GEARFLOW LAB PRODUCTION SRE RUNBOOK 07 | DOCKER + NGINX + POSTGRES + N8N + OLLAMA

# DevOps & AI Stack Error Log Triage Matrix (11 Critical Signatures)

Open Log Decoder ->

**Incident Context:** Production container incidents rarely surface as isolated single-line faults. Upstream kernel cgroup `memory.max` breaches (`Exit 137`) or PostgreSQL connection exhaustion (`53300`) immediately manifest downstream as Nginx `502`/`504` gateway errors. Always isolate the upstream causal layer first.

## 1. Top 11 Production Error Signatures, Root Causes & Zero-Downtime Fixes

| Signature | Subsystem / Log Token | Protocol / Kernel Root Cause | Zero-Downtime Remediation |
|---|---|---|---|
| `42P05` | Postgres / PgBouncer `prepared statement "s1" exists` | `pool_mode = transaction` rotating backend sockets without protocol statement tracking. | Set `max_prepared_statements = 100` in `pgbouncer.ini` (>= 1.21) and run `RELOAD;`. |
| `53300` | Postgres `FATAL: sorry, too many clients already` | Direct ORM/webhook fan-out exceeding `max_connections` ( 9.5 MB RSS per backend). | Route apps via PgBouncer `:6432` (`default_pool_size = 20`, `max_client_conn = 400`). |
| `502` | Nginx `upstream prematurely closed` / `too big header` | Upstream container OOM crash, 5s Uvicorn keepalive race, or >4k JWT/ cookie response header. | Enable `keepalive 64; Connection "";`, `proxy_buffer_size 16k;`, `--timeout-keep-alive 65`. |
| `504` | Nginx `upstream timed out (110)` at 60s | Default `proxy_read_timeout 60s` or `fastcgi_read_timeout` hit during SSE / LLM / import. | Set `proxy_buffering off; proxy_read_timeout 3600s;` + emit 15s `: keepalive` SSE pings. |
| `499` | Nginx `client closed request (status 499)` | Caller SDK timeout shorter than upstream p99 latency or `TIME_WAIT` socket exhaustion. | Align SDK timeout > proxy timeout; pool upstream sockets (`keepalive 64`). |
| `413` | Nginx / n8n `Request Entity Too Large` | Webhook/multipart body exceeds Nginx `1m` `client_max_body_size` or `N8N_PAYLOAD_SIZE_MAX`. | Set `client_max_body_size 32m;` in Nginx and `N8N_PAYLOAD_SIZE_MAX=32` in Compose. |
| `Exit 137` | Docker / Kernel `128 + 9` (`SIGKILL` / `OOMKilled`) | Container cgroup v2 `memory.max` breach or host RAM exhaustion (`Out of memory: Killed`). | Configure `zstd` ZRAM, cap Node V8 `--max-old-space-size`, and right-size `mem_limit`. |
| `SQLITE_BUSY` | SQLite / n8n `database is locked` (code 5) | Concurrent webhook writers contending on single-file SQLite write lock under burst traffic. | Enable `PRAGMA journal_mode=WAL; busy_timeout=5000;` or migrate n8n to PostgreSQL 17. |
| `EACCES` | Docker Bind Mount `permission denied` | Host directory UID/GID (`root:root`) mismatches non-root container user (`node` UID `1000`). | `chown -R 1000:1000 ./data` on host or align Compose `user: "1000:1000"`. |
| `nvidia-smi` | GB10 UMA `Memory-Usage: Not Supported` | Grace Blackwell 128GB coherent LPDDR5x UMA shares host page tables; discrete FB query N/A. | Read `/proc/meminfo` (`MemAvailable`), `nvtop`, or `tegrastats` instead of discrete FB bars. |
| `lsof +L1` | Linux VFS `No space left` (`df` vs `du` gap) | `rm` on active `*-json.log` unlinks filename while `dockerd` holds open FD blocks. | Reclaim in-place via `sudo truncate -s 0 /proc/<pid>/fd/<fd>` + `max-size: "10m"`. |

## 2. Multi-Layer Cascading Failure Chains (Always Fix Leftmost Trigger First)

- **Chain A (Memory OOM Cascade):** `cgroup v2 memory.max hit` -> `Kernel SIGKILL (Exit 137)` -> `Nginx 502 upstream prematurely closed` -> `Webhook retry storm`.
- **Chain B (Connection Pool Cascade):** `Uncapped webhook burst` -> `PostgreSQL SQLSTATE 53300` -> `Queue worker stalls` -> `Nginx 504 / Cloudflare 524 at 100s`.
- **Chain C (Phantom Disk Cascade):** `Unrotated *-json.log deleted via rm` -> `Open inode holds 100% disk (lsof +L1)` -> `ENOSPC aborts WAL / SQLite commits`.

## 3. Read-Only Host & Container Triage One-Liners

```
# 1. Check Exit 137 OOMKilled state & kernel cgroup OOM log
docker inspect <name> --format 'Exit={{.State.ExitCode}} OOM={{.State.OOMKilled}}'
sudo dmesg -T | grep -iE 'oom|killed process|cgroup' | tail -n 8

# 2. Detect unlinked deleted log files holding disk inodes
sudo lsof -nP +L1 | grep -E 'deleted.*json\.log'

# 3. Audit wildcard 0.0.0.0 ports bypassing Ubuntu UFW
sudo ss -tulpn | grep '0.0.0.0:'
```

## 4. Database, Proxy & UMA Verification One-Liners

```
# 4. Audit PgBouncer transaction pool & prepared statement cap
psql -h 127.0.0.1 -p 6432 -U pgbouncer pgbouncer -c "SHOW POOLS;"
psql -h 127.0.0.1 -p 6432 -U pgbouncer pgbouncer -c "SHOW CONFIG;" \
  | grep -E 'pool_mode|max_prepared_statements'

# 5. Measure true unified memory headroom on NVIDIA GB10 / Linux
awk '/MemTotal|MemAvailable|SwapFree/ {printf "%s %.1f GiB\n",$1,$2/1048576}' /proc/meminfo
```

**1-Click Browser Triage Presets (0 B Server Upload):** 42P05 Preset | Exit 137 OOM | 502 Cascade | GB10 UMA | lsof +L1 | All Tools (/tools/)