



# VIRTUAL INTERNSHIP

**DATA SCIENCE  
LISUM01**

**CLOUD AND API DEPLOYMENT**

**Samuel Alejandro Cueva Lozano**

**10/07/2021**

## **Sumario**

1. Introduction.....	3
2. Enable App Engine in a Cloud project.....	3
3. Environment configuration.....	5
4. Upload the trained model to a Google Cloud Storage bucket.....	6
5. Clone on Cloud Shell.....	6
6. Test the app before deploying it.....	8
7. Deploying the application.....	11

# 1. Introduction

In this document, I'm going to show how I deploy a Tensorflow model to detect COVID-19 in chest x-rays on App Engine from Google Cloud Platform.

App Engine was chosen due to its scalability and the complexity of the model.

There are two environments on App Engine, Standard and Flexible, because the model is large and it needs some resources like a certain size of memory, CPUs, and specific runtime, it will be deployed on App Engine flexible environment.

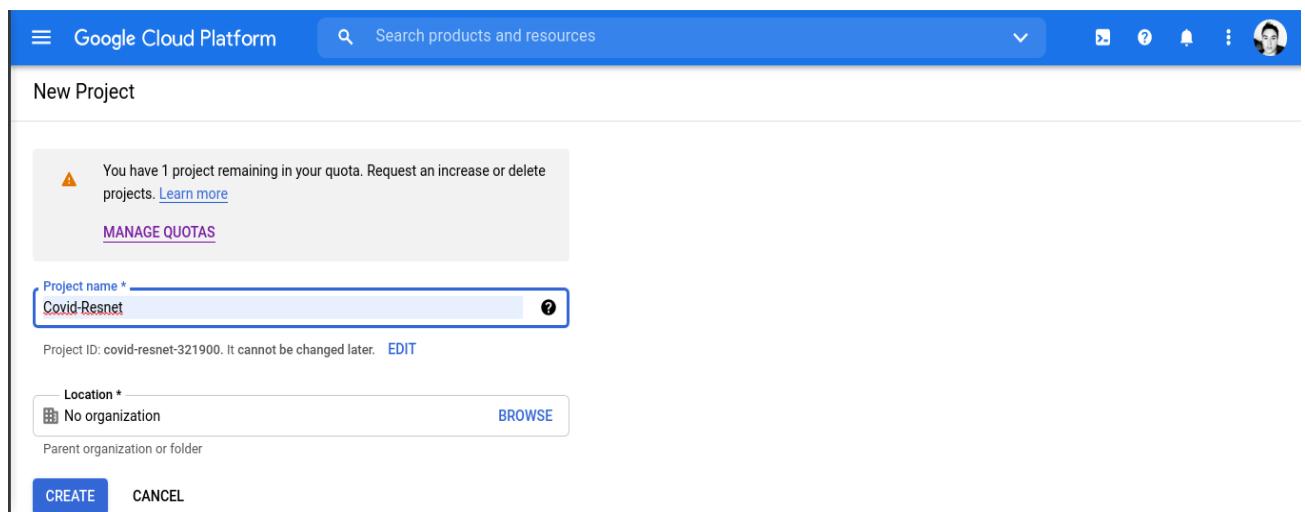
The code to train the model is on my [Github](#).

The trained weights are on my [Google Drive](#).

The full code used in this document to deploy on App Engine is on my [Github](#).

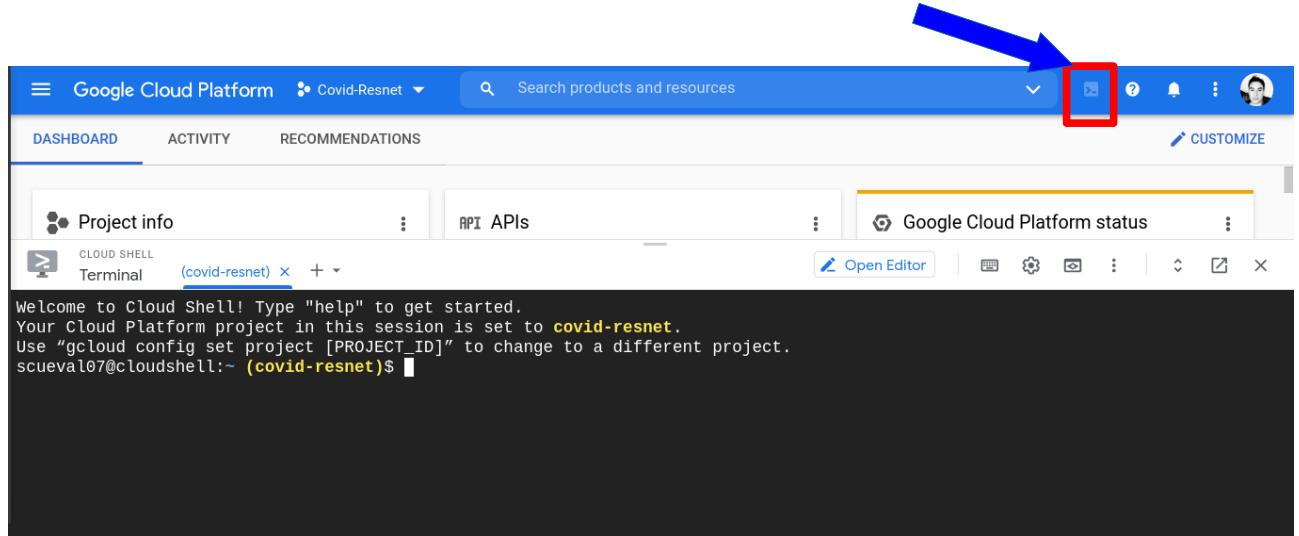
# 2. Enable App Engine in a Cloud project

First, head to [Google Cloud Console](#) and create a new project.



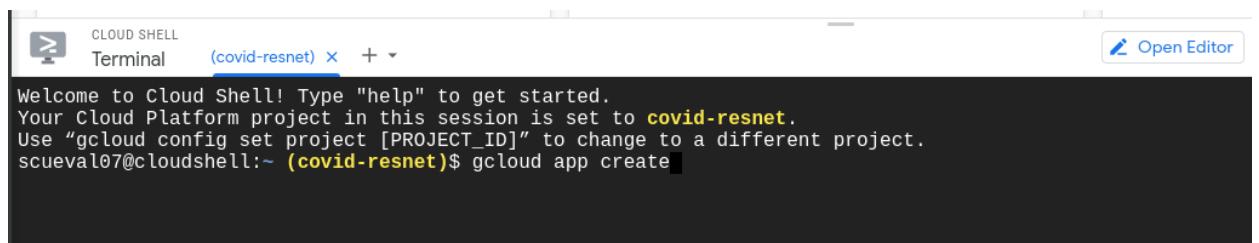
The screenshot shows the 'New Project' dialog in the Google Cloud Platform. At the top, there's a blue header bar with the 'Google Cloud Platform' logo, a search bar, and various icons. Below the header, the title 'New Project' is displayed. A warning message says: '⚠ You have 1 project remaining in your quota. Request an increase or delete projects. [Learn more](#)' with a 'MANAGE QUOTAS' link. The 'Project name \*' field is filled with 'Covid-Resnet'. Below it, the 'Project ID' is listed as 'covid-resnet-321900. It cannot be changed later.' with an 'EDIT' link. The 'Location \*' section shows 'No organization' with a 'BROWSE' button. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

Once the project was created, open *Cloud Shell*.

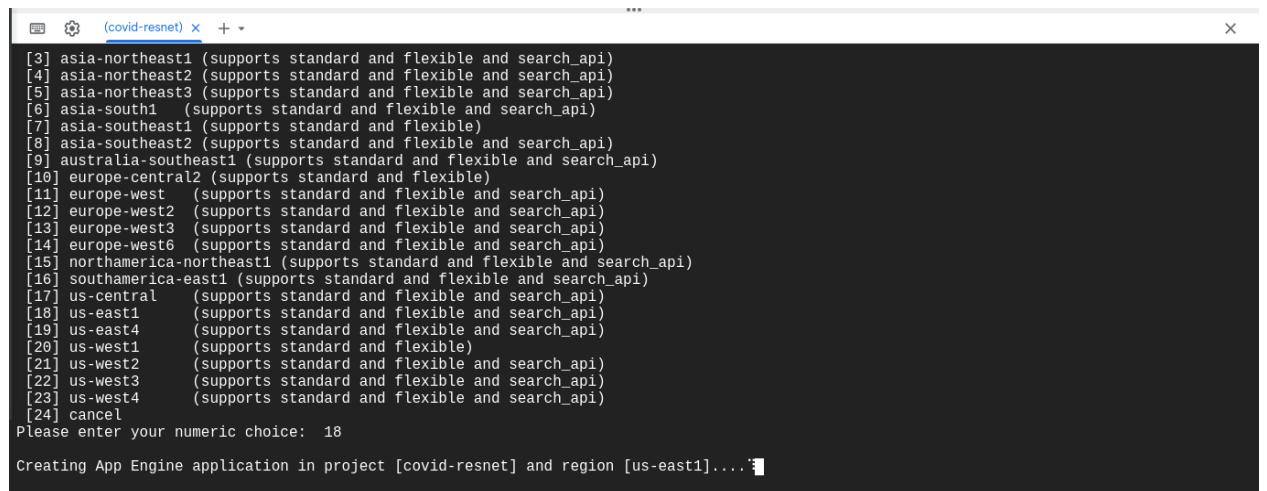


**Cloud shell** is a machine hosted in Google's infrastructure which comes with command-line tools.

### Enable App Engine



Choose a region that's closest to you.

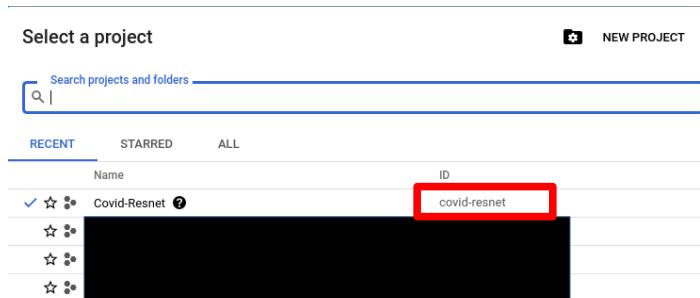


The process is complete when you get this message: Success! The app is now created.

### 3. Environment configuration

Set some environment variables in your local machine.

Find out the project ID in Google Cloud Console.



Set the environment variable PROJECT\_ID in your local machine and use Google SDK to set it as the main project.

```
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ export PROJECT_ID=covid-resnet
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ gcloud config set project $PROJECT_ID
Updated property [core/project].
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ gcloud config list project
[core]
project = covid-resnet

Your active configuration is: [default]
```

The gsutil tool is used to see the buckets in the project.

```
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ gsutil list
gs://covid-resnet.appspot.com/
gs://staging.covid-resnet.appspot.com/
gs://us.artifacts.covid-resnet.appspot.com/
```

Set the environment variable BUCKET\_NAME with the bucket that has a domain like 'gs://ProjectName.appspot.com'.

```
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ gsutil list
gs://covid-resnet.appspot.com/
gs://staging.covid-resnet.appspot.com/
gs://us.artifacts.covid-resnet.appspot.com/
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ export BUCKET_NAME=covid-resnet.appspot.com
samuelcueva@debiantsc:~/machine_learning/deploy_gcp/Deploy_COVID-RESNET$ 
```

## 4. Upload the trained model to a Google Cloud Storage bucket

This step is done because the application needs to load the model that weighs more than 100 MB, this is better than uploading the model to *Github*.

The *gsutil* tool is used again to upload the model to Google Cloud Storage from a local machine.

```
 samuelcueva@debiantsc:~$ gsutil cp covid_resnet_model.zip gs://covid-resnet.appspot.com
Updates are available for some Cloud SDK components. To install them,
please run:
  $ gcloud components update
Copying file://covid_resnet_model.zip [Content-Type=application/zip]...
\ [1 files][113.1 MiB/113.1 MiB]      0.0 B/s
Operation completed over 1 objects/113.1 MiB.
```

## 5. Clone on Cloud Shell

Well, now App Engine is ready to deploy our model which was pushed to Github previously.

```
 app.yaml
 cloud_build.yaml
 download_model.py
 images
   └── image
 main.py
 model.py
 requirements.txt
 static
   ├── script.js
   └── style.css
 templates
   └── index.html
 3 directories, 10 files
 local
```

samuelcueva decimal correction		
		119ceab 8 hours ago 17 commits
📁 images	requirements were added	20 days ago
📁 static	requirements were added	20 days ago
📁 templates	correction of some errors	19 days ago
📄 .gitignore	correction of some errors	19 days ago
📄 app.yaml	ready to App Engine	yesterday
📄 cloud_build.yaml	rename cloud build config file	9 hours ago
📄 download_model.py	add download_model.py	10 days ago
📄 main.py	decimal correction	8 hours ago
📄 model.py	better performance	25 days ago
📄 requirements.txt	change to flex env	3 days ago

*Github*

The app consists of:

**main.py** The main Python file that will call the functions *download\_model.py* and *model.py* to create the model and load the serialized weights, also it will respond to our requests when uploading our x-ray image.

**app.yaml** An App Engine configuration file with the required settings like runtime, environment type, resources, etc. These settings are required for deployment.

**Requirements.txt** A text file listing package dependencies that the app requires such as *Tensorflow*.

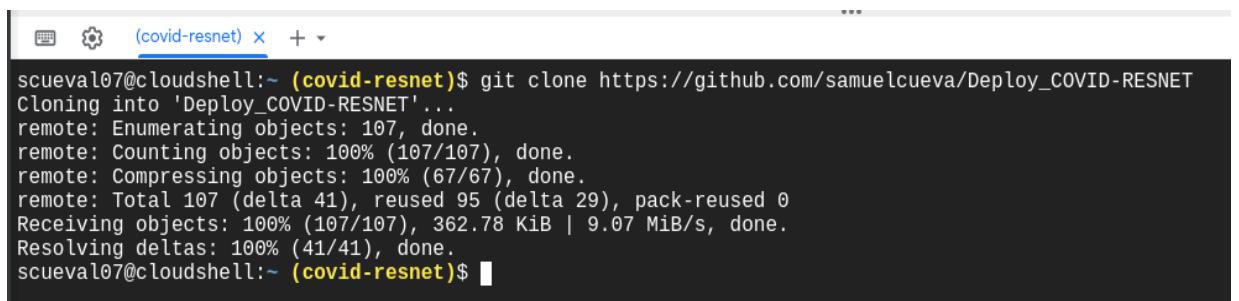
**cloud\_build.yaml** Configuration file for application CI/CD on *Google Cloud Build*.

**model.py** Python file with the functions for image preprocessing and to create the *Tensorflow* model.

**download\_model.py** Python file to download the model from a *Google Cloud Storage* bucket.

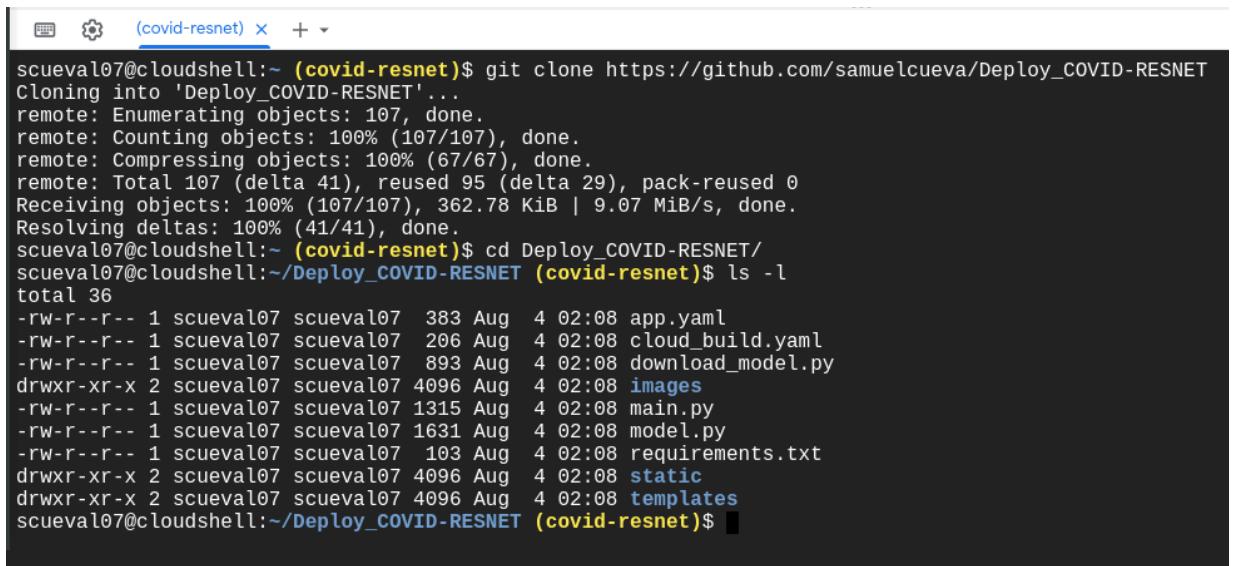
The folders **templates**, **static**, and **images** are used for the application's front-end.

Cloning the repository from [Github](#) on *Cloud Shell*.



```
scueval07@cloudshell:~ (covid-resnet)$ git clone https://github.com/samuelcueva/Deploy_COVID-RESNET
Cloning into 'Deploy_COVID-RESNET'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 107 (delta 41), reused 95 (delta 29), pack-reused 0
Receiving objects: 100% (107/107), 362.78 KiB | 9.07 MiB/s, done.
Resolving deltas: 100% (41/41), done.
scueval07@cloudshell:~ (covid-resnet)$
```

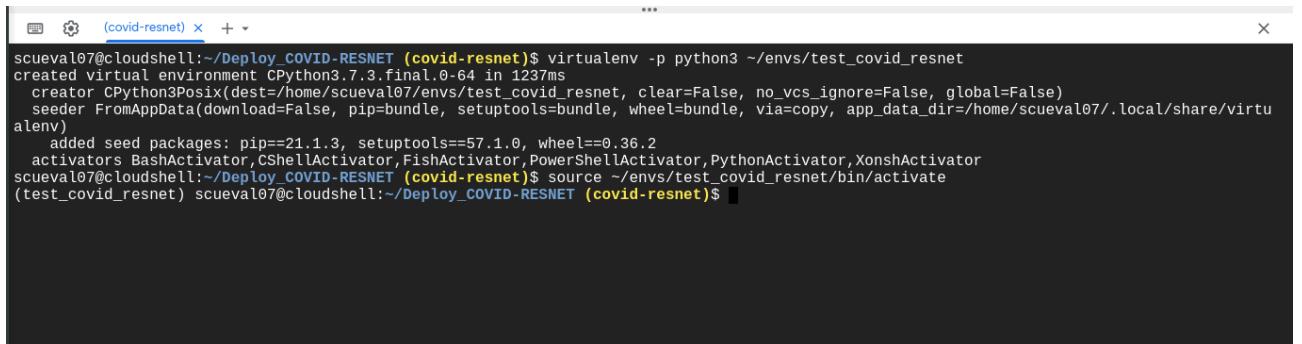
Move to the app's directory and take a look at the files.



```
scueval07@cloudshell:~ (covid-resnet)$ git clone https://github.com/samuelcueva/Deploy_COVID-RESNET
Cloning into 'Deploy_COVID-RESNET'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 107 (delta 41), reused 95 (delta 29), pack-reused 0
Receiving objects: 100% (107/107), 362.78 KiB | 9.07 MiB/s, done.
Resolving deltas: 100% (41/41), done.
scueval07@cloudshell:~ (covid-resnet)$ cd Deploy_COVID-RESNET/
scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ ls -l
total 36
-rw-r--r-- 1 scueval07 scueval07 383 Aug  4 02:08 app.yaml
-rw-r--r-- 1 scueval07 scueval07 206 Aug  4 02:08 cloud_build.yaml
-rw-r--r-- 1 scueval07 scueval07 893 Aug  4 02:08 download_model.py
drwxr-xr-x 2 scueval07 scueval07 4096 Aug  4 02:08 images
-rw-r--r-- 1 scueval07 scueval07 1315 Aug  4 02:08 main.py
-rw-r--r-- 1 scueval07 scueval07 1631 Aug  4 02:08 model.py
-rw-r--r-- 1 scueval07 scueval07 103 Aug  4 02:08 requirements.txt
drwxr-xr-x 2 scueval07 scueval07 4096 Aug  4 02:08 static
drwxr-xr-x 2 scueval07 scueval07 4096 Aug  4 02:08 templates
scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$
```

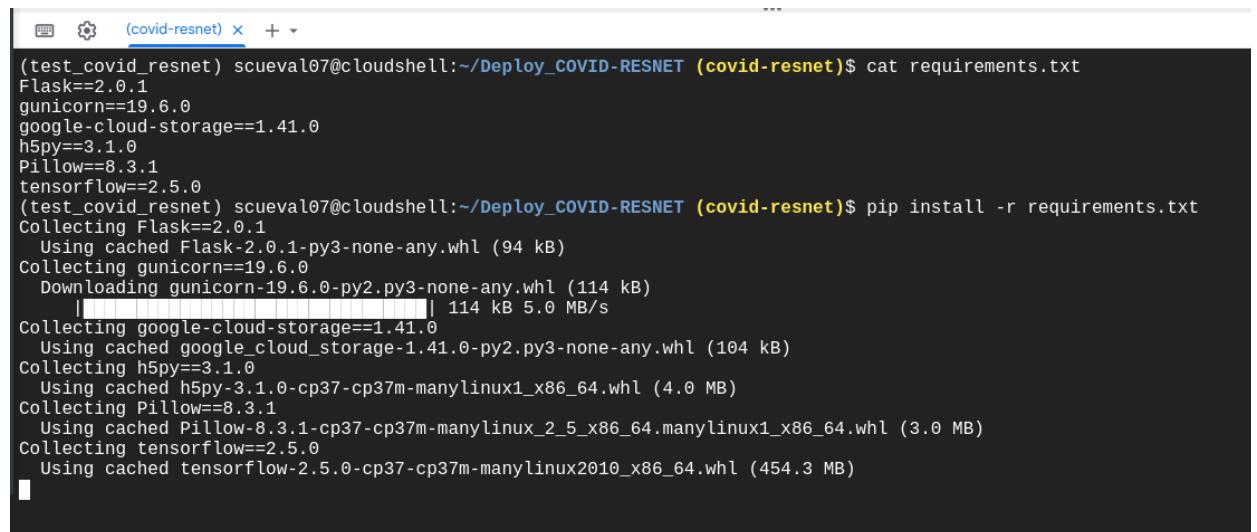
## 6. Test the app before deploying it

Create and activate a Python virtual environment in which to test the app. *Virtualenv* is used to create the virtual environment.

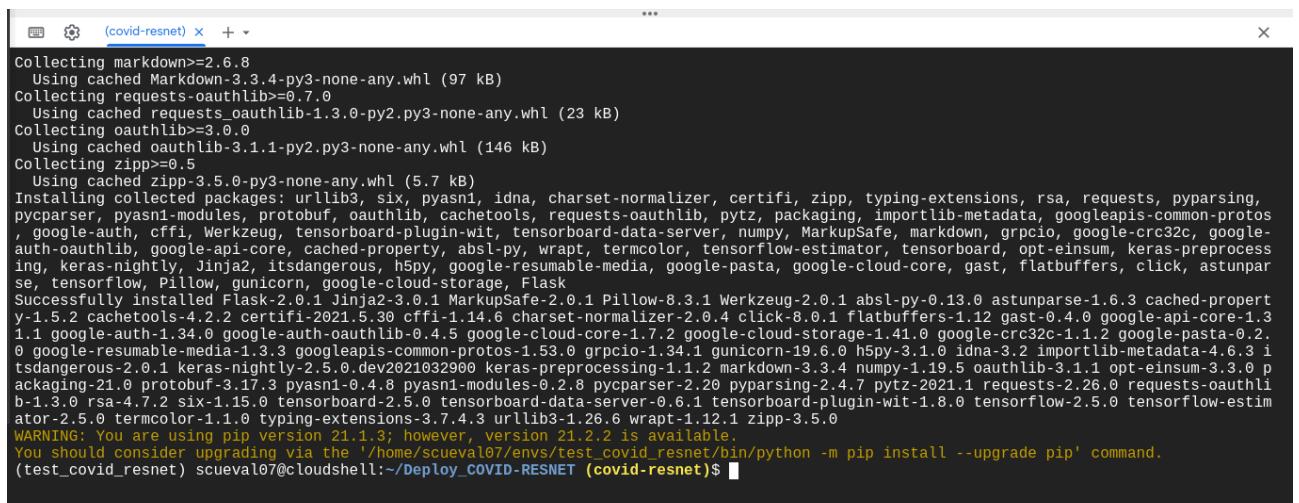


```
scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ virtualenv -p python3 ~/envs/test_covid_resnet
created virtual environment CPython3.7.3.final.0-64 in 1237ms
  creator CPython3Posix(dest=/home/scueval07/envs/test_covid_resnet, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/scueval07/.local/share/virtualenvs)
    added seed packages: pip==21.1.3, setuptools==57.1.0, wheel==0.36.2
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ source ~/envs/test_covid_resnet/bin/activate
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$
```

Take a look at **requirements.txt** file and install all the dependencies that are in.



```
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ cat requirements.txt
Flask==2.0.1
gunicorn==19.6.0
google-cloud-storage==1.41.0
h5py==3.1.0
Pillow==8.3.1
tensorflow==2.5.0
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ pip install -r requirements.txt
Collecting Flask==2.0.1
  Using cached Flask-2.0.1-py3-none-any.whl (94 kB)
Collecting gunicorn==19.6.0
  Downloading gunicorn-19.6.0-py2.py3-none-any.whl (114 kB)
    |██████████| 114 kB 5.0 MB/s
Collecting google-cloud-storage==1.41.0
  Using cached google_cloud_storage-1.41.0-py2.py3-none-any.whl (104 kB)
Collecting h5py==3.1.0
  Using cached h5py-3.1.0-cp37-cp37m-manylinux1_x86_64.whl (4.0 MB)
Collecting Pillow==8.3.1
  Using cached Pillow-8.3.1-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (3.0 MB)
Collecting tensorflow==2.5.0
  Using cached tensorflow-2.5.0-cp37-cp37m-manylinux2010_x86_64.whl (454.3 MB)
```



```
Collecting markdown>=2.6.8
  Using cached Markdown-3.3.4-py3-none-any.whl (97 kB)
Collecting requests-oauthlib>=0.7.0
  Using cached requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Collecting oauthlib>=3.0.0
  Using cached oauthlib-3.1.1-py2.py3-none-any.whl (146 kB)
Collecting zipp>=0.5
  Using cached zipp-3.5.0-py3-none-any.whl (5.7 kB)
Installing collected packages: urllib3, six, pyasn1, idna, charset-normalizer, certifi, zipp, typing-extensions, rsa, requests, pyparsing, pycparser, pyasn1-modules, protobuf, oauthlib, cachetools, requests-oauthlib, pytz, packaging, importlib-metadata, googleapis-common-protos, google-auth, cffi, Werkzeug, tensorboard-plugin-wit, tensorboard-data-server, numpy, MarkupSafe, markdown, grpcio, google-crc32c, google-auth-oauthlib, google-api-core, cached-property, absl-py, wrapt, termcolor, tensorflow-estimator, tensorboard, opt-einsum, keras-preprocessing, keras-nightly, Jinja2, itsdangerous, h5py, google-resumable-media, google-pasta, google-cloud-core, gast, flatbuffers, click, astunparse, tensorflow, Pillow, gunicorn, google-cloud-storage, Flask
Successfully installed Flask-2.0.1 Jinja2-3.0.1 MarkupSafe-2.0.1 Pillow-8.3.1 Werkzeug-2.0.1 absl-py-0.13.0 astunparse-1.6.3 cached-property-1.5.2 cachetools-4.2.2 certifi-2021.5.30 cffi-1.14.6 charset-normalizer-2.0.4 click-8.0.1 flatbuffers-1.12 gast-0.4.0 google-api-core-1.3.1.1 google-auth-1.34.0 google-auth-oauthlib-0.4.5 google-cloud-core-1.7.2 google-cloud-storage-1.41.0 google-crc32c-1.1.2 google-pasta-0.2.0 google-resumable-media-1.3.3 googleapis-common-protos-1.53.0 grpcio-1.34.1 gunicorn-19.6.0 h5py-3.1.0 idna-3.2 importlib-metadata-4.6.3 itsdangerous-2.0.1 keras-nightly-2.5.0.dev2021032900 keras-preprocessing-1.1.2 markdown-3.3.4 numpy-1.19.5 oauthlib-3.1.1 opt-einsum-3.3.0 pckaging-21.0 protobuf-3.17.3 pyasn1-0.4.8 pyasn1-modules-0.2.8 pycparser-2.20 pyparsing-2.4.7 pytz-2021.1 requests-2.26.0 requests-oauthlib-1.3.0 rsa-4.7.2 six-1.15.0 tensorboard-2.5.0 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.0 tensorflow-2.5.0 tensorflow-estimator-2.5.0 termcolor-1.1.0 typing-extensions-3.7.4.3 urllib3-1.26.6 wrapt-1.12.1 zipp-3.5.0
WARNING: You are using pip version 21.1.3; however, version 21.2.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$
```

Once all dependencies were installed, we can run the **main.py** file.

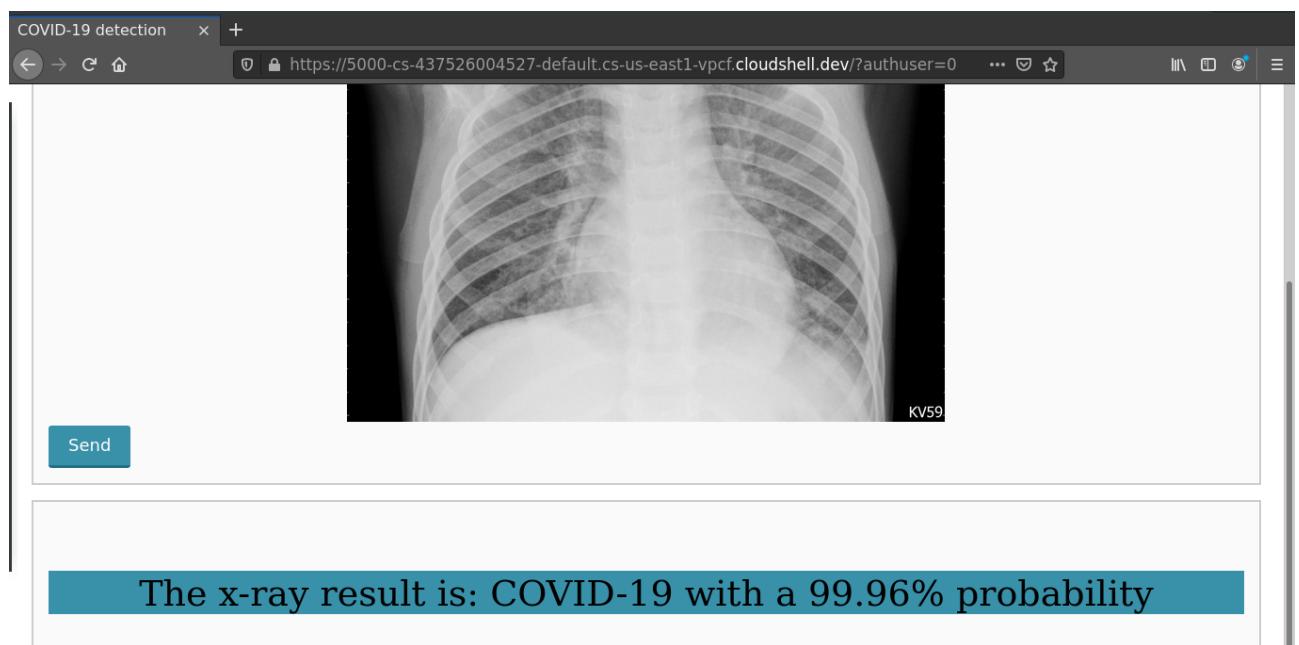
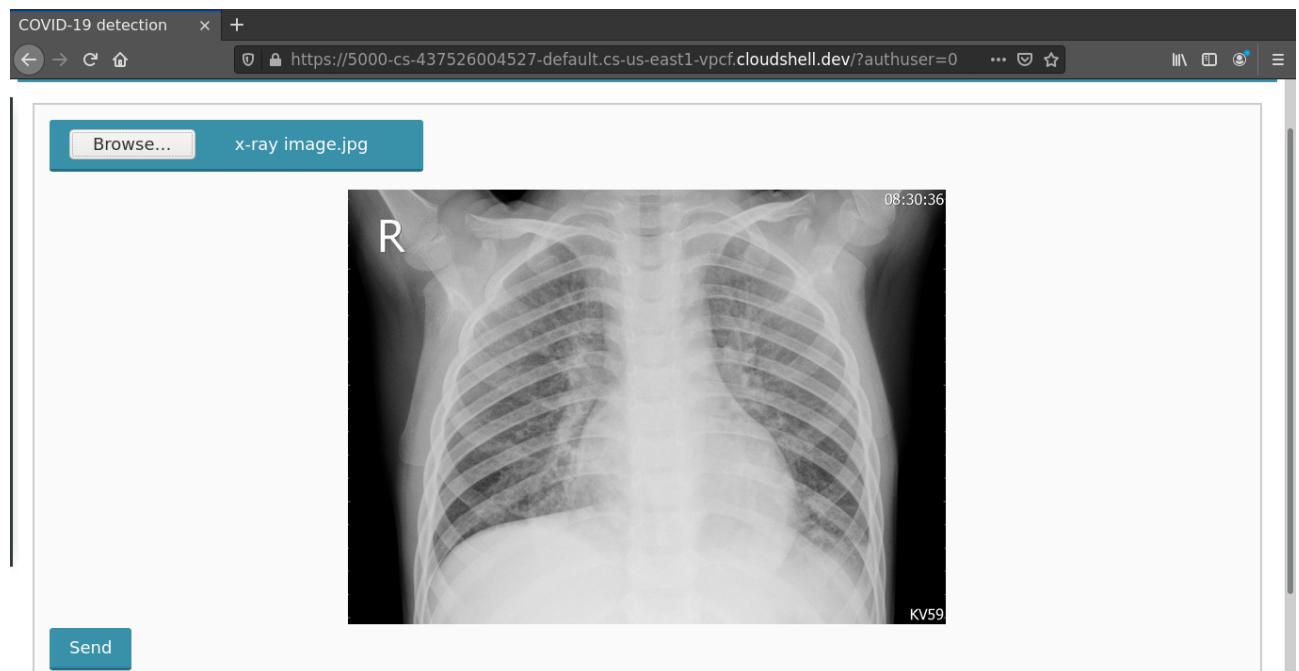
```
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ python main.py
2021-08-04 02:45:39.373951: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open shared object file: No such file or directory
2021-08-04 02:45:39.374002: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
Blob covid_resnet_model.zip downloaded to model.zip.
File Name                                Modified      Size
covid_resnet_model/                         2021-07-23 21:48:12    0
covid_resnet_model/assets/                  2021-07-23 21:48:12    0
covid_resnet_model/variables/               2021-07-23 21:46:52    0
covid_resnet_model/variables/variables.data-00000-of-00001 2021-07-23 21:46:18  128083931
covid_resnet_model/variables/variables.index 2021-07-23 21:46:52   18462
covid_resnet_model/saved_model.pb          2021-07-08 23:46:22   4887357
Extracting all the files now...
```

```
File Name                                Modified      Size
covid_resnet_model/variables/               2021-07-23 21:46:52    0
covid_resnet_model/variables/variables.data-00000-of-00001 2021-07-23 21:46:18  128083931
covid_resnet_model/variables/variables.index 2021-07-23 21:46:52   18462
covid_resnet_model/saved_model.pb          2021-07-08 23:46:22   4887357
Extracting all the files now...
Done!
WARNING:tensorflow:SavedModel saved prior to TF 2.5 detected when loading Keras model. Please ensure that you are saving the model with model.save() or tf.keras.models.save_model(), *NOT* tf.saved_model.save(). To confirm, there should be a file named "keras_metadata.pb" in the SavedModel directory.
2021-08-04 02:45:44.835582: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-08-04 02:45:44.835661: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)
2021-08-04 02:45:44.835714: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (cs-437526004527-default-default-ztcm4): /proc/driver/nvidia/version does not exist
2021-08-04 02:45:44.836799: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:tensorflow:No training configuration found in save file, so the model was 'not' compiled. Compile it manually.
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

After that, the app is running locally, and to view it click on Web preview.

```
Cloud Shell Editor
File Name                                Modified      Size
covid_resnet_model/                         2021-07-23 21:48:12    0
covid_resnet_model/assets/                  2021-07-23 21:48:12    0
covid_resnet_model/variables/               2021-07-23 21:46:52    0
covid_resnet_model/variables/variables.data-00000-of-00001 2021-07-23 21:46:18  128083931
covid_resnet_model/variables/variables.index 2021-07-23 21:46:52   18462
covid_resnet_model/saved_model.pb          2021-07-08 23:46:22   4887357
Extracting all the files now...
Done!
WARNING:tensorflow:SavedModel saved prior to TF 2.5 detected when loading Keras model. Please ensure that you are saving the model with model.save() or tf.keras.models.save_model(), *NOT* tf.saved_model.save(). To confirm, there should be a file named "keras_metadata.pb" in the SavedModel directory.
2021-08-04 02:45:44.835582: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-08-04 02:45:44.835661: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)
2021-08-04 02:45:44.835714: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (cs-437526004527-default-default-ztcm4): /proc/driver/nvidia/version does not exist
2021-08-04 02:45:44.836799: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:tensorflow:No training configuration found in save file, so the model was 'not' compiled. Compile it manually.
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

## Previsualization

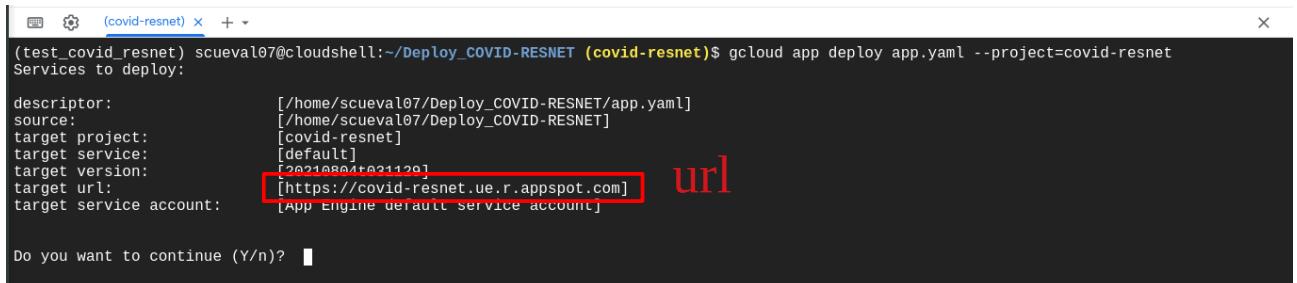


The application is working correctly and is ready for deployment

To stop the application only type Ctrl + c

## 7. Deploying the application

The file **app.yaml** is passed to *gcloud app deploy* command.

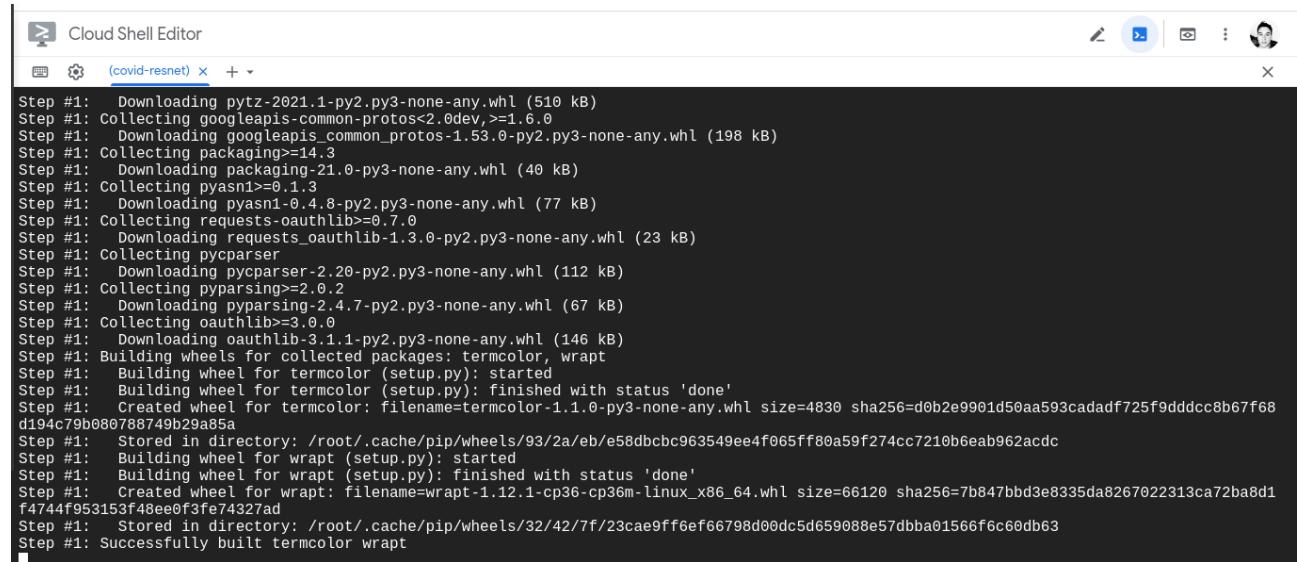


```
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$ gcloud app deploy app.yaml --project=covid-resnet
Services to deploy:
descriptor: [/home/scueval07/Deploy_COVID-RESNET/app.yaml]
source: [/home/scueval07/Deploy_COVID-RESNET]
target project: [covid-resnet]
target service: [default]
target version: [20210804t021129]
target url: [https://covid-resnet.ue.r.appspot.com] url
target service account: [App Engine default service account]

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

If you agree, type Y.

The app is built, this process may take several minutes



```
Cloud Shell Editor
(covid-resnet) x + v
Step #1: Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
Step #1: Collecting googleapis-common-protos<2.0dev,>=1.6.0
Step #1: Downloading googleapis_common_protos-1.53.0-py2.py3-none-any.whl (198 kB)
Step #1: Collecting packaging<14.3
Step #1: Downloading packaging-21.0-py3-none-any.whl (40 kB)
Step #1: Collecting pyasn1<0.1.3
Step #1: Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
Step #1: Collecting requests-oauthlib<0.7.0
Step #1: Downloading requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Step #1: Collecting pycparser
Step #1: Downloading pycparser-2.20-py2.py3-none-any.whl (112 kB)
Step #1: Collecting pyparsing<2.0.2
Step #1: Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
Step #1: Collecting oauthlib<3.0.0
Step #1: Downloading oauthlib-3.1.1-py2.py3-none-any.whl (146 kB)
Step #1: Building wheels for collected packages: termcolor, wrapt
Step #1: Building wheel for termcolor (setup.py): started
Step #1: Building wheel for termcolor (setup.py): finished with status 'done'
Step #1: Created wheel for termcolor: filename=termcolor-1.1.0-py3-none-any.whl size=4830 sha256=d0b2e9901d50aa593cadadf725f9ddcc8b67f68d194c79b080788749b29a85a
Step #1: Stored in directory: /root/.cache/pip/wheels/93/2a/eb/e58dbc963549ee4f065ff80a59f274cc7210b6eab962acdc
Step #1: Building wheel for wrapt (setup.py): started
Step #1: Building wheel for wrapt (setup.py): finished with status 'done'
Step #1: Created wheel for wrapt: filename=wrapt-1.12.1-cp36-cp36m-linux_x86_64.whl size=66120 sha256=7b847bbd3e8335da8267022313ca72ba8d1f4744f953153f48ee0f3fe74327ad
Step #1: Stored in directory: /root/.cache/pip/wheels/32/42/7f/23cae9ff6ef66798d00dc5d659088e57dbba01566f6c60db63
Step #1: Successfully built termcolor wrapt
```

Done, the application is already deployed.



```
9f29a7bebcd: Pushed
1905213e2720: Pushed
dbc2e39a730: Pushed
latest: digest: sha256:8741840ea1eb40824962f0de28db5ff3742fc1d1246172a82f098a5ae6121862 size: 3677
DONE
-----
Updating service [default] (this may take several minutes)...:
Updating.service-[default].[this.may.take.several.minutes]...done.size 100x28 from a smaller client
Setting traffic split for service [default]...done.
Deployed service [default] to [https://covid-resnet.ue.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
(test_covid_resnet) scueval07@cloudshell:~/Deploy_COVID-RESNET (covid-resnet)$
```

Now, anyone can use the application by using the url: <https://covid-resnet.ue.r.appspot.com>

The screenshot shows two consecutive screenshots of a web application titled "COVID-19 DETECTION BY CHEST X-RAY".

**Screenshot 1:** The top part shows a file upload interface. A "Browse..." button is highlighted, and the message "No file selected." is displayed. Below it is a "Send" button.

**Screenshot 2:** The bottom part shows the result of a chest X-ray analysis. The X-ray image is centered, showing the lungs and heart area. The text "KV59" is visible at the bottom right of the image. Below the image is another "Send" button.

**Result Panel:** A teal-colored bar at the bottom of the second screenshot contains the text "The x-ray result is: COVID-19 with a 99.96% probability".

#### Notes:

- Due to the size of the model and the resources it consumes, keeping it online is expensive, therefore if you want to test the application you can write to me at [scueval07@gmail.com](mailto:scueval07@gmail.com) to activate it for a certain period.
- The model provided in this article was created by personal research and is not ready to be used in production, it should not be used for the clinical diagnosis of COVID-19, the diagnosis must be given by a competent specialist.