Push code -> SOURCE CODE REPO -> BUILD SERVER -> CREATE ARTIFACT-> PUSH TO DEPLOYMENT SERVER ->
Build server will check that our code is properly executing or not. If not then the developer will make changes and again start building .
CONTINUOUS DEPLOYMENT : Fully automated , no manual intervention , code is continuously built and deployed .
Continuous delivery : Release to production , may involve manual approval , it will make sure delivery are fast ,

GCP SERVICES FOR CICD :
SOURCE CODE MANAGEMENT - BITBUCKET , MERCURIAL , GITLAB - PUBLIC REPOS .
GOOGLE - CLOUD SOURCE REPOSITORY .

BUILD - It depends on the runtime env we use . if we build some java code which will generate some jar/war/ear . So the build server is responsible for managing that . JENKINS , CIRCLECI . TEAMCITY . - GOOGLE - CLOUD BUILD .

ARTIFACT STORAGE - DOCKER HUB - to store docker images , jfrog artifactory , amazon s3
CONTAINER REGISTRY , ARTIFACT REGISTRY - we can store ear , war , jar or docker images etc .

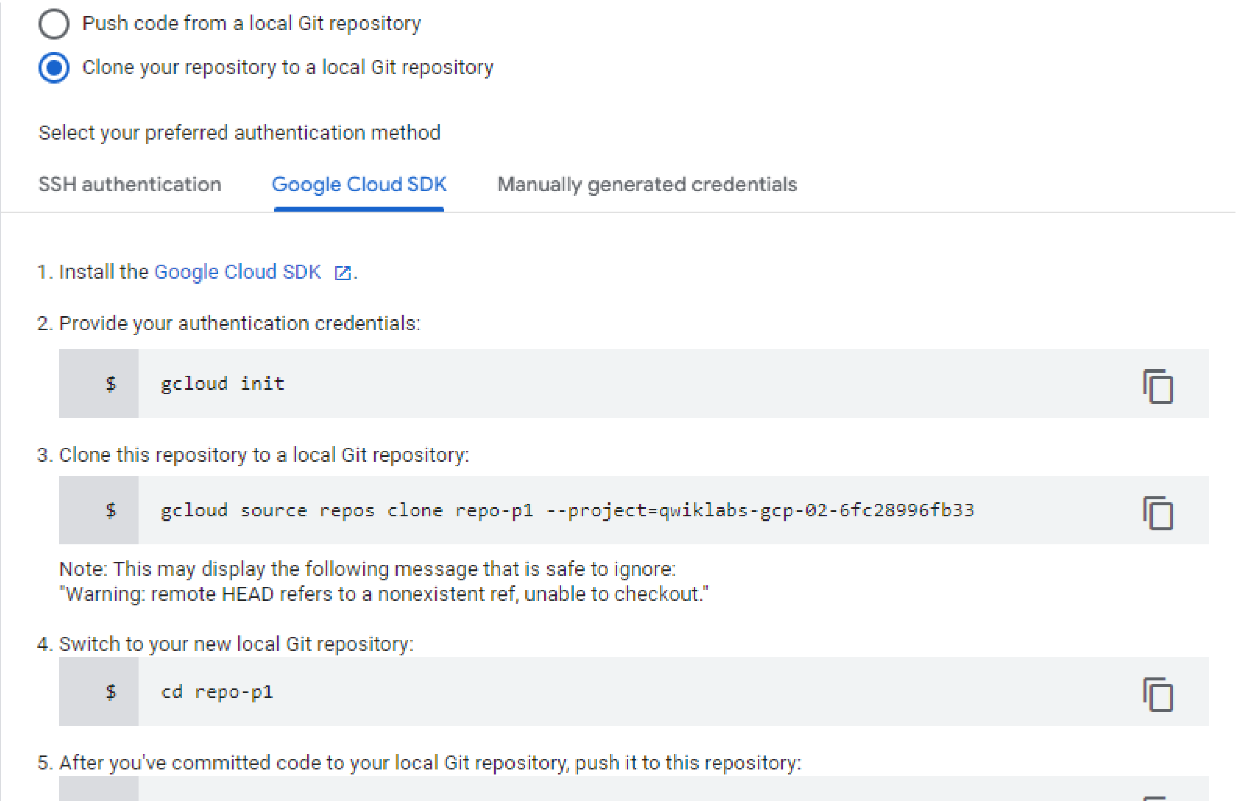
DEPLOYMENT - Where we want to deploy - compute engine , kubernetes , App engine , Cloud run , CLoud function .

SETTING UP CICD PIPELINE IN GCP :

Create a docker image and push to the container registry . Whenever we push new code to the source code repo a new docker image will be created and pushed to the container repo .
STEPS :
1) For source code we need - Dockerfile and main.py .
2) CLoud build - to build docker images
Cloud build is used for - we define build execution steps , for docker image .
3) Push the image to the registry .

Naming : use repo-p1 , build-p1 , image-p1 .

SETUP
SOURCE CODE REPO : Create new repo / connect external repo . Create a new repo for this hands on . > repo-p1 > Create .



○ Push code from a local Git repository

● Clone your repository to a local Git repository

Select your preferred authentication method

1. Install the Google Cloud SDK.
2. Provide your authentication credentials:

```
$ gcloud init
```

3. Clone this repository to a local Git repository:

```
$ gcloud source repos clone repo-p1 --project=qwiklabs-gcp-02-6fc28996fb33
```

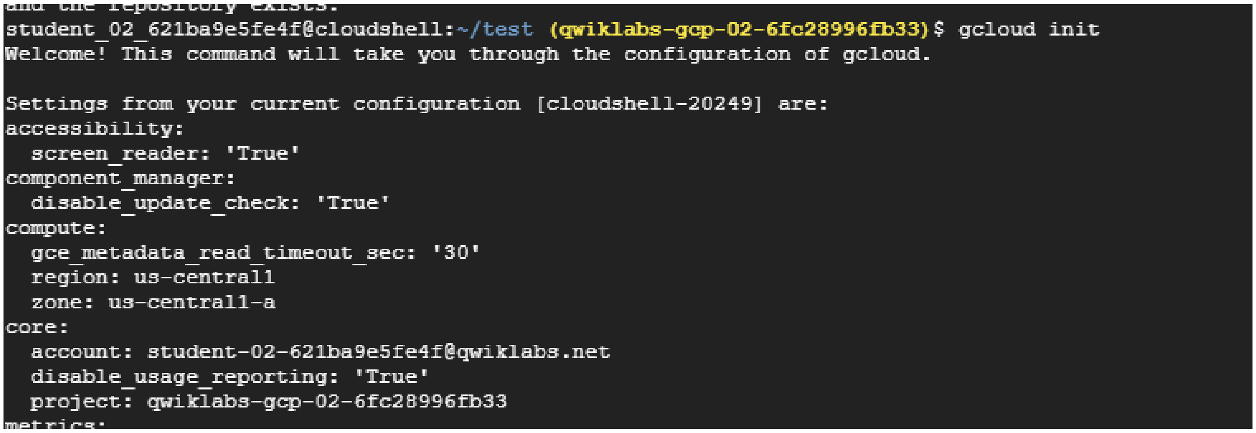
Note: This may display the following message that is safe to ignore:
"Warning: remote HEAD refers to a nonexistent ref, unable to checkout."

4. Switch to your new local Git repository:

```
$ cd repo-p1
```

5. After you've committed code to your local Git repository, push it to this repository:

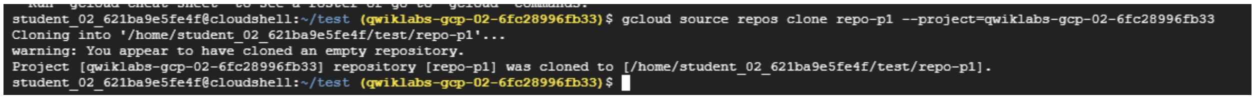
Use google cloud sdk steps .
gcloud init



```
and the repository exists:
student_02_621ba9e5fe4f@cloudshell:~/test (qwiklabs-gcp-02-6fc28996fb33)$ gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [cloudshell-20249] are:
accessibility:
  screen_reader: 'True'
component_manager:
  disable_update_check: 'True'
compute:
  gce_metadata_read_timeout_sec: '30'
  region: us-central1
  zone: us-central1-a
core:
  account: student-02-621ba9e5fe4f@qwiklabs.net
  disable_usage_reporting: 'True'
  project: qwiklabs-gcp-02-6fc28996fb33
metrics:
```

create dir . clone the repo .



```
student_02_621ba9e5fe4f@cloudshell:~/test (qwiklabs-gcp-02-6fc28996fb33)$ gcloud source repos clone repo-p1 --project=qwiklabs-gcp-02-6fc28996fb33
Cloning into '/home/student_02_621ba9e5fe4f/test/repo-p1'...
warning: You appear to have cloned an empty repository.
Project [qwiklabs-gcp-02-6fc28996fb33] repository [repo-p1] was cloned to [/home/student_02_621ba9e5fe4f/test/repo-p1].
student_02_621ba9e5fe4f@cloudshell:~/test (qwiklabs-gcp-02-6fc28996fb33)$
```

add Dockerfile and main.py file to repo .
Push the changes to repo .

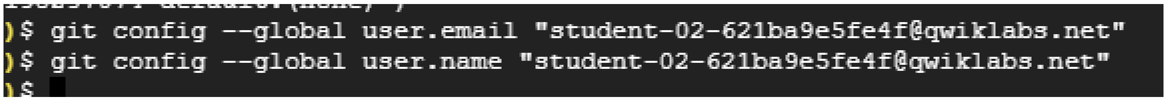
cat Dockerfile :
FROM python
RUN pip install flask

```
WORKDIR /myapp
COPY ./main.py /myapp/
CMD ["python","/myapp/main.py"]
```

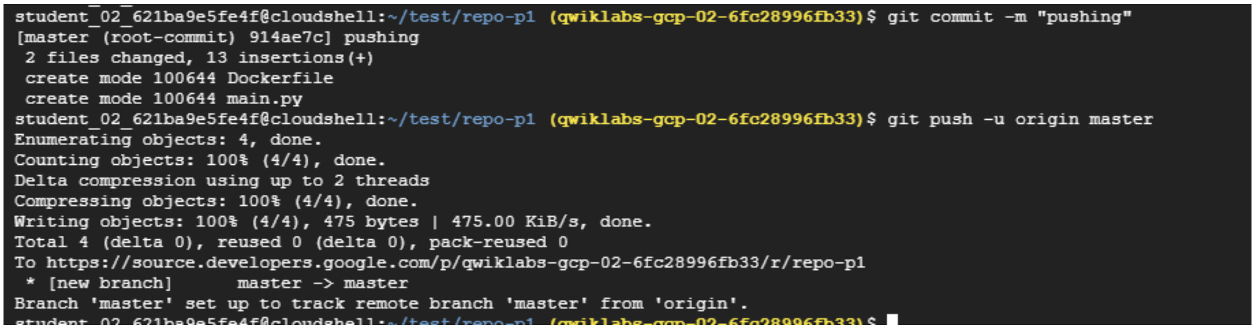
cat main.py :

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
   return 'Welcome to V1'
if __name__=='__main__':
   app.run(host='0.0.0.0',port=8080)
```

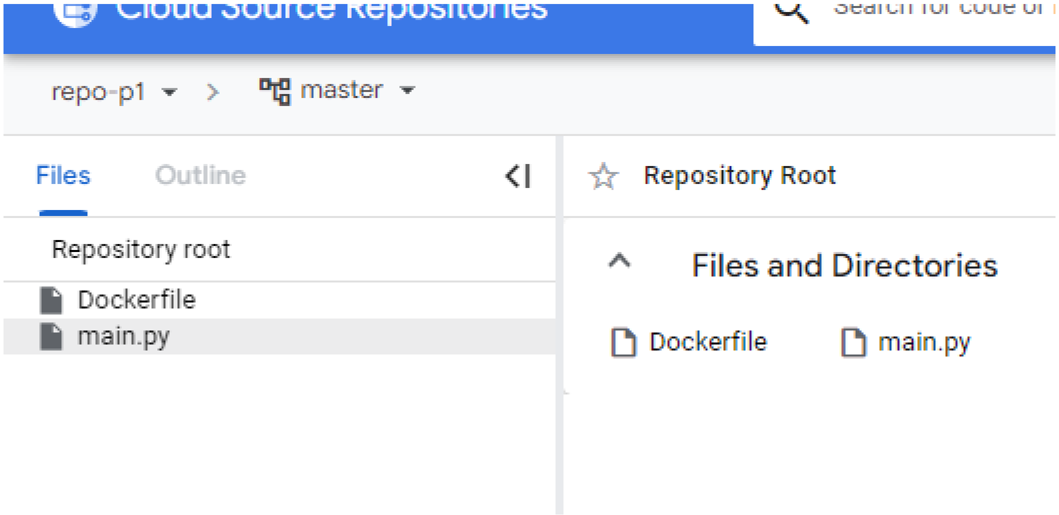
Push to repo .



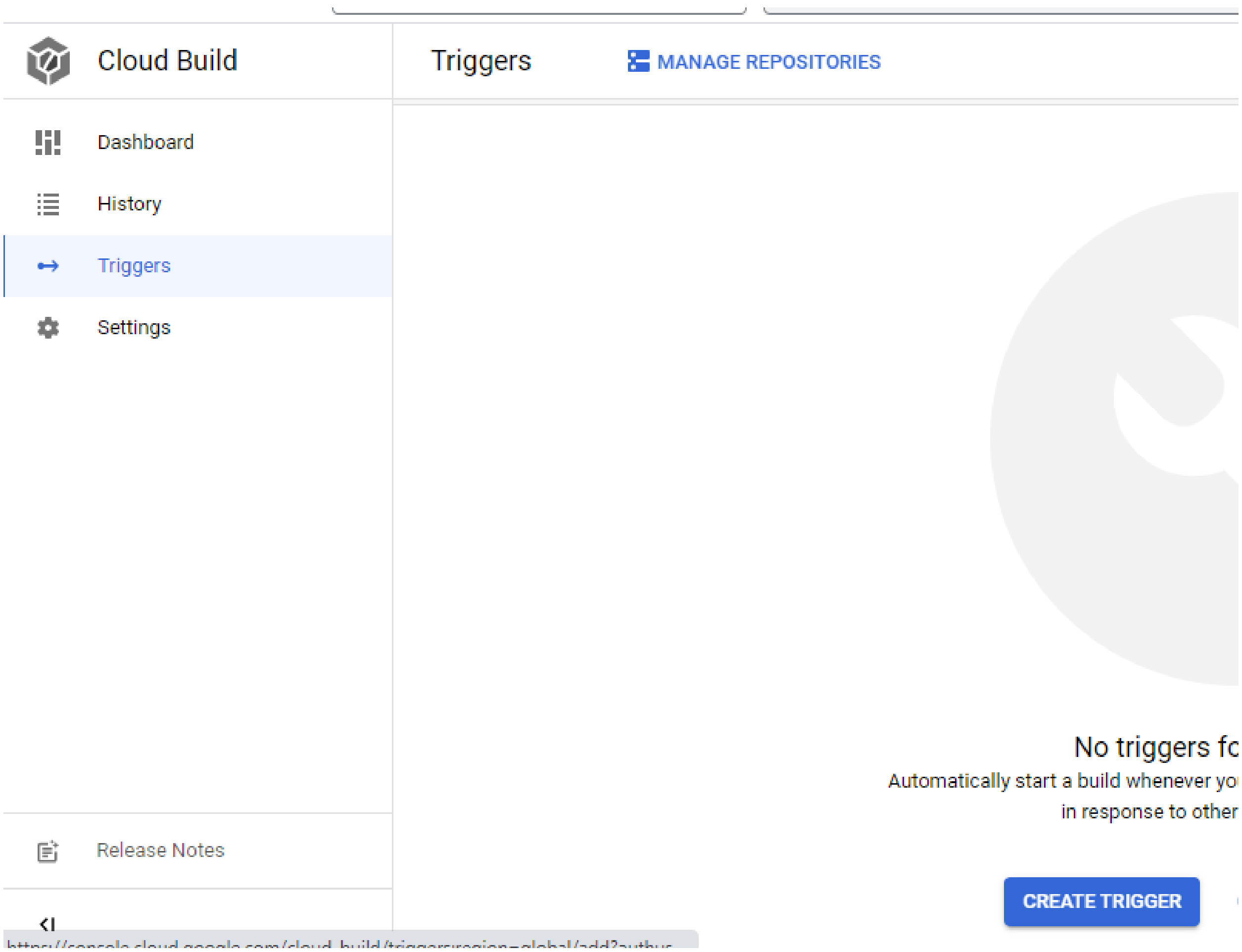
```
)$ git config --global user.email "student-02-621ba9e5fe4f@qwiklabs.net"
)$ git config --global user.name "student-02-621ba9e5fe4f@qwiklabs.net"
)$
```



```
student_02_621ba9e5fe4f@cloudshell:~/test/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ git commit -m "pushing"
[master (root-commit) 914ae7c] pushing
 2 files changed, 13 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 main.py
student_02_621ba9e5fe4f@cloudshell:~/test/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 475 bytes | 475.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://source.developers.google.com/p/qwiklabs-gcp-02-6fc28996fb33/r/repo-p1
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
student_02_621ba9e5fe4f@cloudshell:~/test/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$
```



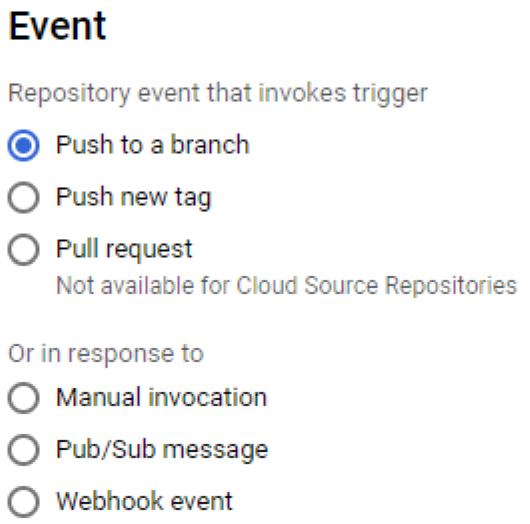
CLOUD BUILD : Create trigger . >



No triggers fc

Automatically start a build whenever yo
in response to other

build-p1 >event (based on which event trigger should happen in cloud build) > push to a branch



Event

Repository event that invokes trigger

● Push to a branch

○ Push new tag

○ Pull request
Not available for Cloud Source Repositories

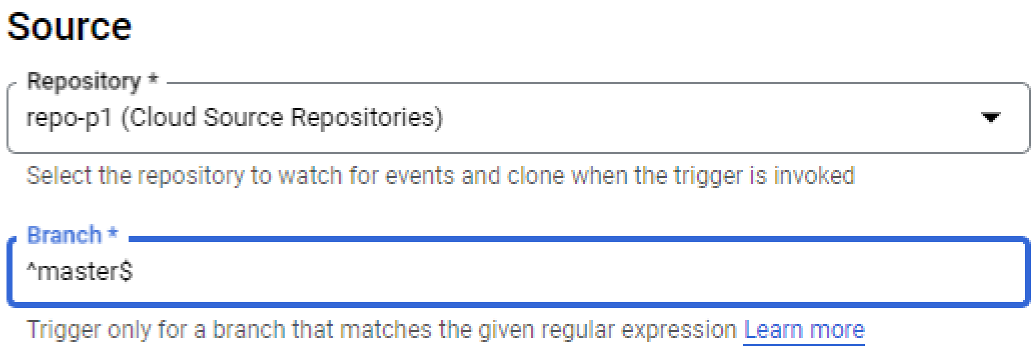
Or in response to

○ Manual invocation

○ Pub/Sub message

○ Webhook event

> select repo > branch - (which branch to check for invoking trigger) .*



Source

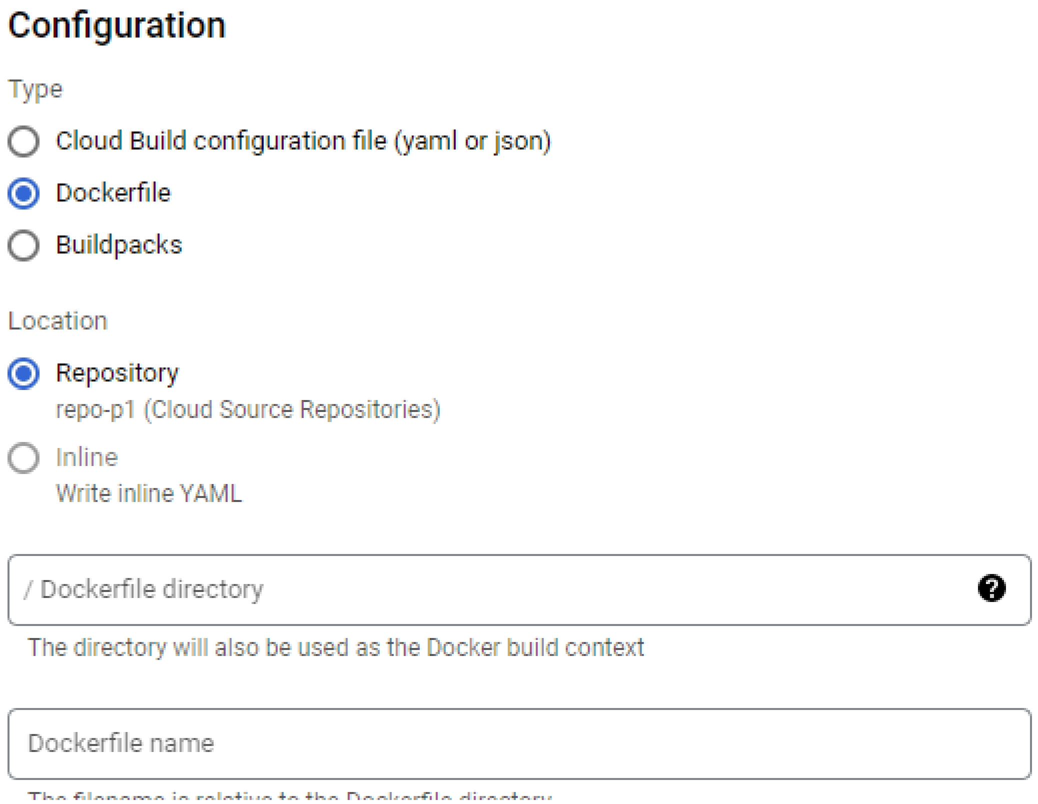
Repository *
repo-p1 (Cloud Source Repositories) ▾

Select the repository to watch for events and clone when the trigger is invoked

Branch *
^master$

Trigger only for a branch that matches the given regular expression Learn more

> we have include filter also if we want to apply filters in trigger . >



Configuration

Type

○ Cloud Build configuration file (yaml or json)

● Dockerfile

○ Buildpacks

Location

● Repository
repo-p1 (Cloud Source Repositories)

○ Inline
Write inline YAML

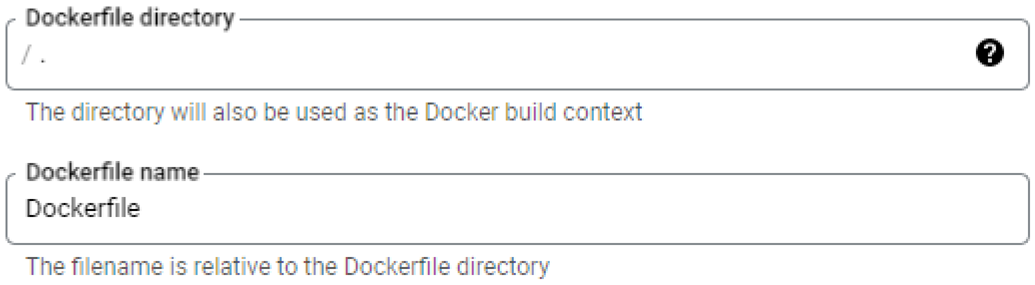
/ Dockerfile directory

The directory will also be used as the Docker build context

Dockerfile name

The filename is relative to the Dockerfile directory

configuration - how this build will happen . - we will use Dockerfile for trigger of cloud build . > location -> repo - we have to specify docker file dir - it will search for dockerfile in the directory



Dockerfile directory
/ .

The directory will also be used as the Docker build context

Dockerfile name
Dockerfile

The filename is relative to the Dockerfile directory

.> image name - rename naming convention to image-p1.

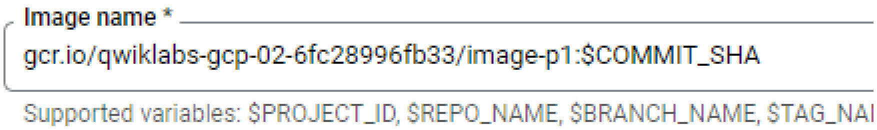
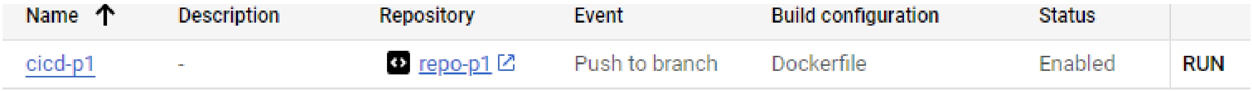


Image name *

gcr.io/qwiklabs-gcp-02-6fc28996fb33/image-p1:$COMMIT_SHA

Supported variables: $PROJECT_ID, $REPO_NAME, $BRANCH_NAME, $TAG_NAI

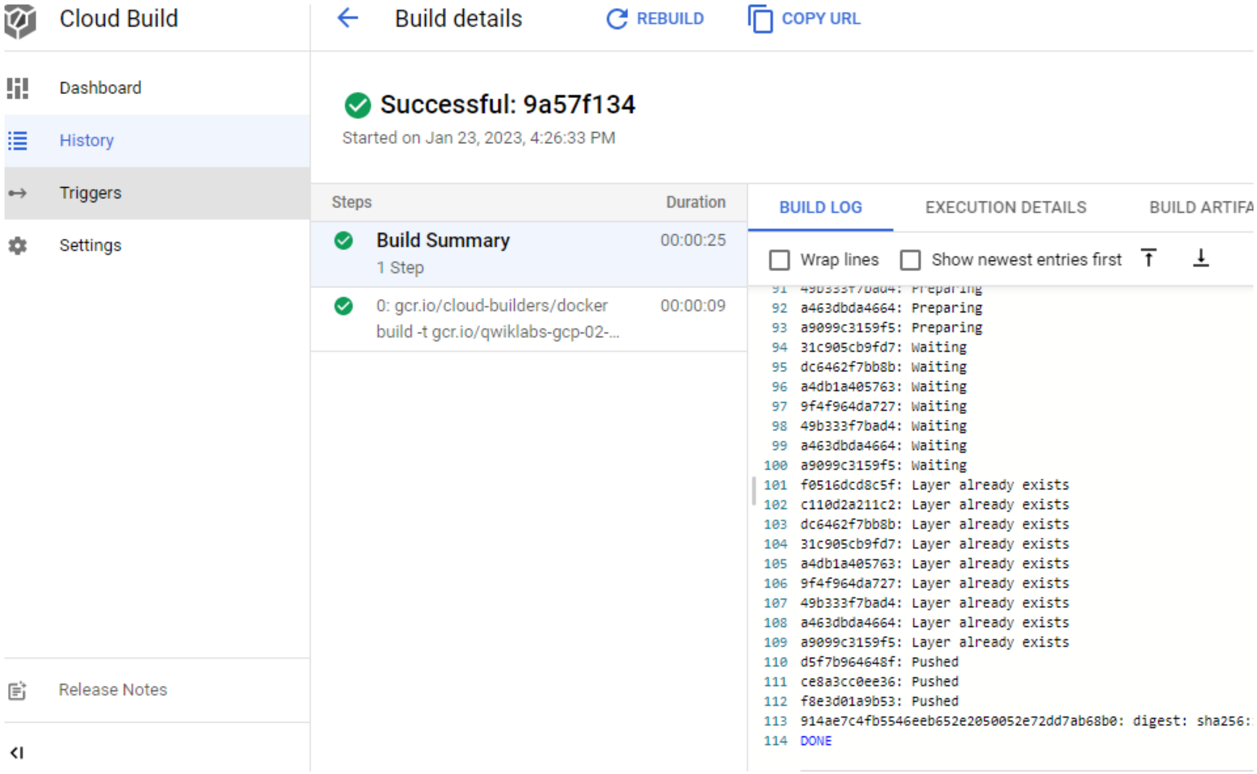
> service account (we can execute cloud build using some service account to implement security in org) we will not use service account > create



Name ↑	Description	Repository	Event	Build configuration	Status	
cicd-p1	-	repo-p1	Push to branch	Dockerfile	Enabled	RUN

CLoud build will run docker build command from the root directory that we provide .

Container registry > here we can see the image which will be pushed by cloud build .
We can run triggers by specifying branch names manually for the first time .
In history we can see triggered builds .



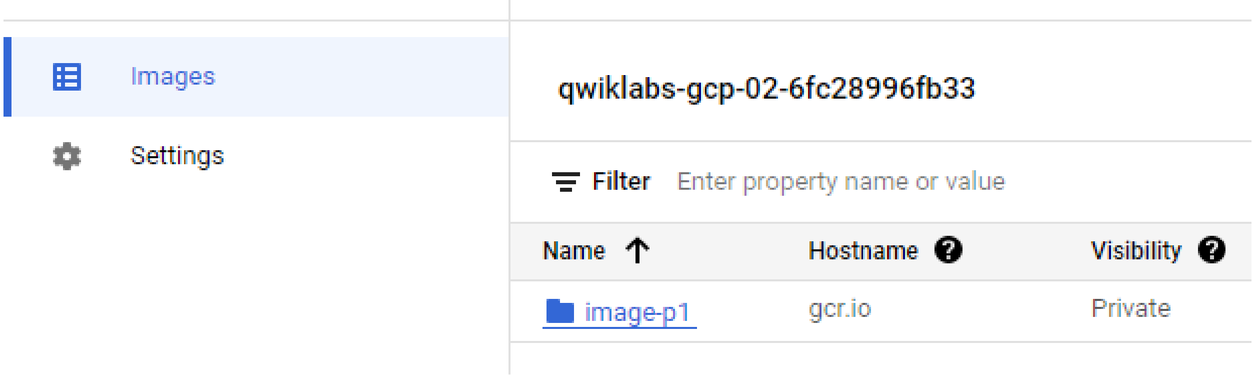
Successful: 9a57f134

Started on Jan 23, 2023, 4:26:33 PM

Steps	Duration
Build Summary 1 Step	00:00:25
0: gcr.io/cloud-builders/docker build -t gcr.io/qwiklabs-gcp-02-...	00:00:09

□ Wrap lines □ Show newest entries first

```
49b333f7bad4: Preparing
a463dbda4664: Preparing
a9099c3159f5: Preparing
31c905cb9fd7: Waiting
dc6462f7bb8b: Waiting
a4db1a405763: Waiting
9f4f964da727: Waiting
49b333f7bad4: Waiting
a463dbda4664: Waiting
a9099c3159f5: Waiting
f0516dcd8c5f: Layer already exists
c110d2a211c2: Layer already exists
dc6462f7bb8b: Layer already exists
31c905cb9fd7: Layer already exists
a4db1a405763: Layer already exists
9f4f964da727: Layer already exists
49b333f7bad4: Layer already exists
a463dbda4664: Layer already exists
a9099c3159f5: Layer already exists
d5f7b964648f: Pushed
ce8a3cc0ee36: Pushed
f8e3d01a9b53: Pushed
914ae7c4fb5546eeb652e2050052e72dd7ab68b0: digest: sha256:
DONE
```

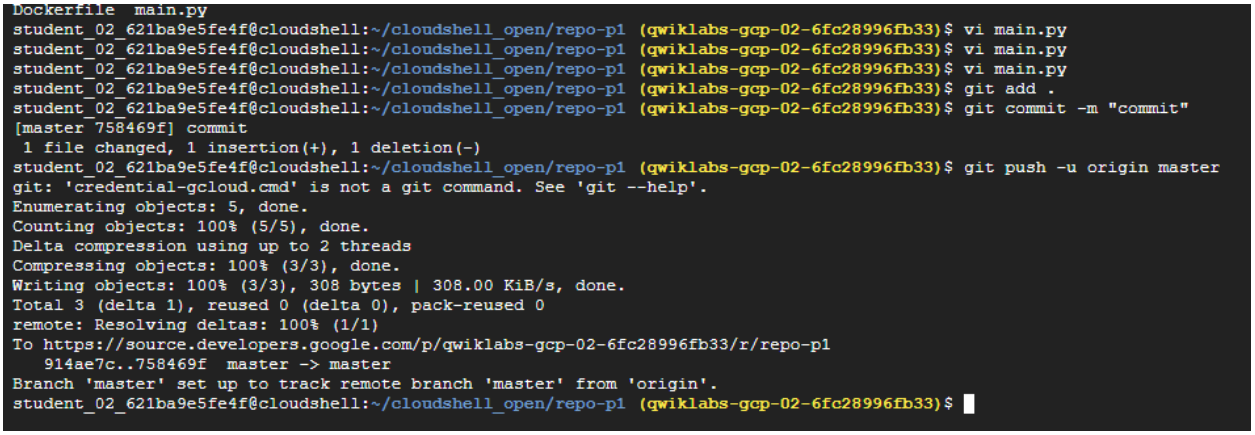


qwiklabs-gcp-02-6fc28996fb33

Filter Enter property name or value

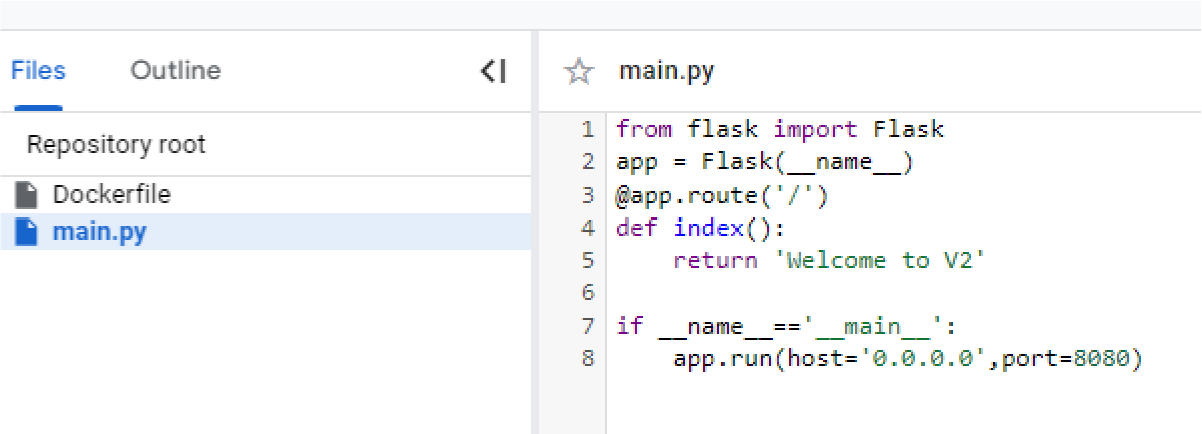
Name ↑	Hostname	Visibility
image-p1	gcr.io	Private

Trigger build after pushing code :
Edit main.py file and push to git . The new trigger will start automatically . We can check in cloud build . Image will be added in the container registry .



```
Dockerfile  main.py
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ vi main.py
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ vi main.py
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ vi main.py
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ git add .
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ git commit -m "commit"
[master 758469f] commit
 1 file changed, 1 insertion(+), 1 deletion(-)
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$ git push -u origin master
git: 'credential-gcloud.cmd' is not a git command. See 'git --help'.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1)
To https://source.developers.google.com/p/qwiklabs-gcp-02-6fc28996fb33/r/repo-p1
   914ae7c..758469f  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p1 (qwiklabs-gcp-02-6fc28996fb33)$
```

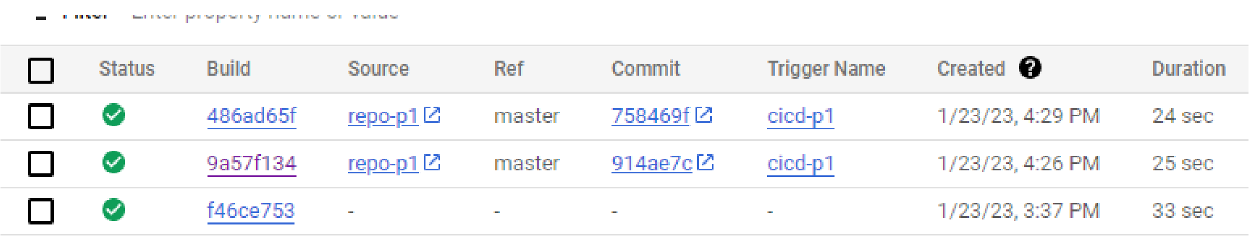
Update version to v2 and push to git .



main.py

```python
from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
    return 'Welcome to V2'

if __name__=='__main__':
    app.run(host='0.0.0.0',port=8080)
```



□	Status	Build	Source	Ref	Commit	Trigger Name	Created	Duration
□	✓	486ad65f	repo-p1	master	758469f	cicd-p1	1/23/23, 4:29 PM	24 sec
□	✓	9a57f134	repo-p1	master	914ae7c	cicd-p1	1/23/23, 4:26 PM	25 sec
□	✓	f46ce753	-	-	-	-	1/23/23, 3:37 PM	33 sec

Trigger has initiated as we push to repo .

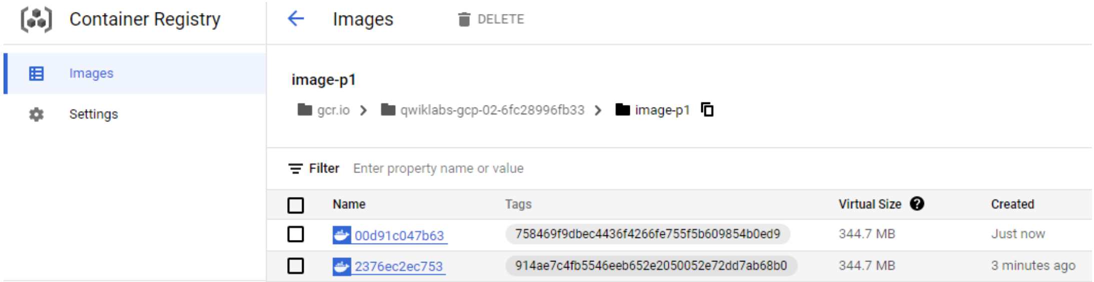


image-p1

gcr.io > qwiklabs-gcp-02-6fc28996fb33 > image-p1

Filter Enter property name or value

□	Name	Tags	Virtual Size	Created
□	00d91c047b63	758469f9dbec4436f4266fe755f5b609854b0ed9	344.7 MB	Just now
□	2376ec2ec753	914ae7c4fb5546eeb652e2050052e72dd7ab68b0	344.7 MB	3 minutes ago

New image is also pushed .