



Implementing DevOps Solutions and Practices using Cisco Platforms

Objectifs

DEVOPS

Version : 1.0
5 Jours

A l'issue de ce cours, le stagiaire sera en mesure d'atteindre ses objectifs :

- Décrire la philosophie et les pratiques DevOps, et comment elles s'appliquent aux défis de la vie réelle
- Expliquer les architectures basées sur des conteneurs et les outils disponibles fournis par Docker
- Décrire l'emballage des applications dans des conteneurs et commencer à créer des images de conteneurs sécurisées
- Utiliser la mise en réseau de conteneurs et déployer une application réseau à trois niveaux
- Expliquer les concepts de pipelines d'éléments de configuration (CI) et quels outils sont disponibles
- Implémentez un pipeline de base avec Gitlab CI qui crée et déploie des applications
- Mettre en œuvre des tests et des validations de build automatisés
- Décrire les principes DevOps appliqués à l'infrastructure
- Mettre en œuvre des environnements de test à la demande et expliquer comment les intégrer à un pipeline existant
- Mettre en œuvre des outils pour la collecte, l'analyse et les alertes de métriques et de journaux
- Décrire les avantages de la surveillance de la santé des applications, de la télémétrie et de l'ingénierie du chaos dans le contexte de l'amélioration de la stabilité et de la fiabilité de l'écosystème
- Décrire comment implémenter des workflows DevOps sécurisés en manipulant en toute sécurité des données sensibles et en validant des applications
- Expliquer la conception et les concepts opérationnels liés à l'utilisation d'une combinaison de déploiements cloud publics et privés
- Décrire la conception d'applications modernes et les architectures de microservices
- Décrire les blocs de construction de Kubernetes et comment utiliser ses API pour déployer une application
- Expliquer les modèles de déploiement Kubernetes avancés et implémenter un pipeline automatisé
- Expliquer comment les concepts de surveillance, de journalisation et de visibilité s'appliquent à Kubernetes

Pour plus d'informations : info@learneo.com ou 01 53 20 37 00

CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, Cisco IOS, Cisco Systems, the Cisco Systems logo, and Networking Academy are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this Web site are the property of their respective owners.



Public Concerné

- Ingénieur Commercial Collaboration
- Développeur de logiciels de collaboration
- Ingénieur commercial
- Ingénieur Systèmes
- Architecte de solutions techniques
- Ingénieur de conception sans fil
- Ingénieur sans fil
- Gestionnaire de compte
- Ingénieur conseil en systèmes
- Administrateur réseau
- Ingénieur réseau
- Gestionnaire de réseau

Pré-requis

Avant de suivre ce cours, le stagiaire doit posséder les compétences et les connaissances suivantes :

- Concepts de base du langage de programmation et familiarité avec Python
- Compréhension de base de la virtualisation informatique
- Possibilité d'utiliser Linux, des interfaces textuelles et des outils CLI, tels que Secure Shell (SSH), bash, grep, ip, vim / nano, curl, ping, traceroute et telnet
- Compréhension fondamentale de l'architecture du système d'exploitation Linux et des utilitaires système
- Connaissances de base en réseau de niveau CCNA®
- Compréhension fondamentale des concepts DevOps
- Connaissance et familiarité avec les concepts d'intégration continue, de déploiement continu et de livraison continue CI / CD)
- Expérience pratique avec Git
- Developing Applications and Automating Workflows using Cisco Core Platforms (DEVASC)
- Developing Applications Using Cisco Core Platforms and APIs (DEVCOR)

Pour plus d'informations : info@learneo.com ou 01 53 20 37 00

CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, Cisco IOS, Cisco Systems, the Cisco Systems logo, and Networking Academy are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this Web site are the property of their respective owners.



Plan du cours détaillé

1. Présentation du modèle DevOps
 - 1.1. Philosophie DevOps
 - 1.2. Pratiques DevOps
2. Présentation des conteneurs
 - 2.1. Architectures basées sur des conteneurs
 - 2.2. Conteneurs Linux
 - 2.3. Présentation de Docker
 - 2.4. Commandes Docker
3. Empaqueter une application à l'aide de Docker
 - 3.1. Dockerfiles
 - 3.2. Images d'or
 - 3.3. Pratiques de traitement sécuritaires
4. Déployer une application à plusieurs niveaux
 - 4.1. Réseaux Linux
 - 4.2. Réseau Docker
 - 4.3. Docker Compose
5. Présentation de CI / CD
 - 5.1. Intégration continue
 - 5.2. Outils CI
 - 5.3. Pipelines DevOps
6. Création du flux DevOps
 - 6.1. Présentation de GitLab
 - 6.2. Présentation de GitLab CI
 - 6.3. Livraison continue avec GitLab
7. Validation du processus de création d'application
 - 7.1. Test automatisé dans le flux CI
8. Création d'un flux de déploiement amélioré
 - 8.1. Validation après le déploiement
 - 8.2. Stratégies de déploiement des versions
9. Extension des pratiques DevOps à l'ensemble de l'infrastructure
 - 9.1. Introduction à NetDevOps
 - 9.2. L'infrastructure comme code
10. Implémentation d'environnements de test à la demande au niveau de l'infrastructure
 - 10.1. Outils de gestion de la configuration
 - 10.2. Présentation de Terraform
 - 10.3. Présentation d'Ansible
 - 10.4. Fichier d'inventaire ansible
 - 10.5. Utilisez le module de configuration Cisco IOS Core
 - 10.6. Modèles Jinja2 et Ansible
 - 10.7. Jinja2 de base avec YAML

Pour plus d'informations : info@learneo.com ou 01 53 20 37 00



- 10.8. Configuration de modèles avec Ansible
- 11. Surveillance dans NetDevOps
 - 11.1. Introduction à la surveillance, aux métriques et aux journaux
 - 11.2. Introduction à Elasticsearch, Beats et Kibana
 - 11.3. Introduction à Prometheus et à l'instrumentation du code Python pour l'observabilité
- 12. Ingénierie pour la visibilité et la stabilité
 - 12.1. Santé et performances des applications
 - 12.2. Présentation d'AppDynamics
 - 12.3. Principes d'ingénierie du chaos
- 13. Sécurisation des workflows DevOps
 - 13.1. Présentation de DevSecOps
 - 13.2. Sécurité des applications dans le pipeline CI / CD
 - 13.3. Sécurité de l'infrastructure dans le pipeline CI / CD
- 14. Explorer les stratégies multicloud
 - 14.1. Déploiement d'applications dans plusieurs environnements
 - 14.2. Introduction à la terminologie du cloud public
 - 14.3. Suivi et projection des coûts du cloud public
 - 14.4. Considérations relatives à la haute disponibilité et à la reprise après sinistre
 - 14.5. IaC pour une consommation répétable de cloud public
 - 14.6. Comparaison de la stratégie des services cloud
- 15. Examen des architectures d'application et de déploiement
 - 15.1. L'application à douze facteurs
 - 15.2. Architectures de microservices
- 16. Décrire Kubernetes
 - 16.1. Concepts de Kubernetes: nœuds, modules et clusters
 - 16.2. Concepts de Kubernetes: stockage
 - 16.3. Concepts de Kubernetes: mise en réseau
 - 16.4. Concepts de Kubernetes: sécurité
 - 16.5. Présentation de l'API Kubernetes
- 17. Intégration de plusieurs déploiements de centre de données avec Kubernetes
 - 17.1. Modèles de déploiement de Kubernetes
 - 17.2. Scénarios d'échec de Kubernetes
 - 17.3. Techniques d'équilibrage de charge Kubernetes
 - 17.4. Espaces de noms Kubernetes
 - 17.5. Déploiement de Kubernetes via des pipelines CI / CD
- 18. Surveillance et journalisation dans Kubernetes
 - 18.1. Pipeline de métriques de ressources Kubernetes
 - 18.2. Kubernetes Full Metrics Pipeline et journalisation

Pour plus d'informations : info@learneo.com ou 01 53 20 37 00

CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, Cisco IOS, Cisco Systems, the Cisco Systems logo, and Networking Academy are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.
All other trademarks mentioned in this Web site are the property of their respective owners.



- Interagir avec GitLab Continuous Integration (CI)
- Explorer les outils de ligne de commande Docker
- Empaqueter et exécuter un conteneur WebApp
- Créez et déployez plusieurs conteneurs pour créer une application à trois niveaux
- Explorez le réseau Docker
- Créer et déployer une application à l'aide de Docker Compose
- Implémenter un pipeline dans Gitlab CI
- Automatisez le déploiement d'une application
- Valider le processus de création d'application
- Valider le déploiement et réparer l'infrastructure
- Créer une infrastructure Yaml Ain't Markup Language (YAML) en tant que spécification de code (IaC) pour l'environnement de test
- Gérez les environnements de test à la demande avec Terraform
- Créez des playbooks Ansible pour gérer l'infrastructure
- Intégrer l'environnement de test dans le pipeline CI / CD
- Mettre en œuvre des contrôles de santé avant le déploiement
- Configurer la journalisation pour les serveurs d'applications et visualiser avec Kibana
- Créer un tableau de bord système axé sur les métriques
- Utiliser les alertes via Kibana
- Surveillance des applications d'instruments
- Utilisez des alertes et des seuils pour notifier l'écouteur Webhook et les salles Cisco Webex® Teams™
- Infrastructure sécurisée dans le pipeline CI / CD
- Explorer la configuration de Kubernetes et déployer une application
- Explorer et modifier un pipeline CI / CD Kubernetes
- Surveillance et métriques Kubernetes: Elasticsearch, Logstash et Kibana (ELK)

Pour plus d'informations : info@learneo.com ou 01 53 20 37 00

CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, Cisco IOS, Cisco Systems, the Cisco Systems logo, and Networking Academy are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this Web site are the property of their respective owners.