

Montréal  
GDC Sandbox Workshop



### Lab Guide

The goal of the workshop is to impart participants with an understanding of GDC sandbox and practical knowledge to deploy workloads. To ensure a smooth experience during the workshop, attendees are requested to install the necessary developer tools beforehand.

The lab section of the workshop is specifically designed for developers and requires these tools to complete the assigned tasks.

#### **Pre- Requisites:**

**Bring Your Own Device** with Admin access or have access to a **Virtual Desktop** with Admin access.

#### **Developer Tools:**

1. Enable WSL ( If you're running a Windows)

Guide: <https://learn.microsoft.com/en-us/windows/wsl/install>

2. Install Git

Guide: <https://git-scm.com/install/linux>

3. Install VSCode (IDE)

Guide: <https://code.visualstudio.com/download>

4. Install gcloud CLI. (Google Cloud Toolkit)

Guide: <https://docs.cloud.google.com/sdk/docs/install#linux>

You may skip the installation instructions if the required developer tools are already present on your workstation.

**Lab steps:**

**1. Clone the repo locally.**

Once all Prerequisites are installed, clone the repo locally on your machine.

Git clone <https://github.com/manzalam-gcp/gdc-sandbox-workshop.git>

**2. Edit the .env file in vi on VSCode**

```
cd ~/gdc-sandbox-workshop  
cp .env.example .env
```

```
# edit .env for specifics  
vi .env
```

**3. Run source .env**

**4. Establish the IAP tunnel**

```
./sandbox.sh tunnel
```

**5. RDP into your Sandbox jumpbox with your assigned sandbox user.**

**6. Clone the repo locally on sandbox.**

Open a terminal and clone the repo locally.

Git clone <https://github.com/manzalam-gcp/gdc-sandbox-workshop.git>

**7. From your workstation copy the .env file to the sandbox.**

```
./sandbox.sh env
```

**8. Edit the project\_config file with your landing zone info**

```
vi projects_config.yaml
```

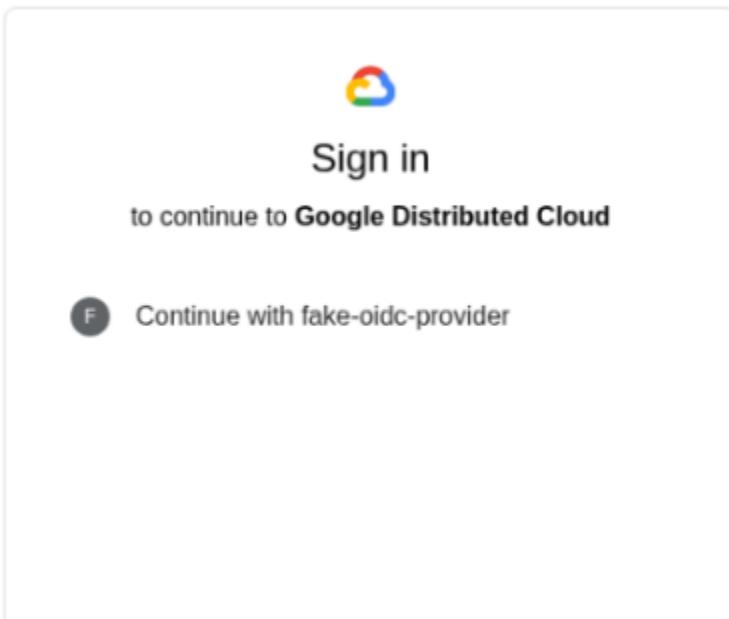
**9. Copy the Landing Zone config file to the sandbox.**

```
./sandbox.sh config
```

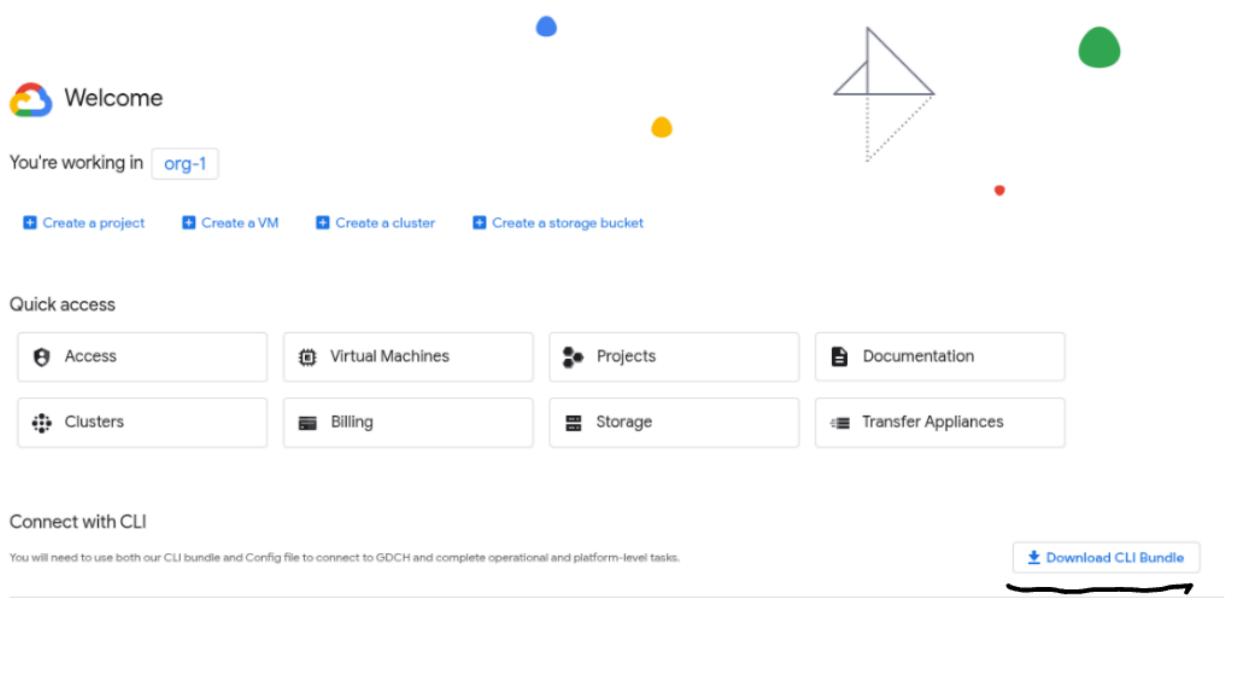
## Landing Zone

1. Download the GDCloud toolkit from the GDC Console.

Navigate to the GDC Console: <https://console.org-1.zone1.google.gdch.test>



2. Login as Platform Admin
3. Download the CLI bundle from the landing page.



4. Run './000-install-gdcloud.sh' to install GDCloud toolkit
5. Run ./001-create-projects.py to create your workloads Project
6. Run ./002-apply-role-bindings.py Create your users and apply role bindings to your project
7. Run ./003-createharborproject.py Create your project on the Harbor Instance
8. Log into Harbor, generate a user secret

Log into the GDC Sandbox console with your login credentials and navigate to the Harbor menu.

## Harbor Container Registry

## Harbor Container Registry

Harbor is an open source registry that secures artifacts with policies and role-based access control, ensures images are scanned and free from vulnerabilities, and signs images as trusted.

Select a zone [Zone: zone1 ▾](#)

### Harbor Instance

Status	Ready
Name	<a href="#">user-haas-instance</a> ↗

### Harbor projects

A project in Harbor contains all repositories of an application. Images cannot be pushed to Harbor before a project is created. To view and manage your Harbor projects, [go to Harbor Instance](#) ↗

[Create A Harbor Project](#)

Click on “Login via IAP provider”



Harbor

Search Harbor...

# Harbor

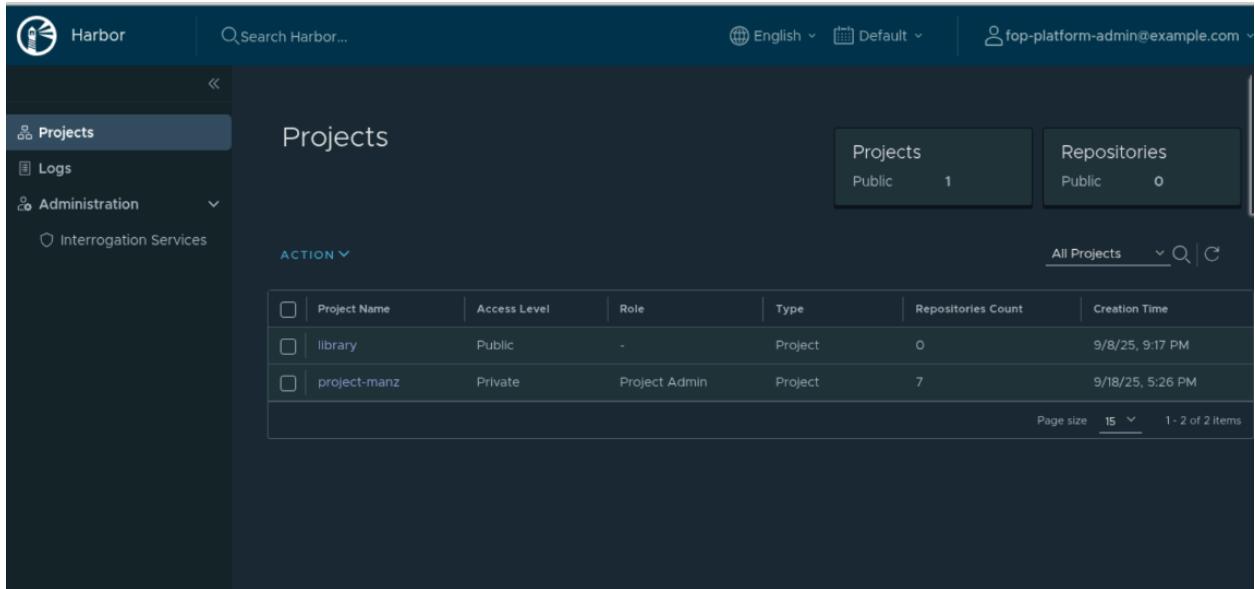
[LOGIN VIA IAP PROVIDER](#)

OR

[LOGIN VIA LOCAL DB](#)

[More info...](#)

Click on your project



Harbor

Search Harbor...

English Default top-platform-admin@example.com

Projects

Logs

Administration

Interrogation Services

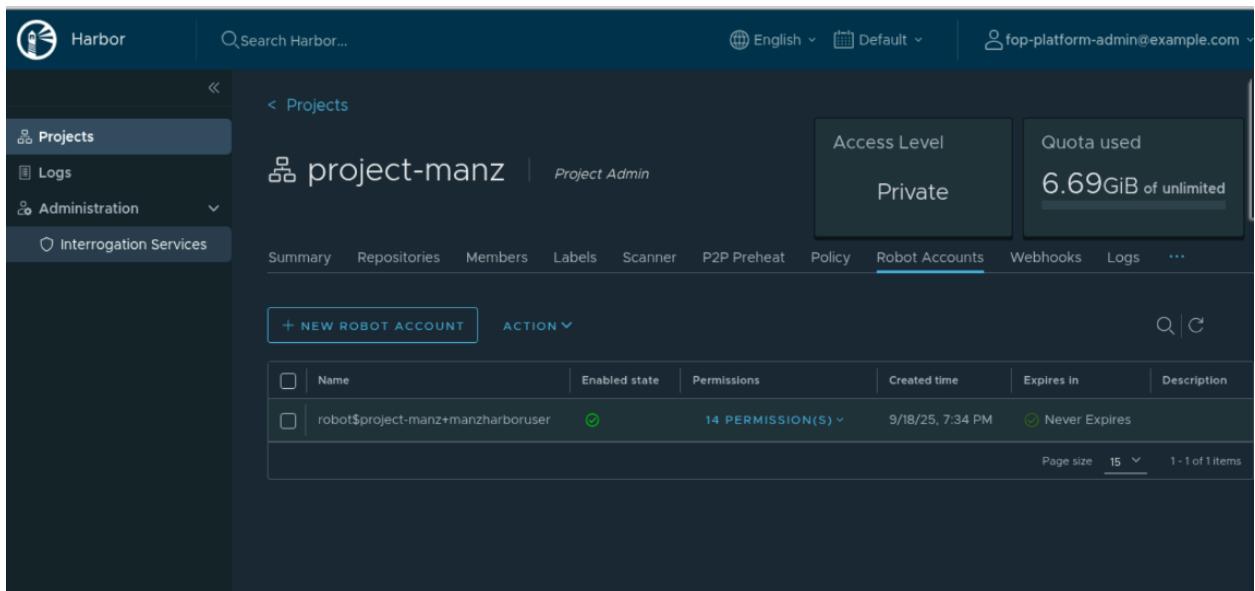
ACTION

All Projects

Project Name	Access Level	Role	Type	Repositories Count	Creation Time
library	Public	-	Project	0	9/8/25, 9:17 PM
project-manz	Private	Project Admin	Project	7	9/18/25, 5:26 PM

Page size 15 1-2 of 2 items

Click on “robot Accounts”



Harbor

Search Harbor...

English Default top-platform-admin@example.com

< Projects

project-manz Project Admin

Access Level Private

Quota used 6.69GiB of unlimited

Summary Repositories Members Labels Scanner P2P Preheat Policy Robot Accounts Webhooks Logs ...

+ NEW ROBOT ACCOUNT ACTION

Name	Enabled state	Permissions	Created time	Expires in	Description
robot\$project-manz+manzharboruser	Enabled	14 PERMISSION(S)	9/18/25, 7:34 PM	Never Expires	

Page size 15 1-1 of 1 items

Create the Robot account and copy the secret.

project-manz | Project Admin

Access Level  
Private

Summary

+ NEW R

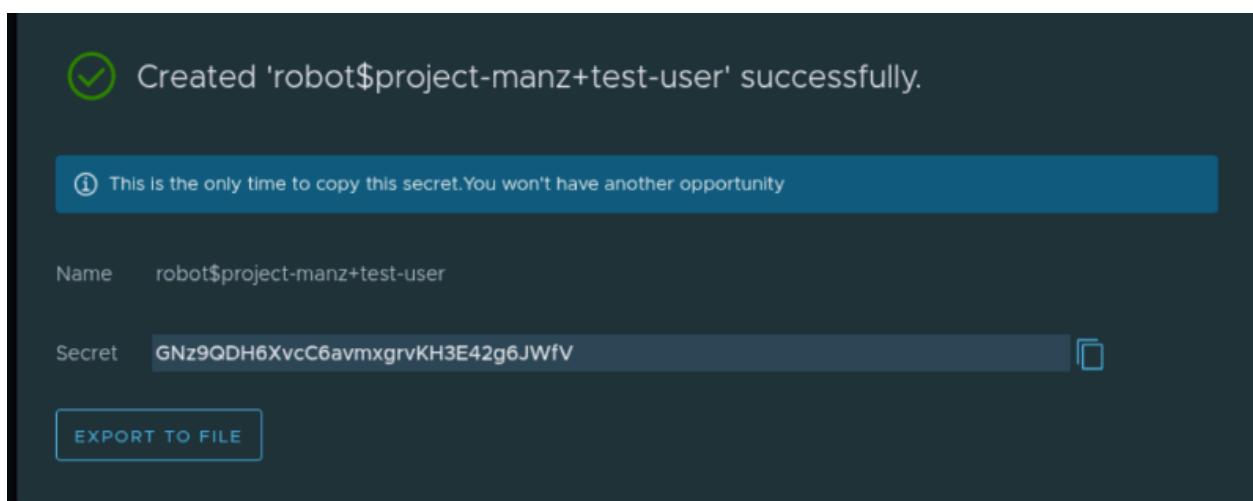
Name  Expiration time

Description

Permissions

CANCEL ADD

This screenshot shows the 'Create Robot Account' dialog box. The 'Name' field is populated with 'test-user'. The 'Expiration time' dropdown is set to 'Never'. There is a large empty 'Description' input field. Below the form, a 'Permissions' section indicates '14 PERMISSION(S)'. At the bottom right are 'CANCEL' and 'ADD' buttons.



Update the harbor username and password in the .env file with the stored username and secret:

```
HARBOR_USERNAME='robot$project_name+harbor'
```

```
HARBOR_PASSWORD=nnnnnnmmmmmm111111
```

9. Run ./004-addharborsecret.py Create docker registry secret

## Labs

### **Lab 1 - Deploy HTML Server**

Follow the Lab Guide on Github:

<https://github.com/manzalam-gcp/gdc-sandbox-workshop/blob/main/LAB-1.md>

### **Lab 2 - Deploy API Server**

Follow the Lab Guide on Github:

<https://github.com/manzalam-gcp/gdc-sandbox-workshop/blob/main/LAB-2.md>

### **Lab 3 - Deploy Elasticsearch**

Follow the Lab Guide on Github:

<https://github.com/manzalam-gcp/gdc-sandbox-workshop/blob/main/LAB-2.md>