**Montez un serveur de fichier sous linux**

# Qu’est-ce qu’un serveur ?

### Le serveur

Un serveur est un logiciel qui fournit un service à d’autres logiciels. Bien souvent, il fournit ce service via le réseau informatique mais il peut aussi être installé sur la même machine que les clients.

Par extension, on parle aussi de serveur pour désigner la machine physique (ou virtuelle) dont le but principal est de faire tourner un ou plusieurs logiciels serveurs.

En pratique, vous verrez que le terme de serveur est employé aussi bien pour désigner le logiciel que le matériel ou même les services fournis. C’est le contexte qui vous permettra de savoir de quoi on parle vraiment.

### Le modèle client-serveur

En fait, c’est exactement la même chose quand vous êtes assis à la terrasse d’un café. Vous êtes le client et vous demandez une boisson au serveur.

**Le serveur fournit un service** : il apporte des boissons. Pour obtenir une boisson, le client doit en faire la **demande** au serveur, il doit faire une **requête**.

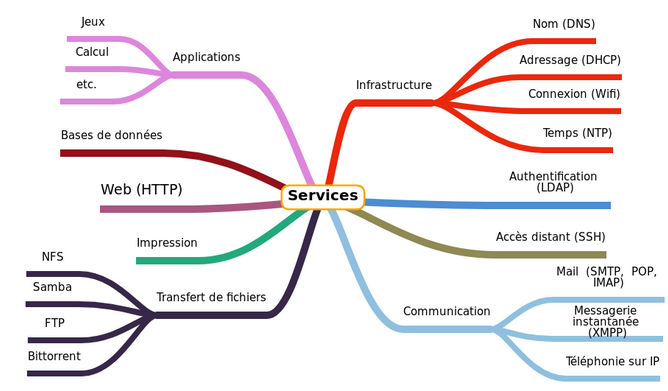
Pour que le client puisse faire une demande au serveur et être compris, il aura besoin d’un langage commun avec le serveur. En informatique, lorsqu’un client et un serveur communiquent, ils utilisent un **protocole réseau**.

Il faut bien comprendre que dans le modèle client-serveur, chaque entité joue le rôle du client ou du serveur dans un contexte donné mais peut **changer de rôle dans un autre contexte**.

Le modèle pair-à-pair (peer-to-peer en anglais) est un cas particulier du modèle client-serveur dans lequel toutes les entités du réseau jouent à la fois le rôle de client et de serveur

Les services

Voici un petit aperçu des services les plus courants :



### Les caractéristiques d’un serveur

* **Un serveur est généralement allumé 24h/24 et 7j/7**.
* Très souvent, **il ne dispose ni d’écran, ni de clavier, ni d’équipements multimédia** (carte son, etc.). Un serveur interagit principalement avec d’autres machines et logiciels donc il est simplement connecté au réseau et l’administration du serveur se fait à distance.
* **Il est généralement entreposé dans une salle machine** dont l’accès est sécurisé et qui présente des sécurités vis-à-vis des risques liés au feu et à l’eau. **L’alimentation en électricité et la connexion au réseau peuvent être redondées.**
* **Un serveur utilise généralement un système d’exploitation spécialisé**. Vous vous souvenez ? Un système d’exploitation c’est par exemple Windows, MacOS, GNU/Linux. La plupart des systèmes d’exploitation proposent des versions spéciales pour serveur (je vous en dirai plus ci-dessous).
* **Un serveur peut disposer de performances physiques plus importantes qu’un poste client**.



### Les OS serveur

* Comme les serveurs restent allumés en permanence, les OS ne sont pas configurés avec des fonctions de mise en veille.
* Les OS Serveur sont fournis avec peu de pilotes multimédia mais sont plutôt configurés pour supporter le matériel qu’on trouve sur les serveurs (support de composants matériels spécifiques, de grandes quantités de RAM ou de nombreux processeurs)
* Les OS Serveur sont installés sans interface graphique pour les systèmes Linux et avec une interface sans effets esthétiques pour les systèmes Windows
* Le programme d’installation est pensé à destination des professionnels avec une esthétique plus sobre et des choix plus techniques
* Les licences et systèmes de tarification sont généralement différents et bien plus chers
* Il est possible de souscrire à des services de support payants pensés pour les entreprises

Au final, retenez que **la notion essentielle qui se cache derrière le terme de serveur est celle de service**.

# Construisez une solution adaptée à vos besoins

Chaque **architecture** reste spécifique à un ensemble de besoins.

### Les fonctionnalités

La première chose à savoir pour construire un système informatique c’est le type de service que vous voulez fournir. Vous pourrez ensuite avoir une première idée de l’environnement logiciel et matériel nécessaire.

*Par exemple*: votre but est d’avoir un blog pour partager vos découvertes avec le monde entier. Après avoir passé du temps à évaluer les fonctionnalités que vous souhaitez pour votre blog et les différents outils disponibles, vous vous tournez vers le logiciel WordPress. Vous voyez que pour utiliser WordPress, vous aurez besoin d’un serveur web (par exemple Apache), d’une base de données (par exemple MySQL) et d’une version récente de PHP. Tous ces logiciels sont disponibles sous les principaux systèmes d’exploitation. Un blog peut tourner sur un matériel générique.

### La performance

Vous aurez également besoin de définir vos **critères d’exigence en terme de performance**, c’est-à-dire les critères objectifs qui montrent que votre serveur fonctionne comme vous voulez. Des critères courants sont **le temps de réponse maximum** de votre serveur et **le nombre d’utilisateurs simultanés** que peut accepter votre serveur.

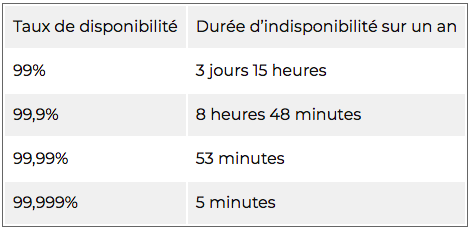
Vous devrez adapter la taille de votre architecture en fonction de ces critères. Pour cela, dans un premier temps, on fait généralement une estimation à partir de ses propres expériences passées. En tant que débutant, vous pouvez chercher sur Internet ou dans votre entourage des retours d’expériences similaires. Il est parfois difficile d’avoir des détails mais rappelez vous qu’à ce stade, on ne cherche que des ordres de grandeur.

Au final, pour être sûr que votre système fournit réellement les performances attendues, vous devrez faire un **test de charge**. Un test de charge permet de simuler, grâce à un logiciel spécifique (**[JMeter](https://jmeter.apache.org/)**, **[Gatling](https://gatling.io/)**, etc.), l’audience maximum que vous attendez pour voir si votre service se comporte comme prévu.

Malheureusement, on ne sait pas toujours prédire son audience et de toute façon, cette dernière sera probablement amenée à grandir avec le temps (on vous le souhaite). Vous devrez donc anticiper cette croissance et vous demander comment faire grossir votre architecture (on parle de “**scalability**” en anglais, ce terme est souvent repris en français quitte à le franciser en “scalabilité”). Une manière courante de gérer la croissance d’un service est de répartir le travail sur un plus grand nombre de machines.

### La disponibilité

Sur un an, quelle durée maximum acceptez-vous que votre service soit indisponible ? Dans le cadre d’un accord avec un partenaire ou un client, on parle de **SLA** (Service Level Agreement, un contrat de qualité de service en français), et c’est généralement exprimé en pourcentage de disponibilité garantie (sous peine de pénalités).

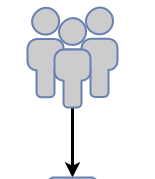


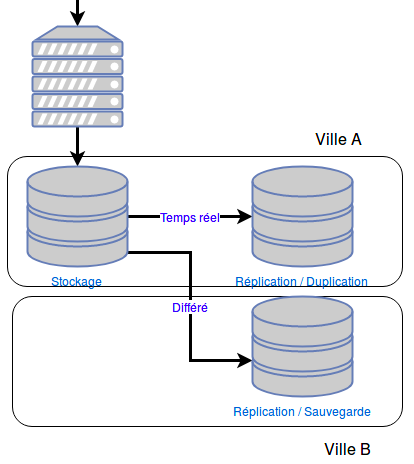
Pour augmenter la disponibilité d’un service, on utilise généralement la redondance de tous les éléments : si un élément est défaillant, un autre prend le relai pour assurer une continuité de service. On cherche à éviter que le service ne dépende que d’un unique élément. Si un tel élément existe, on parle de **SPOF** (Single Point Of Failure, point unique de défaillance).

### La sécurité

La sécurité concerne la prévention de tous les **risques accidentels**.

L’élément le plus critique du point de vue de la sécurité concerne probablement **les données**. Le moyen le plus courant pour protéger des données est la duplication. Pour un niveau de sécurité supérieur, il est nécessaire d’avoir un système de duplication différée pour éviter qu’une erreur humaine ou logicielle ne supprime des données sur toutes les copies en même temps.





### La sureté

La sûreté concerne la prévention des **actes malveillants**. Votre architecture peut jouer un rôle important dans la sûreté de votre système. Vous devez veiller à n’exposer que le minimum de services nécessaires sur Internet. En fonction de votre niveau d’exigence, vous pouvez également décider d’installer des logiciels dédiés à la sécurité tels que pare-feu ou système de détection d’intrusion.

Par exemple une base de données ne doit généralement pas être directement accessible sur Internet.

### Vos ressources

Autre point à prendre en compte : vos ressources qui peuvent être votre budget, vos connaissances, le nombre de personnes disponibles, le temps dont vous disposez, vos soutiens (appel ponctuel à un ami ou support payant par exemple) ou encore les services, logiciels et matériels disponibles sur le marché.

Un niveau d’exigence élevé dans les critères vus auparavant aura tendance à augmenter le coût et la complexité technique de votre architecture mais vous devrez trouver un équilibre entre vos exigences et vos ressources disponibles.

Enfin, vous devrez évaluer vos ressources dans deux contextes différents :

* au moment de l’installation de votre service
* pour l’exploitation au quotidien.

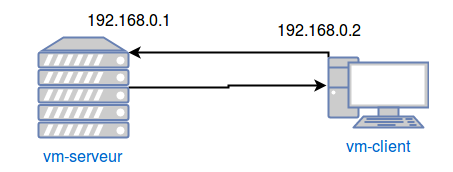
Pour les entreprises, on parle de CAPEX pour les investissements (achats de machines, etc.) et d’OPEX pour les coûts d’exploitation (salaire des employés, location mensuelle d’une machine chez un hébergeur, etc.). Dans les grosses structures, ce sont souvent des budgets séparés car ils sont différents d’un point de vue comptable.

### Vos contraintes spécifiques.

Pour construire une architecture réellement adaptée à vos besoins, vous devez également faire le bilan de vos contraintes. Peut-être que votre patron, votre client ou vos partenaires vous imposent certaines conditions ? Ou encore des dispositions législatives vous imposent de conserver les données dans votre pays d’origine ou un niveau de sûreté particulièrement élevé ? Ou enfin, peut-être que le respect de vos valeurs vous amène à ne vouloir utiliser que des logiciels libres ?

Créez votre environnement de travail

En termes d’architecture, pour les besoins de ce cours, nous allons faire très simple: une machine virtuelle pour le serveur et une autre machine virtuelle pour le client.



### Installation du client

Avec VirtualBox on installe Ubuntu Desktop

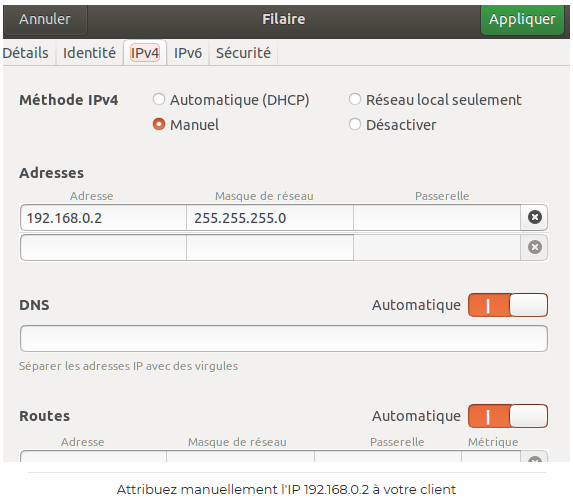
### Configuration du réseau de votre client

Éteignez votre machine virtuelle “vm-client” puis dans le gestionnaire de machine VirtualBox, sélectionnez votre machine et cliquez sur “configuration” -> “Réseau”.

Vous voyez bien votre carte 1 en “NAT”, cliquez sur la carte 2 et sélectionnez “Réseau interne”. Cliquez sur “OK” et démarrez votre machine virtuelle : elle aura maintenant deux cartes réseaux et nous allons configurer cette deuxième carte.

Une fois authentifié, cliquez sur l’icône réseau en haut à droite puis sur la carte réseau marquée comme “éteinte” et “paramètres filaire”.

L’application de configuration du réseau s’ouvre. Cliquez sur la petite roue crantée pour configurer votre nouvelle carte.



Dans l’onglet IPv4, choisissez simplement “manuel” et entrez l’adresse “192.168.0.2" et le masque “255.255.255.0”. Cliquez sur “Appliquer”. Vous avez terminé la configuration réseau de votre client.

Il ne reste plus qu’une petite chose à faire pour se simplifier la vie plus tard : ajouter la ligne suivante à votre fichier  /etc/hosts : 192.168.0.1 HostnameDuServeur

### Installation du serveur

Avec VirtualBox on installe la version LTS d’Ubuntu Server

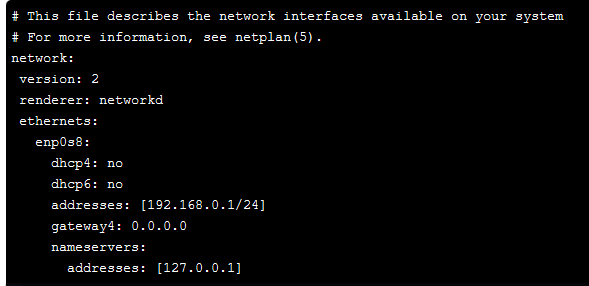
LTS signifie Long Term Support, soit Support Étendu. Elle correspond à la version d’Ubuntu qui sort tous les deux ans et qui bénéficie d’un support étendu à 5 ans au lieu de neuf mois habituellement. On préfèrera cette version d’Ubuntu sur un serveur pour éviter d’avoir à tout réinstaller deux fois par an.

Attention car le clavier sera en anglais

### Configuration du réseau de votre serveur

Par la commande  ip a  vous pouvez voir l’interface locale "lo” ainsi que vos deux interfaces réseaux. Normalement, celle que l’on vient de rajouter et qu’on va configurer ici est celle qui n’a pas encore d’adresse IP (ligne qui commence par  inet  ). Notez son nom : chez moi elle s’appelle  enp0s8 .

Pour attribuer une IP statique à cette interface, créez le fichier  /etc/netplan/60-lan\_statique.yaml :



Vous remarquerez la ligne  renderer: networkd  qui indique le programme qui va appliquer les instructions de ce fichier de configuration. Vous avez le choix entre deux programmes : *NetworkManager* plutôt utilisé sur les versions “Desktop” (allez voir vos fichiers /etc/netplan/\* sur votre client) et *networkd* plutôt utilisé sur les versions “Server” et qu’on utilise donc ici.

Ensuite, on trouve la description de nos cartes Ethernet. La mienne répond au doux nom d’enp0s8 mais vous devrez adapter cette ligne au nom de votre carte. On configure cette carte pour ne pas utiliser DHCP, on lui alloue une IP statique, une passerelle par défaut et un serveur DNS.

Une fois le fichier enregistré, vous pouvez appliquer la nouvelle configuration par la commande :



En refaisant un  ip a, vous devriez voir que votre interface a bien une IP.

Enfin, pour vous simplifier la vie, ajoutez la ligne suivante à votre fichier   /etc/hosts

Ensuite avec chaque machine on ping l’autre afin de vérifier qu’elles communiquent bien ensemble

Puis lance sur chacune un sudo apt-get update && sudo apt-get upgrade

# Maîtrisez le partitionnement et le montage

D’ici la fin de ce cours, vous aimeriez pouvoir utiliser votre nouveau serveur pour partager des fichiers sur votre réseau local. Nous allons pour cela dédier un disque dur au stockage de ces données.

Ajoutez un disque dur à votre serveur

Partitionnez votre nouveau disque

Formatez votre nouveau disque

Montez votre nouveau disque

1