**Runbook: Scale-Out (Capacity & Latency Response)**

**Project:** MCPX-KendoBridge Admin Portal  
**Runbook ID:** TJ-MCPX-RB-05  
**Version:** 1.0.0  
**Status:** Draft  
**Date:** 2025-09-27  
**Owner:** DevSecOps / SRE (Technijian)  
**Confidentiality:** Technijian Internal

**Document Control**

| **Version** | **Date** | **Author** | **Change Summary** | **Status** |
| --- | --- | --- | --- | --- |
| 1.0.0 | 2025-09-27 | SRE | Initial scale-out procedure + validation/rollback | Draft |

**Approvals**

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

**1) Purpose**

Provide a **safe, incremental** procedure to increase capacity and reduce latency/error rates when demand or workloads exceed targets. Includes **pre-checks, execution**, **validation**, and **rollback to prior capacity**.

**2) Scope**

* **Admin API (.NET 8)** instances (Windows Server 2022 VMs or app hosts)
* **Admin Web (React build)** hosting tier (instances/edge cache)
* **Child MCP worker processes** (STDIO) per API host
* **SQL Server 2022** (scale-up; read-intent replicas optional)
* **LB/DNS** updates, health probes, CORS allow-list, observability

**3) SLO Targets (Prod)**

* **Availability** ≥ 99.9% (API/Web)
* **JSON read p95** ≤ **300 ms** (p99 ≤ 600 ms)
* **JSON write p95** ≤ **500 ms** (p99 ≤ 900 ms)
* **/healthz p95** ≤ 150 ms
* **SSE TTFB** ≤ **200 ms**; **heartbeat gap** ≤ **10 s**
* **5xx error rate** ≤ **0.5%**

Scale-out is considered **successful** when the above are met for **≥ 30 minutes** post-change without alert flaps.

**4) Preconditions & Safety Checks (Blocking)**

* ✅ Current **deploy** green (Runbook RB-01), no active **incident** P1/P2 (RB-03)
* ✅ **Parity** (RTM↔Prod) has **0 critical diffs** (RTM only)
* ✅ **Audit/Change ticket** opened with objective, target env, and rollback plan
* ✅ **CORS allow-list** covers all client origins behind the LB/edge
* ✅ Dashboards live: Executive, API Perf, **SSE Health**, Readiness
* ✅ k6 **SSE smoke** script path confirmed (tests/perf/k6\_sse\_ttfb.js)

**5) Triggers (When to Scale)**

Scale-out if any persist ≥ 10–15 minutes (after confirming no incidents/regressions):

* p95 (reads/writes) above target; or rising **queue depth** / **worker saturation**
* **SSE** first-event drifting > 200 ms or heartbeat gaps > 10 s
* **5xx** trending upward; threadpool/exhaustion symptoms
* New load profile (feature launch, traffic ramp) planned

**6) What We Can Scale**

1. **API Tier (horizontal)** — increase instance count **N → N+1(+2)**
2. **API Host (vertical)** — increase vCPU/RAM; tune Kestrel/concurrency (post-change validation mandatory)
3. **Child MCP Workers** — raise **max child processes per host** cautiously (watch CPU and SSE TTFB)
4. **Web Tier** — add instances / enable edge cache for static assets
5. **SQL Server**
   * **Scale-up** (vCPU/RAM/IO)
   * Optional **read-intent replica** for eligible reads; ensure connection strings and SPs permit it
6. **LB/Probe** — widen healthy target pool; confirm health checks; keep slow start enabled

**Rule of thumb:** Prefer **horizontal** changes first; make **one change class at a time**, validate, then proceed.

**7) Step-by-Step Procedure (Incremental)**

**Step 1 — Announce & Snapshot (5 min)**

* Post in #release: scale-out start, env, current instances, metrics snapshot (p95/5xx/SSE).
* Enable **slow start/warm-up** on LB to avoid traffic spikes to new nodes.

**Step 2 — API Horizontal Scale (preferred first)**

1. Increase API instances by **+1** (or +20% if N ≥ 5).
2. Wait for **/healthz** and **/ready** green on new nodes.
3. Verify **CORS** pass from intended origins.
4. **Observe 10 minutes**: p95, 5xx, **SSE TTFB** & heartbeat, CPU/RAM.

**If improved & stable:** consider another +1 iteration.  
**If no improvement or worse:** revert (Step 10) and assess vertical/DB options.

**Step 3 — Child MCP Worker Concurrency (optional)**

* Increase **max child processes per host** by **+1** (e.g., 2 → 3).
* Verify CPU headroom ≥ 30% and **SSE TTFB** stays ≤ 200 ms.
* Observe 10 minutes; if stable, keep; else revert.

**Step 4 — Web Tier / Static Delivery**

* Add **+1** web instance or ensure CDN/edge caching headers for static assets (immutable build).
* Validate: login → /dashboard TTI p95 ≤ 2.0 s.

**Step 5 — API Vertical Scale (if needed)**

* Scale host SKU (vCPU/RAM) **up one notch**.
* Validate threadpool growth, Kestrel connections, GC not thrashing.
* Observe 10 minutes.

**Step 6 — SQL Server (only if DB-bound)**

* **Scale-up** compute/IO; confirm MAXDOP/TempDB baseline ok.
* Optionally configure **read-intent** for safe read paths (if app supports ApplicationIntent=ReadOnly).
* Run small **read-heavy** probe (Config/Audit queries) and watch Query Store.

**Step 7 — Rebalance & Health**

* Confirm LB distribution evenness; no hot node.
* Ensure all **probes** pass and **slow start** cool-down completed.

**Step 8 — Validation (Mandatory)**

Run the following and capture outputs:

* **k6 SSE Smoke** → **TTFB ≤ 200 ms**, **heartbeat ≤ 10 s**
* Contract smokes: /healthz, /ready, /config/effective
* UI smokes: login → /dashboard cards; /jobs stream; /flags toggle

**Step 9 — Observe (30–60 min)**

* p95 (read/write), 5xx, **SSE metrics**, CPU/RAM, GC pauses, queue depth
* Ensure alerting quiet and no flapping

**Step 10 — Rollback to Prior Capacity (if needed)**

* Reduce added instances (API/Web) to previous N; restore worker limits
* Revert DB/host SKU change if made
* Re-run validation (Step 8)
* If still degraded → **Runbook RB-02 (Rollback)** to LKG release

**8) Post-Change Tasks**

* Update **capacity notes** in TREE.md/ops README (optional)
* Attach validation results to the **Evidence Pack** (release)
* Close change ticket with before/after metrics and screenshots

**9) Monitoring Focus & Alerts (during/after scale-out)**

| **Signal** | **Expectation** | **Action if breached** |
| --- | --- | --- |
| JSON p95 | ↓ into targets within 10–15 min | Add another node or rollback |
| 5xx | No rise; ideally ↓ | Investigate hot node/logs; rollback if sustained |
| **SSE TTFB** | ≤ 200 ms (median) | Reduce MCP workers / add API node / rollback |
| **SSE heartbeat gap** | ≤ 10 s; alert at > 15 s | Investigate worker/network; rollback worker bump |
| CPU per API node | ≤ 70% steady | Add node / scale up |
| SQL read/write latency | Stable or ↓ | Scale up DB / review hot queries |
| LB health/slow start | All nodes healthy; slow start complete | Pause further changes until stable |

**10) Rollback Plan (Quick)**

1. Announce rollback of capacity to prior state.
2. Remove added API/Web nodes; restore worker limits; revert host/DB SKU changes.
3. Validate health, **SSE smoke**, UI smokes.
4. If app behavior remains degraded → **RB-02 Rollback** to LKG release.

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

* Before/after **dashboard screenshots** (Executive, API Perf, **SSE Health**)
* k6 SSE results (TTFB/heartbeat)
* Contract/UI smoke outputs
* Change ticket link; summary of instance counts and timings

Retention: **≥ 1 year**.

**12) Quick Commands (reference)**

**SSE Smoke:**

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

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

-e TOKEN={bearer}

**Health/Config:**

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

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

**13) Acceptance Criteria (Scale-Out)**

* Scale change applied **one class at a time** with 10–15 min stabilization between steps
* **SSE TTFB ≤ 200 ms**, heartbeat ≤ 10 s; JSON p95 within SLOs; 5xx not elevated
* No alert flapping; metrics stable for **≥ 30 minutes**
* Evidence attached; change ticket closed with results
* Rollback performed if improvement not achieved or regressions observed

**14) Open Issues**

* Automate capacity plans (HPA/VPA equivalent) with safeguards
* Add synthetic load profiles per route to better predict saturation
* Consider read-intent path for **non-mutating** config queries if/when app supports it

**End of Runbook — TJ-MCPX-RB-05 v1.0.0**