This topic describes how to automate builds using Cloud Build and Cloud Source Repositories.

You can configure Cloud Build to automatically build a new image any time a user pushes a change to files stored in Cloud Source Repositories. Events that initiate automatic builds are called *build triggers*. These triggers can help ensure that your container images are kept up to date. You can also use them to build and test feature branches.

- Enable the Cloud Build API.

  Enable the Cloud Build API (https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapis.com&redirect=https://cloud.google.com//sourc

- Ensure you have source code in Cloud Source Repositories.
- Ensure you have either a Dockerfile (#using_a_dockerfile) or a Cloud Build build config file (#using_a_build_config_file).

In addition to the prerequisites, you might find the following information helpful:

- Build triggers use *shallow clones* of a repository. With shallow clones, only the single commit that triggered an automatic build is checked out in the workspace. For more information, and to learn how to include more of your repositories history, see Unshallowing clones (#unshallowing_clones).
- If you use another hosted Git provider, such as on GitHub or Bitbucket, and still need to mirror the repository into Cloud Source Repositories, you must have the `cloudbuilds.builds.create` permission for the Google Cloud project with which you're working. This permission is typically granted through the `cloudbuild.builds.editor` (/cloud-build/docs/securing-builds/configure-access-control#roles) role.

  When you set up a build trigger with an external repository for the first time, you'll need to set up authorization with that repository. For more information, see Adding a repository as a remote (/source-repositories/docs/adding-repositories-as-remotes).

  After you set up your external repository, Cloud Source Repositories creates a mirror of your repository.

- For information on the quotas and limits for Cloud Build, see Quotas and limits (/cloud-build/quotas) in the Cloud Build documentation.

To create a new build trigger, follow these steps:

1. In the Google Cloud Console, open the Cloud Build **Build triggers** page.

   Open the Build triggers page (https://console.cloud.google.com/cloud-build/triggers)

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

   The **Triggers** page opens.

3. Click **Create Trigger**.

   The **Create trigger** page opens.

4. In the **Repository** drop-down list, select a repository from the list of available repositories.
5. Fill out the following options:
   - Optional. In the **Description** field, enter a name for your trigger.
   - Optional. In the **Trigger type** list, you can set a trigger to start a build on commits that are made to a particular *branch*, or on commits that contain a particular *tag*. In either case, you can specify a regular expression with the branch or tag value to match.

★ **Note:** Whether based on branch commits or tag commits, builds are only triggered on *pushes to the remote origin*. Builds don't trigger on local changes, pre-submits, or pull requests.

- In the **Build configuration** list, select a config file type to use for each build that the trigger starts. You can select and specify a Dockerfile, or you can select a Cloud Source Repositories configuration file (located in the remote repository). If necessary, review the information on this page about building with a Dockerfile (#using_a_dockerfile) or a build config file (#using_a_build_config_file).

★ **Note:** You can use built-in variables (/cloud-build/docs/api/reference/rest/v1/projects.builds#Build) in the build config file.

6. Click **Create trigger**.

To use a Dockerfile for your build configuration, you'll need to specify the Dockerfile directory and supply a name for the resulting image.

After you provide the Dockerfile and image name, you'll see a preview of the `docker build` command that your build executes and a summary of the trigger configuration.

To use a build config file for your build configuration, you'll need to provide the location of a build config file (/cloud-build/docs/how-to/build-requests).

Once you've set the location, you'll see a summary of the trigger.

To manually test a build trigger, click **Run trigger** on your trigger's entry in the triggers list.

In some cases, you might want to change your source code but don't want to trigger a build, for example, when you update documentation or configuration files.

In such cases, you can include `[skip ci]` or `[ci skip]` in the commit message, and a build will not be triggered.

For example:

If you want to run a build on that commit later, use the **Run trigger** button.

To build your source on a Git repository, Cloud Source Repositories performs a *shallow clone* of the repository. When Cloud Source Repositories performs a shallow clone, it checks out from the workspace only the single commit that triggered the build, and then it builds from that source. Cloud Source Repositories doesn't check out any other branches or history. This is done for efficiency. Builds aren't delayed while Cloud Source Repositories fetches the whole repository and history just to build from a single commit.

To include more of your repo's history in the build, add a build step in your build config file to "unshallow" the clone. For example:

For more information on `git fetch`, see git reference (https://git-scm.com/docs/git-fetch). For instructions on writing a build config file, see Build config overview (/cloud-build/docs/build-config).

- Learn how to start builds manually (/cloud-build/docs/running-builds/start-build-manually) in Cloud Build.
- Learn more about building, testing, and deploying artifacts (/cloud-build/docs/configuring-builds/build-test-deploy-artifacts).