

# DevOps pratique : Docker, CI/CD et déploiement d'applications

## Introduction à la formation DevOps pratique

Présentation des objectifs de la formation.

Comprendre la différence entre DevOps comme culture et DevOps comme pratique technique.

Identifier les étapes d'un cycle de livraison moderne.

Comprendre les problèmes classiques : configuration manuelle, environnements incohérents, déploiements risqués et manque de visibilité.

## Rappel des principes DevOps

Collaboration entre développement, opérations et assurance qualité.

Automatisation du cycle de livraison.

Feedback rapide.

Intégration continue.

Déploiement continu.

Amélioration continue.

## Préparation d'une application pour la livraison

Organisation du code source.

Configuration par environnement.

Variables d'environnement.

Gestion des dépendances.

Tests automatisés.

Build reproductible.

Atelier pratique :

Préparer une application existante pour l'automatisation.

Identifier les éléments à externaliser.

Valider le build local.

## Docker pour la livraison applicative

Rappel sur les images et les conteneurs.

Création d'un Dockerfile.

Bonnes pratiques de conteneurisation.

Build multi-stage.

Ports, variables d'environnement et logs.

Atelier pratique :

Créer une image Docker pour une application.

Exécuter l'application dans un conteneur.

Modifier la configuration sans modifier le code.

## Environnement complet avec Docker Compose

Pourquoi utiliser Docker Compose.

Application et base de données.

Réseaux internes.

Volumes persistants.

Fichier .env.

Health checks.

Dépendances entre services.

Atelier pratique :

Créer un environnement complet avec Docker Compose.

Connecter l'application à une base de données.

Valider le fonctionnement de l'environnement.

Diagnostiquer un problème de configuration.

## Introduction au CI/CD

Comprendre l'intégration continue.

Comprendre la livraison continue.

Différence entre build, test, artefact, image et déploiement.

Structure d'un pipeline.

Déclencheurs : push, merge request, pull request, tags et branches.

## Création d'un pipeline CI/CD

Présentation d'un outil de pipeline : GitLab CI/CD, GitHub Actions ou Azure DevOps.

Structure d'un fichier YAML.

Étapes du pipeline.

Variables du pipeline.

Exécution automatisée du build.

Exécution automatisée des tests.

### Atelier pratique :

Créer un pipeline d'intégration continue.

Compiler automatiquement l'application.

Exécuter les tests.

Faire échouer et corriger une étape du pipeline.

## CI/CD avec Docker

Construire une image Docker dans le pipeline.

Taguer une image.

Publier une image dans un registre.

Gérer les variables et les secrets.

Séparer les environnements.

### Atelier pratique :

Ajouter un build Docker au pipeline.

Publier une image dans un registre.

Utiliser un secret pour l'authentification.

## Déploiement d'une application

Différence entre livraison et déploiement.

Déploiement manuel et déploiement automatisé.

Déploiement sur un environnement de test.

Mise à jour d'une version.

Rollback simple.

Validation après déploiement.

### Atelier pratique :

Déployer une application avec Docker Compose.

Mettre à jour l'image utilisée.

Valider la nouvelle version.

Revenir à une version précédente.

## Configuration et secrets

Configuration par environnement.

Variables d'environnement.

Fichiers de configuration.

Secrets dans les pipelines.

Risques liés aux secrets dans le code source.

Bonnes pratiques de gestion des informations sensibles.

### Atelier pratique :

Remplacer une configuration codée en dur.

Utiliser des variables sécurisées.

Corriger une mauvaise pratique liée aux secrets.

## Logs, health checks et diagnostic

Lire les logs d'un conteneur.

Comprendre les logs applicatifs.

Ajouter ou utiliser un endpoint de santé.

Valider l'état d'un service.

Diagnostiquer une erreur de démarrage.

#### **Atelier pratique :**

Lire les logs de l'application.

Identifier une erreur.

Corriger la configuration.

Valider la santé du service.

#### **Introduction à l'observabilité**

Pourquoi surveiller une application.

Logs, métriques et traces.

Health checks.

Tableaux de bord.

Introduction aux outils comme Prometheus, Grafana ou équivalents.

#### **Atelier pratique :**

Activer des métriques de base.

Lire l'état de l'application.

Interpréter un problème simple.

#### **Sécurité DevOps de base**

Risques fréquents dans une chaîne CI/CD.

Secrets exposés.

Images vulnérables.

Dépendances obsolètes.

Permissions excessives.

Bonnes pratiques de sécurité pour Docker et les pipelines.

## Atelier pratique :

Identifier une mauvaise pratique.

Analyser une image ou une dépendance.

Corriger une configuration non sécurisée.

## Exemple de Projet Développé

Application à conteneuriser.

Environnement Docker Compose.

Pipeline CI/CD.

Build et publication d'image.

Déploiement.

Validation par logs et health checks.

Correction d'un incident simple.