

Your First App on Google App Engine - Easy Tutorial

This tutorial will walk you through creating and hosting your first (and simple) app on Google App Engine.

What We're Doing

We're going to:

- Set up Google App Engine (it's free to start)
- Install some tools on your computer
- Make a simple web app
- Put it on the internet
- See it work

Let's start.

Step 1: Starting with App Engine

Screenshot of the Google Cloud Console showing the App Engine dashboard for the "My First Project" project.

The dashboard includes a sidebar with links to Services, Versions, Instances, Task queues, Cron jobs, Firewall rules, Quotas, Memcache, Search, and Settings. A callout box highlights the "Release Notes" link at the bottom of the sidebar.

The main content area displays the "App Engine" section with a welcome message: "Welcome to App Engine. Build scalable apps in any language on Google's infrastructure." It features a "Create Application" button and a "Try Cloud Run" button. A decorative graphic of a stylized mountain or bridge is shown above the text.

The top navigation bar shows the URL "console.cloud.google.com / App Engine – App Engine – My First Project" and various Google Cloud navigation icons.

This is Google Cloud Console looking at the App Engine. This is where you basically manage everything.

Step 2: Creating Your App

console.cloud.google.com / Create app – App Engine – My First Project

Google Cloud My First Project

- App Engine / Dashboard / Start / Create

Dashboard Create app Learn

Services Versions Instances Task queues Cron jobs Firewall rules Quotas Memcache Search Settings

① Configure application — ② Get started

Region

Select a region for your App Engine application. Please remember, once selected the region is permanently tied to the project.

Keyboard shortcuts | Map data ©2025 | Terms

Select a region *

us-central

Identity and API access

Select a service account

If no service account is selected the default App Engine service account will be used.

Next

Release Notes

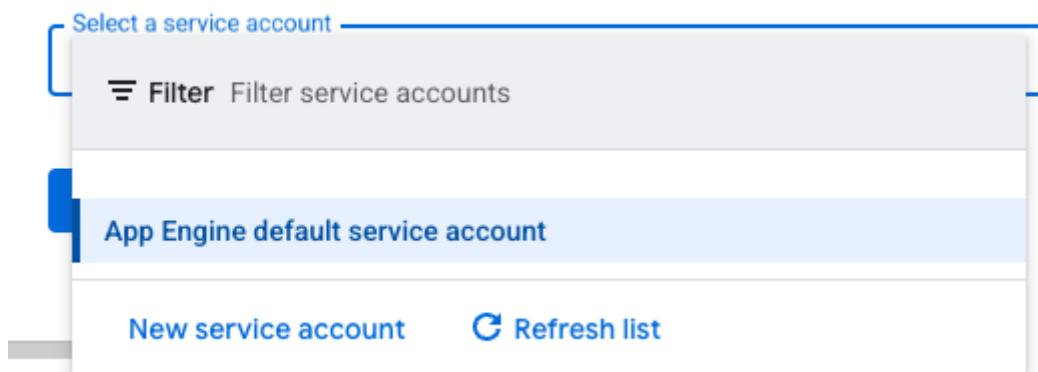
Now we're creating the app.

IMPORTANT: You need to select a region where your app will be hosted (like US, Europe, Asia, etc). **You cannot change this later**, so choose the region closest to where you or your users are located.

You also need to select a service account. Just use the default one.

Step 3: Picking the Service Account

Identity and API access



A dialog box appears asking about service accounts. Service accounts define what permissions your app has.

Select "App Engine default service account". It's already configured and ready to use.

Step 4: Creating... Please Wait

The screenshot shows the Google Cloud Platform interface for creating an application. The top navigation bar includes links for 'Dashboard', 'My First Project', and various system icons like search, refresh, and notifications. The main content area is titled 'Create app' and displays the 'Get started' step under 'Configure application'. A message box states 'Cloud Run is available in all the Google Cloud supported regions.' with a 'Try Cloud Run' button. On the left, a sidebar lists various services: Services, Versions, Instances, Task queues, Cron jobs, Firewall rules, Quotas, Memcache, Search, and Settings. At the bottom of the sidebar, there's a link to 'Release Notes'.

console.cloud.google.com / Create app – App Engine – My First Project

Google Cloud My First Project

- App Engine / Dashboard / Start / Create

Dashboard

Create app

Configure application —

2 Get started

Cloud Run is available in all the Google Cloud supported regions.

Try Cloud Run

Creating application...

Services

Versions

Instances

Task queues

Cron jobs

Firewall rules

Quotas

Memcache

Search

Settings

Release Notes

The application is being created. This takes a minute or two.

Step 5: App Created! Now What?

The screenshot shows the Google Cloud Platform App Engine dashboard for a project named "My First Project". The left sidebar lists various project management options like Services, Versions, Instances, Task queues, Cron jobs, Firewall rules, Quotas, Memcache, Search, and Settings. The main content area has a "Get started" section with a green success message: "Your App Engine application has been created. Choose the deployment method that works best for you." Below this is a "Resources" section where the user can select the language (set to Python) and environment (set to Standard). It also includes links to documentation and GitHub code samples. A "Deploy with Google Cloud SDK" section provides instructions for initializing the SDK with the command "\$ gcloud init" and deploying with "\$ gcloud app deploy". At the bottom, there's a note about Cloud Run and a "Try Cloud Run" button.

console.cloud.google.com / Get started – App Engine – My First Project

Google Cloud My First Project

- App Engine / Dashboard / Start / Get started

Dashboard

Get started

Learn

Services

Versions

Instances

Task queues

Cron jobs

Firewall rules

Quotas

Memcache

Search

Settings

Your App Engine application has been created. Choose the deployment method that works best for you.

Resources

Language: Python

Environment: Standard

Read App Engine Python Standard Environment [Documentation](#).

Visit [Github](#) for Python Standard Environment code samples.

Deploy with Google Cloud SDK

Download the Cloud SDK

Initialize your SDK

```
$ gcloud init
```

Deploy to App Engine

```
$ gcloud app deploy
```

Build web applications in any language, any framework supported by App Engine on Cloud Run. Cloud Run supports a variety of workloads from simple functions to powerful services and jobs.

Try Cloud Run

Your App Engine application is ready.

It shows:

- Language: Python (what we're using)

- Environment: Standard

To deploy your code, you need the Google Cloud SDK installed on your computer. This is software that lets you interact with Google Cloud from your terminal.

The commands shown (`gcloud init` and `gcloud app deploy`) are what we'll use later.

Step 6: Getting the SDK

The screenshot shows the Google Cloud homepage with the URL "cloud.google.com / Cloud SDK - Libraries and Command Line Tools | Google Cloud". The navigation bar includes the Google Cloud logo, a search icon, a three-dot menu, and a profile picture. A blue "Contact Us" button is visible on the right.

Cloud SDK

Libraries and tools for interacting with Google Cloud products and services. **Cloud SDK is available at no charge** for users with a Google Cloud account.

[Install Google Cloud CLI](#)

[Contact sales](#)

- ✓ Integrate APIs using Client Libraries for [Java](#), [C++](#), [Python](#), [Node.js](#), [Ruby](#), [Go](#), [.NET](#), [PHP](#), [Rust](#), and [ABAP](#)
- ✓ Script or interact with cloud resources at scale using the [Google Cloud CLI](#)
- ✓ Accelerate local development with emulators for [Pub/Sub](#), [Spanner](#), [Bigtable](#), and [Datastore](#)

A video player interface with the title "What is the Google Cloud SDK?", a play button, and a duration of "3:00". Below the video, there's a thumbnail image featuring the Google Cloud logo and some abstract shapes.

KEY FEATURES

Key features

SDK Client Libraries for popular programming languages

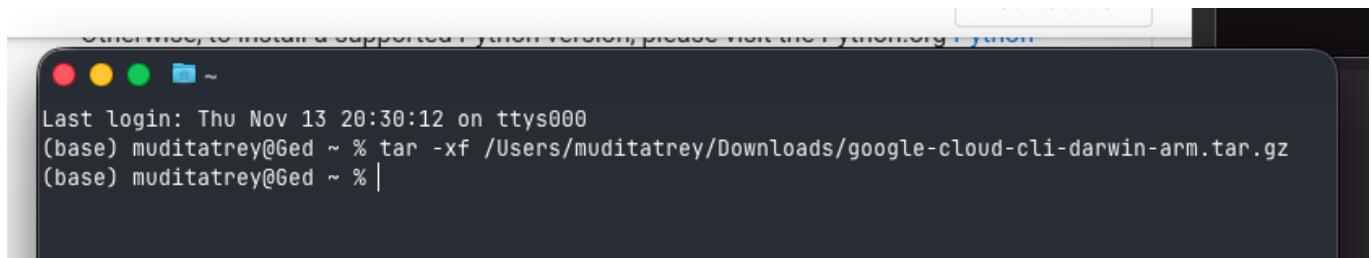
Cloud SDK provides language-specific Cloud Client Libraries supporting each language's natural conventions and styles. This allows you to interact with Google Cloud APIs from your favorite programming language.



This is the download page for Google Cloud SDK. It's a set of tools that let you interact with Google Cloud from your computer's terminal.

Download it for your operating system. If you're on Mac with an M1/M2 chip, get the "darwin-arm" version. If you're on Windows or Linux, get the appropriate version.

Step 7: Extracting the Download



A screenshot of a macOS terminal window. The title bar says "Otherwise, to install a supported, older version, please visit the Python.org Python page". The window itself shows a command-line session:

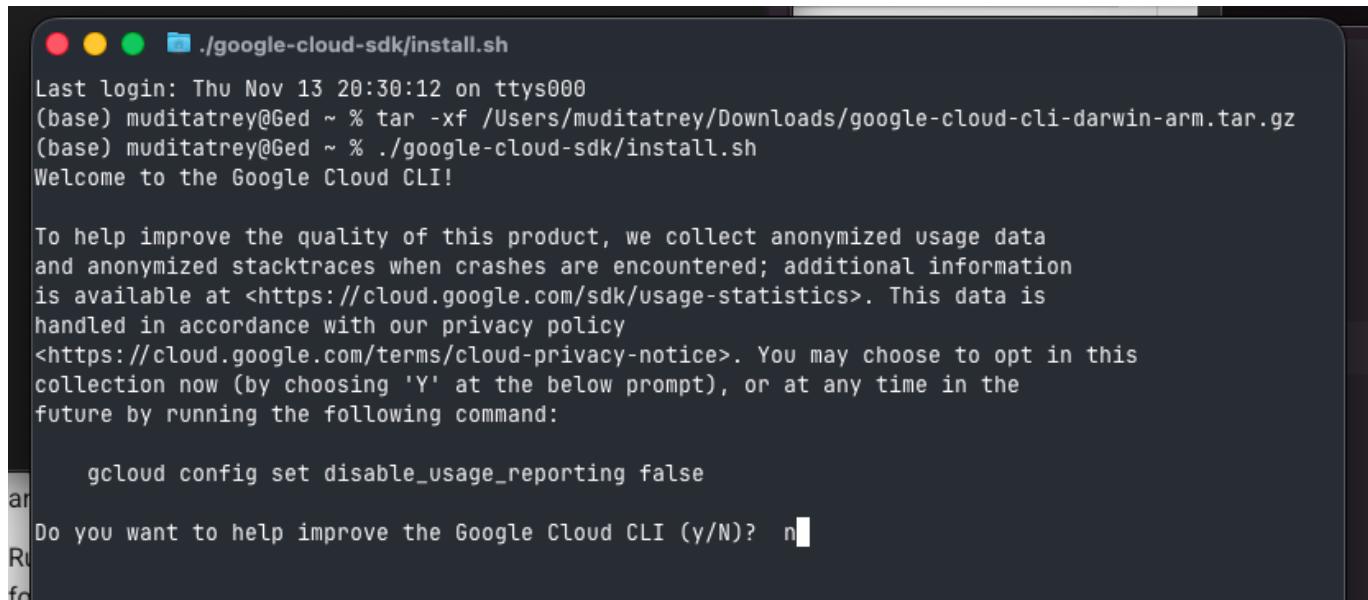
```
Last login: Thu Nov 13 20:30:12 on ttys000
(base) muditatrey@Ged ~ % tar -xf /Users/muditatrey/Downloads/google-cloud-cli-darwin-arm.tar.gz
(base) muditatrey@Ged ~ % |
```

After downloading, you need to extract the files. Run this command:

```
tar -xf /Users/muditatrey/Downloads/google-cloud-cli-darwin-arm.tar.gz
```

This creates a folder called **google-cloud-sdk** in your home directory where all the tools are located.

Step 8: Installing the SDK



```
./google-cloud-sdk/install.sh
Last login: Thu Nov 13 20:30:12 on ttys000
(base) muditatrey@Ged ~ % tar -xf /Users/muditatrey/Downloads/google-cloud-cli-darwin-arm.tar.gz
(base) muditatrey@Ged ~ % ./google-cloud-sdk/install.sh
Welcome to the Google Cloud CLI!

To help improve the quality of this product, we collect anonymized usage data
and anonymized stacktraces when crashes are encountered; additional information
is available at <https://cloud.google.com/sdk/usage-statistics>. This data is
handled in accordance with our privacy policy
<https://cloud.google.com/terms/cloud-privacy-notice>. You may choose to opt in this
collection now (by choosing 'Y' at the below prompt), or at any time in the
future by running the following command:

    gcloud config set disable_usage_reporting false

Do you want to help improve the Google Cloud CLI (y/N)? n
```

Run the install script:

```
./google-cloud-sdk/install.sh
```

It will ask if you want to help improve the tool by sending anonymous usage data. You can answer "n" for no or "y" for yes. This is optional.

Step 9: What Gets Installed

```
./google-cloud-sdk/install.sh

| Not Installed | gke-gcloud-auth-plugin | gke-gcloud-auth-plugin
|   3.3 MiB |
| Not Installed | istioctl | istioctl
|   26.6 MiB |
| Not Installed | kpt | kpt
|   14.5 MiB |
| Not Installed | kubectl | kubectl
|   < 1 MiB |
| Not Installed | kubectl-oidc | kubectl-oidc
|   20.9 MiB |
| Not Installed | pkg | pkg
|
| Installed | BigQuery Command Line Tool | bq
|   1.8 MiB |
| Installed | Cloud Storage Command Line Tool | gsutil
|   12.4 MiB |
| Installed | Google Cloud CLI Core Libraries | core
|   23.2 MiB |
| Installed | Google Cloud CRC32C Hash Tool | gcloud-crc32c
|   1.4 MiB |

To install or remove components at your current Google Cloud CLI version [547.0.0], run:
$ gcloud components install COMPONENT_ID
$ gcloud components remove COMPONENT_ID

To update your Google Cloud CLI installation to the latest version [547.0.0], run:
$ gcloud components update

Modify profile to update your $PATH and enable shell command completion?

Do you want to continue (Y/n)? y
./google-cloud-sdk/install.sh --help
```

The installer shows all the components. Some are installed by default:

- **bq** - for BigQuery
- **gsutil** - for Cloud Storage
- **core** - the main libraries
- **gcloud-crc32c** - a hash tool

The default installation is sufficient for what we need.

Step 10: Adding to Your PATH

```
./google-cloud-sdk/install.sh

| 14.5 MiB |
| Not Installed | kubectl | kubectl
| < 1 MiB |
| Not Installed | kubectl-oidc | kubectl-oidc
| 20.9 MiB |
| Not Installed | pkg | pkg
|
| Installed | BigQuery Command Line Tool | bq
| 1.8 MiB |
| Installed | Cloud Storage Command Line Tool | gsutil
| 12.4 MiB |
| Installed | Google Cloud CLI Core Libraries | core
| 23.2 MiB |
| Installed | Google Cloud CRC32C Hash Tool | gcloud-crc32c
| 1.4 MiB |

To install or remove components at your current Google Cloud CLI version [547.0.0], run:
$ gcloud components install COMPONENT_ID
$ gcloud components remove COMPONENT_ID

To update your Google Cloud CLI installation to the latest version [547.0.0], run:
$ gcloud components update

Modify profile to update your $PATH and enable shell command completion?

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

The Google Cloud SDK installer will now prompt you to update an rc file to bring the Google Cloud
CLIs into your environment.

Enter a path to an rc file to update, or leave blank to use [/Users/muditatrey/.zshrc]:
```

The installer asks if it should modify your shell profile so you can run `gcloud` from anywhere. Answer "y" for yes.

If you're on Mac with zsh (the default shell), just press enter when it asks for the rc file path. It will use `.zshrc` automatically.

Important: After this finishes, close your terminal and open a new one so the changes take effect.

Step 11: Authentication Done

The screenshot shows a web browser window with the Google Cloud logo at the top left. The top navigation bar includes links for 'Technology areas', 'Cross-product', 'Search', 'Language', 'Console', and a user profile icon. Below the navigation is a header bar with 'Google Cloud SDK' on the left and a 'Contact Us' button on the right. The main content area has a breadcrumb trail 'Application development > Google Cloud SDK'. The main message is 'You are now authenticated with the gcloud CLI!' with a 'Send feedback' button. A sidebar on the left is titled 'On this page' and lists 'Information about command-line tools and client libraries', 'Tutorials', and 'Feedback'. Below the main message, a note says 'The authentication flow has completed successfully. You may close this window, or check out the resources below.' Under the heading 'Information about command-line tools and client libraries', there are three paragraphs: one about the gcloud CLI guide, one about access to various Google Cloud services via the gcloud CLI, and one about client libraries explained.

Application development > Google Cloud SDK

You are now authenticated with the gcloud CLI!

[Send feedback](#)

On this page

- Information about command-line tools and client libraries
- Tutorials
- Feedback

The authentication flow has completed successfully. You may close this window, or check out the resources below.

Information about command-line tools and client libraries

To learn more about Google Cloud CLI commands, see the [gcloud CLI guide](#).

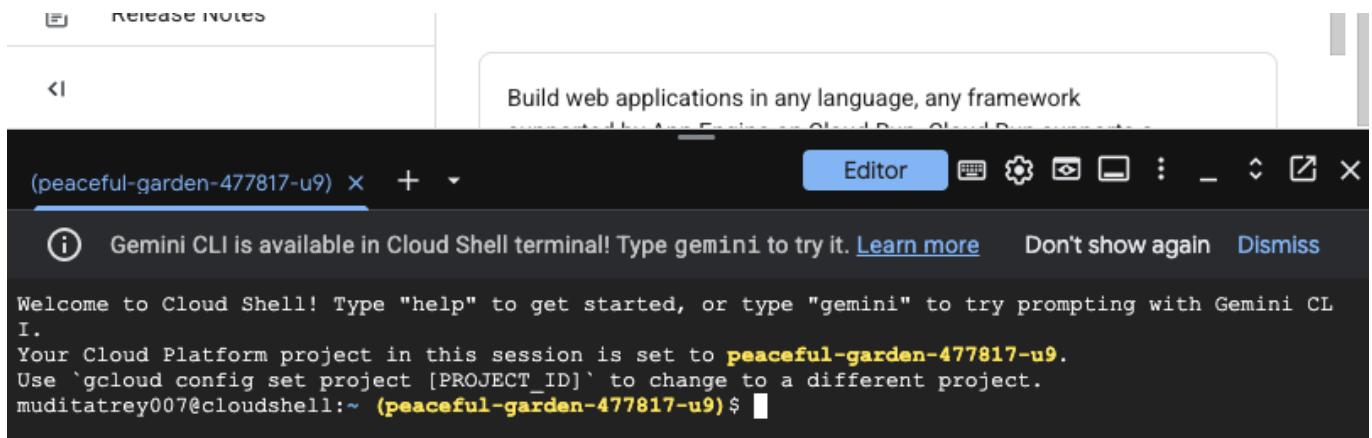
To learn more about the command-line tools for App Engine, Compute Engine, Cloud Storage, BigQuery, Cloud SQL, and Cloud DNS (which are all bundled with the gcloud CLI), see [Accessing services with the gcloud CLI](#).

If you're a client application developer and want to find out more about accessing Google Cloud services with a programming language or framework, see [Client Libraries Explained](#).

This page appears in your browser confirming you're authenticated. You've given your computer permission to access your Google Cloud account.

You can close this tab.

Step 12: Cloud Shell (Optional)



This is Cloud Shell. It's a terminal that runs in your browser. You don't need this since we installed the SDK on your computer, but it's useful to know it exists.

You can use Cloud Shell if you ever need to run commands without accessing your local machine.

Step 13: Choosing Your Project

```
./google-cloud-sdk/bin/gcloud init

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=MiypnxEUmjTZ2LTiETU4Yk0Iz8ruV&access_type=offline&code_challenge=Lj06PY7mW1CzkIclLWrXtN5e-0eEdVetCTdokk92Sxw&code_challenge_method=S256

You are signed in as: [muditatrey007@gmail.com].

Pick cloud project to use:
[1] indigo-terra-335116
[2] jarvis-5b558
[3] jarvis-82c4d
[4] optimal-iris-442104-d1
[5] peaceful-garden-477817-u9
[6] symbolic-app-446214-s5
[7] Enter a project ID
[8] Create a new project
Please enter numeric choice or text value (must exactly match list item): 5

Your current project has been set to: [peaceful-garden-477817-u9].

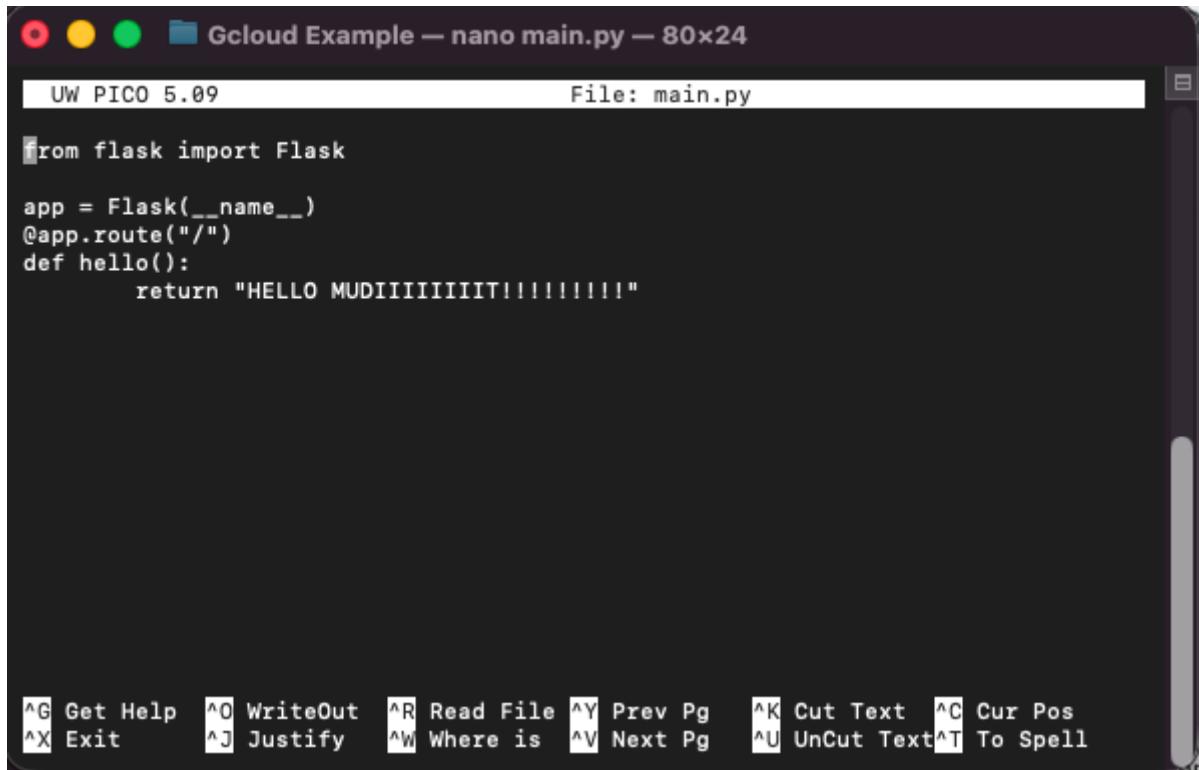
Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the
https://console.developers.google.com/apis page.
```

When you run `gcloud init`, it shows all your Google Cloud projects and asks which one to use.

In this example, option 5 was selected: "peaceful-garden-477817-u9". Your project name will be different.

Type the number of your project and press enter.

Step 14: Writing Your App Code



The screenshot shows a terminal window titled "Gcloud Example — nano main.py — 80x24". The window has three colored tabs at the top: red, yellow, and green. The title bar also displays "File: main.py". The terminal content is a Python script:

```
UW PICO 5.09
File: main.py

from flask import Flask

app = Flask(__name__)
@app.route("/")
def hello():
    return "HELLO MUDIIIIIIIT!!!!!!!"
```

At the bottom of the terminal, there is a menu bar with various keyboard shortcuts:

**^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell**

Now we write the actual app. Create a file called `main.py`:

```
from flask import Flask

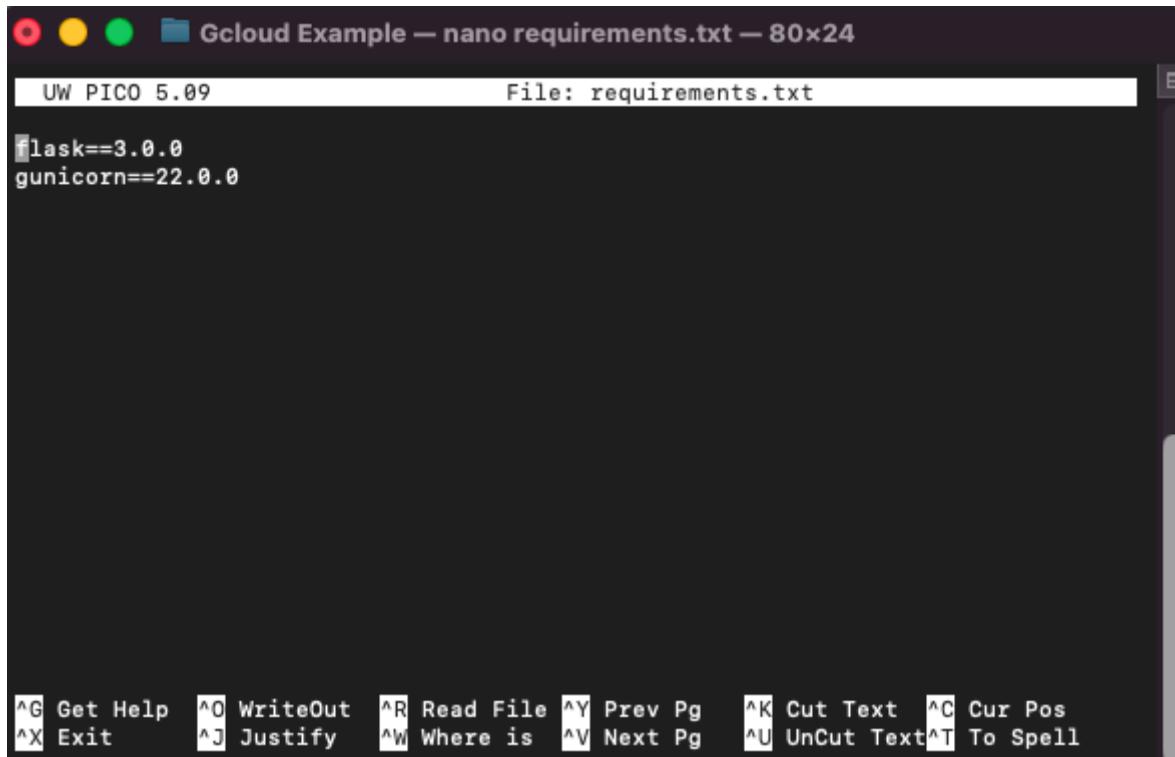
app = Flask(__name__)

@app.route("/")
def hello():
    return "HELLO MUDIIIIIIIT!!!!!!!"
```

This is a simple Flask app that displays a message when you visit it. Flask is a web framework for Python that makes building web apps easier.

You can change the message to whatever you want.

Step 15: Listing Dependencies



```
flask==3.0.0
gunicorn==22.0.0
```

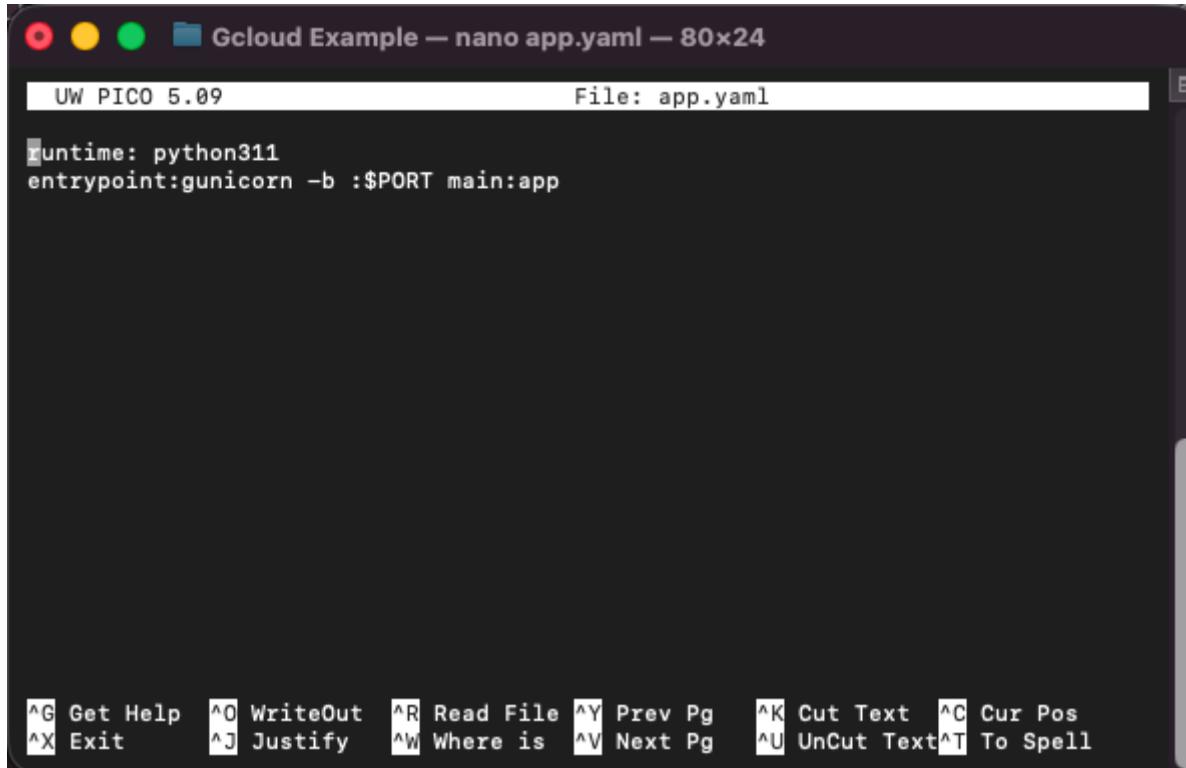
**^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text^T To Spell**

Create a file called `requirements.txt`:

```
Flask==3.0.0
gunicorn
```

This tells Google Cloud what Python packages to install. We need Flask and gunicorn. Gunicorn is a server that runs the app in production.

Step 16: App Engine Config



The screenshot shows a terminal window titled "Gcloud Example — nano app.yaml — 80x24". The window contains the following YAML code:

```
runtime: python311
entrypoint: gunicorn -b :$PORT main:app
```

The terminal interface includes a status bar at the top with "UW PICO 5.09" and "File: app.yaml". At the bottom, there is a menu bar with various keyboard shortcuts.

Create the last file called `app.yaml`:

```
runtime: python311
entrypoint: gunicorn -b :$PORT main:app
```

This tells App Engine:

- Use Python 3.11
- Start the app using gunicorn

The `$PORT` variable is important. App Engine uses this to tell your app which port to listen on.

Step 17: Running gcloud init

```
* Run `gcloud cheat-sheet` to see a roster of go-to `gcloud` commands.  
(base) muditatrey@Ged ~ % ./google-cloud-sdk/bin/gcloud init
```

Back to the terminal. Run:

```
./google-cloud-sdk/bin/gcloud init
```

This sets up the configuration. Once gcloud is in your PATH, you can just type `gcloud init` without the full path.

Step 18: Logging In

```
(base) muditatrey@Ged ~ % ./google-cloud-sdk/bin/gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=lP6H2JgU3aac8TEBYsGjzE07Db8ZVr&access_type=offline&code_challenge=oDi_wt-UpphYoWTi6RbVNna6Ph0s_s1PDJiSVQMc6KM&code_challenge_method=S256

You are now logged in as [muditatrey007@gmail.com].
Your current project is [peaceful-garden-477817-u9]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
```

Run:

```
./google-cloud-sdk/bin/gcloud auth login
```

This opens your browser to login with your Google account. After logging in, you'll see confirmation in the terminal.

Step 19: Setting the Project

```
(base) muditatrey@Ged ~ % ./google-cloud-sdk/bin/gcloud config set project peaceful-garden-477817-u9
```

To make sure everything points to the right project, run:

```
./google-cloud-sdk/bin/gcloud config set project peaceful-garden-477817-u9
```

Replace with your actual project ID.

Step 20: Project Set

```
$ gcloud config set project PROJECT_ID
(base) muditatrey@Ged ~ % ./google-cloud-sdk/bin/gcloud config set project peaceful-garden-477817-u9
INFORMATION: Project 'peaceful-garden-477817-u9' has no 'environment' tag set. Use either 'Production', 'Development', 'Test', or 'Staging'. Add an 'environment' tag using `gcloud resource-manager tags bindings create`.
Updated property [core/project].
(base) muditatrey@Ged ~ % |
```

You'll see "Updated property [core/project]" which means it worked.

The message about environment tags is optional and can be ignored.

Step 21: Go to Your Project Folder

```
updated property [core/project].  
(base) muditatrey@Ged ~ % cd ~/Documents/Codes/Gcloud\ Example  
(base) muditatrey@Ged Gcloud Example % ls  
app.yaml      main.py      requirements.txt
```

Navigate to where you saved your three files:

```
cd ~/Documents/Codes/Gcloud\ Example
```

Run `ls` to verify you have:

- `app.yaml`
- `main.py`
- `requirements.txt`

All three files should be there.

Step 22: Deploy Time

```
(base) muditatrey@Ged Gcloud Example % ~/google-cloud-sdk/bin/gcloud app deploy
WARNING: You might be using automatic scaling for a standard environment deployment, without providing a value for automatic_scaling.max_instances. Starting from March, 2025, App Engine sets the automatic scaling maximum instances default for standard environment deployments to 20. This change doesn't impact existing apps. To override the default, specify the new max_instances value in your app.yaml file, and deploy a new version or redeploy over an existing version. For details on max_instances, see https://cloud.google.com/appengine/docs/standard/reference/app-yaml.md#scaling_elements.

Services to deploy:

descriptor: [/Users/muditatrey/Documents/Codes/Gcloud Example/app.yaml]
source: [/Users/muditatrey/Documents/Codes/Gcloud Example]
target project: [peaceful-garden-477817-u9]
target service: [default]
target version: [20251113t212119]
target url: [https://peaceful-garden-477817-u9.el.r.appspot.com]
target service account: [peaceful-garden-477817-u9@appspot.gserviceaccount.com]

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

Beginning deployment of service [default] ...
[ Uploading 0 files to Google Cloud Storage ] ...

File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://peaceful-garden-477817-u9.el.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
(base) muditatrey@Ged Gcloud Example % |
```

Run:

```
~/google-cloud-sdk/bin/gcloud app deploy
```

It shows a summary of what it's about to do and asks "Do you want to continue (Y/n)?"

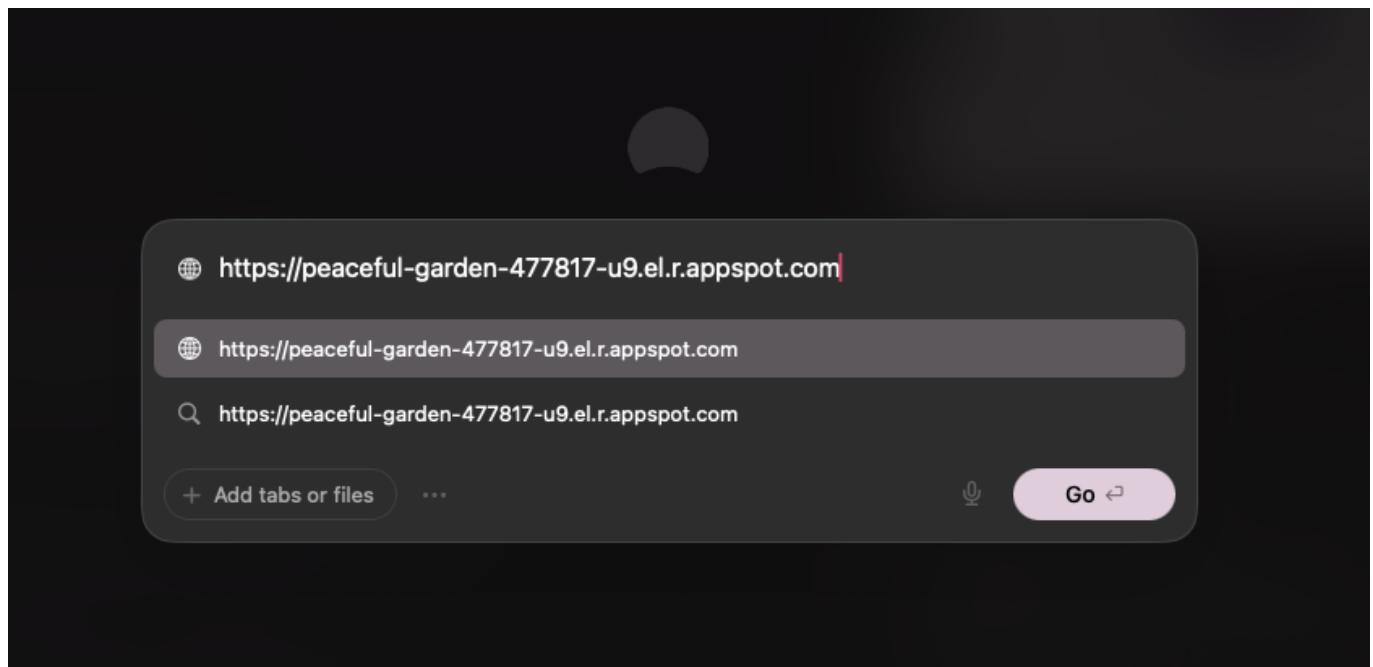
Type **y** and press enter.

It will upload your files, install the dependencies, and deploy everything. This takes a minute or two. You'll see:

- "Uploading 3 files..."
- "File upload done"
- "Updating service..."
- "Deployed service [default]"

At the end it gives you a URL. That's your app.

Step 23: Opening Your App

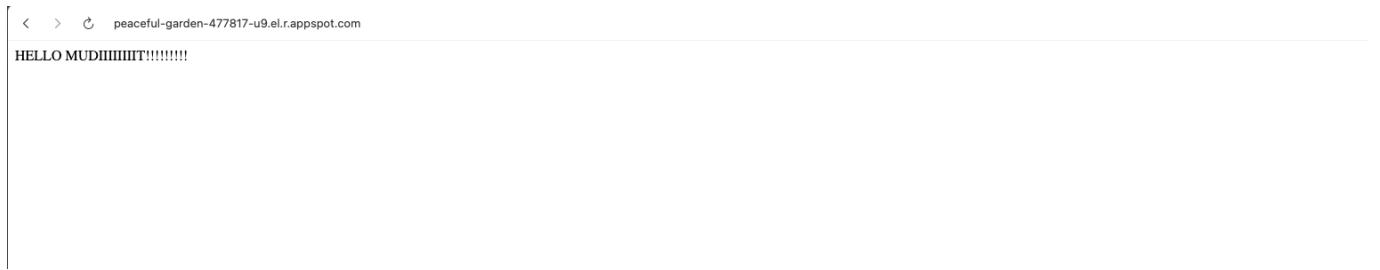


Copy the URL and paste it in your browser. It will look something like:

<https://peaceful-garden-477817-u9.el.r.appspot.com>

The page is loading.

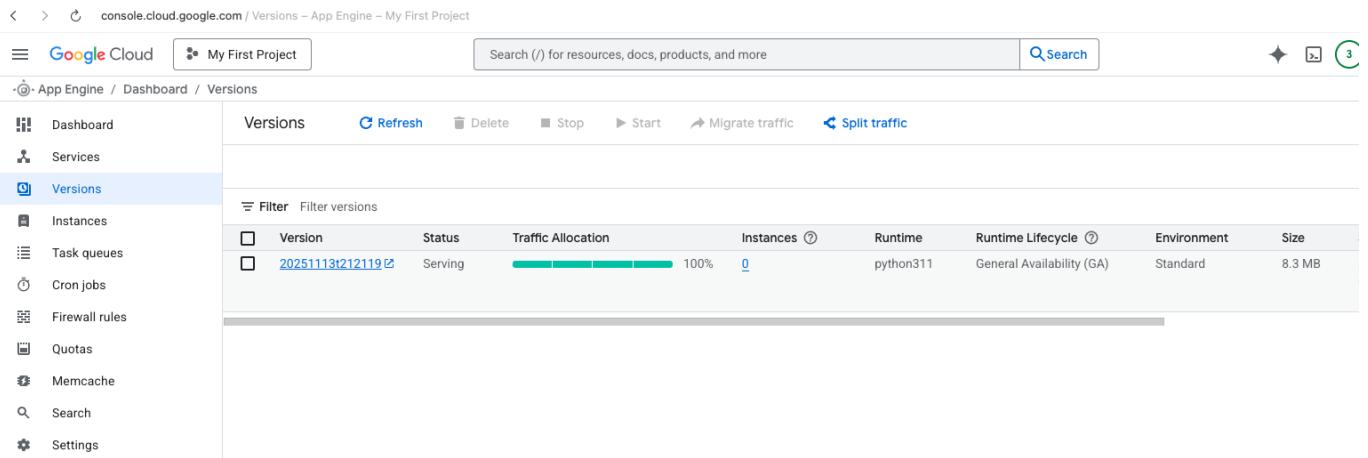
Step 24: It Works



Your message appears on the internet.

Your app is now live. Anyone in the world can visit that URL and see it.

Step 25: Checking the Dashboard



The screenshot shows the Google Cloud Console interface for an App Engine project named "My First Project". The left sidebar has a "Versions" section selected. The main area is titled "Versions" and contains a table with one row. The table columns are: Version, Status, Traffic Allocation, Instances, Runtime, Runtime Lifecycle, Environment, and Size. The single row shows a version named "20251113t212119" which is "Serving" with 100% traffic allocation, 0 instances, runtime "python311", lifecycle "General Availability (GA)", environment "Standard", and size "8.3 MB".

Version	Status	Traffic Allocation	Instances	Runtime	Runtime Lifecycle	Environment	Size
20251113t212119	Serving	100%	0	python311	General Availability (GA)	Standard	8.3 MB

Back in the Cloud Console, go to the Versions page. You can see:

- Your version (it uses a timestamp as the name)
- Status: **Serving** (green means it's running)
- Traffic: 100% (all visitors go to this version)
- Runtime: python311
- Size: 8.3MB

If you deploy again later, you'll have multiple versions here. You can switch between them.

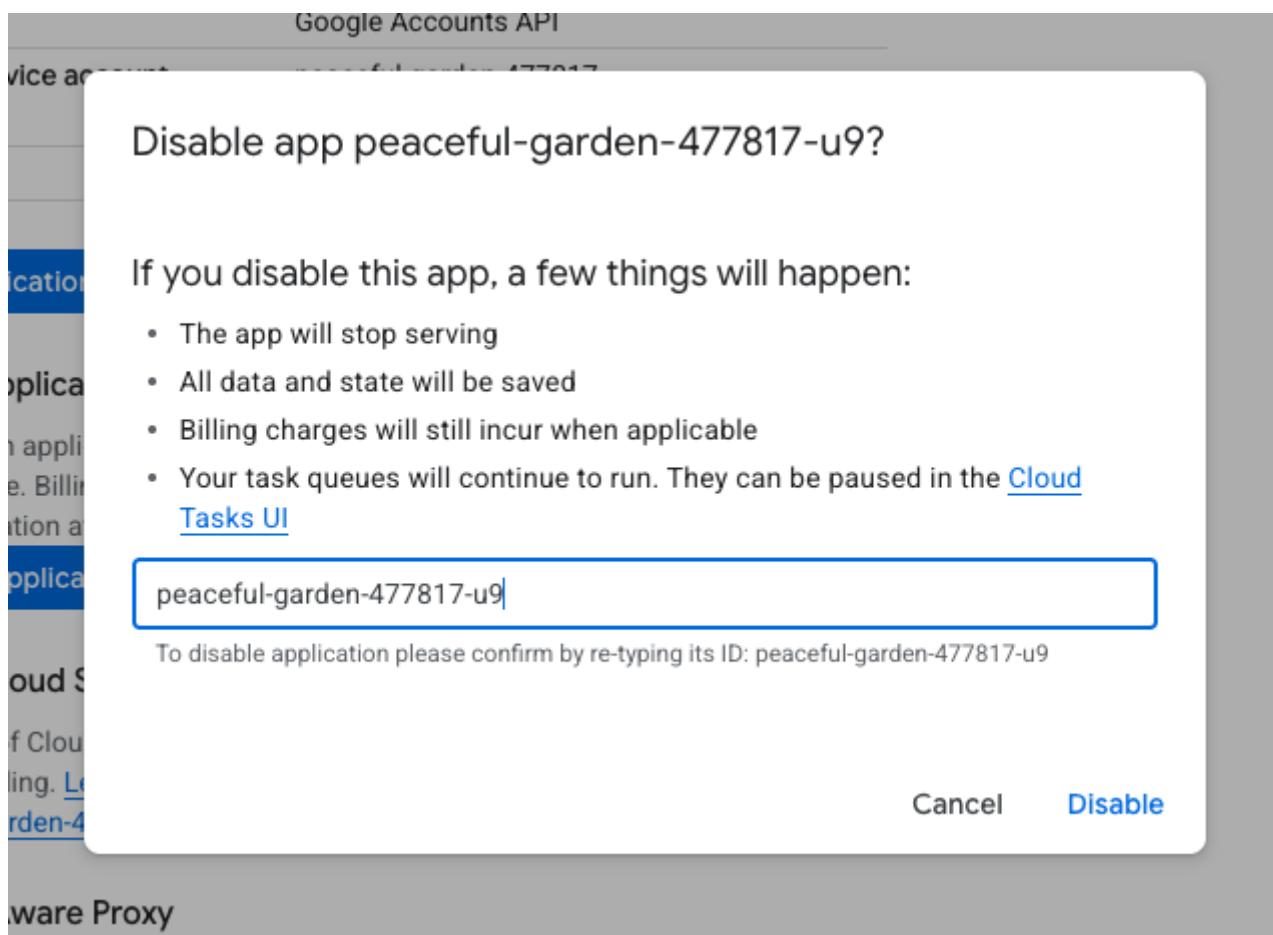
Step 26: Looking at Settings

The screenshot shows the Google Cloud Platform interface for managing an App Engine application named "My First Project". The left sidebar lists various management options like Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Firewall rules, Quotas, Memcache, and Search, with "Settings" selected. The main content area is titled "Settings" and contains tabs for "Application settings", "Custom domains", "SSL certificates", and "Email senders". Under "Application settings", there are sections for "Google login cookie expiration" (set to "Default (1 day)"), "Referrers" (set to "Google Accounts API"), "Default service account" (set to "peaceful-garden-477817-u9@appspot.gserviceaccount.com"), and "SSL Policy" (set to "TLS 1.0+ (Obsolete)"). A blue button labeled "Edit Application Settings" is visible. Below this, there's a section titled "Disable application" with a note about stopping serving requests without losing data or state, and a blue "Disable application" button. Further down are sections for "Default Cloud Storage Bucket" (with a note about enabling billing and a link to "peaceful-garden-477817-u9.appspot.com") and "Identity-Aware Proxy" (with a note about managing access and a blue "Configure Now" button).

The Settings page shows more information about your app. Most of these settings don't need to be changed.

One important thing: the "Disable application" button. This is how you turn off your app. Note that even when disabled, you might still incur some charges.

Step 27: Disabling Your App



If you click the Disable button, you must type the project ID to confirm. This prevents accidental shutdowns.

When you disable the app:

- It stops serving requests (nobody can visit it)
- Your data stays saved
- You can turn it back on anytime
- You might still get some charges

Step 28: App is Disabled

The screenshot shows the Google Cloud App Engine Versions dashboard. On the left, there's a sidebar with links like Dashboard, Services, Versions (which is selected and highlighted in blue), Instances, Task queues, Cron jobs, Firewall rules, Quotas, Memcache, Search, and Settings. The main area has a header with Refresh, Delete, Stop, Start, Migrate traffic, Split traffic, and Learn buttons. Below that, a search bar says 'Search (/) for resources, docs, products, and more' with a 'Search' button. A yellow warning box says '⚠ Your app is currently disabled' with a 'Enable in settings' button. Underneath, there's a table with columns: Version, Status, Traffic Allocation, Instances, Runtime, Runtime Lifecycle, and Envir. One row is shown: '20251113t212119' with 'Stopped' status, 100% traffic allocation, 1 instance, runtime 'python311', lifecycle 'General Availability (GA)', and environment 'Standard'. There are also 'Filter' and 'Instances' buttons.

The dashboard now shows "Your app is currently disabled" and the status is **Stopped** (grey).

You can enable it again from the settings page.

Summary

You've deployed a real app to Google Cloud. It's a simple app, but the process is the same for bigger, more complex applications.

What You Learned:

- How to set up App Engine
- Installing Google Cloud SDK
- Creating a Flask app
- Deploying with gcloud
- Managing your app in the dashboard

The Three Important Files:

1. **main.py** - Your app code
2. **requirements.txt** - What packages to install
3. **app.yaml** - Configuration for App Engine

Main Commands to Remember:

```
gcloud init          # Setup
gcloud auth login    # Login to Google
gcloud app deploy     # Deploy your app
gcloud app browse      # Open your app in a browser
gcloud app logs tail    # See live logs
```

Note: When you're done and don't need the app anymore, disable it (or delete the project) so you don't get charged for resources you're not using.