

PHP (<https://cloud.google.com/php/>) [Guides](#)

# Hosting WordPress in the App Engine flexible environment

This tutorial shows how to deploy a small WordPress site to the App Engine flexible environment.

## Objectives

- Create a Cloud SQL Second Generation instance.
- Configure a sample WordPress site.
- Deploy the sample WordPress site to the App Engine flexible environment.

## Costs

**Important:** This tutorial uses the following billable components of Google Cloud:

- [Compute Engine](https://cloud.google.com/compute/pricing) (<https://cloud.google.com/compute/pricing>)
- [Cloud SQL](https://cloud.google.com/sql/pricing) (<https://cloud.google.com/sql/pricing>)

To generate a cost estimate based on your projected usage, use the [pricing calculator](https://cloud.google.com/products/calculator) (<https://cloud.google.com/products/calculator>). New Google Cloud users might be eligible for a [free trial](https://cloud.google.com/free-trial) (<https://cloud.google.com/free-trial>).

## Before you begin

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Google Cloud project.

**Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

**[GO TO THE PROJECT SELECTOR PAGE](https://console.cloud.google.com/projectselector)** (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECTOR)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](https://cloud.google.com/billing/docs/how-to/modify-project) (https://cloud.google.com/billing/docs/how-to/modify-project).

4. Enable the required APIs.

**[ENABLE THE APIS](https://console.cloud.google.com/flows/enableapi?APIID=SQLADMIN)** (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=SQLADMIN)

5. [Install and initialize the Cloud SDK](https://cloud.google.com/sdk/docs/) (https://cloud.google.com/sdk/docs/).

6. Create credentials by following these steps:

- In the Cloud Console, go to the **Credentials** page.

**[OPEN THE API CREDENTIALS PAGE](https://console.cloud.google.com/apis/credentials)** (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APIS/CREDENTIALS)

- Click **Create credentials** and select **Service account key**.
- Select **Service account > App Engine default service account**.
- Click **Create**.
- Save the downloaded key in a secure place.

7. Install [PHP](http://php.net/manual/en/install.php) (http://php.net/manual/en/install.php) and [Composer](https://getcomposer.org) (https://getcomposer.org).

8. [Download the Cloud SQL Proxy](https://cloud.google.com/sql/docs/mysql-connect-proxy#install)

(https://cloud.google.com/sql/docs/mysql-connect-proxy#install) and make it executable. Also add the location of the Cloud SQL Proxy executable file to your `PATH` environment variable.

9. Install a [MySQL client](https://www.mysql.com) (https://www.mysql.com) and verify that the location of the `mysql` executable file is in your `PATH` environment variable.

## Creating and configuring a Cloud SQL Second Generation instance

## 1. Create a Cloud SQL Second Generation instance:

```
gcloud sql instances create tutorial-sql-instance \  
  --activation-policy=ALWAYS \  
  --tier=db-n1-standard-1 \  
  --region=us-central1
```

## 2. Set the root password for your instance:

```
gcloud sql users set-password root --instance tutorial-sql-instance \  
  --password [YOUR_SQL_ROOT_PASSWORD] \  
  --host %
```

where `[YOUR_SQL_ROOT_PASSWORD]` is a secure password of your choice.

## 3. Download and run the [Cloud SQL Proxy](https://cloud.google.com/sql/docs/mysql/sql-proxy) (<https://cloud.google.com/sql/docs/mysql/sql-proxy>):

```
wget https://dl.google.com/cloudsql/cloud_sql_proxy.linux.amd64 -O cloud_sql_pr  
cloud_sql_proxy \  
  -dir /tmp/cloudsql \  
  -instances=[YOUR_PROJECT_ID]:us-central1:tutorial-sql-instance=tcp:3306 \  
  -credential_file=[PATH_TO_YOUR_SERVICE_ACCOUNT_JSON]
```

where

- `[YOUR_PROJECT_ID]` is your Google Cloud project ID.
- `[PATH_TO_YOUR_SERVICE_ACCOUNT_JSON]` is the path to the service account JSON file that you downloaded previously.

The following output indicates that the proxy is ready for new connections:

```
Listening on 127.0.0.1:3306 for [YOUR_PROJECT_ID]:us-central1:tutorial-sql-inst  
Ready for new connections
```

## 4. In another terminal window, create a new database and a user:

```
mysql -h 127.0.0.1 -u root --password=[YOUR_SQL_ROOT_PASSWORD]  
mysql> create database tutorialdb;  
mysql> create user 'tutorial-user'@'%' identified by '[YOUR_DATABASE_PASSWORD]'  
mysql> grant all on tutorialdb.* to 'tutorial-user'@'%';  
mysql> exit
```

where:

- `[YOUR_SQL_ROOT_PASSWORD]` is the root password for your Cloud SQL instance.
- `[YOUR_DATABASE_PASSWORD]` is a secure password of your choice.

## Setting up the WordPress project

1. Clone the sample repository:

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

2. Go to the directory that contains the sample code:

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

3. Install dependencies:

```
composer install
```

4. Run the helper script:

```
php wordpress.php setup -n \  
  --dir=./wordpress-project \  
  --db_instance=tutorial-sql-instance \  
  --db_name=tutorialdb \  
  --db_user=tutorial-user \  
  --project_id=[YOUR_PROJECT_ID] \  
  --db_password=[YOUR_DATABASE_PASSWORD]
```

where:

- `[YOUR_PROJECT_ID]` is your project ID.
- `[YOUR_DATABASE_PASSWORD]` is your database password.

The `-dir` parameter specifies the location of your WordPress project.

5. The helper script writes information to `wordpress-project/wordpress/wp-config.php`. Inspect the content of `wp-config.php` to verify that your names, project ID, and database password are correct.

```
if ($onGae) {  
    /** Production environment */  
    define('DB_HOST', ':/cloudsql/[YOUR_PROJECT_ID]:us-central1:tutorial-sql-in  
    /** The name of the database for WordPress */  
    define('DB_NAME', 'tutorialdb');  
    /** MySQL database username */  
    define('DB_USER', 'tutorial-user');  
    /** MySQL database password */  
    define('DB_PASSWORD', '[YOUR_DATABASE_PASSWORD]');  
} else {  
    /** Local environment */  
    define('DB_HOST', '127.0.0.1');  
    /** The name of the database for WordPress */  
    define('DB_NAME', 'tutorialdb');  
    /** MySQL database username */  
    define('DB_USER', 'tutorial-user');  
    /** MySQL database password */  
    define('DB_PASSWORD', '[YOUR_DATABASE_PASSWORD]');  
}
```

## Deploying the WordPress project to the App Engine flexible environment

1. Go to your WordPress project directory:

```
cd wordpress-project
```

2. Deploy the WordPress project:

```
gcloud app deploy \  
    --promote --stop-previous-version app.yaml cron.yaml
```

3. Enter this URL in your browser:

```
https://[YOUR_PROJECT_ID].appspot.com
```

where `[YOUR_PROJECT_ID]` is your Google Cloud project ID.

## Updating WordPress, plugins, and themes

It's important to keep WordPress, plugins, and themes up to date. You can keep these items updated by using the `wp` tool. After an update, you need to re-deploy the WordPress project.

### 1. Update WordPress itself:

```
vendor/bin/wp core update --path=wordpress
```



### 2. Update plugins:

```
vendor/bin/wp plugin update --all --path=wordpress
# Just in case it updates any of the dropins, copy the files:
cp wordpress/wp-content/plugins/batcache/advanced-cache.php \
  wordpress/wp-content/plugins/memcached/object-cache.php \
  wordpress/wp-content
```



### 3. Update themes:

```
vendor/bin/wp theme update --all --path=wordpress
```



### 4. Deploy the project again:

```
gcloud app deploy \  
  --promote --stop-previous-version app.yaml cron.yaml
```



## Cleaning up

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial:

### Delete the project

The easiest way to eliminate billing is to delete the project that you created for the tutorial.

To delete the 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.

If you plan to explore multiple tutorials and quickstarts, reusing projects can help you avoid exceeding project quota limits.

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/PROJ)

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.

## Delete non-default versions of your app

If you don't want to delete your project, you can reduce costs by deleting the non-default versions of your app.

To delete an app version:

1. In the Cloud Console, go to the **Versions** page for App Engine.

[GO TO THE VERSIONS PAGE](https://console.cloud.google.com/appengine/versions) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/APPENGINE/VERSIONS)

2. Select the checkbox for the non-default app version you want to delete.

**Note:** The only way you can delete the default version of your App Engine app is by deleting your project. However, you can [stop the default version in the Cloud Console](https://console.cloud.google.com/appengine/versions) (<https://console.cloud.google.com/appengine/versions>). This action shuts down all instances associated with the version. You can restart these instances later if needed.

In the App Engine standard environment, you can stop the default version only if your app has manual or basic scaling.


3. Click **Delete**  to delete the app version.

## Delete your Cloud SQL instance

To delete a Cloud SQL instance:

1. In the Cloud Console, go to the **SQL Instances** page.

**[GO TO THE SQL INSTANCES PAGE \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/SQL/INSTANCES\)](https://console.cloud.google.com/sql/instances)**

2. Click the name of the SQL instance you want to delete.
3. Click **Delete**  to delete the instance.

## What's next

- Learn how to [run the PHP Bookshelf sample app in the App Engine flexible environment](https://cloud.google.com/php/getting-started/tutorial-app) (<https://cloud.google.com/php/getting-started/tutorial-app>).
- Learn how to [run the PHP Bookshelf sample app on Google Kubernetes Engine](https://cloud.google.com/php/tutorials/bookshelf-on-kubernetes-engine) (<https://cloud.google.com/php/tutorials/bookshelf-on-kubernetes-engine>).
- Learn how to [run the PHP Bookshelf sample app on Compute Engine](https://cloud.google.com/php/tutorials/bookshelf-on-compute-engine) (<https://cloud.google.com/php/tutorials/bookshelf-on-compute-engine>).
- Learn about [WordPress](https://wordpress.org/) (<https://wordpress.org/>).
- Learn about [using WordPress multisite with your own domains and Mailgun](https://wp.gaeflex.ninja/2016/03/25/using-wordpress-multisite/) (<https://wp.gaeflex.ninja/2016/03/25/using-wordpress-multisite/>).
- Try out other Google Cloud Platform features for yourself. Have a look at our [tutorials](https://cloud.google.com/docs/tutorials) (<https://cloud.google.com/docs/tutorials>).

---

*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 our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated December 4, 2019.*