Chapter 2 — Infrastructure Overview

2.1 DigitalOcean Account Structure

The Platform operates under a single dedicated DigitalOcean account owned by SBS (the "SBS Account"). A dedicated account — rather than a shared or Ledger Hub-owned account — keeps billing, resource ownership, and access control unambiguously with SBS and satisfies the tenant-separation intent of the IAM control family at the infrastructure root.

Account-level configuration (target):

- **Account ownership** — SBS-controlled email alias (not an individual mailbox), so ownership survives personnel changes. Execution owner: Nelson Santos (IT Director).
- **Account MFA** — phishing-resistant MFA enforced on the account root and on every team member, consistent with IAM-03. No shared credentials.
- **Team access** — DigitalOcean Teams with role-scoped membership. Ledger Hub Networks platform engineering joins as team members with the minimum role required for build activities; access is reviewed under IAM-06 and revocable by SBS without Ledger Hub involvement.
- **API tokens** — issued per-purpose (IaC pipeline, monitoring integrations), scoped read or write as required, stored exclusively in Vault (Chapter 8), rotated per the secrets lifecycle. No tokens in repositories, plaintext CI variables, or local files — per INF-02 and INF-03.
- **Billing alerts** — configured at account level so anomalous resource creation (a common compromise indicator) is surfaced to the account owner immediately.

Nonconformance — credential handling (CRITICAL). As of 2026-07-17 the operational changelog carried plaintext credentials: a shared root password stated to cover all project droplets, the GitHub account, the wiki admin account, and the Grafana admin account. All four credential sets are treated as compromised. Remediation (Build Register item 15): rotate all four immediately; migrate to Vault upon its deployment (Chapter 8) with interim storage in an approved password manager; strip the credentials section from the changelog and purge it from any version-control history; enforce distinct per-service credentials. No shared passwords are permitted for any host or service.

2.2 Project Topology

Resources are organized into two DigitalOcean Projects:

Project	Project ID	Purpose
SBS The Dash Dev (Default)	`ac656e28-2196-410e-847c-bf3b60363710`	Development and staging. Build and integration work occurs here first; no tenant data.
SBS The Dash Prod	`c49c646f-f268-4b51-a00e-ad9fb5055569`	Production. Live Platform resources: application services, observability stack, documentation portal, audit-log storage.

2.2.1 Droplet Inventory (As Built, 2026-07-18)

Droplet	Project	Region / VPC	Size	Public IP	Private IP	OS	Role
`Dev-SBS-server`	Dev	NYC2 / default-nyc2	8 GB / 160 GB	162.243.252.138	10.100.0.2	Ubuntu 24.04 LTS	Development server. Docker 29.1.3 / Compose 2.40.3.
`sbsdash-server-prod`	Prod	NYC2 / default-nyc2	8 GB / 160 GB	107.170.72.145	10.100.0.3	TBC	Production application server; tenant surfaces. Docker 29.1.3 / Compose 2.40.3. Administered via jump host (§3.3).
`monitor-servers`	Prod	NYC2 / default-nyc2	4 GB / 120 GB	162.243.28.132	10.100.0.4	Ubuntu 26.04 LTS	Observability — Grafana 13.1.0, Prometheus, nginx 1.28.3, TLS live at monitor.sbsdash.com. Docker 29.1.3 / Compose 2.40.3. De facto SSH bastion (§3.3).
`sbs-wiki`	Prod	**NYC1 / default-nyc1** ⚠	2 GB / 60 GB	167.172.135.88	10.116.0.2	TBC	Documentation portal — BookStack; wiki.sbsdash.com.
`Dev-Gitlab-server` ⚠	**Unregistered**	VPC 10.120.0.x	TBC	138.197.222.154	TBC	TBC	**Undocumented droplet currently receiving the `sbsdash.com` apex A record.** Disposition pending (Build Register item 22).

⚠ Two inventory nonconformances: (a) `sbs-wiki` region (§2.3); (b) `Dev-Gitlab-server` exists outside the two-project inventory yet holds production DNS — see §2.2.2.

2.2.2 Nonconformance — Production DNS Target (CRITICAL)

The apex record `sbsdash.com` → 138.197.222.154 currently resolves to `Dev-Gitlab-server`, not to `sbsdash-server-prod` (107.170.72.145). The primary tenant domain therefore points at an uninventoried development host. Required action (Build Register item 16), one of:

- **(a) Repoint** — move the apex A record to `sbsdash-server-prod` once its proxy/TLS stack is ready (Chapter 5), and either decommission `Dev-Gitlab-server` or inventory it properly under the Dev project with its own firewall posture; or

- **(b) Hold** — if the GitLab box is intentionally serving a temporary landing page, record it in this inventory, apply the Dev firewall posture, and schedule the repoint as a dated cutover.

Until resolved, `Dev-Gitlab-server` is inside the security boundary of the engagement and subject to all INF-family controls.

2.2.3 Project Topology Rules

- **Promotion path** — configurations are built and verified in Dev, then promoted to Prod exclusively through the infrastructure-as-code workflow (Chapter 10). No manual, console-driven resource creation in Prod; console access to Prod is break-glass only and logged.
- **No shared resources (target)** — no droplet, database, volume, or network resource is a member of both environments, and Dev and Prod reside on separate VPC networks with no peering. **As built, this rule is not met:** Dev, Prod, and monitor droplets all share `default-nyc2` (10.100.0.0/20). See §3.1 for the remediation decision.
- **No production data in Dev** — Dev operates on synthetic or sanitized data only. Tenant data, Compartment data, and audit-log content never leave the Prod boundary.
- **Naming convention** — new resources carry environment-prefixed names (`dev-` / `prod-`). Existing droplet names are grandfathered in the Build Register; renaming may occur at a maintenance window (renames are cosmetic and zero-downtime).

2.3 Region Selection

Primary region: **NYC2 (New York)**.

Rationale:

- **Latency to operating footprint** — SBS headquarters is in Miami with operations concentrated on the U.S. East Coast and Puerto Rico; the New York regions are the lowest-latency full-service option for the Miami–New York–San Juan corridor.
- **Data residency** — all tenant data, Compartment data, and audit-log content remain within the continental United States.
- **Single-region baseline** — both environments deploy in NYC2. Backup archives replicate to SFO3 for geographic redundancy (Chapter 14); no live workload runs outside NYC2.

Nonconformance — sbs-wiki (NYC1). DigitalOcean VPC networks are region-scoped: the NYC1 wiki droplet cannot join any NYC2 VPC. As built, the monitor scrapes the wiki's node_exporter **over the public internet, unencrypted**, admitted by a ufw rule scoped to the monitor's public IP. Metric content is low-sensitivity (CPU/RAM/disk), so this is an accepted interim state — but it is eliminated, not mitigated, by the wiki rebuild into NYC2 (Build Register item, carried from prior revision). Sequencing: rebuild **before** VPC enrollment and firewall standardization (Chapter 3); rebuilding after DNS/TLS/proxy work would force rework of all three.

Managed-service availability. NYC2 predates some newer DigitalOcean managed-service capacity. Availability of Managed Databases (PostgreSQL, MongoDB, Redis) in NYC2 is to be confirmed before Chapter 7 is finalized; the fallback design is self-hosted data services under

Docker Compose on dedicated droplets within the Prod VPC.

2.4 Environment Topology

Attribute	Dev (Staging)	Prod
DigitalOcean Project	SBS The Dash Dev	SBS The Dash Prod
Compute	`Dev-SBS-server`	`sbsdash-server-prod`, `monitor-servers`, `sbs-wiki` (pending NYC2 rebuild)
VPC (target)	Dedicated Dev VPC (NYC2)	Dedicated Prod VPC (NYC2), no peering with Dev
VPC (as built)	default-nyc2, shared ⚠	default-nyc2, shared ⚠ (wiki: default-nyc1)
OS	Ubuntu 24.04 LTS	Mixed: 26.04 (monitor), others TBC — standardization decision pending (Build Register item 21)
Public surfaces	None (target); as built, `sbsdash.com` apex resolves to a Dev-side host ⚠ (§2.2.2)	sbsdash.com, admin., clients. → `sbsdash-server-prod` (target); monitor.sbsdash.com → `monitor-servers` (**live**); wiki.sbsdash.com → `sbs-wiki`
Data services	PostgreSQL, MongoDB, Redis — reduced sizing, synthetic data (not yet installed)	PostgreSQL, MongoDB, Redis — production sizing, per-Compartment separation (Chapter 7) (not yet installed)
Observability	node_exporter on VPC-private IP (deployed)	Full stack on `monitor-servers` (deployed and verified 2026-07-17); node_exporter on prod/wiki (deployed)
Documentation	—	BookStack on `sbs-wiki` (deployed)
Runtime toolchain	Docker 29.1.3 / Compose 2.40.3 (deployed). Node.js (pnpm), Rust (Cargo), Python pending — versions pinned per Chapter 11 at install	Same
Source control	Single GitHub repository set (platform code + IaC), branch-protected per INF-03 — repository setup pending	Same; Prod deploys from protected release branches only

Toolchain versions are pinned identically across both environments (Chapter 11) so that Dev verification is meaningful for Prod promotion — version drift between environments is treated as a build defect. The current OS mix (24.04 vs 26.04) is a tracked instance of exactly this class of drift.

2.5 Server Users and Host Access

Target (INF-04): named user accounts per individual; no shared accounts; no direct root login (`PermitRootLogin no`); SSH key authentication only, password authentication disabled at the

daemon; privilege elevation via sudo, logged, without blanket NOPASSWD; authoritative user list maintained in the IaC repository (Chapter 10); administrative access paths and IdP enforcement per Chapter 9.

As built (2026-07-17/18) — nonconformant:

- Root SSH login is enabled fleet-wide and in active use.
- A single SSH keypair (`id_ed25519`, one operator workstation) is authorized for both `root` and `sbsRoot` on all droplets — one shared identity for the entire fleet.
- Shared `sbsRoot` account (uid 1000) exists on prod, dev, and monitor with `NOPASSWD:ALL` sudo and `docker` group membership (docker group membership is root-equivalent).
- Prod is reachable only via jump host through `monitor-servers` (§3.3) — the one element of current practice retained in the target design.

Remediation (Build Register item 17): create per-person named users with individually generated keys; disable root SSH; replace blanket NOPASSWD with scoped sudo (or passworded sudo); retire or restrict `sbsRoot` to break-glass with vaulted credentials; capture the authorized-user list in IaC. Until complete, individual attribution of host actions is impossible, which also degrades MON-01 (unified audit trail).

Revision #2
Created 2026-07-17 23:22:01 UTC by SBS Admin
Updated 2026-07-18 05:08:06 UTC by SBS Admin