Google Cloud

Professional Agentic Architect

Certification exam guide

A Google Cloud Certified Professional Agentic Architect is a technical practitioner who designs and manages autonomous, AI-driven agentic workflows in Google Cloud. This individual is an experienced developer or architect who builds agentic solutions while considering reliability, performance, cost, security, and scalability. This individual has deep experience utilizing large language models (LLMs), applying agent design patterns, writing code, and integrating data sources in Google Cloud.

Section 1: Building agents using low-code tools (~13% of the exam)

1.1 Configuring agentic workflows and behavior using low-code tools. Considerations include:

- Configuring state-based workflows (pages, transition routes, and event handlers) using Gemini Enterprise tools (e.g., Gemini Enterprise Workflow Builder and Customer Experience Agent Studio [CX Agent Studio])
- Creating system instructions and in-console prompt templates (e.g., few-shot and chain-of-thought) to guide agent behavior (e.g., Workflow Builder and CX Agent Studio)

1.2 Connecting enterprise data to Gemini Enterprise. Considerations include:

- Configuring agents to securely connect and query enterprise proprietary data sources (e.g., Gemini Enterprise and Agent Search)
- Ingesting and processing unstructured multimodal data (e.g., videos, audio, and images) into the agentic workflow

Section 2: Using coding agents for application development (~17% of the exam)

2.1 Using coding agents effectively. Considerations include:

- Configuring coding agents with Model Context Protocol (MCP) servers, custom skills, and access to tools (e.g., Antigravity and Claude Code on Google Cloud)
- Using coding agents in secure sandboxes (e.g., Google Kubernetes Engine [GKE], Cloud Workstations, and Antigravity)
- Using coding agents to refactor source code, optimize execution runtimes, and patch application-layer vulnerabilities

2.2 Customizing coding agents for enterprise workflows. Considerations include:

- Creating skills, plugins, extensions hooks, rules, and subagents using Antigravity
- Augmenting Antigravity with Agents CLI to build, scale, govern, and optimize deployed agents

Section 3: Developing custom agents (~33% of the exam)

3.1 Designing and building agentic workflows in code. Considerations include:

- Selecting and configuring the appropriate language model (e.g., large language model [LLM] vs. small language model [SLM], self-hosted vs. software as a service [SaaS], and open-source software [OSS] vs. proprietary LLM) considering cost, security, and agent architecture
- Building custom agents using open-source libraries (e.g., Agent Development Kit [ADK])
- Configuring sessions and memory (e.g., Agent Platform Memory Bank and managed sessions)
- Configuring skills using Agents CLI (e.g., plugins and agent vs. human mode)

3.2 Integrating enterprise domain knowledge. Considerations include:

- Designing, configuring, and managing retrieval-augmented generation (RAG) pipelines and vector retrieval systems (e.g., embedding models, similarity scoring, and reranking) using appropriate services such as vector databases (e.g., Vector Search and Agent Retrieval)
- Configuring agent permissions (e.g., Agent Identity)
- Using Google Cloud tools (e.g., Agent Registry, Google Cloud MCP servers) to configure prebuilt and custom capabilities (e.g., custom integration layers for managed databases, API integrations, and MCP server that connects agents to third-party SaaS tools and remote servers)

3.3 Orchestrating and coordinating agentic workflows. Considerations include:

- Orchestrating agents using agentic protocols (e.g., MCP and Agent2Agent [A2A])
- Selecting and coordinating multiagent handoffs and workflows (e.g., parallel agents, sequential agents, and graph workflow) using Google Cloud tools (e.g., Agent Identity, Agent Registry, Agent Runtime, and agent policies)

Section 4: Evaluating and deploying agentic workflows (~22% of the exam)

4.1 Evaluating agents in development and in production. Considerations include:

- Creating test sets for agent evaluation (e.g., golden data, prompts, and edge cases)

- Creating continuous evaluation pipelines to assess an agent's tool execution based on established success criteria
- Determining the appropriate evaluation framework and tooling (e.g., ADK evaluation tooling (evalset), Agent Platform Gen AI evaluation service, and custom autoraters)
- Evaluating an agentic system against a golden dataset to assess agent response and retrieval quality (e.g., using ADK)

4.2 Deploying and scaling production workloads. Considerations include:

- Selecting optimal deployment runtime based on the use case, requirements, and cost (e.g., Agent Runtime, Cloud Run, and GKE)
- Troubleshooting agent issues (e.g., drift, tool invocation latency, agent reasoning loops, and system failures)
- Monitoring and optimizing agents for performance, reliability, and cost (e.g., identify logic errors, latency bottlenecks, and hallucinations)

Section 5: Securing and governing agentic workflows (~15% of the exam)

5.1 Configuring agent security and governance. Considerations include:

- Implementing authentication and secure tool execution (e.g., agent-to-tool API calls using OAuth 2.0)
- Configuring Principal Access Boundary (PAB) policies using Agent Identity
- Configuring Agent Gateway to monitor traffic and track agents
- Designing and configuring agentic governance and policy enforcement (e.g., Agent Registry and Model Armor)

5.2 Implementing secure agent behavior and execution. Considerations include:

- Designing appropriate safety frameworks and guardrails (e.g., Agent Gateway, Model Armor, and human-in-the-loop [HITL])
- Configuring secure access to data and identity propagation (e.g., Agent Gateway and Agent Registry)

The following tools are in scope for this exam:

- Agent Development Kit (ADK)
- Agent Gateway
- Agent Identity
- Agent Registry
- Agent Retrieval and Vector Search
- Agent Runtime (formerly Agent Engine)
- Agent Search (formerly Vertex AI Search)
- Agentic protocols (e.g., Agent2Agent [A2A] and MCP)
- Agents CLI in Agent Platform
- Antigravity (CLI, SDK, App)
- BigQuery
- Cloud Run
- Cloud SQL
- Cloud Storage
- Firestore
- Gemini Enterprise
- Gemini LLMs
- Gen AI evaluation service
- Google Cloud Observability (Cloud Logging and Cloud Trace)
- Google Kubernetes Engine (GKE)
- Memorystore for Redis
- Model Armor
- Model Context Protocol (MCP) servers
- Model Garden
- RAG Engine
- Sensitive Data Protection
- Skill Registry
- Workflow Builder (formerly Agent Designer)