

Justification des choix d'architecture

- **Type et nombre de machines**

Serveur Web :

- Héberger l'application web, gérer les requêtes utilisateurs.
- Assure l'affichage des pages et la navigation entre celles-ci.
- Nombre : 1.

Serveur de base de données :

- De type Oracle SQL
- Stocker toutes les informations relatives aux événements, utilisateurs, messages et autres.
- Un deuxième serveur pourra être mis en place pour sauvegarde.
- Nombre : 1 et potentiellement 2 si l'application grandit.

Serveur DHCP :

- Sert à distribuer automatiquement les adresses IP aux dispositifs connectés au réseau.
- Nombre : 1.

Routeur :

- Assure la connexion entre les serveurs et l'extérieur.
- Nombre : 2.

Switch :

- Connecte les serveurs entre eux pour un réseau local.
- Nombre : 1.

Gateway :

- Pour communiquer avec l'extérieur.
- Nombre : 1.

- **Plan d'adressage**

Pour le plan d'adressage, c'est le serveur DHCP qui s'occupera de distribuer les adresses IP dynamiquement, mais il faut quand même lui renseigner les plages d'adresses

réservées pour chaque machine.

Pour les autres machines que le serveur DHCP ne prend pas en charge, nous avons configuré les adresses IP à la main. Nous avons également séparé les machines en différents sous-réseaux LAN pour plus de sécurité.

LAN 1 : 192.168.0.0/24

Serveur DHCP : 192.168.0.13/24

Routeur 1 : 192.168.0.13/24

Le reste des adresses disponibles sera pour les utilisateurs dynamiques

LAN 2 : 192.168.1.0/24

Serveur DHCP : 192.168.1.13/24

Serveur WEB : 192.168.1.11/24

Serveur BD : 192.168.1.12/24

Routeur 1 : 192.168.1.14/24

Routeur 2 : 192.168.1.15/24

LAN 3 : 192.168.255.0/24

Routeur 2 : 192.168.255.15/24

Gateway : 192.168.255.2

• Choix des services déployés

Serveur BD :

Avec la distribution `wheezy-serv-mysql`, le service MySQL est déjà installé sur le serveur, nous avons donc choisi de garder cela car c'est plus simple à l'installation et nous savons déjà utiliser MySQL.

Serveur WEB :

Pour que le serveur web fonctionne correctement, il y a deux services à installer, `apache2` et `php5` en utilisant la commande `aptitude install`.

```
aptitude install apache2
aptitude install php5
```

Puis le module `mysql` de `php` pour pouvoir se connecter à la BD

```
aptitude install php5-mysql
```

• Configuration des services

Serveur BD :

Le service MySQL étant déjà installer, il y a déjà une configuration par défaut, nous avons quand même modifier le fichier de configuration

`/etc/mysql/my.conf` pour changer la ligne `bind-address = 127.0.0.1`, qui est la ligne par défaut, en `bind-address = 192.168.1.12` qui est l'adresse IP du serveur de BD.

Ensuite il n'y a plus qu'à démarrer les services `/etc/init.d/mysql start`. Cette à été mise dans le fichier `/etc/rc.local`.

Serveur Web :

A l'installation, les services apache2 et php5 sont déjà préconfigurés. Il faut simplement démarrer les services `/etc/init.d/apache2 start`. Cette et également à mettre dans le fichier `/etc/rc.local`.

Sur cette machine il y a également le service SSH qui été déjà installer par défaut, nous allons cependant modifier le fichier de configuration `/etc/ssh/sshd_config` pour changer la ligne `PermitRootLogin yes` à `no`.

Et ensuite redémarrer le service avec la commande qui et aussi à mettre dans le fichier `/etc/rc.local`.

`/etc/init.d/ssh start`.

Enfin, nous avons créer un utilisateur autre que root pour pouvoir se connecter avec un utilisateur qui n'est pas administrateur. `useradd -d /home/groupe7 -m groupe7`, cette commande créer l'utilisateur groupe7 avec le répertoire par défaut `/home/groupe7`.

On lui définis ensuite un mot de passe, ici, `groupe7`.

`passwd groupe7`.

• Réalisation du POC

Pour la réalisation du POC nous eu besoin de créer une mini base données pour pouvoir tester la communication avec le serveur BD et le serveur Web.

Nous avons donc créer la base appWeb. avec les tables Utilisateur, Playlist, et TitreMusique, que nous avons ensuite peuplées

Pour montrer les tables au prof :

Sur le serveur BD :

`mysql -u root -p` : se connecter à MySQL en tant que root

`show DATABASES;` : afficher toutes les bases de données

`use appWeb;` : entrer dans la base appWeb

`SELECT * FROM Utilisateur;` : afficher le contenus de la table Utilisateur.

`SELECT * FROM Playlist;` : afficher le contenu de la table Playlist.

`SELECT * FROM TitreMusique;` : afficher le contenu de la table TitreMusique.

Nous avons ensuite créé deux fichiers php pour le serveur web. Voir le contenu :

`nano /etc/www/index.php`

`nano /etc/www/playlistUtilisateur.php`

dans le premier fichier nous avons fait un petit formulaire dans lequel on demande à l'utilisateur de choisir un des utilisateurs depuis la base de données. Une fois le formulaire validé cela nous ramène au deuxième fichier qui est la deuxième page du site, dans lequel nous affichons simplement toutes les musiques de la playlist de l'utilisateur sélectionné.