

PRACTICE QUESTIONS

Google Cloud

Professional Cloud Developer

SHARED BY [u/smooth_badger_94](#) on aiexamcerts

EDITION Free Set 2

QUESTIONS 10

SHARED October 15, 2026

EXAM Google Cloud Professional Cloud Developer

Shared by [u/smooth_badger_94](#) on aiexamcerts

10 practice questions for the Google Cloud Professional Cloud Developer exam, each followed by its answer and a short explanation. More free sets and the full practice exam:

www.aiexamcerts.com/google-cloud/professional-cloud-developer. Spotted a mistake? Email support@aiexamcerts.com with the exam name and the question number.

1. A service calls a payment API that sometimes returns HTTP 503 for brief periods. Which client behavior is most appropriate?

- A Retry immediately in a tight loop with no limit on attempts
- B Treat the first 503 as a permanent failure and stop calling the API
- C Retry idempotent operations with capped exponential backoff and jitter
- D Cache the 503 response and return it to callers for 24 hours

Answer: C

Exponential backoff with jitter spreads retries out so a recovering dependency is not overwhelmed, and the cap stops endless attempts. Immediate tight-loop retries amplify load during an outage, which is the retry storm the backoff is meant to prevent. Caching a transient 503 would keep returning a failure long after the API has recovered.

2. Setting min-instances to 1 on a Cloud Run service removes cold starts for every request, including requests that arrive during a traffic spike.

- A True
- B False

Answer: B (False)

One warm instance avoids a cold start only for requests it can take. During a spike, Cloud Run starts additional instances to absorb the load, and those new instances can still cold-start. The setting reduces cold-start latency for baseline traffic rather than eliminating it.

3. A developer wants to test a Pub/Sub publisher locally without using a real cloud project. Which tool is designed for this?

- A gsutil
- B gcloud compute instances create
- C bq mk
- D gcloud beta emulators pubsub start

Answer: D

The Pub/Sub emulator runs locally and lets client code publish and subscribe without touching a real project, which keeps tests fast and free of side effects. gsutil manages Cloud Storage objects, gcloud compute creates virtual machines, and bq works with BigQuery. None of them give a local Pub/Sub endpoint.

4. The built-in Cloud Build substitution _____ holds the ID of the project that runs the build.

Answer: \$PROJECT_ID (also accepted: PROJECT_ID)

PROJECT_ID is one of Cloud Build's built-in substitutions, so it does not need an underscore prefix. Built-in values such as PROJECT_ID, BUILD_ID and COMMIT_SHA are available in every step. User-defined substitutions are the ones that must begin with an underscore.

5. Which two Cloud Run settings help a service absorb sudden traffic spikes? (Choose two.)

- A Set concurrency to 1 so each request gets a full instance
- B Set min-instances above zero to cover the baseline traffic
- C Set max-instances to 1 to control costs
- D Tune concurrency to match what one instance can really handle
- E Disable autoscaling and keep one fixed instance running

Answer: B, D

A warm baseline removes cold starts from the first wave of a spike, and a realistic concurrency value lets each instance serve its share without overload. Concurrency of 1 multiplies the instances needed and slows scaling, while a single fixed or capped instance cannot absorb a spike at all.

6. Which deployment strategy runs the new version alongside the old one, switches all traffic at once, and keeps the old version for instant rollback?

- A Rolling update that replaces pods in place
- B Recreate deployment that stops old instances first
- C Blue/green deployment
- D In-place patching of running containers

Answer: C

Blue/green keeps two complete environments, so the old one stays ready and switching back is a single traffic change. A rolling update gradually replaces instances and does not keep the old version intact for instant rollback. Recreate causes downtime, and in-place patching changes running containers, which breaks reproducibility.

7. A Cloud Run service must read objects in a Cloud Storage bucket. Which authentication approach is recommended?
- A Embed a service account JSON key in the container image
 - B Grant allUsers the Storage Admin role on the bucket
 - C Attach a service account to the service and grant it a narrow role on the bucket
 - D Reuse a personal user's OAuth token stored in an environment variable

Answer: C

Attaching a dedicated service account and granting it a narrow role on the bucket gives the service short-lived credentials with least privilege and no key to rotate. Embedding a key in an image leaks it to anyone who can pull the image. Granting allUsers access makes the bucket public to everyone.

8. Cloud Tasks replaces Cloud Scheduler for cron-style schedules, so Cloud Scheduler is not needed when Cloud Tasks is in use.
- A True
 - B False

Answer: B (False)

Cloud Tasks dispatches tasks that an application enqueues, with rate limits and retries, but it is not a cron service. Cloud Scheduler is the service for recurring schedules, and the two are often used together.

9. A team wants to count log lines that match a specific error message and alert when the count rises. Which feature creates the numeric series?
- A A log-based metric
 - B A log sink to Cloud Storage
 - C A log exclusion filter
 - D A log bucket retention setting

Answer: A

A log-based metric counts entries that match a filter and exposes the count as a time series that alerting policies can use. A sink copies logs elsewhere, an exclusion drops entries before storage, and a retention setting controls how long they are kept. None of those produce a count.

10. Which tool profiles where CPU time is spent in production code without stopping the application?
- A Cloud Trace
 - B Cloud Profiler
 - C Error Reporting
 - D A Cloud Monitoring dashboard

Answer: B

Cloud Profiler samples CPU and memory use in running code and shows which functions consume the most, with low overhead. Cloud Trace measures timing between services rather than inside one function. Error Reporting groups exceptions, and a dashboard shows metrics without locating the code responsible.