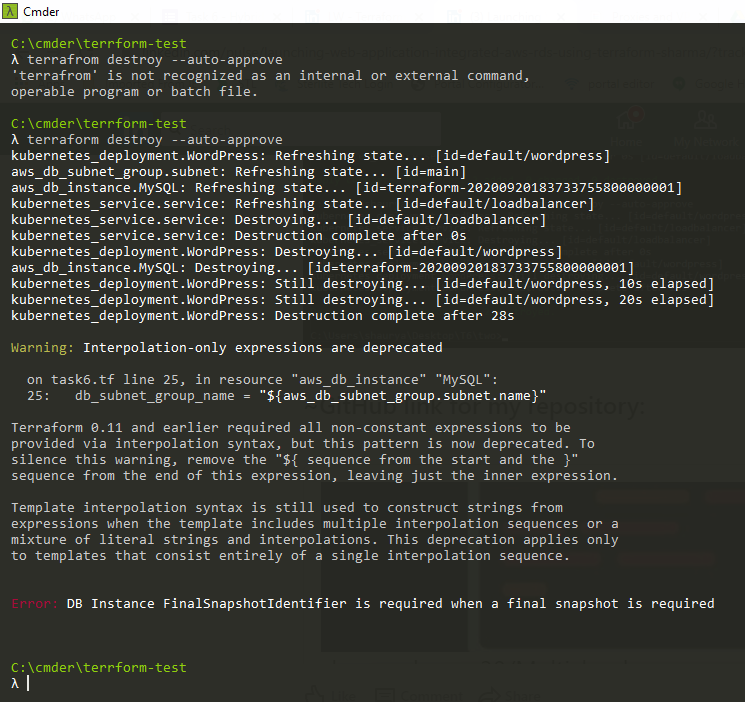
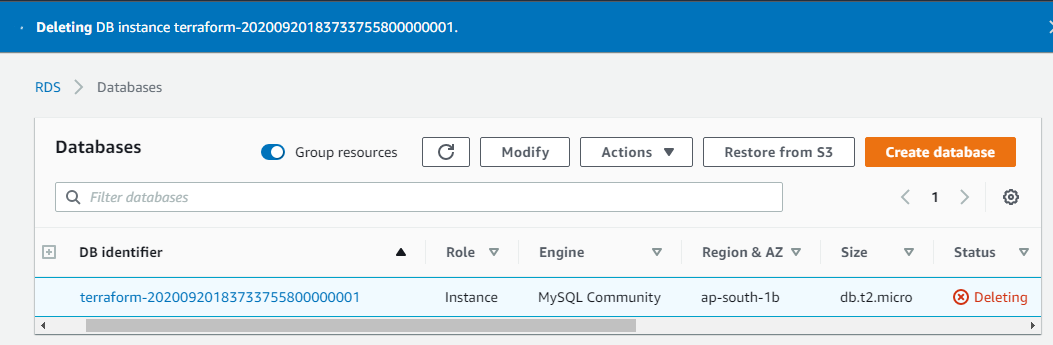
and here we go !!!





To delete the whole Infrastructure we run the following command:

terraform destroy --auto-approve

My github link :- <https://github.com/pawankulaura/hybridmulticloud-task6>

This is my project under the guidance of the World Record Holder & my mentor **Mr. Vimal Daga**, in which I developed a terraform code to launch a WordPress Web Application integrated with Amazon RDS on top of Kubernetes.

Thank you so much for your time !!!