
HAProxy to Set Up MySQL Load Balancing

Droplet 2 - Node 1

Hostname: mysql-1

OS: ubuntu 14.04 Private IP: 172.17.0.2

Droplet 2 - Node 2

Hostname: mysql-2

OS: ubuntu 14.04 Private IP: 172.17.0.3

Droplet 1 - Load Balancer

Hostname: haproxy

OS: ubuntu 14.04 Private IP: 172.17.0.4

172.17.0.2 and 172.17.0.3 were already configured with master-master replication.

Prepare MySQL Servers

We need to prepare the MySQL servers by creating two additional users for HAProxy. The first user will be used by HAProxy to check the status of a server.

Because of the two mysql servers are in master-master configuration, configure the following only in anyone of the mysql server.

```
# mysql -u root -p -e "INSERT INTO mysql.user (Host,User) values  
( '172.17.0.4', 'haproxy_check' ); FLUSH PRIVILEGES;"
```

A MySQL user is needed with root privileges when accessing the MySQL cluster from HAProxy. The default root user on all the servers are allowed to login only locally. While this can be fixed by granting additional privileges to the root user, it is better to have a separate user with root privileges.

```
# mysql -u root -p -e "GRANT ALL PRIVILEGES ON *.* TO 'haproxy_root'@'172.17.0.4'  
IDENTIFIED BY 'password' WITH GRANT OPTION; FLUSH PRIVILEGES"
```

Replace **haproxy_root** and **password** with your own secure values. It is enough to execute these queries on one MySQL master as changes will replicate to others.

Install MySQL Client

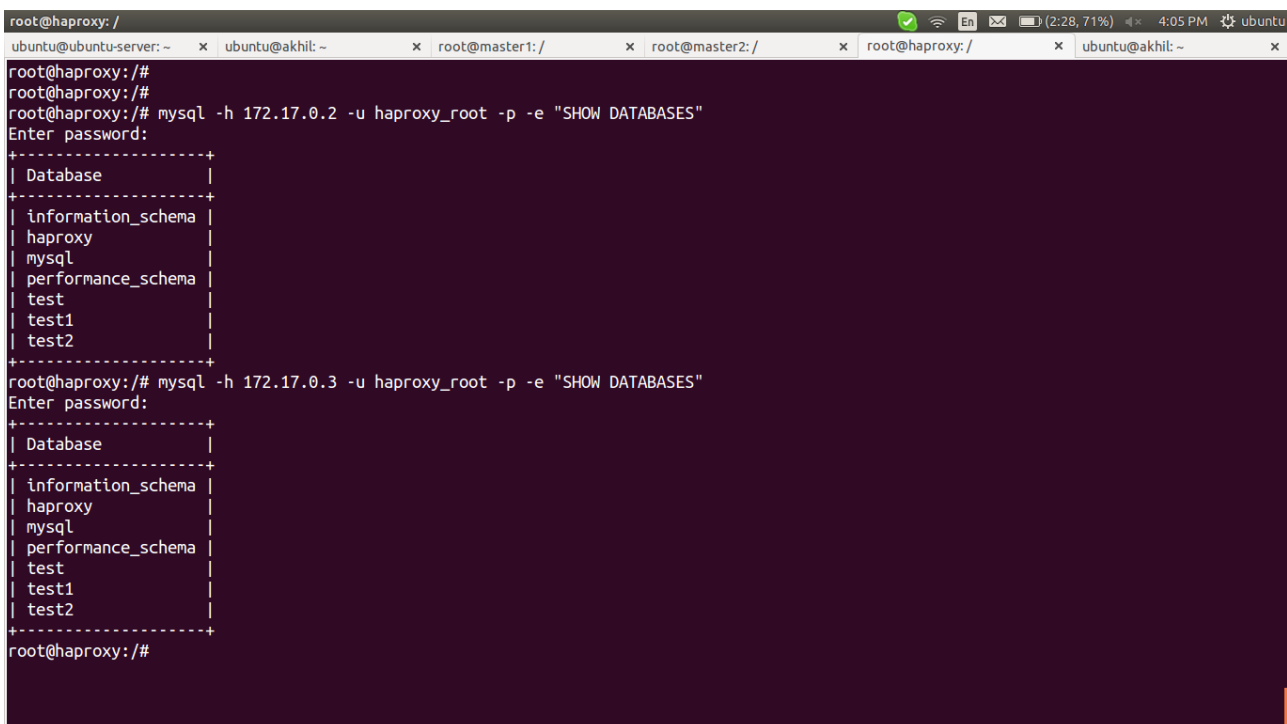
MySQL client has to be installed on the HAProxy droplet to test connectivity.

```
# apt-get install mysql-client
```

Now try executing a query on one of the masters as the **haproxy_root** user.

```
# mysql -h 172.17.0.2 -u haproxy_root -p -e "SHOW DATABASES"
```

```
# mysql -h 172.17.0.3 -u haproxy_root -p -e "SHOW DATABASES"
```

A terminal window screenshot showing the execution of MySQL client commands on a haproxy server. The terminal has a dark purple background. The prompt is root@haproxy:/. The first command executed is mysql -h 172.17.0.2 -u haproxy_root -p -e "SHOW DATABASES". It prompts for a password, and then displays a list of databases: information_schema, haproxy, mysql, performance_schema, test, test1, and test2. The second command executed is mysql -h 172.17.0.3 -u haproxy_root -p -e "SHOW DATABASES". It also prompts for a password and displays the same list of databases. The terminal window has several tabs open at the top, including ubuntu@ubuntu-server, ubuntu@akhil, root@master1, root@master2, and root@haproxy. The system status bar at the top right shows 2:28, 71% battery, and 4:05 PM on ubuntu.

```
root@haproxy: /
ubuntu@ubuntu-server: ~ x ubuntu@akhil: ~ x root@master1: / x root@master2: / x root@haproxy: / x ubuntu@akhil: ~ x
root@haproxy: /#
root@haproxy: /#
root@haproxy: /# mysql -h 172.17.0.2 -u haproxy_root -p -e "SHOW DATABASES"
Enter password:
+-----+
| Database |
+-----+
| information_schema |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
root@haproxy: /# mysql -h 172.17.0.3 -u haproxy_root -p -e "SHOW DATABASES"
Enter password:
+-----+
| Database |
+-----+
| information_schema |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
root@haproxy: /#
```

Installing HAProxy

On the HAProxy server install the package.

```
# apt-get install software-properties-common
# add-apt-repository ppa:vbernat/haproxy-1.6
# apt-get update
# apt-get install haproxy
```

Enable HAProxy to be started by the init script.

```
# nano /etc/default/haproxy
```

```
ENABLED=1
```

Configuring HAProxy

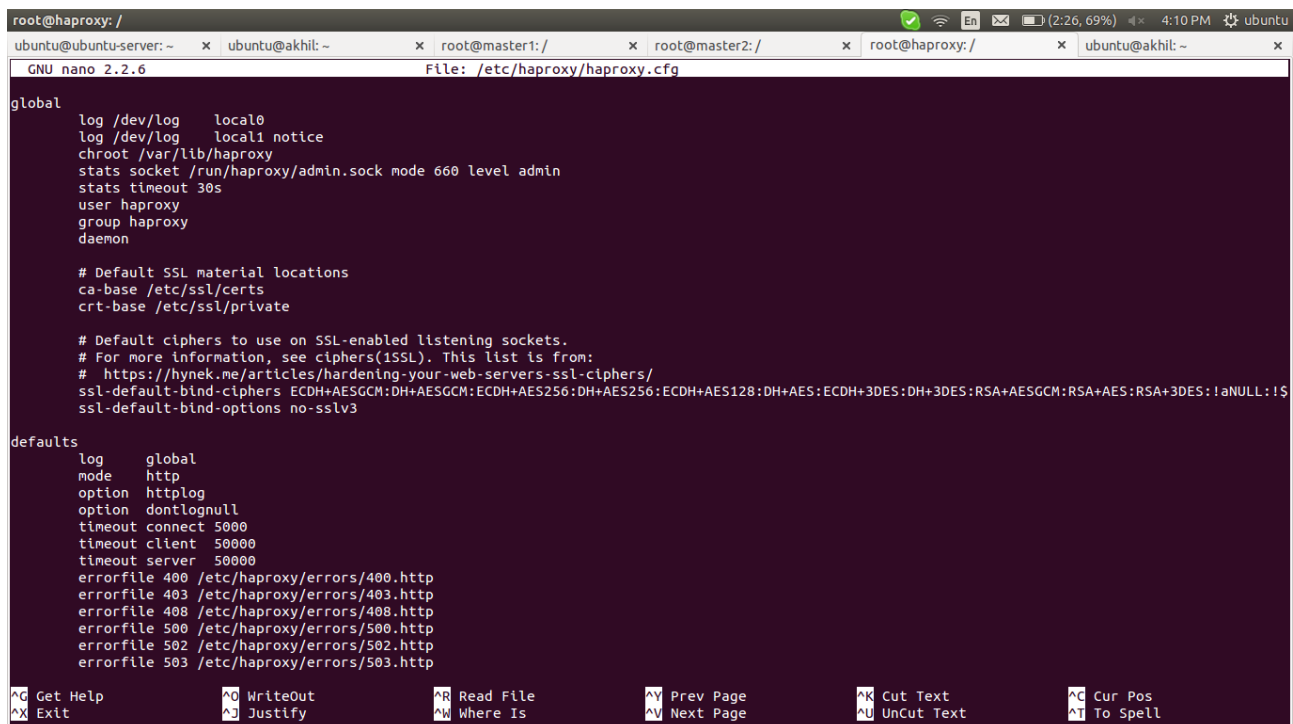
Rename the original configuration file

```
# cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.dup
```

Create and edit a new one

```
# nano /etc/haproxy/haproxy.cfg
```

The first block is the global and defaults configuration block.

A screenshot of a terminal window with a dark background. The terminal shows the nano text editor editing the file /etc/haproxy/haproxy.cfg. The editor's title bar indicates 'GNU nano 2.2.6' and 'File: /etc/haproxy/haproxy.cfg'. The configuration content is divided into two main sections: 'global' and 'defaults'. The 'global' section includes settings for logging, chroot, stats socket, timeout, user, group, daemon, and SSL material locations. The 'defaults' section includes settings for log, mode, option, timeout, errorfile, and server. The terminal window has several tabs open at the top, including 'root@haproxy: /' and 'ubuntu@akhil: ~'. The bottom status bar of the nano editor shows various keyboard shortcuts like '^G Get Help', '^X Exit', '^O WriteOut', '^J Justify', '^R Read File', '^W Where Is', '^Y Prev Page', '^V Next Page', '^K Cut Text', '^U UnCut Text', '^C Cur Pos', and '^T To Spell.

Moving to the main configuration part.

```
listen webui # used for web interface http://IP:8888
```

```
bind *:8888
```

```
stats enable
```

```
stats uri /
```

```
mode http
```

```
stats auth admin:admin
```

```
listen mysql-cluster # used for connecting mysql servers to haproxy server
```

```
bind *:3306
```

```
mode tcp
```

```
option mysql-check user haproxy_check
```

```
balance roundrobin
```

```
server master1 172.17.0.2:3306 check weight 2
```

```
server master1 172.17.0.3:3306 check
```

weight 2 means whenever the clients write on mysql, haproxy choose master1 mysql server 2 times when master2 mysql server choose once.

service haproxy start

http://172.17.0.4:8888

Statistics Report for HAProxy - Mozilla Firefox

172.17.0.4:8888

HAProxy version 1.6.9, released 2016/08/30

Statistics Report for pid 4599

> General process information

pid = 4599 (process #1, nbproc = 1)
uptime = 0d 0h 13m 37s
system limits: memmax = unlimited; ulimit-n = 4034
maxsock = 4034; maxconn = 2000; maxpipes = 0
current conns = 1; current pipes = 0/0; conn rate = 0/sec
Running tasks: 1/7; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance

backup UP
backup UP, going down
backup DOWN, going up
not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
Scope :
Hide DOWN servers
Refresh now
CSV export

External resources:
Primary site
Updates (v1.5)
Online manual

Queue		Session rate			Sessions					Bytes		Denied		Errors		Warnings		Server													
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtle		
webui																															
Frontend	0	0	0	1	-	1	1	2 000	4	0	0s	6 951	242 967	0	0	0	0	0	0	0	0	OPEN			0	0	0	0			
Backend	0	0	0	0		0	0	200	0	0	0s	6 951	242 967	0	0	0	0	0	0	0	0	13m37s UP			0	0	0	0			

Queue		Session rate			Sessions					Bytes		Denied		Errors		Warnings		Server													
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtle		
mysql-cluster																															
Frontend	0	0	0	0	-	0	0	2 000	0	0	0s	0	0	0	0	0	0	0	0	0	0	OPEN									
master1	0	0	-	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	13m37s UP	L7OK/0 in 0ms	1	Y	-	0	0	0s	-		
master2	0	0	-	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	13m37s UP	L7OK/0 in 0ms	1	Y	-	0	0	0s	-		
Backend	0	0	0	0		0	0	200	0	0	?	0	0	0	0	0	0	0	0	0	13m37s UP		2	2	0	0	0	0s			

Testing Load Balancing

To check if load balancing is working query the **server_id** variable twice or more.

Use haproxy ip address to access mysql server from remote machine

mysql -h 172.17.0.4 -u haproxy_root -p -e "show variables like 'server_id'"

```
root@haproxy: /
root@master1: / x root@master2: / x root@haproxy: / x
root@haproxy: /# mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+
| Variable_name | Value |
+-----+
| server_id     | 1     |
+-----+
root@haproxy: /# mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+
| Variable_name | Value |
+-----+
| server_id     | 1     |
+-----+
root@haproxy: /# mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+
| Variable_name | Value |
+-----+
| server_id     | 2     |
+-----+
root@haproxy: /# mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+
| Variable_name | Value |
+-----+
| server_id     | 1     |
+-----+
root@haproxy: /# mysql -h 127.0.0.1 -u haproxy_root -p -e "show variables like 'server_id'"
Enter password:
+-----+
| Variable_name | Value |
+-----+
| server_id     | 1     |
+-----+
root@haproxy: /#
```

try to create databases from another machine with haproxy ip address

remote machine using haproxy server ip address

mysql -u haproxy_root -h 172.17.0.4 -p

```
root@haproxy: /
root@master1: / x root@master2: / x root@haproxy: / x
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
7 rows in set (0.00 sec)

mysql> create database akhil;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akhil |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
8 rows in set (0.00 sec)

mysql>
```

master1 (mysql server 1)

```
root@master1: /
root@master1: /
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
7 rows in set (0.08 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akhll |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
8 rows in set (0.01 sec)

mysql>
```

master2 (mysql server 2)

```
root@master2: /
root@master1: /
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
7 rows in set (0.06 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akhll |
| haproxy |
| mysql |
| performance_schema |
| test |
| test1 |
| test2 |
+-----+
8 rows in set (0.00 sec)

mysql>
```

link ::

<https://www.digitalocean.com/community/tutorials/how-to-use-haproxy-to-set-up-mysql-load-balancing--3>
