SaltProject

01-Description

A - Description :

VM en remplacement du serveur physique installé le 30/07/2013 (Dell R610)
Type : VMware

Distrib : Ubuntu 22.04
Nom DNS : admin-l.v.sdem.fr
Adresse IP : 192.168.1.15

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-minion`s'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.

```
# Configguration 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.

02-Installation

A - Installation

1. Installation des outils de base :

```
apt-get install curl wget git swaks
```

B - Cacti

> TODO

C - Postfix

1. Installation de postfix :

Postfix est utilisé pour l'envoi de mail des scripts.

D - Rancid

> TODO

E - Samba

1. Description :

Le serveur Samba est utilisé pour la sauvegarde des données du SI provenant de l'AD vers le serveur d'admin. Un script Powershell se lance tous les jours pour transfert les données.
En cas de problème d'accès à l'AD, les données (Zim et Keepass) restent accessible afin de réparer l'infra.

2. Documentation :

> https://computingforgeeks.com/install-and-configure-samba-server-share-on-debian-ubuntu/

3. Installation :

```
sudo apt -y install samba
```

4. Configuration :

Configurer le partage sécurisé de Samba :

```
# Création du répertoire à partager dans Samba :
sudo mkdir -p /home/share

# Définir la propriété de groupe du répertoire de partage Samba sur le groupe sambashare
sudo chgrp sambashare /home/share

# Création des utilisateurs de partage de samba :
sudo useradd -M -d /home/share/backup-si -s /usr/sbin/nologin -G sambashare backup-si
sudo mkdir /home/share/backup-si
sudo chown backup-si:sambashare /home/share/backup-si
sudo chmod 2770 /home/share/backup-si

# Définir le mot de passe utilisateur et confirmez :
sudo smbpasswd -a backup-si

# Activer le compte samba après avoir défini le mot de passe :
sudo smbpasswd -e backup-si
```

Configurer le fichier de paramétrage Samba :

Editer le fichier de paramétrage de Samba "/etc/samba/smb.conf"
Contenu des paramètres :

```
[backup-si]
    path = /home/share/backup-si
    read only = no
    browseable = no
    force create mode = 0660
    force directory mode = 2770
    valid users = @backup-si @adminme @sambashare
    # Tunning :
    case sensitive = true
    default case = lower
    preserve case = no
    short preserve case = no
```

Redémarrez le démon samba après avoir effectué les modifications.

```
sudo systemctl restart smbd nmbd
```

Ajout du compte adminme au dossier backup-si :

```
vi /etc/group
# Modifier la ligne ci-dessous :
sambashare:x:122:backup-si,adminme
```

F - Smokeping

1. Description :

Les problèmes réseaux et de qualités sont souvent difficiles à comprendre et à analyser ...encore faut il les trouver. Chercher la source des perturbations chez votre client peut être un longue traversée du désert.

Smokeping peut vous y aider, cet article vous servira pour savoir comment installer et configurer ce logiciel de supervision.

SmokePing est un logiciel qui vous permet de conserver un historique de la latence de votre réseau.

Il est conçu par Tobi Oetiker, le créateur de MRTG et RRDtool. SmokePing est prévu pour calculer le RTT (Real Trip Time) entre les différents composants de votre réseau et peut le faire aussi bien avec un basique ping ICMP qu'avec des pings plus spécifiquement adaptés aux protocoles HTTP, SMTP, LDAP, DNS...

2. Documentation :

https://blog.wildix.com/fr/installation-configuration-de-smokeping/
https://oss.oetiker.ch/smokeping/doc/index.en.html

Installation :

```
apt install smokeping
```

Configuration :

```
cd /etc/smokeping/config.d/
```

> URL d'accès : http://admin-l.v.sdem.fr/cgi-bin/smokeping.cgi

G- Syslog

> TODO

03-Exploitation

B - Backup-SI

1. Description :

Le serveur d'admin est accessible en SSH soit avec le clé SSH ou bien avec le login "adminme" et son mot de passe (Identique au Keepass Informatique afin que le service informatique puisse le connaître).

Le service Samba est installé permettant de transférer depuis l'AD01 les informatique du SI (Keepass et Zim) vers un espace local sur le serveur admin.v.sdem.fr:/home/share/backup-si/
Un lien symbolique est créé sur le compte adminme dans "/home/adminme"

```
lrwxrwxrwx 1 adminme adminme   22 Aug 18 08:59 backup-si -> /home/share/backup-si//
```

2. Emplacement du backup :

Le "Zim" et le "Keepass" du service informatique sont copiés dans "adminme@admin.v.sdem.fr:/home/share/backup-si/%YYYYMMDD%".

Le backup est géré par un script Powershell sur le serveur AD01 qui va se connecter en SSH sur le serveur admin.v.sdem.fr avec le compte samba "backup-si".

C - Procédure-récupération-données-SI

Copie des données de l'infrastructure du SI de Morbihan Energies vers le serveur physique d'administration local avec une rétention de 30 jours.

Sources copiées :
    - Dossier Keepass

- Dossier Zim

Destination :
admin.v.sdem.fr:/home/share/backup-si/%YYYYMMDD% (AnnéeMoisJour)

Accès aux données :
En cas de besoin d'accès aux données, suivre la procédure ci-dessous :
- Se connecter avec un client FTP (Filezilla) sur adminme@admin.v.sdem.fr (Password identique au Keepass Informatique)
- Cliquer sur le lien symbolique backup-si
- Sélectionner le dossier correspondant à la date souhaitez et le transférer localement sur le poste

D - crontab

```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  *  user command to be executed
#
# Archive - mailreboot :
#20 12 * * 1,3 cd /root && ./mailreboot.sh
#

# ssl-cert-check :
00 01 * * * cd /usr/local/bin/ssl-cert-check/ && ./ssl-cert-check.sh -a -f ssldomains -q -x 60 -e
hostmaster@morbihan-energies.fr

## url-check :
# Demarrage script url-check.sh - Intervale 5mn :
*/05 * * * * cd /usr/local/bin/url-check/ && ./url-check.sh >/dev/null 2>&1

# Purge dossier de cache URL toutes les heures :
```

```
* */01 * * * rm -f /usr/local/bin/url-check/cache/*

# Purge dossier /home/share/backup-si des dossiers de plus de 30 jours :
00 01 * * * cd /usr/local/bin/ && ./purge-backup-si.sh 2>&1
```

E - Racktables - ARCHIVE

1. Création d'un accès en lecture seul pour script de Cable-ID :

```
# Création du compte :
CREATE USER 'readonly'@'localhost' IDENTIFIED BY 'PASSWORD';

# Attribue des accès en lecture seule :
GRANT SELECT ON *.* TO 'readonly'@'localhost';

# Application des changements :
FLUSH PRIVILEGES;
```

Cette procédure est à transférer sur le serveur racktables lors de la migration de Racktables en CT LXC.

F - Scripts

Ce serveur permet d'orchestrer un ensemble de script prévu pour l'administration du système dr'information de Morbihan Energies.

```
Emplacement des scripts :
/usr/local/bin/
```

1. purge-si-backup.sh

Description :

Purger le dossier "/home/share/backup-si" de tous les dossiers dont la date de création est supérieur à 30 jours.

Emplacement du script : /usr/local/bin/purge-backup-si.sh

```
#!/bin/bash


# Author : jkermorvant
# Version : 1.0
# Description : Purge du dossier /home/share/backup-si de tous les dossiers dont la date de création est suppérieur à 30 jours.
find /home/share/backup-si/* -ctime +30 -exec rm -rf {} \;
```

2. ssl-cert-check-master

Description :

Ce script permet d'alerter quand l'expiration des certificats approche des 60 jours.

Source :

> https://github.com/Matty9191/ssl-cert-check
> https://prefetch.net/articles/checkcertificate.html

Fonctionnement :

Le fichier "/usr/local/bin/ssl-cert-check-master/ssldomains" contient la liste des sites en SSL à scanner

```
morbihan-energies.fr 443
sdem.fr 443
```

Le fichier "/usr/local/bin/ssl-cert-check-master/ssl-cert-check" est le script qui permettra de vérifier l'état des certificats des sites web déclarés dans ssldomains.
La ligne 388 du script a été modifié pour configurer le FROM :

```
#   FROM="${1}"
    FROM="hostmaster@morbihan-energies.fr"
```

Exemple d'utilisation :

```
# Vérifier l'expiration des certificats listés dans le fichier ssldomains
./ssl-cert-check -f ssldomains

# Vérifier
./ssl-cert-check -a -f ssldomains -q -x 60 -e hostmaster@morbihan-energies.fr
```

Crontab :

Une tâche planifiée est créé pour analyser tous les jours à 1h00 l'état des certificats.

```
# Editer la crontab
crontab -e

# Crontab pour le ssl-cert-check :
00 01 * * * cd /usr/local/bin/ssl-cert-check-master/ && ./ssl-cert-check -a -f ssldomains -q -x 60 -e
hostmaster@morbihan-energies.fr
```

3. testssl.sh

Description :

La mise en place du chiffrement sur un serveur Web peut être parfois assez complexe en fonction du niveau de sécurité que vous souhaitez atteindre tout en prenant en compte la compatibilité avec les différents navigateurs et terminaux du marché.
Pour tester que votre serveur possède une configuration correcte, deux options s'offrent à vous :
- Dans le cas le plus simple, votre serveur possède déjà une entrée DNS publique, vous pourrez tout simplement utiliser le site SSLLabs pour vérifier la configuration de celui-ci. Une note de A est assez facilement atteignable en suivant les recommandations de safeciphers.
- Dans le cas contraire, si le DNS de votre serveur n'est pas encore exposé sur Internet, il faudra utiliser une autre solution : Testssl.sh

Testssl.sh est un script Unix permettant de vérifier la sécurité de votre configuration HTTPS. Il permettra de vous confirmer la compatibilité de votre site avec une exhaustivité de systèmes : Android, IE, Safari, Java ...

Documentation :

> https://www.geeek.org/testssl-test-tls-ssl-serveur/

Installation :

```
cd /usr/local/bin/
git clone https://github.com/drwetter/testssl.sh
mv testssl.sh testssl
cd testssl
chmod +x testssl.sh
```

Exploitation :

./testssl.sh site.morbihan-energies.fr

testssl.png
Image not found or type unknown

4. url-check

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"
```

telechargement - lufi

A - Création du conteneur LXC

1. Configuration du serveur :

Nom LXC : TELECHARGEMENT
Adresse IP (réseau non rroté LXC) : 192.168.25.160
Nom DNS Interne : telechargement.v.sdem.fr
Nom DNS Externe : telechargement.morbihan-energies.fr
URL : telechargement.morbihan-energies.fr
Reverse Proxy : telechargement.morbihan-energies.fr
OS : Ubuntu 22.04

Rôle :

Serveur de transfert de fichiers volumineux

1. Création d'un CT :

Le container est installé sur le serveur LXC01.v.sdem.fr

```
# Déclarer la variable :
CTNAME=TELECHARGEMENT
# Création du conteneur :
lxc launch ubuntu:22.04 $CTNAME


# Configurer l'autostart d'un CT :
lxc config set $CTNAME boot.autostart true
```

2. Paramétrage :

Editer le fichier de configuration afin de paramétrer une adresse IP fixe et le forward de port.

Configuration réseau et forward de port :

Il est nécessaire de bien renseigner le fichier de zone DNZ "lxc.morbihan-energies.fr" afin d'attribuer une adresse IP sur le réseau LXC (non routé).

```
# Editer le fichier de configuration :
lxc config edit $CTNAME
```

Ajouter les lignes suivantes :

```
  devices:
    eth0:
      ipv4.address: 192.168.25.160
      name: eth0
      network: lxdbr0
      type: nic
  port53087:
      connect: tcp:0.0.0.0:80
      listen: tcp:192.168.1.31:53087
      type: proxy
```

B - Paramétrage :

1. Gestion date et heure :

```
# Changer la date :
date -s "02/18/2017 12:34:00"

# Changer le fuseau horaire :
rm /etc/localtime
ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime
# Pour que le changement de fuseau survive au redémarrage de la machine, il faut aussi le mettre dans le bon fichier.
vi /etc/timezone
ZONE="Europe/Paris"
```

2. Documentations :

> https://framacloud.org/fr/cultiver-son-jardin/lufi
> https://framagit.org/fiat-tux/hat-softwares/lufi/-/wikis/installation

3. Prérequis :

Prérequis :

Lufi est codé en Perl, pour le faire fonctionner il est nécessaire d’installer Carton, un gestionnaire de modules Perl.

```
apt-get install build-essential libpq-dev
cpan Carton
```

carton est un outil en ligne de commande pour suivre les dépendances du module Perl pour votre application Perl.
Les dépendances requises sont gérées via un fichier nommé cpanfile et suivies via le fichier carton.lock.
Cela facilite les déploiements et permet aux autres développeurs de votre application d'avoir exactement les mêmes versions des modules.

C - Installation des prérequis

Configuration réseau :

Nom DNS : telechargement.morbihan-energies.fr

Type de serveur : CT LXC

Adresse IP : 192.168.25.160
DNS : 192.168.1.50
Distrib : Ubuntu 22.04

Documentations :

L'ensemble de l'installation se fait en root.

D - Installation

2. Installation de LUFI :

Tout d'abord, connectez-vous en tant que root sur votre serveur et créez un compte utilisateur lufi ainsi que le dossier /var/www/html/lufi dans lequel seront copiés les fichiers avec les droits d'accès correspondants.

```
useradd lufi
mkdir /home/lufi
chown -R lufi:lufi /home/lufi


mkdir /var/www/lufi
# chown -R www-data: /var/www/lufi
chown -R www-data:www-data /var/www/lufi
```

Téléchargez les fichiers de la dernière version sur le dépôt officiel (« Download zip » en bas à droite ou bien en ligne de commande avec git).
Copiez son contenu dans le dossier /var/www/html/lufi et attribuez les droits des fichiers à l'utilisateur lufi

```
cd /var/www
git clone https://framagit.org/fiat-tux/hat-softwares/lufi.git
chown lufi:lufi -R /var/www/lufi
```

Connectez-vous avec l'utilisateur lufi : su lufi -s /bin/bash et lancez la commande d'installation des dépendances depuis le dossier

```
su lufi -s /bin/bash
cd /var/www/lufi
carton install
exit
```

Maintenant que tout est prêt, modifiez le fichier de configuration de Lufi lufi.conf avec votre éditeur de texte préféré sur le modèle du fichier lufi.conf.template.
Par défaut le logiciel est configuré pour écouter sur le port 8080 de l'adresse 127.0.0.1 (localhost).

```
cp lufi.conf.template lufi.conf
vi lufi.conf
```

L’ensemble des paramètres sont facultatifs à l'exception du paramètre contact (pensez bien à le configurer et à le décommenter).
Si vous utilisez Lufi derrière un serveur web comme Nginx (comme dans ce tutoriel), pensez bien à décommenter le paramètre proxy => 1,.

A ce stade de l'installation, il n'est plus nécessaire de se connecter avec l'utilisateur spécifiquement créé lufi. Afin d'améliorer la sécurité, on empêchera à présent la possibilité de se connecter à celui-ci avec la commande.

```
usermod -s /bin/false lufi
```

Lufi en tant que service
À présent, le serveur tournera lorsque qu’on lancera cette commande :

```
carton exec hypnotoad script/lufi
```

Pour éviter de devoir relancer le serveur à la main à chaque redémarrage du serveur, on va donc lancer Lufi sous forme de service. Deux possibilités s'offrent alors :

3. Démarrage auto pour InitV :

Démarrage auto pour Systemd :

```
cp utilities/lufi.service /etc/systemd/system
```

Modifiez ce fichier pour qu'il corresponde à votre installation (User=lufi, WorkingDirectory=/var/www/lufi, ...)

```
vi /etc/systemd/system/lufi.service
```

Faites en sorte que Systemd le prenne en compte

```
systemctl daemon-reload
```

Et qu'il le lance à chaque démarrage

```
systemctl enable lufi.service
```

4. Paramétrage :

À ce stade, si tout s’est bien passé, lorsque vous exécutez la commande service lufi start, Lufi est pleinement fonctionnel. Vous n’avez qu’à vous rendre sur l’URL http://127.0.0.1:8080 pour pouvoir l’utiliser.
Nous allons maintenant configurer Lufi pour le rendre accessible depuis un nom de domaine avec Nginx (vous pouvez également utiliser Apache ou Varnish puisque seule la fonctionnalité de proxy inverse nous intéresse).

Nginx :

Installez le paquet :

```
apt-get install nginx
```

Créez le fichier de configuration de votre domaine /etc/nginx/sites-available/votre-nom-de-domaine pour y mettre ceci (en remplaçant « votre-nom-de-domaine ») et le port 8080 si vous l’avez changé dans la configuration de Lufi :

```
vi /etc/nginx/sites-available/telechargement.morbihan-energies.fr.conf
```

```
server { listen 80;

    server_name telechargement.morbihan-energies.fr;

    access_log /var/log/nginx/telechargement.success.log;
    error_log /var/log/nginx/telechargement.error.log;

    location / {
        # Add cache for static files
        if ($request_uri ~* ^/(img|css|font|js)/) {
            add_header Expires "Thu, 31 Dec 2037 23:55:55 GMT";
            add_header Cache-Control "public, max-age=315360000";
        }
        # HTTPS only header, improves security
        #add_header Strict-Transport-Security "max-age=15768000";

        # Adapt this to your configuration
        proxy_pass  http://127.0.0.1:8082;

        # Really important! Lufi uses WebSocket, it won't work without this
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
```

```
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        # If you want to log the remote port of the file senders, you'll need that
        proxy_set_header X-Remote-Port $remote_port;

        proxy_set_header X-Forwarded-Proto $scheme;

        # We expect the downstream servers to redirect to the right hostname, so don't do any rewrites here.
        proxy_redirect     off;
    }
}
```

Activez votre fichier :

```
ln -s /etc/nginx/sites-available/telechargement.morbihan-energies.fr.conf /etc/nginx/sites-enabled/telechargement.morbihan-energies.fr.conf
```

Enfin, relancez nginx :

```
/etc/init.d/nginx restart
```

B - Personnalisation :

1. Gestion des Logos :

Emplacement des logos : /var/www/lufi/themes/default/public/img
Liste des logos à configurer en 24x24 Pixels :

```
- favicon.png
- lufi-min.png
- lufi_favicon.png
```

2. Gestion des thèmes :

Pour personnaliser votre instance de Lufi, il faut lancer la commande carton :

```
exec script/lufi theme morbihan-energies
```

Ce qui créera un nouveau dossier "themes/votre-theme".
Il contiendra un fichier "Makefile", un fichier "lib/Lufi/I18N.pm" ainsi que des dossiers vides pour vous montrer le « squelette » d'un thème.
Ensuite il ne vous reste plus qu'à copier les fichiers du thème default dans votre dossier de thème et à les adapter à votre gré. Vous pouvez ainsi modifier les fichiers css, images, javascript (dossier public), les modèles de pages (dossier templates) et même la traduction en conservant un thème propre sur lequel basculer en cas de problème.

3. Modification de la traduction :

Le cas de la modification des traductions est un peu particulier :

Si vous souhaitez ajouter des chaînes de caractères traduites dans votre thème, ajoutez-les dans les fichiers du dossier "templates" en prenant exemple sur le thème par défaut (<%= l('Ceci est une chaîne traduite') %>) puis exécutez "make locales" depuis votre dossier de thème (themes/votre-theme, pas depuis le dossier d'installation de Lufi).
Il vous restera alors à ajouter la traduction dans les fichiers présents dans "themes/votre-theme/lib/Lufi/I18N/".
si vous souhaitez modifier une traduction, repérez la chaîne dans les fichiers de "themes/default/lib/Lufi/I18N/", puis recopiez les lignes commençant par msgid et msgstr qui s'y rapportent (ne copiez surtout pas les commentaires qui s'y rapportent). Enfin, modifiez la traduction (le contenu de msgstr) dans vos fichiers.
Modifiez ensuite le fichier "lufi.conf" pour préciser quel thème vous utilisez. Pour pouvoir personnaliser et observer vos modifications en direct, il vous faudra stopper temporairement le service service lufi stop et le démarrer avec la commande :

```
carton exec morbo script/lufi --listen=http://127.0.0.1:8080
```

C - lufi.conf

Contenu du fichier lufi.conf :
Emplacement :

```
# vim:set sw=4 ts=4 sts=4 ft=perl expandtab:
{
  ####################
```

```
# Hypnotoad settings
####################
# see http://mojolicio.us/perldoc/Mojo/Server/Hypnotoad for a full list of settings
hypnotoad => {
    # array of IP addresses and ports you want to listen to
    # you can specify a unix socket too, like 'http+unix://%2Ftmp%2Flufi.sock'
    listen => ['http://127.0.0.1:8082'],
    # if you use Lufi behind a reverse proxy like Nginx, you want to set proxy to 1
    # if you use Lufi directly, let it commented
    proxy  => 1,

    # Please read http://mojolicious.org/perldoc/Mojo/Server/Hypnotoad#workers
    # to adjust this to your server
    workers => 30,
    clients => 1,
},

# Put a way to contact you here and uncomment it
# You can put some HTML in it
# MANDATORY
contact       => '<a href="https://morbihan-energies.fr/contact/">Contact page</a>',

# Put an URL or an email address to receive file reports and uncomment it
# It's for make reporting illegal files easy for users
# MANDATORY
report => 'telechargement@morbihan-energies.fr',

# Array of random strings used to encrypt cookies
# optional, default is ['fdjsofjoihrei'], PLEASE, CHANGE IT
#secrets       => ['fdjsofjoihrei'],

# Name of the instance, displayed next to the logo
# optional, default is Lufi
instance_name => 'Telechargement.morbihan-energies.fr',

# Choose a theme. See the available themes in `themes` directory
# Optional, default is 'default'
#theme         => 'default',
```

```
    # Length of the random URL
    # optional, default is 8
    #length           => 8,

    # How many URLs will be provisioned in a batch ?
    # optional, default is 5
    #provis_step      => 5,

    # Max number of URLs to be provisioned
    # optional, default is 100
    #provisioning     => 100,

    # Length of the modify/delete token
    # optional, default is 32
    #token_length     => 32,

    # Max file size, in octets
    # You can write it 100*1024*1024
    # optional, no default
    #max_file_size    => 104857600,
    # JKT - Config 4Go
    max_file_size     => 4294967296,
    # If you want to have piwik statistics, provide a piwik image tracker
    # Only the image tracker is allowed, no javascript
    # optional, no default
    #piwik_img        => 'https://piwik.example.org/piwik.php?idsite=1&amp;rec=1',

    # Broadcast_message which will displayed on the index page
    # optional, no default
    #broadcast_message => 'Maintenance',

    # Default time limit for files
    # Valid values are 0, 1, 7, 30 and 365
    # optional, default is 0 (no limit)
    #default_delay    => 0,

    # Number of days after which the files will be deleted, even if they were uploaded with "no delay" (or value
superior to max_delay)
    # A warning message will be displayed on homepage
```

```
    # optional, default is 0 (no limit)
    max_delay         => 30,

    # Size thresholds: if you want to define max delays for different sizes of file
    # The keys are size in Bytes, you can't have 10*1000*10000 as key
    # If a file is smaller than the smallest configured size, it will have a expiration delay of max_delay (see above)
    # optional, default is using max_delay (see above) for all sizes
    #delay_for_size  => {
    #    10000000   => 90, # between 10MB and 50MB => max is 90 days, less than 10MB => max is max_delay
(see above)
    #    50000000   => 60, # between 50MB ans 1GB  => max is 60 days
    #    1000000000 => 2,  # more than 1GB         => max is 2 days
    #},

    # URL sub-directory in which you want Lufi to be accessible
    # example: you want to have Lufi under https://example.org/lufi/
    # => set prefix to '/lufi' or to '/lufi/', it doesn't matter
    # optional, defaut is /
    #prefix        => '/',

    # Array of authorized domains for API calls.
    # If you want to authorize everyone to use the API: ['*']
    # optional, no domains allowed by default
    #allowed_domains => ['http://1.example.com', 'http://2.example.com'],

    # String of the URL to be redirected to when accessing /logout
    # optional, default is no redirection after logging out
    #logout_custom => 'https://sso.example.com/logout?redirect_uri=https%3A%2F%2Fexample.com',

    # Define a path to the upload directory, where the uploaded files will be stored
    # You can define it relative to lufi directory or set an absolute path
    # Remember that it has to be in a directory writable by Lufi user
    # optional, default is 'files'
    #upload_dir => 'files',

    #!!!!!!!!!!!!!!!
    # EXPERIMENTAL !
    #!!!!!!!!!!!!!!!
    # You can store files on Swift object storage (https://en.wikipedia.org/wiki/OpenStack#Swift) instead of
```

```
filesystem
    # Please read https://metacpan.org/pod/Net::OpenStack::Swift#SYNOPSIS to know how to configure this setting
    # IMPORTANT: add a `container` key in it, to let Lufi know which container to use. This is not a regular Net::OpenStack::Swift setting, but Lufi need it.
    # EXPERIMENTAL: if the upload or download of files are stucked, reload Lufi and create a cron task to reload Lufi once a day
    # You can copy Lufi files to Swift object storage by launching the command `carton exec script/lufi copyFilesToSwift` (can take a long time)
    # optional, no default
    #swift => {
    #  auth_url    => 'https://auth-endpoint-url/v2.0',
    #  user        => 'userid',
    #  password    => 'password',
    #  tenant_name => 'project_id',
    #  container   => 'lufi'
    #},

    # Allow to add a password on files, asked before allowing to download files
    # optional, default is 0
    #allow_pwd_on_files => 0,

    # Force all files to be in "Burn after reading mode"
    # optional, default is 0
    #force_burn_after_reading => 0,

    # If set, the files' URLs will always use this domain
    # optional, no default
    #fixed_domain => 'example.org',

    # Abuse reasons
    # Set an integer in the abuse field of a file in the database and it will not be downloadable anymore
    # The reason will be displayed to the downloader, according to the reasons you will configure here.
    # optional, no default
    #abuse => {
    #   0 => 'Copyright infringment',
    #   1 => 'Illegal content',
    #},
```

```
###############
# Mail settings
###############

# Mail configuration
# See https://metacpan.org/pod/Mojolicious::Plugin::Mail#EXAMPLES
# optional, default to sendmail method with no arguments
mail => {
#    # Valid values are 'sendmail' and 'smtp'
   how => 'smtp',
   howargs => ['relais.v.sdem.fr']
},

# Email sender address
# optional, default to no-reply@lufi.io
mail_sender => 'hostmaster@morbihan-energies.fr',

#############
# DB settings
#############

# Choose what database you want to use
# Valid choices are sqlite, postgresql and mysql (all lowercase)
# optional, default is sqlite
#dbtype => 'sqlite',

# SQLite ONLY - only used if dbtype is set to sqlite
# Define a path to the SQLite database
# You can define it relative to lufi directory or set an absolute path
# Remember that it has to be in a directory writable by Lufi user
# optional, default is lufi.db
#db_path          => 'lufi.db',

# PostgreSQL ONLY - only used if dbtype is set to postgresql
# These are the credentials to access the PostgreSQL database
# mandatory if you choosed postgresql as dbtype
#pgdb => {
#    database => 'lufi',
#    host     => 'localhost',
```

```
#    # optional, default is 5432
#    #port     => 5432,
#    user     => 'DBUSER',
#    pwd      => 'DBPASSWORD',
#    # https://mojolicious.org/perldoc/Mojo/Pg#max_connections
#    # optional, default is 1
#    #max_connections => 1,
#},

# MySQL ONLY - only used if dbtype is set to mysql
# These are the credentials to access the MySQL database
# mandatory if you choosed mysql as dbtype
#mysqldb => {
#    database => 'lufi',
#    host     => 'localhost',
#    # optional, default is 3306
#    #port     => 3306,
#    user     => 'DBUSER',
#    pwd      => 'DBPASSWORD',
#    # https://metacpan.org/pod/Mojo::mysql#max_connections
#    # optional, default is 5 (set to 0 to disable persistent connections)
#    #max_connections => 5,
#},

################################################
# LDAP settings (authentication and features)
################################################

# Set `ldap` if you want that only authenticated users can upload files
# Please note that everybody can still download files
# optional, no default
#ldap => {
#    uri         => 'ldaps://ldap.example.org',               # server URI
#    user_tree   => 'ou=users,dc=example,dc=org',             # search base DN
#    bind_dn     => 'uid=ldap_user,ou=users,dc=example,dc=org', # search bind DN
#    bind_pwd    => 'secr3t',                                 # search bind password
#    user_attr   => 'uid',                                    # user attribute (uid, mail, sAMAccountName, etc.)
#    user_filter => '(!(uid=ldap_user))',                     # user filter (to exclude some users, etc.)
#    # optional start_tls configuration. See https://metacpan.org/pod/distribution/perl-
```

```
ldap/lib/Net/LDAP.pod#start_tls
    #    # don't set or uncomment if you don't want to configure it
    #    start_tls => {
    #       verify     => 'optional',
    #       clientcert => '/etc/ssl/certs/ca-bundle.pem'
    #    }
    #},

    # If you've set ldap above, the session will last `session_duration` seconds before
    # the user needs to reauthenticate
    # optional, default is 3600
    #session_duration => 3600,

    # If you use `ldap` for authentication, you can map some attributes from LDAP to be able to access them in
Lufi
    # Those attributes will be accessible with:
    #   $c->current_user->{lufi_attribute_name} in Lufi backend files (all that is in `lib` directory)
    #   <%= $self->current_user->{lufi_attribute_name} %> in templates files (in `themes` directory)
    #
    # Define the attributes like this: `lufi_attribute_name => 'LDAP_attribute_name'`
    # Note that you can’t use `username` as a Lufi attribute name: this name is reserved and will contain the
login of the user
    # optional, no default
    #ldap_map_attr => {
    #    displayname => 'cn',
    #    mail        => 'mail'
    #},

    # When using LDAP authentication, LDAP users can invite people (by mail) to use Lufi to send them files
without
    # being authenticated.
    # This is where you configure the behavior of the invitations.
    # You may need to fetch some attributes from LDAP to use some invitations settings. See `ldap_map_attr`
above.
    # optional, no default
    #invitations => {
    #   # The name of the key set in `ldap_map_attr` (above) that corresponds to the mail of the LDAP user
    #   # optional, default is `mail`
    #   mail_attr => 'mail',
```

```
    #   # The `From` header of invitation mail can be the mail of the LDAP user
    #   # Be sure to have a mail system that will correctly send the mail from your users! (DKIM, SPF...)
    #   # To enable this feature, set it to 1
    #   # optional, disabled by default
    #   send_invitation_with_ldap_user_mail => 1,
    #   # The user is able to set an expiration delay for the invitation.
    #   # This expiration delay can’t be more than this setting (in days).
    #   # optional, default is 30 days
    #   max_invitation_expiration_delay => 30,
    #   # Once the guest has submitted his files, he has an additional period of time to submit forgotten files.
    #   # You can set that additional period of time in minutes here.
    #   # To disable that feature, set it to 0 or less
    #   # optional, default is 10 minutes
    #   max_additional_period => 10,
    #   # Lufi follows privacy-by-design, so, by default, no files URLs (with the decode secret) are stored in database.
    #   # However, the concern is different for this case. Storing files URLs makes users able to retrieve the guests’ sent files
    #   # from their `invitations` page.
    #   # Set to 1 to store guests’ files URLs in database
    #   # optional, default is 0 (disabled)
    #   save_files_url_in_db => 0,
    #   # Users can resend the invitation to their guest. This does not extend the invitation’s expiration delay unless you
    #   # set this option to 1.
    #   # optional, default is 0 (disabled)
    #   extend_invitation_expiration_on_resend => 0,
    #},

    ############################
    # Htpasswd authentication
    ############################

    # Set `htpasswd` if you want to use an htpasswd file instead of ldap
    # See 'man htpasswd' to know how to create such file
    #htpasswd => 'lufi.passwd',

    ########################
    # HTTP Headers settings
```

```
    ######################

    # Content-Security-Policy header that will be sent by Lufi
    # Set to '' to disable CSP header
    # https://content-security-policy.com/ provides a good documentation about CSP.
    # https://report-uri.com/home/generate provides a tool to generate a CSP header.
    # optional, default is "base-uri 'self'; connect-src 'self' ws://YOUR_HOST; default-src 'none'; font-src 'self'; form-action 'self'; frame-ancestors 'none'; img-src 'self' blob:; media-src blob:; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'"
    #csp => "",

    # X-Frame-Options header that will be sent by Lufi
    # Valid values are: 'DENY', 'SAMEORIGIN', 'ALLOW-FROM https://example.com/'
    # Set to '' to disable X-Frame-Options header
    # See https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
    # Please note that this will add a "frame-ancestors" directive to the CSP header (see above) accordingly
    # to the chosen setting (See https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors)
    # optional, default is 'DENY'
    #x_frame_options => 'DENY',

    # X-Content-Type-Options that will be sent by Lufi
    # See https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
    # Set to '' to disable X-Content-Type-Options header
    # optional, default is 'nosniff'
    #x_content_type_options => 'nosniff',

    # X-XSS-Protection that will be sent by Lufi
    # See https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection
    # Set to '' to disable X-XSS-Protection header
    # optional, default is '1; mode=block'
    #x_xss_protection => '1; mode=block',

    ##########################
    # Lufi cron jobs settings
    ##########################

    # Expired files will be kept for 2 additional days after the expiration time has passed!
    # The reasoning behind this is to allow downloads to complete and avoid deleting them while
```

```
    # they are still being tranfered.

    # Number of days senders' IP addresses are kept in database
    # After that delay, they will be deleted from database (used with script/lufi cron cleanbdd)
    # optional, default is 365
    #keep_ip_during    => 365,

    # Max size of the files directory, in octets
    # Used by script/lufi cron watch to trigger an action
    # optional, no default
    #max_total_size    => 10*1024*1024*1024,

    # Default action when files directory is over max_total_size (used with script/lufi cron watch)
    # Valid values are 'warn', 'stop-upload' and 'delete'
    # Please, see README.md
    # optional, default is 'warn'
    #policy_when_full  => 'warn',

    # Files which are not viewed since delete_no_longer_viewed_files days will be deleted by the cron cleanfiles
task
    # If delete_no_longer_viewed_files is not set, the no longer viewed files will NOT be deleted
    # optional, no default
    #delete_no_longer_viewed_files => 90,
};
```

E - ProFTPd

Serveur FTP :

Un serveur FTP est configuré pour répondre à plusieurs besoins interne nottamment par le servie Energies.
Nom de connexion : ftp.morbihan-energies.fr
Port : 21