PUBLICATION

EXAM-TM40O-POST

POST-TEST — SL 4O: PLATFORM ENGINEER

Maven Smart System (MSS) — USAREUR-AF

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

26 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

POST-TEST — SL 4O: PLATFORM ENGINEER

MAVEN SMART SYSTEM (MSS) — USAREUR-AF

Field	Detail
Course	SL 4O: Platform Engineer
Form	Post-Test
Level	SL 4O (Specialist)
Audience	Platform engineers completing SL 4O training
Time Allowed	45 minutes
Passing Score	70% (39/56)

INSTRUCTIONS

This assessment evaluates knowledge and skills gained during SL 4O training. Answer all questions. A score of 70% or higher is required for course credit.

SECTION 1 — MULTIPLE CHOICE

Circle the letter of the best answer. (2 points each)

1. The "platform-as-product" mindset means:

A. The platform team sells infrastructure services to application teams B. The platform is treated as an internal product serving application developers — with users, a backlog, a roadmap, and self-service capabilities C. The platform uses the same technology stack as the applications D. The platform team reports to the product management organization

2. In a GitOps workflow, what happens if someone manually applies a change directly to the cluster without committing to Git?

A. The change persists permanently B. Git automatically updates to match the cluster C. The cluster crashes D. The GitOps controller detects the drift and reverts the cluster to match the Git state

3. When hardening a container image for MSS, which of the following is NOT a required step?

A. Use an Iron Bank base image B. Run as non-root with dropped capabilities C. Install debugging tools (bash, curl, netcat) for troubleshooting in production D. Pin the image by SHA256 digest in deployment manifests

4. A CI/CD security gate for "SCA" (Software Composition Analysis) checks:

A. Whether the application's third-party dependencies contain known vulnerabilities B. Whether the application's source code follows coding standards C. Whether the container runs as root D. Whether the deployment manifest has proper RBAC

5. "Default deny" network policy means:

A. All network traffic is allowed unless explicitly blocked B. All network traffic is blocked unless explicitly allowed — pods cannot communicate with anything until allow rules are created C. Only DNS traffic is denied by default D. Network policies are disabled by default

6. In an air-gapped deployment, container images must be:

A. Pre-bundled and transferred via approved media — nothing downloads from external sources during deployment B. Downloaded from Docker Hub at deploy time C. Built from source code on the air-gapped cluster D. Stored as base64-encoded text in Kubernetes ConfigMaps

7. Kubernetes liveness probes are used to:

A. Check if a pod has enough CPU resources B. Verify that the container image is signed C. Monitor network traffic between pods D. Determine if a container is still running and healthy — if the probe fails, Kubernetes restarts the container

8. Resource quotas in Kubernetes prevent:

A. Users from creating too many Git commits B. Pods from communicating across namespaces C. Any single namespace from consuming disproportionate cluster resources — enforcing CPU, memory, and object count limits D. Container images from exceeding a maximum file size

---

SECTION 2 — SHORT ANSWER

Answer in 2–3 sentences. (5 points each)

9. Explain the difference between "imperative" and "declarative" infrastructure management. Why does SL 4O require declarative approaches?

10. What is "drift detection" in a GitOps context? Why is drift dangerous, and how is it resolved?

11. Describe two security benefits of pinning container images by SHA256 digest instead of by tag.

12. What is the purpose of a readiness probe vs. a liveness probe in Kubernetes? What happens if you configure a liveness probe but not a readiness probe?

SECTION 3 — SCENARIO

Answer in 5–8 sentences. (10 points each)

13. You are deploying an MSS application to an air-gapped environment. During deployment, the CI/CD pipeline fails because it cannot download a Python dependency from PyPI. Describe: (a) why this happened, (b) what should have been done during the bundling phase to prevent it, (c) how you would fix it now, and (d) what process change you would implement to prevent recurrence.

14. An application team reports that their application is running out of memory and getting OOMKilled by Kubernetes. They ask you to "just increase the memory limit." Describe your response: what would you investigate before changing limits? What platform-level controls should be in place? What is the risk of simply increasing limits without investigation?

SCORING SUMMARY

Section	Questions	Points Each	Total Points
Multiple Choice	8	2	16
Short Answer	4	5	20
Scenario	2	10	20
Total	—	—	**56**

Passing: 39/56 (70%) — Post-test only. Pre-test is diagnostic.

ANSWER KEY — INSTRUCTOR USE ONLY

Do not distribute to students.

Multiple Choice: 1. B — Platform-as-product treats the platform as an internal product with users, a backlog, a roadmap, and self-service capabilities. 2. D — The GitOps controller detects drift between Git and cluster state, then reverts the cluster to match Git. 3. C — Debugging tools (bash, curl, netcat) should NOT be installed in hardened production images; they increase attack surface. 4. A — SCA checks third-party dependencies for known vulnerabilities. 5. B — Default deny blocks all traffic unless explicitly allowed; pods cannot communicate until allow rules are created. 6. A — Air-gapped deployments require pre-bundled images transferred via approved media; nothing downloads externally during deployment. 7. D — Liveness probes determine if a container is running and healthy; Kubernetes restarts the container if the probe fails. 8. C — Resource quotas prevent any single namespace from consuming disproportionate cluster resources (CPU, memory, object count limits).

Short Answer Guidance:

SA-9. Full credit: imperative = issuing step-by-step commands to reach a desired state ("create this pod, then scale to 3 replicas"); declarative = describing the desired end state and letting the system figure out how to get there ("I want 3 replicas of this pod running"). SL 4O requires declarative because: it is auditable (the desired state is in Git), reproducible (any engineer can re-apply the same manifest), and drift-detectable (GitOps controllers compare declared state to actual state). Partial credit (3 pts) for correct distinction without connecting to SL 4O requirements.

SA-10. Full credit: drift = the running cluster state no longer matches what is declared in Git; it is dangerous because the system behaves differently from what engineers expect, changes are undocumented and unreproducible, and a rebuild from Git would lose the drifted changes; resolution: the GitOps controller continuously reconciles — detecting drift and reverting the cluster to match Git (or alerting if manual intervention is needed). Partial credit (3 pts) for correct definition without explaining danger and resolution.

SA-11. Full credit: two benefits — (1) immutability: a SHA256 digest is a content hash that cannot be changed without changing the image itself, preventing tag mutation attacks (someone pushes a malicious image to the same tag); (2) reproducibility: the digest guarantees the exact same image bytes are deployed every time, regardless of when or where the deployment runs. Additional valid benefit: auditability — the digest creates a verifiable link between the deployed artifact and the build pipeline that produced it. Must name two distinct security benefits. Partial credit (3 pts) for one benefit with explanation.

SA-12. Full credit: readiness probe = determines if a container is ready to accept traffic; liveness probe = determines if a container is still alive (should be restarted if not). If you configure liveness but not readiness: Kubernetes will restart unhealthy containers (good), but it will also route traffic to containers that are still starting up (bad) — users see errors during pod startup, rolling updates, or restarts because traffic hits pods before they are ready to serve. Must distinguish both probes and explain the consequence of missing readiness. Partial credit (3 pts) for correct definitions without consequence explanation.

Scenario Guidance:

S-13. Full credit (10 pts): (a) the failure occurred because the air-gapped environment has no internet access — PyPI is unreachable; (b) during bundling: all Python dependencies should have been downloaded to a local registry or vendored into the deployment package using `pip download` or a requirements bundle — the CI/CD pipeline should include a dependency completeness check before transfer; (c) immediate fix: on a connected system, download the missing dependency, transfer it via approved media to the air-gapped environment, add it to the local package repository, and re-run the pipeline; (d) process change: add a dependency completeness gate to the bundling pipeline — run a full `pip install --dry-run` against only the bundled repository to verify all dependencies resolve before transfer. Must address all four parts (a–d). Partial credit (5 pts) for two parts. Deduct 3 pts if student suggests connecting the air-gapped network to the internet.

S-14. Full credit (10 pts): before changing limits, investigate: (1) is the application actually leaking memory, or is the limit set too low for legitimate workload? Check memory usage trends over time; (2) review the application's resource requests vs. limits vs. actual usage; (3) check for memory leaks (steadily increasing memory over time vs. a spike at peak load); (4) platform-level controls: resource quotas should cap what any namespace can consume (prevents one team from claiming all cluster memory), LimitRanges should set default and max limits for pods; (5) risk of blindly increasing: if the application has a memory leak, increasing the limit only delays the OOMKill — the leak continues growing and may eventually impact other workloads on the node, or the application could consume memory that other pods need. Must investigate before acting, cite platform controls, and explain the risk. Partial credit (5 pts) for investigation without platform controls or risk explanation.

USAREUR-AF Operational Data Team TM-40O Post-Test | Version 1.0 | March 2026