CERTLABZ CERTIFICATION BOARD

Certified DevSecOps Engineer (CDSO)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working DevSecOps engineer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1 — Domain 1: DevSecOps Foundations — DATA

Emma Wilson maps each DevSecOps principle to its aim, with one aim missing.

Principle	Aim
Shift left	Find flaws early
Shared ownership	(to choose)
Automation	Reduce manual gates
Fast feedback	Alert on new risk

Table 1.1 Principles and their aims.

What is the aim of shared ownership?

A. Audit the team yearly
B. Security is everyones job
C. Ship without any tests
D. One team signs all risk

Correct answer: B

Shared ownership means the whole team, not a separate gate, is accountable for security, so the aim is that security is everyones job. Yearly audits, skipping tests, or a single sign-off team all run against the collaborative model DevSecOps promotes.

Question 2 — Domain 2: Secure CI/CD Pipelines — CONFIG

James Carter reviews a pipeline where one security stage is placed too late to block a bad build.

```
stages:
  - build
  - unit_test
  - deploy_prod
  - sast_scan
```

Listing 2.1 Pipeline stage order.

What best hardens this pipeline?

A. Delete the unit tests
B. Run SAST after deploy
C. Deploy twice per day
D. Gate before deploy

Correct answer: D
The scan runs after production deploy, so a fix is to move it earlier and gate the deploy on its result, blocking a bad build before release. Deploying more often, dropping tests, or scanning after deploy leaves the vulnerable code live.

Question 3 — Domain 3: Application Security Testing — DATA

Olivia Davis reviews scan findings, each with a severity and a fix status.

Finding	Severity	Status
SQL injection	Critical	Open
Weak cipher	Medium	Fixed
Verbose error	Low	Open
XSS in form	High	Open

Table 3.1 Scan findings and status.

Which finding should be fixed first?

A. Weak cipher
B. SQL injection
C. Verbose error
D. XSS in form

Correct answer: B
The SQL injection is critical and still open, so it carries the highest risk and should be fixed first. The weak cipher is already fixed, and the verbose error and cross-site scripting rank lower in severity than the open critical finding.

Question 4

Domain 4: Infrastructure and Container Security

CONFIG

Liam Anderson reviews a Dockerfile that runs the app with more privilege than it needs.

```
FROM base:latest
COPY app /app
EXPOSE 8080
CMD ["/app/run"]
# runs as root
```

Listing 4.1 Container image build.

What best reduces the container risk?

A. Expose more ports
B. Add a USER line
C. Use latest tag
D. Skip the health check

Correct answer: B
The image runs as root, so adding a non-root USER line limits what a compromised container can do on the host. Using the latest tag hides the version, and exposing more ports or skipping checks widens the attack surface rather than shrinking it.

Question 5

Domain 5: Secrets, Identity and Supply Chain

DATA

Ava Thompson maps each practice to what it protects, with one protection missing.

Practice	Protects
Vault storage	Keeps secrets out of code
Short-lived tokens	(to choose)
Signed artifacts	Proves the build source
SBOM review	Tracks dependencies

Table 5.1 Practices and protection.

What do short-lived tokens protect against?

A. Slow build times
B. Large image size
C. Reuse of leaked creds

D Missing unit tests

Correct answer: C
Short-lived tokens expire quickly, so a leaked credential is useful to an attacker for only a brief window, limiting reuse. Build speed, image size, and test coverage are unrelated to the exposure window that rotating short-lived tokens is meant to shrink.

Question 6

Domain 6: Monitoring, Compliance and Culture

CONFIG

David Brown lists the signals a team collects to detect and prove response to incidents.

```
audit_log:   enabled
alerting:    on_high_sev
retention:   90 days
access_log:  enabled
```

Listing 6.1 Monitoring settings.

Which setting most supports an audit of who did what?

A Alerting
B Retention window
C Audit log
D High-sev filter

Correct answer: C
The audit log records who took which action, so it is the setting that most supports proving accountability during a review. Alerting flags events as they happen, and the retention window and severity filter shape volume rather than capturing the actor.

Question 7

Domain 1: DevSecOps Foundations

A firm keeps its security team as a final gate that reviews code only the week before release, and fixes pile up too late to fix. What change best reflects DevSecOps?

A Add a later gate
B Review once a year
C Embed checks early
D Remove all reviews

Correct answer: C
DevSecOps shifts security left by embedding automated checks early in development so issues surface while they are cheap to fix. Adding a later gate, reviewing yearly, or removing reviews all keep or worsen the late, batched feedback described.

Question 8

Domain 2: Secure CI/CD Pipelines

A pipeline stores its deploy credentials as plain text variables printed in the build log, and anyone with log access can read them. What best fixes this?

A. Print them once
B. Use a secrets store
C. Email the log daily
D. Shorten the log

Correct answer: B
A managed secrets store injects credentials at run time and masks them from logs, so a secrets store keeps them out of readable output. Printing once, shortening, or emailing the log does not stop the credential from being exposed to log readers.

Question 9 — Domain 4: Infrastructure and Container Security

A team pulls base images tagged latest, so builds pull different contents over time and a scanned image differs from the one deployed. What best restores trust?

A. Pin image digests
B. Pull more often
C. Scan less often
D. Use bigger images

Correct answer: A
Pinning a base image by digest means every build pulls the exact same contents, so the scanned image matches the deployed one. Pulling more often, scanning less, or using larger images does nothing to make the base image reproducible.

Question 10 — Domain 5: Secrets, Identity and Supply Chain

A build imports a public package by a floating version, and one release added hidden code that reached production before anyone noticed. What best guards the supply chain?

A. Trust the publisher
B. Import more packages
C. Skip the lockfile
D. Pin and verify deps

Correct answer: D
Pinning dependency versions and verifying their integrity stops an unexpected release from slipping into the build unchecked. Trusting the publisher blindly, adding packages, or skipping the lockfile all widen the exposure the incident revealed.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CDSO-001. https://certlabz.com/certifications/certified-devsecops-engineer.html