Managing Google Cloud Projects, App Engine Applications, and Billing

Python | Java (/appengine/docs/flexible/java/managing-projects-apps-billing) | Node.js
(/appengine/docs/flexible/nodejs/managing-projects-apps-billing) | Go
(/appengine/docs/flexible/go/managing-projects-apps-billing) | Ruby
(/appengine/docs/flexible/ruby/managing-projects-apps-billing) | PHP
(/appengine/docs/flexible/php/managing-projects-apps-billing) | .NET
(/appengine/docs/flexible/dotnet/managing-projects-apps-billing)

A Google Cloud project contains your App Engine application as well as other Google Cloud resources.

From the Google Cloud Console, Cloud SDK (/sdk/docs), and App Engine Admin API (/appengine/docs/admin-api), you can perform all of your administrative tasks, including the common tasks covered in this topic.

Before you can deploy your apps (/appengine/docs/flexible/python/testing-and-deploying-your-app) to the App Engine flexible environment, you typically need to create or set up the following:

1. A Cloud project (#create)

2. An App Engine application (#create)

3. A billing account (#billing)

Creating a project and application

You can select or create a new Google Cloud project and App Engine application to create and manage a collection of settings, credentials, and your app's metadata.

Only the Owner role (/appengine/docs/flexible/python/access-control) has privileges to create an App Engine application in a Cloud project and that application must exist before any other account can perform tasks on that application. For example, the App Engine application must be created before an account with the App Engine Deployer role can deploy an app using a service account.

tant: Each Cloud project can contain only a single App Engine application, and once created you cannot chang
n (/appengine/docs/locations) of your App Engine application.

gcloudAPI (#api)Console (#console)

After installing Cloud SDK (/appengine/docs/flexible/python/download), an account with the Cloud
project Owner role can run the following `gcloud` commands to create a Cloud project and App Engine
application:

1. Run the following command to create a Cloud project:

   `gcloud projects create` (/sdk/gcloud/reference/projects/create)

2. Run the following command to select a region and create an App Engine application:

   `gcloud app create` (/sdk/gcloud/reference/app/create)

3. Enable billing in the Billing page of the Cloud Console, see Enabling billing (#billing) for details.

⭐ **Tips:**

- If you deploy your apps with the `gcloud app deploy`
  (/sdk/gcloud/reference/app/deploy) command, the `gcloud` tool will ensure that your
  Cloud project includes an App Engine application and prompt you to select a region
  when needed.

- To check if an App Engine application exists in your Cloud project, you can run the
  `gcloud app describe` (/sdk/gcloud/reference/app/describe) command.

Enabling billing

To deploy your apps, you must enable billing. Your account will not be charged if you stay
within your free quota (/appengine/quotas). If your application needs resources that exceed the

free quota, you will be charged for the additional usage.

If you have a billing account when you create a Cloud project, then billing is automatically enabled on that project.

To enable billing on a Cloud project:

1. In the Cloud Console, go to the Billing page:

   Go to the Billing page (https://console.cloud.google.com/projectselector/billing)

2. Select or create a Cloud project.

3. Depending on if a billing account exists or if the selected Cloud project is associated with an account, the Billing page displays one of the following:

   - If billing is already enabled for the selected Cloud project, then the details about the billing account are listed.

   - If no billing account exists, you are prompted to create a billing account and associate it with the selected Cloud project.

   - If a billing account exists, you are prompted to enable billing if the selected Cloud project is not already associated with a billing account. You can also click **Cancel** and then click **Create account** to create and associate a new billing account.

After you enable billing, there is no limit to the amount that you might be charged. To gain more control over your application's costs, you can create a budget and set alerts (#creating_budgets_and_setting_alerts).

Managing billing

Use the Billing page to manage your billing accounts:

1. Go to the Billing page in the Cloud Console:

   Go to the Billing page (https://console.cloud.google.com/billing)

2. Select the account that you want to manage and then navigate to the corresponding page to perform the following management tasks:

- See an overview of your billing account, make payments, and add billing account administrators from the **Overview** page.

- <u>Create budgets and alerts</u> (#creating_budgets_and_setting_alerts) on the **Budgets & alerts** page.

- View your transaction history and download invoices from the **Transactions** page.

- Export your billing data to a BigQuery Dataset from the **Billing export** page.

- Configure your payment account and contacts on the **Payment settings** page.

- Set the method of payment on the **Payment method** page.

For more information about billing, see the <u>Pricing</u> (/appengine/pricing#managing) page.

Creating budgets and setting alerts

You can create a budget for a Cloud project to avoid surprises on your bill and monitor all of your Google Cloud charges from one place. With a budget, you can create alerts that send emails to your billing administrators when charges exceed a certain amount.

To create a budget and set alerts in a Cloud project:

1. In the Cloud Console, go to the Billing page:

   <u>Go to the Billing page</u> (https://console.cloud.google.com/billing)

2. Click the billing account for which you want to create a budget.

3. Click **Budgets & alerts** and then **Create budget** to begin creating a budget for your Cloud project.

4. Define your budget in the **Set budget** section and then specify the percentages for which you want to receive email alerts in the following **Set budget alerts** section.

5. Click **Save** to set the budget and enable alerts for your Cloud project. If you have multiple Cloud projects, you must individually set budgets and alerts in each project.

Disabling an application

Disable an App Engine application to stop your app from running instances and serving requests. No data loss will occur when you disable your app, nor will any configuration settings change. Simply re-enable your App Engine application to startup instances and continue serving the traffic to your app.

Your app's resources in both the standard and flexible environment are disabled and therefore, won't incur charges. However, billing charges can still occur for the other services in your Cloud project, for example you can still incur storage charges for your stored data.

If your app is actively processing a request, it will continue to complete that task and can take up to an hour before your app is completely disabled.

To disable an App Engine application:

1. Go to the Application settings page:

   Go to the Application settings page (https://console.cloud.google.com/appengine/settings)

2. Click **Disable application** and then follow the prompts.

When you want your app to continue serving requests, enable the App Engine application by returning to the same *Application settings* page and then clicking **Enable application**.

Disabling your application stops all serving requests, but your data and state are retained. You will still be billed for applicable charges already incurred or charges for other services running in your project, such as Cloud SQL. To release all the resources used within the project, shut down your project.

Shutting down a project

You can disable billing and release all the Google Cloud resources that are being used in your Cloud project by shutting down that project.

gcloudAPI (#api)Console (#console)

After installing the Google Cloud SDK (/sdk/docs), you run the following `gcloud` command to shut down a Cloud project:

`gcloud projects delete` (/sdk/gcloud/reference/projects/delete)