Deployment of Wordpress in Google Cloud using GKE

What is Google Cloud ?

Google Cloud Platform (***GCP***), is a *service* offered by Google, it’s essentially a suite of cloud computing services that runs under the same infrastructure that Google uses internally for its end-user products, such as Google Search Engine , Gmail , drive 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.

It’s most mainstream use cases are:

- **Compute Engine**
- **Storage & Databases**
- **Networking**
- **Big Data**
- **Cloud AI**
- **IoT**
- **GKE**
- **API Platform**

And these are just a few examples of the multitude of services GCP provides..

If we use any service or a tool ,We usually are given two options :

GUI or either from CLI .

But it’s recommended to use CLI , The gcloud CLI can authenticate, local configuration, developer workflow, interactions within Google Cloud APIs. With the gcloud command-line tool,we can perform many common cloud tasks, like creating a Compute Engine VM instance, managing a Google Kubernetes Engine cluster, and deploying an App Engine application, and so on~

For further information on it *click here*.

What is Kubernetes ?

It’s an open-source COE, which stands for Container-Orchestration Engine . This will maintain the process of automating computer application deployment, scaling, and monitoring.

Task Objectives:

◇ *Create multiple projects namely developer and production*

◇ *Create VPC network for both the projects*

◇ *Create a link between both the VPC networks using VPC Peering*

◇ *Create a Kubernetes Cluster in* ***Developer project*** *and launch any web application with the Load balancer*

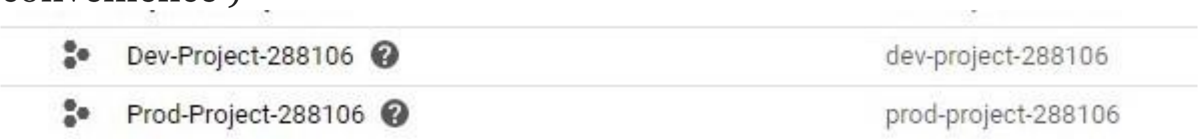
◇ *Create a SQL server in the* ***Production project*** *and create a database*

◇ *Connect the SQL database to the web application launched in the Kubernetes cluster*

Procedure:

→Step-1:

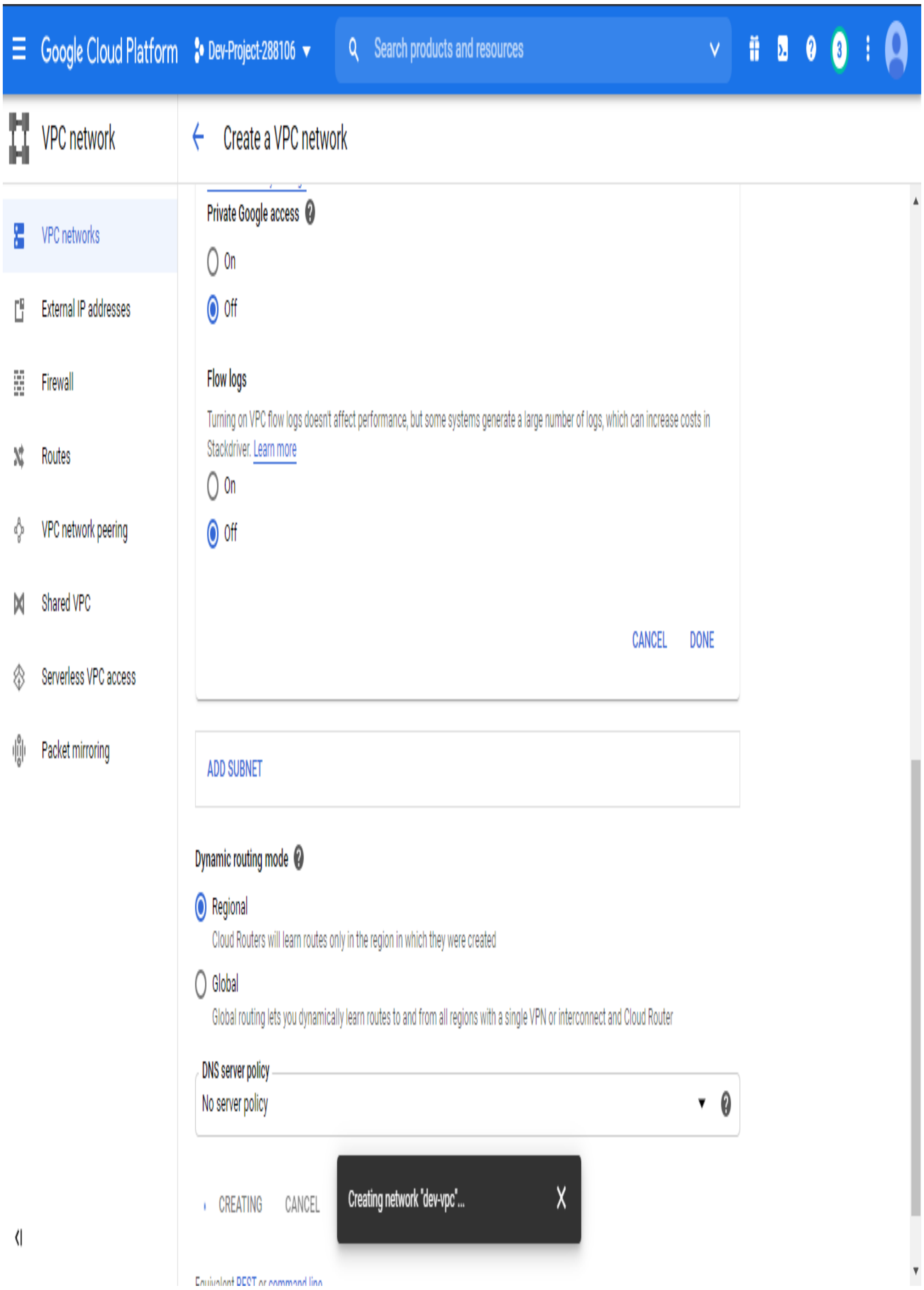
We start off , by creating two different Projects with unique IDs (for convenience)



Dev-Project-288106	dev-project-288106
Prod-Project-288106	prod-project-288106

→ Step 2:

Now , we create VPCs for the two projects so that we can work in a safe Virtual Private Network that we defined , this works in a similar manner to a conventional day-to-day network . But we can build our own virtual DataCenter using this:



Create a VPC network

Private Google access

○ On

● Off

Flow logs

Turning on VPC flow logs doesn't affect performance, but some systems generate a large number of logs, which can increase costs in Stackdriver. Learn more

○ On

● Off

CANCEL DONE

ADD SUBNET

Dynamic routing mode

● Regional

Cloud Routers will learn routes only in the region in which they were created

○ Global

Global routing lets you dynamically learn routes to and from all regions with a single VPN or interconnect and Cloud Router

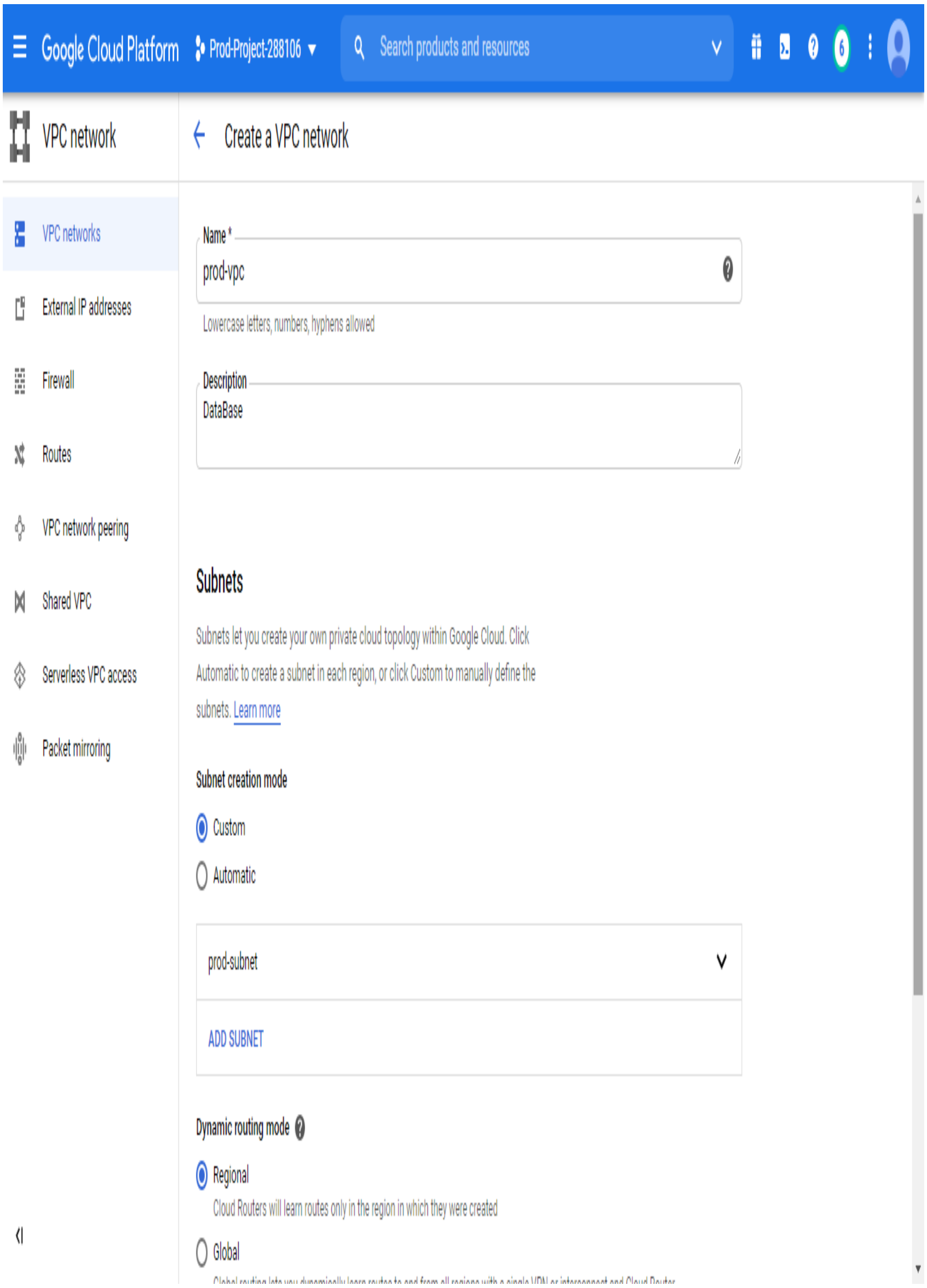
DNS server policy

No server policy ▾

CREATING CANCEL

Creating network "dev-vpc"... X

Equivalent REST or command line



Create a VPC network

Name *
prod-vpc

Lowercase letters, numbers, hyphens allowed

Description
DataBase

Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. Learn more

Subnet creation mode

● Custom

○ Automatic

prod-subnet

Dynamic routing mode

● Regional

Cloud Routers will learn routes only in the region in which they were created

○ Global

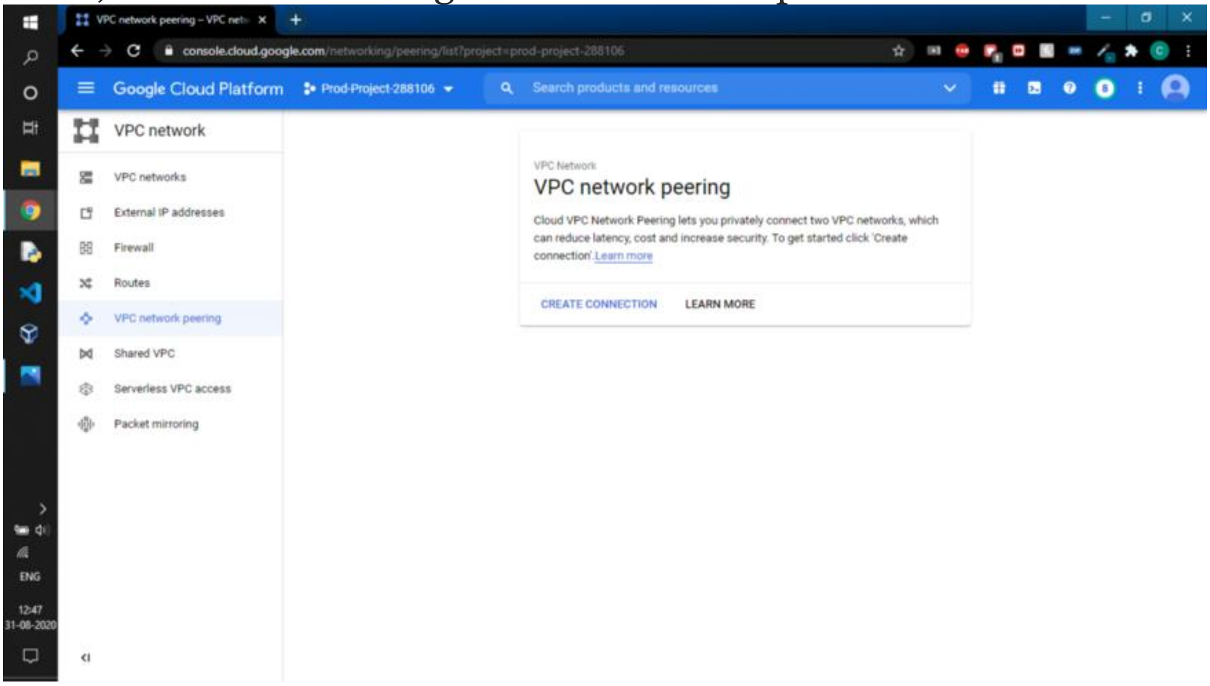
Global routing lets you dynamically learn routes to and from all regions with a single VPN or interconnect and Cloud Router

Here we created two VPCs for individual projects

Now , we have a virtual topology along with configured prefix ranges for two subnets with proper network policies .All custom-configured to match the needs.

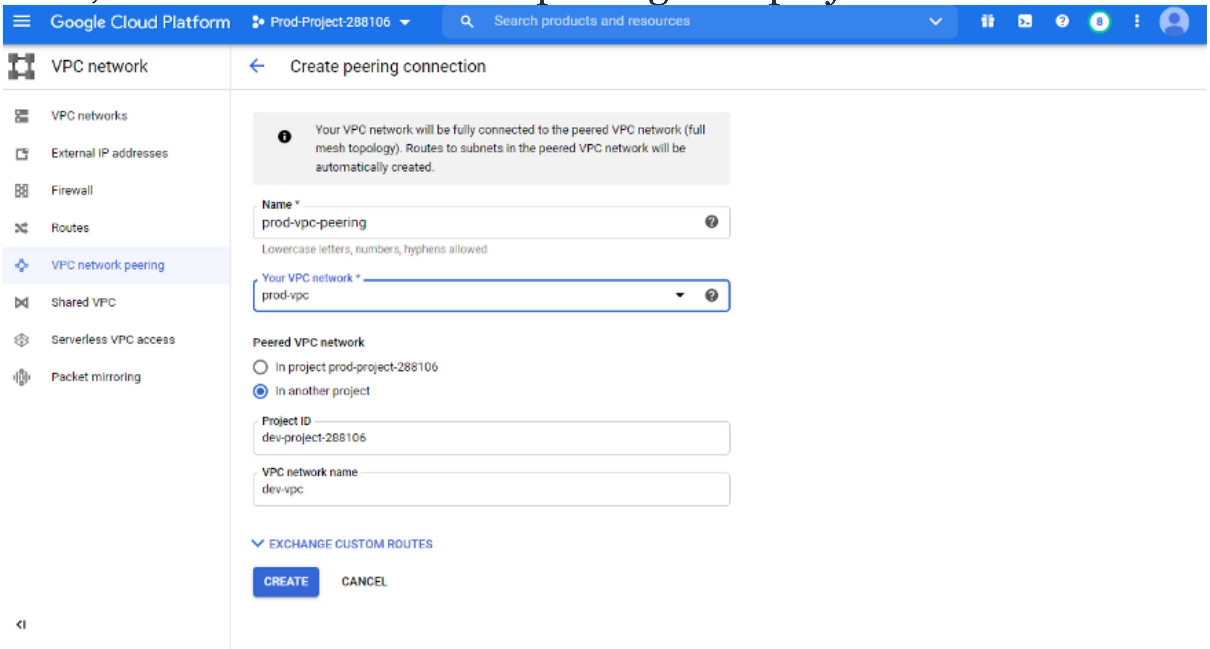
→ Step 3:

Next , we create a Peering between both the private networks :



Cloud VPC Network Peering lets you privately connect two VPC networks, which can reduce latency, cost and increase security. To get started click 'Create connection'. Learn more

Here, we have to create a VPC peering for a project



Your VPC network will be fully connected to the peered VPC network (full mesh topology). Routes to subnets in the peered VPC network will be automatically created.

Name *: prod-vpc-peering

Lowercase letters, numbers, hyphens allowed

Your VPC network *: prod-vpc

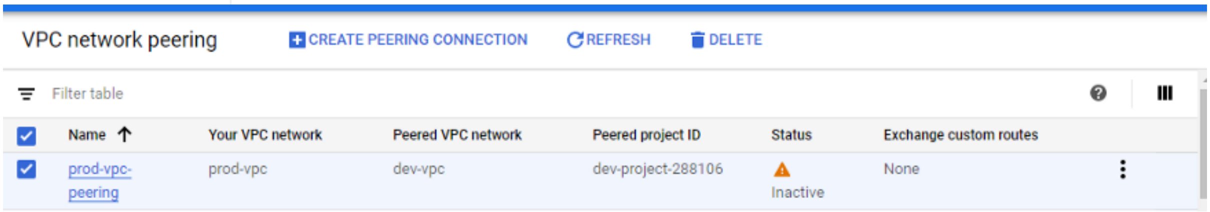
Peered VPC network

○ In project prod-project-288106

● In another project

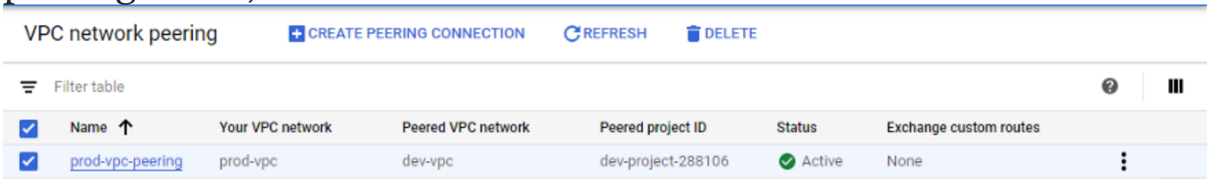
Project ID: dev-project-288106

VPC network name: dev-vpc



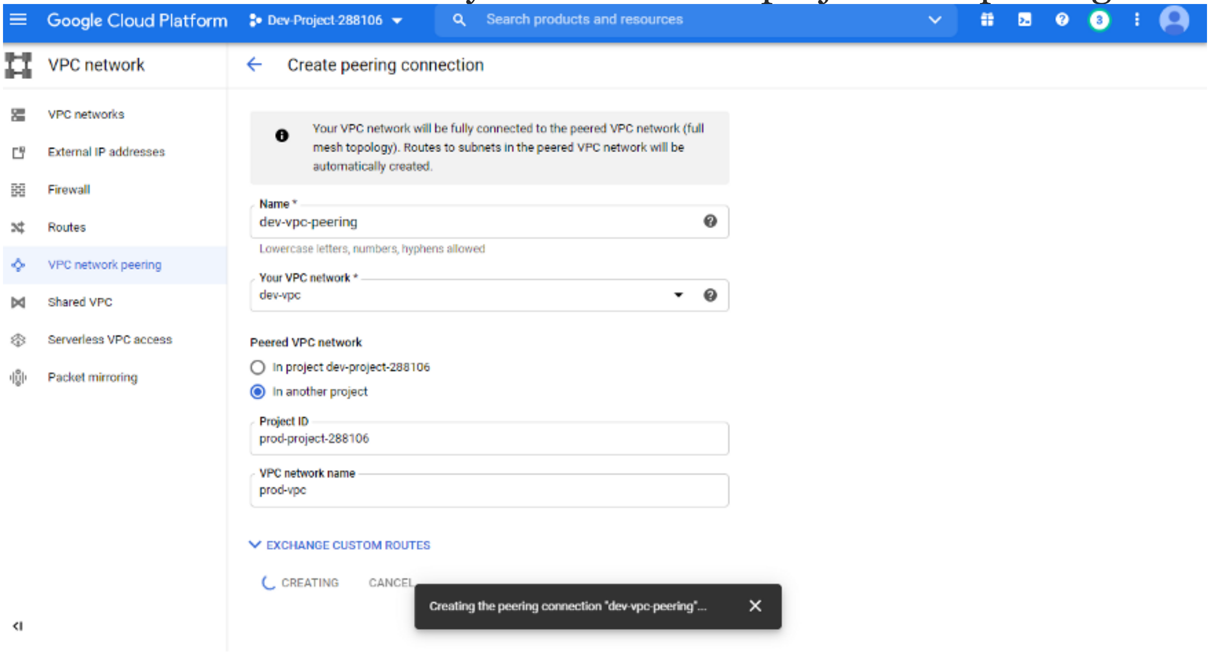
☑	Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
☑	prod-vpc-peering	prod-vpc	dev-vpc	dev-project-288106	⚠ Inactive	None

Now, that VPC peering for the prod-projects is created .But at the moment it’s inactive due to the peering connection, this will established if and only when the other project is also performing peering .Then, the status will activate.



☑	Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
☑	prod-vpc-peering	prod-vpc	dev-vpc	dev-project-288106	✓ Active	None

In a similar fashion we may create the Dev-project VPC peering



Your VPC network will be fully connected to the peered VPC network (full mesh topology). Routes to subnets in the peered VPC network will be automatically created.

Name *
dev-vpc-peering
Lowercase letters, numbers, hyphens allowed

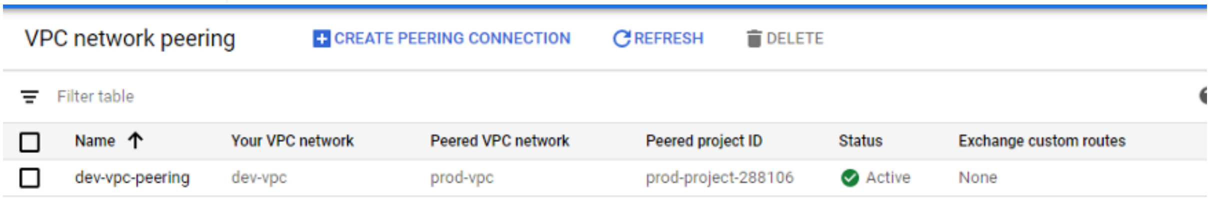
Your VPC network *
dev-vpc

Peered VPC network
○ In project dev-project-288106
● In another project

Project ID
prod-project-288106

VPC network name
prod-vpc

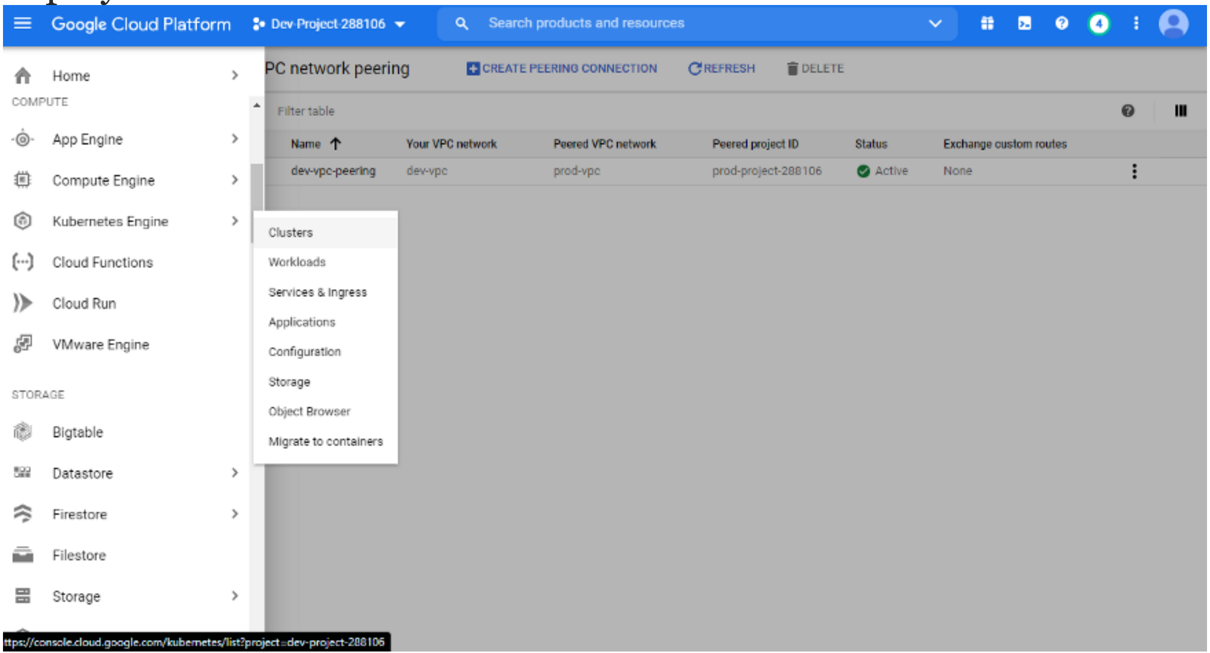
Creating the peering connection "dev-vpc-peering"...



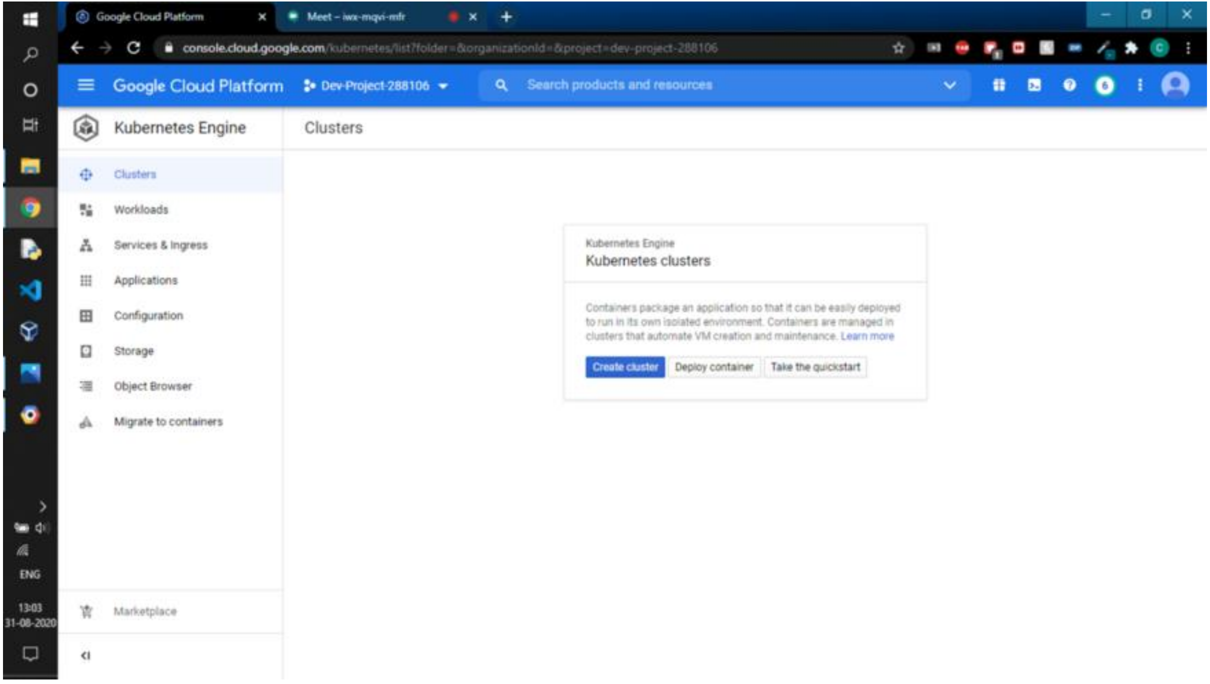
□	Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
□	dev-vpc-peering	dev-vpc	prod-vpc	prod-project-288106	✓ Active	None

→ Step 4:

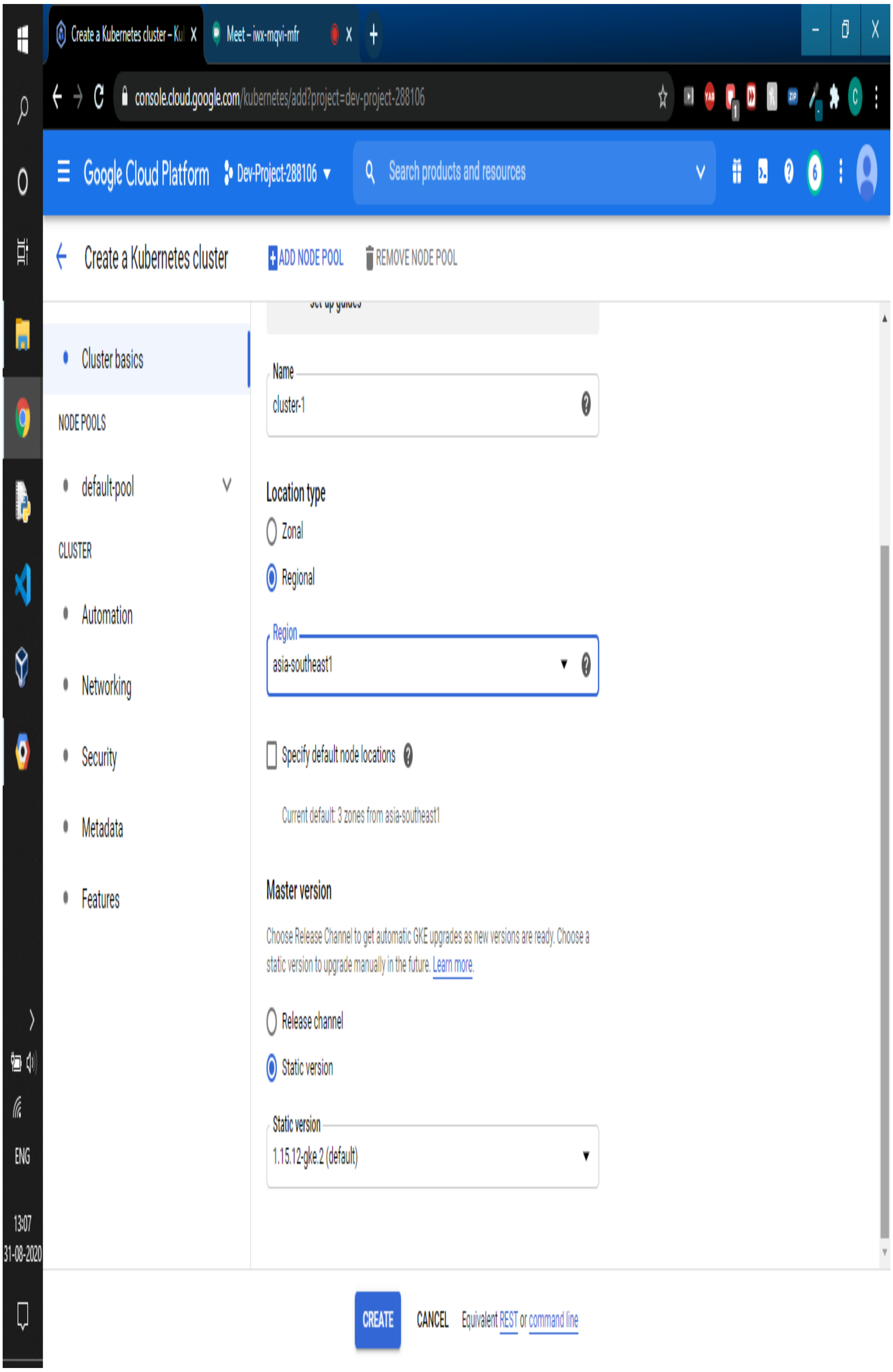
Now, we create a kubernertes cluster then launch a few Kubernetes pods. We decided to go for **GKE (Google Kubernetes Engine)** service, this will let us host a webserver via pods. Now we launch the webserver in a cluster so that we can use the cluster to deploy the server:



Name	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
dev-vpc-peering	dev-vpc	prod-vpc	prod-project-288106	Active	None



Containers package an application so that it can be easily deployed to run in its own isolated environment. Containers are managed in clusters that automate VM creation and maintenance. Learn more



Name
cluster-1

Location type

○ Zonal

● Regional

Region
asia-southeast1

□ Specify default node locations

Current default: 3 zones from asia-southeast1

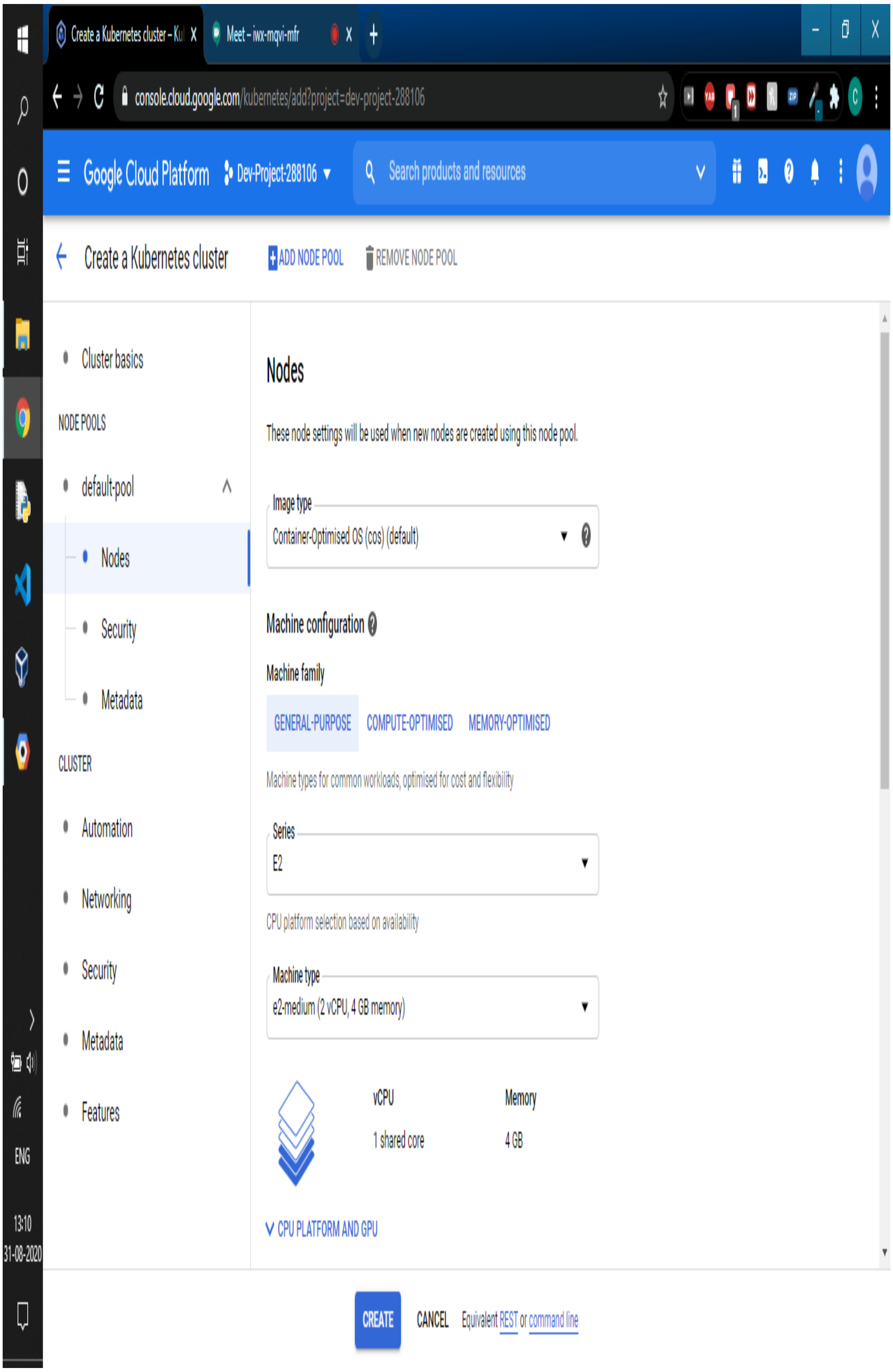
Master version

Choose Release Channel to get automatic GKE upgrades as new versions are ready. Choose a static version to upgrade manually in the future. Learn more.

○ Release channel

● Static version

Static version
1.15.12-gke.2 (default)



Nodes

These node settings will be used when new nodes are created using this node pool.

Image type: Container-Optimised OS (cos) (default)

Machine configuration

Machine family: GENERAL-PURPOSE | COMPUTE-OPTIMISED | MEMORY-OPTIMISED

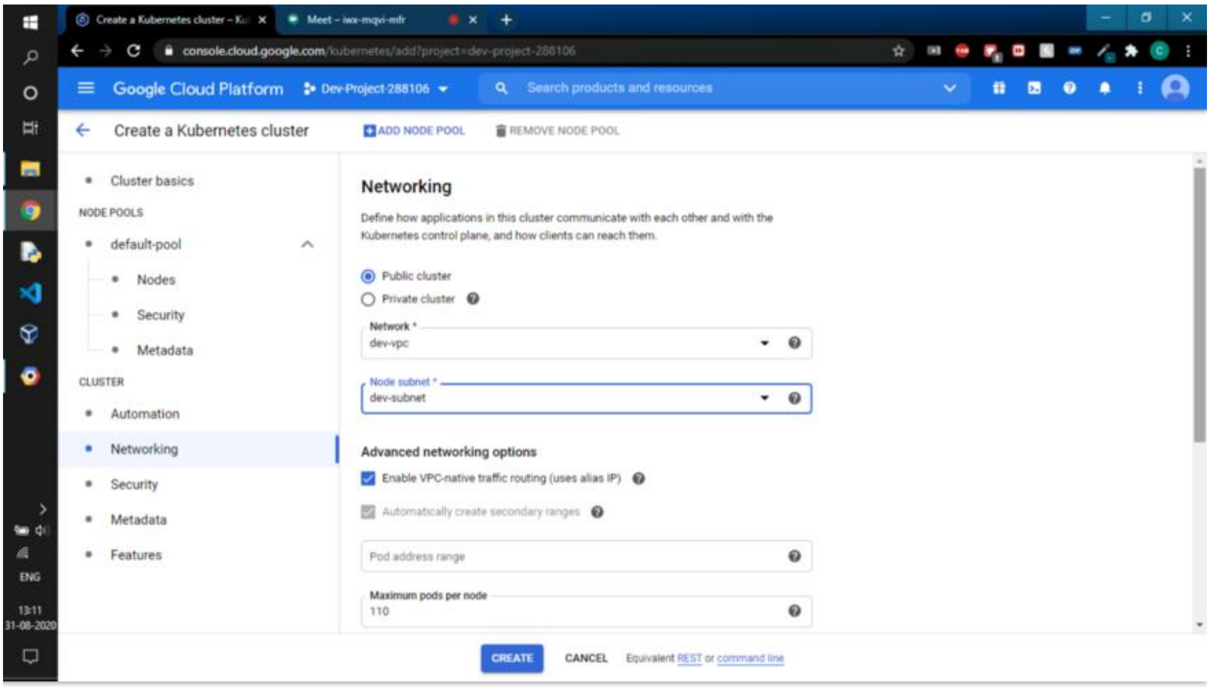
Machine types for common workloads, optimised for cost and flexibility

Series: E2

CPU platform selection based on availability

Machine type: e2-medium (2 vCPU, 4 GB memory)

vCPU	Memory
1 shared core	4 GB



Networking

Define how applications in this cluster communicate with each other and with the Kubernetes control plane, and how clients can reach them.

● Public cluster
○ Private cluster

Network *: dev-vpc

Node subnet *: dev-subnet

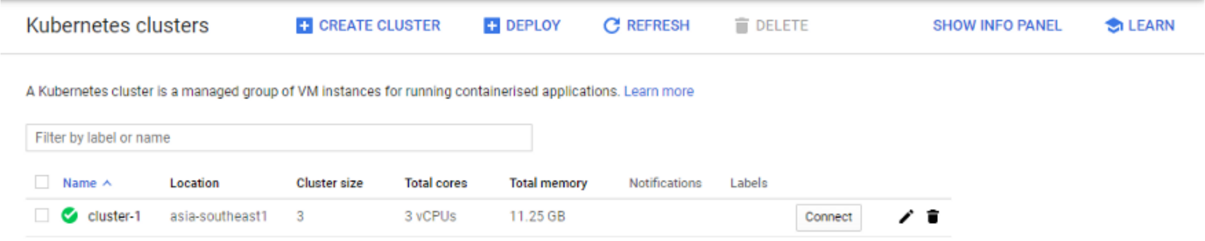
Advanced networking options

☑ Enable VPC-native traffic routing (uses alias IP)

☑ Automatically create secondary ranges

Pod address range

Maximum pods per node: 110

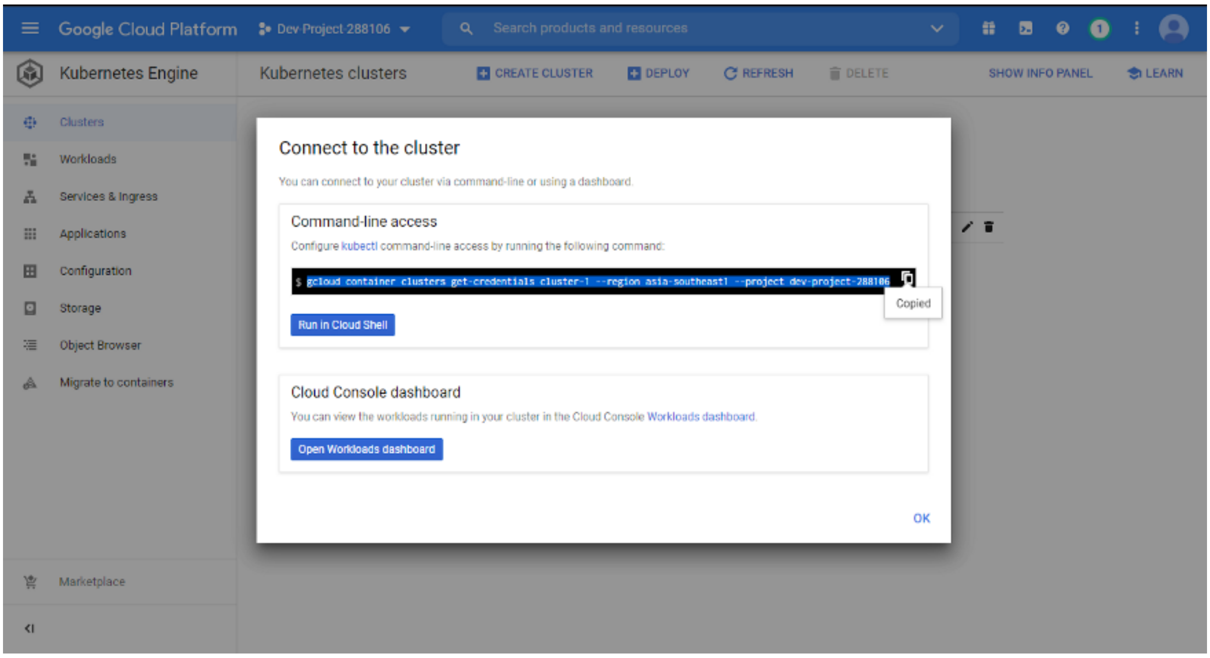


A Kubernetes cluster is a managed group of VM instances for running containerised applications. Learn more

Name	Location	Cluster size	Total cores	Total memory	Notifications	Labels	
✓ cluster-1	asia-southeast1	3	3 vCPUs	11.25 GB			Connect

Now that it’s created we can continue back at our Terminal in our system for convenience . This can be done when we create the cluster we also get an option to connect it back to our local system with Kubernetes installed

And the connect option lets you a prompt that will provide us a command which we’ll have to run it in the system’s command prompt or terminal



Connect to the cluster

You can connect to your cluster via command-line or using a dashboard.

Command-line access

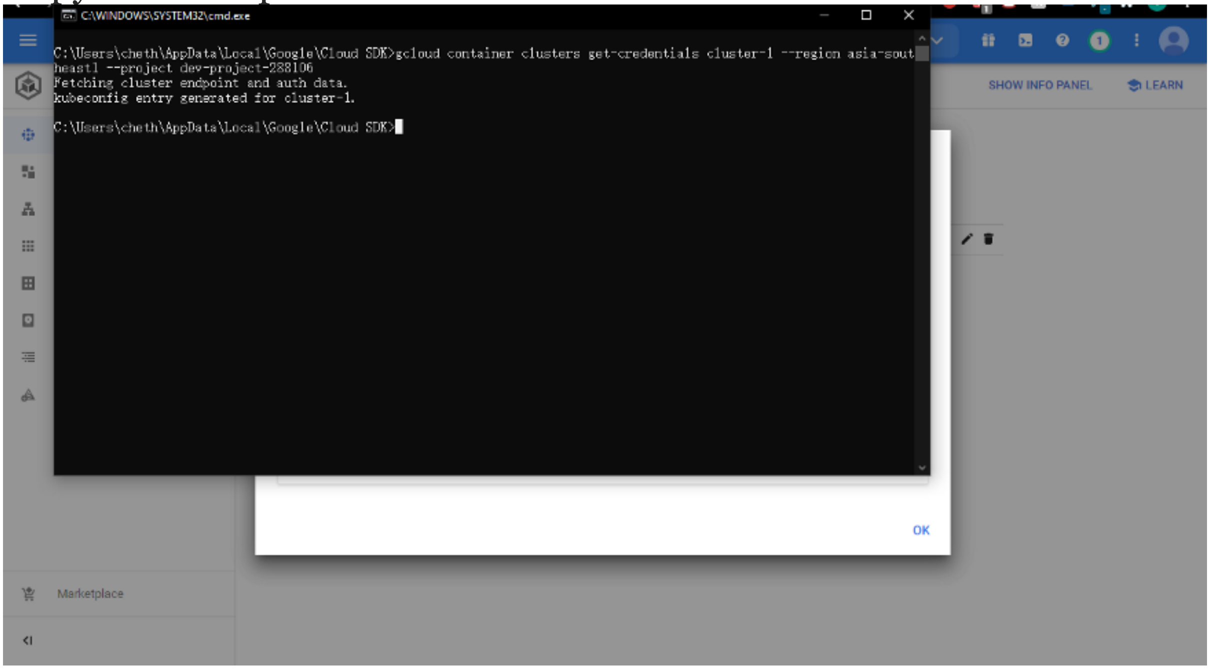
Configure kubectl command-line access by running the following command:

```
$ gcloud container clusters get-credentials cluster-1 --region asia-southeast1 --project dev-project-288106
```

Cloud Console dashboard

You can view the workloads running in your cluster in the Cloud Console Workloads dashboard.

Copy that and paste it in the terminal



```
C:\Users\cheth\AppData\Local\Google\Cloud SDK>gcloud container clusters get-credentials cluster-1 --region asia-sout
heast1 --project dev-project-288106
Fetching cluster endpoint and auth data.
kubeconfig entry generated for cluster-1.

C:\Users\cheth\AppData\Local\Google\Cloud SDK>
```

Now to start the Deployment procedure we do as follows:

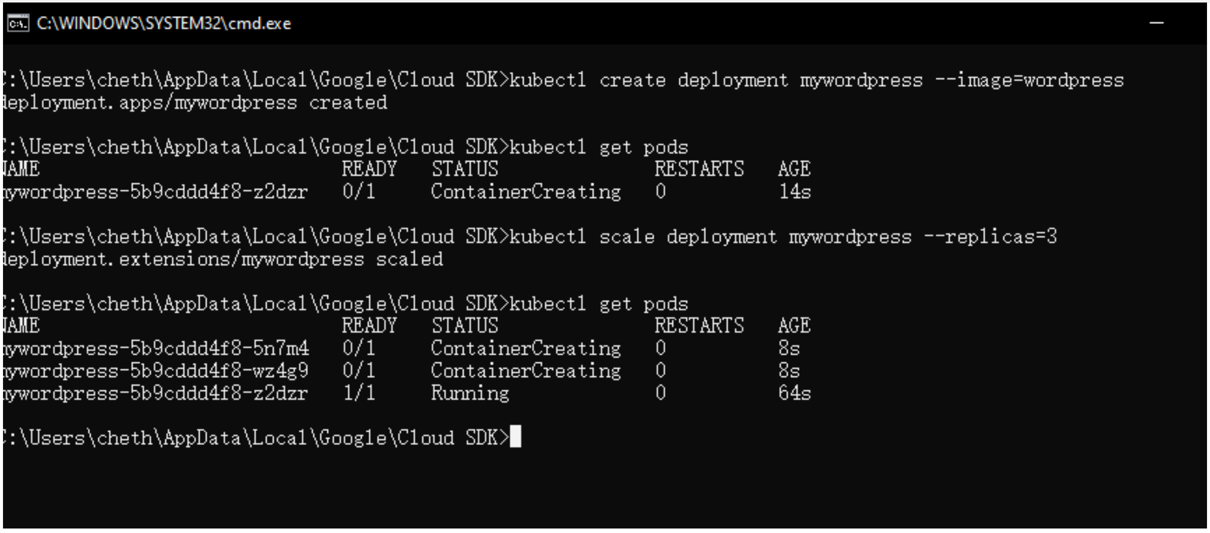
```
kubectl create deployment word-press --image=wordpress
```

You may wanna test whether they are successfully launched or not, that can be done by:

```
kubectl get pods
```

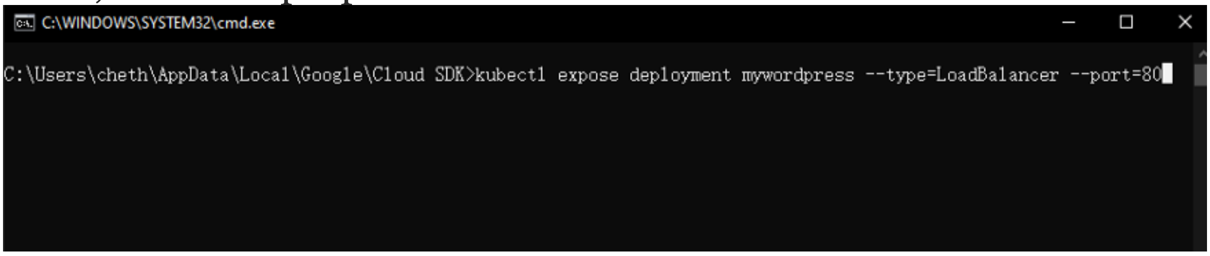
Now to create a couple of replicas , For more number of pods

```
kubectl scale deployment word-press --replicas=3
```



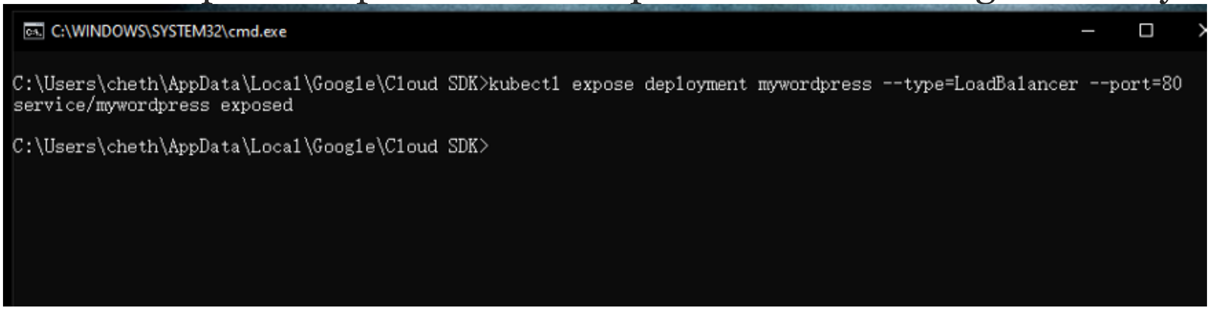
```
:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl create deployment mywordpress --image=wordpress
leployment.apps/mywordpress created

:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl get pods
JAME                          READY   STATUS              RESTARTS   AGE
nywordpress-5b9cddd4f8-z2dzr   0/1     ContainerCreating   0          14s

:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl scale deployment mywordpress --replicas=3
leployment.extensions/mywordpress scaled

:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl get pods
JAME                          READY   STATUS              RESTARTS   AGE
nywordpress-5b9cddd4f8-5n7m4   0/1     ContainerCreating   0          8s
nywordpress-5b9cddd4f8-wz4g9   0/1     ContainerCreating   0          8s
nywordpress-5b9cddd4f8-z2dzr   1/1     Running             0          64s

:\Users\cheth\AppData\Local\Google\Cloud SDK>
```

Now , to have a proper traffic control we create a LoadBalancer



```
C:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl expose deployment mywordpress --type=LoadBalancer --port=80
```

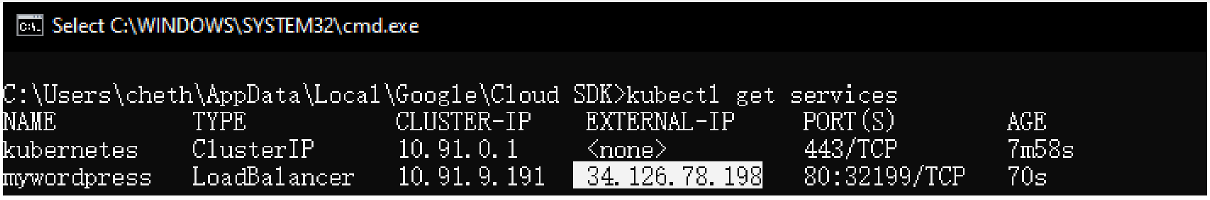
Then we expose the pods under the port 80 for accessing externally:



```
C:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl expose deployment mywordpress --type=LoadBalancer --port=80
service/mywordpress exposed

C:\Users\cheth\AppData\Local\Google\Cloud SDK>
```

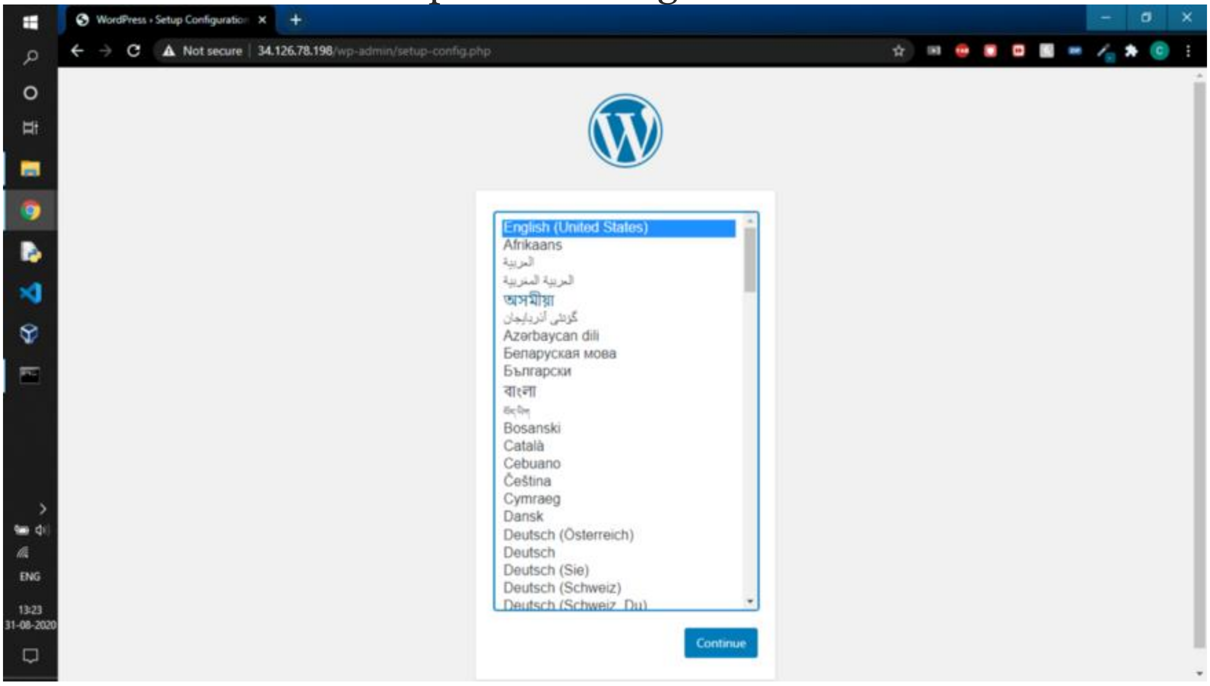
Now to access it via its IP

```
kubectl get services
```

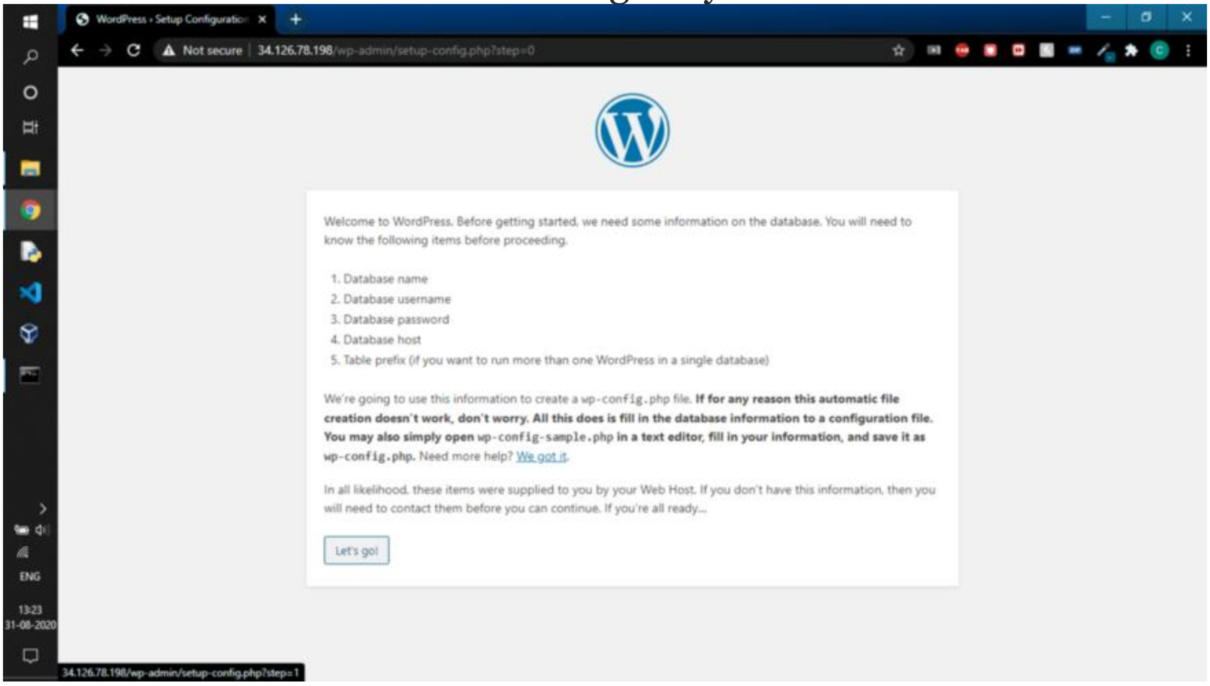


```
C:\Users\cheth\AppData\Local\Google\Cloud SDK>kubectl get services
NAME          TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)        AGE
kubernetes    ClusterIP      10.91.0.1     <none>          443/TCP        7m58s
nywordpress   LoadBalancer   10.91.9.191   34.126.78.198   80:32199/TCP   70s
```

Now our WebServer is up and running . But



We need a database before we can go any further…


WordPress › Setup Configuration
Not secure | 34.126.78.198/wp-admin/setup-config.php?step=0
ENG
13:23
31-08-2020
34.126.78.198/wp-admin/setup-config.php?step=1

Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. **If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? We got it.

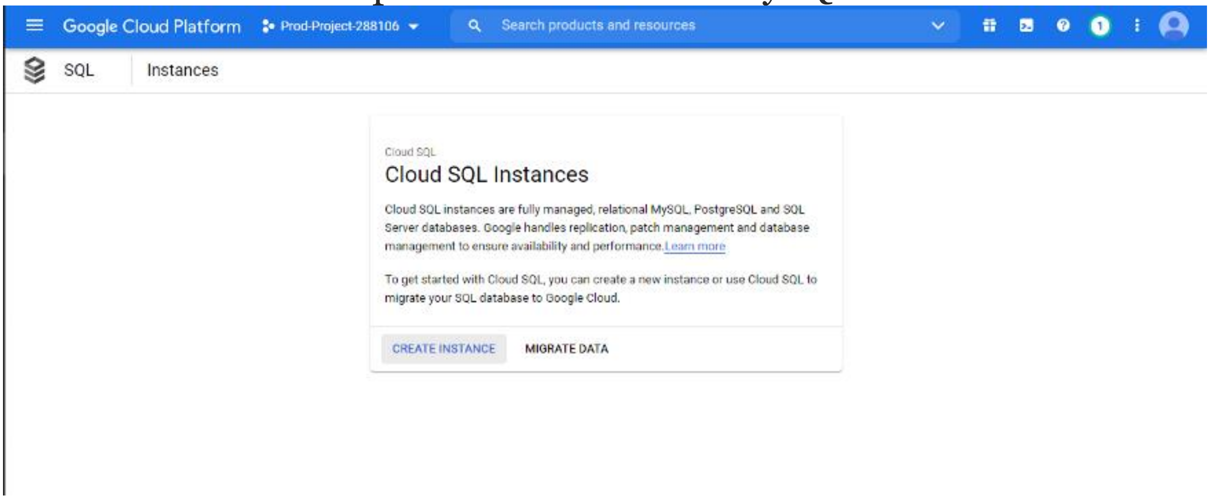
In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready…

Let's go!

→ Step 5:

Now we’ll create a database server which will be integrated into our WordPress server , So that the data used by the user that can be stored it in our database.

Within the prod-project we create a database server , so when we’ll launch te it then in a parallel it’ll create a MySQL instances for us.



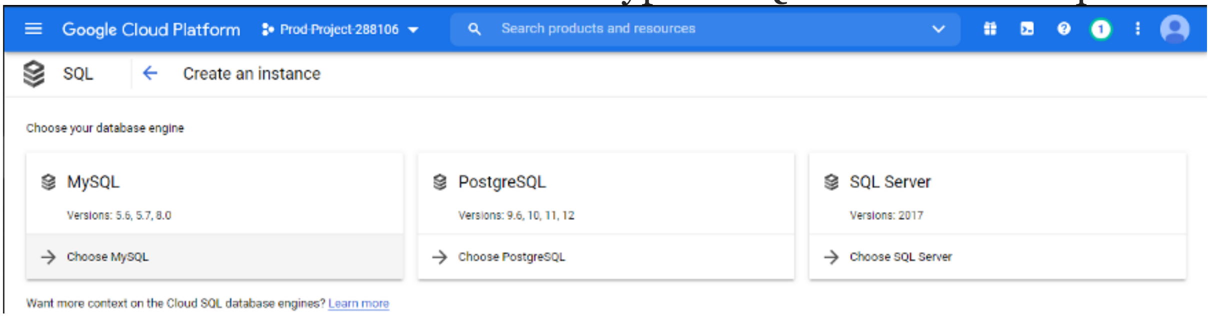
Cloud SQL

Cloud SQL Instances

Cloud SQL instances are fully managed, relational MySQL, PostgreSQL and SQL Server databases. Google handles replication, patch management and database management to ensure availability and performance.Learn more

To get started with Cloud SQL, you can create a new instance or use Cloud SQL to migrate your SQL database to Google Cloud.

Now we can choose whichever the type of SQL services we require:



Choose your database engine

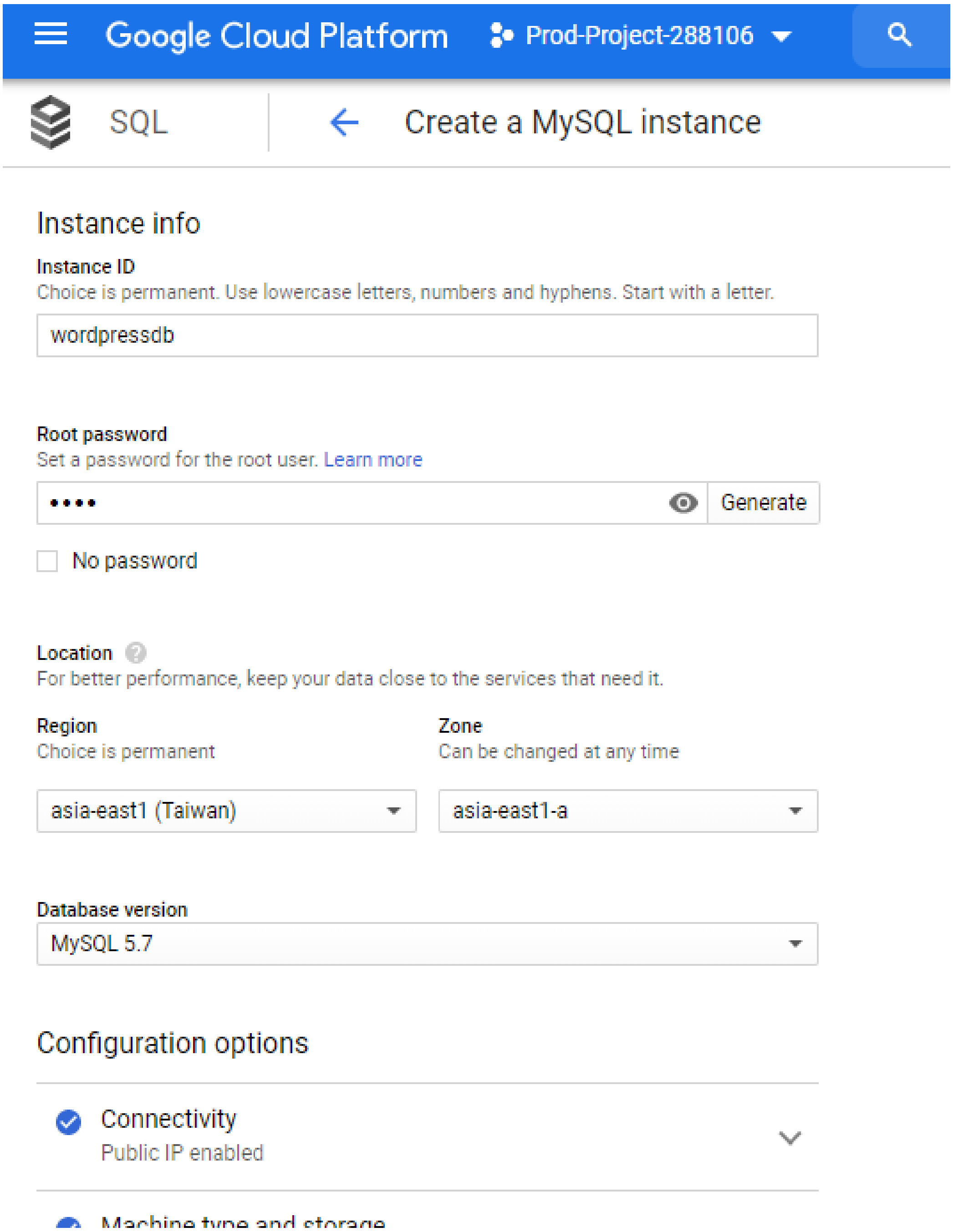
MySQL
Versions: 5.6, 5.7, 8.0

PostgreSQL
Versions: 9.6, 10, 11, 12

SQL Server
Versions: 2017

Want more context on the Cloud SQL database engines? Learn more

In our case, it’s ought to be MySQL server since it’s what backs up the WordPress server behind the scenes



Instance info

Instance ID

Choice is permanent. Use lowercase letters, numbers and hyphens. Start with a letter.

wordpressdb

Root password

Set a password for the root user. Learn more

••••

□ No password

Location

For better performance, keep your data close to the services that need it.

Region

Choice is permanent

asia-east1 (Taiwan)

Zone

Can be changed at any time

asia-east1-a

Database version

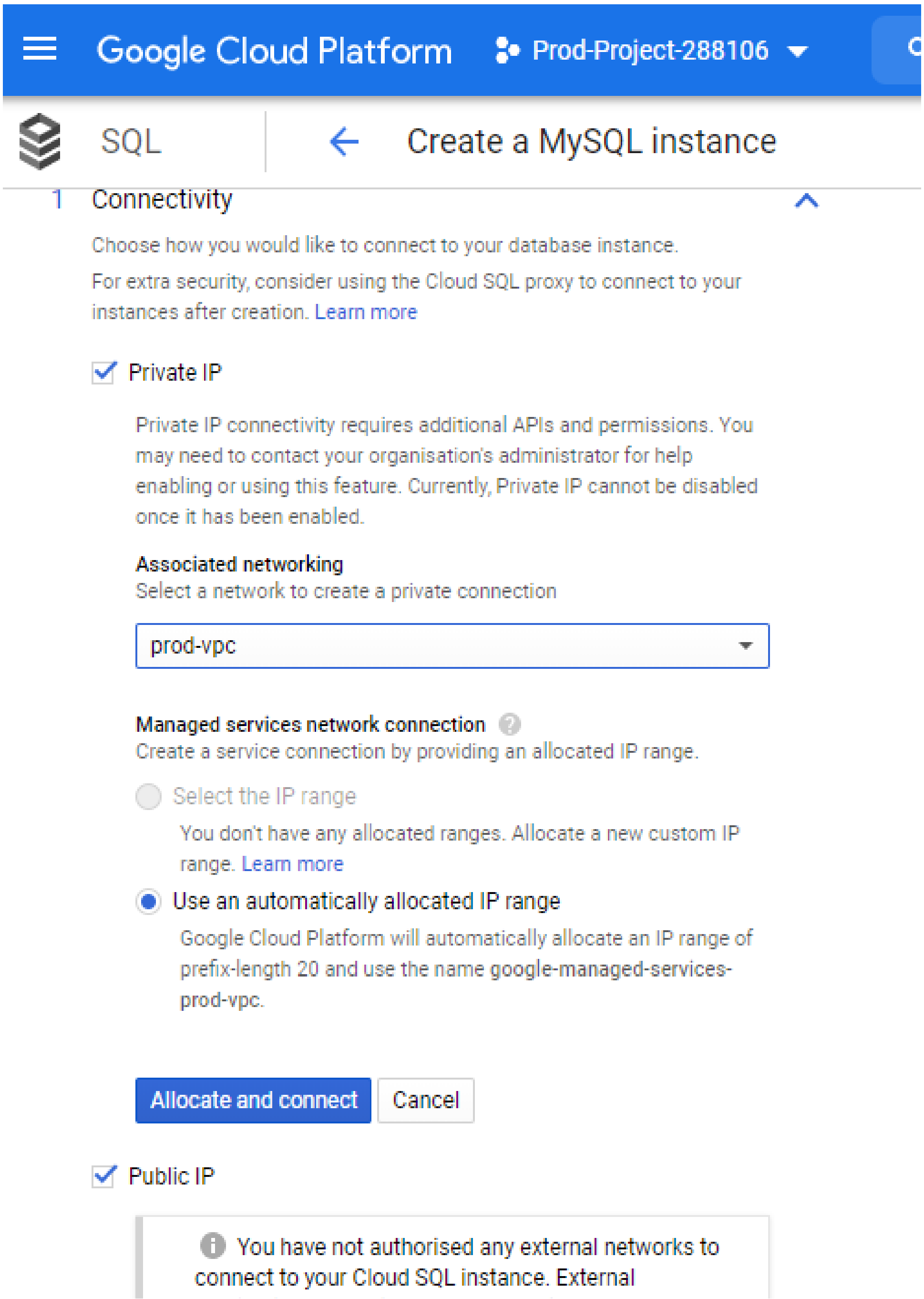
MySQL 5.7

Configuration options

Connectivity

Public IP enabled

Machine type and storage



Create a MySQL instance

1 Connectivity

Choose how you would like to connect to your database instance.

For extra security, consider using the Cloud SQL proxy to connect to your instances after creation. Learn more

☑ Private IP

Private IP connectivity requires additional APIs and permissions. You may need to contact your organisation's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

Associated networking
Select a network to create a private connection

prod-vpc

Managed services network connection
Create a service connection by providing an allocated IP range.

○ Select the IP range

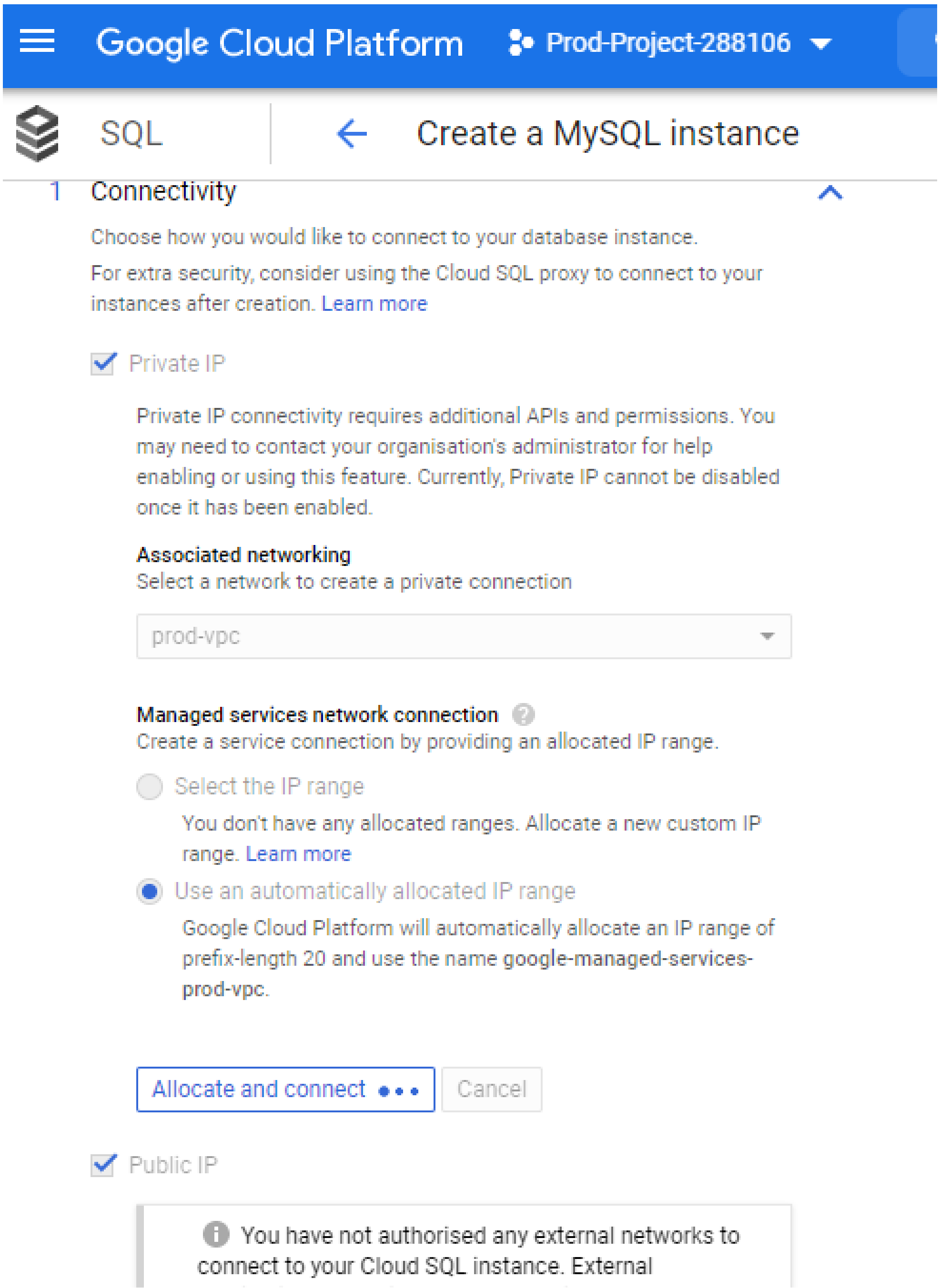
You don't have any allocated ranges. Allocate a new custom IP range. Learn more

● Use an automatically allocated IP range

Google Cloud Platform will automatically allocate an IP range of prefix-length 20 and use the name google-managed-services-prod-vpc.

☑ Public IP

You have not authorised any external networks to connect to your Cloud SQL instance. External



1 Connectivity

Choose how you would like to connect to your database instance.

For extra security, consider using the Cloud SQL proxy to connect to your instances after creation. Learn more

☑ Private IP

Private IP connectivity requires additional APIs and permissions. You may need to contact your organisation's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

Associated networking
Select a network to create a private connection

prod-vpc

Managed services network connection
Create a service connection by providing an allocated IP range.

○ Select the IP range

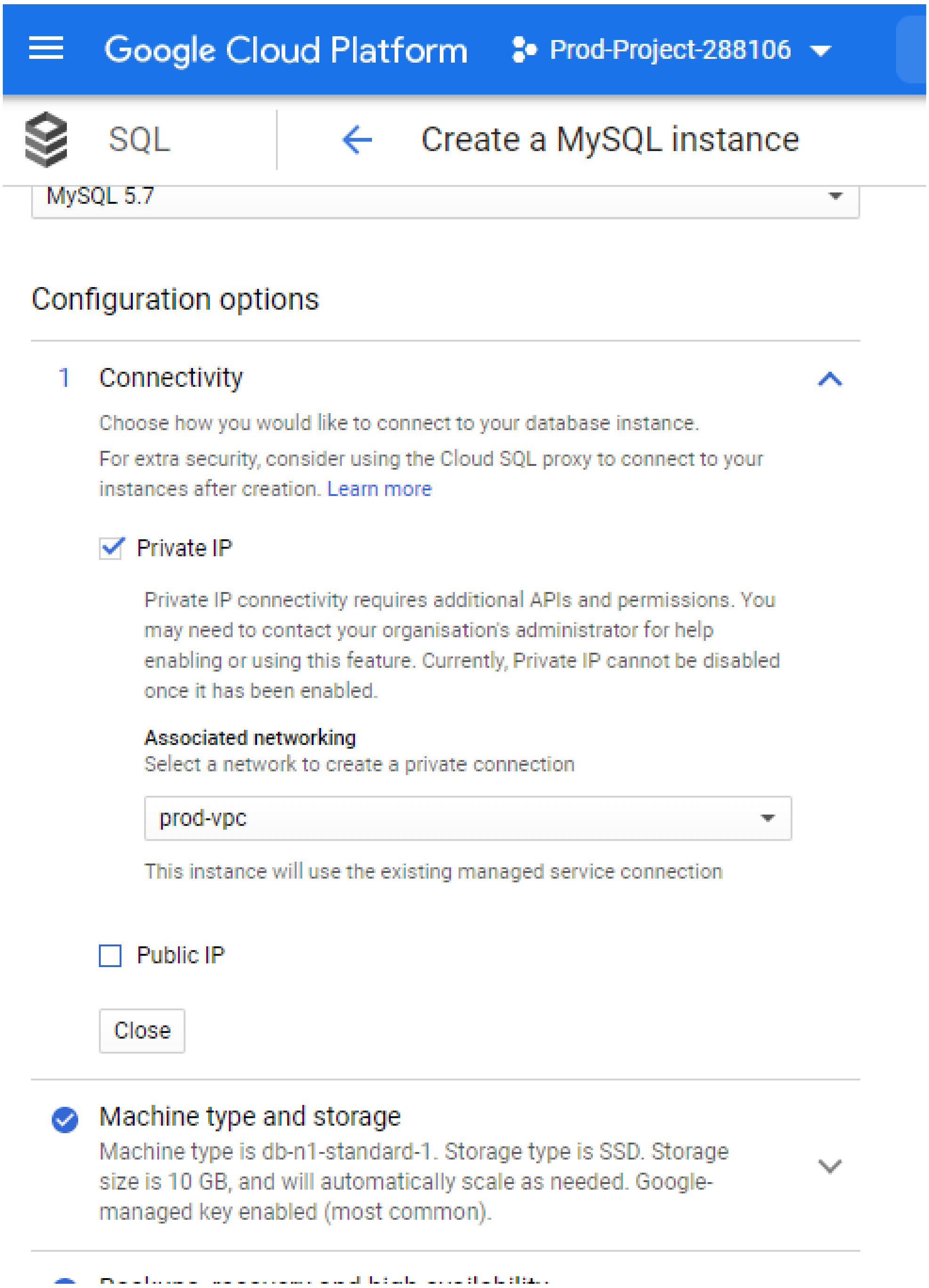
You don't have any allocated ranges. Allocate a new custom IP range. Learn more

● Use an automatically allocated IP range

Google Cloud Platform will automatically allocate an IP range of prefix-length 20 and use the name google-managed-services-prod-vpc.

☑ Public IP

You have not authorised any external networks to connect to your Cloud SQL instance. External



MySQL 5.7

Configuration options

1 Connectivity

Choose how you would like to connect to your database instance.

For extra security, consider using the Cloud SQL proxy to connect to your instances after creation. Learn more

☑ Private IP

Private IP connectivity requires additional APIs and permissions. You may need to contact your organisation's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

Associated networking
Select a network to create a private connection

prod-vpc

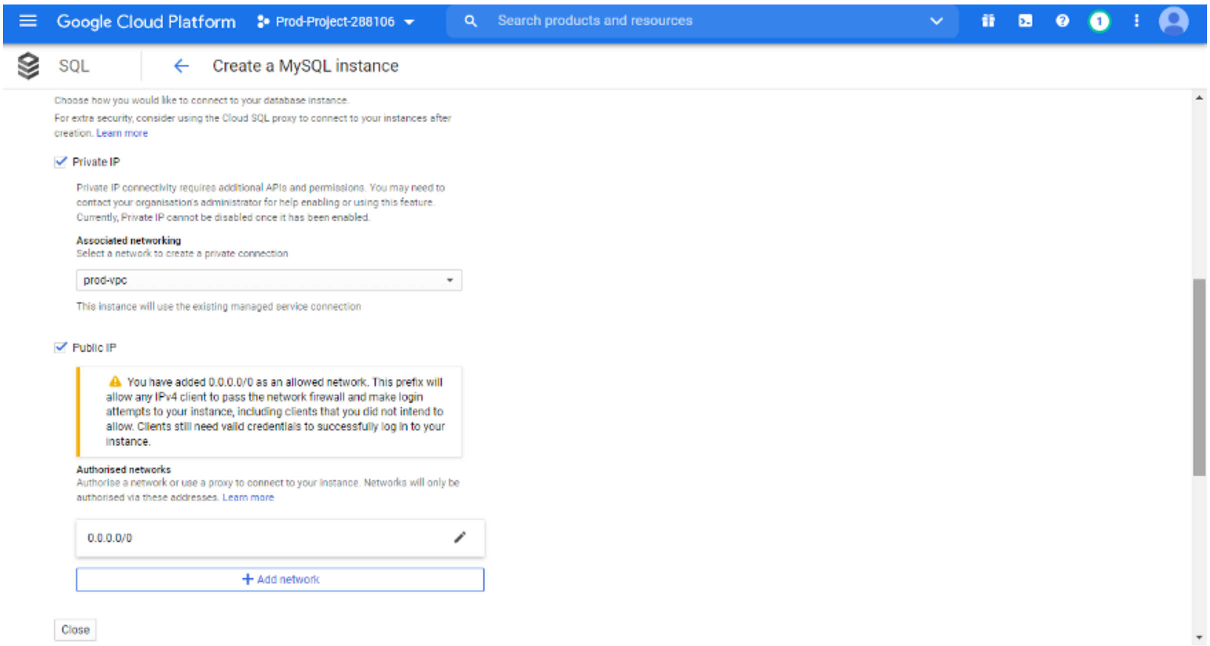
This instance will use the existing managed service connection

□ Public IP

Close

Machine type and storage
Machine type is db-n1-standard-1. Storage type is SSD. Storage size is 10 GB, and will automatically scale as needed. Google-managed key enabled (most common).

Backups, recovery and high availability



Choose how you would like to connect to your database instance.

For extra security, consider using the Cloud SQL proxy to connect to your instances after creation. Learn more

☑ Private IP

Private IP connectivity requires additional APIs and permissions. You may need to contact your organisation's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

Associated networking
Select a network to create a private connection

prod-vpc

This instance will use the existing managed service connection

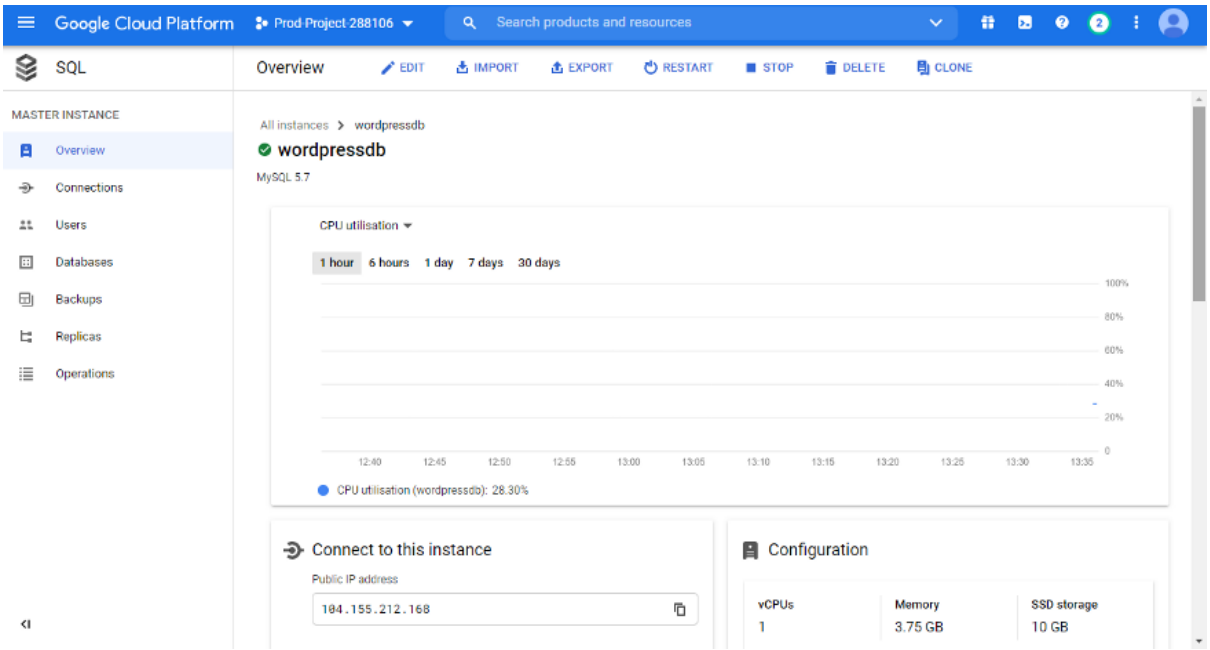
☑ Public IP

⚠ You have added 0.0.0.0/0 as an allowed network. This prefix will allow any IPv4 client to pass the network firewall and make login attempts to your instance, including clients that you did not intend to allow. Clients still need valid credentials to successfully log in to your instance.

Authorised networks
Authorise a network or use a proxy to connect to your instance. Networks will only be authorised via these addresses. Learn more

0.0.0.0/0

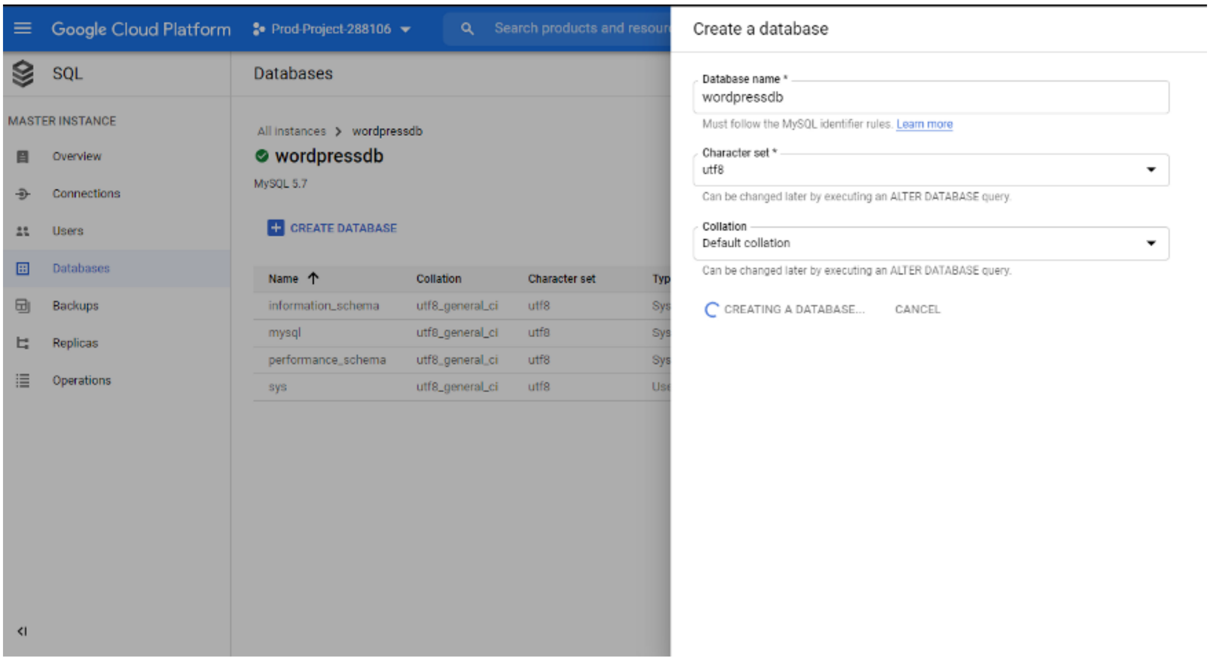
And with that the instance will be created:



Public IP address: 104.155.212.168

vCPUs	Memory	SSD storage
1	3.75 GB	10 GB

Now that our MySQL server is created,we can make some modifications to our database (like : setting up some network changes , allow users and so on)



Name	Collation	Character set	Typ
information_schema	utf8_general_ci	utf8	Sys
mysql	utf8_general_ci	utf8	Sys
performance_schema	utf8_general_ci	utf8	Sys
sys	utf8_general_ci	utf8	Use

Database name *: wordpressdb

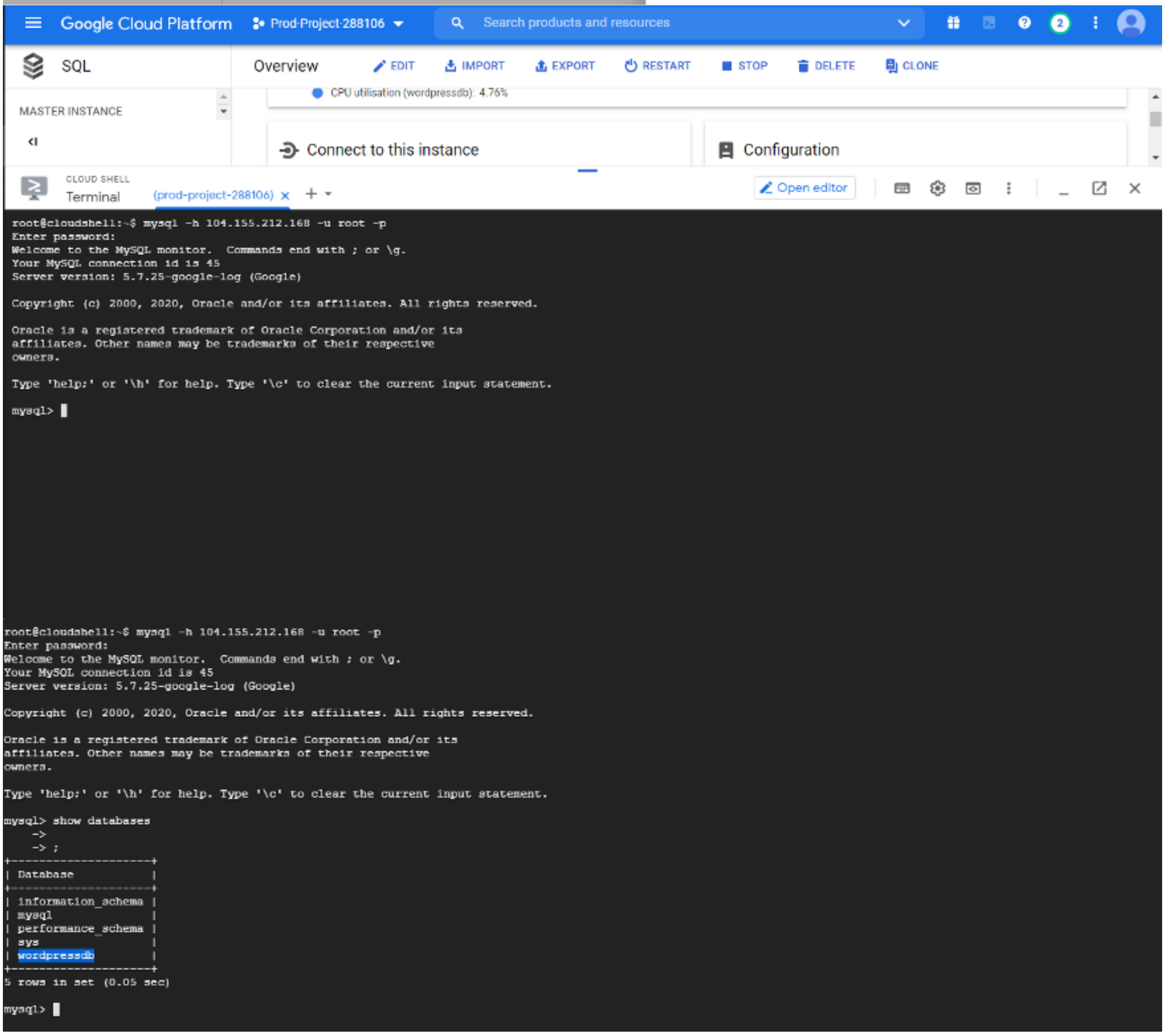
Must follow the MySQL identifier rules. Learn more

Character set *: utf8

Can be changed later by executing an ALTER DATABASE query.

Collation: Default collation

Can be changed later by executing an ALTER DATABASE query.

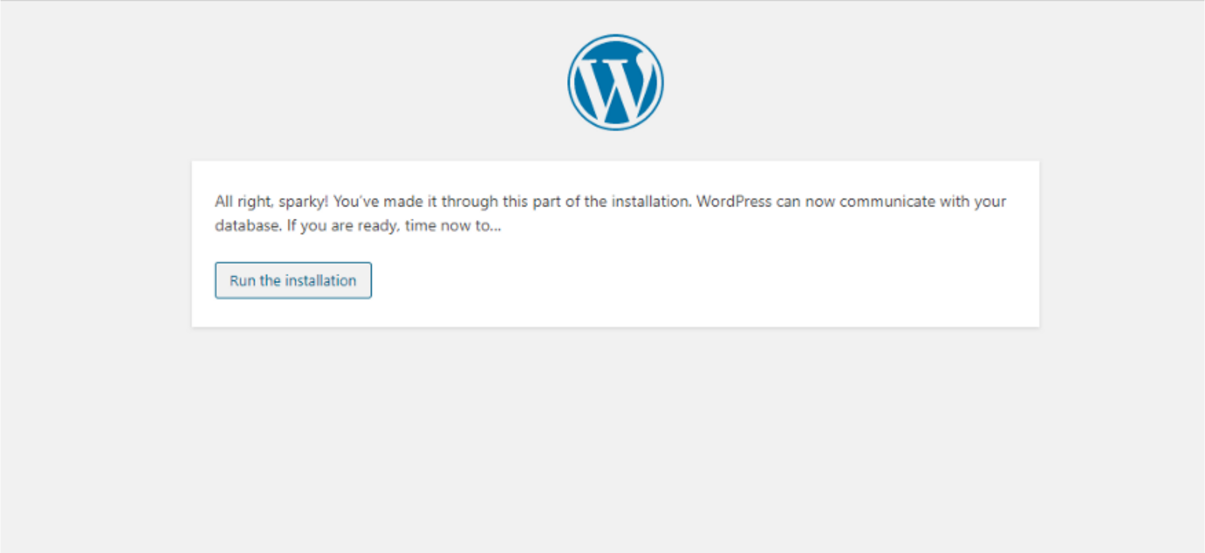


```
root@cloudshell:~$ mysql -h 104.155.212.168 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.7.25-google-log (Google)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

```
root@cloudshell:~$ mysql -h 104.155.212.168 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.7.25-google-log (Google)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    ->
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpressdb        |
+--------------------+
5 rows in set (0.05 sec)

mysql> 
```

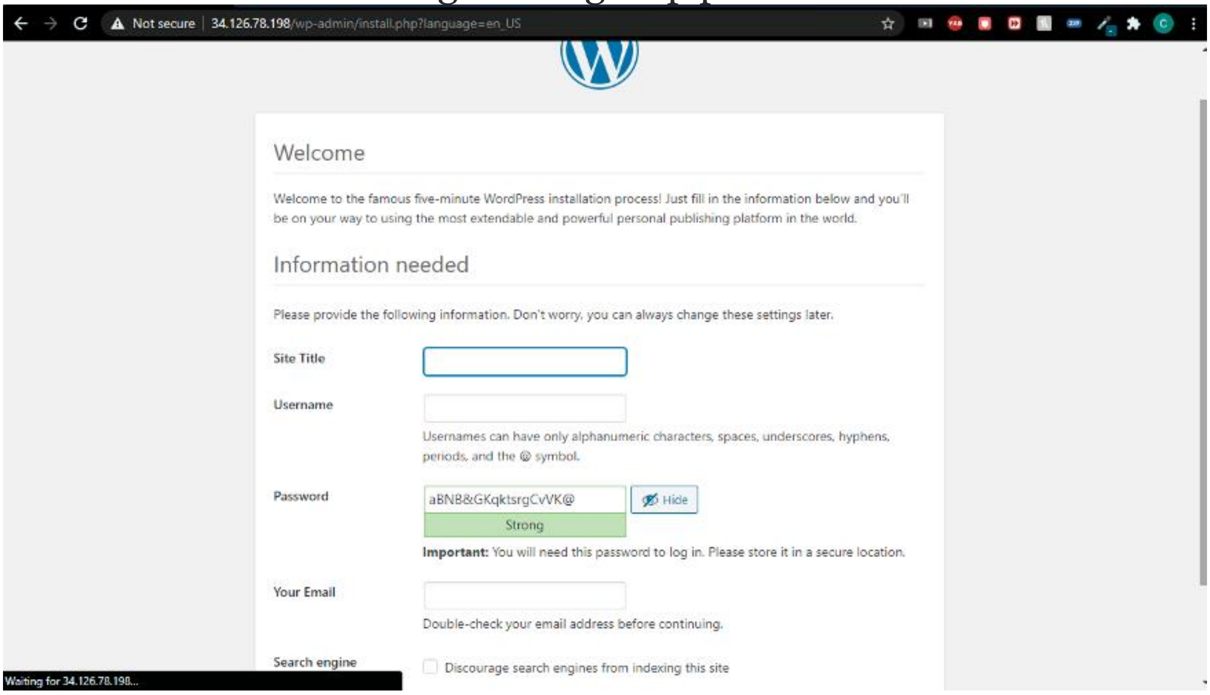
Then we add usernames for connecting with the WordPress..

Now, we get back to WordPress pod which we launched a bit earlier



All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to…

Now we can follow through the sign-up process :



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: aBNB&GKqktsrgCvVK@

Important: You will need this password to log in. Please store it in a secure location.

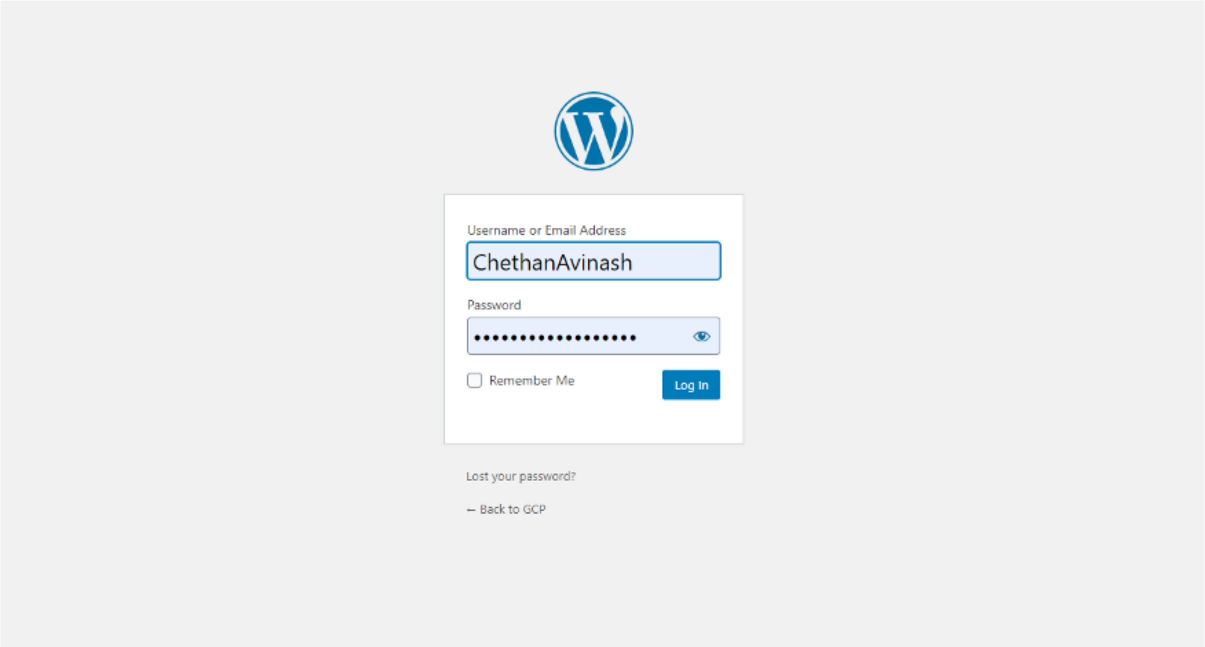
Your Email

Double-check your email address before continuing.

Search engine: □ Discourage search engines from indexing this site

Waiting for 34.126.78.198...

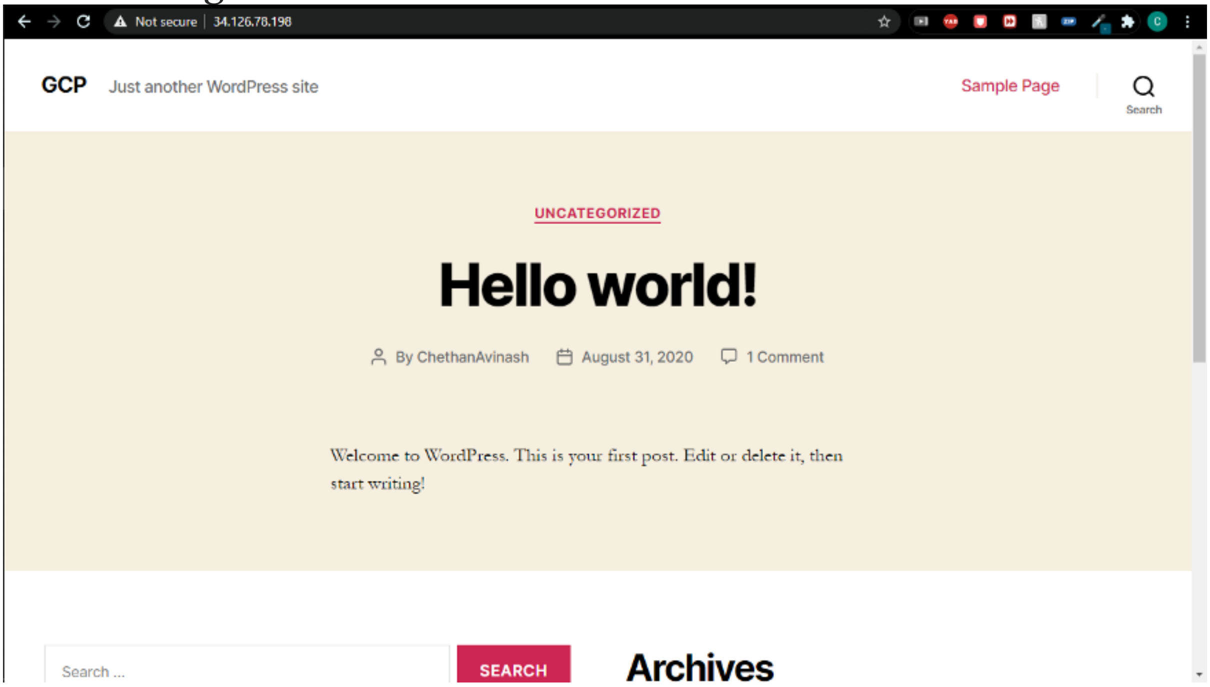
Then Log-in :



Username or Email Address: ChethanAvinash

Password: ••••••••••••••••••

And proceed to what we are gonna do in WordPress . Which is , to create a blog:



Not secure | 34.126.78.198

GCP Just another WordPress site

Hello world!

By ChethanAvinash August 31, 2020 1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Now, the task is complete as we have created two different networks and peered them together with one having a cluster of WordPress pods connected to a back-end MySQL server.

Conclusion:

And that’s how easy it is, to deploy Wordpress on Google Cloud using GKE