UAE Information Assurance Standard, version 2.1 (November 2025): technical evidence

Regime	UAE Information Assurance Standard, version 2.1 (November 2025), UAE Cyber Security Council, United Arab Emirates
Organisation	Example Bank
Cluster	example-production
Collected	2026-11-07T14:39:23Z
Collector	Bahriya Tersane Compliance and Assurance Convoy, collector 1.x.y

This report records technical evidence from one Kubernetes cluster. It maps the requirements of the regime named above to what the cluster shows, through the controls of NIST SP 800-53 Rev. 5. It does not state that the organisation complies with the regime.

The mapping of each control to NIST SP 800-53 Rev. 5 is the standard's own, from its Annex D, used unchanged.

Organisational requirements are the organisation's to evidence, and the report marks them so. Where a requirement is only partly technical, the cluster evidences its part and the organisation the rest. Whether the organisation complies is determined by its assessor.

Requirement identifiers are the regulator's; each summary is Mamluk's own paraphrase, not the regulator's text. The files of this run are listed with their SHA-256 digests in manifest.sha256.

This report is the evidence. The ClusterComplianceReport tersane-compliance-nist that the cluster also holds is an indicative summary of trivy-operator's checks by NIST control, in which a check that has not run is shown as passed; it is not evidence for any regime.

Exempt namespaces: kube-system, kube-node-lease, kube-public, tigera-operator, calico-system, calico-apiserver, longhorn-system, cert-manager, kong, kuma-system, logging, bahriya-system, kyverno, trivy-system, local-path-storage, compliance, compliance-node. No policy this chart installs applies in these namespaces.

Summary

134 requirements: 4 enforced by a policy, 9 checked, 10 provided by the platform, 111 organisational.

Outcome	Requirements	Meaning
Fail	13	A check behind the requirement failed.
Not provided	1	A control the requirement needs is turned off.
Incomplete	2	The checks that could be judged passed; others could not be judged.
Not evaluated	3	No check behind the requirement could be judged.
Pass	1	Every check behind the requirement passed.
Recorded	3	Evidence is recorded for review; no verdict is reached.
Customer evidence	111	An organisational requirement: the customer evidences it.

8 object(s) fail at least one check: 4 cluster-wide and 4 in 2 namespace(s). They are listed under Fix first, those that fail an always applicable requirement first, then the most severe first.

Fix first

Each object that fails a check, with the requirements it fails, what is wrong, how to fix it and the command that shows the full result. Objects are grouped by namespace; objects that belong to no namespace are cluster-wide. A group's place is set by its most severe finding, and within a group the most severe come first. Fixing an object clears its failure in every requirement listed against it. Failures in exempt namespaces are not listed here; they are listed, not counted, under each check.

Cluster-wide

4 object(s), most severe Critical.

ClusterRole **tigera-operator** Fails T5.2.3, T5.3.2, T5.3.4 Always applicable: T5.2.3	
CRITICAL	AVD-KSV-0041: Manage secrets. ClusterRole 'tigera-operator' shouldn't have access to manage resource 'secrets' Check trivy.rbac-privilege, trivy-operator AVD-KSV-0041. **How to fix** Manage secrets are not allowed. Remove resource 'secrets' from cluster role (trivy-operator)
CRITICAL	AVD-KSV-0114: Manage webhookconfigurations. ClusterRole 'tigera-operator' should not have access to resources ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"] for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"] Check trivy.rbac-privilege, trivy-operator AVD-KSV-0114. **How to fix** Remove webhook configuration resources/verbs, acceptable values for verbs ['get', 'list', 'watch'] (trivy-operator)
Full result	`kubectl get clusterrbacassessmentreport clusterrole-tigera-operator -o yaml`

ClusterRole **tigera-operator-secrets** Fails T5.2.3, T5.3.2, T5.3.4 Always applicable: T5.2.3	
CRITICAL	AVD-KSV-0041: Manage secrets. ClusterRole 'tigera-operator-secrets' shouldn't have access to manage resource 'secrets' Check trivy.rbac-privilege, trivy-operator AVD-KSV-0041. **How to fix** Manage secrets are not allowed. Remove resource 'secrets' from cluster role (trivy-operator)
Full result	`kubectl get clusterrbacassessmentreport clusterrole-tigera-operator-secrets -o yaml`

Node **example-production-control-plane** Fails T3.4.1, T3.4.2 Always applicable: T3.4.1, T3.4.2

MEDIUM	no cipher suites are set, so the defaults, which include CBC suites, are offered Check node.kubelet-ciphers. **How to fix** On a Bahriya Kubernetes Engine cluster, set compliance.controlPlane.enabled to true in config.yaml (the default) and upgrade each node with upgrade.sh, which writes tlsMinVersion and tlsCipherSuites into every kubelet's configuration. Elsewhere, set tlsMinVersion to VersionTLS12 and tlsCipherSuites to AEAD suites only (GCM or ChaCha20-Poly1305) in the kubelet's configuration, then restart the kubelet. (Bahriya Tersane)
MEDIUM	the minimum TLS version is not set, which allows TLS 1.0 Check node.kubelet-ciphers. **How to fix** On a Bahriya Kubernetes Engine cluster, set compliance.controlPlane.enabled to true in config.yaml (the default) and upgrade each node with upgrade.sh, which writes tlsMinVersion and tlsCipherSuites into every kubelet's configuration. Elsewhere, set tlsMinVersion to VersionTLS12 and tlsCipherSuites to AEAD suites only (GCM or ChaCha20-Poly1305) in the kubelet's configuration, then restart the kubelet. (Bahriya Tersane)
	What was read on the node is in this run's snapshot.json, under nodes.

Platform **API server audit log** Fails T3.2.2 Always applicable: T3.2.2

MEDIUM	kept 30 days before rotation; the regime requires 365 Check platform.audit-rotation. **How to fix** Keep the audit log for at least the period the regime requires. Either forward it to a central log store that keeps it that long, which BKE's Fluent Bit can do, and give that store to your assessor as evidence; or raise compliance.audit.maxAgeDays in /etc/bke/config.yaml, with maxBackups and maxSizeMB large enough to hold that period, and run upgrade.sh at the version the cluster already runs on each control-plane node, then apply.sh --commit. (Bahriya Tersane)
Full result	`kubectl -n bahriya-system get configmap bke-platform -o jsonpath='{.data.audit\.rotation}'`

Namespace payments

3 object(s), most severe High.

Deployment **legacy-batch** Fails T3.1.1, T6.2.3 Always applicable: T3.1.1, T6.2.3

HIGH	images not signed by example-bank: registry.example.com/payments/legacy-batch:1.x.y Check policy.images-signed. **How to fix** Sign the image with cosign and add the public key, with a pattern matching the image, under signatures.keys in the Convoy's values. If the image is already signed, add the key that signed it. Then upgrade the Convoy. (Bahriya Tersane)
MEDIUM	AVD-KSV-0012: Runs as root user. Container 'app' of ReplicaSet 'legacy-batch-779458f596' should set 'securityContext.runAsNonRoot' to true Check trivy.run-as-root, trivy-operator AVD-KSV-0012. **How to fix** Set 'containers[].securityContext.runAsNonRoot' to true. (trivy-operator)
MEDIUM	AVD-KSV-0104: Seccomp policies disabled. container "app" of replicaset "legacy-batch-779458f596" in "payments" namespace should specify a seccomp profile Check trivy.seccomp, trivy-operator AVD-KSV-0104. **How to fix** Specify seccomp either by annotation or by seccomp profile type having allowed values as per pod security standards (trivy-operator)
LOW	containers without CPU and memory limits: app Check policy.resource-limits. Also failed by trivy-operator AVD-KSV-0011 (Low) and AVD-KSV-0018 (Low). **How to fix** Set resources.limits.cpu and resources.limits.memory on every container of the Pod template, at values the workload has been measured to need. (Bahriya Tersane)
LOW	AVD-KSV-0030: Runtime/Default Seccomp profile not set. Either Pod or Container should set 'securityContext.seccompProfile.type' to 'RuntimeDefault' Check trivy.seccomp, trivy-operator AVD-KSV-0030. **How to fix** Set spec.securityContext.seccompProfile.type to RuntimeDefault on the Pod, and remove any container's seccompProfile that sets another type. (Bahriya Tersane)

Full results	`kubectl get configauditreport replicaset-legacy-batch-779458f596 -n payments -o yaml`
	`kubectl get policyreport 37ec704c-84bb-42c9-ae23-78958efceed3 -n payments -o yaml`
	Kyverno names each policy report after the UID of the object it covers.

Pod **legacy-batch-779458f596-fwgcd** Fails T6.2.3 Always applicable: T6.2.3

HIGH	images not signed by example-bank: registry.example.com/payments/legacy-batch:1.x.y Check policy.images-signed. **How to fix** Sign the image with cosign and add the public key, with a pattern matching the image, under signatures.keys in the Convoy's values. If the image is already signed, add the key that signed it. Then upgrade the Convoy. (Bahriya Tersane)
Full result	`kubectl get policyreport 715816ab-6d7f-462a-83f5-88a2c3c1acfb -n payments -o yaml` Kyverno names each policy report after the UID of the object it covers.

ReplicaSet **legacy-batch-779458f596** Fails T6.2.3 Always applicable: T6.2.3

HIGH	images not signed by example-bank: registry.example.com/payments/legacy-batch:1.x.y Check policy.images-signed. **How to fix** Sign the image with cosign and add the public key, with a pattern matching the image, under signatures.keys in the Convoy's values. If the image is already signed, add the key that signed it. Then upgrade the Convoy. (Bahriya Tersane)
Full result	`kubectl get policyreport 60ee71ad-b7f6-490e-8185-a36093c453d0 -n payments -o yaml` Kyverno names each policy report after the UID of the object it covers.

Namespace default

1 object(s), most severe Medium.

Namespace **default** Fails T1.3.1, T1.3.2, T3.1.4, T4.1.1, T4.1.3 Always applicable: T1.3.1, T1.3.2, T4.1.1	
MEDIUM	namespace default holds no NetworkPolicy Check policy.namespace-network-policy. **How to fix** Add NetworkPolicies to the namespace that allow only the traffic its workloads need. To start every new namespace closed, set generate.defaultDenyNetworkPolicy.enabled to true in the Convoy's values. (Bahriya Tersane)
LOW	the namespace needs the label tersane.bahriya.cloud/data-classification, one of: top-secret, secret, restricted, public Check policy.namespace-classification. **How to fix** Label the namespace tersane.bahriya.cloud/data-classification with one of the values in namespaces.classifications in the Convoy's values. (Bahriya Tersane)
LOW	the namespace needs the label tersane.bahriya.cloud/environment, one of: production, staging, test, development Check policy.namespace-environment. **How to fix** Label the namespace tersane.bahriya.cloud/environment with one of the values in namespaces.environments in the Convoy's values. (Bahriya Tersane)
Full result	`kubectl get clusterpolicyreport 01c02383-ae51-479c-a362-355224387a37 -o yaml` Kyverno names each policy report after the UID of the object it covers.

Requirements

M1 Strategy and Planning

M1.1 Entity Context and Leadership

M1.1.1	The internal and external context of the security programme identified: interested parties and their requirements, the sector and national context, and the entity's capabilities. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-1, PM-11.	**Customer evidence**
M1.1.2	Top management leads information security: it sets policy and objectives in line with strategy, provides resources and builds security into the entity's processes. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-2, PM-3, PM-29.	**Customer evidence**
M1.1.3	Security roles and authorities assigned and communicated, with an accountable senior leader, a security function and an Information Security Committee. Organisational. Always applicable, priority P1. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1, PM-2, PM-6, PM-29.	**Customer evidence**

M1.2 Information Security Policy

M1.2.1	An information security policy approved by top management, committed to legal compliance and improvement, communicated, and reviewed at planned intervals. Organisational. Always applicable, priority P1. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1.	**Customer evidence**

M1.2.2	Security objectives, measurable where practicable, consistent with the policy and the risk assessment, documented, approved and kept current. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-1, PM-3, PM-4, PM-6, PM-9, PM-14, PM-28, PM-30, PM-31.	**Customer evidence**
M1.2.3	Topic-specific security policies approved, published, acknowledged by staff and reviewed at planned intervals. Organisational. Always applicable, priority P2. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1.	**Customer evidence**

M1.3 Support

M1.3.1	Adequate people and budget for the security programme, with competences defined and evidenced. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-3, PM-13.	**Customer evidence**
M1.3.2	A communication plan for security matters, including contact with the UAE cyber security authorities. Organisational. Always applicable, priority P2. NIST SP 800-53: PM-1, PM-15, PM-28, PM-31.	**Customer evidence**
M1.3.3	Security documentation approved, version-controlled, distributed and disposed of according to its classification, and compliance with the mandatory controls documented. Organisational. Always applicable, priority P2. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1, CP-2, IR-8, PL-2, PM-4, PM-9, PM-28, PM-30, PM-31, SA-5.	**Customer evidence**
M1.3.4	Security built into project management, with risks identified early and treated throughout the project. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: SA-3, SA-9, SA-15.	**Customer evidence**
M1.3.5	Contact, as far as possible, with relevant security interest groups and forums, with rules on what may be shared. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: SI-5, PM-15.	**Customer evidence**

M2 Information Security Risk Management

M2.1 Information Security Risk Management Policy

M2.1.1	An approved risk management policy setting scope, roles, risk acceptance and the criteria for assessing risk. Organisational. Always applicable, priority P1. NIST SP 800-53: RA-1, PM-9.	**Customer evidence**

M2.2 Information Security Risk Assessment

M2.2.1	Risks to the confidentiality, integrity and availability of information identified, with an owner for each. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-9, PM-28, RA-3.	**Customer evidence**
M2.2.2	Identified risks analysed for consequence and likelihood, and their levels determined. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-9, PM-28, RA-3.	**Customer evidence**

M2.2.3	Analysed risks evaluated against the criteria and prioritised for treatment, with the results documented and shared with the UAE lead entities for cyber security where required. Organisational. Always applicable, priority P1. NIST SP 800-53: PM-9, PM-28, RA-3.	**Customer evidence**

M2.3 Information Security Risk Treatment

M2.3.1	A treatment option chosen for each risk, with retained risks approved by their owners. Organisational. Always applicable, priority P1. NIST SP 800-53: RA-7.	**Customer evidence**
M2.3.2	The controls needed for each risk treatment identified, starting from this standard's controls. Organisational. Always applicable, priority P1. NIST SP 800-53: RA-7.	**Customer evidence**
M2.3.3	A risk treatment plan with actions, resources, responsibilities and target dates. Organisational. Always applicable, priority P1. NIST SP 800-53: RA-7.	**Customer evidence**
M2.3.4	A Statement of Applicability listing the controls, the reason for each, their status and the justification for any exclusion. Organisational. Always applicable, priority P1. NIST SP 800-53: none mapped by the regime.	**Customer evidence**

M2.4 Ongoing Information Security Risk Management

M2.4.1	The risk assessment and treatment reviewed at a defined frequency and on major change. Organisational. Always applicable, priority P1. NIST SP 800-53: none mapped by the regime.	**Customer evidence**
M2.4.2	Risk information shared with stakeholders and decision-makers throughout the risk management process. Organisational. Always applicable, priority P1. NIST SP 800-53: none mapped by the regime.	**Customer evidence**

M3 Awareness and Training

M3.1 Awareness and Training Policy

M3.1.1	An approved awareness and training policy. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AT-1.	**Customer evidence**

M3.2 Awareness and Training Planning

M3.2.1	An approved awareness and training programme, with its effectiveness evaluated and records kept. Organisational. Always applicable, priority P2. NIST SP 800-53: AT-2.	**Customer evidence**

M3.3 Security Training

M3.3.1	Security skills needed identified and compared with those held, to find training gaps. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AT-3, CP-3, IR-2, PM-13.	**Customer evidence**
M3.3.2	A training plan with a solution and timeline for each training need. Organisational. Always applicable, priority P2. NIST SP 800-53: AT-3, CP-3, IR-2, PM-13.	**Customer evidence**
M3.3.3	Training delivered according to the plan. Organisational. Always applicable, priority P2. NIST SP 800-53: AT-3, CP-3, IR-2, PM-13.	**Customer evidence**

M3.3.4	The effectiveness of training measured, and ineffective training improved or replaced. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: AT-3, AT-6.	**Customer evidence**
M3.3.5	Training records kept for everyone who needs security training, and reviewed periodically. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: AT-4.	**Customer evidence**

M3.4 Security Awareness

M3.4.1	Security awareness campaigns for staff and third parties, planned, delivered and recorded. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AT-2.	**Customer evidence**

M4 Human Resource Security

M4.1 Prior to Employment

M4.1.1	Background checks proportionate to role and risk on candidates, contractors and third-party staff. Organisational. Always applicable, priority P2. NIST SP 800-53: PS-3, SA-21.	**Customer evidence**
M4.1.2	Security terms and conditions in employment and third-party contracts, signed by each person. Organisational. Always applicable, priority P2. NIST SP 800-53: PL-4, PS-6.	**Customer evidence**

M4.2 During Employment

M4.2.1	Staff, contractors and third parties informed of the security policies that apply to them, and accepting them at first access. Organisational. Always applicable, priority P2. NIST SP 800-53: PL-4, PS-6, PS-7, SA-9.	**Customer evidence**
M4.2.2	A formal disciplinary process for breaches of security policy. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: PS-8.	**Customer evidence**

M4.3 Termination or Change of Employment

M4.3.1	Security duties that continue after termination or change of employment defined and communicated. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: PS-4, PS-5.	**Customer evidence**
M4.3.2	Assets returned on termination or change of employment. Organisational. Always applicable, priority P2. NIST SP 800-53: PS-4, PS-5.	**Customer evidence**

M5 Compliance

M5.1 Compliance with Information Security Legal Requirements

M5.1.1	Legal, regulatory and contractual security requirements identified, documented and reviewed periodically. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1.	**Customer evidence**
M5.1.2	Intellectual property and software licensing requirements identified and controlled. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: CM-10.	**Customer evidence**

M5.1.3	Records with legal or contractual protection requirements identified and protected against loss, destruction, falsification and unauthorised access or release. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-3, AC-23, AU-9, AU-10, CP-9, SC-8, SC-8(1), SC-13, SC-28, SC-28(1).	**Customer evidence**
M5.1.4	Compliance with data protection and privacy law shown, through a privacy and protection policy, privacy risk assessments of processes handling personal data, controls for it and a process to detect, respond to and report breaches. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: none mapped by the regime.	**Customer evidence**

M5.2 Compliance with Information Policies and Standards

M5.2.1	Compliance with security policies and standards checked routinely, and non-compliance reported for remediation. Organisational. Always applicable, priority P2. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1, CA-2, CA-2(1), SA-11(3).	**Customer evidence**

M5.3 Information Systems Audit Considerations

M5.3.1	Audit access to operational systems approved and controlled to avoid disruption. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: AU-9.	**Customer evidence**

M6 Performance Evaluation and Improvement

M6.1 Performance Evaluation Policy

M6.1.1	A documented performance evaluation policy. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: none mapped by the regime.	**Customer evidence**

M6.2 Performance Evaluation

M6.2.1	Security performance and effectiveness monitored and evaluated, with methods, timing and responsibilities defined. Organisational. Always applicable, priority P2. NIST SP 800-53: CA-1, CA-7, PM-6, PM-31.	**Customer evidence**
M6.2.2	Impartial internal audits against a documented audit programme, reported to management. Organisational. Always applicable, priority P2. NIST SP 800-53: CA-1, CA-2, CA-5, CA-7, PM-4.	**Customer evidence**
M6.2.3	Top management reviews the security programme at planned intervals. Organisational. Always applicable, priority P2. NIST SP 800-53: CA-6, PM-1, PM-4, PM-9, PM-10, PM-29.	**Customer evidence**

M6.3 Improvement

M6.3.1	Nonconformities corrected, their causes analysed and corrective actions reviewed. Organisational. Always applicable, priority P2. NIST SP 800-53: CA-5, PL-2, PM-4, PM-31, RA-7.	**Customer evidence**

M6.3.2	Security controls continually improved.	**Customer evidence**
	Organisational. Always applicable, priority P2. NIST SP 800-53: PM-1, PM-9, PM-30, PM-31.	

T1 Information Asset Management

T1.1 Information Asset Management Policy

T1.1.1	An approved asset management policy, assigning owners and governing personal devices.	**Customer evidence**
	Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: none mapped by the regime.	

T1.2 Responsibility for Assets

T1.2.1	An accurate, current inventory of information assets, with an owner for each.	**Recorded**
	Checked, coverage partial. Always applicable, priority P1. NIST SP 800-53: CM-8, PM-5.	
	Scope of this evidence The inventory of the cluster's namespaces, workloads and the images they run is recorded for review. The inventory of other assets, and their owners, are the entity's to evidence.	
	cluster.workload-inventory — **Recorded** — 13 namespaces, 17 workloads, 15 distinct images.	
T1.2.2	Acceptable use rules for information and assets, accepted by staff and third parties.	**Customer evidence**
	Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-20, PL-4.	

T1.3 Information Protection

T1.3.1	A classification scheme for information, implemented across the entity and kept current.	**Fail**
	Enforced, coverage partial. Always applicable, priority P1. NIST SP 800-53: RA-2.	
	Scope of this evidence The evidence is that each namespace outside the exempt namespaces carries a classification label. The scheme, its communication and the classification of other information are the entity's to evidence.	
	Fails in Namespace default.	
	policy.namespace-classification — **Fail** — 1 object(s) fail the policy.	
T1.3.2	Information and related assets labelled according to their classification.	**Fail**
	Enforced, coverage partial. Always applicable, priority P2. NIST SP 800-53: MP-3, PE-22.	
	Scope of this evidence The evidence is that each namespace outside the exempt namespaces carries a classification label. Labelling other information and assets is the entity's to evidence.	
	Fails in Namespace default.	
	policy.namespace-classification — **Fail** — 1 object(s) fail the policy.	

T1.3.3	Information processed, stored and communicated according to its classification.	**Not evaluated**

Platform, coverage partial. Always applicable, priority P2. NIST SP 800-53: MP-2, MP-4, MP-5, MP-6, MP-7, PE-16, PE-18, PE-20, SC-8, SC-28.

Scope of this evidence The evidence is TLS on traffic into the cluster and to its control plane, and the encryption of Kubernetes Secrets at rest. Handling procedures, and the handling of other information, are the entity's to evidence.

platform.control-plane-tls	**Not evaluated**	The control-plane TLS and profiling settings: the platform could not observe it.
platform.secret-encryption	**Not evaluated**	The platform could not observe Secret encryption.
policy.ingress-tls	**Not evaluated**	The policy has judged no object yet: Kyverno has reported no result for it.

T1.3.4	Rules for transferring information inside and outside the entity, by classification, and only authorised information released on social media and other public platforms.	**Not evaluated**

Platform, coverage partial. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-4, AC-17, AC-18, AC-19, AC-20, CA-3, PE-17, SC-7, SC-8, SC-15, PS-6, SA-9. Evidence counted: platform.control-plane-tls, policy.ingress-tls; the rest of the regime's mapping is not evidenced by the cluster.

Scope of this evidence The evidence is the encryption in transit of traffic into the cluster and to its control plane. The rules, agreements and other channels of transfer are the entity's to evidence.

platform.control-plane-tls	**Not evaluated**	The control-plane TLS and profiling settings: the platform could not observe it.
policy.ingress-tls	**Not evaluated**	The policy has judged no object yet: Kyverno has reported no result for it.

T1.3.5	Data leakage prevention that detects and blocks the disclosure of classified information.	**Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: PE-19.

T1.3.6	Information deleted securely when no longer needed, by classification and law, and the deletion process reviewed periodically.	**Customer evidence**

Organisational. Always applicable, priority P2. NIST SP 800-53: none mapped by the regime.

T1.4 Storage Media

T1.4.1	Removable media managed through their life cycle, restricted and monitored.	**Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: MP-2, MP-4, MP-5, MP-6, MP-7.

T1.4.2	Media holding classified information re-used or disposed of securely, with disposals logged.	**Customer evidence**

Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: MP-6.

T1.5 User Endpoint Devices

T1.5.1	User endpoint devices secured, with automatic lock, management of personal devices and logging.	**Customer evidence**

Organisational. Always applicable, priority P1. NIST SP 800-53: AC-17, AC-18, AC-19, AC-3, PE-17, AC-11.

T2 Physical and Environmental Security

T2.1 Physical and Environmental Security Policy

T2.1.1	An approved physical and environmental security policy. Organisational. Always applicable, priority P2. NIST SP 800-53: PE-1.	**Customer evidence**

T2.2 Secure Areas

T2.2.1	Secure areas classified by what they hold, each with a defined perimeter and protection. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: PE-3.	**Customer evidence**
T2.2.2	Entry to secure areas authenticated and logged, with visible identification, visitors and contractors escorted, and delivery areas controlled. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: PE-2, PE-3, PE-4, PE-5, PE-16.	**Customer evidence**
T2.2.3	The public kept away from key facilities, which carry no sign of what they hold. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: PE-3, PE-5.	**Customer evidence**
T2.2.4	Physical protection against natural disasters, attack and accident, based on site risk assessments. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: CP-6, CP-7, PE-9, PE-13, PE-14, PE-15, PE-18, PE-19, PE-23.	**Customer evidence**
T2.2.5	Rules for working in secure areas, with activity supervised, devices controlled and the areas under surveillance. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-19(4), SC-42.	**Customer evidence**

T2.3 Equipment Security

T2.3.1	Equipment protected against unauthorised access and damage, with environmental conditions monitored. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: PE-9, PE-13, PE-14, PE-15, PE-18, PE-19, PE-23.	**Customer evidence**
T2.3.2	Uninterruptible power and emergency lighting for information systems. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: CP-8, PE-9, PE-10, PE-11, PE-12, PE-14, PE-15.	**Customer evidence**
T2.3.3	Power and telecommunication cables protected and segregated. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: PE-4, PE-9.	**Customer evidence**
T2.3.4	Equipment maintained as its supplier recommends, with maintenance logged and classified data protected. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: MA-2, MA-6.	**Customer evidence**
T2.3.5	Equipment used off the premises protected. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: AC-19, AC-20, MP-5, PE-17, MA-2, PE-16.	**Customer evidence**
T2.3.6	Equipment disposed of or re-used securely, with disposals recorded. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: MP-6.	**Customer evidence**

T2.3.7 A clear desk and clear screen policy. **Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-11, MP-2, MP-4.

T3 Operations Management

T3.1 Operational Procedures and Responsibilities

T3.1.1 Minimum security configurations defined for common technology, implemented, monitored for deviation and reviewed periodically. **Fail**

Checked, coverage partial. Always applicable, priority P1. NIST SP 800-53: CM-1, CM-2, CM-3, CM-6, CM-9.

Scope of this evidence The verdict concerns the configuration of the cluster's nodes, control plane and workloads against a hardening baseline; the platform's versions and the inventory are recorded beside it. Configurations of other technology are the entity's to evidence.

Fails in Deployment payments/legacy-batch.

cluster.workload-inventory	**Recorded**	13 namespaces, 17 workloads, 15 distinct images.
node.etcd-data	**Pass**	etcd's data directory is owned by root and closed to other users: so on all 1 node(s).
node.file-permissions	**Pass**	Kubernetes' configuration, credentials and keys are owned by root and no more open than their limits: so on all 1 node(s).
node.kubelet-hardening	**Pass**	The kubelet keeps its read-only port closed and rotates its certificates: so on all 1 node(s).
platform.control-plane-tls	**Not evaluated**	The control-plane TLS and profiling settings: the platform could not observe it.
platform.mesh-cni	**Not evaluated**	The service mesh CNI plugin: the platform could not observe it.
platform.pod-security	**Not evaluated**	Pod Security admission defaults: the platform could not observe it.
platform.versions	**Recorded**	Bahriya Kubernetes Engine 1.x.y, Kubernetes v1.x.y, observed 2026-11-07T14:34:28Z.
policy.resource-limits	**Fail**	1 object(s) fail the policy.
trivy.limits	**Fail**	1 object(s) fail.
trivy.run-as-root	**Fail**	1 object(s) fail.
trivy.seccomp	**Fail**	1 object(s) fail.

T3.1.2 Operating procedures documented, reviewed periodically and available to those who need them. **Customer evidence**

Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: AC-1, AT-1, AU-1, CA-1, CM-1, CP-1, IA-1, IR-1, MA-1, MP-1, PE-1, PL-1, PM-1, PS-1, PT-1, RA-1, SA-1, SC-1, SI-1, SR-1, SA-5.

T3.1.3 A change management process recording each change, its security impact and approval, with rollback and recovery plans. **Customer evidence**

Organisational. Always applicable, priority P2. NIST SP 800-53: CM-3, CM-5, SA-10.

T3.1.4 Development, test and production environments separated, with rules for access and promotion between them. **Fail**

Enforced, coverage partial. Applicable by risk assessment, priority P2. NIST SP 800-53: CM-4(1), CM-5.

Scope of this evidence The evidence is that each namespace outside the exempt namespaces carries an environment label, so production can be told apart. Separate environments, and the rules for access and promotion, are the entity's to evidence.

Fails in Namespace default.

policy.namespace-environment	**Fail**	1 object(s) fail the policy.

T3.1.5 Capacity measured, future needs estimated and capacity adjusted. **Customer evidence**

Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: AU-4, CP-2(2), SC-5(2).

T3.1.6 Up-to-date protection against malware on servers and endpoints, with scans, logging and alerts. **Customer evidence**

Organisational. Always applicable, priority P1. NIST SP 800-53: AT-2, SI-3.

T3.1.7 Backup scope, frequency and protection set, procedures documented and backups tested. **Customer evidence**

Organisational. Always applicable, priority P1. NIST SP 800-53: CP-9.

T3.1.8 Software installed only by authorised administrators, originals of installed software kept, installations logged and unauthorised software removed. **Customer evidence**

Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: CM-5, CM-7(4), CM-7(5), CM-11.

T3.2 Monitoring

T3.2.1 A monitoring policy setting what is monitored, how and how often, and the requirements for logging and clock synchronisation. **Customer evidence**

Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: AU-1.

T3.2.2 Auditable events and log content defined for all systems, with a minimum review frequency and retention, logs reviewed by trained staff, and critical logs protected. **Fail**

Platform, coverage partial. Always applicable, priority P1. NIST SP 800-53: AU-3, AU-6, AU-11, AU-12, AU-14, AU-9.

Scope of this evidence The evidence is the cluster's API audit log and how long it is kept locally, judged against a default of one year that the entity may change. The protection of logs, logging by other systems and applications, and the review of logs are the entity's to evidence.

Fails in Platform API server audit log.

platform.audit	**Pass**	API audit logging: in effect on every control-plane node.
platform.audit-rotation	**Fail**	The API server keeps its audit log for 30 days, fewer than the 365 the regime requires. A log collected to a central store may meet the requirement; that store is the customer's to evidence.

T3.2.3 Networks, systems and applications monitored for anomalous behaviour, with what is gathered, how often it is reviewed and by whom, and how long it is kept, defined; potential incidents evaluated. **Customer evidence**

Organisational. Always applicable, priority P1. NIST SP 800-53: AU-6, AU-7, SI-4.

T3.2.4 One date and time format across systems, with clock synchronisation checked regularly. **Not provided**

Platform, coverage partial. Applicable by risk assessment, priority P3. NIST SP 800-53: AU-8.

Scope of this evidence The evidence is the time source of the cluster's nodes and their synchronisation. Other systems, and the date and time format, are the entity's to evidence.

node.time-sync	**Not evaluated**	The platform names no time servers, so there is nothing to compare the nodes with.
platform.time-servers	**Not provided**	No time servers are set for the cluster; each node keeps its own.

T3.3 Threat and Vulnerability Management

T3.3.1 Threat information gathered from chosen sources, analysed and acted upon. **Customer evidence**

Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: RA-10.

T3.3.2 Vulnerabilities in new systems found and planned for before production, production systems tested, and vulnerabilities addressed. **Pass**

Checked, coverage partial. Always applicable, priority P1. NIST SP 800-53: RA-3, RA-5, SI-2, SI-5.

Scope of this evidence The verdict concerns the images of the workloads outside the exempt namespaces and the freshness of their scans; the platform's versions and the age of open vulnerabilities are recorded beside it. Testing the nodes, the platform and other systems is the entity's to evidence.

cluster.scan-freshness	**Pass**	Every workload outside the exempt namespaces was scanned within 48 hours.
cluster.vulnerability-age	**Recorded**	No critical or high vulnerability is open outside the exempt namespaces.
platform.versions	**Recorded**	Bahriya Kubernetes Engine 1.x.y, Kubernetes v1.x.y, observed 2026-11-07T14:34:28Z.
trivy.vulnerabilities	**Pass**	No container outside the exempt namespaces runs an image with a critical or high vulnerability.

T3.4 Cryptography and Key Management

T3.4.1 Cryptographic governance, with encryption requirements set from risk and weak or outdated algorithms replaced promptly. **Fail**

Checked, coverage partial. Always applicable, priority P2. NIST SP 800-53: IA-7, SC-12, SC-13, SC-17.

Scope of this evidence The verdict concerns the TLS settings of the cluster's control plane and kubelets; the provider of the key that encrypts Secrets is recorded beside it. The policy, and cryptography elsewhere, are the entity's to evidence.

Fails in Node example-production-control-plane.

node.kubelet-ciphers	**Fail**	The kubelet offers TLS 1.2 or later with AEAD cipher suites only: not so on 1 node(s).
platform.control-plane-tls	**Not evaluated**	The control-plane TLS and profiling settings: the platform could not observe it.
platform.key-provider	**Not evaluated**	The platform could not observe the encryption provider.

T3.4.2 All classified data, credentials included, encrypted at rest and in transit on every device and medium. **Fail**

Platform, coverage partial. Always applicable, priority P1. NIST SP 800-53: SC-13.

Scope of this evidence The verdict concerns the strength of the TLS cipher suites the cluster's control plane and kubelets offer. The encryption at rest of Kubernetes Secrets and TLS on Ingress traffic are reported under T1.3.3. Encryption of other classified data at rest and in transit, and on devices and media, is the entity's to evidence.

Fails in Node example-production-control-plane.

node.kubelet-ciphers	**Fail**	The kubelet offers TLS 1.2 or later with AEAD cipher suites only: not so on 1 node(s).
platform.control-plane-tls	**Not evaluated**	The control-plane TLS and profiling settings: the platform could not observe it.

T3.4.3 Cryptographic keys managed securely through their life cycle, stored securely and recoverable. **Not evaluated**

Platform, coverage partial. Always applicable, priority P1. NIST SP 800-53: SC-12, SC-17.

Scope of this evidence The provider of the key that encrypts the cluster's Secrets is recorded for review. The management of other keys is the entity's to evidence.

platform.key-provider	**Not evaluated**	The platform could not observe the encryption provider.

T3.4.4 An inventory of systems using public key cryptography, with post-quantum standards tested before use and staff prepared for the transition. **Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: none mapped by the regime.

T4 Network Security

T4.1 Network Security Management

T4.1.1 Networks controlled: an approved policy, a diagram, authentication of connecting devices, risk-assessed network components, and filtering and monitoring of traffic in and out. **Fail**

Platform, coverage partial. Always applicable, priority P1. NIST SP 800-53: AC-3, AC-17, AC-18, AC-20, SC-7, SC-8, SC-10. Evidence counted: platform.host-firewall, policy.default-deny-generated, policy.namespace-network-policy, trivy.external-ip; the rest of the regime's mapping is not evidenced by the cluster.

Scope of this evidence The evidence is the cluster's host firewall and its network policies; the verdict requires the Convoy's generated default-deny network policy. The policy, the diagram, authentication of connecting devices, monitoring of traffic, the risk assessment of network components and other networks are the entity's to evidence.

Fails in Namespace default.

platform.host-firewall	**Not evaluated**	The platform could not count the host endpoints.
policy.default-deny-generated	**Not provided**	The policy tersane-compliance-default-deny is not in the cluster: it is set to Off, or the Convoy's policies are not installed.
policy.namespace-network-policy	**Fail**	1 object(s) fail the policy.
trivy.external-ip	**Pass**	Every object outside the exempt namespaces passes.

T4.1.2 Security requirements set and met for each network service, and written into service agreements. **Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: CA-3, SA-9.

T4.1.3 Networks segregated by defined criteria, with segregation built in and evaluated periodically. **Fail**

Platform, coverage partial. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-4, SC-7.

Scope of this evidence The evidence is the cluster's host firewall and its network policies, which separate namespaces; the verdict requires the Convoy's generated default-deny network policy. Segregation of other networks, and its periodic evaluation, are the entity's to evidence.

Fails in Namespace default.

platform.host-firewall	**Not evaluated**	The platform could not count the host endpoints.
policy.default-deny-generated	**Not provided**	The policy tersane-compliance-default-deny is not in the cluster: it is set to Off, or the Convoy's policies are not installed.
policy.namespace-network-policy	**Fail**	1 object(s) fail the policy.
trivy.external-ip	**Pass**	Every object outside the exempt namespaces passes.
trivy.network-policy-selectors	**Pass**	Every object outside the exempt namespaces passes.

T4.1.4 Wireless networks surveyed and controlled, with guest access kept apart. **Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-18.

T4.1.5 Access to illegal, malicious and phishing websites blocked. **Customer evidence**

Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: SC-7.

T5 Identity and Access Management

T5.1 Identity and Access Management

T5.1.1 An identity and access management policy. **Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-1.

T5.2 User Access Management

T5.2.1 A unique identity and account for each user, never shared, identities removed promptly when no longer needed, and activity of restricted identities logged. **Incomplete**

Platform, coverage partial. Always applicable, priority P1. NIST SP 800-53: AC-2.

Scope of this evidence The verdict concerns whether sign-in to the cluster is delegated to the entity's identity provider; the cluster's role bindings are recorded beside it. Whether shared credentials, such as kubeadm's admin.conf, are still in use, the removal of identities and the logging of restricted identities' activity are the entity's to evidence.

cluster.access-review	**Recorded**	109 role bindings to subjects; 2 to cluster-admin across the cluster.
platform.oidc	**Not evaluated**	The platform could not read the authentication configuration.

T5.2.2 Access rights granted and removed through a formal procedure, with approval and implementation by different people, records of all rights, removal at once on change of role or departure, and temporary and inactive accounts checked periodically. **Recorded**

Checked, coverage partial. Always applicable, priority P1. NIST SP 800-53: AC-2, IA-2, IA-4, IA-5, IA-8. Evidence counted: cluster.access-review; the rest of the regime's mapping is not evidenced by the cluster.

Scope of this evidence The cluster's role bindings, the record of who holds which rights, are recorded for review. The procedure, and the removal of rights, are the entity's to evidence.

cluster.access-review	**Recorded**	109 role bindings to subjects; 2 to cluster-admin across the cluster.

T5.2.3 Privileged rights restricted and recorded, administrators using separate accounts with multi-factor authentication, and their activity logged. **Fail**

Checked, coverage partial. Always applicable, priority P1. NIST SP 800-53: AC-2, AC-3, AC-6, CM-5.

Scope of this evidence The verdict concerns which subjects and workloads hold privileged access in the cluster, and that sign-in is delegated to the identity provider; the role bindings are recorded beside it. Whether that provider requires a second factor, and the logging and review of administrators' activity, are the entity's to evidence.

Fails in ClusterRole tigera-operator, ClusterRole tigera-operator-secrets.

cluster.access-review	**Recorded**	109 role bindings to subjects; 2 to cluster-admin across the cluster.
node.kubelet-authz	**Pass**	The kubelet authorises requests through the API server: so on all 1 node(s).
platform.anonymous	**Not evaluated**	Anonymous requests restricted to the health endpoints: the platform could not observe it.
platform.oidc	**Not evaluated**	The platform could not read the authentication configuration.
policy.no-default-token	**Pass**	Every object the policy judged passes it.
trivy.rbac-exec	**Pass**	Every object outside the exempt namespaces passes.
trivy.rbac-privilege	**Fail**	2 object(s) fail.
trivy.token-automount	**Pass**	Every object outside the exempt namespaces passes.

T5.2.4 User access rights reviewed periodically and on any change in a user's status. **Recorded**

Checked, coverage partial. Always applicable, priority P1. NIST SP 800-53: AC-2, PE-2. Evidence counted: cluster.access-review; the rest of the regime's mapping is not evidenced by the cluster.

Scope of this evidence The cluster's role bindings are recorded for the review. The review itself is the entity's to evidence.

cluster.access-review	**Recorded**	109 role bindings to subjects; 2 to cluster-admin across the cluster.

T5.3 Secure Authentication and Authorization

T5.3.1 Strong authentication on every system, secure log-on and log-off, session limits on sensitive systems and idle sessions ended. **Customer evidence**

Organisational. Always applicable, priority P1. NIST SP 800-53: AC-7, AC-8, AC-9, IA-6.

T5.3.2 Access to information and assets restricted by role and business need. **Fail**

Checked, coverage partial. Applicable by risk assessment, priority P2. NIST SP 800-53: AC-3, AC-24.

Scope of this evidence The verdict concerns the cluster's role-based access control: privileged bindings, the kubelet's authorisation and anonymous access. Access to other systems is the entity's to evidence.

Fails in ClusterRole tigera-operator, ClusterRole tigera-operator-secrets.

node.kubelet-authz	**Pass**	The kubelet authorises requests through the API server: so on all 1 node(s).
platform.anonymous	**Not evaluated**	Anonymous requests restricted to the health endpoints: the platform could not observe it.
trivy.rbac-privilege	**Fail**	2 object(s) fail.

T5.3.3 Remote access only over secure channels with authentication, and remote administration approved and monitored. **Incomplete**

Platform, coverage partial. Always applicable, priority P1. NIST SP 800-53: AC-17, AC-18, AC-20, CA-3, IA-2, IA-8.

Scope of this evidence The evidence is that sign-in to the cluster is delegated to the identity provider, anonymous access is refused, kubelets authenticate their callers and the host firewall restricts access. Authentication and encryption of SSH access to the nodes, other remote access to the entity's systems, and the approval and monitoring of remote administration are the entity's to evidence.

node.kubelet-authn	**Pass**	The kubelet refuses anonymous requests and authenticates clients by certificate and webhook: so on all 1 node(s).
platform.anonymous	**Not evaluated**	Anonymous requests restricted to the health endpoints: the platform could not observe it.
platform.host-firewall	**Not evaluated**	The platform could not count the host endpoints.
platform.oidc	**Not evaluated**	The platform could not read the authentication configuration.

T5.3.4 Utility programs that can override controls restricted to authorised people, and their use monitored. **Fail**

Checked, coverage partial. Applicable by risk assessment, priority P4. NIST SP 800-53: AC-3, AC-6. Evidence counted: trivy.rbac-exec, trivy.rbac-privilege; the rest of the regime's mapping is not evidenced by the cluster.

Scope of this evidence The verdict concerns which subjects may run commands inside containers or hold privileged rights in the cluster. Monitoring their use, and other utility programs, are the entity's to evidence.

Fails in ClusterRole tigera-operator, ClusterRole tigera-operator-secrets.

trivy.rbac-exec	**Pass**	Every object outside the exempt namespaces passes.
trivy.rbac-privilege	**Fail**	2 object(s) fail.

T5.4 Authentication Information

T5.4.1 Credentials issued, revoked and re-issued through a secure process. **Customer evidence**

Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: IA-5.

T5.4.2 Good practice for using credentials set and communicated to users. **Customer evidence**

Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: IA-5.

T5.4.3	A credential management system that automates verified credential changes, enforces strength, lifetime and reuse limits, forces a change of default passwords at first login, and stores credentials hashed or encrypted. Organisational. Always applicable, priority P1. NIST SP 800-53: IA-5.	**Customer evidence**

T6 Third Party Security

T6.1 Third Party Security Policy

T6.1.1	An approved third-party security policy. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: SR-1.	**Customer evidence**

T6.2 Management of Third Party and Supply Chain Risks

T6.2.1	Suppliers classified by risk, supply chain risks assessed and mitigated, and monitored. Organisational. Always applicable, priority P1. NIST SP 800-53: SR-2.	**Customer evidence**
T6.2.2	Criteria for assessing suppliers' security, applied in selecting them. Organisational. Always applicable, priority P2. NIST SP 800-53: SR-6.	**Customer evidence**
T6.2.3	Security requirements for procured hardware and software, products evaluated and tested before use, and upgrades applied promptly. Enforced, coverage partial. Always applicable, priority P2. NIST SP 800-53: SR-3, SR-7, SR-9, SR-10, SR-11. **Scope of this evidence** The evidence is that the images of the workloads outside the exempt namespaces come from the registries the entity allows, by a fixed tag or digest, and carry a signature from a configured key (an image no key covers fails); by default a failing Pod is admitted with a warning and recorded. Evaluating and testing products, applying upgrades and maintenance promptly, and other procurement are the entity's to evidence. Fails in Deployment payments/legacy-batch, Pod payments/legacy-batch-779458f596-fwgcd, ReplicaSet payments/legacy-batch-779458f596. policy.images-signed **Fail** 3 object(s) fail the policy. policy.no-latest-tag **Pass** Every object the policy judged passes it. policy.registries-allowed **Pass** Every object the policy judged passes it. trivy.image-sources **Pass** Every object outside the exempt namespaces passes.	**Fail**
T6.2.4	Security requirements, including compliance with applicable law, written into contracts with suppliers. Organisational. Always applicable, priority P2. NIST SP 800-53: SA-4, SR-3.	**Customer evidence**
T6.2.5	Suppliers' adherence to their security obligations monitored, and changes to their services managed. Organisational. Always applicable, priority P2. NIST SP 800-53: SA-9, SR-6, RA-9, SR-7.	**Customer evidence**

T7 Information Systems Acquisition, Development and Maintenance

T7.1 Information Systems Acquisition, Development and Maintenance Policy

T7.1.1	An approved policy for secure acquisition, development and maintenance of systems. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: SA-3, SA-15, SA-17.	**Customer evidence**

T7.2 Security Requirements of Information Systems

T7.2.1	Security requirements identified when developing or acquiring applications. Organisational. Always applicable, priority P1. NIST SP 800-53: SA-4.	**Customer evidence**
T7.2.2	Secure engineering principles documented and applied. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: SA-8.	**Customer evidence**

T7.3 Secure Software Development

T7.3.1	Approved secure coding practices, automated security testing and developer training. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: SI-3.	**Customer evidence**
T7.3.2	Outsourced development governed by quality assurance, contract terms and checks of the delivered code. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: SA-4, SA-10, SA-11, SA-15, SR-2, SR-4.	**Customer evidence**
T7.3.3	Access to source code, development tools and libraries controlled, reviewed and logged. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: AC-3, AC-6, CM-5.	**Customer evidence**

T7.4 Secure Testing and Deployment

T7.4.1	Changes controlled: tracked, every version kept and verified, rollback defined and changes tested before production. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: CM-3, SA-10, SI-2.	**Customer evidence**
T7.4.2	Test data protected: production data copied to test only with approval and erased after testing. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: SA-15(9).	**Customer evidence**
T7.4.3	Security testing of new or changed systems, with acceptance criteria, before they go into operation. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: CM-3, CM-4, SI-2, CA-2, SA-11, SA-4, SR-5(2).	**Customer evidence**

T8 Information Security Incident Management

T8.1 Information Security Incident Management Policy

T8.1.1	An incident management policy, with the Emirate, sector and national reporting requirements. Organisational. Applicable by risk assessment, priority P2. NIST SP 800-53: IR-1.	**Customer evidence**
T8.1.2	Staff and third parties aware of their duty to report security events, and of how to do so. Organisational. Always applicable, priority P1. NIST SP 800-53: AU-6, IR-6, SI-2.	**Customer evidence**

T8.2 Management of Information Security Incidents and Improvements

T8.2.1	An approved incident response plan and an Incident Response Team. Organisational. Always applicable, priority P1. NIST SP 800-53: IR-4, IR-7, IR-8.	**Customer evidence**

T8.2.2	Incidents classified and prioritised taking account of the Cyber Security Council's issuances, and those to be reported to national agencies identified. Organisational. Always applicable, priority P2. NIST SP 800-53: none mapped by the regime.	**Customer evidence**
T8.2.3	Training and certification for incident response staff. Organisational. Always applicable, priority P2. NIST SP 800-53: IR-2.	**Customer evidence**
T8.2.4	Incident response tested periodically and gaps remedied. Organisational. Always applicable, priority P2. NIST SP 800-53: IR-3.	**Customer evidence**
T8.2.5	The data to collect about each incident defined, recorded and protected. Organisational. Always applicable, priority P2. NIST SP 800-53: IR-5.	**Customer evidence**
T8.2.6	Records of incidents and their lessons kept, with controls added to prevent recurrence. Organisational. Applicable by risk assessment, priority P4. NIST SP 800-53: IR-4.	**Customer evidence**
T8.2.7	Procedures for identifying, collecting and preserving evidence, with chain of custody. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: AU-4, AU-9, AU-10(3), AU-11.	**Customer evidence**

T9 Information Systems Continuity Management

T9.1 ICT Readiness for Business Continuity

T9.1.1	ICT readiness for continuity planned and tested, with continuity plans for response and recovery. Organisational. Always applicable, priority P2. NIST SP 800-53: CP-2, CP-6, CP-7, CP-8, CP-9, CP-10, CP-11, CP-13.	**Customer evidence**
T9.1.2	Security controls kept working during disruption, with compensating controls where they cannot be. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: CP-2, CP-6, CP-7.	**Customer evidence**

T9.2 Testing, Maintaining and Reassessing Plans

T9.2.1	Continuity plans tested periodically and kept current from the lessons learned. Organisational. Applicable by risk assessment, priority P3. NIST SP 800-53: CP-4.	**Customer evidence**

Checks

cluster.access-review Recorded

Every subject bound to a role, with what it may do, for the periodic access review.

109 role bindings to subjects; 2 to cluster-admin across the cluster.

- cluster-admin: Group kubeadm:cluster-admins (kubeadm:cluster-admins)
- cluster-admin: Group system:masters (cluster-admin)

Scope	Subject	Role
cluster	Group kubeadm:cluster-admins	ClusterRole cluster-admin
cluster	Group system:authenticated	ClusterRole calico-tiered-policy-passthrough

cluster	Group system:authenticated	ClusterRole system:basic-user
cluster	Group system:authenticated	ClusterRole system:discovery
cluster	Group system:authenticated	ClusterRole system:public-info-viewer
cluster	Group system:bootstrappers:kubeadm:default-node-token	ClusterRole kubeadm:get-nodes
cluster	Group system:bootstrappers:kubeadm:default-node-token	ClusterRole system:certificates.k8s.io:certificatesigningrequests:nodec
cluster	Group system:bootstrappers:kubeadm:default-node-token	ClusterRole system:node-bootstrapper
cluster	Group system:masters	ClusterRole cluster-admin
cluster	Group system:monitoring	ClusterRole system:monitoring
cluster	Group system:nodes	ClusterRole system:certificates.k8s.io:certificatesigningrequests:selfno
cluster	Group system:serviceaccounts	ClusterRole system:service-account-issuer-discovery
cluster	Group system:unauthenticated	ClusterRole system:public-info-viewer
cluster	ServiceAccount calico-system/calico-apiserver	ClusterRole calico-crds
cluster	ServiceAccount calico-system/calico-apiserver	ClusterRole calico-extension-apiserver-auth-access
cluster	ServiceAccount calico-system/calico-apiserver	ClusterRole calico-webhook-reader
cluster	ServiceAccount calico-system/calico-apiserver	ClusterRole system:auth-delegator
cluster	ServiceAccount calico-system/calico-cni-plugin	ClusterRole calico-cni-plugin
cluster	ServiceAccount calico-system/calico-kube-controllers	ClusterRole calico-kube-controllers
cluster	ServiceAccount calico-system/calico-node	ClusterRole calico-node
cluster	ServiceAccount calico-system/calico-typha	ClusterRole calico-typha
cluster	ServiceAccount compliance/compliance-collector	ClusterRole compliance-collector
cluster	ServiceAccount kube-system/attachdetach-controller	ClusterRole system:controller:attachdetach-controller
cluster	ServiceAccount kube-system/certificate-controller	ClusterRole system:controller:certificate-controller
cluster	ServiceAccount kube-system/clusterrole-aggregation-controller	ClusterRole system:controller:clusterrole-aggregation-cont
cluster	ServiceAccount kube-system/coredns	ClusterRole system:coredns
cluster	ServiceAccount kube-system/cronjob-controller	ClusterRole system:controller:cronjob-controller

cluster	ServiceAccount kube-system/daemon-set-controller	ClusterRole system:controller:daemon-set-controller
cluster	ServiceAccount kube-system/deployment-controller	ClusterRole system:controller:deployment-controller
cluster	ServiceAccount kube-system/device-taint-eviction-controller	ClusterRole system:controller:device-taint-eviction-controll
cluster	ServiceAccount kube-system/disruption-controller	ClusterRole system:controller:disruption-controller
cluster	ServiceAccount kube-system/endpoint-controller	ClusterRole system:controller:endpoint-controller
cluster	ServiceAccount kube-system/endpointslice-controller	ClusterRole system:controller:endpointslice-controller
cluster	ServiceAccount kube-system/endpointslicemirroring-controller	ClusterRole system:controller:endpointslicemirroring-contr
cluster	ServiceAccount kube-system/ephemeral-volume-controller	ClusterRole system:controller:ephemeral-volume-controlle
cluster	ServiceAccount kube-system/expand-controller	ClusterRole system:controller:expand-controller
cluster	ServiceAccount kube-system/generic-garbage-collector	ClusterRole system:controller:generic-garbage-collector
cluster	ServiceAccount kube-system/horizontal-pod-autoscaler	ClusterRole system:controller:horizontal-pod-autoscaler
cluster	ServiceAccount kube-system/job-controller	ClusterRole system:controller:job-controller
cluster	ServiceAccount kube-system/kube-dns	ClusterRole system:kube-dns
cluster	ServiceAccount kube-system/kube-proxy	ClusterRole system:node-proxier
cluster	ServiceAccount kube-system/legacy-service-account-token-cleaner	ClusterRole system:controller:legacy-service-account-toke cleaner
cluster	ServiceAccount kube-system/namespace-controller	ClusterRole system:controller:namespace-controller
cluster	ServiceAccount kube-system/node-controller	ClusterRole system:controller:node-controller
cluster	ServiceAccount kube-system/persistent-volume-binder	ClusterRole system:controller:persistent-volume-binder
cluster	ServiceAccount kube-system/pod-garbage-collector	ClusterRole system:controller:pod-garbage-collector
cluster	ServiceAccount kube-system/pv-protection-controller	ClusterRole system:controller:pv-protection-controller
cluster	ServiceAccount kube-system/pvc-protection-controller	ClusterRole system:controller:pvc-protection-controller
cluster	ServiceAccount kube-system/replicaset-controller	ClusterRole system:controller:replicaset-controller
cluster	ServiceAccount kube-system/replication-controller	ClusterRole system:controller:replication-controller

cluster	ServiceAccount kube-system/resource-claim-controller	ClusterRole system:controller:resource-claim-controller
cluster	ServiceAccount kube-system/resourcequota-controller	ClusterRole system:controller:resourcequota-controller
cluster	ServiceAccount kube-system/root-ca-cert-publisher	ClusterRole system:controller:root-ca-cert-publisher
cluster	ServiceAccount kube-system/route-controller	ClusterRole system:controller:route-controller
cluster	ServiceAccount kube-system/selinux-warning-controller	ClusterRole system:controller:selinux-warning-controller
cluster	ServiceAccount kube-system/service-account-controller	ClusterRole system:controller:service-account-controller
cluster	ServiceAccount kube-system/service-cidrs-controller	ClusterRole system:controller:service-cidrs-controller
cluster	ServiceAccount kube-system/service-controller	ClusterRole system:controller:service-controller
cluster	ServiceAccount kube-system/statefulset-controller	ClusterRole system:controller:statefulset-controller
cluster	ServiceAccount kube-system/ttl-after-finished-controller	ClusterRole system:controller:ttl-after-finished-controller
cluster	ServiceAccount kube-system/ttl-controller	ClusterRole system:controller:ttl-controller
cluster	ServiceAccount kube-system/validatingadmissionpolicy-status-controller	ClusterRole system:controller:validatingadmissionpolicy-st controller
cluster	ServiceAccount kube-system/volumeattributesclass-protection-controller	ClusterRole system:controller:volumeattributesclass-protec controller
cluster	ServiceAccount kyverno/engine-admission-controller	ClusterRole kyverno:admission-controller
cluster	ServiceAccount kyverno/engine-admission-controller	ClusterRole view
cluster	ServiceAccount kyverno/engine-background-controller	ClusterRole kyverno:background-controller
cluster	ServiceAccount kyverno/engine-background-controller	ClusterRole view
cluster	ServiceAccount kyverno/engine-cleanup-controller	ClusterRole kyverno:cleanup-controller
cluster	ServiceAccount kyverno/engine-reports-controller	ClusterRole kyverno:reports-controller
cluster	ServiceAccount kyverno/engine-reports-controller	ClusterRole view
cluster	ServiceAccount local-path-storage/local-path-provisioner-service-account	ClusterRole local-path-provisioner-role
cluster	ServiceAccount tigera-operator/tigera-operator	ClusterRole tigera-operator

cluster	ServiceAccount trivy-system/trivy-operator-engine	ClusterRole trivy-operator
cluster	User kube-apiserver-kubelet-client	ClusterRole system:kubelet-api-admin
cluster	User system:kube-controller-manager	ClusterRole calico-tier-getter
cluster	User system:kube-controller-manager	ClusterRole system:kube-controller-manager
cluster	User system:kube-proxy	ClusterRole system:node-proxier
cluster	User system:kube-scheduler	ClusterRole system:kube-scheduler
cluster	User system:kube-scheduler	ClusterRole system:volume-scheduler
bahriya-system	ServiceAccount compliance/compliance-collector	ClusterRole bke-platform-reader
calico-system	ServiceAccount calico-system/goldmane	Role goldmane
calico-system	ServiceAccount tigera-operator/tigera-operator	ClusterRole tigera-operator-secrets
compliance-node	ServiceAccount compliance/compliance-collector	Role compliance-node-collector
kube-public	ServiceAccount kube-system/bootstrap-signer	Role system:controller:bootstrap-signer
kube-public	User system:anonymous	Role kubeadm:bootstrap-signer-clusterinfo
kube-system	Group system:bootstrappers:kubeadm:default-node-token	Role kube-proxy
kube-system	Group system:bootstrappers:kubeadm:default-node-token	Role kubeadm:kubelet-config
kube-system	Group system:bootstrappers:kubeadm:default-node-token	Role kubeadm:nodes-kubeadm-config
kube-system	Group system:nodes	Role kubeadm:kubelet-config
kube-system	Group system:nodes	Role kubeadm:nodes-kubeadm-config
kube-system	ServiceAccount calico-system/calico-apiserver	Role extension-apiserver-authentication-reader
kube-system	ServiceAccount kube-system/bootstrap-signer	Role system:controller:bootstrap-signer
kube-system	ServiceAccount kube-system/cloud-provider	Role system:controller:cloud-provider
kube-system	ServiceAccount kube-system/kube-controller-manager	Role system::leader-locking-kube-controller-manager
kube-system	ServiceAccount kube-system/kube-scheduler	Role system::leader-locking-kube-scheduler
kube-system	ServiceAccount kube-system/leader-election-controller	Role system::leader-locking-kube-controller-manager
kube-system	ServiceAccount kube-system/token-cleaner	Role system:controller:token-cleaner

kube-system	User system:kube-controller-manager	Role extension-apiserver-authentication-reader
kube-system	User system:kube-controller-manager	Role system::leader-locking-kube-controller-manager
kube-system	User system:kube-scheduler	Role extension-apiserver-authentication-reader
kube-system	User system:kube-scheduler	Role system::leader-locking-kube-scheduler
kyverno	ServiceAccount kyverno/engine-admission-controller	Role kyverno:admission-controller
kyverno	ServiceAccount kyverno/engine-background-controller	Role kyverno:background-controller
kyverno	ServiceAccount kyverno/engine-cleanup-controller	Role kyverno:cleanup-controller
kyverno	ServiceAccount kyverno/engine-reports-controller	Role kyverno:reports-controller
local-path-storage	ServiceAccount local-path-storage/local-path-provisioner-service-account	Role local-path-provisioner-role
tigera-operator	ServiceAccount tigera-operator/tigera-operator	ClusterRole tigera-operator-secrets
trivy-system	ServiceAccount trivy-system/trivy-operator-engine	Role trivy-operator-engine
trivy-system	ServiceAccount trivy-system/trivy-operator-engine	Role trivy-operator-engine-leader-election

cluster.scan-freshness Pass

When each workload was last scanned and how current the vulnerability database was.

Every workload outside the exempt namespaces was scanned within 48 hours.

- scans older than 48 hours fail
- 2 workload(s) outside the exempt namespaces

cluster.vulnerability-age Recorded

How long each open critical and high vulnerability has been reported.

No critical or high vulnerability is open outside the exempt namespaces.

- ages are counted from the first run of this collector that reported each one

cluster.workload-inventory Recorded

An inventory of the cluster's namespaces, workloads and the images they run.

13 namespaces, 17 workloads, 15 distinct images.

Namespace	Workload	Images
compliance	CronJob compliance-collect-nca-ecc	images.example.com:5000/tersane/convoy/compliance/core/collector:1.x.y@sh

compliance	CronJob compliance-collect-qcb	images.example.com:5000/tersane/convoy/compliance/core/collector:1.x.y@sh
compliance	Deployment compliance-web-view	images.example.com:5000/tersane/core/nginxinc/nginx-unprivileged:1.x.y@sha256:f5116ed4606f3eb015dd4a8e28cb9b351476dc0ad4
compliance-node	Job node-facts-888mp	images.example.com:5000/tersane/convoy/compliance/core/collector:1.x.y@sh
kube-system	DaemonSet kube-proxy	registry.k8s.io/kube-proxy:1.x.y
kube-system	Deployment coredns	registry.k8s.io/coredns/coredns:1.x.y
kyverno	Deployment engine-admission-controller	images.example.com:5000/tersane/core/kyverno/kyvernopre:1.x.y@sha256:27 images.example.com:5000/tersane/core/kyverno/kyverno:1.x.y@sha256:c8c4c
kyverno	Deployment engine-background-controller	images.example.com:5000/tersane/core/kyverno/background-controller:1.x.y@sha256:c46e124f21ef8a6bb4b48de57106808b1007edcd5ddd
kyverno	Deployment engine-cleanup-controller	images.example.com:5000/tersane/core/kyverno/cleanup-controller:1.x.y@sha256:d7340aa7c9315080c3dafa7b765d090ced37353e89e5
kyverno	Deployment engine-reports-controller	images.example.com:5000/tersane/core/kyverno/reports-controller:1.x.y@sha2
local-path-storage	Deployment local-path-provisioner	docker.io/kindest/local-path-provisioner:1.x.y
payments	Deployment ledger	registry.example.com/payments/ledger:1.x.y
payments	Deployment legacy-batch	registry.example.com/payments/legacy-batch:1.x.y
tigera-operator	Deployment tigera-operator	quay.io/tigera/operator:1.x.y
trivy-system	Deployment trivy-operator-engine	images.example.com:5000/tersane/core/aquasecurity/trivy-operator:1.x.y@sha
trivy-system	Job scan-vulnerabilityreport-5cbf456f74	images.example.com:5000/tersane/core/aquasecurity/trivy:1.x.y@sha256:790f
trivy-system	StatefulSet trivy-server	images.example.com:5000/tersane/core/aquasecurity/trivy:1.x.y@sha256:790f

node.etcd-data Pass

etcd's data directory on each control-plane node is owned by root and closed to other users (700).

etcd's data directory is owned by root and closed to other users: so on all 1 node(s).

- example-production-control-plane: 1 path(s) examined

node.file-permissions Pass

The Kubernetes configuration, credentials, certificates and keys on each node are owned by root and no more open than their limits (600 for credentials, keys and manifests, 644 for certificates).

Kubernetes' configuration, credentials and keys are owned by root and no more open than their limits: so on all 1 node(s).

- example-production-control-plane: 33 path(s) examined

node.kubelet-authn Pass

The kubelet refuses anonymous requests and authenticates clients by certificate and webhook.

The kubelet refuses anonymous requests and authenticates clients by certificate and webhook: so on all 1 node(s).

- Read from the kubelet's configuration and flag files: a change made after the kubelet last started is not yet in effect, and is not visible to the node collector.

node.kubelet-authz Pass

The kubelet authorises requests through the API server (Webhook mode).

The kubelet authorises requests through the API server: so on all 1 node(s).

- Read from the kubelet's configuration and flag files: a change made after the kubelet last started is not yet in effect, and is not visible to the node collector.

node.kubelet-ciphers Fail

The kubelet offers TLS 1.2 or later with AEAD cipher suites only.

The kubelet offers TLS 1.2 or later with AEAD cipher suites only: not so on 1 node(s).

- Read from the kubelet's configuration and flag files: a change made after the kubelet last started is not yet in effect, and is not visible to the node collector.
- example-production-control-plane: minimum not set, suites not set

How to fix On a Bahriya Kubernetes Engine cluster, set compliance.controlPlane.enabled to true in config.yaml (the default) and upgrade each node with upgrade.sh, which writes tlsMinVersion and tlsCipherSuites into every kubelet's configuration. Elsewhere, set tlsMinVersion to VersionTLS12 and tlsCipherSuites to AEAD suites only (GCM or ChaCha20-Poly1305) in the kubelet's configuration, then restart the kubelet. (Bahriya Tersane)

Fails (2)

Object	What is wrong	Full result
Node example-production-control-plane	the minimum TLS version is not set, which allows TLS 1.0	`What was read on the node is in this run's snapshot.json, under nodes.`
Node example-production-control-plane	no cipher suites are set, so the defaults, which include CBC suites, are offered	`What was read on the node is in this run's snapshot.json, under nodes.`

node.kubelet-hardening Pass

The kubelet keeps its read-only port closed and rotates its client certificates.

The kubelet keeps its read-only port closed and rotates its certificates: so on all 1 node(s).

- Read from the kubelet's configuration and flag files: a change made after the kubelet last started is not yet in effect, and is not visible to the node collector.
- example-production-control-plane: protectKernelDefaults is false (recorded, not judged: turning it on requires kernel settings the platform does not yet apply)

node.time-sync Not evaluated

Each node runs a time service configured with the servers the platform names.

The platform names no time servers, so there is nothing to compare the nodes with.

platform.anonymous Not evaluated

Anonymous requests to the API server are limited to the health endpoints.

Anonymous requests restricted to the health endpoints: the platform could not observe it.

- compliance.anonymous.configured: unknown
- compliance.anonymous.observed: unknown

platform.audit Pass

The API server writes an audit log under BKE's audit policy, which records requests by authenticated users, including privileged ones.

API audit logging: in effect on every control-plane node.

- compliance.audit.configured: yes
- compliance.audit.observed: yes
- audit.policy: bke

platform.audit-rotation Fail

How long the API server keeps its audit log locally before rotation.

The API server keeps its audit log for 30 days, fewer than the 365 the regime requires. A log collected to a central store may meet the requirement; that store is the customer's to evidence.

- audit.rotation: maxage=30 maxbackup=10 maxsize=100
- the regime's log-retention-days: 365

How to fix Keep the audit log for at least the period the regime requires. Either forward it to a central log store that keeps it that long, which BKE's Fluent Bit can do, and give that store to your assessor as evidence; or raise compliance.audit.maxAgeDays in /etc/bke/config.yaml, with maxBackups and maxSizeMB large enough to hold that period, and run upgrade.sh at the version the cluster already runs on each control-plane node, then apply.sh --commit. (Bahriya Tersane)

Fails (1)

Object	What is wrong	Full result
Platform API server audit log	kept 30 days before rotation; the regime requires 365	`kubectl -n bahriya-system get configmap bke-platform -o jsonpath='{.data.audit\.rotation}'`

platform.control-plane-tls Not evaluated

Profiling is off and only AEAD TLS cipher suites are offered by the control plane and the kubelets.

The control-plane TLS and profiling settings: the platform could not observe it.

- compliance.control-plane.configured: unknown
- compliance.control-plane.observed: unknown

platform.host-firewall Not evaluated

The nodes' own ports are filtered by the cluster's host firewall. The collector reads that BKE has declared host endpoints for the nodes; what they admit from outside the cluster (the API server, SSH, the NodePorts and ICMP) is BKE's host firewall policy (SR-2026-010), which the collector does not read.

The platform could not count the host endpoints.

- host-network.host-endpoints: unknown
- host-network.ssh-ports: unknown

platform.key-provider Not evaluated

Where the encryption key is held (a local key, or a key management service through KMS v2).

The platform could not observe the encryption provider.

- encryption.provider: unknown

platform.mesh-cni Not evaluated

Meshed Pods run without elevated privileges, through the mesh's CNI plugin.

The service mesh CNI plugin: the platform could not observe it.

- compliance.mesh-cni.configured: unknown
- compliance.mesh-cni.observed: unknown

platform.oidc Not evaluated

People sign in to the cluster through the organisation's identity provider, which enforces its own authentication factors.

The platform could not read the authentication configuration.

- authentication.oidc-issuers: unknown

platform.pod-security Not evaluated

Pod Security Admission defaults are applied to namespaces, with the exemptions listed.

Pod Security admission defaults: the platform could not observe it.

- compliance.pod-security.configured: unknown
- compliance.pod-security.observed: unknown
- pod-security.defaults: unknown
- pod-security.exemptions: unknown

platform.secret-encryption Not evaluated

Secrets, and optionally ConfigMaps, are encrypted at rest in etcd.

The platform could not observe Secret encryption.

- encryption.observed: unknown
- encryption.resources.configured: unknown

platform.time-servers Not provided

The time servers each node synchronises with.

No time servers are set for the cluster; each node keeps its own.

- compliance.time-servers.configured: unknown

How to fix Name the organisation's time servers in compliance.time.servers in /etc/bke/config.yaml and run apply.sh --commit, then upgrade.sh --node-settings --commit on every node. That command moves no version and restarts only the time service. (Bahriya Tersane)

platform.versions Recorded

The BKE and Kubernetes versions the cluster runs, against which supported versions are judged.

Bahriya Kubernetes Engine 1.x.y, Kubernetes v1.x.y, observed 2026-11-07T14:34:28Z.

- bke-version: 1.x.y
- kubernetes-version: v1.x.y
- observed-at: 2026-11-07T14:34:28Z

policy.default-deny-generated Not provided

A default-deny NetworkPolicy is generated in every application namespace (when the customer turns it on).

The policy tersane-compliance-default-deny is not in the cluster: it is set to Off, or the Convoy's policies are not installed.

How to fix Set generate.defaultDenyNetworkPolicy.enabled to true in the Convoy's values and upgrade the Convoy. Where the policy is present but not ready, Kyverno's own status for it says why: kubectl describe generatingpolicy tersane-compliance-default-deny. (Bahriya Tersane)

policy.images-signed Fail

Every image outside the exempt namespaces carries a signature from a configured key: Kyverno verifies the images that match a key's patterns, and the collector fails any image that no key covers. A failing Pod is admitted with a warning and recorded, or refused, as policies.defaultAction sets (Warn by default).

3 object(s) fail the policy.

- tersane-compliance-images-signed: ready
- tersane-compliance-images-signed-example-bank: ready
- 3 object(s) passed, 3 failed
- the collector was not given the image patterns the signing keys cover, so images no key covers are not counted

How to fix Sign the image with cosign and add the public key, with a pattern matching the image, under signatures.keys in the Convoy's values. If the image is already signed, add the key that signed it. Then upgrade the Convoy. (Bahriya Tersane)

Fails (3)

Object	What is wrong	Full result
Deployment payments/legacy-batch	images not signed by example-bank: registry.example.com/payments/legacy-batch:1.x.y	`kubectl get policyreport 37ec704c-84bb-42c9-ae23-78958efceed3 -n payments -o yaml`
Pod payments/legacy-batch-779458f596-fwgcd	images not signed by example-bank: registry.example.com/payments/legacy-batch:1.x.y	`kubectl get policyreport 715816ab-6d7f-462a-83f5-88a2c3c1acfb -n payments -o yaml`

ReplicaSet payments/legacy-batch-779458f596	images not signed by example-bank: registry.example.com/payments/legacy-batch:1.x.y	`kubectl get policyreport 60ee71ad-b7f6-490e-8185-a36093c453d0 -n payments -o yaml`

policy.ingress-tls Not evaluated

Every Ingress serves its hosts over TLS.

The policy has judged no object yet: Kyverno has reported no result for it.

- tersane-compliance-ingress-tls: ready
- 0 object(s) passed, 0 failed

policy.namespace-classification Fail

Every application namespace carries a data classification label.

1 object(s) fail the policy.

- tersane-compliance-namespace-classification: ready
- 1 object(s) passed, 1 failed

How to fix Label the namespace tersane.bahriya.cloud/data-classification with one of the values in namespaces.classifications in the Convoy's values. (Bahriya Tersane)

Fails (1)

Object	What is wrong	Full result
Namespace default	the namespace needs the label tersane.bahriya.cloud/data-classification, one of: top-secret, secret, restricted, public	`kubectl get clusterpolicyreport 01c02383-ae51-479c-a362-355224387a37 -o yaml`

policy.namespace-environment Fail

Every application namespace carries an environment label, one of the values in namespaces.environments.

1 object(s) fail the policy.

- tersane-compliance-namespace-environment: ready
- 1 object(s) passed, 1 failed

How to fix Label the namespace tersane.bahriya.cloud/environment with one of the values in namespaces.environments in the Convoy's values. (Bahriya Tersane)

Fails (1)

Object	What is wrong	Full result
Namespace default	the namespace needs the label tersane.bahriya.cloud/environment, one of: production, staging, test, development	`kubectl get clusterpolicyreport 01c02383-ae51-479c-a362-355224387a37 -o yaml`

policy.namespace-network-policy Fail

Every application namespace has at least one NetworkPolicy.

1 object(s) fail the policy.

- tersane-compliance-namespace-network-policy: ready
- 1 object(s) passed, 1 failed

How to fix Add NetworkPolicies to the namespace that allow only the traffic its workloads need. To start every new namespace closed, set generate.defaultDenyNetworkPolicy.enabled to true in the Convoy's values. (Bahriya Tersane)

Fails (1)

Object	What is wrong	Full result
Namespace default	namespace default holds no NetworkPolicy	`kubectl get clusterpolicyreport 01c02383-ae51-479c-a362-355224387a37 -o yaml`

policy.no-default-token Pass

Pods say whether they mount a service account token, so that none receives one by default.

Every object the policy judged passes it.

- tersane-compliance-no-default-token: ready
- 2 object(s) passed, 0 failed

policy.no-latest-tag Pass

Images are referenced by a fixed tag or digest, never by a moving tag.

Every object the policy judged passes it.

- tersane-compliance-no-latest-tag: ready
- 2 object(s) passed, 0 failed

policy.registries-allowed Pass

Containers run images only from the registries the customer allows.

Every object the policy judged passes it.

- tersane-compliance-registries-allowed: ready
- 2 object(s) passed, 0 failed

policy.resource-limits Fail

Every container declares memory and CPU limits.

1 object(s) fail the policy.

- tersane-compliance-resource-limits: ready
- 1 object(s) passed, 1 failed

How to fix Set resources.limits.cpu and resources.limits.memory on every container of the Pod template, at values the workload has been measured to need. (Bahriya Tersane)

Fails (1)

Object	What is wrong	Full result
Deployment payments/legacy-batch	containers without CPU and memory limits: app	`kubectl get policyreport 37ec704c-84bb-42c9-ae23-78958efceed3 -n payments -o yaml`

trivy.external-ip Pass

Services with external IP addresses.

Every object outside the exempt namespaces passes.

- trivy checks: AVD-KSV-0108
- 15 object(s) with a passing result, 0 failing, 0 failing in exempt namespaces

trivy.image-sources Pass

Images from public registries (AVD-KSV-0034), or referenced by a moving tag (AVD-KSV-0013). AVD-KSV-0125 is not used: its list trusts only three cloud providers' registries, and policy.registries-allowed judges the registries against your own list instead. See https://tersane.bahriya.cloud/docs/charts/compliance/#checks-the-convoy-does-not-use-and-why.

Every object outside the exempt namespaces passes.

- trivy checks: AVD-KSV-0013, AVD-KSV-0034
- 17 object(s) with a passing result, 0 failing, 0 failing in exempt namespaces

trivy.limits Fail

Containers without CPU or memory limits.

1 object(s) fail.

- trivy checks: AVD-KSV-0011, AVD-KSV-0018
- 11 object(s) with a passing result, 1 failing, 5 failing in exempt namespaces

How to fix AVD-KSV-0011 Set a limit value under 'containers[].resources.limits.cpu'. (trivy-operator)

How to fix AVD-KSV-0018 Set a limit value under 'containers[].resources.limits.memory'. (trivy-operator)

Fails (2)

Object	What is wrong	Full result
Deployment payments/legacy-batch	AVD-KSV-0011: CPU not limited. Container 'app' of ReplicaSet 'legacy-batch-779458f596' should set 'resources.limits.cpu' Listed under Fix first with policy.resource-limits.	`kubectl get configauditreport replicaset-legacy-batch-779458f596 -n payments -o yaml`
Deployment payments/legacy-batch	AVD-KSV-0018: Memory not limited. Container 'app' of ReplicaSet 'legacy-batch-779458f596' should set 'resources.limits.memory' Listed under Fix first with policy.resource-limits.	`kubectl get configauditreport replicaset-legacy-batch-779458f596 -n payments -o yaml`

Fails in exempt namespaces, not counted (28)

Object	What is wrong	Full result
APIServer calico-system/default	AVD-KSV-0011: CPU not limited. Container 'calico-apiserver' of ReplicaSet 'calico-apiserver-7876c5c67c' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-calico-apiserver-7876c5c67c -n calico-system -o yaml`
APIServer calico-system/default	AVD-KSV-0018: Memory not limited. Container 'calico-apiserver' of ReplicaSet 'calico-apiserver-7876c5c67c' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-calico-apiserver-7876c5c67c -n calico-system -o yaml`
Deployment tigera-operator/tigera-operator	AVD-KSV-0011: CPU not limited. Container 'tigera-operator' of ReplicaSet 'tigera-operator-686489b657' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-tigera-operator-686489b657 -n tigera-operator -o yaml`

Deployment tigera-operator/tigera-operator	AVD-KSV-0018: Memory not limited. Container 'tigera-operator' of ReplicaSet 'tigera-operator-686489b657' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-tigera-operator-686489b657 -n tigera-operator -o yaml`
Goldmane calico-system/default	AVD-KSV-0011: CPU not limited. Container 'goldmane' of ReplicaSet 'goldmane-7d6f669648' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-goldmane-7d6f669648 -n calico-system -o yaml`
Goldmane calico-system/default	AVD-KSV-0018: Memory not limited. Container 'goldmane' of ReplicaSet 'goldmane-7d6f669648' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-goldmane-7d6f669648 -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'calico-csi' of DaemonSet 'csi-node-driver' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'calico-kube-controllers' of ReplicaSet 'calico-kube-controllers-7c69d5967' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-calico-kube-controllers-7c69d5967 -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'calico-node' of DaemonSet 'calico-node' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'calico-typha' of ReplicaSet 'calico-typha-7864dc55f5' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-calico-typha-7864dc55f5 -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'cni-plugins' of DaemonSet 'calico-node' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'csi-node-driver-registrar' of DaemonSet 'csi-node-driver' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'ebpf-bootstrap' of DaemonSet 'calico-node' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'flexvol-driver' of DaemonSet 'calico-node' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0011: CPU not limited. Container 'install-cni' of DaemonSet 'calico-node' should set 'resources.limits.cpu'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'calico-csi' of DaemonSet 'csi-node-driver' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'calico-kube-controllers' of ReplicaSet 'calico-kube-controllers-7c69d5967' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-calico-kube-controllers-7c69d5967 -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'calico-node' of DaemonSet 'calico-node' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`

Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'calico-typha' of ReplicaSet 'calico-typha-7864dc55f5' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-calico-typha-7864dc55f5 -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'cni-plugins' of DaemonSet 'calico-node' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'csi-node-driver-registrar' of DaemonSet 'csi-node-driver' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'ebpf-bootstrap' of DaemonSet 'calico-node' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'flexvol-driver' of DaemonSet 'calico-node' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0018: Memory not limited. Container 'install-cni' of DaemonSet 'calico-node' should set 'resources.limits.memory'	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Whisker calico-system/default	AVD-KSV-0011: CPU not limited. Container 'whisker' of ReplicaSet 'whisker-7469665d8' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-whisker-7469665d8 -n calico-system -o yaml`
Whisker calico-system/default	AVD-KSV-0011: CPU not limited. Container 'whisker-backend' of ReplicaSet 'whisker-7469665d8' should set 'resources.limits.cpu'	`kubectl get configauditreport replicaset-whisker-7469665d8 -n calico-system -o yaml`
Whisker calico-system/default	AVD-KSV-0018: Memory not limited. Container 'whisker' of ReplicaSet 'whisker-7469665d8' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-whisker-7469665d8 -n calico-system -o yaml`
Whisker calico-system/default	AVD-KSV-0018: Memory not limited. Container 'whisker-backend' of ReplicaSet 'whisker-7469665d8' should set 'resources.limits.memory'	`kubectl get configauditreport replicaset-whisker-7469665d8 -n calico-system -o yaml`

trivy.network-policy-selectors Pass

NetworkPolicies whose selectors select nothing.

Every object outside the exempt namespaces passes.

- trivy checks: AVD-KSV-0038
- 185 object(s) with a passing result, 0 failing, 0 failing in exempt namespaces

trivy.rbac-exec Pass

Roles that allow executing into Pods or impersonating privileged groups.

Every object outside the exempt namespaces passes.

- trivy checks: AVD-KSV-0053, AVD-KSV-0043
- 185 object(s) with a passing result, 0 failing, 3 failing in exempt namespaces

Fails in exempt namespaces, not counted (3)

Object	What is wrong	Full result
ClusterRole admin	Kubernetes default role; AVD-KSV-0053: Exec into Pods. ClusterRole 'admin' should not have access to resource '["pods/exec"]' for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-admin -o yaml`
ClusterRole edit	Kubernetes default role; AVD-KSV-0053: Exec into Pods. ClusterRole 'edit' should not have access to resource '["pods/exec"]' for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-edit -o yaml`
ClusterRole system:aggregate-to-edit	Kubernetes default role; AVD-KSV-0053: Exec into Pods. ClusterRole 'system:aggregate-to-edit' should not have access to resource '["pods/exec"]' for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-7b884bc5d8 -o yaml`

trivy.rbac-privilege Fail

Roles and bindings that grant administrative access, wildcard access, access to Secrets, or anonymous access.

2 object(s) fail.

- trivy checks: AVD-KSV-0041, AVD-KSV-0044, AVD-KSV-0045, AVD-KSV-0047, AVD-KSV-0051, AVD-KSV-0052, AVD-KSV-0111, AVD-KSV-0112, AVD-KSV-0113, AVD-KSV-0114, AVD-KSV-0122, AVD-KSV-0123
- 185 object(s) with a passing result, 2 failing, 17 failing in exempt namespaces

How to fix AVD-KSV-0041 Manage secrets are not allowed. Remove resource 'secrets' from cluster role (trivy-operator)

How to fix AVD-KSV-0114 Remove webhook configuration resources/verbs, acceptable values for verbs ['get', 'list', 'watch'] (trivy-operator)

Fails (3)

Object	What is wrong	Full result
ClusterRole tigera-operator	AVD-KSV-0041: Manage secrets. ClusterRole 'tigera-operator' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-tigera-operator -o yaml`
ClusterRole tigera-operator	AVD-KSV-0114: Manage webhookconfigurations. ClusterRole 'tigera-operator' should not have access to resources ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"] for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-tigera-operator -o yaml`
ClusterRole tigera-operator-secrets	AVD-KSV-0041: Manage secrets. ClusterRole 'tigera-operator-secrets' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-tigera-operator-secrets -o yaml`

Fails in exempt namespaces, not counted (18)

Object	What is wrong	Full result

ClusterRole admin	Kubernetes default role; AVD-KSV-0041: Manage secrets. ClusterRole 'admin' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-admin -o yaml`
ClusterRole cluster-admin	Kubernetes default role; AVD-KSV-0044: No wildcard verb and resource roles. Role permits wildcard verb on wildcard resource	`kubectl get clusterrbacassessmentreport clusterrole-cluster-admin -o yaml`
ClusterRole edit	Kubernetes default role; AVD-KSV-0041: Manage secrets. ClusterRole 'edit' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-edit -o yaml`
ClusterRole kyverno:admission-controller	AVD-KSV-0114: Manage webhookconfigurations. ClusterRole 'kyverno:admission-controller' should not have access to resources ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"] for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-78fb9f7b45 -o yaml`
ClusterRole kyverno:admission-controller:core	AVD-KSV-0114: Manage webhookconfigurations. ClusterRole 'kyverno:admission-controller:core' should not have access to resources ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"] for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-66dd849fcd -o yaml`
ClusterRole kyverno:cleanup-controller	AVD-KSV-0114: Manage webhookconfigurations. ClusterRole 'kyverno:cleanup-controller' should not have access to resources ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"] for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-5fc98dd666 -o yaml`
ClusterRole kyverno:cleanup-controller:core	AVD-KSV-0114: Manage webhookconfigurations. ClusterRole 'kyverno:cleanup-controller:core' should not have access to resources ["mutatingwebhookconfigurations", "validatingwebhookconfigurations"] for verbs ["create", "update", "patch", "delete", "deletecollection", "impersonate", "*"]	`kubectl get clusterrbacassessmentreport clusterrole-757c5df67 -o yaml`

ClusterRole system:aggregate-to-edit	Kubernetes default role; AVD-KSV-0041: Manage secrets. ClusterRole 'system:aggregate-to-edit' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-7b884bc5d8 -o yaml`
ClusterRole system:controller:legacy-service-account-token-cleaner	Kubernetes default role; AVD-KSV-0041: Manage secrets. ClusterRole 'system:controller:legacy-service-account-token-cleaner' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-864fd65956 -o yaml`
ClusterRole system:kube-controller-manager	Kubernetes default role; AVD-KSV-0041: Manage secrets. ClusterRole 'system:kube-controller-manager' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-b754c4cc6 -o yaml`
ClusterRole system:node	Kubernetes default role; AVD-KSV-0041: Manage secrets. ClusterRole 'system:node' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-54cdc9b678 -o yaml`
ClusterRole trivy-operator	AVD-KSV-0041: Manage secrets. ClusterRole 'trivy-operator' shouldn't have access to manage resource 'secrets'	`kubectl get clusterrbacassessmentreport clusterrole-trivy-operator -o yaml`
ClusterRole trivy-operator	AVD-KSV-0047: Do not allow privilege escalation from node proxy. Role permits privilege escalation from node proxy	`kubectl get clusterrbacassessmentreport clusterrole-trivy-operator -o yaml`
Role kyverno/kyverno:admission-controller	AVD-KSV-0113: Manage namespace secrets. Role 'kyverno:admission-controller' shouldn't have access to manage secrets in namespace 'kyverno'	`kubectl get rbacassessmentreport role-78fb9f7b45 -n kyverno -o yaml`
Role kyverno/kyverno:background-controller	AVD-KSV-0113: Manage namespace secrets. Role 'kyverno:background-controller' shouldn't have access to manage secrets in namespace 'kyverno'	`kubectl get rbacassessmentreport role-586b8c778d -n kyverno -o yaml`
Role kyverno/kyverno:cleanup-controller	AVD-KSV-0113: Manage namespace secrets. Role 'kyverno:cleanup-controller' shouldn't have access to manage secrets in namespace 'kyverno'	`kubectl get rbacassessmentreport role-5fc98dd666 -n kyverno -o yaml`
Role kyverno/kyverno:reports-controller	AVD-KSV-0113: Manage namespace secrets. Role 'kyverno:reports-controller' shouldn't have access to manage secrets in namespace 'kyverno'	`kubectl get rbacassessmentreport role-899885f55 -n kyverno -o yaml`
Role trivy-system/trivy-operator-engine	AVD-KSV-0113: Manage namespace secrets. Role 'trivy-operator-engine' shouldn't have access to manage secrets in namespace 'trivy-system'	`kubectl get rbacassessmentreport role-trivy-operator-engine -n trivy-system -o yaml`

trivy.run-as-root Fail

Containers that run as the root user.

1 object(s) fail.

- trivy checks: AVD-KSV-0012, AVD-KSV-0105
- 185 object(s) with a passing result, 1 failing, 2 failing in exempt namespaces

How to fix AVD-KSV-0012 Set 'containers[].securityContext.runAsNonRoot' to true. (trivy-operator)

Fails (1)

Object	What is wrong	Full result
Deployment payments/legacy-batch	AVD-KSV-0012: Runs as root user. Container 'app' of ReplicaSet 'legacy-batch-779458f596' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport replicaset-legacy-batch-779458f596 -n payments -o yaml`

Fails in exempt namespaces, not counted (10)

Object	What is wrong	Full result
Deployment tigera-operator/tigera-operator	AVD-KSV-0012: Runs as root user. Container 'tigera-operator' of ReplicaSet 'tigera-operator-686489b657' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport replicaset-tigera-operator-686489b657 -n tigera-operator -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'calico-csi' of DaemonSet 'csi-node-driver' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'calico-node' of DaemonSet 'calico-node' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'cni-plugins' of DaemonSet 'calico-node' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'csi-node-driver-registrar' of DaemonSet 'csi-node-driver' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'ebpf-bootstrap' of DaemonSet 'calico-node' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'flexvol-driver' of DaemonSet 'calico-node' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0012: Runs as root user. Container 'install-cni' of DaemonSet 'calico-node' should set 'securityContext.runAsNonRoot' to true	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0105: Containers must not set runAsUser to 0. securityContext.runAsUser should be set to a value greater than 0	`kubectl get configauditreport daemonset-calico-node -n calico-system -o yaml`
Installation calico-system/default	AVD-KSV-0105: Containers must not set runAsUser to 0. securityContext.runAsUser should be set to a value greater than 0	`kubectl get configauditreport daemonset-csi-node-driver -n calico-system -o yaml`

trivy.seccomp Fail

Containers without the runtime's default seccomp profile.

1 object(s) fail.

- trivy checks: AVD-KSV-0030, AVD-KSV-0104
- 15 object(s) with a passing result, 1 failing, 1 failing in exempt namespaces

How to fix AVD-KSV-0030 Set spec.securityContext.seccompProfile.type to RuntimeDefault on the Pod, and remove any container's seccompProfile that sets another type. (Bahriya Tersane)

How to fix AVD-KSV-0104 Specify seccomp either by annotation or by seccomp profile type having allowed values as per pod security standards (trivy-operator)

Fails (2)

Object	What is wrong	Full result
Deployment payments/legacy-batch	AVD-KSV-0030: Runtime/Default Seccomp profile not set. Either Pod or Container should set 'securityContext.seccompProfile.type' to 'RuntimeDefault'	`kubectl get configauditreport replicaset-legacy-batch-779458f596 -n payments -o yaml`
Deployment payments/legacy-batch	AVD-KSV-0104: Seccomp policies disabled. container "app" of replicaset "legacy-batch-779458f596" in "payments" namespace should specify a seccomp profile	`kubectl get configauditreport replicaset-legacy-batch-779458f596 -n payments -o yaml`

Fails in exempt namespaces, not counted (2)

Object	What is wrong	Full result
Deployment tigera-operator/tigera-operator	AVD-KSV-0030: Runtime/Default Seccomp profile not set. Either Pod or Container should set 'securityContext.seccompProfile.type' to 'RuntimeDefault'	`kubectl get configauditreport replicaset-tigera-operator-686489b657 -n tigera-operator -o yaml`
Deployment tigera-operator/tigera-operator	AVD-KSV-0104: Seccomp policies disabled. container "tigera-operator" of replicaset "tigera-operator-686489b657" in "tigera-operator" namespace should specify a seccomp profile	`kubectl get configauditreport replicaset-tigera-operator-686489b657 -n tigera-operator -o yaml`

trivy.token-automount Pass

Pods that mount a service account token.

Every object outside the exempt namespaces passes.

- trivy checks: AVD-KSV-0036
- 185 object(s) with a passing result, 0 failing, 0 failing in exempt namespaces

trivy.vulnerabilities Pass

Every workload's images are scanned for known vulnerabilities, classified by severity, against a vulnerability database refreshed every six hours.

No container outside the exempt namespaces runs an image with a critical or high vulnerability.

- 25 container image(s) scanned

Fails in exempt namespaces, not counted (10)

Object	What is wrong	Full result
CronJob compliance/compliance-collect-nca-ecc (collect)	container collect: 3 critical, 37 high	`kubectl get vulnerabilityreport cronjob-compliance-collect-nca-ecc-collect -n compliance -o yaml`

CronJob compliance/compliance-collect-qcb (collect)	container collect: 3 critical, 37 high	`kubectl get vulnerabilityreport cronjob-compliance-collect-qcb-collect -n compliance -o yaml`
Deployment kyverno/engine-admission-controller (kyverno)	container kyverno: 0 critical, 1 high	`kubectl get vulnerabilityreport replicaset-engine-admission-controller-f8dc4d544-kyverno -n kyverno -o yaml`
Deployment kyverno/engine-admission-controller (kyverno-pre)	container kyverno-pre: 0 critical, 1 high	`kubectl get vulnerabilityreport replicaset-engine-admission-controller-f8dc4d544-kyverno-pre -n kyverno -o yaml`
Deployment kyverno/engine-background-controller (controller)	container controller: 0 critical, 1 high	`kubectl get vulnerabilityreport replicaset-engine-background-controller-d76fdbc67-controller -n kyverno -o yaml`
Deployment kyverno/engine-cleanup-controller (controller)	container controller: 0 critical, 1 high	`kubectl get vulnerabilityreport replicaset-engine-cleanup-controller-664cc866f8-controller -n kyverno -o yaml`
Deployment kyverno/engine-reports-controller (controller)	container controller: 0 critical, 1 high	`kubectl get vulnerabilityreport replicaset-engine-reports-controller-554c5f5bb4-controller -n kyverno -o yaml`
Installation calico-system/default (calico-node)	container calico-node: 1 critical, 8 high	`kubectl get vulnerabilityreport daemonset-calico-node-calico-node -n calico-system -o yaml`
Installation calico-system/default (ebpf-bootstrap)	container ebpf-bootstrap: 1 critical, 8 high	`kubectl get vulnerabilityreport daemonset-calico-node-ebpf-bootstrap -n calico-system -o yaml`
StatefulSet trivy-system/trivy-server (trivy-server)	container trivy-server: 1 critical, 4 high	`kubectl get vulnerabilityreport statefulset-trivy-server-trivy-server -n trivy-system -o yaml`