Production readiness checklist

From ArchitectureCompass (https://architecturecompass.com). Free to use and adapt.

Run this with the team that will operate the service, before it takes real traffic. Many items come from well-known public outages; see the failure library at https://architecturecompass.com/case-studies/failure-library/

Ownership

- [ ] A named team owns the service and is on call for it.
- [ ] Escalation paths and contacts are documented outside the system itself.
- [ ] The service is listed in the service catalog with its dependencies.

Observability

- [ ] Request rate, error rate and latency (p50, p95, p99) are measured and visible on a dashboard.
- [ ] Alerts fire on user-facing symptoms and page a human only when action is needed.
- [ ] Logs are structured, centralized and carry a correlation or trace ID.
- [ ] Distributed tracing covers calls to every dependency.
- [ ] Monitoring keeps working if this service, its network or its main database fails.

Reliability

- [ ] Every outbound call has a timeout and a bounded retry policy with backoff and jitter.
- [ ] Circuit breakers or fallbacks protect against slow or failing dependencies.
- [ ] The service runs in more than one availability zone, or the risk is accepted in writing.
- [ ] Health checks distinguish "alive" from "ready to take traffic".
- [ ] Load has been tested to at least the expected peak plus headroom.
- [ ] Graceful shutdown drains in-flight work.
- [ ] Recovery after an outage is gradual (traffic ramp-up, cache warm-up), not all at once.

Data

- [ ] Backups run automatically, failures alert someone, and a full restore has been tested recently.
- [ ] Point-in-time recovery is available for critical data.
- [ ] Destructive scripts and bulk operations require confirmation and use soft deletes where possible.
- [ ] Data migrations are reversible or have a tested rollback plan.

Change and deployment

- [ ] Deployments are automated and identical on every server.

- [ ] Code, configuration and content changes all roll out in stages, with health checks between stages.
- [ ] Rollback is one step and has been practised.
- [ ] Risky features are behind flags that can be turned off without a deployment.
- [ ] Tools that remove capacity or change infrastructure enforce minimums and rate limits.

Security

- [ ] Secrets come from a secrets manager and can be rotated without downtime.
- [ ] Dependencies are scanned, and there is a process to patch critical vulnerabilities quickly.
- [ ] Service-to-service calls are authenticated.
- [ ] Access to production is audited and least-privilege.

Capacity and cost

- [ ] Capacity limits and service quotas (including cloud account limits) are known, monitored and raised ahead of peaks.
- [ ] Autoscaling limits are set deliberately, and the provisioning path itself has been load-tested.
- [ ] Cost per request or per customer is estimated and tracked.

Incident readiness

- [ ] Runbooks cover the top five likely failures.
- [ ] The team has run at least one game day or failure drill.
- [ ] A status page or customer communication channel exists and doesn't depend on this service.
- [ ] A blameless post-incident review process is agreed.