



DOCKER FONDAMENTAUX SPECIF. LA POSTE BSCC



PRE-REQUIS

- ✗ Bonnes connaissances de développement
- ✗ Bonnes notions de systèmes et réseaux

OBJECTIFS

- ✗ Comprendre le principe et l'intérêt de l'orchestration de conteneurs
- ✗ Comprendre l'architecture et les concepts de Kubernetes
- ✗ Maîtriser les différents types de ressources Kubernetes
- ✗ Maîtriser la conception et le déploiement d'une application sur Kubernetes
- ✗ Savoir anticiper et corriger les problèmes d'exploitation courants

DUREE

- ✗ 3 jours

PROFIL DES STAGIAIRES

- ✗ Développeurs
- ✗ Administrateurs systèmes
- ✗ Culture DevOps préférable

PROGRAMME

Bref historique (Introduction)

- ✗ Historique de la conteneurisation
- ✗ Historique de Docker

Cloud et Conteneurisation

- ✗ Rappels sur le Cloud
- ✗ L'architecture des applications dans le cloud
- ✗ Docker comme levier du DevOps

Les fondamentaux de la conteneurisation

- ✗ La conteneurisation vs la virtualisation
- ✗ Origine de Docker
- ✗ Avantages de Docker
- ✗ Architecture de Docker
- ✗ Les namespaces et les cgroups
- ✗ Images, Conteneurs
- ✗ Les volumes et le réseau
- ✗ Les registres Docker
- ✗ Ecosystème de base
- ✗ Docker à La Poste
 - Les registres de la Poste
 - Organisation du poste de développement Docker
 - Quelles images utiliser
 - Comment mettre en place la parité Dev/Prod
 - Quels apports dans les images sous licence OpenShift
 - Quels apports dans les images privées LaPoste
 - Processus d'ajout d'image au registre de production
 - Registre Docker Hub vs Registre OpenShift
- ✗ Les Orchestrateurs
- ✗ Les initiatives CNCF et OCI

Docker par la pratique

Les commandes de base

- ✗ Lister les images Docker
- ✗ Redémarrer un conteneur arrêté



- ✗ Arrêter un conteneur
- ✗ Afficher les logs d'un conteneur
- ✗ Exécuter une commande à l'intérieur d'un conteneur
- ✗ [TP / Commandes de base](#)

La gestion des images

- ✗ Rappels sur la notion de couches et COW
- ✗ Caractéristiques des images
- ✗ Focus sur les notions de registres, repository, tags, comment nommer ses images
- ✗ Manipulation des images
 - Rechercher une image
 - Télécharger une image
 - Lister les images disponibles localement
 - Afficher les étapes de création de l'image
 - Supprimer une image localement
 - Afficher les différences entre un conteneur et son image
 - Créer une image depuis un conteneur modifié
- ✗ [TP / Manipulation des images et construction manuelle](#)

Les Dockerfiles

- ✗ Avantage et description
- ✗ Construction d'une image
- ✗ Comprendre la construction étape par étape
- ✗ Les commandes du Dockerfile
 - FROM, RUN, LABEL, ENV, COPY, ADD, WORKDIR EXPOSE, VOLUME, ENTRYPOINT, CMD
- ✗ [TP / Construction d'images avec Docker Build](#)

Les volumes

- ✗ Présentation de la problématique
- ✗ Les conteneurs immuables
- ✗ Les types de stockages
 - Volume
 - Binds
 - Tmpfs
- ✗ [TP / Volumes docker](#)

Les réseaux

- ✗ Présentation de la problématique
- ✗ Capacité des réseaux dans docker
- ✗ Fonctionnement des drivers

- BRIDGE
- HOST
- OVERLAY
- ✗ Quelles interactions avec l'hôte ?
- ✗ La commande docker network
- ✗ TP / Réseaux Docker

Les compositions de service

- ✗ Présentation de la problématique
- ✗ La réponse de Docker avec Docker Compose
- ✗ La notion de Service
- ✗ Introduction au fichier docker-compose.yaml
- ✗ Gestion des services dans docker-compose
- ✗ Gestion des volumes dans docker-compose
- ✗ Gestion des réseaux dans docker-compose
- ✗ Présentation des commandes de docker-compose (up, down, start, stop, rm, ...)
- ✗ TP / Docker-Compose

Introduction à Docker Swarm ou Kubernetes

- ✗ Présentation de la problématique
- ✗ Différences avec Docker Compose
- ✗ Brève introduction technique de Swarm ou Kubernetes
- ✗ Présentation d'OpenShift
 - Du CaaS au PaaS
 - Les processus de build et de déploiement d'OpenShift
 - Le processus de livraison OpenShift de LaPoste
 - Description technique du processus, que fait OpenShift dans les différentes étapes

Mise en pratique avancée

- ✗ Démarrage depuis une application micro-services utilisant des technologies PHP, Java, et JavaScript
- ✗ Atelier réparti selon l'assistance et ses compétences :
 - Création d'un groupe « Front »
 - Création d'un groupe « Spring/Java »
 - Création d'un groupe « Symfony/PHP » ou « Laravel/PHP »
 - Création du groupe « Base de données/MySQL »
- ✗ Création des images Docker (Dockerfile) pour chaque groupe
- ✗ Création de la composition des images pour le poste de développement
- ✗ Création de la composition pour tester les images de déploiement
- ✗ Selon le temps disponible :



- Création d'un groupe « Queue/Kafka »
- Création d'un groupe « Base de données Cassandra »
- ✗ Ajout des images dans la composition de développement et de déploiement

Conclusion

EXTRAIT SUPPORT DE COURS

Conteneurisation bref historique

- Unix v7 : Chroot
- FreeBSD Jails : Partition des ressources plus avancée
- Linux VServer : Linux Jails
- Solaris Containers : Virtualisation au niveau de l'OS qui utilise les zones Solaris
- Open VZ (Open Virtuozzo) : Virtualisation au niveau de l'OS qui repose sur des patch du noyau linux
- Process Containers : Contrôle des ressources utilisées par un groupe de processus (Groups)
- LXC : Linux Containers, la première implémentation de conteneurs pour linux, utilise les cgroups et les namespaces
- Warden : Le système de conteneurisation de Cloud Foundry, intègre sa propre implémentation de LXC
- LMCTFY Let Me Contain That For You, deviendra libcontainer et intégré à l'Open Container Foundation
- Docker : Utilisation de LXC, puis d'une librairie interne, et passage à libcontainer
- Arrivée sur le marché d'offres commerciales d'hébergement de conteneurs

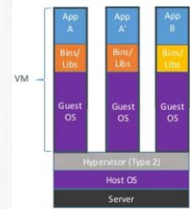
10/01/2019

Copyright Detailair

12

Rappel : la virtualisation

- Un serveur physique héberge un hyperviseur
- Ce dernier abstrait le matériel de son hôte et offre une interface matérielle « virtuelle »
- Des « machines » peuvent ainsi être instanciées, sur lesquelles un OS entier est installé



10/01/2019

Copyright Detailair

136

La conteneurisation

- S'appuie sur les fonctionnalités d'isolation du noyau linux
- Un gestionnaire de conteneurs est installé sur l'hôte
- Des « conteneurs » sont instanciés et sont isolés les uns des autres, bien qu'ils s'exécutent techniquement sur l'hôte



10/01/2019

Copyright Detailair

137

Pourquoi conteneuriser ?

- La solution adoptée en logistique : le conteneur
- Un format de chargement pouvant accueillir n'importe quel format de produit
- Ce format étant lui même adaptable sur n'importe quel moyen de transport

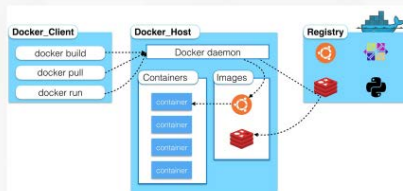


10/01/2019

Copyright Detailair

140

Docker, architecture générale



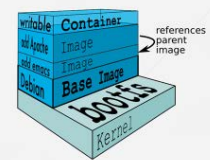
10/01/2019

Copyright Detailair

142

Les images docker

- Les images sont définies grâce à un langage déclaratif
- Une image est un ensemble de « couches » de fichiers (comme des calques)
- Chaque image peut s'appuyer sur une autre image comme couche de démarrage



10/01/2019

Copyright Detailair

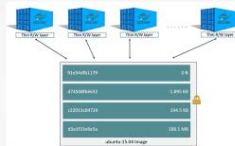
147

Images et conteneurs

Les Conteneurs docker

- Conséquences du système de couches de Docker :

- Économie de place, les couches de l'image ne sont pas copiées
- Un conteneur qui écrit beaucoup prend plus de place qu'un conteneur qui ne fait que de la lecture
- Les conteneurs démarrent très rapidement (la seule opération est la création de la couche supérieure)
- Données supprimées en même temps que les conteneurs



11/01/2019

Copyright Detailair

109

Les images Docker

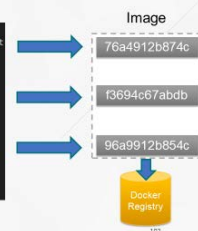
Chaque commande : FROM, COPY, RUN => Un layer

```
#Dockerfile de conteneur apache
FROM alim.sncf.fr:80/technu/ubuntu-16.04-its-amd64_mu_tools-ospit

#Install packages
RUN apt-get update \
    && apt-get install -y --no-install-recommends apache2-bin \
    && apt-get clean

#Creation user applicatif
RUN groupadd -g 1802 web000p1 \
    && useradd -m -s /home/web000p1 -g web000p1 -u 1802 web000p1

#Lancement et exposition serveur apache
CMD ["/home/web000p1/inst001/8001"]
```



18/01/2019

Copyright Detailair

102

Les commandes de Dockerfile

VOLUME <chemin>

- Déclare l'existence d'un volume externe à l'emplacement donné
- Le montage du volume sera résolu au moment du run
- Si des fichiers existent dans l'emplacement déclaré, docker les ajoutera au volume monté

```
FROM ubuntu
RUN mkdir /myvol
RUN echo "hello world" > /myvol/greeting
VOLUME /myvol
```

19/01/2019

Copyright Datallor

153

Gestion des images

• Attention

- Plusieurs tags peuvent pointer vers la même image (par ex. php:latest = php:7.3)
- Registry, repository et tag sont définis d'un coup par la commande docker tag

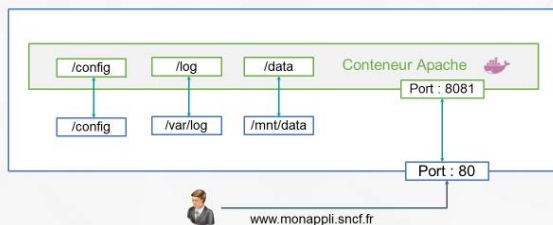


19/01/2019

Copyright Datallor

126

Les volumes et le réseau



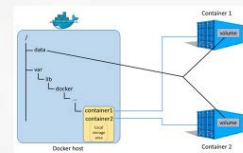
19/01/2019

Copyright Datallor

105

Volumes

- L'option --mount permet d'associer un espace de stockage au conteneur créé.
- Il existe 3 types d'espaces de stockage :
 - Les volumes
 - Les binds
 - Les montages tmpfs



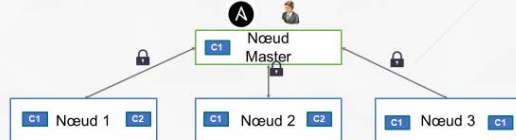
Copyright Datallor

10/01/2019

159

Les Orchestrateurs

- Gérer une ensemble de machines (VMs)
- Haute disponibilité et Elasticité
- Gestion unifiée d'un volume important de conteneurs
- Mutualiser les ressources systèmes et les environnements



11/01/2019

Copyright Datallor

109

Docker Trusted Registry

- Intégration Entreprise (AD, ...)
- Signature et certifications des images
- Gestion RBAC avancée
- Garbage collector
- Scans de sécurité



11/01/2019

Copyright Datallor

108