Awwvision: Cloud Vision API from a Kubernetes Cluster

GSP066

Overview

The Awwvision lab uses Kubernetes and Cloud Vision API to demonstrate how to use the Vision API to classify (label) images from Reddit's /r/aww subreddit and display the labelled results in a web app.
Awwvision has three components:

1. A simple Redis instance.
2. A web app that displays the labels and associated images.

3. A worker that handles scraping Reddit for images and classifying them using the Vision API. Cloud Pub/Sub is used to coordinate tasks between multiple worker instances.

Setup and Requirements

Qwiklabs setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

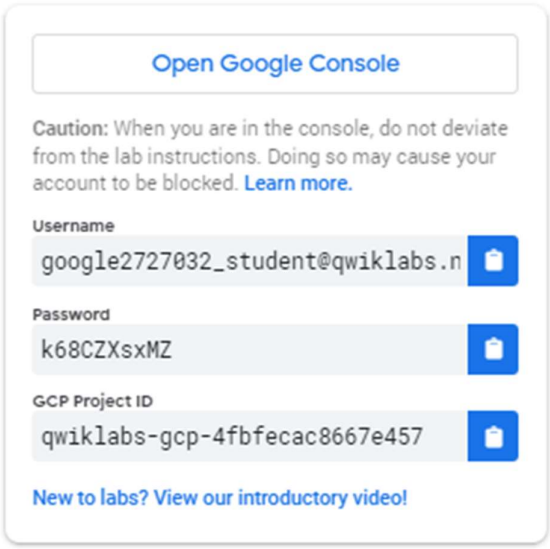
To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.
  Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

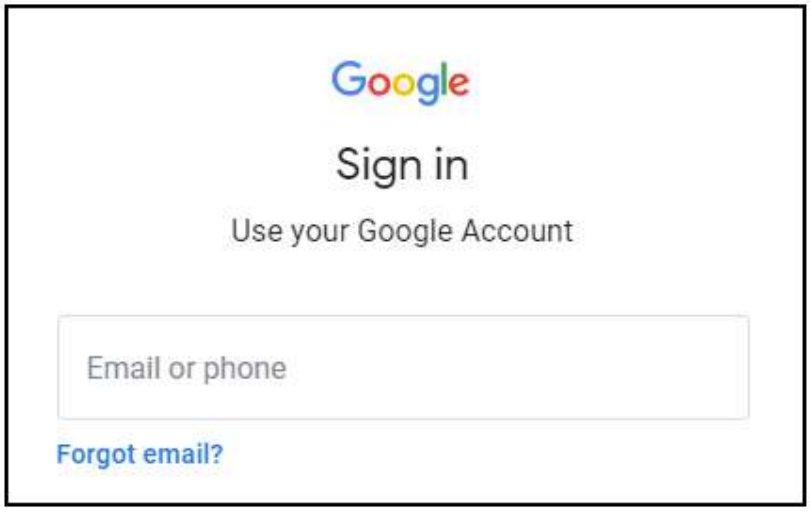
Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.

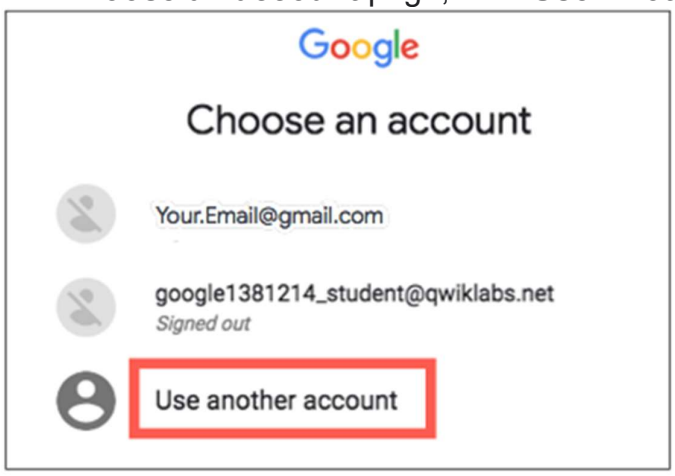


2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another Account**.



3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

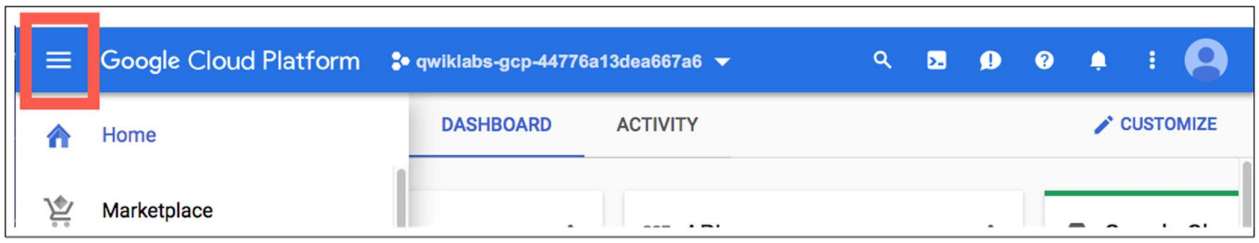
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. Click through the subsequent pages:

   - Accept the terms and conditions.
   - Do not add recovery options or two-factor authentication (because this is a temporary account).
   - Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

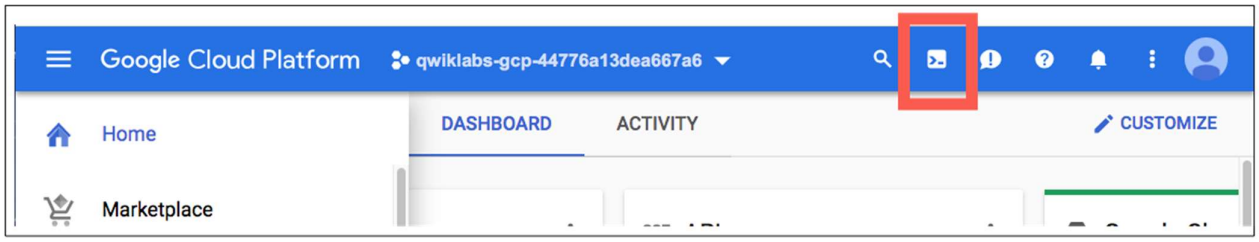
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



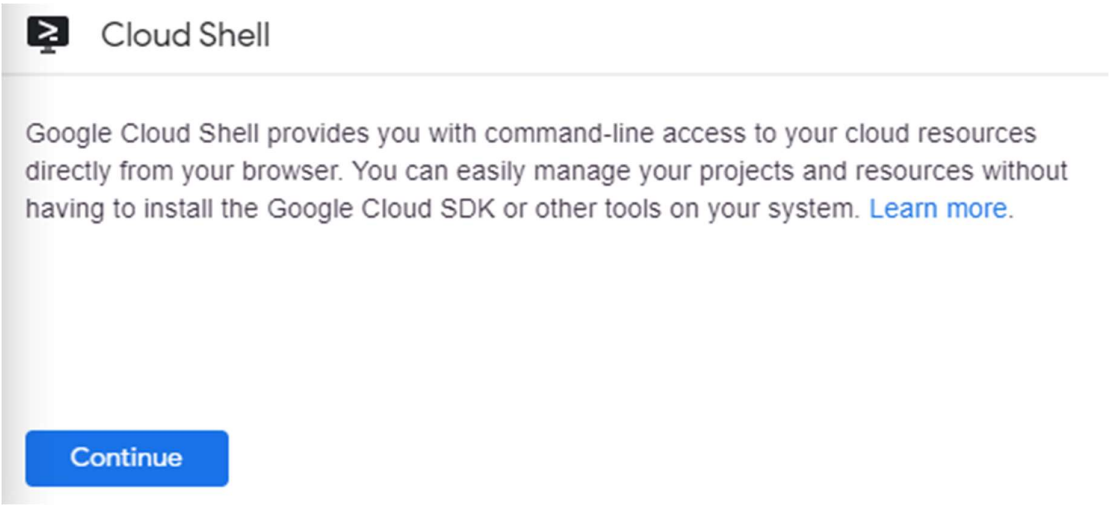
Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

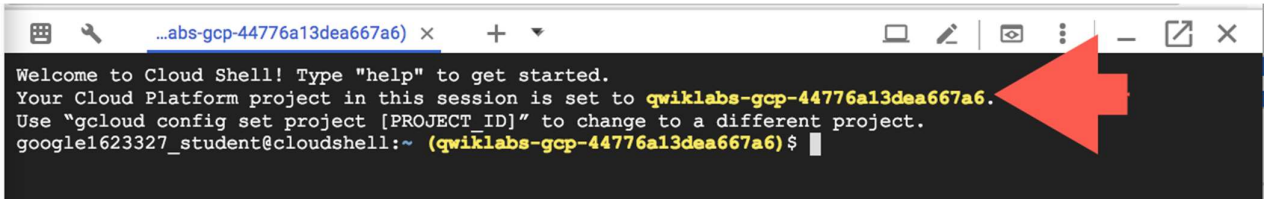
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

```
gcloud auth list
```

(Output)

```
Credentialed accounts:
 - <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]
project = <project_ID>
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the gcloud command-line tool overview.

Create a Kubernetes Engine cluster

In this lab you will use gcloud, Google Cloud's command-line tool, to set up a Kubernetes Engine cluster. You can specify as many nodes as you want, but you need at least one. The cloud platform scope is used to allow access to the Pub/Sub and Vision APIs.
In Cloud Shell, run the following to create a cluster in the `us-central1-a` zone:

```
gcloud config set compute/zone us-central1-a
```

Then start up the cluster by running:

```
gcloud container clusters create awwvision \
    --num-nodes 2 \
    --scopes cloud-platform
```

Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully created a Kubernetes cluster, you will see an assessment score.

Run the following to use the container's credentials:

```
gcloud container clusters get-credentials awwvision
```

Verify that everything is working using the `kubectl` command-line tool:

```
kubectl cluster-info
```

Create a virtual environment

Execute the following command to download and update the packages list.

```
sudo apt-get update
```

Python virtual environments are used to isolate package installation from the system.

```
sudo apt-get install virtualenv
```

If prompted [Y/n], press `Y` and then `Enter`.

```
virtualenv -p python3 venv
```

Activate the virtual environment.

```
source venv/bin/activate
```

Get the Sample

Now add sample data to your project by running:

```
gsutil -m cp -r gs://spls/gsp066/cloud-vision .
```

Deploy the sample

In Cloud Shell, change to the `python/awwvision` directory in the cloned cloud-vision repo:

```
cd cloud-vision/python/awwvision
```

Once in the `awwvision` directory, run `make all` to build and deploy everything:

```
make all
```

As part of the process, Docker images will be built and uploaded to the Google Container Registry private container registry. In addition, `yaml` files will be generated from templates, filled in with information specific to your project, and used to deploy the `redis`, `webapp`, and `worker` Kubernetes resources for the lab.

Check the Kubernetes resources on the cluster

After you've deployed, check that the Kubernetes resources are up and running.

First, list the pods by running:

```
kubectl get pods
```

You should see something like the following, though your pod names will be different. Make sure all of your pods have a Running before executing the next command.

```
NAME                     READY     STATUS    RESTARTS   AGE
awwvision-webapp-vwmr1   1/1       Running   0          1m
awwvision-worker-oz6xn   1/1       Running   0          1m
awwvision-worker-qc0b0   1/1       Running   0          1m
awwvision-worker-xpe53   1/1       Running   0          1m
redis-master-rpap8       1/1       Running   0          2m
```

Next, list the deployments by running:

```
kubectl get deployments -o wide
```

You can see the number of replicas specified for each, and the images used.

```
NAME               READY   UP-TO-DATE   AVAILABLE   AGE       CONTAINERS          IMAGES
SELECTOR
awwvision-webapp   1/1     1            1           1m        awwvision-webapp
gcr.io/your-project/awwvision-webapp   app=awwvision
awwvision-worker   3/3     3            3           1m        awwvision-worker
gcr.io/your-project/awwvision-worker   app=awwvision
redis-master       1/1     1            1           1m        redis-master        redis
app=redis
```

Once deployed, get the external IP address of the webapp service by running:

```
kubectl get svc awwvision-webapp
```

It may take a few minutes for the assigned external IP to be listed in the output. You should see something like the following, though your IPs will be different.

```
NAME               TYPE           CLUSTER_IP      EXTERNAL_IP    PORT(S)          AGE
awwvision-webapp   LoadBalancer   10.163.250.49   23.236.61.91   80:31925/TCP     13m
```

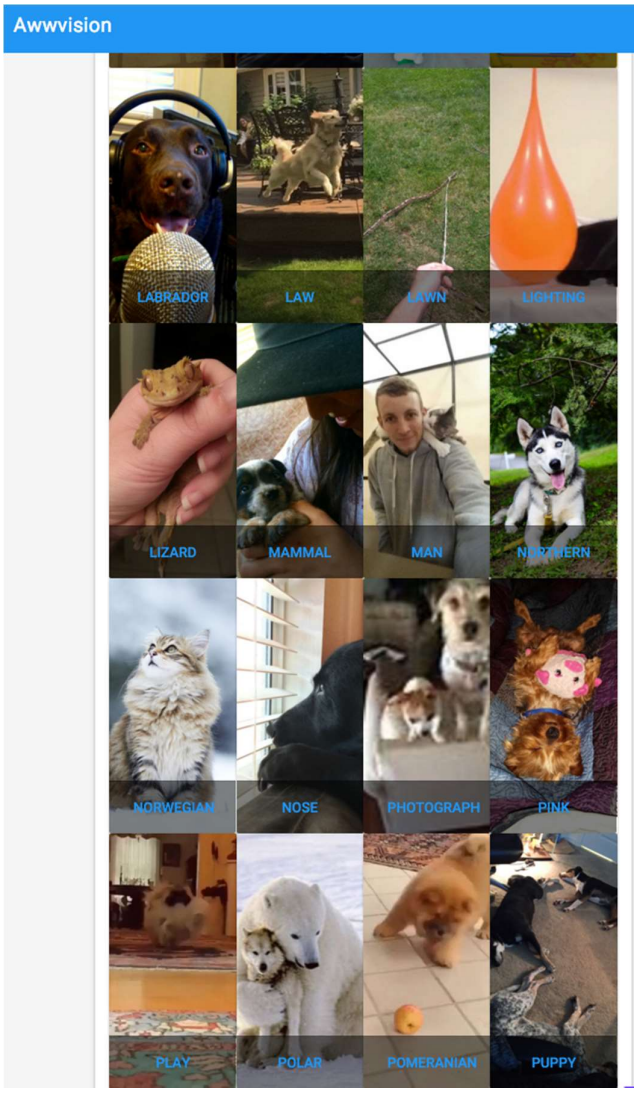
Test Completed Task

Click **Check my progress** to verify your performed task. If you have successfully deployed the sample app, you will see an assessment score.

Visit your new web app and start its crawler

Copy and paste the external IP of the `awwvision-webapp` service into a new browser to open the webapp, then click **Start the Crawler** button.

Next, click **go back** and you should start to see images from the /r/aww subreddit classified by the labels provided by the Vision API. You will see some of the images classified multiple times, when multiple labels are detected for them. (You can reload in a bit, in case you brought up the page before the crawler was finished).
Your results will look something like this:



Test your Understanding

Below are a multiple choice questions to reinforce your understanding of this lab's concepts. Answer them to the best of your abilities.

____ allows developers to easily integrate vision detection features within applications, including image labeling, face and landmark detection and much more.
Cloud Vision API

Congratulations

Finish Your Quest

This self-paced lab is part of the Qwiklabs Machine Learning APIs, Kubernetes Solutions, and Advanced ML: ML Infrastructure Quests. A Quest is a series of related labs that form a learning path. Completing a Quest earns you a badge to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in one of the above Quests and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Take your next lab

Try out another lab on Machine Learning APIs, like Running Dedicated Game Servers in Google Kubernetes Engine or Distributed Load Testing using Kubernetes.

Next steps

- Sign up for the full Coursera Course on Machine Learning

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated January 8, 2021

Lab Last Tested January 8, 2021