Foreman (Draft)

- Préparation
- Installation du serveur Foreman (mono-nœud)
- Configuration HAProxy
- Installation de l'agent Puppet sur les nœuds clients
- Vérifications et dépannage
- Configurations avancées
- Sauvegarde et maintenance
- Architecture de Foreman

Préparation

Prérequis

- Serveur Rocky Linux 9 avec au moins 4 Go de RAM et 2 vCPUs
- Accès root
- Nom de domaine configuré : foreman.mondomaine.com
- Firewall configuré pour autoriser les ports nécessaires

Checklist avant de commencer

- ☐ **Serveur HAProxy** : IP = `<IP_HAPROXY>` (serveur mutualisé séparé)
- ☐ **Serveur Foreman** : IP = `<IP_SERVEUR_FOREMAN>`
- ☐ Les deux serveurs peuvent communiquer entre eux
- ☐ DNS configuré : `foreman.mondomaine.com` → `<IP_HAPROXY>`
- ☐ Certificats SSL disponibles (ou génération prévue)
- ☐ Accès root sur les deux serveurs

Informations à préparer

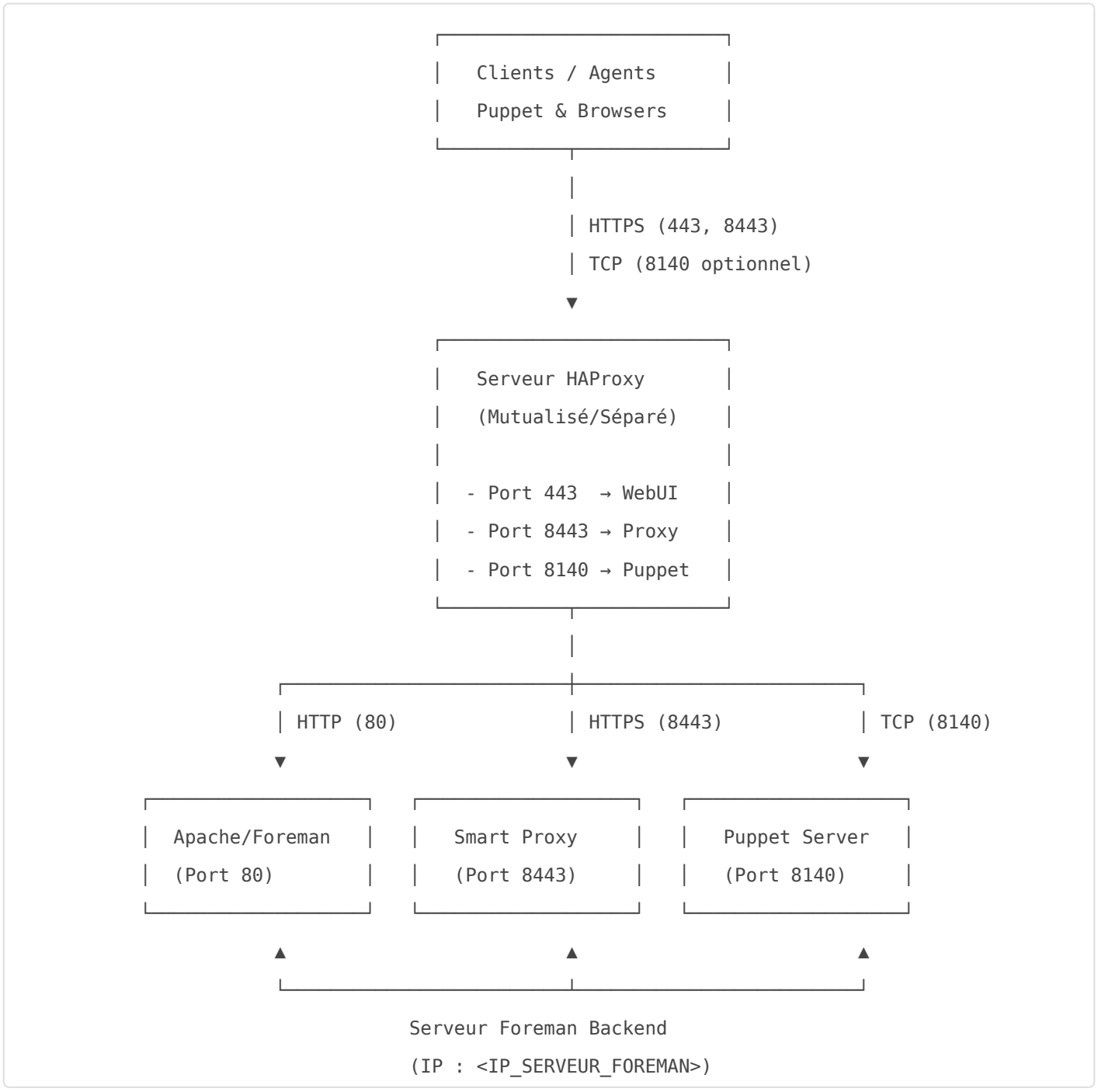
Information	Valeur à renseigner
IP HAProxy	`<IP_HAPROXY>`
IP Foreman	`<IP_SERVEUR_FOREMAN>`
Nom de domaine	`foreman.mondomaine.com`
Organisation	`MonOrganisation`
Location	`MonDatacenter`
Mot de passe admin Foreman	`MotDePasseSecurise`

Architecture

- **Serveur HAProxy** (serveur mutualisé séparé) : Frontend (ports 443 et 8443)
- **Serveur Foreman** : Backend (ports 80, 8443, 8140)
- Interface Web : https://foreman.mondomaine.com (HAProxy → Foreman:80)
- Smart Proxy : https://foreman.mondomaine.com:8443 (HAProxy → Foreman:8443)

Important : HAProxy et Foreman sont sur des serveurs distincts. HAProxy agit comme reverse proxy SSL/TLS et redirige le trafic vers le serveur Foreman backend.

Schéma de l'architecture



Flux de communication

1. **WebUI** : Client → HAProxy:443 (HTTPS) → Foreman:80 (HTTP)
2. **Smart Proxy** : Client → HAProxy:8443 (HTTPS) → Foreman:8443 (HTTPS)
3. **Puppet** : Agent → HAProxy:8140 (TCP) → Foreman:8140 (TCP) OU Agent → Direct Foreman:8140

Installation du serveur Foreman (mono-nœud)

Configuration du hostname et du fichier hosts

```
# Configuration du hostname
sudo hostnamectl set-hostname foreman.mondomaine.com

# Vérification
hostnamectl
```

Editer le fichier ***/etc/hosts*** et configurer les entrées de la manières suivantes :

```
127.0.0.1   localhost localhost.localdomain
<VOTRE_IP_SERVEUR>   foreman.mondomaine.com foreman
```

Installation des prérequis

```
# Mise à jour du système
sudo dnf update -y

# Installation des dépendances de base
sudo dnf install -y vim wget curl net-tools
```

Optimisation du noyau (optionnel)

Créer le fichier "/etc/sysctl.d/99-foreman.conf" pour optimiser les paramètres du noyau avec le contenu suivant :

```
# ================================================================
# CONFIGURATION SYSCTL POUR SERVEUR FOREMAN / PUPPET
# ================================================================

# --- RÉSEAU : Désactivation de l'IPv6 ---
net.ipv6.conf.all.disable_ipv6 = 1
```

```
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

# --- RÉSEAU : Optimisations TCP pour Puppet & API ---
# Augmente la file d'attente pour gérer les pics de connexions des agents
net.core.somaxconn = 1024
# Augmente la plage de ports pour éviter l'épuisement lors de gros déploiements
net.ipv4.ip_local_port_range = 10240 65535
# Accélère la réutilisation des sockets en état TIME_WAIT
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 15

# --- MÉMOIRE : Optimisation PostgreSQL & Java ---
# Réduit l'utilisation du swap pour privilégier la RAM (très important)
vm.swappiness = 10
# Définit le seuil à partir duquel les données sales sont écrites sur disque
vm.dirty_ratio = 15
vm.dirty_background_ratio = 5
# Nécessaire pour les processus Java (Candlepin/Elasticsearch)
vm.max_map_count = 262144

# --- SYSTÈME DE FICHIERS : Gestion des dépôts (Pulp) ---
# Augmente la limite globale de fichiers ouverts
fs.file-max = 2097152

# --- SÉCURITÉ ---
# Empêche les attaques de type IP spoofing
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
# Désactive l'acceptation des redirections ICMP (prévention MITM)
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
```

Appliquer les paramètres sans redémarrer :

```
sudo sysctl --system
```

Désactivation de SELinux (optionnel mais recommandé pour débuter)

```
sudo setenforce 0
sudo sed -i 's/^SELINUX=.*/SELINUX=permissive/' /etc/selinux/config
```

> **Note** : En production, il est préférable de configurer SELinux correctement plutôt que de le désactiver.

Configuration du firewall

```
# Autoriser les ports pour Foreman (backend)
# IMPORTANT : Le trafic viendra du serveur HAProxy, pas directement des clients

# Autoriser HTTP depuis HAProxy (le SSL est terminé sur HAProxy)
sudo firewall-cmd --permanent --add-port=8443/tcp  # Smart Proxy
sudo firewall-cmd --permanent --add-port=443/tcp  # Nginx Local en HTTPS

# Puppet Server
sudo firewall-cmd --permanent --add-port=8140/tcp

# Services optionnels
sudo firewall-cmd --permanent --add-port=53/tcp    # DNS (si activé)
sudo firewall-cmd --permanent --add-port=53/udp
sudo firewall-cmd --permanent --add-port=67-69/udp # DHCP/TFTP (si activé)

sudo firewall-cmd --reload
```

Installation des dépôts Foreman

```
# Installer le dépôt Foreman
sudo dnf -y install https://yum.theforeman.org/releases/3.17/el9/x86_64/foreman-release.rpm

# Installer le dépôt Puppet
sudo dnf -y install https://yum.puppet.com/puppet8-release-el-9.noarch.rpm
```

```
# Nettoyer le cache
sudo dnf clean all
```

Installation de Foreman avec l'installateur

```
# Installer le paquet foreman-installer
sudo dnf install -y foreman-installer

# Lancer l'installation avec les options de base
sudo foreman-installer \
  --foreman-initial-admin-username admin \
  --foreman-initial-admin-password "MotDePasseSecurise" \
  --enable-foreman-proxy \
  --enable-puppet \
  --puppet-server true \
  --puppet-server-foreman-url https://foreman.mondomaine.com
```

> **Important** : Notez bien le mot de passe admin et les informations affichées en fin d'installation.

Vérification de l'installation

```
# Vérifier que les services sont actifs
sudo systemctl status foreman
sudo systemctl status httpd
sudo systemctl status puppetserver
sudo systemctl status foreman-proxy

# Tester l'accès en local de la WebUI
curl -k https://foreman.mondomaine.com

# Tester l'accès en local de la page de status
curl -k https://foreman.mondomaine.com/status

# Tester l'accès en local du Smart Proxy
curl -k https://foreman.mondomaine.com:8443/version
```

```
### Doit  retourner :
{"version":"<version>","modules":{"puppetca":"<version>","puppet":"<version>","logs":"<version
>"}}

# Vérifier que le module puppet fonctionne
curl --cert /etc/puppetlabs/puppet/ssl/certs/foreman.mondomaine.com.pem \
     --key /etc/puppetlabs/puppet/ssl/private_keys/foreman.mondomaine.com.pem \
     --cacert /etc/puppetlabs/puppet/ssl/certs/ca.pem \
     https://foreman.mondomaine.com:8443/puppet/environments
### Doit  retourner :  ["production","common"]

# Vérifier que le module logs fonctionne
curl --cert /etc/puppetlabs/puppet/ssl/certs/foreman.mondomaine.com.pem \
     --key /etc/puppetlabs/puppet/ssl/private_keys/foreman.mondomaine.com.pem \
     --cacert /etc/puppetlabs/puppet/ssl/certs/ca.pem \
     https://foreman.mondomaine.com:8443/puppet/environments
### Doit  retourner des logs parsés en json
```

Configuration HAProxy

***IMPORTANT** : Cette section concerne le **serveur HAProxy mutualisé séparé**, pas le serveur Foreman.*

1 Installation de HAProxy

```
# Sur le serveur HAProxy (serveur séparé du serveur Foreman)
sudo dnf install -y haproxy
```

2 Génération des certificats SSL

Si vous n'avez pas encore de certificats SSL, vous pouvez utiliser Let's Encrypt ou générer des certificats auto-signés pour les tests :

```
# Certificats auto-signés (UNIQUEMENT POUR LES TESTS)
sudo mkdir -p /etc/haproxy/certs

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/haproxy/certs/foreman.key \
  -out /etc/haproxy/certs/foreman.crt \
  -subj "/C=FR/ST=IDF/L=Paris/O=MonOrganisation/CN=foreman.mondomaine.com"

# Combiner la clé et le certificat pour HAProxy
sudo cat /etc/haproxy/certs/foreman.crt /etc/haproxy/certs/foreman.key >
/etc/haproxy/certs/foreman.pem
sudo chmod 600 /etc/haproxy/certs/foreman.pem
```

3 Configuration HAProxy

Sauvegarder la configuration par défaut :

```
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.backup
```

Créer la nouvelle configuration :

```
sudo vim /etc/haproxy/haproxy.cfg
```

Contenu de `/etc/haproxy/haproxy.cfg` :

```
#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
global
    log         127.0.0.1 local2
    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

    # SSL/TLS configuration
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-
ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

#---------------------------------------------------------------------
# Defaults settings
#---------------------------------------------------------------------
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option                  http-server-close
    option                  forwardfor except 127.0.0.0/8
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
```

```
    maxconn                 3000

#---------------------------------------------------------------------
# Frontend HTTP - Redirection vers HTTPS
#---------------------------------------------------------------------
frontend http_front
    bind *:80
    mode http

    # Redirection de tout le trafic HTTP vers HTTPS
    redirect scheme https code 301 if !{ ssl_fc }

#---------------------------------------------------------------------
# Frontend HTTPS - Interface Web Foreman (port 443)
#---------------------------------------------------------------------
frontend https_front
    bind *:443 ssl crt /etc/haproxy/certs/foreman.pem
    mode http

    # Headers de sécurité
    http-response set-header Strict-Transport-Security "max-age=31536000; includeSubDomains"
    http-response set-header X-Frame-Options "SAMEORIGIN"
    http-response set-header X-Content-Type-Options "nosniff"

    # ACL pour la WebUI
    acl is_foreman_ui hdr(host) -i foreman.mondomaine.com

    # Headers pour le backend
    http-request set-header X-Forwarded-Proto https
    http-request set-header X-Forwarded-Port 443
    http-request add-header X-Forwarded-Ssl on

    # Route vers le backend Foreman
    use_backend foreman_webui if is_foreman_ui
    default_backend foreman_webui

#---------------------------------------------------------------------
# Frontend HTTPS - Smart Proxy (port 8443)
#---------------------------------------------------------------------
```

```
frontend smartproxy_front
    bind *:8443 ssl crt /etc/haproxy/certs/foreman.pem
    mode http

    # Headers de sécurité
    http-response set-header Strict-Transport-Security "max-age=31536000; includeSubDomains"

    # ACL pour Smart Proxy
    acl is_smartproxy hdr(host) -i foreman.mondomaine.com

    # Headers pour le backend
    http-request set-header X-Forwarded-Proto https
    http-request set-header X-Forwarded-Port 8443
    http-request add-header X-Forwarded-Ssl on

    # Route vers le backend Smart Proxy
    use_backend foreman_smartproxy if is_smartproxy
    default_backend foreman_smartproxy

#---------------------------------------------------------------------
# Backend - Foreman WebUI
#---------------------------------------------------------------------
backend foreman_webui
    mode http
    balance roundrobin
    option httpchk GET /

    # Configuration du backend
    http-request set-header X-Forwarded-Host %[req.hdr(Host)]

    # Serveur Foreman backend
    # REMPLACER <IP_SERVEUR_FOREMAN> par l'IP réelle de votre serveur Foreman
    server foreman1 <IP_SERVEUR_FOREMAN>:80 check inter 5s rise 2 fall 3

    # Exemple avec une IP : server foreman1 192.168.1.100:80 check inter 5s rise 2 fall 3

#---------------------------------------------------------------------
# Backend - Smart Proxy
#---------------------------------------------------------------------
```

```
backend foreman_smartproxy
    mode http
    balance roundrobin
    option httpchk GET /

    # Configuration du backend
    http-request set-header X-Forwarded-Host %[req.hdr(Host)]

    # Serveur Smart Proxy backend
    # REMPLACER <IP_SERVEUR_FOREMAN> par l'IP réelle de votre serveur Foreman
    server smartproxy1 <IP_SERVEUR_FOREMAN>:8443 check ssl verify none inter 5s rise 2 fall 3

    # Exemple avec une IP : server smartproxy1 192.168.1.100:8443 check ssl verify none inter
5s rise 2 fall 3

#---------------------------------------------------------------------
# Stats page (optionnel)
#---------------------------------------------------------------------
listen stats
    bind *:9000
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth admin:VotreMotDePasse
    stats refresh 30s
```

4 Démarrage et activation de HAProxy

```
# Vérifier la configuration
sudo haproxy -c -f /etc/haproxy/haproxy.cfg

# Si pas d'erreur, activer et démarrer HAProxy
sudo systemctl enable haproxy
sudo systemctl start haproxy

# Vérifier le statut
sudo systemctl status haproxy
```

```
# Configurer le firewall pour HAProxy
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --permanent --add-port=443/tcp
sudo firewall-cmd --permanent --add-port=8443/tcp
sudo firewall-cmd --permanent --add-port=8140/tcp  # Si option 1 pour Puppet
sudo firewall-cmd --permanent --add-port=9000/tcp  # Stats (optionnel)
sudo firewall-cmd --reload
```

Test de connectivité HAProxy → Foreman

Avant de continuer, vérifier que HAProxy peut bien communiquer avec le serveur Foreman :

```
# Depuis le serveur HAProxy, tester la connectivité
ping <IP_SERVEUR_FOREMAN>

# Tester les ports
nc -zv <IP_SERVEUR_FOREMAN> 80
nc -zv <IP_SERVEUR_FOREMAN> 8443
nc -zv <IP_SERVEUR_FOREMAN> 8140

# Tester l'accès HTTP
curl -v http://<IP_SERVEUR_FOREMAN>/
curl -kv https://<IP_SERVEUR_FOREMAN>:8443/

# Si tout fonctionne, tester via HAProxy depuis HAProxy lui-même
curl -kv https://localhost/
curl -kv https://localhost:8443/
```

5 Configuration Foreman pour fonctionner derrière HAProxy

Sur le **serveur Foreman**, modifier la configuration pour accepter les requêtes via proxy :

```
sudo vim /etc/foreman/settings.yaml
```

Ajouter/modifier :

```
:trusted_proxies:
  - <IP_SERVEUR_HAPROXY>
  # Remplacer <IP_SERVEUR_HAPROXY> par l'IP réelle de votre serveur HAProxy
  # Exemple : - 192.168.1.50
```

```
:require_ssl: true
```

IMPORTANT : Vous devez impérativement ajouter l'IP du serveur HAProxy dans la liste des proxies de confiance, sinon Foreman rejettera les requêtes.

Redémarrer les services Foreman :

```
sudo systemctl restart foreman
sudo systemctl restart httpd
```

6 Configuration Apache sur le serveur Foreman

Foreman doit être configuré pour accepter les connexions depuis HAProxy et gérer correctement les headers X-Forwarded :

```
# Éditer la configuration SSL de Foreman
sudo vim /etc/httpd/conf.d/05-foreman-ssl.conf
```

Ajouter dans le VirtualHost (avant la ligne `</VirtualHost>`) :

```
  # Configuration pour reverse proxy
  RemoteIPHeader X-Forwarded-For
  RemoteIPInternalProxy <IP_SERVEUR_HAPROXY>

  # Activer les modules nécessaires
  <IfModule mod_remoteip.c>
    RemoteIPHeader X-Forwarded-For
    RemoteIPTrustedProxy <IP_SERVEUR_HAPROXY>
  </IfModule>
```

Vérifier que le module remoteip est activé :

```
sudo dnf install -y mod_remoteip
sudo systemctl restart httpd
```

---

7 Configuration réseau et DNS

Puisque HAProxy et Foreman sont sur des serveurs séparés, vous devez configurer correctement le DNS et le routage.

Sur le serveur DNS (ou /etc/hosts des clients)

Le nom `foreman.mondomaine.com` doit pointer vers l'IP du serveur **HAProxy**, pas du serveur Foreman :

```
# DNS ou /etc/hosts
<IP_HAPROXY>  foreman.mondomaine.com
```

Sur le serveur HAProxy

Ajouter l'entrée pour le serveur Foreman :

```
sudo vim /etc/hosts
```

```
<IP_SERVEUR_FOREMAN>  foreman-backend.mondomaine.com foreman-backend
```

Sur le serveur Foreman

Le serveur Foreman doit pouvoir être contacté par HAProxy. Vérifier la configuration réseau :

```
# Vérifier la connectivité depuis HAProxy
ping <IP_SERVEUR_FOREMAN>

# Vérifier que les ports sont ouverts depuis HAProxy
nc -zv <IP_SERVEUR_FOREMAN> 80
nc -zv <IP_SERVEUR_FOREMAN> 8443
```

Configuration du hostname Foreman

Sur le serveur Foreman, le hostname doit correspondre au nom public :

```
sudo hostnamectl set-hostname foreman.mondomaine.com
```

Le fichier `/etc/hosts` sur le serveur Foreman :

```
127.0.0.1   localhost localhost.localdomain
<IP_LOCALE_FOREMAN>   foreman.mondomaine.com foreman
```

Installation de l'agent Puppet sur les nœuds clients

1 Installation sur EL9

```
# Installer le dépôt Puppet 7
sudo dnf install -y https://yum.puppet.com/puppet7-release-el-9.noarch.rpm

# Installer l'agent Puppet
sudo dnf install -y puppet-agent

# Ajouter Puppet au PATH
echo 'export PATH=/opt/puppetlabs/bin:$PATH' >> ~/.bashrc
source ~/.bashrc
```

2 Configuration de l'agent Puppet

```
# Éditer la configuration Puppet
sudo vim /etc/puppetlabs/puppet/puppet.conf
```

Ajouter/modifier dans la section `[main]` :

```
[main]
certname = client.mondomaine.com
server = foreman.mondomaine.com
environment = production
runinterval = 30m
```

3 Configuration du fichier hosts sur le client

```
sudo vim /etc/hosts
```

Ajouter (pointer vers le serveur **HAProxy**, pas Foreman) :

```
<IP_HAPROXY>  foreman.mondomaine.com foreman
```

Important : Les clients doivent pointer vers HAProxy, qui fait office de reverse proxy SSL/TLS.

4 Premier run

```
# Premier run pour générer le certificat
sudo /opt/puppetlabs/bin/puppet agent --test

# Sur le serveur Foreman, lister les certificats en attente
sudo puppetserver ca list

# Signer le certificat du client
sudo puppetserver ca sign --certname client.mondomaine.com

# Ou signer tous les certificats en attente
sudo puppetserver ca sign --all

# Relancer l'agent sur le client
sudo /opt/puppetlabs/bin/puppet agent --test
```

5 Activation du service Puppet

```
# Activer et démarrer le service
sudo systemctl enable puppet
sudo systemctl start puppet

# Vérifier le statut
sudo systemctl status puppet
```

6 Sur un client Debian/Ubuntu

```
# Télécharger et installer le dépôt Puppet
wget https://apt.puppet.com/puppet7-release-$(lsb_release -cs).deb
sudo dpkg -i puppet7-release-$(lsb_release -cs).deb
sudo apt update

# Installer l'agent
sudo apt install -y puppet-agent

# Suivre les mêmes étapes de configuration que pour Rocky Linux
```

7 Sur un client Windows

1. Télécharger l'installateur depuis : https://downloads.puppetlabs.com/windows/puppet7/
2. Exécuter l'installateur

3. Pendant l'installation, spécifier `foreman.mondomaine.com` comme serveur Puppet
4. Après installation, ouvrir PowerShell en administrateur :

```
# Premier run
puppet agent --test

# Activer le service
Set-Service -Name puppet -StartupType Automatic
Start-Service -Name puppet
```

Vérifications et dépannage

1 Vérifications et tests

Accès à l'interface Web

Ouvrir un navigateur et aller sur :

- **WebUI** : https://foreman.mondomaine.com
- **Smart Proxy** : https://foreman.mondomaine.com:8443
- **Stats HAProxy** : http://<IP_HAPROXY>:9000/stats

Connexion :

- Utilisateur : `admin`
- Mot de passe : celui défini lors de l'installation

Vérification des hosts dans Foreman

Dans l'interface Foreman :

1. Aller dans **Hosts** > **All Hosts**
2. Vérifier que vos clients apparaissent
3. Vérifier le statut Puppet

2 Emplacement des logs

Sur le serveur Foreman :

```
# Logs Foreman
sudo tail -f /var/log/foreman/production.log

# Logs Puppet Server
sudo tail -f /var/log/puppetlabs/puppetserver/puppetserver.log

# Logs Apache
sudo tail -f /var/log/httpd/error_log
sudo tail -f /var/log/httpd/foreman-ssl_error_log
```

3 Dépannage

Problème de connexion à l'interface Web

```
# Vérifier que les services sont actifs
sudo systemctl status foreman httpd haproxy

# Vérifier les ports en écoute
sudo netstat -tlnp | grep -E '(80|443|8443)'

# Tester la connectivité
curl -k https://localhost/
curl -k https://foreman.mondomaine.com/
```

Problème de certificat Puppet

```
# Sur le serveur, lister tous les certificats
sudo puppetserver ca list --all

# Nettoyer un certificat
sudo puppetserver ca clean --certname client.mondomaine.com

# Sur le client, supprimer les certificats
sudo rm -rf /etc/puppetlabs/puppet/ssl/

# Refaire la demande de certificat
sudo /opt/puppetlabs/bin/puppet agent --test
```

HAProxy ne démarre pas

```
# Vérifier la syntaxe de la configuration
sudo haproxy -c -f /etc/haproxy/haproxy.cfg

# Vérifier les logs
sudo journalctl -u haproxy -n 50

# Vérifier SELinux
sudo ausearch -m avc -ts recent
```

4 Commandes utiles

Commandes Foreman

```
# Redémarrer tous les services Foreman
sudo systemctl restart foreman httpd puppetserver foreman-proxy
```

```
# Vérifier la version
foreman-rake -v

# Console Rails
sudo foreman-rake console
```

Commandes Puppet

```
# Version de Puppet
puppet --version

# Tester l'agent
puppet agent --test --noop  # Mode dry-run

# Voir les facts d'un nœud
facter
```

Commandes HAProxy

```
# Recharger la configuration sans interruption
sudo systemctl reload haproxy

# Statistiques en temps réel
echo "show stat" | sudo socat stdio /var/lib/haproxy/stats
```

Configurations avancées

1 Activation de la gestion DNS

Si vous souhaitez que Foreman gère le DNS :

```
sudo foreman-installer \
  --foreman-proxy-dns true \
  --foreman-proxy-dns-managed true \
  --foreman-proxy-dns-provider nsupdate \
  --foreman-proxy-dns-server "127.0.0.1" \
  --foreman-proxy-dns-zone "mondomaine.com"
```

2 Activation de la gestion DHCP

```
sudo foreman-installer \
  --foreman-proxy-dhcp true \
  --foreman-proxy-dhcp-managed true \
  --foreman-proxy-dhcp-interface "eth0" \
  --foreman-proxy-dhcp-gateway "192.168.1.1" \
  --foreman-proxy-dhcp-range "192.168.1.100 192.168.1.200" \
  --foreman-proxy-dhcp-nameservers "8.8.8.8,8.8.4.4"
```

3 Configuration SSL avec Let's Encrypt

```
# Installer certbot
sudo dnf install -y certbot

# Obtenir un certificat
sudo certbot certonly --standalone -d foreman.mondomaine.com

# Les certificats seront dans /etc/letsencrypt/live/foreman.mondomaine.com/

# Créer le fichier combiné pour HAProxy
sudo cat /etc/letsencrypt/live/foreman.mondomaine.com/fullchain.pem \
     /etc/letsencrypt/live/foreman.mondomaine.com/privkey.pem \
     > /etc/haproxy/certs/foreman.pem

sudo chmod 600 /etc/haproxy/certs/foreman.pem
```

```
# Redémarrer HAProxy
sudo systemctl restart haproxy
```

Sauvegarde et maintenance

1 Sauvegarde de Foreman

```
# Sauvegarder la base de données
sudo foreman-rake db:dump

# Sauvegarder la configuration
sudo tar -czf /root/foreman-backup-$(date +%Y%m%d).tar.gz \
  /etc/foreman \
  /etc/puppetlabs \
  /etc/haproxy \
  /var/lib/puppet \
  /usr/share/foreman
```

2 Mise à jour de Foreman

```
# Mettre à jour les paquets
sudo dnf update -y

# Relancer l'installateur
sudo foreman-installer

# Redémarrer les services
sudo systemctl restart foreman httpd puppetserver foreman-proxy
```

Architecture de Foreman

Vue d'ensemble

Foreman est une plateforme complète de gestion du cycle de vie des serveurs. C'est bien plus qu'un simple outil de gestion de configuration : c'est une solution intégrée qui combine plusieurs technologies pour offrir un système complet d'orchestration et de provisioning.

Qu'est-ce que Foreman ?

Foreman permet de :

- **Provisionner** des serveurs physiques et virtuels (bare-metal et cloud)
- **Configurer** les systèmes via Puppet, Ansible, Salt ou Chef
- **Gérer** le cycle de vie complet des machines (de l'installation à la destruction)
- **Monitorer** l'état de configuration et les rapports
- **Automatiser** les tâches répétitives d'infrastructure

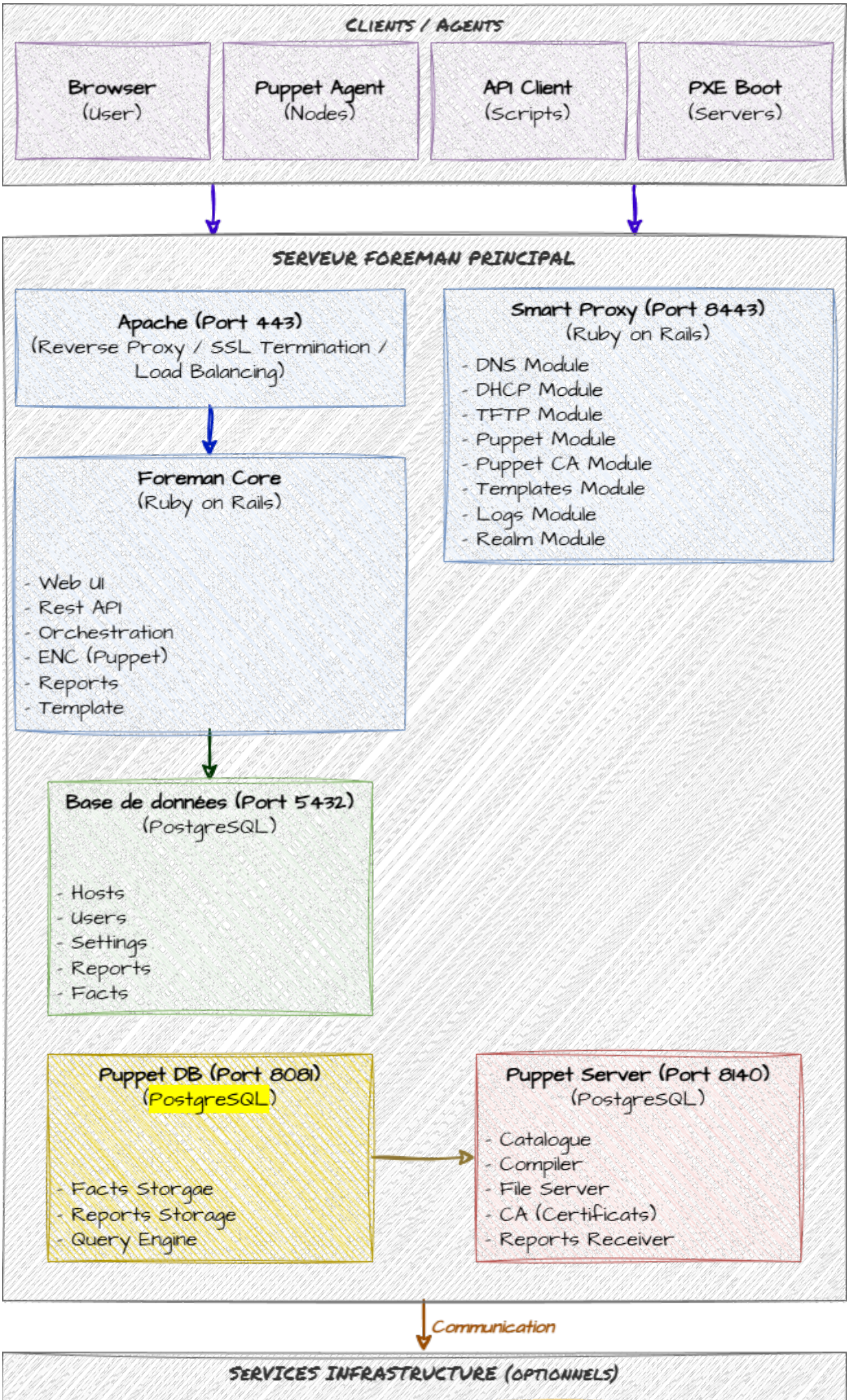
Philosophie d'architecture

Foreman suit une architecture modulaire où chaque composant a un rôle spécifique :

- **Foreman Core** : Interface web et logique métier
- **Smart Proxy** : Agent de communication avec l'infrastructure
- **Puppet Server** : Gestion de configuration
- **Base de données** : Stockage des données
- **Services d'infrastructure** : DNS, DHCP, TFTP

Architecture détaillée

Schéma d'architecture complète


CLIENTS / AGENTS
Browser
(User)
Puppet Agent
(Nodes)
API Client
(Scripts)
PXE Boot
(Servers)
SERVEUR FOREMAN PRINCIPAL
Apache (Port 443)
(Reverse Proxy / SSL Termination / Load Balancing)
Smart Proxy (Port 8443)
(Ruby on Rails)
- DNS Module
- DHCP Module
- TFTP Module
- Puppet Module
- Puppet CA Module
- Templates Module
- Logs Module
- Realm Module
Foreman Core
(Ruby on Rails)
- Web UI
- Rest API
- Orchestration
- ENC (Puppet)
- Reports
- Template
Base de données (Port 5432)
(PostgreSQL)
- Hosts
- Users
- Settings
- Reports
- Facts
Puppet DB (Port 8081)
(PostgreSQL)
- Facts Storgae
- Reports Storage
- Query Engine
Puppet Server (Port 8140)
(PostgreSQL)
- Catalogue
- Compiler
- File Server
- CA (Certificats)
- Reports Receiver
Communication
SERVICES INFRASTRUCTURE (OPTIONNELS)

Les principaux composants

1. Foreman Core (Application Rails)

Rôle : Cerveau du système, interface web centrale

Caractéristiques :

- Application Ruby on Rails
- Interface web responsive
- API REST complète
- Gestion des utilisateurs et permissions (RBAC)
- Orchestration des autres composants

Fonctionnalités :

- Gestion des hôtes (inventaire)
- Templates de provisioning
- Gestion des paramètres et variables
- Rapports et monitoring
- Gestion des rôles et permissions
- Intégration avec les fournisseurs cloud (AWS, Azure, GCP, etc.)

Processus :

```
Service : foreman.service (Puma/Passenger)
Port : Aucun (accès via Apache/Nginx)
Utilisateur : foreman
```

2. Apache/Nginx (Serveur Web)

Rôle : Serveur web frontal pour Foreman

Caractéristiques :

- Proxy inverse pour l'application Foreman
- Terminaison SSL/TLS
- Authentification (optionnel)
- Compression et cache

Configuration par défaut :

```
Port HTTP : 80 (redirigé vers HTTPS)
Port HTTPS : 443
Serveur backend : Puma (socket Unix)
```

VirtualHosts principaux :

- `/` : Application Foreman
- `/pulp` : Gestion de contenu (si activé)
- `/pub` : Fichiers publics (kickstart, preseed, etc.)

3. Smart Proxy (foreman-proxy)

Rôle : Agent d'infrastructure, bras armé de Foreman

Le Smart Proxy est le composant qui permet à Foreman d'interagir avec l'infrastructure réseau et les services. Il peut être installé sur le même serveur que Foreman ou sur des serveurs distants.

Fonctionnalités modulaires :

DNS

- Gestion des enregistrements DNS
- Création/suppression automatique d'entrées A, PTR, AAAA
- Support de BIND, PowerDNS, Route53, etc.
- Mode API ou nsupdate

DHCP

- Attribution d'adresses IP
- Création de réservations
- Support ISC DHCP, MS DHCP, Infoblox
- Gestion des options DHCP (next-server, filename, etc.)

TFTP

- Distribution des fichiers de boot PXE
- Templates de boot (pxelinux, grub2, iPXE)
- Gestion automatique des fichiers

Puppet CA

- Gestion des certificats Puppet
- Signature automatique ou manuelle
- Révocation de certificats

- Liste des certificats en attente

Puppet

- Déclenchement de runs Puppet
- Import de classes et environnements
- Récupération de facts
- Gestion des environnements

Templates

- Proxy pour les templates de provisioning
- Distribution des kickstart/preseed/cloud-init
- Génération à la volée

Logs

- Récupération de logs systèmes
- Agrégation des logs de boot
- Debugging du provisioning

Realm (Kerberos/AD)

- Intégration Active Directory
- Intégration FreeIPA
- Jointure automatique au domaine

Configuration :

```
Port : 8443 (HTTPS)
Protocole : REST API
Authentification : Certificat SSL client
Configuration : /etc/foreman-proxy/settings.yml
Modules : /etc/foreman-proxy/settings.d/*.yml
```

4. Puppet Server

Rôle : Serveur de gestion de configuration

Le Puppet Server est le composant central pour la gestion de configuration. Il stocke les manifests, modules et données, et les distribue aux agents.

Composants Puppet :

Puppet Server (JVM)

- Compile les catalogues
- Sert les fichiers et modules
- Gère les certificats (CA)
- Collecte les rapports

PuppetDB (optionnel mais recommandé)

- Base de données des facts
- Historique des catalogues
- Stockage des rapports
- Requêtes PQL (Puppet Query Language)

Puppet CA

- Autorité de certification
- Génération des certificats clients/serveurs
- Gestion du CRL (Certificate Revocation List)

Ports et services :

```
Puppet Server : 8140 (HTTPS)
PuppetDB : 8081 (HTTP/HTTPS)
Catalogue Compiler : Intégré dans 8140
```

Workflow Puppet :

```
1. Agent demande un certificat → Puppet CA
2. Agent envoie ses facts → Puppet Server
3. Server compile le catalogue → Renvoie à l'agent
4. Agent applique le catalogue
5. Agent renvoie le rapport → Puppet Server → PuppetDB
6. Foreman récupère le rapport → Affichage dans l'UI
```

5. Base de données

Rôle : Stockage persistant des données Foreman

Base de données supportées :

- **PostgreSQL** (recommandé en production)
- **MySQL/MariaDB** (supporté)
- **SQLite** (développement uniquement)

Données stockées :

- Inventaire des hôtes
- Configuration des hôtes
- Utilisateurs et permissions
- Paramètres et variables
- Templates
- Rapports Puppet (si PuppetDB non utilisé)
- Logs et audits

Configuration :

```
Base de données : foreman
Utilisateur : foreman
Port : 5432 (PostgreSQL) / 3306 (MySQL)
```

6. Services d'infrastructure (optionnels)

DNS (BIND, PowerDNS, etc.)

- Résolution de noms
- Zones directes et inverses
- Mises à jour dynamiques (DDNS)

DHCP (ISC DHCP, etc.)

- Attribution d'adresses IP
- Options PXE boot
- Réservations

TFTP

- Boot PXE
- Distribution des images de boot
- Chainloading (pxelinux, grub, iPXE)

Conclusion

L'architecture de Foreman est modulaire et extensible :

- **Foreman Core** : Orchestrateur central et interface utilisateur
- **Smart Proxy** : Extension de Foreman vers l'infrastructure
- **Puppet Server** : Moteur de configuration
- **Services d'infrastructure** : DNS, DHCP, TFTP pour le provisioning

Les agents Puppet se connectent via mTLS sur le port 8140, récupèrent leur configuration depuis Foreman (via ENC), et appliquent les changements de manière déclarative.

L'ajout de HAProxy permet de centraliser les accès, gérer le SSL/TLS, et protéger le serveur Foreman des accès directs.

Cette architecture offre :

- **Scalabilité** : Smart Proxies distribués
- **Sécurité** : mTLS, RBAC, certificats
- **Flexibilité** : Plugins, API, intégrations multiples
- **Automatisation** : Provisioning complet de l'installation à la configuration