

# Haute disponibilité MySQL avec UCARP

## Présentation

Ressources utilisées :

- <http://www.unix-experience.fr/2013/ucarp/>

Vous avez configuré une **réplication de données** avec **deux** serveurs MySQL.

Maintenant vous allez mettre en oeuvre une solution de **haute disponibilité** du service de base de données MySQL en configurant un **IP failover**. Le service UCARP va permettre, en cas de dysfonctionnement d'un serveur MySQL Maître, au serveur MySQL répliquat de prendre le relais jusqu'à ce que le maître revienne à la normale.

En préalable, vous devez configurer une réplication **maître ↔ esclave** et non seulement **maître → esclave** sinon vous risquez de perdre des données qui se retrouveraient uniquement sur le répliquat lors de la remise en service du serveur maître.

Le principe d'UCARP est le **partage d'une adresse IP et adresse MAC** sur le réseau.

## Fonctionnement du protocole

Le **protocole (U)CARP** utilise une surcouche sur la configuration des cartes réseau. Il doit être **configuré dans chaque répliquat** associé à un service donné en l'occurrence MySQL.

Toutes les x secondes une trame multicast est envoyée afin d'avertir que le protocole VRRP/UCARP est utilisé sur tel ID avec telle priorité. cette priorité définit sur un degré de 1 à 255 quel sera le serveur qui répondra avant les autres et sera le répliquat du moment.

Lorsque le serveur maître ou un serveur de plus haute priorité est de retour, le serveur rend alors l'adresse IP à celui-ci, permettant ainsi de garantir la bonne gestion du service.

## Interface réseau virtuelle

Le **partage d'une même adresse IP et d'une même adresse MAC** entre les deux serveurs consiste à utiliser une adresse **IP virtuelle** et donc **identique** sur chaque serveur.

Voici les adresses IP à utiliser :

Groupe	Adresse IP virtuelle	Etudiants
01	10.xxx.xxx.62	
02	10.xxx.xxx.65	
03	10.xxx.xxx.68	
04	10.xxx.xxx.71	
05	10.xxx.xxx.74	
06	10.xxx.xxx.77	
07	10.xxx.xxx.80	
08	10.xxx.xxx.83	
09	10.xxx.xxx.86	
10	10.xxx.xxx.89	
11	10.xxx.xxx.92	
12	10.xxx.xxx.95	
13	10.xxx.xxx.98	

## Configuration des interfaces réseau du serveur Maître

- Installer **ucarp** `# apt install ucarp`
- Modifiez le fichier **/etc/network/interfaces** `# nano /etc/network/interfaces`

Dans le fichier interfaces suivant :

- **eth0** a l'adresse que vous avez définie pour votre serveur MySQL (10.xxx.xxx.mmm)
- **eth0:ucarp** est l'interface IP virtuelle (en exemple 10.xxx.xxx.vvv)

```
iface eth0 inet static
```

```

        address 10.xxx.xxx.mmm
        netmask 255.255.255.0
        gateway 10.xxx.xxx.254
#####
##### UCARP Configuration #####
ucarp-vid      1
ucarp-vip      10.xxx.xxx.vvv
ucarp-password btssio
ucarp-advskew  1
ucarp-advbase  1
ucarp-master   yes

```

```

# Interface Carp en tant qu'alias de eth0
iface eth0:ucarp inet static
    address 10.xxx.xxx.vvv
    netmask 255.255.255.0

```

- **Relancer** la configuration des interfaces réseaux

```
# systemctl restart networking
```

- le serveur a maintenant, en plus de l'interface réseau **eth0**, une interface virtuelle **eth0:ucarp**
  - un **ping** depuis le serveur **Esclave** sur l'adresse **10.xxx.xxx.vvv** montre bien que le serveur **Maître** répond avec cette adresse comme l'indique la consultation de la **table ARP** qui associe l'**adresse IP 10.xxx.xxx.vvv** à l'**adresse MAC** du serveur Maître.

## Configuration des interfaces réseau du serveur Esclave

- Installer **ucarp** `# apt install ucarp`
- Modifiez le fichier **/etc/network/interfaces** `# nano /etc/network/interfaces`

Dans le fichier interfaces suivant :

- **eth0** a l'adresse attribuée par le serveur DHCP du réseau (10.xxx.xxx.eee)
- **eth0:ucarp** est l'interface IP virtuelle (en exemple 10.xxx.xxx.vvv)

```

auto eth0
iface eth0 inet static
    address 10.xxx.xxx.eee
    netmask 255.255.255.0
    gateway 10.xxx.xxx.254
#####
##### UCARP Configuration #####
ucarp-vid      1
ucarp-vip      10.xxx.xxx.vvv
ucarp-password btssio
ucarp-advskew  1
ucarp-advbase  2
ucarp-master   no

```

```

# Interface Carp en tant qu'alias de eth0
iface eth0:ucarp inet static
    address 10.xxx.xxx.vvv
    netmask 255.255.255.0

```

- Relancer la configuration des interfaces réseaux

```
# service networking restart
```

- Le serveur **Esclave** n'a pas out l'instant son interface virtuelle active

## Quelques précisions

- **vid**

L'identifiant Virtual Host ID. C'est un numéro unique utilisé pour identifier le groupe de redondance parmi les autres groupes, et pour distinguer les différents groupes sur un même réseau. Les valeurs acceptables vont de 1 à 255. Elles doivent être identiques sur chacun des membres du groupe.

- **vip** Virtual IP est l'adresse virtuelle que ce partagerons le maître et l'esclave.
- **password** Le mot de passe d'authentification à utiliser lors de la communication avec d'autres hôtes CARP dans le même groupe de redondance. Ce mot de passe doit être partagé entre tous les membres du groupe.
- **advbase** Ce paramètre optionnel spécifie le nombre de secondes qui s'écoule entre chaque annonce CARP. La valeur par défaut est 1 seconde. Les valeurs acceptables sont de 1 à 255.
- **advskew** Ce paramètre optionnel spécifie le biais à introduire au niveau de advbase lors de l'envoi d'annonces CARP. En manipulant advskew, l'hôte maître CARP peut être choisi. Plus grand est ce nombre, moindres sont les chances pour que l'hôte soit retenu lorsqu'un maître est choisi. La valeur par défaut est 0. Les valeurs acceptables sont de 1 à 254.

## Simulation de failover

### Test simple du failover

- **Ouvrez** une session avec **Putty** sur l'**IP virtuelle**,
- tapez la commande **hostname** qui doit vous retourner le **nom du serveur maître**,
- **Arrêter** le serveur maître,
- Retaper la commande **hostname** qui maintenant retourner le **nom du serveur esclave**,
- en **relançant** le serveur maître la commande **hostname** doit vous retourner à nouveau son nom.

### Visualisation des échanges sur le réseau

Pour **visualisez les échanges** entre serveurs, vous allez capturer les trames qui circulent sur le réseau avec l'outil **tcpdump**

### Installer tcpdump

```
# apt-get install tcpdump
```

### Informations pour exemple de simulation

- serveur **Maître** : 10.xxx.xxx.195
- serveur **Esclave** : 10.xxx.xxx.196
- adresse **IP virtuelle** : 10.xxx.xxx.130

### Capture de trame sur le serveur Maître

```
# tcpdump -i eth0 vrrp
```

Une trame de ce type sera envoyée toutes les 1 seconde (champ intvl) avec la priorité 1 (priorité maximale) et le vrid 1 (correspondant à l'ID de failover, attention il doit être identique sur le maître et l'esclave).

Tout se passe bien la machine maître est en ligne.

### Capture de trame sur le serveur Esclave

- Lancez une capture de trame sur le **serveur Esclave**

```
# tcpdump -i eth0 vrrp
```

Vous visualisez les **mêmes trames**.

- Arrêtez ou **Suspendez** le serveur Maître. Vous verrez que le serveur Esclave prend le relais.

Le **serveur Maître** a émis une dernière trame à 13h10 et 37 secondes (adresse IP 10.xxx.xxx.195).

Le **serveur Esclave** a pris le relais à 13h10 et 41 secondes (adresse IP 10.xxx.xxx.196).

En regardant les interfaces du **serveur Esclave**, on peut constater que l'**interface virtuelle est active**.

## Retour au dossier sur la réplication des données ...

[Fiche Cours \(C6\) : La réplication des données](#)

From:

/ - **Les cours du BTS SIO**

Permanent link:

[/doku.php/si7/a10ucarp](#)

Last update: **2021/03/18 15:41**

