+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 19/12/2025

S'inscrire

Formation Bazel : Build, Test et CI/CD à l’échelle

3 jours (21 heures)

Présentation

Bazel est un système de build open?source pour les entreprises. Cette plateforme d’industrialisation pour des applications multi?langages et multi?plateformes fournit des outils pour compiler tester et packager de manière incrémentale et reproductible.

Notre formation Bazel : Build, Test et CI/CD à l’échelle vous permettra de structurer vos projets, gérer les dépendances et accélérer vos pipelines en tirant parti du cache et de l’exécution distante.

Vous serez en mesure d’obtenir plus d’informations sur les performances de build, de configurer un environnement Bazel complet et de définir les bonnes pratiques de CI/CD et de sécurité.

Bazel facilite l’orchestration des tests et la corrélation des résultats avec les métriques de pipeline. De plus, Bazel s’adapte aux dépôts monorepo comme aux architectures distribuées, du poste de développeur aux agents de CI.

À la suite de cette formation, vous serez en mesure d’installer, configurer et exploiter Bazel, ainsi que de comprendre son architecture et ses mécanismes d’optimisation. Comme toutes nos formations, celle?ci vous présentera la dernière version stable en date et ses nouveautés.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Comprendre l’architecture et les concepts clés de Bazel
- Industrialiser un environnement de build polyglotte
- Réduire les temps de build via cache et remote execution
- Automatiser les tests et intégrer Bazel en CI/CD

- Garantir portabilité, sécurité et traçabilité des artefacts

Public visé

- Développeurs logiciels
- Ingénieurs DevOps
- Architectes applicatifs

Pré?requis

- Aisance en ligne de commande et Git
- Notions de développement (Java, Python, C++ ou Go)
- Premières bases en intégration continue

Programme de notre fromation Bazel : Build, Test et CI/CD à l’échelle

Introduction à Bazel et notions fondamentales

- Comprendre l’architecture de Bazel et son positionnement
- Comparer avec Maven, Gradle, Make et autres outils
- Installation et configuration initiale sur votre poste
- Rôle des fichiers WORKSPACE et BUILD
- Gestion des dépendances externes et Bzlmod
- Atelier : Installation et première configuration d’un projet Bazel

Structure des projets et gestion des dépendances

- Organisation en packages, targets et monorepo polyglotte
- Règles natives : cc_binary, java_library, py_test…
- Introduction à Starlark pour créer des règles personnalisées
- Déclaration/consommation des dépendances et contraintes de plateforme
- Stratégies de modularisation et réutilisation des macros
- Atelier : Structurer un projet multi-langages avec Bazel

Optimisation des builds

- Principe du build incrémental et graphes de dépendances
- Cache local vs remote caching : quand et comment
- Remote execution et parallélisation à grande échelle
- Réglages de performance (sandboxing, stratégies de spawn, RAM/CPU)
- Mesure et diagnostic : profils, build event protocol
- Atelier : Mettre en place un cache distant et mesurer les gains

Gestion des tests avec Bazel

- Tests unitaires, d’intégration et end?to?end avec règles test
- Herméticité, reproductibilité et flaky tests
- Paramètres clés : --test_output, --test_filter, --runs_per_test
- Intégration avec JUnit, pytest, Go test…
- Rapports, temps d’exécution et sharding
- Atelier : Écrire et exécuter une suite de tests multi?langages

Bazel et CI/CD

- Intégrer Bazel dans Jenkins, GitHub Actions, GitLab CI
- Stratégies de pipeline : builds incrémentaux, branch builds, releases
- Gestion des artefacts, cache partagé et isolation des jobs
- Optimisation des temps de cycle et du coût d’exécution
- Bonnes pratiques de fiabilité (timeouts, retry, cache keys)
- Atelier : Créer un pipeline CI/CD complet avec Bazel

Développement multiplateforme

- Builds pour Linux, macOS, Windows
- Cross?compilation et toolchains (C/C++, Java, Go, Python…)
- Dépendances natives et liens dynamiques/statiques
- Intégration avec Docker et images reproductibles
- Portabilité, compatibilité et CI multi?agents
- Atelier : Construire et tester un projet pour plusieurs plateformes

Extension et personnalisation

- Écosystème des rules_* et sélection des extensions
- Écriture de règles en Starlark : providers, actions, attrs
- Macros et factorisation du code de build
- Publication interne de règles partagées
- Gestion des dépendances, versions et compatibilité
- Atelier : Créer une règle Bazel personnalisée

Sécurité et gouvernance

- Politiques d’accès, audit et traçabilité des builds
- Validation et signature des artefacts (SBOM, attestations)
- Gestion des dépendances open?source et licensing
- Analyse de surface d’attaque et durcissement
- Conformité et exigences réglementaires
- Atelier : Mettre en place des contrôles de sécurité dans la chaîne

Maintenance et bonnes pratiques

- Structurer un monorepo pour la maintenabilité
- Convention de nommage, revue de BUILD files
- Suivi de performance et métriques long terme
- Automatisation de la maintenance (linters, buildifier)
- Migration depuis d’autres outils (Maven/Gradle/Make)
- Atelier : Optimiser un projet Bazel existant

Clôture et plan d’implémentation

- Bilan des acquis et des accélérateurs clés
- Checklist d’industrialisation Bazel en CI/CD
- Modèle de .bazelrc et patterns réutilisables
- Roadmap d’adoption et stratégie de changement
- Ressources et veille (releases, LTS, roadmap)
- Atelier : Élaborer votre plan d’adoption Bazel

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d’auto-évaluation nous permettant d’apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d’anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l’acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.

Organisme de formation enregistré sous le numéro 11 75 54743 75. Cet enregistrement ne vaut pas agrément de l'État.
© Ambient IT 2015-2025. Tous droits réservés. Paris, France - Suisse - Belgique - Luxembourg