

<b>Public</b>	Administrateurs systèmes en charge du déploiement d'applications.
<b>Durée</b>	3 jours - 21 heures
<b>Pré-requis</b>	Connaissances de base de l'administration Linux.
<b>Objectifs</b>	<p>Comprendre les concepts fondamentaux de Docker</p> <p>Construire et exécuter des conteneurs simples</p> <p>Manipuler les commandes de base de Docker</p> <p>Construire des images Docker pour des applications Python d'IA/ML</p> <p>Gérer les volumes pour persister des données</p> <p>Utiliser Docker Compose</p> <p>Intégrer Docker dans un workflow CI/CD avec Git</p> <p>Sécuriser et optimiser les images Docker</p> <p>Déployer les conteneurs sur un serveur distant</p>
<b>Méthodes pédagogiques</b>	<p>Pour bien préparer la formation, le stagiaire remplit une évaluation de positionnement et fixe ses objectifs à travers un questionnaire.</p> <p>La formation est délivrée en présentiel ou distanciel (e-learning, classe virtuelle, présentiel et à distance). Le formateur alterne entre méthodes démonstratives, interrogatives et actives (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.</p> <p>Cette formation est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par Audit Conseil Formation.</p>
<b>Moyens techniques</b>	<p>1 poste de travail complet par personne</p> <p>De nombreux exercices d'application</p> <p>Mise en place d'ateliers pratiques</p> <p>Remise d'un support de cours</p> <p>Passage de certification(s) dans le cadre du CPF</p> <p>Remise d'une attestation de stage</p>
<b>Modalité d'évaluation des acquis</b>	<p>Evaluation des besoins et objectifs en pré et post formation</p> <p>Evaluation technique des connaissances en pré et post formation</p> <p>Evaluation générale du stage</p>
<b>Délai d'accès</b>	L'inscription à cette formation est possible jusqu'à 5 jours ouvrés avant le début de la session
<b>Accessibilité handicapés</b>	Au centre d'affaires ELITE partenaire d'ACF à 20 m. Guide d'accessibilité à l'accueil.

## JOUR 1 : INTRODUCTION À DOCKER & CONTENEURISATION

### 1. POURQUOI DOCKER ?

- Différences avec les machines virtuelles
- Isolation, portabilité, reproductibilité
- Cas d'usage en IA/ML

### 2. ARCHITECTURE DOCKER

- Docker Engine, Docker Daemon, CLI, Docker Hub

### 3. COMMANDES DE BASE

- docker version, docker info
- docker run, docker ps, docker images, docker rm, docker rmi, docker stop, docker start
- Différence entre docker run et docker exec

## 4. TRAVAIL AVEC LES IMAGES

- Téléchargement d'images (docker pull)
- Création d'images (docker build)

## 5. EXERCICES

- Installation de Docker sur CentOS/Ubuntu
- Lancer un conteneur Python : `docker run -it --rm python:3.9 bash`
- Créer un premier Dockerfile : Construire une image de base pour un script Python simple. Exécuter le conteneur.
- Gestion des conteneurs et images : Lister, supprimer des conteneurs et images

## 6. CAS PRATIQUE IA/ML

- Créer un Dockerfile minimal pour exécuter un script de post-traitement en Python 3.9.

# JOUR 2 : DOCKER POUR L'IA/ML ET GESTION DES VOLUMES

## 1. OPTIMISATION DES IMAGES DOCKER POUR PYTHON ET IA/ML

- Choix des images de base (python:3.9-slim, jupyter/scipy-notebook)
- Réduction du poids des images
- Gestion du cache avec `pip install --no-cache-dir`

## 2. VOLUMES ET PERSISTANCE DES DONNÉES

- `docker volume create`
- `docker run -v $(pwd)/data:/app/data`

## 3. UTILISATION DE DOCKER COMPOSE

- Définition d'un `docker-compose.yml` pour orchestrer plusieurs services
- Exemple : Conteneur Python + Conteneur PostgreSQL

## 4. EXERCICES

- Créer une image Docker optimisée pour un projet IA/ML avec numpy, pandas, scikit-learn.
- Créer un volume Docker et lire/écrire des données d'entraînement.
- Docker Compose pour une app IA/ML : Un service Flask pour exposer un modèle ML, un conteneur PostgreSQL pour stocker les résultats

## 5. CAS PRATIQUE AVANCÉ

- Déployer une API Flask dans un conteneur qui charge un modèle entraîné en IA/ML.

# JOUR 3 : CI/CD AVEC DOCKER ET SÉCURITÉ DES CONTENEURS

## 1. DOCKER ET GIT : VERSIONNER LES DOCKERFILES

- Bonnes pratiques pour commit/push des Dockerfiles
- Automatisation des builds avec GitHub Actions ou Jenkins

## 2. DOCKER ET CI/CD

- Construire et publier une image automatiquement sur Docker Hub/Nexus
- docker build -t monrepo/ml-image:latest .
- docker push monrepo/ml-image:latest

## 3. SÉCURITÉ DES CONTENEURS

- Éviter d'utiliser root dans les conteneurs
- Scanner une image avec docker scan
- Limiter les ressources (--memory, --cpu)

## 4. DÉPLOIEMENT ET ORCHESTRATION

- Introduction à Kubernetes Kind
- Déploiement d'un pod avec un conteneur IA/ML

## 5. EXERCICES

- Configurer un pipeline CI/CD Docker avec GitHub Actions
- Scanner une image Docker pour détecter des vulnérabilités
- Limiter l'usage des ressources d'un conteneur IA/ML

## 6. CAS PRATIQUE FINAL

- Créer un pipeline CI/CD complet : Build de l'image, push vers un registry Docker privé, déploiement automatique sur un serveur

---

## NOUS CONTACTER

### Siège social

16, ALLÉE FRANÇOIS VILLON  
38130 ÉCHIROLLES

### Téléphone

04 76 23 20 50 - 06 81 73 19 35

### Centre de formation

87, RUE GÉNÉRAL MANGIN  
38000 GRENOBLE

### E-mail

contact@audit-conseil-formation.com

Suivez-nous sur les réseaux sociaux, rejoignez la communauté !



ACF Audit Conseil Formation



@ACF\_Formation



ACFauditconseilformation