naos
Nouvelle-Aquitaine
Open Source

Système d'Information Libre

version 1 | 10/2022

RÉGION
Nouvelle-Aquitaine

SOMMAIRE

Introduction....4

Licence....4

1. BIGBLUEBUTTON....5

A. Introduction....6
B. Installation....7
C. Branding NAOS ou autre....8
D. Mise à jour....8
E. Sauvegarde....9
F. Restauration....9
G. Migration....10

2. NEXTCLOUD....11

A. Introduction....12
B. Installation....12
C. Mise à jour....22
D. Sauvegarde....23
E. Restauration....23
F. Migration....24

3. GITLAB....25

A. Introduction....26
B. Installation....26
C. Mise à jour....27
D. Sauvegarde....28
E. Restauration / Migration....28

4. MATTERMOST....30

A. Introduction....31
B. Installation....31
C. Mise à jour....35
D. Sauvegarde / Restauration....35

5. NAOS HUB....36

A. Introduction....37

B. Installation....37
C. Configuration....38
D. Branding....39
E. Mise à jour....39
F. Sauvegarde....40

6. Site Wordpress....41

A. Introduction....42
B. Installation et configuration d'un site Wordpress....42
C. Mise à jour....43
D. Sauvegarde (UpdraftPlus)....43
E. Restauration (UpdraftPlus)....44
F. Migration (UpdraftPlus)....44

7. UpdraftCentral....46

A. Introduction....47
B. Installation et configuration du site de Supervision....47
C. Ajouter un site à superviser....48
D. Mise à jour du site de supervision....49

8. YOURLS....50

A. Introduction....51
B. Installation et configuration....51
C. Mise à jour....53
D. Sauvegarde....53
E. Restauration....53
F. Utilisation....54

9. SIGGEN....55

A. Introduction....56
B. Installation....56
C. Utilisation....57

Introduction

Ce document regroupe les informations permettant de mettre en œuvre des solutions libres pour les TPE-PME. Il constitue une base de connaissances techniques qui est amenée à évoluer à partir de contributions sur la plateforme GitLab :

https://forge.naos-cluster.tech/naos/sil

Ainsi, vous trouverez sur ce document les informations concernant les logiciels suivants :

- **BigBlueButton** : Système de visioconférence développé pour la formation à distance.
- **Nextcloud** : Logiciel libre de site d'hébergement de fichiers et une plateforme de collaboration.
- **Gitlab** : Logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.
- **Mattermost** : Logiciel et un service de messagerie instantanée libre auto-hébergeable. Il est conçu comme un chat interne pour les organisations et les entreprises.
- **Humhub** : Logiciel opensource permettant de mettre en place un réseau social.
- **Wordpress** : Système de gestion de contenu gratuit, libre et open-source.
- **UpdraftCentral** :Plugin Wordpress permettant de gérer plusieurs instances Wordpress (mise à jour, sauvegarde).
- **YOURLS** : Une application de raccourcissement d'URL.
- **SIGGEN** : Une application PHP permettant la génération de signature.

Pour l'hébergement de ces services, nous faisons notamment appel à **DigitalForest** (hébergement web écologique en France).

Licence

SIL
Copyright (C) 2022 NAOS.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation ; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

1

BIGBLUEBUTTON

Gestion d’une instance BigBlueButton

A. Introduction

BigBlueButton est un système de visioconférence développé pour la formation à distance. Il permet le partage de la voix et de l'image vidéo, de présentations avec ou sans tableau blanc, l'utilisation d'outils de clavardage publics et privés, le partage d'écran, la voix sur IP, des sondages en ligne, ainsi que l'utilisation de documents bureautiques.

Notre infrastructure de visioconférence est composé d'un serveur hébergeant BigBlueButton et Greenlight (le serveur BigBlueButton) et d'un serveur TURN. Les deux serveurs sont sous distribution Ubuntu 20.04.

Dans certaines configurations réseau, un pare-feu bloquera la réception et l'envoi sur certains ports. Il est possible que l'utilisateur ne puisse communiquer qu'avec le port 80 ou 443. Or, BigBlueButton utilise d'autres ports pour la VoIP et la vidéo, ce qui posera problème. Pour palier ce problème, on met en place un serveur TURN qui recevra tous les paquets provenant de l'utilisateur (VoIP ou vidéo) sur le port 443 et qui renverra le flux vers le serveur BigBlueButton (sur les bons ports).

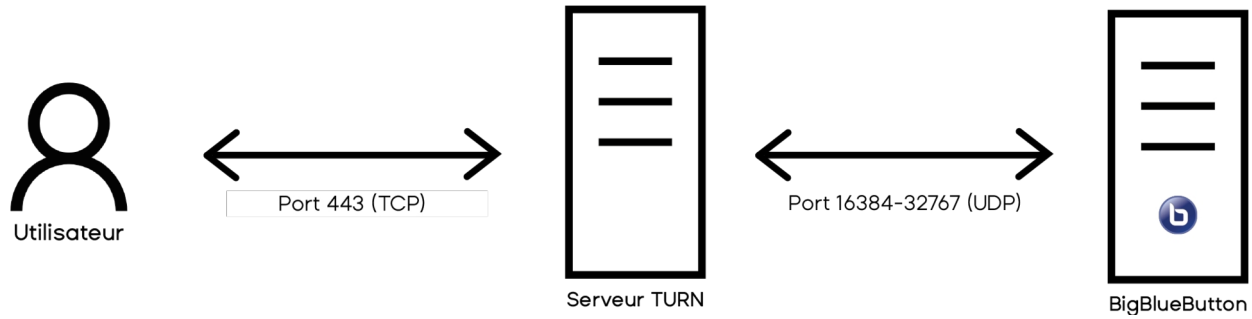


Schéma du fonctionnement de BigBlueButton et TURN

Notre instance **BigBlueButton** est hébergée sur un serveur dédié géré par **Ikoula** et notre instance **Turn** sur un serveur privé virtuel hébergé par **MVPS**.

B. Installation

La mise en place d'une instance BigBlueButton se fait à partir du script bash **bbb-install-2.5.sh** (https://github.com/bigbluebutton/bbb-install). Elle nécessite l'installation d'une distribution Ubuntu 20.04.

Lancer l'installation de BigBlueButton (sur le serveur BigBlueButton) :

```
wget -qO- https://ubuntu.bigbluebutton.org/bbb-install-2.5.sh | bash -s -- -v focal-250 -s visio.naos-cluster.tech -e sysadmin@naos-cluster.tech -w -g
```

-v : La version de BigBlueButton. Dans notre cas, on installe la version 2.5 pour Ubuntu Focal

-s : Le domaine dédié au serveur de visioconférence

-e : Adresse mail qui sera fournie à Let's Encrypt lors de la génération du certificat

-w : Installation de UFW pour faciliter la gestion du pare-feu

-g : Installation de Greenlight

Lorsque l'installation de BigBlueButton se termine, lancez l'installation du serveur TURN (sur le serveur TURN) :

```
wget -qO- https://ubuntu.bigbluebutton.org/bbb-install-2.5.sh | bash -s -- -c turn.naos-cluster.tech:1234abcd -e sysadmin@naos-cluster.tech
```

-c : Spécifier l'adresse et le secret partagé du serveur TURN. Ce secret sera utilisé par le serveur BigBlueButton lors de la liaison avec le serveur TURN.

-e : Adresse mail qui sera fournie à Let's Encrypt lors de la génération du certificat

Et terminez par la liaison du serveur BigBlueButton au serveur TURN (à lancer sur le serveur BigBlueButton) :

```
wget -qO- https://ubuntu.bigbluebutton.org/bbb-install-2.5.sh | bash -s -- -v focal-250 -s visio.naos-cluster.tech -e sysadmin@naos-cluster.tech -c turn.naos-cluster.tech:1234abcd
```

Comme pour l'installation de BigBlueButton, on doit fournir la version, le domaine dédié au serveur, l'adresse électronique qui sera fournie à Let's Encrypt, le nom de domaine permettant d'atteindre le serveur TURN et le secret partagé.

Lorsque toutes les installations seront terminées, on devrait avoir un support fonctionnel.

C. Branding NAOS ou autre...

Après avoir installé notre instance BigBlueButton, nous l'avons adaptée à la charte graphique de NAOS.

Pour éviter d'avoir à ré exécuter à chaque mise à jour les commandes permettant de charter notre instance, un script bash a été mis en place : https://forge.naos-cluster.tech/naos-internal/bbb-rebrand.

Il suffit de cloner le dépôt et de lancer le script **rebrand.sh**. Il remplacera les différents éléments graphiques par ceux de NAOS (ou de votre structure).

D. Mise à jour

1. BigBlueButton (et TURN)

Pour mettre à jour **BigBlueButton**, il suffit de relancer le script **bbb-install.sh** (celui adapté à la nouvelle version) et ce dernier mettra automatiquement à jour l'instance BigBlueButton.

Exemple : Si le serveur BigBlueButton est en 2.4, il suffit de lancer le script d'installation en 2.5 avec les mêmes paramètres utilisés à l'installation :

```
wget -qO- https://ubuntu.bigbluebutton.org/bbb-install-2.5.sh | bash -s -- -v focal-250 -s visio.naos-cluster.tech -e sysadmin@naos-cluster.tech -w -g
```

Ne pas oublier de mettre à jour le serveur TURN en même temps :

```
wget -qO- https://ubuntu.bigbluebutton.org/bbb-install-2.5.sh | bash -s -- -c turn.naos-cluster.tech:1234abcd -e sysadmin@naos-cluster.tech
```

2. Greenlight

Greenlight est une application conteneurisée. Pour la mettre à jour, il suffit d'arrêter les conteneurs, télécharger la nouvelle version des images de Greenlight et relancer les conteneurs :

```
cd ~/greenlight
docker-compose down
docker-compose pull
docker-compose up -d
```

La partie Greenlight est maintenant à jour.

E. Sauvegarde

Nous avons mis en place un script permettant de sauvegarder la base de données des utilisateurs Greenlight. Ce script peut aussi être utilisé pour sauvegarder les enregistrements générés par une instance BigBlueButton.

1. Script de sauvegarde

Lien vers le script : https://forge.naos-cluster.tech/naos-internal/bbb-backup

Pour réaliser la sauvegarde sur le serveur distant hébergeant l'instance BigBlueButton, ce script se connecte en SSH.

Pré-requis

ATTENTION : Le script se connecte en SSH avec la méthode **publickey**, il est donc nécessaire de configurer une connexion SSH avec la méthode d'authentification par clé publique avant de l'utiliser.

Utilisation du script

Sauvegarde de la base de données de Greenlight et des enregistrements :

```
ssh user@host 'bash -s' < bbb-backup.sh --greenlight --records 3> backup_name 2>
/dev/null
scp user@host:$(cat backup_name) /path/to/backup_folder
rm backup_name
```

F. Restauration

Pour pouvoir réaliser la restauration, il faut que vous ayez utilisé le script de sauvegarde fourni dans la partie **Script de sauvegarde**. L'archive de la sauvegarde réalisé par le script contient la base de données de Greenlight et les enregistrements de l'instance BigBlueButton.

Avant de lancer la restauration, nous devons arrêter les conteneurs de Greenlight et renommer le dossier **greenlight** en **greenlight_old**. Connectez-vous en SSH et suivez les commandes ci-dessous :

```
cd greenlight
docker-compose down
cd ..
mv greenlight greenlight_old
```

Pour restaurer une sauvegarde, on l'envoie sur notre serveur où est hébergé notre instance BigBlueButton, on décompresse l'archive et on place les dossiers au bon endroit.

```
unzip backup_20221407.zip
rm backup_20221407.zip
mv backup_20221407/greenlight ~
cp -r backup_20221407/records/* /var/bigbluebutton/
rm -r backup_20221407
```

Puis on redémarre les conteneurs de Greenlight

```
ssh user@host 'cd greenlight; docker-compose up -d'
```

Lorsque les conteneurs sont lancés, vérifiez que tout a bien été restauré.

G. Migration

Pour réaliser une migration d'une instance BigBlueButton, commencez par réaliser la procédure de sauvegarde sur votre serveur hébergeant votre BigBlueButton.

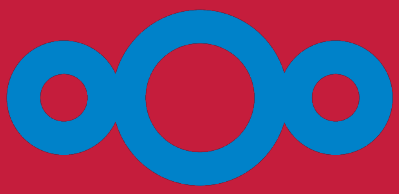
Monter une nouvelle instance BigBlueButton sur un nouveau serveur en suivant la procédure d'installation.

Et, terminez la migration en suivant la procédure de restauration à partir de la sauvegarde réalisée sur l'ancien serveur hébergeant votre instance BigBlueButton.

2

NEXTCLOUD

Gestion d’une instance Nextcloud



A. Introduction

Nextcloud est un logiciel libre de site d'hébergement de fichiers et une plateforme de collaboration. À l'origine accessible via WebDAV, n'importe quel navigateur web, ou des clients spécialisés, son architecture ouverte a permis de voir ses fonctionnalités s'étendre depuis ses origines.

Nos instances **Nextcloud** (*cloud.naos-cluster.tech* et *cloud.comena.net*) sont hébergées sur des serveurs dédié gérés par **Ikoula**.

B. Installation

Dans cette procédure d'installation, nous allons mettre en place une instance Nextcloud sur une machine possédant la distribution Debian 11 Server.

Le sous-domaine qui permettra d'accéder au cloud est **cloud.naos-cluster.tech**. Modifiez-le lors de la configuration d'Apache et la génération du certificat Let's Encrypt pour l'adapter à votre infrastructure.

Pour le bon fonctionnement de l'instance Nextcloud, il sera nécessaire d'installer Apache2, MariaDB et PHP 8.0 sur la machine.

1. Installation des pré-requis

Installation des paquets nécessaire pour la configuration de dépôts

```
sudo apt install -y ca-certificates apt-transport-https software-properties-common wget curl lsb-release unzip
```

Configuration des dépôts de PHP 8.0

```
curl -sSL https://packages.sury.org/php/README.txt | sudo bash -x
```

Recherche des mises à jour disponibles

```
sudo apt update
```

2. Mise en place des paquets requis pour le bon fonctionnement de Nextcloud

Installation d'Apache2, MariaDB, PHP 8.0 et les extensions PHP requises pour le bon fonctionnement de Nextcloud

```
apt install -y apache2 mariadb-server php8.0 libapache2-mod-php8.0 imagemagick
apt install -y php8.0-{curl,gd,mbstring,xml,zip,mysql,bz2,intl}
apt install -y php8.0-{apcu,imagick,cli,common,gmp,bcmath}
```

Configuration d'Apache2

Création du fichier de configuration Apache2 nommé nextcloud.conf dédié à Nextcloud. Ce fichier est à créer dans le dossier **/etc/apache2/sites-available/**, voici son contenu :

```
<VirtualHost *:80>
     ServerAdmin contact@naos-cluster.tech
     ServerName cloud.naos-cluster.tech

     # Redirection forcée vers HTTPS
     Redirect permanent / https://cloud.naos-cluster.tech/
</VirtualHost>

<VirtualHost *:443>
     ServerAdmin contact@naos-cluster.tech
     DocumentRoot /var/www/nextcloud
     ServerName cloud.naos-cluster.tech

     Header add Strict-Transport-Security: "max-age=15768000;includeSubdomains"

     <Directory /var/www/nextcloud/>
          Options FollowSymlinks
          AllowOverride All
          Require all granted
     </Directory>

     <Directory /var/www/nextcloud/>
            RewriteEngine on
            RewriteBase /
            RewriteCond %{REQUEST_FILENAME} !-f
            RewriteRule ^(.*) index.php [PT,L]
    </Directory>
</VirtualHost>
```

On peut remarquer que le fichier est pré configuré pour supporter le protocole HTTPS.

Activation de la configuration Apache2 pour Nextcloud :

```
a2ensite nextcloud
```

Activation des différents modules PHP requis pour Nextcloud :

```
a2enmod rewrite
a2enmod headers
a2enmod env
a2enmod dir
a2enmod ssl
```

Installation d'un certificat Let's Encrypt

L'installation d'un certificat Let's Encrypt nous permettra de sécuriser notre site internet.

Installation de certbot et du module Apache de certbot :

```
apt install certbot python3-certbot-apache
```

(plus d'infos)

Génération du certificat Let's Encrypt :

```
certbot --apache -d cloud.naos-cluster.tech
```

(plus d'infos)

Redémarrer Apache2 :

```
systemctl restart apache2
```

Configuration de MariaDB

NOTE : Avant de réaliser l'étape suivante, prévoir une chaîne de caractères. Cette chaîne de caractères sera utilisée en tant que mot de passe pour l'utilisateur ROOT de MYSQL. Voici le script que nous utilisons pour générer un mot de passe :

https://forge.naos-cluster.tech/-/snippets/5

Avant de créer la base de données, nous allons réaliser une installation sécurisée de MYSQL avec la commande suivante :

```
sudo mysql_secure_installation
```

Lorsque le mot de passe actuel de ROOT est demandé, on ne rentre aucun mot de passe. Toutes les valeurs doivent être laissées par défaut (lettre en Majuscule).

Création de la base de données

Pour que Nextcloud fonctionne, il lui faut une base de données.

Tout d'abord, on se connecte à mysql avec la commande suivante :

```
sudo mysql
```

Lorsque la connexion à mysql est réussie, on poursuit par la création de la base de données nommée **nextcloud**, puis celle de l'utilisateur nommé **nextcloud** qui permettra d'y accéder, et on termine par le rechargement de la tables des privilèges :

```
# Création de la base de données
CREATE DATABASE nextcloud;

# Création de l'utilisateur
GRANT ALL PRIVILEGES on nextcloud.* to 'nextcloud'@'localhost' identified by '<mot de
passe>';

# Rechargement de la tables des privilèges
FLUSH PRIVILEGES;
```

Configuration de PHP

Pour que Nextcloud fonctionne correctement, il est nécessaire d'augmenter la mémoire PHP à la valeur recommandée (512 Mo). Pour régler ce problème, il suffit de modifier la ligne suivante :

```
memory_limit = 128M
```

Par

```
memory_limit = 512M
```

Dans le fichier de configuration de PHP (**/etc/php/8.0/apache2/php.ini**).

Nous allons aussi mettre en place OPcache

Editez le fichier **/etc/php/8.0/apache2/php.ini**, dé commentez (ou ajoutez) les lignes suivantes et modifiez leurs valeurs comme ci-dessous :

```
opcache.enable=1
opcache.enable_cli=1
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=10000
opcache.memory_consumption=128
opcache.save_comments=1
opcache.revalidate_freq=1
```

3. Nextcloud

Installation

On récupère l'archive contenant les fichiers de Nextcloud et on les déplace dans le dossier **/var/www/ :**

```
wget https://download.nextcloud.com/server/releases/latest.zip
unzip latest.zip
rm latest.zip
mv nextcloud /var/www/
```

On change le propriétaire du dossier /var/www/nextcloud pour qu'Apache n'est pas de problèmes lors de l'accès aux fichiers de nextcloud :

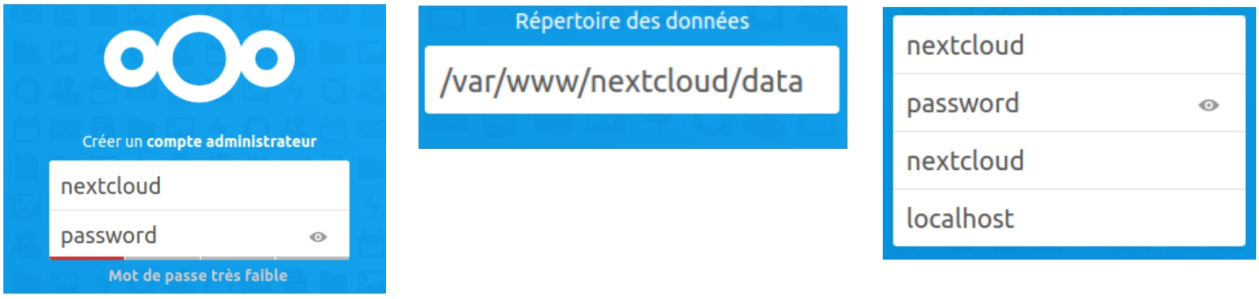
```
chown -R www-data. /var/www/nextcloud
```

Si tout s'est bien déroulé, le cloud est disponible à l'adresse **https://cloud.naos-cluster.tech/**.

Configuration

Lorsque qu'on accès au cloud, une page avec plusieurs champs est chargée. Cette page nous permettra de configurer l'utilisateur par défaut de notre instance Nextcloud, le chemin vers le dossier où seront stockées les fichiers placés sur cloud et la base de données. Pour le chemin vers les ces fichiers, on utilisera celui par défaut et on appellera notre utilisateur nextcloud **admin** et son mot de passe sera **password**. Les informations permettant d'accéder à la base de données on était définies dans la partie **Création de la base de données**.

Nous devrions donc avoir l'interface suivante :



Il ne nous reste plus qu'à lancer l'installation. Pour l'étape d'après, nous refusons l'installation des plugins recommandés.

Après ça, nous arrivons sur le tableau de bord de notre Nextcloud qui maintenant installé. Avant de faire quoi que ce soit, nous allons vérifier si nous avons des erreurs concernant la configuration.

Correction des erreurs concernant la configuration

Dans la partie **Paramètres > Vue d'ensemble**, il est possible que Nextcloud nous disent que nous avons des erreurs de configuration. Avant toutes choses, nous allons les régler.

Serveur de messagerie

Nous allons commencer par définir un serveur de mail pour que notre instance Nextcloud puisse envoyer des mails. Cette configuration est à mettre en place dans la partie :

Paramètres > Paramètres de base

Dans cette partie, nous allons pouvoir définir un serveur SMTP. Lorsque la configuration du serveur SMTP est terminé, nous réalisons un test avec le bouton **Envoyer un e-mail**. Si la configuration est correcte, un message de validation s'affiche et le message d'erreur ne s'affiche plus dans la partie **Paramètres > Vue d'ensemble**.

Pour notre part, nous utilisons les serveurs mails d'O2switch.

Préfixe de région par défaut

Pour régler cette erreur, il suffit de modifier le fichier de configuration de Nextcloud et d'y ajouter l'attribut **default_phone_region** avec comme valeur **fr**.

Votre fichier de configuration devrait ressembler à ça :

```
<?php
$CONFIG = array (
  'instanceid' => '****',
  'passwordsalt' => '****',
  'secret' => '****',
  'trusted_domains' =>
  array (
    0 => 'cloud.naos-cluster.tech',
  ),
  'datadirectory' => '/var/www/nextcloud/data',
  'dbtype' => 'mysql',
  'version' => '24.0.2.1',
  'overwrite.cli.url' => 'https://cloud.naos-cluster.tech',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'nextcloud',
  'dbpassword' => 'password',
  'installed' => true,

  # On ajoute l'attribut default_phone_region
  'default_phone_region' => 'fr',
);
```

Configuration du cache mémoire

Éditer le fichier de configuration de Nextcloud (**/var/www/nextcloud/config/config.php**) et ajouter la ligne suivante :

```
'memcache.local' => '\\OC\\Memcache\\APCu',
```

Votre fichier de configuration devrait ressembler à ça :

```
<?php
$CONFIG = array (
  'instanceid' => '****',
  'passwordsalt' => '****',
  'secret' => '****',
  'trusted_domains' =>
```

```
  array (
    0 => 'cloud.naos-cluster.tech',
  ),
  'datadirectory' => '/var/www/nextcloud/data',
  'dbtype' => 'mysql',
  'version' => '24.0.2.1',
  'overwrite.cli.url' => 'https://cloud.naos-cluster.tech',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'nextcloud',
  'dbpassword' => 'password',
  'installed' => true,
  'default_phone_region' => 'fr',

  # Ajout de l'attribut memcache.local
  'memcache.local' => '\\OC\\Memcache\\APCu',
);
```

Plugins installés

Voici les plugins supplémentaires installés sur notre instance Nextcloud :

Group folders – Créer un dossier pour un groupe
ONLYOFFICE – Editer les documents directement sur l'interface Web de Nextcloud.
Deck - Kanban
Calendar (relié à BlueMind) – Gérer son calendrier et avoir des calendriers en commun
Custom menu – Customiser le menu de Nextcloud

4. Mise en place d'un serveur de document OnlyOffice

Nous allons utiliser la puissance du serveur de document OnlyOffice pour l'édition de document sur notre instance Nextcloud.

Installation de Onlyoffice Document Server et PostgreSQL

Installation et configuration de PostgreSQL

```
apt install postgresql
```

Création de la base de données et de l'utilisateur qui permettra à Onlyoffice de communiquer avec la base de données

ATTENTION : l'utilisateur postgresql doit impérativement s'appeler **onlyoffice**. Notre mot de passe sera **password**

```
sudo -i -u postgres psql -c "CREATE DATABASE onlyoffice;"
sudo -i -u postgres psql -c "CREATE USER onlyoffice WITH password 'password';"
sudo -i -u postgres psql -c "GRANT ALL privileges ON DATABASE onlyoffice TO onlyoffice;"
```

Ajout de la clé GPG :

```
mkdir -p ~/.gnupg
chmod 700 ~/.gnupg
gpg --no-default-keyring --keyring gnupg-ring:/tmp/onlyoffice.gpg --keyserver
hkp://keyserver.ubuntu.com:80 --recv-keys CB2DE8E5
chmod 644 /tmp/onlyoffice.gpg
sudo chown root:root /tmp/onlyoffice.gpg
sudo mv /tmp/onlyoffice.gpg /etc/apt/trusted.gpg.d/
```

Ajout des dépôts de OnlyOffice :

```
echo "deb https://download.onlyoffice.com/repo/debian squeeze main" | sudo tee
/etc/apt/sources.list.d/onlyoffice.list
```

Recherche des mise à jour disponibles :

```
apt update
```

Installation de **mscorefonts :**

```
apt install ttf-mscorefonts-installer
```

Installation de OnlyOffice :

```
apt install onlyoffice-documentserver
```

NOTE : Lors de l'installation, le mot de passe pour l'utilisateur PostgreSQL **onlyoffice** vous est demandé.

Configuration du JWT

Pour la configuration du JWT, on commence par activer l'utilisation du token.

Nous éditons le fichier **/etc/onlyoffice/documentserver/local.json** et modifions les valeurs suivantes en **true** :

- services.CoAuthoring.token.enable.request.inbox
- services.CoAuthoring.token.enable.request.outbox
- services.CoAuthoring.token.enable.browser

Ensuite, nous générons une chaîne de caractères aléatoire qui servira de secret pour notre JWT. Ce secret doit être placé dans le même fichier, dans la partie **services.CoAuthoring.secret**. Le secret doit être le même pour **inbox**, **outbox** et **session**.

Votre fichier devrait ressembler à ça :

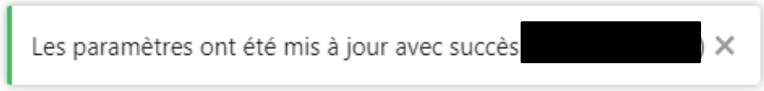
```
{
  "services": {
    "CoAuthoring": {
      "sql": {
        "type": "postgres",
        "dbHost": "localhost",
        "dbPort": "5432",
        "dbName": "onlyoffice",
        "dbUser": "onlyoffice",
        "dbPass": "password"
      },
      "token": {
        "enable": {
          "request": {
            "inbox": true,
            "outbox": true
          },
          "browser": true
        },
        "inbox": {
          "header": "Authorization"
        },
        "outbox": {
          "header": "Authorization"
        }
      },
      "secret": {
        "inbox": {
          "string": "<secret>"
        },
        "outbox": {
          "string": "<secret>"
        },
        "session": {
          "string": "<secret>"
        }
      }
    }
  },
  "rabbitmq": {
    "url": "amqp://guest:guest@localhost"
  }
}
```

Pour terminer la configuration du JWT, nous redémarrons tous les services :

```
supervisorctl restart all
```

Liaison entre Nextcloud à Onlyoffice

Pour lier votre instance Onlyoffice à votre Nextcloud, allez dans la partie **Paramètres > Administration > ONLYOFFICE**, renseignez l'adresse de votre serveur Onlyoffice et le secret que nous avons défini. Terminez en cliquant sur **Enregistrer**. si tout se passe bien, le message suivant est affiché :



Installation de nouvelles polices sur le serveur de document OnlyOffice

Récupérez les fichiers des polices et placez dans le dossier **/usr/share/fonts**
Ensuite, rechargez le cache des fonts du système et celui du serveur de document avec les commandes suivantes :

```
sudo fc-cache -fv
sudo documentserver-generate-allfonts.sh
```

5. Configuration du client Nextcloud

Il est possible d'installer un client Nextcloud. Ce dernier permet la synchronisation des fichiers présents dans le cloud sur votre machine.

Installation

Sur le lien ci-dessous, plusieurs fichiers d'installation sont disponibles :

https://nextcloud.com/install/#install-clients

Nous n'allons aborder que l'installation du client sous Linux.

Lorsque vous arrivez sur la page web de Nextcloud, dépliez **DOWNLOAD FOR DESKTOP** et téléchargez **Linux AppImage**.

Il ne reste plus qu'à autoriser l'exécution du fichier et à double-cliquer dessus.

Pour autoriser l'exécution du fichier, faites un clic droit sur celui-ci et aller dans Propriétés > Permissions et cliquez sur **Autoriser l'exécution du fichier comme un programme**.

Il est possible que rien ne se passe lorsque vous double-cliquez sur le fichier. Dans ce cas, lancez un terminal, déplacez-vous dans le dossier où se trouve le fichier AppImage, et lancez le fichier :

```
cd ~/Téléchargements
```

```
./Nextcloud-3.6.0-x86_64.AppImage
```

Une erreur sera affichée. En général, l'erreur provient du fait que, pour lancer un fichier AppImage, **FUSE** doit-être installé. Installez le avec les commandes suivantes :

```
sudo apt update
sudo apt install fuse libfuse2
```

Lorsque l'installation de **FUSE** est terminée, relancez le fichier AppImage. Vous devriez voir apparaître une page vous demandant de vous authentifier ou bien de vous enregistrer avec un fournisseur. Cliquez sur **Log in** et renseignez l'adresse de votre instance Nextcloud.

Le programme ouvre votre navigateur sur une page vous demandant de vous connecter, faites-le. Cette étape permettra de lier votre client Nextcloud à votre instance Nextcloud préalablement mise en place.

Lorsque vous vous êtes connecté sur votre navigateur, retournez sur votre client pour choisir les dossiers à synchroniser et le dossier dans le quel sera stocké les fichiers synchronisés. Terminez en cliquant sur **Connexion**.

Le programme synchronise vos fichiers qui seront disponibles dans le dossier que vous avez défini.

C. Mise à jour

Il y a deux façons de mettre à jour une instance Nextcloud : depuis l'interface Web et en lignes de commande.

Nextcloud possède un système de mise à jour développé qui peut être exécuté depuis l'interface Web de Nextcloud. Pour pouvoir l'utiliser, il faut se rendre sur la page **Paramètres > Vue d'ensemble** dans la partie **Version**. Si une mise à jour est disponible, un bouton permettant de la faire sera affiché.

Il est possible qu'aucun bouton ne soit affiché mais que votre Nextcloud ne soit pas à jour (en général, cela se produit lors des passages des grosses versions). Il faudra donc exécuter le script de mise à jour se trouver dans le dossier /var/www/nextcloud/updater :

```
sudo -u www-data php /var/www/nextcloud/updater/updater.phar
```

Attention, il est impératif d'exécuter le script avec l'utilisateur **www-data** pour éviter d'avoir, par la suite, des problèmes de permission.

D. Sauvegarde

Contrairement à d'autres applications web, Nextcloud n'embarque pas de système de sauvegarde intégré. Néanmoins, sa sauvegarde se fait de façon relativement simple.

Avant de commencer la sauvegarde des différents éléments, activez le mode maintenance de Nextcloud :

```
sudo -u www-data php occ maintenance:mode --on
```

Créez le dossier où sera stockée la sauvegarde et copiez les fichiers de notre instance Nextcloud à l'intérieur :

```
cd /var/www/
mkdir nextcloud_bkp
cp -r nextcloud nextcloud_bkp/nextcloud_bkp$(date +"%Y%m%d")
```

Faites une sauvegarde la base de données et stockez le fichier dans le dossier de sauvegarde créé :

```
mysqldump nextcloud >  nextcloud_bkp/nextcloud_bkp$(date +%Y%m%d).sql
```

Et terminez par la compression du dossier :

```
tar zcvf nextcloud_bkp$(date +%Y%m%d).tar.gz nextcloud_bkp/
```

Si vous possédez un ou plusieurs emplacement de stockage externe, n'oubliez pas de les sauvegarder également.

E. Restauration

Pour la restauration, il suffit de réaliser le processus inverse de la sauvegarde.

Avant toutes choses, nous allons récupérer notre archive contenant la sauvegarde de notre instance Nextcloud à un moment T. Dans notre cas, l'archive s'appellera **nextcloud_bkp20220704.tar.gz**.

Décompressez l'archive et placez le dossier de notre instance dans le dossier **/var/www** en lui attribuant le bon propriétaire.

```
tar xf nextcloud_bkp20220704.tar.gz
cd nextcloud

# Si un dossier nextcloud est déjà créé dans le dossier /var/www/, on le supprime
rm -r /var/www/nextcloud

mv nextcloud_bkp20220704 /var/www/nextcloud
```

```
chown www-data. -R /var/www/nextcloud
```

Pour la restauration de la base de données, le processus est un peu plus compliqué que lors de la sauvegarde. Il faut, avant la restauration de la base de données, en recréer une ainsi que son utilisateur :

```
# Connexion à mysql
mysql

# Création de la base de données
CREATE DATABASE nextcloud;

# Création de l'utilisateur utilisé par Nextcloud
GRANT ALL PRIVILEGES on nextcloud.* to 'nextcloud'@'localhost' identified by 'password';

# Rechargement de la table des permissions
FLUSH PRIVILEGES;
```

Lorsque la base de données et l'utilisateur sont recréés, importez la sauvegarde de la base de données dans la nouvelle base de données :

```
mysql nextcloud < nextcloud_bkp20220704.sql
```

Une fois la procédure terminée, vérifiez que la restauration n'ait rien endommagé sur l'instance (Vérification des plugins, vérification des fichiers, etc ...)

F. Migration

Maintenant que nous avons réalisé les procédures d'installation, de sauvegarde et de restauration, il est assez simple d'imaginer la procédure de migration.

Commencez par réaliser une sauvegarde de l'instance à migrer. Ensuite, réalisez la procédure d'installation sur la nouvelle machine qui accueillera notre instance Nextcloud. La seule différence est que nous allons remplacer la partie installation et configuration de Nextcloud par notre procédure de restauration.

3

GITLAB

Gestion d’une instance GitLab

A. Introduction

GitLab est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.

Notre instance **Gitlab** est hébergée sur un serveur dédié géré par **Ikoula**.

B. Installation

Dans cette procédure d'installation, nous allons mettre en place une instance GitLab sur une machine possédant la distribution Debian 11 Server.

Le sous-domaine qui permettra d'accéder au GitLab est **forge.naos-cluster.tech**.

1. Installation des pré-requis

Installation des paquets nécessaire pour la configuration de dépôts :

```
sudo apt update
sudo apt install -y curl ca-certificates perl
```

2. Installation de Postfix

```
sudo apt install postfix
```

Postfix permettra à GitLab d'envoyer des mails.

3. Installation de GitLab

Configuration des dépôts de GitLab (Community Edition) :

```
curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh |
sudo bash
```

Installation de GitLab

Attention : Lors de l'installation, il est nécessaire de spécifier l'adresse permettant d'accéder à l'interface web de notre GitLab

```
sudo EXTERNAL_URL="https://forge.naos-cluster.tech" apt install gitlab-ce
```

4. Configuration SMTP

Pour modifier la configuration SMTP de notre instance GitLab, ouvrez le fichier **/etc/gitlab/gitlab.rb** et éditez les lignes suivantes :

```
gitlab_rails['smtp_enable'] = true
gitlab_rails['smtp_address'] = "smtp.server"
gitlab_rails['smtp_port'] = 465
gitlab_rails['smtp_user_name'] = "smtp user"
gitlab_rails['smtp_password'] = "smtp password"
gitlab_rails['smtp_domain'] = "example.com"
gitlab_rails['smtp_authentication'] = "login"
gitlab_rails['smtp_enable_starttls_auto'] = true
gitlab_rails['smtp_openssl_verify_mode'] = 'peer'

# If your SMTP server does not like the default 'From: gitlab@localhost' you
# can change the 'From' with this setting.
gitlab_rails['gitlab_email_from'] = 'gitlab@example.com'
gitlab_rails['gitlab_email_reply_to'] = 'noreply@example.com'
```

Lorsque vous avez terminé la configuration SMTP de l'instance GitLab, rechargez la configuration de GitLab avec la commande suivante :

```
sudo gitlab-ctl reconfigure
```

5. Première connexion au GitLab

Lors de l'installation de notre GitLab, un compte **root** (par défaut) a été créé. Son mot de passe est stocké sur la machine qui l'héberge dans le fichier **/etc/gitlab/initial_root/password**.

Vous avez le choix entre vous approprier ce compte (modification du nom, de l'adresse mail, du mot de passe, etc ...) ou bien en créer un nouveau et désactiver celui ci.

C. Mise à jour

GitLab est installé avec **APT**. Il nous suffit donc de lancer les commandes suivantes pour mettre à jour notre instance :

```
sudo apt update
sudo apt upgrade
```

Il est possible de voir la liste des paquets qui seront mis à jour grâce à la commande suivante :

```
sudo apt list --upgradable
```

Cette commande est à lancer avant la commande **upgrade**.

D. Sauvegarde

Avant toute chose, nous allons éditer le fichier **/etc/gitlab/gitlab.rb** et définir le dossier qui contiendra les sauvegardes (le chemin par défaut est **/var/opt/gitlab/backups/**)

```
gitlab_rails['backup_path'] = '/var/opt/gitlab/backups/'
```

Rechargez la configuration GitLab :

```
sudo gitlab-ctl reconfigure
```

Lorsque le dossier de sauvegarde est défini, il suffit de lancer la commande suivante pour créer une sauvegarde :

```
sudo gitlab-backup create
```

Lorsque la sauvegarde est terminée, vous trouverez votre archive dans le dossier que vous avez préalablement défini dans le fichier de configuration de GitLab.

Pour la restauration et la migration, les fichiers **/etc/gitlab/gitlab-secrets.json** et **/etc/gitlab/gitlab.rb** ainsi que les dossiers **/etc/gitlab/trusted-certs** et **/etc/gitlab/ssl**.

E. Restauration / Migration

Avant de pouvoir restaurer la sauvegarde, vérifiez que la version de votre instance GitLab soit la même que celle de la sauvegarde.

Pour cette partie, le nom de notre sauvegarde sera **11493107454_2018_04_25_10.6.4-ce_gitlab_backup.tar**.

Si le fichier de sauvegarde a été importé sur le serveur, il faut le déplacer dans le dossier contenant les sauvegardes de GitLab (chemin définir dans la partie **Sauvegarde**) et lui attribuer le bon propriétaire :

```
sudo cp 11493107454_2018_04_25_10.6.4-ce_gitlab_backup.tar /var/opt/gitlab/backups/
sudo chown git:git /var/opt/gitlab/backups/11493107454_2018_04_25_10.6.4-ce_gitlab_backup.tar
```

Ensuite, nous arrêtons les processus connectés à la base de données :

```
sudo gitlab-ctl stop puma
sudo gitlab-ctl stop sidekiq

# Vérification
sudo gitlab-ctl status
```

Et nous restaurons la sauvegarde en spécifiant une partie du nom de la sauvegarde (la partie avant **_gitlab_backup.tar**) :

```
sudo gitlab-backup restore BACKUP=11493107454_2018_04_25_10.6.4-ce
```

Après avoir restauré l'archive de sauvegarde, restaurer les autres fichiers/dossiers si nécessaire (**/etc/gitlab/gitlab-secrets.json**, **/etc/gitlab/gitlab.rb**, **/etc/gitlab/trusted-certs** et **/etc/gitlab/ssl**).

Lorsque la restauration de l'archive et des fichiers/dossiers est terminée, rechargez et relancez l'instance GitLab :

```
sudo gitlab-ctl reconfigure
sudo gitlab-ctl restart
```

Et pour finir, nous vérifions que l'instance se soit bien restaurée :

```
sudo gitlab-rake gitlab:check SANITIZE=true
```

4

MATTERMOST

Gestion d’une instance Mattermost

A. Introduction

Mattermost est un logiciel et un service de messagerie instantanée libre auto-hébergeable. Il est conçu comme un chat interne pour les organisations et les entreprises, et il est présenté comme une alternative à Slack et Microsoft Teams.

Nous allons installer Mattermost dans un conteneur à partir d'une image docker. Il sera placé derrière un reverse proxy lui aussi conteneurisé à partir d'une image docker.

Les conteneurs tourneront sur une machine sous Debian 11. Il sera nécessaire d'adapter quelques commandes en fonction de votre distribution.

Notre machine sera un serveur virtuel privée géré par **MVPS**.

B. Installation

Le sous-domaine qui permettra d'accéder au cloud est **chat.naos-cluster.tech**. Modifiez le lors de la configuration de l'URL d'accès et la génération du certificat Let's Encrypt pour l'adapter à votre infrastructure.

Pour le bon fonctionnement de l'instance Mattermost, il sera nécessaire d'installer Docker ainsi que docker-compose sur la machine.

1. Installation des pré-requis

Installation des paquets nécessaire pour la configuration de dépôts :

```
sudo apt update
sudo apt install -y ca-certificates curl gnupg lsb-release
```

Ajout de la clé GPG officielle de Docker (Pour Debian 11)

```
sudo mkdir -p /etc/apt/keyrings
 curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
```

Configuration du dépôt

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Installation de Docker et docker-compose :

```
apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-
compose
```

L'installation de Docker est terminée.

2. Mise en place du Reverse Proxy

Clonage du dépôt :

```
git clone --recurse-submodules https://github.com/evertramos/nginx-proxy-automation.git
```

Lorsque le clonage du dépôt se termine, allez dans le dossier et lancez la commande suivante :

```
./fresh-start.sh --yes -e <votre_email@domaine> --skip-docker-image-check
```

Remplacez **<votre_email@domaine>** par l'adresse électronique qui sera utilisée pour Let's Encrypt.

Votre reverse proxy est maintenant lancé.

3. Installation de Mattermost

Clonage du dépôt :

```
git clone https://github.com/EugenMayer/mattermost-boilerplate.git
```

Aller dans le dossier et le éditer fichier .env et modifier la dernière ligne :

```
DB_USER=mattermost
DB_PASSWORD=<password>
DB_NAME=mattermostdb
VERSION=5.33.2
MM_SERVICESETTINGS_SITEURL=https://<votre_nom_de_domaine>
```

Modifier le mot de passe de l'utilisateur ayant accès à la base de données et modifier l'URL permettant d'accéder à votre instance Mattermost.

Ajouter la ligne suivante à la suite du fichier .env :

```
VIRTUAL_HOST=<votre_nom_de_domaine>
```

Remplacer **<votre_nom_de_domaine>** par votre nom de domaine.

Vous devez avoir un fichier dont le contenu ressemble à ça :

```
DB_USER=mattermost
DB_PASSWORD=<password>
DB_NAME=mattermostdb
VERSION=5.33.2
MM_SERVICESETTINGS_SITEURL=https://<votre_nom_de_domaine>
VIRTUAL_HOST=<votre_nom_de_domaine>
```

Éditer le fichier **docker-compose.yml** :

- Ajouter le bloc suivant à la fin du fichier (Ajout du réseau créé par le conteneur nginx-proxy et du réseau mattermost qui permettra la communication entre l'application et la base de données)

```
networks:
  mattermost:
  proxy:
    external: true
```

- Ajouter le bloc suivant à la fin du service **app**. Ce bloc permet d'ajouter votre conteneur au réseau **proxy**

```
    networks:
      - proxy
      - mattermost
```

- Ajouter le bloc suivant à la fin du service **db**. Ce bloc permet aux conteneurs **app** et **db** de communiquer via le réseau **mattermost**

```
    networks:
      - mattermost
```

- Ajouter les deux lignes suivantes aux variables d'environnement du service **app**. Ces variables à Nginx de récupérer un certificat SSL pour votre nom de domaine.

```
      VIRTUAL_HOST: ${VIRTUAL_HOST}
      VIRTUAL_PORT: 8065
```

Votre fichier final devrait ressembler à ça :

```
version: "2"

services:
  app:
    image: mattermost/mattermost-team-edition:release-5.37
    environment:
      MM_USERNAME: ${DB_USER}
      MM_PASSWORD: ${DB_PASSWORD}
      MM_DBNAME: ${DB_NAME}
      MM_SERVICESETTINGS_SITEURL: ${MM_SERVICESETTINGS_SITEURL}
      DB_HOST: db
      DB_PORT_NUMBER: 5432
      VIRTUAL_HOST: ${VIRTUAL_HOST}
      VIRTUAL_PORT: 8065
```

```
      MM_SQLSETTINGS_DATASOURCE: "postgres://${DB_USER}:${DB_PASSWORD}@db:5432/$
{DB_NAME}?sslmode=disable&connect_timeout=10"



    volumes:
      - mattermost_config:/mattermost/config
      - mattermost_data:/mattermost/data
      - mattermost_logs:/mattermost/logs
      - mattermost_plugins:/mattermost/plugins
      - mattermost_client_plugins:/mattermost/client/plugins
    networks:
      - proxy
      - mattermost

  db:
    image: postgres:10
    volumes:
      - mattermost_db:/var/lib/postgresql/data
    environment:
      POSTGRES_DB: ${DB_NAME}
      POSTGRES_USER: ${DB_USER}
      POSTGRES_PASSWORD: ${DB_PASSWORD}
      POSTGRES_ENCODING: "UNICODE"
      POSTGRES_COLLATE: "C"
      POSTGRES_COLLATE_TYPE: "C"
    networks:
      - mattermost

volumes:
  mattermost_config:
    driver: local
  mattermost_data:
    driver: local
  mattermost_logs:
    driver: local
  mattermost_plugins:
    driver: local
  mattermost_client_plugins:
    driver: local
  mattermost_db:
    driver: local

networks:
  mattermost:
  proxy:
    external: true
```

C. Mise à jour

Le fait d'avoir une application « dockerisée » nous facilite la tâche au niveau des mises à jour. En effet, il nous suffit de changer la version de l'application dans le fichier **.env** et de relancer les conteneurs avec les commandes suivantes (vous devez être placé dans le dossier où se trouve le fichier docker-compose.yml) :

```
docker-compose down
docker-compose up -d
```

ATTENTION : Bien que la procédure de mise à jour soit relativement simple, il est tout de même conseillé de faire une sauvegarde avant de la réaliser.

D. Sauvegarde / Restauration

Étant une application conteneurisée, une sauvegarde des différents volumes est suffisante. Pour nous faciliter la tâche, nous avons mis en place un script script permettant d'exporter les volumes (volume et bind mount) d'un conteneur :

https://forge.naos-cluster.tech/naos-internal/docker-volume-backup

Pour exporter les volumes d'un conteneur, il suffit de taper la commande suivante :

```
./docker-container-backup.sh <backup_path> <container_name>
```

<backup_path> : Le chemin où sera stockée la sauvegarde.

<container_name> : Le conteneur dont les volumes seront sauvegardés.

Le dépôt contient également un script permettant, à partir d'une sauvegarde réaliser avec le script **docker-container-backup.sh**, la restauration des volumes d'un conteneur :

```
./docker-container-restore.sh <container_backup_path>
```

<container_backup_path> : Le chemin où est stockée la sauvegarde des volumes du conteneur.

À noter qu'il est possible d'exporter et restaurer qu'un seul volume grâce aux scripts **docker-volume-backup.sh** et **docker-volume-restore.sh**.

5

NAOS HUB

Gestion d’une instance Humhub

A. Introduction

HumHub est un logiciel de réseau social gratuit et open source écrit au-dessus du framework PHP Yii qui fournit une boîte à outils facile à utiliser pour créer et lancer votre propre réseau social.

Comme pour les sites Wordpress, NAOS utilise un prestataire pour l'hébergement de son raccourcisseur de lien, nous n'allons donc pas aborder la partie installation et configuration d'un serveur Apache2, de PHP et de MySQL.

Notre application web sera hébergé par **o2switch**.

B. Installation

Commencez par créer une base de données qui sera utilisée par notre réseau social.

En général, les hébergeurs web fournissent un accès à votre/vos base(s) de données via PhpMyAdmin. Si la base de données est déjà créée, vous pouvez sauter cette étape. Si ce n'est pas le cas, vous devez la créer.

Si votre hébergeur web ne fourni pas d'accès via PhpMyAdmin, voici comment créer en ligne de commande une base de données et un utilisateur dédié à celle-ci :

```
# Création de la base de données
CREATE DATABASE humhub;

#Création d'un utilisateur et attribution des droits d'accès à la base de données
GRANT ALL PRIVILEGES on humhub.* to 'humhub'@'localhost' identified by 'password';

# Mise à jour de la table des privilèges
FLUSH PRIVILEGES;
```

Lorsque la création de notre base de données et de notre utilisateur est terminée, nous récupérons l'archive contenant la dernière version de HumHub, nous l'extrayons puis nous envoyons les fichiers sur notre hébergement web (via FTP ou SFTP).

Lien vers la dernière version de HumHub :

https://www.humhub.com/download/package/humhub-1.12.1.zip

Lorsque les fichiers sont sur votre hébergeur, accédez au site.

C. Configuration

Lorsque vous arrivez sur votre site, vous avez la possibilité de cliquer sur « Suivant » ou de changer la langue.

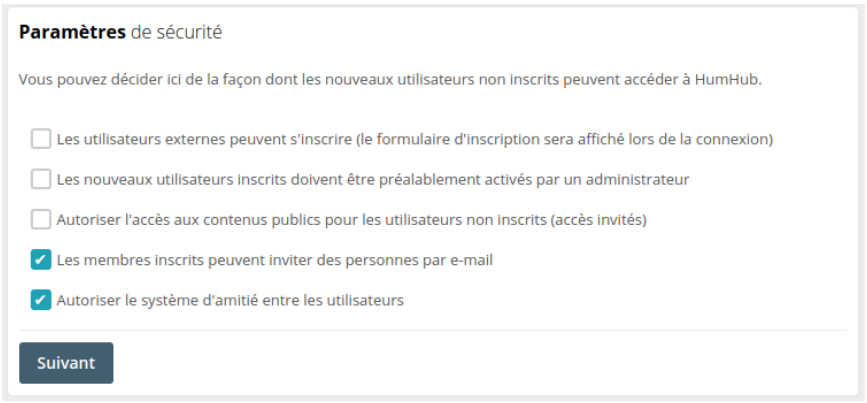
Lorsque vous cliquez sur suivant, vous arrivez dans la partie « Vérification du **système** ». Cette partie vérifie que les différentes extensions PHP nécessaire au bon fonctionnement de HumHub sont installées. Si vous ne gérez pas la partie système, vous pouvez normalement cliquer sur « Suivant ».

Dans la prochaine partie, vous devrez configurer la connexion à votre base de données.

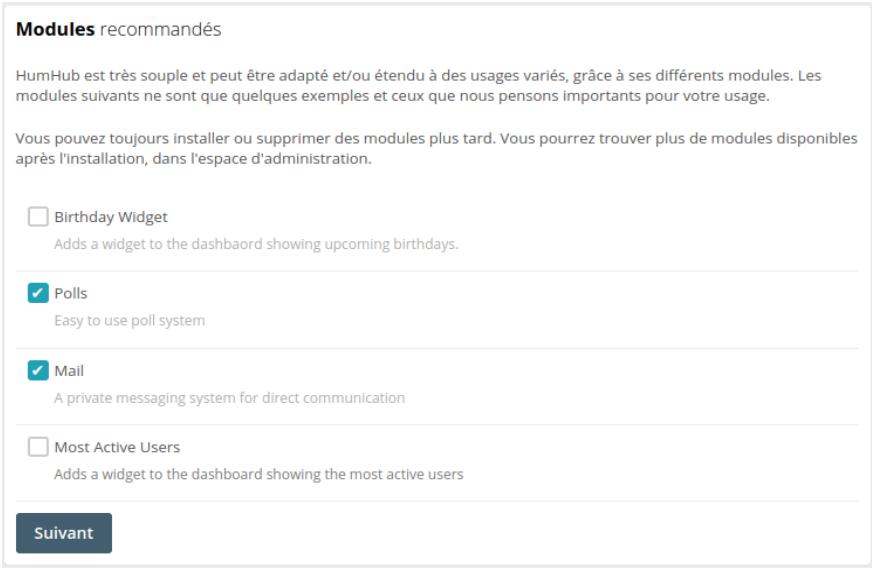
Dans la suite de cette partie, les paramètres choisis sont ceux de NAOS HUB, libre à vous de les modifier.

Il vous sera ensuite demandé de choisir :

- un nom, celui qui vous convient
- un type, nous avons choisi « ma communauté »
- Les paramètres de sécurité pour votre réseau social :



- Les modules recommandés à installer :



Maintenant, vous devrez créer le compte Administrateur.

Et pour finir, cliquez sur « **Installer des exemples de contenu (recommandé)** », terminez l'installation et cliquez sur « **Se connecter** ».

Il est possible d'installer des modules depuis la partie **Paramtères > Modules**. Voici les modules installés sur notre instance : Gallery, Wiki, ONLYOFFICE Connector for HumHub, HumHub Updater, draw.io – Diagrams, Custom Pages, Mail, Polls, Files et Tasks.

D. Branding

Pour la customisation du thème, nous commençons par la création d'un thème qui nous permettra de changer les couleurs du site.

Dans le dossier **themes**, faire une copie du dossier HumHub dans le même dossier et le renommer en **NAOS**.

Dans ce dossier, éditer le fichier **variables.less** présent dans le dossier **less** et y ajouter les lignes suivantes :

```
@default: #f3f3f3;
@primary: #435f6f;
@info: #21A1B3;
@success: #97d271;
@warning: #FFC107;
@danger: #FC4A64;
```

Ensuite, se placer dans le dossier du thème et exécuter la commande suivante :

```
lessc --clean-css less/build.css css/theme.css
```

Cette commande permet de créer le fichier theme.css qui sera utilisé par HumHub.

Pour appliquer le thème, il suffit de se rendre dans la partie **Paramètres > Apparence**, choisir votre thème et cliquer sur enregistrer. Cette partie permet également de modifier l'icône ainsi que le logo utilisé pour le site.

E. Mise à jour

HumHub possède un module de mise à jour nommé « HumHub Updater ». Il permet, en quelques clics, de mettre à jour l'application web.

Il y a tout de même quelques actions à réaliser en amont dans le cas où un problème surviendrait lors de la mise à jour :

- Réaliser une sauvegarde des fichiers du site.
- Réaliser une sauvegarde de la base de données.

Il est aussi conseillé de mettre à jour les modules avant et après la mise à jour de HumHub et de remettre le thème par défaut.

F. Sauvegarde

Pour réaliser une sauvegarde de l'application, il suffit de faire un export de la base de données et une copie des fichiers du site.

Pour réaliser une restauration, il suffit d'importer la base de données ainsi que les fichiers du site.

6

Site Wordpress

Mise en place d’un site Wordpress

A. Introduction

WordPress est un système de gestion de contenu (SGC ou content management system (CMS) en anglais) gratuit, libre et open-source.

NAOS utilise un prestataire pour l'hébergement de ses sites Wordpress, nous n'allons donc pas aborder la partie installation et configuration d'un serveur Apache2, de PHP et de MySQL.

B. Installation et configuration d'un site Wordpress

1. Installation

Nous commençons par créer une base de données qui servira à notre site Wordpress.

En général, les hébergeurs web fournissent un accès à votre/vos base(s) de données via PhpMyAdmin. Si la base de données est déjà créée, vous pouvez sauter cette étape. Si ce n'est pas le cas, vous devez la créer.

Si votre hébergeur web ne fourni pas d'accès via PhpMyAdmin, voici comment créer en ligne de commande une base de données et un utilisateur dédié à la base de données :

```
# Création de la base de données
CREATE DATABASE wordpress;

#Création d'un utilisateur et attribution des droits d'accès à la base de données
GRANT ALL PRIVILEGES on wordpress.* to 'wordpress'@'localhost' identified by 'password';

# Mise à jour de la table des privilèges
FLUSH PRIVILEGES;
```

Lorsque la création de notre base de données et de notre utilisateur est terminée, nous récupérons l'archive contenant la dernière version de wordpress, nous l'extrayons puis envoyons les fichiers sur notre hébergement web (via FTP ou SFTP)

Lien vers la dernière version de wordpress : https://fr.wordpress.org/latest-fr_FR.zip

2. Configuration

Liste des plugins essentiels installés sur tous nos sites Wordpress :

- **Cookie Notice & Compliance for GDPR / CCPA** (Affichage de la barre pour les cookies)

- **Disable XML-RPC** (Désactivation du XMLRPC)
- **Elementor et Elementor Pro** (Constructeur de page)
- **hCaptcha** (Captcha pour les formulaires)
- **UpdraftPlus** (Sauvegarde et restauration)
- **Wordfence Security** (Sécurité du site)
- **WP Mail SMTP** (Configuration SMTP avancée pour l'envoi de mail)
- **WP-Optimize** (Optimisation des fichiers, mise en cache des pages)
- **WPS HideLogin** (Changer l'url de la page pour se connecter au back-office)
- **Yoast SEO** (Référencement)

Comment installer un plugin

Dans la partie **Extensions > Ajouter**, vous pouvez rechercher des plugins et les installer.

Configuration du plugin UpdraftPlus

Dans la partie **Réglages > Sauvegardes UpdraftPlus** dans le menu de gauche du back-office, dans l'onglet **Réglages**, vous pourrez planifier la sauvegarde des fichiers et celle de la base de données.

Il est possible de modifier le dossier où seront stockées les sauvegardes et même d'envoyer les sauvegardes sur un serveur de stockage distant.

Nous n'utiliserons pas la fonction de stockage sur un serveur distant.

Comment définir une page Elementor en tant que d'accueil

Dans la partie **Réglages > Lecture** dans le back-office, définissez la page d'accueil sur une page statique, choisissez votre page Elementor dans le menu dépliant **Page d'accueil**, et enregistrez les modifications.

C. Mise à jour

Dans la partie **Tableau de bord > Mises à jour** dans le menu de gauche du back-office, vous avez la listes des mises à jour disponibles. A partir de cette page, vous pourrez mettre à jour la version du Wordpress, la traduction du Wordpress (dans la langue choisie) et les plugins installés.

D. Sauvegarde (UpdraftPlus)

Dans la partie **Réglages > Sauvegardes UpdraftPlus** dans le menu de gauche du back-office, vous verrez un bouton **Sauvegarder**. Ce bouton vous permet de faire une sauvegarde complète ou partiel (exemple: uniquement base de données) en un clic.

Une fois que votre sauvegarde se termine, vous la verrez apparaître dans la liste en dessous du bouton **Sauvegarder** et plusieurs informations seront disponibles (date de la sauvegarde, données sauvegardées) ainsi que plusieurs actions (Restaurer, Supprimer et Voir le journal).

E. Restauration (UpdraftPlus)

Dans la liste des sauvegardes que nous avons vu dans la partie **Sauvegarde**, choisissez en une et cliquez sur **Restaurer**. Vous n'avez rien de plus à faire.

Il est recommandé de tester le site après une restauration.

F. Migration (UpdraftPlus)

Pour la migration, il nous faut d'abord récupérer une sauvegarde réalisée à partir du plugin UpdraftPlus. Les fichiers de sauvegarde se trouve dans le dossier **wp-content/updraft** au format suivant :

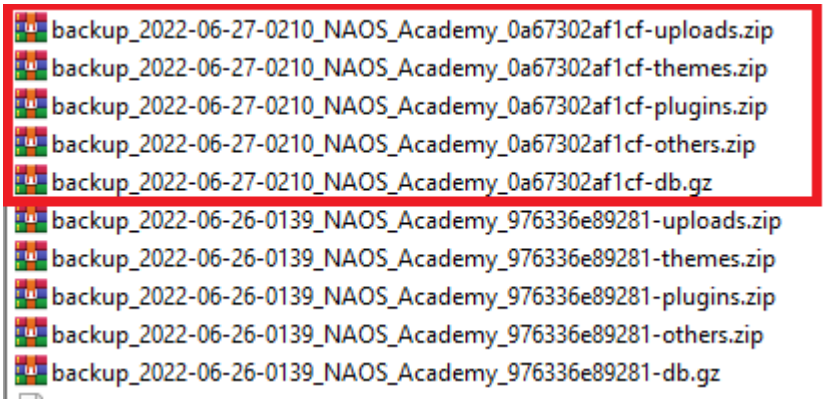
backup_2022-06-26-0139_NAOS_Academy_976336e89281-db.gz

Le nom du fichier est composé de :

- La date de la sauvegarde : **2022-06-26-0139**
- Du nom du site wordpress : **NAOS_Academy**
- L'identifiant de la sauvegarde : **976336e89281**
- Du type de la sauvegarde : **db**

Suivant le type de sauvegarde, l'extension peut varier (**gz** pour les sauvegardes de base de données et **zip** pour le reste).

Pour pouvoir réaliser une restauration complète, nous aurons besoin de récupérer une sauvegarde entière :



Sur la capture d'écran précédente, nous avons 2 sauvegardes complètes. Nous allons récupérer tous les fichiers de la sauvegarde la plus récente (les fichiers présents dans le cadre rouge).

Pour migrer un site Wordpress avec UpdraftPlus, il nous faudra installer un site Wordpress avec le plugin UpdraftPlus sur le nouvel hébergeur web. Lorsque le site et le plugin UpdraftPlus sont installés et configurés, nous envoyons la sauvegarde sur notre nouvel hébergeur (dans le même dossier de sauvegarde que sur notre ancien hébergeur).

Lorsque l'envoi des fichiers est terminé, nous allons demander à UpdraftPlus de scanner le nouveau dossier de sauvegarde. Pour le faire, dans la partie back-office de wordpress, allez dans **Réglages > Sauvegardes UpdraftPlus** et en dessous de **Sauvegardes existantes**, cliquer sur **Scanner le dossier local pour recherche de nouveaux lots de sauvegarde**. Attendez un petit moment et votre sauvegarde devrait apparaître. Lorsque vous verrez votre sauvegarde dans la liste, restaurez la.

Comme pour la partie **Restauration**, Il est recommandé de tester le site après une restauration.

7

UpdraftCentral

Gestion de plusieurs sites Wordpress

UpdraftCentral

A. Introduction

UpdraftCentral est un plugin Wordpress permettons de gérer plusieurs fonctionnalités de plusieurs sites Wordpress possédant UpdraftPlus à partir d'un seul site. Nous l'utilisons principalement pour la gestion des sauvegardes et des mises à jour.

NAOS utilise un prestataire pour l'hébergement de ses sites Wordpress, nous n'allons donc pas aborder la partie installation et configuration d'un serveur Apache2, de PHP et de MySQL.

Notre application web sera hébergé par **o2switch**.

B. Installation et configuration du site de Supervision

1. Installation

Commencez par créer une base de données qui servira à notre site Wordpress.

En général, les hébergeurs web fournissent un accès à votre/vos base(s) de données via PhpMyAdmin. Si la base de données est déjà créée, vous pouvez sauter cette étape. Si ce n'est pas le cas, vous devez la créer.

Si votre hébergeur web ne fourni pas d'accès via PhpMyAdmin, voici comment créer en ligne de commande une base de données et un utilisateur dédié à la base de données :

```
# Création de la base de données
CREATE DATABASE wordpress;

#Création d'un utilisateur et attribution des droits d'accès à la base de données
GRANT ALL PRIVILEGES on wordpress.* to 'wordpress'@'localhost' identified by 'password';

# Mise à jour de la table des privilèges
FLUSH PRIVILEGES;
```

Lorsque la création de notre base de données et de notre utilisateur est terminée, nous récupérons l'archive contenant la dernière version de wordpress, nous l'extrayons puis envoyons les fichiers sur notre hébergement web (via FTP ou SFTP).

Lien vers la dernière version de wordpress :

https://fr.wordpress.org/latest-fr_FR.zip

2. Configuration

Liste des plugins essentiels installés sur tous nos sites Wordpress :

- **Disable XML-RPC** (Désactivation du XMLRPC)
- **hCaptcha** (Captcha pour les formulaires)
- **UpdraftCentral Dashboard** (Gestion des instances Wordpress à distance)
- **Wordfence Security** (Sécurité du site)
- **WP Mail SMTP** (Configuration SMTP avancée pour l'envoi de mail)
- **WPS HideLogin** (Changer l'url de la page pour se connecter au back-office)

Comment installer un plugin

Dans la partie **Extensions > Ajouter**, vous pouvez rechercher des plugins et les installer.

Configuration du plugin UpdraftCentral

Avant toute chose, nous allons créer la page qui nous permettra de gérer nos sites Wordpress.

Pour ça, dans le back-office, cliquez sur **Pages > Ajouter**. Nous allons lui donner un nom et lui ajouter un code court pour le contenu :

Notre page s'appelle **ma page de supervision** et son contenu est **[updraft_central]**.

[updraft_central] est un code cour qui permettra d'afficher le tableau de bord de UpdraftCentral.

Il est possible de définir cette page comme page par défaut de votre site Wordpress en allant dans la partie **Réglages > Lecture** et en ajoutant la page que vous venez de créer en page d'accueil.

C. Ajouter un site à superviser

Sur le site Wordpress que vous voulez faire superviser par UpdraftCentral, dans le back-office puis dans la partie Réglages > Sauvegardes UpdraftPlus, choisissez l'onglet Outils avancés et ensuite UpdraftCentral. Sélectionnez Tableau de bord auto-hébergé, spécifiez l'url de votre site Wordpress où est installé UpdraftCentral et cliquez sur suivant puis Créer.

Cette opération va créer une paire de clés qui permettra à votre site de supervision de communiquer avec votre site supervisé.

Récupérez la clé créée et aller sur votre site de supervision, sur la page que nous avons créé dans la partie Configuration du plugin UpdraftCentral. Lorsque vous arrivez sur la page, cliquez en haut à droite sur le bouton +, insérez la clé et cliquez sur Add Website.

Votre site apparaît dans la liste des sites supervisés.

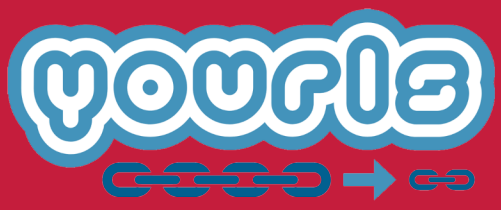
D. Mise à jour du site de supervision

Notre site est créé avec Wordpress, il nous suffit donc d'aller dans la partie **Tableau de bord > Mises à jour** dans le menu de gauche du back-office pour pouvoir mettre à jour la version du Wordpress, la traduction du Wordpress (dans la langue choisie) et les plugins installés.

8

YOURLS

Raccourcisseur de lien



A. Introduction

YOURLS signifie **Your Own URL Shortener**. Il s'agit d'un petit ensemble de scripts PHP permettant d'avoir son propre service de raccourcissement d'URL (comme TinyURL ou bitly). Cette application va beaucoup plus loin en analysant la provenance des utilisateurs accédant au lien court.

Comme pour les sites Wordpress, NAOS utilise un prestataire pour l'hébergement de son raccourcisseur de lien, nous n'allons donc pas aborder la partie installation et configuration d'un serveur Apache2, de PHP et de MySQL.

Notre application web sera hébergé par **o2switch**.

B. Installation et configuration

1. Installation

Nous commençons par créer une base de données qui sera utilisée par notre raccourcisseur d'URL.

En général, les hébergeurs web fournissent un accès à votre/vos base(s) de données via PhpMyAdmin. Si la base de données est déjà créée, vous pouvez sauter cette étape. Si ce n'est pas le cas, vous devez la créer.

Si votre hébergeur web ne fourni pas d'accès via PhpMyAdmin, voici comment créer en ligne de commande une base de données et un utilisateur dédié à celle-ci :

```
# Création de la base de données
CREATE DATABASE yourls;

#Création d'un utilisateur et attribution des droits d'accès à la base de données
GRANT ALL PRIVILEGES on yourls.* to 'yourls'@'localhost' identified by 'password';

# Mise à jour de la table des privilèges
FLUSH PRIVILEGES;
```

Lorsque la création de notre base de données et de notre utilisateur est terminée, nous récupérons l'archive contenant la dernière version de YOURLS, nous l'extrayons puis envoyons les fichiers sur notre hébergement web (via FTP ou SFTP)

Lien vers la dernière version de YOURLS :

https://github.com/YOURLS/YOURLS/archive/refs/tags/1.9.1.zip

2. Configuration

Lorsque les fichiers sont sur le serveur, aller dans le dossier **user**, dupliquer le fichier config-sample.php et renommer la copie en **config.php**

Éditer le fichier **config.php** :

- Remplir les informations permettant la connexion à la base de données

```
/** MySQL database username */
define( 'YOURLS_DB_USER', '<utilisateur>' );

/** MySQL database password */
define( 'YOURLS_DB_PASS', '<mot_de_passe>' );

/** The name of the database for YOURLS
 ** Use lower case letters [a-z], digits [0-9] and underscores [_] only */
define( 'YOURLS_DB_NAME', '<nom_de_la_base_de_données>' );

/** MySQL hostname.
 ** If using a non standard port, specify it like 'hostname:port', e.g. 'localhost:9999'
or '127.0.0.1:666' */
define( 'YOURLS_DB_HOST', '<hôte>' );

/** MySQL tables prefix
 ** YOURLS will create tables using this prefix (eg `yourls_url`, `yourls_options`, ...)
 ** Use lower case letters [a-z], digits [0-9] and underscores [_] only */
define( 'YOURLS_DB_PREFIX', 'yourls_' );
```

- Définir l'URL d'accès

```
/** YOURLS installation URL
 ** All lowercase, no trailing slash at the end.
 ** If you define it to "http://sho.rt", don't use "http://www.sho.rt" in your browser
(and vice-versa)
 ** To use an IDN domain (eg http://héhé.com), write its ascii form here (eg http://xn--
hh-bjab.com) */
define( 'YOURLS_SITE', 'https://<votre_nom_de_domaine>' );
```

- Définir le hash secret aléatoire permettant de hasher les cookies

```
/** A random secret hash used to encrypt cookies. You don't have to remember it, make it
long and complicated
 ** Hint: copy from http://yourls.org/cookie */
define( 'YOURLS_COOKIEKEY', '<chaîne_de_caractères_aléatoire>' );
```

Cette chaîne de caractères sera utilisée pour hasher les cookies

– Définir un nom d'utilisateur et un mot de passe

```
$yourls_user_passwords = [
      '<user>' => '<password>' /* Password encrypted by YOURLS */ ,
      // 'username2' => 'password2',
      // You can have one or more 'login'=>'password' lines
];
```

Ces identifiants vous permettront de vous connecter au panel d'administration. De plus, le mot de passe sera hashé lors de l'installation de YOURLS.

Lorsque l'édition du fichier **config.php** est terminée, aller à l'adresse suivante :

https://<votre_nom_de_domaine>/admin

Et cliquez sur le bouton « **Install YOURLS** »

C. Mise à jour

La procédure de mise à jour de YOURLS est simple. Il suffit de télécharger l'archive contenant la nouvelle version et de la décompresser son contenu dans le dossier où sont déjà présent les anciens fichiers de YOURLS (la configuration et la base de données ne seront pas affectées).

Il est tout de même conseillé de réaliser une sauvegarde avant de mettre à jour l'application.

D. Sauvegarde

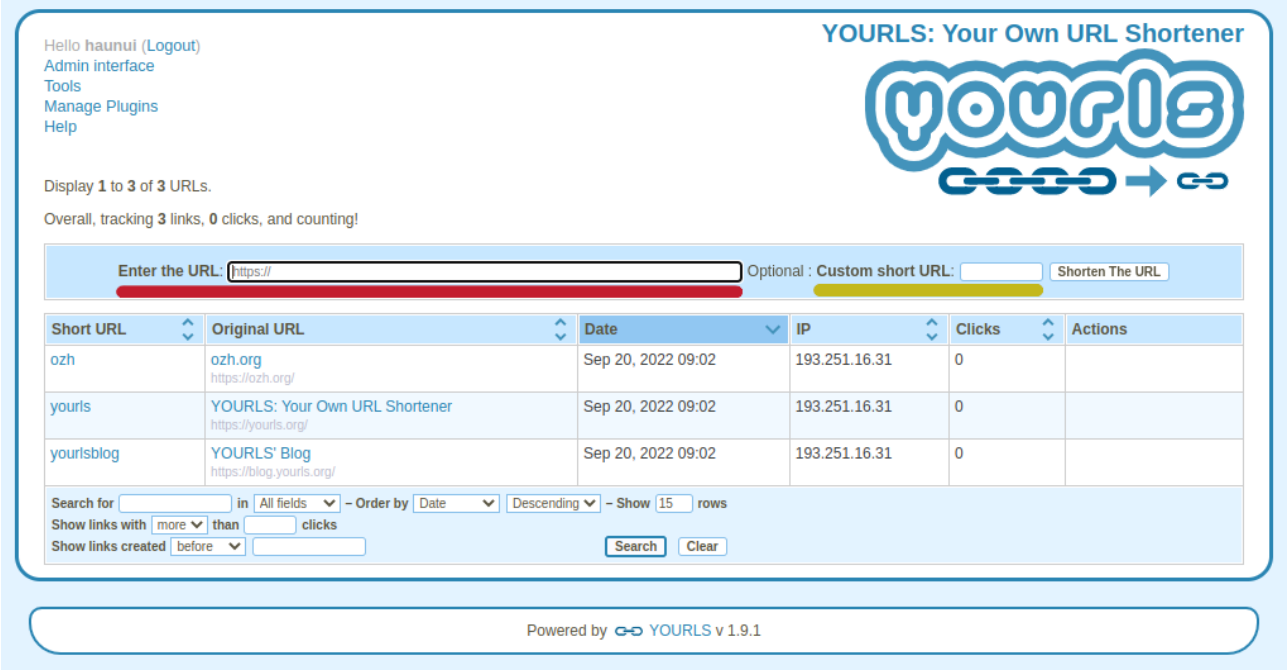
Un export de la base de données ainsi qu'une copie du fichier de configuration suffiront pour sauvegarder l'application.

E. Restauration

Pour restaurer l'application, il suffit de réimporter la base de données sur le serveur de base de données et le fichier de configuration dans le dossier **user**.

F. Utilisation

Voici la page d'accueil de YOURLS. Elle permet de créer des liens courts.



La partie soulignée en rouge permet d'ajouter le lien vers le quel l'utilisateur sera redirigé lorsque qu'il accédera au lien court.

La partie soulignée en jaune permet de customiser le lien court. Cette partie est optionnelle.

Lorsque vous survolez un lien, vous verrez apparaître plusieurs icônes dans la partie « **Actions** ». La première permet de visualiser les statistiques du trafic du lien court (combien de clics dans un période donnée, la provenance des clics et les sources).

La deuxième icône permet d'éditer l'URL et la troisième de la supprimer.

9

SIGGEN

Générateur de signature d’adresse mail NAOS

A. Introduction

Pour une homogénéité des signatures de nos adresses mail, une application PHP a été mise en place. Cette application permet de générer une signature de mail en HTML.

Notre application web sera hébergé par **o2switch**.

B. Installation

1. Pré-requis

Pour fonctionner, notre application aura besoin d'un serveur Web (Apache2 ou Nginx), de PHP et de certbot (pour générer un certificat SSL). Dans notre cas, nous utiliserons un serveur Apache2.

Recherche des mises à jour disponibles :

```
sudo apt update
```

Installation d'Apache2, PHP et certbot

```
apt update
apt install apache2 php certbot python3-certbot-apache unzip
```

Configuration d'Apache2

Création du fichier de configuration **siggen.naos-cluster.tech.conf** dans le dossier **/etc/apache2/sites-availables/**

```
<VirtualHost *:80>
	ServerAdmin sysadmin@naos-cluster.tech
	ServerName siggen.naos-cluster.tech

	Redirect permanent / https://siggen.naos-cluster.tech/
</VirtualHost>

<VirtualHost *:443>
	ServerAdmin sysadmin@naos-cluster.tech
	ServerName siggen.naos-cluster.tech

	DocumentRoot /var/www/html/
	<Directory /var/www/html/>
		Order allow,deny
		Allow from all
		Require all granted
	</Directory>
```

```
</VirtualHost>
```

Activation du mode **rewrite** et **ssl** d'Apache2 :

```
a2enmod rewrite
a2enmod ssl
```

Le mode **ssl** nous permettra d'utiliser le protocole HTTPS et le mode **rewrite** nous permettra de forcer la connexion en HTTPS.

Activation du site **siggen.naos-cluster.tech :**

```
a2ensite siggen.naos-cluster.tech
```

Il ne reste plus qu'à redémarrer Apache2 :

```
systemctl restart apache2
```

2. Installation de l'application

Lien vers le dépôt de l'application : https://forge.naos-cluster.tech/naos-internal/siggen

Récupération et installation des fichiers de l'application dans le dossier **/var/www/html/**

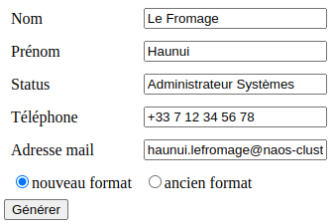
```
cd /var/www/html/
wget https://forge.naos-cluster.tech/naos-internal/siggen/-/archive/master/siggen-master.zip
unzip siggen-master.zip
mv siggen-master/* .
rm -r siggen-master*
```

Si l'installation s'est bien déroulée, le générateur de signature est disponible à l'adresse https://siggen.naos-cluster.tech/.

Pour modifier la signature, il vous suffit d'éditer le fichier **pattern.txt**

C. Utilisation

Pour générer une signature, il suffit de rentrer les informations demandées comme ci-dessous.



Lorsque vous appuyez sur le bouton **Générer**, un cadre contenant un aperçu de la signature suivi du code source sont affichés. Il ne reste plus qu'à appuyer sur le bouton **Copier dans le presse papier** et à coller votre signature dans votre client à l'endroit permettant de la définir.

naos
Nouvelle-Aquitaine
Open Source

Nouvelle-Aquitaine Open Source

Cité Numérique

406, Boulevard Jean-Jacques BOSC 33130 BÈGLES

+33 (0)9 72 38 71 54

contact@naos-cluster.com

naos-cluster.com

RÉGION Nouvelle-Aquitaine