Architektur (RADOS)

RADOS & Daemons

RADOS (Reliable Autonomic Distributed Object Store) ist das Fundament; RBD, CephFS, RGW und librados setzen darauf auf.
MON: hält die Cluster-Maps (MonMap, OSDMap, CRUSH-Map, MDSMap, PGMap), Konsens via Paxos. Ungerade Zahl ≥3, Mehrheit (Quorum) muss up sein. Nur State, keine Nutzdaten.
MGR: aktiv/standby, Module (Dashboard, Prometheus-Exporter, Balancer, Autoscaler, Orchestrator/cephadm). Genau ein active, mind. 2 für HA.
OSD: je einer pro Datenträger. Daten, Replikation, Recovery, Scrubbing, Rebalance. Backend BlueStore.
MDS: nur CephFS, hält POSIX-Metadaten; die Daten liegen in RADOS.
RGW: HTTP-Daemon mit S3- und Swift-API.

CRUSH & Placement

CRUSH (Controlled Replication Under Scalable Hashing) ist deterministisch: der Client rechnet Objekt → PG → OSD selbst, keine zentrale Lookup-Tabelle.
CRUSH-Map = Bucket-Baum (**root** → **dc** → **rack** → **host** → **osd**) plus Rules plus Device-Classes (**hdd**/**ssd**/**nvme**).
PG (Placement Group): Objekt →(Hash)→ PG → Acting-Set (OSDs). Der **pg_autoscaler** steuert die PG-Zahl.
Pool: Replikation oder EC, CRUSH-Rule, **pg_num**, **size**/**min_size**, Quota, **application**-Tag (**rbd**/**cephfs**/**rgw**).

Redundanz & BlueStore

Replikation **size=3**/**min_size=2** (Default, empfohlen): 3-fach Overhead, einfaches Recovery.
Erasure Coding k+m (z. B. **k=4,m=2** → Faktor 1,5, verträgt 2 OSD-Ausfälle): platzsparend, aber CPU- und latenzintensiv, primär RGW-Bulk und CephFS-Datenpool.
BlueStore: raw Device + RocksDB, kein Dateisystem dazwischen. FileStore ist seit Nautilus deprecated und ab Reef (v18) entfernt, also ab Reef BlueStore-only.

Zugriffsschichten

RBD (Block)

Thin-provisionierte Block-Images, **krbd** (Kernel) oder **librbd** (QEMU). Layering, Trim/Discard.
Snapshots und CoW-Clones aus einem geschützten Snapshot (**snap protect** → **clone**).
rbd-mirror für DR: journal-based (2× Writes, feingranular) vs. snapshot-based (Delta, weniger Overhead); one-way oder two-way.

CephFS (POSIX)

MDS hält die Metadaten, die Nutzdaten liegen in RADOS. **max_mds**>1 gibt mehrere aktive MDS (dynamisches Subtree-Partitioning) plus Standby(-replay).
Getrennte Daten- und Metadaten-Pools. Snapshots je Verzeichnis (**.snap**).
Subvolumes und Subvolume-Groups sind die Basis für den Kubernetes-CSI.

RGW (Objekt) & librados

RGW: S3 + Swift, Bucket Policies, ACLs, STS, Versioning, Lifecycle, Notifications.
Multisite: Realm → Zonegroup → Zone, active-active seit Kraken (jede Zone beschreibbar, **radosgw**-Daemons synchronisieren selbst, kein separater Sync-Agent). Status: **radosgw-admin sync status**.
librados: native Lib (C/C++/Python), atomare Ops, Object Classes.

NVMe-oF Gateway

Exportiert RBD-Images als NVMe/TCP-Targets an Nicht-Ceph-Clients (VMware, Bare-Metal).
Produktiv ab v20 (Tentacle): Gateway-Groups, mehrere Namespaces, Multi-Cluster, OAuth2, KMIP. In Squid (v19) nur eingeschränkt.

Deployment

cephadm (Default)

Seit Octopus (v15), containerisiert (Podman/Docker), kein externes Config-Management nötig.
Orchestrator: Hosts (**ceph orch host add**), Labels (frei; **_admin** verteilt **ceph.conf** + Keyring), Service Specs (mon/mgr/osd/mds/rgw/nfs/rbd-mirror/monitoring), Placement (Host-Liste, Label, **count** oder Host-Pattern).

```
cephadm bootstrap --mon-ip 10.0.0.10
ceph orch host add node2 10.0.0.11 mon,osd
ceph orch apply mon --placement="label:mon"
ceph orch apply osd --all-available-devices
ceph orch ls          # Service-Status
ceph orch ps          # Daemon-Instanzen
```

Rook & ceph-ansible

Rook: CNCF-Graduated-Operator, der offizielle Weg für Ceph in Kubernetes (oder einen externen Cluster via CSI an K8s anbinden).
Faustregel: cephadm für Bare-Metal, Rook für Kubernetes.
ceph-ansible: deprecated, Migration nach cephadm. (**cephadm-ansible** sind nur Helper-Playbooks, nicht verwechseln.)

Kern-Commands

```
ceph -s                  # Cluster-Status
ceph health detail       # Warnungen im Klartext
ceph df                  # Kapazitaet je Pool
ceph osd df tree         # Auslastung je OSD/Host
ceph versions            # Daemon-Versionen
# OSD-Management
ceph osd tree
ceph osd set noout       # keine Migration (Wartung)
ceph osd out 5           # OSD 5 evakuieren
ceph osd reweight 5 0.8 # temporaeres Gewicht
ceph osd purge 5 --yes-i-really-mean-it
# Pools & Autoscaler
ceph osd pool create data 128 128 replicated
ceph osd pool set data size 3
ceph osd pool set data min_size 2
ceph osd pool set data pg_autoscale_mode on
ceph osd pool autoscale-status
ceph osd pool application enable data rbd
# Erasure-Profil & EC-Pool
ceph osd erasure-code-profile set ec42 \
  k=4 m=2 crush-failure-domain=host plugin=jerasure
ceph osd pool create ecpool erasure ec42
# CRUSH
ceph osd crush tree --show-shadow
ceph osd crush rule create-replicated ssd-rule \
  default host ssd
ceph osd crush set-device-class ssd osd.5
ceph osd crush move node7 rack=rack2
# cephx / Auth
ceph auth ls
ceph auth get-or-create client.app \
  mon 'allow r' osd 'allow rwx pool=data'
# Caps: <daemon> 'allow <r|w|x|rwx|*>
#   [pool= | profile rbd | path=/...]'
# RBD: Image, Snap, Clone, Mirror
rbd create data/vol1 --size 100G
rbd snap create data/vol1@base
rbd snap protect data/vol1@base
rbd clone data/vol1@base data/clone1
rbd mirror image enable data/vol1 snapshot
# CephFS
ceph fs volume create cephfs
ceph fs subvolumegroup create cephfs csi
ceph fs subvolume create cephfs v1 --group_name csi
ceph fs set cephfs max_mds 2
ceph fs status
# RGW (S3) & Balancer/Orchestrator
radosgw-admin user create --uid=app --display-name="App"
radosgw-admin key create --uid=app --key-type=s3 \
  --gen-access-key --gen-secret
radosgw-admin sync status
ceph balancer mode upmap && ceph balancer on
```

Betrieb & Diagnose

PG-States

Ziel: **active+clean** (bedient I/O, alle Replikate konsistent).
degraded: nicht alle Replikate da. **undersized**: Acting-Set < **size**. **peering**: OSDs einigen sich auf den PG-Stand. **remapped**: temporär anderes Set bis Backfill fertig.
backfilling/**backfill_wait**: Voll-Kopie läuft/wartet (**backfill_toofull** = Ziel-OSD zu voll). **recovering**: Delta aus dem Log.
incomplete: fehlende Schreib-Historie, Datenverlust-Risiko. **inconsistent**: Scrub-Abweichung → **ceph pg repair <pgid>**. **down**/**stale**/**peered**: keine aktuelle Kopie erreichbar.

Scrubbing & Rebalancing

Scrub light (täglich, Metadaten) vs. deep-scrub (wöchentlich, byteweiser Vergleich, I/O-intensiv). Flags **noscrub**/**nodeep-scrub**.
Rebalance entsteht durch CRUSH-Änderung, OSD in/out oder neue Kapazität. Drosseln über **osd_max_backfills** und **osd_recovery_max_active**; Flags **norebalance**/**nobackfill**.

Disk-Tausch (cephadm)

noout setzen, OSD sauber evakuieren und den Slot halten, Disk physisch tauschen (cephadm re-deployt über die OSD-Spec), dann **noout** lösen.

```
ceph osd set noout
ceph orch osd rm 5 --replace   # evakuieren, Slot halten
# Disk tauschen -> cephadm re-deployt via osd-Spec
ceph osd unset noout
```

Health-Warnungen (typisch)

PG_DEGRADED/**PG_AVAILABILITY**: Redundanz bzw. Verfügbarkeit einer PG.
OSD_NEARFULL/**OSD_BACKFILLFULL**/**OSD_FULL**: Kapazität, **FULL** blockt Writes.
MON_CLOCK_SKEW: NTP/chrony driftet. **SLOW_OPS**/**SLOW_REQUEST**: hängende Ops.
Weitere: **POOL_NEARFULL**, **MON_DISK_LOW**, **TOO_MANY_PGS**/**TOO_FEW_PGS**, **MDS_SLOW_METADATA_IO**.

```
# Diagnose
ceph tell osd.5 bench           # Durchsatz je OSD
ceph tell mon.* mon_status      # Quorum / MonMap
ceph daemon osd.5 perf dump     # Admin-Socket, lokal
ceph daemon osd.5 dump_ops_in_flight
ceph pg 3.1a query              # Peering-/Recovery-Detail
ceph osd perf                   # commit/apply-Latenz
```

Fallen & Tunables

Sicherheitsrelevante Defaults

min_size=1 ist gefährlich: Schreiben mit nur einer Kopie, ein weiterer Ausfall bedeutet Datenverlust. Bei Replikation **min_size=2** halten.
Kapazitäts-Ratios: **nearfull_ratio** 0.85 (Warnung), **backfillfull_ratio** 0.90 (kein Backfill mehr), **full_ratio** 0.95 (Pool wird read-only, Stillstand). Puffer für Recovery lassen.
MON-Zahl immer ungerade (3 oder 5), sonst kein sauberes Quorum.
mon_max_pg_per_osd: Default-Richtwert um 250, aber versionsabhängig, vor Produktion gegenprüfen.

Neueres seit 2024

Crimson & SeaStore (Preview)

Crimson = Next-Gen-OSD auf Seastar (thread-per-core, DPDK für NVMe): in Tentacle Tech Preview, nicht produktionsreif, unterstützt inzwischen EC-Pools. Klassischer OSD + BlueStore bleiben Default und Produktion.
SeaStore = NVMe-optimiertes Objektstore-Backend für Crimson, ebenfalls Tech Preview und früh.

Stretch-Cluster & NVMe-oF (prod)

Stretch-Cluster: 2 Datacenter plus Tiebreaker-MON in einer dritten Site (VM, hohe Latenz ok). Typisch 5 MON (2+2+1), Pool **size=4** (2 je Site), **min_size=2**; Tiebreaker als **disallowed_leader**.
NVMe-oF produktiv ab v20 (Tentacle): NVMe/TCP, Gateway-Groups, Multi-Cluster, OAuth2, KMIP.

Versionen & Historie

Release-Stand (Juli 2026)

Tentacle (v20): neueste Serie (seit 18.11.2025), aktuell 20.2.2. Empfohlen für Neu-Deployments.
Squid (v19): reife Stable-Serie (seit 26.09.2024), aktuell 19.2.4. Primär-Zielversion dieses Sheets.
Reef (v18): 18.2.8 final, EOL um 03/2026. Quincy (v17): EOL 2025.
Codenamen alphabetisch nach Kopffüßern: Pacific(16) → Quincy(17) → Reef(18) → Squid(19) → Tentacle(20). Kadenz rund jährlich, Stable-Serie lebt etwa 24 Monate.
Upgrade seriell, kein Major-Skip (Quincy → Reef → Squid → Tentacle).

Herkunft

2004 UC Santa Cruz (Sage Weil) → 2012 Inktank → 2014 Red Hat kauft Inktank → 2018 Ceph Foundation (directed fund der Linux Foundation) → 2019 IBM kauft Red Hat → seit 2023 als Enterprise-Produkt bei IBM. Der Upstream bleibt offen und community-getrieben.

ceph-cheatsheet.de

Ceph Cheatsheet von OMNI52

Weiterführen, nicht selbst neu erfinden

Storage-Architektur & Betrieb

OMNI52™ hilft Plattform-Teams, Ceph-Cluster für regulierte Enterprise-Umgebungen sauber aufzustellen und zu betreiben.

Cluster-Review, CRUSH- und Pool-Design, EC- vs. Replikations-Sizing, DR mit rbd-mirror und RGW-Multisite, Stretch-Cluster, Upgrade- und Disk-Tausch-Runbooks. Output landet als GitOps-Repo und Runbooks bei euch im Cluster. Euer Team operiert weiter, nicht wir.

OMNI52 GmbH, Ebern · istio-consulting.de · kostenloses 30-min Initial-Assessment

Verwandte Cheatsheets

Ebenfalls von OMNI52: kubernetes-cheatsheet.de (Ceph als K8s-Storage via Rook/CSI) · rke2-cheatsheet.de (gehärtete Kubernetes-Plattform) · rancher-cheatsheet.de (Cluster-Management).