

Using secrets in GitHub Actions — GitHub Docs

Chapter 4

academy.kspl.tech | Koenig AI Academy

Learning Objectives

- Configure a GitHub Actions pipeline that runs terraform fmt -check, validate, and plan on every pull request
- Post terraform plan output as a PR comment artifact visible to all repository collaborators
- Place the manual approval gate on the apply job so reviewers see the full diff before approving
- Inject cloud credentials and sensitive variables using TF_VAR_ environment variables sourced from GitHub secrets
- Diagnose and fix Error: No value for required variable failures in CI pipelines

The Four-Stage Validation Funnel

- It makes zero network calls, completes in under a second, and never modifies files — it only reports.
- No provider API calls, no backend access, no credentials required — safe to run on every fork and external PR.
- It exits non-zero with a machine-readable JSON report when it finds a problem.
- (Remote state backend setup is covered in [02-remote-state-locking-ml-teams](#).)

Posting Plan Output as a PR Comment

- CI logs require pipeline read access to view.
- A PR comment is visible to every collaborator with repository read permission.
- One constraint to plan around: GitHub PR comments are capped at 65,535 characters.

Continue Learning

Full course available at:

academy.kspl.tech