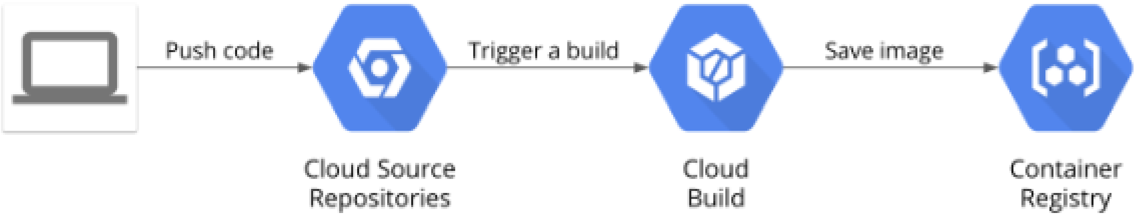
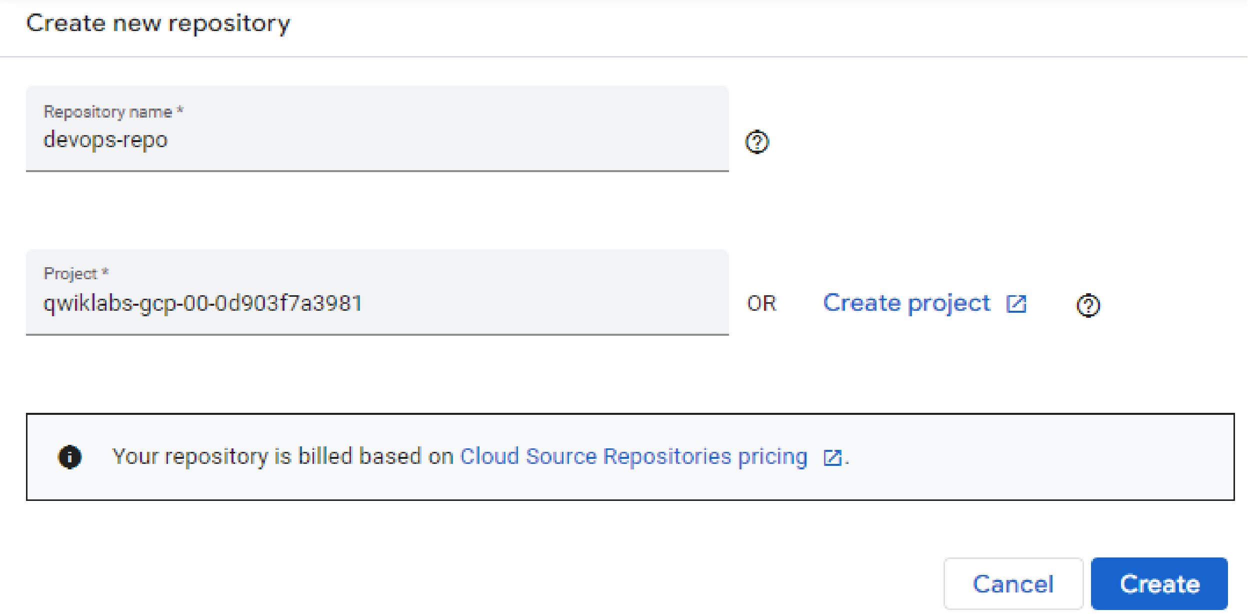
In this lab, you will build a continuous integration pipeline using Cloud Source Repositories, Cloud Build, build triggers, and Container Registry.



In this lab, you will learn how to perform the following tasks:

- Create a Git repository
- Create a simple Python application
- Test Your web application in Cloud Shell
- Define a Docker build
- Manage Docker images with Cloud Build and Container Registry
- Automate builds with triggers
- Test your build changes

First, you will create a Git repository using the Cloud Source Repositories service in Google Cloud. This Git repository will be used to store your source code. Eventually, you will create a build trigger that starts a continuous integration pipeline when code is pushed to it.



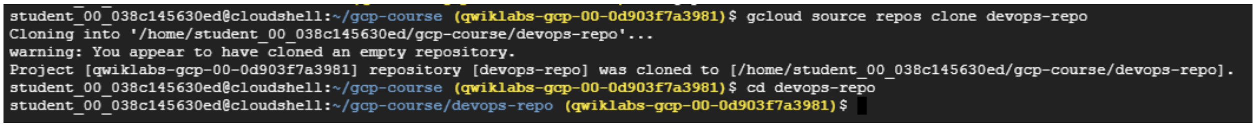
Create new repository

Repository name *
devops-repo

Project *
qwiklabs-gcp-00-0d903f7a3981

Your repository is billed based on Cloud Source Repositories pricing.

11. Now clone the empty repository you just created:



```
student_00_038c145630ed@cloudshell:~/gcp-course (qwiklabs-gcp-00-0d903f7a3981)$ gcloud source repos clone devops-repo
Cloning into '/home/student_00_038c145630ed/gcp-course/devops-repo'...
warning: You appear to have cloned an empty repository.
Project [qwiklabs-gcp-00-0d903f7a3981] repository [devops-repo] was cloned to [/home/student_00_038c145630ed/gcp-course/devops-repo].
student_00_038c145630ed@cloudshell:~/gcp-course (qwiklabs-gcp-00-0d903f7a3981)$ cd devops-repo
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$
```

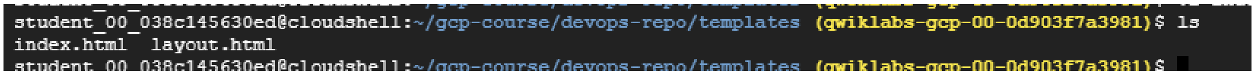
Task 2. Create a simple Python application

You need some source code to manage. So, you will create a simple Python Flask web application. The application will be only slightly better than "hello world", but it will be good enough to test the pipeline you will build.

Create main.py:

```
from flask import Flask, render_template, request
app = Flask(__name__)
@app.route("/")
def main():
   model = {"title": "Hello DevOps Fans."}
   return render_template('index.html', model=model)
if __name__ == "__main__":
   app.run(host='0.0.0.0', port=8080, debug=True, threaded=True)
```

Create dir templates



```
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo/templates (qwiklabs-gcp-00-0d903f7a3981)$ ls
index.html  layout.html
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo/templates (qwiklabs-gcp-00-0d903f7a3981)$
```

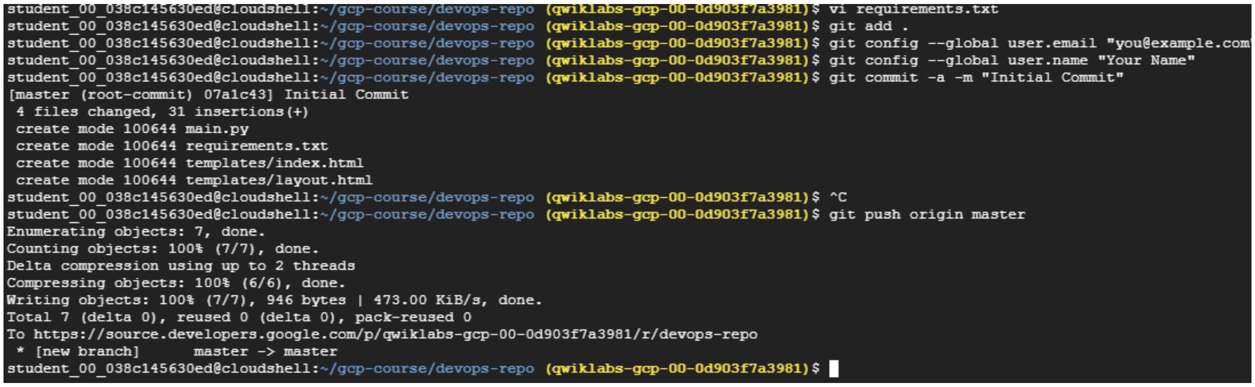
layout.html

```
<!doctype html>
<html lang="en">
<head>
   <title>{{model.title}}</title>
   <!-- Bootstrap CSS -->
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
</head>
<body>
   <div class="container">
      {% block content %}{% endblock %}
      <footer></footer>
   </div>
</body>
</html>
```

```
index.html :
{% extends "layout.html" %}
{% block content %}
<div class="jumbotron">
    <div class="container">
        <h1>{{model.title}}</h1>
    </div>
</div>
{% endblock %}
```

create a **New File** outside template folder and add the following to that file and save it as `requirements.txt`

```
Flask==2.0.3
Now commit all files to git repo .
```



```
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ vi requirements.txt
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git add .
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git config --global user.email "you@example.com
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git config --global user.name "Your Name"
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git commit -a -m "Initial Commit"
[master (root-commit) 07a1c43] Initial Commit
 4 files changed, 31 insertions(+)
 create mode 100644 main.py
 create mode 100644 requirements.txt
 create mode 100644 templates/index.html
 create mode 100644 templates/layout.html
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ ^C
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 946 bytes | 473.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://source.developers.google.com/p/qwiklabs-gcp-00-0d903f7a3981/r/devops-repo
 * [new branch]      master -> master
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ 
```

Task 3. Define a Docker build

The first step to using Docker is to create a file called **Dockerfile**. This file defines how a Docker container is constructed. You will do that now.

The file *Dockerfile* is used to define how the container is built.

2. At the top of the file, enter the following:

```
FROM python:3.7
```

This is the base image. You could choose many base images. In this case, you are using one with Python already installed on it.

3. Enter the following:

```
WORKDIR /app
COPY . .
```

These lines copy the source code from the current folder into the /app folder in the container image.

4. Enter the following:

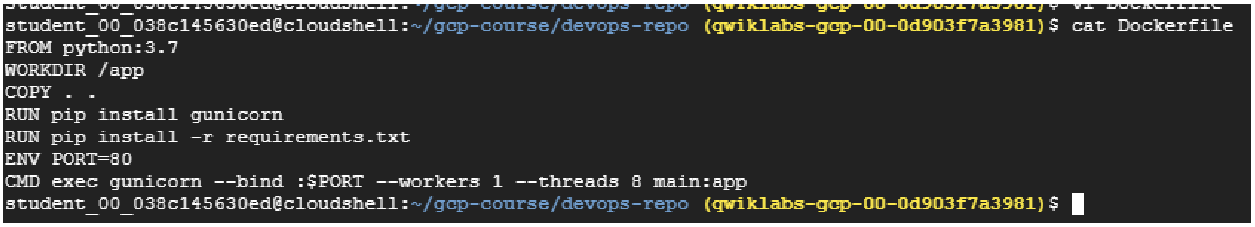
```
RUN pip install gunicorn
RUN pip install -r requirements.txt
```

This uses pip to install the requirements of the Python application into the container. Gunicorn is a Python web server that will be used to run the web app.

5. Enter the following:

```
ENV PORT=80
CMD exec gunicorn --bind :$PORT --workers 1 --threads 8 main:app
```

The environment variable sets the port that the application will run on (in this case, 80). The last line runs the web app using the gunicorn web server.



```
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ vi Dockerfile
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ cat Dockerfile
FROM python:3.7
WORKDIR /app
COPY . .
RUN pip install gunicorn
RUN pip install -r requirements.txt
ENV PORT=80
CMD exec gunicorn --bind :$PORT --workers 1 --threads 8 main:app
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$
```

Task 4. Manage Docker images with Cloud Build and Container Registry

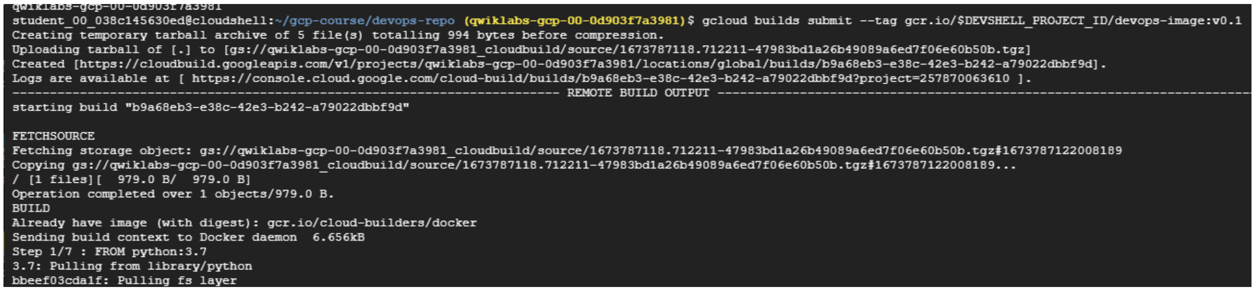
The Docker image has to be built and then stored somewhere. You will use **Cloud Build** and **Container Registry**.

2. The Cloud Shell environment variable DEVSHELL_PROJECT_ID automatically has your current project ID stored. The project ID is required to store images in the Container Registry. Enter the following command to view your project ID:

```
echo $DEVSHELL_PROJECT_ID
```

3. Enter the following command to use Cloud Build to build your image:

```
gcloud builds submit --tag gcr.io/$DEVSHELL_PROJECT_ID/devops-image:v0.1 .
```



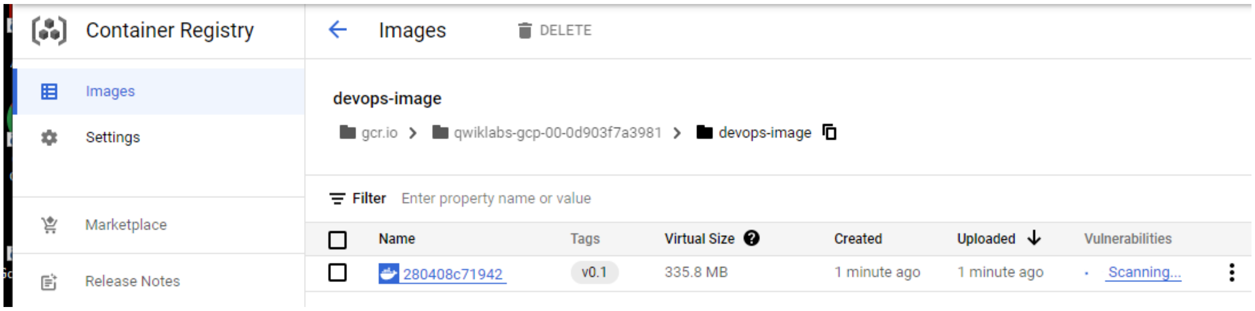
```
qwiklabs-gcp-00-0d903f7a3981
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ gcloud builds submit --tag gcr.io/$DEVSHELL_PROJECT_ID/devops-image:v0.1
Creating temporary tarball archive of 5 file(s) totalling 994 bytes before compression.
Uploading tarball of [.] to [gs://qwiklabs-gcp-00-0d903f7a3981_cloudbuild/source/1673787118.712211-47983bd1a26b49089a6ed7f06e60b50b.tgz]
Created [https://cloudbuild.googleapis.com/v1/projects/qwiklabs-gcp-00-0d903f7a3981/locations/global/builds/b9a68eb3-e38c-42e3-b242-a79022dbbf9d].
Logs are available at [ https://console.cloud.google.com/cloud-build/builds/b9a68eb3-e38c-42e3-b242-a79022dbbf9d?project=257870063610 ].
------------------------------------------------------------------------- REMOTE BUILD OUTPUT -------------------------------------------------------------------------
starting build "b9a68eb3-e38c-42e3-b242-a79022dbbf9d"

FETCHSOURCE
Fetching storage object: gs://qwiklabs-gcp-00-0d903f7a3981_cloudbuild/source/1673787118.712211-47983bd1a26b49089a6ed7f06e60b50b.tgz#1673787122008189
Copying gs://qwiklabs-gcp-00-0d903f7a3981_cloudbuild/source/1673787118.712211-47983bd1a26b49089a6ed7f06e60b50b.tgz#1673787122008189...
/ [1 files][  979.0 B/  979.0 B]
Operation completed over 1 objects/979.0 B.
BUILD
Already have image (with digest): gcr.io/cloud-builders/docker
Sending build context to Docker daemon  6.656kB
Step 1/7 : FROM python:3.7
3.7: Pulling from library/python
bbeef03cda1f: Pulling fs layer
```

The image will be stored in Container Registry.

Note: In Container Registry, the image name always begins with **`gcr.io/`**, followed by the project ID of the project you are working in, followed by the image name and version.

The period at the end of the command represents the path to the Dockerfile: in this case, the current directory.

Container Registry. Your image should be on the list.

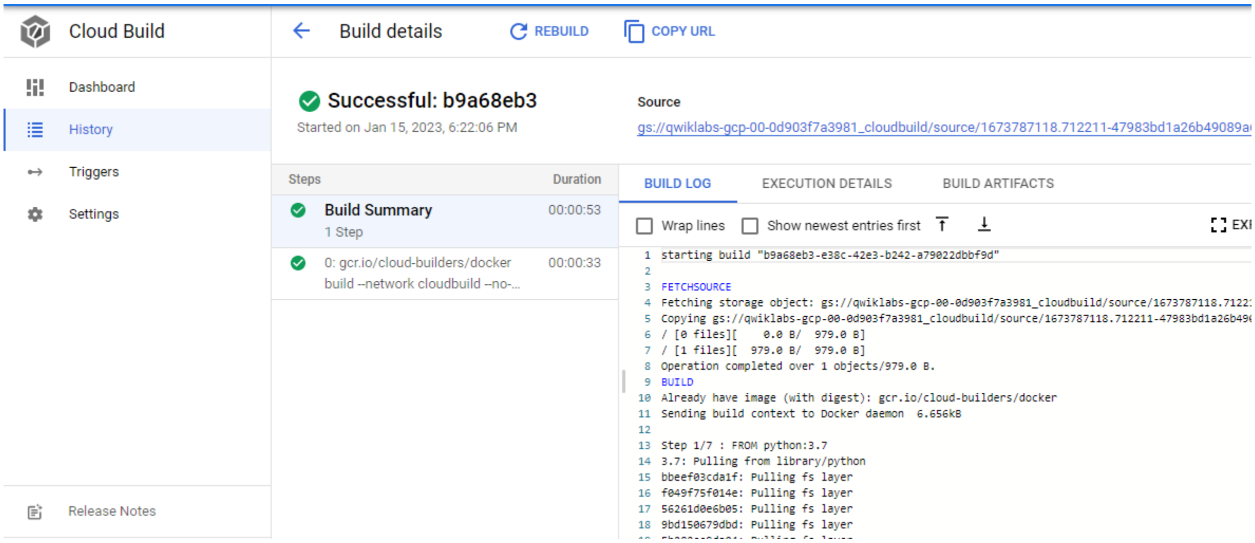


devops-image

gcr.io > qwiklabs-gcp-00-0d903f7a3981 > devops-image

□	Name	Tags	Virtual Size	Created	Uploaded	Vulnerabilities
□	280408c71942	v0.1	335.8 MB	1 minute ago	1 minute ago	Scanning...

5. Now navigate to the **Cloud Build** service, and your build should be listed in the history.



Successful: b9a68eb3

Started on Jan 15, 2023, 6:22:06 PM

Source

gs://qwiklabs-gcp-00-0d903f7a3981_cloudbuild/source/1673787118.712211-47983bd1a26b49089a

Steps	Duration
Build Summary 1 Step	00:00:53
0: gcr.io/cloud-builders/docker build --network cloudbuild --no-...	00:00:33

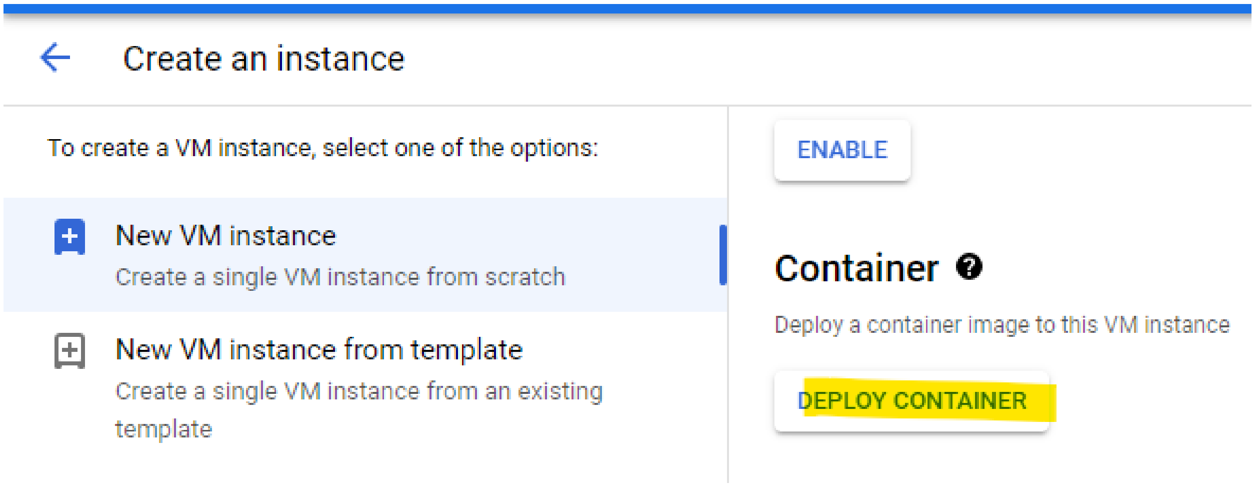
```
starting build "b9a68eb3-e38c-42e3-b242-a79022dbbf9d"

FETCHSOURCE
Fetching storage object: gs://qwiklabs-gcp-00-0d903f7a3981_cloudbuild/source/1673787118.7122
Copying gs://qwiklabs-gcp-00-0d903f7a3981_cloudbuild/source/1673787118.712211-47983bd1a26b49
/ [0 files][    0.0 B/  979.0 B]
/ [1 files][  979.0 B/  979.0 B]
Operation completed over 1 objects/979.0 B.
BUILD
Already have image (with digest): gcr.io/cloud-builders/docker
Sending build context to Docker daemon  6.656kB

Step 1/7 : FROM python:3.7
3.7: Pulling from library/python
bbeef03cda1f: Pulling fs layer
f049f75f014e: Pulling fs layer
56261d0e6b05: Pulling fs layer
9bd150679dbd: Pulling fs layer
```

You will now try running this image from a Compute Engine virtual machine.

7. Navigate to the **Compute Engine** service.
8. Click **Create Instance** to create a VM.
9. On the **Create an instance** page, specify the following, and leave the remaining settings as their defaults:

Property	Value
Container	Click DEPLOY CONTAINER
Container image	`gcr.io/<your-project-id-here>/devops-image:v0.1` (change the project ID where indicated) and click SELECT
Firewall	Allow HTTP traffic

10. Click **Create**.



Create an instance

To create a VM instance, select one of the options:

New VM instance

Create a single VM instance from scratch

New VM instance from template

Create a single VM instance from an existing template

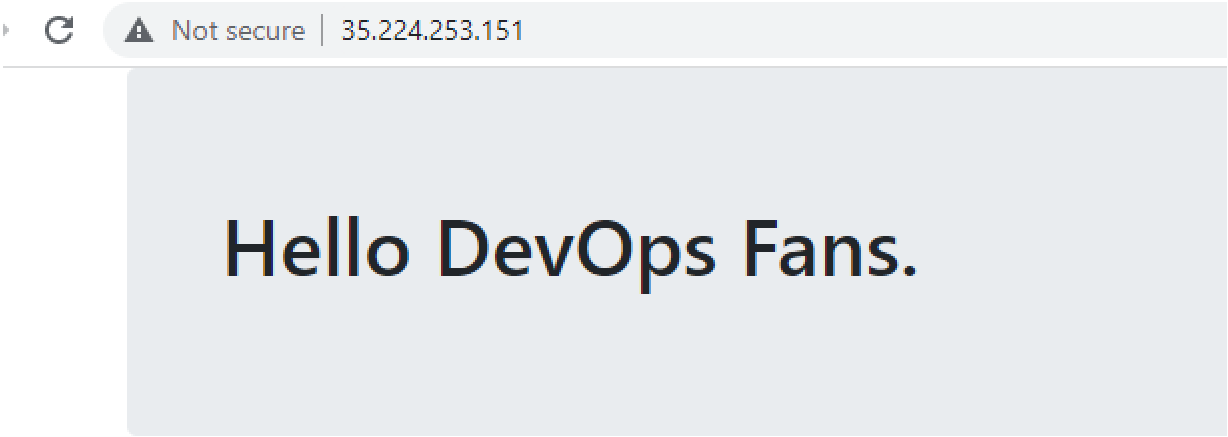
ENABLE

Container

Deploy a container image to this VM instance

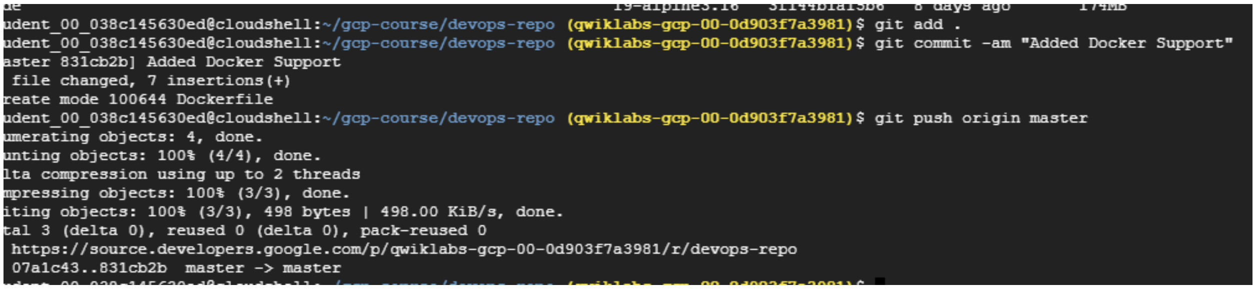
DEPLOY CONTAINER

10. Once the VM starts, create a browser tab and make a request to this new VM's external IP address. The program should work as before.



Hello DevOps Fans.

14. Push your changes to Cloud Source Repositories:

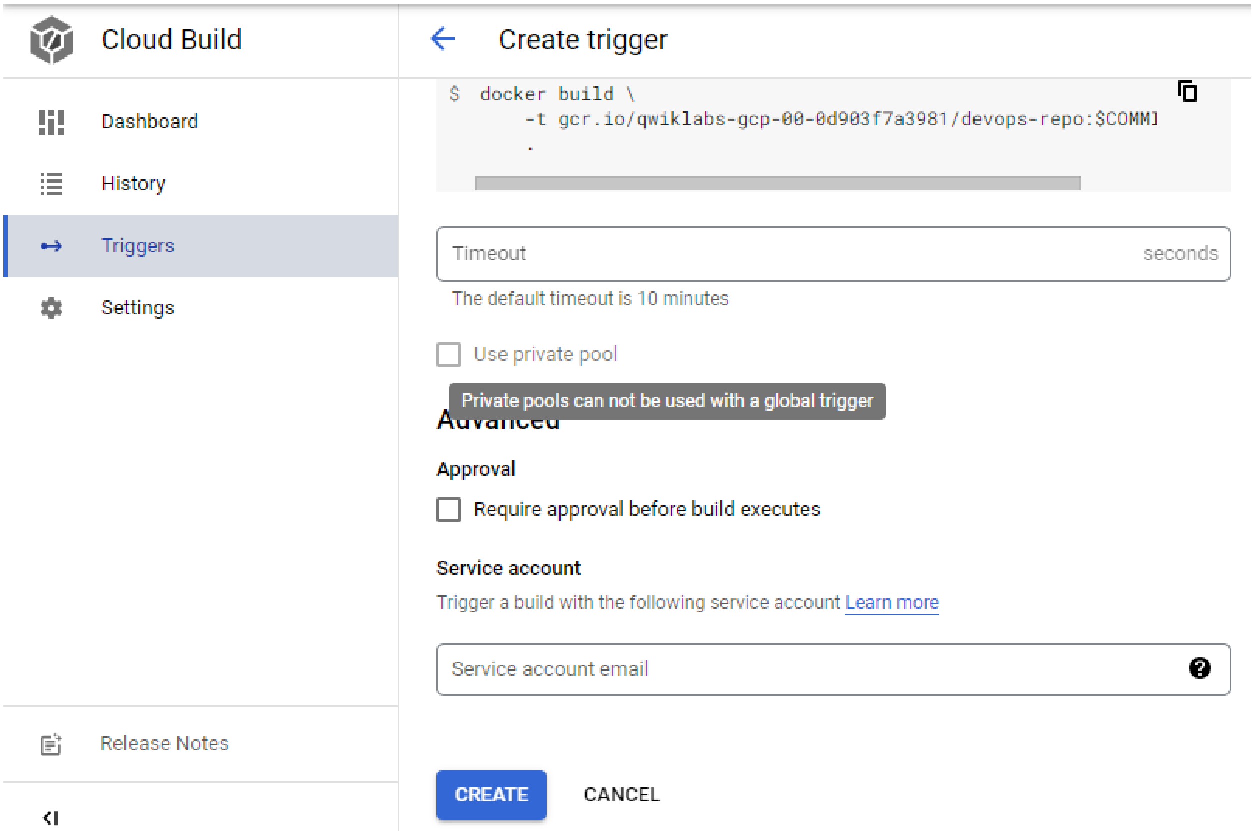


```
udent_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git add .
udent_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git commit -am "Added Docker Support"
aster 831cb2b] Added Docker Support
 file changed, 7 insertions(+)
reate mode 100644 Dockerfile
udent_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git push origin master
umerating objects: 4, done.
unting objects: 100% (4/4), done.
lta compression using up to 2 threads
mpressing objects: 100% (3/3), done.
iting objects: 100% (3/3), 498 bytes | 498.00 KiB/s, done.
tal 3 (delta 0), reused 0 (delta 0), pack-reused 0
 https://source.developers.google.com/p/qwiklabs-gcp-00-0d903f7a3981/r/devops-repo
 07a1c43..831cb2b  master -> master
```

Task 5. Automate builds with triggers

1. Click **Create trigger**.
2. Name the trigger `devops-trigger`.
3. Select your **devops-repo** Git repository under repository dropdown.
4. Select **.*(any branch)** for the branch.
5. Choose **Dockerfile** for **Configuration** and select the default image.

6. Accept the rest of the defaults, and click **Create**.



```
$ docker build \
    -t gcr.io/qwiklabs-gcp-00-0d903f7a3981/devops-repo:$COMM]
    .
```

Timeout seconds

The default timeout is 10 minutes

□ Use private pool

Private pools can not be used with a global trigger

Advanced

Approval

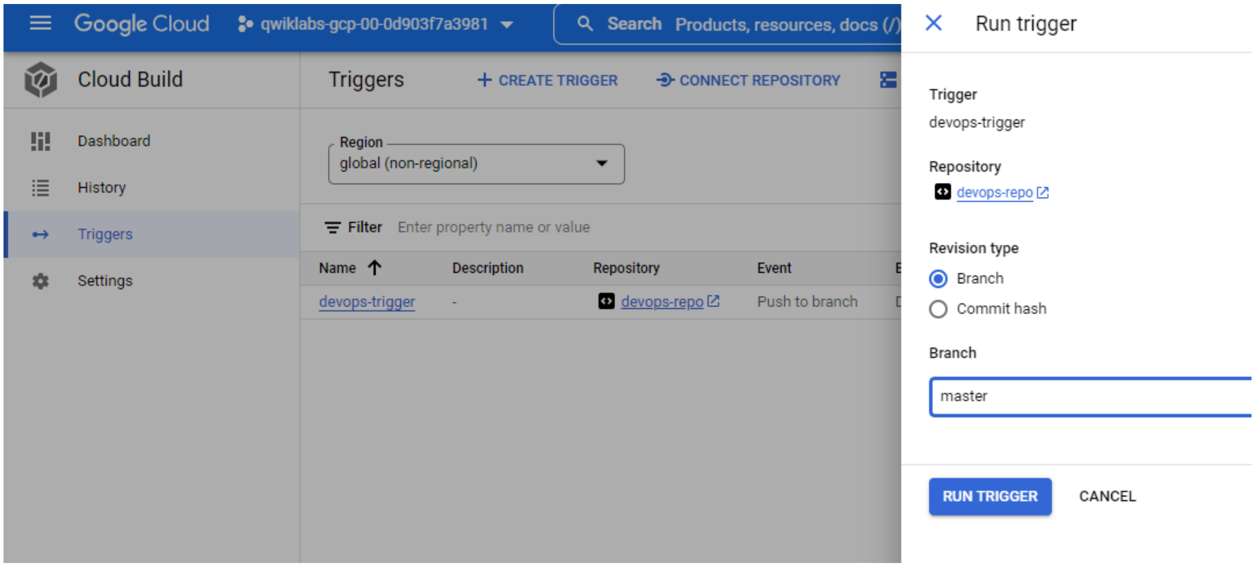
□ Require approval before build executes

Service account

Trigger a build with the following service account Learn more

Service account email

7. To test the trigger, click **Run** and then **Run trigger**.



Region: global (non-regional)

Name	Description	Repository	Event
devops-trigger	-	devops-repo	Push to branch

Trigger: devops-trigger

Repository: devops-repo

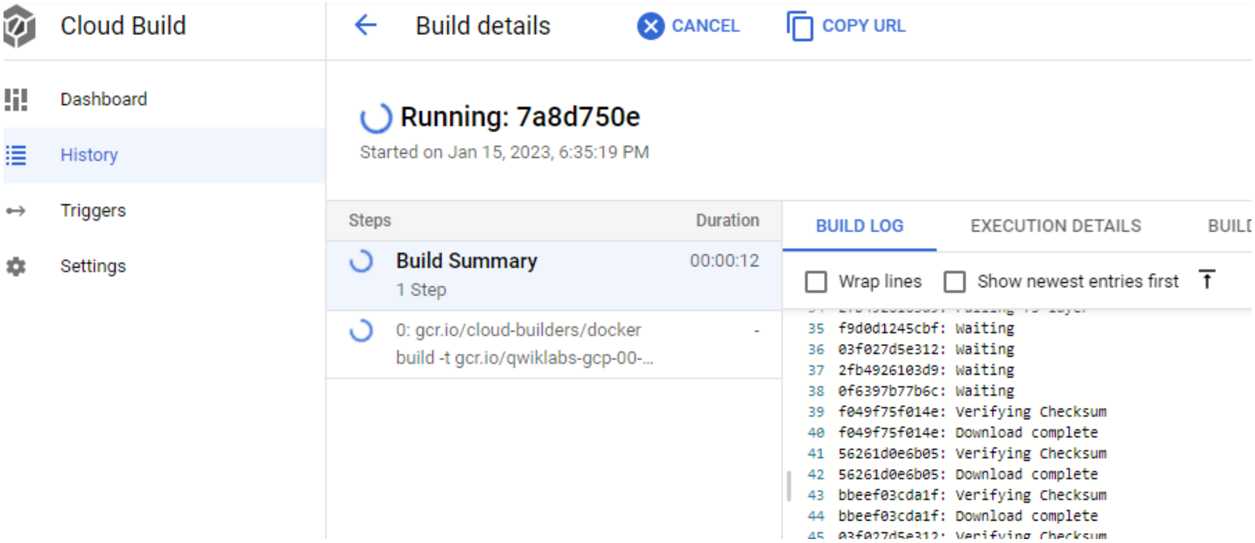
Revision type

● Branch

○ Commit hash

Branch: master

8. Click the **History** link and you should see a build running. Wait for the build to finish, and then click the link to it to see its details.



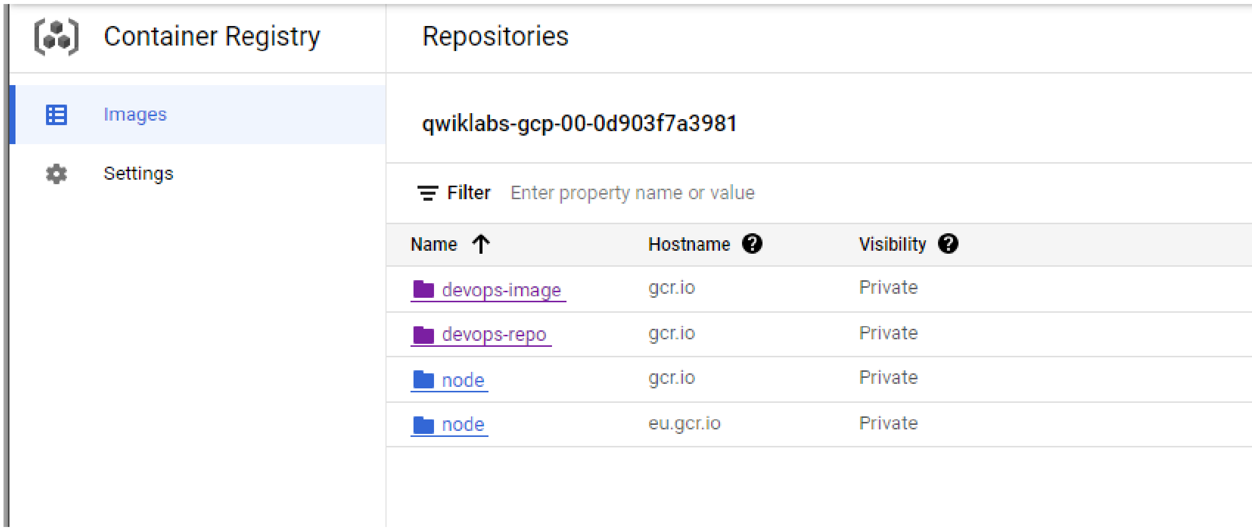
Running: 7a8d750e
Started on Jan 15, 2023, 6:35:19 PM

Steps	Duration
Build Summary 1 Step	00:00:12
0: gcr.io/cloud-builders/docker build -t gcr.io/qwiklabs-gcp-00-...	-

□ Wrap lines □ Show newest entries first

```
f9d0d1245cbf: Waiting
03f027d5e312: Waiting
2fb4926103d9: Waiting
0f6397b77b6c: Waiting
f049f75f014e: Verifying Checksum
f049f75f014e: Download complete
56261d0e6b05: Verifying Checksum
56261d0e6b05: Download complete
bbeef03cda1f: Verifying Checksum
bbeef03cda1f: Download complete
03f027d5e312: Verifying Checksum
```

9. Scroll down and look at the logs. The output of the build here is what you would have seen if you were running it on your machine.
10. Return to the Container Registry service. You should see a new folder, **devops-repo**, with a new image in it.

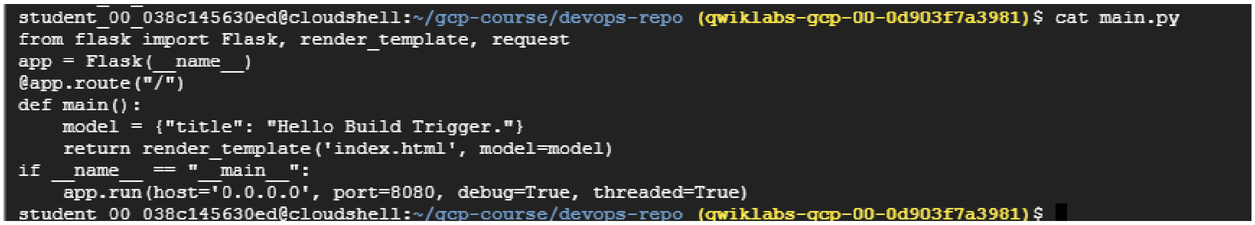


qwiklabs-gcp-00-0d903f7a3981

Filter Enter property name or value

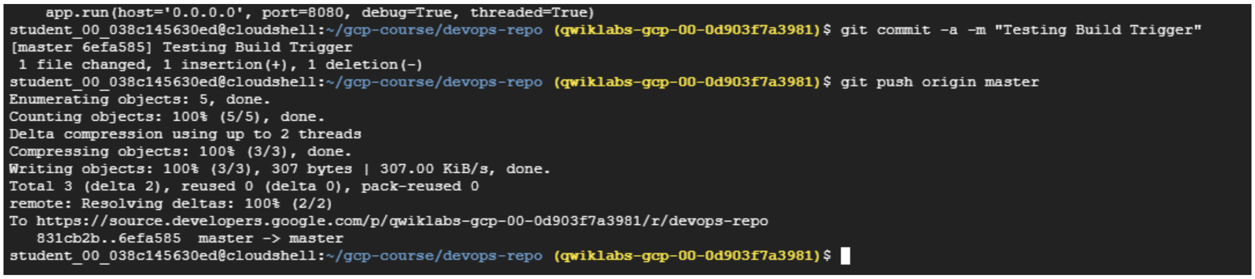
Name ↑	Hostname	Visibility
devops-image	gcr.io	Private
devops-repo	gcr.io	Private
node	gcr.io	Private
node	eu.gcr.io	Private

In the main() function, change the title property to `"Hello Build Trigger."`



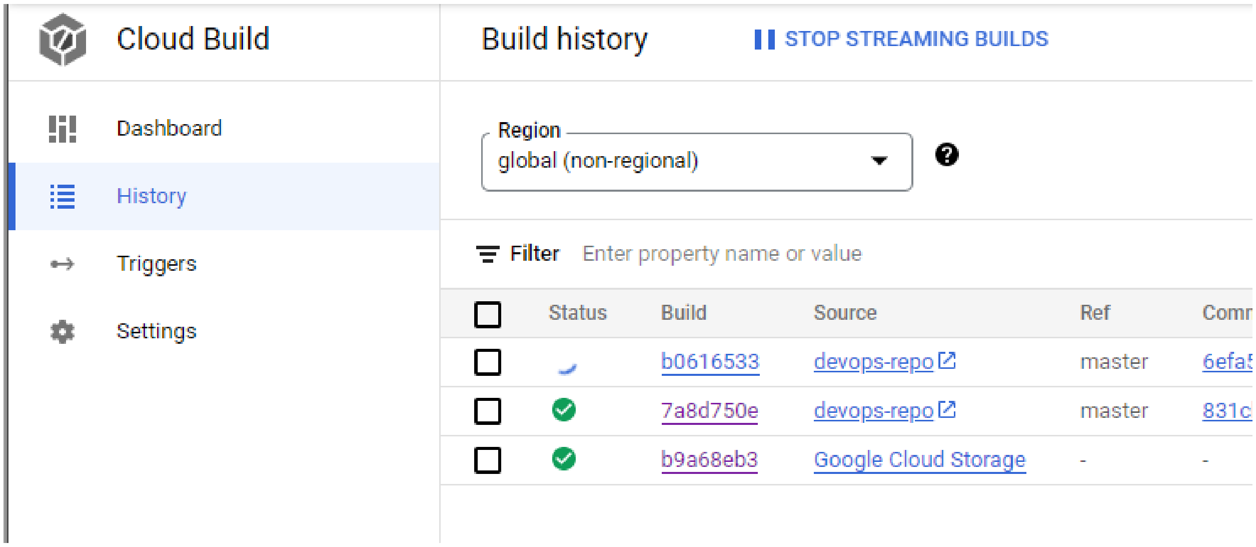
```
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ cat main.py
from flask import Flask, render_template, request
app = Flask(__name__)
@app.route("/")
def main():
    model = {"title": "Hello Build Trigger."}
    return render_template('index.html', model=model)
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8080, debug=True, threaded=True)
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$
```

push your changes to Cloud Source Repositories:



```
    app.run(host='0.0.0.0', port=8080, debug=True, threaded=True)
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git commit -a -m "Testing Build Trigger"
[master 6efa585] Testing Build Trigger
 1 file changed, 1 insertion(+), 1 deletion(-)
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$ git push origin master
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), 307 bytes | 307.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2)
To https://source.developers.google.com/p/qwiklabs-gcp-00-0d903f7a3981/r/devops-repo
   831cb2b..6efa585  master -> master
student_00_038c145630ed@cloudshell:~/gcp-course/devops-repo (qwiklabs-gcp-00-0d903f7a3981)$
```

18. Return to the Cloud Console and the **Cloud Build** service. You should see another build running.



Region: global (non-regional)

Filter: Enter property name or value

□	Status	Build	Source	Ref	Comr
□		b0616533	devops-repo	master	6efa5
□	✓	7a8d750e	devops-repo	master	831c
□	✓	b9a68eb3	Google Cloud Storage	-	-

Task 6. Test your build changes

1. When the build completes, click on it to see its details. Under **Execution Details**, copy the **Image** link, format should be `gcr.io/qwiklabs-gcp-00-f23112/devops-repoxx34345xx`.

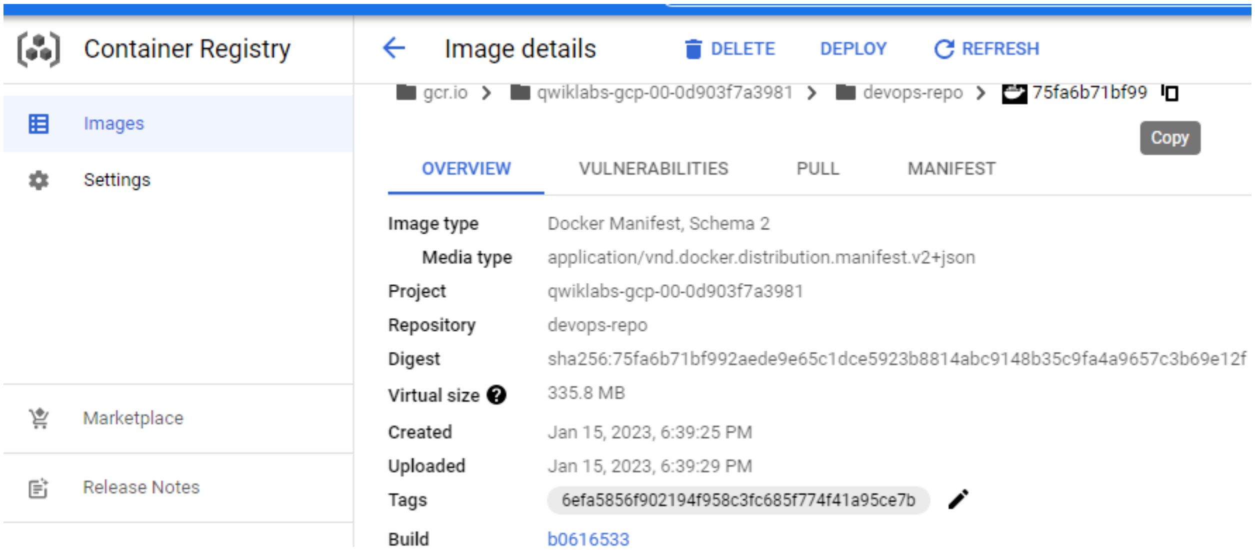
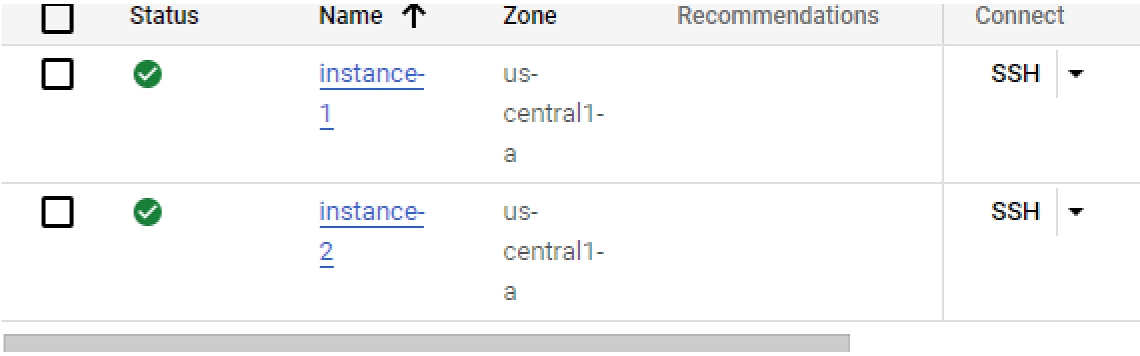
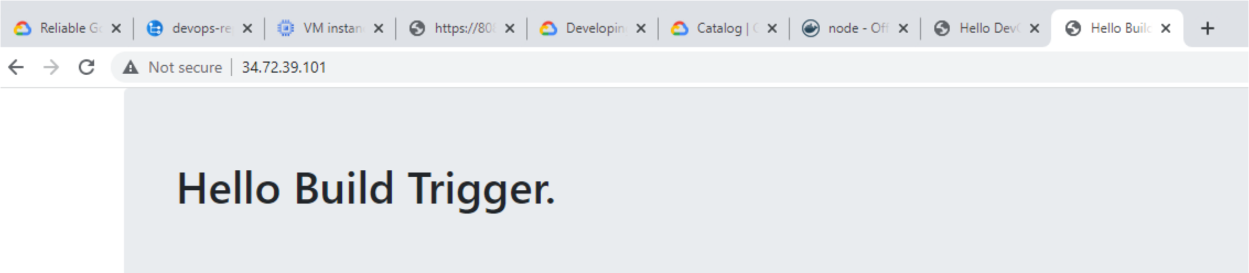


Image type	Docker Manifest, Schema 2
Media type	application/vnd.docker.distribution.manifest.v2+json
Project	qwiklabs-gcp-00-0d903f7a3981
Repository	devops-repo
Digest	sha256:75fa6b71bf992aede9e65c1dce5923b8814abc9148b35c9fa4a9657c3b69e12f
Virtual size	335.8 MB
Created	Jan 15, 2023, 6:39:25 PM
Uploaded	Jan 15, 2023, 6:39:29 PM
Tags	6efa5856f902194f958c3fc685f774f41a95ce7b
Build	b0616533

2. Go to the **Compute Engine** service. As you did earlier, create a new virtual machine to test this image. Click **DEPLOY CONTAINER** and paste the image you just copied.
3. Select **Allow HTTP traffic**.
4. When the machine is created, test your change by making a request to the VM's external IP address in your browser. Your new message should be displayed.



□	Status	Name ↑	Zone	Recommendations	Connect
□	✓	instance-1	us-central1-a		SSH ▾
□	✓	instance-2	us-central1-a		SSH ▾



Hello Build Trigger.

Machine is created and deployed with new docker image .