

Projet d'Exposition Picasso

Tamara Crétard - 15.01.2024

Table des matières

1. Contexte.....	2
2. Gestion du projet.....	2
2.1. Plan.....	2
2.2. Issues (Problèmes).....	3
3. Solution applicative.....	3
4. Hébergement.....	3
4.1. Installer une machine virtuelle sur la ferme de serveurs.....	3
4.2. Installer SSH pour se connecter à distance.....	5
4.3. Transférer des fichiers du PC hôte au serveur web.....	7
4.4. Activer un firewall sur le serveur web.....	8
4.5. Passer le site en HTTPS avec un certificat auto-signé.....	9
4.6. Installer PHP sur le serveur.....	10
4.7. Installer MariaDB sur le serveur.....	12
4.8. Mettre en place un serveur Linux Debian dans AWS.....	12
4.9. Créer un nom de domaine DNS.....	14
4.10. Configurer Apache du serveur AWS pour utiliser le nom de domaine.....	15
4.11. Installer Certbot pour générer automatiquement un certificat SSL.....	16

1. Contexte

Le musée du film d'animation de la ville d'Annecy, grâce à un accord exceptionnel avec de grands musées du monde entier, a réussi à réunir 9 œuvres majeures de Pablo Picasso (images fournies dans Classroom), artiste ayant inspiré nombre de réalisateurs de films d'animation.

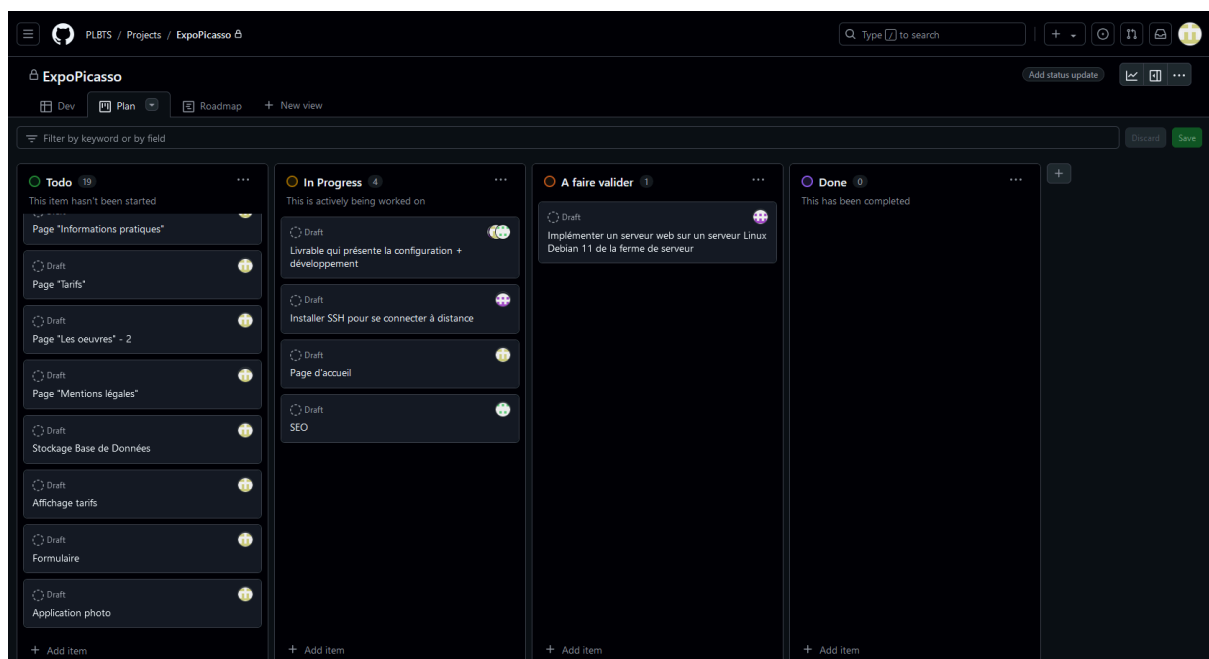
Ces œuvres seront exposées du 10 au 22 mai 2025 dans la salle d'exposition temporaire du musée.

La direction du musée fait appel aux BTS SIO du lycée Gabriel Fauré pour mettre en œuvre le site web de l'exposition temporaire et pour l'hébergement.

2. Gestion du projet

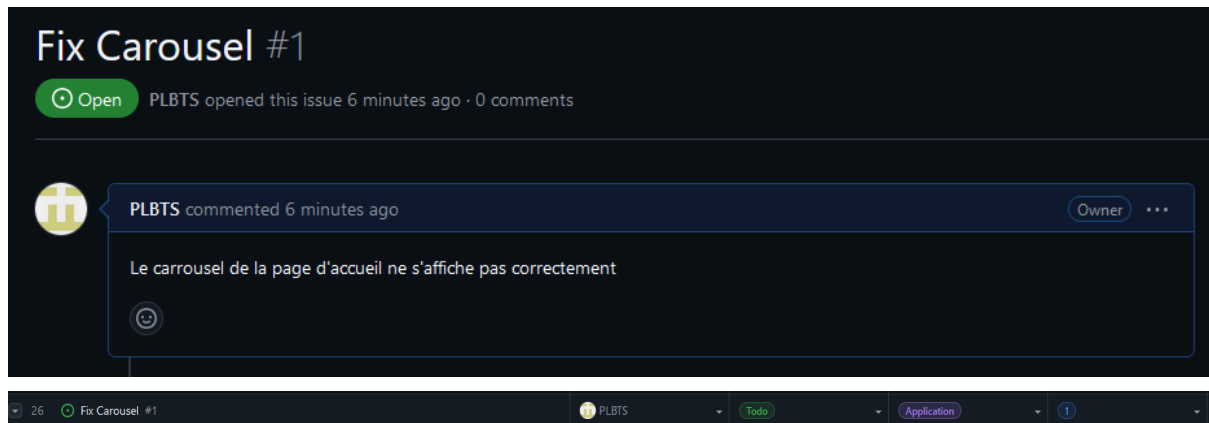
2.1. Plan

Afin de mener à bien ce projet, nous avons utilisé GitHub pour visualiser les différentes étapes à réaliser, celles qui sont en cours de mise en place, celles à faire vérifier et celles terminées:



2.2. Issues (Problèmes)

Lors du travail sur certaines étapes, il y a parfois eu des problèmes et il a fallu en garder une trace. C'est ce que nous avons fait, toujours dans GitHub:



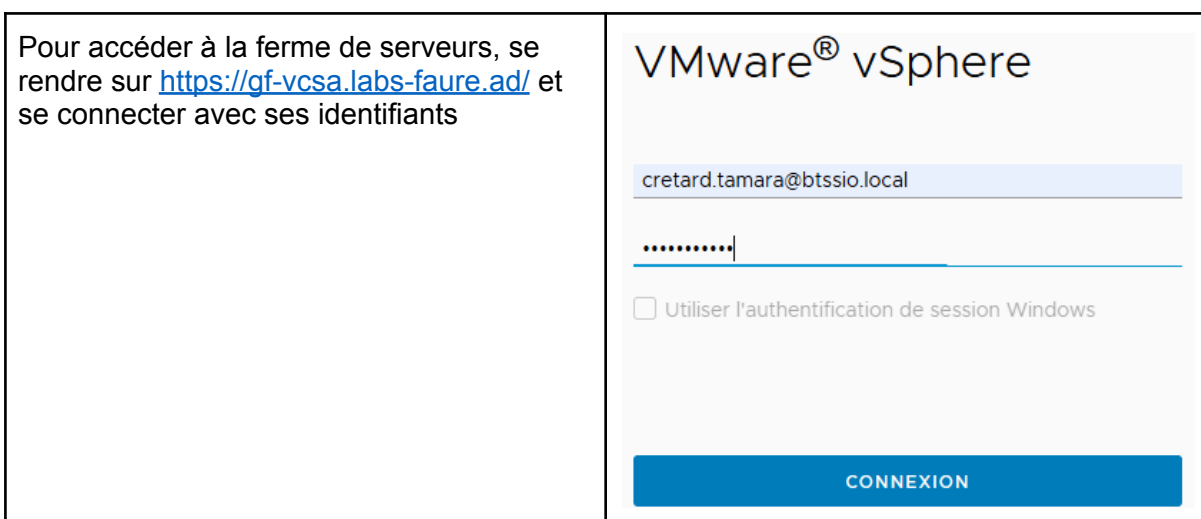
3. Solution applicative


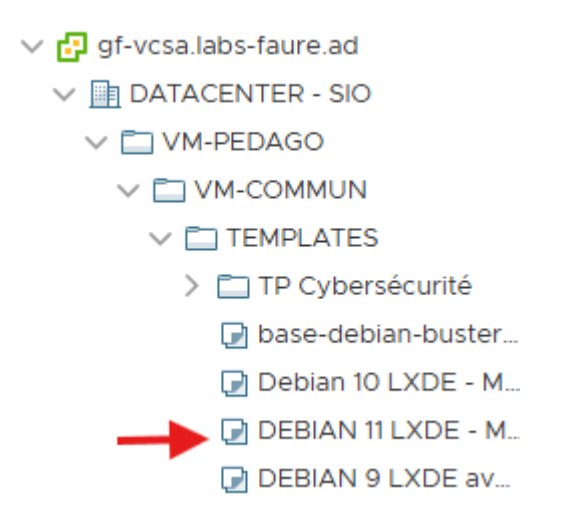

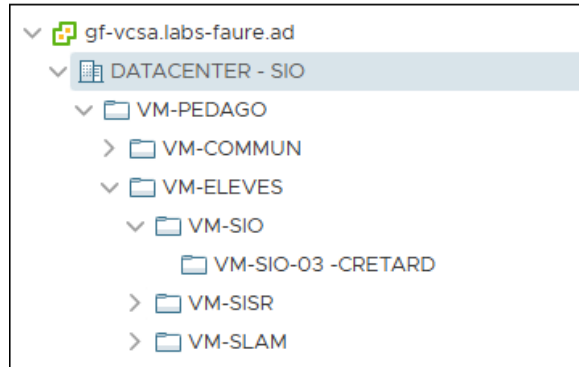
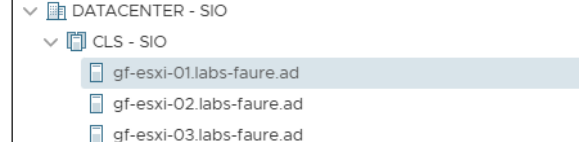
Je ne me suis pas occupée de la solution applicative et mes camarades n'ont pas souhaité prendre de captures d'écran pour montrer l'évolution du projet.

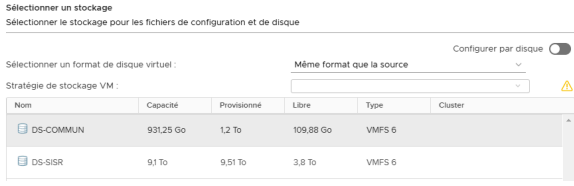
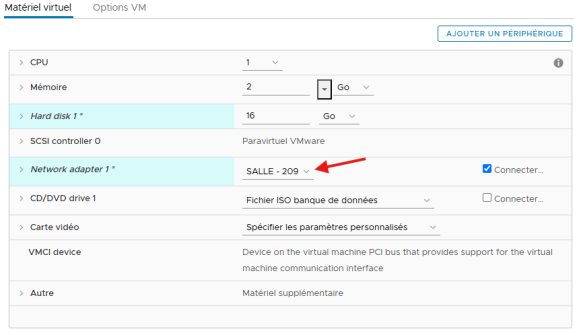
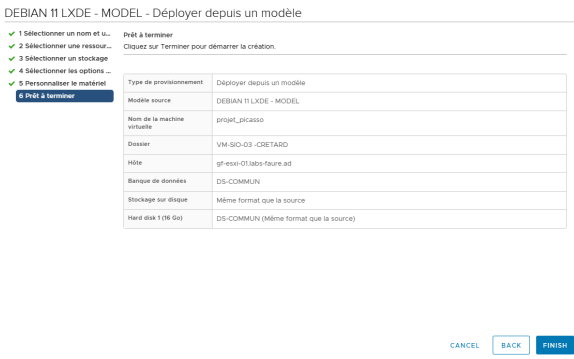
4. Hébergement

4.1. Installer une machine virtuelle sur la ferme de serveurs

Dans un premier temps, il est nécessaire d'installer une machine virtuelle sur le serveur. Voici les étapes à suivre pour le réaliser:



<p>Une fois connecté, aller dans le menu « VM et Modèles »</p>	
<p>Trouver le template DEBIAN11 LXDE dans : VM-PEDAGO / VM-COMMUN / TEMPLATES</p>	
<p>faire click-droit, et « Nouvelle VM à partir de ce modèle »</p>	
<p>lui donner un nom et la ranger dans le répertoire situé dans VM-PEDAGO / VM-ELEVES.</p>	<p>Sélectionner un nom et un dossier Spécifiez un nom unique et un emplacement cible</p> <p>Nom de la machine virtuelle : <input type="text" value="projet_picasso"/></p> <p>Sélectionnez un emplacement pour la machine virtuelle.</p> 
<p>Pour la ressource de calcul, sélectionner au choix l'une des 3 ressources disponibles</p>	<p>Sélectionner une ressource de calcul Sélectionnez la ressource de calcul de destination pour cette opération</p> 

<p>Pour le stockage, sélectionner DS-COMMUN</p>	
<p>Cocher « Personnaliser le matériel de cette machine virtuelle »</p>	<p><input type="checkbox"/> Personnaliser le système d'exploitation</p> <p><input checked="" type="checkbox"/> Personnaliser le matériel de cette machine virtuelle</p> <p><input type="checkbox"/> Mettre sous tension la machine virtuelle après la création</p>
<p>Comme « Network Adapter 1 », vous sélectionnez « SALLE – 209 »</p>	
<p>Déployer la machine en cliquant sur "FINISH"</p>	

4.2. Installer SSH pour se connecter à distance

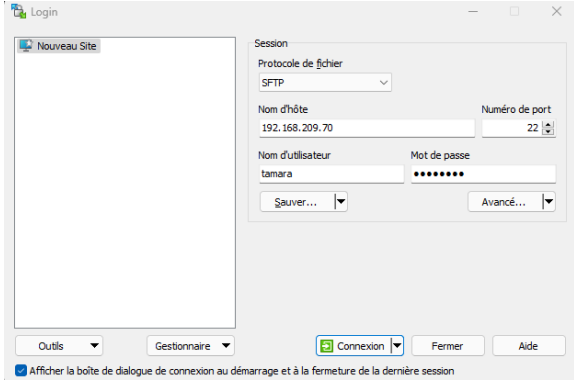
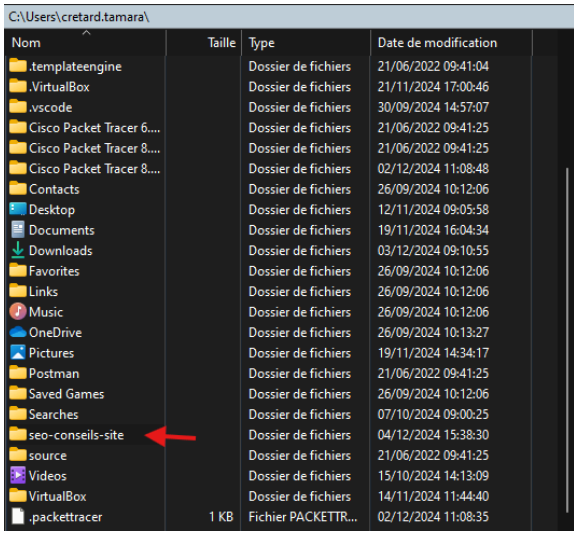
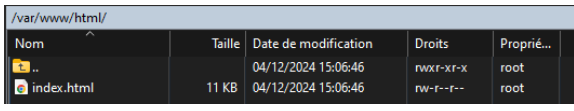
Afin de rendre possible la connexion à distance, il est nécessaire d'installer SSH sur notre machine virtuelle. Pour se faire, voici les étapes à suivre:

<p>Démarrer la machine virtuelle et se connecter</p>	
--	--

<p>Ouvrir le terminal en allant dans le menu Debian -> Applications -> Interpréteurs de commandes -> Bash.</p>	
<p>Mettre à jour votre système avec la commande : <code>sudo apt update</code> suivi de la commande <code>sudo apt upgrade</code></p>	
<p>Installer le paquet openssh-server avec la commande: <code>sudo apt install openssh-server</code></p>	<pre>sudo apt install openssh-server</pre>
<p>Activer et démarrer le service SSH avec les commandes suivantes: <code>sudo systemctl enable ssh</code> <code>sudo systemctl start ssh</code></p>	<pre>sudo systemctl enable ssh sudo systemctl start ssh</pre>
<p>Créer un utilisateur avec la commande suivante: <code>sudo adduser nom_utilisateur</code> et choisir un mot de passe</p>	
<p>Ajouter le compte utilisateur au groupe sudo avec la commande suivante: <code>usermod -aG sudo utilisateur</code> et vérifier de quel groupe est l'utilisateur avec la commande <code>groups utilisateur</code></p>	
<p>Se connecter à la machine virtuelle depuis le PC hôte avec la commande suivante: <code>ssh utilisateur@adresse_ip_debian</code> et entrer le mot de passe. Pour connaître l'adresse IP, faire "ifconfig" sur Debian</p>	
<p>Mettre à jour la liste des paquets disponibles et les paquets déjà installés avec la commande <code>sudo apt update && sudo apt upgrade -y</code></p>	<pre>tamara@debian:~\$ sudo apt update && sudo apt upgrade -y</pre>
<p>Installer le serveur Apache avec la commande <code>sudo apt install apache2 -y</code> et vérifier le statut avec <code>sudo systemctl status apache2</code></p>	

4.3. Transférer des fichiers du PC hôte au serveur web

Pour pouvoir transférer des fichiers du PC hôte au serveur web, il faut passer par un client. Dans notre cas, nous utiliserons WinSCP déjà installé. Voici les étapes à suivre pour ce transfert:

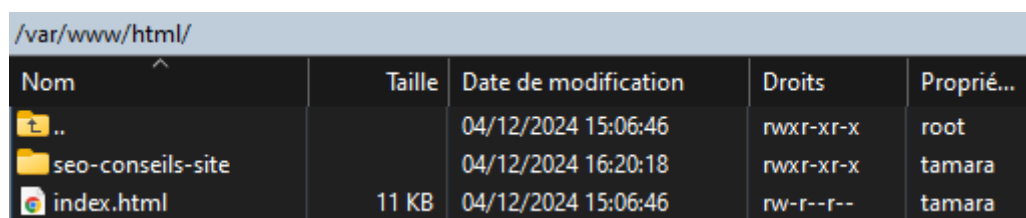
<p>Entrer les informations concernant le serveur web et se connecter</p>	
<p>Trouver les fichiers à transférer dans le panneau de gauche qui représente le PC hôte</p>	
<p>Dans le panneau de droite, naviguer vers l'emplacement où Apache héberge les fichiers web: /var/www/html</p>	
<p>Glisser-déposer les fichiers ou dossiers du panneau de gauche (PC) vers le panneau de droite (serveur)</p>	

Un message code d'erreur 3 Permission Denied peut s'afficher. L'erreur est due au fait que l'utilisateur ne possède pas les permissions du répertoire dans lequel les fichiers vont être transférés. Pour la régler, voici les étapes à suivre:

<p>Se connecter à la machine virtuelle depuis le terminal du PC hôte</p>	<pre>ssh tamara@192.168.209.70 tamara@192.168.209.70's password:</pre>
<p>Vérifier qui est le propriétaire du répertoire</p>	<pre>tamara@debian:~\$ ls -ld /var/www/html drwxr-xr-x 2 root root 4096 4 déc. 15:06 /var/www/html</pre>

avec <code>ls -ld /var/www/html</code> . On observe que le propriétaire est <code>root</code> et non <code>tamara</code>	
Pour attribuer les permissions à l'utilisateur souhaité, entrer la commande suivante: <code>sudo chown -R utilisateur:utilisateur /var/www/html</code>	<pre>tamara@debian:~\$ sudo chown -R tamara:tamara /var/www/html</pre>
On peut denouveau vérifier qui est le propriétaire du répertoire avec <code>ls -ld /var/www/html</code> . Le propriétaire est maintenant <code>tamara</code>	<pre>tamara@debian:~\$ ls -ld /var/www/html drwxr-xr-x 2 tamara tamara 4096 4 déc. 15:06 /var/www/html</pre>

Il est maintenant possible de “glisser-déposer” les fichiers ou dossiers du panneau de gauche (PC) vers le panneau de droite (serveur):



4.4. Activer un firewall sur le serveur web

Pour sécuriser le serveur web, il faut installer un Firewall. Nous installerons un Firewall UFW qui permettra de bloquer des requêtes qui n'ont pas été autorisées. Pour ce faire, voici les étapes à suivre:

Se connecter à la machine virtuelle depuis le terminal du PC hôte	<pre>ssh tamara@192.168.209.70 tamara@192.168.209.70's password:</pre>
Mettre à jour le système avec la commande <code>sudo apt update && sudo apt upgrade -y</code>	<pre>tamara@debian:~\$ sudo apt update && sudo apt upgrade -y</pre>
Installer UFW avec la commande <code>sudo apt install ufw</code>	<pre>tamara@debian:~\$ sudo apt install ufw</pre>
Ajouter les règles suivantes: <code>sudo ufw allow ssh</code> , <code>sudo ufw allow https</code> , <code>sudo ufw allow 22</code>	<pre>tamara@debian:~\$ sudo ufw allow ssh Rules updated Rules updated (v6) tamara@debian:~\$ sudo ufw allow 22 Rules updated Rules updated (v6) tamara@debian:~\$ sudo ufw allow https Rules updated Rules updated (v6)</pre>
Activer UFW avec la commande suivantes: <code>sudo ufw enable</code>	<pre>tamara@debian:~\$ sudo ufw enable Command may disrupt existing ssh connections. Proceed with operation (y/n)? y Firewall is active and enabled on system startup</pre>

<p>Vérifier que tout fonctionne correctement avec <code>sudo ufw status verbose</code></p>	<pre>tamara@debian:~\$ sudo ufw status verbose Status: active Logging: on (Low) Default: deny (incoming), allow (outgoing), disabled (routed) New profiles: skip To Action From --- 22/tcp ALLOW IN Anywhere 22 ALLOW IN Anywhere 443 ALLOW IN Anywhere 22/tcp (v6) ALLOW IN Anywhere (v6) 22 (v6) ALLOW IN Anywhere (v6) 443 (v6) ALLOW IN Anywhere (v6)</pre>
--	--

4.5. Passer le site en HTTPS avec un certificat auto-signé

Il est nécessaire de sécuriser les échanges entre le serveur et le client ainsi que d'authentifier le site web. Pour ce faire, il faut passer le site en HTTPS avec un certificat auto-signé. Voici les étapes à suivre:

<p>Se connecter à la machine virtuelle depuis le terminal du PC hôte</p>	<pre>ssh tamara@192.168.209.70 tamara@192.168.209.70's password:</pre>
<p>Créer un répertoire pour stocker les certificats avec <code>sudo mkdir /etc/ssl/self-signed</code></p>	<pre>tamara@debian:~\$ sudo mkdir /etc/ssl/self-signed [sudo] Mot de passe de tamara :</pre>
<p>Générer une clé privée avec <code>sudo openssl genrsa -out /etc/ssl/self-signed/selfsigned.key 2048</code></p>	<pre>tamara@debian:~\$ sudo openssl genrsa -out /etc/ssl/self-signed/selfsigned.key 2048 Generating RSA private key, 2048 bit long modulus (2 primes)+++++ e is 65537 (0x010001)</pre>
<p>Créer une demande de certificat (CSR) et un certificat auto-signé avec la commande <code>sudo openssl req -x509 -new -nodes -key /etc/ssl/self-signed/selfsigned.key \ -sha256 -days 365 -out /etc/ssl/self-signed/selfsigned.crt</code></p>	<pre>tamara@debian:~\$ sudo openssl req -x509 -new -nodes -key /etc/ssl/self-signed/selfsigned.key \ -sha256 -days 365 -out /etc/ssl/self-signed/selfsigned.crt [sudo] Mot de passe de tamara : You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter '.', the field will be left blank. ----- Country Name (2 letter code) [AU]:FR State or Province Name (full name) [Some-State]:Auvergne-Rhône-Alpes Locality Name (eg, city) []:Annecy Organization Name (eg, company) [Internet Widgits Pty Ltd]:SIO Organizational Unit Name (eg, section) []:SIO1 Common Name (e.g. server FQDN or YOUR name) []:192.168.209.70 Email Address []:tamara.cretard@lycee-faure.fr</pre>
<p>Activer le module SSL avec <code>sudo a2enmod ssl</code> et redémarrer Apache avec la commande <code>sudo systemctl restart apache2</code></p>	<pre>tamara@debian:~\$ sudo a2enmod ssl Considering dependency setenvif for ssl: Module setenvif already enabled Considering dependency mime for ssl: Module mime already enabled Considering dependency socache_shmcb for ssl: Enabling module socache_shmcb. Enabling module ssl. See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates. To activate the new configuration, you need to run: systemctl restart apache2</pre>
<p>Lister les fichiers avec <code>ls -l /etc/apache2/sites-available/</code> s'il n'y a pas notre site, créer le fichier de configuration avec <code>sudo nano /etc/apache2/sites-available/seo-conseils-site.conf</code></p>	<pre>tamara@debian:~\$ ls -l /etc/apache2/sites-available/ total 12 -rw-r--r-- 1 root root 1496 4 déc. 17:28 00-default.conf -rw-r--r-- 1 root root 6338 4 oct. 17:21 default-ssl.conf tamara@debian:~\$ sudo nano /etc/apache2/sites-available/seo-conseils-site.conf</pre>

<p>Modifier le contenu du fichier: <code>sudo nano</code> <code>/etc/apache2/sites-available/seo-conseils-site.conf</code></p>	<pre><VirtualHost *:80> ServerAdmin webmaster@localhost ServerName tamaral.sio.local DocumentRoot /var/www/html/seo-conseils-site DirectoryIndex seo-conseils.html <Directory /var/www/html/seo-conseils-site> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined </VirtualHost> <VirtualHost *:443> ServerAdmin webmaster@localhost ServerName tamaral.sio.local DocumentRoot /var/www/html/seo-conseils-site DirectoryIndex seo-conseils.html ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined <Directory /var/www/seo-conseils-site> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> SSLEngine on SSLCertificateFile /etc/ssl/self-signed/selfsigned.crt SSLCertificateKeyFile /etc/ssl/self-signed/selfsigned.key </VirtualHost></pre>
<p>Activer le site avec <code>sudo a2ensite seo-conseils-site.conf</code> et redémarrer Apache avec <code>sudo systemctl reload apache2</code></p>	<pre>tamara@debian:~\$ sudo a2ensite seo-conseils-site.conf Enabling site seo-conseils-site. To activate the new configuration, you need to run: systemctl reload apache2 tamara@debian:~\$ sudo systemctl reload apache2</pre>

4.6. Installer PHP sur le serveur

Afin d'installer PHP sur le serveur, voici les étapes à suivre:

<p>Mettre à jour le système avec <code>sudo apt update && sudo apt upgrade -y</code></p>	<pre>tamara@debian:~\$ sudo apt update && sudo apt upgrade -y</pre>
<p>Installer PHP et les extensions avec: <code>sudo apt install php libapache2-mod-php php-mysql -y</code></p> <p>php : le langage PHP libapache2-mod-php : permet à Apache d'interpréter les fichiers PHP php-mysql : extension pour que PHP interagisse avec MariaDB</p>	<pre>tamara@debian:~\$ sudo apt install php libapache2-mod-php php-mysql -y</pre>
<p>Vérifier l'installation avec <code>php -v</code></p>	<pre>tamara@debian:~\$ php -v PHP 7.4.33 (cli) (built: Dec 7 2024 22:44:42) (NTS) Copyright (c) The PHP Group Zend Engine v3.4.0, Copyright (c) Zend Technologies with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies</pre>
<p>Vérifier que le module est bien installé avec <code>dpkg -l grep libapache2-mod-php</code></p>	<pre>tamara@debian:~\$ dpkg -l grep libapache2-mod-php ii libapache2-mod-php 2:7.4.33-1+deb11u7 amd64 server-side, HTML-embedded scri ii libapache2-mod-php7.4 7.4.33-1+deb11u7 amd64 server-side, HTML-embedded scri ii libapache2-mod-php7.4 7.4.33-1+deb11u7 amd64 server-side, HTML-embedded scri</pre>

<p>Vérifier les modules disponibles: <code>ls /usr/lib/apache2/modules grep php</code></p>	<pre>tamara@debian:~\$ ls /usr/lib/apache2/modules grep php libphp7.4.so</pre>
<p>Activer le module voulu avec <code>sudo a2enmod php7.4</code> et redémarrer Apache <code>sudo systemctl restart apache2</code></p>	<pre>tamara@debian:~\$ sudo a2enmod php7.4 Considering dependency mpm_prefork for php7.4: Considering conflict mpm_event for mpm_prefork: Considering conflict mpm_worker for mpm_prefork: Module mpm_prefork already enabled Considering conflict php5 for php7.4: Module php7.4 already enabled</pre>
<p>Modifier le fichier de configuration Apache en ajoutant un DirectoryIndex index.php et redémarrer denouveau Apache</p>	<pre><VirtualHost *:80> ServerAdmin webmaster@localhost ServerName tamaral.sio.local DocumentRoot /var/www/html/seo-conseils-site DirectoryIndex seo-conseils.html index.php <Directory /var/www/html/seo-conseils-site> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined </VirtualHost> <VirtualHost *:443> ServerAdmin webmaster@localhost ServerName tamaral.sio.local DocumentRoot /var/www/html/seo-conseils-site DirectoryIndex seo-conseils.html index.php ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined <Directory /var/www/seo-conseils-site> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> SSLEngine on SSLCertificateFile /etc/ssl/self-signed/selfsigned.crt SSLCertificateKeyFile /etc/ssl/self-signed/selfsigned.key </VirtualHost></pre>
<p>Tester PHP en créant un fichier de test dans le répertoire par défaut d'Apache: <code>sudo nano /var/www/html/seo-conseils-site/index.php</code> et en ajoutant: <code><?php</code> <code>echo "PHP fonctionne !";</code> <code>?></code> en allant sur <code>192.168.209.70/info.php</code> il devrait y avoir une page d'informations PHP</p>	<pre>sudo nano /var/www/html/info.php</pre> <pre><?php echo "PHP fonctionne !"; ?></pre> <p>← → ↻ ⚠ Non sécurisé 192.168.209.70/index.php</p> <p>PHP fonctionne !</p>

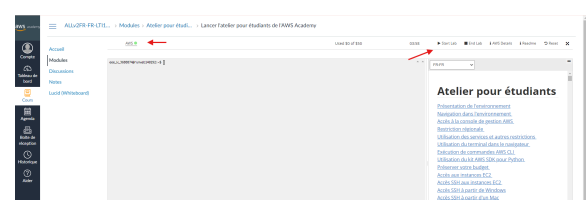
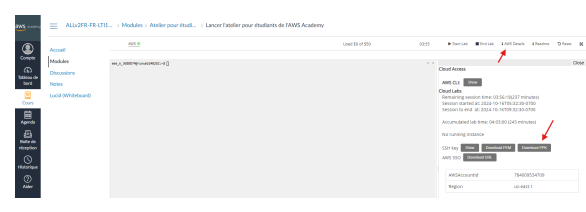

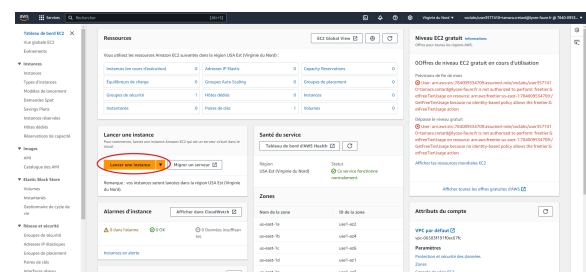
4.7. Installer MariaDB sur le serveur

Afin d'installer MariaDB sur le serveur, voici les étapes à suivre:

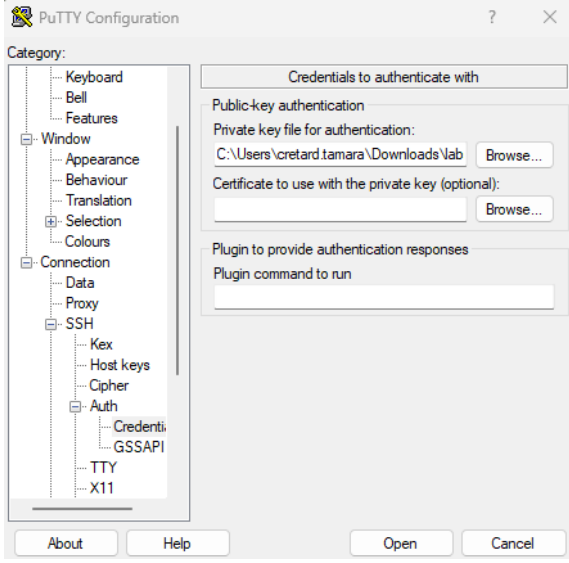
<p>Installer MariaDB avec <code>sudo apt install mariadb-server -y</code></p>	<pre>sudo apt install mariadb-server -ymariadb-server -y</pre>
<p>Vérifier que MariaDB est actif avec <code>sudo systemctl status mariadb</code></p>	<pre>root@kali:~/# sudo systemctl status mariadb mariadb.service - MariaDB 10.5.26 database server Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled) Active: active (running) since Wed 2020-12-11 14:28:32 CEST; 3min ago Docs: man:mariadb(8) https://mariadb.com/kb/en/library/systemd/ Main PID: 11793 (mysqld) Status: "Faking your SQL requests now..." Tree: s (Limit: 15229) Memory: 26.2M CPU: 539ms CGroup: /system.slice/mariadb.service └─11793 /usr/sbin/mysqld déc. 11 14:28:32 debian mariadb[11793]: Version: '10.5.26-MariaDB-0+deb11u2' socket: '/run/mysqld/mysqld.sock' port: 3306 déc. 11 14:28:32 debian systemd[1]: Started MariaDB 10.5.26 database server. déc. 11 14:28:32 debian /etc/mysql/debian-start[11719]: Upgrading MySQL tables if necessary. déc. 11 14:28:33 debian /etc/mysql/debian-start[11722]: Loading for 'mariadb' as: /usr/bin/mariadb déc. 11 14:28:33 debian /etc/mysql/debian-start[11722]: Loading for 'mariadb-check' as: /usr/bin/mariadb-check déc. 11 14:28:33 debian /etc/mysql/debian-start[11722]: This installation of MariaDB is already upgraded to 10.5.26-MariaDB. déc. 11 14:28:33 debian /etc/mysql/debian-start[11722]: There is no need to run mysql_upgrade again for 10.5.26-MariaDB. déc. 11 14:28:33 debian /etc/mysql/debian-start[11722]: You can use --force if you still want to run mysql_upgrade déc. 11 14:28:33 debian /etc/mysql/debian-start[11731]: Checking for insecure root accounts. déc. 11 14:28:33 debian /etc/mysql/debian-start[11735]: Triggering myisam-recover for all MyISAM tables and aria-recover lines 1-29/29 (END)</pre>

4.8. Mettre en place un serveur Linux Debian dans AWS

A présent, il faut mettre en place un serveur Linux Debian dans AWS:

<p>Aller sur le site awsacademy, dans "cours" puis "lancer l'atelier pour étudiants de l'AWS Academy". Cliquer sur le bouton "Start Lab" et attendre que le bouton AWS devienne vert.</p>	
<p>Aller dans "AWS Details" et télécharger le fichier PPK</p>	
<p>Aller dans la console AWS, puis rechercher le service AWS et cliquer dessus</p>	
<p>Dans la fenêtre qui s'ouvre, cliquer sur "lancer une instance".</p>	

<p>Donner un nom à l'instance et sélectionner Debian.</p>	
<p>Sélectionnez le nom de la paire de clés "vokey". Cliquez ensuite sur "lancer l'instance".</p>	
<p>Une fois le lancement de l'instance réussi, cliquez sur "Afficher toutes les instances" puis, cliquez sur l'ID de votre instance.</p>	
<p>Notez l'adresse IP publique de votre machine ainsi que son nom DNS public.</p>	
<p>Sur le PC, lancer l'application Putty. Dans la zone « Host Name (or IP address), entrer l'adresse IP publique</p>	

<p>Dans le volet Catégorie, développer Connexion, développer SSH, puis développer Auth et choisir Credentials. Suivre les instructions suivantes :</p> <ol style="list-style-type: none"> 1. Choisir Parcourir. 2. Sélectionner le fichier .ppk que téléchargé précédemment <p>Cliquer sur Open pour se connecter à la machine Debian</p>	
<p>Dans la fenêtre qui s'ouvre cliquer sur "accept" et entrer admin</p>	
<p>Mettre à jour le système avec <code>sudo apt update && sudo apt upgrade -y</code></p>	<pre>admin@ip-172-31-22-107:~\$ sudo apt update && sudo apt upgrade -y</pre>
<p>Installer Apache avec <code>sudo apt install apache2 -y</code></p>	<pre>admin@ip-172-31-22-107:~\$ sudo apt install apache2 -y</pre>
<p>Utilisez la commande systemctl pour configurer le serveur Web Apache afin qu'il soit lancé à chaque démarrage système.</p>	<pre>sudo systemctl enable apache2</pre>

4.9. Créer un nom de domaine DNS

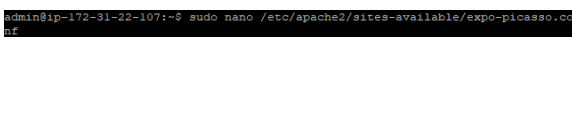
Maintenant, il nous faut créer un domaine DNS pour notre site d'exposition:

<p>Aller sur https://www.cloudns.net/, sélectionner "Sign-up for free" et créer un compte</p>	
<p>Cliquer sur "Create Zone" dans DNS Hosting</p>	

<p>Sélectionner "Free Zone"</p>	
<p>Entrer le nom de domaine</p>	
<p>Aller dans A → L'enregistrement "A" associe un nom d'hôte à une adresse IP</p>	
<p>Cliquer sur "Add new record"</p>	
<p>Entrer le Host ainsi que l'adresse IP du serveur</p>	

4.10. Configurer Apache du serveur AWS pour utiliser le nom de domaine

Puis, il nous faut configurer Apache du serveur AWS pour utiliser le nom de domaine:

<p>Créer le fichier de configuration avec <code>sudo nano /etc/apache2/sites-available/expo-picasso.conf</code></p>	
---	--

<p>Modifier le fichier pour ajouter le nom de serveur et la page index</p>	<pre><VirtualHost *:80> ServerAdmin webmaster@localhost ServerName www.expo-picasso.ip-ddns.com DocumentRoot /var/www/html/expo-picasso DirectoryIndex expo-picasso.html <Directory /var/www/html/expo-picasso> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined </VirtualHost> <VirtualHost *:443> ServerAdmin webmaster@localhost ServerName www.expo-picasso.ip-ddns.com DocumentRoot /var/www/html/expo-picasso DirectoryIndex expo-picasso.html ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined <Directory /var/www/expo-picasso> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> </VirtualHost></pre>
<p>Activer le site avec <i>sudo a2ensite expo-picasso.conf</i> et redémarrer Apache avec <i>sudo systemctl reload apache2</i></p>	<pre>admin@ip-172-31-22-107:~\$ sudo a2ensite expo-picasso.conf</pre>

4.11. [Installer Certbot pour générer automatiquement un certificat SSL](#)

Enfin, il faut installer Certbot pour générer automatiquement un certificat SSL:

<p>Installer Snapd (Certbot est généralement installé via Snapd): <i>sudo apt install snapd</i></p>	<pre>admin@ip-172-31-22-107:~\$ sudo apt install snapd</pre>
<p>Activer Snapd avec <i>sudo systemctl enable --now snapd.socket</i></p>	<pre>admin@ip-172-31-22-107:~\$ sudo systemctl enable --now snapd.socket</pre>
<p>Installer Certbot via Snapd avec: <i>sudo snap install --classic certbot</i></p>	<pre>admin@ip-172-31-22-107:~\$ sudo snap install --classic certbot</pre>
<p>Faire en sorte que la commande certbot peut être utilisée: <i>sudo ln -s /snap/bin/certbot /usr/bin/certbot</i></p>	<pre>admin@ip-172-31-22-107:~\$ sudo ln -s /snap/bin/certbot /usr/bin/certbot</pre>
<p>Créer un fichier conf: <i>sudo nano /etc/apache2/sites-enabled/expo-picasso.conf</i></p>	<pre>admin@ip-172-31-22-107:~\$ sudo nano /etc/apache2/sites-enabled/expo-picasso.conf</pre>

<p>Obtenir un certificat automatiquement avec certbot: <i>sudo certbot --apache</i></p>	<pre> admin@ip-172-31-22-107:~\$ sudo certbot --apache Saving debug log to /var/log/letsencrypt/letsencrypt.log Which names would you like to activate HTTPS for? 0. I don't know what to do 1. I want to activate HTTPS for: ----- 1: www.expo-picasso.ip-ddns.com ----- Select the appropriate numbers separated by commas and/or spaces, or leave input blank to select all options shown (Enter 'c' to cancel): 1 Certificate not yet due for renewal. You have an existing certificate that has exactly the same domains or certificate name you requested and isn't close to expiry: /etc/letsencrypt/live/www.expo-picasso.ip-ddns.com.conf What would you like to do? 0. I want to renew all my existing certificates 1. I want to renew only the following certificate(s): ----- 1: www.expo-picasso.ip-ddns.com ----- Select the appropriate numbers separated by commas and/or spaces, or leave input blank to select all options shown (Enter 'c' to cancel): 2 Renewing an existing certificate for www.expo-picasso.ip-ddns.com Successfully received certificate. Certificate is saved at: /etc/letsencrypt/live/www.expo-picasso.ip-ddns.com/fullchain.pem Key is saved at: /etc/letsencrypt/live/www.expo-picasso.ip-ddns.com/privkey.pem This certificate expires on 2023-03-26. These files will be updated when the certificate renews. Certbot has set up a scheduled task to automatically renew this certificate in the background. Deploying certificate Successfully deployed certificate for www.expo-picasso.ip-ddns.com to /etc/apache2/sites-enabled/seo-conseils-site.conf Your existing certificate has been successfully renewed, and the new certificate has been installed. ----- If you like Certbot, please consider supporting our work by: * Donating to EFF: https://letsencrypt.org/donate * Donating to EFF: https://eff.org/donate-in </pre>
<p>Modifier le fichier conf: <i>sudo nano /etc/apache2/sites-enabled/expo-picasso.conf</i></p>	<pre> <VirtualHost *:80> ServerAdmin webmaster@localhost ServerName www.expo-picasso.ip-ddns.com DocumentRoot /var/www/html/seo-conseils-site DirectoryIndex seo-conseils.html <Directory /var/www/html/seo-conseils-site> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined </VirtualHost> <VirtualHost *:443> ServerAdmin webmaster@localhost ServerName www.expo-picasso.ip-ddns.com DocumentRoot /var/www/html/seo-conseils-site DirectoryIndex seo-conseils.html ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined SSLEngine on SSLCertificateFile /etc/letsencrypt/live/www.expo-picasso.ip-ddns.com/fullchain.pem SSLCertificateKeyFile /etc/letsencrypt/live/www.expo-picasso.ip-ddns.com/privkey.pem <Directory /var/www/html/seo-conseils-site> Options Indexes FollowSymLinks AllowOverride All Require all granted </Directory> </VirtualHost> </pre>
<p>Redémarrer Apache: <i>sudo systemctl restart apache2</i></p>	<pre> admin@ip-172-31-22-107:~\$ sudo systemctl restart apache2 </pre>