**Best practices**

**1. Data Strategy**

**1.1 Purpose**

Define how project data will be collected, processed, stored, secured, and disposed of in compliance with business and regulatory requirements.

**1.2 Data Classification**

| **Classification** | **Examples** | **Security Level** |
| --- | --- | --- |
| Public | Marketing material, press releases | Low |
| Internal | Internal docs, test data | Medium |
| Confidential | PII, financial data, source code | High |

**1.3 Data Lifecycle**

1. **Collection:** APIs, input forms, batch uploads.
2. **Storage:**
   * Structured: BigQuery / Cloud SQL
   * Unstructured: Cloud Storage (encrypted)
3. **Processing:** Dataflow, Dataproc, AI/ML tools.
4. **Archiving & Retention:** Retention policy based on classification.
5. **Disposal:** Secure deletion (lifecycle rules, cryptographic erase).

**1.4 Data Security**

* Encryption at rest (GCP-managed or CMEK via Cloud KMS).
* Encryption in transit (TLS 1.2+).
* IAM integration for data access.

**1.5 Backup & Recovery**

* Daily incremental backups, weekly full backups.
* Multi-region storage; quarterly DR drills.

**1.6 Compliance**

* Placeholder: Confirm client compliance requirements (GDPR, HIPAA, etc.).
* Use DLP API for sensitive data scanning.

**2. Identity and Access Management (IAM)**

**2.1 Principles**

* Apply **Principle of Least Privilege**.
* Separate human and service accounts.
* Enforce MFA for privileged accounts.

**2.2 IAM Roles**

| **Role Type** | **GCP Examples** | **Purpose** |
| --- | --- | --- |
| Viewer | roles/viewer | Read-only |
| Editor | roles/editor | Modify resources |
| Admin | roles/owner | Limited admin |
| Custom | roles/customRole | Project-specific |

**2.3 Authentication & Authorization**

* Identity Federation / SSO.
* Strong password policies (≥12 chars, complexity rules).

**2.4 Audit & Monitoring**

* Enable **Cloud Audit Logs** for admin, data, and system events.
* Enable **Security Command Center** for monitoring.

**2.5 Policy Reviews**

* Quarterly IAM review; immediate removal for offboarding.

**3. Network Security**

**3.1 Network Design**

* Separate VPCs per environment.
* Private subnets for sensitive workloads.
* Enable VPC Service Controls.

**3.2 Firewall Rules**

| **Rule** | **Source** | **Destination** | **Action** | **Notes** |
| --- | --- | --- | --- | --- |
| Allow SSH (Admin) | Admin IPs | VM Instances | Allow | Restricted access |
| Deny All Else | \* | \* | Deny | Default posture |

**3.3 Connectivity**

* Use Cloud VPN or Dedicated Interconnect.
* Private Google Access for API calls.

**3.4 Threat Protection**

* Use **Cloud Armor** for DDoS mitigation.
* Run Web Security Scanner for vulnerabilities.

**3.5 Monitoring & Incident Response**

* Cloud Logging & Monitoring with alerts.
* Maintain incident response playbook.

**4. API Strategy, Security, and Management**

**4.1 API Data Strategy**

* Apply data classification to API payloads.
* Return only required data fields.
* Encrypt API traffic (TLS 1.2+).
* Monitor API usage for anomalies.

**4.2 API IAM**

* Use OAuth 2.0, OpenID Connect, or JWT for authentication.
* Do not embed API keys in public code; apply restrictions.
* Rotate keys regularly.
* Centralize security using API Gateway, Apigee, or Endpoints.

**4.3 API Network Security**

* Protect APIs with WAF (Cloud Armor).
* Enforce throttling & rate limiting.
* Validate all requests.
* Use private networking for sensitive APIs.

**4.4 API Management & Monitoring**

* Use Apigee for full lifecycle management.
* Enable logging and metrics collection.
* Conduct regular API security testing.

**5. API Management in Google Cloud – Tool Comparison**

| **Feature / Capability** | **Apigee** | **API Gateway** | **Cloud Endpoints** |
| --- | --- | --- | --- |
| Primary Use Case | Enterprise-grade, full API lifecycle | Lightweight, serverless proxy | Simple proxy with basic security/monitoring |
| Best For | APIs as products, monetization, enterprise governance | Serverless & fast deployments | Internal/microservice APIs |
| Supported Protocols | REST, SOAP, gRPC, GraphQL | REST, gRPC | REST, gRPC |
| Security Features | OAuth 2.0, JWT, API key, WAF, quotas, threat detection | API key, JWT, IAM auth | API key, JWT, IAM auth |
| Traffic Management | Advanced (quotas, caching) | Basic | Basic |
| Analytics | Advanced analytics | Basic | Logging & metrics |
| Developer Portal | Yes | No | No |
| Deployment Options | Cloud, hybrid, on-prem | Fully managed | Fully managed |
| Complexity | High | Low–Medium | Low |
| Cost Tier | High | Low | Low |

**6. GCS (Google Cloud Storage) Implementation Guidelines**

**6.1 Data Strategy in GCS**

* **Storage Classes:** Use Standard, Nearline, Coldline, or Archive based on access frequency.
* **Lifecycle Management:** Automated transitions between classes; automated deletion after retention period.
* **Encryption:** Default Google-managed keys or CMEK via Cloud KMS.
* **Backups:** Versioning enabled; use dual-region/multi-region replication.

**6.2 IAM in GCS**

* **Uniform Bucket-Level Access:** Enforce bucket-level rather than object-level permissions.
* **Roles:**
  + Read: roles/storage.objectViewer
  + Write/Manage: roles/storage.objectAdmin
  + Avoid broad roles like Owner unless required.
* **Authentication:** MFA for privileged access; use short-lived credentials via Workload Identity Federation.
* **Audit Logging:** Enable for bucket operations; integrate with Security Command Center.
* **Periodic Reviews:** Remove stale accounts and unused permissions.

**6.3 Network Security in GCS**

* **VPC Service Controls:** Restrict bucket access to trusted networks/projects.
* **Private Google Access:** Prevent public internet routing for VM-to-GCS communication.
* **Firewall & Access Controls:** Restrict instance access paths to GCS.
* **Encryption in Transit:** Enforced by TLS 1.2+.
* **Threat Protection:** Monitor usage patterns and access logs for anomalies.

**6.4 Recovery Objectives in GCS**

* **RTO (Recovery Time Objective):** Define max downtime allowed before restoration (e.g., 4 hours).
* **RPO (Recovery Point Objective):** Define acceptable maximum data loss window (e.g., 1 hour).
* **Implementation:**
  + Use bucket replication across regions.
  + Schedule frequent snapshots/exports matching RPO goals.
  + Test restore operations against RTO goals.

**7. Client Inputs Needed**

* Data residency preferences.
* Applicable compliance regulations.
* Recovery objectives: RTO and RPO (business impact-based).
* External integration requirements.