

GOOGLE CLOUD PLATFORM(GCP)

\*\*Deployment of WordPress Web Application by Integrating Google Cloud Platform with Kubernetes\*\*

**>>What is GCP?**

* Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning.Registration requires a credit card or bank account details.
* Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.
* GCP consists variety of services like GCE, VPC,GAE,IAM,GKE and many more.

**>>Some basic functions of GCP:-**

* **Google Compute Engine,** which is an infrastructure-as-a-service (***IAAS***) offering that provides users with virtual machine instances for workload hosting.
* **Google App Engine,** which is a platform-as-a-service (***PASS***) offering that gives software developers access to Google's scalable hosting. Developers can also use a software developer kit (***SDK***) to develop software products that run on App Engine.
* **Google Cloud Storage,** which is a cloud storage platform designed to store large, unstructured data sets. Google also offers database storage options, including Cloud Datastore for***NoSQL*** nonrelational storage, Cloud SQL for ***MySQL***fully relational storage and Google's native Cloud Bigtable database.
* **Google Container Engine,**which is a management and orchestration system for ***Docker*** containers that runs within Google's public cloud. Google Container Engine is based on the Google ***Kubernetes*** container orchestration engine.

### ****Task description:-****

1. Create multiple projects namely developer and production

2. Create VPC network for both the projects

3. Create a link between both the VPC networks using VPC Peering

4. Create a Kubernetes Cluster in developer project and launch any web application with the Load balancer

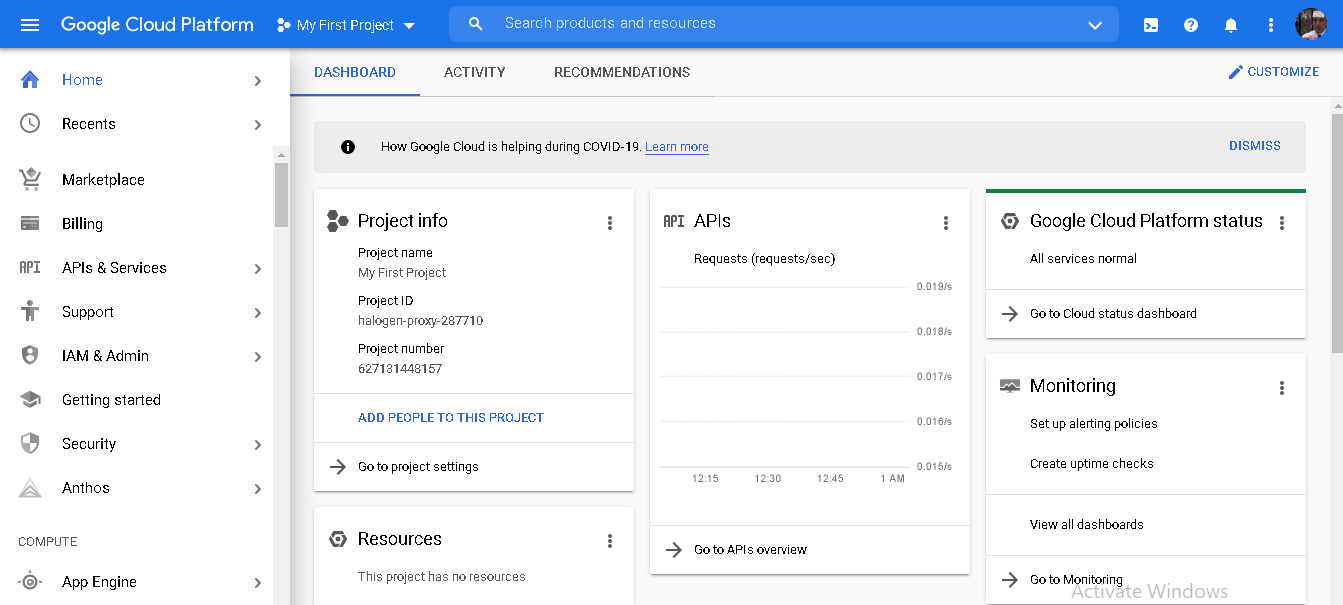
5. Create a SQL server in the production project and create a database

6. Connect the SQL database to the web application launched in the Kubernetes cluster

Let’s begin with the

Project:-

\*This is the google cloud console –

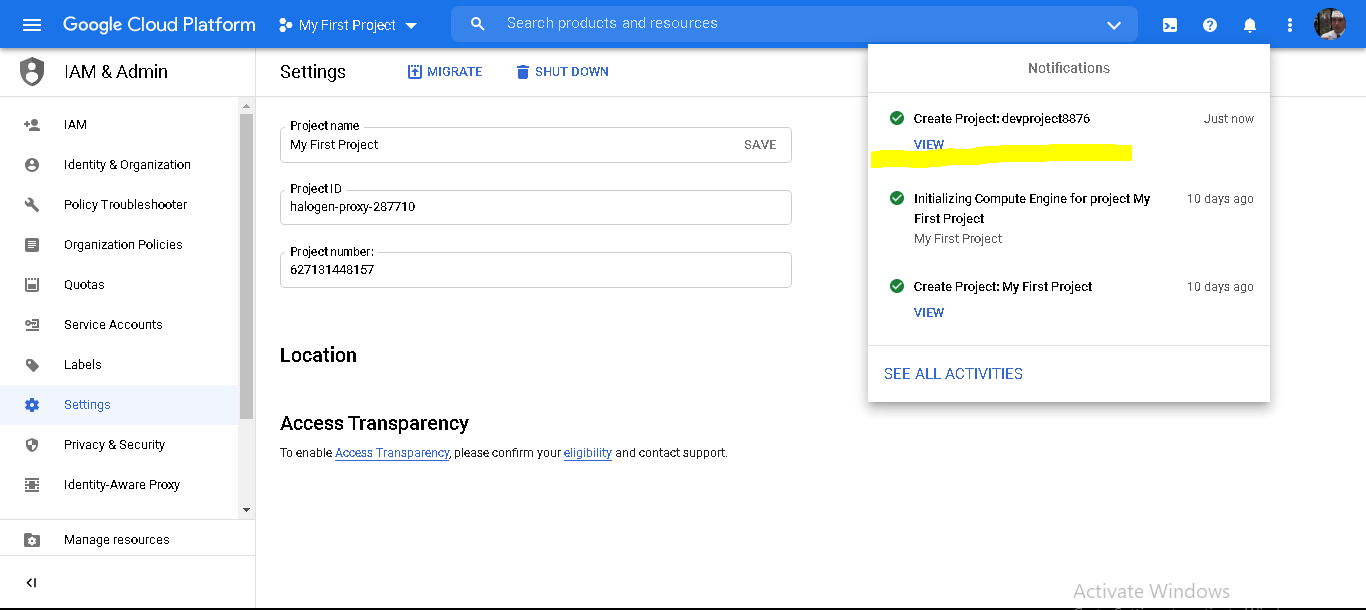


Step1:-

Lets create two projects :-

a) Developer project b) Production project

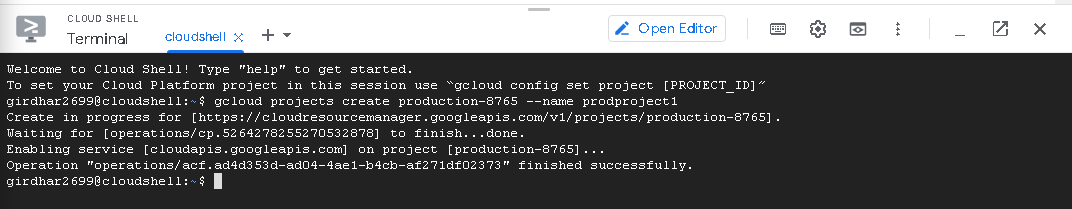
\*I Have created Developer project as (devproject8876)



\*CREATING PRODUCTION PROJECT :- (devproject8876)

In the Cloud Console, on the project selector page, create a Cloud project named developer.

[Activate Cloud Shell:](https://console.cloud.google.com/?cloudshell=true) AT the bottom of the Cloud Console, a [Cloud Shell](https://cloud.google.com/shell/docs/features) session starts and displays a command-line prompt. Cloud Shell is a shell environment with the Cloud SDK already installed, including the gcloud command-line tool, and with values already set for your current project. It can take a few seconds for the session to initialize



\*Finally we get our two projects name as:-

a) devproject8876,

b)prodproject8876

u can check it in project section

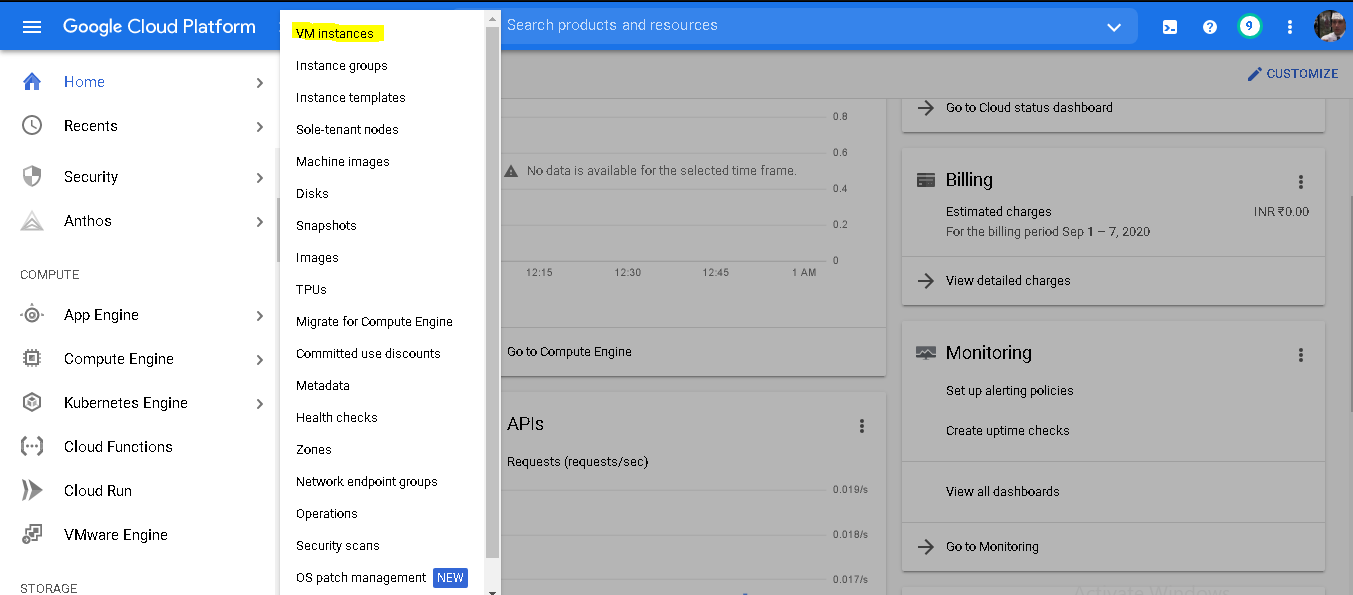
Step 2:-

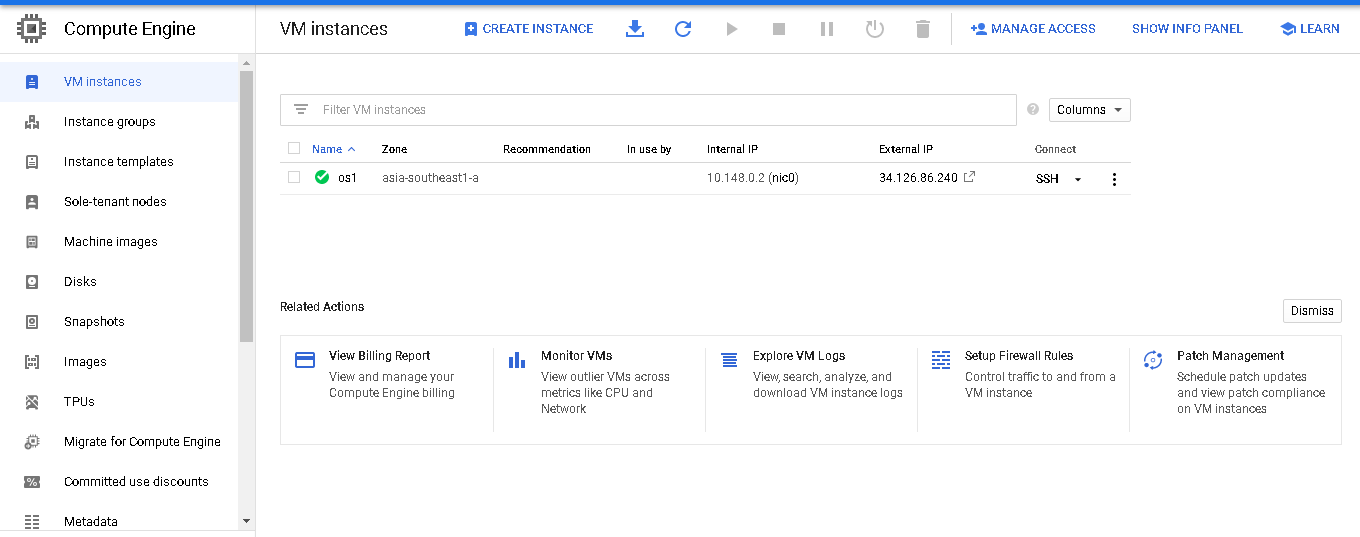
Lets create Vpc for both the projects :-

>>What is Vpc?

* The Virtual Private Cloud networks that you define have global scope.
* They can have subnets in any GCP region worldwide and subnets can span the zones that make up a region.

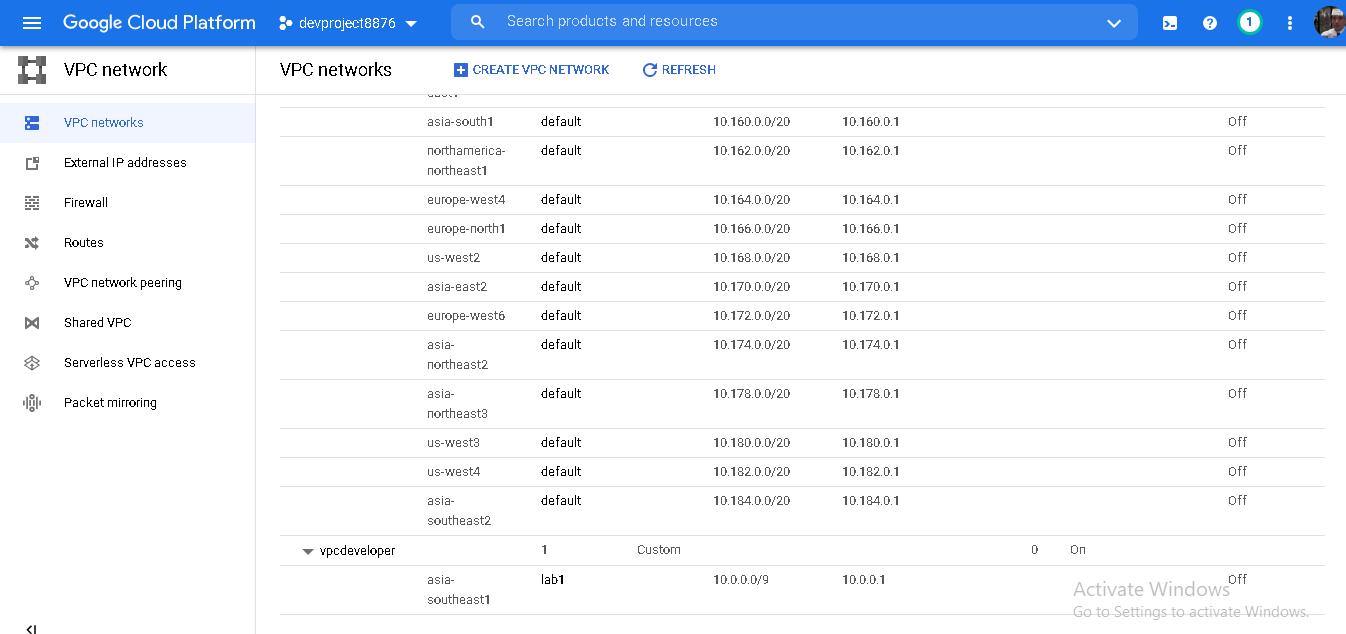
* This architecture makes it easy for you to define your own network layout with global scope.
* You can also have resources in different zones on the same subnet. You can dynamically increase the size of a subnet in a custom network by expanding the range of IP addresses allocated to it.
* Doing that doesn't affect already configured VMs.
* An instance named os1 and a VPC n/w named vpcdeveloper including the subnet lab1 is created in Singapore (asia-southeast1) region in the devproject8876.



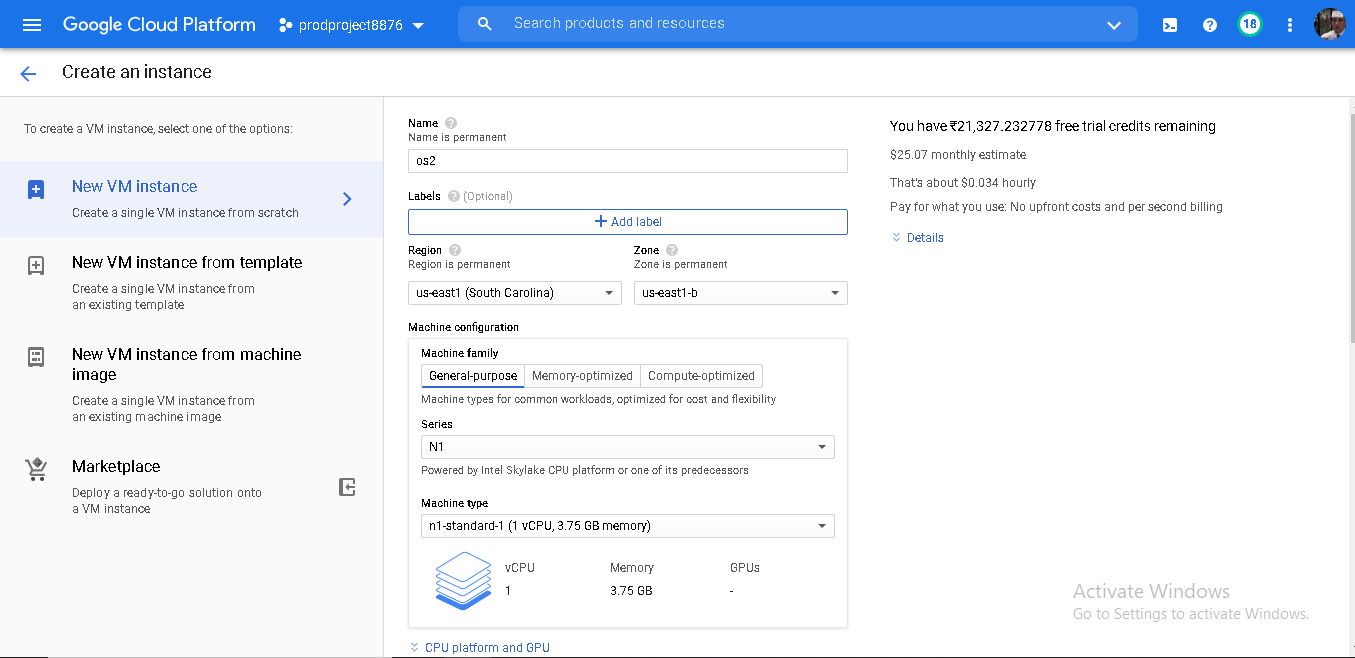


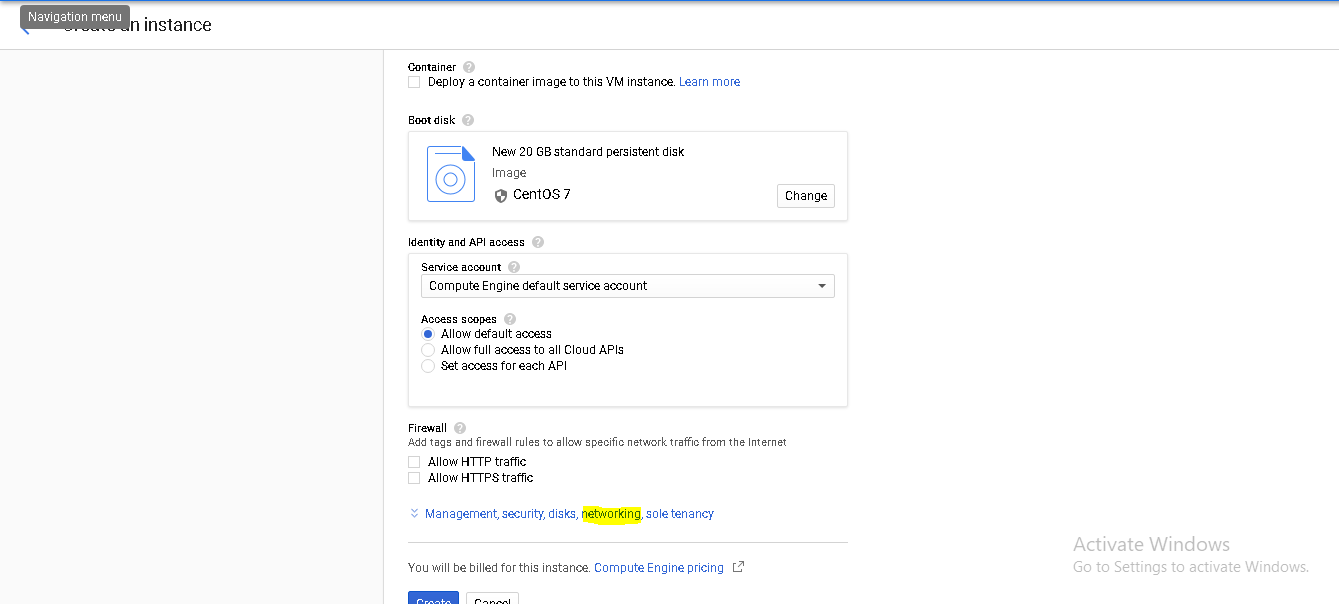
>>CREATE VPC IN DEVELOPER PROJECT

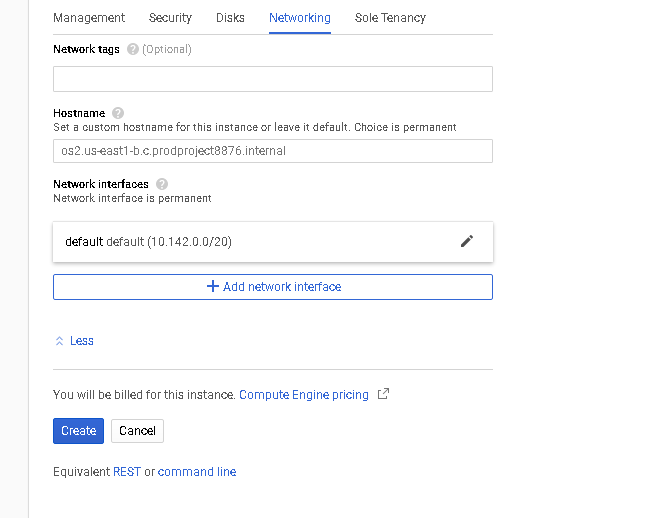
**\*\*Now in developer project(devproject8876) enable API for Compute Engine because it is necessary for creating VPC network**\*\*

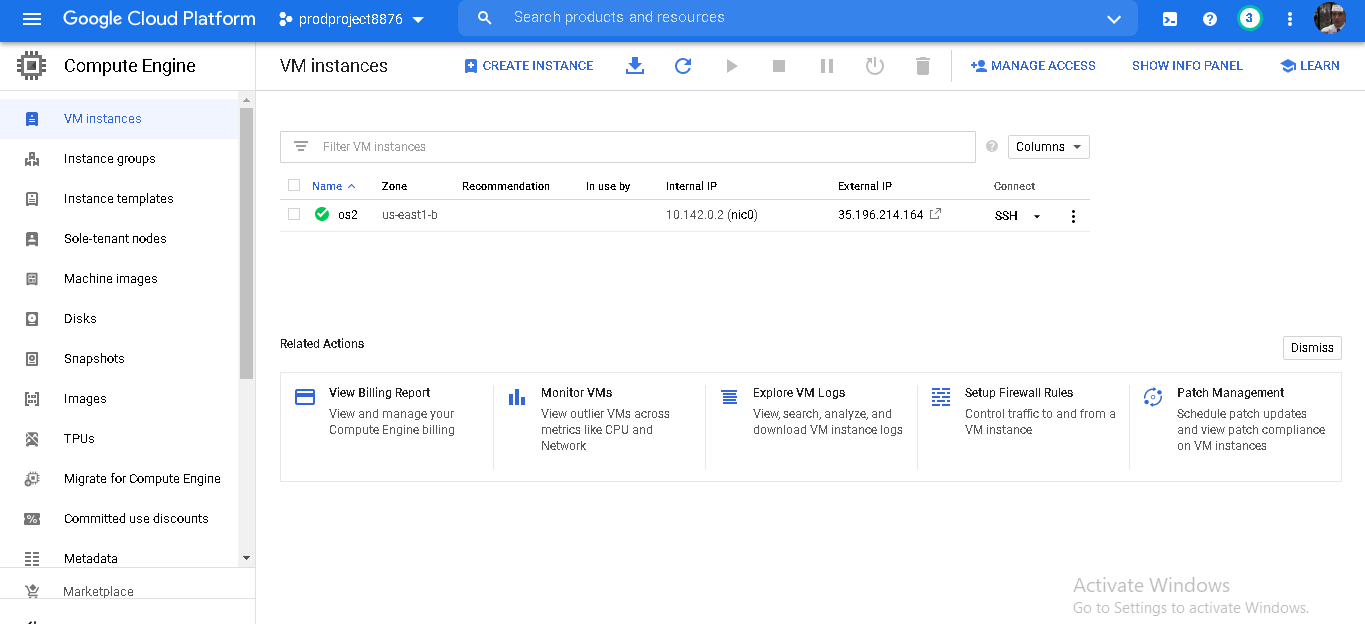


* An instance named os2 and a VPC n/w named vpcproduction including the subnet lab2 is created in south calorina (us-east-1) region in the prodproject8876 project.



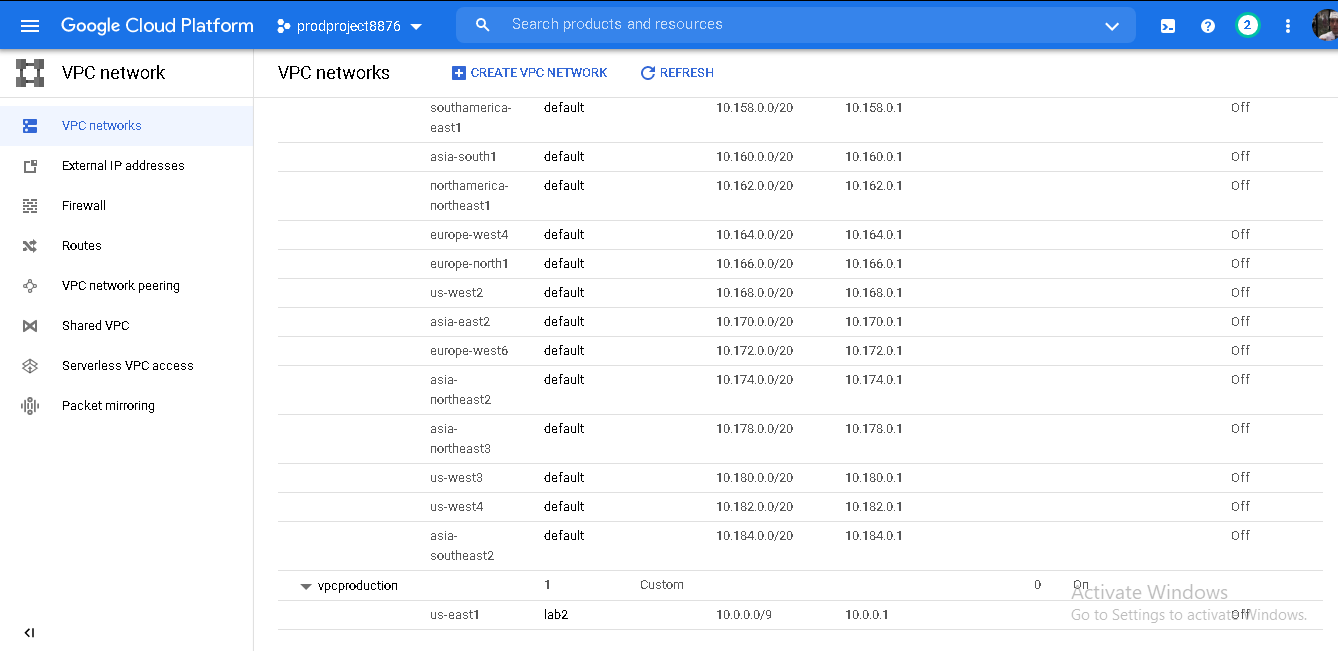




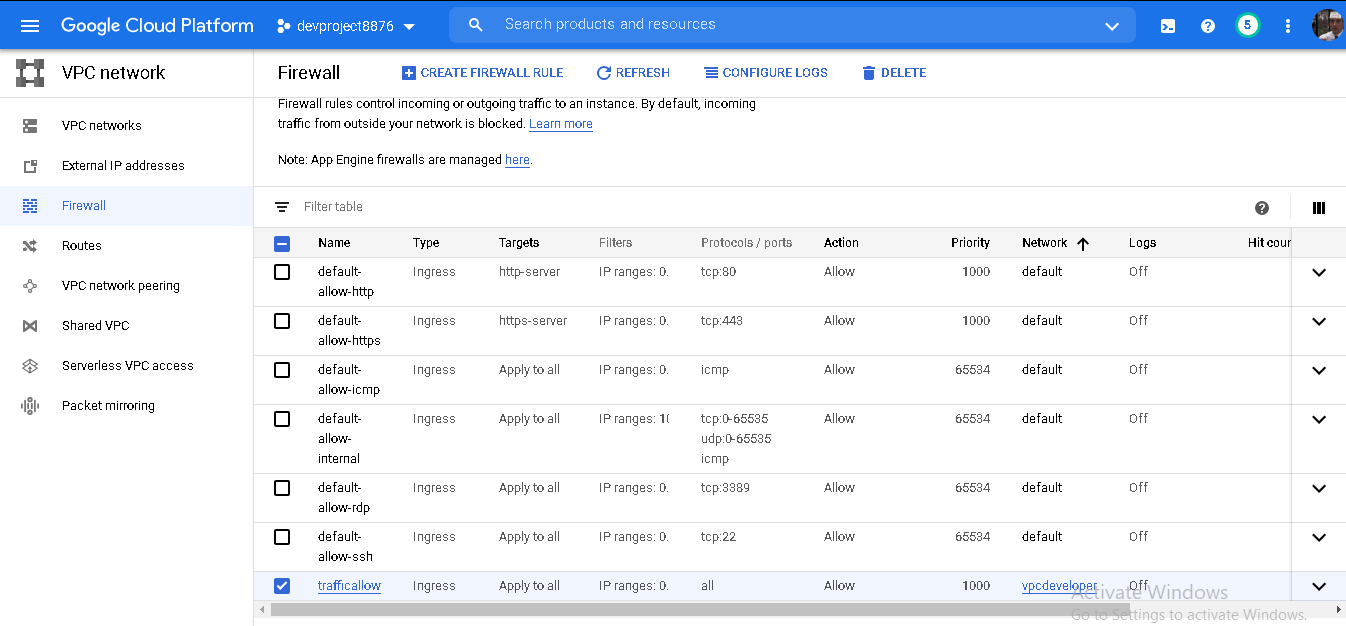


>>CREATE VPC FOR PRODUCTION PROJECT

**\*\*Now in developer project(prodproject8876) enable API for Compute Engine because it is necessary for creating VPC network**\*\*



* A firewall rule named ***trafficallow*** is also created in both the VPC's to allow
* ***Ingress Traffic***and also traffic through some specified ports.

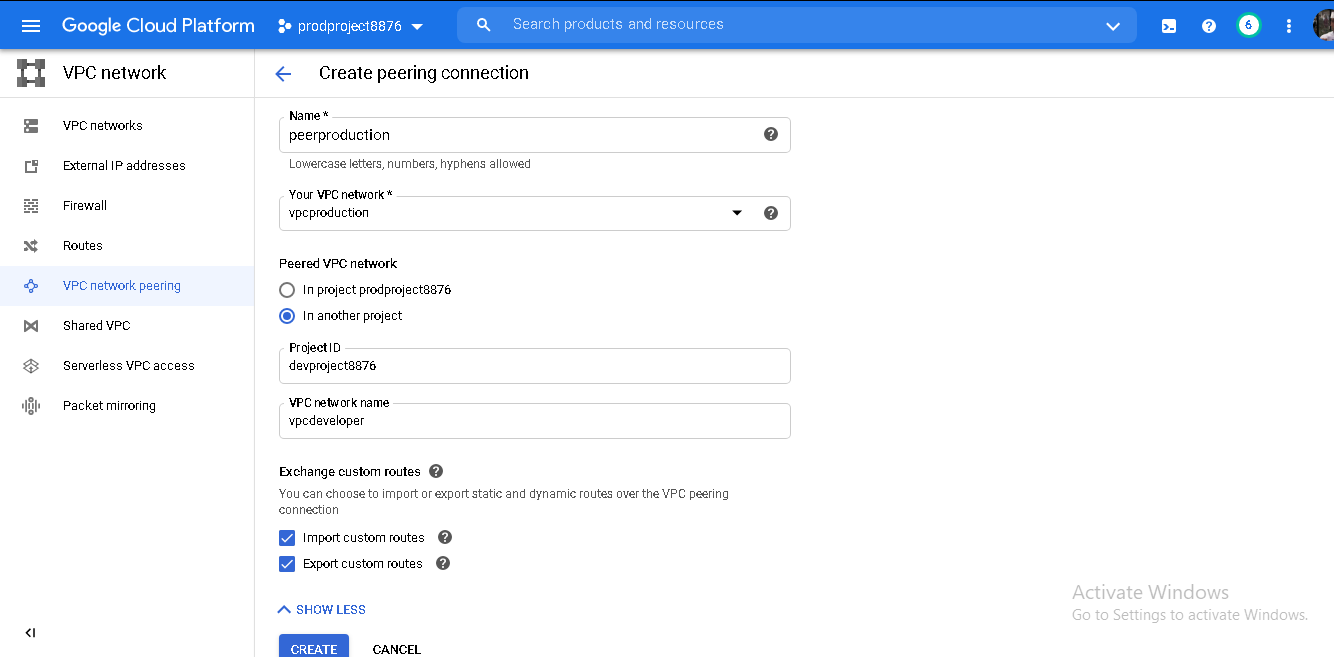


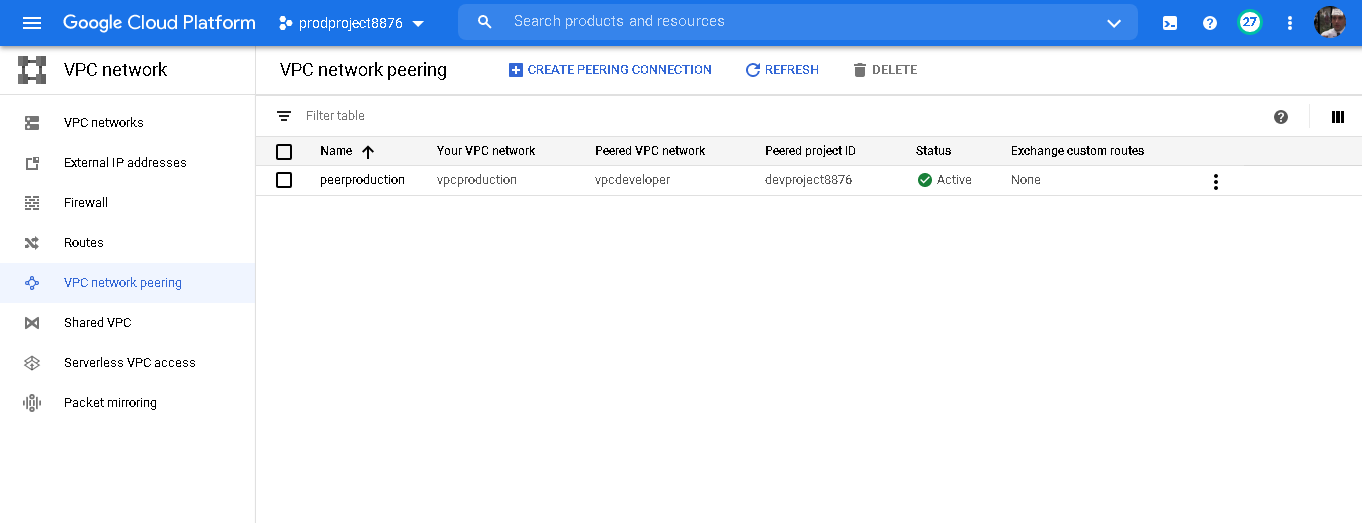
STEP 3:-

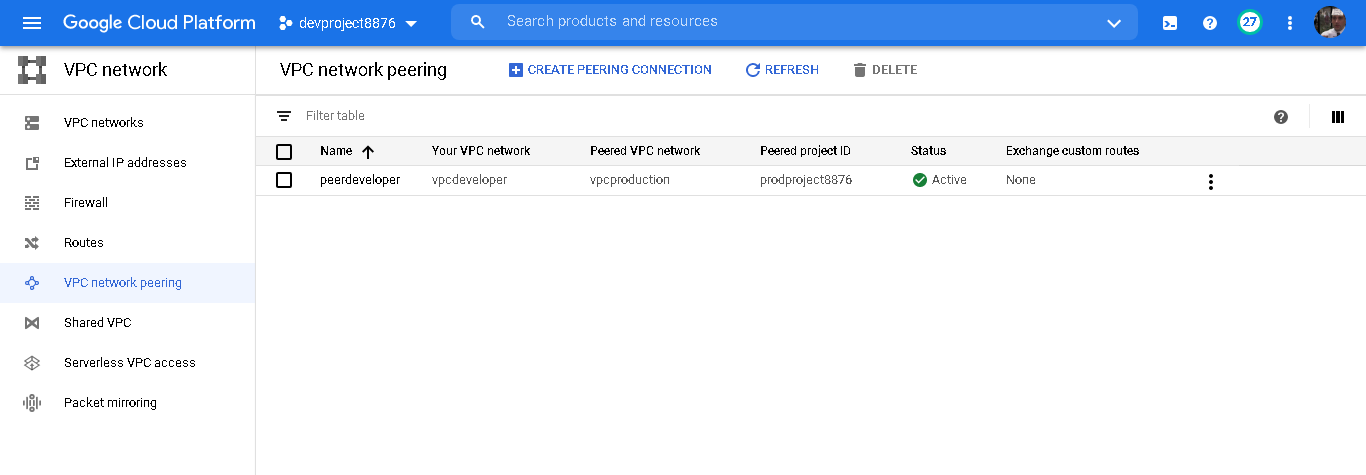
Create a link between both the VPC networks using

VPC Peering:-

* Google Cloud VPC Network Peering allows internal IP address connectivity across two Virtual Private Cloud (VPC) networks regardless of whether they belong to the same project or the same organization.
* After creating the two vpc network in different region lets do VPC peering.







VPC network peering is active. That means we have successfully linked two VPCs

Step 4:- Create a Kubernetes Cluster in developer project and launch any web application with the Load balancer

>>What is Kubernetes?

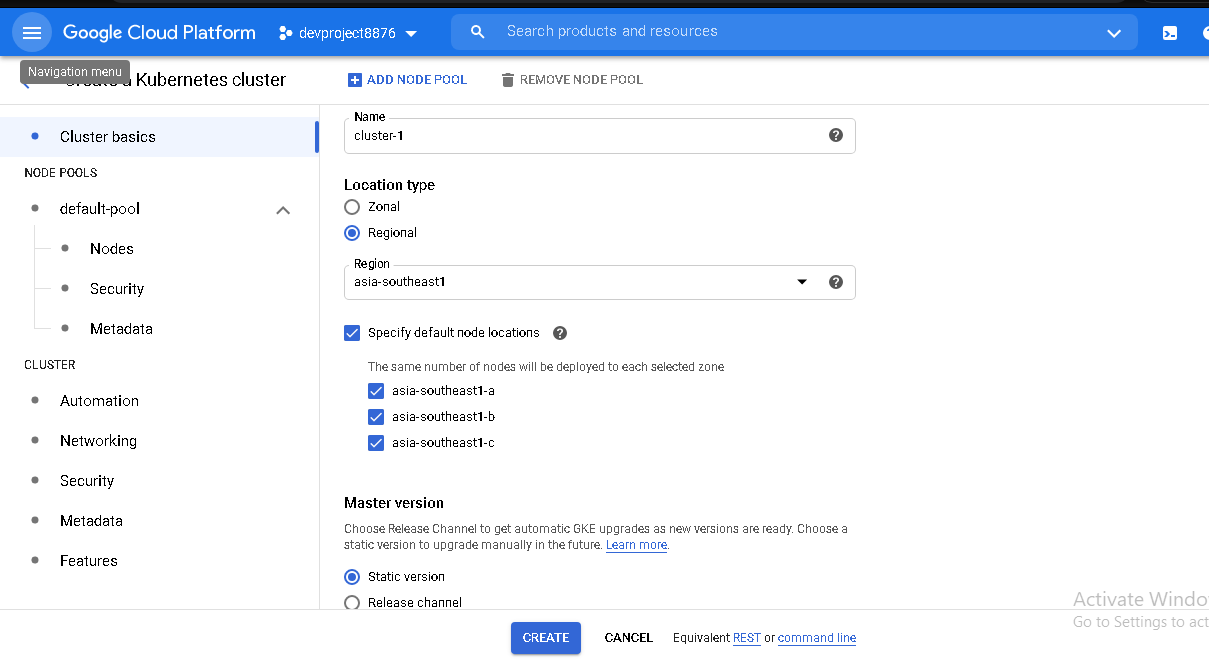
* Kubernetes Engine: It's like an Infrastructure as a Service offering in that it saves you infrastructure chores. It's also like a platform as a service offering, in that it was built with the needs of developers in mind.

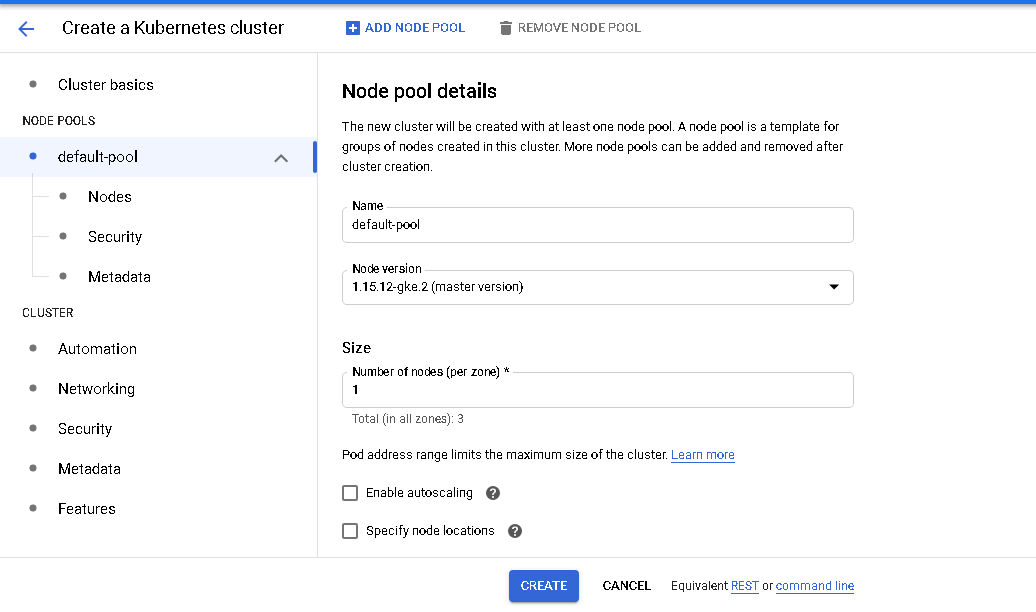
>> WHY DO WE NEED KUBERNETES?

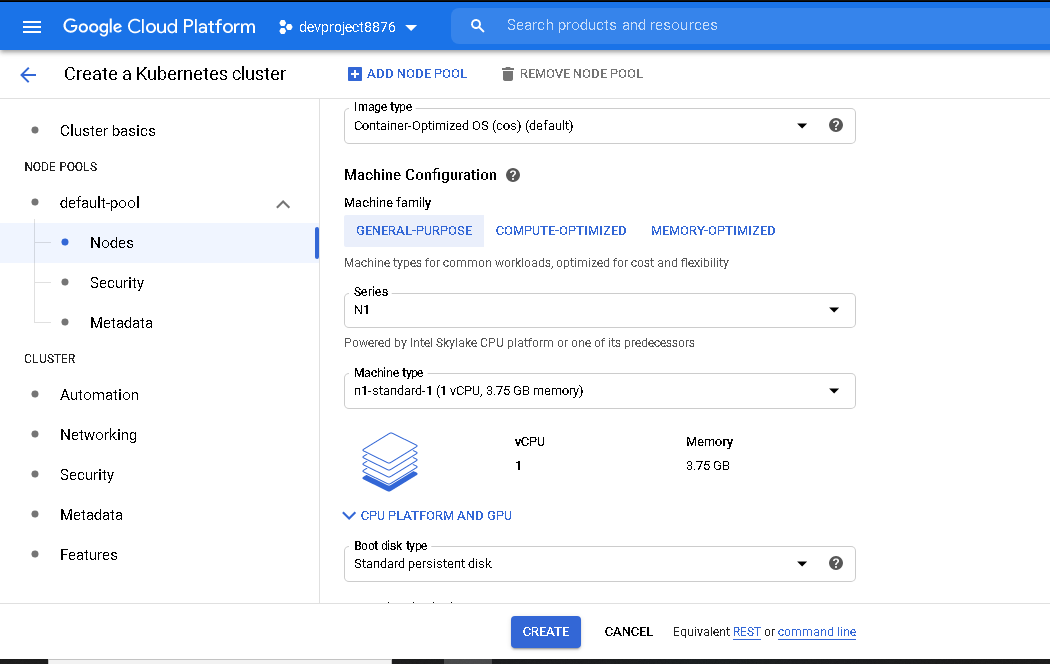
* Virtual Machine are highly configurable, and you can install and run your tools of choice. So you can configure the underlying system resources such as disks and networking, and you can install your own web server database or a middleware. But suppose your application is a big success. As demand for it increases, you have to scale out in units of an entire Virtual Machine with a guest operating system for each. That can mean your resource consumption grows faster than you like.
* Kubernetes, at its basic level, is a system for running and coordinating containerized applications across a cluster of machines. It is a platform designed to completely manage the life cycle of containerized applications and services using methods that provide predictability, scalability, and high availability.

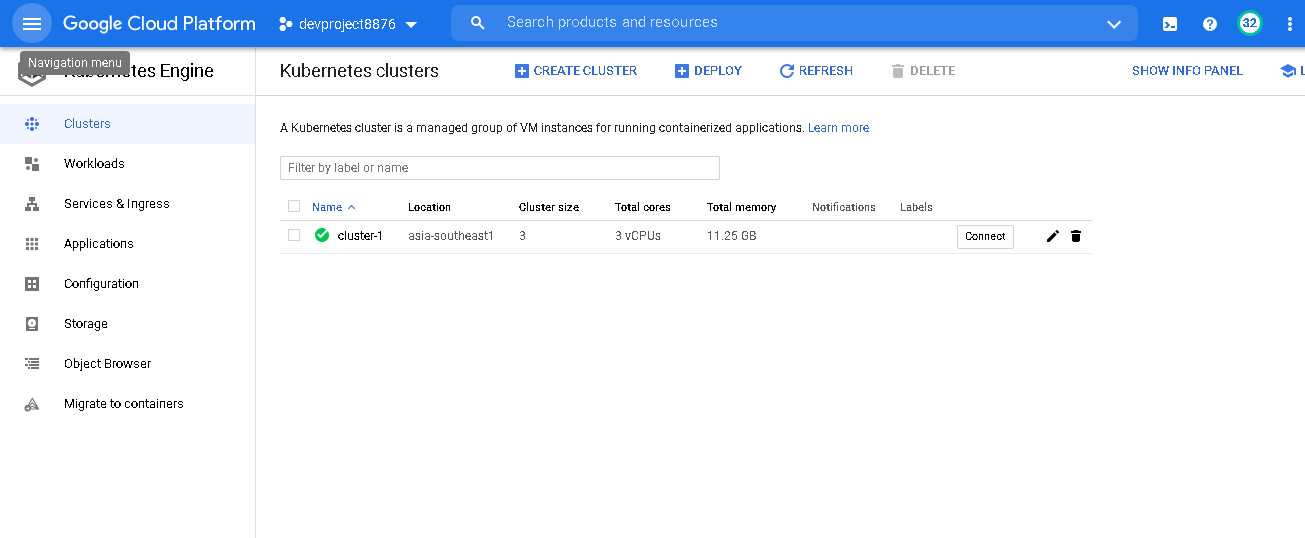
Ler's create our Kubernetes cluster.

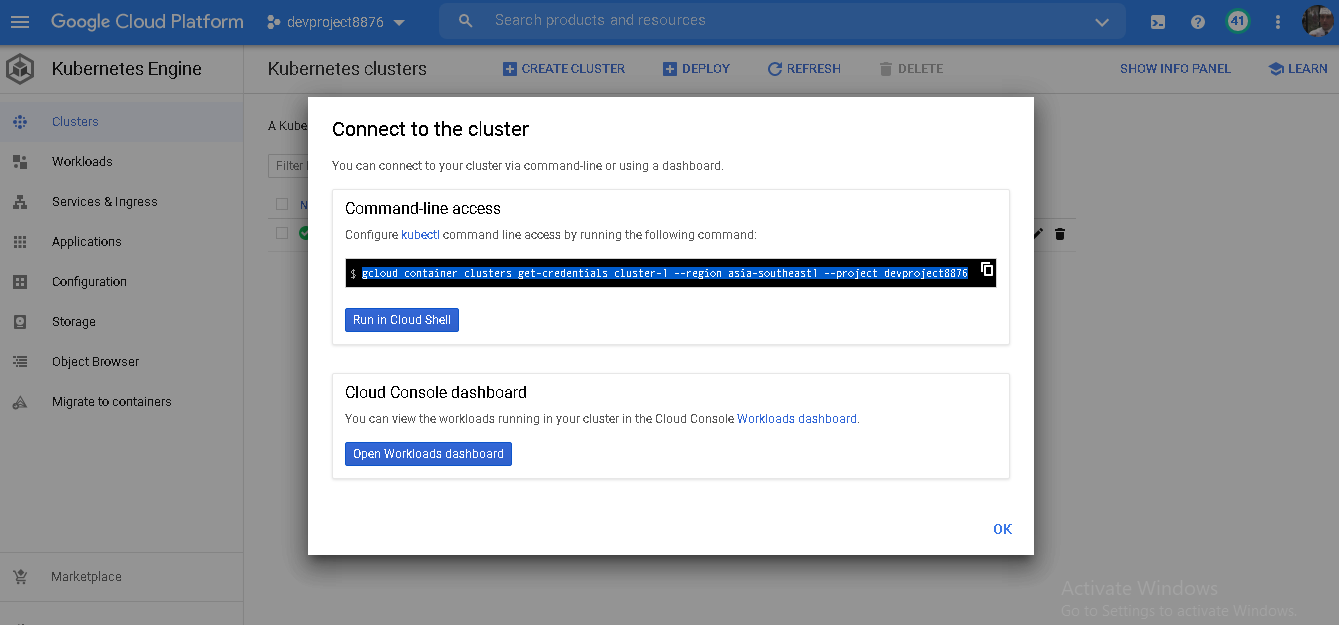
* For that I am going to use CLI and gcloud command. You can install the gcloud sdk at https://cloud.google.com/sdk/docs/downloads-versioned-archives.
* With this the first half of the project is now completed. Moving on to the second half of the project.
* Launch a kubernetes cluster in Singapore region i.e in the ***devpro*** project for launching the frontend i.e the WORDPRESS application inside a POD. The cluster is launched in Regional type so that the nodes can be distributed among different zones in a region.

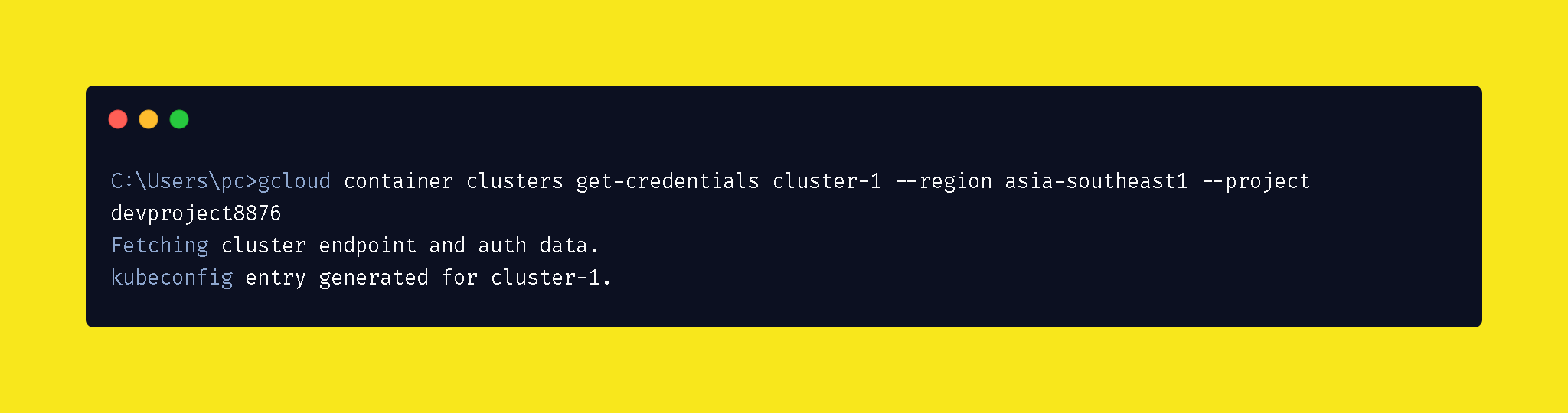




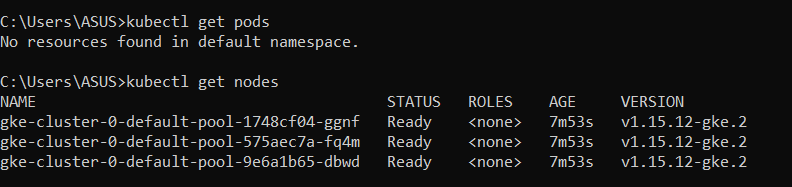




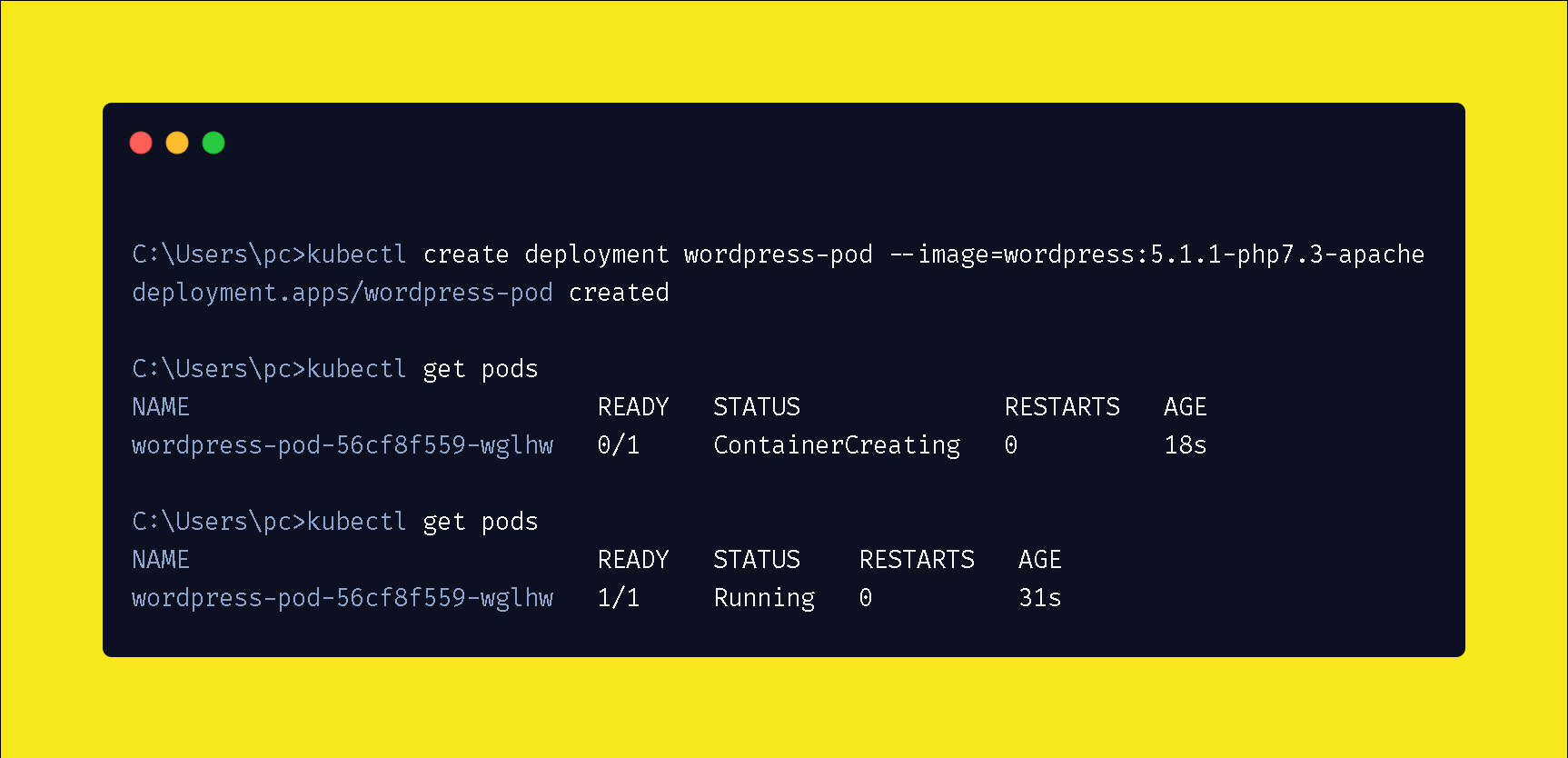




Through the command line we can also see that how many pods and nodes are there in our cluster.



**LAUNCH WORDPRESS APPLICATION WITH LOAD BALANCER**

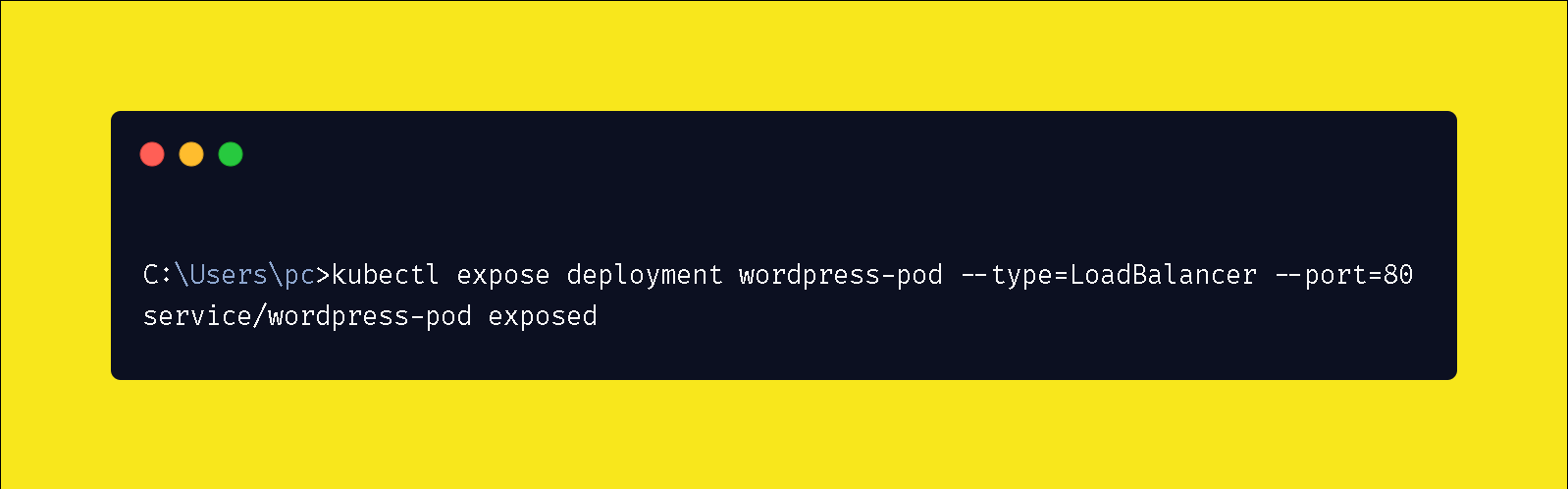


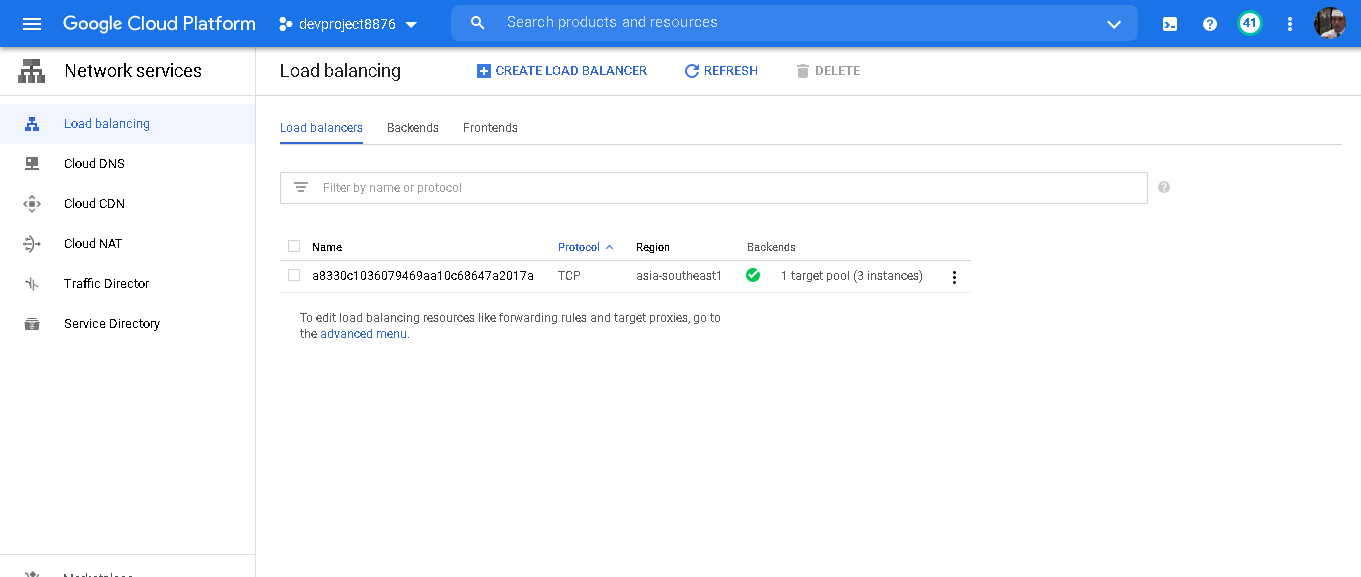
>>Now you can see that the wordpress pod is successfully running.

>>>Here GKE (Google Kubernetes Engine) will be used for deploying the

Wordpress application in the k8s cluster.Here, you can even see the Load

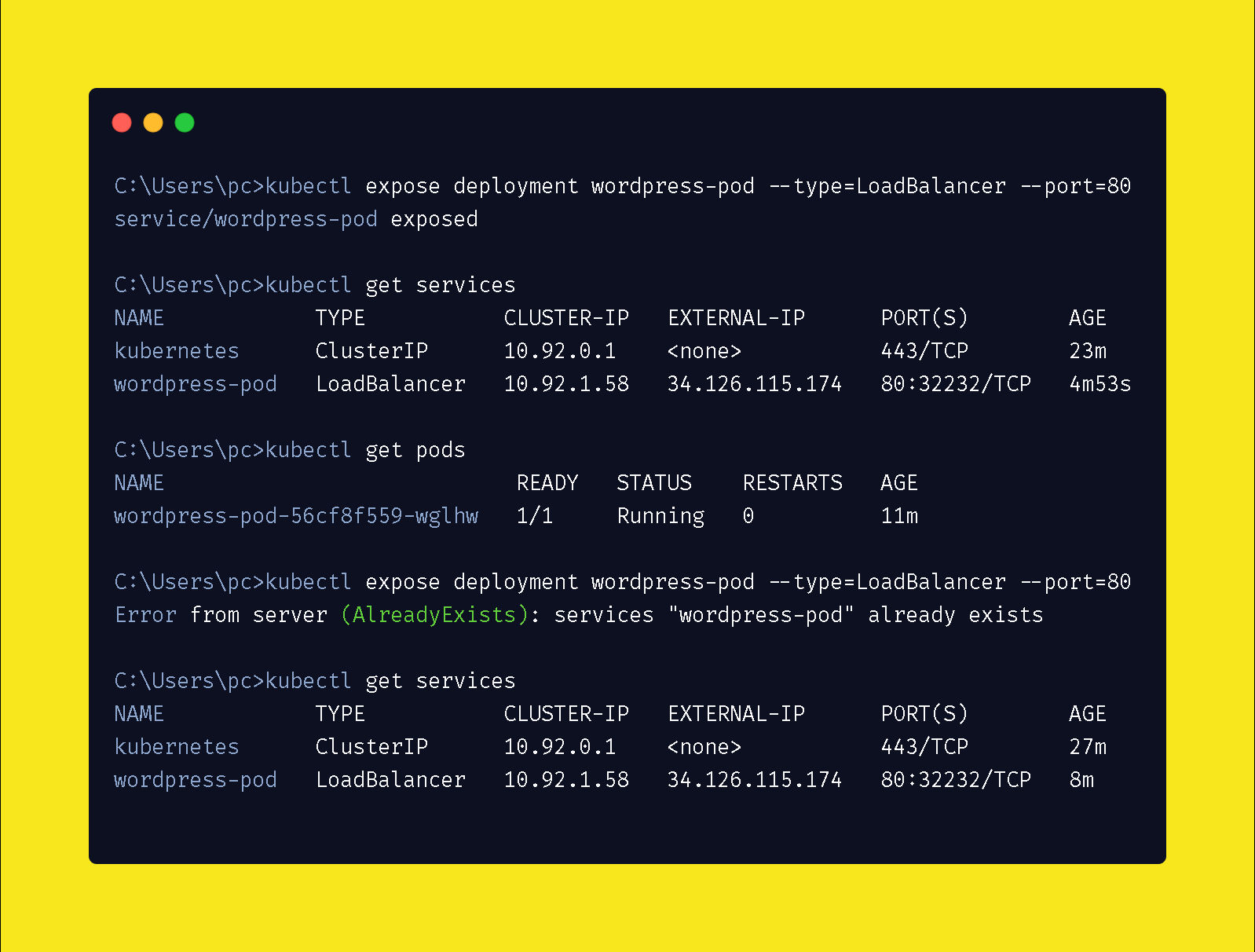
Balancers and their external IP's connected to ***wordpress*** .



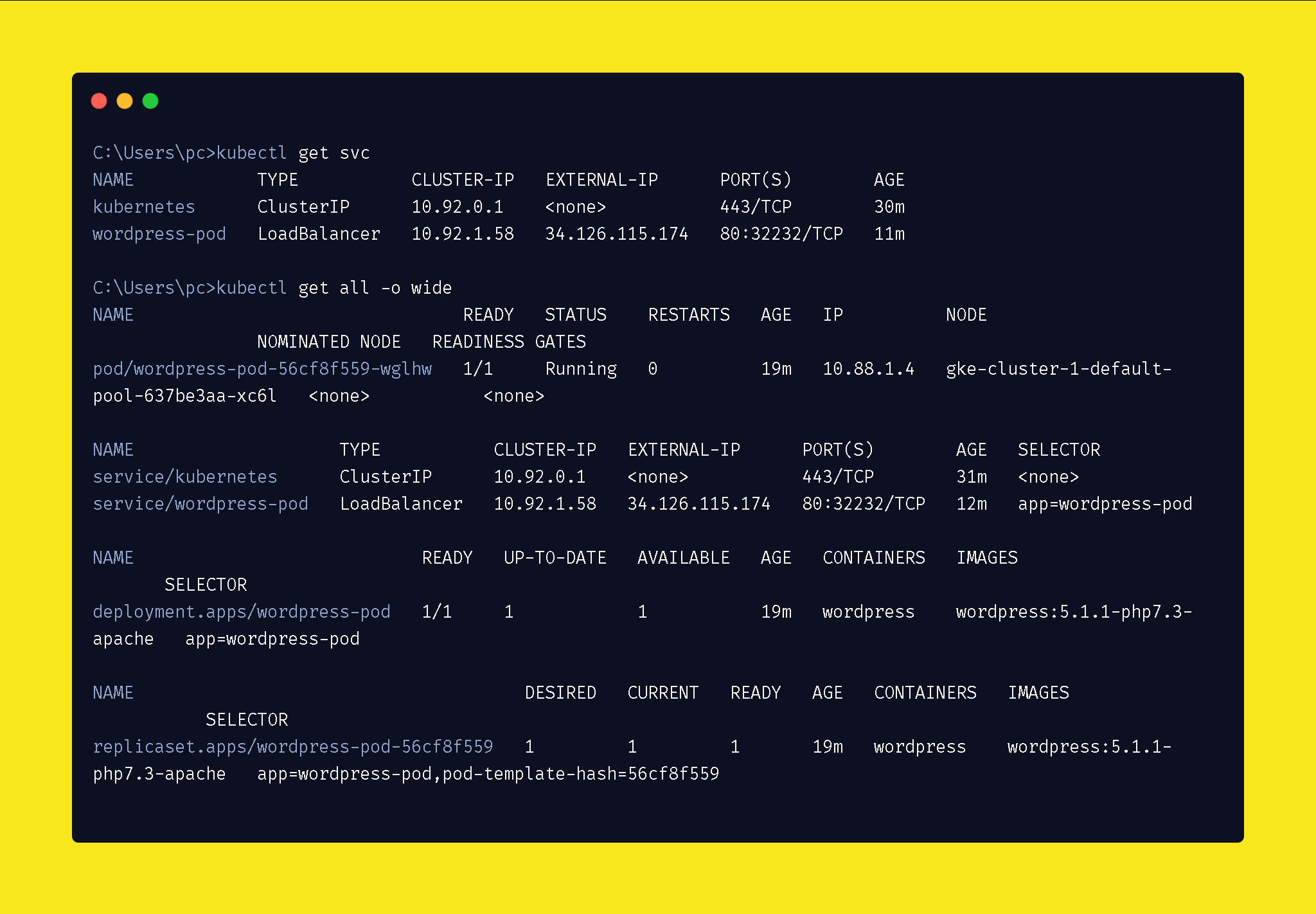


Expose the WordPress service:

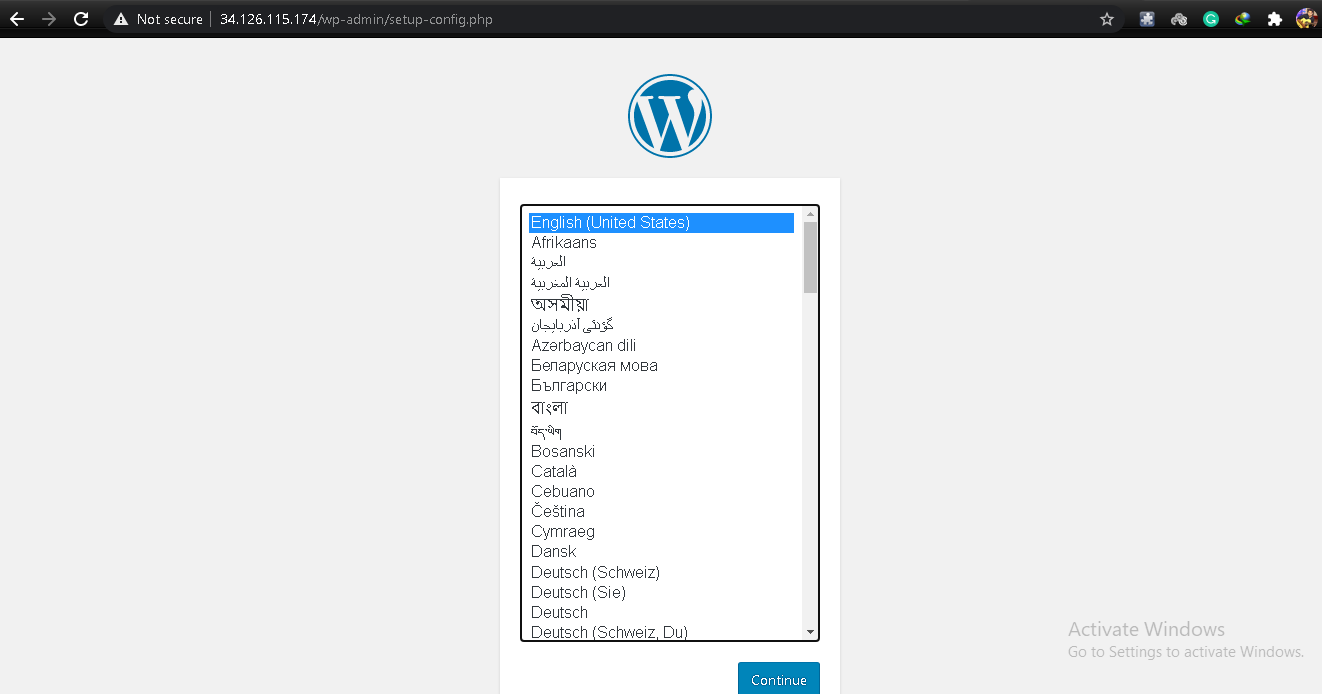
* In the previous step, you deployed a WordPress container, but it's currently not accessible from outside your cluster because it doesn't have an external IP address. You can expose your WordPress app to traffic from the internet by creating and configuring a Kubernetes Service [with](https://cloud.google.com/kubernetes-engine/docs/concepts/network-overview) [an attached external load balancer.](https://cloud.google.com/kubernetes-engine/docs/concepts/network-overview)

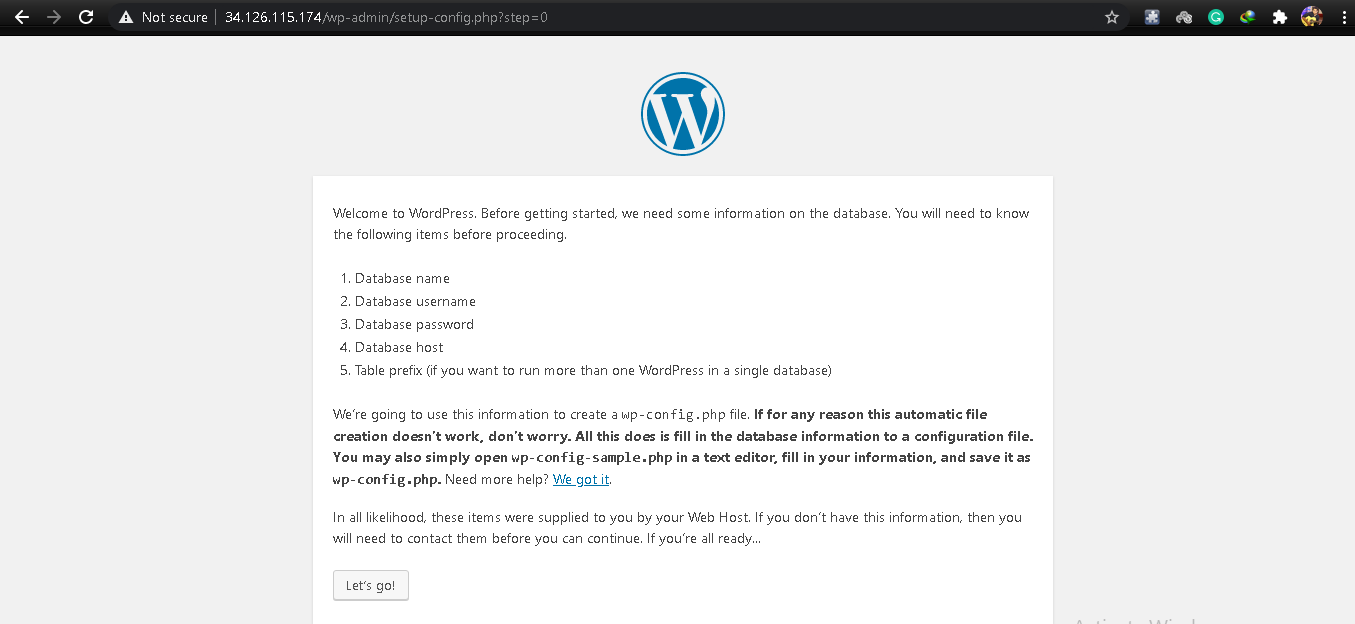


**Here, load balancer is launched.**



Now using the external IP i can go to the outside world and can run the site.

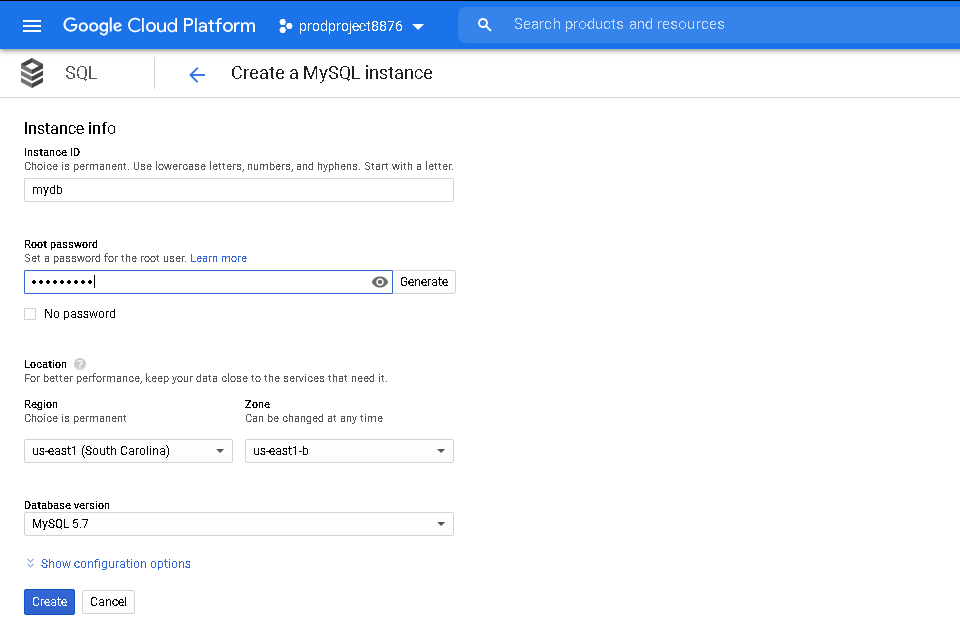


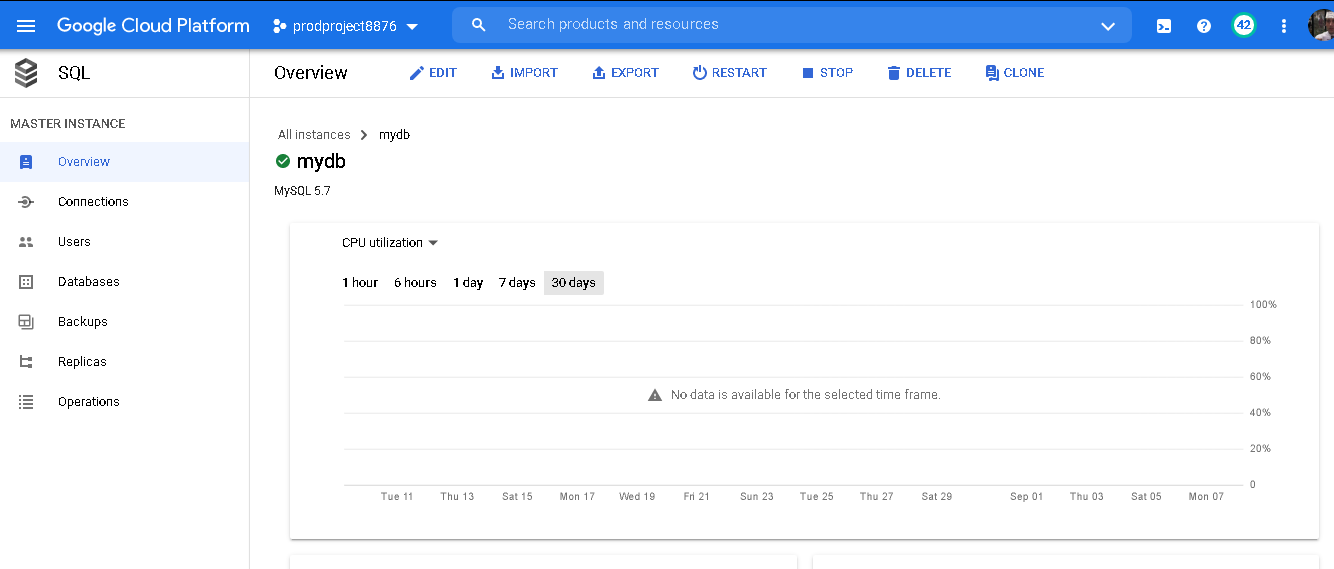


**Wordpress launched successfully**

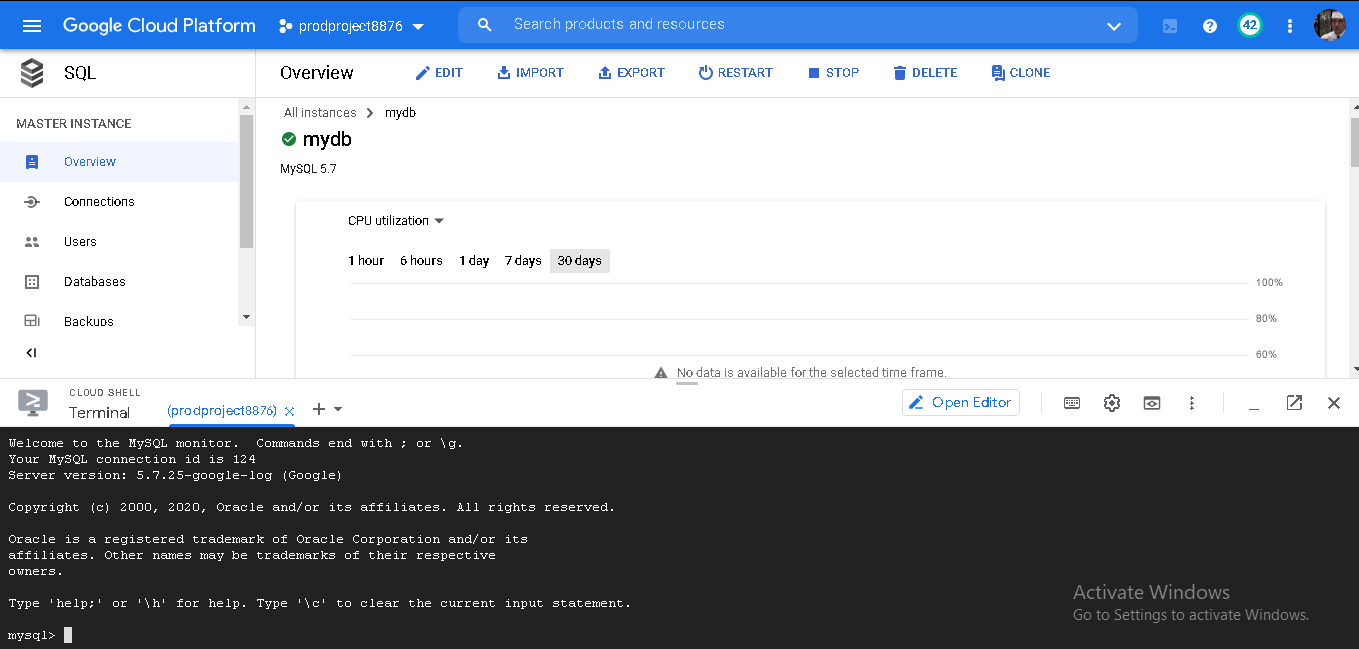
**Step 5:-**

Create a SQL server in the production project and create a database



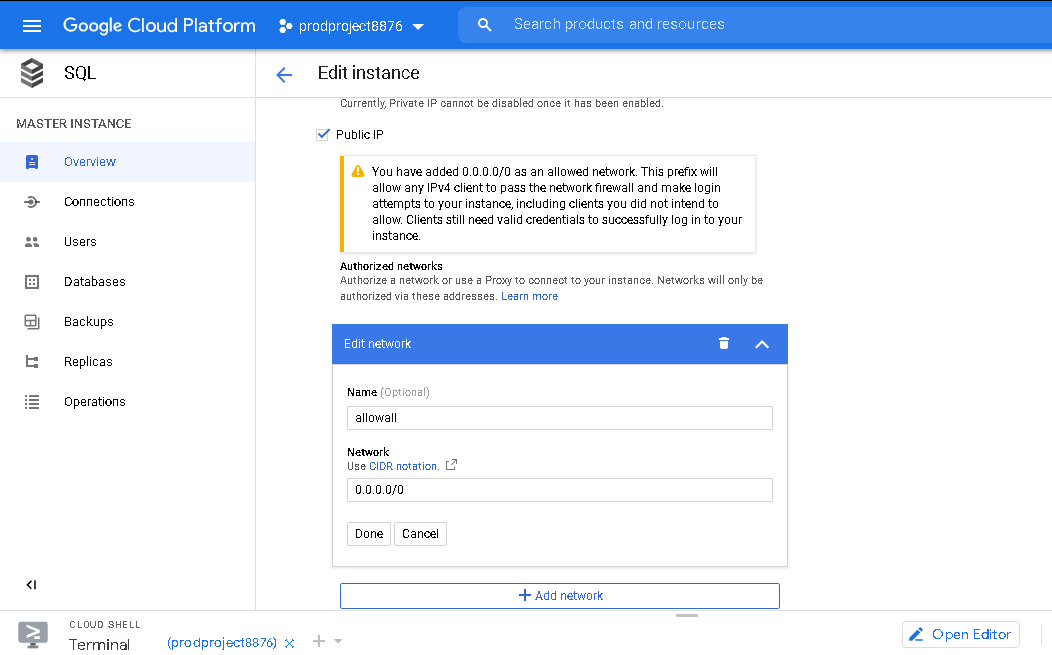


* I created a database named: mydatabase and a user : adi



You may face thiss fail because of the network firewall which does not allow you to log in to your instance

SQL → Overview → Edit

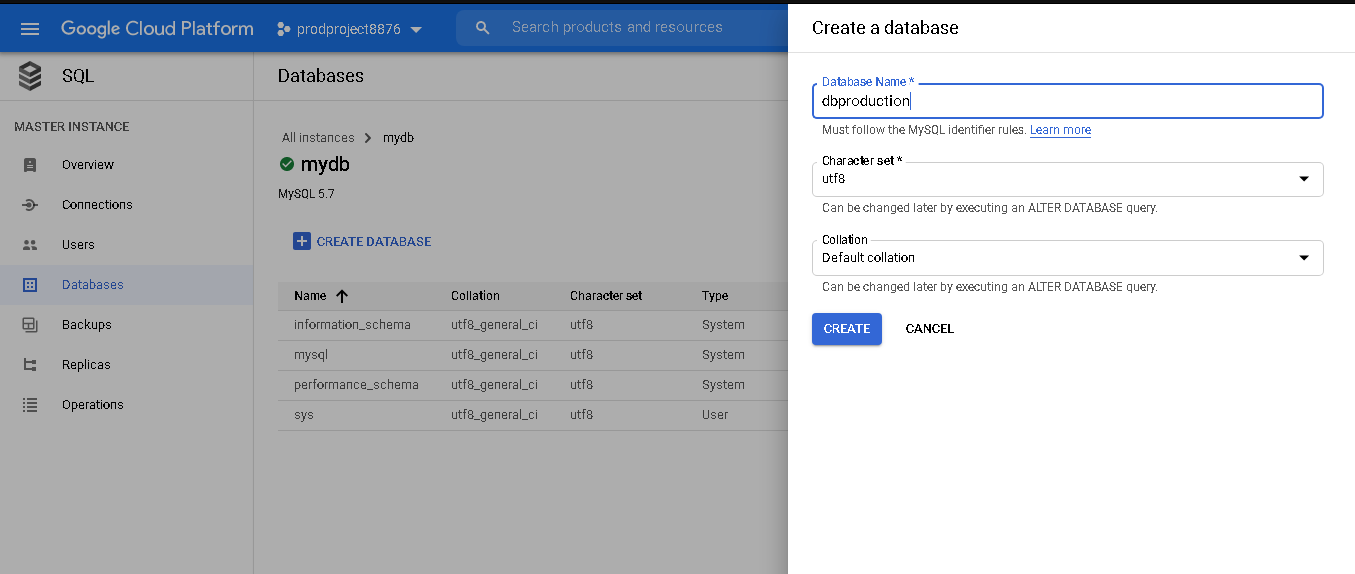


Now you are successfully able to log in to MySQL server.

**CREATE DATABASE NAMED (mydb):-**

**>>Now create a new Database in this instance.**

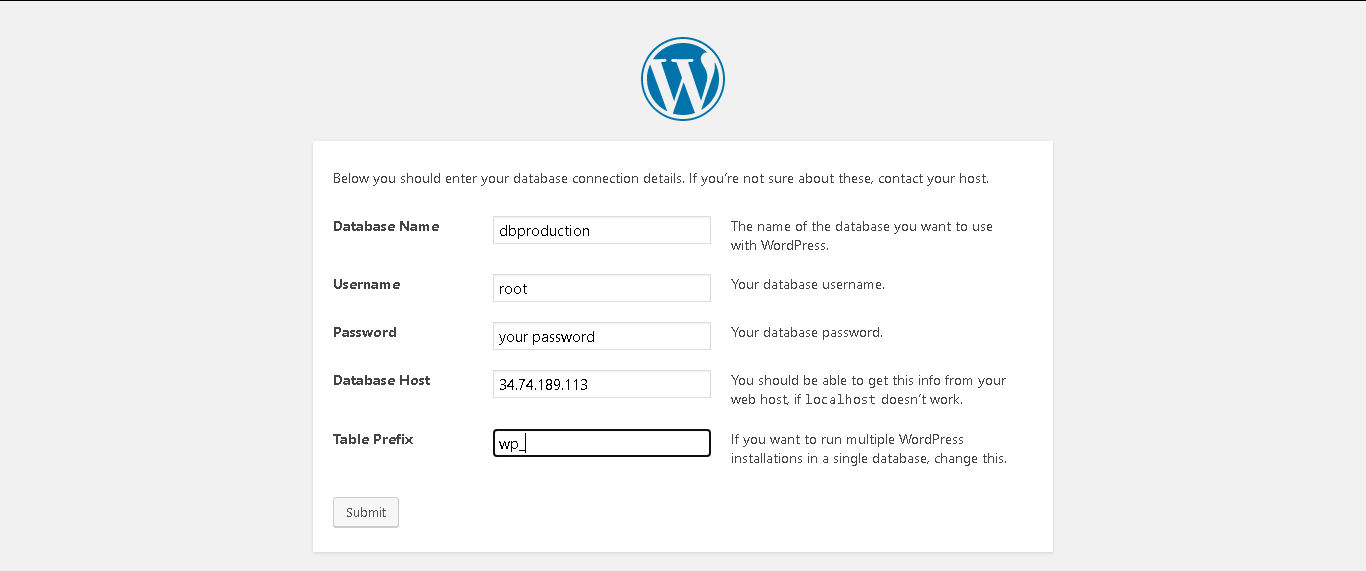
SQL → Databases → Create Database

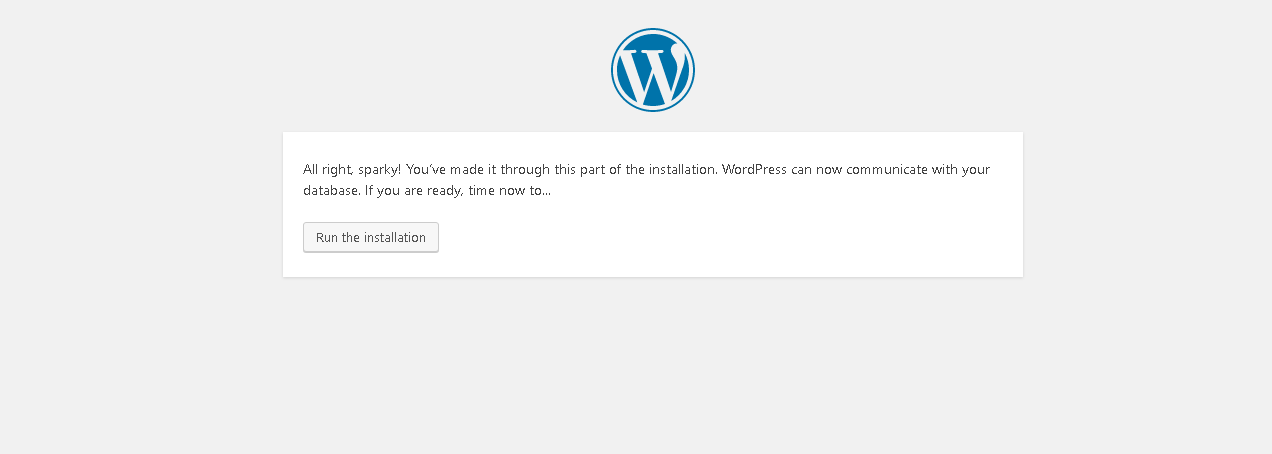


**Step 6:- CONNECT THE SQL DATABASE TO THE WORDPRESS APPLICATION LAUNCHEDINTHEKUBERNETES CLUSTER**

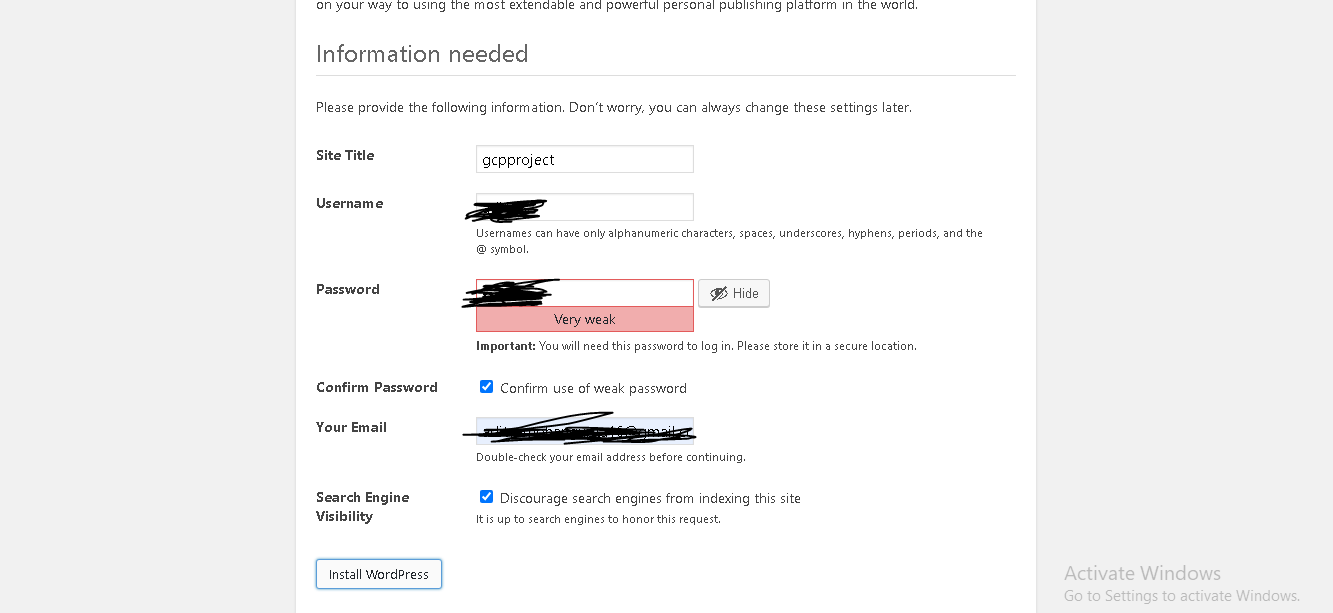
In this section, you set up your WordPress blog.

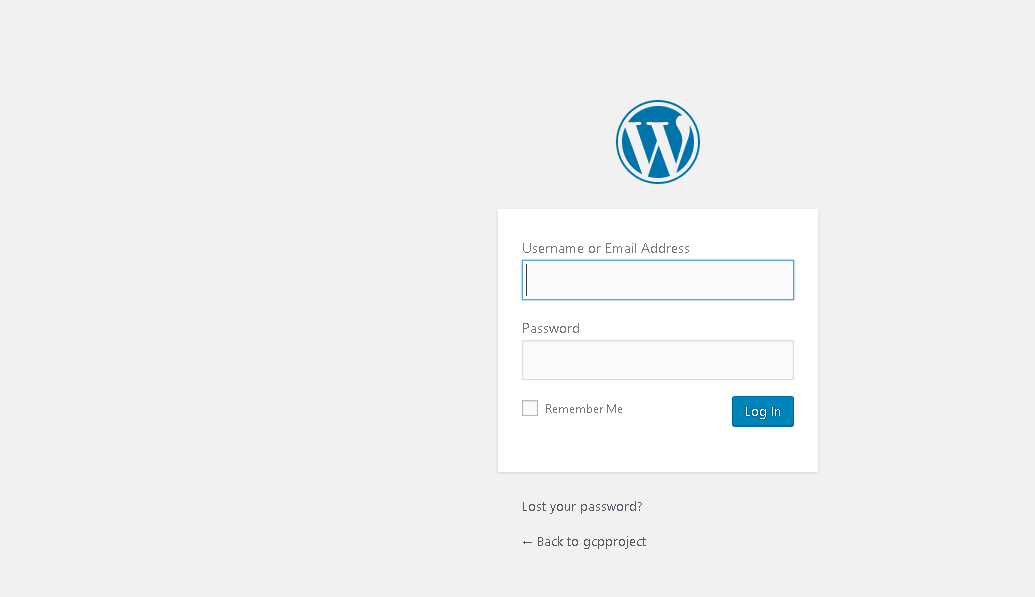
* In your browser, go to the following URL, replacing external-ip-address with the EXTERNAL\_IP address of the service that exposes your WordPress instance:
* On the WordPress installation page, select a language, and then click Continue.
* Complete the Information needed page, and then click Install WordPress.
* Click Log In.
* Enter the username and password that you previously created.

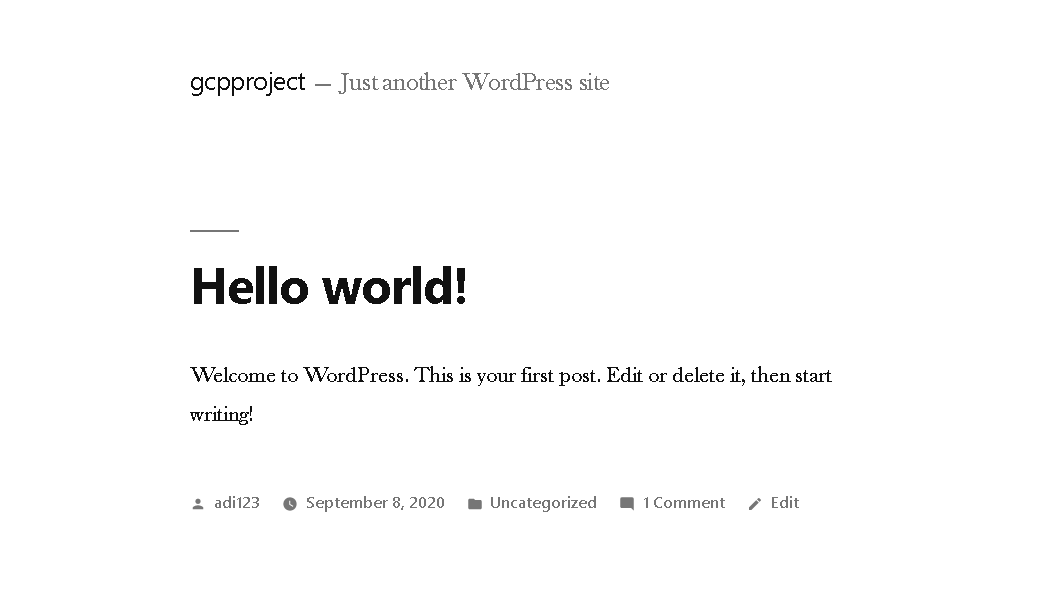




* Wordpress can now communicate with my database.







**Deployment completed…**

In this way I completed my GCP project----🡪\*\*Deployment of WordPress Web Application by Integrating Google Cloud Platform with Kubernetes\*\*

-----Thanku everyone for giving your valuable time reading this—

Any suggestions are welcomed on:- <https://www.linkedin.com/in/aditya-gupta-3a6202135/>

Github Repo:-

<https://github.com/adityamg16/Cloud/upload>