Dashboard

Wednesday, January 5, 2022 11:42 AM

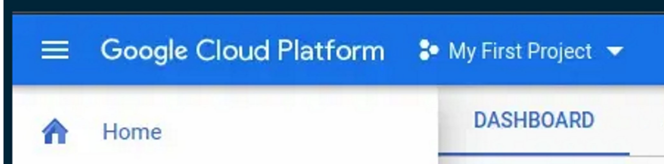
Setting Up Kubernetes Dashboard on GKE (Google Kubernetes Engine) on GCP.

Step 1 - Setup kubernetes cluster on GKE(Google Kubernetes Engine)

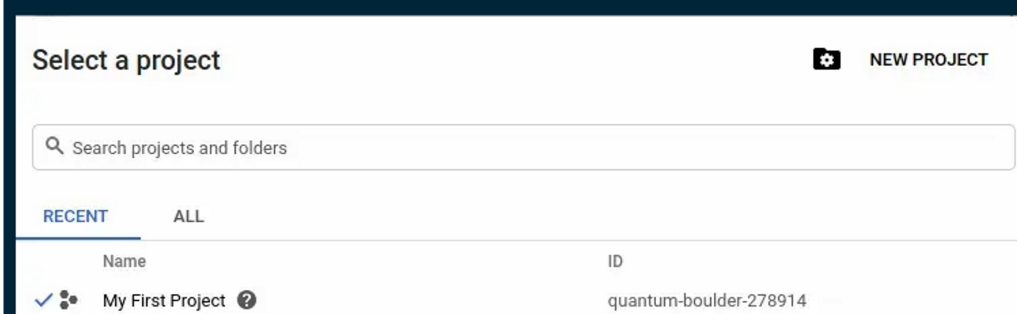
Goto - https://cloud.google.com/

(Note - If you do not have Google Cloud Account active then you can use the free credit provided by google to utilize its services. Google provides 300$ credit for 1 year and it is sufficient enough to start working with google cloud services)

So I am assuming now you have Google Cloud account, now you can goto *My First Project*

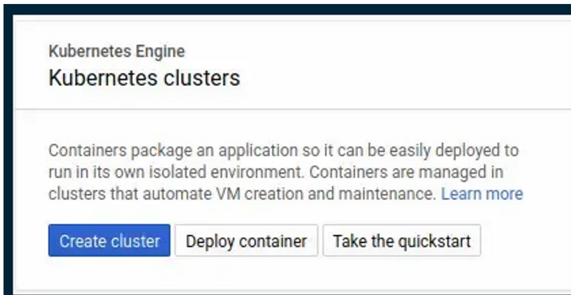


Then you can create a New Project



In this case I am creating a sample project with name - *jhooq-sprinboot-k8s-demo*

Now on the left hand navigation menu search for the menu **"Kubernetes Engine"** then click on "**Clusters**"



Now click on **Create Cluster** then it will show up Cluster Basics form where you need to fill in the **Cluster Name**.

So in our case I am putting the cluster name as - **jhooq-kubernetes-dashboard**

And after that click on the create cluster.(Cluster creation generally takes little time so be patient.)

After the cluster creation click on the **Connect** option to access the **Cluster Command Line Console**

On console it will by default put an command to get the credentials, so you just need to press enter or execute that command.

Step 2 - Setup dashboard or install dashboard

Now you need to run the dahsboard install command using kubectl

BASH

```
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.
```

$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml

```
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
```

Step 3 - Create service Account

Run the following command on the gcloud terminal for creating service account

BASH

```bash
cat <<EOF | kubectl create -f -
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```

cat <<EOF | kubectl create -f -
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
EOF

Step 4 - Create ClusterRoleBinding with the user which we have created in Step 3.

BASH

```bash
cat <<EOF | kubectl create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```


```
cat <<EOF | kubectl create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```

Step 5 - Start kubectl proxy

Now we need to start the kubectl proxy server

```bash
$ kubectl proxy
```

Step 6 - Generate secret token for accessing dashaboard

Use the following command to generate secret token because we need this token to access the dashboard

```bash
$ kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep a
```

$ kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep admin-user | awk '{print $1}')

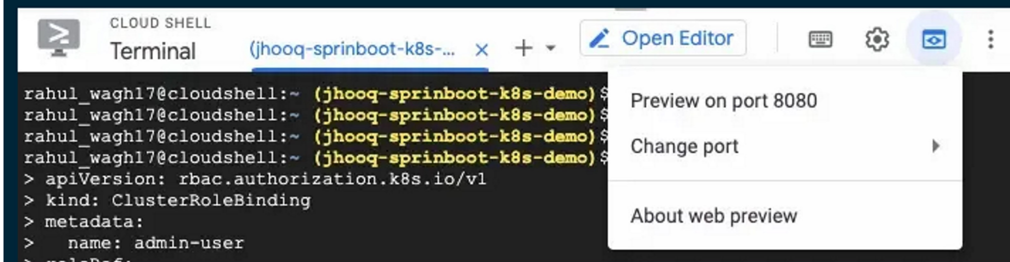
It should return something similar

```bash
Name:         default-token-sv6pq
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: default
              kubernetes.io/service-account.uid: df19ff8d-b64a-11ea-a6e6-42010a80021d

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1119 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXI
```

Step 7 - Access the dashboard

Since you are working on Google Cloud url will be little different for you.

To get the URL first you need to click on the **Web Preview** option inside the Google Cloud console.



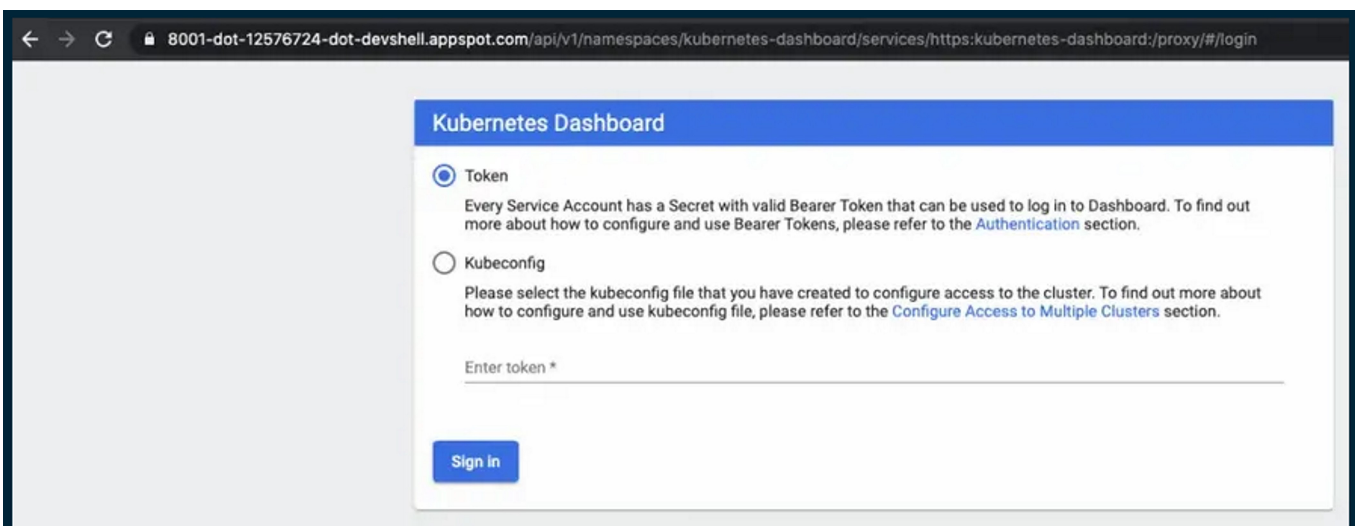
It will open a new tab in the browser with URL which is similar to https://8080-dot-12576724-dot-devshell.appspot.com/?authuser=0

You need to change few thing in the url for accessing the kubernetes dashboard

Change the port from 8080 to 8001 and replace the end url from /?authuser=0 to /api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login

So the final URL should look like this

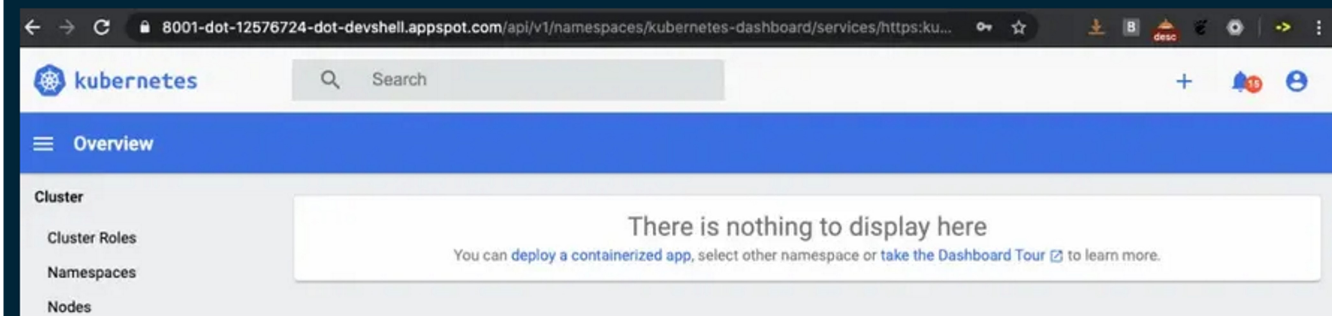
https://8001-dot-12576724-dot-devshell.appspot.com/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login



Step 8 - Use the token from Step 6

Select the option Token and paste the secret token generated in the Step 6 and then click on sign in.

And you should be able to access the dashboard on Google Kubernetes Engine



Ref : https://jhooq.com/