

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 :

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 LUF1 :

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/flat-tux/hat-sofwareas/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 « `vousre-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 :

```
In -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 ['fdjsfjoihrei'], PLEASE, CHANGE IT
#secrets    => ['fdjsfjoihrei'],

# 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&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 and 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, default 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 infringement',
# 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

Revision #1

Created 8 February 2024 08:11:23 by Jérôme

Updated 3 March 2026 08:05:50 by Jérôme