GEARFLOW LAB PRODUCTION SRE RUNBOOK 10 | DOCKER HEALTHCHECK + SYSTEMD + MACOS LAUNCHD

Deadlocked PID 1 Container Watchdog & Host Auto-Heal Matrix

Open Watchdog Generator ->

Incident Context: In standalone Docker Compose (outside Swarm/K8s), `restart: unless-stopped` **only** fires when PID 1 terminates. When a Node.js event loop, database socket, or GPU runner deadlocks while PID 1 stays alive, Docker marks the container `Up (unhealthy)` and leaves it wedged indefinitely while Nginx returns `HTTP 504`.

1. Production Probe & Memory Guard Matrix (`127.0.0.1` IPv4 Avoids Alpine `::1` False Failures)

Service	Image-Safe IPv4 `healthcheck.test` Command	Grace / Win	`mem_limit`	Critical Runtime Guard
n8n (:5678)	`wget -qO- http://127.0.0.1:5678/healthz || exit 1`	30s / 45s	1536m-2048m	`--max-old-space-size=1536` + `init: true`
ollama (:11434)	`ollama list >/dev/null 2>&1 || exit 1`	25s / 60s	16384m+	`OLLAMA_KEEP_ALIVE=15m` (no `curl` in image!)
postgres (:5432)	`pg_isready -h 127.0.0.1 -p 5432 -U postgres`	40s / 30s	2048m-4096m	`stop_grace_period: 60s` (clean WAL checkpoint)
nginx (:8080)	`wget -qO- http://127.0.0.1:8080/healthz || exit 1`	15s / 30s	256m-512m	Stub `/healthz return 200` off access_log
openobserve	`wget -qO- http://127.0.0.1:5080/healthz || exit 1`	20s / 45s	1024m	Bind `127.0.0.1:5080:5080` + `10m` log cap
mcp_http (:8000)	`python3 -c "import urllib.request; ..."`	20s / 45s	1024m	Works in `python:3.12-slim` without `curl`/`wget`

2. Linux `systemd` Event-Driven `unhealthy` Auto-Healer

```
# /usr/local/bin/gfl-docker-autoheal.sh (chmod 0755, 0% idle CPU)
#!/usr/bin/env bash
set -euo pipefail
declare -A LAST=()
docker events --filter "type=container" \
  --filter "event=health_status: unhealthy" \
  --filter "label=com.gearflowlab.autoheal=true" \
  --format "{{.ID}} {{.Actor.Attributes.name}}" |
while read -r CID CNAME; do
  NOW=$(date +%s); PREV=${LAST[$CNAME]:-0}
  (( NOW - PREV < 120 )) && continue
  LAST[$CNAME]=$NOW
  logger -t gfl-autoheal "Restarting wedged container $CNAME"
  docker restart --time 20 "$CID"
done
```

```
# /etc/systemd/system/gfl-docker-autoheal.service
[Service]
ExecStart=/usr/local/bin/gfl-docker-autoheal.sh
Restart=always
RestartSec=5s
MemoryMax=64M
```

3. macOS `LaunchAgent` Guard + Staging `SIGSTOP` Drill

```
<!-- ~/Library/LaunchAgents/
com.gearflowlab.ollama.watchdog.plist -->
<plist version="1.0"><dict>
  <key>Label</
key><string>com.gearflowlab.ollama.watchdog</string>
  <key>ProgramArguments</key>
  <array><string>/usr/local/bin/gfl-ollama-macos-
guard.sh</string></array>
  <key>StartInterval</key><integer>60</integer>
  <key>ThrottleInterval</key><integer>30</integer>
  <key>ProcessType</key><string>Background</string>
  <key>RunAtLoad</key><true/>
</dict></plist>
```

```
# Safe Staging Chaos Drill: freeze PID 1 without killing it!
docker kill --signal=SIGSTOP gfl-n8n
# Confirm healthy -> unhealthy after (interval * retries)s & auto-restart:
docker inspect --format '{{.State.Status}} {{.State.Health.Status}}' gfl-n8n
sudo journalctl -u gfl-docker-autoheal.service -n 15 --no-pager
# Manual unfreeze fallback if testing passive mode (deadlock=0):
docker kill --signal=SIGCONT gfl-n8n
```

1-Click Prefill Watchdog Policies: n8n + systemd Auto-Heal | Ollama + macOS LaunchAgent | PostgreSQL 17 pg_isready | Error Decoder | All Tools (/tools/)