03-01 - Exploitation de SaltProject

A - Commandes de base :

1. Commande de base :

```
salt '*' grains.ls
salt '*' grains.items
    locale_info:
        ----------
        defaultencoding:
            UTF-8
        defaultlanguage:
            fr_FR
        detectedencoding:
            utf-8
        timezone:
            UTC
    lsb_distrib_description:
        Ubuntu 22.04.3 LTS
    mem_total:
        1963
    num_cpus:
        2

salt '*' sys.doc
```

2. Gestion des grains :

Source : https://docs.saltproject.io/en/latest/ref/modules/all/salt.modules.grains.html

```
salt -G 'os:CentOS' test.version
salt -G 'cpuarch:x86_64' grains.item num_cpus
salt -G 'ec2_tags:environment:*production*'

salt '*' grains.item os osrelease oscodename
salt '*' grains.item host sanitize=True
salt '*' grains.items
salt '*' grains.items sanitize=True

# Renvoie une liste de tous les grains disponibles
salt '*' grains.ls
```

1. Gestion des groupes de minions :

Sources

https://salt-zh.readthedocs.io/en/latest/topics/targeting/nodegroups.html
https://docs.saltproject.io/en/master/topics/targeting/compound.html#targeting-compound
https://docs.saltproject.io/en/master/topics/targeting/nodegroups.html#defining-nodegroups-as-lists-of-minion-ids

Gestion des groupes

Créer un fichier .conf dans le dossier **/etc/salt/master.d** contenant les informations de groupe :

```
# Création d'un groupe dans un fichier dédié :
nodegroups:
  grp-srv-authorized_keys-dev-apside:
    - 'proto-novo.v.sdem.fr'

# Test du groupe :
salt -N grp-srv-authorized_keys-dev test.version
```

3. Commandes simples :

```
# Effectuer un test de ping sur les minions
salt nextcloud.v.sdem.fr test.ping
salt '*' test.ping
```

```
salt '*.v.sdem.fr' test.ping

# Vérifier la version de Salt sur les minions :
salt '*' test.version

# Afficher la version des Distrib des minions :
salt '*' cmd.run 'uname -a'
salt '*' cmd.run 'lsb_release -a'
salt '*' grains.item os osrelease oscodename

# Afficher l'uptime des minions :
salt '*' cmd.run 'uptime'

# Afficher la configuration réseau :
salt '*' network.interfaces

# Lancer des commandes à distance :
salt '*' cmd.run 'ls -l /etc'
salt '*' cmd.run 'du -sh /*'
salt '*' cmd.run 'find /home/share/backup-si/* -ctime +1 -exec rm -rf {} \;'
salt '*' cmd.run 'shutdown -r now'

# Afficher les informations des disques des minions :
salt '*' disk.usage

# Afficher la conso de la RAM sur les minions :
salt '*' cmd.run 'free -m'

# Vérifiez le package bash à l'aide de la commande salt ci-dessous.
salt '*' pkg.show bash

# Installez un package sur tous les serveurs minion.
salt '*' pkg.install swaks
salt '*' pkg.install tree
salt '*' pkg.install postfix

# Afficher les référentiels disponibles sur tous les serveurs Minion.
salt '*' pkg.refresh_db
```

```
# Vérifiez la liste des référentiels sur tous les serviteurs du serveur.
salt '*' pkg.list_repos

# Consultez la liste des mises à jour de packages disponibles sur tous les serveurs Minion.
salt '*' pkg.list_upgrades

#### Un autre excellent exemple ici est le module de service qui vous permet de gérer des services sur plusieurs distributions Linux, y compris la distribution avec systemd.
# Vérifiez si le service apache2 est disponible ou non.
salt '*' service.available apache2

# Activez le service apache2 pour qu'il démarre au démarrage du système sur tous les serveurs minion.
salt '*' service.enabled apache2

# Vérification de la liste des services en cours d'exécution sur tous les serveurs.
salt '*' service.get_running

# Vérifiez la commande ExecStart= pour chaque service disponible sur tous les serveurs.
salt '*' service.execs

# Afficher le stockage BTRFS des Serveurs LXC :
salt -N grp-srv-lxc cmd.run 'btrfs filesystem show'
```

4. Copie d'un fichier vers un minion :

```
# Copier récursivement un répertoire depuis le salt master
salt '*' cp.get_dir salt://path/to/dir/ /minion/dest

# Copier un fichier vers les minions :
salt '*' cp.get_file salt://path/to/file /minion/dest
salt '*' cp.get_file "salt://{{grains.os}}/vimrc" /etc/vimrc template=jinja
salt 'nextcloud*' cp.get_file salt:/srv/salt/files/test.txt /home/adminme/test.txt

salt 'nextcloud.v.sdem.fr' cp.get_file salt://files/ssh-key-pub-listing-infra.txt /home/adminme/.ssh/authorized_keys
```

5. Déploiement des clés SSH :

############ Préparation des fichiers de conf :

Création du dossier sshd_config.d s'il n'existe pas, le fichier sshd_config utilise l'include du dossier sshd_config.d
salt '*' cmd.run "mkdir /etc/ssh/sshd_config.d"

Déploiement du fichier de gestion des clés SSH :
salt '*' cmd.run "rm -rf /etc/ssh/sshd_config.d/authorizedKeysFile.conf"
salt '*' cmd.run "echo "#AuthorizedKeysFile\nAuthorizedKeysFile .ssh/authorized_keys-infra .ssh/authorized_keys-dev .ssh/authorized_keys-dev-apside\n " >> /etc/ssh/sshd_config.d/authorizedKeysFile.conf"

Pour les Ubuntu 18.04, le Include du dossier sshd_config.d ne fonctionne pas :
vi /etc/ssh/sshd_config
JKT - Ajout AuthorizedKeys :
AuthorizedKeysFile .ssh/authorized_keys-infra .ssh/authorized_keys-dev .ssh/authorized_keys-dev-apside

Déploiement du fichier de configuration dédié guacamole uniquement sur les Ubuntu 22.04 'Jammy' :
salt -G 'oscodename:jammy' cmd.run 'echo "# JKT-20220517 - Connexion Guacamole :\nHostkeyAlgorithms +ssh-rsa\nPubkeyAcceptedAlgorithms +ssh-rsa\n " >> /etc/ssh/sshd_config.d/guacamole-auth-ssh.conf'

Redémarrer le deamon SSHD :
salt '*' cmd.run 'service sshd restart'

############ Déploiement des clés SSH :
Options 1 - Accès service INFRA uniquement - Fichier -> .ssh/authorized_keys-infra :
salt '*' cp.get_file salt://files/authorized_keys-infra /home/adminme/.ssh/authorized_keys-infra
salt '*' cmd.run 'chown adminme:adminme /home/adminme/.ssh/*'

Options 2 - Accès service DEV - Fichier -> .ssh/authorized_keys-dev :
Utilisation du groupe de minion grp-srv-authorized_keys-dev
salt -N grp-srv-authorized_keys-dev cp.get_file salt://files/authorized_keys-dev /home/adminme/.ssh/authorized_keys-dev
salt -N grp-srv-authorized_keys-dev cmd.run 'chown adminme:adminme /home/adminme/.ssh/*'

Options 3 - Accès service Prestataire DEV-APSIDE - Fichier -> .ssh/authorized_keys-dev-apside :
Utilisation du groupe de minion grp-srv-authorized_keys-dev-apisde
salt -N grp-srv-authorized_keys-dev-apside cp.get_file salt://files/authorized_keys-dev-apside

```
/home/adminme/.ssh/authorized_keys-dev-apside
salt -N grp-srv-authorized_keys-dev-apside cmd.run 'chown adminme:adminme /home/adminme/.ssh/*'

# Suppression de l'ancien fichier de configuration après déploiement des clés spécifiques :
salt '*' cmd.run 'rm /home/adminme/.ssh/authorized_keys'
```

6. Salt avec VMware :

Source

https://docs.vmware.com/fr/VMware-vRealize-Automation-SaltStack-Config/8.6/use-manage-saltstack-config-guide.pdf
https://docs.saltproject.io/salt/extensions/salt-ext-modules-vmware/en/latest/ref/modules/saltext.vmware.modules.vm.html

B - Gestion par script :

1. Description :

L'ensemble des scripts sont installés sur le serveur admin-l.v.sdem.fr dans /srv/salt/scripts/

Les scripts sont disponible dans git

1. Script de déploiement de Lynis :

Lynis est un outil de scan de vulnérabilité. Il est déployé automatiquement avec Salt sur l'ensemble du parc.

Emplacement du script : /srv/salt/scripts/

Nom du script : salt-deploy-lynis.sh

```
#############################################################
# Auteur : jkermorvant
# Date de modification : 20231115
#
# Description : Script de déploiement de l'outil de sca de vulnérabilité Lynis
```

```
#
#################################################################
#! /bin/bash
#
#################################################################
# Variables :

####################################
# Déploiement de l'outil de scan de vulnérabiloté Lynis :
salt '*' cp.get_file salt://files/lynis-3.0.9/lynis-3.0.9.tar.gz /tmp
salt '*' cmd.run 'tar -xzf /tmp/lynis-3.0.9.tar.gz -C /usr/local'
salt '*' cmd.run 'chmod +x /usr/local/lynis/lynis'
salt '*' cmd.run 'ln -s /usr/local/lynis/lynis /usr/local/bin'

# Démarrage du scan de vulnérabilité :
salt '*' cmd.run 'sudo lynis audit system'

# Récupération des logs vers le dossier /var/cache/salt/master/minions/minion-id/files :
# Prérequis :
# Source : https://docs.saltproject.io/en/latest/ref/modules/all/salt.modules.cp.html#salt.modules.cp.push
# Allow minions to push files to the master. This is disabled by default, for security purposes.
# file_recv: False -> True
# Restart Salt-Master
salt '*' cp.push /var/log/lynis.log
salt '*' cp.push /var/log/lynis-report.dat
```

1. Script de déploiement de la supervision :

Ce script permet de déployer la supervision sur une machine spécifique Linux.

Emplacement du script : /srv/salt/scripts/

Nom du script : salt-deploy-supervision.sh

```
#################################################################
# Auteur : jkermorvant
# Date de modification : 20231115
```

```
#
# Description : Script de déploiement de la supervision NAGIOS
#
#################################################################
#! /bin/bash
#
#################################################################
# Variables :
ARG=N
DEST=grp-srv-deploy-labs

# Déclaration du serveur
SERVER=web01
DOMAIN=.v.sdem.fr

# Déclaration de l'adresse IP du serveur
IP1=192
IP2=168
IP3=1
IP4=219

# Déclaration du serveur de supervision
SUPERVISION=supervision.v.sdem.fr

###################################
# Installation et configuration de l'agent NRPE sur le serveur cible

# Installation du paquet NRPE :
salt ${SERVER}${DOMAIN} pkg.install nagios-nrpe-server nagios-plugins

# Déploiement des fichiers de configuration sur le serveur à superviser :
salt ${SERVER}${DOMAIN} cp.get_file salt://files/supervision/nrpe.cfg /etc/nagios
salt ${SERVER}${DOMAIN} cp.get_file salt://files/supervision/check_mem /usr/lib/nagios/plugins/

# Redémarrage du service nagios-nrpe :
salt ${SERVER}${DOMAIN} cmd.run 'service nagios-nrpe-server restart'

###################################
# Déployer le fichier de configuration Nagios sur le serveur de supervision
```

```
# Déploiement du fichier de configuration sur le serveur de supervision :
cp /srv/salt/files/supervision/template.v.sdem.fr.cfg /srv/salt/files/supervision/files-
config/${SERVER}${DOMAIN}.cfg

# Paramétrage du fichier de configuration Nagios - Remplacement du nom :
for file in /srv/salt/files/supervision/files-config/${SERVER}${DOMAIN}.cfg
do
  echo "Traitement de ${file} ..."
  sed -i -e "s/template\.v\.sdem\.fr/${SERVER}\.v\.sdem\.fr/g" "${file}"
done

# Paramétrage du fichier de configuration Nagios - Remplacement de l'adresse IP :
for file in /srv/salt/files/supervision/files-config/${SERVER}${DOMAIN}.cfg
do
  echo "Traitement de $file ..."
  sed -i -e "s/192\.168\.1\.XXX/${IP1}\.${IP2}\.${IP3}\.${IP4}/g" "${file}"
done

# Déploiement du fichier de configuration sur le serveur de supervision :
salt ${SUPERVISION} cp.get_file salt://files/supervision/files-config/${SERVER}${DOMAIN}.cfg
/usr/local/nagios/etc/objects/sdem/

# Application des droits sur le fichier :
salt ${SUPERVISION} cmd.run "chown sdem:sdem
/usr/local/nagios/etc/objects/sdem/${SERVER}${DOMAIN}.cfg"

# Redémarrage du service nagios-nrpe :
salt ${SUPERVISION} cmd.run "sudo systemctl restart nagios.service"
```

1. Script de mise à jour de Metabase sur Dataviz :

Ce script permet de déployer la mise à jour de Metabase sur les serveurs

Emplacement du script : /srv/salt/scripts/

Nom du script : salt-deploy-supervision.sh

```
#################################################################
# Auteur : jkermorvant
# Date de modification : 20231215
#
# Description : Script de mise à jour de Metabase
#
#################################################################
#! /bin/bash
#
#################################################################
# Variables :
SERVER=dataviz-dev.v.sdem.fr
VERSION=0.47.9

#################################################################
# 1 - Executer le script de dump postgresql
salt ${SERVER} cmd.run "sudo -u postgres pg_dump upciti_db > /backup/postgresql/upciti_db_bck"

# 2 - Stopper le service
salt ${SERVER} cmd.run "sudo systemctl stop metabase.service"

# 3 - Télécharger metabase.jar dans le répertoire d'installation (VERSION = 0.47.8 actuellement)
salt ${SERVER} cmd.run "sudo rm /data/metabase_data/metabase.jar"
salt ${SERVER} cmd.run "sudo wget -P /data/metabase_data
https://downloads.metabase.com/v${VERSION}/metabase.jar"
salt ${SERVER} cmd.run "chown -R metabase_usr:metabase_grp /data/metabase_data/metabase.jar"

# 4 - Démarrer Metabase
salt ${SERVER} cmd.run "systemctl start metabase.service"
salt ${SERVER} cmd.run "systemctl status metabase.service"
```

Revision #1
Created 8 February 2024 08:11:04 by Jérôme
Updated 3 March 2026 08:05:50 by Jérôme