

Quickstart for Python 3 in the App Engine Standard Environment

Python [2.7/3](#) | Java [8/11](#) (/appengine/docs/standard/java11/quickstart) | PHP [5/7](#) (/appengine/docs/standard/php7/quickstart) | [Ruby](#) (/appengine/docs/standard/ruby/quickstart) | Go [1.11](#) (/appengine/docs/standard/go111/quickstart) / [1.12+](#) (/appengine/docs/standard/go/quickstart) | [Node.js](#) (/appengine/docs/standard/nodejs/quickstart)

This quickstart shows how to deploy a sample app on App Engine. If you want to learn about coding an app for App Engine, see [Building an app](#) (/appengine/docs/standard/python3/building-app).

Costs

There are no costs associated with running this guide. Running this sample app alone does not exceed your [free quota](#) (/appengine/quotas).

Before you begin

Before running and deploying this quickstart, install the Cloud SDK and then set up a Cloud project for App Engine:

1. Download and install Cloud SDK:

[Download the SDK](#) (/sdk/docs)

Note: If you already have the Cloud SDK installed, update it by running the following command:

```
gcloud components update
```

2. Create a new project:

```
gcloud projects create [YOUR_PROJECT_ID] --set-as-default
```

Replace `[YOUR_PROJECT_ID]` with a string of characters

(`/sdk/gcloud/reference/projects/create#PROJECT_ID`) that uniquely identifies your project. For example, `my-project-24`.

Verify the project was created:

```
gcloud projects describe [YOUR_PROJECT_ID]
```


You see project details that might look like the following:

```
createTime: year-month-hour
lifecycleState: ACTIVE
name: project-name
parent:
id: '433637338589'
type: organization
projectId: project-name-id
projectNumber: 499227785679
```

3. Initialize your App Engine app with your project and choose its region:

```
gcloud app create --project=[YOUR_PROJECT_ID]
```

When prompted, select the region (#before-you-begin) where you want your App Engine application located.

 **Caution:** You cannot change an app's region once it has been set.

4. Make sure billing is enabled for your project. A billing account needs to be linked to your project in order for the application to be deployed to App Engine.

Enable billing (<https://console.cloud.google.com/projectselector/billing?lang=python3&st=true>)

★ **Note:** Running this sample app does not exceed your [free quota](/appengine/quotas). (</appengine/quotas>)

5. Install the following prerequisites:

- [Download and install Git](https://git-scm.com/) (<https://git-scm.com/>).
- Run the following command to install the [gcloud component](/sdk/docs/managing-components) (</sdk/docs/managing-components>) that includes the App Engine extension for Python 3:

```
gcloud components install app-engine-python
```

- ## 6. Prepare your environment for Python development.
- It is recommended that you have the latest version of Python, `pip`, and other related tools installed on your system. For instructions, refer to the [Python Development Environment Setup Guide](/python/setup) (</python/setup>).

This quickstart demonstrates a simple Python app written with the [Flask](http://flask.pocoo.org/) (<http://flask.pocoo.org/>) web framework that can be deployed to App Engine. Although this sample uses Flask, you can use any web framework that satisfies the requirements above. Alternative frameworks include [Django](https://www.djangoproject.com) (<https://www.djangoproject.com>), [Pyramid](http://www.pylonsproject.org/) (<http://www.pylonsproject.org/>), [Bottle](http://bottlepy.org/) (<http://bottlepy.org/>), and [web.py](http://webpy.org/) (<http://webpy.org/>).

Download the Hello World app

We've created a simple Hello World app for Python 3 so you can quickly get a feel for deploying an app to the Google Cloud.

1. Clone the Hello World sample app repository to your local machine.

```
git clone https://github.com/GoogleCloudPlatform/python-docs-samples
```

Alternatively, you can [download the sample](https://github.com/GoogleCloudPlatform/python-docs-samples/archive/master.zip)

(<https://github.com/GoogleCloudPlatform/python-docs-samples/archive/master.zip>) as a zip file and extract it.

2. Change to the directory that contains the sample code.

```
cd python-docs-samples/appengine/standard_python3/hello_world
```

Run Hello World on your local machine

To run the Hello World app on your local computer:

Mac OS / LinuxWindows (#windows)

1. Create an isolated Python environment in a directory external to your project and activate it:

```
python3 -m venv env
source env/bin/activate
```

2. Navigate to your project directory and install dependencies:


```
cd YOUR_PROJECT
pip install -r requirements.txt
```

3. Run the application:

```
python main.py
```

4. In your web browser, enter the following address:

```
http://localhost:8080
```

★ **Note:** If you are using Cloud Shell, in the toolbar, click **Web Preview**  and select **Preview on port 8080** instead.

The **Hello World** message from the sample app displays on the page. In your terminal window, press **Ctrl+C** to exit the web server.

Deploy and run Hello World on App Engine

To deploy your app to the App Engine standard environment:

1. Deploy the Hello World app by running the following command from the `standard_python3/hello_world` directory:

```
gcloud app deploy
```

Learn about the [optional flags \(#deploy_and_run_hello_world_on_app_engine\)](#).

2. Launch your browser to view the app at `https://PROJECT_ID.REGION_ID(#appengine-urls).r.appspot.com`

```
gcloud app browse
```

where **PROJECT_ID** represents your Google Cloud project ID.

This time, the page that displays the Hello World message is delivered by a web server running on an App Engine instance.

Congratulations! You've deployed your first Python 3 app to App Engine standard environment!

See the following sections for information about cleaning up as well as links to possible next steps that you can take.

Clean up


To avoid incurring charges, you can delete your Cloud project to stop billing for all the resources used within that project.

! **Caution:** Deleting a project has the following effects:

- **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
- **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such as an `appspot.com` URL, delete selected resources inside the project instead of deleting the whole project.

1. In the Cloud Console, go to the **Manage resources** page.

[Go to the Manage resources page \(https://console.cloud.google.com/iam-admin/projects\)](https://console.cloud.google.com/iam-admin/projects)

2. In the project list, select the project that you want to delete and then click **Delete** .

3. In the dialog, type the project ID and then click **Shut down** to delete the project.

What's next

Learn the whole platform

Now that you know what it's like to develop and deploy App Engine apps, you can explore the rest of Google Cloud. You already have the Cloud SDK installed which gives you the tools to interact with products like Cloud SQL, Cloud Storage, Firestore, and more.

Learn about the App Engine standard environment

Here are some topics to help continue your learning about App Engine:

- [An overview of App Engine \(/appengine/docs/standard/python3/an-overview-of-app-engine\)](/appengine/docs/standard/python3/an-overview-of-app-engine)
- [Request routing \(/appengine/docs/standard/python3/how-requests-are-routed\)](/appengine/docs/standard/python3/how-requests-are-routed)

- [Request handling](/appengine/docs/standard/python3/how-requests-are-handled) (/appengine/docs/standard/python3/how-requests-are-handled)
- [Managing instances](/appengine/docs/standard/python3/how-instances-are-managed) (/appengine/docs/standard/python3/how-instances-are-managed)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-08-14 UTC.