trutz.io

SPICKZETTEL · REDIS → VALKEY

# Der Ausweg-Check

Zeitleiste, Diagnose, Kompatibilitätsmatrix und Cutover-Ablauf. Begleitmaterial zum Video, damit du weißt, ob dein Rückweg noch offen ist, bevor ein Image-Tag ihn zumacht.

---

DIE EINE REGEL, AUS DER ALLES FOLGT

**Das Image ist austauschbar, das Volume nicht. Ein Rollback holt das Image zurück, nicht die Daten.**

SÄULE 1

### Tiefes technisches Verständnis

Du weißt, dass die ersten neun Bytes von `dump.rdb` darüber entscheiden, welche Version sie noch öffnet. Damit erkennst du das Zeitfenster, solange es offen ist.

SÄULE 2

### Open-Source-Lizenz

Valkey führt den Stand 7.2.4 unter BSD in der Linux Foundation weiter. Ohne diesen Fork gäbe es kein Ziel, zu dem du wechseln könntest.

## 01 Die Zeitleiste: was wann passiert ist

Video 01:09

| WANN | WAS |
| --- | --- |
| März 2024 | Redis wechselt auf RSAL und SSPL, beide nicht OSI-anerkannt. Letzte freie Version: 7.2.4 |
| März 2024 | Linux Foundation kündigt Valkey an, Fork an 7.2.4, weiter unter BSD |
| Juli 2024 | Redis 7.4: neue Hash-Kodierungen, RDB-Format 11 wird 12 |
| Mai 2025 | Redis 8 bringt AGPLv3, anerkannt offen. Am getrennten Datenformat ändert das nichts |

## 02 Wo stehst du? Drei Befehle

Video 02:31

- Version abfragen. Valkey meldet zusätzlich `redis_version:7.2.4`, das ist der Fork-Punkt und zugleich die Kompatibilitätsangabe.
- Formatversion auf der Platte lesen. **Diese neun Bytes entscheiden**, nicht der Image-Tag.
- Den laufenden Digest notieren. Der Tag ist eine Bitte, der Digest ist die Tatsache.

```
redis-cli INFO server | grep redis_version
head -c 9 /data/dump.rdb
# REDIS0011 = offen, REDIS0012 = zu, VALKEY080 = Valkey
kubectl get pod redis-0 -o jsonpath='{..imageID}'
```

## 03 Die Einbahnstraße

Video 03:27

- Der Tag-Bump läuft als sauberes Rolling Update durch, Kubernetes meldet nichts. Aus seiner Sicht passiert nichts Ungewöhnliches.
- Redis 7.4 schreibt den Snapshot beim nächsten `BGSAVE` neu. In der Standardkonfiguration reicht ein sauberes Herunterfahren, denn `save` ist gesetzt und SIGTERM speichert.
- Der Rollback endet im `CrashLoopBackOff`, auch über `rollout undo`. Den abgestürzten Pod ersetzt das StatefulSet nicht von allein.

```
# nach dem ersten Speichern unter 7.4:
head -c 9 /data/dump.rdb   ->   REDIS0012
# Rollback auf 7.2.4, Log des Pods:
# Can't handle RDB format version 12
```

## 04 Warum niemand überspringen kann

Video 05:51

Ein RDB-Eintrag nennt **erst den Typ, dann die Länge**. Ohne die Null davor ist die Sieben nur irgendein Byte:

```
00                    # Typ: String
07                    # Länge des Schlüssels
6b 75 6e 64 65 3a 31  # kunde:1
0a                    # Länge des Werts
54 72 75 74 7a 20 …   # Trutz GmbH
```

Wer den Typ nicht kennt, weiß nicht, wie weit er springen müsste. Weiterraten wäre stiller Datenverlust, deshalb bricht der Server ab. Redis 7.4 brauchte für `HEXPIRE` zwei neue Eintragstypen, und neue Opcodes heben die Formatversion.

## 05 Cutover zu Valkey, ohne Downtime

Video 08:43

- [ ] Valkey leer daneben stellen, per `REPLICAOF` anhängen. In Kubernetes über den headless Service, die Replikation adressiert einen einzelnen Knoten.
- [ ] Das `OK` beweist nichts. Erst `master_link_status:up` und ein echter Lesetest zeigen, dass der Sync steht.
- [ ] Konfiguration, Passwörter und Monitoring wandern **nicht** mit. Bei gesetztem Passwort vorher `CONFIG SET masterauth`.
- [ ] Mit Sentinel ist die Reihenfolge die Arbeit: erst die Sentinels, dann die Replicas, dann ein Failover, der alte Master zuletzt.

```
valkey-cli REPLICAOF redis-0.redis 6379
valkey-cli INFO replication   ->
master_link_status:up
# Gegenprobe mit echten Daten, dann lösen:
redis-cli SET foo bar   &&   valkey-cli GET foo
valkey-cli REPLICAOF NO ONE
```

## 06 Die Matrix, alle neun Kombinationen

Video 10:11

| MASTER → REPLICA | REDIS 7.2 | REDIS 7.4 | VALKEY 9 |
|---|---|---|---|
| Redis 7.2 | up | up | up |
| Redis 7.4 | down | up | down |
| Valkey 9.1.1 | up | up | up |

Gemessen mit einem String-Key. `down` heißt immer `Can't handle RDB format version 12`: Eine Replikation ist ein Snapshot über die Leitung, also dieselbe Grenze wie auf der Platte. **Der Fork hat die Kompatibilität behalten, das Original hat sie aufgegeben.**

## 07

Video 13:34

# Vorsorge: drei Dinge, ab sofort

- [ ] **Keine gleitenden Tags bei Datenbanken.** Auf Digest pinnen, dann entscheidet kein Registry-Push über dein Datenformat.
- [ ] **Datenbank-Images raus aus der automatischen Aktualisierung.** Für einen zustandslosen Dienst ist ein Bot-Vorschlag harmlos, für einen mit Volume ist er eine Migrationsentscheidung.
- [ ] **Vor jedem großen Sprung ein Volume-Snapshot.** Er ist hier das einzige Mittel, das den Rückweg tatsächlich offenhält.
- [ ] `appendonly yes` rettet dich nicht. Die Basis-Datei im `appendonlydir/` ist selbst eine RDB-Datei, und der Rewrite läuft automatisch. AOF verzögert das Problem, es löst es nicht.

> DAS LEISE FEHLERBILD KENNEN
>
> Die Rückrichtung Valkey → Redis funktioniert nur, solange nichts im Datensatz liegt, das nur der Fork kennt. Ein Hash mit Feld-Ablaufzeiten reicht, und der Sync scheitert. Im Log steht dann kein Wort über Format oder Kompatibilität, nur „I/O error reading bulk count from MASTER“ und eine Reconnect-Schleife. Die laute Meldung findest du sofort, diese hier suchst du.

---

## Entschieden wird es beim ersten Befehl, den es auf der anderen Seite nicht gibt.

Dieser Spickzettel gehört zum Video über Redis 7.4, das RDB-Format und den Weg zu Valkey. Jede Woche praxisnahe Antworten auf Docker- und Kubernetes-Probleme: der Newsletter.

trutz.io

Newsletter