

Google cloud platform project

Deploy your flask app on Google Cloud (App Engine)

by Abdulaziz alqahtani

Guidelines:

- First make your own GCP account.
- second after you make your account go to app engine and create your application
- third deploying your flask app

Hands-on-deck \$:~

App Engine settings:

here you make your first application.

Google Cloud cj

Flask-app ▾

- Cloud overview
- Products & solutions
- APIs & Services
- Billing
- IAM & Admin
- Marketplace
- Compute Engine
- Kubernetes Engine
- Cloud Storage
- BigQuery
- VPC network
- Cloud Run
- SQL
- Security
- Google Maps Platform

You're working on project [Flask-app](#) ⓘ
Number: 548872705427 ID: flask-app-419713

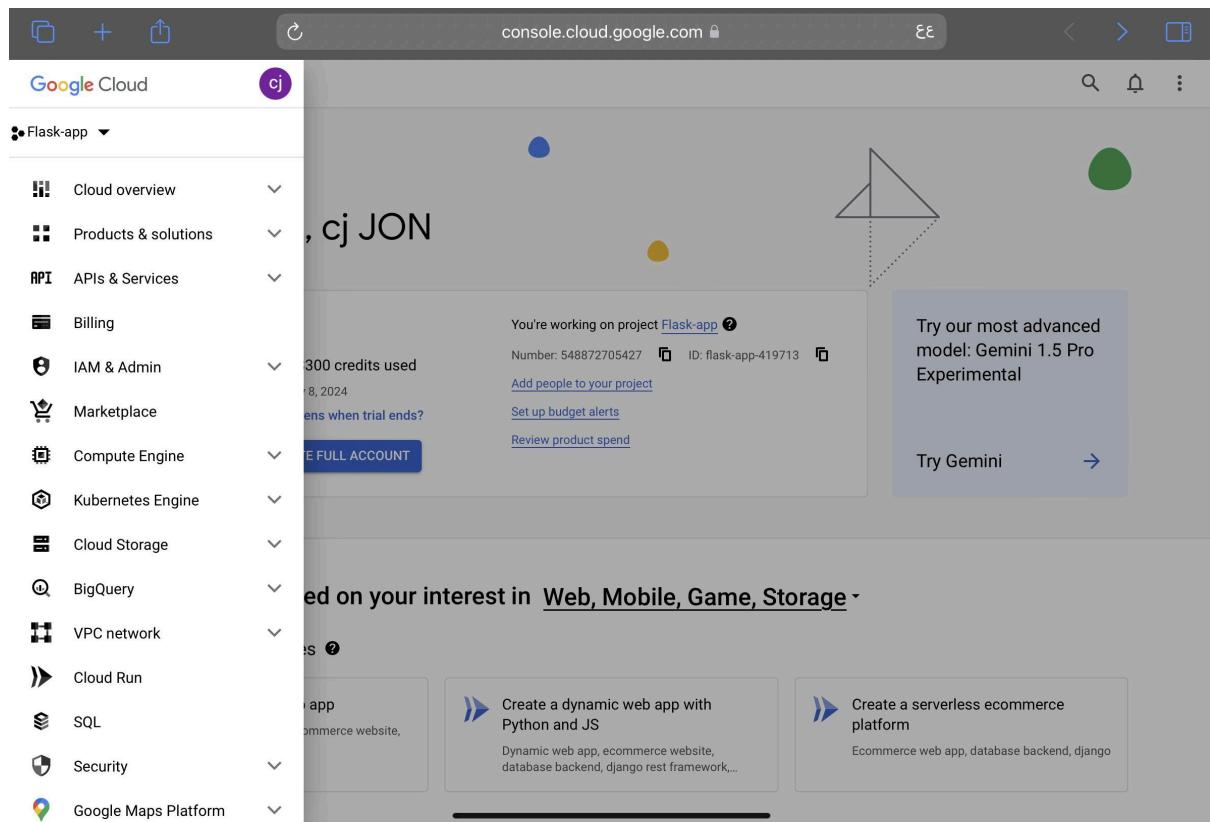
Add people to your project
Set up budget alerts
Review product spend

Try our most advanced model: Gemini 1.5 Pro Experimental
Try Gemini →

Based on your interest in [Web, Mobile, Game, Storage](#) ▾

Create a dynamic web app with Python and JS
Dynamic web app, ecommerce website, database backend, django rest framework,...

Create a serverless ecommerce platform
Ecommerce web app, database backend, django



Google Cloud cj

Flask-app ▾

DISTRIBUTED CLOUD

- Appliances
- Edge

SERVERLESS

- Cloud Run
- Cloud Functions
- App Engine
- API Gateway
- Endpoints

STORAGE

- Filestore
- Cloud Storage
- NetApp Volumes
- Storage Transfer

You're working on project [Flask-app](#) ⓘ
Number: 548872705427 ID: flask-app-419713

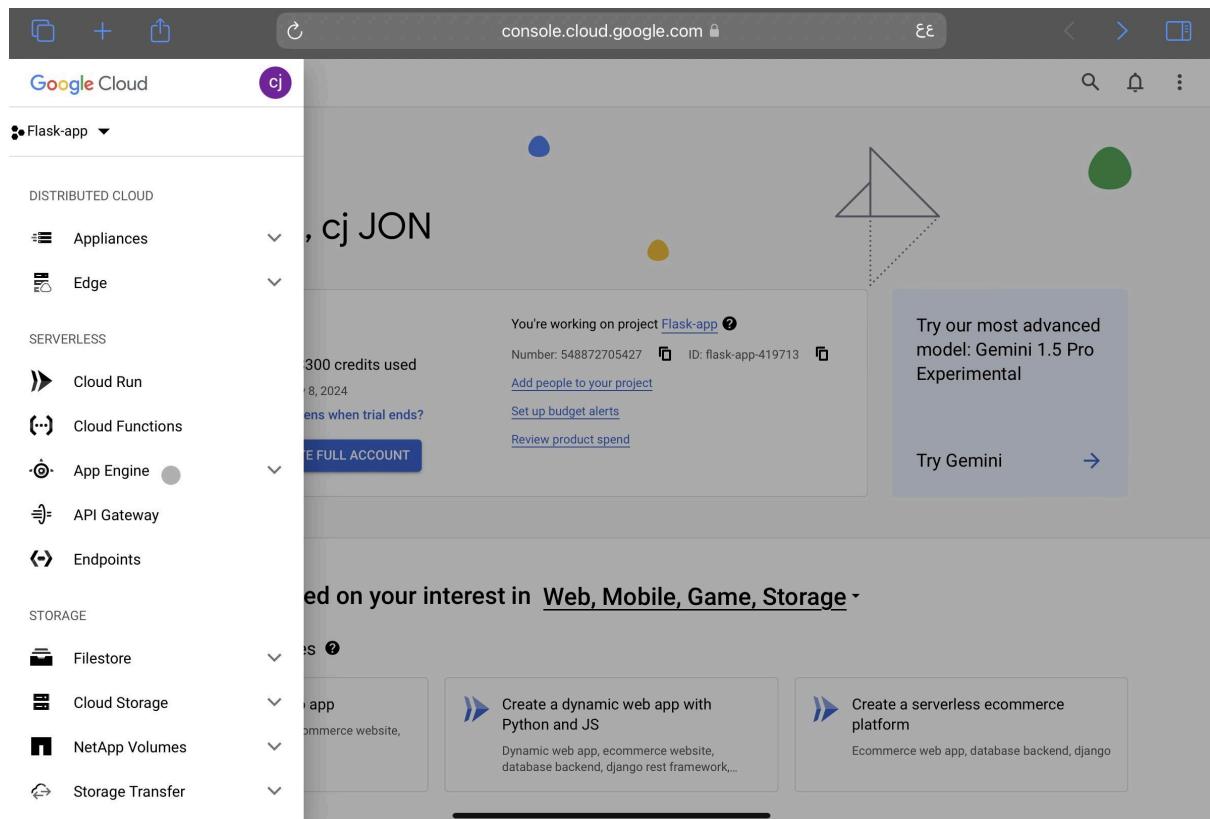
Add people to your project
Set up budget alerts
Review product spend

Try our most advanced model: Gemini 1.5 Pro Experimental
Try Gemini →

Based on your interest in [Web, Mobile, Game, Storage](#) ▾

Create a dynamic web app with Python and JS
Dynamic web app, ecommerce website, database backend, django rest framework,...

Create a serverless ecommerce platform
Ecommerce web app, database backend, django



Google Cloud cj

Flask-app

Edge

SERVERLESS

Cloud Run

Cloud Functions

App Engine

Dashboard

Services

Versions

Instances

Task queues

Cron jobs

Security scans

Firewall rules

Quotas

Memcache

Search

You're working on project [Flask-app](#)

300 credits used

Number: 548872705427 ID: flask-app-419713

Add people to your project

Set up budget alerts

Review product spend

Try our most advanced model: Gemini 1.5 Pro Experimental

Try Gemini →

Based on your interest in [Web](#), [Mobile](#), [Game](#), [Storage](#) -

Create a dynamic web app with Python and JS

Dynamic web app, ecommerce website, database backend, django rest framework,...

Create a serverless ecommerce platform

Ecommerce web app, database backend, django

And click on create.

Google Cloud

App Engine

Dashboard

Services

Versions

Instances

Task queues

Cron jobs

Security scans

Firewall rules

Quotas

Memcache

Search

Settings

Release Notes

Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

CREATE APPLICATION

The application settings.

Screenshot of the Google Cloud Console showing the App Engine "Create app" configuration page.

The URL in the browser is `console.cloud.google.com`.

The left sidebar shows the "App Engine" dashboard with various options: Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. The "Dashboard" option is selected.

The main content area is titled "Configure application" (Step 1) and "Get started" (Step 2). It features a "Region" section with a world map showing regions like North America, South America, Europe, Africa, Asia, and Oceania. A callout box highlights the "us-central" region. Below the map is a dropdown menu labeled "Select a region *".

The "Identity and API access" section includes a dropdown menu labeled "Select a service account".

Choose the API access.

Screenshot of the Google Cloud App Engine 'Create app' page. The left sidebar shows navigation options like Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. The main area is titled 'Create app' and 'Region'. It displays a world map with various regions marked by location pins. A modal window titled 'App Engine default service account' is open, showing a dropdown menu with 'App Engine default service account' selected. Below the dropdown are buttons for 'NEW SERVICE ACCOUNT' and 'REFRESH LIST'. A note at the bottom states: 'If no service account is selected the default App Engine service account will be used.' The browser address bar shows 'console.cloud.google.com'.

Now click create.

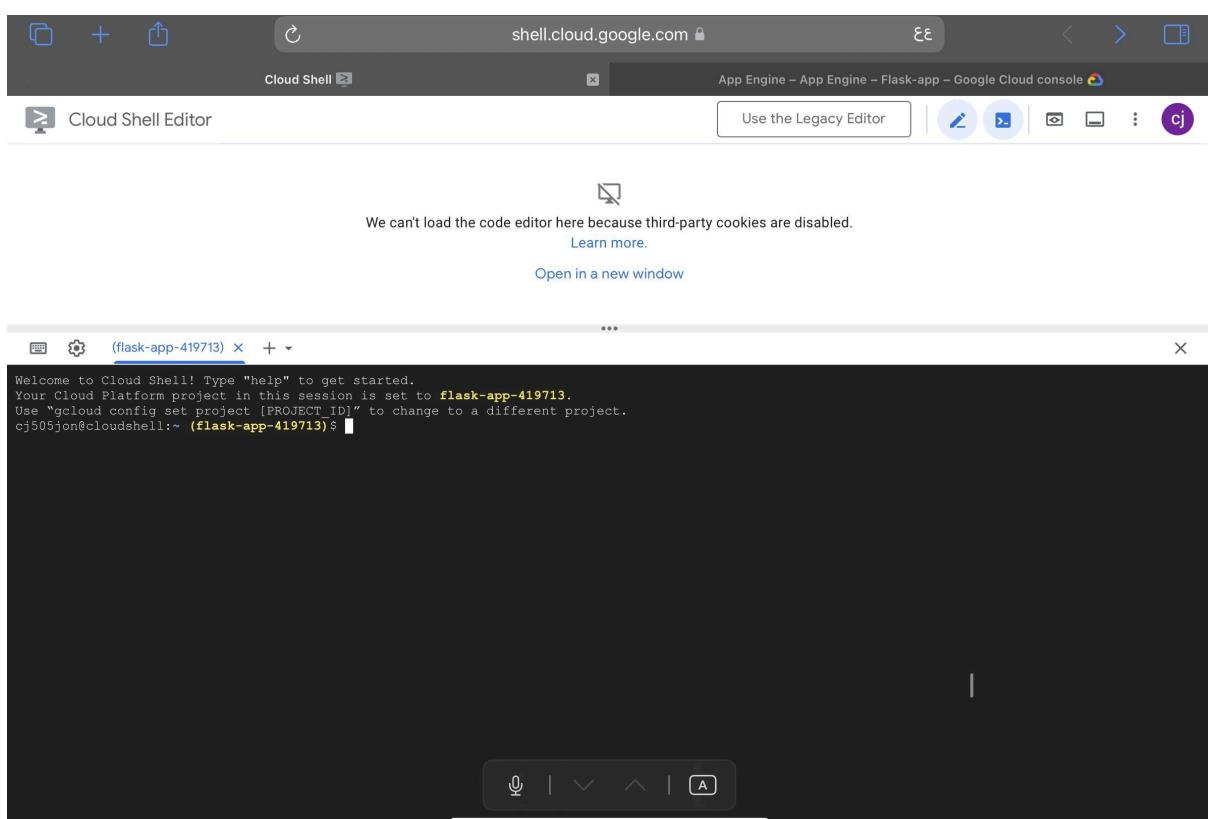
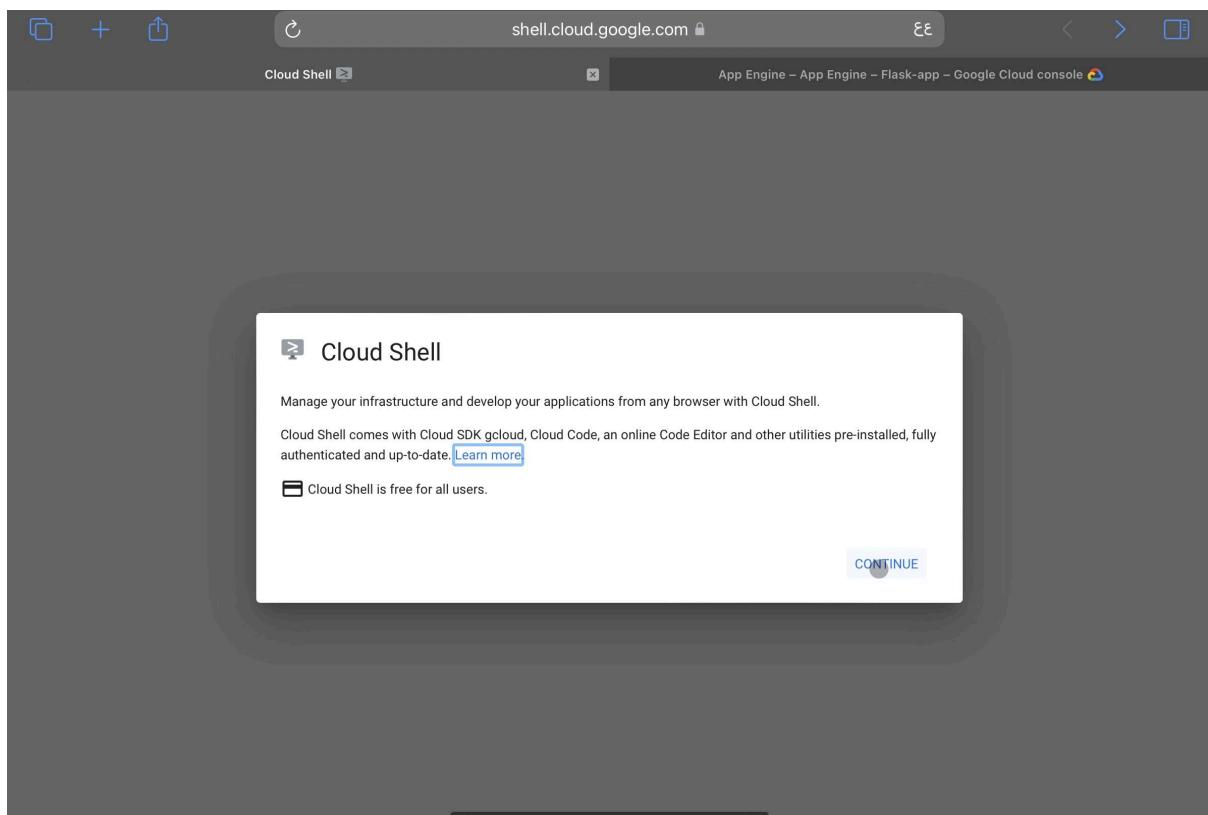
Screenshot of the Google Cloud App Engine 'Create app' page, showing the 'Identity and API access' section. The 'Select a service account' dropdown is set to 'App Engine default service account'. A note below it says: 'If no service account is selected the default App Engine service account will be used.' A large blue 'NEXT' button is visible at the bottom. The rest of the interface is identical to the previous screenshot, including the sidebar and the world map.

The screenshot shows the Google Cloud App Engine dashboard at console.cloud.google.com. The left sidebar lists various services: Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. Below these are Release Notes and a back arrow. The main content area has tabs for 'App Engine' and 'Get started'. Under 'Get started', there's a 'Resources' section with dropdowns for Language (Python) and Environment (Standard), and links to documentation and GitHub. To the right is a 'Deploy with Google Cloud SDK' section with buttons for 'DOWNLOAD THE CLOUD SDK' and 'INITIALIZE YOUR SDK' (with a command \$ gcloud init), and another for 'DEPLOY TO APP ENGINE' (with a command \$ gcloud app deploy).

Cloud Shell:
After we create our first application

The screenshot shows the Google Cloud App Engine dashboard. The left sidebar contains links for Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. Below these are Release Notes and a back arrow. The main content area displays a "Welcome to App Engine" message: "Your App Engine application has been created. Let us help you deploy to your application by pointing you at the relevant resources based on your programming language." A blue "GET STARTED" button is at the bottom.

This screenshot is identical to the one above, but it includes a vertical sidebar menu on the right. The menu items are: Open Duet AI, Activate Cloud Shell (with a toggle switch), Get support (Live and self-service support options), Browse documentation (View architecture guides, View help FAQs, Start a tutorial), Configure keyboard shortcuts, Send feedback, Billing account management (Free trial), Payment method, Preferences, Downloads (Cloud Partners), Terms of service, Privacy, and Project settings.



After we open the shell cmd: `git clone https://github.com/CJ-2/upload-app.git`

The screenshot shows a browser window for shell.cloud.google.com. At the top, there's a toolbar with icons for refresh, search, and tabs. The address bar shows 'shell.cloud.google.com'. Below the address bar, it says 'Cloud Shell' and 'App Engine – App Engine – Flask-app – Google Cloud console'. A user icon 'cj' is in the top right. A message at the top center says 'We can't load the code editor here because third-party cookies are disabled.' with a 'Learn more.' link and an 'Open in a new window' button. The main area is a terminal window titled '(flask-app-419713)'. It contains the command: `cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git`. The terminal has a dark theme with light-colored text and a light gray background.

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
```

cmd: `ls`

The screenshot shows a browser window for shell.cloud.google.com. The setup is identical to the first one, with the same toolbar, address bar, and user icon. The message about disabled cookies is still present. The terminal window now shows the output of the 'git clone' command, indicating a successful cloning of the 'upload-app' repository. The terminal has a dark theme with light-colored text and a light gray background.

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
```

cmd: `cd upload-app`

We can't load the code editor here because third-party cookies are disabled.
[Learn more.](#)
[Open in a new window](#)

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
```

cmd: **ls**

We can't load the code editor here because third-party cookies are disabled.
[Learn more.](#)
[Open in a new window](#)

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
cj505jon@cloudshell:/upload-app (flask-app-419713)$ ls
README.md storage
cj505jon@cloudshell:/upload-app (flask-app-419713)$
```

cmd: **cd storage**

Cloud Shell Editor

We can't load the code editor here because third-party cookies are disabled.
[Learn more.](#)

Open in a new window

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
README.md storage
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ cd storage
```

cmd: `ls`

Cloud Shell Editor

We can't load the code editor here because third-party cookies are disabled.
[Learn more.](#)

Open in a new window

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
README.md storage
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ cd storage
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ ls
main.py templates
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$
```

cmd: `python main.py`

We can't load the code editor here because third-party cookies are disabled.
Learn more.

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README.md upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
main.py templates
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ python main.py
```

We can't load the code editor here because third-party cookies are disabled.
Learn more.

```
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README.md upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
main.py templates
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ python main.py
 * Serving Flask app 'main'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.88.0.4:5000
Press CTRL+c to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-392-381
```

Click on web preview and follow the steps.

The screenshot shows a Cloud Shell Editor interface. At the top, there are tabs for 'Cloud Shell' and 'App Engine – App Engine – Flask-app – Google Cloud console'. Below the tabs, there's a toolbar with icons for code editor, legacy editor, and various file operations. A user profile icon 'cj' is visible. A message box states: 'We can't load the code editor here because third-party cookies are disabled.' with a 'Learn more.' link and an 'Open in a new window' button. The main terminal window displays a command-line session:

```
cj505jon@cloudshell:~ (flask-app-419713)$ clear
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
README.md storage
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ python main.py
 * Serving Flask app 'main'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.88.0.4:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-392-381
```

The screenshot shows the same Cloud Shell Editor interface as the first one, but with a context menu open over the message box. The menu options are: 'Preview on port 8080', 'Change port', and 'About web preview'. The terminal window below shows the same command-line session as the first screenshot.

```
cj505jon@cloudshell:~ (flask-app-419713)$ clear
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/CJ-2/upload-app.git
Cloning into 'upload-app'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
README.md storage
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ python main.py
 * Serving Flask app 'main'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.88.0.4:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-392-381
```

The screenshot shows a Google Cloud Shell interface with two main windows.

Top Window (Terminal Session):

- URL: shell.cloud.google.com
- Cloud Shell Editor tab is active.
- Message: "We can't load the code editor here because third-party cookies are disabled." with a "Learn more." link and "Open in a new window" button.
- Terminal command history:

```
cj505jon@cloudshell:~ (flask-app-419713)$ clear
cj505jon@cloudshell:~ (flask-app-419713)$ git clone https://github.com/cj505jon/upload-app.git
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 0), reused 0 (delta 0), pack-received 0 (delta 0), pack-delta-received 0 (delta 0), pack-delta-reuse 0 (delta 0)
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt  upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app/
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
README.md  storage
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ python main.py
  * Serving Flask app 'main'
  * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on all addresses (0.0.0.0)
  * Running on http://127.0.0.1:5000
  * Running on http://10.88.0.4:5000
Press CTRL+C to quit
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 123-392-381
```
- Port number input field: "5000".
- Buttons: CANCEL and CHANGE AND PREVIEW.

Bottom Window (File Upload Interface):

- URL: 3f-8d5f-4e4b-9e90-7e6ebc239142.cs-europe-west4-fycr.cloudshell.dev
- Upload new File button.
- Cloud Shell tab is active.
- Message: "لم يتم تحديد أي ملف" (No file selected) and "اخبار ملف" (File info).
- Upload button.
- Explore Uploaded Files link.

Return to shell and cmd: **CTRL+C**

shell.cloud.google.com

Cloud Shell Editor

We can't load the code editor here because third-party cookies are disabled.
Learn more.

Open in a new window

```
remote: Total 33 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 8.27 MiB | 15.01 MiB/s, done.
Resolving deltas: 100% (5/5), done.
cj505jon@cloudshell:~ (flask-app-419713)$ ls
README-cloudshell.txt upload-app
cj505jon@cloudshell:~ (flask-app-419713)$ cd upload-app
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ ls
README.md storage
cj505jon@cloudshell:~/upload-app (flask-app-419713)$ cd storage
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ ls
main.py templates
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ python main.py
 * Serving Flask app 'main'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.88.0.4:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-392-381
127.0.0.1 - - [08/Apr/2024 14:06:29] "GET /?authuser=1&redirectedPreviously=true HTTP/1.1" 200 -
127.0.0.1 - - [08/Apr/2024 14:06:29] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [08/Apr/2024 14:08:31] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [08/Apr/2024 14:08:32] "GET /uploads/IMG_9220.mov HTTP/1.1" 200 -
127.0.0.1 - - [08/Apr/2024 14:08:32] "GET /uploads/IMG_9220.mov HTTP/1.1" 206 -
127.0.0.1 - - [08/Apr/2024 14:08:32] "GET /uploads/IMG_9220.mov HTTP/1.1" 206 -
127.0.0.1 - - [08/Apr/2024 14:08:37] "GET /explore HTTP/1.1" 200 -
```

cmd: **clear**

shell.cloud.google.com

Cloud Shell Editor

We can't load the code editor here because third-party cookies are disabled.
Learn more.

Open in a new window

```
cj505jon@cloudshell:~/upload-app/storage/uploads (flask-app-419713)$ clear
```

cmd: ls

The screenshot shows the Google Cloud Shell interface. At the top, there's a navigation bar with tabs like 'Cloud Shell Editor', 'Explore Files', 'Cloud Shell', and 'App Engine'. Below the navigation bar is a message: 'We can't load the code editor here because third-party cookies are disabled.' with a 'Learn more.' link and an 'Open in a new window' button. The main area is a terminal window titled '(flask-app-419713)'. It displays the output of the 'ls' command:

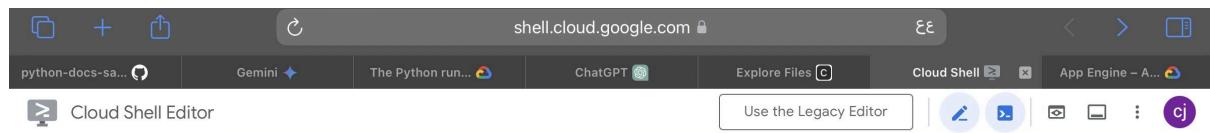
```
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ ls
app.yaml main.py requirements.txt templates uploads
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$
```

cmd: gcloud app deploy

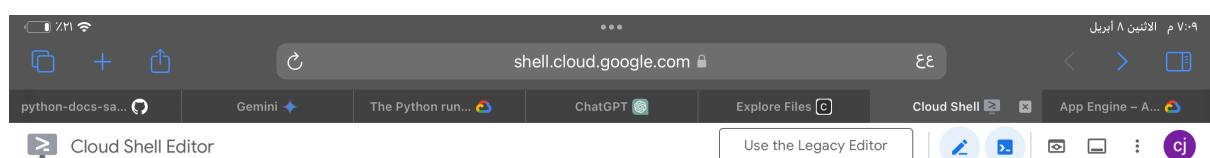
The screenshot shows the Google Cloud Shell interface. At the top, there's a navigation bar with tabs like 'Cloud Shell Editor', 'Explore Files', 'Cloud Shell', and 'App Engine'. Below the navigation bar is a message: 'We can't load the code editor here because third-party cookies are disabled.' with a 'Learn more.' link and an 'Open in a new window' button. The main area is a terminal window titled '(flask-app-419713)'. It displays the output of the 'gcloud app deploy' command:

```
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ ls
app.yaml main.py requirements.txt templates uploads
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ gcloud app deploy
```

cmd: enter y



```
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ ls  
app.yaml main.py requirements.txt templates uploads  
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ gcloud app deploy  
Services to deploy:  
  
descriptor:      [/home/cj505jon/upload-app/storage/app.yaml]  
source:         [/home/cj505jon/upload-app/storage]  
target project: [flask-app-419713]  
target service: [default]  
target version: [20240408t160909]  
target url:     [https://flask-app-419713.uc.r.appspot.com]  
target service account: [flask-app-419713@appspot.gserviceaccount.com]  
  
Do you want to continue (Y/n)? y
```



```
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ ls  
app.yaml main.py requirements.txt templates uploads  
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ gcloud app deploy  
Services to deploy:  
  
descriptor:      [/home/cj505jon/upload-app/storage/app.yaml]  
source:         [/home/cj505jon/upload-app/storage]  
target project: [flask-app-419713]  
target service: [default]  
target version: [20240408t160909]  
target url:     [https://flask-app-419713.uc.r.appspot.com]  
target service account: [flask-app-419713@appspot.gserviceaccount.com]  
  
Do you want to continue (Y/n)? y  
  
Beginning deployment of service [default]...  
Uploading 2 files to Google Cloud Storage  
50%  
100%  
100%  
File upload done.  
Updating service [default]...working..
```

The screenshot shows a Cloud Shell Editor window with the title '(flask-app-419713)'. The terminal output displays the deployment process for a service named 'default' with target version '[20240408t160909]'. It shows files being uploaded to Google Cloud Storage, progress from 50% to 100%, and the deployment completed successfully at '[https://flask-app-419713.uc.r.appspot.com]'. The user is prompted to continue with 'y'.

```
target version: [20240408t160909]
target url: [https://flask-app-419713.uc.r.appspot.com]
target service account: [flask-app-419713@appspot.gserviceaccount.com]

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

Beginning deployment of service [default]...
Uploading 2 files to Google Cloud Storage
50%
100%
100%
File upload done.
Updating service [default]...working...
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://flask-app-419713.uc.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
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$
```

cmd: **gcloud app browse**

The screenshot shows a Cloud Shell Editor window with the title '(flask-app-419713)'. The terminal output displays the deployment process for a service named 'default' with target version '[20240408t160909]'. It shows files being uploaded to Google Cloud Storage, progress from 50% to 100%, and the deployment completed successfully at '[https://flask-app-419713.uc.r.appspot.com]'. The user is prompted to continue with 'y'.

```
target version: [20240408t160909]
target url: [https://flask-app-419713.uc.r.appspot.com]
target service account: [flask-app-419713@appspot.gserviceaccount.com]

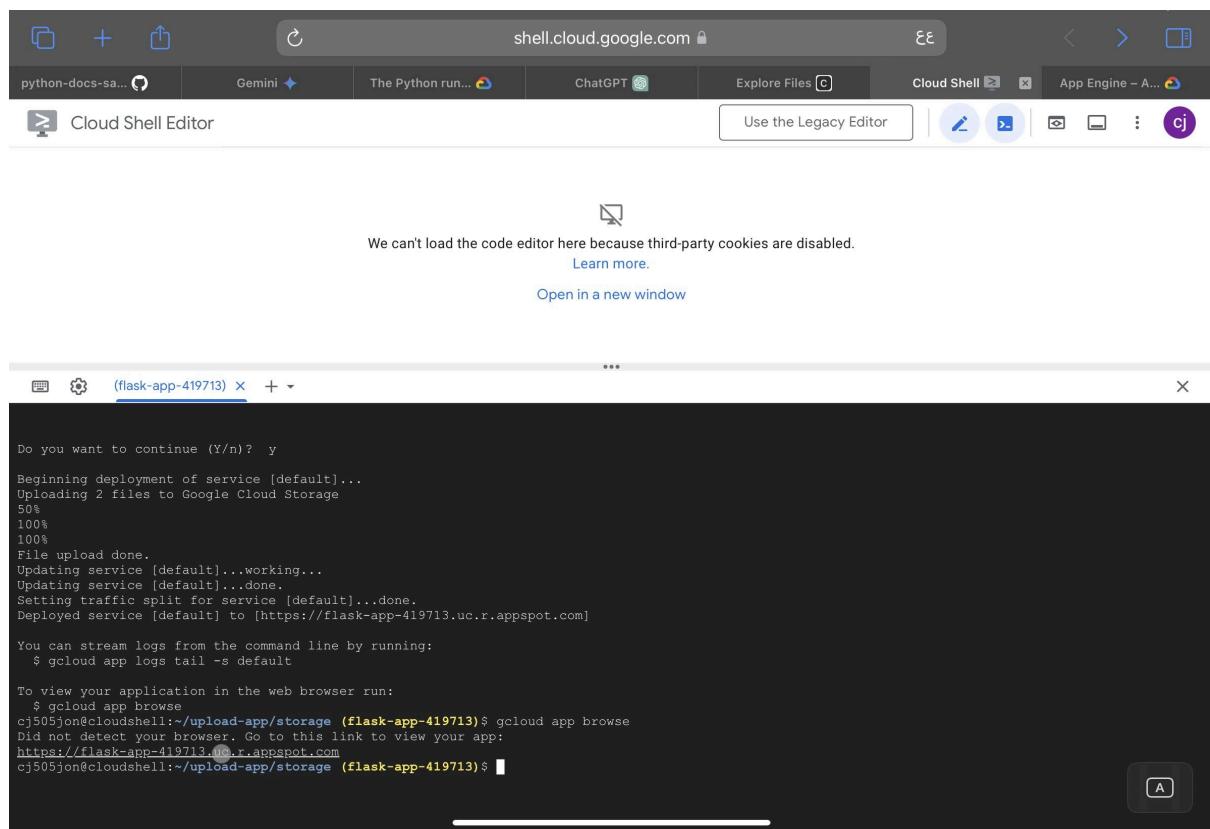
Do you want to continue (Y/n)? y

Beginning deployment of service [default]...
Uploading 2 files to Google Cloud Storage
50%
100%
100%
File upload done.
Updating service [default]...working...
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://flask-app-419713.uc.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
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ gcloud app browse
```

After that click on the link.



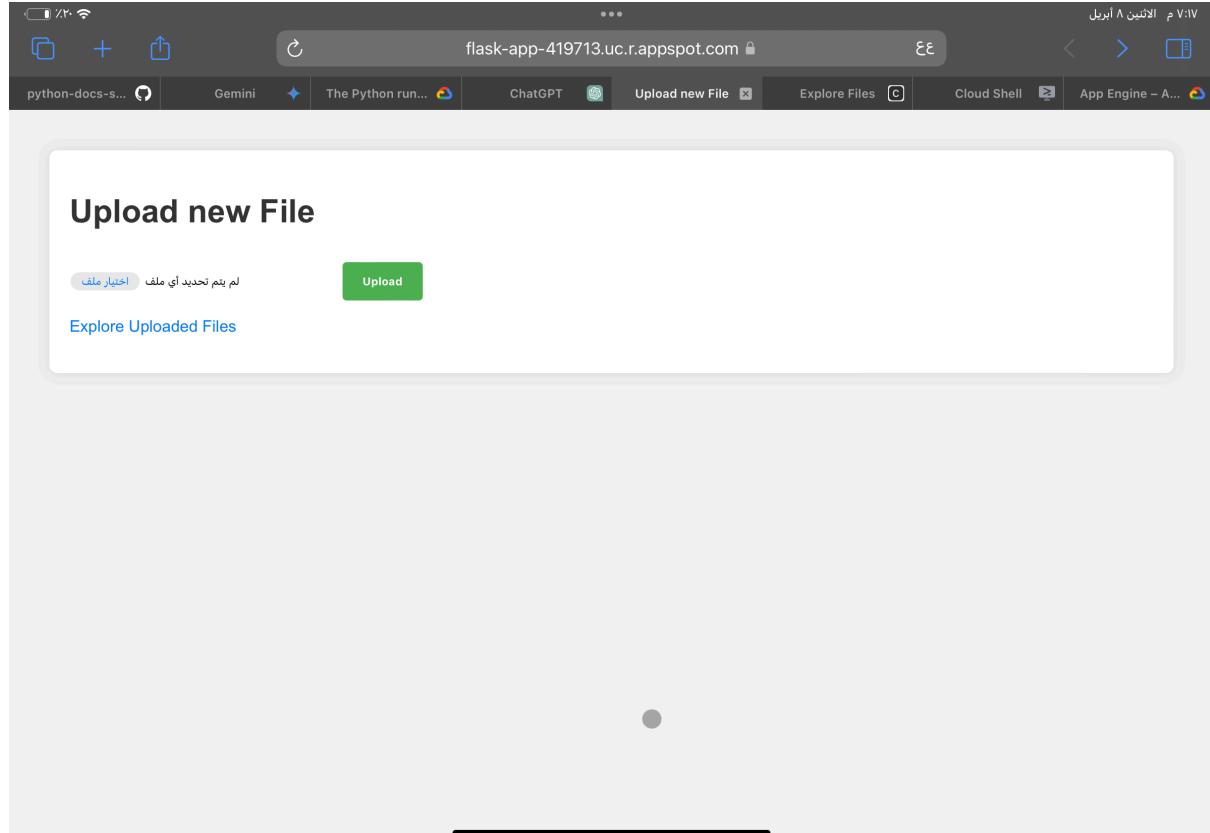
We can't load the code editor here because third-party cookies are disabled.
[Learn more.](#)
[Open in a new window](#)

```
Do you want to continue (Y/n)? y
Beginning deployment of service [default]...
Uploading 2 files to Google Cloud Storage
50%
100%
100%
File upload done.
Updating service [default]...working...
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://flask-app-419713.uc.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
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://flask-app-419713.uc.r.appspot.com
cj505jon@cloudshell:~/upload-app/storage (flask-app-419713)$
```

And now your app is deployed congratulations 🎉.



Upload new File

لم يتم تحديد أي ملف [اختر ملف](#)

[Upload](#)

[Explore Uploaded Files](#)

Go back to dashboard and select Versions.

The screenshot shows the Google Cloud App Engine dashboard. On the left, a sidebar menu lists various options: Dashboard, Services, Versions (which is selected and highlighted in grey), Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. Below this is a link to Release Notes. The main content area is titled "Welcome to App Engine" and includes a message: "Your App Engine application has been created. Let us help you deploy to your application by pointing you at the relevant resources based on your programming language." A blue "GET STARTED" button is at the bottom of this section.

Here's your version.

The screenshot shows the Google Cloud App Engine "Versions" page. The sidebar menu is identical to the previous screenshot. The main content area is titled "Versions" and includes a "Filter versions" section. A table displays the following data:

| | Version | Status | Traffic Allocation | Instances | Runtime | Environment | S |
|--------------------------|---------------------------------|---------|--------------------------------------|-----------|----------|-------------|---|
| <input type="checkbox"/> | 20240408t160909 | Serving | <div style="width: 100%;">100%</div> | 1 | python39 | Standard | 7 |

I hope that the project will help you 🙏.