Distributed Image Processing in Cloud Dataproc

GSP010

Google Cloud Self-Paced Labs

Overview

In this hands-on lab, you will learn how to use Apache Spark on Cloud Dataproc to distribute a computationally intensive image processing task onto a cluster of machines. This lab is part of a series of labs on processing scientific data.

What you'll learn

- How to create a managed Cloud Dataproc cluster with Apache Spark pre-installed.
- How to build and run jobs that use external packages that aren't already installed on your cluster.
- How to shut down your cluster.

Prerequisites

This is an **advanced level** lab. Familiarity with Cloud Dataproc and Apache Spark is recommended, but not required. If you're looking to get up to speed in these services, be sure to check out the following labs:

- Dataproc: Qwik Start - Command Line
- Dataproc: Qwik Start - Console
- Introduction to Cloud Dataproc: Hadoop and Spark on Google Cloud

Once you're ready, scroll down to learn more about the services that you'll be using in this lab.

Introduction

Cloud Dataproc is a managed Spark and Hadoop service that lets you take advantage of open source data tools for batch processing, querying, streaming, and machine learning. Cloud Dataproc automation helps you create clusters quickly, manage them easily, and save money by turning clusters off when you don't need them. With less time and money spent on administration, you can focus on your jobs and your data.

Consider using Cloud Dataproc to scale out compute-intensive jobs that meet these characteristics:

1. The job is **embarrassingly parallel** —in other words, you can process different subsets of the data on different machines.
2. You already have **Apache Spark** code that does the computation or you are familiar with Apache Spark.
3. The distribution of the work is pretty **uniform** across your data subsets.

If different subsets require different amounts of processing (or if you don't already know Apache Spark), Apache Beam on Cloud Dataflow is a compelling alternative because it provides autoscaling data pipelines.

In this lab, the job that you will run outlines the faces in the image using a set of image processing rules specified in OpenCV. The Vision API is a better way to do this, since these sort of hand-coded rules don't work all that well, but this lab is an example of doing a compute-intensive job in a distributed way.

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:

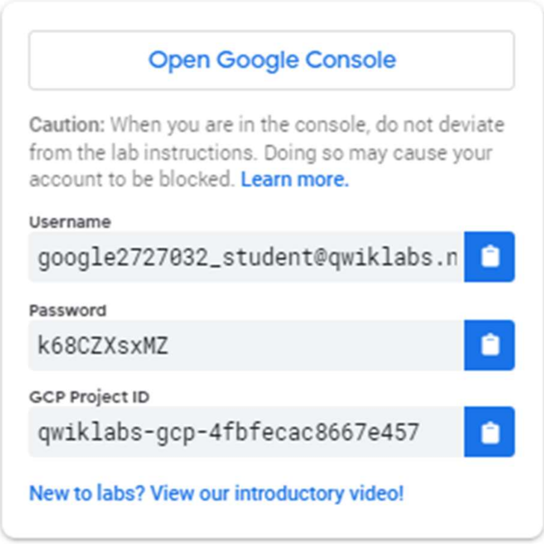
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

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.



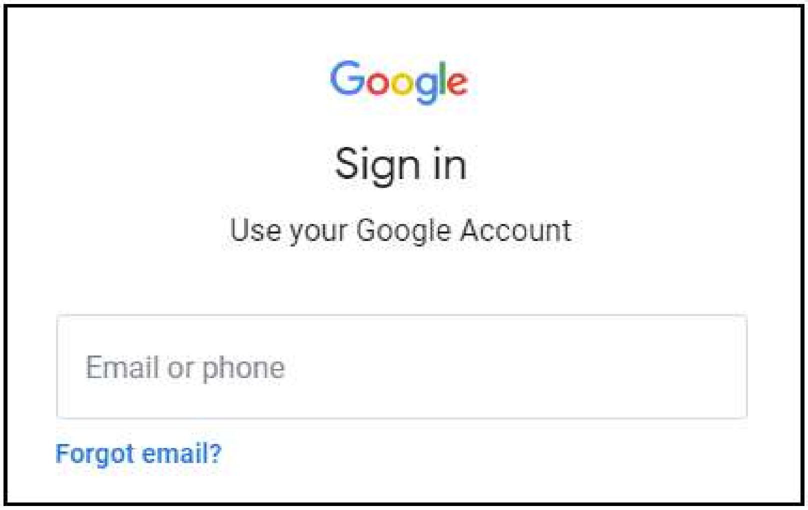
Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Username: google2727032_student@qwiklabs.n

Password: k68CZXsxMZ

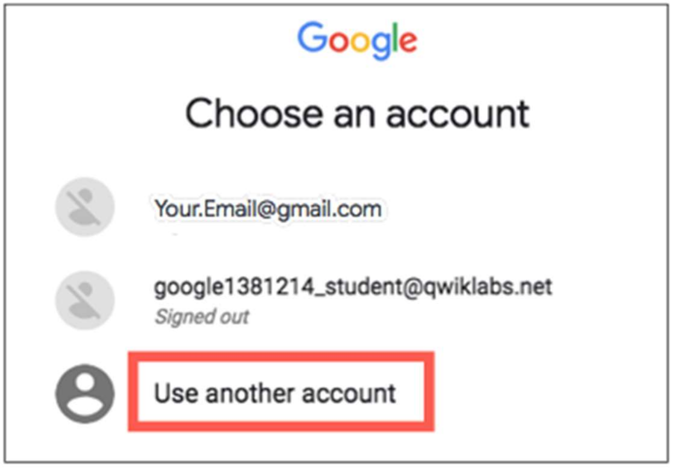
GCP Project ID: qwiklabs-gcp-4fbfecac8667e457

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.

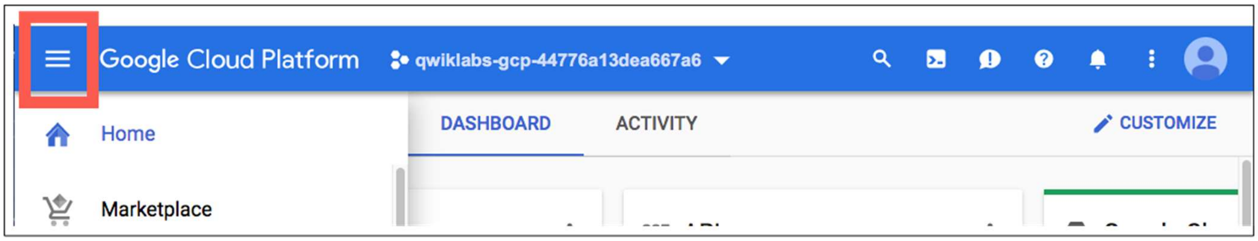
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.

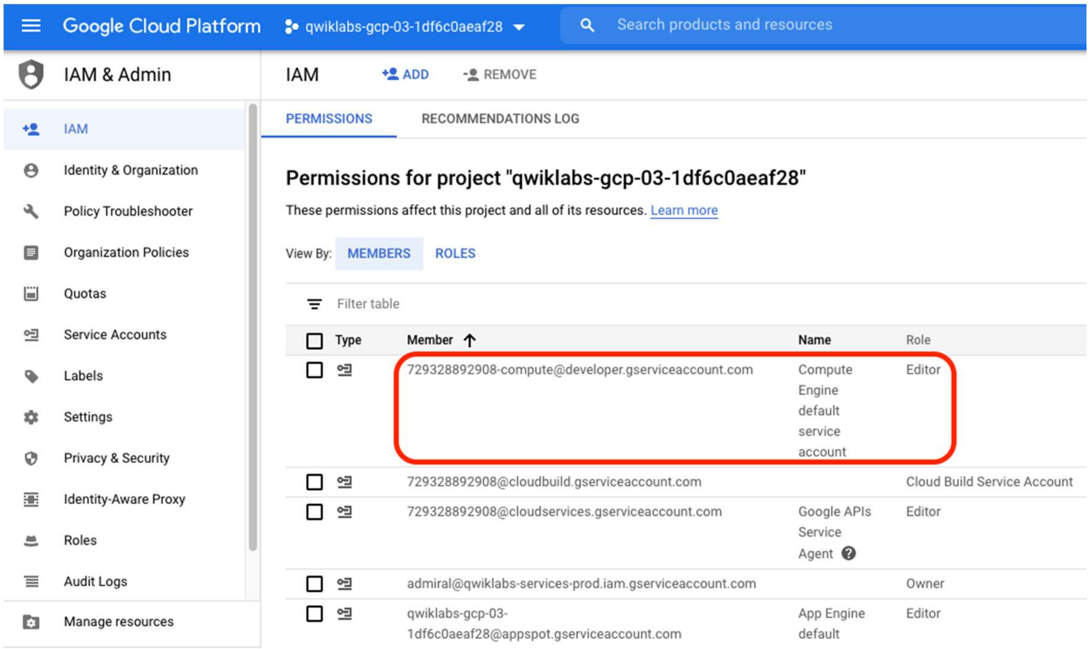


Check project permissions

Before you begin your work on Google Cloud, you need to ensure that your project has the correct permissions within Identity and Access Management (IAM).

1. In the Google Cloud console, on the **Navigation menu** (☰), click **IAM & Admin** > **IAM**.

2. Confirm that the default compute Service Account `{project-number}-compute@developer.gserviceaccount.com` is present and has the `editor` role assigned. The account prefix is the project number, which you can find on **Navigation menu** > **Home**.



Permissions for project "qwiklabs-gcp-03-1df6c0aeaf28"

These permissions affect this project and all of its resources. Learn more

□	Type	Member ↑	Name	Role
□		729328892908-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor
□		729328892908@cloudbuild.gserviceaccount.com		Cloud Build Service Account
□		729328892908@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor
□		admiral@qwiklabs-services-prod.iam.gserviceaccount.com		Owner
□		qwiklabs-gcp-03-1df6c0aeaf28@appspot.gserviceaccount.com	App Engine default	Editor

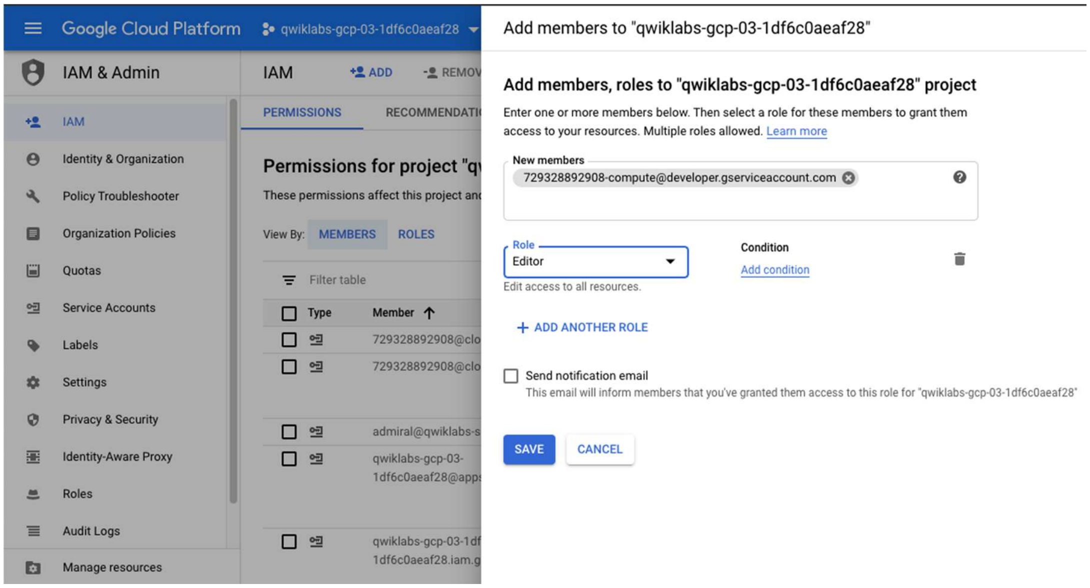
If the account is not present in IAM or does not have the `editor` role, follow the steps below to assign the required role.

- In the Google Cloud console, on the **Navigation menu**, click **Home**.
- Copy the project number (e.g. `729328892908`).
- On the **Navigation menu**, click **IAM & Admin** > **IAM**.
- At the top of the **IAM** page, click **Add**.
- For **New members**, type:

```
{project-number}-compute@developer.gserviceaccount.com
```

Replace `{project-number}` with your project number.

- For **Role**, select **Project** (or Basic) > **Editor**. Click **Save**.



Add members to "qwiklabs-gcp-03-1df6c0aeaf28"

Add members, roles to "qwiklabs-gcp-03-1df6c0aeaf28" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. Learn more

New members: 729328892908-compute@developer.gserviceaccount.com

Role: Editor

Edit access to all resources.

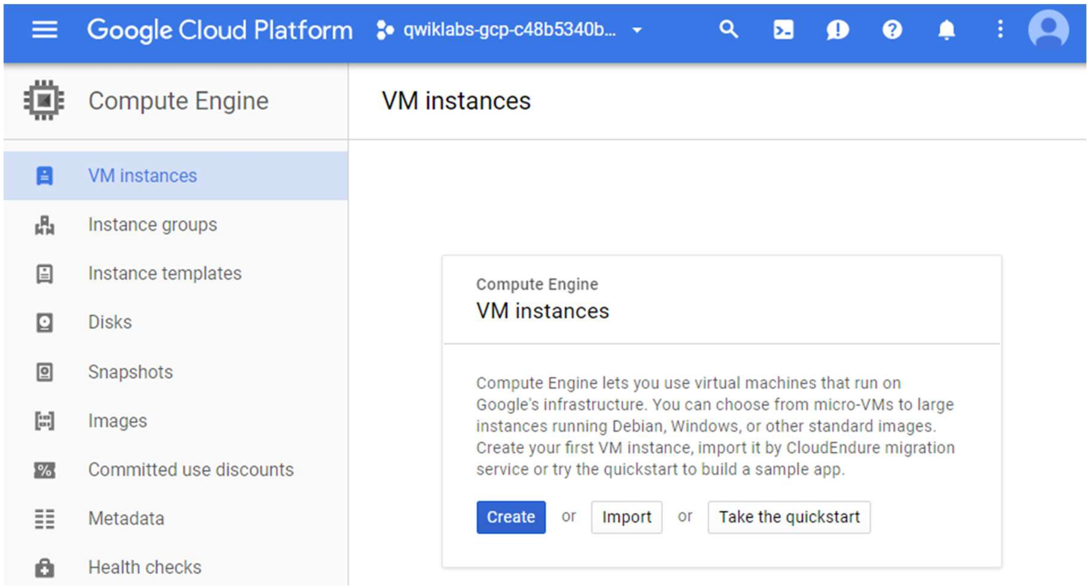
Condition: Add condition

□ Send notification email

This email will inform members that you've granted them access to this role for "qwiklabs-gcp-03-1df6c0aeaf28"

Create a development machine in Compute Engine

First, you create a virtual machine to host your services. In the Cloud Console, go to **Compute Engine** > **VM Instances** > **Create**.



Compute Engine lets you use virtual machines that run on Google's infrastructure. You can choose from micro-VMs to large instances running Debian, Windows, or other standard images. Create your first VM instance, import it by CloudEndure migration service or try the quickstart to build a sample app.

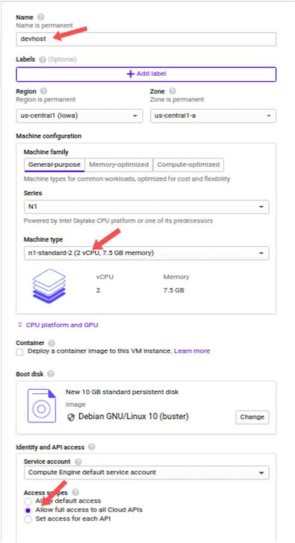
Configure the following fields, leave the others at their default value.

Name: devhost

Series: N1

Machine Type: 2 vCPUs (n1-standard-2 instance)

Identity and API Access: Allow full access to all Cloud APIs.



Name
Name is permanent
devhost

Labels (Optional)

Region
Region is permanent
us-central1 (Iowa)

Zone
Zone is permanent
us-central1-a

Machine configuration

Machine family
General-purpose | Memory-optimized | Compute-optimized
Machine types for common workloads, optimized for cost and flexibility

Series
N1
Powered by Intel Skylake CPU platform or one of its predecessors

Machine type
n1-standard-2 (2 vCPU, 7.5 GB memory)

vCPU	Memory
2	7.5 GB

Container
□ Deploy a container image to this VM instance. Learn more

Boot disk
New 10 GB standard persistent disk
Image
Debian GNU/Linux 10 (buster)

Identity and API access

Service account
Compute Engine default service account

Access scopes
○ Allow default access
● Allow full access to all Cloud APIs
○ Set access for each API

Click **Create**. This will serve as your development ‘bastion' host.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will granted with an assessment score.

Create a development machine in Compute Engine.

Check my progress

Now SSH into the instance by clicking the **SSH** button on the Console.

Install Software

Now set up the software to run the job. Using `sbt`, an open source build tool, you'll build the JAR for the job you'll submit to the Cloud Dataproc cluster. This JAR will contain the program and the required packages necessary to run the job. The job will detect faces in a set of image files stored in a Cloud Storage bucket, and write out image files with the faces outlined, to either the same or to another Cloud Storage bucket.

Step 1: Set up Scala and sbt

In the SSH window, install `Scala` and `sbt` with the following commands so that you can compile the code:

```
sudo apt-get install -y dirmngr unzip

sudo apt-get update

sudo apt-get install -y apt-transport-https

echo "deb https://dl.bintray.com/sbt/debian /" | \
sudo tee -a /etc/apt/sources.list.d/sbt.list

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 642AC823
sudo apt-get update
sudo apt-get install -y bc scala sbt
```

Step 2: Set up the Feature Detector Files

Now you'll build the Feature Detector files. The code for this lab is a slight modification of a solution which exists in the Cloud Dataproc repository on GitHub. You'll download the code, then `cd` into the directory for this lab and build a "fat JAR" of the feature detector so that it can be submitted to Cloud Dataproc. Run the following commands **in the SSH window**:

```
sudo apt-get update
gsutil cp gs://spls/gsp124/cloud-dataproc.zip .
unzip cloud-dataproc.zip
cd cloud-dataproc/codelabs/opencv-haarcascade
```

Step 3: Launch build

This command builds a "fat JAR" of the Feature Detector so that it can be submitted to Cloud Dataproc:

```
sbt assembly
```

Note: This step will take a while to process, approximately five or more minutes. Please be patient.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will granted with an assessment score.

Install Software in the development machine.

Check my progress

Create a Cloud Storage bucket and collect images

Now that you built your Feature Detector files, create a Cloud Storage bucket and add some sample images to it.

Step 1

Fetch the Project ID to use to name your bucket.

```
GCP_PROJECT=$(gcloud config get-value core/project)
```

Name your bucket and set a shell variable to your bucket name. The shell variable will be used in commands to refer to your bucket.

```
MYBUCKET="${USER//google}-image-${RANDOM}"

echo MYBUCKET=${MYBUCKET}
```

Step 2

Use the `gsutil` program, which comes with `gcloud` in the Cloud SDK, to create the bucket to hold your sample images:

```
gsutil mb gs://${MYBUCKET}
```

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will granted with an assessment score.

Create a Cloud Storage bucket.

Check my progress

Step 3

Download some sample images into your bucket:

```
curl https://www.publicdomainpictures.net/pictures/20000/velka/family-of-three-871290963799xUk.jpg | gsutil cp - gs://${MYBUCKET}/imgs/family-of-three.jpg
curl https://www.publicdomainpictures.net/pictures/10000/velka/african-woman-331287912508yqXc.jpg | gsutil cp - gs://${MYBUCKET}/imgs/african-woman.jpg
curl https://www.publicdomainpictures.net/pictures/10000/velka/296-1246658839vCW7.jpg | gsutil cp - gs://${MYBUCKET}/imgs/classroom.jpg
```

You just downloaded the following images into your Cloud Storage bucket:

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will granted with an assessment score.

Download some sample images into your bucket.

Check my progress

Step 4

Run this to see the contents of your bucket:

```
gsutil ls -R gs://${MYBUCKET}
```

Output, do not copy

```
gs://gcpstaging20392-student-image-23218/imgs/:
gs://gcpstaging20392-student-image-23218/imgs/african-woman.jpg
gs://gcpstaging20392-student-image-23218/imgs/classroom.jpg
gs://gcpstaging20392-student-image-23218/imgs/family-of-three.jpg
```

Create a Cloud Dataproc cluster

Step 1

Run the following commands **in the SSH window** to name your cluster and to set the `MYCLUSTER` variable. You'll be using the variable in commands to refer to your cluster:

```
MYCLUSTER="${USER/_/-}-qwiklab"
echo MYCLUSTER=${MYCLUSTER}
```

Step 2

Set a global Compute Engine region to use and create a new cluster:

```
gcloud config set dataproc/region us-central1
gcloud dataproc clusters create ${MYCLUSTER} --bucket=${MYBUCKET} --worker-machine-
type=n1-standard-2 --master-machine-type=n1-standard-2 --initialization-
actions=gs://spls/gsp010/install-libgtk.sh --image-version=2.0
```

If prompted to use a zone instead of a region, enter **Y**.

This might take a couple minutes. The default cluster settings, which include two worker nodes, should be sufficient for this lab. `n1-standard-2` is specified as both the worker and master machine type to reduce the overall number of cores used by the cluster.

For the `initialization-actions` flag, you are passing a script which installs the `libgtk2.0-dev` library on each of your cluster machines. This library will be necessary to run the code.

If your cluster fails to create, try deleting your cluster (`gcloud dataproc clusters delete ${MYCLUSTER}`) and then retrying the previous cluster creation command.

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will granted with an assessment score.

Create a Cloud Dataproc cluster.

Check my progress

Note: See the Cloud SDK gcloud dataproc clusters create command for information on using command line flags to customize cluster settings.

Submit your job to Cloud Dataproc

In this lab the program you're running is used as a face detector, so the inputted `haar` classifier must describe a face. A `haar` classifier is an XML file that is used to describe features that the program will detect. You will download the haar classifier file and include its Cloud Storage path in the first argument when you submit your job to your Cloud Dataproc cluster.

Step 1

Run the following command **in the SSH window** to load the face detection configuration file into your bucket:

```
curl
https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_fr
ontalface_default.xml | gsutil cp -
gs://${MYBUCKET}/haarcascade_frontalface_default.xml
```

Step 2

Use the set of images you uploaded into the `imgs` directory in your Cloud Storage bucket as input to your Feature Detector. You must include the path to that directory as the second argument of your job-submission command.

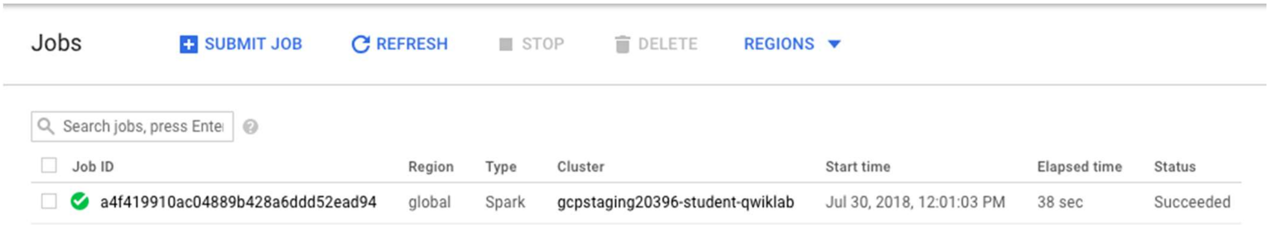
Submit your job to Cloud Dataproc:

```
cd ~/cloud-dataproc/codelabs/opencv-haarcascade
gcloud dataproc jobs submit spark \
--cluster ${MYCLUSTER} \
--jar target/scala-2.12/feature_detector-assembly-1.0.jar -- \
gs://${MYBUCKET}/haarcascade_frontalface_default.xml \
gs://${MYBUCKET}/imgs/ \
gs://${MYBUCKET}/out/
```

You can add any other images to use to the Cloud Storage bucket specified in the second argument.

Step 3

Monitor the job, in the Console go to **Navigation menu** > **Dataproc** > **Jobs**. Move on to the next step when you get a similar output:



□	Job ID	Region	Type	Cluster	Start time	Elapsed time	Status
□	✓ a4f419910ac04889b428a6ddd52ead94	global	Spark	gcpstaging20396-student-qwiklab	Jul 30, 2018, 12:01:03 PM	38 sec	Succeeded

Test Completed Task

Click **Check my progress** to verify your performed task. If you have completed the task successfully you will granted with an assessment score.

Submit your job to Cloud Dataproc.

Check my progress

Step 4

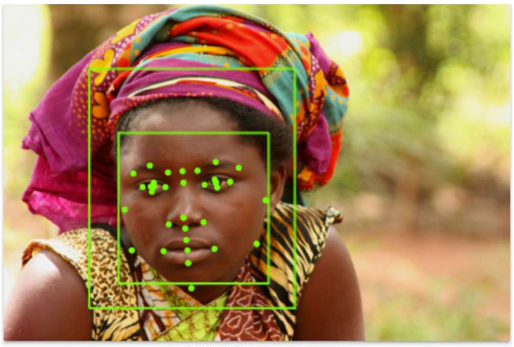
When the job is complete, go to **Navigation menu** > **Storage** and find the bucket you created (it will have your username followed by `student-image` followed by a random number) and click on it. Then click on an image in the **Out** directory. Click on **Download** icon, the image will download to your computer.

How accurate is the face detection? The Vision API is a better way to do this, since this sort of hand-coded rules don't work all that well. You can see how it works next.

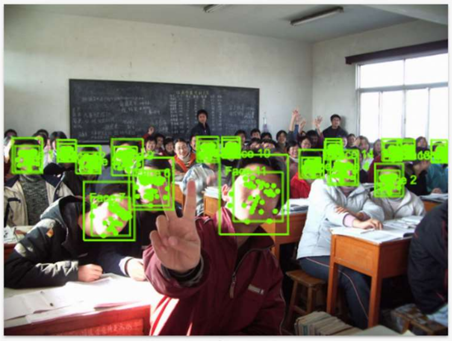
Step 5 (Optional)

In your bucket go to the `imgs` folder and click on the other images you uploaded to your bucket. This will download the three sample images. Save them to your computer.

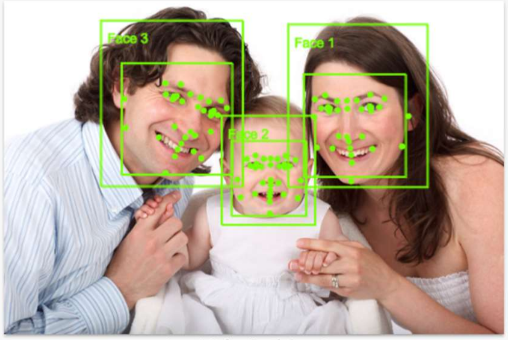
Click on this link to go to the Vision API page, scroll down to the **Try the API** section and upload the images you downloaded from your bucket. You'll see the results of the image detection in seconds. The underlying machine learning models keep improving, so your results may not be the same:



imgs%2Fafrican-woman.jpg



imgs%2Fclassroom.jpg



imgs%2Ffamily-of-three.jpg

Step 6 (Optional)

If you want to experiment with improving the Feature Detector, you can make edits to the `FeatureDetector` code, then re-run `sbt assembly` and the `gcloud dataproc` and `jobs submit` commands.

Test your Understanding

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

Bucket names can be represented as an IP address in dotted-decimal notation.
close~~True~~
checkFalse
Cloud Dataproc is a fully-managed cloud service for running _____ cluster.
close~~Container~~
close~~Kubernetes~~
checkApache Spark
Submit
Bucket names cannot begin with the goog prefix.
checkTrue
close~~False~~

Congratulations!

You learned how to spin up a Cloud Dataproc cluster and run jobs!

Finish Your Quest

This self-paced lab is part of the Qwiklabs Quest Scientific Data Processing. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, 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 this Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Take Your Next Lab

Continue your Quest with Distributed Computation of NDVI from Landsat Images using Cloud Dataflow, or try one of these:

- Analyzing Natality Data Using Datalab and BigQuery
- Predicting Baby Weight with TensorFlow on Cloud ML Engine

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 March 12, 2021
Lab Last Tested March 12, 2020