Backup & Disaster Recovery

Backup schedule, RPO/RTO targets, restore procedures, DR test cadence

- Chapter 14 — Backup & Disaster Recovery

Chapter 14 — Backup & Disaster Recovery

14.1 Current State and Scope

As of 2026-07-18, **tier 1 (DigitalOcean Backups) is enabled on the droplet fleet** — the first backup layer in production. Tiers 2 and 3 remain unbuilt: no database dumps (no databases yet), no repo mirrors, no Vault snapshots (no Vault yet), no offline custody artifacts. Single-droplet loss is now recoverable; application-consistent recovery, region loss, and account compromise are not yet covered. Scope of this chapter: everything required to reconstruct the Platform (data, configuration, state, secrets, evidence); the recovery procedures per failure class; and the test cadence that keeps "we have backups" meaning "we can restore." Chapter 13's audit anchors are the integrity counterpart; this chapter is survivability. Controlling family: the backup/DR control set imported into KO-001.

14.2 Backup Architecture — Three Tiers

The design principle: **IaC-first recovery** — most of the platform is rebuilt from code, not restored from images. Three complementary tiers, each covering a failure class the others cannot:

Tier	Mechanism	State	Covers	Cannot cover
1 — Disk images	DigitalOcean Backups (native)	**Live**	Fast single-droplet restore	Application consistency (crash-consistent only); account compromise (backups are not downloadable and live inside the DO control plane)
2 — Application artifacts	Encrypted dumps + state archives to Spaces, replicated NYC3→SFO3	Unbuilt	Application-consistent data recovery; region loss	Hostile DO account — see tier 3
3 — Off-account survivability	Repo mirrors, offline age identity (SBS custody), Vault Shamir shares, audit anchors (Rekor + SBS email)	Unbuilt	Total account/custody compromise (§14.6)	— (this is the floor)

14.2.1 Tier 1 — DigitalOcean Backups (Live)

Enabled on the fleet as of 2026-07-18. Target configuration:

Droplet	Frequency (target)	Retention (target)

`sbsdash-server-prod`	Daily	7 days
`prod-vault` (at creation)	Daily	7 days
`monitor-servers`	Daily	7 days
`sbs-wiki` (carries through NYC2 rebuild)	Daily	7 days
`Dev-SBS-server`	Weekly	4 weeks

Open verification items against the live enablement (Build Register item 96 residual):

- **Plan and settings audit** — confirm which plan is active (usage-based, GA since Nov 2025 and priced by restorable file size with configurable frequency/retention, vs. legacy flat-percentage) and the actual per-droplet frequency selected (usage-based plans allow 4 h / 6 h / 12 h / daily / weekly). Align to the target table or amend it.
- **Codify in tofu** at the §10.6 import — enablement was performed via console; once imported, a droplet created without backups surfaces as a plan diff rather than a discovered gap. Console-enabled today is acceptable as bootstrap; console-managed permanently is drift.
- **`Dev-Gitlab-server`** — uninventoried (§2.2.2) but currently serving the apex record; take one backup/snapshot of it before the DNS repoint as cheap insurance, regardless of its final disposition.

Ongoing tier-1 mechanics: monthly, one Prod backup per droplet is **converted to a snapshot** and retained on a 12-month rolling basis (backups convert to snapshots for indefinite storage — the medium-term image tier). Restore paths are in-place restore or new-droplet-from-backup; creation runs roughly 2 min/GB used, keeping backup windows in minutes at our footprint.

Position in the design, stated plainly: tier 1 is the **speed layer for single-droplet loss only**. Disk images are crash-consistent — a mid-write PostgreSQL image can restore dirty, so databases recover from tier 2; and an image restores *whatever existed at image time*, drift and compromise included, so the IaC rebuild (§10) remains the clean-room path.

14.2.2 Tier 2 — Application Artifacts (What Code Cannot Regenerate)

Asset	Reconstruction source	Backup artifact
Droplets (OS + services)	Rebuilt — tofu + Ansible (§10); tier-1 image as fast path	—
Infra + app repos, IaC state	GitHub (off-platform) + tofu state in Spaces (§10.2)	Nightly `git bundle` mirror of both repos to `sbs-dash-backups` — GitHub org loss is inside the threat model (§2.1 revocability cuts both ways)
PostgreSQL	—	`pg_dump` per-database (per-Compartment, §7.4) nightly + WAL archiving continuous once the PITR trigger fires (§14.4)

Asset	Reconstruction source	Backup artifact
MongoDB	—	`mongodump` per-database nightly
Redis	Declared AOF keyspaces only (§7.1)	AOF-covered; nightly RDB copy off-host. Cache loss accepted by design
Vault	—	**Raft snapshot every 6 h** (§8.2) — the single most critical artifact; losing Vault + its snapshots = re-keying the platform
Audit segments	Already in `sbs-dash-audit`	Cross-region replication NYC3→SFO3 (§13.4); anchors survive dual-bucket loss
Grafana dashboards, alert rules, nginx, compose, nftables	Repo (file-provisioned, §12.5/§10.3)	Covered by repo mirror — this is why §12.5 bans console-authored dashboards
TLS keys	**Not backed up** — reissued via ACME (§5.2)	Deliberate: reissuance ≤ minutes; escrow risk > loss cost
BookStack (wiki)	—	Nightly DB dump + uploads archive (runbooks are operational capital)

All dumps are **encrypted client-side before upload** (age; key in Vault, with the age *identity* also held offline in SBS custody per the §9.6 break-glass pattern — an encrypted backup whose only key lives in the Vault it must survive is a circular dependency, broken here) and land in `sbs-dash-backups` (NYC3) with lifecycle replication to SFO3.

14.3 Schedule and Retention

Artifact	Frequency	Retention
DO Backups (disk images) — **live**	Daily Prod / weekly Dev (target; audit pending), DO-scheduled	7 d / 4 w native; 12 monthly via snapshot conversion
DB dumps (PG, Mongo, BookStack)	Nightly 03:00 ET	14 nightly, 8 weekly, 12 monthly (GFS)
WAL archive (when enabled)	Continuous	14 days
Vault Raft snapshot	6-hourly	28 snapshots (7 d) + 12 monthly
Repo mirrors	Nightly	30 days
Audit replication	On segment close (§13.2)	7 y (§13.4)
Redis RDB	Nightly	7 days

Every tier-2 job emits completion + size + age metrics to the `integrity` job (§12.2); tier-1 backup age is **polled from the DO API per droplet** and exported to the same job — now actionable immediately and prioritized, since it is the only automated proof the live backups are actually completing. **Any backup age > 26 h (tier-1 daily set or tier-2 nightly set) is `critical`** (§12.4) — a silently failing backup is the canonical DR failure and is alerted, not discovered.

14.4 RPO / RTO Targets

Per failure class — one blended number would be fiction:

Failure class	RPO	RTO	Recovery path
Single service/container failure	0	≤ 15 min	Compose restart / redeploy pinned digest (§6.7)
Single droplet loss (non-data)	0 (stateless)	**≤ 30 min** (tier-1 image restore) / ≤ 2 h (IaC clean-room rebuild)	Image restore when the failure is mechanical; IaC rebuild when state is suspect
Prod app droplet loss (interim data services, §6.5)	≤ 24 h (nightly dumps); ≤ 5 min once WAL PITR enabled	≤ 4 h	Image-restore or rebuild droplet, restore tier-2 dumps, re-assert ACLs (§7.4), redeploy
Vault droplet loss	≤ 6 h	≤ 2 h	New droplet (IaC or image) + Raft snapshot restore + quorum unseal (§8.2) — RTO includes assembling 3 shareholders; that human dependency is the honest bottleneck
Monitor/bastion loss	0 config / 30 d metrics history accepted lost	≤ 4 h	Image restore or IaC rebuild; **interim SSH path:** temporary direct-SSH firewall rule to prod via emergency PR (§10.4), removed at bastion restoration
Wiki loss	≤ 24 h	≤ 4 h	Image restore or IaC rebuild + dump restore
Region loss (NYC2)	≤ 24 h (SFO3 replicas; tier-1 images are region-bound and do not participate)	**≤ 3 business days**	Full IaC re-deploy to alternate region (region is a tofu variable, §10.3 globals); DNS repoint (§4.3); restore from SFO3. Not hot-standby — stated cost decision at this scale, revisited at Ch. 16 triggers
Account compromise (DO or GitHub org)	Bounded by tier-3 artifacts	Days, incident-governed	§14.6 — tier 1 and in-account tier 2 presumed lost

Until tier 2 exists, the honest fleet-wide posture is: **RPO for any future data = undefined; recoverability = single-droplet mechanical failure only.** This line is deleted from the specification when items 86–92 close — it is written here so the gap is a stated fact, not an implication. Database RPO ≤ 24 h is the accepted Day-1 posture once dumps run; the WAL-archiving upgrade to ≤ 5 min is **triggered when the first Compartment carries production**

tenant data — not before (cost honesty), mandatorily not after.

14.5 Restore Procedures

Each is a wiki runbook (§1.4 publication target) with exact commands and a verification tail; summarized:

1. **Single-droplet fast path (tier 1 — available today):** restore image or create-from-backup → verify drift (`tofu plan` + Ansible `--check` clean against repo state once §10 exists; until then, manual diff against SERVER_CHANGELOG.md — the interim reality, stated) → rejoin monitoring → smoke test.
2. **Database restore (tier 2):** provision clean engine (IaC) → decrypt dump (age key from Vault, or offline custody copy if Vault is the casualty) → restore per-database → **re-assert per-Compartment roles/ACLs from IaC before any application start** (a restored database with default grants is an isolation breach, §7.4) → row-count comparison vs. manifest → smoke test.
3. **Vault restore:** IaC/image droplet → `raft snapshot restore` → quorum unseal → audit-device continuity check → **rotate any credential issued after snapshot timestamp** (derivable from the Vault audit stream).
4. **Full-region DR:** run-order: VPCs → Vault (secrets first — everything authenticates through it) → data services + restores → app droplets + deploys → monitor → DNS cutover (TTLs drop to 300 at DR declaration) → §12.7-style verification sweep before declaring recovery.
5. **Restore-time isolation rule:** Compartment agent workloads **stopped** during recovery — agents never run against partially restored data (Guardian policy state could lag data state); restart is gated on the verification tail and is itself an audit-chain event (§13.3).

14.6 Worst-Case: Account/Custody Compromise

The survivability floor. If the DO account *and* GitHub org are both lost or hostile: tier 1 is gone by definition (backups are not downloadable and live inside the account); in-account tier 2 is presumed gone. Reconstruction requires exactly the tier-3 set — repo mirrors (SFO3 replicas), encrypted dumps + **offline age identity** (SBS custody), Vault snapshot + **Shamir shares** (5 named holders, §8.2), and **audit anchors** (Rekor + SBS email, §13.2). Each lives outside the compromised trust domain by design. The runbook is the incident-response DR annex (MON-05 linkage); the §14.7 annual tabletop rehearses this scenario specifically.

14.7 DR Test Cadence

Untested backups are hypotheses. Each test produces a dated wiki report (MON-04/quarterly-review evidence):

Test	Cadence	Pass criterion

Tier-1 restore drill — first execution scheduled now (the layer is live; prove it): restore latest backup of a non-critical droplet (wiki or dev) to a new droplet, boot, inspect, destroy	First: immediate. Then quarterly within the rotation below	Boots, services start, data present as of backup time
Automated restore-verify: latest PG/Mongo dump into throwaway Dev container, row counts + ACL assertions	Weekly, automated (CI), from tier-2 build	Green `integrity` metric; failure = `critical`
Single-droplet recovery drill, **alternating quarterly between tier-1 image restore and IaC clean-room rebuild**	Quarterly	RTO ≤ target; post-restore drift check clean
Vault snapshot restore into isolated instance (test shares, no prod unseal)	Quarterly	Unseal + read probe passes
Full-region tabletop + partial technical (Vault+DB restore into SFO3)	Annually	Runbook gaps filed as Build Register items
Account-compromise tabletop (§14.6)	Annually	Custody chain confirmed current (share holders, offline key, alias access)

14.8 Verification (Build Gates)

Gates (LH-SBS-INST-001): DO Backups confirmed active on all five droplets with plan/frequency audit closed and settings codified in tofu (item 96 residual); tier-1 backup-age visible via DO API and exported to `integrity` per droplet; first tier-1 restore drill passed; every §14.2.2 artifact present in `sbs-dash-backups` with age metric green; SFO3 replication object-count parity; dump decrypts with the **offline-custody key copy** (circular-dependency break proven, not just the Vault copy); weekly restore-verify green ≥ 2 consecutive cycles pre-Cutover; Vault snapshot restore drill passed; backup-age alert fires in induced-failure drill; monthly snapshot-conversion automation present in IaC; restore runbooks on wiki with last-tested dates; WAL-PITR trigger condition documented; region variable proven by clean `tofu plan` targeting SFO3; `Dev-Gitlab-server` pre-repoint backup taken.