

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

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


Google Cloud My First Project

App Engine / Dashboard / Start

Dashboard

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

App Engine [Learn](#)



Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

Create Application

Cloud Run offers the most modern fully managed application hosting experience with lower minimum billable times and support for GPUs on demand for your AI/ML workloads. Deploy code written in any programming language supported by App Engine on Cloud Run.

Try Cloud Run

Release Notes

<|

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

🔍

✦

📄

2

?

⋮

👤

🔗 App Engine / Dashboard / Start / Create

☰ Dashboard

👤 Services

📅 Versions

📄 Instances

☰ Task queues

🕒 Cron jobs

🔗 Firewall rules

📄 Quotas

🔗 Memcache

🔍 Search

⚙️ Settings

Release Notes

<1


Create app

📖 Learn

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

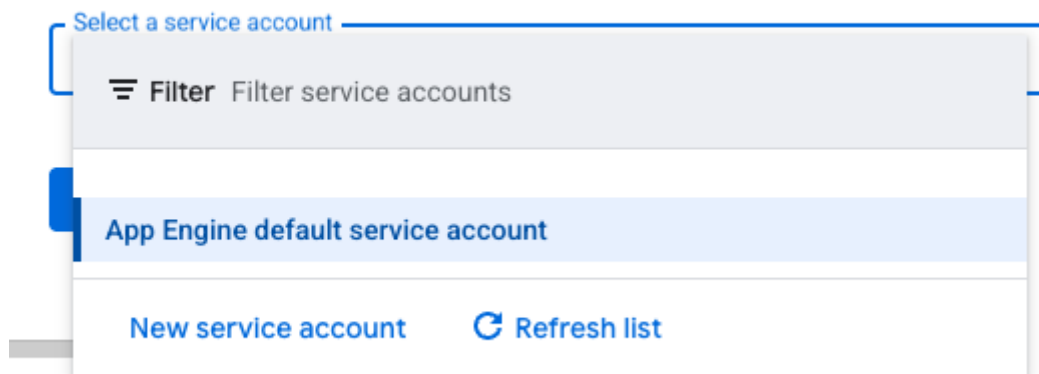
Now we're creating the app.

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

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

Google Cloud My First Project

App Engine / Dashboard / Start / Create

Dashboard

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

Create app [Learn](#)

Configure application —

2 Get started

Creating application...

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

[Try Cloud Run](#)

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 console interface. At the top, the breadcrumb navigation reads 'console.cloud.google.com / Get started – App Engine – My First Project'. The left sidebar contains a menu with 'Dashboard' selected, and other options like 'Services', 'Versions', 'Instances', 'Task queues', 'Cron jobs', 'Firewall rules', 'Quotas', 'Memcache', 'Search', and 'Settings'. The main content area is titled 'Get started' and features a green success message: 'Your App Engine application has been created. Choose the deployment method that works best for you.' Below this, the 'Resources' section shows 'Language' set to 'Python' and 'Environment' set to 'Standard'. It includes links to 'Read App Engine Python Standard Environment Documentation' and 'Visit Github for Python Standard Environment code samples'. The 'Deploy with Google Cloud SDK' section has a 'Download the Cloud SDK' button, followed by instructions to 'Initialize your SDK' with the command '\$ gcloud init' and 'Deploy to App Engine' with the command '\$ gcloud app deploy'. At the bottom, there's a section for 'Build web applications in any language, any framework supported by App Engine on Cloud Run' with 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

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

Get started [Learn](#)

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

Resources

Language

Environment

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

[<](#) [>](#) [cloud.google.com / Cloud SDK - Libraries and Command Line Tools | Google Cloud](#)

Contact Us

JUMP TO

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](#)

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 makes it easier for you to interact with Google Cloud APIs in

VIDEO

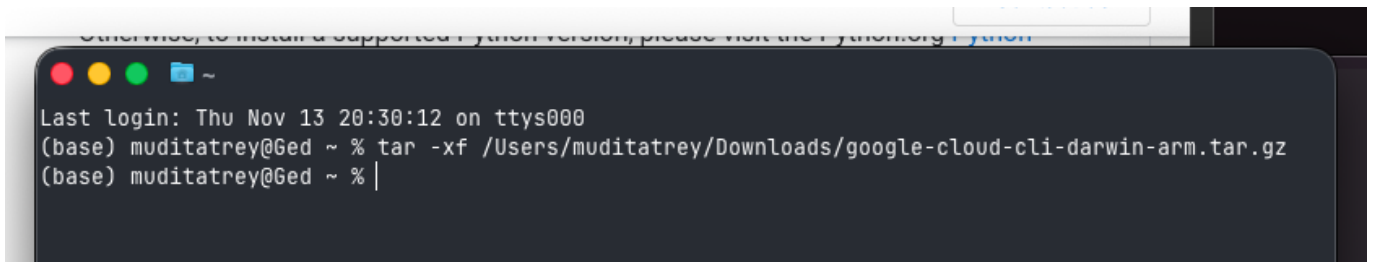
What is the Google Cloud SDK?

3:00

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 window has a dark background and a title bar with red, yellow, and green window control buttons. The text inside the terminal shows the last login time and the command to extract a tar.gz file. The prompt is '(base) muditatrey@Ged ~ %'.

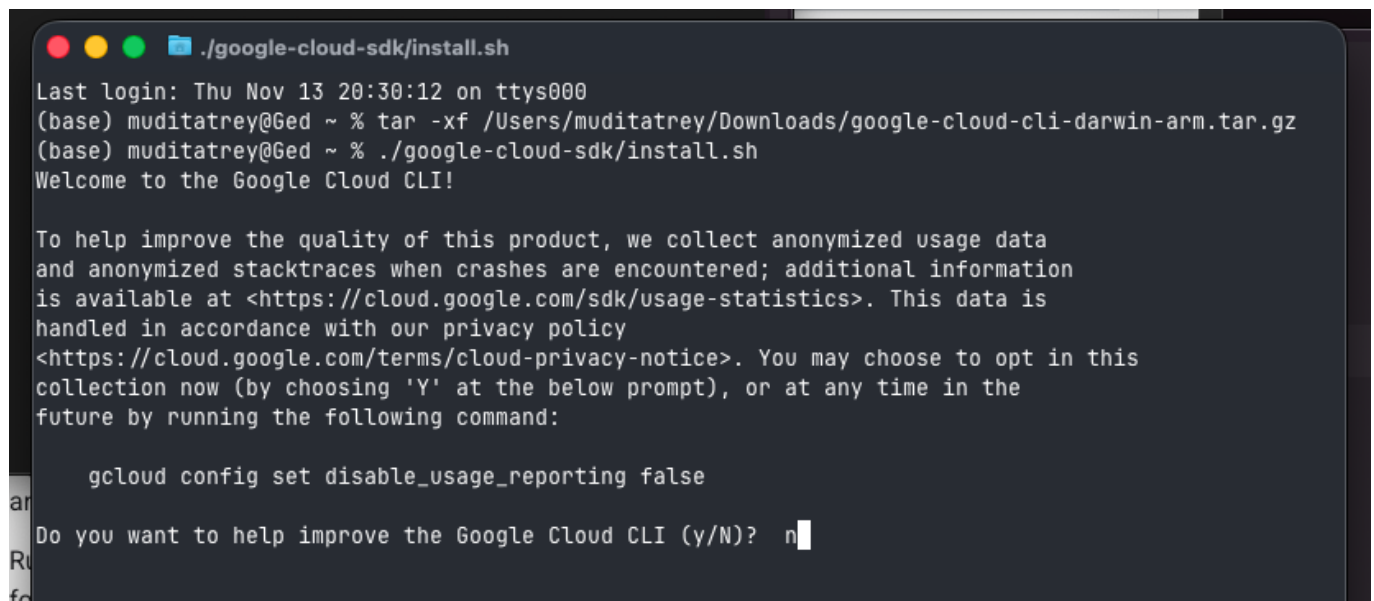
```
Last login: Thu Nov 13 20:30:12 on ttys000
(base) muditatrey@Ged ~ % tar -xf /Users/muditatrey/Downloads/google-cloud-cli-darwin-arm.tar.gz
(base) muditatrey@Ged ~ % |
```

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

A terminal window with a dark background and light text. The title bar shows three colored circles (red, yellow, green) and a blue icon, followed by the text `./google-cloud-sdk/install.sh`. The terminal output shows the user's login history, the command to extract the SDK tarball, and the execution of the install script. The script displays a welcome message and a privacy notice. It then prompts the user to opt in for usage data collection, with the user responding 'n' for no.

```
./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 | kubectrl | kubectrl |
| < 1 MiB | | |
| Not Installed | kubectrl-oidc | kubectrl-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 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.

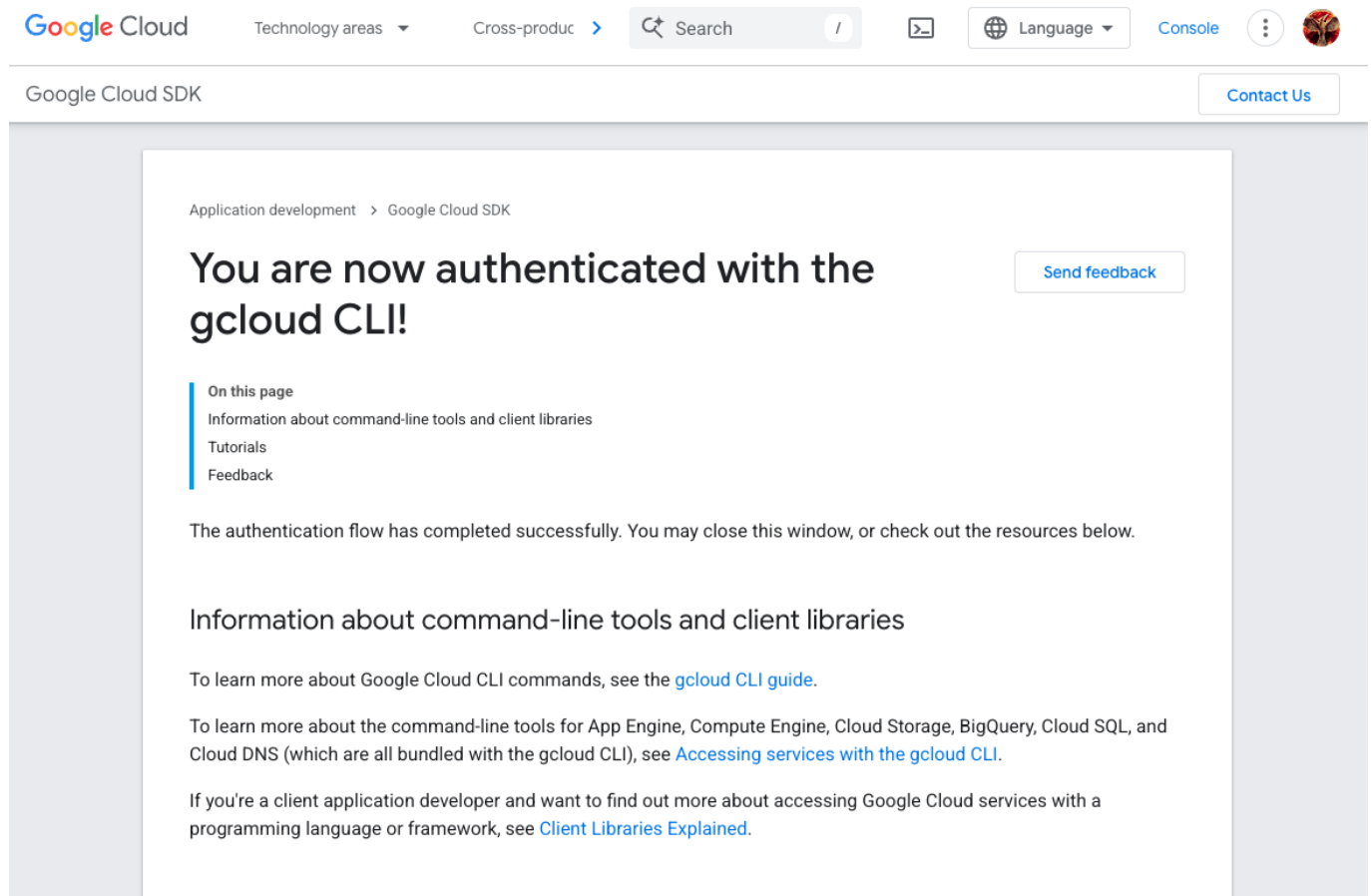
Enter a path to an rc file to update, or leave blank to use [/Users/muditatrey/.zshrc]:
```

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 the Google Cloud SDK web interface. At the top is a navigation bar with the Google Cloud logo, 'Technology areas' dropdown, 'Cross-product' dropdown, a search bar, a 'Language' dropdown, a 'Console' link, and a user profile icon. Below the navigation bar, the page title 'Google Cloud SDK' is on the left and a 'Contact Us' button is on the right. The main content area has a breadcrumb 'Application development > Google Cloud SDK'. The primary heading is 'You are now authenticated with the gcloud CLI!', with a 'Send feedback' button to its right. A 'On this page' sidebar lists 'Information about command-line tools and client libraries' (highlighted), 'Tutorials', and 'Feedback'. The text below states: 'The authentication flow has completed successfully. You may close this window, or check out the resources below.' This is followed by a section 'Information about command-line tools and client libraries' containing three paragraphs with links to the 'gcloud CLI guide', 'Accessing services with the gcloud CLI', and 'Client Libraries Explained'.

Google Cloud Technology areas Cross-product Search Language Console

Google Cloud SDK Contact Us

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](#).

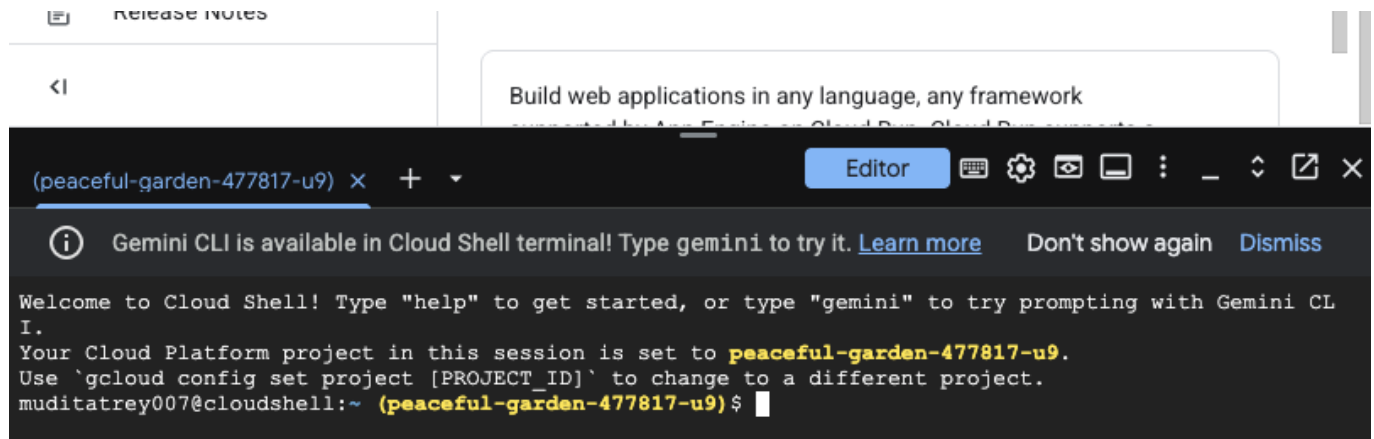
To learn more about the command-line tools for App Engine, Compute Engine, Cloud Storage, BigQuery, Cloud SQL, and Cloud DNS (which are all bundled with the gcloud CLI), see [Accessing services with the gcloud CLI](#).

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.googleu
sercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googlea
pis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservic
e.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2
Faccounts.reauth&state=MiyprENumjTZ2LTiETU4Yk0Iz8ruV&access_type=offline&code_challenge=Lj06PY7mW1
CzkIclLWrXtN5e-0eEdVetCTdokk92Sxw&code_challenge_method=S256

You are signed in as: [muditatreya007@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].

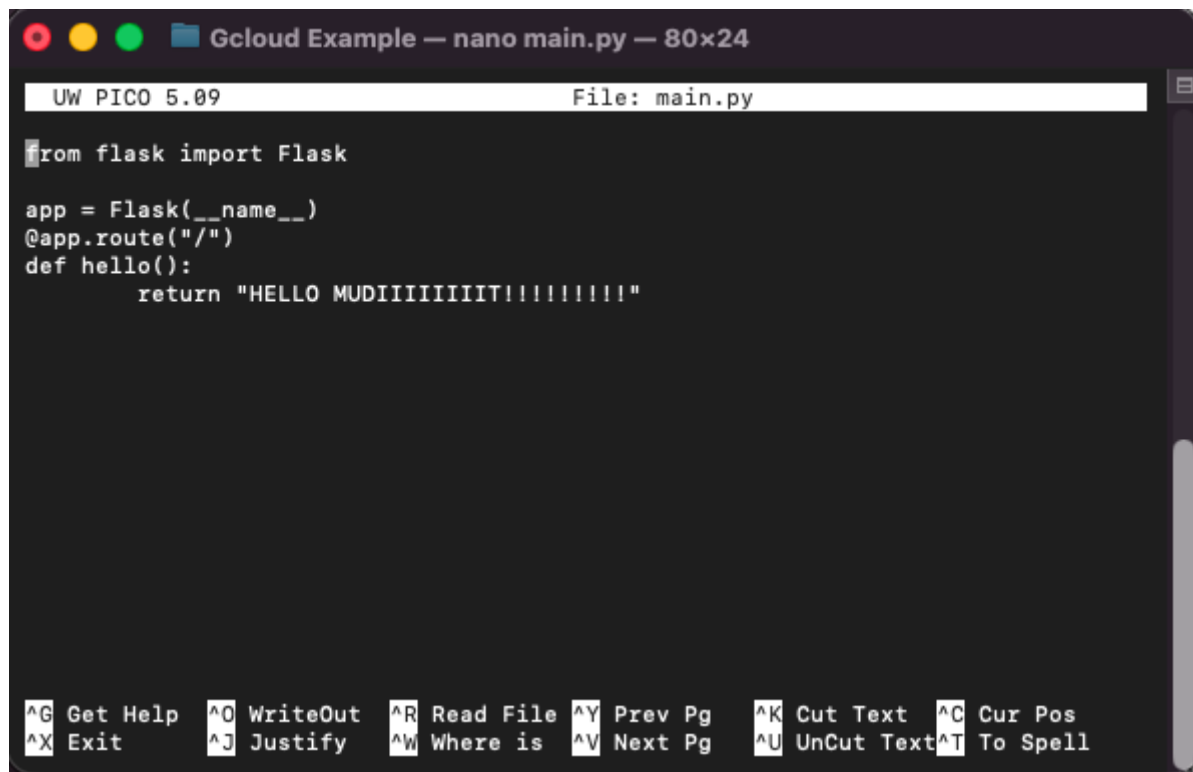
Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud 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



```
UW PICO 5.09 File: main.py

from flask import Flask

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

^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

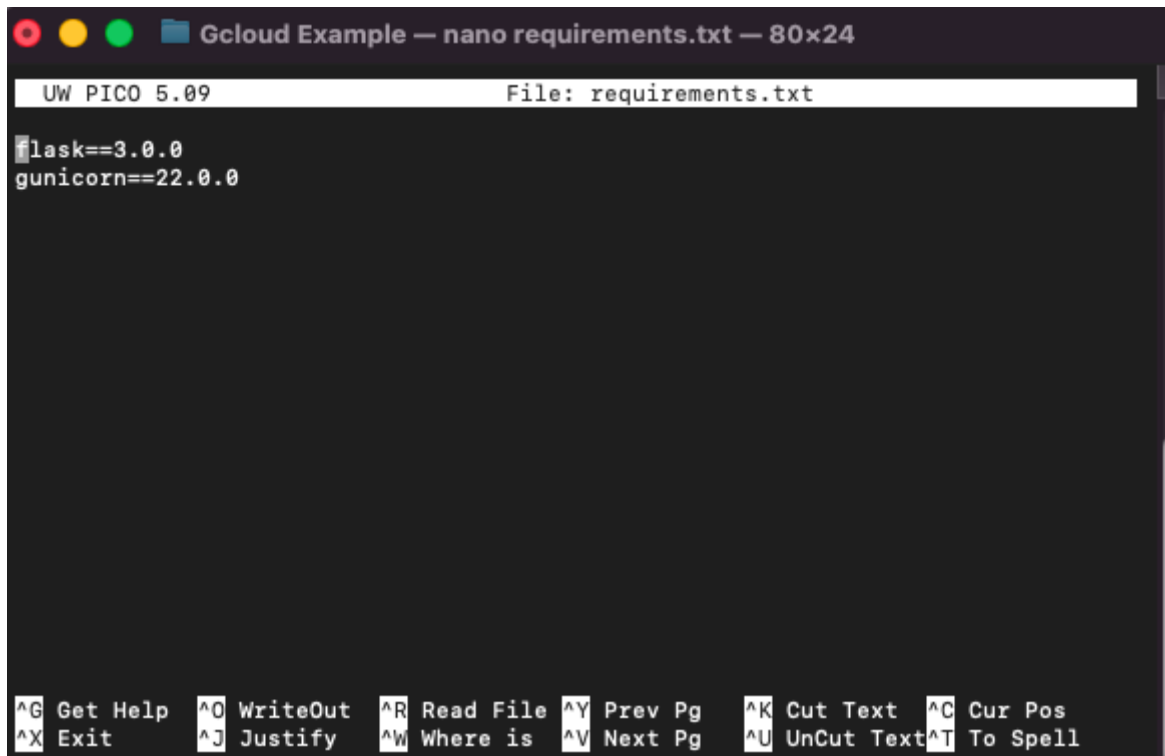
app = Flask(__name__)

@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



```
UW PICO 5.09 File: requirements.txt
Flask==3.0.0
gunicorn==22.0.0
```

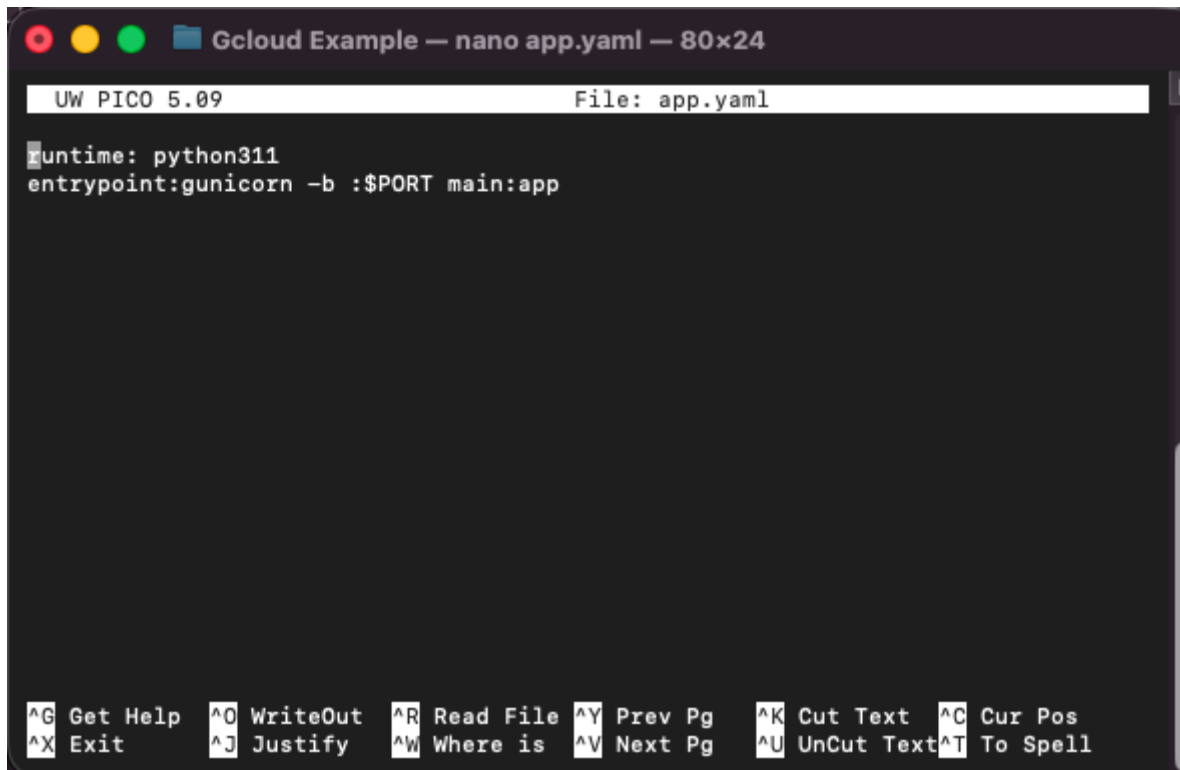
^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

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



```
UW PICO 5.09 File: app.yaml
runtime: python311
entrypoint: gunicorn -b :$PORT main:app

^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 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.googleu
sercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googlea
pis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservic
e.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2
Faccounts.reauth&state=lp6H2JgU3aac8TEBYs6jzE07Db8ZVr&access_type=offline&code_challenge=oDi_wt-Upp
hYoWTi6RbVNna6Ph0s_s1PDJiSVQM6KM&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  
(base) muditatrey@Ged Gcloud Example %
```

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]...

[Progress bar: 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)?"

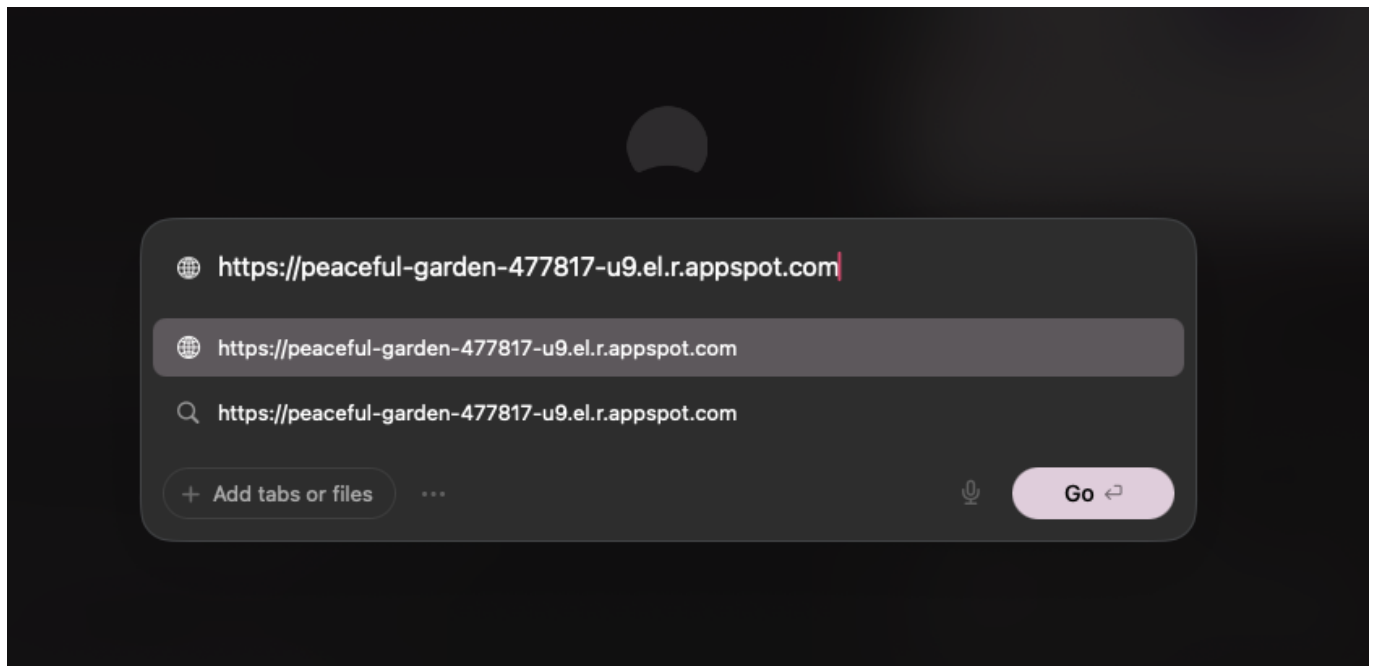
Type **y** and press enter.

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

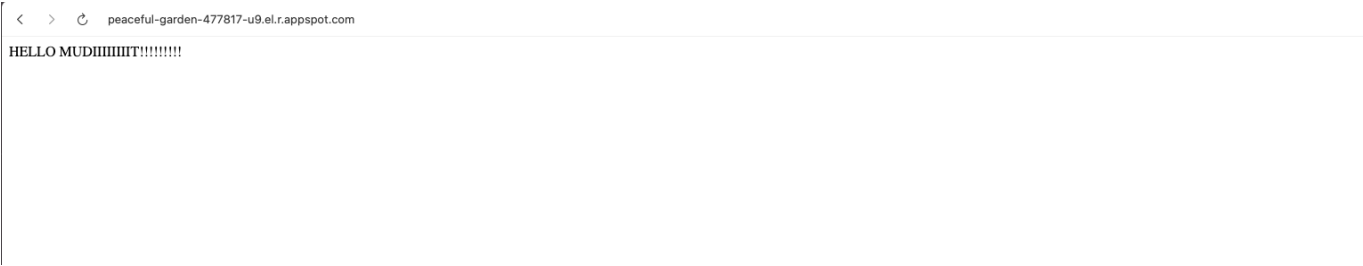


Copy the URL and paste it in your browser. It will look something like:

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

The page is loading.

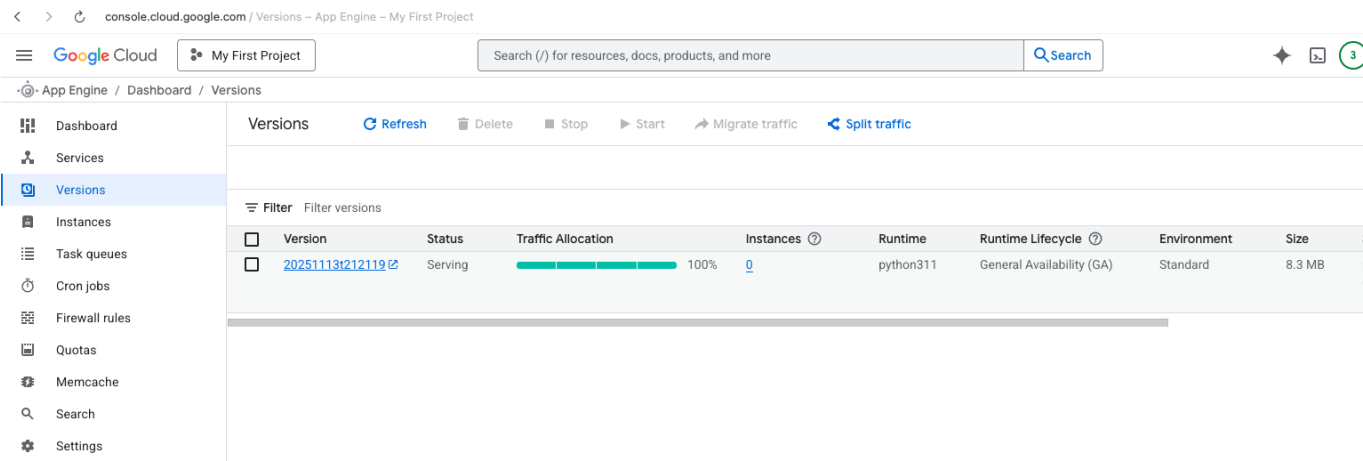
Step 24: It Works



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



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

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

Google Cloud My First Project Search (/) for resources, docs, products, and more

App Engine / Dashboard / Settings

Settings

Application settings Custom domains SSL certificates Email senders

Google login cookie expiration	Default (1 day)
Referrers	Google Accounts API
Default service account	peaceful-garden-477817-u9@appspot.gserviceaccount.com
SSL Policy	TLS 1.0+ (Obsolete)

Edit Application Settings

Disable application

Disabling an application will stop all serving requests, but you will not lose any data or state. Billing charges will still incur when applicable. You can re-enable your application at any time.

Disable application

Default Cloud Storage Bucket

Up to 5GB of Cloud Storage may be used with App Engine applications without enabling billing. [Learn more](#).
peaceful-garden-477817-u9.appspot.com

Identity-Aware Proxy

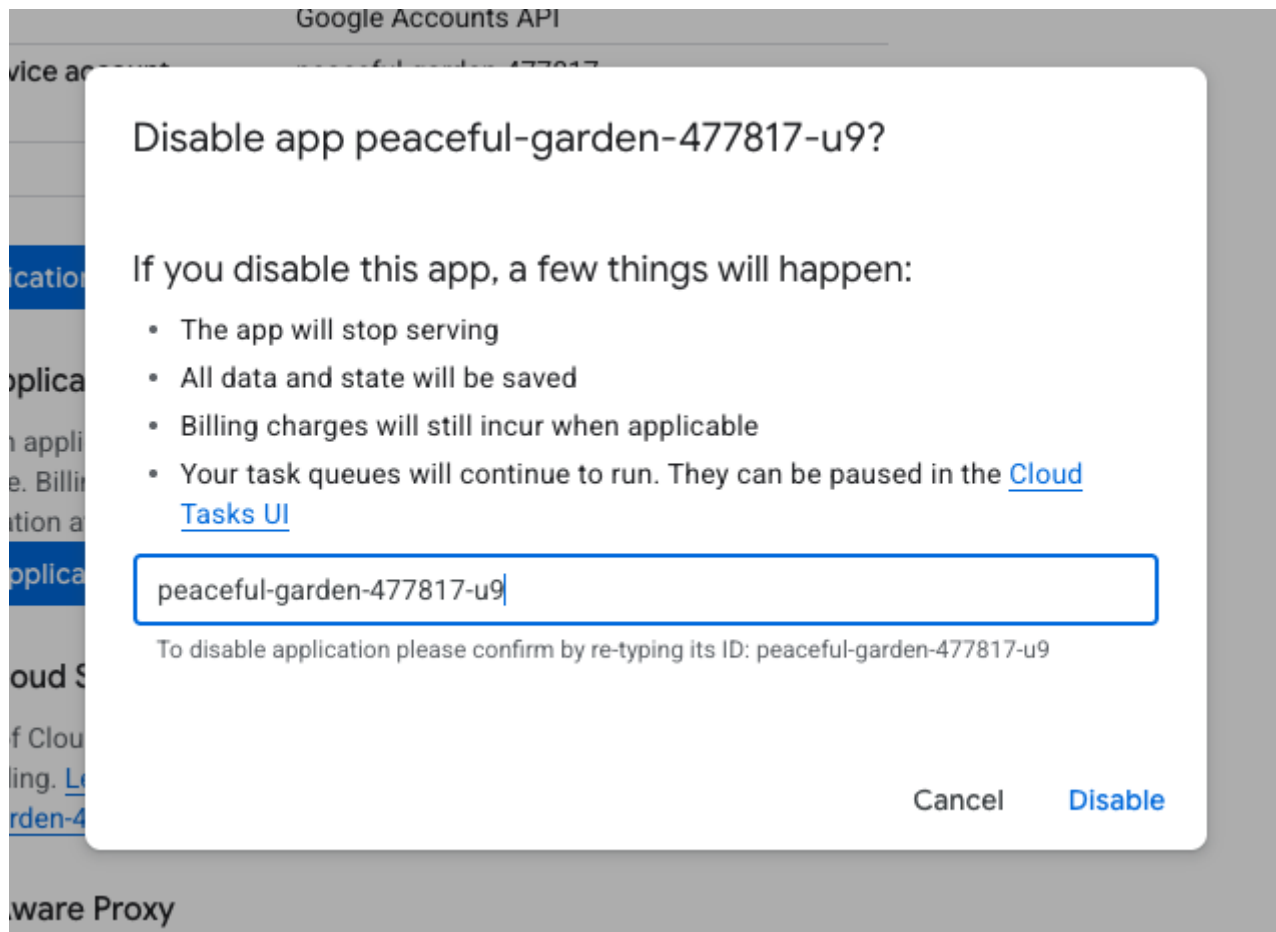
Manage access to services hosted on App Engine.
Currently access is DISABLED.

Configure Now

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

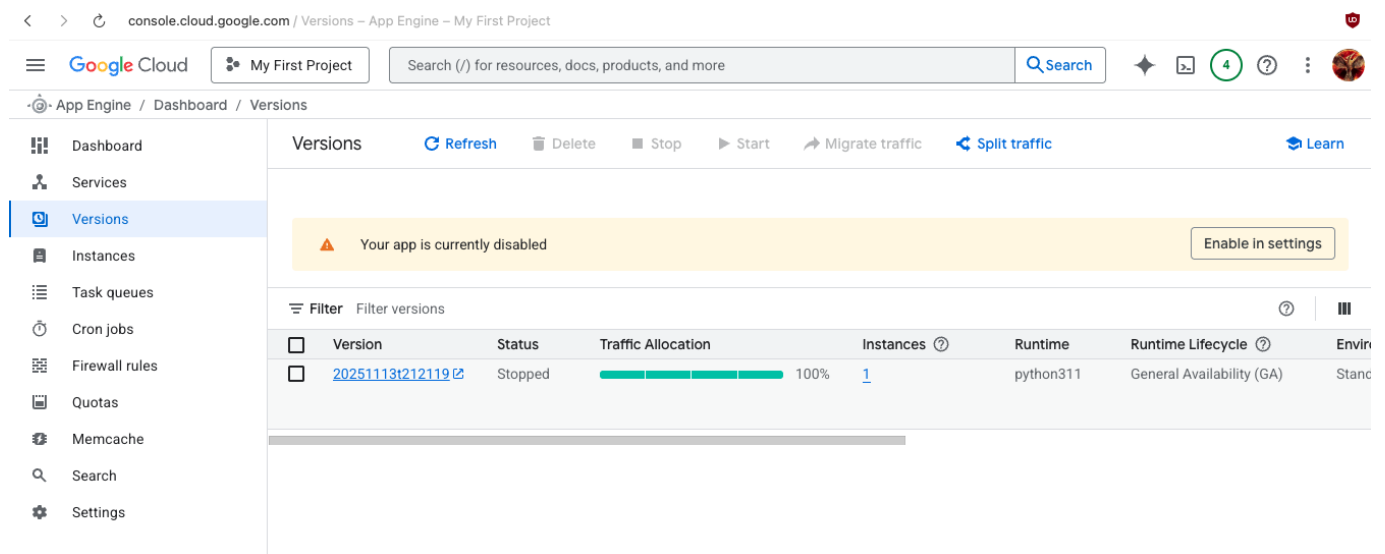


If you click the Disable button, you must type the project ID to confirm. This prevents accidental shutdowns.

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 console interface for the 'My First Project' App Engine dashboard. The left sidebar contains navigation links for Dashboard, Services, Versions (selected), Instances, Task queues, Cron jobs, Firewall rules, Quotas, Memcache, Search, and Settings. The main content area is titled 'Versions' and includes a 'Refresh' button, 'Delete', 'Stop', 'Start', 'Migrate traffic', and 'Split traffic' actions. A yellow banner at the top states 'Your app is currently disabled' with an 'Enable in settings' button. Below this is a table of versions. The table has columns for Version, Status, Traffic Allocation, Instances, Runtime, Runtime Lifecycle, and Environment. One version is listed with ID '202511131212119', status 'Stopped', 100% traffic allocation, 1 instance, python311 runtime, and General Availability (GA) lifecycle.

Version	Status	Traffic Allocation	Instances	Runtime	Runtime Lifecycle	Environment
202511131212119	Stopped	100%	1	python311	General Availability (GA)	Stanc

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.