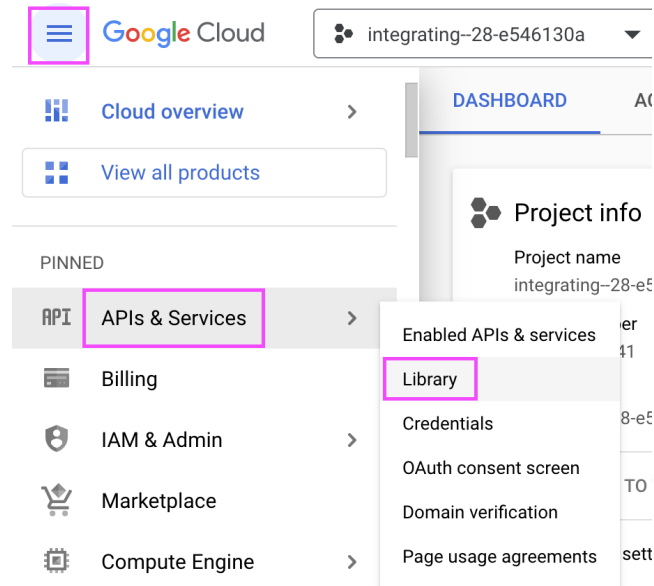1. Enable the necessary APIs for: Cloud Source Repositories API, Cloud Functions, Cloud Build
   a. "Navigation menu"; "APIs & Services", "Library"

   b. Type each service into the search bar individually, and click to "enable" each API
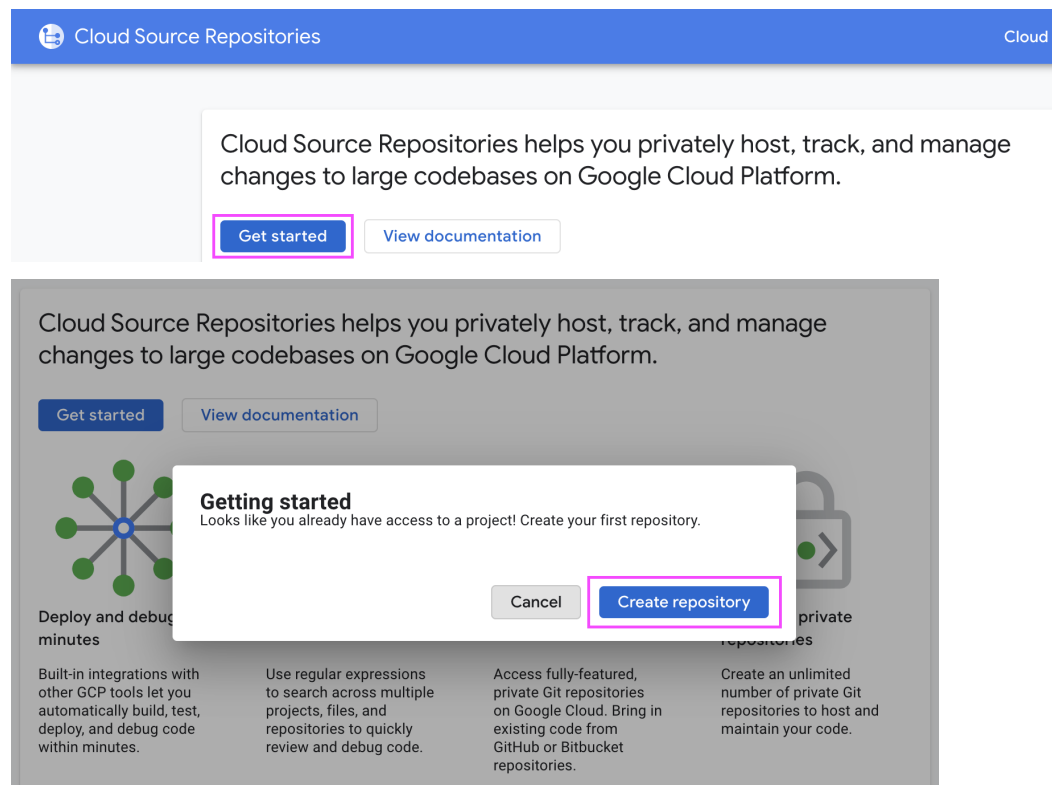

   "source repo"

2. Set up the source repository that will store our code
   a. "Navigation menu"; "Source Repositories" (opens in new tab)
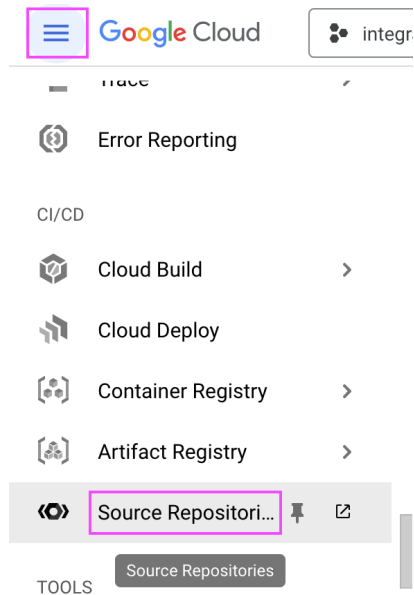
   b. "Get started"; "Create repository"

   c. "Create new repository"; "Continue"

Add a repository

Select one of the following options to continue:

◉ Create new repository
Choose this option to create an empty repository.

○ Connect external repository
Choose this option to mirror a repository from a hosted service, such as GitHub or Bitbucket.

Cancel   Continue

d. Assign "Repository name" and choose the "Project". Since we are integrating Cloud Functions and the repository, we will select the project starting with "integrating"; "Create"

Cloud Source Repositories

Create new repository

Repository name *
integrationFR   ?

Project *
Type or select a project ID   OR   Create project   ?

custom-images-251316 (custom-images)

custom-images-staging (custom-images-staging)

pricing.

integrating--28-e546130a (integrating--28-e546130a)

Cancel   Create

e. An alert will show stating "your repository is currently empty"
3. Populate the repository using Cloud Shell
   a. In our open Google Cloud tab, "Activate Cloud Shell", in the upper right hand corner; "Continue" upon startup

Google Cloud   integrating--28-e546130a ▼   Search for resources, docs, products, and...   🔍 Search   [>_]   ③

Activate Cloud Shell

Welcome

You're working in integrating--28-e546130a

Project number: 187237745841   Project ID: integrating--28-e546130a

Dashboard   Recommendations

➕ Create a VM   ➕ Run a query in BigQuery   ➕ Create a GKE cluster   ➕ Create a storage bucket

b. To use commands on this project, copy the project ID, located on the "Select a Project" page, and in Cloud Shell type the following command, followed by enter:
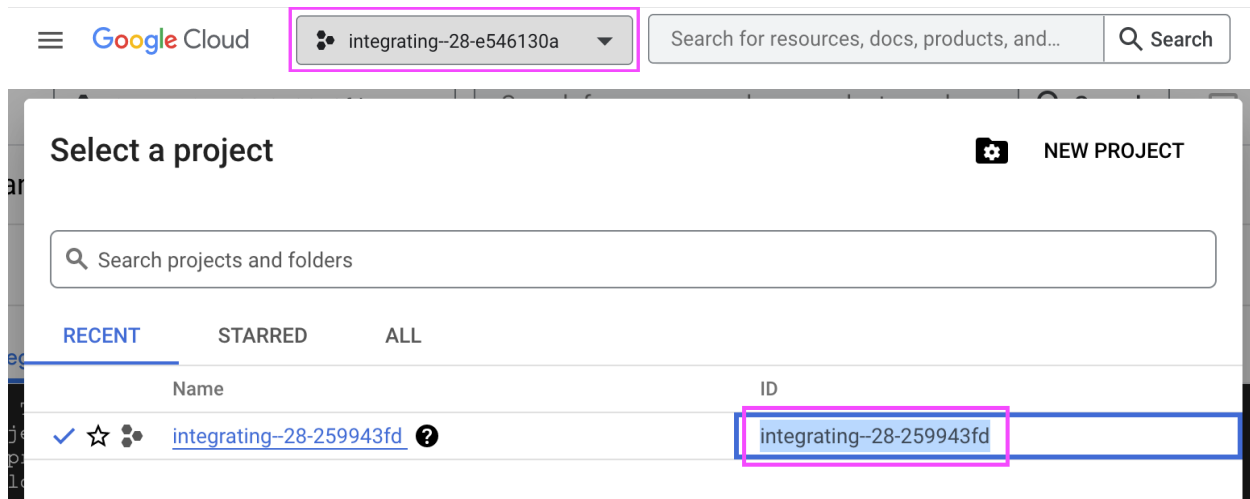
i. gcloud config set project integrating–28-259943fd

c. "Authorize"

d. Clone the repository we created into the Cloud Shell using the following command, followed by enter:

i. gcloud source repos clone integrationFR

e. Bring the repository in from GitHub

i. git clone https://github.com/linuxacademy/content-gcpro-developer

f. Copy the GitHub files just brought in, to the source repository

    i. cp content-gcpro-developer/cloud-functions-lab/* integrationFR

4. Open the GitHub file using the following command followed by enter

    a. cd integrationFR

5. Configure two Git variables so the files can be worked with

    a. git config user.name "Elisha Waller"

    b. git config user.email "ewallertech@gmail.com"

6. Commit the files to our main repositories with the following three commands; enter after each command
    a. git add .
    b. git commit -m "Initial commit"
    c. git push origin master

7. Revisit the open Cloud Source Repositories tab; refresh the page; and now, our three files are included within the repository

8. Create a Cloud Function
   a. "Navigation menu"; "Cloud Functions"
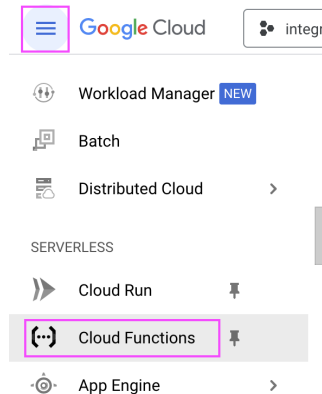
   b. "Create Function"
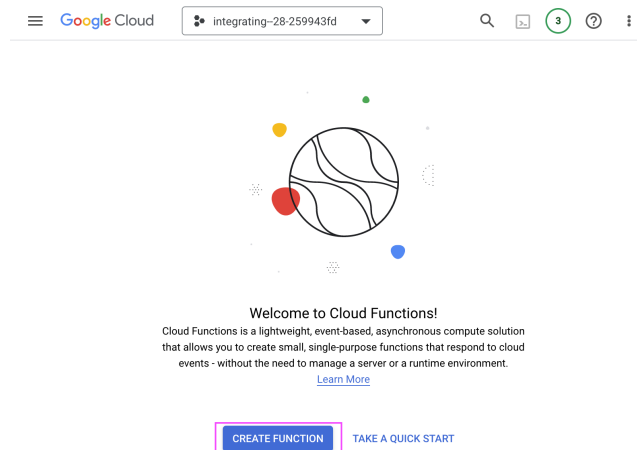
   c. Assign function a "Function name"; ex: function1
   d. Under "Authentication", select "Allow unauthenticated invocations"; "Save"; "Next"
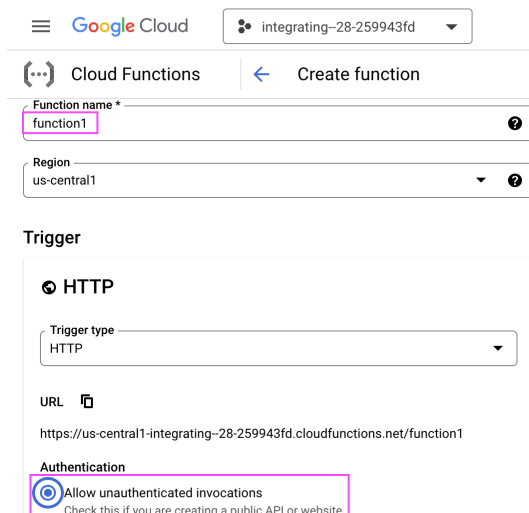
e. Assign a "Runtime"; ex: "Python 3.9"
f. For "Source code" choose "Cloud Source repository"
g. "Entry point"; ex: greetings_http
h. "Repository"; ex: integrationFR
i. "Deploy"

9. Test the Cloud Function to make sure it's working
   a. Click on the function's name

   b. "Trigger"
   c. Click on the trigger link

d. Web page opens with default message
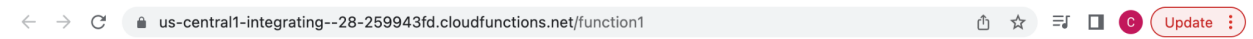
Greetings from Linux Academy, my friend!

Via Source Control Repo

e. Type "?name=Elisha" at end of URL in search bar; enter

Greetings from Linux Academy, my friend!

Via Source Control Repo

f. Web page now displays the name entered in default message
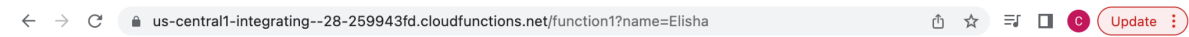
Greetings from Linux Academy, Elisha!

Via Source Control Repo

g. In the Cloud Functions tab click on "Testing"
h. Under the "Configure Triggering Event" section, type inside of brackets: {"name":"Elisha Waller"}
i. "Test the Function"

Cloud Functions — Function details — EDIT — DELETE — COPY — LEARN

function1 1st gen

Version
Version 1, deployed at Oct 17, 2022, 3:13:45 P...

METRICS | DETAILS | SOURCE | VARIABLES | TRIGGER | PERMISSIONS | LOGS | **TESTING**

Configure Triggering Event ❓

Press Alt+F1 for Accessibility Options.

```
1   {"name":"Elisha Waller"}
```

(··) TEST THE FUNCTION

Test Command TEST IN CLOUD SHELL

```
curl -m 70 -X POST https://us-central1-integrating--28-259943fd.c   lfu
-H "Authorization: bearer $(gcloud auth print-identity-token)" \
-H "Content-Type: application/json" \
-d '{"name":"Elisha Waller"}'
```

j. "Output" section shows the completed greeting

console.cloud.google.com/functions/details/us-central1/function1?env=gen1&project=integrating--28-259943... — Update

Google Cloud — integrating--28-259943fd — Search

Cloud Functions — Function details — EDIT — DELETE — COPY — LEARN

function1 1st gen

Version
Version 1, deployed at Oct 17, 2022, 3:13:45 P...

METRICS | DETAILS | SOURCE | VARIABLES | TRIGGER | PERMISSIONS | LOGS | **TESTING**

(··) TEST THE FUNCTION

Testing in the Cloud Console has a 5 minute timeout. Note that this is different from the limit set in the function configuration.

Output ✔ Complete

```
$  <h1 style="margin:20px auto;width:800px;">Greetings from Linux Academy, Elisha Waller!</h1><h2>Via Source Control Repo</h2>
```