Secrets (KV)

- Pfadschema und Teamtrennung
- Secret anlegen
- Secret-Versionen und Löschen

Pfadschema und Teamtrennung

> Zielgruppe: Anwender und Betreuer · Stand: 06.11.2026

In OpenBao liegen Secrets in KV-Speichern (KV-v2-Mounts). Diese Seite erklärt, wo ein Secret hingehört und wer es lesen und schreiben darf. Wer die Regeln kennt, findet Secrets wieder und legt sie gleich an der richtigen Stelle ab.

> Gilt bisher nur im **Lab** (Lab-PoC). Für DEV und Produktion ist dasselbe Schema vorgesehen.

Ein Mount je Abteilung

Jede Abteilung hat einen eigenen Mount:

Mount	Abteilung
`cci/`	Cloud Core Infrastructure (CCI)
`cpi/`	Container Platform Infrastructure (CPI)
`cpcs/`	Container Platform Core Services (CPCS)
`cpds/`	Container Platform DEXPRO Services (CPDS)

Pfade

Standardschema für alle Abteilungen außer CCI:

Pfad	Wofür
`<abt>/<team>/<stage>/…`	Secrets eines Teams
`<abt>/shared/<stage>/…`	Secrets der ganzen Abteilung (`shared` ist als Teamname reserviert)
`<abt>/<team>/<stage>/nomad/<job>/…`	Secrets eines Nomad-Jobs

CCI ist **ein** Team und gliedert sich deshalb nach Standort und Projekt:

Pfad	Wofür

`cci/shared/<stage>/…`	für die ganze Abteilung, z. B. zentrale DNS- oder IPAM-Zugänge
`cci/<standort>/<stage>/shared/…`	für alle Projekte eines Standorts, z. B. `bremen1`
`cci/<standort>/<stage>/<projekt>/…`	nur für dieses Projekt

`<stage>` ist immer der Branch-Name: `pre-development`, `development`, `staging` oder `production`. Das Lab nutzt `pre-development`.

Woher Stage und Team kommen

- **Pipelines:** Die Stage ergibt sich aus dem Branch, auf dem die Pipeline läuft, das Team aus der GitLab-Gruppe des Projekts. Beides steckt im GitLab-JWT. An `production` kommen nur Pipelines auf geschützten Branches.
- **Menschen:** Das Team ergibt sich aus der Entra-Gruppe (in DEV, siehe Kapitel Anmeldung).
- **Nomad-Jobs:** Die Stage hängt am Nomad-Cluster, der Job am Job-Namen.

> Metadaten an einem Secret (`custom_metadata`) sind nur eine Beschriftung. Sie schützen nichts. Zugriff regeln allein Pfad und Policy.

Wer darf was

	Lesen	Schreiben	Endgültig löschen
Team-Pipeline	eigener Teampfad und `shared/` der Abteilung	eigener Teampfad (`production` nur von geschützten Branches)	nein (nur Soft-Delete)
Mensch im Team	eigener Teampfad und `shared/`	eigener Teampfad und `shared/`	ja
Nomad-Job	nur `…/nomad/<job>/`	nein (Anlegen und Rotieren macht die Team-Pipeline)	nein
Globale OpenBao-Admins	alle KV-Mounts	alle KV-Mounts	ja

Ausnahme CCI: Hier dürfen auch Pipelines in beide `shared`-Ebenen schreiben (Soft-Delete ja, endgültig löschen nein). Ein Projekt schreibt nur in seinen eigenen Projektpfad.

Die globalen Admins sind eine eigene Entra-Gruppe. Sie dürfen Secrets verwalten, aber OpenBao nicht konfigurieren (keine Policies, Anmeldeverfahren oder Mounts). Die Konfiguration entsteht als Code.

>

> Wer Policies oder Anmeldeverfahren schreiben darf, kann sich selbst jedes Recht geben. Diese Rechte gibt es deshalb nur für die Konfiguration als Code und das Admin-Token, nie für Teams.

Weiterführend

- OpenBao: KV Secrets Engine Version 2
- OpenBao: JWT/OIDC-Anmeldung

Secret anlegen

> Zielgruppe: Betreuer · Stand: 06.11.2026

Secrets entstehen auf genau zwei Wegen: Menschen legen sie nach dem Login per OIDC in der OpenBao-Oberfläche an, Pipelines schreiben sie mit ihrem GitLab-JWT. Andere Wege wie Admin-Token oder lokale Skripte sind nicht vorgesehen. Eine Anleitung für Anwender folgt, sobald es OpenBao in DEV gibt.

> **Lab-PoC:** Der OIDC-Login (Entra ID) kommt erst mit DEV. Bis dahin entstehen Secrets im Lab nur über Pipelines.

Voraussetzungen

- Du kennst Abteilung, Team bzw. Standort und Projekt sowie die Stage (siehe Pfadschema und Teamtrennung).
- **Oberfläche:** Du bist Mitglied der Entra-Gruppe deines Teams und im internen Netz. OpenBao ist nie über den Edge erreichbar.
- **Pipeline:** Das Projekt nutzt die CI-Component `ci-components/deployment`. Für das Projekt gibt es eine JWT-Rolle, die an Projekt, Branch und geschützten Branch gebunden ist.

Vorgehen: Oberfläche (OIDC)

1. OpenBao-Oberfläche öffnen und als Methode **OIDC** wählen, dann mit dem DEXPRO-Konto anmelden.
2. Den Mount deiner Abteilung öffnen, z. B. `cpds/`.
3. **Create secret** wählen und als Pfad den Pfad ohne Mount eintragen, z. B. `<team>/<stage>/<name>`.
4. Schlüssel und Werte eintragen und speichern.

Vorgehen: Pipeline (GitLab-JWT)

1. Die CI-Component meldet die Pipeline per GitLab-JWT an OpenBao an und stellt das Token als `BAO_TOKEN` bereit. Module und Jobs melden sich nicht selbst an.

2. Terraform schreibt das Secret mit dem Vault-Provider (Ressource `vault_kv_secret_v2`) in den eigenen Pfad, z. B. ein von `random_password` erzeugtes Passwort.
3. Die Pipeline darf nur in ihren eigenen Pfad schreiben (bei CCI auch in die `shared`-Ebenen), `production` nur von geschützten Branches.

Prüfen

In der Oberfläche zeigt das Secret unter **Version History** die neue Version mit Zeitstempel. In der Pipeline zeigt der Terraform-Plan bzw. das Apply die Ressource, aber nie den Wert.

> Erscheint die neue Version mit dem erwarteten Zeitpunkt, ist das Secret angelegt.

Stolperfallen

> Die Stage im Pfad muss exakt der Branch-Name sein (`pre-development`, nicht `lab` oder `dev`). Sonst findet die Pipeline das Secret nicht.

- Speichern in der Oberfläche legt eine neue Version an, die **alle** Felder ersetzt. Felder, die fehlen, sind in der neuen Version weg.
- Werte, die Terraform schreibt, stehen wie heute im Terraform-State.
- Für CCI gilt die Gliederung nach Standort und Projekt, nicht nach Team.

Weiterführend

- Pfadschema und Teamtrennung
- Secret-Versionen und Löschen
- OpenBao: KV Secrets Engine Version 2

Secret-Versionen und Löschen

> Zielgruppe: Betreuer · Stand: 06.11.2026

OpenBao hebt je Secret mehrere Versionen auf, und Löschen geht in zwei Stufen. Hier steht, wie du eine alte Version zurückholst und wann ein Secret wirklich weg ist. Menschen arbeiten dafür nach dem OIDC-Login in der Oberfläche, Pipelines über Terraform.

Regeln

- Je Secret bleiben die letzten **drei Versionen** erhalten. Ältere werden automatisch entfernt.
- Versionen laufen nicht nach Zeit ab.
- **Soft-Delete** markiert eine Version als gelöscht. Sie lässt sich wiederherstellen. Pipelines und Menschen dürfen das.
- **Endgültig löschen** (Destroy, Metadaten löschen) dürfen nur Menschen.

Vorgehen (Oberfläche)

1. Nach dem OIDC-Login das Secret öffnen und **Version History** wählen.
2. **Ältere Version zurückholen:** die Version öffnen und daraus eine neue Version erstellen. Der alte Inhalt wird damit zur aktuellen Version.
3. **Soft-Delete / Wiederherstellen:** bei der Version „Delete“ bzw. „Undelete“ wählen.

> **Endgültig und nicht umkehrbar:** „Destroy“ löscht die Daten einer Version, das Löschen der Metadaten entfernt das Secret mit allen Versionen. Vorher prüfen, dass keine Pipeline und kein Dienst das Secret noch nutzt.

Pipelines

Pipelines dürfen laut Policy nur soft-löschen. Endgültig löschen kann nur ein Mensch in der Oberfläche.

Prüfen

In **Version History** sind soft-gelöschte Versionen als gelöscht markiert, zerstörte als „destroyed“.

Stolperfallen

- Eine aus einer alten Version erstellte Version zählt als neue Version. Bei drei aufbewahrten Versionen fällt dadurch die älteste heraus.
- Im Lab gibt es keine automatischen Snapshots. Was endgültig gelöscht ist, ist weg.
- Was ein Terraform-Abbau genau löscht (nur die aktuelle Version oder mehr), ist noch nicht nachgewiesen. Vor dem ersten Abbau eines Stacks mit Secrets prüfen.

Weiterführend

- Pfadschema und Teamtrennung
- OpenBao: KV Secrets Engine Version 2