Reconfiguração automática do haproxy.cfg do HAPROXY

ChatGPT:

Perfeito! Aqui está um script cron completo e seguro para fazer o reload do haproxy.cfg em todos os containers haproxy do Swarm, sem reiniciar os containers — apenas recarregando o processo haproxy com kill -HUP.

 Script: reload-haproxy-containers.sh

bash

INICIO SCRIPT

```
#!/bin/bash
# Nome da imagem usada no serviço haproxy
IMAGE_NAME="haproxytech/haproxy-debian:3.3"
# Comando de reload usado dentro do container
RELOAD_CMD="haproxy -c -f /usr/local/etc/haproxy/haproxy.cfg && kill -HUP 1"
# Busca containers que usam a imagem especificada
CONTAINERS=$(docker ps --filter "ancestor=$IMAGE_NAME" --format "{{.ID}}")

if [ -z "$CONTAINERS" ]; then
    echo "[INFO] Nenhum container HAProxy em execução com imagem '$IMAGE_NAME'"
    exit 0
fi
echo "[INFO] Recarregando configuração HAProxy em $(echo "$CONTAINERS" | wc -l)
container(s)..."
for cid in $CONTAINERS; do
    echo " → Validando e recarregando HAProxy no container $cid..."
    docker exec "$cid" sh -c "$RELOAD_CMD" && \
        echo "  Container $cid recarregado com sucesso." || \
        echo "  Erro ao recarregar o container $cid."
done
```

FIM SCRIPT

□□ Como usar

1. Salve o script

sudo mkdir -p /usr/local/bin
sudo nano /usr/local/bin/reload-haproxy-containers.sh
Cole o conteúdo, salve, e depois:
sudo chmod +x /usr/local/bin/reload-haproxy-containers.sh

2. Agende via crontab

Edite o cron para o usuário que gerencia o Swarm (geralmente root):
sudo crontab -e
Adicione a linha para executar de hora em hora:
0 * * * * /usr/local/bin/reload-haproxy-containers.sh >> /var/log/reload-haproxy.log 2>&1

□□ Segurança adicional (opcional)

Se quiser ser ainda mais seletivo, você pode filtrar por nome do serviço ou label, ex:
docker ps --filter "name=mystack_haproxy" --format "{{.ID}}"
Ou com label:
docker ps --filter "label=com.docker.stack.namespace=mystack" --format "{{.ID}}"
□ Resultado
Com isso, seu Swarm:

□ Executará reload leve e seguro do haproxy.cfg de hora em hora

□ Sem precisar redeploy, sem downtime

□ Sincronizado via GlusterFS automaticamente

□ Sem scripts dentro dos containers

Revision #2
Created 2025-07-22 14:35:20 UTC by Administrador
Updated 2025-07-22 14:44:10 UTC by Administrador