

Formation Puppet, Automatisation de la gestion de ses configurations

Présentation

Cette formation intermédiaire vous apprend à analyser des besoins d'industrialisation, à mettre en œuvre des manifestes et modules Puppet maintenables et à configurer une infrastructure d'automatisation robuste. En trois jours, vous couvrirez la modélisation d'état, la gestion d'inventaire, l'intégration CI/CD et les bonnes pratiques de sécurité et de conformité. Les TP aboutissent à un socle Puppet opérationnel avec pipeline de déploiement contrôlé.

Durée : 21,00 heures (3 jours)

Tarif INTRA : Nous consulter

Objectifs de la formation

- Analyser un parc systèmes et formuler des états cibles exprimés en code Puppet.
- Mettre en œuvre des manifestes, classes et modules réutilisables avec gestion des données via Hiera.
- Configurer une infrastructure Puppet (Server/Agent), l'inventaire et la distribution des catalogues.
- Mettre en œuvre des tests, du CI/CD et des revues de code pour fiabiliser les changements.
- Analyser sécurité/conformité et optimiser performance, gouvernance et exploitation.

Prérequis

- Bases d'administration Linux (services, paquets, fichiers, permissions).
- Notions de Git (branches, commit, pull request).
- Connaissances élémentaires de scripting (Shell/PowerShell).

Public

- Ingénieurs systèmes/DevOps, administrateurs Linux/Windows, SRE et équipes d'exploitation souhaitant industrialiser.



- Contextes : standardisation de serveurs/applications, durcissement et conformité, réduction MTTR, scalabilité d'exploitation.
- Expérience attendue : maîtrise de base Linux/CLI, notions Git/YAML, premières expériences d'automatisation.

Programme de la formation

Jour 1 – Modéliser l'état et structurer le code Puppet

Session du matin :

- Rappels d'infrastructure as code : modèles déclaratifs, idempotence, ressources et catalogues
- Langage Puppet DSL : ressources, classes, defines, templates (ERB/EPP)
- Structuration des modules : conventions, métadonnées, dépendances (metadata.json), Forge

Session de l'après-midi :

- Gestion des données avec Hiera : hiérarchies, backends YAML/JSON, séparation code/données
- Bonnes pratiques : params class vs data in Hiera, profils/rôles (roles/profiles pattern)
- Premier module : installation/gestion d'un service applicatif multi-OS

TP / Exercice :

- Création d'un module role/web et profile/nginx avec variables Hiera (ports, vhosts). Livrable : repo Git avec tests basiques et README

Points clés & takeaways :

- Savoir modéliser un besoin d'exploitation en ressources et classes
- Structurer des modules maintenables et orientés réutilisation

Jour 2 – Infrastructure Puppet, déploiement de catalogues et orchestration

Session du matin :

- Topologie : Puppet Server, agents, certificats, environnement de code, compilars
- r10k/Code Manager : déploiement d'environnements à partir de Git, contrôle de versions des modules
- Inventaire et classification : node definitions, External Node Classifier (ENC), regroupements

Session de l'après-midi :

- Orchestration et exécution distante : Puppet Bolt (aperçu) et plans simples
- Gestion multi-OS et dépendances de paquets, services et fichiers ; ordonnancement et notifications
- Observabilité : rapports d'exécution, logs, tableaux de bord ; dépannage courant des runs

TP / Exercice :

- Mise en place d'un Puppet Server de lab, déploiement d'environnements via r10k et application d'un rôle sur plusieurs nœuds. Livrable : dépôt control-repo documenté

Points clés & takeaways :

- Standardiser le déploiement du code via control-repo et r10k
- Comprendre le cycle de compilation et la distribution des catalogues

Jour 3 – Tests, CI/CD, sécurité et exploitation avancée

Session du matin :

- Qualité et tests : linting (puppet-lint), unit tests (rspec-puppet), validation Hiera ; tests d'intégration (serverspec/inspec – aperçu)
- CI/CD : pipelines GitHub Actions/Azure DevOps pour valider, packager et déployer via r10k/Code Manager

- Sécurité : gestion des secrets (Hiera-eyaml), certificats, contrôle des accès et journalisation

Session de l'après-midi :

- Performance et scalabilité : compile masters, TTL, tuning JRuby/GC, stratégies d'exécution
- Conformité & gouvernance : politiques, séparation rôles/profils, revue de code et approbations
- Opérations : runbooks, gestion des changements et rollback, gestion de parc hétérogène

TP / Exercice :

- Pipeline CI/CD de validation et déploiement d'un module profile/app vers un environnement dev puis prod. Livrable : fichier YAML du pipeline et captures d'exécution

Points clés & takeaways :

- Fiabiliser les livraisons par tests et pipeline standardisé
- Mettre sous contrôle sécurité, performance et conformité

Organisation

Formateur

Les formateurs de Docaposte Institute sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont par nos référents internes. Riches de leur expérience sur le sujet, ils sauront accompagner vos collaborateurs dans leur montée en compétences.

Moyens pédagogiques et techniques

- Apports des connaissances communes.
- Mises en situation sur le thème de la formation et des cas concrets.
- Méthodologie d'apprentissage attractive, interactive et participative.

- Equilibre théorie / pratique : 60 % / 40 %.
- Supports de cours fournis au format papier et/ou numérique.
- Ressources documentaires en ligne et références mises à disposition par le formateur.
- Pour les formations en présentiel dans les locaux mis à disposition, les apprenants sont accueillis dans une salle de cours équipée d'un réseau Wi-Fi, d'un tableau blanc ou paperboard. Un ordinateur avec les logiciels appropriés est mis à disposition (le cas échéant).

Dispositif de suivi de l'exécution et de l'évaluation des résultats de la formation

En amont de la formation :

- Recueil des besoins des apprenants afin de disposer des informations essentielles au bon déroulé de la formation (profil, niveau, attentes particulières...).
- Auto-positionnement des apprenants afin de mesurer le niveau de départ.

Tout au long de la formation :

- Évaluation continue des acquis avec des questions orales, des exercices, des QCM, des cas pratiques ou mises en situation...

A la fin de la formation :

- Auto-positionnement des apprenants afin de mesurer l'acquisition des compétences.
- Evaluation par le formateur des compétences acquises par les apprenants.
- Questionnaire de satisfaction à chaud afin de recueillir la satisfaction des apprenants à l'issue de la formation.
- Questionnaire de satisfaction à froid afin d'évaluer les apports ancrés de la formation et leurs mises en application au quotidien.

Accessibilité

Nos formations peuvent être adaptées à certaines conditions de handicap. Nous contacter pour toute information et demande spécifique.