

TP N°5 : Haute disponibilité et tolérance aux pannes d'un serveur web avec HEARTBEAT



Table des matières

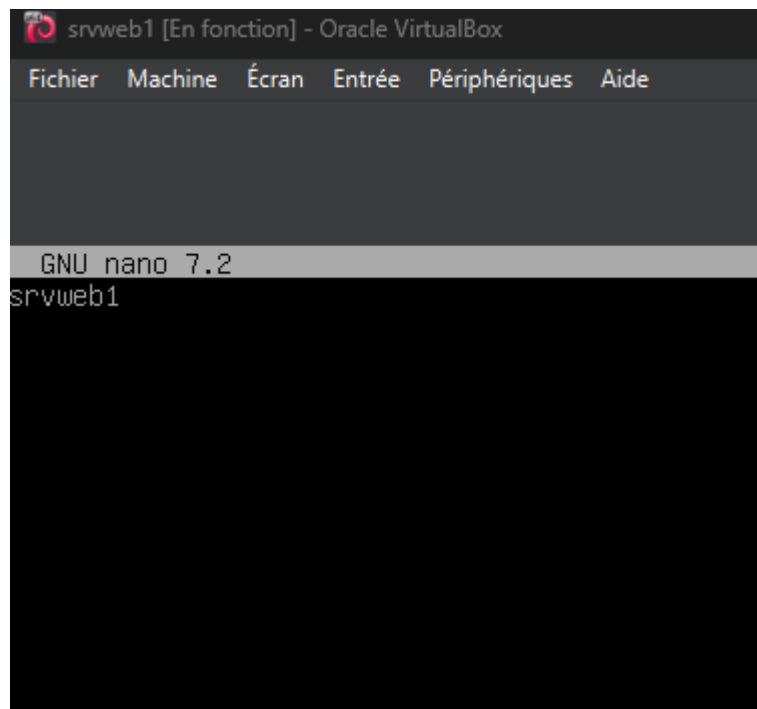
Etape 1 : Création d'un second serveur web nommé « SRVWEB2 ».....	2
Etape 2 : Mise à jour des fichiers de configuration d'Apache 2.0 des deux serveurs web.....	5
Etape 3 : Création et configuration du cluster de serveurs web avec Hearbeat.....	7
Étape 4 : Testez la haute disponibilité du site web de la ville des Abymes.....	11

Étape 1 : Création d'un second serveur web nommé « SRVWEB2 »

Modification de "l'hostname" du serveur web 1 :

```
sudo nano /etc/hostname
```

Puis modifier avec srvweb1 :

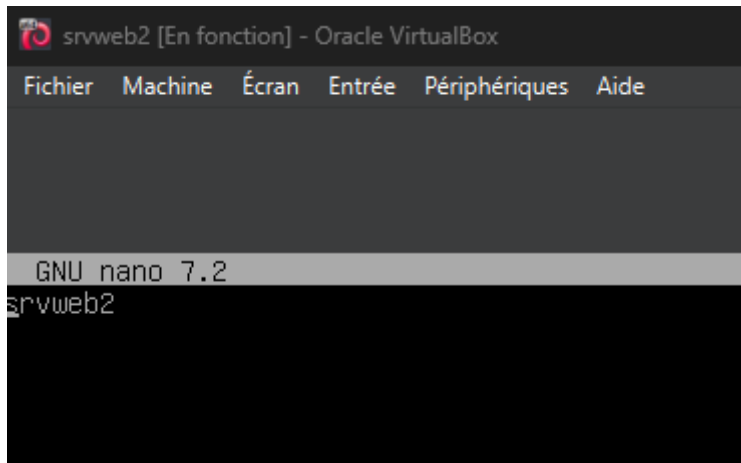


```
srvweb1 [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 7.2
srvweb1
```

Modification de "l'hostname" du serveur web 2 :

```
sudo nano /etc/hostname
```

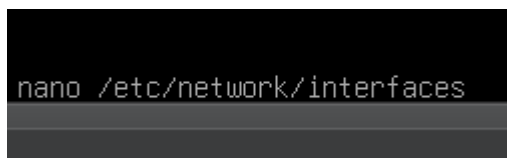
Puis modifier avec srvweb2 :



```
srvweb2 [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 7.2
srvweb2
```

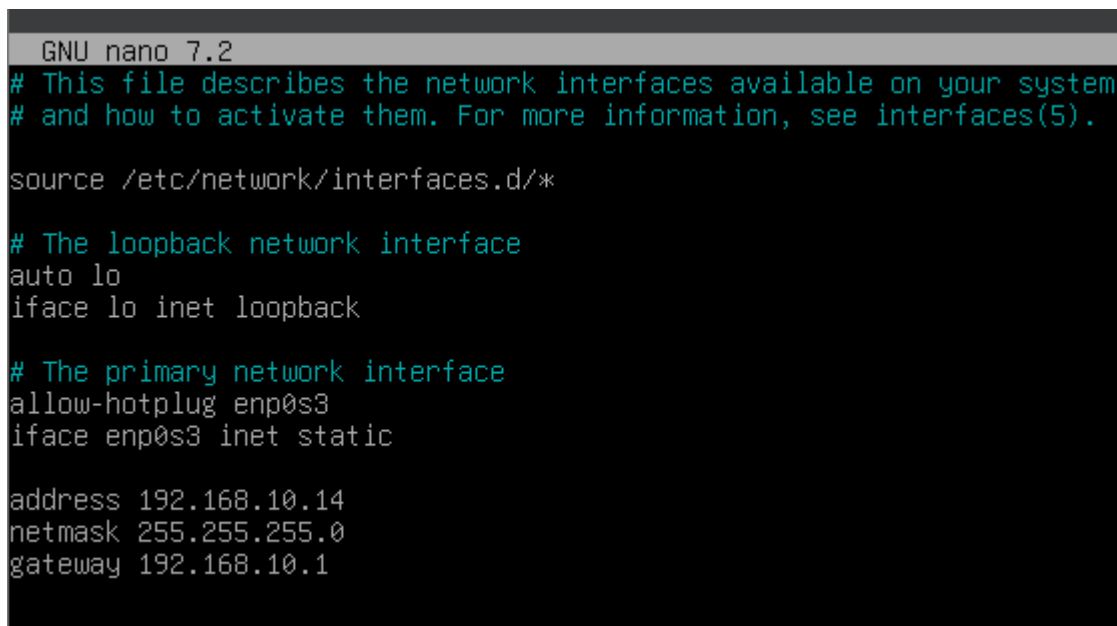
Modification du plan d'adressage IP du serveur web 1 :

Accéder au interfaces réseau grâce à la commande suivante :



```
nano /etc/network/interfaces
```

Puis ajouter ou modifier des lignes pour ce résultat :



```
GNU nano 7.2
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

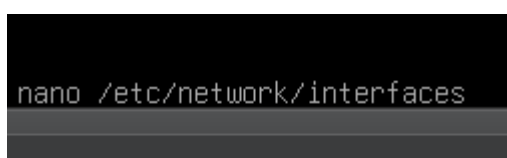
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static

address 192.168.10.14
netmask 255.255.255.0
gateway 192.168.10.1
```

Modification du plan d'adressage IP du serveur web 2 :

Accéder au interfaces réseau grâce à la commande suivante :



```
nano /etc/network/interfaces
```

Puis ajouter ou modifier des lignes pour ce résultat :

```
GNU nano 7.2
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static

address 192.168.10.15
netmask 255.255.255.0
gateway 192.168.10.1
```

Étape 2 : Mise à jour des fichiers de configuration d'Apache 2.0 des deux serveurs web

Modification apportée pour le site du srvweb1 :

```
GNU nano 7.2 /var/www/html/villeabymes/
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <header>
    <h1>Bienvenue sur le site n°1 de la_ville des abymes</h1>
  </header>
</body>
</html>
```

Modification apportée pour le site du srvweb2 :

```
GNU nano 7.2 /var/www/html/ville
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <header>
    <h1>Bienvenue sur le site n°2 de la ville des abymes</h1>
  </header>
</body>
</html>
```

Test d'accessibilité aux différents sites :

Pour tester la connectivité des différents sites il faut utiliser le navigateur d'un client et y taper les IP des serveurs pour accéder aux pages HTML.

Site SRVWEB1 :



Site SRVWEB2 :



Étape 3 : Création et configuration du cluster de serveurs web avec Hearbeat

Configuration de “/etc/hosts/” :

SRVWEB1 :

```
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    srvweb1
192.168.10.15 srvweb2

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

SRVWEB2 :

```
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    srvweb2
192.168.10.14 srvweb1

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Le fichier /etc/hosts permet la résolution des noms de domaine, en permettant de traduire un nom d'hôte en adresse IP sans passer par un serveur DNS.

Test de communication entre les serveurs à partir de leurs noms d'hôtes

SRVWEB1 :

```
root@srvweb1:/etc/ha.d# ping srvweb2
PING srvweb2 (192.168.10.15) 56(84) bytes of data.
64 bytes from srvweb2 (192.168.10.15): icmp_seq=1 ttl=64 time=0.472 ms
64 bytes from srvweb2 (192.168.10.15): icmp_seq=2 ttl=64 time=0.407 ms
64 bytes from srvweb2 (192.168.10.15): icmp_seq=3 ttl=64 time=0.637 ms
^C
--- srvweb2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.407/0.505/0.637/0.096 ms
root@srvweb1:/etc/ha.d#
```

Le ping à fonctionner.

SRVWEB2 :

```
root@srvweb2:/etc/ha.d# ping srvweb1
PING srvweb1 (192.168.10.14) 56(84) bytes of data.
64 bytes from srvweb1 (192.168.10.14): icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from srvweb1 (192.168.10.14): icmp_seq=2 ttl=64 time=0.395 ms
64 bytes from srvweb1 (192.168.10.14): icmp_seq=3 ttl=64 time=0.735 ms
^C
--- srvweb1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2217ms
rtt min/avg/max/mdev = 0.395/0.813/1.309/0.377 ms
root@srvweb2:/etc/ha.d#
```

Le ping à fonctionner.

Création du fichier ha.cf sur les srvweb1 et 2 grâce a la commande ci-dessous :

```
root@srvweb1:/etc/ha.d# nano ha.cf _
```

Contenu du fichier :

```
GNU nano 7.2
bcast enp0s3
warntime 4
deadtime 5
initdead 15
keepalive 2
auto_failback on
node SRVWEB1
node SRVWEB2
```

Indication des rôles des lignes de ce fichier :

bcast enp0sX

→ Définit la méthode de communication entre les nœuds du cluster.

wartime 4

→ Définit le temps (en secondes) avant qu'un avertissement ne soit généré si un nœud ne répond plus.

deadtime 5

→ Indique le temps (en secondes) après lequel un nœud est considéré comme "**mort**" s'il ne répond plus.

initdead 15

→ Spécifie le délai d'attente (en secondes) après le démarrage du cluster avant de considérer un nœud comme inactif. Cela permet d'éviter des bascules prématurées après un redémarrage.

keepalive 2

→ Définit l'intervalle (en secondes) entre deux messages Heartbeat envoyés aux autres nœuds.

auto_failback on

→ Active le **retour automatique** des services sur le nœud principal lorsqu'il redevient disponible après une panne.

node SRVWEB1

→ Déclare SRVWEB1 comme un nœud du cluster.

node SRVWEB2

→ Déclare SRVWEB2 comme un autre nœud du cluster.

Création du fichier authkeys sur les srvweb1 et 2 grâce à la commande ci-dessous :

```
root@srvweb2:/etc/ha.d# nano authkeys
```

Contenu du fichier :

```
GNU nano 7.2
auth 1
1 md5 mac1e
```

Indication des rôles des lignes de ce fichier :

auth 1 → Cette ligne définit un identifiant d'authentification (1), qui sera utilisé pour sécuriser les communications entre les nœuds du cluster.

1 → Identifiant de la clé d'authentification.

md5 → Indique que l'algorithme **MD5** est utilisé pour sécuriser les communications entre les nœuds du cluster.

mac1e → C'est la clé secrète partagée utilisée pour sécuriser les échanges.

Modification des droits d'accès du fichier authkeys :

```
root@srvweb1:/etc/ha.d# chmod 600 authkeys
root@srvweb1:/etc/ha.d#
```

Cette commande sert à modifier les **permissions** du fichier authkeys.

600 signifie que :

- **6 (lecture et écriture pour le propriétaire)**
- **0 (aucune permission pour le groupe)**
- **0 (aucune permission pour les autres utilisateurs)**

Changer les droits d'accès au fichier authkeys est une **mesure de sécurité** pour protéger les clés d'authentification utilisées par **Heartbeat**.

Création du fichier haresources sur les srvweb1 et 2 grâce a la commande ci-dessous :

```
root@srvweb1:/etc/ha.d# nano haresources _
```

Contenu du fichier :

```
SRVWEB1 192.168.10.4/24/enp0s3:0 apache2
```

Indication des rôles des lignes de ce fichier :

SRVWEB1

→ Nom du serveur (hôte) dans la configuration Heartbeat.

192.168.X.4/24/enp0sX

- **192.168.X.4/24** : Adresse IP du serveur avec son masque
- **enp0sX** : Interface réseau sur laquelle Heartbeat va gérer cette IP flottante.

:0

- Indique que l'IP flottante sera associée à l'alias de l'interface réseau principale (eth0:0 ou enp0sX:0).

apache2

- Indique que Heartbeat doit gérer le service **Apache2**.

Étape 4 : Testez la haute disponibilité du site web de la ville des Abymes

SRVWEB1 :

```
srvweb1 [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

root@srvweb1:/etc/ha.d# ping 192.168.10.15
PING 192.168.10.15 (192.168.10.15) 56(84) bytes of data.
64 bytes from 192.168.10.15: icmp_seq=1 ttl=64 time=0.399 ms
64 bytes from 192.168.10.15: icmp_seq=2 ttl=64 time=0.382 ms
64 bytes from 192.168.10.15: icmp_seq=3 ttl=64 time=0.502 ms
^C
--- 192.168.10.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2134ms
rtt min/avg/max/mdev = 0.382/0.427/0.502/0.053 ms
root@srvweb1:/etc/ha.d# ping 192.168.10.4
PING 192.168.10.4 (192.168.10.4) 56(84) bytes of data.
64 bytes from 192.168.10.4: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 192.168.10.4: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 192.168.10.4: icmp_seq=3 ttl=64 time=0.061 ms
^C
--- 192.168.10.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.041/0.050/0.061/0.008 ms
root@srvweb1:/etc/ha.d# _
```

SRVWEB 2 :

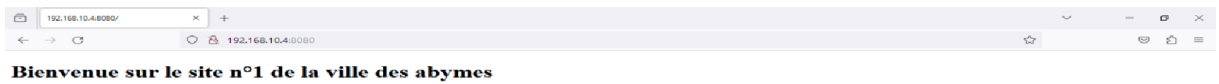
```
srvweb2 [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

root@srvweb2:/etc/ha.d# ping 192.168.10.14
PING 192.168.10.14 (192.168.10.14) 56(84) bytes of data.
64 bytes from 192.168.10.14: icmp_seq=1 ttl=64 time=0.398 ms
64 bytes from 192.168.10.14: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 192.168.10.14: icmp_seq=3 ttl=64 time=0.608 ms
^C
--- 192.168.10.14 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.398/0.493/0.608/0.086 ms
root@srvweb2:/etc/ha.d# ping 192.168.10.4
PING 192.168.10.4 (192.168.10.4) 56(84) bytes of data.
64 bytes from 192.168.10.4: icmp_seq=1 ttl=64 time=0.426 ms
64 bytes from 192.168.10.4: icmp_seq=2 ttl=64 time=0.479 ms
64 bytes from 192.168.10.4: icmp_seq=3 ttl=64 time=0.933 ms
^C
--- 192.168.10.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
rtt min/avg/max/mdev = 0.426/0.612/0.933/0.227 ms
root@srvweb2:/etc/ha.d# _
```

Tous les pings ont été un succès, on peut en déduire que l'adresse IP virtuelle qui crée une liaison entre les deux serveurs est en état de marche.

Test de la haute disponibilité :

Pour tester la haute disponibilité nous allons accéder à la page html via le client avec l'IP virtuelle comme ci-dessous



Puis maintenant on éteint le SRVWEB1 et on rafraichit la page on a le résultat ci-dessous :



La page affiche le contenu HTML du SRVWEB2 cela affirme le bon fonctionnement de la haute disponibilité.