



# Build from a Dockerfile

We will look for and use a `Dockerfile` at the service's root if it exists.

Railway notifies you when it's using the `Dockerfile` in the build process with the following message in the logs:

```
=====
Using detected Dockerfile!
=====
```

## Custom Dockerfile Path

By default, we look for a file named `Dockerfile` in the root directory. If you want to use a custom filename or path, you can set a variable defining the path.

In your service variables, set a variable named `RAILWAY_DOCKERFILE_PATH` to specify the path to the file.

For example, if your Dockerfile was called `Dockerfile.origin`, you would specify it like this:

```
RAILWAY_DOCKERFILE_PATH=Dockerfile.origin
```

If your Dockerfile is in another directory, specify it like this:

```
RAILWAY_DOCKERFILE_PATH=/build/Dockerfile
```

## Use Config as Code

You can also set your custom Dockerfile path using config as code.

## Using Variables at Build Time

If you need to use the environment variables that Railway injects at build time, which include [variables that you define](#) and [Railway-provided variables](#), you must specify them in the Dockerfile using the `ARG` command.

For example:

```
# Specify the variable you need
ARG RAILWAY_SERVICE_NAME
# Use the variable
RUN echo $RAILWAY_SERVICE_NAME
```

Be sure to declare your environment variables in the stage they are required in:

```
FROM node

ARG RAILWAY_ENVIRONMENT
```

## Cache Mounts

Railway supports cache mounts in your Dockerfile in the following format:

```
--mount=type=cache,id=s/<service id>-<target path>,target=<target path>
```

Replace `<service id>` with the id of the service.

## Target Path

Unsure of what your target path should be? Refer to the [Nixpacks source code](#).

Within the providers directory, find the file that aligns with your respective language or runtime, and check for the variable that indicates the `CACHE_DIR`.

## Example

As an example, within the [python provider definition](#), you can see the

```
PIP_CACHE_DIR is /root/.cache/pip .
```

So the mount command is specified like this:

```
--mount=type=cache,id=s/<service id>/root/cache/pip,target=/root/.cache/pip
```

## Docker Compose

Railway doesn't support docker compose at the moment.



Not finding the answer you need?

[Start a Discussion](#)

[Prev](#)

**Build Configuration**

[Next](#)

**Deployments**

[Edit this file on GitHub](#)