

Réponse à l'appel d'offre Serveur web en haute disponibilité

Introduction :

La « haute disponibilité » (en anglais « high availability ») regroupe de nombreuses techniques et processus permettant de garantir un certain pourcentage de disponibilité d'une ressource, d'un service...

Par exemple, un taux de 99 % de disponibilité assure une disponibilité d'environ 361 jours sur 364 alors qu'un taux de 99,5 % assure une disponibilité de plus de 363 jours sur 365. La réalité économique fait que les organisations tendent de plus en plus vers des taux encore plus grands comme 99,9 % ou 99,99 % notamment sur certains services critiques. En effet, les conséquences d'une interruption de service sont innombrables et peuvent coûter très cher à tous points de vue.

Par exemple, on pouvait lire qu'une interruption de service de 40 mn aurait fait perdre à Amazon près de 5 millions de dollars.

(<http://www.zdnet.fr/actualites/comme-google-amazon-a-subie-une-panne-informatique-39793254.htm>).

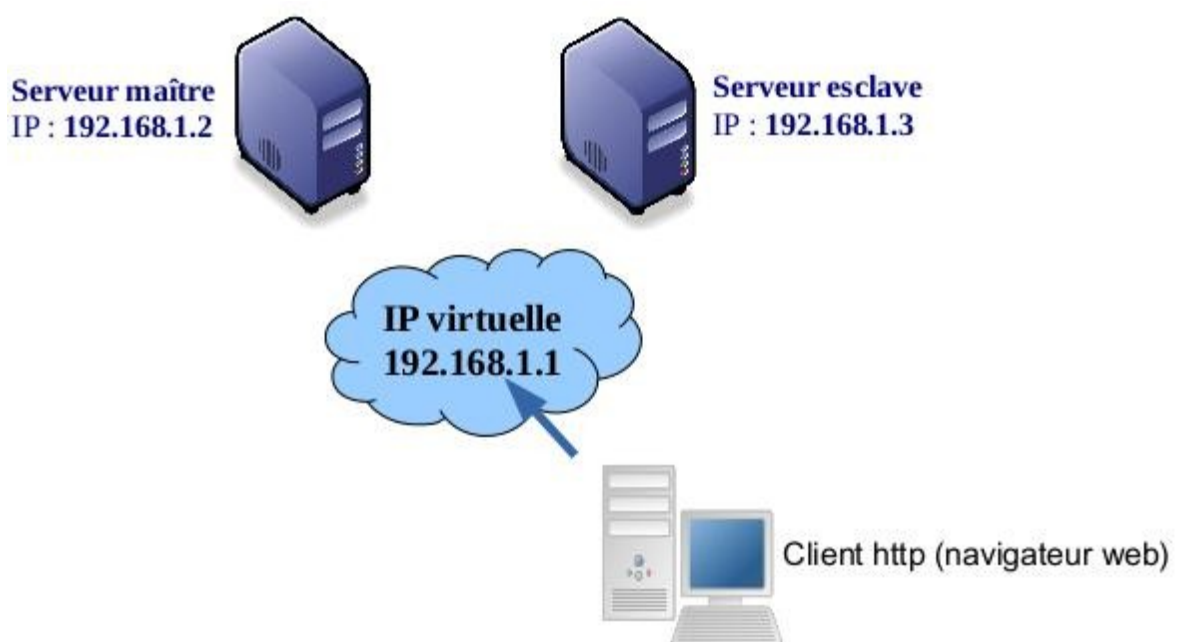
Pour améliorer la haute disponibilité, il existe de nombreuses techniques possibles.

Après analyse du Cahier des charges de l'appel d'offre, choix s'est porté sur la solution UCARP.

Ucarp est un outil de « haute disponibilité » permettant l'usage d'une adresse IP virtuelle communes à plusieurs hôtes redondants dans un réseau TCP/IP.

Tous les serveurs UCARP (à partir de 2) utiliseront la même adresse IP virtuelle pour remplacer le serveur défaillant ou en maintenance.

La maquette :



Installation et configuration ucarp sur les deux serveurs web

Pour améliorer la haute disponibilité, il existe de nombreuses configurations possibles. Dans une configuration très simple que nous allons découvrir, la haute disponibilité nécessite la présence d'un serveur secondaire (backup), fonctionnant sous le même système d'exploitation et fournissant un accès aux services que l'on souhaite rendre « hautement » disponibles.

Ce second **serveur backup configuré à l'identique**, surveillera le premier en permanence.

En cas de panne du serveur maître master , il la détectera et prendra la relève, devenant alors le nouveau serveur actif.

Exécutez les commandes suivantes dans un shell (sur les deux serveurs web):

```
$ sudo apt install ucarp apache2 net-tools
```

```
$ sudo mkdir -p /etc/ucarp/interface
```

```
$ sudo nano /etc/ucarp/interface/ucarp0
```

#Debut

```
UCARP_PIDFILE=/var/run/ucarp0.pid
```

```
UCARP_BASE=2 #UCARP_BASE=1 pour le serveur secondaire
```

```
INTERFACE=ens33:carp0 # remplacer eth0 par le nom de votre interface réseau
```

```
INSTANCE_ID=1
```

```
PASSWORD=bobo
```

```
VIRTUAL_ADDRESS=192.168.1.1
```

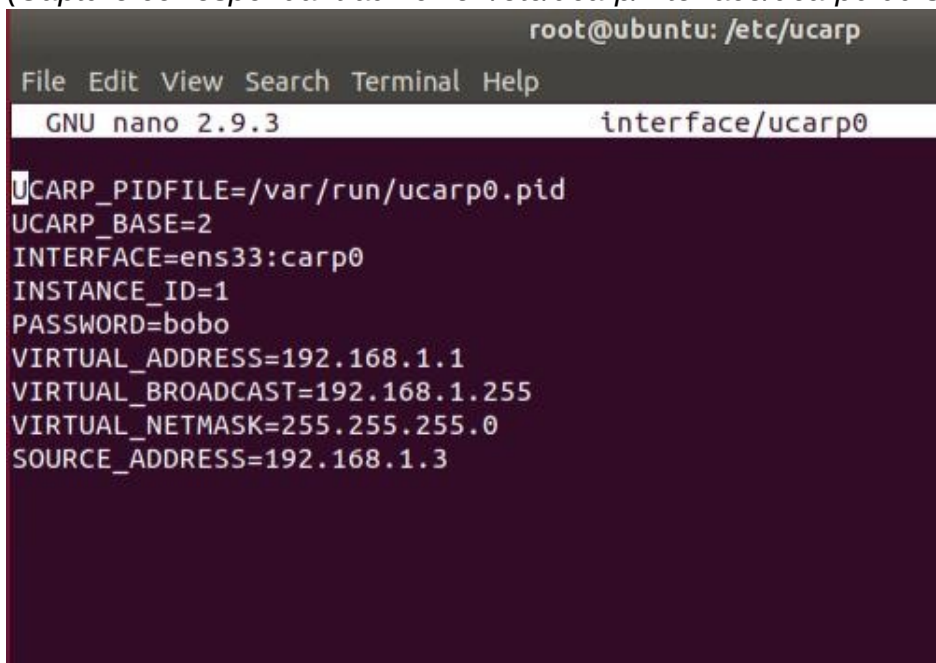
```
VIRTUAL_BROADCAST=192.168.1.255
```

```
VIRTUAL_NETMASK=255.255.255.0
```

```
SOURCE_ADDRESS=192.168.1.2 #Fixer SOURCE_ADDRESS=192.168.1.3 sur le serveur backup
```

#Fin (sauvegarder & exit)

(Capture correspondant au fichier /etc/ucarp/interface/ucarp0 du serveur backup.)



```
root@ubuntu: /etc/ucarp
File Edit View Search Terminal Help
GNU nano 2.9.3 interface/ucarp0
UCARP_PIDFILE=/var/run/ucarp0.pid
UCARP_BASE=2
INTERFACE=ens33:carp0
INSTANCE_ID=1
PASSWORD=bobo
VIRTUAL_ADDRESS=192.168.1.1
VIRTUAL_BROADCAST=192.168.1.255
VIRTUAL_NETMASK=255.255.255.0
SOURCE_ADDRESS=192.168.1.3
```

Explication des paramètres :

UCARP_PIDFILE=/var/run/ucarp0.pid

Description : Emplacement du fichier PID (Process ID) de UCARP.

Rôle : Sert à stocker le numéro du processus UCARP en cours d'exécution, pour pouvoir le gérer (arrêter, redémarrer, vérifier l'état, etc.).

Exemple : /var/run/ucarp0.pid est souvent utilisé pour l'instance associée à eth0 :carp0.

UCARP_BASE=1

Description : Définit la **priorité de base** du nœud dans le cluster CARP.

Rôle :

Le serveur avec la **plus grande valeur** de base (ex. 10) a **plus de priorité** et devient **le maître** (Master).

Le ou les autres serveurs avec une base plus faible seront **en secours** (Backup).

Exemple typique :

Serveur principal (master): UCARP_BASE=1

Serveur secondaire (backup): UCARP_BASE=2

INTERFACE=eth0:carp0

Description : Nom de l'interface réseau virtuelle utilisée par UCARP.

Rôle : UCARP crée une interface virtuelle (ici eth0:carp0) liée à l'interface physique (eth0), sur laquelle l'adresse IP virtuelle sera attachée.

Remarque : Le suffixe :carp0 est arbitraire, tu pourrais aussi avoir eth1:vip ou autre.

INSTANCE_ID=1

Description : Identifiant unique du groupe CARP.

Rôle :

Permet de distinguer plusieurs groupes CARP sur le même réseau.

Tous les serveurs partageant l'**adresse virtuelle vip** doivent avoir **le même** INSTANCE_ID.

Valeur : nombre entre 1 et 255 doit être identique sur tous les nœuds du même groupe).

PASSWORD=bobo

Description : Mot de passe partagé entre tous les nœuds du groupe CARP.

Rôle : Sert à authentifier les paquets CARP échangés entre les nœuds, pour éviter qu'un serveur non autorisé se fasse passer pour un membre du groupe.

VIRTUAL_ADDRESS=192.168.1.1

Description : Adresse IP **virtuelle partagée** entre les serveurs du groupe CARP.

Rôle :

C'est cette adresse que les clients utilisent pour joindre le service (ex : base de données, site web, etc.).

Elle « flotte » automatiquement entre les serveurs selon qui est maître.

VIRTUAL_BROADCAST=192.168.1.255

Description : Adresse de diffusion (broadcast) du réseau contenant l'IP virtuelle.

Rôle : Permet à UCARP de diffuser certains paquets réseau pour annoncer l'état du nœud.

VIRTUAL_NETMASK=255.255.255.0

Description : Masque de sous-réseau associé à l'adresse virtuelle.

Rôle : Nécessaire pour configurer correctement l'interface virtuelle et la communication réseau.

SOURCE_ADDRESS=192.168.36.2

Description : Adresse IP **réelle** du serveur local, sur son interface physique.

Rôle :

Sert à identifier le serveur dans le groupe CARP (chaque nœud a une adresse réelle différente).

C'est l'adresse d'où partent les paquets CARP et les annonces de basculement.

Créez et Éditez le fichier /etc/default/ucarp comme ceci :

```
$sudo nano /etc/default/ucarp
```

```
#Debut
```

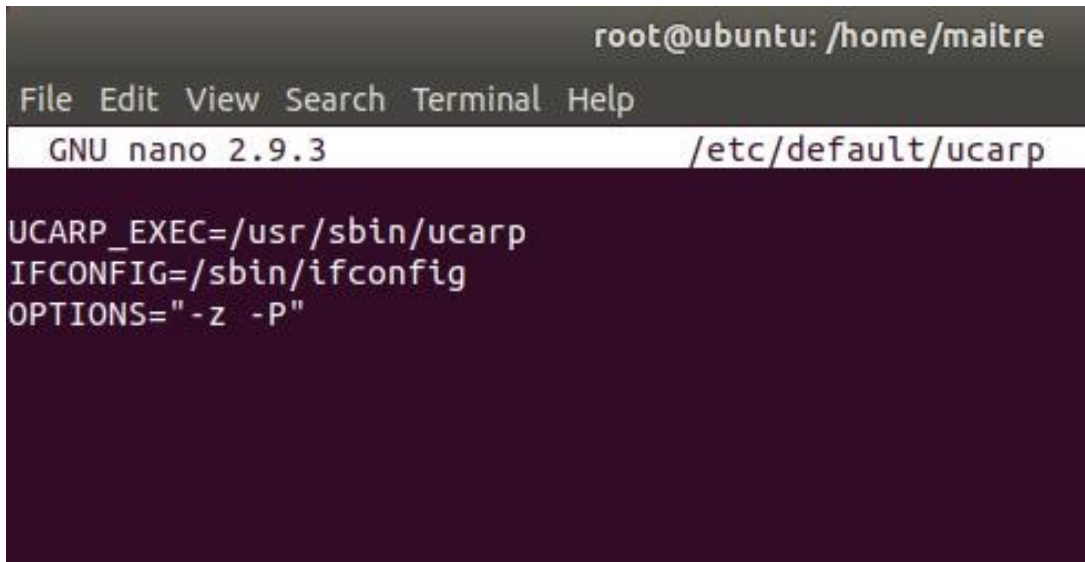
```
UCARP_EXEC=/usr/sbin/ucarp
```

```
IFCONFIG=/sbin/ifconfig
```

```
OPTIONS="-z -P"
```

#Fin (sauvegarder & exit)

(Cette capture correspond à celle du serveur master.)



```
root@ubuntu: /home/maitre
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/default/ucarp
UCARP_EXEC=/usr/sbin/ucarp
IFCONFIG=/sbin/ifconfig
OPTIONS="-z -P"
```

Explication des paramètres :

UCARP_EXEC=/usr/sbin/ucarp

Spécifie le **chemin absolu vers le binaire UCARP**.

C'est le programme qui sera exécuté pour lancer le service CARP (géré souvent par un script d'init ou un service systemd).

IFCONFIG=/sbin/ifconfig

Définit le chemin vers la commande **ifconfig**, utilisée par UCARP pour **monter ou démonter l'adresse IP virtuelle** sur l'interface réseau.

OPTIONS="-z -P"

-z

Signification : Lance UCARP en **mode daemon**, c'est-à-dire **en arrière-plan**.

Effet :

Le processus se détache du terminal.

UCARP continue de tourner comme un service, même après la fermeture du shell.

-P

Signification : Cette option est un **alias pour “préfixe de priorité” ou “priority”** dans certains scripts UCARP (selon la version / distribution).

Rôle général : Indique que UCARP doit **prendre en compte la priorité (UCARP_BASE)** du nœud pour déterminer qui devient maître.

Détail :

Si -P est présent, UCARP utilise la valeur définie par UCARP_BASE pour comparer les nœuds.

Sans -P, certains scripts peuvent ignorer cette priorité et basculer sur la première instance détectée.

Téléchargez le script `ucarp.sh` à l'adresse suivante :

```
$cd ; wget http://192.168.90.22/ucarp.sh
```

Déplacez le script `ucarp.sh` dans le répertoire `/etc/init.d`

```
$sudo mv /home/<user>/ucarp.sh /etc/init.d
```

Ajoutez les droits (d'exécution)d'accès à `ucarp.sh`

```
$sudo chmod 744 ucarp.sh
```

Lancez le service UCARP (sur les deux serveur)

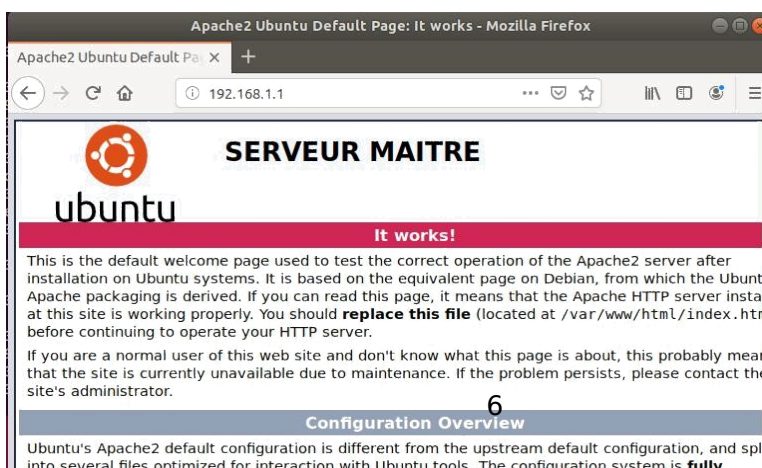
```
$sudo /etc/init.d/ucarp.sh start
```

Recette :

Testez l'accès au service Web à partir d'un client (Firefox) à l'adresse IP virtuelle 192.168.1.1

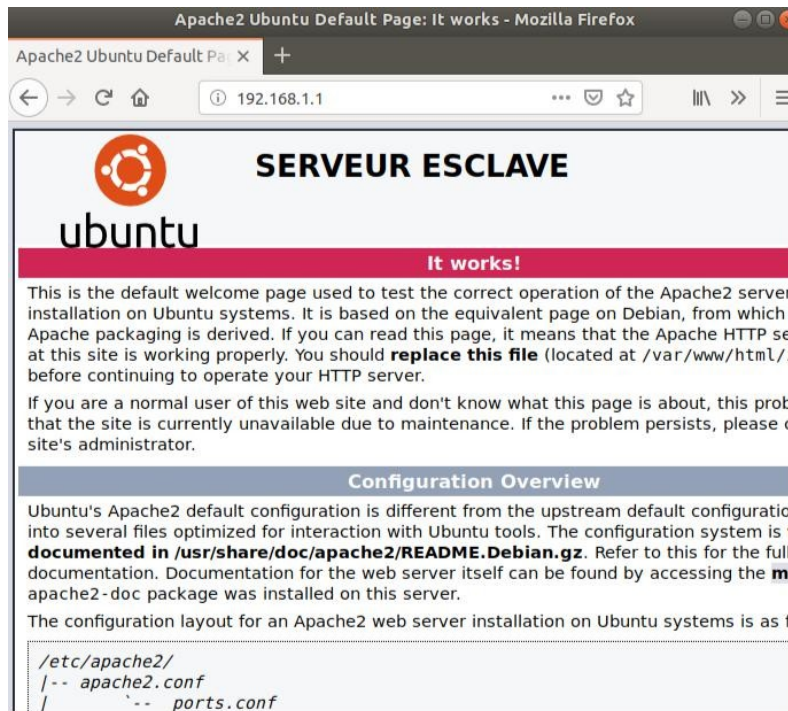
(astuce : modifiez le fichier `/var/www/html/index.html` des 2 serveurs pour faciliter l'identification du serveur qui répond à la requête du client)

(Cette capture correspond à la page web (`index.html`) du serveur master.



Déconnectez ou arrêtez le serveur master.
On refait la même requête sur le client web pour tester le basculement (failover IP) vers le serveur backup

(Cette capture correspond à la page web (index.html) du serveur slave)



Observez le basculement en éditant en temps réel le fichier log:

```
$sudo tail -f /var/log/syslog
```

Lors du redémarrage de ucarp sur le serveur master (capture ci-dessous) on peut confirmer le passage (switching) de l'état **BACKUP** à **MASTER** de celui-ci .

```
Jan  8 03:15:06 ubuntu ucarp[9974]: [WARNING] Switching to state: BACKUP
Jan  8 03:15:09 ubuntu ucarp[9974]: [WARNING] Switching to state: MASTER
```