

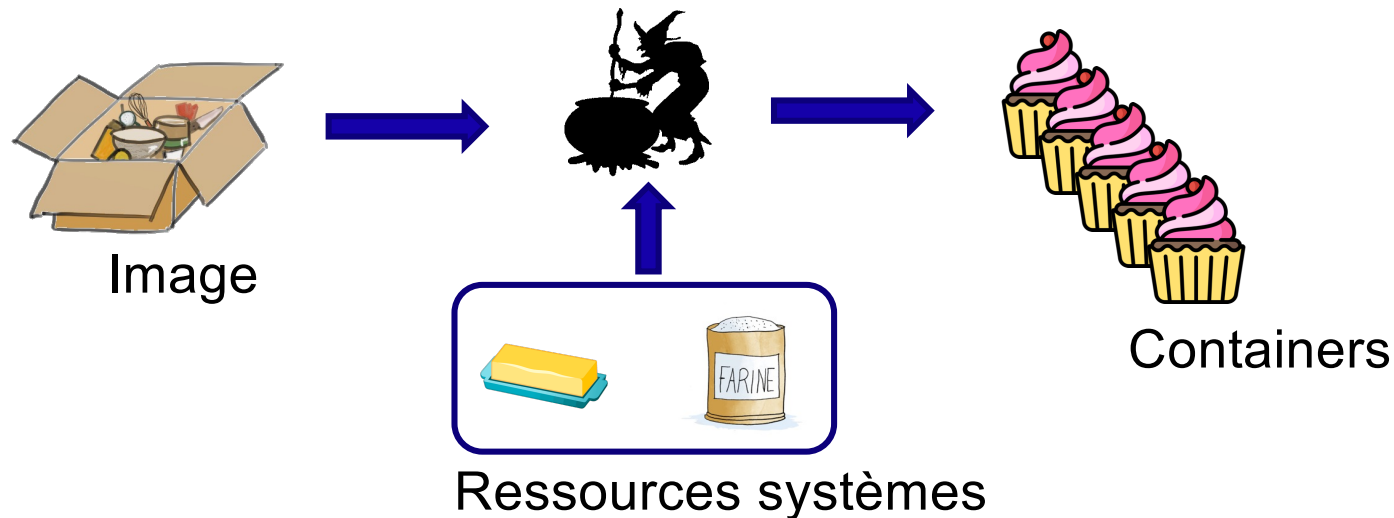
L'environnement TP



1. Pourquoi les containers ?
2. Docker !!!
3. Comment ça marche
4. Les premières commandes
5. Docker compose
6. L'environnement du module

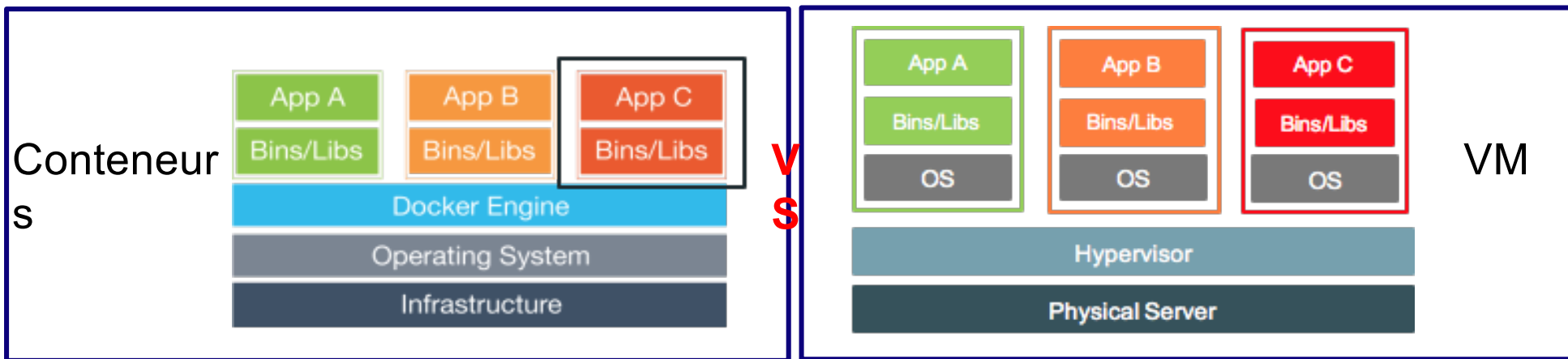
Pourquoi les containers ?

- ❑ Le principe consiste à regrouper votre application et toutes ses dépendances dans une image.
 - Garantie le fonctionnement de votre application quelque soit l'environnement où l'application est exécutée (Windows, Linux ...).
- ❑ Pour démarrer l'application contenu dans l'image, docker alloue des ressources systèmes (mémoire et RAM) pour exécuter le code applicatif.
- ❑ Le container virtualise l'environnement d'exécution et s'appuie sur l'OS de la machine hôte
- ❑ On appelle container l'instanciation d'une image.

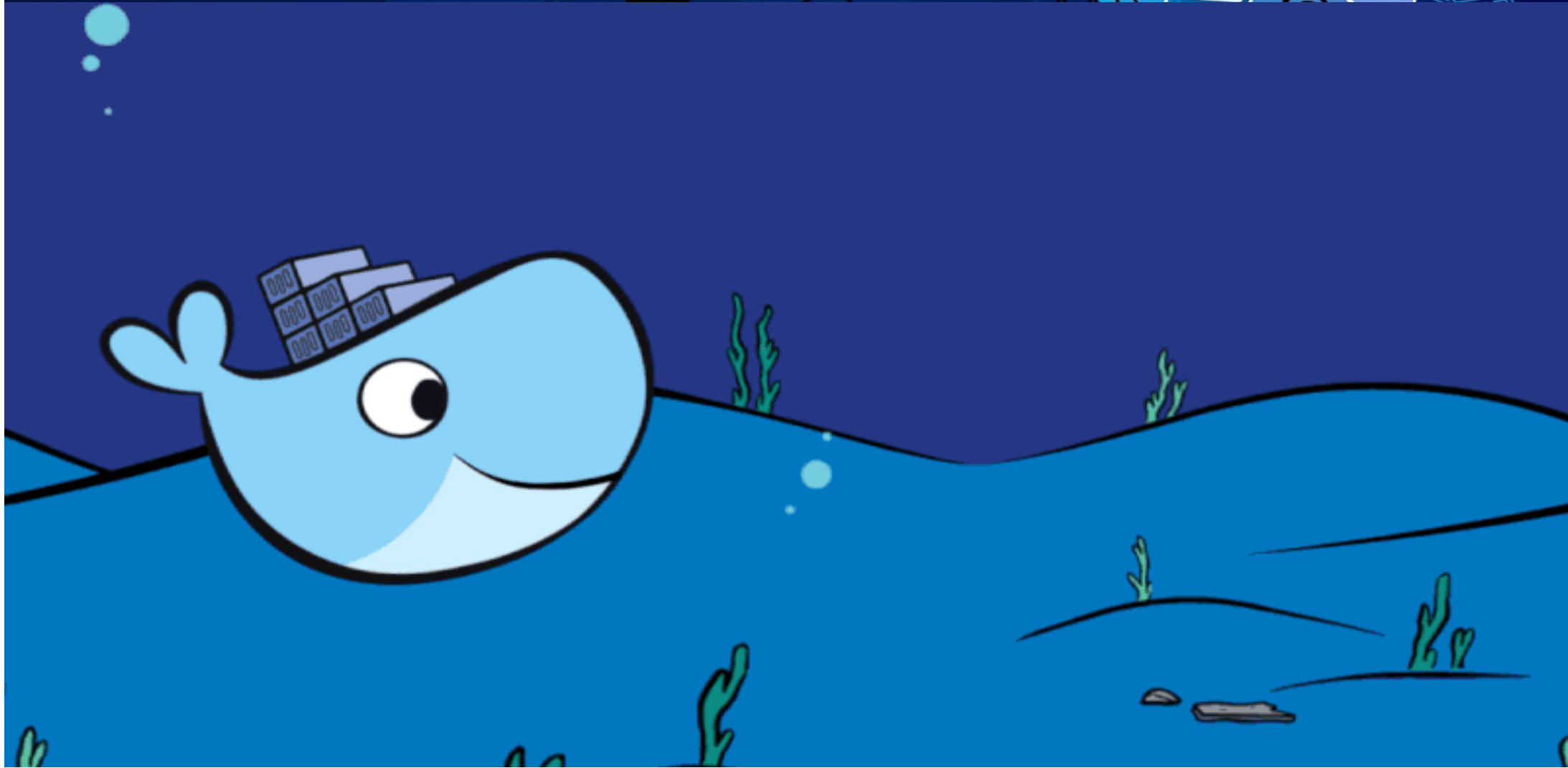


Pourquoi les containers ?

- ❑ Une Machine Virtuelle dispose de son propre OS lequel est dupliqué pour chaque Machine Virtuelle.
- ❑ Avec les containers, le principe est d'isoler l'exécution de nombreux processus sur un même hôte. Les conteneurs partagent le même noyau et une grande partie des services de l'OS hôte.
- ❑ Les ressources sur la machine physique sont moins sollicitées en RAM, CPU et disque.
- ❑ Les containers s'appuient sur la notion de **cgroups et namespaces**.
 - Cgroups définit les ressources systèmes à fournir à un processus.
 - Namespaces c'est une fonctionnalité du noyau Linux, qui permettent de faire en sorte que les processus ne voient pas les ressources utilisées par les autres.



Docker !!!



Les premières commandes

liste les containers en cours d'exécution

```
docker ps
```

liste les containers arrêtés et en cours d'exécution

```
docker ps -a
```

L'aide pour utiliser docker

```
docker --help
```

Instanciation d'un container à partir d'une image

```
docker run -d --name mongo mongo:latest
```

- * run instancie un container

- * -d execute le container en tâche de fond (detach)

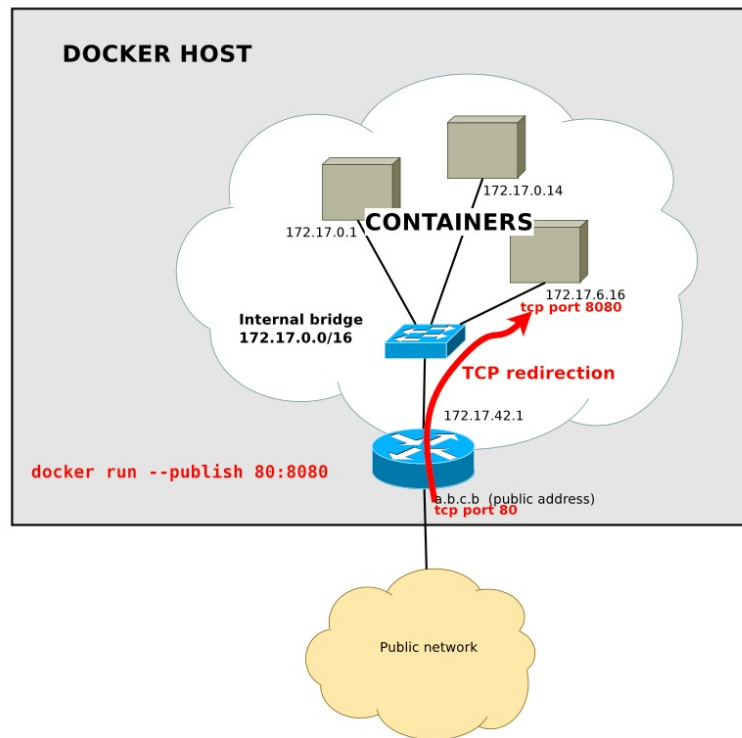
- * --name : nom associé au container produit

- * nginx nom de l'image à instancier

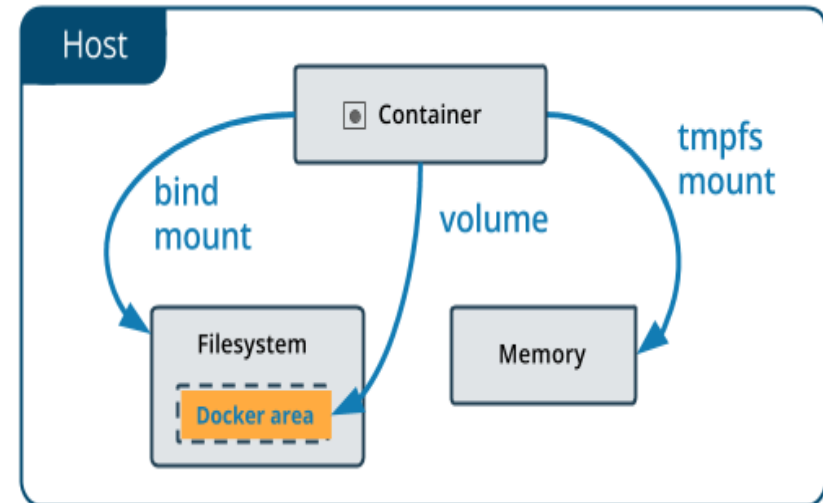
- * latest : version de l'image à instancier

L'environnement d'un container

Docker permet de gérer le réseau.



Docker permet de gérer le stockage.



Docker Compose

docker-compose.yml

```
#####  
# SERVEUR MONO-INSTANCE Mongo  
#####<  
  
mongosingle:  
  image: mongo  
  hostname: mongosingle  
  container_name: 'mongosingle'  
  volumes:  
    - ./mongoddb:/dataimport  
  ports:  
    - "27017:27017"  
  networks:  
    - mongonet
```

Docker fournit différentes options pour gérer le stockage et le réseau .

Pour faciliter la gestion des conteneurs nous utilisons la fonctionnalité compose qui permet de configurer l'exécution des conteneurs.

Un docker-compose.yml est disponible dans le répertoire git que vous avez téléchargé.

Les commandes docker seront fournies dans les énoncés des TP.

```
docker compose up -d mongosingle
```

```
docker compose stop mongosingle
```

```
docker compose start mongosingle
```

```
docker compose rm -f mongosingle
```

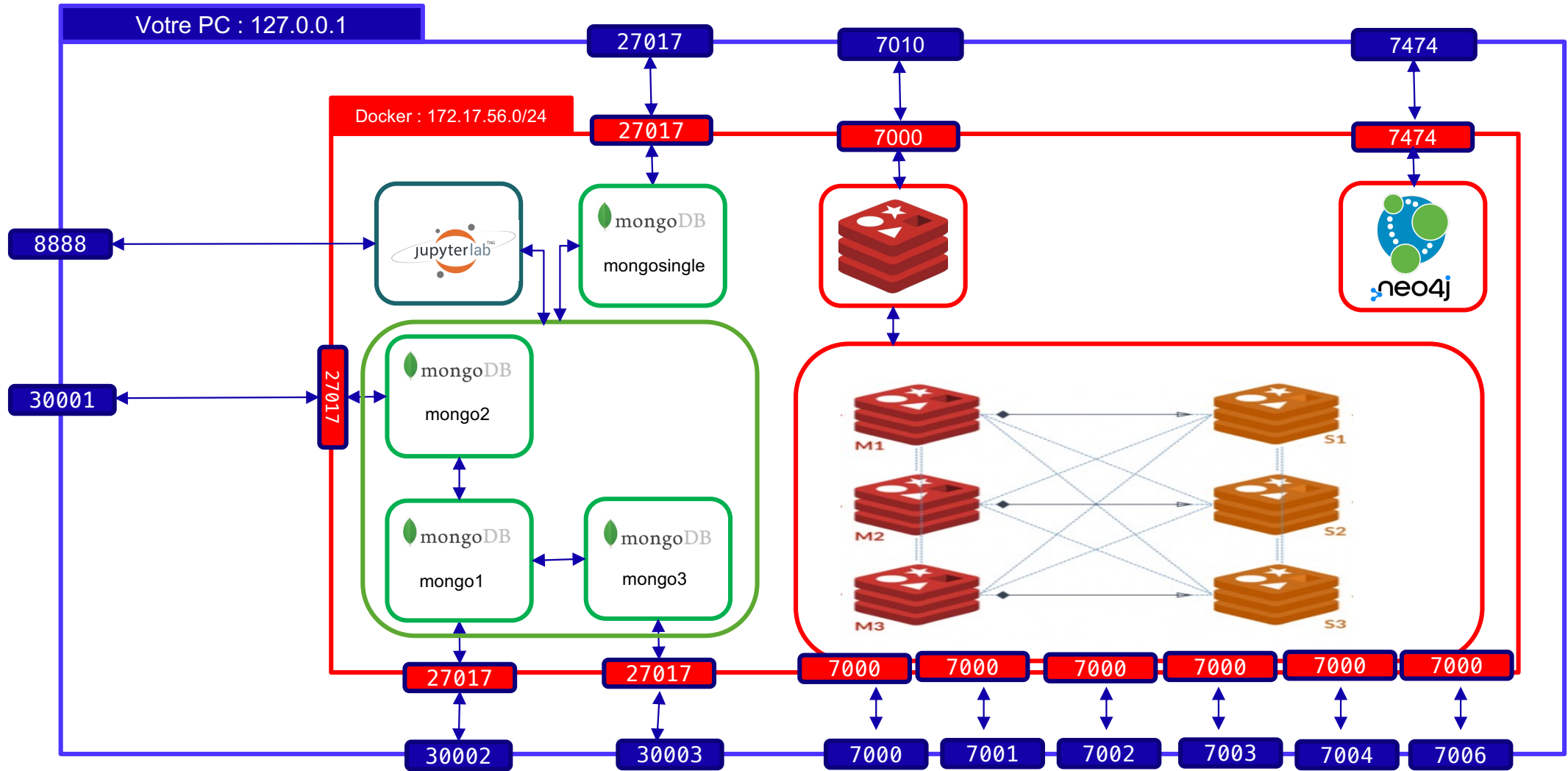

Docker Compose

```
# la commande docker compose permet d'opérer sur l'ensemble des containers décrit dans le fichier docker-  
compose.yml  
# Par défaut compose utilise le répertoire courant pour localiser le fichier docker-compose.yml mais il est  
possible de lui spécifier manuellement.  
docker compose -f /my/path/to/docker-compose.yml [option]  
Ou  
cd /my/path/to/  
docker compose -f /my/path/to/docker-compose.yml [option]  
  
# Avec l'option -d les containers sont exécutés en tâche de fond  
docker compose -f /my/path/to/docker-compose.yml up -d mongosingle  
  
# Une fois qu'un container est instancier avec la commande up il peut ensuite être arrêté et démarré autant de  
fois que vous le souhaitez.  
docker compose -f /my/path/to/docker-compose.yml stop mongosingle  
  
# Démarrage les containers instanciés  
docker compose -f /my/path/to/docker-compose.yml start mongosingle  
  
# Pour consulter les traces du container  
docker compose -f /my/path/to/docker-compose.yml log mongosingle
```

Liste complète des commandes docker-compose :

https://docs.docker.com/engine/reference/commandline/compose_down/

Environnement du module



TP Environnement

