

CI/CD Pipeline — Deploy Path Validated (2026-07-07)

CI/CD Pipeline — Deploy Path Validated (2026-07-07)

MC: #104863

Status: ☐ COMPLETE — First fully green end-to-end demo deploy run

Evidence: https://dev.azure.com/alai-holding/QODY/_build/results?buildId=272

Summary

QODY's Azure DevOps CI/CD pipeline is now **fully validated** end-to-end. Build #272 (2026-07-07 00:22 UTC) was the first run to successfully complete all 4 stages:

1. **CI Gates** — Backend tests (Java 21 + Gradle) + frontend builds (npm ci + TypeScript/Vite) + Gitleaks secret scan
2. **Build images → ACR** — 4 Docker images (qody-api, qody-guest, qody-admin, qody-staff-kitchen) built and pushed to qodydemoacr.azurecr.io
3. **Security: Trivy image scan** — All 4 images scanned (HIGH + CRITICAL severity, exit-code 1 on findings) — **CLEAN** (jackson-databind 2.18.8, netty-bom 4.1.135.Final, Alpine patched)
4. **Deploy → QODY demo ACA** — Canary deploy (qody-api: 0%→smoke curl→100%) + single-revision updates (guest/admin/kitchen) + **flagship pay-to-kitchen UAT** (Playwright hard gate) — **PASS**

Live verification (John, 00:24 UTC):

- <https://demo.api.qody.ba/health> → `200`
`{"status": "ok", "db": {"rlsRoleCheck": {"bypassRls": false, "status": "PASS"}}`
 - demo.app/admin/kuhinja.qody.ba → All `200` with non-empty HTML
-

Root Cause (Why Pipeline Was Dead)

The Real Problem (tool-verified):

The `azure-qody` Azure RM service connection (and `qody-demo` ADO environment) had **never been explicitly authorized** for use by pipeline definition #3 ("QODY CI-CD"). Azure DevOps resource protection requires this one-time authorization the first time a pipeline run reaches a job that references the resource.

Every previous run either:

- Had `deployDemo=false` (never reached `Build_Images` stage), OR
- Was a tag run (e.g., `v1.7.0-onboarding-rc1`) that **silently stalled forever** at `Checkpoint.Authorization` (no error surfaced, just `status: inProgress` indefinitely)

Fixed via:

```
PATCH .../pipelines/pipelinePermissions/endpoint/62c293a8-2241-465a-a9ac-93af127472de
PATCH .../pipelines/pipelinePermissions/environment/4
body: {"pipelines":[{"id":3,"authorized":true}]}
```

Both returned `authorized: true` immediately. The stuck `Checkpoint.Authorization` record transitioned `inProgress` → `completed` / `succeeded` within the same second — direct, unambiguous proof.

Initial hypothesis (tag/branch trigger mismatch) was a red herring: The tag run *did* fire (build #236 exists, its `CI_Gates` stage succeeded), it just hung afterward on the authorization gap. The tag/branch filter theory is likely correct as a matter of ADO semantics, but it was NOT the blocker.

Path to Green (5 Runs, 5 Real Bugs Fixed)

Each failed run exposed a genuine latent issue:

Run	Failure	Root Cause	Fix
-----	---------	------------	-----

#253	Build_Images failed	Missing apps/guest/nginx.default.conf + apps/staff-kitchen/nginx.default.conf (referenced by Dockerfile COPY lines since MC #104749, but never committed to git)	Committed the 2 missing .conf files (HSTS + security headers, identical pattern to admin)
#265	Security_Image_Scan failed	16 HIGH CVEs in jackson-databind + 2 HIGH in p11-kit	Bumped jackson-databind → 2.18.8, netty-bom → 4.1.135.Final, apk upgrade in Dockerfiles
#267	Security_Image_Scan failed	14 HIGH CVEs in netty (after jackson fix)	Explicit netty-bom override in build.gradle.kts
#269	Deploy_Demo flagship UAT failed	deploy_demo job used checkout: none → \$(Build.SourcesDirectory)/e2e didn't exist for Playwright test script	Added checkout: self before UAT step
#272	🟢 SUCCESS	All stages green	N/A

Each failure was a real bug (not flakiness), and each fix made the system more correct.

CVE Fixes (Image Security)

Before: 16 HIGH jackson-databind CVEs + 14 HIGH netty CVEs + 2 HIGH p11-kit Alpine CVEs
After (run #272): CLEAN — Trivy scan returned exit-code 0 for all 4 images

Dependency bumps:

- jackson-databind → 2.18.8 (latest, no HIGH/CRITICAL CVEs)
- netty-bom → 4.1.135.Final (all netty-* artifacts now inherit this version, CVEs resolved)
- apk upgrade in all Dockerfiles (p11-kit 0.25.6-r0 → patched)

Production Deploy Path (Prepared, Not Yet Live)

Service Principal: gody-azdo-pipeline-prod (separate identity from demo's gody-azdo-pipeline)
Authentication: Workload Identity Federation (no static secret)
Scope: Least-privilege — Azure Container Apps Contributor on rg-gody-prod (not full Contributor), AcrPush on qodyprodacr only

Securion Verdict: ☐ **PASS** (v2 proposal, 2 blocking findings fixed, 4 required verifications folded in)

Deploy_Prod stage: Drafted at `infrastructure/pipeline-proposals/deploy-prod-stage.yml.proposal` (deferred from live `azure-pipelines.yml`) — includes:

- Separate prod images (`prod-${SHORT_SHA}` in `qodyprodacr`, different build-args from demo)
- Trivy scan for prod images (inside `Deploy_Prod` stage, before canary/update steps)
- Manual-only trigger (`Build.Reason: Manual`, not auto-trigger on tag push)
- ADO environment `qody-prod` with **manual approval** (approver: `alem@alai.no`) + **branch control** (main-only)

Status: Proposal only, NOT executed. Activation = separate task requiring CEO approval.

Full proposal (with Securion review history): `/Users/makinja/system/evidence/104863/d2-prod-sp-proposal.md`

Deploy Path Truth (As of 2026-07-07)

Path	Status	Evidence
Manual run with <code>deployDemo=true</code>	☐ PROVEN end-to-end	Build #272 (all stages green, flagship UAT PASS)
Semver tag <code>v*</code> from <code>main</code>	☐ Analytically sound	Tag/branch ancestry checked + trigger semantics verified, but NOT empirically confirmed (no executed tag run post-fix). Recommend: watch the first real <code>v*</code> tag deploy as final validation.

Caveat (Proveo): Run #272 was triggered `reason: manual` with `templateParameters: {deployDemo: "true"}` — NOT a `v*` tag push. The tag-trigger mechanism itself was deliberately not re-tested with a second live deploy (to honor the task's "ONE live validation deploy" constraint). Tag trigger proof rests on static evidence, not an executed tag run.

Evidence Location

Directory: `/Users/makinja/system/evidence/104863/`

Key files:

- `d1-run272-success.md` — Build #272 timeline + live verification curls

- `d1-root-cause-and-fix.md` — Full narrative: hypothesis evolution, ADO API proofs, safety actions (cancelled stale runs)
 - `d1-run253-failure.md`, `d1-run265-failure.md`, `d1-run269-failure.md` — Each failure's root cause + demo-health curl (proving no rollback needed)
 - `d1-cve-bump.md` — CVE details + dependency versions
 - `proveo-validation.md` — Independent Proveo (Angie Jones) verification against primary sources (ADO REST API, live curls, git blobs)
 - `d2-prod-sp-proposal.md` — Production SP proposal (Securion PASS, not yet executed)
 - `securion-sp-verdict.md` — Securion review history (v1 BLOCK → v2 PASS)
-

Guardrails Added

1. **RUNBOOK.md D3 guardrail (committed to main):** Every QODY deploy task (demo or prod) MUST cite the Azure DevOps **pipeline run URL** in MC evidence. `mc.js done/ready` on a QODY deploy task without a run URL is incomplete evidence (ZAKON PI2).
 2. **Revision-suffix collision fix (committed to main):** `Deploy_Demo`'s `qody-api` canary step used `REV_SUFFIX="$(SHORT_SHA)"` — a redeploy of the exact same commit would collide with the existing ACA revision name and fail. Fixed to `REV_SUFFIX="$(SHORT_SHA) - $(Build.BuildId)"` — unique per pipeline run regardless of `SourceVersion`. Mirrored into the deferred `Deploy_Prod` proposal.
-

Next Steps

1. **First real `v*` tag from `main`:** Watch as empirical confirmation of the tag trigger path (analytically sound, but not yet executed post-authorization-fix).
 2. **Prod deploy activation:** Separate task, requires CEO approval. Execute `d2-prod-sp-proposal.md` commands, enable `Deploy_Prod` stage in `azure-pipelines.yml`, verify manual-only + approval gate.
 3. **CI contention (optional, out-of-scope for #104863):** The `alai-holding` ADO org shares Microsoft-hosted parallel-job capacity across QODY/Bilko/LumisCare. Consider requesting additional parallelism or self-hosted agents.
-

Last Updated: 2026-07-07

Author: Skillforge (documentation) + FlowForge (pipeline execution) + John (orchestration)

Proveo Verdict: ☐ PASS (all 4 claims verified against primary sources)

Revision #1

Created 2026-07-07 00:40:57 UTC by John

Updated 2026-07-07 00:40:57 UTC by John