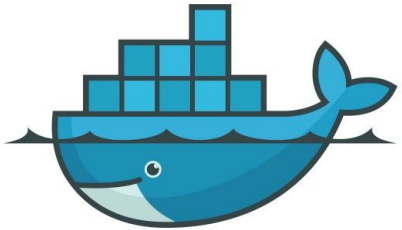
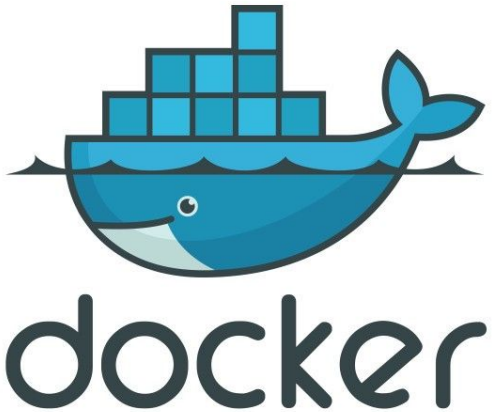


Docker

dockerizing your project

So what is this docker?

An open source project to pack up the environment (libraries, operating system, potentially files) as a lightweight container. If you're a current or aspiring software developer or data scientist, Docker is in your future.



Why docker is so awesome for data science?

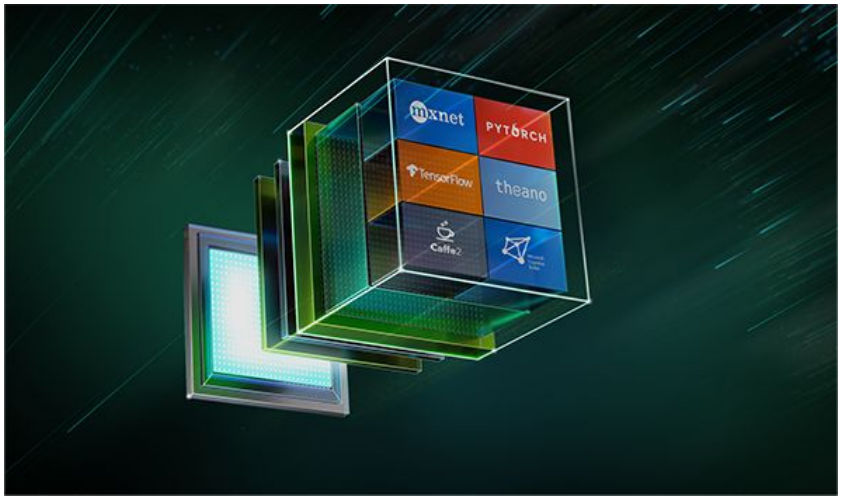
Ever heard these comments from your teammates?

- “Not sure why it’s not working on your computer, it’s working on mine.”
- “Can’t install the package that you used, can you help me out?”

Source: towards data science

Off-the-shelf amazing docker images

You can just use someone else's Docker images and have the best version of things right away.



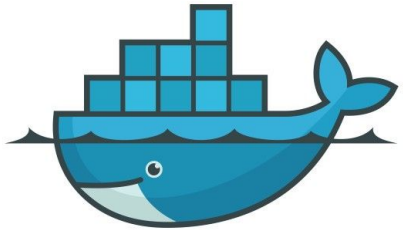
Source: [nvidia gpu cloud]

What will your life be without the docker?

If you want to use multiple machines (for parallel work),

without

Manually install everything per machine, get code over, run code per machine.



with

Create a docker image that works for your code, pull and run on ANY machine as much as you want, it will work without any intervention.

Container

Docker container

Like a physical plastic container, a Docker container:

- **Holds things**
- **Is portable** — It can be used on your local machine, your coworker's machine, or a cloud provider's servers (e.g. GCP).
- **Has clear interfaces for access** — Docker container has several mechanisms for interfacing with the outside world.
- **Can be obtained from a remote location** — an offsite registry keeps an image.Then when you need a container you can make one from the image.
- **Can run multiple containers on a single machine** — you can use the same machine with many different environments, libraries, and operating systems without interfering with each other.

Some concepts

- **Virtual Machines:** Docker is, in a very basic sense, a “virtual machine” (VM) manager. It allows you to “spin up” a virtual machine in a quick and straightforward way.
- **Container:** These are running instances of Docker images.
- **Image:** The file system and configuration(read-only) application which is used to create containers. It contains the **Dockerfile**, libraries, and code your application needs to run, all bundled together.
- **Dockerfile**: is a file with instructions for how Docker should build your image.
- **Registry**: If you want other people to be able to make containers from your image, you send the image to a container registry. Docker Hub is the largest registry and the default.

Spin up your first docker container

```
docker run docker/whalesay cowsay hey
```

One docker image -
docker/whalesay

Container for Deep Learning project

If you quickly setup an environment for a Deep Learning project:


```
docker run -it nvcr.io/nvidia/tensorflow:18.01-py3
```

Elif, you have special requirement for the system setting:

Build your own Docker image from customized Dockerfile

Docker Volume

Let's spin up a container.

```
docker run -it nvcr.io/nvidia/tensorflow:19.07-py3
```

Now lets, mount a directory from our laptop to /data and run the same command as above.

```
docker run -it --rm -v local_dir:container_dir nvcr.io/nvidia/tensorflow:19.07-py3
```

This way anything the container writes to the /data directory will be persisted as long as the content exist in ~/data directory we mounted on.

Docker Volume

There are three main use cases for Docker data volumes

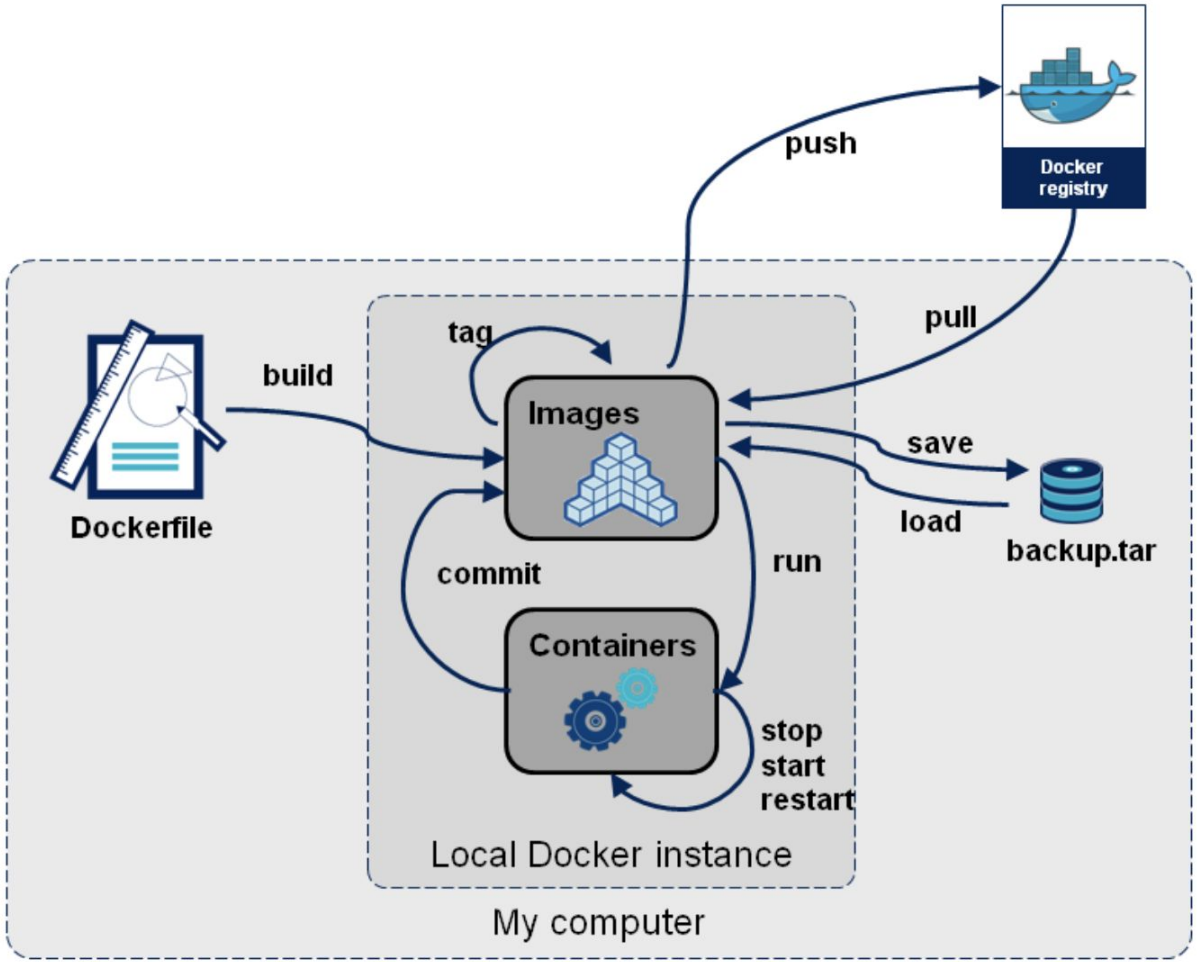
1. To keep data around when a container is removed
2. To share data between the filesystem on your laptop and the Docker container
3. To share data with other Docker containers

Dockerfile

A Dockerfile instruction is a capitalized word at the start of a line followed by its arguments. Each line in a Dockerfile can contain an instruction.

```
FROM nvcr.io/nvidia/tensorflow:19.07-py3

WORKDIR /usr/src/app

COPY requirements.txt ./
RUN pip install --no-cache-dir -r
requirements.txt

COPY . .

CMD [ "python", "./main_script.py" ]
```

`FROM` — specifies the base (parent) image.
`COPY` — copies files and directories to the container.
`RUN` — is used to install packages into an image.
`CMD` — provides Docker a command to run when a container is started.


Docker registry
push
pull
tag
build
Images
save
load
backup.tar
Dockerfile
commit
run
Containers
stop
start
restart
Local Docker instance
My computer