

## Formation Puppet Initiation + Approfondissement

<b>Durée :</b>	5 jours
<b>Public :</b>	Administrateurs systèmes
<b>Pré-requis :</b>	Avoir suivi la formation Linux : administration système (bases + services), ou avoir des connaissances équivalentes
<b>Objectifs :</b>	Comprendre l'intérêt du mouvement DevOps - Installer Puppet - Ecrire des manifestes pour déployer une infrastructure avec Puppet - Maîtriser les bonnes pratiques et assurer la cohérence du SI avec Puppet
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	DEV866-F
<b>Note de satisfaction des participants:</b>	4,79 / 5

### Introduction DevOps

- L'origine du mouvement DevOps
- Le besoin d'industrialisation pour l'opérationnel
- Tour d'horizon des outils
- Positionnement de Puppet dans le paysage actuel
- Présentation de git

### Présentation de Puppet

- Fonctionnement du catalogue, architecture de l'ensemble
- L'architecture autonome (stand-alone) avec Puppet apply
- L'architecture agent/maître (agent/master)
- Implications sécuritaires
- Installation, présentation des répertoires de configuration

**Atelier pratique : installation du Puppet master, mise en place d'un catalogue trivial**

### Découverte du langage

- Les ressources dans Puppet
- La couche d'abstraction des ressources (RAL) : types, attributs, valeurs
- Les manifestes
- Idempotence, états souhaités
- Ordonnancement des ressources
- Variables, faits (facts, avec Factor) et conditions
- Les classes et les modules
- Les classes paramétrées

Les types définis, ou « super-classes »

**Atelier pratique : écriture de manifestes mettant en jeu les ressources de base, création de classes, découverte des modules.**

### **Puppet dans l'infrastructure**

Organisation du manifeste : retour sur les classes et modules  
La définition des modèles (templates ERB) pour les fichiers  
Utilisation du serveur de fichiers intégré à Puppet  
Bonnes pratiques et patterns pour les classes et modules  
Classification des nœuds, utilisation du bloc node et des include

**Atelier pratique : mise en place de modules personnalisés, paramétrage du serveur, paramétrage de nœuds complets hétérogènes via Puppet.**

### **Utilisation de Factor**

Rappel de l'utilisation des faits Puppet  
Création de faits avec résolutions simples et complexes  
Bonnes pratiques sur la structuration des faits Puppet

**Atelier pratique : définition de Facts personnalisés, exploitation dans un scénario réaliste**

### **Programmation plus avancée avec Puppet**

Écrire du code réutilisable avec Puppet  
La forge et les modules puppetlabs  
Un peu de Ruby pour les templates  
Comprendre la classification et les hiérarchies  
Fonctionnement de Hiera  
Utilisation des hiérarchies, introduction aux ENC  
Mise en place d'un ENC

**Atelier pratique : écriture de manifestes complexes, découverte des modules de la forge Puppet, mise en place d'un scénario grandeur nature avec Hiera**

### **Configuration de production avancée**

Détail de la configuration en mode production  
La gestion des certificats SSL avec Puppet  
Optimisation des ressources avec PuppetDB  
Utilisation des types virtuels, réalisation de ressources  
Collecteur de ressources exportées

**Atelier pratique : configuration apache2+passenger, gestion de certificats, mise en place de PuppetDB**