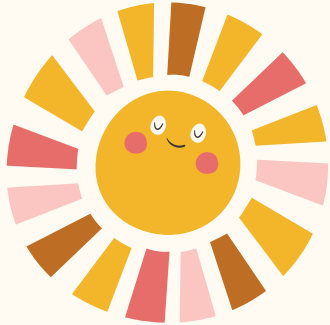


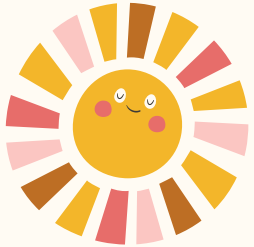
Container technologies

(Docker and Singularity)

by Weijia Luo

Table of contents

01



Why Container

Before Container

All the codes work now!

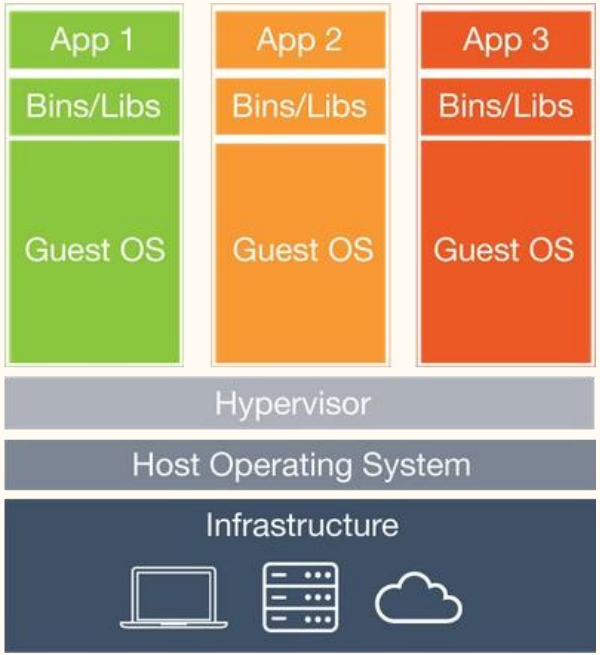
Developer

Oh no! Nothing works on my system!

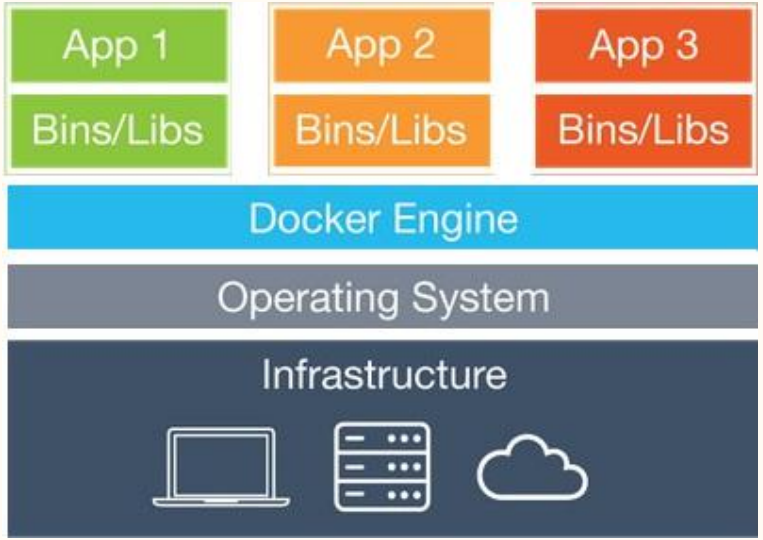
Tester

How to solve this problem?

VMs vs Containers



Virtual Machines



Containers

Why Container?

Criteria	Virtual Machine	Container (Docker)
OS Support	Occupies more space	Occupies less space
Boot-up Time	Long	Short
Performance	Unstable when running >1 VMs	Better (hosted in a single Docker engine)
Scaling	Difficult	Easy
Efficiency	Low	High
Portability	Low	High

After Container

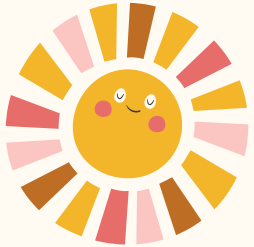
All the codes work now!

Developer

Yay! The code works on my system too!

Tester

02



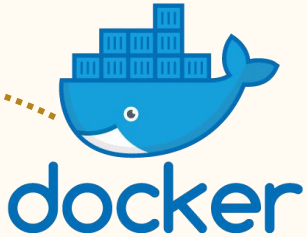
What is a Container

What is a Container

a **software package**

packages up **code** and all **dependencies**

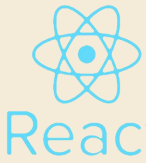
so the application runs **quickly and reliably** from one computing **environment** to another.



Example:

A simple web App

Code



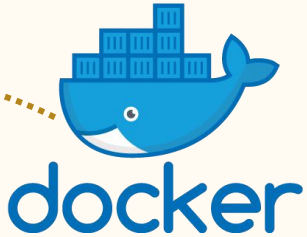
...

What is a Container

a **software package**

packages up **code** and all **dependencies**

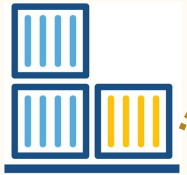
so the application runs **quickly and reliably** from one computing **environment** to another.



can run on the **same hardware**

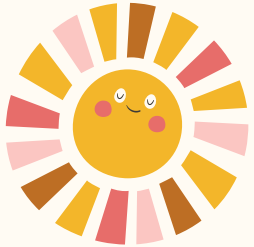
can **share the OS kernel** with other containers

each running as **isolated processes** in user space

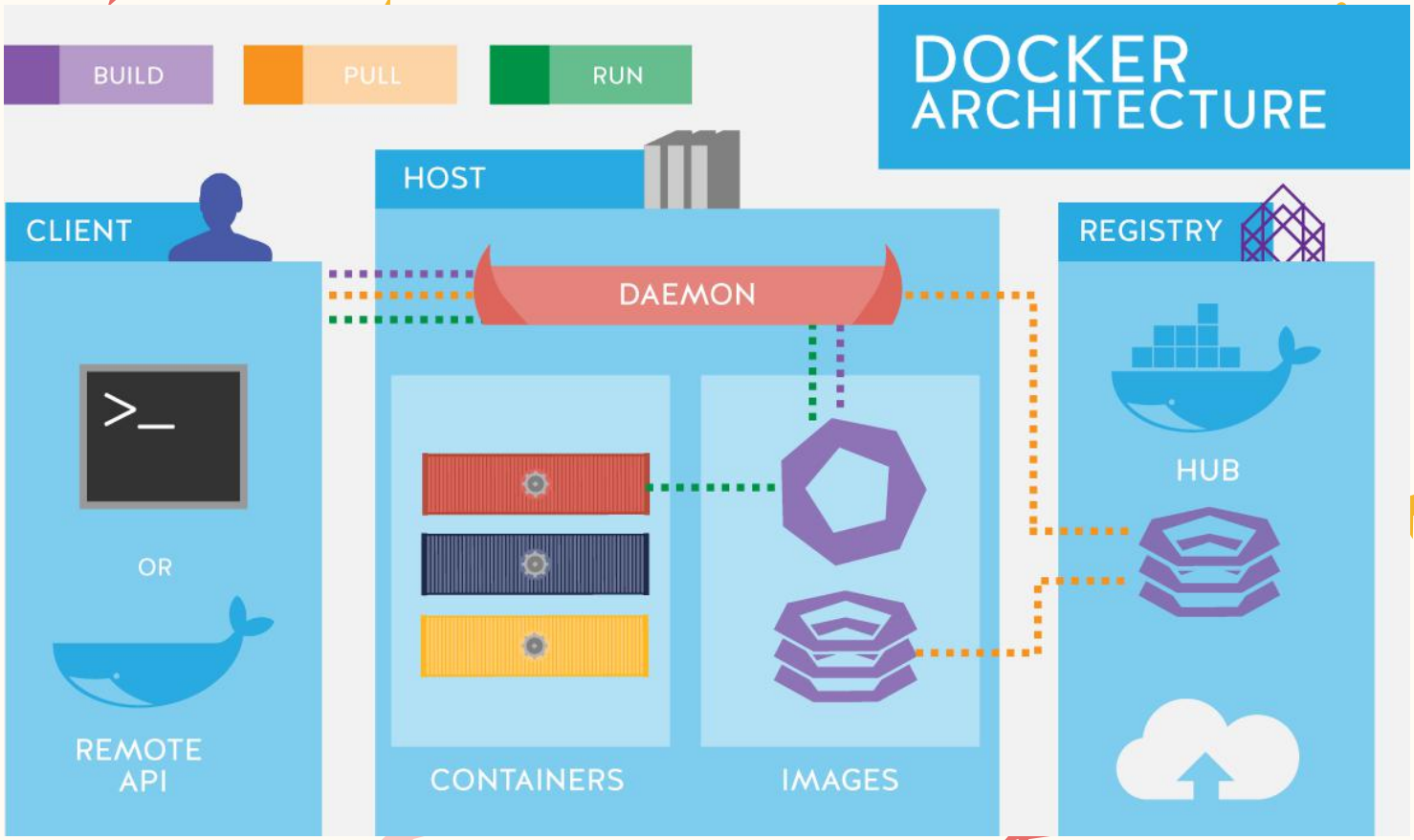


Multiple Containers

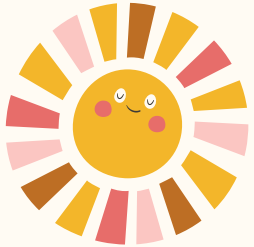
03



How does Docker work


BUILD
PULL
RUN
DOCKER ARCHITECTURE
HOST
CLIENT
REGISTRY
DAEMON
>_
OR
REMOTE API
CONTAINERS
IMAGES
HUB

04



Practical examples

Example 1: Run CentOS in Docker

1. Find the image on the registry. on docker hub or run “docker search centos”

2. Pull the image from the registry. docker pull centos

3. See images. docker images

4. Run centos image. docker run -it --name centos1 centos

```
(base) ivana@moodle:~$ sudo docker run -it --name centos1 centos
[root@da255d7c9175 /]# ls
bin  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@da255d7c9175 /]# cd home
[root@da255d7c9175 home]# ls
[root@da255d7c9175 home]# exit
exit
```

Example 1: Run CentOS in Docker

5. See all containers. docker ps -a

6. Start container centos1. docker start da255d7c9175

7. Run the container in terminal. docker exec -it da255d7c9175 bash

8. Stop the container. docker stop da255d7c9175

Example 2: build an image from our project

1. Create a project.
2. Maven clean and package.
3. Copy the .jar file under the foler of the project.
4. Create Dockerfile.
5. Build an image.

```
docker build -t weijias-demo .
```

6. See images.

```
docker images
```

Example 2: build an image from our project

7. Run the image in a container.

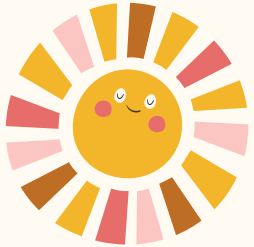
```
docker run -d -p 8080:8080 --name
demo1 weijias-demo
```

8. Test the project.

8.1. in the docker container: curl localhost:8080/hello

8.2. in the browser on host machine: localhost:8080/hello

05



Singularity

Why Singularity?

Singularity addresses the need for containers in the HPC community.

Advantages compared to other container engines

Seamless integration into HPC ecosystem. E.g. workload managers like slurm

Better security model, privilege seperation, tools to define and control access rights

Standardized image format, guarantees authenticity and immutability

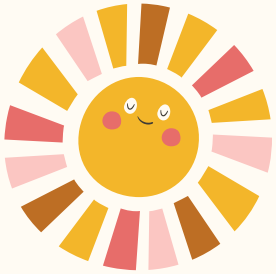
By default support for important libraries and software. E.g. OpenMPI

Better support for older RHEL systems

Easier to setup specialized hardware, including CUDA GPUs

Thanks!

Do you have any questions?



Resources

https://www.tukuppt.com/muban/oakrpxaa.html
https://www.iconfinder.com/icons/1398912/circle_correct_mark_success_tick_yes_check_icon
http://616pic.com/sucai/1m9i792x2.html
https://dlpng.com/png/6502799
https://storage.googleapis.com/xebia-blog/1/2016/11/Container-vs-VMs.jpg
https://www.vectorstock.com/royalty-free-vector/container-logistic-logo-icon-design-vector-22461709
https://www.clipartmax.com/middle/m2i8d3H7b1K9b1H7_logo-logo-docker/
https://www.logo.wine/logo/Spring_Framework
https://www.thecuriousdev.org/new-in-spring-boot-2/spring-boot-logo/
https://logos-download.com/32169-apache-tomcat-logo-download.html
https://de.wikipedia.org/wiki/Datei:Java-Logo.svg