**Deploy Webpage Using MySQL (AWS RDS)**

**What is Amazon RDS ?**

Amazon RDS is a service which provides database connectivity through the Internet. RDS makes it very simple and easy to set-up a relational database in the cloud.

Instead of concentrating on database features, you can concentrate more on the application to provide high availability, security, and compatibility. RDS is a fully managed RDBMS service.

**Benefits of Amazon RDS**



Database Engines

There are six database engines which RDS provides, and they are:

* Amazon Aurora
* PostgreSQL
* MySQL
* MariaDB
* Oracle Database
* Microsoft SQL Server

***Task Overview***

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 the 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 .

***Task Description***

Let’s begin the task !

Here I am going to write a terraform code to create mysql database using RDS service of aws and wordpress application on the top of minikube . So there are some prerequisite for this ..

1. Installation of AWS CLI in our OS . Here I am going to use Windows as my base OS .
2. Create an IAM user and a profile

Here is my article in which I explain how to create an IAM user account and profile

**[Web Portal Using Personal VPC](https://www.linkedin.com/pulse/web-portal-using-personal-vpc-rupali-gurjar" \t "_blank)**

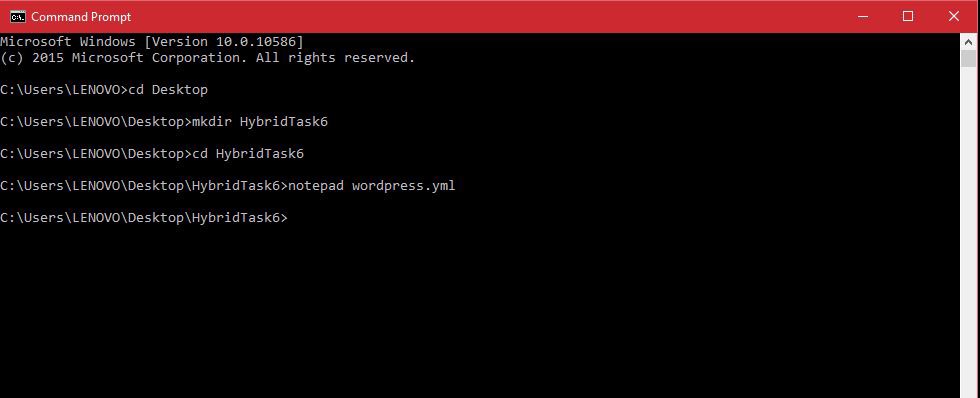
[VIRTUAL PRIVATE CLOUD ( VPC ) A virtual private cloud (VPC) is an on-demand configurable pool of shared computing…](https://www.linkedin.com/pulse/web-portal-using-personal-vpc-rupali-gurjar" \t "_blank)

[www.linkedin.com](https://www.linkedin.com/pulse/web-portal-using-personal-vpc-rupali-gurjar" \t "_blank)

3. Here I am going to deploy wordpress on the top of minikube so first we have to start it



It is always a good practice to create a separate workspace so that we can easily manage the things .



\

***Step 1 :****Write an Infrastructure as code using terraform which automatically deploy the Wordpress Application .*

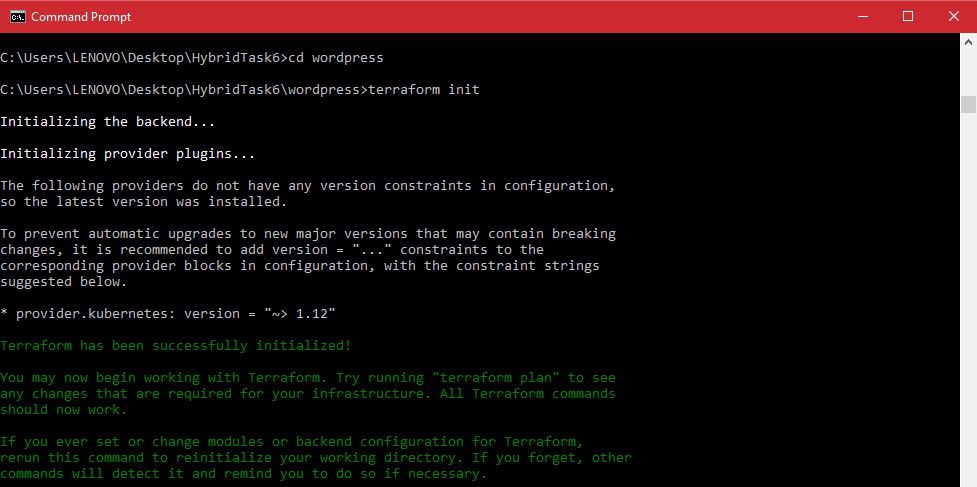
Here is my terraform code

provider "kubernetes" {  
 config\_context\_cluster = "minikube"  
}resource "kubernetes\_service" "service" {  
 metadata {  
 name = "wordpress"  
 }  
 spec {  
 selector = {  
 app = "wordpress"  
 }  
 session\_affinity = "ClientIP"  
 port {  
 port = 80  
 target\_port = 80  
 node\_port = 30100  
 }type = "NodePort"  
 }  
}resource "kubernetes\_deployment" "deployment" {  
 metadata {  
 name = "wordpress"  
 labels = {  
 app = "wordpress"  
 }  
 }spec {  
 replicas = 3selector {  
 match\_labels = {  
 app = "wordpress"  
 }  
 }template {  
 metadata {  
 labels = {  
 app = "wordpress"  
 }  
 }spec {  
 container {  
 image = "wordpress"  
 name = "wordpress"  
 }  
 }  
 }  
 }  
}

Here I create 3 replicas of wordpress pod and to access it from the outside world , I create a service . It will expose the wordpress service on the port no “30100" . This port is known as Nodeport .

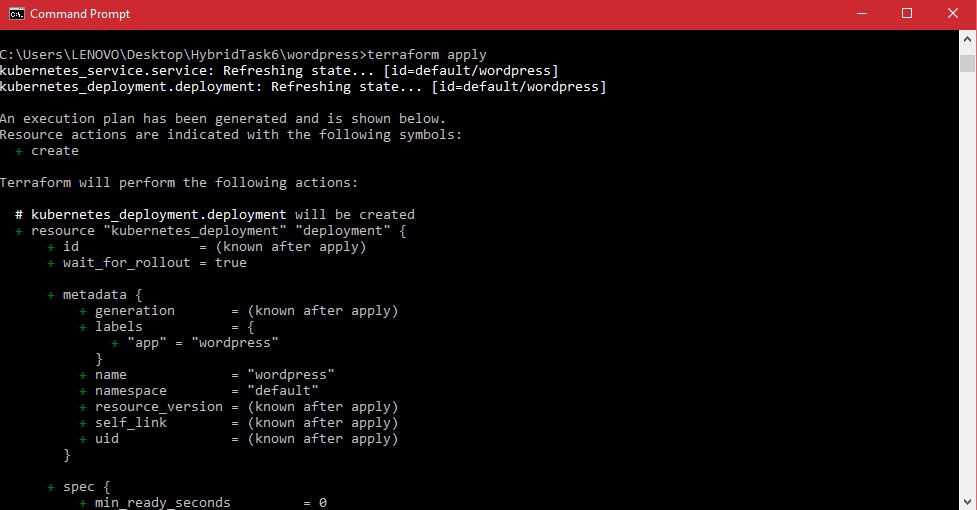
Before running this code , we have to run this command . It will download all the required plugins .

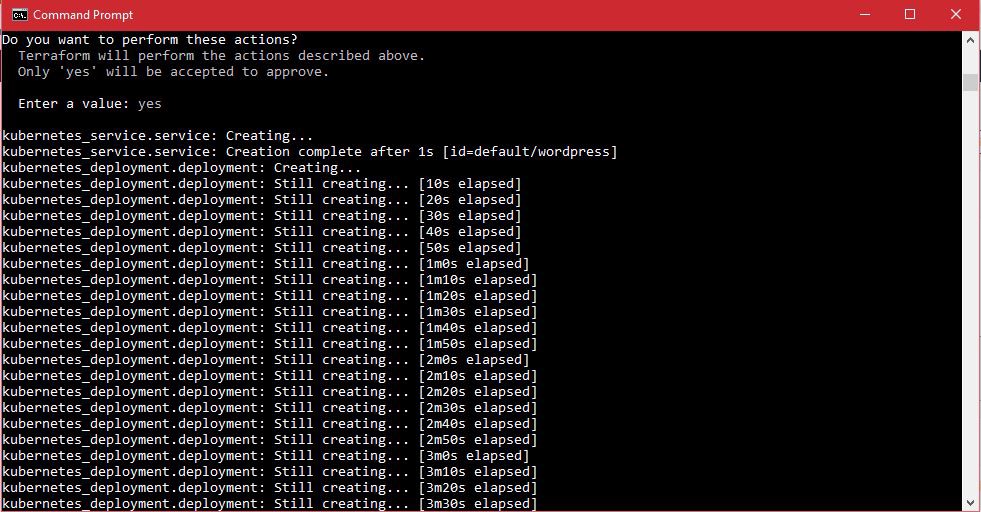
terraform init

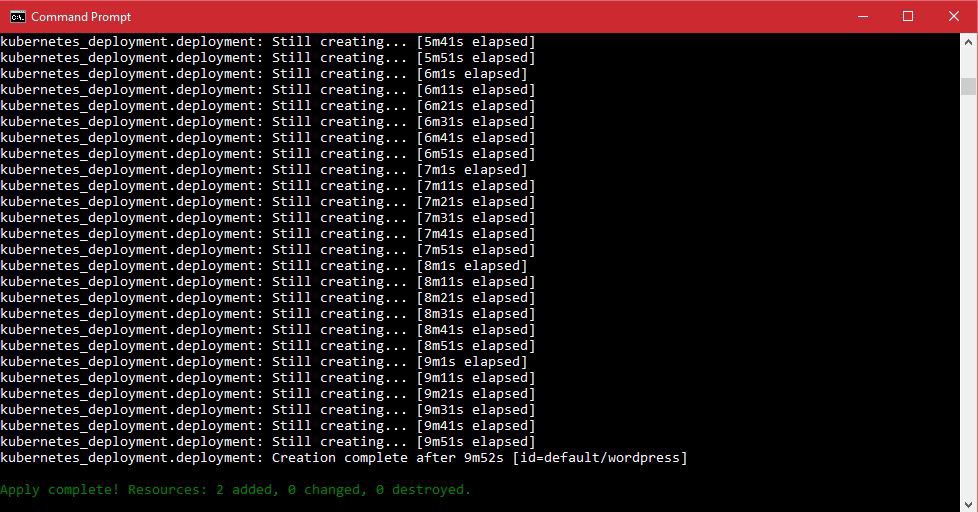


Now apply this code

terraform apply



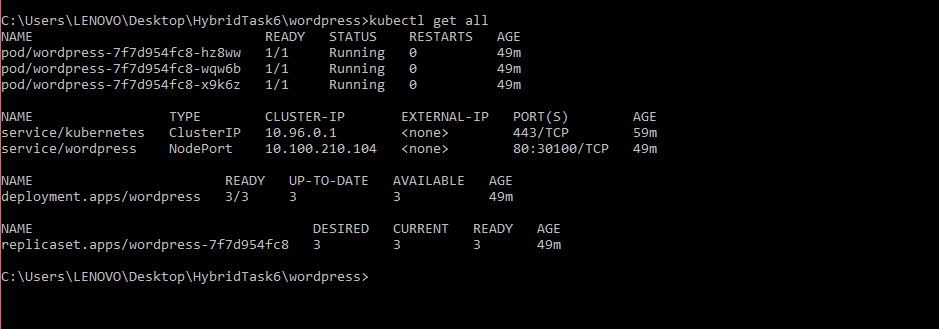




But here we have to write ‘yes’ manually so instead of using the above command, we run this command.

***“terraform apply -auto-approve”***

Here we can see ..



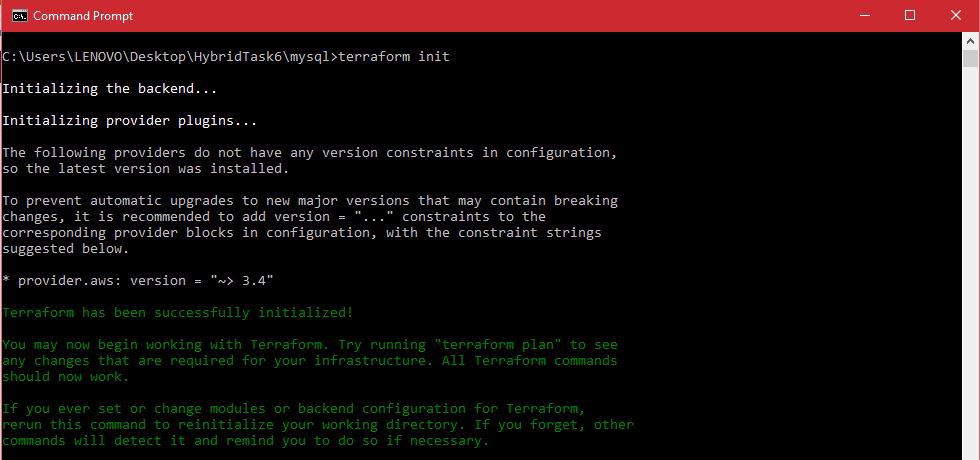
***Step 2:****Write a terraform code which automatically create a MySQL database using RDS*

Here is my terraform file of mysql

provider "aws" {  
 region = "ap-south-1"  
 profile = "Rupali"  
}resource "aws\_db\_instance" "mysql" {  
 allocated\_storage = 20  
 storage\_type = "gp2"  
 engine = "mysql"  
 engine\_version = "5.7.30"  
 identifier = "rups-database"  
 instance\_class = "db.t2.micro"  
 name = "rupali08database"  
 username = "admin"  
 password = "rupali04"  
 parameter\_group\_name = "default.mysql5.7"  
 iam\_database\_authentication\_enabled = true  
 publicly\_accessible = true  
 skip\_final\_snapshot = true  
}

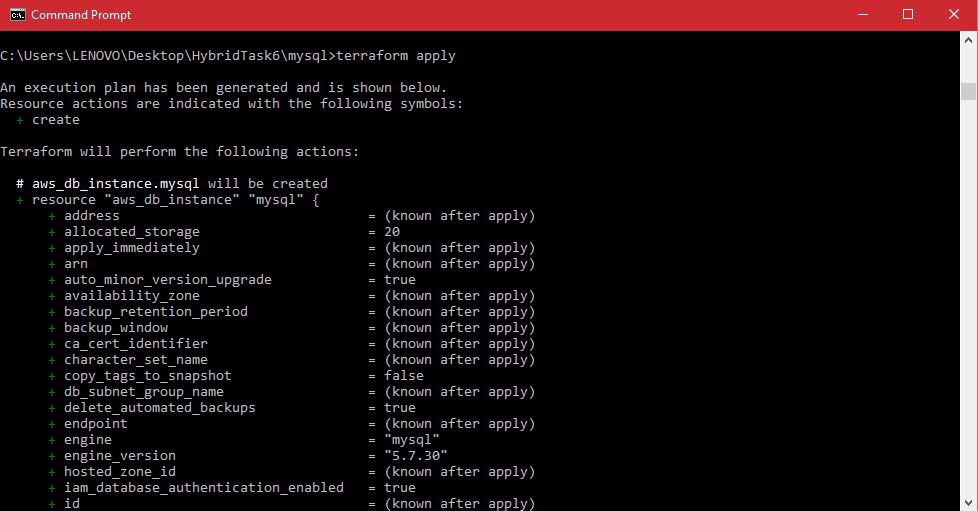
Here I used profile “Rupali” that I have created using IAM user account .In this file I have set user name , password and database name .

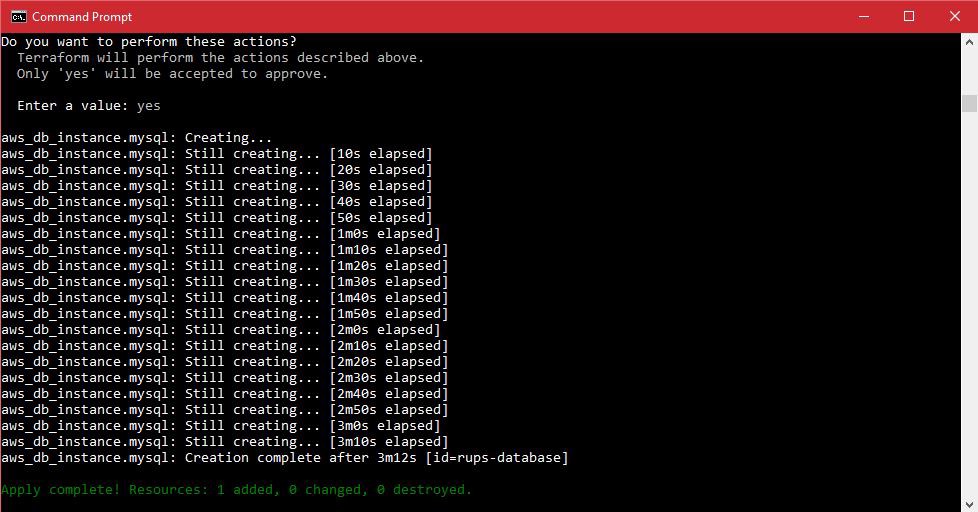
First Initialize the plugins then apply this code ..



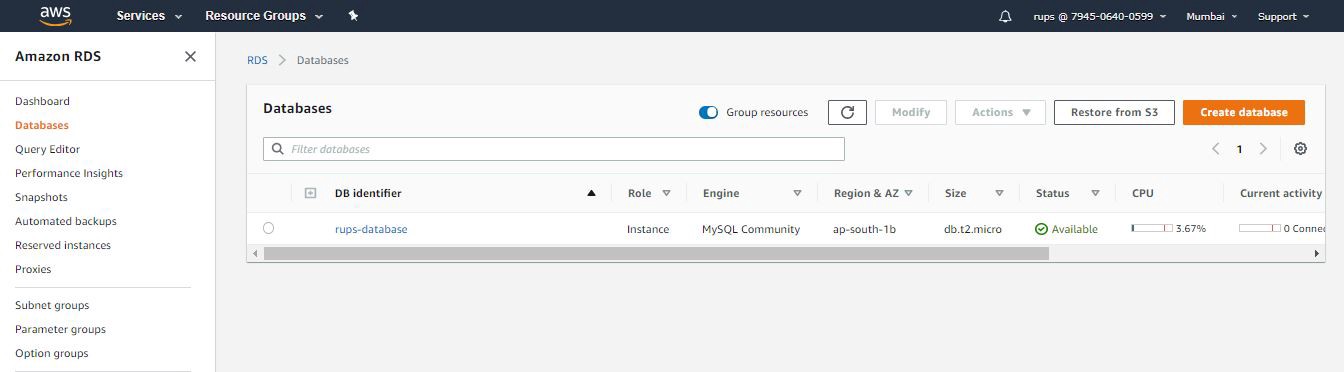
**terraform apply -auto-approve**

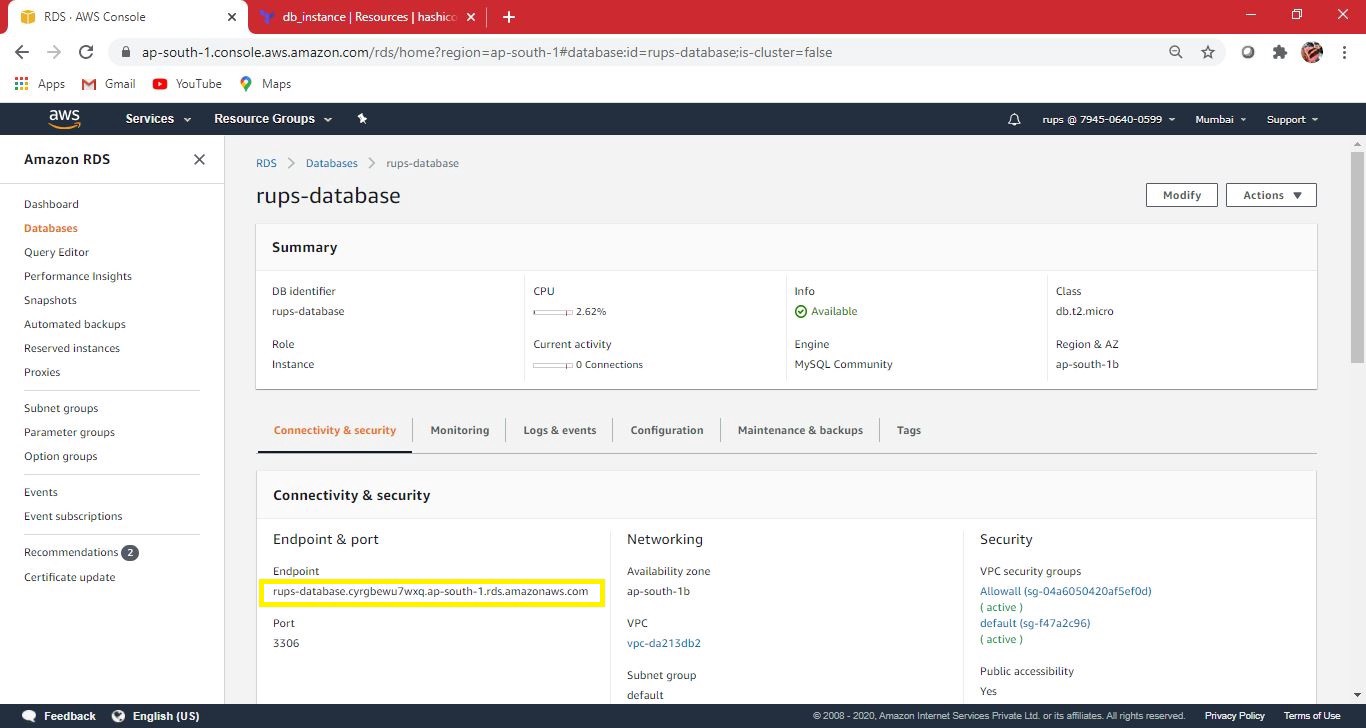
https://miro.medium.com/max/60/1*y2G8sbFoY8BG9kLIKDxMZw.jpeg?q=20



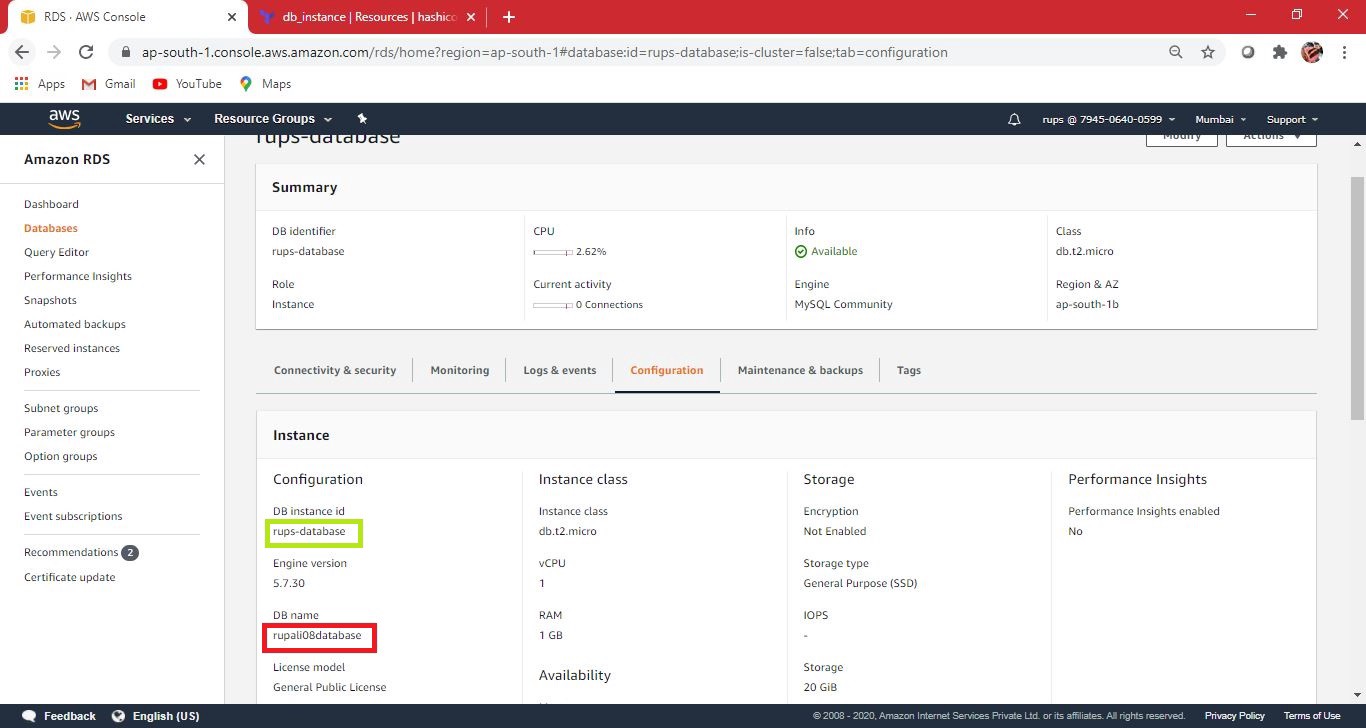


It is successfully created , we can also see it here

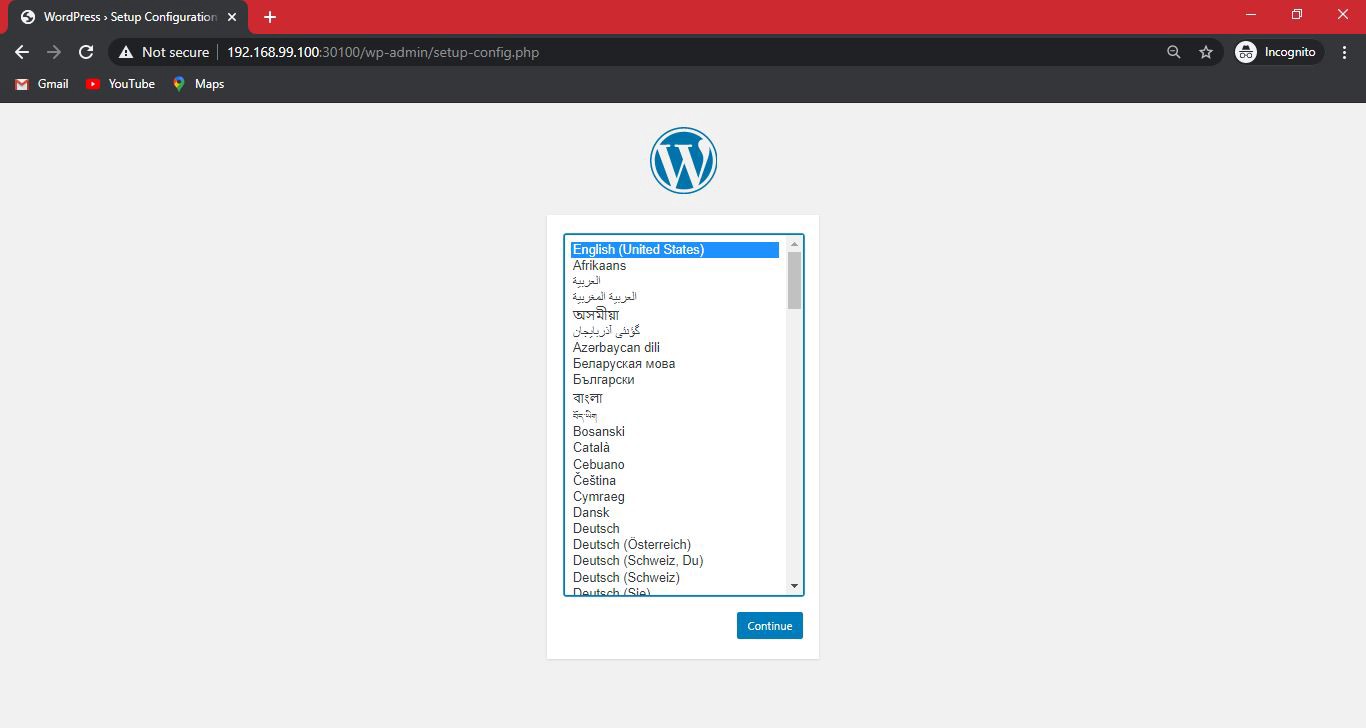


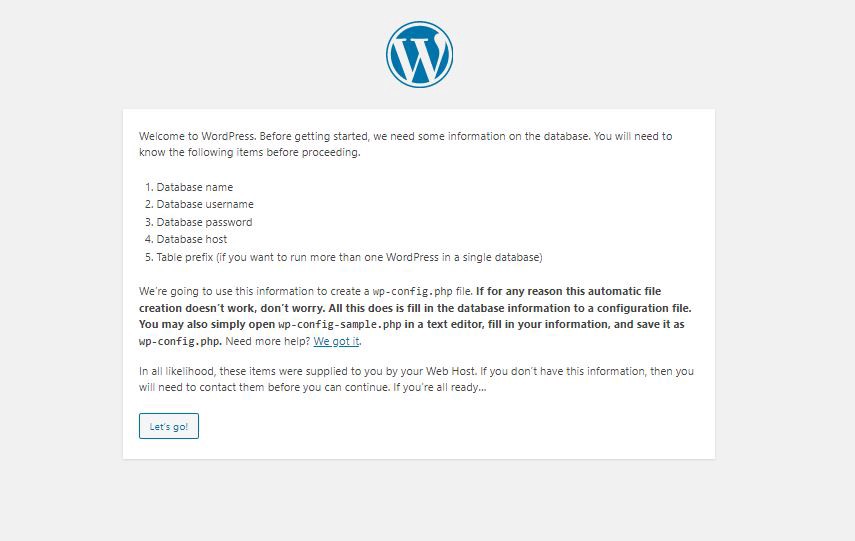


**Note :** we will use this endpoint as database host .

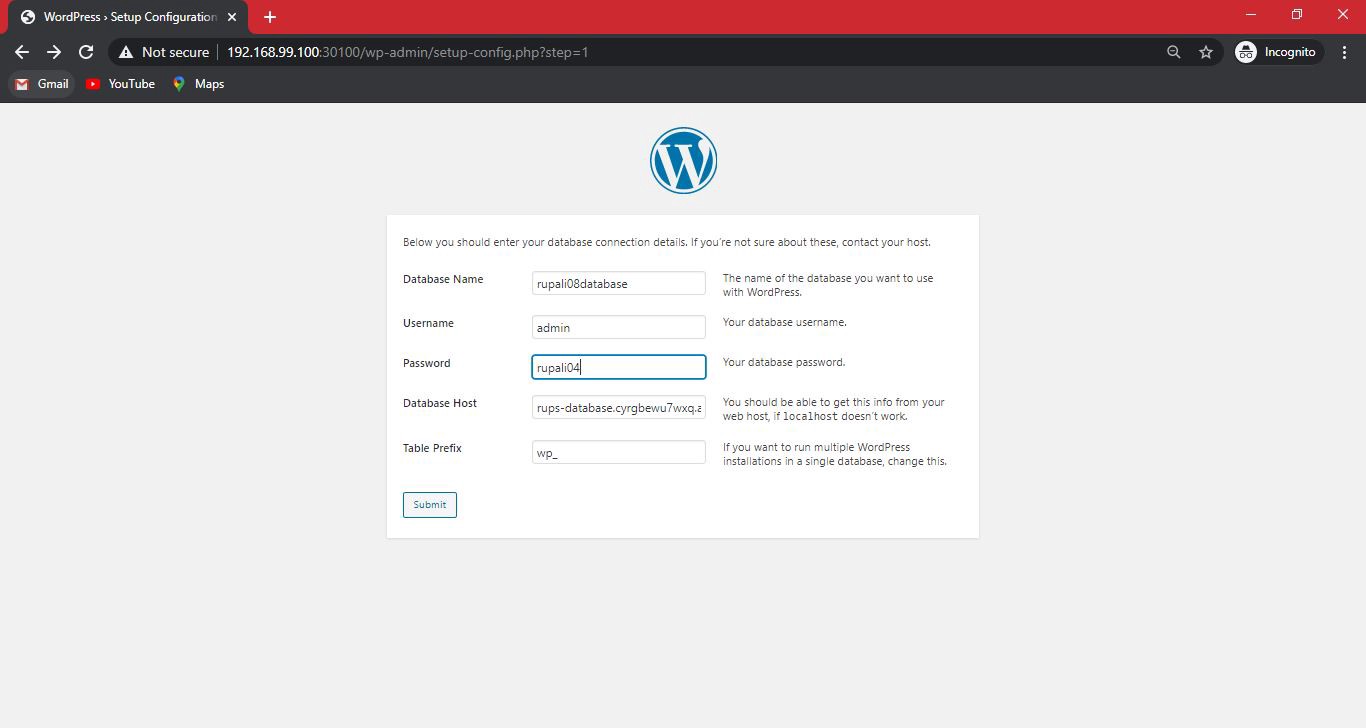


Now connect wordpress to mysql database

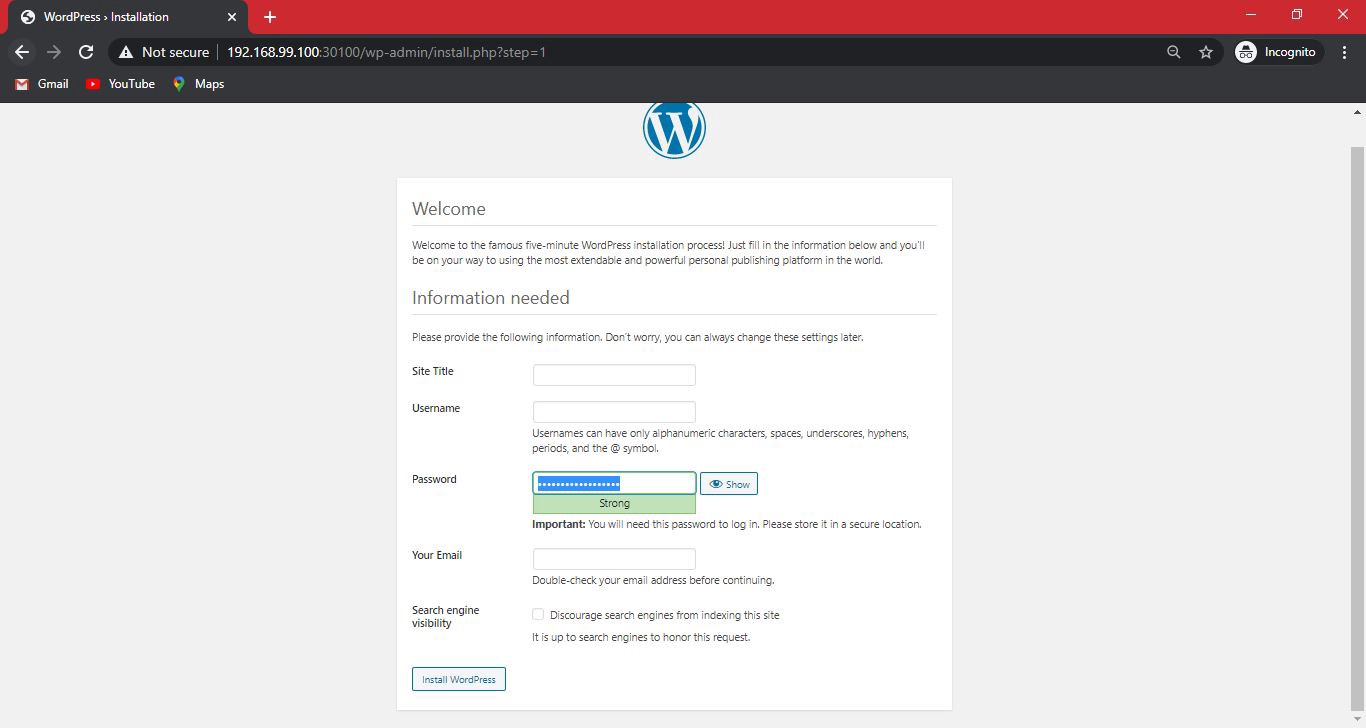


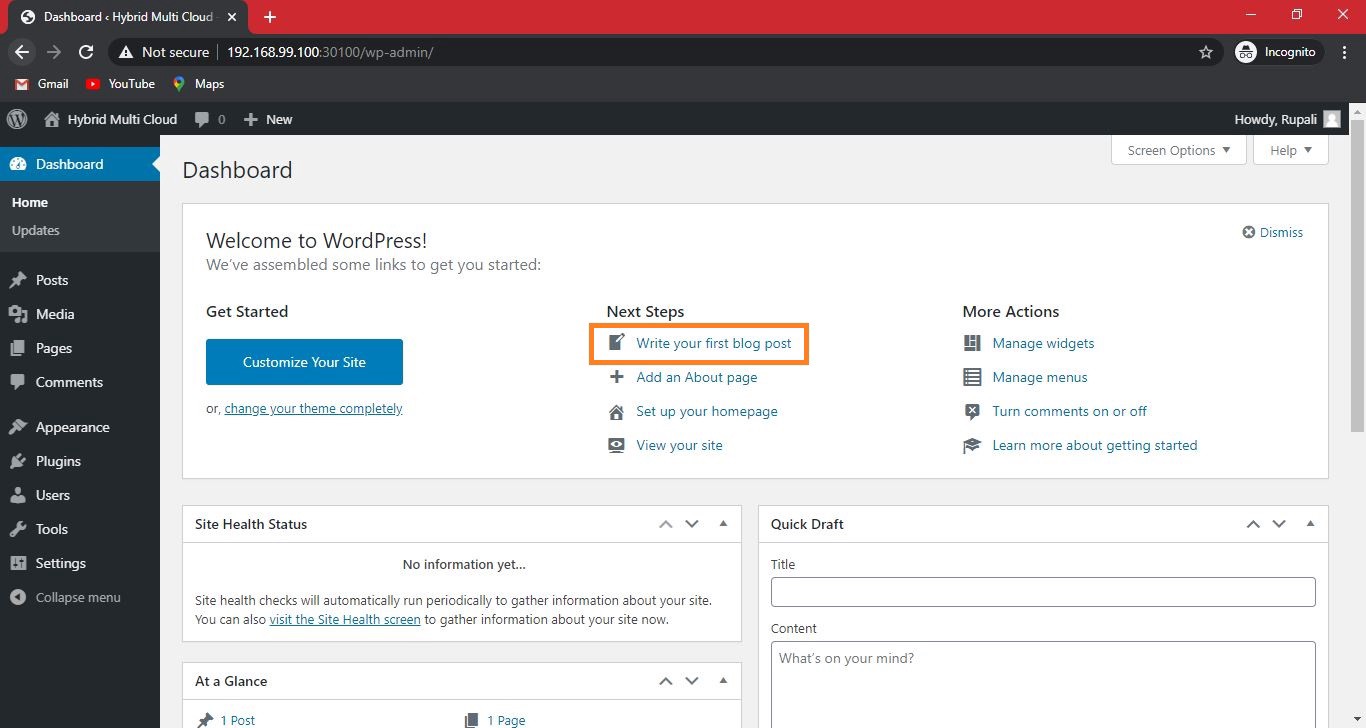


Here we have to fill all the details of database



Now set user name and password for wordpress application



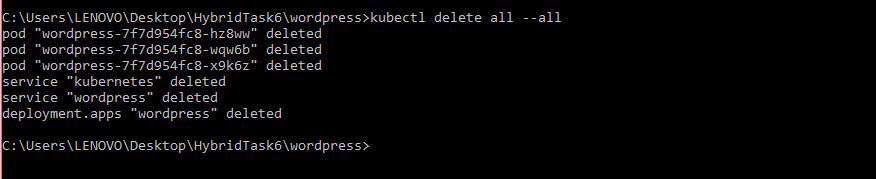


Here is the final output : )

Now just use one single command to destroy this setup

for wordpress :-

kubectl delete all --all



for mysql :-

**terraform destroy** **-auto-approve**

