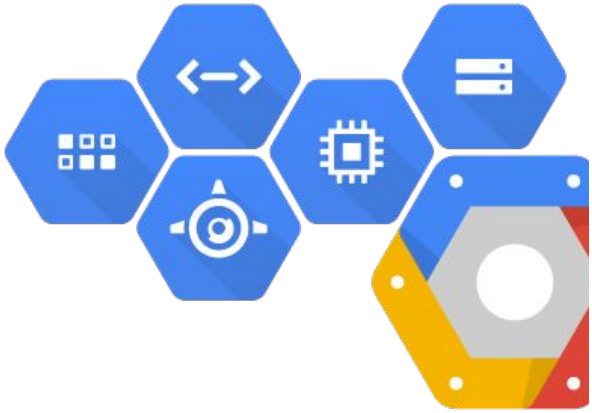
Google Cloud

Day 2: Using GCP

Google Cloud Infrastructure
2-Day Training



April 2019

Sunnyvale, CA

Welcome to Google Cloud Platform Training!

This 2-day training is designed to provide an overview to help clients adopt Google Cloud Platform.

Day 1

GCP Foundational Architecture

Day 1 is focused on setting up, and configuring GCP including key concepts, terminology, and exercises

- Cloud Identity
- Identity and Access Management
- Network Configuration
- Monitoring And Logging
- Automated Operations
- Billing

Day 2

Using GCP

Day 2 is focused on moving (and developing) client application workloads into GCP including key architectures and technologies

- GKE
- Customer Use Cases
- Path to GCP Certification
- Deconstructing an Architecture Case Study

Expectations & Rules of the Road

- Assume You Have Some Basic Knowledge about Cloud And GCP Infrastructure Products and Offerings
- Please Limit Distractions (e.g., Email) And Leaving The Room
- Ask Questions and Engage!

Google Cloud

Day 2 Agenda

Topic
Day 1 Recap
Working with GCP
Introduction to Kubernetes
Project Flex Overview
Customer Case #1
Path to GCP Certification
GCP Certification Process
Deconstructing an Architecture Case Study
JencoMart
Dress4win
Mountkirk Games
TerramEarth

Google Cloud

Day 1 Recap

01

Day 1 Recap

Cloud Identity

Cloud Identity is a scalable, secured, and reliable IDaaS solution for GCP user identities

Resolve conflict accounts before user provisioning

While an organization has several options for provisioning G Suite user accounts, it is recommended to use Google Cloud Directory Sync

Centralize authentication including multi-factor either through Google or a third party Identity Provider

Identity and Access Management

Utilize user groups to manage IAM roles to avoid the need to manage roles for individual user accounts

Service accounts authenticate applications to other GCP services and can be treated either as an identity or as a resource

Users, service accounts, and groups can be assigned IAM roles at a project level and through these roles can be granted access to resources

Google Cloud

Day 1 Recap

Networking

A single VPC can span multiple regions and can be shared across an entire organization

Firewalls provide flexible grouping mechanism for instances by using tags

Load balancing is a managed service that is software-defined and globally distributed

GCP offers three types of connectivity options with on-prem infrastructure - Direct Peering, Carrier Interconnect and Private Interconnect

Monitoring and Logging

Stackdriver can monitor the health of your applications on GCP and/or AWS

Always create a Stackdriver account in a separate project to monitor resources in other projects

Export logs to BigQuery for advanced analysis and long term storage

3rd party tools such as, PagerDuty and Splunk, can be integrated with Stackdriver to meet additional requirements

Google Cloud

Day 1 Recap

Automated Operations

Continuous integration and continuous delivery tools that can be used and how it can be integrated with GCP

Use Deployment Manager to programmatically provision GCP resources

Always use version control software to manage source code, config files and test scripts

Monitor build and test results for failures and leverage metrics for continuous improvement

Billing

Use billing labels to customize reports based on organizational needs and accounting strategy

Data from billing reports can be exported to analytics tools in order to analyze spending patterns and predict future spending

Configure monthly thresholds for billing accounts as a monitoring and cost control mechanism

Use predefined billing roles to manage billing, reporting and budgeting

Google Cloud

Working with GCP

Introduction to Kubernetes

Introduction to Kubernetes

This section will focus on the following key topics.

Objectives

- High level overview of Kubernetes fundamentals
- Key considerations for designing and architecting a Kubernetes infrastructure

Key Learnings

- Understanding high level concepts related to Containers, Dockers, and Kubernetes.
- OVerview of architecture components and design principles
- Learn about kubernetes resource types

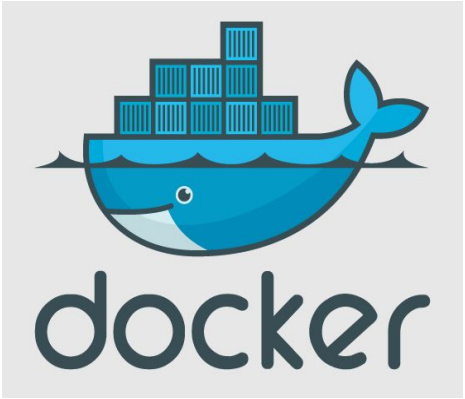
Google Cloud

What are Containers?

Containers are a method of packaging an application executable and its dependencies (runtime, system tools, system libraries, configuration), and running the package as a set of resource-isolated processes.

In simple terms

- Container: Process set in cgroup/chroot jail
- Container image: Executable + (OS libs – kernel) + all other dependencies
- Container registry (Docker/GCR/Quay): Central image/binary repository

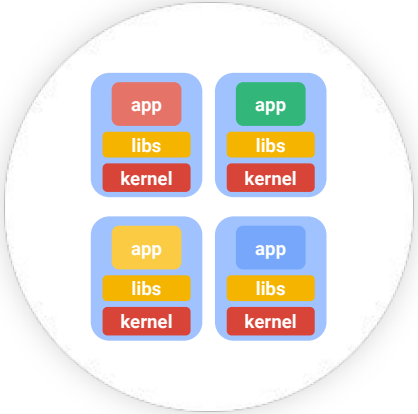


Google Cloud

Comparison with Virtualization and Shared Hosts

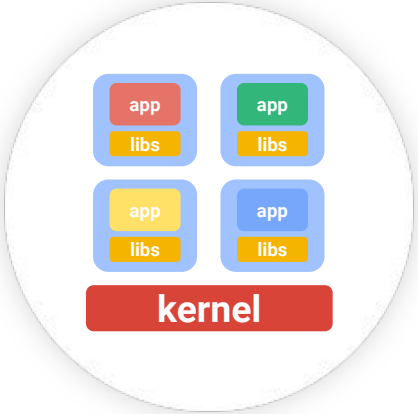
Shared machines

- No isolation
- Common libs
- Highly coupled apps and OS



VMs/Bare metal

- Isolation
- No common libs
- Expensive, inefficient
- Hard to manage



Containers

- Isolation
- No common libs
- Less overhead
- Less dependency on host OS

Google Cloud

K8s provides Container-centric Infrastructure

Once specific containers are no longer bound to specific machines/VMs, host-centric infrastructure no longer works.

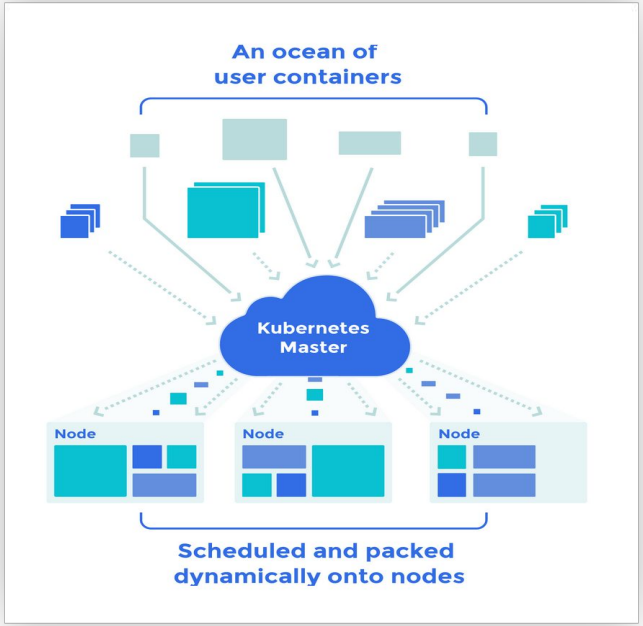
- **Scheduling:** Decide where my containers should run
- **Lifecycle and health:** Keep my containers running despite failures
- **Scaling:** Make sets of containers bigger or smaller
- **Naming and discovery:** Find where my containers are now
- **Load balancing:** Distribute traffic across a set of containers
- **Storage volumes:** Provide data to containers
- **Logging and monitoring:** Track what's happening with my containers
- **Debugging and introspection:** Enter or attach to containers
- **Identity and authorization:** Control who can do things to my containers

In simple terms...

Think of Kubernetes as the OS for your compute fleet.

It provides features similar to an OS for a host:

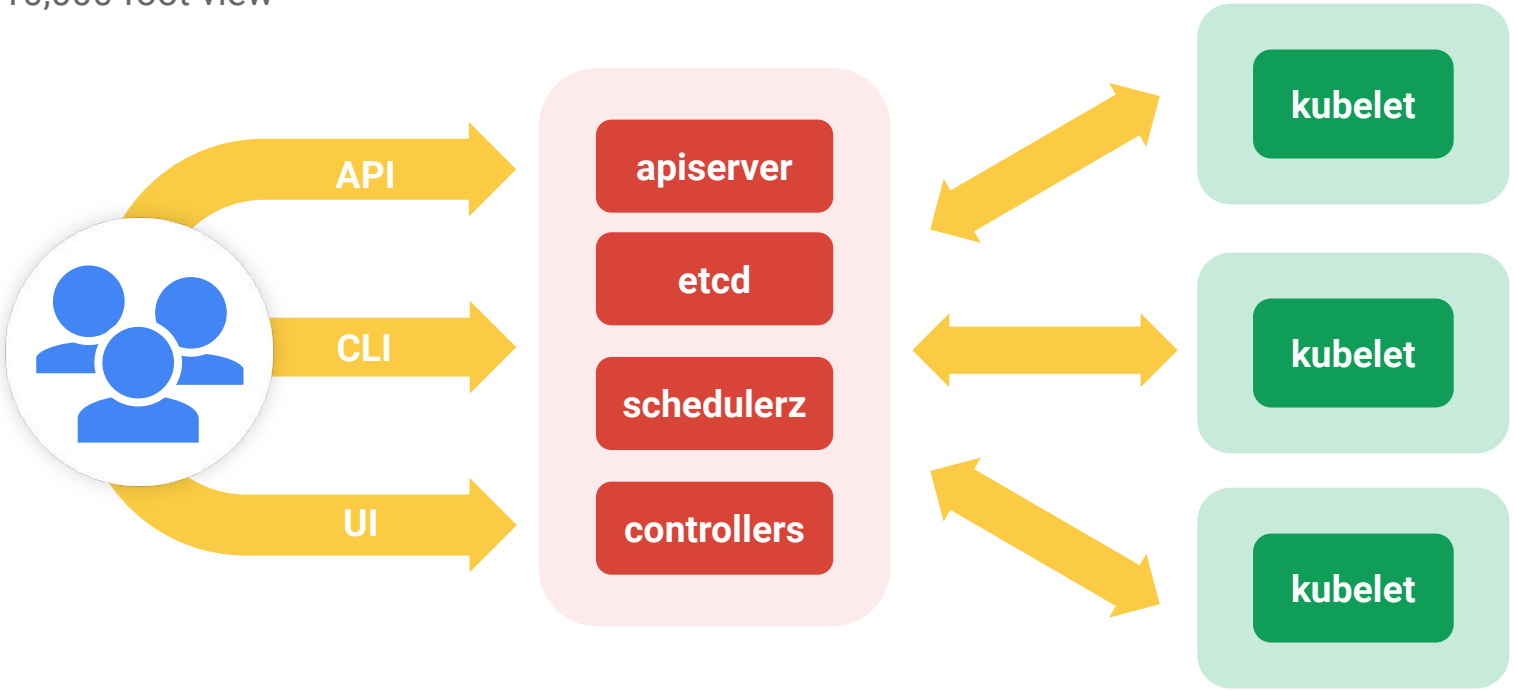
- Scheduling workload
- Finding the right host to fit your workload
- Monitoring health of the workload
- Scaling it up and down as needed
- Moving it around as needed



Google Cloud

Architecture and Design Principles

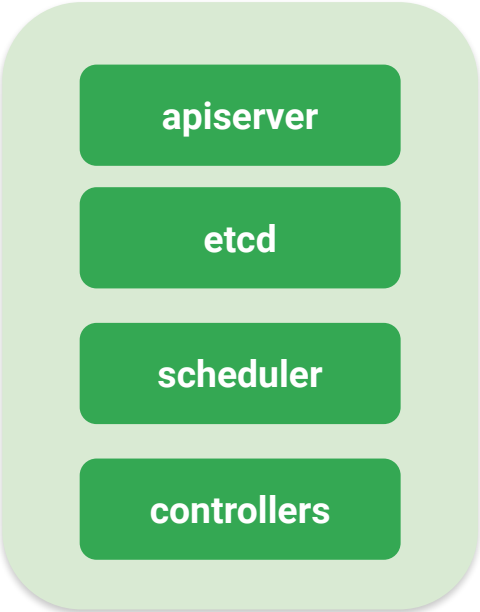
The 10,000 foot view



Google Cloud

Master Components

- API server (kube-apiserver)
- etcd - reliable distributed key-value store
- Scheduler (kube-scheduler)
- Controllers
  - Kube controller (kube-controller-manager)
    - Replication controller
    - Endpoints controller
    - Service account and token controllers
  - Cloud controller (cloud-controller-manager)
- Add-ons
  - Kube DNS (kube-dns)
  - Web UI (dashboard)
  - Container resource
  - Monitoring
  - Cluster-level logging



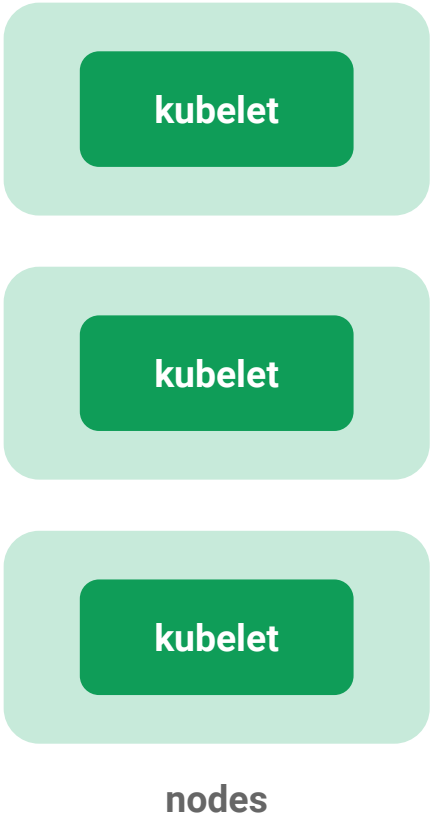
master /control plane

Google Cloud

Node Components

- Kubelet (kubelet)
- Kube proxy (kube-proxy)
- Container runtime
  - Docker (containerd)
- Monitoring/Logging
  - Supervisord
  - Fluentd

Clients use Kube Control (kubectl) CLI to interact with the cluster



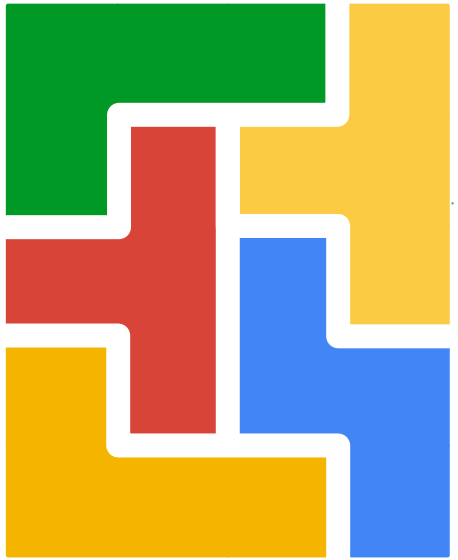
The 10,000 foot view

- API server (via etcd)
  - Single source of truth
  - Provides the ability to watch for events
- Everything is a resource (aka object)
  - kind
  - apiVersion
  - metadata
  - spec ← represents desired state
  - status ← represents current state
- Controllers
  - Respond to changes in objects
  - Move current state toward desired state

```
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
```

Google Cloud

Modularity



- Loose coupling is a goal everywhere
  - Simpler
  - Composable
  - Extensible
- Code-level plugins where possible
- Multi-process where possible
- Isolate risk by interchangeable parts

Examples: Scheduler, IngressController, Container Runtime, ThirdPartyResources

Google Cloud

Control Loops

- Drive **current state → desired state**
- Act independently
- APIs — no **shortcuts** or back doors
- Observed state is truth
- Recurring pattern in the system

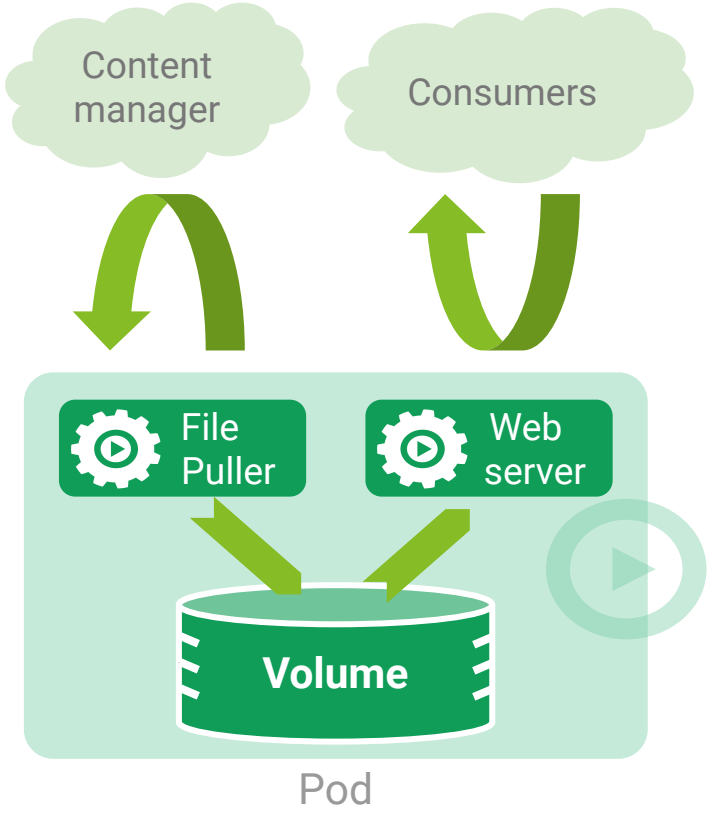
Examples: ReplicationController, IngressController

Google Cloud

Kubernetes Resource Types

Pods

- **Small group** of containers and volumes
- **Tightly coupled**
- The atom of scheduling and placement
- Shared namespace
  - Share IP address and localhost
  - Share IPC, etc.
- Managed lifecycle
  - Bound to a node, restart in place
  - Can die, cannot be reborn with same ID

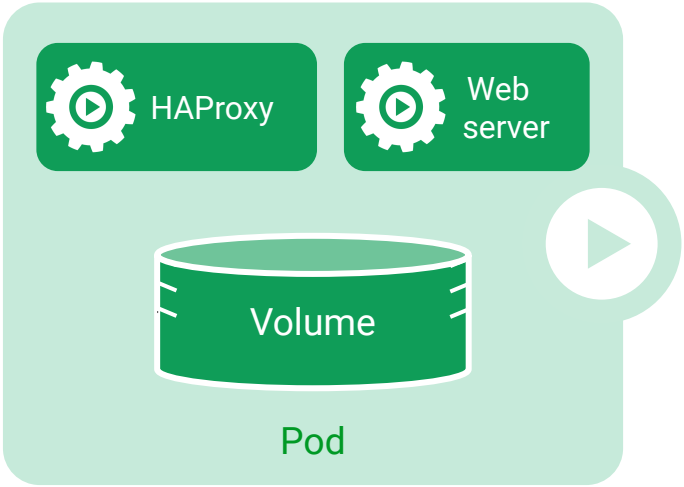


Google Cloud

Kubernetes Resource Types

Pods - in simple terms, think of...

- Pods as VMs of the container world
- Containers in a Pod as processes on a VM
- Volumes as disks on a VM
- VMs have individual IPs → So do pods
- VMs run processes → Pods run containers
- Processes in a VM
  - Share the host's IP address
  - Need to coordinate port allocation
  - Share the disk



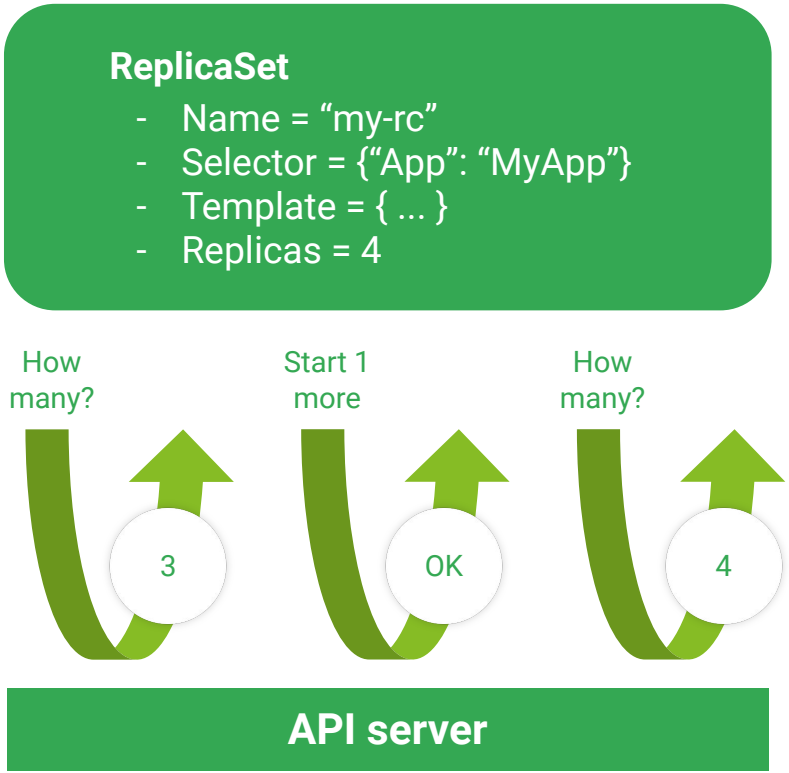
Do not confuse pods with nodes in a cluster
Nodes are the VMs where pods run

Google Cloud

Kubernetes Resource Types

Replication - ReplicaSets*

- A simple control loop
- Runs out-of-process wrt API server
- **One job:** Ensure N copies of a pod
  - Grouped by a selector
  - Too few? Start some
  - Too many? Kill some
- Layered on top of the public pod API
- Replicated pods are **fungible**
- No implied order or identity



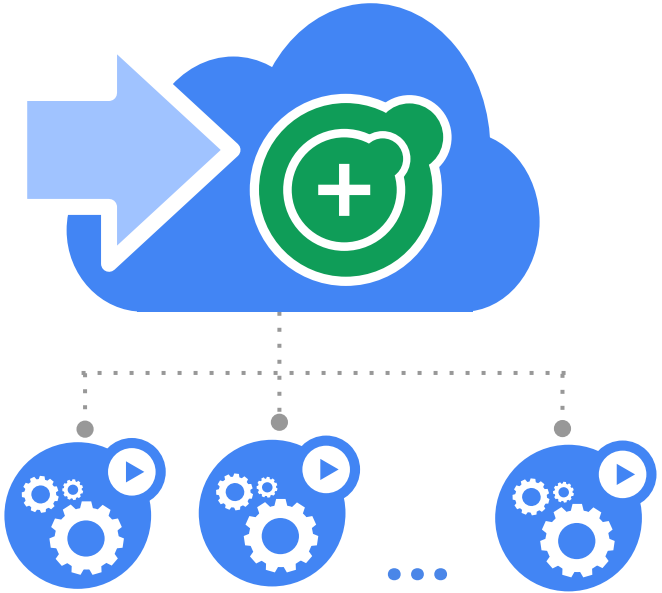
Google Cloud

Kubernetes Resource Types

Replication - Deployments

Updates-as-a-Service

- Rolling update is imperative, client-side
- Deployment manages replica changes for you
  - Stable object name
  - Updates are configurable, done server-side
  - kubectl edit or kubectl apply
- Aggregates stats
- Can have multiple updates in flight



Google Cloud

Kubernetes Resource Types

Other **replication** controllers provide different semantics for workflow scheduling

- ReplicaSet → Maintain specified replicas
- Deployment → Support for container updates
- Job → Run once
- Cron Job → Run once based on schedule
- DaemonSet → Run exactly one instance on each node
- StatefulSet → Stable names for pods and preset build and tear-down order

Kubernetes Resource Types

Labels

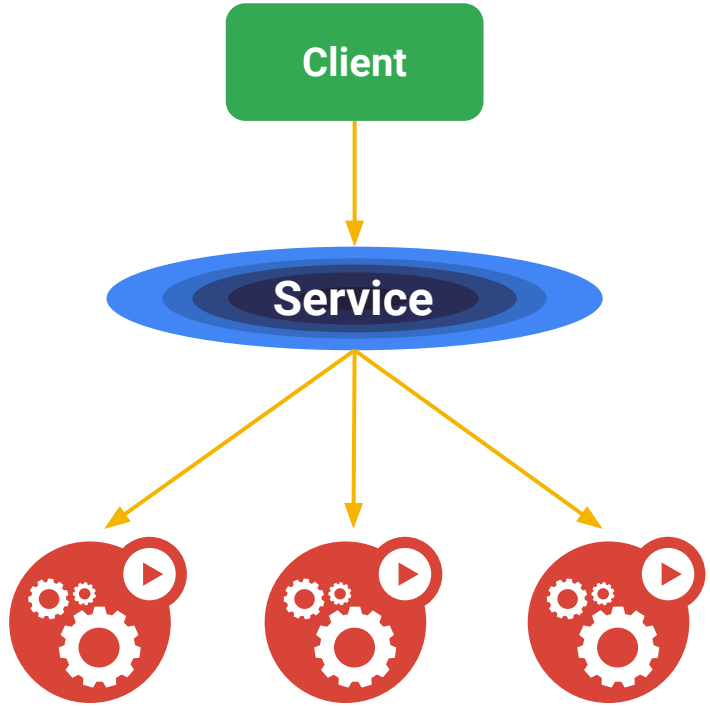
- Arbitrary metadata
- Attached to any API object
- Generally represent **identity**
- Queryable by **selectors**
  - Think SQL '*select ... where ...*'
- The **only** grouping mechanism
  - Pods under a ReplicationController
  - Pods in a Service
  - Capabilities of a node (constraints)

Google Cloud

Kubernetes Resource Types

Services

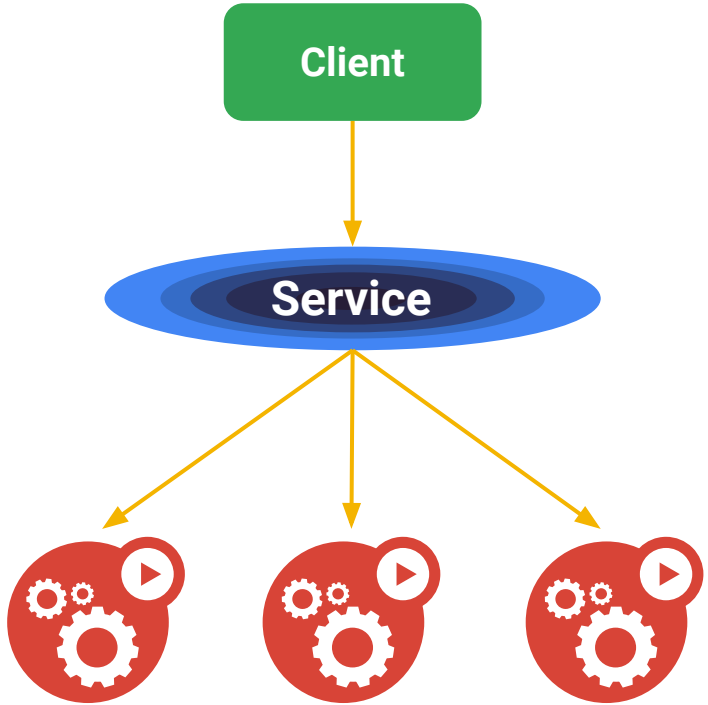
- A group of pods that work together
  - Grouped by a selector
- Defines access policy
  - “Load balanced” or “headless”
- Can have a stable virtual IP and port
  - Also a DNS name
- VIP is managed by kube-proxy
  - Watches all services
  - Updates iptables when backends change
  - Default implementation — can be replaced!
- Hides complexity



Kubernetes Resource Types

Services - Service types

- ClusterIP
  - Within the VPC (network) where the cluster exists
- NodePort
  - Within the VPC (network)
  - Wherever the nodes are accessible from
- LoadBalancer
  - External using load balancer with a public IP
- Think of these as concentric
  - ClusterIP is at the center
  - NodePort includes ClusterIP
  - LoadBalancer includes NodePort and ClusterIP



Google Cloud

Kubernetes Resource Types

External Services

- Services VIPs are only available inside the VPC
- Need to receive traffic from “the outside world”
- Service “type”
  - NodePort: expose on a port on every node
  - LoadBalancer: provision a cloud load-balancer *(Requires integration with a cloud provider)*
- DiY load-balancer solutions
  - Haproxy
  - nginx
- Ingress (L7 LB)

Google Cloud

Recap

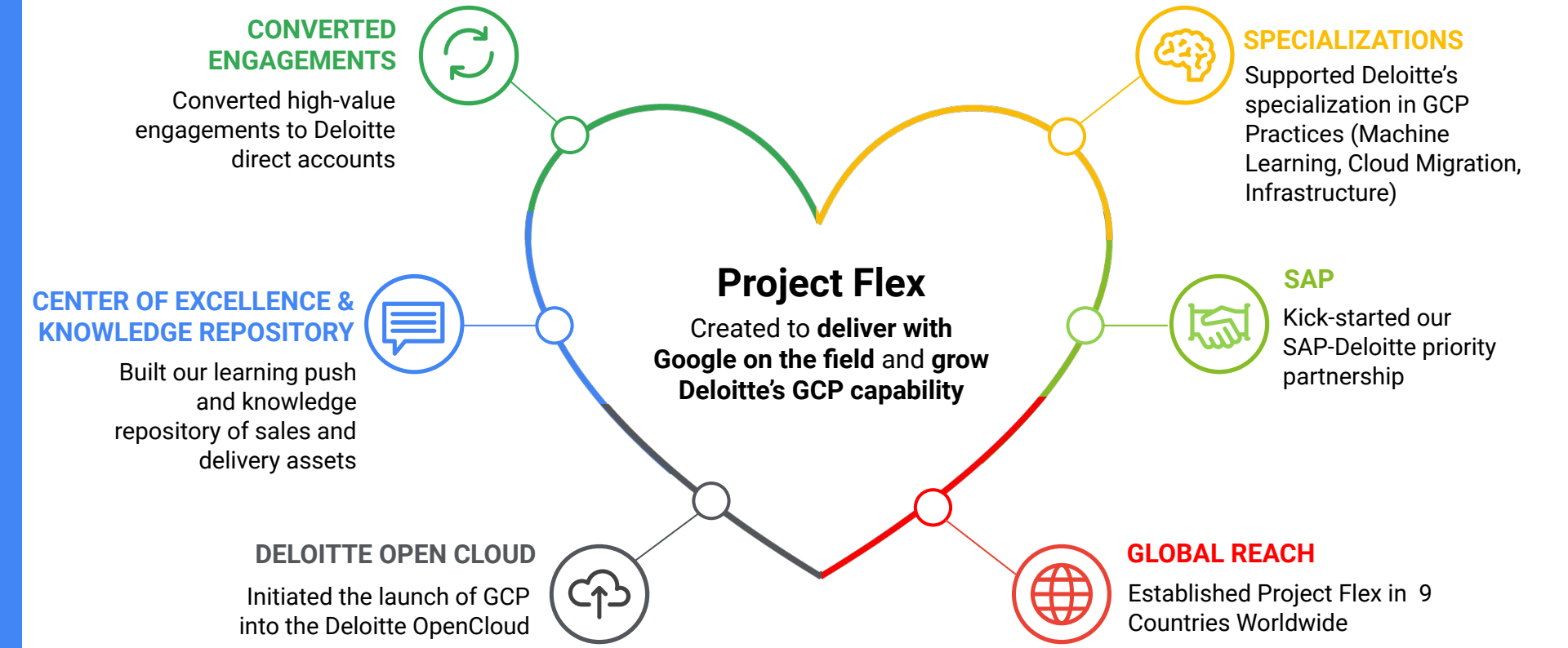
- **Container:** A sealed application package that can be run in isolation
- **Pod:** A small group of tightly coupled containers. *Example: content syncer and web server*
- **Controller:** A loop that drives current state toward desired state. *Example: replication controller*
- **Service:** A set of running pods that work together. *Example: load-balanced backends*
- **Labels:** Identifying metadata attached to other objects. *Example: phase=canary vs. phase=prod*
- **Selector:** A query against labels, producing a set result. *Example: all pods where label phase == prod*

Google Cloud

Project Flex Overview

03

Project Flex - The “Heart” of our Cloud Relationship



Google Cloud

Project Flex our US and Global Delivery

Flex US | Established January 2017

41 | Projects Delivered

- **04** Data & Analytics
- **26** Infrastructure
- **07** Machine Learning

5 | Active Projects

- **02** Data & Analytics
- **03** Infrastructure
- **00** Machine Learning

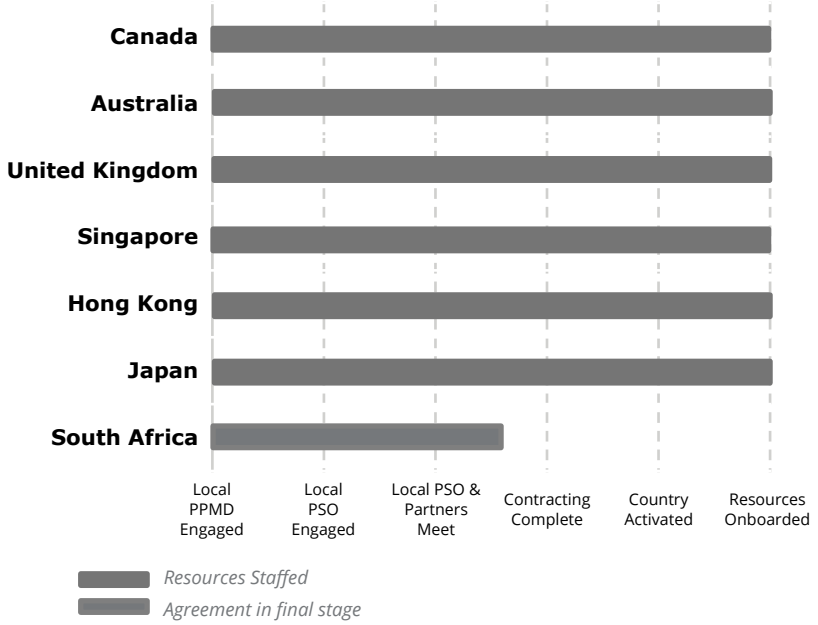
20 | Unique Clients

7 | Distinct Industries

Offering Enhancement & Sales Enablement

- Enhanced the Google Cloud Secure offering
- Developing Kubernetes assets for GKE
- Deployed Flex Support for SAP delivery
- Developing a Netezza to GCP Migration acceleration playbook

Flex International | Established March 2018



Google Cloud

Deloitte and Google Cloud

The cloud journey ahead is dynamic, fast moving, and full of competitive advantages. **Capitalize on it.** At Deloitte, we help you explore new possibilities, navigate obstacles and opportunities, align your virtual goals with real results, and provide industry-specific solutions and experiences to unlock value every step of the way. As a premier partner of Google Cloud and 2017 Global Services Partner of the Year, we help you get to the cloud fast—and at scale. That means extended presence at the drop of a dime, agile capacity that can anticipate and react to business change quickly, and a new competitive landscape waiting to be discovered.

Application Services
PaaS and IaaS strategy, implementation and beyond.

As our clients increasingly leverage cloud native services to modernize their existing application portfolios and build new and innovative products and services for their customers, they turn to us to help drive increased value through Google Cloud.

Premier Partner
Google Cloud
Cloud Migration

Analytics
Focus on insights, not infrastructure.

As a recognized global leader in business analytics and strategy, Deloitte is able to help clients tap into Google Cloud's array of big data processing and analytics capabilities to enable data-driven insights and help solve complex business challenges.

Machine Learning
Unlocked and empowered.

Deloitte's strong analytics, IoT, and cognitive computing practices combine with Google's managed, scalable machine learning engine to provide clients with advanced automation and machine learning capabilities.

Premier Partner
Google Cloud
Machine Learning

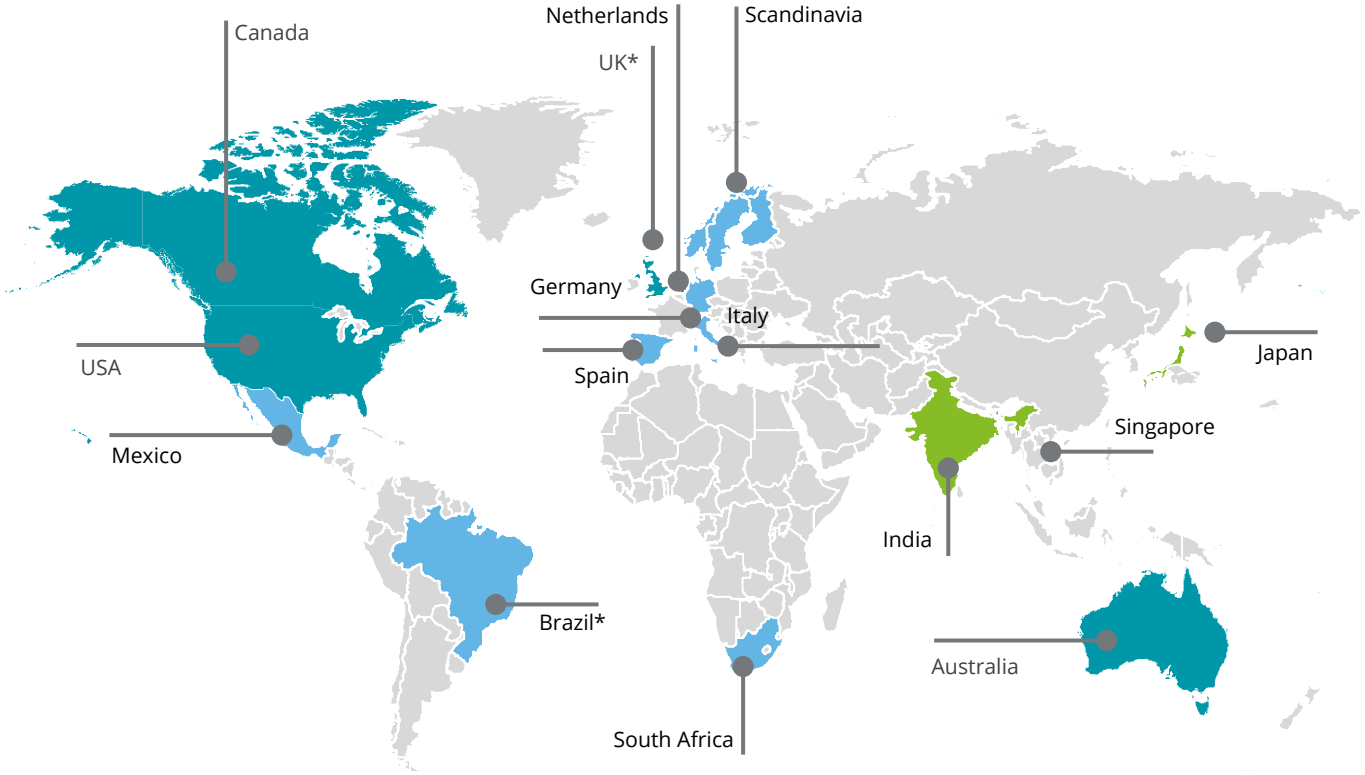
SAP on Google Cloud
Intelligent Enterprise. Delivered.

Through our alliance with SAP, Deloitte can bring the people, knowledge, and experience to help you capture the transformative potential of SAP running on Google's fast, reliable, global platform.

Security
Security at the core.

Google Cloud's infrastructure is designed, built and operated with a rigorous attention to security. Deloitte has spent decades helping clients protect their businesses from security threats. Together, we can help you move securely to the cloud.

Delivering Impact Globally



- Alliance Agreement + Cardinal Program
- Cardinal Program
- Alliance Agreement In Process

1,200+

Practitioners devoted to the Google/Alphabet Practice

1000+

Google Cloud Trained Practitioners

Google Cloud

*Additional Cardinal activation in EMEA managed through UK
*Additional Cardinal activation in LATAM managed through Brazil

Achieving Global Accolades and Certifications

Premier Partner

Google Cloud

Preferred Partnership

Deloitte. Google Cloud SAP

Preferred Partnership

Deloitte. Google Cloud salesforce

Machine Learning
Specialization

Advanced Solutions Lab
Preferred Partner

Trusted Tester
Partner

Cloud Migration
Specialization

Google Marketing Platform
Certified

Google Marketing Platform
Sales Partner

2017 GOOGLE CLOUD
GLOBAL SERVICES
PARTNER OF
THE YEAR

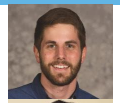
Flex Team Members

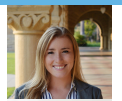
Infrastructure Experts

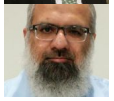

Matt Devine

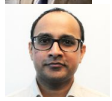

Alharith Hussin


Nitin Ashok


David Wright


Kristen DeStefano


Farooq Ashraf


Michael Liedike


Vaibhav Dhingra

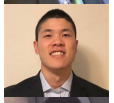

Shashank Joshi

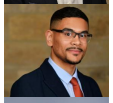

Milind Patel

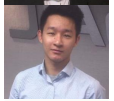
CSP Pod


Eric Procopio


Cole Whitley


Daniel Lu


Cristian Caraballo

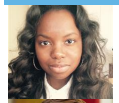

Daniel He

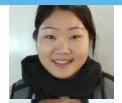

Harshil Shah


Sandeep Raveesh Babu


Trey Nguyen

Data Analytics & Machine Learning Experts


Aramide Kehinde


Jinxu Gao


Konstantinos Vasilakakis


Ankit Gupta


Matt Iames


Rajeshwar Mann

2017 GOOGLE CLOUD

GLOBAL SERVICES

PARTNER OF THE YEAR

Deloitte.

Google Cloud

Premier Partner

Customer Case #1

Customer Case 1

Problem Statement

- The client is a **North American home retailer** with multiple brands in domestics merchandise, home furnishings, and juvenile merchandise
- It has over **1,500 stores** as well as **eCommerce sites** serving **millions of customers** per day
- Client is experiencing **inflated infrastructure costs** and **customer frustration due to slow website performance**

Client Objectives

- Client wants to adopt a **public cloud-based solution to host its redesigned eCommerce platform** in order to **expand online operations**, **improve website performance** for key business areas and better **meet customer demands** while **reducing infrastructure costs**.
- The client would like to break its monolithic application into **decoupled microservices**

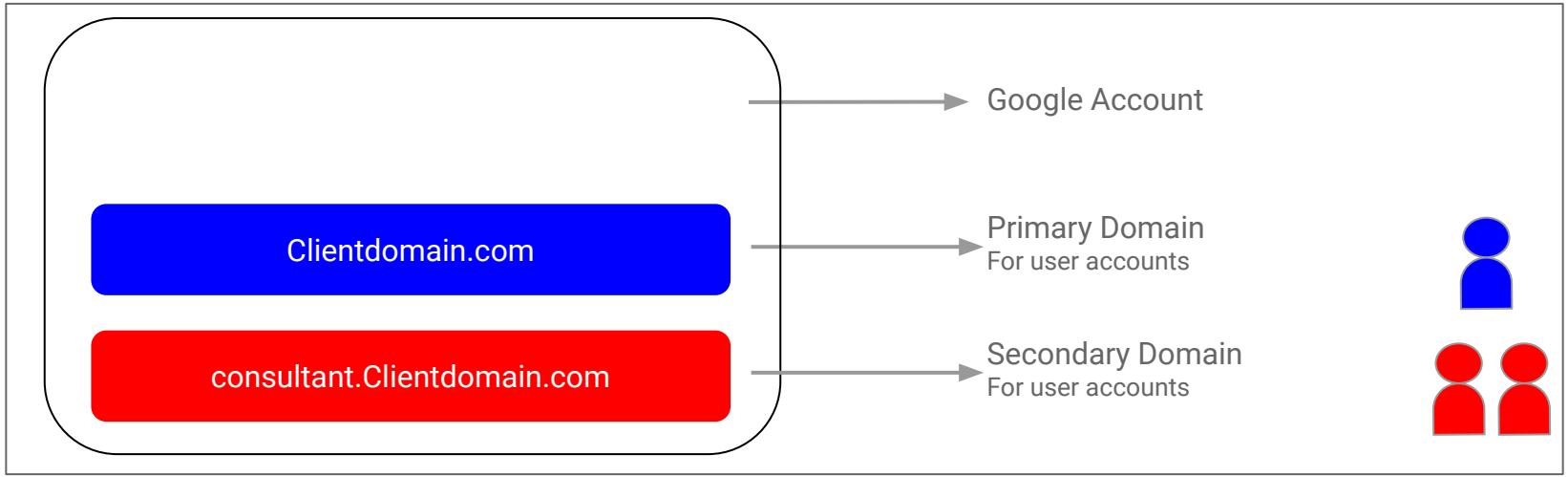
Google Cloud

Detailed Solution

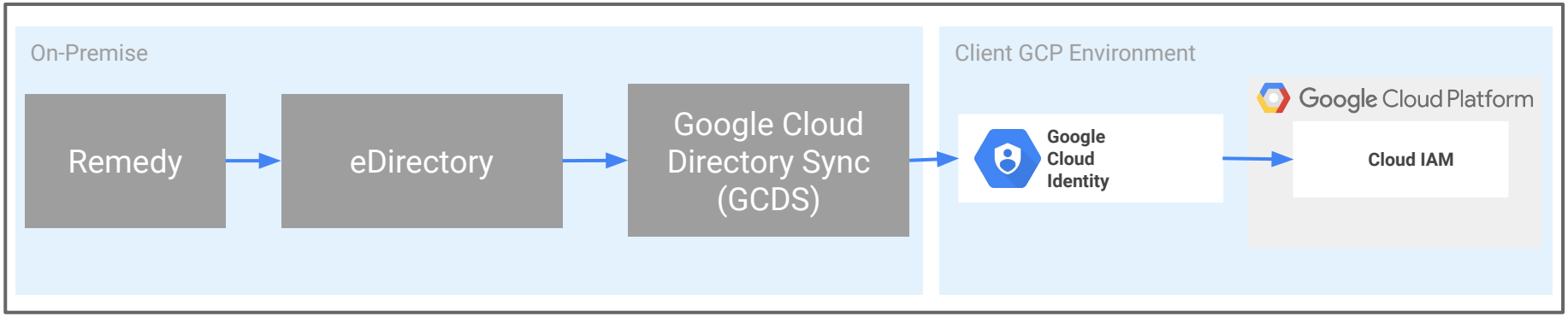
1. Client Account and Domain Setup for GCP
2. Client GCP User Provisioning Process Diagram
3. Client IAM Roles and Groups
4. Client Organizational Folder Structure
5. Client Network Architecture
6. Cloud VPN
7. Shared VPC Overview
8. Client Logging Design
9. Application Architecture

Google Cloud

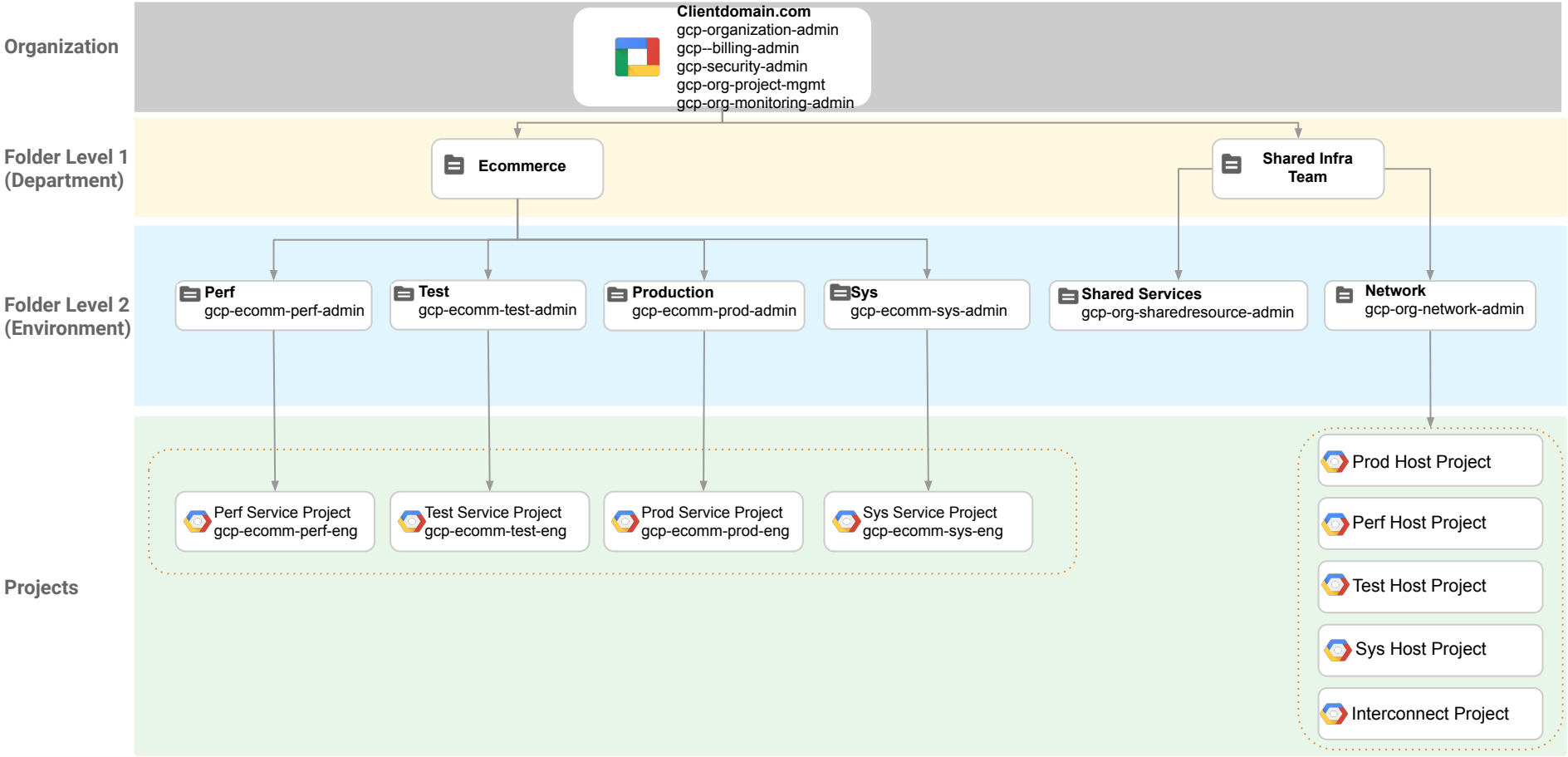
Client Account and Domain Setup for GCP



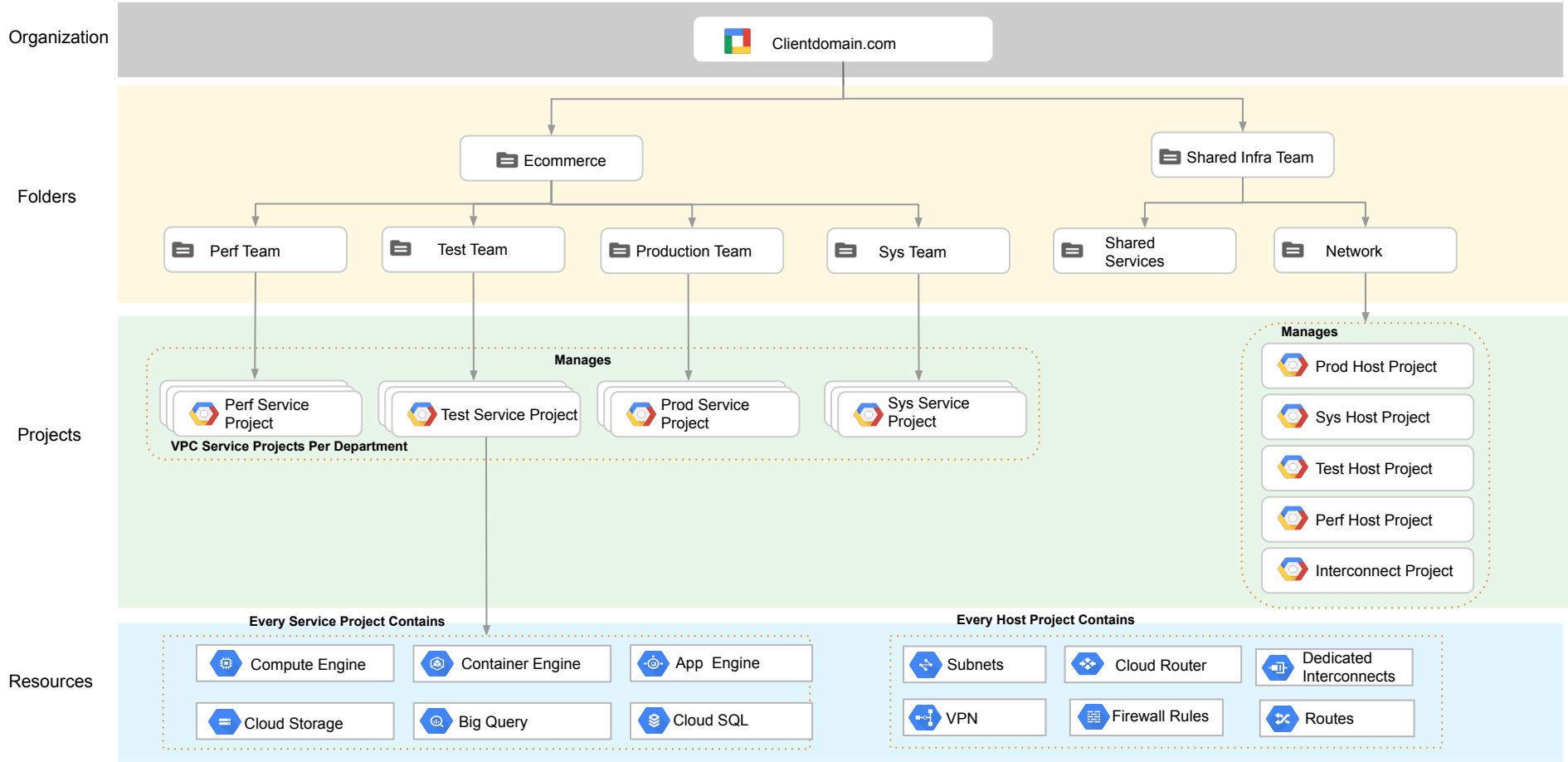
Client GCP User Provisioning Process Diagram



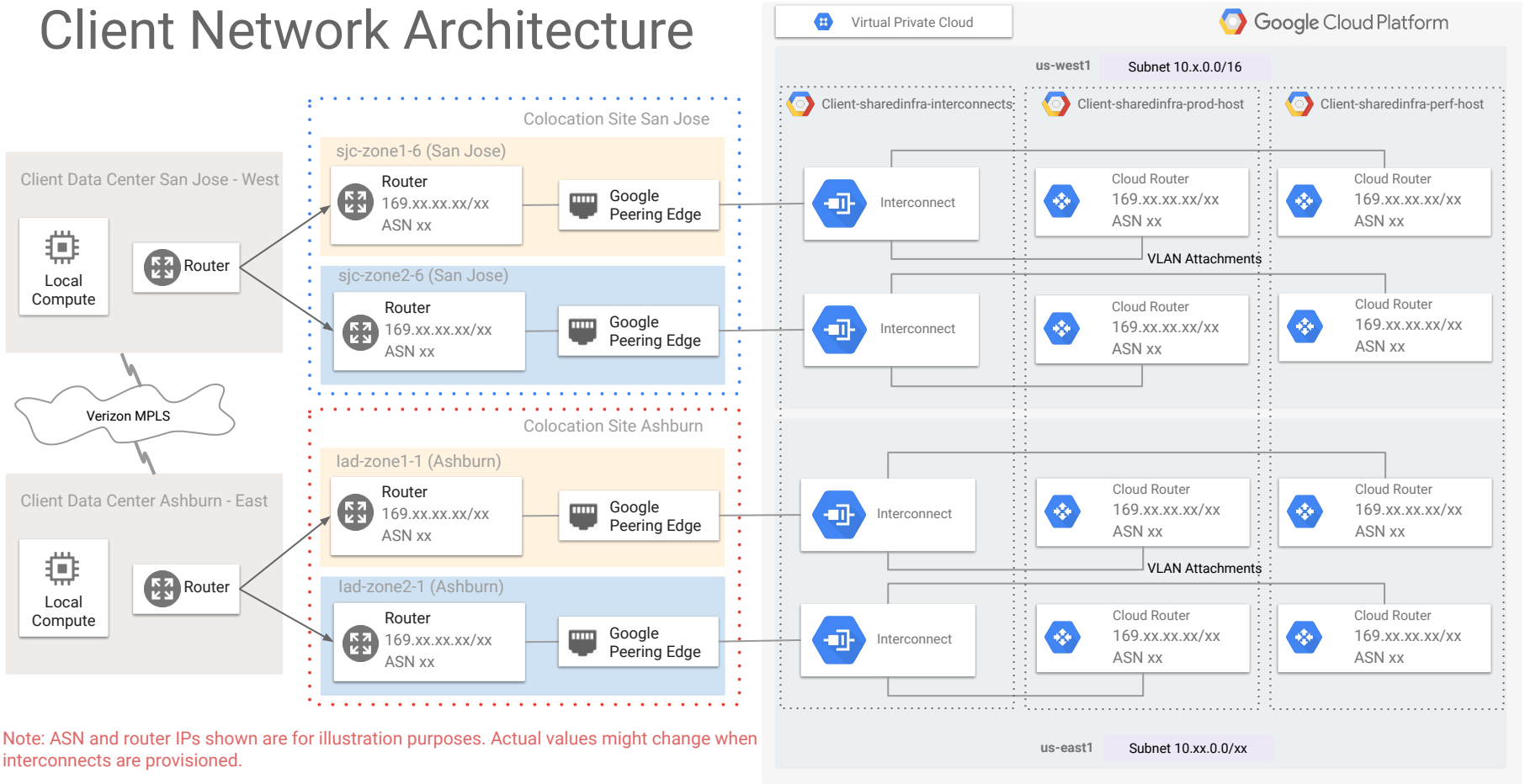
IAM Roles and Groups



Client Organizational Folder Structure



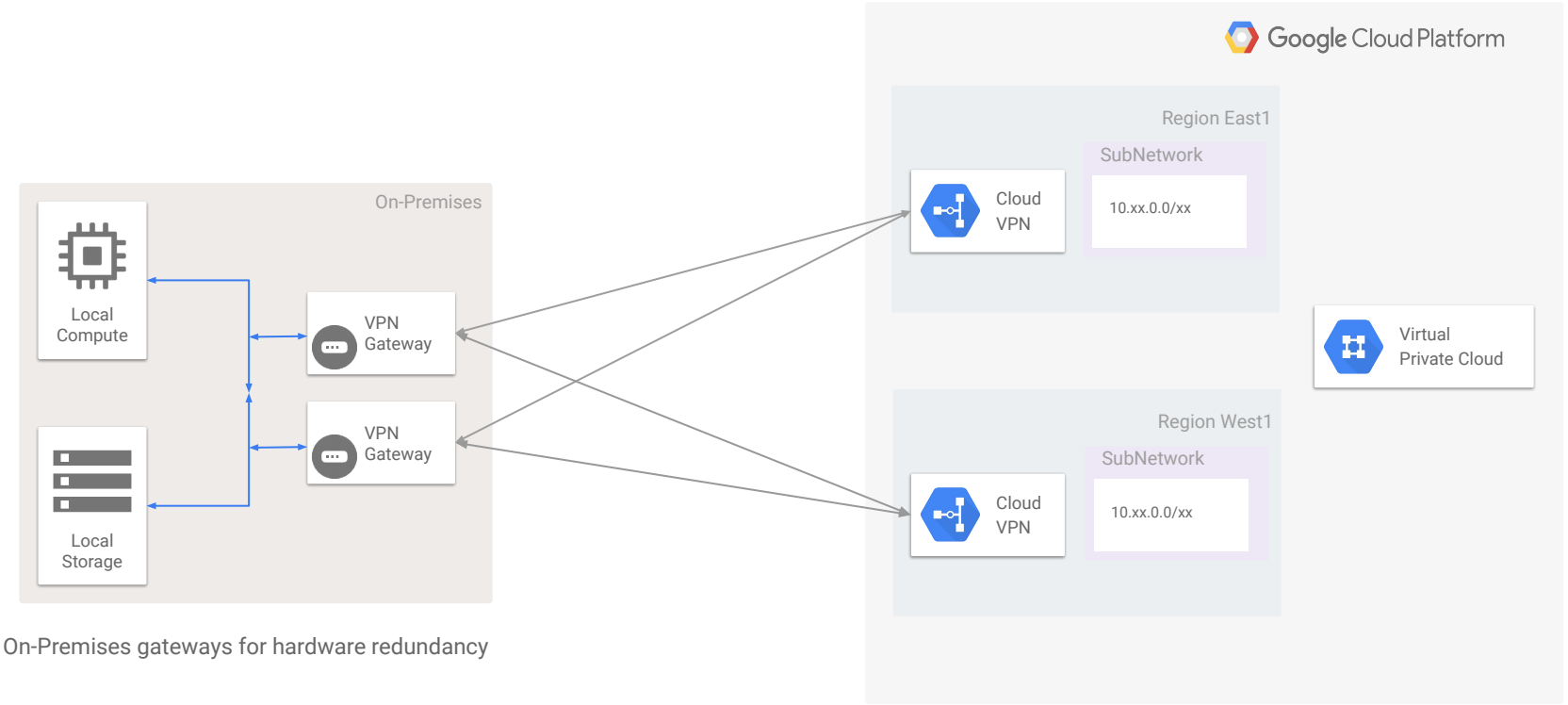
Client Network Architecture



Note: ASN and router IPs shown are for illustration purposes. Actual values might change when interconnects are provisioned.

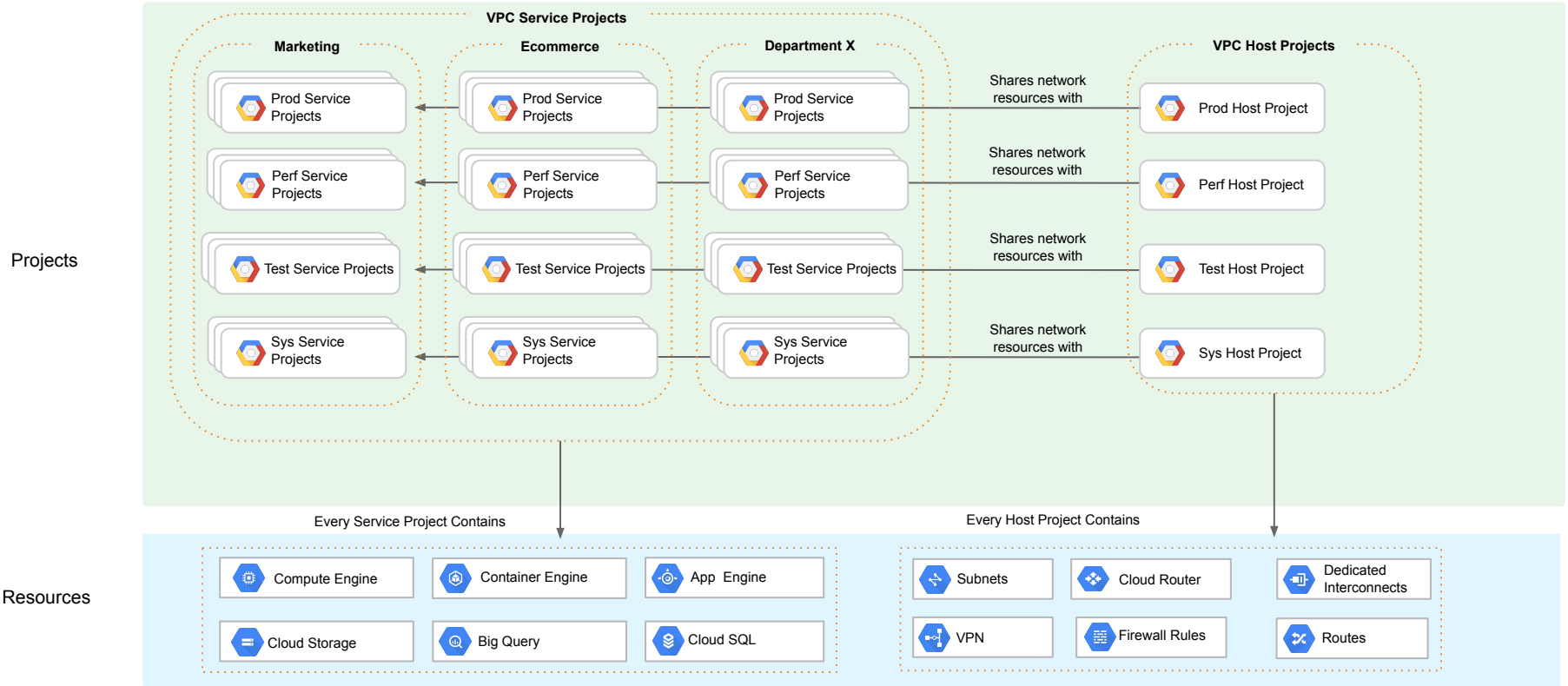
Google Cloud

Cloud VPN

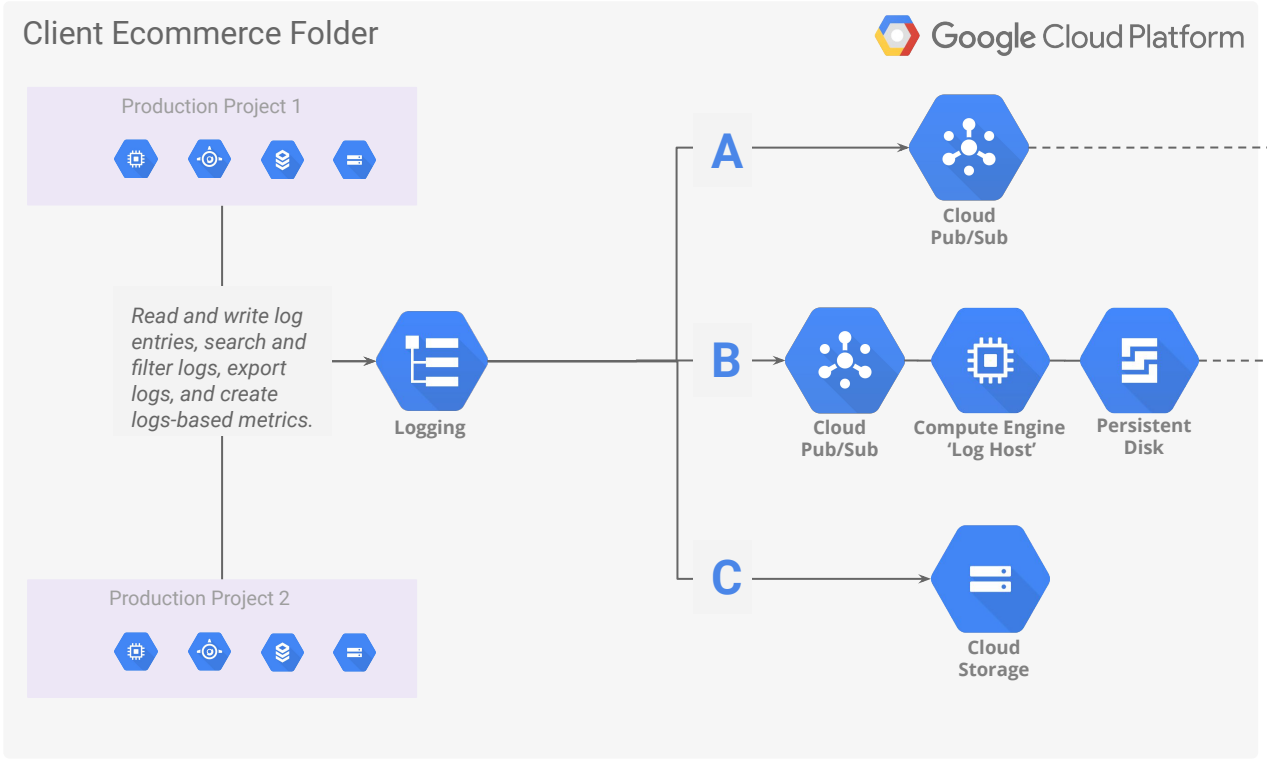


Two On-Premises gateways for hardware redundancy

Shared VPC



Client Logging Design

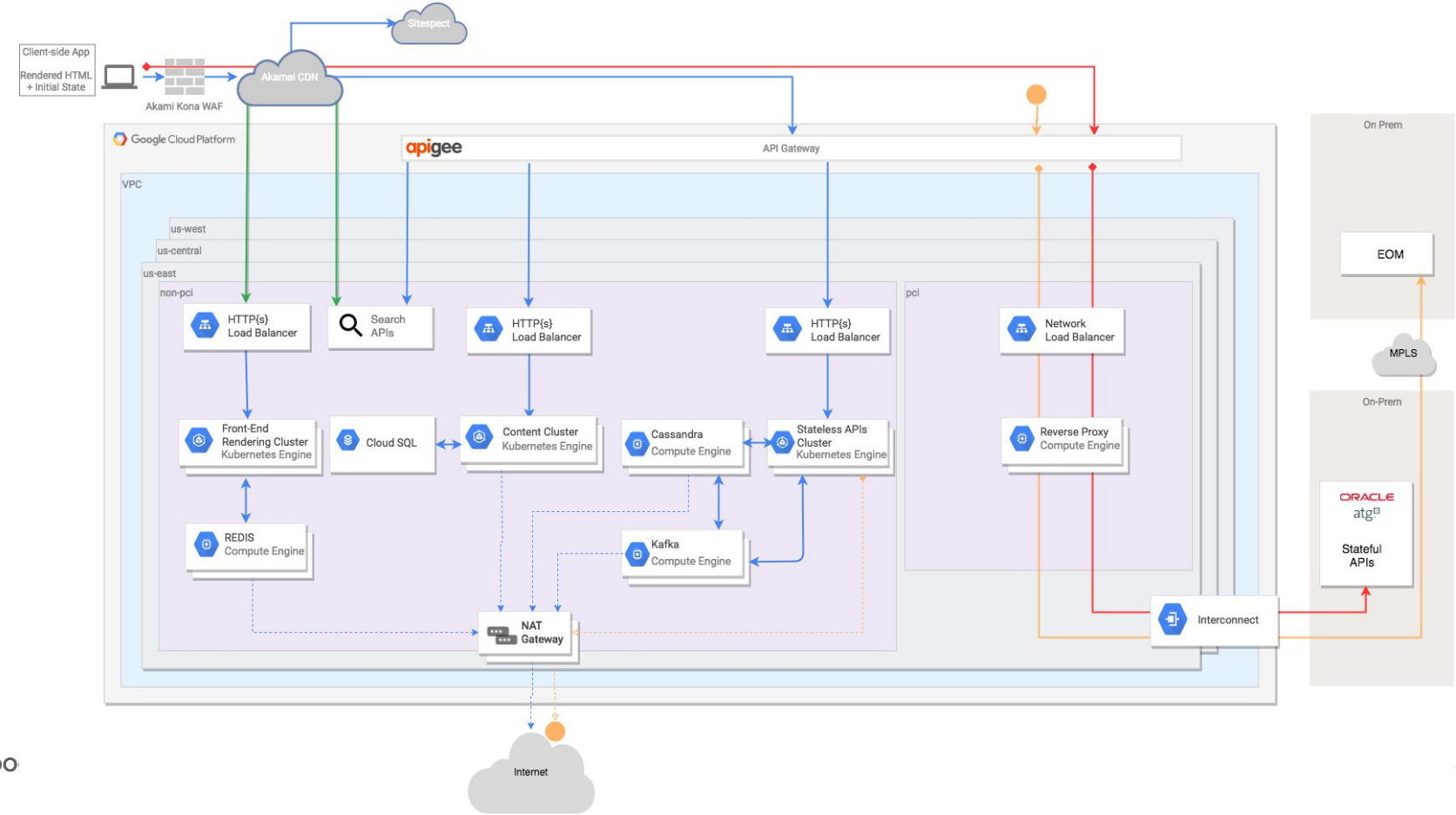


A. *Leverage pub/sub integration for 3rd party tools (e.g. sumologic)*

B. *Leverage pub/sub to deliver data to GCE for 3rd party tools*

C. *Export logging data to GCS for long-term retention*

Application Architecture



Path to GCP Certification

Google Cloud Platform Certifications

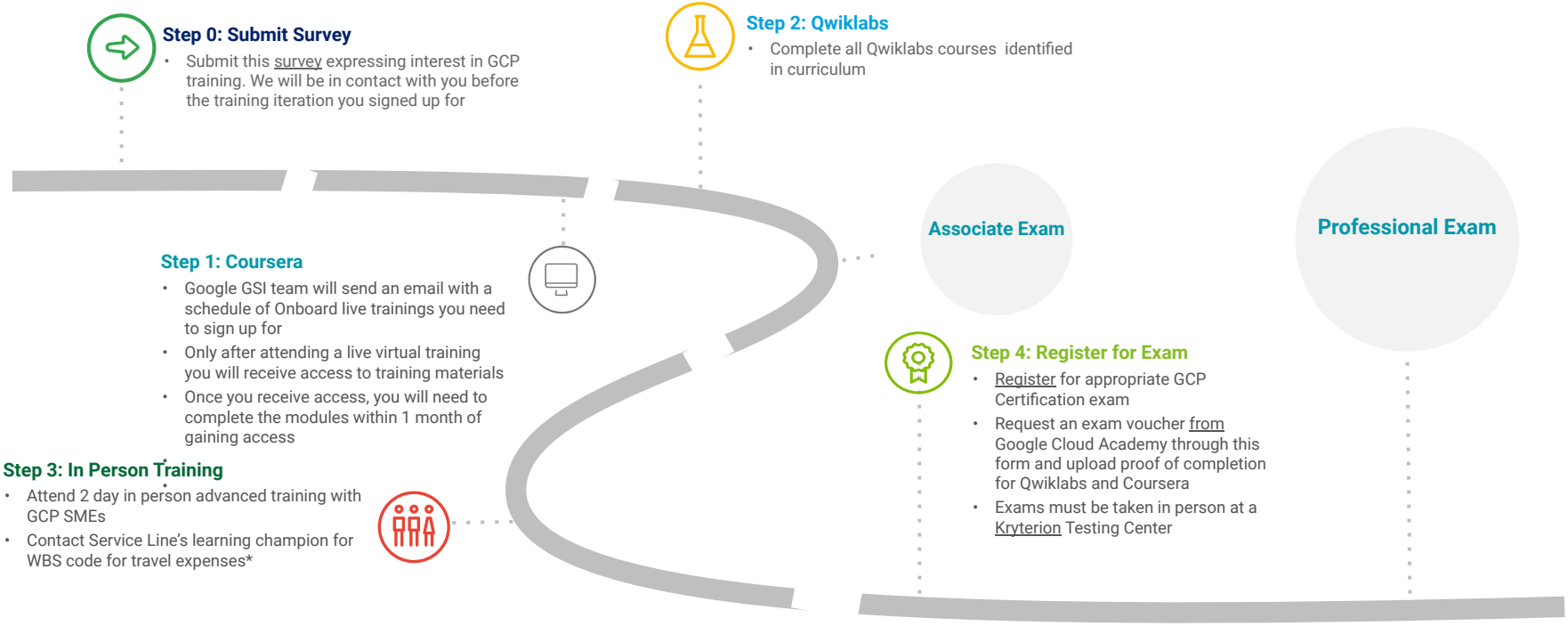
GCP offers two Infrastructure Certifications and one Analytics Certification.

	Associate Cloud Engineer **Infrastructure Level 1**	**Professional Cloud Architect** **Infrastructure Level 2**	**Professional Data Engineer** **Analytics Level 2**
Target Audience	Demonstrate your ability to deploy applications, monitor operations, and maintain cloud projects on GCP.	Demonstrate your proficiency to design, build and manage solutions on GCP.	Demonstrate your proficiency to design and build data processing systems and create machine learning models on GCP.
Required Skills	Practitioners who can use • Google Cloud Console • Command – Line Interface to perform common platform-based tasks should target this exam.	Practitioners who can design, develop and manage robust, secure, scalable, highly available, and dynamic solutions to drive business objectives should target this exam.	Practitioners who can make data-driven decisions by collecting, transforming and visualizing data should target this exam.
Exam Focus	• GCE vs GKE vs GAE • Cloud SDK commands • No case studies	• Designing Cloud Architecture • Networking concepts • Case studies	• Data storage options • Data processing strategies • ML service offerings

Google Cloud

Path to Certification

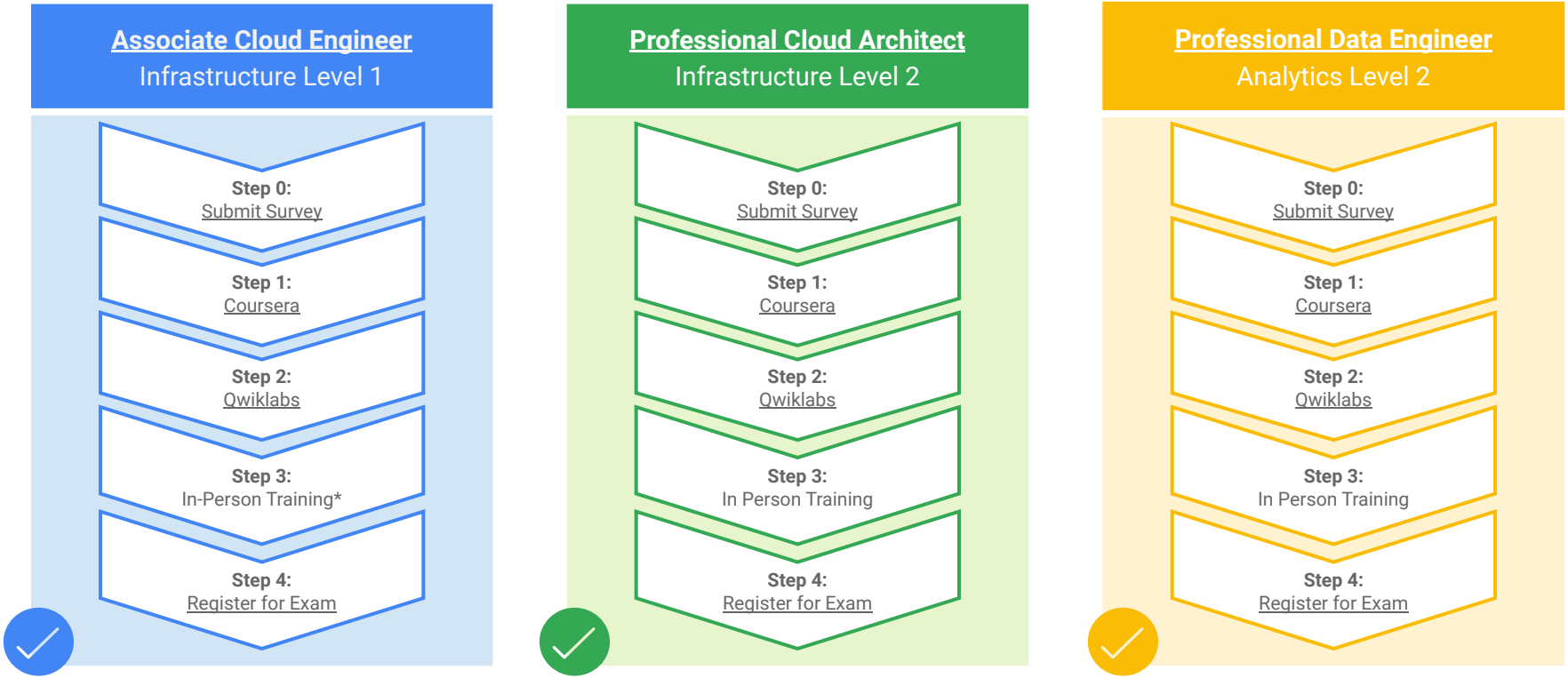
Follow these steps to enroll in Google Cloud Academy and earn certification.



Google Cloud

Certification Steps

Follow the below steps to earn certification in GCP, based on exam type.



Google Cloud

Note: In-person training is not necessary for the Associate Cloud Engineer exam

Associate Cloud Engineer Exam

Key learning materials and labs you need to complete certification are as follows.

Step 1: Coursera
Theoretical Foundation

a. Google Cloud Platform Fundamentals: Core Infrastructure
b. Essential Cloud Infrastructure: Foundation
c. Essential Cloud Infrastructure: Core Services
d. Elastic Cloud Infrastructure: Scaling and Automation
e. Elastic Cloud Infrastructure: Containers and Services
f. Reliable Cloud Infrastructure: Design and Process

Step 2: Qwiklabs Quests
Hands On Experience

a. GCP Essentials
b. Baseline: Infrastructure
c. Baseline: Deploy & Develop
d. Cloud Architecture
e. Google Cloud Solutions I: Scaling Your Infrastructure

Other Resources

a. Review Kubernetes Concepts
b. Review GCP Case Studies
c. Read the SRE Book
d. Review Google Cloud Platform Solutions

Google Cloud

Professional Cloud Architect Exam

Key learning materials and labs you need to complete certification are as follows.

Step 1: Coursera
Theoretical Foundation

a. Google Cloud Platform Fundamentals: Core Infrastructure
b. Essential Cloud Infrastructure: Foundation
c. Essential Cloud Infrastructure: Core Services
d. Elastic Cloud Infrastructure: Scaling and Automation
e. Elastic Cloud Infrastructure: Containers and Services
f. Reliable Cloud Infrastructure: Design and Process

Step 2: Qwiklabs Quests
Hands On Experience

a. GCP Essentials
b. Baseline: Infrastructure
c. Baseline: Deploy & Develop
d. Cloud Architecture
e. Google Cloud Solutions I: Scaling Your Infrastructure

Step 3: In Person Training
Two Day Training

a. Cloud Architecture
b. Target State Design
c. Application Migration
d. Integration & Optimization

Other Resources to Support Exam Preparation

- Read the SRE Book
- Review Kubernetes Concepts
- Review GCP Case Studies
- Review Google Cloud Platform Solutions
- Take the Official Google Cloud Architect Practice Exam

Google Cloud

Google Cloud Academy: Infrastructure Curriculum

The Google Cloud Academy addresses all steps to prepare for the Associate Cloud Engineer and Professional Cloud Architect exams.

Phase One: Self Study Training – 8 Weeks

Learning Objective: Understanding cloud architecture and Google technology, to design, develop, and manage robust, secure, scalable, highly available, and dynamic solutions to drive business objectives

Coursera

1a. Google Cloud Platform Fundamental: Core Infrastructure	1b. Essential Cloud Infrastructure: Foundation	1c. Essential Cloud Infrastructure: Core Services	1d. Elastic Cloud Infrastructure: Scaling and Automation	1e. Elastic Cloud Infrastructure: Containers and Services	1f. Reliable Cloud Infrastructure: Design and Process
6-10 hours	7-9 hours	7-9 hours	7-9 hours	7-9 hours	12-16 hours

Qwiklabs Quests

2a. GCP Essentials	2b. Baseline: Infrastructure	2c. Baseline: Deploy & Develop	2d. Cloud Architecture	2e. Google Cloud Solutions I: Scaling Your Infrastructure
3-5 hours	5-7 hours	4-6 hours	8-10 hours	6-8 hours

Phase Two: In-person Training – 2 Days

Learning Objective: Develop GCP Migration Subject Matter Expertise - Google Trained

3a. **Cloud Architecture**	3b. **Target State Design**	3c. **GCP Infrastructure Application**	3d. **Certification Ramp-up**

Google Cloud

Deconstructing an Architecture Case Study

Cloud Architect Certification Exam

Exam Sections

- Designing and planning a cloud solution architecture
- Managing and provisioning solution Infrastructure
- Designing for security and compliance
- Analyzing and optimizing technical and business processes
- Managing implementation
- Ensuring solution and operations reliability

https://cloud.google.com/certification/guides/cloud-architect/

Google Cloud

Cloud Architect Practice Exam

The Cloud Architect **practice exam** will familiarize you with types of questions you may encounter on the certification exam and help you determine your **readiness** or if you need more preparation and/or experience.

Successful completion of the practice exam does not guarantee you will pass the certification exam as the actual exam is longer and covers a wider range of topics.

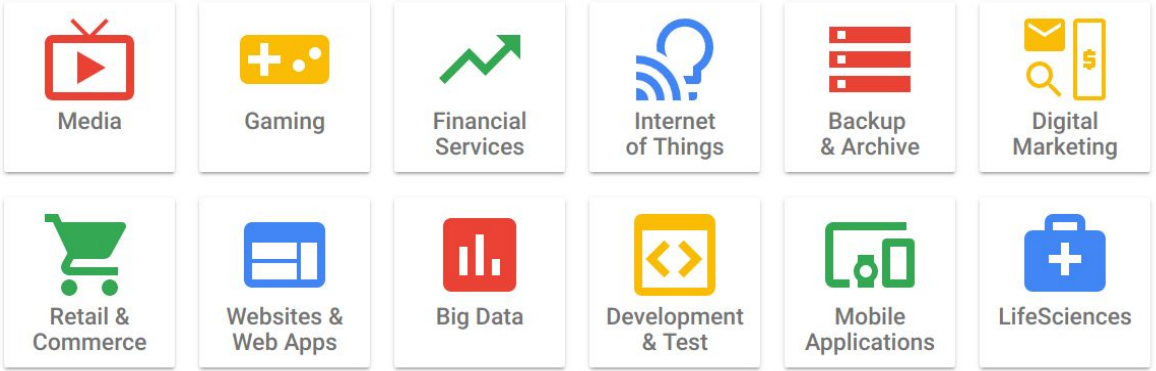
- There is no limit to the number of times you can take this practice exam.
- You can't save your progress. If you close the practice exam window, you must start from the beginning.
- There is no time limit for the practice exam, but we recommend completion in 45 minutes or less.

https://cloud.google.com/certification/practice-exam/cloud-architect

Google Cloud

Cloud Solutions Architecture Reference

Treehouse beta



https://gcp.solutions/

Visit the site for industry specific solutions that can be architected on GCP

Google Cloud

JencoMart

Business Overview

Key Business Points

- Retailer; 10,000 stores; 16 countries; Sales - store 75%:online 25%
- Core values: customer service, reduce carbon footprint by 50% in 5 years
- Immediate business goals
  - Asia expansion
  - Reduce on-prem footprint
  - Capacity optimisation - peak and off-peak periods
  - Guarantee service availability and support

Key Assumptions

- Managed services where possible
- Use of Data Analytics and Machine Learning to build stronger connections with customers
  - Questions here that could cover Dataproc, Dataflow and BigQuery
- Focus on cost optimisation
- High availability, and redundancy are a must
- Potential for applications to move to AppEngine or Kubernetes
- Products used must enable easy expansion into multiple regions
- Latency reduction

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
4 DC's (3 North America, 1 Europe)		
Oracle - 20TB, complex, clean data, strong backup		
PostgresSQL: US only, 12 hour backup, 100% uptime SLA, used for authentication		

Google Cloud

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
4 DC's (3 North America, 1 Europe)	Multi-zone redundancy Asia expansion	Cloud DNS, HTTP/TCP load balancing Cloud CDN
Oracle - 20TB, complex, clean data, strong backup	Oracle cannot be moved onto GCP today Potential for data migration Latency to on-prem	Interconnect / Cloud VPN in case Oracle DB stays and the GCP applications need to be connected to it
PostgresSQL: US only, 12 hour backup, 100% uptime SLA, used for authentication	Cloud SQL: First generation impact to performance with FLUSH TABLES WITH READ LOCK. Cost of these backups is included in instance cost; their storage space does not count against the allotted storage for the instance. Cloud SQL: Second Generation instances, the storage used by backups is charged at a reduced rate	Cloud SQL 2nd Gen with automated backup.

Google Cloud

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
Compute: - 30 machines US West, twin dual-core CPU, 32GB RAM, 250GB HDD - 20 machines US East, single dual-core CPU, 24GB RAM, 250GB HDD		
Applications most dual-homed. One customer loyalty portal: LAMP stack app hosted in 2 DC's in US		
Storage: 100TB SAN in each location, weekly tape backup		

Google Cloud

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
Compute: - 30 machines US West, twin dual-core CPU, 32GB RAM, 250GB HDD - 20 machines US East, single dual-core CPU, 24GB RAM, 250GB HDD	Cross regional load balancing Twin core vs single dual-core Persistent disk Backend security	Managed Instance Groups using Instance Templates / Custom machines / Highmem configuration Autoscaling HTTP Load Balancer Cloud DNS Internal Load Balancer Firewall Rules for backend security
Applications most dual-homed. One customer loyalty portal: LAMP stack app hosted in 2 DC's in US	Despite hosted in US, could potentially be extended to other regions or other regions may require access. Potential for LAMP Cloud Launcher for IaaS Possibility of Kubernetes / GCE	AppEngine if no write to disk required and PHP 5.5 - Also AppEngine standard can scale to 0 when no load AppEngine Flex if write to disk and PHP > 5.5
Storage: 100TB SAN in each location, weekly tape backup	Potential for backup to Nearline / Coldline	GCS Multi-regional for backup If re-architecture possible, then Datastore could be used with AppEngine

Google Cloud

Dress4win

Business Overview

Key Business Points

- Online web-based wardrobe management service
- Serves user base via WebUI + Mobile App
- Taps into social networks for user signals
- Monetizes using Ads, e-commerce, referrals
- Have grown very fast from >10 servers to >100 servers
- Capacity not enough for future growth
- Decided on moving to cloud
- Preference for managed services
- Considerations: competition, new features, elastic, auto-scaling, cost

Key Assumptions

- Will migrate in phases - phase 1 will be dev/test
- Future migration will be dependent on phase 1 - need to be optimised early on
- Disaster recovery and HA will be consideration
- Will migrate least critical software components first
- Plans to reduce Ops cost
- Wants to move towards DevOps model
- Increased velocity of product / feature releases (CICD approach)

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
Databases: - MySQL: static data, inventory, user data - Reddis: social graph, meta-data, caching	Two types of data: - Static stored in relational DB - In-memory / high throughput solution for social / meta-data / caching	**Cloud SQL** as replacement for MySQL Either containerised Redis or Memcache for AppEngine - **BigTable** **Datastore** for mobile backend
MySQL database that needs to be replaced - currently have scaling issue	Microservice based architecture Also serving static content If VMs are lifted-n-shifted, then multi-regional approach required	**K8S / Containers** for microservice deployment for some s/w components - if re-architecturing possible **GCE** for VMs, **Cloud Launcher** for Nginx Batch processing using **Dataflow** **App Engine** for application layer - autoscaling **Load Balancer** for High Availability / traffic routing
Storage: - iSCSi for VMs - Fiber Channel for DBs - NAS for logs & backups, image storage	Already using virtual machines; lift n shift possible Different classes of storage used for various services	**Local SSD** for VM hosts **SSD persistent disks** for databases incase if managed Cloud SQL is not opted **GCS** for backups, **Stackdriver** for logs, images storage (could go in image repository) - multi-regional for DR

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
Hadoop / Spark: - Data analysis - Real-time trending	Offline batch processing probably for social graph etc Analytics also done using the same platform	**Dataproc** for Hadoop / Spark workloads - If **HDFS** then connect to **GCS** - If **HBase** then connect to **BigTable** **Big Query** could be used for real-time trending / analytics
MQ servers: - Messaging, social notifications, events		**Pub/Sub** for messaging and social notifications **Cloud Functions** for event triggers
Networking Encryption Security scanners Jenkins	If Google supplied encryption not enough Security and access controls CICD	**Cloud VPN** **CSEK** for encryption **Firewall** rules **Cloud Launcher** for Jenkins apps (containers / VMs)

Google Cloud

Mountkirk Games

Business Overview

Key Business Points

- Online games platform
- New game being developed with high usage expectations
- Already have a plan in place for how the infrastructure should look like - “Only fully managed services”
- Issues with previous cloud provider
- Need for KPIs on speed and stability + other deeper insights

Key Assumptions

- Planned solution might/could not be ideal
- Strong focus on analytics and custom metrics
- Potentially different storage solutions for game itself and analytics

Google Cloud

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
On-premise and servers in different DCs - not to be replaced immediately; only for the new game.	Compute ● GCE with customized Linux distro ● Autoscaling ● Load balancing ● Low latency	GCE as suggested with managed instances groups HTTP(S) load balancer with autoscaling Stackdriver logging with custom metric to scale up / down A solution with GAE frontends and Cloud Endpoints for scaling (also covering some aspects mentioned above)
MySQL database that needs to be replaced - currently have scaling issue	Storage ● Managed NoSQL database ● SQL querying of 10TB historic data	Datastore for game state; fully managed > If that's the same as for the analytics wf then BigQuery Pokemon Go example for Datastore (also for HTTP(S) LB)
Analytics workflow ● Statistics written to File ● ETL > storing data in MySQL ● Reporting	Data ingestion ● Live metrics from game server ● "Late data" due to slow mobile networks ● Regularly uploaded data from mobile devices	**Pub/Sub** for buffering of live and late data **Cloud storage** for data uploads **Dataflow** for bulk and stream processing **BigQuery** for storage and analytics - this can also contain the 10TB historic data >> Example solution

TerramEarth

Business Overview

Key Business Points

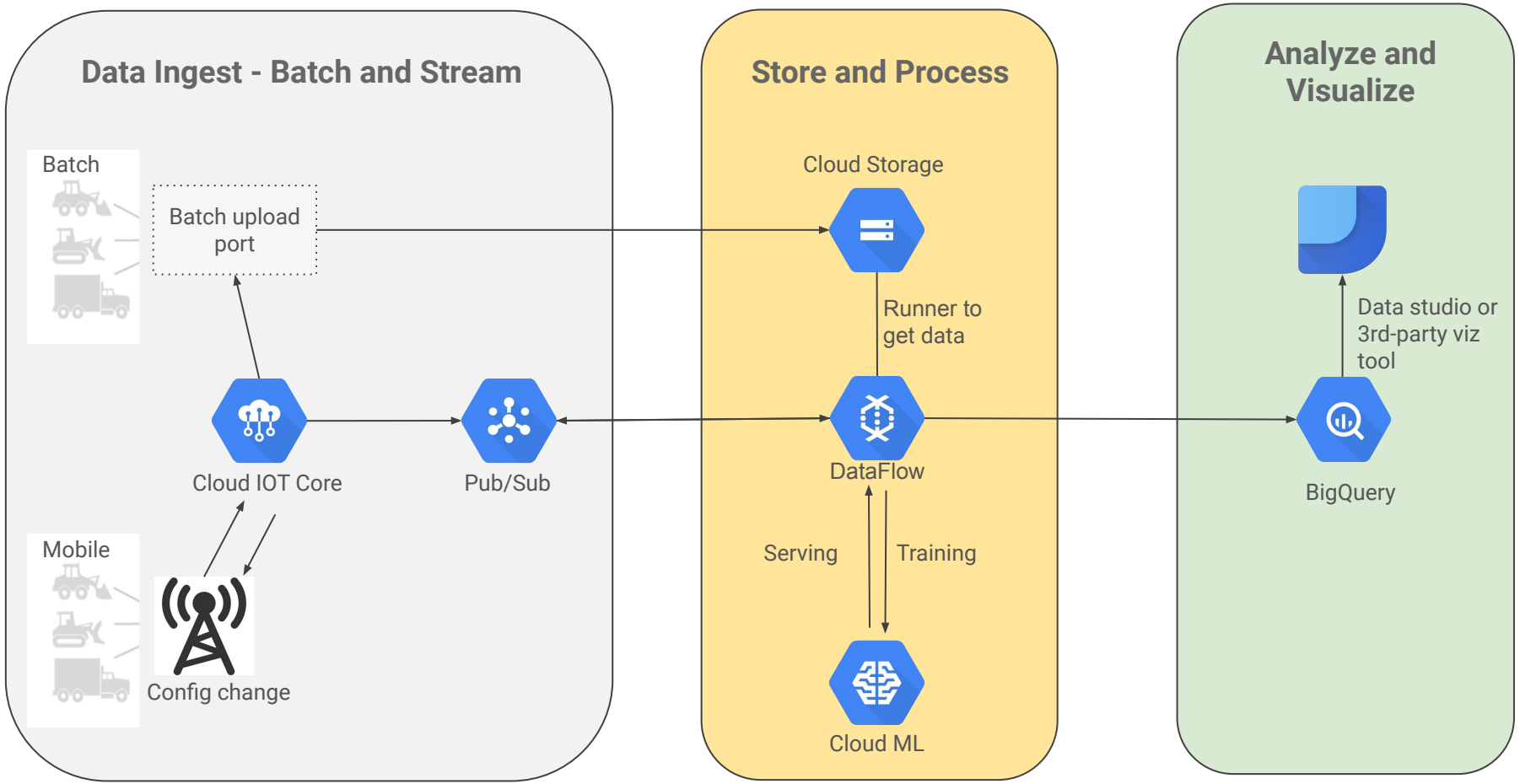
- Background - 80-20 Mining Agriculture, Family owned business from 1946, CEO and CTO does not see eye to eye on Technology approach
- 500 dealers, service centers over 100 countries
- Vehicle composition
  - 200,000 vehicles are connected to a cellular network, collects data directly
  - 120 fields data/sec - 22 hours of ops per data, 9TB/day
  - 19 million 800k vehicles data stored on vehicle, downloaded when serviced
- Data visibility 3 week delay -> Cause increase in planned parts stock
- Goals
  - Decrease downtime max 4 weeks -> avg < 1 week
  - Give dealers more visibility to data on customers
  - Ability to partner with different partner in AG industry

Key Assumptions

- Supply chain partners/processes for parts delivery stays the same
- OK with carrying current level of surplus inventory
- Change management team and training team in play - Address C-suite levels and training tech to staff

Technical Evaluation

Existing Environment	Technical WatchPoints	Potential Product/Solution
Infrastructure • OS - LInux based in DC Data transfer • Gzip CSV files -> FTP server -> ETL -> Data warehouse • Currently 3 weeks delay real to serve	• Able to handle 900 TB per day • Do not need Low latency I/O • 120 fields structured data • Smart machines	Database • Big Query for data warehouse and adv. Analytics serves Data processing • DataFlow to take care of ETL portion Machine Learning • For updating configs
Vehicles - Connected • Connected to Cellular (200k)	• 9TB/day • Over cellular network • Potential cellular network outages • Assumed that connected device can update config	Device management • Cloud IOT core Messaging • PubSub
Vehicles - Batch • Data dump during service (19.8M)	• 991 TB/day • Comes in batch • Updates configuration when serviced	Device management • Cloud IOT core Message management • Cloud Storage for storing • Dataflow runner job to get the batch data
Data visualization • Nonexistent or minimal	• Would like to give info to supply chain partners	• Data studio



Google Cloud

Questions?

Contact Details

- Alharith Hussin - alhussin@deloitte.com
- Eric Procopio - eprocopio@deloitte.com
- Shashank Joshi - shasjoshi@deloitte.com
- Kristen DeStefano - kdestefano@deloitte.com

Google Cloud

Appendix

Graveyard

Analyze Current State Architecture

How is identity and access managed for users?

Discussions related to IAM setup in the on-premise infrastructure will help in understanding access and authentication for users.

Key Questions	Use Cases	Applicable Google Component
How are users authenticated?	Multi-factor authentication	Admin Console
How are new users provisioned, managed and deleted?	Manual vs scripts	Admin Console
Which directory service is used to manage corporate identities?	Active Directory, G-Suite, LDAP	G Suite, Google Cloud Directory Sync
What are some current user groups being managed?	Network Admin, SysAdmin, Developers, Corp User, Corp Admin	Cloud IAM
What are various levels of access and permissions?	Edit, View, Publish etc.	Cloud IAM

What is the current networking footprint?

Understanding the mapping of various components in the on-premise network(s) will help create a blueprint of the current network architecture.

Key Questions	Use Cases	Applicable GCP Component
How many networks and subnets do you have?	Separate network/subnet for each environment or application or business unit	VPN, XPN
Do you have any specific network speed and latency requirements?	Some applications may have certain speed/latency thresholds	Direct Peering, Carrier Interconnect, Private Interconnect
Public vs Private IPs?	Restrict users from creating VMs with external IPs to keep application data within corporate network	VMs w/o external IPs
What security features do you have on your network i.e. firewalls, load balancers, bastion host etc.?	Applications with firewalls and LBs to manage network traffic	Load balancers, Firewall Rules, VPC Controls etc.

How are resource activities monitored and logged?

Resource monitoring conversations will identify performance tracking metrics and logging processes.

Key Questions	Use Cases	Applicable GCP Component
How are resources, performance and applications monitored?	Monitor network performance, data transfer rates, application workloads	Stackdriver
What 3rd party tools are being using to monitor resources?	Splunk, Logic Monitor, MetricStream	Stackdriver
What are some benchmarks and metrics being tracked?	IOPS, HTTP(s) Response times HTTP(s) error rates - average and max Connections, Traffic, Database Queries	N/A
Are there any regulatory retention policy for logs?	Some clients require logs to be stored for multiple years for audit purposes	BigTable, Cloud Storage

What is the disaster recovery and backup strategy?

Data and application recovery objectives will help evaluate uptime and redundancy requirements.

Key Questions	Use Cases	Applicable GCP Component
What is the current DR strategy?	Hot standby, warm standby, cold standby, warm static	GCP resources in different regions
What are the SLAs for public facing applications and RTO/RPO for internal applications?	Some applications have pre-defined availability to maintain e.g. 99.9% (3-9s), 99.999% (5-9s)	N/A
How is redundancy achieved for applications?	Multi-region, multi-platform	GCP resources in different regions
Does application data have any regulatory archiving requirements?	Some applications require logs to stored for predefined timeframes	Nearline Storage, Coldline Storage

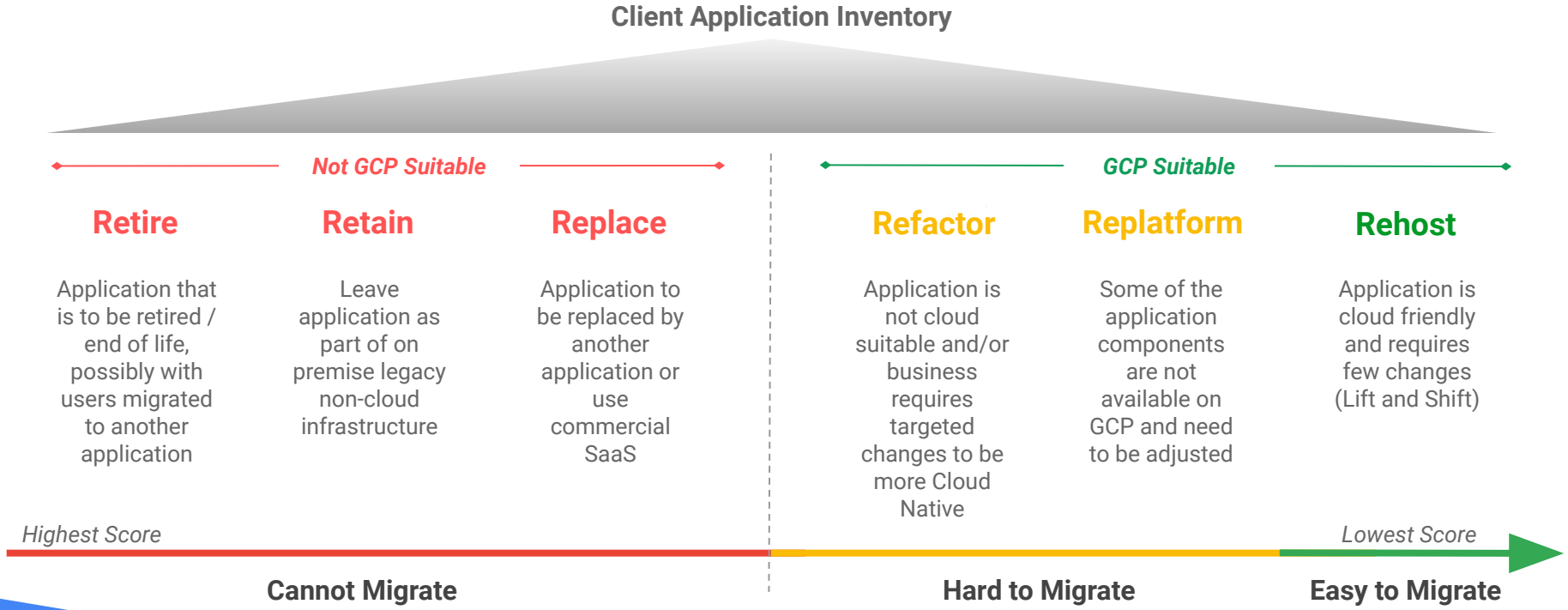
What is the current CI/CD and automation process?

Processes identified will help assess client maturity related to CI/CD and automation.

Key Questions	Use Cases	Applicable GCP Component
How are resources provisioned?	Manual, Automated	Cloud Deployment Manager
What is the current infrastructure configuration management process?	Standalone vs Master/Agent	Cloud Deployment Manager
What is the current CI/CD process?	Commit, build, automate, deploy	N/A
What tools are being used for CI/CD?	Chef, Puppet, Ansible, Salt, Jenkins, Bamboo	N/A
Is there a maturity roadmap already designed and being implemented?	Some clients have already developed a DevOps future state and approach..	N/A

How to determine the appropriate migration path?

Based on the suitability score calculation, applications can follow Rehost, Replatform or Refactor as a potential migration path to GCP.



Google Cloud

Data Migration

How to choose the appropriate storage solution?

Identifying data types and application requirements helps select the suitable GCP storage solution.

Persistent Disk

- Highest raw performance
- SSD or mechanical
- Attached to a GCE virtual machine
- Multiple VM can mount a single volume (read-only)
- Need to manage own ingest, publish and security
- Inherits host OS file system types and semantics

Cloud Storage

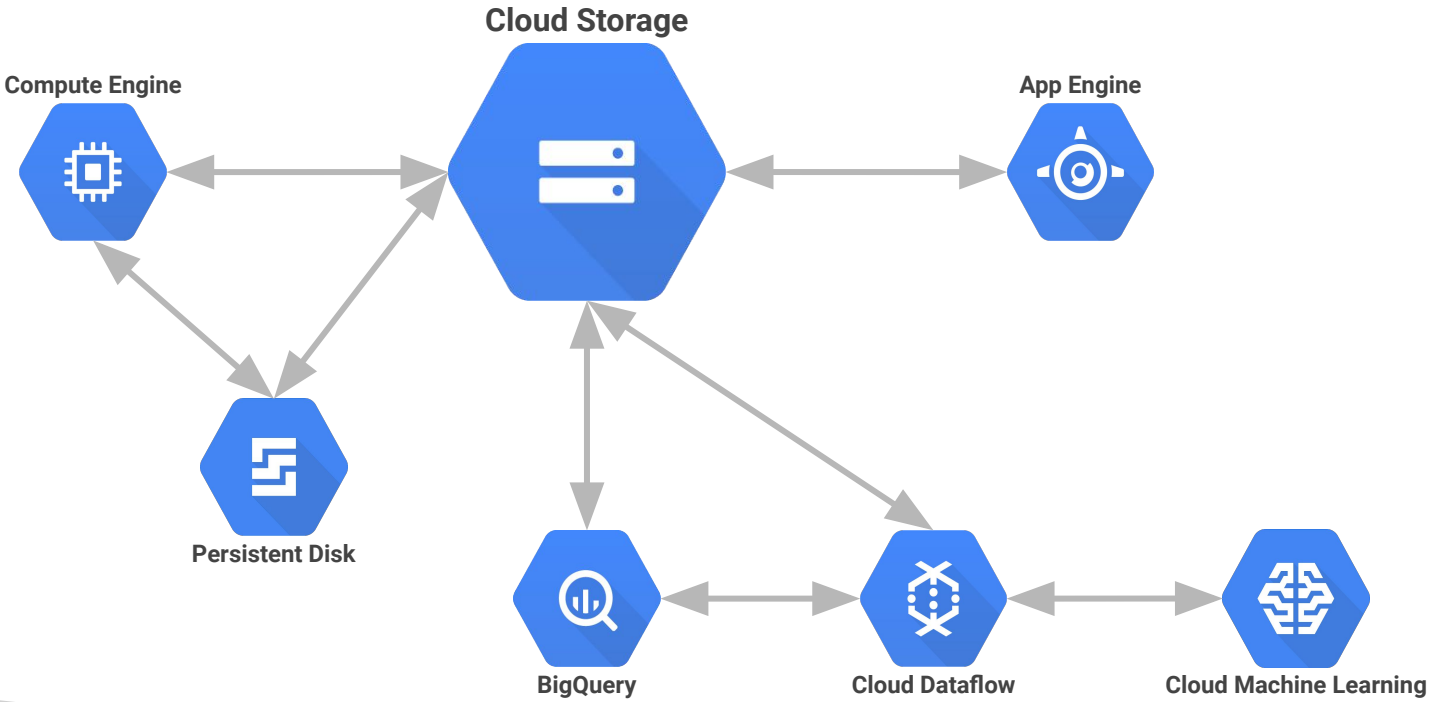
- Inherently higher access latency
- Choice of durability classes
- Access control seamlessly integrated
- In many cases, automatic versioning of objects can be applied
- Typically exposed via RESTful APIs

Database

- Data structures don't conform to "file" semantics from an application standpoint
- Optimized for specific types of access
- Provides indexing facilities for query performance
- RDBMS can be configured to enforce the referential integrity of entities

How can GCS enable data migration?

Google Cloud Storage (GCS) central to moving data to GCP as it is a common repository for data and integrates natively with other GCS services.



What options are available to move data into GCS?

Based on the data source, there are various options available to move data into GCS.

Cloud Storage Transfer Service	Data source is Amazon S3, another GCS bucket or a public URL Access to transfer service can be made through an API available both in JSON and XML
Gsutil (Cloud SDK)	Data exists on a disk with command-line access Powerful command-line transfer capabilities including Unix-like cp and rsync
Client Libraries	Client Libraries exist for many common languages (C#, Go, Java, Node.js, PHP, Python, Ruby) to integrate the ability to directly read from and write to GVS
3rd Party Service Providers	Move offline data by sending your physical media to third party service providers

How can you do offline media import/export ?

GCP leverages third party service providers for moving large data from your offline physical drives

Offline Media Import / Export is a third party solution for uploading offline data into Google Cloud Storage

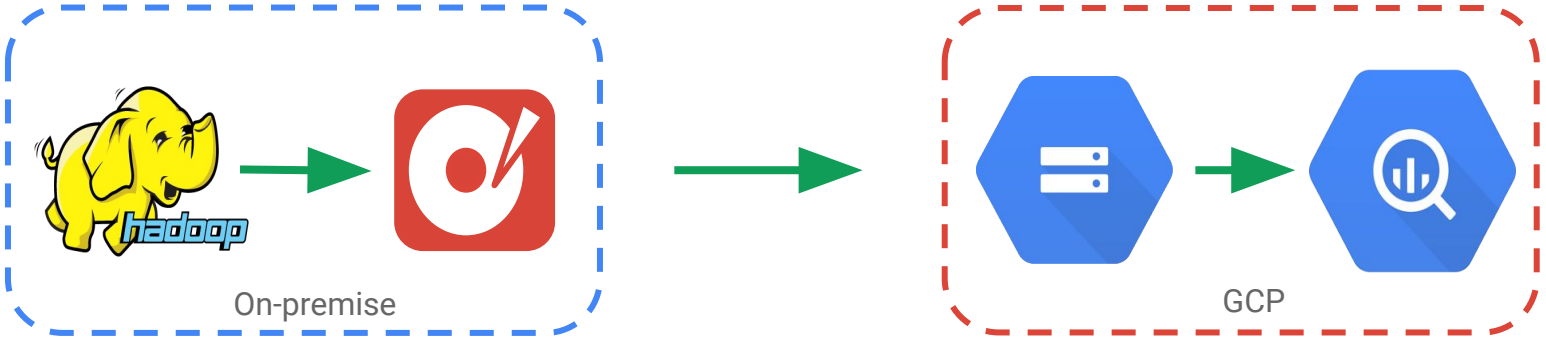
Recommended if you're limited to a slow, unreliable, or expensive Internet connection for data transfer

Organizations must make their own **arrangements directly with the third party service provider** chosen

What is a use case for migrating data to GCP?

One potential use case is a two-phased migration. It is applicable when services and software do not have the ability to read from a source and write directly to a destination.

Migrating Hadoop to BigQuery



Export data in Avro format from hadoop to a local file system

Copy the data to a GCS bucket

Load the data into BigQuery

What are best practices for data migration ?

Key considerations and recommendations for data migration into GCP.

1. Know your cloud options - understand and evaluate the available data migration tools
2. Define a process to validate data after the migration
3. Prepare plans and evaluate tools for adequate backup and data recovery strategy
4. Define IAM roles and demarcation points for your data
5. Use cloud KMS to manage your own encryption key for sensitive data
6. Identify and document your data compliance and audit requirements

Workload Migration

Can existing Microsoft licenses be also migrated?

License Mobility for GCP enables deployment of eligible Microsoft Windows Server applications using existing licenses.

- Available to Volume Licensing customers with active Software Assurance contracts
- Applies specifically to applications that run on windows server and not to the windows operating system
- Windows must be purchased as part of your VM consumption. Google provides images for:
    - Windows 2008 R2 Datacenter Edition
    - Windows 2012 R2 Datacenter Edition
- Customer must maintain appropriate Client Access Licenses (CALs) for accessing the application servers
- Active Directory can be deployed onto GCP or remotely connected through Cloud VPN

What are the different options for migrating workloads?

Based on application complexity, there are three architectures that can be followed for workload migration.

Lift and Shift

Shut down existing applications and copy data and disk images to the cloud

Straight P2V or V2V machine migration

Small modifications, such as network and user reconfiguration

Redesign

Essentially the same path as creating a new application in the cloud

Moderate to large modifications of the application architecture, replacing major components

Replacing legacy systems with cloud native services

Hybrid

Variation of lift and shift and requires fewer steps

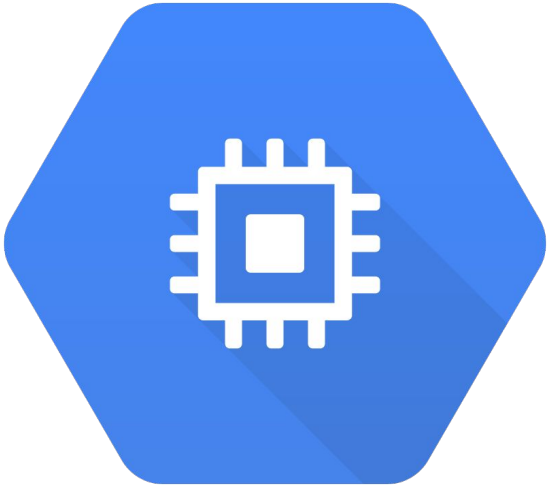
Only a part of the application moves to the cloud

Typically application front end is migrated and back end services remain with existing infrastructure

What is the key aspect of a lift and shift migration ?

Importing disk images from existing systems to compute engine is central to the lift and shift architecture.

- Disk images can be transferred by two methods:
  - **Block Copy**: Third party agents are installed on the source and destination systems which write disk blocks to a secondary PD mounted on a GCE instance and then swap disks
  - **Image**: Write disk images into a GCS bucket and use this image when creating a new GCE instance

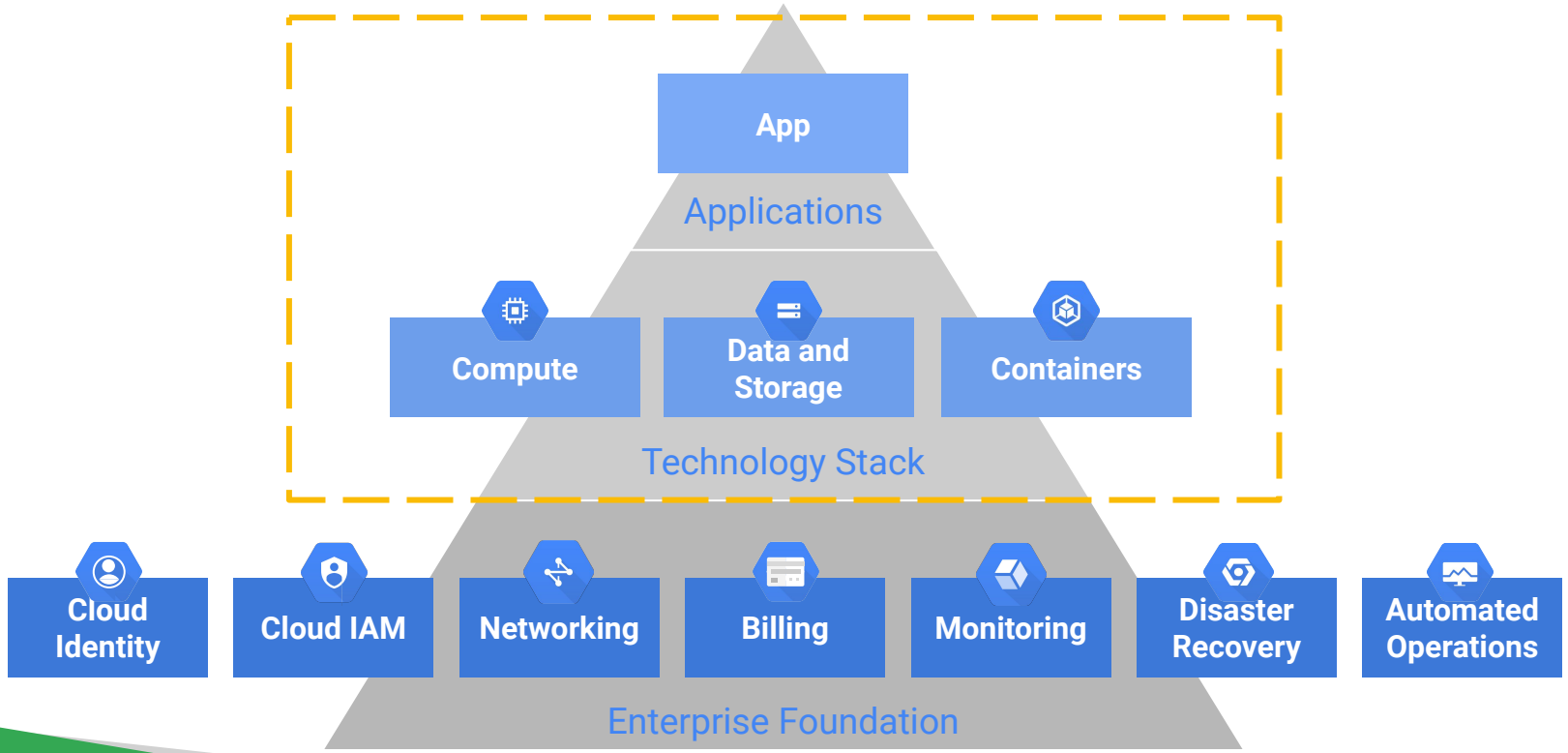


- Once a GCE instance is running, use startup scripts or automation tools for provisioning and configuration management:
  - Chef / Puppet / Ansible / Salt

Design and Architect Cloud Native Applications

Designing and Architecting Applications in GCP

After the enterprise foundation is set up, organizations can be developed and architecting applications using the technology stack.



Google Cloud

Application Patterns

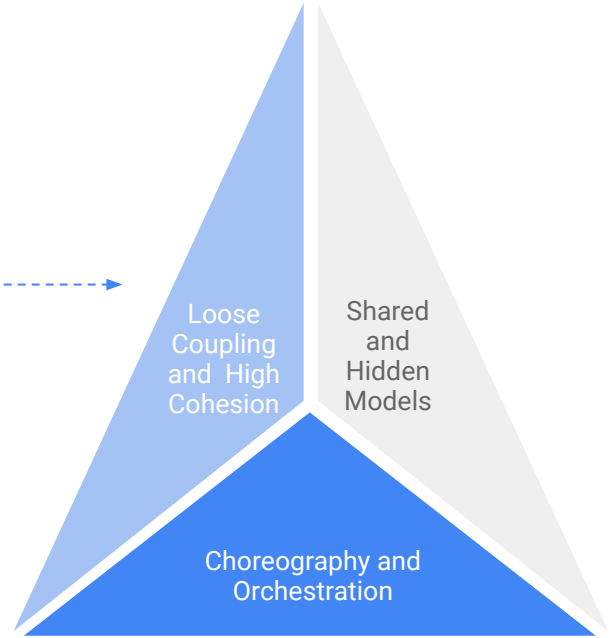
What are microservices?

Microservices are a way of architecting applications whereby responsibilities are defined with clear boundaries, services do not affect each other, and related behaviours are grouped.

Define the service's **bounded context**

Microservices are just that - they shouldn't suffer from functional sprawl.

Clear **responsibility delegation** and **decomposition** are hallmarks of a healthy microservice ecosystem.



What are shared and hidden models?

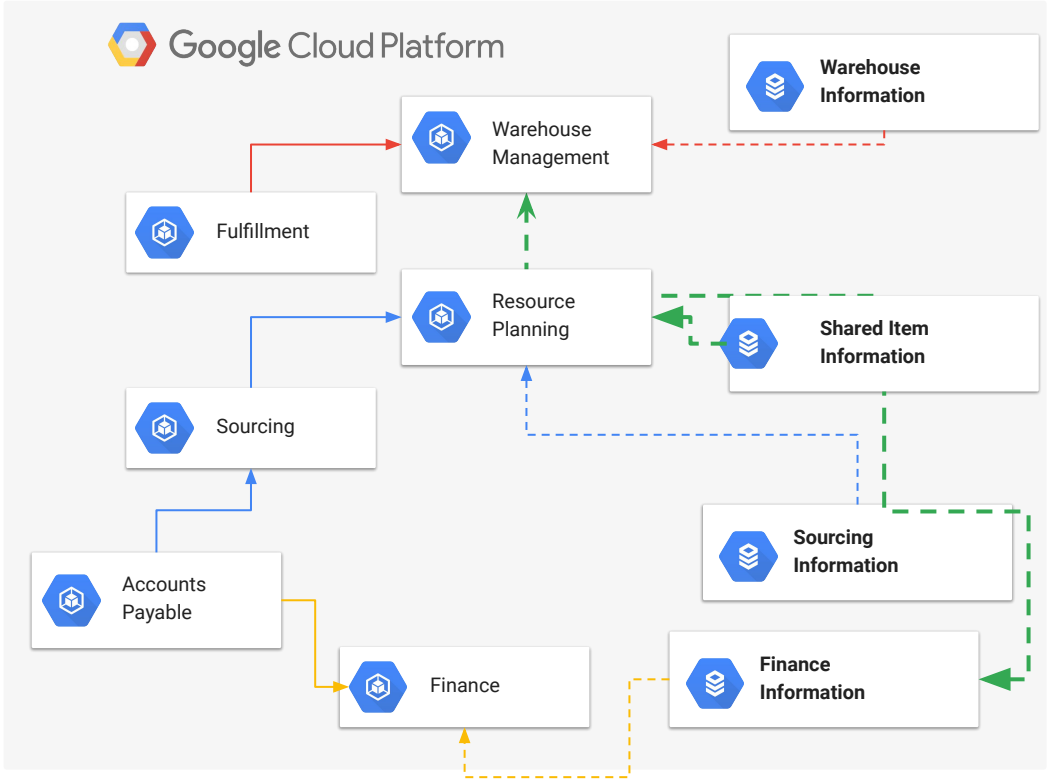
Each application keeps a hidden view of the data model. Upstream services will consume downstream services that will “hide” the persistent information schema and make it easier to evolve the application.

Entity Model: *Vendor*

Warehouse perspective
Receiving
Picking
Packing

Finance perspective
Accounts Payable
GAAP reporting

Sourcing perspective
Purchasing
Manufacturing
Shipping



What is choreography?

Choreography asynchronously coordinates the activities of arbitrary systems and applications.

Systems-of-record publish **lifecycle events** to a relevant messaging topic. Publisher is client-agnostic but maintains a **mapping of events to topics**.

Depending on when the subscriber receives message, it may **sit in queue**. Hence, Choreography does **not** operate in real-time.

Loose coupling (along **well-defined** bounded contexts or APIs)

Easily **extensible**

Reactive (events trigger behavior)

Guaranteed delivery

Additional **effort** spent to **monitor** all relevant **participant components**

What is orchestration?

Orchestration executes a defined sequence of events in succession.

A **pipeline** is established to execute the workload, with the **output of one stage becoming the input** of the subsequent stage.

Orchestrations do operate in **real-time,** barring any artificial delay injection by a processing stage.

A pipeline may be **halted at any arbitrary stage**, as compared to Choreography, which **fires-and-forgets**

Simple and easy to rationalize when the pipelines are kept modest

Finer-grained **control** of event **execution** and **timing**

Requires **coordination** among independent **delivery teams**

May lead to tighter coupling

What are backends and front ends for applications?

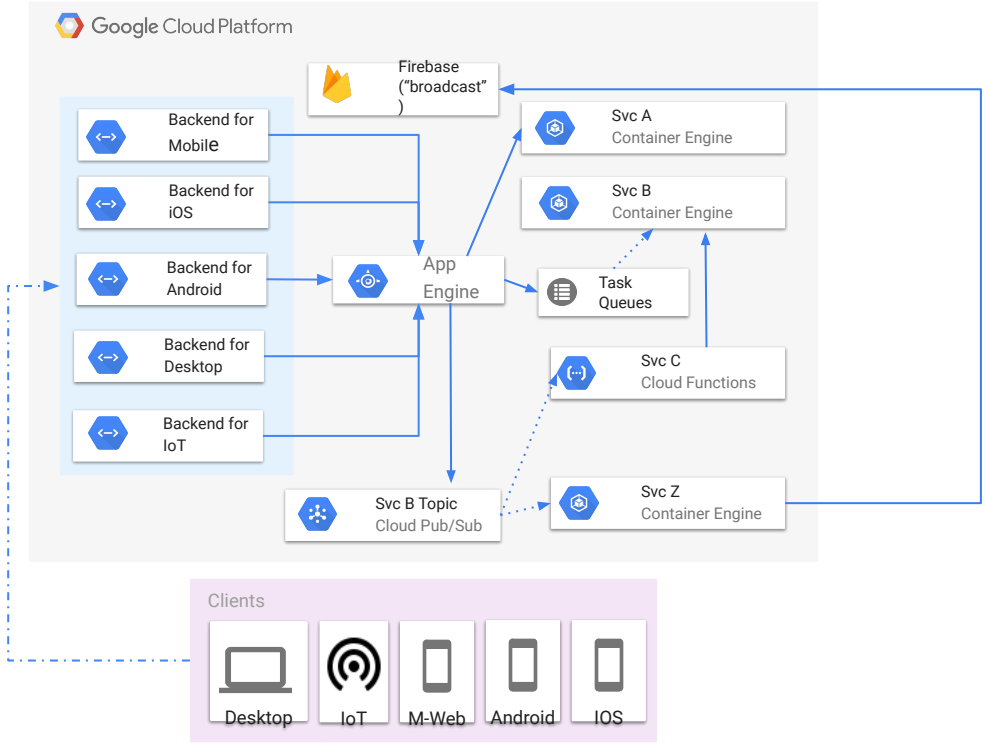
Backends and frontends are a way of decoupling services and capabilities of the an application.

Separate **presentation logic** from **core**

Provides an **additional tier** for **scaling**

Enables **session management**

Supports **asynchronous communication**



How to design a data pump for applications?

Data generated by an application can be decoupled from the services layer using a data pump to asynchronously deliver data to be properly transformed for consumption by the clients.

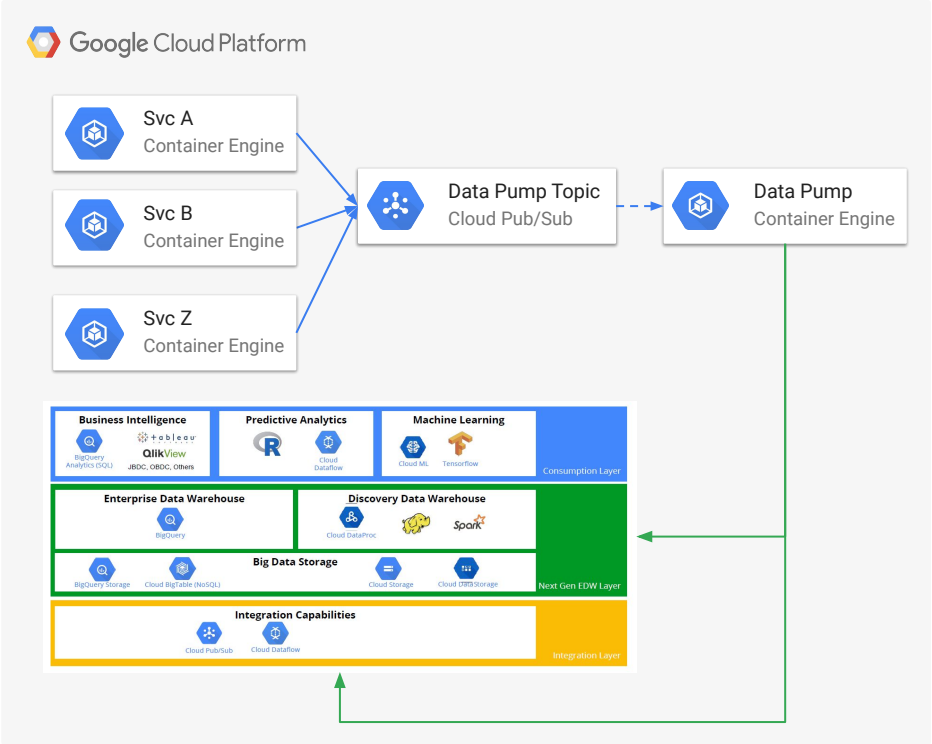
Consolidate diverse data sources to a **single destination**

Individual platform services send **data points** to a **centralized infrastructure**

Fed by **systems-of-record**

Good use for **changed data notification** features supported downstream

Facilitates analytics across **disparate** enterprise **data sources**



Application Dependencies

What are some examples of application dependencies?

Identifying and understanding application dependencies are important to map, record, and manage applications effectively .

Software and Client Packages

Runtime configurations (database connections)

Secret Management

Packaging and deployment

Google managed OS/package mirrors

Image families on GCE

What are key considerations for managing dependencies?

Key considerations include a clear vision of application dependencies, managing dependencies securely, and externalizing configurations for designing your architecture, managing and debugging effectively.

A clear vision of application dependencies

Identify and map **upstream/downstream** and **internal/external dependencies** your application has and design your architecture accordingly.

Manage dependencies securely

Employ **IAM rules** and **private** and **secure connections** such as VPN to connect to your external dependencies/

Externalize configurations

Externalize configuration components and **dependency variables** from code for easier management and debugging

How to manage dependencies on GCP?

GCP provides various dependency management tools to Runtime Config API, Deployment Manager, and Repositories.

Runtime Config API

Dynamically configure and expose variables through GCP.

Set Watchers and Waiters to watch for changes to data and return based on certain conditions.

Deployment Manager

Provision application resources in sequence

Define infrastructure configurations declaratively

Repositories

Google Container Registry

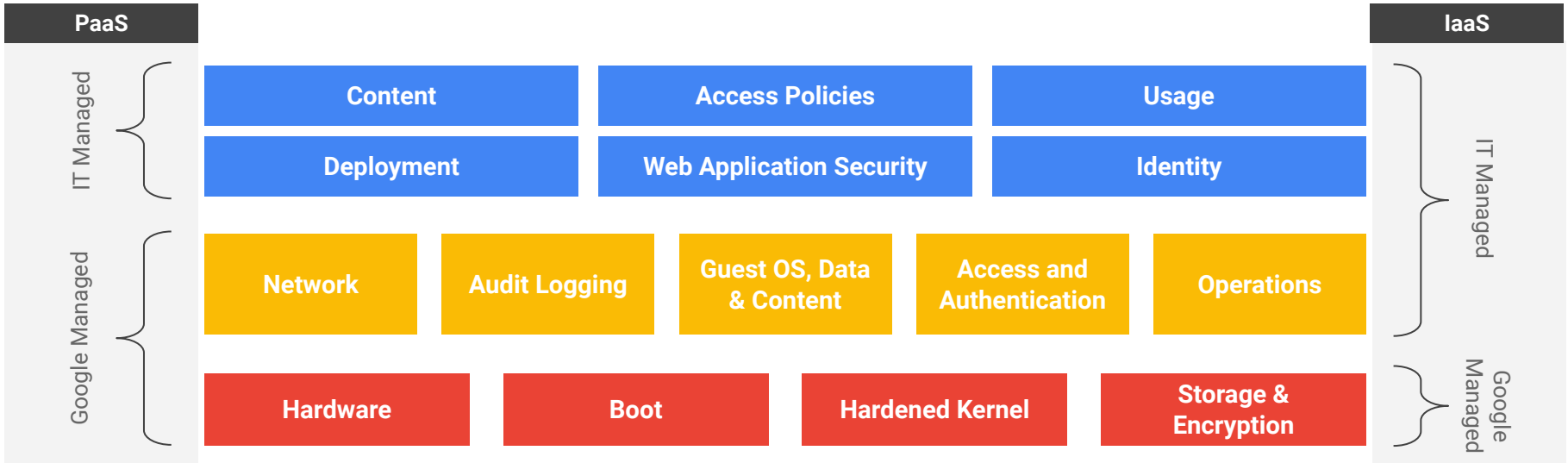
Google Code Repository

Google Image Repository

Application Security

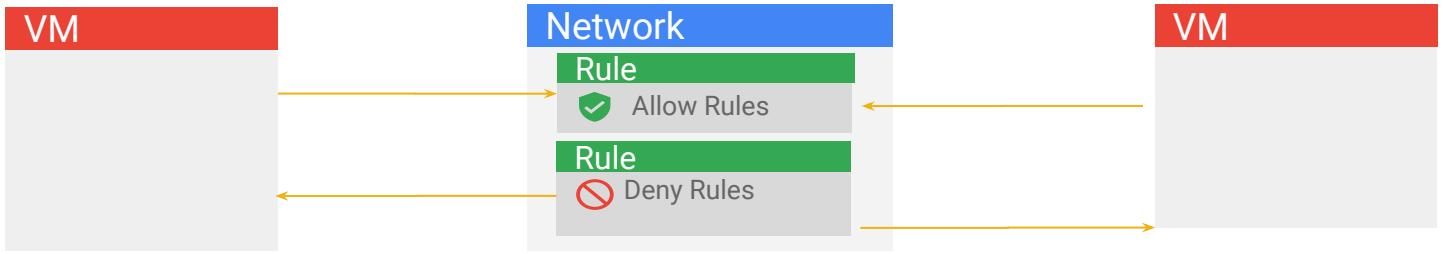
What is a shared security responsibility?

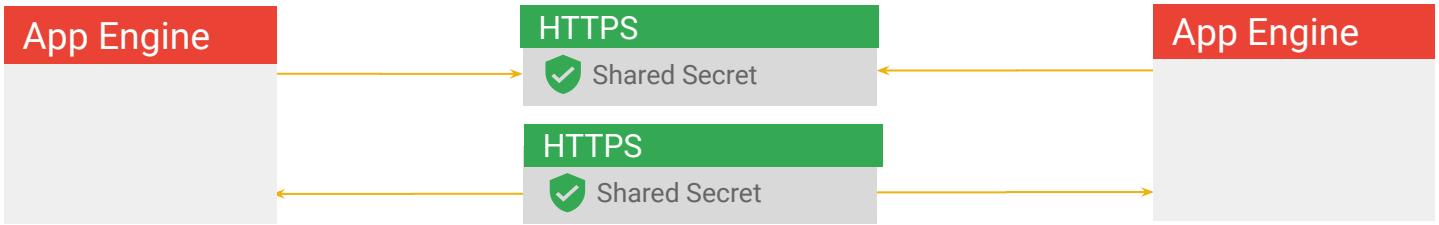
A shared responsibility model is a security framework that outlines which components of the platforms are managed by Google and which are managed by customers based on the level of abstraction chosen.



How to secure internal communication?

Communications between applications can be secured through the use of firewall rules and shared secrets.

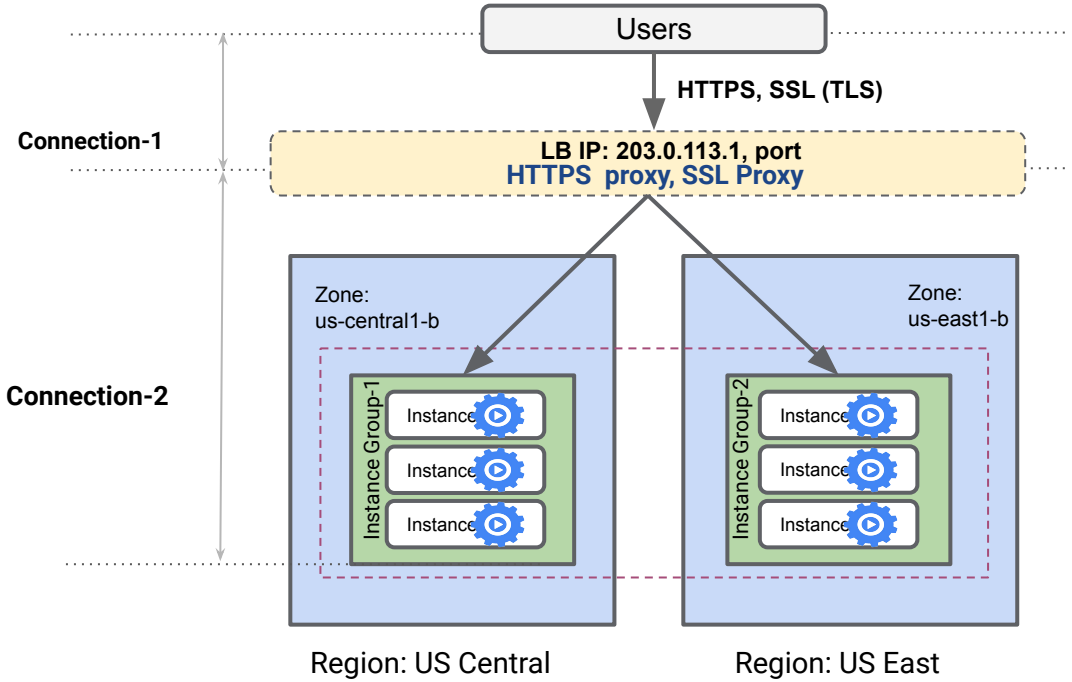




How to secure end user communication?

App Engine supports SNI and can host multiple domains for the same application. SSL and HTTPS load balancers too provide this.

App engine, HTTPS Load Balancers and SSL Load Balancers can host and serve SSL certificates.



How to manage application security in GCP?

Google offers several products to assist customers in managing security for their applications.

Web Security Scanner

- Alerts for XSS, Flash injection, Mixed-content, insecure javascript libraries.
- Google built, not a third party product. Virtualized browsers.
- Currently runs against GAE standard only

Certificate Management

- TLS termination near end client.
- Google takes responsibility for securing and serving certificate.

Identity Aware Proxy

- Authenticate internal accounts to your application.
- Require Google Accounts.

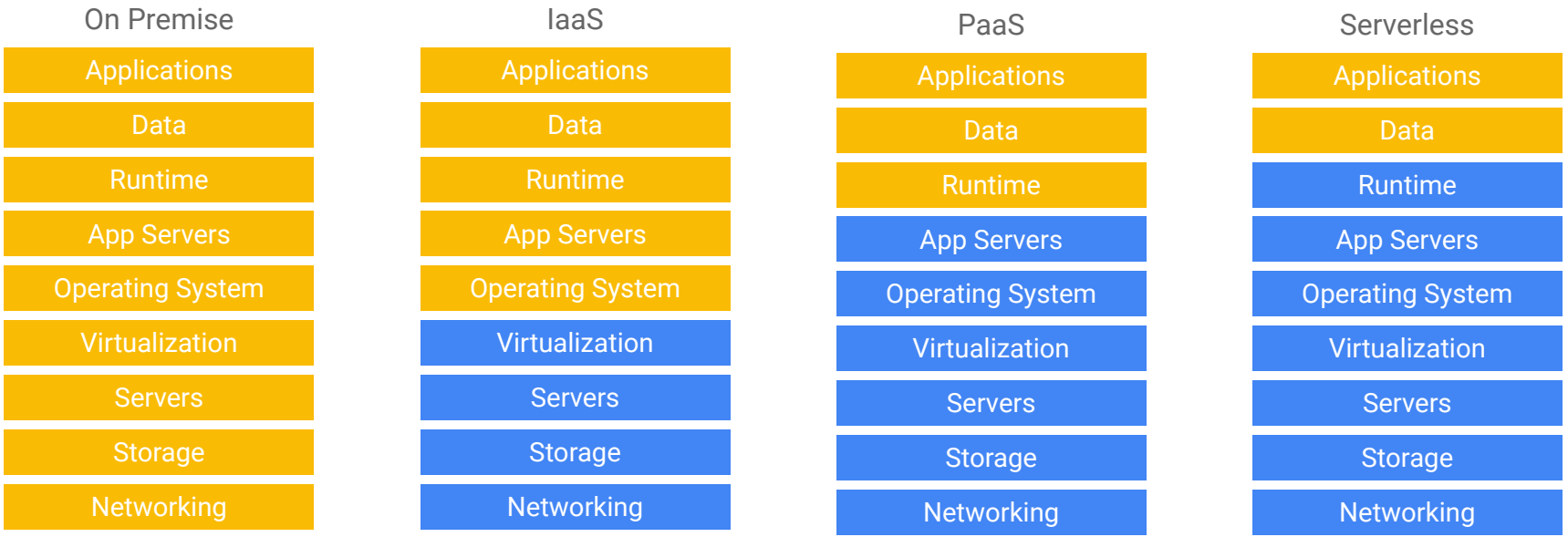
Forseti

- Scan resource ACLs and organization level policies.

Compute

What are the different layers of abstraction?

Moving from on premise infrastructure to cloud allows organization choose from various levels of abstraction based on their applications and business requirements.



What options are availability for compute?

At each level of abstraction, GCP offers several compute options to choose from when architecting a cloud native application.

Compute Engine

- Image
- Disk
- Network
- Instance
- Instance group
- Memory
- Core

Container Engine

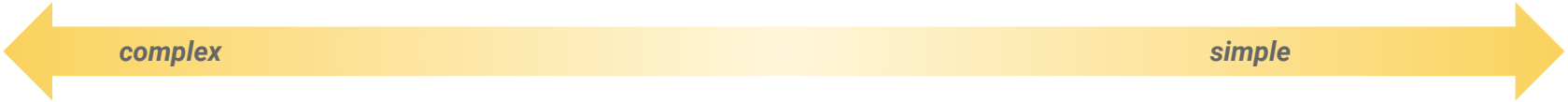
- Container
- Workloads
- Nodes
- Cluster
- Services

App Engine

- Application
- Service
- Version
- Instance

Cloud Functions

- Function
- Event
- Trigger



Google Cloud

What is Google Compute Engine?

Google Compute Engine (GCE) offers high performance virtual machines, and customer-friendly pricing using Google's worldwide fiber network.

Good Fit

- Existing systems - lift and shift
- Specific OS / kernel required
- License requirements
- Using GPUs
- Platform flexibility - particular criteria in your operating environment
- Run databases
- Network protocols beyond HTTP/S

Considerations

- Operational overhead
- Managing software software updates
- 7000 instances per VPC Network
- Max per VM
  - 96 vCPUs
  - 624 GB of RAM
  - 64 TB of Disk (PD)
  - 3 TB of Local SSD
- Larger disk has higher throughput

What is Google Container Engine?

Google Container Engine (GKE) is a powerful cluster management and orchestration system for running Docker containers.

Good Fit

- Run container in multiple cloud environments
- Less dependency on host machine
- Take full advantage of containers
- Have or want CI/CD pipeline

Considerations

- Must use containers
- Architectural considerations (stateless app)
- Learning curve compared to Google Compute Engine and App Engine

What is Google App Engine?

Google App Engine (GAE) is a fully managed platform that completely abstracts away infrastructure so organizations can focus only on code and application development.

Good Fit

- HTTP/S request-response
- Stateless serving applications
- Scaling to high traffic
- Fully managed infrastructure (low-ops)

Considerations

- Support for Python, Java, PHP, & Go
- Low ops cost vs higher runtime costs
- Requires Public IP
- Less platform flexibility
- Websockets and session affinity not supported (on roadmap)

What is Google Cloud Functions?

Google Cloud Functions is a fully-managed, serverless environment where Google handles infrastructure, operating systems, and runtime environments.

Good Fit

- Serverless - fully managed execution environment
- Isolated execution of functions
- Event driven - connect Cloud services via Cloud Pub/Sub, Cloud Storage & HTTP Requests
- No need to worry about runtime data transformations (ETL)

Considerations

- One runtime: JS on Node.js
- Function level granularity
- Must interact via events

How do the options compare?

Language support, usage model, and primary use cases should be considered as key factors when selecting which compute option to use.

					
	Compute Engine	**Container Engine**	**App Engine Standard**	**App Engine Flexible**	**Cloud Functions**
Language support	Any	Any	Java 7 Python 2.7 Go PHP 5	Java 8 Python 2.7/3.5 Go, PHP 5/7 Node.js Ruby Custom Runtimes	Triggers (Node.js)
Usage model	IaaS	IaaS PaaS	PaaS	PaaS	Microservices Architecture
Abstraction Level	Server	Cluster	Autoscaling managed servers		Serverless
Primary use case	General Workloads	Container Workloads	Scalable web applications Mobile backend applications		Lightweight Event Actions

What drives a particular selection?

Flexibility in products and processes, maximum control, languages, and libraries are the driving factors when looking at the compute options.

Compute Engine

- Maximum control
- Flexibility
- Legacy products or processes
- Bespoke tools, languages or libraries

Container Engine

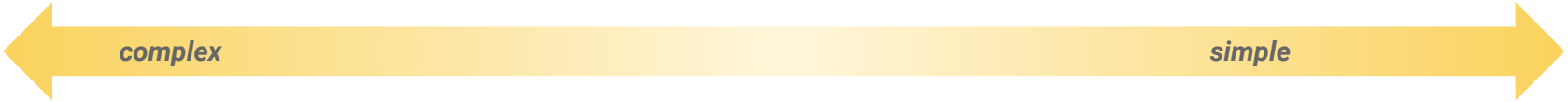
- Application management and deployment model
- Containerization of application resources
- Clustered topologies

App Engine

- Statelessness of application tiers
- Ease of deployment
- Automatic scaling

Cloud Functions

- Event-driven
- Algorithmic encapsulation
- Ad-hoc service integration



What are some compute design considerations?

Location, service level objectives, data-driven approach are the key areas to consider during application design and right sizing the application infrastructure.

Logic travels light, so **co-locate compute resources** in close **proximity to data**

Service-level objectives should drive the **application design**

Adopt a **data-driven** approach to **right-sizing** application infrastructure

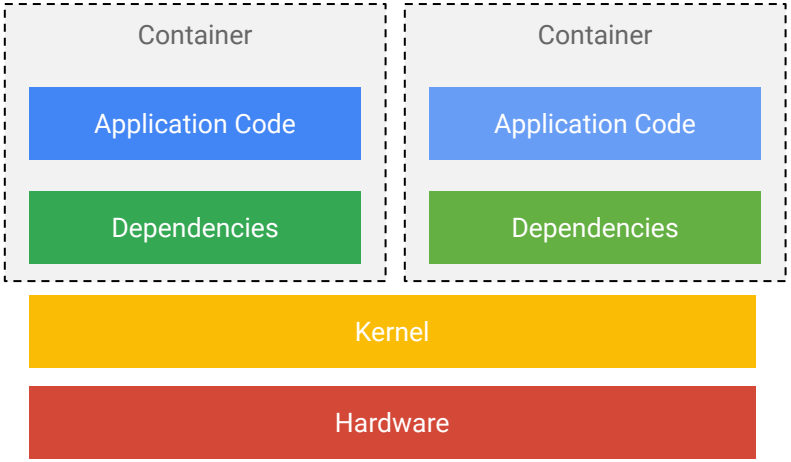
Google Cloud

Container Orchestration

Why use Containers?

Containers loosen the coupling between application and OS layers allowing consistency across your environments.

- "Separation of code and compute"
  - Consistency across dev, test, and production
  - Consistency across bare-metal, VMs, and cloud
  - No more "it worked on my computer"
- Packaged applications
  - Agile application creation and deployment
  - Continuous Integration/Delivery
- A path to microservices
  - Introspectable
  - Isolated/loosely coupled, distributed, and elastic



Key Benefits

- Support consistency across development, testing and production environments
- Loose coupling between application and operating system layers
- Much simpler to migrate workloads between on-premise and cloud environments
- Support agile development

What is Kubernetes?

Kubernetes is an open-source system for automating deployment, scaling and management of containerized applications.

Scheduling: Decide where my containers should run

Lifecycle and health: Keep my containers running despite failures

Scaling: Make sets of containers bigger or smaller

Naming and discovery: Find where my containers are now

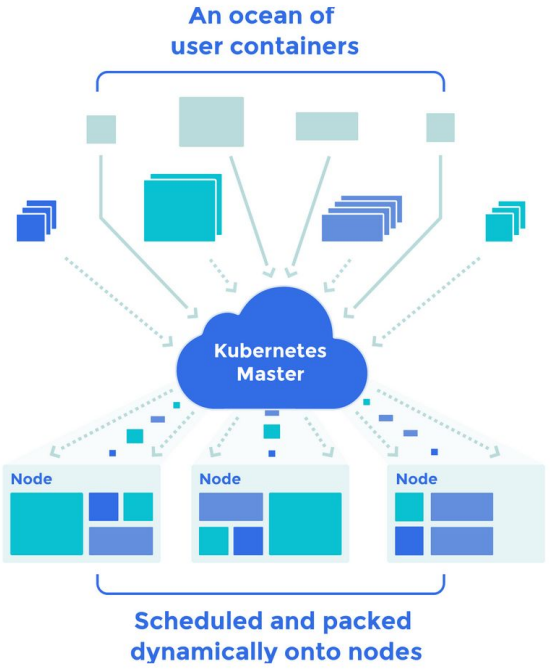
Load balancing: Distribute traffic across a set of containers

Storage volumes: Provide data to containers

Logging and monitoring: Track what's happening with my containers

Debugging and introspection: Enter or attach to containers

Identity and authorization: Control who can do things to my containers



What are Pods?

A Pod is the basic building block of Kubernetes–the smallest and simplest unit in the Kubernetes object model that you create or deploy. A Pod represents a running process on your cluster.

Small group of containers & volumes

Tightly coupled - in the same node

The **atom** of cluster scheduling & placement

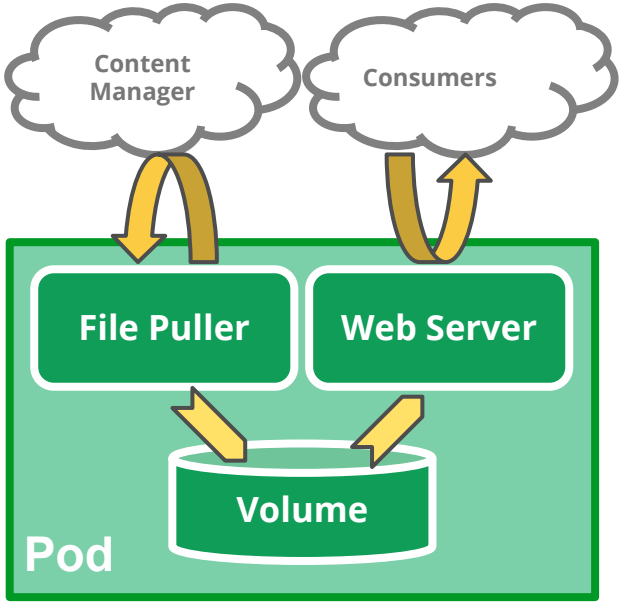
Each pod has its own IP address

- **shared namespace: share IP** address & localhost

Ephemeral

- can die and be replaced

Example: data puller & web server



What are Volumes?

A volume is a directory on disk or in another container. A volume outlives any containers that run within the Pod, and data is preserved across Container restarts.

Pod-scoped storage

Support many types of volume plugins

- Empty dir (and tmpfs)
- Host path
- Git repository
- GCE Persistent Disk
- AWS Elastic Block Store
- Azure File Storage
- iSCSI
- Flocker
- NFS
- vSphere
- GlusterFS
- Ceph File and RBD
- Cinder
- FibreChannel
- Secret, ConfigMap, DownwardAPI
- Flex (exec a binary)
- ...

What are Replica Set?

Replica Set ensures that a specified number of pod replicas are running at any give time. It ensures that a pod or a homogeneous set of pods is always up and available.

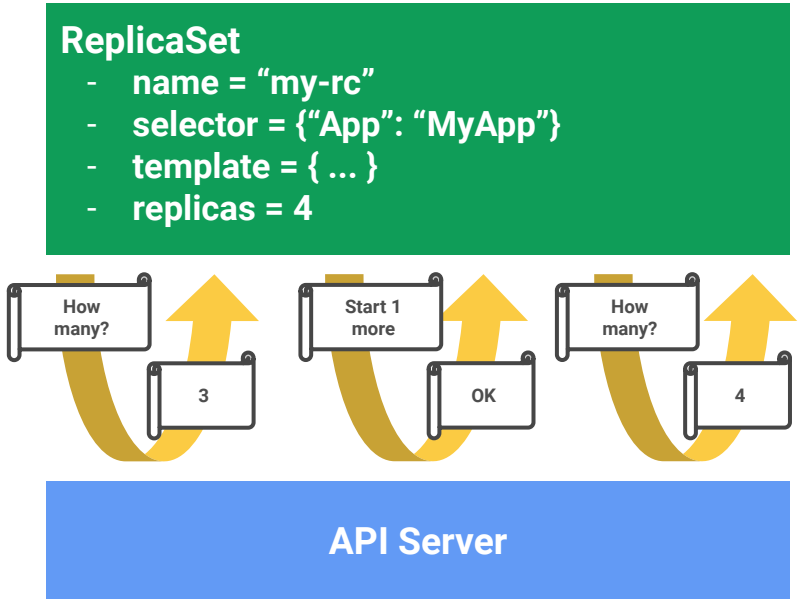
A simple control loop

Runs out-of-process wrt API server

One job: ensure N copies of a pod
- grouped by a selector
- too few? start some
- too many? kill some

Layered on top of the public Pod API

Replicated pods are **fungible**
- No implied order or identity



How to access additional resources?

Learn more about containers and kubernetes through public documentation links, tutorials, labs and videos.

Documentation

- Container Engine
- Kubernetes Concepts
- Kubernetes Basics

Tutorials

- Compute Engine & Kubernetes (Container Engine)
- Running a Container in Kubernetes with Container Engine
- Orchestrating the Cloud with Kubernetes
- Hello Node Kubernetes Codelab

Videos

- Kubernetes and Google Container Engine
- ABCs of Google Container Engine: tips and best practices
- Google Container Engine - The easiest way to use containers in production

What are crucial factors when considering storage access?

Proximity and latency are two crucial when determining where and how to store application data.

Proximity

Data has physicality, but is accessed logically

- Data with which we interact must ultimately travel from **physical coordinates**

Compute and storage are optimally co-located

- Large datasets have gravity and inertia
- Want to compute at storage PoPs

Balancing performance and resiliency

- Replicas too far apart?

  Replication lag: quantified data loss exposure (RPO)
- Replicas too close together?

  They share a **blast radius**

Latency

Speed-of-light constant + infrastructure overhead

- Impossible to get a packet from NY to CA in 10ms
- These constraints are critical when architecting business continuity and disaster recovery solutions

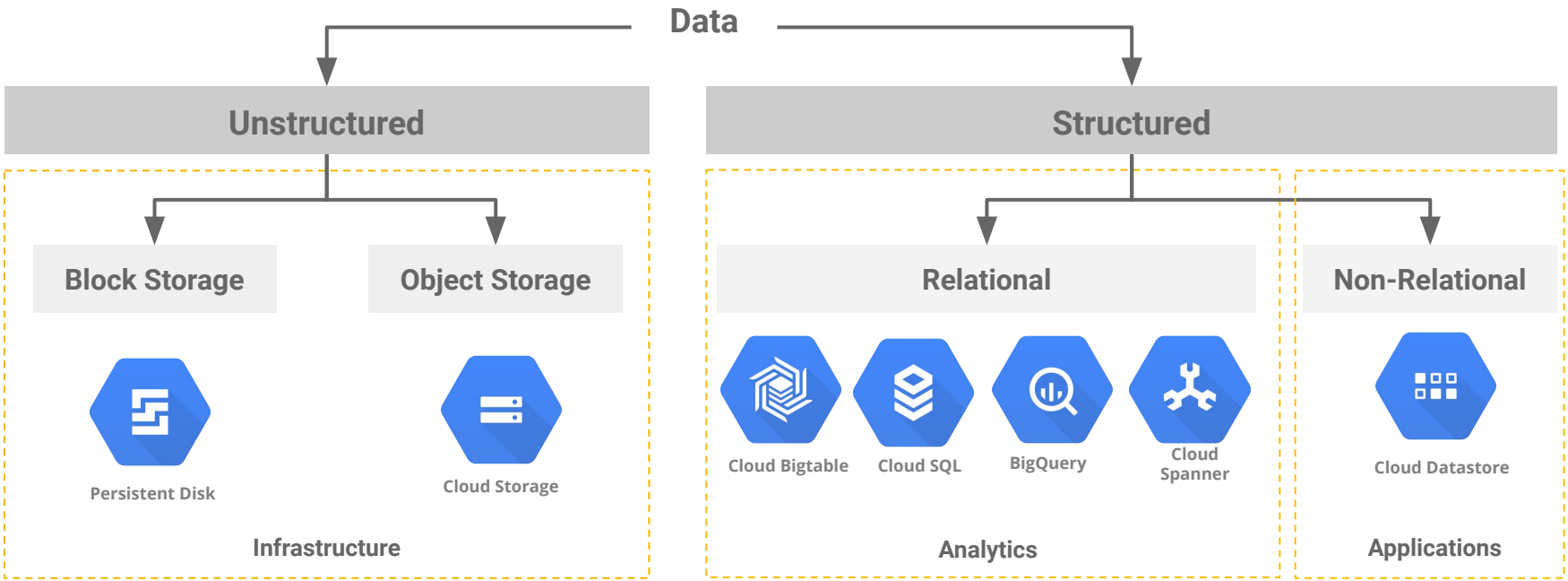
Core to the perception of application performance

- Edge computing / CDN keeps resources closer to end-user for this reason

Data and Storage Management

How does GCP handle different types of data?

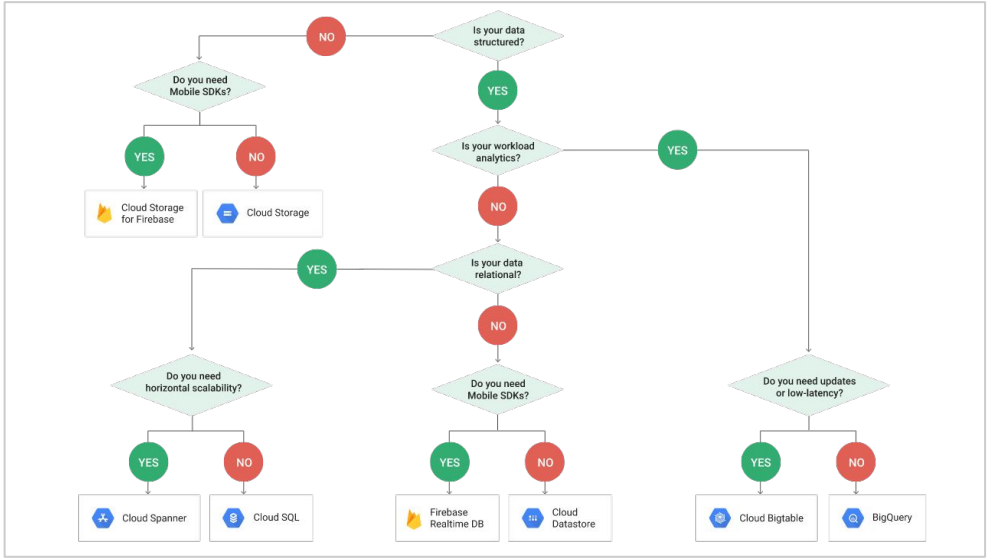
GCP offers various storage and database solutions based on the data types and application requirements.



How to select the right GCP storage service?

Data storage requirements for applications and workloads help evaluate and determine the appropriate GCP storage service.

- What are workload and data requirements of my application portfolio?
- Is my data structured / unstructured / relational or non-relational?
- Will I require more than one type of storage?
- Does the selected service pass through my various use cases?
- How will I handle exceptions?



Google Cloud

What unstructured storage solutions does GCP offer?

Following GCP managed storage services eliminate the overhead of managing unstructured storage solutions.

Cloud Storage

A scalable, fully-managed, and cost-efficient object / blob store.

Use Cases

- Storing and streaming multimedia (Images, pictures, and videos)
- Storage for custom data analytics pipelines
- Archive, backup, and disaster recovery

Persistent Disk

Fully-managed, price-performant block storage that is suitable for virtual machines, containers and snapshots.

Use Cases

- Disks for virtual machines
- Sharing read-only data across multiple virtual machines
- Rapid, durable backups of running virtual machines

What structured storage solutions does GCP offer? (1 of 3)

Following GCP managed storage services eliminate some of the overhead of managing databases.

Cloud Bigtable

A scalable, fully-managed NoSQL wide-column database that is suitable for real-time access and analytics workloads with low latency read/write access.

Use Cases

- IoT, finance, adtech
- Personalization, recommendations
- Monitoring
- Geospatial datasets
- Graphs
- Native time series

Cloud SQL

A fully-managed MySQL and PostgreSQL database service built for web frameworks and OLTP workloads.

Use Cases

- Websites, blogs, and content management systems (CMS)
- Business Intelligence (BI) applications
- ERP, CRM, and eCommerce applications
- Geospatial applications

What structured storage solutions does GCP offer? (2 of 3)

Following GCP managed storage services eliminate some of the overhead of managing databases.

Cloud Spanner

Mission-critical, relational database service with transactional consistency, global scale and high availability.

Use Cases

- High transactions
- Adtech
- Financial services
- Global supply chain
- Retail

BigQuery

A scalable, fully-managed Enterprise Data Warehouse (EDW) with SQL and fast response times for OLAP workloads up to petabyte-scale

Use Cases

- Reporting via Business Intelligence (BI) tools
- Analytical reporting on large data
- Data Science and advanced analyses
- Big Data processing using SQL

What structured storage solutions does GCP offer? (3 of 3)

Following GCP managed storage services eliminate some of the overhead of managing databases.

Cloud Datastore

A scalable, fully-managed NoSQL document database for use with semi-structured application, hierarchical, and durable key-value data.

Use Cases

- User profiles
- Product catalogs
- Game state

How to select data archive options in GCP?

GCP offers two options for data archival that can be determined based on frequency of access and minimum storage duration.

GCS Nearline

Very low cost per GB stored with data retrieval costs and a 30-day minimum storage duration

Use Cases

- Data you do not expect to access frequently (i.e., no more than once per month). Ideal for back-up and serving long-tail multimedia content.

GCS Coldline

Lowest cost per GB with data retrieval costs and a 90-day minimum storage duration

Use Cases

- Data you expect to access infrequently (i.e., no more than once per year). Typically this is for disaster recovery, or data that is archived and may or may not be needed at some future time.

What are crucial factors when considering storage access?

Proximity and latency are two crucial when determining where and how to store application data.

Proximity

Data has physicality, but is accessed logically

- Data with which we interact must ultimately travel from **physical coordinates**

Compute and storage are optimally co-located

- Large datasets have gravity and inertia
- Want to compute at storage PoPs

Balancing performance and resiliency

- Replicas too far apart?

  Replication lag: quantified data loss exposure (RPO)
- Replicas too close together?

  They share a **blast radius**

Latency

Speed-of-light constant + infrastructure overhead

- Impossible to get a packet from NY to CA in 10ms
- These constraints are critical when architecting business continuity and disaster recovery solutions

Core to the perception of application performance

- Edge computing / CDN keeps resources closer to end-user for this reason

What are the phases of the data lifecycle?

The data lifecycle has four phases - Ingest, Store, Process & Analyze, and Explore & Visualize.

Ingest

The first stage is to pull in the raw data, such as streaming data from devices, on-premises batch data, application logs, or mobile-app user events and analytics.

Store

After the data has been retrieved, it needs to be stored in a format that is durable and can be easily accessed.

Process & Analyze

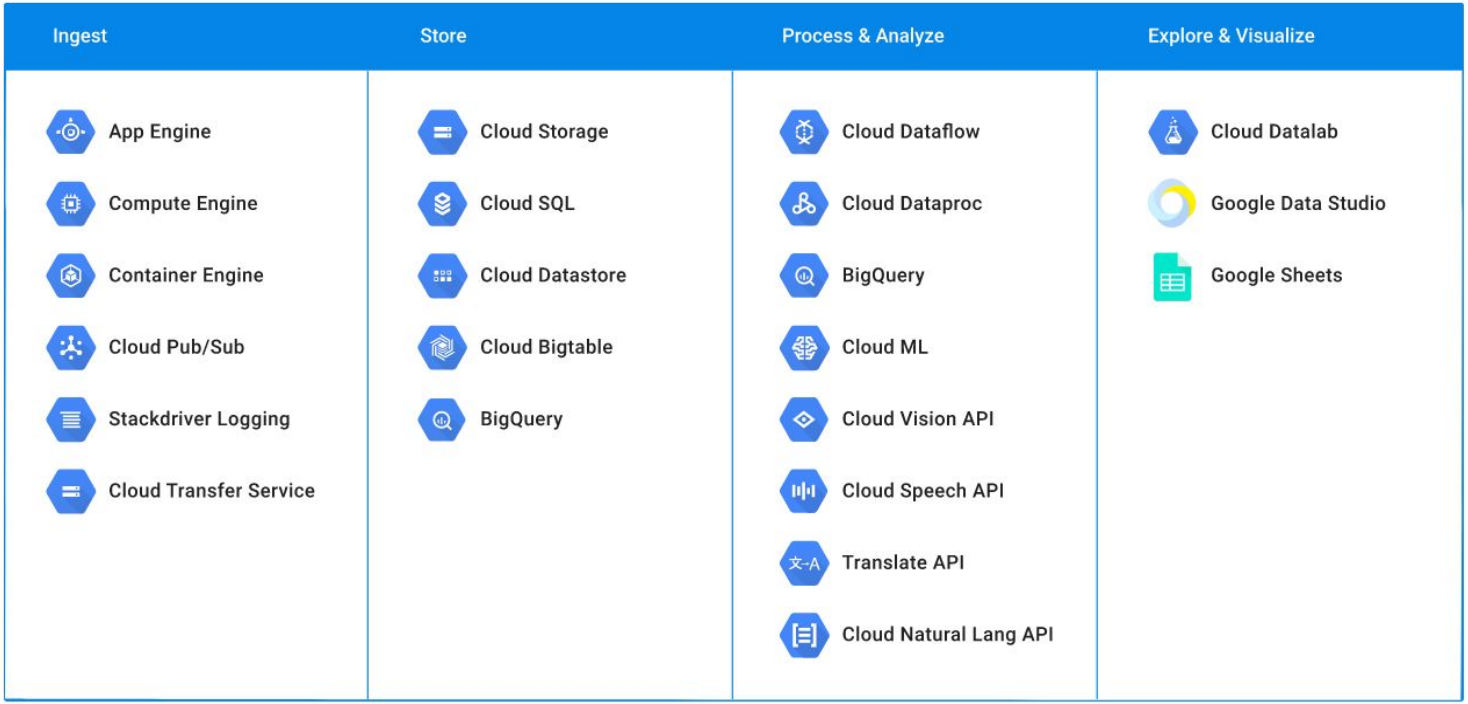
In this stage, the data is transformed from raw form into actionable information.

Explore & Visualize

The final stage is to convert the results of the analysis into a format that is easy to draw insights from and to share with colleagues and peers.

How to manage data in GCP?

At each stage, GCP provides multiple services to manage your data. This means you can select a set of services tailored to your data and workflow.



What are some key considerations when designing storage?

When designing storage for applications, it is important to adhere to key considerations and best practices to ensure optimum performance of the data.

Co-locate data and compute resources

Shuttling **large volumes of data** to compute enclaves is **inefficient** and **yields poor performance**

Prefer SSD for latency-sensitive workloads

If using Compute Engine instances for **latency-sensitive workloads**, local **SSD** offers **large performance gains** over HDD.

Replicate for resiliency

Replication of data to **multiple geographies** is often a prerequisite for an **effective disaster recovery** and **business continuity** strategy

Solve security

Both users and administrators need to be comfortable with the **security and privacy model** for **sensitive data**. Define a plan and **test well**.