**Runbook: Rollback (Rapid Revert to Last Known Good)**

**Project:** MCPX-KendoBridge Admin Portal  
**Runbook ID:** TJ-MCPX-RB-02  
**Version:** 1.1.0  
**Status:** Draft  
**Date:** 2025-09-27  
**Owner:** Release Manager / DevSecOps (Technijian)  
**Confidentiality:** Technijian Internal

**Document Control**

| **Version** | **Date** | **Author** | **Change Summary** | **Status** |
| --- | --- | --- | --- | --- |
| 1.0.0 | 2025-09-26 | RM | Initial rollback procedure | Draft |
| 1.1.0 | 2025-09-27 | RM | Added SSE/Parity triggers, add-only DB guidance, evidence updates | Draft |

**Approvals**

| **Role/Team** | **Name** | **Signature/Date** | **Comment** |
| --- | --- | --- | --- |
| Director of Software Eng. |  |  |  |
| Systems Architect |  |  |  |
| DevSecOps / SRE |  |  |  |
| Security & Compliance |  |  |  |
| QA Lead |  |  |  |

**1) Purpose**

Provide a fast, safe, **one-button mindset** to revert API/Web to the **last known good (LKG) release** when a deployment causes **P1/P2 impact**, while preserving auditability and avoiding destructive DB changes (add-only schema policy).

**2) Scope**

* **API (.NET 8)** and **Web (React + KendoReact Fluent 2)** rollback
* **Config/Flags** rollback via SPs (non-secret runtime config only)
* **DB schema**: *roll-forward only* (no destructive DDL). Data hotfixes are done via new patch migrations.
* Environments: **Alpha, Beta, RTM, Prod**

**3) When to Roll Back (Triggers)**

Roll back immediately if any of the below occur after a deploy:

* **Availability/Readiness**: /healthz or /ready failing > 5 minutes, or flapping > 3 times / 10 minutes.
* **Error Rate**: 5xx > **1%** for 5 minutes (Prod) or sustained spike beyond normal in lower envs.
* **Latency**: p95 latency breach sustained (JSON > 600 ms for 10 minutes).
* **SSE**: First-event **TTFB > 200 ms** median for 10 minutes, or **heartbeat gap > 15 s** on any instance.
* **CORS**: Widespread origin\_forbidden from intended origins after an allow-list change.
* **Parity (RTM only)**: Promotion bypassed policy or critical drift discovered post-deploy.
* **Security**: Unintended exposure (policy breach) or secret leak suspicion.
* **Business**: Material regression on core admin workflows (config/flags/jobs/audit).

If in doubt, **roll back first**, investigate after. Rollback is reversible; customer trust is not.

**4) Roles & Communications**

* **Incident Commander (IC):** Release Manager (RM)
* **Deputy:** SRE on call
* **Decision Owner:** Director of Software Engineering
* **Comms:** #release and #incident Slack channels; status note to stakeholders

**Start message template (Slack):**

Declaring **rollback** in **{env}** for MCPX-KendoBridge. Current release {tag\_current} failing due to {reason}. Reverting to LKG {tag\_prev} now. IC: @{RM}, Deputy: @{SRE}.

**5) Decision Tree**

1. **Is the issue config/flag-related?**  
   → **Yes**: Revert config/flag via SPs (Section 7) and validate. If stable, no full rollback needed.  
   → **No**: Continue.
2. **Is it Web-only (UI regression) with stable API?**  
   → Roll back **Web** to LKG. Validate and stop.  
   → Otherwise continue.
3. **API regression / contract break / SSE failure?**  
   → Roll back **API** (and Web if tightly coupled) to LKG.
4. **DB migration caused issue?**  
   → **Do not** drop columns/tables. Apply **hotfix forward** migration to remediate. Roll back app to LKG while preparing hotfix.

**6) Rollback Steps (API/Web)**

Prereqs: Previous **LKG release tag** exists with full **Evidence Pack**.

**Step 1 — Freeze & Announce**

* Pause traffic shifting / further deploys.
* Announce rollback start.

**Step 2 — Restore API to LKG**

* Deploy prior API artifact (release {tag\_prev}) to the target environment (slot swap or rolling)
* Wait for **/healthz** and **/ready** green.

**Step 3 — Restore Web to LKG (if required)**

* Publish prior static build ({tag\_prev}) to hosting.
* Run synthetic: login (MSAL) → /dashboard renders; env badge correct.

**Step 4 — Validate**

* Contract tests on core endpoints.
* **SSE smoke** (k6\_sse\_ttfb.js): **TTFB ≤ 200 ms**, heartbeat **≤ 10 s**.
* UI spot checks: /config, /flags, /jobs, /evidence.

**Step 5 — Evidence & Comms**

* Update the **incident** with start/end time, release IDs.
* Append rollback note to current release; cross-link LKG Evidence Pack.
* Announce success with metrics snapshot.

**Success message template:**

Rollback complete in **{env}**. Reverted to {tag\_prev}. Health/Ready ✅. SSE TTFB {ms}, heartbeat {s}. Investigating root cause on {tag\_current}.

**7) Config / Feature Flag Rollback (No App Rollback)**

Sometimes only config or flags caused the issue. Use **SP-only** mutations (audited):

**Disable a risky feature flag**

EXEC dbo.sp\_Feature\_Set

@FlagKey = N'EnableLegacyHttpSse',

@IsEnabled = 0,

@Scope = N'{Alpha|Beta|RTM|Prod}',

@TargetRole = NULL,

@Description = N'Emergency disable after regression',

@Actor = N'rollback-bot',

@RequestId = N'{requestId}';

**Revert a config key**

EXEC dbo.sp\_Config\_SetValue

@ConfigKey = N'Network:SseKeepAliveSeconds',

@Value = N'10',

@ValueType = N'seconds',

@Scope = N'{Alpha|Beta|RTM|Prod}',

@Description = N'Rollback to prior stable',

@Tags = N'network,sse',

@Actor = N'rollback-bot',

@RequestId = N'{requestId}';

All changes write **AuditEvent** (who/what/when/before→after). Capture diffs as part of the incident record.

**8) Database Guidance (Add-Only Schema)**

* **Never** attempt to “roll back” schema by dropping columns/tables.
* If a migration breaks behavior, **roll back app** to LKG and create a **forward hotfix migration** (e.g., add a new column, default value, or view).
* Keep API principal **EXECUTE-only**; do not grant table DML to fix data—write a controlled SP or ad-hoc script reviewed by DBA/SEC.

**9) Parity & RTM Special Case**

* If drift is detected in **RTM** post-deploy (should be blocked by gate, but if discovered):
  + Generate **parity report**.
  + **Reapply** intended Prod config via SPs or correct the **expected** file, then re-run parity until **0 critical**.
  + If app behavior still regresses, roll back to LKG and re-test parity.

**10) Post-Rollback Actions**

* **Stabilize**: watch dashboards (p95, error %, **SSE** first-event trend, heartbeat gaps) for 30–60 minutes.
* **Root Cause Analysis (RCA)**: open a postmortem with:
  + Failing version {tag\_current} vs {tag\_prev}
  + Key metrics before/after
  + Log excerpts with **requestId** and error envelope codes
  + Config/flag diffs and any SP signature changes
  + Decision timeline (times, approvers)
* **Follow-ups**: hotfix migration, test additions (E2E/perf), documentation updates.

**11) Evidence (attach to current release)**

* Rollback decision note (reason, timestamps, env, approvers)
* Health/Ready and **SSE smoke** outputs after rollback
* Metric screenshots (Executive, **SSE Health**)
* Any config/flag audit entries (before→after)
* Links to LKG **Evidence Pack** and current incident ticket

Retention: **≥ 1 year**.

**12) Runbook Snippets (Quick Reference)**

**SSE Smoke (k6):**

k6 run tests/perf/k6\_sse\_ttfb.js \

-e BASE\_URL=https://{env-host}/api \

-e TOKEN={bearer}

**Contract Smoke (example curl):**

curl -sS -H "Authorization: Bearer $TOKEN" https://{env-host}/api/healthz

curl -sS -H "Authorization: Bearer $TOKEN" https://{env-host}/api/config/effective

**Flag safety (Legacy SSE OFF):**

EXEC dbo.sp\_Feature\_Set @FlagKey=N'EnableLegacyHttpSse', @IsEnabled=0, @Scope=N'Prod',

@TargetRole=NULL, @Description=N'Keep legacy disabled', @Actor=N'rollback-bot', @RequestId=N'{requestId}';

**13) Acceptance Criteria (Rollback)**

* Prior **LKG** restored (API/Web) within the agreed target (aim ≤ 30 minutes).
* Health/Ready ✅; **SSE** TTFB and heartbeat back within SLOs.
* No lingering p95/5xx alerts after 30–60 minutes.
* **AuditEvent** entries exist for any config/flag changes.
* Evidence updated; incident captured with RCA plan.

**14) Open Issues**

* Automate one-click rollback pipeline step with built-in SSE smoke and post-checks.
* Store per-env “LKG pointer” to accelerate target selection.
* Add contract smoke to rollback pipeline as a blocking check.

**End of Runbook — TJ-MCPX-RB-02 v1.1.0**