Serverless Computing (https://cloud.google.com/products/serverless/)

App Engine (https://cloud.google.com/appengine/)

Documentation (https://cloud.google.com/appengine/docs/)

Flexible Environment (https://cloud.google.com/appengine/docs/flexible/)

PHP (https://cloud.google.com/appengine/docs/flexible/php/) Guides

Quickstart for PHP in the App Engine Flexible Environment

Python (https://cloud.google.com/appengine/docs/flexible/python/quickstart) | Java
(https://cloud.google.com/appengine/docs/flexible/java/quickstart) | Node.js
(https://cloud.google.com/appengine/docs/flexible/nodejs/quickstart) | Go
(https://cloud.google.com/appengine/docs/flexible/go/quickstart) | Ruby
(https://cloud.google.com/appengine/docs/flexible/ruby/quickstart) | **PHP** | .NET
(https://cloud.google.com/appengine/docs/flexible/dotnet/quickstart)

This quickstart shows you how to create a small App Engine app that displays a short
message.

Before you begin

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

1. Download and install Cloud SDK:

   DOWNLOAD THE SDK (HTTPS://CLOUD.GOOGLE.COM/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
   ```

   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=P

   Your flexible environment deployment will incur costs while it is active. Clean up your
   project (#clean-up) when you are finished to avoid ongoing costs.

5. Install the following prerequisites:

   - Download and install Git (https://git-scm.com/).

   - Install PHP (https://secure.php.net/downloads.php)

   - To manage dependencies, download Composer (https://getcomposer.org/download/)
     and make sure the Composer executable is installed globally
      (https://getcomposer.org/doc/00-intro.md#globally).

This quickstart assumes that you are familiar with the PHP programming language.

Download the Hello World app

We've created a simple Hello World app for PHP 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/php-docs-samples.git
```

Alternatively, you can download the sample
(https://github.com/GoogleCloudPlatform/php-docs-samples/archive/master.zip) as a zip file and extract it.

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

```
cd php-docs-samples/appengine/flexible/helloworld
```

Run Hello World on your local machine

To run the Hello World app on your local computer:

1. Install dependencies.

```
composer install
```

2. Start a local web server.

```
php -S localhost:8080 -t web/
```

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

http://localhost:8080 (http://localhost:8080)

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 flexible environment:

1. Deploy the Hello World app by running the following command from the `appengine/flexible/helloworld` 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 `http://YOUR_PROJECT_ID.appspot.com`

```
gcloud app browse
```

where *YOUR_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 PHP app to App Engine flexible 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 Google 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/PRO

2. In the project list, select the project you want to delete and 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.

For a guided walkthrough that teaches you how to create an app that uses the entire platform, not just App Engine, check out our quickstart on creating the Bookshelf app (https://cloud.google.com/php/getting-started/tutorial-app).

Learn about the App Engine flexible environment

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

- An overview of App Engine
  (https://cloud.google.com/appengine/docs/flexible/php/an-overview-of-app-engine)

- How requests are routed
  (https://cloud.google.com/appengine/docs/flexible/php/how-requests-are-routed)

- How requests are handled
  (https://cloud.google.com/appengine/docs/flexible/php/how-requests-are-handled)

- How instances are managed
  (https://cloud.google.com/appengine/docs/flexible/php/how-instances-are-managed)

Hello World code review

Hello World is the simplest possible App Engine app, as it contains only one service, has only one version, and all of the code is located within the app's root directory. This section describes each of the app files in detail.

`index.php`

The `index.php` file contains the PHP code to start a server and respond to requests. This example uses Silex (http://silex.sensiolabs.org/), but you can use other PHP web frameworks (https://github.com/GoogleCloudPlatform/php-docs-samples).

This code creates a new Silex `Application` object and then defines two `GET` routes:

appengine/flexible/helloworld/web/index.php
 (https://github.com/GoogleCloudPlatform/php-docs-samples/blob/master/appengine/flexible/helloworld/web/index.php)

.ATFORM/PHP-DOCS-SAMPLES/BLOB/MASTER/APPENGINE/FLEXIBLE/HELLOWORLD/WEB/INDEX.PHP)

```php
require_once __DIR__ . '/../vendor/autoload.php';

$app = new Silex\Application();

$app->get('/', function () {
    return 'Hello World';
});

$app->get('/goodbye', function () {
    return 'Goodbye World';
});

// @codeCoverageIgnoreStart
if (PHP_SAPI != 'cli') {
    $app->run();
}
// @codeCoverageIgnoreEnd

return $app;
```

`app.yaml`

The `app.yaml`
 (https://cloud.google.com/appengine/docs/flexible/php/configuring-your-app-with-app-yaml) file describes an app's deployment configuration:

[appengine/flexible/helloworld/app.yaml](https://github.com/GoogleCloudPlatform/php-docs-samples/blob/master/appengine/flexible/helloworld/app.yaml)
 (https://github.com/GoogleCloudPlatform/php-docs-samples/blob/master/appengine/flexible/helloworld/app.yaml)

OUDPLATFORM/PHP-DOCS-SAMPLES/BLOB/MASTER/APPENGINE/FLEXIBLE/HELLOWORLD/APP.YAML)

```
runtime: php
env: flex

runtime_config:
  document_root: web

# This sample incurs costs to run on the App Engine flexible environment.
# The settings below are to reduce costs during testing and are not appropriate
# for production use. For more information, see:
# https://cloud.google.com/appengine/docs/flexible/php/configuring-your-app-with-app
manual_scaling:
  instances: 1
resources:
  cpu: 1
  memory_gb: 0.5
  disk_size_gb: 10
```

Here, `app.yaml` specifies the runtime used by the app, and sets `env: flex`, specifying that the app uses the flexible environment (https://cloud.google.com/appengine/docs/flexible/).

- For more information on how the PHP runtime works, see The PHP runtime
  (https://cloud.google.com/appengine/docs/flexible/php/runtime).

- For more details about how to design your app to take advantage of versions and services, see An overview of App Engine
  (https://cloud.google.com/appengine/docs/flexible/php/an-overview-of-app-engine).

- For more details about the configuration settings for App Engine, see Configuring your app with app.yaml
  (https://cloud.google.com/appengine/docs/flexible/php/configuring-your-app-with-app-yaml).

`composer.json`

`composer.json` (https://getcomposer.org/doc/01-basic-usage.md#composer-json-project-setup) is used to declare, install, and update application dependencies:

appengine/flexible/helloworld/composer.json
(https://github.com/GoogleCloudPlatform/php-docs-
samples/blob/master/appengine/flexible/helloworld/composer.json)

ATFORM/PHP-DOCS-SAMPLES/BLOB/MASTER/APPENGINE/FLEXIBLE/HELLOWORLD/COMPOSER.JSON)

```json
{
  "require": {
    "php": "5.6.*|7.0.*|7.1.*|7.2.*",
    "silex/silex": "^2.3"
  },
  "require-dev": {
    "symfony/browser-kit": "^4.4",
    "symfony/http-kernel": "^4.4"
  }
}
```