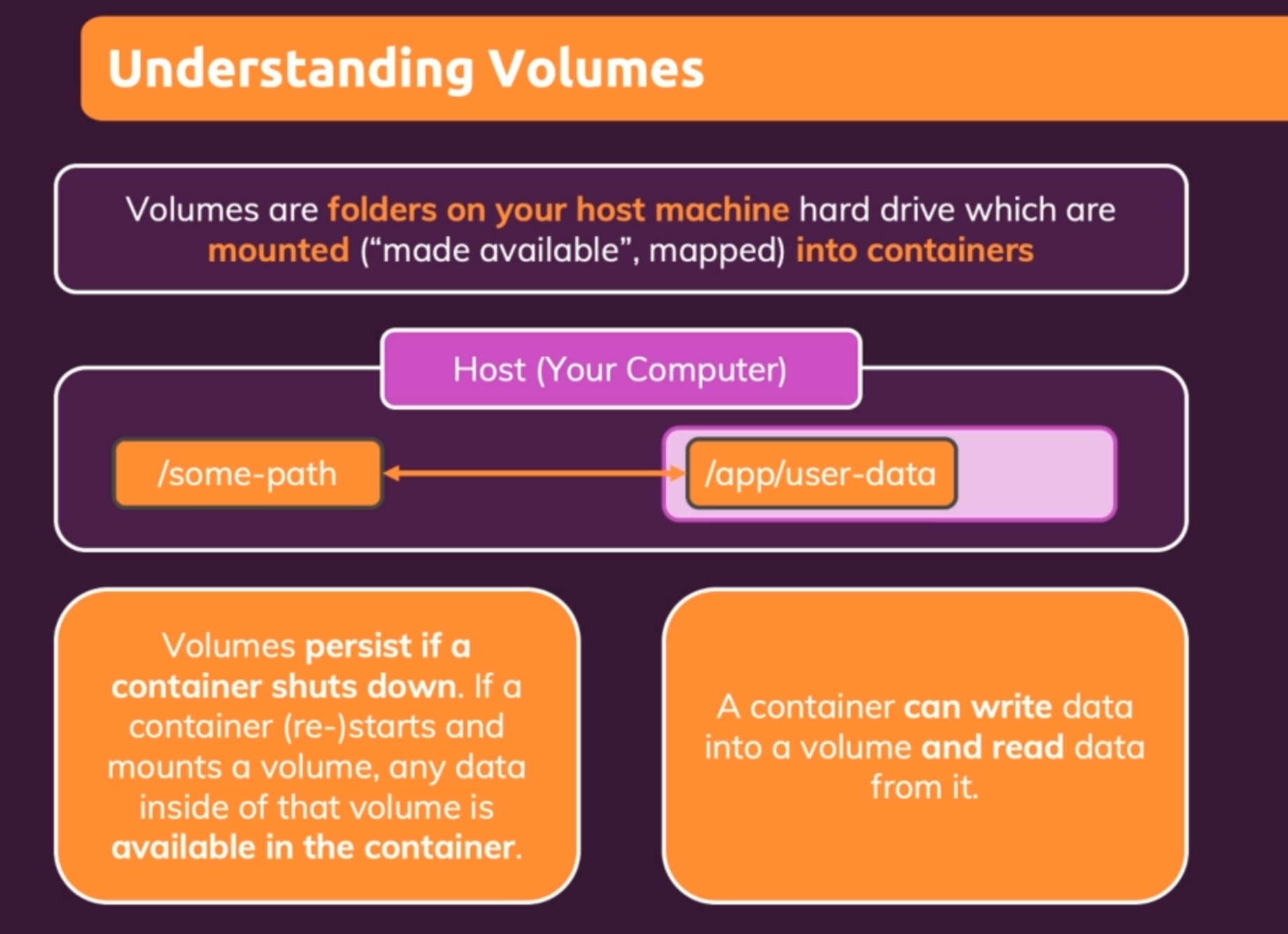
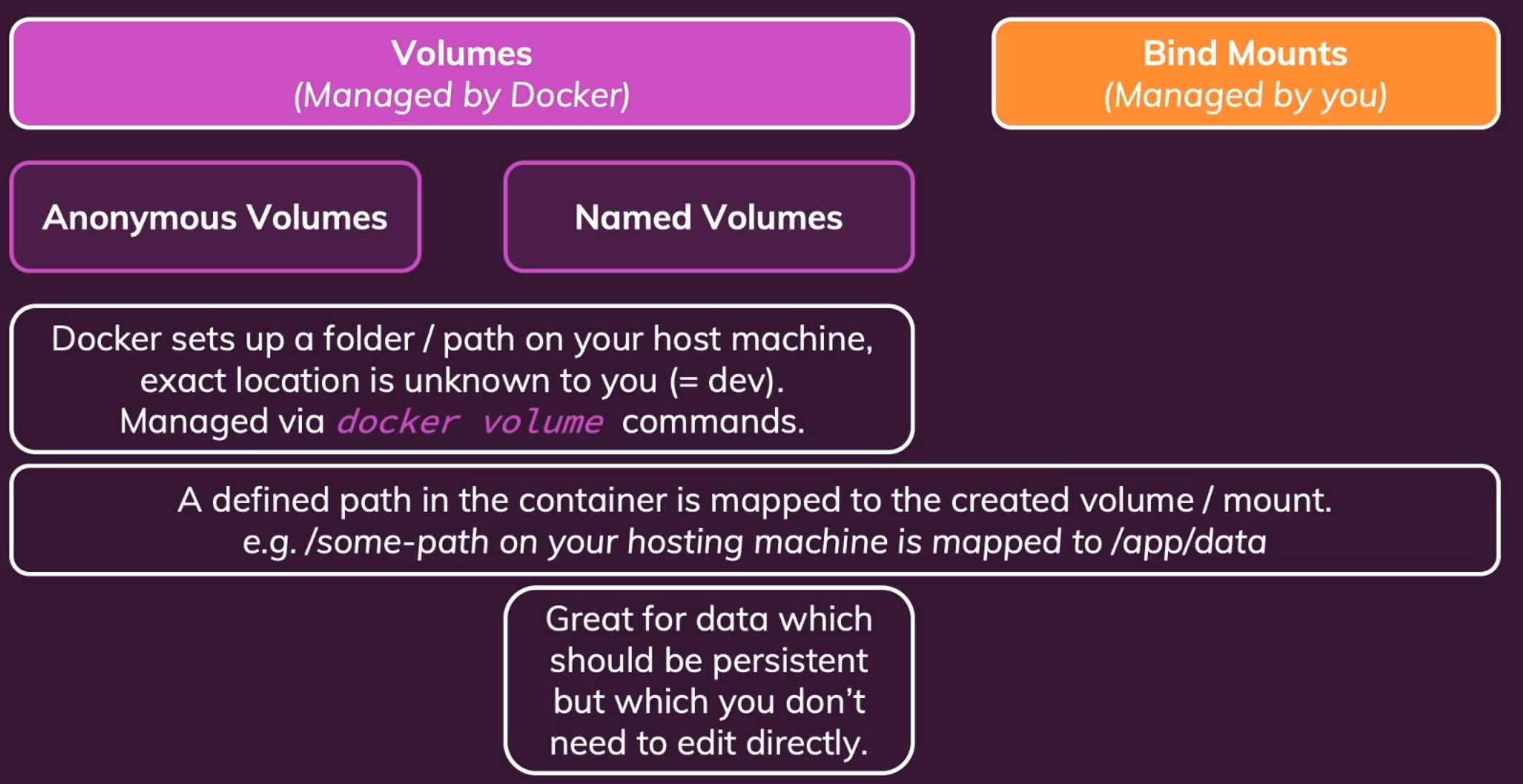
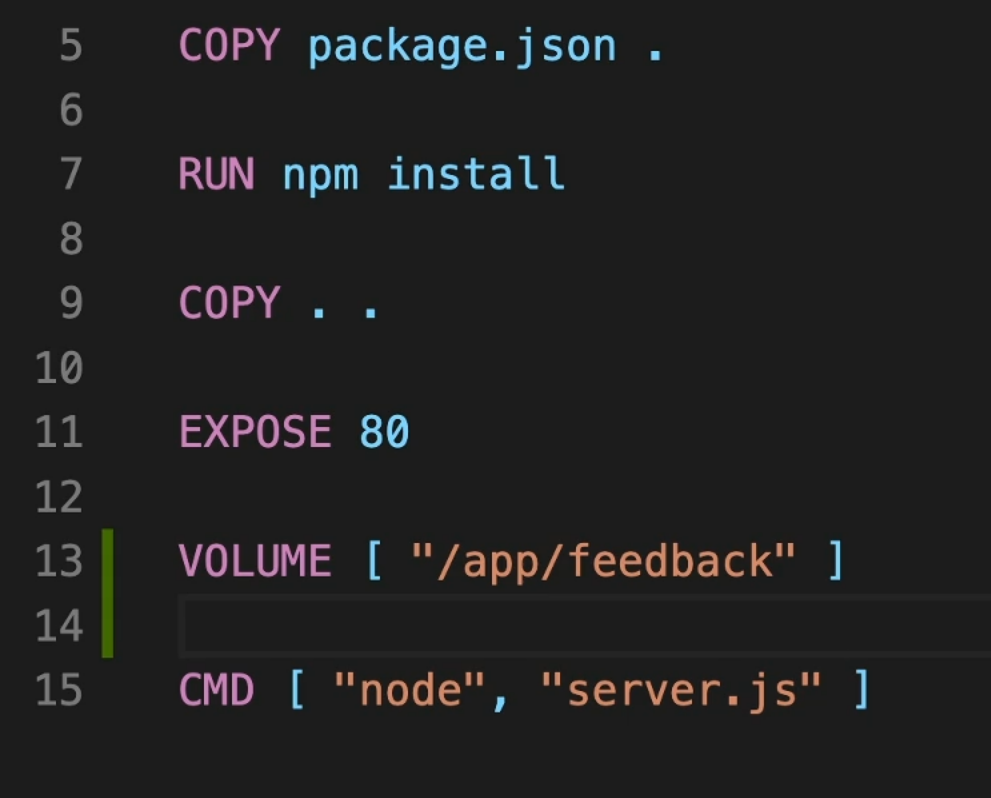
**Managing Data and Working with Volumes**

****

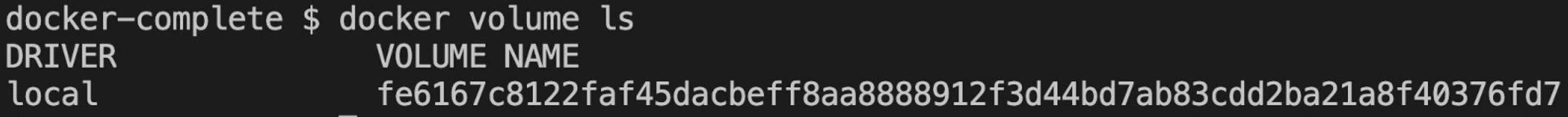






List all the volumes that docker is currently managing

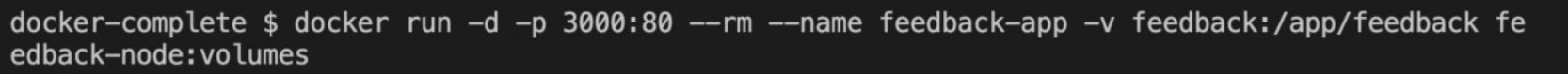
Creating Anonymous Volume



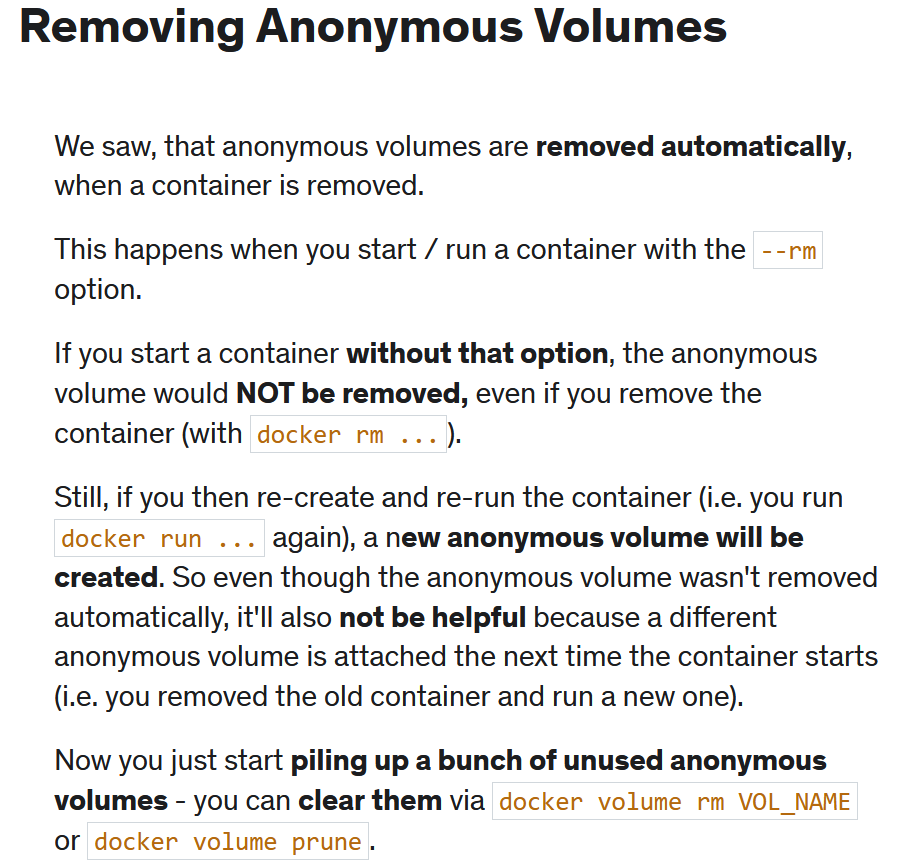
Automatically generated name for anonymous volumes

Note: If we will stop our running container, the anonymous volume will be gone as we have run the docker command with “-rm”, so the container will be deleted and the volume will be gone.

**Named Volume:** To create a named volume we do not need to write instruction in the Dockerfile. We will run the command as below:

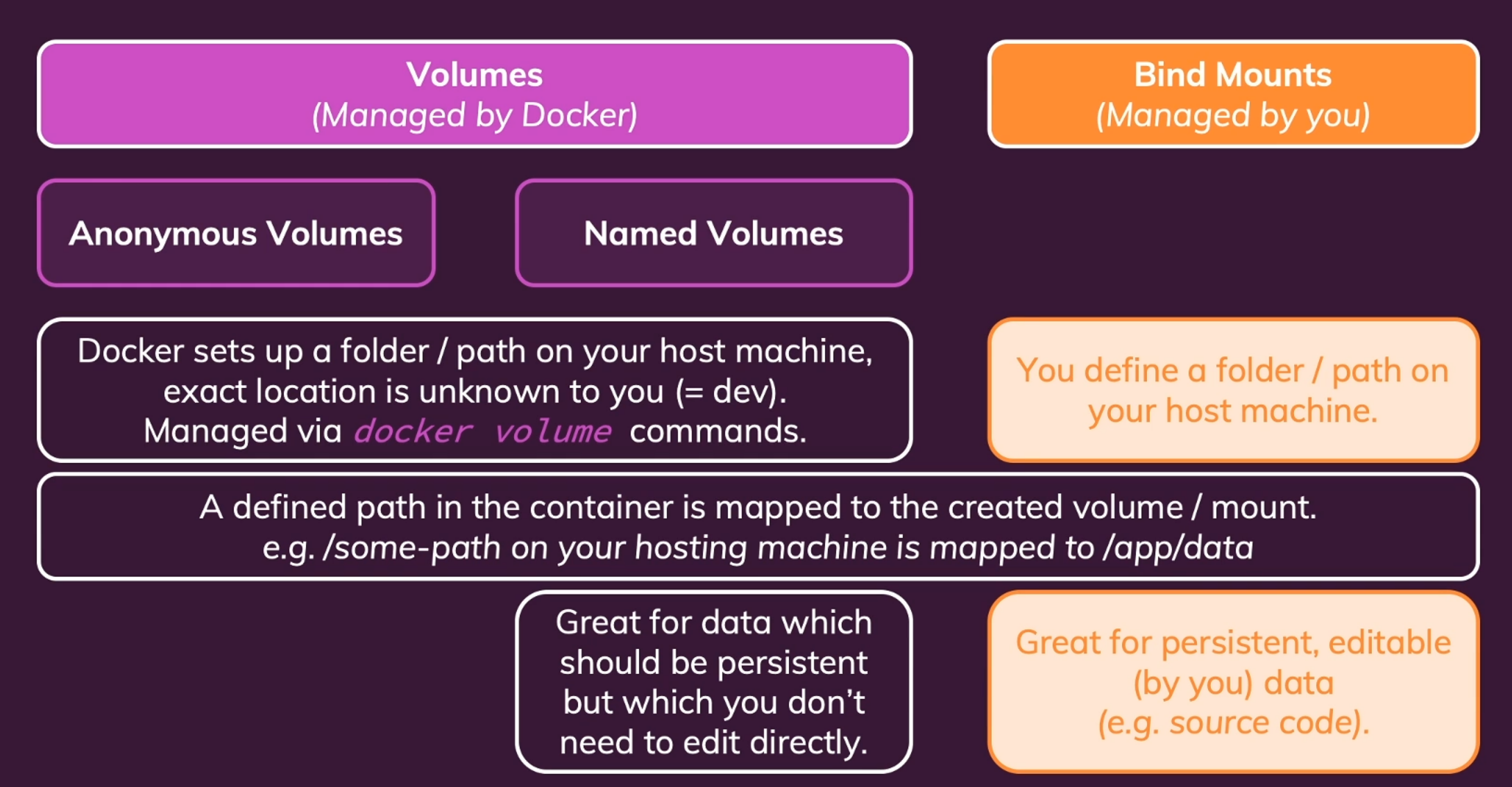


“**-v**” to create volume and *<Volume\_Name>****:****<Path\_to\_the\_folder\_within\_container>*

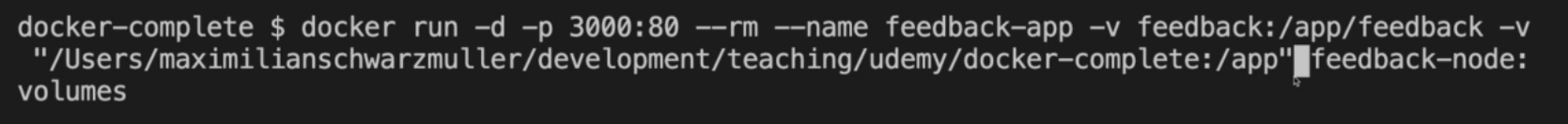


Bind Volumes:

A Bind Mount is **a storage area (file/directory) on your local machine available inside your container**. So any changes you make to this storage space (file/directory) from the outside container will be reflected inside the docker container and vice-versa.



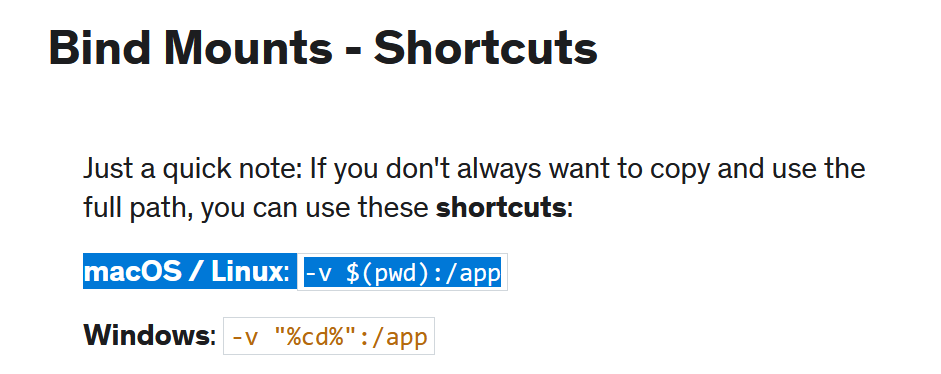
Note: We need to setup the Bind Mount from the Terminal when running the container.



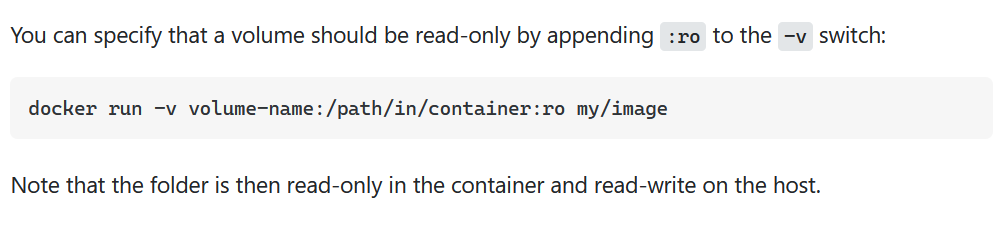
-v for creating a bind mount

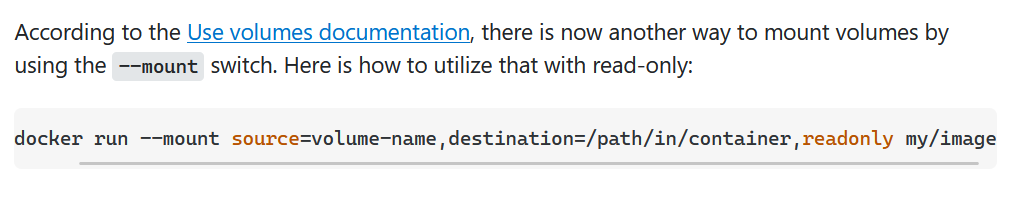
docker run -d -p 3000:80 --rm ---name my\_app ***-v <path\_to\_the\_host\_folder>:<path\_within\_container>*** my\_app:latest

Make Sure that Docker has access to the folder on the host.

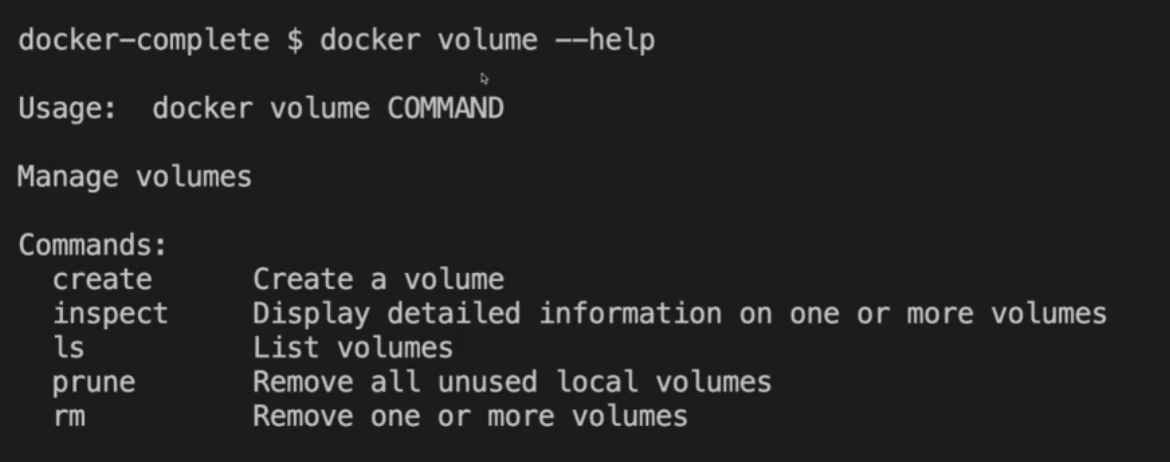


**Read-Only Volumes in Docker**

****

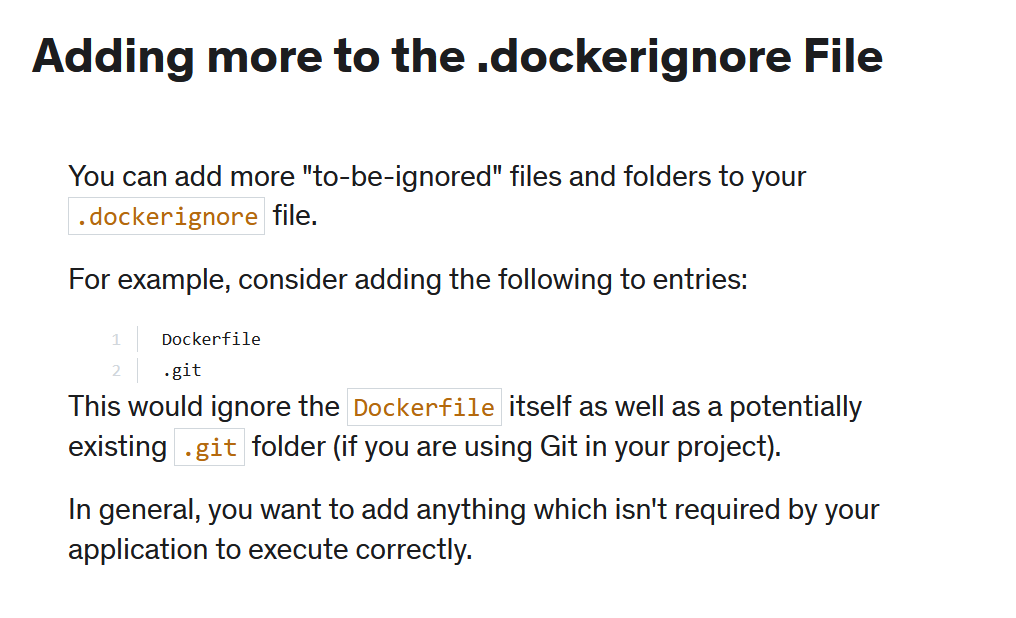
****

**Note:** Bind Mount are mostly used only in development but not in production

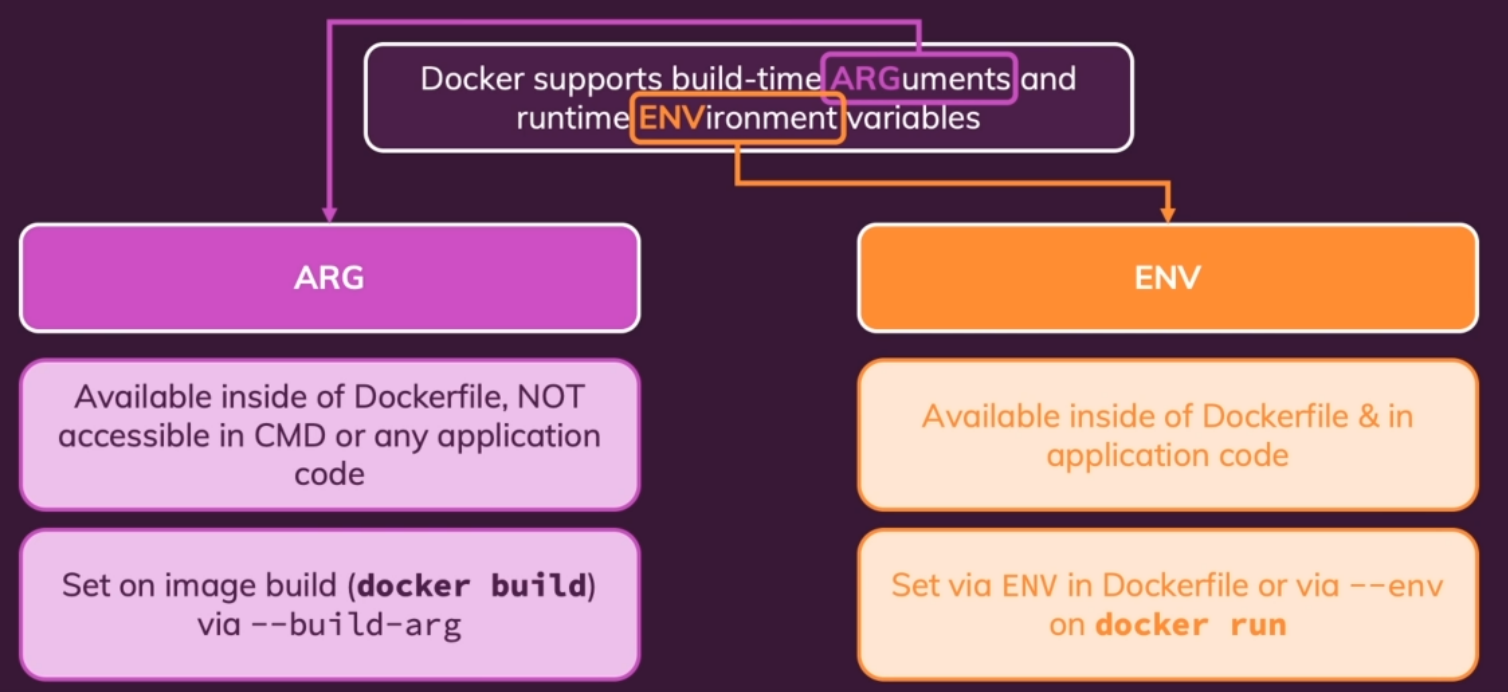
****

What is a Dockerignore file?

A: dockerignore is **a configuration file that describes files and directories that you want to exclude when building a Docker image.**

****

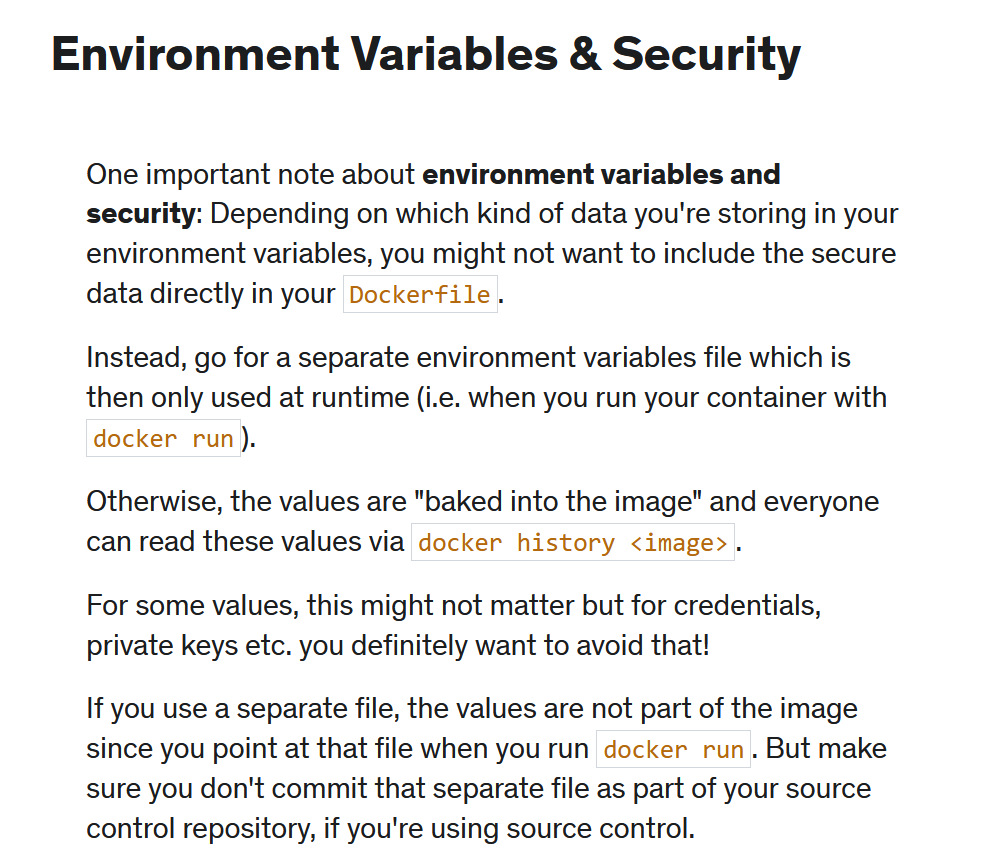
****

****

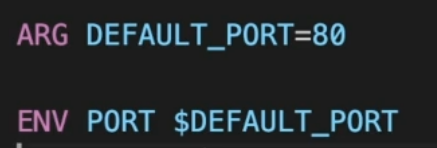
**Example:**

****

We can also create a file by name **.env** and use **--env-file** in the docker run command to give path to that file.



Example of **ARG**uments



Example of passing the **ARG** at docker build command.





