

## 1. Create a GKE Cluster

- **Cluster Name:** Choose a unique name for your cluster.
- **Region and Zone:** Choose a suitable region and zone, for example, us-central1-a.
- **Number of Nodes:** The assignment specifies at least two nodes for the cluster. For example, you can create a 2-node cluster.
- **Machine Type:** Choose a machine type based on the workload, such as e2-medium.
- **Network Configuration:** Ensure that the VPC network is configured to allow external access to the web server. This typically involves setting up firewall rules to allow HTTP (port 80) and HTTPS (port 443) traffic.

The screenshot shows the Google Cloud Platform console with an SSH-in-browser session open. The terminal displays the command to create a GKE cluster and its output.

```
kumishadrack75@shadrackasianvm:~$ gcloud container clusters create testcluster1 \
--zone us-central1-a \
--num-nodes 2 \
--machine-type e2-medium \
--enable-autoscaling --min-nodes 2 --max-nodes 3
Note: The Kubelet readonly port (10255) is now deprecated. Please update your workloads to use the recommended al
ternatives. See https://cloud.google.com/kubernetes-engine/docs/how-to/disable-kubelet-readonly-port for ways to
check usage and for migration instructions.
Note: Your Pod address range ('--cluster-ipv4-cidr') can accommodate at most 1008 node(s).
Creating cluster testcluster1 in us-central1-a... Cluster is being health-checked (Kubernetes Control Plane is h
ealthy)...done.
Created [https://container.googleapis.com/v1/projects/ace-mission-435516-m0/zones/us-central1-a/clusters/testclus
ter1].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload/_gcloud/us-c
entral1-a/testcluster1?project=ace-mission-435516-m0
kubeconfig entry generated for testcluster1.
```

NAME	LOCATION	MASTER_VERSION	MASTER_IP	MACHINE_TYPE	NODE_VERSION	NUM_NODES	STA
testcluster1	us-central1-a	1.30.6-gke.1125000	34.123.122.214	e2-medium	1.30.6-gke.1125000	2	RUN

Below the table, the terminal shows the prompt `kumishadrack75@shadrackasianvm:~$`.

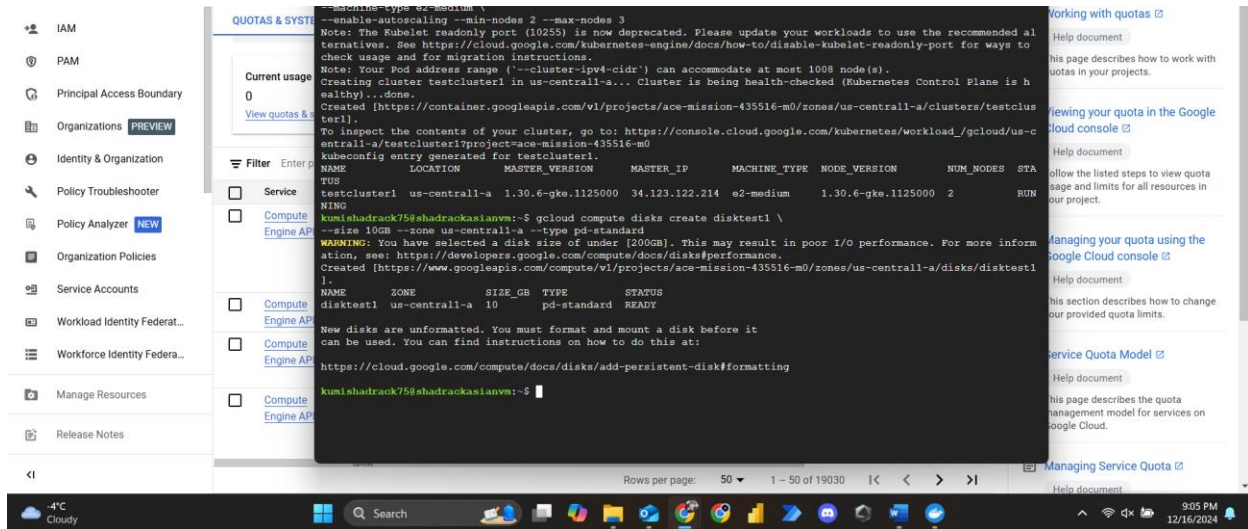
On the left sidebar, the 'Quotas & Systems' section is expanded, showing 'Current usage' as 0. Below this, there is a list of services with checkboxes, including 'Service', 'Compute Engine API', and 'Compute Engine AP'.

On the right sidebar, there are several links and help documents related to quotas, such as 'Working with quotas', 'Viewing your quota in the Google Cloud console', and 'Managing your quota using the Google Cloud console'.

## 2. Create a Persistent Disk

- **Persistent Disk Size:** Choose an appropriate size for your web server's content storage. For example, 10GB may be sufficient for a basic web server.
- **Storage Class:** Use the default pd-standard storage class unless you have specific needs.

- **Mount Path:** The Persistent Disk should be mounted inside the container at a path like /mnt/data.

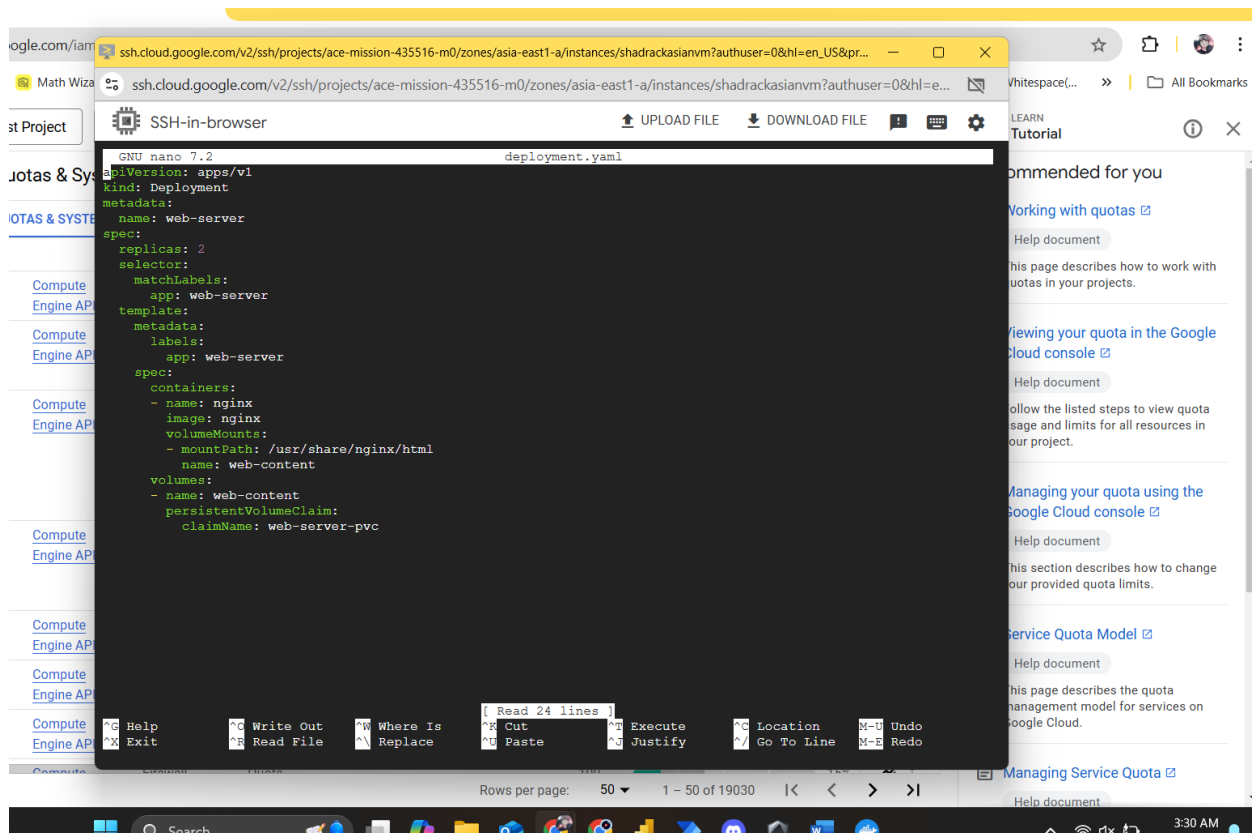


**Command to mount Persistent Disk to Kubernetes:** In your Kubernetes deployment manifest, you will need to create a PersistentVolume (PV) and a PersistentVolumeClaim (PVC) to bind the Persistent Disk to the container.

### 3. Deploy the Web Server Container

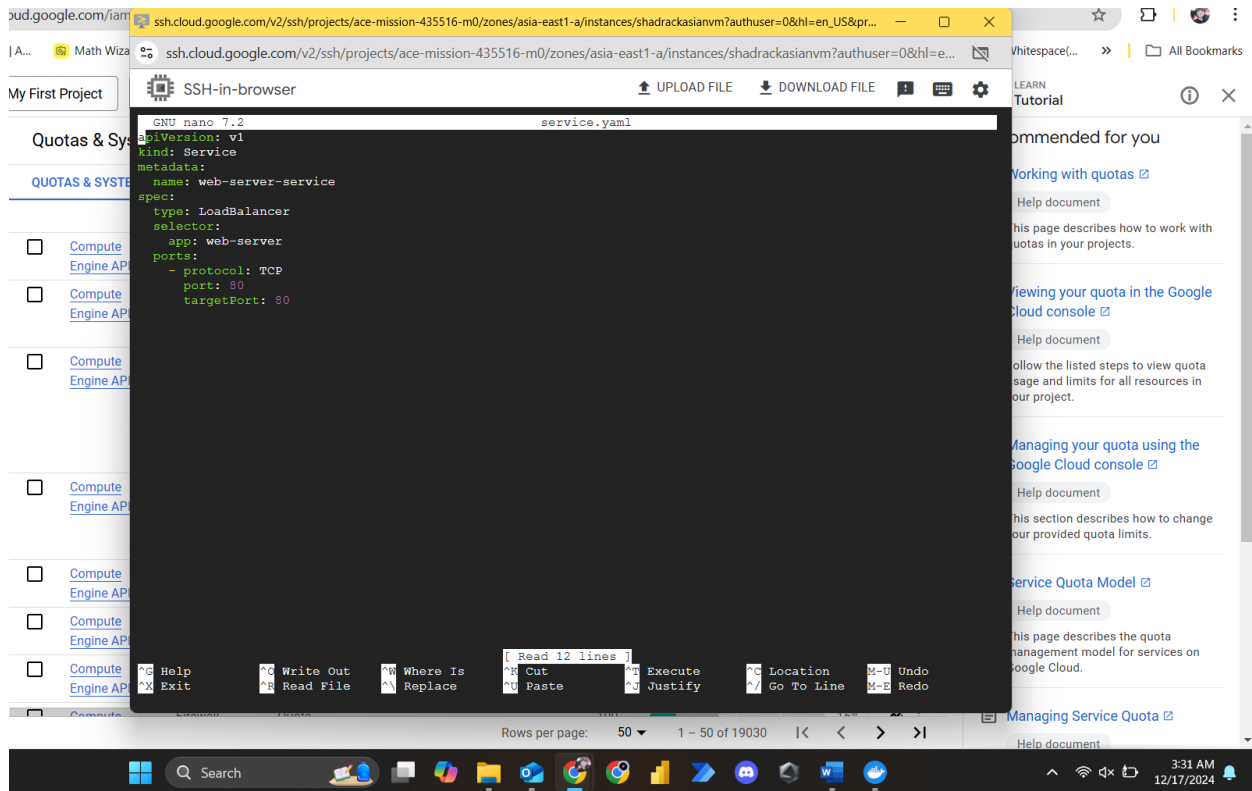
- **Container Image:** Use a pre-built web server image, such as nginx or httpd (Apache HTTP Server). You can find these images on Docker Hub or Google Container Registry.
- **Deployment Manifest:** Create a Kubernetes Deployment manifest (deployment.yaml) to specify the desired number of replicas for your web server.

Example deployment.yaml for an Nginx web server:



- **Service Manifest:** Create a Kubernetes Service of type LoadBalancer to expose the web server externally.

Example service.yaml:



## 4. Deploy the Application

Once the Kubernetes manifests (deployment.yaml and service.yaml) are ready, you can apply them to your GKE cluster using kubectl.

**Commands to deploy:**

Quotas & Sys
QUOTAS & SYSTEM

☐
[Compute Engine AP](#)

☐
[Compute Engine AP](#)

☐
[Compute Engine AP](#)

☐
[Compute Engine AP](#)

☐
[Compute Engine AP](#)

☐
[Compute Engine AP](#)

```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 17 01:21:55 2024 from 35.235.244.81
kumishadrack75@shadrackasianvm:~$ nano deployment.yaml
kumishadrack75@shadrackasianvm:~$ nano service.yaml
kumishadrack75@shadrackasianvm:~$ nano deployment.yaml
kumishadrack75@shadrackasianvm:~$ nano service.yaml
kumishadrack75@shadrackasianvm:~$ kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
deployment.apps/web-server created
service/web-server-service created
kumishadrack75@shadrackasianvm:~$

```

## 5. Configure the Web Server

- Custom Content:** You can configure your web server by mounting specific content (HTML files, configurations) onto the Persistent Disk, or you can modify the container's entrypoint to serve specific content.
- SSL/TLS Configuration (Optional):** To secure the communication, you can configure SSL/TLS by using Let's Encrypt or manually adding certificates.

## 6. Test the Web Server

- Use kubectl get svc to find the external IP of the web server's load balancer.
- Open the external IP in a web browser or use curl to verify that the web server is accessible and serving the correct content.

### Command to get external IP:

kubectl get svc web-server-service

The screenshot shows a web browser window with an SSH session to a Google Cloud VM. The terminal output is as follows:

```
Linux shadrackasianvm 6.1.0-28-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 17 01:21:55 2024 from 35.235.244.81
kumishadrack75@shadrackasianvm:~$ nano deployment.yaml
kumishadrack75@shadrackasianvm:~$ nano service.yaml
kumishadrack75@shadrackasianvm:~$ nano deployment.yaml
kumishadrack75@shadrackasianvm:~$ nano service.yaml
kumishadrack75@shadrackasianvm:~$ kubectl apply -f deployment.yaml
deployment.apps/web-server created
service/web-server-service created
kumishadrack75@shadrackasianvm:~$ kubectl get svc web-server-service
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
web-server-service  LoadBalancer      34.118.229.95    34.42.185.231    80:30911/TCP     92s
kumishadrack75@shadrackasianvm:~$
```

The left sidebar of the Google Cloud console shows the 'Quotas & Systems' menu. The right sidebar shows a 'Recommended for you' section with links to 'Working with quotas', 'Viewing your quota in the Google Cloud console', and 'Managing your quota using the Google Cloud console'.

#### 4. Add the Persistent Volume and Claim

You also need to create a PersistentVolumeClaim (PVC) to bind the Persistent Disk to the pods. Add the following content in a separate file named pvc.yaml:



My First Project

SSH-in-browser

UPLOAD FILE

DOWNLOAD FILE

LEARN Tutorial

Quotas & Systems

QUOTAS & SYSTEMS

Compute Engine API

Compute Engine API

Compute Engine API

Compute Engine API

Compute Engine API

Compute Engine API

Compute Engine API

Compute Engine API

individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Tue Dec 17 01:21:55 2024 from 35.235.244.81

kumishadrack75@shadrackasianvm:~\$ nano deployment.yaml

kumishadrack75@shadrackasianvm:~\$ nano service.yaml

kumishadrack75@shadrackasianvm:~\$ nano deployment.yaml

kumishadrack75@shadrackasianvm:~\$ nano service.yaml

kumishadrack75@shadrackasianvm:~\$ kubectl apply -f deployment.yaml

kumishadrack75@shadrackasianvm:~\$ kubectl apply -f service.yaml

deployment.apps/web-server created

service/web-server-service created

kumishadrack75@shadrackasianvm:~\$ kubectl get svc web-server-service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

web-server-service LoadBalancer 34.118.229.95 34.42.185.231 80:30911/TCP 92s

kumishadrack75@shadrackasianvm:~\$ nano deployment.yaml

kumishadrack75@shadrackasianvm:~\$ nano pvc.yaml

kumishadrack75@shadrackasianvm:~\$ kubectl apply -f pvc.yaml

persistentvolumeclaim/web-server-pvc created

kumishadrack75@shadrackasianvm:~\$ kubectl apply -f deployment.yaml

deployment.apps/web-server unchanged

service/web-server-service unchanged

kumishadrack75@shadrackasianvm:~\$ kubectl get all

NAME READY STATUS RESTARTS AGE

pod/web-server-c4c579777-knl8x 0/1 ContainerCreating 0 7m35s

pod/web-server-c4c579777-spj7m 1/1 Running 0 7m35s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 34.118.224.1 <none> 443/TCP 6h43m

service/web-server-service LoadBalancer 34.118.229.95 34.42.185.231 80:30911/TCP 7m33s

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/web-server 1/2 2 1 7m36s

NAME DESIRED CURRENT READY AGE

replicaset.apps/web-server-c4c579777 2 2 1 7m36s

kumishadrack75@shadrackasianvm:~\$

Rows per page: 50

1 - 50 of 19030

1<

<

>

>1

Managing Service Quota

## 6. Test the Deployment

- Get the external IP of the LoadBalancer service:

`kubectl get svc web-server-service`

Open the external IP in a browser or use curl to test:

`curl http://<external-ip>`



Quotas & Sy

QUOTAS & SYSTE

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

☐ Compute Engine AP

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
web-server-service	LoadBalancer	34.118.229.95	34.42.185.231	80:30911/TCP	8m24s

```
kumishadrack75@shadrackasianvm:~$ curl http://34.42.185.231
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.27.3</center>
</body>
</html>
kumishadrack75@shadrackasianvm:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
web-server-c4c579777-knl8x   0/1     ContainerCreating   0          12m
web-server-c4c579777-spj7m   1/1     Running            0          12m
kumishadrack75@shadrackasianvm:~$ nano pvc.yaml
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- ls /usr/share/nginx/html
lost+found
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-knl8x -- ls /usr/share/nginx/html
error: Internal error occurred: unable to upgrade connection: container not found ("nginx")
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- sh -c "echo '<h1>Hello from Nginx!</h1>' > /usr/share/nginx/html/index.html"
-bash: !: event not found
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- ls /usr/share/nginx/html
lost+found
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- sh -c "echo '<h1>Hello from Nginx!</h1>' > /usr/share/nginx/html/index.html"
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- sh -c 'echo "<h1>Hello from Nginx!</h1>" > /usr/share/nginx/html/index.html'
kumishadrack75@shadrackasianvm:~$ set +H
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- sh -c "echo '<h1>Hello from Nginx!</h1>' > /usr/share/nginx/html/index.html"
set -H
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-c4c579777-spj7m -- ls /usr/share/nginx/html
index.html lost+found
kumishadrack75@shadrackasianvm:~$ curl http://<EXTERNAL-IP>
-bash: syntax error near unexpected token `newline'
kumishadrack75@shadrackasianvm:~$ curl http://34.42.185.231
<h1>Hello from Nginx!</h1>
kumishadrack75@shadrackasianvm:~$
```

34.42.185.231

34.42.185.231

Google Lens

YouTube Browse Training | A... Math Wizard Cars Sales Weather forecasting... Salary Prediction Us... How to Convert Tex... OverAPI.com | Colle... Linux Class Java - isWhitespac... All Bookmarks

Hello from Nginx!

6°C Mostly cloudy

Search

3:50 AM 12/17/2024

