Introduction à Docker Kubernetes

Laurent Grateau / Nicolas Peulvast

Agenda

- Presentation
- Introduction
- La containerisation (30 min)
- Pause (15min)
- Premiere partie - Travaux Pratiques (30 min)
- Seconde partie - Travaux Pratiques (15 min)
- Troisième partie - Travaux Pratiques (20 min)
- Kubernetes (15 min)
- Conclusion

Accurate as of Sept 15, 2017

Introduction

IBM France Lab :

- R&D IBM en France à Saclay & Sophia Antipolis
- 350 développeurs
- Spécialisé dans l'automatisation de la décision et l'Intelligence Artificielle pour les entreprises

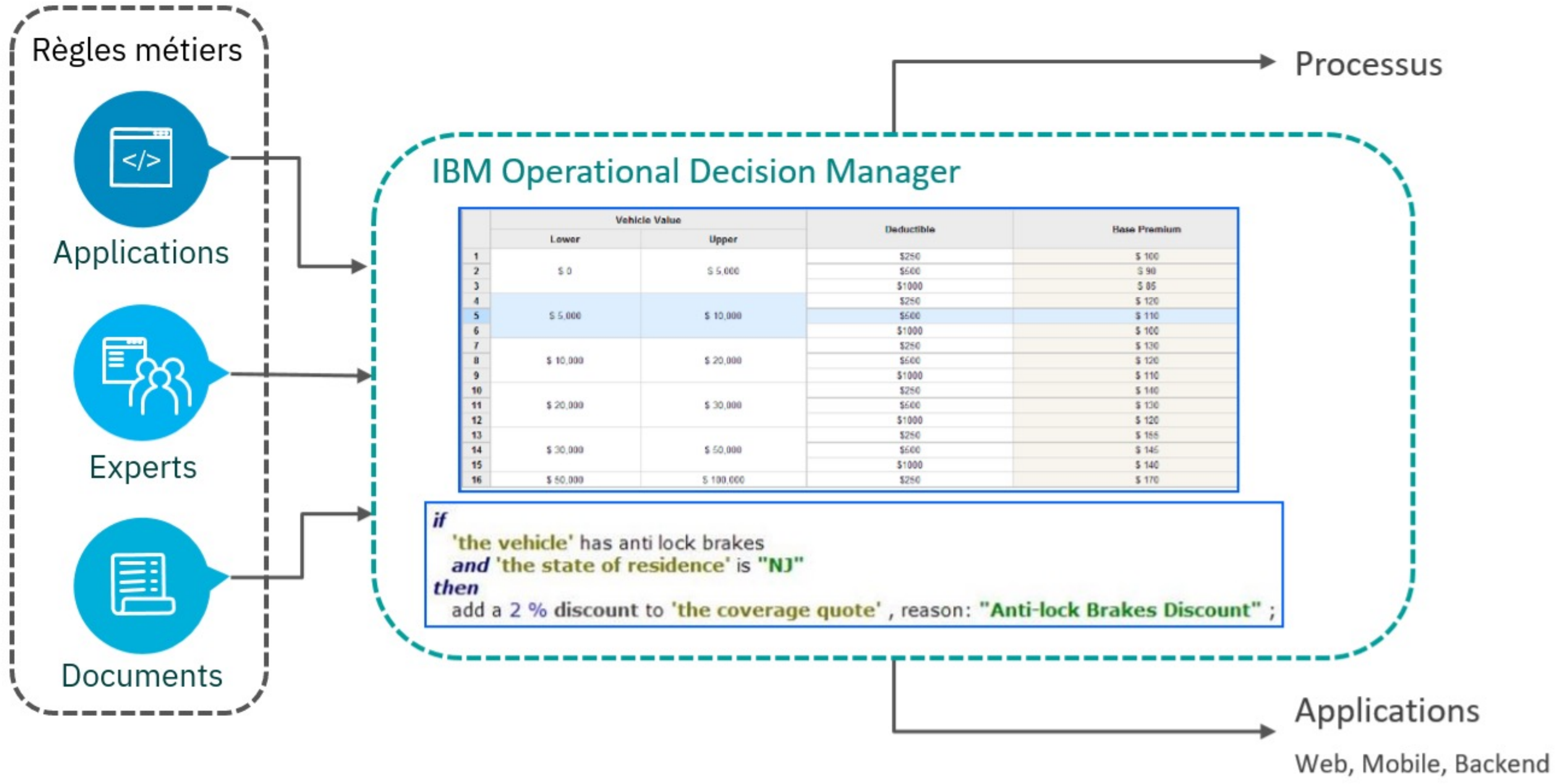
Tech Lead Cloud
Senior Software Dev

Laurent Grateau
laurent.grateau@fr.ibm.com

Performance Architect
Senior Software Dev

Nicolas Peulvast
nicolas.peulvast@fr.ibm.com

Qu'est-ce que ODM ?



	Vehicle Value		Deductible	Base Premium
	Lower	Upper		
1	$ 0	$ 5,000	$250	$ 100
2			$500	$ 90
3			$1000	$ 85
4	$ 5,000	$ 10,000	$250	$ 120
5			$500	$ 110
6			$1000	$ 100
7	$ 10,000	$ 20,000	$250	$ 130
8			$500	$ 120
9			$1000	$ 110
10	$ 20,000	$ 30,000	$250	$ 140
11			$500	$ 130
12			$1000	$ 120
13	$ 30,000	$ 50,000	$250	$ 155
14			$500	$ 145
15			$1000	$ 140
16	$ 50,000	$ 100,000	$250	$ 170

```
if
  'the vehicle' has anti lock brakes
  and 'the state of residence' is "NJ"
then
  add a 2 % discount to 'the coverage quote' , reason: "Anti-lock Brakes Discount" ;
```

Cas d'usage :

- Approbation de prêt
- Détection de fraudes
- Traitement des demandes de remboursements
- Recommandations d’actions client dans le respect d’une stratégie d’entreprise

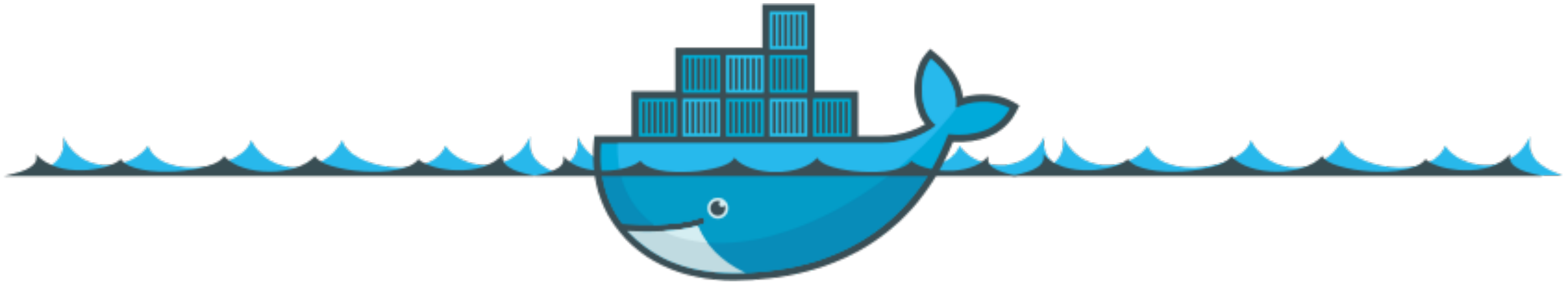
La containerisation

docker

kubernetes

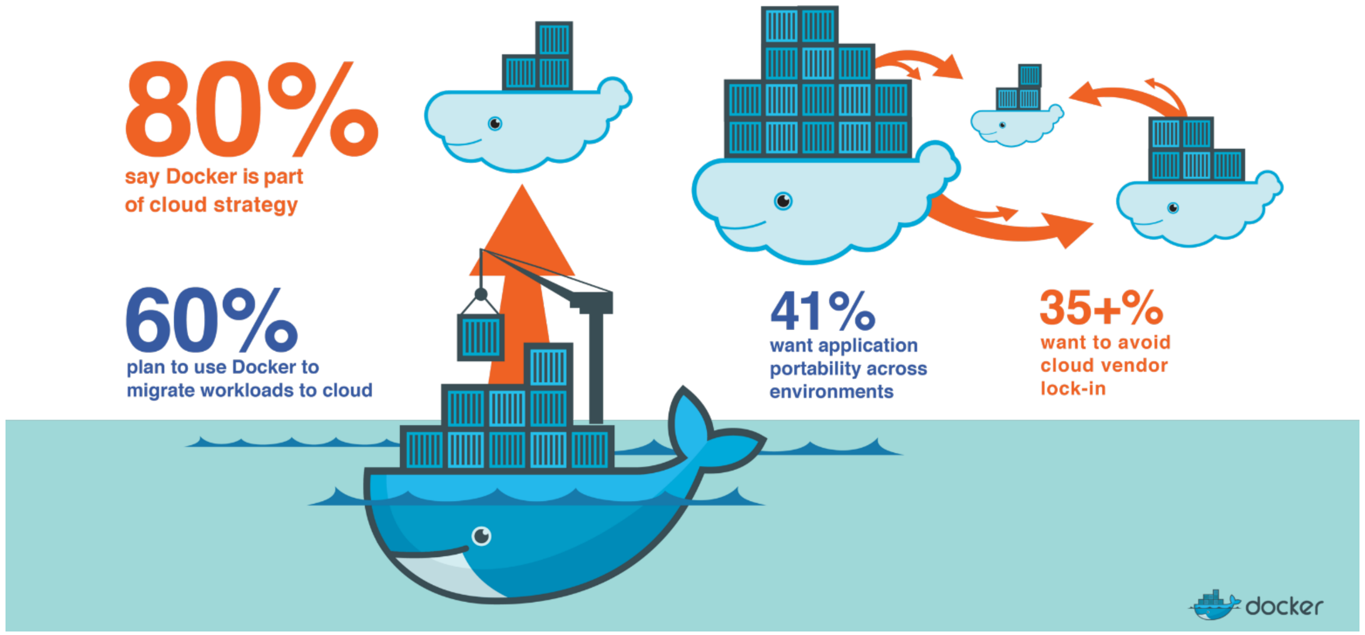
Why Docker ?

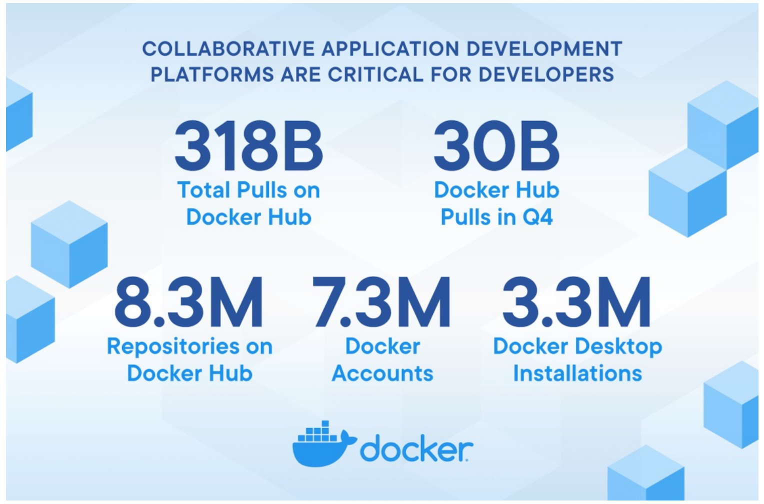
- **Docker** is a container-based technology.
- It automates the deployment of applications inside software containers.
- It provides an additional layer of abstraction and automatization of operating system–level virtualization on Linux.
- It is an open platform for developers and system admins to build, ship, and run distributed applications.



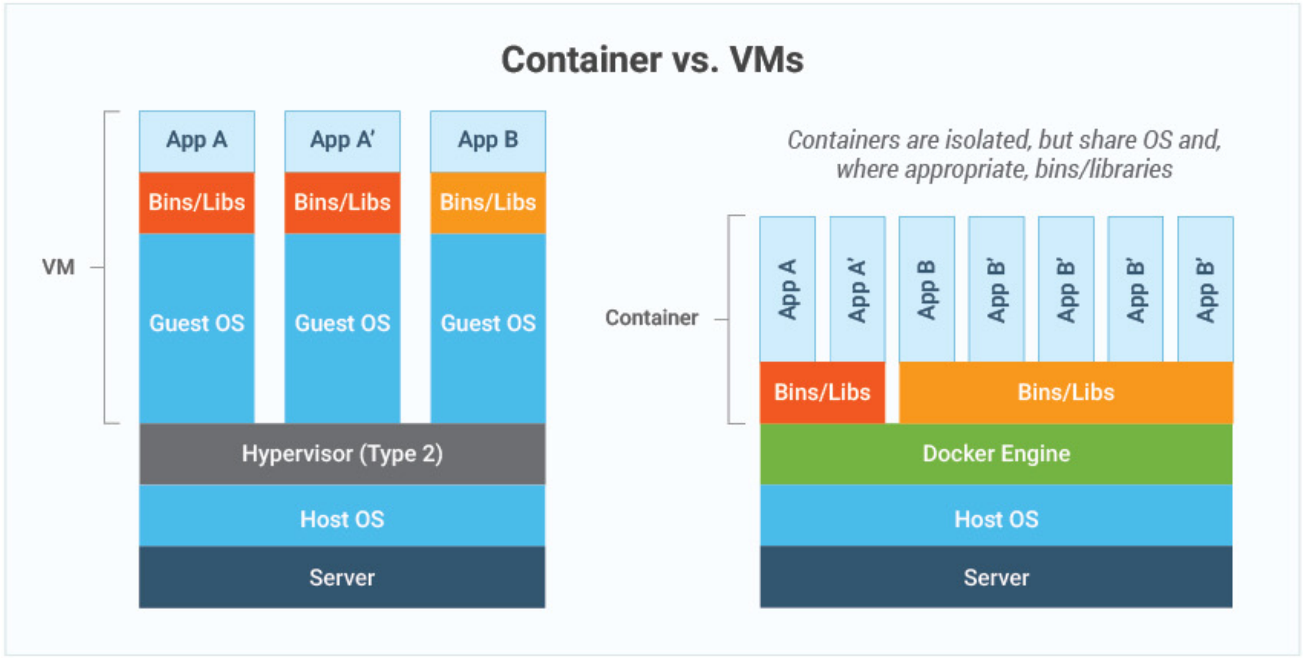
Why Docker ?

- Build any app in any language using any stack
- Dockerized apps can run anywhere on anything
- On premises & on multiple vendor clouds
- Unites Developers and Sys Admins in the fight against those pesky dependency demons!
- 15% of hosts running Docker and increasing.
- Central form factor for a majority of public and private clouds


80%
say Docker is part of cloud strategy
60%
plan to use Docker to migrate workloads to cloud
41%
want application portability across environments
35+%
want to avoid cloud vendor lock-in
docker


COLLABORATIVE APPLICATION DEVELOPMENT PLATFORMS ARE CRITICAL FOR DEVELOPERS
318B
Total Pulls on Docker Hub
30B
Docker Hub Pulls in Q4
8.3M
Repositories on Docker Hub
7.3M
Docker Accounts
3.3M
Docker Desktop Installations
docker

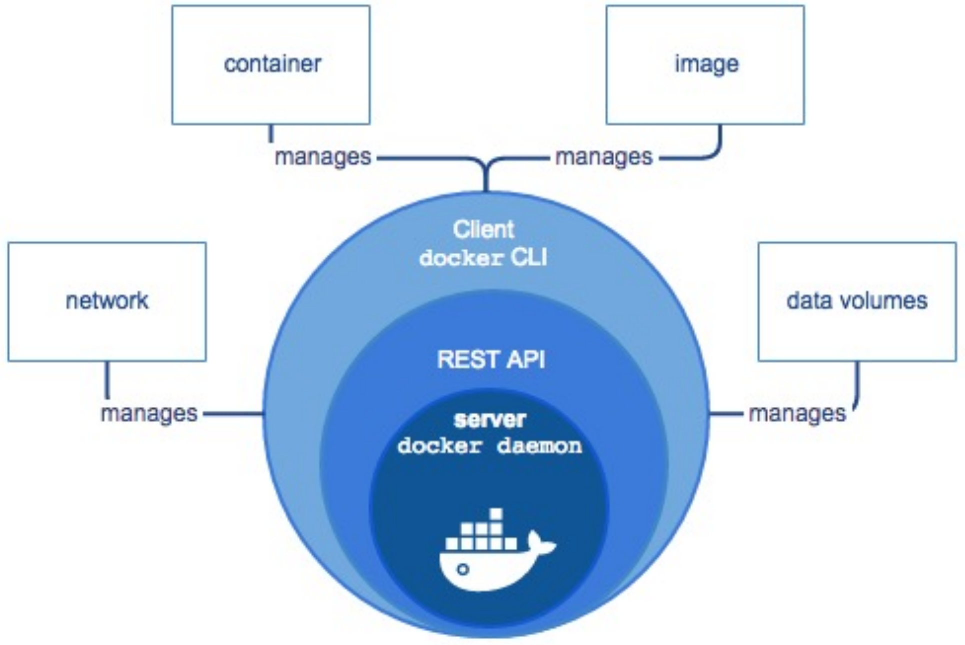
Container vs VM

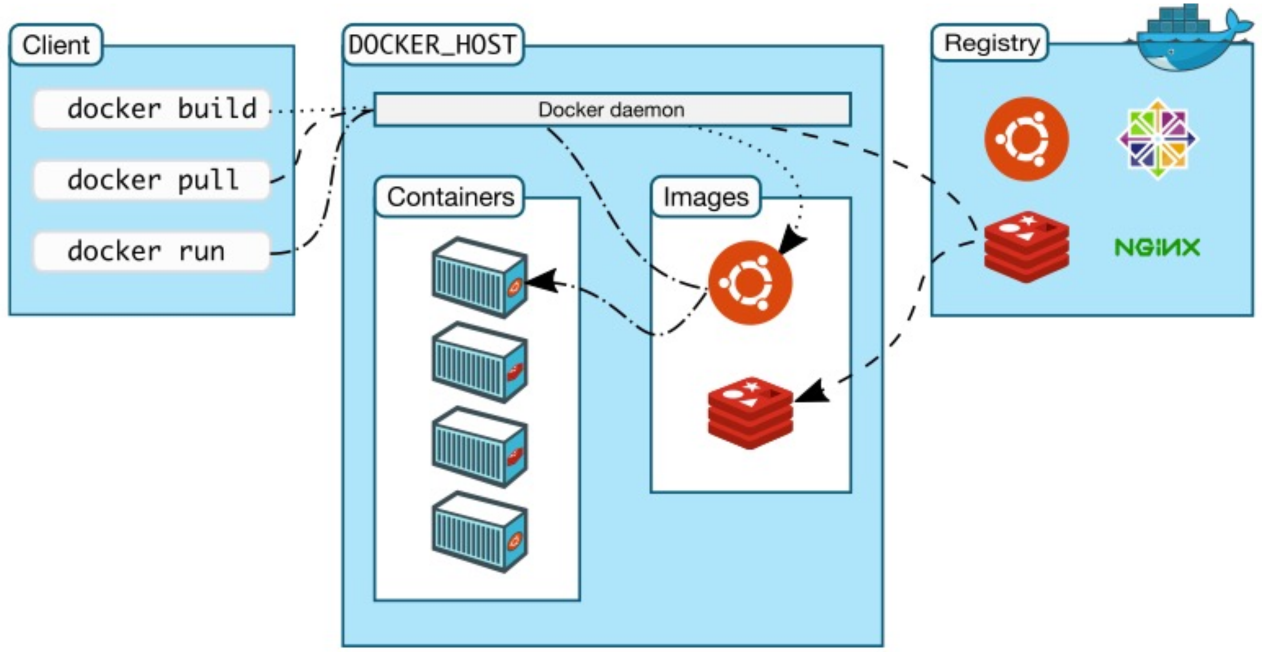


- *Better resources utilization (less overhead) : CPU, RAM*
- *Faster to stop/start applications (seconds)*
- *Enable powerful portability*
- *Multiple application on the same hosts*
- Abstract systems

Inside Docker

- Docker Registry: Repository of images
- Docker Engine : Client / Server
- Docker Swarm: Cluster Multi-hosting support
- Docker Compose : Orchestration / Scripting



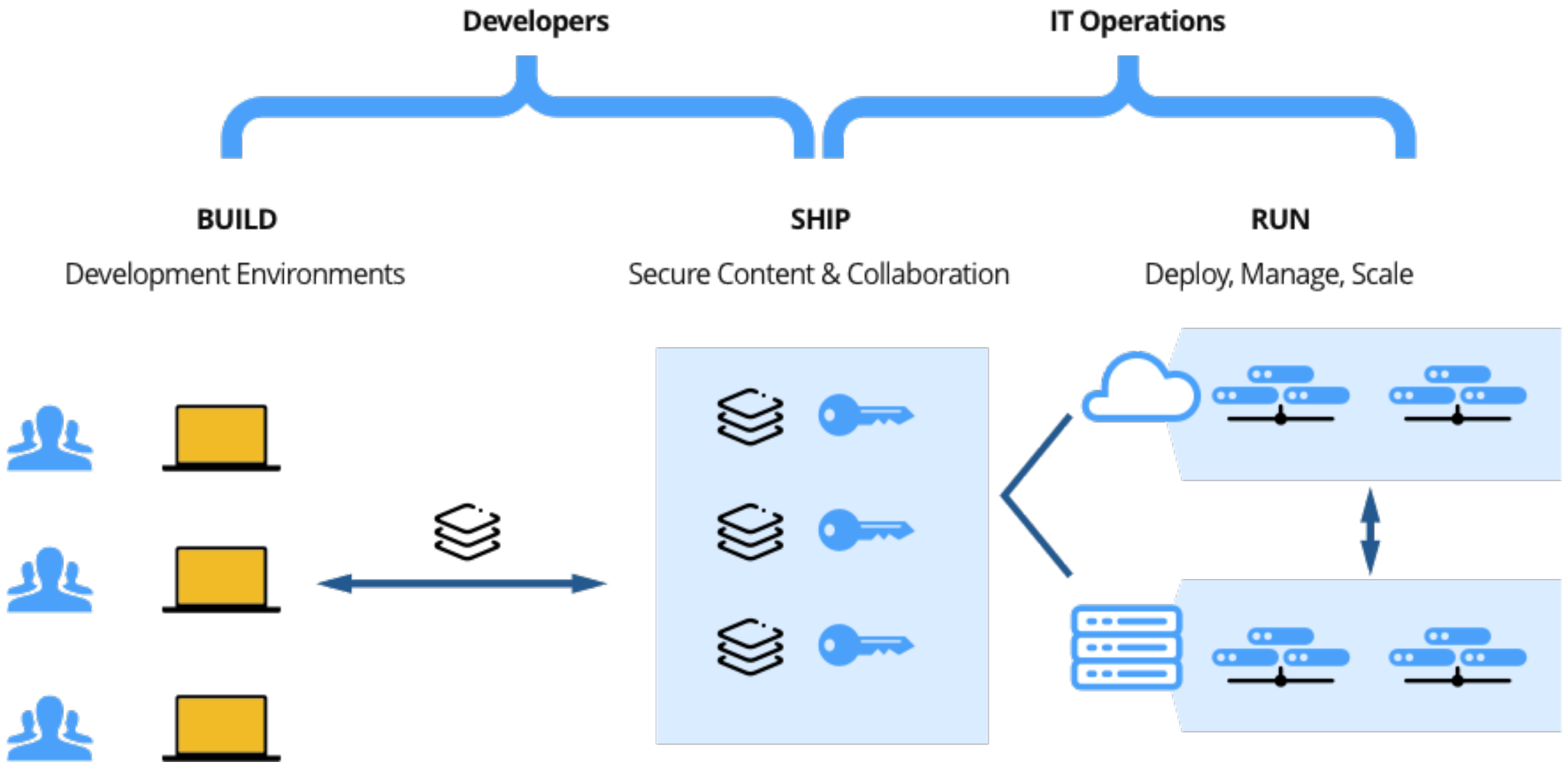


Docker image and container

- An image is a lightweight, stand-alone, executable package that includes
- everything needed to run a piece of software:
- • Contains the os, the application executable and their dependencies
- • Built with instructions from a Dockerfile

- A container is a runtime instance of an image – what the image becomes
- in memory when actually executed:
- • Run apps natively on the host machine’s kernel
- • Running in a discrete process (isolated environment)
- • Containers on the same machine share a single kernel

DevOps C’est quoi ce truc ?



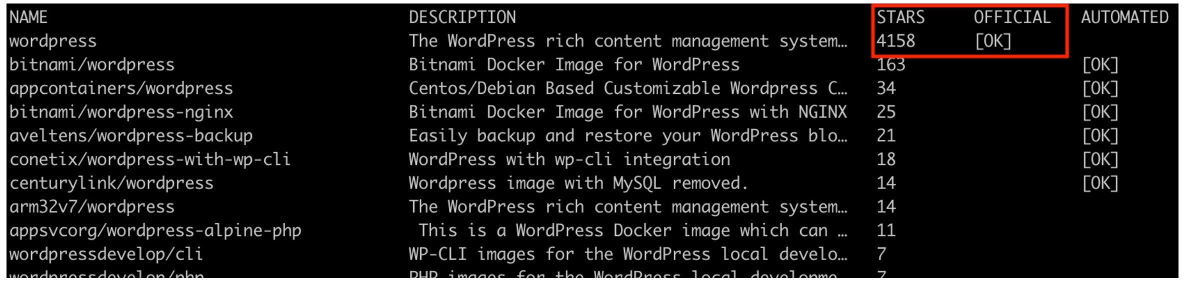
Travaux Pratiques

Docker Installation

- Documentation :
  - Windows 10 : https://docs.docker.com/docker-for-windows/install/
  - Windows 7 : https://docs.docker.com/toolbox/toolbox_install_windows/ (Legacy)
  - Mac : https://docs.docker.com/docker-for-mac/install/
  - Ubuntu : https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/

- Play with Docker
  - http://labs.play-with-docker.com

Ex 1 - Docker Basics



```
NAME                              DESCRIPTION                                     STARS     OFFICIAL    AUTOMATED
wordpress                         The WordPress rich content management system…   4158      [OK]
bitnami/wordpress                 Bitnami Docker Image for WordPress              163                   [OK]
appcontainers/wordpress           Centos/Debian Based Customizable Wordpress C…   34                    [OK]
bitnami/wordpress-nginx           Bitnami Docker Image for WordPress with NGINX   25                    [OK]
aveltens/wordpress-backup         Easily backup and restore your WordPress blo…   21                    [OK]
conetix/wordpress-with-wp-cli     WordPress with wp-cli integration               18                    [OK]
centurylink/wordpress             Wordpress image with MySQL removed.             14                    [OK]
arm32v7/wordpress                 The WordPress rich content management system…   14
appsvcorg/wordpress-alpine-php     This is a WordPress Docker image which can …   11
wordpressdevelop/cli              WP-CLI images for the WordPress local develo…   7
wordpressdevelop/php              PHP images for the WordPress local developme…   7
```

In a Web browser:

- **Go to: https://hub.docker.com/ and type wordpress in the search box**
- Select Wordpress Official

-> You get all the information about the available versions and how to use them

In a shell terminal:

- **docker search :** *The docker search command is used to search Docker Hub for images*

  ```
  $ docker search wordpress
  ```

  ```
    NAME         DESCRIPTION                                         STARS   OFFICIAL
  wordpress      The WordPress rich content management system…      4542      [OK]
  ```

- **docker pull:** *The docker pull command is used to pull/download a docker image from Docker hub or another registry*

  ```
  $ docker pull wordpress
  ```

  -> *This command will retrieve the latest wordpress version*

Ex 1 - Docker Basics

- **docker images:** *The docker images command is used to view the docker images that are on the system*

  $ **docker images**

```
REPOSITORY  TAG          IMAGE ID      CREATED       SIZE
wordpress   latest       d4f1eb34e2f5  10 days ago   616MB
```

Ex 1 - Docker Basics

- **docker run:** *The docker run command is used to start a docker container*
  -> You can find the instructions on the MariaDB web page on the Docker Hub web site: https://hub.docker.com/_/mariadb/

```
$ docker run --name some-wordpress -p 8080:80 -d wordpress
```

--name some-wordpress : *(Optional) Allows to specify a given name for the resulting container*

-d : *Allows to run the command in background*

-p 8080:80 : Exposes the container port 80 to the host port 8080. The wordpress is accessible on the port 8080.

-e <name>=<value> : Allows to set environment variables

Then, access it via http://localhost:8080 or http://host-ip:8080 in a browser.

Ex 1 - Docker Basics

- **docker exec:** *The docker exec command is used to run a command in a container*

```
$ docker exec -ti some-wordpress echo “Hello from container!”
```

-> *Hello from container !*

```
$ docker exec -ti some-wordpress bash
```

-> *You are running a bash inside the container*

```
# ls
```

-> *You can see the container local file system*

```
# exit
```

(exit from the container)

Ex 1 - Docker Basics

- **docker ps:** *The docker ps command is used to view running containers*

```
$ docker ps
CONTAINER ID   IMAGE       COMMAND                   PORTS                   NAMES
80b45fb18d33   wordpress   "docker-entrypoint.s…"   0.0.0.0:8080->80/tcp    some-wordpress
$ docker ps  -a
```

  List all the container instances even if they are stopped

- **docker stop:** *The docker stop command is used to stop a running container*

```
$ docker stop some-wordpress
```

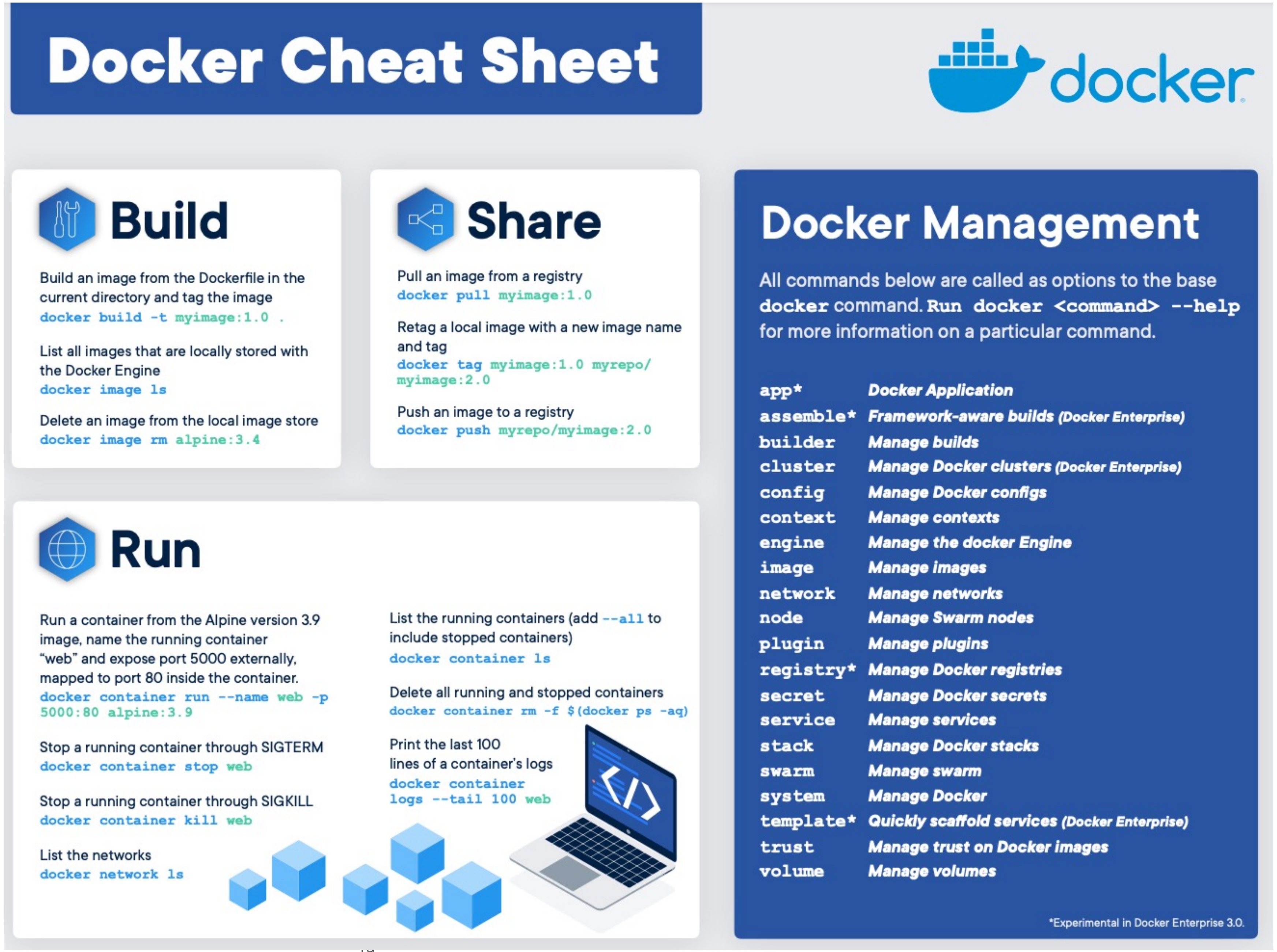
- **docker logs:** *The docker logs command is used to view log data from a container*

```
$ docker logs some-wordpress

WordPress not found in /var/www/html - copying now...
Complete! WordPress has been successfully copied to /var/www/html
….
```

- **docker rm :** *The docker rm command is used to delete a stopped container*

```
$ docker rm some-wordpress
```



Docker Cheat Sheet

Build

Build an image from the Dockerfile in the current directory and tag the image
```
docker build -t myimage:1.0 .
```

List all images that are locally stored with the Docker Engine
```
docker image ls
```

Delete an image from the local image store
```
docker image rm alpine:3.4
```

Share

Pull an image from a registry
```
docker pull myimage:1.0
```

Retag a local image with a new image name and tag
```
docker tag myimage:1.0 myrepo/
myimage:2.0
```

Push an image to a registry
```
docker push myrepo/myimage:2.0
```

Run

Run a container from the Alpine version 3.9 image, name the running container "web" and expose port 5000 externally, mapped to port 80 inside the container.
```
docker container run --name web -p
5000:80 alpine:3.9
```

Stop a running container through SIGTERM
```
docker container stop web
```

Stop a running container through SIGKILL
```
docker container kill web
```

List the networks
```
docker network ls
```

List the running containers (add `--all` to include stopped containers)
```
docker container ls
```

Delete all running and stopped containers
```
docker container rm -f $(docker ps -aq)
```

Print the last 100 lines of a container's logs
```
docker container
logs --tail 100 web
```

Docker Management

All commands below are called as options to the base **`docker`** command. Run **`docker <command> --help`** for more information on a particular command.

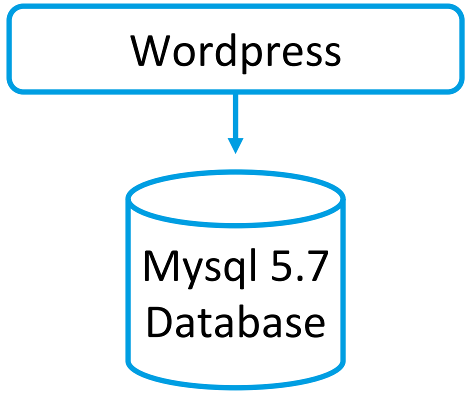
`app*`	***Docker Application***
`assemble*`	***Framework-aware builds (Docker Enterprise)***
`builder`	***Manage builds***
`cluster`	***Manage Docker clusters (Docker Enterprise)***
`config`	***Manage Docker configs***
`context`	***Manage contexts***
`engine`	***Manage the docker Engine***
`image`	***Manage images***
`network`	***Manage networks***
`node`	***Manage Swarm nodes***
`plugin`	***Manage plugins***
`registry*`	***Manage Docker registries***
`secret`	***Manage Docker secrets***
`service`	***Manage services***
`stack`	***Manage Docker stacks***
`swarm`	***Manage swarm***
`system`	***Manage Docker***
`template*`	***Quickly scaffold services (Docker Enterprise)***
`trust`	***Manage trust on Docker images***
`volume`	***Manage volumes***

*Experimental in Docker Enterprise 3.0.

Ex 1 - Docker Security and Tips

Good Practice:

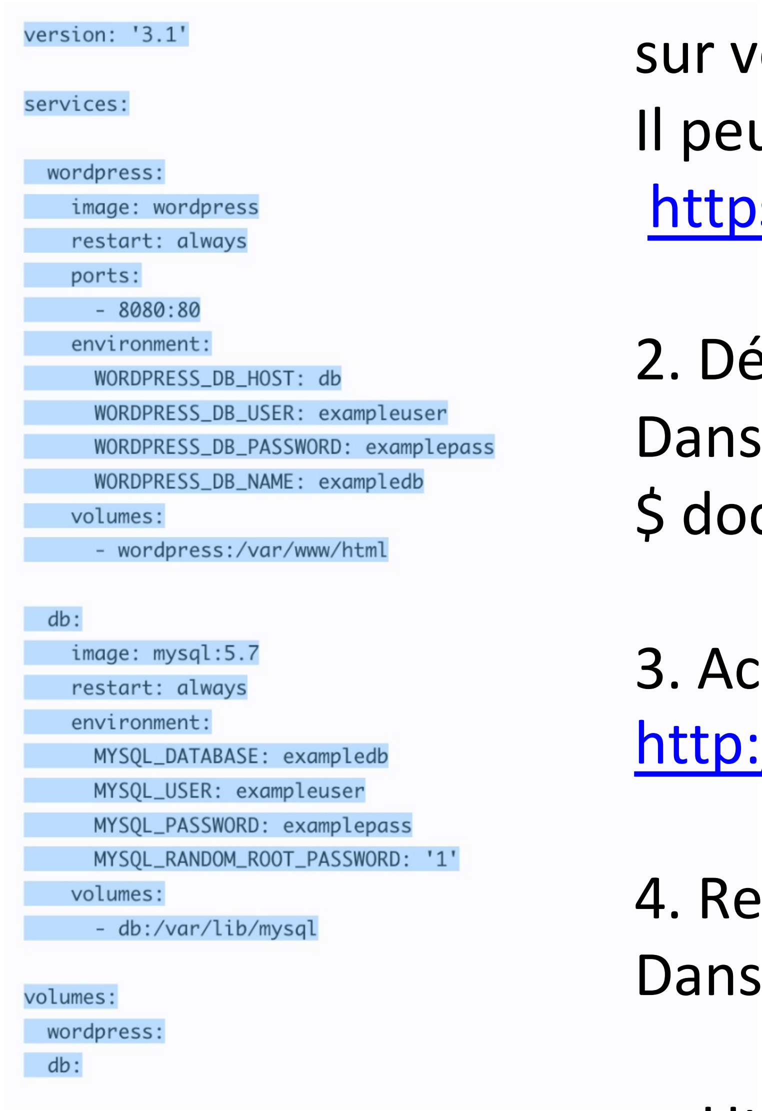
Ex 2 - Docker Advanced



Docker compose :

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application’s services. Then, with a single command, you create and start all the services from your configuration.

Ex 2 - Docker Advanced



```yaml
version: '3.1'

services:

  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:
```

1. Stocker ce contenu dans un fichier docker-compose.yaml sur votre machine.
Il peut être trouvé dans la page
https://hub.docker.com/_/wordpress

2. Démarrer la topologie (Wordpress + MySql)
Dans un shell (dans le repertoire où il y a le fichier)
$ docker-compose up

3. Acceder au site
http://localhost:8080

4. Regarder les logs de la topologie
Dans un shell (dans le repertoire où il y a le fichier)

-> Utilisation de la ligne de commande docker-compose

Ex 2 - Docker Advanced

5. Aller dans le container wordpress et cherche le fichier configuration PHP: php.ini-production.

6. Extraire le fichier de configuration. (Utiliser une commande docker ou docker-compose)

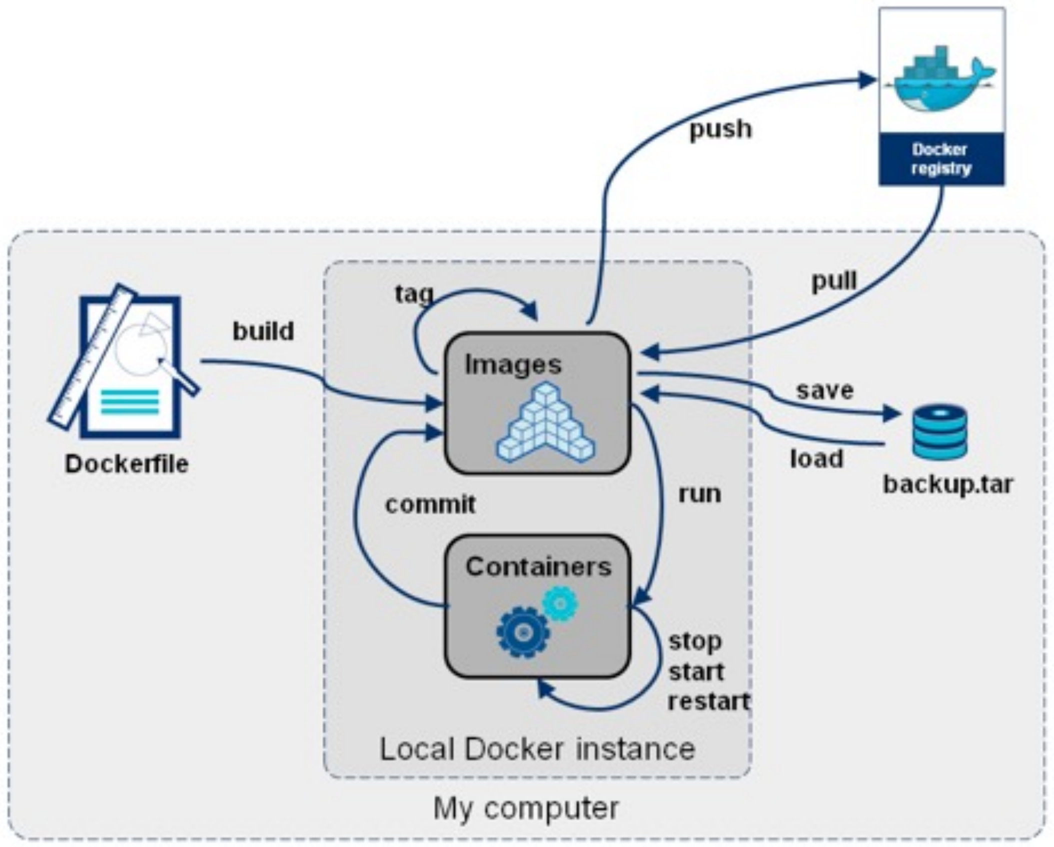
7. Vérifier la consommation CPU et Memoire des containeurs tournants sur votre machine. (Commande docker …)

8. Changer le port d'accès au wordpress
    - Doit être accéssible en http://localhost:9080

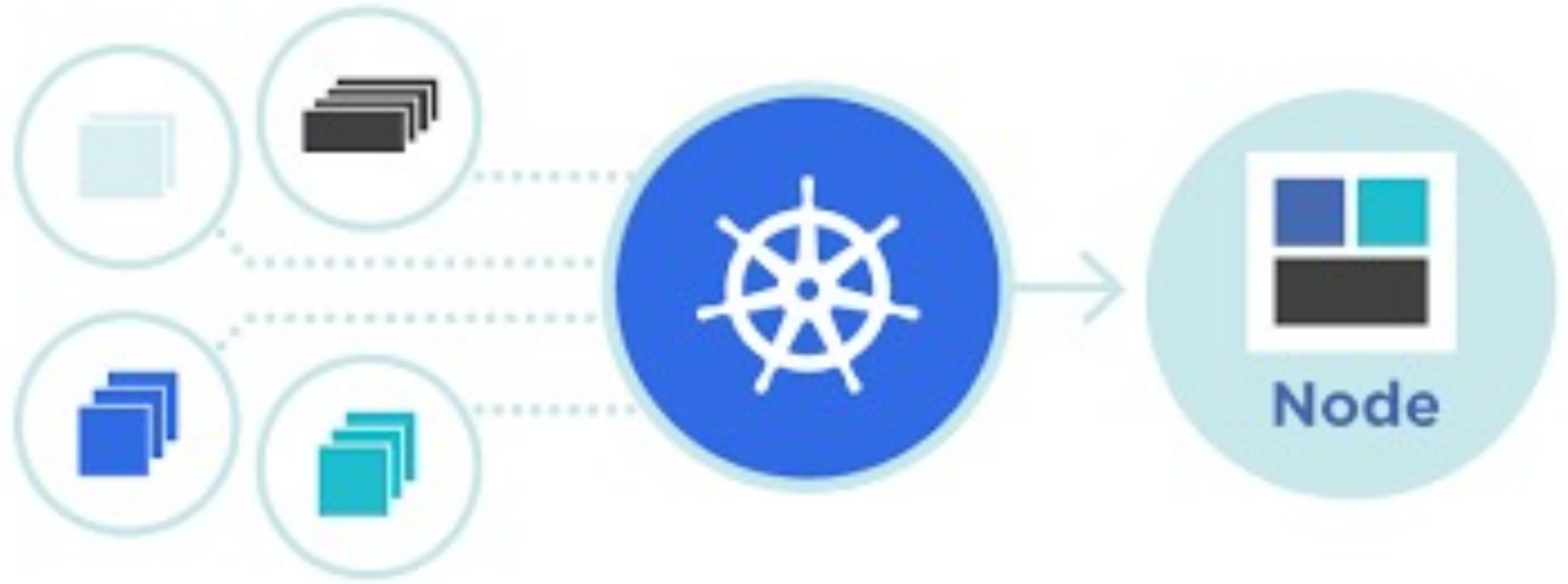
9. Changer le nom de la base de données.

Ex 3 - Docker build

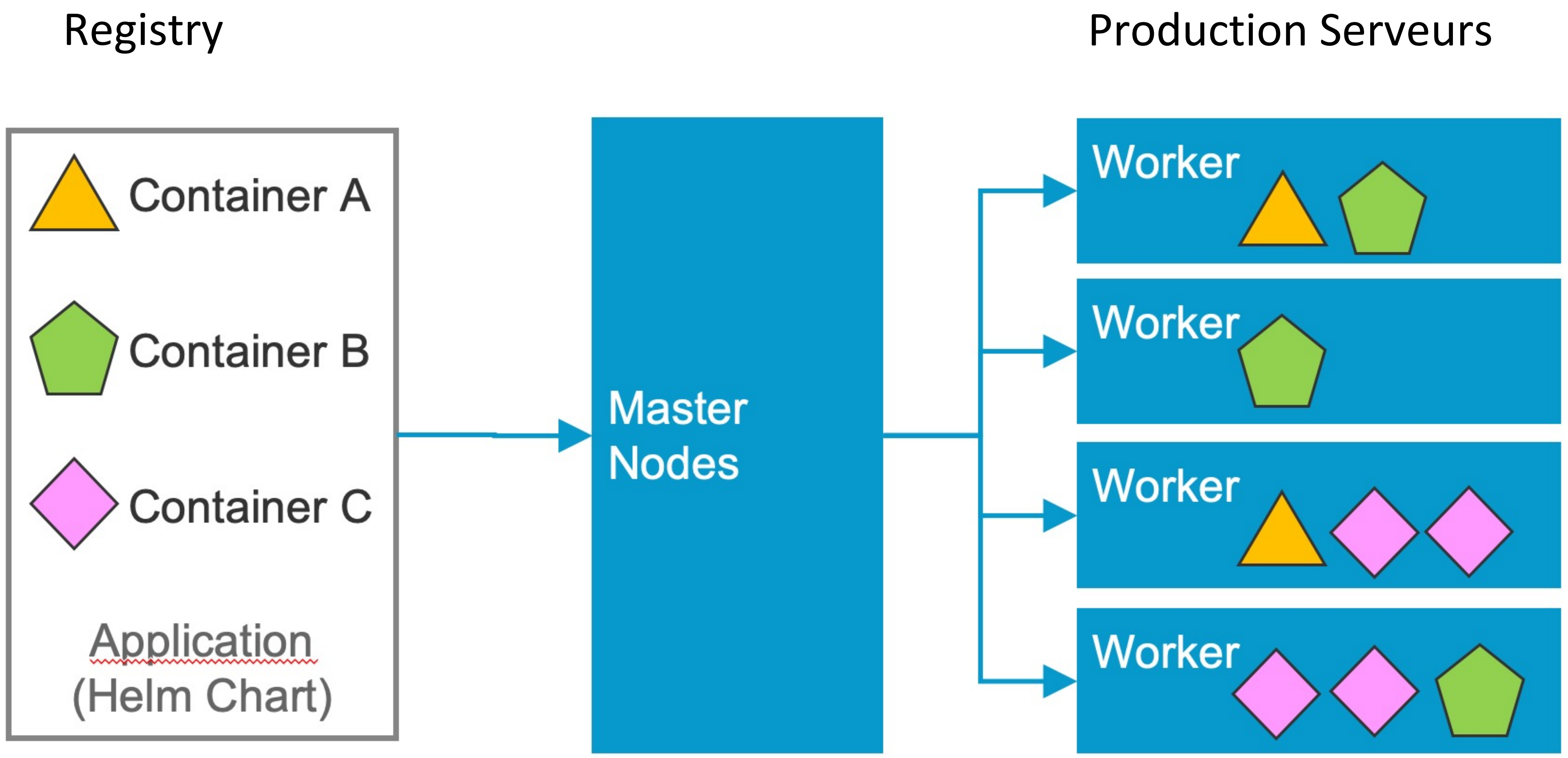
- https://www.docker.com/blog/how-to-use-the-official-nginx-docker-image/



Kubernetes



Problématique : Orchestration



Container Orchestration

Layer 6	Development Workflow Opinionated Containers	Cloud Foundry, OpenShift, Docker Cloud, Deis, Apcera, Apprenda
Layer 5	**Orchestration/Scheduling Service Model**	**Kubernetes, Docker Swarm, Marathon/Mesos, Nomad, Diego**
Layer 4	Container Engine	Docker, rkt, runC (OCI), Osv, LXC, LXD, CRIO
Layer 3	Operating System	Ubuntu, RHEL, CoreOS, Unikernels
Layer 2	Virtual Infrastructure	vSphere, EC2, GCP, Azure, OpenStack
Layer 1	Physical Infrastructure	Raw Compute, Storage, Network

What is Kubernetes ?

- Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.

- **Container orchestrator**
- Runs and manages containers
- Supports **multiple cloud and bare-metal** environments
- Inspired and informed by Google's experiences and internal systems
- **100% Open source**, written in Go
- **Manage applications**, not machines
- Rich ecosystem of plug-ins for scheduling, storage, networking

What does Kubernetes do?

- Provide a runtime environment for Docker containers
- Scale and load balance docker containers
- Abstract away the infrastructure containers run on
- Monitor/health check containers
- Declarative definition for running containers
- Update containers (also rolling updates)
- Storage mounting (allow abstracting infrastructure)
- Service discovery and exposure
- Labelling and selection of any kind of object (we‘ll get to this)

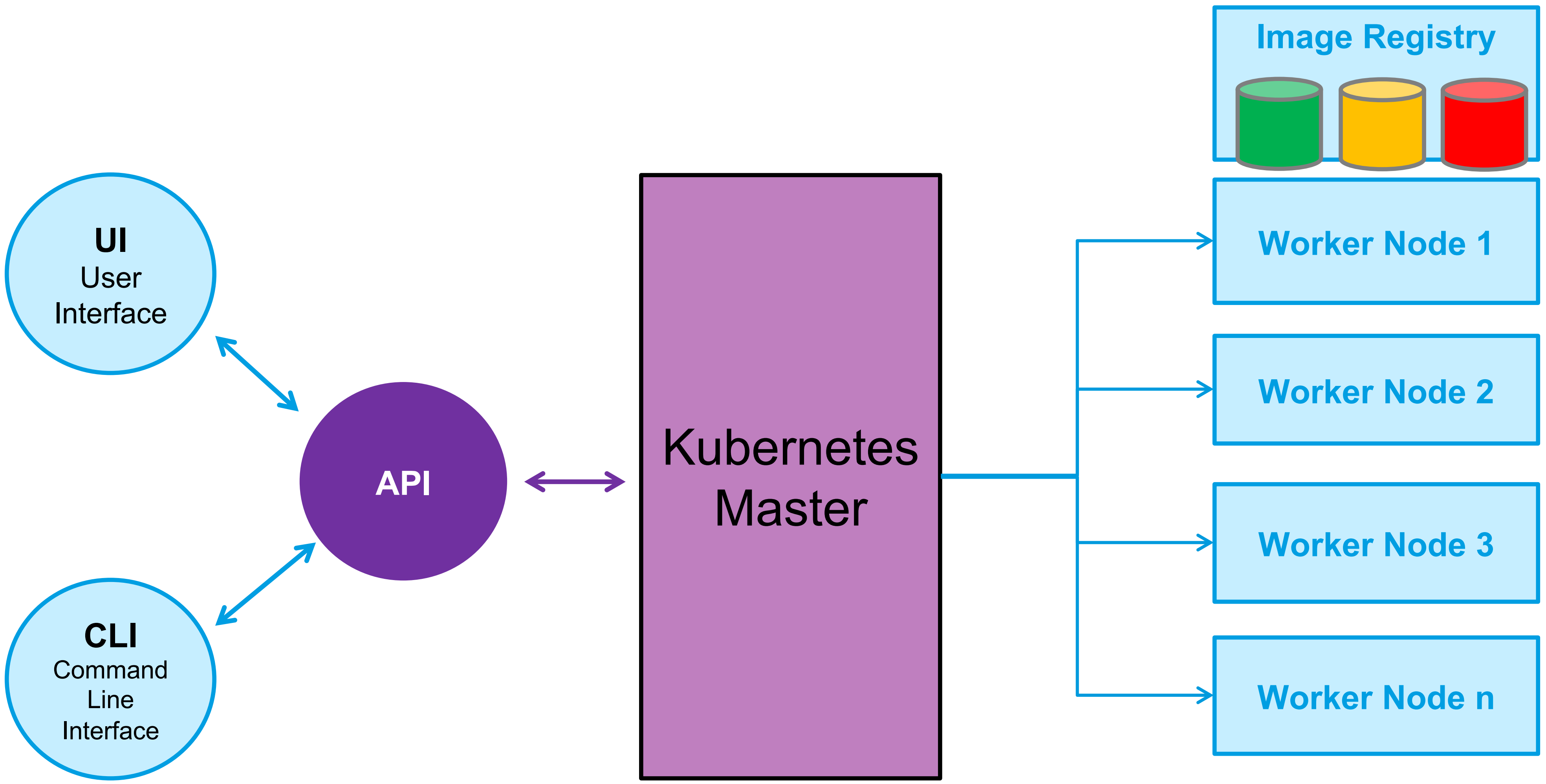
Kubernetes Terminology

- Node
    - Hosts that run Kubernetes applications
- Master node
    - Controls and manages the cluster
    - Kubectl – command line client
    - REST API – used for communicating with the workers
    - Scheduling and replication logic
    - Generally 2 or more master nodes for resiliency, but are not used for scaling out the cluster
- Worker node
    - Node that hosts the K8 services
    - Kubelet – K8s agent that accepts commands from the master
    - Kubeproxy – network proxy service on a node level
        - Responsible for routing activities for inbound or ingress traffic
    - Docker host
- Containers
    - Units of packaging
- Pods
    - A collection of containers that run on a worker node
    - A pod can contain more than one service
    - Each pod has it's own IP
    - A pod shares a PID namespace, network, and hostname

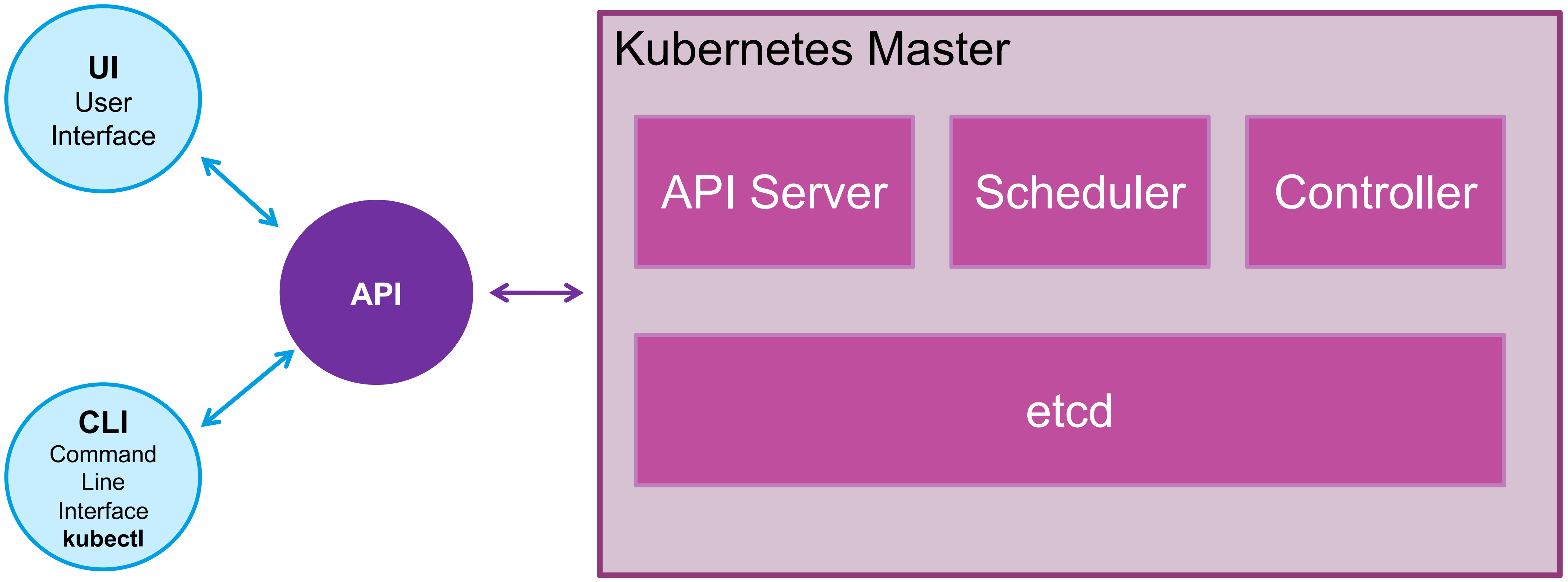
Kubernetes Terminology (cont)

- Replication Controller
    - Ensures availability and scalability
    - Responsible for maintaining as many pods as requested by the user
    - Uses a template that describes specifically what each pod should contain
- Labels
    - Metadata assigned to K8 resources – such as pods, services
    - Key-Value pairs for identification
    - Critical to K8s as it relies on querying the cluster for resources that have certain labels
- Services
    - Collection of pods exposed as an endpoint
    - Information stored in the K8 cluster state and networking info propagated to all worker nodes
- Secrets
    - Sensitive information that containers need to read or consume
    - Are special volumes mounted automatically so that the containers can read its contents
    - Each entry has it’s own path
- Proxy
    - A load balancer for pods
- Etcd
    - A metadata service providing the backend data store

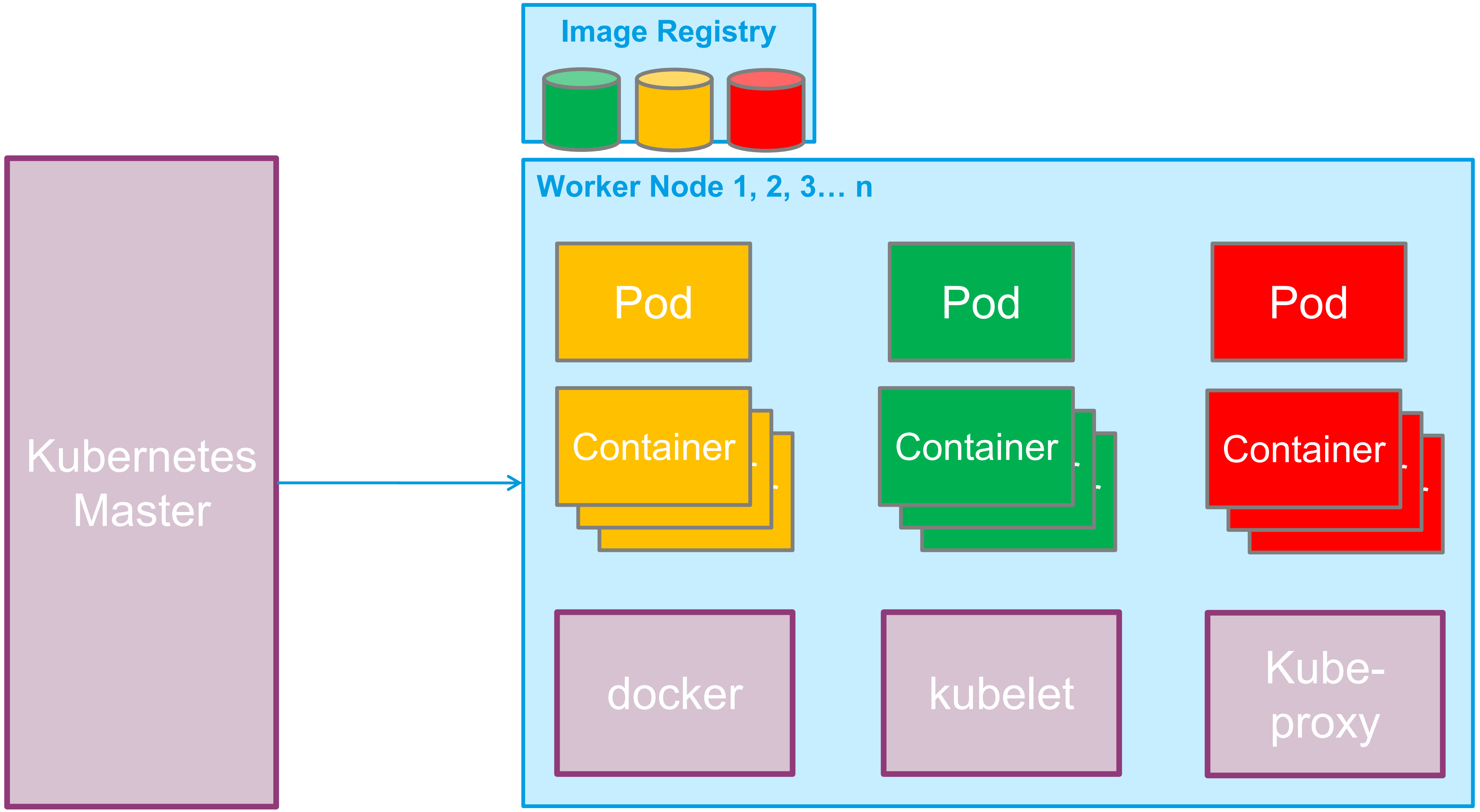
Kubernetes Architecture



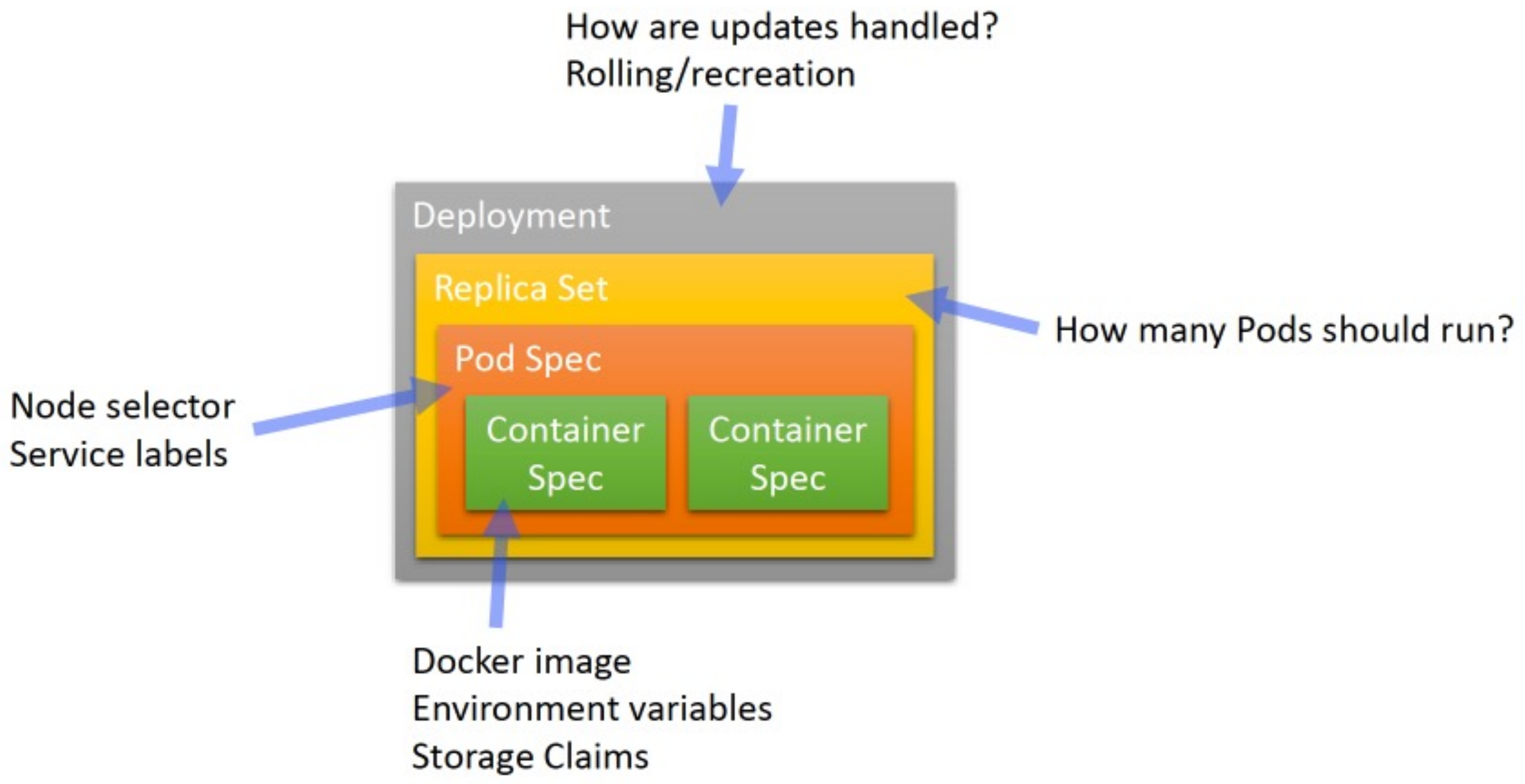
Kubernetes Architecture



Kubernetes Architecture



Kubernetes Concepts : Deployment, Replica, Pod



Kubernetes recap

- **Container image** : Docker container image, contains your application code in an isolated environment.
- **Pod** : A pod is a management unit in Kubernetes comprised of one or more containers. Each pod has its own unique IP address and storage namespaces. All containers share these networking and storage resources.
- **Deployment** : A Deployment is a new way to handle High Availability (HA) in Kubernetes in place of the Replication Controller. A pod by itself is "mortal" but with a Deployment, Kubernetes can make sure that the number of Pods that a user specifies is always up and running in the system. A Deployment specifies how many instances of a pod will run. A YAML file is used to define a Deployment.
- **Service**
  - According to the official Kubernetes website, "A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them – sometimes called a micro-service."
  - Cluster IP mode is internal to Kubernetes, and the NodePorts mode are the published IP addresses for external users to access the services. Routes to services are based on labels. As with Pods and Deployments, a service is also defined by a YAML file.

THANK YOU !