

Running builds on GitHub

Cloud Build provides a [Cloud Build GitHub app](https://github.com/marketplace/google-cloud-build) (<https://github.com/marketplace/google-cloud-build>) that allows you to automatically build your code each time you push a new commit to GitHub.

This tutorial explains how to install and configure the app, and how to automatically trigger builds on GitHub.

Objectives

In this tutorial, you will:

- Prepare a GitHub repo with some source code to build.
- Install and configure the Google Cloud Build GitHub app.
- Make changes to your source code on GitHub and create a pull request for the changes.
- Observe that the Google Cloud Build app builds your code and publishes results to a pull request.
- View the build results on GitHub and the Cloud Console.
- Learn about the different ways to configure your builds.

Costs

This tutorial uses the following billable components of Google Cloud:

- Cloud Build

First 120 build-minutes per day are free; you will be charged for builds consumed above this threshold. For more information see the [Pricing \(/cloud-build/pricing\)](/cloud-build/pricing) page.

- Container Registry

If you're building using a Dockerfile and storing the built image in Container Registry, you will be charged for storage and the network egress used by your Docker images. For information see the [Container Registry Pricing \(/container-registry/pricing\)](/container-registry/pricing) page.

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 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/projectselector2/home/dashboard) (https://console.cloud.google.com/projectselector2/home/dashboard)

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

4. Create a [GitHub account](https://github.com/) (https://github.com/) if you don't have one already.

5. Enable the Cloud Build API in the target Cloud project.

[Enable Cloud Build API](https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapiv1) (https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapiv1)

Preparing a GitHub repository with source files

To use the Google Cloud Build app, your repository must contain either a [Dockerfile](https://docs.docker.com/get-started/part2/#define-a-container-with-dockerfile) (https://docs.docker.com/get-started/part2/#define-a-container-with-dockerfile) or a [cloudbuild.yaml](/cloud-build/docs/build-config) (/cloud-build/docs/build-config) file to configure your build. Your [Dockerfile](#) or [cloudbuild.yaml](#) file can be located in the root of your repository or in a sub-directory of your repository.

The [Dockerfile](#) is the config file for building Docker containers. If you're using the app for Docker builds, it is sufficient if your repo contains a [Dockerfile](#). The example in this tutorial is configured with a [Dockerfile](#).

[cloudbuild.yaml](#) is the config file for Cloud Build. Use a [cloudbuild.yaml](#) in the following scenarios:

- If you wish to fine-tune your Docker builds, you can provide a `cloudbuild.yaml` in addition to the `Dockerfile`. If your repository contains a `Dockerfile` and a `cloudbuild.yaml`, the Google Cloud Build app will use the `cloudbuild.yaml` to configure the builds.
- If you wish to use the Google Cloud Build app for non-Docker builds.

For instructions on creating a `cloudbuild.yaml`, see [Build configuration overview](#) (`/cloud-build/docs/build-config`) and [Creating a basic build configuration file](#) (`/cloud-build/docs/configuring-builds/create-basic-configuration`).

Forking the example repository

To work through the example in this tutorial, you need a GitHub repository with some source code to build. We are providing a sample repository for this purpose that you must **fork** before continuing.

Use the following steps to fork the sample repository. The source files in the repository include a `helloworld.sh` file and a `Dockerfile`, which you will use to build your Docker image.

1. On GitHub, navigate to [/GoogleCloudBuild/gcbapp-dockerfile-example](#) (<https://github.com/GoogleCloudBuild/gcbapp-dockerfile-example>).
2. On the top-right corner of the page, click **Fork**.



Now you have a copy of the `gcbapp-dockerfile-example` repo with source files.

Note: The steps in this tutorial use a public GitHub repository, but you can use the Google Cloud Build app with private repositories as well.

Installing the Google Cloud Build app

In this section you'll install the **Google Cloud Build** app. This will allow you to connect your GitHub repository with your Cloud project and set up continuous integration for `gcbapp-`


dockerfile-example.

During the installation and set up process you will first be asked to authorize the Google Cloud Build app to connect to Google Cloud Platform. After authorizing, you are redirected to Cloud Console where you'll select your Cloud project. You'll then be redirected back to GitHub.

The steps below provide instructions for installing the app only for the `gcbapp-dockerfile-example` repo; but you can choose to install the app for more or all your repositories.

1. If you have not already done so, [enable the Cloud Build API](https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapis.com) (<https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapis.com>) in the target Cloud project.
2. Go to the GitHub marketplace page for the Google Cloud Build app.
[Open Cloud Build app page](https://github.com/marketplace/google-cloud-build) (<https://github.com/marketplace/google-cloud-build>)
3. Scroll down and click **Setup with Google Cloud Build** at the bottom of the page.
4. If prompted, sign in to GitHub.
5. In the **Edit your plan** page, select or update your billing information and click **grant this app access**. If you have previously purchased the GitHub Cloud Build app and are reinstalling it, skip this step.
6. Select one of the following options based on your business need:
 - **All repositories** - enable all current and future GitHub repositories for access via the Cloud Build app.
 - **Only select repositories** - use the **Select repositories** drop-down to only enable specific repositories for access via the Cloud Build app. You will be able to enable additional repositories at a later time.
7. Click **Install**.
8. Sign in to Google Cloud.

The Authorization page is displayed where you are asked to authorize the **Google Cloud Build** app to connect to Google Cloud Platform.



Authorize Google Cloud Build by GoogleCloudBuild

Authorizing will redirect to
<https://console.cloud.google.com>

9. Click **Authorize Google Cloud Build by GoogleCloudBuild**.

You are redirected to the Cloud Console.

10. Select your Cloud project.

11. Check the consent checkbox and click **Next**.

12. In the **Select repository** page that appears, connect your GitHub repositories to your Cloud project as follows:

a. Confirm the correct GitHub account has been selected.

b. Select the checkbox next to each target repository.

c. Read the consent disclaimer and select the checkbox next to it to indicate that you accept the presented terms.

d. Click **Connect repository**.

If you don't see one or more of your target repositories, click **Edit repositories on GitHub** and repeat the steps above to enable additional repositories in the Cloud Build app.

13. If you wish to create one or more initial default trigger(s) that will trigger builds on any branch push, use the **Create a push trigger** page that appears to those triggers. Select the boxes next to each target repository and click **Create push triggers**. Otherwise, skip this step.

★ **Note:** You can create further triggers as well as modify the ones you create in this step at a later time. See [Creating GitHub app triggers \(/cloud-build/docs/create-github-app-triggers\)](/cloud-build/docs/create-github-app-triggers) for more information.

You now have successfully installed the Google Cloud Build app, connected your chosen repositories to your Cloud project, and created push triggers that will launch your build(s).

Connecting additional repositories

To connect additional repositories to your Cloud project , follow the steps below:

1. Go to the [Cloud Build Manage your GitHub installations page](http://console.cloud.google.com/gcb-github-registration) (<http://console.cloud.google.com/gcb-github-registration>).
2. Select your GitHub account and click **Continue**.
3. Click **Add another project**.
4. Under **Project settings**, select your project or create a new project.
5. Click **Connect repository**.
6. Under **Repository selection**, select all repositories to connect with your project.
7. Optionally, under **Trigger settings**, select all repositories for which you would like to create a push trigger.
8. To create a trigger, click **Create push trigger**. Otherwise, click **Skip for now** to be taken back to your list of connected repositories.
9. Click **Done**.

Build using the Google Cloud Build app

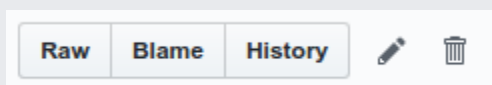
The source files in `gcbapp-dockerfile-example` consist of a simple `helloworld.sh` file and a Dockerfile. In this section, you'll make changes to the code in `helloworld.sh` and create a pull request to check in your changes.

The Google Cloud Build app builds your code when you push a new commit to the repository. Builds for pushed commits included in pull requests will be included in the pull request UI.

1. Open `helloworld.sh` in `gcbapp-dockerfile-example`:

[https://github.com/\[GITHUB_USERNAME\]/gcbapp-dockerfile-example/blob/master/hell](https://github.com/[GITHUB_USERNAME]/gcbapp-dockerfile-example/blob/master/hell)

2. Click on the pencil icon to edit the file.



3. Add the following line at the end of the file:

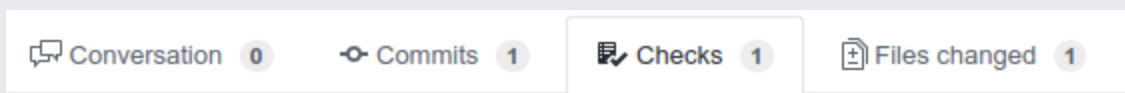
```
echo "The time is $(date)."
```

4. Select **Create a new branch for this commit and start a pull request.** and click **Propose file change.**

5. Click **Create pull request.**

This initiates Cloud Build to build your code.

6. Go to the **Checks** tab.

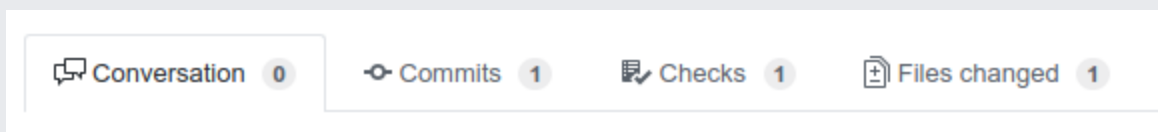


You'll see that Cloud Build has built your changes and you should see that your build has succeeded. You'll also see other build details such as the time it took to build your code, the build ID, etc.

7. Click **View more details on Google Cloud Build.**

The **Build details** page in Cloud Console opens where you can see build information such as status, logs, and build steps.

8. Go to the GitHub tab in your browser and go to the **Conversation** tab.



9. Click **Merge pull request** and then **Confirm merge.**

That's it! You've verified that your code changes builds correctly and checked in your changes.

Note: When you merge your changes, the Google Cloud Build app builds your code again. But the build details of this second build are not displayed in the **Checks** tab. Go to the [Build history page in Cloud Console](https://console.cloud.google.com/cloud-build) (<https://console.cloud.google.com/cloud-build>) to view all your Cloud Build builds.

Additional examples

Here are some sample repositories that contain code examples that use `cloudbuild.yaml` as the configuration file. You can fork the repos and use the steps described in this tutorial to build the code:

- [/GoogleCloudBuild/gcbapp-example](https://github.com/GoogleCloudBuild/gcbapp-example) (<https://github.com/GoogleCloudBuild/gcbapp-example>): A sample repo with source code that contains both a `Dockerfile` and a `cloudbuild.yaml`.
- [/GoogleCloudBuild/gcbapp-buildconfig-example](https://github.com/GoogleCloudBuild/gcbapp-buildconfig-example) (<https://github.com/GoogleCloudBuild/gcbapp-buildconfig-example>): A sample repo with a simple Go application and a `cloudbuild.yaml`.

Cleaning up

After you've finished the Cloud Build tutorial, you can clean up the resources that you created on Google Cloud so they won't take up quota and you won't be billed for them in the future. The following sections describe how to delete or turn off these resources.

Removing the connection between the repo and the Google Cloud Build app

To stop triggering builds using the Cloud Build app, remove the connection between your repository and the app using the steps below:

1. Go to your GitHub profile's settings page:

[Open GitHub profile page](https://github.com/settings/profile) (<https://github.com/settings/profile>)

2. Login to your GitHub account.
3. Under **Personal settings**, click **Applications**.
4. Locate the row with **Google Cloud Build**, and click **Configure**.
5. Under **Repository access**, locate `[YOUR_GITHUB_USERNAME]/gcbapp-dockerfile-example` and click **X**.
6. Click **Save**

You will be redirected to Google Cloud.

7. Login to Google Cloud.
8. In the Cloud Build page, locate the connection between your Cloud project and `gcb-dockerfile-example`, and click **Unlink**.

You've now removed the connection between `gcb-dockerfile-example` and the Google Cloud Build app, and Cloud Build will not trigger builds for the code in `gcb-dockerfile-example`.

Uninstalling the Google Cloud Build app

1. Go to the GitHub app page for Google Cloud Build.

[Open Cloud Build app page \(https://github.com/apps/google-cloud-build\)](https://github.com/apps/google-cloud-build)

2. Click **Configure**.
3. Select your username or organization where you installed the app.
4. Click **Uninstall**.

Deleting the GitHub repository

1. On GitHub, navigate to the main page of the repository.

`https://github.com/[GITHUB_USERNAME]/gcbapp-dockerfile-example`

2. Under your repository name, click **Settings**.
3. Under Danger Zone, click **Delete this repository**.
4. Type the name of your repository to confirm and then click **I understand the consequences, delete this repository**.

Deleting the container images

1. Open the Container Registry page in the Google Cloud Console.

[Open the Container Registry page \(https://console.cloud.google.com/gcr\)](https://console.cloud.google.com/gcr)

2. Select your project and click **Open**.

3. Open **gcbapp-dockerfile-example**.
4. Select all the images and click **Delete**.

The images that you created as part of this tutorial are deleted from your project.

Deleting 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/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 how to [create GitHub app triggers](/cloud-build/docs/create-github-app-triggers) (/cloud-build/docs/create-github-app-triggers).
- Learn how to [access private GitHub repositories](/cloud-build/docs/access-private-github-repos) (/cloud-build/docs/access-private-github-repos).

- [Learn how to create and manage build triggers](#)
(</cloud-build/docs/running-builds/create-manage-triggers>).

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 2020-06-23 UTC.