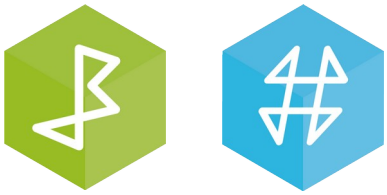
Chu Thomas
Guillou Aurélien
Giraud Thomas
Lachat Antoine

IUT de Belfort
Département Informatique
S2-B1
Groupe numéro 11

Date du rapport : 1 avril 2022

Compte rendu de choix pour l'hébergement de l'application *Peintures*



Sommaire

1 Description des choix d’hébergement possibles....4
1.1 Choisir un hébergeur web....4
1.2 Virtualiser son serveur....5
2 Description des choix logiciels possibles....5
2.1 Choix du serveur web....5
2.2 Mise en production de l’application....6
2.3 Sécurisation du service web....6
2.4 Service de base de données....7
2.5 Uploader son application....7
2.6 Sécurisation de l’accès....7

1 Description des choix d’hébergement possibles

1.1 Choisir un hébergeur web

Un hébergeur désigne l'entité qui propose comme service l'hébergement d'un site Web. Il permet aux internautes d’accéder au site via internet. L’hébergement est permis grâce à des serveurs.

Les types d’hébergement possibles sont :

- Le **mutualisé :** consiste à louer une partie de l’espace de stockage d’un serveur pour y héberger son site. Cette solution est la plus communément utilisée car elle offre l’avantage d’être bon marché et de convenir parfaitement à une majorité de sites, malgré le fait que les performances d’un site sont directement impactées par celles des autres sites.
- Le **dédié :** l’hébergement d’un site web se fait sur la totalité d’un serveur. Il reste donc un moyen très performant et plus sécurisé que le type mutualisé . Par ailleurs, il est plus coûteux en terme d’argent et demande des connaissances techniques de configuration.
- Le **VPS :** (*Virtual Private Server)* découpe un serveur physique en plusieurs serveurs virtuels grâce à des logiciels permettant d’héberger plusieurs sites web. Il a un bon rapport performance/prix.

Les performances de l’hébergeur dépendent de la bande passante (quantité de données envoyées sur le réseau simultanément). Plus il y a d’informations envoyées simultanément, plus la vitesse de chargement des pages sera longue.

La **RAM** (mémoire vive), permet de stocker temporairement les données et requêtes répétitives nécessaires au bon fonctionnement de votre site. Plus le site est riche en fichiers, plus il demandera de mémoire vive.

Nous vous proposons d’utiliser l'hébergeur Hostinger car il propose beaucoup d’avantages, notamment un accès SSH permettant de protéger les communications avec les serveurs distants, une bande passante illimitée assurant la fonctionnalité du site si le trafic dépasse les limites de la bande passante, un SSL permettant de chiffrer les données et de s’assurer qu’elles ne seront jamais compromises et un DNS pour se protéger des attaques DDoS pour un prix de 2,99€ par mois sachant que l’abonnement le moins cher serait insuffisant car il ne permet pas une sécurisation suffisante (absence de SSH, aucun DNS).

1.2 Virtualiser son serveur

La virtualisation d’un serveur consiste à utiliser les ressources de la machine pour faire fonctionner plusieurs OS en simultané à l’aide d’un hyperviseur, qui va permettre de créer et de gérer plusieurs machines virtuelles en répartissant les ressources matérielles en fonction des besoins des différentes machines virtuelles. Cette opération a plusieurs avantages :

- réduit les coûts lié à la gestion des serveurs
- facilite la maintenance du parc informatique
- sécurise l’activité en séparant les différentes activités
- facilite l’accès à distance aux données et aux logiciels de l’utilisateur

La méthode la plus courante de virtualisation est l’utilisation de machines virtuelles, mais il existe aussi la possibilité d’utiliser Docker, qui est un outil utilisant des conteneurs pour faciliter la création, le déploiement et l'exécution d’une application, et qui lie l’application et ses dépendances à un conteneur. La différence entre la virtualisation au sens machine virtuelle et la virtualisation au sens conteneurisation est donc principalement que la docker n’est pas faite pour isoler les applications les unes des autres comme le fait une machine virtuelle, mais va plutôt exécuter plusieurs services séparément en les faisant communiquer.

2 Description des choix logiciels possibles

2.1 Choix du serveur web

Passons maintenant à une sélection de choix logiciels possibles :

Pour le choix de serveur, nous avons retenu deux possibilités : Apache et NGINX

Ils sont à l’heure actuelle les deux serveurs web les plus performant. Les deux offrent une performance de qualité, ils sont capables, flexibles et puissants. Choisir quel serveur est le plus adapté à votre situation dépend grandement de vos besoins personnels et des comportements que vous prédisez pour votre site. Par exemple, à titre comparatif, Apache crée des processus qui gèrent les connections et est un serveur grandement utilisé à travers le monde, NGINX est dit plus tourné vers une logique événementielle, non-bloquante et asynchrone. NGINX est également très rapide et gère le contenu statique plus rapidement qu’Apache, il est aussi plus léger. Enfin, une note qui revient souvent est que NGINX est facile à configurer comparé à Apache qui lui est célèbre pour sa configuration difficile.

Globalement, d'après de nombreuses sources sur Internet, NGINX semble monter en popularité. Apache stagne un peu bien que toujours très utilisé à travers le globe. Une note importante à ajouter et que NGINX nécessite un abonnement pour des fonctionnalités plus poussées.

2.2 Mise en production de l'application

Maintenant, admettons que vous êtes sur un système Linux Debian, que vous avez choisi par exemple le serveur Gunicorn et vous voulez faire la mise en production de votre application web. Vous aurez tout d'abord besoin d'installer pour créer votre environnement python. Les paquets python3-pip et quelques autres nécessaires au développement tel que build-essential libssl-dev python3-dev libffi-dev python3-setuptools. Maintenant que ces paquets sont installés, passons à la création de l'environnement virtuel :

- Installez le module venv avec le paquet python3-venv puis créer un environnement virtuel avec la commande venv

En admettant que vous avez déjà installé Flask et Gunicorn, passons à la configuration du Gunicorn. Dans votre projet, vous allez utiliser un fichier python qui vous servira de point d'entrée, cela permettra au serveur Gunicorn d'interagir avec l'application, nous l'appellerons entry.py pour cet exemple. Vous pouvez vérifier que Gunicorn peut bien servir l'application en utilisant la commande gunicorn le paramètre –bind et le nom du point d'entrée dans notre cas, entry:app.

Vous pouvez maintenant désactiver l'environnement virtuel, gérer les métadonnées et les dépendances du projet dans le fichier .service de votre projet contenu dans */etc/systemd/system* . Dans le même fichier, vous pouvez préciser vos informations dans la section Service. Adaptez la section Service à votre projet, nous voulons que ce service démarre quand l'application multi-user est en train de tourner. Vous pouvez normalement lancer le service Gunicorn que vous venez de créer et l'activer avec *enable* pour lui permettre de boot.

2.3 Sécurisation du service web

Il faut maintenant vous offrir la possibilité de travailler en toute sécurité, depuis n'importe quel accès internet, sur toutes les ressources de votre entreprise comme si vous y étiez via une connexion sécurisée. Pour cela une solution que nous vous proposons est le certificat SSL, un fichier de données qui lie une clé cryptographique aux informations d'une organisation, installé sur un serveur, activé par le protocole HTTPS. Le certificat possédera notamment les directives :

- **SSLCACertificateFile** : spécifie le chemin vers le fichier qui contient les certificats racines
- **SSLCertificateFile** : spécifie l'emplacement du certificat SSL qui doit être utilisé par une machine spécifique

- **SSLCertificateKeyFile** : spécifie le chemin vers la clé privée
- **SSLEngine** : directive qui détermine si le protocole SSL est activé ou non pour un serveur virtuel spécifique

2.4 Service de base de données

Il vous faut un service de base de données allant avec votre site de e-commerce. On pense tout de suite aux deux SGBD les plus utilisés à savoir Oracle et MySQL. Oracle SQL a une licence payante et est plus difficile à apprendre que MySQL. Oracle aurait été un bon choix si le site nécessitait une grosse base de données, mais ce n'est pas le cas pour votre projet. Nous vous conseillons donc MySQL qui est open-source et très simple à utiliser.

2.5 Uploader son application

Maintenant que vous disposez d'un serveur, il est très utile d'avoir un outil vous permettant de transférer vos fichiers depuis votre ordinateur vers votre serveur et inversement. Il existe différents protocoles qui vous permettent de réaliser cette opération dont : le FTP.

Sur Debian, le serveur FTP devrait s'appeler vsftpd, s'il n'est pas installé, installez-le, passez à la configuration. Le fichier devrait se trouver dans le répertoire */etc*. Définissez *local_enable write_enable* et *chroot_local_user* à *YES* et redémarrez le serveur FTP. Le protocole STFP est également un protocole de transfert de fichier développé pour résoudre quelques problèmes du protocole FTP. Ce protocole fait partie de l'application SSH et nécessite donc une configuration additionnelle de SSH sur votre machine. Il est important de noter que ces deux protocoles sont différents, le protocole STFP transfère par les fichiers sous forme de données binaires et non pas sous forme de texte par exemple.

2.6 Sécurisation de l'accès

Enfin, des précautions additionnelles ne sont pas négligeable pour la mise en service de votre site de vente. Une des premières choses que l'on peut faire, c'est configurer SSH. Dans le fichier /*etc/ssh/sshd_config*, nous pouvons changer le port de connexion par défaut pour éviter certaines attaques. Après modification, redémarrez le service SSH dans le répertoire */etc/init.d*.

Vous pouvez également faire des *chmod* pour n'autoriser les compilateurs et installeurs que pour le super utilisateur.

Fail2ban est également une bonne alternative, c'est un script surveillant l'accès réseau grâce aux logs des serveurs. Ce script peut détecter les erreurs d'authentification répétées et prend des contre-mesures en bannissant l'adresses IP grâce à *iptables*. Après installation, vous pouvez configurer le service dans le fichier /*etc/fail2ban/fail2ban.conf* pour notamment définir le fichier ou vous pourrez visionner les logs et le niveau de détail de ces derniers. Après modification, pensez à redémarrer fail2ban dans /*etc/init.d* .