Chapter 10 — Infrastructure as Code

10.1 Current State and Scope

As of 2026-07-18 **no IaC exists**: all resources to date (droplets, DNS records, ufw rules, Grafana/Prometheus configs) were created by console or shell, documented after the fact in SERVER_CHANGELOG.md. That changelog discipline is the right instinct executed at the wrong layer — this chapter moves the source of truth from *prose describing what was done* to *code that does it*, with the changelog demoted to narrative context. Scope: DigitalOcean resources, DNS zone (post-cutover, §4.2), host configuration, firewall rulesets, Vault policy/mount layout, compose templates, and monitoring config. Out of scope: application source code (its CI/CD is platform-repo concern; supply-chain controls shared with Chapter 11).

10.2 Tooling

Two tools, sharply divided by responsibility — resisting the single-tool temptation because each is poor at the other's job:

Layer	Tool	Manages
Provisioning (things that exist)	**OpenTofu** (Terraform-compatible, MPL-licensed — no BSL exposure) with the `digitalocean` provider	Droplets, VPCs, Cloud Firewalls, Volumes, Spaces buckets + keys, DO project membership, DNS records via the post-cutover provider API (§4.2), reserved IPs
Configuration (state of things that exist)	**Ansible**	OS baseline (users/keys post-item-17, sshd config, unattended-upgrades), nftables rulesets rendered from templates (§3.6), nginx server blocks + TLS params (§5.3), node_exporter binds, Vault server config, compose file rendering + deploy (§6.7), internal-CA leaf distribution (§7.5)

Supporting: **gitleaks** (pre-commit + CI, §8.6), **promtool**/`nginx -t`/`vault policy fmt` as validation steps in CI, **Infracost** not adopted (fleet too small to justify).

State backend: OpenTofu state in the `sbs-dash-backups` Spaces bucket (S3-compatible backend) with state locking via DO Spaces conditional writes; state contains secrets-adjacent values (IPs, IDs — not credentials, which stay in Vault per §8.5) and the bucket is private + encrypted (§7.3).

Ansible is stateless by design; its "state" is the repo.

10.3 Repository Structure

One infrastructure repository (`sbs-dash-infra`) in the GitHub organization — separate from application code so infra reviewers and app reviewers are distinct CODEOWNERS populations:

```
sbs-dash-infra/
├── tofu/
│   ├── environments/
│   │   ├── prod/          # prod VPC, droplets, firewalls, volumes
│   │   └── dev/           # dev VPC, droplet
│   ├── modules/
│   │   ├── droplet/       # standard droplet: VPC member, tags, monitoring
│   │   ├── firewall/      # tag-scoped Cloud Firewall pattern (§3.5)
│   │   └── spaces/        # bucket + scoped key pattern (§7.3)
│   ├── dns/               # zone records post-cutover (§4.3 target table)
│   └── globals/           # admin SSH allowlist, region, naming — the named
│                          #   variables referenced throughout this spec
├── ansible/
│   ├── inventory/         # generated FROM tofu output — single source for IPs
│   ├── roles/
│   │   ├── base/          # users, sshd, patching (§2.5 target, INF-05)
│   │   ├── nftables/      # §3.6 templates incl. per-Compartment chains
│   │   ├── nginx-edge/    # §5.3 TLS params + §5.5 header template
│   │   ├── monitoring/    # prometheus jobs, exporter binds, dashboards
│   │   ├── vault/         # §8.2 server config
│   │   └── compose/       # §6.4 Compartment template renderer
│   └── playbooks/
├── vault/policies/        # §8.3 policy HCL, applied via CI
├── .gitleaks.toml
└── CODEOWNERS
```

Repository controls (INF-03, shared baseline with Ch. 11): default branch protected — PR required, ≥1 review from CODEOWNERS, no force-push, no direct commit, signed commits required, CI green as merge condition; admin bypass disabled; the GitHub org and repo live under SBS ownership with Ledger Hub engineers as members (same revocability principle as §2.1).

10.4 Change Workflow

Every infrastructure change follows one path — the §2.2.3 promotion rule made concrete:

1. **Branch + PR** against `sbs-dash-infra`. PR description states intent and links the Build Register item or Change Order where applicable (LH-SBS-CO- series for scope-level changes; routine parameter changes need only the PR).
2. **CI validation** on PR: `tofu validate` + `tofu plan` (both environments, plan output posted to the PR), `ansible-lint`, template syntax checks (`nginx -t` in container, `promtool check config`), gitleaks sweep. A plan showing destruction of stateful resources (Volumes, databases, Spaces buckets) requires an explicit `destructive-change` label + second reviewer.
3. **Review** by CODEOWNERS — for firewall, Vault-policy, or IAM-adjacent paths, the reviewer set includes the security owner (Alexandra Del Rey routing).
4. **Apply to Dev** on merge: CI runs `tofu apply` (dev workspace) + Ansible against dev inventory automatically.
5. **Apply to Prod** as a **manually approved CI job** (GitHub environment protection, named approvers) — never from an operator laptop. The CI identity authenticates to Vault via its own AppRole (§8.5) for any secret-touching render; the DO token it uses is the pipeline-scoped token of §2.1.
6. **Emergency path:** break-glass console/shell changes are permitted to restore service (MON-05 incident context), and must be back-ported to the repo within 24 h — the next drift run (§10.5) enforces this mechanically by flagging anything that wasn't.

10.5 Drift Detection

Drift — divergence between repo-declared state and live state — is the failure mode this whole chapter exists to prevent, and every preceding chapter's "console edits are drift and reverted" clauses land here as mechanism:

- **Provisioning drift:** scheduled CI job runs `tofu plan -detailed-exitcode` nightly per environment. Exit code 2 (changes present) → alert to the ops Slack channel (Ch. 12 routing) with the plan diff. Triage outcomes: revert the live change (apply), or adopt it (PR that codifies it) — silent coexistence is not an outcome.
- **Configuration drift:** nightly `ansible-playbook --check --diff` against full inventory; any changed-task output alerts identically. Ansible runs are also **idempotent enforcement** — the weekly scheduled *real* run re-asserts declared state, so unrecorded manual edits have a maximum lifetime of seven days.
- **Ruleset integrity:** live nftables ruleset hash and running-container-vs-compose comparison (§3.6, §6.7) are exported as Prometheus metrics; mismatch alerts in minutes rather than nightly — these two are the fastest-moving tamper indicators (MON-01 inputs).
- **DNS drift** (post-cutover): nightly diff of provider-API zone dump vs. `tofu/dns/` — catches registrar-console edits (§4.7).
- Drift alerts are **not** auto-remediated on Prod (auto-apply of a malicious or mistaken repo state is its own attack path); human triage stands between detection and correction, with the weekly enforcement run as the backstop.

10.6 Bootstrap and Migration of As-Built Resources

The existing fleet was built by hand; it is **imported, not rebuilt**, except where the §3.1 rebuild wave already mandates recreation:

1. Stand up repo, branch protection, CI skeleton, gitleaks (already Build Register item 47).
2. `tofu import` current resources: 4–5 droplets, existing ufw-era firewall facts recorded as-is, Spaces buckets when created. Import produces an honest baseline whose first `plan` output *is the formal nonconformance list* — the delta between as-built and this specification, machine-generated.
3. Author target state per this spec (dedicated VPCs, Cloud Firewalls, prod-vault droplet); the §3.1 rebuild wave then executes as reviewed PRs — the rebuild becomes the first real exercise of the §10.4 workflow rather than another manual event.
4. Ansible baseline role applied fleet-wide, which *is* the execution vehicle for item 17 (per-person users, sshd hardening) and item 20 (host firewalls).
5. SERVER_CHANGELOG.md thereafter records narrative/context only; "what exists" questions are answered by the repo and its state, and the changelog's §1.3 role is updated accordingly.

10.7 Verification

Gates (LH-SBS-INST-001): repo exists under SBS org with branch protection verified (settings API dump); direct push to default branch rejected (negative test); PR without CI green cannot merge; `tofu plan` clean on both environments (zero drift at gate time); nightly drift jobs scheduled and alerting path tested (induced drift: manual test-tag change → alert received → reverted); Prod apply requires named approval (job history evidence); emergency-change backport rule exercised in drill; gitleaks blocks a seeded dummy secret in pre-commit and CI; inventory generated from tofu output matches live fleet.

Revision #1
Created 2026-07-18 11:45:48 UTC by SBS Admin
Updated 2026-07-18 11:47:22 UTC by SBS Admin