

Rapport BTS SIO — Mise en place d'une Haute Disponibilité (UCARP)

Contexte

Dans le cadre du BTS SIO option SISR, nous devons concevoir, déployer et documenter une solution de Haute Disponibilité (HA) pour un service Web. L'objectif est d'assurer la continuité du service Apache via une adresse IP virtuelle flottante, gérée par UCARP, en configuration actif/passif sur deux machines.

Membres du binôme

Étudiant 1 : ...

Étudiant 2 : ...

Encadrant / tuteur : ...

Machines (VirtualBox, mode pont) :

- node1 (serveur maître) : 192.168.81.41
- node2 (serveur backup) : 192.168.81.42
- client : 192.168.81.43
- IP virtuelle choisie : 192.168.81.100

Objectifs pédagogiques

- Comprendre et configurer la redondance réseau et la tolérance de panne.
- Installer et configurer des services (Apache, UCARP) sur Ubuntu.
- Rédiger un rapport technique et fournir des scripts reproductibles.

Prérequis

Machines Ubuntu à jour, accès SSH entre les machines, droits sudo. Vérifier que l'IP virtuelle (192.168.81.100) n'est pas utilisée sur le réseau.

Résumé de la solution

La solution utilise UCARP pour fournir une IP virtuelle. Lorsque le nœud maître possède la VIP, il répond aux requêtes HTTP via Apache. Si le maître échoue, le nœud backup prend la VIP et démarre Apache. Les scripts up/down gèrent l'ajout/retrait de l'adresse et le démarrage/arrêt d'Apache.

Procédure détaillée (commandes exécutables)

Sur les deux serveurs (node1 et node2) — mise à jour et installation de base :

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y openssh-server apache2 ucarp
sudo systemctl enable ssh
sudo systemctl enable apache2
```

Configurer une IP statique via netplan (exemple pour node1)

`sudo nano /etc/netplan/00-installer-config.yaml`

Contenu (node1 - 192.168.81.41) :

```
network:
version: 2
ethernets:
  enp0s3:
    dhcp4: false
    addresses: [192.168.81.41/24]
    gateway4: 192.168.81.1
    nameservers:
      addresses: [8.8.8.8,1.1.1.1]
```

Appliquer : `sudo netplan apply`

Créer des pages distinctes pour test Apache

```
node1 : echo "Page du SERVEUR MASTER (node1)" | sudo tee /var/www/html/index.html
node2 : echo "Page du SERVEUR BACKUP (node2)" | sudo tee /var/www/html/index.html
sudo systemctl restart apache2
```

Configuration UCARP — fichiers et scripts

Nous fournissons ci-dessous les fichiers prêts à l'emploi. Copier les fichiers sur chaque machine en adaptant l'adresse serveur (SERVER_IP) pour node1/node2 lorsque nécessaire.

Fichiers créés : `/etc/systemd/system/ucarp@.service` (template), `/etc/ucarp/up.sh`, `/etc/ucarp/down.sh`, `/etc/ucarp/ucarp.pass`

Copier ces fichiers sur chaque machine, adapter SERVER_IP dans le service pour chaque nœud, puis activer le service (instructions ci-dessous).

Déploiement — commandes finales à exécuter sur chaque nœud

```
# Sur chaque nœud (adapter SERVER_IP dans le service pour la machine)
sudo mkdir -p /etc/ucarp
sudo cp /mnt/data/ucarp_scripts/up.sh /etc/ucarp/up.sh
sudo cp /mnt/data/ucarp_scripts/down.sh /etc/ucarp/down.sh
sudo cp /mnt/data/ucarp_scripts/ucarp.pass /etc/ucarp/ucarp.pass
sudo chmod 600 /etc/ucarp/ucarp.pass
sudo chmod +x /etc/ucarp/up.sh /etc/ucarp/down.sh
sudo cp /mnt/data/ucarp_scripts/ucarp@.service /etc/systemd/system/ucarp@.service
# Éditer /etc/systemd/system/ucarp@.service et remplacer SERVER_IP par l'adresse locale (ex:
192.168.81.41)
sudo systemctl daemon-reload
sudo systemctl enable ucarp@vrid1
sudo systemctl start ucarp@vrid1
sudo systemctl status ucarp@vrid1
```

Tests et validation (à inclure dans la recette)

- 1) Vérifier quelle machine possède la VIP : `ip a | grep 192.168.81.100`
- 2) Depuis le client : `curl -I http://192.168.81.100` ou ouvrir dans le navigateur

- 3) Simuler une panne du maître : `sudo systemctl stop ucarp@vrid1` sur node1
-> vérifier que node2 prend la VIP et répond
4) Remettre en service node1 : `sudo systemctl start ucarp@vrid1`
-> vérifier comportement selon politique (failback automatique ou non)

Annexes — points importants

- Choix de l'interface réseau : remplacez `enp0s3` par l'interface correcte (ip a pour la trouver).
- VRID (-v) doit être identique sur les deux nœuds, le mot de passe doit être identique.
- Le paramètre -P (préemption) force la reprise si priorité plus élevée est configurée.
- Attention aux routes et au firewall (ufw/iptables) : autoriser le trafic ACARP/VRRP si filtré.
- Pour le rapport BTS SIO, documentez les tests effectués (captures, logs `/var/log/syslog`).

Conclusion

La solution proposée avec UCARP est simple à mettre en place et répond aux exigences d'une architecture actif/passif pour la continuité d'un service Web. Les scripts fournis permettent une réplication facile en TP et une intégration dans un dossier de stage.

Annexes — contenu des scripts fournis

/etc/systemd/system/ucarp@.service

[Unit]

Description=UCARP Virtual Router Redundancy for %i

After=network.target

[Service]

Type=forking

Remplacer SERVER_IP par l'IP locale (ex: 192.168.81.41)

ExecStart=/usr/sbin/ucarp -i %i -s SERVER_IP -v 1 -p motdepasseVRRP -a 192.168.81.100 -u

/etc/ucarp/up.sh -d /etc/ucarp/down.sh -P 100

ExecStop=/usr/bin/pkill -f "ucarp -i %i"

[Install]

WantedBy=multi-user.target

/etc/ucarp/up.sh

#!/bin/bash

Script up : exécuté quand le nœud devient maître

IP_VIP=192.168.81.100

IFACE=enp0s3

Ajouter l'IP virtuelle

/sbin/ip addr add \${IP_VIP}/24 dev \${IFACE} || true

Assurer que Apache tourne

systemctl start apache2

/etc/ucarp/down.sh

#!/bin/bash

Script down : exécuté quand le nœud perd le rôle maître

IP_VIP=192.168.81.100

IFACE=enp0s3

Supprimer l'IP virtuelle

/sbin/ip addr del \${IP_VIP}/24 dev \${IFACE} || true

Optionnel: arrêter Apache pour éviter conflit (peut être commenté si désiré)

/bin/systemctl stop apache2

/etc/ucarp/ucarp.pass

motdepasseVRRP