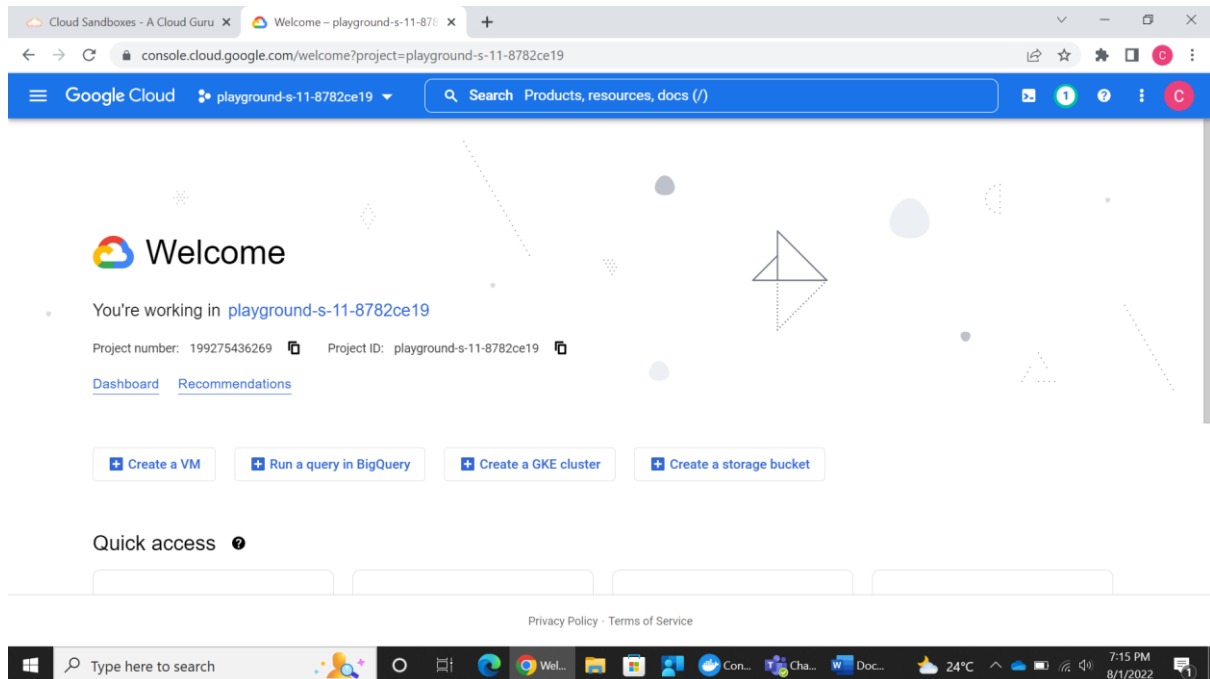


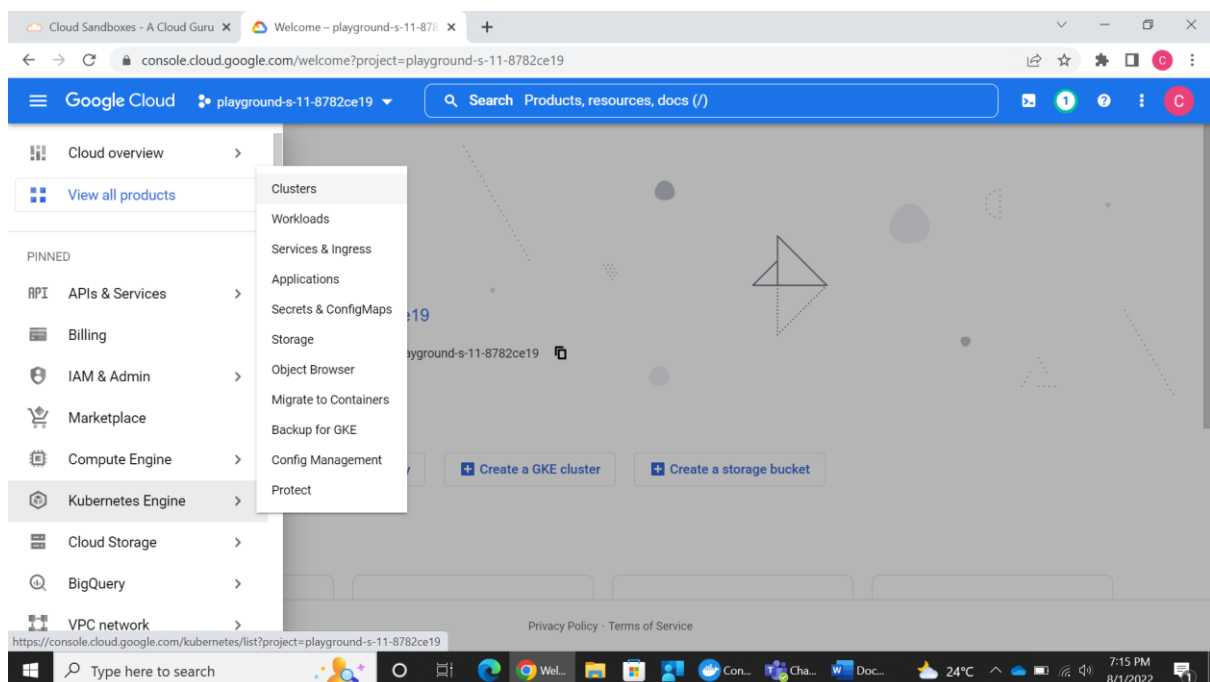
Kubernetes (K8S) - 3 Tier Architecture Implementation

In this project, created 3 tier architecture using Kubernetes. I uploaded all the files (including screenshots file Kubernetes_project.docx) to the github (<https://github.com/sudheerkumar19/Kubernetes-Project.git>). I cloned that file, and I opened that folder and executed all the commands.

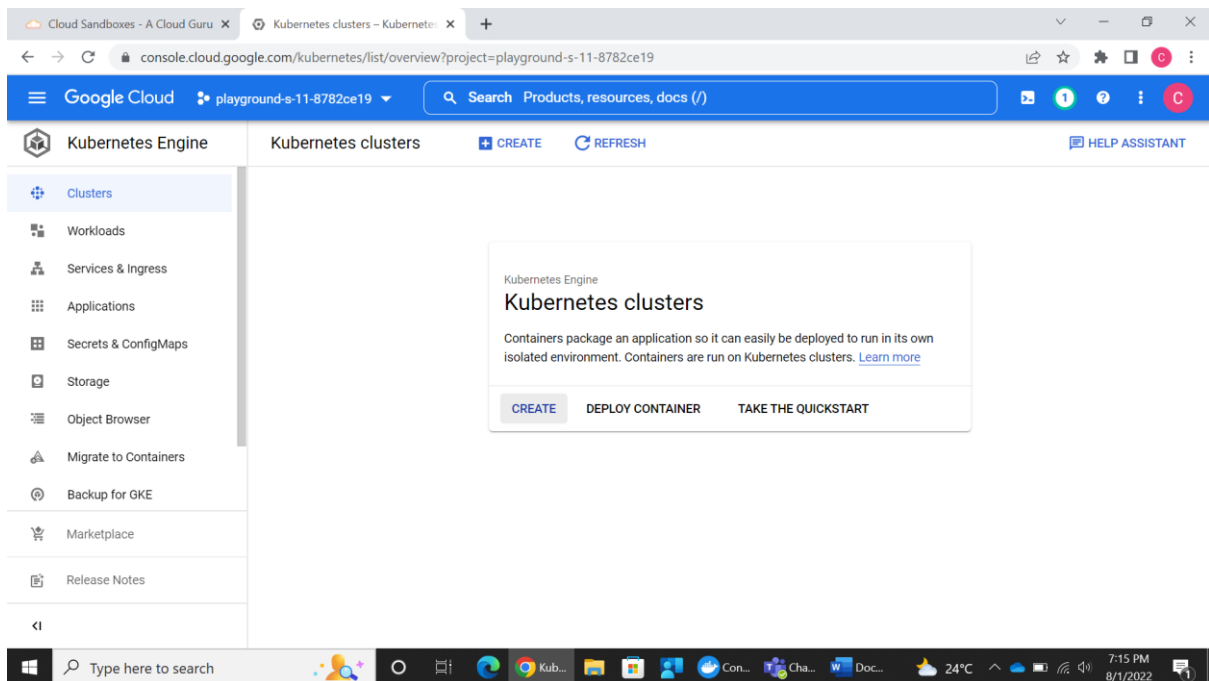
1. I logged in GCP through acloud guru playground.



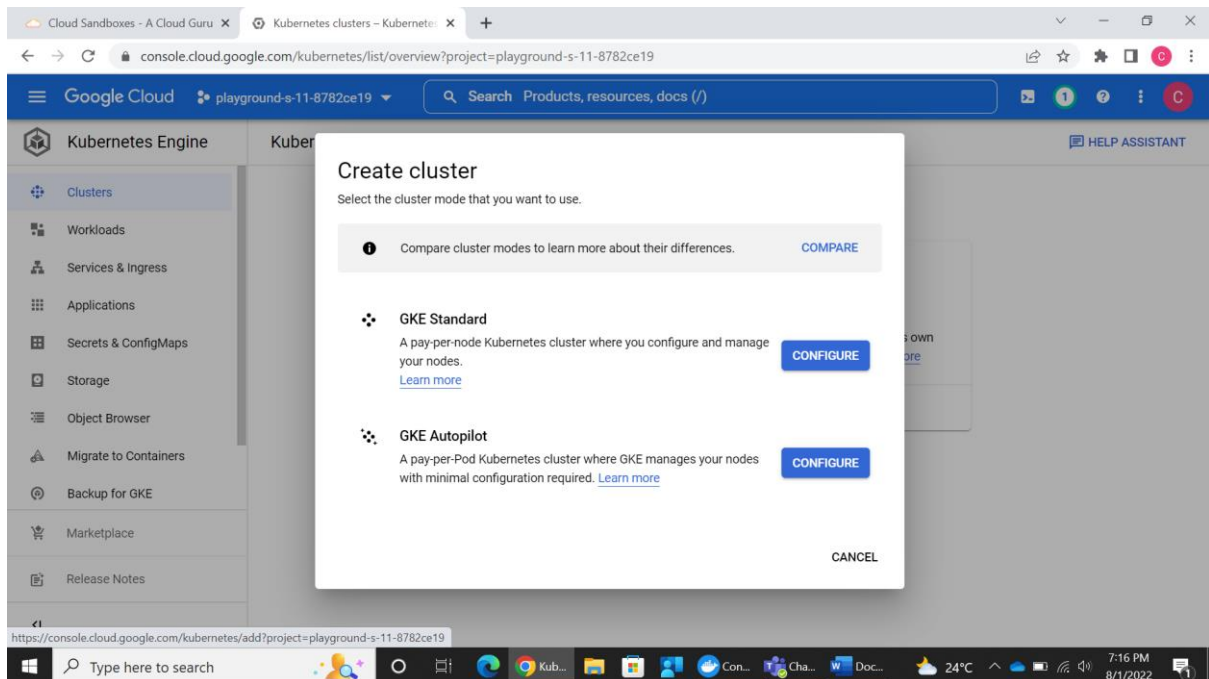
2. Choose Services --> Kubernetes Engine --> Clusters



3. Click on CREATE, GCP creates a controller and 3 nodes.




4. Click on GKE Standard → COFIGURE



Create cluster

Select the cluster mode that you want to use.

 Compare cluster modes to learn more about their differences.

[COMPARE](#)

GKE Standard

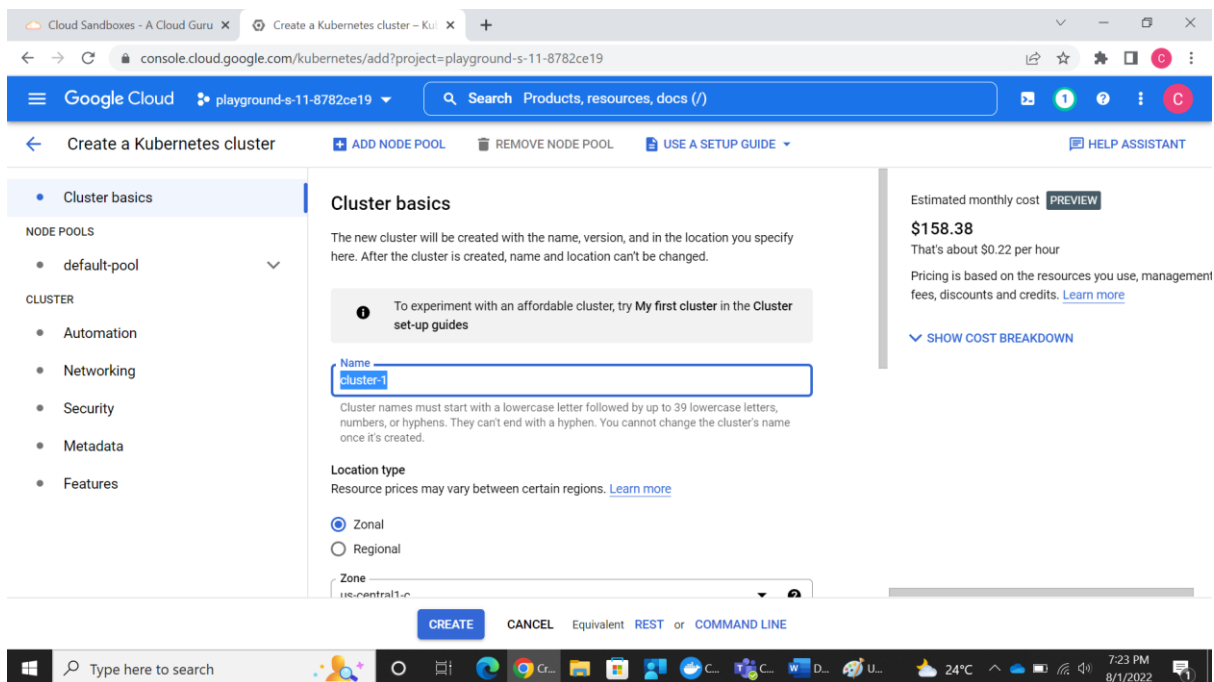
A pay-per-node Kubernetes cluster where you configure and manage your nodes.

[Learn more](#)

[CONFIGURE](#)



5. Specify cluster name and click on CREATE.



The screenshot shows the Google Cloud console interface for creating a Kubernetes cluster. The browser address bar shows the URL: `console.cloud.google.com/kubernetes/add?project=playground-s-11-8782ce19`. The page title is "Create a Kubernetes cluster". The left sidebar shows the "Cluster basics" tab selected. The main content area is titled "Cluster basics" and contains the following information:

- 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.
- Name**: The name field is set to "cluster-1". A tooltip indicates that cluster names must start with a lowercase letter followed by up to 39 lowercase letters, numbers, or hyphens. They can't end with a hyphen. You cannot change the cluster's name once it's created.
- Location type**: The location type is set to "Zonal". A tooltip indicates that resource prices may vary between certain regions. A link to "Learn more" is provided.
- Zone**: The zone is set to "us-central1-c".
- Estimated monthly cost**: The estimated monthly cost is \$158.38. A tooltip indicates that this is about \$0.22 per hour. A link to "SHOW COST BREAKDOWN" is provided.

The "CREATE" button is highlighted in blue. Other buttons include "CANCEL", "Equivalent", "REST", and "COMMAND LINE".

NODE POOLS

default-pool

Automation

Networking

Security

Metadata

Features

Control plane version

Choose whether you'd like to upgrade the cluster's control plane version manually or let GKE do it automatically. [Learn more](#)

Static version

Manually manage the version upgrades. GKE will only upgrade the control plane and nodes if it's necessary to maintain security and compatibility, as described in the release schedule. [Learn more](#)

Release channel

Let GKE automatically manage the cluster's control plane version. [Learn more](#)

Release channel

Regular channel (default)

Version

1.22.10-gke.600 (default)

These versions have passed internal validation and are considered production-quality, but don't have enough historical data to guarantee their stability. Known issues generally have known workarounds. [Release notes](#)

\$158.38

That's about \$0.22 per hour

Pricing is based on the resources you use, management fees, discounts and credits. [Learn more](#)

SHOW COST BREAKDOWN

CREATE

CANCEL

Equivalent

REST

or

COMMAND LINE

6. Processing to create Cluster.

The screenshot shows the Google Cloud console interface for Kubernetes Engine. The left sidebar contains navigation links: Clusters, Workloads, Services & Ingress, Applications, Secrets & ConfigMaps, Storage, Object Browser, and Migrate to Containers. The main content area is titled 'Kubernetes clusters' and includes buttons for CREATE, DEPLOY, and REFRESH, along with links to OPERATIONS and HELP ASSISTANT. Below these are tabs for OVERVIEW, OBSERVABILITY, and PREVIEW. The OVERVIEW tab is active, displaying a table of clusters. The table has columns for Status, Name, Location, Number of nodes, Total vCPUs, Total memory, Notifications, and Labels. A single cluster, 'mycluster1', is listed in the 'us-central1-c' region with 3 nodes and 0 vCPUs.

Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input type="checkbox"/>	mycluster1	us-central1-c	3	0	0 GB		

7. Cluster has created.

The screenshot shows the Google Cloud console interface for the Kubernetes Engine. The top navigation bar includes the Google Cloud logo, the project name 'playground-s-11-8782ce19', and a search bar. The left sidebar contains a menu with options: Clusters (selected), Workloads, Services & Ingress, Applications, Secrets & ConfigMaps, Storage, Object Browser, and Migrate to Containers. The main content area is titled 'Kubernetes Engine' and shows '1 Kubernetes cluster selected'. Below this, there are tabs for 'OVERVIEW' (selected), 'OBSERVABILITY', and 'COST OPTIMIZATION'. A 'Filter' input field is present. A table lists the clusters:

<input checked="" type="checkbox"/>	Status	Name ↑	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input checked="" type="checkbox"/>		mycluster1	us-central1-c	3	6	12 GB	—	⋮

OVERVIEW

OBSERVABILITY

PREVIEW

COST OPTIMIZATION

Filter

Enter property name or value

?

|||

<input checked="" type="checkbox"/>	Status	Name ↑	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input checked="" type="checkbox"/>		mycluster1	us-central1-c	3	6	12 GB		—

8. Connecting through cloudshell and cloning the github project files into our cluster.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Shell platform project in this session is set to playground-s-11-479dd284.
cloud_user_p_d55c6f68@cloudshell:~ (playground-s-11-479dd284) $ gcloud container clusters get-credentials mycluster1 --zone us-central1-a --project playground-s-11-479dd284
Fetching cluster endpoint and auth data.
kubeconfig entry generated for mycluster1.
cloud_user_p_d55c6f68@cloudshell:~ (playground-s-11-479dd284) $ git clone https://github.com/sunildevops77/kube_project_durga.git
```

9. Go to the specific project folder.

```
cloud_user_p_d55c6f68@cloudshell:~ (playground-s-11-479dd284) $ cd kube_project_durga
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

10. Executing all .yaml files. These files are related to

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f voting-app-pod.yaml
pod/voting-app-pod created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
voting-app-pod 1/1     Running   0           16s
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f redis-pod.yaml
pod/redis-pod created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f worker-app-pod.yaml
pod/worker-app-pod created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f postgres-pod.yaml
pod/postgres-pod created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f result-app-pod.yaml
pod/result-app-pod created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f voting-app-service.yaml
service/voting-service created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f redis-service.yaml
service/redis-service created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f postgres-service.yaml
service/db-service created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl create -f result-app-service.yaml
service/result-service created
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

11. To know all the information related to cluster, we use the following command.

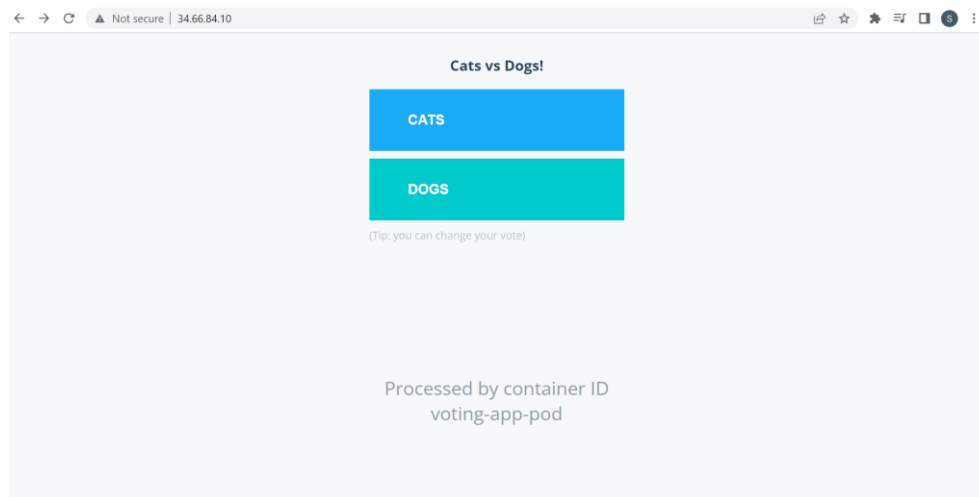
```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/postgres-pod	0/1	CrashLoopBackOff	5 (39s ago)	3m43s
pod/redis-pod	1/1	Running	0	4m52s
pod/result-app-pod	1/1	Running	0	3m8s
pod/voting-app-pod	1/1	Running	0	6m6s
pod/worker-app-pod	0/1	CrashLoopBackOff	5 (59s ago)	4m16s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/db-service	ClusterIP	10.100.2.20	<none>	5432/TCP	69s
service/kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	12m
service/redis-service	ClusterIP	10.100.14.105	<none>	6379/TCP	106s
service/result-service	LoadBalancer	10.100.2.124	34.134.11.222	80:30103/TCP	34s
service/voting-service	LoadBalancer	10.100.13.196	34.66.84.10	80:31065/TCP	2m34s

```
cloud_user_p_d55c6f68@cloudshell:~/kube_project_durga (playground-s-11-479dd284) $
```

12. By using external IP, we can execute in the browser. Then it displays the output on the screen.



13. After choosing our option, it looks like below screenshot.

