

Mode Projet : Site vitrine

Installation de l'os avec une clé bootable Ventoi	2
Nommage machine / attribution ip	3
Création du nom de domaine sur le DNS interne	4
Réservation de l'adresse IP sur le serveur DHCP	5
Installation des paquets pour SSH et outils habituels du système	6
Positionnement dans la baie	7
Créer un utilisateur classique et un utilisateur root pour chaque membre du projet	8
Installation et configuration Fail2ban	9
Installation et configuration ClamAV	10
Installation et configuration Hk Hunter	11
Installation et configuration Apache et PHP	12
Création d'un virtual host	13
Installation et configuration Mysql	14
Installation et configuration Phpmyadmin	15
Installation de Wordpress	16
Mise en place des backup de la machine	17
Mise en place d'un formulaire de contact	18
Intégration Captcha	19
RGPD	20
Mise en place des cookies	21
Mise en place des backup de Wordpress	22
Création du site	23
Tests de fonctionnement	24

TABLEAU DES TACHES A REALISER

DUREE CHEMIN CRITIQUE : 30 UNITES		TACHES CHEMIN CRITIQUE : A, B,C, G, M, N, O, P, S, T			
TACHES	RESSOURCES	DESCRIPTION	TACHES ANTERIEURES	DUREE ESTIMEE (unité 15 min)	TABLE DES MARGES (unité 15mn)
A	Rayan	Choix de la machine + Installation de l'os avec une clé bootable Ventoi	—	3	0
B	Rayan	Installation des paquets pour SSH et outils habituels du système	A	1	0
C	Tom	Nommage machine / attribution IP	B	1	0
D	Benjamin	Création du nom de domaine sur le DNS interne	—	1	12
E	Maxime	Réservation de l'adresse IP sur le serveur DHCP	A	2	23
F	Hugo	Positionnement dans la baie	C	1	22
G	Rayan	Créer un utilisateur et un utilisateur appartenent au groupe root pour chaque membre du projet	C	2	0
H	Tom	Installation et configuration Fail2ban	G	2	19
I	Rayan	Installation et configuration ClamAV	G	1	19
J	Benjamin	Installation et configuration Rk Hunter	G	2	19
K	Hugo	Installation et configuration Apache et PHP	G	2	1
L	Benjamin	Création d'un virtual host	K	1	1
M	Maxime	Installation et configuration Mysql	G	4	0
N	Maxime	Installation et configuration Phpmyadmin	L M	2	0
O	Hugo	Installation de Wordpress	D N	2	0
P	Hugo /TOM	Création du site à partir des éléments existant	O	10	0
Q	Tom	Mise en place d'un formulaire de contact	P	2	1
R	Rayan	Intégration Captcha	P	1	2
S	Benjamin	Mise en place des cookies + RGPD	P	3	0
T	Hugo	Tests de fonctionnement + Mise en production	E F H I J Q R S	2	0
DUREE CHEMIN CRITIQUE : 30 UNITES		TACHES CHEMIN CRITIQUE : A, B,C, G, M, N, O, P, S, T			

Récupération de la machine

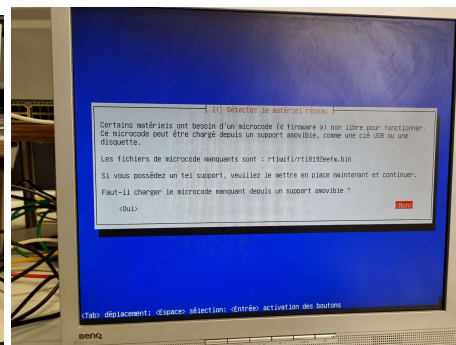
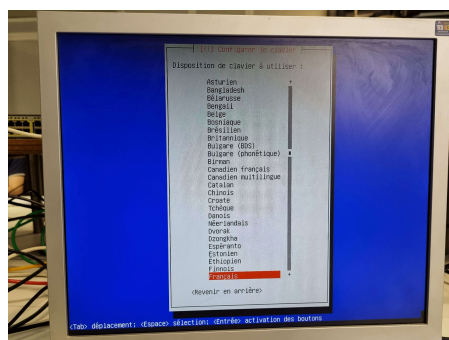
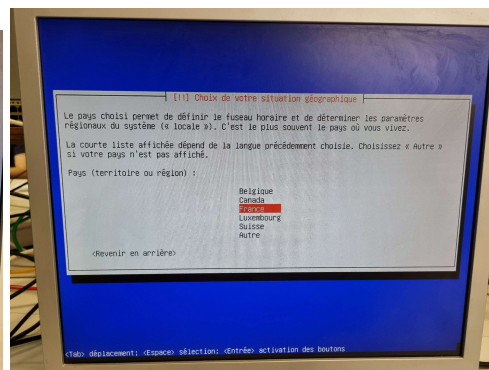
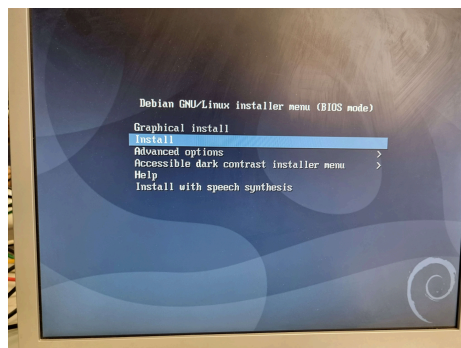
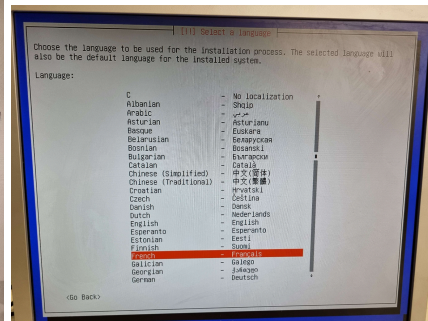
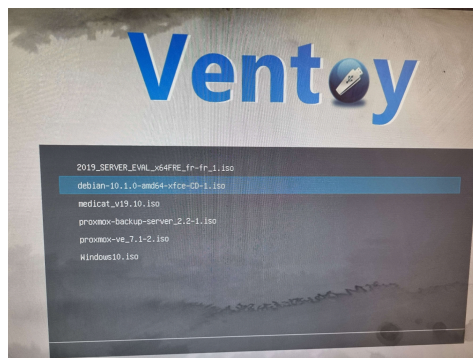


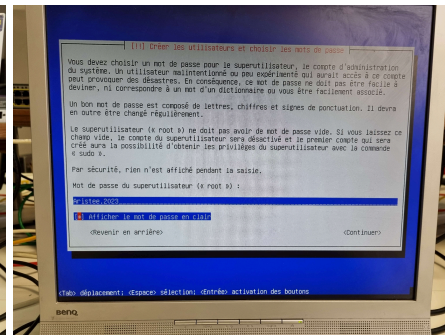
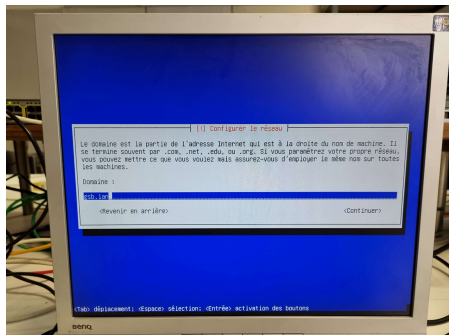
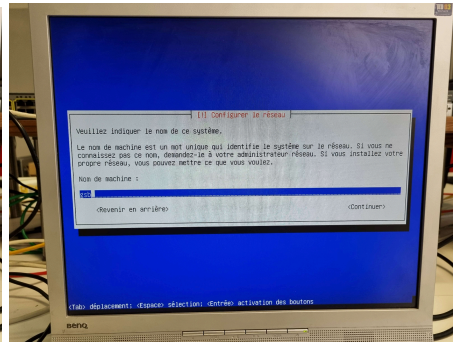
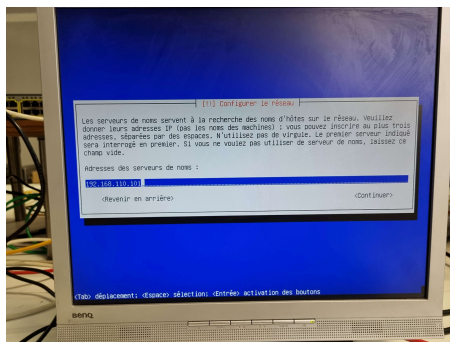
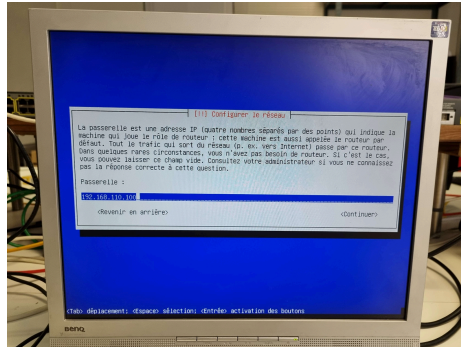
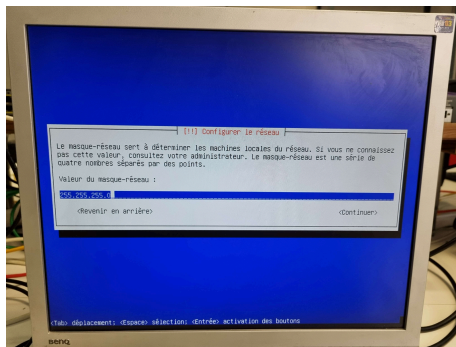
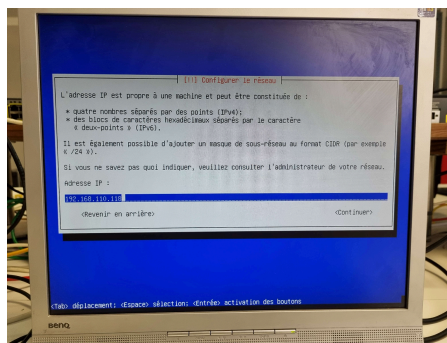
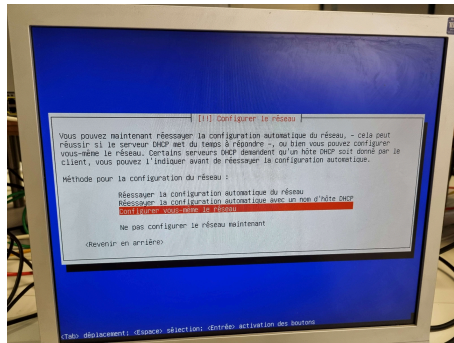
Temps de recherche : 3 jours.

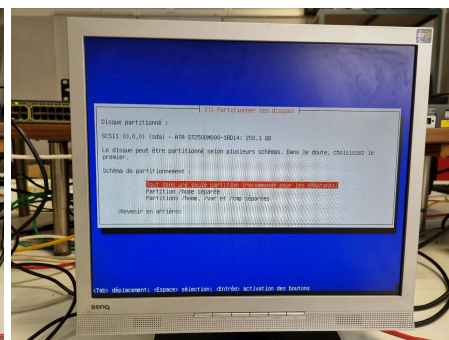
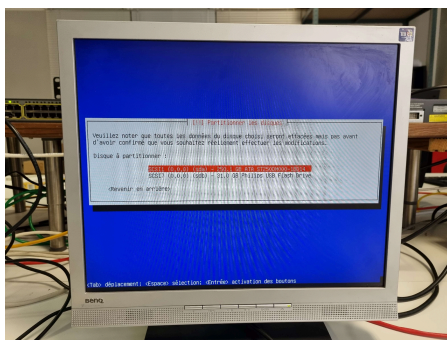
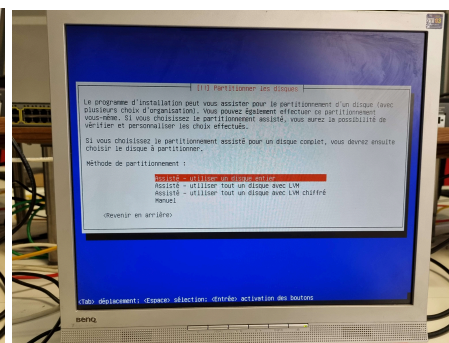
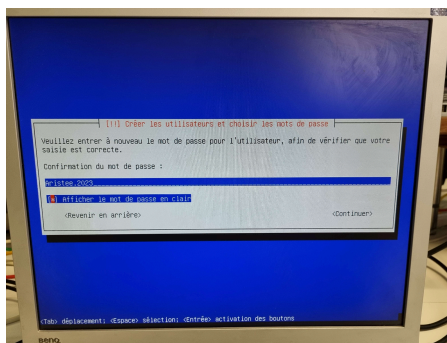
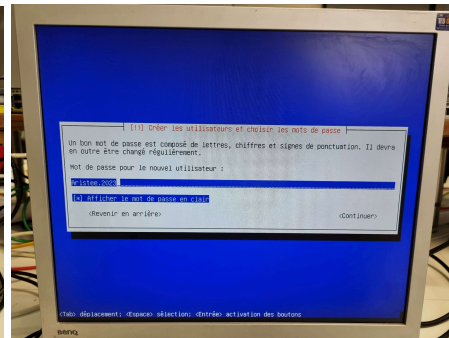
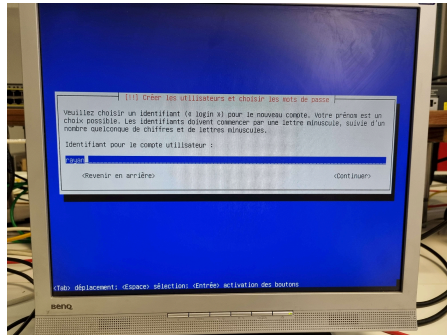
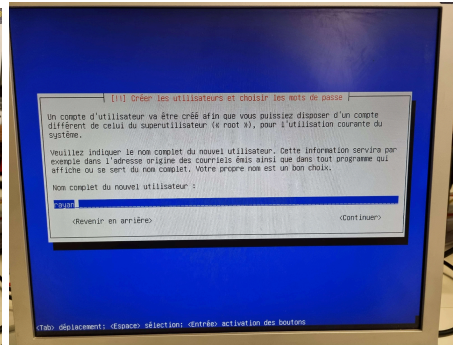
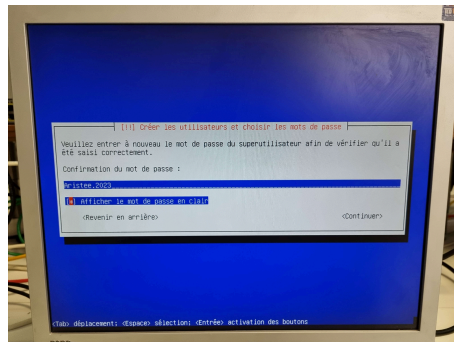
Temps d'installation : 1 jour.

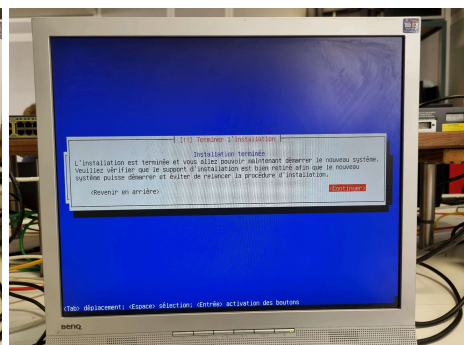
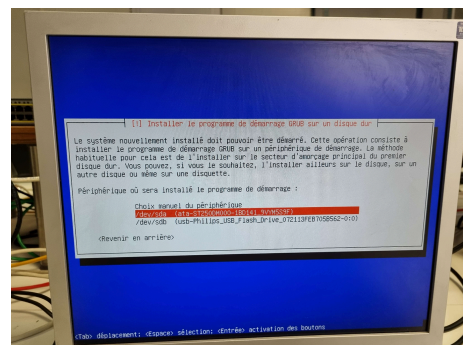
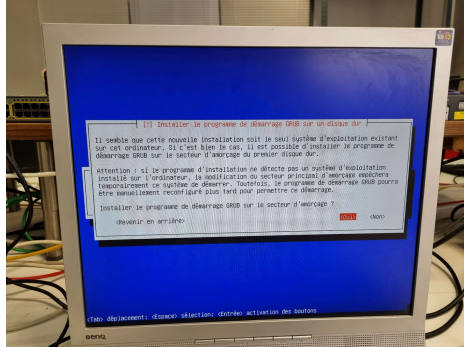
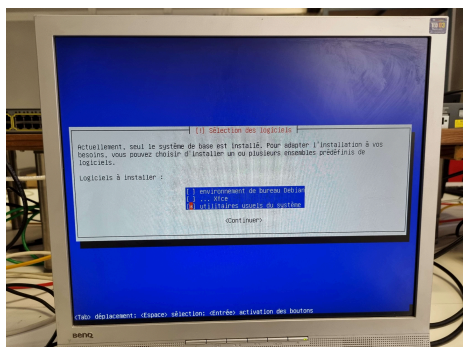
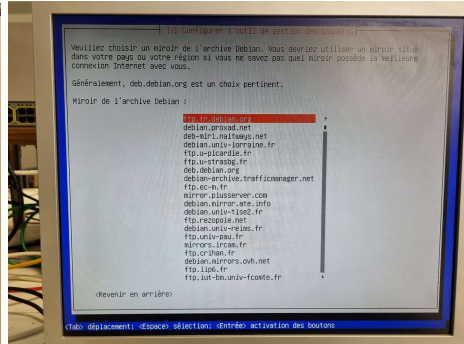
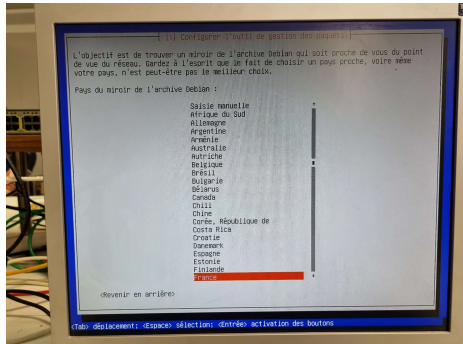
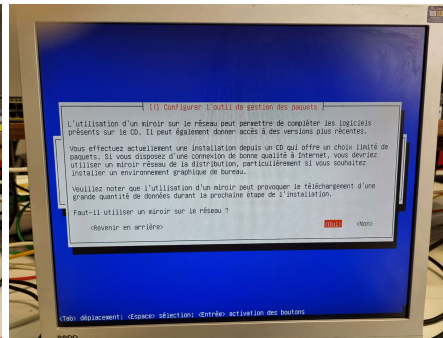
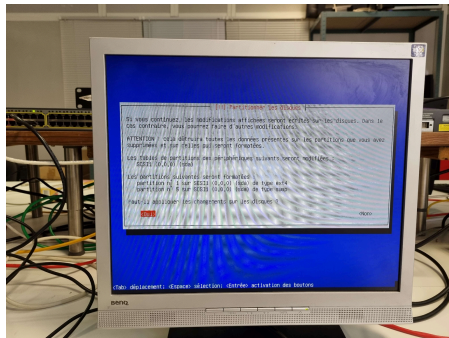
Installation et Configuration de l'OS avec une clé bootable Ventoy

Sélectionner debian 10.1.0
Et suivre les images ci-dessous.











Nommage machine + Configuration Adressage IP

Nom de la machine : gsb

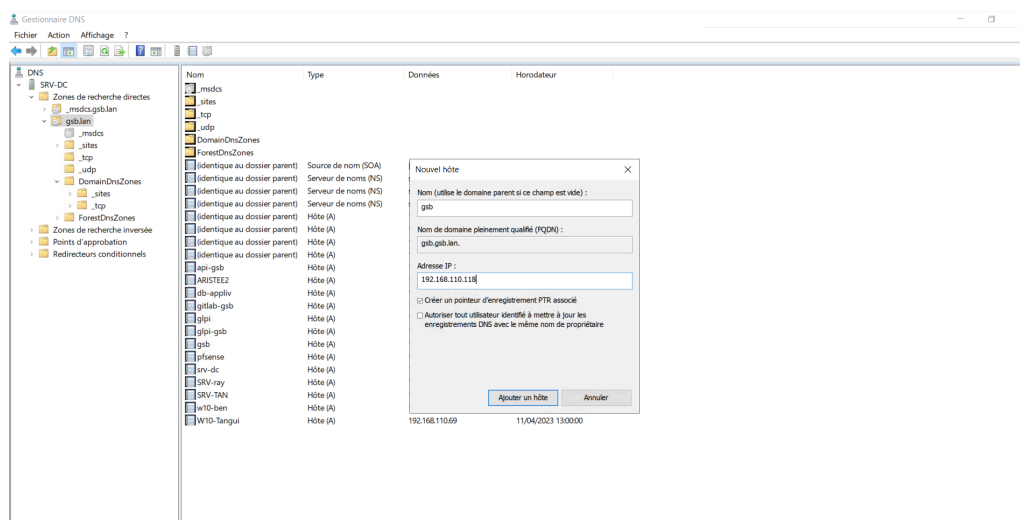
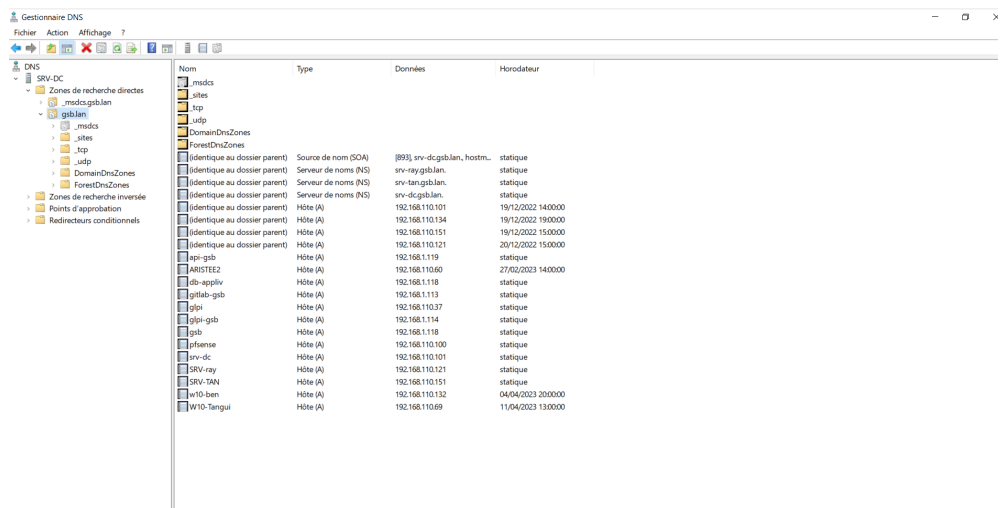
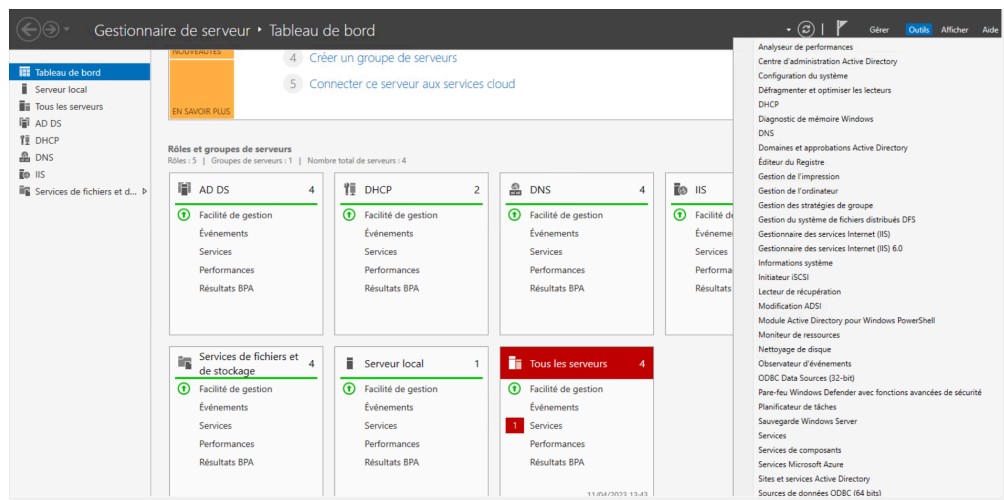
Adresse IP : 192.168.110.118

(voir section "[Installation et Configuration de l'OS avec une clé bootable Ventoy](#)") (page : X)

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

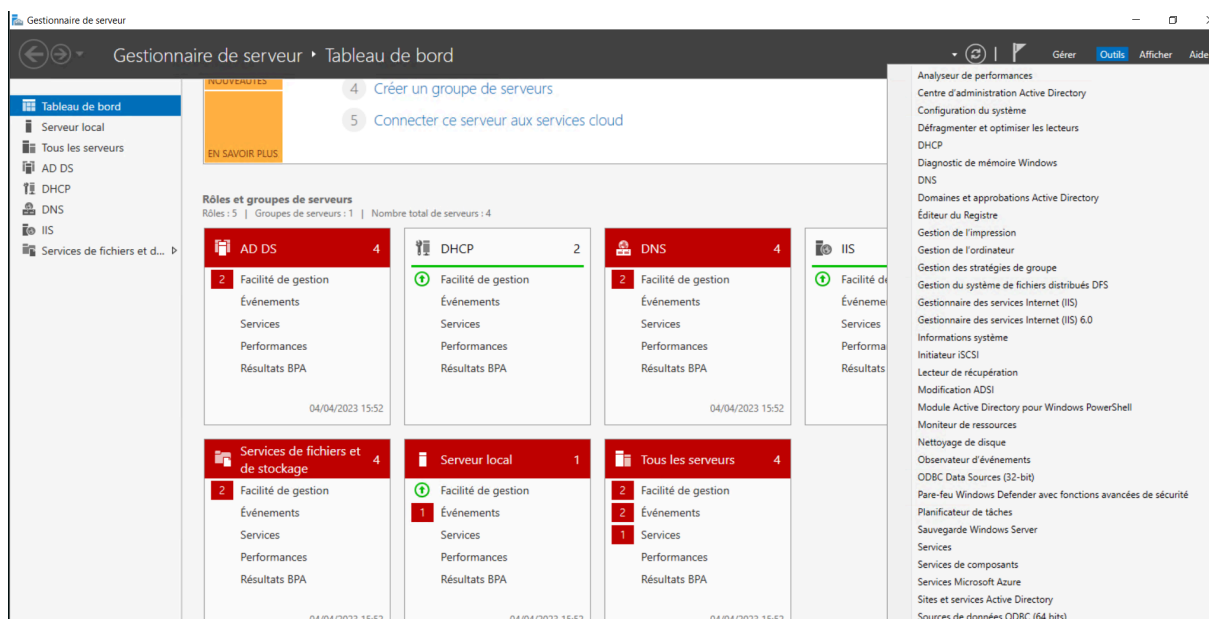
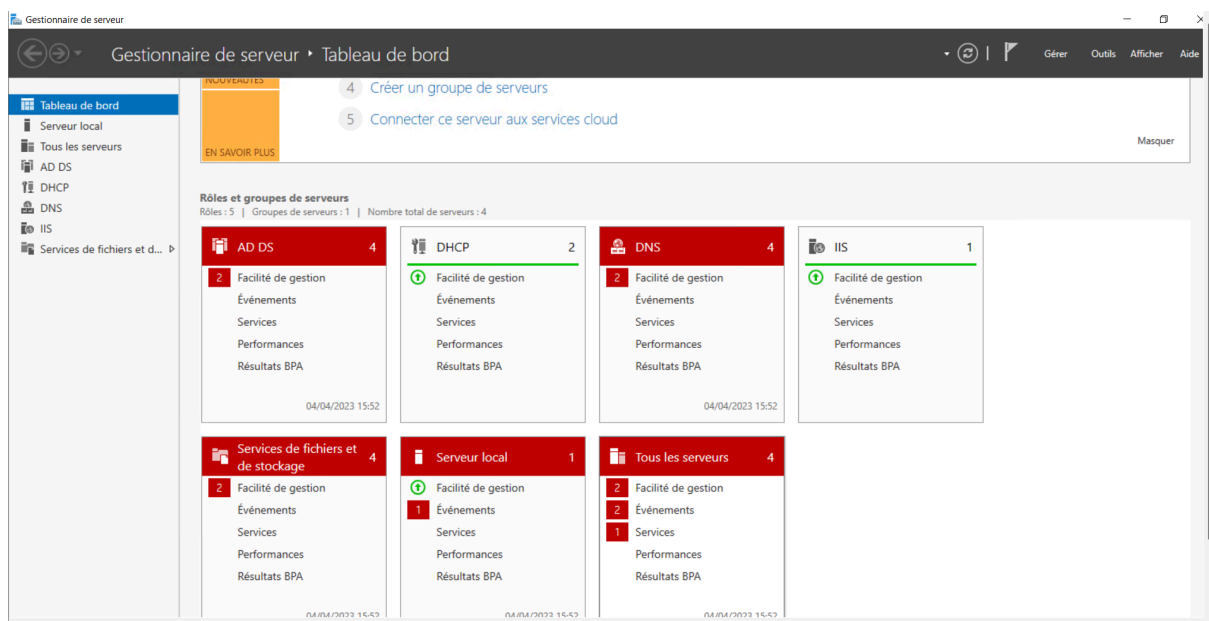
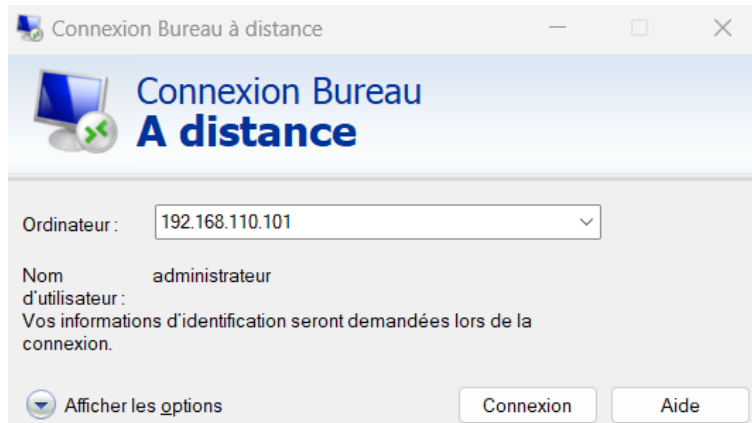
Création du nom de domaine sur le DNS interne

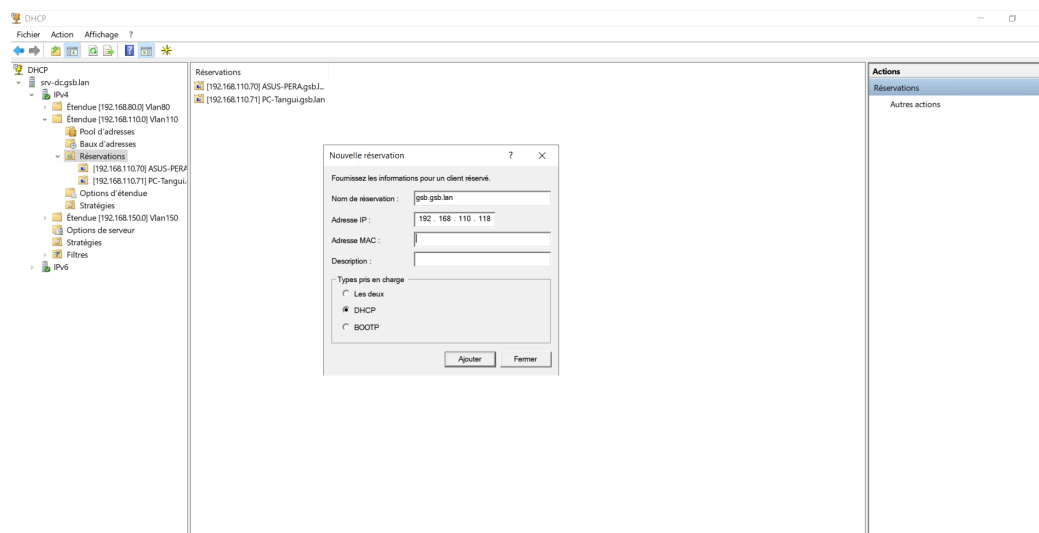
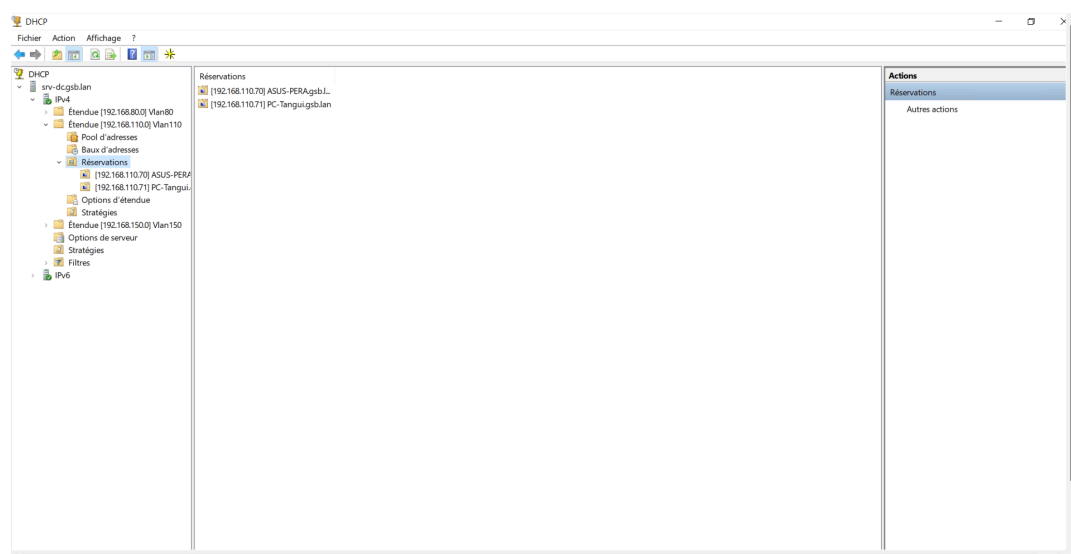
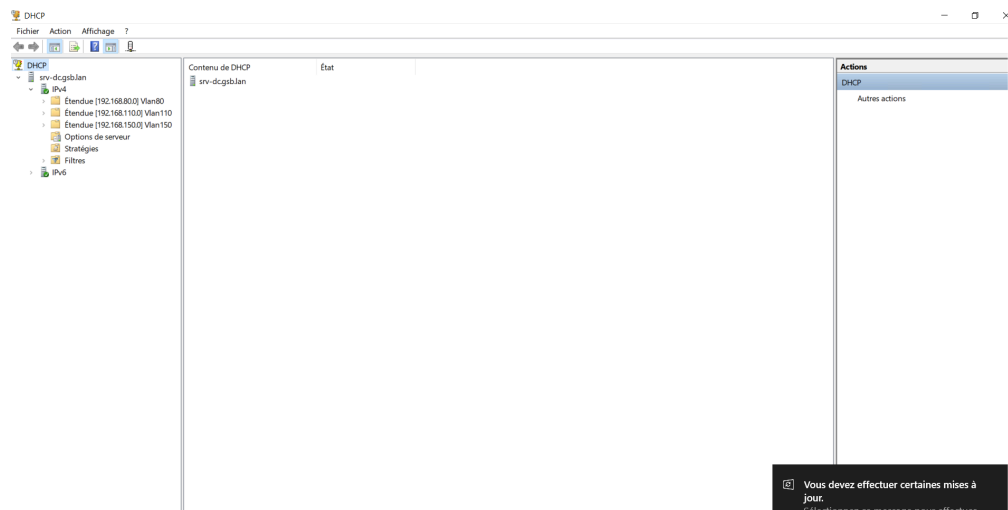


Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Réservation de l'adresse IP sur le serveur DHCP





Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation des paquets pour SSH et outils habituels du système

Il faudra simplement taper la commande suivante :

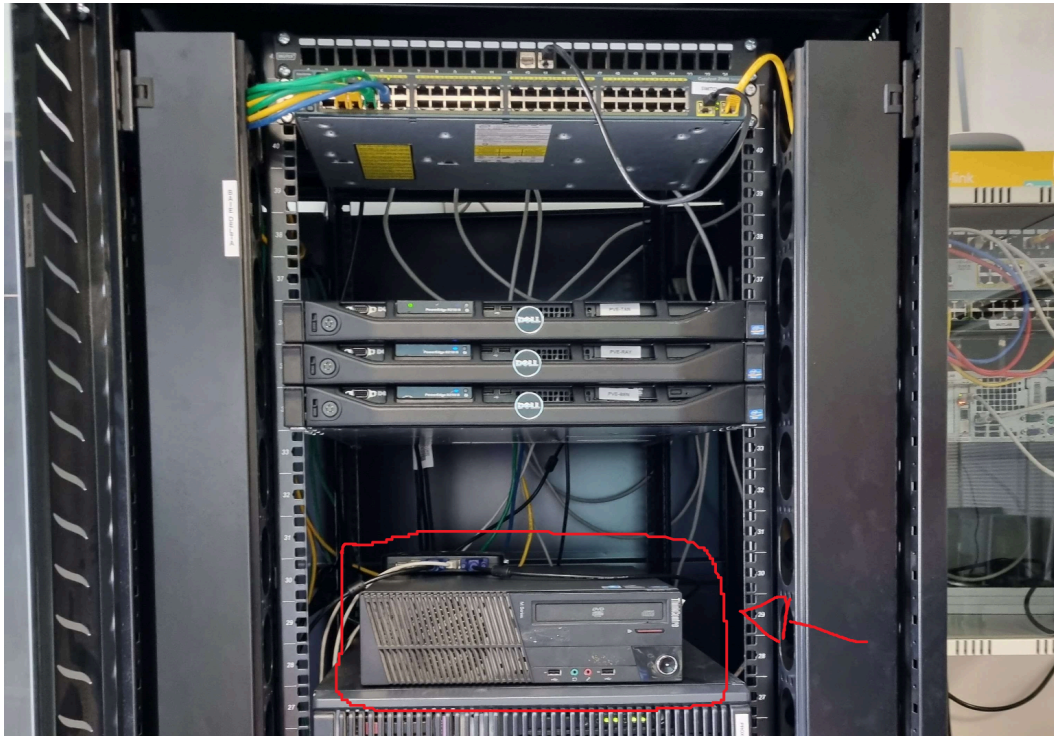
- apt install ssh

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Positionnement dans la baie de la machine debian

Installation du serveur physique dans la baie.



Connexion du serveur au réseau GSB-DELTA



Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Créer un utilisateur classique et un utilisateur root pour chaque membre du projet

Users : hugo / maxime / ben / tanguy / tom / root / rayan

Mot de passe : Aristee.2023

(voir section "Installation et Configuration de l'OS avec une clé bootable Ventoy") (page : X)

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation et configuration Fail2ban

Installation de Fail2ban :

Connexion à la machine distante avec un utilisateur; élever ses privilèges en root avec la commande “su -” et le mot de passe root

Installation de fail2ban avec la commande “*apt install ...*”

```
tom@gsb:~$ su -  
Mot de passe :  
root@gsb:~# apt install fail2ban
```

Configuration du fichier fail2ban.conf :

Ajout de “ignoreip” qui va nous permettre de whitelister les ip qu’on ne veut pas bannir.

```
# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban  
# will not ban a host which matches an address in this list. Several addresses  
# can be defined using space (and/or comma) separator.  
ignoreip = 127.0.0.1/8 ::1 192.168.110.0/24
```

Ajout de “bantime” qui va nous permettre de déterminer le temps de bannissement

Ajout de “findtime” qui va permettre d’aller vérifier si le nombre de maxretry a été atteint durant l’intervalle de temps défini.

Ajout de “maxretry” qui va permettre de déterminer un nombre d’essai maximum

```
# "bantime" is the number of seconds that a host is banned.  
bantime = 300  
  
# A host is banned if it has generated "maxretry" during the last "findtime"  
# seconds.  
findtime = 600  
  
# "maxretry" is the number of failures before a host get banned.  
maxretry = 3
```

Configuration du fichier fail2ban.local :

Création de fail2ban pour le service sshd

On a mis “enabled” à “True” pour que le service sshd soit pris en compte et que nos paramètres aussi.

On a mis le “port 22” qui est le port ssh, et on a choisis un “bantime” et un “findtime”

```
[sshd]  
enabled = true  
port = 22  
bantime = 300  
findtime = 600
```

Test du service Fail2ban :

Je teste si la jail que l'on vient de créer marche en regardant dans les log de fail2ban

```
2023-04-13 09:55:57,527 fail2ban.actions      [6304]: NOTICE [sshd] Unban 192.168.110.61
2023-04-13 09:56:34,854 fail2ban.filter      [6304]: INFO     [sshd] Found 192.168.110.61 - 2023-04-13 09:56:34
2023-04-13 09:56:36,764 fail2ban.filter      [6304]: INFO     [sshd] Found 192.168.110.61 - 2023-04-13 09:56:36
2023-04-13 09:56:40,231 fail2ban.filter      [6304]: INFO     [sshd] Found 192.168.110.61 - 2023-04-13 09:56:40
2023-04-13 09:56:40,809 fail2ban.actions      [6304]: NOTICE [sshd] Ban 192.168.110.61
```

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation et configuration ClamAV

Installation de ClamAV:

Connexion à la machine distante avec un utilisateur; élever ses privilèges en root avec la commande “su -” et le mot de passe root.

Installation de fail2ban avec la commande “*apt install ...*”.

```
root@gsb:/# apt install clamav
```

Création d'un cron qui met à jour ClamAV :

Création d'un cron avec la commande “*crontab -e*”.

```
root@gsb:/# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
```

On définit quand est ce qu'on veut que le cron s'exécute.

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 11,18 * * * /usr/bin/freshclam
```

Ici le cron s'exécute tous les jours à 11h et à 18h.

On vérifie que le cron a bien été créé.

```
crontab: installing new crontab
```

Création d'un cron qui fait un scan des fichiers :

Création du cron de scan des fichiers du site web afin de vérifier qu'il n'y ai pas de fichier malveillant, en ajoutant cette ligne dans le fichier de cron avec la commande "*crontab -e*".

```
0 * * * * clamscan -r /var/www/
```

Ici le cron s'exécute toutes les heures.

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation et configuration Hk Hunter

Installation de RK Hunter:

Connexion à la machine distante avec un utilisateur; élever ses privilèges en root avec la commande “*su -*” et le mot de passe root.

Installation de fail2ban avec la commande “*apt install ...*”.

```
root@gsb:/# apt install rkhunter
```

Création d'un cron qui met à jour RK Hunter:

Création d'un cron avec la commande “*crontab -e*”.

```
root@gsb:/# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
```

On définit quand est ce qu'on veut que le cron s'exécute. Ici il s'exécutera toutes les heures.

```
0 * * * * rkhunter --update
```

On vérifie que le cron a bien été créé.

```
crontab: installing new crontab
```

Création d'un cron qui fait un scan des fichiers :

Création du cron de scan des fichiers du site web afin de vérifier qu'il n'y ai pas de fichier malveillant, en ajoutant cette ligne dans le fichier de cron avec la commande “*crontab -e*”.

```
0 * * * * rkhunter -c /var/www/
```

Ici le cron s'exécute toutes les heures.

On vérifie que le cron a bien été créé.

```
crontab: installing new crontab
```

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation et configuration Apache et PHP

On commence par mettre à jour le cache des paquets :

```
sudo apt-get update
```

Ensuite, on installe le paquet "apache2" afin d'obtenir la dernière version d'Apache 2.4.

```
sudo apt-get install -y apache2
```

Pour qu'Apache démarre automatiquement en même temps que Debian, saisissez la commande ci-dessous (même si normalement c'est déjà le cas) :

```
sudo systemctl enable apache2
```

```
Synchronizing state of apache2.service with SysV service script with  
/lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

PHP va venir se greffer sur notre serveur Apache, comme une extension, afin de pouvoir traiter les scripts intégrés aux pages ".php". Afin d'y aller progressivement, installons le paquet "php" en lui-même :

```
sudo apt-get install -y php
```

On peut voir que cette commande va installer une multitude de paquets :

```
libapache2-mod-php7.4  libsodium23  php-common  php7.4  php7.4-cli  
php7.4-common  php7.4-json  php7.4-opcache  php7.4-readline
```

C'est très bien, nous avons quelques modules de base indispensables et "libapache2-mod-php7.4" qui permet l'intégration avec Apache.

Actuellement, c'est PHP 7.4 qui est dans les dépôts de Debian, même si PHP 8 est déjà disponible, toutes les applications ne sont pas encore compatibles. **Il faut savoir que le support de PHP 7.4 assure les mises à jour de sécurité jusqu'au 28 novembre 2022.** Ce qui laisse un peu de temps, mais il faut garder en tête qu'il faudra envisager de passer sur PHP 8.

Avant d'aller plus loin, nous allons installer quelques paquets supplémentaires pour compléter l'installation de PHP sur notre serveur. Par exemple, pour permettre les interactions entre PHP et notre instance MariaDB.

```
sudo apt-get install -y php-pdo php-mysql php-zip php-gd php-mbstring  
php-curl php-xml php-pear php-bcmath
```

Suite à cette installation, je vous invite à vérifier quelle version de PHP vous venez d'installer. Exécutez la commande suivante :

```
php -v  
PHP 7.4.21 (cli) (built: Jul 2 2021 03:59:48) ( NTS )
```

Maintenant, pour nous assurer que notre moteur de script PHP est bien actif, nous allons créer un fichier "*phpinfo.php*" (ou un autre nom) à la racine de notre site Web :

```
sudo nano /var/www/html/phpinfo.php
```

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Création d'un virtual host

Pour créer ce vhost nous allons tout d'abord nous connecter a la machine en SSH avec les accès ci-dessus :

Ensuite ici on créer le vhost dans le répertoire « sites-available » d'apache2. Et on le nomme « gsb.gsb.lan.conf ».

```
root@gsb:~# cd /etc/apache2/sites-available/  
root@gsb:/etc/apache2/sites-available# nano gsb.gsb.lan.conf  
root@gsb:/etc/apache2/sites-available#
```

On édite le vhost a l'aide de la commande « nano ».

```
VirtualHost *:80>  
    ServerName gsb.gsb.lan  
    ServerAlias www.gsb.gsb.lan  
    ServerAdmin webmaster@gsb.gsb.lan  
    DocumentRoot /var/www/html/gsb/  
  
    <Directory /var/www/html/gsb/>  
        Options -Indexes +FollowSymLinks  
        AllowOverride All  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/gsb.gsb.lan-error.log  
    CustomLog ${APACHE_LOG_DIR}/gsb.gsb.lan-access.log combined  
</VirtualHost>
```

Ensuite on va créer le dossier dans lequel on a fait la redirection dans le vhost à l'aide de la commande « mkdir » suivi du chemin dans lequel on veut créer le dossier ensuite on change les permissions de root vers www-data afin de ne pas se bloquer plus tard.

```

root@gsb:~# mkdir /var/www/html/gsb
root@gsb:~# ls -lia /var/www/html
total 28
2491781 drwxr-xr-x  4 root    root      4096 avril 13 10:19 .
2491780 drwxr-xr-x  3 root    root      4096 avril 13 09:41 ..
2498036 drwxr-xr-x  2 root    root      4096 avril 13 10:19 gsb
2492421 -rw-r--r--  1 root    root     10701 avril 13 09:41 index.html
2490391 drwxr-xr-x 12 www-data www-data  4096 avril 13 10:14 phpMyAdmin
root@gsb:~# chown www-data:www-data /var/www/html/gsb/
root@gsb:~# ls -lia /var/www/html
total 28
2491781 drwxr-xr-x  4 root    root      4096 avril 13 10:19 .
2491780 drwxr-xr-x  3 root    root      4096 avril 13 09:41 ..
2498036 drwxr-xr-x  2 www-data www-data  4096 avril 13 10:19 gsb
2492421 -rw-r--r--  1 root    root     10701 avril 13 09:41 index.html
2490391 drwxr-xr-x 12 www-data www-data  4096 avril 13 10:14 phpMyAdmin

```

Ensuite on active le vhost à l'aide de la commande « a2ensite » suivi de gsb.gsb.lan, puis on recharge apache2.

```

root@gsb:~# a2ensite gsb.gsb.lan
Enabling site gsb.gsb.lan.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@gsb:~# systemctl restart apache2

```

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation et configuration Mysql

On commence par installer Mysql

```
apt install mariadb-server mariadb-client -y
```

Que ce soit avec MariaDB ou MySQL, vous pouvez vous connecter à la console de votre instance avec la commande suivante :

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

Saisissez le mot de passe "root" de votre instance : une console va s'ouvrir, prête à recevoir des commandes SQL.

Première étape : la création de la base de données. Ne donnez pas un nom trop évident, mais parlant malgré tout, par exemple cela peut être : **gsb**.

```
CREATE DATABASE gsb;  
# Retour dans la console :  
Query OK, 1 row affected (0.001 sec)
```

Vous pouvez lister les bases de données de votre instance avec la commande suivante :

```
SHOW DATABASES;
```

On peut voir que notre base de données apparaît bien dans la liste

Deuxième étape : créer l'utilisateur qui sera administrateur de la base de données WordPress. Cet utilisateur sera nommé "gsbweb" et il aura comme mot de passe "Aristee.2023".

Ce qui donne la requête SQL suivante :

```
CREATE USER 'gsbweb'@'localhost' IDENTIFIED BY Aristee.2023;
```

Troisième étape : donner tous les droits à l'utilisateur "agsbweb" sur la base de données WordPress. Notre serveur Web et la base de données étant

sur le même serveur, nous allons donner ces droits pour une connexion locale.
Ce qui donne :

```
GRANT ALL PRIVILEGES ON gsb.* TO gsbweb@localhost;
```

Enfin, il faut exécuter la commande suivante pour actualiser les droits et activer les nouveaux privilèges sur notre base de données :

```
FLUSH PRIVILEGES;
```

La base de données pour WordPress est prête. Pour le moment elle est vide, mais WordPress va créer sa structure de tables lors de l'installation. Quittez la console MariaDB / MySQL :

```
exit
```

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation et configuration PhpMyAdmin

L'installation de PhpMyAdmin ne s'effectue pas comme un paquet classique, mais plutôt sur le même principe qu'une application web. Il faut que l'on télécharge les sources à partir du site officiel, directement dans le dossier "/tmp" (ou ailleurs) :

```
cd /tmp
wget
https://files.phpmyadmin.net/phpMyAdmin/5.1.3/phpMyAdmin-5.1.3-all-lang
uages.zip
```

Avant de faire la commande "wget" je vous invite à accéder à [la page de téléchargement de PhpMyAdmin](#) afin de récupérer le lien vers la dernière version. Ici, c'est bien la dernière version stable : 5.1.3.

Ensuite, nous devons extraire le contenu de cette archive ZIP avec la commande "unzip". Elle n'est pas installée par défaut sur Debian 11. Vous pouvez l'installer avec cette commande :

```
sudo apt-get update
sudo apt-get install unzip
```

Ensuite, on décompresse l'archive ZIP dans le répertoire courant :

```
unzip phpMyAdmin-5.1.3-all-languages.zip
```

On va déplacer le dossier complet vers "/usr/share" dans un nouveau dossier nommé "phpmyadmin". Ce qui donne :

```
sudo mv phpMyAdmin-5.1.3-all-languages /usr/share/phpmyadmin
```

Ensuite, on crée un dossier distinct pour les fichiers temporaires :

```
sudo mkdir -p /var/lib/phpmyadmin/tmp
```

Puis, on attribue les droits sur le dossier racine "*phpmyadmin*" à l'utilisateur associé à Apache (www-data) afin qu'il soit propriétaire. Nous précisons le chemin vers le dossier "tmp" dans la configuration de PhpMyAdmin.

```
sudo chown -R www-data:www-data /var/lib/phpmyadmin/
```

PhpMyAdmin est fourni avec un template pour le fichier de configuration, alors on va créer une copie de ce template pour ne pas partir de zéro :

```
cp /usr/share/phpmyadmin/config.sample.inc.php  
/usr/share/phpmyadmin/config.inc.php
```

Afin d'utiliser le mode d'authentification basé sur les cookies, nous devons générer une chaîne aléatoire qui est une sorte de passphrase au sein du fichier de configuration. Il doit s'agir d'une chaîne de 32 caractères. Un cookie permanent stockera l'identifiant sur votre machine tandis que le mot de passe est géré par un cookie temporaire.

On peut générer cette chaîne aléatoire avec la commande suivante :

```
openssl rand -base64 32
```

Copiez la valeur retournée en sortie. Nous allons l'insérer dans le fichier de configuration de PhpMyAdmin. Ouvrez le fichier avec nano (ou un autre éditeur de texte) :

```
nano /usr/share/phpmyadmin/config.inc.php
```

Collez la valeur au niveau de l'option "*blowfish_secret*", comme ceci :

```
$cfg['blowfish_secret'] =  
'deJ8reLGV1cXPYd32454/um/EGWRef/14Jo7tg112WM=';
```

Sauvegardez et fermez le fichier

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Installation de Wordpress

Nous allons utiliser le site par défaut d'Apache, qui a pour racine "/var/www/html" afin de stocker les données de notre site WordPress. Au préalable, on supprime la page d'index créée par défaut par Apache :

```
sudo rm /var/www/html/index.html
```

Ensuite, on installe le paquet « zip » sur notre serveur pour pouvoir décompresser l'archive de WordPress :

```
sudo apt-get update  
sudo apt-get install zip
```

On décompresser l'archive dans "/var/www/html" grâce à la commande suivante (en étant positionné dans le dossier où l'on a téléchargé le fichier latest.zip) :

```
sudo unzip latest.zip -d /var/www/html
```

L'option "-d" permet de définir là où sera décompressée l'archive. Le dossier WordPress apparaîtra donc dans "/var/www/html" qui est le dossier où sont stockées les pages web par défaut.

Le problème, c'est que là on vient de décompresser le contenu de l'archive ZIP dans un dossier nommé "wordpress", ce qui donne : /var/www/html/wordpress. Du coup, pour accéder à notre site, il faudra faire : <http://domaine.fr/wordpress/>. Ce n'est pas top, nous allons corriger cela dès maintenant.

Déplacez-vous dans le dossier "/var/www/html" :

```
cd /var/www/html
```

Ensuite, exécutez la commande ci-dessous pour déplacer tout le contenu du dossier "wordpress" à la racine de notre site :

```
sudo mv wordpress/* /var/www/html/
```

Puisque le dossier "wordpress" ne sert plus à rien, on va le supprimer :

```
sudo rm wordpress/ -Rf
```

Enfin, on termine en donnant les droits à l'utilisateur "www-data" (correspondant à Apache) sur tous les fichiers de notre site, de manière réursive :

```
sudo chown -R www-data:www-data /var/www/html/
```

On obtient une belle liste de fichiers et dossiers. Au niveau des droits et pour des raisons de sécurité, vous devez avoir 755 sur les dossiers et 644 sur les fichiers. Ce qui est le cas par

défaut si vous n'avez pas fait de modifications. En aucun cas vous ne devez poser des droits "777" sur un dossier ou un fichier.

Pour les fichiers, exécutez cette commande :

```
sudo find /var/www/html/ -type f -exec chmod 644 {} \;
```

Pour les dossiers, exécutez cette commande :

```
sudo find /var/www/html/ -type d -exec chmod 755 {} \;
```

Passez à la suite : ce sera à partir d'un navigateur.

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

Procédure PLUGIN Wordpress

CookieYes / GDPR Cookie Consent permet de rendre votre site Web conforme au GDPR (RGPD, DSGVO) en ajoutant une bannière de cookies à votre site. De plus, ce plugin WordPress GDPR prend également en charge la conformité des cookies avec le LGPD du Brésil, la CNIL de France et le California Consumer Privacy Act (CCPA/CPRA) qui est une loi d'État visant à renforcer les droits à la vie privée et la protection des consommateurs.



CookieYes | GDPR Cookie Consent & Compliance Notice (CCPA Ready)

[Installer maintenant](#)[Plus de détails](#)

Easily set up cookie notice, cookie policy and get GDPR cookie compliance. Supports GDPR (DSGVO, RGPD), LGPD, CCPA/CPRA, CNIL, and POPIA.

Par *CookieYes*



1 million et + installations
actives

Dernière mise à jour : il y a 3 mois

✓ **Compatible** avec votre version de WordPress

WPForms : Plugin WordPress freemium qui permet de créer différents types de formulaires (contact, paiement, abonnement, sondage, etc.) sans coder, grâce à des modèles prêts à l'emploi et une interface en glisser-déposer. Il se caractérise notamment par sa facilité d'usage et sa flexibilité.



Contact Form by WPForms – Drag & Drop Form Builder for WordPress

[Installer maintenant](#)[Plus de détails](#)

The best WordPress contact form plugin. Drag & Drop online form builder to create beautiful contact forms, payment forms, & other custom forms ...

Par *WPForms*



5 millions et + installations
actives

Dernière mise à jour : il y a 22 heures

✓ **Compatible** avec votre version de WordPress

CAPTCHA 4WP : Permet d'implémenter facilement CAPTCHA dans n'importe quel formulaire WordPress intégré.



CAPTCHA 4WP

[Installer maintenant](#)

Stop spam bots, fake accounts, and fake orders and allow prospects and customers to interact with your website with ease - add CAPTCHA to any form on ...

[Plus de détails](#)

Par *WP White Security*



200 000+ installations actives

Dernière mise à jour : il y a 1 semaine

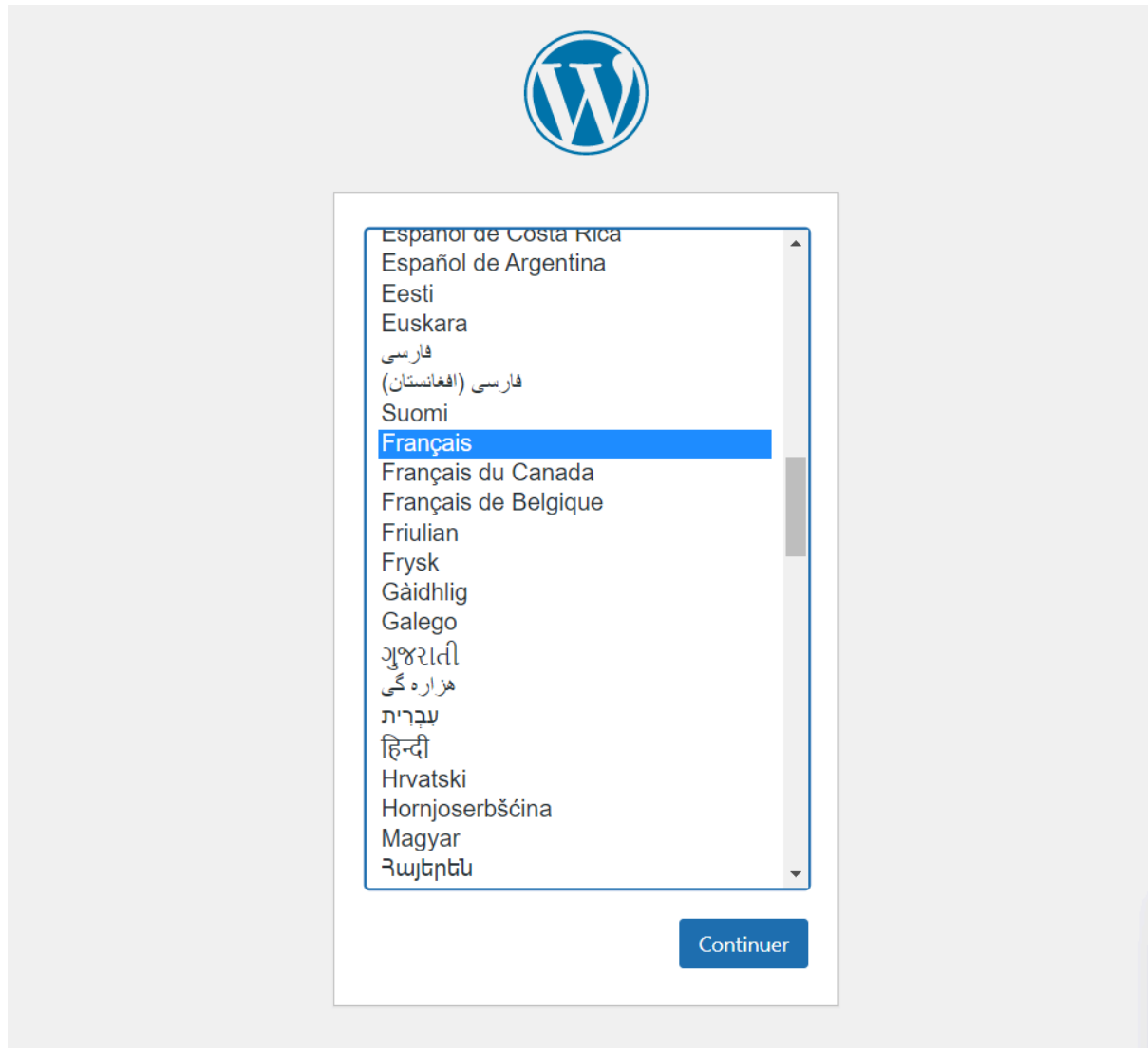
✓ **Compatible** avec votre version de WordPress

Temps de recherche : 3 jours.

Temps d'installation : 1 jour.

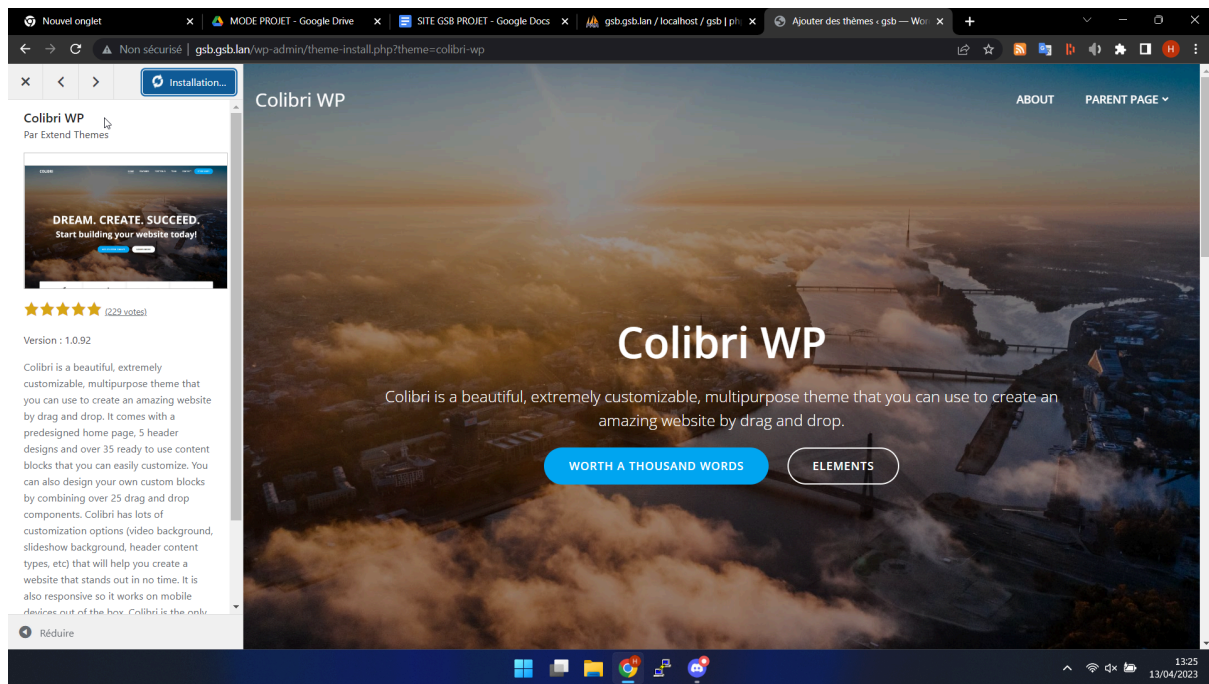
Création du site

Tout d'abord, il faut choisir la langue.

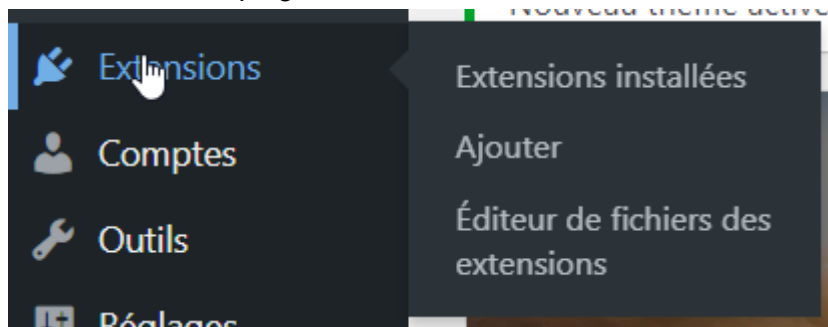


Maintenant il faut remplir les informations relatives au site web

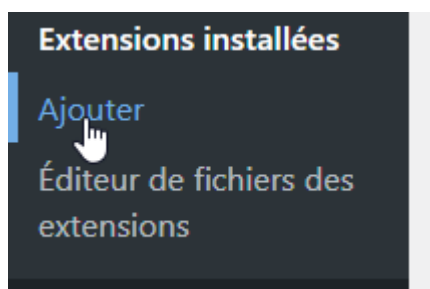
Voici la page d'accueil du site internet GSB après avoir importé le thème colibri.



Pour installer des plugins, il faut se rendre dans “Extensions”

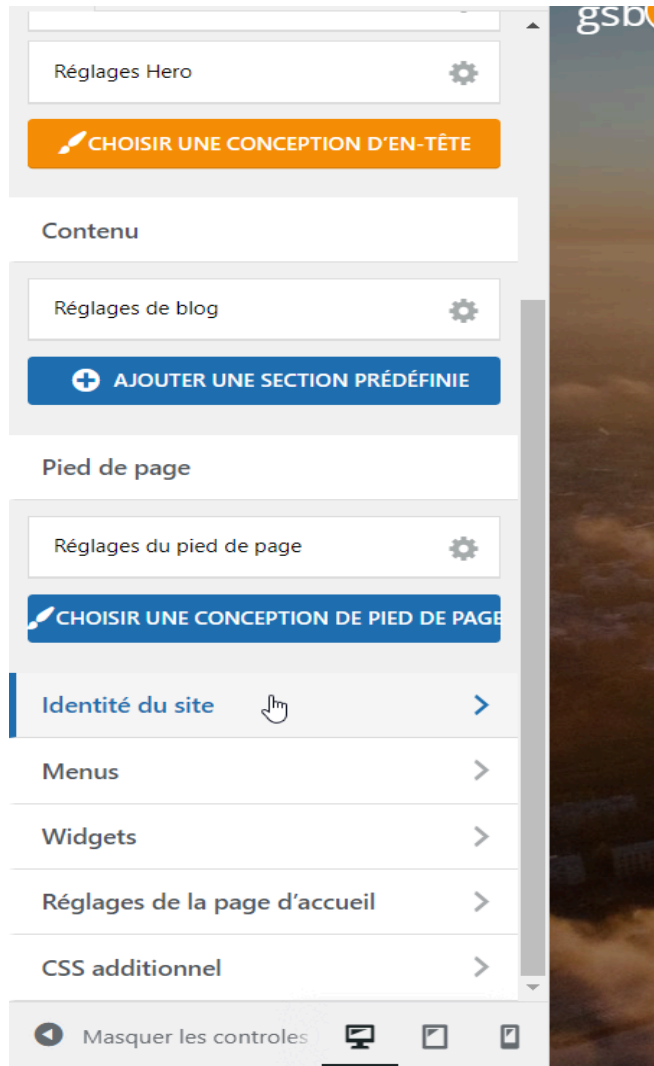


Puis faire “Ajouter”.

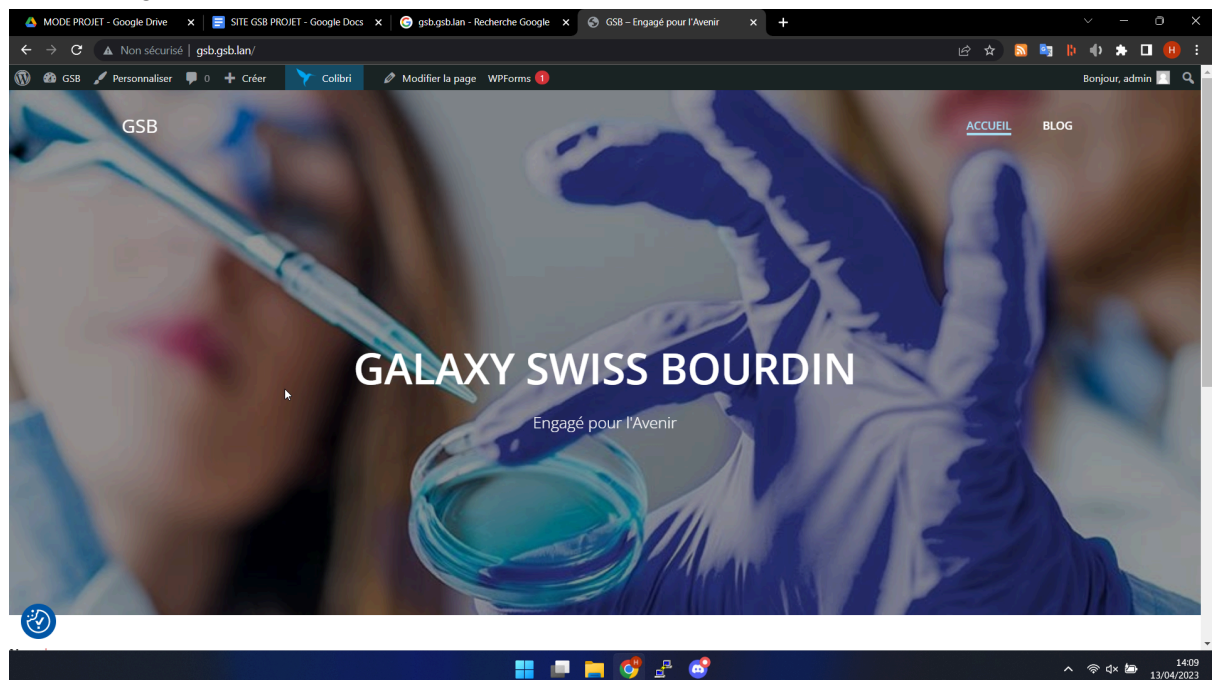


Pour personnaliser le site, il faut se rendre dans “Personnaliser” dans l’onglet sur la partie supérieure de l’écran.

Une nouvelle page s'ouvre et permet de faire diverses modifications.



Voici la page d'accueil du site après modifications.



J'ai ajouté un formulaire de contact avec WPFrom.

GSB

[ACCUEIL](#) [BLOG](#)

Nom *

Prénom

Nom

E-mail *

Commentaire ou message

Envoyer

© 2023 GSB. Created for free using WordPress and Colibri

Greenshot
Exported to: Save as (displaying dialog)

14:09
13/04/2023

Nous pouvons voir que le formulaire est fonctionnel.

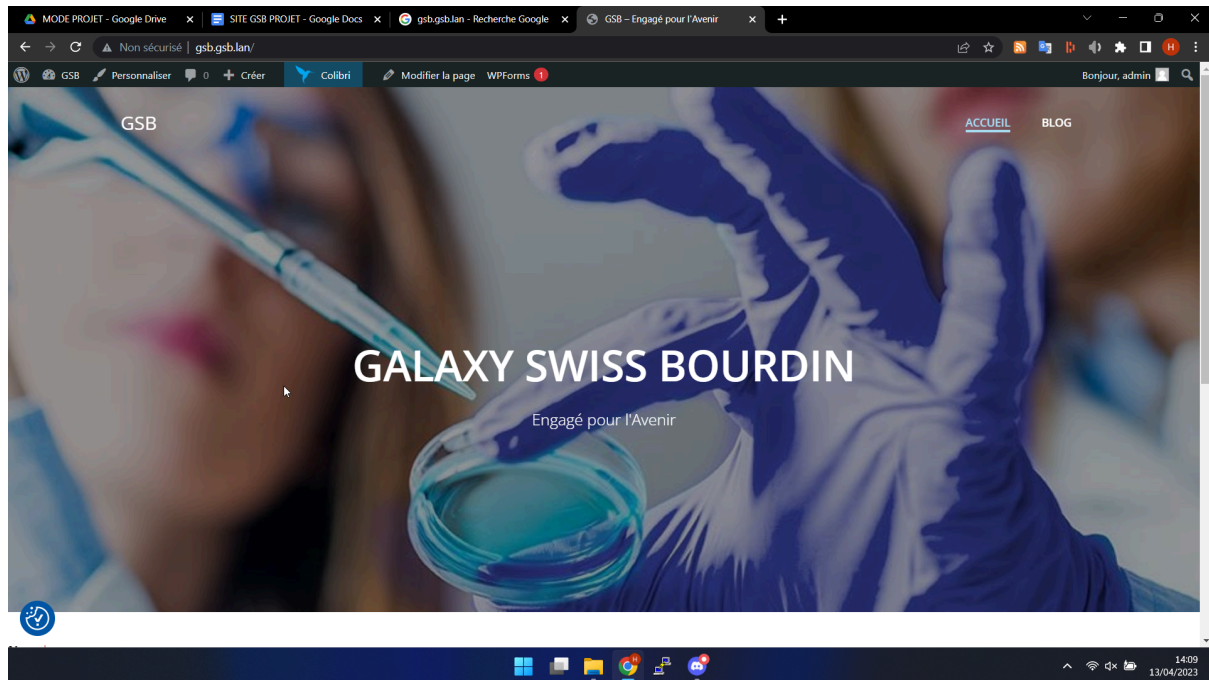
Merci de nous avoir contacté ! Nous allons revenir vers vous rapidement.

Temps de recherche : 3 jours.

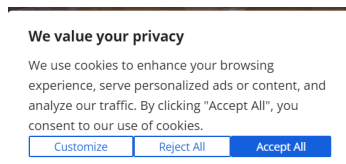
Temps d'installation : 1 jour.

Tests de fonctionnement

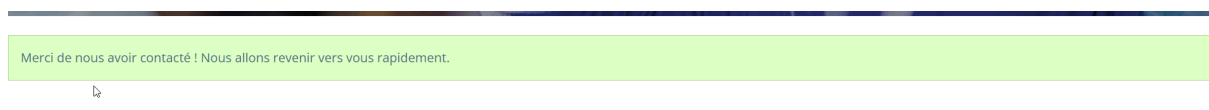
Nous pouvons constater que le site est bien joignable.



Le plugin de cookie est fonctionnel.



Le formulaire de contact est bien fonctionnel.



Temps de recherche : 3 jours.

Temps d'installation : 1 jour.