

Professional Cloud Architect Renewal Beta

Certification exam guide

A Google Cloud Certified Professional Cloud Architect enables organizations to leverage Google Cloud technologies. Through an understanding of cloud architecture and Google technology, this individual designs, develops, and manages robust, secure, scalable, highly available, and dynamic solutions to drive business objectives. The Cloud Architect should be proficient in all aspects of enterprise cloud strategy, solution design, and architectural best practices. The Cloud Architect should also be experienced in software development methodologies and approaches including multi-tiered distributed applications which span multicloud or hybrid environments.

Case Studies

The Professional Cloud Architect Renewal Beta exam presents questions on case studies that describe a fictitious business and solution concept. The business is using Google Cloud's generative AI solutions to solve real-world challenges. The case studies are intended to provide additional context to help you choose your answers. Review these case studies which may be used in the exam.

[Cymbal Retail Case Study](#)

[Altostrat Media Case Study](#)

Section 1: Designing and planning a cloud solution architecture

1.1 Designing a solution infrastructure that meets business requirements. Considerations include:

- Business use cases and product strategy
- Business continuity plan
- Cost optimization
- Supporting the application design
- Integration patterns with external systems

Google Cloud

- Movement of data
- Design decision trade-offs
- Workload disposition strategies (e.g., build, buy, modify, or deprecate)
- Success measurements (e.g., key performance indicators [KPI], return on investment [ROI], metrics)
- Security and Compliance
- Observability

1.2 Designing a solution infrastructure that meets technical requirements. Considerations include:

- High availability and failover design
- Flexibility of cloud resources
- Scalability to meet growth requirements
- Performance and latency
- Gemini Cloud Assist
- Backup and recovery

1.3 Designing network, storage, and compute resources. Considerations include:

- Integration with on-premises/multicloud environments
- Vertex AI solutions (e.g., Gemini LLMs, Agent Builder, Model Garden, Gemini models)
- Cloud-native networking (e.g., VPC, peering, firewalls, container networking, shared VPC, PSC)
- Choosing data processing solutions
- Choosing appropriate storage types (e.g., object, file, databases)
- Mapping compute needs to platform products (e.g., Google Kubernetes Engine [GKE], Cloud Run, Cloud Run functions)
- Choosing compute resources (e.g., spot VMs, custom machine types, specialized workload)

1.4 Creating a migration plan (i.e., documents and architectural diagrams). Considerations include:

- Integrating solutions with existing systems
- Migrating systems and data to support the solution (e.g., Cloud bridge)
- Software license implications
- Network planning
- Testing and proofs of concept

Google Cloud

- Dependency management planning
- Financial impact

1.5 Envisioning future solution improvements. Considerations include:

- Cloud and technology improvements
- Evolution of business needs
- Enablement and advocacy

Section 2: Managing and provisioning a solution infrastructure

2.1 Configuring network topologies. Considerations include:

- Extending to on-premises environments (hybrid networking)
- Extending to a multicloud environment that may include Google Cloud to Google Cloud communication
- Security protection (e.g. intrusion protection, access control, firewalls)
- VPC design and load balancing (e.g., access to cloud, internet, and cloud adjacent services)

2.2 Configuring individual storage systems. Considerations include:

- Data storage allocation
- Data processing/compute provisioning
- Security and access management
- Configuration for data transfer and latency
- Data retention and data life cycle management
- Data growth planning
- Data protection (e.g., backup, recovery)

2.3 Configuring compute systems. Considerations include:

- Compute resource provisioning
- Compute volatility configuration (spot vs. standard)
- Cloud-native network configuration for compute resources (e.g., Google Compute Engine, GKE, serverless networking, VMware Engine)
- Infrastructure orchestration, resource configuration, and patch management
- Container orchestration
- Serverless computing

2.4 Monitoring, logging, profiling, and alerting solutions. Considerations include:

- Monitoring and logging
- Profiling and benchmarking
- Alerting strategies

Section 3: Designing for security and compliance

3.1 Designing for security. Considerations include:

- Identity and access management (IAM)
- Resource hierarchy (organizations, folders, projects)
- Data security (key management, encryption, secret management)
- Separation of duties (SoD)
- Security controls (e.g., auditing, VPC Service Controls, context aware access, organization policy, hierarchical firewall policy)
- Managing customer-managed encryption keys with Cloud Key Management Service
- Remote access (e.g., IAP, service account impersonation, beyondcorp, RDP, oslogin (SSH), Workload identity federation)
- Securing software supply chain
- Gemini in Security

3.2 Designing for compliance. Considerations include:

- Legislation and regulation (e.g., health record privacy, children's privacy, data privacy, ownership, data sovereignty)
- Commercial (e.g., sensitive data such as credit card information handling, personally identifiable information [PII])
- Industry certifications (e.g., SOC 2)
- Audits (including logs)

Section 4: Managing implementation

4.1 Advising development and operations teams to ensure successful deployment of the solution. Considerations include:

- Application and infrastructure deployment
- API management best practices (e.g., Apigee)
- Testing frameworks (load/unit/integration)
- Data and system migration and management tooling
- Gemini Cloud Assist

4.2 Interacting with Google Cloud programmatically. Considerations include::

- Cloud Shell Editor, Cloud Code, and Cloud Shell Terminal
- Google Cloud SDK (e.g., gcloud, gsutil and bq)
- Gemini Code Assist
- Cloud Emulators (e.g. Bigtable, Spanner, Pub/Sub, Firestore)
- Infrastructure as code (e.g., IaC)
- Accessing Google API best practices
- Google API client libraries