

ETL Pipeline Project: CI/CD Simulation on GitHub

This project simulates a real-world data pipeline workflow using GitHub branches and automation. It covers everything from local development to production-ready releases—without requiring external cloud infrastructure.

Project Structure

etl-pipeline-india-nocloud/

- data/ # Source CSV files
- scripts/ # ETL modules (extract, transform, load)
- database.db # SQLite target
- requirements.txt # Python dependencies
- CHANGELOG.md # Auto-generated changelog
- .github/workflows/ # GitHub Action workflows
- README.md

Local Development

- git init
- python -m venv venv
- source venv/bin/activate # Windows: .\venv\Scripts\activate
- pip install -r requirements.txt
- python scripts/run_etl.py # Run ETL pipeline locally

Branching Strategy (Simulated Environments)

- Dev Integration - dev
- SIT / UAT - sit
- Production - main

Commands Used

- git checkout -b dev
- git checkout -b sit
- git checkout -b main

Promotions:

- dev → sit → main via pull requests
- GitHub Actions validate ETL code at every level

GitHub Workflows

etl-pipeline.yml – Code Validation

- Triggers: Push/PR to dev, sit, main
- Steps:
 - Checkout code
 - Set up Python
 - Install requirements
 - Run scripts/run_etl.py

tag-on-main.yml – Auto-Release with Timestamp

- Trigger: Push to main
- Behavior:
 - Generate version tag like v20250627-145121
 - Create GitHub release via CLI (gh)
 - Authenticated using PAT_TOKEN

changelog.yml – Auto Changelog Generation

- Trigger: Push to main or manual trigger
- Tool: github-changelog-generator
- Output: Commits and PRs logged in CHANGELOG.md
- Pushes back to main automatically

Branch Protection Rules (Recommended)

Apply via GitHub → Settings → Branches → Protection Rules

- Protect main, sit
- Require pull requests before merging
- Require GitHub Actions (CI) to pass
- Prevent force pushes and deletions

GitHub Repository Secrets

- PAT_TOKEN: Personal access token with repo:write and contents:write scopes

Common Git Commands Used

- # Create and push branches
 - o `git checkout -b dev`
 - o `git push -u origin dev`
- # Trigger CI via push
 - o `git add .`
 - o `git commit -m "Update script comment"`
 - o `git push`
- # Promote changes via GitHub PR: dev → sit → main