

Putting it all together | 29 Septembre, 2025

Auteur : GhostHunt3rx

Email : anonymousinconnu10@gmail.com

Tâche 1 : Tout mettre en place

Tout mettre en place

Dans les modules précédents, vous avez appris que beaucoup de choses se passent en arrière-plan lorsque vous demandez une page Web dans votre navigateur.

En résumé, lorsque vous demandez un site web, votre ordinateur doit connaître l'adresse IP du serveur avec lequel il doit communiquer ; pour cela, il utilise le DNS. Votre ordinateur communique ensuite avec le serveur web à l'aide d'un ensemble spécial de commandes appelé protocole HTTP ; le serveur web renvoie alors du code HTML, JavaScript, CSS, des images, etc., que votre navigateur utilise pour formater et afficher correctement le site web.



Il existe également quelques autres composants qui contribuent à améliorer l'efficacité du Web et offrent des fonctionnalités supplémentaires.

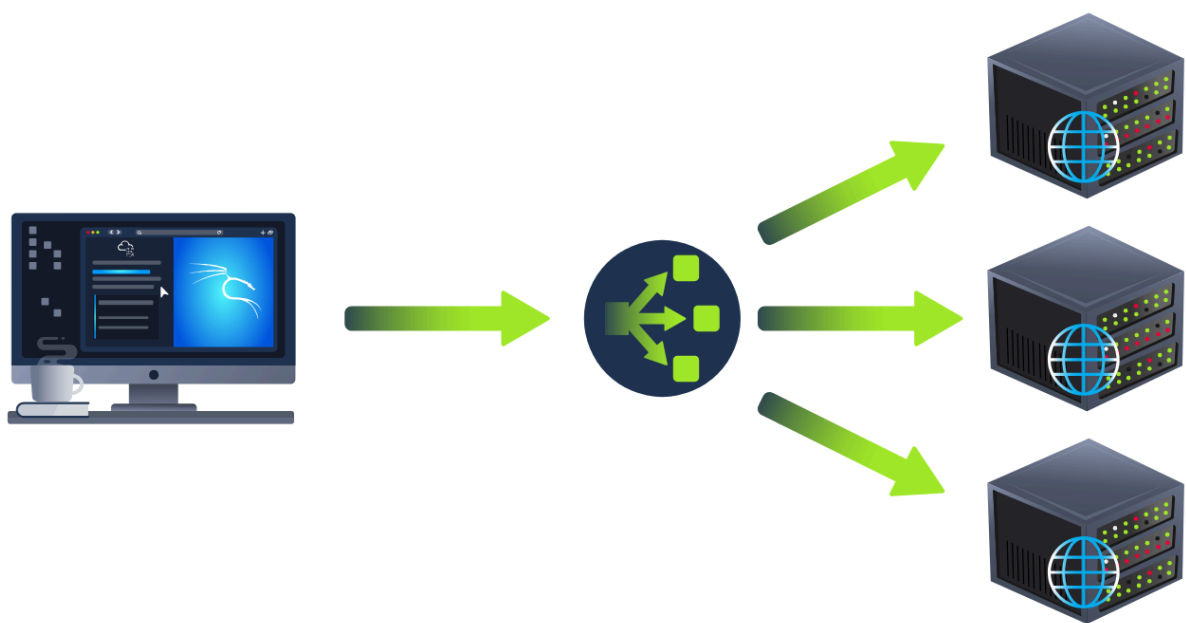
Tâche 2 : Autres composants

Équilibreur de charge

Lorsque le trafic d'un site Web commence à devenir très important ou qu'il exécute une application qui nécessite une haute disponibilité, un seul serveur Web peut ne plus suffire. Les équilibreurs de charge offrent deux fonctionnalités principales : ils garantissent que les sites Web à fort trafic peuvent gérer la charge et assurent un basculement si un serveur ne répond plus.

Lorsque vous effectuez une requête sur un site Web avec un équilibreur de charge, celui-ci reçoit d'abord votre requête, puis la transmet à l'un des multiples serveurs qui se trouvent derrière lui. Le répartiteur de charge utilise différents algorithmes pour déterminer quel serveur est le mieux à même de traiter la requête. Parmi ces algorithmes, on peut citer le round-robin, qui envoie la requête à chaque serveur à tour de rôle, ou le weighted, qui vérifie le nombre de requêtes traitées par chaque serveur et envoie la requête au serveur le moins sollicité.

Les **équilibreurs de charge** effectuent également des vérifications périodiques auprès de chaque serveur afin de s'assurer qu'ils fonctionnent correctement ; c'est ce qu'on appelle un contrôle d'intégrité. Si un serveur ne répond pas correctement ou ne répond pas du tout, l'équilibreur de charge cessera d'envoyer du trafic jusqu'à ce qu'il réponde à nouveau correctement.



CDN (Content Delivery Networks)

A CDN can be an excellent resource for cutting down traffic to a busy website. It allows you to host static files from your website, such as JavaScript, CSS, Images, Videos, and host them across thousands of servers all over the world. When a user requests one of the hosted files, the CDN works out where the nearest server is physically located and sends the request there instead of potentially the other side of the world.

Databases

Often websites will need a way of storing information for their users. Webservers can communicate with databases to store and recall data from them. Databases can range from just a simple plain text file up to complex clusters of multiple servers providing speed and resilience. You'll come across some common databases: MySQL, MSSQL, MongoDB, Postgres, and more; each has its specific features.

WAF (Web Application Firewall)

A WAF sits between your web request and the web server; its primary purpose is to protect the webserver from hacking or denial of service attacks. It analyses the web requests for common attack techniques, whether the request is from a real browser rather than a bot. It also checks if an excessive amount of web requests are being sent by utilising something called rate limiting, which will only allow a certain amount of requests from an IP per second. If a request is deemed a potential attack, it will be dropped and never sent to the webserver.



Tâche 3: Fonctionnement des serveurs Web

Comment fonctionnent les serveurs Web.

Qu'est-ce qu'un serveur Web ?

Un **serveur Web** est un logiciel qui écoute les connexions entrantes, puis utilise le **protocole HTTP** pour fournir du contenu Web à ses clients.

Les logiciels de serveur Web les plus courants sont *Apache*, *Nginx*, *IIS* et *NodeJS*.

Un serveur Web fournit des fichiers à partir de ce qu'on appelle son répertoire racine, qui est défini dans les paramètres du logiciel. Par exemple, Nginx et Apache partagent le même emplacement par défaut **/var/www/html** dans les systèmes d'exploitation Linux, et IIS utilise **C:\inetpub\wwwroot** pour les systèmes d'exploitation Windows. Ainsi, si vous demandez le fichier `http://www.example.com/picture.jpg`, il enverra le fichier `/var/www/html/picture.jpg` depuis son disque dur local.

Hôtes virtuels

Les serveurs web peuvent héberger plusieurs sites web avec des noms de domaine différents ; pour ce faire, ils utilisent des hôtes virtuels. Le logiciel du serveur web vérifie le nom d'hôte demandé dans les en-têtes HTTP et le compare à ses hôtes virtuels (les hôtes virtuels sont simplement des fichiers de configuration au format texte). S'il trouve une correspondance, le site web correct sera fourni. Si aucune correspondance n'est trouvée, le site web par défaut sera fourni à la place.

Les hôtes virtuels peuvent avoir leur répertoire racine mappé à différents emplacements sur le disque dur. Par exemple, `one.com` peut être mappé à `/var/www/website_one`, et `two.com` à `/var/www/website_two`.

Il n'y a pas de limite au nombre de sites Web différents que vous pouvez héberger sur un serveur Web.

Contenu statique ou dynamique

Comme son nom l'indique, le contenu statique est un contenu qui ne change jamais. Il s'agit généralement d'images, de javascript, de CSS, etc., mais cela peut également inclure du HTML qui ne change jamais. De plus, ces fichiers sont directement servis par le serveur web sans aucune modification.

Le contenu dynamique, en revanche, est un contenu qui peut changer en fonction des différentes requêtes. Prenons l'exemple d'un blog. La page d'accueil du blog affiche les dernières entrées. Si une nouvelle entrée est créée, la page d'accueil est alors mise à jour avec la dernière entrée. Un deuxième exemple pourrait être une page de recherche sur un blog. Selon le mot que vous recherchez, différents résultats s'affichent.

Ces modifications de ce que vous voyez sont effectuées dans ce qu'on appelle le backend à l'aide de langages de programmation et de script. On parle de backend car tout ce qui est fait se passe en arrière-plan. Vous ne pouvez pas consulter le code HTML du site web et voir ce qui se passe dans le backend, alors que le code HTML est le résultat du traitement effectué par le backend. Tout ce que vous voyez dans votre navigateur s'appelle le frontend.

Langages de script et backend

Langages de script et langages backend

Les langages backend offrent des possibilités quasi illimitées et permettent de rendre un site web interactif pour l'utilisateur. Voici quelques exemples de ces langages (sans ordre particulier :p) : PHP, Python, Ruby, NodeJS, Perl et bien d'autres encore. Ces langages peuvent interagir avec des bases de données, appeler des services externes, traiter les données de l'utilisateur, et bien plus encore. Voici un exemple très simple en PHP : imaginez que vous consultiez le site web <http://example.com/index.php?name=adam>.

Si le fichier index.php était construit comme suit :

```
<html><body>Bonjour <?php echo $_GET["name"]; ?></body></html>
```

Le résultat affiché au client serait le suivant :

```
<html><body>Bonjour adam</body></html>
```

Vous remarquerez que le client ne voit aucun code PHP, car celui-ci se trouve dans le backend. Cette interactivité soulève de nombreux problèmes de sécurité pour les applications web qui n'ont pas été créées de manière sécurisée, comme vous l'apprendrez dans les modules suivants.

Tache 4 : Quiz

Le quiz

Cliquez sur le bouton « Afficher le site » à droite. En utilisant tout ce que vous avez appris dans les autres modules, faites glisser et déposez les tuiles dans l'ordre correct correspondant au fonctionnement d'une requête vers un site web afin de révéler le drapeau.

Remarque : lorsque vous placez une tuile à la bonne position, elle s'affiche en vert. Lorsqu'une tuile est mal placée, elle s'affiche en rouge. Veillez à ne pas actualiser la page, car cela réinitialiserait toutes les tuiles !