Rollen und Module versionieren

> Zielgruppe: Betreuer · Stand: 06.11.2026

Terraform-Module und Ansible-Rollen werden unterschiedlich versioniert. Module haben Versionslinien, Rollen nur zwei Branches. Kompatibilität entsteht immer im Modul, nie in der Rolle.

Ansible-Rollen

- Zwei Branches: `latest` (Default, Entwicklung) und `stable` (Konsum). Es gibt keine Tags.
- Module binden Rollen immer auf `stable`.
- Eine Rolle bekommt keine Aliase oder Fallbacks für alte Variablennamen.

Terraform-Module

- Versionslinien als Branches `<major>.<minor>`, z. B. `0.3`. Jeder Merge in den Linien-Branch veröffentlicht die nächste Patchversion.
- Stacks pinnen eine Linie, z. B. `~> 0.3.1` bzw. `>= 0.3.1, < 0.4.0`.
- Ein inkompatibler Umbau bekommt eine neue Linie (z. B. `0.2` → `0.3`). Die alte Linie bleibt für den Bestand.
- Die gerenderten Ansible-Requirements (`requirements.*.yml`) erzeugt ein CI-Job im Modul. Er pusht selbst einen Commit.

Kompatibilität bei Umbenennungen

1. Die Rolle benennt Variablen einfach um, ohne Alias.
2. Das Modul übergibt alte **und** neue Namen nebeneinander an das Playbook.
3. Geht das nicht, entsteht eine neue Modul-Linie.
4. Modul-MRs werden gemergt, **bevor** die Rolle auf `stable` kommt.

Globale Kontext-Variablen (`site`, `stage`, `role_slug`, `state`, `checkmk_server`) bleiben ohne Präfix.

Stolperfallen

>

> Das Anlegen eines neuen Linien-Branches veröffentlicht sofort `X.Y.0` mit dem Inhalt der Vorgängerlinie. Diese Version nicht verwenden, erst ab `X.Y.1` pinnen.

- Ein Modul, das eine Rolle auf einen Feature-Branch pinnt, hängt an diesem Branch. Den Branch nicht löschen oder force-pushen, bevor das Modul umgestellt ist.
- Nach einer Änderung an CI-Variablen auf den Folgecommit des Requirements-Jobs achten.

Weiterführend

- Merge heißt Deploy

Revision #1
Created 2026-11-06 18:10:49 UTC by Alexander Golembek
Updated 2026-11-06 18:10:49 UTC by Alexander Golembek