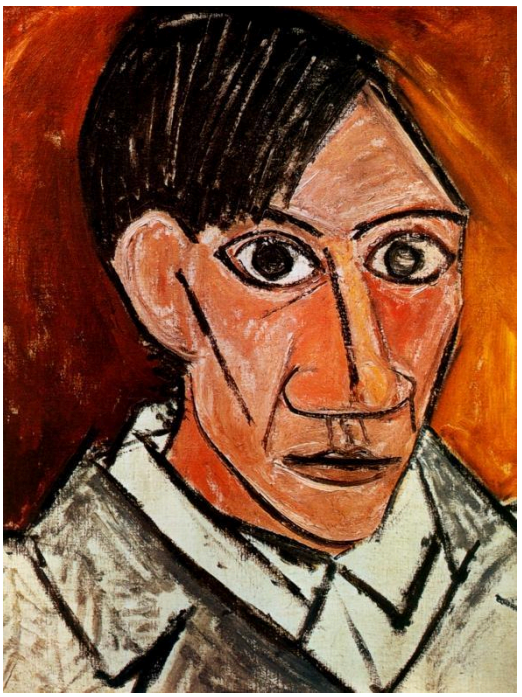


Projet d’Exposition Picasso

Tamara Crétard - 15.01.2024

Table des matières

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.

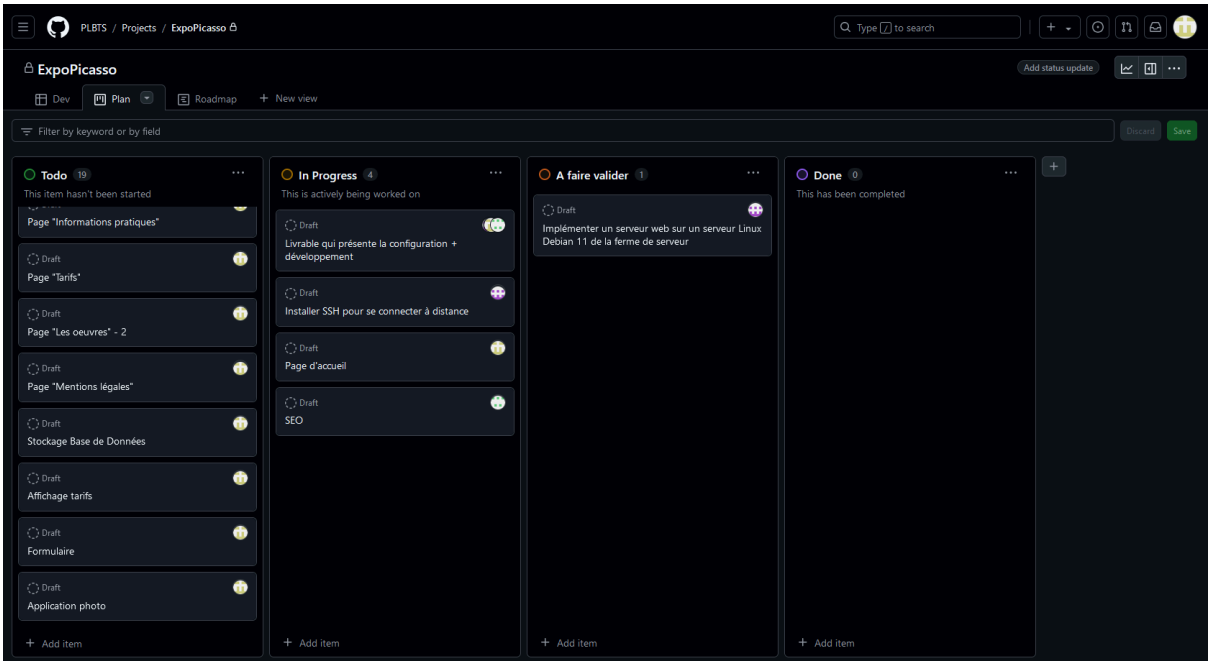
Ces œuvres seront exposées du 10 au 22 mai 2025 dans la salle d'exposition temporaire du musée.

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

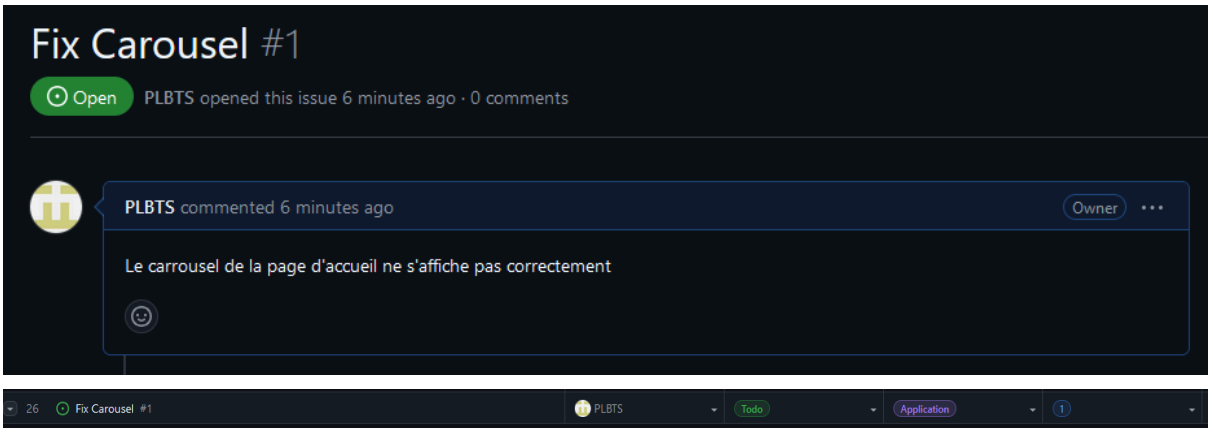
2.1. Plan

Afin de mener à bien ce projet, nous avons utiliser GitHub pour visualiser les différentes étapes é 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:



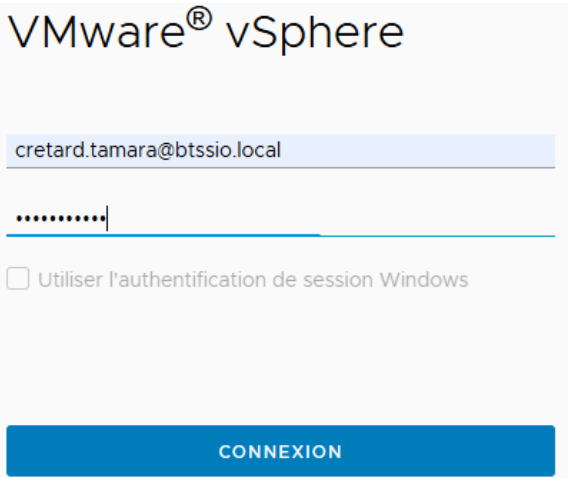
3. Solution applicative

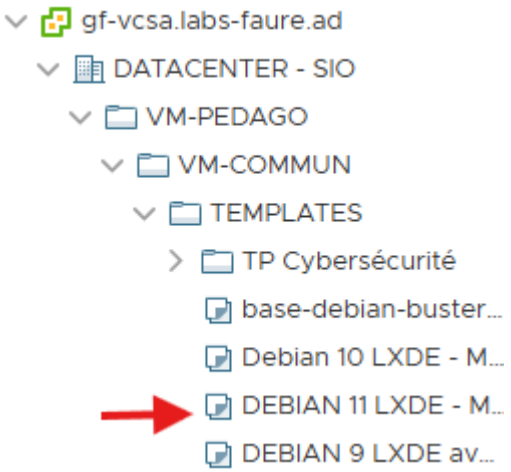
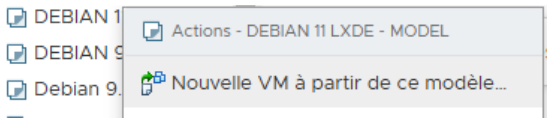
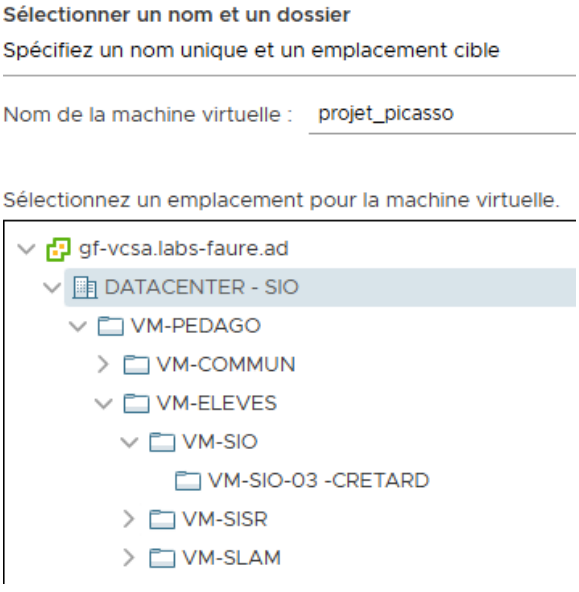
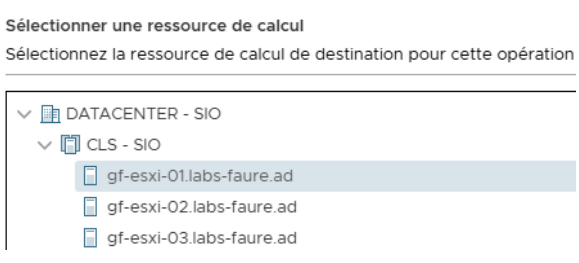
Je ne me suis pas occupée de la solution applicative et mes camarades n'ont pas souhaité prendre de captures d'écran pour montrer l'évolution du projet.

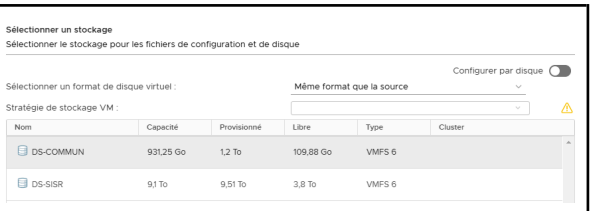
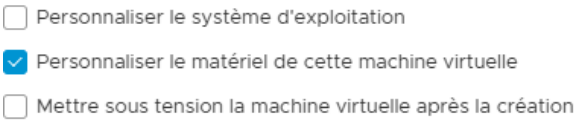
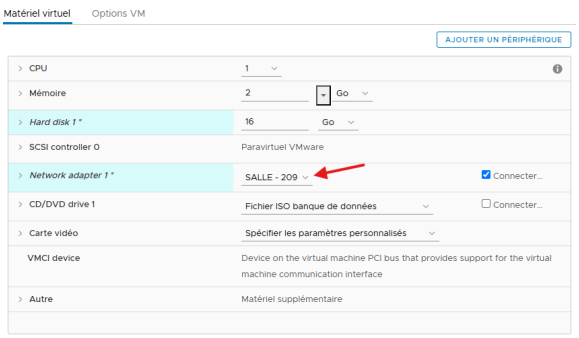
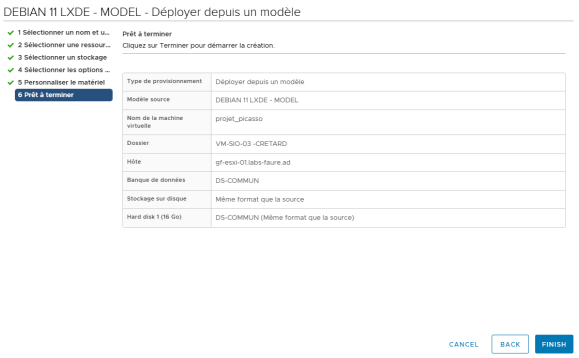
4. Hébergement

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:

Pour accéder à la ferme de serveurs, se rendre sur https://gf-vcsa.labs-faure.ad/ et se connecter avec ses identifiants	

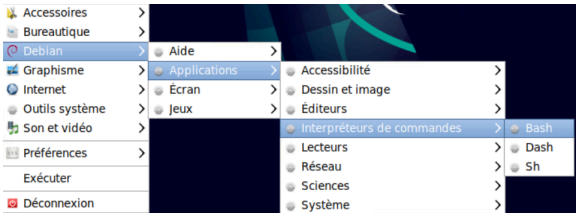
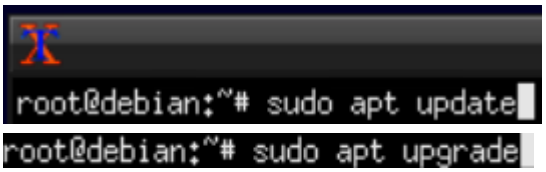
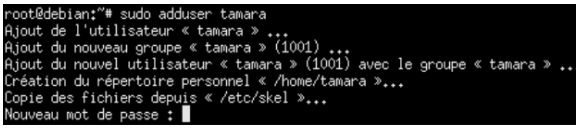
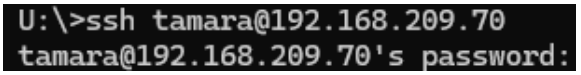
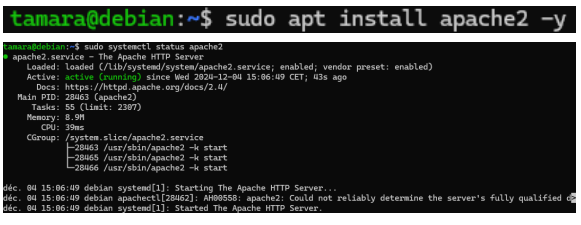
Une fois connecté, aller dans le menu « VM et Modèles »	
Trouver le template DEBIAN11 LXDE dans : VM-PEDAGO / VM-COMMUN / TEMPLATES	
faire click-droit, et « Nouvelle VM à partir de ce modèle »	
lui donner un nom et la ranger dans le répertoire situé dans VM-PEDAGO / VM-ELEVES.	
Pour la ressource de calcul, sélectionner au choix l'une des 3 ressources disponibles	

Pour le stockage, sélectionner DS-COMMUN	
Cocher « Personnaliser le matériel de cette machine virtuelle »	
Comme « Network Adapter 1 », vous sélectionnez « SALLE – 209 »	
Déployer la machine en cliquant sur “FINISH”	

4.2. Installer SSH pour se connecter à distance

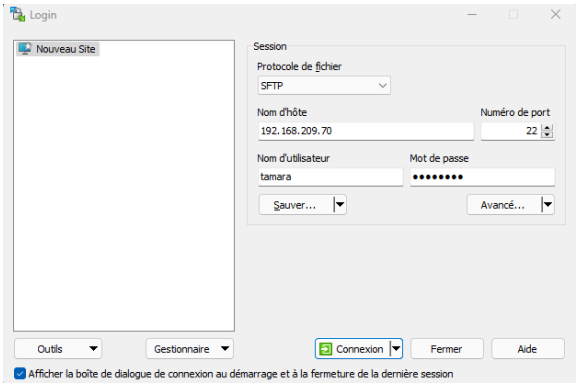
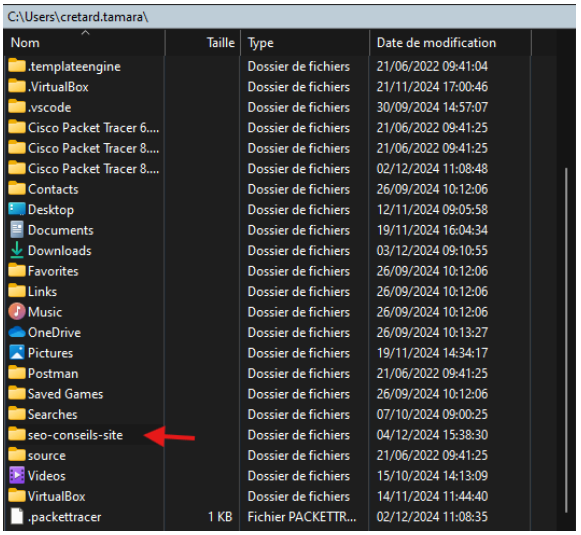
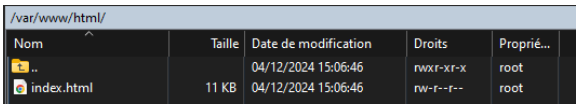
Afin de rendre possible la connexion à distance, il est nécessaire d’installer SSH sur notre machine virtuelle. Pour se faire, voici les étapes à suivre:

Démarrer la machine virtuelle et se connecter	

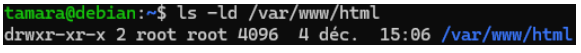
Ouvrir le terminal en allant dans le menu Debian -> Applications -> Interpréteurs de commandes -> Bash.	
Mettre à jour votre système avec la commande : *sudo apt update* suivi de la commande *sudo apt ugrade*	
Installer le paquet openssh-server avec la commande: *sudo apt install openssh-server*	`sudo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

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:

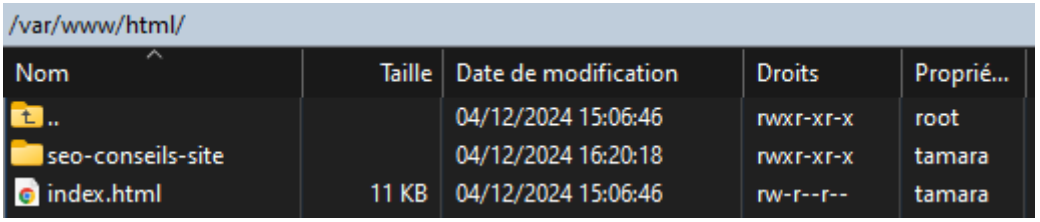
Entrer les informations concernant le serveur web et se connecter	
Trouver les fichiers à transférer dans le panneau de gauche qui représente le PC hôte	
Dans le panneau de droite, naviguer vers l'emplacement où Apache héberge les fichiers web: /var/www/html	
Glisser-déposer les fichiers ou dossiers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

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:

Se connecter à la machine virtuelle depuis le terminal du PC hôte	
Vérifier qui est le propriétaire du répertoire	

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

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



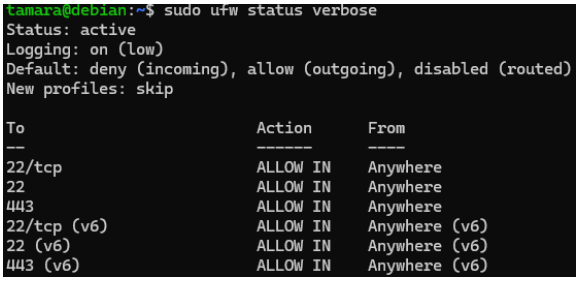
4.4. Activer un firewall sur le serveur web

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	`ssh tamara@192.168.209.70` `tamara@192.168.209.70's password:`
Mettre à jour le système avec la commande *sudo apt update && sudo apt upgrade -y*	`tamara@debian:~$ sudo apt update && sudo apt upgrade -y`
Installer UFW avec la commande *sudo apt install ufw*	`tamara@debian:~$ sudo apt install ufw`

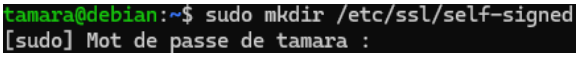
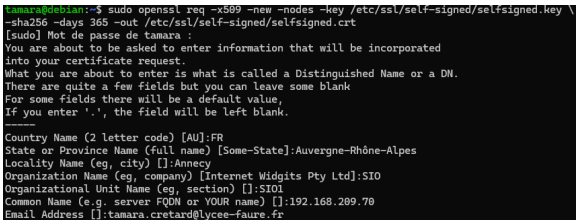
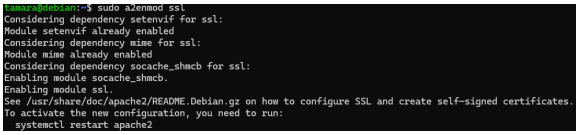
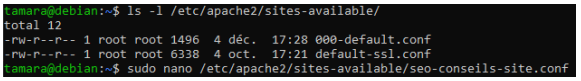
Ajouter les règles suivantes: *sudo ufw allow ssh* , *sudo ufw allow https* , *sudo ufw allow 22*	`tamara@debian:~$ sudo ufw allow ssh`
	`Rules updated`
	`Rules updated (v6)`
	`tamara@debian:~$ sudo ufw allow 22`
	`Rules updated`
	`Rules updated (v6)`
	`tamara@debian:~$ sudo ufw allow https`
	`Rules updated`
	`Rules updated (v6)`

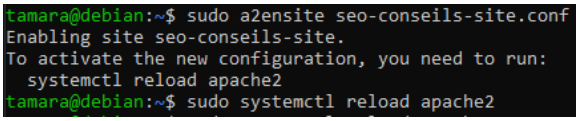
Activer UFW avec la commande suivantes: *sudo ufw enable*	`tamara@debian:~$ sudo ufw enable`
	`Command may disrupt existing ssh connections. Proceed with operation (y|n)? y`
	`Firewall is active and enabled on system startup`

Vérifier que tout fonctionne correctement avec *sudo ufw status verbose*	

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:

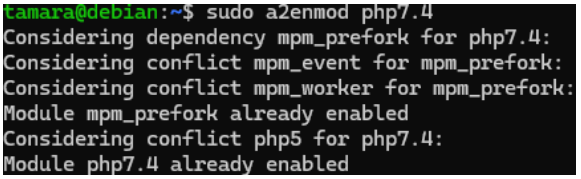
Se connecter à la machine virtuelle depuis le terminal du PC hôte	
Créer un répertoire pour stocker les certificats avec *sudo mkdir /etc/ssl/self-signed*	
Générer une clé privée avec *sudo openssl genrsa -out /etc/ssl/self-signed/selfsigned.key 2048*	
Créer une demande de certificat (CSR) et un certificat auto-signé avec la commande *sudo openssl req -x509 -new -nodes -key /etc/ssl/self-signed/selfsigned.key \ -sha256 -days 365 -out /etc/ssl/self-signed/selfsigned.crt*	
Activer le module SSL avec *sudo a2enmod ssl* et redémarrer Apache avec la commande *sudo systemctl restart apache2*	
Lister les fichiers avec *ls -l /etc/apache2/sites-available/* s'il n'y a pas notre site, créer le fichier de configuration avec *sudo nano /etc/apache2/sites-available/seo-conseils-site.conf*	

Modifier le contenu du fichier: *sudo nano /etc/apache2/sites-available/seo-conseils-site.conf*	
Activer le site avec sudo a2ensite *seo-conseils-site.conf et redémarrer* Apache avec *sudo systemctl reload apache2*	

4.6. Installer PHP sur le serveur

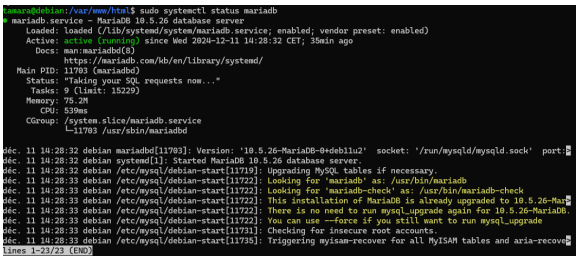
Afin d'installer PHP sur le serveur, voici les étapes à suivre:

Mettre à jour le système avec *sudo apt update && sudo apt upgrade -y*	
Installer PHP et les extensions avec: *sudo apt install php libapache2-mod-php php-mysql -y* **php** : le langage PHP **libapache2-mod-php** : permet à Apache d'interpréter les fichiers PHP **php-mysql** : extension pour que PHP interagisse avec MariaDB	
Vérifier l'installation avec *php -v*	
Vérifier que le module est bien installé avec *dpkg -l | grep libapache2-mod-php*	

Vérifier les modules disponibles: *ls /usr/lib/apache2/modules | grep php*	
Activer le module voulu avec *sudo a2enmod php7.4* et redémarrer Apache *sudo systemctl restart apache2*	
Modifier le fichier de configuration Apache en ajoutant un DirectoryIndex index.php et redémarrer denouveau Apache	
Tester PHP en créant un fichier de test dans le répertoire par défaut d'Apache: sudo nano /var/www/html/seo-conseils-site/index.php et en ajoutant: <?php echo "PHP fonctionne !"; ?> en allant sur 192.168.209.70/info.php il devrait y avoir une page d'informations PHP	

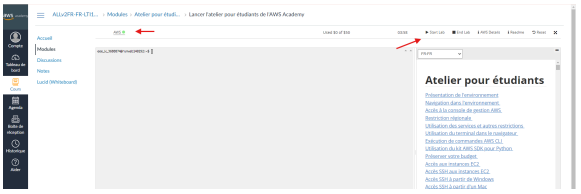
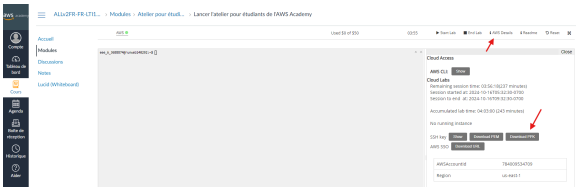
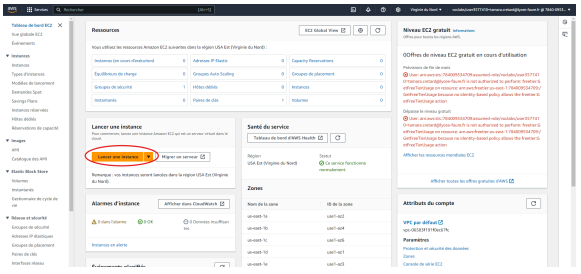
4.7. Installer MariaDB sur le serveur

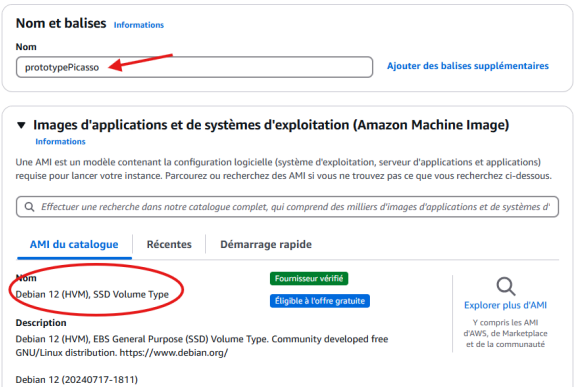
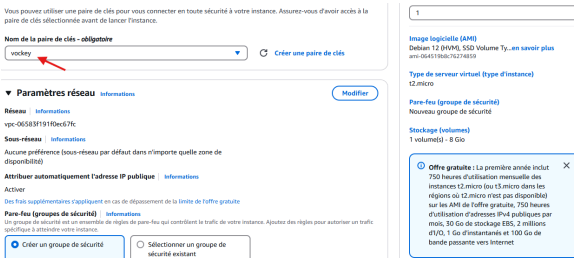
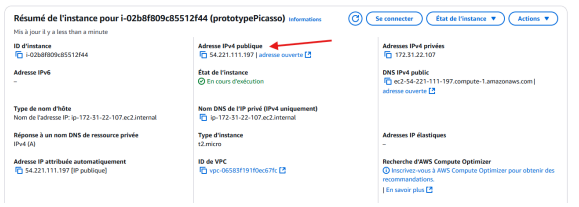
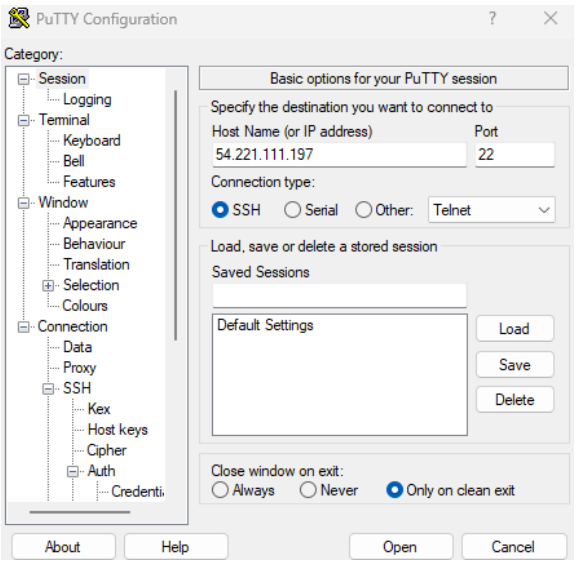
Afin d'installer MariaDB sur le serveur, voici les étapes à suivre:

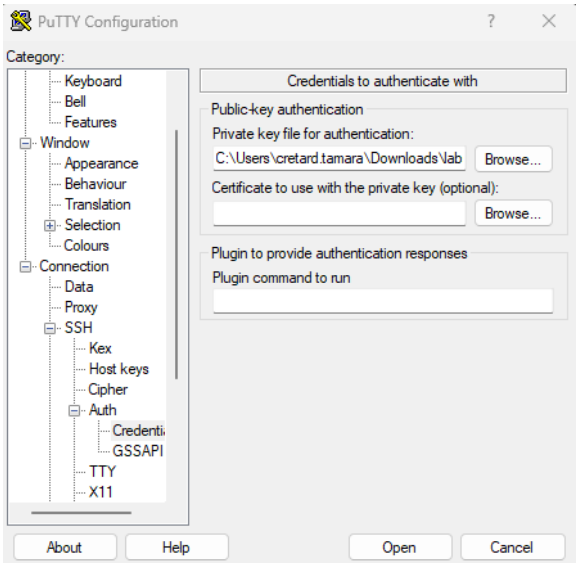
Installer MariaDB avec *sudo apt install mariadb-server -y*	`sudo apt install mariadb-server -ymariadb-server -y`
Vérifier que MariaDB est actif avec *sudo systemctl status mariadb*	

4.8. Mettre en place un serveur Linux Debian dans AWS

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

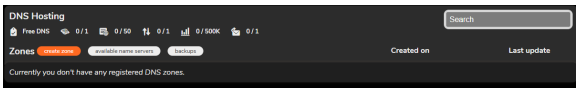
Aller sur le site awsacademy, dans “cours” puis “lancer l’atelier pour étudiants de l’AWS Academy”. Cliquer sur le bouton “Start Lab” et attendre que le bouton AWS devienne vert.	
Aller dans “AWS Details” et télécharger le fichier PPK	
Aller dans la console AWS, puis rechercher le service AWS et cliquer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

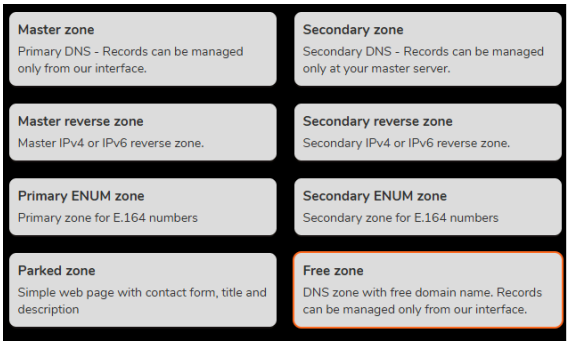
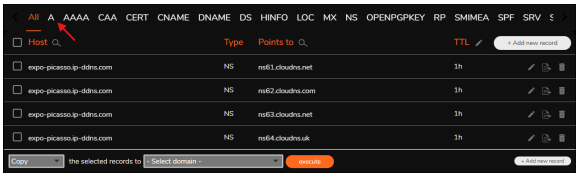
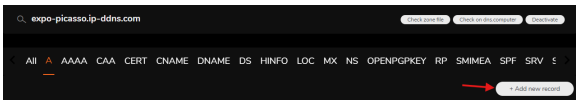
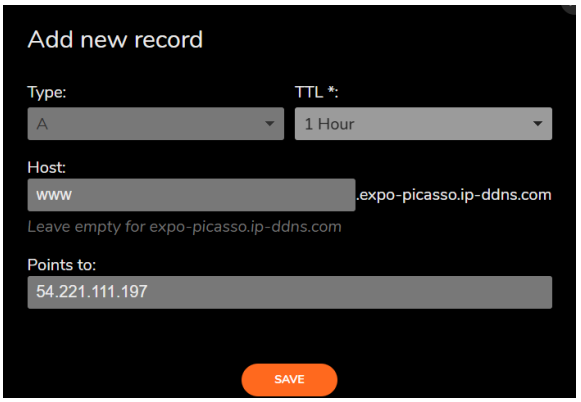
Donner un nom à l'instance et sélectionner Debian.	
Sélectionnez le nom de la paire de clés "vockey". Cliquez ensuite sur "lancer l'instance".	
Une fois le lancement de l'instance réussi, cliquez sur "Afficher toutes les instances" puis, cliquez sur l'ID de votre instance.	
Notez l'adresse IP publique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

Dans le volet **Catégorie**, développer **Connexion**, développer **SSH**, puis développer **Auth** et choisir **Credentials**. Suivre les instructions suivantes : 1. Choisir **Parcourir**. 2. Sélectionner le fichier .ppk que téléchargé précédemment Cliquer sur Open pour se connecter à la machine Debian	
Dans la fenêtre qui s'ouvre cliquer sur "accept" et entrer admin	
Mettre à jour le système avec *sudo apt update && sudo apt upgrade -y*	`admin@ip-172-31-22-107:~$ sudo apt update && sudo apt upgrade -y`
Installer Apache avec *sudo apt install apache2 -y*	`admin@ip-172-31-22-107:~$ sudo apt install apache2 -y`
Utilisez la commande systemctl pour configurer le serveur Web Apache afin qu'il soit lancé à chaque démarrage système.	`sudo systemctl enable apache2`

4.9. Créer un nom de domaine DNS

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

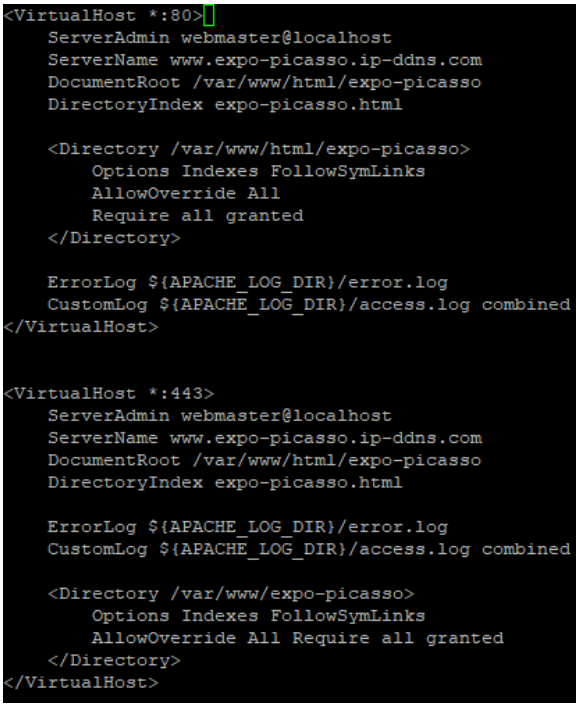
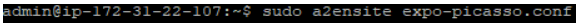
Aller sur https://www.cloudns.net/, sélectionner "Sign-up for free" et créer un compte	
Cliquer sur "Create Zone" dans DNS Hosting	

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

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

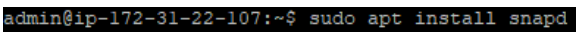
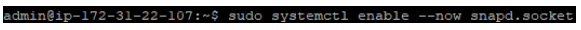
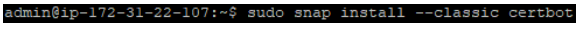
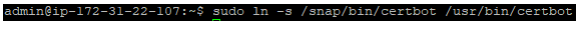
Puis, il nous faut configurer Apache du serveur AWS pour utiliser le nom de domaine:

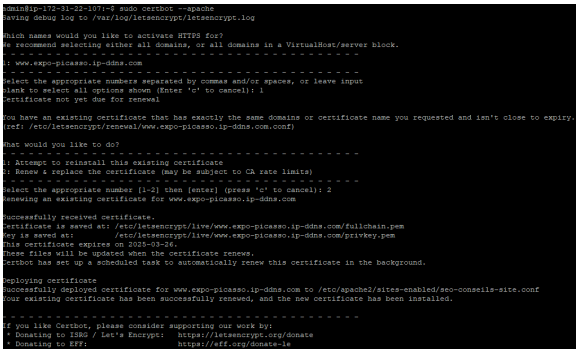
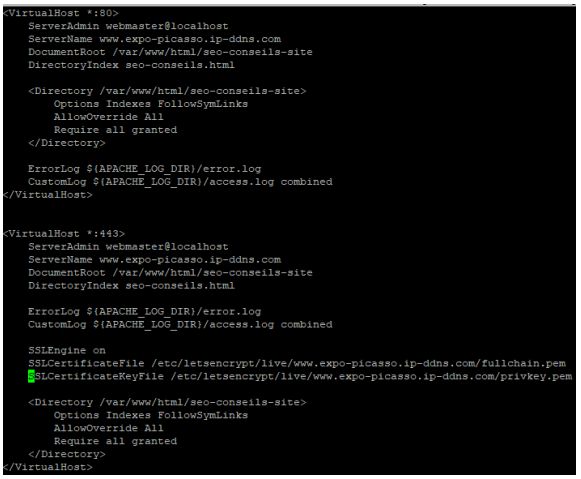
Créer le fichier de configuration avec *sudo nano /etc/apache2/sites-available/expo-picasso.conf*	

Modifier le fichier pour ajouter le nom de serveur et la page index	
Activer le site avec *sudo a2ensite expo-picasso.conf* et redémarrer Apache avec *sudo systemctl reload apache2*	

4.11. Installer Certbot pour générer automatiquement un certificat SSL

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

Installer Snapd (Certbot est généralement installé via Snapd): *sudo apt install snapd*	
Activer Snapd avec *sudo systemctl enable --now snapd.socket*	
Installer Certbot via Snapd avec: *sudo snap install --classic certbot*	
Faire en sorte que la commande certbot peut être utilisée: *sudo ln -s /snap/bin/certbot /usr/bin/certbot*	
Créer un fichier conf: *sudo nano /etc/apache2/sites-enabled/expo-picasso.conf*	

Obtenir un certificat automatiquement avec certbot: *sudo certbot --apache*	
Modifier le fichier conf: *sudo nano /etc/apache2/sites-enabled/expo-picasso.conf*	
Redémarrer Apache: *sudo systemctl restart apache2*