Puppet

Vocabulaire Puppet

```
Noeud (Node) : serveur ou poste de travail administré par Puppet ;
Site : ensemble des noeuds gérés par le Puppet Master ;
Classe : moyen dans Puppet de séparer des morceaux de code ;
Module : unité de code Puppet qui est réutilisable et pouvant être partagé ;
Catalogue : ensemble des classes de configuration à appliquer à un nœud ;
Facter : librairie multi-plateforme qui fournit à Puppet sous forme de variables
les informations propres au système (nom, adresse ip, système d’exploitation,
etc.) ;
Ressource (Resource): objet que Puppet peut manipuler (fichier, utilisateur,
service, package, etc.) ;
Manifeste (Manifest) : regroupe un ensemble de ressource.
```

Architecture

Puppet conseille de coupler son fonctionnement avec un gestionnaire de version type « git ». Un serveur PuppetMaster contient la configuration commune et les points de différence entre machines ;

Chaque client fait fonctionner puppetd qui :

- applique la configuration initiale pour le nœud concerné ;
- applique les nouveautés de configuration au fil du temps ;
- s’assure de manière régulière que la machine correspond bien à la configuration voulue.

La communication est assurée via des canaux chiffrés, en utilisant le protocole https et donc TLS (une mini-pki est fournie).

Toute la configuration (le référentiel) de Puppet est centralisée dans l’arborescence /etc/puppet du serveur de référence :

```
/etc/puppet/manifests/site.pp : est le premier fichier analysé par Puppet pour
définir son référentiel. Il permet de définir les variables globales et d’importer
des modules ;

/etc/puppet/manifests/node.pp : permet de définir les nœuds du réseau. Un nœud
doit être défini par le nom FQDN de la machine ;

/etc/puppet/modules/<module> : sous-répertoire contenant un module.
```

Mes instances Aws Ec2

```
CentOS 7 (x86_64) - with Updates HVM
t3.small 2vcpu 2go
2Gm RAM
2vcpu
20Gb SSD
le meme vpc
SG Tous les ICMP - IPv4 Tous N/A 0.0.0.0/0
centos  root
```

- oussama-Puppet-Master : 44.201.37.132 : 172.31.15.248
- oussama-Puppet-Slave : 3.80.10.60 : 172.31.8.58

Pré-installation de Puppet

Dans un premier temps, nous allons procéder à une installation et à une configuration à la fois sur les serveurs Puppet Master et Puppet Agent.

Avoir un nom fqdn pour nos instances

- Nous allons établir un nom FQDN pour nos 2 instances afin que nos certificat fonctionne corectement.
- Mes instances vos dialoguer sur notre réseau privé, ainsi je configure le fichier localhost.

dans /etc/hosts il faut ajouter l'IP et le nom de la 2e machine

```
Définir un nom de domaine pour l'environnement Tout d'abord, utilisez un nom de
domaine pour votre environnement. Pensez que vous allez configurer un
environnement de Puppets pour la société ABC, vous pouvez définir le domaine pour
cela comme « abc.com » ou « dc1.abc.com » (centre de données 1 de la société ABC).
Si vous le faites à des fins de test, il est conseillé d'utiliser 'example.com'. «
exemple.com » est un nom de domaine réservé à des fins de documentation et
d'exemple et personne ne peut enregistrer ce domaine, ce qui évitera de nombreux
problèmes de résolution DNS.
```

```
Donnez un nom de domaine complet approprié pour chaque hôte Définissez un nom de
domaine complet (FQDN) pour chaque hôte au sein de l'environnement Puppet, y
compris le nœud Master Puppet. Cela réduira de nombreux problèmes liés au SSL. Il
ne suffit pas de simplement donner un nom d'hôte car la plupart des systèmes
ajoutent un domaine (via DHCP) qui introduira quelques problèmes.
Exécutez 'hostname' et 'hostname -f' et voyez la différence.
```

Puppet Master

```
=> ifconfig
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
```

```
        inet 172.31.15.248
```

=> sudo hostnamectl set-hostname puppetmaster.oussama.galere

=> sudo vi /etc/hosts

```
127.0.0.1 localhost
172.31.15.248 puppetmaster.oussama.galere
172.31.8.58 puppetslave.oussama.galere
```

=> ping puppetslave.oussama.galere => sudo reboot

Puppet Slave

=> ifconfig

```
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.8.58
```

=> sudo hostnamectl set-hostname puppetslave.oussama.galere => sudo reboot

=> sudo vi /etc/hosts

```
127.0.0.1 localhost
172.31.15.248 puppetmaster.oussama.galere
172.31.8.58 puppetslave.oussama.galere
```

=> ping puppetmaster.oussama.galere => sudo reboot => sudo yum update => sudo reboot

Installation de Puppet

- CentOS 7 = puppet7-release-el-7

Ajouter le dépôt de Puppet Afin d'ajouter le référentiel de Puppet au système, exécutez la commande rpm ci-dessous. https://yum.puppet.com/puppet7-release-el-7.noarch.rpm

Installation Puppet Master

enable the Enterprise Linux 7 repository:

sudo rpm -Uvh https://yum.puppet.com/puppet7-release-el-7.noarch.rpm sudo yum update -y

l'agent sera installé automatiquement

sudo yum install puppetserver -y => sudo reboot

Configurer le Master Puppet

Mémoire maximale

```
Une fois l'installation terminée, nous devons configurer l'allocation de mémoire
pour puppetserver. Nous allons maintenant définir l'allocation de mémoire maximale
pour Puppetserver à 1 Go. Pour cela, éditez la configuration 'puppetserver'.
```

=> sudo vi /etc/sysconfig/puppetserver

```
* Avant
JAVA_ARGS="-Xms2g -Xmx2g -
Djruby.logger.class=com.puppetlabs.jruby_utils.jruby.Slf4jLogger"
* Apres
JAVA_ARGS="-Xms1g -Xmx1g -
Djruby.logger.class=com.puppetlabs.jruby_utils.jruby.Slf4jLogger"
```

- Enregistrez le fichier et quittez.

Allez dans le répertoire de configuration de puppet et éditez le fichier 'puppet.conf'.

sudo vi /etc/puppetlabs/puppet/puppet.conf

Ajoutez la configuration suivante:

```
[master]
dns_alt_names=puppetmaster.oussama.galere

[main]
certname = puppetmaster.oussama.galere
server = puppetmaster.oussama.galere
environment = production
runinterval = 1h
```

Ensuite, enregistrez le fichier et quittez. sudo reboot

- Démarrez le puppetserver et activez-le pour qu'il se lance à chaque démarrage.

sudo systemctl start puppetserver systemctl status puppetserver

Installation Puppet Slave (agent)

Installer l'agent Puppet

Nous allons maintenant installer l'agent Puppet sur le serveur "puppetslave.oussama.galere". Pour ce faire, exécutez la commande ci-dessous pour installer l'agent Puppet.

```
sudo rpm -Uvh https://yum.puppet.com/puppet7-release-el-7.noarch.rpm
sudo yum update -y
sudo yum install puppet-agent -y
sudo reboot
```

Accédez au répertoire de configuration de puppet et modifiez le fichier puppet.conf.

Configurer l'agent Puppet

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

Collez ensuite la configuration ci-dessous:

```
[main]
certname = puppetslave.oussama.galere
server = puppetmaster.oussama.galere
environment = production
runinterval = 1h
```

Enregistrez et quittez le fichier.

Configurer Puppet Master

Après avoir installé le serveur Puppet, avant de le démarrer pour la première fois, utilisez la commande de configuration puppetserver ca pour créer une autorité de certification intermédiaire par défaut.

```
[centos@puppetmaster ~]$ puppetserver ca setup
Generation succeeded. Find your files in
/home/centos/.puppetlabs/etc/puppetserver/ca
[centos@puppetmaster ~]$ sudo su
```

Lister des machines qui ont un certificate signé

```
[root@puppetmaster centos]# puppetserver ca list --all
Signed Certificates:
    puppetmaster.oussama.galere
    (SHA256)
9E:4C:C9:5A:27:0F:B2:0E:64:7D:CE:60:73:D1:4F:B8:81:B6:3C:7B:3F:2C:7D:AD:47:E0:FE:6
5:3A:47:EB:CC alt names:
    ["DNS:puppet", "DNS:puppetmaster.oussama.galere"]
```

```
    authorization extensions:
    [pp_cli_auth: true]
```

cela signifi que je pourrai envoyé des scripts sur cette machines

Configurer Puppet Slave

```
Nous allons maintenant enregistrer l'agent de Puppet auprès du Master Puppet.
```

- Sur le shell de l'agent Puppet:

sudo /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true

```
Notice: /Service[puppet]/ensure: ensure changed 'stopped' to 'running'
service { 'puppet':
  ensure   => 'running',
  enable   => 'true',
  provider => 'systemd',
}
```

```
L'agent Puppet est maintenant en cours d'exécution sur le serveur Slave et il
tente de s'enregistrer auprès du Puppet Master.
```

- Revenez au shell du Master de Puppet.

Configurer Puppet Master

```
[root@puppetmaster centos]# puppetserver ca list --all
Requested Certificates:
    puppetslave.oussama.galere
    (SHA256)
7A:76:F0:B3:F0:EB:56:34:C8:74:75:F7:AF:B5:83:5F:CE:41:11:6A:2C:0C:A2:46:40:F2:DA:8
2:35:A3:D2:BB

Signed Certificates:
    puppetmaster.oussama.galere
    (SHA256)
9E:4C:C9:5A:27:0F:B2:0E:64:7D:CE:60:73:D1:4F:B8:81:B6:3C:7B:3F:2C:7D:AD:47:E0:FE:6
5:3A:47:EB:CC alt names:
    ["DNS:puppet", "DNS:puppetmaster.oussama.galere"]
    authorization extensions: [pp_cli_auth: true]
```

Ensuite, vous obtiendrez la demande de signature de certificat (CSR) en attente du serveur d'agents de Puppets 'puppetslave.oussama.galere'.

Ensuite, exécutez la commande ci-dessous pour signer le certificat.

- Exécutez la commande ci-dessous pour signer le certificat:

```
[root@puppetmaster centos]# puppetserver ca sign --certname
puppetslave.oussama.galere
Successfully signed certificate request for puppetslave.oussama.galere
```

Alors maintenant, l'agent de Puppet s'exécute sur le système et le certificat de l'agent a été signé par le Master Puppet.

Puppet Slave

Vérifier la configuration de l'agent Puppet

Une fois que le maître de Puppets a signé le fichier de certificat pour l'agent, exécutez la commande ci-dessous sur l'agent de Puppets pour vérifier la configuration.

[centos@puppetslave ~]$ sudo su

[root@puppetslave centos]# puppet agent --test

Vous devez voir un résultat comme ci-dessous:

```
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for puppetslave.oussama.galere
Info: Applying configuration version '1640187698'
Notice: Applied catalog in 0.01 seconds
```

L'agent Puppet a extrait la configuration du Puppet Master et l'a appliquée au serveur sans aucune erreur.

```
[root@puppetmaster centos]# ls /etc/puppetlabs/
code  puppet  puppetserver  pxp-agent
[root@puppetmaster centos]# ls /etc/puppetlabs/code/
environments  modules
[root@puppetmaster centos]# ls /etc/puppetlabs/code/environments/
production
[root@puppetmaster centos]# ls /etc/puppetlabs/code/environments/production/
data  environment.conf  hiera.yaml  manifests  modules
```

- manifests

Les programmes de Puppet sont appelés manifestes. Les manifestes sont composés de code Puppet et leurs noms de fichiers utilisent l'extension .pp. Le manifeste principal par défaut dans Puppet installé via apt est /etc/puppetlabs/code/environments/production/site.pp.

vi /etc/puppetlabs/code/environments/production/manifests/site.pp

```
node 'puppetslave.oussama.galere' {
        file { '/tmp/bonjour':
        ensure => 'directory',
        owner => 'root',
        group => 'root',
        mode => '0755',
}
        user { 'oussama':
        ensure => present,
        shell => '/bin/bash',
        home => '/home/oussama'
}
}
```

Info: Using environment 'production' Info: Retrieving pluginfacts Info: Retrieving plugin Info: Caching catalog for puppetslave.oussama.galere Info: Applying configuration version '1640191711' Notice: /Stage[main]/Main/Node[puppetslave.oussama.galere]/File[/tmp/bonjour]/ensure: created Notice: /Stage[main]/Main/Node[puppetslave.oussama.galere]/User[oussama]/ensure: created Notice: Applied catalog in 0.06 seconds