**MCPX‑KendoBridge — Rollback Runbook (Graceful SSE Drain & Image Flip)**

**Document:** runbooks/rollback.docx  
**Version:** 2.0.0 (Kendo Migration)  
**Last Updated:** 2025‑09‑27  
**Owner:** SRE Lead (Responsible) — DoSE (Accountable) — DocFactory (Author)  
**Applies to:** API (.NET 8, Streamable‑HTTP + SSE), Admin Portal (KendoReact Fluent v12, read‑only)

**Purpose.** Provide a **repeatable, auditable** rollback procedure that **preserves active SSE streams**, honors **sticky sessions by Mcp‑Session‑Id**, and restores the **last known‑good image** with minimal impact. This runbook aligns with Technijian GitHub‑first SDLC, four environments (**Alpha → Beta → RTM → Prod**), and Evidence Pack retention (≥ 1 year).

**DB & Secrets Compliance (always in effect):**  
**Add‑only** schema; **Stored‑procedure‑only** DAL; **No‑Hard‑Coding** of dynamic values (child cmd/args/cwd, request timeout, heartbeat cadence, Origin allow‑list, feature flags). All dynamic values are **DB‑sourced** via sp\_Config\_\*, sp\_Feature\_IsEnabled, sp\_Lookup\_Get. **Secrets** (SQL connection strings, Telerik license) are **never** stored in code/DB/logs; configure them **only** in **GitHub Environments**.

**1) Rollback Triggers (when to use this runbook)**

Initiate rollback if any of the following are **sustained** (after a quick triage ≤ 10 min):

| **Trigger** | **Threshold / Signal** | **Source** |
| --- | --- | --- |
| Availability degradation | < 99% over 10 min or spike of 5xx | API availability SLI |
| **SSE TTFB** regression | p95 > 200 ms for 15 min | Streaming TTFB SLI |
| Readiness failures | /ready failing > 2 min | Readiness probe |
| Policy regression | Surge in origin\_forbidden or feature\_disabled not explained by traffic | Error‑code counters |
| Child instability | Rising child\_restart\_count, frequent bad\_gateway\_child\_unavailable | Metrics/logs |
| Security/exposure | Suspected secret leakage (stop and rotate separately) | Secret scanning / incident channel |

If regressions are **limited to configuration** (e.g., incorrect allow‑list), prefer **Config Rollback** (§3.1) before code rollback.

**2) Preconditions & Inputs**

* **Incident declared** (SEV level set) and comms channel open (SRE + Dev + QA + DoSE).
* **Last Known‑Good (LKG) image digest** from Evidence Pack or release notes.
* **Ingress SSE pass‑through** confirmed (no buffering) and **sticky routing** by Mcp‑Session‑Id available.
* **Environment approvals** available (Alpha/Beta/RTM/Prod).
* **DB policy** unchanged (no destructive DDL; SP contracts intact).

**3) Rollback Types (choose the least‑disruptive first)**

**3.1 Config Rollback (DB‑sourced, non‑secret)**

**When:** Origin allow‑list error, timeout/heartbeat mis‑tune, legacy flag toggled by mistake.  
**What:** Update **AppConfig/FeatureFlag** via **SPs** or idempotent seed migration (add‑only).  
**Steps:**

1. Identify offending key(s) in /config/effective (non‑secret surface).
2. Prepare an **idempotent MERGE** seed to restore prior value(s).
3. Execute via approved DBA path; capture before/after snapshots.
4. Verify: /ready=ok; /config/effective reflects rollback; error spikes subside.
5. Attach evidence to release.

**Never** store or rollback **secrets** via DB—secrets live in **GitHub Environments** only.

**3.2 Feature Disable (fast stopgap)**

**When:** Legacy endpoints caused issues.  
**What:** Set EnableLegacyHttpSse=false using sp\_Feature\_IsEnabled path; verify 403 feature\_disabled.  
**Verify:** Post‑change errors abate; primary /mcp unaffected.

**3.3 Code Rollback (image flip to LKG)**

**When:** Binary/regression defect in current image.  
**What:** Revert deployment to **LKG digest**; **gracefully drain** SSE.  
**Verify:** SLOs within budgets; errors normalize; post‑rollback checks pass.

**4) Procedure — Code Rollback with Graceful SSE Drain**

The sequence below is framed for Kubernetes‑style rollouts; adapt equivalent controls to your platform. Keep **sticky routing** by Mcp‑Session‑Id throughout.

**4.1 Quiesce new traffic (per replica)**

1. **Set readiness to fail** (temporarily) to stop new connections:
   * Reduce pod **readiness** (e.g., set temp readiness gate or scale down ***new*** ReplicaSet if a partial rollout has started).
   * Keep **liveness** intact to avoid hard kill.
2. **Maintain SSE drain:** Ensure ingress is **not buffering** text/event-stream and timeouts accommodate the configured heartbeat cadence (Network:SseKeepAliveSeconds).

**4.2 Drain active SSE streams**

1. Track per‑pod **active stream count** and **oldest stream age**.
2. Allow in‑flight streams to complete, up to **RequestTimeoutSeconds** (DB‑sourced).
3. If streams exceed the window, send final message (if supported) and close gracefully.

**4.3 Flip image to LKG (deployment‑level)**

1. Update deployment image to **LKG digest** (preferred) or tag.
2. Apply with **rolling update**; keep **PodDisruptionBudget (PDB)** to avoid mass eviction.
3. Ensure **sticky session routing** keeps returning clients on existing pods until new ones are ready.

**4.4 Warm‑up & re‑admit traffic**

1. Wait for /ready=ok on **LKG** pods (child spawn probe ok; DB reachable).
2. Re‑enable readiness for new pods; phase out old pods after SSE drain.
3. Confirm:
   * /mcp JSON latency p50/p95 within budget (≤ 300 ms / ≤ 800 ms).
   * **SSE TTFB p95 ≤ 200 ms**; heartbeats at configured cadence.
   * Error codes (origin\_forbidden, missing\_session\_id, feature\_disabled) at normal baseline.

**4.5 Evidence & communication**

1. Capture /ready, /healthz, /config/effective snapshots.
2. Export monitoring images (Availability, Latency, **TTFB**, Readiness timeline).
3. Update incident with **timeline** and **root cause candidate**; attach evidence to the Release (retain ≥ 1 year).

**5) Environment‑Specific Guidance**

| **Env** | **Guidance** |
| --- | --- |
| **Alpha** | Prefer Config Rollback for speed; code rollback acceptable without canary. |
| **Beta** | Run perf smoke after rollback; confirm budgets and SSE quality before resuming tests. |
| **RTM** | **Validates against Prod DB (read‑only)** — code rollback should **not** alter DB state. Confirm **parity** again after rollback. |
| **Prod** | Abort canary immediately; route back to stable LKG; execute code rollback with **graceful drain**; perform **24‑hour** post‑rollback checks as with a release. |

**6) Quick Commands (illustrative; adapt to your platform)**

Replace placeholders (<ns>, <deploy>, <digest>, <base>) as appropriate.

**Identify current vs LKG image**

kubectl -n <ns> get deploy <deploy> -o jsonpath='{.spec.template.spec.containers[0].image}'

# LKG digest noted from Release evidence: registry.example/mcp-proxy@sha256:<digest>

**Quiesce new traffic & observe**

# Option A: temporarily reduce max surge/unavailable to slow admission

kubectl -n <ns> patch deploy <deploy> -p '{"spec":{"strategy":{"rollingUpdate":{"maxSurge":0,"maxUnavailable":1}}}}'

# Option B: scale down the new RS if mid-rollout (careful with PDB)

**Flip to LKG image**

kubectl -n <ns> set image deploy/<deploy> api=registry.example/mcp-proxy@sha256:<digest>

kubectl -n <ns> rollout status deploy/<deploy> --timeout=5m

**Validate readiness & streaming**

curl -fsS <base>/ready | jq .

# SSE TTFB probe

curl -N -H 'Accept: text/event-stream' \

-H 'Mcp-Session-Id: rollback-check' \

-H 'Content-Type: application/json' \

-w '\nTTFB(ms)=%{time\_starttransfer}\n' \

-d '{"jsonrpc":"2.0","id":"1","method":"ping","params":{"stream":true}}' \

<base>/mcp

**7) Post‑Rollback Verification Checklist**

* /ready=ok; child spawn probe stable; DB reachable (SP execution ok).
* **Latency** p50/p95 (JSON) within budgets.
* **SSE TTFB** p95 ≤ 200 ms; heartbeat cadence nominal (±1 s).
* Error rates back to baseline; no surge of origin\_forbidden/feature\_disabled.
* UI renders Dashboard, Config; no external egress; CSP intact (if portal present).
* **Evidence Pack** updated (snapshots, monitoring images, timeline). Retain ≥ 1 year.

**8) Special Cases & Decision Tree**

**A. Only Origin failures?**  
→ Config Rollback: fix Security:AllowedOrigins via DB; no code rollback.

**B. Child flapping on new image?**  
→ Code Rollback + inspect Mcp:Child\* config; ensure sp\_Config\_GetValue returns expected values.

**C. Legacy endpoints caused instability?**  
→ Disable EnableLegacyHttpSse (DB flag); expect 403 feature\_disabled.

**D. Ingress buffering SSE?**  
→ Fix ingress (disable buffering; extend read/idle timeouts), then re‑test before rolling forward.

**9) RACI (Rollback)**

| **Activity** | **A** | **R** | **C** | **I** |
| --- | --- | --- | --- | --- |
| Decision to rollback | DoSE | SRE Lead | Dev Lead, SecLead | QA, Stakeholders |
| Execute rollback | DoSE | SRE | Dev Lead | QA |
| SSE drain oversight | DoSE | SRE | Dev Lead | QA |
| Evidence capture | DoSE | SRE/CI | DocFactory, QA | All |
| Post‑mortem | DoSE | Dev Lead | SRE, QA, SecLead | All |

**10) Risks & Mitigations**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| Hard kill of streaming pods | Truncated SSE streams; user impact | Enforce readiness‑first drain; extend terminationGracePeriod; verify no ingress buffering |
| Sticky routing misconfigured | Session breakage | Hash/cookie affinity on Mcp‑Session‑Id; verify with test session |
| Parity drift (RTM) | Prod regression later | Re‑run parity checks post‑rollback; block promotion until resolved |
| Secret exposure | Compliance breach | Stop‑the‑line; rotate secrets; sanitize logs; follow incident runbook |

**11) References**

* **Deploy Runbook:** runbooks/deploy.docx (promotion, SSE ingress settings).
* **Incident Runbook:** runbooks/incident.docx (SEV handling, comms, post‑mortem).
* **Scale‑out Runbook:** runbooks/scale\_out.docx (PDB, sticky routing).
* **Compliance:** docs/13\_compliance.docx (CSP/egress, secrets policy; DB/SP rules).
* **Monitoring:** docs/11\_monitoring.docx (SLOs, alerts, 24‑h checks).
* **OpenAPI:** api/openapi/mcp-proxy.yaml (transport, headers, error envelope).

**12) Assumptions**

1. Ingress supports **SSE** without buffering and respects timeouts required for heartbeats.
2. Sticky routing can target the same replica for requests sharing Mcp‑Session‑Id.
3. RTM uses **Prod DB (read‑only)** at all times; rollback must not require DB writes.

**13) Next Steps**

* Automate **rollback scripts** with SSE‑aware drain hooks and Evidence capture.
* Add a **game day** scenario quarterly: simulate streaming regression → practice rollback + post‑mortem.
* Ensure **LKG image digests** are recorded in Release notes/Evidence for quick reuse.

**Footer (optional for Word header/footer):**  
*MCPX‑KendoBridge • Rollback Runbook • v2.0.0 • 2025‑09‑27 • Confidential — Technijian Internal*