

Installer HeartBeat et le configurer

Mise en place d'une infrastructure haute disponibilité

1. Introduction à la haute disponibilité

La haute disponibilité, également appelée High Availability (HA), constitue un pilier fondamental dans l'administration des systèmes informatiques. Elle vise à garantir l'accès continu aux services et aux applications, même en cas de défaillance matérielle ou durant des interventions de maintenance. Pour y parvenir, on met en place divers dispositifs tels que la duplication des composants, le basculement automatique en cas de panne, et un système de surveillance anticipée pour prévenir les incidents.



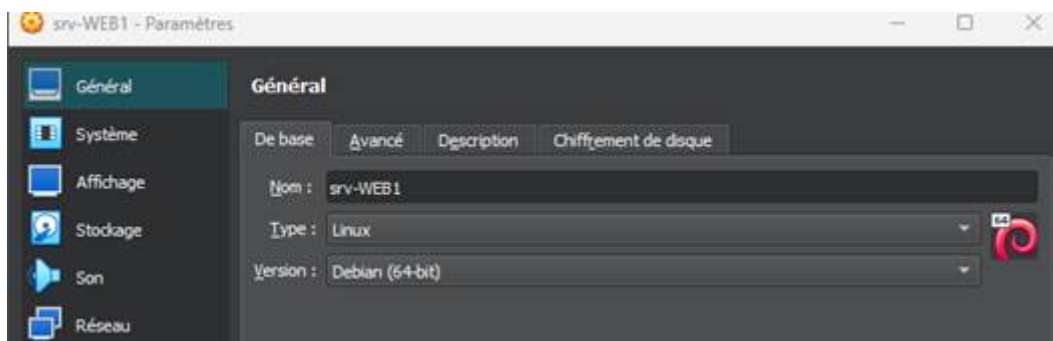
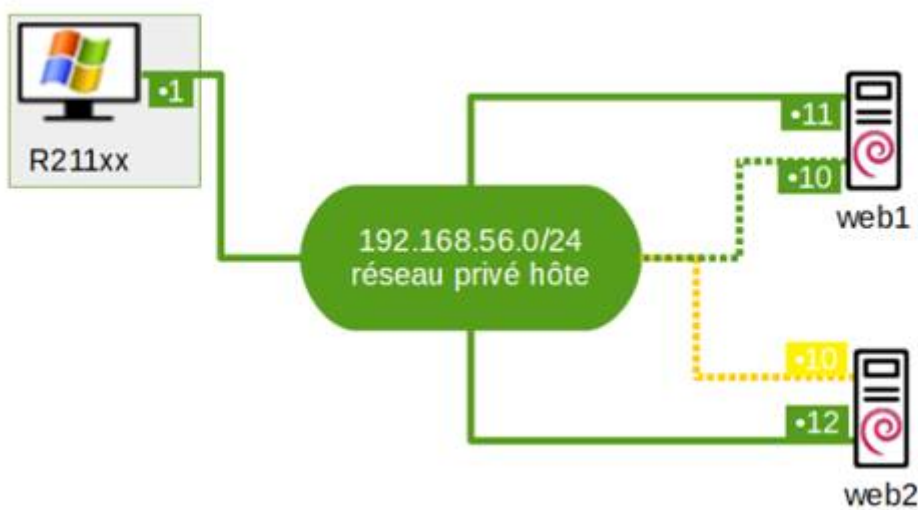
2. Heartbeat et la redondance de serveurs

Afin d'assurer la continuité du service, un outil de supervision appelé « heartbeat » est mis en place. Il contrôle en permanence l'état du serveur principal. Si celui-ci devient indisponible, un

serveur de secours prend automatiquement le relais. L'ensemble des serveurs, appelés « nœuds », forme un cluster. Grâce à une adresse IP virtuelle partagée, les utilisateurs peuvent accéder au service sans se préoccuper de quel nœud est actif à un instant donné.

3. Étapes de mise en œuvre

La première étape consiste à installer Apache2 sur chacun des deux serveurs web. Ensuite, il faut attribuer un nom d'hôte et une adresse IP spécifique à chaque machine. Pour les différencier visuellement, il est recommandé de personnaliser la page d'accueil par défaut d'Apache avec un nom unique à chaque serveur.



Poursuivez en installant le paquet **heartbeat** sur les deux serveurs. Le bon fonctionnement du système repose sur trois fichiers de configuration essentiels : `ha.cf`, `authkeys` et `haresources`. Il est impératif de sécuriser le fichier contenant la clé partagée à l'aide de la commande suivante :

```
chmod 600 /etc/ha.d/authkeys
```

4. Paramétrage détaillé

Le fichier `ha.cf` permet de configurer les paramètres de surveillance, notamment l'intervalle entre deux signaux (*keepalive*) et le temps d'attente avant de considérer un nœud comme inactif (*deadtime*). Il doit également contenir la liste complète des nœuds du cluster. De son côté, le fichier `haresources` définit les services à maintenir disponibles ainsi que l'adresse IP virtuelle utilisée pour les connexions clients.

 `/etc/ha.d/ha.cf`

 `/etc/ha.d/authkeys`

 `/etc/ha.d/haresources`

5. Test et validation

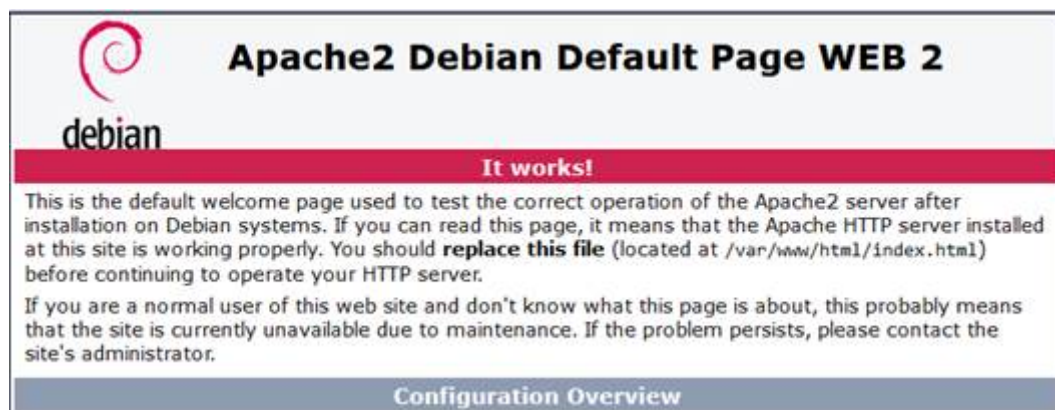
Il faut d'abord stopper le service Apache2 sur les deux serveurs, puis empêcher son démarrage automatique. Une fois cela fait, activez **heartbeat** afin qu'il prenne en charge la gestion du service. Vous pouvez ensuite vérifier le bon fonctionnement du mécanisme en simulant l'arrêt du serveur principal : le serveur secondaire doit automatiquement prendre le relais, assurant ainsi la continuité du service sans interruption visible.

 `/etc/ha.d/authkeys`

6. Notions de répartition de charge

Le **load balancing**, ou répartition de charge, est une technique qui permet de distribuer les requêtes entrantes entre plusieurs serveurs. Cela optimise les performances globales du système et renforce sa résilience face aux pannes. Cette gestion est assurée par un serveur spécifique, appelé **Load Balancer**, qui oriente le trafic selon des règles précises, comme l'algorithme de

rotation (round robin) ou une pondération définie.



Révision #2

Créé 30 avril 2025 13:30:43 par Julien

Mis à jour 30 avril 2025 13:36:40 par Julien