#SOSC24

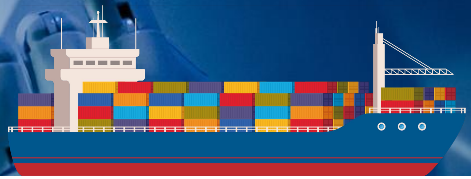
CONTAINERS 101

Daniele Spiga, Diego Ciangottini

INFN PERUGIA

spiga@pg.infn.it

What are containers?



Containers are **a form of virtualization technology** that allows you **to stick an application and all of its dependencies into a single package**. This makes your application **portable, shareable, and reproducible**.

Containers foster portability and reproducibility because they package ALL of an applications dependencies... **including its own tiny operating system!**

This means your application won't break when you port it to a new environment. **Your app brings its environment with it**.

Key concepts (once more)

- **Isolation:** Containers provide a secure and isolated environment for applications, preventing conflicts with other applications or the host system.
- **Portability:** Containers can run consistently across different environments, making it easy to develop and deploy applications.
- **Efficiency:** Containers are efficient in terms of resource usage, as they share the host OS kernel.

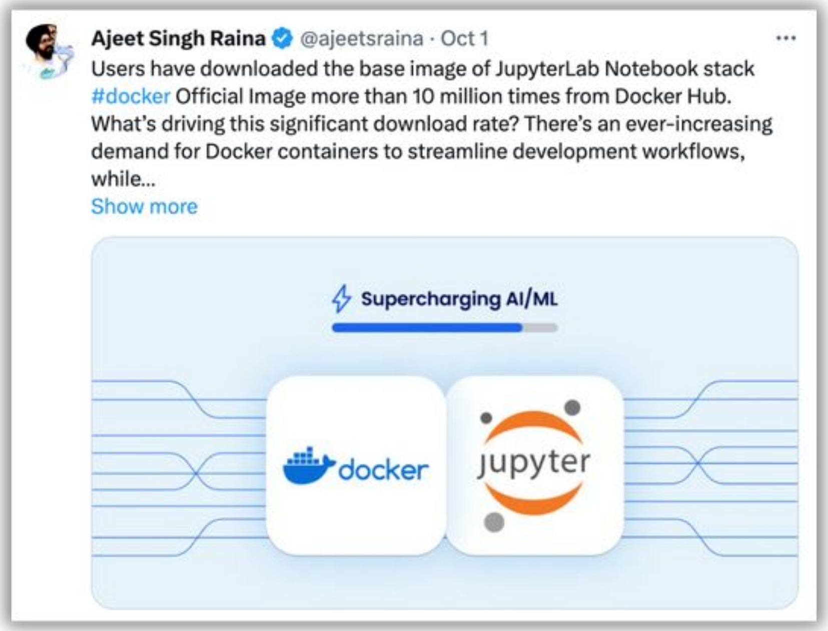
Why do we care?

(Data) science ecosystem speaks container -> unavoidable

The cloud paradigm for resource provisioning is based on containers

In addition you get:

- **Reproducibility:**
  - bring and share your software with ease
- **Manage and scale**
  - your code will be able to leverage a wide set of opportunities on ~any cloud/HPC/HTP provider



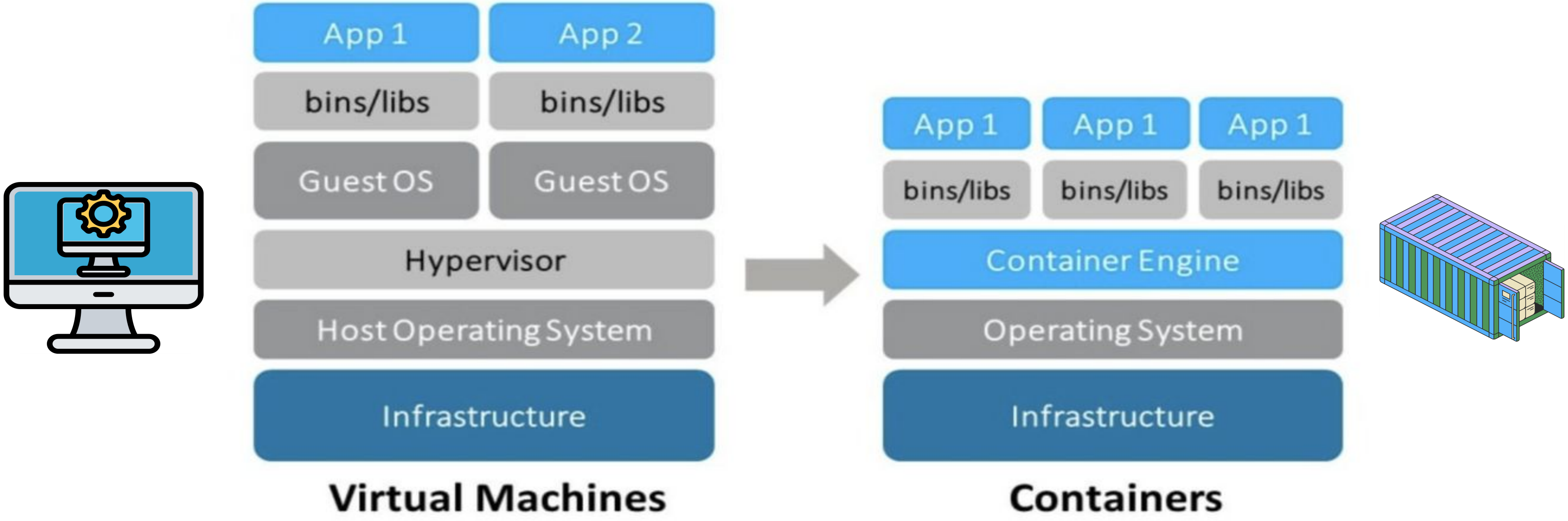
Ajeet Singh Raina @ajeetsraina · Oct 1

Users have downloaded the base image of JupyterLab Notebook stack #docker Official Image more than 10 million times from Docker Hub. What's driving this significant download rate? There's an ever-increasing demand for Docker containers to streamline development workflows, while...

Show more

Virtual Machines vs Containers

The Containers work on the concept of OS-level virtualization, i.e. the **kernel's ability to make multiple isolated environments** on a single host.



Install every last bit of an operating system (OS) right down to the core software that allows the OS to control the hardware (called the kernel)

Containers share a kernel with the host OS.

Pros and cons

SOSC24

VIRTUAL MACHINE

👍

- Provide strong isolation and offer flexibility in choosing different operating systems.
- Heavier, slower to start, and consume more resources due to their independent OS.

👎

CONTAINER

👍

- Lightweight and efficient, sharing the host OS kernel, faster startup and efficient resource usage.
- Limited OS compatibility and less isolation compared to VMs

👎

Who Win ???

VIRTUAL MACHINE VS CONTAINER

Because of their differences, VMs and containers serve different purposes and should be favored under different circumstances.

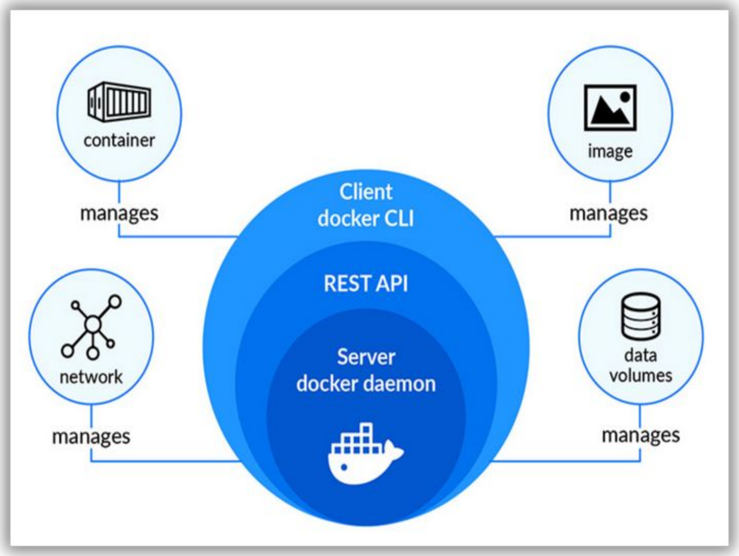
- VMs are good for long **running interactive sessions where you may want to use several different applications**. (Checking email on your preferred client and using Microsoft Word and Excel etc).

- Containers are better suited to **running one or two applications, often non-interactively, in their own custom environments.**

Docker

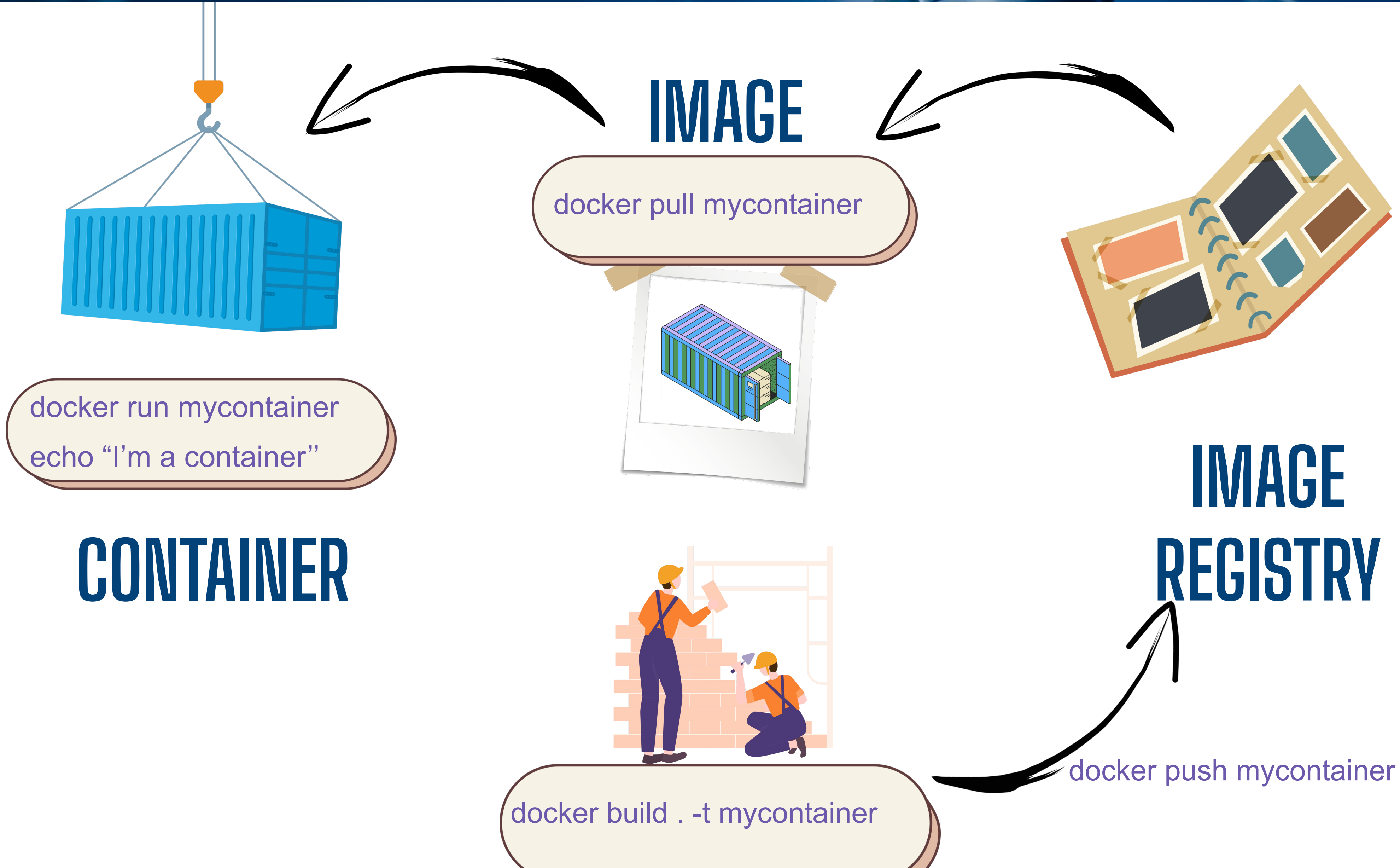
Docker is an **open source platform for building, deploying, and managing containerized applications**

A client-server architecture:



NOTE is currently the most widely used container software. It has several strengths and weaknesses that make it a good choice for some projects but not for others.

Ecosystem components



IMAGE

```
docker pull mycontainer
```

CONTAINER

```
docker run mycontainer
echo “I’m a container”
```

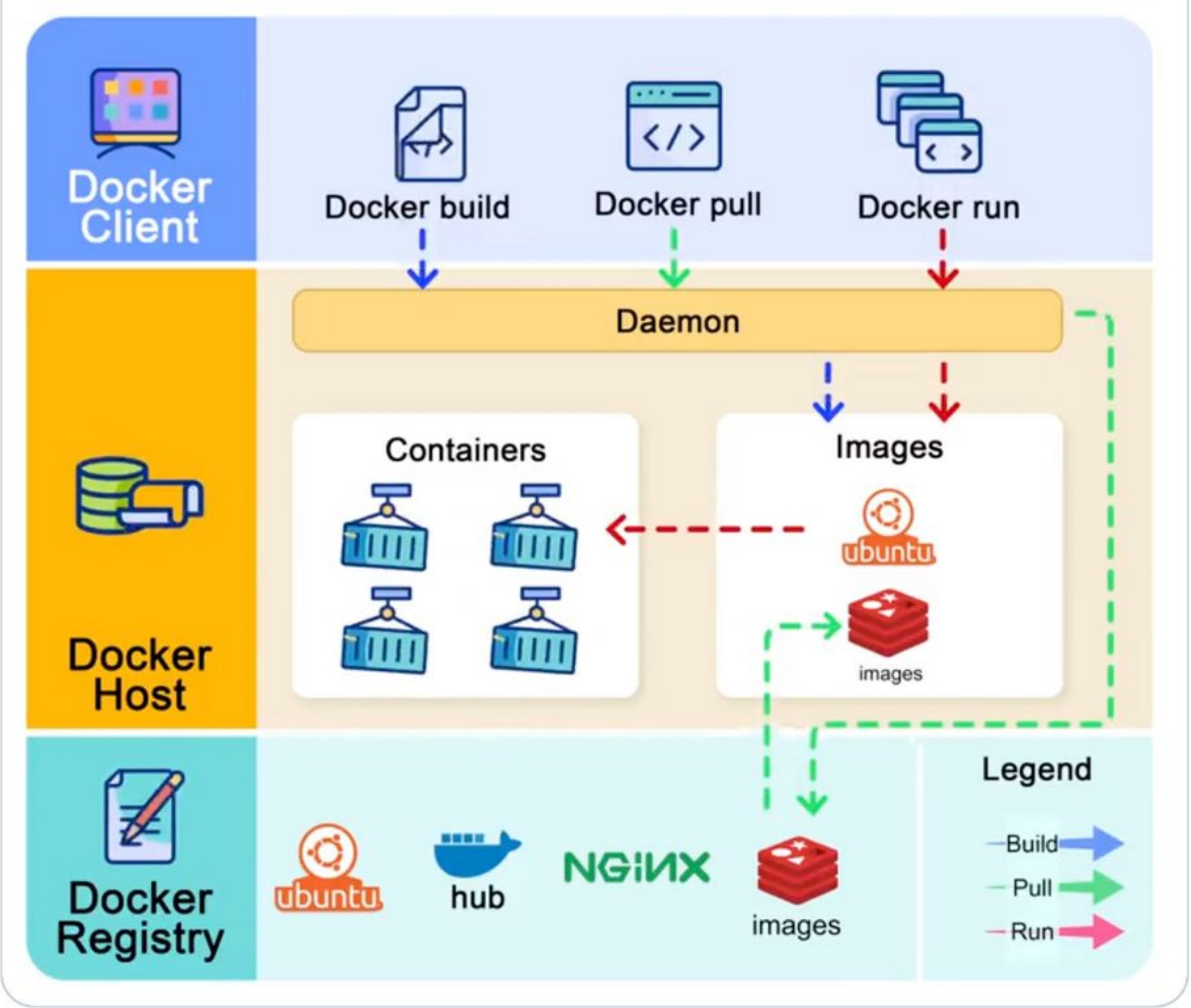
IMAGE REGISTRY

```
docker push mycontainer
```

```
docker build . -t mycontainer
```

We will learn the docker way

SOSC24

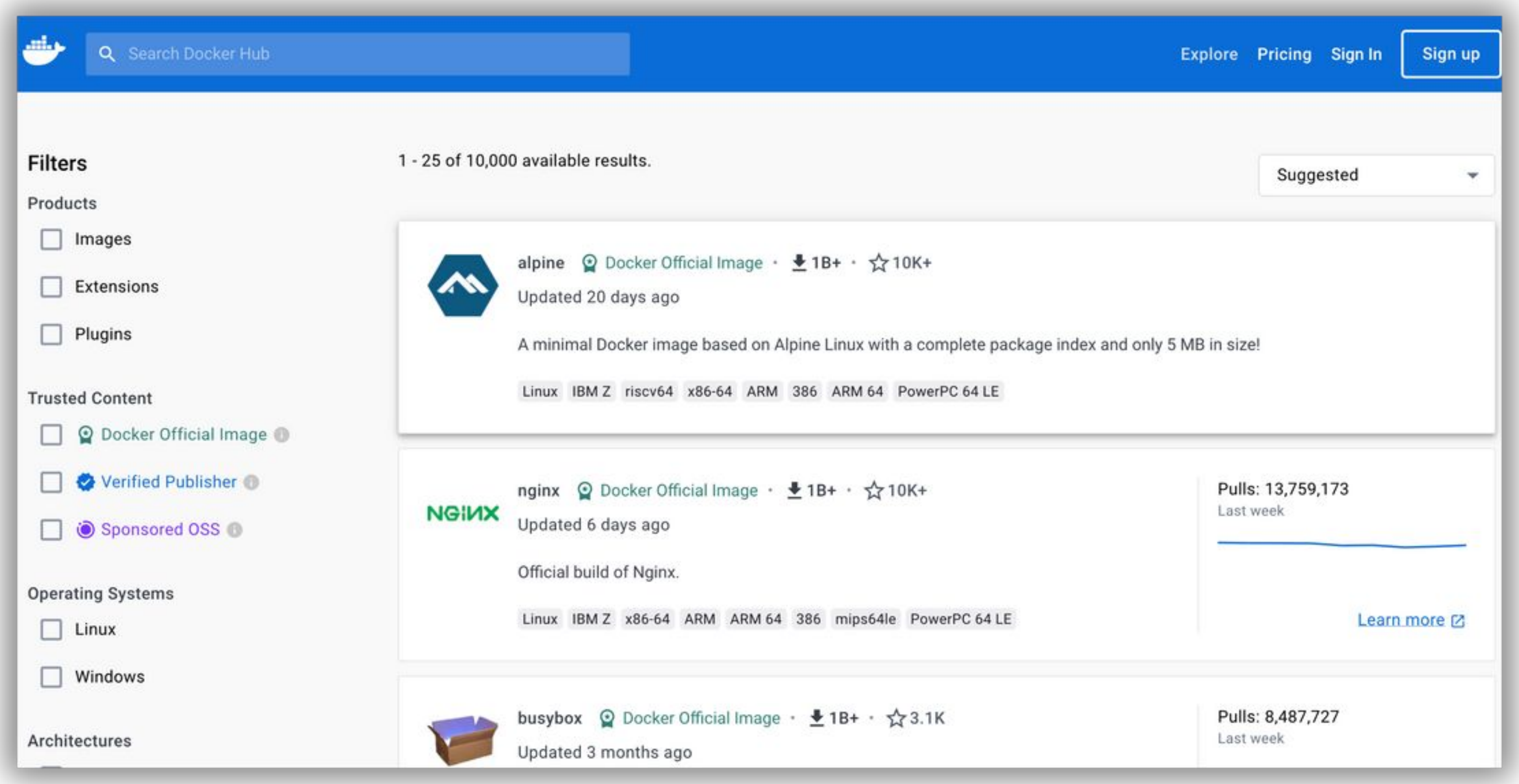


DockerHUB

Default registry if you use docker CLI

(free with some limitation on space and traffic)

https://hub.docker.com/



Filters

Products

□ Images

□ Extensions

□ Plugins

Trusted Content

□ Docker Official Image

□ Verified Publisher

□ Sponsored OSS

Operating Systems

□ Linux

□ Windows

Architectures

1 - 25 of 10,000 available results.

Suggested

alpine · Docker Official Image · 1B+ · ☆ 10K+
Updated 20 days ago
A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!
Linux IBM Z riscv64 x86-64 ARM 386 ARM 64 PowerPC 64 LE

nginx · Docker Official Image · 1B+ · ☆ 10K+
Updated 6 days ago
Official build of Nginx.
Linux IBM Z x86-64 ARM ARM 64 386 mips64le PowerPC 64 LE
Pulls: 13,759,173
Last week
Learn more

busybox · Docker Official Image · 1B+ · ☆ 3.1K
Updated 3 months ago
Pulls: 8,487,727
Last week

Other registries

Other public registries worth mentioning (free to use):

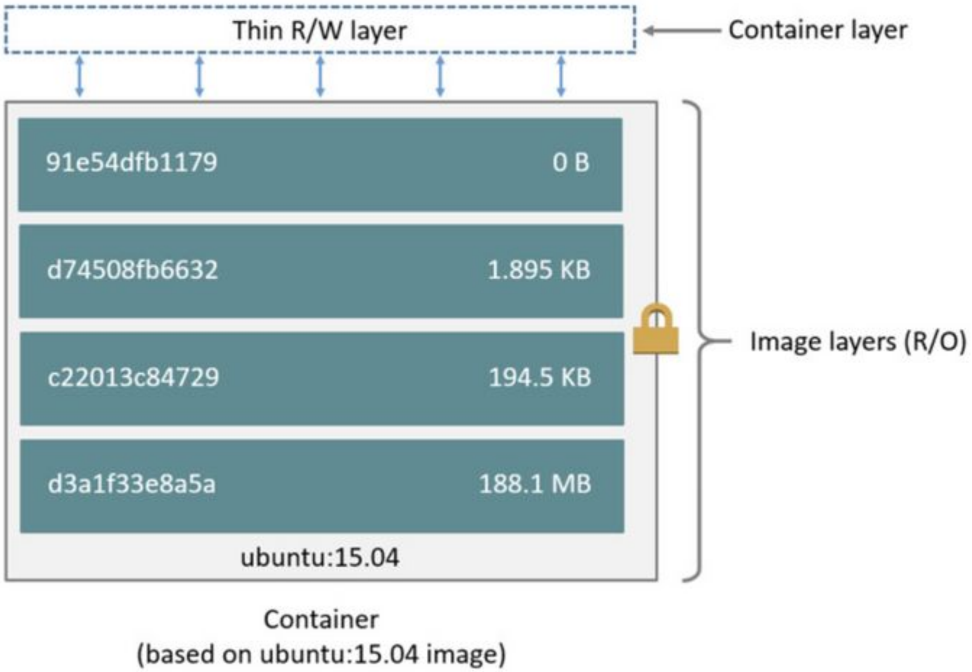
- **GithubContainerRegistry** (ghcr.io)
  - fully integrate in Github CI/CD for images autobuild

Private registries (on-prem)

- **GitLab Container Registry** is tightly integrated with GitLab CI's workflow, with minimal setup.
- **Harbor** (CNCF Graduated project) is an open source registry that secures artifacts with policies and role-based access control, ensures images are scanned and free from vulnerabilities, and signs images as trusted.

Layer by layer

- A Docker Image consists of read-only layers built on top of each other.
- Docker uses the **Union File System** (UFS) to build an image.
- The image is shared across containers.
- Each time Docker launches a container from an image, it adds a thin writable layer, known as the **container layer**, which stores all changes to the container throughout its runtime.

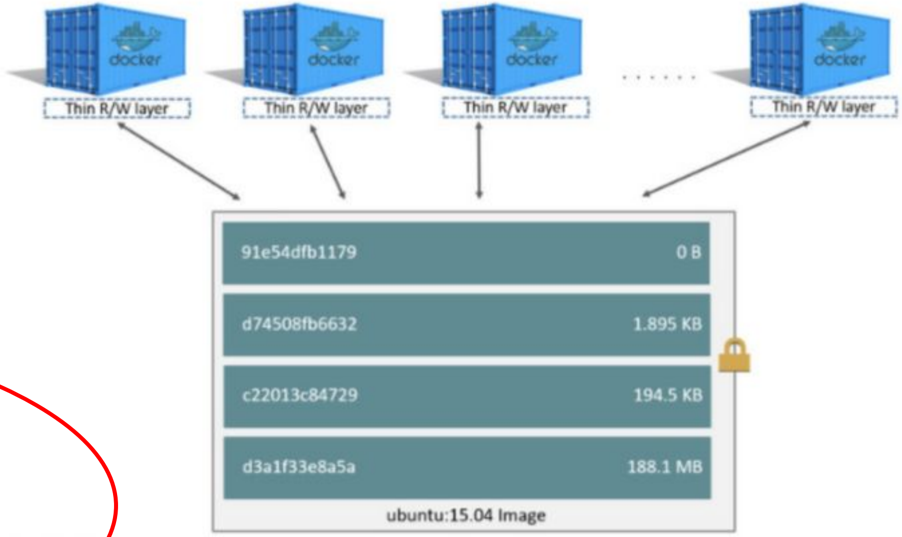


One image many containers

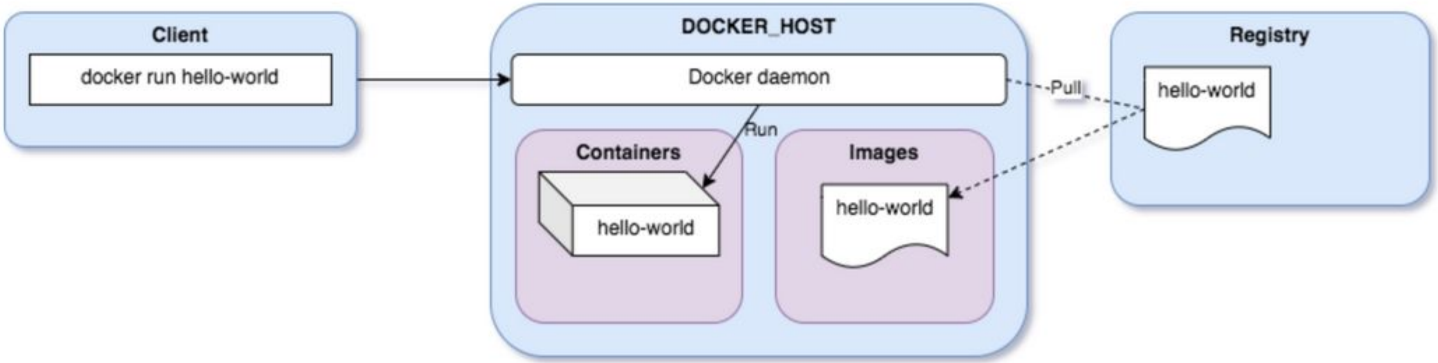
Each container has its own writable container layer, and all changes are stored in this container layer.

Multiple containers can share access to the same underlying image and yet have their own data state.

When the container is deleted, the writable layer is also deleted. The underlying image remains unchanged.



Run a container



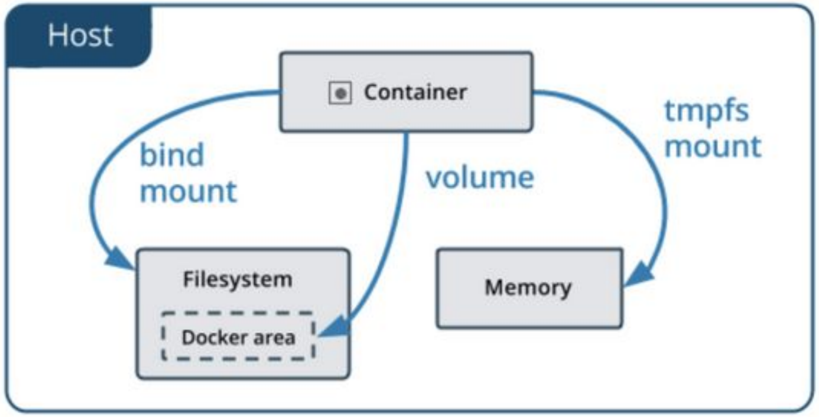
HANDS-ON TIME

[Link to first hands-on]()

Persist data with volumes

Docker provides the following options for containers to store files in the host machine, so that the files are persisted even after the container stops

- volumes
- bind mounts
- tmpfs

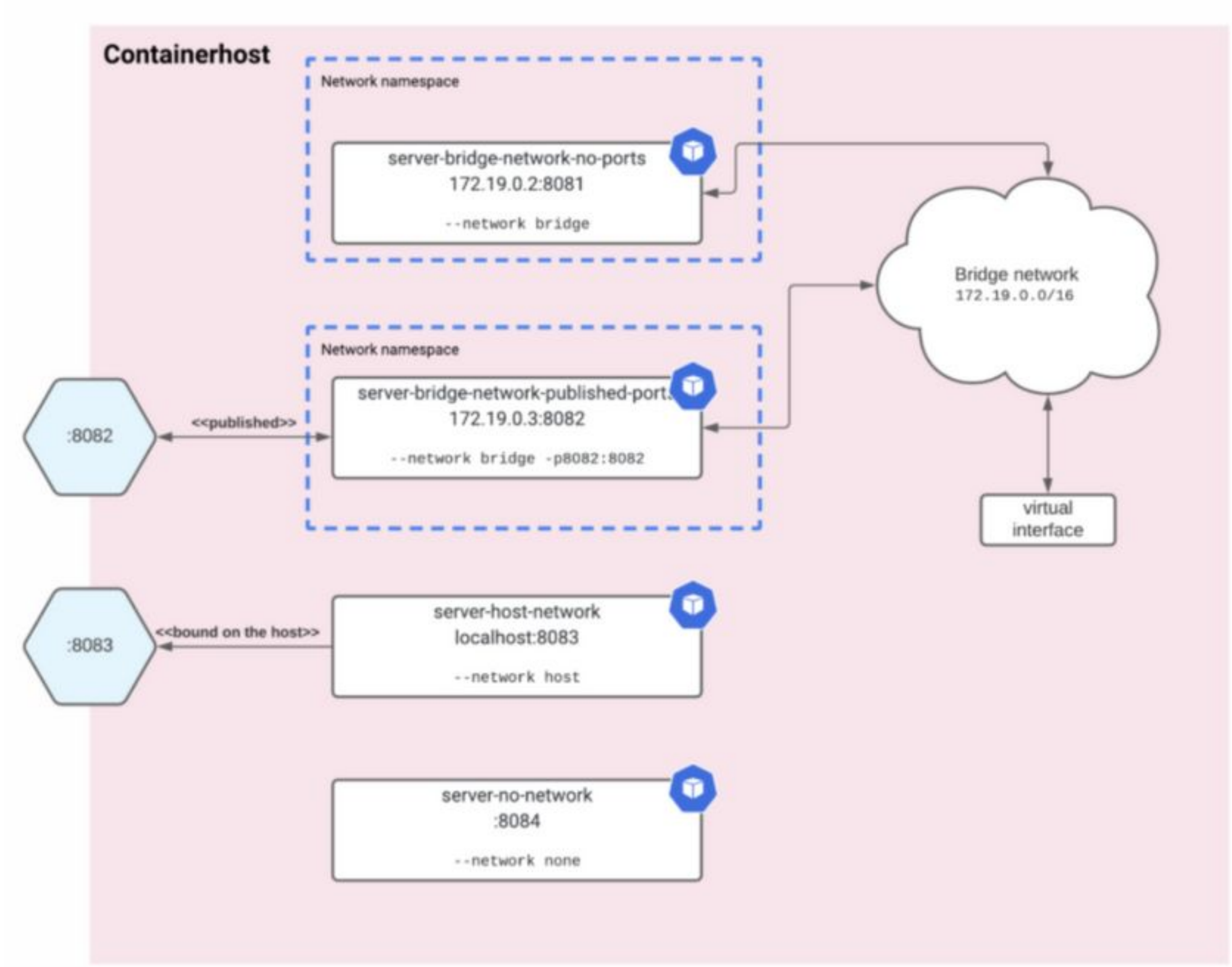


HANDS-ON TIME

[Link to volume exercise]()

Docker networking

Link to network exercise



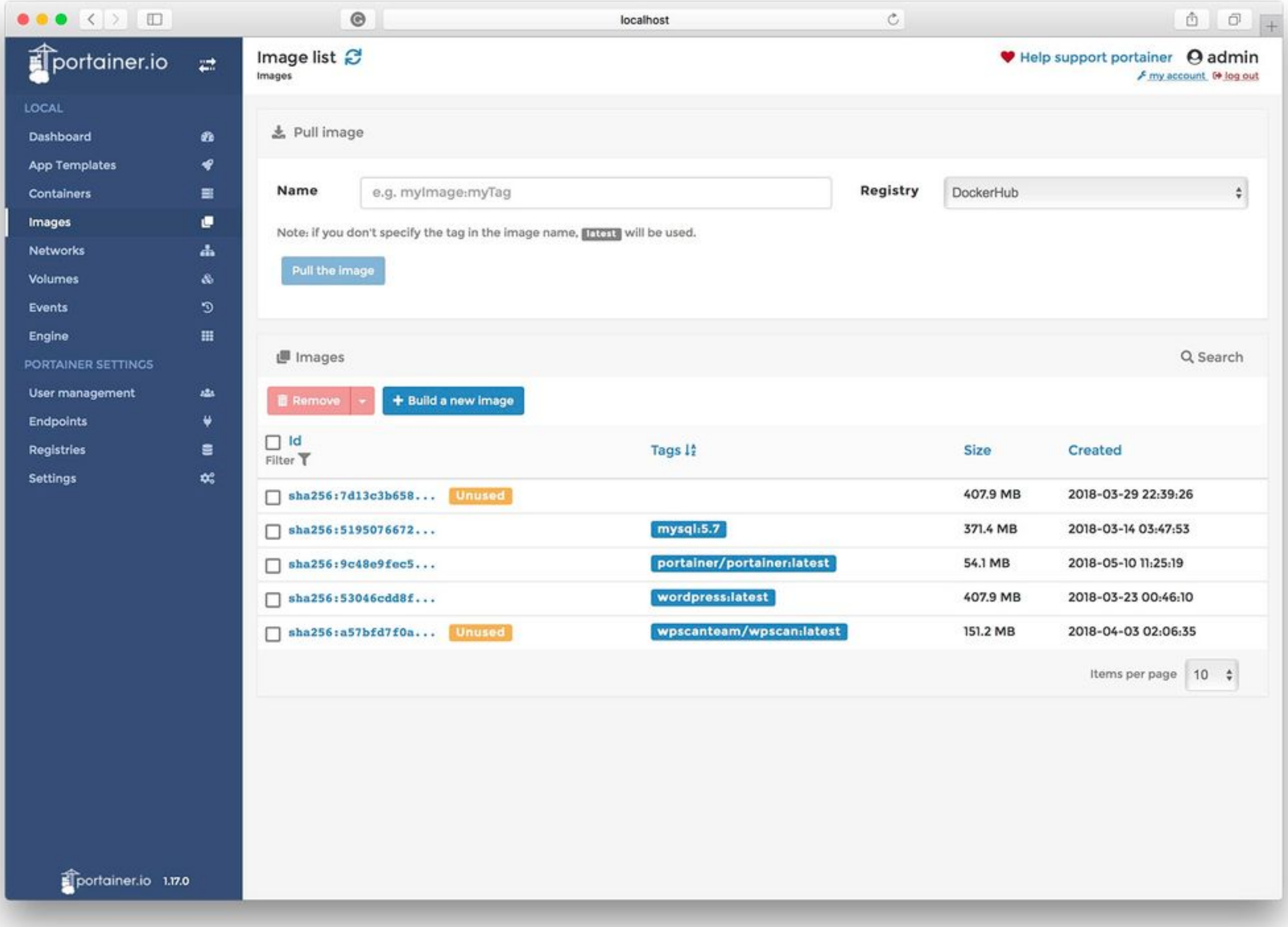
HANDS-ON TIME

MANAGING CONTAINERS

Graphical Interface

Portainer

portainer.io



localhost

portainer.io

LOCAL

- Dashboard
- App Templates
- Containers
- Images
- Networks
- Volumes
- Events
- Engine

PORTAINER SETTINGS

- User management
- Endpoints
- Registries
- Settings

portainer.io 1.17.0

Image list

Images

Help support portainer admin

my account log out

Pull image

Name: e.g. myImage:myTag

Registry: DockerHub

Note: if you don't specify the tag in the image name, latest will be used.

Pull the image

Images

Search

Remove | + Build a new image

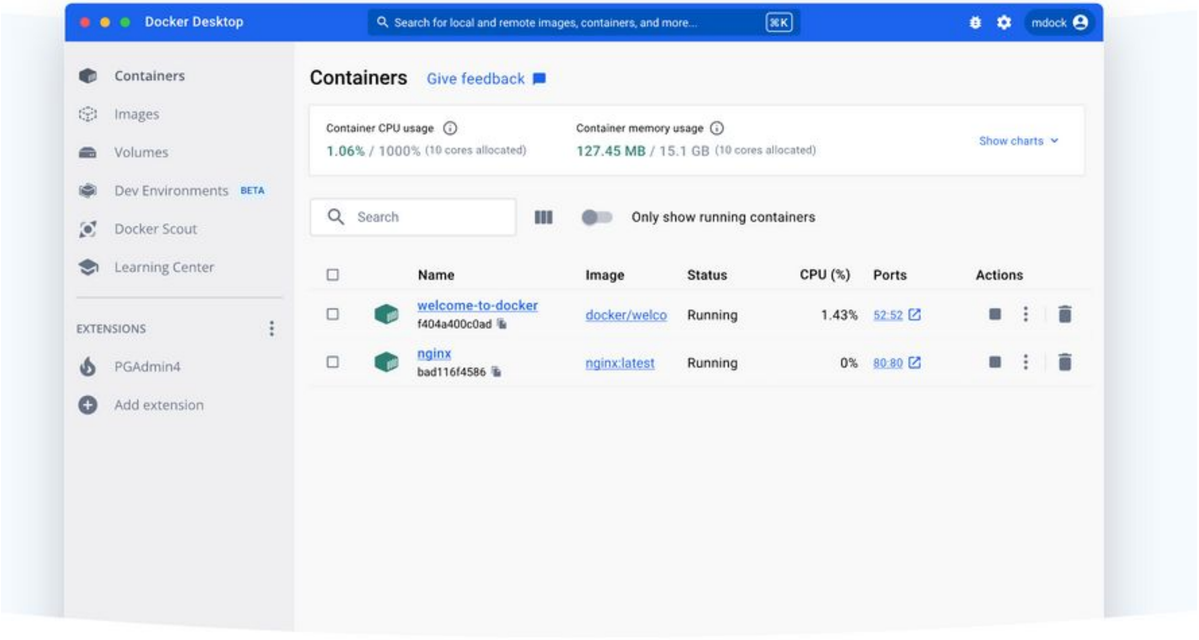
Id	Tags	Size	Created
□ sha256:7d13c3b658... Unused		407.9 MB	2018-03-29 22:39:26
□ sha256:5195076672...	mysql:5.7	371.4 MB	2018-03-14 03:47:53
□ sha256:9c48e9fec5...	portainer/portainer:latest	54.1 MB	2018-05-10 11:25:19
□ sha256:53046cdd8f...	wordpress:latest	407.9 MB	2018-03-23 00:46:10
□ sha256:a57bfd7f0a... Unused	wpscanteam/wpscan:latest	151.2 MB	2018-04-03 02:06:35

Items per page 10

Docker on your laptop

SOSC23

Docker desktop



Container CPU usage
1.06% / 1000% (10 cores allocated)

Container memory usage
127.45 MB / 15.1 GB (10 cores allocated)

□	Name	Image	Status	CPU (%)	Ports	Actions
□	welcome-to-docker f404a400c0ad	docker/welco	Running	1.43%	52:52	
□	nginx bad116f4586	nginx:latest	Running	0%	80:80	

What's Docker Desktop?

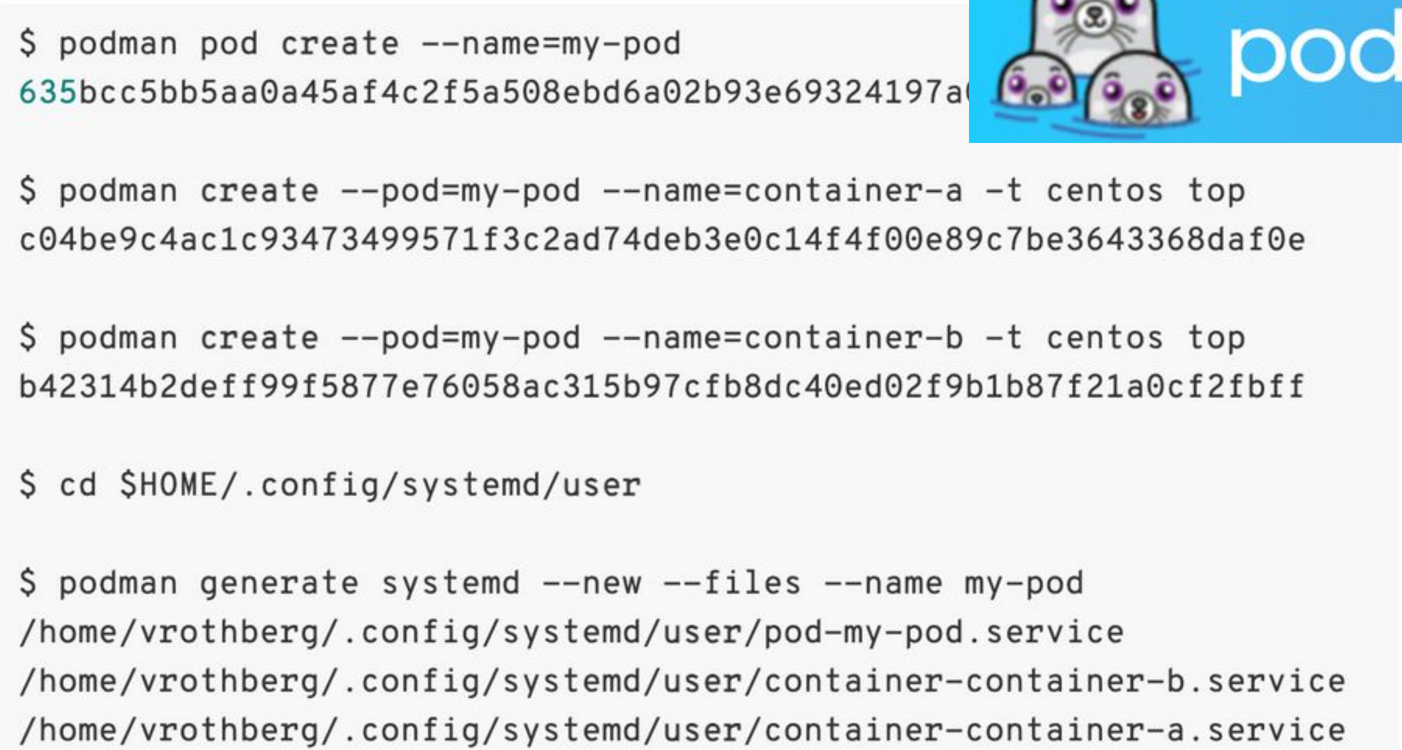
The fastest way to containerize applications

Docker Desktop is secure, out-of-the-box containerization software offering developers and teams a robust, hybrid toolkit to build, share, and run applications anywhere.

Container on systemd with Podman

Podman is an alternative to Docker engine for running containers

WHEN -> it provides features particularly thought for managing long running services



```
$ podman pod create --name=my-pod
635bcc5bb5aa0a45af4c2f5a508ebd6a02b93e69324197a

$ podman create --pod=my-pod --name=container-a -t centos top
c04be9c4ac1c93473499571f3c2ad74deb3e0c14f4f00e89c7be3643368daf0e

$ podman create --pod=my-pod --name=container-b -t centos top
b42314b2deff99f5877e76058ac315b97cfb8dc40ed02f9b1b87f21a0cf2fbff

$ cd $HOME/.config/systemd/user

$ podman generate systemd --new --files --name my-pod
/home/vrothberg/.config/systemd/user/pod-my-pod.service
/home/vrothberg/.config/systemd/user/container-container-b.service
/home/vrothberg/.config/systemd/user/container-container-a.service
```

SOSC24

BUILD YOUR FIRST IMAGE

Build you first container image

SOSC24

[Link to build image - part 1](#)

[Dockerfile](#)

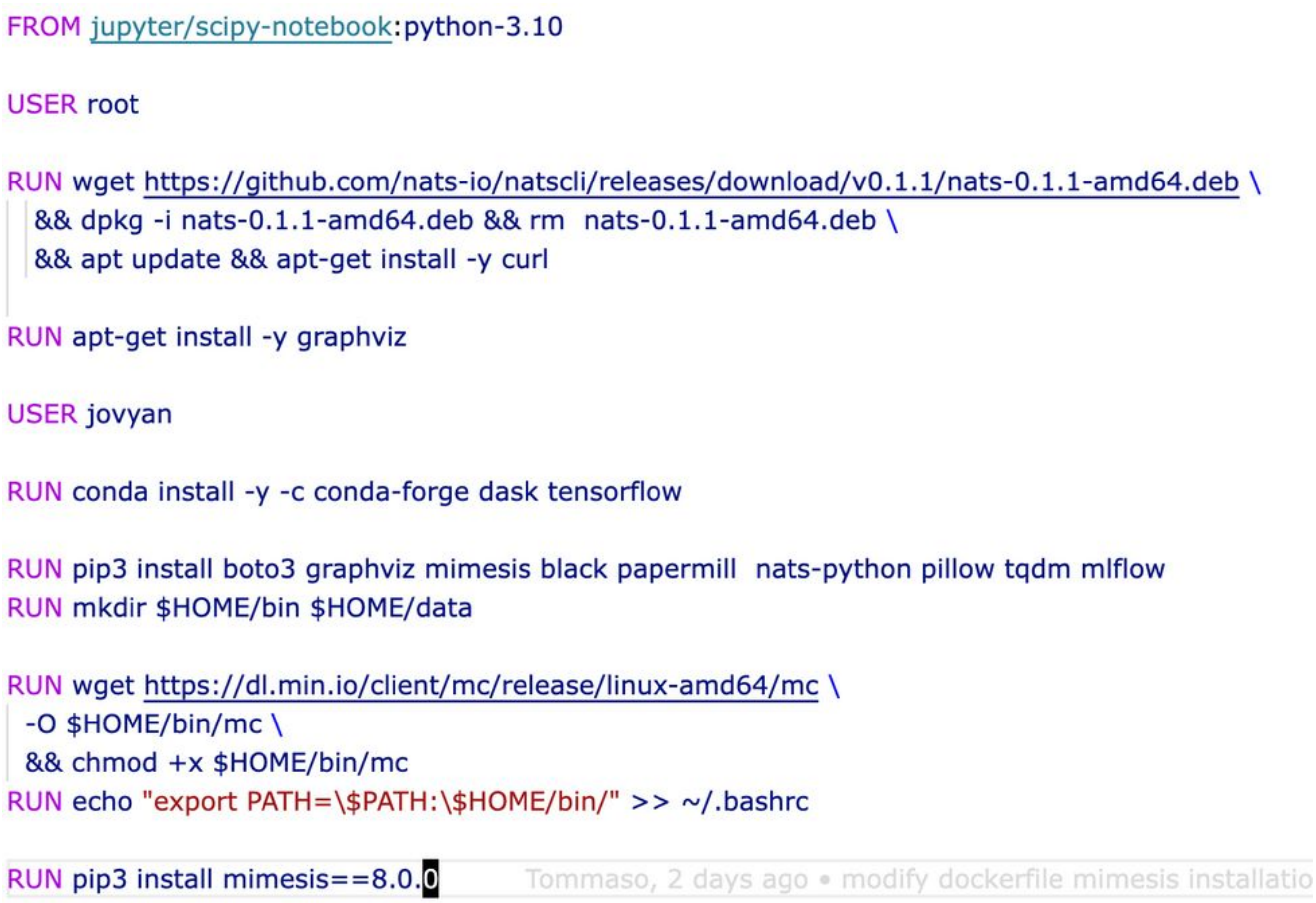
[Create your first dockerfile](#)

[“Advanced…”](#)

HANDS-ON TIME

Sneak peak to... the image we prepared for you

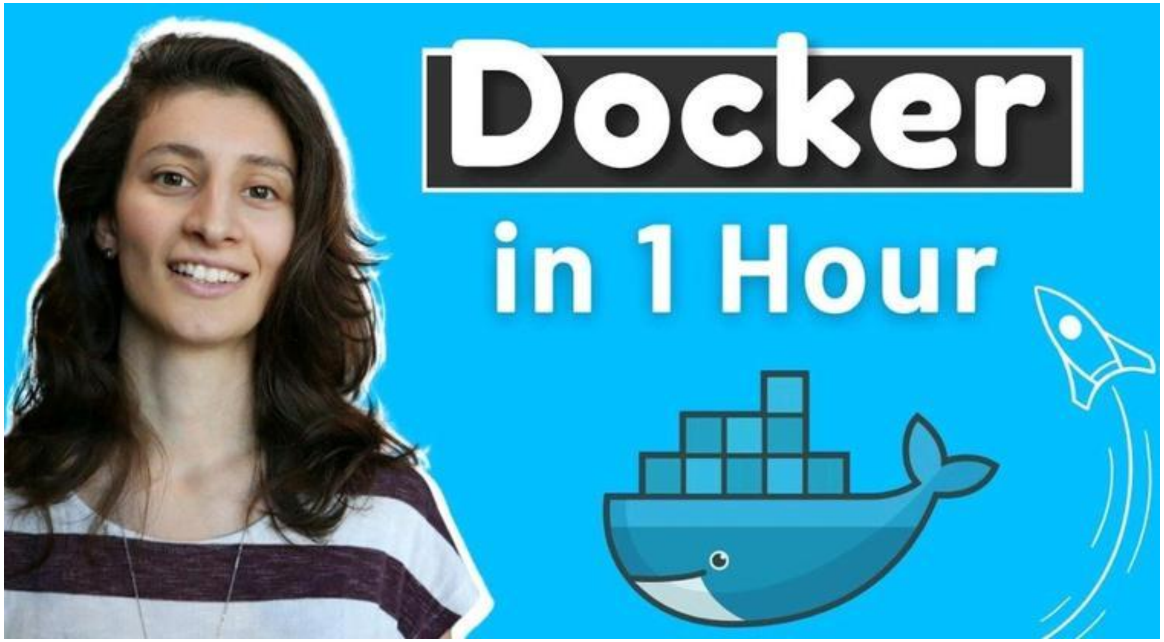
We customized an official image with interactive framework (JupyterLab) + all majors data science python packages



```
FROM jupyter/scipy-notebook:python-3.10

USER root

RUN wget https://github.com/nats-io/natscli/releases/download/v0.1.1/nats-0.1.1-amd64.deb \
    && dpkg -i nats-0.1.1-amd64.deb && rm  nats-0.1.1-amd64.deb \
    && apt update && apt-get install -y curl

RUN apt-get install -y graphviz

USER jovyan

RUN conda install -y -c conda-forge dask tensorflow

RUN pip3 install boto3 graphviz mimesis black papermill  nats-python pillow tqdm mlflow
RUN mkdir $HOME/bin $HOME/data

RUN wget https://dl.min.io/client/mc/release/linux-amd64/mc \
   -O $HOME/bin/mc \
   && chmod +x $HOME/bin/mc
RUN echo "export PATH=\$PATH:\$HOME/bin/" >> ~/.bashrc

RUN pip3 install mimesis==8.0.0        Tommaso, 2 days ago • modify dockerfile mimesis installatio
```

Suggested reference course

https://youtu.be/pg19Z8LL06w?si=W5QXikaYfqkFYjIf



But wait… not only Docker

Apptainer (formerly Singularity) is another container platform.

- it appears similar to Docker from a user perspective
- in the system's architecture, it is fundamentally different.

Apptainer/Singularity is particularly well-suited to running on distributed, High Performance Computing (HPC) infrastructure

Apptainer/Singularity assumes (more or less) that each application will have its own container.

Apptainer (vs Docker)

Docker shines for DevOPs teams providing cloud-native micro-services to users.

Singularity/Apptainer shines for scientific software running in an HPC environment. We will use it later in the week .

Apptainer/Singularity is a relatively new container software invented by Greg Kurtzer while at Lawrence Berkley National labs and now developed by his company Sylabs. It was developed with security, scientific software, and HPC systems in mind.

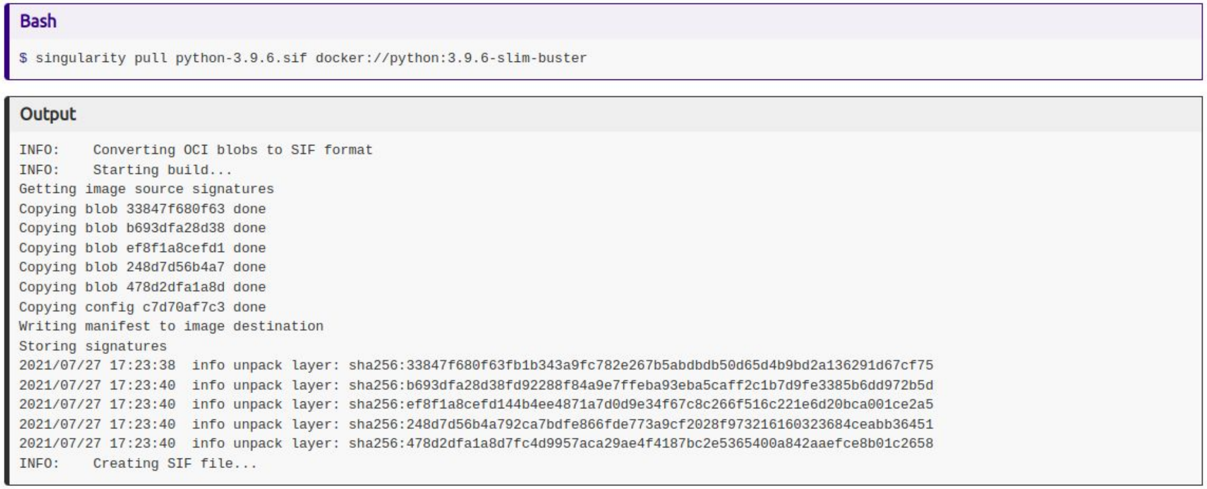
- Easy to learn and use (relatively speaking)
- Approved for HPC (installed on some of the biggest HPC systems in the world)
- **Can convert Docker containers to Singularity and run containers directly from Docker Hub**

Using Docker images with Singularity

Singularity can also start containers directly from Docker images, opening up access to a huge number of existing container images available on Docker Hub and other registries.

While Singularity doesn’t actually run a container using the Docker image (it first converts it to a format suitable for use by Singularity), the approach used provides a seamless experience for the end user.

- When you direct Singularity to run a container based on pull a Docker image, Singularity pulls the slices or layers that make up the Docker image and converts them into a single-file Singularity SIF image.



Bash

```
$ singularity pull python-3.9.6.sif docker://python:3.9.6-slim-buster
```

Output

```
INFO:    Converting OCI blobs to SIF format
INFO:    Starting build...
Getting image source signatures
Copying blob 33847f680f63 done
Copying blob b693dfa28d38 done
Copying blob ef8f1a8cefd1 done
Copying blob 248d7d56b4a7 done
Copying blob 478d2dfa1a8d done
Copying config c7d70af7c3 done
Writing manifest to image destination
Storing signatures
2021/07/27 17:23:38  info unpack layer: sha256:33847f680f63fb1b343a9fc782e267b5abdbdb50d65d4b9bd2a136291d67cf75
2021/07/27 17:23:40  info unpack layer: sha256:b693dfa28d38fd92288f84a9e7ffeba93eba5caff2c1b7d9fe3385b6dd972b5d
2021/07/27 17:23:40  info unpack layer: sha256:ef8f1a8cefd144b4ee4871a7d0d9e34f67c8c266f516c221e6d20bca001ce2a5
2021/07/27 17:23:40  info unpack layer: sha256:248d7d56b4a792ca7bdfe866fde773a9cf2028f973216160323684ceabb36451
2021/07/27 17:23:40  info unpack layer: sha256:478d2dfa1a8d7fc4d9957aca29ae4f4187bc2e5365400a842aaefce8b01c2658
INFO:    Creating SIF file...
```