

# With Docker

This examples shows how to use Docker with Next.js based on the [deployment documentation](#). Additionally, it contains instructions for deploying to Google Cloud Run. However, you can use any container-based deployment host.

## How to use

Execute `create-next-app` with `npm` or `Yarn` to bootstrap the example:

```
npx create-next-app --example with-docker nextjs-docker
# or
yarn create next-app --example with-docker nextjs-docker
# or
pnpm create next-app -- --example with-docker nextjs-docker
```

## Using Docker

1. [Install Docker](#) on your machine.
2. Build your container: `docker build -t nextjs-docker .`
3. Run your container: `docker run -p 3000:3000 nextjs-docker .`

You can view your images created with `docker images` .

## In existing projects

To add support for Docker to an existing project, just copy the `Dockerfile` into the root of the project and add the following to the `next.config.js` file:

```
// next.config.js
module.exports = {
  // ... rest of the configuration.
  experimental: {
    outputStandalone: true,
  },
}
```

This will build the project as a standalone app inside the Docker image.

## Deploying to Google Cloud Run

1. Install the [Google Cloud SDK](#) so you can use `gcloud` on the command line.
2. Run `gcloud auth login` to log in to your account.
3. [Create a new project](#) in Google Cloud Run (e.g. `nextjs-docker` ). Ensure billing is turned on.
4. Build your container image using Cloud Build: `gcloud builds submit --tag gcr.io/PROJECT-ID/helloworld --project PROJECT-ID` . This will also enable Cloud Build for your project.

5. Deploy to Cloud Run: `gcloud run deploy --image gcr.io/PROJECT-ID/helloworld --project PROJECT-ID --platform managed` . Choose a region of your choice.

- You will be prompted for the service name: press Enter to accept the default name, `helloworld` .
- You will be prompted for [region](#): select the region of your choice, for example `us-central1` .
- You will be prompted to **allow unauthenticated invocations**: respond `y` .

Or click the button below, authorize the script, and select the project and region when prompted:



## Running Locally

First, run the development server:

```
npm run dev
# or
yarn dev
```

Open <http://localhost:3000> with your browser to see the result.

You can start editing the page by modifying `pages/index.js` . The page auto-updates as you edit the file.

[API routes](#) can be accessed on <http://localhost:3000/api/hello>. This endpoint can be edited in `pages/api/hello.js` .

The `pages/api` directory is mapped to `/api/*` . Files in this directory are treated as [API routes](#) instead of React pages.