

DOCKER : MISE EN ŒUVRE

CODE STAGE : D0003

OBJECTIFS

Comprendre et savoir mettre en oeuvre Docker, et les produits de l'écosystème pour déployer des images tout en intégrant les contraintes de production.

DURÉE

2 jours

PUBLIC

Administrateurs, chefs de projet et toute personne souhaitant mettre en oeuvre Docker pour déployer ses applications.

PRÉ-REQUIS

Il est demandé aux participants de connaître les bases du système Unix/Linux.

PROGRAMME

1. Introduction

Présentation docker, principe, fonctionnalités

Besoins : packaging d'applications, déploiements rapides, coexistence de plusieurs versions d'une application sur un même serveur.

Les différentes éditions et leurs fonctionnalités : Docker Enterprise Edition, Docker Community Edition, Docker Cloud

2. Principe et architecture

Les conteneurs lxc.

Présentation de lxc : Linux containers, historique, principe de fonctionnement.

Les Cgroups. L'isolation de ressources, création d'un environnement utilisateur.

Positionnement par rapport aux autres solutions de virtualisation.

Travaux pratiques : vérification puis activation des cgroups.



Installation lxc.

Exemples de fonctionnalités simples des lxc :

vérification de la configuration du noyau,

configuration et création de conteneurs avec lxc-create.

Exemple de busybox en pratique.

Les systèmes de fichiers superposés.

Présentation de aufs.

Apports de Docker : Docker Engine pour créer et gérer des conteneurs Dockers.

Plate-formes supportées.

Lcode2utf('8217',0)écosystème Docker : Docker Machine, Docker Compose, Kitematic, Docker Swarm, Docker Registry

3. Installation et configuration

Prérequis techniques.

Tavaux pratiques : installation sur Linux

Mise en œuvre des scripts fournis par Docker pour lcode2utf('8217',0)installation.

Création dcode2utf('8217',0)un groupe Docker.

Mise en oeuvre en ligne de commande.

Démarrage dcode2utf('8217',0)un container simple.

Configuration de Docker et des containers.

Travaux pratiques :

démarrage automatique des containers, contrôle avec systemd,

limitation des ressources.

4. Gestion des images et des conteneurs

Création de nouvelles images.

Principe des DockerFile.

Travaux pratiques :

Utilisation de DockerFile pour créer des images personnalisées:

principales instructions (RUN, FROM, ENV, EXPOSE, etc ...)

Recommandations et bonne pratiques dcode2utf('8217',0)écriture de DockerFile.

Gestion des conteneurs : création, affichage, sauvegarde de lcode2utf('8217',0)état

Exemple de déploiement dcode2utf('8217',0)une application web avec des containers.

Présentation du Docker Hub.

Publication d'images vers un registry.

5. Volumes de données

Initialisation des volumes de données lors de la création d'un container.

Travaux pratiques:

ajout de volumes de données, contrôle avec la commande `Docker inspect`;

Sauvegarde, migration, restauration de volumes

Création de conteneurs de volumes de données

6. Administration

Présentation des outils Swarm, Compose, Docker Machine

Fonctionnalités de swarm : cluster Docker, principe du mode swarm, load balancing.

Démonstrations de load balancing.

Applications de multi-containers avec Compose

Méthode d'administration des containers en production.

Orchestration avec Docker Machine

Configuration réseau et sécurité dans Docker

Présentation des plugins Docker

