The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Assignment-2: Scaling Services

Instructions:

1. *This is an individual homework. Each student must submit his/her solution to the problem through Assignment-2 dropbox.*
2. All submissions are to be via the Quercus Submission System; no other way ofsubmission is accepted.
3. All submissions must be submitted through the dropbox by the due date or by the closure date, Clouse Date allows 3 days for late submissions.

Purpose and Objective

This assignment addresses distributed systems fault-tolerance, synchronization, data consistency, and service performance issues.

Keywords

Cluster, K8s, container composition, container deployment, serverless computing, authentication, authorization, Borg.

Resources

1. The Artifact Registry Documentation: https://cloud.google.com/artifact-registry/docs
2. Google Cloud Managed Service for Prometheus.
3. Cloud Monitoring Overview: https://cloud.google.com/monitoring/docs/monitoring-overviewLinks to an external site.
4. Example: Collecting Apache Web Server Metrics: https://cloud.google.com/monitoring/monitor-compute-engine-virtual-machineLinks to an external site.
    a. **Generate a traffic to a web server: timeout 120 bash -c -- 'while true; do curl localhost; sleep $((RANDOM % 4)) ; done'**
    b. Bash
        i. https://ss64.com/bash/syntax.html
        ii. https://ss64.com/bash/
    c. curl command, Linux Manual: https://ss64.com/bash/curl.html
    d. timeout command, Linux Manual: https://ss64.com/bash/timeout.html
    e. sleep command, Linux manual: https://ss64.com/bash/sleep.html
5. Google Compute Engine Metrics Collection (Ops Agent and Open Telemetry Protocol): https://cloud.google.com/monitoring/agent/ops-agentLinks to an external site.
6. Platforms Monitoring: https://cloud.google.com/stackdriver/docs/solutions/agents/ops-agent/third-partyLinks to an external site.

1. Google implemented the Borg system to manage clusters of servers/cells, identify the basic three functionalities that Borg accomplishes. Explain how separation of duties/isolation is accomplished in Borg. Identify the main components of Borg and describe how these components define its architecture.

   [20 marks]

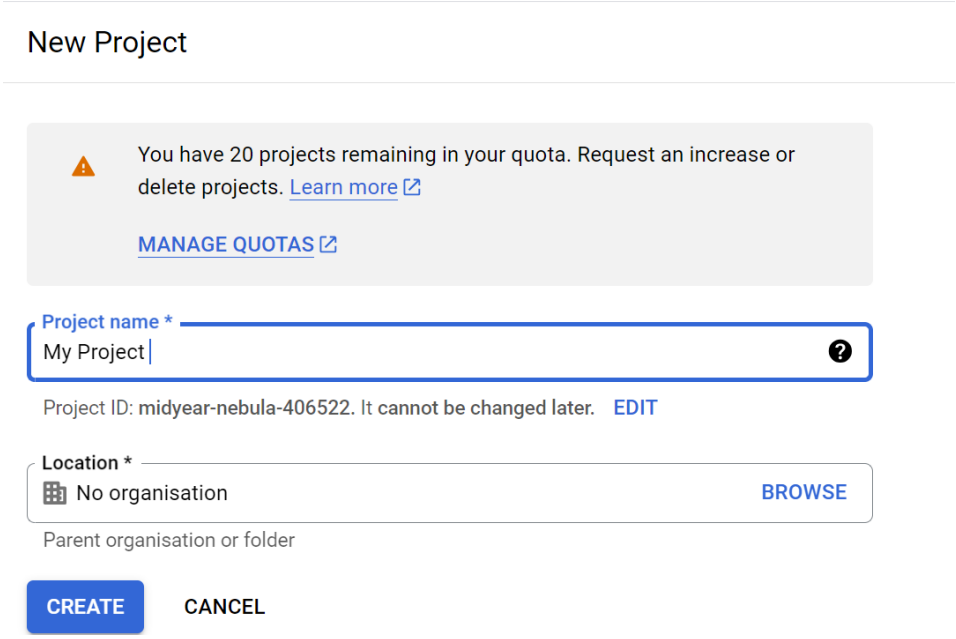
2. Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.
   - The name Kubernetes originates from Greek, meaning helmsman or pilot. K8s as an abbreviation results from counting the eight letters between the "K" and the "s". Google open-sourced the Kubernetes project in 2014. Kubernetes combines over 15 years of Google's experience running production workloads at scale with best-of-breed ideas and practices from the community.

     2.1 Identify when would you deploy Kubernetes.
     2.2 Identify the main components of Kubernetes.

     [20 marks]

3. Using the Google Cloud platform, create a serverless web service that scales automatically. Observe the performance of the service at high demand to verify the autoscaling feature. **[30 marks]**

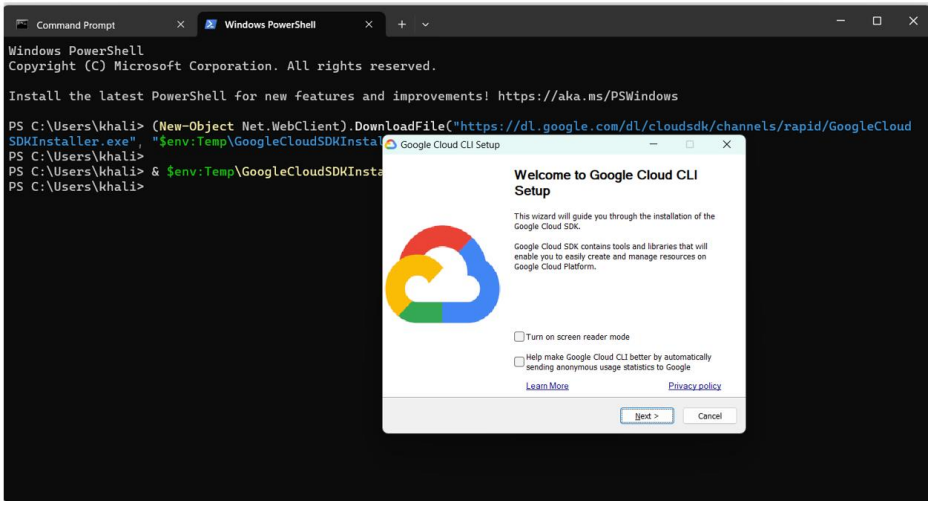
a. Start a new project on Google Cloud.
b. Name your project as q3assignment2

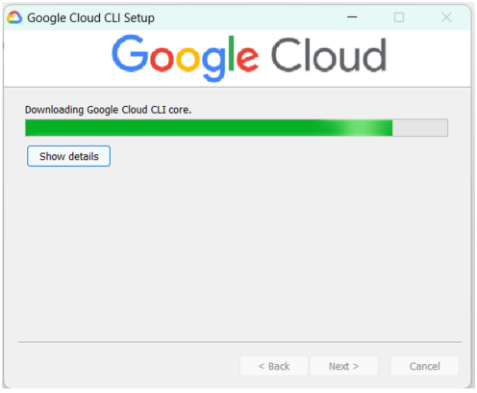


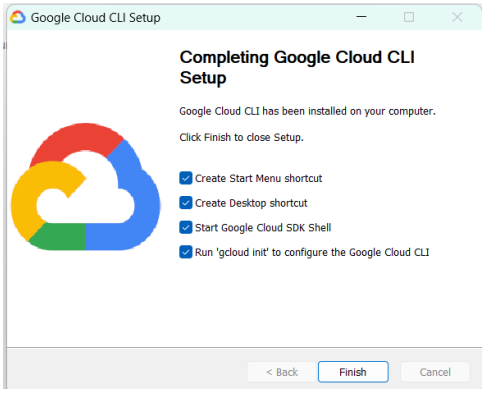
c. Install the Google Cloud Command Line Interface that matches your platform: https://cloud.google.com/sdk/docs/install

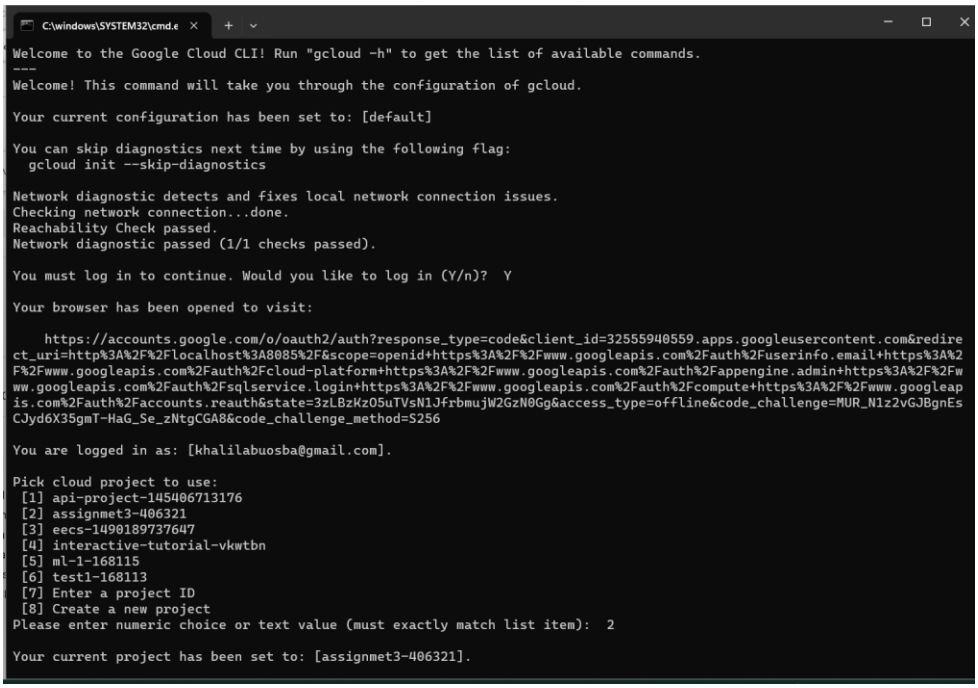
   a. If your platform is Windows, start a PowerShell window and run the following command to download the CLI installer:
   (New-Object Net.WebClient).DownloadFile("https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe", "$env:Temp\GoogleCloudSDKInstaller.exe") & $env:Temp\GoogleCloudSDKInstaller.exe

b.


Command Prompt
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\khali> (New-Object Net.WebClient).DownloadFile("https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe", "$env:Temp\GoogleCloudSDKInstal
PS C:\Users\khali>
PS C:\Users\khali> & $env:Temp\GoogleCloudSDKInsta
PS C:\Users\khali>
Google Cloud CLI Setup
Welcome to Google Cloud CLI Setup
This wizard will guide you through the installation of the Google Cloud SDK.
Google Cloud SDK contains tools and libraries that will enable you to easily create and manage resources on Google Cloud Platform.
Turn on screen reader mode
Help make Google Cloud CLI better by automatically sending anonymous usage statistics to Google
Learn More
Privacy policy
Next >
Cancel

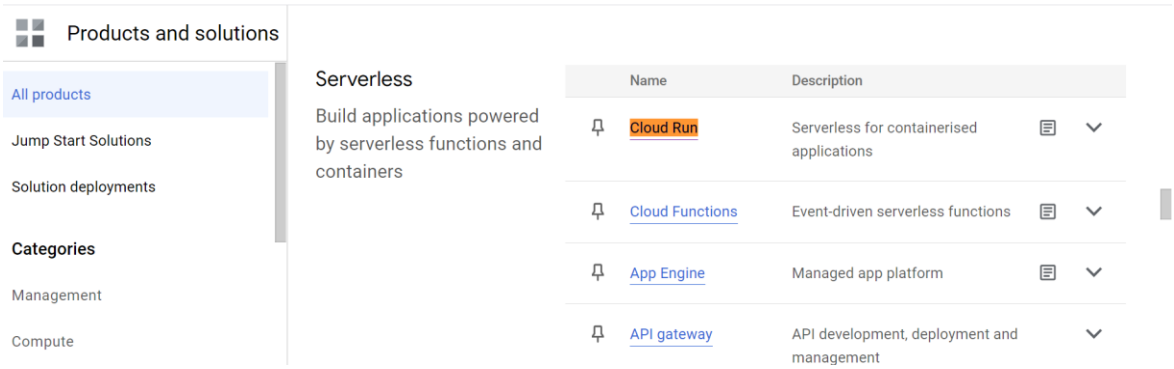

Google Cloud CLI Setup
Google Cloud
Downloading Google Cloud CLI core.
Show details
< Back
Next >
Cancel


Google Cloud CLI Setup
Completing Google Cloud CLI Setup
Google Cloud CLI has been installed on your computer.
Click Finish to close Setup.
Create Start Menu shortcut
Create Desktop shortcut
Start Google Cloud SDK Shell
Run 'gcloud init' to configure the Google Cloud CLI
< Back
Finish
Cancel

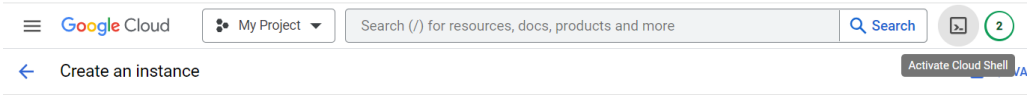

C:\windows\SYSTEM32\cmd.e
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.

Welcome! This command will take you through the configuration of gcloud.
Your current configuration has been set to: [default]
You can skip diagnostics next time by using the following flag:
gcloud init --skip-diagnostics
Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).
You must log in to continue. Would you like to log in (Y/n)? Y
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=3zLBzKzO5uTVsN1JfrbmujW2GzN0Gg&access_type=offline&code_challenge=MUR_N1z2vGJBgnEsCJyd6X35gmT-HaG_Se_zNtgCGA8&code_challenge_method=S256
You are logged in as: [khalilabuosba@gmail.com].
Pick cloud project to use:
[1] api-project-1454067131176
[2] assignmet3-406321
[3] eecs-1490189737647
[4] interactive-tutorial-vkwtbn
[5] ml-1-168115
[6] test1-168113
[7] Enter a project ID
[8] Create a new project
Please enter numeric choice or text value (must exactly match list item): 2
Your current project has been set to: [assignmet3-406321].

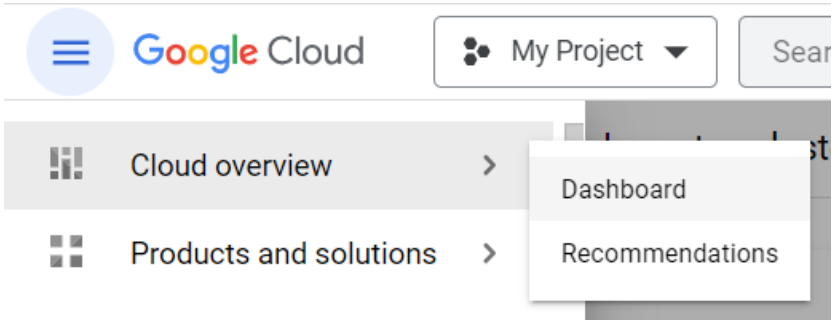
d. Run the CLI initialization command and pick the project to use: **gcloud init**
e. Under Serverless menu, of Products and Service, select Cloud Run:



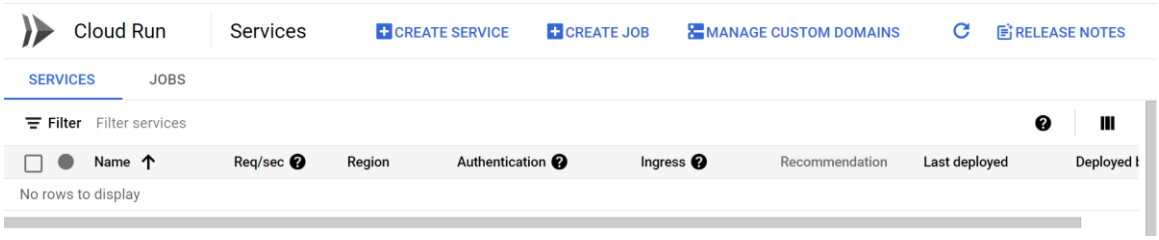
f. Click > to activate the cloud shell



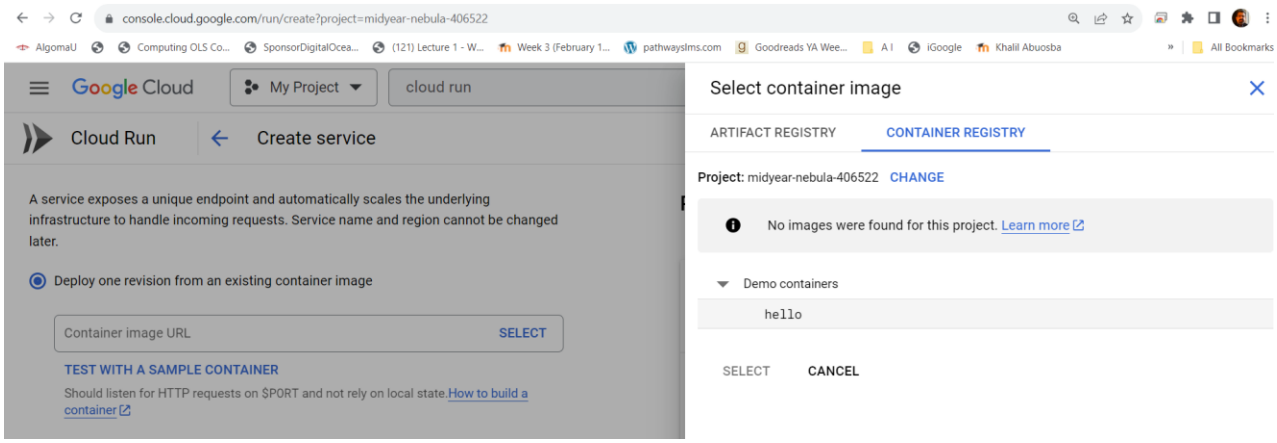
g. To initialize the environment, list the projects and select your project, Run the command > **gcloud init**
h. You can select the project using the command
> **gcloud config set** project_name
i. Click on ➔ Navigation Menu ➔ Dashboard



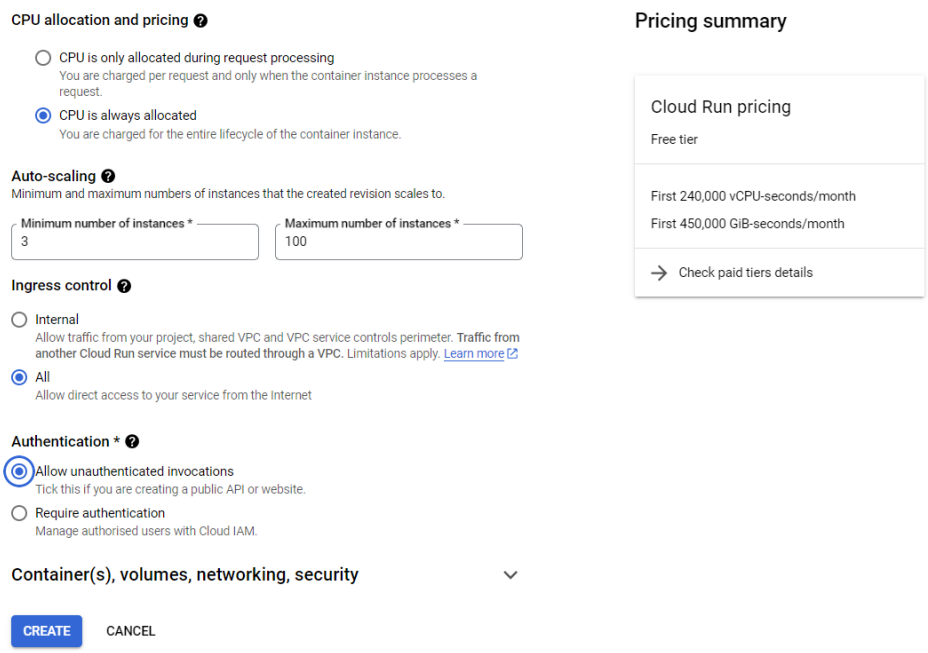
j. Click ➔ Cloud Run



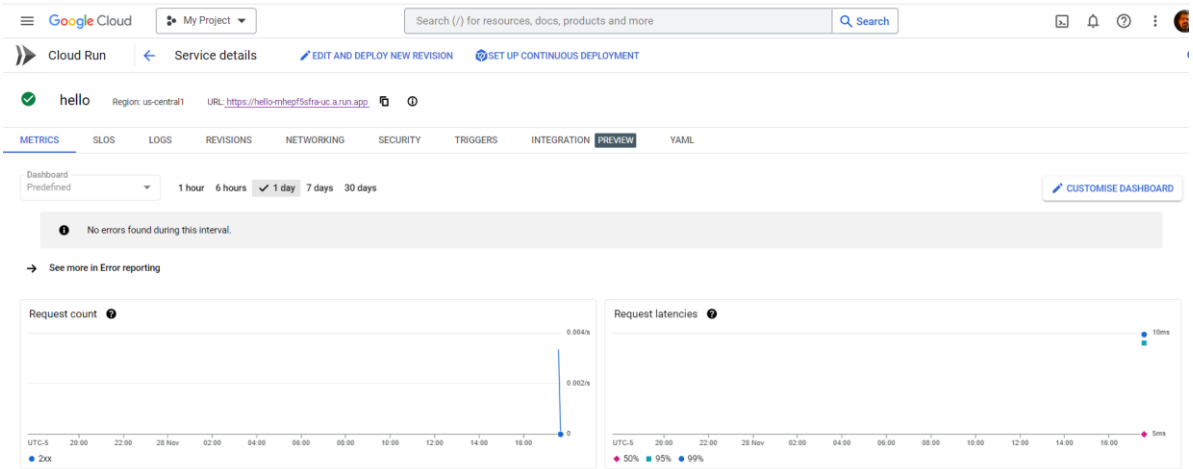
k. Click Select ➔ Container Registry ➔ Demo Containers ➔ hello ➔ Select



l. Configure the environment as the following except minimum number of instances as 1 instead of 3, click ➔ Create



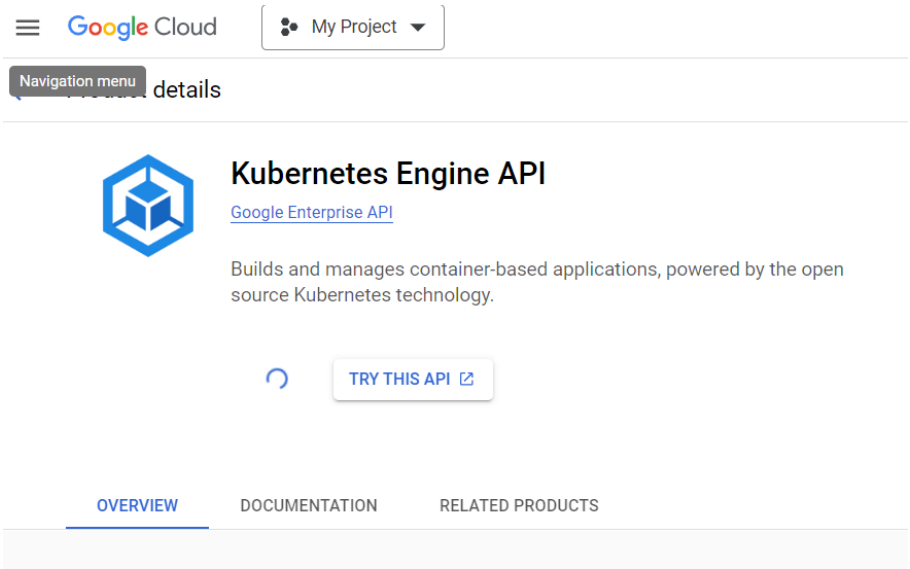
m. From the YAML file, observer the number of replicas; how many replicas where specified?
n. What is the URL specified for consuming the service.
o. What is the result of parsing the URL in the browser.
p. Click on View on Cloud Council:



q. Attempt to flood the service with http requests for 10 to 30 minutes (i.e. you may use curl, open multiple terminal or cmd sessions and request the contents of the homepage using the redundant curl commands).
r. What is the analytical summary output of the Container Instance Count and container CPU Utilization?

4. Using the Google Kubernetes Engine (GKE) services, create a K8s cluster and deploy a service to the cluster. [30 marks]

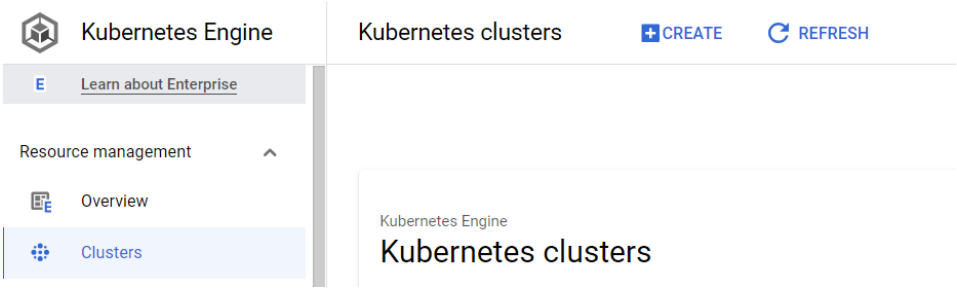
1. **Make sure that the GKE API is enabled ➔ Navigation Menu ➔ Kubernetes Engine ➔ Kubernetes API ➔ Enable.**



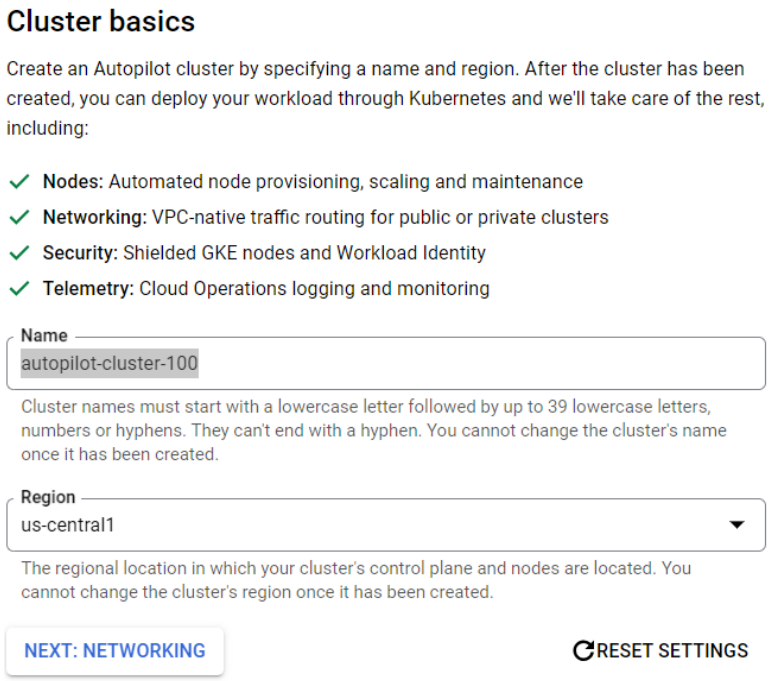
a. Review the Create Cluster | Deploy Container tutorial: https://console.cloud.google.com/kubernetes/list/overview?project=midyear-nebula-406522

2. **Create a Kubernetes *cluster*, which provides compute, storage, networking, and other services for applications, similar to a virtual data center.**

   a. Click Kubernetes Engine ➔ Cluster ➔ Create

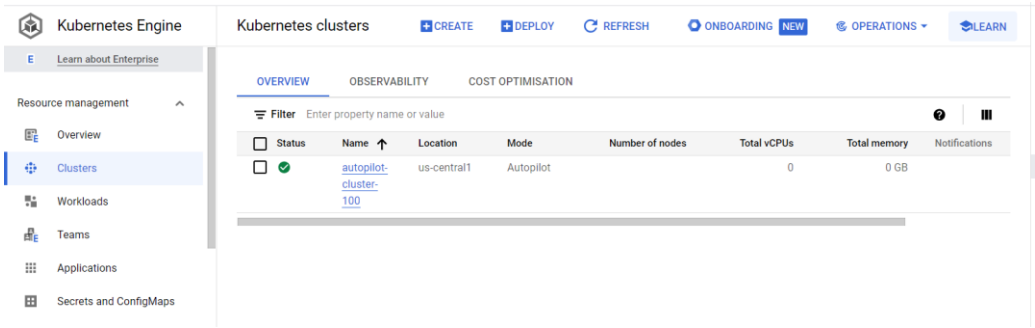


b. Specify the name of the cluster as autopilot-cluster-100 ➔ Next



c. Next ➔ Next ➔ Next (Default Values) ➔ Review and Create

d. ➔ Create Cluster.



e. Click observability.

b. Capture your version of the Container Error Logs/Second chart and paste in your report.

3. Service Composition:

a. This app (hello-app) composes a web server written in Go that responds to requests with the message Hello World!

b. The process requires the source code that may be imported from the github Google Cloud Platform repository using the command:

`git clone https://github.com/GoogleCloudPlatform/kubernetes-engine-samples; cd kubernetes-engine-samples/quickstarts/hello-app`

c. Set the Project_Name environment variable to your Google Cloud project name in the export command below.
The Project_Name variable associates the container image with your project's Artifact Registry:
`export PROJECT_ID = Your_Project_Name`

d. Verify the project name using the echo command: **`echo ${PROJECT_ID}`**

e. **Build the Docker image**

a. Create a repository (create a repository to store your container image)

1. specify a region for your repository. Run the following command to see a list of possible regions, then select the one closest to you:
`gcloud artifacts locations list`

2. Create a **REGION** variable to hold your selected region. The following command sets REGION to us-west1:
`export REGION=us-west1`

3. Run the following command to create your repository.
`gcloud artifacts repositories create hello-repo --repository-format=docker --location=${REGION} --description="Docker repository"`

GKE accepts Docker images as the application deployment format. Before deploying the hello-app application, you must package the source code as a Docker image.

To build a Docker image, you need source code and a Dockerfile. A Dockerfile contains instructions on how the image is built.

b. Build and tag the Docker image for hello-app:
`docker build -t ${REGION}-docker.pkg.dev/${PROJECT_ID}/hello-repo/hello-app:v1 .`

```
kstarts/hello-app (midyear-nebula-406522)$ docker build -t us-west1-docker.pkg.dev/midyear-nebula-406522/hello-repo/hello-app:v1 .
[+] Building 36.7s (13/13) FINISHED                                                                                docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 983B                                                                                          0.0s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [internal] load metadata for gcr.io/distroless/base-debian11:latest                                                       0.4s
 => [internal] load metadata for docker.io/library/golang:1.21.0                                                              0.8s
 => [builder 1/5] FROM docker.io/library/golang:1.21.0@sha256:b490ae1f0ece153648dd3c5d25be59a63f966b5f9e1311245c947de4506981aa  17.4s
 => => resolve docker.io/library/golang:1.21.0@sha256:b490ae1f0ece153648dd3c5d25be59a63f966b5f9e1311245c947de4506981aa          0.0s
 => => sha256:b490ae1f0ece153648dd3c5d25be59a63f966b5f9e1311245c947de4506981aa 2.36kB / 2.36kB                                0.0s
 => => sha256:434a1905832963ddddafdfb5a8f329a0c0b78a3529c99d57259f0888e0feb973 7.06kB / 7.06kB                                0.0s
 => => sha256:de4cac68b6165c40cf6f8b30417948c31be03a968e233e55ee40221553a5e570 49.56MB / 49.56MB                              0.6s
```

This command instructs Docker to build the image using the Dockerfile in the current directory and tag it with a name, such as us-west1-docker.pkg.dev/your-project_name/hello-repo/hello-app:v1. The image is pushed to Artifact Registry in the next section

1. Run the docker images command to verify that the build was successful:
docker images

2. **Push the image to the repository**

3. Enable the registry API:
   gcloud services enable artifactregistry.googleapis.co

4. Configure the Docker command-line tool to authenticate to Artifact Registery:
   gcloud auth configure-docker ${REGION}-docker.pkg.dev

5. Push the image to the repository
   docker push ${REGION}-docker.pkg.dev/${PROJECT_ID}/hello-repo/hello-app:v1

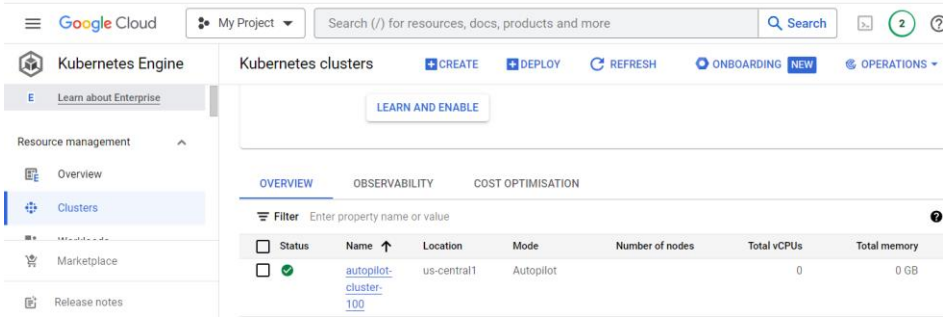
```
khalilabuosba@cloudshell:~/kubernetes-engine-samples/quickstarts/hello-app/kubernetes-engine-samples/quickstarts/hello-app/kubernetes-engine-samples/quic
kstarts/hello-app (midyear-nebula-406522)$ docker push ${REGION}-docker.pkg.dev/midyear-nebula-406522/hello-repo/hello-app:v1
The push refers to repository [us-west1-docker.pkg.dev/midyear-nebula-406522/hello-repo/hello-app]
9ac1c5cd32f7: Preparing
9ac1c5cd32f7: Pushed
714f56238fb5: Pushed
f33e343848bd: Pushed
4cb10dd2545b: Pushed
```

6. List all repositories
   gcloud artifacts repositories list

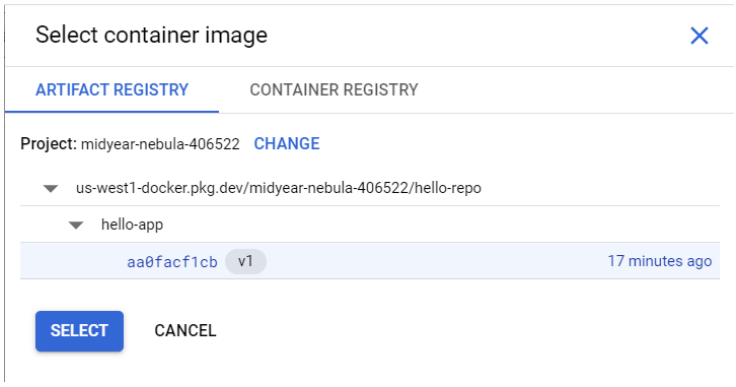
```
kstarts/hello-app (midyear-nebula-406522)$ gcloud artifacts repositories list
Listing items under project midyear-nebula-406522, across all locations.

ARTIFACT_REGISTRY

REPOSITORY: hello-repo
FORMAT: DOCKER
MODE: STANDARD_REPOSITORY
DESCRIPTION: Docker repository
LOCATION: us-west1
LABELS:
ENCRYPTION: Google-managed key
CREATE_TIME: 2023-11-29T06:27:03
```

4. **Deploy a sample app to the K8s cluster. Apps and their associated services that are running in Kubernetes are called *workloads*.**

   a. Click Kubernetes Engine➔ Clusters ➔ Deploy

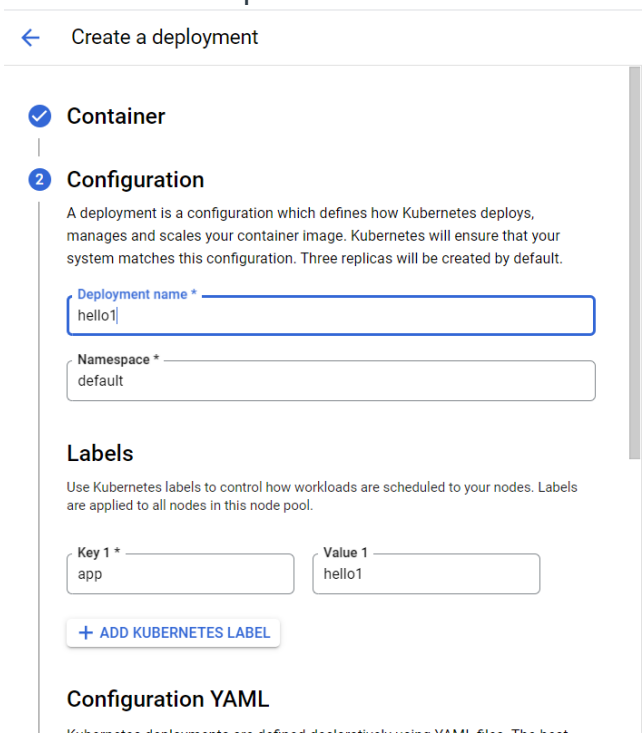


   b. In Image Path, click ➔ Select ➔ Artifact Registry
      ➔ Select the hello-repo repository
      ➔ Select hello-app image
      ➔ Click SELECT

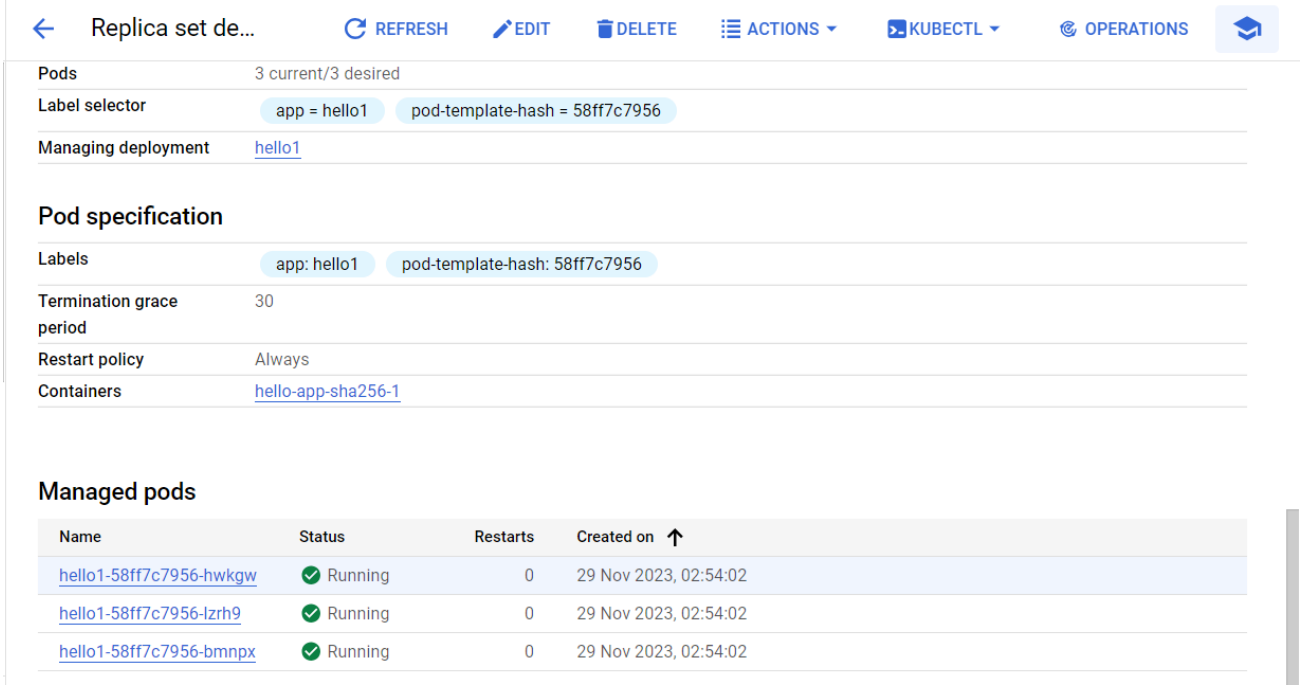


c. Specify a deployment name,

d. Refer to the YAML deployment configuration file, what are the minimum and maximum replicas values?



e. ➔ Click Deploy.



f. Click Workloads ➔ Expose