OSD
OPEN DIGITAL SERVICE

Scannez-moi

WHATSAPP
+213 674 05 38 95

SITE WEB
osdigitalis.com

FICHE FORMATION · OSD

Conteneuriser des applications avec Docker

Images, conteneurs, volumes, réseaux et Docker Compose, jusqu'au déploiement

20 heures | 10 séances | Linux | Docker Compose

Une formation pratique pour **empaqueter, lancer et déployer des applications dans des conteneurs** avec Docker et Docker Compose, de la première commande à une pile complète prête pour la production.

20 h	10 × 2 h	intermédiaire	labos Linux
DURÉE	SÉANCES	NIVEAU	FORMAT

Description

Docker a changé la façon de livrer les logiciels : une application et toutes ses dépendances tiennent dans une image qui fonctionne à l'identique sur le poste du développeur, en test et en production. C'est aujourd'hui la base de l'intégration continue, du cloud et de l'orchestration.

Les participants conteneurisent **une application réelle à plusieurs services** (API, base de données, front-end, proxy inverse), écrivent leurs propres Dockerfile, gèrent les données et les réseaux, puis déploient la pile complète avec **Docker Compose** derrière **Traefik**.

Public visé

- Développeurs qui veulent livrer leurs applications simplement
- Administrateurs systèmes et futurs DevOps
- Étudiants en systèmes, réseaux ou développement

Prérequis

- Bases de la ligne de commande Linux (fichiers, droits, éditeur de texte)
- Notions de réseau (adresse IP, port)
- Un PC avec 8 Go de RAM et Docker installé (ou une VM Linux)

À l'issue de la formation, vous saurez

☐	expliquer la différence entre **conteneur** et **machine virtuelle** ;
☐	lancer, inspecter et dépanner des **conteneurs** ;
☐	écrire des **Dockerfile** optimisés (couches, multi-étapes, images légères) ;
☐	persister les données avec les **volumes** et relier les services par des **réseaux** ;
☐	décrire une pile complète avec **Docker Compose** ;
☐	publier des images sur un **registre** et appliquer les bonnes pratiques de **sécurité**.

Méthode pédagogique

- Courts exposés illustrés, puis **ateliers en ligne de commande** à chaque séance.
- Une **application fil rouge** à plusieurs services, conteneurisée pas à pas.
- Pannes provoquées volontairement pour apprendre à lire journaux et inspections.
- Bonnes pratiques de production : images minimales, utilisateur non root, secrets, limites de ressources.

Programme — 10 séances

1 **Pourquoi les conteneurs** 2 h

Du serveur physique au conteneur, conteneur et machine virtuelle, architecture de Docker, installation et premier conteneur.

`conteneur vs VM` `docker run` `installation`

2 **Images et conteneurs** 2 h

Registres, images et couches, cycle de vie d'un conteneur, journaux, exec, inspection et nettoyage.

`images` `logs · exec` `Docker Hub`

3 **Écrire un Dockerfile** 2 h

Instructions principales, contexte de construction, cache des couches, construire et étiqueter sa première image.

`Dockerfile` `build` `cache`

4 **Optimiser les images** 2 h

Constructions multi-étapes, images légères (Alpine, distroless), .dockerignore, utilisateur non root.

`multi-stage` `Alpine` `non-root`

5 **Données persistantes** 2 h

Volumes nommés et montages liés, sauvegarde et restauration de données, base de données en conteneur.

`volumes` `bind mounts` `sauvegarde`

6 **Réseaux Docker** 2 h

Réseaux bridge, résolution des noms entre conteneurs, ports publiés, isolation des services.

`bridge` `DNS interne` `ports`

7 **Docker Compose** 2 h

Décrire une pile multi-services, variables d'environnement, dépendances, démarrage et mise à jour.

`compose.yaml` `.env` `depends_on`

8 **Proxy inverse et HTTPS** 2 h

Exposer plusieurs applications derrière Traefik, routage par nom de domaine, certificats automatiques.

`Traefik` `HTTPS` `labels`

9 **Registres, sécurité et supervision** 2 h

Publier sur un registre privé, analyser les vulnérabilités, gérer les secrets, limiter les ressources, superviser.

`registre` `scan` `limites`

10 **Projet final et évaluation** 2 h

Conteneuriser et déployer une application complète à partir d'un cahier des charges, puis soutenance.

`projet` `évaluation /20`

Évaluation et validation

- Ateliers vérifiés à chaque séance.
- **Projet final** : application multi-services conteneurisée et déployée avec Compose, présentée et notée sur 20.
- Attestation de formation délivrée par OSD — Open Digital Service.

Modalités

Durée	20 heures, en 10 séances de 2 heures
Format	présentiel en salle informatique, ou intra-entreprise / établissement
Effectif	groupes de 8 à 20 participants
Matériel	un poste par participant avec Docker (Docker Desktop ou Linux) et VS Code
Supports remis	fiches de séance, aide-mémoire Docker et Compose, dépôts d'exercices
Formateur	Mohammed Slimani — architecte DevOps, ancien administrateur systèmes et responsable informatique

Intéressé(e) par cette formation ?

Programme adaptable (Podman, Kubernetes en introduction, volume horaire, public). Contact : contact@osdigitalis.com · WhatsApp +213 674 05 38 95 · osdigitalis.com

Open Digital Service

Mohammed Slimani — Formateur & consultant

+213 (0)6 74 05 38 95 · contact@osdigitalis.com · msslimani@osdigitalis.com

osdigitalis.com · LinkedIn : in/mohammed-slimani-osdigitalis