Pulled from **system design considerations**

Zones and Regions:

Design questions

- In what geographical regions are the users for your applications?
- Which Google Cloud regions are closest to your users?
- Do you have any regulatory requirements based on geography?
- Do you need global deployment or will a regional deployment meet your requirements?

Recommendations

- Select a region or set of regions that are closest geographically to your end users to minimize latency when serving traffic to external users.
- Select a specific region or set of regions to meet any geographic requirements
- Use a Load Balancer to provide a single IP which is routed to your application when you are serving a global user base.
- Connect your on-premises or colocation networks to Google Cloud through Cloud Interconnect for high-speed, private network connections.

Resource Management:

Design questions

- Which roles in your organization require access to your Google Cloud infrastructure?
- What access requirements do members of each role have for Google Cloud resources?
- How will your organizational structure map to the Google Cloud resource hierarchy?
- Do you have governance conventions for resource labeling?

Recommendations

- Create an organization node in your domain.
- Define a resource hierarchy that maps to your Google Cloud business needs.
- Define your project structure. For example:
    - Anonymize information in project names.
    - Follow a project naming pattern like *{company-initial-identifier}-{environment}-{app-name}*, where the placeholders are unique but don't reveal company or application names.
- Automate project creation, delegate billing, and set up IAM governance.
- Prevent accidental deletion by leveraging project liens.
- Identify and plan for zonal, regional, and multi-regional deployment for your workloads.

Identity and Access Management:

Design questions

- How will you manage identities?
- Will you federate from an existing identity source?
- How do you plan to delegate admin access?

- Do you have a governance process to create, update, and audit access control?
- Do you group users and enforce multi factor authentication (MFA) based on access sensitivity?

Recommendations

- Secure organization admin access.
- Federate your identity provider with Google Cloud.
- Use Cloud Identity for user account identity if you don't have an identity provider.
- Use Google Accounts and appropriate IAM policies for every user.
- Create your own custom service accounts to limit IAM permissions to least privilege.
- Migrate unmanaged accounts.
- Secure access to resources through least privilege.
- Use groups and service accounts.
- Use a group naming convention.
- Audit the group membership request workflow.
- Enforce MFA whenever possible, especially for users with high privilege access.
- Review super admin access.
- Leverage service account and audit access and how it is used.
- Remove default IAM Organization policies.
- Audit access management changes regularly.

Compute Resources:

Design questions

- How are you planning to use compute?
- Are your applications containerized or do they have any legacy dependency?
- Is your application Stateful or Stateless?
- Do you have complex distributed service deployment (high inter-node networking)?
- How do you manage instance access (including SSH keys)?

Recommendations

- Choose the Google Cloud region closest to your user base or depending on compliance requirements.
- Evaluate latency requirements for your workloads.
- Determine application-end user latency requirements and choose a single region or multi-region deployment strategy.
- Ensure that instances are not configured to use the default service account with full access to all Cloud APIs.
- Ensure that IP forwarding is not enabled on instances unless needed.
- Ensure that Compute Engine instances do not have public IP addresses when not needed. Instead use NAT Gateway.

Networking:

Design questions

- How complex is your application service connectivity deployment?
- What are some networking requirements for your inter-application deployments?

- If you have external services, how will you connect to the Google Cloud network?
- If connecting your VPC and on-premises network, how much bandwidth do you require?
- How do you segment and access control your network? Based on applications? Teams?
- Do you have a governance process to create or update new or existing networking deployments? How frequently do you audit?
- Do you have a separate network for sensitive applications? How do you monitor and restrict access?

Recommendations

- Document your network design: Cross projects or hybrid deployments. Use a network topology graph to verify connectivity.
- Use clear and consistent naming conventions for services like service accounts, network tags, and firewall rules.
- Grant the network user role at the subnet level.
- Choose an appropriate project:
    - Use a single host project if resources require multiple network interfaces.
    - Create a single VPC per project to map VPC quotas to projects.
    - Use multiple host projects if resource requirements exceed the quota of a single project.
    - Use multiple host projects if you need separate administration policies for each VPC.
- Create a VPC for each autonomous team, with shared services in a common VPC.
- Isolate sensitive data in its own VPC or project.
- While using VPC Network Peering, evaluate if you won't exceed network peering quota limits (forwarding rules, firewall rules, routes, and so on).
- Use multi-NIC virtual appliances to control traffic between VPCs through a cloud device.
- Use VPC for administration of multiple working groups.
- Create a shared services VPC if multiple VPCs need access to common resources but not to each other.
- Use dynamic routing whenever possible.
- Centralize network control.
- Use Private DNS zones for name resolution whenever possible.
- Frequently audit network access permissions and control.
- Ensure SSH/RDP access is restricted from the internet.
- Enable VPC flow logs for critical projects.

Storage:

Design questions

- How much and what types of storage do you require?
- What are the access modes for your requirements?
- Do you need active or archival storage?
- Are you looking to host static objects for web hosting? CDN?
- Do you store and process sensitive data? How do you monitor and manage access?
- Do you have process and governance requirements for encryption?

Recommendations

- Determine application storage requirements and choose appropriate storage options.

- Make every bucket name unique across the entire Cloud Storage namespace. Do not include sensitive information in a bucket name. Choose bucket and object names that are difficult to guess.
- Do a back-of-the-envelope estimate of the amount of traffic that will be sent to Cloud Storage in order to calculate transfer time.
- If you are hosting public content, try using CDN to minimize egress cost.
- Store your data in a region closest to your application's users.
- Keep compliance requirements in mind when choosing a location for user data.
- For data that will be served at a high rate with high availability, use the Multi-Regional Storage or Regional Storage class. For data that will be infrequently accessed and can tolerate slightly lower availability, use the Nearline Storage or Coldline Storage class.
- Ensure that your Cloud Storage bucket is not anonymously or publicly accessible.

Database:

Design questions

- What databases are you running? How are they used ?
- Do you have any specific requirements (latency, replication, consistency)?
- Do you have any legacy dependency on certain databases or versions?
- How many of these are structured or unstructured?
- How do you govern access to your database? At the application level and for internal consumption?

Recommendations

- Choose the right schema for your table.
- Choose the right key name to avoid key hotspotting, especially for non-relational databases.
- Shard your database instance whenever possible.
- Use good connection management practices, such as connection pooling and exponential backoff.
- Avoid very large transactions.
- Design and test your application's response to maintenance updates on databases.
- Secure and isolate connections to your database.

Analytics:

Design questions

- How do you ingest and analyze your data?
- Do you currently have an ETL pipeline setup? What does it look like?
- What type of data do you typically analyze? Any proprietary data formats?
- Do you have an estimate of your existing data and expected growth?
- Do you perform any machine learning? Do you plan to use a managed or unmanaged service?
- Do you have SLAs for jobs or workflows? How do you monitor them?

Recommendations

- Determine whether your application needs an "exactly once" or "guaranteed once" delivery pipeline.
- Decouple your ETL functions into small functions using Pub/Sub as a buffer to make the pipeline scalable.
- Use the Jobs API to scale Dataproc clusters, which helps reduce cost by running jobs in existing clusters.
- Evaluate your query performance and partition your BigQuery datasets to minimize query cost.