02-01-Installation SaltSlack Orchestration

A - Description :

Le système Salt est un framework d'exécution à distance open source basé sur Python pour la gestion de la configuration, l'automatisation, le provisionnement et l'orchestration.

SaltStack est un logiciel de gestion de configuration et d'automatisation gratuit, open source et basé sur Python. Il s'agit d'un outil de ligne de commande qui vous aide à gérer votre infrastructure à partir d'un emplacement central. SaltStack est composé de quatre composants. Une brève explication de chaque composant est présentée ci-dessous :

- Salt Master agit comme un contrôleur de ligne de commande pour ses sbires. Il est utilisé pour contrôler et gérer un certain nombre de serviteurs.
- Les Salt Minions sont des démons esclaves qui reçoivent des configurations et des commandes du maître.
- La formule est constituée de fichiers de gestion de configuration.
- L'exécution est un nombre de commandes et de modules qui sont exécutés sur les minions.

Salt se distingue de ses homologues car son auteur a eu la bonne idée de ne pas construire un monstre s'appuyant sur des solutions de messagerie à la complexité douteuse mais a plutôt misé sur la librairie *ZeroMQ*[2], s'assurant ainsi vélocité et simplicité. Ce choix est loin d'être anodin car, sans aucun effort supplémentaire, *Salt* offre d'emblée des fonctionnalités de gestion de configuration **et** d'orchestration. Et nous allons voir que mettre le pied à l'étrier du cheval *Salt* n'a rien de l'insurmontable. Enfin, mais c'est évidemment très personnel, je n'ai aucune affinité avec le langage *Ruby* et au contraire des solutions sus-citées, *Salt* est écrit en python, plus doux à mes yeux. Cela peut paraître dérisoire à première vue, mais devant la forte probabilité de tentation d'écriture de modules, le langage de la solution **a** son importance.

Bien que jeune (Mai 2011), le projet a d'ores et déjà le soutien de noms prestigieux tels que HP, LinkedIn ou encore PayPal, se décline déjà en version commerciale et sait adresser la quasi totalité des *Clouds* publics et privés.

Enfin, la liste des services Libres que *Salt* est en mesure de gérer est impressionnante et ne cesse de grossir, tant le développement de modules est simple avec un strict minimum de connaissances en *python*[4].

1. Documentations :

https://connect.ed-diamond.com/GNU-Linux-Magazine/glmf-166/salt-l-autre-chef-d-orchestre

https://docs.saltproject.io/salt/install-guide/en/latest/topics/overview.html

https://fr.linux-console.net/?p=3335

https://fr.linux-console.net/?p=3486

https://docs.saltproject.io/salt/user-guide/en/latest/topics/overview.html

B - Installation du master :

1. Documentation d'installation en fonction de la distrib :

Documentation d'installation en fonction de la distrib/OS (Même Windows) :

https://docs.saltproject.io/salt/install-guide/en/latest/topics/install-by-operating-system/index.html

2. Installation de Salt Master :

```
# # Installation sur Ubuntu 22.04 :
sudo curl -fsSL -o /etc/apt/keyrings/salt-archive-keyring-2023.gpg
https://repo.saltproject.io/salt/py3/ubuntu/22.04/amd64/SALT-PROJECT-GPG-PUBKEY- 2023.gpg
echo "deb [signed-by=/etc/apt/keyrings/salt-archive-keyring-2023.gpg arch=amd64]
https://repo.saltproject.io/salt/py3/ubuntu/22.04/amd64 /dernier plat principal confit" | sudo tee
/etc/apt/sources.list.d/salt.list
```

```
# Mise à jour des dépôts :
sudo apt-get update

# Installation de salt :
sudo apt-get install salt-master salt-minion salt-ssh salt-syndic salt-cloud salt-api

# Activer et démarrer les services :
sudo systemctl enable salt-master && sudo systemctl start salt-master
sudo systemctl enable salt-minion && sudo systemctl start salt-minion
sudo systemctl enable salt-syndic && sudo systemctl start salt-syndic
sudo systemctl enable salt-api && sudo systemctl start salt-api
```

3. Configurer le Salt Master :

Documentation :

https://docs.saltproject.io/salt/install-guide/en/latest/topics/configure-master-minion.html

Fichier de configuration :

Pour une configuration de base de Salt, il vous suffit de modifier le fichier de configuration du minion Salt pour ajouter l'adresse IP du maître Salt auquel il se connectera. Voir Configuration de base des minions pour plus d'informations.

- Le `salt-master`service est livré avec des configurations de serveur par défaut.
- La configuration YAML principale par défaut `/etc/salt/master`contient tous les paramètres commentés.
- Recommandé : vous pouvez ajouter des paramètres personnalisés dans YAML sous `/etc/salt/master.d/`forme `.conf` de fichiers sur Salt master.
- Utilisez le fichier maître par défaut comme référence pour divers paramètres selon vos besoins.

Bien que le `/etc/salt/master`fichier puisse accepter les paramètres de configuration, la meilleure pratique consiste à utiliser le `/etc/salt/master.d/`répertoire de configuration. L'utilisation de ce répertoire vous permet de placer les options de configuration dans des séparations logiques.

Par exemple, si vous souhaitez configurer un nombre différent de `worker_threads`, vous pouvez stocker ces configurations dans le `/etc/salt/master.d/tuning.conf` répertoire et conserver toutes les configurations liées au réglage dans ce fichier.

Paramètre du réseau :

```
# Editer le fichier (A créer) :
/etc/salt/master.d/network.conf

# Contenu :
# The network interface to bind to
interface: 192.168.1.14

# The Request/Reply port
ret_port: 4506

# The port minions bind to for commands, aka the publish port
publish_port: 4505
```

Gestion des processus Salt Master :

Si votre cluster contient des milliers de serviteurs et que vos rapports sur les serviteurs sont bloqués, le maître peut retarder les réponses des serviteurs de la tâche. Cela peut signifier que les serviteurs ont échoué dans leur travail, mais cela pourrait plutôt signifier que le maître ne dispose pas de suffisamment de threads de travail pour traiter tous les rapports.

Pour gérer les `salt-minion`appels de retour, le maître enchaîne les processus de travail avec le `worker_threads`paramètre. La limite par défaut pour les processus est de cinq travailleurs. La limite minimale est de trois travailleurs.

Exemple de paramètre dans un fichier de configuration principal :

```
# Editer le fichier :
vi /etc/salt/master.d/thread_options.conf

# Contenu :
worker_threads: 5
```

Normes pour les environnements très fréquentés :

- Utilisez un thread de travail pour 200 serviteurs.
- La valeur de `worker_threads`ne doit pas dépasser 1½ fois les cœurs de processeur disponibles.

C - Installation des Minions :

1. Installation des Minions :

Documentation :

https://docs.saltproject.io/en/latest/py-modindex.html
https://salt-zh.readthedocs.io/en/latest/ref/modules/all/salt.modules.cp.html

Description :

- Le `salt-minion`service est livré par défaut avec une configuration de configuration DNS/nom d'hôte.
- La configuration YAML du minion par défaut `/etc/salt/minion`contient tous les paramètres commentés.
- Recommandé : vous pouvez ajouter des paramètres personnalisés dans YAML sous `/etc/salt/minion.d/` forme `.conf`de fichiers sur le serviteur.
- Utilisez le fichier minion par défaut comme référence pour divers paramètres selon vos besoins.

REPO - CentOS 7 :

```
sudo yum install https://repo.saltstack.com/yum/redhat/salt-repo-latest.el7.noarch.rpm
sudo yum clean expire-cache
sudo yum install salt-minion
sudo yum update
```

REPO - Ubuntu 18.04 :

```
sudo mkdir /etc/apt/keyrings
sudo curl -fsSL -o /etc/apt/keyrings/salt-archive-keyring.gpg
https://repo.saltproject.io/salt/py3/ubuntu/18.04/amd64/latest/salt-archive-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/salt-archive-keyring.gpg arch=amd64]
https://repo.saltproject.io/salt/py3/ubuntu/18.04/amd64/latest bionic main" | sudo tee
/etc/apt/sources.list.d/salt.list


apt update
sudo apt-get install salt-minion
```

REPO -Ubuntu 20.04 :

```
mkdir /etc/apt/keyrings
sudo curl -fsSL -o /etc/apt/keyrings/salt-archive-keyring.gpg
https://repo.saltproject.io/py3/ubuntu/20.04/amd64/3004/salt-archive-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/salt-archive-keyring.gpg arch=amd64]
https://repo.saltproject.io/py3/ubuntu/20.04/amd64/3004 focal main" | sudo tee /etc/apt/sources.list.d/salt.list

apt update
sudo apt-get install salt-minion
```

REPO -Ubuntu 22.04 :

```
apt update
sudo apt-get install salt-minion
```

2. Configurer le Salt Minion :

Documentation :

https://docs.saltproject.io/salt/install-guide/en/latest/topics/configure-master-minion.html

Bonnes pratiques :

Bien que `/etc/salt/minion`le fichier puisse accepter les paramètres de configuration, la meilleure pratique consiste à utiliser le `/etc/salt/minion.d/`répertoire de configuration. L'utilisation de ce répertoire vous permet de placer les options de configuration dans des séparations logiques.

> **Avertissement :**
> Lorsque vous utilisez plusieurs .conffichiers, veillez à ne pas utiliser de paramètres de configuration en double. (Par exemple, définir le nombre de threads de travail dans plusieurs fichiers de configuration.)
> Salt applique les paramètres du dernier .conffichier qu'il évalue et évalue les fichiers par ordre alphabétique. Si vous utilisez des paramètres de configuration en double, vous pourriez accidentellement remplacer le paramètre que vous vouliez appliquer.

Configuration du Minion :

Se connecter sur l'hôte esclave (le minion)

```
# Configuration de la connexion au Salt Master :
vi /etc/salt/minion.d/master.conf

# Contenu :
master: admin-l.v.sdem.fr
```

Le `salt-minions`'identifiera auprès du maître par le nom d'hôte du système, sauf indication contraire :

La plupart des chaînes sont autorisées. Si vous décidez de personnaliser vos identifiants de serviteur, essayez de garder l'identifiant bref mais descriptif de son rôle. Par exemple, vous pouvez utiliser `apache-server-1`pour nommer l'un de vos serveurs Web ou utiliser le nom `datacenter-3-rack-2`de son emplacement dans un centre de données. L'objectif est de rendre les noms descriptifs et utiles pour référence future.

```
# Configgguration de l'identifiant du Minion :
vi /etc/salt/minion.d/id.conf

# Contenu
id: FQDN
```

Redémarrer le Minion :

```
service salt-minion restart
```

D - Gestion des clés des Minions :

1. Documentations :

https://docs.saltproject.io/salt/install-guide/en/latest/topics/accept-keys.html#accept-keys

2. Description :

La dernière étape du processus d'installation consiste pour le maître Salt à accepter les clés de minion Salt. Une fois les clés acceptées, le maître Salt peut émettre des commandes au serviteur

et recevoir des messages entrants du serviteur. Les serviteurs Salt ne reçoivent pas de données du maître Salt tant que la clé n'est pas acceptée.

Pour des raisons de sécurité, Salt utilise une authentification par clé.

Deux types de clés sont utilisés dans Salt :

- RSA
- AES

Les clés RSA constituent l'épine dorsale du modèle d'authentification et de chiffrement utilisé par Salt. Tous les démons Salt fonctionnent avec des clés RSA uniques. Les serviteurs et le maître génèrent des clés RSA lors de leur premier démarrage, puis les utilisent pour l'authentification basée sur PKI.

Ces clés sont utilisées pour authentifier la clé AES auprès du maître Salt, fournissant ainsi une communication sécurisée en cryptant les données. Chaque serviteur présente une clé publique au maître Salt. La clé est ensuite examinée, comparée et explicitement acceptée par un administrateur.

Le maître envoie également une clé AES tournante qui est utilisée pour chiffrer et déchiffrer les messages envoyés par le maître Salt. La clé AES renvoyée est chiffrée à l'aide de la clé publique initialement envoyée par le minion Salt et ne peut donc être déchiffrée que par le même minion Salt.

image.png
Image not found or type unknown

La clé AES subit une rotation dans l'une ou l'autre de ces conditions :

- Toutes les 24 heures sur le master
- Lorsque le maître est redémarré
- Lorsqu'une clé de serviteur est supprimée.

La rotation des clés permet au maître de verrouiller les serviteurs qui ne sont pas authentifiés et permet le chiffrement des communications à l'échelle du système.

3. Gestion des clés :

La plupart des gestions des minions sont gérées via un client appelé `salt-key` qui s'exécute sur le maître Salt. Ce client est utilisé pour gérer les serviteurs disponibles pour un maître.

Vous pouvez utiliser la `salt-key` commande pour vous connecter au système d'authentification afin d'accepter, de rejeter et de gérer les clés.

Accepter les clés :

Lorsqu'un nouveau serviteur s'enregistre, la clé attendra jusqu'à ce qu'elle soit acceptée.`Unaccepted keys`

Appelez `salt-key`pour voir l’état actuel de la gestion des clés :

```
# Appelez salt-keypour voir l’état actuel de la gestion des clés :
salt-key

# Exemple de réponse :
oot@admin-l:/etc/salt/master.d# salt-key
Accepted Keys:
nextcloud.v.sdem.fr
Denied Keys:
Unaccepted Keys:
web01.v.sdem.fr
Rejected Keys:
```

Commande de gestion des clés :

```
# Accepter une clé :
salt-key -a FQDN

# Accepter toutes les clés :
salt-key -A

# Supprimer une clé :
salt-key -d FQDN

# Supprimer toutes les clés :
salt-key -D

# Supprimer des clés à l'aide d'un filtre :
salt-key -d 'web*'
```

Il est possible d'accepter ou rejeter automatiquement les clés.

Revision #1
Created 8 February 2024 08:09:59 by Jérôme
Updated 3 March 2026 08:05:49 by Jérôme