Du concept à l'expertise

Docker

Virtualisation

docker

- [x] Théorie
- [ ] Pratique

Martin LEKPA

Programme

1. Virtualisation
2. Conteneurisation
3. Observabilité

© Freepik


01
Virtualisation
Définition et concepts

Virtualisation

- Serveur hôte
- Serveur privé virtuel
- Environnement isolé
- Système d'exploitation

Avantages

- Utilisation optimale des ressources
- Installation, déploiement et migration facile
- Sécurisation et/ou Isolation d'un réseau
- Socle de base du Cloud Computing
- ...

Inconvénients

- Parfois inadapté (Ex : I/O intense).
- Un recours à des machines puissantes
- Une complexité accrue de l'analyse d'erreurs
- ...

Terminologie

- Le **système hôte (host)** est l'OS principal de l'ordinateur.
- Le **système invité (guest)** est l'OS installé à l'intérieur d'une machine virtuelle.
- Une **machine virtuelle (VM)** est un ordinateur virtuel qui utilise un système invité.
- Un ordinateur virtuel est aussi appelé serveur privé virtuel **(Virtual Private Server ou VPS)** ou environnement virtuel (Virtual Environment ou VE)

Domaines de virtualisation

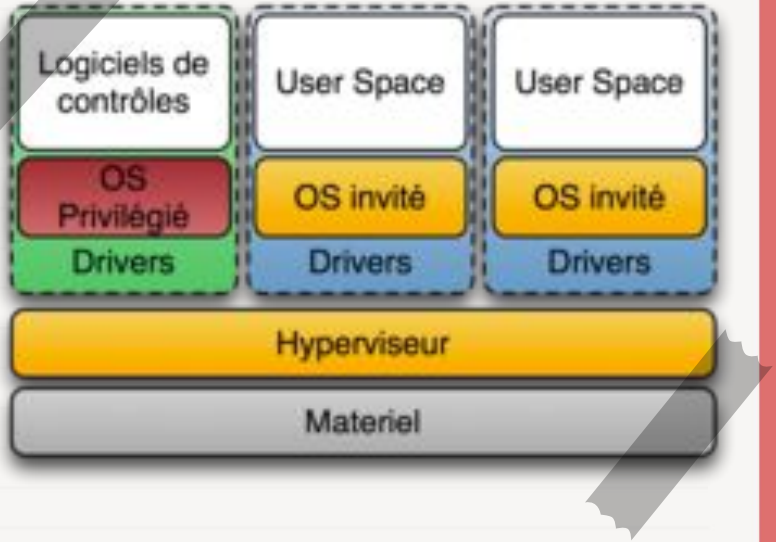
- Virtualisation de serveurs
- Virtualisation d'applications*
- Virtualisation du stockage
- Virtualisation du réseau

* Non développé dans ce cours

Virtualisation de serveurs

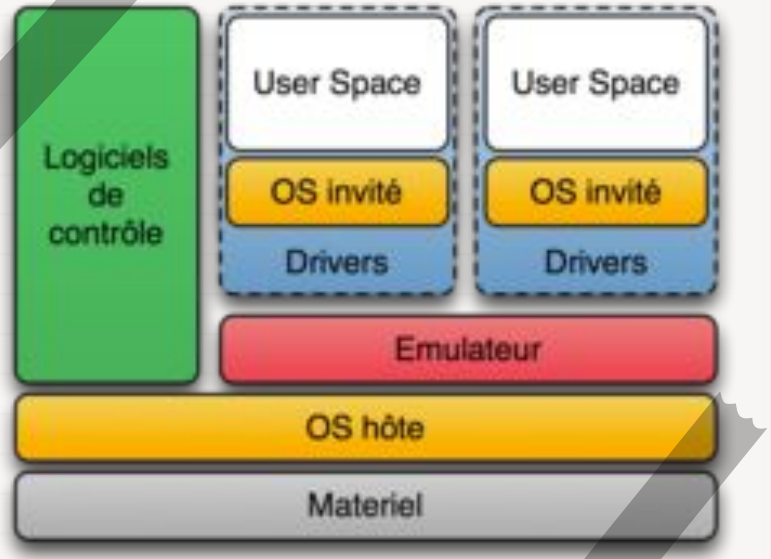
Hyperviseur de type 1

- Noyau système très léger
- Gestion noyaux OS invités
  - VMware vSphere
  - Microsoft Hyper-V Server
  - Proxmox VE
  - Oracle VM, KVM, ...



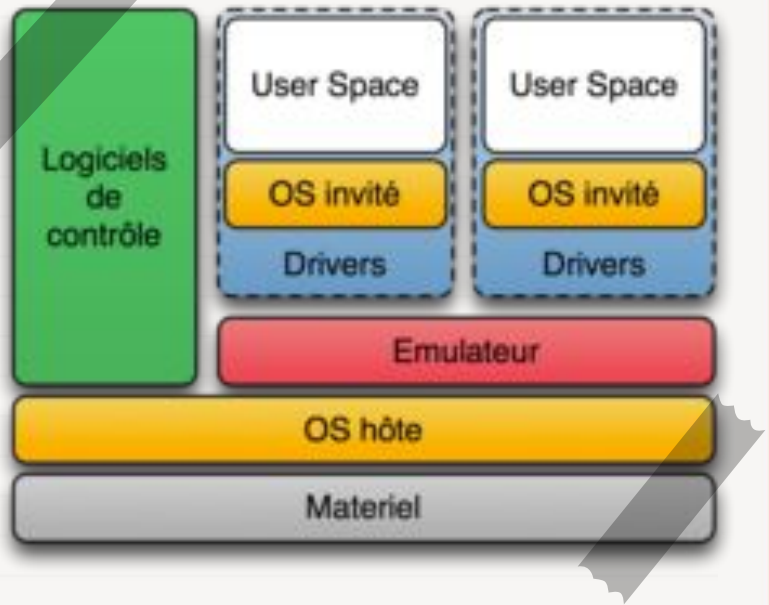
Hyperviseur de type 2

- Logiciel sur OS hôte
- Gestion OS invités
  - Vmware Fusion/Player
  - Oracle VM Virtualbox
  - QEMU
  - Microsoft VirtualPC/Virtual Server ...



Isolateur

- Logiciel sur OS hôte
- Isolation des exécutions
  - chroot
  - LXC
  - Docker
  - ...



Les usages

- Hyperviseur de type 1 -> Professionnel
- Hyperviseur de type 2 -> Professionnel / Particulier
- Isolateur -> Professionnel / Particulier
- Noyau en espace utilisateur* -> Développement de noyaux

* Non développé dans ce cours

02

La Conteneurisation

Cas de Docker

Conteneurs

- Environnement isolé
- Système hôte
- Fonctions de configuration
- Dépend de l'hôte/kernel

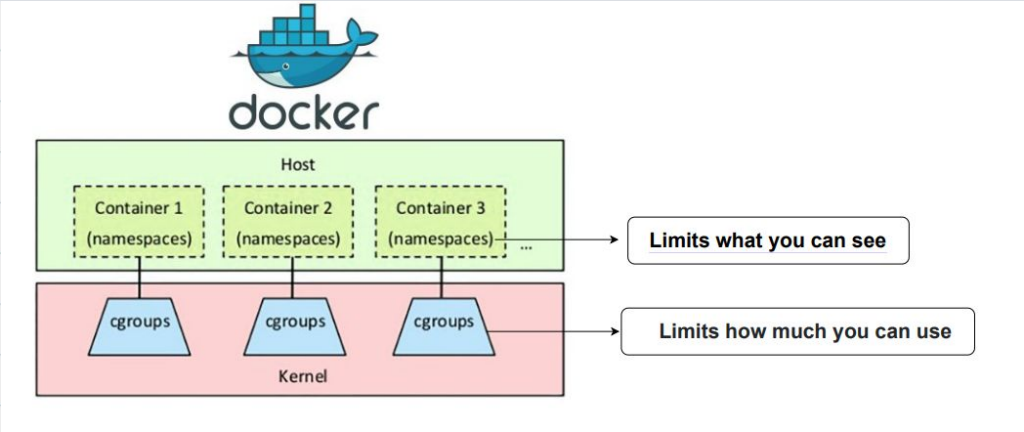
Les types de conteneurs

- **System** : simule une séquence de boot complète avec un init process ainsi que plusieurs processus (LXC, OpenVZ).

- **Process** : un conteneur exécute un ou plusieurs processus directement, en fonction de l'application conteneurisée (Docker, Rkt).

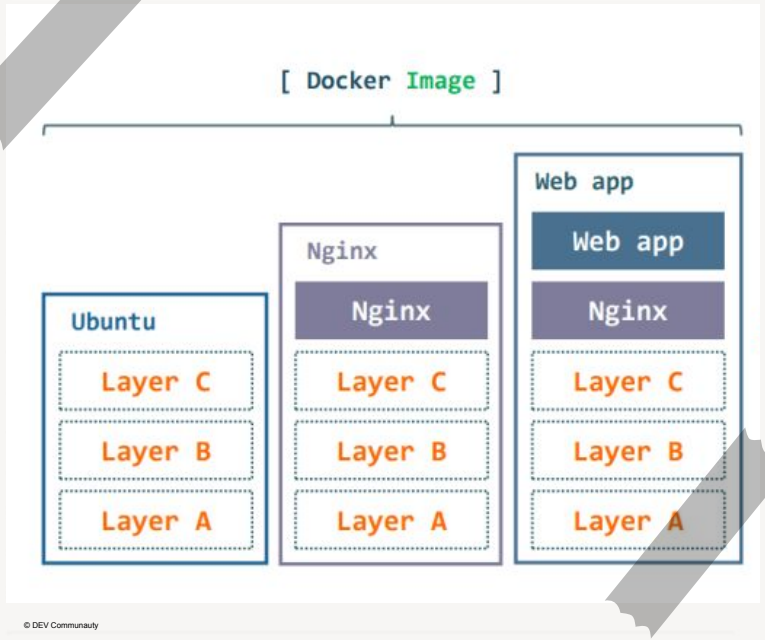
Notions importants

- Cgroup (Control Groups)
- Namespaces



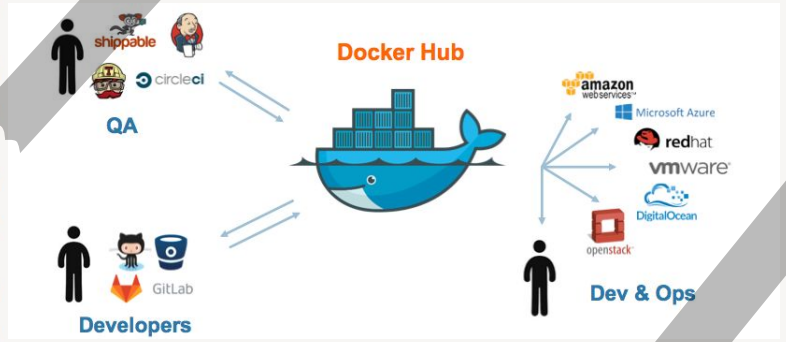
Les images

- Point de départ
- Archives / Snapshot
- On ne refait pas la roue



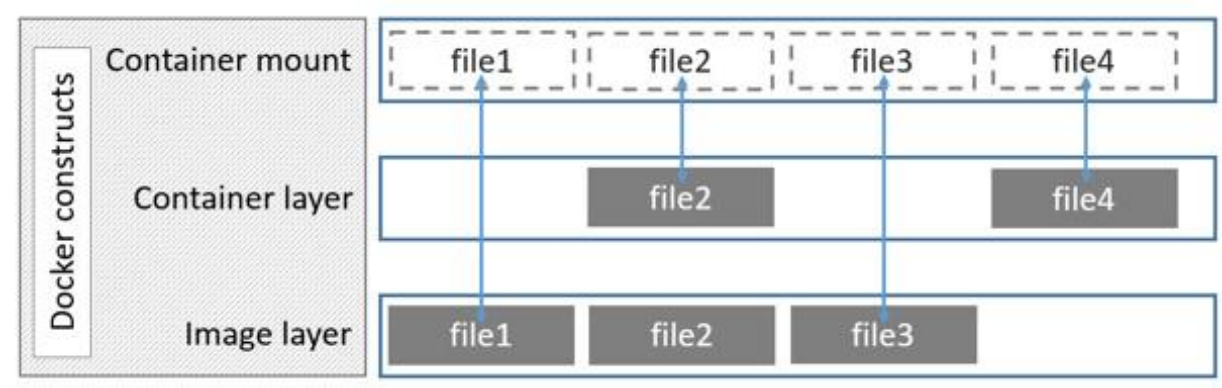
Le registre

- Distributeur d'images
- Sécurité ?
- Docker Hub



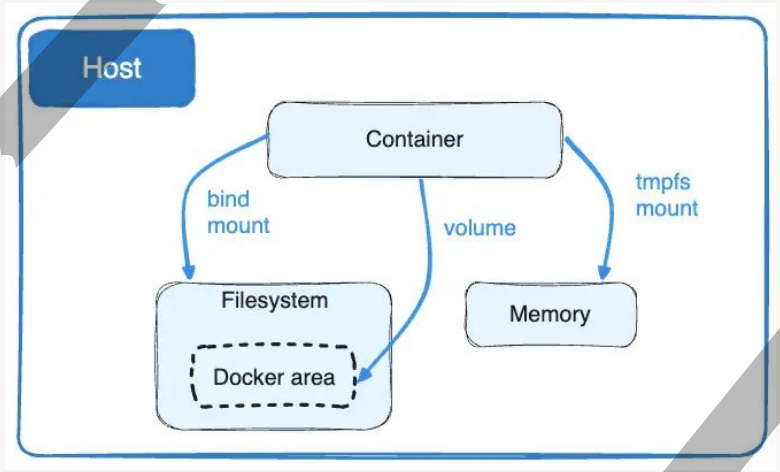
L'empilement des couches (Layers)

- Structure en couche
- Partage de couches



Le stockage

- AUFS
- DeviceMapper
- OverlayFS
- Plugins

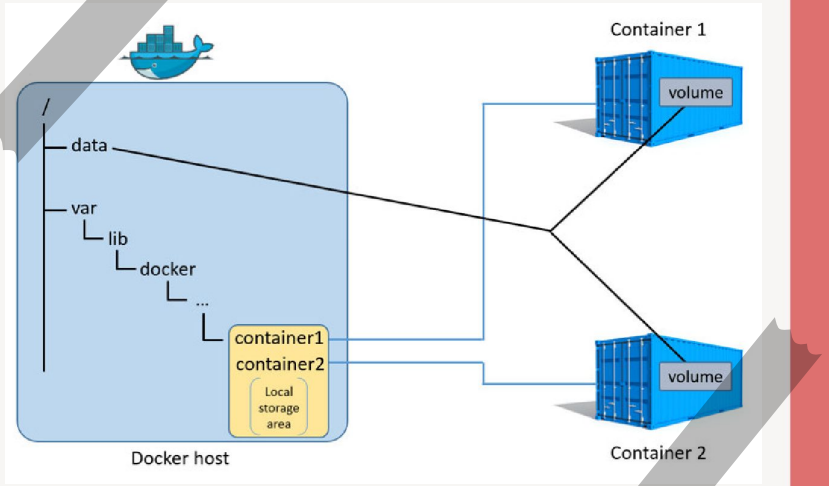


Les volumes

- Persistance des données
- Indépendance vis à vis des conteneurs/layers
- Deux types de volume:

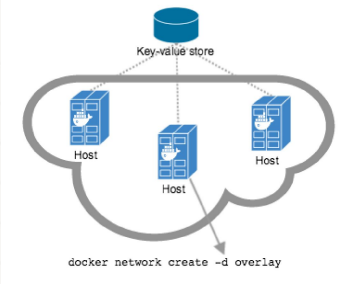
  Conteneur : data conteneur

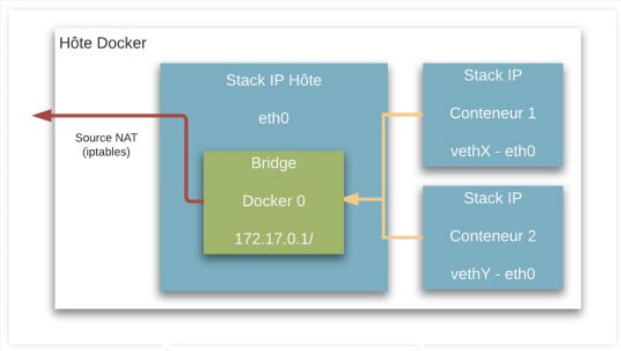
  Hôte : dossier

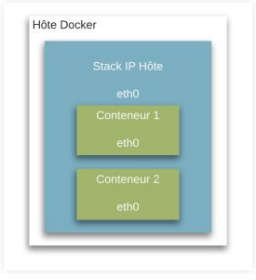


Réseau(Network) - Les types

- Bridge
- Host
- None
- Overlay

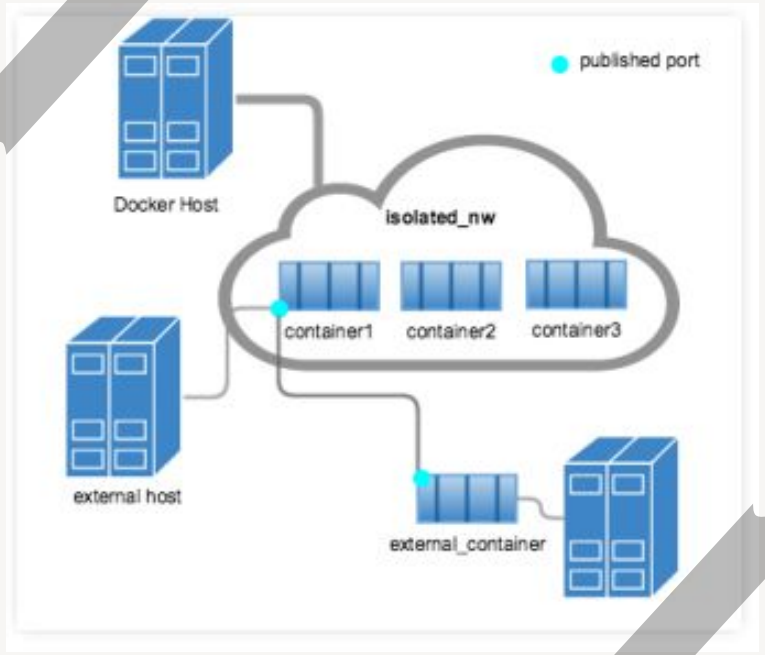






Réseau (Network)
Publication des ports (Publish)

- Dans le cas d'un réseau différent de l'hôte
- Les conteneurs ne sont pas accessible depuis l'extérieur
- Possibilité de publier des ports depuis l'hôte vers le conteneur (iptables)
- L'hôte sert de proxy au service

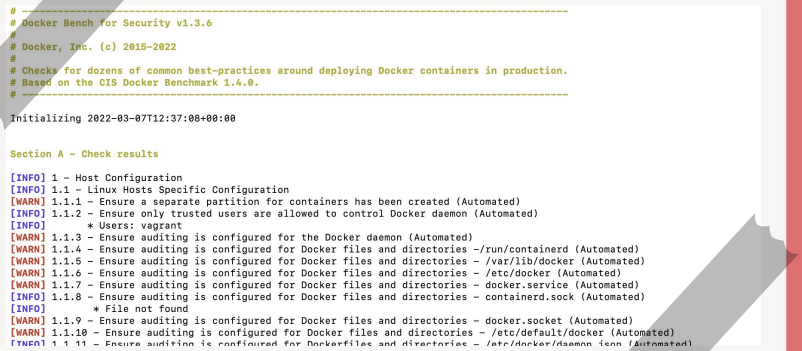


Le réseau (Network) - Les liaisons (Link)

- Les conteneurs d'un même réseau peuvent communiquer via IP
- Les liens permettent de lier deux conteneurs par nom
- Système de DNS rudimentaire (/etc/hosts)
- Remplacées par « discovery services »

Sécurité (Security)

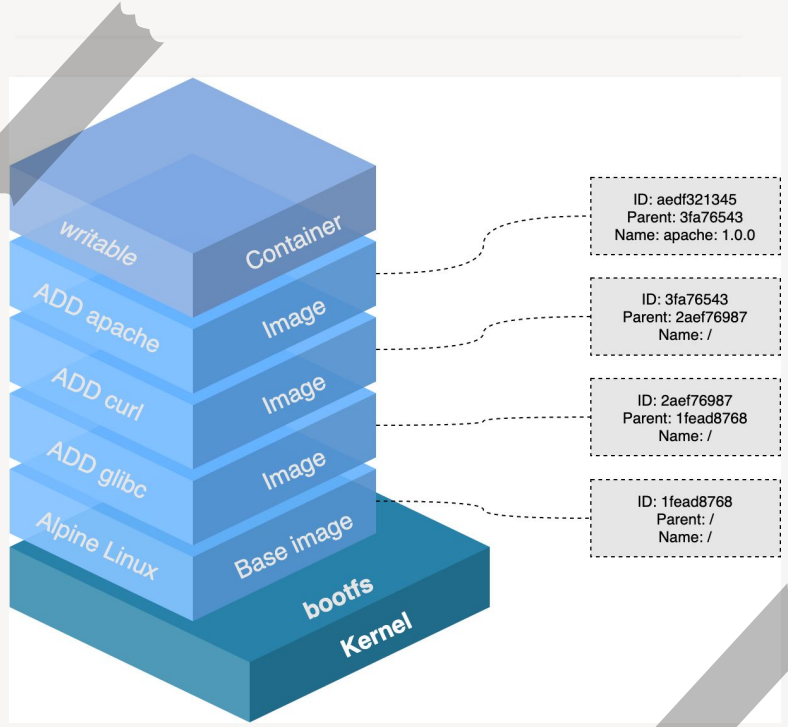
- Les conteneurs sont très sûr mais pour cela il faut respecter les bonnes pratiques de sécurité
- La remédiation peut être assez fastidieuse. Il est donc préférable de mettre en place à l'initialisation
- https://github.com/docker/docker-bench-security



```
# ------------------------------------------------------------------------------------------
# Docker Bench for Security v1.3.6
#
# Docker, Inc. (c) 2015-2022
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Based on the CIS Docker Benchmark 1.4.0.
# ------------------------------------------------------------------------------------------

Initializing 2022-03-07T12:37:08+00:00


Section A - Check results

[INFO] 1 - Host Configuration
[INFO] 1.1 - Linux Hosts Specific Configuration
[WARN] 1.1.1 - Ensure a separate partition for containers has been created (Automated)
[INFO] 1.1.2 - Ensure only trusted users are allowed to control Docker daemon (Automated)
[INFO]      * Users: vagrant
[WARN] 1.1.3 - Ensure auditing is configured for the Docker daemon (Automated)
[WARN] 1.1.4 - Ensure auditing is configured for Docker files and directories -/run/containerd (Automated)
[WARN] 1.1.5 - Ensure auditing is configured for Docker files and directories - /var/lib/docker (Automated)
[WARN] 1.1.6 - Ensure auditing is configured for Docker files and directories - /etc/docker (Automated)
[WARN] 1.1.7 - Ensure auditing is configured for Docker files and directories - docker.service (Automated)
[INFO] 1.1.8 - Ensure auditing is configured for Docker files and directories - containerd.sock (Automated)
[INFO]      * File not found
[WARN] 1.1.9 - Ensure auditing is configured for Docker files and directories - docker.socket (Automated)
[WARN] 1.1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker (Automated)
[INFO] 1.1.11 - Ensure auditing is configured for Dockerfiles and directories - /etc/docker/daemon.json (Automated)
```

Dockerfile - Définition

- Fichier de configuration d'une image docker
- Standardisation par rapport à l'export d'un conteneur en image
- Succession d'instructions. Chaque instruction représente une couche (layer)



Dockerfile - Les instructuctions 1/3

- FROM : socle de l'image.
- LABEL : défini des métadonnées de l'image. Auteur, version, ...
- COPY: Copie des fichiers dans l'image.
- ADD: Copie des fichiers dans l'image (accepte les URL)
- ENV : défini des variables d'environnement
- RUN : Commandes à exécuter

Dockerfile - Les instructructions 2/3

- EXPOSE: déclare les ports d'écoute du conteneur
- USER: défini l'utilisateur(groupe) qui exécutera le(s) processus
- VOLUME: déclare les points de montage des volumes
- WORKDIR : défini le répertoire de travail
- HEALTHCHECK : commande de test de bon fonctionnement
- ARG :

Dockerfile - Les instructructions 3/3

- CMD, SHELL, ENTRYPOINT, ONBUILD, ...

Dockerfile - Les bonnes pratiques 1/2

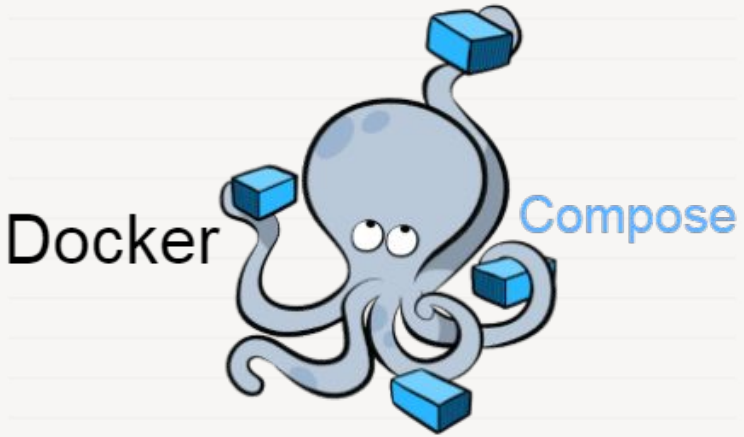
- FROM : Utiliser des images minimales et officielles
- FROM: Utiliser des images avec des versions spécifiques
- LABEL: Indiquer des méta-data sur l'image
- RUN: Installer le minimum de dépendances
- USER : Appliquer le principe de moindre privilèges (PoLP)
- EXPOSE : Exposer les ports

Dockerfile - Les bonnes pratiques 2/2

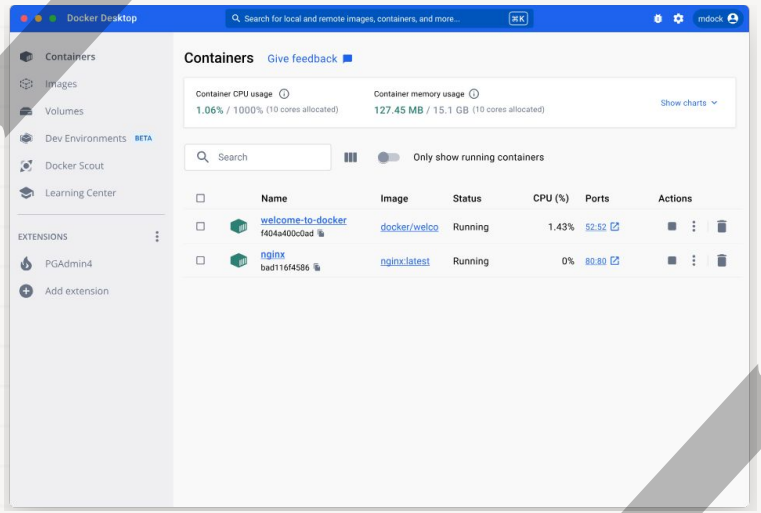
- Regrouper au maximum les instructions
- Utiliser les volumes
- Ne pas stocker les informations sensibles et locales
- Utiliser le multi-stage et les cibles pour chaque environnement
- .dockerignore : Contrôler les fichiers copiés dans l'image

Docker compose

- Permet d'exécuter plusieurs conteneurs sur Docker
- Simplifie le déploiement de stack : LAMP, supervision...
- docker-compose.yml



Docker Desktop



Les langages d'échanges de données

- JSON : JavaScript Object Notation. Avec séparateur et indentation uniquement visuelle.
- YAML : Yet Another Markup Language. Sans séparateur et une indentation (doubles espaces) obligatoire.
- Ils sont facile à lire ou à écrire pour des humains.
- Sont actuellement très utilisés comme langage pour des entrées et sorties

03

Observabilité

Définition et concepts

Du concept à l'expertise

Pour en savoir plus
https://elearning.lekpa.fr

Martin LEKPA
https://martin.lekpa.fr