

# Docker- 3 jours

## Objectifs

- Appréhender la containerisation
- Créer et modifier des containers en local et en remote
- Être autonome sur l'exploitation d'une machine Docker
- Connaitre les fondamentaux de Docker (volumes / networks / reprises / etc)
- Connaitre docker compose et la théorie simple sur les orchestrateurs

## Public

Développeurs, Exploitants

## Prérequis

Connaissances lignes de commandes Unix (notions seulement)

## Méthodes pédagogiques

Approche orientée Dev & Ops

50% Théorie / 50% Pratique

Après chaque point vu en théorie, un TP de mise en pratique

## Qu'est-ce que Docker? Pourquoi utiliser docker ?

### Docker VS Machine Virtuelle

### L'exploitation de docker

### Containers, Images et Dockerfile

### Registry Docker

### Cycle de vie d'un container

### Création d'un container

- Choisir son image
- Le Dockerfile
- Compilation de l'image

### Lancement container

- Les commandes de base de Docker
- Docker instance
- Images
- Registry
- Containers
- La commande "run"
- Docker Clean

### Récupération de Docker

- Docker et ses montées de version
- Installation

### Les Volumes

- Le partage de répertoire depuis la machine hôte
- Le Volume Docker

### Network

- Les commandes
- Network "docker" : le bridge particulier

### Démos fonctionnalités

- Healthcheck
- Spring Boot
- Entrypoint / cmd surcharge
- Valeur par défaut
- Restart
- Load-Balancer

### Docker compose

- Versionning
- Notion de service
- Commandes de docker-compose
- Structure du docker-compose.yml
- Quelques balises courantes
- Les substitutions de variable
- Surcharges de docker compose

### Docker remote

### Docker en production

### Les orchestrateurs

- Docker Swarm
- Kubernetes
- Amazon ECS, Google Container Service (GCS), Microsoft Azure

### Y a t'il d'autres "Container Runtime"?

