Docker 101

Laurent Magnin
UQÀM, MGL7320

Introduction

UQÀM

Hello World *in Python*

Do you want to `print("Hello World!")`?

To do so, you have to have access to a Python interpreter…

- Install a version on your PC? Or through a Web interface?
- Wrong version? Conflicting versions?
- Missing libraries?
- What about also using another language?
- What about OS specificities [“/” (Linux) vs. “\” (Windows)]?
- …

UQÀM

With Docker...

```
$ docker run -it python python
Python 3.8.5 (default, Jul 22 2020, 12:28:11)
[GCC 8.3.0] on linux
>>> print(" Hello World!")
This line will be printed.
>>> quit()
```


UQÀM

... and even more!

To call (in parallel) another version of Python:

```
$ docker run -it python:2.7.7 python
```

To call Python with the SciPy library:

```
$ docker run -it alectolytic/scipy python
```

To compile Java Code:

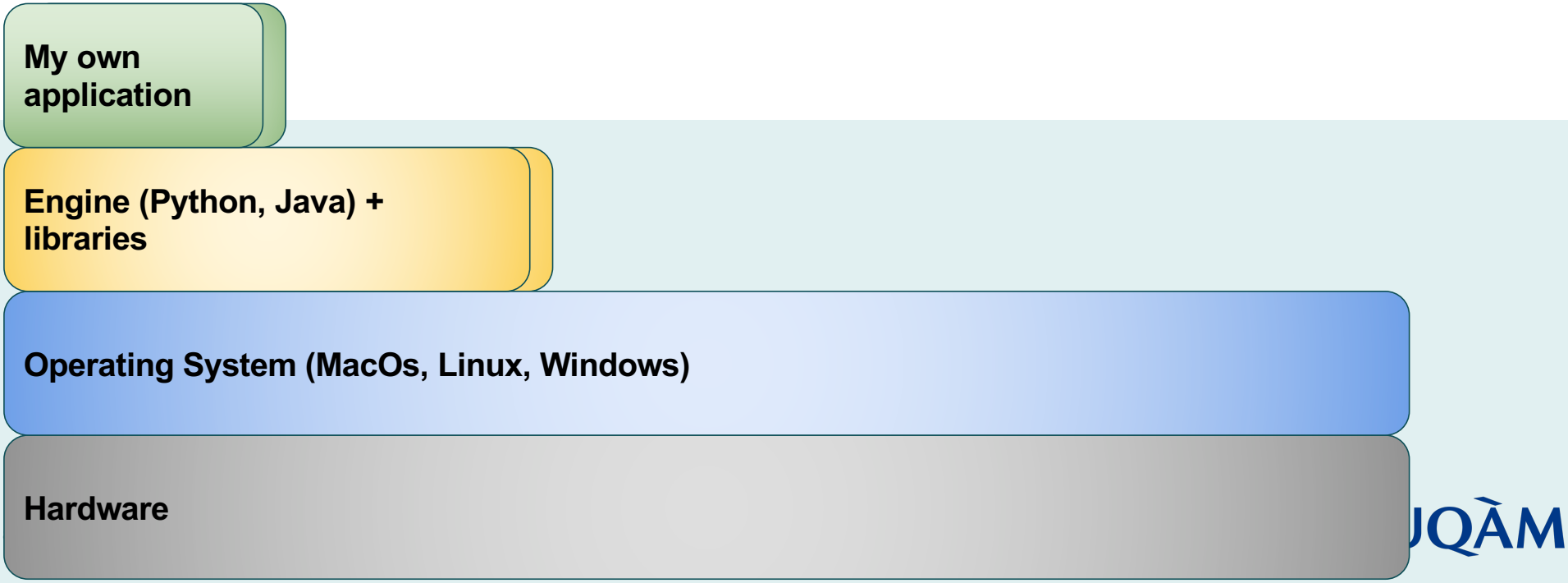
```
$ docker run --rm -v "$PWD":/usr/src/myapp -w /usr/src/myapp openjdk:7
javac Main.java
```


UQÀM

Ok, but how does it work?

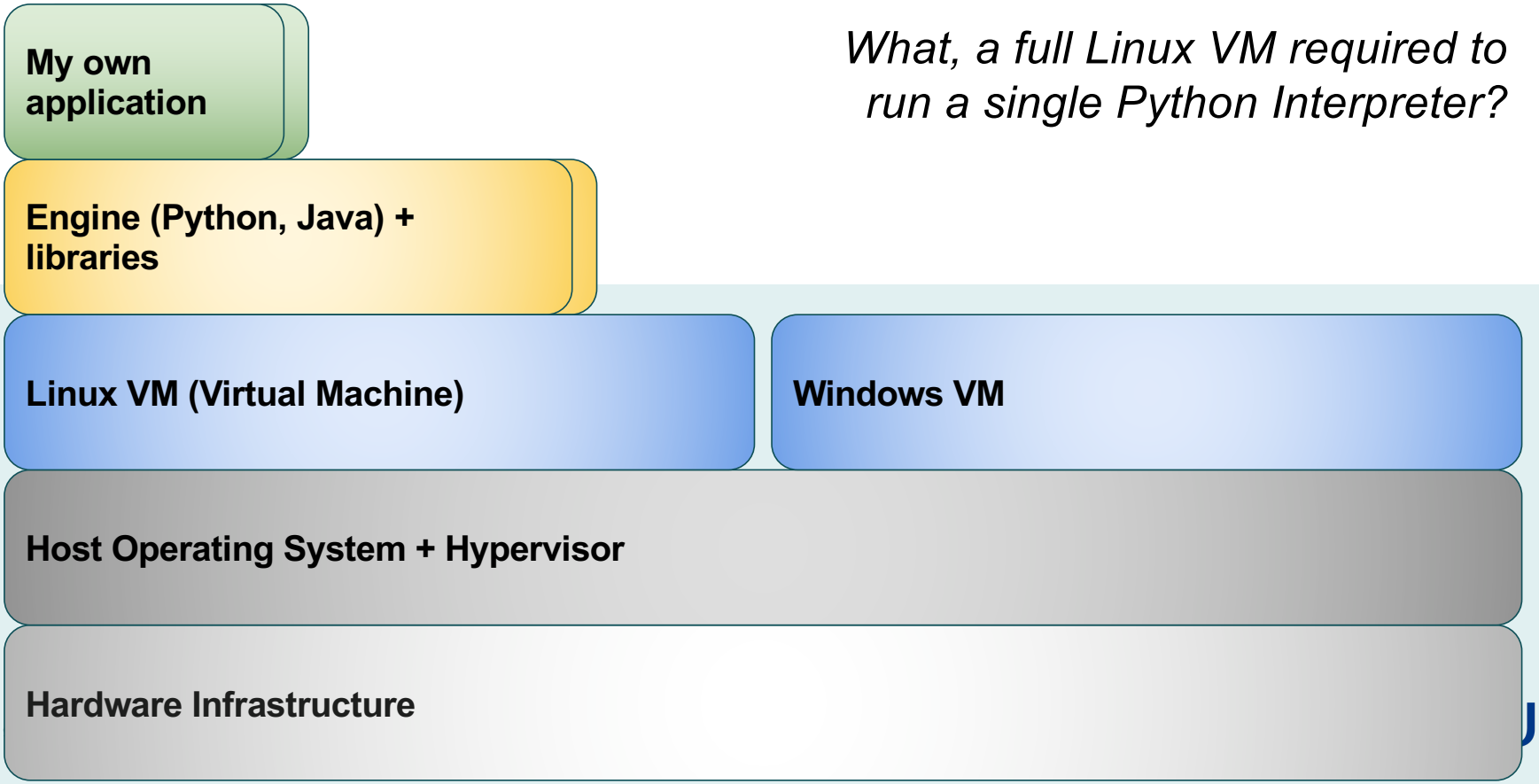
UQÀM

Let’s start with the (simplified) basics



UQÀM

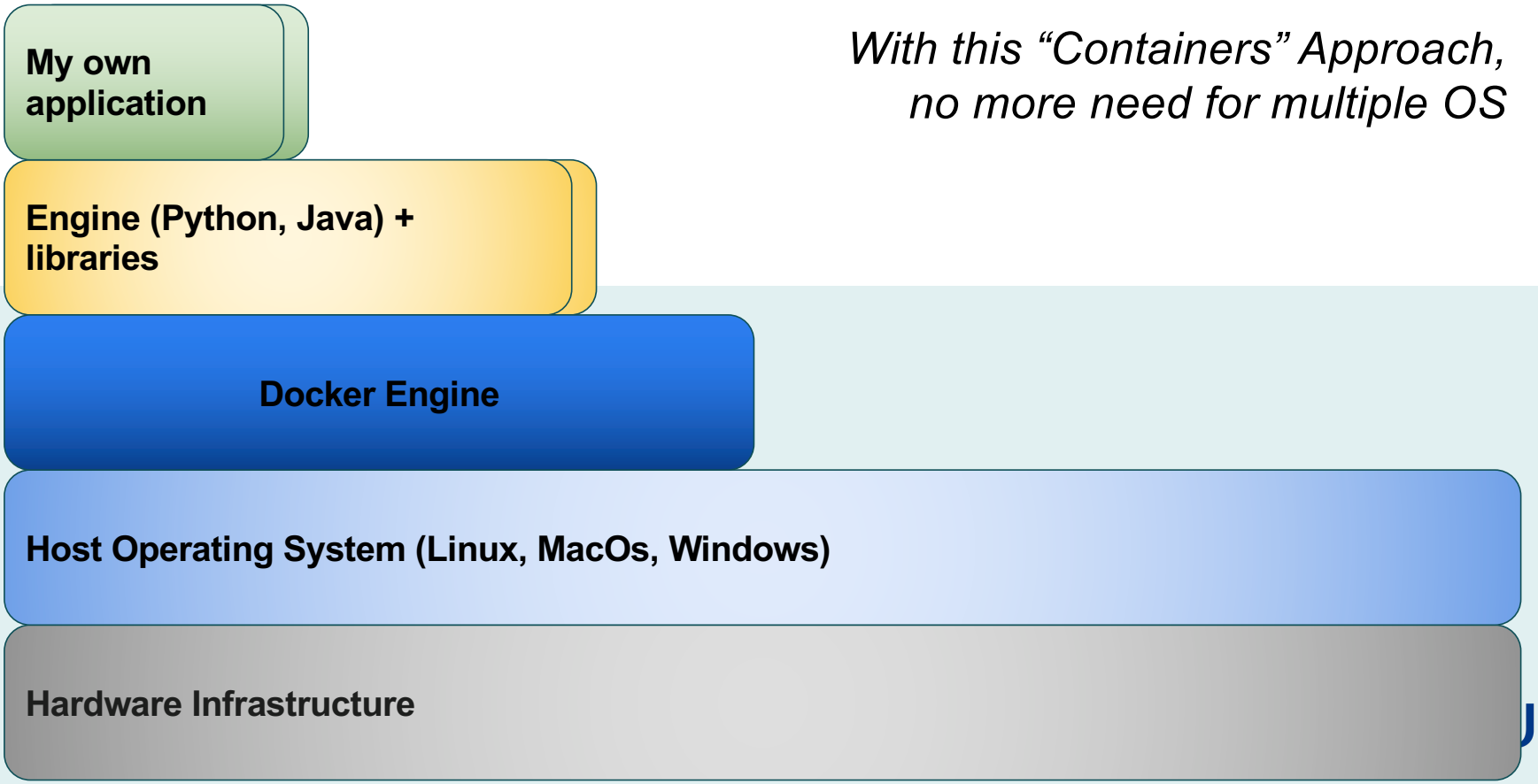
The Virtual Machine Approach



What, a full Linux VM required to run a single Python Interpreter?

UQÀM

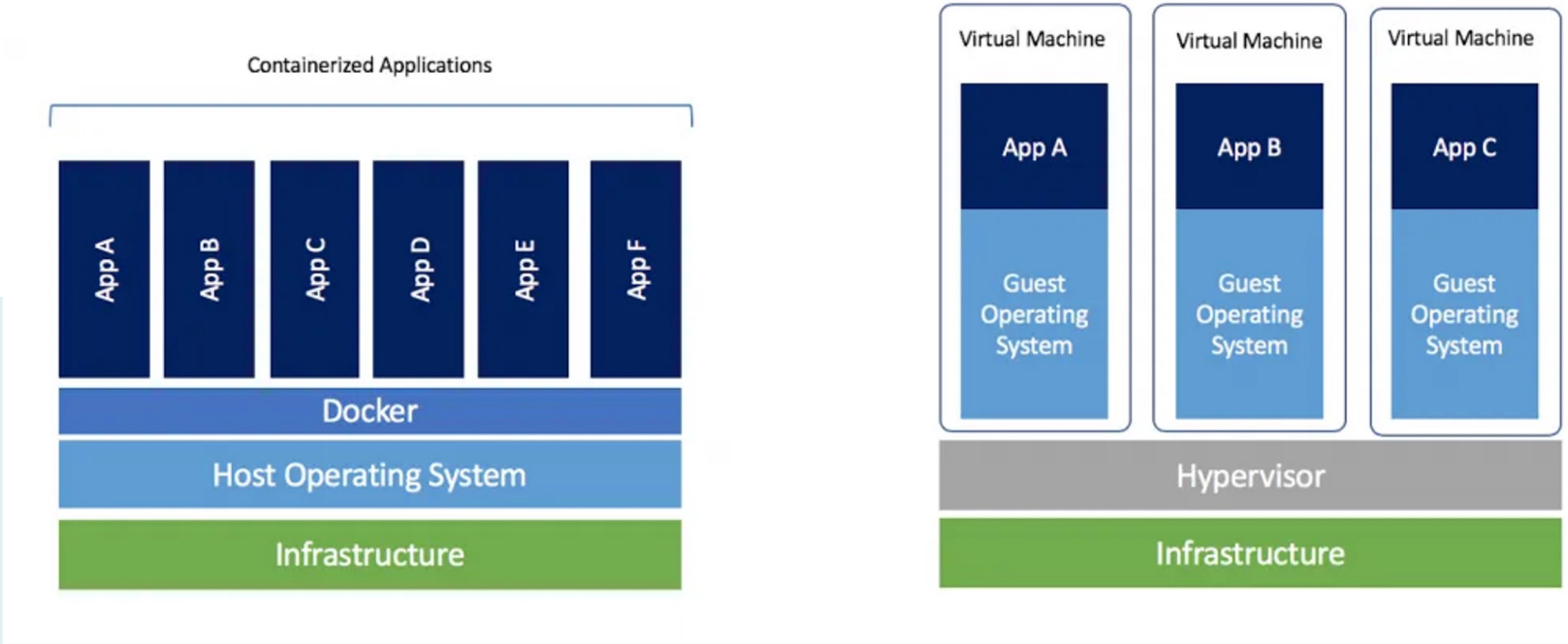
The Containers Approach



With this "Containers" Approach, no more need for multiple OS

UQÀM

The Virtual Machine Approach



https://www.docker.com/blog/containers-replacing-virtual-machines/

UQÀM

Virtual Machine vs. Container

A VM is required when:

- A Graphical User Interface (GUI) is needed
- To run “classic” heavy applications

Containers are perfectly suitable when:

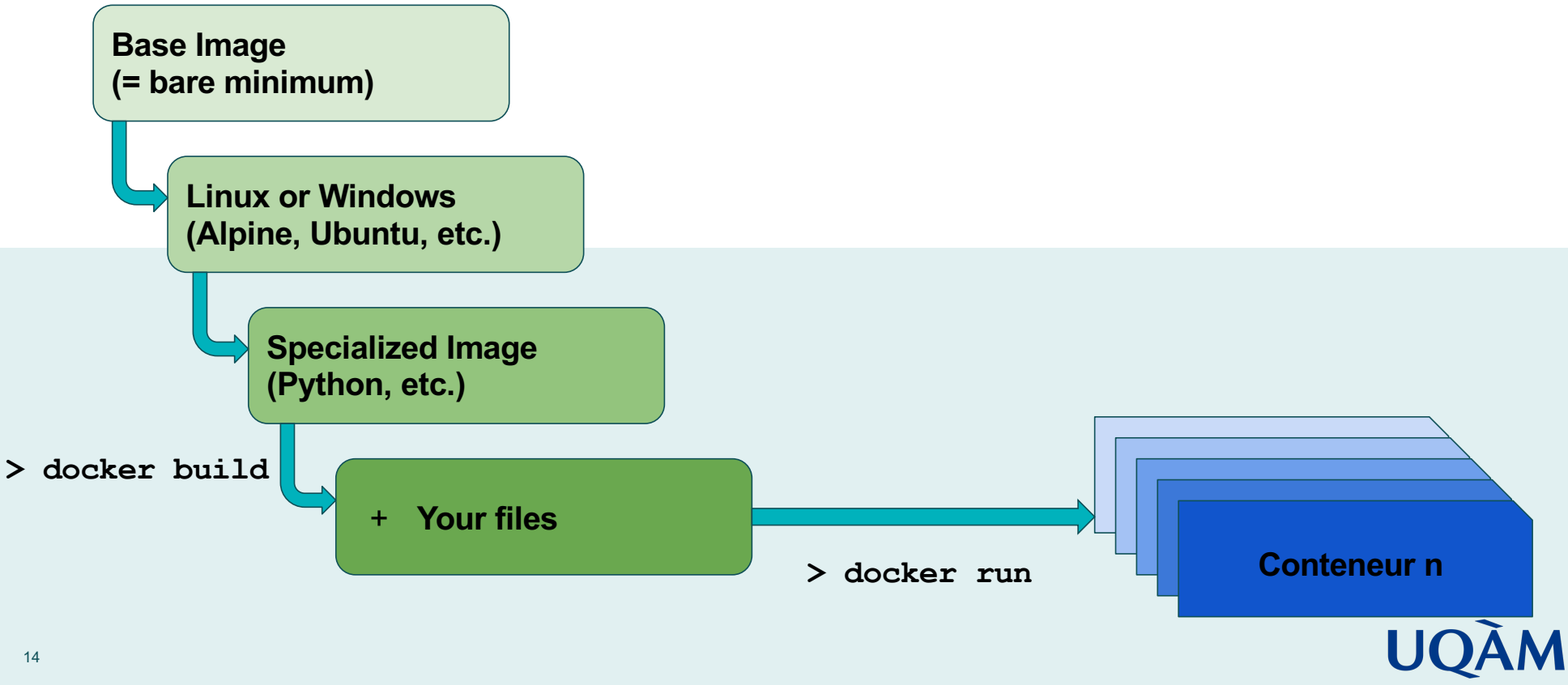
- Your “application” is performing a single task (one process)
- Your application is reached through
  - a command line
  - an url (REST, Web, etc.)

UQÀM

Docker’s life cycle

UQÀM

From Images to Containers



UQÀM

“Dockerfile”, the root of the Docker Universe

```
FROM python:3.8

RUN apt-get install -yqq unzip

RUN mkdir /usr/script
WORKDIR /usr/script
COPY ./scripts .

CMD ["sh", "-c", "python3 myapp.py $ENV_VAR"]
```


UQÀM

You’re now in command!

```
$ ls
Dockerfile      scripts

$ docker build -t myapp:test .

$ docker images
REPOSITORY  TAG          IMAGE ID          CREATED             SIZE
python      latest       4e2d08f34f6d      4 days ago          934MB
myapp       test         919ab681963b      13 days ago         1.63GB

$ docker run myapp:test
```


UQÀM

You’re now in command! (Cont’d)

```
$ docker images
REPOSITORY  TAG       IMAGE ID       CREATED          SIZE
myapp       test      919ab681963b   3 days ago       934MB

$ docker container ls

$ docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

$ docker rmi [OPTIONS] IMAGE [IMAGE...]
$ docker rm [OPTIONS] CONTAINER [CONTAINER...]

$ docker --help
```


UQAM

Docker Compose (multiple containers)

From a `docker-compose.yml` file, such as

```
version: '2.0'
services:
  web:
    build: .
    ports:
    - "5000:5000"
    volumes:
    - .:/code
    - logvolume01:/var/log
    links:
    - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```


UQÀM

Docker Compose

It becomes possible to start together a set of related and interconnected Containers:

```
$ docker-compose up –d

$ docker container ls

$ docker-compose down
```


UQÀM

How to Install and/or call Docker?

UQÀM

To Install Docker *locally*

UQÀM

Docker Desktop (“Dev” mode)

https://docs.docker.com/engine/install/

Docker Desktop for Mac

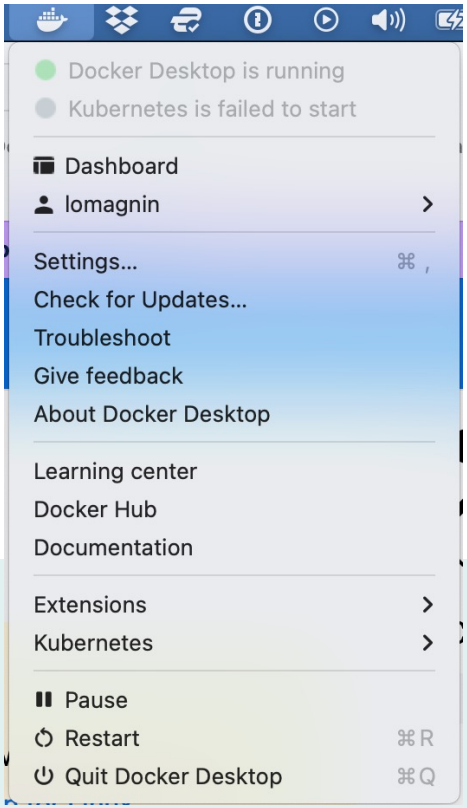
A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.

Docker Desktop for Windows

A native Windows application which delivers all Docker tools to your Windows computer.

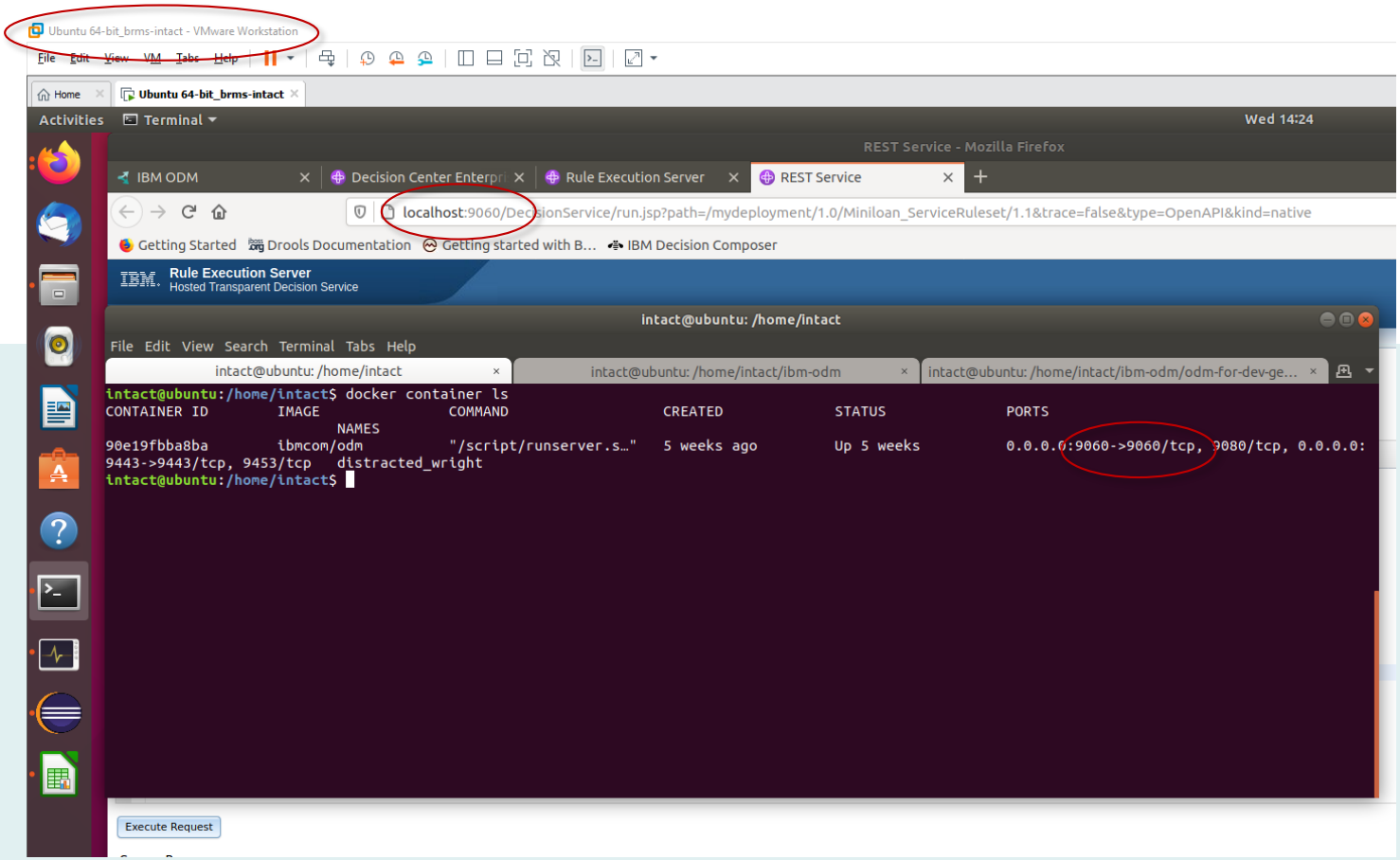
Docker for Linux

Install Docker on a computer which already has a Linux distribution installed.



UQÀM

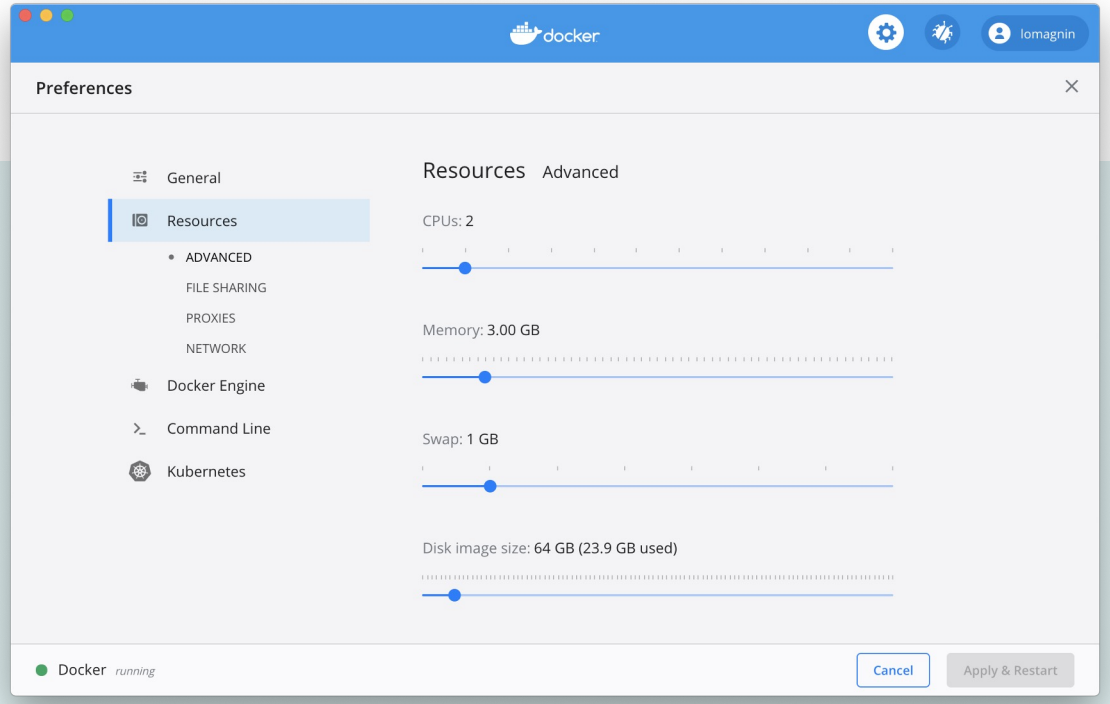
Running Docker inside a VM...



UQÀM

Is that fundamentally different?

Actually, Docker Desktop *on Windows (or MacOS)* runs inside a dedicated Linux based Virtual Machine with preemptive resources…



UQÀM

To run Docker
on a K8s Cluster

UQÀM

Kubernetes, *the* Containers Orchestrator

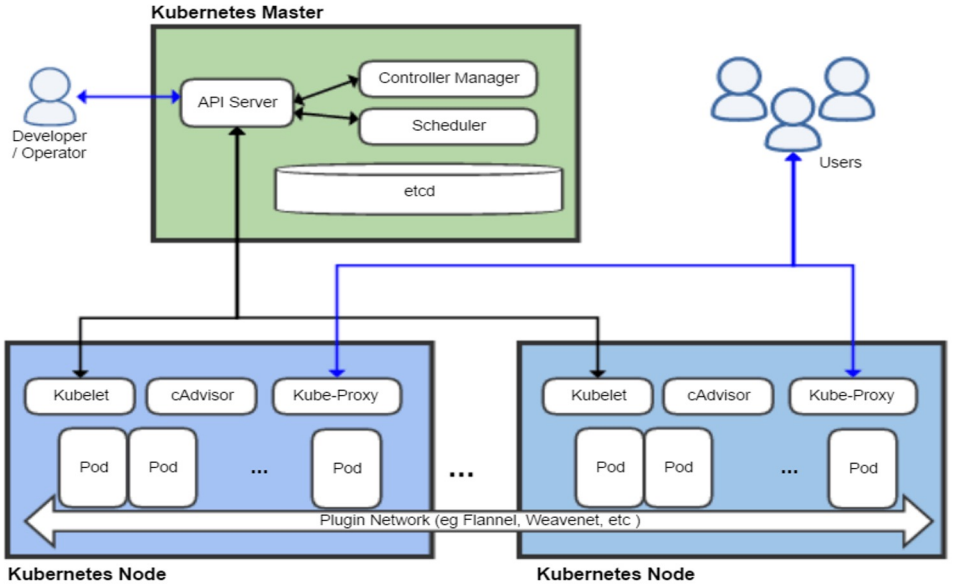
Kubernetes (commonly stylized as **k8s**) is an open-source container-orchestration system for automating computer application deployment, scaling, and management.

It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation. It aims to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts". It works with a range of container tools, including Docker.

Many cloud services offer a Kubernetes-based platform or infrastructure as a service (PaaS or IaaS) on which Kubernetes can be deployed as a platform-providing service. Many vendors also provide their own branded Kubernetes distributions.

UQÀM

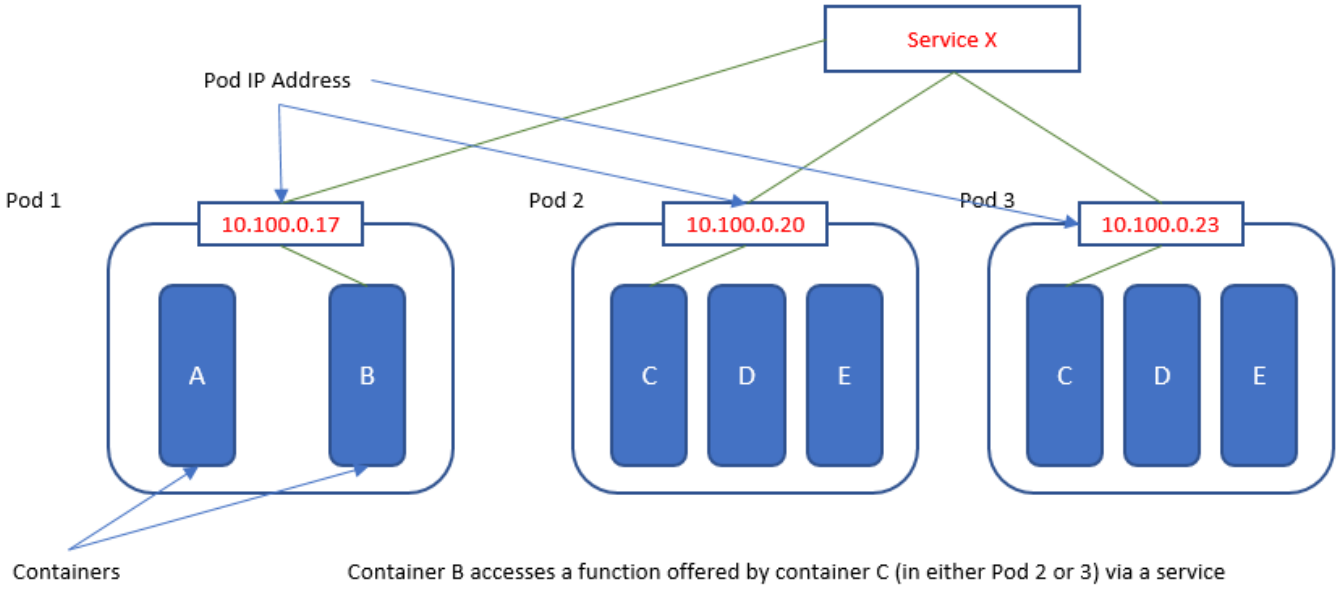
K8s, a complex machinery...



By Khtan66 —Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=53571935

UQÀM

K8s, *containers, pods & services*



By Marvin The Paranoid - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=75140812

UQÀM

Helm, *the K8s Chart Template*

```
mychart/
 Chart.yaml
 values.yaml
 charts/
 templates/
 ...
```

```yaml
apiVersion: v1
 kind: ConfigMap
metadata:
 name: {{ .Release.Name }}-configmap
data:
 myvalue: "Hello World"
```

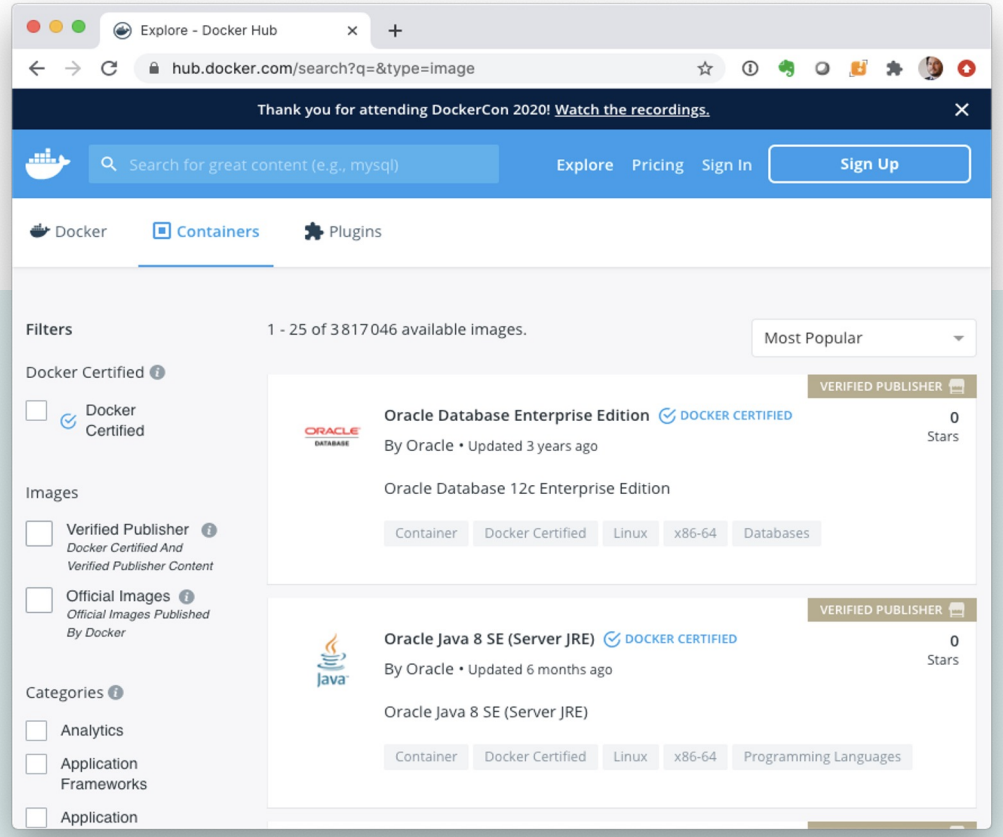
mychart/templates/configmap.yaml

UQÀM

The right Docker Image...

UQÀM

Docker Hub, *the* Docker Images Registry

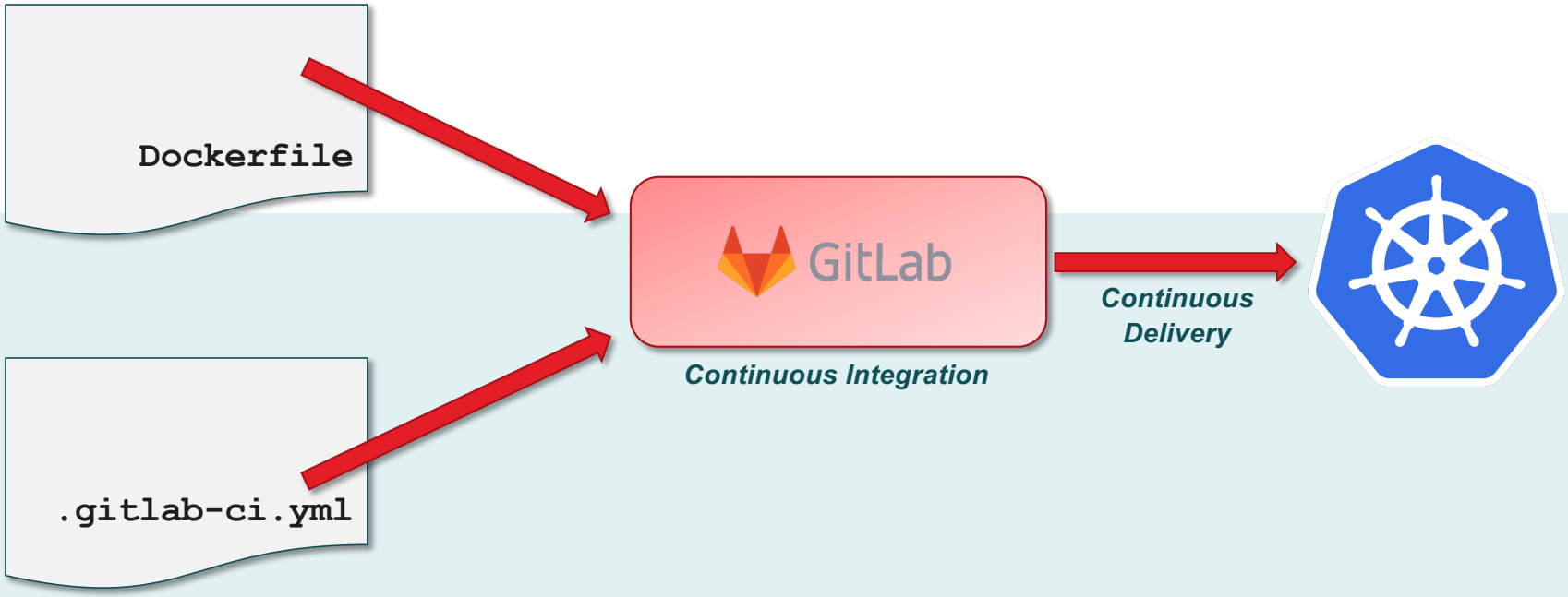


UQÀM

Deploying Docker Containers...

UQÀM

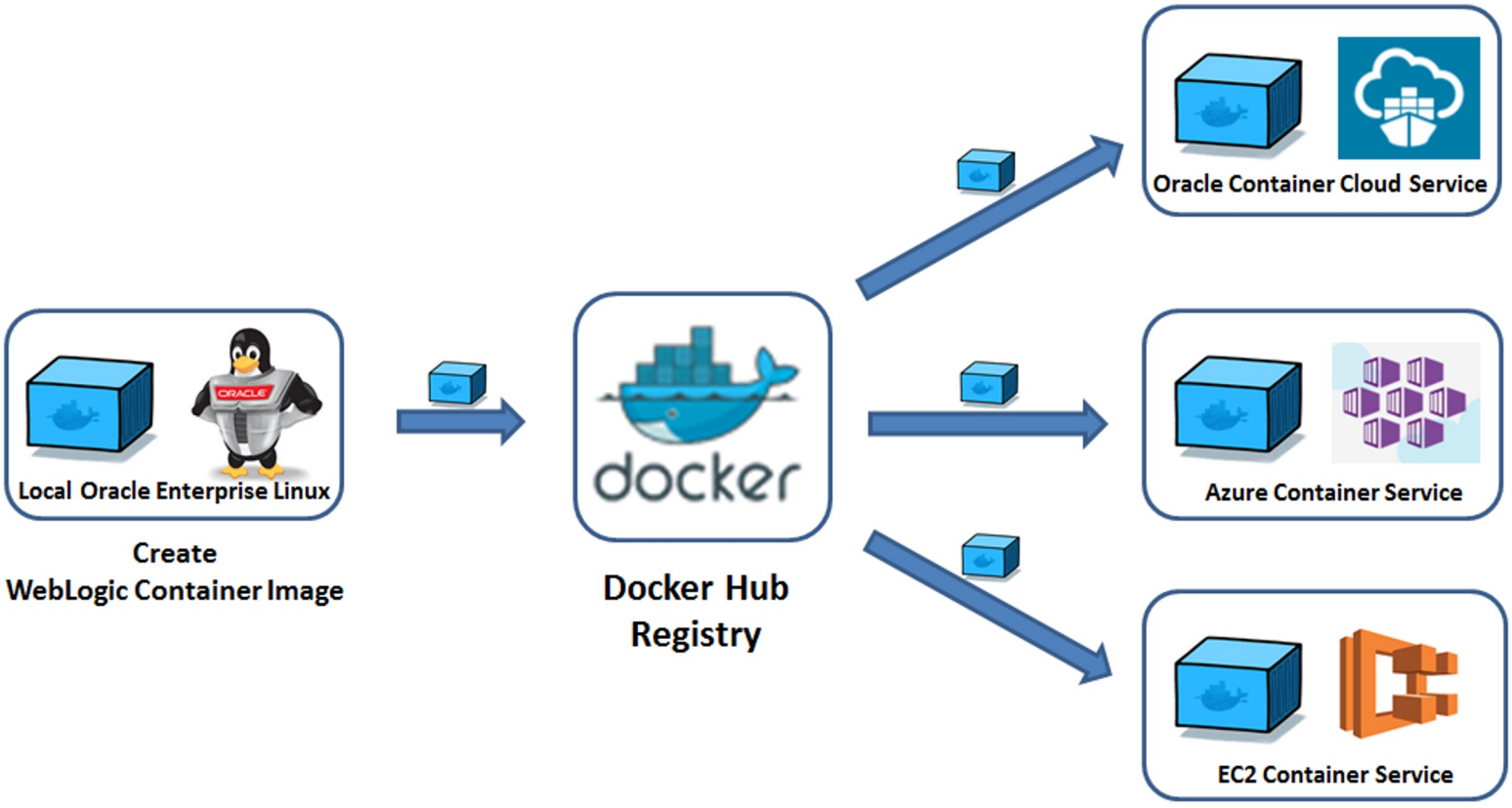
Gitlab CI/CD Pipeline is your friend!



https://confluence.tooling.intactfc.cloud/display/DVE/GitLab+and+Pipeline+Creation+Notes

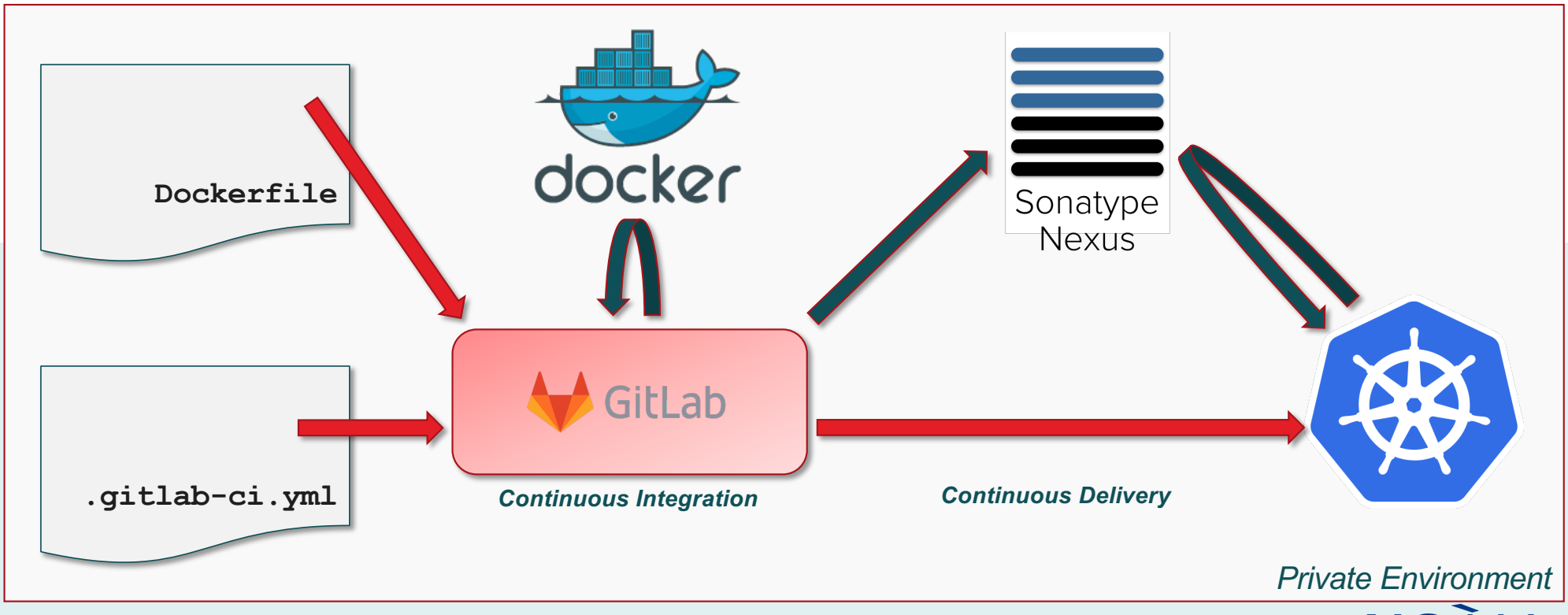
UQÀM

The Docker Images Hub



UQÀM

Nexus, *a typical* Docker Registry



UQÀM

Questions?

UQÀM