

Énoncé du travail pratique 2

Administration système

Philippe Pépos Petitclerc

Administration système

Lien FAQ

Un FAQ est disponible ici pour les questions qui reviennent plus souvent.

Introduction

L'objectif du travail pratique 2 est de vous permettre de mettre en application les outils et les concepts couverts par la deuxième partie du cours. Vous devrez adresser des problèmes conceptuels en appliquant les changements pratiques qui les résoud sur un système donné.

Il vous est donné une machine virtuelle que vous devrez administrer. Plus bas dans cet énoncé, vous trouverez une liste de problèmes à adresser (tâches). Vous devrez appliquer les changements requis pour régler les problèmes sur la machine virtuelle de façon à ce que les changements et les corrections soient permanents. En d'autres mots, la correction doit être effective même après un redémarrage de la machine virtuelle.

Vous devrez écrire un rapport dans lequel vous expliquerez vos décisions, votre cheminement et détaillerez vos actions.

Les livrables sont: la machine virtuelle dans son état final et le rapport qui aura été assemblé dans une archive de remise.

La machine virtuelle

Importation

Il vous est fourni une machine virtuelle au format `.ova`. C'est une machine virtuelle Ubuntu 64bits. Afin de l'importer, télécharger le `.ova` puis dans la fenêtre principale de VirtualBox, choisissez *File* puis *Import Appliance*. Choisissez ensuite le fichier `.ova` que vous venez de télécharger. Vous pouvez ensuite ajuster les allocations de ressources (mémoire et CPU) comme vous le souhaitez.

La machine virtuelle

Vidéo à venir

Exportation

Pour la remise, vous devrez exporter votre machine virtuelle finale. Pour ce faire, dans la fenêtre principale de VirtualBox, choisissez *File* puis *Export Appliance*.

1. Choisissez votre machine virtuelle à exporter
2. Choisissez le format *Open Virtualization Format 1.0*
3. Incluez seulement les adresses MAC du NAT.
4. Cochez d'inclure le manifest
5. Décochez d'inclure le fichier ISO.

Vidéo à venir

Accès

Vous pouvez vous authentifier sur le système avec le compte **user** et le mot de passe **user**. Cet utilisateur a accès **sudo**. Vous pourrez donc vous connecter en tant que **user** puis passer en **root** au besoin via **sudo**.

Rapport

Le rapport sert de journal dans lequel vous relaterez comment vous avez adressé chaque correction à faire ainsi que votre cheminement. Incluez:

- Vos actions: Les commandes ou séquences de commandes pertinentes.
- Vos raisonnements: Pourquoi faire ceci plutôt que cela? Pourquoi chercher ici? Pourquoi est-ce que je cherche quelque chose qui fait X? Comment pourrais-je adresser cette situation? etc.
- Vos démarches: Si vous faites des investigation pour trouver la cause d'un problème ou vous cherchez des processus en particulier, ou des fichiers en particulier.

En lisant le rapport, nous devrions être en mesure de comprendre les actions qui ont été prises, pourquoi, en quoi cela devrait adresser les problèmes et comment ces actions ont été mis en oeuvre (les commandes).

Le gabarit **rapport-tp2.md** est disponible ici.

Le format Markdown

Le gabarit **rapport-tp2.md** que vous allez remettre **doit respecter le format Markdown**, qui est un format texte à balisage léger. Le site officiel qui décrit ce format se trouve ici.

Faites bien attention de respecter le format actuel du gabarit et d'utiliser une syntaxe comparable à celle employée dans la mission 1, dont la solution complète est décrite, pour vous donner un exemple de ce qui est attendu. **Vous pouvez vérifier le rendu de votre fichier solutions-tp1.md** sur ce site par exemple.

Attention, le non-respect du format Markdown sera pénalisé, donc, prenez le temps de vérifier votre document sur le site mentionné précédemment.

Problèmes à corriger

Sommaire

Voici la liste potentielle des problèmes que vous pouvez corriger. Les détails de chaque cas se trouvent plus bas. Certaines de ces tâches sont obligatoires (O) et d'autres sont facultatives (F). La somme des points attribués pour les tâches obligatoires est de 100. Vous n'avez donc aucune obligation d'en faire plus. Par contre, vous pouvez aller chercher des points qui pourront compenser les points que vous pourriez perdre dans les tâches obligatoire en complétant des tâches facultatives.

Tâche	Catégorie	Points
Compte utilisateur	O	10
Utilisateur administrateur	F	5
Service problématique	O	10
PostgreSQL	O	15
Utilisateur postgres	O	10
Service postgres	O	5
Réseautique PostgreSQL	F	5
Réseautique Apache	O	15
Configuration Apache	O	5
Débogage Apache/PHP	O	20
Rédaction des journaux	O	10
Débogage PHP	F	5

Un script vous est fourni pour vérifier la correction des problèmes.

Détails des problèmes

Compte utilisateur Créez un compte utilisateur qui porte le nom de votre code permanent (AAAA99999999). Le répertoire par défaut de l'utilisateur doit être `/home/<CODE_PERM>/` et avoir les bonnes permissions pour un répertoire HOME. Le shell par défaut pour l'utilisateur devrait être `bash`.

Utilisateur administrateur La façon la plus simple de donner les droits `sudo` à un utilisateur est de l'ajouter dans le groupe `sudo`. Une autre façon, qui permet une gestion plus minutieuse des permissions `sudo` d'un utilisateur. La configuration des permissions `sudo` que l'on veut attribuer est dans la configuration `sudoers`. Vous pouvez vous référer au chapitres *SUDOERS FILE FORMAT* et *EXAMPLES* du manuel de `sudoers(5)`. Il est fortement recommandé d'utiliser l'application `visudo` lorsque vous modifier la configuration `sudoers`.

Donnez les droits `sudoers` à votre utilisateur code permanent. Nous voulons qu'il ait tous les droits `sudo` en fournissant son mot de passe.

Service problématique Un service en particulier occupe beaucoup de ressources sur le système. Trouvez le service coupable de l'utilisation excessive de ressources puis arrêtez-le. Assurez-vous également qu'il ne démarre plus automatiquement.

- `pstree`
- `systemctl status`

PostgreSQL Installer la version 14rc1 de *PostgreSQL* à partir de l'archive des sources disponible ici. Nous souhaitons que vous l'installiez dans `/opt/postgres` et que vous installiez les binaires dans `/usr/local/bin`. La documentation est disponible ici.

La compilation requiert le programme *Make* et un compilateur *C* ainsi que quelques bibliothèques de développement. Le paquet `build-essential` sur *Ubuntu* devrait installer ces deux premiers points. Les bibliothèques de développements requises sont spécifiées dans la documentation mais les noms de paquets exacts ne sont pas mentionnés. Pour vous aider, les bibliothèques de développement sous *Ubuntu* sont généralement trouvables dans les paquets qui ont un format comme `libXXX-dev` ou `XXX-dev`. De plus, les entêtes de développement pour l'interaction avec les démons *SystemD* (`sd-daemon.h`) est disponible dans le paquet `libsystemd-dev`.

Important: Passez également les options suivantes à la configuration (commande `./configure`): `--libdir=/usr/lib --includedir=/usr/include --with-systemd`

Les options `--libdir` et `--includedir` permettent aux exécutables de charger correctement. L'option `--with-systemd` ajoute les capacités de bien interagir avec `systemd` au démon `postgres`. C'est ce qui permettra de configurer l'unité `systemd` plus tard.

Utilisateur postgres Pour permettre au service *PostgreSQL* de s'exécuter en tant que son propre utilisateur, nous devons le créer. Créez donc un utilisateur `postgres`. **Cet utilisateur doit être un utilisateur système.** Les pages du manuel de `useradd`, `adduser` et `usermod` contiennent l'information nécessaire. Placez son répertoire par défaut (*HOME*) à `/var/lib/postgres/data`. Créez ce répertoire et mettez `postgres` comme utilisateur propriétaire.

Initialisez le répertoire de données de *Postgres* en créant le répertoire `/var/lib/postgres/data` puis mettez l'utilisateur `postgres` comme utilisateur propriétaire du répertoire et le groupe `postgres` comme groupe propriétaire.

À ce moment, initialisez le répertoire de données en devenant l'utilisateur `postgres` et en exécutant la commande suivante:

```
initdb -D /var/lib/postgres/data/
```

Si vous rencontrez des erreurs à ce moment, lisez bien les messages d'erreurs de la commande précédente. Normalement, ce qui ne va pas y est expliqué. Voici quand même quelques pistes de solutions:

- Les permissions sur le répertoire `/var/lib/postgres/` ou le répertoire `/var/lib/postgres/data/` ne permettent pas à l'utilisateur `postgres` d'y écrire.
- Le répertoire `/var/lib/postgres/data/` contient déjà des fichiers. Supprimez tout ce qui est contenu puis essayez.

Si tout c'est bien passé, la procédure d'initialisation de postgres vous affichera une commande qui permettra de lancer temporairement la base de données postgres. La commande sera similaire à celle-ci tirée de la documentation:

```
/usr/local/bin/pg_ctl -D /var/lib/postgres/data -l logfile start
```

Attention: La commande tente d'écrire les logs dans le répertoire courant (`-l logfile`). Assurez vous de lancer le serveur de quelque part où l'utilisateur `postgres` a les droits d'écritures. Elle doit également être lancée en tant que l'utilisateur `postgres`.

Si votre installation est un succès jusqu'à maintenant, les commandes suivantes s'exécuteront correctement et configureront votre base de données pour plus tard.

```
sudo -u postgres psql # Devenez l'utilisateur postgres si vous ne l'etes pas deja
postgres=# create database demo;
postgres=# create user demouser with encrypted password 'demopass';
postgres=# grant all privileges on database demo to demouser;
```

Attention aux ; après chaque commande postgresql.

Service postgres Créez une unité *SystemD* pour gérer le service *Postgres*. Vous pouvez vous inspirer de la définition suivante et ajuster les chemins et l'utilisateur.

```
[Unit]
```

```
Description=PostgreSQL database server
```

```
Documentation=man:postgres(1)
```

```
[Service]
```

```
Type=forking
```

```
User=pgsql
```

```
ExecStart=/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start
```

```
ExecReload=/bin/kill -HUP $MAINPID
```

```
KillMode=mixed
```

```
KillSignal=SIGINT
```

```
TimeoutSec=0
```

[Install]

WantedBy=multi-user.target

Le service doit s'appeler `postgresql.service` et être placé dans `/lib/systemd/system/`.
Démarez le service et activez le pour qu'il se lance au démarrage.

Réseautique PostgreSQL Par défaut, *Postgres* écoute sur une interface réseau locale nommée `localhost` (numériquement, `127.0.0.1`). Pour rendre le service disponible à d'autres systèmes sur le réseau, on voudrait plutôt faire écouter le service sur tous les interfaces. Pour ce faire, vous pouvez faire écouter le service sur `0.0.0.0`. Ce qui rendra le service disponible sur le réseau. N'oubliez pas de vous assurer que la configuration a été rechargée. La commande `ss` vous aidera à vérifier.

Réseautique Apache Nous souhaitons démarrer le service `apache2.service`. Malheureusement, un autre programme nuit au bon démarrage d'*Apache*. Sans modifier la configuration d'*Apache* (pour le moment), trouvez le programme qui nuit, arrêtez le et assurez-vous que l'effet est permanent. La commande `ss -lpn` peut encore une fois vous être utile. S'il vous faut arrêter ou désactiver des services, c'est OK.

Configuration Apache Une application web *PHP* est installée dans `/opt/app`. Ajustez la configuration d'*Apache* afin que le démon serve le répertoire de l'application correctement. Vérifiez également les permissions de l'application web installée dans `/opt/app`. L'utilisateur qui est utilisé par le serveur web doit être en mesure de lire et exécuter les fichiers de l'application (et doit être le seul à pouvoir à l'exception de root). Il ne doit pas pouvoir modifier les fichiers de l'application ni pouvoir modifier ses permissions sur ces fichiers. Il faut modifier la configuration d'hôte virtuelle et la configuration des permissions de répertoire.

Lors que la configuration est corrigée et effective, vous devriez être en mesure de visiter le lien suivant: `http://localhost/robots.txt` et voir le contenu du fichier `/opt/app/robots.txt`

Débogage Apache/PHP En visitant l'index de l'application web, vous remarquez que vous n'obtenez pas le résultat prévu. *Apache* vous sert le contenu textuel du code source de l'application plutôt que de l'exécuter.

Pour pouvoir exécuter une application *PHP* et servir de résultat, *Apache* doit avoir d'installé un module spécifique pour PHP. Le module est disponible dans le paquet `libapache-mod-php` et devra être activé.

Une fois activé, vous devriez voir le bon résultat en visitant `http://localhost/`

Expurgation des journaux Vous trouverez dans le répertoire `/root/redaction` un script qui affiche certaines lignes des journaux. Il vous y est demandé de masquer certaines informations des journaux afin que les informations

sensibles n'apparaissent plus lorsque l'on exécute le script. Complétez la ligne de commande et les expressions régulières nécessaires dans le script afin de respecter ce qui y est demandé.

Débogage PHP Il y a un bogue simple qui nuit au bon fonctionnement de l'application *PHP* dont vous venez de finaliser l'installation. Le message d'erreur se trouve dans les journaux d'*Apache*. Trouvez le message puis corrigez le bogue.

Script d'autocorrection

Un script est disponible pour valider vos corrections. Vous devrez également inclure la sortie de ce script dans votre rapport.

Pour valider vos corrections, vous pouvez exécuter la commande suivante **en tant que root**.

```
python3 check.pyc
```

Le script est dans le repo. Vous pouvez le télécharger ici

Modalités de remise

Vous devrez remettre votre machine virtuelle finale ainsi que l'archive de remise.

L'archive de remise peut-être préparée en lançant la commande suivante sur la machine virtuelle **en tant que root**:

```
python3 remise.pyc <CHEMIN VERS VOTRE RAPPORT>
```

Par exemple:

```
python3 remise.pyc /home/PEPP09090909/rapport-tp2.md
```

Le script est dans le repo. Vous pouvez le télécharger ici

Votre rapport doit inclure la sortie du script d'autocorrection.

Vous devrez donc exporter votre machine virtuelle comme expliqué précédemment puis la déposer sur votre compte *OneDrive* étudiant. Ensuite, vous devrez la partager avec votre enseignant et les correcteurs éventuels qu'il aura désigné, **mais personne d'autre**.

L'archive de remise devra être remise sur Moodle à l'endroit prévu à cet effet. Vous devez remettre **un seul fichier** nommé *exactement* **remise-tp2.tgz**.

La date de remise est fixée au 15 décembre 23h55.

Une pénalité de **2 points** par heure de retard sera appliquée, jusqu'à un maximum de 24 heures. **Après ce délai, AUCUN TP NE SERA ACCEPTÉ.**

Aucune remise par courriel ne sera acceptée, peu importe le motif. Plus précisément, **si vous envoyez votre travail par courriel, il sera considéré comme non remis**. Il est donc de votre responsabilité de vous assurer d'être

capable de faire une remise par l'intermédiaire de Moodle quelques jours avant la remise.

FAQ

Est-ce que je peux changer la configuration Réseau?

Vous pouvez configurer le réseau comme vous le voulez.

- Avec un NAT, votre VM aura accès à l'internet mais ne sera pas accessible sur le réseau. C'est bien si vous ne prévoyez pas connecter à votre VM par SSH.
- Avec un Host-Only Adapter, votre VM sera joignable de votre hôte principal. Par contre, elle n'aura pas d'accès internet direct. Pas conseillé.
- Avec un Bridged Adapter, votre VM sera disponible sur le même réseau que votre hôte. Votre VM aura donc une adresse IP fournie par votre routeur et aura donc accès internet ET sera joignable par SSH. Par contre, vous devrez créer un *Bridge*: de la fenêtre principale de VirtualBox, *File* -> *Host Network Manager* -> *Create*

J'ai une erreur de configuration en important la VM dans la section réseau.

Placez votre souris sur l'icône de l'erreur et vous aurez plus de détails.