

## Formation Docker : Approfondissement

<b>Durée :</b>	2 jours
<b>Public :</b>	Utilisateurs Docker - Administrateurs Systèmes et Réseaux
<b>Pré-requis :</b>	Avoir suivi le cours "Docker initiation" ou posséder les connaissances équivalentes
<b>Objectifs :</b>	Appréhender et configurer les fonctionnalités avancées de Docker - Maitriser la sécurité de sa plateforme Docker - Maitriser le cycle de vie des images - Administrer sa registry privée - Maitriser les configurations multi-hôtes avec Swarm - Déployer des services et stacks applicatives - Maîtriser les bonnes pratiques
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	OUT101040-F
<b>Note de satisfaction des participants:</b>	4,79 / 5

### Introduction

Rappel des concepts et objets fondamentaux  
Registres, images, conteneurs, volumes...

### Manipuler le démon Docker

Administrer le démon docker : environnement, rôle, configuration, logs  
Expliquer le fichier de configuration  
Configurer l'architecture client-serveur de Docker  
Manipuler les fonctionnalités avancées et commandes importantes  
Illustrer le fonctionnement de l'API REST  
Observer les options de socket pour les accès en réseau.  
Examiner les "storage driver"  
Configurer des connections derrière un proxy

**Atelier pratique : Installation, configuration et démarrage du démon Docker. Préparation accès réseau client/hôte. Personnalisation de la configuration Docker. Manipulations de l'API**

### Optimiser la conception d'images

Rappels sur la conception des "Dockerfiles"  
Développer une conception et une gestion fine du cycle de vie des images  
Justifier la gestion du cache avec les "layers"  
Rompre avec le système d'idempotence

Construite une image en "multi-stage builds"

Contrôler l'état applicatif dans l'image

Identifier les projets communautaires incontournables : analyse, métriques, reverse-proxy, sécurité

**Atelier pratique : Création d'images via le multistage-build. Analyse des layers, optimisation du cache. Utilisation de la registry privée. Utilisation d'images de la communauté**

### **Restructurer docker-compose**

Rappels sur les concepts

Assembler les ressources (les services, les réseaux, les volumes)

Intégrer intelligemment les variables d'environnement

Adapter les contextes de build : "Dockerfile"

Résoudre les dépendances entre services

Mettre en place un contrôle de l'exécution

Industrialiser une stack docker-compose

**Atelier pratique : Conception d'une application à travers docker-compose et dockerfile. Industrialisation de la création de stack docker-compose avec l'outil cookiecutter.**

### **Construire un environnement de production : clustering, orchestration et monitoring**

Définir les enjeux d'un orchestrateur

Expliquer Swarm et ses fonctionnalités

Illustrer la notion de nodes (manager, worker)

Examiner le réseau et les "topologies mesh"

**Atelier pratique : Installation d'un cluster swarm (avec docker-machine). Installation d'outils de visualisation et monitoring de l'infra. Démarrage d'un service et mise à l'échelle au sein du cluster.**

### **Gérer le cycle de vie approfondi des conteneurs**

Consolider les ressources à travers les commandes docker update

Délimiter les domaines d'exécutions des conteneurs (les labels, placements par node)

Utiliser les mécanismes de "rolling update" et "rollback"

**Atelier pratique: Sur la base de l'infrastructure précédemment mise en place, nous allons effectuer des montées de version, puis effectuer des rollbacks. Optimisation de l'emplacement des conteneurs.**

### **Appréhender la sécurité pour Docker**

Sécuriser la plateforme avec TLS/SSL (client, notes, registry)

Identifier les risques : noyau, service Docker, containers, déni de service, accès réseau

Utiliser des mécanismes de protection : "subnet" spécifique par application, limitations de ressources par les "cgroups", restrictions des droits d'accès sur les sockets, politique de sécurité des containers

Examiner les "events" docker

Fiabiliser les images déployées dans Docker : présentation de "Content Trust" pour signer les images

**Atelier pratique : Mise en évidence de failles de sécurité et des bonnes pratiques à adopter. Isolations et limitations des ressources. Création et déploiement d'images signées.**