
Le Rapport de Projet «Mise en place d'un serveur Foreman/Puppet»

p1512310 LUCA Gheorghe

p1512677 PARSHUTO Hanna

Sommaire

Introduction	3
Matériel et Méthode	3
Étudier la théorie sur Foreman et Puppet en général	3
Travail en collaboration.	4
Configurer le réseau	5
Conditions nécessaires avant d' Installation de Foreman.	5
Installation de Foreman.	8
Installer la première machine virtuelle – client avec Ubuntu.	14
Installer le Puppet. Le lier avec Foreman.	18
Écrire les manifestes Puppet.	21
Installation d'autres machines avec CentOS et Debian.	23
Résultats et perspectives sur le projet	24
État final du projet	24
Améliorations et extensions possibles	24
Intérêt de ce projet	24
Conclusion	25

1. Introduction

Le but de notre projet était de mettre en place le serveur avec qui on installe :

- les systèmes d'exploitation sur les machines
- les logiciels différents (paquets)
- créer l'utilisateur à distance.

Pour cela, parmi des contraintes données, il fallait utiliser le logiciel Foreman et Puppet. Foreman est un serveur capable d'installer le système d'exploitation d'une machine, Puppet est un serveur agissant sur une machine disposant d'un système de base et capable de vérifier et modifier les configurations de cette dernière.

Pour atteindre ce but, on s'est proposé d'étudier quelques questions relatives au développement de notre projet :

- 1) Savoir plus sur Foreman. Le but, la modélisation et l'utilisation.
- 2) Comprendre le fonctionnement de Puppet.

2. Matériel et Méthode

Avant tout, nous avons découpé notre projet en quelques phases pour se répartir le travail au mieux. Nous avons divisé notre projet en neuf phases suivantes :

Phase 1 : Étudier la théorie sur Foreman et Puppet en général.

Phase 2: Se mettre d'accord sur les méthodes de travail en collaboration.

Phase 3 : Gérer le réseau

Phase 4 : Commencer par installation de Foreman.

Phase 5 : Installer la première machine virtuelle – client avec Ubuntu.

Phase 6: Installer le Puppet. Le lier avec Foreman.

Phase 7 : Écrire les manifestes Puppet.

Phase 8 : Installer d'autres machines avec RedHat et Debian.

Phase 9 : Proposer quelques améliorations.

2.1 Étudier la théorie sur Foreman et Puppet en général

Avant de commencer le projet nous avons étudié quelques questions théoriques sur le fonctionnement de Foreman et Puppet. Pour cela nous avons trouvé beaucoup d'information utile sur l'internet, y inclure sur Youtube, sur les forums différents. Comme il n'y avait pas beaucoup

d'informations sur le fonctionnement de Foreman parce que Foreman est un logiciel open-source, nous avons dû faire la recherche en trois langues : russe, français et anglais.

2.2 Travail en collaboration.

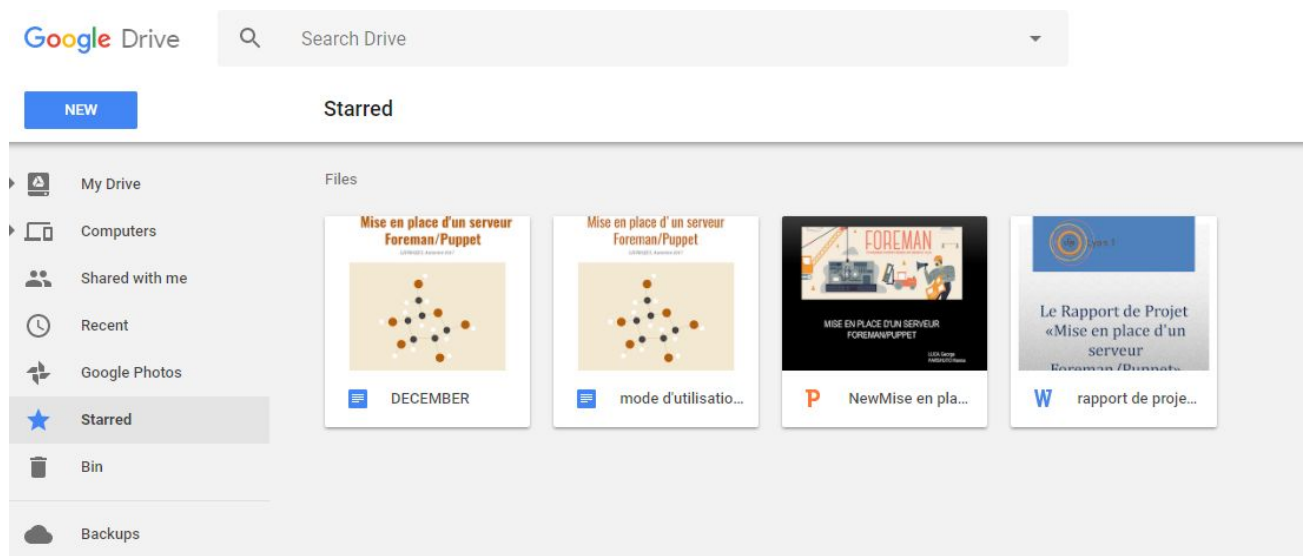
Pour bien travailler en équipe nous avons divisé les tâches du début.

Premièrement, Gheorghe a essayé d'installer le Foreman, en même temps Anna a cherché comment configurer les machines virtuelles. Mais en effet, nous avons compris que cela serait presque impossible de diviser les tâches, parce que chaque étape a été liée avec l'étape précédente. Donc chaque membre d'équipe devait avoir toutes les fonctionnalités pour pouvoir avancer le projet.

Au début, nous avons voulu travailler depuis chez nous, au lieu de rester à la faculté les horaires supplémentaires, mais en effet, nous avons passé énormément beaucoup de temps à la faculté, parce que nous avons eu beaucoup de problèmes avec la gestion de réseaux, les logiciels etc.. Nous avons eu besoin d'être à l'université pour pouvoir au moins comparer les erreurs apparues sur deux ordinateurs, pour proposer ces idées de résolution et aussi pour demander de l'aide dans les cas spécifiques de notre professeur Fabien Rico.

Pour réaliser les manifestes puppets, nous avons utilisé le système de gestion de versions – Git avec Github, pour que chaque membre puisse le consulter quand il veut.

Pour bien faire le rapport, nous avons utilisé Google Docs pour travailler en collaboration.



Travail en collaboration avec Google Docs

2.3 Configurer le réseau

Avant d'installer le Foreman il fallait bien gérer le réseau.

La configuration n'était pas facile, car nous avons eu deux cas différents:

La difficulté pour Gheorghe était que son serveur est installé sur la machine réelle. Mais en étant réelle, elle changeait les réseaux tout le temps(à la maison – un réseau, à la faculté –un autre, etc). Comme il est apparu, le Foreman n'était pas fait pour les réseaux qui se changent.

Pour Hanna, c'était un autre problème. Elle a travaillé avec Virtual Box. Son serveur était sur la machine virtuelle, et les clients aussi. Même si dans ce cas-là on évite le problème avec le changement de réseau, sa machine virtuelle n'a pas du tout été configurée : absence de hostname, adresse IP static, etc.

Du coup il fallait bien configurer toutes les paramètres, parce que si un paramètre n'était pas bon, nous avons dû réinstaller le Foreman, parce que si on voudrait changer quelque chose fait avant, la meilleure solution c'est de supprimer Foreman et de refaire tout du début.

De plus, il fallait bien choisir le réseau qu'on va utiliser pour le Virtual box. Parce qu'il y avait quelques possibilités (Host Only, ou Bridged, ou Internal) :

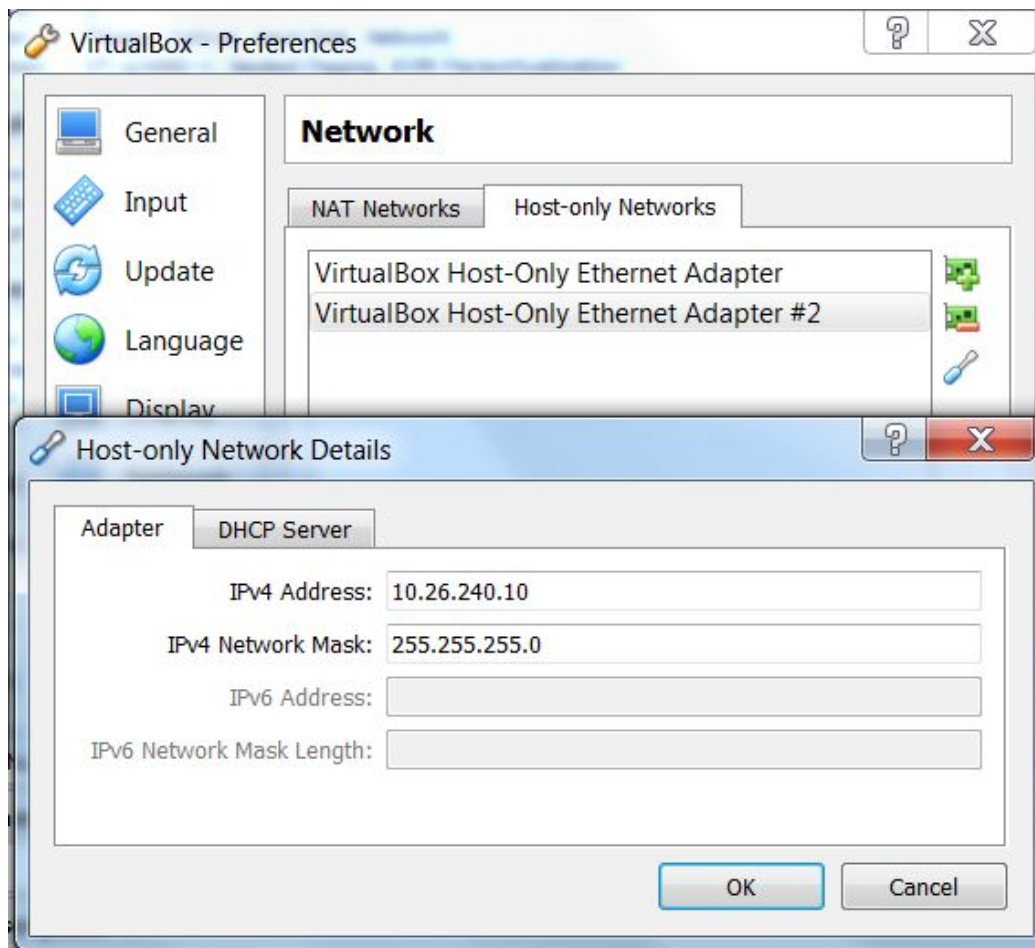
	VM ↔ Host	VM1 ↔ VM2	VM → Internet	VM ← Internet
Host-only	+	+	-	-
Internal	-	+	-	-
Bridged	+	+	+	+
NAT	-	-	+	Port forwarding
NAT Network	-	+	+	Port forwarding

Nous avons choisi Host Only avec Nat pour que les machines aient l'accès à l'internet et soient dans le même réseau. De plus ce type de réseau convient pour travailler comme avec les machines réelles - virtuelles (cas de Gheorghe), ainsi que pour les machines virtuelles - virtuelles(cas d'Hanna).

2.4 Conditions nécessaires avant d' Installation de Foreman.

Avant tout, il faut bien avoir une connexion Internet, une machine sous UNIX (on a choisi UBUNTU 16.04) et au moins Virtual Box / VMware pour installer les machines virtuelles.

Dans la configuration de Virtualbox il faut créer la carte réseau Host Only avec l'adress IP qui sera dans le même réseau que toutes les autres machines.



Si serveur est sur la machine virtuelle, alors il faut choisir dans les settings deux adaptateurs: Host Only et Nat. Sur Host Only il faut penser à désactiver DHCP. Parce que c'est Foreman qui gère DHCP.

Il faut ajouter notre adresse Ip dans le static. Pour cela, il faut modifier le fichier **/etc/network/interfaces.d**

```
#interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

#my network hostonly
auto enp0s3
iface enp0s3 inet static
address 10.26.240.11
network 10.26.240.0
netmask 255.255.255.0
broadcast 10.26.240.255
```

Après il faut exécuter cette commande.

```
Ifconfig #ici on doit verifier si enp0s3 est deja avec l'adress IP ecrit  
#sinon->faire  
sudo ifconfig enp0s3 10.26.240.11 netmask 255.255.255.0 up
```

Hostname est configuré sur Ubuntu dans deux fichiers: `/etc/hostname` et `/etc/hosts`. Fichier `/etc/hostname` définit le nom d'hôte réel du système, alors que `/etc/hosts` est utilisé pour la résolution du nom local.

Dans `/etc/hostname` il faut écrire votre nom de serveur(host).

```
anna.my-server.com
```

Dans `/etc/hosts` il faut écrire votre nom de serveur(host) avec l'adresse IP choisi.

À côté de nom de serveur il faut écrire le mot `puppet` pour pouvoir utiliser le puppet après. De même, il faut préciser l'adresse IP de Nat et son hostname.

```
127.0.0.1      localhost  
10.26.240.11   anna.my-server.com puppet puppet.my-server.com  
10.0.3.15      anna.nat.com  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```



C'est très important de bien choisir le hostname qui n'existe pas sur le Web. Parce qu'en travaillant avec le Puppet vous allez avoir les problèmes, quand votre machine installée va chercher votre serveur, et en effet elle va trouver une autre adress IP au lieu de la votre, parce que ce hostname existe déjà avec une autre adress IP .

Bon exemple : `server.giku-local.asd`

Error: Domain name, giku-local.asd cannot be mapped to an IP address.

Mauvais : `anna.my-server.com`

Domain Name	IP Address
my-server.com	69.172.201.153 New York, New York US Dosarrest Internet Security LTD (AS19324)

2.5 Installation de Foreman.

Nous avons comme le système d'exploitation Ubuntu 16.04. Installation était faite avec l'aide du site Foreman.org. Pour Ubuntu 16.04 (Xenial) on a fait:

Pour utiliser Puppet 4.x avec Puppet Agent and Puppet Server:

```
apt-get -y install ca-certificates
wget https://apt.puppetlabs.com/puppetlabs-release-pc1-xenial.deb
dpkg -i puppetlabs-release-pc1-xenial.deb
```

Activer le repo Foreman

```
echo "deb http://deb.theforeman.org/ xenial 1.15" >
/etc/apt/sources.list.d/foreman.list
echo "deb http://deb.theforeman.org/ plugins 1.15" >>
/etc/apt/sources.list.d/foreman.list
apt-get -y install ca-certificates
wget -q https://deb.theforeman.org/pubkey.gpg -O- | apt-key add -
```

Installer Foreman-installer

```
apt-get update && apt-get -y install foreman-installer
```

Après l'installation il faut obtenir cette information, que Foreman a bien été installé :

```
Success!
* Foreman is running at https://anna.my-server.com
  Initial credentials are admin / nPDVe3zJzkBVuWoH
* Foreman Proxy is running at https://anna.my-server.com:8443
* Puppetmaster is running at port 8140
The full log is at /var/log/foreman-installer/foreman.log
```



Sur cette étape nous avons eu quelques problèmes:

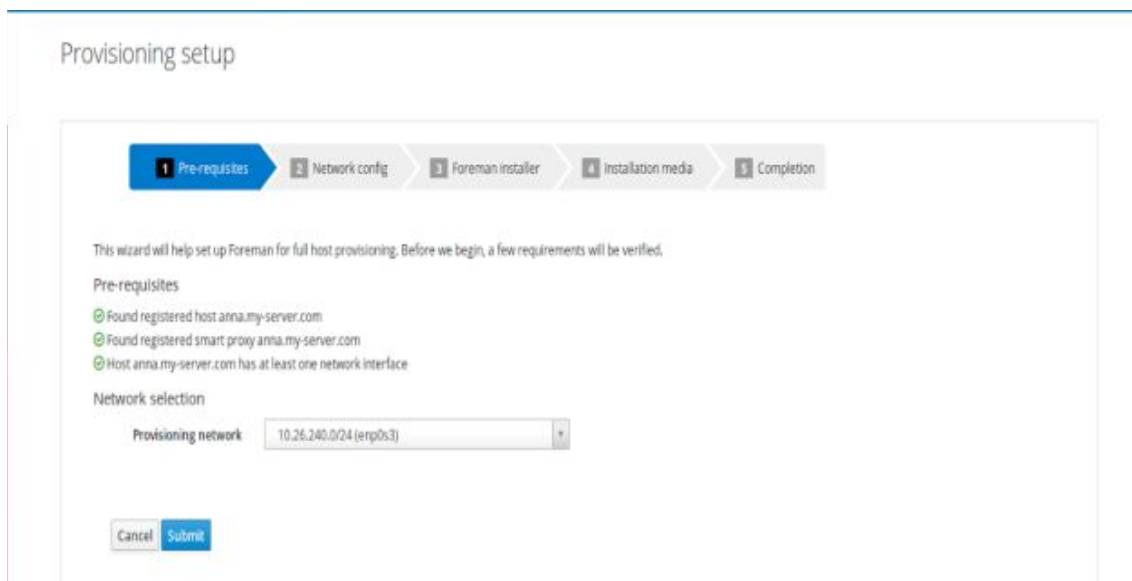
1) Problèmes avec l'installation du Puppet. Nous allons décrire cela plus précisément dans partie 2.7 (Installer le Puppet. Le lier avec Foreman). Pour bien voir que le problème a été fait à cause de Puppet, nous avons désactivé le puppet dans la configuration foreman: il faut lancer le foreman avec `foreman installer -i`, après choisir dans le menu point 5 - Configurer Puppet, et le désactiver. Après on sauvegarde les modifications faites et relance le Foreman. S'il n'y a pas d'erreurs - tout est bon.

- 2) Smart-proxy problème (problèmes avec des packages de ruby) il est apparu qu'il y avait deux packages installés de versions 1.6.0 et 2.3.1. Qui n'ont pas permis d'installer correctement foreman. Avec l'aide de commande `gem list` on a listé tous les packages installés.

Après il fallait trouver le gem avec deux versions, supprimer deux packages, faire `autoremove` et installer de nouveau ce package avec la commande `gem ruby nom_de_gem`.

- 3) Si on a choisi une adresse qui n'existe pas(server.giku-local.asd) il faut être sûr, que l'Ip ne se change pas. Parce que cela va générer des erreurs de non-compatibilité des certificats où Foreman va demander de faire toujours override d'adresse pour les clients.

Après l'installation du Foreman, sur la page web du Foreman, il faut aller voir la fenêtre **Provisioning Setup**.



Alors comme nous n'avons pas nos IP static, on utilise le DHCP.

Il faut installer provisioning avec DHCP. C'est-à-dire executer le script suivant.

```
foreman-installer \
  --enable-foreman-proxy \
  --foreman-proxy-tftp=true \
  --foreman-proxy-tftp-servername=10.26.240.11 \
  --foreman-proxy-dhcp=true \
  --foreman-proxy-dhcp-interface=enp0s3 \
  --foreman-proxy-dhcp-gateway= \
```

```
--foreman-proxy-dhcp-nameservers="10.26.240.11" \  
--foreman-proxy-dns=true \  
--foreman-proxy-dns-interface=enp0s3 \  
--foreman-proxy-dns-zone=my-server.com \  
--foreman-proxy-dns-reverse=240.26.10.in-addr.arpa \  
--foreman-proxy-dns-forwarders=127.0.1.1 \  
--foreman-proxy-foreman-base-url=https://anna.my-server.com \  
--foreman-proxy-oauth-consumer-key=zc2Hkq8TUZUG9ccHWAKdwZsWs38aR3yK \  
--foreman-proxy-oauth-consumer-secret=7X4DmPQeWpncthZGc7B8zikjuLn9iSsk
```



Problème avec DHCP.

Si on n'exécute pas le scripte donné par Foreman, c'est très probablement qu'il y aura une erreur.

Ce script permet de gérer les points-clés d'adresse Ips dans le réseau. C'est DHCP qui attribue nouvelle adresse IP pour la machine créée. Sans cela on ne pourrait pas bien installer automatiquement les nouvelles machines.

```
root@anna: /etc/dhcp  
root@anna: /etc/dhcp x root@anna: ~ x +  
GNU nano 2.5.3 File: dhcpd.conf~  
  
# my-server.com  
subnet 10.26.240.0 netmask 255.255.255.0 {  
    Range 10.26.240.20 10.26.240.30;  
    router 10.26.240.11;  
  
    Default-lease-time 220 ;  
    Max-lease-time 360 ;  
    option subnet-mask 255.255.255.0;  
}  
  
root@anna: /etc/dhcp  
root@anna: /etc/dhcp x root@anna: ~ x +  
GNU nano 2.5.3 File: dhcpd.conf  
  
    filename "grub2/bootx64.efi";  
} elsif option architecture = 00:09 {  
    filename "grub2/bootx64.efi";  
} else {  
    filename "pxelinux.0";  
}  
  
log-facility local7;  
  
include "/etc/dhcp/dhcpd.hosts";  
  
# my-server.com  
subnet 10.26.240.0 netmask 255.255.255.0 {  
  
    option subnet-mask 255.255.255.0;  
}
```

Même si on écrit dans le fichier de configuration **/etc/dhcp/dhcpd.conf** l'information

manquant, à lancement prochain, Foreman va supprimer cette information!

Si on n'a pas ce script de Foreman, la seule possibilité de gérer le DHCP, c'est d'aller dans les paramètres de Foreman avec `foreman installer -i`, après choisir dans le menu le point 4 - Configurer Foreman-proxy, et écrire toute l'information manquante. Après on sauvegarde les modifications faites et relance le Foreman.

Ensuite il faut installer les Smart Proxies, Subnet et Domain.

Smart Proxies

Name	URL	Features	Status	Actions
anna.server.my.new	https://anna.server.my.new:8443	Logs, Puppet, Puppet CA, and TFTP		<input type="button" value="Edit"/>

Displaying 1 entry

Subnets

Name	Network address	Domains	VLAN ID	DHCP Proxy	Actions
reseaubhostonly	10.26.240.0/24	server.my.new			<input type="button" value="Delete"/>
reseanat	10.0.3.0/24	nat.new			<input type="button" value="Delete"/>

Displaying all 2 entries

la Machine Virtuelle (host), il faut préciser deux réseaux : pour HostOnly et Nat.

Name

reseaubhostonly
reseanunat

Subnet
Domains
Proxies
Parameters

Name *

reseaubhostonly

Protocol *

☒ IPv4
☐ IPv6

Network address *

10.26.240.0

Network prefix *

24

Suffix or prefix length for this subnet, e.g. 32

Network mask *

255.255.255.0

Netmask for this subnet

Gateway address

10.26.240.11

Optional: Gateway for this subnet

Primary DNS server

10.26.240.11

Optional: Primary DNS for this subnet

Secondary DNS server

Optional: Secondary DNS for this subnet

IPAM ⓘ

DHCP

Start of IP range

10.26.240.20

Optional: Starting IP Address for IP auto suggestion

End of IP range

10.26.240.50

Optional: Ending IP Address for IP auto suggestion

VLAN ID

Optional: VLAN ID for this subnet

Boot mode

DHCP

Default boot mode for interfaces assigned to this subnet, applied to hosts from provisioning templates

Name

reseaubhostonly
reseanunat

Subnet
Domains
Proxies
Parameters

Domains

All items
Filter
+

nat.new

Selected items
-

server.my.new

Domains in which this subnet is part

Name

reseaubhostonly
reseanunat

Subnet
Domains
Proxies
Parameters

TFTP Proxy

anna.server.my.new
x

TFTP Proxy to use within this subnet

Name

reseaubhostonly

reseanaut

Subnet

Domains

Proxies

Parameters

Name *

reseanaut

Protocol *

IPv4

IPv6

Network address *

10.0.3.0

Network prefix *

24

Suffix or prefix length for this subnet, e.g. 32

Network mask *

255.255.255.0

Netmask for this subnet

Gateway address

Optional: Gateway for this subnet

Primary DNS server

Optional: Primary DNS for this subnet

Secondary DNS server

Optional: Secondary DNS for this subnet

IPAM ⓘ

DHCP

Start of IP range

10.0.3.10

Optional: Starting IP Address for IP auto suggestion

End of IP range

10.0.3.50

Optional: Ending IP Address for IP auto suggestion

VLAN ID

Optional: VLAN ID for this subnet

Boot mode

DHCP

Default boot mode for interfaces assigned to this subnet, applied to hosts from provisioning templates

Domains

Filter ...

Q Search

Create Domain

Help

Description	Hosts	Actions
domain for nat	0	Delete
domain for anna.server.my.new	1	Delete

Displaying all 2 entries

Description

domain for nat

domain for anna.server.my.new

Domain

Parameters

DNS domain *

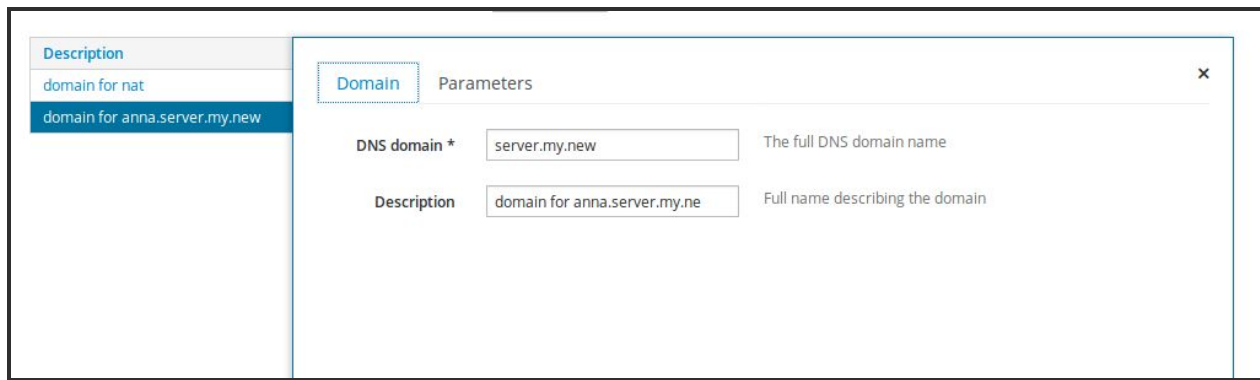
hat.new

The full DNS domain name

Description

domain for nat

Full name describing the domain

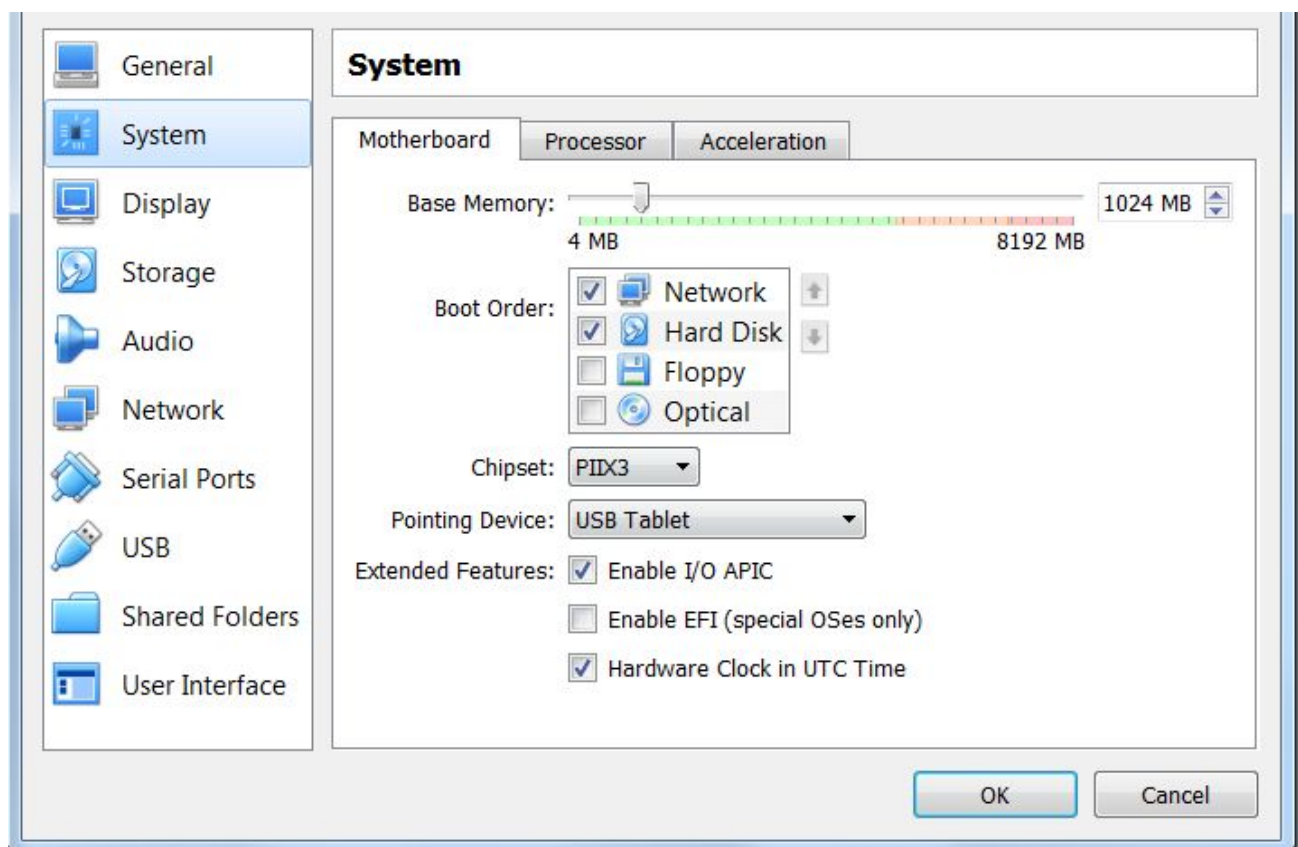


2.6 Installer la première machine virtuelle – client avec Ubuntu.

Pour installer la première machine, il faut:

- 1) savoir l'adresse MAC de cette machine.

Pour le travail avec Virtualbox il faut choisir dans les paramètres les périphériques de démarrage et leurs ordres: Network - pour que les machines puissent se communiquer avec serveur et Hard Disc pour écrire les modifications quelque part.



L'adresse MAC se trouve dans la fenêtre Network:

Ensuite, sur Foreman il faut aller sur **Hosts-Create Host** et bien remplir toutes les fenêtres.

Dans la partie **Interfaces** il faut saisir l'adress MAC de machine créée. Ici il faut vérifier si Foreman bien propose les nouvelles adresses IPv4.

Interface enp0s3

Type: Interface

MAC address: 08:00:27:3b:07:96

Device identifier: enp0s3

DNS name: ubuntu

Domain: my-server.com

IPv4 Subnet: 10.26.240.0/24

IPv6 Subnet: No subnets

IPv4 address: 10.26.240.27
[Suggest new](#)

IPv6 address:

Managed: ☒

Primary: ☒

Provision: ☒

Virtual NIC: ☐

Cancel Ok

Dans la fenêtre **Operating System**, il faut choisir le système d'exploitation souhaité, ainsi que le **Media**(ou va se trouver le fichier d'installation ou cela sera un lien web), de plus il faut bien gérer les templates. Pour chaque système d'exploitation ils sont différents. Pour Ubuntu il faut:

- PXELinux
- Finish Template
- iPXE template
- Provisionning template.

Host	Puppet Classes	Interfaces	Operating System	Parameters	Additional Information
Architecture *					
x86_64					
Operating system *					
Ubuntu 16.04.3 LTS					
Media *					
Ubuntu mirror					
Partition table *					
Preseed default					
PXE loader					
PXELinux BIOS					
Custom partition table					
<p>What ever text(or ERB template) you use in here, would be used as your OS disk layout options If you want to use the partition table option, delete all of the text from this field</p>					
Root password *					
<div>*****</div> <div>Password must be 8 characters or more</div>					
Provisioning templates					
<div>Resolve</div> <div>Display the templates that will be used to provision this host</div> <div> <div> <div>✓</div> <div>Templates resolved for this operating system</div> <div> <div>PXELinux template Template</div> <div>Preseed default PXELinux</div> </div> <div> <div>Finish template Template</div> <div>Preseed default finish</div> </div> <div> <div>IPXE template Template</div> <div>Preseed default IPXE</div> </div> <div> <div>Provisioning template Template</div> <div>Preseed default</div> </div> </div> </div>					

Quand toutes les champs sont remplies, nous pouvons lancer la nouvelle machine. Au début une fenêtre apparait qui disent que machine a pu se connecter avec le serveur et l'installation se commence.

```

NEW machine [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
iPXE (PCI C8:00.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+ -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS TFTP HTTP PXE PXEXT Menu

net0: 08:00:27:3b:07:96 using 82540em on PCI00:03.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086101)]
Waiting for link-up on net0... ok
DHCP (net0 08:00:27:3b:07:96)..... ok
net0: 10.26.240.27/255.255.255.0 gw 10.26.240.11
Next server: 10.26.240.11
Filename: pxelinux.0
tftp://10.26.240.11/pxelinux.0... ok

PXELINUX 6.03 PXE 20151222 Copyright (C) 1994-2014 H. Peter Anvin et al
Loading boot/Ubuntu-16.04-x86_64-linux... ok
Loading boot/Ubuntu-16.04-x86_64-initrd.gz..._

```

2.7 Installer le Puppet. Le lien avec Foreman.

Si on regarde la table de compatibilité de mois de décembre 2017, on peut voir que la meilleur version pour Puppet doit être 3.8.

3.1.3 Puppet Compatibility

Foreman integrates with Puppet and Facter in a few places, but generally using a recent, stable version will be fine. The exact versions of Puppet, Puppet Server and Facter that Foreman is compatible with are listed below.

Puppet version	Foreman installer	Smart proxy	Report/fact processors	External node classifier
0.25.x	Not supported	Untested	Untested	Supported *
2.6.0 - 2.6.5	Not supported	Untested	Untested	Supported *
2.6.5+	Not supported	Supported	Supported	Supported
2.7.x	Not supported	Supported	Supported	Supported
3.0.x - 3.5.x	Not supported	Supported	Supported	Supported
3.6.x - 3.8.x	Supported	Supported	Supported	Supported
4.x (AIO)	Supported	Supported	Supported	Supported
4.x (non-AIO)	Untested	Not Supported	Untested	Untested

Par contre pour le mois de janvier, ce tableau a été changé.

3.1.3 Puppet Compatibility

Foreman integrates with Puppet and Facter in a few places, but generally using a recent, stable version will be fine. The exact versions of Puppet, Puppet Server and Facter that Foreman is compatible with are listed below.

Puppet version	Foreman installer	Smart proxy	Report/fact processors	External node classifier
0.25.x	Not supported	Untested	Untested	Supported *
2.6.0 - 2.6.5	Not supported	Untested	Untested	Supported *
2.6.5+	Not supported	Supported	Supported	Supported
2.7.x	Not supported	Supported	Supported	Supported
3.0.x - 3.5.x	Not supported	Supported	Supported	Supported
3.6.x - 3.8.x	Not supported	Supported	Supported	Supported
4.x (AIO)	Supported	Supported	Supported	Supported
4.x (non-AIO)	Untested	Not Supported	Untested	Untested
5.x (AIO)	Supported	Supported	Supported	Supported
5.x (non-AIO)	Untested	Not Supported	Untested	Untested



Les seules erreurs qui peuvent apparaître sont causées par la non-compatibilité de Puppet avec Foreman.

Cela arrive, parce que Foreman installe toujours la version plus récente du Puppet.

Pour résoudre il faut installer une version compatible:

```
apt-cache policy puppet  
apt-get install -y puppet=version_choisi
```

Maintenant le Puppet est installé.

Il faut faire la commande `puppet agent --test`. Pour la première fois, il y a toujours une erreur qui apparaît. C'est normale, il faut continuer.

```
anna@anna:~$ puppet agent --test  
Info: Creating a new SSL key for anna.my-server.com  
Error: Could not request certificate: Failed to open TCP connection to puppet:8140 (getaddrinfo: No address associated with hostname)  
Exiting; failed to retrieve certificate and waitforcert is disabled
```

Ensuite, il est nécessaire d'installer le module NTP. NTP est un protocole qui permet de synchroniser, via un réseau informatique, l'horloge locale d'ordinateurs sur une référence d'heure.

```
puppet module install -i /etc/puppet/environments/production/modules saz/ntp
```

Le résultat d'installation est présenté sur cette image:

```
anna@anna:~$ sudo puppet module install -i /etc/puppet/environments/production/modules saz/ntp  
Notice: Preparing to install into /etc/puppet/environments/production/modules ..  
.  
Notice: Created target directory /etc/puppet/environments/production/modules  
Notice: Downloading from https://forgeapi.puppetlabs.com ...  
Notice: Installing -- do not interrupt ...  
/etc/puppet/environments/production/modules  
└─ saz-ntp (v2.5.0)
```



Ici, avant de commencer l'installation de module NTP, c'est très important de vérifier l'heure de l'ordinateur. Avec la commande `timedatectl`.

```
anna@anna:~$ timedatectl
Local time: mar. 2018-01-09 16:15:50 CET
Universal time: mar. 2018-01-09 15:15:50 UTC
RTC time: mar. 2018-01-09 15:15:51
Time zone: Europe/Paris (CET, +0100)
Network time on: yes
NTP synchronized: yes
RTC in local TZ: no
```

Si toutes les horloges sont synchronisées, on peut bien continuer. Sinon il faut configurer l'heure.

Quand NTP est installé, il faut faire ce script, qui pourra lancer puppet agent correctement sans erreurs.

```
sed -i s/START=no/START=yes/g /etc/default/puppet

sed -i '/\var\log\puppet/a \server=anna.my-server.com'
/etc/puppet/puppet.conf

sed -i 's/templatedir/#templatedir/' /etc/puppet/puppet.conf
```

Ensuite on relance Foreman, et très probablement il y aura des erreurs. Donc il faut faire `puppet agent --test`.

Si tout est bon, il faut vérifier les certificats signés avec la commande `puppet cert --list --all`. Le resultat d'execution est:

```
root@anna:~# puppet cert --list --all
+ "anna.my-server.com" (SHA256) 1B:C2:77:40:A8:D9:F6:42:35:8F:6B:5B:F7:0A:81:1
7:CB:C7:3C:4C:C0:31:DF:61:6A:FD:3E:9B:10:4C:55:52 (alt names: "DNS:anna.my-serve
r.com", "DNS:puppet", "DNS:puppet.my-server.com")
+ "client.my-server.com" (SHA256) 56:B8:8C:3A:40:09:C3:18:23:1E:3E:FE:B2:27:D0:B
3:7E:19:66:CA:5E:C4:2E:FD:C0:B5:C2:D6:3C:81:AE:86
```

ou + est le certificat signé, - n'est pas signe. On peut signer les certificats sur la page Web du Foreman.



Si jamais après le lancement du `puppet agent --test` ou `puppet cert --list --all`, il y a une erreur qui dit que puppet n'a pas été trouvé ou le fichier `/usr/bin/puppet` n'a pas été trouvé, Il faut ajouter le chemin de Puppet dans `$PATH`. `export PATH=/opt/puppetlabs/bin:$PATH`

ou encore mieu dans votre `.profile` or `.bashrc`.

Finalement il faut redemarrer le systeme, lancer le Foreman, et le Puppet doit marcher.

On vérifie avec la commande `puppet agent --test`


```

Installing                                     Done [100%]
Success!
* Foreman is running at https://anna.my-server.com
  Initial credentials are admin / nPDVe3zJzkBVuWoH
* Foreman Proxy is running at https://anna.my-server.com:8443
* Puppetmaster is running at port 8140
The full log is at /var/log/foreman-installer/foreman.log
root@anna:~# puppet cert --list --all
+ "anna.my-server.com" (SHA256) 1B:C2:77:40:A8:D9:F6:42:35:8F:6B:5B:F7:0A:81:1
7:CB:C7:3C:4C:C0:31:DF:61:6A:FD:3E:9B:10:4C:55:52 (alt names: "DNS:anna.my-serve
r.com", "DNS:puppet", "DNS:puppet.my-server.com")
+ "client.my-server.com" (SHA256) 56:B8:8C:3A:40:09:C3:18:23:1E:3E:FE:B2:27:D0:B
3:7E:19:66:CA:5E:C4:2E:FD:C0:B5:C2:D6:3C:81:AE:86
root@anna:~# puppet agent --test
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for anna.my-server.com
Info: Applying configuration version '1515514006'
Notice: Applied catalog in 0.13 seconds
root@anna:~# █

```

2.8 Écrire les manifestes Puppet.

Après l'installation du Puppet et son configuration, on va essayer d'écrire des règles(manifestes : test.pp) Puppet pour gérer un système d'exploitation.

Le code Puppet est composé par des ressources de déclaration. Une ressource décrit l'état du système (un fichier qui doit exister ou un package qui doit être installé).

Premier exemple: création d'un nouvel utilisateur

```

user { 'mitchell':
  ensure => present,
  uid    => '1000',
  gid    => '1000',
  shell  => '/bin/bash',
  home   => '/home/mitchell'
}

```

Nous pouvons également installer des packages :

```

$packages_list = ['git', 'gcc', 'g++', 'geany', 'valgrind']
package { $packages_list:
  ensure => 'installed',
}

```

Pour appliquer le code plus rapide on peut utiliser les Classes. Une classe peut être appelé n'importe où dans le code. De plus Foreman bien voit toutes les changements faites, et après c'est possible d'affecter une classe pour un client, l'autre - pour un autre client.

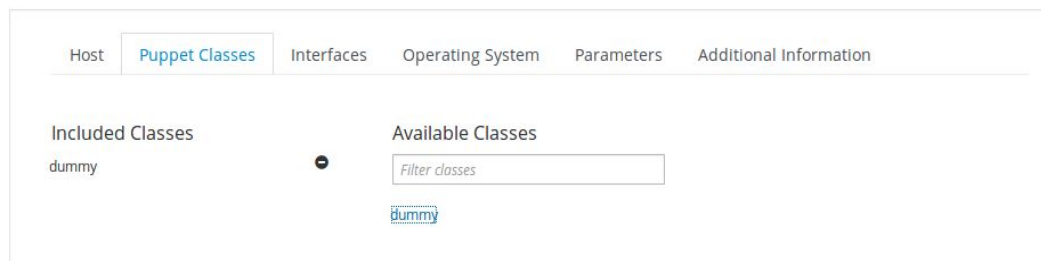
```

/modules/dummy/manifests - gedit
1 class dummy {
2   file { ["/etc/puppet.txt":
3     owner => root,
4     group => root,
5     mode => 644,
6     source => "puppet:///dummy/puppet.txt"
7   ]
8 }

```

Après, pour pouvoir utiliser les manifests avec Foreman il faut les sauvegarder dans “**~/etc/puppet/modules/nomDuClasse/manifest/file.pp**” et relancer foreman-install pour mettre à jour les modifications si les classes n'apparaissent pas.

Ensuite on trouve ces manifests dans Puppet Classes de chaque host.





Il faut toujours bien penser à ajouter les classes ntp-ntp dans les Puppet Classes de clients.

Edit ubuntu.my-server.com

Host

Puppet Classes

Interfaces

Operating System

Parameters

Additional Information

Included Classes

ntp
forall
programmer

Available Classes

Filter classes

+ delete_installed
+ forall
+ ntp

+ programmer
+ sysadmin

Finalement pour appliquer les classes de serveur, il faut faire la commande `puppet agent --test` sur client.

Supposons que le package était:

```
class dummy{
  package {'vim':
    ensure => 'installed',
  }
}
```

Le résultat d'installation de ce package est bien visible sur cette image.

```
root@ubuntu:~# dpkg -l vim
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version          Architecture    Description
+++-=====
un vim            <none>          <none>          (no description available)
root@ubuntu:~# puppet agent --test
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for ubuntu.my-server.com
Info: Applying configuration version '1515077192'
Notice: /Stage[main]/Foo/Package[vim]/ensure: ensure changed 'purged' to 'present'
Notice: Finished catalog run in 3.09 seconds
root@ubuntu:~# dpkg -l vim
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version          Architecture    Description
+++-=====
ii vim            2:7.4.1689-3ub amd64          Vi Improved - enhanced vi editor
root@ubuntu:~#
```

2.9 Installation d'autres machines avec CentOS et Debian.

Pour installer **CentOs** ou **OpenSUSE** sur la machine, il faut refaire les manipulations écrit dans 2.6. (Installer la première machine), sauf que:

- il faut créer un autre système d'exploitation CentOS(OpenSUSE) dans la fenêtre **Operating System**;
- il faut bien gérer les **Templates** et **Partition table**.

	CentOS	OpenSUSE
Partition table	Kickstart default	AutoYaST LVM
Provisioning template	Kickstart default	AutoYaST default
PXELinux template	Kickstart default PXELinux	AutoYaST default iPXE
Finish template	Kickstart default finish	Preseed default PXELinux
iPXE template	Kickstart default iPXE	AutoYaST default user data

3. Résultats et perspectives sur le projet

3.1. État final du projet

À ce jour, les différents points fonctionnant dans le projet sont les suivants:

- 1) Configuration de Foreman qui permet d'effectuer les tâches nécessaires.
- 2) Installation des machines avec 3 OS différents: Ubuntu, CentOS, OpenSUSE.
- 3) Gestion de packages, utilisateurs, groupes avec les manifestes Puppet,y inclure manifestes "simples" et "dur" comme installation de NetBeans.

3.2. Améliorations et extensions possibles

- 1) Installer le Windows via Foreman. Windows reste toujours l'un de systèmes le plus utilisé. C'est pourquoi c'est très important de pouvoir installer sur les machines virtuelles ou réelles le Windows. Installation de Windows ne sera pas très facile, parce que Foreman ne propose pas le lien pour installer le Windows depuis le miroir web. Cela signifie que le fichier d'installation doit être quelque part sur le serveur. Pour savoir plus comment installer le Windows, il existe une vidéo sur Youtube avec le tutoriel.<https://www.youtube.com/watch?v=R8-0KAkoPc>
- 2) Open Stack. C'est possible d'utiliser OpenStack à la place de Virtualbox, cela permet de gagner en mémoire. Avec OpenStack des machines Virtuelles seront accessibles à tout le monde. Aussi avec l'aide de Foreman on peut les gérer comme des ordinateurs simples.
- 3) Installation de machines physiques.

Pour réaliser cette installation il faut disposer d'une machine physique avec une carte réseaux à l'aide de laquelle on va faire la connexion entre le serveur et le client. Un vrai exemple est les ordinateurs qu'on peut les trouver à Nautibus(Bâtiment d'informatique à UCBL). Chaque ordinateur est connecté au serveur de l'université et chacun d'entre eux peut être manipulé à distance.

3.3. Interet de projet

- 1) Économie du temps et d'argent :
 - Gestion des ordinateurs à distance. L'entreprise avec des bureaux dans quelques pays)

Cette configuration Foreman-Puppet permet de gérer les ordinateurs à distance. C'est très pratique pour les entreprises où il faut installer les systèmes d'exploitation pour les nouveaux ordinateurs.

- Installation manuelle contre installation automatique. Ce projet ne nécessite pas d'administrateur de Systèmes à temps plein.

Au lieu de passer et insérer la clé USB ou disque dans chaque ordinateur pour administrateur de systèmes, c'est plus facile de gérer le parc d'ordinateurs avec l'aide de Foreman.

- Paquets différents pour chacun utilisateur qui correspond à ses besoins

Pour chaque client c'est possible de personnaliser les logiciels qu'il utilise. Par exemple, comme nous avons fait, c'est assez simple de créer deux classes avec Puppet : programmeur et administrateur de systèmes. Après il faut juste appliquer cette classe pour les machines précises.

2) Ordinateur-serveur sur lequel Foreman est installé, peut être utilisé comme "la clé USB" avec toutes les distributions de unix ou Windows.

3) Sécurité.

Les nouveaux mis à jour d'urgence s'installent facilement sur toutes les machines (ex. logiciel malveillant Wanna Cry 2017 à cause beaucoup de problèmes. La rapidité de décisions ici est très importante).

4. Conclusion

Ce projet a été très enrichissant pour nous car il nous a permis de faire les premiers pas dans la gestion et supervision d'un parc des ordinateurs. Dans un premier temps, nous avons dû nous documenter sur les différentes pages web pour comprendre qu'est-ce que c'est un parc d'ordinateurs et comment le gérer. Néanmoins, il a fallu se familiariser avec la méthode et les logiciels, ce qui prend un peu de temps. De plus, étant un logiciel Open Source la documentation est peu fournie, mais l'aide des forums ainsi que divers tutoriels faits par des utilisateurs aguerris facilitent cet apprentissage.

Premièrement nous avons installé Foreman qui nous a permis de faire les connexions entre serveurs - client et d'installer un système d'opération sur chacun des clients. Après pour enrichir les clients avec des logiciels, utilisateurs, les groupes et d'autres fichiers nous avons utilisé le puppet. Puppet permet de livrer automatiquement les packages, les installer et puis mettre à jour, contrôler le système d'exploitation et la sécuriser.

Pour diversifier notre projet, nous avons décidé d'utiliser le projet pour l'installation des machines à des différents clients(programmeur, administrateur du système, utilisateur général). Pour chacun des clients on a essayé d'installer les logiciels et les systèmes d'opération dont ils ont besoin.

En travaillant sur ce projet nous n'avons pas remarqué beaucoup de logiciels qui permettent de mettre en place un parc d'ordinateurs. Ce qu'on a réussi trouvé, le logiciel qui pourra faire presque le même travail que Foreman, c'est Windows SCCM. Windows SCCM est proposé par MICROSOFT. Nous avons décidé de faire une petite comparaison entre Foreman et Windows SCCM pour voir quel logiciel serait le meilleur pour ce genre de projet.

Windows SCCM vs Foreman :

1)Foreman est un logiciel open-source contre SCCM qui est une propriété de MICROSOFT.

2)SCCM est compatible avec WINDOWS, on peut dire qu'ils ont une bonne liaison parce que les deux sont développés par MICROSOFT, mais SCCM on ne peut pas gérer d'autres systèmes d'opération.

Foreman peut gérer LINUX, WINDOWS et aussi MAC OS,

3) SCCM et Foreman - facile à apprendre et à utiliser

4) SCCM est une propriété, donc il faut payer pour l'utiliser.

Foreman open-source et aussi gratuit.

Après tous les 4 points permettent d'observer que Foreman est mieux en comparaison avec SCCM. Mais contrairement à SCCM, la documentation pour utiliser Foreman est peu fournie.

En conclusion, nous sommes tout de même contents de ces mois de travail. Nous nous rendons compte qu'il a fallu apporter beaucoup du sien chaque jour pendant les mois et qu'il y a eu des moments difficiles quand rien ne marchait pas comme on le voulait, comme des moments de réussite où l'on réussissait à faire exactement ce que l'on voulait avec les logiciels.

Un grand merci pour notre chef du projet Monsieur Fabien RICO, qui a déposé un grand effort pour nous aider à comprendre comment doit marcher en général le projet et aussi de nous aider à résoudre les problèmes qui ont bloqué le déroulement du projet.