

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

PUBLICATION

# EXAM-TM400-PRE



---

## PRE-TEST — SL 40: PLATFORM ENGINEER

---

*Maven Smart System (MSS) — USAREUR-AF*

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

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

**26 MARCH 2026**

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

## PRE-TEST — SL 40: PLATFORM ENGINEER

### MAVEN SMART SYSTEM (MSS) — USAREUR-AF

Field	Detail
Course	SL 40: Platform Engineer
Form	Pre-Test
Level	SL 40 (Specialist)
Audience	Platform engineers; prerequisite: SL 1+20+30 + Linux admin + containers + Git familiarity
Time Allowed	30 minutes
Passing Score	N/A — diagnostic only

### INSTRUCTIONS

This diagnostic assessment establishes your baseline knowledge before training. Your score does not affect course eligibility. Answer honestly — results help the instructor tailor instruction to gaps.

### SECTION 1 — MULTIPLE CHOICE

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

**1. In Kubernetes, a "Pod" is:**

A. A virtual machine running a Linux operating system  
B. The smallest deployable unit — one or more containers that share networking and storage  
C. A load balancer that distributes traffic across nodes  
D. A configuration file that defines cluster settings

**2. A Kubernetes "Deployment" provides:**

A. A way to run a single container once and then terminate B. Persistent storage for database files C. Network routing rules for external traffic D. Declarative updates for Pods — managing replicas, rolling updates, and rollback

**3. "Infrastructure as Code" means:**

A. Writing application code that runs on infrastructure B. Using a programming language to build hardware C. Managing infrastructure through version-controlled configuration files rather than manual processes D. Documenting infrastructure architecture in a wiki

**4. A container image "tag" (e.g., `nginx:latest`) is problematic in production because:**

A. Tags are mutable — someone can push a different image to the same tag, so `latest` today and `latest` tomorrow may be different images B. Tags are too long for Kubernetes manifests C. Tags consume more storage than image IDs D. Tags are not supported in Kubernetes

**5. "GitOps" means:**

A. Using Git only for application source code B. Using Git as the single source of truth for both application and infrastructure state, with automated reconciliation between Git and the running system C. Using GitHub instead of GitLab D. Running Git commands manually to deploy applications

**6. In a CI/CD pipeline, "shift left" refers to:**

A. Moving deployment to earlier stages of development B. Shifting work from developers to operations C. Moving the pipeline configuration to the left side of the screen D. Moving security testing, quality checks, and validation as early as possible in the development lifecycle — catching issues sooner

**7. An "air-gapped" network is:**

A. A network that is physically isolated from external networks — no internet or external connectivity B. A network with high bandwidth but high latency C. A wireless network with gaps in coverage D. A network that uses air-cooling instead of liquid-cooling for its servers

**8. RBAC (Role-Based Access Control) in Kubernetes is used to:**

A. Control which containers can communicate with each other B. Manage container image versions C. Define which users and service accounts can perform which actions on which resources D. Configure storage classes for persistent volumes

---

## SECTION 2 — SHORT ANSWER

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

**9. What is the difference between a container and a virtual machine? When would you choose one over the other?**

10. Explain what "resource requests" and "resource limits" mean in Kubernetes. Why are both important?
11. What is a "network policy" in Kubernetes? What happens if no network policy is applied to a namespace?
12. Describe the difference between a "rolling update" and a "blue/green" deployment strategy. When would you choose each?

## SECTION 3 — SCENARIO

Answer in 3–5 sentences. (10 points)

13. A developer on your team manually SSH'd into a production server and changed a configuration file to fix an urgent bug. The fix worked. Explain why this is problematic from an IaC/GitOps perspective, and what should have been done instead.

## SCORING SUMMARY

Section	Questions	Points Each	Total Points
Multiple Choice	8	2	16
Short Answer	4	5	20
Scenario	1	10	10
<b>Total</b>	—	—	<b>46</b>

Passing: N/A — Pre-test is diagnostic only.

## ANSWER KEY — INSTRUCTOR USE ONLY

Do not distribute to students.

**Multiple Choice:** 1. B — A Pod is the smallest deployable unit in Kubernetes: one or more containers sharing networking and storage. 2. D — A Deployment provides declarative updates for Pods, managing replicas, rolling updates, and rollback. 3. C — IaC manages infrastructure through version-controlled

configuration files rather than manual processes. 4. A — Tags are mutable; someone can push a different image to the same tag, making deployments non-reproducible. 5. B — GitOps uses Git as the single source of truth for application and infrastructure state with automated reconciliation. 6. D — Shift left moves security testing, quality checks, and validation as early as possible in the development lifecycle. 7. A — An air-gapped network is physically isolated from external networks with no internet or external connectivity. 8. C — RBAC defines which users and service accounts can perform which actions on which Kubernetes resources.

### Short Answer Guidance:

SA-9. Full credit: a container shares the host OS kernel and isolates the application in a user-space process — lightweight, fast startup, portable; a VM includes its own full OS kernel running on a hypervisor — heavier, slower startup, stronger isolation. Choose containers for microservices, stateless workloads, and rapid scaling; choose VMs when full OS isolation is required (e.g., running different OS versions, security-sensitive workloads requiring hardware-level isolation). Partial credit (3 pts) for correct distinction without use-case reasoning.

SA-10. Full credit: resource requests = the guaranteed minimum resources Kubernetes will schedule for the container (used for scheduling decisions); resource limits = the maximum resources the container is allowed to consume (enforced at runtime — container is throttled or OOMKilled if exceeded). Both are important: requests without limits allow a single pod to consume all node resources (noisy neighbor); limits without requests may cause pods to be scheduled on nodes that cannot actually support them. Partial credit (3 pts) for correct definitions without explaining why both matter.

SA-11. Full credit: a network policy is a Kubernetes resource that controls ingress and egress traffic to/from pods based on labels, namespaces, or IP blocks. Without any network policy applied, the default behavior is "allow all" — every pod can communicate with every other pod in the cluster, which violates the principle of least privilege and increases the blast radius of a compromised pod. Partial credit (3 pts) for correct definition without explaining the default behavior risk.

SA-12. Full credit: rolling update replaces pods incrementally (old pods are terminated as new pods become healthy) — zero downtime but both versions run simultaneously during the transition; blue/green deploys a complete parallel environment (green) alongside the existing one (blue), then switches traffic all at once. Choose rolling update for routine deployments with backward-compatible changes; choose blue/green when you need instant rollback capability or when the old and new versions cannot coexist. Partial credit (3 pts) for correct descriptions without guidance on when to choose each.

### Scenario Guidance:

S-13. Full credit (10 pts): (1) the manual SSH change creates configuration drift — the running system no longer matches the Git-committed state; (2) the change is undocumented, unreviewable, and unreproducible — if the server is rebuilt or another instance is deployed, the fix will be missing; (3) the GitOps controller may detect the drift and revert the change, undoing the fix; (4) correct approach: commit the configuration change to Git, let the CI/CD pipeline apply it through the normal process — this ensures the fix is reviewed, versioned, tested, and reproducible; (5) even for urgent fixes, the Git-first process

should be followed (expedited review, not bypassed review). Must address drift, reproducibility, and the correct procedure. Partial credit (5 pts) for identifying the problem without describing the correct procedure. Deduct 3 pts if student endorses the manual fix as acceptable for emergencies.

---

*USAREUR-AF Operational Data Team TM-400 Pre-Test | Version 1.0 | March 2026*

DRAFT