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

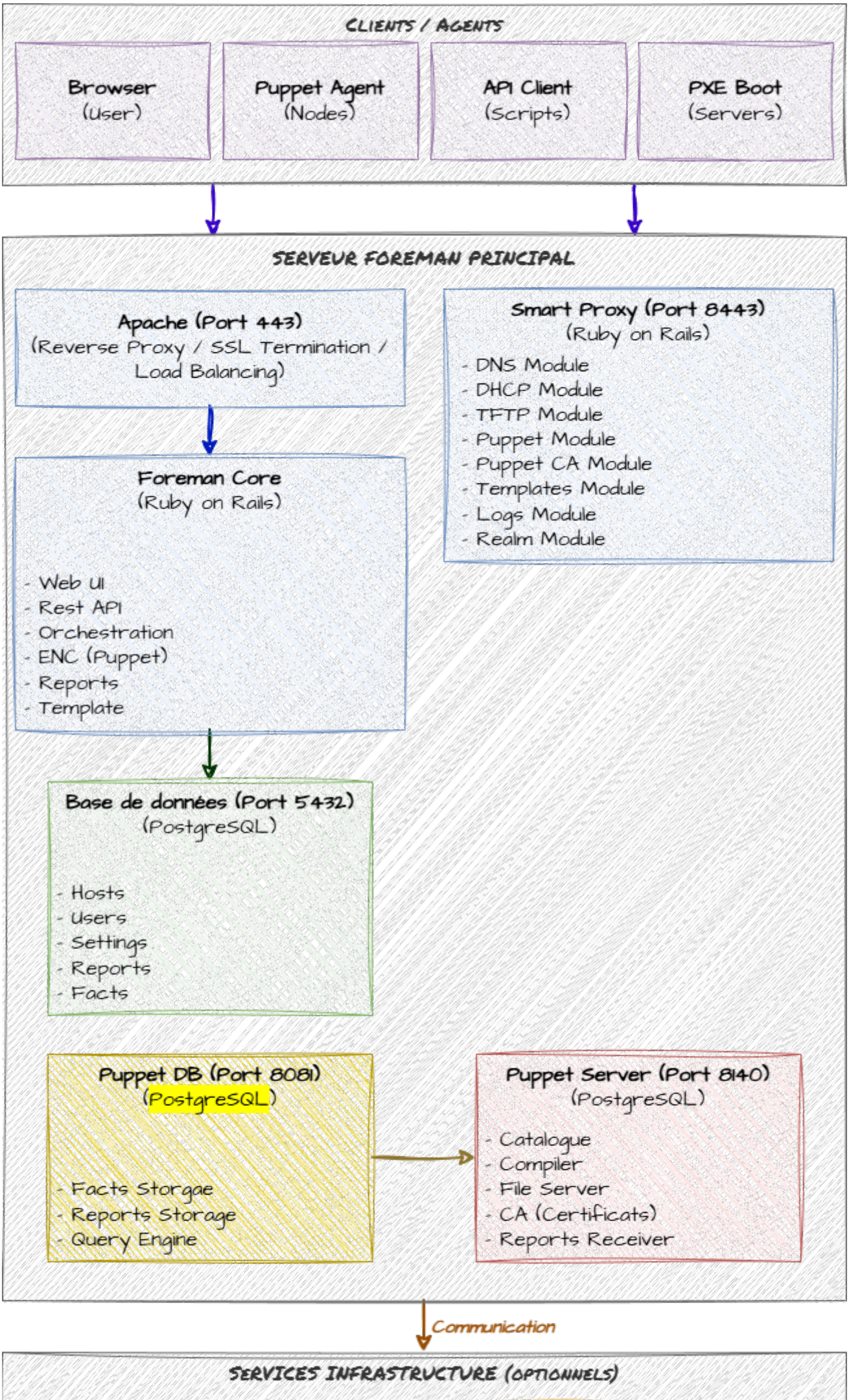
Philosophie d'architecture

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

---

Revision #5
Created 2026-02-02 12:02:36 UTC by NoNo
Updated 2026-02-02 15:00:16 UTC by NoNo