

5. Migration from other devops platform

Overview

i This guide outlines the standardized process for migrating repositories, metadata, and CI/CD pipelines from GitLab to GHE. To ensure we meet our **Definition of Done (DoD)**, every migrated project must align with our new naming conventions and security guardrails (ORCA/GHAS).

This guide outlines the standardized process for migrating repositories, metadata, and CI/CD pipelines from GitLab to GHE. To ensure we meet our **Definition of Done (DoD)**, every migrated project must align with our new naming conventions and security guardrails (ORCA/GHAS).

i This is the most critical step. We are moving from `.gitlab-ci.yml` to `.github/workflows/main.yml`. **Do not simply copy-paste logic; refactor for efficiency.**

Logic Mapping Table	
GitLab CI Concept	GitHub Actions Equivalent
stages	jobs (run in parallel by default; use <code>needs</code> for sequencing)
script	run
artifacts	actions/upload-artifact
variables	<code>env</code> (Job/Step level) or GitHub Secrets
only/except	<code>on:</code> (push, pull_request, workflow_dispatch)
include	<code>uses:</code> (for Reusable Workflows)
tags	<code>runs-on:</code> (e.g., self-hosted, azure-prod)

i Leveraging AI for Refactoring

Pro Tip: Use **GitHub Copilot**. Open your legacy `.gitlab-ci.yml` and ask Copilot: "Convert this GitLab CI pipeline to a GitHub Actions workflow using our corporate private runners and adding an ORCA security scan step."

Legacy Secrets: Do not hardcode secrets. Migrate all GitLab "CI/CD Variables" to **GitHub Secrets** at either the Repository or Environment level.

OIDC Strategy: For Azure/GCP deployments, stop using long-lived Service Account keys. Use **Workload Identity Federation (OIDC)** to allow GitHub Actions to authenticate directly with the cloud provider.



Every migrated pipeline **must** include these two blocks before it is considered "Production Ready":

ORCA Security Scan: To check for IaC misconfigurations (Terraform/Bicep) and container vulnerabilities.	
Dependency Review: To ensure no "Critical" vulnerabilities are introduced via open-source packages.	



To prevent "Split-Brain" development:

Set GitLab to Read-Only: Archive the project in GitLab immediately after a successful GHE migration.	
Update README: Add a notice in the GitLab repo pointing users to the new GHE URL.	
Update Webhooks: Redirect any external integrations (Jira, Slack, ServiceNow) to the new GitHub App/Webhook.	