Stages und Branches

> Zielgruppe: Anwender und Betreuer · Stand: 06.11.2026

Jedes Deployment-Repo kennt vier Stages, und jede Stage ist ein eigener Branch. Ein Commit wandert von Stage zu Stage. Ob er weiterwandert, entscheiden feste Gates: mal manuell, mal automatisch.

Die vier Stages

Stage = Branch	Zweck	Weiter zur nächsten Stage
`pre-development`	Einstieg, Default-Branch; im Lab die einzige Stage	manuell (`promote-next-branch`)
`development`	Entwicklungsumgebung	automatisch
`staging`	Abnahme vor Produktion	manuell (`approve-and-promote-production`)
`production`	Produktion	–

- Einen `main`-Branch gibt es nicht.
- Jede Stage hat ihren eigenen Terraform-State. Der State-Name ist der Branch-Name.
- Alle vier Stage-Branches sind geschützt. Daran hängen Secrets und Freigaben.

Stages ein- und ausschalten

- `enabled-stages` legt fest, welche Stages wirklich deployen. Inaktive Stages laufen in der Kaskade mit, ändern aber nichts.
- `promote-from` und `promote-to` begrenzen die Kaskade. `promote-to: pre-development` ist der Lab-Modus ohne Promotion.
- Eine Stage wird erst aktiv, wenn ihre Hosts definiert und ihre Variablen gesetzt sind.

Wie die Stage an anderen Stellen heißt

Ort	Schreibweise
Branch, OpenBao-Pfad, Terraform (`TF_VAR_deployment_stage`)	`pre-development`, `development`, `staging`, `production`

Ort	Schreibweise
Maschinen-Module (VM-Variable `stage`)	nur `dev`, `mngmt`, `prd`. Lab-VMs laufen als `dev`, `staging` läuft mit `prd`
Consul-KV und Namen im Lab	`lab`, z. B. `services/lab/…`

> Das sind drei Vokabulare für ähnliche Dinge. Für Secret-Pfade gilt immer der Branch-Name. `lab`, `dev` oder `prd` im OpenBao-Pfad findet keine Pipeline.

Weiterführend

- Lab, DEV und Produktion
- README der CI-Component, Abschnitte „Promotion“ und „Stages und Kippschalter“

Revision #1
Created 2026-11-06 18:06:22 UTC by Alexander Golembek
Updated 2026-11-06 18:06:22 UTC by Alexander Golembek