

GitLab CI/CD, maîtriser la gestion du cycle de vie de vos développements logiciels de l'intégration continue au déploiement continu

Cours Pratique de 3 jours - 21h

Réf : GLN - Prix 2024 : 2 280CHF HT

GitLab n'est pas seulement un gestionnaire de versions, c'est aussi une solution compétitive pour l'ingénierie DevOps. Ce cours pratique vous apprendra à utiliser cet outil dans le but d'améliorer la collaboration dans un projet et d'automatiser un cycle de développement complet (constructions, tests, déploiements...).

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Connaître l'offre GitLab

Pratiquer la gestion de versions avec Git et collaborer avec GitLab

Mettre en place l'intégration continue (CI) et le déploiement continu (CD) avec GitLab

Appréhender les éléments constitutifs d'une usine logicielle DevOps

TRAVAUX PRATIQUES

Alternance de présentations théoriques et de mises en application.

LE PROGRAMME

dernière mise à jour : 11/2023

1) L'offre GitLab

- GitLab CE (Community edition). GitLab EE (Enterprise edition).
- Les différents types d'utilisation de GitLab : gitlab.com, on premise, on cloud (AWS, GCP).
- Procédure d'inscription sur gitlab.com.

Travaux pratiques : Création d'un compte GitLab sur gitlab.com. Parcours de l'interface.

2) Rappels sur Git et son utilisation avec GitLab

- Les concepts Git : blob, tree, commit, revision, branche, tag...
- Gestion locale des fichiers. Consultation et modification de l'historique de travail.
- Gestion des branches. Fusions des branches et gestion des conflits.

Travaux pratiques : Mise en place d'un dépôt distant et simulation d'un travail collaboratif.

3) La gestion du dépôt avec GitLab

- La gestion des collaborateurs d'un projet et leurs droits.
- Le système d'issues et le lien avec les commits.
- Présentation du wiki et des snippets.

Travaux pratiques : Paramétrage des droits et création de merge requests.

4) GitLab CI/CD

- Présentation de GitLab CI/CD et des GitLab runners.
- Présentation de Docker.
- Le fichier manifeste gitlab-ci.yml, présentation du langage YAML.
- Les balises essentielles pour décrire des étapes, des jobs, des traitements (stages, images, script...).

PARTICIPANTS

Développeurs, chefs de projet, administrateurs systèmes, architectes.

PRÉREQUIS

Connaissances de base des commandes Linux. Connaissances de base de la gestion de versions avec Git.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Le suivi d'exécution du pipeline. Jobs automatiques, manuels et planifiés.
- Les artifacts et l'amélioration des performances avec le cache.
- La documentation officielle relative à la syntaxe du fichier manifeste.
- La gestion des environnements.

Travaux pratiques : Exemples simples de création et d'enchaînement de jobs.

5) Plus loin dans l'utilisation de GitLab

- Les types de runners (shared runner, specific runner et group runner).
- Les shared runners disponibles sur gitlab.com gérés par l'équipe GitLab.
- Scalabilité des runners avec Docker Machine.
- Le Docker executor et les images disponibles : MySQL, PostgreSQL, Redis, MongoDB, Ruby...
- Les tests d'intégration, les tests fonctionnels, les tests de montée en charge.
- Le déploiement vers un cloud.
- Le Kubernetes executor et la répartition des jobs dans un cluster Kubernetes.
- Intégration de l'outil de monitoring Prometheus.

Travaux pratiques : Mise en place d'un specific runner. Exécution d'enchaînements de jobs sur différents runners.

6) Fonctionnalités complémentaires de GitLab

- Shared runners sous Windows.
- Infrastructure as code avec Terraform.
- Auto DevOps.
- Tests de sécurité dynamiques (DAST).

LES DATES

CLASSE À DISTANCE

2024 : 26 juin, 14 oct., 16 déc.