

# Puppet après 10 ans

ALBERT SHIH

Observatoire de Paris/Direction information de l'Observatoire

13/11/2023

# Puppet

- Écrit en ruby/Jruby.
- Client — Serveur.
- Client en mode daemon, cron, rsync + run local, etc.
- Utilisation PKI.
- Facter/Provider sur le client.
- Ressources exportés.
- Environnement de développement (par utilisateur).

# Puppet

- Langage spécifique.
- Utilisation de modules (forge, github, etc.).
- Développement modules maison/Facter/Provider.

# Situation 2013

- Situation 2006—2010 :
  - Tout fait à main.
  - Cauchemard.
  - Travail monstre.
- Besoin d'un *outils* vers 2010.
- Test vers 2011-2012.
- Début du passage 2013.

# Début

- Installation packages par défaut.
- Configuration de `/etc/resolv.conf`.
- Creation comptes admin.
- Configuration ssh.
- etc.

# Maintenant

- $\geq$  20000 commits.
- $\approx$  70 modules maison.
- 507 serveurs (linux+FreeBsd)
- Gestion des pkg/configuration de bases.
- Gestion du supervision.
- Gestion du sauvegarde.
- Développement du premier provider maison.

# Maintenant

- module `serveur_hp` charge/configure
  - `serveur_physique`.
  - Installation `amsd`.
  - Configuration supervision hardware via `redfish`.
- module `serveur_physique` charge/configure.
  - `ntp`, `lldp`.
  - Supervision des disques.
  - Sauvegarde.
  - Supervision garantie.
  - Supervision `bmc`.

# Mise à jours

- Pénible puppet 3  $\rightarrow$  puppet 4.
- $\geq$  Puppet 4 sans soucis.
- Migration PKI : un peu chronophage.
- Passage “un jour” à puppet 8.
- Mise à jour des modules : Presque jamais eu de problème.
- Module pour mise à jours des agents (linux).



# Bilan

- Homogénéité.
- *Pas* d'oublis.
- Historique.
- Unicité de l'informations.
- Facilité de modification globale (par exemple serveurs LDAP).
- Ré-utilisation modules maison.
- Chargement profiles.
- etc.

# Difficultés

- Pas développeur.
- Historique à *trainer*, évolutions puppet (profiles/rôles).
- Ré-utilisation du code.
- Impossible de tout mettre yaml ou classes.
- Compliqué fichiers data dynamiques.
- Documentations *difficiles* à maintenir.
- Applications/configurations *automatiques* (curl, bash).
- Applications/configurations *modernes* (conteneurs).
- Équilibre temps développement modules/gain pas facile à gérer.
- Lenteurs avec les ressources exportés.

# Futur

- Puppet Co racheté par Salesforce.
- Pas de paquets pour Debian 12.
- Banalisation des conteneurs.