

## Formation CFEngine

<b>Durée :</b>	4 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 - Mettre en place un système de gestion de configuration basé sur CFEngine
<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>	DEV926-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Comprendre le mouvement DevOps

Origines, caractéristiques et intérêts du mouvement DevOps  
Le besoin d'automatisation des infrastructures du SI  
CAMS : Culture, Automation, Measure, Share  
Tour d'horizon des outils

### Mettre en place un projet d'automatisation

Gestion de la configuration (convergence, idempotence, Single Copy Nirvana, ...)  
Comparaison des principales approches  
Organisation de l'équipe, objectifs, ...  
Pièges à éviter  
CFEngine : présentation, fonctionnalités  
Architecture CFEngine : agents, distribution des configurations  
Référentiels de configuration, Distribution de fichiers, Architecture type  
Modèle d'état cible vs l'approche procédurale

#### **Atelier : Exemples d'approches d'automatisation, comparaisons**

### Adopter CFEngine

Prise en main de l'outil : syntaxe, exécution, utilisation de classes  
Attributs de promesses, variables, conteneur (body) Promise types Templating de fichiers : fichiers statiques, Mustache, génération de fichiers variables  
Gestion de services Structure de l'agent  
CFEngine : composants, démons Modification de fichiers existants  
Utilisation de tableaux (arrays), lecture de paramètres depuis des fichiers externes (Content Driven Policy ou CDP)

#### **Atelier : Installation, lancement et débogage, génération et traitement de fichiers**

## **Mettre en place une architecture client/serveur CFEngine**

Architecture client / serveur, apports  
Mise en place  
Limites

**Atelier : Mise en place client/serveur CFEngine**

## **Réaliser des tâches avancées**

Variables et classes avancées  
Classes persistantes, automatiques  
Gestion avancée des fichiers : copies récursives / partielles, filtrage des fichiers, détection de changements (tripwire), édition avancée (gestion de champs dans les fichiers tabulaires, édition XML et JSON, ...)

**Atelier : Manipulation de classes persistantes et traitement avancés sur des fichiers**

## **Interfacer et Administrer CFEngine**

Commandes externes pour interfacer CFEngine (execresult, returnszero, transformer, ...)  
Extension via des modules (protocole, usage)  
Test de montée en charge d'une architecture CFEngine : planification, load balancing, analyse/optimisations  
Gestion des logs  
Génération de rapports  
Monitoring et analyse d'incidents

**Atelier : Multiples tâches d'administration, tests de montée en charge, scénarios de gestion des incidents**