BUOYANT
Creators of LINKERD

Linkerd Production Readiness Pre-Launch Checklist

1. Installation and GitOps: No pets allowed!

- [ ] Install and managed Linkerd via Helm with all configuration stored in Git and applied through a GitOps workflow (Argo CD, Flux, or Terraform with Helm)
- [ ] Document a rollback procedure and regularly tested it (if you don't test it, it's not actually a procedure!)

2. Certificates and identity: This is your security boundary

- [ ] Create the trust anchor and issuer certificates outside the CLI, and their keys are under proper control
- [ ] Ensure cert-manager, or an equivalent CA workflow, issues and automatically rotates the issuer certificate
- [ ] Document and test procedure for both issuer and trust anchor rotation, including restarting the control plane and workloads where required

3. Control plane high availability: Design for failure

- [ ] Run Linkerd in High Availability mode
- [ ] Set resource requests and limits for all control plane pods, and monitor these values and adjust them as the cluster grows
- [ ] Monitor the Linkerd components to observe their health over time

4. Proxy and workload tuning: Respect the edge cases

- [ ] Identify high-traffic, latency-sensitive, or stateful workloads, such as databases, Kafka, or large monoliths, and reviewed their proxy requirements
- [ ] Increase CPU and memory for proxies using global or per-workload overrides where necessary
- [ ] Mark ports that carry application-level TLS or mutual TLS as opaque or skipped so Linkerd does not attempt to inspect or interfere
- [ ] Explicitly configure protocols such as gRPC so you get correct metrics and behavior rather than relying solely on protocol detection

5. Observability and alerts: One pane for when it hurts

- [ ] Feed a working metrics pipeline (e.g., Prometheus) into a visualization tool like Grafana
- [ ] Centrally aggregate logs from both applications and proxies centrally and searchable by service, namespace, and pod
- [ ] Create a single incident dashboard that shows control plane status, overall success rate and latency, and the health of your most important workloads
- [ ] Configure basic alerts for control plane availability, mesh success rate and latency, and upcoming certificate expirations

6. Process and environment parity: Practice before it counts

- [ ] Use the same Helm and GitOps workflow across dev, staging, and prod for Linkerd installs and upgrades
- [ ] Practice at least one control plane upgrade and one issuer rotation in a lower environment, including rolling workloads and validating behavior
- [ ] Keep any DR or secondary clusters reasonably in sync in terms of versions, configuration, and certificates, and you have run at least a basic failover or smoke test
- [ ] Explicitly include developer-facing or internal environments in your production definition where other teams depend on your platform