



Certified Cloud Migration Specialist (CMGS)

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 cloud migration 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: Migration Strategy and Business Case

DATA

Emma Wilson assigns a migration strategy to four applications.

Application	Situation
Payroll app	Low value, strong SaaS equivalent exists
Reporting tool	Unused for over a year
Order service	High value, worth becoming cloud-native
Internal wiki	Needed fast, modernise later

Table 1.1 Applications and their situations.

Which set of strategies fits best?

- ☐ A Payroll refactor, reporting rehost, order retire, wiki repurchase, matching each app to the most ambitious option available.
- ☐ B Payroll repurchase, reporting retire, order refactor, wiki rehost, matching each app to its value, usage and timeline.
- ☐ C Payroll retain, reporting refactor, order rehost, wiki retire, keeping the low-value apps and rewriting the unused ones first.
- ☐ D Payroll rehost, reporting repurchase, order retain, wiki refactor, lifting everything unchanged and rewriting the wiki now.

Correct answer: B

A low-value app with a good SaaS equivalent is a repurchase, an unused app is a retire, a high-value app worth going cloud-native is a refactor, and an app needed fast is a rehost to be modernised later. The other options misassign these, for example refactoring an app that should simply be retired.

Question 2

Domain 2: Discovery, Assessment and Dependencies

DATA

James Carter reviews discovery data for a database server before migration.

Metric	Value
On-premises spec	32 vCPU, 256 GB RAM
Average CPU	4%
Peak CPU	11%
Average memory used	38 GB

Table 2.1 Utilisation of the database server.

What should the migration plan do with this server, and why?

- A** Match the exact 32 vCPU, 256 GB spec in the cloud, because keeping the same size guarantees the migration cannot affect performance.
- B** Right-size to a much smaller instance sized to real utilisation, because the on-premises server is heavily oversized for its load.
- C** Refactor the database engine before moving, because low CPU usage always indicates the engine itself must be replaced first.
- D** Retain the server on-premises indefinitely, because a database at low utilisation is not a suitable candidate for any migration.

Correct answer: B

The server runs at 4% average and 11% peak CPU using 38 GB of 256 GB, so it is heavily oversized. Right-sizing to real utilisation is a core migration saving. Matching the oversized spec wastes money, low CPU does not imply the engine must change, and low utilisation is not a reason to keep a workload on-premises.

Question 3

Domain 4: Migration Execution and Data Migration

CONFIG

Olivia Davis plans a database cutover. The draft plan is below.

```
plan:
  1. take a nightly dump of the source DB
  2. next morning, import the dump into the cloud DB
  3. switch the app to the cloud DB
  # tolerance: 30 minutes downtime, minimal data loss
```

Listing 3.1 Draft database cutover plan.

Why does this plan fail the stated tolerance, and what fixes it?

- A** It fails because the dump is too small; taking several dumps a day and importing the largest one would meet the tolerance.
- B** It fails because all writes between the nightly dump and the switch are lost, and the load spans downtime; use bulk load plus change data capture.
- C** It fails only because the import runs in the morning; running the same nightly-dump plan overnight instead would meet the tolerance.
- D** It does not fail; a nightly dump and morning import is a zero-downtime, zero-loss approach that already meets the stated tolerance.

Correct answer: B

A single nightly dump misses every change made after it was taken, breaching the minimal-data-loss goal, and the app is down while the dump imports, breaching the 30-minute window. A bulk load followed by change data capture keeps the target in sync so the final cutover is short and loses little or no data.

Question 4

Domain 2: Discovery, Assessment and Dependencies

DATA

Liam Anderson reviews dependencies before assigning waves.

Component	Dependency
Orders API	Reads and writes the shared Sales DB heavily
Billing API	Reads the shared Sales DB heavily
Sales DB	Used by Orders and Billing
Marketing site	No dependency on the others

Table 4.1 Components and their dependencies.

How should these be waved?

- A** Migrate the Sales DB alone in wave one, then Orders and Billing in later separate waves, splitting the group across three waves.
- B** Migrate Orders, Billing and the Sales DB together in one wave, and the independent Marketing site can go in any wave.
- C** Migrate Orders in wave one and Billing a year later, leaving the shared Sales DB on-premises to serve both across sites.
- D** Migrate the Marketing site last, and split Orders, Billing and the Sales DB across three waves to spread the workload evenly.

Correct answer: B

Orders, Billing and the Sales DB are tightly coupled through heavy shared-database traffic, so they belong in the same wave to avoid chatty, high-latency cross-site calls. The Marketing site has no dependency and can move whenever convenient. Splitting the coupled group across waves is exactly what creates cross-site latency and cost.

Question 5

Domain 3: Planning, Landing Zone and Cutover Design

CONFIG

Ava Thompson reviews the readiness of a migration programme before wave one.

```
landing zone      : not built yet
on-prem connectivity: none
runbooks          : none
rollback plans    : none
plan              : "just start migrating servers Monday"
```

Listing 5.1 Programme readiness check.

What must happen before wave one begins?

- ☐ A Start migrating on Monday as planned and build the landing zone, connectivity and runbooks in parallel as issues come up.
- ☐ B Build the governed landing zone and hybrid connectivity, and write runbooks with rollback, before any production workload moves.
- ☐ C Skip the landing zone entirely, because modern migration tooling provisions all networking and identity automatically per server.
- ☐ D Migrate only the most critical systems first, because starting with high-value workloads proves the programme fastest to leadership.

Correct answer: B

A governed landing zone, hybrid connectivity and runbooks with rollback are prerequisites: migrating into an ungoverned environment with no rollback risks security gaps and unrecoverable outages. Tooling does not remove the need for a designed landing zone, and critical systems should come after early low-risk waves prove the process.

Question 6

Domain 5: Application Modernisation and Refactoring

DATA

A fragile monolith must reach the cloud within a tight deadline, then gain cloud benefits over time.

Option	Approach
A	Big-bang rewrite into microservices before moving
B	Rehost now, then strangle into services incrementally
C	Retain on-premises indefinitely
D	Repurchase without checking business fit

Table 6.1 Options for the monolith.

Which option is the pragmatic path, and why?

- ☐ A Option A, because rewriting the whole monolith into microservices before moving is the fastest way to hit a tight deadline.
- ☐ B Option C, because a fragile monolith should never be migrated and is always best left running on-premises for the long term.

- C** Option B, because rehosting meets the deadline and the strangler pattern then modernises incrementally with far lower risk.
- D** Option D, because repurchasing to any SaaS product immediately removes the monolith regardless of whether it fits the business.

Correct answer: C

Rehosting meets the deadline, and the strangler pattern then replaces the monolith piece by piece with much lower risk than a big-bang rewrite. A full rewrite before the deadline is high-risk and slow, retaining forever forgoes cloud benefits, and repurchasing without checking fit can replace the monolith with something that does not do the job.

Question 7

Domain 1: Migration Strategy and Business Case

A migration business case claims large savings, but the team only compared cloud instance prices to on-premises server prices. What is wrong with the case?

- A** Nothing is wrong, because comparing instance prices to server prices is the complete and correct way to build a migration business case.
- B** It omits operations, licensing, data transfer, right-sizing and the cost of the migration itself, all of which change the real total cost.
- C** It is too detailed, because a business case should only ever consider the headline instance price and ignore every other cost factor.
- D** The only flaw is currency, because instance prices and server prices simply need to be converted before the comparison becomes valid.

Correct answer: B

A credible business case compares total cost of ownership: operations, licensing (which may not transfer), data transfer, right-sizing and the one-off cost of migrating. Comparing raw instance prices to server prices ignores most of the real cost and routinely produces misleading savings figures.

Question 8

Domain 3: Planning, Landing Zone and Cutover Design

A critical workload can tolerate only 30 minutes of downtime at cutover. Which cutover approach fits, and what else is essential?

- A** A weekend-long lift and shift with the application offline, because a longer window makes the migration simpler and safer overall.
- B** Continuous replication with a short final cutover, together with a tested rollback plan in case validation fails after the switch.
- C** A rebuild from backups over two days, because restoring from backup is always the most reliable way to migrate a critical workload.
- D** A live migration with no rollback, because tolerating only 30 minutes of downtime means there is no time to prepare a rollback plan.

Correct answer: B

A tight downtime tolerance calls for continuous replication so the final cutover is short, and every critical cutover needs a tested rollback in case validation fails. A long offline window breaches the tolerance, a two-day rebuild is far too slow, and going live without rollback risks an unrecoverable outage of a critical system.

Question 9

Domain 6: Cutover, Validation, Optimisation and Governance

Two weeks after a lift-and-shift migration, cloud costs are well above the business case. What is the disciplined first response?

- ☐ A Migrate everything back on-premises immediately, because costs above the business case prove the cloud is the wrong platform.
- ☐ B Ignore the overrun, because cloud costs always settle down on their own within the first few months after any migration.
- ☐ C Analyse usage, right-size resources, remove idle environments and apply commitments where demand is stable, then re-baseline.
- ☐ D Buy the largest instances available to guarantee performance, on the basis that performance problems are the usual cause of cost overruns.

Correct answer: C

Cost overruns after a lift and shift are almost always optimisation gaps: oversized resources, idle environments and no commitments. The disciplined response is to analyse usage, right-size, remove waste and commit where demand is stable, then re-baseline. Reverting the migration is drastic, costs do not fix themselves, and larger instances would increase spend.

Question 10

Domain 2: Discovery, Assessment and Dependencies

A team plans a migration from a spreadsheet of servers built from memory, with no automated discovery. What is the main risk?

- ☐ A There is no risk, because a spreadsheet built from the team's knowledge is always a complete and accurate inventory of the estate.
- ☐ B It misses undocumented systems and hidden dependencies, which are the "unknown unknowns" that cause migrations to fail at cutover.
- ☐ C The only risk is formatting, because the data is correct and simply needs to be moved into a database before the migration begins.
- ☐ D It overstates the estate, because manual inventories always list far more servers than actually exist and inflate the migration scope.

Correct answer: B

Automated discovery finds what people forget: undocumented systems and hidden dependencies on other services or on-premises components. These unknown unknowns are a leading cause of migration failures, when a workload is moved without something it silently depends on. A remembered spreadsheet cannot be relied on for a safe migration.