MISE EN PLACE D'UNE SOLUTION DE TYPE REVERSE PROXY SOUS LINUX

Document qui explique la mise en place de la solution de type reverse proxy

ASSURMER

Services informatiques

Version : 8.0

Service IT

14/12/2022

Kevin ORTIZ

Antoine ENGASSER

Antonyn HIBOUX

I. Fonctionnalités d'un serveur Proxy

Comment fonctionne un Serveur Proxy ? A quoi sert-il ?

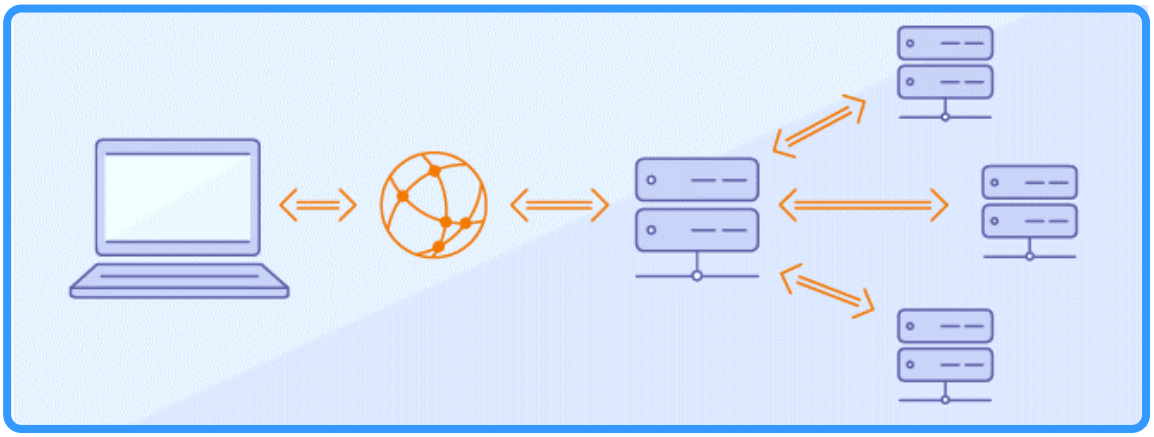
Un serveur proxy est un serveur qui agit en tant que relais pour les requêtes des clients d'un réseau vers d'autres serveurs. Il peut avoir plusieurs fonctionnalités, notamment :

- **Cacher les détails de l'architecture du réseau des clients** : en agissant en tant que relais, le serveur proxy peut cacher les détails de l'architecture du réseau des clients aux serveurs distants, ce qui peut aider à améliorer la sécurité du réseau en empêchant les attaquants de cibler directement les clients.
- **Contrôler l'accès à Internet** : les administrateurs peuvent utiliser un serveur proxy pour contrôler l'accès des clients à Internet en autorisant ou en bloquant l'accès à certaines parties du Web.
- **Améliorer les performances** : le serveur proxy peut stocker en cache les données des réponses aux requêtes fréquemment utilisées, ce qui peut accélérer les temps de réponse pour les utilisateurs et réduire la charge sur les serveurs distants.
- **Filtrer le contenu** : les administrateurs peuvent utiliser un serveur proxy pour filtrer le contenu qui passe à travers lui en bloquant ou en autorisant l'accès à certains types de contenu en fonction de critères prédéfinis.

En résumé, un serveur proxy peut être utilisé pour améliorer la sécurité, la vitesse et la flexibilité d'un réseau en agissant en tant que relais pour les requêtes des clients.

Qu'est-ce qu'un Reverse proxy ?

Un Reverse Proxy est un type de serveur qui sert d'intermédiaire entre les clients et d'autres serveurs sur un réseau. Le proxy inverse est souvent utilisé pour améliorer l'évolutivité et la disponibilité d'un site web ou d'une application en permettant à plusieurs serveurs de partager la charge de travail et de travailler plus efficacement. En outre, le Reverse Proxy peut également fournir certaines fonctionnalités supplémentaires, telles que la mise en cache du contenu, le filtrage du trafic et la protection contre les attaques par déni de service. Dans l'ensemble, le proxy inverse est un outil précieux pour toute organisation qui doit améliorer les performances et la sécurité de son site web ou de son application.



Quels sont les avantages d'un serveur Proxy ?

Les avantages d'un serveur proxy peuvent varier en fonction du type de serveur proxy utilisé et de la manière dont il est configuré. Comme pour toutes les technologies, les serveurs proxy ont également des inconvénients, les avantages et inconvénients sont :

Avantages	Inconvénients
• **Meilleure sécurité** : en agissant en tant que relais pour les requêtes des clients, un serveur proxy peut aider à protéger les clients en masquant leur adresse IP et en empêchant les attaquants de cibler directement les clients. • **Meilleures performances** : en stockant en cache les données des réponses aux requêtes fréquemment utilisées, un serveur proxy peut accélérer les temps de réponse pour les utilisateurs et réduire la charge sur les serveurs distants. • **Contrôle de l'accès** : les administrateurs peuvent utiliser un serveur proxy pour contrôler l'accès des clients à Internet en autorisant ou en bloquant l'accès à certaines parties du Web. • **Filtre du contenu** : les administrateurs peuvent utiliser un serveur proxy pour filtrer le contenu qui passe à travers lui en bloquant ou en autorisant l'accès à certains types de contenu en fonction de critères prédéfinis.	• **Coût** : l'installation et la maintenance d'un serveur proxy peuvent être coûteuses, en particulier pour les entreprises qui ont besoin de serveurs proxy puissants pour gérer un grand nombre de requêtes. • **Latence** : en agissant en tant que relais pour les requêtes des clients, un serveur proxy peut ajouter une certaine latence à la communication, ce qui peut ralentir les temps de réponse pour les utilisateurs. • **Faiblesse de sécurité** : bien que les serveurs proxy puissent offrir une certaine sécurité en masquant l'adresse IP des clients, ils peuvent également être la cible d'attaques si les paramètres de sécurité ne sont pas correctement configurés. • **Incompatibilité avec certains protocoles** : certains protocoles de communication, tels que le protocole de transfert de fichiers (FTP) ou le protocole de transfert de courrier électronique (SMTP), ne peuvent pas être utilisés avec certains types de serveurs proxy.

Les différents types de serveurs Proxy

Il existe plusieurs types de serveurs proxy, qui peuvent être classés en fonction de différents critères tels que leur fonctionnement, leur configuration ou leur utilisation :

- **Serveur proxy transparent** : ce type de serveur proxy agit en tant que relais pour les requêtes des clients sans qu'ils aient à configurer explicitement leur navigateur pour utiliser le proxy. Le serveur proxy transparent peut être utilisé pour améliorer les performances en mettant en cache les réponses aux requêtes fréquemment utilisées.
- **Serveur proxy anonyme** : ce type de serveur proxy ne transmet pas les détails d'identification des clients aux serveurs distants, ce qui permet aux clients de naviguer sur Internet de manière anonyme. Le serveur proxy anonyme peut être utilisé pour protéger la vie privée des utilisateurs en empêchant les sites Web de collecter des informations sur leur activité en ligne.

- **Serveur proxy à authentification** : ce type de serveur proxy nécessite que les utilisateurs fournissent des informations d'authentification avant de pouvoir accéder au contenu bloqué. Cela peut être utilisé pour restreindre l'accès à certains types de contenu à certaines personnes ou groupes.
- **Serveur proxy inverse** : ce type de serveur proxy est utilisé pour acheminer les requêtes des clients vers un serveur Web spécifique. Il est souvent utilisé dans les environnements d'entreprise pour diriger les utilisateurs vers un serveur interne lorsqu'ils tentent d'accéder à un site Web spécifique.
- **Serveur proxy d'application** : ce type de serveur proxy est spécialisé dans un type particulier d'application, comme le courrier électronique ou le transfert de fichiers, et agit en tant que relais pour les requêtes de cette application.
- **Serveur proxy distribué** : ce type de serveur proxy est en réalité un groupe de serveurs proxy qui travaillent ensemble pour distribuer la charge et améliorer les performances. Cela peut être utile pour les sites Web à très forte fréquentation pour gérer efficacement un grand nombre de requêtes.

En résumé, il existe différents types de serveurs proxy qui peuvent être utilisés dans différents contextes pour différentes fonctionnalités.

II. La Cybersécurité des serveurs Proxy

La cybersécurité des serveurs proxy

La cybersécurité des serveurs proxy est l'ensemble des mesures de sécurité qui visent à protéger les serveurs proxy contre les attaques et les menaces en ligne. Les serveurs proxy peuvent être cibles d'attaques telles que des attaques par déni de service (DoS), des attaques par injection de code, des attaques de phishing, et d'autres types d'attaques visant à compromettre la sécurité des données ou des informations sensibles stockées sur ces serveurs.

Quel est le rôle d'un serveur proxy dans la cybersécurité ?

Le rôle d'un serveur proxy dans la cybersécurité est de fournir une couche de protection supplémentaire pour les réseaux et les systèmes informatiques en agissant en tant que pare-feu et en contrôlant l'accès aux réseaux et aux données sensibles. Un serveur proxy peut être configuré pour filtrer les données entrantes et sortantes, en bloquant les paquets de données malveillants ou non autorisés, et en autorisant uniquement les paquets de données légitimes. En agissant comme une barrière entre les utilisateurs et le réseau ou les systèmes informatiques, un serveur proxy peut aider à prévenir les attaques informatiques et à protéger les données sensibles.

Les attaques le plus fréquentes

Les attaques les plus fréquentes contre les serveurs proxy sont généralement similaires aux attaques courantes contre d'autres types de serveurs, et peuvent inclure des attaques par déni de service (DoS), des attaques par injection de code, des attaques de phishing, et d'autres types d'attaques visant à compromettre la sécurité des données ou des informations sensibles stockées sur ces

Serveurs. Pour protéger les serveurs proxy contre ces attaques, il est important de mettre en place des mesures de sécurité adéquates, telles que des pares-feux et des logiciels de détection d'intrusion, et de mettre à jour régulièrement les logiciels et les systèmes d'exploitation pour éviter les vulnérabilités connues.

Comment prévenir les attaques ?

Il existe plusieurs mesures que vous pouvez prendre pour prévenir les attaques contre un serveur proxy, notamment :

- Utiliser un pare-feu pour contrôler l'accès à votre serveur proxy et bloquer les paquets de données non autorisés ou malveillants.
- Installer et configurer un logiciel de détection d'intrusion pour détecter et bloquer les attaques en temps réel.
- Mettre à jour régulièrement votre système d'exploitation et les logiciels utilisés sur votre serveur proxy pour corriger les vulnérabilités connues.
- Configurer des stratégies de mot de passe fortes pour protéger l'accès à votre serveur proxy et à vos données sensibles.
- Utiliser des protocoles de chiffrement pour sécuriser les données transitant par votre serveur proxy.
- Former régulièrement le personnel à la cybersécurité et sensibiliser les utilisateurs à la sécurité en ligne.

En suivant ces recommandations, vous pouvez aider à protéger votre serveur proxy contre les attaques courantes et à maintenir la sécurité de vos données sensibles.

III. Analyse des solutions de type Reverse Proxy

C'est quoi une solution de Reverse Proxy sous linux ?

Une solution de reverse proxy sous Linux est un logiciel ou un ensemble de logiciels qui permet de configurer un serveur Linux en tant que reverse proxy. Un reverse proxy est un type de serveur qui agit en tant que point d'entrée pour les requêtes entrantes vers un ou plusieurs serveurs backend. Le reverse proxy transfère les requêtes des utilisateurs vers les serveurs backend, puis renvoie les réponses des serveurs backend aux utilisateurs.

Une solution de reverse proxy sous Linux peut être utilisée pour améliorer les performances d'un site Web en répartissant la charge entre plusieurs serveurs backend, pour sécuriser les communications entre les utilisateurs et les serveurs backend en utilisant des protocoles de chiffrement, ou pour cacher les détails de l'architecture du réseau en masquant les adresses IP des serveurs backend. Il existe plusieurs solutions de reverse proxy pour Linux, notamment Nginx, Apache, et HAProxy.

- **Haproxy**

HAProxy est un logiciel libre de reverse proxy et de charge balancing pour les systèmes d'exploitation Linux, Unix et BSD. Il permet de configurer un serveur en tant que reverse proxy pour gérer les requêtes entrantes et répartir la charge entre plusieurs serveurs backend. Haproxy est souvent utilisé pour améliorer les performances d'un site Web en répartissant la charge entre plusieurs serveurs, pour sécuriser les communications en utilisant des protocoles de chiffrement, ou pour cacher les détails de l'architecture du réseau en masquant les adresses IP des serveurs backend. Haproxy est un logiciel très performant et scalable, et est utilisé par de nombreux sites Web et services en ligne pour gérer les requêtes entrantes.

Fonctionnement d'Haproxy

Le fonctionnement d'Haproxy est d'agit en tant que reverse proxy pour transférer les requêtes entrantes des utilisateurs vers les serveurs backend, puis renvoie les réponses des serveurs backend aux utilisateurs. Lorsqu'une requête arrive à un serveur exécutant Haproxy, celui-ci analyse la requête et la transfère vers le serveur backend approprié en fonction des règles de configuration définies. Le serveur backend traite la requête et envoie une réponse à Haproxy, qui la transfère à l'utilisateur.

Haproxy permet également de répartir la charge entre plusieurs serveurs backend en utilisant différents algorithmes de load balancing, tels que le *round-robin*, le *least-conn*, ou le source. Cela permet de gérer efficacement les requêtes entrantes et d'optimiser les performances du site Web ou du service en ligne. Haproxy peut également être configuré pour utiliser des protocoles de chiffrement pour sécuriser les communications entre les utilisateurs et les serveurs backend, ou pour masquer les adresses IP des serveurs backend pour cacher les détails de l'architecture du réseau.

Points Forts	Points Faibles
• Haproxy est très performant et scalable, ce qui le rend idéal pour gérer les requêtes entrantes de sites Web ou de services en ligne à fort trafic. • Haproxy permet de répartir la charge entre plusieurs serveurs backend en utilisant différents algorithmes de load balancing, ce qui permet d'optimiser les performances du site Web ou du service en ligne. • Haproxy peut être configuré pour utiliser des protocoles de chiffrement pour sécuriser les communications entre les utilisateurs et les serveurs backend. • Haproxy peut être utilisé pour masquer les adresses IP des serveurs backend, ce qui peut être utile pour cacher les détails de l'architecture du réseau.	• Haproxy est un logiciel complexe qui nécessite une certaine connaissance technique pour être configuré et utilisé correctement. Cela peut être un obstacle pour les utilisateurs moins expérimentés ou pour les entreprises qui ne disposent pas d'équipes informatiques dédiées. • Haproxy n'offre pas de fonctionnalités de gestion des utilisateurs ou de contrôle d'accès. Si vous avez besoin de ces fonctionnalités, vous devrez utiliser un autre logiciel en plus d'HAProxy. • Haproxy n'est disponible que pour les systèmes d'exploitation Linux, Unix et BSD. Si vous utilisez un autre système d'exploitation, vous devrez utiliser un autre logiciel de reverse proxy.

- **HeartBeat**

Heartbeat est un logiciel libre et open-source qui permet de surveiller l'état des services et des machines dans un réseau. Il peut être utilisé pour détecter les pannes de service ou de matériel, et pour redémarrer automatiquement les services ou les machines en cas de problème.

Fonctionnement d'HeartBeat

Le fonctionnement d'Heartbeat consiste à envoyer des paquets de données à intervalles réguliers à différents services et machines d'un réseau, puis vérifie si ces services et machines répondent correctement aux paquets de données envoyés. Si un service ou une machine ne répond pas, Heartbeat peut considérer que cela constitue une panne et redémarrer le service ou la machine en question pour essayer de résoudre le problème.

Pour configurer Heartbeat, vous devez définir les services et les machines à surveiller, ainsi que les intervalles de temps à utiliser pour envoyer les paquets de données. Vous pouvez également définir des règles pour déterminer quand un service ou une machine est considéré comme en panne, et pour déterminer les actions à effectuer en cas de panne (par exemple, redémarrer le service ou la machine).

Une fois Heartbeat configuré, il fonctionne en arrière-plan et envoie des paquets de données aux services et machines d'un réseau à intervalles réguliers. Si un service ou une machine ne répond pas aux paquets de données envoyés, Heartbeat peut prendre des mesures pour essayer de résoudre le problème et maintenir le service en ligne.

Points Forts	Points Faibles
• Fiabilité : Le heartbeat permet de s'assurer que les différents éléments du système sont en bonne santé et fonctionnent correctement, ce qui contribue à maintenir la fiabilité du système. • Résilience : Si un élément du système cesse de répondre aux messages heartbeat, le système peut prendre des mesures pour remédier à la situation, ce qui lui permet de rester opérationnel même en cas de problème. • Scalabilité : Le heartbeat peut être utilisé dans des systèmes de grande taille avec de nombreux éléments, ce qui en fait un outil idéal pour les applications à fort trafic. • Flexibilité : Le heartbeat peut être utilisé pour vérifier l'état de différents types d'éléments, comme des serveurs, des applications ou des services, ce qui en fait un outil très flexible.	• Fausses alertes : Le heartbeat peut parfois générer des alertes pour des éléments qui ne sont pas réellement en panne. Cela peut être dû à des problèmes de connexion temporaires ou à des erreurs de configuration, et peut entraîner des mesures inutiles pour remédier à des problèmes qui n'existent pas. • Limitations : Le heartbeat ne permet pas de vérifier l'état de tous les éléments d'un système. Par exemple, il ne peut pas vérifier la qualité des données stockées ou le bon fonctionnement des algorithmes utilisés par le système. Cela peut entraîner des problèmes non détectés par le système de heartbeat. • Dépendance : Le fonctionnement du système peut être affecté si le heartbeat lui-même ne fonctionne pas correctement. Si le système de heartbeat est en panne, il ne pourra pas détecter les problèmes des autres éléments du système, ce qui peut entraîner des pannes ou des dysfonctionnements.

IV. Choix de la solution de type Reverse Proxy sous linux

Heartbeat et Haproxy sont deux outils différents qui peuvent être utilisés dans des contextes différents. Heartbeat est une fonctionnalité utilisée pour vérifier la communication et la performance des différents composants d'un système informatique. Haproxy, en revanche, est un équilibreur de charge et un serveur proxy qui peut être utilisé pour répartir les requêtes entrantes sur plusieurs serveurs de destination afin de gérer la charge et d'améliorer les performances d'un système. En d'autres termes, Heartbeat peut être utilisé pour vérifier que les différents composants d'un système fonctionnent correctement, tandis que HAProxy peut être utilisé pour gérer la façon dont les requêtes sont routées vers ces composants.

Voici un tableau comparant les deux solutions et ce qu'elles offrent :

Haproxy	Heartbeat
HAProxy est un équilibreur de charge qui offre plusieurs solutions pour améliorer les performances et la disponibilité d'un système informatique. Voici quelques exemples de solutions offertes par HAProxy : • Répartition des demandes entrantes sur plusieurs serveurs pour réduire la charge sur un seul serveur et améliorer les performances. • Gestion des connexions et des sessions pour éviter les surcharges et les temps d'attente pour les utilisateurs. • Monitoring des serveurs pour détecter les problèmes et transférer les demandes vers des serveurs de secours en cas de panne d'un serveur. • Filtrage et contrôle d'accès pour gérer les autorisations d'accès aux ressources et aux services en fonction des utilisateurs et des groupes. • Mise en cache des données pour accélérer les temps de réponse et réduire la charge sur les serveurs backend. • Mise en place de stratégies de répartition des demandes pour équilibrer la charge sur les serveurs en fonction de différents critères, tels que la charge, la latence, le nombre de connexions, etc. • Support de plusieurs protocoles de communication, tels que HTTP, HTTPS, TCP, et UDP, pour une compatibilité avec une large gamme d'applications et de services.	Le Heartbeat est une fonctionnalité utilisée pour maintenir une connexion active entre un client et un serveur. Cela permet au serveur de savoir que le client est toujours en ligne et de prévenir toute interruption de la connexion. • Il permet de maintenir une connexion active entre un client et un serveur, ce qui est particulièrement utile pour les applications qui nécessitent une connexion en temps réel. • Il peut être utilisé pour détecter les interruptions de connexion et rétablir la connexion si nécessaire, ce qui améliore la fiabilité des applications. • Il peut également être utilisé pour surveiller la qualité de la connexion et adapter le débit des données en conséquence, ce qui peut améliorer les performances des applications. • Le Heartbeat est généralement facile à implémenter et peut être utilisé avec un large éventail d'applications et de protocoles de communication.

Choix de l'outil de type Reverse Proxy

Heartbeat et Haproxy sont deux logiciels couramment utilisés pour surveiller la santé des serveurs et assurer la haute disponibilité des environnements de serveurs. Cependant, il existe quelques différences essentielles entre les deux.

Heartbeat est un programme open-source conçu pour surveiller la disponibilité des serveurs et pour assurer un basculement automatique en cas de défaillance d'un serveur. Il utilise un simple fichier de configuration et peut être facilement configuré pour surveiller la santé des serveurs et des services, et pour prendre les mesures appropriées si un problème est détecté.

Haproxy, quant à lui, est un équilibreur de charge et un serveur proxy plus complet. Il est capable de fournir un équilibrage de charge, une haute disponibilité et des capacités de proxy pour les applications basées sur TCP et HTTP. Il est souvent utilisé conjointement avec Heartbeat afin de fournir une solution complète pour assurer la haute disponibilité et l'équilibrage de charge dans les environnements de serveurs.

En termes de différences, l'une des principales différences entre les deux est que Heartbeat est principalement axé sur la surveillance de la disponibilité des serveurs, tandis que Haproxy est davantage axé sur la fourniture de capacités d'équilibrage de charge et de proxy. En outre, Heartbeat est généralement plus facile à installer et à configurer que Haproxy, qui peut être plus complexe en raison de ses fonctionnalités supplémentaires.

Dans l'ensemble, Heartbeat et Haproxy sont tous deux des outils utiles pour assurer la haute disponibilité et maintenir la santé des environnements de serveurs, mais ils servent des objectifs différents et peuvent être utilisés de différentes manières selon les besoins spécifiques d'une organisation.

Mais Il y a plusieurs raisons pour lesquelles nous préférons choisir Haproxy. Tout d'abord, Haproxy est connu pour être un logiciel très performant et fiable, offrant une latence et une disponibilité élevées. En outre, Haproxy est très configurable et peut être utilisé dans de nombreux environnements différents, ce qui en fait une solution très flexible. Enfin, Haproxy est également très populaire et bien documenté, ce qui signifie qu'il y a une grande communauté d'utilisateurs et de développeurs qui peuvent vous aider si vous rencontrez des problèmes ou si vous avez des questions. En résumé, les raisons pour lesquelles vous pourriez choisir Haproxy incluent sa performance, sa fiabilité, sa flexibilité, sa configurabilisé et sa communauté active.

V. Conclusion

D'après les différentes études qu'on a réalisé sur les deux outils, nous avons décidé de choisir la solution d'Haproxy car Haproxy est très configurable et fiable, Haproxy nous offre plus de fonctionnalités que HeartBeat. Haproxy nous offre des capacités d'équilibrage de charge et de proxy pour les applications basées sur TCP et http.