

Cloud Computing In-Class Lab 1

Name - Ravi Rajpurohit

UTA ID - 1002079916

Part B

Task 1 - Setup a default compute zone

Setup the VM instance and assigned a default compute zone as us-central1

The screenshot shows the Google Cloud Console interface for the 'ravi-cc-lab' project. The 'Compute Engine' section is active, displaying the 'VM instances' page. A table lists one instance named 'instance-1' in the 'us-central1-a' zone. The 'Related actions' section includes links for 'Explore Actifio GO', 'View billing report', 'Monitor VMs', and 'Explore VM logs'. The right sidebar shows the 'Select an instance' panel with tabs for 'PERMISSIONS', 'LABELS', and 'MONITORING'. The bottom status bar indicates a successful connection to the VM.

Status	Name	Zone	Recommendations	SSH
✓	instance-1	us-central1-a		SSH

Task 2 - Create a GKE cluster

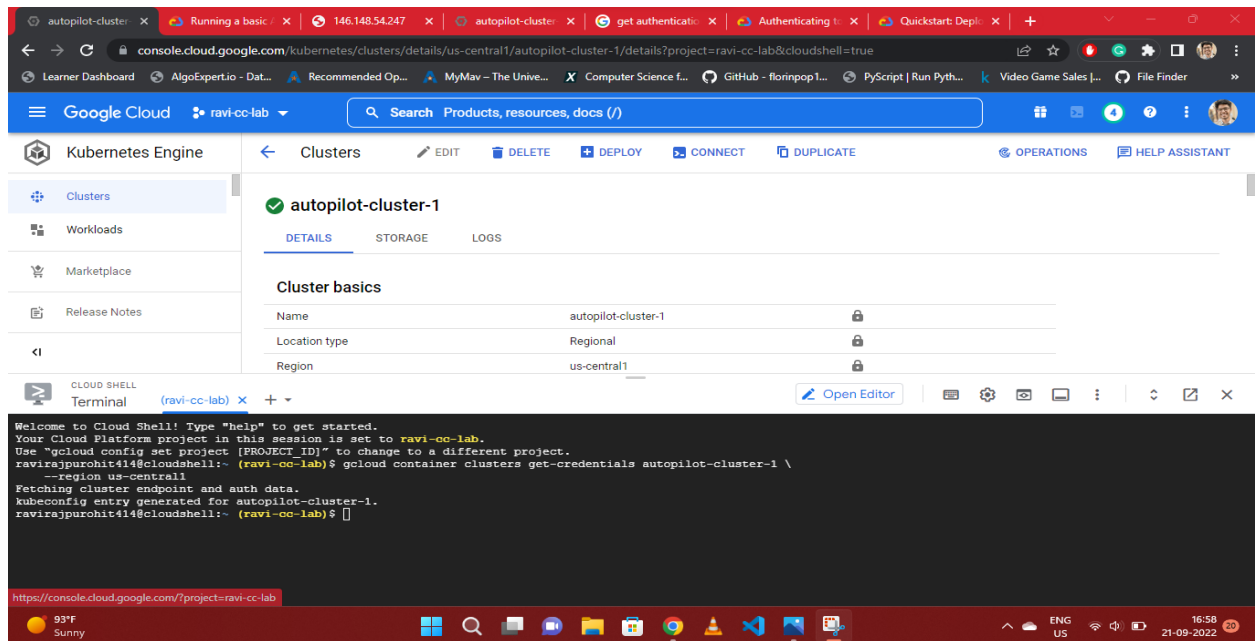
Created a Google Kubernetes Engine with the default configuration.

The screenshot shows the Google Cloud Console interface for the 'ravi-cc-lab' project, displaying the 'Kubernetes Engine' section. The 'Clusters' page shows a single cluster named 'autopilot-cluster-1'. The 'Cluster basics' section provides details about the cluster, including its location, region, and node configuration. A notification at the bottom states 'The cluster is created successfully.' The right sidebar shows the 'Tutorial' section with links for 'Configuring cluster access for kubectl', 'Upgrading a cluster or node pool', 'Set up HTTP(S) Load Balancing with Ingress', 'Exposing applications using services', and 'Installing Cloud Operations for GKE support'.

Name	Location type	Region	Default node zones	Release channel	Version	Endpoint
autopilot-cluster-1	Regional	us-central1	us-central1-f us-central1-a us-central1-b us-central1-c	Regular channel	1.22.12-gke.300	34.171.87.116

Task 3 - Get authentication credentials for the cluster

Used get-credentials command on console's terminal to fetch the information about the specific cluster



The screenshot displays the Google Cloud Console interface for the 'autopilot-cluster-1' cluster. The left sidebar shows the 'Kubernetes Engine' section with 'Clusters' selected. The main panel shows the 'DETAILS' tab for the cluster. The 'Cluster basics' section lists the following information:

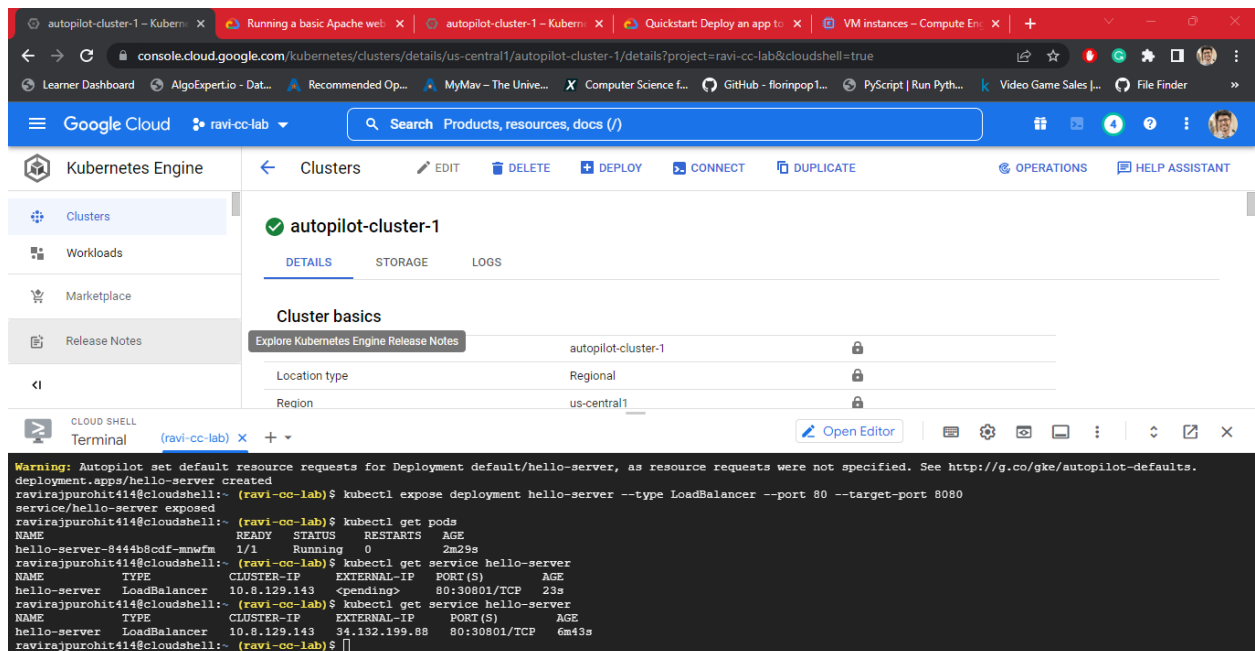
Field	Value
Name	autopilot-cluster-1
Location type	Regional
Region	us-central1

The terminal at the bottom shows the following commands and output:

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to ravi-cc-lab.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ gcloud container clusters get-credentials autopilot-cluster-1 \
--region us-central1
Fetching cluster endpoint and auth data.
kubeconfig entry generated for autopilot-cluster-1.
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$
```

Task 4 - Deploy an application to the cluster

Deployed a hello world app to the autopilot-cluster-1



The screenshot displays the Google Cloud Console interface for the 'autopilot-cluster-1' cluster. The left sidebar shows the 'Kubernetes Engine' section with 'Clusters' selected. The main panel shows the 'DETAILS' tab for the cluster. The 'Cluster basics' section lists the following information:

Field	Value
Name	autopilot-cluster-1
Location type	Regional
Region	us-central1

The terminal at the bottom shows the following commands and output:

```
Warning: Autopilot set default resource requests for Deployment default/hello-server, as resource requests were not specified. See http://g.co/gke/autopilot-defaults.
deployment.apps/hello-server created
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl expose deployment hello-server --type LoadBalancer --port 80 --target-port 8080
service/hello-server exposed
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl get pods
NAME READY STATUS RESTARTS AGE
hello-server-8444b8cdf-mnwfm 1/1 Running 0 2m29s
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl get service hello-server
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
hello-server LoadBalancer 10.8.129.143 <pending> 80:30801/TCP 23s
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl get service hello-server
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
hello-server LoadBalancer 10.8.129.143 34.132.199.88 80:30801/TCP 6m43s
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$
```

Task 5 - Deleting the cluster

To stop the use of services, deleted the cluster

```
hello-server    LoadBalancer    10.8.129.143    <pending>      80:30801/TCP    23s
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl get service hello-server
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
hello-server    LoadBalancer  10.8.129.143    34.132.199.88  80:30801/TCP    6m43s
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl get service hello-server
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)        AGE
hello-server    LoadBalancer  10.8.129.143    34.132.199.88  80:30801/TCP    32m
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ kubectl delete service hello-server
service "hello-server" deleted
ravirajpurohit414@cloudshell:~ (ravi-cc-lab)$ gcloud container clusters delete hello-cluster \
--region us-central1
The following clusters will be deleted.
- [hello-cluster] in [us-central1]
```