

# Introduction à la ligne de commande et à l'écosystème Unix/Linux

## 0. Création de la machine

- Installer Virtualbox, puis créer une nouvelle machine virtuelle
  - choisissez comme type Linux / Other-Linux (64 bit)
  - 2048 Mo de RAM devraient suffir
  - au moment de spécifier le disque dur virtuel, utiliser l'image de disque provenant de OSboxes.org

## 1. Démarrer et se logger

- Démarrer la machine et *observer* son démarrage
- L'utilisateur par défaut est `osboxes` et le mot de passe `osboxes.org`. Attention, si le clavier est en qwerty mais que vous tapez depuis un clavier azerty, cela donne `osboxes:org`
- Une fois loggué, confirmez que la disposition du clavier est Français/Azerty. (Si nécessaire, la changer en allant dans le "Menu démarrer" > Keyboards, puis '+' pour ajouter la disposition de clavier. Enfin, supprimez la disposition Qwerty avec le '-').
- Pour que l'écran de la machine virtuelle (VM) s'adapte automatiquement et joliment à la taille de l'écran, il vous faut utiliser le menu "Périphériques" (tout en haut de l'écran), puis, en bas du menu, "Insérer le CD des add-ons invités". Ensuite, accepter d'exécuter le CD dans Linux Mint. Un terminal s'ouvre automatiquement pour installer des logiciels. Une fois terminé, redémarrer la machine, et l'écran devrait s'adapter automatiquement à la taille de la fenêtre.

## 2. Premier contact avec la ligne de commande commandes

- Changer le mot de passe en tapant `passwd` puis *Entrée* et suivre les instructions
- Taper `pwd` puis *Entrée* et observer
- Taper `ls` puis *Entrée* et observer
- Taper `cd /var` puis *Entrée* et observer
- Taper `pwd` puis *Entrée* et observer
- Taper `ls` puis *Entrée* et observer
- Taper `ls -l` puis *Entrée* et observer
- Taper `echo 'Je suis dans la matrice'` puis *Entrée* et observer

## 3. La ligne de commande

- 3.1 - Rendez-vous dans `/usr/bin` et listez le contenu du dossier

- 3.2 - Y'a-t-il des fichiers cachés dans votre répertoire personnel ?
- 3.3 - Quand a été modifié le fichier `/etc/shadow` ?
- 3.4 - Identifiez à quoi sert l'option `-h` de la commande `ls` via son `man` .
- 3.5 - Cherchez une option de `ls` qui permet de trier les fichiers par date de modification
- 3.6 - Identifiez ce que fait la commande `sleep` via son `man` .
- 3.7 - Lancer `sleep 30` et arrêter l'exécution de la commande avant qu'elle ne se termine.
- 3.8 - Pour vous entraîner à utiliser [Tab] et ↑, tentez le plus rapidement possible et en utilisant le moins de touches possible de lister successivement le contenu des dossiers `/usr` , `/usr/share` , `/usr/share/man` et `/usr/share/man/man1` .
- 3.9 - Se renseigner sur ce que font `date` et `cal`
- 3.10 - Afficher le calendrier pour l'année 2022, puis juste le mois de Février 2022
- 3.11 - Se renseigner sur ce que fait la commande `free` , et interpréter la sortie de `free -h`
- 3.12 - Se renseigner sur ce que fait la commande `ping` et interpréter la sortie de `ping 8.8.8.8`

## 4. Le système de fichier

- 4.1 - En utilisant `mkdir` et `touch` , créez dans votre répertoire personnel l'arborescence suivante :

```
~/
├── Documents/
│   ├── formation_linux/
│   │   ├── slides_1_les_bases.html
│   │   └── exo_1_les_bases.pdf
│   └── mon_pokedex/
│       ├── index.html
│       ├── all_pokemons.txt
│       ├── mon_equipe_de_pokemons.csv
│       └── assets/
│           ├── css/
│           │   └── pokedex.css
│           ├── fonts/
│           │   └── pokefont.ttf
│           └── img/
│               ├── logo.png
│               ├── pikachu.jpg
│               └── carapuce.jpg
```

- 4.2 - Téléchargez la liste de tous les pokémons connus ( `all_pokemons.txt` ) depuis le serveur du formateur à l'aide de `wget` .
- 4.3 - À l'aide de `nano` , remplissez `mon_pokedex/mon_equipe_de_pokemons.csv` avec quelque chose comme:

Pokemon;Niveau  
bulbizarre;17  
rattata;8  
roucoups;15

Vérifiez que le contenu a bien été pris en compte en l'affichant avec `cat` .

- **4.4** - Aller dans `~/Documents/mon_pokedex/assets/` puis, **en utilisant uniquement des chemins relatifs** et en vous aidant de la touche [Tab], déplacez-vous successivement vers :
  - `~/Documents/mon_pokedex/assets/img`
  - `~/Documents/formation_linux`
  - `~/.local/` (ou `~/.config/` si `~/.local/` n'existe pas)
  - `~/Documents/mon_pokedex/assets/fonts`
  - `/usr/share/doc/`
  - `~/`
- **4.5** - Créez un fichier `dracaufeu.jpg` dans `~/Documents/formation_linux` ... Vous réalisez ensuite que vous auriez voulu mettre ce fichier dans `~/Documents/mon_pokedex/assets/img` ! Utilisez alors la commande `mv` pour déplacer le fichier vers le bon dossier.
- **4.6** - Renommez le dossier `mon_pokedex/` en `ma_collection_de_pokemons/`
- **4.7** - Supprimez le fichier `carapuce.jpg` dans `~/Documents/mon_pokedex/assets/img` **en restant là où vous êtes actuellement, i.e. sans utiliser `cd`**
- **4.8** - Créez un dossier `~/sauvegardes` et dedans, créer un dossier `collection_bkp` qui sera une copie récursive de `~/Documents/ma_collection_de_pokemons`
- **4.9** - Supprimez tout le dossier `~/sauvegardes` récursivement
- **4.10** - Depuis là où vous êtes (i.e. sans utiliser `cd` !):
  - affichez le contenu de `/etc/os-release` : devinez-vous à quoi correspondent ces informations ?
  - affichez le contenu de `/etc/hostname` : à quoi correspond cette information ?
  - affichez le contenu de `/etc/timezone` : à quoi correspond cette information ?
  - affichez le contenu de `/etc/default/locale` : à quoi correspond cette information ?
- **4.11** - Regardez le contenu de `/etc/nanorc` :
  - à quoi correspond ce fichier ?
  - en utilisant `less` , cherchez toutes les occurrences du mot `set` .
  - même chose mais cette fois en ouvrant le fichier avec `nano` (il existe un raccourci clavier pour chercher un mot dans `nano` )

- 4.12 - Utilisez une commande pour compter le nombre de ligne du fichier `/etc/nanorc`
- 4.13 - Copiez le fichier `/etc/nanorc` dans `~/.nanorc` . Éditez ensuite cette copie pour décommenter la ligne `# set linenumbers` (c'est à dire enlever le `#` devant la ligne pour activer l'option `linenumbers` ). Qu'avons-nous fait avec cette manipulation ? Pourquoi avoir copié le fichier dans notre répertoire personnel pour faire cela ?
- 4.14 - (Avancé) Créez (puis supprimez) un fichier qui s'appelle littéralement `*.py`
- 4.15 - (Avancé) Créez (puis supprimez) un fichier qui s'appelle littéralement `-f`

## 5. Utilisateurs, groupes et shells

- 5.1 - Ouvrir un premier terminal en tant que `padawan`
- 5.2 - Ouvrir un deuxième terminal. Dedans, ouvrir un sous-shell en `root` à l'aide des commandes `sudo` et/ou `su` .
- 5.3 - Dans votre deuxième terminal (en `root` )
  - créez un utilisateur `r2d2`
  - définissez un mot de passe pour l'utilisateur `r2d2` à l'aide de la commande `passwd`
  - créez un groupe `droid`
  - ajoutez `r2d2` au groupe `droid`
  - constatez que les infos de `r2d2` et du groupe `droid` sont bien dans `/etc/passwd` , `/etc/shadow` et `/etc/group`
- 5.4 - Ouvrir un troisième terminal. Dedans, ouvrir un sous-shell en tant que `r2d2` à l'aide des commandes `sudo` et/ou `su` .
  - l'invite de commande obtenue est différente de celle de `r2d2` et `root` . Pourquoi ? (On pourra comparer le résultat de la commande `echo $SHELL` )
  - comment peut-on procéder pour changer le shell par défaut de `r2d2` ? (Indice: regarder `/etc/passwd` , ou bien la commande `chsh` )
  - regardez le résultat des commandes `whoami` , `id` et `groups` et comparez à ce que vous obtenez pour ces commandes dans le premier terminal (en tant que `padawan` )
- 5.5 - Inspectez le contenu de `/etc/sudoers` :
  - en lisant les commentaires du fichier, chercher comment faire pour donner le droit à `r2d2` d'utiliser `sudo`
  - (après avoir fait la manip, n'oubliez pas de relancer le terminal/shell dans lequel vous êtes pour propager le changement!)
  - depuis un shell en tant que `r2d2` , validez que vous êtes en mesure de faire des commandes avec `sudo` (Par exemple: `sudo ls -la /root/` ).
- 5.6 - Constatez que les commandes exécutées avec `sudo` sont logguées dans le fichier `/var/log/auth.log` (on pourra utiliser `tail` pour afficher seulement les dernières lignes du fichier)

## 6. Permissions

- 6.1 - Créez un fichier `xwing.conf` que seul vous et votre groupe pouvez lire

- 6.2 - Créez un fichier `private` et supprimer toutes les permissions dessus
- 6.3 - Ajoutez successivement à `private` le droit de lecture au propriétaire, le droit d'écriture au groupe et au propriétaire, et les droits d'exécution pour tout le monde.
- 6.4 - Resupprimez toutes les permissions de `private`
- 6.5 - Remettez les mêmes permissions qu'avant mais avec une seule commande en utilisant la notation octale
- 6.6 - Modifier les permissions de votre répertoire personnel pour que seul vous ayez le droit d'écriture et de traverse (x) dessus
- 6.7 - Interdisez à tous les "autres" utilisateurs de fouiller et modifier les fichier dans `~/documents` , avec une seule commande qui aura un effet récursif
- 6.8 - Créez un répertoire personnel pour `r2d2` . Définir `r2d2` comme propriétaire de son dossier personnel + s'assurer que les permissions lui permettent (à lui et à lui seul) de lire, écrire et entrer dans son repertoire.
- 6.9 - Créez un fichier `droid.conf` dans son dossier personnel, le définir comme propriétaire, et définir le groupe comme 'droid'.
- 6.10 - Créez des fichier `beep.wav` , `boop.wav` et `blow.wav` que seul `r2d2` peut exécuter.
- 6.11 - Êtes-vous capable de créer un dossier qui contient des fichiers qu'il est possible de lire, mais pas de lister ?
- 6.12 - En tant qu'utilisateur `padawan` , arrivez-vous à donner un de vos fichier à `r2d2` ?
- 6.13 - (Avancé) Utilisez `setfacl` pour autoriser le groupe `droid` à lister et rentrer dans votre home. Confirmez l'effet attendu, d'une part avec `ls -l` et `getfacl` , et d'autre part depuis un shell en étant connecté en tant que `r2d2`
- 6.14 - (Avancé) Même chose, mais cette fois-ci donnez le droit de list et rentrer dans `/home/r2d2` à l'utilisateur `padawan` .

## 7. Processus

- 7.1 - En utilisant `ps -ef --forest` (ou `top` ) pour:
  - identifiez le processus qui fait tourner votre environnement graphique (généralement il s'appelle cinnamon)
  - identifiez l'un de vos terminaux et son PID
  - identifiez le processus qui consomme actuellement le plus de CPU
  - identifiez le processus qui consomme actuellement le plus de RAM
  - trouvez un processus qui ne tourne ni en tant que `root` , ni en tant que `padawan`
- 7.2 - Récupérez le programme `fibonacci_forever.sh` auprès du formateur, puis lancez `bash fibonacci_forever.sh` dans un terminal.
- 7.3 - Mettez ce processus en arrière-plan. Vérifiez avec `jobs` qu'il continue de s'exécuter.
- 7.4 - Depuis un autre shell, identifiez le PID de ce processus à l'aide de `ps -ef --forest` , et servez-vous de ce PID pour tuer le processus.
- 7.5 - Relancez le processus directement en arrière plan cette fois (avec `&` )
- 7.6 - Identifiez cette fois le shell qui a lancé ce processus. Qu'arrive-t-il si vous tuez ?
- 7.7 - Lancez une session `screen` , puis dedans, lancez de nouveau le programme `fibonacci_forever.sh` . Détachez la session, puis ré-attachez-la dans un

autre terminal.

- 7.8 - Dans une autre console, identifiez via `ps` le PID de la session screen et tentez de tuer ce processus.
- 7.9 - (Avancé) Identifiez le PID de votre shell, puis regardez la sortie de `ls -l /proc/<PID>/cwd` (en remplaçant `<PID>` par le PID de votre shell). À quoi cela correspond-t-il ?
- 7.10 - (Avancé) Test de l'impact de la priorité des processus sur la rapidité d'exécution
  - Lancer la commande `openssl speed -multi 4` - puis refaite le test
  - Tout en laissant `openssl speed -multi 4` s'exécuter, lancer la commande `ls /bin/` avec la priorité la plus faible possible. Que se passe-t-il ?
  - Réduisez drastiquement "à chaud" la priorité de la commande `openssl speed -multi 4` en train de s'exécuter. Si vous relancer `ls /bin/` toujours avec la priorité la plus basse, comment la situation évolue-t-elle ?
  - Comment pouvez-vous tuer d'un seul coup tous les processus `openssl` ?

## 8. Personnaliser son environnement

- 8.1 - Personnaliser l'apparence de votre invite de commande (syntaxe, couleurs) en modifiant la variable PS1 : par exemple, afficher le nom de la machine en jaune.
- 8.2 - Ajouter la personnalisation du PS1 à votre `.bashrc` et propagez ces changements sur vos shells ouverts.
- 8.3 - Ajouter aussi un message de bienvenue comme "May the force be with you!" qui s'affichera à chaque ouverture d'un shell (rappel : `echo` peut être utilisé pour afficher un tel message).
- 8.4 - Changer le `.bashrc` de root pour que son invite de commande soit en rouge !
- 8.5 - S'assurer que vous disposez de l'alias `ll` (pour `ls -l`), et que `--color=auto` est activé implicitement lorsque vous utilisez `ls`.
- 8.6 - Créer un alias `testinternet` qui exécute la commande `ping` sur l'IP 8.8.8.8 avec l'option `-q` et l'option `-c 1`
- 8.7 - Créer un alias `r2d2` qui permet d'ouvrir un shell en tant que `r2d2` avec `sudo` et `su`.
- 8.8 - En utilisant `echo`, comment faire pour faire en sorte que la commande `ls` retourne systématiquement 'Bof, G pas envie' au lieu de son comportement normal ?
- 8.9 - (Avancé) Se renseigner sur `LS_COLORS` et personnaliser cette variable.

## 9 - redirections et assemblages

- 9.1 - Créer un fichier `hello.txt` qui contient "Hello!", à l'aide la commande `echo` et d'une redirection.
- 9.2 - À l'aide d'une deuxième commande `echo` et d'une autre redirection, ajoutez à la suite (sur une nouvelle ligne) le mot `World!` dans `hello.txt`.
- 9.3 - Stockez la sortie de `ls /usr/bin` dans un fichier, et observez ce fichier avec la commande `less`. Observez-vous une différence entre le format de ce fichier, et la sortie

de `ls /usr/bin` lorsqu'elle est affichée directement dans le terminal ?

- **9.4** - Lancez le script `bash fibonacci_forever.sh` en redirigeant sa sortie vers un fichier. La commande tournera jusqu'à ce qu'elle soit interrompue : attendez donc quelques secondes puis appuyez sur Ctrl+C pour l'interrompre. Inspectez le contenu du fichier ensuite.
- **9.4** - `bc` est un utilitaire permettant de faire de petits calculs. Testez `bc` en mode interactif pour faire quelques additions (Ctrl+D pour quitter). Mettez maintenant une suite de calcul dans un fichier (par exemple, `2+2` , `6*7` , `10/3` ), que vous injecterez directement dans `bc` .
- **9.5** - Même question que précédemment, mais en injectant directement une chaîne contenant un calcul dans `bc` (sans passer par un fichier).
- **9.6** - Écrivez **une seule ligne de commande** (qui comportera plusieurs sous-commandes séparées par `;` , mais sans utiliser `cd` !) qui :
  - créer le dossier `~/formation_linux/calculs`
  - créer un fichier `~/formation_linux/calculs/formule` contenant `6*7`
  - effectuer le calcul contenu dans le fichier `formule` , et stocke le résultat dans `~/formation_linux/calculs/reponse`
- **9.7** - `curl` est un utilitaire qui permet de télécharger et le code source d'une page (par exemple, `fr.wikipedia.org` ) dans la console.
  - Comment pouvez-vous faire pour sauvegarder cette page dans un fichier ?
  - Retentez l'expérience avec une adresse qui n'existe pas, et modifiez votre commande pour supprimer les erreurs mais afficher tout de même "Ça n'a pas marché" dans le cas où vous tentez de télécharger une page qui n'existe pas...
- **9.8** - (Avancé) Créez un fichier `/tmp/chat` dans lequel `padawan` et `r2d2` peuvent tous les deux écrire. Renseignez-vous sur l'option `-f` de la commande `tail` puis :
  - Lancez `tail -f /tmp/chat` en tâche de fond dans deux terminaux (l'un de `padawan` , l'autre de `r2d2` );
  - À l'aide de redirections, envoyez des messages dans le fichier `/tmp/chat` et observez les deux terminaux ;
  - Améliorez le système en créant un alias `say` qui affiche un message préfixé de votre nom d'utilisateur (ex: `[r2d2] beep boop` ).

## 10 - pipes et boîte à outils

- **10.1** - Si ce n'est pas déjà le cas, ajoutez un alias pour activer automatiquement `--color=auto` chaque fois que la commande `grep` est utilisée. Essayez quelques manipulations avec `grep` pour confirmer que les occurrences trouvées sont bien mises en valeur.
- **10.2** - Lister les lignes de `/etc/passwd` qui correspondent aux utilisateurs ayant `/bin/bash` comme shell
- **10.3** - Même chose, mais cette fois en affichant uniquement le nom des utilisateurs
- **10.4** - Lister les utilisateurs (uniquement leur nom !) qui ont comme shell `nologin`
- **10.5** - Lister les utilisateurs (uniquement leur nom) qui ont un mot de passe non vide (rappel : historiquement les passwords se trouvaient dans `/etc/passwd` , mais ce n'est



plus le cas de nos jours !)

- 10.6 - Écrivez un alias `estcequecestleweekend` qui vérifie si on est Samedi ou Dimanche en utilisant `date`
- 10.7 - Sachant que pour `grep`, `^` et `$` désignent un début et une fin de ligne, pouvez-vous afficher le contenu de `/etc/login.defs`
  - sans les commentaires (ce sont les lignes commençant par `#`)
  - puis sans les commentaires ni les lignes vides ? (Indice : une ligne vide est une ligne qui se commence puis se termine tout de suite)
- 10.8 - Modifiez votre alias `testinternet` pour qu'il affiche "Connecté!" ou "Pas de connexion!" suivant si le ping fonctionne (en masquant la sortie initiale du ping).
- 10.9 - À l'aide des pages de man de `grep`, trouvez un moyen de lister toutes les occurrences du mot `daemon` dans tous les fichiers à l'intérieur de `/etc/` (recursivement)
- 10.10 - À l'aide de `ps`, `sort` et `uniq` générer un bilan du nombre de processus actuellement en cours par utilisateur
- 10.11 - À l'aide de `sort` et `uniq`, analysez le fichier `loginattempts.log` (demander au formateur comment l'obtenir), et produisez un résumé du nombre de tentative de connections par ip

## 11. - Se connecter en SSH

- 11.1 - Pingez votre serveur, connectez-vous dessus en root (si possible en vérifiant la fingerprint du serveur) et **changer le mot de passe** ! (Choisir un mot de passe un minimum robuste : il sera mis à l'épreuve !!!). Dans une autre console, constater qu'il y a maintenant une entrée correspondant à votre serveur dans `~/.ssh/known_hosts`.
- 11.2 - **Sur votre serveur**, familiarisez-vous avec le système :
  - de quelle distribution s'agit-il ? ( `lsb_release -a` ou regarder `/etc/os-release` )
  - quelle est la configuration en terme de CPU, de RAM, et d'espace disque ? ( `cat /proc/cpuinfo`, `free -h` et `df -h` )
  - quelle est son adresse IP locale et globale ?
- 11.3 - **Sur votre serveur** : donnez un nom à votre machine avec `hostnamectl set-hostname <un_nom>`. (Attention, ce nom est purement cosmétique et interne à la machine. Il ne s'agit pas d'un vrai nom de domaine résolvable et accessible par n'importe qui sur internet, à la différence de celui qui sera configuré à la question 5.8)
- 11.4 - **Sur votre serveur** : créez un utilisateur destiné à être utilisé plutôt que de se connecter en root.
  - Créez-lui un répertoire personnel et donnez-lui les permissions dessus.
  - Définissez-lui un mot de passe.
  - Ajoutez-le au groupe `ssh`.
  - Assurez-vous qu'il a le droit d'utiliser `sudo`.
- 11.5 - **Depuis votre machine de bureau (VM Ubuntu)** : ajoutons maintenant une vraie clef SSH :
  - générez une clef SSH pour votre utilisateur avec `ssh-keygen -t rsa -b 4096 -C "un_commentaire"` ;



- identifiez le fichier correspondant à la clef publique créé (généralement `~/.ssh/un_nom.pub`);
- utilisez `ssh-copy-id -i clef_publique user@machine` pour copier et activer la clef sur votre serveur;
- (notez que sur le serveur, il y a maintenant une ligne dans `~/.ssh/authorized_keys`);
- tentez de vous connecter à votre utilisateur en utilisant désormais la clef (`ssh -i clef_privee user@machine`);
- 11.6 - Depuis votre machine de bureau (VM Ubuntu), configurez `~/.ssh/config` avec ce modèle de fichier. Vous devriez ensuite être en mesure de pouvoir vous connecter à votre machine simplement en tapant `ssh nom_de_votre_machine`

```
Host nom_de_votre_machine
  User votre_utilisateur
  Hostname ip_de_votre_machine
  IdentityFile chemin_vers_clef_privee
```

## 12 - Installer et configurer un serveur web

NB: les opérations suivantes sont à effectuer **sur votre serveur** au travers de SSH depuis votre machine Ubuntu

- 12.1 - Installer `nginx` puis vérifier que le service tourne bien avec `systemctl status nginx`. On pourra aussi utiliser `ps -ef --forest` pour constater qu'un processus nginx tourne bien, ainsi que `netstat -tulpn` pour constater qu'il écoute bien sur le port 80.
- 12.2 - Tester d'accéder à votre serveur depuis un navigateur web en tapant `http://ip.de.votre.serveur` (vérifiez bien que vous utilisez `http` et non `https` qui n'est pas configuré sur le serveur). Comparez la page visible dans votre navigateur au fichier qui se trouve dans `/var/www/html/`.
- 12.3 - Nous voudrions maintenant servir notre propre contenu web plutôt que l'exemple de nginx. Créer un dossier `monsite` dans `/var/www/` et à l'intérieur, créer un fichier `index.html` qui contient par exemple :

```
<html>
<h3>Hello world !</h3>

</html>
```

Ensuite, modifier le fichier `/etc/nginx/sites-enabled/default` : trouvez l'instruction à modifier pour servir le dossier `/var/www/monsite/` plutôt que `/var/www/html/`. Vérifiez ensuite que vos changements ne causent pas de problèmes grâce à `nginx -t`, puis si tout est ok, recharger le service avec `systemctl reload nginx`. Arrivez-vous maintenant à accéder à votre page web ?

- 12.4 - Normalement, votre page affiche bien 'Hello world' mais nous n'avons pas réellement mis d'image de chaton. Nous pouvons en télécharger une aléatoire grâce

à `wget https://thecatapi.com/api/images/get` . Assurez vous de renommer le fichier en `chatonmignon.jpg` et de le mettre dans `/var/www/monsite/` .

- 12.5 - Rendez-vous dans `/var/log/nginx/` et lancer une surveillance du log `access.log` à l'aide de `tail -f access.log` . Depuis votre navigateur, rechargez plusieurs fois la page de votre site et étudiez les lignes qui apparaissent dans votre console.
- 12.6 - À l'aide de `ps -ef` , identifiez le(s) processus qui font tourner nginx. Quels seraient alors le propriétaire et les permissions adéquate à appliquer aux dossier et fichiers `/var/www/monsite` ?
- 12.7 - Que se passe-t-il si vous arrêter nginx avec `systemctl stop nginx` ?

## 13. Déploiement (simplifié) d'une application PHP/Mysql : Nextcloud

Dans cette partie, on se propose de déployer une application basée sur PHP / Mysql, ce qui est un exemple classique d'application "dynamique" (c.f. architecture LAMP).

Une telle installation implique typiquement les étapes suivantes :

- téléchargement de l'application et extraction dans le bon dossier
- installation de dépendances
- création de la base de données
- configuration du serveur web
- configuration de l'application
- test et finalisation de l'installation

Les instructions suivantes ne viennent pas de `/dev/urandom` : elles ont été récupérées depuis le site officiel de Nextcloud (et aussi du script d'installation de l'app YunoHost !).

- 13.1 - **Téléchargez l'archive** de la dernière version de Nextcloud (c.f. lien fourni dans les documents du formateur). Décompresser l'archive à l'aide de `tar -xvf <archive>` et mettre son contenu dans `/var/www/nextcloud` .
- 13.2 - **Installez les dépendances** de Nextcloud (c.f. liste fournie dans les documents du formateur) à l'aide de la commande `sudo apt install <paquet1> <paquet2> <paquet3> ...` . Vérifier qu'il y a bien des services `nginx` , `php7.4-fpm` et `mysql` (ou `mariadb` ) à l'aide de `ps` et de `systemctl status nginx` (idem avec `php7.4-fpm` et `mysql` ).
- 13.3 - **Base de données** : Créez un utilisateur `nextcloud` (NB: au sens de mysql, PAS au sens "Unix"/`useradd` !) et une base de donnée portant le même nom. Pour ceci, il faut ouvrir une console mysql et utiliser les incantations suivantes (éventuellement, remplacez `password` par un vrai mot de passe) (aussi : n'oubliez pas les `;` !)

```
$ mysql -u root
MariaDB [(none)]> CREATE USER 'nextcloud'@'localhost'
IDENTIFIED BY 'password';
```

```
MariaDB [(none)]> CREATE DATABASE IF NOT EXISTS nextcloud;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nextcloud.*
                    TO 'nextcloud'@'localhost'
                    IDENTIFIED BY 'password';
MariaDB [(none)]> FLUSH privileges;
MariaDB [(none)]> quit
```

- **13.4 - Configurons Nextcloud (1/2)** pour utiliser la base de donnée qui vient d'être créée. Pour cela, rendez-vous dans `/var/www/nextcloud`. Assurez-vous qu'il existe un fichier `occ` dans ce dossier. Lancez ensuite la commande suivante (êtes-vous capable de comprendre le rôle de ses différents morceaux ?). Il vous faudra peut-être remplacer `password` par le mot de passe précédemment choisi.

```
$ php occ maintenance:install \
  --database "mysql" --database-name "nextcloud" \
  --database-user "nextcloud" --database-pass "password" \
  --admin-user "admin" --admin-pass "password"
```

- **13.5 - Configurons Nextcloud (2/2)** : il nous faut aussi définir le domaine derrière lequel Nextcloud est hébergé : éditez le fichier `config/config.php` de Nextcloud, et ajoutez l'IP de votre machine dans les "trusted domains". Vous pouvez connaître l'IP de votre machine à l'aide de : `curl ifconfig.me`. Ajoutez également le paramètre `overwriteprotocol` avec la valeur `http`. Cela devrait ressembler à quelque chose comme :

```
'trusted_domains' =>
[
  '12.34.567.89'
],
```

```
'overwriteprotocol' => 'http',
```

- **13.6 - Configurons nginx (le serveur web)** pour servir l'application Nextcloud. Pour cela, récupérez et étudiez le modèle de configuration (toujours dans les documents du formateur). Copiez-le à un endroit approprié à l'intérieur de `/etc/nginx/sites-enabled/default`. Dans le fichier, notez l'existence d'une ligne mentionnant `/var/run/php/php7.4-fpm.sock`. À votre avis, à quoi sert ce fichier et cette ligne ?
- **13.7 - Testez** que la configuration nginx semble valide avec `nginx -t`, rechargez la configuration nginx avec `systemctl restart nginx`, vérifiez que le service s'est bien relancé avec `systemctl status nginx`, puis tentez d'accéder à votre application via un navigateur web depuis votre machine de Windows en tapant l'IP dans un navigateur. Si elle ne fonctionne pas correctement (c'est probable !), **investiguez les logs d'erreur de nginx** dans `/var/log/nginx/`. Comparez les messages aux permissions de `/var/www/nextcloud`, et à l'utilisateur avec lequel tournent les processus `php-`

**fpm . Comment faut-il modifier les permissions pour que l'application fonctionne correctement ?**

- 13.8 - Une fois le problème résolu, tester que l'application fonctionne correctement et découvrir Nextcloud (téléversez des fichiers, créez des dossier, etc...). (Il est même possible d'installer une application Nextcloud sur votre smartphone pour synchroniser les fichiers !)