



## Référentiel

B2.2 - Installer, tester et déployer une solution d'infrastructure réseau.

Installer et configurer des éléments nécessaires pour assurer la qualité de service.

## Définition

(Source : <https://www.syloe.com>)

La **haute disponibilité** (ou High Availability ou HA) permet d'assurer et de garantir le bon fonctionnement des services ou applications proposées et ce 7j/7 et 24h/24. Cela consiste donc à mettre en place toutes les actions et dispositions techniques pour qu'une infrastructure informatique soit toujours disponible en appliquant certains principes tels que la réplication des données, la sauvegarde, la répartition de la charge, la redondance, etc. pour limiter l'indisponibilité d'un SI. Aujourd'hui, c'est un enjeu essentiel. Pour qu'une entreprise puisse se développer et fonctionner, un système d'information **disponible et fiable** est primordial. Auquel cas, il y a un risque de pertes de productivité (baisse du CA), de matériels, mais également de coûts supplémentaires (liées aux pannes, aux ressources à déployer, etc.)

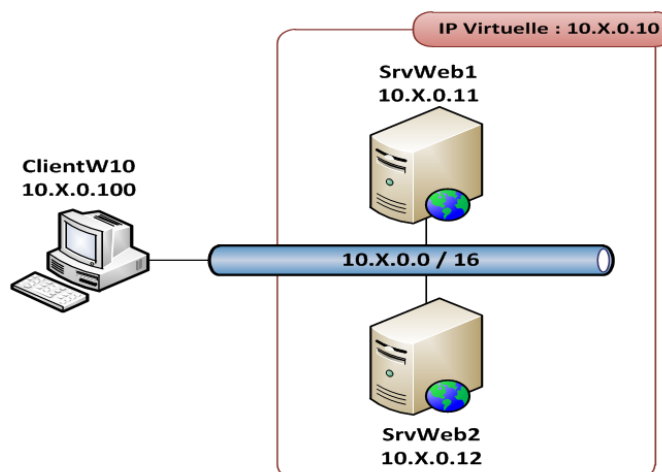
Le **Fail Over** (tolérance aux pannes) consiste à rediriger un utilisateur vers un autre serveur lorsque que le serveur principal tombe en panne laissant le temps aux administrateurs système de régler la panne. L'intérêt de ce type d'architecture et de pouvoir délivrer un service en continu en s'appuyant sur la redondance des matériels mis en place (duplication d'un composant ou matériel par des éléments identiques).

## Sommaire

La Tolérance de pannes	1
Schéma du réseau	1
Partie n°1 : Les deux serveurs Apache	1
Partie n°2 : Installation de Heartbeat	1
Partie n°3 : Les tests	2

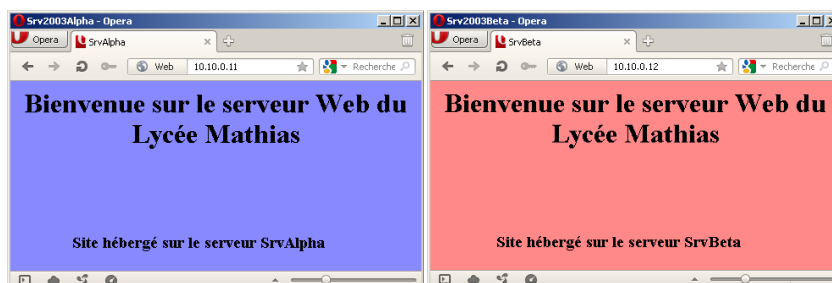
## La Tolérance de pannes

### Schéma du réseau



### Partie n°1 : Les deux serveurs Apache

Installez sur chaque serveur un serveur Apache. L'installation minimale est suffisante. Différenciez les 2 sites créés par 2 couleurs distinctes (bleu et rouge par exemple) :



### Partie n°2 : Installation de Heartbeat

Vous allez maintenant découvrir comment configurer **Heartbeat** pour faire fonctionner une solution avec une seule machine active.

Vous devrez avoir un droit d'administration pour effectuer toutes les commandes. Vous devrez vous connecter en **root** sur votre machine, ou utiliser le mécanisme de **sudo**.

### Règles de base

Voici les quelques points à respecter pour que tout se passe bien :

- Ne pas configurer l'IP virtuelle sur votre système Linux. C'est **Heartbeat** qui le fera.
- Les fichiers de paramètres doivent être **identiques** sur les deux serveurs pour fonctionner.

### Installation du service

Le paquet pour mettre en place la solution s'appelle **heartbeat**. Son installation se fait par la commande suivante :

**apt install heartbeat**

### Configuration du service

Pour le faire fonctionner, il faut créer 3 fichiers de configuration, dans le dossier **/etc/heartbeat**

#### **/etc/heartbeat/ha.cf**

Ce fichier détermine la liste des machines à utiliser et la manière de dialoguer entre elles.

```
logfile      /var/log/ha-log
logfacility  local0
keepalive    2
deadtime     10
bcast        ens33
node         SrvWeb1 SrvWeb2
auto_failback no
```

Fichier : **/etc/heartbeat/ha.cf**

#### **/etc/heartbeat/haresources**

Ce fichier indique les opérations à effectuer au démarrage de la haute disponibilité sur une machine. La syntaxe est la suivante :

```
SrvWeb1 IPaddr::10.X.0.10/16
SrvWeb1 apache2
```

Fichier : **/etc/heartbeat/haresources**

#### **/etc/heartbeat/authkeys**

Ce fichier détermine la clé et le protocole de protection utilisé. Voici un exemple du fichier :

```
auth 3
3 md5 mot_de_passe
```

Fichier : **/etc/heartbeat/authkeys**

Protégez ce fichier par la commande suivante : **chmod 600 /etc/heartbeat/authkeys**

### Démarrage du programme

Assurez-vous que les fichiers de configuration des deux serveurs sont identiques et que les services apache sont arrêtés par la commande : **service apache2 stop**

Il faut ensuite faire en sorte que les services gérés par Heartbeat ne soient plus lancés automatiquement au démarrage de Linux. Saisissez la commande suivante sur les deux serveurs : **update-rc.d -f apache2 remove**

Vous pouvez maintenant saisir (sur les 2 serveurs) : **service heartbeat start**

Au bout de quelques instants, les services se sont normalement lancés sur la première machine, pendant que l'autre est en attente. Pour voir si l'adresse **IP virtuelle** est active, tapez **ip a**

Sur la machine passive, vous n'aurez que vos interfaces habituelles. Sur celle active, une nouvelle interface est apparue, avec '0' à la fin. Vérifiez son bon fonctionnement avec un **ping**.

## Partie n°3 : Les tests

Depuis votre client, ouvrez votre navigateur, et saisissez l'adresse **IP virtuelle** du cluster (10.X.0.10).

Arrêtez ensuite **Heartbeat** sur le serveur maître par **service heartbeat stop** et actualisez le navigateur de votre client.

Relancez le service **Heartbeat** sur le serveur maître et actualisez à nouveau votre navigateur