Intro Brief:

This demonstration is an end-to-end project using Google Cloud Build, focusing on deploying a Node.js application to Google Cloud Run. This project requires setting up a GitHub repository, configuring Cloud Build triggers, and deploying the application to Cloud Run. I will walk through each step of the process, from setting up the GitHub repository to triggering Automatic Builds upon code push.

Google Cloud Build end-to-end project demonstration

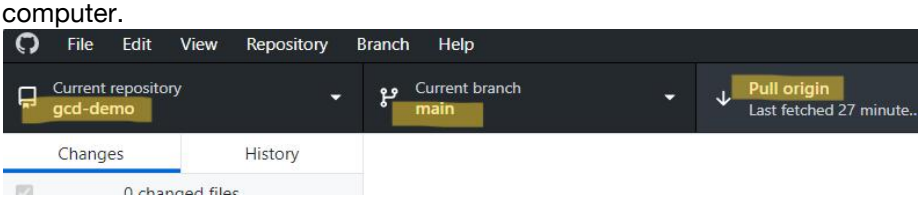
Requirement: Google Cloud SDK
Github Account

1. Set up a GitHub repository for a source code.

2. Create a new repository in a GitHub account. Name it **gcd-demo**.

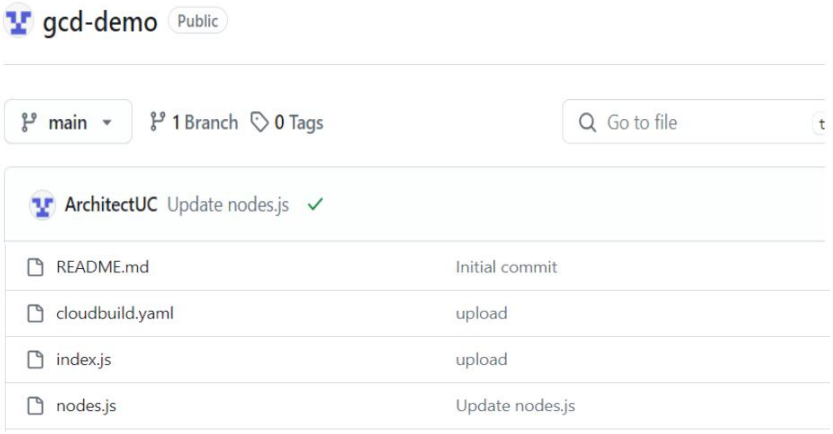
3. Add a .gitignorefile for Node to streamline commits.

4.In GitHub desktop, edit the local path and click clone to synchronize a repository to a local computer.



5.Will deploy an example **Node js** app to Google Cloud.

6.Put files for an example **Node js** app into a local computer's repository folder, **index.js** and **package.json**. (https://github.com/ArchitectUC/gcd-demo)



7. Deploy to Google Cloud Run.

8. Add a **Dockerfile** for our app. (https://github.com/ArchitectUC/gcd-demo)

```
FROM node:16

# Create app directory
WORKDIR /usr/src/app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./

RUN npm install
# If you are building your code for production
# RUN npm ci --only=production

# Bundle app source
COPY . .

EXPOSE 8080
CMD [ "node", "index.js" ]
```

9. Select an existing Google Cloud project, ***gcloud config set project gcloud-demo-deploy***.

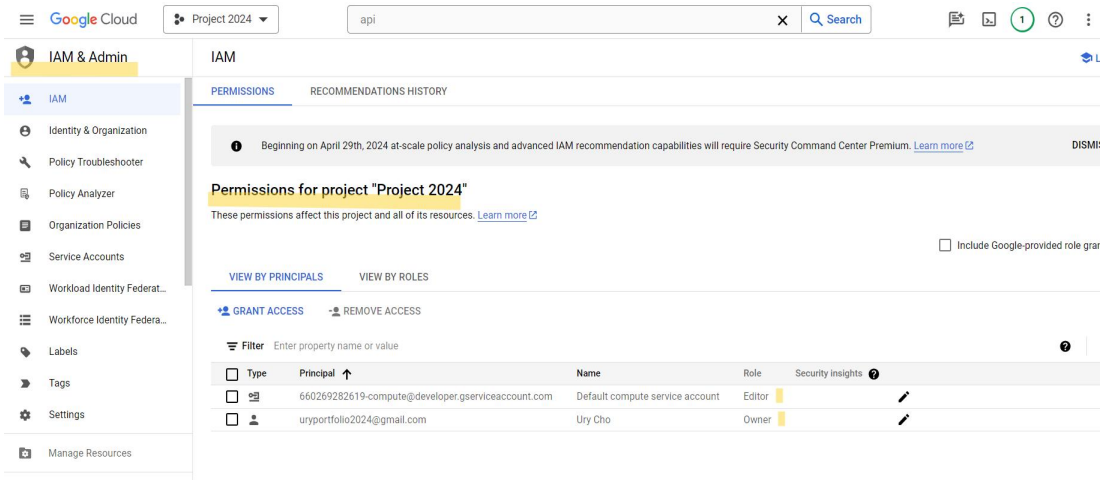
10.deploy it with this command, ***gcloud run deploy hello-world --source . --allow-unauthenticated - -region us-central1***.
After wait, see that the app is on the public URL.

```
PS C:\Users\urych\AppData\Local\Google\Cloud SDK>  gcloud run deploy hello-world --source . --allow-unauthenticated --region us-central1
This command is equivalent to running `gcloud builds submit --pack image=[IMAGE] .` and `gcloud run deploy hello-world --image [IMAGE]`

Building using Buildpacks and deploying container to Cloud Run service [hello-world] in project [successproject2024] region [us-central1]
OK Building and deploying... Done.
  OK Uploading sources...
  OK Building Container... Logs are available at [https://console.cloud.google.com/cloud-build/builds/b6266bca-27a0-4f9
  e-ae38-e5daf08aaa93?project=660269282619].
  OK Creating Revision...
  OK Routing traffic...
  OK Setting IAM Policy...
Done.
Service [hello-world] revision [hello-world-00002-wss] has been deployed and is serving 100 percent of traffic.
Service URL: https://hello-world-bwimda3fea-uc.a.run.app
```

11.To use Cloud Build, we need a **cloudbuild.yaml** configuration file. Here's a file that works with Cloud Run. (https://github.com/ArchitectUC/gcd-demo)

```
steps:
# deploy to Cloud Run
- name: 'gcr.io/google.com/cloudsdktool/cloud-sdk'
  entrypoint: 'bash'
  args: ['-c', 'gcloud run deploy hello-world --source . --allow-unauthenticated --region us-central1']
```

12. Enable the Cloud Run admin role at the Cloud Build settings page of Google Cloud .



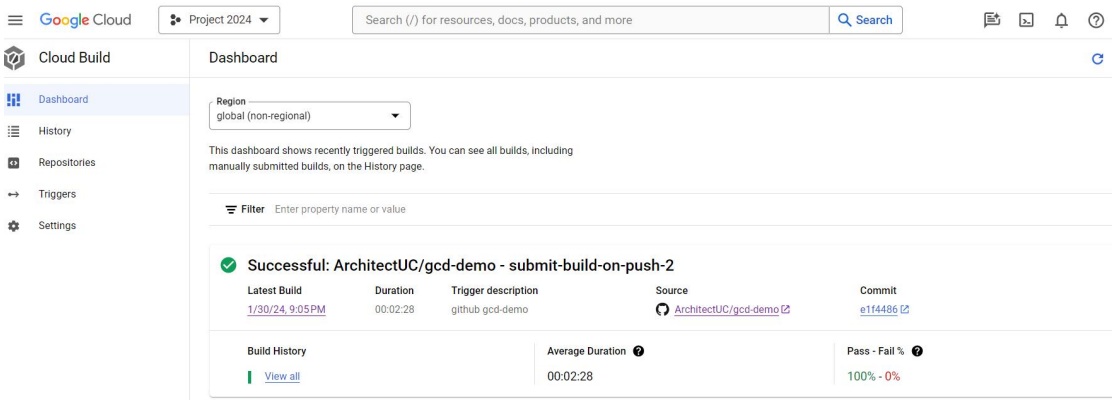
13. Modified the **index.js** file to mention *Cloud Build*.

```
index.js > app.get('/') callback
1   const express = require('express');
2   const app = express();
3
4   app.get('/', (req, res) => {
5     res.send('Hello World with Cloud Run and Cloud Build');
6   });
7
8   const port = parseInt(process.env.PORT) || 8080;
9   app.listen(port, () => {
10    console.log(`helloworld: listening on port ${port}`);
11  });
```

14. To submit The Cloud Build, command ***gcloud builds submit***. After wait, we see that the app is updated.

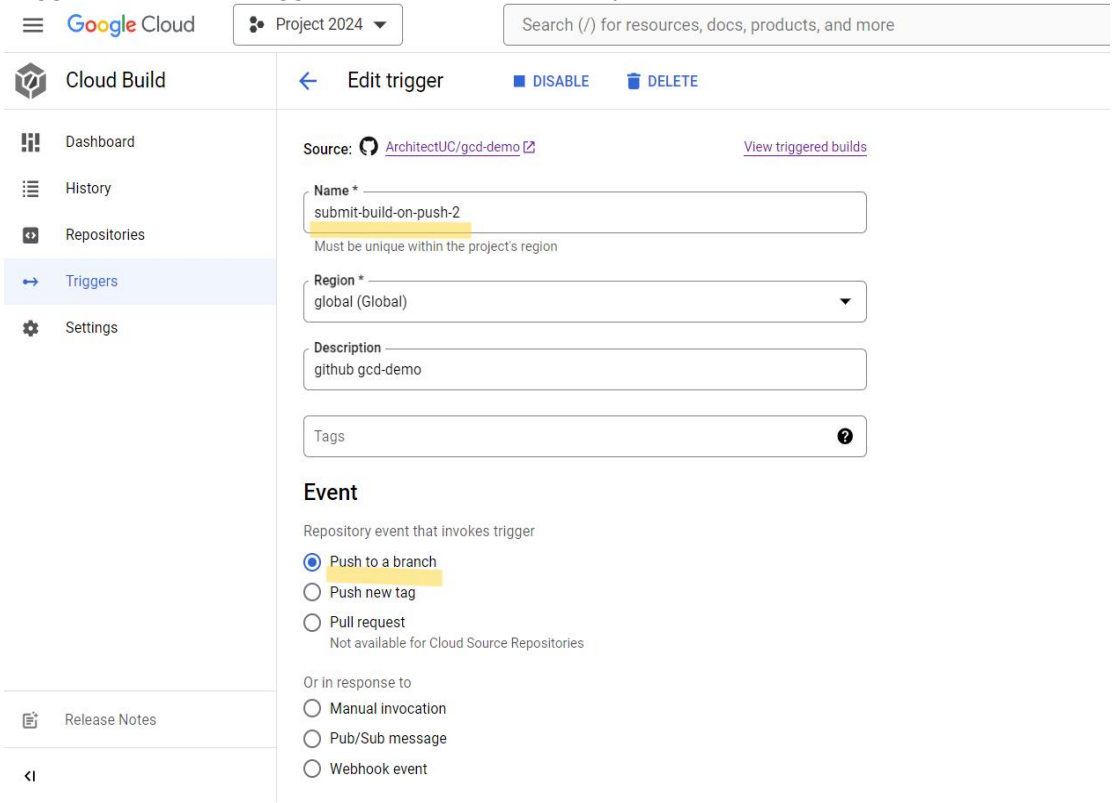
```
    Directory: C:\Users\urych\AppData\Local\Google\Cloud SDK


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2024-01-30   5:22 PM                gcd
d-----        2024-01-30   2:21 PM                google-cloud-sdk
-a----        2024-01-30   5:49 PM            204 cloudbuild.yaml
-a----        2024-01-30   2:18 PM            263 cloud_env.bat
-a----        2024-01-30   3:20 PM              0 gcloud
-a----        2024-01-30   2:22 PM              1 install_mode
-a----        2022-02-02   8:44 AM           1150 supercloud-16x16.ico
-a----        2024-01-30   2:22 PM          63592 uninstaller.exe


PS C:\Users\urych\AppData\Local\Google\Cloud SDK> gcloud builds submit
Creating temporary tarball archive of 62362 file(s) totalling 1.3 GiB before compression.
Uploading tarball of [.] to [gs://successproject2024_cloudbuild/source/1706665779.083225-73f2d2cebf7644af92192d7ebe1171fc.tgz]
Created [https://cloudbuild.googleapis.com/v1/projects/successproject2024/locations/global/builds/52ba501a-a41f-4386-ba5e-d9c7a1206154].
Logs are available at [ https://console.cloud.google.com/cloud-build/builds/52ba501a-a41f-4386-ba5e-d9c7a1206154?project=660269282619 ].
------------------------------------------------------------------------ REMOTE BUILD OUTPUT ------------------------------------------------------------------------
starting build "52ba501a-a41f-4386-ba5e-d9c7a1206154"

FETCHSOURCE
Fetching storage object: gs://successproject2024_cloudbuild/source/1706665779.083225-73f2d2cebf7644af92192d7ebe1171fc.tgz#1706666334337131
Copying gs://successproject2024_cloudbuild/source/1706665779.083225-73f2d2cebf7644af92192d7ebe1171fc.tgz#1706666334337131...
/ [1 files][310.1 MiB/310.1 MiB]
Operation completed over 1 objects/310.1 MiB.
BUILD
Pulling image: gcr.io/google.com/cloudsdktool/cloud-sdk
Using default tag: latest
latest: Pulling from google.com/cloudsdktool/cloud-sdk
e455cf41eadb: Already exists
120492d90f3d: Pulling fs layer
6cae549e2eb0: Pulling fs layer
9b0f017ca475: Pulling fs layer
```

15. See successful build in the **Cloud Build** history page.



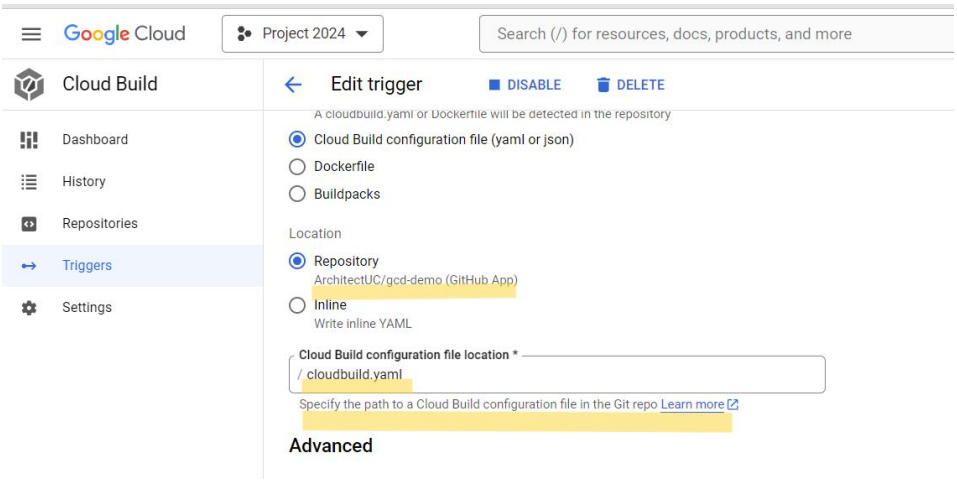
16. In GitHub desktop, we see the example app files. Click Push Origin to synchronize the changes with the remote repository.

17. Add a build trigger to deploy pushing code to GitHub. We can combine the steps of pushing code and submitting the build. Go to the Cloud Build Triggers page and click Create Trigger. We create a trigger called *submit-build-on-push-2*.

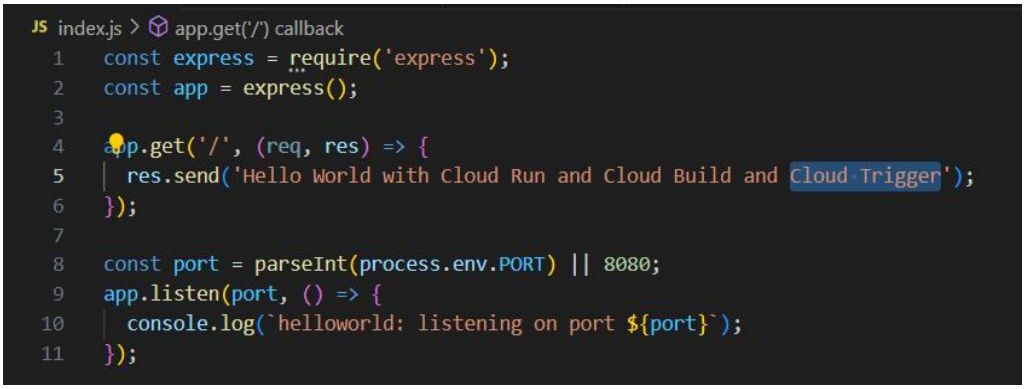


18. In source, connect our GitHub repository. We specified the main branch.

19. Select Cloud Build configuration file and set the file as **cloudbuild.yaml**. Click Create.



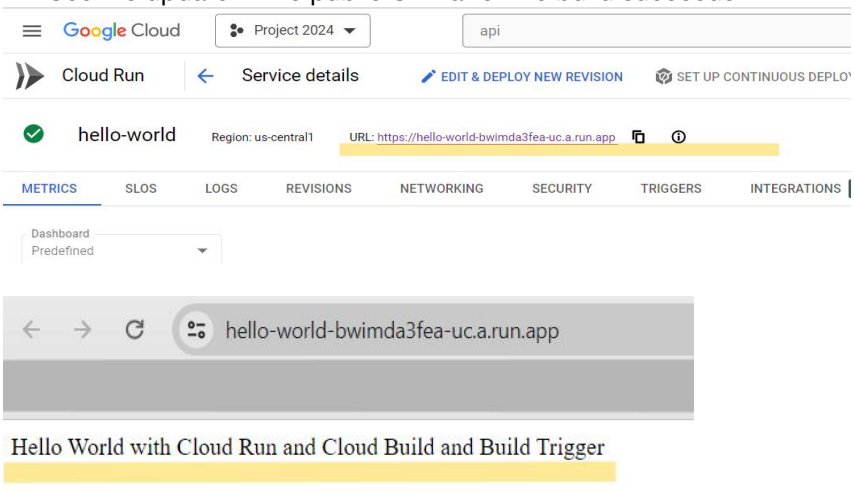
19. Modify the **index.js** file to mention to *Cloud Trigger*.



This time, instead of running the command to submit a build we go to GitHub desktop and commit the update to the main branch.

20. Click Push Origin. See that a new build started automatically Cloud Build history page.

22. See the update in the public URL after the build succeeds.



*Code Reference : LinkedInLearning/gcd-learning-the-cloud-build-ci-cd-platform-3094987:06_01e