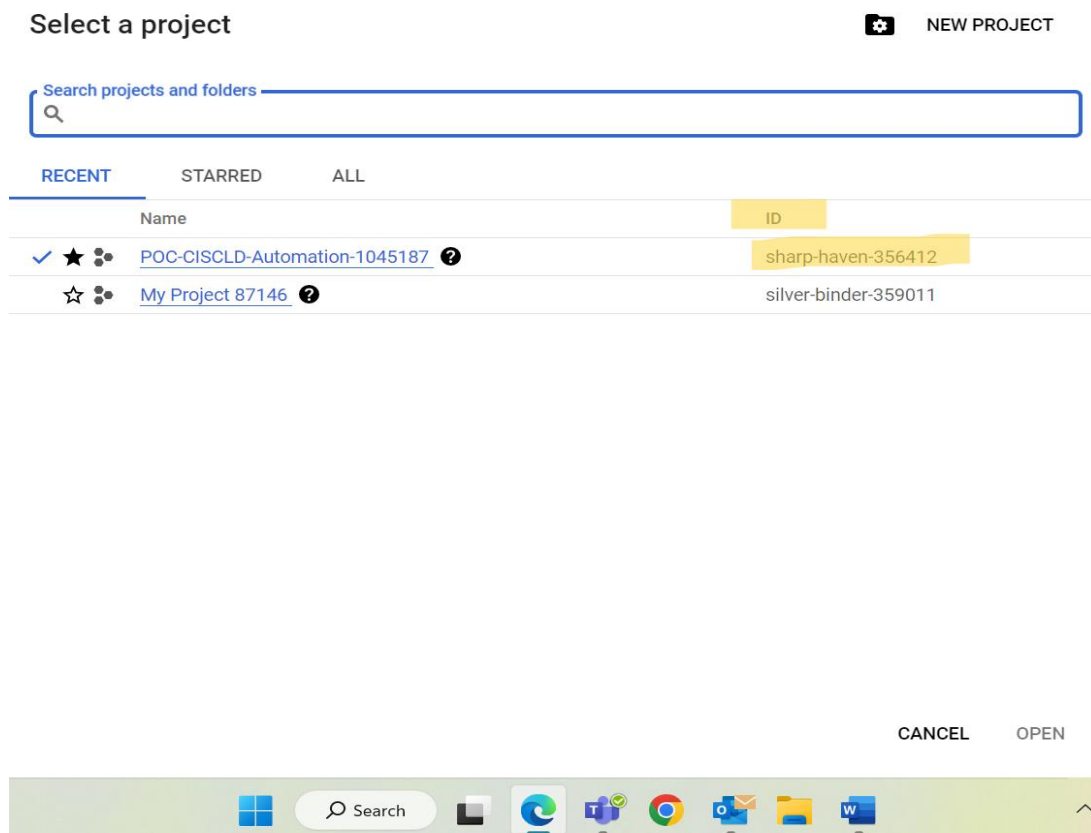


Google Kubernetes Engine Integrated with Google Container Registry (GKE with GCR)

1. Firstly, push an existing/ built container image into the Google Container Registry (GCR) by using the following ***git*** command:
 - Eg: git clone <https://github.com/GoogleCloudPlatform/kubernetes-engine-samples>
2. Set the project ID in which we need to deploy the GKE with GCR as a variable using the command
 - export PROJECT_ID=<PROJECT ID>



3. To verify that the right project-id has been assigned to the variable use:

- `echo ${PROJECT_ID}`

4. If we intend to build/use the image from the github repo use the following command with the name of the image and add a tag to it.

- `docker build -t <hostname>/${PROJECT_ID}/<imagename>:tag .`

Eg:

- `docker build -t gcr.io/${PROJECT_ID}/hello-app:v1 .`

5. To have a look at the images built/available use the command:

- `docker images`

6. If the Google Container Registry API & Docker Authentication aren't yet enabled for your GCP account, using the following commands would enable it.

- `gcloud services enable containerregistry.googleapis.com`
- `gcloud auth configure-docker`

7. Push the docker image into the host location using the command:

- `docker push gcr.io/${PROJECT_ID}/hello-app:v2`

8. Connect to the cluster using the connect function in GCP console or the following command in Cloud Shell.

- **Eg:**

```
gcloud container clusters get-credentials cac-testgke-22 --region  
us-central1 --project sharp-haven-356412
```

9. Run the below command to check the pods status is in running state.

➤ `kubectl get pods`

```
navmeet6447@cloudshell:~ (sharp-haven-356412)$ gcloud container clusters get-credentials cac-testgke-22 --region us-central1 --project sharp-haven-356412
Fetching cluster endpoint and auth data.
kubeconfig entry generated for cac-testgke-22.
navmeet6447@cloudshell:~ (sharp-haven-356412)$ kubectl get pods
No resources found in default namespace.
navmeet6447@cloudshell:~ (sharp-haven-356412)$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
gke-cac-testgke-22-cac-testnodepool-2-329e624d-f3jr	Ready	<none>	146m	v1.25.5-gke.2000
gke-cac-testgke-22-cac-testnodepool-2-adeaf317-06jj	Ready	<none>	146m	v1.25.5-gke.2000

```
navmeet6447@cloudshell:~ (sharp-haven-356412)$
```

10. To deploy the service, got to GCP Console and choose Workloads section in the GKE page and select the container image that has been pushed to our host location GCR and expose it to port number 8080 to obtain the external IP to our GKE. Then run the command:

➤ `kubectl get service`

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to sharp-haven-356412.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
navmeet6447@cloudshell:~ (sharp-haven-356412)$ kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
ksncvknknf-service	LoadBalancer	10.79.251.86	35.188.124.221	8080:30039/TCP	56s
kubernetes	ClusterIP	10.79.240.1	<none>	443/TCP	165m

```
navmeet6447@cloudshell:~ (sharp-haven-356412)$
```

11. Copy the External Ip and paste it in the Web Browser and the application is up-end and running.

