



HEALTHCHECKS E SCRIPTS AUTOMATIZADOS NO DOCKER

Responsável: João Pedro Toledo Gonçalves.

Data: 26/01/2026

MANUAL TÉCNICO - HEALTHCHECKS E SCRIPTS AUTOMATIZADOS NO DOCKER

Código: ITGINF 0011/26 | **Classificação:** RESTRITO

Responsável: João Pedro Toledo Gonçalves | **Data:** 26/01/2026

1. HISTÓRICO DE REVISÃO

NOTA

REGRA DE OURO:

1. **Autor:** João Pedro Toledo Gonçalves.
2. **Descrição:** Criação do documento.

Data	Versão	Descrição	Autor
26/01/2026	1.0	Criação Inicial	João Pedro Toledo Gonçalves

2. OBJETIVO

Implementar verificações de saúde (Healthchecks) para auto-cura de aplicações e configurar tarefas agendadas (Cron) de forma compatível com containers.

3. PRÉ-REQUISITOS

[] Conhecimento de Dockerfile e Docker Compose.

4. PASSO A PASSO (EXECUÇÃO)

Etapa 1: Definindo HEALTHCHECK no Dockerfile

NOTA

NOTA: O Docker executa esse comando periodicamente. Se falhar X vezes, marca o container como `unhealthy`.

1. [x] Edite seu `Dockerfile`.
2. [x] Adicione a instrução (exemplo para web server):

```
HEALTHCHECK --interval=30s --timeout=3s --start-period=5s --retries=3 \
  CMD curl -f http://localhost/ || exit 1
```

Etapa 2: Definindo HEALTHCHECK no Compose

1. [x] No `docker-compose.yml`, você pode sobrescrever ou definir:

```
version: '3.8'
services:
  web:
    image: nginx
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localhost"]
      interval: 1m30s
      timeout: 10s
      retries: 3
```

Etapa 3: Scripts Agendados (Cron)

IMPORTANTE

IMPORTANTE: Containers devem ter UM processo principal. Rodar `cron` junto com `apache` no mesmo container é má prática ("Fat Container").

Abordagem A: Container Dedicado (Sidecar)

1. [x] Crie um container pequeno (Alpine) só para rodar o script e sair.
2. [x] Agende a execução desse container pelo `cron` do **HOST**.

```
# No crontab do HOST (crontab -e)
# Roda backup do banco postgres todo dia as 03:00
0 3 * * * docker exec meu-postgres pg_dumpall > /backups/db.sql
```

Abordagem B: Ofelia (Job Scheduler for Docker)

1. [x] Use uma imagem especializada que escuta events do Docker.

```
services:
  ofelia:
    image: mcuadros/ofelia:latest
    depends_on:
      - web
    command: daemon --docker
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock

  web:
    image: my-app
    labels:
      ofelia.enabled: "true"
      ofelia.job-exec.datecron.schedule: "@every 10s"
      ofelia.job-exec.datecron.command: "echo 'Backup process...'"
```

5. SOLUÇÃO DE PROBLEMAS (TROUBLESHOOTING)

Problema 1: Container sempre em `starting`

- **Causa:** O healthcheck nunca passa com sucesso.
- **Solução:**
 1. [x] Rode o comando do healthcheck manualmente: `docker exec -it [ID] curl ...`.
 2. [x] Verifique se o `curl`/`wget` está instalado na imagem.

Problema 2: Cron dentro do container não roda

- **Causa:** Variáveis de ambiente não passam automaticamente para o cron.
- **Solução:** Sempre use caminhos absolutos nos scripts e carregue as variáveis se necessário.

6. VALIDAÇÃO FINAL

- [] `docker ps` mostra status `(healthy)`?
- [] O comando agendado foi executado no horário previsto?