

Vier Schichten

#	Schicht	Typische Tools	Trigger
1	Sprach	hadolint · tflint · shellcheck · yamllint · actionlint	jeder PR
2	Format	markdownlint · Prettier · editorconfig-checker	jeder PR
3	Konvention	commitlint · branch-naming-policy	jeder PR (vor 1 + 2)
4	Domain	Spectral · gherkin-lint · helm lint · kubeconform	bei relevanten File-Changes

Drei Faustregeln

1. **Pinne alle Tool-Versionen.** Keine :latest-Tags. Ein Tool-Upgrade ist ein PR, kein versteckter Fail.
2. **Lokal = CI.** Was lokal mit `just lint` läuft, muss exakt dasselbe sein wie in der Pipeline.
3. **Eine Schicht pro Quartal.** Wer noch nicht geschichtet hat, beginnt mit Schicht 1. Alle vier auf einmal heißt Team-Frust.

Wann es sich NICHT lohnt

- Repo unter 100 Dateien und ein Maintainer → MegaLinter ist ehrlicher.
- Multi-Sprach-Repo mit mehr als acht produktiven Sprachen → MegaLinter spart Pflege.

Generische CI-Skizze

```
stages: [konvention, lint]

konvention:
  stage: konvention
  image: commitlint:18-alpine
  script: commitlint --from=$BASE_SHA

sprach:
  stage: lint
  needs: [konvention]
  image: hadolint:2.10-alpine
  script: hadolint Dockerfile

format:
  stage: lint
  needs: [konvention]
  image: markdownlint-cli2:0.5.1
  script: markdownlint-cli2 '**/*.md'

domain:
  stage: lint
  needs: [konvention]
  image: spectral:6.4
  rules:
    - changes: ['openapi/**']
  script: spectral lint openapi/*.yaml
```