DEPLOY PYTHON WEB APP TO APP ENGINE

SOURCE CODE : app.yaml , main.py , requirements.txt
main.py

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
  return 'Web App with Python Flask!'
if __name__==’__main__’:
app.run(host=’0.0.0.0’,port=8080)
```

—----------

requirements.txt - we have used flask so we have to add flask version
CHEck flask version in cloud shell : Python3
Import flask
print(flask.__version__),

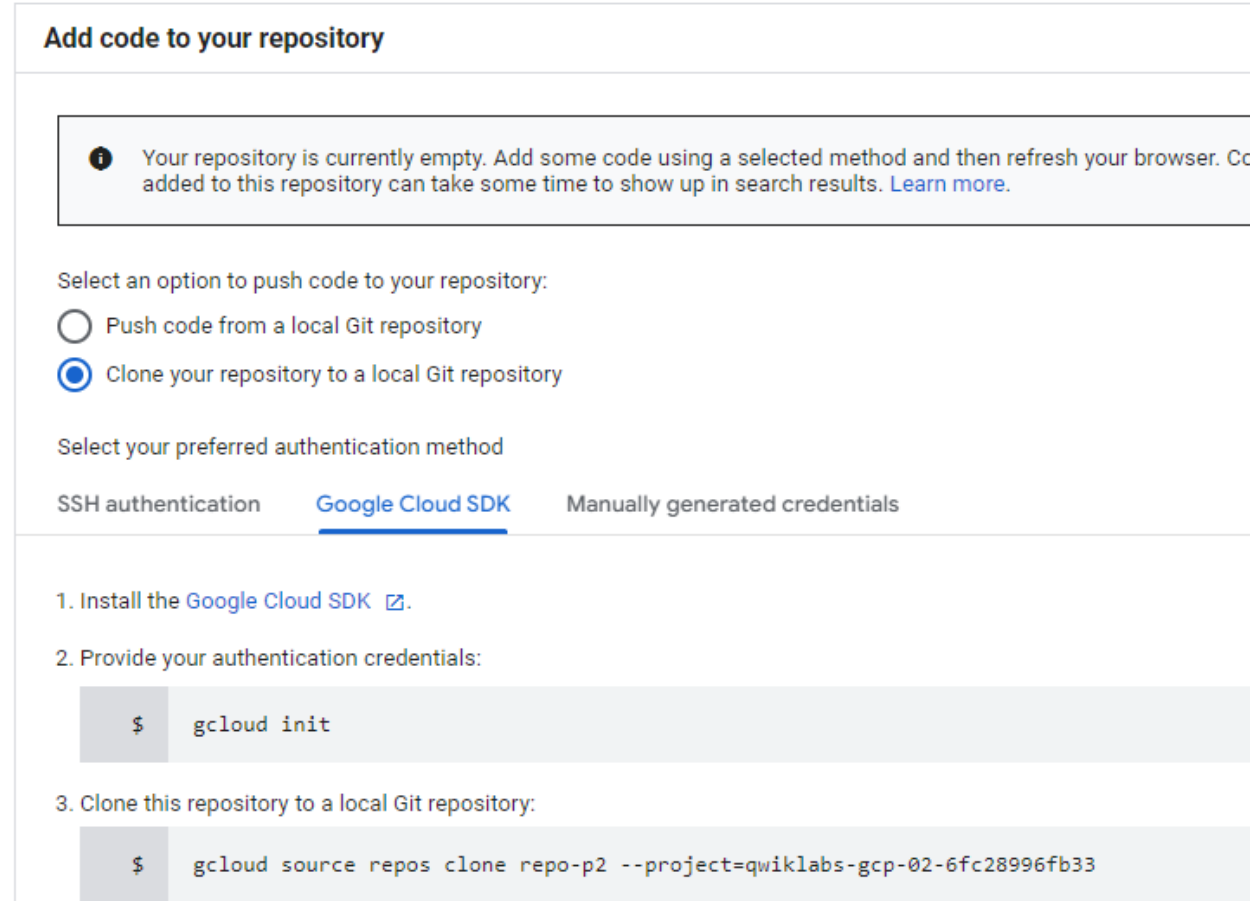
- Flask==version.

—----------

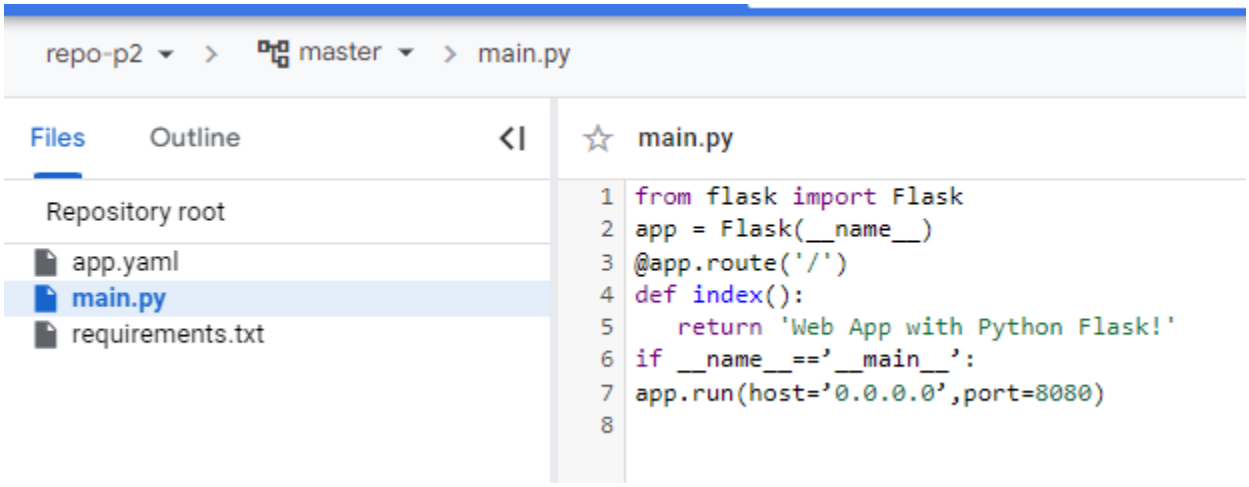
App.yaml :
runtime: python37
CLOUD BUILD : cloudbuild.yaml - we will write all execution steps required for building source code .
DEPLOY TO APP ENGINE :

repo-p2 , cicd-p2 ,

Create cloud source repo repo-p2 > create .
Push app.yaml , main.py , requirements.txt to cloud source repo .




```
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open (qwiklabs-gcp-02-6fc28996fb33)$ gcloud source repos clone repo-p2 --project=qwiklabs-gcp-02-6fc28996fb33
Cloning into '/home/student_02_621ba9e5fe4f/cloudshell_open/repo-p2'...
warning: You appear to have cloned an empty repository.
Project [qwiklabs-gcp-02-6fc28996fb33] repository [repo-p2] was cloned to [/home/student_02_621ba9e5fe4f/cloudshell_open/repo-p2].
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open (qwiklabs-gcp-02-6fc28996fb33)$ cd repo-p2
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p2 (qwiklabs-gcp-02-6fc28996fb33)$
```



Use all files used for manual deployment to the app engine .
We will deploy the above files to the app engine .
Cloudbuild.yaml :

This is a build config file we can create in json or yaml format .
For each task we have a cloud builder present which we can include in the cloudbuild.yaml file .

Builder	Name	Example
bazel	gcr.io/cloud-builders/bazel	bazel example
docker	gcr.io/cloud-builders/docker	docker example
git	gcr.io/cloud-builders/git	git example
gcloud	gcr.io/cloud-builders/gcloud	gcloud example
gke-deploy	gcr.io/cloud-builders/gke-deploy	gke-deploy exampl
gradle	gcr.io/cloud-builders/gradle	gradle example
maven	gcr.io/cloud-builders/mvn	maven example

Above is a list of cloud builds builder which can be included in cloudbuild yaml file .
We can add steps for our execution tasks .
For each step we will need a builder .
Eg if we want to perform some functionality related to docker then we can use docker builder for it .
ABove are full fledged container images provided by google . We can also create custom build steps .
We can also use a community provided cloud builder .

To deploy an application to the app engine we use the command - gcloud app deploy .
So in the cloudbuild.yaml file we have to perform the steps below .

- We will use cloud sdk images of cloud builders .
- Enter to bash
- We need to fire 2 command :
  Gcloud config set timeout 1600 - if deployment takes beyond 1600 sec it will timeout .
  Gcloud app deploy - to deploy from source code repository .
- Logging - cloud logging we will use - we can also use bucket level logging so all log will be redirected to cloud storage bucket .

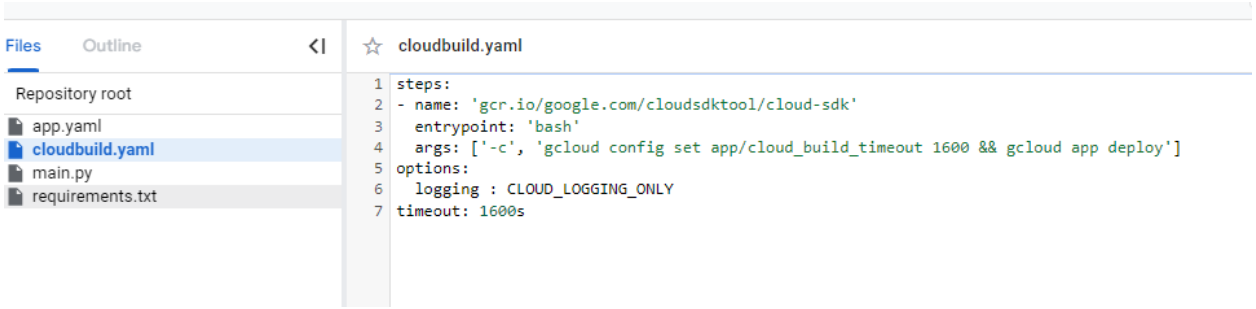
Cat cloudbuild.yaml :
cat cloudbuild.yaml

```
steps:
- name: 'gcr.io/google.com/cloudsdktool/cloud-sdk'
  entrypoint: 'bash'
  args: ['-c', 'gcloud config set app/cloud_build_timeout 1600 && gcloud app deploy']
```

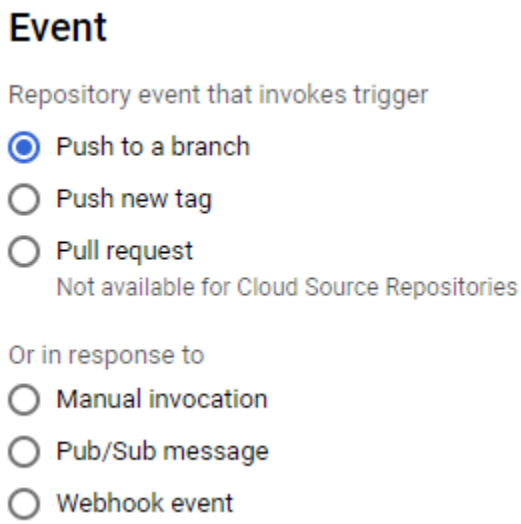
options:
  logging : CLOUD_LOGGING_ONLY
timeout: 1600s

Details about cloudbuild.yaml config :
https://cloud.google.com/build/docs/configuring-builds/create-basic-configuration

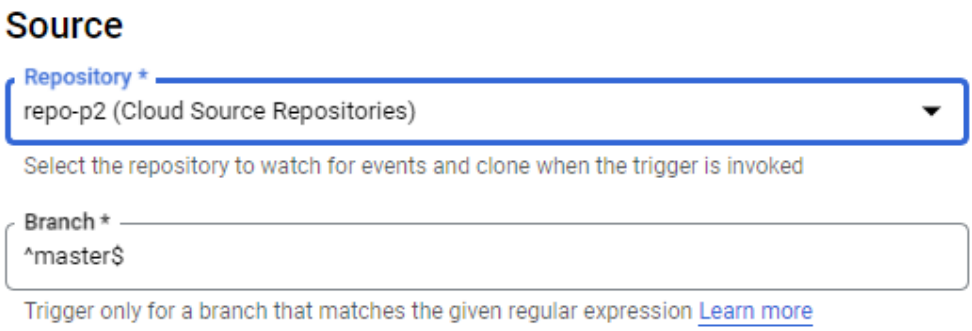
Now push the code to source code repo .



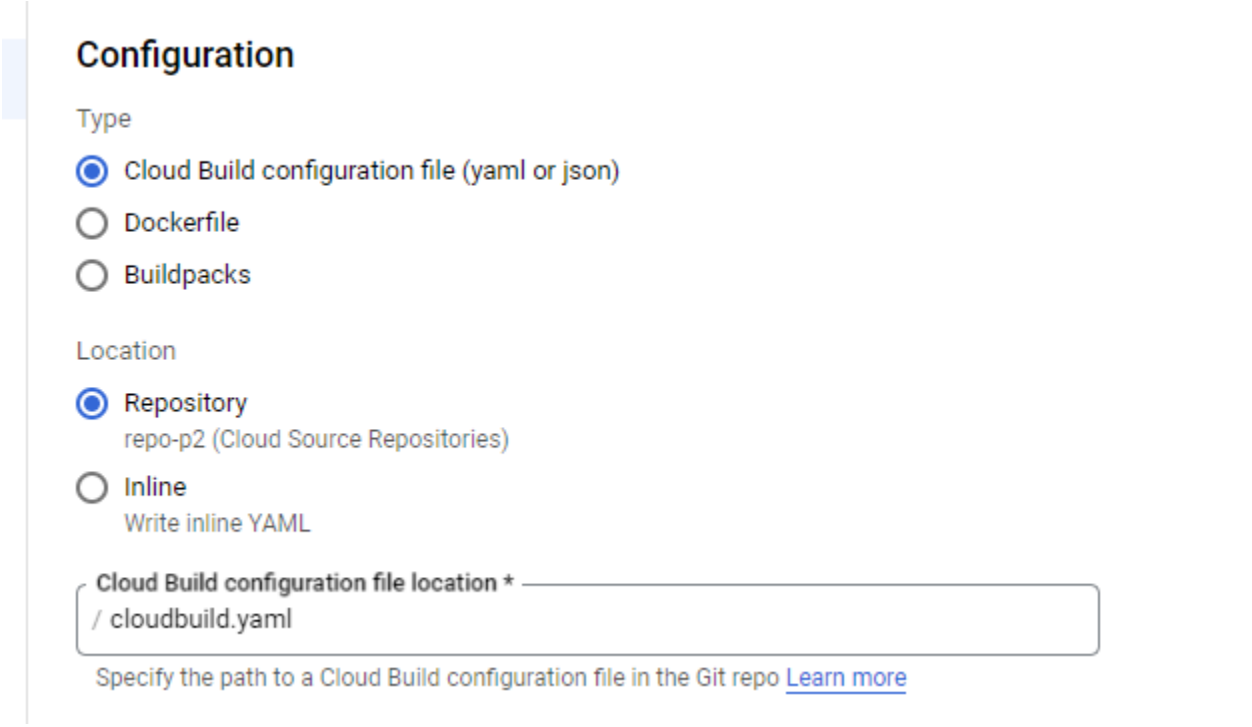
Create trigger :
Select event on which we want trigger (push to branch - for any commit) >



select repo where our code is stored >



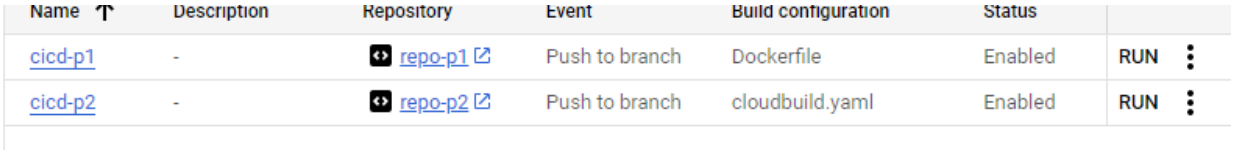
Configuration(select the type of config used in our case cloudbuild.yaml file , we can also dockerfile)> specify location of cloudbuild.yaml file(/path to file/cloudbuild.yaml)



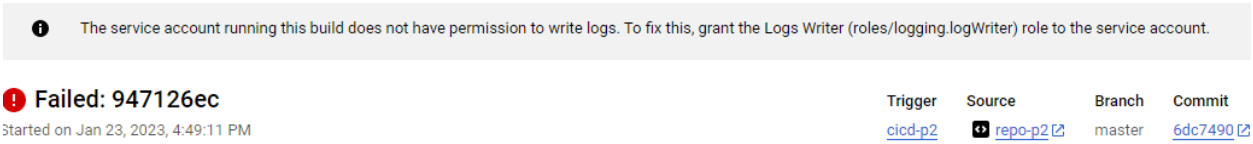
> Approval () if any approval required before trigger then we can select this > service account - create service account and use that - service account should have permission to deploy to app engine (create 1 service account ha)>

Go to iam > create service account > for testing don't apply any role > create >

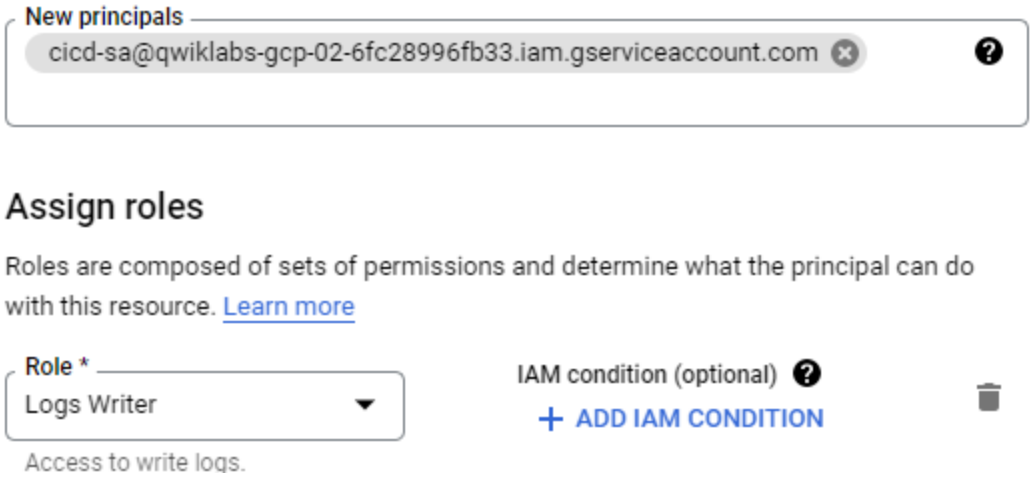
USe above created service account in trigger . >run the cicd 2 . > it will fail - we can check error that is service account don't have permission >



Go to iam > Roles > assign logs writed permission to service account > from add permission .



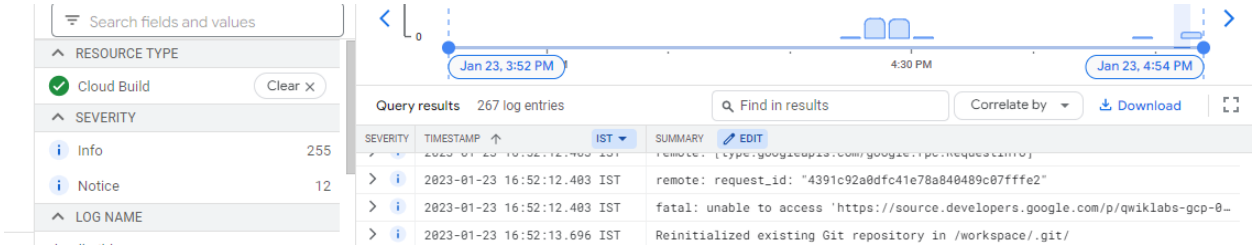
grant the Logs Writer (roles/logging.logWriter) role to the service account.



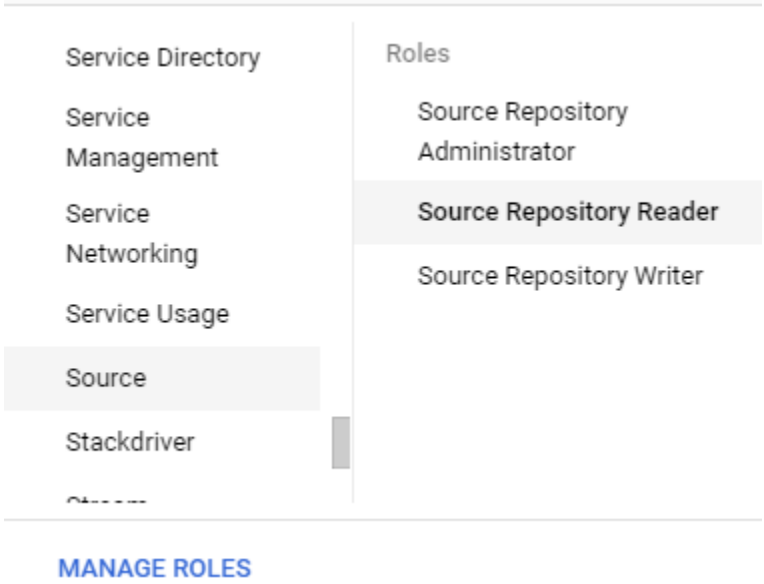
Now run the trigger . We will again get errors .

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} --member \
       serviceAccount:${SERVICE_ACCOUNT_EMAIL} \
       --role "roles/logging.logWriter"
```

For log we can check in cloud logging > Cloud build > check error



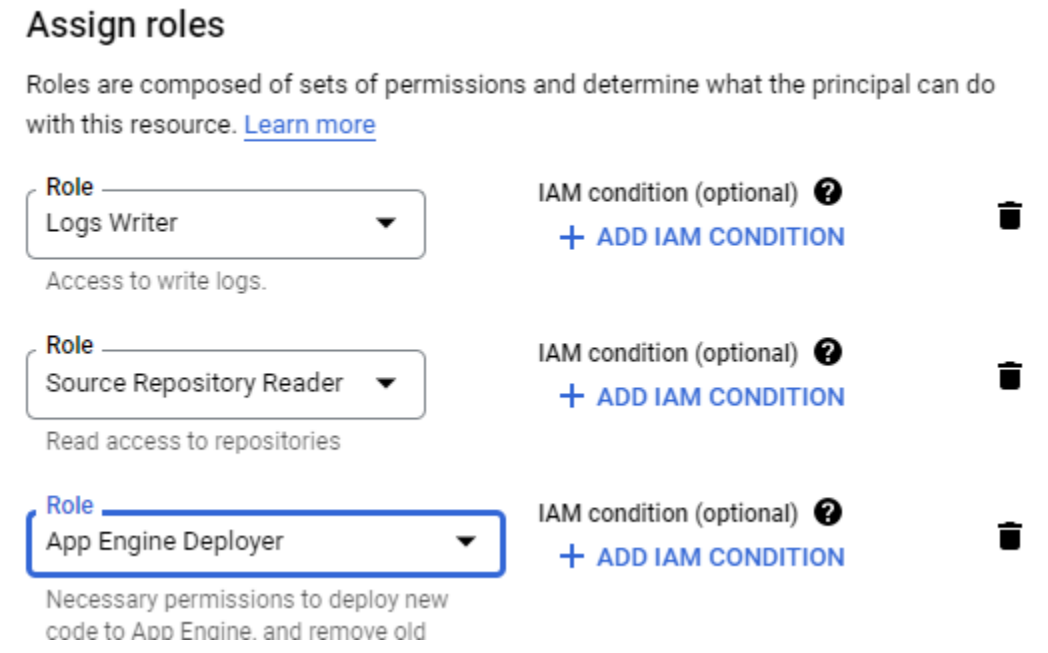
> assign permission for service account > goto iam and edit principal permission >



NOw run the trigger .

Check the error in cloud build and resolve it .

```
textPayload:
"ERROR: (gcloud.app.deploy) Permissions error fetching application [apps/qwiklabs-gcp-02-6fc28996fb33]. Please make sure that you have permission to view applications on the project and that cicd-sa@qwiklabs-gcp-02-6fc28996fb33.iam.gserviceaccount.com has the App Engine Deployer (roles/appengine.deployer) role."
timestamp: "2023-01-23T11:27:50.379479312Z"
}
```



2023-01-23 17:02:43.636 IST	ERROR: (gcloud.app.deploy) The current Google Cloud project [qwiklabs-gcp-02-6fc28996fb33] does not contain an App Engine application. Use `gcloud app create` to initialize an App Engine application within the project.

Open in Logs Explorer

```
{
  insertId: "0203f190-29b4-48ec-91ca-dd57bc05f14b-57"
  labels: {1}
  logName: "projects/qwiklabs-gcp-02-6fc28996fb33/logs/cloudbuild"
  receiveTimestamp: "2023-01-23T11:32:44.217067379Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "ERROR: (gcloud.app.deploy) The current Google Cloud project [qwiklabs-gcp-02-6fc28996fb33] does not contain an App Engine application. Use `gcloud app create` to initialize an App Engine application within the
```

Create an app engine .

2023-01-23 17:24:39.190 IST	ERROR: (gcloud.app.deploy) 403 Could not list bucket [staging.qwiklabs-gcp-02-6fc28996fb33.appspot.com]: cicd-sa@qwiklabs-gcp-02-6fc28996fb33.iam.gserviceaccount.com does not have storage.objects.list access to the Google Cloud Storage bucket. Permission 'storage.objects.list' denied on resource (or it may not exist).

cicd-sa@qwiklabs-gcp-02-6fc28996fb33.iam.gserviceaccount.com	cicd-sa	App Engine Deployer Logs Writer Source Repository Reader Storage Object Admin

```
textPayload:
"ERROR: (gcloud.app.deploy) Error Response: [7] Failed to create cloud build: IAM authority does not have
the permission 'cloudbuild.builds.create' required for action CreateBuild on resource 'projects/qwiklabs-
gcp-02-6fc28996fb33'. Explanation:  Security Context:
RecordingSecurityContext{delegate=ValidatedSecurityContextWithSystemAuthorizationPolicy{delegate=ValidatedSe
curityContextWithRegistryHandle{delegate=ValidatedSecurityContextWithObligations{delegate=ContextWithGaiaMin
tToken{delegate=ValidatedIamSecurityContext{user=gaiauser/0x348f1725e2,
```

prod.iam.gserviceaccount.com		
cicd-sa@qwiklabs-gcp-02-6fc28996fb33.iam.gserviceaccount.com	cicd-sa	App Engine Deployer Cloud Build Editor Logs Writer Source Repository Reader Storage Object Admin

2023-01-23 17:33:58.269 IST	ERROR: Invalid requirement: 'Flask=2.2.2' (from line 1 of requirements.txt)

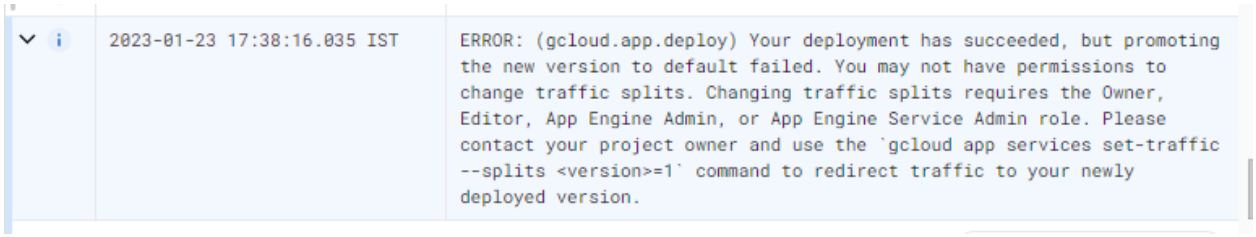
Open in Logs

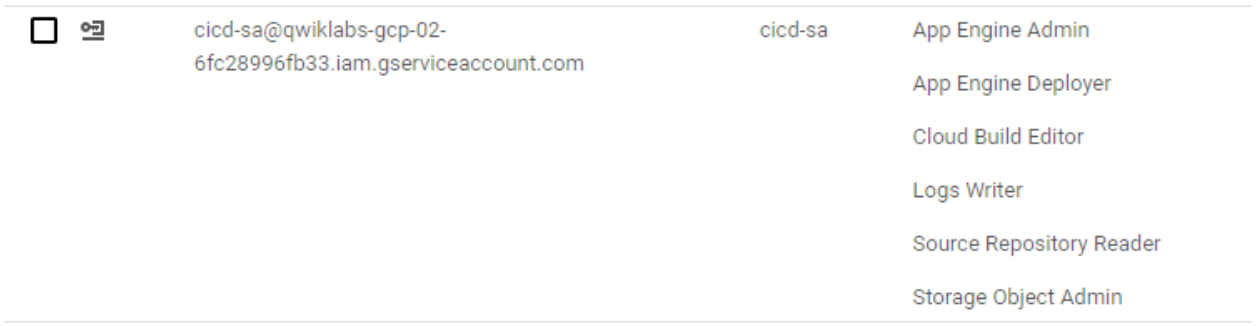
```
{
  insertId: "5ac4ba4d-3df7-4364-9270-021f219898c1-76"
  labels: {1}
  logName: "projects/qwiklabs-gcp-02-6fc28996fb33/logs/cloudbuild"
  receiveTimestamp: "2023-01-23T12:03:59.149634532Z"
  resource: {2}
  severity: "INFO"
  textPayload: "ERROR: Invalid requirement: 'Flask=2.2.2' (from line 1 of requirements.txt)"
  timestamp: "2023-01-23T12:03:58.269427733Z"
}
```

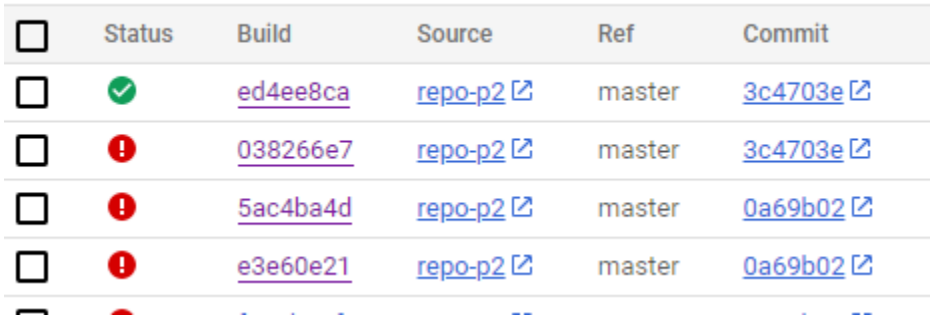
2023-01-23 17:33:58.269 IST	Hint: = is not a valid operator. Did you mean == ?

Update requirements.txt

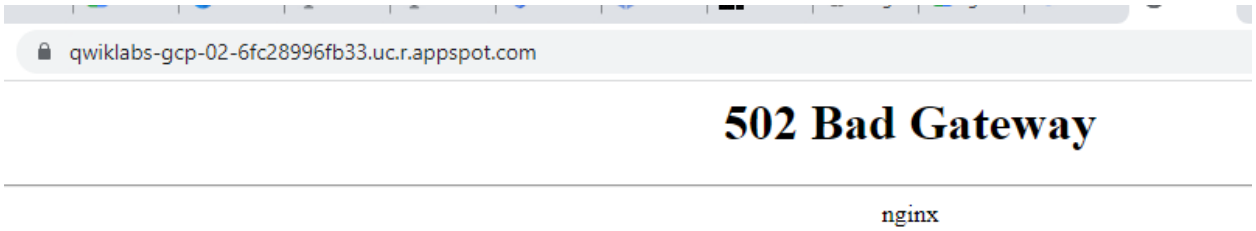
```
Branch 'master' set up to track remote branch 'master' from 'origin'.
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p2-3 (qwiklabs-gcp-02-6fc28996fb33)$ cat requirements.txt
Flask==2.2.2
student_02_621ba9e5fe4f@cloudshell:~/cloudshell_open/repo-p2-3 (qwiklabs-gcp-02-6fc28996fb33)$
```



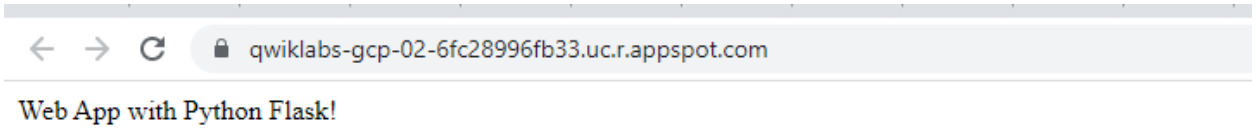




Build success .
Enable app engine admin api
And again run the trigger . We also need to assign app engine admin , logs writer , cloud build service account , service account user permission .
It will work , now we can edit main.py and commit to git the trigger will run automatically.



502 checked main.py had a mistake .
App engine will be updated based on triggers .



We were getting an error that more than 10 instances did not allow deleted old versions in the app engine .
HW : Now do all above manual steps using cloud build .
—----------