Kubernetes & Containers Assessment

Montclair Digital

01 September 2026 - 05 September 2026
Prepared by Vulnotes Security Consulting
Version 1.0.0
CONFIDENTIAL

1. Document control

Version	Issued	Author	Review
1.0.0	2026-09-17	Julien Perrin	Sophie Roux / 16 September 2026

Authorized distribution

Restricted distribution. Intended roles: Lina Bernard, platform lead; Julien Perrin, assessment author; Sophie Roux, quality reviewer.

This report describes observations within the agreed scope and assessment period. Subsequent changes may affect its conclusions. Distribution is restricted to authorized recipients.

Revision history

Version 1.0.0 issued 2026-09-16. Initial assessment period 1-5 September 2026; editorial review completed 16 September 2026. No remediation retest has yet taken place.

Table of Contents

2. Executive assessment

01 / Assessment boundary

3 findings affect mc-stage-01 / orders and billing namespaces: 0 High, 1 Medium, 2 Low. The assessment reviewed the agreed configuration and application records against the approved service baseline. Correction priorities reflect the impact, required access and dependencies documented in each finding.

02 / Correction priorities	03 / Verified controls
Assign F-1 to Lina Bernard, platform lead for correction by 30 September 2026. Address F-2 and F-3 by 16 October 2026, then perform the documented acceptance checks before closure.	Workloads use a private image registry and have named owners. The reviewed ingress configuration requires encrypted client connections.

3. Assessment analysis

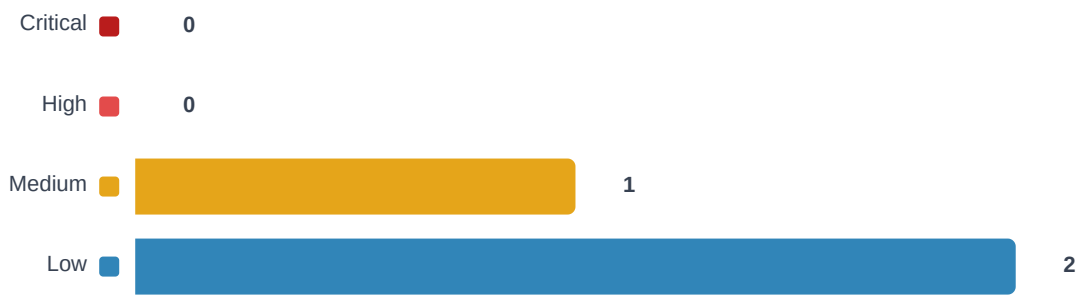
The assessment of mc-stage-01 / orders and billing namespaces identified 3 findings: 0 Critical, 0 High, 1 Medium, 2 Low. Two namespaces, eight workload manifests, associated service accounts and namespace policies. Read-only review of exported desired-state and running-state records; host access, denial-of-service tests and provider control plane excluded.

The principal improvement concerns workloads have no effective resource limits. Unexpected resource consumption could impair neighboring services and reduce the effectiveness of capacity planning. Correction ownership rests with Lina Bernard, platform lead. The priority is to reduce this exposure without interrupting the service's required business functions.

The remaining observations concern application workloads run as the root user and unused service-account tokens are mounted into workloads. They require separate acceptance criteria because changing one policy does not establish that the other controls operate effectively. Review effective runtime behavior as well as the saved configuration.

Workloads use a private image registry and have named owners. The reviewed ingress configuration requires encrypted client connections. These controls provide a basis for remediation, but their presence does not compensate for the documented exceptions. The action plan separates implementation evidence from successful retest results.

Risk distribution



4. Scope and conditions

The following register defines the assessed assets, access conditions and accountable owners.

Cluster / version	Namespace / workload	Layer assessed	Platform owner
mc-stage-01 / Kubernetes 1.34	orders and billing / eight workloads	Workload, namespace and service-account records	Lina Bernard, platform lead

Detailed scope register

Two namespaces, eight workload manifests, associated service accounts and namespace policies. Read-only review of exported desired-state and running-state records; host access, denial-of-service tests and provider control plane excluded.

Exclusions and restrictions

Only the supplied staging assets and evidence are in scope. Production changes, disruption, social engineering and third-party systems are excluded. Sampling does not establish complete coverage.

5. Assessment coverage

Record the method, reference version, observation date and supporting evidence for each assessed area. A reference mapping does not constitute certification.

Assessment area	Status	Evidence / reference
Cluster and control-plane configuration	Not tested	Provider control plane excluded
RBAC and service accounts	Gap observed	E-03: unnecessary mounted tokens
Workload security settings	Gap observed	E-01/E-02: workload settings
Namespace and network isolation	Not tested	Network isolation not assessed
Secret lifecycle	Gap observed	E-03: credential requirements
Image provenance and update process	Verified	E-00: private registry ownership

Allowed statuses: Verified, Gap observed, Not tested, Not applicable. Justify exclusions and not-applicable decisions.

Coverage rationale

Configuration and documentary evidence were reviewed from 1 to 5 September 2026. Each observation is linked to E-01, E-02 or E-03. Areas without evidence remain Not tested; a successful sampled check does not imply coverage of the entire environment.

Reference: engagement baseline DEMO-2026-09, revision 1.0, approved for this engagement.

6. Platform layers

Assessed environment

mc-stage-01 / orders and billing namespaces. Managed staging cluster / eight sampled workloads. Service ownership: Lina Bernard, platform lead. The evidence bundle is frozen at assessment close; later changes are outside this issue.

Evidence requirements

Record cluster, namespace, workload, configuration source and observation timestamp. Identify inherited policy and whether evidence concerns desired or running configuration.

Evidence register

E-01: E-01: eight manifest records and two namespace-policy records, 3 September 2026; five workloads have no effective limits.

E-02: E-02: three workload security-context records, build release 2026.09.2, reviewed 4 September 2026.

E-03: E-03: service-account inventory cross-checked against application dependency records, 5 September 2026.

7. Findings register

Findings use the report order. Keep the same reference in the action plan and retest record.

F-1 Workloads have no effective resource limits

FINDING F-1	Low / 3.3 / 10

F-2 Application workloads run as the root user

FINDING F-2	Medium / Not applicable

F-3 Unused service-account tokens are mounted into workloads

FINDING F-3	Low / Not applicable

8. Workloads have no effective resource limits

FINDING F-1	Low / 3.3 / 10

Observation and affected context

Five of eight sampled workloads have no CPU or memory limits. The namespace records do not establish equivalent defaults.

Technical analysis

Requests inform scheduling while limits bound container resource consumption; namespace policy must be checked for effective defaults. A ResourceQuota constrains aggregate allocation and does not by itself supply every container with an appropriate budget. Desired manifests and admitted Pod specifications may differ after policy mutation.

Affected scope and limits

Five of eight workload records lack effective CPU and memory limits, and the two namespace records provide no equivalent defaults. No stress test, node exhaustion or outage was performed. Limit values cannot be chosen responsibly from this documentary sample alone.

Impact

Unexpected resource consumption could impair neighboring services and reduce the effectiveness of capacity planning.

- Availability: unbounded growth in a workload can increase contention for neighboring orders or billing services on shared infrastructure.
- Operations: capacity estimates and troubleshooting become less reliable without an agreed resource envelope. CPU throttling and memory-related termination are also possible if remediation introduces unsuitable limits, so rollout must be measured.

Correction

Agree workload budgets with owners, define requests and limits, and verify normal and peak operation before rollout.

1. Measure representative baseline and peak usage with owners; set per-container requests and justified CPU and memory limits, including sidecars and initialization requirements.
2. Use LimitRange and ResourceQuota where appropriate to enforce namespace policy without treating their defaults as application-specific sizing.
3. Deploy through the existing release process, monitor latency, restarts and saturation, and retain the prior configuration for rollback. Review autoscaling assumptions when requests change.

Accountable owner: Lina Bernard, platform lead. Preserve release and rollback evidence before requesting closure.

9. Supporting evidence

F-1 / Workloads have no effective resource limits

evidence record

E-01: eight manifest records and two namespace-policy records, 3 September 2026; five workloads have no effective limits.

E-01 / KUBERNETES SECURITY

Reviewed item	Scope	Recorded state
Workload sample	8 workloads	5 without effective limits
Namespace sample	orders / billing	No equivalent default limits

Evidence register / Review window: 01-05 September 2026

E-01 - Reviewed record extract, September 2026.

Acceptance criterion

All eight workloads have documented resource budgets and pass a representative non-production workload test.

Closure evidence to collect

1. Compare all eight desired workload records with the effective admitted Pod settings and approved budgets.
2. Run a representative, bounded staging workload and retain resource, latency and restart measurements.
3. Confirm the two namespaces reject or default nonconforming configuration according to the agreed policy; document exceptions with an owner and expiry.

Retest status: pending. These are planned validation requirements, not successful test results. Record the corrected version, review date, owner and residual exceptions before closure.

CVSS v3.1: 3.3 (Low)

CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:L

The vector rates the affected component and prerequisites described in this finding. Business priority additionally considers exposure, data sensitivity and delivery dependencies.

Workload budget - owner validation required

```
resources:
  requests: {cpu: "250m", memory: "256Mi"}
  limits: {cpu: "1000m", memory: "512Mi"}
```

References

Engagement baseline DEMO-2026-09, revision 1.0; evidence E-01.

Kubernetes - resource management

10. Application workloads run as the root user

FINDING F-2	Medium / Not applicable

Observation and affected context

Three application workload records permit root execution and do not declare a non-root requirement. The application baseline requires an unprivileged runtime identity.

Technical analysis

The reviewed runtime identity permits UID 0 and lacks an explicit non-root requirement. Image metadata and the Pod or container securityContext both influence the effective identity. A declared image USER alone is not equivalent to an admission-enforced runtime restriction.

Affected scope and limits

Three application records are affected. Root inside a container is not automatically host root; host mounts, capabilities, user namespaces and runtime isolation affect the boundary. This finding establishes unnecessary in-container privilege, not a container escape or access to the managed control plane.

Impact

An application compromise would have more privilege within the container than required. No container escape is claimed.

- Integrity: application compromise could give more control over writable container files and processes than the service needs.
- Defense in depth: unnecessary privilege reduces the value of runtime isolation controls. The practical reach depends on mounts and other restrictions; cross-namespace access and host compromise were not established.

Correction

Use a tested non-root runtime identity, review required file permissions and enforce the approved workload policy.

1. Select a supported non-zero runtime identity and set runAsNonRoot; define runAsUser and group settings where deterministic ownership is required.
2. Review writable paths, volume ownership, initialization tasks and application startup behavior. Avoid fixing permissions by reintroducing unrestricted root execution.
3. Validate complementary restrictions such as allowPrivilegeEscalation: false and reduced capabilities, then enforce an appropriate Pod Security policy with a controlled migration and exception process.

Accountable owner: Lina Bernard, platform lead. Preserve release and rollback evidence before requesting closure.

11. Supporting evidence

F-2 / Application workloads run as the root user

evidence record

E-02: three workload security-context records, build release 2026.09.2, reviewed 4 September 2026.

E-02 / KUBERNETES SECURITY

Reviewed item	Scope	Recorded state
Affected sample	3 application workloads	Root execution permitted
Required state	Runtime identity baseline	Explicit non-root requirement

Evidence register / Review window: 01-05 September 2026

E-02 - Reviewed record extract, September 2026.

Acceptance criterion

The three applications start, serve normal requests and pass maintenance tests under the approved non-root identity.

Closure evidence to collect

1. Confirm the effective identity for each of the three corrected workloads, not only its image metadata.
2. Exercise normal startup, file writes, scheduled maintenance and health checks under the intended identity.
3. Retain admitted configuration and regression results; verify policy enforcement prevents unintended root execution without disrupting approved workloads.

Retest status: pending. These are planned validation requirements, not successful test results. Record the corrected version, review date, owner and residual exceptions before closure.

CVSS: not applicable. This is a control-assurance or hardening observation without an independently characterized exploit path. Its contextual risk rating remains applicable.

Runtime policy

```
securityContext:
  runAsNonRoot: true
  allowPrivilegeEscalation: false
  capabilities:
    drop: ["ALL"]
```

References

Engagement baseline DEMO-2026-09, revision 1.0; evidence E-02.

Kubernetes - Pod Security Standards

12. Unused service-account tokens are mounted into workloads

FINDING F-3	Low / Not applicable

Observation and affected context

Four workloads that do not use the cluster API inherit automatic service-account token mounting.

Technical analysis

Automatic credential projection makes a service-account token available to the workload even where its documented dependencies do not include the Kubernetes API. Pod-level automountServiceAccountToken settings take precedence over the service-account setting; the effective Pod configuration therefore matters.

Affected scope and limits

Four workloads have no documented cluster-API dependency. Token presence does not establish useful administrative permissions: RBAC scope, audience, validity and API connectivity still constrain use. The sample does not demonstrate credential misuse or cluster-wide access.

Impact

Unnecessary credentials are made available to application processes, increasing avoidable exposure if an application is compromised.

- Credential exposure: an application process receives an identity credential it does not need for its documented function.
- Least privilege: removing unused credentials reduces avoidable authentication material in the runtime. The Low rating assumes no elevated effective RBAC grant has been established; new permission evidence would require reassessment.

Correction

Disable token mounting for workloads without an API requirement and document justified exceptions with narrowly scoped permissions.

1. Set automountServiceAccountToken: false for workloads without an API dependency and verify the effective Pod-level setting after rollout.
2. For justified exceptions, use a dedicated service account with narrowly scoped permissions and an explicitly documented credential requirement.
3. Review sidecars and operational agents before disabling mounts. Keep exception ownership, expected audience and renewal behavior in the platform inventory.

Accountable owner: Lina Bernard, platform lead. Preserve release and rollback evidence before requesting closure.

13. Supporting evidence

F-3 / Unused service-account tokens are mounted into workloads

evidence record

E-03: service-account inventory cross-checked against application dependency records, 5 September 2026.

E-03 / KUBERNETES SECURITY

Reviewed item	Scope	Recorded state
Affected sample	4 workloads	Automatic token mounting inherited
Dependency review	Application inventory	No cluster-API requirement documented

Evidence register / Review window: 01-05 September 2026

E-03 - Reviewed record extract, September 2026.

Acceptance criterion

The four workloads operate without a mounted token; exceptions have an approved API dependency and permission review.

Closure evidence to collect

1. Inspect the four corrected Pod specifications and volume metadata to confirm the automatic token projection is absent; do not collect token contents.
2. Exercise normal application and operational-agent behavior to confirm no hidden API dependency is broken.
3. For exceptions, retain the permission review and approved dependency record, and re-evaluate them when the workload changes.

Retest status: pending. These are planned validation requirements, not successful test results. Record the corrected version, review date, owner and residual exceptions before closure.

CVSS: not applicable. This is a control-assurance or hardening observation without an independently characterized exploit path. Its contextual risk rating remains applicable.

Service-account policy

```
spec:
  automountServiceAccountToken: false
```

References

Engagement baseline DEMO-2026-09, revision 1.0; evidence E-03.

Kubernetes - service accounts for Pods

14. Remediation and retest

Finding	Owner / target	Status	Acceptance evidence
F-1 to F-3	Lina Bernard, platform lead / 30 Sep-16 Oct 2026	Open; retest pending	Finding-specific acceptance checks in the action plan

Action plan

F-1: Lina Bernard, platform lead. Target 30 September 2026. All eight workloads have documented resource budgets and pass a representative non-production workload test.

F-2: Lina Bernard, platform lead. Target 16 October 2026. The three applications start, serve normal requests and pass maintenance tests under the approved non-root identity.

F-3: Lina Bernard, platform lead. Target 16 October 2026. The four workloads operate without a mounted token; exceptions have an approved API dependency and permission review.

Retest record

Not retested as of 16 September 2026. Proposed retest window: 19-20 October 2026, subject to delivery evidence. All three findings remain Open; no correction or risk acceptance is asserted.

15. Rating and limitations

Rating method

Technical vulnerabilities use CVSS v3.1. Each scored finding includes its vector and prerequisites. Business priority also considers exposure and service dependencies. Control-assurance and governance observations are rated contextually and marked Not applicable for CVSS. CVSS is owned by FIRST and used by permission.

Residual uncertainty

Conclusions apply to the recorded configuration versions and assessment window. Untested areas are listed in the coverage register. Changes after evidence collection require revalidation. All findings remain open until the documented acceptance evidence has been reviewed.

Technical severity and remediation priority may differ. Document the reason when exposure, business context or correction dependencies change the order of work.