



Application Development

GCP Professional Cloud Developer | Cloud Run, App Engine, Cloud Functions & Design Patterns

Cloud Run

Cloud Run is a fully managed serverless platform for running containers. It scales to zero, handles HTTPS automatically, and supports any language/framework that can be containerized. It is the recommended serverless compute for most new applications.



Cloud Run Architecture

Services	Long-running containers serving HTTP/gRPC. Auto-scales 0 to N instances. Revisions for versioning. Traffic splitting for canary/blue-green.
Jobs	Run containers to completion (no serving). For batch processing, migrations, scheduled tasks. Array jobs for parallelism (up to 10,000 tasks).
Concurrency	Up to 1,000 concurrent requests per instance (default 80). Tune based on app threading model. Higher = fewer instances = lower cost.
Sidecars	Multi-container support. Run proxy, logging agent, or service mesh sidecar alongside main container.
Volume mounts	Mount Cloud Storage (FUSE), NFS (Filestore), in-memory (tmpfs). For shared data, model files, etc.
Min instances	Set min=1 to avoid cold starts. Billed for idle instances but keeps container warm.
Startup/liveness probes	HTTP, TCP, or gRPC health checks. Startup probe gates traffic. Liveness probe restarts unhealthy instances.

- Cloud Run integrations: Cloud SQL, Memorystore, Secret Manager, VPC connector
- Execution environments: gen1 (gVisor sandbox), gen2 (full Linux, network filesystem, GPU)

Commands

```
gcloud run deploy SERVICE --image=IMAGE --region=REGION --allow-unauthenticated
gcloud run services update-traffic SERVICE --to-revisions=REV=50
```

Cloud Run

Containers, scale-to-zero

Cloud Functions

Event-driven, serverless

App Engine

PaaS, managed runtimes

GKE

Full Kubernetes control

Cloud Functions

Gen2 (recommended)	Built on Cloud Run. Longer timeout (60 min), larger instances (16 GB), concurrent requests, traffic splitting. Eventarc triggers.
Gen1 (legacy)	Original runtime. 9 min timeout, 8 GB max, single concurrent request. HTTP and background triggers.
HTTP trigger	Invoke via HTTPS URL. Auth via IAM (require authentication) or allow unauthenticated.
Eventarc triggers	Cloud Storage events, Pub/Sub, Firestore, Cloud Audit Logs, custom events. Gen2 only.
CloudEvent format	Gen2 uses CloudEvents standard. Structured event with source, type, data, id, time.

Feature	Gen1	Gen2
Timeout	9 min (HTTP), 9 min (event)	60 min
Memory	Up to 8 GB	Up to 32 GB
vCPUs	Up to 2	Up to 8
Concurrency	1 per instance	Up to 1,000
Traffic splitting	No	Yes
Triggers	HTTP, Pub/Sub, GCS, Firestore	Eventarc (100+ sources)

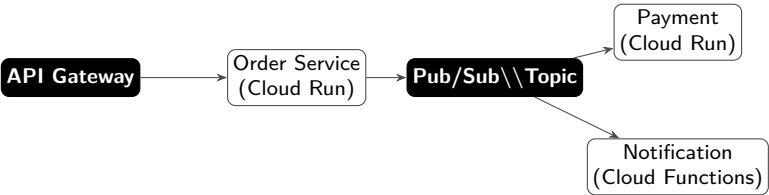
App Engine

Feature	Standard	Flexible
Runtimes	Python, Java, Go, Node, PHP, Ruby	Any Docker container
Scale to zero	Yes	No (min 1 instance)
Instance classes	F1, F2, F4, F4_1G	Custom machine types
Local disk	No (in-memory only)	Yes (ephemeral)
SSH access	No	Yes
Deployment	Source code deploy	Docker image deploy
Startup time	Seconds	Minutes

- App Engine is per-project (one app per project). Cannot delete, only disable
- Versions and traffic splitting for gradual rollouts
- Memcache (Standard), Task Queues (migrated to Cloud Tasks), Cron (cron.yaml or Cloud Scheduler)
- For new projects: prefer Cloud Run over App Engine

Application Design Patterns

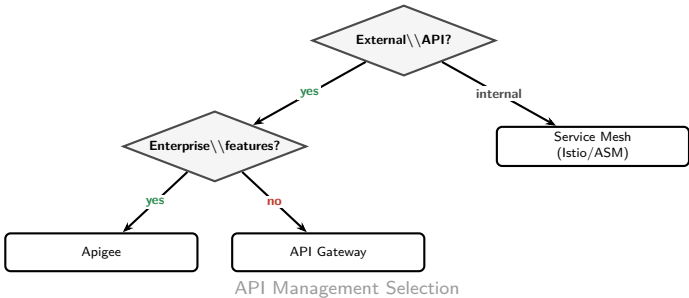
12-Factor App	Methodology for building cloud-native apps: codebase in Git, explicit dependencies, config in environment, stateless processes, port binding, disposability.
Microservices	Decompose into small, independently deployable services. Each owns its data. Communicate via APIs or events.
Event-driven	Services react to events via Pub/Sub or Eventarc. Loose coupling. Async processing. Dead-letter for failed events.
API Gateway pattern	Single entry point for clients. API Gateway or Apigee for routing, auth, rate limiting, versioning.
Strangler fig	Gradually replace legacy monolith. Route traffic to new microservice as features are migrated. Reverse proxy pattern.
CQRS	Command Query Responsibility Segregation. Separate read/write models. Write to Spanner/Firestore, read from Big-Query/Memorystore.
Saga pattern	Distributed transactions across microservices. Choreography (events) or Orchestration (Workflows). Compensating transactions for rollback.



Event-Driven Microservices

API Management

Apigee	Enterprise API management platform. API proxies, developer portal, analytics, monetization, security policies. For external APIs.
API Gateway	Lightweight API gateway for Cloud Functions and Cloud Run. OpenAPI spec-based. Auth, rate limiting, monitoring.
Cloud Endpoints	API management for gRPC and REST. OpenAPI or gRPC service config. ESP proxy for auth and monitoring.
Service mesh (Anthos)	Istio-based mesh for inter-service communication. mTLS, traffic management, observability. For GKE microservices.

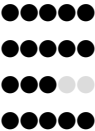


Cloud Run developer experience

Cloud Functions simplicity

App Engine legacy support

GKE flexibility



Official resources

⚠ Check the current exam guide

Google Cloud is updating its exams to the products announced at Google Cloud Next '26 (Gemini Enterprise Agent Platform, the data and analytics stack), per the certification hub checked 2026-10-05: read the current exam guide before these sheets.



Professional Cloud Developer exam guide

[services.google...english.pdf](#)

domains and topics



Professional Cloud Developer...

[cloud.google...-developer](#)

format, price, sample questions link, renewal



Professional Cloud Developer learning...

[skills.google/paths/19](#)

Google Skills (ex Cloud Skills Boost)



gcloud CLI cheat sheet

[cloud.google...cheatsheet](#)

the commands on one page

SERVICE	docs	reference	pricing	
Cloud Run	docs	reference	Pricing	3/3
Cloud Run functions	docs			1/3
Pub/Sub	docs		Pricing	2/3
Firestore	docs		Pricing	2/3
Cloud Build	docs		Pricing	2/3
Artifact Registry	docs			1/3
per kind	6/6	1/6	4/6	

11 of 18 documentation links (61% coverage) · greyed: no official page given