HAProxy — Connection Routing

Clients shouldn't connect to a specific database node — that node might be a replica today and gone tomorrow. **HAProxy gives them one stable address** and always routes to the right place, using Patroni's REST API as the health check. Run this page on `pg-haproxy`.

The trick: Patroni's REST health checks

Patroni exposes an HTTP endpoint on port `8008` on every node:

- `GET /primary` returns `200` **only on the current leader** (otherwise `503`).
- `GET /replica` returns `200` **only on healthy replicas**.

So HAProxy doesn't need to know who the primary is — it just asks each node, and the answers change automatically during a failover. (These `GET` endpoints need no authentication.)

Install

```
sudo apt-get install -y haproxy postgresql-client-18
```

Configure

Replace `/etc/haproxy/haproxy.cfg` with:

```
global
    maxconn 2000
    log /dev/log local0

defaults
    log global
    mode tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
```

```
    timeout check 5s

listen stats
    mode http
    bind *:7000
    stats enable
    stats uri /
    stats refresh 5s

# Writes -> whichever node answers 200 on /primary (the leader)
listen postgres_write
    bind *:5432
    option httpchk GET /primary
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server pg-sv01 10.100.100.104:5432 maxconn 200 check port 8008
    server pg-sv02 10.100.100.105:5432 maxconn 200 check port 8008
    server pg-sv03 10.100.100.106:5432 maxconn 200 check port 8008

# Reads -> any node that answers 200 on /replica (the replicas)
listen postgres_read
    bind *:5433
    option httpchk GET /replica
    http-check expect status 200
    balance roundrobin
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server pg-sv01 10.100.100.104:5432 maxconn 200 check port 8008
    server pg-sv02 10.100.100.105:5432 maxconn 200 check port 8008
    server pg-sv03 10.100.100.106:5432 maxconn 200 check port 8008
```

Note that the **health check runs against port `8008`** (Patroni) while traffic is proxied to port `5432` (PostgreSQL) — that's the `check port 8008` directive.

Firewall (UFW)

```
sudo ufw allow from 10.100.100.0/24 to any port 5432 proto tcp
sudo ufw allow from 10.100.100.0/24 to any port 5433 proto tcp
sudo ufw allow from 10.100.100.0/24 to any port 7000 proto tcp
```

Start and verify

```
sudo systemctl enable --now haproxy

# Writes go to the primary (in_recovery = f):
PGPASSWORD=ChangeMe_Postgres psql -h pg-haproxy -p 5432 -U postgres \
  -c "SELECT inet_server_addr() AS server, pg_is_in_recovery();"

# Reads go to a replica (in_recovery = t):
PGPASSWORD=ChangeMe_Postgres psql -h pg-haproxy -p 5433 -U postgres \
  -c "SELECT inet_server_addr() AS server, pg_is_in_recovery();"
```

The write port returns the primary's address with `in_recovery = f`; the read port returns a replica with `in_recovery = t`. Your application now uses **`pg-haproxy:5432`** for writes and **`:5433`** for reads. The HAProxy stats page is at `http://pg-haproxy:7000/`.

Revision #1
Created 2026-06-08 16:43:25 UTC by Eslam Shapsough
Updated 2026-06-08 16:43:25 UTC by Eslam Shapsough