



- Introduction
- Installation
- Learn the basics
- Best practices
- DOCS
- Guides
- Earthfile reference
- The earthly command
- Configuration reference
- Examples
- Misc

CI INTEGRATION

Overview

Build An Earthly CI Image

Pull-Through Cache

Remote BuildKit

Vendor-Specific Guides

Powered By GitBook

Overview

Continuous Integration systems are as varied as the companies that use them. Fortunately, Earthly is flexible enough to fit into most (and where we don't, let us know!). This document serves as a starting point to configuring Earthly in your CI environment.

Setting up Earthly is as easy as three steps:

1. [Installing Dependencies](#)
2. [Installing Earthly](#)
3. [Configuration](#)

We also have instructions for [specific CI systems](#); and special-case instructions for other scenarios (explore the "CI Integrations" category)

Copy link

CONTENTS

- Dependencies
- Installation
 - Option 1: Direct install
 - Option 2: Image
- Configuration
 - Git
 - Docker
 - Earthly
 - Networking & Security
- Examples

Dependencies

Earthly has two software dependencies: `docker` and `git`. Because `earthly` will not install these for you, please ensure they are present before proceeding. These tools are very common, so many environments will already have them installed. If you choose to use our prebuilt containers, these dep

`docker` is used to glean information about the containerization environment, and manage our `earthly-buildkitd` daemon. It is also used to do things like save images locally on your machine after they have been built by Earthly. To install `docker`, use the most recent version of [Docker](#). The versions packaged for many distributions tend to fall behind.

`git` is used to help fetch remote targets, and also provides metadata for Earthly. To install `git`, you can typically use your distributions package manager.

Cookies [Reject all](#)

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept our [cookie policy](#).

Installation

Once you have ensured that the dependencies are available, you'll need to install `earthly` itself.

Option 1: Direct install

This is the simplest method for adding `earthly` to your CI. It will work best on dedicated computers, or in scripted/auto-provisioned build environments. You can pin it to a specific version like so:

```
wget https://github.com/earthly/earthly/releases/download/v0.6.14/earthly-linux-amd64
chmod +x /usr/local/bin/earthly && \
/usr/local/bin/earthly bootstrap
```

It is recommended to install `earthly` as part of the new host's configuration, and not as part of your build. This will speed up your builds, since you do not need to download `earthly` each time; and it will also provide stability in case a future version of `earthly` changes the behavior of a command.

Don't forget to run `earthly bootstrap` when you are done to finish configuration!

Option 2: Image

If a local installation isn't possible, Earthly currently offers two official images:

- `earthly/earthly`, which is a 1-stop shop. It includes a built-in `earthly` and accepts a target to be built as a parameter. It requires a mount for your source code, and an accessible `DOCKER_HOST`.
- `earthly/buildkitd`, which is the same `earthly-buildkitd` container that `earthly` will run on your host. This is useful in more advanced configurations, such as [remotely sharing](#) a single `buildkitd` machine across many workers, or isolating the privileged parts of builds. This feature is experimental.

If you need to provide additional configuration or tools, [consider building your own image for CI](#).

Cookies [Reject all](#)

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept our [cookie policy](#).

Configuration

While `earthly` is fairly configurable by itself, it also depends on the configuration of its dependencies. In a CI environment, you will need to ensure all of them are configured correctly.

Git

If you plan to build any private, or otherwise secure repositories, `git` will need to be configured to have access to these repositories. Please see our [documentation for how to configure access](#).

Docker

Like `git`, `docker` also needs to be configured to have access to any private repositories referenced in the Earthfiles you want to build. Please see our [documentation for how to log in](#) when pushing to many popular repositories.

If your private registry can use a [credential helper](#), configure it according to your `earthly` can also make use of these to provide access when needed. If you need help provisioning `docker` for use with Earthly, see our [guides on configuring many popular registries](#) for details. Finally, the `earthly-buildkitd` daemon requires running in `--privileged` mode, which means that the `docker` daemon needs to be configured to allow this as well. Rootless configurations are currently unsupported.

Cookies [Reject all](#)

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept our [cookie policy](#).

Earthly

earthly has quite a few configuration options that can either be set through a configuration file or environment variables. See our [configuration reference](#) for a complete list of options.

You can also configure earthly by using the `earthly config` command from within a script. This can be useful for some dynamic configuration.

Some options that may make sense in a CI environment are:

Variable	Description
<code>CNI_MTU</code>	In some environments, the MTU externally may be different than the MTU on the internal CNI network, causing the internet to be unavailable. This lets you configure the internal network for when earthly auto-configures the MTU incorrectly.
<code>NO_COLOR</code> / <code>FORCE_COLOR</code>	Lets you force on/off the ANSI color codes. Use this when earthly misinterprets the presence of a terminal. See no-color to enable or disable colors.
<code>EARTHLY_BUILDKIT_HOST</code>	Use this when you have a buildkit instance you would like to use instead of the one earthly manages.

Earthly also has some special command-line switches to ensure best practices are followed within your CI. These come *highly* recommended. Enable these with the `--ci` option, which is shorthand for `--save-inline-cache --strict --no-output`.

Earthly also has a special `--push` option that can be used when invoking a target. In a CI, you may want to ensure this flag is present to push images or run commands that are not typically done as part of a normal development workflow.

If you would like to do cross-platform builds, you will need to install some `binfmt_misc` entries. This can be done by running: `docker run --rm --privileged multiarch/qemu-user-static --reset -p yes`. This installs the needed entries and `qemu-user-static` binaries on your system. This will need to be repeated on each physical box (only once, since its a kernel level change, and the kernel is shared across containers).

To share secrets with earthly, use the `--secret` option to inject secrets into your builds. You could also use our [cloud secrets](#), for a more seamless experience.

Networking & Security

unsupported.

If earthly is connecting to a remote `earthly-buildkitd`, then you will need to take additional steps. See this article for [running a remote BuildKit instance](#).

Examples

Below are links to CI systems that we have more specific information for. If you run into anything in your CI that wasn't covered here, we would love to add it to our documentation. Pull requests are welcome!

- [Jenkins](#)
- [Kubernetes](#)
- [Circle CI](#)
- [AWS CodeBuild](#)
- [GitHub Actions](#)
- [Google Cloud Build](#)
- [GitLab CI/CD](#)

←

Previous
Data collection

Next - CI Integration
Build An Earthly CI Image

→

Last modified 1 mo ago

WAS THIS PAGE

✕
Cookies [Reject all](#)

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept our [cookie policy](#).

✕
Cookies [Reject all](#)

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept our [cookie policy](#).