



Welcome to Cloud OnBoard

Application Development with GCP

Google Cloud



Welcome

Chandra B

Google Cloud Authorised Trainer



Module 1

Introduction to Application Development



Developing Applications with GCP

Best Practices for Application Development

Build for the cloud



Global Reach



Scalability and
High Availability



Security

Implement best practices to build scalable, more secure, and highly available applications



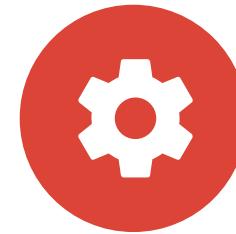
Manage your application's code and environment



Code Repository



Dependency Management



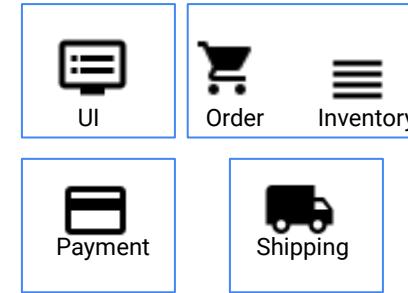
Configuration Settings

Consider implementing microservices



Monolithic application

- Codebase becomes large.
- Packages have tangled dependencies.

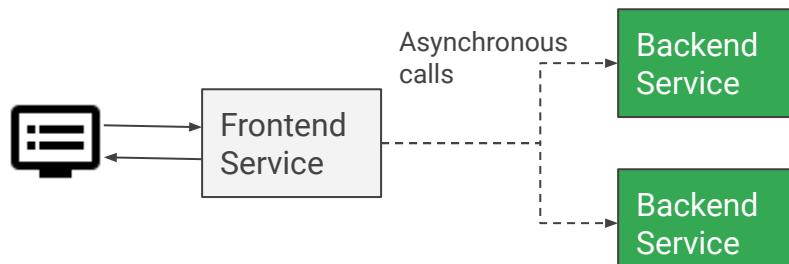


Microservices

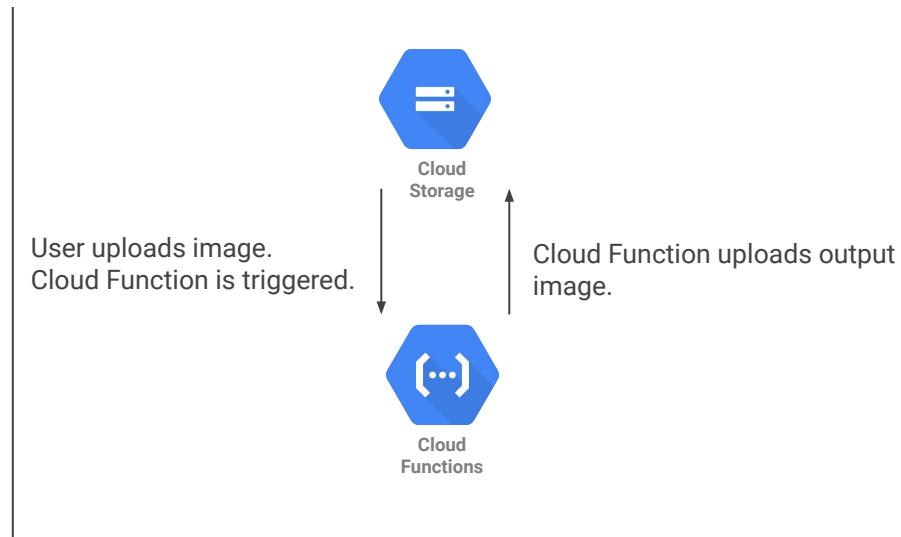
- Service boundaries match business boundaries.
- Codebase is modular.
- Each service can be independently updated, deployed, and scaled.

Perform asynchronous operations

Keep UI responsive; perform backend operations asynchronously.

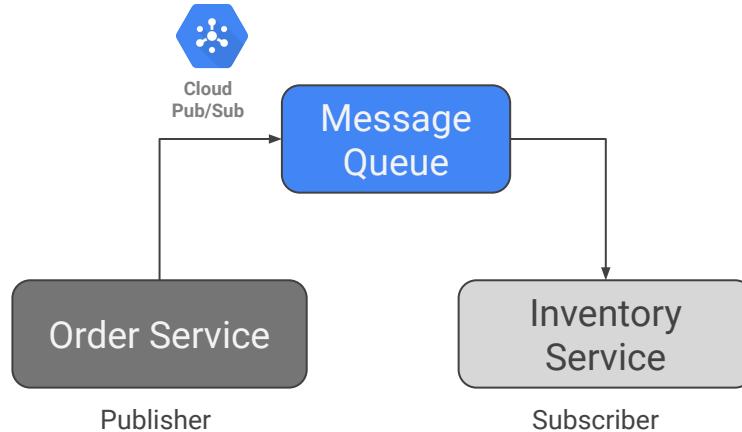


Use event-driven processing.

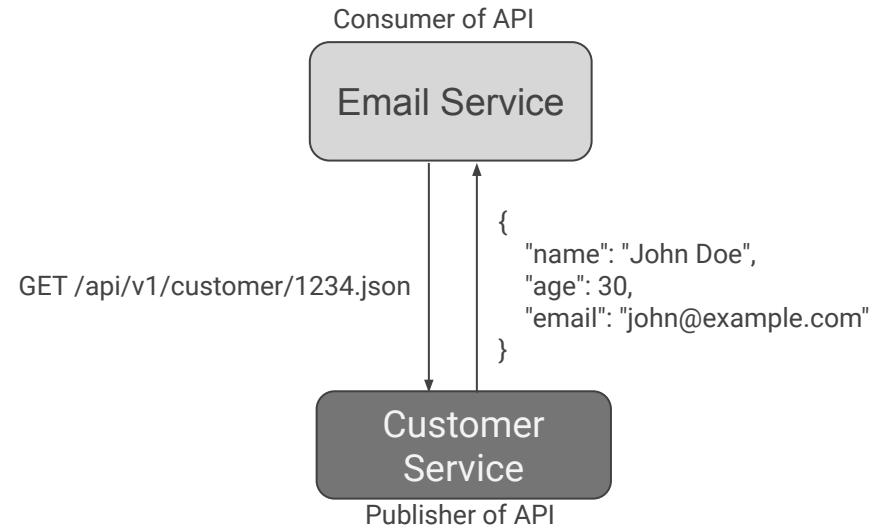


Design for loose coupling

Publishers and subscribers are loosely coupled.



Consumers of HTTP APIs should bind loosely with publisher payloads.



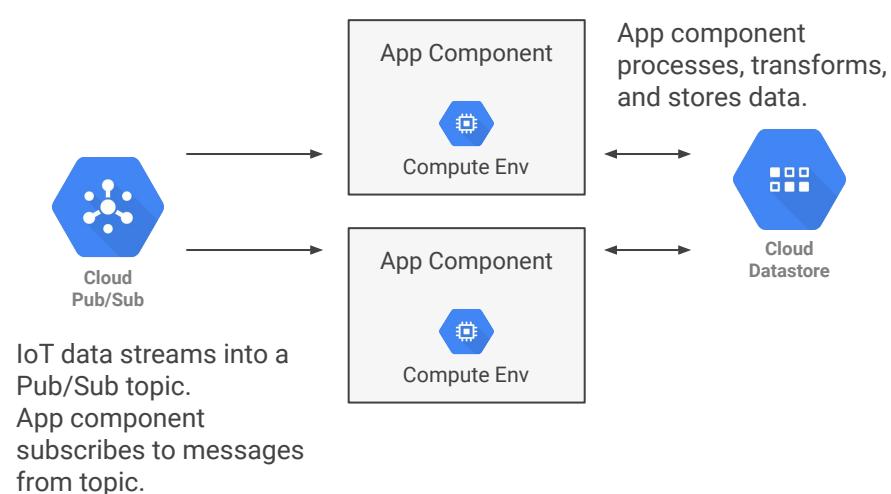
Implement stateless components for scalability



Worker
pattern

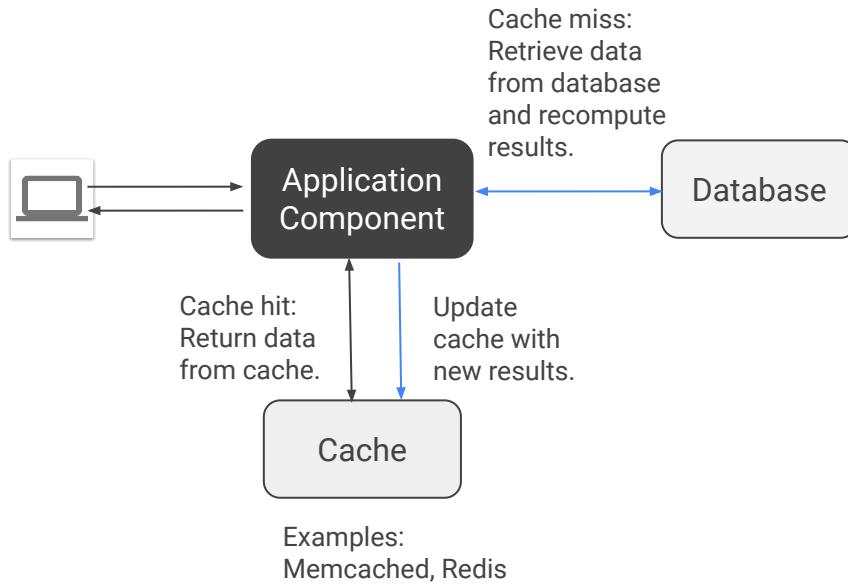
Workers perform compute tasks without sharing state.
Workers can scale up and down reliably.

IoT data streams into a Pub/Sub topic.
Cloud Function is triggered.



Cache content

Cache application data.



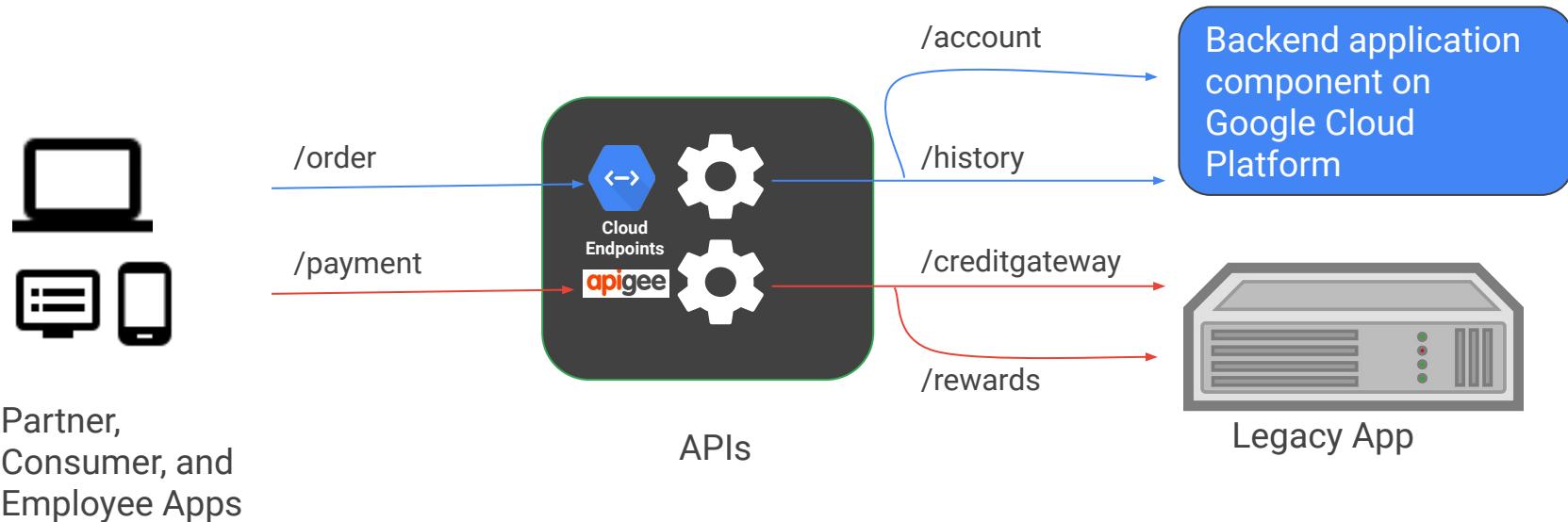
Cache frontend content.



Cloud CDN

- Cache load-balanced frontend content that comes from Compute Engine VM instance groups.
- Cache static content that is served from Cloud Storage.

Implement API gateways to make backend functionality available to consumer applications



Use federated identity management

Sign in with Google

Sign in with Facebook

Sign in with Twitter

Sign in with GitHub

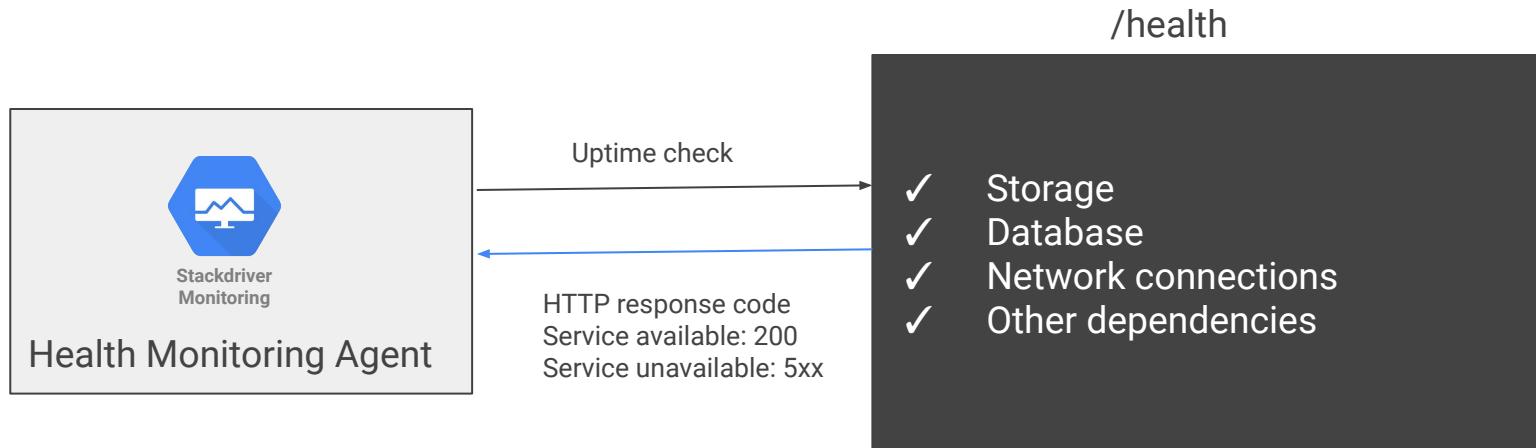
Sign in with email



Firebase Authentication

Authenticate users by
using external identity
providers.

Implement health-check endpoints



Set up logging and monitor your application's performance

App Component
`print`

App Component
`console.log`

App Component



Stackdriver



- Debugging
- Error reporting
- Tracing
- Logs-based metrics
- Monitoring

Handle transient and long-lasting errors gracefully



Transient errors:
Retry with exponential backoff.



Service availability errors:
Implement a circuit breaker.

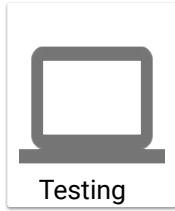
Consider data sovereignty and compliance requirements

EU-U.S. & Swiss-U.S.
Privacy Shield Framework



Perform high availability testing and develop disaster recovery plans

In addition to functional and performance testing, perform high-availability testing and develop disaster recovery plans.



- Identify failure scenarios.
- Create disaster recovery plans (people, processes, tools).
- Perform tabletop tests.
- Perform canary testing and blue/green deployments.
- Validate your disaster recovery plan.

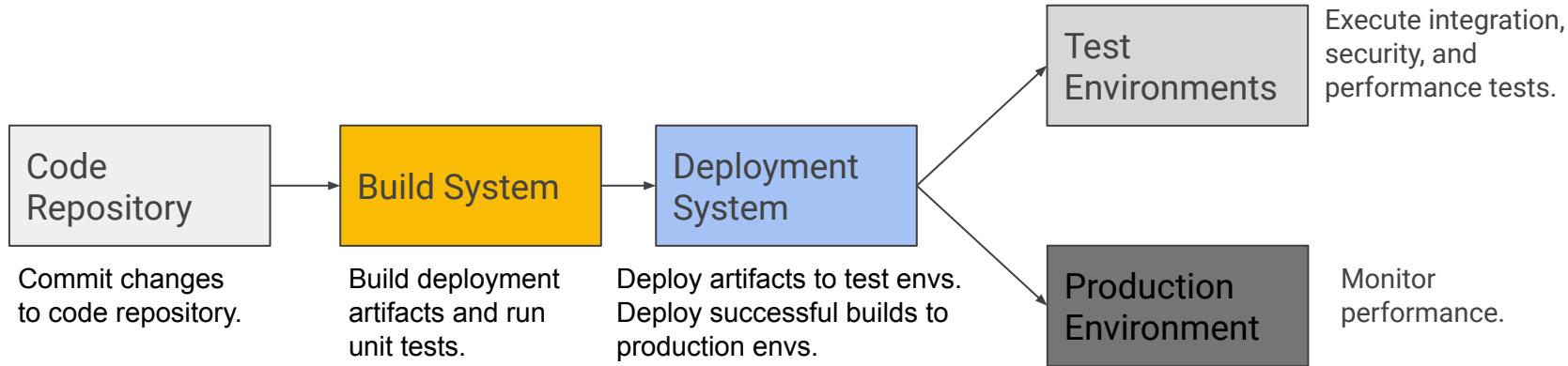
Example failure scenarios:

- Connectivity failure
- On-premises data center or other cloud-provider failure
- GCP zonal or regional failure
- Deployment rollback
- Data corruption caused by network or application issues

Implement continuous integration and delivery pipelines

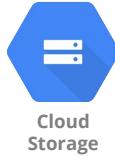
Continuous Integration

Continuous Delivery



A green wavy banner with the text "SecDevOps" and "Automate security checks".

GCP provides a full suite of storage service options



Cloud
Storage



Cloud
Datastore



Cloud
Bigtable



Cloud
SQL



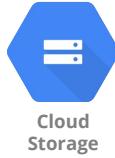
Cloud
Spanner



BigQuery

- Cost-effective
- Varied choices based on your:
 - Application
 - Workload

Cloud Storage



Cloud
Storage



Cloud
Datastore



Cloud
Bigtable



Cloud
SQL



Cloud
Spanner



BigQuery

Overview	Ideal for
<ul style="list-style-type: none">• Fully managed, highly reliable• Cost-efficient, scalable object/blob store• Objects access via HTTP requests• Object name is the only key	<ul style="list-style-type: none">• Images and videos• Objects and blobs• Unstructured data• Static website hosting

Cloud Datastore

Cloud
StorageCloud
DatastoreCloud
BigtableCloud
SQLCloud
Spanner

BigQuery

Overview	Ideal for
<ul style="list-style-type: none">• Fully managed NoSQL document database• Scalable	<ul style="list-style-type: none">• Semi-structured application data• Durable key-value data• Hierarchical data• Managing multiple indexes• Transactions

Cloud Bigtable



Overview	Ideal for
<ul style="list-style-type: none">High performance wide column NoSQL database serviceSparsely populated tableCan scale to billions of rows and thousands of columnsCan store TB to PB of data	<ul style="list-style-type: none">Operational applicationsAnalytical applicationsStoring large amounts of single-keyed dataMapReduce operations

Cloud SQL



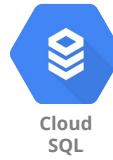
Cloud Storage



Cloud Datastore



Cloud Bigtable



Cloud SQL



Cloud Spanner



BigQuery

Overview	Ideal for
<ul style="list-style-type: none">● Managed service<ul style="list-style-type: none">○ Replication○ Failover○ Backups● MySQL and PostgreSQL● Relational database service● Proxy allows for secure access to your Cloud SQL Second Generation instances without whitelisting	<ul style="list-style-type: none">● Web frameworks● Structured data● OLTP workloads● Applications using MySQL/PGS

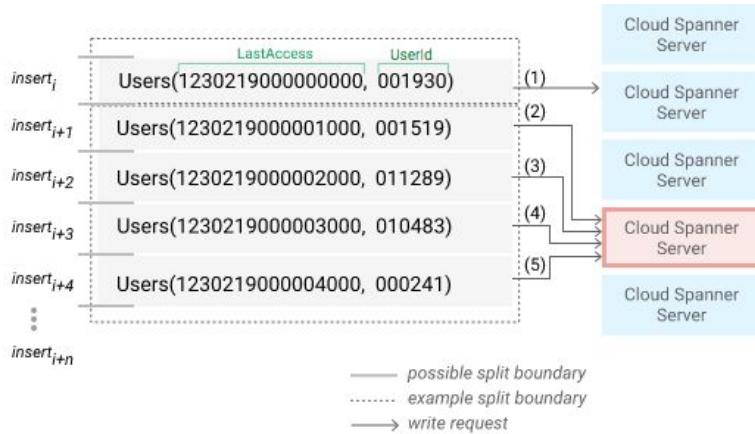
Cloud Spanner



Overview	Ideal for
<ul style="list-style-type: none">● Mission-critical relational database service● Transactional consistency● Global scale● High availability● Multi-region replication● 99.999% SLA	<ul style="list-style-type: none">● Mission-critical applications● High transactions● Scale and consistency requirements

Spanner Considerations

Avoid monotonically increasing keys.



Make sure writes are well distributed and load the data using multiple workers.

Use interleaved tables to establish hierarchy.

```
-- Schema hierarchy:
-- + Singers
--   + Albums (interleaved table, child table of Singers)

CREATE TABLE Singers (
    SingerId INT64 NOT NULL,
    FirstName STRING(1024),
    LastName STRING(1024),
    SingerInfo BYTES(MAX),
) PRIMARY KEY (SingerId);

CREATE TABLE Albums (
    SingerId INT64 NOT NULL,
    AlbumId INT64 NOT NULL,
    AlbumTitle STRING(MAX),
) PRIMARY KEY (SingerId, AlbumId),
INTERLEAVE IN PARENT Singers ON DELETE CASCADE;
```

Don't create non-interleaved indexes on columns with monotonically increasing or decreasing keys.

BigQuery



Overview	Ideal for
<ul style="list-style-type: none">● Low-cost enterprise data warehouse for analytics● Fully managed● Petabyte scale● Fast response times● Serverless	<ul style="list-style-type: none">● Online Analytic Processing (OLAP) workloads● Big data exploration and processing● Reporting via Business Intelligence (BI) tools

Run Microsoft SQL Server on GCP

- You can run SQL Server images on Google Compute Engine.
- Compute Engine VMs can be preloaded with SQL Server.
- Licensing from Microsoft is included automatically.
- Supported versions include:
 - SQL Server Standard
 - SQL Server Web
 - SQL Server Enterprise

Storage Options for Mobile

	Cloud Storage for Firebase	Firebase Realtime Database	Firebase Hosting
Overview	<ul style="list-style-type: none"> ● Mobile and web access to Google Cloud Storage ● Serverless third-party authentication and authorization 	<ul style="list-style-type: none"> ● Realtime ● NoSQL JSON database 	<ul style="list-style-type: none"> ● Web and mobile content hosting ● Production-grade
Ideal for	<ul style="list-style-type: none"> ● Images, pictures, and videos ● Objects and blobs ● Unstructured data 	<ul style="list-style-type: none"> ● Mobile and web applications ● Realtime 	<ul style="list-style-type: none"> ● Atomic release management ● JS app support ● Firebase integration

Cache your Application Data

You can use Redis Labs Memcached Cloud to cache application data on GCP

- Fully managed third-party service
- Offers a variety of options for running Memcached
- 30 MB of cache storage provided at no charge

Storage at a glance

Product	Simple Description	Ideal for	Not Ideal for
 Cloud Storage	Binary/object store	Large or rarely accessed unstructured data	Structured data, building fast apps
 Datastore	Scalable store for structured serve	GAE apps, structured pure-serve use cases	Relational or analytic data
 Bigtable	High-volume, low-latency database	"Flat," heavy read/write, or analytical data	High structure or transactional data
 CloudSQL	Well-understood VM-based RDBMS	Web frameworks, existing applications	Scaling, analytics, heavy writes
 Spanner	Relational DB service	Low-latency transactional systems	Analytic data
 BigQuery	Auto-scaling analytic data warehouse	Interactive analysis of static datasets	Building fast apps

Technical considerations for storage options

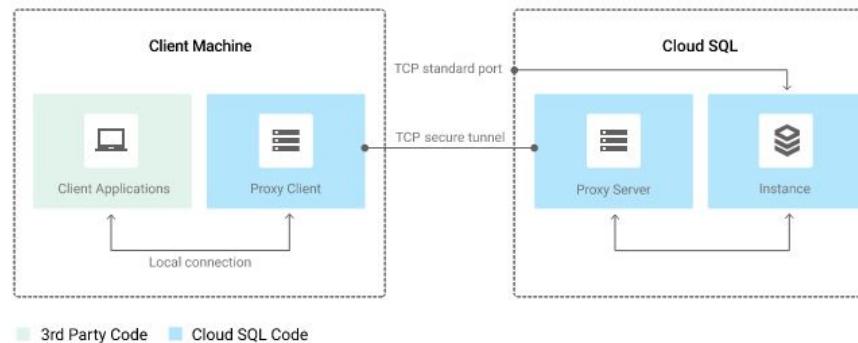
Product	R/W Latency	Typical Size	Storage Type
 Cloud Storage	Medium (100s of ms)	Any	Object
 Datastore	Medium (10s of ms)	< 200 TB	Document
 Bigtable	Low (ms)	2TB–10 PB	Key-Value
 CloudSQL	Low (ms)	< 10 TB	Relational
 Spanner	Low (ms)	Any	Relational
 BigQuery	High (s)	Any	Columnar

Connecting Securely to a Cloud SQL Database by using Cloud SQL Proxy

Why Cloud SQL Proxy?

Access your Cloud SQL Second Generation instance without having to:

- Whitelist IP addresses.
- Configure SSL.



Demo: Connecting Securely to a Cloud SQL Database

1. Create a Cloud SQL instance
2. Connect to Cloud SQL from Cloud Shell using gcloud sql
3. Create a Compute Engine instance
4. Connect to Cloud SQL using the Cloud SQL Proxy

Module 2

Storage & Monitoring



Developing Applications with GCP

Performing Operations on Buckets and Objects

Google Cloud Storage Concepts

Resources are entities in Google Cloud Platform including:

- Projects
- Buckets - the basic Cloud Storage container
- Objects - the individual pieces of data that you store in Google Cloud Storage

Storage classes

Storage Class	Characteristics	Use Cases	Price (per GB per month)	Name for APIs
Multi-Regional Storage	<ul style="list-style-type: none"> ● 99.95% availability ● Geo-redundant 	<ul style="list-style-type: none"> ● Serving website content ● Streaming videos ● Mobile apps 	\$0.026	multi_regional
Regional Storage	<ul style="list-style-type: none"> ● 99.9% availability ● Data stored in a narrow geographic region 	<ul style="list-style-type: none"> ● Data analytics 	\$0.02	regional
Nearline Storage	<ul style="list-style-type: none"> ● 99.0% availability ● Data retrieval costs ● Higher per-operation costs ● 30-day minimum storage duration 	<ul style="list-style-type: none"> ● Back-up ● Serving long-tail multimedia content 	\$0.01	nearline
Coldline Storage	<ul style="list-style-type: none"> ● 99.0% availability ● Data retrieval costs ● higher per-operation costs ● 90-day minimum storage duration 	<ul style="list-style-type: none"> ● Disaster recovery ● Data archiving 	\$0.007	coldline

Demo: Explore Cloud Storage

1. Create a Cloud Storage bucket in the Console.
2. Upload files to Cloud Storage in the Console.
3. Manipulate Cloud Storage using Cloud Shell.
4. Enable and manage object versioning.

The following operations are strongly consistent

Read-after-write

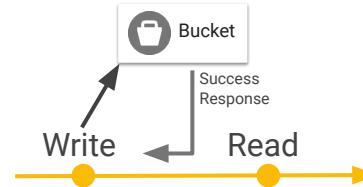
Read-after-metadata-update

Read-after-delete

Bucket listing

Object listing

Granting access to resources

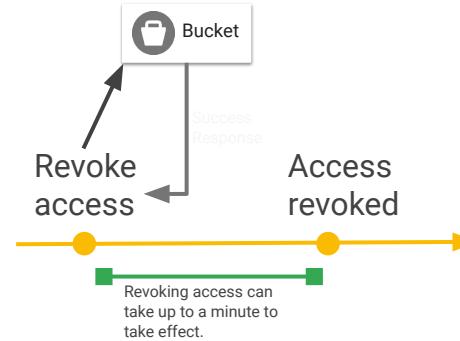


Strongly consistent: when you perform an operation in Cloud Storage and receive a success response, the object is *immediately available* for download and metadata operations.

The following operations are eventually consistent

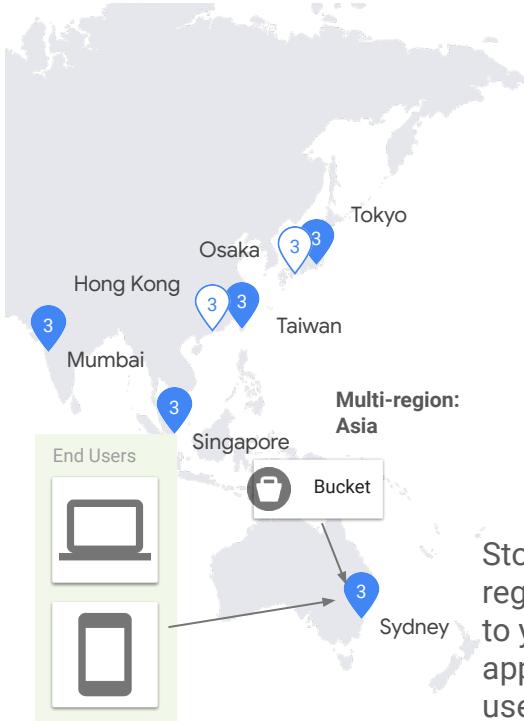
Revoking access from objects

Accessing publicly readable cached objects

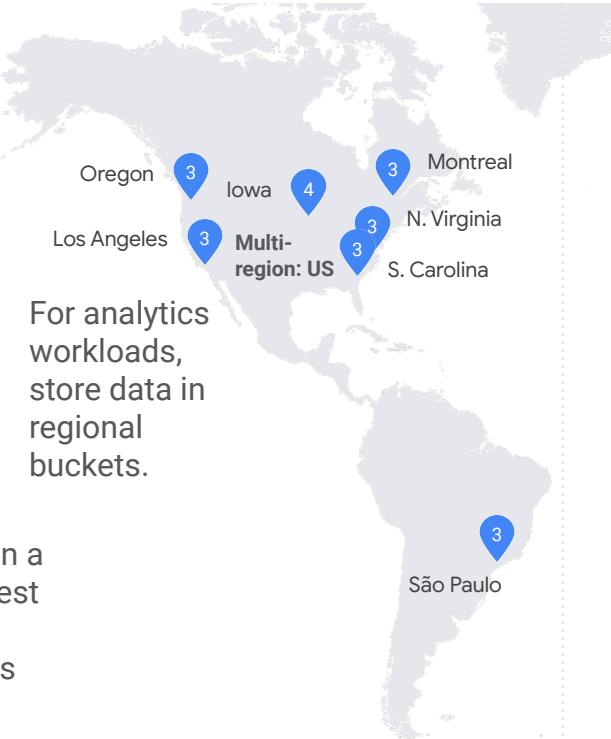


Eventually consistent: when you perform an operation, it may take some time for the operations to take effect.

Consider location and availability of your data for storage options



Store data in a region closest to your application's users.



For analytics workloads, store data in regional buckets.



Consider compliance requirements when choosing data location.



More location and availability considerations

- Multi-Regional and Regional Storage:
 - Provide the best availability.
 - Are good options for data served at a high rate with high availability.
- Nearline and Coldline Storage are good options for:
 - Infrequently accessed data.
 - Data that tolerates slightly lower availability.

Secure your buckets using the following options

Use Identity and Access Management (IAM) permissions to grant:



- Access to buckets
- Bulk access to a bucket's objects

Use Access Control Lists (ACLs) to grant:

- Read or write access to users for individual buckets or objects
- Access when fine-grained control over individual objects is required

Signed URLs (query string authentication):

- Provide time-limited read or write access to an object through a generated URL
- Can be created using gsutil or programmatically.

Use Signed Policy Documents to:

- Specify what can be uploaded to a bucket.
- Control size, content type, and other upload characteristics.

Firebase Security Rules provide:



- Granular, attribute-based access control to mobile and web apps using the Firebase SDKs for Cloud Storage

Developing Applications with GCP

Debugging & Monitoring

Google Stackdriver is a multi-cloud service



Error Reporting

Error notifications
Error dashboard



Debugger

Production debug snapshots
Conditional snapshots
IDE integration



Logging

Platform, system, and app logs
Log search/view/filter
Logs-based metrics



Monitoring

Platform, system, and app metrics
Uptime/health checks
Dashboards
Alerts



Trace

Latency reporting
Per-URL latency sampling



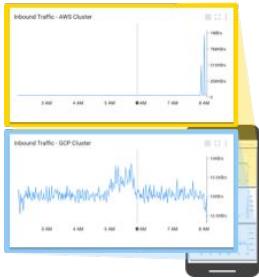
Profiler^{Beta}

Low-impact profiling of
applications in production



Google Cloud

Stackdriver enables you to increase application reliability



Monitor GCP, AWS, and Multi-Cloud Environments

Get the insight that you need with minimal configuration. Monitor hosted services and cloud architectures.

Identify Trends, Prevent Issues

Visualize trends via flexible charts and dashboards. Identify risks using scoring, anomaly detection, and prediction.

Reduce Monitoring Overhead

Spend less time correlating metrics, alerts, and logs across disparate systems. Don't worry about scaling tools.

Improve Signal-to-Noise

Reduce false positives and alert fatigue with advanced alerting designed for modern distributed systems.

Fix Problems Faster

Uptime and health checks notify you quickly when endpoints become inaccessible to your users. Drill down from alerts to dashboards to logs and traces to get to the root cause quickly.

Debugging Your Application

Debug your application in development and production

Stackdriver Error Reporting

Errors in the last hour

Occurrences	Error
 679	IndexOutOfBoundsException: Index: 4, Size: 4 com.callbyreference.demos.markov.Markov.isPromising (Markov.java)

Stack trace sample

[Parsed](#) [Raw](#)

```
java.lang.IndexOutOfBoundsException: Index: 4, Size: 4
...
    at com.callbyreference.demos.markov.Markov.isPromising (Markov.java:76)
    at com.callbyreference.demos.markov.Markov.generate (Markov.java:55)
    at com.callbyreference.demos.markov.Template$MarkovDiv.generateBegin (Template.java:113)
    at com.callbyreference.demos.markov.Template.generate (Template.java:20)
    at com.callbyreference.demos.markov.Template.generate (Template.java:22)
    at com.callbyreference.demos.markov.Template.generate (Template.java:22)
    at com.callbyreference.demos.markov.MarkovTemplate.generate (MarkovTemplate.java:57)
    at com.callbyreference.demos.markov.MarkovServlet doGet (MarkovServlet.java:115)
```

Debugger automatically creates debug snapshots

Stackdriver Debugger

```

65     private boolean isPromising(List<Object> markers) {
66         if (markers.size() > 32)
67             return false;
68         Object m0 = markers.get(0);
69         Object m_1 = markers.get(markers.size() - 1);
70         if (m0 == m_1) {
71             return false;
72         }
73         Object m1 = markers.get(1);
74         Object m2 = markers.get(2);
75         Object m3 = markers.get(3);
76         Object m4 = markers.get(4);
77         if (m0 != m1 || m0 != m2 || m0 != m3 || m0 != m4
78             || m0 != markers.get(5)) {
79             return false;

```

You do not have to restart the application to generate new debug snapshots.

Expressions: (Optional)

Type an expression

Variables



2017-09-13 (08:38:22)
View request logs

- ▶ this
- ▶ markers
- ▶ m0 "fairytales.txt"

Call Stack

com.callbyreference.demos.markov.Markov.isP...
Markov.java:76

Logging

Install Stackdriver Logging Agent to capture logs

Google Compute Engine

Amazon EC2

Stackdriver Logging is preconfigured in other compute environments

Google Cloud Dataflow

Google Cloud Functions

Google App Engine Flexible
and Standard Environments

Google Kubernetes Engine

Set up logs-based metrics and alerts



Stackdriver Logging and Monitoring

```
09:21:51.246 GET 200 1.13 KB6 exampleapp/  
- - [14/Sep/2017:09:21:51 -0700] "GET / HTTP/1.1" 200 1156 - "exampleapp"  
"exampleapp-git.appspot.com" ms=6 cpu_ms=11  
cpm_usd=1.291929999999998e-7 loading_request=0 instance=some_instance_id  
app_engine_release=1.9.54
```

[Expand all](#) | [Collapse all](#)

```
{  
  httpRequest: {  
    status: 200  
  }  
}
```

Create custom
logs-based metric:
HTTP_Success

Alert if **HTTP_Success**
metric is:
Below 400 per second
for 5 minutes

Demo: Logs-based metrics and alerts



Stackdriver Logging and Monitoring

```
09:21:51.246 GET 200 1.13 KB6 exampleapp/  
- - [14/Sep/2017:09:21:51 -0700] "GET / HTTP/1.1" 200 1156 - "exampleapp"  
"exampleapp-git.appspot.com" ms=6 cpu_ms=11  
cpm_usd=1.291929999999998e-7 loading_request=0 instance=some_instance_id  
app_engine_release=1.9.54
```

[Expand all](#) | [Collapse all](#)

```
{  
  httpRequest: {  
    status: 200  
  }  
}
```

Create custom
logs-based metric:
HTTP_Success

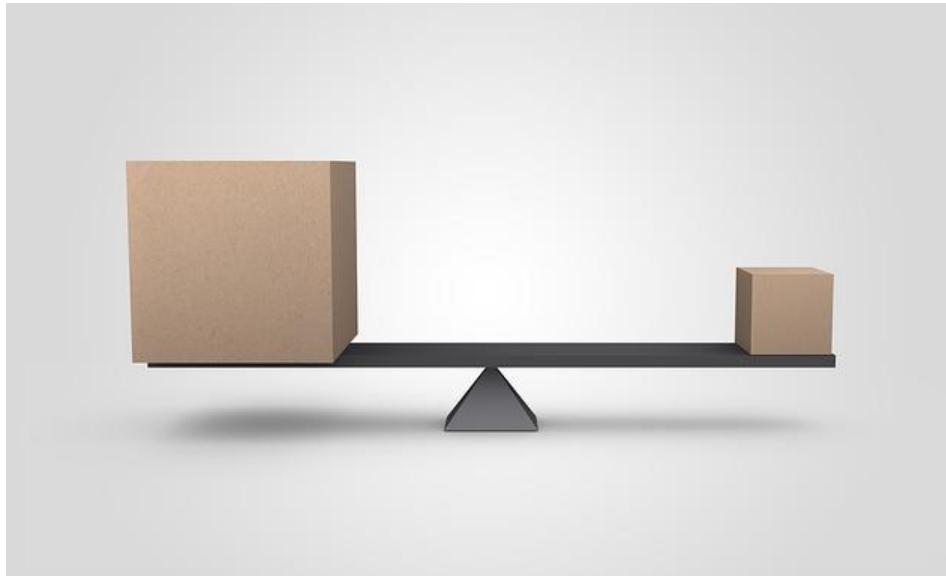
Alert if **HTTP_Success**
metric is:
Below 400 per second
for 5 minutes

Monitoring and Tuning Performance

Monitor to analyze long-term trends



Monitor to compare results over time or between experimental configurations



Monitor to raise alerts when something is broken or about to be broken



Monitor to perform ad hoc retrospective analysis



Identify APIs and resources that you want to monitor

Examples

- Public and private endpoints
- Multi-cloud resources such as Compute Engine VM instances, Cloud Storage buckets, Amazon EC2 instances, and Amazon RDS databases

Identify service-level indicators and objectives



Service-Level Indicator (SLI): Latency

Service-Level Objective (SLO): 99.9% of requests over 30 days have latency <100ms

Create dashboards that include four golden signals

Latency

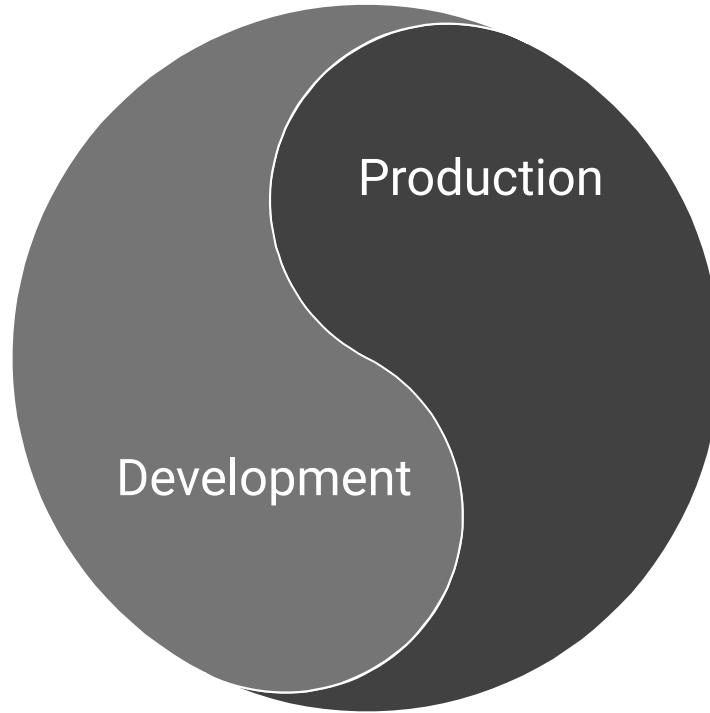
Traffic

Errors

Saturation

Identifying and Troubleshooting Performance Issues

Monitor performance in development and production



Add performance tests to your test suite

Development



Check performance watchpoints related to incoming requests

Web Authoring

Web page design and implementation

Cold-Boot Performance

Operations during initial boot of VM

Self-Inflicted Load

Service-to-service or browser-to-service calls

Review application code and logs

Application Errors

HTTP errors and other exceptions

Runtime Code Gen.

Aspect-oriented programming

Static Resources

Static web pages, images

Caching

Database retrieval and computation

One-at-a-Time Retrieval

Multiple serial requests

Error-Handling

Exponential backoff

Check performance watchpoints related to incoming requests in production

External User Load

Most frequent and slowest requests

Periodic Load

Traffic over an extended period of time

Malicious Load

Source of traffic is expected and legitimate

Production

Review deployment settings

Scaling

Autoscaling

Region

Source of traffic

Cron Jobs

Schedule

Production

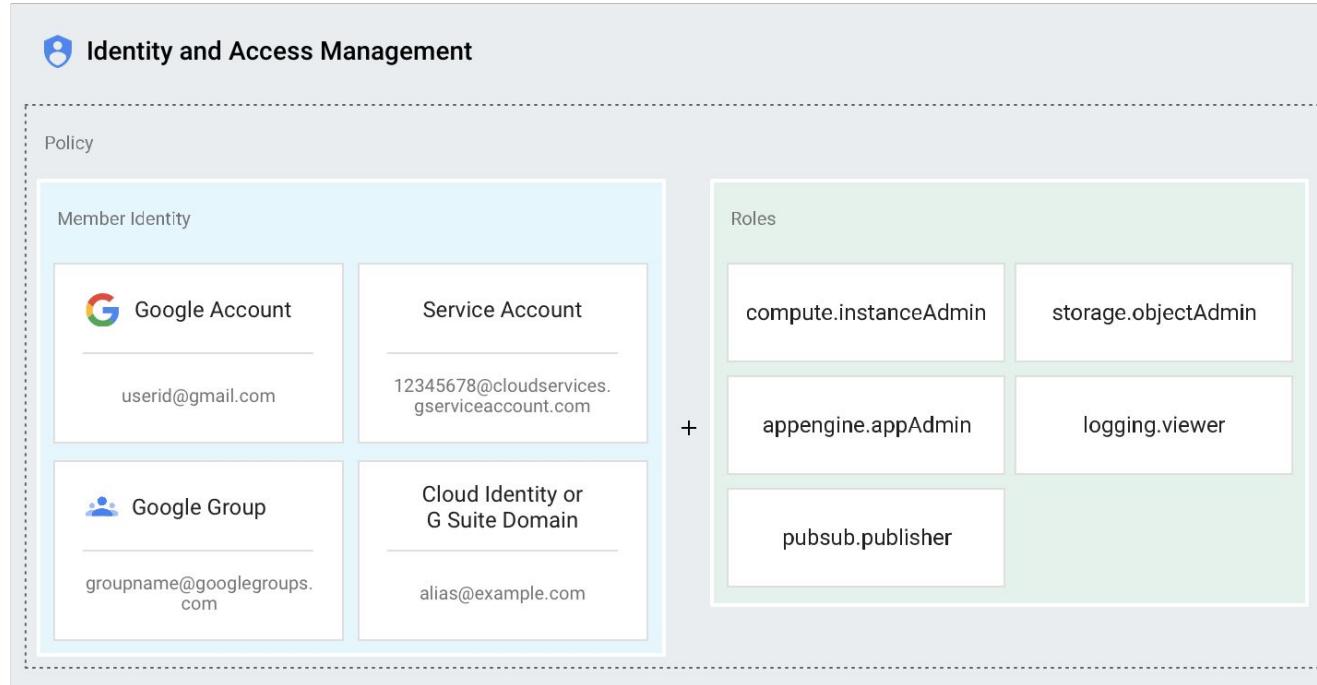
Traditional SRE blessing

May the queries flow, and the pager stay silent.

Developing Applications with GCP

Handling Authentication and Authorization

Cloud Identity and Access Management



Specify who has access with IAM members

Types of IAM members:

- Google account
- Service account
- Google group
- G Suite domain
- Cloud Identity domain

Specify what resources the members have access to

- Grant access to users for specific GCP resources
- Resources include:
- GCP projects
- Compute Engine instances
- Cloud Storage buckets
- Pub/Sub topics

Specify what *operations* are allowed on resources

Permissions are represented with the following syntax:

<service>.<resource>.<verb>

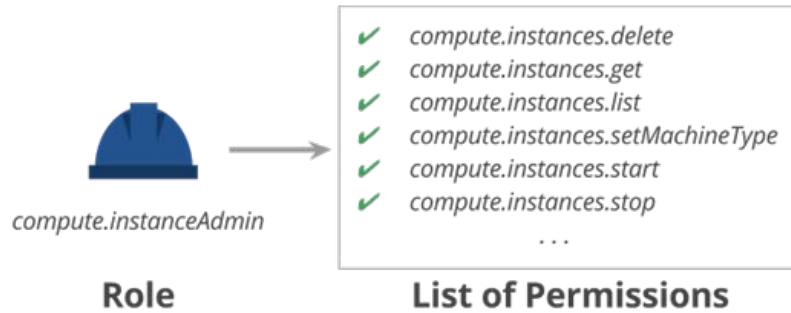
Examples:

pubsub.subscriptions.consume

storage.objects.list

compute.disktypes.list

Assign permissions using roles



There are three kinds of IAM roles:

Primitive
Predefined
Custom

Apply primitive roles at the project level

Role Name	Role Title	Permissions
roles/viewer	Viewer	Read-only actions that preserve state.
roles/editor	Editor	Viewer permissions plus actions that modify state.
roles/owner	Owner	Editor permissions plus ability to: <ul style="list-style-type: none">• Manage access control for a project and all its resources.• Set up billing for a project.

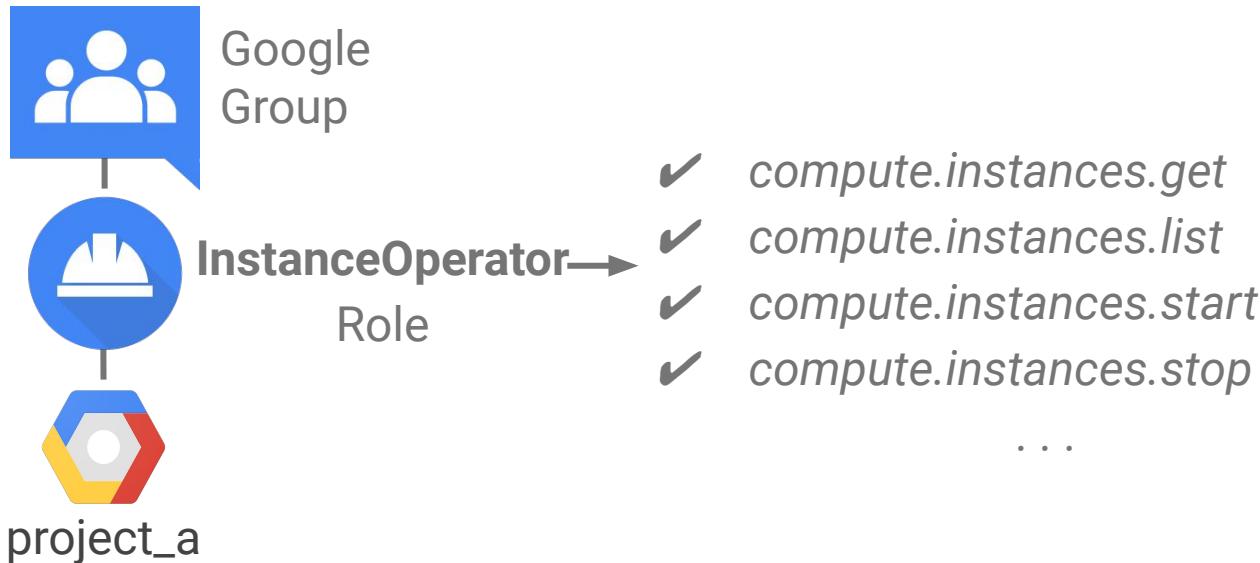
Apply predefined roles for granular access to GCP resources

Predefined roles give granular access to specific GCP resources.

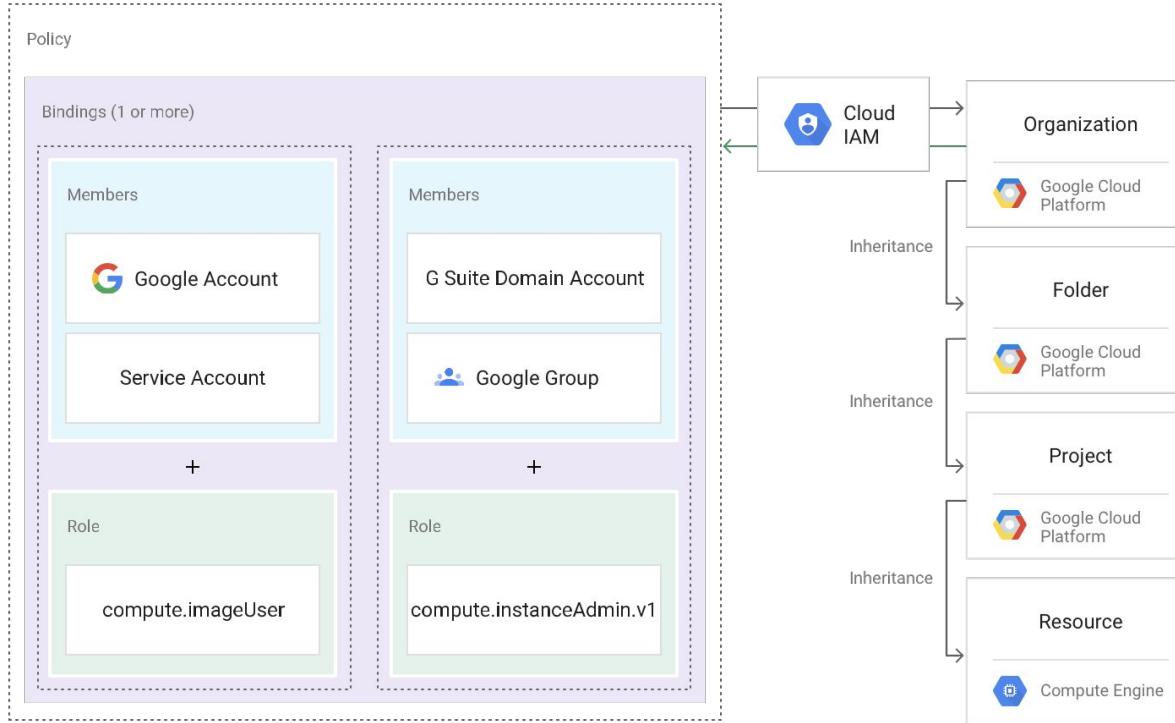
You can grant multiple roles to the same user.

Role Name	Role Title	Description	Resource Type
roles/bigtable.admin	Cloud Bigtable Admin	Administers all instances within a project, including the data stored in tables. Can create new instances. Intended for project administrators.	Organization Project Instance
roles/bigtable.user	Cloud Bigtable User	Provides read-write access to the data stored in tables. Intended for application developers or service accounts.	Organization Project Instance
roles/bigtable.reader	Cloud Bigtable Reader	Provides read-only access to the data stored in tables. Intended for data scientists, dashboard generators, and other data-analysis scenarios.	Organization Project Instance

IAM custom roles let you define a precise set of permissions



Define who has *what type* of access using policies



Cloud IAM Policy example

```
{  
  "bindings": [  
    {  
      "role": "roles/owner",  
      "members": [  
        "user:alice@example.com",  
        "group:admins@example.com",  
        "Domain:google.com",  
        "serviceAccount:my-other-app@appspot.gserviceaccount.com"  
      ]  
    },  
    {  
      "role": "roles/viewer",  
      "members": ["user:bob@example.com"]  
    }]  
}
```

Policy owners

Policy viewer

Cloud IAM API methods:

```
setIAMPolicy()  
getIAMPolicy()  
testIamPermissions()  
( )
```

Use service accounts to authenticate your applications when invoking Google APIs

Service accounts:

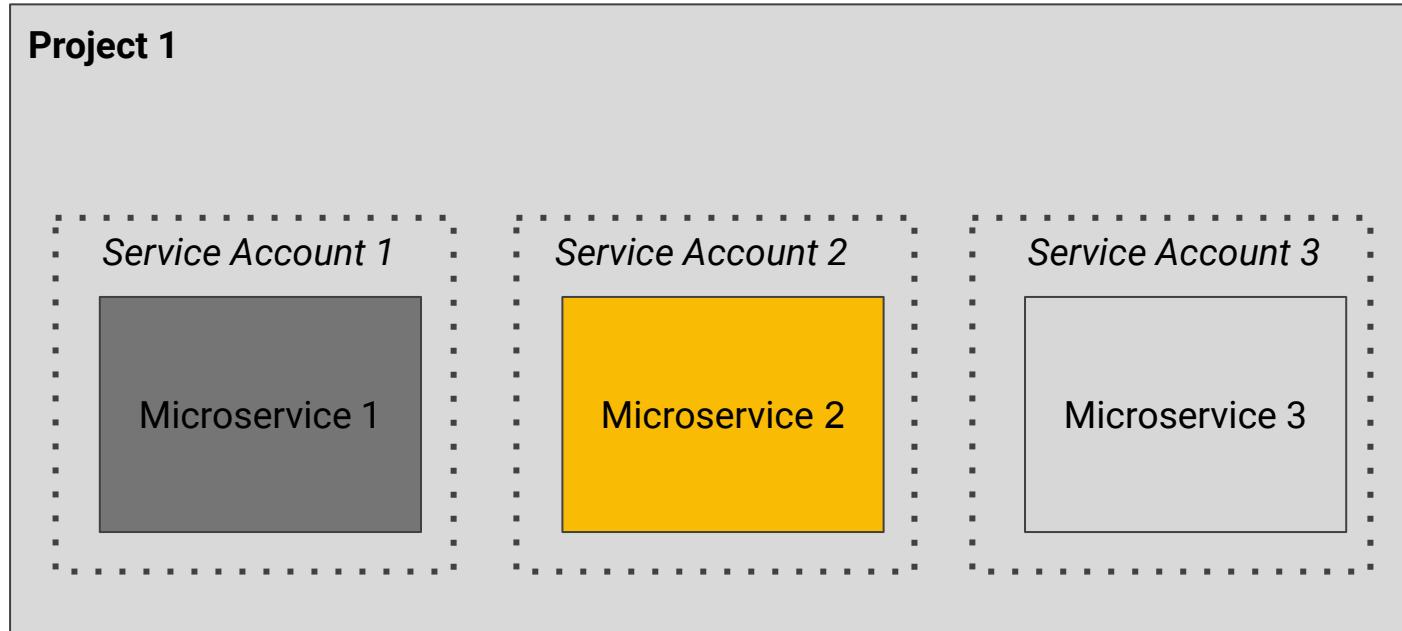
- Belong to your application or VM.
- Are used by your application to call the Google API or service so users aren't directly involved.
- Are identified by their unique email addresses.
- Are associated with a key pair.
- Can have up to 10 keys associated with them to facilitate key rotation (done daily by Google).
- Are supported by all GCP APIs.
- Enable authentication and authorization: you can assign specific IAM roles to a service account.

Use external keys for use from outside GCP

External keys:

- Can be created for use from outside GCP.
- Require that you be responsible for security of the private key and other management operations such as key rotation.
- Are manageable through the:
 - IAM API.
 - gcloud command-line tool.
 - Service Accounts page in the GCP Console.

You can define many service accounts in a project



Follow these steps to use a service account in your application

1. Create the service account via the console.
2. Generate and download your credentials file.
3. Set an environment variable to provide credentials to your application.
4. Authenticate in your code with the default credentials.

Linux or OS X:

```
export GOOGLE_APPLICATION_CREDENTIALS=<path_to_service_account_file>
```

Windows:

```
set GOOGLE_APPLICATION_CREDENTIALS=<path_to_service_account_file>
```

```
def implicit():
    from google.cloud import storage

    storage_client = storage.Client()

    # Make an authenticated API request
    buckets = list(storage_client.list_buckets())
    print(buckets)
```

If you don't specify credentials when constructing the client, the client library will look for credentials in the environment.

Use Application Default Credentials (ADC) to authenticate between applications

ADC checks for credentials in the following order:

1. Checks for

GOOGLE_APPLICATION_CREDENTIALS
environment variable.

2. Checks for default service accounts.

3. If 1 and 2 aren't found, an error is thrown.

If you don't specify credentials when constructing the client, the client library will look for credentials in the environment.

```
def implicit():

    from google.cloud import storage

    storage_client = storage.Client()

    buckets = list(storage_client.list_buckets())

    print(buckets)
```

Make an authenticated API request.

Module 3

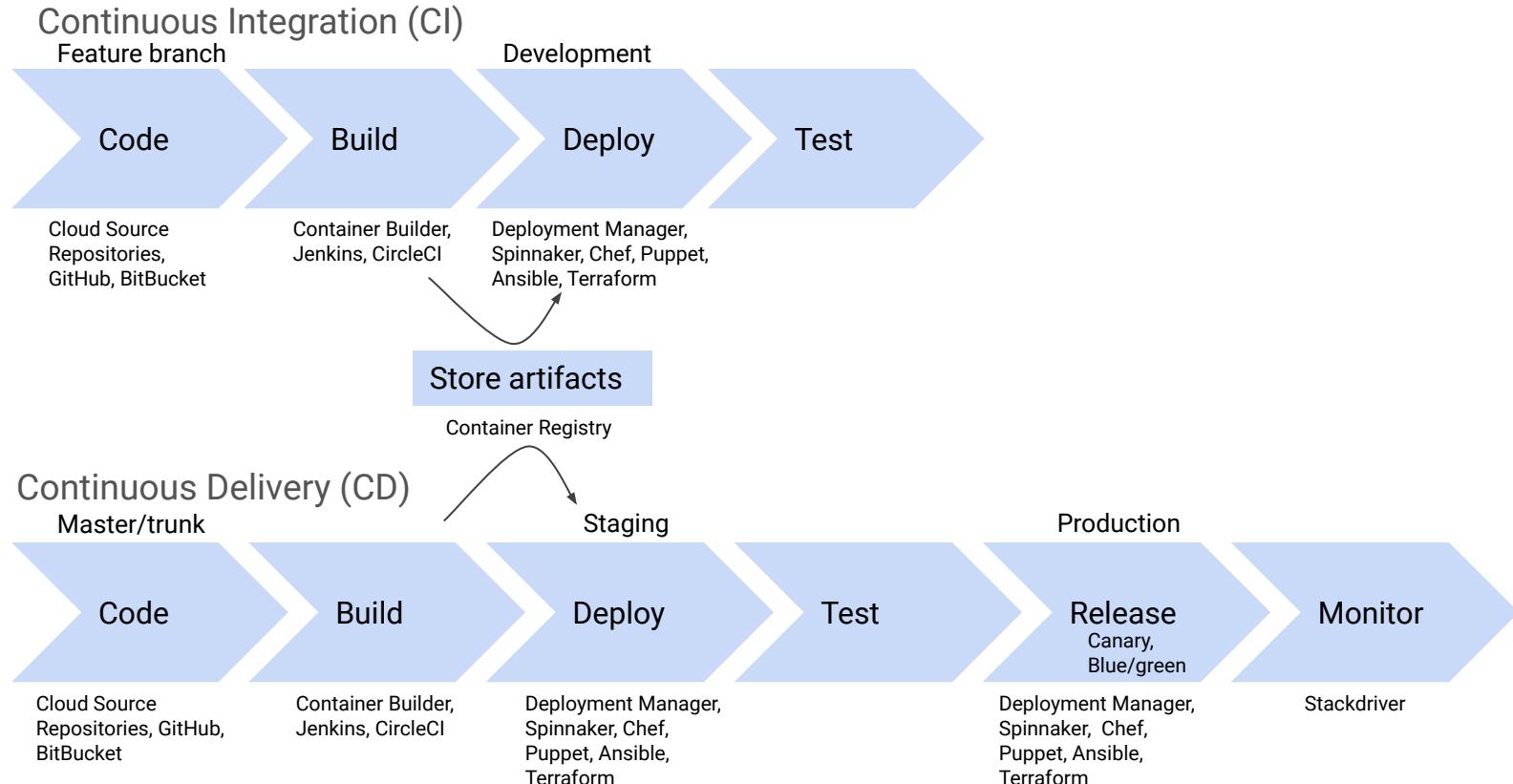
Compute



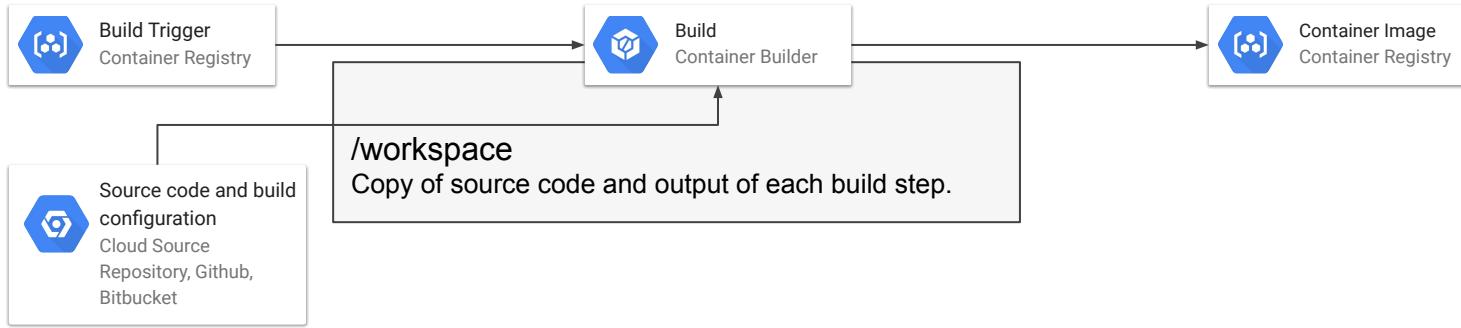
Developing Applications with GCP

Deploying Applications Using Containers

Implement continuous integration and delivery for reliable releases



Use Container Builder and Container Registry to create application images

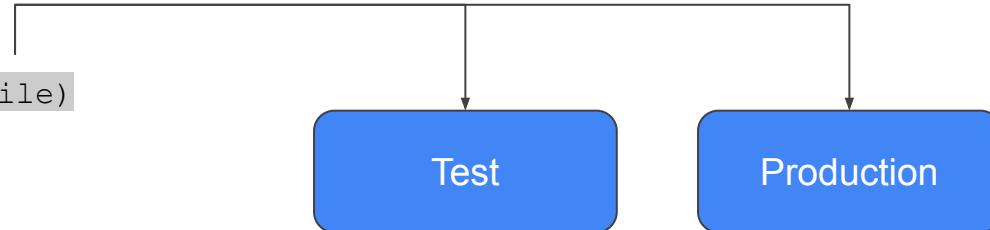


```
cloudbuild.yaml
steps:
- name: 'gcr.io/cloud-builders/docker'
  args: [ 'build', '-t', 'gcr.io/$PROJECT_ID/cb-demo-img', '.' ]
images:
- 'gcr.io/$PROJECT_ID/cb-demo-img'
tags:
- "test"
- "2.0b35"
```

Use Deployment Manager to launch your Google Cloud Platform resources

```
Deployment Configuration (YAML file)
imports:
- path: path/to/template.jinja
  name: my-template
- path: path/to/another/template.py
```

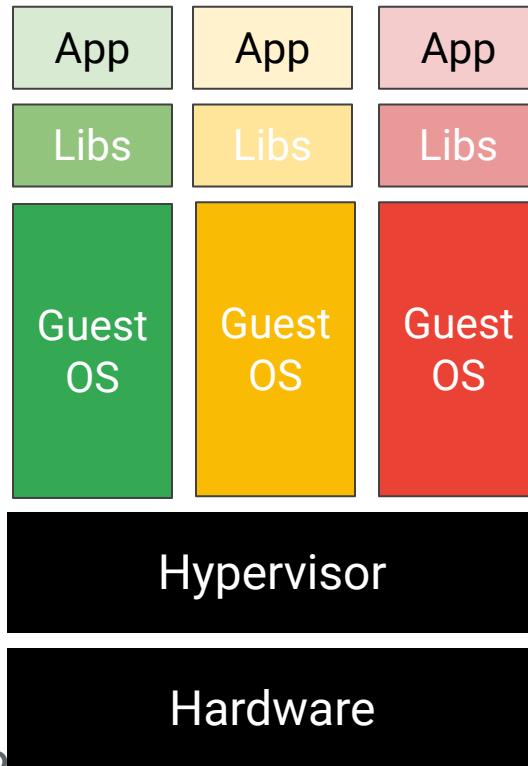
```
resources:
- name: resource-a
  type: type-of-resource
  properties:
    property-a: value
    property-b: value
  ...
```



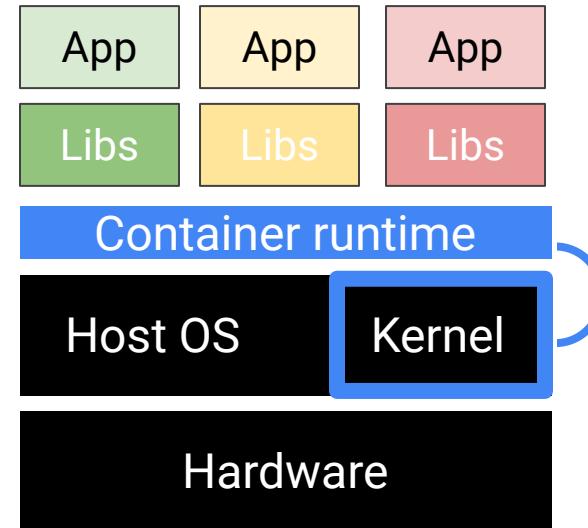
- ✓ Define template properties and environment variables.
- ✓ Define conditions to create different deployments based on a single configuration.
- ✓ Specify dependencies between resources.
- ✓ Specify startup scripts to run when VM launches.
- ✓ Define outputs.

Containers: an efficient way to isolate code and manage workloads

Hypervisor-based virtualization



Container-based virtualization



Why use containers?

Kubernetes Engine is GCP's fully managed system for running containers

Consistency

Across development, testing, and production environments

Loose coupling

Between application and operating system layers

Workload migration

Simplified between on-premises and cloud environments

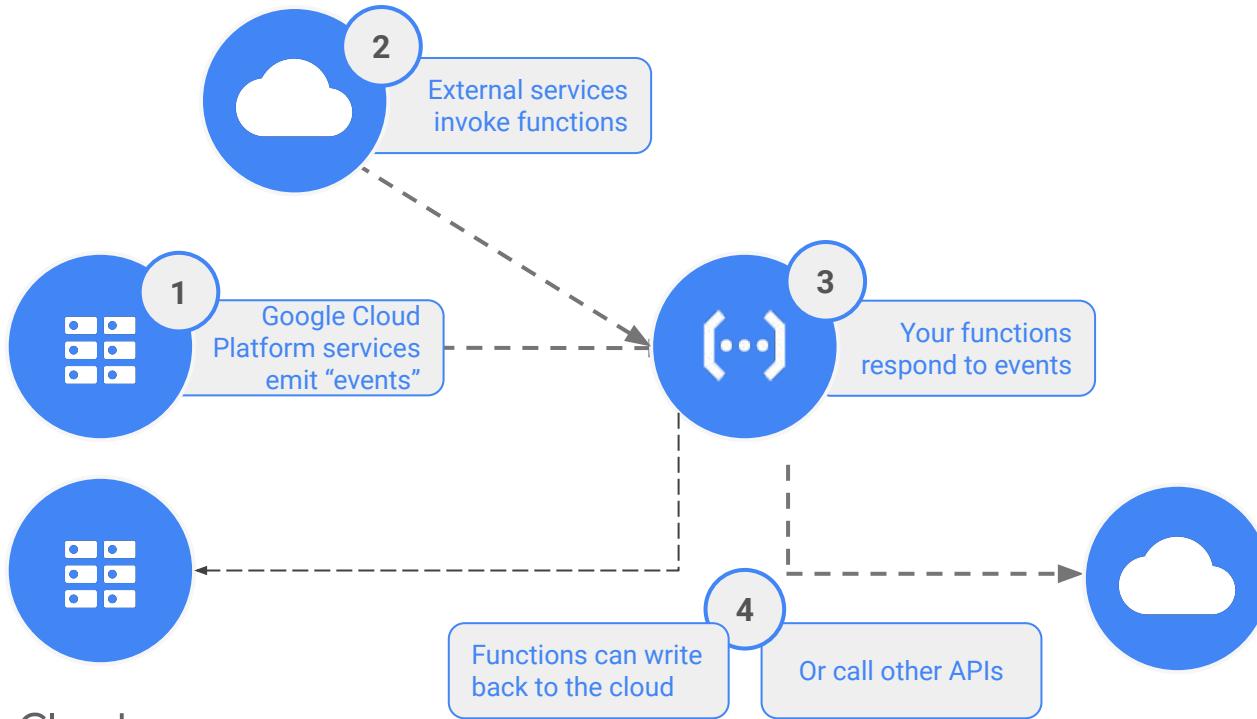
Agility

Agile development and operations

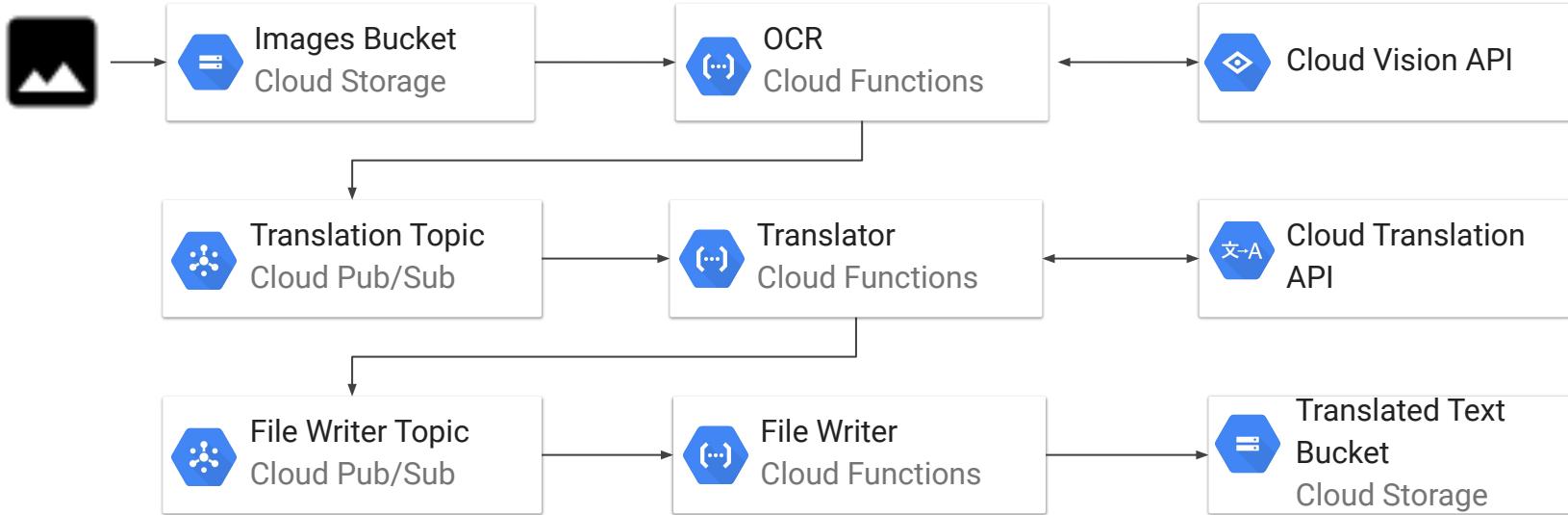
Developing Applications with GCP

Using Google Cloud Functions for Event-Driven Processing

Overview

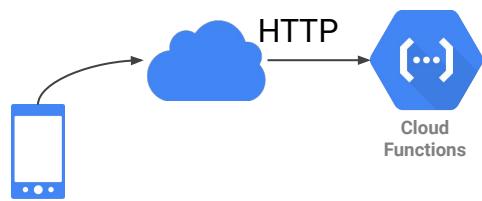


Cloud Functions enable event-driven, serverless, highly scalable microservices

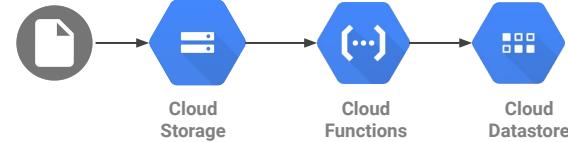


Use cases

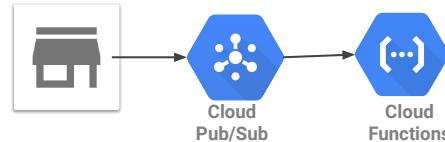
Webhooks



Lightweight ETL

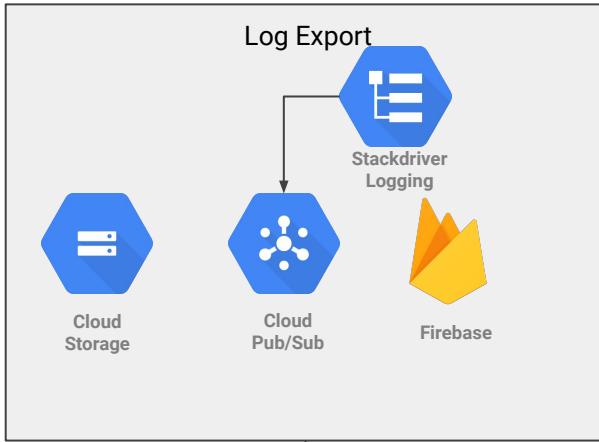


IoT



Cloud Functions can have asynchronous and synchronous triggers

Background Function—Asynchronous

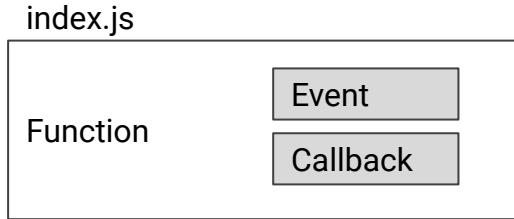


HTTP Function—Synchronous

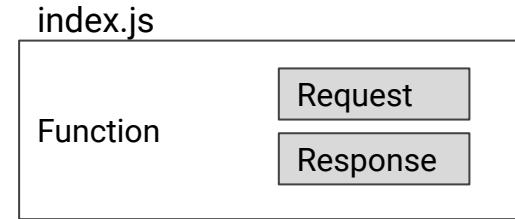


Write your Cloud Function

Background Function

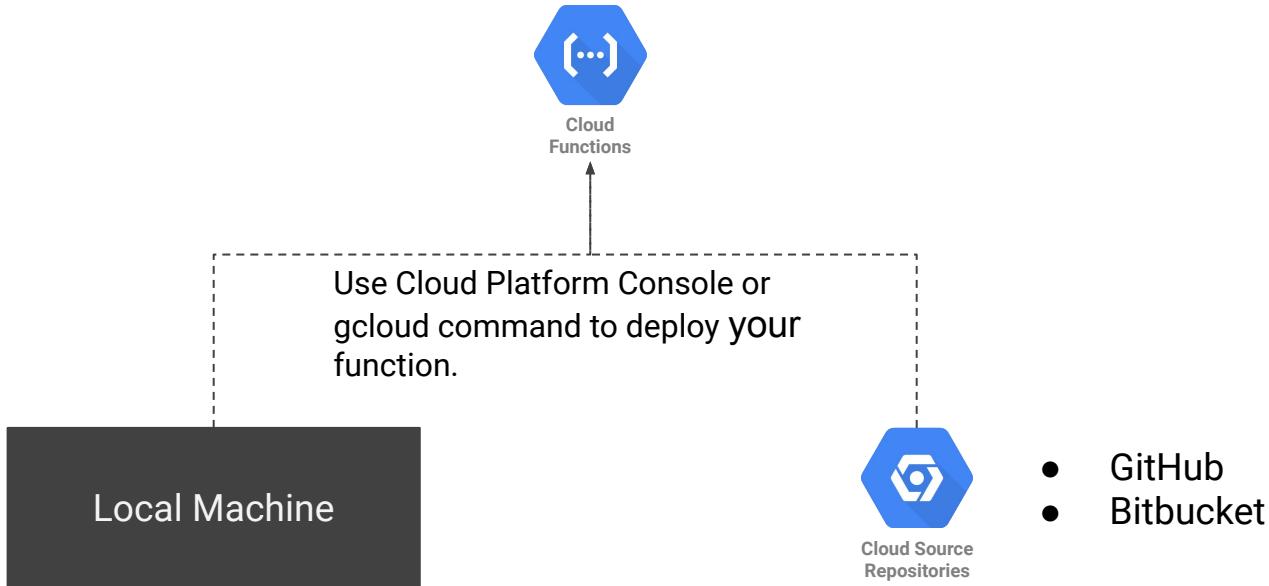


HTTP Function



Specify dependencies in a package.json file

Deploy your Cloud Function



Cloud Functions supports logging, error reporting, and monitoring

INFO log level: `console.log(...)`

ERROR log level: `console.error(...)`

DEBUG log level: internal system messages



Stackdriver
Logging

Errors thrown or reported manually



Stackdriver
Error Reporting

Number of invocations, execution time, memory usage



Cloud
Functions

Developing Applications with GCP

App Engine Flexible Environment

Deploy scalable web and mobile backends in any language with App Engine flexible environment

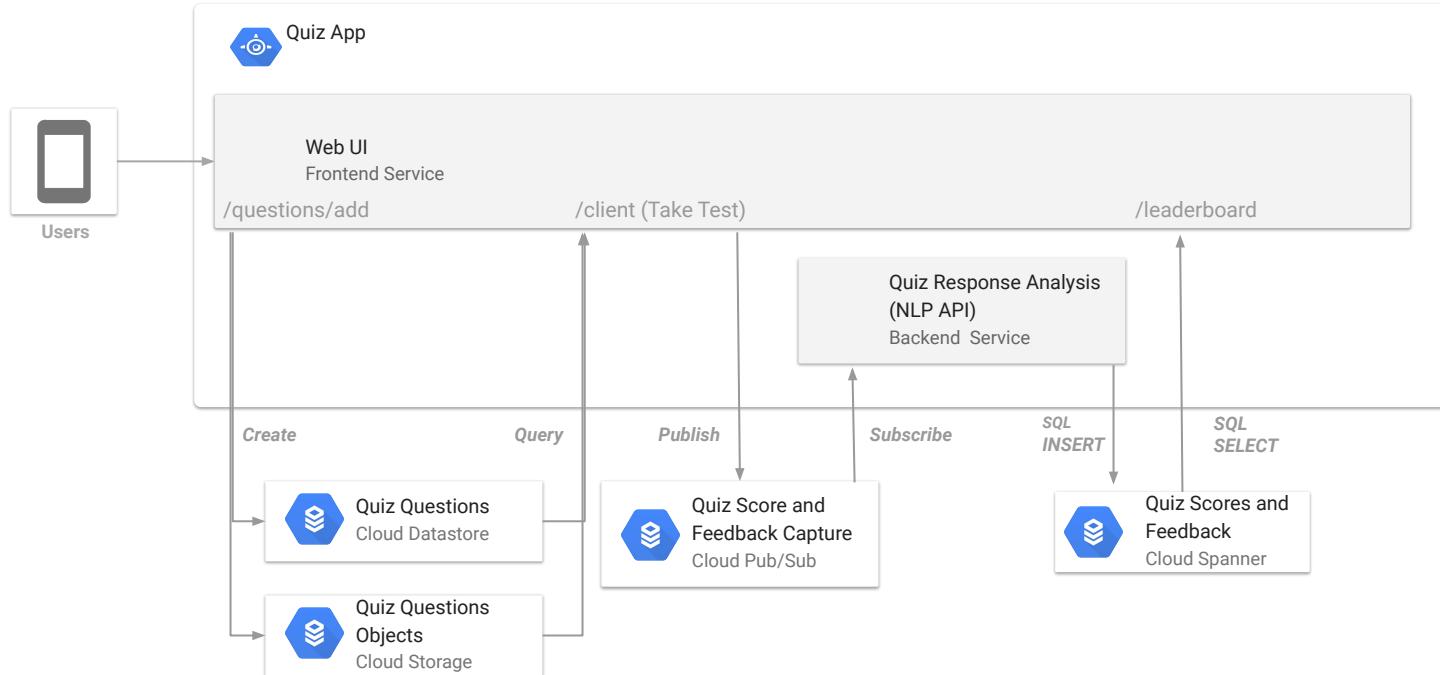


Native runtimes

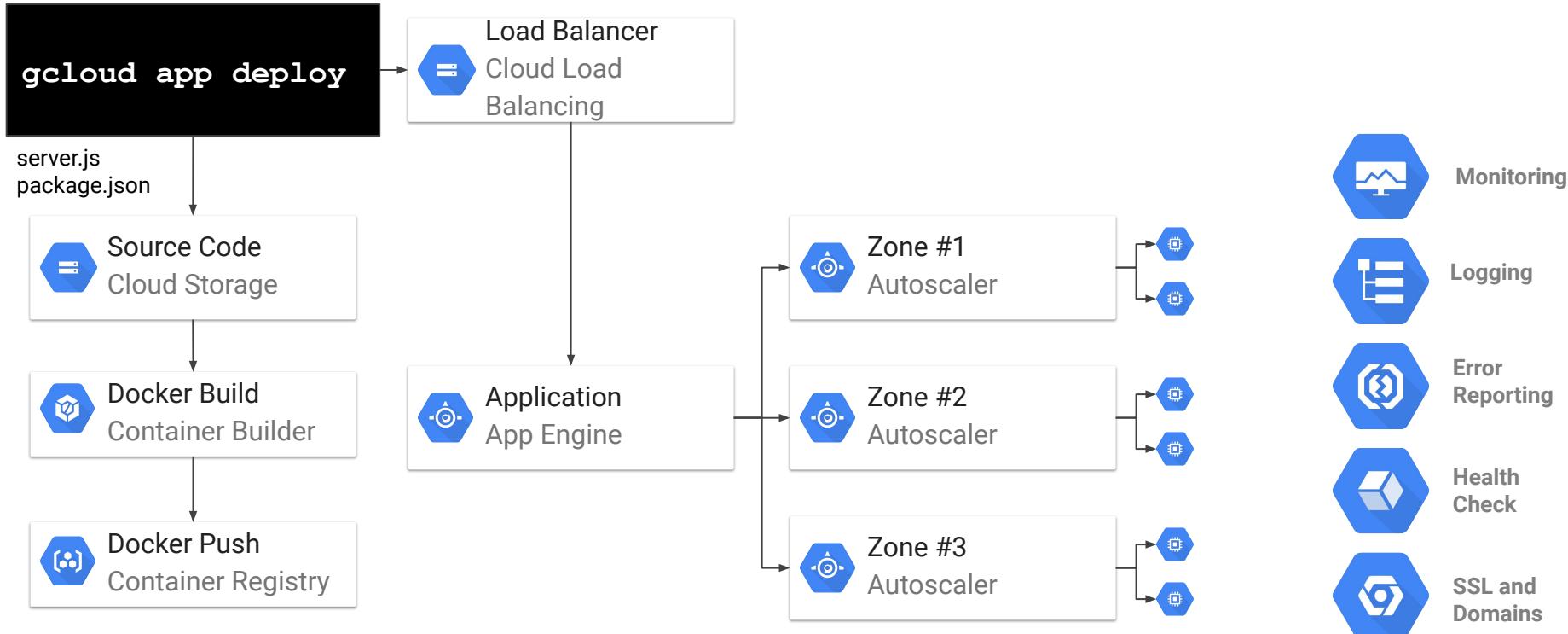


Custom runtimes

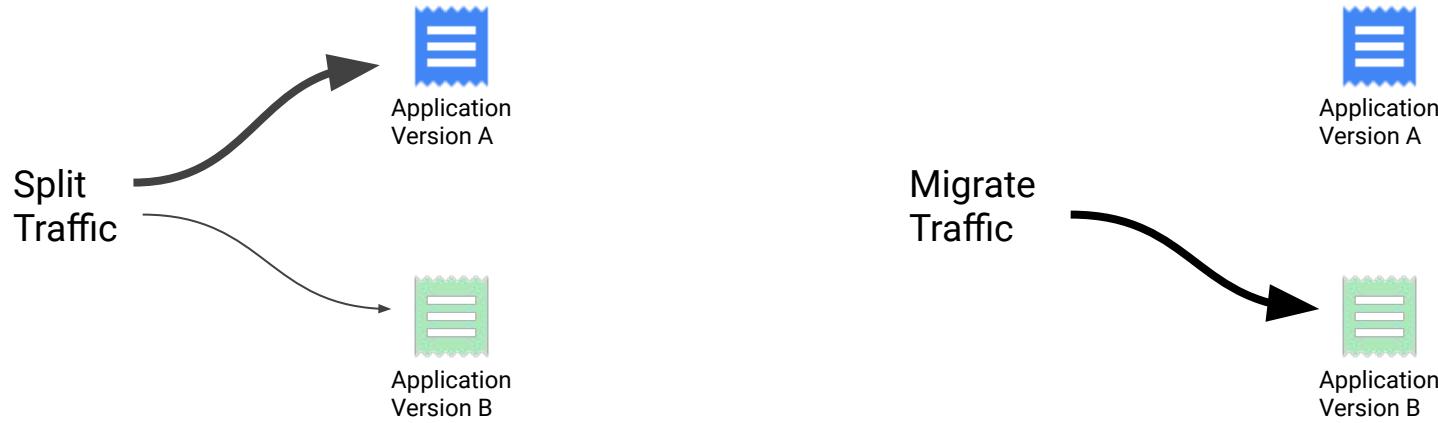
Develop and deploy microservices



Go from code to production with a single command



Deploy safely with zero downtime



Use App Engine flexible environment for highly scalable web-focused applications

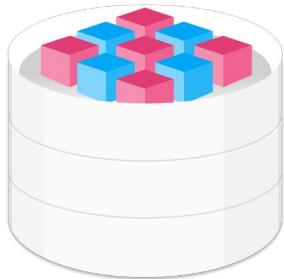
HTTP/S

- ✓ Applications that are based on HTTP/s request-responses
- ✓ Applications that deploy public endpoints
- ✓ Continuous integration and delivery (CI/CD) pipelines with Jenkins or Spinnaker

Consider other compute environments

~~HTTP/S~~

Protocols Other
Than HTTP/S



Persistent Disks



Spiky or Very Low Traffic

App Engine standard environment is an option for applications with spiky or very low traffic

App Engine Standard Environment

App Engine Standard APIs



App Engine flexible environment

Google Cloud Client Libraries

Developing Applications with GCP

Cloud Run

What is Serverless

Operational Model



No Infra Management



Managed Security



Pay only for usage

Programming Model



Service-based



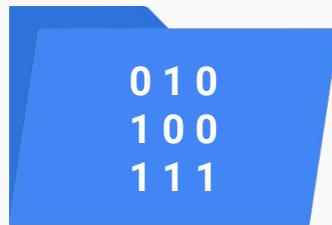
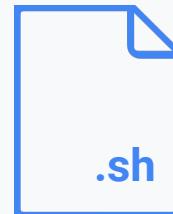
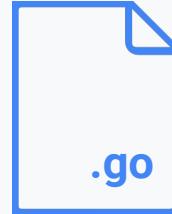
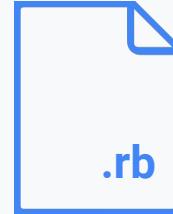
Event-driven



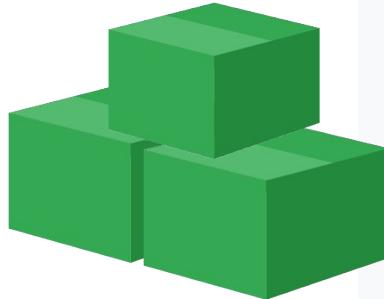
Open

Containers

- Any Language
- Any Library
- Any Binary
- Ecosystem of base images



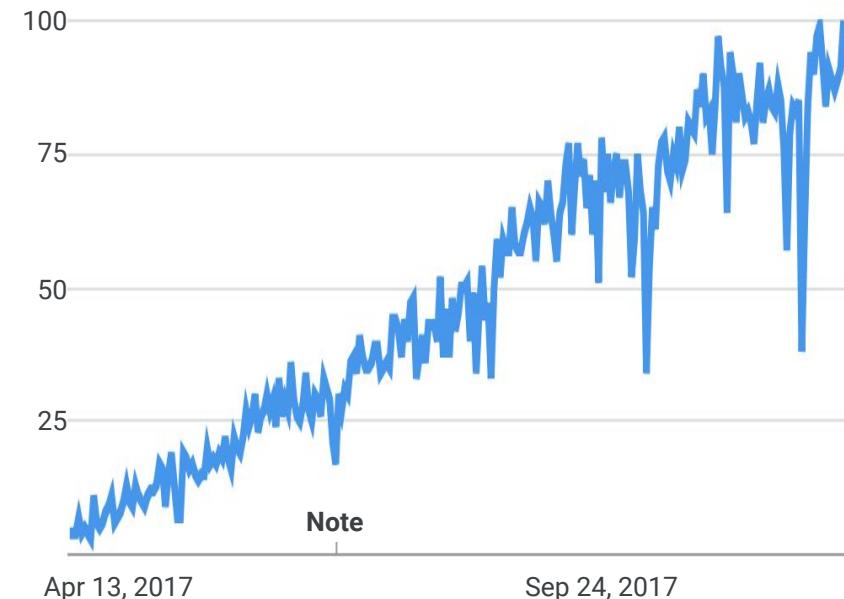
Containers: an industry standard



Interest over time

Google Trends

● docker container



Introducing Cloud Run

Bringing serverless to containers



Cloud Run



**Container to
production in
seconds**



**Natively
Serverless**



**One experience,
where you want it**

Serverless containers, where you want them



Cloud Run (fully managed)

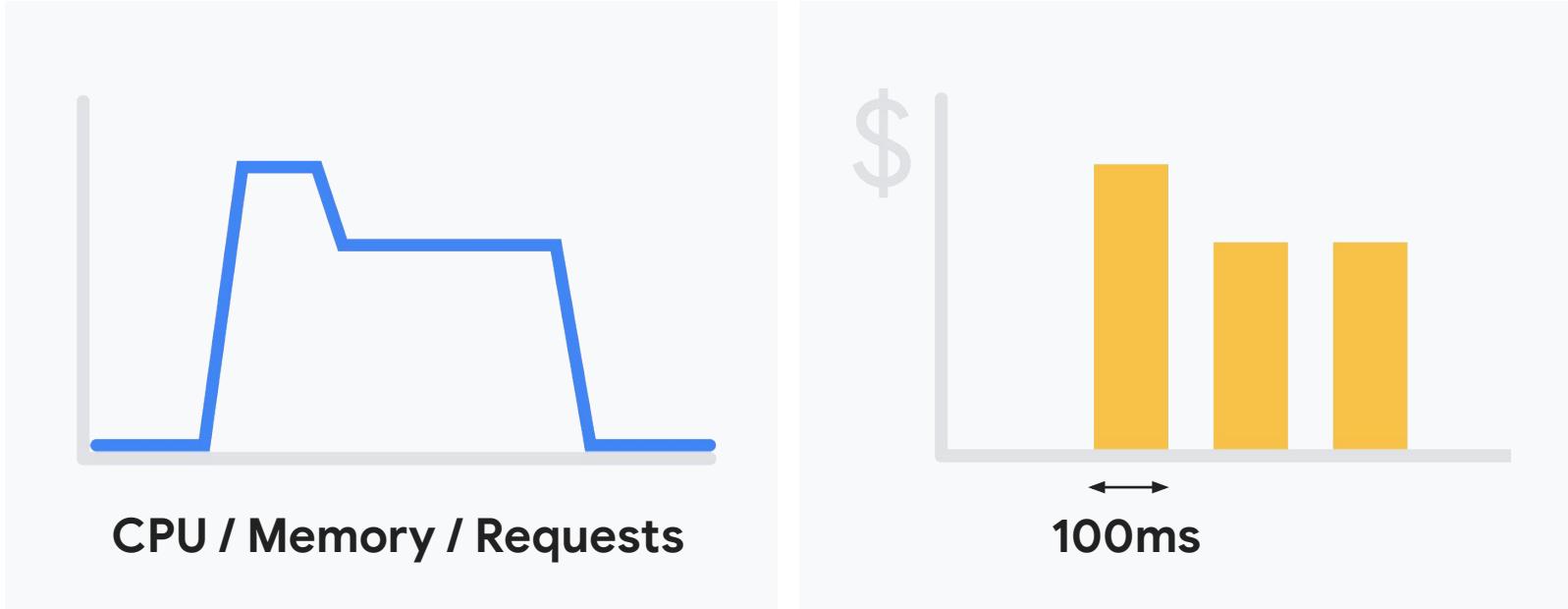
- Fully serverless
- No cluster to manage
- Pay for what you use



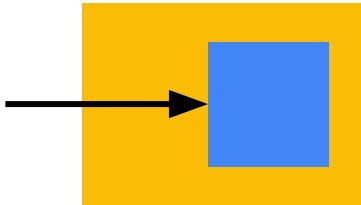
Cloud Run for Anthos

- Serverless developer experience
- Runs in your GKE cluster or on premise

Cloud Run: Pay-per-use

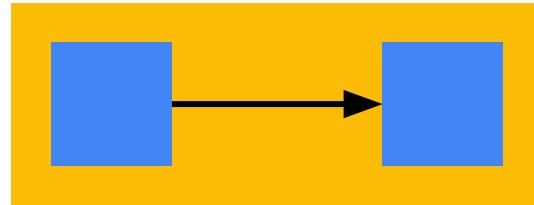


Use cases



Public

- Website
- API endpoint
- Mobile backend
- Webhook



Private

- Microservices
- Asynchronous tasks

Serverless compute options

Serverless
functions



Cloud
Functions

Source-based
HTTP and Events

Serverless
applications



App
Engine

Source-based
HTTP and async

Serverless
HTTP workloads



Cloud
Run

Containers
HTTP and async

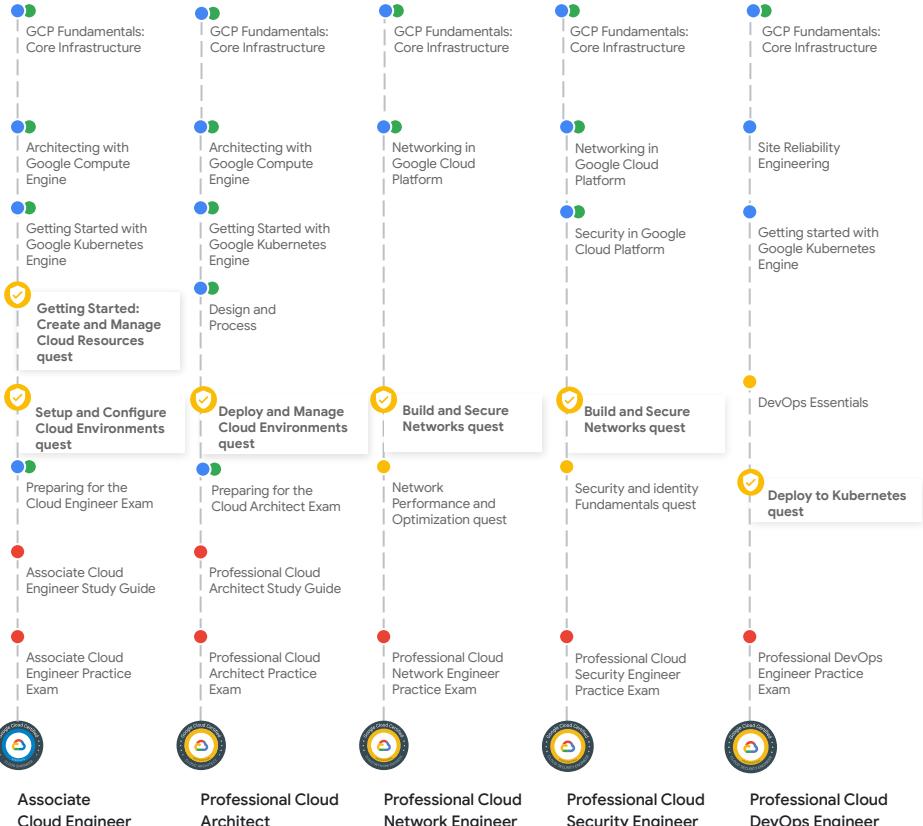
What's next?

Journey to certification

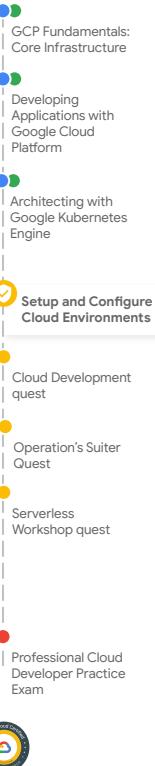
Whether you're new to the cloud or building on an existing skill set, find curriculum tailored to your role or interest. See more training at cloud.google.com/training

- On-demand Training
- Virtual and in-person Classroom Training
- Self-paced Labs
- Resource
- Complete & earn a skill badge

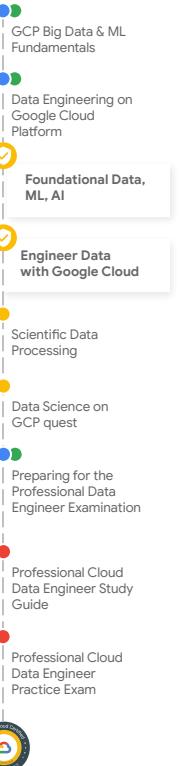
Infrastructure Modernization



Application Modernization



Smart Analytics and Data Management



Productivity and Collaboration





Thank you



Repository Slides

Google Cloud

Cloud OnBoard Icons

