

- [Accueil](#)
- Missions 1 à 5
  - [Mission 1 : Configuration réseau](#)
  - [Mission 2 : GPG](#)
  - [Mission 3 : Clonezilla](#)
  - [Mission 4 : BackupPC](#)
  - [Mission 5 : MariaDB](#)
- Missions 6 à 10
  - [Mission 6 : DHCP](#)
  - [Mission 7 : Failover](#)
  - [Mission 8 : DNS](#)
  - [Mission 9 : Nginx](#)
  - [Mission 10 : FTP](#)
- [Mission 11 : SSL/TLS](#)
- [Mission 12 : OPNsense](#)
- [Mission 13 : Zabbix](#)

## Nginx

### I) Qu'est-ce qu'Nginx ?

## A) Nginx

Nginx (prononcé “engine-x”) est un serveur web open source de haute performance, ainsi qu'un serveur proxy inverse. Il a été conçu pour être léger, rapide et capable de gérer un grand nombre de connexions simultanées. Initialement créé par Igor Sysoev en 2004, Nginx est aujourd'hui largement utilisé pour servir des sites web et des applications web, en concurrence avec d'autres serveurs web tels qu'Apache.

Nginx est un serveur web populaire en raison de sa rapidité, de sa légèreté, de sa flexibilité et de sa capacité à gérer efficacement les charges de trafic importantes. Il est souvent utilisé comme composant essentiel dans l'infrastructure web pour améliorer les performances et la disponibilité des applications en ligne.

## B) Nginx ou Apache ?

Le choix entre Nginx et Apache dépend largement des besoins spécifiques de l'application ou du site web, ainsi que des caractéristiques de performance, de configuration et de fonctionnalités recherchées.

### Utiliser Nginx si :

Haute performance : Nginx excelle dans la gestion d'un grand nombre de connexions simultanées et est réputé pour ses performances élevées, ce qui en fait un bon choix pour les sites web à fort trafic.

Serveur proxy inverse : Nginx est souvent privilégié pour créer un serveur proxy inverse pour équilibrer la charge entre plusieurs serveurs en raison de ses performances et de son efficacité.

Contenu statique : Si le site web délivre principalement du contenu statique (images, CSS, JavaScript), Nginx est particulièrement efficace dans ce domaine.

Réécriture d'URL : Nginx offre une syntaxe claire et puissante pour les fonctionnalités avancées de réécriture d'URL pour personnaliser les liens et les redirections.

Configuration simple : Nginx a une configuration simple et lisible, ce qui facilite la gestion et la compréhension des fichiers de configuration.

### Utiliser Apache si :

Compatibilité avec les modules : Si l'application nécessite des modules spécifiques qui sont mieux pris en charge par Apache, il peut être préférable de choisir Apache. Certains modules peuvent ne pas être disponibles ou avoir une prise en charge limitée dans Nginx.

Hébergement partagé : Apache est souvent utilisé dans des environnements d'hébergement partagé en raison de sa flexibilité et de sa facilité d'utilisation pour les configurations virtuelles.

.htaccess : Apache offre un meilleur support pour l'utilisation des fichiers .htaccess pour la configuration dynamique au niveau du répertoire.

Le choix entre Nginx et Apache dépend donc de la nature spécifique des projet, préférences personnelles, compétences et fonctionnalités requises. Certains sites web utilisent également les deux serveurs en tandem, avec Nginx en amont (en tant que proxy inverse) et Apache en aval pour traiter les requêtes. Cela dépendra de la configuration et des exigences spécifiques des infrastructure web.

## II) Création des serveurs web

Nous clonons dans un premier temps la VM template et lui attribuons une nouvelle adresse IP (10.31.177.80 (srv-web1-1) ou 10.31.178.80 (srv-web2-1)) et un nouveau nom d'hôte dans le DHCP:

```
dhclient -r && dhclient -v
```

```
# Change le nom d'hôte à l'aide d'une commande  
hostnamectl set-hostname srv-web1-1 # Ou srv-web2-1
```

```
# Change le nom d'hôte directement depuis le fichier de configuration  
nano /etc/hosts
```

Nous modifions dans le fichier /etc/hosts le nom d'hôte de la machine et nous nous déconnectons pour actualiser l'affichage du nom.

Nous installons les paquets nécessaires :

```
apt update && apt upgrade  
apt install nginx
```

Nous pouvons maintenant cloner le serveur web pour et lui attribuons une nouvelle adresse IP (10.31.185.80 (srv-web1-2) ou 10.31.185.80 (srv-web2-2)) et un nouveau nom d'hôte dans le DHCP:

```
dhclient -r && dhclient -v
```

```
# Change le nom d'hôte à l'aide d'une commande  
hostnamectl set-hostname srv-web1-2 # Ou srv-web2-2
```

```
# Change le nom d'hôte directement depuis le fichier de configuration  
nano /etc/hosts
```

Nous modifions dans le fichier /etc/hosts le nom d'hôte de la machine et nous nous déconnectons pour actualiser l'affichage du nom.

### III) Dokuwiki

Notre site de documentation sera situé sur notre serveur web privé, de même que notre serveur NextCloud, que nous mettrons en place lors de notre prochaine mission : [Mission 11 : Chiffrement des communications](#) . En effet, la configuration de NextCloud est adaptée pour le protocole HTTPS et très complexe à adapter pour le protocole HTTP.

Nous commençons donc par créer le dossier qui accueillera le DokuWiki :

```
mkdir -p /home/htdocs/gsb.org/wiki/
```

Nous devons également installer les paquets liés à PHP qui permettront au Wiki de fonctionner, ainsi que les paquets nginx :

```
apt install php php-xml php-fpm nginx
```

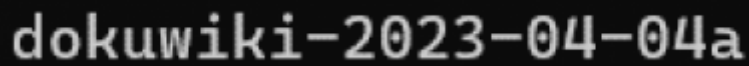
Nous devons maintenant télécharger la dernière archive de DokuWiki :

```
wget https://download.dokuwiki.org/src/dokuwiki/dokuwiki-stable.tgz
```

Nous extrayons l'archive nouvellement téléchargée :

```
tar xf dokuwiki-stable.tgz
```

Nous vérifions la version de notre DokuWiki téléchargé à l'aide de la commande ls.



```
dokuwiki-2023-04-04a
```

Renommons notre dossier :

```
mv dokuwiki-2023-04-04a wiki
```

Nous créons le fichier de configuration DokuWiki /etc/nginx/sites-available/documentation.asie.gsb.org :

```
server {
    server_name documentation.asie.gsb.org;
    listen 80;
    listen [::]:80;
    autoindex off;
    #client_max_body_size 15M;
    #client_body_buffer_size 128k;
    index index.html index.htm index.php doku.php;
    root /home/htdocs/gsb.org/wiki;

    location / {
        try_files $uri $uri/ @dokuwiki;
    }

    location ~ ^/lib.*\.(gif|png|ico|jpg)$ {
        expires 30d;
    }

    location = /robots.txt { access_log off; log_not_found off; }
    location = /favicon.ico { access_log off; log_not_found off; }
    location ~ /\. { access_log off; log_not_found off; }
    deny all; }
    location ~ ~$ { access_log off; log_not_found off; }
    deny all; }

    location @dokuwiki {
        rewrite ^/_media/(.*) /lib/exe/fetch.php?media=$1
    }
    last;
```

```
        rewrite ^/_detail/(.*) /lib/exe/detail.php?media=$1
last;
        rewrite ^/_export/([^\/]+)/(.*)
/doku.php?do=export_$1&id=$2 last;
        rewrite ^/(.*) /doku.php?id=$1 last;
    }

    location ~ /\.php$ {
        try_files $uri =404;
        fastcgi_pass    unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_index    index.php;
        fastcgi_param    SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include /etc/nginx/fastcgi_params;
        fastcgi_param    QUERY_STRING        $query_string;
        fastcgi_param    REQUEST_METHOD      $request_method;
        fastcgi_param    CONTENT_TYPE        $content_type;
        fastcgi_param    CONTENT_LENGTH      $content_length;
        fastcgi_intercept_errors             on;
        fastcgi_ignore_client_abort         off;
        fastcgi_connect_timeout 60;
        fastcgi_send_timeout 180;
        fastcgi_read_timeout 180;
        fastcgi_buffer_size 128k;
        fastcgi_buffers 4 256k;
        fastcgi_busy_buffers_size 256k;
        fastcgi_temp_file_write_size 256k;
    }

    location ~ /(data|conf|bin|inc)/ {
        deny all;
    }

    location ~ /\.ht {
        deny all;
    }

}
```

Nous créons maintenant notre lien symbolique permettant à Nginx d'activer le fichier de configuration lié :

```
ln -s /etc/nginx/sites-available/documentation.asie.gsb.org
/etc/nginx/sites-enable/documentation.asie.gsb.org
```

Nous redémarrons le service Nginx :

```
systemctl restart nginx
```

Nous pouvons dès à présent accéder à l'interface de configuration de DokuWiki à l'adresse [documentation.asie.gsb.org](https://documentation.asie.gsb.org)



## DokuWiki Installer

Wiki Name

☒ Enable ACL (recommended)

Superuser

Real name

E-Mail

Password

once again

Initial ACL policy

☐ Allow users to register themselves

Please choose the license you want to put your content under:

☐ CC0 1.0 Universal [\[?\]](#)

☐ Public Domain [\[?\]](#)

☐ CC Attribution 4.0 International [\[?\]](#)

☒ CC Attribution-Share Alike 4.0 International [\[?\]](#)

☐ GNU Free Documentation License 1.3 [\[?\]](#)

☐ CC Attribution-Noncommercial 4.0 International [\[?\]](#)

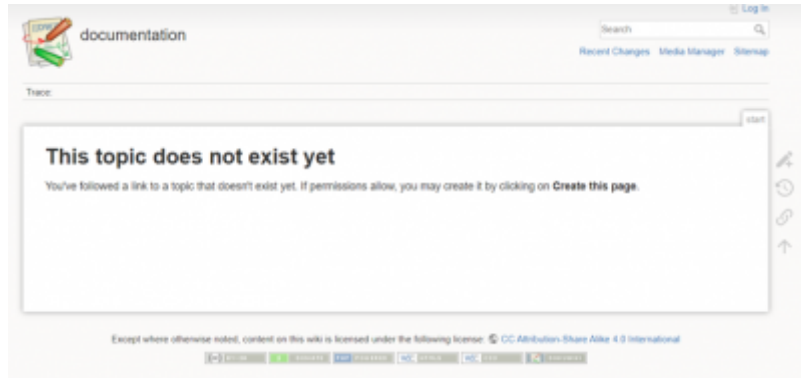
☐ CC Attribution-Noncommercial-Share Alike 4.0 International [\[?\]](#)

☐ Do not show any license information

Please, help us to improve the DokuWiki experience:

☒ Once a month, send anonymous usage data to the DokuWiki developers [\[?\]](#)

Après avoir configuré notre DokuWiki, nous pouvons accéder à notre page d'accueil (qui n'existe pas encore) :



## IV) Wordpress

# A) Configuration de la base de données

Wordpress est un CMS nécessitant une base de données. Nous allons donc commencer par configurer cette base de données. Sur notre machine virtuelle hébergeant nos bases de données (10.31.177.33 ou 10.31.178.33), nous commençons par nous connecter sur l'interface MariaDB :

```
mysql -u root -p
```

Nous devons créer deux bases de données possédant des comptes distincts. Nous créons la base de données liée au site [www.gsb.org](http://www.gsb.org) :

```
CREATE DATABASE wordpress;  
CREATE USER 'adminwp'@'%' identified by 'password';  
GRANT ALL PRIVILEGES ON wordpress.* TO 'adminwp'@'%;  
FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE USER 'adminwp'@'%' identified by 'password';  
Query OK, 0 rows affected (0,003 sec)  
  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'adminwp'@'%;  
Query OK, 0 rows affected (0,002 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0,000 sec)
```

Nous créons la base de données liée au site [www.asie.gsb.org](http://www.asie.gsb.org) :

```
CREATE DATABASE wordpressasie;  
CREATE USER 'adminwpasie'@'%' identified by 'password';  
GRANT ALL PRIVILEGES ON wordpressasie.* TO 'adminwpasie'@'%;  
FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE DATABASE wordpresssie;  
Query OK, 1 row affected (0,000 sec)  
  
MariaDB [(none)]> CREATE USER 'adminwpasie'@'%' identified by 'password';  
Query OK, 0 rows affected (0,001 sec)  
  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpresssie.* TO 'adminwpasie'@'%';  
Query OK, 0 rows affected (0,001 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0,000 sec)
```

## B) Installation et configuration de Wordpress

Nos sites web Wordpress se situeront sur notre site web public.

Nous téléchargeons les paquets nécessaires pour le fonctionnement de Wordpress :

```
apt update && apt upgrade  
apt install php-fpm
```

Nous devons à présent créer les répertoires qui accueilleront nos nouveaux sites internet :

```
# Création du dossier accueillant le site www.asie.gsb.org  
mkdir -p /home/htdocs/gsb.org/asie/  
# Création du dossier accueillant le site www.gsb.org  
mkdir -p /home/htdocs/gsb.org/gsb/
```

Nous nous déplaçons dans le dossier dont nous voulons créer le site et téléchargeons la dernière version de Wordpress :

```
wget https://wordpress.org/latest.tar.gz
```

```
--2023-11-10 14:04:46-- https://wordpress.org/latest.tar.gz  
Résolution de wordpress.org (wordpress.org)... 198.143.164.252  
Connexion à wordpress.org (wordpress.org)[198.143.164.252]:443_ connecté.  
requête HTTP transmise, en attente de la réponse... 200 OK  
Taille : 24479162 (23M) [application/octet-stream]  
Sauvegarde en : « latest.tar.gz »  
  
latest.tar.gz      18%[==>] 4,39M 61,3KB/s tps 5m 15s |
```

Nous extrayons l'archive Wordpress téléchargée précédemment :

```
tar -xvzf latest.tar.gz
```

Nous nous déplaçons dans le dossier créé nommé wordpress et changeons le nom du fichier



exemple de configuration :

```
cd wordpress
cp wp-config-sample.php wp-config.php
```

Nous devons maintenant modifier des éléments de notre fichier de configuration. Pour ce faire, nous modifions le fichier wp-config.php :

```
nano wp-config.php
```

Nous devons rajouter dans les endroits prévus à cet effet le nom de notre base de données créée précédemment, le nom de l'utilisateur de cette base de données, son mot de passe ainsi que l'adresse IP de la machine sur laquelle est stockée la base de données.

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'adminwp' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
define( 'DB_HOST', '10.31.178.33' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

fichier de configuration /home/htdocs/gsb.org/asiel/wordpress/wp-config.php

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'adminwp' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
define( 'DB_HOST', '10.31.178.33' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

fichier de configuration /home/htdocs/gsb.org/gsb/wordpress/wp-config.php

A présent, nous devons créer les fichiers de configuration Nginx et activer notre site internet. Commençons tout d'abord par créer les fichiers /etc/nginx/sites-available/www.asiel.gsb.org.conf et /etc/nginx/sites-available/asiel.gsb.org.conf :

```
server {
    server_name www.asie.gsb.org;
    listen 80;
    listen [::]:80;
    autoindex off;
    #client_max_body_size 15M;
    #client_body_buffer_size 128k;
    index index.html index.htm index.php doku.php;
    root /home/htdocs/gsb.org/asie/;

    location / {
        try_files $uri $uri/ @dokuwiki;
    }

    location ~ ^/lib.*\.(gif|png|ico|jpg)$ {
        expires 30d;
    }

    location = /robots.txt { access_log off; log_not_found off; }
    location = /favicon.ico { access_log off; log_not_found off; }
    location ~ /\. { access_log off; log_not_found off; }
    deny all; }
    location ~ ~$ { access_log off; log_not_found off; }
    deny all; }

    location @dokuwiki {
        rewrite ^/_media/(.*) /lib/exe/fetch.php?media=$1
    last;

        rewrite ^/_detail/(.*) /lib/exe/detail.php?media=$1
    last;

        rewrite ^/_export/([^/]+)/(.*)
/doku.php?do=export_$1&id=$2 last;
        rewrite ^/(.*) /doku.php?id=$1 last;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include /etc/nginx/fastcgi_params;
        fastcgi_param QUERY_STRING $query_string;
        fastcgi_param REQUEST_METHOD $request_method;
        fastcgi_param CONTENT_TYPE $content_type;
        fastcgi_param CONTENT_LENGTH $content_length;
        fastcgi_intercept_errors on;
        fastcgi_ignore_client_abort off;
        fastcgi_connect_timeout 60;
        fastcgi_send_timeout 180;
        fastcgi_read_timeout 180;
```

```
        fastcgi_buffer_size 128k;
        fastcgi_buffers 4 256k;
        fastcgi_busy_buffers_size 256k;
        fastcgi_temp_file_write_size 256k;
    }

    location ~ /(data|conf|bin|inc)/ {
        deny all;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Nous créons maintenant un lien symbolique entre ce fichier de configuration et le dossier /etc/nginx/sites-enabled/ :

```
ln -s /etc/nginx/sites-available/www.asie.gsb.org.conf
/etc/nginx/sites-enabled/www.asie.gsb.org.conf
ln -s /etc/nginx/sites-available/www.gsb.org.conf /etc/nginx/sites-
enable/www.gsb.org.conf
```

Nous redémarrons notre service afin que les changements soient effectifs :

```
systemctl restart nginx
```

Nous pouvons maintenant accéder à l'interface graphique de Wordpress via nos adresses web [www.asie.gsb.org](http://www.asie.gsb.org) et [www.gsb.org](http://www.gsb.org). Nous commençons par sélectionner la langue de l'interface.



Español de Ecuador  
Español de República Dominicana  
Español de Perú  
Español de Uruguay  
Español de Chile  
Español de Puerto Rico  
Español de Guatemala  
Eesti  
Euskara  
فارسی  
فارسی (افغانستان)  
Suomi  
**Français**  
Français du Canada  
Français de Belgique  
Friulian  
Frysk  
Gàidhlig  
Galego  
ગુજરાતી  
هزاره گی  
עברית

[Continuer](#)

Nous configurons maintenant notre Wordpress en entrant le titre du site, l'identifiant de l'utilisateur, son mot de passe ainsi que son e-mail. Nous devons également cocher l'option "Confirmer l'utilisation du mot de passe faible" étant donné que notre mot de passe "password" n'est pas sécurisé.

### Informations nécessaires

Veuillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.

Titre du site

asie

Identifiant

std

Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ["\_"], des traits d'union ["-"], des points et le symbole @.

Mot de passe

password

Very weak

Masquer

Important :

Vous aurez besoin de ce mot de passe pour vous connecter. Pensez à le stocker dans un lieu sûr.

Confirmer le mot de passe

☒ Confirmer l'utilisation du mot de passe faible

Votre e-mail

exemple@gmail.com

Vérifiez bien cette adresse e-mail avant de continuer.

Visibilité par les moteurs de recherche

☐ Demander aux moteurs de recherche de ne pas indexer ce site

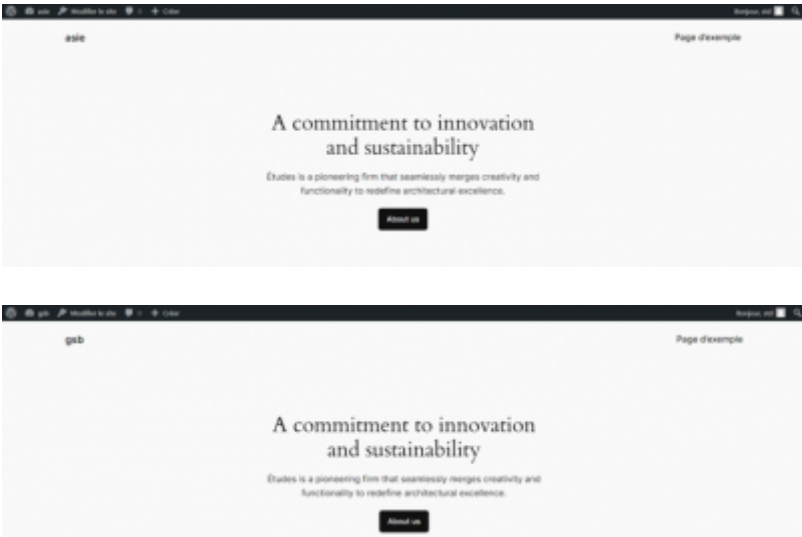
Certains moteurs de recherche peuvent décider de l'indexer malgré tout.

Installer WordPress

Notre site Wordpress est maintenant configuré.



Nous appliquons ces mêmes étapes pour configurer notre 2e site Wordpress.



## V) .htaccess

Nous devons mettre en place une interface de connexion pour les espaces utilisateurs. Nous commençons par créer le dossier /home/std/public\_html qui servira de dossier utilisateur accessible en ligne, et lui attribuons les droits nécessaires :

```
mkdir /home/std/public_html
chmod a+x /home/std
```

Cependant, contrairement à Apache, nous pouvons configurer directement les différentes directives dans le fichier de configuration du Virtual Host. Nous insérons donc dans chaque fichier de configuration un nouveau bloc "location" permettant de définir le dossier /home/std/public\_html :

```
location ~ ^/~(.*?)(/.*)?$ {
    alias /home/$1/public_html$2;
    index index.html index.htm;
    autoindex on;
    auth_basic "Zone sécurisée - Authentification requise";
    auth_basic_user_file /etc/nginx/.htpasswd;
}
```

```
location ~ ^/~(.*?)(/.*)?$ {
    alias /home/$1/public_html$2;
    index index.html index.htm;
    autoindex on;
    auth_basic "Zone sécurisée - Authentification requise";
    auth_basic_user_file /etc/nginx/.htpasswd;
}
```

Nous devons maintenant créer un fichier htpasswd. Pour ce faire, nous commençons tout d'abord par télécharger le paquet apache2-utils :

```
apt update && apt upgrade
apt install apache2-utils
```

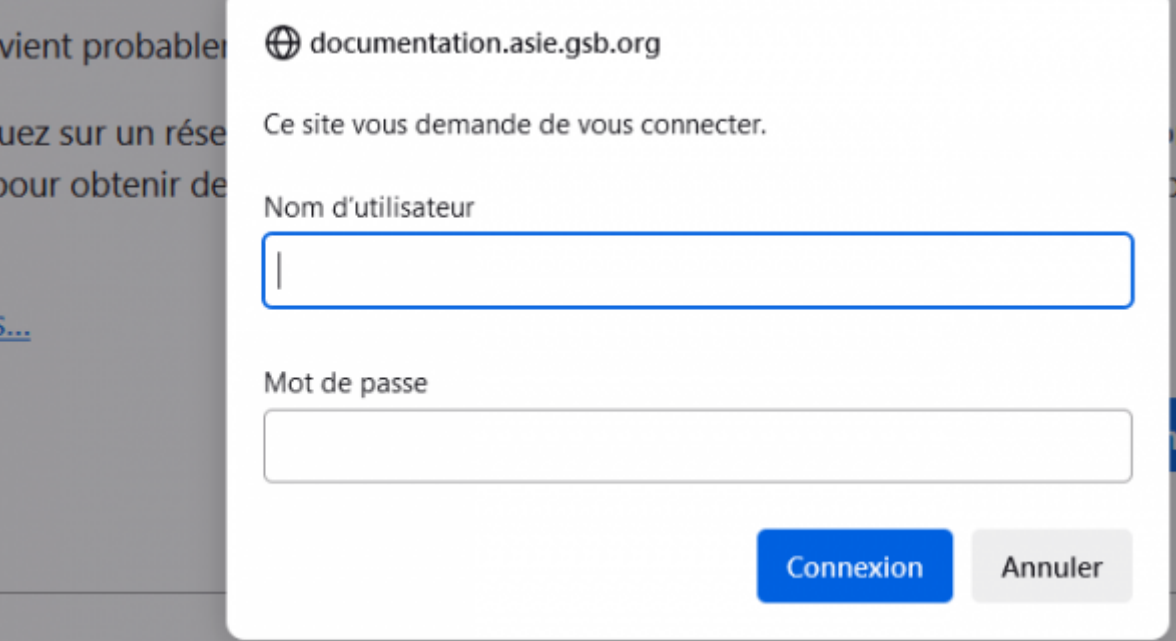
Le fichier htpasswd contient les identifiants et mots de passes des utilisateurs. Nous devons créer l'utilisateur "test" qui aura pour mot de passe "password" :

```
htpasswd -c /etc/nginx/.htpasswd test
```

```
root@srv-web2-1:~# htpasswd -c /etc/nginx/.htpasswd test
New password:
Re-type new password:
Adding password for user test
```

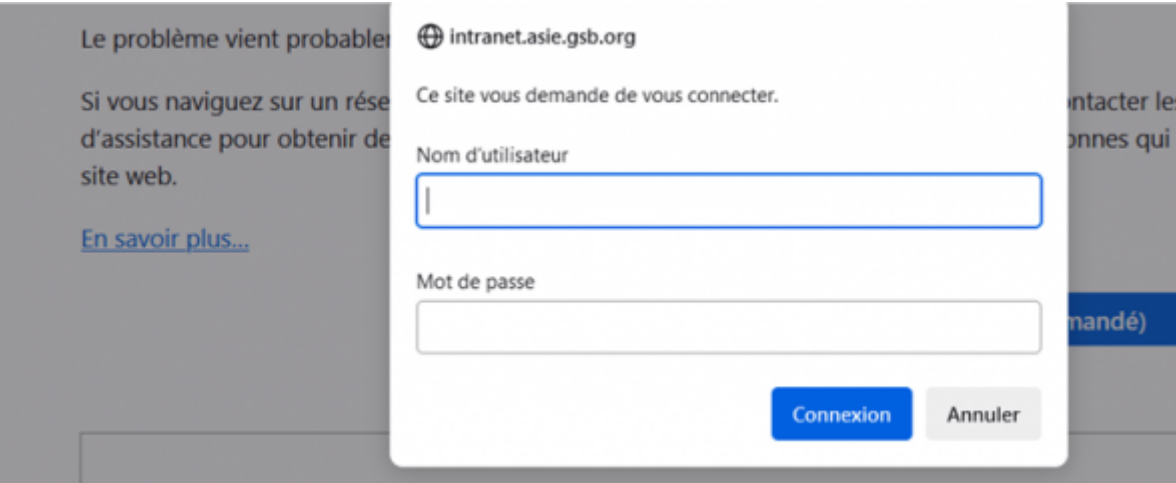
Nous vérifions maintenant que chaque site est protégé :

documentation.asie.gsb.org :



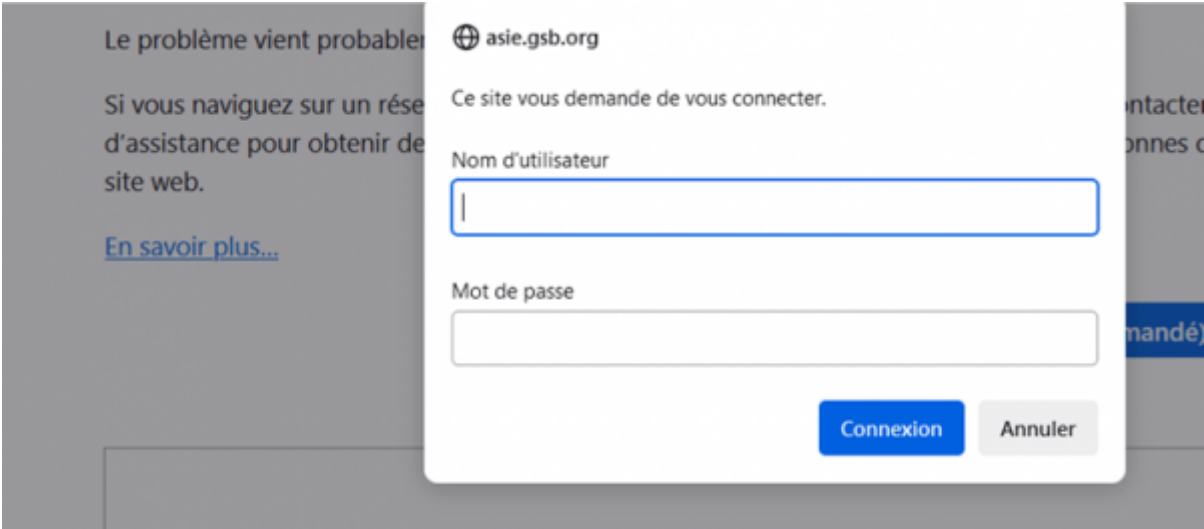
The screenshot shows a login modal for the website documentation.asie.gsb.org. The modal has a title bar with the site's icon and name. Below the title, it says "Ce site vous demande de vous connecter." (This site asks you to connect). There are two input fields: "Nom d'utilisateur" (Username) and "Mot de passe" (Password). At the bottom right, there are two buttons: "Connexion" (Login) in blue and "Annuler" (Cancel) in grey. The background of the page is partially visible, showing some text about a problem and a link "En savoir plus...".

intranet.asie.gsb.org :

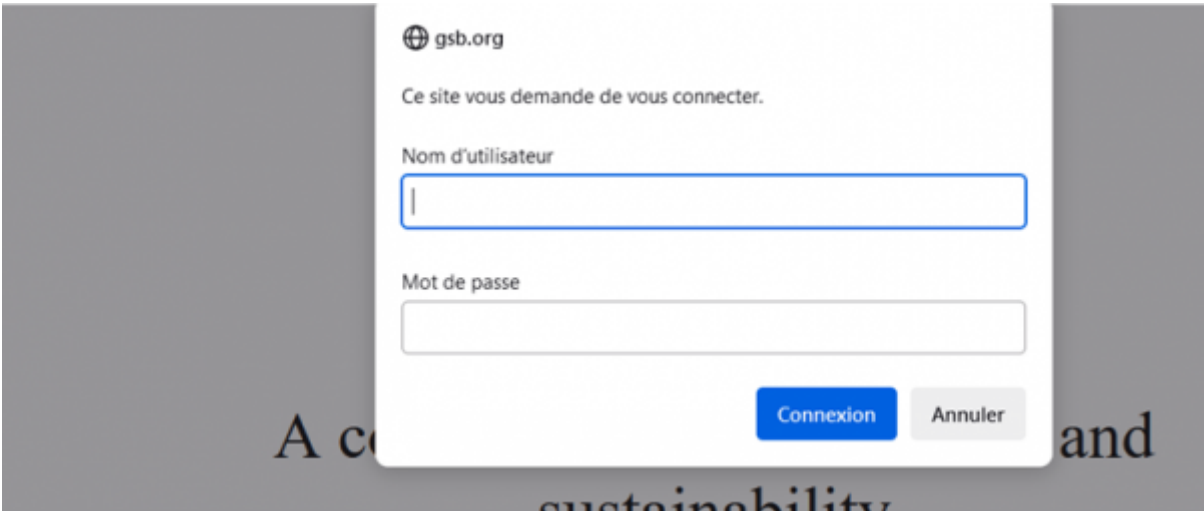


The screenshot shows a login modal for the website intranet.asie.gsb.org. The modal has a title bar with the site's icon and name. Below the title, it says "Ce site vous demande de vous connecter." (This site asks you to connect). There are two input fields: "Nom d'utilisateur" (Username) and "Mot de passe" (Password). At the bottom right, there are two buttons: "Connexion" (Login) in blue and "Annuler" (Cancel) in grey. The background of the page is partially visible, showing some text about a problem and a link "En savoir plus...".

[www.asie.gsb.org](http://www.asie.gsb.org) :



[www.gsb.org](http://www.gsb.org) :



From:  
<https://sisr2.beaupeyrat.com/> - **Documentations SIO2 option SISR**

Permanent link:  
<https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie:mission9>

Last update: **2023/12/15 09:38**

