

Docker, administration avancée

2 j (14 heures)

Ref : DOCK

Public

Administrateurs système, ingénieurs DevOps, développeurs ou architectes ayant déjà utilisé Docker et souhaitant approfondir la gestion et l'industrialisation des conteneurs en environnement professionnel.

Pré-requis

Connaissance de base de Docker (ligne de commande, images, conteneurs).

Maîtrise de l'environnement Linux et des réseaux TCP/IP recommandée.

Moyens pédagogiques

Mise à disposition d'un poste de travail sur nos formations en Présentiel.

Mise à disposition de nos environnements de visio sur nos formations en Distanciel

Remise d'une documentation pédagogique numérique pendant la formation

La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions Dans le cas d'une formation sur site Entreprise, le client s'engage à avoir toutes les ressources pédagogiques nécessaires (salle, équipements, accès internet, TV ou Paperboard...) au bon déroulement de l'action de formation conformément aux prérequis indiqués dans le programme de formation

Modalités de suivi et d'évaluation

Auto-positionnement des stagiaires avant la formation Émargement des stagiaires et formateur par 1/2 journée

Exercices de mise en pratique ou quiz de connaissances tout au long de la formation permettant de mesurer la progression des stagiaires

Auto-évaluation des acquis de la formation par les stagiaires

Questionnaire de satisfaction à chaud et à froid à l'issue de la formation

Objectifs

Gérer des images complexes et des volumes persistants

Configurer des réseaux Docker pour des architectures multi-conteneurs

Orchestrer et automatiser les déploiements avec Docker Compose et Swarm

Mettre en œuvre les bonnes pratiques de sécurité et de monitoring

Optimiser la taille, la performance et la maintenabilité des images

Intégrer Docker dans une logique DevOps et CI/CD

Programme détaillé

JOUR 1

01 Gestion avancée des images Docker

- Création d'images optimisées et multi-étapes (multi-stage builds)
- Réduction de la taille des images via les bonnes pratiques (alpine, clean)
- Utilisation de fichiers Dockerfile complexes avec variables et contextes
- Gestion du cache et des layers pour des builds performants
- Travaux pratiques : Construire une image multi-étapes optimisée pour le déploiement

02 VOLUMES, PERSISTANCE ET PLUGINS DE STOCKAGE

- Création et gestion de volumes nommés
 - Montage de répertoires de données sur l'hôte
 - Utilisation de plugins de stockage tiers (ex : NFS, CIFS)
 - Gestion de la persistance dans un environnement éphémère
 - Travaux pratiques : Démarrer une base de données conteneurisée avec volume persistant
-

06 Intégration DevOps et CI/CD

- Utilisation de Docker dans une chaîne CI/CD (GitLab CI, GitHub Actions)
 - Build et push automatisé d'images vers des registries privées
 - Scan de vulnérabilités avec Docker Scout ou Trivy
 - Présentation d'extensions IA (ex : création de Dockerfile assistée)
 - Travaux pratiques : Automatiser le build/push d'une image avec GitHub Actions
-

04 ORCHESTRATION AVEC DOCKER COMPOSE ET SWARM

- Écriture de fichiers docker-compose.yml pour déploiement multi-services
- Variables d'environnement, fichiers .env et gestion des secrets
- Utilisation de Docker Swarm : init, nodes, stacks
- Comparatif rapide avec Kubernetes (logique de montée en compétence)
- Travaux pratiques : Déployer un stack applicatif complet avec Compose et Swarm

05 Sécurité, supervision et logs

- Meilleures pratiques de sécurité dans les Dockerfile
- Gestion des droits, utilisateur non-root, capabilities
- Supervision avec Prometheus, Grafana ou outils intégrés
- Gestion des logs centralisés et audit des conteneurs
- Travaux pratiques : Analyse de logs d'un service planté et mise en place de règles de sécurité de base