Ikusi

Finanzas API CI/CD Hardening - Implementation Summary

Finanzas SD Documentation / Documentación de Finanzas SD

Finanzas SD Documentation
Documentación de Finanzas SD

Ikusi
November 10, 2025

1 Finanzas API CI/CD Hardening - Implementation Summary

1.1 Overview

This implementation addresses all issues identified in the problem statement for PR #21, ensuring reliable CI/CD workflows for the Finanzas API.

1.2 Changes Implemented

1.2.1 1. Deploy Workflow (`deploy-api.yml`)

Issues Fixed:

- Replaced local OIDC action path with official `aws-actions/configure-aws-credentials@v4`
- Added sensible defaults for AWS_REGION, FINZ_API_STACK, FINZ_API_STAGE
- Added COGNITO_USER_POOL_ID and COGNITO_USER_POOL_ARN to environment and preflight check
- Added npm ci step to install dependencies (including esbuild)
- Set PATH to include node_modules/.bin for esbuild access
- Enhanced parameter overrides formatting
- Improved summary output with comprehensive curl examples

Key Improvements:

```
[] # Before: Local action (would fail if not present) uses:./.github/actions/oidc-configure-aws
# After: Official AWS action uses:aws-actions/configure-aws-credentials@v4
```

New Steps:

- Install dependencies step added before SAM build
- PATH configuration for esbuild in build step
- Comments clarifying required vs optional variables

1.2.2 2. Test Workflow (`test-api.yml`) - NEW

Purpose: Validates code quality before merging to main

Features:

- Runs on pull requests to main branch
- Manual dispatch trigger available
- Installs SAM CLI
- Runs unit tests (npm ci && npm test)
- Builds with SAM
- Starts SAM local and performs smoke tests on public endpoints
- Generates success summary

Endpoints Tested:

- `GET /health` - Expects 200 OK
- `GET /catalog/rubros` - Expects 200 OK with rubros data

1.2.3 3. Documentation (`WORKFLOW_SETUP.md`) - NEW

Contents:

- Complete workflow descriptions
- Required variables and secrets (with example values)
- OIDC authentication explanation
- API endpoints status (implemented vs stubbed)
- Local testing instructions
- Comprehensive troubleshooting guide

1.3 Configuration Requirements

1.3.1 Repository Variables (Required)

Set in: **Settings Secrets and variables Actions Variables**

Variable	Example Value	Has Default?
AWS_REGION	us-east-2	Yes
FINZ_API_STACK	finanzas-sd-api-dev	Yes
FINZ_API_STAGE	dev	Yes
COGNITO_USER_POOL_ID	us-east-2_FyHLtOhiY	No - Required
COGNITO_USER_POOL_ARN	arn:aws:cognito-idp:	No - Required
COGNITO_USER_POOL_CLIENT_ID	dshos5iou44tuach7ta3ici5m	No - Required

1.3.2 Repository Secrets (Required)

Set in: **Settings Secrets and variables Actions Secrets**

- `OIDC_AWS_ROLE_ARN` - IAM role ARN for OIDC authentication

1.4 API Implementation Status

1.4.1 Implemented (5 endpoints)

- `GET /health` - Health check (public)
- `GET /catalog/rubros` - Budget categories (public)
- `POST /projects` - Create project (authenticated)
- `GET /projects` - List projects (authenticated)
- `POST /projects/{id}/handoff` - Project handoff (authenticated)

1.4.2 Stubbed (9 endpoints returning 501)

- `POST/GET /projects/{id}/rubros`
- `PUT /projects/{id}/allocations:bulk`
- `GET /projects/{id}/plan?mes=YYYY-MM`
- `POST /payroll/ingest`
- `POST /close-month?mes=YYYY-MM`
- `POST/GET /adjustments`
- `GET /alerts`

- `POST/GET /providers`
- `POST/GET /prefacturas/webhook`

All stubbed handlers contain TODO comments with implementation guidance.

1.5 Testing Results

1.5.1 Local Tests

```
cd services/finanzas-api
npm ci
npm test

 4 tests passed
  - pro-rata forward split
  - pro-rata 3-month split with rounding
  - coverage vs payroll
  - coverage percentage calculation
```

1.5.2 SAM Build

```
cd services/finanzas-api
export PATH="$PWD/node_modules/.bin:$PATH"
sam build

Build Succeeded
Built Artifacts: .aws-sam/build
Built Template: .aws-sam/build/template.yaml
```

1.5.3 Security Scan

- CodeQL analysis: 0 alerts found
- No vulnerabilities detected in workflow configurations

1.6 What Happens Next

1.6.1 For Test Workflow

1. Triggers automatically on PRs to main
2. Runs unit tests
3. Builds with SAM
4. Starts SAM local
5. Smoke tests /health and /catalog/rubros
6. Reports success/failure

1.6.2 For Deploy Workflow

1. Triggers on push to module/finanzas-api-mvp or manual dispatch
2. Validates all required variables exist
3. Authenticates via OIDC with AWS

4. Verifies identity with `aws sts get-caller-identity`
5. Installs dependencies
6. Builds with SAM
7. Deploys to AWS (creates/updates CloudFormation stack)
8. Retrieves API ID and URL from stack outputs
9. Performs smoke test on deployed /health endpoint
10. Generates comprehensive summary with curl examples

1.7 Files Modified

File	Lines Changed	Description
`.github/workflows/deploy-api.yml`	+23/-11	Fixed OIDC, added deps install, improved config
`.github/workflows/test-api.yml`	+57 new	New test workflow for PR validation
`services/finanzas-api/WORKFLOW_SETUP.md`	+165 new	Comprehensive setup and troubleshooting guide

Total: 245 insertions, 11 deletions

1.8 Known Limitations

1. **SAM local startup time:** Fixed 12-second sleep is sufficient for smoke tests but not production-grade
2. **Cognito variables:** No defaults provided as they are environment-specific
3. **Stub endpoints:** Return 501 by design, will be implemented in future iterations
4. **No prod workflow:** This implementation covers dev environment only

1.9 Problem Statement Compliance

All requirements from the original problem statement have been addressed:

Requirement	Status	Notes
1. Fix missing repo Variables	Done	Documented with defaults where appropriate
2. Replace local OIDC action	Done	Now uses aws-actions/configure-aws-credentials@v4
3. Split into test + deploy workflows	Done	test-api.yml and deploy-api.yml
4. Test stubs return 501	Done	Documented in README and integration file

Requirement	Status	Notes
5. Secure API expectations	Done	HTTP API with Cognito JWT authorizer
6. Dependencies install	Done	npm ci added to both workflows
7. esbuild availability	Done	PATH configured for esbuild access

1.10 Next Steps

1. **Set Repository Variables** - Configure the 5 required variables in GitHub settings
2. **Set Repository Secret** - Configure OIDC_AWS_ROLE_ARN
3. **Test Deploy Workflow** - Trigger manually or push to module/finanzas-api-mvp branch
4. **Verify OIDC** - Check workflow run for successful STS identity verification
5. **Validate Deployment** - Use curl commands from summary to test deployed API
6. **Implement Stubs** - Begin implementing the 9 stubbed endpoints

1.11 Support

- See `services/finanzas-api/WORKFLOW_SETUP.md` for detailed setup instructions
- See `services/finanzas-api/README.md` for API documentation
- See `services/finanzas-api/tests/integration/sam-local.http` for endpoint testing examples