

Plan de cours N° : 1433

Durée : 2 jours (14h)

CI/CD

PARTICIPANTS / PRE-REQUIS

Développeurs et devops

Connaître les principes de base du développement logiciel. Savoir utiliser Git pour les opérations courantes (réaliser un commit, effectuer un push, fusionner des modifications, créer une branche...)

OBJECTIFS PEDAGOGIQUES

Expliquer les principes et les enjeux du CI/CD. Appliquer une stratégie Git adaptée à la CI. Construire et diagnostiquer un pipeline CI. Intégrer des tests automatisés dans un pipeline. Mettre en œuvre des contrôles de qualité et de sécurité. Optimiser un pipeline et gérer ses artefacts. Configurer les environnements et les validations de déploiement. Concevoir un pipeline de déploiement automatisé. Évaluer un pipeline et proposer des améliorations.

MOYENS PEDAGOGIQUES

Tour de table au début de chaque formation pour définir les objectifs de chaque participant,
Alternance entre apports théoriques (en moyenne 30%) et exercices pratiques (en moyenne 70%),
Utilisation de cas concrets issus de l'expérience professionnelle de nos formateurs,
Remise d'un support de cours,
Assistance post-formation d'une durée de 1 an sur le contenu de la formation via notre adresse mail dédiée formateurs@atp-formation.com

MOYENS PERMETTANT LE SUIVI DE L'EXECUTION ET DES RESULTATS

Positionnement préalable oral ou écrit,
Evaluation des acquis tout au long de la formation par des exercices de synthèse,
Attestation de stage remise à chaque apprenant, avec son niveau d'acquisition pour chaque objectif pédagogique,
Feuille de présence signée par demi-journée,
Questionnaire de satisfaction pour évaluer la qualité de l'enseignement,
En option : passage certification possible selon les thématiques

MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation, équipée d'ordinateurs, d'un vidéo projecteur d'un tableau blanc

MOYENS TECHNIQUES DES CLASSES A DISTANCE

Grâce à un logiciel comme Teams, suivez une formation en temps réel et entièrement à distance. Lors de la classe en ligne, les apprenants interagissent et communiquent entre eux et avec le formateur.

Nous vous conseillons très fortement l'utilisation de votre webcam et de disposer d'un double écran.

Pour toute question avant et pendant le parcours, une assistance technique et pédagogique est à disposition par téléphone au 04.76.41.14.20..

ORGANISATION

Les cours ont lieu de 9h00-12h30 13h30-17h00 (adaptable à la demande).

PROFIL FORMATEUR

Nous recrutons méticuleusement nos formateurs selon 3 critères : expertise, pédagogie et agilité.

ACCESSIBILITE

Les personnes atteintes de handicap souhaitant suivre nos formations sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités d'organisation.

MISE A JOUR

12/10/2026

Siège social :

31 avenue du Granier
38240 MEYLAN

Agences :

170 rue de Chatagnon
38430 Moirans

Le Thélème

1501/1503 route des Dolines
06560 Valbonne

Plan de cours N° : 1433

Durée : 2 jours (14h)

CI/CD

Introduction au CI/CD

Pourquoi le CI/CD : intégration tardive, livraisons manuelles, risques et délais

Définitions : intégration continue, livraison continue, déploiement continu

Métriques DORA

- Fréquence de déploiement

- Délai de livraison des changements

- Taux d'échec des changements

- Temps de restauration du service

Panorama des outils (GitHub Actions, GitLab CI)

Présentation du fil rouge : une petite appli web (API + tests) à automatiser de bout en bout

Git pour la CI

Stratégies de branches

- Trunk-based development

- GitFlow

- Avantages et limites de chacune pour la CI

Pull requests et revue de code

Protection de branches : checks obligatoires avant fusion

Premier pipeline

Anatomie d'un workflow

- Evènements déclencheurs (events)

- Jobs et dépendance entre jobs

- Steps et actions réutilisables

- Runners

Syntaxe YAML d'un workflow

Pipeline de base : build + lint sur chaque push

Lecture des logs et diagnostic d'un pipeline en échec

Tests automatisés

Pyramide des tests : unitaires, intégration, de bout en bout

Couverture de code et rapports

Intégration des tests dans le pipeline

Seuil de couverture : faire échouer le pipeline s'il n'est pas atteint

Qualité et sécurité

Analyse statique du code et linters

Scan des dépendances (vulnérabilités connues)

Détection de secrets dans le dépôt

SBOM (inventaire des composants logiciels)

Artefacts et optimisation

Cache des dépendances

Matrices de build (plusieurs versions, plusieurs environnements)

Artefacts : produire, conserver et réutiliser entre jobs

Mesure et réduction de la durée des pipelines

Gestion des secrets dans le pipeline

Environnements et stratégies de déploiement

Environnements : dev, staging, production

Approbations manuelles

Déploiement automatisé

Choix de la cible : VM, Docker Compose ou Kubernetes selon le contexte

Pipeline complet

- Commit

- Tests

- Construction de l'image

- Déploiement en staging

- Validation manuelle

- Déploiement en production

Bonnes pratiques et pièges

Pipelines fragiles

Tests flaky (instables)

Dette CI

Gouvernance

Siège social :

31 avenue du Granier
38240 MEYLAN

Agences :

170 rue de Chatagnon
38430 Moirans

Le Thélème

1501/1503 route des Dolines
06560 Valbonne