

# CLOUD PROJECT

## Deploy Web App using GCP



Compute



Storage & Database



Networking



Big Data



Developer Tools



Google Cloud Platform



Identity & Security



Internet of Things



Cloud AI



Management Tools



Data Transfer

*Google Cloud Platform and Services*

**By Raman Verma**

# TABLE OF CONTENTS

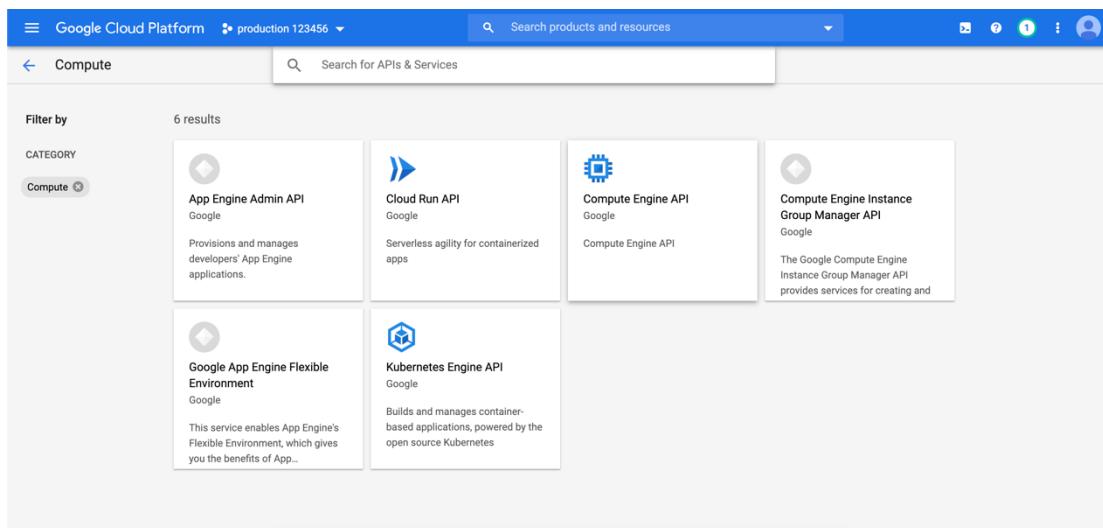
<b>GCP SERVICES .....</b>	<b>1</b>
Products.....	1
<b>PROJECT WORK .....</b>	<b>2</b>
Task -1 [Motivation] .....	2
Setup GCP .....	2
Enable APIs .....	3
Custom VPC and Create VM Instance.....	4-6
Create Firewall Rules.....	7
<b>VPC PEERING .....</b>	<b>8-9</b>
Task - 2 [Motivation] .....	10
<b>KUBERNETES CLUSTER ENGINE .....</b>	<b>11-12</b>
Install Kubectl, Configure GCP .....	13-14
Deploy WordPress on Server and Add LoadBalancer .....	14-15
Add Load Balancer.....	15
Create Cloud SQL Instance .....	15-17
Configure WordPress.....	18
Login .....	19
Publish First Post .....	19

# GCP

# Services

Google Cloud Platform, 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.

GCP provides API for each resource which can be utilized to enable in projects to access them. e.g. Compute APIs



The screenshot shows the Google Cloud Platform interface with the search bar set to 'Search for APIs & Services'. A filter is applied for the 'Compute' category, resulting in six service cards displayed in a grid:

- App Engine Admin API** (Google): Provisions and manages developers' App Engine applications.
- Cloud Run API** (Google): Serverless agility for containerized apps.
- Compute Engine API** (Google): Compute Engine API
- Compute Engine Instance Group Manager API** (Google): The Google Compute Engine Instance Group Manager API provides services for creating and...
- Google App Engine Flexible Environment** (Google): This service enables App Engine's Flexible Environment, which gives you the benefits of App...
- Kubernetes Engine API** (Google): Builds and manages container-based applications, powered by the open source Kubernetes

## PRODUCTS

**“Google lists over 90 products under the GCP brand.”**

- *Compute*
- *Storage and Databases*
- *Networking*
- *Big Data*
- *Cloud AI*
- *Management Tools*
- *Identity & Security*
- *IoT*
- *API Platform*

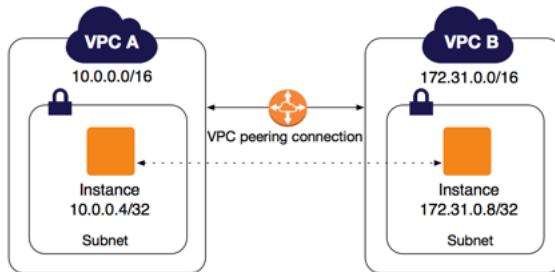
## PROJECT WORK – USABILITY OF GCP

### Task - 1

**Motivation:** Utilizing Google Private network to share data or for high speed communication between 2 VM instances launched in separate Projects and Regions.

- *Create multiple projects namely "developer" and "production".*
- *Create custom VPC network for both the projects*
  - *Production [USA] - IP: 10.0.1.0/24*
  - *Developer [Singapore] - IP: 10.0.2.0/24*
- *Create a link between both of the VPC using VPC Peering for the faster communication and transfer of data.*

**"A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses."**



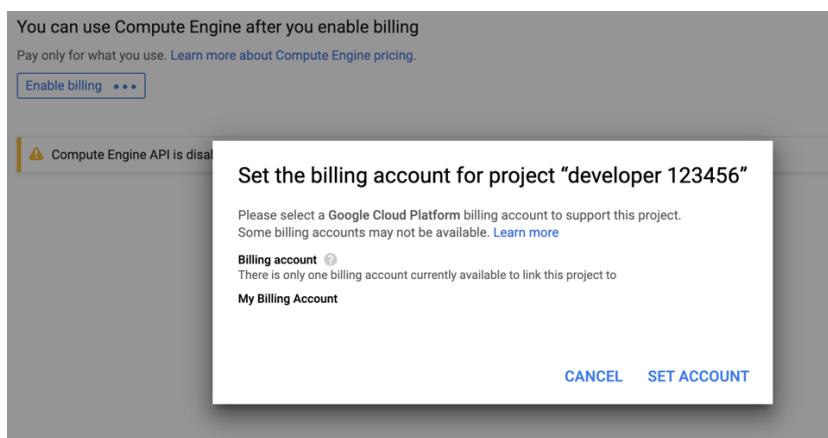
Follow steps to Setup your own private network using GCP VPC peering services.

- *Sign in with Gmail ID to google cloud. [Free \$300 to try Google Cloud availed new user]*
- *Create new 2 Projects with Project name {Unique in the region} and respective Project ID will be generated.*

- By default, all APIs is **DISABLE**, as **Billing** is not enabled for respective project. So, we have to enable it.

**"Click Enable Billing"**

This Message will pop-up, set your account by adding credit card information to GCP account.



After which you can go to Compute Engine API to Enable it for the project

- Create custom VPC Network in both of the Projects {dev, prod}, so that we can create our VM instances using these VPCs later on.

New subnet

Name \* lab1  
Region \* us-east1  
IP address range \* 10.0.1.0/24

Create secondary IP range  
Private Google access  
 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](#)  
 Off

CANCEL    DONE

LAB-1 in USA [IP range: 10.0.1.0/24]

New subnet

Name \* lab2  
Region \* asia-southeast1  
IP address range \* 10.0.2.0/24

Create secondary IP range  
Private Google access  
 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](#)  
 Off

CANCEL    DONE

LAB-2 in Singapore [IP range: 10.0.2.0/24]

- Display custom created VPC under VPC Network.

VPC network						VPC networks																																																																																																																																																																																																																																								
<ul style="list-style-type: none"> <li>VPC networks</li> <li>External IP addresses</li> <li>Firewall</li> <li>Routes</li> <li>VPC network peering</li> <li>Shared VPC</li> <li>Serverless VPC access</li> <li>Packet mirroring</li> </ul>						<ul style="list-style-type: none"> <li>VPC networks</li> <li>External IP addresses</li> <li>Firewall</li> <li>Routes</li> <li>VPC network peering</li> <li>Shared VPC</li> <li>Serverless VPC access</li> <li>Packet mirroring</li> </ul>																																																																																																																																																																																																																																								
<table border="1"> <thead> <tr> <th></th> </tr> </thead> <tbody> <tr> <td>vpcprodproject</td> <td>1</td> <td>Custom</td> <td>asie-southeast1</td> <td>default</td> <td>10.148.0.0/20</td> <td>10.148.0.1</td> <td>asia-southeast1</td> <td>default</td> <td>10.148.0.0/20</td> <td>10.148.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>us-east4</td> <td>default</td> <td>10.150.0.0/20</td> <td>10.150.0.1</td> <td></td> <td>us-east4</td> <td>default</td> <td>10.150.0.0/20</td> <td>10.150.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>australia-southeast1</td> <td>default</td> <td>10.152.0.0/20</td> <td>10.152.0.1</td> <td></td> <td>australia-southeast1</td> <td>default</td> <td>10.152.0.0/20</td> <td>10.152.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>europe-west2</td> <td>default</td> <td>10.154.0.0/20</td> <td>10.154.0.1</td> <td></td> <td>europe-west2</td> <td>default</td> <td>10.154.0.0/20</td> <td>10.154.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>eu-west3</td> <td>default</td> <td>10.156.0.0/20</td> <td>10.156.0.1</td> <td></td> <td>eu-west3</td> <td>default</td> <td>10.156.0.0/20</td> <td>10.156.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>southamerica-east1</td> <td>default</td> <td>10.158.0.0/20</td> <td>10.158.0.1</td> <td></td> <td>southamerica-east1</td> <td>default</td> <td>10.158.0.0/20</td> <td>10.158.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>asia-south1</td> <td>default</td> <td>10.160.0.0/20</td> <td>10.160.0.1</td> <td></td> <td>asia-south1</td> <td>default</td> <td>10.160.0.0/20</td> <td>10.160.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>northamerica-northeast1</td> <td>default</td> <td>10.162.0.0/20</td> <td>10.162.0.1</td> <td></td> <td>northamerica-northeast1</td> <td>default</td> <td>10.162.0.0/20</td> <td>10.162.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>europe-west4</td> <td>default</td> <td>10.164.0.0/20</td> <td>10.164.0.1</td> <td></td> <td>europe-west4</td> <td>default</td> <td>10.164.0.0/20</td> <td>10.164.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>eu-north1</td> <td>default</td> <td>10.166.0.0/20</td> <td>10.166.0.1</td> <td></td> <td>eu-north1</td> <td>default</td> <td>10.166.0.0/20</td> <td>10.166.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>us-west2</td> <td>default</td> <td>10.168.0.0/20</td> <td>10.168.0.1</td> <td></td> <td>us-west2</td> <td>default</td> <td>10.168.0.0/20</td> <td>10.168.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>asia-east2</td> <td>default</td> <td>10.170.0.0/20</td> <td>10.170.0.1</td> <td></td> <td>asia-east2</td> <td>default</td> <td>10.170.0.0/20</td> <td>10.170.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>europe-west5</td> <td>default</td> <td>10.172.0.0/20</td> <td>10.172.0.1</td> <td></td> <td>europe-west5</td> <td>default</td> <td>10.172.0.0/20</td> <td>10.172.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>asia-northeast2</td> <td>default</td> <td>10.174.0.0/20</td> <td>10.174.0.1</td> <td></td> <td>asia-northeast2</td> <td>default</td> <td>10.174.0.0/20</td> <td>10.174.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>asia-northeast3</td> <td>default</td> <td>10.178.0.0/20</td> <td>10.178.0.1</td> <td></td> <td>asia-northeast3</td> <td>default</td> <td>10.178.0.0/20</td> <td>10.178.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>us-west3</td> <td>default</td> <td>10.180.0.0/20</td> <td>10.180.0.1</td> <td></td> <td>us-west3</td> <td>default</td> <td>10.180.0.0/20</td> <td>10.180.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>us-west4</td> <td>default</td> <td>10.182.0.0/20</td> <td>10.182.0.1</td> <td></td> <td>us-west4</td> <td>default</td> <td>10.182.0.0/20</td> <td>10.182.0.1</td> </tr> <tr> <td></td> <td></td> <td></td> <td>asia-southeast2</td> <td>default</td> <td>10.184.0.0/20</td> <td>10.184.0.1</td> <td></td> <td>asia-southeast2</td> <td>default</td> <td>10.184.0.0/20</td> <td>10.184.0.1</td> </tr> </tbody> </table>																		vpcprodproject	1	Custom	asie-southeast1	default	10.148.0.0/20	10.148.0.1	asia-southeast1	default	10.148.0.0/20	10.148.0.1				us-east4	default	10.150.0.0/20	10.150.0.1		us-east4	default	10.150.0.0/20	10.150.0.1				australia-southeast1	default	10.152.0.0/20	10.152.0.1		australia-southeast1	default	10.152.0.0/20	10.152.0.1				europe-west2	default	10.154.0.0/20	10.154.0.1		europe-west2	default	10.154.0.0/20	10.154.0.1				eu-west3	default	10.156.0.0/20	10.156.0.1		eu-west3	default	10.156.0.0/20	10.156.0.1				southamerica-east1	default	10.158.0.0/20	10.158.0.1		southamerica-east1	default	10.158.0.0/20	10.158.0.1				asia-south1	default	10.160.0.0/20	10.160.0.1		asia-south1	default	10.160.0.0/20	10.160.0.1				northamerica-northeast1	default	10.162.0.0/20	10.162.0.1		northamerica-northeast1	default	10.162.0.0/20	10.162.0.1				europe-west4	default	10.164.0.0/20	10.164.0.1		europe-west4	default	10.164.0.0/20	10.164.0.1				eu-north1	default	10.166.0.0/20	10.166.0.1		eu-north1	default	10.166.0.0/20	10.166.0.1				us-west2	default	10.168.0.0/20	10.168.0.1		us-west2	default	10.168.0.0/20	10.168.0.1				asia-east2	default	10.170.0.0/20	10.170.0.1		asia-east2	default	10.170.0.0/20	10.170.0.1				europe-west5	default	10.172.0.0/20	10.172.0.1		europe-west5	default	10.172.0.0/20	10.172.0.1				asia-northeast2	default	10.174.0.0/20	10.174.0.1		asia-northeast2	default	10.174.0.0/20	10.174.0.1				asia-northeast3	default	10.178.0.0/20	10.178.0.1		asia-northeast3	default	10.178.0.0/20	10.178.0.1				us-west3	default	10.180.0.0/20	10.180.0.1		us-west3	default	10.180.0.0/20	10.180.0.1				us-west4	default	10.182.0.0/20	10.182.0.1		us-west4	default	10.182.0.0/20	10.182.0.1				asia-southeast2	default	10.184.0.0/20	10.184.0.1		asia-southeast2	default	10.184.0.0/20	10.184.0.1						
vpcprodproject	1	Custom	asie-southeast1	default	10.148.0.0/20	10.148.0.1	asia-southeast1	default	10.148.0.0/20	10.148.0.1																																																																																																																																																																																																																																				
			us-east4	default	10.150.0.0/20	10.150.0.1		us-east4	default	10.150.0.0/20	10.150.0.1																																																																																																																																																																																																																																			
			australia-southeast1	default	10.152.0.0/20	10.152.0.1		australia-southeast1	default	10.152.0.0/20	10.152.0.1																																																																																																																																																																																																																																			
			europe-west2	default	10.154.0.0/20	10.154.0.1		europe-west2	default	10.154.0.0/20	10.154.0.1																																																																																																																																																																																																																																			
			eu-west3	default	10.156.0.0/20	10.156.0.1		eu-west3	default	10.156.0.0/20	10.156.0.1																																																																																																																																																																																																																																			
			southamerica-east1	default	10.158.0.0/20	10.158.0.1		southamerica-east1	default	10.158.0.0/20	10.158.0.1																																																																																																																																																																																																																																			
			asia-south1	default	10.160.0.0/20	10.160.0.1		asia-south1	default	10.160.0.0/20	10.160.0.1																																																																																																																																																																																																																																			
			northamerica-northeast1	default	10.162.0.0/20	10.162.0.1		northamerica-northeast1	default	10.162.0.0/20	10.162.0.1																																																																																																																																																																																																																																			
			europe-west4	default	10.164.0.0/20	10.164.0.1		europe-west4	default	10.164.0.0/20	10.164.0.1																																																																																																																																																																																																																																			
			eu-north1	default	10.166.0.0/20	10.166.0.1		eu-north1	default	10.166.0.0/20	10.166.0.1																																																																																																																																																																																																																																			
			us-west2	default	10.168.0.0/20	10.168.0.1		us-west2	default	10.168.0.0/20	10.168.0.1																																																																																																																																																																																																																																			
			asia-east2	default	10.170.0.0/20	10.170.0.1		asia-east2	default	10.170.0.0/20	10.170.0.1																																																																																																																																																																																																																																			
			europe-west5	default	10.172.0.0/20	10.172.0.1		europe-west5	default	10.172.0.0/20	10.172.0.1																																																																																																																																																																																																																																			
			asia-northeast2	default	10.174.0.0/20	10.174.0.1		asia-northeast2	default	10.174.0.0/20	10.174.0.1																																																																																																																																																																																																																																			
			asia-northeast3	default	10.178.0.0/20	10.178.0.1		asia-northeast3	default	10.178.0.0/20	10.178.0.1																																																																																																																																																																																																																																			
			us-west3	default	10.180.0.0/20	10.180.0.1		us-west3	default	10.180.0.0/20	10.180.0.1																																																																																																																																																																																																																																			
			us-west4	default	10.182.0.0/20	10.182.0.1		us-west4	default	10.182.0.0/20	10.182.0.1																																																																																																																																																																																																																																			
			asia-southeast2	default	10.184.0.0/20	10.184.0.1		asia-southeast2	default	10.184.0.0/20	10.184.0.1																																																																																																																																																																																																																																			
<table border="1"> <thead> <tr> <th></th> </tr> </thead> <tbody> <tr> <td>vpcdepproject</td> <td>1</td> <td>Custom</td> <td>asie-southeast1</td> <td>lab1</td> <td>10.0.1.0/24</td> <td>10.0.1.1</td> <td></td> <td>asie-southeast1</td> <td>lab2</td> <td>10.0.2.0/24</td> <td>10.0.2.1</td> </tr> </tbody> </table>																		vpcdepproject	1	Custom	asie-southeast1	lab1	10.0.1.0/24	10.0.1.1		asie-southeast1	lab2	10.0.2.0/24	10.0.2.1																																																																																																																																																																																																																	
vpcdepproject	1	Custom	asie-southeast1	lab1	10.0.1.0/24	10.0.1.1		asie-southeast1	lab2	10.0.2.0/24	10.0.2.1																																																																																																																																																																																																																																			

- Let's start with interesting part, creating a VM instance attached with our custom VPCs in both of the projects.
- Go to Compute Engine -> VM Instances.
  - Provide Name
  - Region and Zone
  - Machine Configuration: "N1" is very basic type
  - Boot image: CentOS7 [Stable]
  - HDD: 20 GB

To create a VM instance, select one of the options:

- New VM instance** Create a single VM instance from scratch
- New VM instance from template** Create a single VM instance from an existing template
- New VM instance from machine image** Create a single VM instance from an existing machine image
- Marketplace** Deploy a ready-to-go solution onto a VM instance

**Name** osprod  
**Labels** (Optional)  
**Region** us-east1 (South Carolina)    **Zone** us-east1-b  
**Machine configuration**  
**Machine family** General-purpose    Memory-optimized    Compute-optimized  
**Series** N1  
**Machine type** n1-standard-1 (1 vCPU, 3.75 GB memory)

You have ₹21,406.50 free trial credits remaining  
\$25.07 monthly estimate  
That's about \$0.034 hourly  
Pay for what you use: No upfront costs and per second billing

- Select Boot Image as CentOS {default: Debian GNU} and choose custom VPC under Networking -> Networking Interfaces.

**Boot disk**  
Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for?  
Public images   Custom images   Snapshots   Existing disks  
Operating system: CentOS  
Version: CentOS 7  
x86\_64 built on 20200811, supports Shielded VM features  
Boot disk type: Standard persistent disk   Size (GB): 20

**Networking**  
Management   Security   Disks   **Networking**   Sole Tenancy  
Network tags (Optional)  
Hostname: osprod.us-east1-b.c.production-123456-287919.internal  
Network interfaces: vpcprodproject lab1 (10.0.1.0/24)  
+ Add network interface

Management Security Disks **Networking** Sole Tenancy

**Network tags** (Optional)

**Hostname** ?  
Set a custom hostname for this instance or leave it default. Choice is permanent  
osprod.us-east1-b.c.production-123456-287919.internal

**Network interfaces** ?  
Network interface is permanent

**Network interface**

**Network** ?  
vpcprodproject

**Subnetwork** ?  
lab1 (10.0.1.0/24)

**Primary internal IP** ?  
Ephemeral (Automatic)

**Show alias IP ranges**

**External IP** ?  
Ephemeral

”Create a VM”

- One important thing to notice, all the things you are doing in GCP, it will be recorded and same notification will be provided to you.

Recent Activity	
	Create VM instance "osdev" and its boot disk "osdev"
	Just now
	developer 123456
	Create VM instance "osprod" and its boot disk "osprod"
	1 minute ago
	production 123456
	Create firewall rule "webhosting-dev"
	10 minutes ago
	developer 123456
	Create firewall rule "webhosting-prod"
	11 minutes ago
	production 123456
	Delete firewall rule "webhosting-prod"
	13 minutes ago
	production 123456
	Delete firewall rule "webhosting-dev"
	13 minutes ago
	developer 123456
	Create network "vpcdevproject"
	19 minutes ago
	developer 123456
	Create network "vpcprodproject"
	20 minutes ago
	production 123456
	Create firewall rule "webhosting-dev"
	27 minutes ago
	developer 123456

- Now if you go newly created VM and try to connect, it will not allow you to do it. Because, just one more thing need to be done. As currently, we have not assigned any firewall rules to VPC.

**VPC network details**

**Firewall rules**

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Logs	Hit count	Last hit
No matching results									

**Create a firewall rule**

**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 (example: 0.0.0.0/0, 192.168.2.0/24)

**Second source filter**: None

**Protocols and ports**:  tcp : 80

"Add firewall rule for both of the VPCs"

i) You can allow all traffic over the network to come to your Instance using "Ingress" and making source IP range (from anywhere):  
0.0.0.0/0

ii) You can allow specific ports

or Allow all ports (recommended).

**Protocols and ports**

Allow all  
 Specified protocols and ports

- *Everything is ready, check connecting with the VM instance. Click SSH*

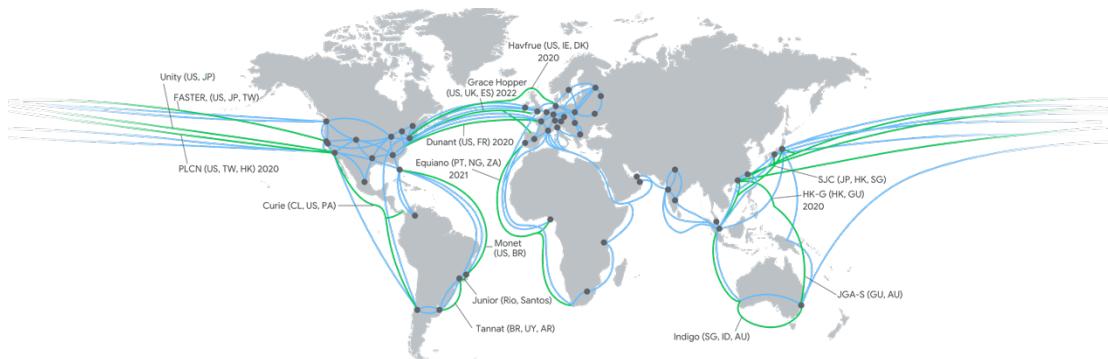
The screenshot shows the Google Cloud Platform Compute Engine interface. Under the 'VM instances' section, there is a table with one row for 'osprod'. The 'External IP' column shows '104.196.51.144'. To the right of the table, there is an 'SSH' button with a dropdown menu.

*Ping to External IPs of the Instances, using cloud shell/terminal/CMD*

The image displays two terminal sessions. The left terminal is on a Mac (rs3vgk@osprod) and the right is on a Linux host (rss3vgk@osdev). Both terminals are running a ping command to the external IP 34.126.96.14. The output shows multiple ICMP packets being sent with TTL=54 and round-trip times consistently around 221 ms.

```
ssh.cloud.google.com/projects/production-123456-287919/zones/us-...
Connected, host fingerprint: ssh-rsa 0 42:17:22:61:39:DD:Ba:3D:84:D1
[...]
Last login: Sat Aug 29 20:39:33 2020 from 35.235.240.81
[rs3vgk@osprod ~]$ ping 34.126.96.14
PING 34.126.96.14 (34.126.96.14) 56(84) bytes of data, 64 bytes from 34.126.96.14:
64 bytes from 34.126.96.14: icmp_seq=1 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=2 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=3 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=4 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=5 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=6 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=7 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=8 ttl=54 time=222 ms
64 bytes from 34.126.96.14: icmp_seq=9 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=10 ttl=54 time=221 ms
[...]
rss3vgk@osdev:~$ ssh.cloud.google.com/projects/developer-123456-287919/zones/as...
Connected, host fingerprint: ssh-rsa 0 93:CA:02:C2:D9:07:42:F3:...
Last login: Sat Aug 29 20:42:39 2020 from 35.235.240.17
[rss3vgk@osdev ~]$ ping 34.126.96.14
PING 34.126.96.14 (34.126.96.14) 56(84) bytes of data, 64 bytes from 34.126.96.14:
64 bytes from 34.126.96.14: icmp_seq=1 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=2 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=3 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=4 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=5 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=6 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=7 ttl=54 time=222 ms
64 bytes from 34.126.96.14: icmp_seq=8 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=9 ttl=54 time=221 ms
64 bytes from 34.126.96.14: icmp_seq=10 ttl=54 time=221 ms
[...]
```

- *Now, Let get your hand dirt by creating VPC peering to use the google private network (super-fast).*



*it very difficult <:) I am Joking...*

## CREATE VPC PEERING

- *Create VPC peering in both projects, required both projects-ID and VPCs name.*
- *We have to provide information like, to which VPC peering we want, within same project or outside project. Sample below:*

- Now Let's ping the Instance from one another using Internal IPs.

### Ping: Development OS → Production OS

```

ssh.cloud.google.com/projects/production-123456-287919/zones/us-east1-b/instances/osprod?useAdminProxy=true&authuser=
Connected, host: fingerprin...
Last login: Sat Aug 29 20:44:32 2020 from 35.235.240.80
[...]
7 packets transmitted, 6 received, 14% packet loss, time 600ms
rtt min/avg/max/mdev = 221.857/222.069/222.979/0.560 ms
[rss3vgk@osprod ~]$ 
    
```

### Ping: Production OS → Development OS

```

ssh.cloud.google.com/projects/developer-123456-287919/zones/asia-southeast1-b/instances/osdev?useAdminProxy=true&a...
Connected, host: fingerprin...
Last login: Sat Aug 29 20:44:50 2020 from 35.235.240.16
[...]
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 223.031/223.380/224.371/0.508 ms
[rss3vgk@osdev ~]$ 
    
```

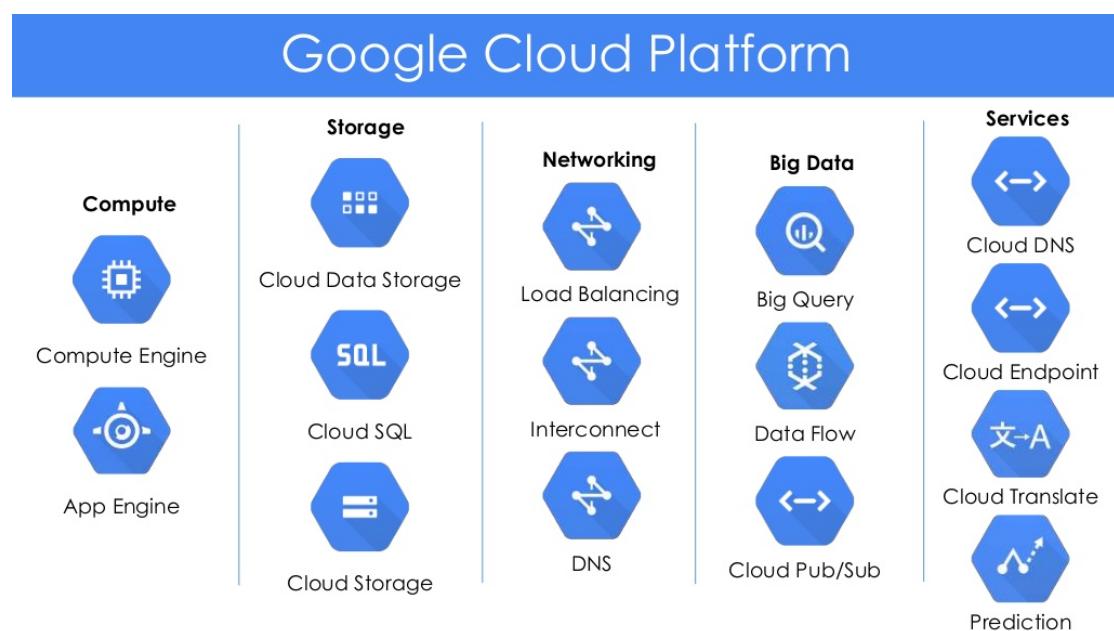
Pinging happening, WOW, Utilize GCP Internal Network.

## Task - 2

**Motivation:** Utilizing Google Kubernetes Service to launch auto manageable Web Application, exposed, easily scalable, traffic handling, secure database (GCP SQL service).

- *Create Kubernetes Cluster in “developer” project and launch Web Application with the Load Balancer.*
- *Create Cloud SQL Instance (MySQL) for creating secure database for Web Application.*
- *Connect the SQL database to the web application.*

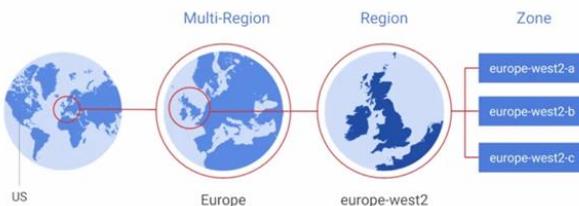
**“Cloud SQL is a fully managed service that makes it easy to set up, manage, and administer relational databases: PostgreSQL, MySQL, and SQL Server.”**



Follow steps to Setup your own private network using GCP VPC peering services.

## GCP Regions and Zones

Google Cloud Platform is organized into regions and zones



**"Regional resources can be used by any resource in that region, regardless of zone, while zonal resources can only be used by other resources in the same zone."**

- *Create Cluster using GCP Kubernetes Cluster Engine:*

Google Cloud Platform developer 123456 Search products and resources

Kubernetes Engine Clusters

Clusters

Workloads

Services & Ingress

Applications

Configuration

Storage

Object Browser

Migrate to containers

Kubernetes Engine  
Kubernetes clusters

Containers package an application so 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](#)

Create cluster Deploy container Take the quickstart

- *Cluster Basics*

- *Name, Location type*
- *Region: Cluster (Singapore), SQL (USA)*

← Create a Kubernetes cluster ADD NODE POOL REMOVE NODE POOL

Cluster basics

The new cluster will be created with the name, version, and in the location you specify here. After the cluster is created, name and location can't be changed.

To experiment with an affordable cluster, try [My first cluster](#) in the Cluster set-up guides

Name: wordpress-clusture

Location type: Regional

Region: asia-southeast1

Specify default node locations: asia-southeast1-a, asia-southeast1-b, asia-southeast1-c

- *Node Pools -> Default Pool*

- *Size – Number of nodes (per zone): 1*

The screenshot shows the 'Create a Kubernetes cluster' interface. On the left, there's a sidebar with sections like 'Cluster basics', 'POOLS' (with 'default-pool' selected), 'CLUSTER' (with 'Automation', 'Networking', 'Security', 'Metadata', and 'Features'), and 'TER'. The main right panel is titled 'Node pool details' and contains fields for 'Name' (set to 'default-pool'), 'Node version' (set to '1.15.12-gke.2 (master version)'), and 'Size' (set to 'Number of nodes (per zone) \* 1'). Below these, it says 'Total (in all zones): 3'.

- *Node Pools -> Default Pool -> Nodes*

- *Image-type: Container-Optimized OS*
- *Machine Family: General Purpose [n1-standard-1]*

The screenshot shows the 'Create a Kubernetes cluster' interface. The left sidebar shows 'Cluster basics', 'NODE POOLS' (with 'default-pool' selected), and 'CLUSTER' (with 'Automation', 'Networking', 'Security', 'Metadata', and 'Features'). The right panel is titled 'Nodes' and contains settings for creating new nodes. It includes 'Image type' (set to 'Container-Optimized OS (cos) (default)'), 'Machine Configuration' (set to 'GENERAL-PURPOSE'), 'Machine family' (set to 'GENERAL-PURPOSE'), 'Series' (set to 'N1'), and 'Machine type' (set to 'n1-standard-1 (1 vCPU, 3.75 GB memory)'). Below this, there's a summary table showing 'vCPU' as 1 and 'Memory' as 3.75 GB.

-> *Create Cluster, Cluster is Ready*

The screenshot shows the 'Kubernetes Engine' section of the Google Cloud Platform console. The left sidebar has options for 'Clusters', 'Workloads', 'Services & Ingress', 'Applications', 'Configuration', and 'Storage'. The main area is titled 'Clusters' and shows a table with one row for 'wordpress-cluster'. The table columns include 'Name' (checkbox, checked for 'wordpress-cluster'), 'Location' (set to 'asia-southeast1'), 'Cluster size' (set to 3), 'Total cores' (set to 3 vCPUs), 'Total memory' (set to 11.25 GB), 'Notifications' (checkbox), and 'Labels' (checkbox). There are also 'Connect', 'Edit', and 'Delete' buttons at the bottom of the table row.

- It is good to use cloud shell/own PC to manager Kubernetes cluster for that we need to install Kubectl and configure GCloud with it.

The screenshot shows a web browser displaying the Kubernetes documentation at [kubernetes.io/docs/tasks/tools/install-kubectl/](https://kubernetes.io/docs/tasks/tools/install-kubectl/). The page title is "Install kubectl on macOS". The left sidebar contains navigation links like Home, Getting started, Concepts, Tasks, Install Tools, and others. The main content area provides instructions for installing the Kubectl binary using curl on macOS. It includes a command example:

```
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/darwin/amd64/kubectl"
```

Below the command, there's a note about replacing the curl command with a specific version if needed, and an example for downloading version v1.19.0:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.19.0/b...
```

### *Installation Command:*

```
~ $curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/darwin/amd64/kubectl"
```

- Now, configure kubectl with gcloud and we can do that by launching one command. Click on "Connect" button in front of newly created Kubernetes Cluster.
- Copy the command and run it either in cloud shell or own PC -> terminal.

The screenshot shows the "Connect to the cluster" section of the documentation. It provides instructions for connecting via command-line or dashboard. Under "Command-line access", it shows the command:

```
$ gcloud container clusters get-credentials wordpress-cluster --region asia-southeast1 --project developer-12345
```

A "Run in Cloud Shell" button is available. Below that, under "Cloud Console dashboard", it says you can view workloads in the Cloud Console Workloads dashboard, with a "Open Workloads dashboard" button.

The screenshot shows a terminal window with the following command and output:

```
[~ $gcloud container clusters get-credentials wordpress-cluster --region asia-sout]
heast1 --project developer-12345-287919
Fetching cluster endpoint and auth data.
kubeconfig entry generated for wordpress-cluster.
~ $
```

- We can see, if kubectl is configured with GCloud by using command: "kubectl config view"

```
[~ $ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://34.87.164.200
    name: gke_developer-123456-287919_asia-southeast1_wordpress-cluster
- cluster:
    certificate-authority: /Users/ramanverma/.minikube/ca.crt
    server: https://192.168.99.100:8443
    name: minikube
contexts:
- context:
    cluster: gke_developer-123456-287919_asia-southeast1_wordpress-cluster
    user: gke_developer-123456-287919_asia-southeast1_wordpress-cluster
    name: gke_developer-123456-287919_asia-southeast1_wordpress-cluster
- context:
    cluster: minikube
    user: minikube
    name: minikube
current-context: gke_developer-123456-287919_asia-southeast1_wordpress-cluster
kind: Config
preferences: {}
users:
- name: gke_developer-123456-287919_asia-southeast1_wordpress-cluster
  user:
    auth-provider:
```

- Get status of all Nodes launched in different zones, as we have chosen launch 1 node for each zone, so we get total 3 nodes.

```
[~ $ kubectl get nodes
NAME                               STATUS   ROLES      AGE   VERSION
gke-wordpress-cluster-default-pool-004a823f-9d32   Ready    <none>    15m   v1.15.12-gke.2
gke-wordpress-cluster-default-pool-9ebbbc42-k65f   Ready    <none>    15m   v1.15.12-gke.2
gke-wordpress-cluster-default-pool-a5323dbc-hwb6   Ready    <none>    15m   v1.15.12-gke.2
~ $ ]
```

- Now let's deploy our web application, for instance I am using WordPress docker image for creating my own blogging website.

```
[~ $ kubectl create deployment myweb --image=wordpress
deployment.apps/myweb created
~ $ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
myweb-7c6995bff5-mtqmc  0/1     ContainerCreating   0          11s
~ $ ]
```

- Get the Status of PODs launched, we can scale up/down according to the increase/decrease in traffic.

```
[~ $ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
myweb-7c6995bff5-mtqmc  1/1     Running     0          118s
[~ $ kubectl scale deploy myweb --replicas=4
deployment.extensions/myweb scaled
[~ $ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
myweb-7c6995bff5-d96wd  0/1     ContainerCreating   0          3s
myweb-7c6995bff5-mtqmc  1/1     Running     0          2m4s
myweb-7c6995bff5-r946m  0/1     ContainerCreating   0          3s
myweb-7c6995bff5-wpbzx  0/1     ContainerCreating   0          3s
~ $ ]
```

- Get information about all PODs and in which node they launched.

```
~ $kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE
NESS GATES
myweb-7c6995bff5-d96wd  1/1    Running   0          28s    10.52.1.3   gke-wordpress-cluster-default-pool-a5323dbc-hwb6
>
myweb-7c6995bff5-mtqmc  1/1    Running   0          2m29s   10.52.2.5   gke-wordpress-cluster-default-pool-004a823f-9d32
>
myweb-7c6995bff5-r946m  1/1    Running   0          28s    10.52.2.6   gke-wordpress-cluster-default-pool-004a823f-9d32
>
myweb-7c6995bff5-wpbzx  1/1    Running   0          28s    10.52.0.10  gke-wordpress-cluster-default-pool-9ebbbc42-k65f
>
```

- But as **traffic increase** on your website, how do we know to balance the load? We require to add a **Load Balancer** to our Deployment which will handle this task smoothly.

Command to use: `kubectl expose deployment your_deployment_name --type=LoadBalancer --port=80`

```
[~ $kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.0.0.1      <none>        443/TCP      25m
myweb      LoadBalancer  10.0.9.44    <pending>      80:30051/TCP  8s
~ $
```

External IP provided which can be provided to user, so whenever user come to our website, Load Balancer automatically manage the resources.

## CREATE CLOUD SQL INSTANCE FOR DATABASE

- Select Production project, and Choose Storage -> SQL to create instance.

The screenshot shows two pages related to Cloud SQL:

- Cloud SQL Instances Page:** This page lists existing Cloud SQL instances. It includes a brief description of Cloud SQL instances as fully managed relational MySQL, PostgreSQL, and SQL Server databases. It also provides links to "CREATE INSTANCE" and "MIGRATE DATA".
- Create Instance Page:** This page allows users to "Choose your database engine". It offers three options: MySQL, PostgreSQL, and SQL Server. Each option has a "Choose [engine]" button.

- Create MySQL instance by providing information:

- Instance ID name
- Root password (logging to instance)
- Region and Zone
- Database version name

The screenshot shows the 'Create a MySQL instance' form in the Google Cloud Platform SQL interface. The 'Instance info' section includes fields for 'Instance ID' (set to 'databaseos'), 'Root password' (a masked password), and 'Location' (Region: 'us-east1 (South Carolina)', Zone: 'Any'). The 'Database version' is set to 'MySQL 5.7'. At the bottom are 'Create' and 'Cancel' buttons.

- Instance created for Database, Go to Edit Configuration.

The screenshot shows the 'Edit Configuration' page for the 'databaseos' instance. It displays connection details (Public IP address: 104.196.120.84, Connection name: production-123456-287919:us-east1:databaseos) and configuration settings (vCPUs: 1, Memory: 3.75 GB, SSD storage: 10 GB). A sidebar lists 'MASTER INSTANCE' options like Overview, Connections, Users, Databases, Backups, Replicas, and Operations.

- Enable the access for all Public IP addresses.

The screenshot shows the Google Cloud Platform SQL configuration options for a specific instance. Under the 'Connectivity' section, the 'Public IP' checkbox is checked. A warning message states: "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 you did not intend to allow. Clients still need valid credentials to successfully log in to your instance." Below this, there is a section for 'Authorized networks' where a new network named 'allowwordpress' is being added. The 'Network' field contains '0.0.0.0/0'. Buttons for 'Done' and 'Cancel' are visible at the bottom of the dialog.

- Create Database for our WordPress website either using login via console or using GCP Web UI.

The screenshot shows the MySQL database list for a master instance. The 'Databases' tab is selected. It lists four databases: 'information\_schema', 'mysql', 'performance\_schema', and 'sys'. The 'CREATE DATABASE' button is visible at the top right of the list.

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

The screenshot shows the 'Create a database' dialog box. The 'Database Name \*' field contains 'wordpressdb'. The 'Character set \*' dropdown is set to 'utf8'. Both fields have a note below them: 'Must follow the MySQL identifier rules.' and 'Can be changed later by executing an ALTER DATABASE query.' At the bottom are 'CREATE' and 'CANCEL' buttons.

## Login using Cloud Shell to MySQL Instance

```
rss3vgk@cloudshell:~ (production-123456-287919)$ mysql -h 104.196.120.84 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to production-123456-287919.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
rss3vgk@cloudshell:~ (production-123456-287919)$ mysql -h 104.196.120.84 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 112
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 (0.23 sec)

mysql> █
```

- *Using the IP address provided by Load Balancer, access the website. Just setup WordPress to MySQL server and we can start blogging on our own website over Public Network.*

34.126.115.191/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	wordpressdb	The name of the database you want to use with WordPress.
Username	root	Your database username.
Password	databaseos1	Your database password.
Database Host	104.196.120.84	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

**Submit**

- Login with credentials you created.



- Website is ready, you can utilize it, publish your first Post 😊

UNCATEGORIZED

# Hello world!

By raman | August 29, 2020 | 1 Comment

I have created own website with the HELP of GCP resources.  
Kubernetes Cluster and MySQL services provided by GCP is very useful.

[Edit](#)