Network Architecture

VPC design, subnets, internal routing, egress control, firewall rules

Chapter 3 — Network Architecture

3.1 VPC Design

Target: two dedicated VPC networks, one per environment, both in NYC2, no peering:

VPC	Environment	Members
`prod-vpc-nyc2`	Prod	`sbsdash-server-prod`, `monitor-servers`, `sbs-wiki` (post NYC2 rebuild)
`dev-vpc-nyc2`	Dev	`Dev-SBS-server`

As built: neither dedicated VPC exists. Current membership:

VPC (actual)	CIDR	Members
`default-nyc2`	10.100.0.0/20	`Dev-SBS-server` (.2), `sbsdash-server-prod` (.3), `monitor-servers` (.4) — **Dev and Prod share one flat private network** ⚠
`default-nyc1`	10.116.0.0/20	`sbs-wiki` (.2) — unreachable from NYC2 private networking
10.120.x	—	`Dev-Gitlab-server` (§2.2.2)

Constraint driving the decision: a DigitalOcean droplet's VPC assignment is fixed at creation — moving a droplet to a new VPC requires snapshot-and-rebuild. Two paths (Build Register item 18):

- **(a) Rebuild into dedicated VPCs — recommended.** At the current fleet size (4–5 droplets, pre-data-services, pre-tenant-data) rebuilds are cheap: snapshot, re-create in target VPC, re-run configuration. Cost grows sharply once databases and Compartments are live. This path restores the "no shared network" rule of §2.2.3 as a structural guarantee. The wiki NYC2 rebuild is already mandatory, so path (a) folds it into one rebuild wave.
- **(b) Accept shared default-nyc2** and compensate with strict host firewalls on every droplet. Dev↔Prod isolation then rests entirely on firewall rulesets rather than network non-adjacency — a weaker claim, and a permanent audit caveat against INF-01.

This specification proceeds on path (a); §3.2 onward describes the target VPCs. If path (b) is elected, §3.1–3.2 are amended by Change Order.

DigitalOcean VPC model notes (unchanged): a VPC is a single flat private network per region — no user-defined subnets, route tables, or security groups; intra-VPC segmentation is enforced by Cloud Firewalls (tag-scoped) plus host firewalls (§3.5, §3.6). VPCs are region-scoped, which is the technical basis for the wiki rebuild requirement.

3.2 Address Plan

Network	CIDR	State
`prod-vpc-nyc2`	10.10.0.0/20	Target — created during rebuild wave
`dev-vpc-nyc2`	10.20.0.0/20	Target — non-overlapping with Prod and with the legacy 10.100/10.116/10.120 ranges, so logs and tooling remain unambiguous during migration
`default-nyc2` (10.100.0.0/20), `default-nyc1` (10.116.0.0/20), 10.120.x	—	Legacy — vacated at end of rebuild wave

Private IPs are recorded in the Build Register per droplet at VPC enrollment; the authoritative address inventory lives in IaC state (Chapter 10), not in this document, to avoid drift.

3.3 Internal Routing and Service Communication

All service-to-service traffic between Prod droplets transits the VPC private interface (`eth1`), never public IPs.

- **Application → data services (target)** — the application stack on `sbsdash-server-prod` reaches PostgreSQL, MongoDB, and Redis exclusively on VPC-private addresses; data services bind to the private interface only, with no public listener (verified at LH-SBS-INST-001 network gates).
- **Prometheus scrape paths (deployed, verified 2026-07-17)** — `monitor-servers` scrapes node_exporter on prod (10.100.0.3:9100) and dev (10.100.0.2:9100) over private networking; exporters bind private-only and are unreachable publicly. On the monitor itself, Prometheus (9090), Grafana (3000), and local node_exporter (9100) bind 127.0.0.1 only, fronted by nginx TLS — confirmed by `ss -tlnp` verification. This matches the target design and is marked **verified** for the NYC2 members.
- **Wiki scrape (interim nonconformance)** — `sbs-wiki` (NYC1) is scraped on its **public IP** (167.172.135.88:9100), unencrypted, admitted only from the monitor's public IP by ufw rule. Accepted temporarily (metrics only, low sensitivity); eliminated by the NYC2 rebuild, after which the wiki exporter rebinds to its private address like the rest of the fleet.
- **Jump-host / bastion pattern (as built, retained in design)** — `sbsdash-server-prod` accepts SSH only via ProxyJump through `monitor-servers` (`ssh -J`). This makes the monitor a de facto bastion. Consequences now normative: the monitor carries the tightest SSH admission (admin allowlist, §3.5); its compromise radius includes prod SSH reachability, so it receives priority hardening, patching, and log scrutiny (MON-01/MON-02); bastion SSH sessions are logged. Whether the bastion role remains on the monitor or

moves to a dedicated minimal droplet is revisited at the rebuild wave.
- **DNS resolution** — droplets use systemd-resolved against DigitalOcean resolvers; internal addressing uses IaC-managed `/etc/hosts` entries (small fixed fleet). Revisit if the fleet exceeds ~10 hosts.

Inter-Compartment note (unchanged): Compartment network isolation is realized **within** `sbsdash-server-prod` at the Docker Compose network level — one dedicated compose network per Compartment, no shared bridges, cross-Compartment traffic denied by non-membership. Docker 29.1.3 / Compose 2.40.3 are now present on prod, dev, and monitor (2026-07-18), so this layer is buildable. Container-level detail in Chapter 6.

3.4 Egress Control

DigitalOcean provides no managed NAT/egress gateway; egress control is enforced at the host firewall layer:

- **Restricted-egress allowlist at the host firewall** on each Prod droplet, permitting: TCP 443 to Anthropic API endpoints; TCP 443 to GitHub (deploy pulls, pinned to published ranges, refreshed by IaC); TCP 443 to OS package mirrors and container registries enumerated in Chapter 11; TCP/UDP 123 (NTP); TCP 443 to Let's Encrypt ACME endpoints; DNS 53 to DigitalOcean resolvers only.
- All other outbound denied and **logged** — egress-deny lines ship to the observability pipeline (Chapter 12) as a MON-02 anomaly input. Unexpected egress from a Compartment container is a primary indicator for AI-02 (tool allowlisting) and LLM01 (prompt-injection exfiltration) violations.
- Compartment containers inherit a **narrower** egress set than the host: Anthropic API plus explicitly allowlisted per-Compartment integration endpoints, enforced via per-compose-network firewall chains, IaC-managed and change-controlled.

As built: no egress control exists on any droplet (all outbound open). Egress enforcement lands with the host-firewall baseline (§3.6, Build Register item 20).

3.5 Cloud Firewall Rules (DigitalOcean Cloud Firewalls, tag-scoped)

As built: no Cloud Firewalls exist. Host-level ufw is active on monitor and wiki only; **prod and dev have no active firewall at any layer** — their public interfaces are protected solely by service bind addresses. This is the most exposed element of the current state and is remediated first in the firewall workstream (Build Register item 19).

Target — four firewalls, attached by droplet tag:

FW `prod-app` → tag `prod` (`sbsdash-server-prod`)

Dir	Proto/Port	Source / Destination	Purpose
In	TCP 443	0.0.0.0/0, ::/0	Tenant surfaces via reverse proxy

Dir	Proto/Port	Source / Destination	Purpose
In	TCP 80	0.0.0.0/0, ::/0	ACME HTTP-01 + 301→443 only
In	TCP 22	`monitor-servers` private IP only	SSH via bastion (§3.3) — no direct public SSH
In	TCP 9100 + app exporters	`monitor-servers` private IP	Prometheus scrape
Out	Per §3.4	—	Restricted egress

FW `prod-monitor` → tag `grafana` (`monitor-servers`)

Dir	Proto/Port	Source	Purpose
In	TCP 443	0.0.0.0/0	monitor.sbsdash.com (Grafana behind IdP, Chapter 9)
In	TCP 80	0.0.0.0/0	ACME + redirect
In	TCP 22	Admin allowlist IPs (Build Register)	Bastion SSH — sole public SSH entry to Prod
—	—	—	3000/9090/9100 bind 127.0.0.1 — no public rule exists (verified as-built)

FW `prod-wiki` → tag `wiki` (`sbs-wiki`, post-rebuild)

Dir	Proto/Port	Source	Purpose
In	TCP 443	0.0.0.0/0	wiki.sbsdash.com (BookStack behind SSO)
In	TCP 80	0.0.0.0/0	ACME + redirect
In	TCP 22	`monitor-servers` private IP	SSH via bastion
In	TCP 9100	`monitor-servers` private IP	node_exporter scrape (rebinds private post-rebuild)

FW `dev-app` → tag `develop` — SSH from admin allowlist; web surfaces admit admin allowlist only (no production DNS points at Dev — enforcement of this depends on resolving §2.2.2).

Deviation from prior draft, deliberate: direct public SSH to `prod-app` is **removed** in favor of bastion-only SSH, codifying the as-built jump-host pattern — it is the stronger posture. Rule hygiene unchanged: firewalls are IaC-defined; console edits are drift and reverted; the admin SSH allowlist is a named IaC variable reviewed at each IAM-06 access review.

3.6 Host Firewall

Cloud Firewalls filter at the platform edge; a host firewall on each droplet provides defense-in-depth and the per-container egress allowlist (§3.4), which Cloud Firewalls cannot express.

As built: ufw on monitor (22/80/443 in) and wiki (22/80/443 + 9100-from-monitor in); ufw inactive on prod and dev; no egress rules anywhere; no nftables rulesets deployed.

Target: nftables on all droplets — required because per-Compartment egress chains (one chain per compose network) and logged default-deny outbound are beyond ufw's expressible policy. Migration order: (1) activate baseline host firewall on prod and dev immediately (ufw acceptable as a stopgap — inbound mirror of the Cloud Firewall admit set); (2) replace fleet-wide with IaC-rendered nftables rulesets at the rebuild wave; (3) add per-Compartment chains when Compartment compose networks are created (Chapter 6). Base policy: inbound default-deny mirroring the cloud firewall, outbound default-deny with the §3.4 allowlist, established/related admitted. Rulesets are rendered from IaC templates and versioned; the live ruleset hash is a monitored value (MON-01) so tampering surfaces as drift.

3.7 Verification

Network gates (LH-SBS-INST-001): public-listener scan — no data-service or Prometheus ports on public interfaces (**passes today** on monitor per `ss -tlnp` evidence; wiki exporter public listener is a tracked exception until rebuild); Dev↔Prod isolation probe — no private path between environments (**fails today**: shared default-nyc2; passes after rebuild wave); egress-allowlist test — denied-domain attempt logged (**not testable today**: no egress control); per-Compartment cross-network denial test (pending Chapter 6 build); bastion path test — prod SSH unreachable except via monitor. INF-01 is the controlling SEC-001 requirement throughout.