PUPPET, GÉRER SES CONFIGURATIONS LOGICIELLES SOUS LINUX

REF **PGCLL**

- ***Comprendre le fonctionnement de Puppet***
- ***Installer et configurer Puppet***
- ***Déployer des logiciels et leur configuration***
- ***Répliquer une configuration existante avec Puppet***

2

Prix : 1690€ / HT

OUTILS PÉDAGOGIQUES

PUBLIC VISÉ

- Administrateurs systèmes Linux/Unix intégrant une équipe qui utilise déjà Puppet ou souhaitant le mettre en œuvre dans un contexte professionnel.

MODALITÉS D'ÉVALUATION

MODALITÉS DE FINANCEMENT

MODALITÉS ET DÉLAIS D'ACCÈS

OBJECTIFS PÉDAGOGIQUES

ACCESSIBILITÉ

LES POINTS FORTS DE LA FORMATION

PRÉ-REQUIS

Connaissances de base et pratique régulière de l'administration Linux/Unix.

MODALITÉS ET DÉLAIS D'ACCÈS

ATTESTATION OBTENUE

EFFECTIF DE LA FORMATION

CERTIFICATION

MODALITÉ PÉDAGOGIQUE

Cours dispensé en mode présentiel avec une alternance d'apports théoriques et méthodologiques, et de mises en situations pratiques

PROCHAINES SESSIONS

PROGRAMMES DE PUPPET, GÉRER SES CONFIGURATIONS LOGICIELLES SOUS LINUX

Présentation

- Pourquoi la gestion d'infrastructure centralisée ?
- Pourquoi Puppet ?
- Les différentes versions et distributions de Puppet.
- Le fonctionnement général de Puppet.

Mise en œuvre

- Installer un Puppet master.
- Configurer un Puppet master.
- Installer et configurer l'agent Puppet.

Travaux pratiques

Installation et configuration initiale des outils. Prise en main de l'environnement.

Prise en main du langage de Puppet

- Les principaux types supportés par Puppet.
- La syntaxe du langage

- Générer des fichiers à partir de templates.
- Les relations entre ressources.
- Les ressources définies.
- La remontée d'informations avec Facter.

Travaux pratiques

Gérer les utilisateurs et les groupes. Déployer des logiciels et leur configuration. Gérer et superviser les services. Décrire la configuration actuelle avec Puppet.

Réutilisation de modules Puppet

- Pourquoi réutiliser des modules existants ?
- Les modules, les classes.
- Utiliser un module existant.
- Pourquoi structurer son code Puppet sous forme de modules réutilisables ?
- Ecrire un nouveau module.

Travaux pratiques

Sélectionner et installer un module existant. Utiliser un module existant. Restructurer du code Puppet sous forme de classes réutilisables. Ecriture d'un module réutilisable.

Connecter des agents au Puppet master

- L'architecture maître-agent.
- L'authentification des clients.
- La gestion des certificats SSL.
- Planifier le lancement de l'agent.

Travaux pratiques

Intégrer un nouveau client à une infrastructure Puppet. Lancer l'agent comme service résident. Lancer l'agent avec cron.

Bonnes pratiques pour déployer et utiliser Puppet

- Le formatage du code.
- Utiliser la gestion de versions et l'intégrer avec Puppet.
- Séparer le code générique des paramètres spécifiques à un système.
- Gérer les données privées, comme les mots de passe et clés privées SSL.
- Superviser les rapports.

Travaux pratiques

Utiliser Puppet-lint pour homogénéiser notre code. Extraire les paramètres spécifiques à un système vers Hiera.