

HAProxy

- [HAProxy Installation](#)
 - [\[How-To\] Install HAProxy on Ubuntu 24.04 LTS](#)
- [HAProxy Configuration](#)
 - [\[How-To\] Configure HAProxy for MariaDB Galera Cluster Load Balancing](#)

HAProxy Installation

[How-To] Install HAProxy on Ubuntu 24.04 LTS

Purpose

Purpose description.

Prerequisites

List of prerequisites:

- Root user or sudo user
- Debian 12 LXC or VM

Instructions

Step 1: Update and Install HAProxy

Install haproxy

```
sudo apt update
sudo apt install -y haproxy
```

Step 2: Enable and Start the HAProxy Service

Now that HAProxy is installed, you can enable the service and start the service:

```
sudo systemctl enable haproxy
sudo systemctl start haproxy
```

Then, check on the status of HAProxy:

```
sudo systemctl status haproxy
```


HAProxy Configuration

[How-To] Configure HAProxy for MariaDB Galera Cluster Load Balancing

Purpose

Explain the process of configuring the cfg file on haproxy to load balance a MariaDB Galera cluster of 3 servers.

Prerequisites

List of prerequisites:

- Root user or sudo user
- MariaDB Galera Clustered Servers (Minimum 3)
- HAProxy Server

Instructions

Step 1: Edit the HAProxy Config File

Edit the config file with:

```
sudo nano /etc/haproxy/haproxy.cfg
```

Use the following base config to either replace or edit the haproxy config on your servers to allow for load balancing of MariaDB Galera clustered servers:

```
global
  log /dev/log    local0
  log /dev/log    local1 notice
```

```
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin
stats timeout 30s
user haproxy
group haproxy
daemon
```

defaults

```
log    global
option tcplog
timeout connect 5s
timeout client 50s
timeout server 50s
```

frontend mariadb_frontend

```
bind 192.168.1.200:3306 # VIP and MariaDB port
mode tcp
default_backend mariadb_backend
```

backend mariadb_backend

```
mode tcp
balance roundrobin # Distribute traffic evenly
option tcp-check   # Health check for TCP (MariaDB port)
server mariadb1 192.168.1.101:3306 check
server mariadb2 192.168.1.102:3306
server mariadb3 192.168.1.103:3306
```

Once you update this config, you can move onto binding your VIP (Virtual IP) to your NIC.

Step 2: Bind VIP to NIC Manually

Now, we can bind the VIP you set in the config to the NIC. First, get the name of your NIC with the following command:

```
ip a
```

Look for something like ens18 or eno1. Once you know the name of your nic, use the following command to bind the VIP to the NIC:

```
sudo ip addr add <VIP>/32 dev <NIC_interface>
```

Verify the VIP is bound to the NIC with the same command:

```
ip a
```

Step 3: Make the VIP Persistent

On Debian, do this:

```
sudo nano /etc/network/interfaces.d/<interface>
```

Then, add:

```
auto <interface>:0
iface <interface>:0 inet static
    address <VIP>
    netmask 255.255.255.255
```

Then, restart networking:

```
sudo systemctl restart networking
```

On Ubuntu, do this:

```
sudo nano /etc/netplan/01-netcfg.yaml
```

Add an alias for the VIP under your interface configuration:

```
network:
  version: 2
  ethernets:
    <interface>:
      addresses:
        - <current-ip>/24
        - <VIP>/32
```

Then, apply netplan with the following command:

```
sudo netplan apply
```


