**Set Up an App Dev Environment on Google Cloud: Challenge Lab**

experimentLabschedule1 houruniversal\_currency\_alt1 Creditshow\_chartIntroductory

**GSP315**



**Introduction**

In a challenge lab you’re given a scenario and a set of tasks. Instead of following step-by-step instructions, you will use the skills learned from the labs in the course to figure out how to complete the tasks on your own! An automated scoring system (shown on this page) will provide feedback on whether you have completed your tasks correctly.

When you take a challenge lab, you will not be taught new Google Cloud concepts. You are expected to extend your learned skills, like changing default values and reading and researching error messages to fix your own mistakes.

To score 100% you must successfully complete all tasks within the time period!

This lab is recommended for students who have enrolled in the [Set Up an App Dev Environment on Google Cloud](https://www.cloudskillsboost.google/course_templates/637) skill badge. Are you ready for the challenge?

**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 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.

**Challenge scenario**

You are just starting your junior cloud engineer role with Jooli inc. So far you have been helping teams create and manage Google Cloud resources.

You are expected to have the skills and knowledge for these tasks, so don’t expect step-by-step guides.

Your challenge

You are asked to help a newly formed development team with some of their initial work on a new project around storing and organizing photographs, called *Memories*. You have been asked to assist the Memories team with initial configuration for their application development environment.

You receive the following request to complete the following tasks:

* Create a bucket for storing the photographs.
* Create a Pub/Sub topic that will be used by a Cloud Function you create.
* Create a Cloud Function.
* Remove the previous cloud engineer’s access from the memories project.

Some Jooli Inc. standards you should follow:

* Create all resources in the REGION region and ZONE zone, unless otherwise directed.
* Use the project VPCs.
* Naming is normally *team-resource*, e.g. an instance could be named **kraken-webserver1**
* Allocate cost effective resource sizes. Projects are monitored and excessive resource use will result in the containing project's termination (and possibly yours), so beware. This is the guidance the monitoring team is willing to share; unless directed, use **e2-micro** for small Linux VMs and **e2-medium** for Windows or other applications such as Kubernetes nodes.

Each task is described in detail below, good luck!

**Task 1. Create a bucket**

You need to create a bucket called Bucket Name for the storage of the photographs. Ensure the resource is created in the REGION region and ZONE zone.

Click *Check my progress* to verify the objective.

Create a bucket called Bucket Name

Check my progress

**Task 2. Create a Pub/Sub topic**

Create a Pub/Sub topic called Topic Name for the Cloud Function to send messages.

Click *Check my progress* to verify the objective.

Create a Pub/Sub topic called Topic Name

Check my progress

**Task 3. Create the thumbnail Cloud Function**

Create a Cloud Function Cloud Function Name that will to create a thumbnail from an image added to the Bucket Name bucket. Ensure the Cloud Function is using the 2nd Generation environment. Ensure the resource is created in the REGION region and ZONE zone.

1. Create a Cloud Function called Cloud Function Name

**Note:** The Cloud Function is required to executes every time an object is created in the bucket created in Task 1. During the process Cloud Function may request permission to enable APIs. Please enable each of the required APIs as requested.

1. Make sure you set the **Entry point** (Function to execute) to Cloud Function Name and **Trigger** to Cloud Storage.
2. Add the following code to the index.js:

const functions = require('@google-cloud/functions-framework');

const crc32 = require("fast-crc32c");

const { Storage } = require('@google-cloud/storage');

const gcs = new Storage();

const { PubSub } = require('@google-cloud/pubsub');

const imagemagick = require("imagemagick-stream");

functions.cloudEvent('', cloudEvent => {

const event = cloudEvent.data;

console.log(`Event: ${event}`);

console.log(`Hello ${event.bucket}`);

const fileName = event.name;

const bucketName = event.bucket;

const size = "64x64"

const bucket = gcs.bucket(bucketName);

const topicName = "";

const pubsub = new PubSub();

if ( fileName.search("64x64\_thumbnail") == -1 ){

// doesn't have a thumbnail, get the filename extension

var filename\_split = fileName.split('.');

var filename\_ext = filename\_split[filename\_split.length - 1];

var filename\_without\_ext = fileName.substring(0, fileName.length - filename\_ext.length );

if (filename\_ext.toLowerCase() == 'png' || filename\_ext.toLowerCase() == 'jpg'){

// only support png and jpg at this point

console.log(`Processing Original: gs://${bucketName}/${fileName}`);

const gcsObject = bucket.file(fileName);

let newFilename = filename\_without\_ext + size + '\_thumbnail.' + filename\_ext;

let gcsNewObject = bucket.file(newFilename);

let srcStream = gcsObject.createReadStream();

let dstStream = gcsNewObject.createWriteStream();

let resize = imagemagick().resize(size).quality(90);

srcStream.pipe(resize).pipe(dstStream);

return new Promise((resolve, reject) => {

dstStream

.on("error", (err) => {

console.log(`Error: ${err}`);

reject(err);

})

.on("finish", () => {

console.log(`Success: ${fileName} → ${newFilename}`);

// set the content-type

gcsNewObject.setMetadata(

{

contentType: 'image/'+ filename\_ext.toLowerCase()

}, function(err, apiResponse) {});

pubsub

.topic(topicName)

.publisher()

.publish(Buffer.from(newFilename))

.then(messageId => {

console.log(`Message ${messageId} published.`);

})

.catch(err => {

console.error('ERROR:', err);

});

});

});

}

else {

console.log(`gs://${bucketName}/${fileName} is not an image I can handle`);

}

}

else {

console.log(`gs://${bucketName}/${fileName} already has a thumbnail`);

}

});

Copied!

content\_copy

1. Add the following code to the package.json:

{

"name": "thumbnails",

"version": "1.0.0",

"description": "Create Thumbnail of uploaded image",

"scripts": {

"start": "node index.js"

},

"dependencies": {

"@google-cloud/functions-framework": "^3.0.0",

"@google-cloud/pubsub": "^2.0.0",

"@google-cloud/storage": "^5.0.0",

"fast-crc32c": "1.0.4",

"imagemagick-stream": "4.1.1"

},

"devDependencies": {},

"engines": {

"node": ">=4.3.2"

}

}

Copied!

content\_copy

Verify the thumbnail was successfully created.

**Note:** If you get a permission denied error stating it may take a few minutes before all necessary permissions are propagated to the Service Agent, wait a few minutes and try again. Ensure you have the relevant roles (Eventarc Service Agent, Eventarc Event Receiver, Service Account Token Creator, and Pub/Sub Publisher) assigned to the correct service accounts.

Click *Check my progress* to verify the objective.

Verify the Cloud Function

Check my progress

**Task 4. Test the Infrastructure**

You must upload one JPG or PNG image into the bucket

1. Upload a PNG or JPG image to Bucket Name bucket.

**Note:** Alternatively Download this image https://storage.googleapis.com/cloud-training/gsp315/map.jpg to your machine. Then upload it to the bucket.

1. You will see a thumbnail image appear shortly afterwards (use **REFRESH** in the bucket details).

**Task 5. Remove the previous cloud engineer**

You will see that there are two users defined in the project.

* One is your account (Username 1 with the role of Owner).
* The other is the previous cloud engineer (Username 2 with the role of Viewer).

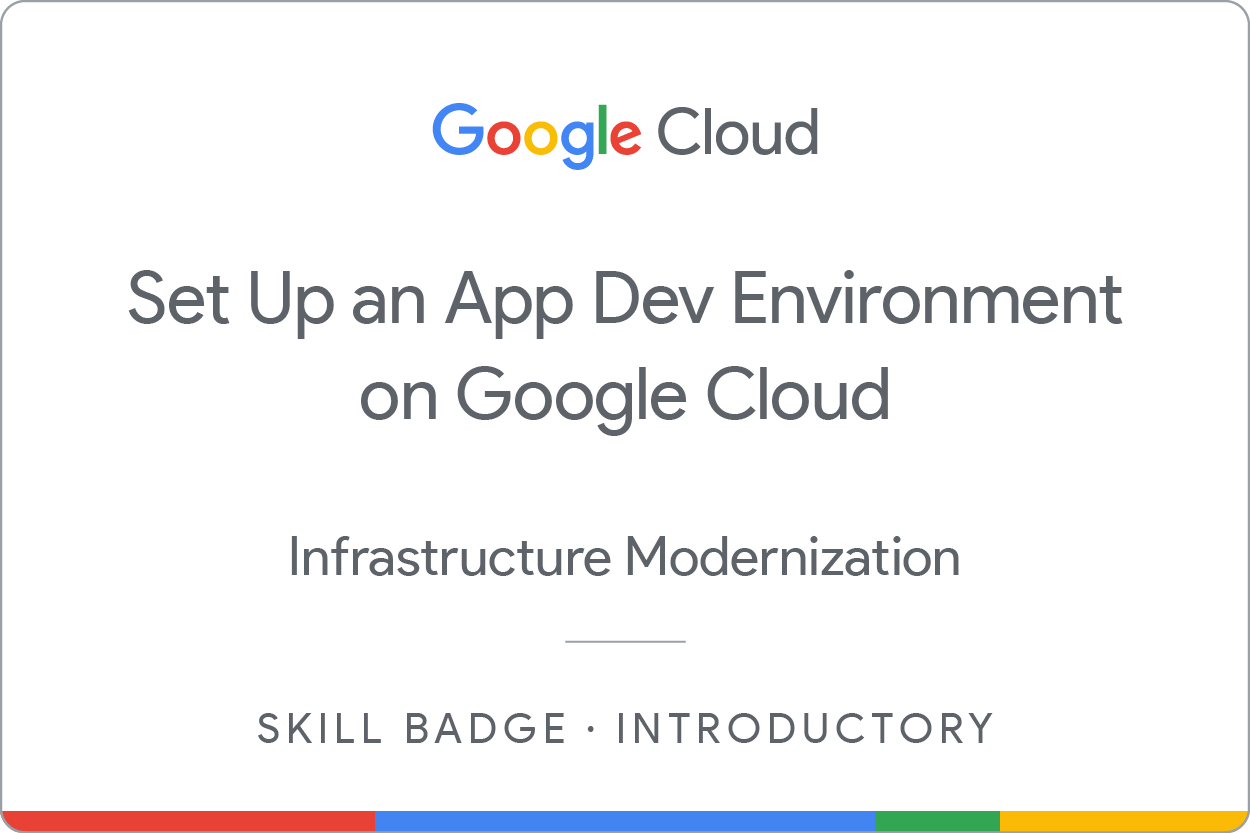
1. Remove the previous cloud engineer’s access from the project.

Click *Check my progress* to verify the objective.

Remove the previous cloud engineer

Check my progress

**Congratulations!**



Next steps / Learn more