

# Lab 4 - Installer l'agent Puppet

## Aperçu

Dans ce lab, nous allons:

- Modifiez manuellement le fichier `/etc/hosts` pour préparer l'installation de l'agent (si ce n'est déjà fait)
- Installez l'agent Puppet sur le nœud **agent**

## Étapes de pré-installation

Assurez-vous que votre VM **agent** est démarrée et y connectez-vous.

```
vagrant up agent
vagrant ssh agent
```

## Modifier le fichier hosts

Encore une fois, notre fichier `/etc/hosts` devrait déjà être configuré, car nous avons utilisé un box Vagrant pré-configuré qui contient déjà les entrées souhaitées, mais par souci d'exhaustivité, revérifions ...

Affichez et/ou modifiez `/etc/hosts` et assurez-vous qu'il ressemble exactement à ceci (supprimez toutes les autres lignes): (Cela devrait être exactement le même fichier d'hôtes que celui que nous avons créé sur le nœud maître **puppet**)

```
127.0.0.1      localhost
192.168.198.10 puppet.example.com puppet
192.168.198.11 agent.example.com  agent
192.168.198.12 gitlab.example.com gitlab
```

Dans un lab ultérieur, nous écrirons le code Puppet pour maintenir les entrées `/etc/hosts`, mais pour l'instant, nous les ajoutons manuellement.

## Installer l'agent

Add the puppet labs repo:

```
rpm -Uvh https://yum.puppetlabs.com/puppet5/puppet5-release-el-7.noarch.rpm
```

Installer le client Puppet:

```
yum install -y puppet-agent
```

Une fois l'installation terminée, accédez au répertoire de configuration de puppet et modifiez le fichier puppet.conf.

```
vi /etc/puppetlabs/puppet/puppet.conf
```

Collez la configuration suivante.

```
[main]
certname = agent.example.com
server = puppet.example.com
environment = production
runinterval = 1h
```

## Exécutez l'agent Puppet

Créez un lien soft du binaire puppet:

```
ln -s /opt/puppetlabs/bin/puppet /usr/bin/puppet
```

Exécutez l'agent Puppet manuellement. Cela entraînera une demande de certificat SSL à générer et à envoyer au Puppet Master (si ce n'est pas déjà envoyée en arrière-plan.)

```
[root@agent ~]# puppet agent -t
Info: Creating a new SSL key for agent.example.com
Info: Caching certificate for ca
Info: Caching certificate_request for agent.example.com
Info: Caching certificate for ca
Exiting; no certificate found and waitforcert is disabled
```

Remarque: l'agent de Puppet fonctionne également en arrière-plan, il peut donc avoir déjà généré la clé SSL, vous risquez donc de ne pas le voir dans votre sortie. Dans ce cas, vous verriez juste ceci:

```
[root@agent ~]# puppet agent -t
Exiting; no certificate found and waitforcert is disabled
```

## Signez le certificat

Ensuite, nous devons signer le certificat de l'agent sur le Master, alors passez à votre terminal **Puppet** et exécutez les commandes suivantes sur le Puppet Master en tant que root:

```
puppet cert list
puppet cert sign agent.example.com
```

La commande **puppet cert list** affiche toutes les demandes de signature de certificat en attente. Vous devriez voir celui qui vient d'être généré par l'exécution de votre agent.

```
[root@puppet ~]# puppet cert list
"agent.example.com" (SHA256)
31:EA:4D:60:DE:44:E8:E1:A1:1A:2E:48:1E:81:CA:40:43:4A:A7:39:E8:B9:61
:63:F3:0F:CF:2E:B7:CC:98:22
```

La commande **puppet cert sign agent.example.com** signe le certificat et supprime la demande de signature.

```
[root@puppet ~]# puppet cert sign agent.example.com
Notice: Signed certificate request for agent.example.com
Notice: Removing file Puppet::SSL::CertificateRequest
agent.example.com at
'/etc/puppetlabs/puppet/ssl/ca/requests/agent.example.com.pem'
```

## Exécuter Puppet ...

Maintenant, revenons sur le nœud de l'agent: exécutons à nouveau puppet (assurez-vous que vous êtes root)

```
puppet agent -t
```

Vous devriez voir beaucoup de sortie à l'écran montrant les changements qui sont appliqués. (Puppet is installing and configuring MCollective on the agent.)

## Réexécutez l'agent Puppet

Exécutez puppet une **seconde** fois, et vous devriez obtenir une exécution propre sans changement.

```
puppet agent -t
```

## Le nom du certificat

Le **certname** est le nom attribué à l'agent Puppet. C'est comme ça que Puppet master identifie chaque nœud. La plupart du temps, le certname sera le même que le nom de domaine complet de l'hôte, mais cela peut être tout ce que vous choisissez. La meilleure pratique consiste à utiliser le FQDN.

Pour en savoir plus sur la transmission de paramètres au programme d'installation de l'agent, consultez: [https://docs.puppet.com/pe/latest/install\\_agents.html#passing-configuration-parameters-to-the-install-script](https://docs.puppet.com/pe/latest/install_agents.html#passing-configuration-parameters-to-the-install-script)

## Sommaire

À ce stade, nous avons:

- un nœud **Puppet Master** (hostname **puppet.example.com**) qui exécute également un agent pour se configurer
- un nœud **Puppet Agent** (hostname **agent.example.com**) qui exécute un agent, et où nous testerons le code et en apprendrons plus sur Puppet