

Formation

Docker & déploiement

Formation en présentiel et en classe virtuelle à la demande. Apprenez à containeriser vos applications, gérer vos images et orchestrer des environnements de déploiement modernes avec Docker et les outils associés.

Formation 100% pratique

Exercices & cas d'usage

Formateurs expérimentés

3 jours

21 h

Format

Atelier
pratique

Modalité

Présentiel ou
en ligne

Niveau

Intermédiaire

INSCRIPTION / RÉSERVATION



Je m'inscris
maintenant



OBJECTIFS PÉDAGOGIQUES

- Comprendre les concepts de base des conteneurs et l'architecture de Docker.
- Savoir construire, versionner et publier des images Docker réutilisables.
- Maîtriser la mise en place d'environnements multi-conteneurs avec Docker Compose pour le développement et la pré-production.
- Préparer et industrialiser le déploiement d'applications containerisées dans des environnements on-premise ou cloud (CI/CD).

PUBLIC CIBLE

- Développeurs souhaitant containeriser et déployer leurs applications.
- Ingénieurs systèmes / Ops / DevOps impliqués dans les chaînes de déploiement.
- Architectes techniques et chefs de projet IT souhaitant comprendre les impacts des conteneurs sur les architectures de déploiement.

PRÉREQUIS

- Bonnes notions de développement (Java, .NET, Node.js, Python ou équivalent).
- Connaissances de base en ligne de commande et environnement Linux.
- Notions de réseaux (ports, IP, DNS) et de gestion de code source (Git) sont un plus.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Introduction aux conteneurs & à Docker

Comprendre les conteneurs

- Virtualisation traditionnelle vs conteneurisation.
- Concepts clés : image, conteneur, registre, orchestrateur, isolation.
- Cas d'usage : développement, intégration continue, déploiement continu, microservices.

Découverte de Docker

- Architecture Docker : daemon, client, Docker Engine.
- Installation de Docker Desktop / Docker Engine.
- Premiers conteneurs : docker run, docker ps, logs.
- Découverte de Docker Hub et des images officielles.

Gestion des images Docker

Dockerfiles & bonnes pratiques

- Structure d'un Dockerfile : FROM, RUN, COPY, CMD, ENTRYPOINT.
- Construction d'une image applicative : build, tag, push, pull.
- Optimisation : réduction de taille, couches, images minimales (Alpine, distroless).

Images multi-stage & variantes

- Multi-stage builds pour séparer compilation et exécution.
- Gestion des tags, versions, stratégies de nommage.
- Nettoyage des images et gestion de l'espace disque.

Gestion des conteneurs & réseaux Docker

Cycle de vie des conteneurs

- Création, démarrage, arrêt, redémarrage, suppression de conteneurs.
- Options d'exécution : détaché, interactif, redémarrage automatique.
- Accès aux logs, shell dans un conteneur, inspection.

Réseaux Docker

- Types de réseaux : bridge, host, none, overlay (notions).
- Exposition de ports, mapping, communication entre conteneurs.
- DNS interne Docker, alias, inspection des réseaux.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Volumes, persistance & configuration

Persistance de données

- Volumes Docker vs bind mounts.
- Gestion des volumes, sauvegarde et restauration.
- Exemple : base de données containerisée avec persistance des données.

Configuration & secrets

- Variables d'environnement, fichiers de configuration.
- Injection de configuration à l'exécution.
- Premières bonnes pratiques de gestion de secrets.

Docker Compose & environnements multi-conteneurs

Introduction à Docker Compose

- Format YAML, services, réseaux, volumes, dépendances.
- Définir un environnement complet : front, API, base de données, outils.
- Commandes clés : up, down, logs, scale.

Scénarios de déploiement avec Compose

- Fichiers de surcharge (override), profils, environnements.
- Utilisation de Compose en développement, test, pré-production.
- Bonnes pratiques d'organisation des fichiers et stacks.

Registries, CI/CD & déploiement

Registries Docker

- Docker Hub, registries privés, registries cloud (ACR, ECR, etc.).
- Authentification, droits d'accès, règles de rétention.
- Stratégies de nommage et de versionnage des images.

Chaînes CI/CD orientées conteneurs

- Principes : build, test, scan, push, déploiement automatisé.
- Exemple de pipeline : build d'image, tests automatisés, push vers un registre distant.
- Intégration avec des plateformes de déploiement (serveurs, PaaS, orchestrateurs).

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Sécurité, supervision & bonnes pratiques

Sécurité des images et des conteneurs

- Images de confiance, scans de vulnérabilités, mise à jour.
- Principes de moindre privilège, rootless containers (notions).

Supervision & dépannage

- Monitoring de base : métriques, logs, inspection des conteneurs et réseaux.
- Analyse de problèmes fréquents (ports, DNS, performance).

Mise en pratique de bout en bout

- Mise en conteneur d'une application existante et déploiement complet en environnement de test.
- Synthèse des bonnes pratiques et recommandations pour les projets des participants.