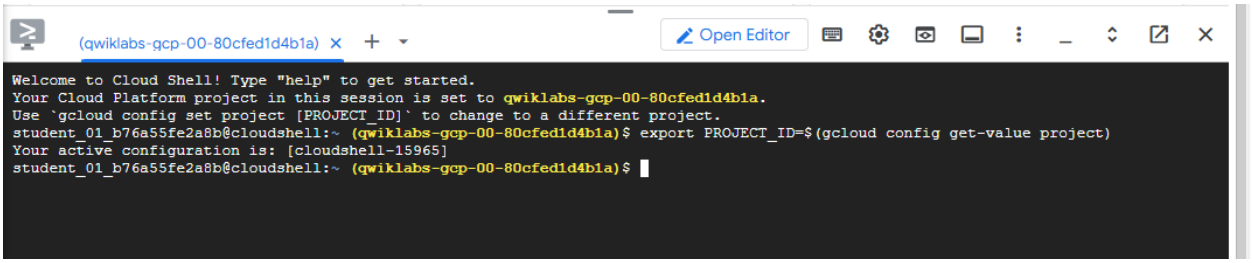
Artifact Registry: Qwik Start

Task 1. Create a Docker repository

In this section, you'll create a private Docker repository within Artifact Registry. This repository will serve as a central location to store and manage your Docker images. You'll use the gcloud command-line tool to create the repository and then verify its creation through the Google Cloud Console.

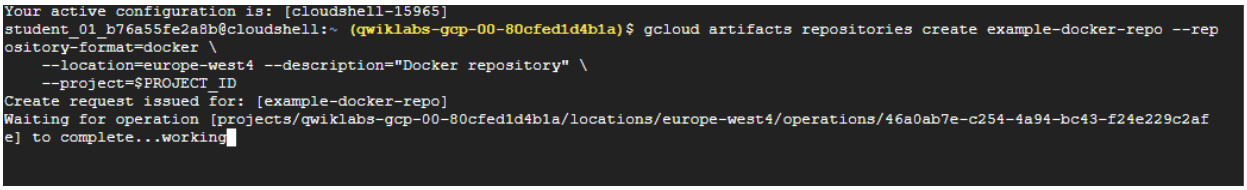
1. Open a new Cloud Shell window by clicking the icon (>_) in the top right corner of the console.
2. Run the following command to get your Project ID and save it as an environment variable:

```
export PROJECT_ID=$(gcloud config get-value project)
```



3. Run the following command to create a new Docker repository named example-docker-repo in the location Region with the description "Docker repository".

```
gcloud artifacts repositories create example-docker-repo --repository-format=docker \
  --location=Region --description="Docker repository" \
  --project=$PROJECT_ID
```



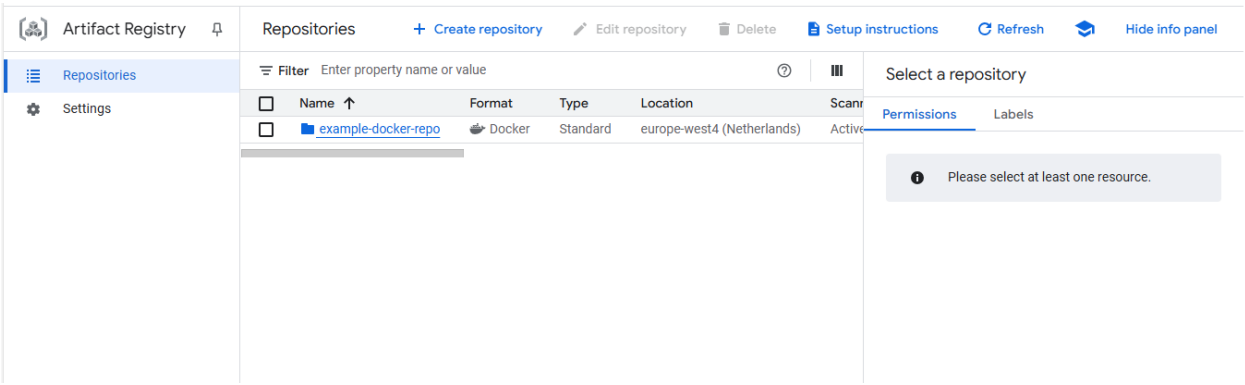
4. Run the following command to verify that your repository was created.

```
gcloud artifacts repositories list \
  --project=$PROJECT_ID
```

```
ARTIFACT_REGISTRY

REPOSITORY: example-docker-repo
FORMAT: DOCKER
MODE: STANDARD_REPOSITORY
DESCRIPTION: Docker repository
LOCATION: europe-west4
LABELS:
ENCRYPTION: Google-managed key
CREATE_TIME: 2025-05-29T04:55:10
UPDATE_TIME: 2025-05-29T04:55:10
SIZE (MB): 0
student_01_b76a55fe2a8b@cloudshell:~ (qwiklabs-gcp-00-80cfed1d4b1a)$
```

5. From the search bar at the top of the console, type **Artifact Registry** and select the first result.
6. On the **Artifact Registry** product page, verify you can see your repository. It should resemble the following:



7. Click the **example-docker-repo** repository. You should notice there are no files inside the repository. In the next sections, you will be adding files to the repository.

Task 2. Configure authentication for Artifact Registry

To push and pull images from your newly created Docker repository, you need to configure Docker to authenticate with Artifact Registry. This involves setting up credentials that allow your Docker client to interact with the repository securely.

Before you can push or pull images, you will need to configure Docker to use the Google Cloud CLI to authenticate requests to Artifact Registry.

1. To set up authentication to Docker repositories in the region Region, run the following command:

   gcloud auth configure-docker Region-docker.pkg.dev

The command updates your Docker configuration. You can now connect with Artifact Registry in your Google Cloud project to push and pull images.

Task 3. Obtain an image to push

You'll need a Docker image to work with in this lab. Instead of building an image from scratch, you'll pull a pre-built sample image from a public repository. This will allow you to focus on interacting with Artifact Registry.

For this lab, you will push a sample image named hello-app.

1. Run the following command to pull version 1.0 of the image.

   docker pull us-docker.pkg.dev/google-samples/containers/gke/hello-app:1.0

   Image paths in Artifact Registry include multiple parts. For this sample image:

- us-docker.pkg.dev is the hostname for container images stored in Artifact Registry Docker repositories, which includes the location of the repository (us).
- google-samples is the project ID.
- containers is the repository ID.
- /gke/hello-app is the path to the image in the repository containers.

Task 4. Add the image to the repository

Now you'll add the sample image to your private repository. This involves tagging the image with the repository name to specify its destination and then pushing it to Artifact Registry.

Before you push the Docker image to Artifact Registry, you must tag it with the repository name.

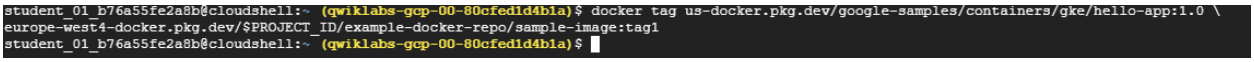
Tag the image with a registry name

Tagging the image ensures it's pushed to the correct location, which for this lab is region-docker.pkg.dev.

1. Run the following command to tag the image as sample-image:tag1:

   docker tag us-docker.pkg.dev/google-samples/containers/gke/hello-app:1.0 \

   Region-docker.pkg.dev/$PROJECT_ID/example-docker-repo/sample-image:tag1



Where:

- Region is the repository location.
- region-docker.pkg.dev is the hostname for the Docker repository you created.
- $PROJECT_ID is your Google Cloud Project ID.
- example-docker-repo is the ID of the repository you created.
- sample-image is the image name you want to use in the repository. The image name can be different than the local image name. For this lab you will store the image directly under the repository ID example-docker-repo.
- tag1 is a tag you're adding to the Docker image. If you didn't specify a tag, Docker will apply the default tag latest.

Push the image to Artifact Registry

After you have configured authentication and tagged the local image, you can push the image to the repository that you created.

To push the Docker image, run the following command:

docker push Region-docker.pkg.dev/$PROJECT_ID/example-docker-repo/sample-image:tag1

```
student_01_b76a55fe2a8b@cloudshell:~ (qwiklabs-gcp-00-80cfed1d4b1a)$ docker push europe-west4-docker.pkg.dev/$PROJECT_ID/example-docker-repo/sample-image:tag1
The push refers to repository [europe-west4-docker.pkg.dev/qwiklabs-gcp-00-80cfed1d4b1a/example-docker-repo/sample-image]
ac973dee44dc: Preparing
6835249f577a: Preparing
24aacbf97031: Preparing
8451c71f8c1e: Preparing
2388d21e8e2b: Preparing
c048279a7d9f: Waiting
1a73b54f556b: Waiting
2a92d6ac9e4f: Waiting
bbb6cacb8c82: Waiting
ac805962e479: Waiting
af5aa97ebe6c: Waiting
4d049f83d9cf: Waiting
9ed498e122b2: Waiting
577c8ee06f39: Waiting
5342a2647e87: Waiting
```

Task 5. Pull the image from Artifact Registry

Finally, you'll pull the image that you just pushed to your private repository. This simulates how you would access and use images stored in Artifact Registry in a real-world scenario.

1. To pull the image from Artifact Registry onto your local machine, run the following command:

   docker pull Region-docker.pkg.dev/$PROJECT_ID/example-docker-repo/sample-image:tag1

```
student_01_b76a55fe2a8b@cloudshell:~ (qwiklabs-gcp-00-80cfed1d4b1a)$ docker pull europe-west4-docker.pkg.dev/$PROJECT_ID/example-docker-repo/sample-image:tag1
tag1: Pulling from qwiklabs-gcp-00-80cfed1d4b1a/example-docker-repo/sample-image
Digest: sha256:9af0e71960f960390f0de6b3f509068da3f0e458a4e591d20dcc751d856fe244
Status: Image is up to date for europe-west4-docker.pkg.dev/qwiklabs-gcp-00-80cfed1d4b1a/example-docker-repo/sample-image:tag1
europe-west4-docker.pkg.dev/qwiklabs-gcp-00-80cfed1d4b1a/example-docker-repo/sample-image:tag1
student_01_b76a55fe2a8b@cloudshell:~ (qwiklabs-gcp-00-80cfed1d4b1a)$
```