SAMPLE — CLIENT DETAILS REDACTED

Kubernetes Security Baseline Review

Executive Summary & Technical Findings

CLIENT
Redacted

DATE
April 2026

ENGAGEMENT
Baseline Review

EXECUTIVE SUMMARY

Overview & Key Findings

This document summarizes the findings of a Kubernetes Security Baseline Review conducted by ClarifyIntel for Redacted. The review covered cluster configuration, workload security posture, policy gaps, and rollout readiness.

SECURITY POSTURE SCORECARD

Area	Current State	Priority	Effort
Image tag hygiene	Issues found	High	Medium
Resource limits	Issues found	High	Medium
Liveness / readiness probes	Issues found	Medium	Low
Privileged containers	No issues found	Pass	Low
Privilege escalation control	No issues found	Pass	Low
Run-as-root enforcement	No issues found	Pass	Low
RBAC over-privilege	No issues found	Pass	Low
NetworkPolicy coverage	No issues found	Pass	Low

AT A GLANCE

3	2	1	42
Total findings	High-priority	Medium-priority	Containers scanned

TECHNICAL FINDINGS

Priority Findings & Remediation Guidance

The following findings are ordered by priority. Each finding includes the observed state, the recommended change, and the expected developer impact. Findings marked High should be addressed before enforcement is enabled.

F-01 Containers using mutable image tags

High

Policy	`disallow-latest-tag` · starter policy available at github.com/clarifyintel/baseline
Observed	36% of containers (15 total) use :latest or have no image tag. Affected application namespaces: default (12 workloads). Rollbacks are unreliable and deployment provenance cannot be traced.
Recommend	Require fixed version tags or digest references in all manifests. Introduce the policy in Audit mode first to surface remaining gaps.
Dev impact	Low — teams pin image versions in manifests or Helm values.

Namespace	Workload	Kind	Containers
default	adservice	Deployment	server
default	cartservice	Deployment	server
default	checkoutservice	Deployment	server
default	currencyservice	Deployment	server
default	emailservice	Deployment	server
default	frontend	Deployment	server
default	loadgenerator	Deployment	main, frontend-check
default	paymentservice	Deployment	server
default	productcatalogservice	Deployment	server
default	recommendationservice	Deployment	server
default	shippingservice	Deployment	server
default	shoppingassistantservice	Deployment	server

F-02 Missing resource requests and limits

High

Policy	`require-resources` · starter policy available at github.com/clarifyintel/baseline
Observed	12% of containers (5 of 42) are missing resource requests and limits. Affected namespaces: default (2 workloads). Scheduling is unpredictable and node resource contention risk is high.
Recommend	Enforce resource definitions for all containers. Use Audit mode to surface the gap, then provide a guidance template based on observed metrics.
Dev impact	Low — values can be set conservatively and tuned over time.

Namespace	Workload	Kind	Containers
default	loadgenerator	Deployment	frontend-check
default	opentelemetrycollector	Deployment	otel-gateway, otel-gateway-init

F-03 Containers missing liveness or readiness probes — Medium

Policy	`require-probes` · starter policy available at github.com/clarifyintel/baseline
Observed	21% of containers (9) are missing one or both health check probes. Affected namespaces: default (3 workloads). Kubernetes cannot reliably detect unhealthy containers without them.
Recommend	Define both livenessProbe and readinessProbe for all containers. Start with readiness to avoid routing traffic to unready pods.
Dev impact	Low — probe endpoints usually already exist; just needs wiring.

Namespace	Workload	Kind	Containers
default	loadgenerator	Deployment	main, frontend-check
default	opentelemetrycollector	Deployment	otel-gateway, otel-gateway-init
default	shoppingassistantservice	Deployment	server

ROLLOUT PLAN

30-Day Recommended Actions

The following plan sequences rollout across 3 identified policies (2 High, 1 Medium) to minimise developer disruption. Enforcement does not begin until the audit phase is complete and developer guidance has been distributed.

Phase 1 Days 1–7 **Audit mode — establish visibility**

- Apply all 3 policies in Audit mode — starter policies available at github.com/clarifyintel/baseline.
- Apply the stage-1-audit kustomize overlay: kubectl apply -k policies/stages/stage-1-audit
- Review policy reports daily: kubectl get policyreport -A
- Document violation inventory by namespace and workload.
- Identify workloads that will need exceptions before enforcement.

Outcome: Full visibility into the current gap. No deployments affected.

Phase 2	Days 8–14	Developer communication & quick wins

- Send developer impact notes to affected teams.
- Fix low-friction violations: image tags, resource limits, probes.
- Open and approve exception requests using the exception model.
- Confirm violation count is trending down before Phase 3.

Outcome: Developers aware, easy fixes in progress, exceptions documented.

Phase 3	Days 15–21	Enforce low-friction controls

- Move require-probes to Enforce.
- Apply to non-production namespaces first.
- Monitor for blocked deployments and address within 24 hours.
- Confirm CI pipelines pass and communicate timeline to production teams.

Outcome: 1 policies enforced in non-production. No production impact.

Phase 4	Days 22–30	Enforce security context controls

- Move disallow-latest-tag, require-resources to Enforce.
- Apply to non-production first, then production after 48 h observation.
- Apply full enforcement overlay in production.
- Conduct 30-day review: violations, exceptions, developer feedback.

Outcome: Full baseline enforced. Exception inventory complete. Rollout documented.

About this report

This Kubernetes Security Baseline Review was prepared by ClarifyIntel for Redacted. Findings are based on static analysis of YAML manifests in the provided repository. Runtime behaviour, network policy, and RBAC are not covered in this engagement.

Starter policies for all findings in this report are available at github.com/clarifyintel/baseline — including audit/enforce overlays, exception model, and developer impact notes.

Next steps

Review findings with your platform team.

Starter policy pack:
github.com/clarifyintel/baseline

hello@clarifyintel.com
clarifyintel.com/services