Bilko CI/CD — Stage?Prod Pipeline (MC #99477)

Overview

Stage pipeline: push-main → bilko-stage-auto-deploy → cloudbuild-stage.yaml → bilko-{web,api}-stage

Prod pipeline: tag v* → bilko-main-deploy → cloudbuild.yaml → bilko-{web,api}

Stage pipeline is optimized for **FAST FEEDBACK** — no quality gates. Prod pipeline has 8 production gates including SHA verification, Trivy scanning, Flyway migrations, and Cloud Build native approval.

Stage Pipeline

Step	Purpose	Image Tag	Duration (avg)
sanity-check	Verify Docker socket + Artifact Registry reachability (environment health, NOT a quality gate)	—	~2.3s
build-web	Build Next.js app with docker buildx (apps/web/Dockerfile)	:stage-${SHORT_SHA} :stage-latest	~3m
push-web	Push image to Artifact Registry (europe-north1-docker.pkg.dev/tribal-sign-487920-k0/bilko/web)	—	~7s
migrate-db	Run Flyway migrations against Cloud SQL bilko-staging-db (POSTGRES_16) via Cloud SQL proxy	—	~22s
deploy-web-stage	Deploy bilko-web-stage Cloud Run service with :stage-${SHORT_SHA} image, --no-traffic	—	~39s

Step	Purpose	Image Tag	Duration (avg)
promote-web-stage	Route 100% traffic to new revision (no canary for stage)	—	~10s
deploy-api-stage	Deploy bilko-api-stage (redeploys EXISTING image only — no API build step, see OCD-1)	—	~19s
smoke-test	curl -sf https://bilko-api-stage-dh4m46blja-lz.a.run.app/api/v1/health — exit 1 if non-200	—	~2.5s

Total duration: ~5 minutes (build 6f2236f6, validated 2026-05-06)

Prod Pipeline

Existing prod pipeline (cloudbuild.yaml) has 8 gates and MUST NOT be rewritten. References:

- SHA verification (Git commit SHA in image metadata)
- Trivy vulnerability scanning
- Flyway migration validation
- Cloud Build native approval (approval_required=true in modules/build/main.tf)
- Smoke tests (health endpoint + web homepage)
- Gradual traffic rollout (0% → 100%)
- Rollback on smoke test failure

Prod pipeline is BLOCKED on OCD-5 (bilko-db Cloud SQL instance does not exist — requires CEO approval for provisioning).

Triggers

Trigger Name	Filename	Branch/Tag	Approval	Service Account
bilko-stage-auto-deploy	infrastructure/gcp/cloudbuild-stage.yaml	^main$	No (auto-deploy)	762788903040@cloudbuild.gserviceaccount.com
bilko-main-deploy	infrastructure/gcp/cloudbuild.yaml	v* (semver tag)	Yes (Cloud Build UI)	762788903040@cloudbuild.gserviceaccount.com

GCP project: `tribal-sign-487920-k0`, region: `europe-north1`

Open Risks — 5 CEO Decisions Required

These items require CEO judgment and are NOT resolved in this implementation:

OCD-1: bilko-api Build Pipeline Gap

Status: OPEN — BLOCKER for API continuous delivery

Current state: bilko-api-stage is live and serving traffic at https://bilko-api-stage-dh4m46blja-lz.a.run.app/api/v1 with image `api:stage-b7e8a59`. No Cloud Build pipeline exists for the Kotlin/Ktor API. Dockerfile path unconfirmed.

Impact: Stage cloudbuild-stage.yaml `deploy-api-stage` step redeploys the EXISTING API image only — cannot build new API images. API deployments must be manual via `gcloud run deploy` until resolved.

CEO decisions needed:

1. What is the canonical Dockerfile path for apps/api?
2. Should API have its own Cloud Build step in cloudbuild-stage.yaml or a separate trigger?
3. Is bilko-api currently deployed manually via `gcloud run deploy`?

OCD-2: Stage Hostname — bilko-stage.alai.no vs Raw .run.app URL

Status: OPEN — affects CORS configuration

Current state: ENV-MATRIX.md CORS_ORIGINS for staging references `staging.bilko.io` (STALE). terraform.tfvars stage_api_url points to raw `.a.run.app` URL. Stage pipeline uses raw .run.app URL as default.

Impact: Frontend CORS errors if staging.bilko.io DNS is ever pointed at stage services.

CEO decision needed: Should `bilko-stage.alai.no` be the canonical stage hostname? If yes: Cloudflare DNS entry (manual — not in Bilko TF stack) + CORS_ORIGINS update required via separate MC.

OCD-3: Postgres Version Mismatch — Stage POSTGRES_16 vs Prod POSTGRES_15

Status: OPEN — CRITICAL for financial data integrity

Current state: bilko-staging-db runs POSTGRES_16 (confirmed live). envs/prod/main.tf line 94 specifies POSTGRES_15 for prod (bilko-db does not exist yet — see OCD-5). Stage validates migrations and queries against PG16; prod would run PG15.

Impact: For a financial accounting SaaS, stage validation on PG16 while prod runs PG15 invalidates the "stage-as-test-environment" premise. Schema compatibility unverified. SQL dialect differences (PG15→PG16) may surface as prod-only bugs.

CEO decision needed: Upgrade prod to POSTGRES_16 (requires maintenance window, pg_upgrade or dump/restore) OR downgrade stage to POSTGRES_15? ALAI standard tech stack (ALAI/CLAUDE.md) mandates POSTGRES_16 for all products, suggesting prod config is non-compliant.

OCD-4: Stage ? Prod SHA Promotion Strategy

Status: OPEN — architectural decision

Current state: Prod trigger fires on semver tag push, rebuilds from source. Stage-validated image digest is NOT carried to prod build. Stage tests one SHA and prod deploys a different build. If a hot dependency updates between stage build and prod build (e.g., npm registry serves new patch version), stage and prod can diverge on identical Git SHAs.

CEO decision needed:

1. **Option A:** Accept rebuild-on-tag (simpler, current model) with acknowledgment of hot-dependency risk.
2. **Option B:** Implement digest promotion where prod trigger accepts an image digest input parameter and skips rebuild. Requires Cloud Build trigger API call from a promotion script or Google Cloud Deploy.

OCD-5: Prod Cloud SQL bilko-db Existence

Status: OPEN — BLOCKER for prod terraform apply

Current state: `gcloud sql instances list --project=tribal-sign-487920-k0` shows ONLY bilko-staging-db. No bilko-db (prod) exists. envs/prod/main.tf explicitly notes "bilko-db (prod) — TBD — audit required" (lines 4-6 and import.sh).

Impact: Any `terraform apply` on envs/prod would attempt to create a REGIONAL HA POSTGRES_15 `db-custom-2-7680` instance (~$100+/month). Without CEO sign-off, prod infra is BLOCKED.

CEO decision needed: Approve prod DB provisioning (cost + data migration strategy if migrating from elsewhere) before ANY envs/prod TF apply is ever run. If bilko-db exists elsewhere (on-prem?

Railway?), import.sh must be run first.

Validation

Evidence file: `/tmp/99477-proveo-evidence.md`

Build ID: 6f2236f6-86ec-444c-96b7-7c22f63cf5a2

Build log: View in GCP Console

Validation date: 2026-05-06T20:28Z

Validator: Angie Jones (Proveo)

Verdict: PASS — 7/7 Acceptance Criteria met

- AC1: Build SUCCESS (all 8 steps SUCCESS)
- AC2: bilko-web-stage HTTP/2 200
- AC3: bilko-api-stage health endpoint 200 {"status":"ok"}
- AC4: New web revision deployed within 5min window
- AC5: Flyway migrate-db ran without error (21.5s)
- AC6: No gate-* steps executed (0 quality gates)
- AC7: Image pushed with :stage-${SHORT_SHA} tag (stage-277dd5a confirmed in Artifact Registry)

Related MCs

- **#99395** — VAT enum-cast genesis (billing_country ENUM cast to TEXT in Flyway migration)
- **#99422** — Sibling task (stage Cloud Run services health check)
- **#99477** — This task (Stage CI/CD pipeline implementation)

ZAKON PI2 Compliance Status

Stage pipeline: COMPLIANT

- DEPLOY-MAP.md exists at repo root
- Pre-flight checks executed (4 probes: triggers, GCS bucket, Cloud Run services, SQL instances)
- Post-deploy validation (curl 200 + Cloud Run revision evidence)
- Evidence files delivered (/tmp/99477-preflight.txt, /tmp/99477-proveo-evidence.md)

Prod pipeline: □ BLOCKED (awaiting OCD-5 bilko-db provisioning approval)

Last Updated

2026-05-06, owner: FlowForge (Kelsey Hightower)

Revision #2
Created 2026-05-06 20:34:12 UTC by John
Updated 2026-06-07 20:01:20 UTC by John