Ansible Cheat sheet

Téléchargé depuis : www.sysadminascode.com/cheat-sheet-ansible

Sysadmin as code

Installation et clés SSH

Installation d'Ansible

```
# Ajout du repo Ansible (sur Débian)
$ sudo apt-add-repository ppa:ansible/ansible
# Update des paquets
$ sudo apt-get update
#Installation d'Ansible
$ sudo apt-get install ansible
# Vérifier la version d'Ansible
$ ansible –version
# Il est possible également d'installer Ansible depuis les sources ou via pip.
```

Partage de clés SSH

```
# Générer la clé RSA de l'utilisateur courant
$ ssh-keygen –t rsa
# Faire le partage de clé rsa vers l'utilisateur john sur le serveur web1.local
$ ssh-copy-id -i ~/.ssh/id_rsa john@web1.local
# Lancer l'agent ssh pour gérer les clés
$ ssh-agent bash
# Charger la clé à utiliser pour les connexions en ssh
$ ssh-add ~/.ssh/id_rsa
```

Pattern

all ou *	Tous les hôtes de l'inventaire
ungrouped	Tous les hôtes qui n'ont pas de groupe
10.0.0.*	Tous les hôtes avec une IP qui commence par 10.0.0.
webservers	Tous les hôtes membre du groupe webservers
webserveurs:!nantes	Les hôtes de webservers non membre du groupe nantes
webserveurs:&nantes	Les hôtes de webservers et du groupe nantes

Inventaire

Les inventaire Ansible listent toutes les plateformes que vous voulez automatiser. Il est possible de manager un seul ou plusieurs hôtes en même temps.

Configuration des hôtes et tests la connexion

```
# Editer le fichier hosts par défaut
$ sudo nano /etc/ansible/hosts
# Tester la connexion à un hôte
$  ansible –m ping <hosts>
# Tester la connexion de tous les serveurs d'un groupe
$ ansible -m ping <nomdugroupe>
# Pour appeler un fichier inventaire autre ajouter l'option suivante en CLI
-i <path>
```

Exemple de fichier d'inventaire

```
ungrouped.example.com          # Un hotes sans groupe

[webservers]                   # Un groue nommé webservers
web1.fr ansible_host = 10.0.0.5  # Un hôtes avec une IP: 10.0.0.5
web2.fr ansible_ssh_user=john  # Un hôtes avec le user de connexion john
web3.fr foo=plop               # foo est une variable d'hôte égale à plop

[clouds]
cloud[0-9].fr                  # Serveurs nommés de cloud0.fr à cloud9.fr

[nantes]
beaujoire.example.com          # Hôte avec résolution DNS
web2.fr                        # Hôte avec résolution DNS

[dev1:children]                # dev1 est un groupe qui contient
webservers                     # tous les hôtes du groupe webservers
clouds                         # et tous les hôtes de groupe clouds
```

Ansible Cheat sheet

Téléchargé depuis : www.sysadminascode.com/cheat-sheet-ansible

Sysadmin as code

Commandes Add-Hoc

Parallélisme et commandes shell

Pour utiliser la connexion avec mot de passe il y a l'option –ask-pass (-K).
Pour limiter l'action des actions à 5 serveurs en parallèle il y a l'option -f
$ ansible france -a "/sbin/reboot" -f 5 –ask-pass
Pour lancer /usr/bin/ansible depuis un autre compte que root
$ ansible france -a "/usr/bin/foo" -u username
Pour lancer une commande avec une élévation de privilège (sudo)
Si le mot de passe utilisateur est nécessaire pour faire sudo, il faut ajouter l'option --ask-become-pass (-K). Le prompt demandera de saisir le mot de passe.
$ ansible france -a "/usr/bin/foo" -u username --become [--ask-become-pass]
Il est possible de sudoer vers un autre compte que root avec --become-user
$ ansible france -a "/usr/bin/foo" -u username --become --become-user otheruser [--ask-become-pass]

Gestion des fichiers

Transférer un fichier vers plusieurs serveur
$ ansible france -m copy -a "src=/etc/hosts dest=/tmp/hosts"
Changer le propriétaire et les droits d'un fichier
$ ansible webservers -m file -a "dest=/srv/foo/a.txt mode=600"
$ ansible webservers -m file -a "dest=/srv/foo/b.txt mode=600 owner=example group=example"
#Créer un répertoire
$ ansible webservers -m file -a "dest=/path/to/c mode=755 owner=example group=example state=directory"
Supprimer un répertoire et les fichiers qu'il contient
$ ansible webservers -m file -a "dest=/path/to/c state=absent"

Déployer depuis un dépôt Git

Repo Git : https://foo.example.org/repo.git
#Destination : /src/myapp
$ ansible webservers -m git -a "repo=https://foo.example.org/repo.git dest=/src/myapp version=HEAD"

Gérer les paquets

S'assurer qu'un paquet est installé sans l'upgrader
$ ansible webservers -m apt -a "name=nano state=present"
S'assurer qu'un paquet est installé dans la bonne version
$ ansible webservers -m apt -a "name=nano-4.8 state=present"
S'assurer qu'un paquet est à jours
$ ansible webservers -m apt -a "name=nano state=latest"
S'assurer qu'un paquet n'est pas installé, il est désinstallé si il est présent.
$ ansible webservers -m apt -a "name=nano state=absent"

Gérer les services

S'assurer qu'un service est bien démarré sur plusieurs serveurs
$ ansible webservers -m service -a "name=httpd state=started"
Redémarrer un service sur plusieurs serveurs
$ ansible webservers -m service -a "name=httpd state=restarted"
S'assurer qu'un service est stoppé sur plusieurs serveurs
$ ansible webservers -m service -a "name=httpd state=stopped

Gestion du système et des users

Récupérer les infos des serveurs
$ ansible webservers -m setup | less
Ajouter une règle de firewall
$ ansible webservers –m firewalld –a "service=https permanent=yes" state=enable“
Ajouter une crontab
$ ansible webservers -m cron -a “name=reboot minute=0 hour=5,2 job=/bin/reboot"
Créer un utilisateur
$ ansible webservers -m user -a "name=john password=<mot de passe crypté>"
Supprimer un utilisateur
$ ansible webservers -m user -a "name=john state=absent

Ansible Cheat sheet

Téléchargé depuis : www.sysadminascode.com/cheat-sheet-ansible

Sysadmin as code

Playbook

Avant d'utiliser les playbook, il faut s'assurer que les partages de clés vers les serveurs cibles sont OK.

Travailler avec des Playbook

```
# Créer un playbook
$ nano <nom_du_playbook>.yml
# Exécuter un playbook (option -vvv pour le mode debug)
$ ansible-playbook <nom_du_playbook>.yml
# Il est possible d'utiliser plusieurs options pour le lancement de playbook
-i pour spécifier l'inventaire ou l'ip d'un serveur directement
--check dry/run mode (aucune modif), --diff voir la différence avec les fichiers
-t pour jouer seulement certain tags
$ man ansible-playbook
https://docs.ansible.com/ansible/latest/user_guide/playbooks.html
```

Exemple de playbook

```
---
- hosts: webservers
  vars:
    http_port: 80
    remote_user: root
  tasks:
  - name: S'assurer que Apache est installé et à jour
    apt: name=httpd state=latest
  - name: Uploader la configuration d'apache
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: Lancer apache (Par défaut au démarrage)
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

YAML

Exemple d'utilisation de YAML

```
# Les commentaires sont précédés d'un signe "#", comme cette ligne.

#####  LES SCALAIRES #####
# Les scalaires sont l'ensemble des types YAML qui ne sont pas des collections
# (listes ou tableaux associatifs).
clé: valeur
valeur_numérique: 100
notation_scientifique: 1e+12
booléen: true
valeur_null: null

###### Les COLLECTIONS ######
# L'imbrication est créée par indentation.
une_map_imbriquée:
    clé: valeur
    autre_clé: autre valeur
    autre_map_imbriquée:
        bonjour: bonjour
# YAML autorise aussi l'usage des collections à l'intérieur des clés,
# Les séquences (équivalent des listes ou tableaux) ressemblent à cela :
une_séquence:
    - Objet 1
    - Objet 2
    - 0.5           # les séquences peuvent contenir des types variés.
    - clé: valeur
      autre_clé: autre_valeur

# YAML étant un proche parent de JSON, vous pouvez écrire directement
# des maps et séquences façon JSON
json_map: {"clé": "valeur"}
json_seq: [1, 2, 3, "soleil"]
```

https://learnxinyminutes.com/docs/fr-fr/yaml-fr/