

Puppet : administration centralisée

Durée: 3 jours

1610 €

26 au 28 février
11 au 13 juin

1er au 3 octobre
10 au 12 décembre

Public:

Administrateurs, exploitants souhaitant utiliser Puppet pour la gestion centralisée des configurations logicielles de multiples serveurs ou machines virtuelles.

Objectifs:

Comprendre le fonctionnement de Puppet, et savoir le mettre en oeuvre pour une administration centralisée des configurations. Ce stage est illustré par de nombreux travaux pratiques sur le développement des manifests puppet.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases du système Unix.

Programme:

- Introduction : Présentation du besoin et des fonctionnalités de Puppet.
Gestion des configurations logicielles.
Collecte et centralisation des informations.
Définition des configurations cibles, mises à jour automatiques ou manuelles.
L'orchestration dans le cas d'un cloud d'entreprise.
La gestion de configuration de machines virtuelles.
Historique de puppet et les différentes versions de puppet.
Présentation de Puppet Enterprise.
- Architecture : Principe client-serveur, modules de configuration, les agents Puppet, la console, l'outil de gestion de cloud, les manifests puppet.
Présentation des plate-formes supportées comme serveur et comme client (agent puppet).
Mode opératoire : définition des configurations, vérification de l'état des clients, simulation des changements proposés par Puppet, application sur les systèmes cibles.

Puppet : administration centralisée

- Installation et configuration** : Prérequis systèmes.
Travaux pratiques : Installation d'une infrastructure Puppet : serveur, base de données, agents.
Etude du fichier de configuration puppet.conf. Configuration des rôles : maître, agent, console... Gestion des certificats sur les clients.
- Le langage puppet** : Introduction : présentation des manifests, modules, templates, des modules disponibles sur la forge puppet.
Les manifests : description du langage déclaratif de puppet. Syntaxe, dépendance entre instructions. Etudes des variables, présentation des facts. Déclarations conditionnelles : if, case, selectors
Travaux pratiques : rédaction d'un exemple de manifest de création d'un fichier et modification des droits, application de ce manifest sur un poste client cible.
Etude détaillée des 'ressources' puppet : file, package, service, user, exec, notify, ..
Exemple de la ressource 'file' : différents attributs disponibles pour déterminer les caractéristiques d'un fichier, s'assurer de sa présence, ...
Exemple de la gestion de dépendance : production de manifest pour l'installation de sshd
Les classes et modules. Installation de modules.
Travaux pratiques : création d'un module pour propager la configuration de bash sur les postes clients.
Définition des templates. Exemple de création de templates en utilisant le langage ERB.
Etude des paramètres de classes. Définition de nouveaux types de ressources.
Mise en oeuvre sur de nombreux travaux pratiques.
Utilisation et génération de documentations au format puppet. Organisation du site .pp.
- Bonnes pratiques**: Retours d'expériences et méthodes d'organisation et de développement des scripts puppet. Présentation des patterns pour puppet.