Designing and Implementing

Google Cloud Training and Google Cloud

Designing and planning a cloud solution architecture

Easy as … cake

TIP

You will want to have key knowledge and skills at every layer

Divide and Iterate

Designing a solution infrastructure that meets business requirements

Exam outline	Tips
Business use cases and product strategy	What does success look like qualitatively?
Cost optimization	A separate step after business logic criteria are met
Supporting the application design	What procedures and processes will be needed?
Integration	Working with existing systems in parallel at first? Are there parts that must remain on premises?
Movement of data	How and when will the data migrate to the new solution?
Tradeoffs	Priorities. Timing can be critical to value.
Build, buy, or modify	Control vs. overhead vs. time
Success measurements	What does success look like quantitatively?

All Solutions are Contextual Solutions

TIP

Even the best design patterns won't work in every situation.
You have to consider the limits and possible issues in the context of use.

Build, Buy, or Modify

TIP

What are the business priorities?

Cloud Architect Case Study 01: Design and Plan

A customer had this interesting business requirement...

Provisioning VMs within their own data center took weeks

Wanted to empower their IT teams to develop new applications in the Cloud with greater autonomy and more quickly. (hybrid Cloud strategy)

We mapped that to technical requirements like this...

A turn-key GCP project for any developer team that requested it

Manually provisioned within 24 hours, following brief review of request

And this is how we implemented that technical requirement.

User identity via Google Cloud Directory Sync (GCDS) from on-premise

Network connectivity between colo and GCP via two VPN tunnels

Organization node, folders by department and use case, project by environment

IAM roles commonly Project Owner, not Project Creator

Designing a solution infrastructure that meets technical requirements

Exam outline	Tips
High availability and failover design	What are the real criteria? What are the measurable goals?
Elasticity of cloud resources	How do you want the solution to behave when busy? How will the system behave when under attack? How will it behave when traffic diminishes?
Scalability to meet growth requirements	Scaling for growth is different from autoscaling to cover temporary demand non-linearities. When is it time to add a node, upgrade a service, or switch services?

TIP

If a requirement is not fully described -- perhaps it is not important to the question being asked.

Scaling, elasticity, and failover -- what will you measure?

1. Begin simply. Iterate.

Compose simple services.
Build **up** to complexity.
Iterate.

2. Plan for failure.

Avoid bottlenecks.
Avoid single points of failure.

TIP

3. Measure stuff.

Consider the situation:
What are the limits?
How might the system fail?

TIP

Business Requirements -- by when?

All business values are real-time values.

Stateful vs. stateless: job shop and assembly line

TIP

Identifying where state information is located in a design will help you analyze scaling characteristics

General Cloud Design Principles: Example

Use DNS and connection load balancing to get requests from users

Many small stateless servers

Some sharded stateful servers

Receive request at frontend servers
Take control of the client request
Initiate server-to-server

This design allows you to control:

- Server discovery
- Request balancing
- Throttling

Designing Network, Storage, and Compute Resources

Exam outline	Tips
Integration with on premises/multi-cloud environments	When to use gsutil, gsutil rsync, and Storage Transfer Service
Cloud native networking (VPC, peering, firewalls, container networking)	Understand all the networking services and how to connect for throughput, security, billing, and so forth.
Identification of data processing pipeline matching data characteristics to storage systems	Velocity (how frequent), Volume (how much), Variety (format, structure), Volatility (how often does it change) -- which services match?
Data flow diagrams	Very helpful to know where the data will travel through the solution and to look for bottlenecks and flow monitoring and control
Storage system structure (e.g., Object, File, RDBMS, NoSQL, NewSQL)	ACID—What is eventual consistency ? BASE. Structure requirements?
Mapping compute needs to platform products	Scale, capacity, control/overhead?

Which Network, Compute, Storage Resources are Critical?

TIP

Identify the bottlenecks

Data Flow Diagrams

What transaction qualities are required?

CAP Theorem

ACID

- Atomicity
- **Consistency**
- Isolation
- Durability

BASE

- Basically **Available**
- Soft state
- Eventual consistency

Creating a Migration Plan

Exam outline	Tips
Integrating solution with existing systems	Parallel?
Migrating systems and data to support the solution	How much, how to synchronize in both directions? Where is the source of data truth?
Licensing mapping	Data center license to cloud license -- may have different license or requirements for licensed software.
Network and management planning	How will on premises connect to the cloud solution?
Testing and proof of concept	Different ways to divide a test audience: random, group, phased, test.

TIP

Where is the source of truth?
How will actions be synchronized?

Think outside the Technical/Theoretical box

Planning: Technical Solution

TIP

Doing: Practical Considerations

Envisioning Future Solution Improvements

Exam outline	Tips
Cloud and technology improvements	When the technology improves, how will the solution embrace the improvements?
Business needs evolution	When the business changes, how will the solution change? Most solutions don't just need to work once, they need to become self-sustaining.
Evangelism and advocacy	Who are the "gatekeepers" and what do they need to know to effectively do their jobs?

How will the solution grow and change once implemented?

Operations

TIP

- Monitoring
- Logging
- Maintenance
- Procedures
- Human Processes

Figure out what to measure

Application parts developed by separate project teams will communicate over RFC1918 addresses.

A. Single project, same VPC
B. Shared VPC, each project a service of the Shared VPC project
C. Parts communicate using HTTPS
D. Communicate over global load balancers, one per project

Application parts developed by separate project teams will communicate over RFC1918 addresses.

A. Single project, same VPC
B. Shared VPC, each project a service of the Shared VPC project ✔
C. Parts communicate using HTTPS
D. Communicate over global load balancers, one per project

Solution

B - Each team has their own project but communicates securely over a single RFC1918 address space.

A - No separation.

C - Doesn't specify separate projects, therefore doesn't meet business requirements.

D - External IPs do not conform to address technical requirements.

https://cloud.google.com/vpc/docs/shared-vpc
https://cloud.google.com/vpc/docs/vpc-peering

Which solution should Dress4Win's real-time trend analysis use in the cloud?

A. Cloud Dataflow
B. BigQuery
C. A Hadoop cluster in Compute Engine
D. Cloud Dataproc

Which solution should Dress4Win's real-time trend analysis use in the cloud?

A. Cloud Dataflow
B. BigQuery
C. A Hadoop cluster in Compute Engine
D. Cloud Dataproc ✔

Solution

D - A managed service that can run the Hadoop application.

Real-time trend analysis ... refer to the case study to learn this is Hadoop/Spark.

A, B - Dataflow and BigQuery are both "serverless services", not managed services, and therefore won't meet the technical requirement to "Use managed services whenever possible."

C - A Hadoop cluster on Compute Engine is not a managed service.

https://cloud.google.com/dataproc/docs/
https://cloud.google.com/solutions/migration/hadoop/hadoop-gcp-migration-overview
https://cloud.google.com/solutions/migration/hadoop/hadoop-gcp-migration-jobs
https://cloud.google.com/solutions/migration/hadoop/hadoop-gcp-migration-data

Managing and provisioning solution infrastructure

Google Cloud Training and Certification

Configuring Network Topologies

Exam outline	Tips
Extending to on-premises (hybrid networking)	VPN. Interconnect.
Extending to a multi-cloud environment	Interoperation interface. Stackdriver (GCP + AWS)
Security	*Discussed in "Designing for security and compliance"*
Data protection	Encryption, access, CSEK, KMS

Network

Internet
ISP
Legacy DC
DMZ
Aggregation Layer
Access Layer
Access Layer
ToR
ToR
ToR
Web Servers
Application Servers
Database Servers
Internet
Google Cloud Platform
Single Global VIP
Web Servers
Compute Engine
BigQuery
App Servers
Compute Engine
Database Storage
Cloud Virtual Network

Subnets extend across zones

1. Subnetworks can extend across zones in the same region.
2. One VM and an alternate VM can be on the same subnet but in different zones.
3. A single firewall rule can apply to both VMs even though they are in different zones.

There are five ways to interconnect

	Dedicated	Shared
Layer 3	**Direct Peering (no SLA)** **VPN**	**Carrier Peering (no SLA)**
Layer 2	**Dedicated Interconnect**	**Partner Interconnect**

https://cloud.google.com/interconnect/

Protect data by practicing defense in depth

Edge protections

VM access/IP addresses
API access/endpoints

Network protections

Firewall is the first line of defense

Infrastructure protections

Isolation and controlled sharing of resources
Server-side encryption and key management

Cloud Architect Case Study 02: Provision and Manage

A customer had this interesting business requirement...

Cloud infrastructure resource provisioning must be documented and auditable

Changes to the cloud infrastructure must go through a review process `

Specific: Must minimize impact to developer velocity

We mapped that to technical requirements like this...

Provisioning infrastructure should be done as Infrastructure as Code

Specific: Decentralize approval/code review process. Teams should own process.

And this is how we implemented that technical requirement.

Terraform/Deployment Manager

IAM Strategy - Least Privilege

Service Accounts

Specific: Utilize source control options. E.g. Github Enterprise

Configuring individual Storage Systems

Exam outline	Tips
Data storage allocation	Where will the data be located? Resiliency? Charges?
Data processing/compute provisioning	Which platform? What capacity?
Security and access management	CSEK and Cloud Storage, for example. *Discussed in "Designing for security and compliance"*
Network configuration for data transfer and latency	Location makes a difference in egress charges and round-trip time.
Data retention and data lifecycle management	What is the Nearline and Coldline policy? When to delete data?
Data growth management	When do you need a different way to organize the data? How much/how big will the current design support?

Comparing Storage and Database

In memory	Relational		Non-relational		Object	Warehouse
App Engine Memcache	Cloud SQL	Cloud Spanner	Cloud Datastore	Cloud Bigtable	Cloud Storage	BigQuery
Good for: Web/mobile apps, gaming	**Good for:** Web frameworks	**Good for:** RDBMS+scale, HA, HTAP	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events	**Good for:** Binary or object data	**Good for:** Enterprise data warehouse
Such as: Game state, user sessions	**Such as:** CMS, eCommerce	**Such as:** User metadata, Ad/Fin/MarTech	**Such as:** User profiles, Game State	**Such as:** AdTech, financial, IoT	**Such as:** Images, media serving, backups	**Such as:** Analytics, dashboards

Comparing Storage options: Technical details

	Cloud Datastore	Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Transactions	Yes	Single-row	No	Yes	Yes	No
Complex queries	No	No	No	Yes	Yes	Yes
Capacity	Terabytes+	Petabytes+	Petabytes+	10,230 GB	Petabytes	Petabytes+
Unit size	1 MB/entity	~10 MB/cell ~100 MB/row	5 TB/object	Determined by DB engine	10,240 MiB/ row	10 MB/row

Comparing Storage options: Use cases

	Cloud Datastore	Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Best for	Getting started, App Engine applications	“Flat” data, Heavy read/write, events, analytical data	Structured and unstructured binary or object data	Web frameworks, existing applications	Large-scale database applications (> ~2 TB)	Interactive querying, offline analytics
Use cases	Getting started, App Engine applications	AdTech, Financial and IoT data	Images, large media files, backups	User credentials, customer orders	Whenever high I/O, global consistency is needed	Data warehousing

Selecting Storage options

Firebase...

Firebase

Cloud Datastore

Ingesting data into your service

These are the key components for ingesting data into GCP and the features they contribute to data ingestion.

- Global access
- High bandwidth
- Low latency

RESTful API Exposure for client apps

- HTTP Post
- Signed URLs
- RESTful APIs

- Authentication
- Ingest coordination
- Web UI for browser users
- API for client apps

The goal of backup isn't archive, it is restore

Lazy deletion

Users care about **Data Integrity**

Data Integrity = Accuracy + Accessibility

Many kinds of error, including human error, can lead to data loss.

How long does data transfer take?

Physical Speeds

Online Speeds

Quantity

Data Transfer Appliance

	1 Mbps	10 Mbps	100 Mbps	1 Gbps	10 Gbps	100 Gbps
1 GB	3 hrs	18 mins	2 mins	11 secs	1 sec	0.1 sec
10 GB	30 hrs	3 hrs	18 mins	2 mins	11 secs	1 sec
100 GB	12 days	30 hrs	3 hrs	18 mins	2 mins	11 secs
1 TB	124 days	12 days	30 hrs	3 hrs	18 mins	2 mins
10 TB	3 years	124 days	12 days	30 hrs	3 hrs	18 mins
100 TB	34 years	3 years	124 days	12 days	30 hrs	3 hrs
1 PB	340 years	34 years	3 years	124 days	12 days	30 hrs
10 PB	3,404 years	340 years	34 years	3 years	124 days	12 days
100 PB	34,048 years	3,404 years	340 years	34 years	3 years	124 days

Cloud Storage is extremely versatile

Backup and archive

Replace/decommission infrastructure

Content storage and delivery

TIP

Bucket classes, access control, object versioning, encryption keys, object lifecycle

How does Cloud Storage work?

Single global namespace simplifies locating buckets and objects

Location to control latency

Durability and availability

Long object names simulate structure

Distinguish frequency of access and location

Cloud Storage simulates a file system

Long object name
simulates file structure

File access `gs://declass/de/modules/02/script.sh`

Web access `https://storage.cloud.google.com/declass/de/modules/02/script.sh`

Choosing among Google Cloud Storage classes

	Multi-regional	Regional	Nearline	Coldline
Intended for data that is...	Most frequently accessed	Accessed frequently within a region	Accessed less than once a month	Accessed less than once a year
Availability SLA	99.95%	99.90%	99.00%	99.00%
Access APIs	***Consistent APIs***			
Access time	***Millisecond access***			
Storage price	Price per GB stored per month			
Retrieval price	Total price per GB transferred			
Use cases	Content storage and delivery	In-region analytics, transcoding	Long-tail content, backups	Archiving, disaster recovery

How does Cloud Bigtable work?

Cloud Bigtable design idea is "simplify for speed"

The Row Key is the index.

And you get only one.

But speed depends on your data and Row Key

Scan | Sort | Search

Flights of the world: Reviewing the data

Make

Model

Age

Origin

Departure

Capacity

Passengers

Destination

Arrival

Flights of the world: Transforming the data

Format of data as it arrives:

What is the best Row Key?

Query: All flights originating in Atlanta and arriving between March 21st and 29th

Cloud Bigtable schema organization

Column Families

Row Key	Flight_Information					Aircraft_Information			
	Origin	Destination	Departure	Arrival	Passengers	Capacity	Make	Model	Age
ATL#arrival#20190321-1121	ATL	LON	20190321-0311	20190321-1121	158	162	B	737	18
ATL#arrival#20190321-1201	ATL	MEX	20190321-0821	20190321-1201	187	189	B	737	8
ATL#arrival#20190321-1716	ATL	YVR	20190321-1014	20190321-1716	201	259	B	757	23

When to use something else...

When should you choose disk storage?

When should you choose mobile storage?

DISK solutions

Persistent HDD Disk

Persistent SSD Disk

Local SSD Disk

RAM Disk

MOBILE solutions

Cloud Storage for Firebase

Firebase Realtime DB

Firebase Hosting

Evolutionary path of data processing and ML

Data Center Migration

Machine Learning Processing

Pre-built Artificial Intelligence

Cloud AutoML builds custom AI models

Cloud AutoML
Natural Language

Cloud AutoML
Translation

Cloud AutoML Vision

Artificial Intelligence architectural strategy

Use pre-built AI

Add Custom Models

Create new Models

BigQuery ML

1 Dataset

2 Create/train

```
CREATE MODEL `bqml_tutorial.sample_model`
OPTIONS(model_type='logistic_reg') AS
SELECT
```

3 Evaluate

```
FROM
  ML.EVALUATE(MODEL
`bqml_tutorial.sample_model`,
    TABLE eval_table)
```

4 Predict/classify

```
FROM
    ML.PREDICT(MODEL
`bqml_tutorial.sample_model`,
      table game_to_predict) )
AS predict
```

Configuring Compute Systems

Exam outline	Tips
Compute system provisioning	Speed, throughput, capacity, burstiness? Turn off when not in use?
Compute volatility configuration (preemptible vs. standard)	Stateful vs. stateless Tolerance for lost data/state
Network configuration for compute nodes	Firewall rules, load balancing, NAT, VPN
Orchestration technology configuration (e.g. Chef/Puppet/Ansible/Terraform)	Deployment Manager … others The value of self-documenting and automated orchestration of infrastructure
Container orchestration (e.g. Kubernetes)	The value of container-based development; portability, scalability, fault isolation, platform agnostic deployment, CI/CD development methods

Selecting Compute Options

Application Platforms

Compute Engine	Kubernetes Engine	App Engine	Cloud Functions
Processing Flexibility	**Platform Independence**	**Code first**	**Microservices**

TIP

Practice deciding which platform to use and know exactly why you would make that choice. It is an important infrastructure skill.

Comparing Compute options: Technical details

	Compute Engine	Kubernetes Engine	App Engine Flex	App Engine Standard	Cloud Functions
Service model	IaaS	Hybrid	PaaS	PaaS	Serverless
Languages supported	Any	Any	Java, Python, Node.js, Ruby, PHP, .NET Core, Go; or supply a runtime	Java, Go, Python, PHP	Node.js
Use cases	General computing workloads	Container-based workloads	Web and mobile applications; container-based workloads	Web and mobile applications	Ephemeral functions responding to events

Choosing load balancing for Compute Engine

Global versus regional load balancing

External versus internal load balancing

Traffic type

Choosing instance groups for Compute Engine

Type of Instance Group		Properties of Instances	Feature
Unmanaged		Heterogeneous	
Managed		Homogeneous	Instance Templates Autoscaling
	Zonal	Same zone	Latency consistency
	Regional	Different zones	Reliability

Microservices: Stateless design complicates logic

TIP

Microservices are not a panacea. They are not the best solution in every case.

Make sure you know where the edge cases are between technologies.

Triggering a Cloud Function

Example of using Cloud Functions for data ingress

Containers and Google Kubernetes Engine

Docker builds containers

Enterprise and Continuous Deployment

What is really in a container...

Dockerfile

```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

docker commands

```
$> docker build -t py-web-server .
$> docker run -d py-web-server
$> docker images
$> docker ps
$> docker logs <container id>
$> docker stop py-web-server
```

Where can you run containers?

Kubernetes cluster has nodes, pods, and containers

Kubernetes jobs run containers on nodes

Advanced operations: A/B testing, Rolling updates

Balance resiliency, scalability, cost

Small stateless servers increase reliability and scalability	**Large stateful servers reduce complexity and latency**
Divide into parts	Unify
Duplicate and coordinate	Simplify and consolidate
Separate and isolate	Coalesce and colocate

Methods of achieving balance in your design

- What are your SLOs? What do your users value?
- What is the optimal size and number of parts?
- Sometimes central control is necessary/optimal
- Plan on adjusting, and build adjustment processes

TIP

Make sure you know the business goals of the exam question, and not just the technical requirements.

Services that combine data and compute

BigQuery

Cloud Dataproc

Cloud Dataflow

Cloud ML Engine

TIP

Distinguish between data processing and computing cases. Sometimes the solution is easier than it appears.

Serverless Services and Managed Services

BigQuery is a Cloud Data Warehouse service

Cloud Dataproc does processing and transformation

Example: Internet of things (IoT)

TIP

Do you know how each of these services contribute to the IoT solution?

Cloud IoT Core

Which network feature could help a company meet its goals to expand service to Asia while reducing latency?

A. HTTP/TCP load balancing
B. Network TCP/UDP
C. Cloud Router
D. Cloud Content Delivery Network (CDN)

Which network feature could help a company meet its goals to expand service to Asia while reducing latency?

A. HTTP/TCP load balancing
B. Network TCP/UDP
C. Cloud Router
D. Cloud Content Delivery Network (CDN) ✔

Solution

D - CDN will enable a company to expand its online presence with a single IP address and global reach leveraging Google's global network

A - Global Load Balancing is part of CDN. It is roughly half, the other half being global cache services. Load balancing would help the client reach its customers in Asia, but it won't necessarily help reduce latency, whereas CDN cache will do that.

B - Network load balancing is internal only. It can help scale the service but not necessarily expand its reach.

C - Cloud Router discovers changed network topology using BGP. That might be valuable if they open offices in Asia, but that doesn't directly expand the services.

https://cloud.google.com/cdn/docs/
https://cloud.google.com/cdn/docs/overview
https://cloud.google.com/cdn/docs/caching

How can you minimize the cost of storing security video files that are processed repeatedly for 30 days?

A. Regional class, then move to Coldline after 30 days.
B. Nearline class, then move to Coldline after 30 days.
C. Regional class, then move to Nearline after 30 days.
D. Multi-regional class, then move to Coldline after 30 days.

How can you minimize the cost of storing security video files that are processed repeatedly for 30 days?

A. Regional class, then move to Coldline after 30 days. ✔
B. Nearline class, then move to Coldline after 30 days.
C. Regional class, then move to Nearline after 30 days.
D. Multi-regional class, then move to Coldline after 30 days.

Solution

A - Local usage, then coldline because it is unlikely to be read.

B - Nearline won't be cost effective because the data is accessed too frequently.

C - Nearline won't be cost effective because the data will not be accessed that frequently.

D - Multi-regional is more expensive than Regional, and not needed.

https://cloud.google.com/storage/docs/storage-classes
https://cloud.google.com/storage/docs/storage-classes#nearline
https://cloud.google.com/storage/docs/storage-classes#coldline

:45

Challenge Lab 01

PCA Prep - Google Cloud Essential Skills: Challenge Lab

Google Cloud Training and Google Cloud