HA flask only

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

Tujuan & Topologi

- **Node:** `serverA` (192.168.56.27), `serverB` (192.168.56.28)
- **VIP uji:** `192.168.56.30/24`
- **Nama kontainer cluster:** `ha-stack`
- **Gaya:** corosync & pacemaker full di **container**, host hanya file config.

---

0) Prasyarat (keduanya)

- Docker & Docker Compose terpasang
- Akses SSH antar host (untuk salin image/berkas)

---

1) Struktur direktori (di **kedua host**)

```
mkdir -p /home/pqri/ha/{docker,supervisor,corosync,flask-frontend}
cd /home/pqri/ha
```

---

2) Image kontrol-plane (build di **serverA**)

Buat `/home/pqri/ha/docker/Dockerfile` :

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

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 pacemaker-cli-utils \
      resource-agents-base resource-agents-extra \
      drbd-utils iproute2 iputils-ping iputils-arping net-tools dnsutils \
      openssl supervisor procps vim less kmod e2fsprogs lsof jq postgresql-client \
 && rm -rf /var/lib/apt/lists/*

# runtime pindah ke /home/cluster (log & pcsd), pacemaker state tetap default /var/lib/pacemaker
RUN groupadd -r haclient 2>/dev/null || true \
 && id -u hacluster >/dev/null 2>&1 || useradd -r -s /usr/sbin/nologin hacluster \
 && mkdir -p /home/cluster/pcsd /home/cluster/log/pcsd /home/cluster/log/supervisor /home/cluster/supervisor/conf.d \
 && mkdir -p /var/lib /var/log \
 && rm -rf /var/lib/pcsd /var/log/pcsd \
 && ln -s /home/cluster/pcsd /var/lib/pcsd \
 && ln -s /home/cluster/log/pcsd /var/log/pcsd

# supervisord default (boleh dioverride file host, opsional)
RUN printf "%s\n" \
"[unix_http_server]" \
"file=/var/run/supervisor.sock" \
"chmod=0700" \
"[supervisord]" \
"nodaemon=true" \
"logfile=/home/cluster/log/supervisor/supervisord.log" \
"pidfile=/home/cluster/supervisord.pid" \
"childlogdir=/home/cluster/log/supervisor" \
"[rpcinterface:supervisor]" \
"supervisor.rpcinterface_factory=supervisor.rpcinterface:make_main_rpcinterface" \
"[supervisorctl]" \
"serverurl=unix:///var/run/supervisor.sock" \
"[include]" \
"files=/home/cluster/supervisor/conf.d/*.conf" \
> /etc/supervisor/supervisord.conf

# program yang diawasi
RUN printf "%s\n" \
"[program:pcsd]" \
"command=/usr/sbin/pcsd" \
"autorestart=true" \
"priority=10" \
"stderr_logfile=/home/cluster/log/pcsd/pcsd.err.log" \
"stdout_logfile=/home/cluster/log/pcsd/pcsd.out.log" \
"" \
"[program:corosync]" \
"command=/usr/sbin/corosync -f" \
"autorestart=true" \
"priority=20" \
"stderr_logfile=/home/cluster/log/corosync.err.log" \
"stdout_logfile=/home/cluster/log/corosync.out.log" \
"" \
"[program:pacemaker]" \
"command=/usr/sbin/pacemakerd -f" \
"autorestart=true" \
"priority=30" \
"stderr_logfile=/home/cluster/log/pacemaker.err.log" \
"stdout_logfile=/home/cluster/log/pacemaker.out.log" \
> /home/cluster/supervisor/conf.d/pacemaker.conf

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"]
```

Build & salin ke **serverB**:

```
cd /home/pqri/ha/docker
docker build -t fiqri/ha-stack:2.2-ha .
docker save fiqri/ha-stack:2.2-ha | ssh pqri@serverB 'docker load'
```

---

3) Konfigurasi Corosync (file host)

Buat `/home/pqri/ha/corosync/corosync.conf` (periksa IP/hostname):

```
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: /home/cluster/log/corosync.log
  to_syslog: yes
  timestamp: on
  debug: off
}
```

Generate `authkey` (di **serverA**), lalu copy ke **serverB**:

```
docker run --rm --entrypoint bash -v /home/pqri/ha/corosync:/cor fiqri/ha-stack:2.2-ha -lc 'corosync-keygen -l && cp /etc/corosync/authkey /cor/authkey'
chmod 400 /home/pqri/ha/corosync/authkey
scp /home/pqri/ha/corosync/authkey pqri@serverB:/home/pqri/ha/corosync/authkey
```

4) Docker Compose (keduanya)

Buat `/home/pqri/ha/docker-compose.yml`:

```
services:
  ha-stack:
    image: fiqri/ha-stack:2.2-ha
    container_name: ha-stack
    hostname: serverA          # di serverB ubah: serverB
    network_mode: host
    privileged: true
    restart: unless-stopped
    environment:
      - TZ=${TZ:-Asia/Jakarta}
    volumes:
      # (opsional override supervisor)
     # - ./supervisor/supervisord.conf:/etc/supervisor/supervisord.conf:ro
     # - ./supervisor/pacemaker.conf:/home/cluster/supervisor/conf.d/pacemaker.conf:ro
      # WAJIB corosync
      - ./corosync/corosync.conf:/etc/corosync/corosync.conf:ro
      - ./corosync/authkey:/etc/corosync/authkey:ro
      # RA docker perlu ini
      - /usr/bin/docker:/usr/bin/docker:ro
      - /var/run/docker.sock:/var/run/docker.sock
      # opsional untuk mount DRBD nantinya
      - /mnt:/mnt:rshared
    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
```

Start kontainer di **keduanya**:

```
cd /home/pqri/ha
docker compose up -d
# (di serverB pastikan hostname: serverB, lalu jalankan juga)
```

---

5) Bootstrap cluster (jalankan di **container serverA**)

```
docker exec -it ha-stack bash -lc '
echo "hacluster:123" | chpasswd;  # opsional, kalau belum

pcs host auth serverA serverB -u hacluster -p 123
pcs cluster setup --name ha-cluster serverA serverB
pcs cluster start --all
pcs cluster enable --all

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

corosync-quorumtool -s
pcs status
'
```

> Pastikan Quorate: Yes dan kedua node Online.

---

6) Build **flask_frontend** (serverA)

Buat `/home/pqri/ha/flask-frontend/app.py`:

```
from flask import Flask
import socket
app = Flask(__name__)

@app.get("/")
def index():
    h = socket.gethostname()
    return f"<h1>HA Frontend</h1><p>Aktif di node: <b>{h}</b></p>"

@app.get("/healthz")
def healthz():
    return "ok", 200
```

Buat `/home/pqri/ha/flask-frontend/Dockerfile`:

```
FROM python:3.12-slim
WORKDIR /app
RUN pip install flask gunicorn
COPY app.py /app/
CMD ["gunicorn", "-b", "0.0.0.0:8080", "app:app"]
```

Build & salin ke **serverB**:

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

7) Resource VIP + Flask (jalankan di container serverA)

Deteksi **NIC** untuk VIP (atau pakai `enp0s8` jika sudah pasti):

Deteksi nama NIC yang tepat (di dalam container serverA)

```
docker exec -it ha-stack bash -lc '
NIC=$(ip route get 192.168.56.30 | sed -n "s/.*dev \([^ ]\+\).*/\1/p"); echo "NIC=$NIC"
'
```

Catat nilai `NIC=` yang keluar (mis. `ens33`, `eth0`, dll).

```
docker exec -it ha-stack bash -lc '
VIP=192.168.56.30
NIC=$(ip route get $VIP | sed -n "s/.*dev \([^ ]\+\).*/\1/p"); echo "NIC=$NIC"

pcs resource create vip ocf:heartbeat:IPaddr2 \
  ip=192.168.56.30 cidr_netmask=24 nic=$NIC \
  op monitor interval=10s timeout=20s

pcs resource create flask_frontend ocf:heartbeat:docker \
  name=flask-frontend-ha image=flask-frontend:latest \
  run_opts="--network=host --name=flask-frontend-ha --restart=unless-stopped" \
  reuse=true force_kill=true \
  op start timeout=90s op stop timeout=90s op monitor interval=20s timeout=60s

pcs resource group add app_group vip flask_frontend
pcs resource defaults update resource-stickiness=200
pcs resource cleanup vip flask_frontend
pcs status
'
```

---

8) Uji akses & failover

Tes HTTP via VIP:

```
curl -sS http://192.168.56.30:8080/ | head -n 5
```

Failover terencana (pindah ke serverB lalu kembali):

```
docker exec -it ha-stack bash -lc '
pcs node standby serverA; sleep 6; pcs resource status;
pcs node unstandby serverA; sleep 3; pcs resource status;
'
```

> Ulangi curl — hostname di halaman harus berubah sesuai node aktif.

---

9) Troubleshooting ringkas

- **`crm_mon` / `cibadmin` tidak ada** → image sudah include `pacemaker-cli-utils`. Kalau container lama, jalankan: `apt-get update && apt-get install -y pacemaker-cli-utils`
- **Corosync FATAL / CS_ERR_LIBRARY** → cek:
  - `corosync.conf` **multiline** (blok `node { ... }` tidak satu baris),
  - `authkey` **mode 400**,
  - `uname -n` di container cocok `name:` di `nodelist`,

  - IP `ring0_addr` benar & reachable.
- **VIP tidak muncul** → set `nic=` yang benar:

  ```
  ip route get 192.168.56.30 | sed -n 's/.*dev \([^ ]\+\).*/\1/p'
  pcs resource update vip nic=<NAMA_NIC>
  ```

- **HTTP 500 / port 8080 bentrok** (pakai `-network=host`):
  - Bebaskan 8080 di host aktif:

    ```
    ss -lntp | grep ':8080 ' || echo bebas
    docker ps -a | awk '/8080->/ {print $1}' | xargs -r docker rm -f
    fuser -k 8080/tcp || true
    ```

  - Lalu:

    ```
    docker exec -it ha-stack bash -lc 'pcs resource cleanup flask_frontend; pcs resource restart flask_frontend'
    ```

  - Alternatif: gunakan mapping port (`run_opts="-p 18080:8080"`) agar tidak bentrok.