HACLUSTER-1

Description	28 Agustus 2025
Due Date	@August 28, 2025
Status	Done
Tags	IMPORTANT & URGENT
Attachments	high-availabiliity-design.pdf
Notes	HA CLUSTER tanpa drbd

HA Cluster: Pacemaker/Corosync in Docker + Flask & PostgreSQL (No DRBD First)

Notasi konteks

- **[HOST]** = jalankan di OS host.
- **[CONTAINER]** = jalankan di dalam container `ha-stack`.
- Jalankan di **serverA** & **serverB** kecuali disebut khusus.
- IP contoh:
  - serverA = `192.168.56.27`
  - serverB = `192.168.56.28`
  - VIP = `192.168.56.30/24`

> Target tahap ini: Flask + PostgreSQL HA tanpa DRBD.
> DRBD/replikasi dikerjakan terakhir setelah layanan stabil.

---

1) Persiapan host & direktori

1.1 [HOST | A & B] Matikan cluster lama (kalau ada)

```
sudo systemctl stop pacemaker corosync pcsd 2>/dev/null || true
sudo systemctl disable pacemaker corosync pcsd 2>/dev/null || true
sudo pkill -9 -f 'corosync|pacemakerd|pcsd' 2>/dev/null || true
```

1.2 [HOST | A & B] Buat direktori bind-mount & log

```
sudo mkdir -p /etc/corosync /etc/pacemaker /var/lib/pacemaker /var/lib/pcsd
sudo mkdir -p /var/log/cluster /var/log/cluster/supervisor /var/log/cluster/pcs
d
sudo mkdir -p /mnt
sudo mount --make-rshared /mnt 2>/dev/null || true
```

---

2) Dockerfile (image kontrol-plane) & build

2.1 [HOST | A] Simpan Dockerfile (`fiqri/ha-stack:1.3`)

```
FROM ubuntu:24.04
ENV DEBIAN_FRONTEND=noninteractive LANG=C.UTF-8 TZ=Asia/Jakarta

RUN apt-get update \
 && apt-get install -y --no-install-recommends software-properties-common
gnupg ca-certificates curl \
 && add-apt-repository -y universe \
 && apt-get update \
 && apt-get install -y --no-install-recommends \
    pacemaker corosync pcs \
    resource-agents-base resource-agents-extra \
    drbd-utils \
    iproute2 iputils-ping net-tools dnsutils \
    openssl \
```

```
      supervisor procps vim less \
 && rm -rf /var/lib/apt/lists/*

RUN groupadd -r haclient 2>/dev/null || true \
 && id -u hacluster >/dev/null 2>&1 || useradd -r -s /usr/sbin/nologin hacluster
\
 && echo 'hacluster:123' | chpasswd \
 && mkdir -p /etc/corosync /etc/pacemaker /var/lib/pacemaker /var/lib/pcsd \
           /var/log/pcsd /var/log/supervisor /etc/supervisor/conf.d \
 && chgrp -R haclient /var/lib/pcsd || true && chmod 750 /var/lib/pcsd

# supervisor main
RUN cat <<'EOF' > /etc/supervisor/supervisord.conf
[unix_http_server]
file=/var/run/supervisor.sock
chmod=0700

[supervisord]
nodaemon=true
logfile=/var/log/supervisor/supervisord.log
pidfile=/var/run/supervisord.pid
childlogdir=/var/log/supervisor

[rpcinterface:supervisor]
supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinter
face

[supervisorctl]
serverurl=unix:///var/run/supervisor.sock

[include]
files = /etc/supervisor/conf.d/*.conf
EOF

# program yang diawasi
RUN cat <<'EOF' > /etc/supervisor/conf.d/pacemaker.conf
```

```
[program:pcsd]
command=/bin/bash -lc "mkdir -p /var/log/pcsd /var/lib/pcsd; chgrp -R haclie
nt /var/lib/pcsd || true; chmod 750 /var/lib/pcsd; exec /usr/sbin/pcsd"
autorestart=true
priority=10
stderr_logfile=/var/log/supervisor/pcsd.err.log
stdout_logfile=/var/log/supervisor/pcsd.out.log

[program:corosync]
command=/usr/sbin/corosync -f
autorestart=true
priority=20
stderr_logfile=/var/log/supervisor/corosync.err.log
stdout_logfile=/var/log/supervisor/corosync.out.log

[program:pacemaker]
command=/usr/sbin/pacemakerd -f
autorestart=true
priority=30
stderr_logfile=/var/log/supervisor/pacemakerd.err.log
stdout_logfile=/var/log/supervisor/pacemakerd.out.log
EOF

HEALTHCHECK --interval=10s --timeout=3s --retries=15 \
  CMD corosync-cmapctl totem.nodeid >/dev/null 2>&1 || exit 1

ENTRYPOINT ["/usr/bin/supervisord","-c","/etc/supervisor/supervisord.conf"]
```

> Opsional untuk kenyamanan CLI: tambahkan paket pacemaker-cli-utils agar crm_mon tersedia di container.

2.2 [HOST | A] Build image & kirim ke serverB

```
cd /home/pqri/hacluster
docker build -t fiqri/ha-stack:1.3 .
docker save fiqri/ha-stack:1.3 | ssh pqri@serverB 'docker load'
```

3) Compose v2 & supervisor config (di-host, dimount ke container)

3.1 [HOST | A & B] Simpan `supervisor/supervisord.conf` (host)

```
[unix_http_server]
file=/var/run/supervisor.sock
chmod=0700

[supervisord]
nodaemon=true
logfile=/var/log/supervisor/supervisord.log
pidfile=/var/run/supervisord.pid
childlogdir=/var/log/supervisor

[rpcinterface:supervisor]
supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface

[supervisorctl]
serverurl=unix:///var/run/supervisor.sock

[include]
files = /etc/supervisor/conf.d/*.conf
```

3.2 [HOST | A & B] Simpan `supervisor/pacemaker.conf` (host)

```
[program:pcsd]
command=/bin/bash -lc "mkdir -p /var/log/pcsd /var/lib/pcsd; chgrp -R haclie
nt /var/lib/pcsd || true; chmod 750 /var/lib/pcsd; exec /usr/sbin/pcsd"
autorestart=true
priority=10
stderr_logfile=/var/log/supervisor/pcsd.err.log
stdout_logfile=/var/log/supervisor/pcsd.out.log

[program:corosync]
command=/usr/sbin/corosync -f
autorestart=true
priority=20
stderr_logfile=/var/log/supervisor/corosync.err.log
stdout_logfile=/var/log/supervisor/corosync.out.log

[program:pacemaker]
command=/usr/sbin/pacemakerd -f
autorestart=true
priority=30
stderr_logfile=/var/log/supervisor/pacemakerd.err.log
stdout_logfile=/var/log/supervisor/pacemakerd.out.log
```

3.3 [HOST | A & B] Simpan `docker-compose.yml`

> Ganti hostname: sesuai node.

```
services:
  ha-stack:
    image: fiqri/ha-stack:1.3
    pull_policy: never
    container_name: ha-stack
    hostname: serverA   # di serverB: serverB
    network_mode: host
```

```
    privileged: true
    restart: unless-stopped
    environment:
      - TZ=Asia/Jakarta
    volumes:
      - /etc/corosync:/etc/corosync
      - /etc/pacemaker:/etc/pacemaker
      - /var/lib/pacemaker:/var/lib/pacemaker
      - /var/lib/pcsd:/var/lib/pcsd
      - /var/log/cluster:/var/log
      - /dev:/dev
      - /lib/modules:/lib/modules:ro
      - /mnt:/mnt:rshared
      - /usr/bin/docker:/usr/bin/docker:ro
      - /var/run/docker.sock:/var/run/docker.sock
      - ./supervisor/supervisord.conf:/etc/supervisor/supervisord.conf:ro
      - ./supervisor/pacemaker.conf:/etc/supervisor/conf.d/pacemaker.conf:ro
    extra_hosts:
      - "serverA:192.168.56.27"
      - "serverB:192.168.56.28"
    healthcheck:
      test: ["CMD-SHELL", "corosync-cmapctl totem.nodeid >/dev/null 2>&1"]
      interval: 10s
      timeout: 3s
      retries: 15
      start_period: 20s
```

4) Corosync config & authkey

4.1 [HOST | A & B] Tulis `/etc/corosync/corosync.conf`

```
sudo tee /etc/corosync/corosync.conf > /dev/null <<'EOF'
totem {
```

```
    version: 2
    transport: knet
    cluster_name: ha-cluster
    crypto_cipher: aes256
    crypto_hash: sha256
    token: 3000
    consensus: 3600
    max_messages: 200
}

nodelist {
    node {
        name: serverA
        nodeid: 1
        ring0_addr: 192.168.56.27
    }
    node {
        name: serverB
        nodeid: 2
        ring0_addr: 192.168.56.28
    }
}

quorum {
    provider: corosync_votequorum
    two_node: 1
    wait_for_all: 1
}

logging {
    to_logfile: yes
    logfile: /var/log/corosync.log
    to_syslog: yes
    timestamp: on
    debug: off
}
```

```
EOF
```

4.2 [HOST | A] Generate & sebar `authkey`

```
sudo corosync-keygen
sudo chmod 400 /etc/corosync/authkey
scp /etc/corosync/authkey pqri@serverB:/home/pqri/
ssh pqri@serverB 'sudo mv /home/pqri/authkey /etc/corosync/ && sudo chmo
d 400 /etc/corosync/authkey'
```

---

5) Jalankan stack & cek pcsd

5.1 [HOST | A & B] Start compose

```
cd /home/pqri/hacluster
docker compose down --remove-orphans || true
docker rm -f ha-stack 2>/dev/null || true
docker compose up -d
```

5.2 [CONTAINER | A & B] Cek supervisor & port

```
supervisorctl -c /etc/supervisor/supervisord.conf status
ss -ltnp | grep 2224 || echo "pcsd belum listen"
corosync -t || true
```

5.3 [CONTAINER | A & B] Jika pcsd belum listen – perbaiki & restart

```
mkdir -p /var/lib/pcsd /var/log/pcsd
chown -R hacluster:haclient /var/lib/pcsd /var/log/pcsd
chmod -R 770 /var/lib/pcsd /var/log/pcsd
supervisorctl -c /etc/supervisor/supervisord.conf restart pcsd
```

---

6) Auth pcsd & bootstrap cluster

6.1 [CONTAINER | A & B] Pastikan password user

```
echo "hacluster:123" | chpasswd
```

6.2 [CONTAINER | serverA] Authorize by IP lalu nama

```
pcs host auth 192.168.56.27 192.168.56.28 -u hacluster -p 123
pcs host auth serverA serverB -u hacluster -p 123
```

6.3 [CONTAINER | serverA] Jika error known-hosts/HTTP 500 – bersihkan & ulang

```
rm -f /var/lib/pcsd/known-hosts
supervisorctl -c /etc/supervisor/supervisord.conf restart pcsd
sleep 2
pcs host auth serverA serverB -u hacluster -p 123
```

6.4 [CONTAINER | serverA] Setup & start cluster

```
pcs cluster setup --name ha-cluster serverA serverB
pcs cluster start --all
```

```
pcs cluster enable --all
```

6.5 [CONTAINER | serverA] Set properti dasar (lab env)

```
pcs property set stonith-enabled=false
pcs property set no-quorum-policy=stop
```

---

7) Cek cluster dasar

7.1 [CONTAINER | A & B] Pasang CLI (opsional)

```
which crm_mon >/dev/null 2>&1 || (apt-get update && apt-get install -y pacem
aker-cli-utils || true)
```

7.2 [CONTAINER | serverA] Cek status & quorum (tanpa crm_mon)

```
pcs status nodes corosync
pcs resource status
corosync-quorumtool -s
```

---

8) **(Bersih-bersih opsional)** Hapus sisa DRBD lama

> Abaikan jika belum pernah set DRBD.

```
pcs property set maintenance-mode=true
pcs resource ungroup app_group 2>/dev/null || true
pcs resource delete drbd_pg --force 2>/dev/null || true
pcs resource delete drbd0-clone --force 2>/dev/null || true
```

```
pcs resource delete drbd0 --force 2>/dev/null || true
pcs resource cleanup || true
pcs property set maintenance-mode=false
```

---

9) Siapkan **PostgreSQL & Flask** (tanpa DRBD)

9.1 [HOST | A & B] Siapkan folder data & image

```
sudo mkdir -p /var/lib/pg-ha
sudo chown -R 999:999 /var/lib/pg-ha
sudo systemctl disable --now postgresql* 2>/dev/null || true
docker pull postgres:16
```

9.2 [HOST | A] Build image Flask & sebar

> Folder app: /home/pqri/flaskapp (kode kamu sudah memakai 127.0.0.1 untuk DB)

```
cd /home/pqri/flaskapp
docker build -t flaskapp:latest .
docker save flaskapp:latest | ssh pqri@serverB 'docker load'
```

---

10) Definisi resource (PG → VIP → Flask)

10.1 [CONTAINER | serverA] Maintenance ON

```
pcs property set maintenance-mode=true
```

10.2 [CONTAINER | serverA] Postgres (RA docker)

```
pcs resource create pg_docker ocf:heartbeat:docker \
  name=postgres-ha image=postgres:16 \
  run_opts="--network=host --name=postgres-ha --restart=unless-stopped \
        -e POSTGRES_USER=hauser -e POSTGRES_PASSWORD=hapassword
-e POSTGRES_DB=locationdb \
        -v /var/lib/pg-ha:/var/lib/postgresql/data" \
  reuse=true force_kill=true \
  op start timeout=120s op stop timeout=120s op monitor interval=20s timeout
=60s
```

10.3 [CONTAINER | serverA] VIP (NIC sesuaikan, mis. `enp0s8`)

```
pcs resource create vip ocf:heartbeat:IPaddr2 \
  ip=192.168.56.30 cidr_netmask=24 nic=enp0s8 \
  op monitor interval=10s timeout=20s
# Jika beda NIC antar node, bisa set nic="" agar auto-pick
# pcs resource update vip nic=""
```

10.4 [CONTAINER | serverA] Flask (RA docker)

```
pcs resource create flask_docker ocf:heartbeat:docker \
  name=flaskapp-ha image=flaskapp:latest \
  run_opts="--network=host --name=flaskapp-ha --restart=unless-stopped \
        -e SERVER_NAME=$(hostname) \
        -e DATABASE_URL=postgresql://hauser:hapassword@127.0.0.1:5432/lo
cationdb" \
  reuse=true force_kill=true \
  op start timeout=90s op stop timeout=90s op monitor interval=20s timeout=
60s
```

10.5 [CONTAINER | serverA] Group & constraints

```
pcs resource group add app_group vip 2>/dev/null || true
pcs resource group add app_group flask_docker 2>/dev/null || true

pcs constraint order start pg_docker then start app_group kind=Mandatory sy
mmetrical=true
pcs constraint colocation add app_group with pg_docker INFINITY

pcs resource defaults update resource-stickiness=200
pcs property set stonith-enabled=false
pcs property set no-quorum-policy=stop
```

10.6 [CONTAINER | serverA] Maintenance OFF & cek

```
pcs property set maintenance-mode=false
pcs resource status
```

---

11) Inisialisasi schema DB (sekali saja, di node aktif)

```
# [HOST] di node tempat postgres-ha berjalan
docker exec -it postgres-ha psql -U hauser -d locationdb -c \
"CREATE TABLE IF NOT EXISTS location_logs (
  id bigserial PRIMARY KEY,
  server_name text,
  location_name text,
  latitude double precision,
  longitude double precision,
  timestamp timestamptz DEFAULT now()
);"
```

---

12) Verifikasi & uji akses

```
# [CONTAINER] status node/resource
pcs status nodes corosync
pcs resource status
# [HOST] cek kontainer
docker ps --format 'table {{.Names}}\t{{.Image}}\t{{.Status}}'
# [HOST] cek VIP di host aktif
ip -br addr | grep 192.168.56.30 || echo "VIP belum aktif di host ini"
# [HOST] uji HTTP
curl -sS http://192.168.56.30:8080/ | head -n 5
```

---

13) Uji failover

13.1 Planned switchover (graceful)

```
# pindah dari serverA ke serverB
docker exec -it ha-stack pcs node standby serverA
sleep 6
docker exec -it ha-stack pcs resource status
curl -sS http://192.168.56.30:8080/ | head -n 3

# kembalikan
docker exec -it ha-stack pcs node unstandby serverA
sleep 6
docker exec -it ha-stack pcs resource status
```

13.2 Unplanned (stop cluster di satu node)

```
# [HOST | node yang dimatikan]
docker exec -it ha-stack supervisorctl -c /etc/supervisor/supervisord.conf sto
p pacemaker
```

```
sleep 1
docker exec -it ha-stack supervisorctl -c /etc/supervisor/supervisord.conf sto
p corosync
# start kembali
docker exec -it ha-stack supervisorctl -c /etc/supervisor/supervisord.conf sta
rt corosync
sleep 1
docker exec -it ha-stack supervisorctl -c /etc/supervisor/supervisord.conf sta
rt pacemaker
```

---

14) Troubleshooting cepat

- **VIP tidak muncul di node tujuan**

```
pcs resource update vip nic=""   # auto-pick NIC
pcs resource restart vip
ip -br addr | grep 192.168.56.30
```

- **Flask 500 / Restarting**

```
docker logs flaskapp-ha --tail=100
# biasanya tabel belum ada → jalankan inisialisasi schema (bagian 11)
# pastikan env DATABASE_URL terpasang di resource (lihat run_opts)
```

- **Postgres tidak jalan di node tujuan**

```
# [HOST | B] pastikan dir & izin, serta image
mkdir -p /var/lib/pg-ha && chown -R 999:999 /var/lib/pg-ha
docker image inspect postgres:16 >/dev/null 2>&1 || docker pull postgres:1
6
# [CONTAINER]
```

```
pcs resource why pg_docker
```

- **Port bentrok**

```
ss -ltnp | egrep ':8080|:5432' || echo "8080 & 5432 bebas"
```

- **PCSD 500 / Unable to authenticate**

```
# [CONTAINER] perbaiki dir pcsd & re-auth
mkdir -p /var/lib/pcsd /var/log/pcsd
chown -R hacluster:haclient /var/lib/pcsd /var/log/pcsd
chmod -R 770 /var/lib/pcsd /var/log/pcsd
rm -f /var/lib/pcsd/known-hosts /var/lib/pcsd/*token* 2>/dev/null || true
supervisorctl -c /etc/supervisor/supervisord.conf restart pcsd
pcs host auth serverA serverB -u hacluster -p 123
```

- **Membersihkan pin/move sementara**

```
pcs resource clear pg_docker 2>/dev/null || true
pcs resource clear app_group 2>/dev/null || true
```

---

15) Catatan untuk tahap DRBD (nantinya)

- Pindahkan volume PG ke `v /mnt/drbd/pgdata:/var/lib/postgresql/data` **setelah** DRBD & Filesystem resource siap.
- Tambahkan resource `drbd0` (promotable), `fs_drbd`, dan **order+colocation**: DRBD → FS → PG → app_group.
- Dengan ini, data ikut pindah saat failover.

Selesai. Pada tahap ini, Flask & PostgreSQL sudah HA dengan VIP dan colocation, *tanpa DRBD*. Replikasi data akan ditambahkan pada tahap berikutnya.