

GOOGLE CLOUD PROJECT DEPLOY WEB APP



Google Cloud

USING GCP



Google Cloud Platform

By: Ayush Milan

Project Work

Task 1. Create multiple projects namely developer and production

2 Projects with Project name and respective Project ID will be generated

https://console.cloud.google.com/projectcreate?previousPage=%2Fhome%2Fdashboard%3Fproject%3Ddeveloper-289215&folder=...

Google Cloud Platform Search products and resources

New Project

You have 10 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)
[MANAGE QUOTAS](#)

Project name *
production

Project ID: production-289215. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Activate Windows
Go to Settings to activate Windows.

https://console.cloud.google.com/projectcreate?previousPage=%2Fcloud-resource-manager%3Ffolder%3D%26organizationId%3D0...

Google Cloud Platform Search products and resources

New Project

You have 10 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)
[MANAGE QUOTAS](#)

Project name *
developer

Project ID: dark-runway-289215. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Activate Windows
Go to Settings to activate Windows.

After creating the project we have to enable the billing for the projects and enable compute engine API ,so that we create the VPC networks.

The screenshot shows the Google Cloud Platform console interface. The browser address bar displays the URL: <https://console.cloud.google.com/marketplace/product/google/compute.googleapis.com?returnUrl=%2Fnetworking%3Ffolder%3D...>. The page header includes the Google Cloud Platform logo, a dropdown menu set to 'developer', and a search bar. The main content area features the 'Compute Engine API' icon and title, with 'Google' as the provider. Below this are 'ENABLE' and 'TRY THIS API' buttons. A navigation bar shows 'OVERVIEW', 'DOCUMENTATION', and 'SUPPORT' tabs, with 'OVERVIEW' selected. The 'Overview' section contains the text: 'Creates and runs virtual machines on Google Cloud Platform.' and 'About Google' with a paragraph about Google's mission. The 'Additional details' section lists: 'Type: APIs & services', 'Last updated: 12/10/19', 'Category: Compute, Networking', and 'Service name: compute.googleapis.com'. An 'Activate Windows' watermark is visible in the bottom right corner.

This screenshot is identical to the one above, showing the 'Compute Engine API' page. However, the dropdown menu in the header is now set to 'production'. The URL in the address bar is truncated: <https://console.cloud.google.com/marketplace/product/google/compute.googleapis.com?returnUrl=%2Fnetworking%3Fnetworks%...>. All other elements, including the 'ENABLE' button, navigation tabs, and 'Overview' content, remain the same.

Task 2. Create VPC network for both the projects

Now we have to create vpc ,here i create vpc for developer project in asia-southeast1 region as lab1

Google Cloud Platform console showing the process of creating a VPC network and its subnets.

Top Screenshot: Create a VPC network

The "Create a VPC network" page shows the "Name" field set to "vpcdeveloper" and the "Subnet creation mode" set to "Custom".

Bottom Screenshot: New subnet configuration

The "New subnet" page shows the "Name" field set to "lab1", the "Region" set to "asia-southeast1", and the "IP address range" set to "10.0.1.0/24". The "Private Google access" and "Flow logs" options are both set to "Off".

Bottom Screenshot: VPC networks list

The "VPC networks" list shows the newly created network "vpcdeveloper" with its subnets.

Network	Subnet	Region	IP Address Range	Private Google Access	Flow Logs
asia-south1	default	asia-south1	10.160.0.0/20	Off	Off
northamerica-northeast1	default	northamerica-northeast1	10.162.0.0/20	Off	Off
europa-west4	default	europa-west4	10.164.0.0/20	Off	Off
europa-north1	default	europa-north1	10.166.0.0/20	Off	Off
us-west2	default	us-west2	10.168.0.0/20	Off	Off
asia-east2	default	asia-east2	10.170.0.0/20	Off	Off
europa-west6	default	europa-west6	10.172.0.0/20	Off	Off
asia-northeast2	default	asia-northeast2	10.174.0.0/20	Off	Off
asia-northeast3	default	asia-northeast3	10.176.0.0/20	Off	Off
us-west3	default	us-west3	10.180.0.0/20	Off	Off
us-west4	default	us-west4	10.182.0.0/20	Off	Off
asia-southeast2	default	asia-southeast2	10.184.0.0/20	Off	Off
vpcdeveloper	lab1	asia-southeast1	10.0.1.0/24	Off	Off

For another project “production” we create vpc in us-east1 region as lab2

Google Cloud Platform console showing the process of creating a VPC network and listing existing VPC networks.

Top Screenshot: Create a VPC network

The "Create a VPC network" page shows the following configuration:

- Name:** vpcproduction
- Description:** (empty)
- 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 (selected), Automatic
- New subnet:**
 - Name:** lab2
 - Region:** us-east1
 - IP address range:** 10.0.2.0/24
 - CREATE SECONDARY IP RANGE:** (button)
 - Private Google access:** Off (selected), On
 - Flow logs:** Off (selected), On

Bottom Screenshot: VPC networks list

The "VPC networks" list shows the following table:

Name	Region	Subnet	IP address range	Secondary IP range	Private Google access	Flow logs
asia-south1	default	10.160.0.0/20	10.160.0.1		Off	
northamerica-northeast1	default	10.162.0.0/20	10.162.0.1		Off	
europa-west4	default	10.164.0.0/20	10.164.0.1		Off	
europa-north1	default	10.166.0.0/20	10.166.0.1		Off	
us-west2	default	10.168.0.0/20	10.168.0.1		Off	
asia-east2	default	10.170.0.0/20	10.170.0.1		Off	
europa-west6	default	10.172.0.0/20	10.172.0.1		Off	
asia-northeast2	default	10.174.0.0/20	10.174.0.1		Off	
asia-northeast3	default	10.178.0.0/20	10.178.0.1		Off	
us-west3	default	10.180.0.0/20	10.180.0.1		Off	
us-west4	default	10.182.0.0/20	10.182.0.1		Off	
asia-southeast2	default	10.184.0.0/20	10.184.0.1		Off	
vpcproduction	1	Custom	0	Off		
us-east1	lab2	10.0.2.0/24	10.0.2.1		Off	

Now we have to create firewall in both the vpc network

← → ↻ 🔒 https://console.cloud.google.com/networking/firewalls/add?network=vpcdeveloper&project=dark-runway-289215&supportedpurvie... ☆ 📄 🏠 👤 ...

Google Cloud Platform developer 🔍 Search products and resources

VPC network

- VPC networks
- External IP addresses
- Firewall
- Routes
- VPC network peering
- Shared VPC
- Serverless VPC access
- Packet mirroring

← Create a firewall rule

Network *
vpcdeveloper

Priority *
1000
Priority can be 0 - 65535 [Check priority of other firewall rules](#)

Direction of traffic
☒ Ingress
☐ Egress

Action on match
☒ Allow
☐ Deny

Targets
All instances in the network

Source filter
IP ranges

Source IP ranges *
0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter
None

Activate Windows
Go to Settings to activate Windows.

← → ↻ 🔒 https://console.cloud.google.com/networking/firewalls/add?network=vpcproduction&project=production-289215&supportedpurvie... ☆ 📄 🏠 👤 ...

Google Cloud Platform production 🔍 Search products and resources

VPC network

- VPC networks
- External IP addresses
- Firewall
- Routes
- VPC network peering
- Shared VPC
- Serverless VPC access
- Packet mirroring

← Create a firewall rule

Network *
vpcproduction

Priority *
1000
Priority can be 0 - 65535 [Check priority of other firewall rules](#)

Direction of traffic
☒ Ingress
☐ Egress

Action on match
☒ Allow
☐ Deny

Targets
All instances in the network

Source filter
IP ranges

Source IP ranges *
0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter
None

Activate Windows
Go to Settings to activate Windows.

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

At first we have to create peering connection in both the projects and share the project id and vpc network in each other projects:

The screenshot shows the Google Cloud Platform console for project 'dark-runway-289215'. The left sidebar shows the 'VPC network' section with 'VPC network peering' selected. The main content area is titled 'Create peering connection'. A message states: '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.' The form fields are: Name (developerpeering), Your VPC network (vpcdeveloper), Peered VPC network (In another project), Project ID (production-289215), and VPC network name (vpcproduction). A button 'EXCHANGE CUSTOM ROUTES' is visible. A status bar at the bottom shows 'CREATING' and a progress indicator.

The screenshot shows the Google Cloud Platform console for project 'production-289215'. The left sidebar shows the 'VPC network' section with 'VPC network peering' selected. The main content area is titled 'Create peering connection'. A message states: '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.' The form fields are: Name (productionpeering), Your VPC network (vpcproduction), Peered VPC network (In another project), Project ID (dark-runway-289215), and VPC network name (vpcdeveloper). A button 'EXCHANGE CUSTOM ROUTES' is visible. A 'CREATE' button is at the bottom.

So, now our VPC peering is established between the vpc network of both the projects and showing as active.

← → ↻ 🔒 https://console.cloud.google.com/networking/peering/list?project=dark-runway-289215

Google Cloud Platform developer 🔍 Search products and resources

VPC network VPC network peering + CREATE PEERING CONNECTION ↻ REFRESH 🗑 DELETE

Filter table

<input type="checkbox"/>	Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
<input type="checkbox"/>	developerpeering	vpcdeveloper	vpcproduction	production-289215	Active	None

Now viewing project "developer" in organization "No organization"

Activate Windows
Go to Settings to activate Windows.

← → ↻ 🔒 https://console.cloud.google.com/networking/peering/list?project=production-289215

Google Cloud Platform production 🔍 Search products and resources

VPC network VPC network peering + CREATE PEERING CONNECTION ↻ REFRESH 🗑 DELETE

Filter table

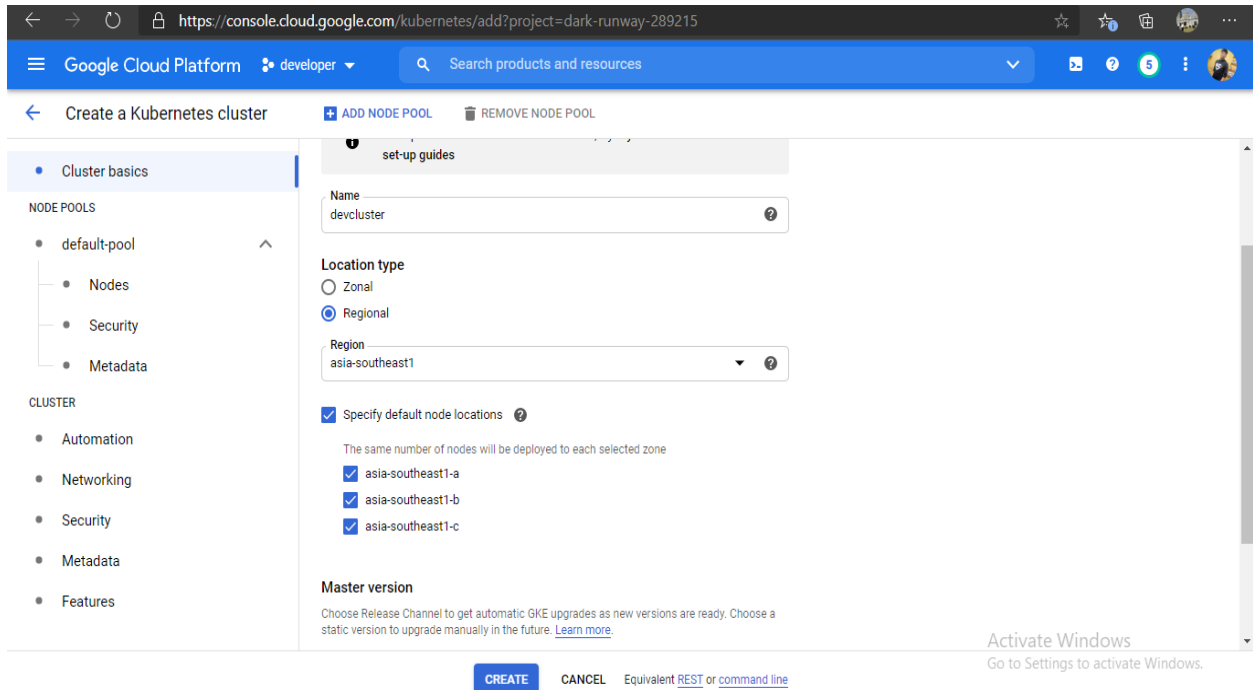
<input type="checkbox"/>	Name ↑	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
<input type="checkbox"/>	productionpeering	vpcproduction	vpcdeveloper	dark-runway-289215	Active	None

Now viewing project "production" in organization "No organization"

Activate Windows
Go to Settings to activate Windows.

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

At first create a Kubernetes Cluster in the project developer:



Google Cloud Platform developer Search products and resources

Create a Kubernetes cluster

Cluster basics

NODE POOLS

- default-pool
 - Nodes
 - Security
 - Metadata

CLUSTER

- Automation
- Networking
- Security
- Metadata
- Features

set-up guides

Name: devcluster

Location type: ☐ Zonal ☒ Regional

Region: asia-southeast1

☒ Specify default node locations

The same number of nodes will be deployed to each selected zone

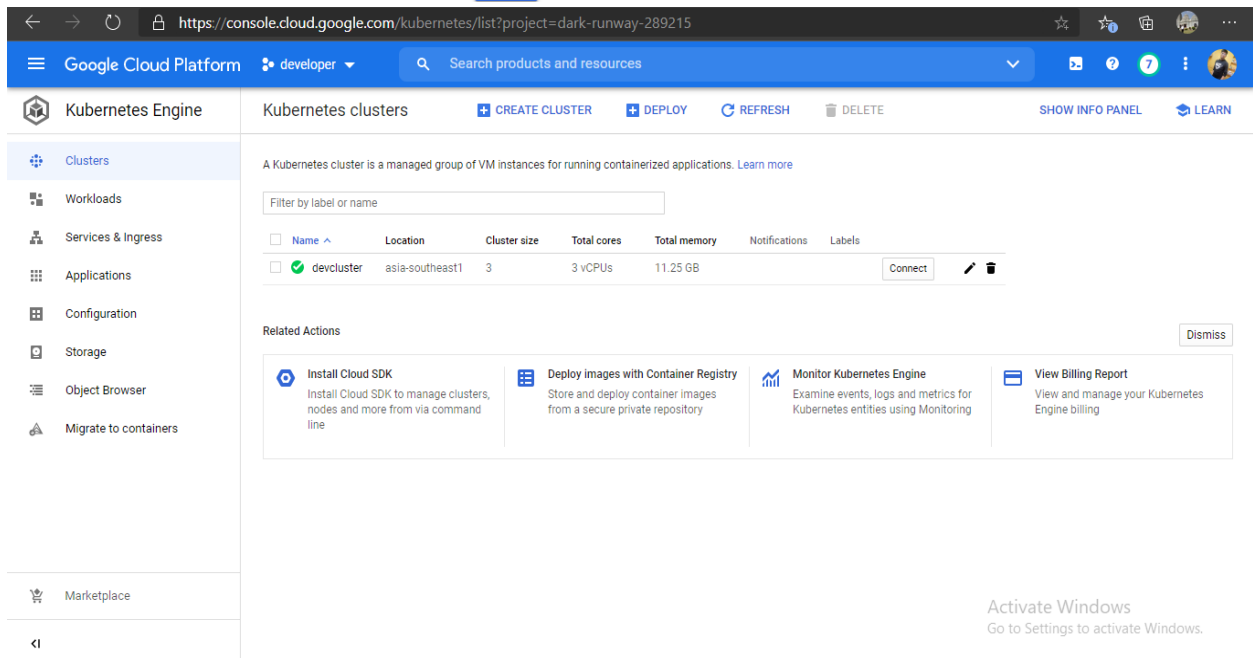
- ☒ asia-southeast1-a
- ☒ asia-southeast1-b
- ☒ asia-southeast1-c

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](#)

CREATE CANCEL Equivalent REST or command line

Activate Windows
Go to Settings to activate Windows.



Google Cloud Platform developer Search products and resources

Kubernetes Engine

Kubernetes clusters

CREATE CLUSTER DEPLOY REFRESH DELETE SHOW INFO PANEL LEARN

A Kubernetes cluster is a managed group of VM instances for running containerized applications. [Learn more](#)

Filter by label or name

<input type="checkbox"/> Name ^	Location	Cluster size	Total cores	Total memory	Notifications	Labels
<input checked="" type="checkbox"/> devcluster	asia-southeast1	3	3 vCPUs	11.25 GB		

Connect

Related Actions

- [Install Cloud SDK](#)
Install Cloud SDK to manage clusters, nodes and more from via command line
- [Deploy images with Container Registry](#)
Store and deploy container images from a secure private repository
- [Monitor Kubernetes Engine](#)
Examine events, logs and metrics for Kubernetes entities using Monitoring
- [View Billing Report](#)
View and manage your Kubernetes Engine billing

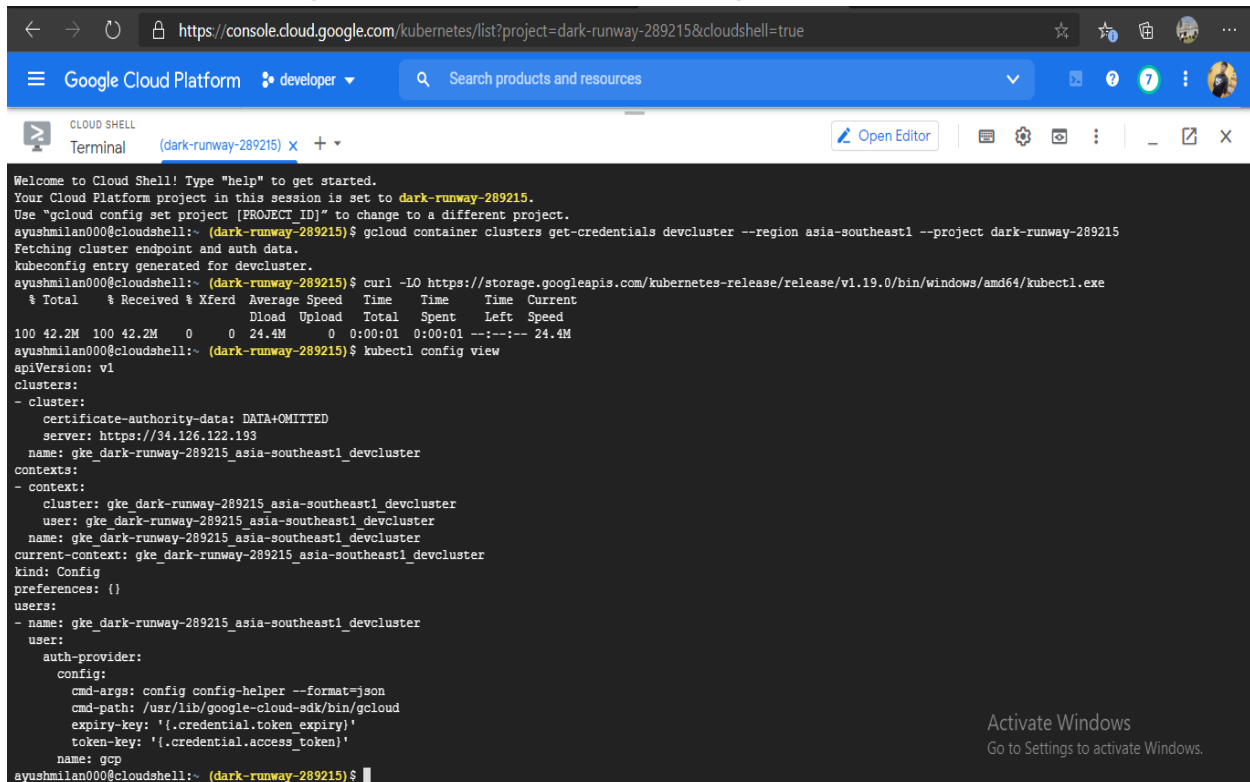
Dismiss

Activate Windows
Go to Settings to activate Windows.

Our cluster is ready.

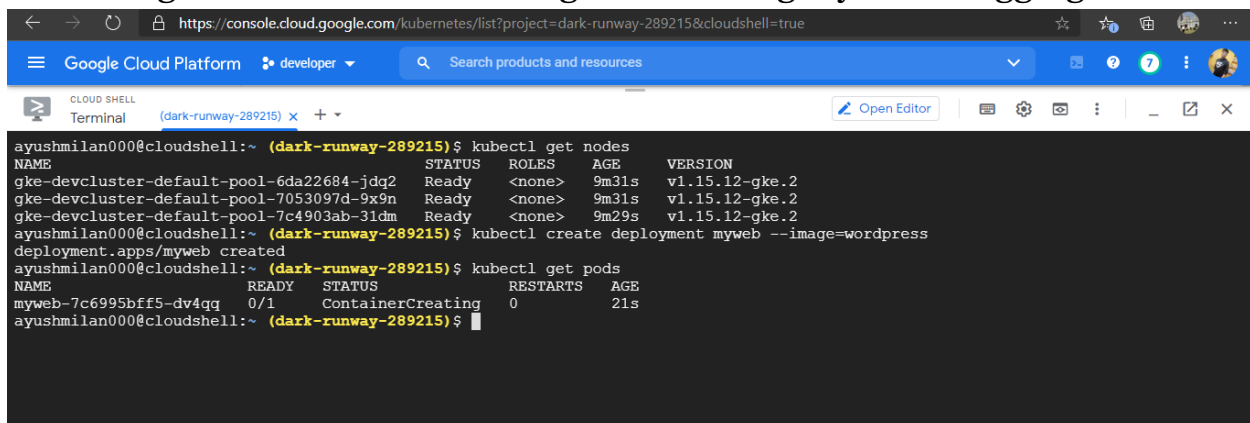
Now we have to configure kubectl with gcloud and we can do that by launching one command. Click on “Connect” button in front of newly created Kubernetes Cluster.

Use “kubectl config view” command to config in shell



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to dark-runway-289215.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
ayushmilan000@cloudshell:~ (dark-runway-289215) $ gcloud container clusters get-credentials devcluster --region asia-southeast1 --project dark-runway-289215
Fetching cluster endpoint and auth data.
kubeconfig entry generated for devcluster.
ayushmilan000@cloudshell:~ (dark-runway-289215) $ curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 42.2M 100 42.2M 0 0 24.4M 0 0:00:01 0:00:01 --:--:-- 24.4M
ayushmilan000@cloudshell:~ (dark-runway-289215) $ kubectl config view
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: DATA+OMITTED
  server: https://34.126.122.193
  name: gke_dark-runway-289215_asia-southeast1_devcluster
contexts:
- context:
  cluster: gke_dark-runway-289215_asia-southeast1_devcluster
  user: gke_dark-runway-289215_asia-southeast1_devcluster
  name: gke_dark-runway-289215_asia-southeast1_devcluster
  current-context: gke_dark-runway-289215_asia-southeast1_devcluster
kind: Config
preferences: {}
users:
- name: gke_dark-runway-289215_asia-southeast1_devcluster
  user:
    auth-provider:
      config:
        cmd-args: config config-helper --format=json
        cmd-path: /usr/lib/google-cloud-sdk/bin/gcloud
        expiry-key: '{.credential.token_expiry}'
        token-key: '{.credential.access_token}'
        name: gcp
  name: gcp
ayushmilan000@cloudshell:~ (dark-runway-289215) $
```

Use command- kubectl get nodes and then deploy our web application,here i am using WordPress docker image for creating my own blogging website.



```
ayushmilan000@cloudshell:~ (dark-runway-289215) $ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
gke-devcluster-default-pool-6da22684-jdq2    Ready    <none>    9m31s    v1.15.12-gke.2
gke-devcluster-default-pool-7053097d-9x9n    Ready    <none>    9m31s    v1.15.12-gke.2
gke-devcluster-default-pool-7c4903ab-31dm    Ready    <none>    9m29s    v1.15.12-gke.2
ayushmilan000@cloudshell:~ (dark-runway-289215) $ kubectl create deployment myweb --image=wordpress
deployment.apps/myweb created
ayushmilan000@cloudshell:~ (dark-runway-289215) $ kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
myweb-7c6995bff5-dv4qq              0/1     ContainerCreating    0            21s
ayushmilan000@cloudshell:~ (dark-runway-289215) $
```

Next we have to check the status of pods launched and get all the information about all pods:

```
← → ↺ 🔒 https://console.cloud.google.com/kubernetes/list?project=dark-runway-289215&cloudshell=true
Google Cloud Platform developer 🔍 Search products and resources
CLOUD SHELL Terminal (dark-runway-289215) x +
Open Editor

deployment.apps/myweb created
ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myweb-7c6995bff5-dv4qq 0/1     ContainerCreating 0          21s
ayushmilan000@cloudshell:~ (dark-runway-289215)$ clear
ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myweb-7c6995bff5-dv4qq 1/1     Running   0          94s
ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl scale deploy myweb --replicas=4
deployment.extensions/myweb scaled
ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myweb-7c6995bff5-7ntbj 0/1     ContainerCreating 0          7s
myweb-7c6995bff5-dv4qq 1/1     Running        0         4m34s
myweb-7c6995bff5-sckhm 0/1     ContainerCreating 0          7s
myweb-7c6995bff5-z28vz 1/1     Running        0          7s
ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE                                     NOMINATED NODE   READINESS GATES
myweb-7c6995bff5-7ntbj 1/1     Running   0          40s   10.8.1.9     gke-devcluster-default-pool-6da22684-jdq2   <none>          <none>
myweb-7c6995bff5-dv4qq 1/1     Running   0          5m7s   10.8.2.6     gke-devcluster-default-pool-7053097d-9x9n   <none>          <none>
myweb-7c6995bff5-sckhm 1/1     Running   0          40s   10.8.0.3     gke-devcluster-default-pool-7c4903ab-31dm   <none>          <none>
myweb-7c6995bff5-z28vz 1/1     Running   0          40s   10.8.2.7     gke-devcluster-default-pool-7053097d-9x9n   <none>          <none>
ayushmilan000@cloudshell:~ (dark-runway-289215)$
```

Now we have to add a Load Balancer to handle the task smoothly.
Load Balancer automatically manage the resources
Use command : `kubectl expose deployment your_deployment_name --type=LoadBalancer -- port=80`

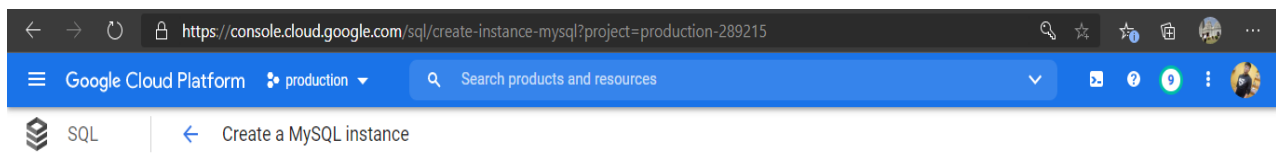
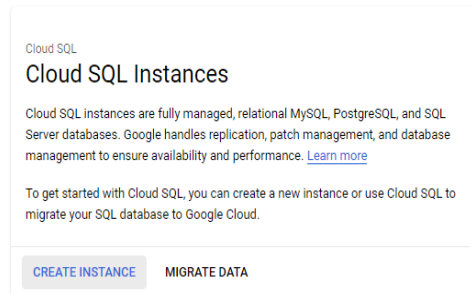
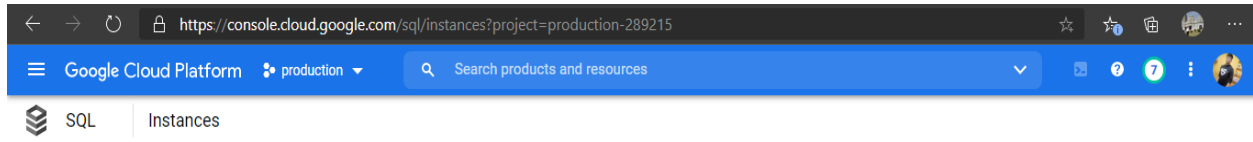
```
← → ↺ 🔒 https://console.cloud.google.com/kubernetes/list?project=dark-runway-289215&cloudshell=true
Google Cloud Platform developer 🔍 Search products and resources
CLOUD SHELL Terminal (dark-runway-289215) x +
Open Editor

ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl expose deployment myweb --type=LoadBalancer --port=80
service/myweb exposed
ayushmilan000@cloudshell:~ (dark-runway-289215)$ kubectl get services
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
kubernetes    ClusterIP     10.12.0.1        <none>       443/TCP          21m
myweb         LoadBalancer 10.12.11.102     <pending>    80:32674/TCP     17s
ayushmilan000@cloudshell:~ (dark-runway-289215)$
```

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

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

Create a mysql instance database:



Instance info

Instance ID

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

databsos

Root password

Set a password for the root user. [Learn more](#)



Generate

☐ No password

Location

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

Region

Choice is permanent

Zone

Can be changed at any time

us-east1 (South Carolina)

Any

Database version

MySQL 5.7

[Show configuration options](#)

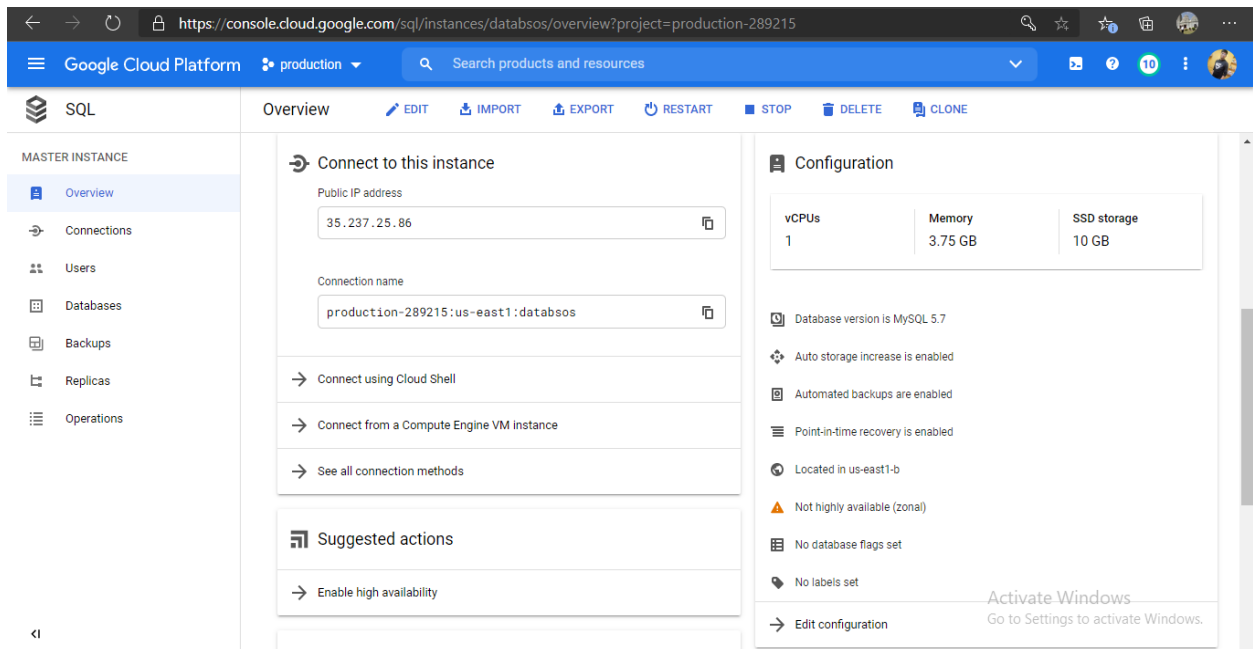
Create

Cancel

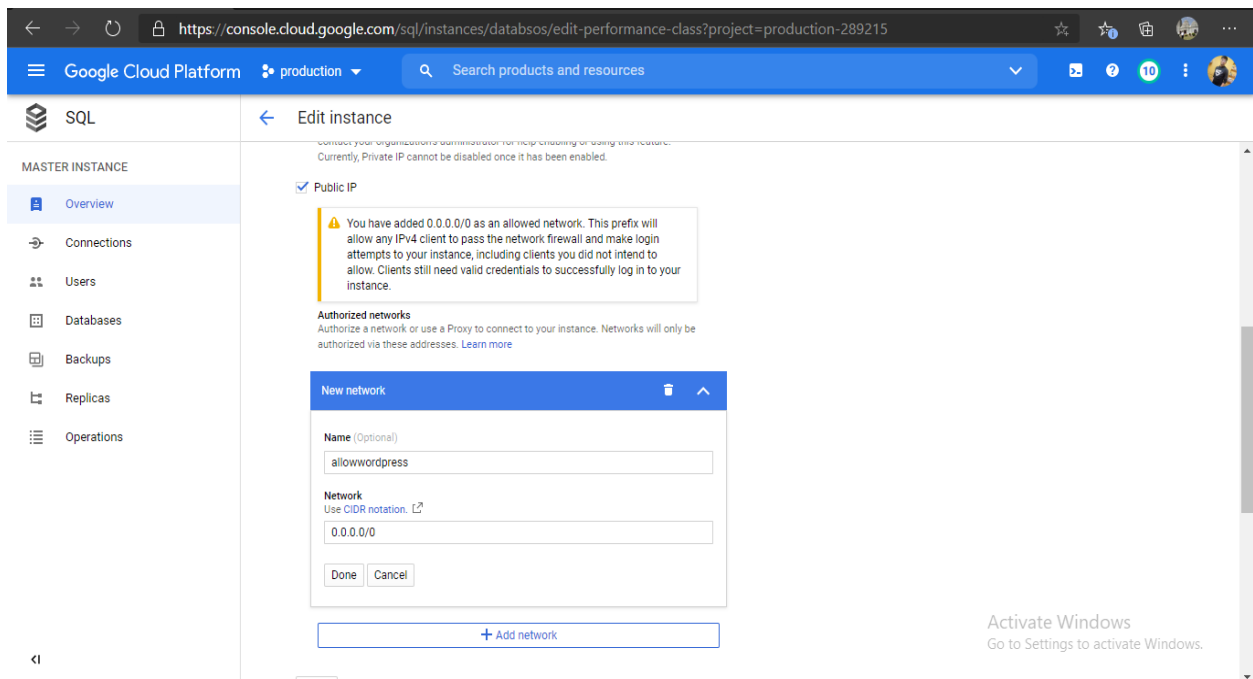
Activate Windows

Go to Settings to activate Windows.

After creating the database go to edit configuration :



enable the public ip address:



Create Database for our WordPress website :

The screenshot shows the Google Cloud Platform console for a MySQL instance named 'databsos'. The left sidebar lists navigation options: Overview, Connections, Users, Databases (selected), Backups, Replicas, and Operations. The main content area shows the instance details and a table of databases.

Name	Collation	Character set	Type
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8_general_ci	utf8	System
sys	utf8_general_ci	utf8	User
wordpressdb	utf8_general_ci	utf8	User

Now login using Cloud Shell to MySQL Instance:

The screenshot shows a Google Cloud Shell terminal window. The user has executed the command `mysql -h 35.237.25.86 -u root -p`. The terminal output shows the MySQL prompt and the result of the `show databases` command.

```
ayushmilan000@cloudshell:~ (production-289215)$ mysql -h 35.237.25.86 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 167
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 (0.22 sec)

mysql>
```

Next ,using the ip address provided by Load Balancer, access the website. Setup WordPress to MySQL server and then we can start blogging on our own website over Public Network.

← → ↻ ⚠ Not secure | 35.197.154.27/wp-admin/setup-config.php?step=1 ☆ 🌐 📁 👤 ⋮




Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpressdb"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="root"/>	Your database username.
Password	<input type="password" value="123456789"/>	Your database password.
Database Host	<input type="text" value="35.237.25.86"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Activate Windows
Go to Settings to activate Windows.


← → ↻ ⚠ Not secure | 35.197.154.27/wp-admin/setup-config.php ☆ 🌐 📁 👤 ⋮



English (United States)
Afrikaans
العربية
العربية المغربية
অসমীয়া
ગુજરાતી
Azerbaycan dili
Беларуская мова
Български
বাংলা
Български
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Österreich)
Deutsch (Schweiz, Du)
Deutsch (Schweiz)
Deutsch (Sie)
Deutsch

Activate Windows
Go to Settings to activate Windows.

← → ↻ ⚠ Not secure | 35.197.154.27/wp-admin/install.php?language=en_US 🔍 ☆ 🌐 📁 👤 ⋮



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

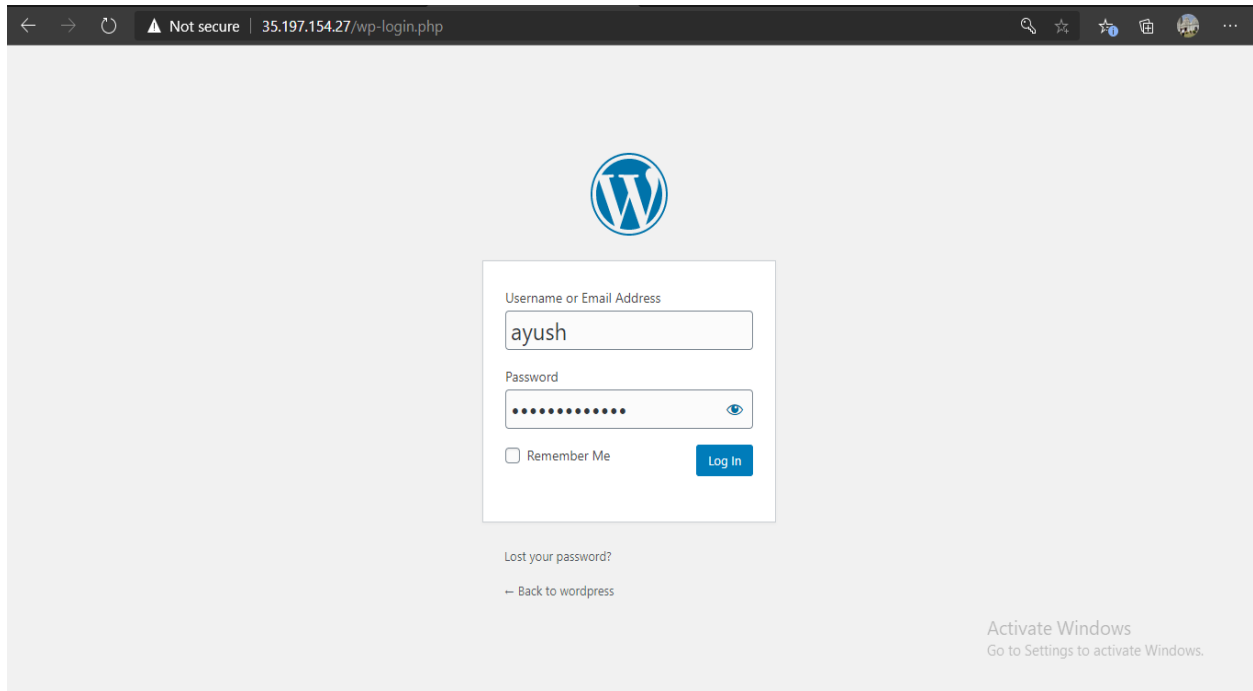
Please provide the following information. Don't worry, you can always change these settings later.

Site Title	<input type="text" value="wordpress"/>
Username	<input type="text" value="ayush"/>
Password	<div><input type="password" value=""/> Strong</div> <div><input type="button" value="Hide"/></div>

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

Activate Windows
Go to Settings to activate Windows.

Login with credentials you created while setup:



Our website is ready, we can utilize it:

