# Load balancing and replication

During this practical work, we will play with load balancing, failover and replication.

## Question 1

In this question you will configure a HA proxy installation.

In the question_1 folder you will find a docker-compose file. It contains 2 apache server that serves the same page. They are behind a HA proxy container. You will have the HA proxy configuration exposed.

- Configure HA proxy in a round robin fashion
- Configure HA proxy for stickyness

You can observe the results in the logs.

## Question 2

In this question, you will set up shared IP mechanisms and experience failover.

You will need to launch virtualbox (as we need kernel modules for UCARP).

- Create 3 machines
- Configure them to communicate over the same (private) network with fixed IPs
- Configure machine 1 and 2 to share an IP address
- Ping the shared ip with machine 3
- Shutdown machine 1
- What do you observe ? How long did it took ?
- Start machine 1 again.
- Shutdown machine 2
- Same question.

## Question 3

In this question, we will set up a highly available NFS server.

- Use the 3 previously created machines. We will rely on their shared IP.
- Install a NFS server on machine 1 and 2
- Create the device
- Designate a master
- Create the file system on the device.
- Mount the file system on master
- Export it over NFS on master (using shared IP)
- Mount it using the client
- Create a file
- Shut down master
- Declare the slave as primary
- mount the volume and export it
- re-connect the client if needed
- See the file that was created

You can now automate the mouting process using UCARP

```
ucarp-downscript  /usr/local/bin/vip-down
ucarp-upscript   /usr/local/bin/vip-up
```

## Miscaleneous

Useful links :

- HA NFS with DRBD and Heartbeat : https://www.howtoforge.com/high-availability-nfs-with-drbd-plus-heartbeat

- UCARP on debian : https://debian-administration.org/article/678/Virtual_IP_addresses_with_ucarp_for_high-availability