How to deploy applications ?
Where do you want to deploy in GCP ?
What is the deployment strategy ?

COMPUTE OPTIONS AVAILABLE :
COmpute engine , Kuberent , App engine , CLoud run , CLoud function .
COMPUTE ENGINE : We install and deploy apps .
KUBERENT AND CLOUD RUN : for deploying containerized apps .
App engine : for web app deployment , app engine is serverless .
Cloud function - event driven .

DEPLOYMENT METHODS :
Blue / green deployment
Rolling deployment
Canary deployment
Traffic splitting deployment

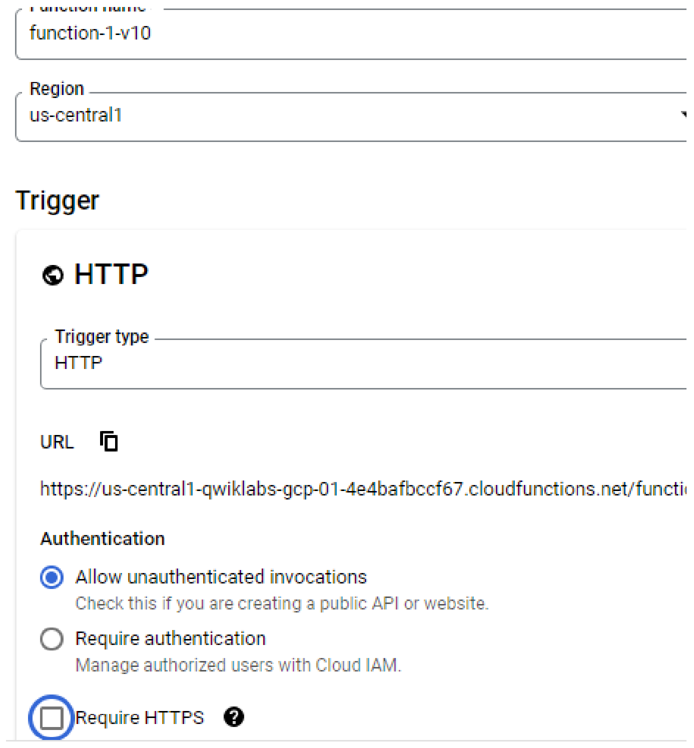
Blue green deployment : We have two copies of the prod system running in parallel .
We have 2 copies with the same version of the prod system . 1 as blue as other green .
So the developer will deploy a new version to the green server . and users will be migrated from blue to green servers . Blue server will become staging server and green will become live server .
This way we keep switching between servers by deploying updates to one server and migrating traffic to that server . In blue green we migrate all users at once . so all users will be affected at one shot if there are any issues . If something doesn't work on the green server then we can switch back to the blue server .

Rolling deployment :
Our app is running on 4 machines with the same version . It will update only 1 server with the new version and we will migrate only 1 user to use this resource . Gradually we will roll out changes to other servers 1 at a tym and migrate a small number of users .
If something fails after we update to the first server then only 1 user will be affected and we can rollback the changes .

CANARY :
Our app is deployed in 8 resources . We update a small % of resources at 1 shot . So from 8 we will update 3 resources with a small number of users at once . Once this is tested we can roll out the deployment to all resources at once .

Traffic splitting :
Small % of users will be served the new version . If everything is fine we will redirect all users to the new version . Useful for A/B testing .

DEPLOY TO GOOGLE CLOUD FUNCTION :
Cloud function is an event driven model .
Cloud fucntion > func-v10 > us central1 > Trigger - http > allow unauthenticated invocation - for public api > require https - uncheck >Runtime > memory - 256 mb > timeout - 60 sec > service account - select default or create new > Connection - allow all traffic > next > deploy code > runtime - python 3.7 > return message - function 1 with version v10 > deploy .



function-1-v10

Region
us-central1

Trigger

HTTP

Trigger type
HTTP

URL

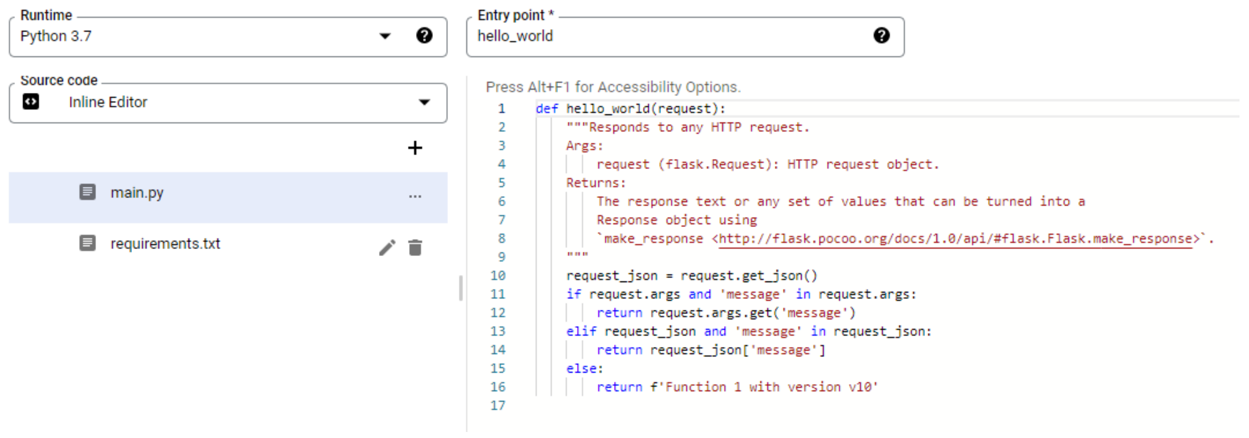
https://us-central1-qwiklabs-gcp-01-4e4bafbccf67.cloudfunctions.net/functi

Authentication

[✓ Allow unauthenticated invocations]
Check this if you are creating a public API or website.

Require authentication
Manage authorized users with Cloud IAM.

Require HTTPS



Runtime
Python 3.7

Entry point *
hello_world

Source code
Inline Editor

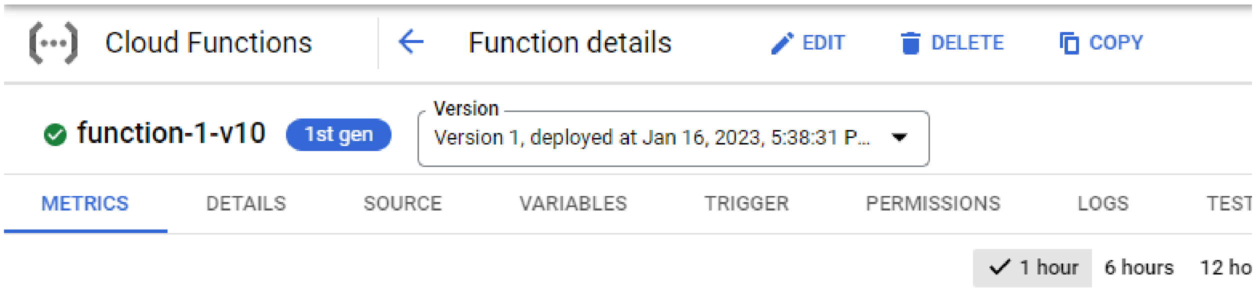
main.py

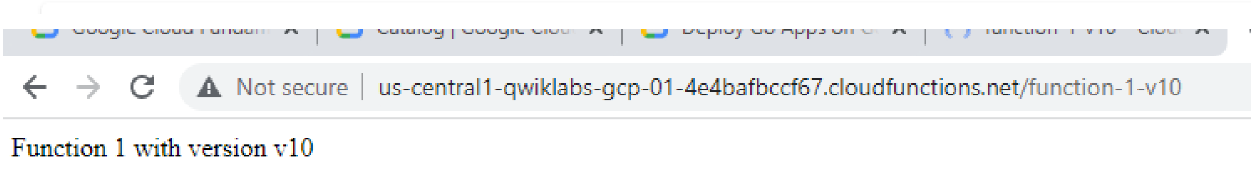
requirements.txt

Press Alt+F1 for Accessibility Options.

```python
def hello_world(request):
    """Responds to any HTTP request.
    Args:
        request (flask.Request): HTTP request object.
    Returns:
        The response text or any set of values that can be turned into a
        Response object using
        `make_response <http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response>`.
    """
    request_json = request.get_json()
    if request.args and 'message' in request.args:
        return request.args.get('message')
    elif request_json and 'message' in request_json:
        return request_json['message']
    else:
        return f'Function 1 with version v10'
```

If we trigger the url it will be working .
Enable cloud build api .
So version v10 is deployed .

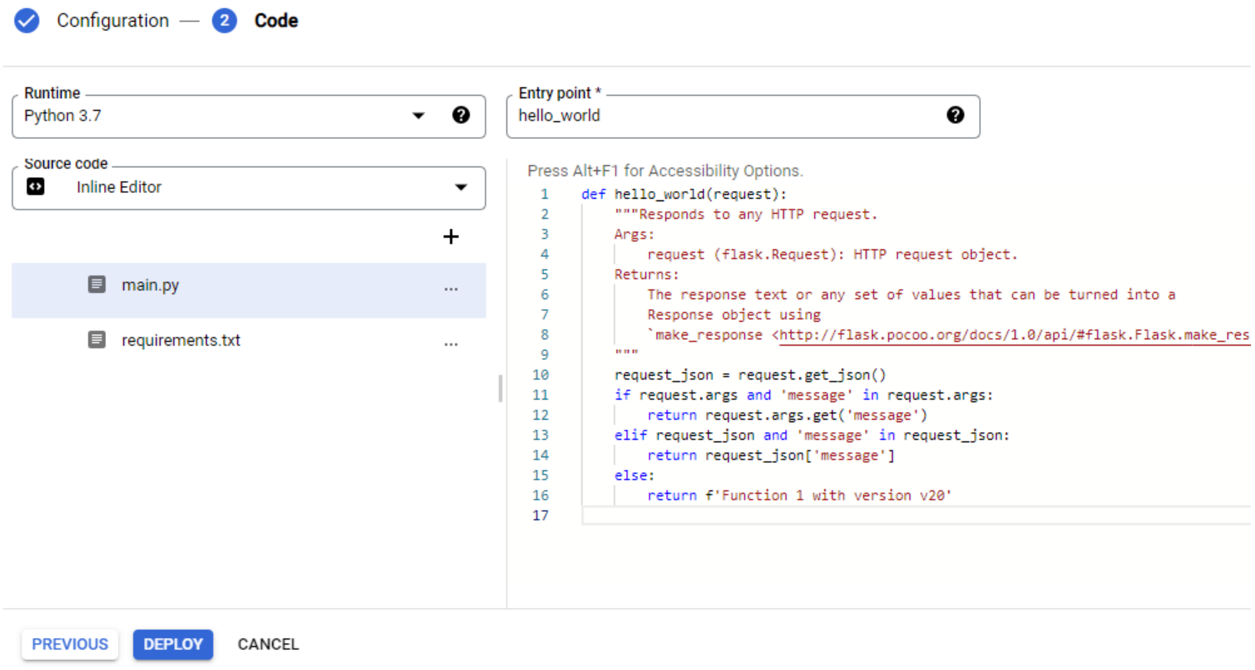



Not secure
us-central1-qwiklabs-gcp-01-4e4bafbccf67.cloudfunctions.net/function-1-v10

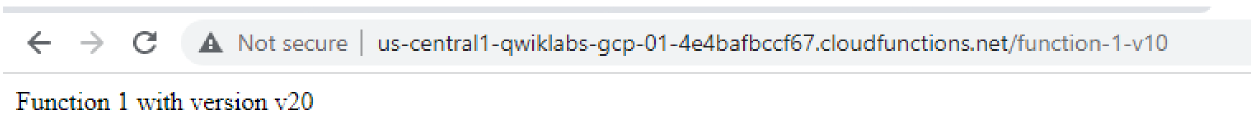
Function 1 with version v10

Let's deploy a new version .

Edit function code - to v2 in the message of code . ANd deploy it ,



```python
def hello_world(request):
    """Responds to any HTTP request.
    Args:
        request (flask.Request): HTTP request object.
    Returns:
        The response text or any set of values that can be turned into a
        Response object using
        `make_response <http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_res
    """
    request_json = request.get_json()
    if request.args and 'message' in request.args:
        return request.args.get('message')
    elif request_json and 'message' in request_json:
        return request_json['message']
    else:
        return f'Function 1 with version v20'

```


Not secure
us-central1-qwiklabs-gcp-01-4e4bafbccf67.cloudfunctions.net/function-1-v10

Function 1 with version v20

it will not create a new function , it will apply changes to the same function only .
Once deployment is done . And trigger url - it will be updated to v2 . So v2 will replace v1 completely . Here we cant rollback to v1 now .
If we want to use both v1 and v2 . Then we have to use 2 functions and internally manage via lb . serve some % of the users with v1 and some with v2 . So this way at dns level we can split

traffic to 2 versions of the function . But in google cloud function we don't have such functionality where 2 versions of 1 function will be stored .

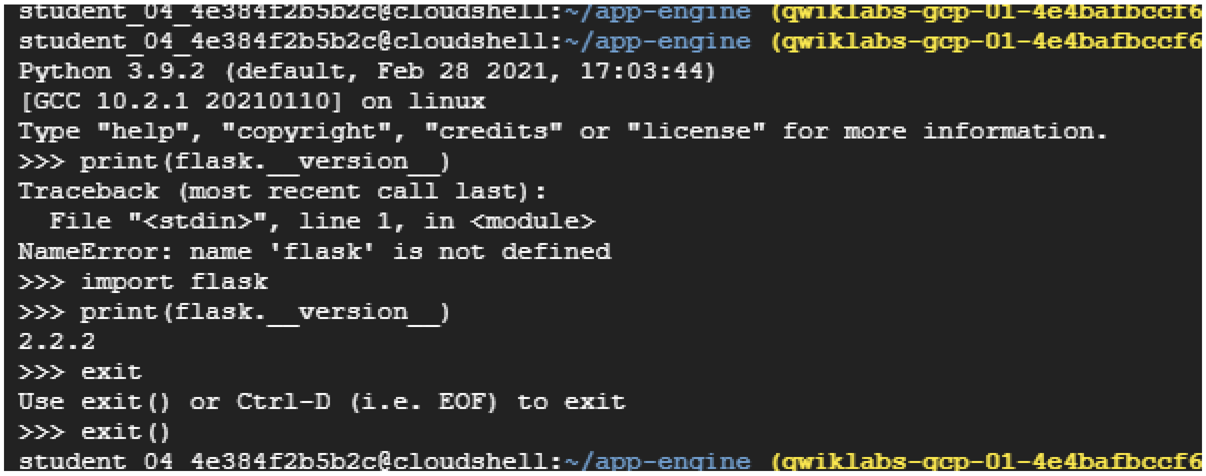
So Cloud Function is mainly used for a single purpose, one single task, one single microservice kind of task when you want to perform based on some kind of event. You can use the Cloud Function.

DEPLOY APP ON GOOGLE APP ENGINE :
If you want to deploy a full-fledged web application, you can go ahead with the App Engine. This is one of the oldest products by Google.
Create a simple python hello world web application .
Create a directory for doing hands on .
Add main.py file ,
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__),

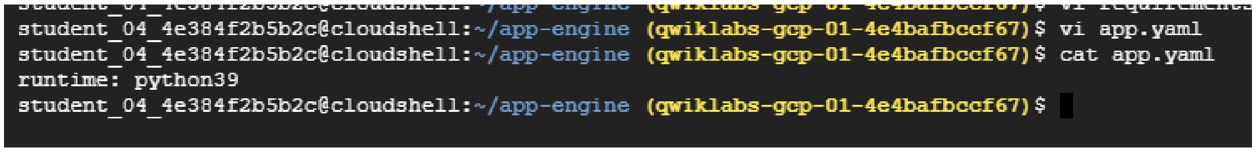
Flask==version.



```
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf6
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf6
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(flask.__version__)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'flask' is not defined
>>> import flask
>>> print(flask.__version__)
2.2.2
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf6
```

We also need an app.yaml file for app engine deployment .

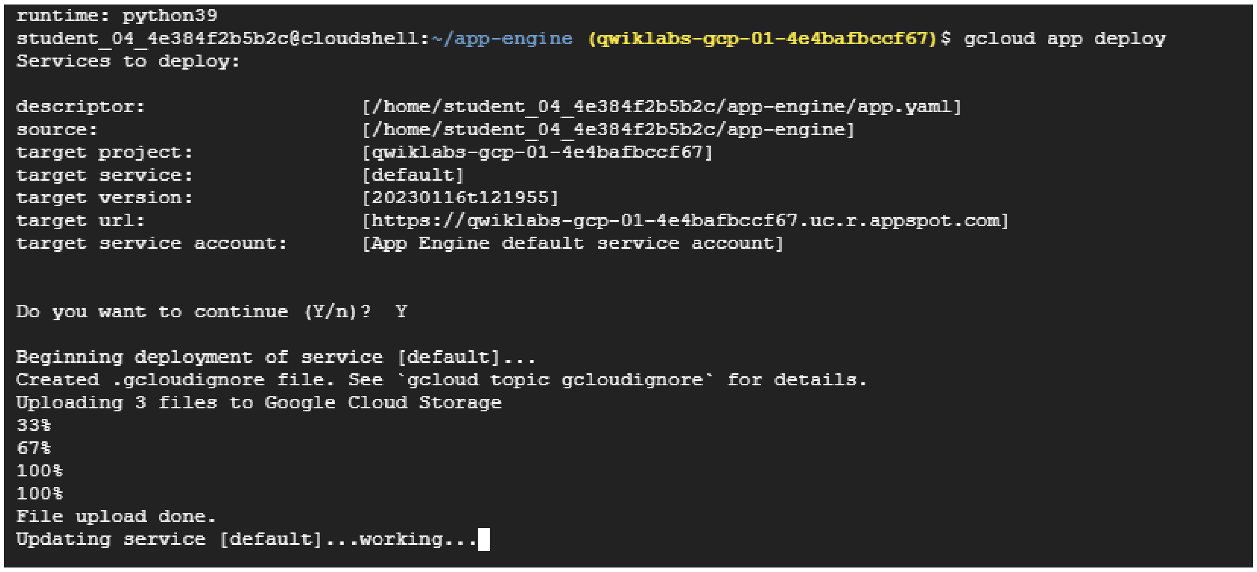
App.yaml :
runtime: python37



```
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ vi app.yaml
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ cat app.yaml
runtime: python39
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$
```

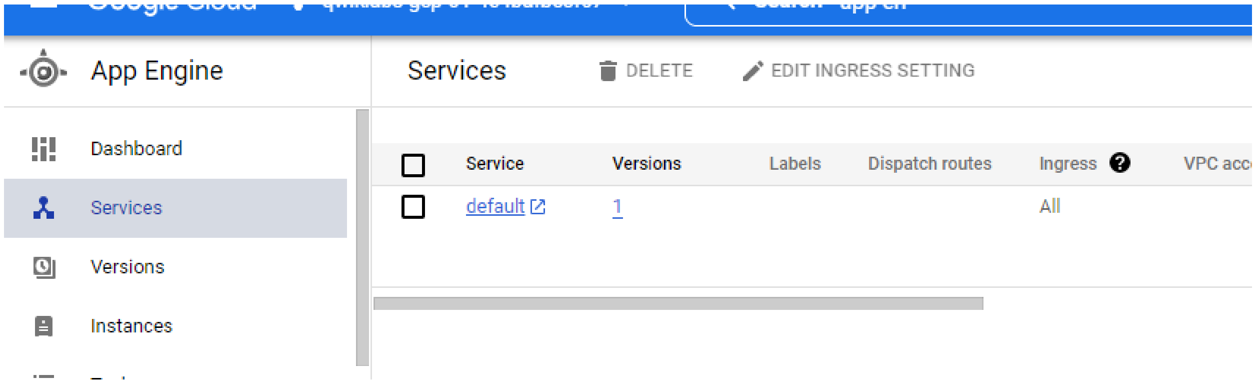
For every single project we can deploy only 1 app engine in gcp .
APp engine > us central > service account - default >

gcloud app deploy - to deploy to the app engine .



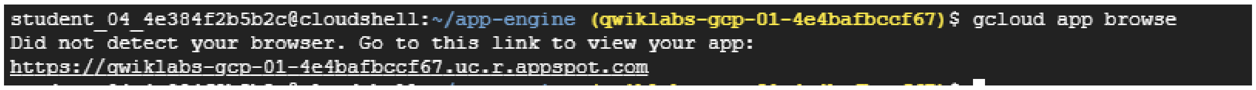
```
runtime: python39
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/student_04_4e384f2b5b2c/app-engine/app.yaml]
source:                      [/home/student_04_4e384f2b5b2c/app-engine]
target project:              [qwiklabs-gcp-01-4e4bafbccf67]
target service:              [default]
target version:              [20230116t121955]
target url:                  [https://qwiklabs-gcp-01-4e4bafbccf67.uc.r.appspot.com]
target service account:      [App Engine default service account]


Do you want to continue (Y/n)?  Y

Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
Uploading 3 files to Google Cloud Storage
33%
67%
100%
100%
File upload done.
Updating service [default]...working...
```

Services will be deployed .

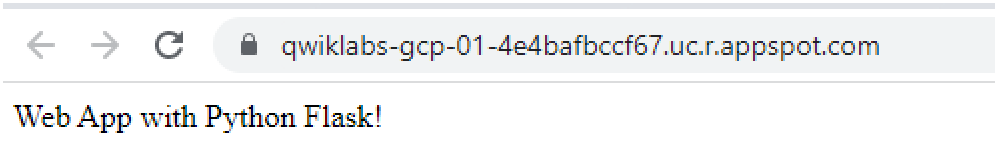


Service	Versions	Labels	Dispatch routes	Ingress	VPC acc
default	1			All	

gcloud app browse - to get the url of the application deployed .

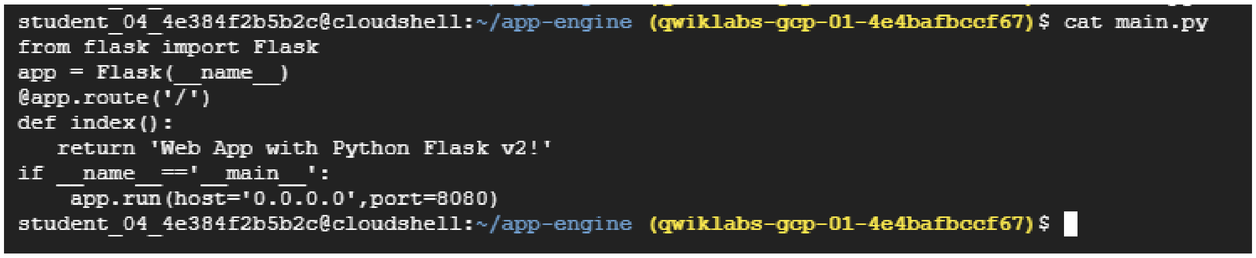


```
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://qwiklabs-gcp-01-4e4bafbccf67.uc.r.appspot.com
```


qwiklabs-gcp-01-4e4bafbccf67.uc.r.appspot.com

Web App with Python Flask!

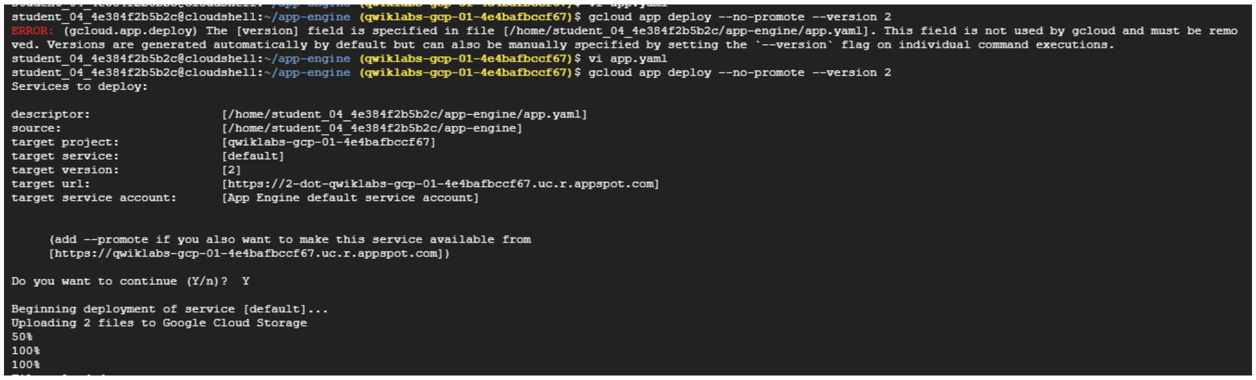
So we can access the 1st version in the browser , let's deploy the 2nd version .
Edit main.py - edit message to v2 .



```
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ cat main.py
from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
   return 'Web App with Python Flask v2!'
if __name__=='__main__':
    app.run(host='0.0.0.0',port=8080)
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$
```

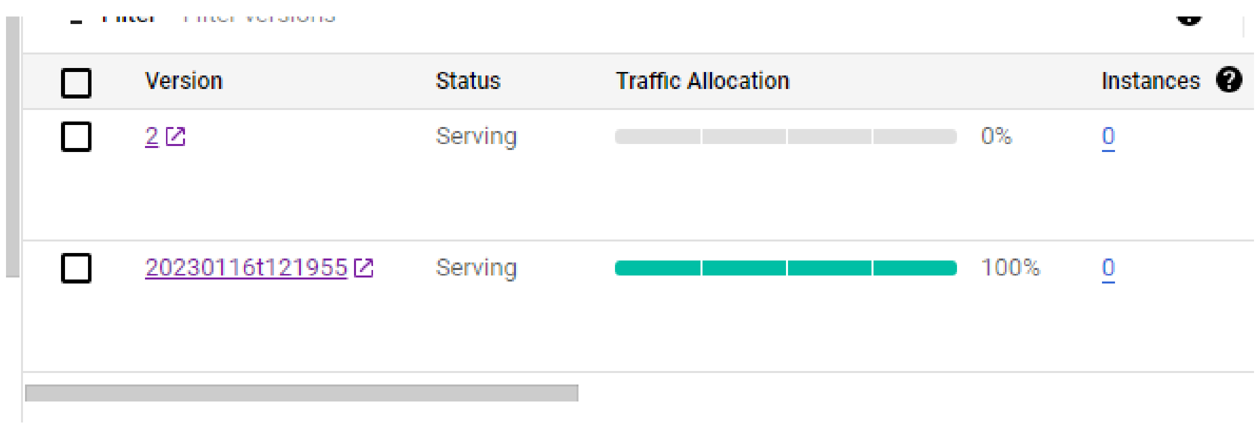
We can define version in app.yaml
version: version

– gcloud app deploy - it will deploy all traffic to version 2 . If we want to deploy without migrating traffic .
gcloud app deploy --no-promote --version 2 - it will deploy a new version but traffic will not be split to the new version .

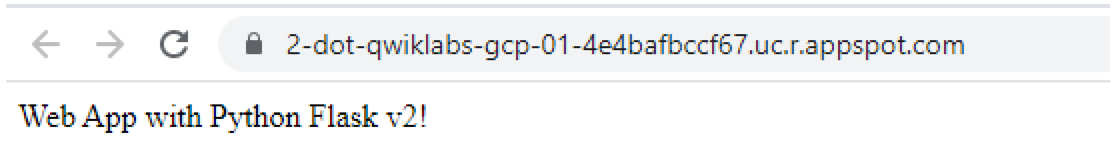


```
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ gcloud app deploy --no-promote --version 2
ERROR: (gcloud.app.deploy) The [version] field is specified in file [/home/student_04_4e384f2b5b2c/app-engine/app.yaml]. This field is not used by gcloud and must be remo
ved. Versions are generated automatically by default but can also be manually specified by setting the `--version` flag on individual command executions.
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ vi app.yaml
student_04_4e384f2b5b2c@cloudshell:~/app-engine (qwiklabs-gcp-01-4e4bafbccf67)$ gcloud app deploy --no-promote --version 2
Services to deploy:

descriptor:                  [/home/student_04_4e384f2b5b2c/app-engine/app.yaml]
source:                      [/home/student_04_4e384f2b5b2c/app-engine]
target project:              [qwiklabs-gcp-01-4e4bafbccf67]
target service:              [default]
target version:              [2]
target url:                  [https://2-dot-qwiklabs-gcp-01-4e4bafbccf67.uc.r.appspot.com]
target service account:      [App Engine default service account]


     (add --promote if you also want to make this service available from
     [https://qwiklabs-gcp-01-4e4bafbccf67.uc.r.appspot.com])

Do you want to continue (Y/n)?  Y

Beginning deployment of service [default]...
Uploading 2 files to Google Cloud Storage
50%
100%
100%
```

Since traffic is not split to v2 , in the url we will access the old version only .
We can access v2 explicitly . We have 2 different urls for 2 versions .



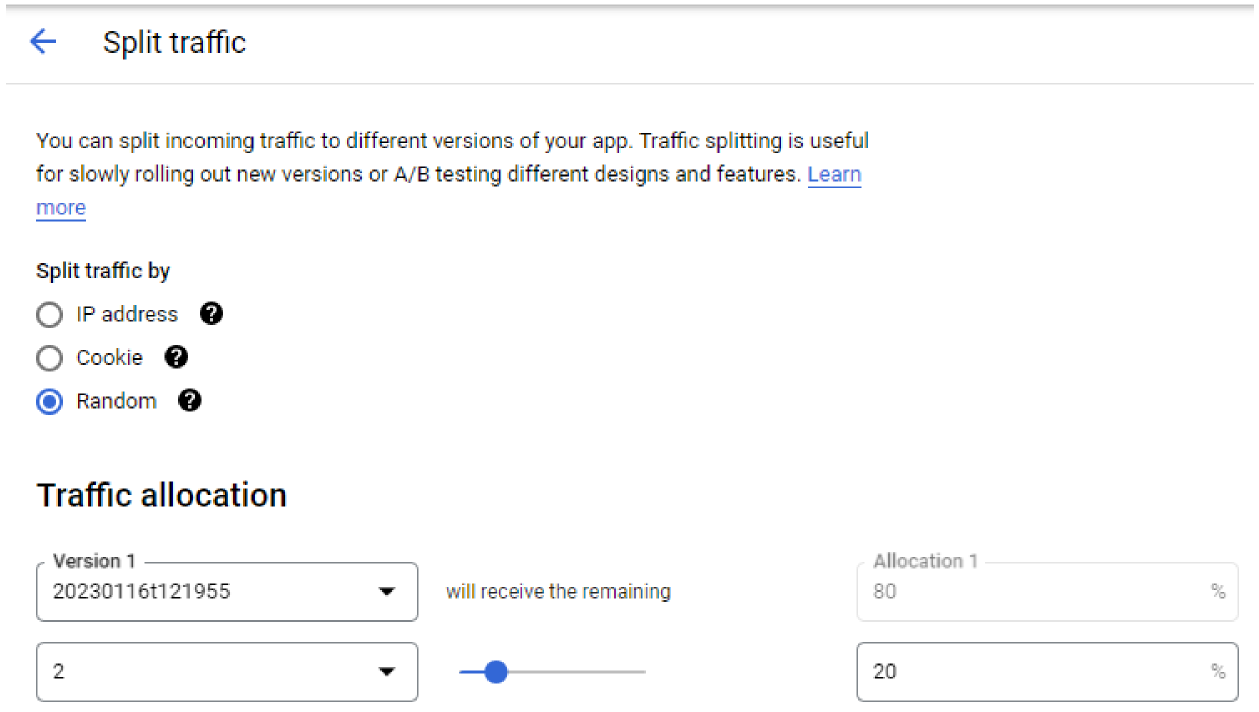
	Version	Status	Traffic Allocation		Instances
☐	2	Serving		0%	0
☐	20230116t121955	Serving		100%	0



2-dot-qwiklabs-gcp-01-4e4bafbccf67.uc.r.appspot.com

Web App with Python Flask v2!

Split traffic by random .- v2 - 20 % allocation . We can check in the url - traffic will split to v2 also for a few times .



Split traffic

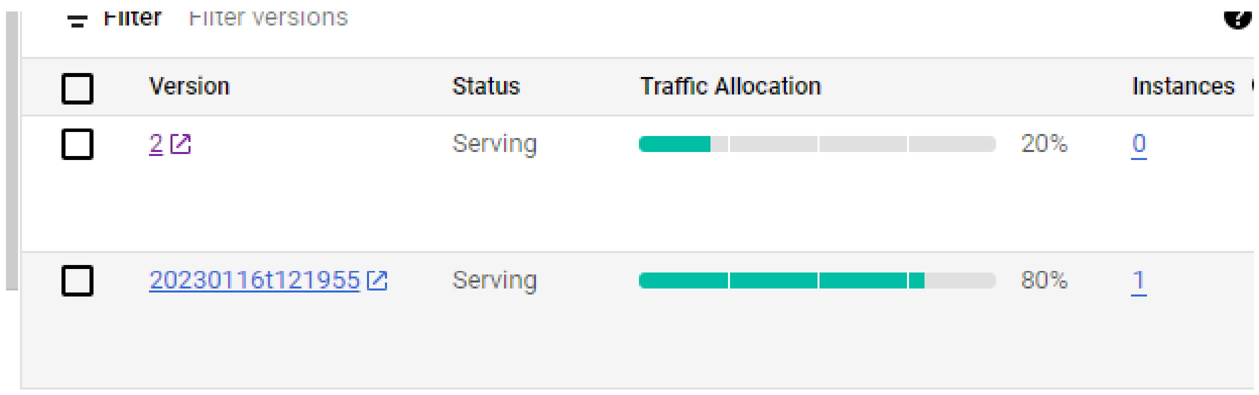
You can split incoming traffic to different versions of your app. Traffic splitting is useful for slowly rolling out new versions or A/B testing different designs and features. Learn more

Split traffic by

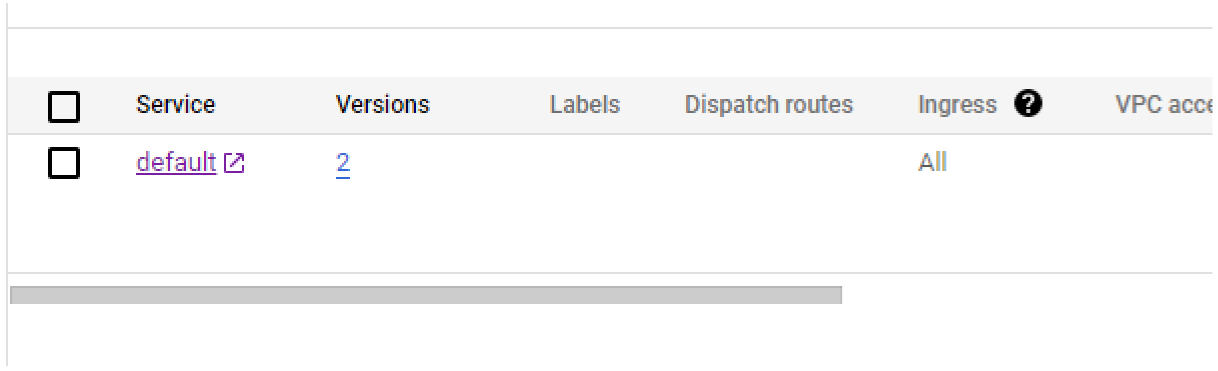
○ IP address
○ Cookie
[✓ Random]

Traffic allocation

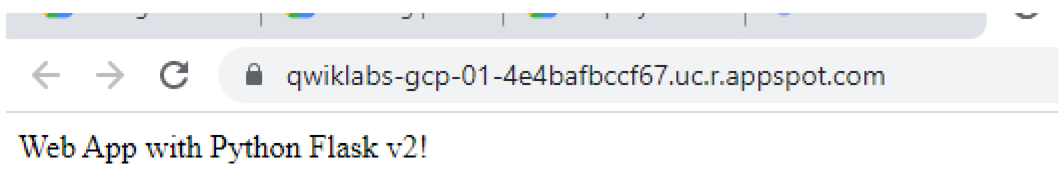
Version 1: 20230116t121955	will receive the remaining	Allocation 1: 80 %
2		20 %



Version	Status	Traffic Allocation	Instances
2	Serving	20%	0
20230116t121955	Serving	80%	1

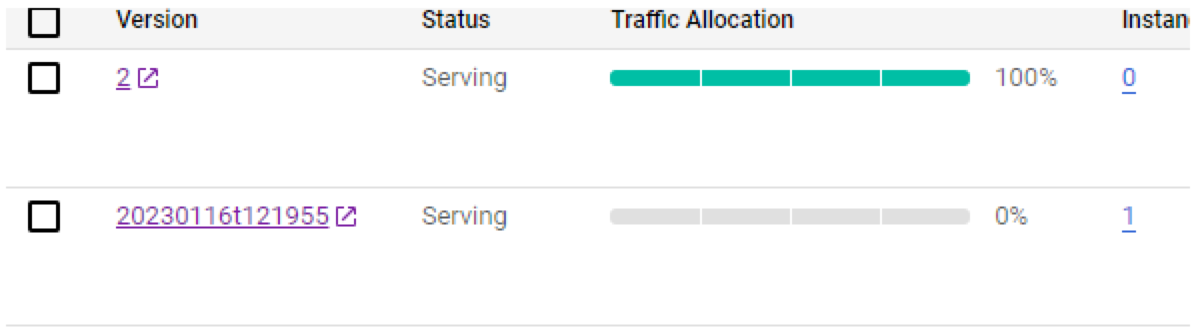


Service	Versions	Labels	Dispatch routes	Ingress	VPC acce
default	2			All	



Web App with Python Flask v2!

If all work fine we can update traffic split to 100 for v2 . In url traffic will be directed to v2 now .



Version	Status	Traffic Allocation	Instan
2	Serving	100%	0
20230116t121955	Serving	0%	1

We can split traffic by - ip address , cookie , random in app engine .
App Engine supports the deployment method like a splitting deployment.
Split your traffic between different versions. So some percentage of users will get from version one, some percentage of users will get version two.
If your spread's satisfied, you can migrate all your traffic to the newest version with all your updated features.