

[Start Lab](#)

01:07:30

Awwvision: Cloud Vision API from a Kubernetes Cluster

Lab 45 minutes No cost Intermediate



GSP066



Google Cloud Self-Paced Labs

[Lab instructions and tasks](#)

GSP066

Overview

Objectives

Setup and requirements

Task 1. Create a Kubernetes Engine cluster



Task 2. Create a virtual environment



Task 3. Get the sample



Task 4. Deploy the sample



Task 5. Check the Kubernetes resources on the cluster



Task 6. Visit your new webapp and start its crawler



Task 7. Test your understanding



Congratulations!



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.

Objectives

In this lab, you will learn how to perform the following tasks:

- Create a Kubernetes Engine cluster
- Deploy the sample awwvision app that uses the Vision API
- Visit your new webapp and start its crawler.

Setup and requirements

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 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.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

- Time to complete the lab---remember, once you start, you cannot pause a lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

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 the **Lab Details** panel with the following:

- The **Open Google Cloud console** button
- Time remaining
- The temporary credentials that you must use for this lab
- Other information, if needed, to step through this lab

2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

The lab spins up resources, and then opens another tab that shows the **Sign in** page.

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

Note: If you see the **Choose an account** dialog, click **Use Another Account**.

3. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

"Username"



You can also find the **Username** in the **Lab Details** panel.

4. Click **Next**.

5. Copy the **Password** below and paste it into the **Welcome** dialog.

"Password"



You can also find the **Password** in the **Lab Details** panel.

6. Click **Next**.

Important: You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.

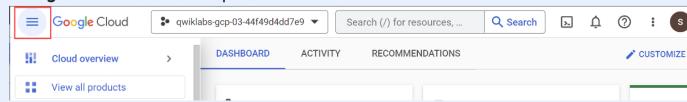
Note: Using your own Google Cloud account for this lab may incur extra charges.

7. 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 Google Cloud console opens in this tab.

Note: To view a menu with a list of Google Cloud products and services, click 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.

1. Click **Activate Cloud Shell**  at the top of the Google Cloud console.

When you are connected, you are already authenticated, and the project is set to your **Project_ID**, **PROJECT_ID**. The output contains a line that declares the **Project_ID** for this session:

Your Cloud Platform project in this session is set to "PROJECT_ID"

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

2. (Optional) You can list the active account name with this command:

```
gcloud auth list
```



3. Click **Authorize**.

Output:

```
ACTIVE: *
ACCOUNT: "ACCOUNT"

To set the active account, run:
$ gcloud config set account `ACCOUNT`
```

4. (Optional) You can list the project ID with this command:

```
gcloud config list project
```



Output:

```
[core]
project = "PROJECT_ID"
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to [the gcloud CLI overview guide](#).

Task 1. 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.

1. In Cloud Shell, run the following to create a cluster in the `Zone` zone:

```
gcloud config set compute/zone Zone
```



Note: For full documentation of `gcloud`, in Google Cloud, refer to [the gcloud CLI overview guide](#).

Task 1. 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.

1. In Cloud Shell, run the following to create a cluster in the `Zone` zone:

```
gcloud config set compute/zone Zone
```



Note: For full documentation of `gcloud`, in Google Cloud, refer to [the gcloud CLI overview guide](#).

Task 1. Create a Kubernetes Engine cluster

In this lab you will use `gcloud`, Google Cloud's command-line tool, to set up a

© 2020 Google LLC. All rights reserved. Google and the Google logo are trademarks of Google LLC. Other products and services may be trademarks of their respective owners.

[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.

1. In Cloud Shell, run the following to create a cluster in the `Zone` zone:

```
gcloud config set compute/zone Zone
```



```
python -m venv venv
```



3. Activate the virtual environment.

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



Task 3. Get the sample

- Add sample data to your project by running:

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



Task 4. Deploy the sample

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

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



2. 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.

Task 5. Check the Kubernetes resources on the cluster

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

1. 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

2. 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
awwvision-webapp	1/1	1	1	1m	awwvision-webapp
awwvision-worker	3/3	3	3	1m	awwvision-worker
redis-master	1/1	1	1	1m	redis-master

3. 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)
awwvision-webapp	LoadBalancer	10.163.250.49	23.236.61.91	80:31925/TCP

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.



Deploy the sample

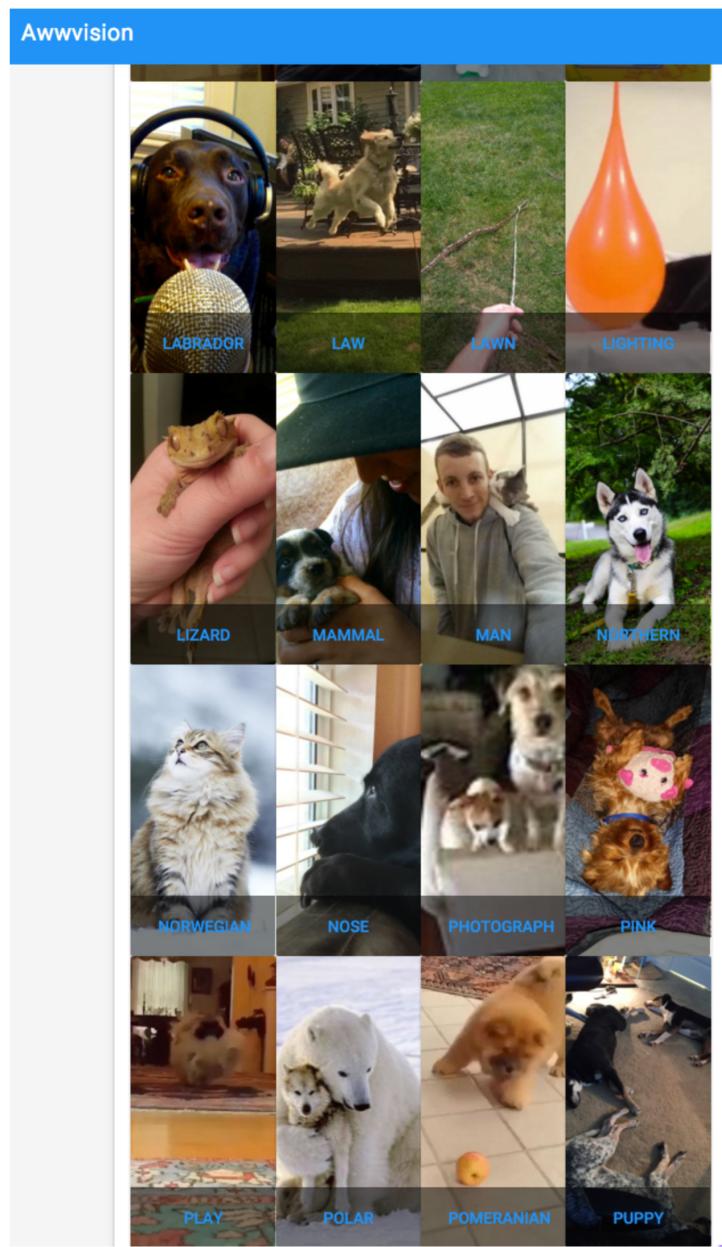
Check my progress

Task 6. Visit your new webapp and start its crawler

1. 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.

2. 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:



Task 7. Test your understanding

Below is a multiple choice question to reinforce your understanding of this lab's concepts. Answer it 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 ML
- Cloud Vision API
- Compute

[Submit](#)

Congratulations!

You used Kubernetes and Cloud Vision API to classify images from Reddit's /r/aww subreddit and displayed the results in a web app.

Google Cloud training and 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 February 14, 2024

Lab Last Tested October 09, 2023

Copyright 2024 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.