MEETUP ANSIBLE MONTRÉAL

ADOPTION D'ANSIBLE AU SEIN D'UNE ÉQUIPE DE SYSADMINS

mercredi 12 avril 2017, par Grégory Colpart

evolix

Hébergement et Infogérance en Logiciels Libres

- Infogérance / Hébergement / Infonuagique
- Managed Hosting Provider
- Linux, infra web, HA, Ansible, Docker
- Une équipe à Montréal + en France (24/7)
- Clients : agences web, SaaS, médias

MÉTIER D'EVOLIX

- Une centaine d'infra clients
- Un à plusieurs dizaines de serveurs par infra
- Total d'environ 700 serveurs hétérogènes infogérés
- Serveurs Debian/BSD
- Culture SysAdmin : shell, 100% outils libres

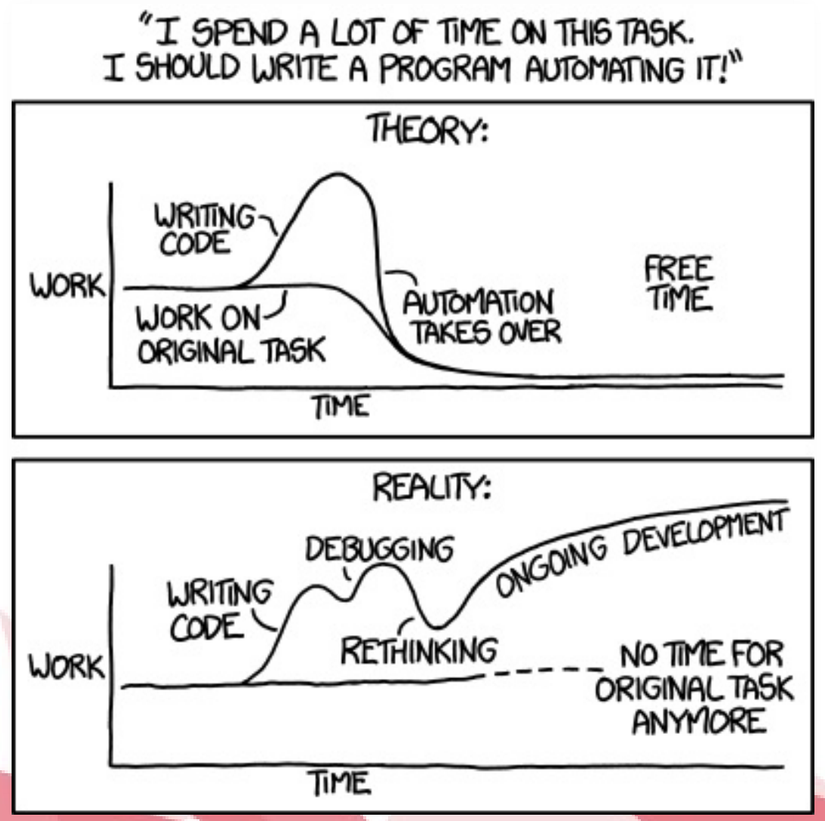
HISTORIQUE : COMMENT INSTALLER UN SERVEUR ?

- tout à la main
- copier/coller
- checks (evocheck)
- script shell privé (evolinux v1)


Are you too busy to improve?
No thanks!
We are too busy
Håkan Forss @hakanforss http://hakanforss.wordpress.com
This illustration is inspired by and in part derived from the work by Scott Simmerman, "The Square Wheels Guy" http://www.performancemanagementcompany.com/

AUTOMATION ?

(Temps d'une tâche) × (nombre d'exécution) VS
Temps d'automatisation



2013 : GESTION D'UNE INFRA CLIENT AVEC PUPPET

2014 : PLAYBOOK ANSIBLE POUR DES PETITES TÂCHES PONCTUELLES

ajout d'une nouvelle adresse IP sur tous nos serveurs

DÉCLIC IMMÉDIAT...

- adoption progressive par certains SysAdmins
- facile à appréhender (module command/shell)
- homogénéité, fiabilité, rapidité

ON L'ADOPTE POUR :

- actions urgentes (màj sécurité comme *openssl)*
- tâches répétitives (utilisateurs, firewall)
- demandes spécifiques (extension cluster, instances de services)

2015 : ADOPTION OFFICIELLE D'ANSIBLE

- inventory généré à partir de notre annuaire LDAP
- migration d'Ansible 1.7 vers 2.0 (*sudo* → *become*)

2016 : CRÉATIONS DE RÔLES ANSIBLE DANS UN DÉPÔT DÉDIÉ/PUBLIC

conversion de notre script shell d'installation en ensemble de playbooks/rôles Ansible

2016 : INSTALLATION D'INFRA CLIENT COMPLEXES COMPLÈTEMENT ORCHESTRÉES AVEC ANSIBLE

2017 : MIGRATION D'ANSIBLE 2.0 VERS VERSION 2.2

LES KILLERS FEATURES D'ANSIBLE

- concept idempotence
- agent-less permettant une adoption progressive

CHOIX D'ORGANISATION

- rôles Ansible publics (déploiement en cours vers Ansible Galaxy)
- outillage Ansible public : tasks, conventions, etc.
- outillage Ansible privé : vault pour variables, tasks, etc.
- scripts de gestion : génération des variables (dialog), sync Git, etc.
- playbooks dans un dépôt partagé avec le client

CONVENTIONS

- la structure des rôles (ansible galaxy, README.md)
- le format YAML (pas de syntaxe compacte)
- l'utilisation de la précédence des variables
- utilisation du check mode
- dépôts publics : commentaires et docs en anglais

```
mailto = {{ log2mail_alert_email or general_alert_email }}
```

CHOIX DE NE PAS UTILISER ANSIBLE TOWER

- 100% libre
- pas d'interface graphique

FOCUS SUR L'INVENTORY

- inventory via LDAP pour la prod
- inventory par client, notamment pour les tests

FOCUS SUR LES MODULES

Il existe des centaines de modules...

CHOIX FIXE DE CERTAINS MODULES + CONVENTIONS D'UTILISATION

lineinfile vs blockinfile vs copy/template

CONVENTIONS (SUITE)

- utilisation des tags
- secrets : *vault* dans un repository privé
- tests pour vérification de l'application des conventions
- Serveurs centraux... mais optionnels

```
[defaults]
inventory = $HOME/.ansible/hosts
gathering = smart

[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=300s
pipelining = True
```

```
[defaults]
inventory = $HOME/.ansible/hosts
[ssh_connection]
ssh_args = -o ControlMaster=no -o ControlPersist=no
```

ENJEU DE TRANSMISSION DU SAVOIR

- tout le monde doit être dans la barque
- les découvertes doivent être transmises
- les conventions et bonnes pratiques prêchées et rabachées
- faire du collectif/collaboratif et pas avoir un leader et des suiveurs

PAIR-PROGRAMMING TRÈS EFFICACE

ADOPTION D'ANSIBLE POUR UNE INFRA ULTRA-HÉTÉROGÈNE

- quelques serveurs gérés ad-hoc
- quelques infras globalisées
- migration très progressive des infras legacy
- maintenance complexe dans un contexte de production

FOCUS SUR LA DOCUMENTATION

important pour faciliter la transmission explicite du savoir, des bonnes pratiques, des conventions.

TRAITER LES PLAYBOOKS/ROLES COMME DU CODE :

- des bons messages de commits
- des commentaires là où c'est pertinent (le pourquoi, pas le quoi/comment)
- normer les README
- centraliser les infos clés (conventions.md…)
- écrire des tests

FOCUS SUR LA SÉCURITÉ

Failles de sécurité, comme CVE-2016-9587

TESTS

Indispensables comme pour du code « classique »

- tester la syntaxe
- lint
- tester l'éxécution sans bug, puis tester l'idempotence (via Kitchen)
- tester le résultat (via serverspec lancé par Kitchen)

- Se servir des tests comme chasse aux bugs, mais aussi pour empêcher les régressions
- Le code Ansible est comme du code classique : il évolue, se refactore… et il est facile de casser quelque chose qui a été validé précédemment

QUELQUES DÉFAUTS D'ANSIBLE

- Les messages de sortie en JSON sont conçus pour Ansible Tower, pas pour des humains
- pas de gestion des accréditations, l'idéal serait un agent façon SSH pour mots de passe vault/sudo/ssh
- la documentation est uniquement orientée pour la dernière version

NOTATION EN OCTAL EN YAML

- mode: 0755 → Good!
- mode: 755 → Bad!
- mode: 1777 → Bad!
- mode: "0755" → Good
- mode: "1777" → Good

CONCLUSION

- Ansible, idéal pour une adoption progressive au sein d'une équipe de SysAdmins
- Être attentif au changement de culture que cela apporte
- améliorations : + de tests et push sur Ansible galaxy

POUR EN SAVOIR PLUS...

- Rôles Ansible Evolix : forge.evolix.org/projects/ansible-roles
- Wiki Evolix : wiki.evolix.org
- Twitter : @EvolixCanada
- Mail : hello@evolix.ca