



Architecture des Systèmes Open Source

Michael Marchal – Arnaud Tillieux



Présentation du cours

- Docker (Arnaud Tillieux)
 - Containers
 - Images
 - Volumes
- Licences open source (Michael Marchal)
 - Contraintes
 - Comparaison



Organisation du cours

- Cours de préparation au laboratoire
 - Cours théoriques
 - Devoirs pratiques
 - COVID-19
 - Moodle
 - Vidéo conférence / rediffusion
 - Teams / aide à distance
 - Limiter les contacts
 - Pause toutes les heures



Examen

- Examen théorique (+- 2h)
 - QCM en ligne Moodle
 - Concepts vu au cours



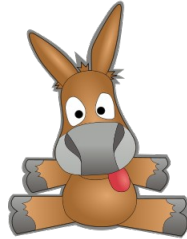
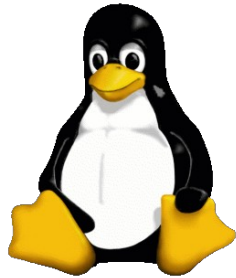
Qu'est ce qu'un système Open Source ?



Open source

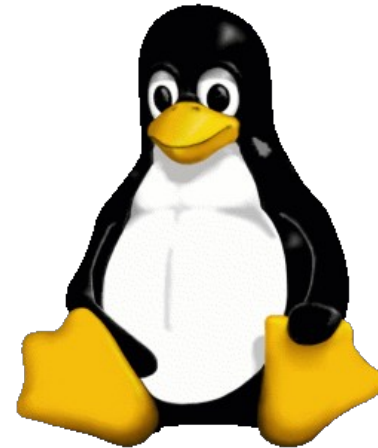
- Code source est disponible
- « Logiciel libre » [free software] désigne des logiciels qui respectent la liberté des utilisateurs. En gros, cela veut dire que les utilisateurs ont la liberté d'exécuter, copier, distribuer, étudier, modifier et améliorer ces logiciels. **Richard Stallman.**
- Plus d'informations dans la deuxième partie du cours

Exemple de logiciels Open Source

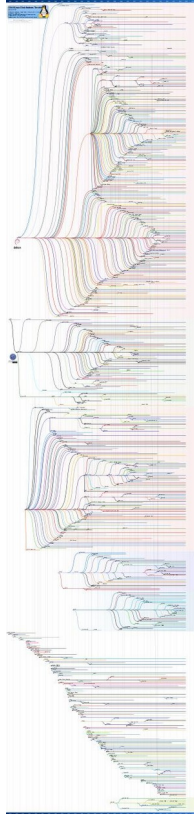


Linux

- Abus de langage pour le système d'exploitation
- Noyau
- GNU (outils open source)
- Linux + GNU = distribution Linux



Distributions Linux

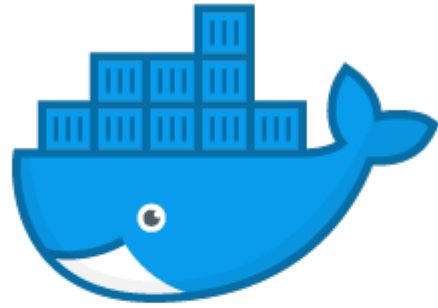


- Distributions « connues »
 - Grand public
 - Analyse réseau (hack)
Kali Linux
 - Développement
- Tout le monde peut créer sa distribution
- Impossible de créer une distribution avec un noyau « Windows »



Linux Rocks

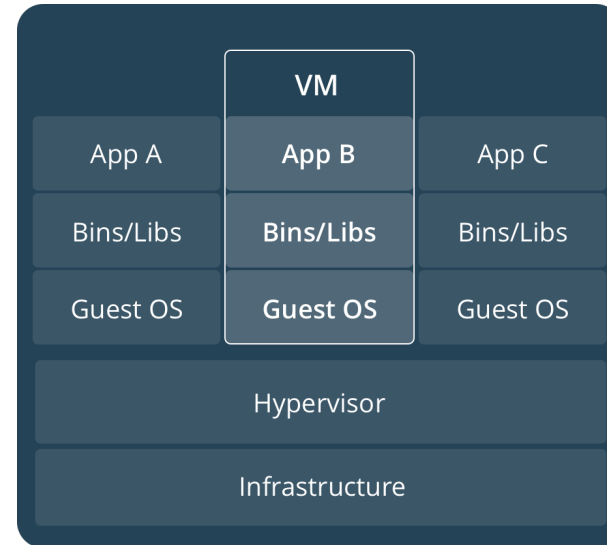
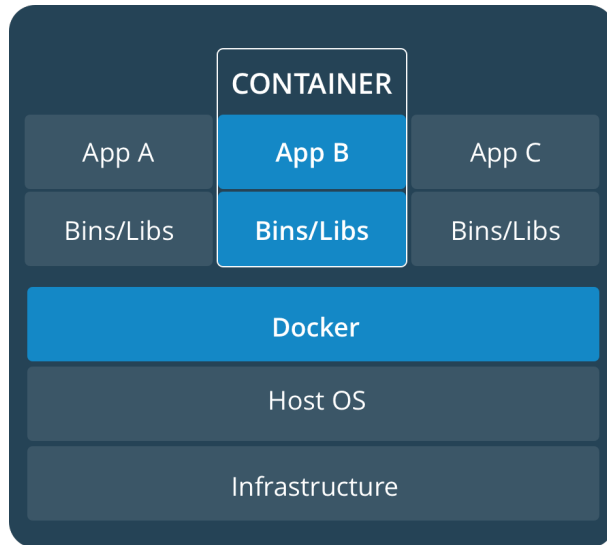
- 100 % Supercomputer sont sur Linux
- 90 % Pour le cloud servers (10 % Windows)
- +- 75 % Des smartphones (Android)
- 3 % Desktop, les habitudes du grand public...



docker

Qu'est ce que Docker ?

- Technologie de virtualisation d'applications



Qu'est ce que Docker ?

- Gain de performances / place
 - On ne « virtualise » que l'application
 - Pas besoin de virtualiser un OS en entier
 - Pas d'hyperviseur
 - Utilise beaucoup moins de RAM
 - ...
- Utilise le kernel Linux de l'OS hôte



Qu'est ce que Docker ?

- Flexible : On peut « virtualiser » n'importe quel application
- Léger : Exploite et partagent le noyau hôte beaucoup plus efficaces en termes de ressources système que la machine virtuelle
- Portable : Peut-être buildé localement puis déployé n'importe où
- Autonome : Les applications « virtualisées » sont autonomes et encapsulés, ce qui vous permet de remplacer ou d'améliorer l'un d'entre eux sans perturber les autres.
- Scalable : Gestion facile des répliquions



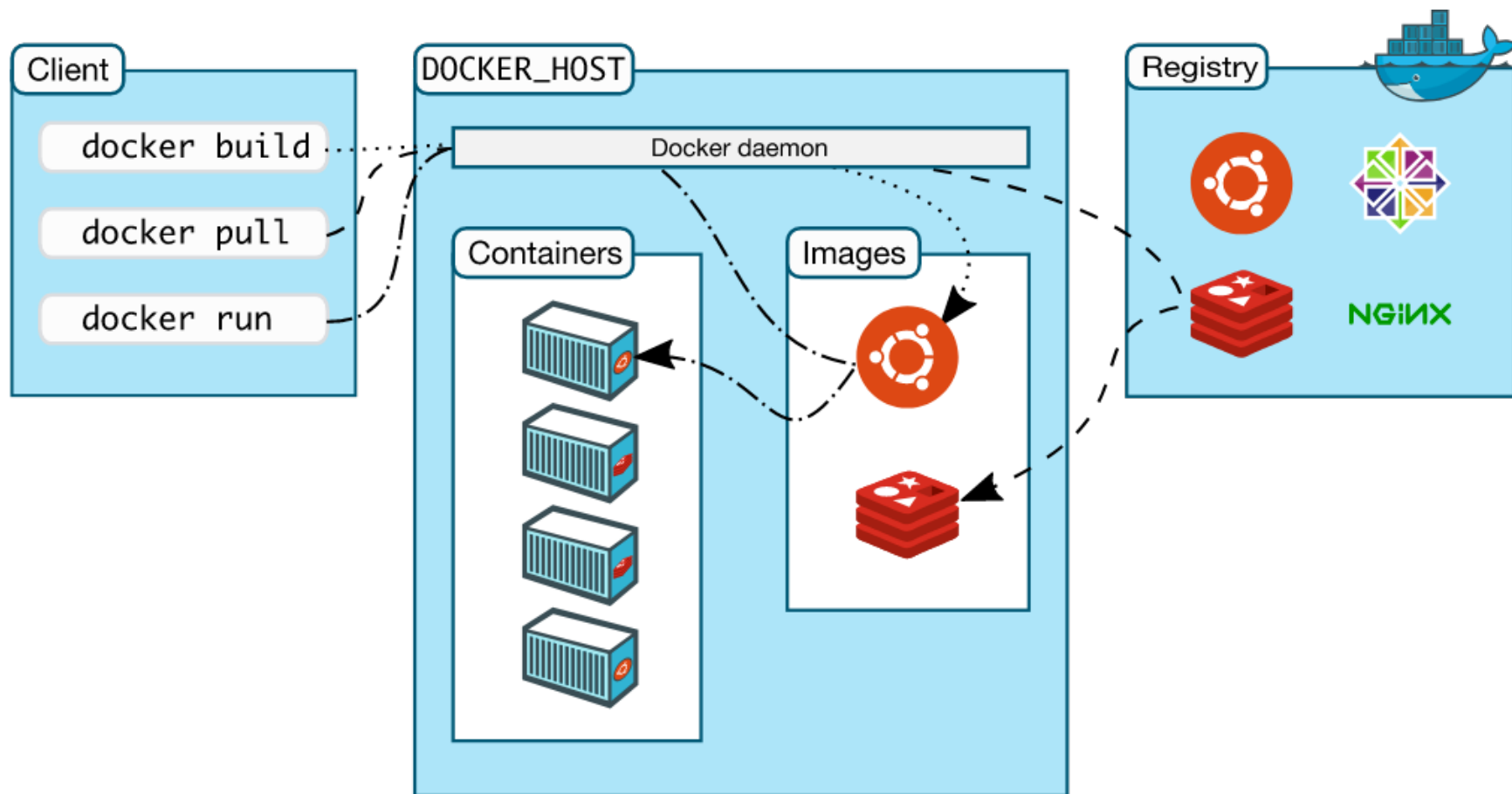
Docker : Les concepts clés

- Conteneur
 - Un processus en cours d'exécution
 - Instance d'une image
 - Encapsulé afin de le garder isolé de l'OS hôte et des autres conteneurs
 - Possède son propre système de fichier



Docker : Les concepts clés

- Images
 - Code source
 - Code exécutable (binarys)
 - Librairies
 - Dépendances
 - Autres objets du file system
- Registry (Docker Hub)
 - Upload des images créées
 - Rendre accessibles facilement





Docker installation

- Windows
 - <https://www.docker.com/get-started>
- MacOS
 - <https://www.docker.com/get-started>
- Linux
 - apt-get
 - pacman
 - ...

Docker commandes usuelles

- docker container
 - help (donne toutes les commandes)
 - ls (-a)
 - kill
 - create
- docker image
 - help
 - ls

Dockerfile

- Permet de créer une image Docker

```
# Use the official image as a parent image.
FROM node:current-slim

# Set the working directory.
WORKDIR /usr/src/app

# Copy the file from your host to your current location.
COPY package.json .

# Run the command inside your image filesystem.
RUN npm install

# Add metadata to the image to describe which port the container is listening on at runtime.
EXPOSE 8080

# Run the specified command within the container.
CMD [ "npm", "start" ]

# Copy the rest of your app's source code from your host to your image filesystem.
COPY . .
```