GEARFLOW LAB PRODUCTION SRE RUNBOOK 06 | LINUX OOM-KILLER + ZRAM + VPS TCO

Docker Exit Code 137 (SIGKILL OOM) & Self-Hosted VPS RAM Sizing

Open RAM Calculator ->

Incident Context: `Exit Code 137` (`128 + 9 = SIGKILL`) strikes when either a container's cgroup v2 `memory.max` (`mem_limit`) is breached or host physical RAM + swap runs dry (`Out of memory: Killed process`). Co-locating n8n Node.js workers, `PostgreSQL 17`, `Redis`, and `OpenObserve` on undersized `2GB-4GB` VPS nodes without `zstd` ZRAM causes midnight webhook OOM kills.

1. Empirical Container Peak RSS & Hardened `mem_limit` Reference Table (`100` Conns Baseline)

Container Service	Measured RSS	Safe `mem_limit`	Runtime Memory Cap Flag (Prevents Blind Heap Growth)
n8n Main / Webhook	`~530 MiB`	`1024m - 1536m`	`NODE_OPTIONS=--max-old-space-size=1152` (`0.75 * mem_limit`) + `EXECUTIONS_MODE=queue`.
n8n Queue Worker (x2)	`~700 MiB`	`1536m` (`768m` ea)	`command: worker --concurrency=10` + cap V8 heap per replica (`--max-old-space-size=576`).
`PostgreSQL 17` (Pooled)	`~590 MiB`	`1024m - 2048m`	`shared_buffers=256MB`, `max_connections=60` behind PgBouncer (`shm_size: "256mb"`).
`PgBouncer` + `Nginx`	`~22 + 36 MiB`	`256m` each	`POOL_MODE=transaction`, `DEFAULT_POOL_SIZE=20`, `MAX_PREPARED_STATEMENTS=100`.
`Redis 7.4` (Queue Broker)	`~77 MiB`	`384m`	`--maxmemory 268mb --maxmemory-policy noeviction` (never evict BullMQ job keys!).
`OpenObserve` Log Engine	`~275 MiB`	`768m - 1024m`	Replaces `3.5 GiB+` ELK stack for JSON container logs & OTLP trace ingestion.
`Ollama` (7B Q4_K_M CPU)	`~4850 MiB`	`6144m+`	Requires `>= 8GB-16GB` node; add `310-460 MiB` Linux kernel/Slab + `30%` spike headroom.

2. `Exit Code 137` Root-Cause Triage CLI (Host vs cgroup)

```
# 1. Confirm whether OOMKilled == true (vs manual docker stop timeout)
docker inspect <container> \
  --format 'Exit={{.State.ExitCode}} OOM={{.State.OOMKilled}} At={{.State.FinishedAt}}'

# 2. Distinguish cgroup mem_limit hit vs whole-host RAM exhaustion
sudo dmesg -T | grep -i -E 'oom|killed process|memory cgroup' | tail -n 15

# 3. Inspect live cgroup v2 oom_kill counter & peak memory.current
docker stats --no-stream --format "table {{.Name}}\t{{.MemUsage}}\t{{.MemPerc}}"
cat /sys/fs/cgroup/system.slice/docker-*.scope/memory.events 2>/dev/null | grep oom
```

3. Kernel `zstd` ZRAM + `sysctl` Cushion Setup

```
# /etc/systemd/zram-generator.conf (Ubuntu 24.04 / Debian 12)
[zram0]
zram-size = min(ram / 2, 4096)
compression-algorithm = zstd
swap-priority = 100
```

```
# Install zram-generator & tune kernel reclaim behavior
sudo apt-get update && sudo apt-get install -y systemd-zram-generator
sudo tee /etc/sysctl.d/99-gfl-oom-guard.conf >/dev/null <<'EOF'
vm.swappiness = 10
vm.vfs_cache_pressure = 50
vm.overcommit_memory = 0
EOF
sudo systemctl daemon-reload && sudo systemctl start systemd-zram-setup@zram0
sudo sysctl --system && zramctl
```

4. Cloud VPS vs Local Mini Server 3-Year TCO Break-Even Formula (`15W` Idle at `$0.16/kWh` = `$2.10/mo`)

`Break_Even_Months = Local_CapEx / (Cloud_Monthly_USD - Local_Power_Monthly_USD)`:

RAM Tier	Cloud VPS/mo	3-Yr Cloud TCO	Local Mini PC 3-Yr TCO	Break-Even
8 GB Stack (n8n + PG + Redis)	`$48 / mo`	`$1,728`	$675 ($599 HW + $76 power)	**`13.1 months`**
16 GB Stack (+ OpenObserve)	`$96 / mo`	`$3,456`	$675 ($599 HW + $76 power)	**`6.4 months`**
32 GB Stack (+ Local `Ollama`)	`$192 / mo`	`$6,912`	$875 ($799 HW + $76 power)	**`4.2 months`**

Quick Prefill Presets & OOM Guide: n8n Queue + PG + Redis | Full Observability Stack | Exit Code 137 Triage (#1996) | All Tools (/tools/)