Containerize your application

1**Introducing Docker init**

When working with containers, you usually need to create a Dockerfile to define your image and a compose.yaml file to define how to run it.

To help you create these files, Docker has a command called docker init. Run this command in a project folder, and Docker will create all the required files needed. In this guide, you will see how this works.

Open your project folder in the terminal and run the following command:

docker init

Docker will detect the language of your project and prompt you to select a language. You can select your language if it is in the list, or select Other if your language isn't in the list.

docker init walks you through a few questions to configure your project with sensible defaults.

BackNext

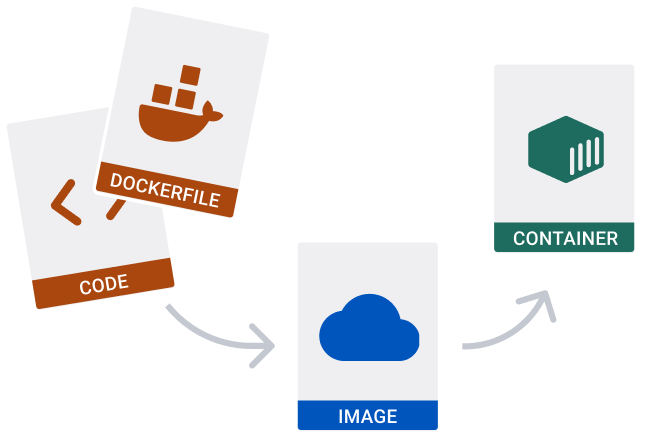
Once you have answered all the questions, you may run **docker compose up** to run your project.

There is a chance, however, that the **Dockerfile** and **compose.yaml** file created for your project need additional changes. In this case, you may need to look up the [Dockerfile reference⁠](https://docs.docker.com/engine/reference/builder/" \t "_blank) and [Compose file reference⁠](https://docs.docker.com/compose/compose-file/) in our documentation. We try our best to do the heavy lifting for you, but sometimes there's some assembly required.

You learned how to containerize your application. Next, learn how to publish your own image to Docker Hub.

Click on publish option

In this guide, you create an image using a Dockerfile and a sample application.



BackNext

Clone the repository at [https://github.com/docker/welcome-to-docker⁠](https://github.com/docker/welcome-to-docker).

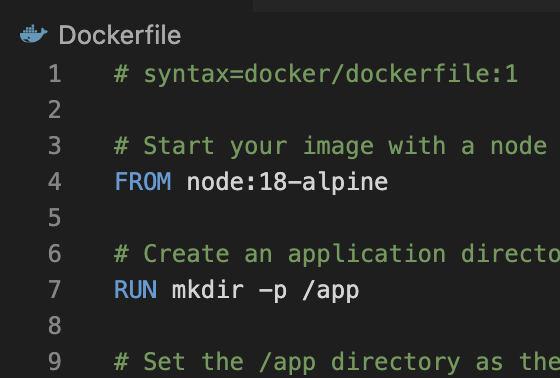
git clone https://github.com/docker/welcome-to-docker

The rest of this guide requires you to run commands in the new project directory. Run the following command before moving on.

cd welcome-to-docker

BackNext

Open the sample application in your IDE. Note that it already has a Dockerfile. For your own projects you need to create this yourself.



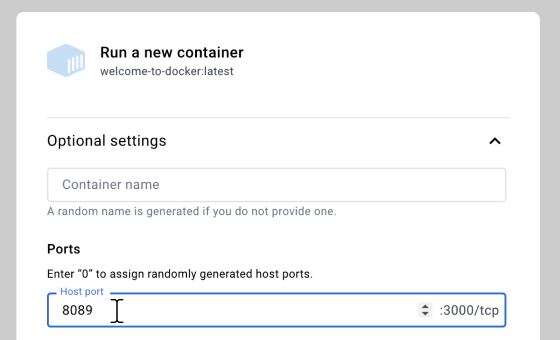
You can build an image using the following **docker build** command via a CLI in your project folder.

docker build -t welcome-to-docker .

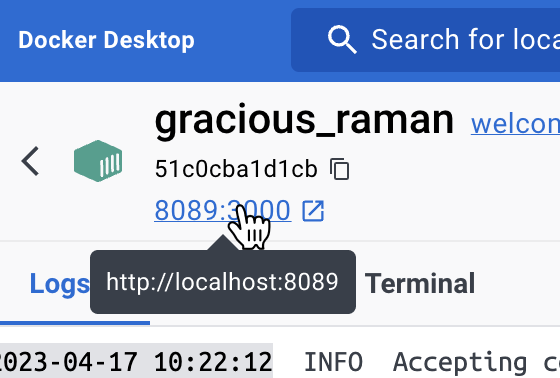
**Breaking down this command**

The **-t** flag tags your image with a name. (**welcome-to-docker** in this case). And the **.** lets Docker know where it can find the Dockerfile.

Once the build is complete, an image will appear in the Images tab. Select the image name to see its details. Select Run to run it as a container. In the Optional settings remember to specify a port number (something like 8089).



You now have a running container. If you don't have a name for your container, Docker provides one. View your container live by selecting the link below the container's name.



Publish your image

1**Get an image**

For this guide, you need at least one image on your computer. If you don't have one, search for the welcome-to-docker image, and select Pull.

