ASSESSMENT

Kubernetes production readiness worksheet

Use this worksheet to turn the checklist into evidence. Mark an item as done only when the team can point to a concrete implementation, test, runbook, policy, dashboard, or decision record.

WORKLOAD / SERVICE

ENVIRONMENT

REVIEW DATE

REVIEW OWNER

Status
Choose **Yes** only with evidence. Use **Unsure** when the implementation needs review.

Evidence
Link to manifests, code, tests, dashboards, runbooks, policies, tickets, or architecture decisions.

Owner
Assign the person who can prove the item, fix the gap, or make the trade-off decision.

Stage 1: Your Application

Application Behavior

ITEM 1.1.1

The application logs to `stdout` and `stderr`

Evidence: logging format, runtime config, and central log collection path.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 1.1.2

Configuration is read from environment variables or files

Evidence: ConfigMap, mounted config file, environment contract, or deployment convention.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 1.1.3

The application handles SIGTERM and shuts down gracefully

Evidence: signal handler, drain behavior, timeout, and test or runbook showing graceful termination.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 1.1.4

The application exposes health signals

Evidence: readiness, liveness, or startup endpoint/command and the behavior each signal represents.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 1.1.5

The application doesn't store state on the local disk

Evidence: persistent state is externalized or explicitly handled through stable storage.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 1.1.6

The application handles long-lived connections correctly

Evidence: reconnect, drain, and shutdown behavior for gRPC, WebSockets, HTTP/2, or pools.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Container Image

ITEM 1.2.1

The container image contains only what is needed to run the application

Evidence: Dockerfile, multi-stage build, image size, and runtime dependency list.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 1.2.2

Image tags are stable and `:latest` is avoided

Evidence: immutable tag or digest, release mapping, and rollback process.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Stage 2: Your Kubernetes Manifests

Runtime Contract

ITEM 2.1.1

Readiness, liveness, and startup probes are defined

Evidence: probe settings and clear distinction between ready, live, and started.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.1.2

Resource requests and limits are set

Evidence: CPU and memory requests, memory limit, and documented sizing rationale.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.1.3

Ephemeral storage usage is bounded

Evidence: ephemeral-storage requests/limits or bounded writable paths such as `emptyDir`.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Rollouts And Configuration

ITEM 2.2.1

Rolling update settings are explicit

Evidence: maxUnavailable, maxSurge, minReadySeconds, progressDeadlineSeconds, and revision history.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.2.2

The workload tolerates old and new Pods running together

Evidence: compatibility plan for API, schema, queue, and data format changes.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.2.3

ConfigMap and Secret updates have a reload strategy

Evidence: rollout trigger, mounted-file reload strategy, or immutable config convention.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Placement And Disruption

ITEM 2.3.1

Non-root, read-only filesystem, and dropped capabilities are configured

Evidence: securityContext with non-root, readOnlyRootFilesystem, dropped capabilities, and seccomp.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.3.2

A PodDisruptionBudget is defined

Evidence: PDB selector, minAvailable or maxUnavailable, and node-drain behavior.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.3.3

Pods are spread across nodes and zones

Evidence: topologySpreadConstraints, topology labels, and observed placement across failure domains.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Secrets And Metadata

ITEM 2.4.1

Secrets are mounted as volumes, not passed as environment variables

Evidence: volume mounts or external secret integration, not env-var secret injection.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.4.2

Resources have meaningful labels

Evidence: consistent app, owner, environment, version, and cost labels.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 2.4.3

Manifests use supported Kubernetes API versions

Evidence: API version audit against the cluster version and upgrade policy.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Stage 3: Your Security

Runtime Access Controls

ITEM 3.1.1	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Pod Security Standards are enforced at the namespace level Evidence: namespace labels, audit/warn/enforce modes, and rollout plan.	☐ Yes ☐ No ☐ Unsure			

ITEM 3.1.2	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Each workload has a dedicated ServiceAccount with minimal RBAC Evidence: named ServiceAccount, Role/RoleBinding, and least-privilege review.	☐ Yes ☐ No ☐ Unsure			

ITEM 3.1.3	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Network access is restricted with NetworkPolicy Evidence: ingress/egress policies, namespace defaults, and tested allowed flows.	☐ Yes ☐ No ☐ Unsure			

Supply Chain And Admission Control

ITEM 3.2.1	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Container images are scanned and pulled from a trusted registry Evidence: scanner, registry policy, CVE threshold, and exception process.	☐ Yes ☐ No ☐ Unsure			

ITEM 3.2.2	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Admission policies validate every manifest Evidence: Kyverno, Gatekeeper, or equivalent policy-as-code in audit/enforce mode.	☐ Yes ☐ No ☐ Unsure			

Cloud Access And Secrets

ITEM 3.3.1	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Workloads use workload identity for cloud resources Evidence: IRSA, Workload Identity, Entra Workload ID, or equivalent short-lived credentials.	☐ Yes ☐ No ☐ Unsure			

ITEM 3.3.2	STATUS	EVIDENCE	OWNER	NOTES / NEXT ACTION
Secrets live in an external secret store Evidence: external secret manager, rotation process, and Kubernetes integration.	☐ Yes ☐ No ☐ Unsure			

Stage 4: Scaling

Scaling Model

ITEM 4.1.1

The application can scale horizontally

Evidence: stateless behavior, replica test, and constraints for stateful components.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 4.1.2

Autoscaler bounds and scale-down behavior are explicit

Evidence: min/max replicas, HPA or KEDA settings, and scale-down policy.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 4.1.3

Vertical scaling is understood as an option, not a default

Evidence: VPA or manual right-sizing decision and known HPA/VPA interactions.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Resource Pressure

ITEM 4.2.1

Resource requests are based on real usage data

Evidence: metrics, load profile, VPA recommendations, or production usage review.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 4.2.2

Priority classes express what should survive resource pressure

Evidence: PriorityClass definitions and mapping of workloads to business priority.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Traffic And Validation

ITEM 4.3.1

Scale-down drains traffic cleanly

Evidence: scale-down test, graceful shutdown behavior, and no dropped in-flight requests.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 4.3.2

The scaling path has been load-tested

Evidence: ramp test, p99/error metrics, HPA behavior, and scale-down validation.

STATUS

- [ ] Yes
- [ ] No
- [ ] Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Stage 5: Going Live

Visibility

ITEM 5.1.1

The current health of the application is visible

Evidence: dashboard, RED/USE metrics, logs, traces, and alert entry points.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 5.1.2

Kubernetes Events are collected for the workload

Evidence: event collection pipeline, retention, and incident investigation path.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Recovery

ITEM 5.2.1

A rollback vs. roll-forward posture has been decided

Evidence: runbook, GitOps process, migration rules, and owner for release decisions.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 5.2.2

You know what happens when a Pod crashes or a node dies

Evidence: failure test, restart visibility, rescheduling behavior, and PDB/topology validation.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

Runbooks And Cost

ITEM 5.3.1

A troubleshooting runbook exists

Evidence: runbook covering common states, logs, describe output, exec/debug, rollback, and escalation.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

ITEM 5.3.2

Cost has been reviewed, and the workload is right-sized

Evidence: CPU/memory usage review, replica review, node utilization, and recurring FinOps checkpoint.

STATUS
- ☐ Yes
- ☐ No
- ☐ Unsure

EVIDENCE

OWNER

NOTES / NEXT ACTION

This worksheet is the short version.

The full Kubernetes production readiness checklist explains every item in detail, with examples, trade-offs, and links to the relevant Kubernetes concepts.

learnkube.com/production-best-practices

Find the risks in your Kubernetes setup before they break production

If your team is preparing a production launch, migration, or internal platform review, **a LearnKube instructor can walk through this checklist with your engineers.**

hello@learnkube.com learnkube.com/production-readiness-review