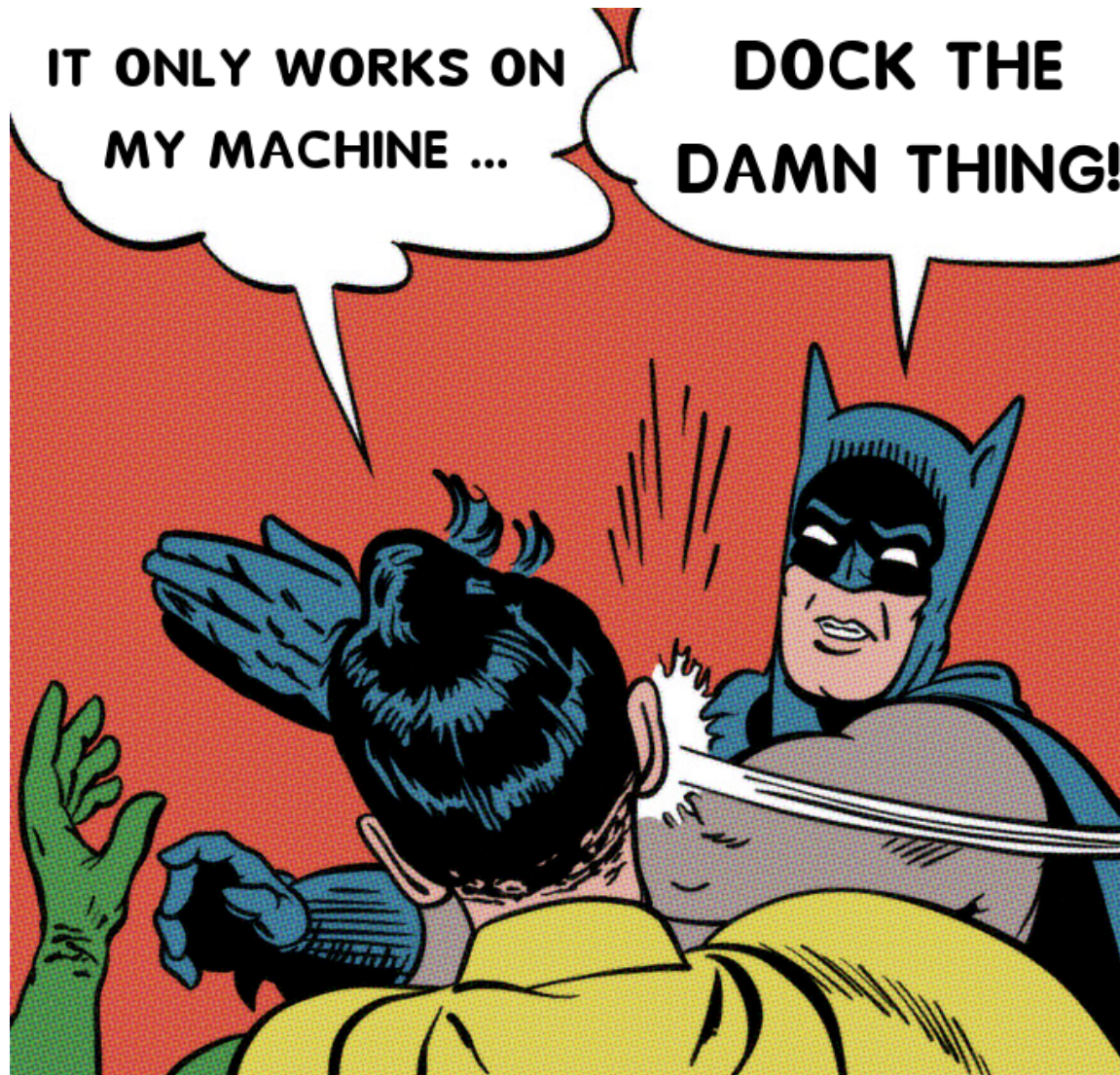


Overview

The Docker platform



"Docker is an open platform for developing, shipping, and running applications. Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allows you to run many containers simultaneously on a given host."

[\(https://docs.docker.com/get-started/overview/\)](https://docs.docker.com/get-started/overview/)

Some Docker concepts

- **Image**: immutable file that contains the source code, libraries, dependencies, tools, and other files needed for an application to run. Think of this as a template.
- **Container**: a light weight isolated virtualized run-time environment with all the requirements for an application. When a Docker image is run, it creates an instance of a container.
- **Volume**: an independent file system entirely managed by Docker and exists as a normal file or directory on the host, where data is persisted

Prerequisites

- Docker
[Get Docker | Docker Documentation](#)
- Docker hub - sign up for an account to be able to access public Docker images and publish your own images. Enable 2FA on this.
[Sign up for a Docker account](#)
- Enable Buildkit
 - This is set by default for Docker Desktop
 - For Linux: set the following
`DOCKER_BUILDKIT=1`

Creating a Docker image

Create a Dockerfile

Following is the Dockerfile used to build a Docker image for pyspi package. This script specifies the architecture of the host system that the Docker image supports, the Python interpreter, installs the required dependencies and sets up a Python shell.

```
FROM --platform=linux/amd64 python:3.9-slim-buster

# Set the working directory
WORKDIR /pyspi_project

# Copy the current directory contents into the container
COPY . .

# Update the package index and install essential packages
RUN apt-get update && apt-get install -y build-essential octave

# Install any needed packages specified in requirements.txt
RUN pip install --upgrade pip setuptools
```

```
RUN pip install --no-cache-dir -r requirements.txt && \
    python setup.py install

# Make port 80 available to communicate with other containers if needed
# EXPOSE 80

# Define environment variable
#ENV NAME

# Run app.py when the container launches
CMD ["python"]
```

Building the Docker image

Navigate to the project directory and run the following in the terminal:

```
docker build --tag <image_name> .
```

Running a container from a Docker image

If the image already exists in your system

You can check what images are available on your system by running the following

```
docker images ls
```

Navigate to the directory of the image and run the following

```
docker run -it <image-name>
```

Note: '-it' flag allows an interactive session, which is what we want in this case since we want to open a Python shell with all the dependencies already installed

Pulling the image from Dockerhub and build

```
# Pull the image from Docker Hub
docker pull <image-name>

# Create a container from the image
docker run --name <container-name> -it <image-name>
```

Building the container from a tar file

If you have a Docker image in a tar file that someone has shared with you, you can use the following steps to load the image into Docker and create a container from it

```
docker load -i /path/to/image.tar
docker run -d --name <container-name> <image-name>
```

Confirming that the container is running

Run the following in the terminal to list the containers that are running

```
docker ps
```

Alternately you can use Docker Desktop to explore existing images, containers and their statuses

Stopping and removing a container

To exit a running container:

```
docker stop <container-name or container-id>
```

To stop and remove a container

```
docker rm <container-name or container-id>
```

Alternatively you can use Docker Desktop to stop and remove containers as well as delete images.

Useful links

Overview

<https://docs.docker.com/get-started/overview/>

<https://docs.docker.com/desktop/>

Dockerfile format

<https://docs.docker.com/engine/reference/builder/>

List of Python base images

https://hub.docker.com/_/python

Good practices

<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008316>

https://docs.docker.com/get-started/09_image_best/

Misc

<https://we-are.bookmyshow.com/understanding-expose-in-dockerfile-266938b6a33d>

<https://pythonspeed.com/articles/docker-build-problems-mac/>