Google Certified Professional - Cloud Architect - Part 3

Architect Exam Preview

Exam Overview

50 questions

2 hours

All questions multiple choice/multiple answer

Roughly one third of questions tied to two case studies

Exam Surprises Many Students

- "Mile wide, inch deep" – VERY broad range of topics.
- Mix of low-level (technical) and high-level (conceptual) topics.
- Many test takers are surprised by the wide range of topics.
- Familiarity with technical topics helps prepare for conceptual questions.
- Must know what GCP services to use for business requirements.
- Official exam guide is a bit abstract:

  This course will break down the exam guide into practical pieces

Official Practice Exam

- Very accurate representation of the exam format.
- We will cover it at the end of the course.

Google Certified Professional - Cloud Architect - Part 3

Case Studies

Case Study Format

- Three available case studies – expect to see all three on the exam
- All exam case studies are available from Google's training site
- Substantial portion of the exam will be from case studies
- Questions on one side, case study on other side

Themes

- Each case study has primary topic, or theme.
- Studying case studies in advance = valuable study tool.
- We will break down each case study in the following lessons.

Layout – Each Section is Important

- Company Overview
- Solution Concept – current goal
- Existing Technical Environment (if applicable)– where they are now
- Requirements (technical/business) – boundaries and measures of success
- Executive statement– what management cares about
    - Side note – bridging the business side with technical side is key role of Cloud Architect

MountKirk Games Case Study

Business Requirements

- Increase to a global footprint
  - Multiple regional instance group backend
    - Served by single global HTTP load balancer
  - Multi-regional storage/processing options
    - Pub/Sub, Datastore, BigQuery, Cloud Storage
  - Cross-zone redudancy/replication
    - Dataflow
- Improve uptime - downtime is loss of players
  - Redudancy across zones (all of the above) or regions
- Increase efficiency of the cloud resources we use
  - Scaling infrastructure (all of the above)
  - Monitor with Stackdriver
- Reduce latency to all customers
  - Multi-regional GCE backends (served by HTTP load balancer), Multi-region Datastore

Technical Requirements

Requirements for Game Backend Platform

- Dynamically scale up or down based on game activity.
  - autoscaling managed instance groups
- Connect to a transactional database service to manage user profiles and game state.
  - Cloud Datastore - NoSQL transactional database - perfect for game user profiles and game states
- Store game activity in a timeseries database service for future analysis.
  - Game activity from servers
  - Store in BigQuery
  - BigQuery vs. Bigtable?
    - Bigtable = millisecond response time, not cross-region
    - BigQuery = response measured in seconds, cross region, scales more efficiently
- As the system scales, ensure that data is not lost due to processing backlogs.
  - Cloud Dataflow - autoscaling data processing pipeline
- Run hardened Linux distro.
  - Managed instance groups - custom images

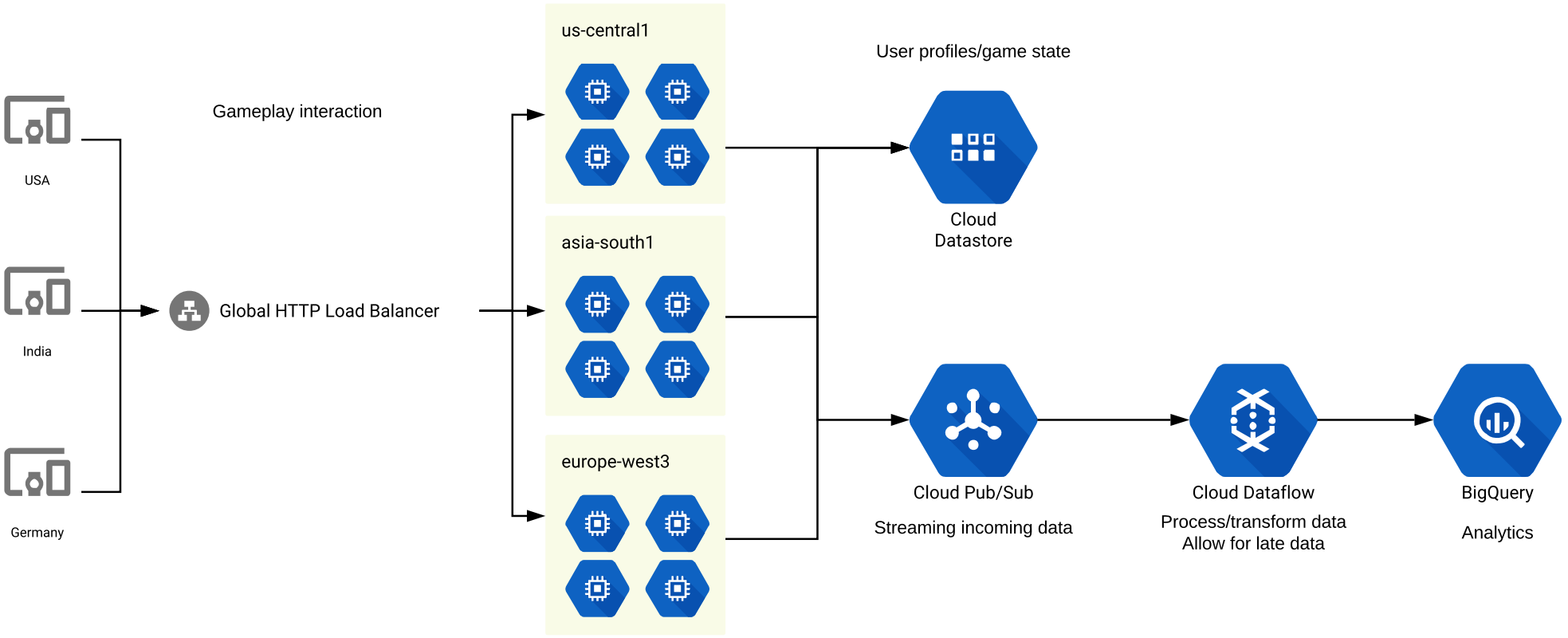
Continued next page

Requirements for Game Analytics Platform

- Dynamically scale up or down based on game activity.
  - Autoscaling services, everything below
- Process incoming data on the fly directly from the game servers.
  - Connect services with Pub/Sub, process with Dataflow
- Process data that arrives late because of slow mobile networks.
  - Dataflow accounts for late/out of order data
- Allow queries to access at least 10 TB of historical data.
  - BigQuery
- Process files that are regularly uploaded by users' mobile devices.
  - Upload to storage (Cloud Storage),
  - Process via Dataflow

Choosing a storage option: https://cloud.google.com/storage-options/

Building a mobile games analytics platform:
https://cloud.google.com/solutions/mobile/mobile-gaming-analysis-telemetry


USA
India
Germany
Gameplay interaction
Global HTTP Load Balancer
us-central1
asia-south1
europe-west3
User profiles/game state
Cloud
Datastore
Cloud Pub/Sub
Streaming incoming data
Cloud Dataflow
Process/transform data
Allow for late data
BigQuery
Analytics

Dress4Win Case Study

Executive priorities:

- Scale
- Contain costs - improve TCO
- Too many resources sitting idle

Solution Concept:

- Moving development and test environments to Google Cloud
  - Use separate projects for different environments
- DR site
  - Hybrid cloud/on-premises environment
  - Connect over VPN

Business Requirements

- Build a reliable and reproducible environment with scaled parity of production.
  - As much as possible, create equivalent setup on cloud without having to re-engineer existing applications
- Improve security by defining and adhering to a set of security and Identity and Access Management (IAM) best practices for cloud.
  - Principle of least privilege
  - Separate test/development environments
- Improve business agility and speed of innovation through rapid provisioning of new resources.
  - Automate infrastructure creation
    - gcloud/Google Cloud SDK
    - Rapid deployment (deployment manager, etc)
- Analyze and optimize architecture for performance in the cloud.
  - Stackdriver
    - Monitor infrastructure with Stackdriver Monitoring
    - Notified of errors with Stackdriver Logging
    - Troubleshoot errors with Stackdriver Debug/Error Reporting

Technical Requirements

- Easily create non-production environments in the cloud.
  - Best practices for migration
  - Move data first, then applications
  - More detail later in this course
- Implement an automation framework for provisioning resources in cloud.
  - gcloud for automated management (scripts)
    - Deployment manager
    - Other infrastructure as code products
- Implement a continuous deployment process for deploying applications to the on-premises datacenter or cloud.
  - Discussed further in this course
  - CI/CD pipeline, Jenkins, etc
- Support failover of the production environment to cloud during an emergency.
  - Replicating environment on Google Cloud
    - MySQL replicating to Cloud SQL
    - On-premises/cloud application servers - DNS cutover
- Encrypt data on the wire and at rest.
  - All data encrypted by default
  - Customer supplied (custom) encryption
- Support multiple private connections between the production data center and cloud environment.
  - VPN

Existing Technical Environment

The Dress4Win application is served out of a single data center location. All servers run Ubuntu LTS v16.04.

Databases:

- MySQL. 1 server for user data, inventory, static data,

MySQL 5.8
8 core CPUs
128 GB of RAM
2x 5 TB HDD (RAID 1)

    - Cloud SQL
        - Native MySQL support
        - 10TB size limit
        - Single region - no global footprint requirement
    - Migration - create replica server managed by Cloud SQL
        - Once replica is synced:
            - Update applications to point to replica
            - Promote replica to stand-alone instance

Redis 3 server cluster for metadata, social graph, caching. Each server is:

- Redis 3.2
- 4 core CPUs
- 32GB of RAM
    - Two options:
        - Run Redis server on Compute Engine
        - Use new Memorystore managed Redis database

Compute: 40 Web Application servers providing micro-services based APIs and static content.

- Tomcat - Java
- Nginx
- 4 core CPUs
- 32 GB of RAM

    - Existing environment has lots of idle time
    - Managed instance groups - autoscaling
    - Use custom machine types

20 Apache Hadoop/Spark servers:

- Data analysis
- Real-time trending calculations
- 8 core CPUs
- 128 GB of RAM
- 4x 5 TB HDD (RAID 1)
    - Use Cloud Dataproc

3 RabbitMQ servers for messaging, social notifications, and events:

- 8 core CPUs
- 32GB of RAM
    - Pub/Sub likely replacement
    - Can also deploy same environment on Compute Engine instance group

- Miscellaneous servers:

Jenkins, monitoring, bastion hosts, security scanners
8 core CPUs
32GB of RAM

Storage appliances:

- iSCSI for VM hosts
- Fiber channel SAN - MySQL databases
- 1 PB total storage; 400 TB available
    - SAN/iSCSI requires block storage
    - Persistent disks working in SAN cluster

- NAS - image storage, logs, backups
    - Cloud Storage will be direct replacement
    - Infinite scalability in a single bucket

100 TB total storage; 35 TB available

TerramEarth Case Study

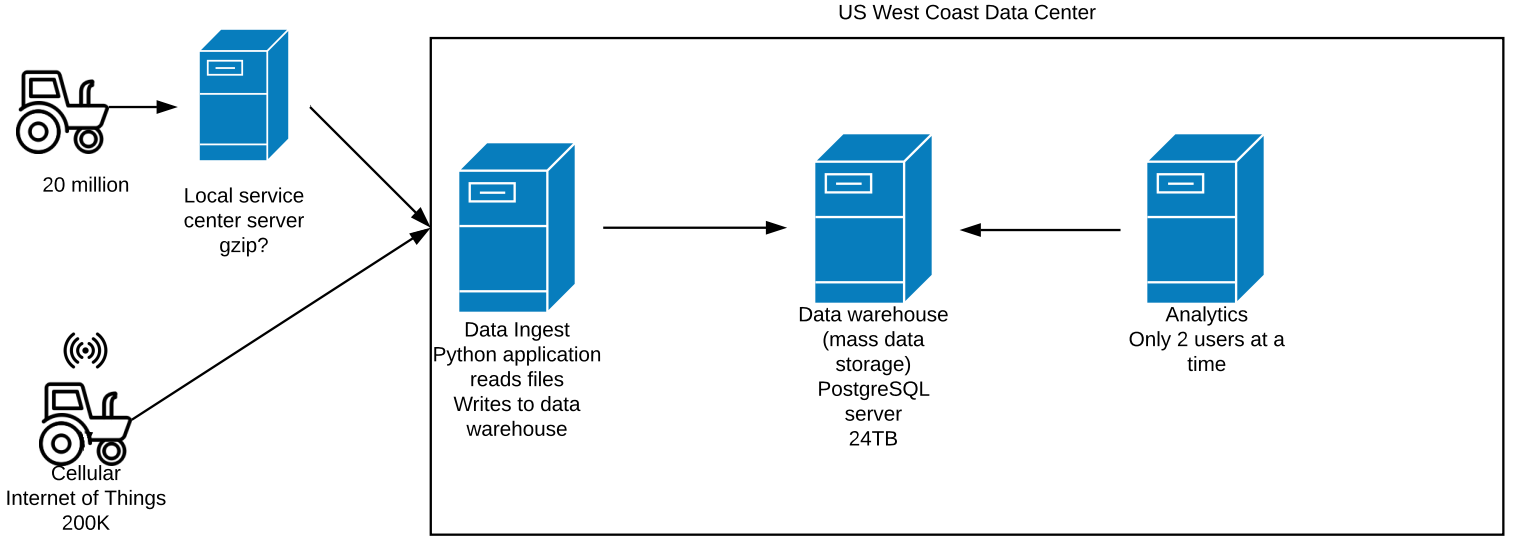
- heavy equipment, mining, agriculture
- bulldozers, tractors, etc
- 500 dealers all over the world
- mission = make customers more productive

Current setup

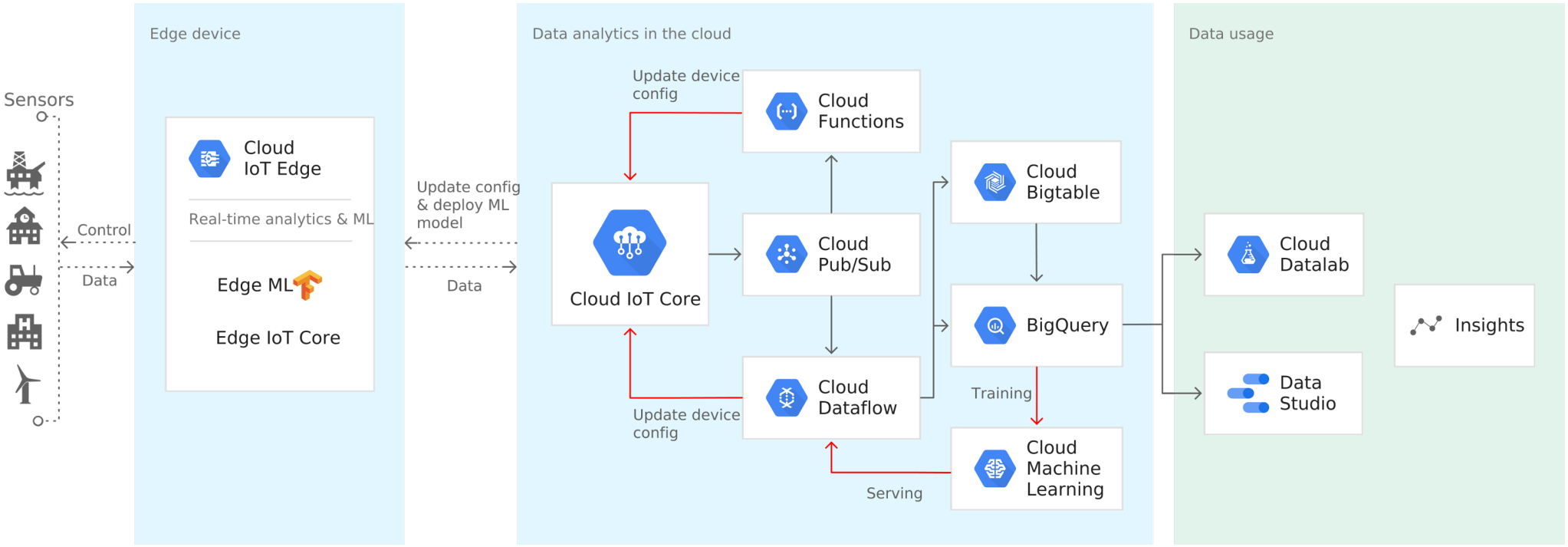
- Collect analytics on vehicles
  - Increase efficiency
  - Predict breakdowns and pre-stage replacement parts
- 20 million vehicles - each collect 120 fields per second
- Data stored locally, then uploaded (batch upload) when at dealer
  - Same port adjusts parameters
- 200,000 (10% of above) use cellular connection
  - Always streaming data
  - 9TB per day total upload

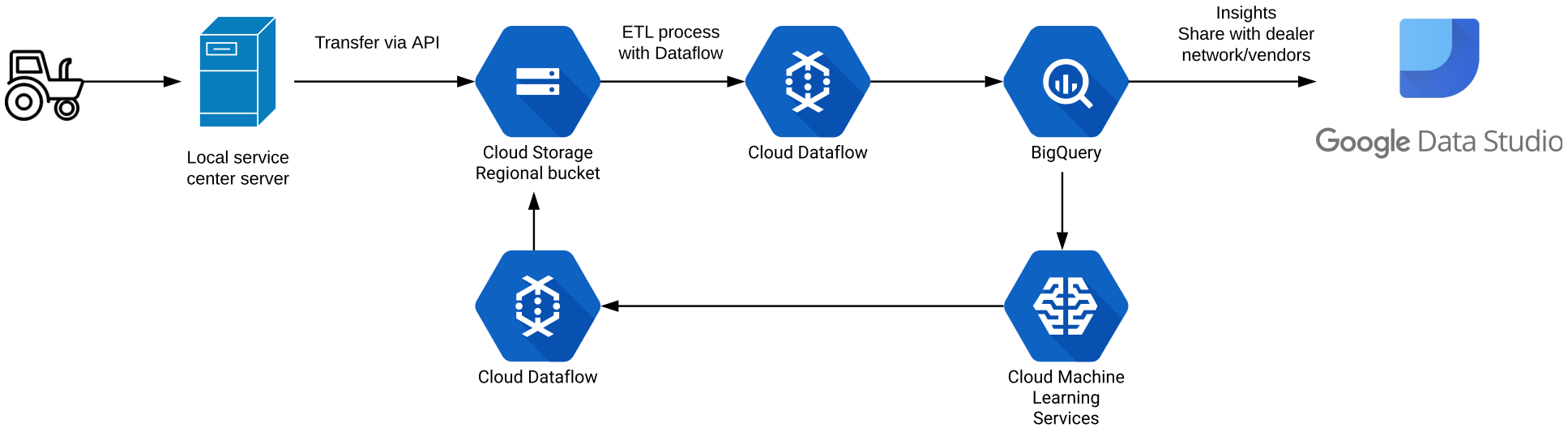
Problem to solve

- Above setup allows TerramEarth to preemptively stock replacement parts = reduced downtime
  - However, turnaround time is about 4 weeks
  - Needs to be 1 week
  - **Management priority = business agility**



Increase cellular connectivity to higher % of fleet
Single biggest contributor to slow turnaround time
Migrate from FTP batch upload to streaming upload





Additional notes

- Have the ability to partner with different companies—especially with seed and fertilizer suppliers in the fast-growing agricultural business—to create compelling joint offerings for their customers.
  - Use API for dealers and partners
  - App Engine + Cloud Endpoints
- Data Transfer to Google Cloud
  - IoT Core is latest capability to do so, can also write directly to Pub/Sub
    - Able to handle massive data ingest

Google Certified Professional - Cloud Architect - Part 3

Making the Case for the Cloud and GCP

Very Simple Questions

- What does Google Cloud Platform do that we can't do now?
- Why should we migrate our resources to GCP?
- Case studies as a reference.
- Start asking 'why?'

  Why should your CEO, CFO, or CIO care?

Why Move to GCP?

- Costs
- Future-proof infrastructure
- Scale to meet demand
- Data analytics/big data
- Greater business agility
- Managed services
- Global reach
- Security at scale

Costs

- ‘Catch-all’ for other reasons.
- Do more for less cost.
- Trade CapEx for OpEx.

  No need to spend big $$$ up front on hardware investments.

Future-proof Infrastructure

- Hardware does not wear out (end of life).
- Migrating data to new hardware every few years is a pain!

Scale to Meet Demand

- Elastic computing (a.k.a. distributed computing)
- Dynamically scale compute up and down as needed.

  Pay for only what you need, at that moment.

Greater Business Agility

- 'Need for speed'
- Create resources faster
- Act on data faster
- Do **everything** faster
- No waiting on hardware
- Rapid resource provisioning = greater flexibility/experimentation.

Managed Services

- 'Serverless'
- Let Google manage infrastructure for you.
- Less administrative overhead.

Global Reach

- Easy worldwide presence
- Multi-national resources on same private network (VPC)

Security at Scale

- Economies of scale at work.
- Over 500 security engineers protecting your data.

  They're really good at this.
- Ease of access management - projects

Google Certified Professional - Cloud Architect - Part 3

Cost Optimization

Why is this Important?

- All businesses want to do more at less cost.

  'Bang for your buck'
- GCP has many unique cost saving features to save money for more performance.
- May be testable.

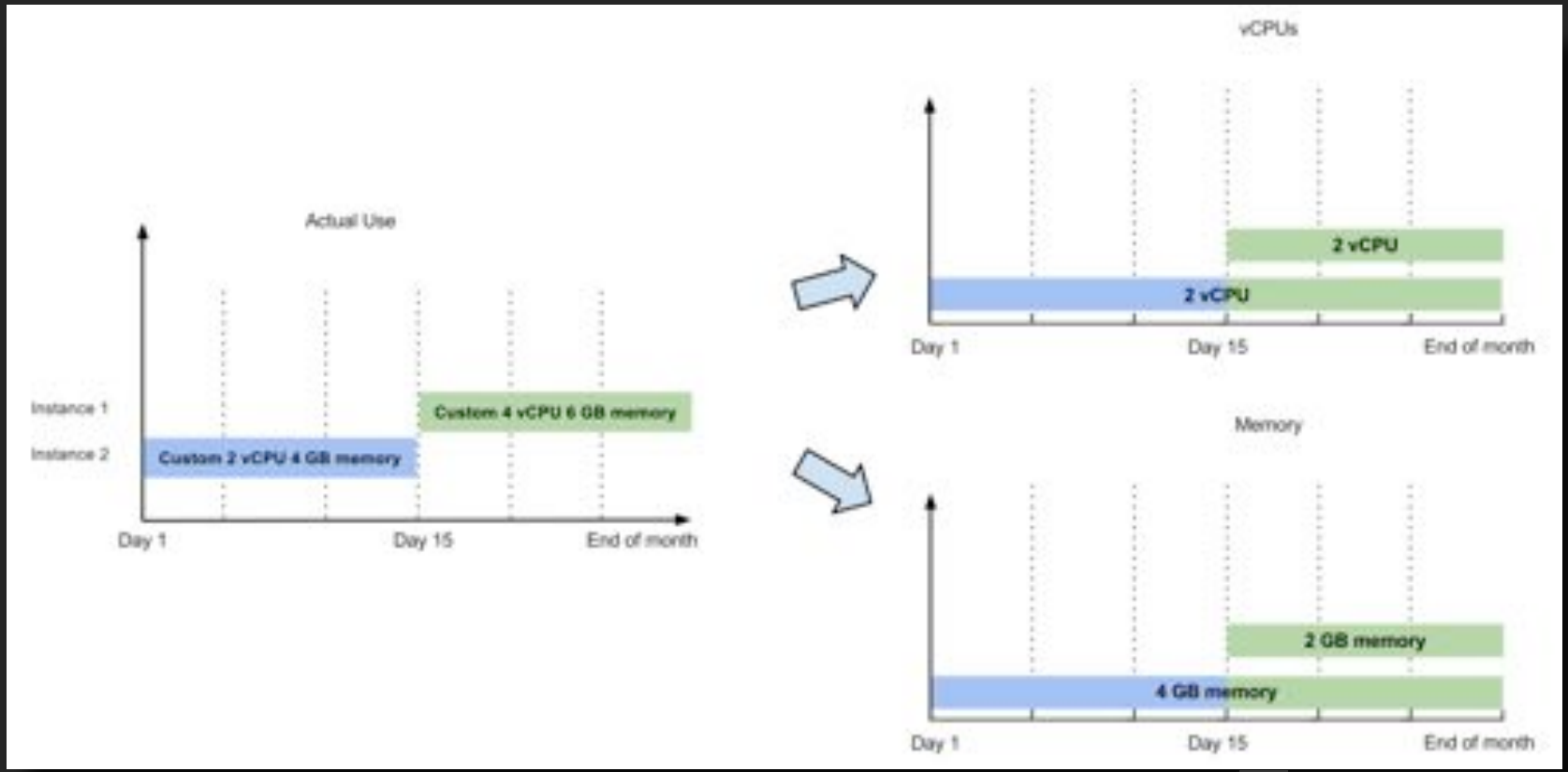
Sustained Use Discounts

- Up to 30% discount on Compute Engine and Cloud SQL VM's.
- The longer a VM runs per month, the greater the discount.
- Discount calculation is applied across different VM's throughout the month.
- High flexibility/agility (both financial and technical).
- No up-front commitment.
- Automatic, and simple.

How it Works


Actual Use
5
4
3
2
1
Day 1
End of month
Inferred Instances
5
2
4
5
1
3
Day 1
End of month
Discount Computation
5
2
4
5
1
3
0%
25%
50%
100%

Different custom machine types



Custom Machine Types

- Unique to GCP
- Customize # CPU's and RAM amounts
  - Greater customization than predefined types.
- JencoMart case study
- Choose from 0.9 to 6.5 GB RAM per CPU

Rightsizing Recommendations (Beta)

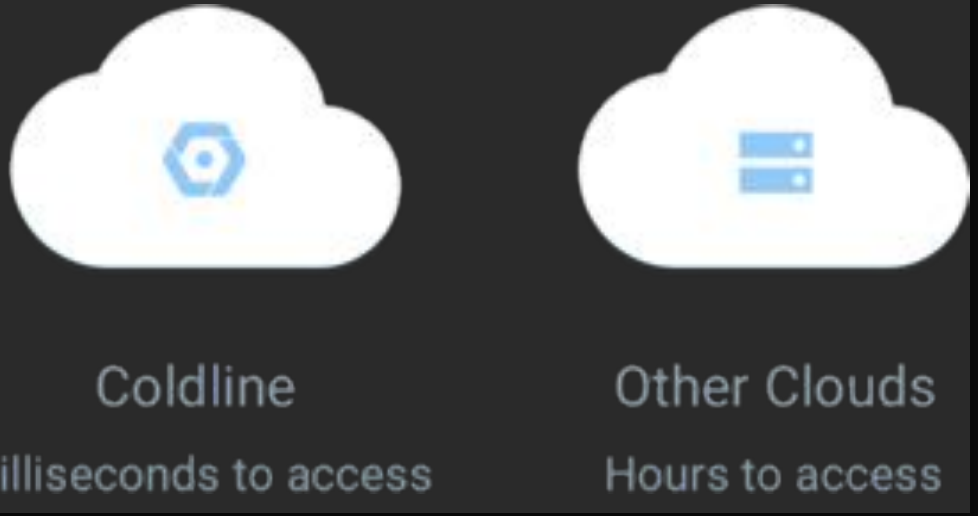
- Automatically recommend machine type resizing for Compute Engine VM's.
- Takes last 8 days of usage.
- Recommend sizing up or down to increase performance/save costs.
- Recommends custom machine types where applicable.

Preemptible VM's

- Low-cost, short life, interruptible VM's
- Up to 80% discount
- Fixed price, not variable market price = easier to budget for
- Ideal for fault tolerant, batch processing workloads
- New – preemptible VM's with GPU's (Beta)

Coldline Storage

- Very low cost cloud storage.
- Ideal for archive/disaster recovery data.
- Unique to GCP – low cost, but same fast access as premium storage.



Committed use discounts

- Commit to 1 or 3 year term for set amount of CPU's/RAM ('pool' of CPU/RAM).
- Billed for CPU/RAM amounts whether or not they're used.
- CPU/RAM pool can be used on multiple CE instances.
  - Discount automatically applied.
- Up to 57% discount.
- Example: commitment of 10 vCPU's, 30 GB RAM.
  - Two VM's with 4 vCPU/10 GB RAM each.
  - One VM with 2 vCPU/10 GB RAM.
  - Committed use discount applied to all machines.
- Does not stack with sustained use discounts.

Google Certified Professional - Cloud Architect - Part 3

Architecting Cloud Applications

App Design Requirements

- Five principles of good cloud app design:
  - High availability
  - Scalability
  - Security
  - Disaster Recovery
  - Cost
- Same principles regardless of compute platform (GCE/GKE/GAE)
- Understand how principles work across compute methods

High Availability

- "Can users access the application with minimal latency?"
- Placement of resources key.
- Who are your users?
  - Local company
  - Public users nationwide
  - International audience
- Regional deployment, multi-regional.
- Serve traffic to multiple regions via global load balancing.

Scalability

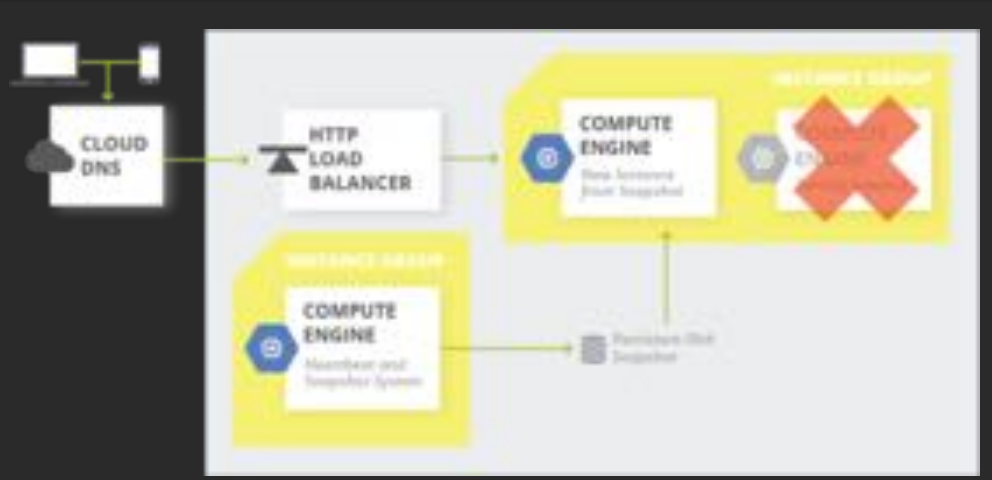
- More compute when needed, less when not
- Autoscaling
- Best practice - Run load tests
- GCE – Managed instance group with autoscaling
- GKE – Cluster with autoscaling enabled
- GAE – Autoscaler built-in
- GAE considerations
  - Standard – daily spending limit
  - Quota limits on API calls

Security

- Limit access to those who need it
- Principle of least privilege
- Secure administrative access
- IAM roles – limit personnel access
- Firewall rules to restrict traffic

Disaster Recovery

- What to do when something goes 'boom'?
- Service not available = lost business
- GCE – snapshots for individual instances
- Failover server
- Backup data to cloud storage bucket
  - Database data
- Manage/rollback app versions:
  - GCE – instance group rolling update
  - GKE – rolling updates
  - GAE – traffic splitting/versions



Costs

- GAE (Standard)– set daily spend limit
- GAE (Flexible) – set custom machine types
- GCE – managed instance groups w/ autoscaling
- Custom machine types for perfect sized VM
- Preemptible VM's

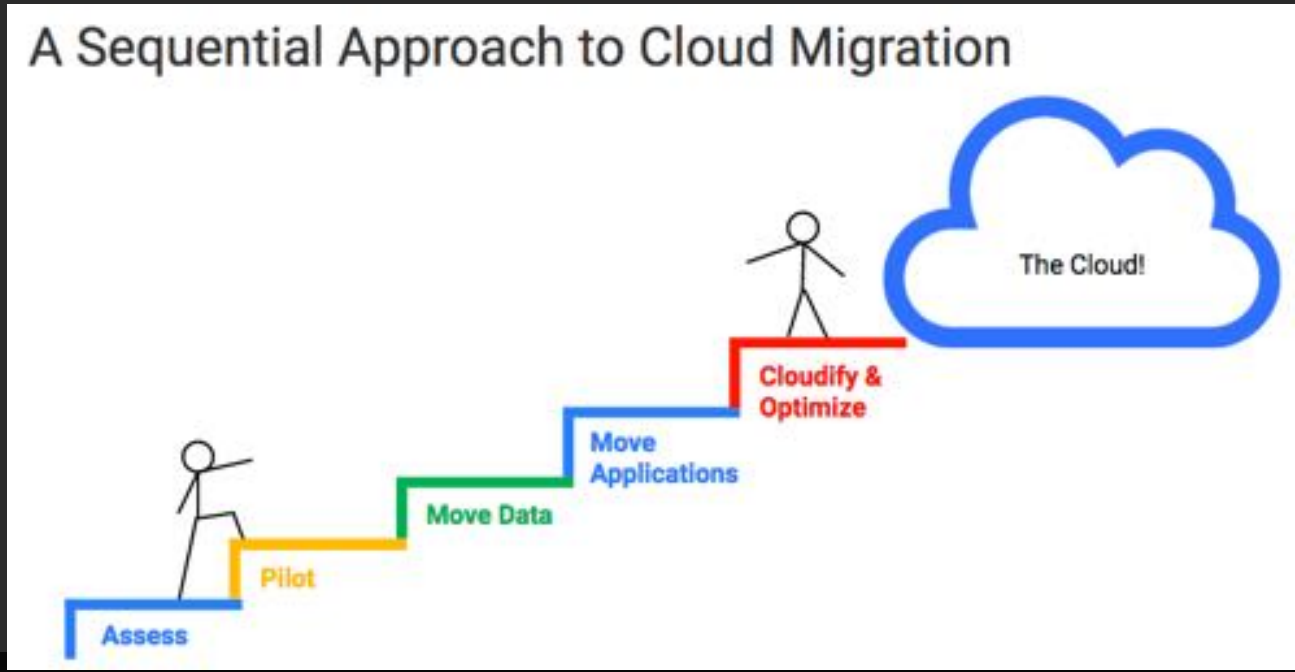
Google Certified Professional - Cloud Architect - Part 3

Planning a Successful Cloud Migration

How do you get your ‘stuff’ into GCP?

Five phases for successful migration

- Assess
- Pilot
- Move Data
- Move Applications
- Optimize



Assess

Three categories:

- Easy to move
- Hard to move
- Can't move

Evaluation criteria

- Criticality of application
- Compliance
- Licenses
- ROI

Consider application dependencies

Largest immediate benefits

Pilot

Proof of concept/test run

Non-critical or easily duplicated services

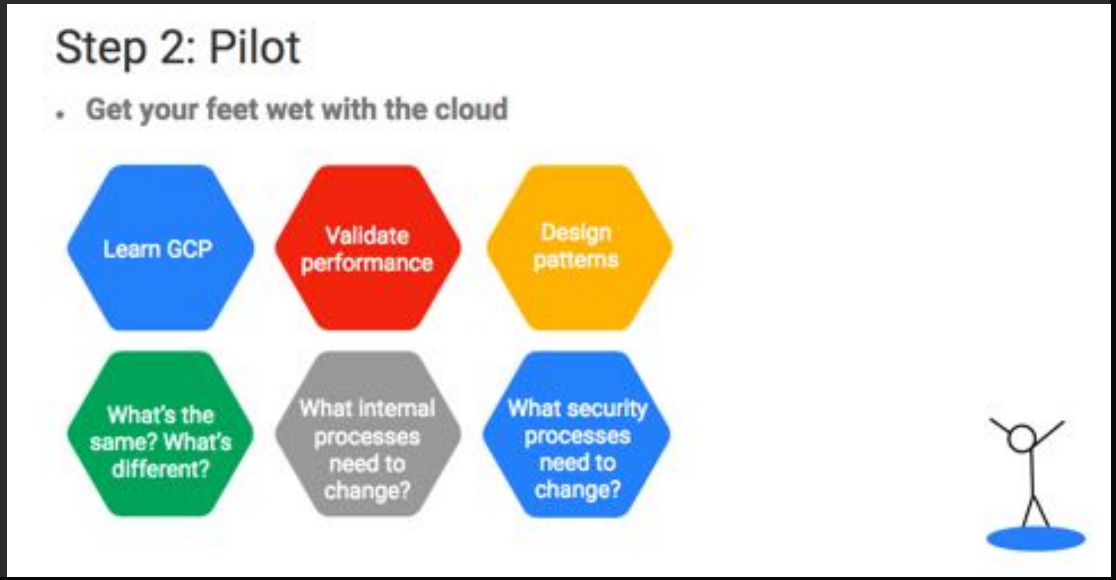
Small steps at first

Considerations

- Licensing
- Roll back plan
- Process changes

Start mapping roles

- Projects
- Separation of duties
- Test/Production environments
- VPC's



Move data

Data before applications

Evaluate storage options

Transfer methods

Source	Destination	Tool(s)
On-prem data	Google Cloud Storage (GCS)	gsutil, transfer appliance, batch upload, drag and drop
On-cloud data (S3)	GCS	Storage Transfer Service
Database (SQL)	GCS/Cloud SQL/Spanner	Batch import mysqldump
Database (Non-SQL)	GCS	Batch upload to GCS
Database (Non-SQL)	Compute Engine	Backup files to persistent disk Stream to persistent disk

Move applications

Self service or partner assisted

Keep it simple (usually) – 'lift and shift' recommended

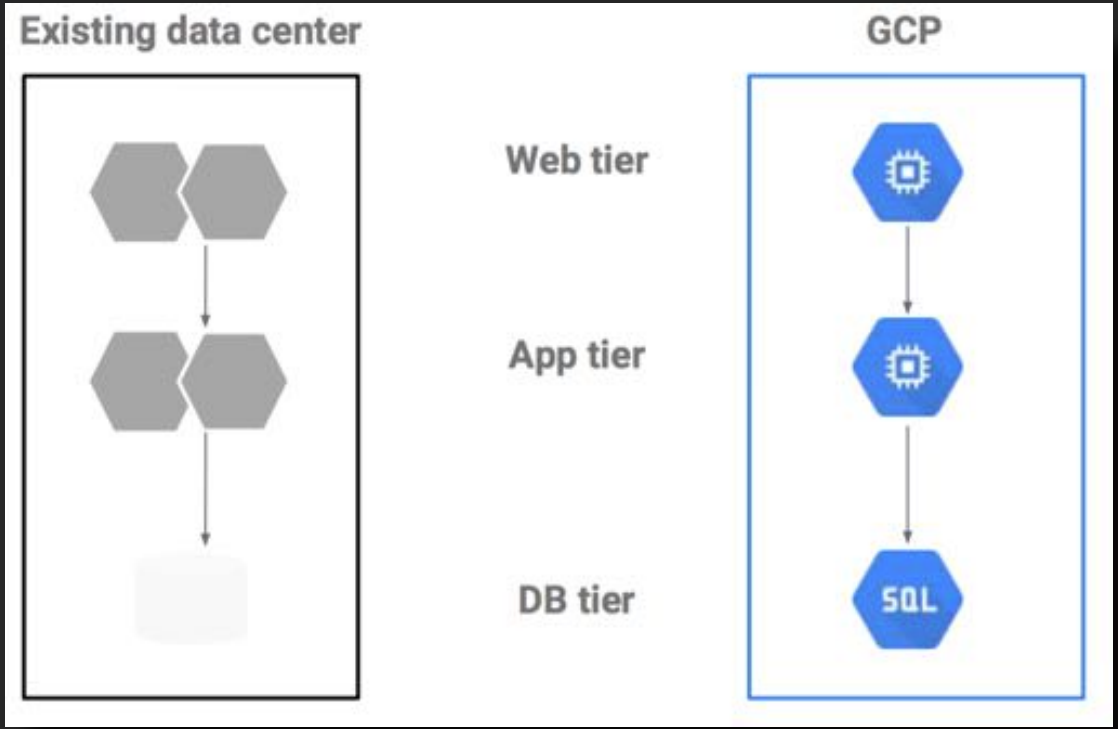
- Create duplicate environment of on-prem resources
- Managed services?

VM import freely available options via CloudEndure

Other options:

- Hybrid – resources in both environments
- Backup-as-migration

Lift and shift example



Optimize!

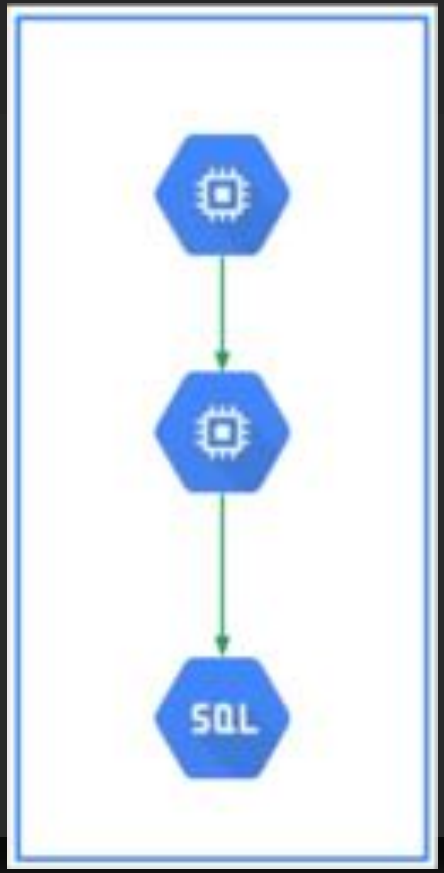
Cloud makeover

Re-tool processes and apps with modern GCP tools

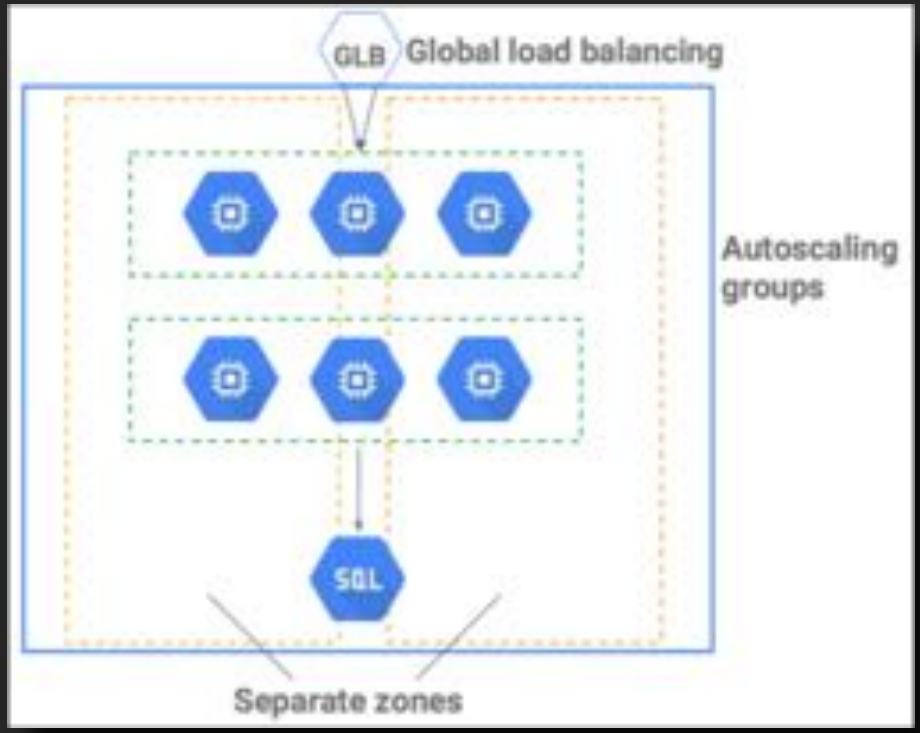
- Offload static assets to Cloud storage
- Enable auto scaling
- Enhance redundancy with different availability zones
- Enhanced monitoring with Stackdriver
- Managed services
- How to launch future resources (with less baggage)
- Decouple stateful storage from application

Example

Simple server stack (web tier, app tier, SQL database)

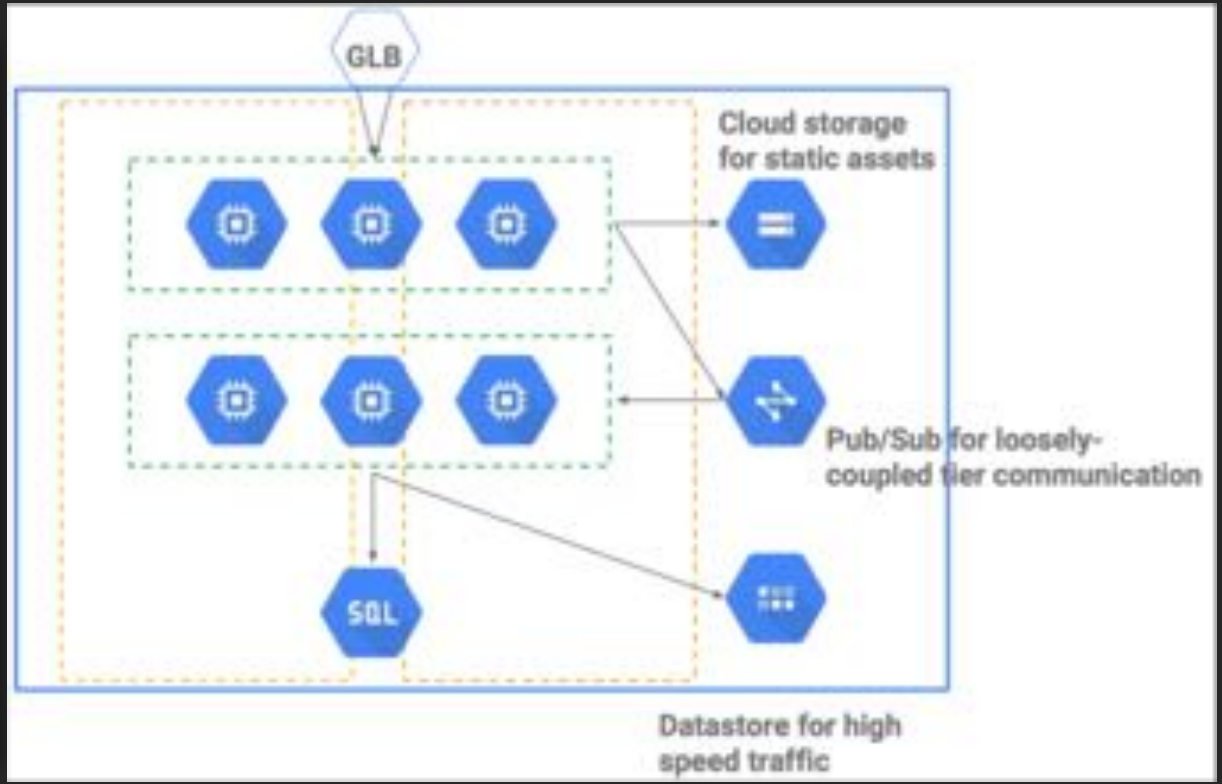


Add availability and elasticity

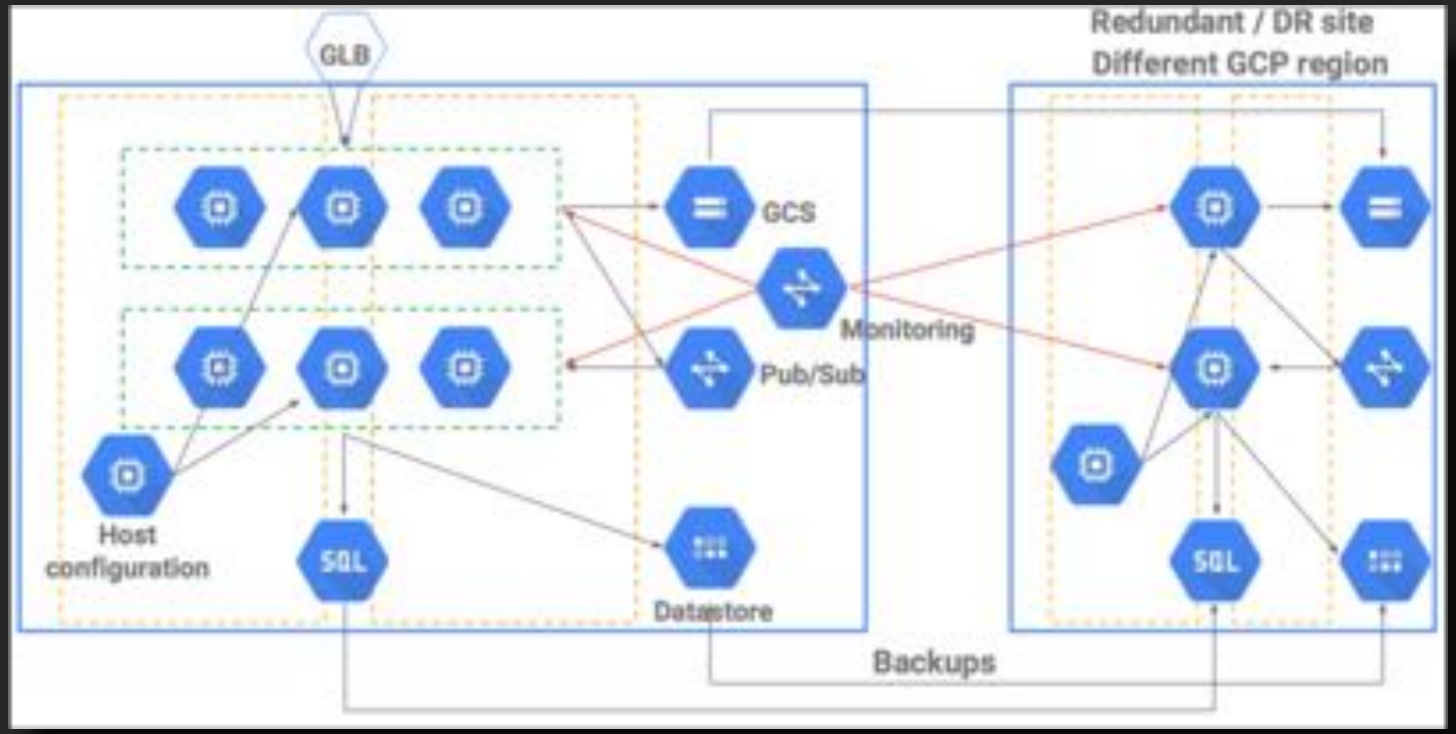


Further optimization

- Consider storage options
- Deployment Manager to create environment
- Monitoring with Stackdriver



Full featured cloud solution



Google Certified Professional - Cloud Architect - Part 3

Storage Transfer Service

Storage Transfer Service

Import online data into Cloud Storage:

- AWS S3 bucket
- HTTP/HTTPS location
- Another Cloud Storage bucket

Import from online data source (above) to data sink:

- Sink = Cloud Storage bucket

Capabilities

- Back up data from other storage providers.
- Move data from one GCS bucket to another .

    Example: Move from Multi-Regional bucket to Nearline bucket to lower costs.

Transfer Operation

- Configured through transfer job:
    - One time or recurring transfer.
    - Delete destination objects if not present in source.
    - Delete source objects after transfer.
    - Periodic sync of data source and data sync.
- Requires owner or editor project IAM role + access to source and sink:
    - Source/sink can be outside of project.
    - Service account accesses source/sink.

gsutil or Storage Transfer Service?

- Cloud storage provider (GCS, AWS, HTTP) - use Storage Transfer Service.
- On-premises location - use gsutil.

Google Certified Professional - Cloud Architect - Part 3

Migrating Applications

Server Migration

- Migrating from on-premises = migrating servers
- Application migration = server migration
- First assess what can be moved
- Map to additional GCP services

Map to GCP Services

Service Type	Data Center	Google Cloud Platform
Compute	Physical hardware, virtualized hardware (VMWare ESXi, Hyper-V, KVM, XEN)	Google Compute Engine
Storage	SAN, NAS, DAS	Persistent disk, Google Cloud Storage
Network	MPLS, VPN, hardware load balancing, DNS	Google Cloud VPN, Google Cloud Interconnect, Compute Engine load balancing, Google Domains, Google Cloud DNS
Security	Firewalls, NACLs, route tables, encryption, IDS, SSL	Compute Engine firewalls, encryption, IDS, SSL
Identity	Active Directory, LDAP	IAM, GCDS, LDAP
Management	Configuration services, CI/CD tools	Deployment Manager, configuration services, continuous integration/continuous delivery (CI/CD) tools

Before Moving the Server...

- Create a project
- Determine network configuration (VPC)
  - Firewall
  - Regions
  - Subnets
- Determine IAM roles – who needs access to what?

“Lift and shift” Choices

- Recreate server environment on Compute Engine public image
  - Create new GCE instance, install application/import data
- Import direct image
  - Carbon copy snapshot
- Recommended to run on GCE public image
- Reasons for direct image import:
  - Require operating system that is not provided as a public image.
  - Already have a set of basic images that you use to create virtual machines in another cloud platform.
  - The work required to migrate application code to one of the public images is greater than the work required to complete the boot disk image import process.

Importing Boot Disk Images – Two Choices

Manually create disk image file → Cloud Storage → import as custom image

- Linux only
- Compress into .tar.gz format (gzip compression)

Migrate entire server system with CloudEndure VM migration service

- Free service

Limitations to VM Migration

Windows

- Microsoft Windows Server 2008 R2 64 bit
- Microsoft Windows Server 2012 R2 64 bit
- Microsoft Windows Server 2016 64 bit
- Windows desktop operating systems **not** supported
- Licensing converted to GCP's pay-as-you-go system

Linux

- SUSE Linux (SLES) 11 or above
- Debian Linux 8
- Kali Linux 2.0
- Ubuntu 12.04 or above
- Red Hat Enterprise Linux (RHEL) 5.0 or above

Google Certified Professional - Cloud Architect - Part 3

Data Migration Best Practices

Moving lots of data, how ‘close’ is it?

‘Close’ defined

From cloud = very close

- Storage Transfer Service

From colocation or on-premises datacenter = close

- Fast bandwidth
- Copy with gsutil (or third party tools)

Slower connections = far

- ‘Mail it in’

Two options to make data transfer easier

Decrease data size

- Dedupe and compress
- Both reduces transfer time and storage costs
- For many small files, compressing and grouping together = faster transfers

Increase network bandwidth

- Public Internet connection
- Direct peering
- Cloud Interconnect

gsutil copy considerations

Limitations

- No network throttling
- Best for one time/manual transfers
- For ongoing, automated transfers, use cron job

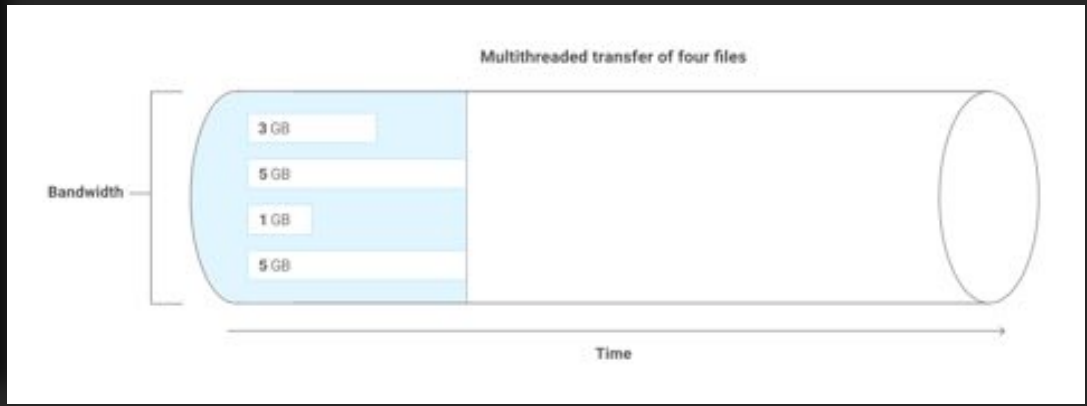
Tools for faster/better transfers

- Multi-threaded/processed. Useful when transferring large number of files.
- Parallel composite uploads. Splits large files, transfers chunks in parallel, and composes at destination.
- Retry. Applies to transient network failures and HTTP/429 and 5xx error codes.
- Resumability. Resumes the transfer after an error.

Multi-threading transfer

-m option

gsutil -m cp -r [SOURCE_DIRECTORY] gs://[BUCKET_NAME]

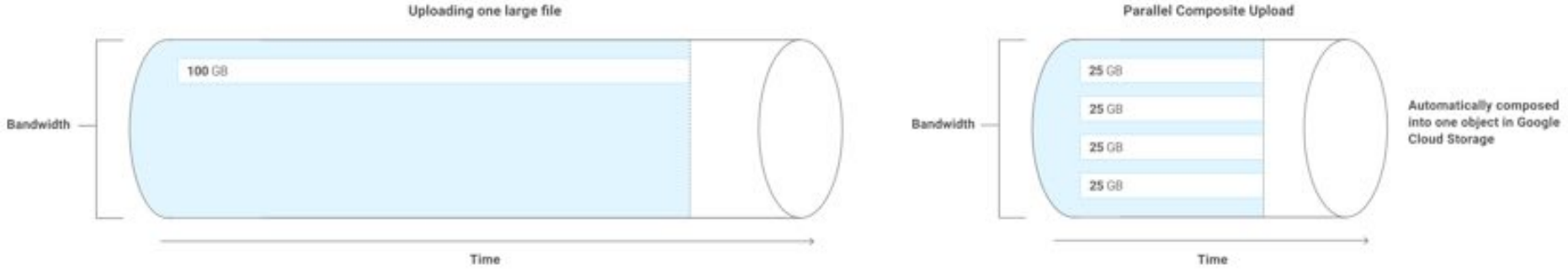


Parallel uploads

Break single file into chunks for parallel upload

Don't use for nearline/coldline buckets – extra charge for 'modifying' files on upload

gsutil -o GSUtil:parallel_composite_upload_threshold=150M cp bigfile gs://your-bucket



‘Far’ option – mail it in

Physical media options

Google Transfer Appliance (Beta)

Recommended when:

- Takes over a week to upload data
- 20TB or more of data, regardless of connection speed

Third party partners offer similar service

Physical media options are encrypted

Google Certified Professional - Cloud Architect - Part 3

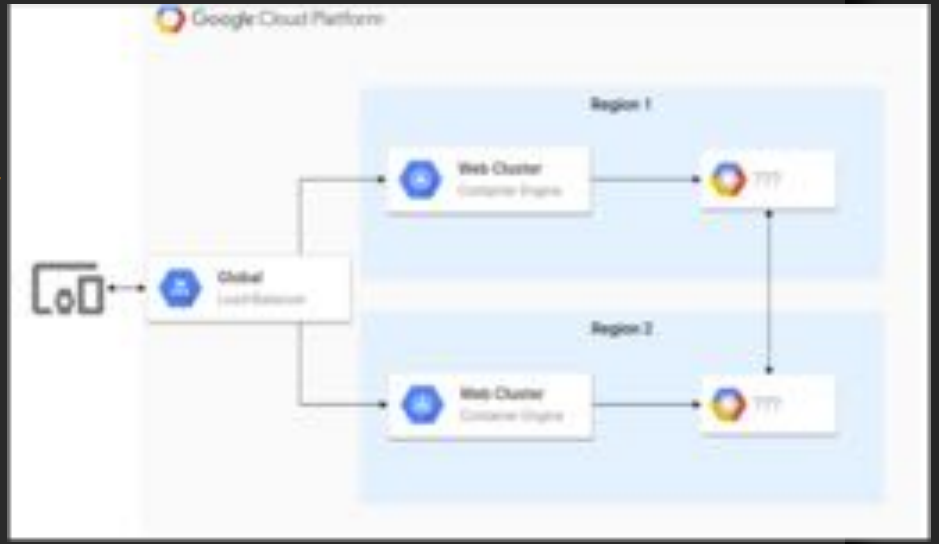
Mapping to Storage Solutions

Mapping Storage Types – at a Glance

- Unstructured Data - Cloud Storage
- Relational Data (SQL) - Cloud SQL and Spanner
- Non-relational (NoSQL) Data - BigTable and Datastore
- Big data analysis (SQL queries) - Google BigQuery
- Other - Persistent disk

Cloud Storage

- Unstructured data 'catch all' repository
- Infinitely scalable
- Multi-regional support
- Multiple sources can write
- Different storage classes to balance price/availability
- Less expensive than persistent disk
- Pay per use – no pre-allocation necessary



Persistent disk

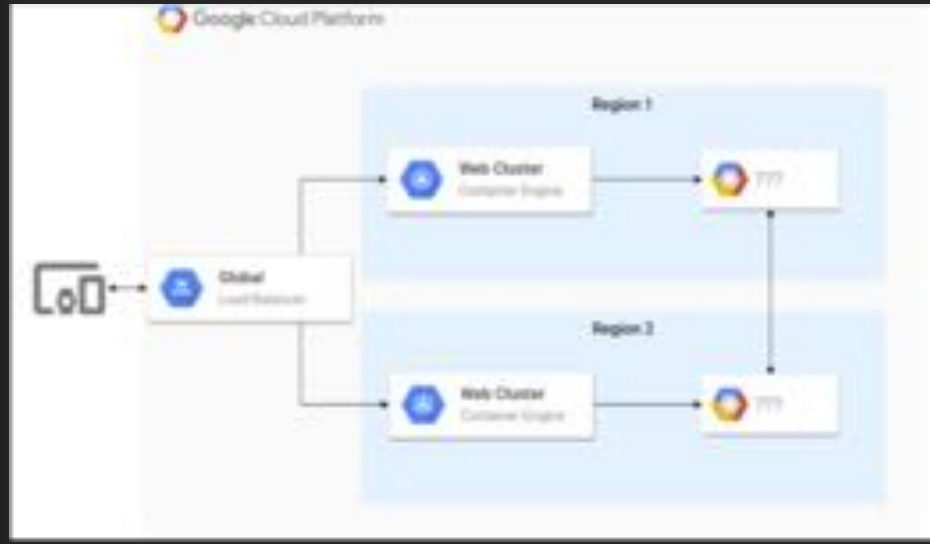
- Non-managed
- Direct mapping to legacy systems
  - "It's a disk"
- All types of data – works with almost everything
- Attached disk for GCE/GKE workload
- Multiple read, only single write
- Same limitations as a disk
- Rapid backups with snapshots

Cloud SQL

- Traditional relational database (SQL)
- Managed single Compute Engine VM
- Single zone instance (though with read replication options)
- Easy to migrate to, but with all limitations of traditional SQL:
  - Limited growth
  - Not horizontally scalable

Cloud Spanner

- Fully managed
- Horizontally scalable relational (SQL) database
- Cross-region availability
- Advantages of relational database, without the drawbacks
- More expensive than Cloud SQL (starting at $0.90/hr/node)



Datastore

- Fully managed
- Non-relational (NoSQL) database
    - Semi-structured, ACID transactions
- Scales from zero to terabytes
- Ideal for web and mobile applications (e.g. MountKirk Games)

Bigtable

- Fully managed
- Non-relational (NoSQL) database
- Compared to Datastore:
    - More ideal for analytics
    - More expensive
    - Requires managing nodes
- HBase compatible
- Ideal for terabyte and larger databases (up to petabytes)

BigQuery

- Fully managed
- High capacity data warehouse/analytics
- Big data exploration and processing
- Not ideal for operational database
- SQL queries
- Mountkirk Games and TerramEarth case studies

Decision Chart

https://cloud.google.com/storage-options/

Google Certified Professional - Cloud Architect - Part 3

Preemptible VM's

What is it?

- Short-lived, low-cost VM:
  - 24 hours max
  - Can be shut down at any time (30-second warning)
- 'Disposable' – not for critical single VM's
- Ideal for fault-tolerant, batch processing workloads:
  - Rendering
  - Media transcoding
  - Big data analytics
- Most often used in managed instance groups:
  - 'Swarm' tactics
  - Throw more CPUs at it!
- Fixed pricing, up to 80% off regular instance price
- Storage and licensing same cost
- Otherwise, exactly like any other VM

220,000 cores and counting: MIT math professor breaks record for largest ever Compute Engine job

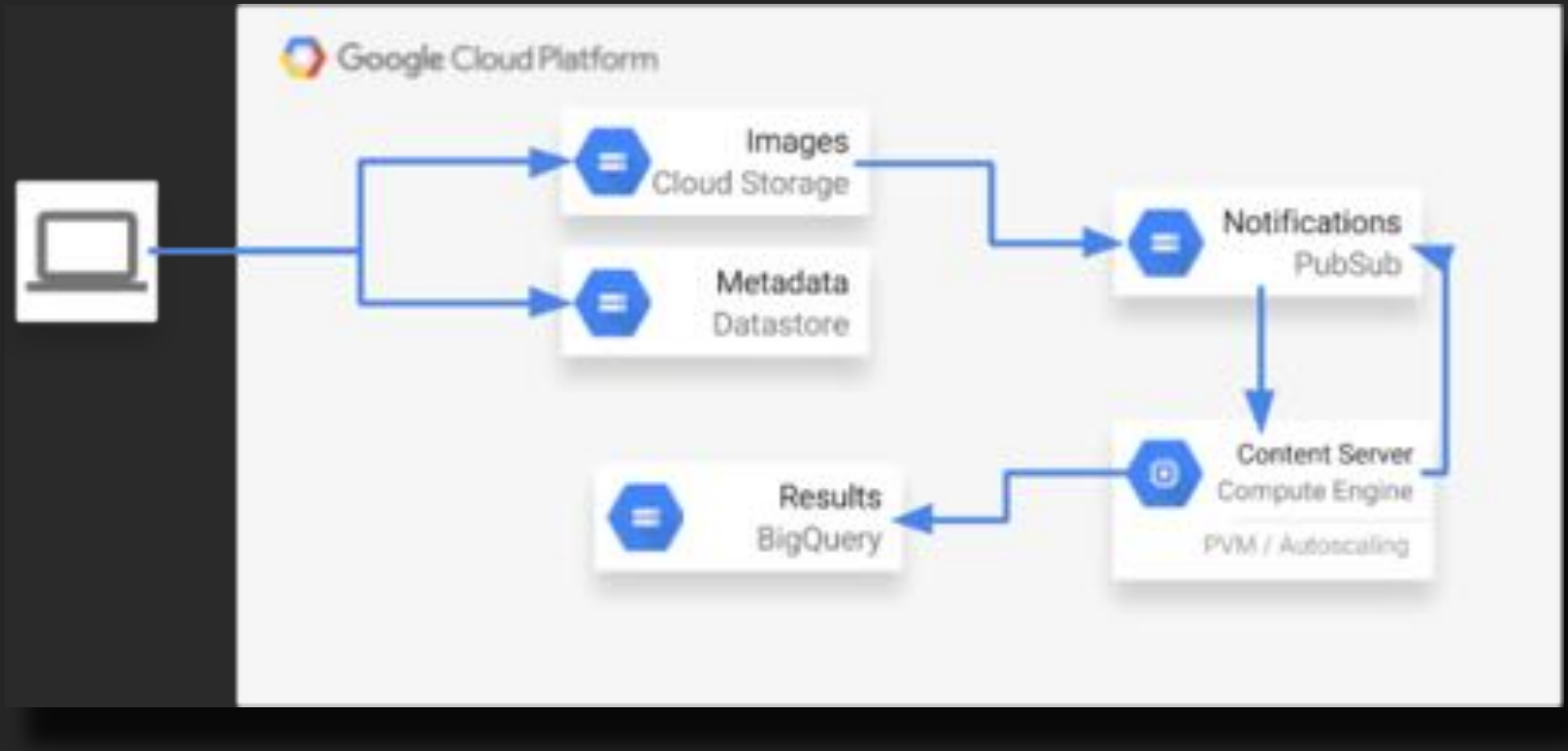
Thursday, April 20, 2017

By Alex Barrett, GCP Blog Editor & Michael Basilyan, Product Manager, Compute Engine

***Editor's Note**: This post was updated on June 12, 2017.*

An MIT math professor recently broke the record for the largest ever Compute Engine cluster, with 220,000 cores on Preemptible VMs, the largest known high-performance computing cluster to ever run in the public cloud.

How it Works


Google Cloud Platform
Images
Cloud Storage
Metadata
Datastore
Notifications
PubSub
Content Server
Compute Engine
PVM / Autoscaling
Results
BigQuery

Best Practices

- Use smaller machine types:
  - Many small machines are better than fewer large
  - Less likely to be shut down
- Run jobs during off peak times:
  - Nights and weekends
- Design application for fault/pre-emption tolerance:
  - Test by manually stopping the instance
  - 'Embarrassingly parallel' operations
- Preserve disk on machine termination (individual instances)
- Use shutdown scripts:
  - Save job progress to pick up where left off

Possible exam scenarios

Create and terminate machine to save costs, but preserve disk state.

- --no-auto-delete --disk example-disk

Managed instance group with PVM's keep recreating every minute.

- Health check/firewall configuration

Google Certified Professional - Cloud Architect - Part 3

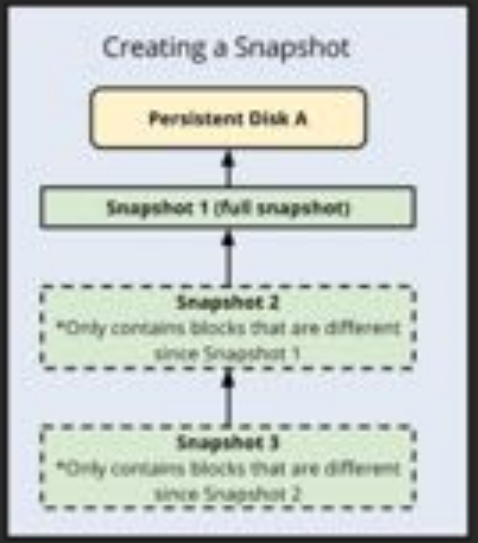
Backups and Disaster Recovery

Exam Backup/DR Topics

- Backup individual GCE instances
- Database backup
- Cloud Storage backup/rollback
- Distributed computing application roll back:
  - GCE managed instance group
  - App Engine
- Scheduling automated backups
- Exam scenarios

Back up Individual GCE Instances

- Disk Snapshots
- Copy of entire disk
- Requires 'freezing' disk activity (stopping services, etc.)
- Incremental:
  - Only backs up what's different from the previous backup



Automated Database Backups (GCE)

- Backup database, not entire disk
- Snapshots require 'freezing' database during backup
  - Requires periodic downtime
- Configure application to back up database to another persistent disk/Cloud Storage
- Use cron job to back up

Cloud Storage Backup/Rollback

- Object versioning + lifecycle management:
  - gsutil versioning set on gs://[BUCKET_NAME]
- Delete + rollback protection
- Revert to earlier version

Distributed Computing Application Rollback

- GCE managed instance group + Google App Engine
- Individual instances are not backed up
- Rollback to previous group versions
- Compute Engine Managed instance group
    - Rolling update → apply previous instance group template
        - Optional: set target % other than 100
- App Engine
    - Versioning control/split traffic

Scheduling Automated Backups

- Use scheduled cron jobs
- Apply to snapshots, database backup

Exam Scenarios

Rollback plan for managed instance group serving website – 100's of instances:

- Object versioning on static data in Cloud Storage
- Rolling updates
- NOT snapshots

Backup critical database with zero downtime and minimal resource usage:

- Scheduled cron job
- Backup database data to another location (persistent disk/Cloud Storage)

App Engine – need to push risky update to live environment:

- Versioning/traffic splitting
- Deploy update to small % of traffic as canary update

Google Certified Professional - Cloud Architect - Part 3

Security Methods in GCP

Purpose of Lesson

- Review of security principles, with some new items
- Exam focus of what security challenges may be tested
- IAM roles play substantial role in most scenarios (but not all)

Separation of Duties

Don't place all organization resources in one project:

- Difficult to limit access

Principle 1: different scopes of access = separate projects:

- Example: Separate development and production environments by team
- Individual user accounts – separate dev and prod projects for each team

Principle 2: give fewest rights necessary – organization and project levels:

- Example: security team needs detailed visibility to all projects in organization
- Only need to view org-wide = organization viewer/project viewer

Securely Interact with Google Cloud Storage

- Three access control methods:
  - IAM
  - ACL
  - Signed URL
- IAM = bucket level permissions
- ACL/Signed URL = object level permissions
- Signed URL does not require GCP account (user uploads)
- Single bucket for separate user assets – each object secured to user
- Example: External customer upload PII data to GCS – does not have GCP account:
  - Single bucket for all users
  - Signed URL for secure access to only their data

Penetration Testing ('pen test')

- Simulated attack on your computer system to find vulnerabilities
- Find holes before the bad guys do
- Exam focus: Choose correct environment to conduct pen test:
  - Pen test should be same avenue as a real attack
  - Example: Publicly available application should test from outside GCP over public Internet

Google Certified Professional - Cloud Architect - Part 3

Network Design for Security and Isolation

Enterprise Focus

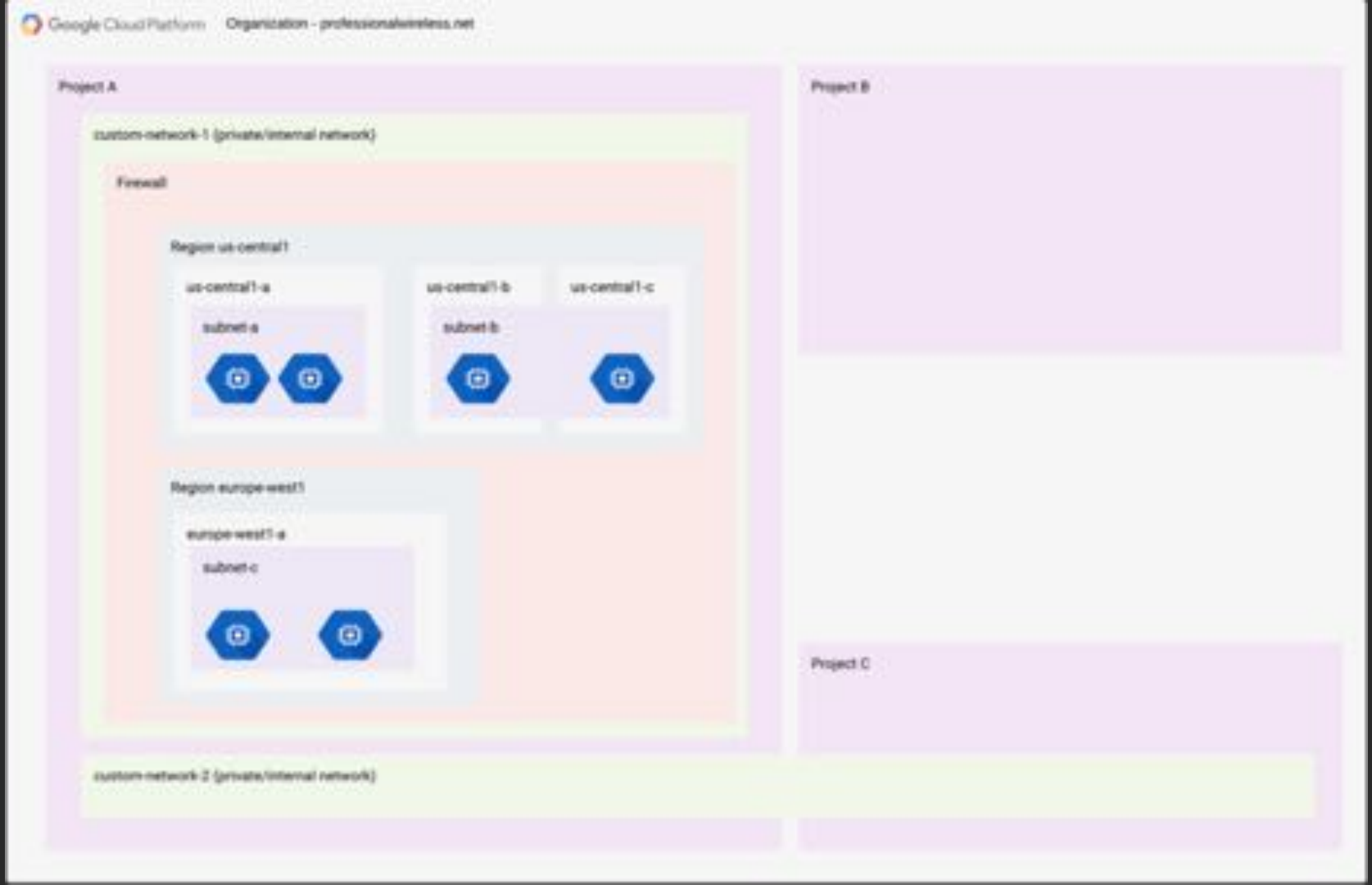
- Designing for ‘skyscrapers’
- Increased layers of complexity
- Exam focus/scenarios

Tools at a Glance

- Methods of isolation, to include:
  - Projects (includes IAM)
  - VPC's
  - Firewall
  - Bonus: Bastion Host
- Principle of least privilege

“Gotta keep ‘em separated”

Organization → Projects → VPC → Regions → Subnets


Google Cloud Platform
Organization - professionalwireless.net
Project A
custom-network-1 (private/internal network)
Firewall
Region us-central1
us-central1-a
subnet-a
us-central1-b
subnet-b
us-central1-c
Region europe-west1
europe-west1-a
subnet-c
custom-network-2 (private/internal network)
Project B
Project C

Projects

- Primary method of full isolation between environments
- Projects further divided into VPC's
- Separation of people/accounts - IAM to restrict access to project resources:
  - Further separation by service, but not by VPC
- Project-wide IAM roles = access to all VPC's within project
- Example: Give Bob access to Project A, but not Project B

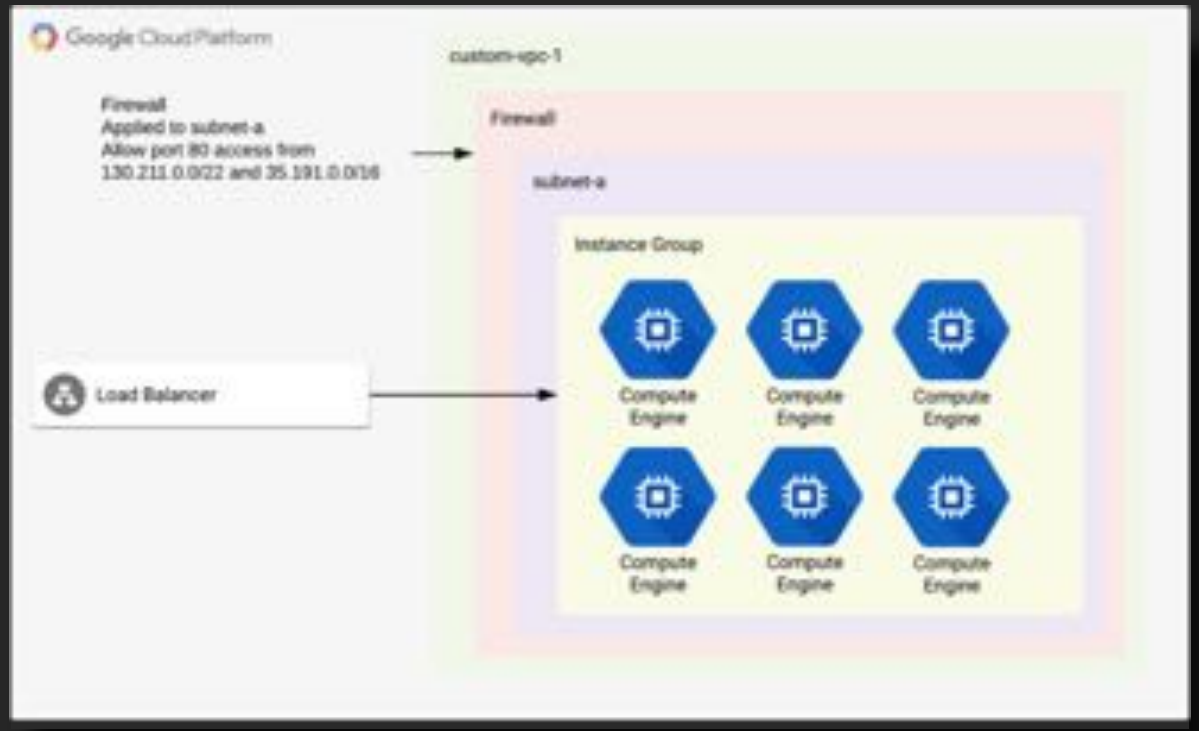
Virtual Private Cloud (VPC's)

- Can have multiple VPC's per project
- Isolate resources (e.g., groups of VM instances)
- All VPC resources in same private IP network (RFC 1918 space):
    - Global access to same private network
- Project users have same access to all VPC's:
    - Granular IAM roles divide access by GCP service, not VPC
    - Cannot give Bob access to custom-network-1 in Project A, but restrict access to custom-network-2 in Project A

Firewall

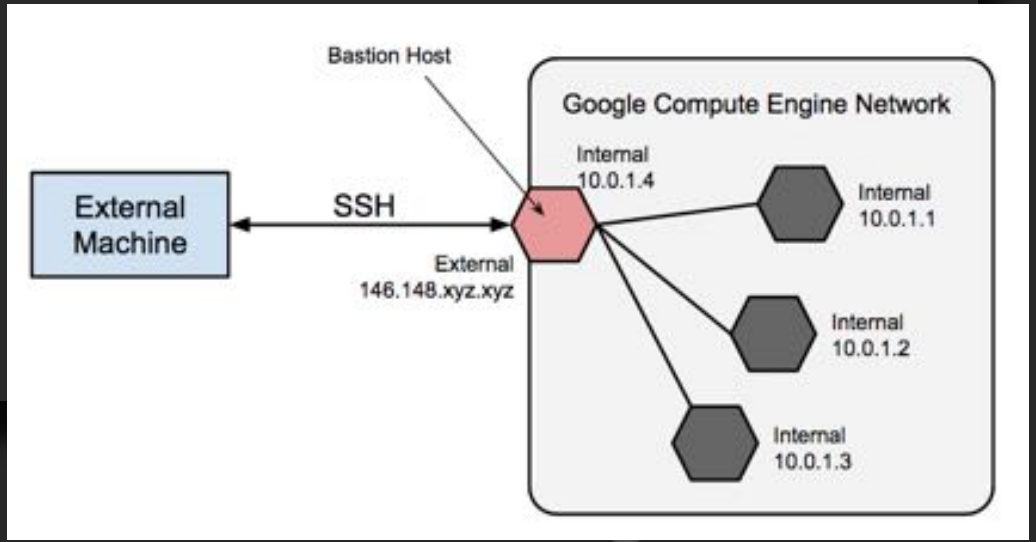
- Separate access by network location
- Control both ingress and egress traffic
- Limit access by:
    - Port
    - IP address/range (internal only, all external, certain IP ranges)
    - Between subnets
    - Tags
- Load balancer/health check interactions:
    - Firewall controls access at instance level, not load balancer
    - Must allow load balancer traffic to connect to backend instances (also allows health check)
    - Network Load Balancer = 209.85.152.0/22, 209.85.204.0/22, and 35.191.0.0/16
    - HTTP(S)/SSL proxy/TCP proxy/internal LB = 130.211.0.0/22 and 35.191.0.0/16

Firewall and Load Balancers



Bonus: Further Isolation by Disabling External IP Address

- 'Air gap' resources by removing external IP address
  - Greatly limits exposure
- Problem: can't administer air gapped resources
- Two primary options:
  - VPN connection – place myself on internal network
  - Bastion host
- Disabling external SSH is different:
  - Can still access via Cloud Shell



Exam scenarios

Instance group VM's keep restarting every minute:

- Failing health check
- Configure firewall to allow proper access to instance group VM's (subnet, tag) from load balancer IP

On-premises network access to proper network resources:

- Restrict ingress firewall access to on-premises network IP range

Failover from on-premises load balancer hosted application to GCP hosted instance group ('hot standby'):

- Consider security and compliance
- Allow firewall access at instance group level (subnet/tag) from outside source

External SSH access disabled, but operations team needs to remotely manage VM's:

- Give operations team access to Cloud Shell
- Not same scenario as removing external IP's

Google Certified Professional - Cloud Architect - Part 3

Legal Compliance and Audits

Exam Outline can be Confusing

Designing for legal compliance. Considerations include:

- Legislation (e.g., Health Insurance Portability and Accountability Act (HIPAA), Children's Online Privacy Protection Act (COPPA), etc.)
- Audits
- Certification (e.g., Information Technology Infrastructure Library (ITIL) framework)

Of the above, audits will be a frequently seen topic.

Audits

If you see the terms: audit, auditor, access logs, compliance, think Stackdriver Logging.

Audits = Stackdriver

HOWEVER!

Similar but different: Billing data is not through Stackdriver:

- Billing data exported directly to Cloud Storage/BigQuery

Automating/Exporting Logging Data for Audits

Scenario: automatically export access log reports for auditors back to Stackdriver again.

In Stackdriver Logging:

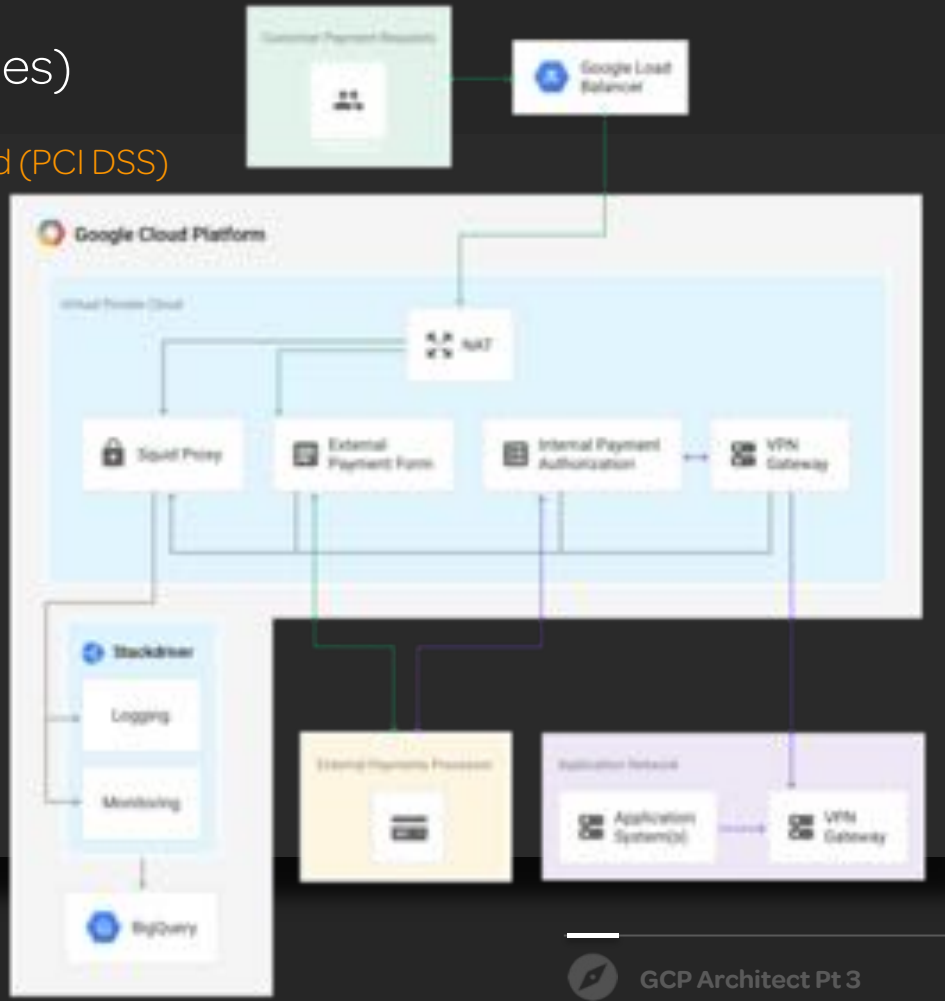
- Create sink based on logs filter.
- Export sink data to destination (Cloud Storage, BigQuery, Pub/Sub).
- BigQuery = analysis
- Cloud Storage = access for external parties (signed URL one option).
- Sink export only exports new data since sink was created.

Scenario solution:

- Create sink for needed log data.
- Export to Cloud Storage/BigQuery.
- Provide auditor access to export location (GCP account/signed URL).

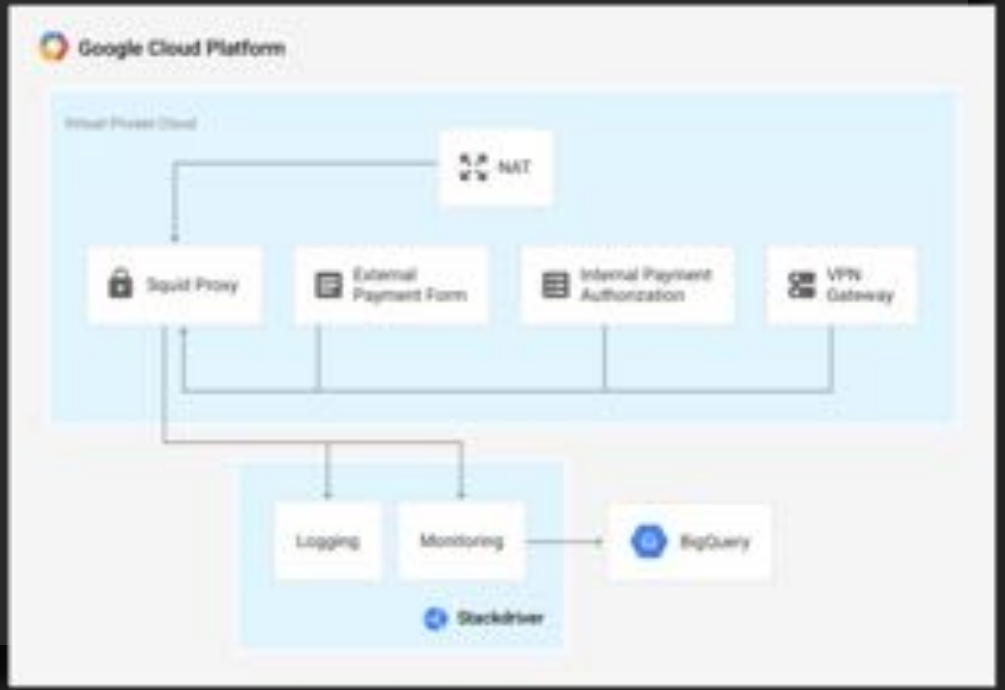
Analyze PCI Data (Credit Card Swipes)

- Payment Card Industry Data Security Standard (PCI DSS)
- Securely handle credit card information
- Exam focus: stream to BigQuery for analysis



Send Log Data to BigQuery for Analysis

- Data travels from Squid Proxy to Stackdriver Logging/Monitoring
- Export from Stackdriver Logging to BigQuery



Securely Migrating Database Data

- Scenario: migrate a database to Datastore. How to authenticate and securely export.
  - Application authentication: authentication application API's with OAuth 2.0
  - Export database info to Cloud Storage → import into Datastore
  - If migrating via application/API = authenticate with OAuth 2.0 with service account and export to GCS
  - If exporting as simply copy = gsutil copy to GCS

Google Certified Professional - Cloud Architect - Part 3

Software Development Lifecycle Concepts

How to Approach Development Topics

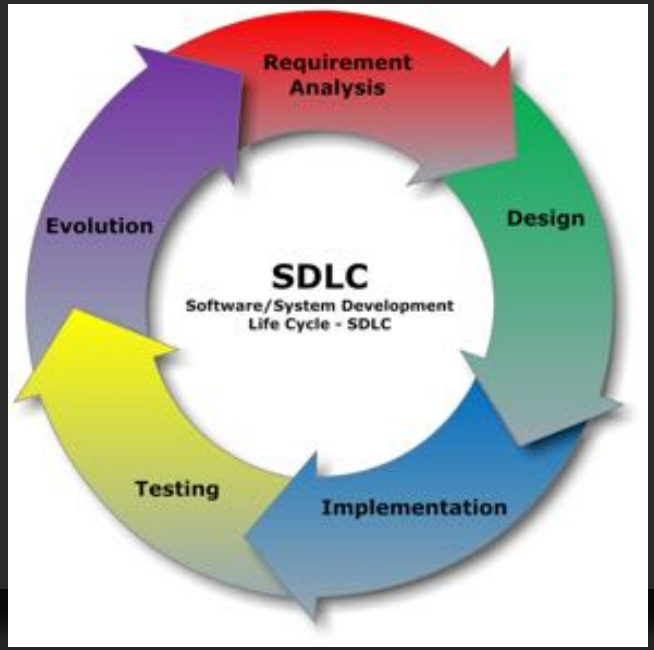
- Different backgrounds
- Not a deep dive
- Match exam's depth

What We'll Cover

- Software Development Lifecycle (SDLC)
- Continuous Integration/Continuous Deployment (CI/CD)
- Blue/green model for deployment
- Application microservices

What is Software Development Life Cycle?

- Produces software with the highest quality and lowest cost in the shortest time
- Plan to develop, alter, maintain, and replace a software system
- Stages of SDLC – not a set list



Why Does this Matter?

- Exam questions assume using SDLC for application development
- Keep environments separate, with different access for different teams
- Same concepts regardless of compute platform (GCE/GKE/GAE)
- Environments in separate projects, with separate levels of access

Continuous Integration/Continuous Deployment (CI/CD)

- Continuous Integration
  - Integrate their code into the main branch of a shared repository early and often
  - Minimize the cost of integration
- Continuous Deployment
  - Focus on automating the software delivery process
  - Automatically deploys each build that passes the full test cycle.
  - No waiting for a human gatekeeper
- GCP Container Builder, Jenkins, Spinnaker

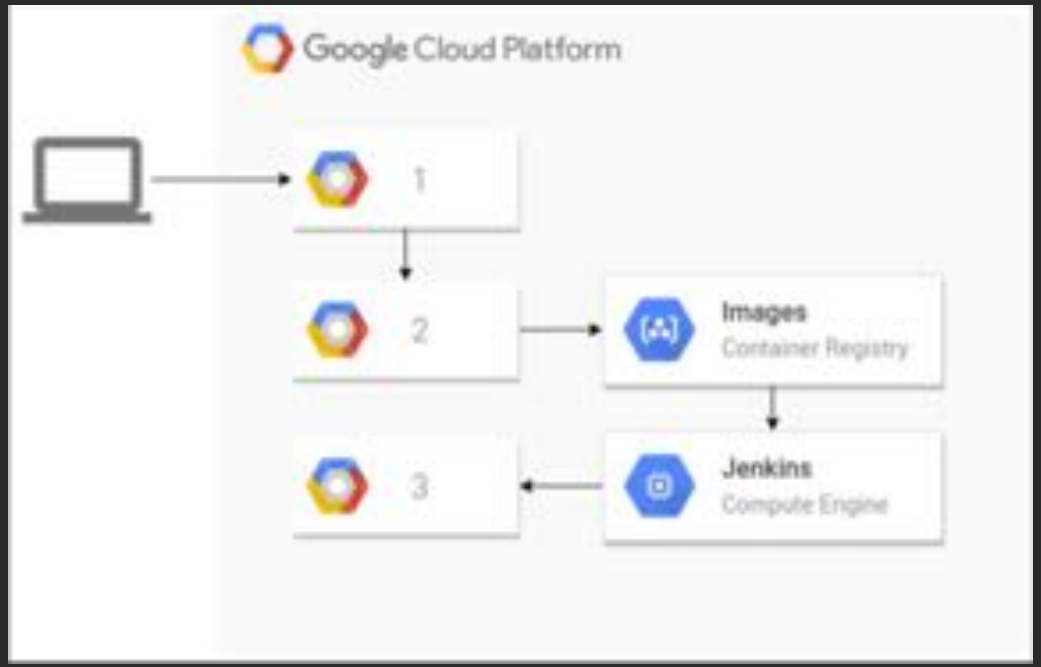
Why this Matters

Basic CI/CD workflow – containers

1 - New code added to Source Repositories

2 - Build container in Container Builder

3 - Run in Kubernetes Engine



Other Deployment Best Practices

- Blue/Green deployment model
- Only one environment is live, while the other goes through SDLC process
- Reduce downtime/risk
    - If something goes wrong in Green, switch back to Blue
- App Engine Versioning/Compute Engine Rolling updater

More Best Practices

- Break monolith application into microservices
- Monolith = all your eggs in one basket
    - Slower development, less flexibility
- Microservices
    - Break single application into smaller pieces
    - Faster deployment, more flexibility

- Exam scenario – reduce unplanned rollbacks due to errors, what best practices?
    - Blue-Green model, break monolith into microservices

Google Certified Professional - Cloud Architect - Part 3

Application Error Examples

Java Digest Error

- Scenario: when updating or compiling Java application, receiving a **digest error**
- Background: JAR files are 'signed' to verify authenticity
- Signed file has digest entry
- Cause: JAR file needs to be re-verified
- Solution: Re-sign JAR file

News Mobile App Caching Under Load

- Need to prevent caching – possible solutions
  - Overwrite Datastore entries
  - Set app to work from single instance
  - Modify API to prevent caching
  - Set HTTP cache flag to -1

- Modifying API depends on exposed function
- Most correct answer is to set HTTP cache flag to -1

Google Certified Professional - Cloud Architect - Part 3

Data Flow – Putting the Pieces Together

Putting the puzzle pieces together

Managing data's lifecycle

Expectation of managing data flow through multiple services to end goal

Big data focus

4 stages

- Ingest – collect your data
- Store – put it somewhere
- Process and Analyze – put your data to work
- Explore and Visualize – view the results – not really tested

The four stages

Ingest

Pull in raw data

From many sources

- devices, in-cloud, on-premises, application logs, mobile apps

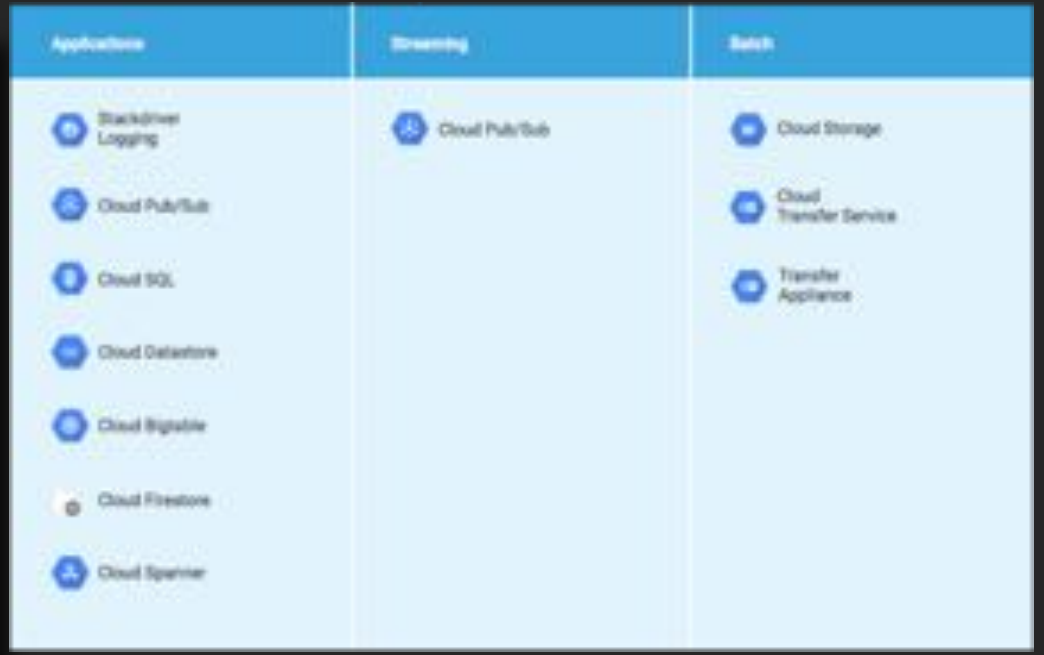
Both streaming and batch

Stream = continuous stream, asynchronous

- Telemetry data, user events, many sources, IoT
- Think Pub/Sub!

Batch = large amounts of stored data

- Transferred in bulk
- Small number of sources



More on Pub/Sub

Publish and subscribe to messages (hence the name)

Real-time messaging

Again, stream = Pub/Sub, Pub/Sub = stream

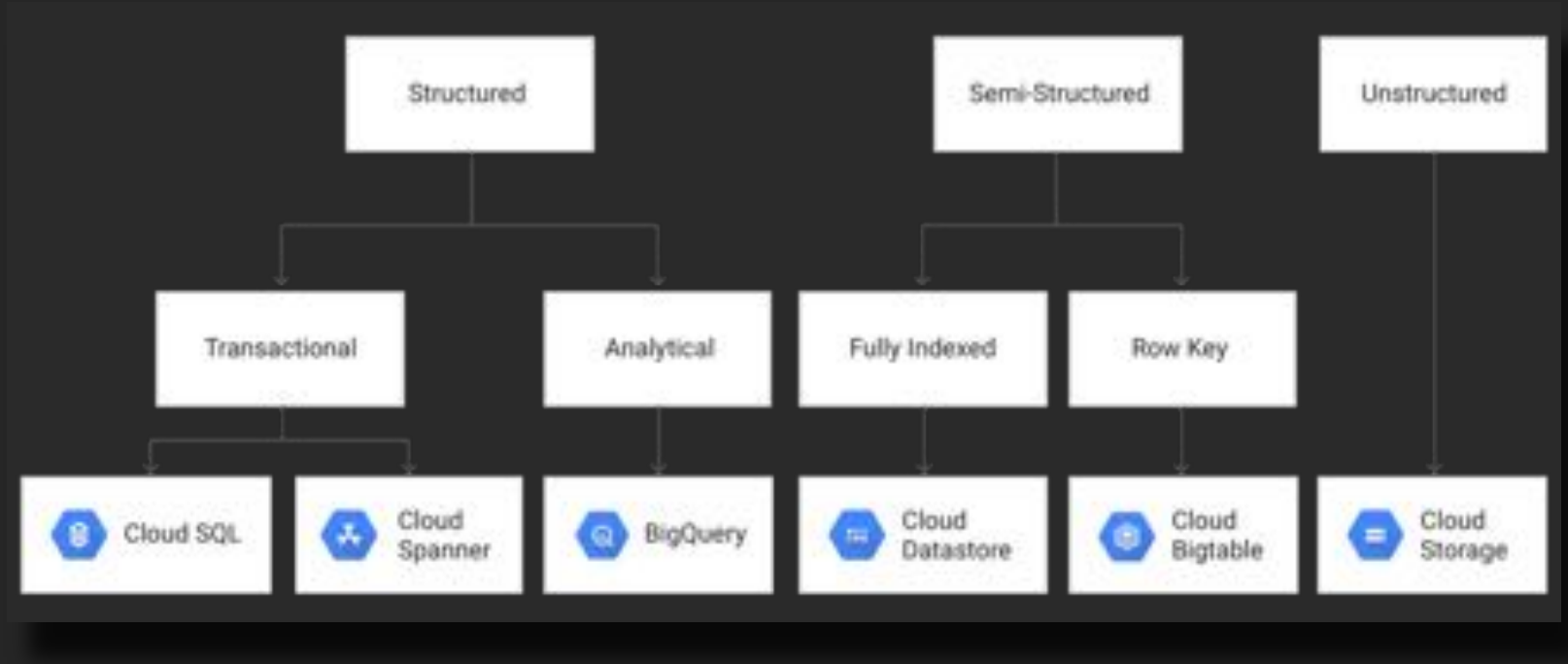
Buffer, or 'shock absorber' for ingesting from huge number of sources

"Millions of devices? Bring it on!"

Does not guarantee order of delivery

- Strict ordering – buffering w/ Dataflow

Storing data = mapping storage solutions



Processing and Analyzing

Cloud Dataproc

- Existing Hadoop/Spark Applications
- Machine Learning / Data Science Ecosystem
- Tunable Cluster Parameters

Cloud Dataflow

- New Data Processing Pipelines
- Unified Streaming & Batch
- Fully-Managed, No-Ops

Cloud Dataprep

- UI-Driven Data Preparation
- Scales On-Demand
- Fully-Managed, No-Ops

Google Certified Professional - Cloud Architect - Part 3

Next Steps

Next steps

Take, and PASS the exam!

Last minute study materials:

Google practice exam

Course practice exam

Study guide materials

Google Cloud YouTube channel - https://www.youtube.com/user/googlecloudplatform

Google Cloud blog - https://cloudplatform.googleblog.com/

Questions? Let us know!

Post in the community of your success!

How did we do? Rate us!