

Préambule

Cette SAÉ (Situation d'Apprentissage et d'Évaluation) vise à installer et configurer des services réseaux permettant de développer et de déployer des applications informatiques communicantes. L'objectif étant de répondre à un besoin exprimé par un client. Cette SAÉ est relative aux ressources R2.04 et R2.05.

Votre mission requiert l'installation et la configuration des services nécessaires au développement d'un site web. En effet, un site Web seul ne suffit pas, il lui faut à minima des modules supplémentaires tels qu'un langage de programmation et une base de données pour générer du contenu dynamique et personnalisé pour différents utilisateurs de différents profils. Souvent, lors de l'installation des services, il est question de repérer et de bien paramétrer les fichiers de configuration du serveur et de ses différents modules. Dans notre cas, nous allons manipuler tout cela sous un environnement Linux avec un serveur Web de type Apache, une base de données de type MySQL, et un langage (code serveur) qui est le PHP.

- Sur le Moodle de la SAÉ, vous trouverez de la documentation Apache, MySQL et PHP dans la section « documentation ». Utilisez aussi le support de TD, comme documentation.

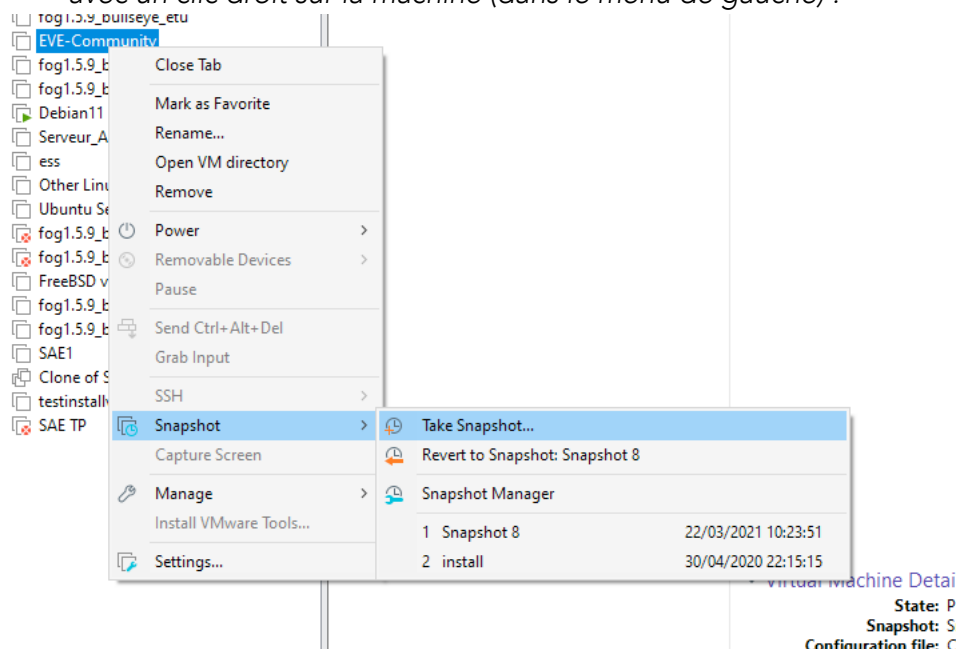
Environnement pratique : instructions importantes

- Nous allons utiliser une machine virtuelle dédiée (VM : debian 11 + apache + php + serveur MySQL) à la réalisation du cahier de charge du projet et au sujet de ce TP. Assurez-vous de bien suivre les instructions suivantes. [💡 trouvez les commandes exactes qui permettent de voir quelles versions exactes sont exigées pour cette SAÉ pour : debian, php et MySQL]. Il est important de garder ces mêmes versions pour l'homogénéité entre membres et surtout pour la correction attendu de vos travaux. :
- **Nouvelle procédure [sous Linux] :**
 - Cette procédure a été présentée lors de votre première séance avec l'enseignant
 - Connectez-vous sous Linux afin de lancer votre VM
 - **⚠ Ne la refaites pas si c'est déjà fait pendant ma première séance de TD. ⚠**
Manipulation à faire une seule fois. Cette manipulation consiste tout simplement à copier le dossier caché nommé /FILER/anonymous/.vmare **vers la racine** (et non un sous dossier) de votre dossier personnel (exemple : /home/etudiant/tlemloum/)
 - Lancez : vmware workstation. Cliquez (une seule fois) sur le nom SAE à gache
 - Cliquez sur le bouton triangle (startup)
 - Loguez vous avec root / lannion

- Rappel de la procédure de partage de dossiers/fichiers entre VM et le système hôte. ⚠ Ne la refaites pas si c'est déjà fait pendant ma première séance de TD

:

- Cliquez droit sur notre VM SAE >> setting >> Options >> Shared Folders >> Always enabled >> Add >> Name (à renseigner): première lettre du prénom suivi par nom de famille sans espace (ex. tlemloum) >> Browse (choisir un dossier ou créer le dans la racine du dossier home pas dans un sous dossier), idéalement nommez le sae203 >>>OK >>> save
 - Le dossier sous la VM est désormais accessible sous
 - /mnt/hgfs/tlemloum (tlemloum étant le dossier indiqué dans les settings de la VM)
 - le dossier en dehors de la VM est désormais accessible sous votre dossier home, sous dossier sae203 :
 - /home/etudiant/tlemloum/sae203
- **Ancienne procédure [sous Windows] :**
 - Connectez-vous sous Windows afin de lancer votre VM
 - Il est **important de commencer par copier le dossier SAE** (et non pas SAETP) depuis le lecteur D **sur la racine de votre espace lecteur Z** (si votre Z est à cours d'espace, faites le ménage dans vos propres dossiers/fichiers existants).
 - Dans votre dossier SAE fraîchement copié sur Z, lancez votre machine en double cliquant sur le fichier nommé **SAE.vmx**.
 - Faites valider par l'enseignant :
 - La copie du bon dossier de la SAE vers votre lecteur Z
 - Le bon lancement de votre VM
 - Les logins « root » (de Debian 11 et du serveur MySQL) ont le mot de passe « lannion »
 - Pour faire un snapshot (une sauvegarde) de la VM, **il est important de procéder comme suit (sinon, si vous procédez autrement, vous allez saturer votre lecteur Z)** :
 - Vous éteignez votre machine VM et ensuite vous pouvez faire le snapshot avec un clic droit sur la machine (dans le menu de gauche) :



- Pour des raisons de sécurité sur nos réseaux de production, vous n'avez pas accès Internet sous la VM avec votre compte root. Lorsqu'il s'agit d'installer le paquetage avec **apt-get install**, utiliser plutôt la commande : **dpkg -i <nom du paquetage>**
 - Les deux paquetages que nous aurons besoins (seulement lorsque c'est demandé plus tard dans le texte, sont déjà sauves dans le dossier /usr/local/src de votre VM). Il s'agit du paquetage php-mysql et de sa dépendance. Pour des raisons pédagogiques, **ne l'installez pas tout de suite, ça sera demandé plus tard dans le texte.**
- **Le service Web avec le serveur Apache**
- Comme dit en TD, pour cette SAÉ, nous optons pour la mise en œuvre d'un serveur Web assuré par le paquetage logicielle Apache. Ce dernier est conçu sur le principe de la modularité. Par conséquent, Apache permet à un administrateur de serveur Web d'ajouter ou de supprimer différents modules afin d'étendre -au besoin- les fonctionnalités primaires du serveur et également améliorer ses performances.
- Avant toutes modification d'un fichier de configuration, créez une version qui garde le même nom et la même extension d'origine mais qui finit par l'extension « .old ». Exemple : « apache2.conf » en « apache2.conf.old ». Cela vous permettra de disposer dans tous les cas d'un fichier opérationnel par défaut en cas de « casse »
- 1. Prenez connaissance de votre environnement pratique (VM de la SAÉ).
- 2. Vérifiez le statut d'exécution du service Web (assuré par le paquetage apache2) avec la commande : **# systemctl status apache2**. Relevez les informations pertinentes qui montre qu'il s'exécute continuellement (ce qui est normal car c'est un serveur). Relevez à quelle heure exacte, le serveur a été lancé la toute dernière fois.



À la place de « **systemctl** », vous pouvez aussi utiliser la commande « **service** ». Mais attention, inversez l'ordre entre « **statut** » et le nom du service. Exemple : **service mysql status**

3. Dans quel fichier de configuration de votre serveur Web, se trouve la directive « DocumentRoot » ? Quelle est sa valeur ? Rappelez l'utilité de cette directive?
4. Repérez le chemin du binaire (exécutable) du service Web (apache2) en utilisant la commande « type » et l'option « a »
5. Identifier la version exacte du serveur Apache. Pourquoi il est important de connaitre la version exacte ?



La dernière version du serveur Apache2, au moment de l'écriture de ce texte, est la version 2.4.53 (sortie le 14 mars 2022)

6. Listez les modules installés nativement (compilé dans le noyau) du le serveur avec la commande « **apache2** » et l'argument de commande « **l** ».
7. D'après vous, quel est le rôle du module « **mod_log_config.c** » ?
8. La commande précédente ne liste pas tous les modules chargés dynamiquement par le serveur Web (comme par exemple ceux chargés avec la directive nommée « **LoadModule** »). Pour lister tous les modules (statique et dynamique), utilisez la commande : **apache2 -M** et relevez les modules affichés.



Pour lister les modules utilisés par le serveur, on peut également utiliser la commande « `apachectl -M` »

9. Notez la différence entre les commandes « `apache2 --version` » et « `apache2 -v` ». « `apache2 -v` » ne donne pas d'erreur mais « `apache2 --version` » peut le faire. S'il y a un message d'erreur, il faut le corriger.
- En fait le message d'erreur, s'il y en a, ce n'est pas une erreur grave mais c'est ce qu'on appelle « un avertissement amical » (en anglais « friendly warning ») et la réponse est dans le message d'erreur.



Solution au « warning »: dans le fichier `/etc/apache2/apache2.conf` (et non pas dans `/etc/apache2/sites-enabled`) on ajoute, au début par exemple, pour préciser le nom de domaine de notre serveur, la ligne suivante : « `ServerName localhost` » (sans les guillemets bien sûr !). N'oubliez pas de recharger ou relancer le serveur Apache pour tester

- Création d'une partie administration (« privée ») du serveur Web accessible uniquement par login/mot de passe géré directement par le serveur Web (et non pas par une base de donnée):
 - Créer un dossier « private » dans la racine du serveur web (`/var/www/html/`)
 - Créer dedans un fichier `index.html`. Personnaliser le contenu et le titre de cette page
 - Depuis la page d'accueil du serveur, créer un lien qui permet d'accéder à l'espace privé (`administration`) et vice-versa.
 - Inspirez-vous du text TD pour n'autoriser l'accès au dossier « privé » qu'après authentification à l'utilisateur de login « admin » et de mot de passe « lannion »



Une fois l'authentification est réussie, votre navigateur Firefox (version 91.8.0esr (64 bits) sur votre VM) va garder les informations d'authentification dans son historique (même si vous êtes en navigation privée et même si vous supprimez les cookies et le cache!). Si vous avez besoin de tester l'authentification de nouveau, effacer l'historique, car c'est là où il garde ces informations.

• Le module PHP

La façon la plus simple de vérifier la configuration PHP, y compris pour voir les modules associées qui sont installés, est de créer un script php de test en utilisant la fonction `phpinfo()`.

- Repérer le dossier des pages Web de votre serveur Apache

1. Créez dedans un dossier secret et créer dedans un fichier texte nommé : `phpinfo.php` et incluant le code suivant :

```
<?php phpinfo(); ?>
```

- Pourquoi est il recommandé de créer un dossier secret ?
- Quelle est l'URL pour accéder à votre fichier php depuis un navigateur Web ?



Sous votre VM et avec un accès local à votre serveur Web, il est recommandé de vider le cache du navigateur car il peut -parfois- vous fournir un contenu ne reflétant pas les dernières modifications des pages du serveur. Sinon, pour ces manipulations, mettez votre

4. Accédez à cette URL. Peut-on confirmer après cet accès que le module PHP est activé par le serveur ?
5. Confirmez que le module PHP est bien activé par le serveur Web en vérifiant : (1) le contenu d'un dossier propre à Apache et (2) le contenu d'un fichier X (X est à préciser, indication : voir TD). (3) Vérifiez avec la commande « ls -l » appliqué à ce fichier X : que le fichier X n'est pas un fichier « classique »
6. Accédez à votre script php : phpinfo.php avec le navigateur et relevez les informations suivantes sur la page affichée par le serveur :
 - i. La version exacte du module PHP utilisé par notre serveur Web
 - ii. Le dossier de configuration du module PHP utilisé par notre serveur Web
 - iii. Le fichier de configuration de php pour le serveur Web
 - b. La valeur de l'étiquette appelée « short_open_tag »
 - c. À votre avis, dans quel fichier de configuration (chemin exacte) peut-on modifier la valeur de cette étiquette « short_open_tag »?
 - d. Si on devrait modifier la valeur de cette étiquette dans un fichier de configuration, faut-il recharger/relancer le serveur Web pour que la nouvelle valeur soit prise en compte ?



Le module PHP indique le début d'un code classique avec la syntaxe `<?php ... ?>`. Cette syntaxe peut être simplifiée vers `<? ... ?>` en activant le tag « short_open_tag » (i.e. en mettant sa valeur à On) dans le fichier `php.ini`. Attention : ce tag risque d'être enlevé dans les prochaines versions pour plusieurs raisons comme le conflit avec les balises XML de syntaxe `<?xml ... ?>`

7. À l'aide de la commande « whereis », trouvez le chemin du binaire (de l'exécutable si vous préférez) du module php
 - i. Confirmez, avec l'option -v du binaire, la version du module php obtenue précédemment
8. Désactiver le module PHP en ligne de commande
9. Faut-il relancer le serveur Apache pour prendre en charge la désactivation ? Si oui faites-le.
10. Quelle est la conséquence de cette désactivation ? Est-elle dangereuse dans le cas où il subsiste des fichiers PHP sur notre serveur ? Effectuez deux tests : un avec un accès URL et un test de vérification du contenu d'un dossier propre à Apache (à préciser) pour montrer la désactivation du PHP.
11. Réactiver le module PHP en ligne de commande. Mêmes questions que précédemment :
 - a. Faut-il redémarrer le serveur ? Si oui faites le
 - b. Effectuez les mêmes deux tests de la question précédente
12. Insérer dans le code HTML d'une page nommée « mapage.html » (au début par exemple) un code PHP arbitraire. Par exemple « `<?php echo "coucou, je suis un code php dans une page HTML !" ?>`. La page mapage.html est à créer dans la racine du serveur Web.
13. Faut-il redémarrer le serveur pour accéder à la page mapage.html ?
14. Essayez d'accéder à votre nouvelle page mapage.html, que constatez-vous ? justifiez votre réponse.

15. En modifiant le fichier de configuration du module PHP, dites à apache d'interpréter les fichiers d'extensions .html et .htm comme des fichiers php. Indication : voir le TD. Faites le test nécessaire pour confirmer que l'interprétation demandée est correcte.
16. Dupliquer le fichier phpinfo.php pour créer un nouveau fichier de nom « phpinfo » (sans l'extension .php). Essayez d'accéder à ce fichier « phpinfo » avec votre navigateur Web. Que remarquez-vous ?
17. En s'inspirant du TD, configurer Apache pour traiter les fichiers sans extension comme du code PHP et confirmer votre configuration par un test.
18. Apache maintenant interprète du code PHP dans du code HTML et c'est super ! Cependant, écrire du code avec la syntaxe `<?php echo "coucou, je suis un code php dans une page HTML !" ?>` peut être rendu plus facile avec une syntaxe simplifiée du genre : `<? echo "coucou, je suis un code php dans une page HTML !" ?>` (i.e. sans préciser à chaque fois php).
 - a. Dans votre page mapage.html changez la syntaxe du code php de `<?php echo "coucou, je suis un code php dans une page HTML !" ?>` vers `<? echo ... ?>`
 - b. Tester votre page depuis un navigateur (attention au cache du navigateur, il faut le vider de temps en temps ou mettez-vous en mode « vie privée stricte » pour éviter le cache pendant les manipulations). Que remarquez-vous ?
 - c. Proposez une solution et testez de nouveau pour confirmer que vous avez réussi à exécuter le code php avec une syntaxe `<? code; ?>` et non pas `<?php code; ?>`. Indication : « short_open_tag » !

• MySQL & PHP

1. Sécurisez le serveur MySQL avec la commande « `mysql_secure_installation` » (le mot de passe root du serveur MySQL doit rester : **lannion**)
2. Vérifier le statut d'exécution du service MySQL (assuré par le paquetage `mysql-server`). Et relevez les informations pertinentes qui montrent que le serveur MySQL s'exécute continuellement (ce qui est normal car c'est un serveur). Relevez à quelle heure exacte, le serveur a été lancé la toute dernière fois.
3. Identifier la version du serveur MySQL. Attention, l'argument à utiliser pour la commande `mysql` est « `-V` » et non « `-v` »



La dernière version du serveur `mysql-server`, au moment de l'écriture de ce texte, est la version 8.0.29 (sortie le 26 avril 2022)

4. Connectez-vous au serveur MySQL avec le login « root » et en utilisant la commande `mysql` et les bons arguments à trouver avec « `mysql --help` » si nécessaire. Le mot de passe configuré au préalable lors de l'installation est « lannion ».
5. Tout en restant connecté au serveur MySQL (en ligne de commande), afficher les bases de données existantes au niveau du serveur MySQL avec la commande (requête SQL) : « `SHOW DATABASES ;` »
6. Essayer d'avoir le même résultat (i.e. la liste des bases de données existantes sur le serveur) avec un code PHP que vous créez sur le serveur (voir TD). Confirmez le résultat en testant votre script avec un accès Web à votre serveur.
 - a. Sous certaines conditions, lorsqu'on essaie d'accéder à la liste des bases de données par le script php, le journal d'erreur d'apache (le `/var/log/apache/error.log`) peut nous donner l'erreur suivante :

```
[Sun May 01 17:00:10.816875 2022] [php7:error] [pid 6142] [client 127.0.0.1:47936] PHP Fatal error: Uncaught Error: Call to undefined function mysqli_connect() in /var/www/html/showdb.php:5\nStack trace:\n#0 {main}\n thrown in /var/www/html/showdb.php on line 5
```

b. Donnez une explication possible à cette erreur

c. Réparez cette erreur en installant le package nécessaire, referrez vous à la section

« **Environnement pratique : instructions importantes** »

7. Tout en restant connecté au serveur MySQL en ligne de commande, créez une base de données nommée avec votre prénom (exemple « julien »). Montrer en testant en ligne de commande que la base a été bien créée.
8. Toujours en ligne de commande, créez une table nommée « étudiants » avec 4 champs : un champ « id » entier qui sera clé primaire, un champ « nom », un champ « date_naissance » et un autre champs « classement ». Montrez que votre table existe bien. Insérez-y 3 enregistrements. Affichez le contenu de ces 3 enregistrements.
9. Supprimer un enregistrement de la table ayant un id égale à une valeur connue. Montrez que ça a bien marché.
10. Modifier la valeur du champ « date_naissance » dans l'enregistrement qui a le plus petite valeur du champs identifiant (i.e. « id »). La nouvelle valeur demandée est « 1er Janvier 1990 »
11. Refaire les manipulations précédentes faites en ligne de commande (depuis la création de la base de données) avec cette fois-ci un script PHP.
12. Commencez maintenant la réalisation de votre cahier de charge du projet