# Google offers customer-friendly pricing

**Billing in sub-hour increments**

For compute, data processing and other services

**Discounts for sustained use**

Automatically applied to virtual machine use over 25% of a month

**Discounts for committed use**

Pay less for steady, long-term workloads

**Discounts for preemptible use**

Pay less for interruptible workloads

**Custom VM instance types**

Pay only for the resources you need for your application

# Open APIs and open source mean customers can leave

**Open APIs; compatibility with open-source services**

Cloud Bigtable

Cloud Dataproc

**Open source for a rich ecosystem**


Kubernetes

Forseti Security

**Multi-vendor-friendly technologies**

Google Stackdriver

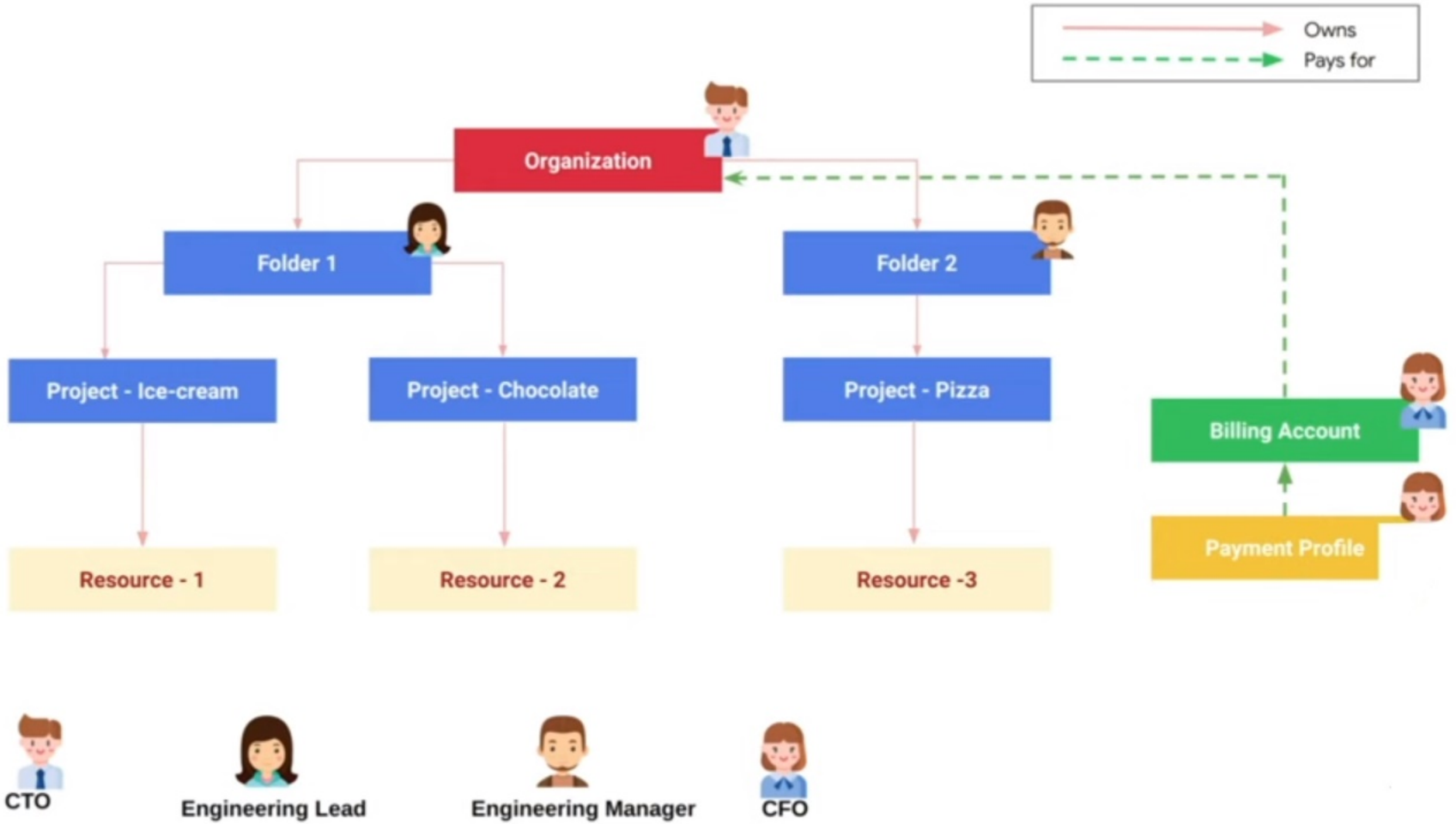
Kubernetes Engine

# Google Cloud Platform offers services for getting value from data

# Security is designed into Google's technical infrastructure

| Layer | Notable security measures (among others) |
| --- | --- |
| Operational security | Intrusion detection systems; techniques to reduce insider risk; employee U2F use; software development practices |
| Internet communication | Google Front End; designed-in Denial of Service protection |
| Storage services | Encryption at rest |
| User identity | Central identity service with support for U2F |
| Service deployment | Encryption of inter-service communication |
| Hardware infrastructure | Hardware design and provenance; secure boot stack; premises security |

# Resource hierarchy



# Set up a cloud solution environment

1. Resource hierarchy
2. Identity and Access Management (IAM)
3. Application Programming Interfaces (APIs)
4. Stackdriver workspace
5. Billing
6. Command-line interface (CLI) knowledge

# Compute options

Google Compute Engine (GCE)

Google Kubernetes Engine (GKE)

Google App Engine (GAE)

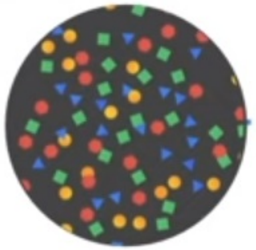
Google Cloud Functions

Highly customizable / Highly managed

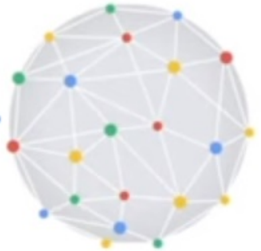
# Plan and configure a cloud solution

Plan and estimate Google Cloud products using the Pricing Calculator

Plan and configure compute resources



Plan and configure data storage options



Plan and configure network resources

# Compute options

Google Compute Engine (GCE)

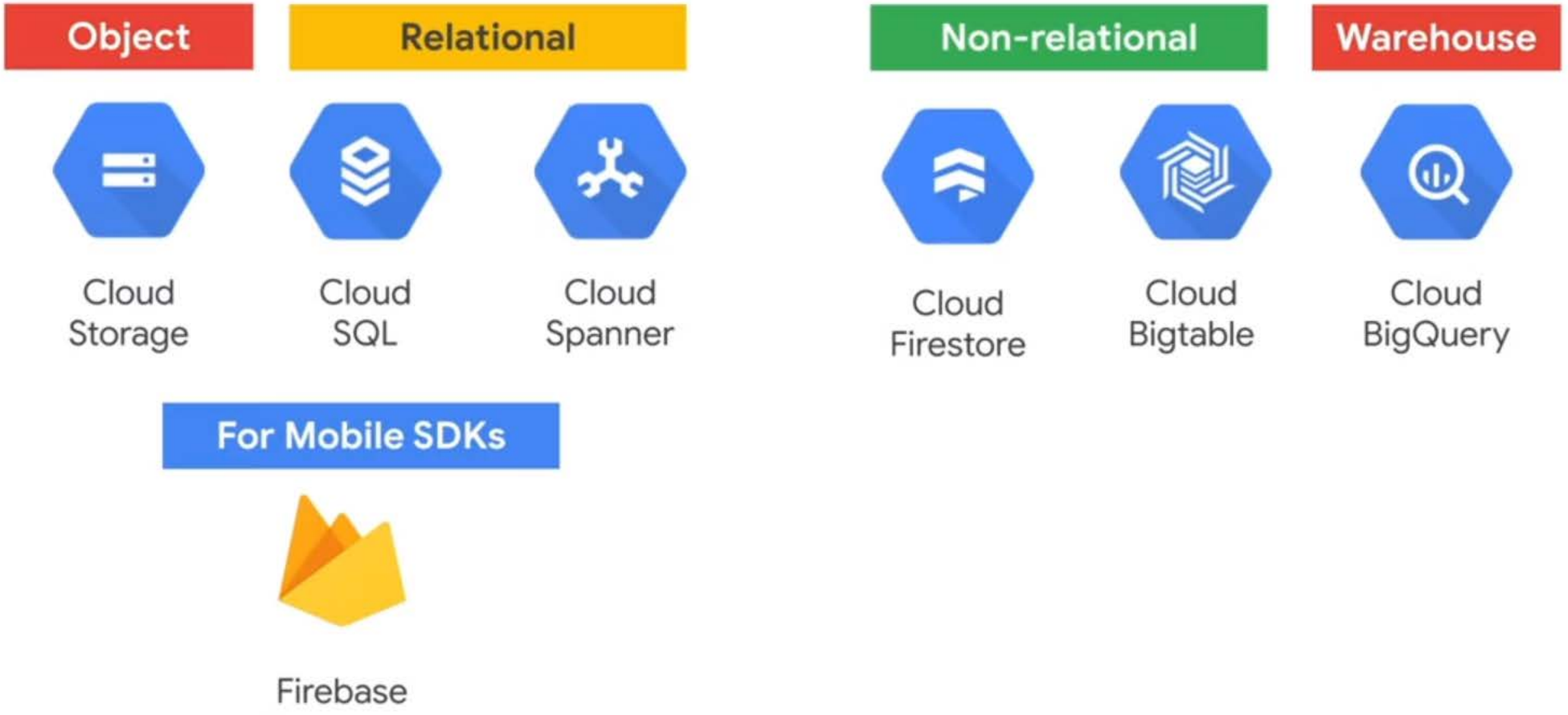
Google Kubernetes Engine (GKE)

Google App Engine (GAE)

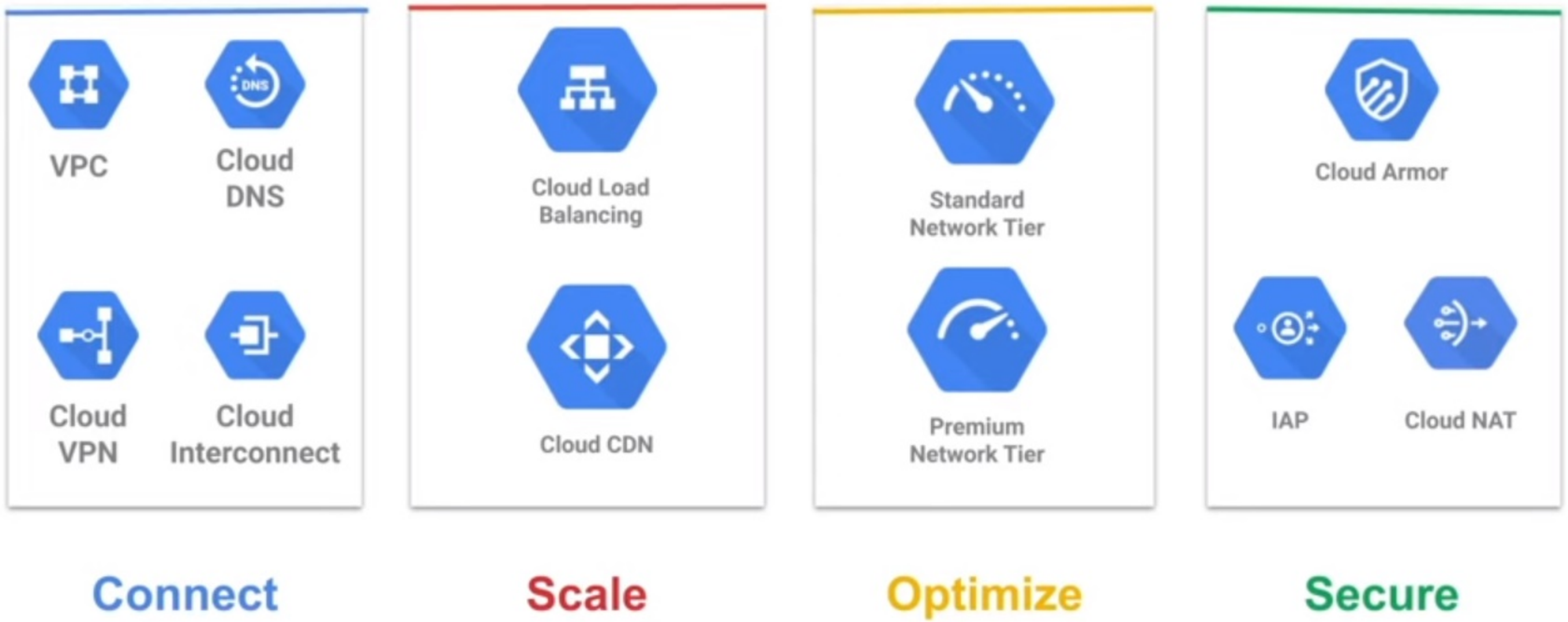
Google Cloud Functions

Highly customizable / Highly managed

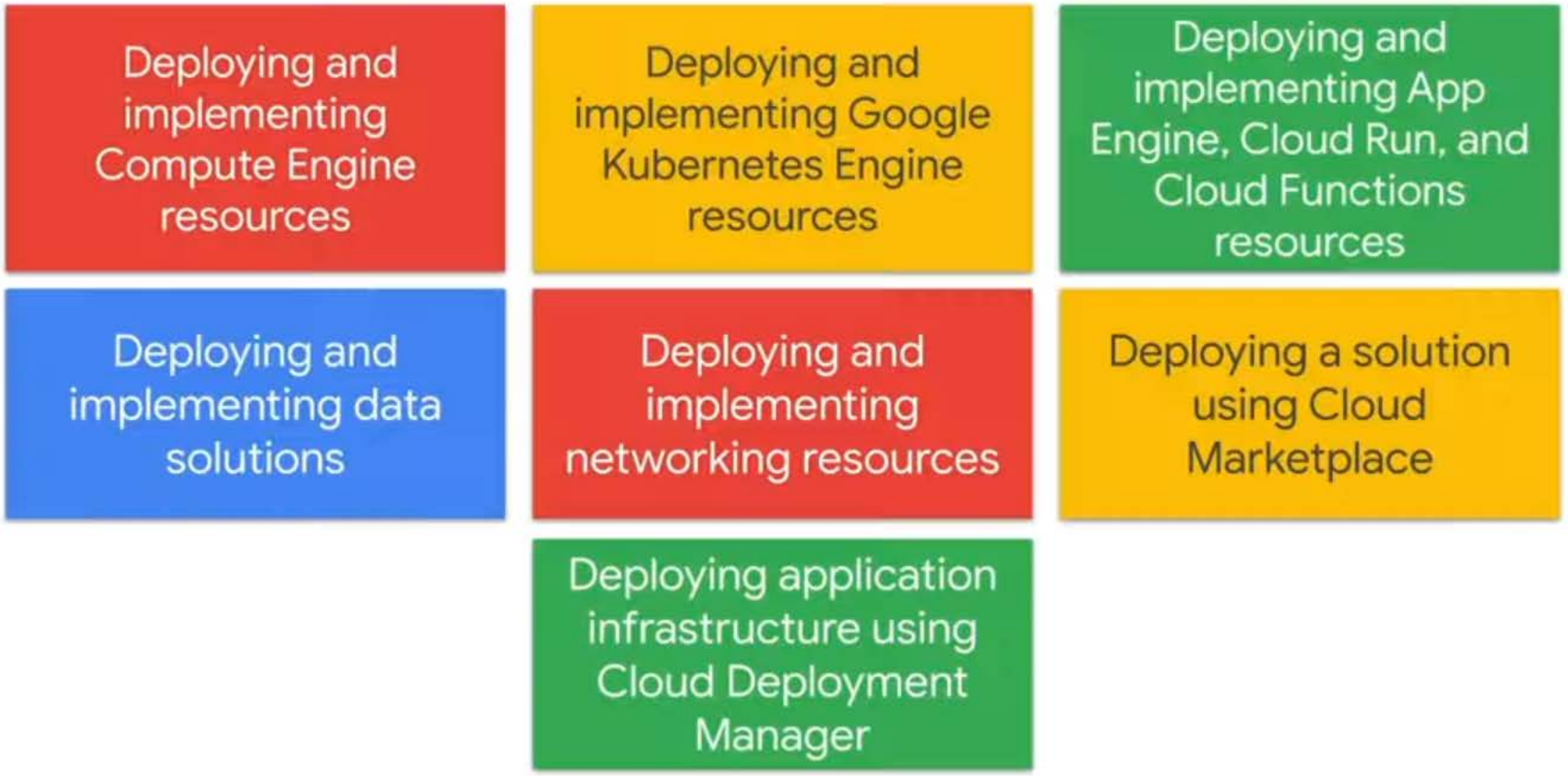
# GCP Cloud Storage Options



# Networking Product in GCP



# Deploying and implementing a cloud solution



# Ensuring successful operation of a cloud solution

- Managing Compute Engine resources
- Managing Google Kubernetes Engine resources
- Managing App Engine and Cloud Run resources
- Managing storage and database solutions
- Managing networking resources
- Monitoring and logging

# Configure access and security

## Manage identity and access management (IAM)

- Viewing IAM role assignments
- Assigning IAM roles to accounts or Google Groups
- Defining custom IAM roles

## Manage service accounts

- Managing service accounts with limited privileges
- Assigning a service account to VM instances
- Granting access to a service account in another project

## View audit logs for project and managed services

# Job Role Description

A CLOUD GURU

> An Associate Cloud Engineer deploys applications, monitors operations of multiple projects, and maintains enterprise solutions to ensure that they meet target performance metrics. This individual has experience working with public clouds and on-premises solutions. They are able to use Google Cloud Console and the command-line interface to perform common platform-based tasks to maintain one or more deployed solutions that leverage Google-managed or self-managed services on Google Cloud.

# Exam Sections (aka “Domains”)

A CLOUD GURU

- Section 1: Setting up a cloud solution environment
- Section 2: Planning and configuring a cloud solution
- Section 3: Deploying and implementing a cloud solution
- Section 4: Ensuring successful operation of a cloud solution
- Section 5: Configuring access and security

into sub-points and then further offers example tasks

Exam Guide (Blueprint)

# Job Role Highlights

A CLOUD GURU

- Deploys applications
- Monitors operations of multiple projects
- Maintains enterprise solutions to ensure they meet target performance metrics
- Experience working with public clouds and on-premises solutions
- Able to use Google Cloud Console and the command-line interface
- Performs common platform-based tasks
- Maintains one or more deployed solutions
- Leverages Google-managed or self-managed services on Google Cloud

on the Google Cloud.

# §1: Setting up a cloud solution environment

A CLOUD GURU

- 1.1 Setting up cloud projects and accounts
- 1.2 Managing billing configuration
- 1.3 Installing and configuring the command line interface (CLI)

Exam Guide (Blueprint)

# §2: Planning and configuring a cloud solution

A CLOUD GURU

- 2.1 Planning and estimating GCP product use using the Pricing Calculator
- 2.2 Planning and configuring compute resources
- 2.3 Planning and configuring data storage options
- 2.4 Planning and configuring network resources

These are the three critical components to data flows

Exam Guide (Blueprint)

# §3: Deploying and implementing a cloud solution

A CLOUD GURU

- 3.1 Deploying and implementing Compute Engine resources
- 3.2 Deploying and implementing Kubernetes Engine resources
- 3.3 Deploying and implementing App Engine and Cloud Functions resources
- 3.4 Deploying and implementing data solutions
- 3.5 Deploying and implementing networking resources
- 3.6 Deploying a Solution using Cloud Launcher
- 3.7 Deploying an Application using Deployment Manager

like Cloud Launcher, Deployment Manager,

Exam Guide (Blueprint)

# §5: Configuring access and security

A CLOUD GURU

The final section in the exam guide is

# §4: Ensuring successful operation of a cloud solution

A CLOUD GURU

- 4.1 Managing Compute Engine resources
- 4.2 Managing Kubernetes Engine resources
- 4.3 Managing App Engine resources
- 4.4 Managing data solutions
- 4.5 Managing networking resources
- 4.6 Monitoring and logging

Exam Guide (Blueprint)

# §5: Configuring access and security

A CLOUD GURU

- 5.1 Managing Identity and Access Management (IAM)
- 5.2 Managing service accounts
- 5.3 Viewing audit logs for project and managed services

we also have the responsibility to make sure

# Google VPC offers a suite of load-balancing options

| Global HTTP(S) | Global SSL Proxy | Global TCP Proxy | Regional | Regional internal |
|---|---|---|---|---|
| Layer 7 load balancing based on load | Layer 4 load balancing of non-HTTPS SSL traffic based on load | Layer 4 load balancing of non-SSL TCP traffic | Load balancing of any traffic (TCP, UDP) | Load balancing of traffic inside a VPC |
| Can route different URLs to different back ends | Supported on specific port numbers | Supported on specific port numbers | Supported on any port number | Use for the internal tiers of multi-tier applications |

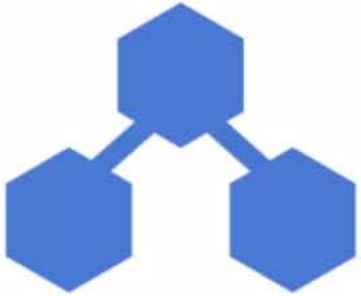
# Google Cloud Platform offers many interconnect options

## VPN

Secure multi-Gbps connection over VPN tunnels

## Direct Peering

Private connection between you and Google for your hybrid cloud workloads



## Carrier Peering

Connection through the largest partner network of service providers

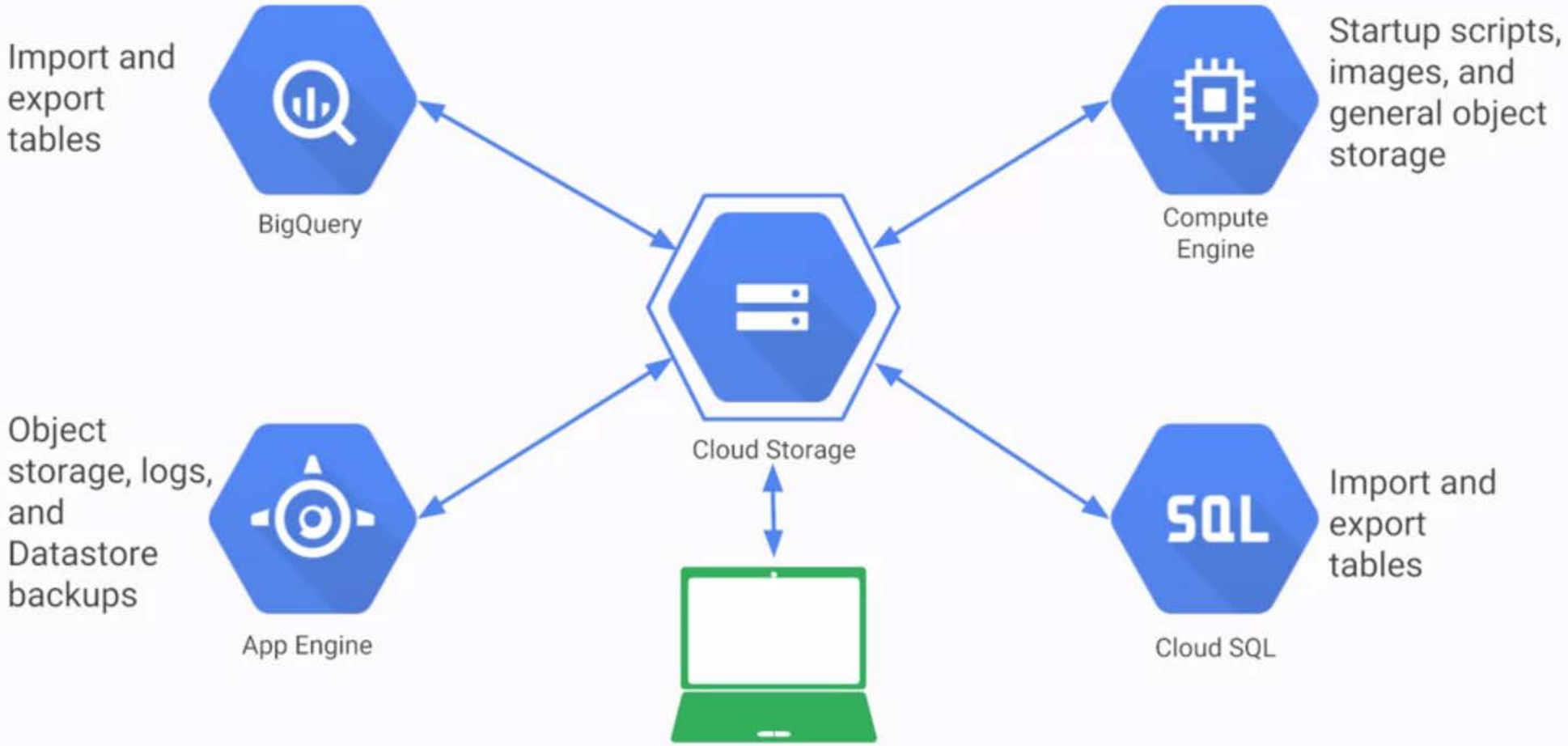
## Dedicated Interconnect

Connect N X 10G transport circuits for private cloud traffic to Google Cloud at Google POPs

# Choosing among Cloud Storage classes

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

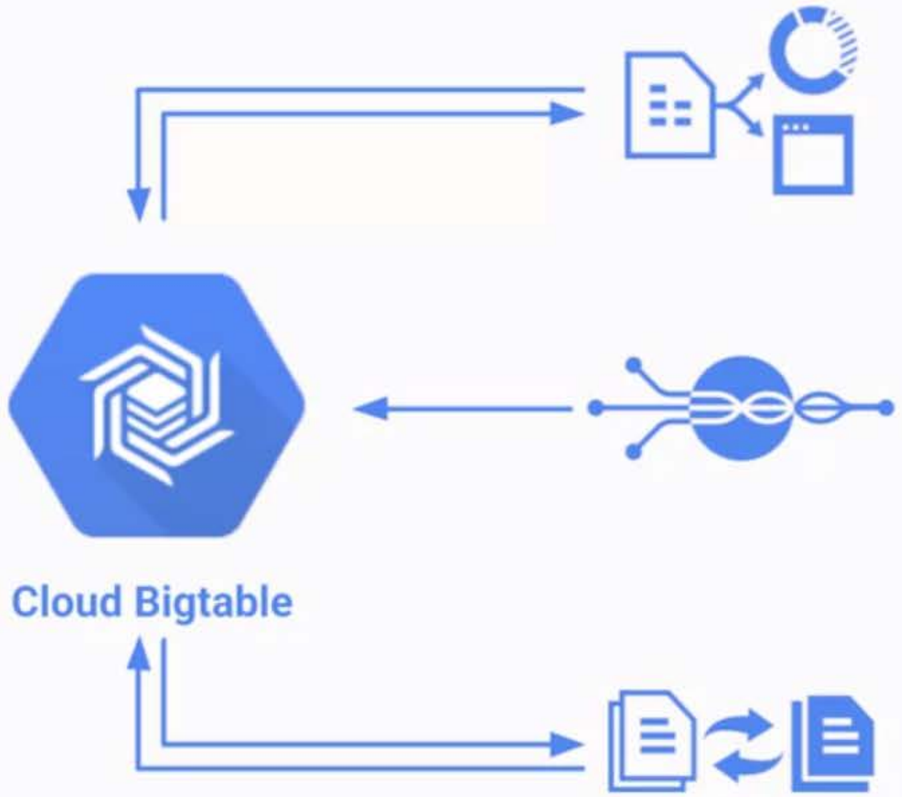
# Cloud Storage works with other GCP services



# Your Cloud Storage files are organized into buckets

| Bucket attributes | Bucket contents |
| --- | --- |
| Globally unique name | Files (in a flat namespace) |
| Storage class | |
| Location (region or multi-region) | |
| IAM policies or Access Control Lists | Access Control Lists |
| Object versioning setting | |
| Object lifecycle management rules | |

# Bigtable Access Patterns




### Application API

Data can be read from and written to Cloud Bigtable through a data service layer like Managed VMs, the HBase REST Server, or a Java Server using the HBase client. Typically this will be to serve data to applications, dashboards, and data services.

### Streaming

Data can be streamed in (written event by event) through a variety of popular stream processing frameworks like Cloud Dataflow Streaming, Spark Streaming, and Storm.

### Batch Processing

Data can be read from and written to Cloud Bigtable through batch processes like Hadoop MapReduce, Dataflow, or Spark. Often, summarized or newly calculated data is written back to Cloud Bigtable or to a downstream database.

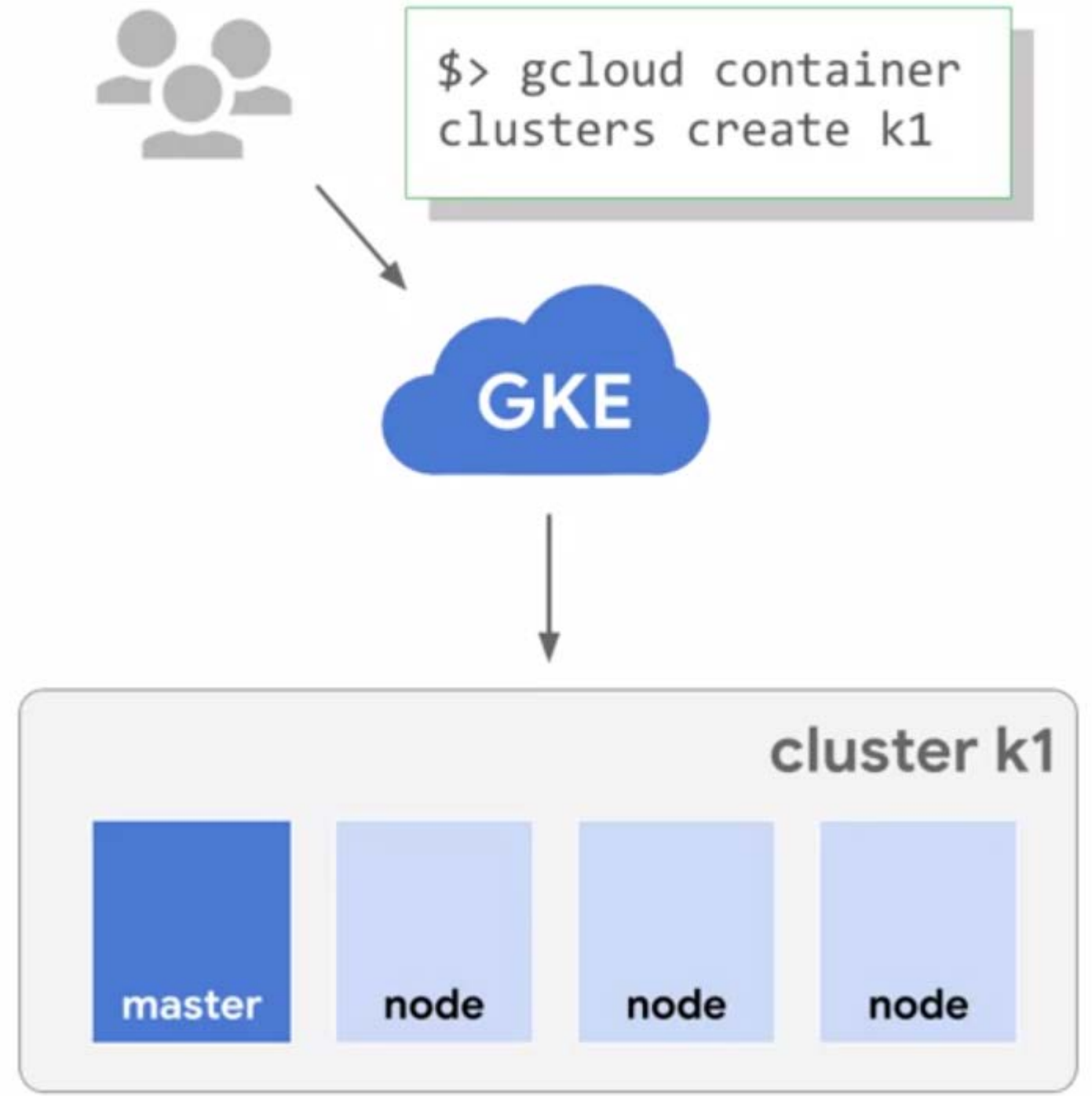
# Comparing storage options: technical details

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

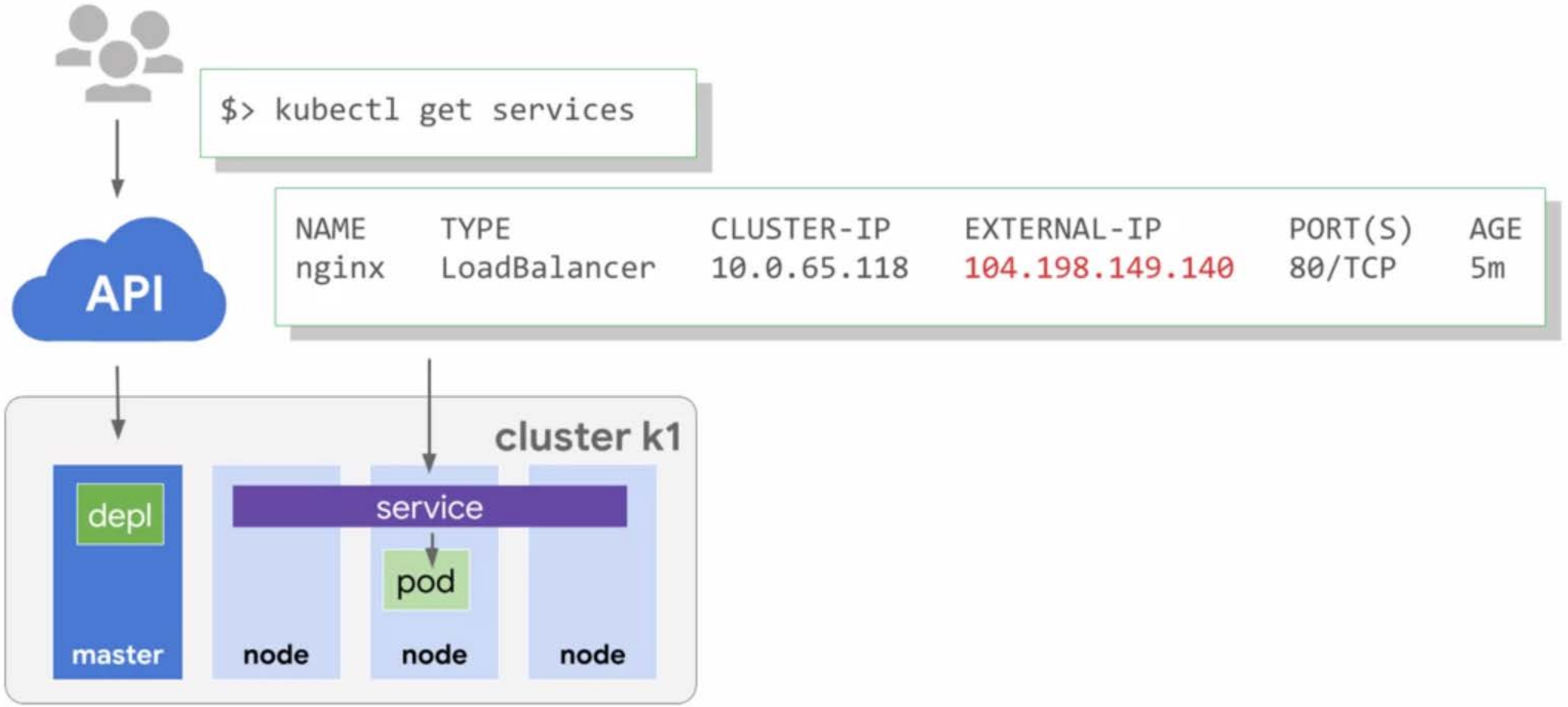
# Comparing storage options: technical details

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

# Kubernetes Engine



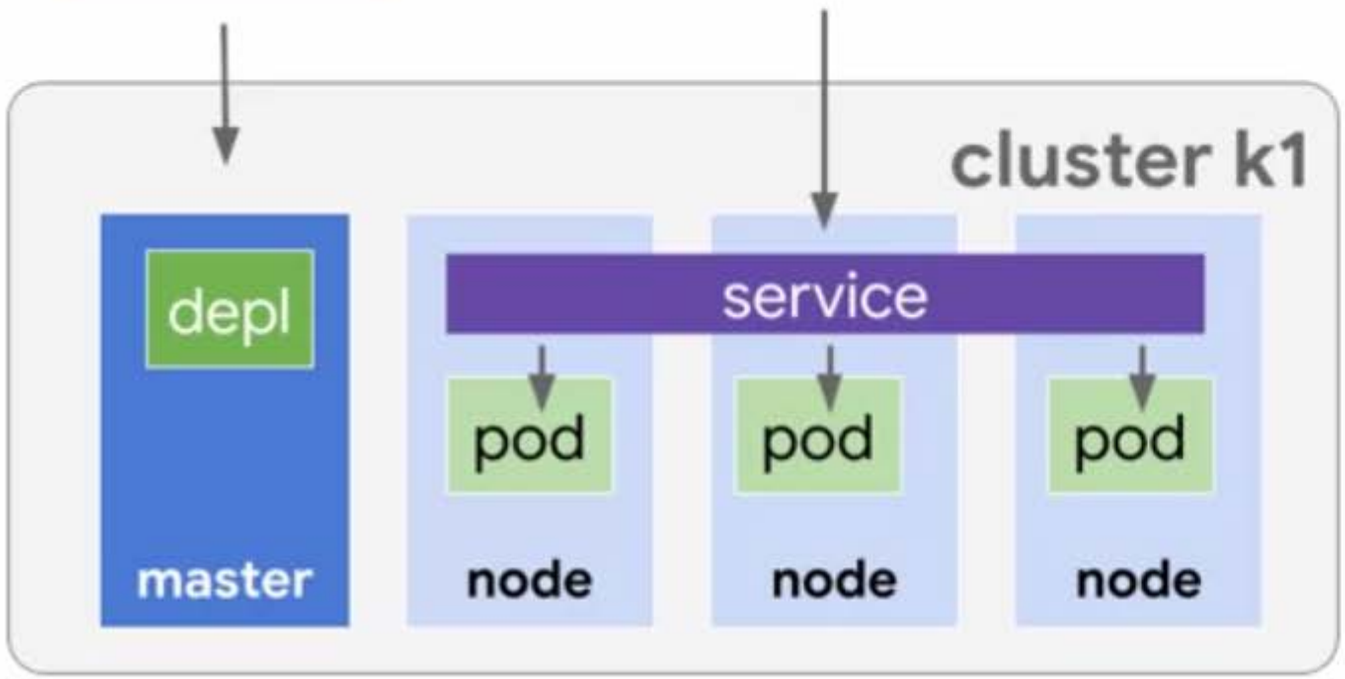
# Kubernetes Engine



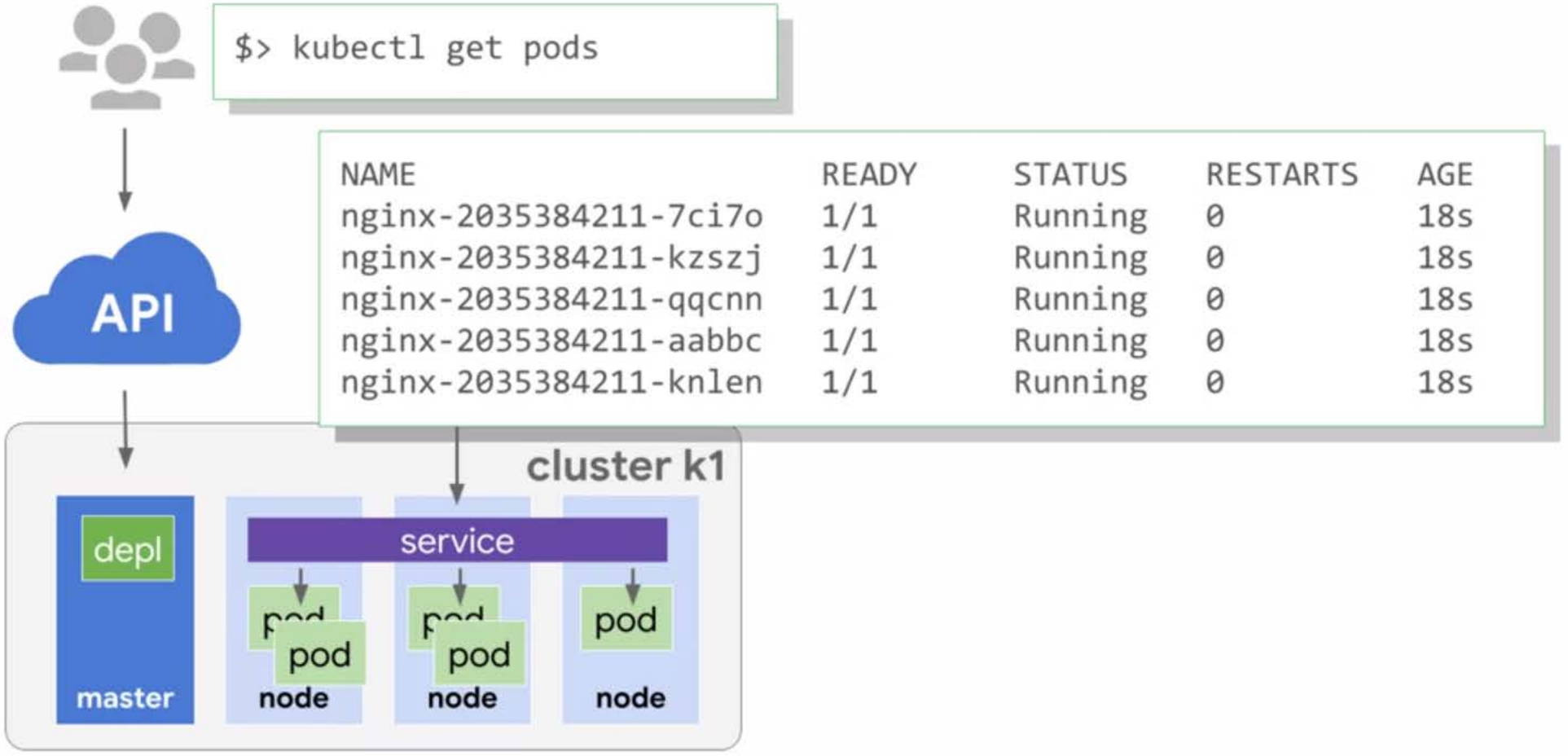
# Kubernetes

```
$> kubectl get replicasets
```

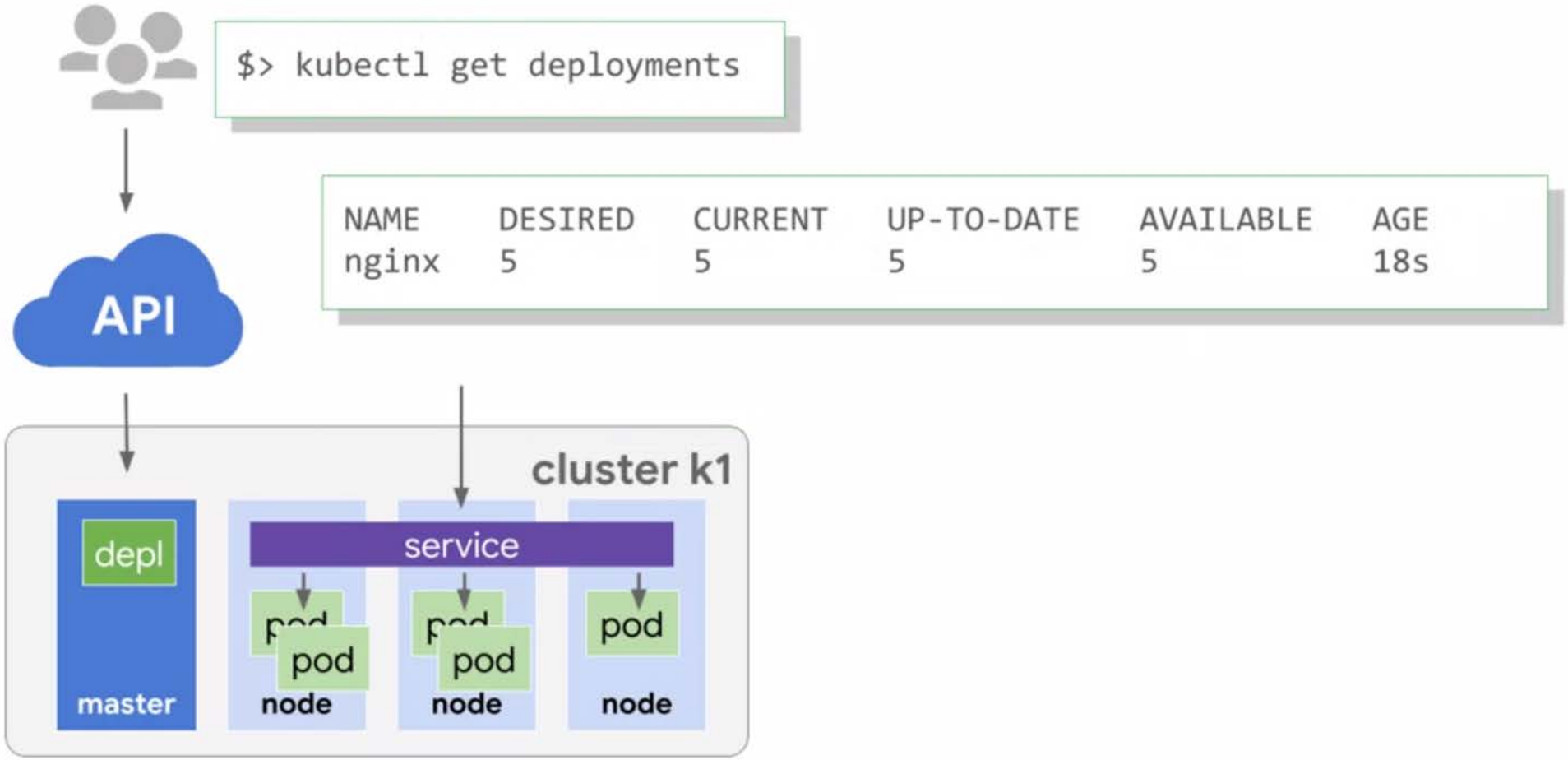
```
NAME                  DESIRED   CURRENT   READY   AGE
nginx-2035384211      5         5         5       18s
```

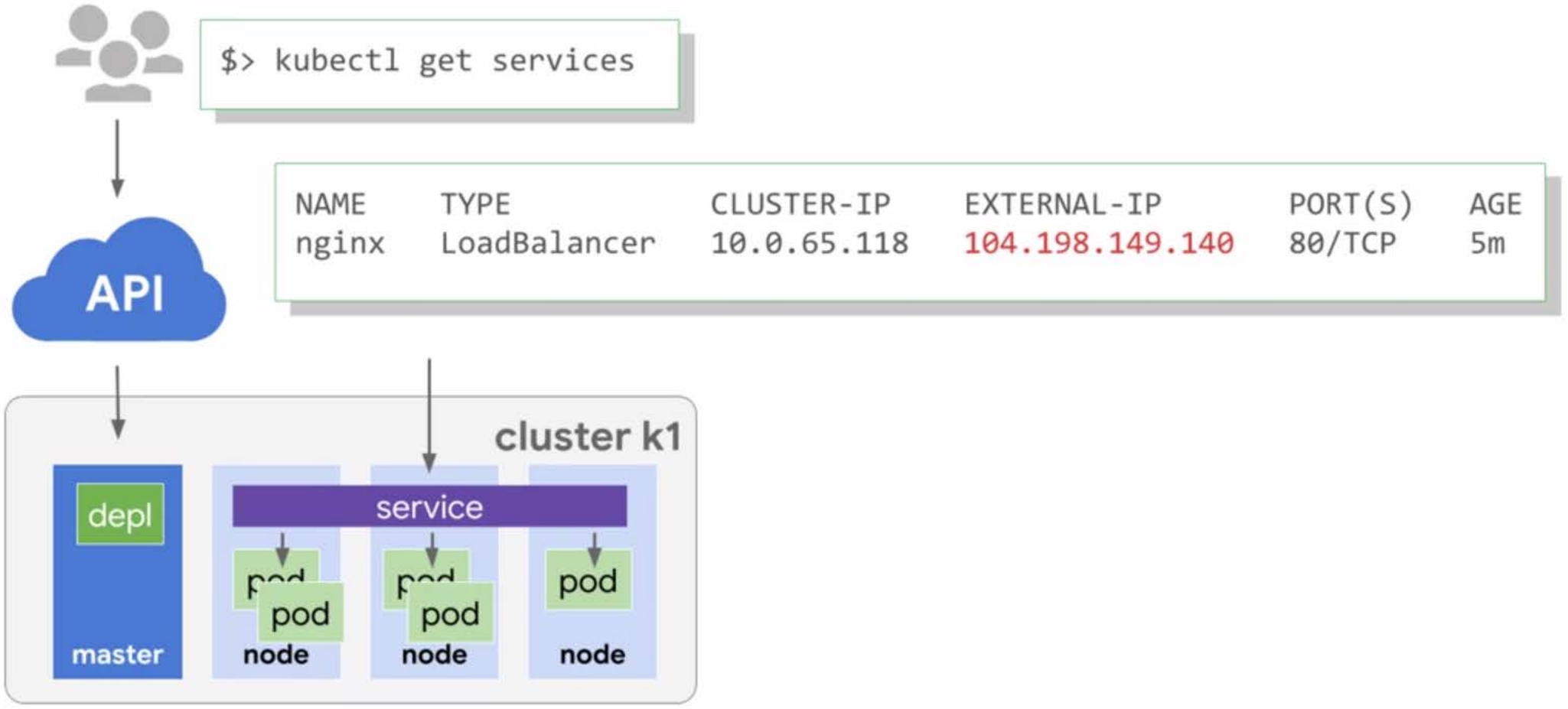
# Kubernetes


$> kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-2035384211-7ci7o 1/1 Running 0 18s
nginx-2035384211-kzszj 1/1 Running 0 18s
nginx-2035384211-qqcnn 1/1 Running 0 18s
nginx-2035384211-aabbc 1/1 Running 0 18s
nginx-2035384211-knlen 1/1 Running 0 18s
API
cluster k1
depl
master
service
pod
pod
pod
pod
pod
node
node
node

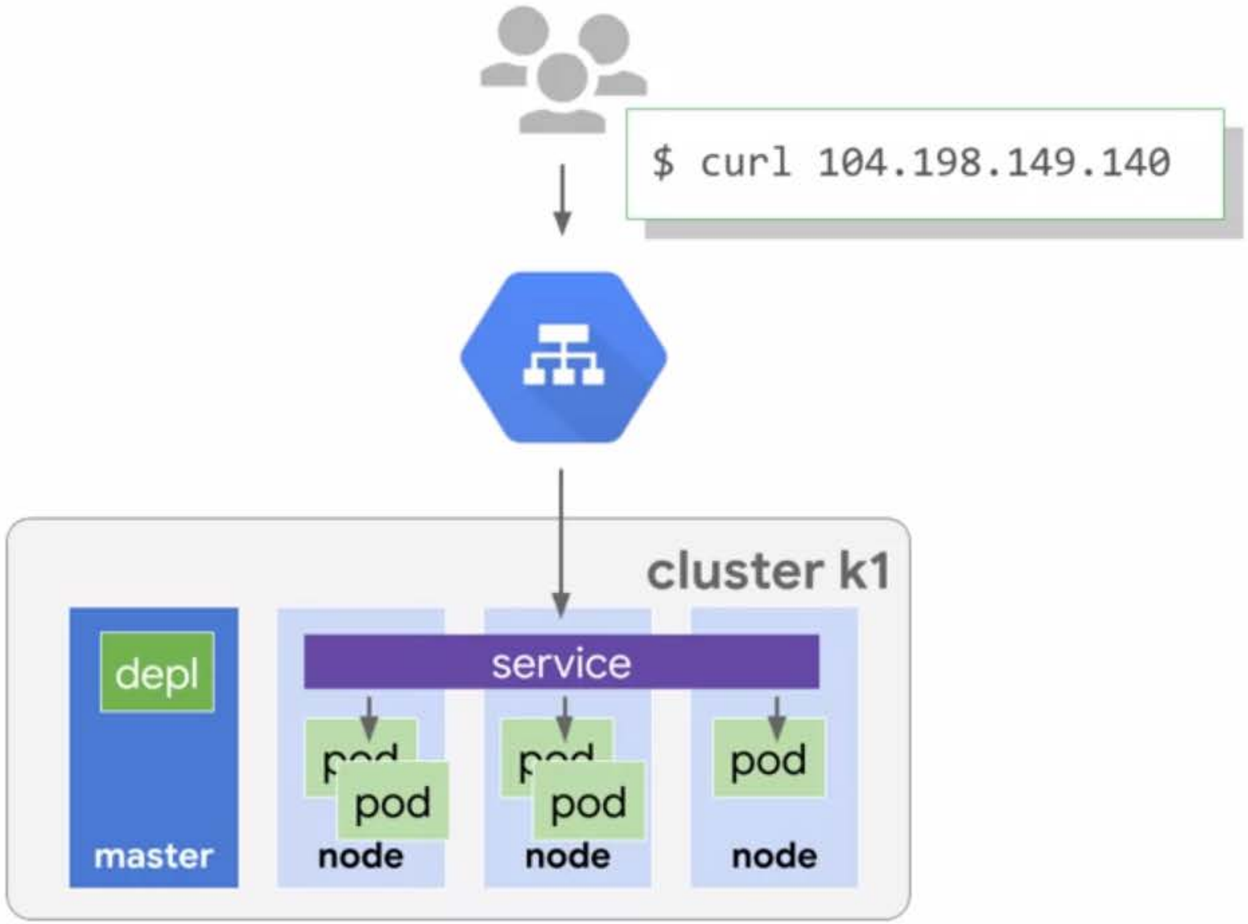
# Kubernetes


$> kubectl get deployments
NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
nginx 5 5 5 5 18s
API
cluster k1
depl
master
service
pod
pod
pod
pod
pod
node
node
node

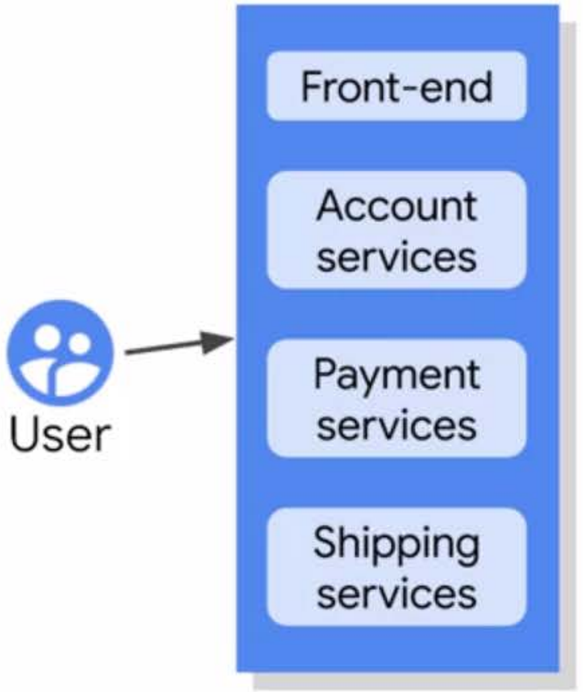
# Kubernetes


$> kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
nginx LoadBalancer 10.0.65.118 104.198.149.140 80/TCP 5m
API
cluster k1
depl
master
service
pod
pod
pod
pod
pod
node
node
node

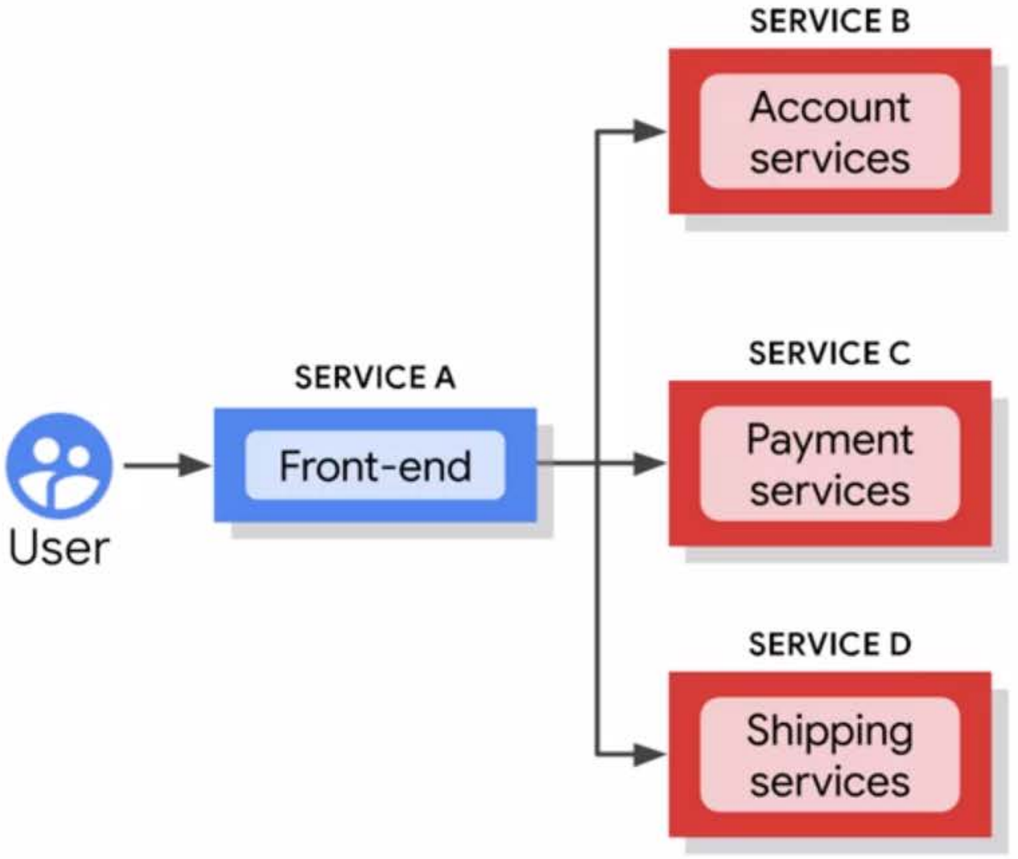
# Kubernetes Engine



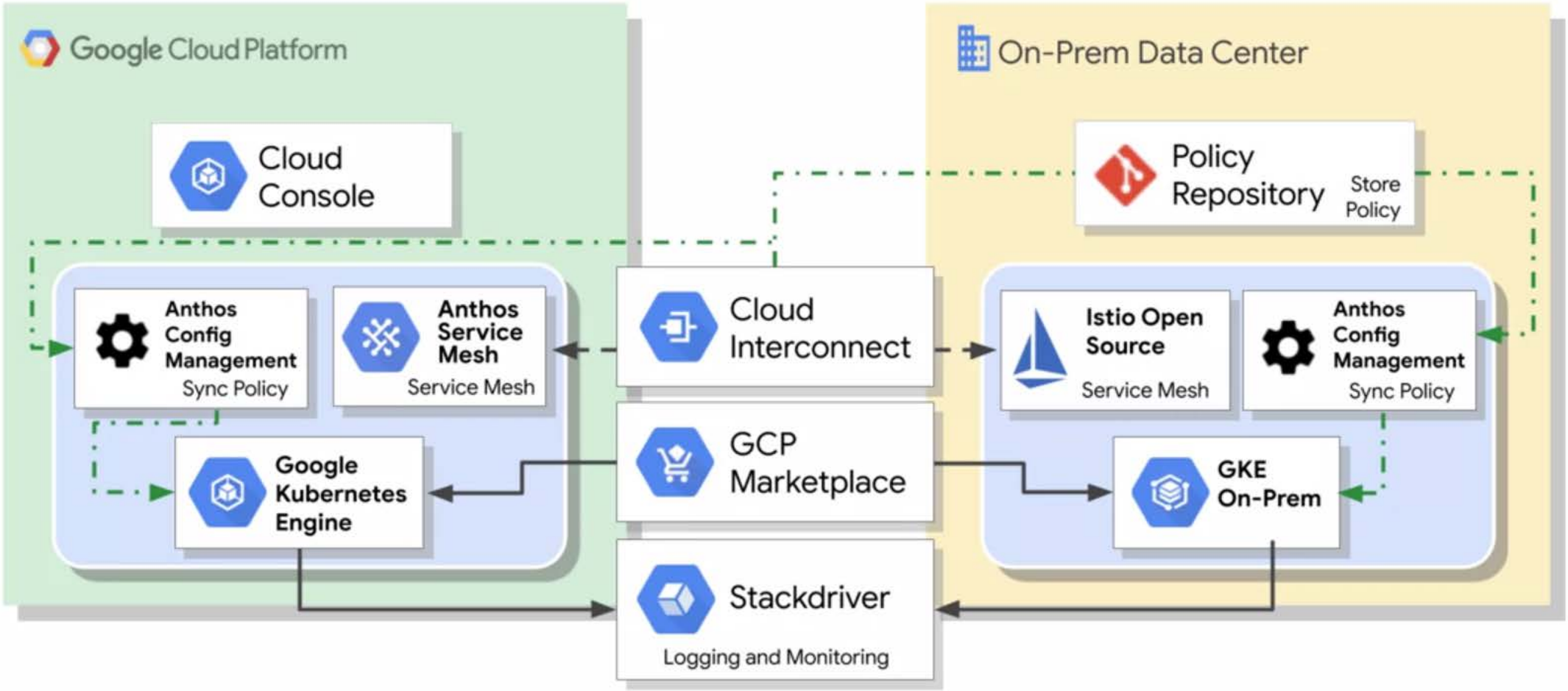
## Monolithic Application




## Containerized Microservices Application



# Configuration Manager is the single source of truth



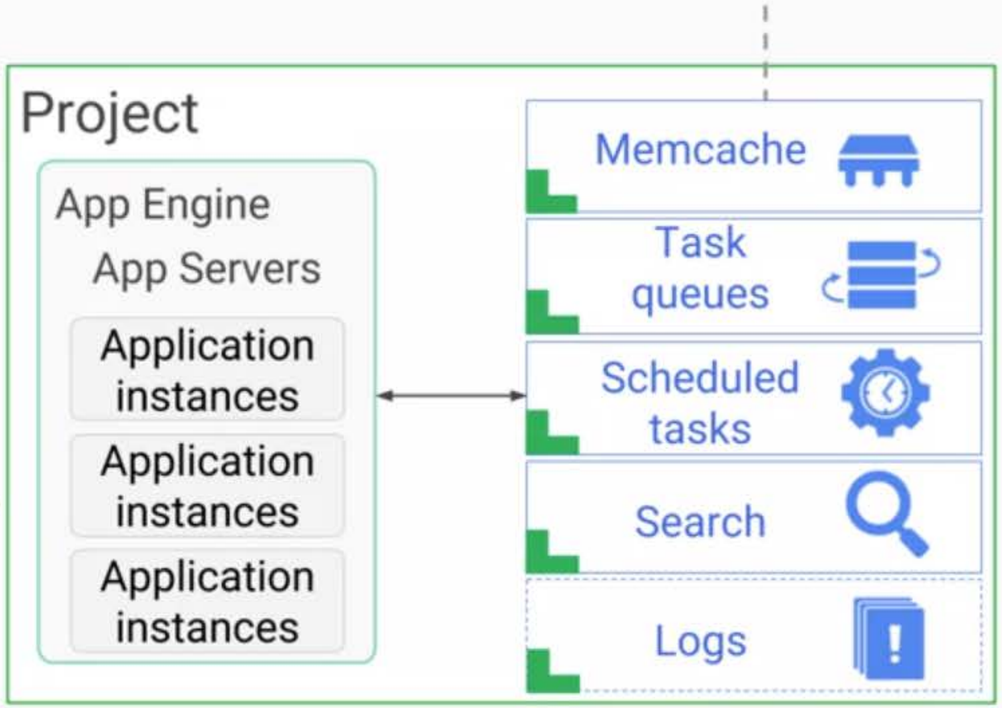
# Example App Engine standard workflow: Web applications

1 Develop & test the web application locally

2 Use the SDK to deploy to App Engine

3 App Engine automatically scales & reliably serves your web application

App Engine can access a variety of services using dedicated APIs



# Comparing the App Engine environments

| | Standard Environment | Flexible Environment |
|---|---|---|
| *Instance startup* | Milliseconds | Minutes |
| *SSH access* | No | Yes (although not by default) |
| *Write to local disk* | No | Yes (but writes are ephemeral) |
| *Support for 3rd-party binaries* | No | Yes |
| *Network access* | Via App Engine services | Yes |
| *Pricing model* | After free daily use, pay per instance class, with automatic shutdown | Pay for resource allocation per hour; no automatic shutdown |

# Deploying Apps: Kubernetes Engine vs App Engine

| | **Kubernetes Engine** | **App Engine Flexible** | **App Engine Standard** |
|---|---|---|---|
| *Language support* | Any | Any | Java, Python, Go, PHP |
| *Service model* | Hybrid | PaaS | PaaS |
| *Primary use case* | Container-based workloads | Web and mobile applications, container-based workloads | Web and mobile applications |

*Toward managed infrastructure*

*Toward dynamic infrastructure*

# Stackdriver offers capabilities in six areas

## Monitoring

Platform, system, and application metrics

Uptime/health checks

Dashboards and alerts

## Error Reporting

Error notifications

Error dashboard

## Logging

Platform, system, and application logs

Log search, view, filter, and export

Log-based metrics

## Debugger

Debug applications

## Trace

Latency reporting and sampling

Per-URL latency and statistics

## Profiler Beta

Continuous profiling of CPU and memory consumption

# Google Cloud's big data services are fully managed and scalable

**Cloud Dataproc**

Managed Hadoop MapReduce, Spark, Pig, and Hive service

**Cloud Dataflow**

Stream and batch processing; unified and simplified pipelines

**BigQuery**

Analytics database; stream data at 100,000 rows per second

**Cloud Pub/Sub**

Scalable and flexible enterprise messaging

**Cloud Datalab**

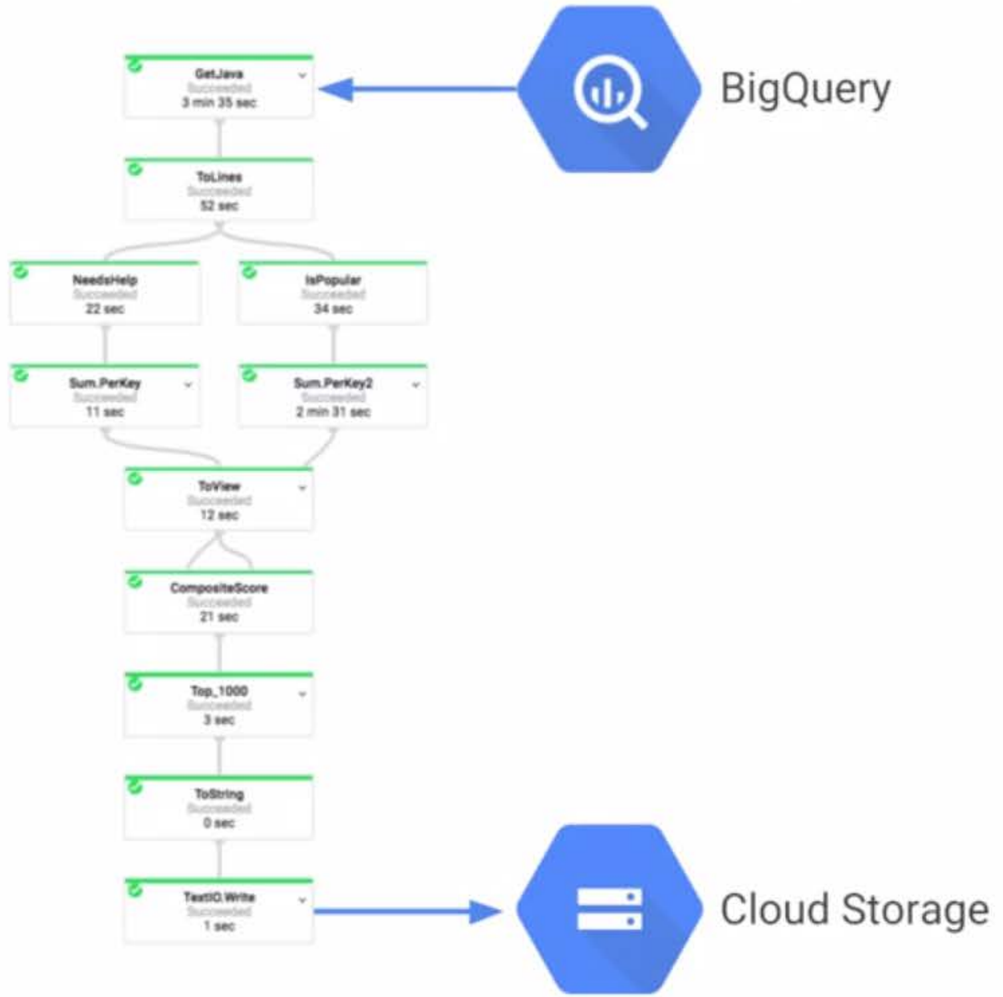
Interactive data exploration

# Dataflow pipelines flow data from a source through transforms

Source

Transforms

Sink



# Cloud Machine Learning Platform


**Open source tool to build and run neural network models**

- Wide platform support: CPU or GPU; mobile, server, or cloud

**Fully managed machine learning service**

- Familiar notebook-based developer experience
- Optimized for Google infrastructure; integrates with BigQuery and Cloud Storage

**Pre-trained machine learning models built by Google**

- Speech: Stream results in real time, detects 80 languages
- Vision: Identify objects, landmarks, text, and content
- Translate: Language translation including detection
- Natural language: Structure, meaning of text

# Why use the Cloud Machine Learning platform?

## For structured data

- Classification and regression
- Recommendation
- Anomaly detection

## For unstructured data

- Image and video analytics
- Text analytics

# Comparing compute options

| | Compute Engine | Kubernetes Engine | App Engine Flex | App Engine Standard | Cloud Functions[Beta] |
|---|---|---|---|---|---|
| Service model | IaaS | Hybrid | PaaS | PaaS | Serverless |
| Use cases | General computing workloads | Container-based workloads | Web and mobile applications; container-based workloads | Web and mobile applications | Ephemeral functions responding to events |

*Toward managed infrastructure*

*Toward dynamic infrastructure*

# Comparing load-balancing options

| Global HTTP(S) | Global SSL Proxy | Global TCP Proxy | Regional | Regional internal |
|---|---|---|---|---|
| Layer 7 load balancing based on load | Layer 4 load balancing of non-HTTPS SSL traffic based on load | Layer 4 load balancing of non-SSL TCP traffic | Load balancing of any traffic (TCP, UDP) | Load balancing of traffic inside a VPC |
| Can route different URLs to different back ends | Supported on specific port numbers | Supported on specific port numbers | Supported on any port number | Use for the internal tiers of multi-tier applications |

# Comparing interconnect options

## VPN

Secure multi-Gbps connection over VPN tunnels

## Direct Peering

Private connection between you and Google for your hybrid cloud workloads

## Carrier Peering

Connection through the largest partner network of service providers

## Dedicated Interconnect

Connect N X 10G transport circuits for private cloud traffic to Google Cloud at Google POPs

# Comparing storage options

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

# Choosing among Google Cloud Storage classes

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

A : ACP (Pre) - Mechanical [ANSYS Mechanical Enterprise]

File Edit View Units Tools Help | Solve | Show Errors | Worksheet

Show Vertices | Close Vertices | 0.21 (Auto Scale) | Wireframe | Show Mesh | Random | Preferences | Size | Location | Convert | Miscellaneous | Tolerances

Reset | Explode Factor: | Assembly Center | Edge Coloring | Thicken

Mesh | Update | Mesh | Mesh Control | Mesh Edit | Metric Graph | Probe

## Outline

Filter: Name

- Model (A4)
  - Geometry
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
  - Coordinate Systems
  - Connections
  - Mesh
    - Face Sizing 2

## Details of "Mesh"

| | |
|---|---|
| Element Order | Program Controlled |
| **Sizing** | |
| Size Function | Proximity and Curvature |
| Relevance Center | Fine |
| Span Angle Center | Coarse |
| Curvature Normal Angle | 18.0 ° |
| Num Cells Across Gap | Default (3) |
| Proximity Size Function Sources | Faces and Edges |
| Min Size | 1.0 mm |
| Proximity Min Size | 1.0 mm |
| Max Face Size | 3.0 mm |
| Growth Rate | Default |
| Automatic Mesh Based Defeaturing | Off |
| Minimum Edge Length | 16.0970 mm |
| **Quality** | |

ANSYS
R18.1

ANSYS Workbench Mesh Status

Meshing...

Highlight

**Part:** ACP-Pre
**Status:** Converting triangles to quads... (1/3) [70%]

Stop

300.00 600.00 (mm)
150.00 450.00

Y Z X

Geometry | Print Preview | Report Preview

## Messages

| | Text | Association | Timestamp |
|---|---|---|---|
| Warninc | One or more parts were found to be unmodified so smart updated. | | Saturday, November 28, 2020 12:11:33 PM |

1 Message | No Selection | Metric (mm, kg, N, s, mV, mA) Degrees rad/s Celsius

89% ENG 12:15 28-11-2020

A : ACP (Pre) - Mechanical [ANSYS Mechanical Enterprise]

File Edit View Units Tools Help | Solve | Show Errors | Worksheet

Show Vertices | Close Vertices | 0.21 (Auto Scale) | Wireframe | Show Mesh | Random | Preferences | Size | Location | Convert | Miscellaneous | Tolerances

Reset | Explode Factor: | Assembly Center | Edge Coloring | Thicken

Mesh | Update | Mesh | Mesh Control | Mesh Edit | Metric Graph | Probe

Outline

Filter: Name

- Model (A4)
  - Geometry
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
    - ACP-Pre\Surface
  - Coordinate Systems
  - Connections
  - Mesh
    - Face Sizing 2

Details of "Face Sizing 2" - Sizing

| | |
|---|---|
| **Scope** | |
| Scoping Method | Geometry Selection |
| Geometry | 1 Face |
| **Definition** | |
| Suppressed | No |
| Type | Element Size |
| Element Size | Default (3. mm) |
| **Advanced** | |
| Size Function | Uniform |
| Behavior | Soft |
| Growth Rate | Default (1.8450 ) |

Face Sizing 2
28-11-2020 12:15
Face Sizing 2

ANSYS
R18.1

ANSYS Workbench Mesh Status

Meshing...

Highlight

**Part:** ACP-Pre
**Status:** Converting triangles to quads... (1/3) [80%]

Stop

300.00 600.00 (mm)
150.00 450.00

Y Z X

Geometry | Print Preview | Report Preview

Messages

| | Text | Association | Timestamp |
|---|---|---|---|
| Warninc | One or more parts were found to be unmodified so smart updated. | | Saturday, November 28, 2020 12:11:33 PM |

1 Message | No Selection | Metric (mm, kg, N, s, mV, mA) Degrees rad/s Celsius

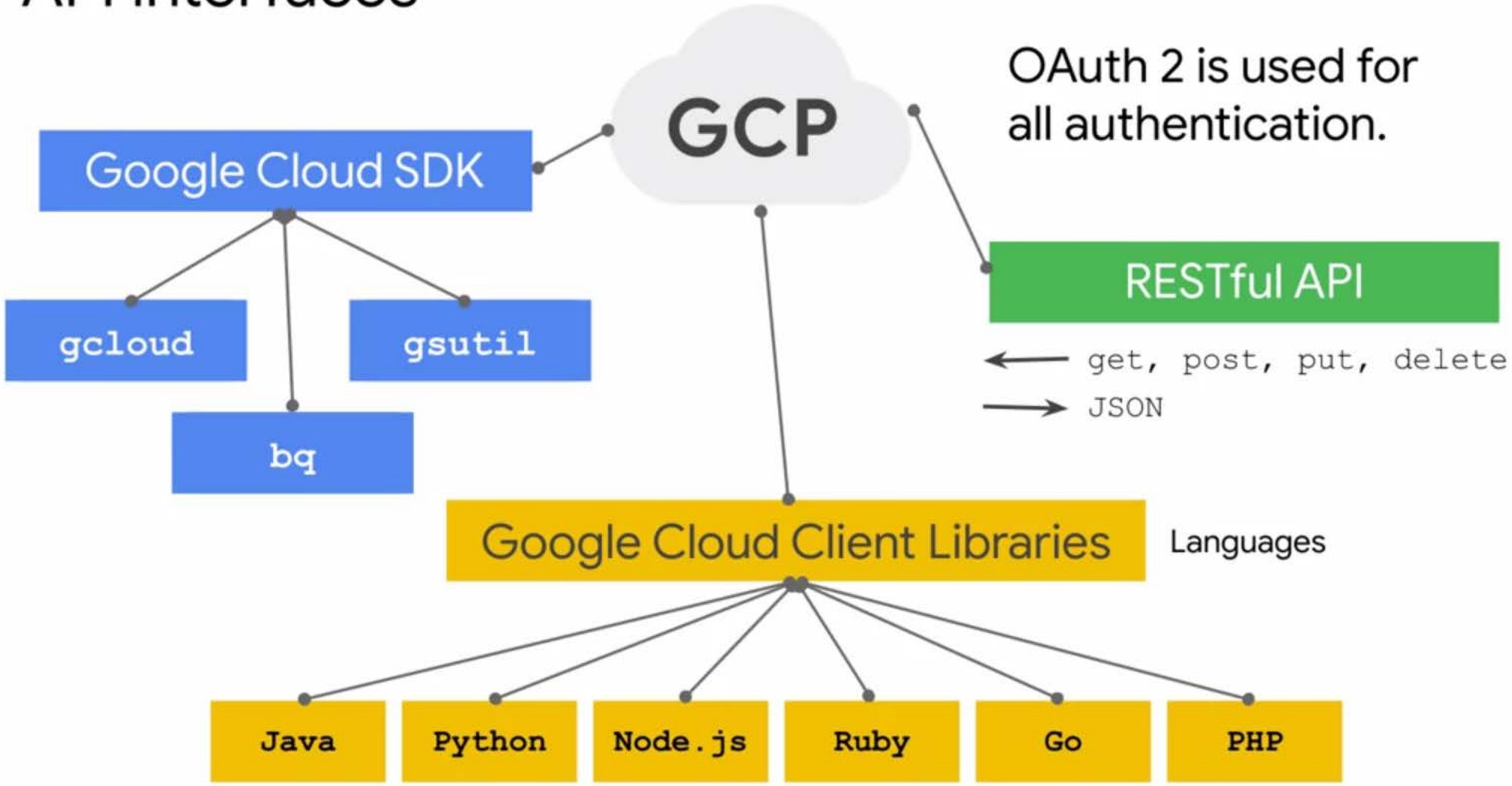
89% ENG 12:16 28-11-2020

# GCP Console, Cloud SDK and Cloud Shell

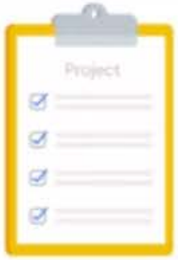

| Name | Zone | Internal IP | External IP | Connect |
|---|---|---|---|---|
| nginxstack-1 | us-central1-f | 10.128.0.3 (nic0) | 35.238.84.245 | SSH |
| nginxstack-2 | us-central1-f | 10.128.0.4 (nic0) | 35.225.177.18 | SSH |
| nginxstack-3 | us-central1-f | 10.128.0.2 (nic0) | 35.239.250.238 | SSH |


```
$ gcloud compute instances list
NAME          ZONE           INTERNAL_IP   EXTERNAL_IP
nginxstack-1  us-central1-f  10.128.0.3    35.238.84.245
nginxstack-2  us-central1-f  10.128.0.4    35.225.177.18
nginxstack-3  us-central1-f  10.128.0.2    35.239.250.238
```

# API interfaces



# Projects and networks




**A project:**

- Associates objects and services with billing.
- Contains networks (up to 5) that can be shared/peered.

**A network:**

- Has no IP address range.
- Is global and spans all available regions.
- Contains subnetworks.
- Is available as default, auto, or custom.

# 3 VPC network types

## Default

- Every project
- One subnet per region
- Default firewall rules

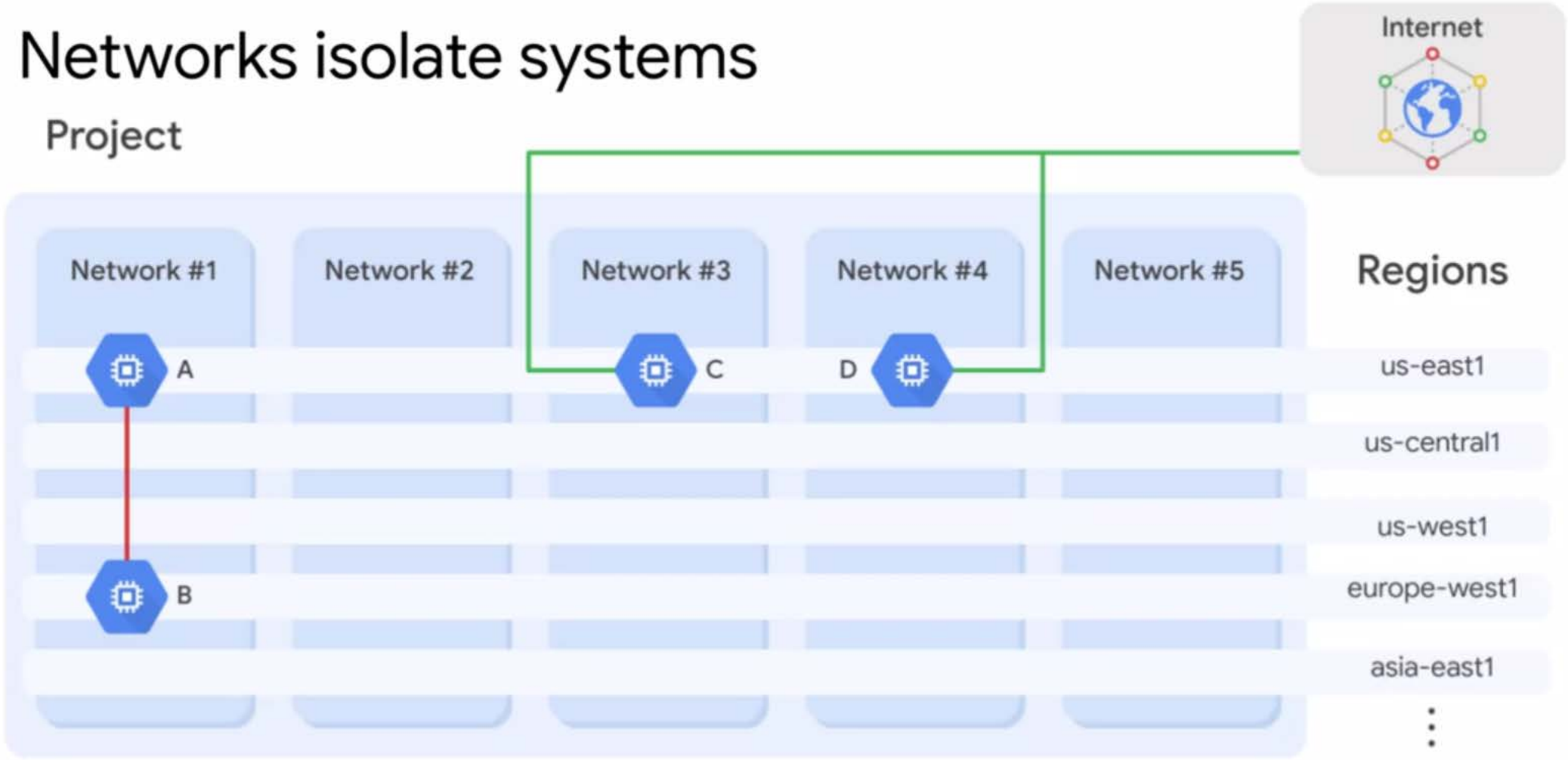
## Auto Mode

- Default network
- One subnet per region
- Regional IP allocation
- Fixed /20 subnetwork per region
- Expandable up to /16

## Custom Mode

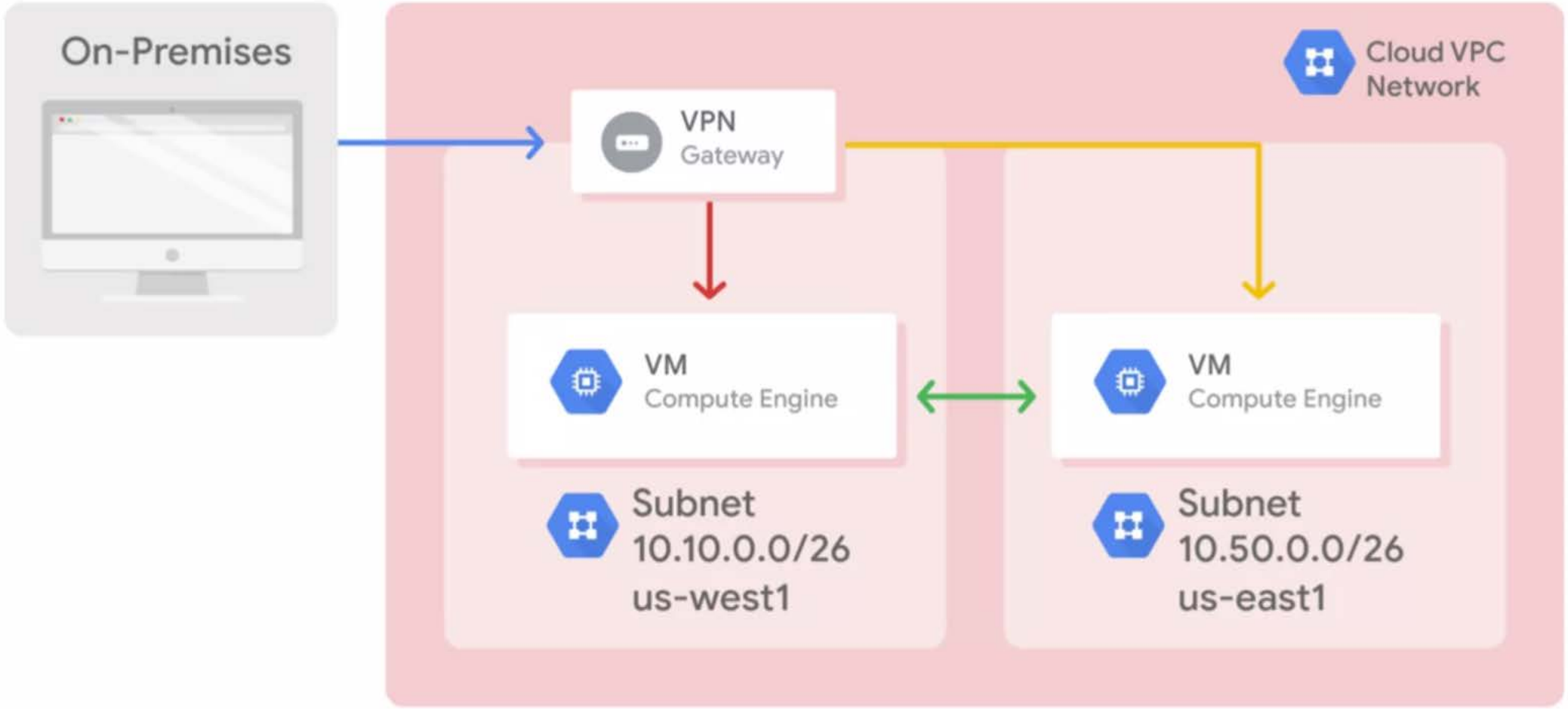
- No default subnets created
- Full control of IP ranges
- Regional IP allocation
- Expandable to any RFC 1918 size

# Networks isolate systems

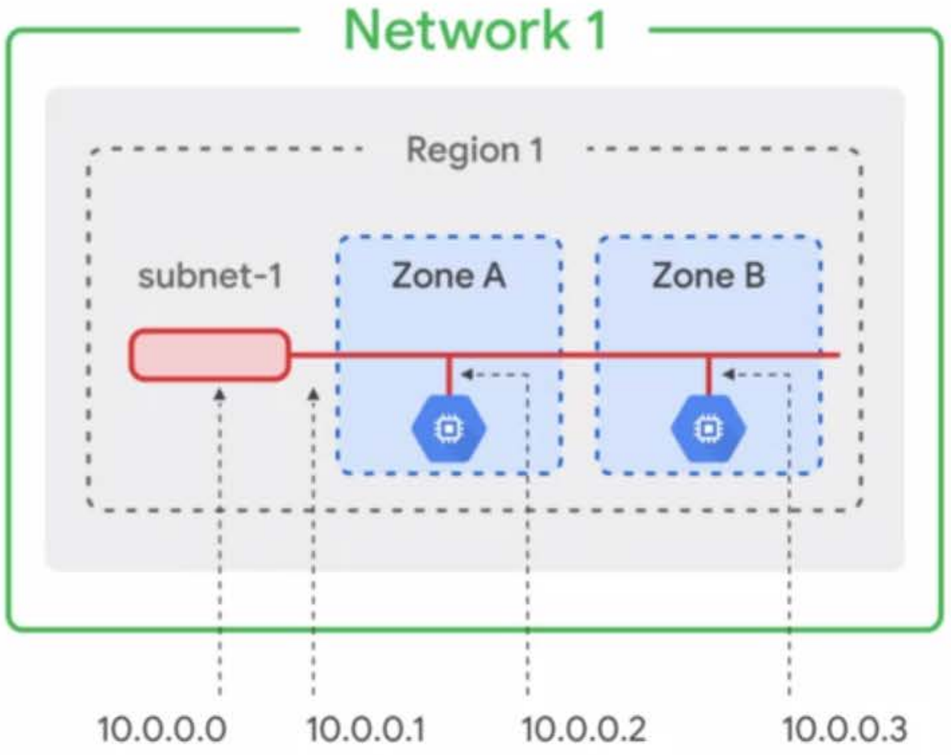



- A and B can communicate over internal IPs *even though they are in different regions.*
- C and D must communicate over external IPs *even though they are in the same region.*

# Google's VPC is global



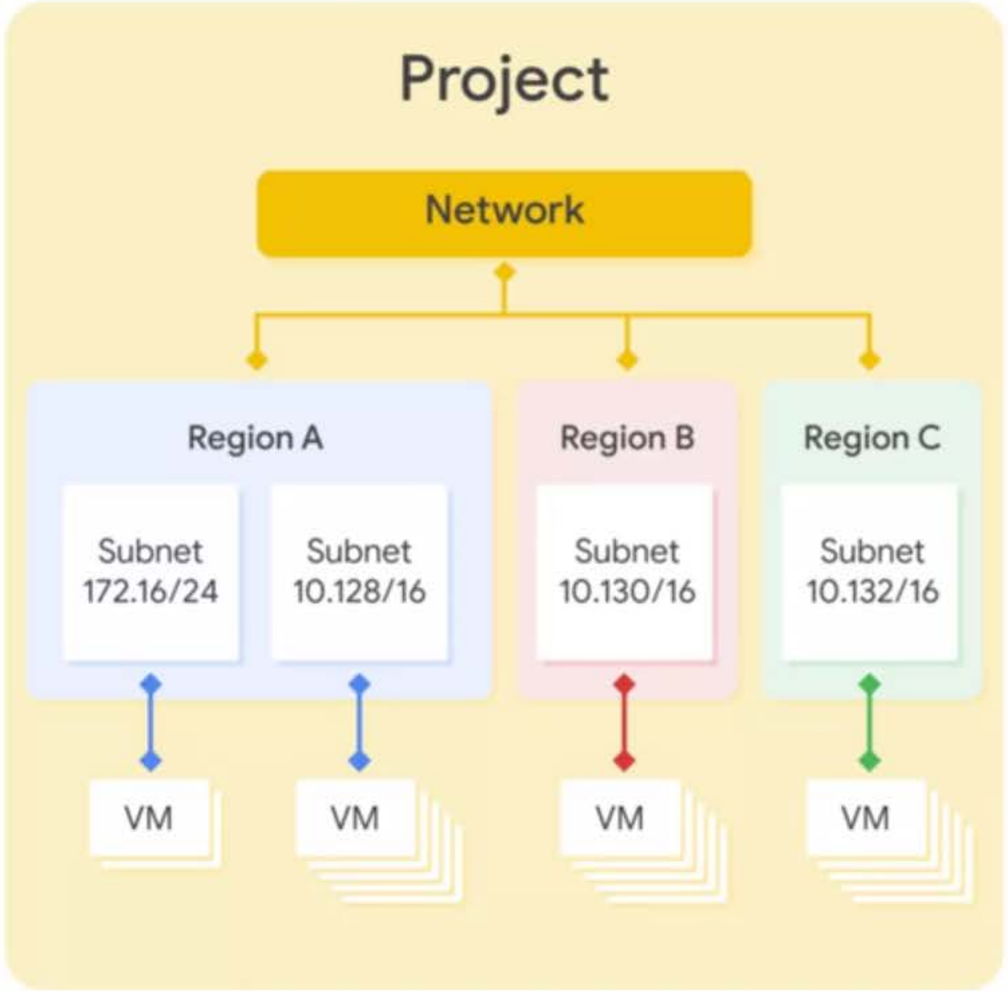
# Subnetworks cross zones




- VMs can be on the same subnet but in different zones.
- A single firewall rule can apply to both VMs.

# Expand subnets without re-creating instances

- Cannot overlap with other subnets
- Must be inside the RFC 1918 address spaces
- Can expand but not shrink
- Auto mode can be expanded from /20 to /16
- Avoid large subnets

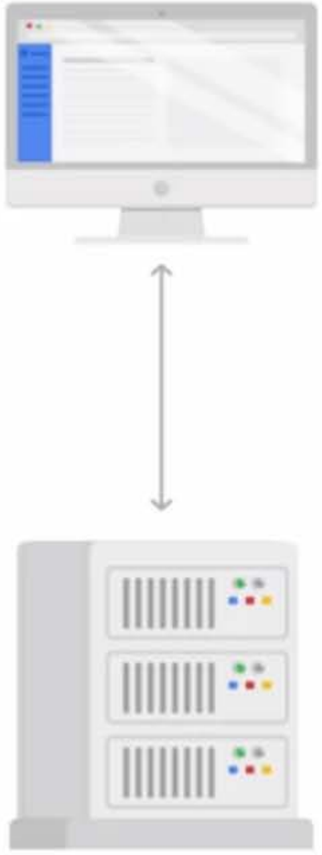


# VMs can have internal and external IP addresses


Cloud External IP Addresses

| Internal IP | External IP |
| --- | --- |
| Allocated from subnet range to VMs by DHCP<br>DHCP lease is renewed every 24 hours<br>VM name + IP is registered with network-scoped DNS | Assigned from pool (ephemeral)<br>Reserved (static) and billed more when not attached to a running VM<br>VM doesn't know external IP; it is mapped to the internal IP |

# DNS resolution for internal addresses



Each instance has a hostname that can be resolved to an internal IP address:

- The hostname is the same as the instance name.
- FQDN is [hostname].[zone].c.[project-id].internal

Example: my-server.us-central1-a.c.guestbook-151617.internal

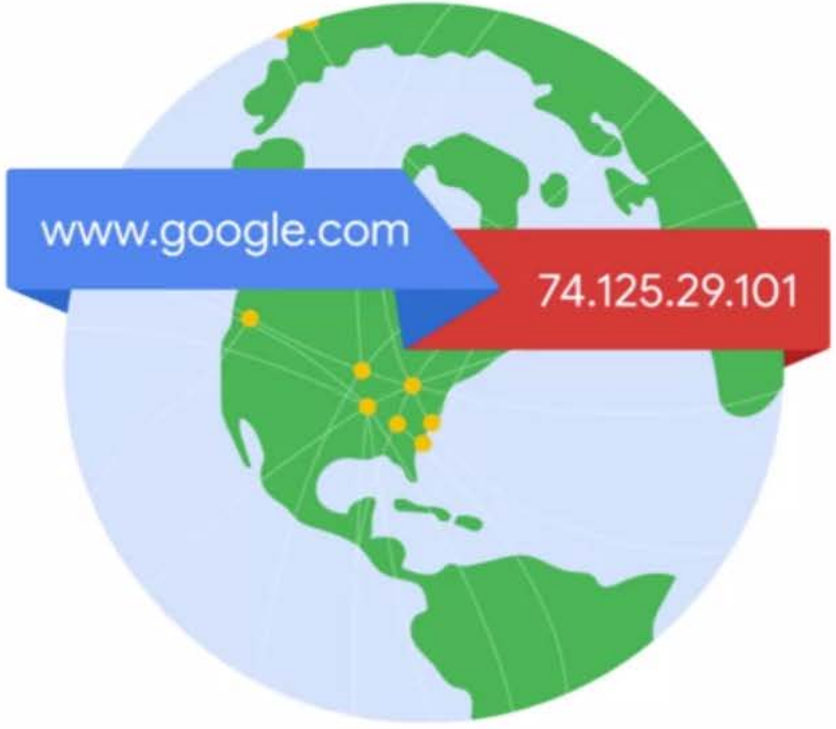
Name resolution is handled by internal DNS resolver:

- Provided as part of Compute Engine (169.254.169.254).
- Configured for use on instance via DHCP.
- Provides answer for internal and external addresses.

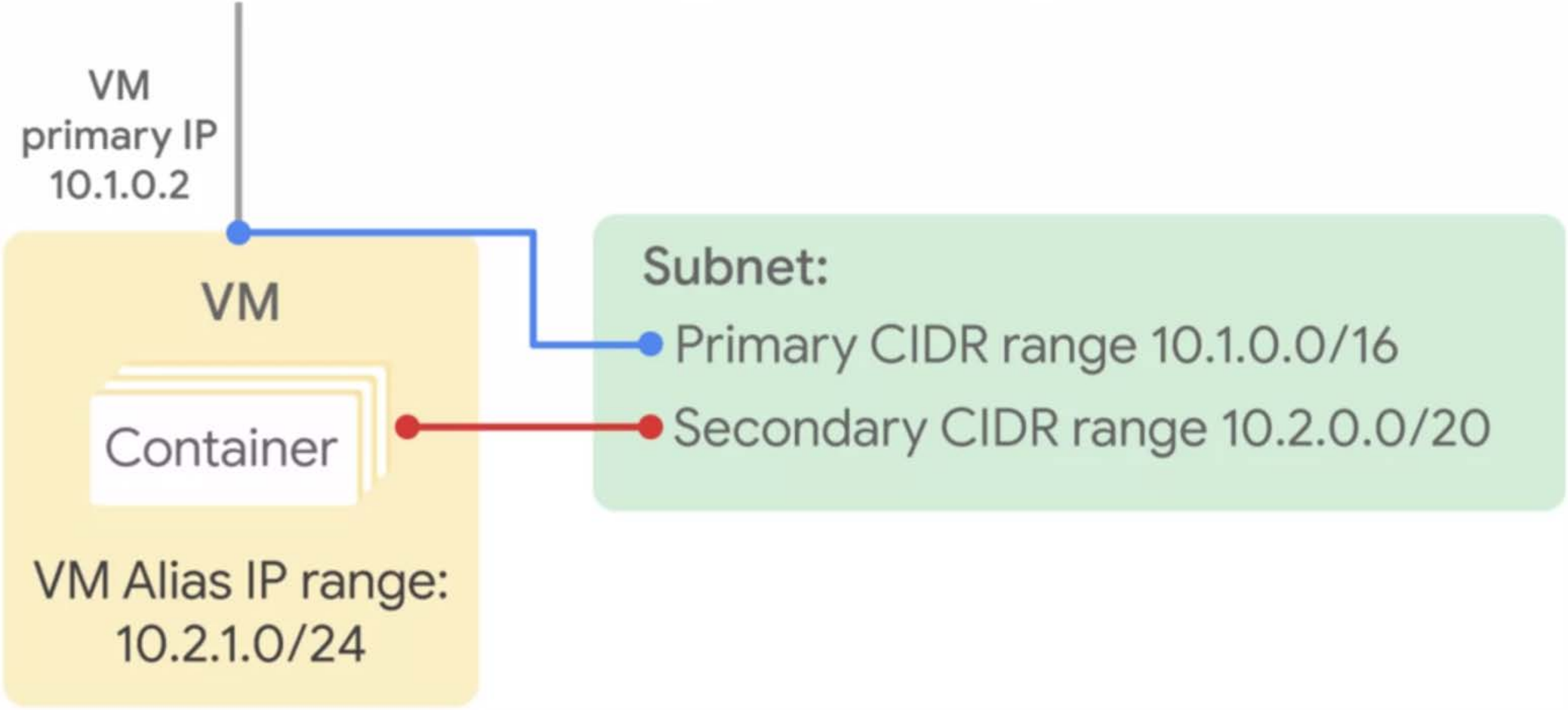
# Host DNS zones using Cloud DNS


Cloud DNS

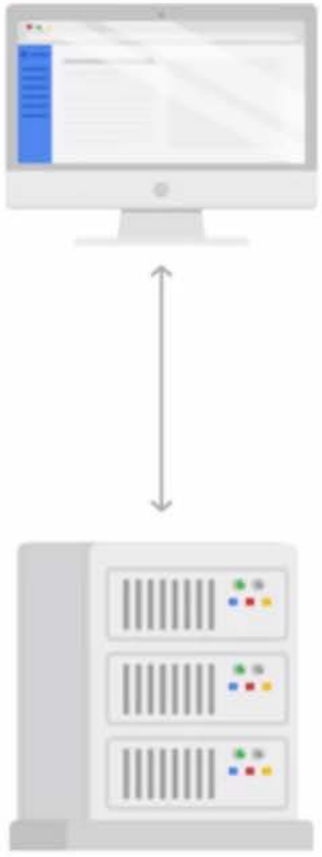
- Google's DNS service
- Translate domain names into IP address
- Low latency
- High availability (100% uptime SLA)
- Create and update millions of DNS records
- UI, command line, or API



# Assign a range of IP addresses as aliases to a VM's network interface using alias IP ranges



# DNS resolution for external addresses



- Instances with external IP addresses can allow connections from hosts outside the project.
  - Users connect directly using external IP address.
  - Admins can also publish public DNS records pointing to the instance.
    - Public DNS records are not published automatically.
- DNS records for external addresses can be published using existing DNS servers (outside of GCP).
- DNS zones can be hosted using Cloud DNS.

# A route is a mapping of an IP range to a destination

Cloud Routes

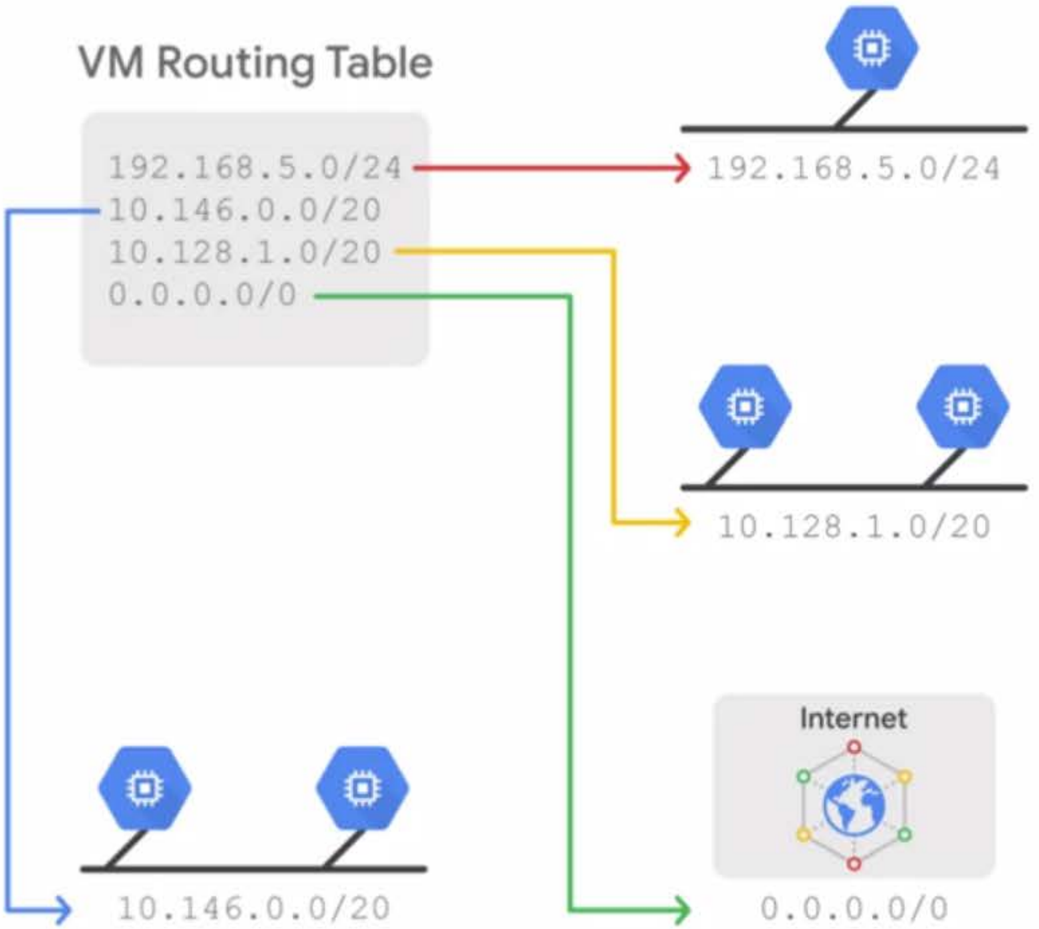
Every network has:

- Routes that let instances in a network send traffic directly to each other.
- A default route that directs packets to destinations that are outside the network.

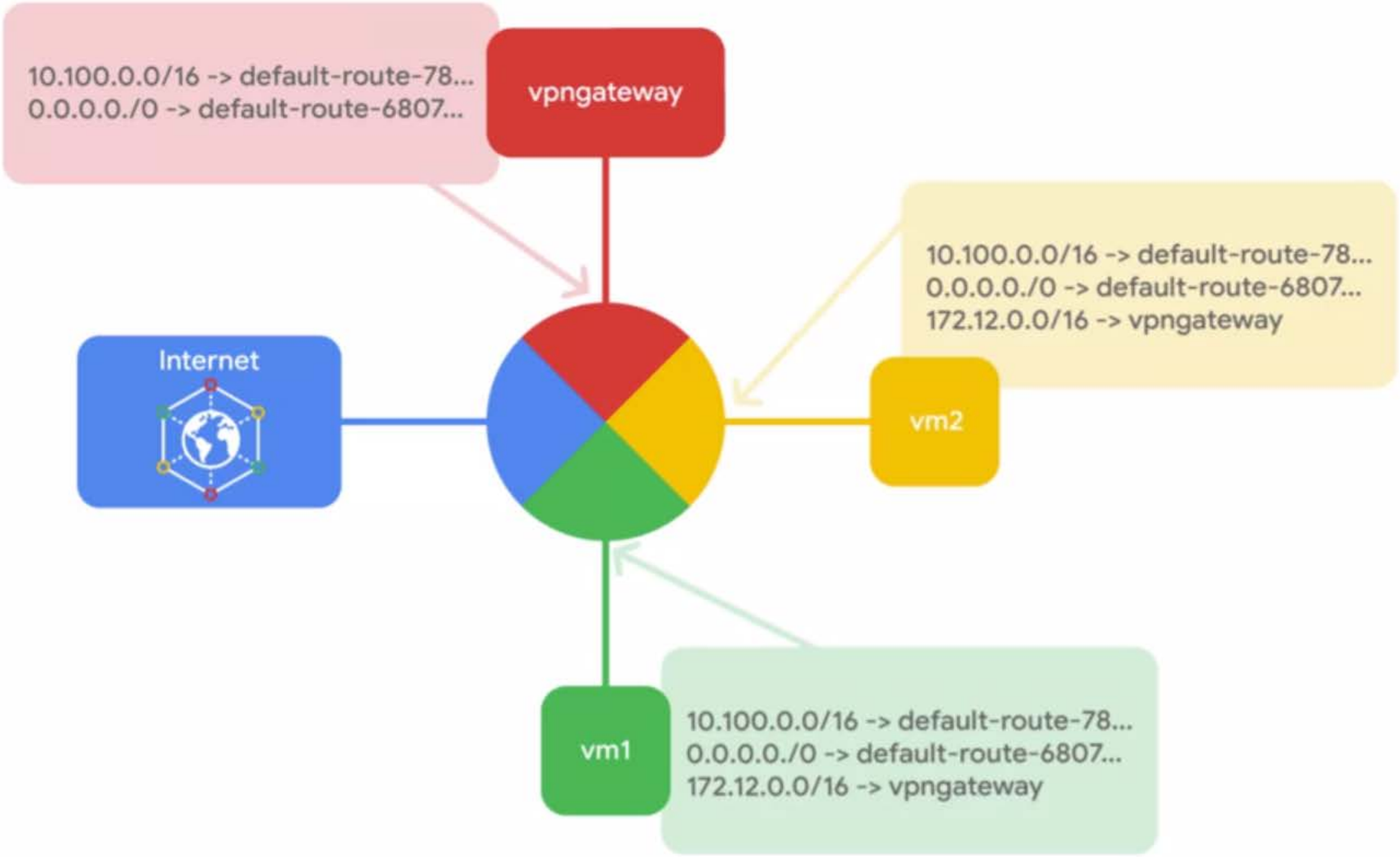
*Firewall rules must also allow the packet.*

# Routes map traffic to destination networks

- Apply to traffic egressing a VM.
- Forward traffic to most specific route.
- Are created when a subnet is created.
- Enable VMs on same network to communicate.
- Destination is in CIDR notation.
- Traffic is delivered only if it also matches a firewall rule.



# Instance routing tables



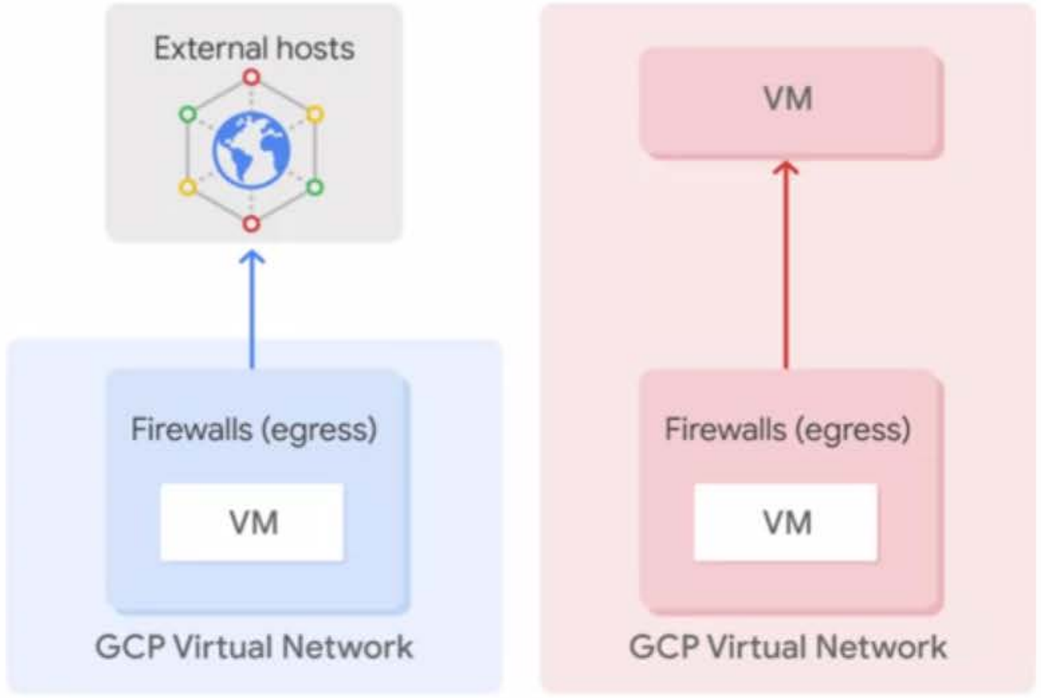
# Firewall rules protect your VM instances from unapproved connections


- VPC network functions as a distributed firewall.
- Firewall rules are applied to the network as a whole.
- Connections are *allowed* or *denied* at the instance level.
- Firewall rules are stateful.
- Implied *deny all* ingress and *allow all* egress.

# Routes map traffic to destination networks

| Parameter | Details |
| --- | --- |
| `direction` | Inbound connections are matched against `ingress` rules only |
| | Outbound connections are matched against `egress` rules only |
| `source or destination` | For the `ingress` direction, `sources` can be specified as part of the rule with IP addresses, source tags, or a source service account |
| | For the `egress` direction, `destinations` can be specified as part of the rule with one or more ranges of IP addresses |
| `protocol` and `port` | Any rule can be restricted to apply to specific protocols only or specific combinations of protocols and ports only |
| `action` | To allow or deny packets that match the direction, protocol, port, and source or destination of the rule |
| `priority` | Governs the order in which rules are evaluated; the first matching rule is applied |
| Rule assignment | All rules are assigned to all instances, but you can assign certain rules to certain instances only |

# GCP firewall use case: Egress



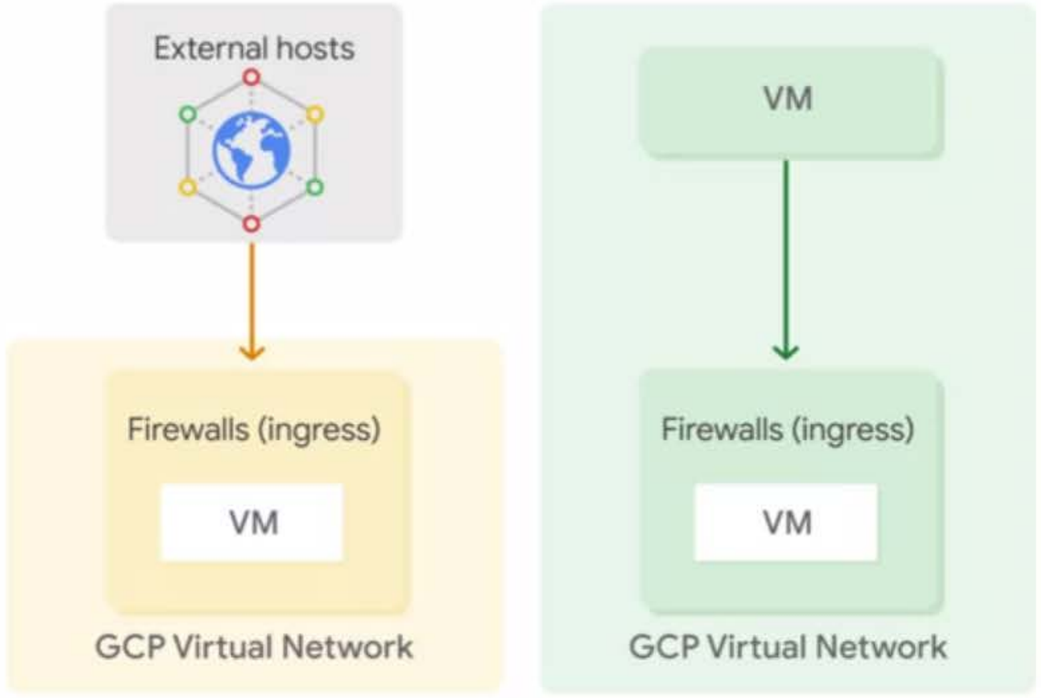

Conditions:

- Destination CIDR ranges
- Protocols
- Ports

Action:

- Allow: permit the matching egress connection
- Deny: block the matching egress connection

# GCP firewall use case: Ingress




Conditions:

- Source CIDR ranges
- Protocols
- Ports

Action:

- Allow: permit the matching ingress connection
- Deny: block the matching ingress connection

# Network pricing (subject to change)

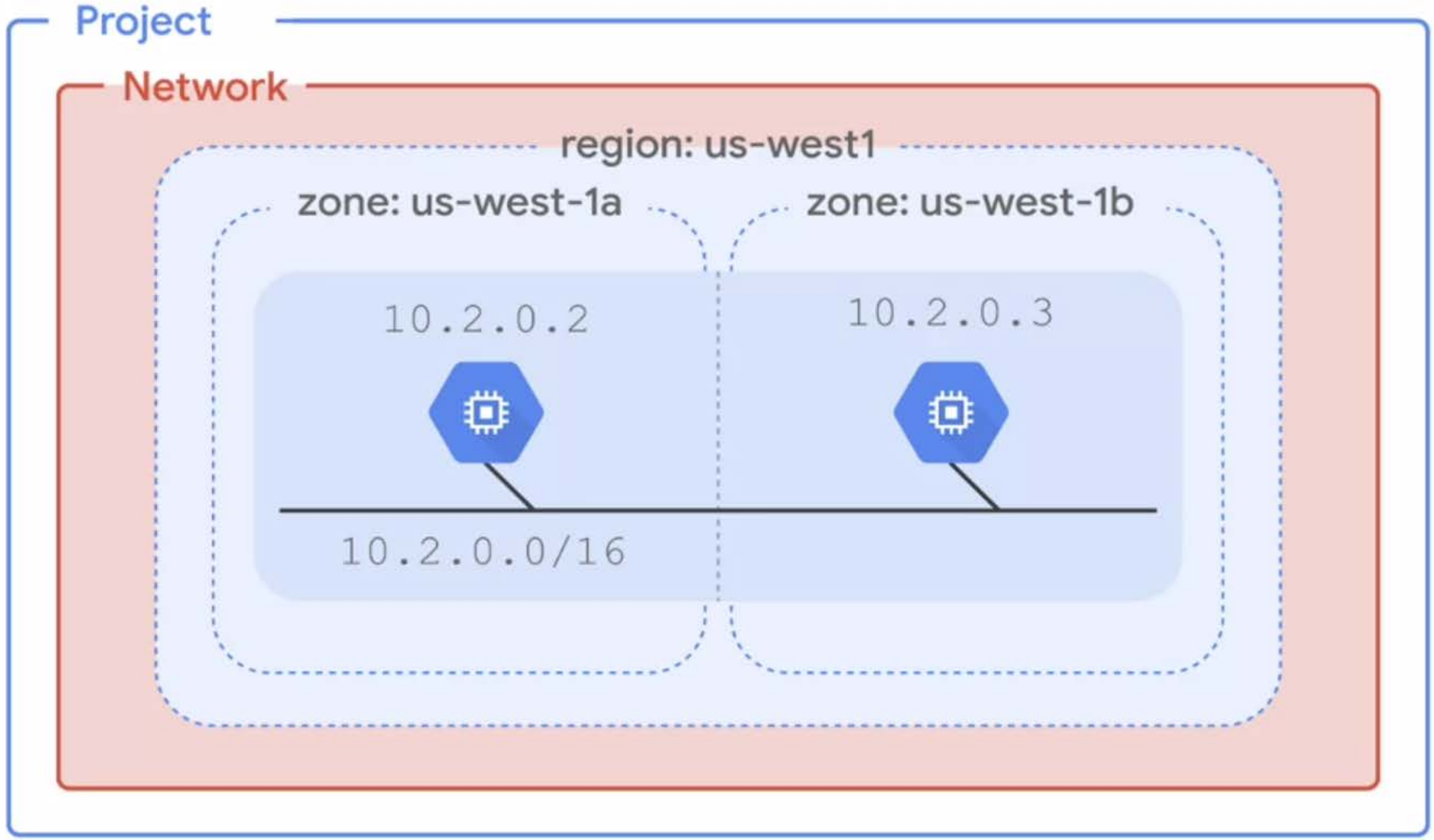
| Traffic type | Price (USD) |
| --- | --- |
| Ingress | No charge |
| Egress to the same zone (internal IP address) | No charge |
| Egress to Google products (YouTube, Maps, Drive) | No charge |
| Egress to a different GCP service (within same region; exceptions) | No charge |
| Egress between zones in the same region (per GB) | $0.01 |
| Egress to the same zone (external IP address, per GB) | $0.01 |
| Egress between regions within the US and Canada (per GB) | $0.01 |
| Egress between regions, not including traffic between US regions | Varies by region |

# External IP address pricing (us-central1)

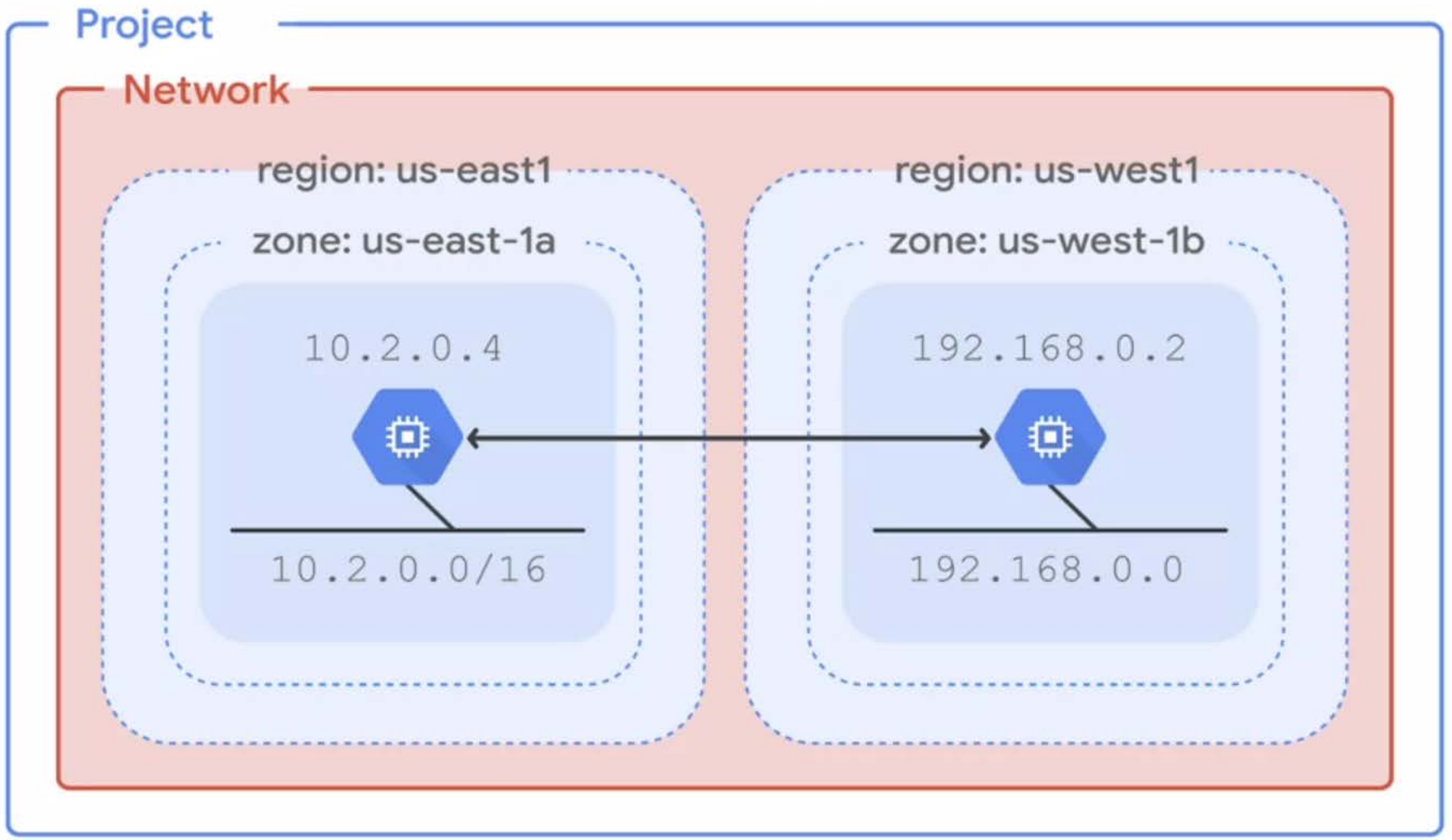
(Subject to change)

| Type | Price/Hour (USD) |
| --- | --- |
| Static IP address (assigned but unused) | $0.010 |
| Static and ephemeral IP addresses in use on **standard** VM instances | $0.004 |
| Static and ephemeral IP addresses in use on **preemptible** VM instances | $0.002 |
| Static and ephemeral IP addresses attached to forwarding rules | No charge |

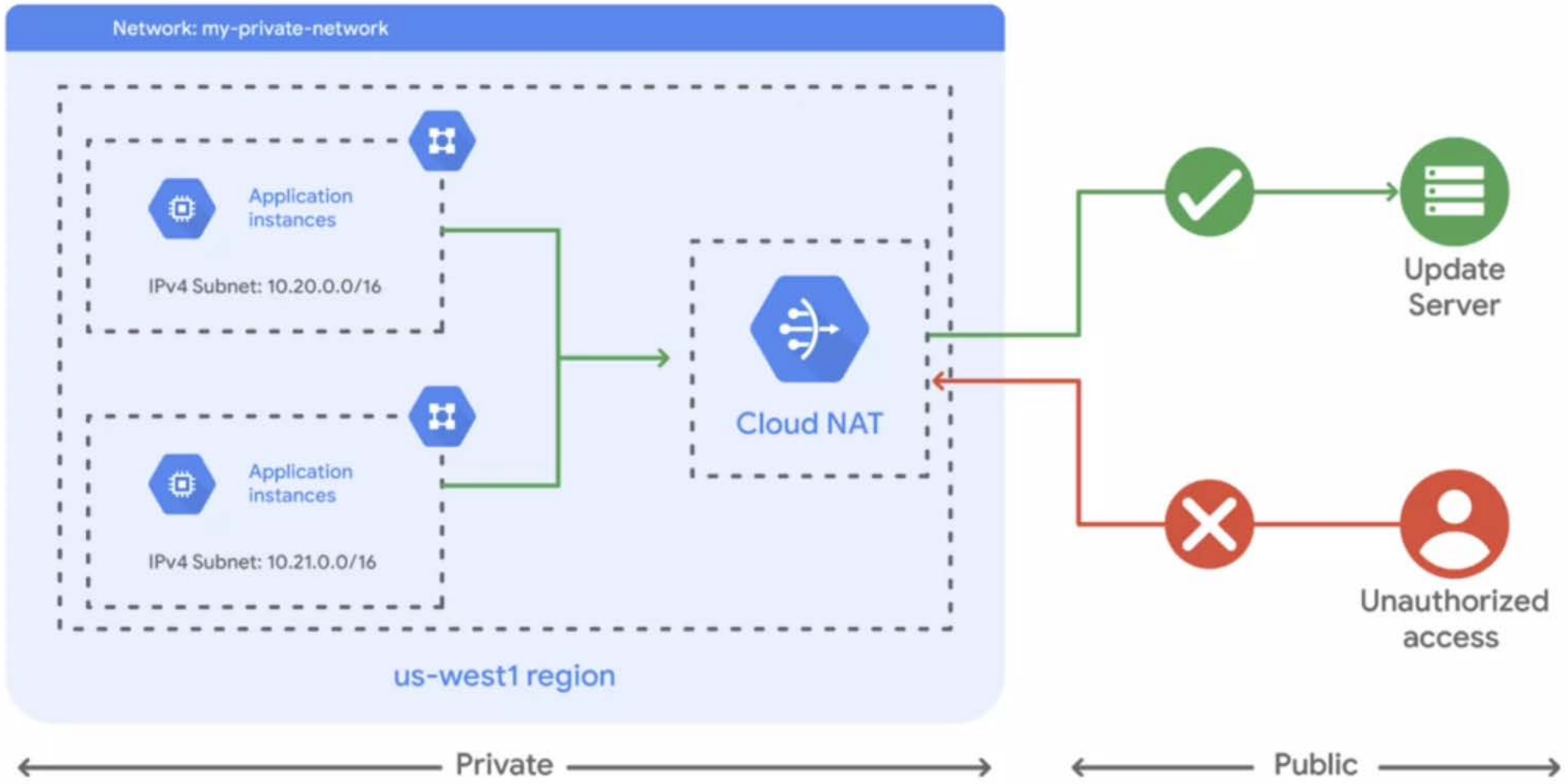
# Increased availability with multiple zones



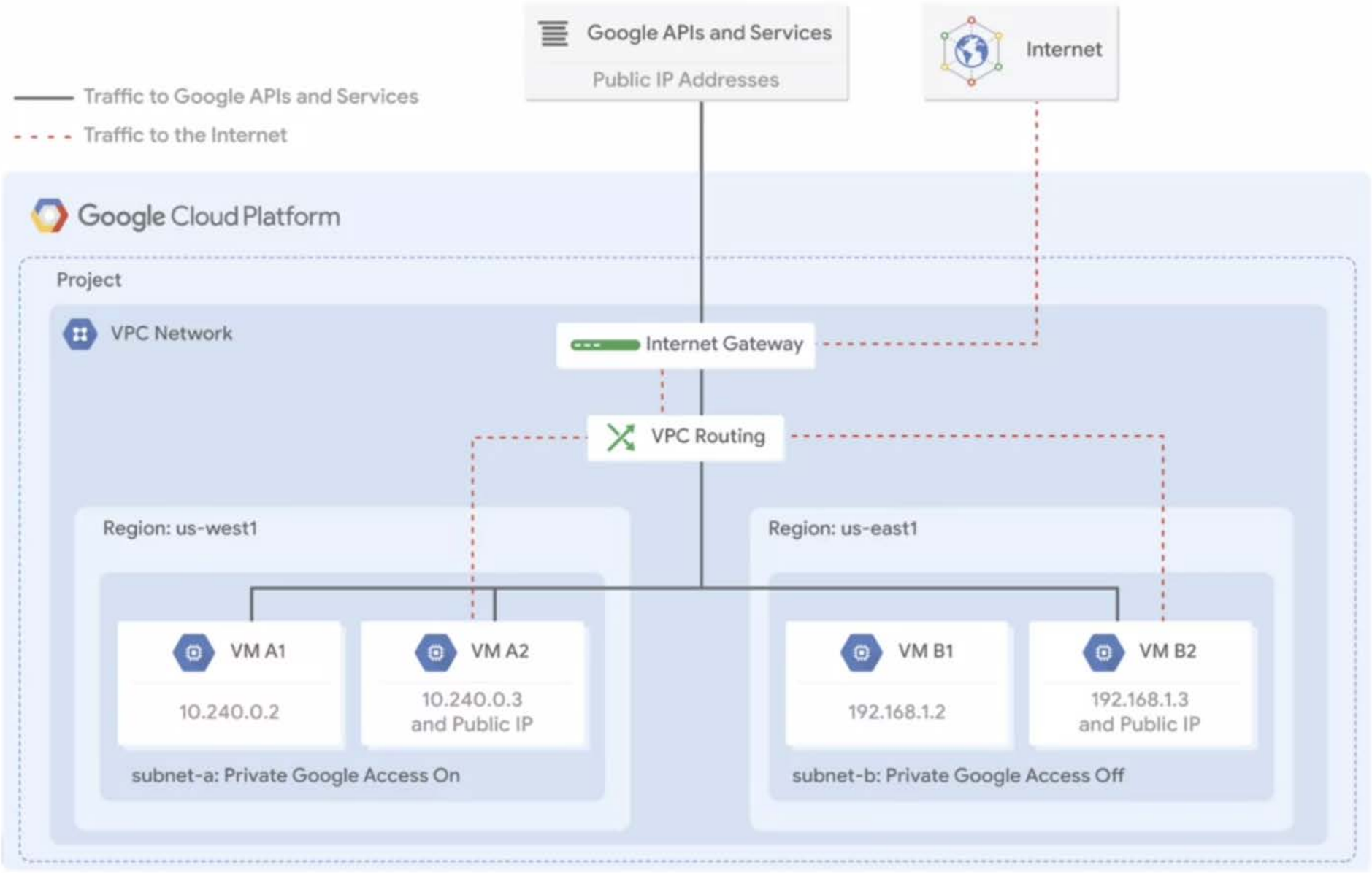
# Globalization with multiple regions



# Cloud NAT provides internet access to private instances



# Private Google Access to Google APIs and services



# GCP compute and processing options

| | Compute Engine | Kubernetes Engine | App Engine Standard | App Engine Flexible | Cloud Functions |
|---|---|---|---|---|---|
| **Language support** | Any | Any | Python<br>Node.js<br>Go<br>Java<br>PHP | Python<br>Node.js<br>Go<br>Java<br>PHP<br>Ruby<br>.NET<br>Custom Runtimes | Python<br>Node.js<br>Go |
| **Usage model** | IaaS | IaaS<br>PaaS | PaaS | PaaS | Microservices Architecture |
| **Scaling** | Server Autoscaling | Cluster | Autoscaling managed servers | | Serverless |
| **Primary use case** | General Workloads | Container Workloads | Scalable web applications<br>Mobile backend applications | | Lightweight Event Actions |

# Compute Engine

Infrastructure as a Service (IaaS)

Predefined or custom machine types:

- vCPUs (cores) and Memory (RAM)
- Persistent disks: HDD, SSD, and Local SSD
- Networking
- Linux or Windows

Compute Engine

# VM access

## Linux: SSH

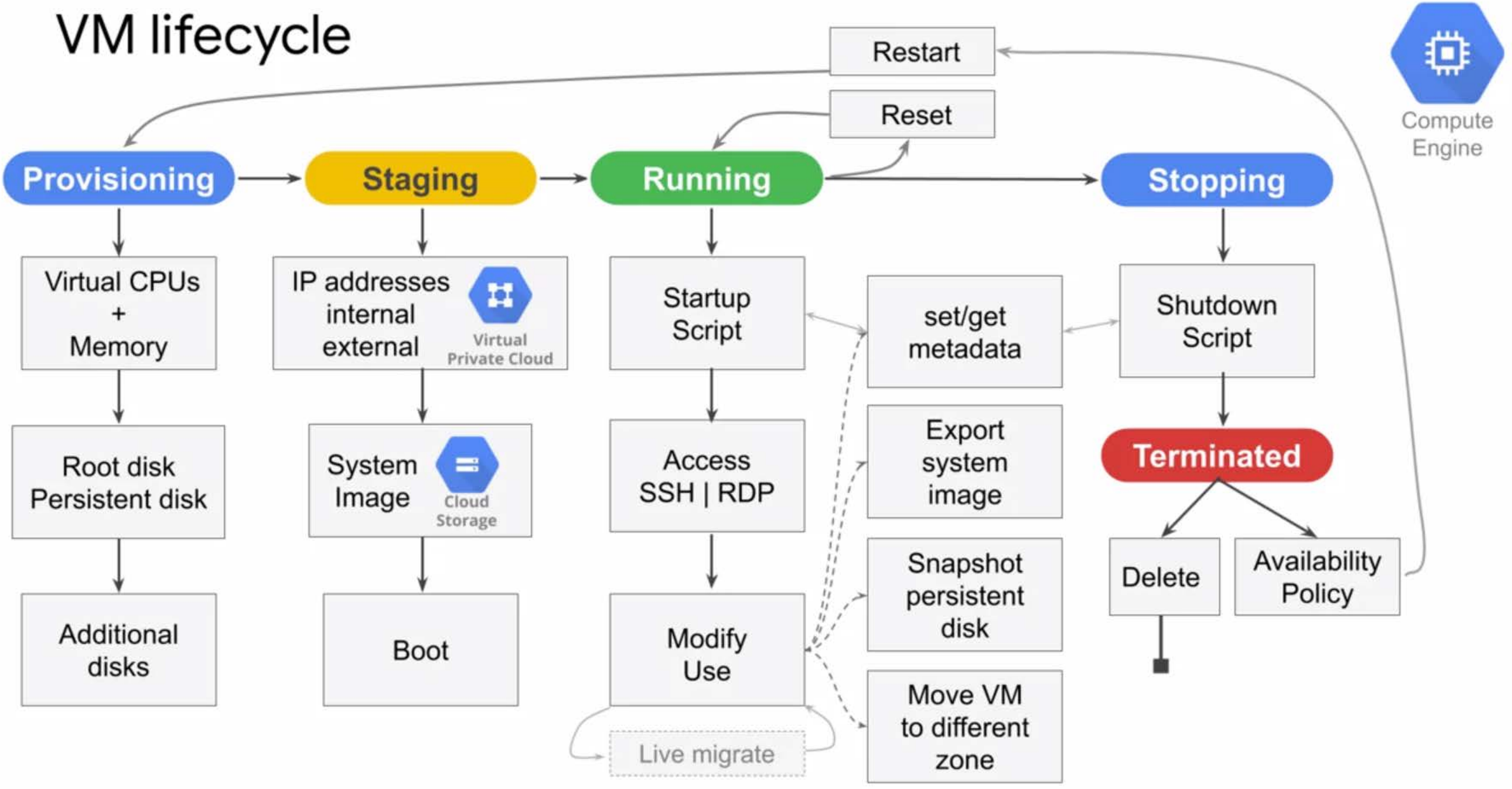
- SSH from GCP Console or CloudShell via Cloud SDK
- SSH from computer or third-party client and generate key pair
- Requires firewall rule to allow tcp:22

## Windows: RDP

- RDP clients
- Powershell terminal
- Requires setting the Windows password
- Requires firewall rule to allow tcp:3389

the GCP Console to grant SSH capability to other users.

# VM lifecycle



# Changing VM state from running

| | methods | Shutdown Script time | state |
|---|---|---|---|
| `reset` | console, `gcloud`, API, OS | no | remains running |
| `restart` | console, `gcloud`, API, OS | no | terminated➡️running |
| `reboot` | OS: `sudo reboot` | ~90 sec | running➡️running |
| `stop` | console, `gcloud`, API | ~90 sec | running➡️terminated |
| `shutdown` | OS: `sudo shutdown` | ~90 sec | running➡️terminated |
| `delete` | console, `gcloud`, API | ~90 sec | running➡️N/A |
| *preemption* | *automatic* | ~30 sec | N/A |

"ACPI Power Off"

# Availability policy: Automatic changes

Called "scheduling options" in SDK/API

Automatic restart

- Automatic VM restart due to crash or maintenance event
  - Not preemption or a user-initiated terminate

On host maintenance

- Determines whether host is live-migrated or terminated due to a maintenance event. Live migration is the default.

Live migration

- During maintenance event, VM is migrated to different hardware without interruption.
- Metadata indicates occurrence of live migration.

# Stopped (Terminated) VM

No charge for stopped VM

- Charged for attached disks and IPs

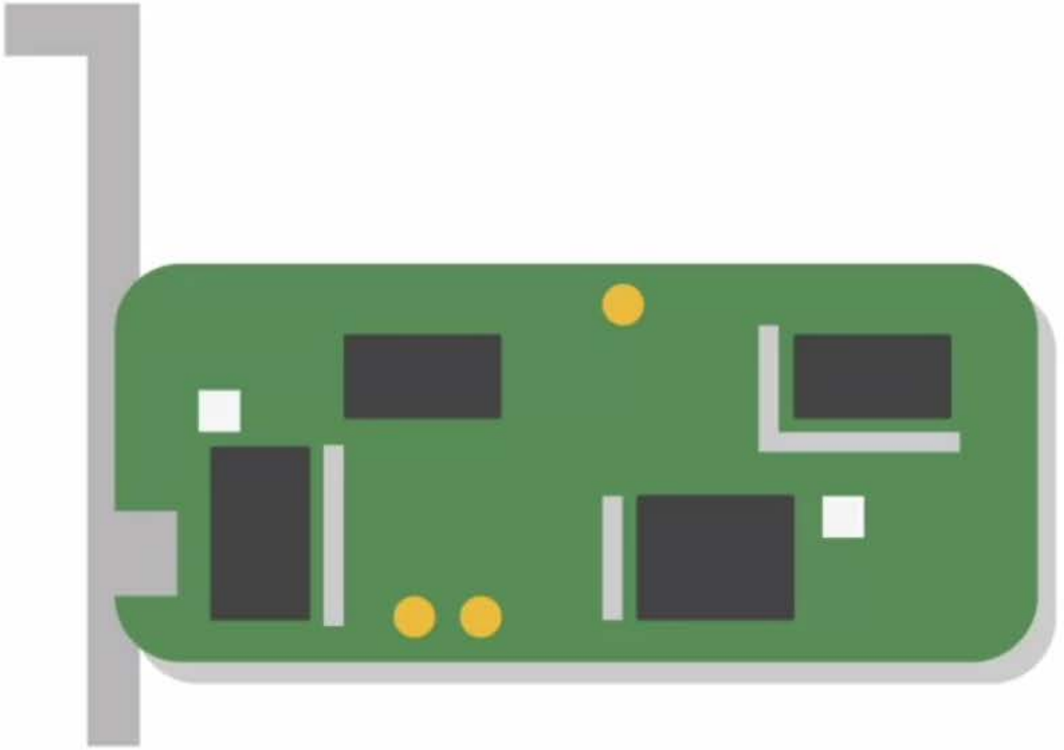
Actions

- Change the machine type.
- Add or removed attached disks; change auto-delete settings.
- Modify instance tags.
- Modify custom VM or project-wide metadata.
- Remove or set a new static IP.
- Modify VM availability policy.
- Can’t change the image of a stopped VM.

# Networking

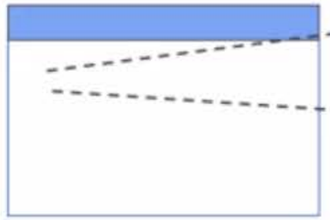
Robust networking features

- Default, custom networks
- Inbound/outbound firewall rules
  - IP based
  - Instance/group tags
- Regional HTTPS load balancing
- Network load balancing
  - Does not require pre-warming
- Global and multi-regional subnetworks



# Creating a VM

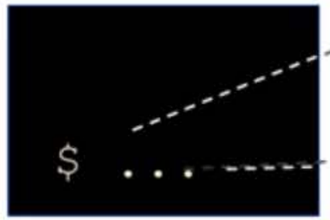
1



console.google.com

VM instances CREATE INSTANCE IMPORT VM

2




command line
*including Cloudshell*

```
$ gcloud compute instances create [instance-name]
```

3

REST API

Many VM options

- Project
- Region
- Zone
- Subnetwork
- Machine type
- Disk options
- Image
- IP options

# Machine types

Predefined machine types: Ratio of GB of memory per vCPU

- Standard
- High-memory
- High-CPU
- Memory-optimized
- Compute-optimized
- Shared-core

Custom machine types:

- You specify the amount of memory and number of vCPUs.

# Standard machine types

3.75 GB of memory

1 vCPU

| Machine name | vCPUs | Memory (GB) | Max # PD | Max total PD size |
|---|---|---|---|---|
| n1-standard-1 | 1 | 3.75 | 128 | 64 TB |
| n1-standard-2 | 2 | 7.50 | | |
| n1-standard-4 | 4 | 15 | | |
| n1-standard-8 | 8 | 30 | | |
| n1-standard-16 | 16 | 60 | | |
| n1-standard-32 | 32 | 120 | | |
| n1-standard-64 | 64 | 240 | | |
| n1-standard-96 | 96 | 360 | | |

# High-memory machine types

6.5 GB of memory

1 vCPU

| Machine name | vCPUs | Memory (GB) | Max # PD | Max total PD size |
|---|---|---|---|---|
| n1-highmem-2 | 2 | 13 | 128 | 64 TB |
| n1-highmem-4 | 4 | 26 | | |
| n1-highmem-8 | 8 | 52 | | |
| n1-highmem-16 | 16 | 104 | | |
| n1-highmem-32 | 32 | 208 | | |
| n1-highmem-64 | 64 | 416 | | |
| n1-highmem-96 | 96 | 624 | | |

# High-CPU machine types

0.9 GB of memory

1 vCPU

| Machine name | vCPUs | Memory (GB) | Max # PD | Max total PD size |
|---|---|---|---|---|
| n1-highcpu-2 | 2 | 1.80 | 128 | 64 TB |
| n1-highcpu-4 | 4 | 3.60 | | |
| n1-highcpu-8 | 8 | 7.20 | | |
| n1-highcpu-16 | 16 | 14.4 | | |
| n1-highcpu-32 | 32 | 28.8 | | |
| n1-highcpu-64 | 64 | 57.6 | | |
| n1-highcpu-96 | 96 | 86.4 | | |

# Memory-optimized machine types

>14 GB of memory

1 vCPU

| Machine name | vCPUs | Memory (GB) | Max # PD | Max total PD size |
|---|---|---|---|---|
| n1-ultramem-40 | 40 | 961 | 128 | 64 TB |
| n1-ultramem-80 | 80 | 1922 | | |
| n1-megamem-96 | 96 | 1433.6 | | |
| n1-ultramem-160 | 160 | 3844 | | |

# Compute-optimized machine types

Highest performance per vCPU (3.8Ghz sustained all-core turbo)

| Machine name | vCPUs | Memory (GB) | Max # PD | Max total PD size |
|---|---|---|---|---|
| c2-standard-4 | 4 | 16 | 128 | 64 TB |
| c2-standard-8 | 8 | 32 | | |
| c2-standard-16 | 16 | 64 | | |
| c2-standard-30 | 30 | 120 | | |
| c2-standard-60 | 60 | 240 | | |

# Shared-core machine types

| Machine name | vCPUs | Memory (GB) | Max # PD | Max total PD size |
|---|---|---|---|---|
| f1-micro | 0.2 | 0.60 | 16 | 3 TB |
| g1-small | 0.5 | 1.70 | | |

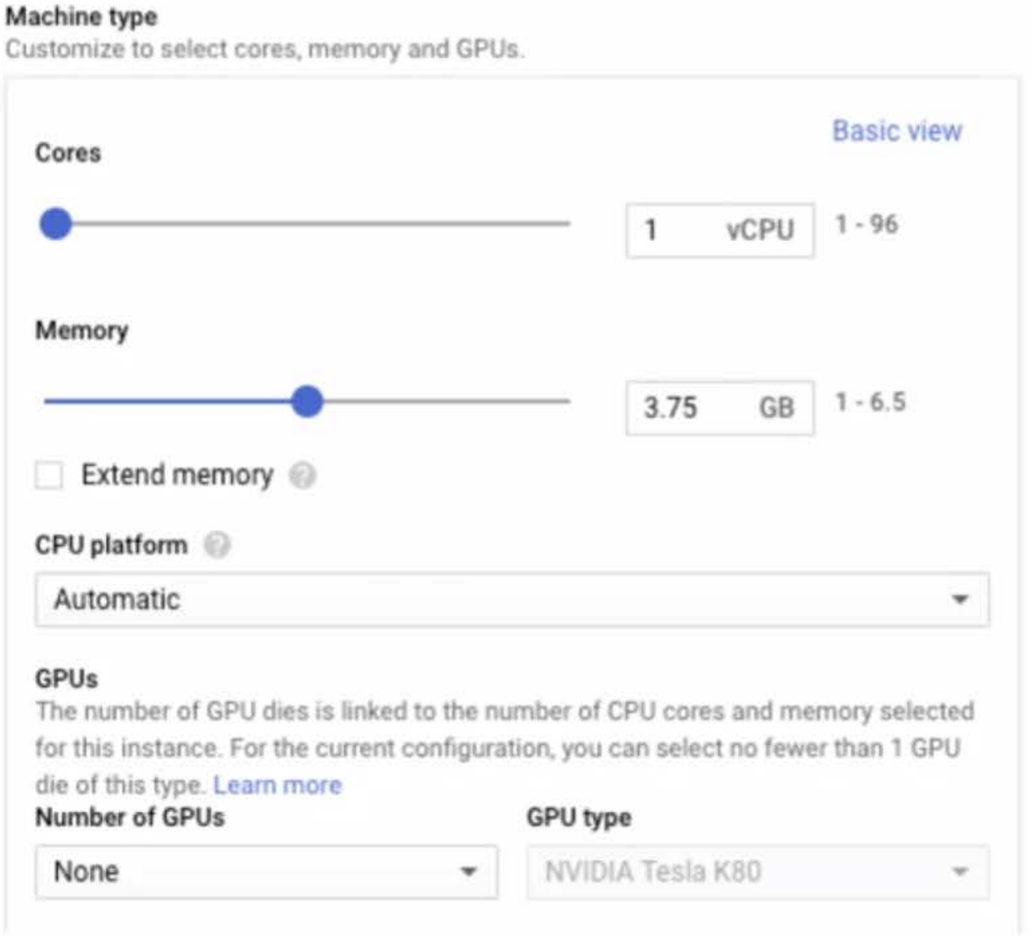
# Creating custom machine types

## When to select custom:

- Requirements fit between the predefined types
- Need more memory or more CPU

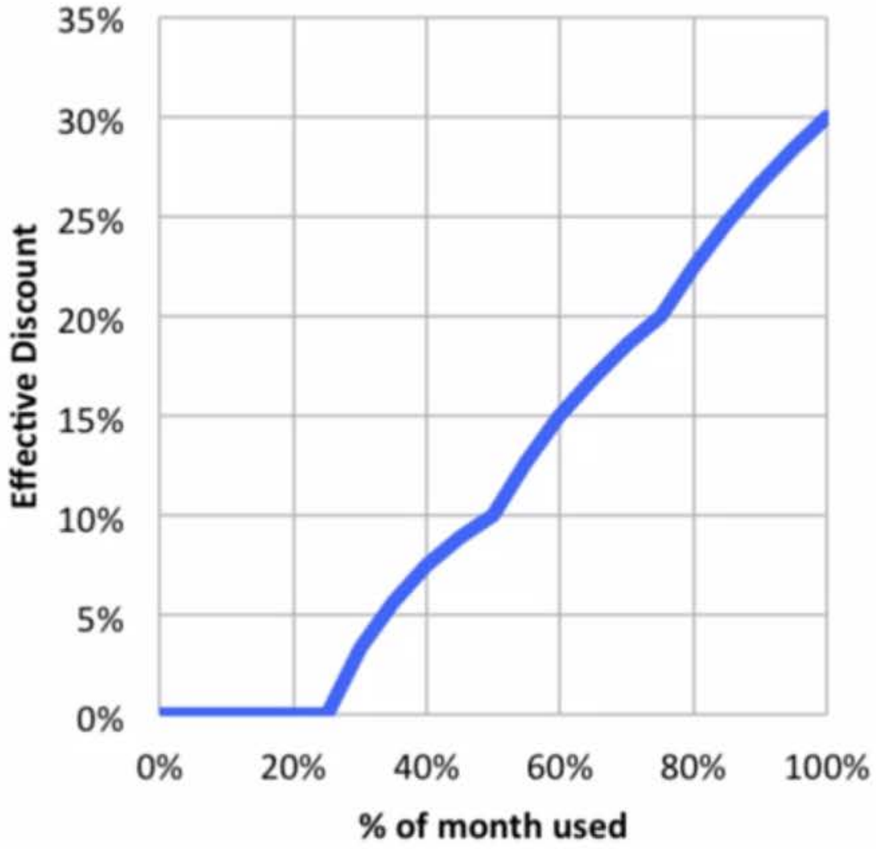
Customize the amount of memory and vCPU for your machine:

- Either 1 vCPU or even number of vCPU
- 0.9 GB per vCPU, up to 6.5 GB per vCPU (default)
- Total memory must be multiple of 256 MB



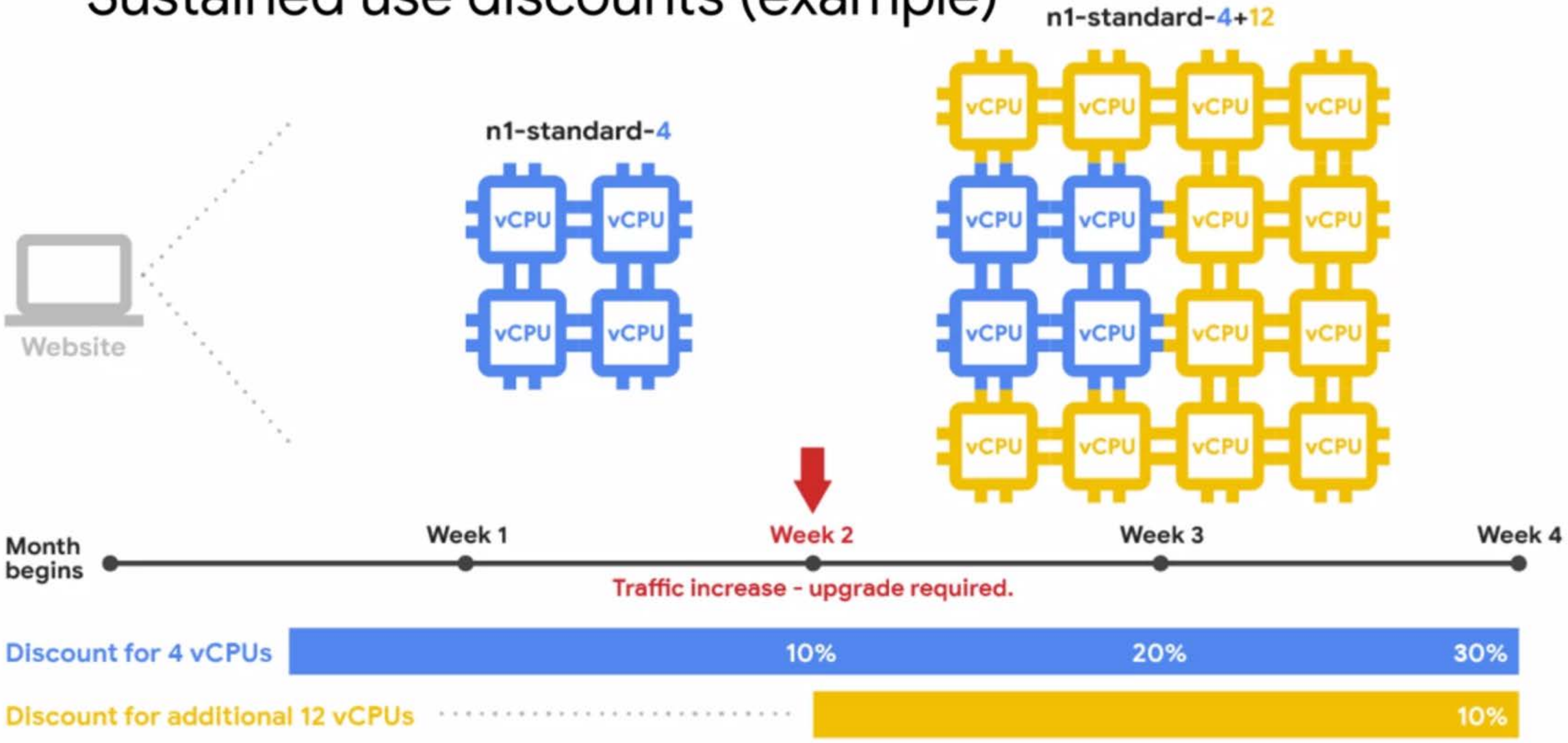
# Sustained use discounts

| Usage Level (% of month) | % at which incremental is charged |
|---|---|
| 0% - 25% | 100% of base rate |
| 25% - 50% | 80% of base rate |
| 50% - 75% | 60% of base rate |
| 75% - 100% | 40% of base rate |




Up to 30% net discount for instances that run the entire month

# Sustained use discounts (example)



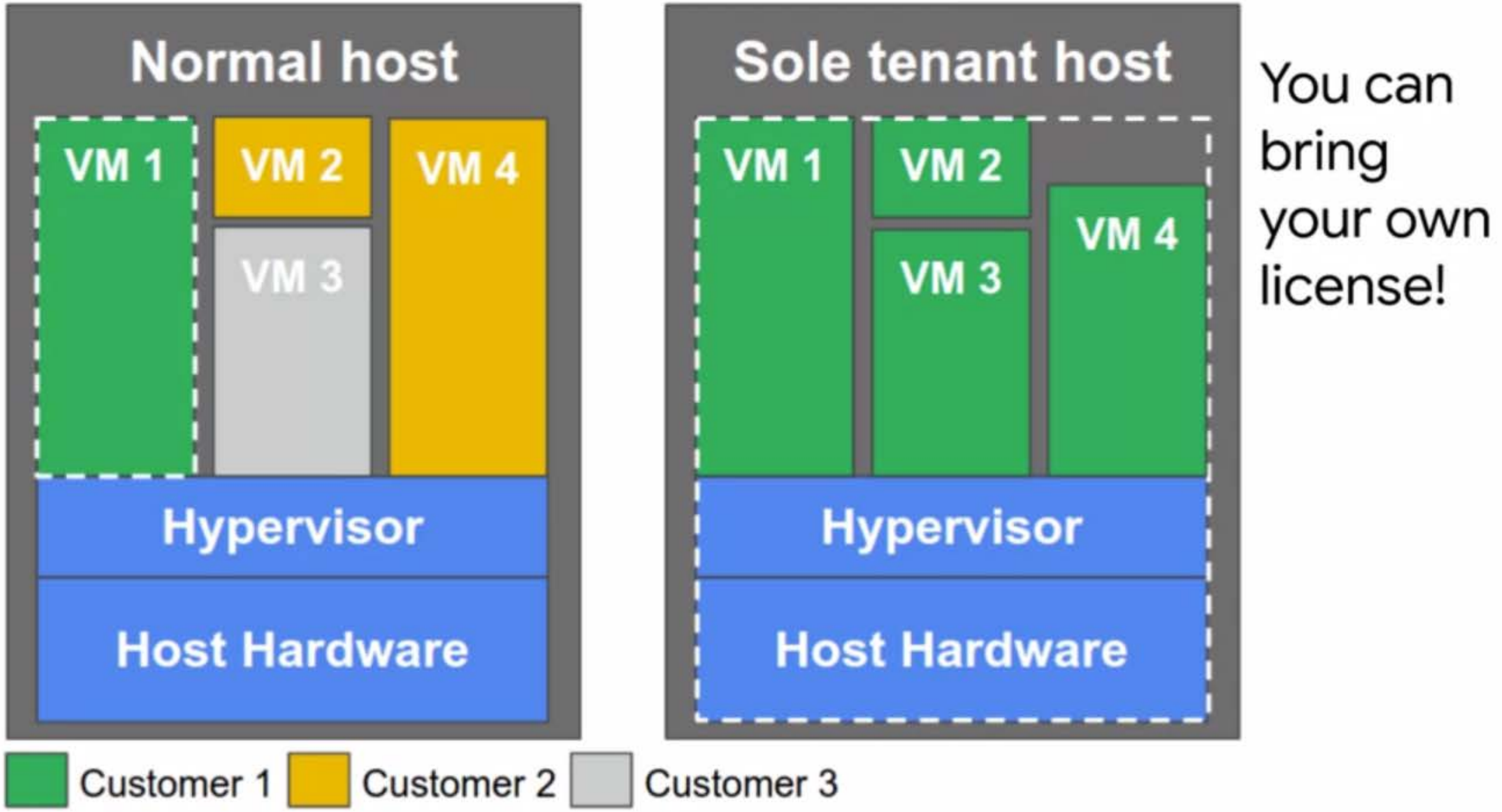
# Pricing

- Per-second billing, with minimum of 1 minute
  - vCPUs, GPUs, and GB of memory
- Resource-based pricing:
  - Each vCPU and each GB of memory is billed separately
- Discounts:
  - Sustained use
  - Committed use
  - Preemptible VM instances
- Recommendation Engine
  - Notifies you of underutilized instances
- Free usage limits

# Preemptible

- Lower price for interruptible service (up to 80%)
- VM might be terminated at any time
  - No charge if terminated in the first minute
  - 24 hours max
  - 30-second terminate warning, but not guaranteed
    - *Time for a shutdown script*
- No live migrate; no auto restart
- You *can* request that CPU quota for a region be split between regular and preemption
  - Default: preemptible VMs count against region CPU quota

# Sole-tenant nodes physically isolate workloads




You can bring your own license!

# Images

- Public base images
  - Google, third-party vendors, and community; Premium images (p)
  - Linux
    - CentOS, CoreOS, Debian, RHEL(p), SUSE(p), Ubuntu, openSUSE, and FreeBSD
  - Windows
    - Windows Server 2019(p), 2016(p), 2012-r2(p)
    - SQL Server pre-installed on Windows(p)
- Custom images
  - Create new image from VM: pre-configured and installed SW
  - Import from on-prem, workstation, or another cloud
  - Management features: image sharing, image family, deprecation

# Boot disk

- VM comes with a single root persistent disk.
- Image is loaded onto root disk during first boot:
  - Bootable: you can attach to a VM and boot from it.
  - Durable: can survive VM terminate.
- Some OS images are customized for Compute Engine.
- Can survive VM deletion if “Delete boot disk when instance is deleted” is disabled.

# Persistent disks

Network storage appearing as a block device

- Attached to a VM  through the network interface
- Durable storage: *can* survive VM terminate
- Bootable: you can attach to a VM and boot from it
- Snapshots: incremental backups
- Performance: Scales with size

Features

- HDD (magnetic) or SSD (faster, solid-state) options
- Disk resizing: even running and attached!
- Can be attached in read-only mode to multiple VMs
- Zonal or Regional
- Encryption keys:
  - Google-managed
  - Customer-managed
  - Customer-supplied

# Local SSD disks are physically attached to a VM

- More IOPS, lower latency, and higher throughput than persistent disk
- 375-GB disk up to eight, total of 3 TB
- Data survives a reset, but not a VM stop or terminate
- VM-specific: cannot be reattached to a different VM

# RAM disk

- `tmpfs`
- Faster than local disk, slower than memory
  - Use when your application expects a file system structure and cannot directly store its data in memory
  - Fast scratch disk, or fast cache
- Very volatile; erase on stop or restart
- May need a larger machine type if RAM was sized for the application
- Consider using a persistent disk to back up RAM disk data

# Summary of disk options

| | Persistent disk HDD | Persistent disk SSD | Local SSD disk | RAM disk |
|---|---|---|---|---|
| Data redundancy | Yes | Yes | No | No |
| Encryption at rest | Yes | Yes | Yes | N/A |
| Snapshotting | Yes | Yes | No | No |
| Bootable | Yes | Yes | No | Not |
| Use case | General, bulk file storage | Very random IOPS | High IOPS and low latency | low latency and risk of data loss |

# Maximum persistent disks

| Machine Type | Disk number limit |
| --- | --- |
| Shared-core | 16 |
| Standard | 128 |
| High-memory | |
| High-CPU | |
| Memory-optimized | |
| Compute-optimized | |

# Persistent disk management differences

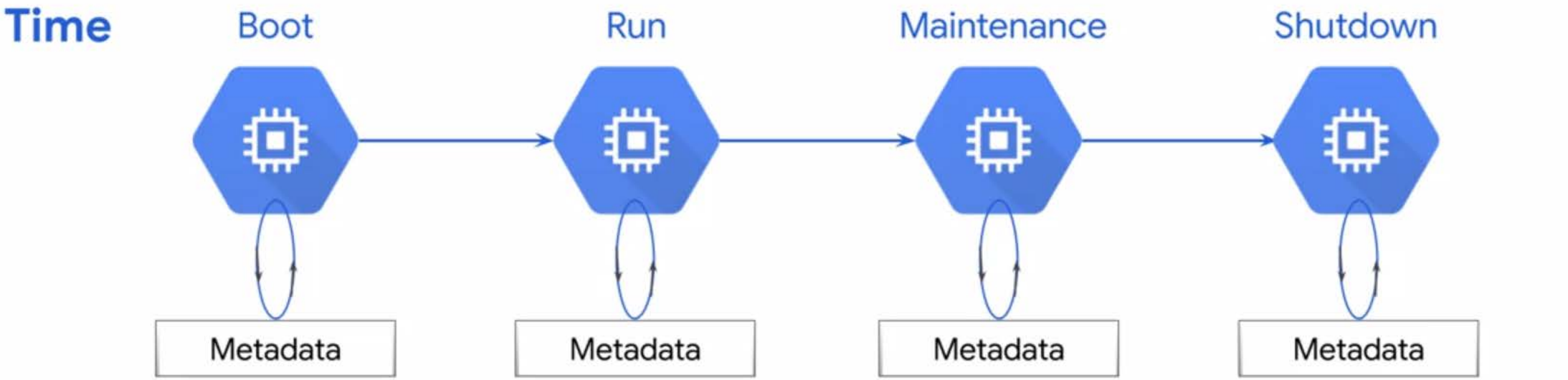
Cloud Persistent Disk

- Single file system is best
- Resize (grow) disks
- Resize file system
- Built-in snapshot service
- Automatic encryption

Computer Hardware Disk

- Partitioning
- Repartition disk
- Reformat
- Redundant disk arrays
- Subvolume management and snapshots
- Encrypt files before write to disk

# Metadata and scripts



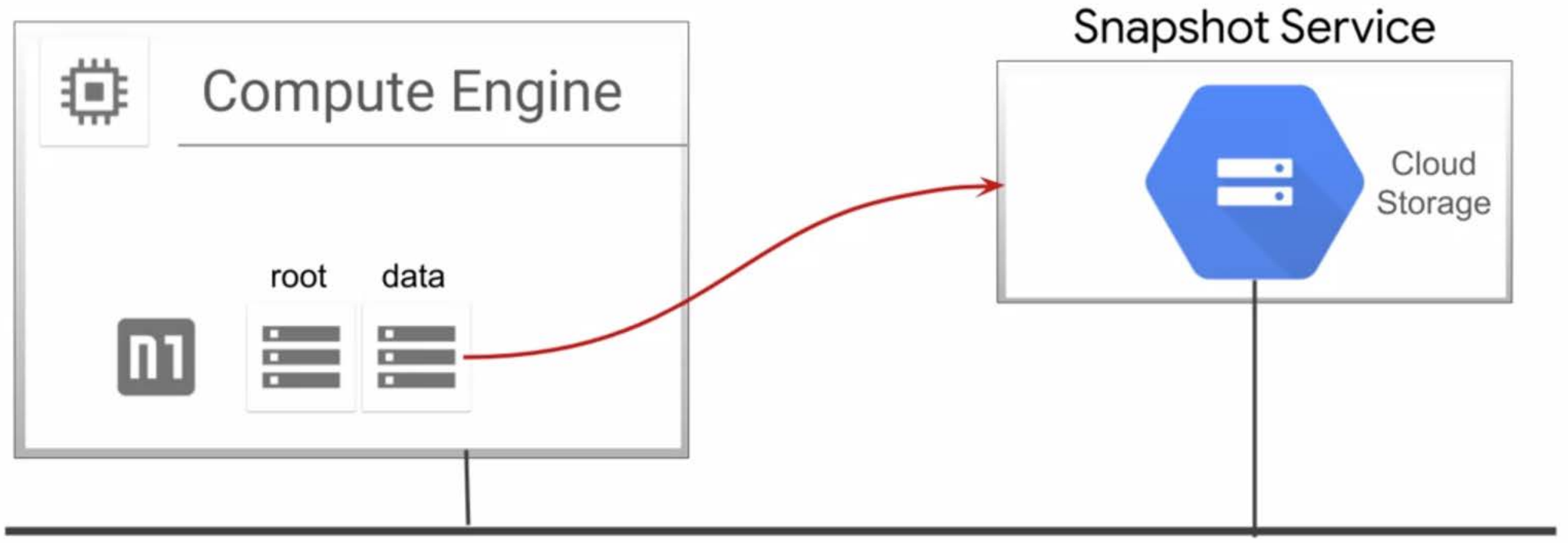
# Move an instance to a new zone

- Automated process (moving within region):
  - `gcloud compute instances move`
  - Update references to VM; not automatic
- Manual process (moving between regions):
  - Snapshot all persistent disks on the source VM.
  - Create new persistent disks in destination zone restored from snapshots.
  - Create new VM in the destination zone and attach new persistent disks.
  - Assign static IP to new VM.
  - Update references to VM.
  - Delete the snapshots, original disks, and original VM.

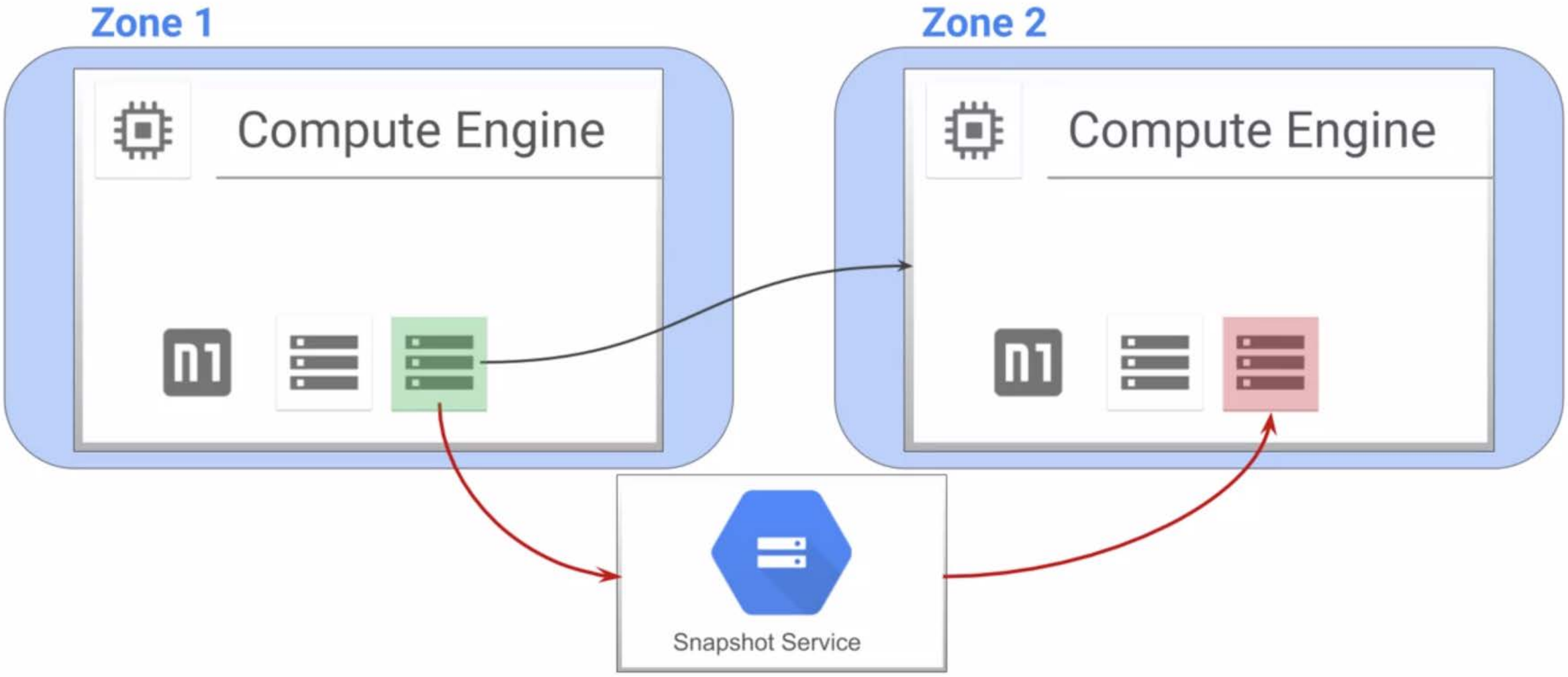
# Persistent disk snapshots

- Snapshot is not available for local SSD.
- Creates an *incremental* backup to Cloud Storage.
  - Not visible in *your* buckets; managed by the snapshot service.
  - Consider cron jobs for periodic incremental backup.
- Snapshots can be restored to a new persistent disk.
  - New disk can be in another region or zone in the same project.
  - Basis of VM migration: "moving" a VM to a new zone.
    - Snapshot doesn't back up VM metadata, tags, etc.

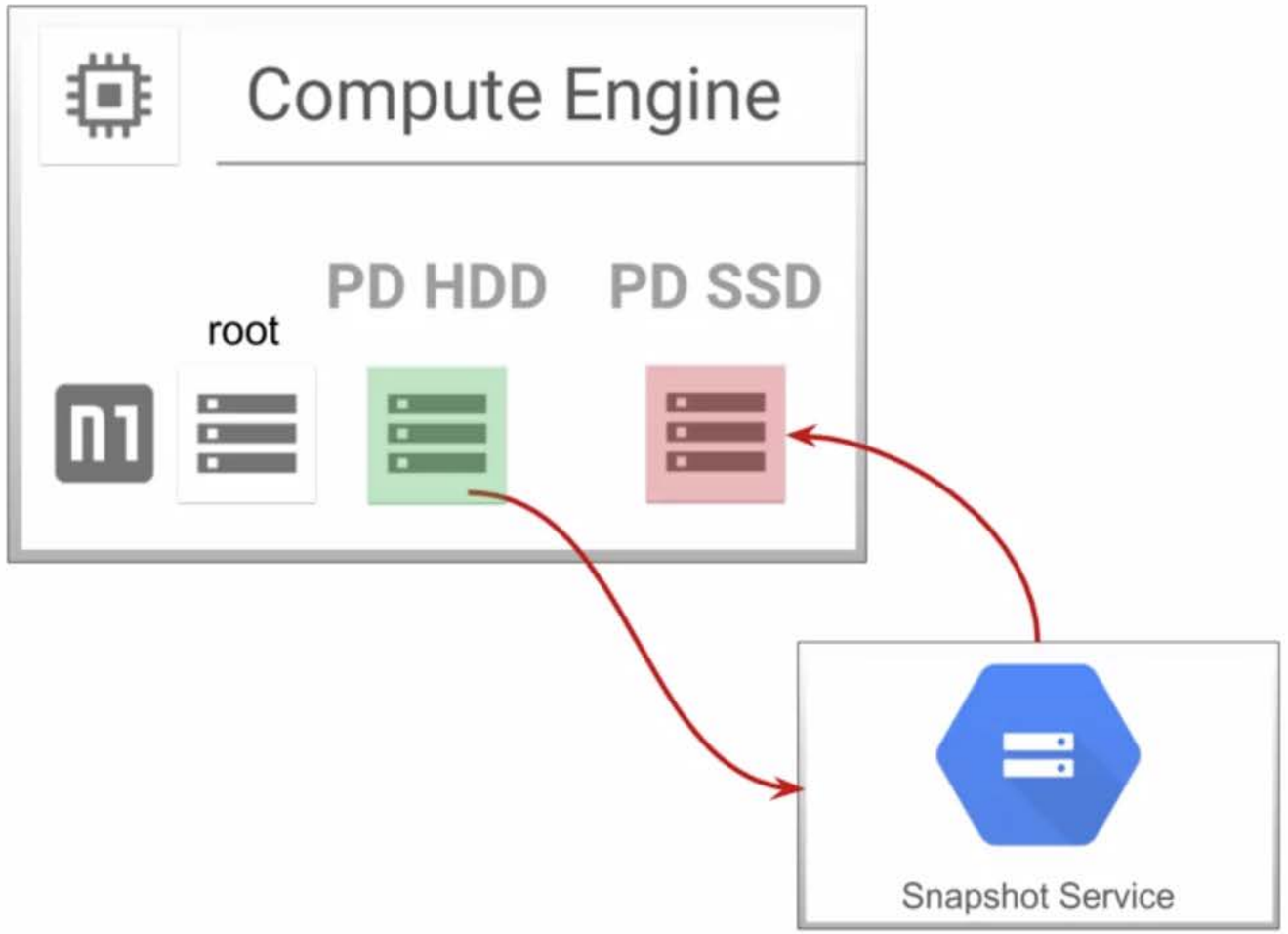
# Snapshot: Back up critical data



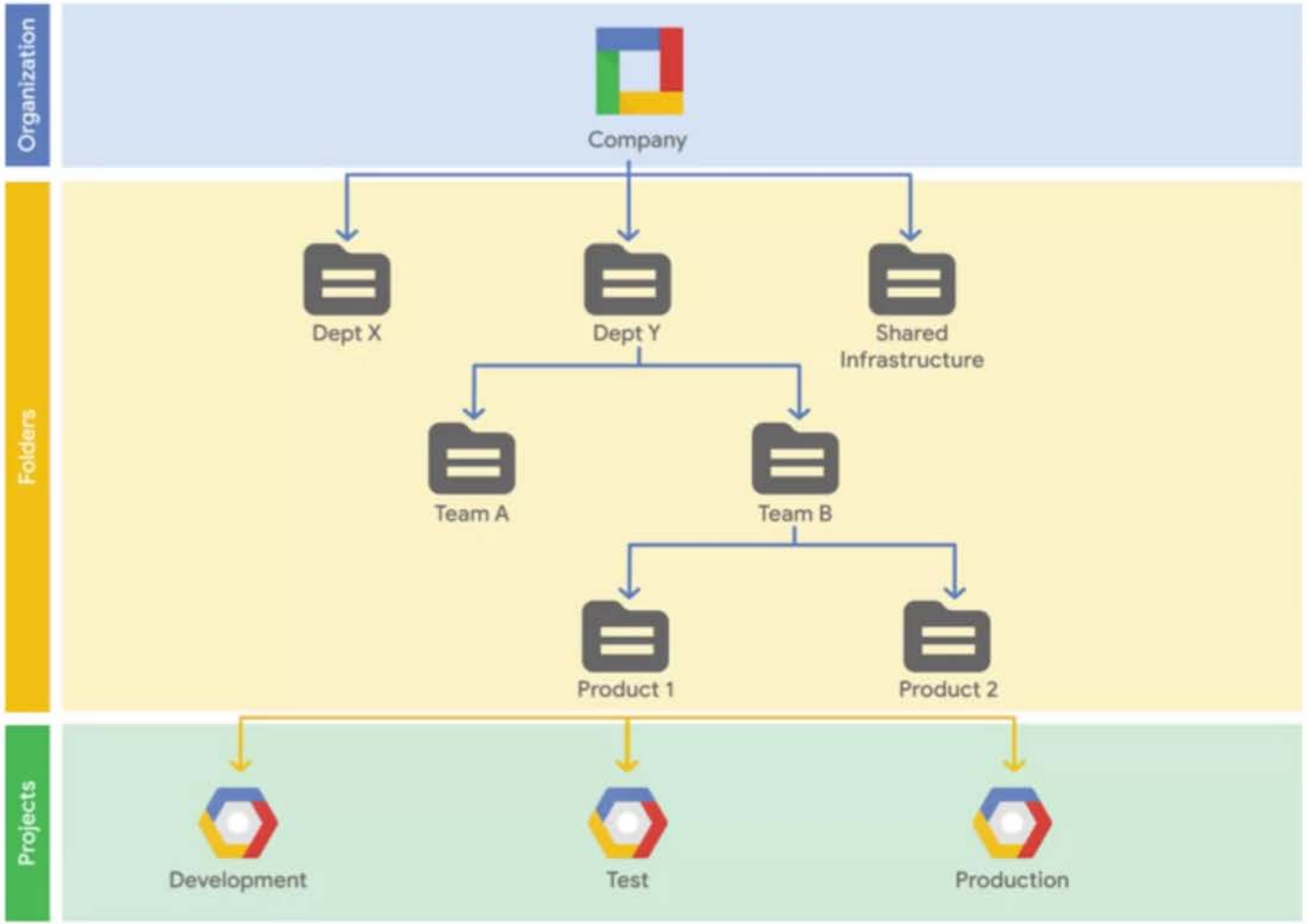
# Snapshot: Migrate data between zones



# Snapshot: Transfer to SSD to improve performance



# Folders




Additional grouping mechanism and isolation boundaries between projects:

- Different legal entities
- Departments
- Teams

Folders allow delegation of administration rights.

# Resource manager roles

## Organization

- **Admin**: Full control over all resources
- **Viewer**: View access to all resources

## Folder

- **Admin**: Full control over folders
- **Creator**: Browse hierarchy and create folders
- **Viewer**: View folders and projects below a resource

## Project

- **Creator**: Create new projects (automatic owner) and migrate new projects into organization
- **Deleter**: Delete projects

# IAM **primitive** roles offer fixed, coarse-grained levels of access

**Owner**

- Invite members
- Remove members
- Delete projects
- And...

**Editor**

- Deploy applications
- Modify code
- Configure services
- And...

**Viewer**

- Read-only access

**Billing Administrator**

- Manage billing
- Add and remove administrators

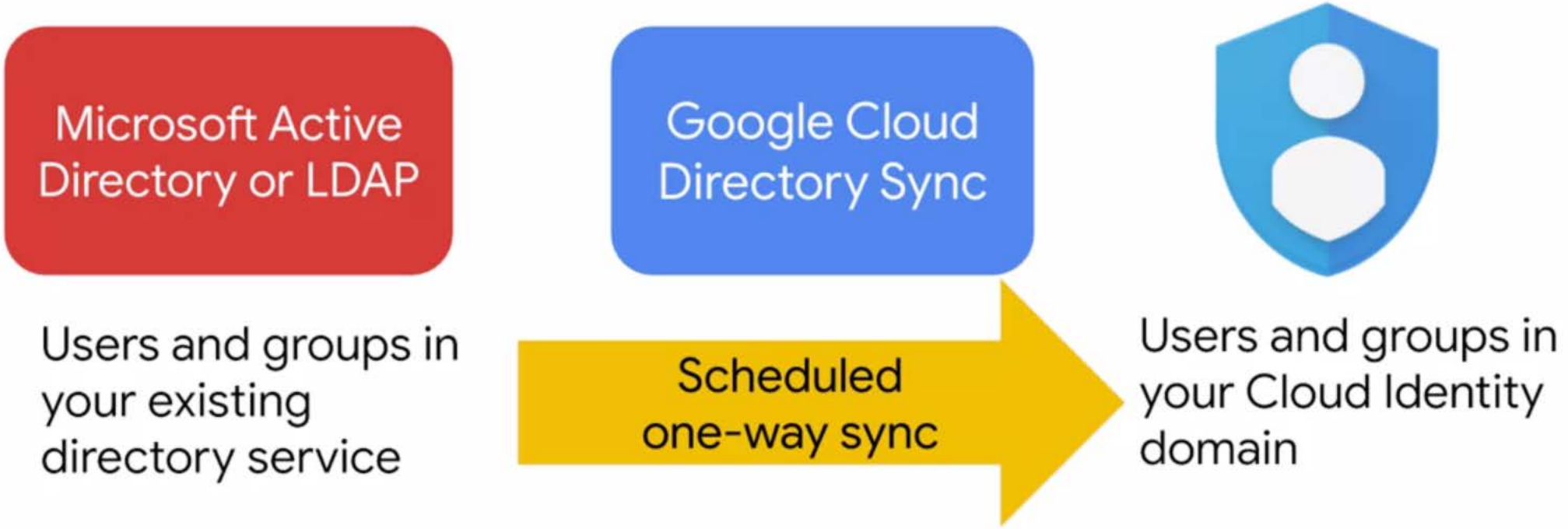
# Compute Engine IAM roles

| Role Title | Description |
| --- | --- |
| Compute Admin | Full control of all Compute Engine resources (compute.*) |
| Network Admin | Permissions to create, modify, and delete networking resources, except for firewall rules and SSL certificates |
| Storage Admin | Permissions to create, modify, and delete disks, images, and snapshots |

# Members


Note: You *cannot* use Cloud IAM to create or manage your users or groups.

# What if I already have a different corporate directory?



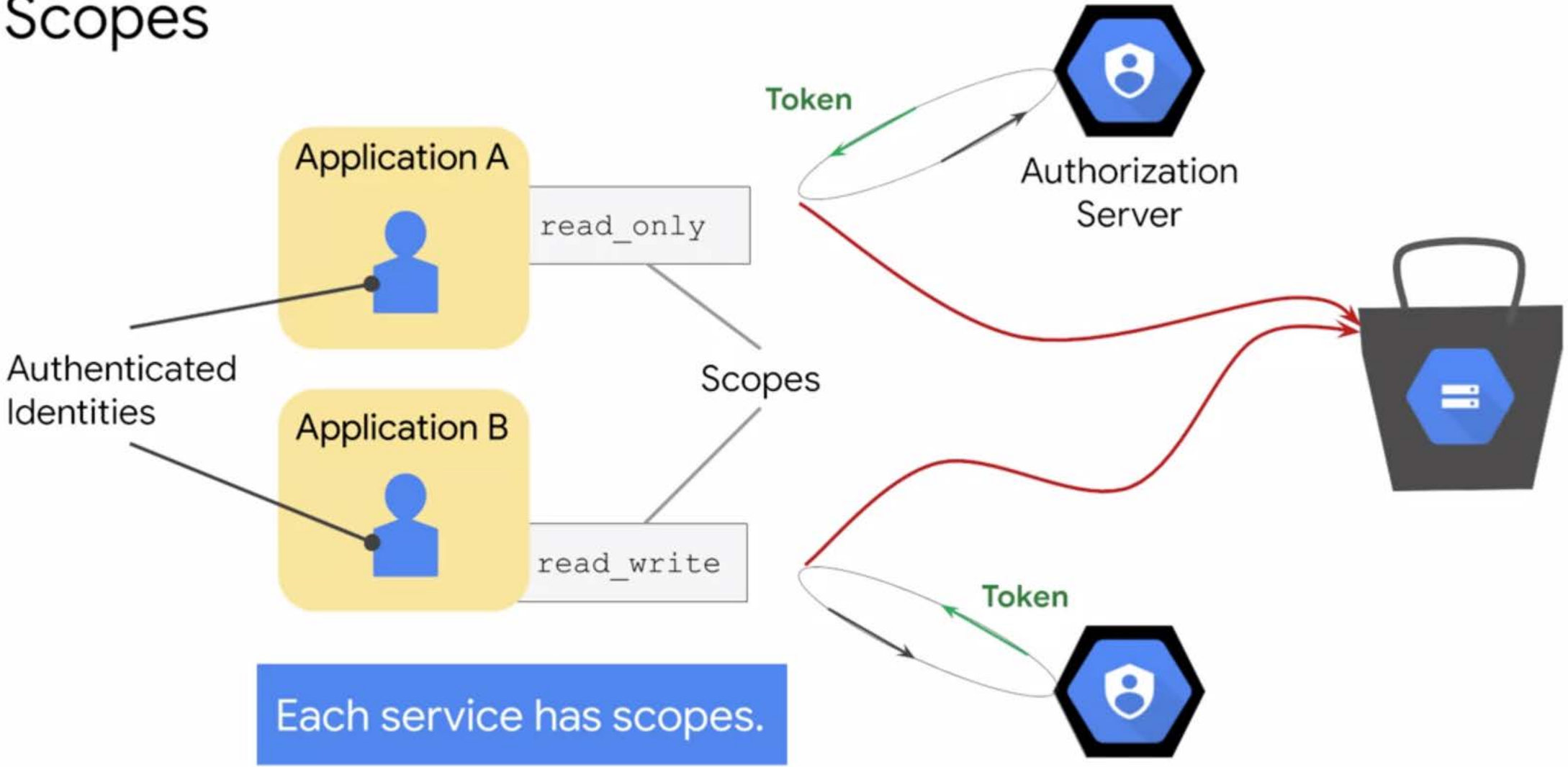
# Service accounts provide an identity for carrying out server-to-server interactions

- Programs running within Compute Engine instances can automatically acquire access tokens with credentials.
- Tokens are used to access any service API in your project and any other services that granted access to that service account.
- Service accounts are convenient when you're not accessing user data.

# Service accounts are identified by an email address

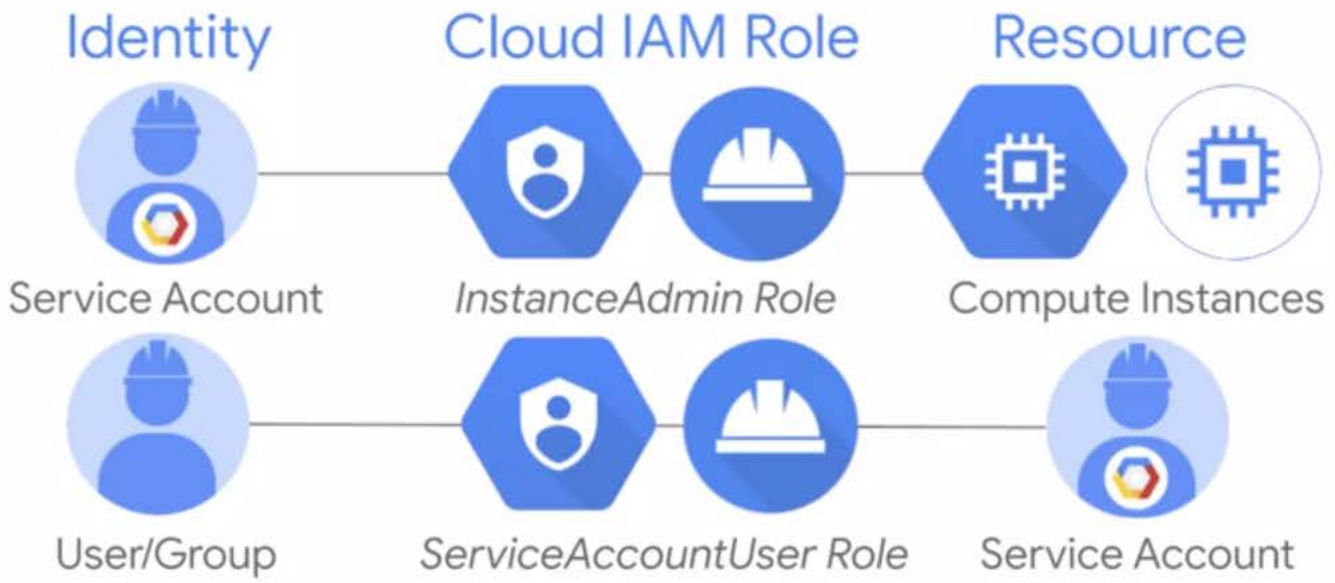
- `123845678986-compute@project.gserviceaccount.com`
- Three types of service accounts:
  - User-created (custom)
  - Built-in
    - Compute Engine and App Engine default service accounts
  - Google APIs service account
    - Runs internal Google processes on your behalf.

# Scopes


Token
Authorization Server
Application A
read_only
Authenticated Identities
Scopes
Application B
read_write
Token
Each service has scopes.

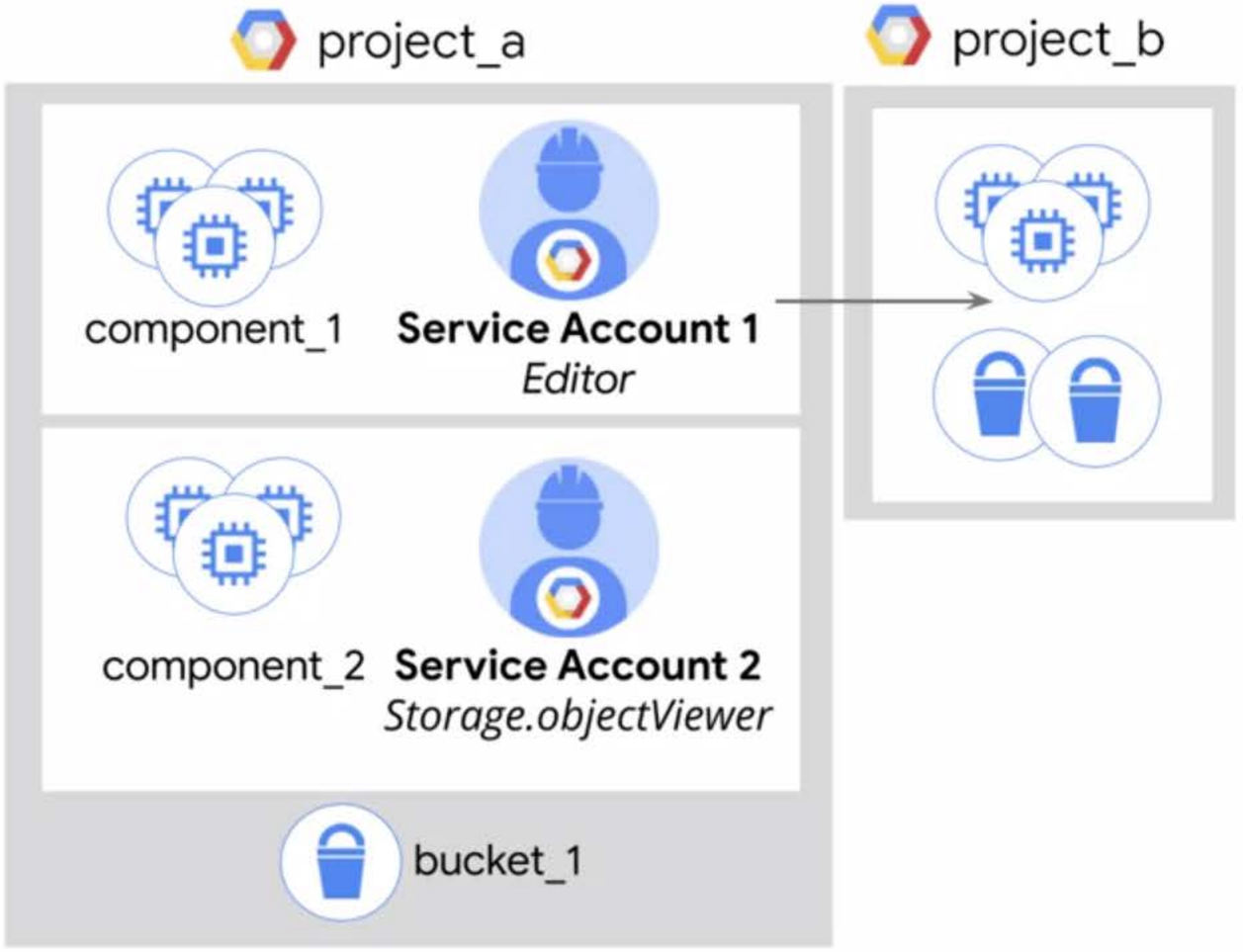
# Service account permissions

- Default service accounts: primitive and predefined roles
- User-created service accounts: predefined roles
- Roles for service accounts can be assigned to groups or users



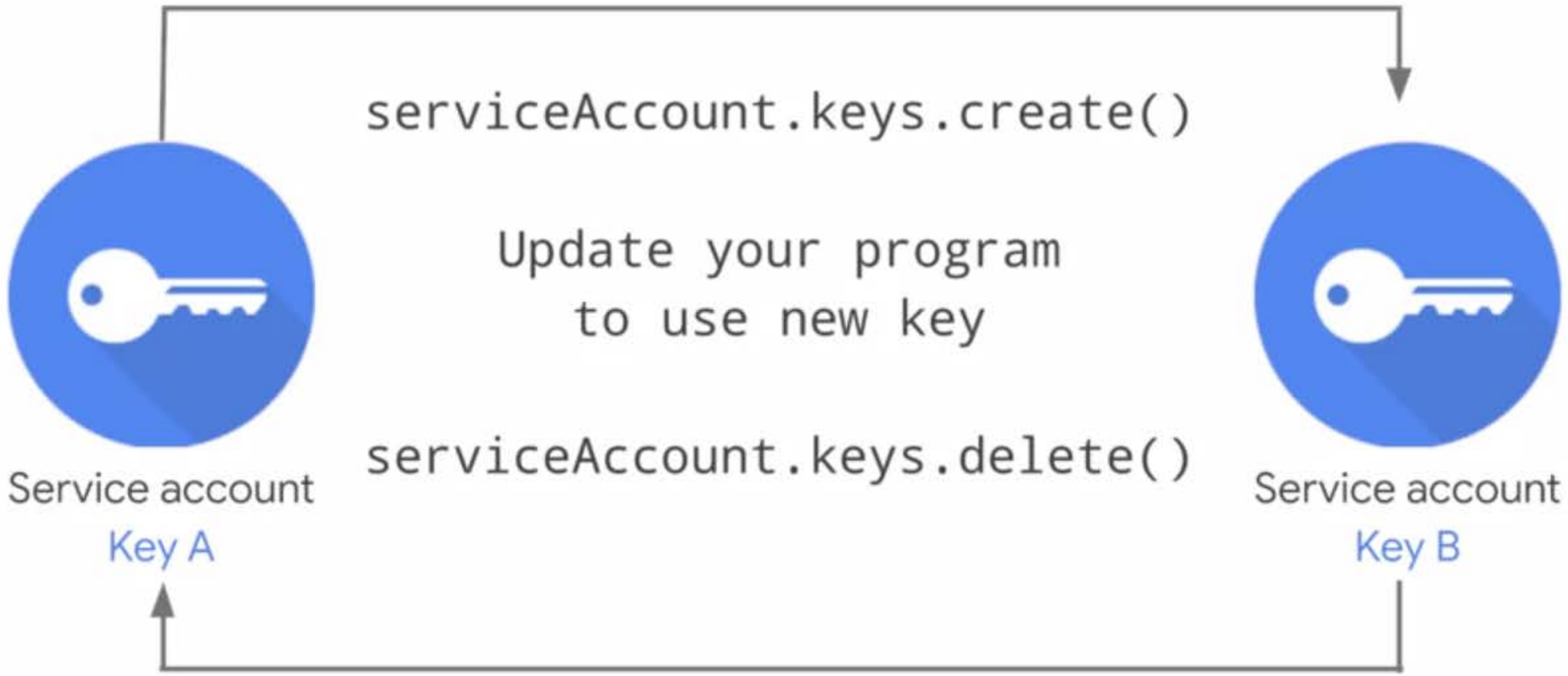
# Example: Service accounts and Cloud IAM

- VMs running component_1 are granted Editor access to project_b using *Service Account 1*.
- VMs running component_2 are granted objectViewer access to bucket_1 using *Service Account 2*.
- Service account permissions can be changed without re-created VMs.



# Service accounts authenticate with keys

- GCP-managed: Cannot be downloaded, and are automatically rotated
- User-managed: Create, manage, and rotate yourself

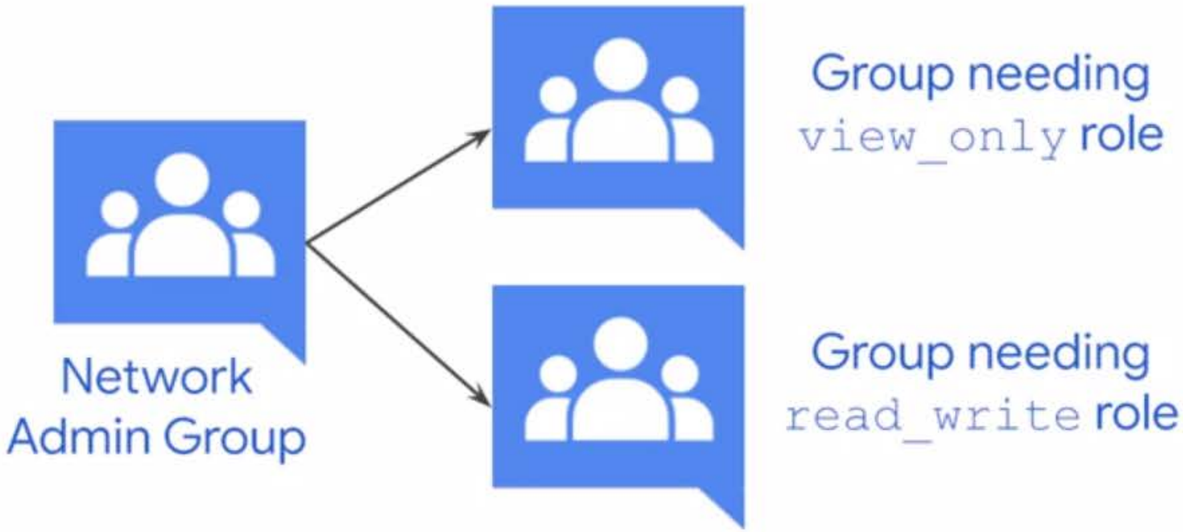


# Leverage and understand the resource hierarchy

- Use projects to group resources that share the same trust boundary.
- Check the policy granted on each resource and make sure you understand the inheritance.
- Use “principles of least privilege” when granting roles.
- Audit policies in Cloud audit logs: `setiampolicy`.
- Audit membership of groups used in policies.

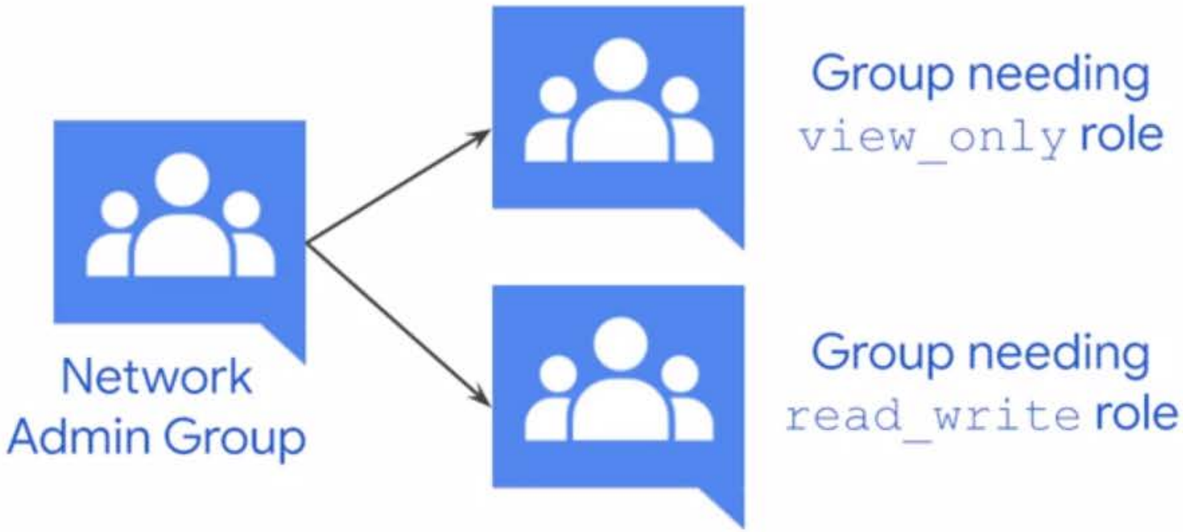
# Grant roles to Google groups instead of individuals

- Update group membership instead of changing Cloud IAM policy.
- Audit membership of groups used in policies.
- Control the ownership of the Google group used in Cloud IAM policies.



# Grant roles to Google groups instead of individuals

- Update group membership instead of changing Cloud IAM policy.
- Audit membership of groups used in policies.
- Control the ownership of the Google group used in Cloud IAM policies.

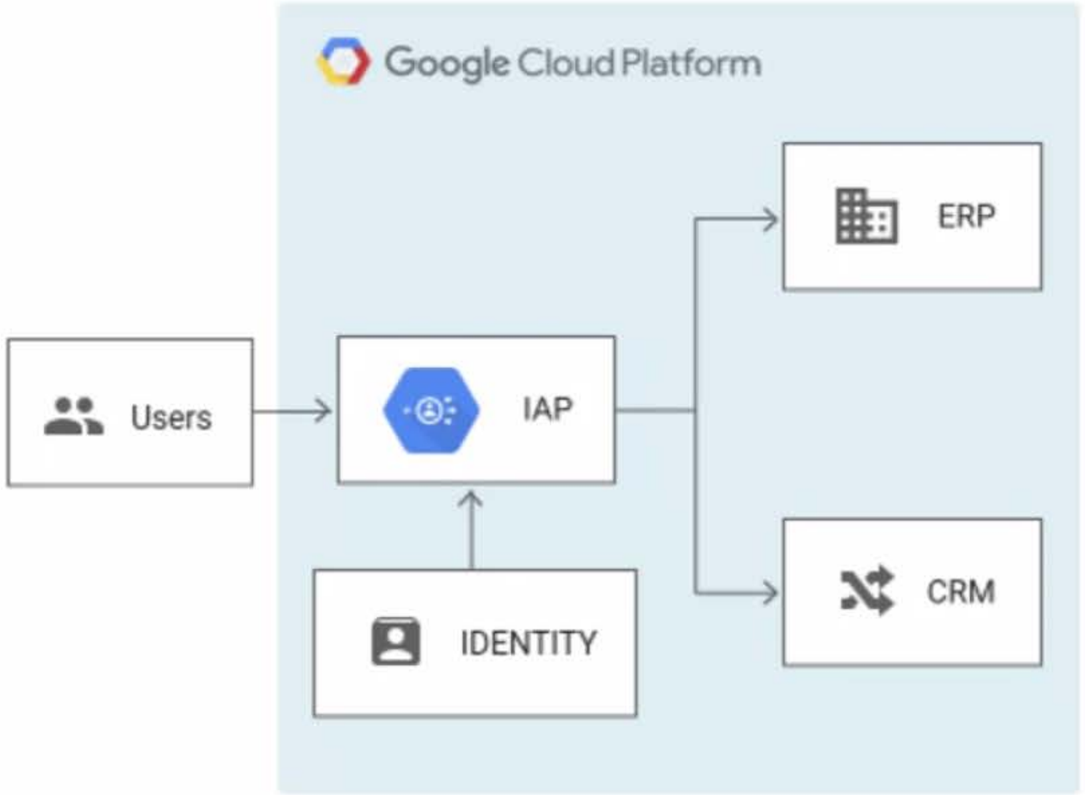


# Cloud Identity-Aware Proxy (Cloud IAP)

Enforce access control policies for applications and resources:

- Identity-based access control
- Central authorization layer for applications accessed by HTTPS

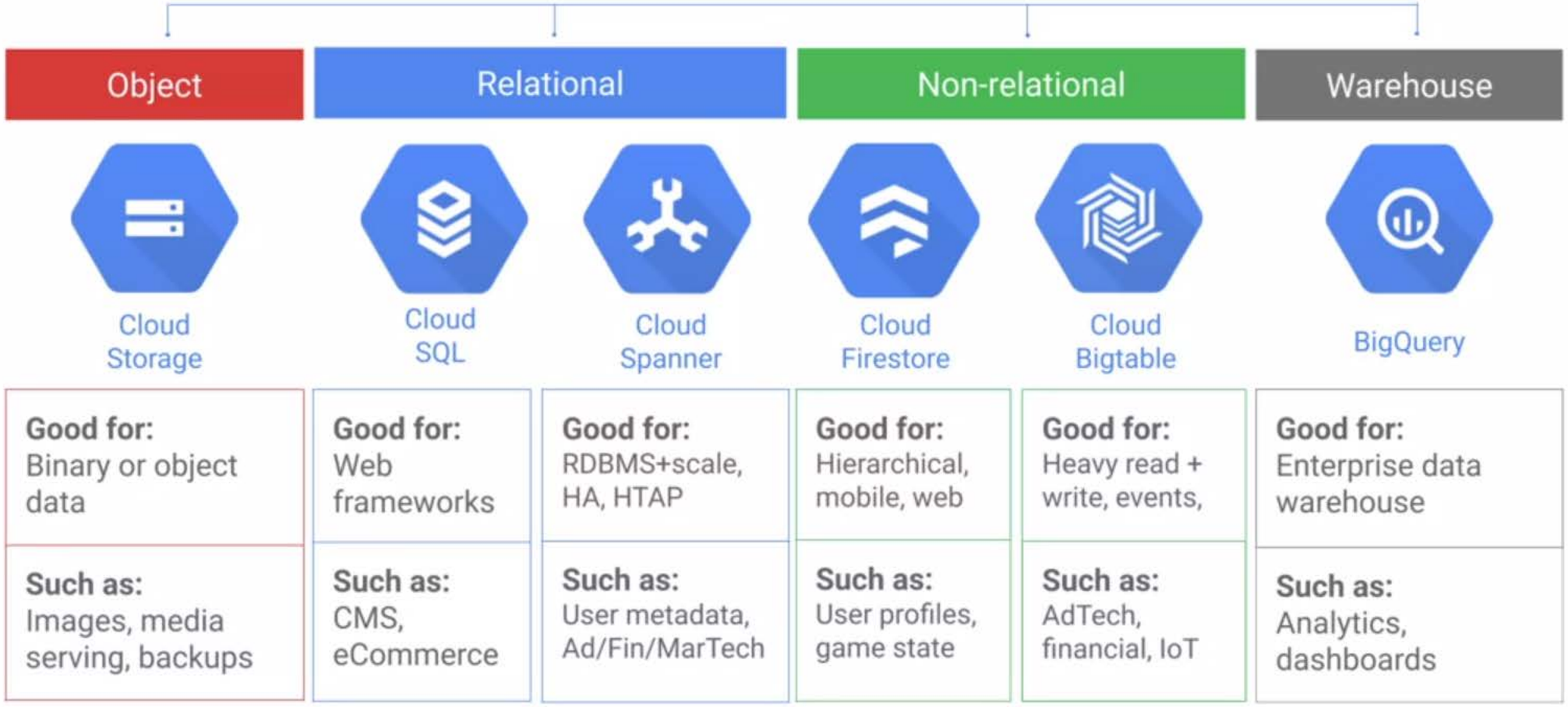
Cloud IAM policy is applied after authentication.



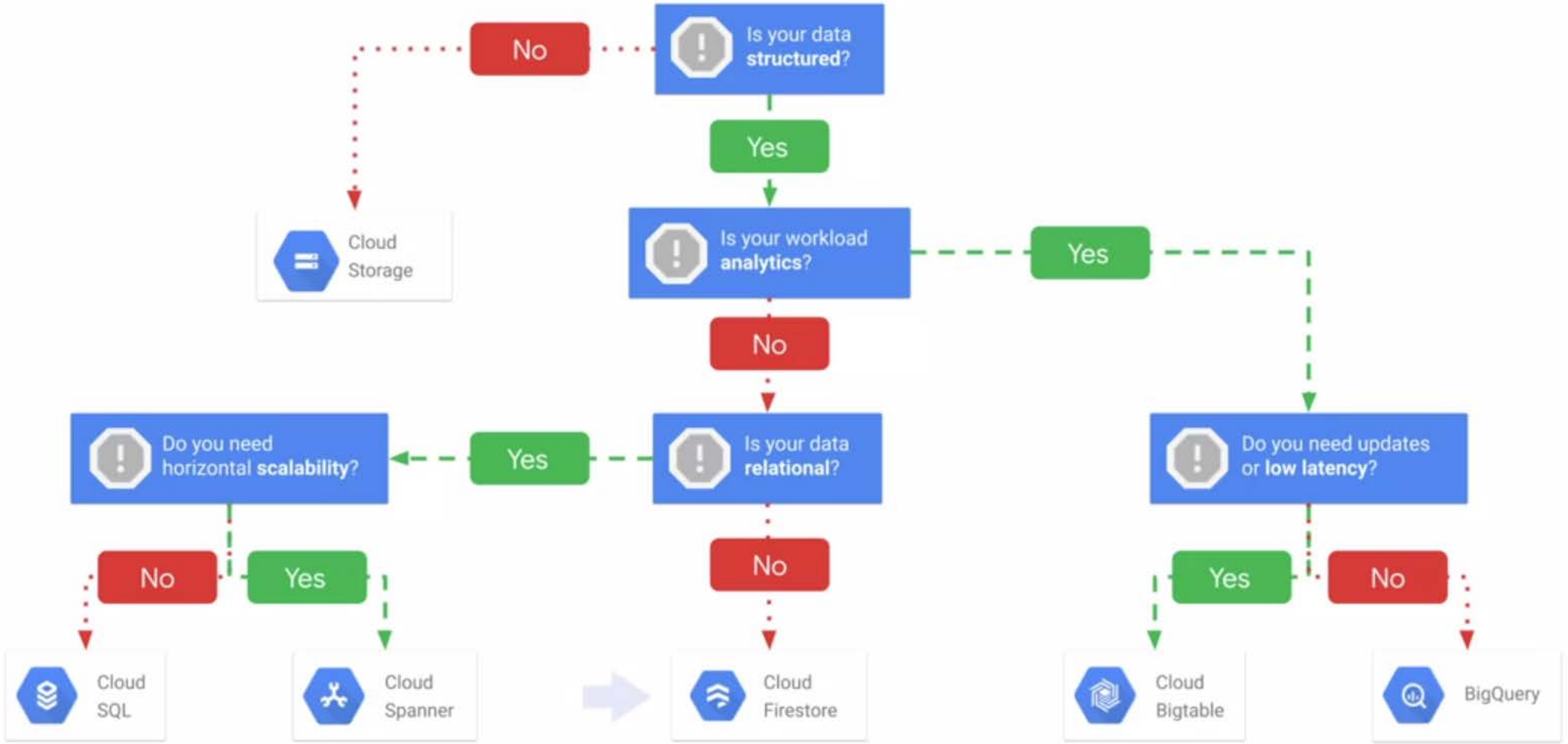
# Service accounts

- Be very careful granting `serviceAccountUser` role.
- When you create a service account, give it a display name that clearly identifies its purpose.
- Establish a naming convention for service accounts.
- Establish key rotation policies and methods.
- Audit with `serviceAccount.keys.list()` method.

# Storage and database services



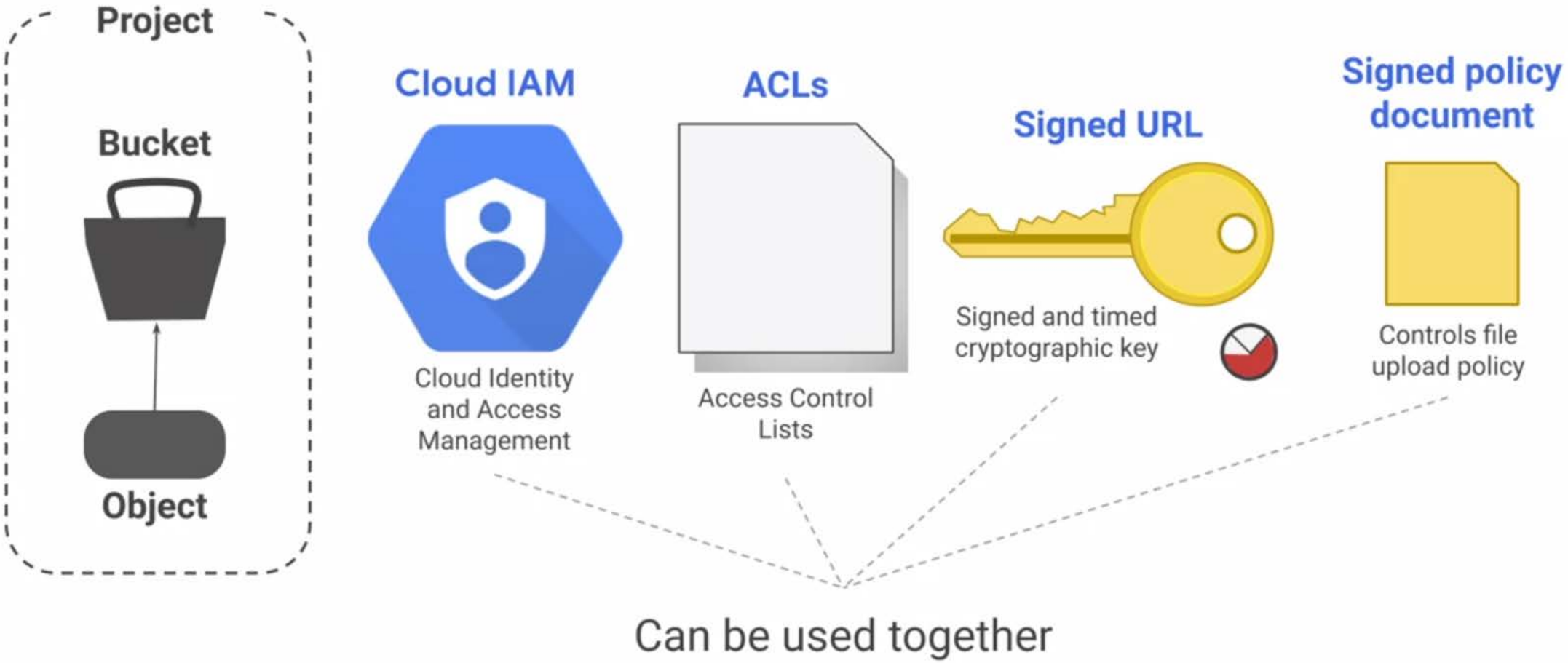
# Storage and database decision chart



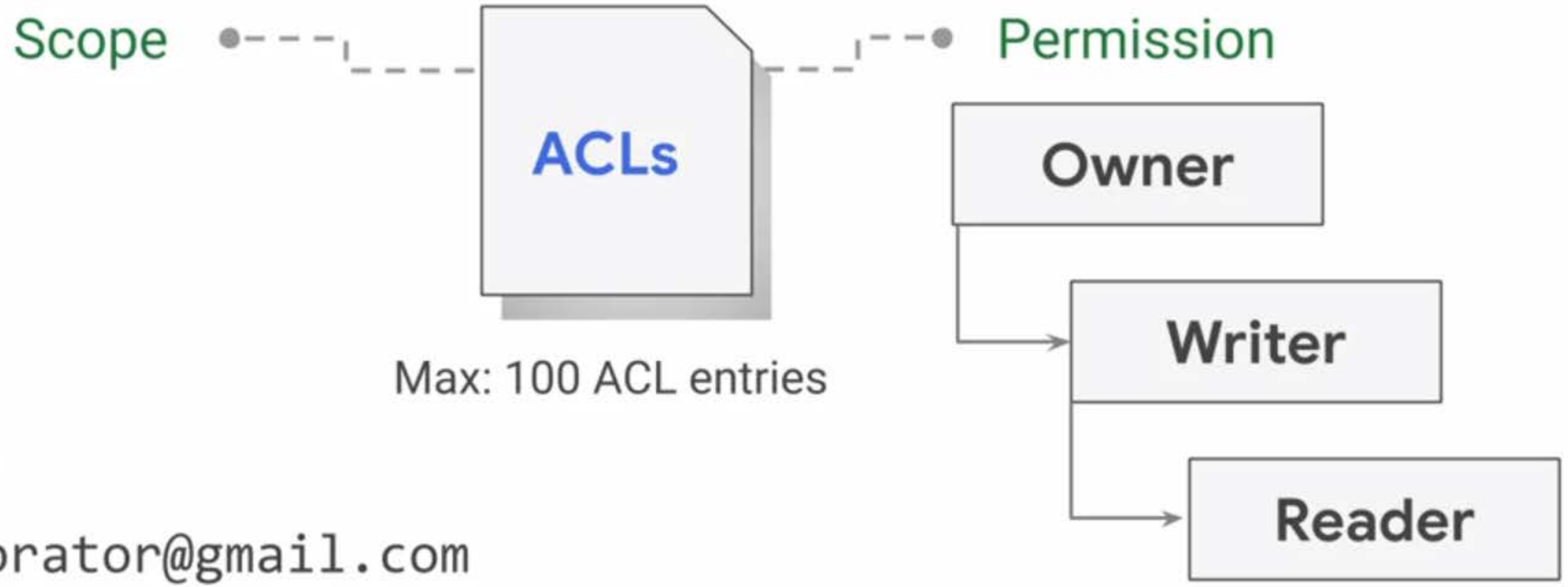
# Overview of storage classes

| | Standard | Nearline | Coldline | Archive |
|---|---|---|---|---|
| Use case | "Hot" data and/or stored for only brief periods of time like data-intensive computations | Infrequently accessed data like data backup, long-tail multimedia content, and data archiving | Infrequently accessed data that you read or modify at most once a quarter | Data archiving, online backup, and disaster recovery |
| Minimum storage duration | None | 30 days | 90 days | 365 days |
| Retrieval cost | None | $0.01 per GB | $0.02 per GB | $0.05 per GB |
| Availability SLA | 99.95% (multi/dual)<br>99.90% (region) | 99.90% (multi/dual)<br>99.00% (region) | | None |
| Durability | 99.999999999% | | | |

# Access control



# Access control lists (ACLs)




Examples:

- `collaborator@gmail.com`
- `allUsers`
- `allAuthenticatedUsers`

# Cloud Storage features

- Customer-supplied encryption key (CSEK)
  - Use your own key instead of Google-managed keys
- Object Lifecycle Management
  - Automatically delete or archive objects
- Object Versioning
  - Maintain multiple versions of objects
- Directory synchronization
  - Synchronizes a VM directory with a bucket
- Object change notification
- Data import
- Strong consistency

# Object Versioning supports the retrieval of objects that are deleted or overwritten


- Objects are immutable.
- Object Versioning:
  - Maintain a history of modifications of objects.
  - List archived versions of an object, restore an object to an older state, or delete a version.

# Object Lifecycle Management policies specify actions to be performed on objects that meet certain rules

- Examples:
  - Downgrade storage class on objects older than a year.
  - Delete objects created before a specific date.
  - Keep only the 3 most recent versions of an object.
- Object inspection occurs in asynchronous batches.
- Changes can take 24 hours to apply.

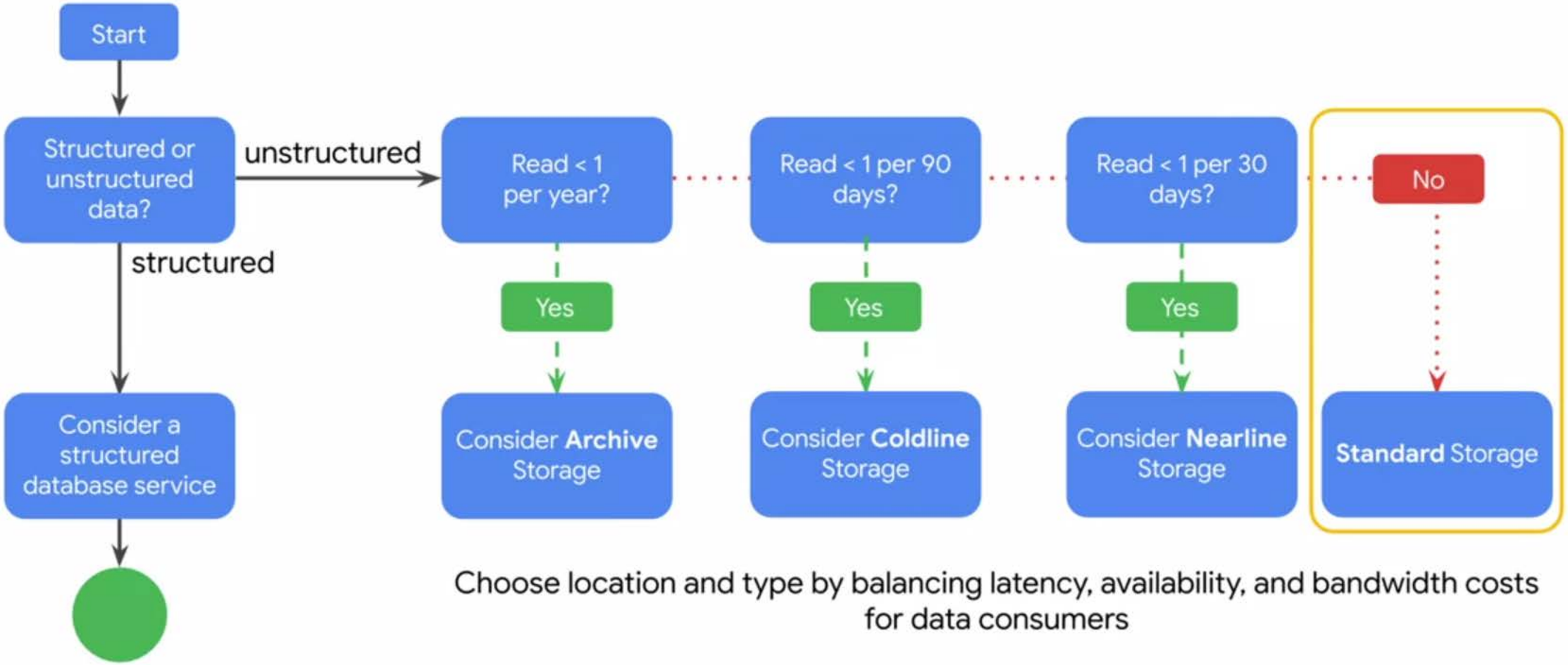
# Data import services

- **Transfer Appliance:** Rack, capture and then ship your data to Google Cloud.
- **Storage Transfer Service:** Import online data (another bucket, an S3 bucket, or web source).
- **Offline Media Import:** Third-party provider uploads the data from physical media.


# Cloud Storage provides strong global consistency

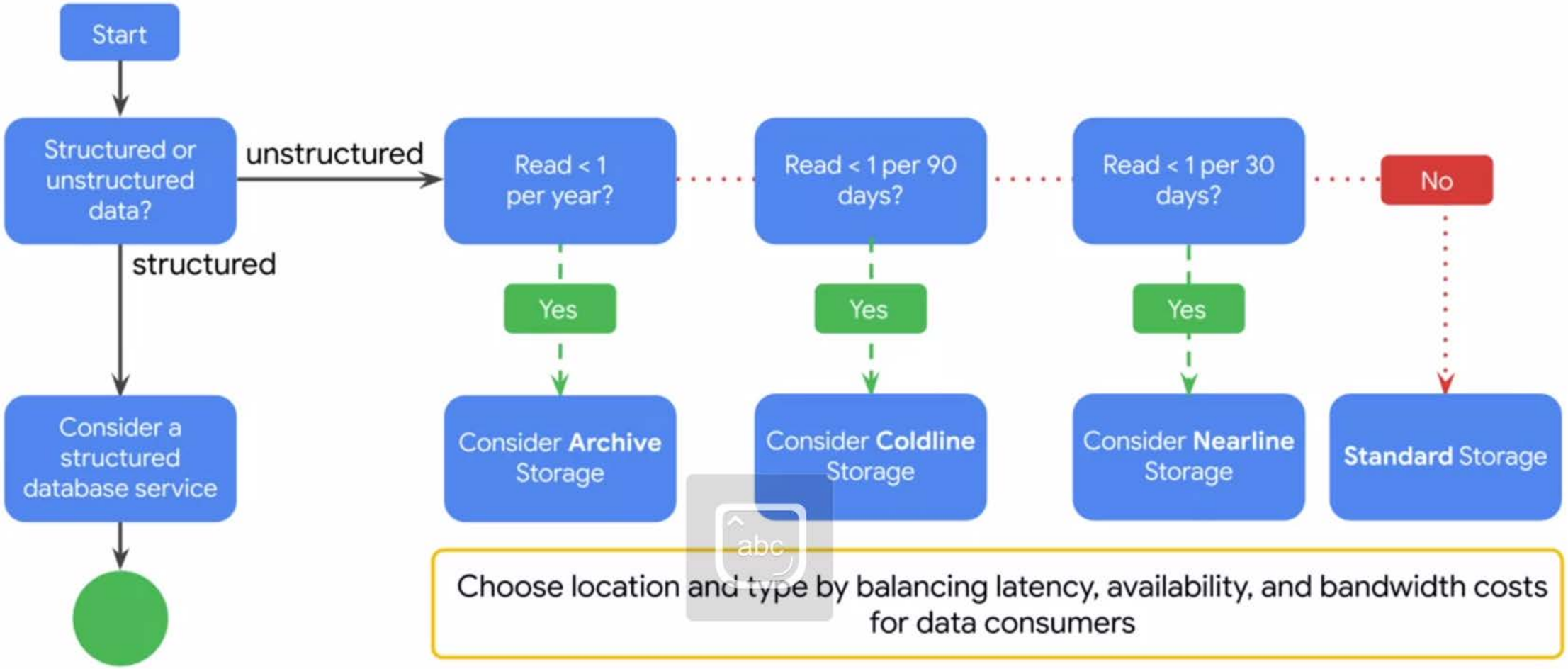
- Read-after-write
- Read-after-metadata-update
- Read-after-delete
- Bucket listing
- Object listing

# Choosing a storage class



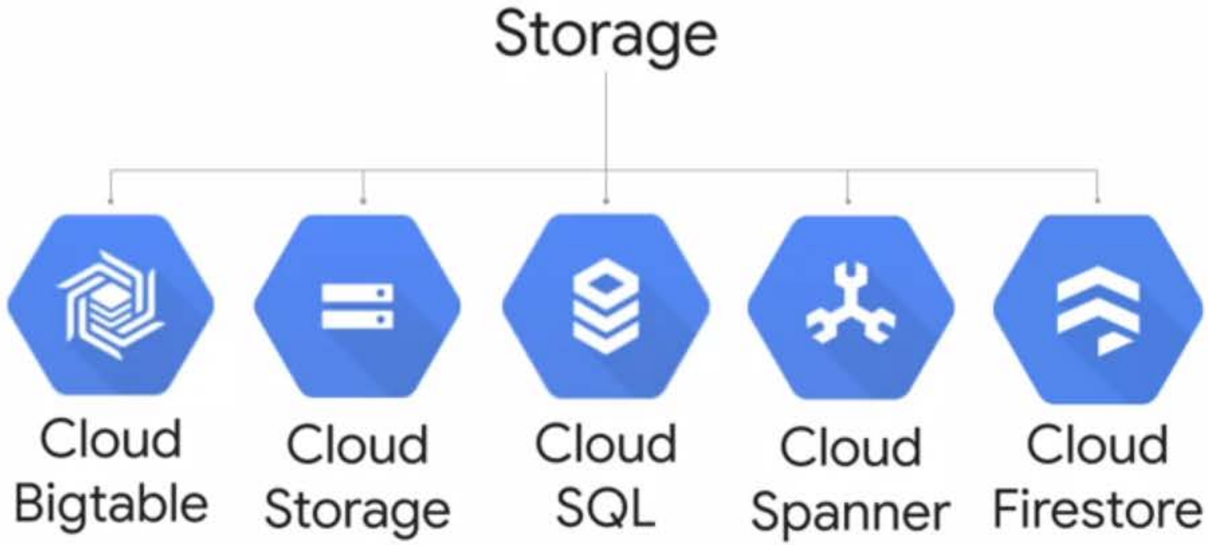

Choose location and type by balancing latency, availability, and bandwidth costs for data consumers

# Choosing a storage class




Choose location and type by balancing latency, availability, and bandwidth costs for data consumers

# Build your own database solution or use a managed service

# Cloud SQL is a fully managed database service (MySQL, PostgreSQL, or Microsoft SQL Server)

- Patches and updates automatically applied
- You administer MySQL users
- Cloud SQL supports many clients
  - `gcloud sql`
  - App Engine, G Suite scripts
  - Applications and tools
    - SQL Workbench, Toad
    - External applications using standard MySQL drivers

Cloud SQL

# Cloud SQL instance

**Performance:**

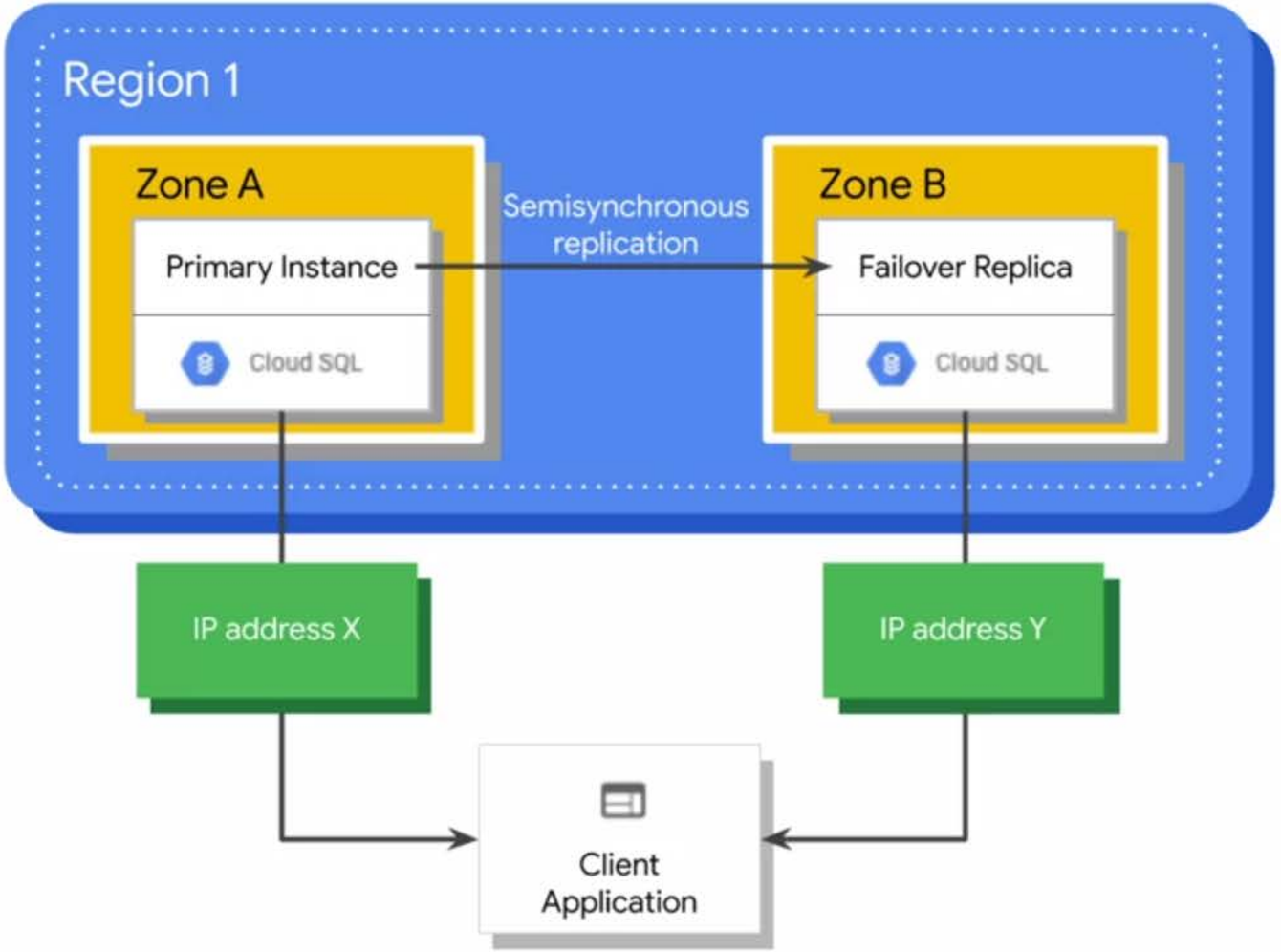
- 30 TB of storage
- 40,000 IOPS
- 416 GB of RAM
- Scale out with read replicas

Choice:

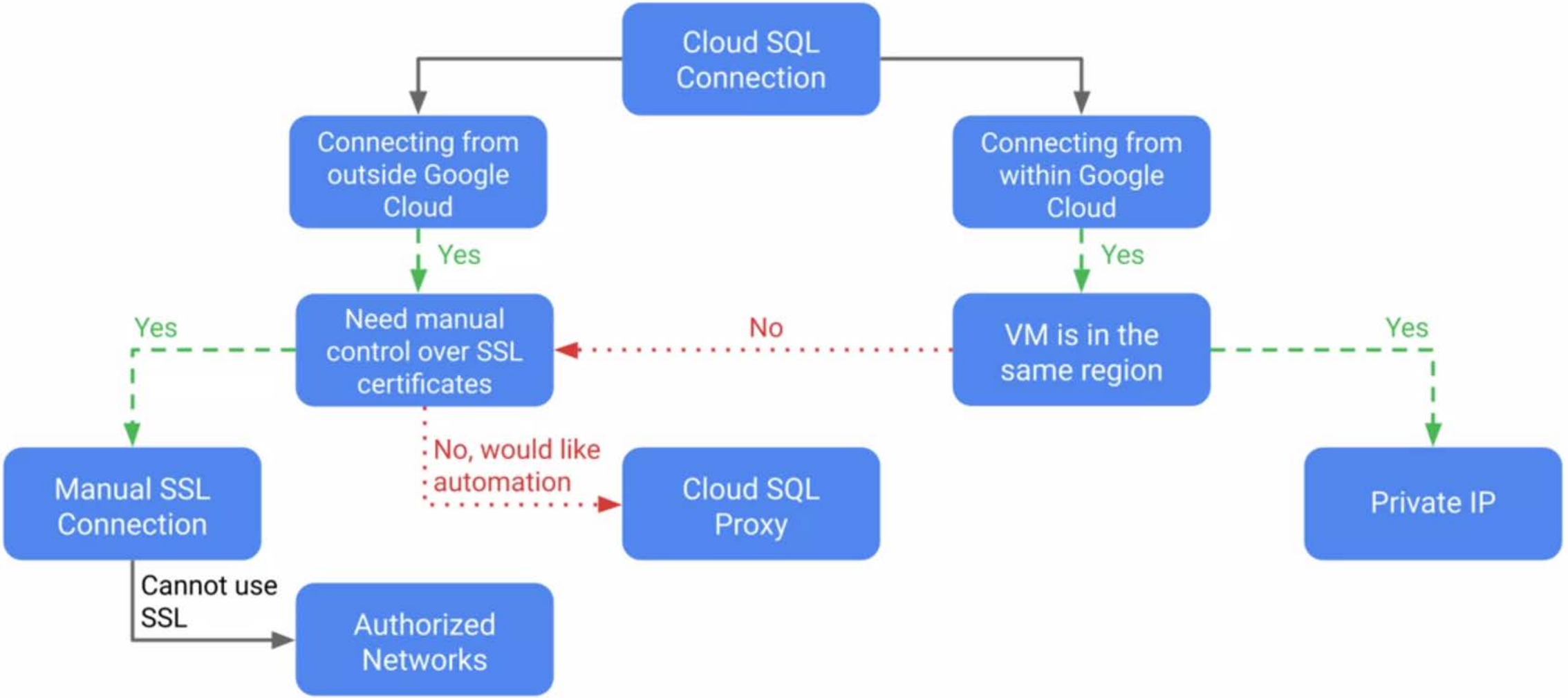
- MySQL 5.6, 5.7 (default), or 8.0
- PostgreSQL 9.6, 10, 11 or 12 (default)
- Microsoft SQL Server 2017

# Cloud SQL services

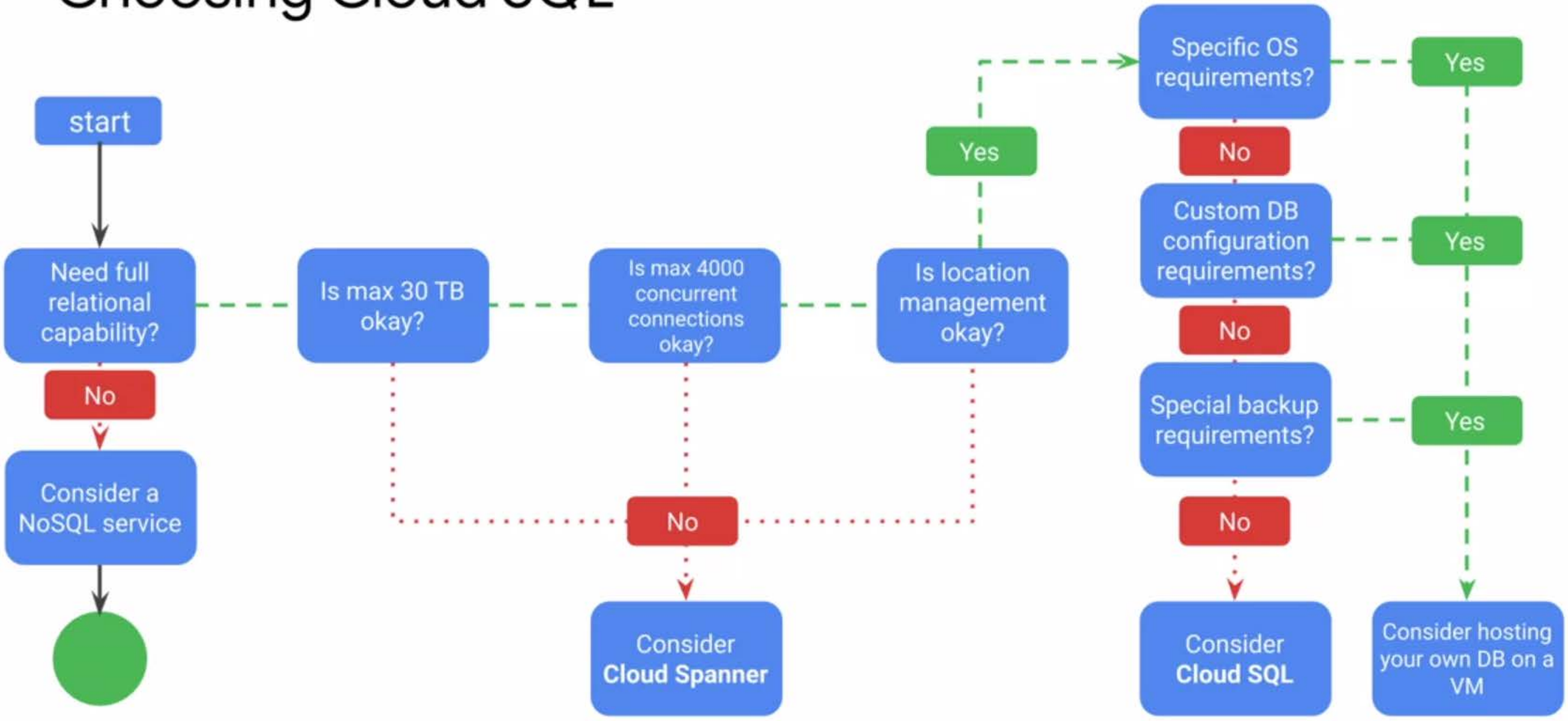
- Replica services
- Backup service
- Import/export
- Scaling
  - Up: Machine capacity
  - Out: Replicas



# Connecting to a Cloud SQL instance



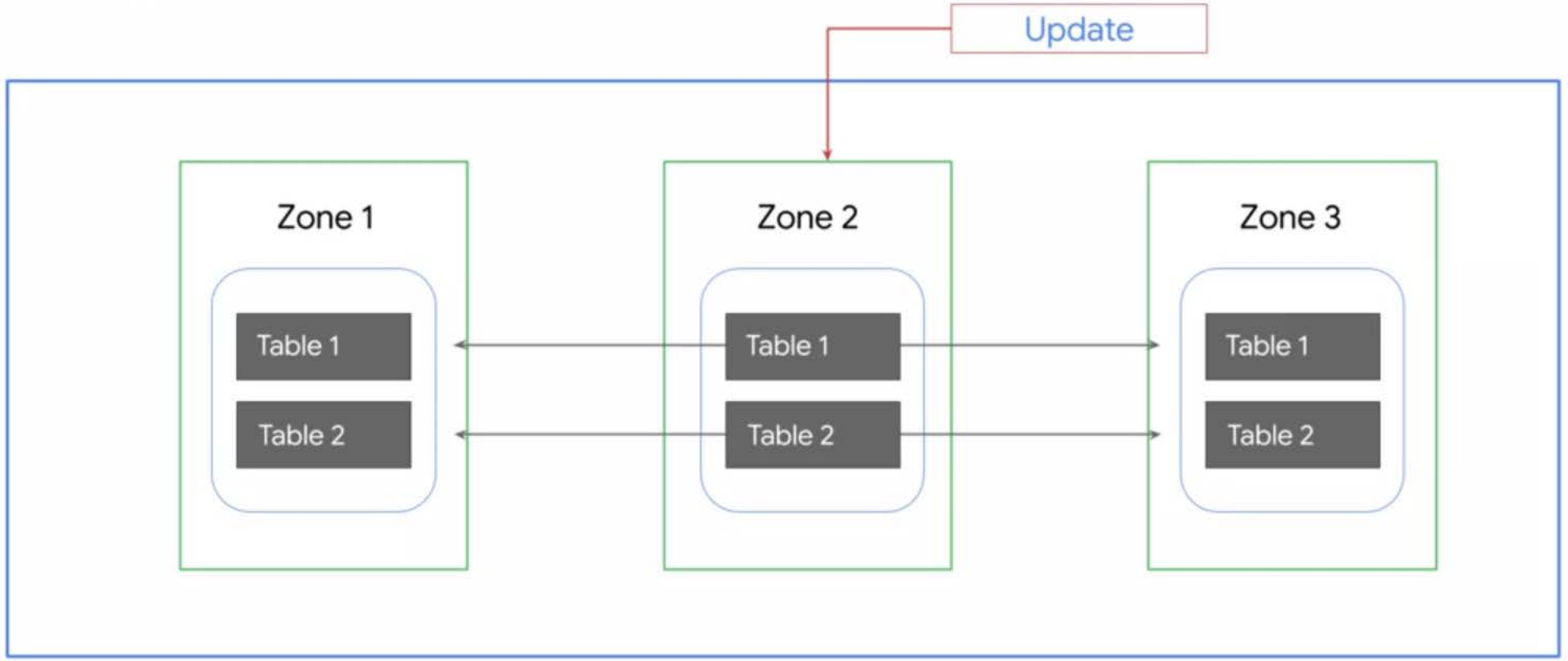
# Choosing Cloud SQL



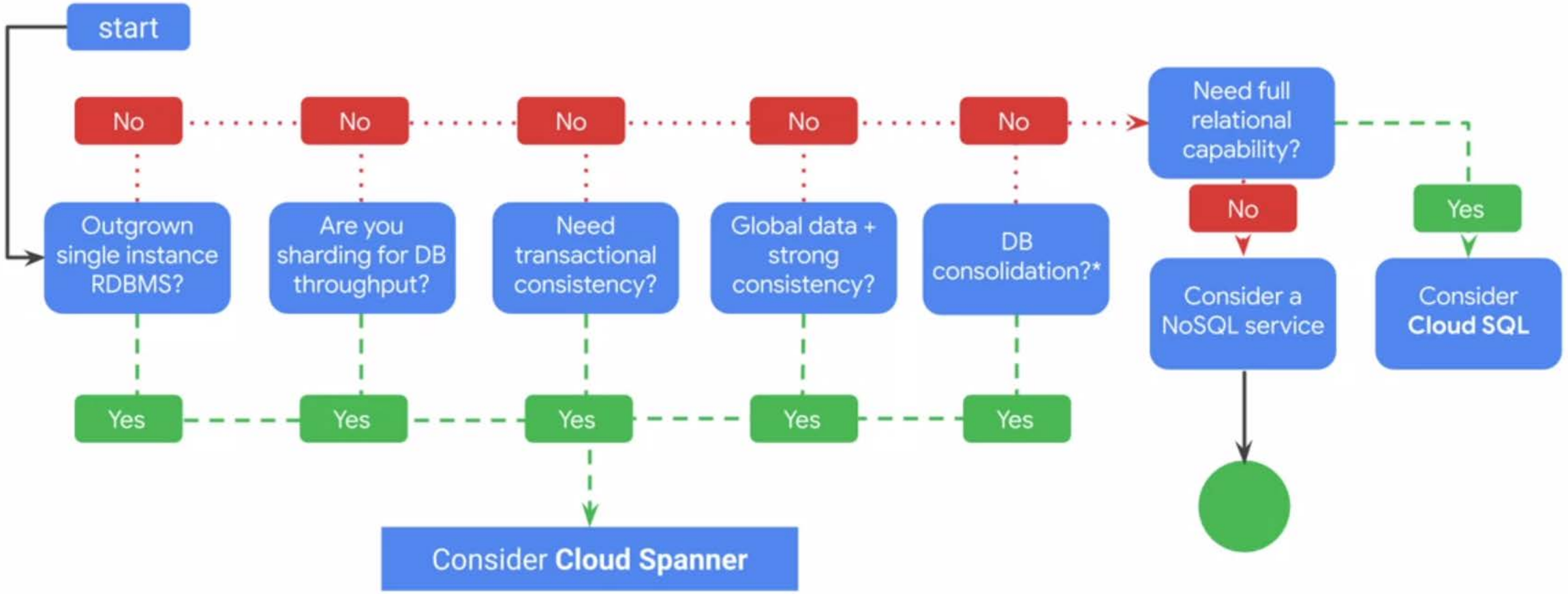
# Characteristics

| | Cloud Spanner | Relational DB | Non-Relational DB |
|---|---|---|---|
| Schema | ✓ Yes | ✓ Yes | ✗ No |
| SQL | ✓ Yes | ✓ Yes | ✗ No |
| Consistency | ✓ Strong | ✓ Strong | ✗ Eventual |
| Availability | ✓ High | ✗ Failover | ✓ High |
| Scalability | ✓ Horizontal | ✗ Vertical | ✓ Horizontal |
| Replication | ✓ Automatic | ↻ Configurable | ↻ Configurable |

# Data replication is synchronized across zones using Google's global fiber network



# Choosing Cloud Spanner



# Cloud Firestore is a NoSQL document database

- Simplifies storing, syncing, and querying data
- Mobile, web, and IoT apps at global scale
- Live synchronization and offline support
- Security features
- ACID transactions
- Multi-region replication
- Powerful query engine

Cloud Firestore

# Cloud Firestore is the next generation of Cloud Datastore

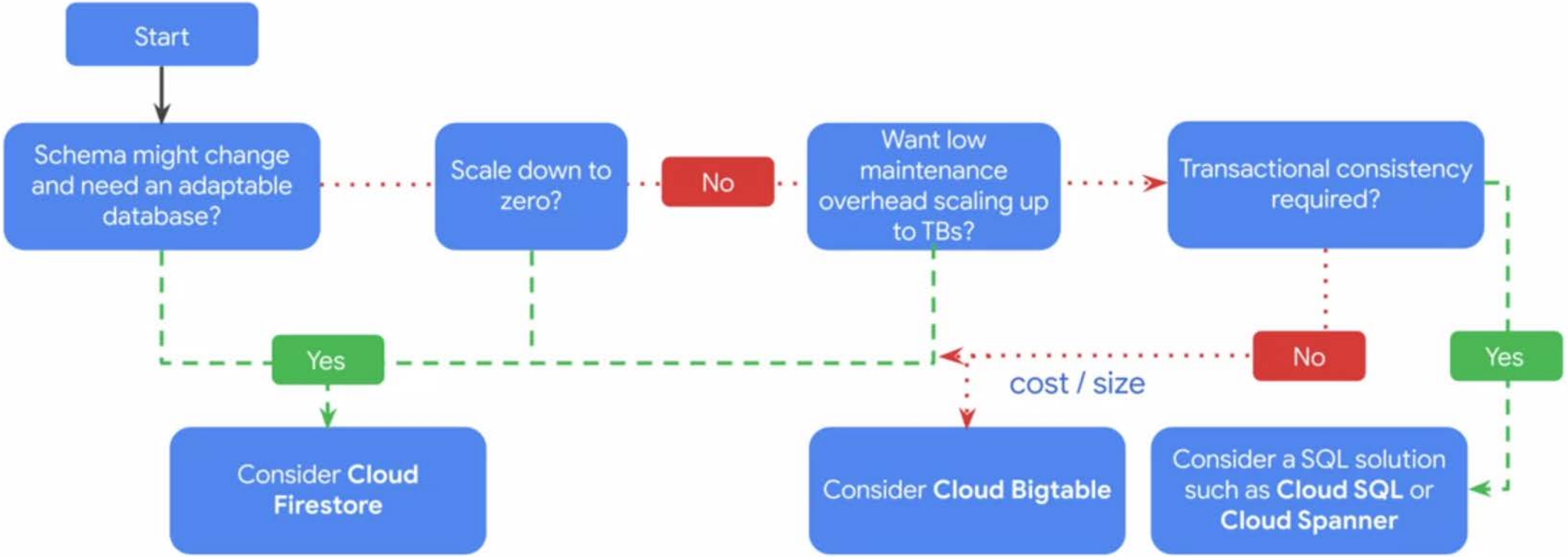
Datastore mode (new server projects):

- Compatible with Datastore applications
- Strong consistency
- No entity group limits

Native mode (new mobile and web apps):

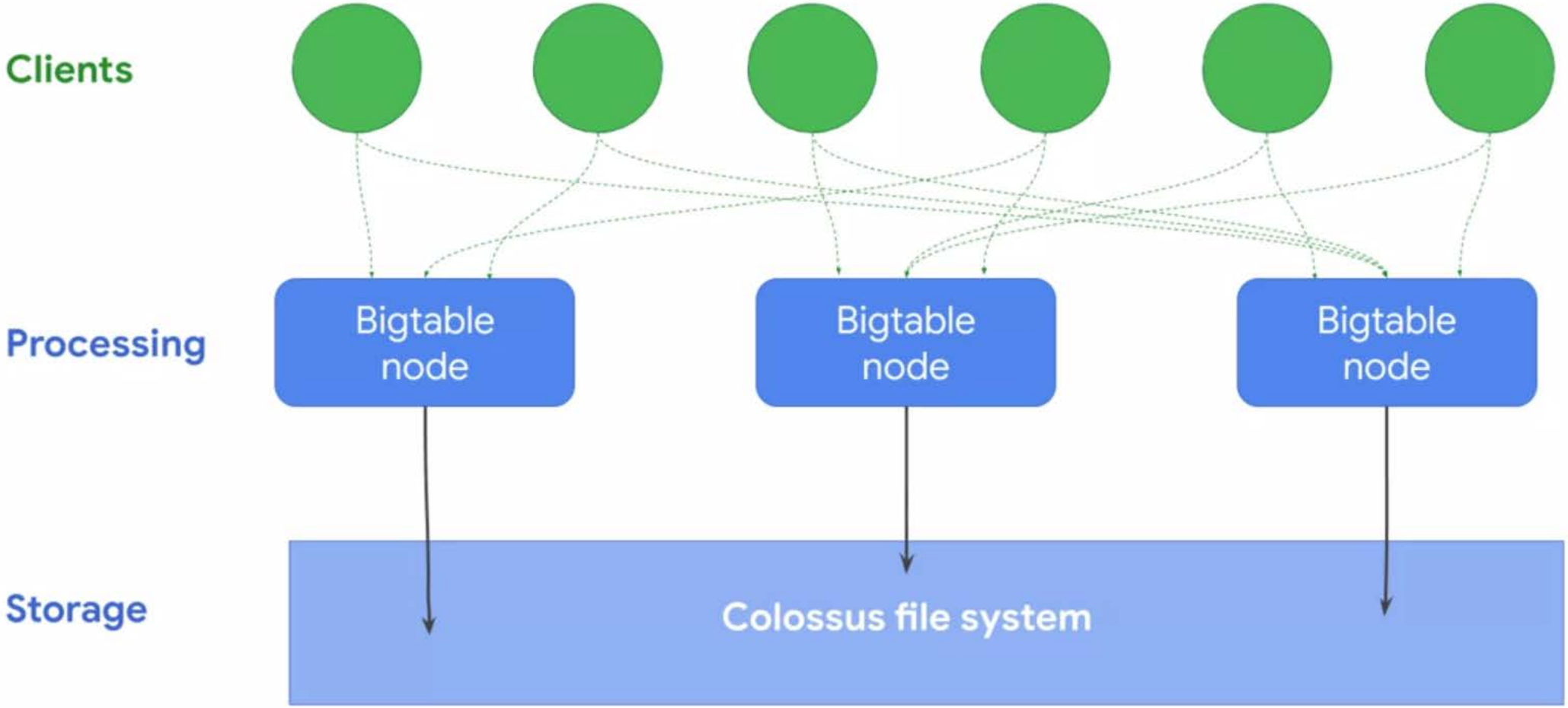
- Strongly consistent storage layer
- Collection and document data model
- Real-time updates
- Mobile and Web client libraries

# Choosing Cloud Firestore

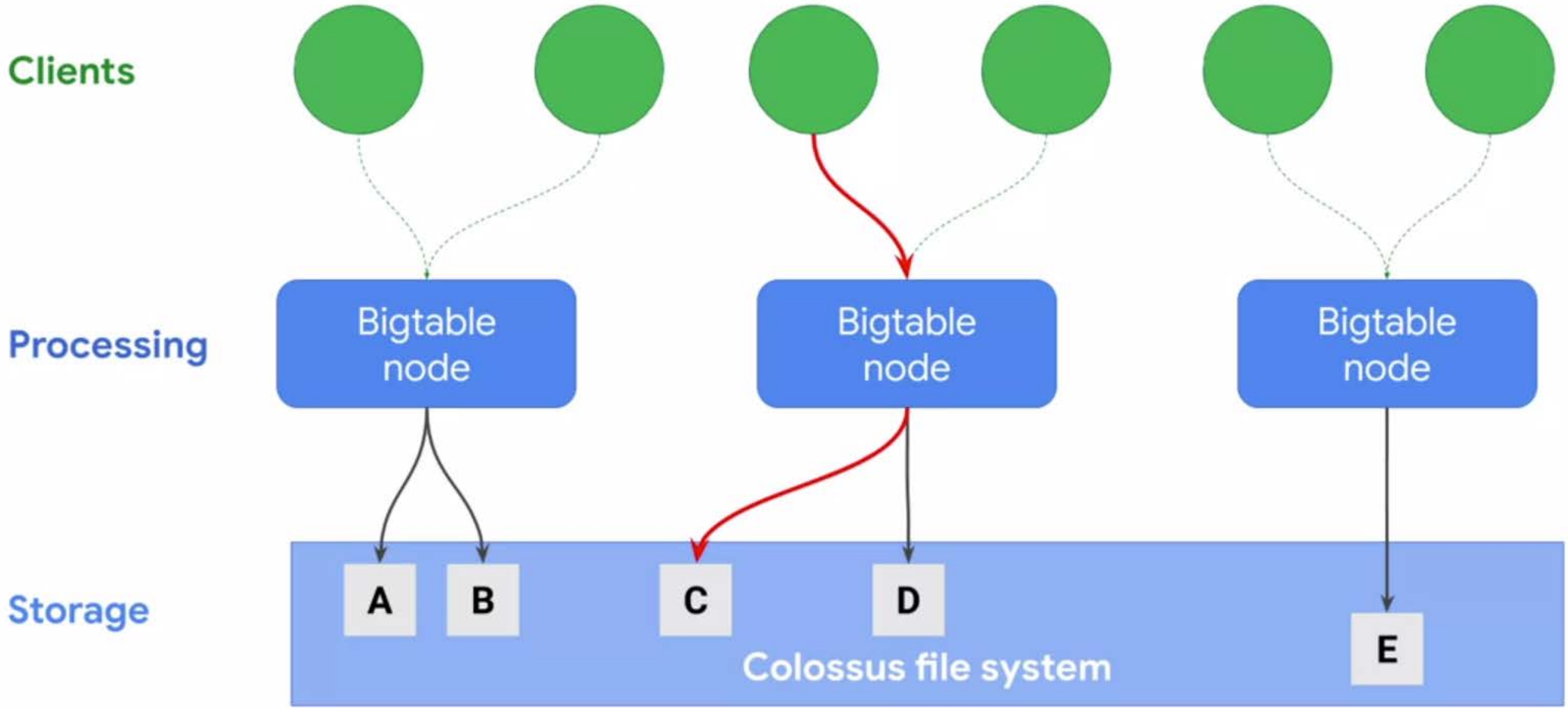


# Cloud Bigtable storage model

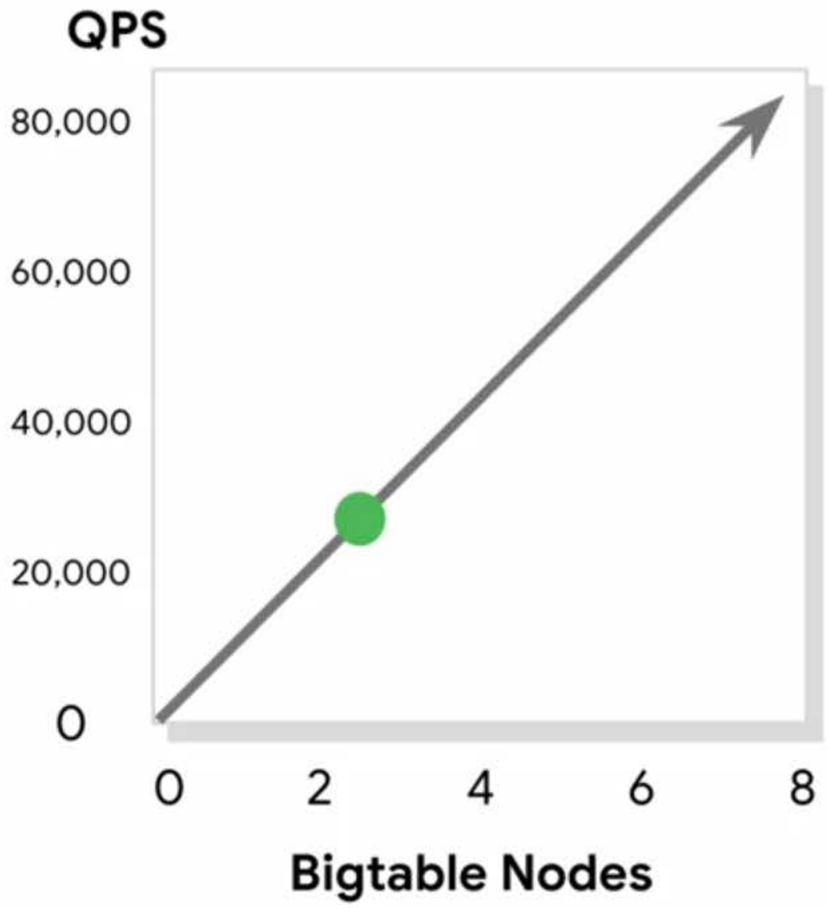
# Processing is separated from storage

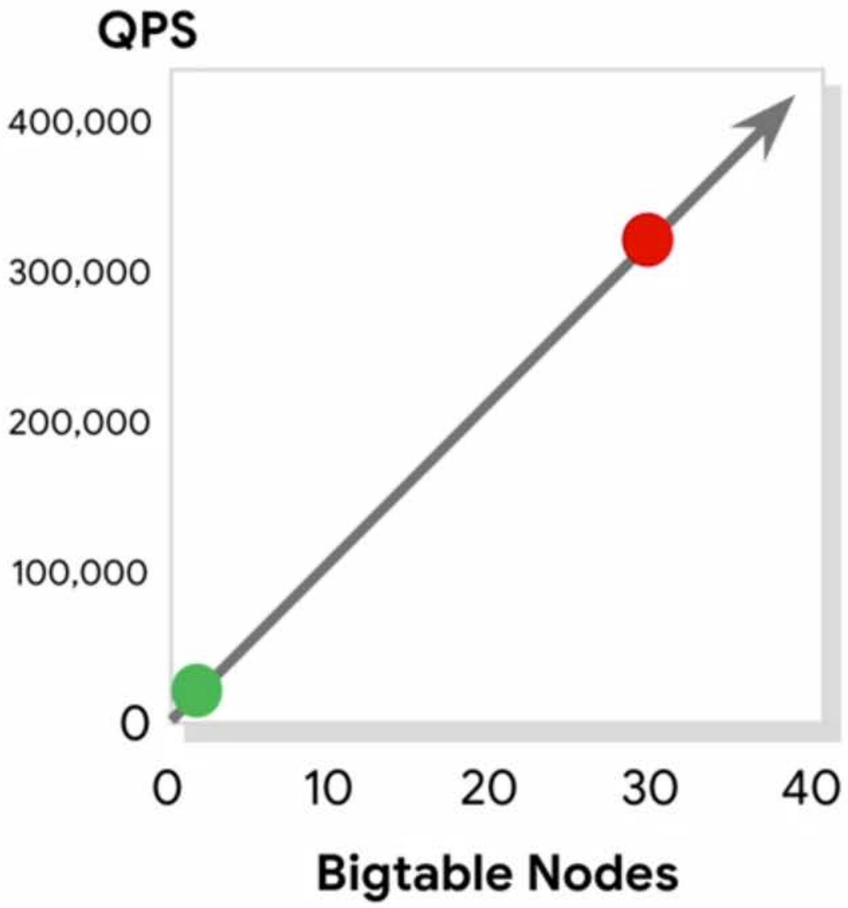


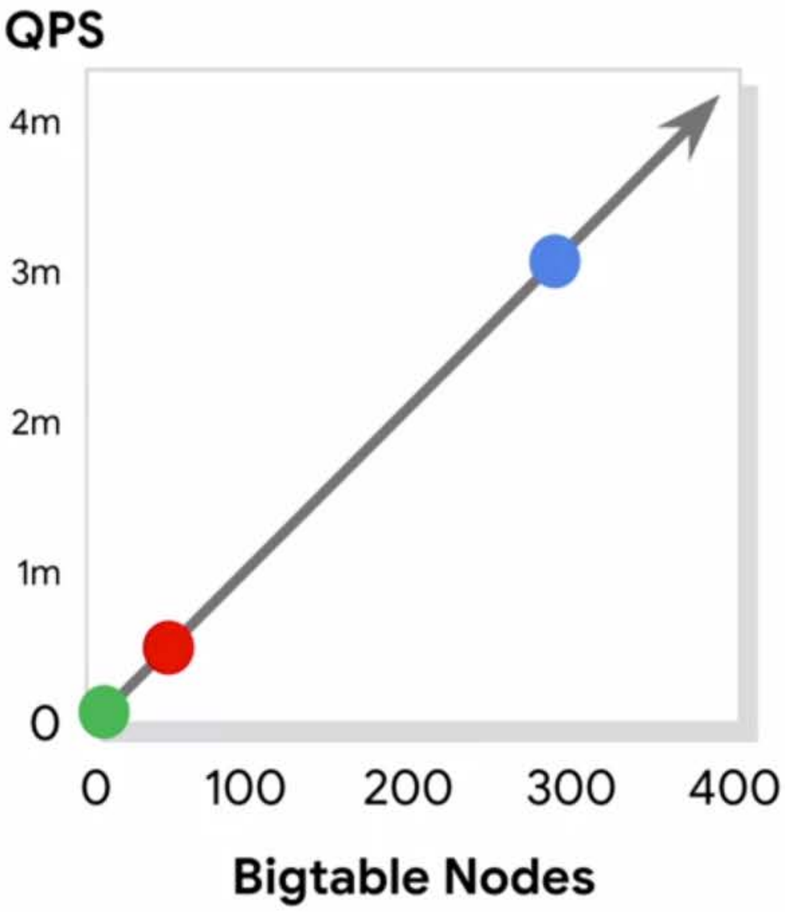
# Rebalances without moving data



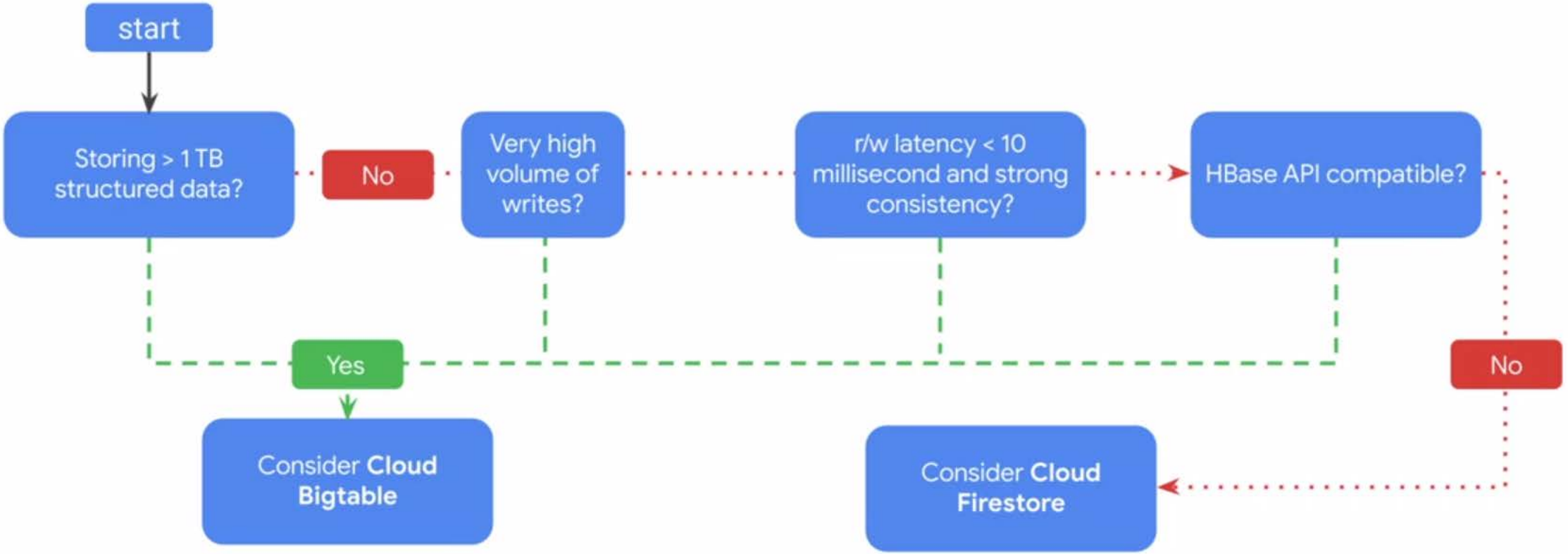
# Throughput scales linearly



# Choosing Cloud Bigtable

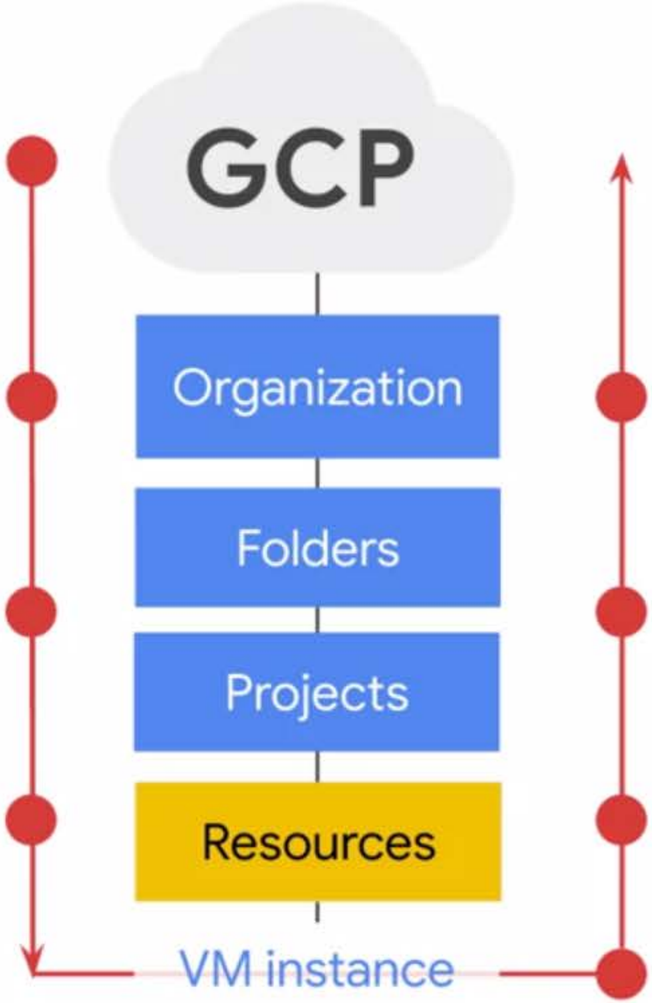



- Bigtable scales UP well
- Cloud Firestore scales DOWN well

# Resource Manager lets you hierarchically manage resources

**Identity and Access Management**

Child policies cannot restrict access granted at the parent level




**Billing and Resource Monitoring**

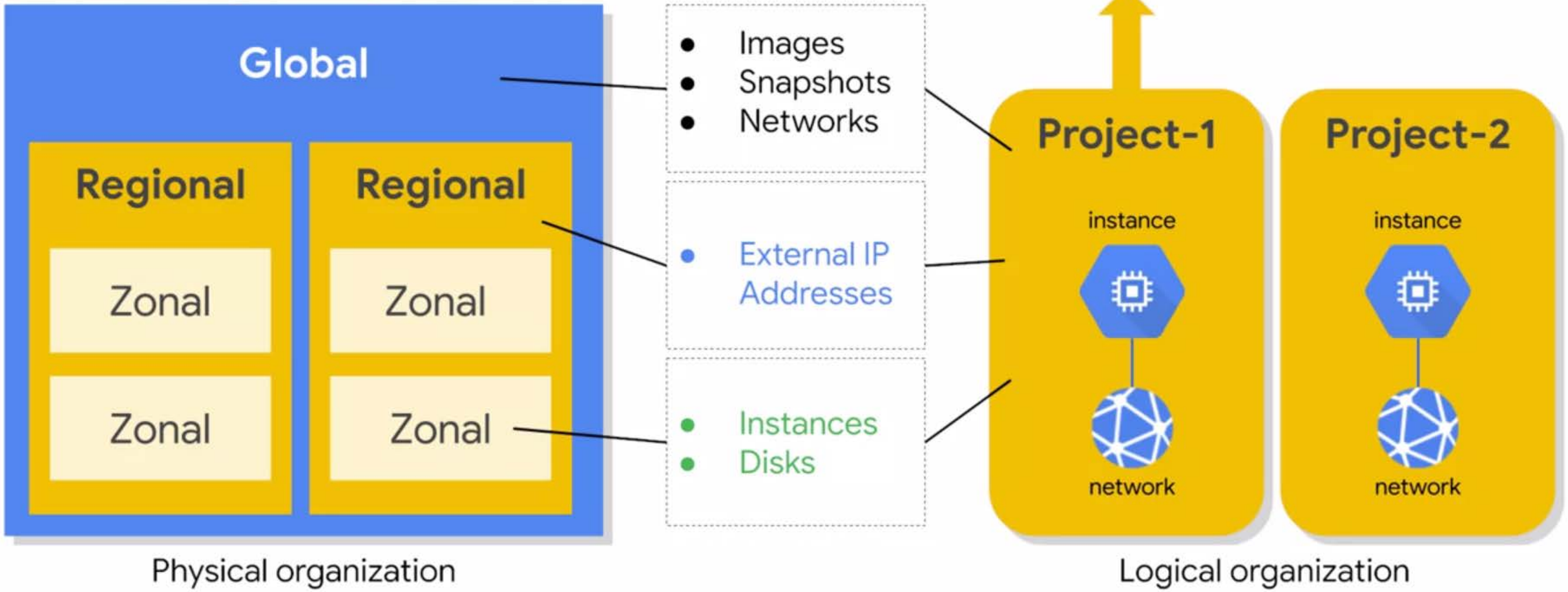
Organization contains all billing accounts

Project is associated with one billing account

A resource belongs to one and only one project

# Resource hierarchy

*Resources are global, regional, or zonal.*




Physical organization

Logical organization

# Project accumulates the consumption of all its resources

- Track resource and quota usage
  - Enable billing
  - Manage permissions and credentials
  - Enable services and APIs
- Projects use three identifying attributes:
  - Project Name
  - Project Number
  - Project ID, also known as Application ID

# All resources are subject to project quotas or limits

- How many resources you can create per project
  - *5 VPC networks/project*
- How quickly you can make API requests in a project: rate limits
  - *5 admin actions/second (Cloud Spanner)*
- How many resources you can create per region
  - *24 CPUs region/project*

**Increase:** Quotas page in GCP Console or a support ticket

# Use labels for ...

- Team or Cost Center
  `team:marketing`
  `team:research`
- Components
  `component: redis`
  `component: frontend`
- Environment or stage
  `environment: prod`
  `environment: test`
- Owner or contact
  `owner:gaurav`
  `contact:opm`
- State
  `state:inuse`
  `state:readyfordeletion`

# Comparing labels and tags

- Labels are a way to organize resources across GCP
  - disks, image, snapshots...
- User-defined strings in `key-value` format
- Propagated through billing

- Tags are applied to instances only
- User-defined strings
- Tags are primarily used for networking (applying firewall rules)

For more information about using

# Google Cloud's operations suite overview

- Integrated monitoring, logging, diagnostics
- Manages across platforms
  - Google Cloud and AWS
  - Dynamic discovery of Google Cloud with smart defaults
  - Open-source agents and integrations
- Access to powerful data and analytics tools
- Collaboration with third-party software

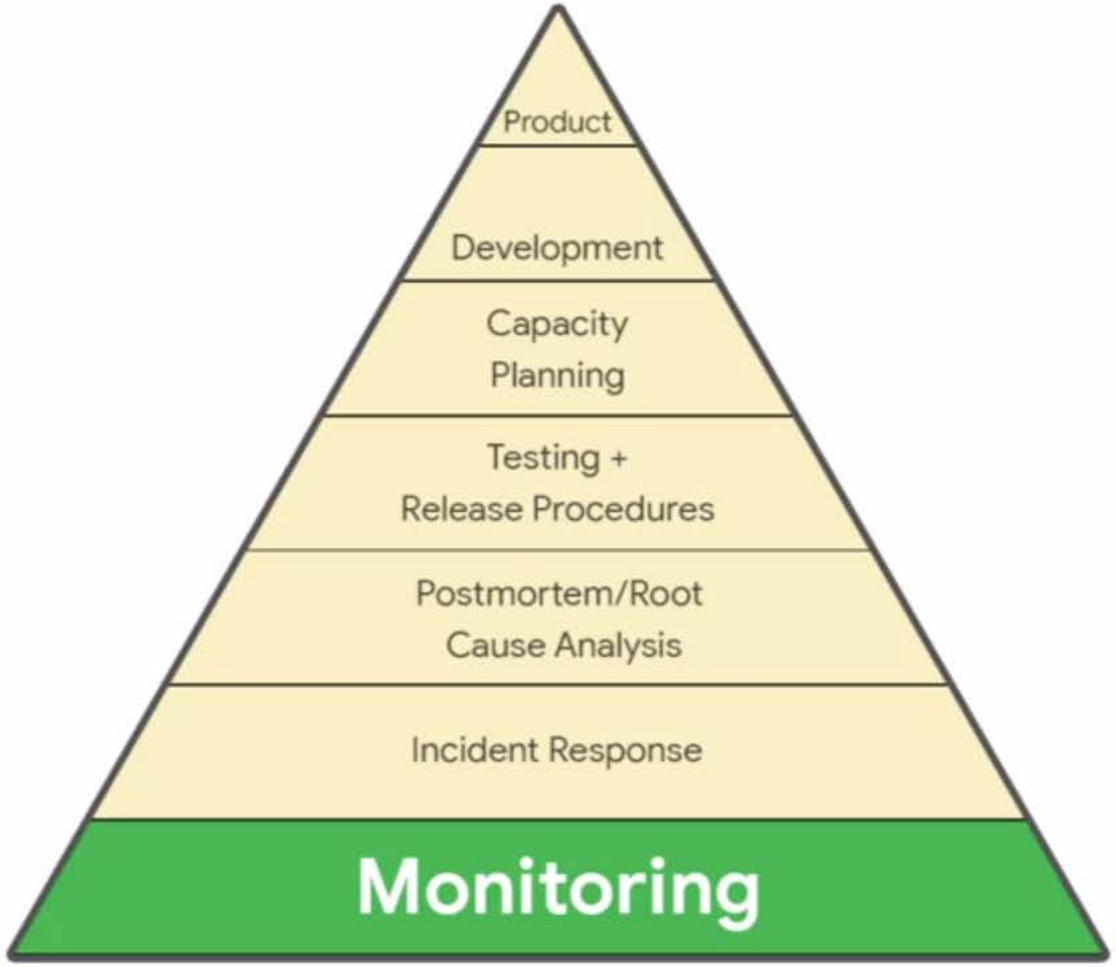
Google Cloud's operations suite

(previously Stackdriver)

# Multiple integrated products


Stackdriver is now Google Cloud's operations suite

# Site reliability engineering



# Monitoring

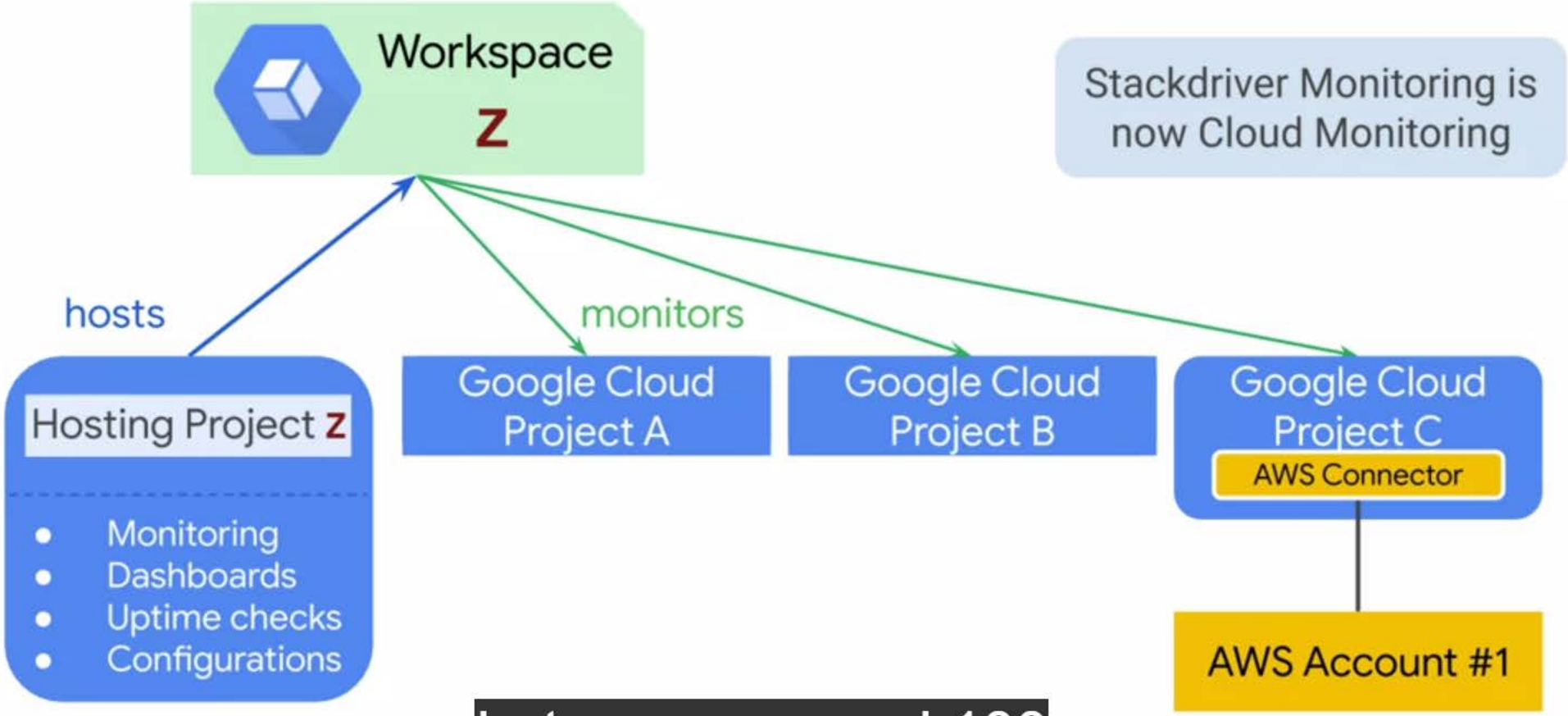
- Dynamic config and intelligent defaults
- Platform, system, and application metrics
  - Ingests data: Metrics, events, metadata
  - Generates insights through dashboards, charts, alerts
- Uptime/health checks
- Dashboards
- Alerts

Cloud Monitoring

(previously Stackdriver Monitoring)

intelligent defaults
that allow you to easily

# Workspace is the root entity that holds monitoring and configuration information




between one and 100 monitored projects,

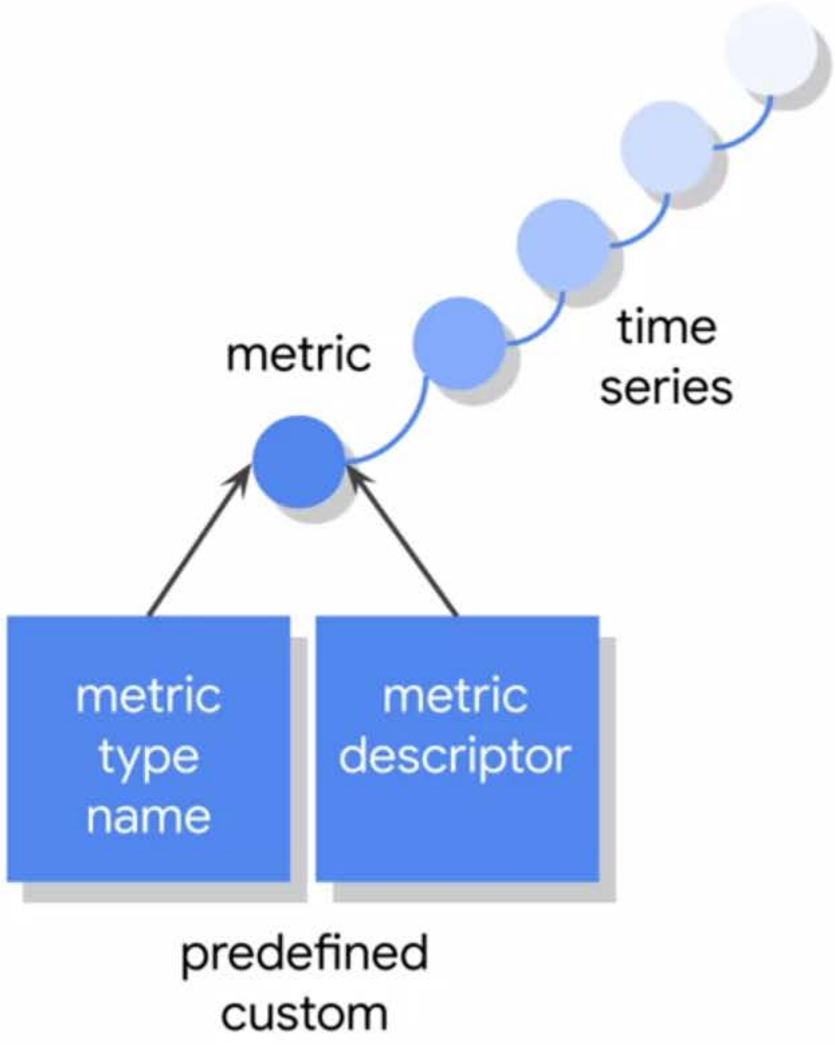
# Custom metrics

Custom metric example in Python:

```
client = monitoring.Client()
descriptor = client.metric_descriptor(
    'custom.googleapis.com/my_metric',

metric_kind=monitoring.MetricKind.GAUGE,
    value_type=monitoring.ValueType.DOUBLE,
    description='This is a simple example
of a custom metric.')
descriptor.create()
```

Stackdriver Monitoring is now Cloud Monitoring



# Logging

- Platform, systems, and application logs
  - API to write to logs
  - 30-day retention
- Log search/view/filter
- Log-based metrics
- Monitoring alerts can be set on log events
- Data can be exported to Cloud Storage, BigQuery, and Pub/Sub

Cloud Logging
(previously
Stackdriver Logging)

# Installing Logging agent

## Install Logging agent

```
curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
sudo bash install-logging-agent.sh
```

# Uptime checks test the availability of your public services


As you can see on this slide,

# Installing Monitoring agent

## Install Monitoring agent (example)

```
curl -sSO https://dl.google.com/cloudagents/add-monitoring-agent-repo.sh
sudo bash add-monitoring-agent-repo.sh
```

a VM instance running Linux
that is being monitored by

# Error Reporting

Aggregate and display errors for running cloud services

- Error notifications
- Error dashboard
- App Engine, Apps Script, Compute Engine, Cloud Functions, Cloud Run, GKE, Amazon EC2
- Go, Java, .NET, Node.js, PHP, Python, and Ruby

Error Reporting

Java,.NET, Node.js, PHP, Python, and Ruby.

# Tracing

Tracing system

- Displays data in near real-time
- Latency reporting
- Per-URL latency sampling

Collects latency data

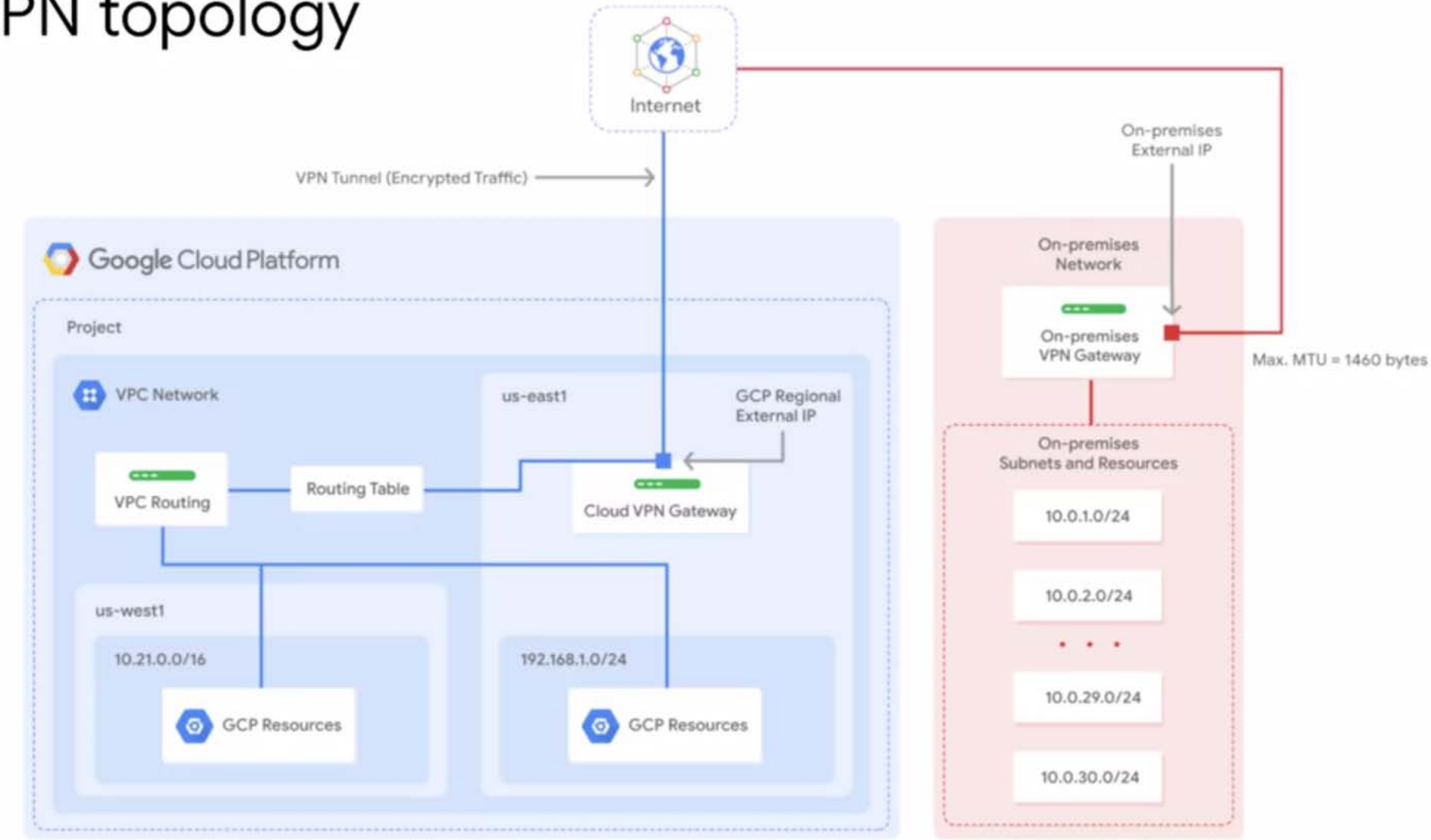
- App Engine
- Google HTTP(S) load balancers
- Applications instrumented with the Cloud Trace SDKs

Cloud Trace

(previously Stackdriver Trace)

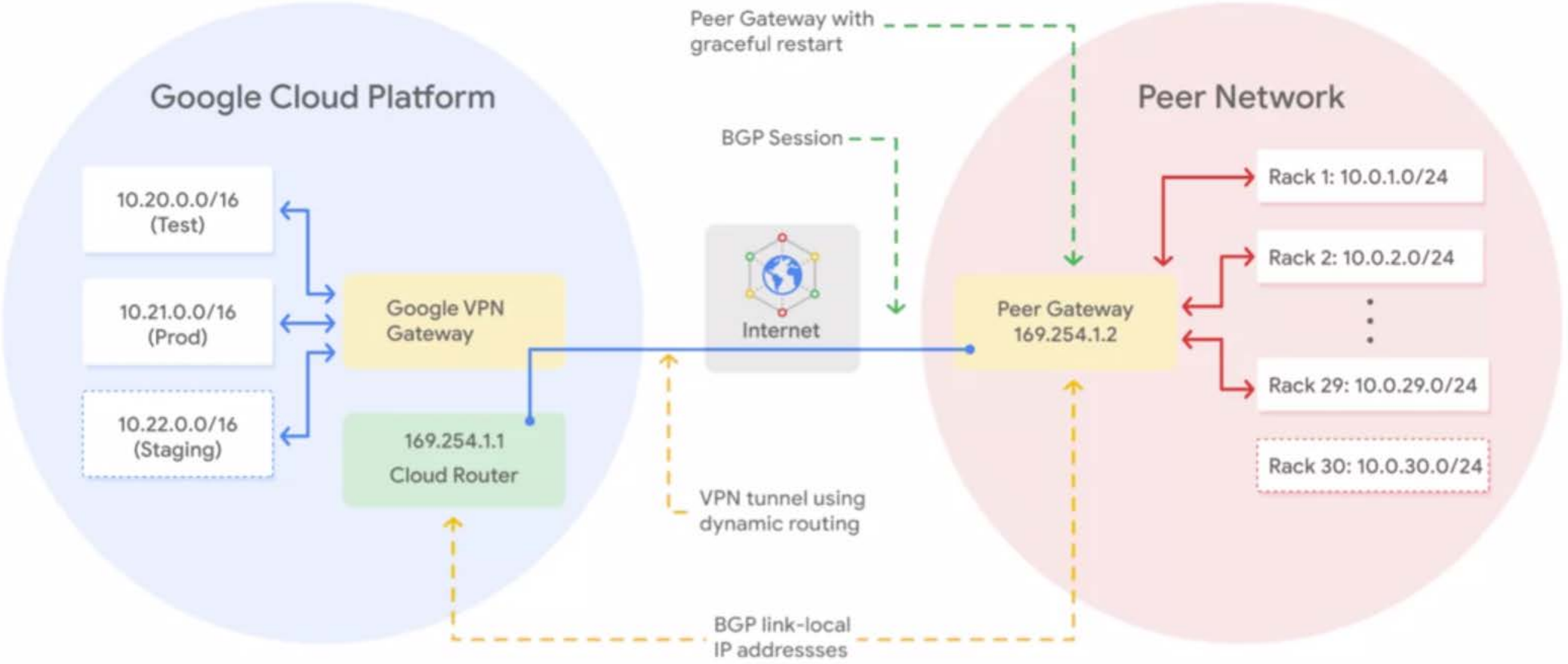
and applications instrumented with the Stackdriver Trace API.

# VPN topology



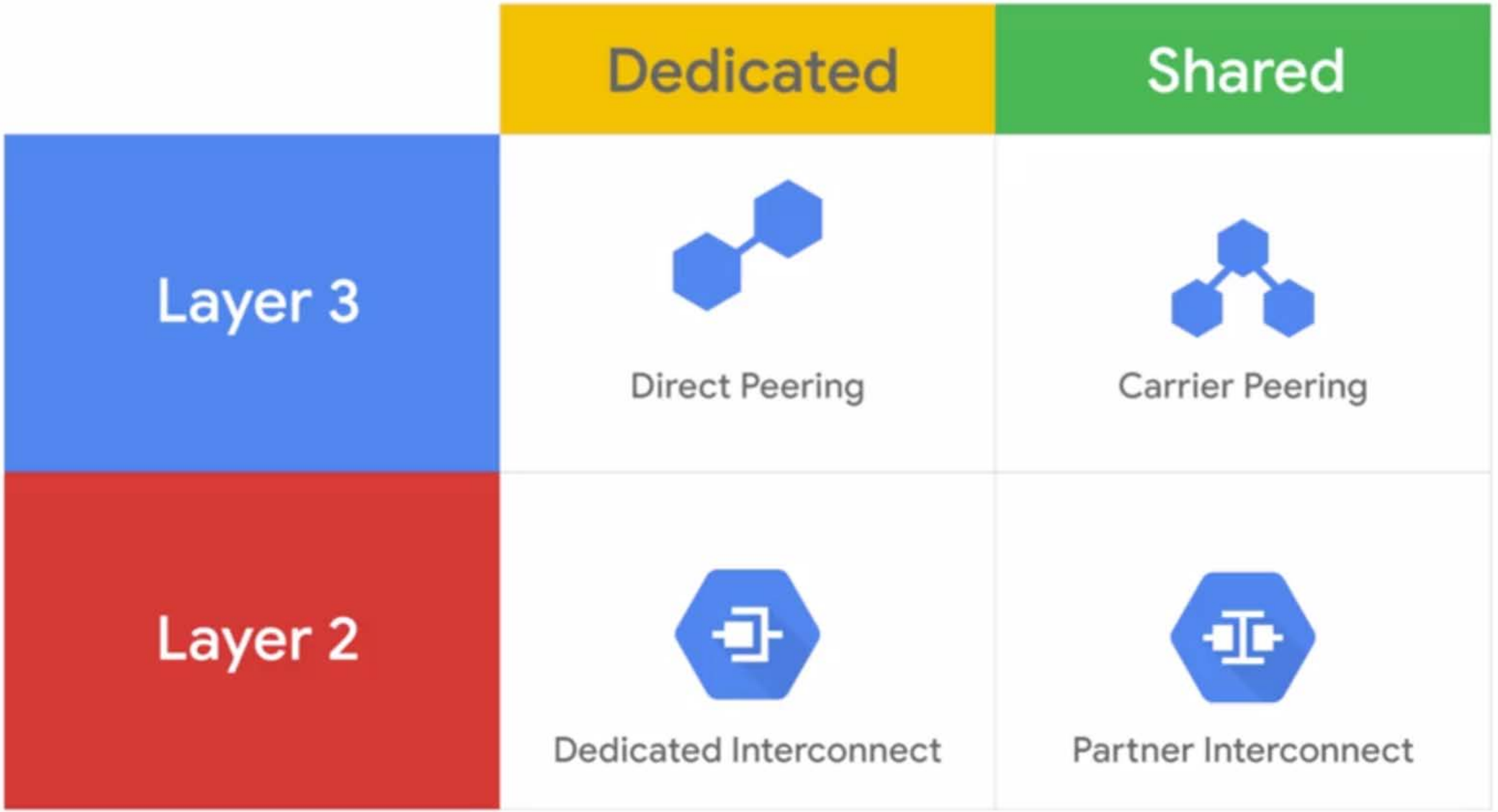
# Dynamic routing with Cloud Router

Cloud Router



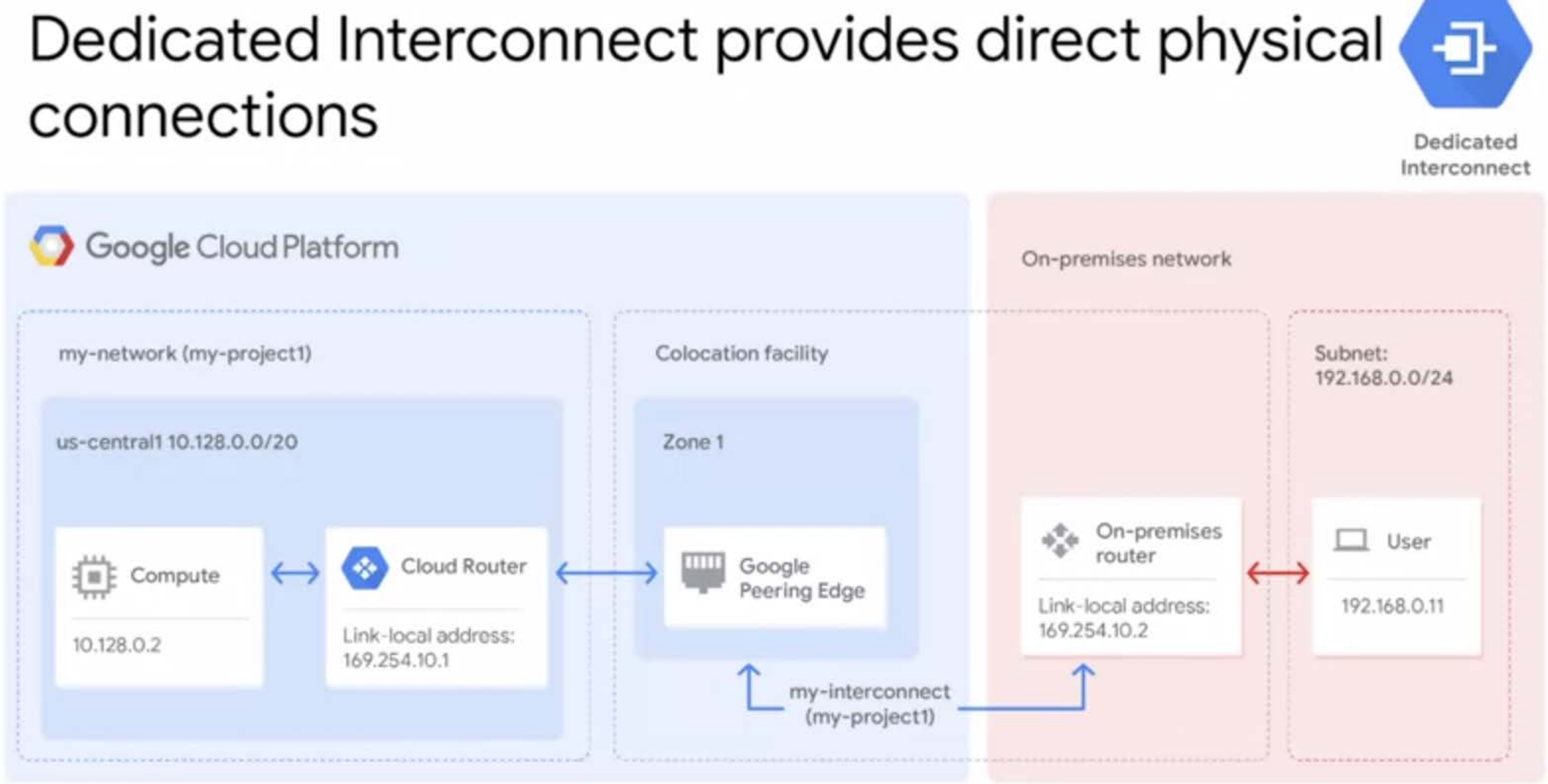
Share

Press Esc to exit full screen

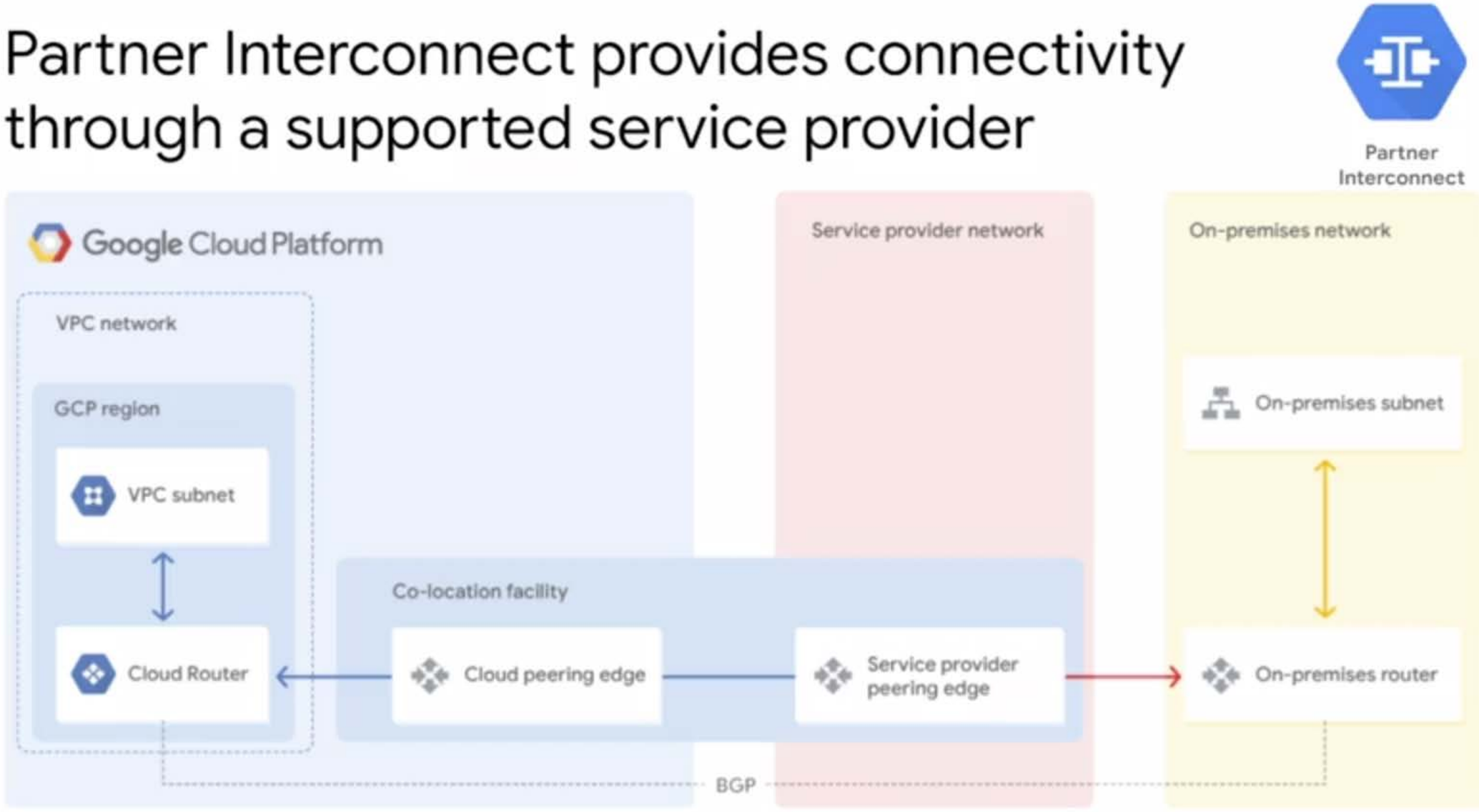



0:10 / 1:25

# Dedicated Interconnect provides direct physical connections



# Partner Interconnect provides connectivity through a supported service provider



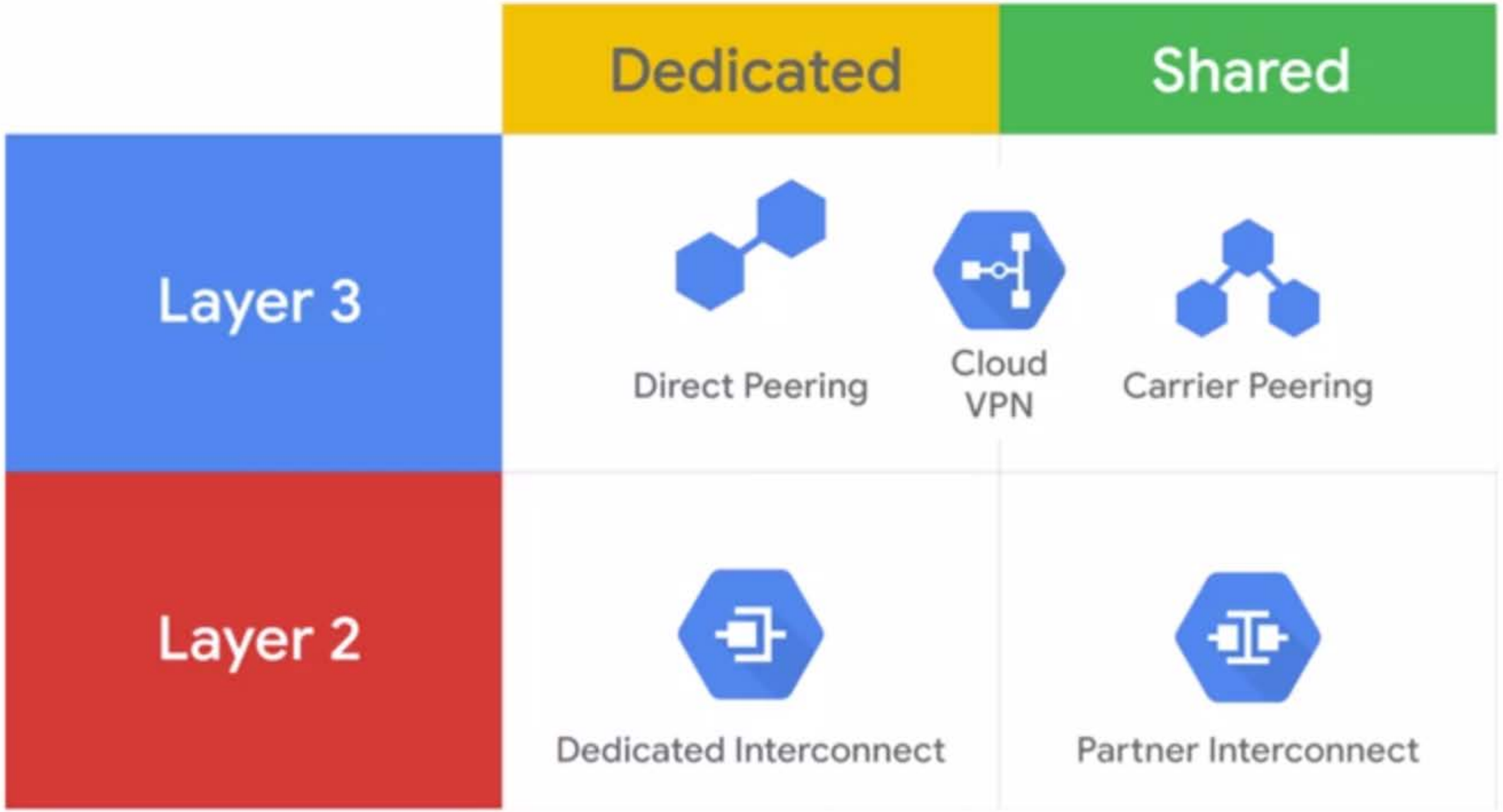
# Comparison of Interconnect options

| Connection | Provides | Capacity | Requirements | Access Type |
|---|---|---|---|---|
| IPsec VPN tunnel | Encrypted tunnel to VPC networks through the public internet | 1.5-3 Gbps per tunnel | On-premises VPN gateway | Internal IP addresses |
| Dedicated Interconnect | Dedicated, direct connection to VPC networks | 10 Gbps per link<br>100 Gbps BETA | Connection in colocation facility | |
| Partner Interconnect | Dedicated bandwidth, connection to VPC network through a service provider | 50 Mbps – 10 Gbps per connection | Service provider | |

# Comparison of Peering options

| Connection | Provides | Capacity | Requirements | Access Type |
|---|---|---|---|---|
| Direct Peering | Dedicated, direct connection to Google's network | 10 Gbps Per link | Connection in GCP PoPs | Public IP addresses |
| Carrier Peering | Peering through service provider to Google's public network | Varies based on partner offering | Service provider | |

# 5 ways to connect your infrastructure to GCP



# Choosing a network connection option

## Interconnect

Direct access to RFC1918 IPs in your VPC – with SLA

Dedicated Interconnect

Partner Interconnect

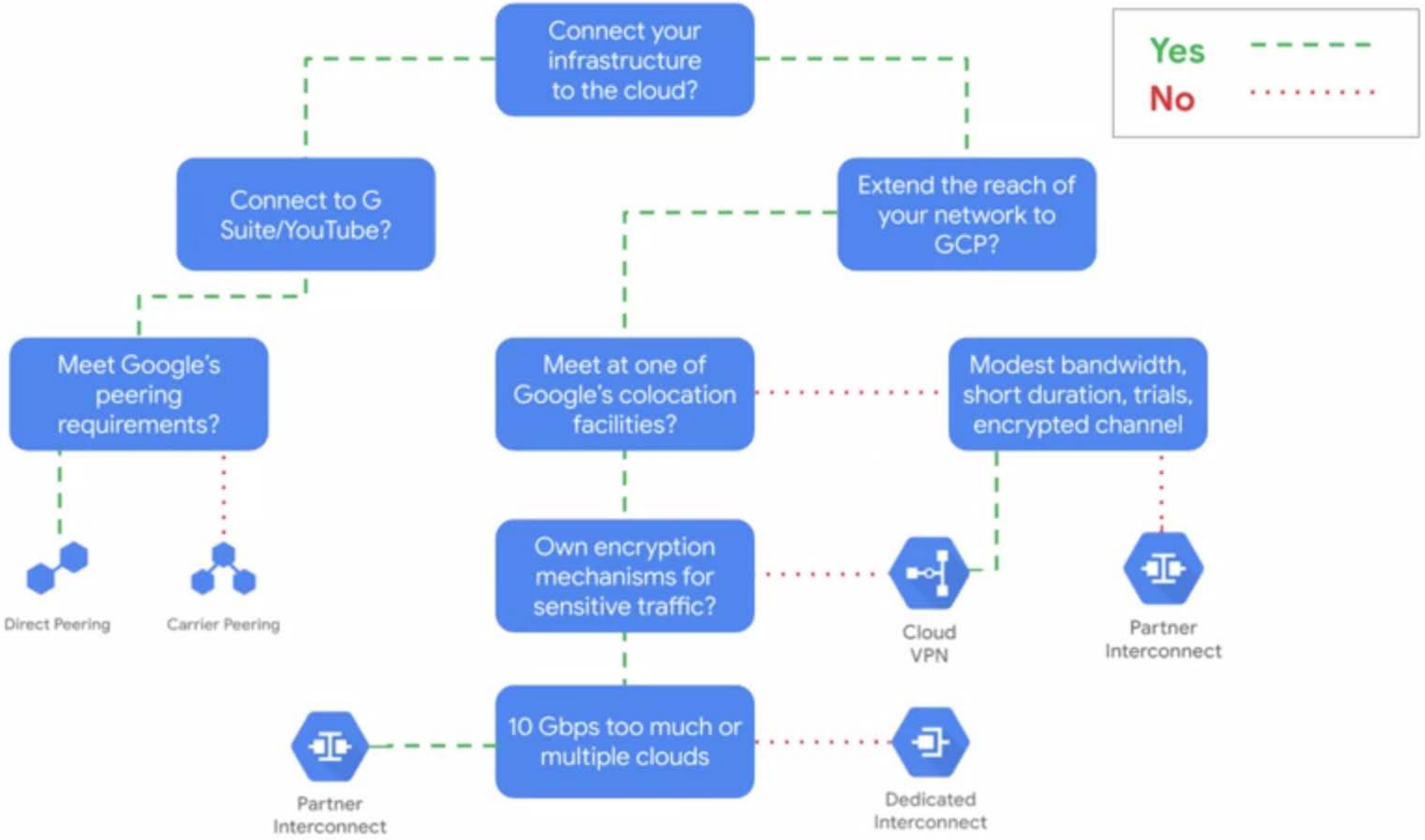
Cloud VPN

## Peering

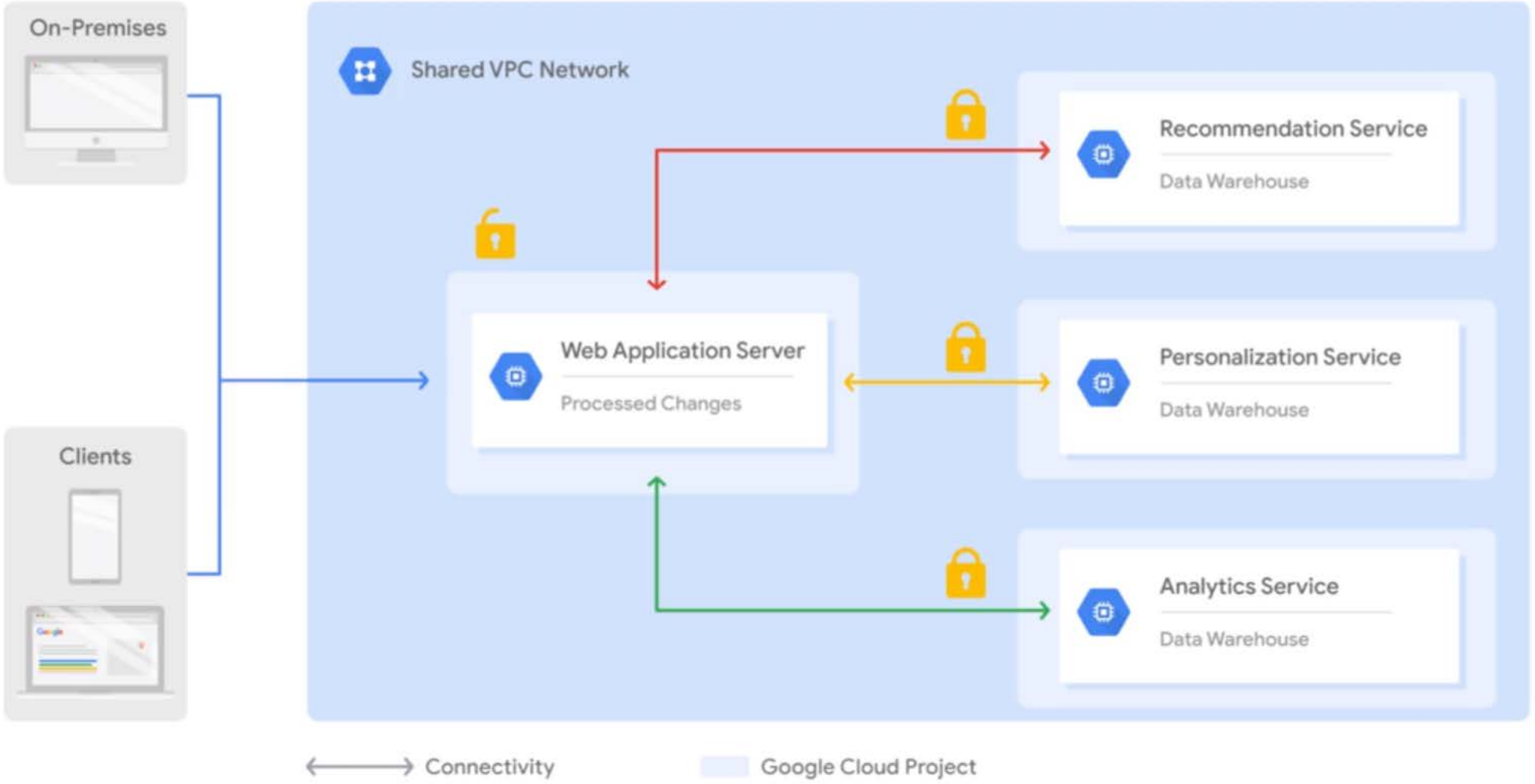
Access to Google public IPs only – without SLA

Direct Peering

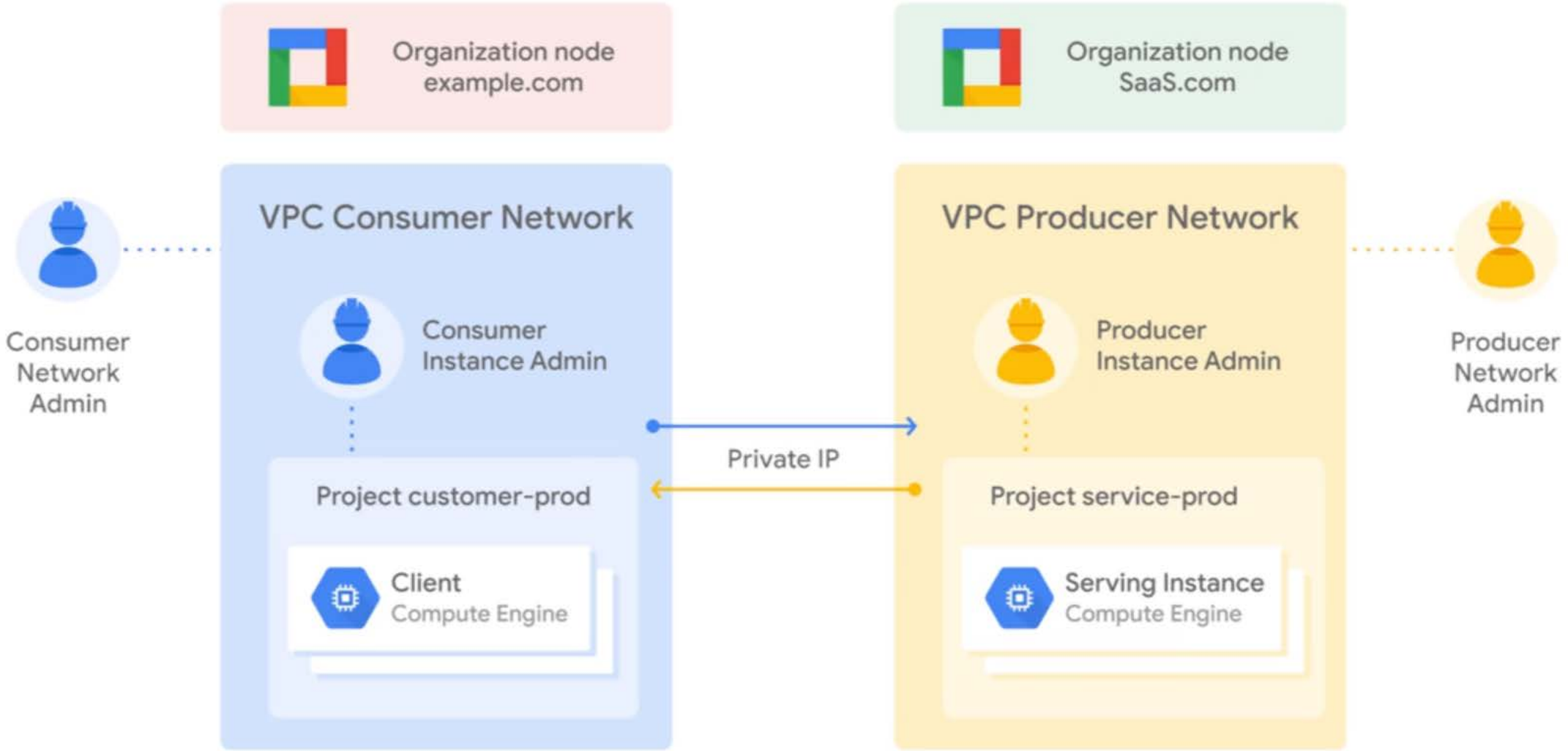
Carrier Peering


Connect your infrastructure to the cloud?
Yes
No
Connect to G Suite/YouTube?
Extend the reach of your network to GCP?
Meet Google's peering requirements?
Meet at one of Google's colocation facilities?
Modest bandwidth, short duration, trials, encrypted channel
Direct Peering
Carrier Peering
Own encryption mechanisms for sensitive traffic?
Cloud VPN
Partner Interconnect
10 Gbps too much or multiple clouds
Partner Interconnect
Dedicated Interconnect

# Shared VPC


On-Premises
Clients
Shared VPC Network
Web Application Server
Processed Changes
Recommendation Service
Data Warehouse
Personalization Service
Data Warehouse
Analytics Service
Data Warehouse
Connectivity
Google Cloud Project

# VPC peering

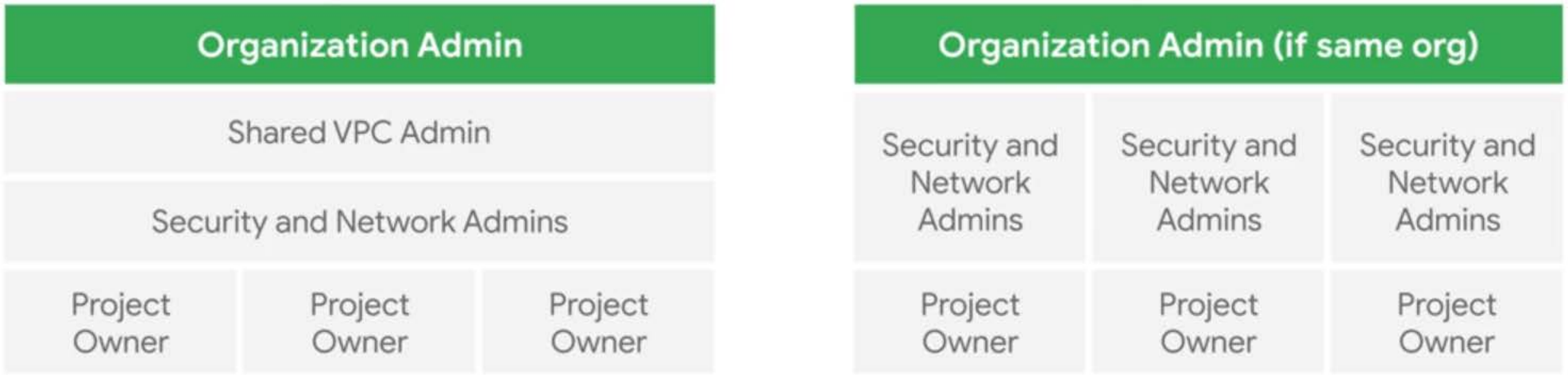

Organization node
example.com
Organization node
SaaS.com
VPC Consumer Network
VPC Producer Network
Consumer
Network
Admin
Consumer
Instance Admin
Producer
Instance Admin
Producer
Network
Admin
Private IP
Project customer-prod
Project service-prod
Client
Compute Engine
Serving Instance
Compute Engine

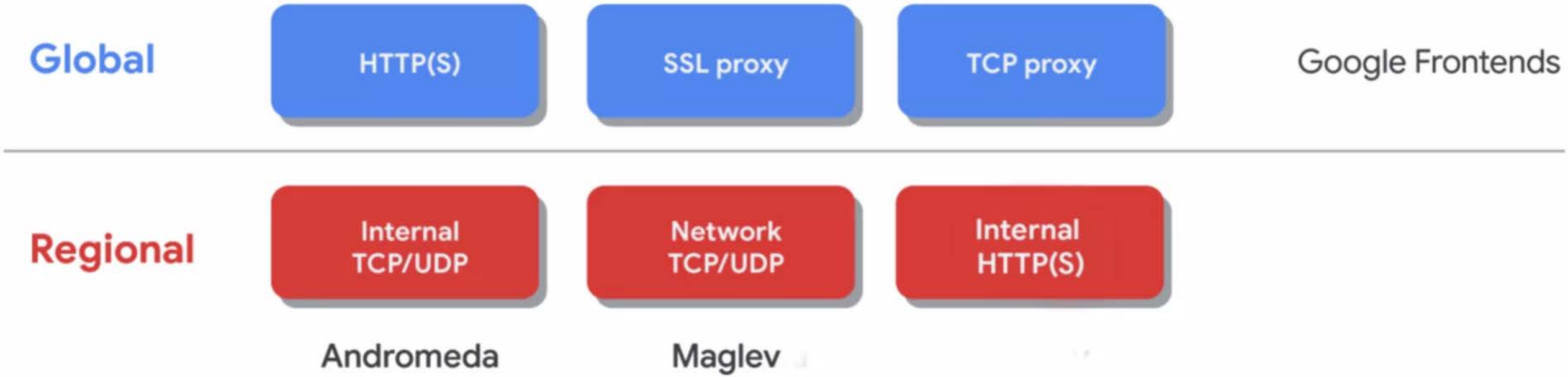
# Shared VPC vs. VPC peering

| Consideration | Shared VPC | VPC Network Peering |
| --- | --- | --- |
| Across organizations | No | Yes |
| Within project | No | Yes |
| Network administration | Centralized | Decentralized |

# Shared VPC vs. VPC peering

| Consideration | Shared VPC | VPC Network Peering |
| --- | --- | --- |
| Across organizations | No | Yes |
| Within project | No | Yes |
| Network administration | Centralized | Decentralized |




Global
HTTP(S)
SSL proxy
TCP proxy
Google Frontends
Regional
Internal TCP/UDP
Network TCP/UDP
Internal HTTP(S)
Andromeda
Maglev

# Managed instance groups

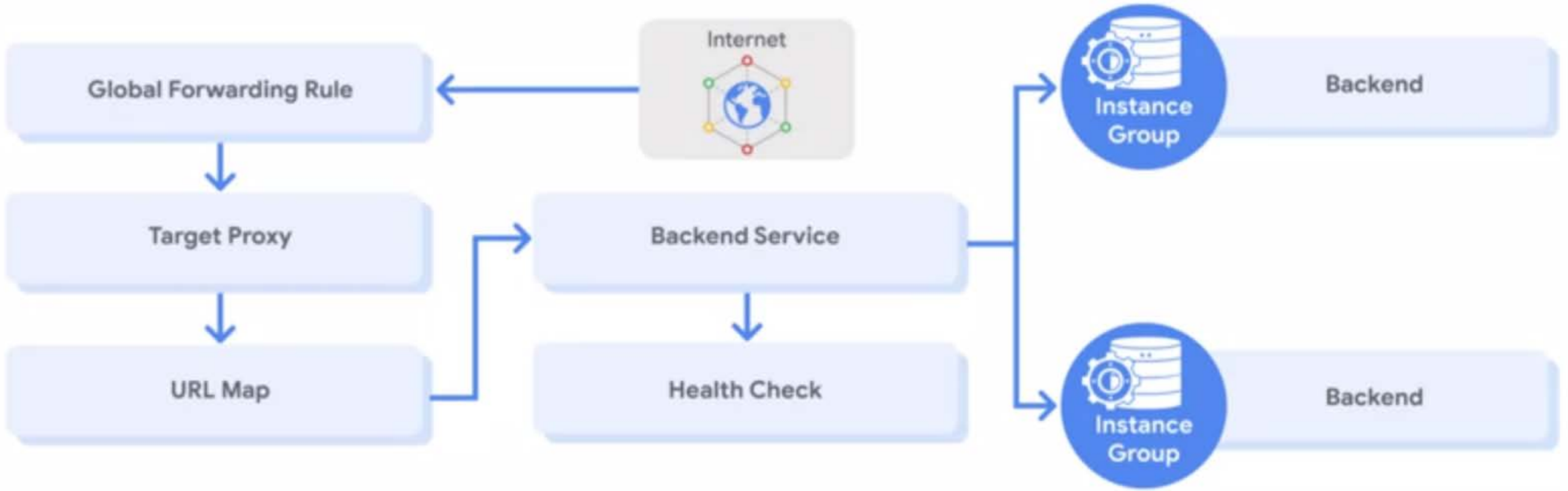
- Deploy identical instances based on instance template
- Instance group can be resized
- Manager ensures all instances are RUNNING
- Typically used with autoscaler
- Can be single zone or regional

# Managed instance groups offer autoscaling capabilities

- Dynamically add/remove instances:
  - Increases in load
  - Decreases in load
- Autoscaling policy:
  - CPU utilization
  - Load balancing capacity
  - Monitoring metrics
  - Queue-based workload

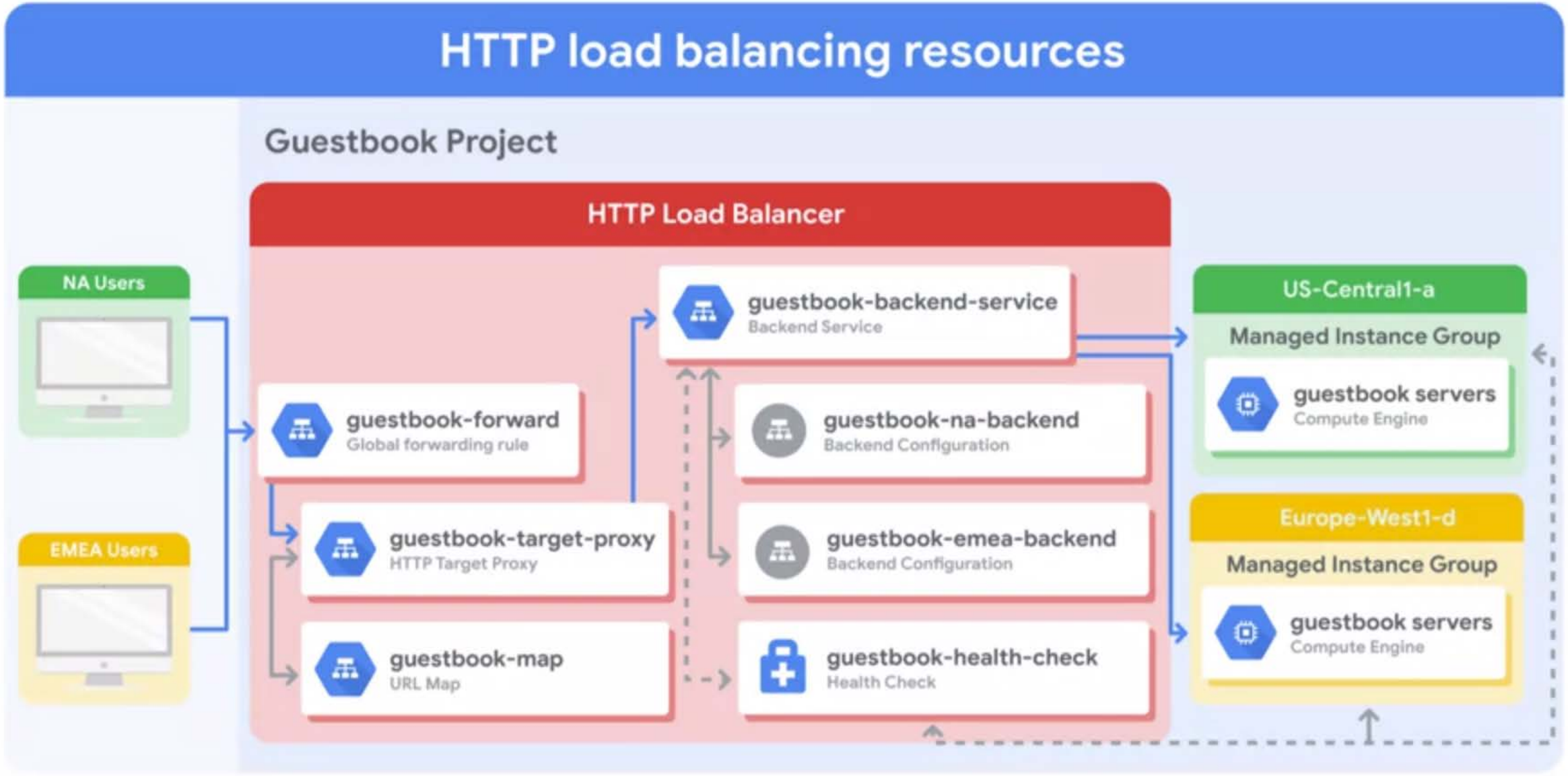

Target CPU utilization = 75%

# Architecture of an HTTP(S) load balancer

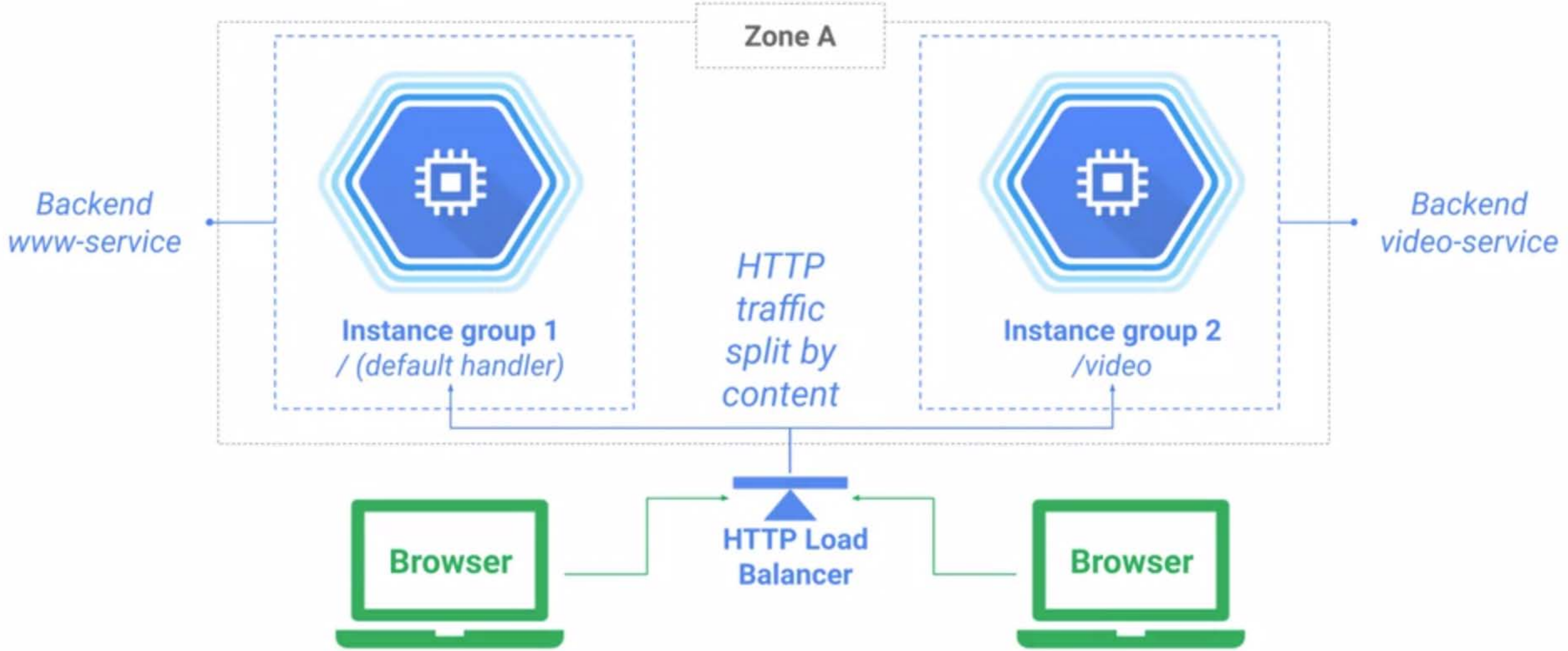


# Backend services

- Health check
- Session affinity (optional)
- Time out setting (30-sec default)
- One or more backends
  - An instance group (managed or unmanaged)
  - A balancing mode (CPU utilization or RPS)
  - A capacity scaler (ceiling % of CPU/Rate targets)

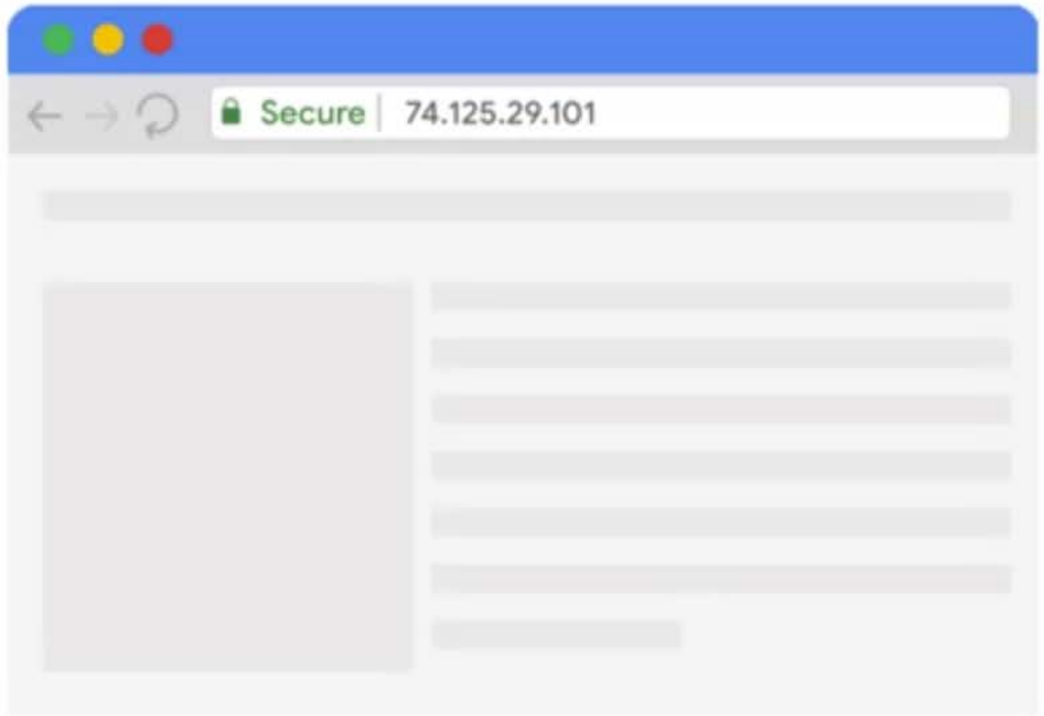

HTTP load balancing resources
Guestbook Project
HTTP Load Balancer
NA Users
EMEA Users
guestbook-forward
Global forwarding rule
guestbook-target-proxy
HTTP Target Proxy
guestbook-map
URL Map
guestbook-backend-service
Backend Service
guestbook-na-backend
Backend Configuration
guestbook-emea-backend
Backend Configuration
guestbook-health-check
Health Check
US-Central1-a
Managed Instance Group
guestbook servers
Compute Engine
Europe-West1-d
Managed Instance Group
guestbook servers
Compute Engine

# Example: Content-based load balancing

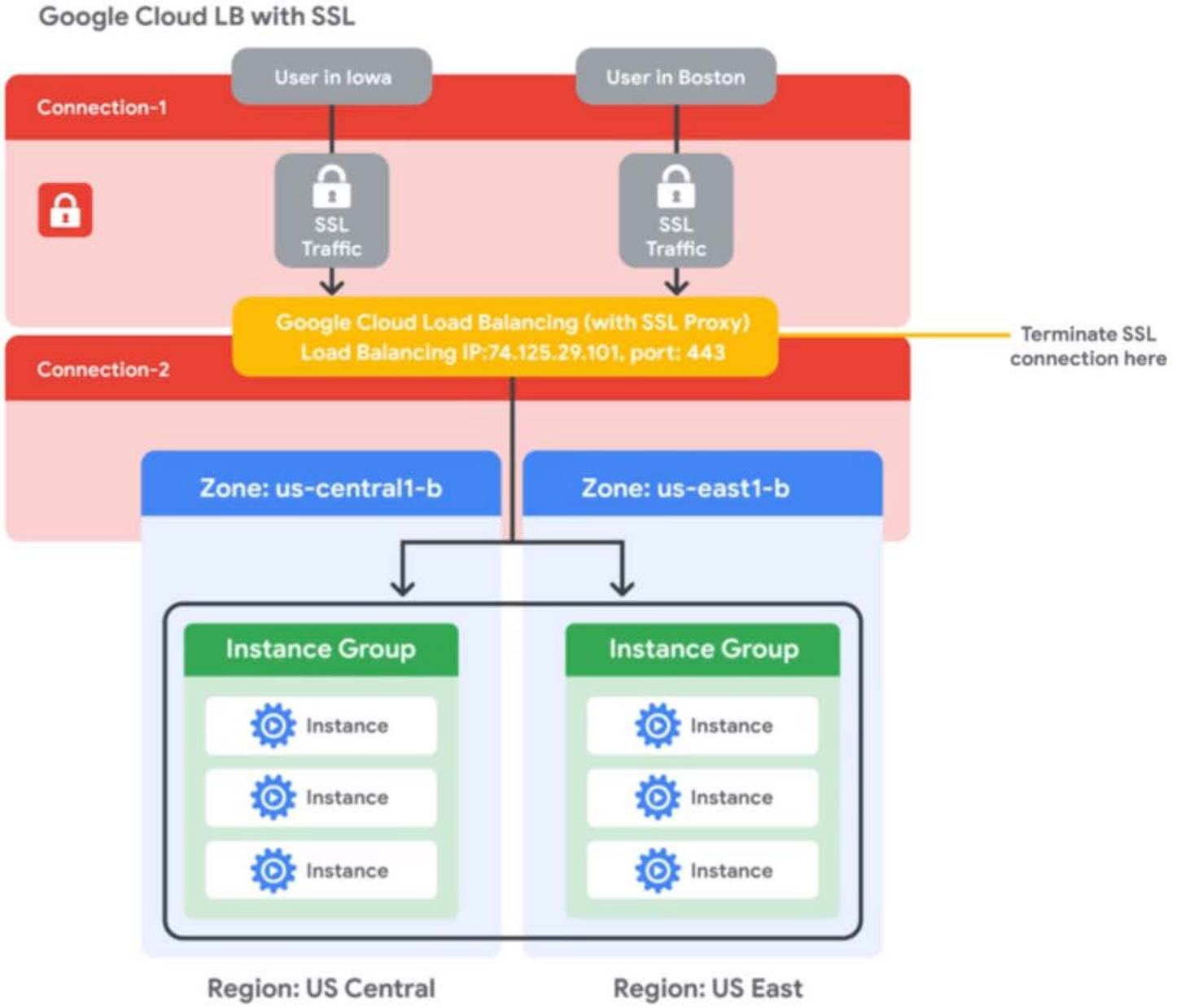


# HTTP(S) load balancing

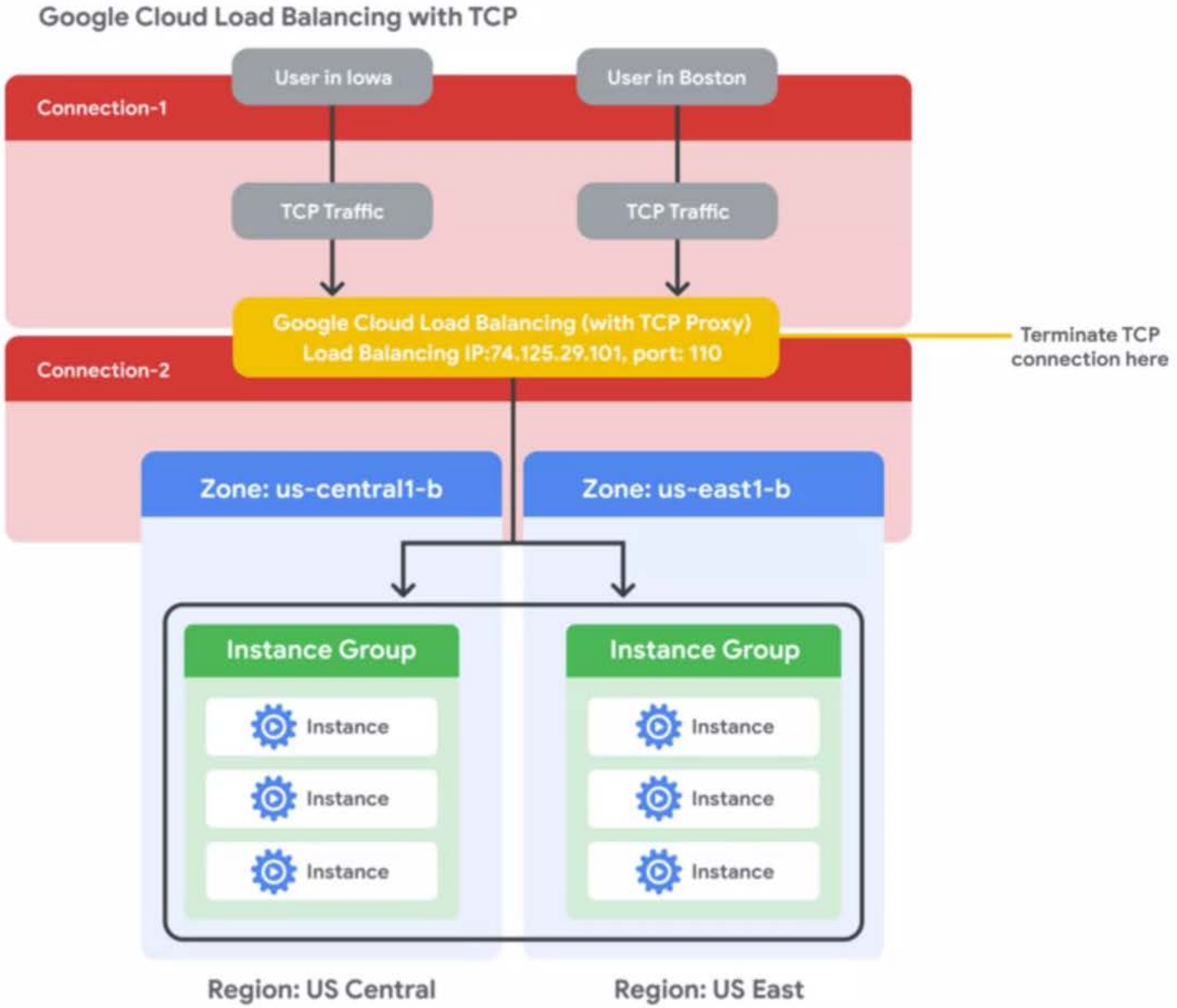
- Target HTTP(S) proxy
- One signed SSL certificate installed (minimum)
- Client SSL session terminates at the load balancer
- Support the QUIC transport layer protocol



# Example: SSL proxy load balancing



# Example: TCP proxy load balancing



# Network load balancing

- Regional, *non-proxied* load balancer
- Forwarding rules (IP protocol data)
- Traffic:
  - UDP
  - TCP/SSL ports
- Backends:
  - Instance group
  - Target pool

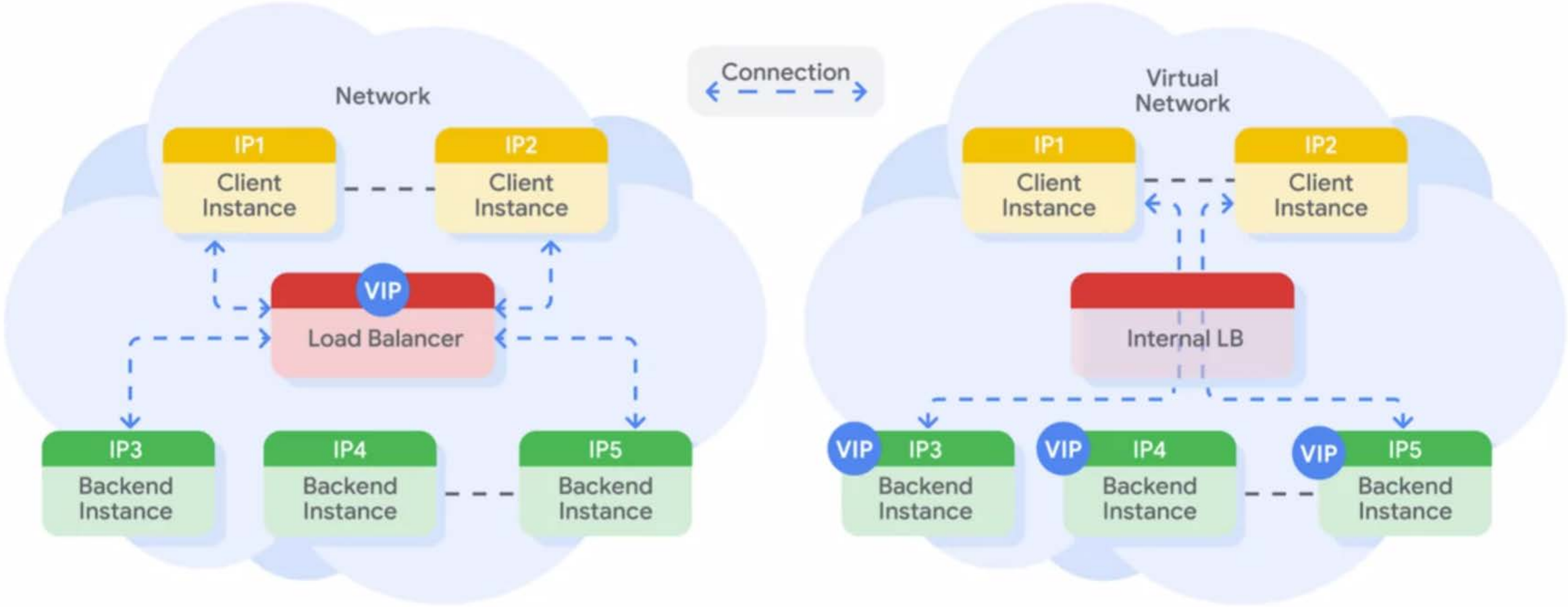
# Target pool resource defines a group of instances that receive incoming traffic from forwarding rules

- Forwarding rules (TCP and UDP)
- Up to 50 per project
- One health check
- Instances must be in the same region

# Internal load balancing

- Regional, private load balancing
  - VM instances in same region
  - RFC 1918 IP addresses
- TCP/UDP traffic
- Reduced latency, simpler configuration
- Software-defined, fully distributed load balancing

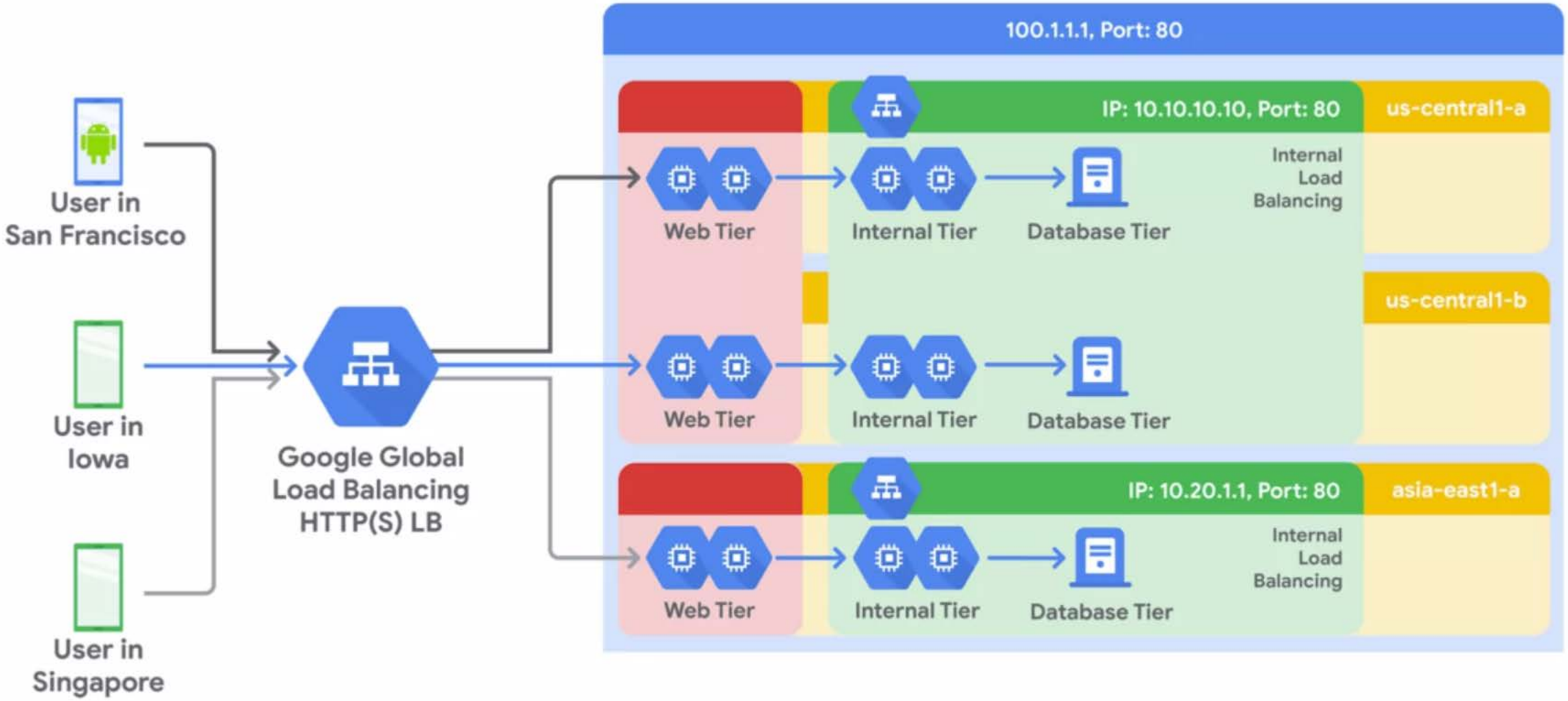
# Software-defined, fully distributed load balancing



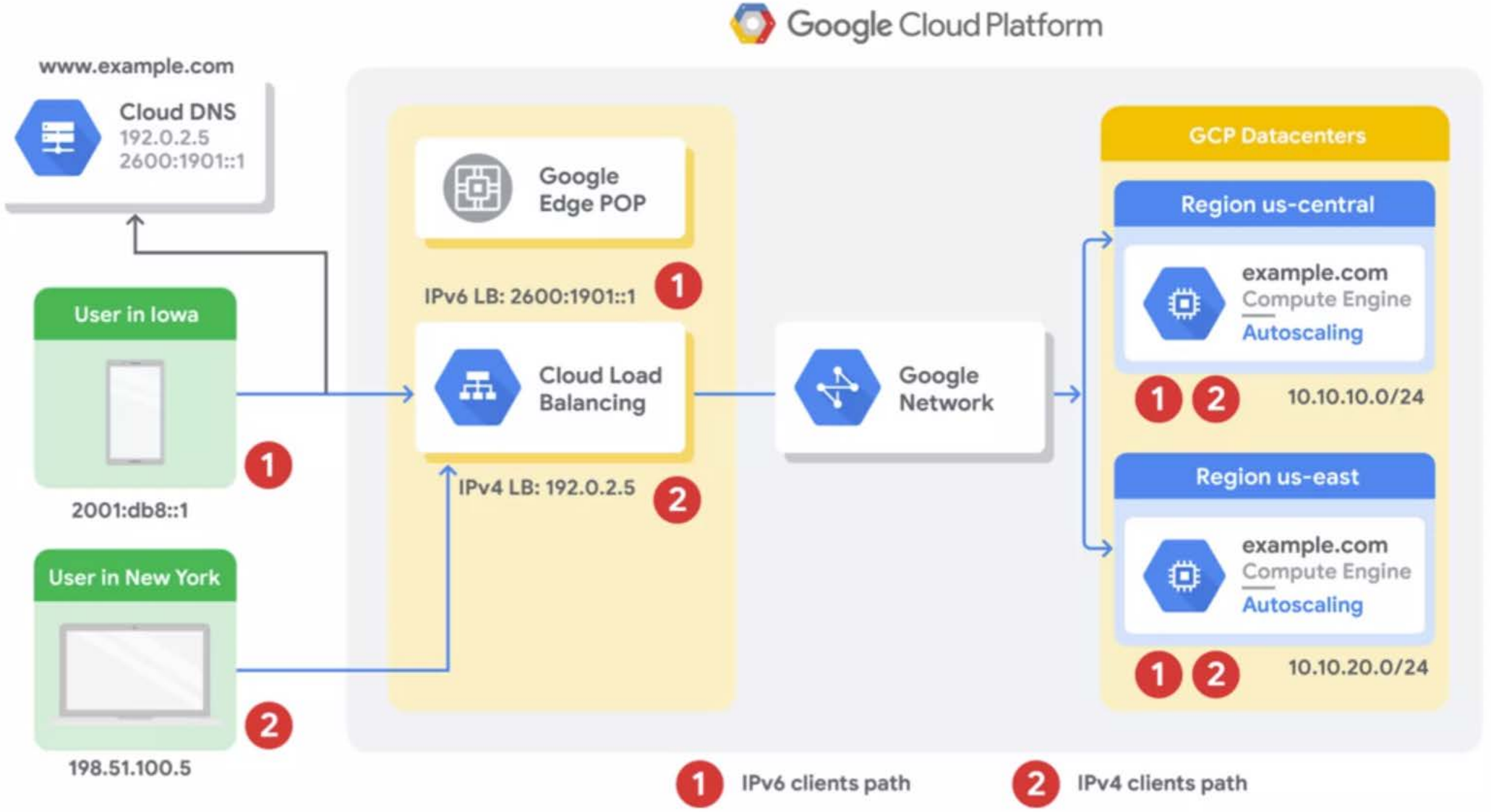

1. Proxy Internal Load Balancing

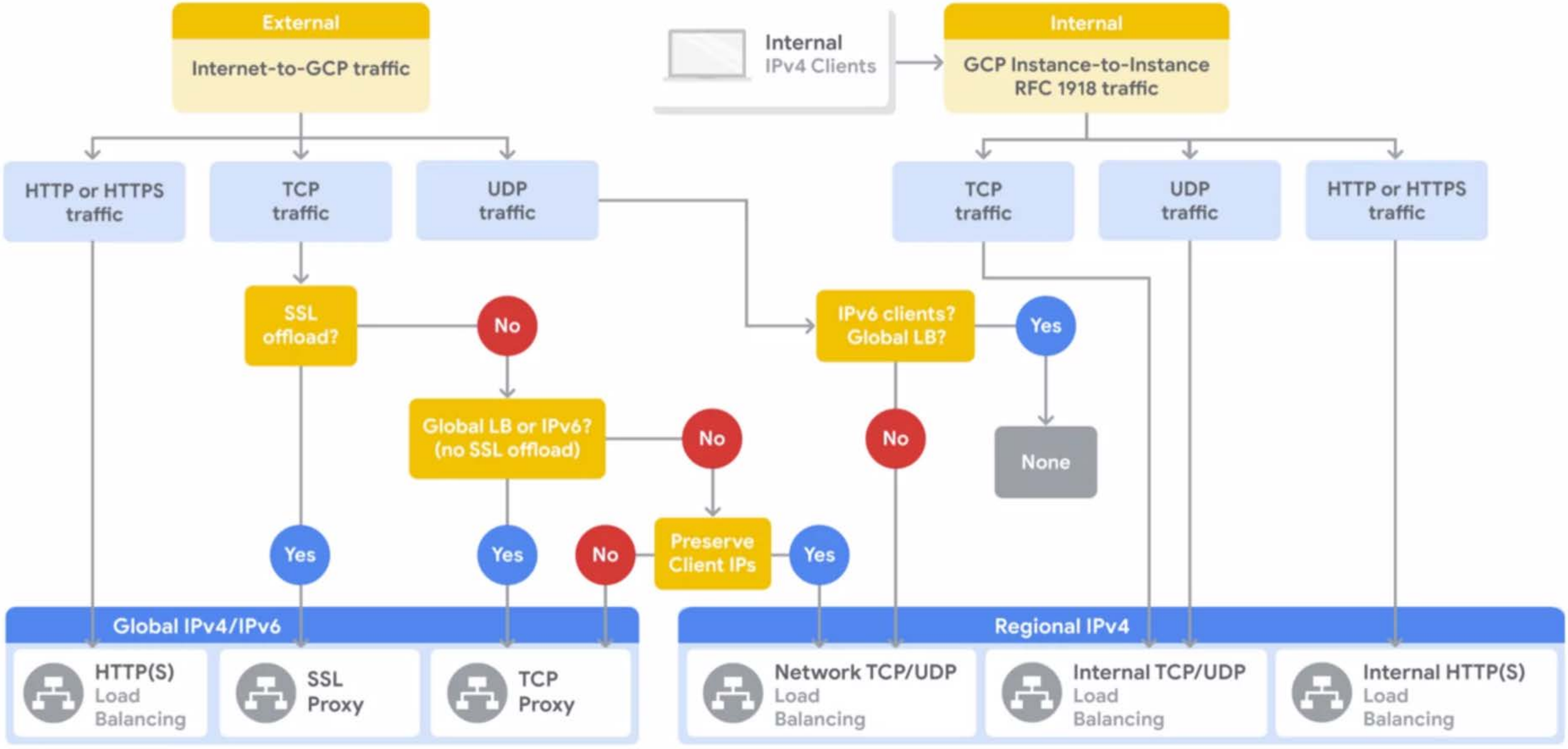
2. Google Cloud Internal Load Balancing

# Internal load balancing supports 3-tier web services



# IPv6 termination for load balancing




External
Internet-to-GCP traffic
Internal
IPv4 Clients
Internal
GCP Instance-to-Instance
RFC 1918 traffic
HTTP or HTTPS traffic
TCP traffic
UDP traffic
TCP traffic
UDP traffic
HTTP or HTTPS traffic
SSL offload?
No
IPv6 clients? Global LB?
Yes
Global LB or IPv6? (no SSL offload)
No
No
None
Yes
Yes
No
Preserve Client IPs
Yes
Global IPv4/IPv6
Regional IPv4
HTTP(S) Load Balancing
SSL Proxy
TCP Proxy
Network TCP/UDP Load Balancing
Internal TCP/UDP Load Balancing
Internal HTTP(S) Load Balancing

# Summary of load balancers

| Load balancer | Traffic type | Global/ Regional | External/ Internal | External ports for load balancing |
|---|---|---|---|---|
| HTTP(S) | HTTP or HTTPS | Global<br>IPv4<br>IPv6 | External | HTTP on 80 or 8080; HTTPS on 443 |
| SSL Proxy | TCP with SSL offload | | | 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222 |
| TCP Proxy | • TCP without SSL offload<br>• Does not preserve client IP addresses | | | 25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222 |
| Network TCP/UDP | • TCP/UDP without SSL offload<br>• Preserves client IP addresses | Regional<br>IPv4 | | Any |
| Internal TCP/UDP | TCP or UDP | | Internal | Any |
| Internal HTTP(S) | HTTP or HTTPS | | | HTTP on 80 or 8080; HTTPS on 443 |

# Deployment Manager is an infrastructure automation tool


- Repeatable deployment process
- Declarative language
- Focus on the application
- Parallel deployment
- Template-driven







automates the creation and

# Example: Auto mode network with HTTP firewall rule

**autonetwork.jinja**

```
resources:
- name: {{ env["name"] }}
  type: compute.v1.network
  properties:
    autoCreateSubnetworks: true
```

**firewall.jinja**

```
resources:
- name: {{ env["name"] }}
  type: compute.v1.firewall
  properties:
    network: {{ properties["network"] }}
    sourceRanges: ["0.0.0.0/0"]
    allowed:
    - IPProtocol: {{ properties["IPProtocol"] }}
      ports: {{ properties["Port"] }}
```

**config.yaml**

```
imports:
- path: autonetwork.jinja
- path: firewall.jinja

resources:
- name: mynetwork
  type: autonetwork.jinja

- name: mynetwork-allow-http
  type: firewall.jinja
  properties:
    network: $(ref.mynetwork.selfLink)
    IPProtocol: TCP
    Port: [80]
```

# GCP Marketplace

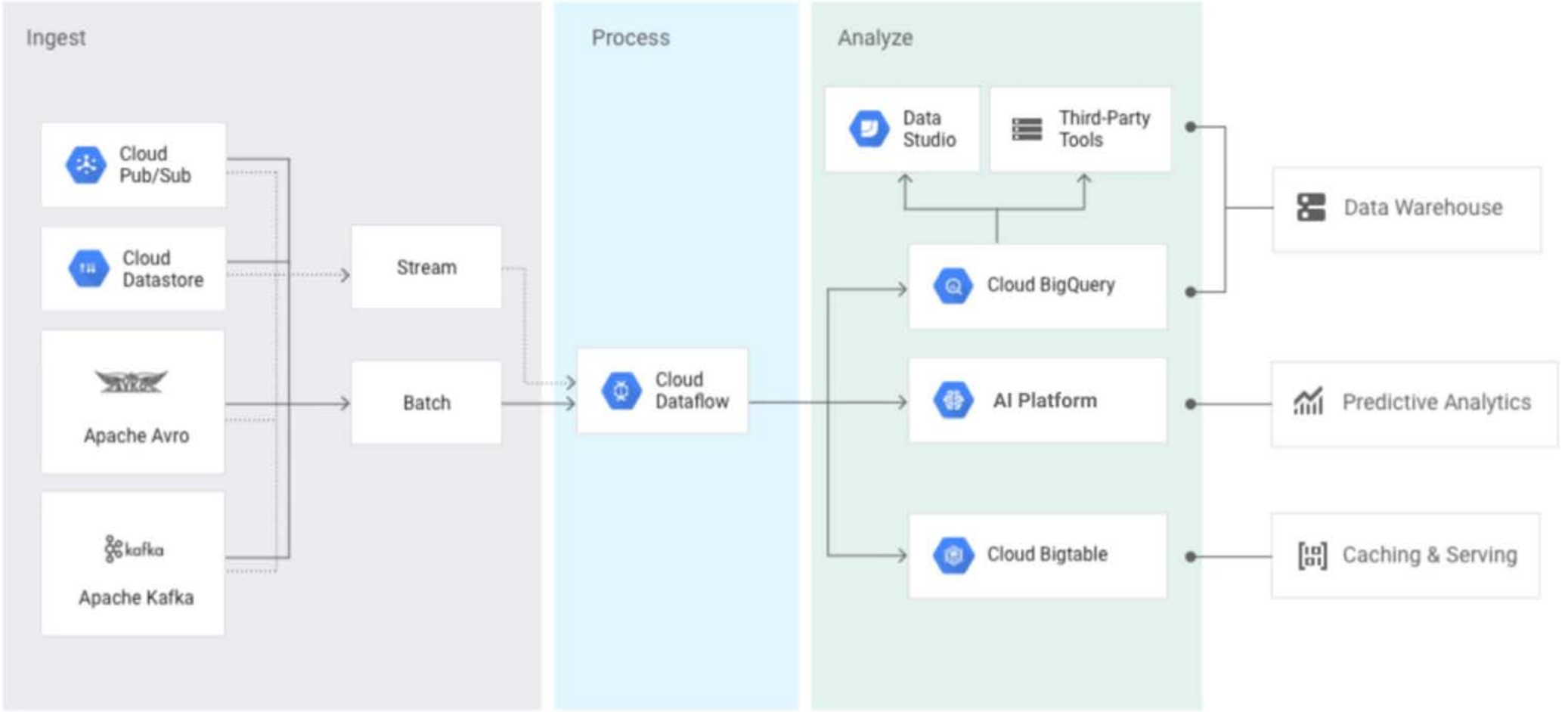
- Deploy production-grade solutions
- Single bill for GCP and third-party services
- Manage solutions using Deployment Manager
- Notifications when a security update is available
- Direct access to partner support

# BigQuery is GCP's serverless, highly scalable, and cost-effective cloud data warehouse

- Fully managed
- Petabyte scale
- SQL interface
- Very fast
- Free usage tier


BigQuery

# Data transformation with Cloud Dataflow

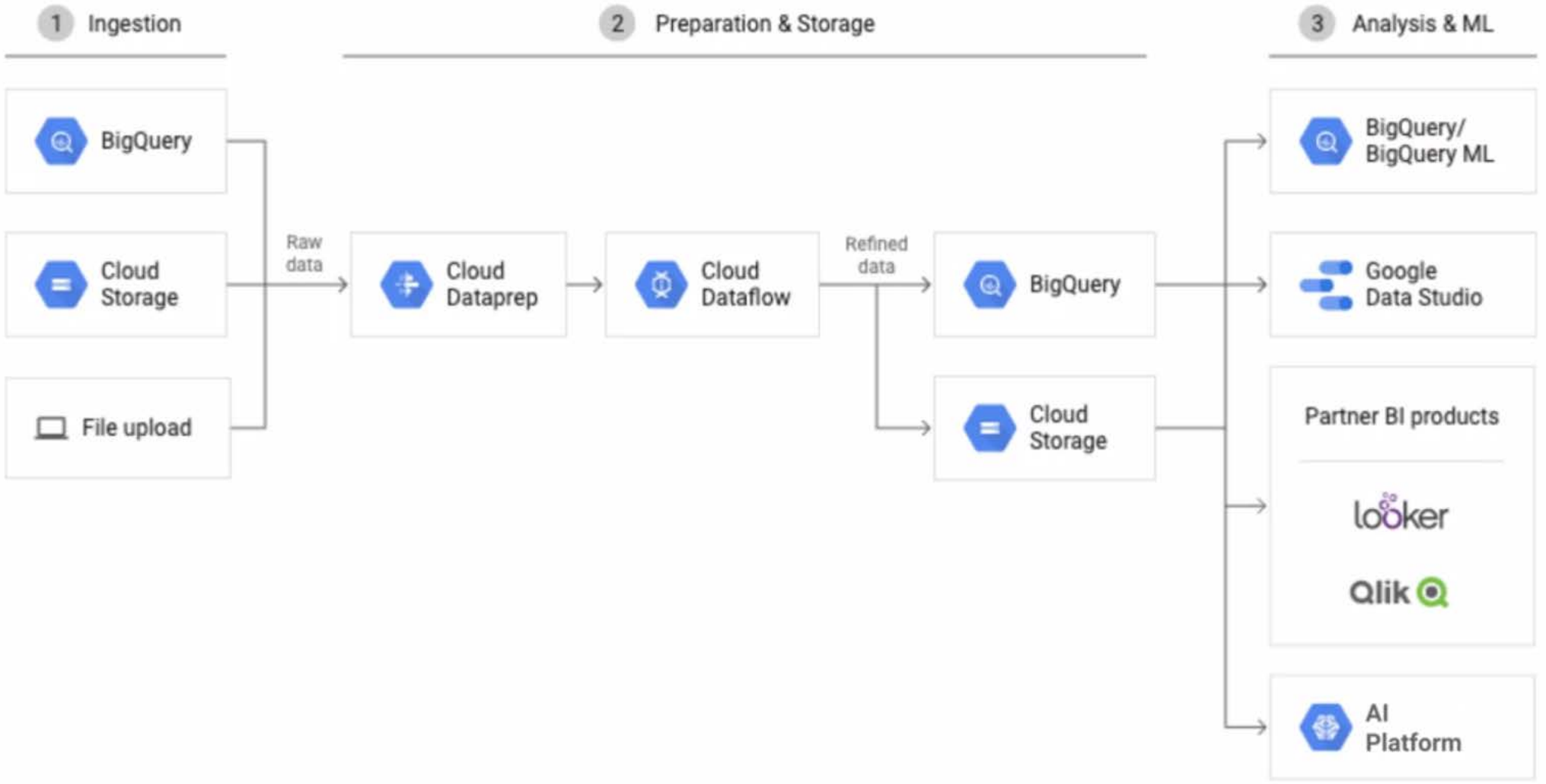


# Use Cloud Dataprep to visually explore, clean, and prepare data for analysis and machine learning

- Serverless, works at any scale
- Suggests ideal data transformation
- Focus on data analysis
- Integrated partner service operated by Trifacta

**Cloud Dataprep**

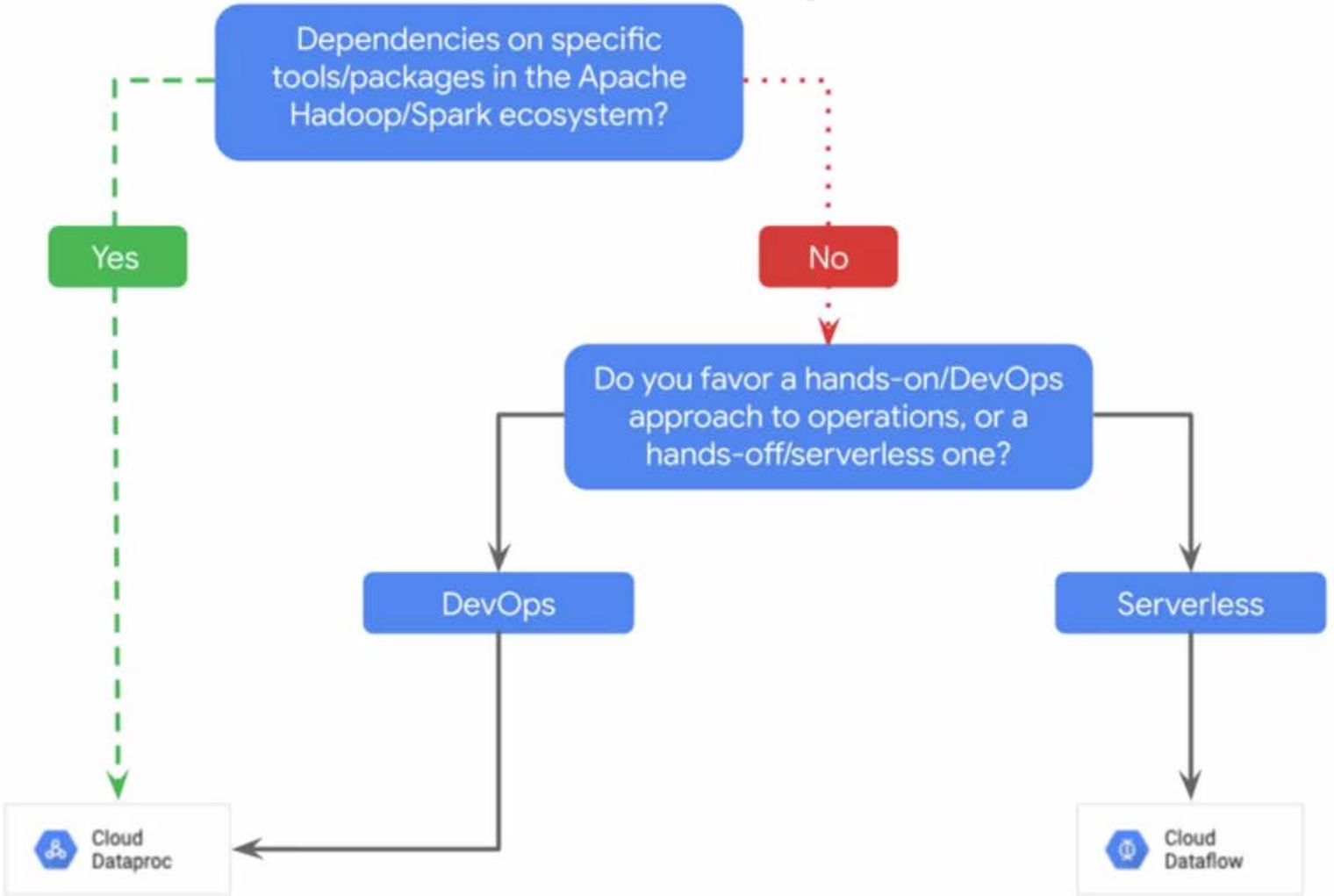
# Cloud Dataprep architecture



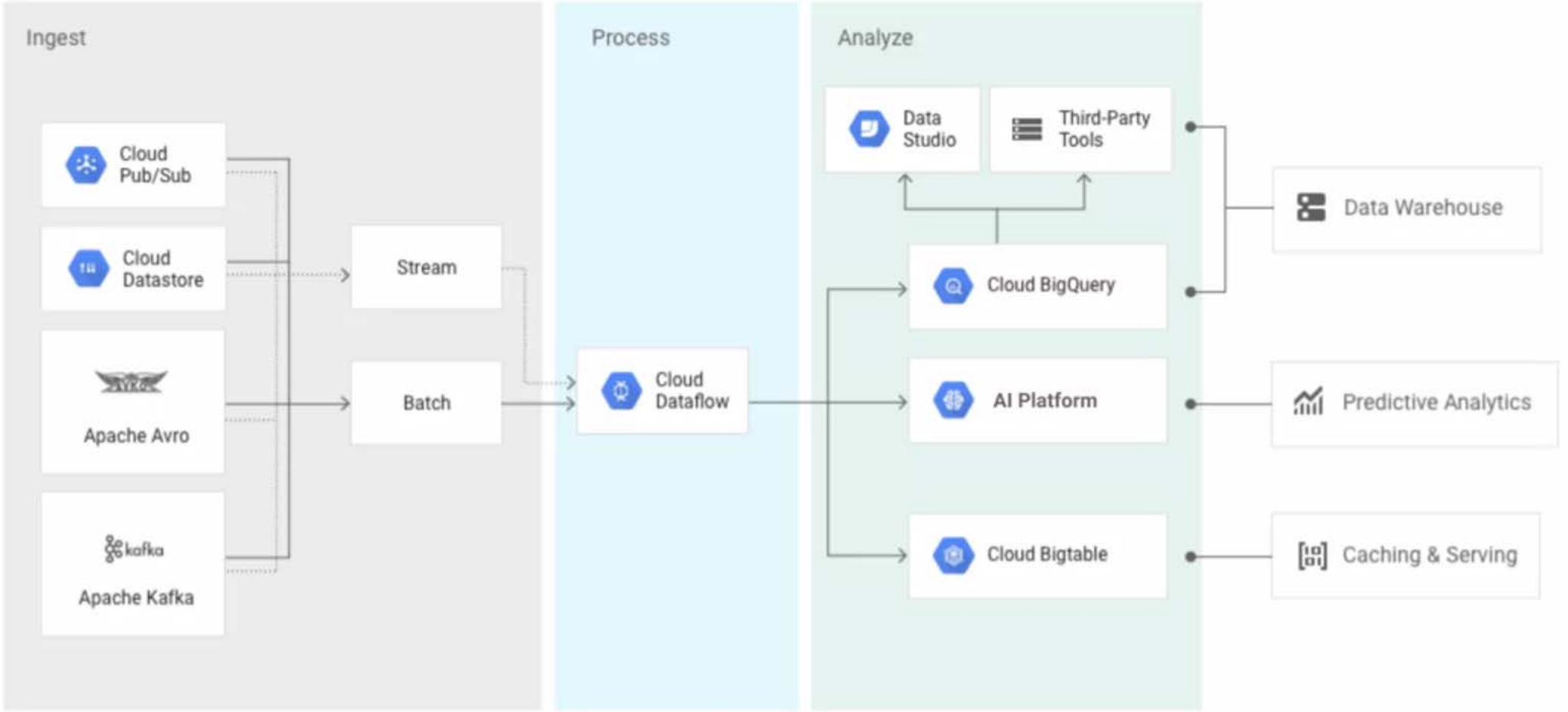
# Cloud Dataproc is a service for running Apache Spark and Apache Hadoop clusters

- Low cost (per-second, preemptible)
- Super fast to start, scale, and shut down
- Integrated with GCP
- Managed service
- Simple and familiar

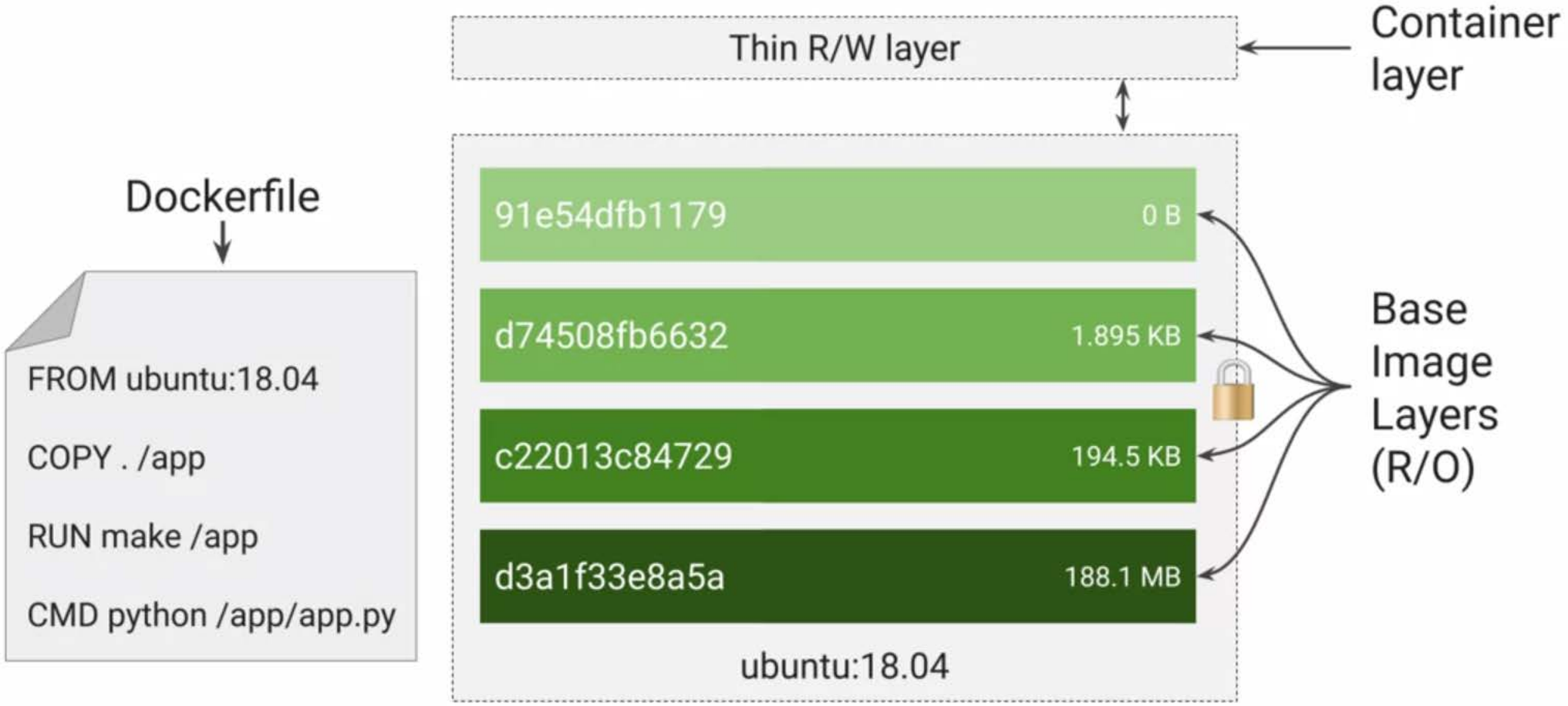
# Cloud Dataflow vs. Cloud Dataproc



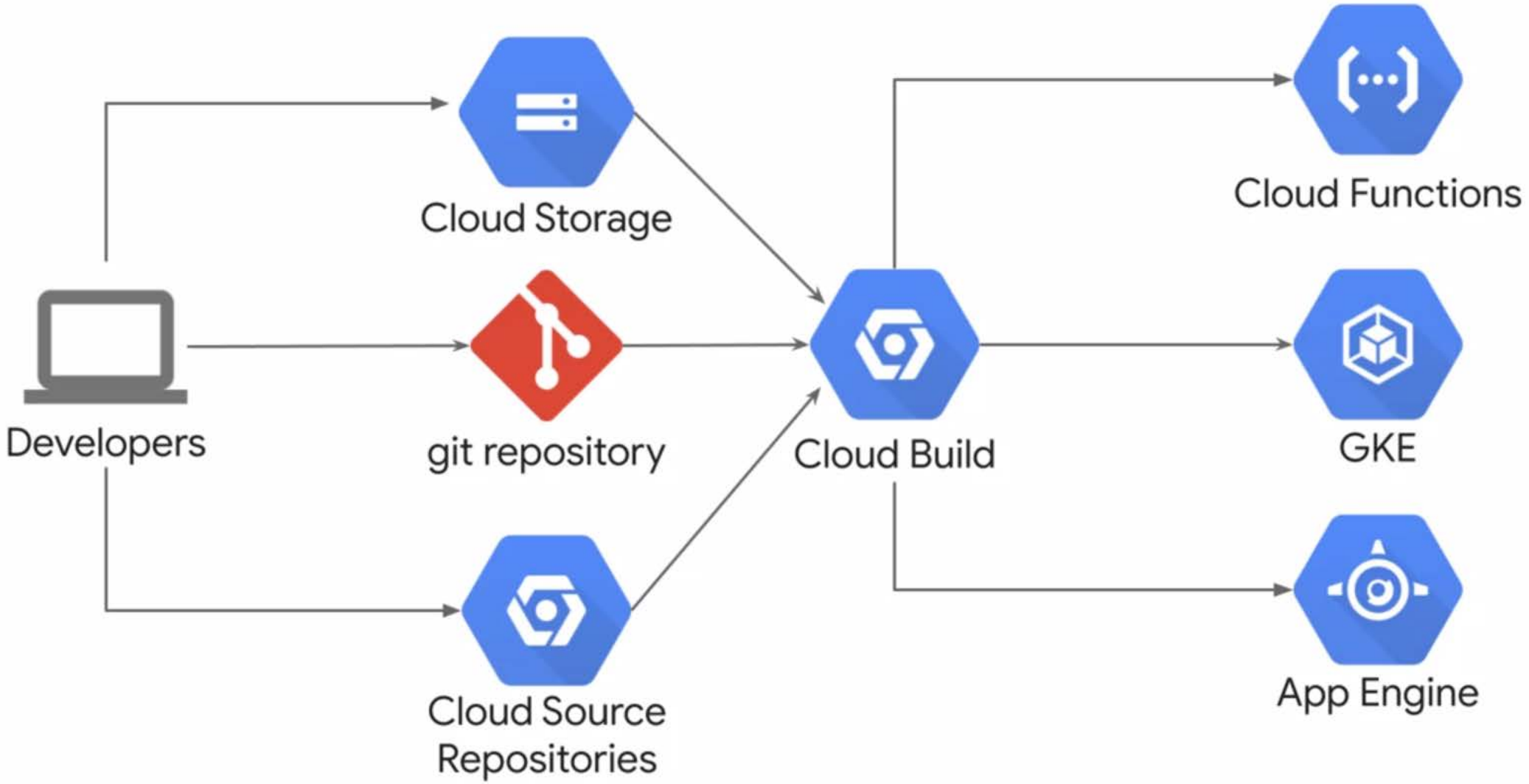
# Data transformation with Cloud Dataflow



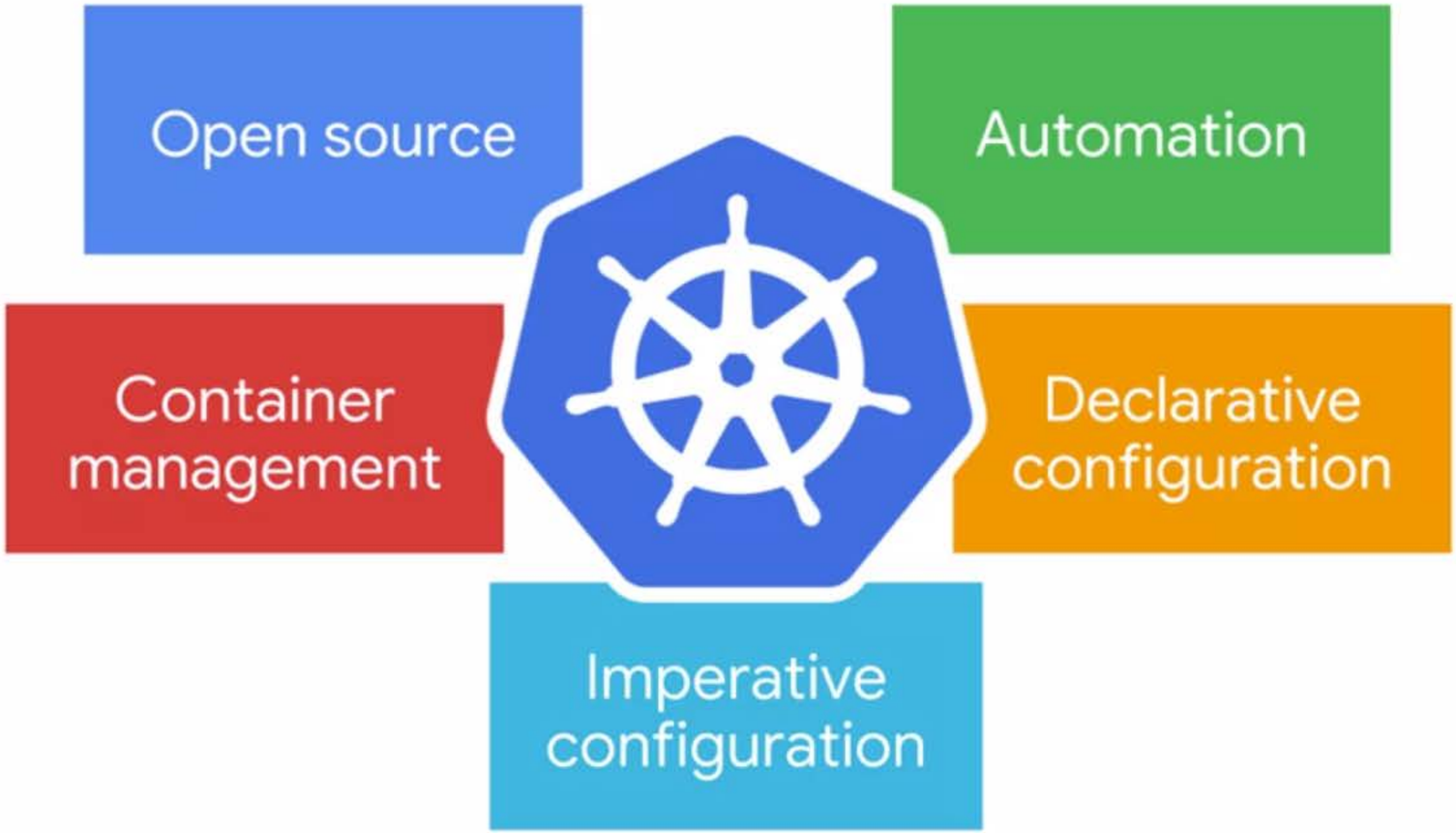
# Containers are structured in layers



# Cloud Build

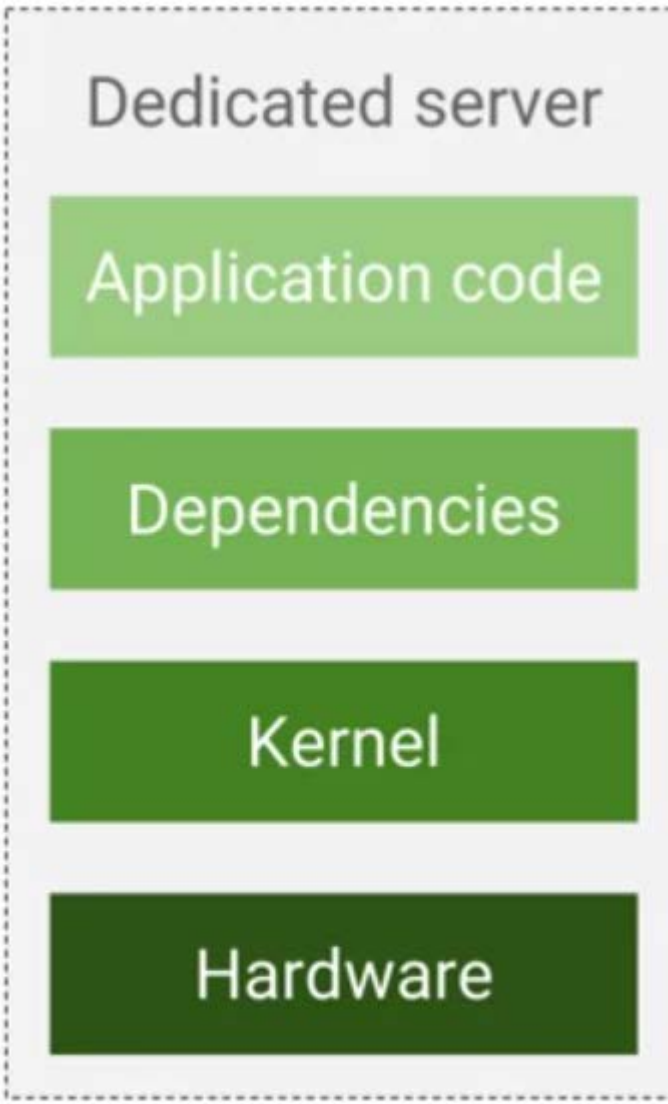

Cloud Storage
Cloud Functions
Developers
git repository
Cloud Build
GKE
Cloud Source Repositories
App Engine

# What is Kubernetes?



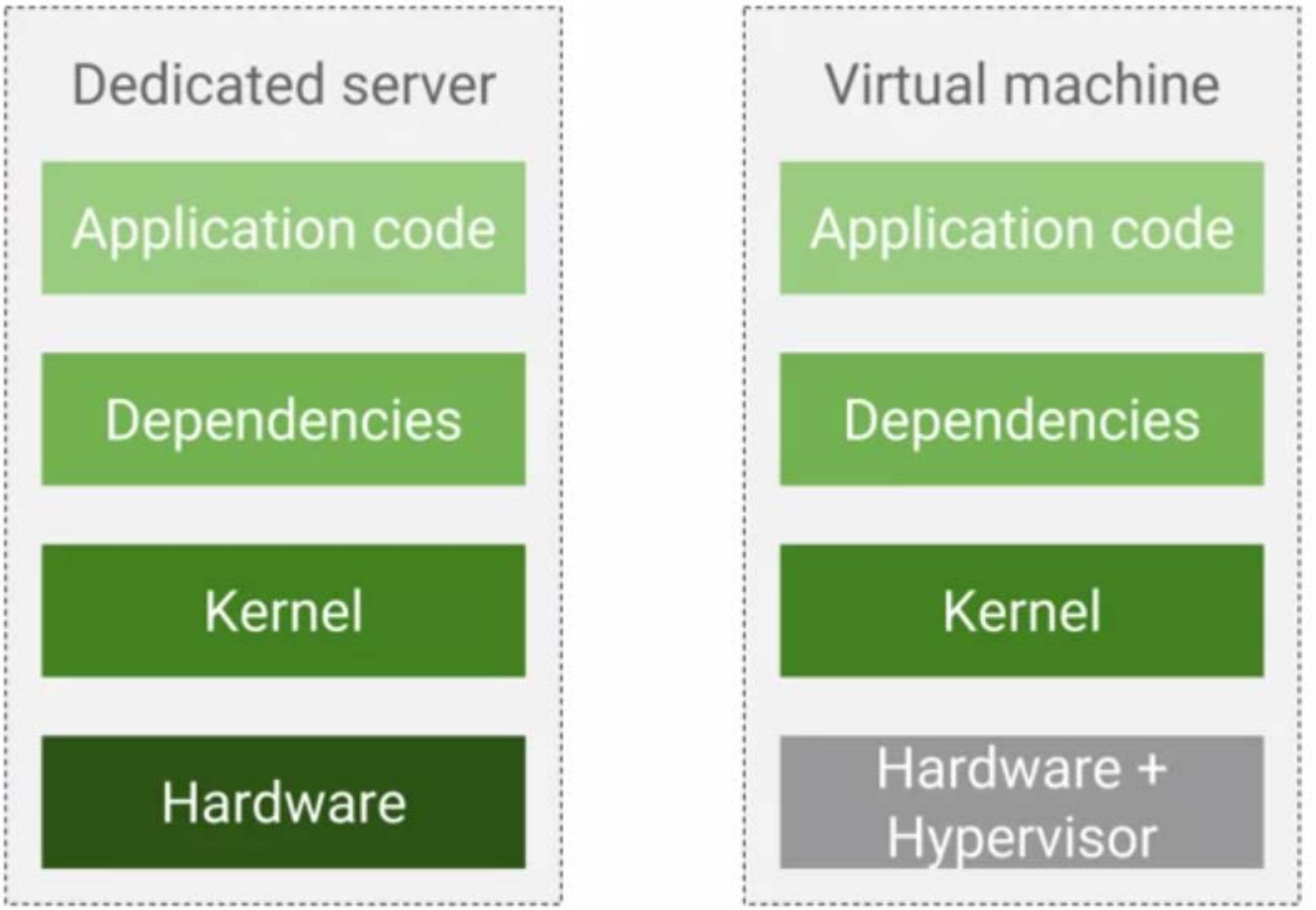
# Kubernetes features

1. Supports both stateful and stateless applications
2. Autoscaling
3. Resource limits
4. Extensibility
5. Portability




Deployment ~months
Low utilization
Not portable

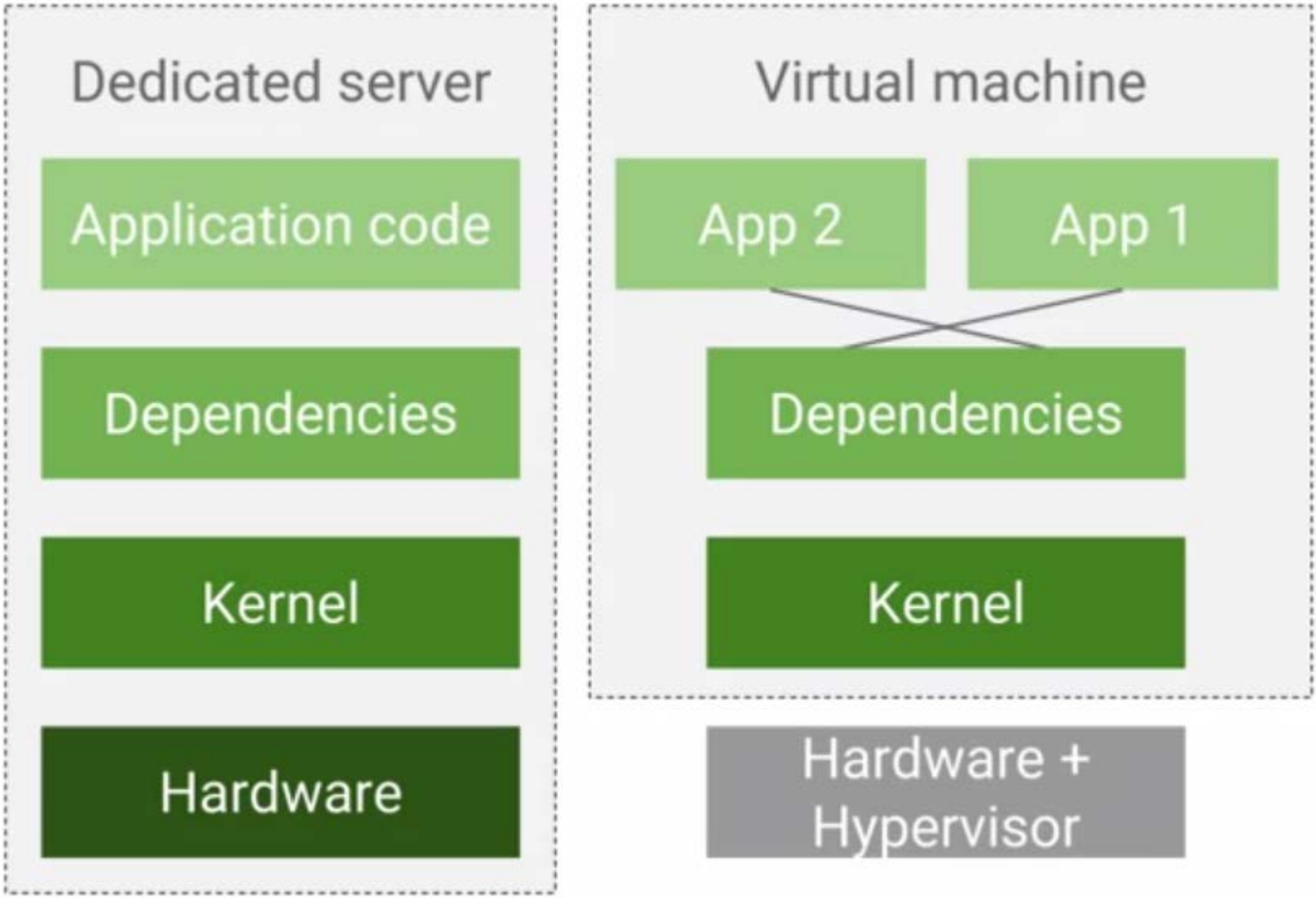
# Hypervisors create and manage virtual machines




Deployment ~months
Low utilization
Not portable

Deployment ~days (mins)
Improved utilization
Hypervisor-specific

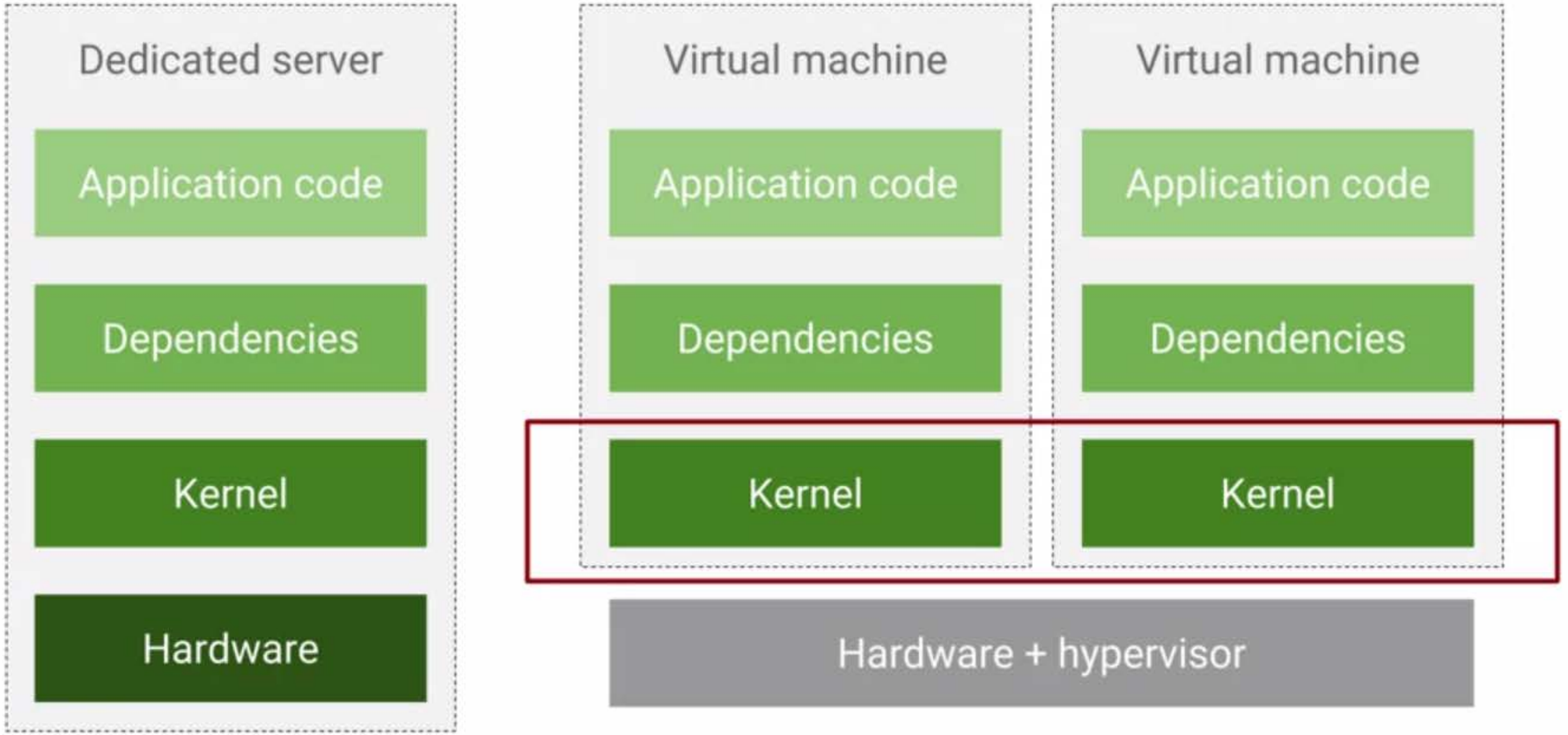
# Running multiple apps on a single VM




Deployment ~months
Low utilization
Not portable

Deployment ~days (mins)
Hypervisor-specific
**Low isolation; tied to OS**

# The VM-centric way to solve this problem

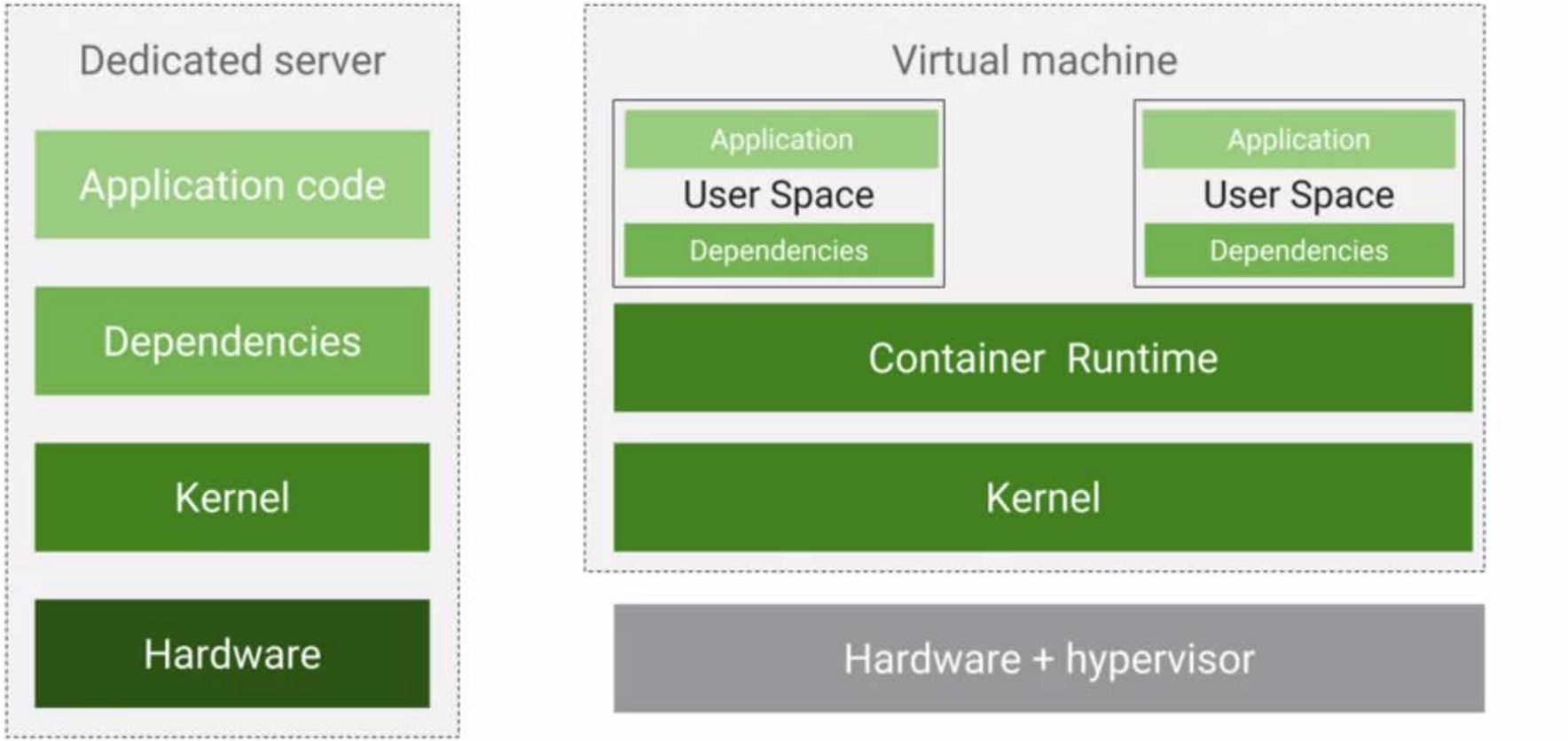



Deployment ~months
Not portable
Low utilization

Deployment ~days (mins)
Hypervisor-specific
Low isolation; tied to OS

Deployment ~days (mins)
Hypervisor-specific
**Redundant OS**

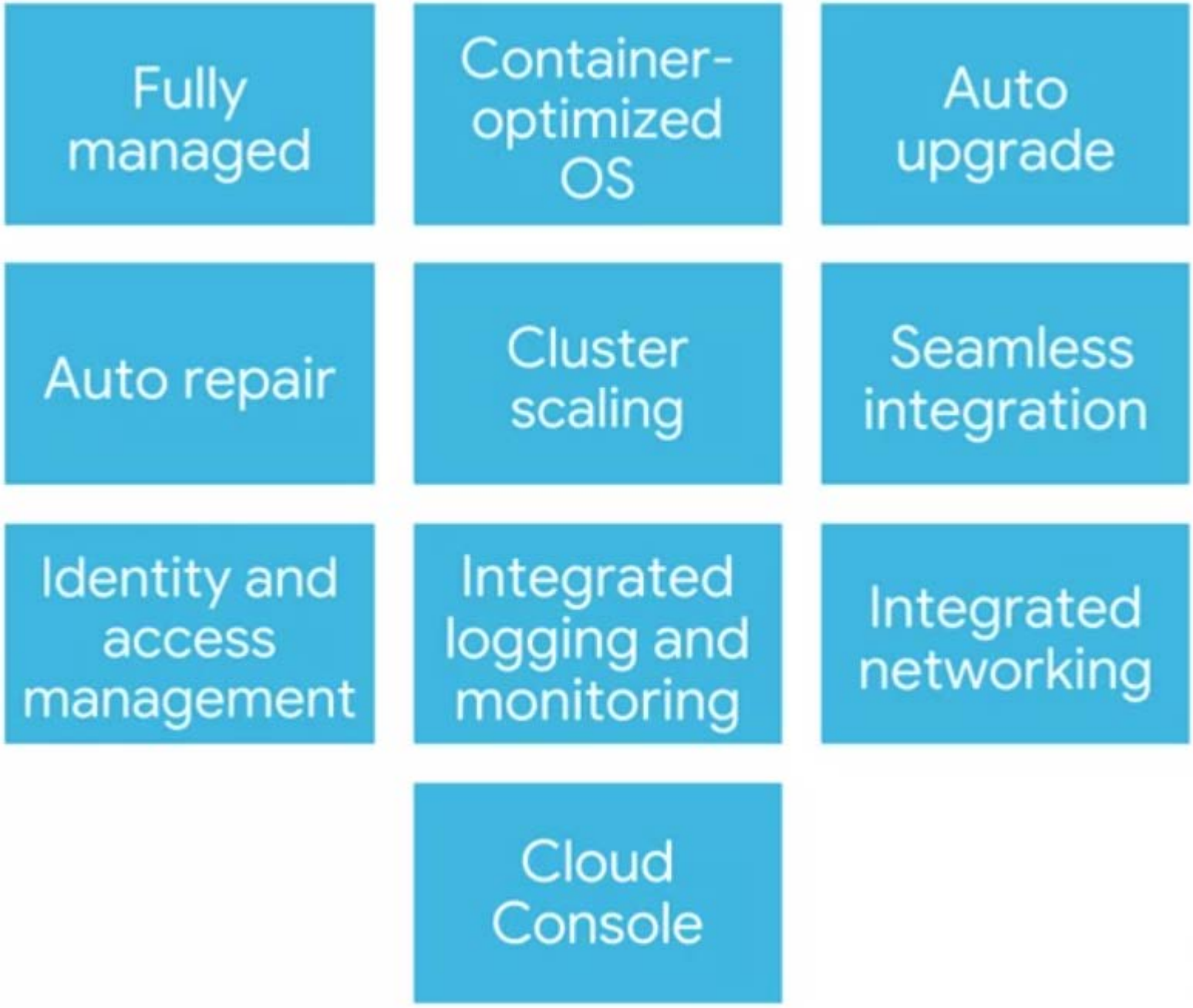
# User space abstraction and containers



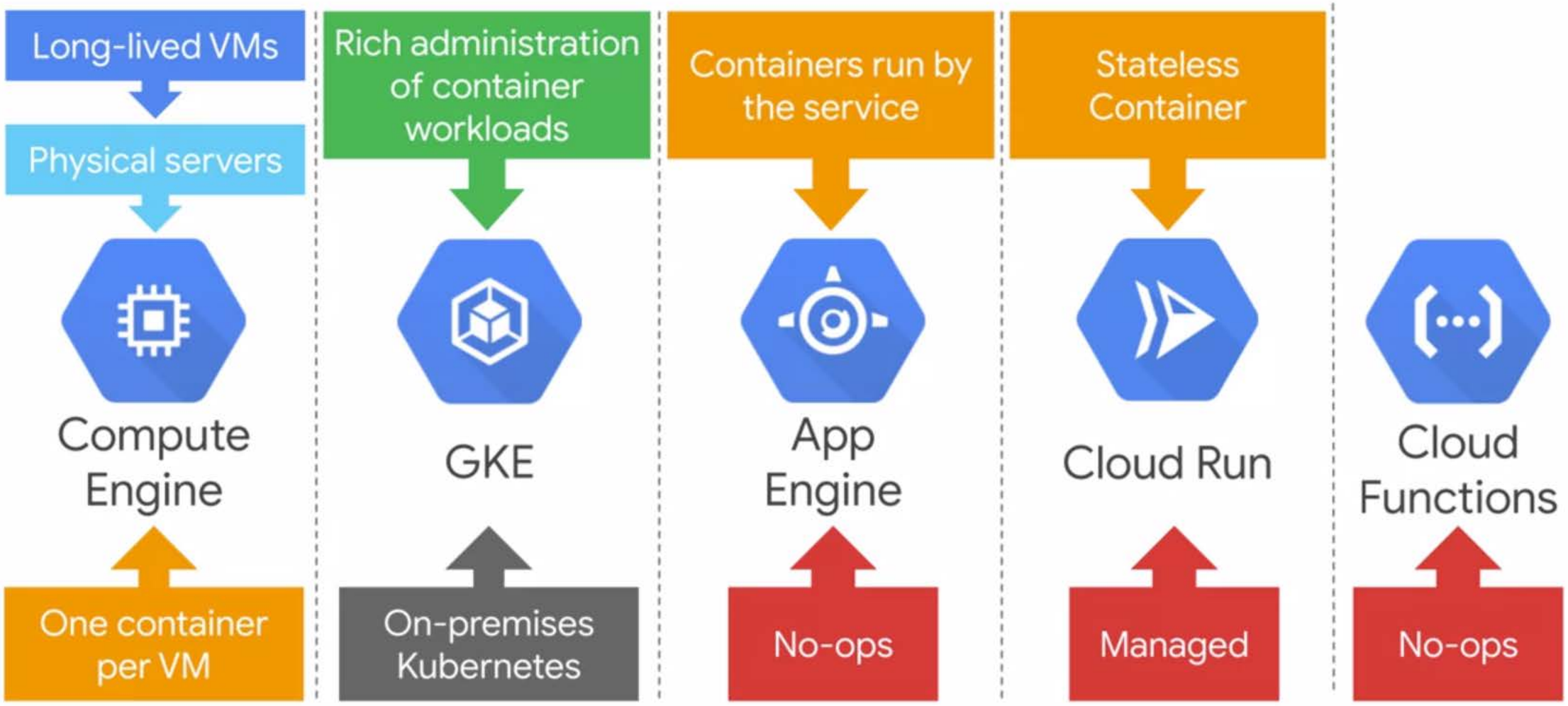

Deployment ~months
Not portable
Low utilization

Deployment ~days (mins)
Hypervisor-specific
Low isolation; tied to OS

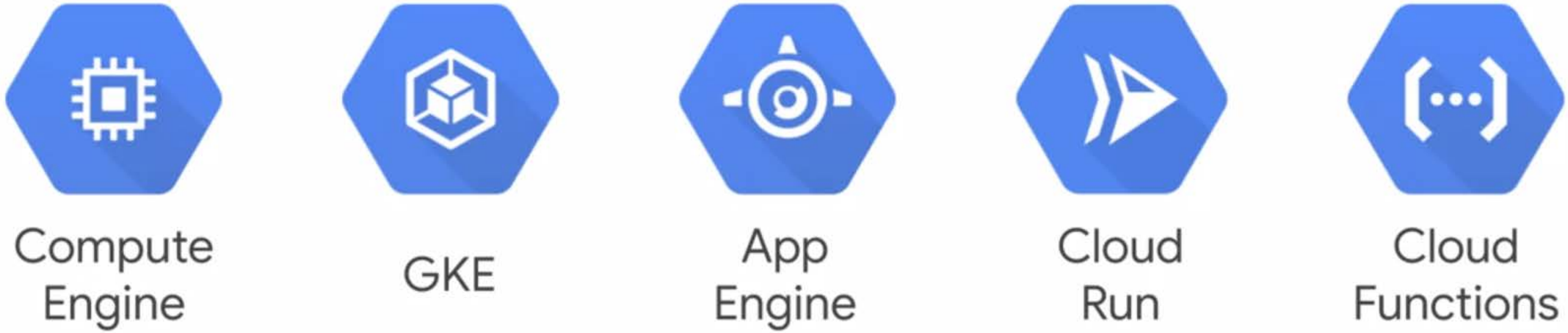
Deployment ~days (mins)
Hypervisor-specific
**Redundant OS**


Fully managed
Container-optimized OS
Auto upgrade
Auto repair
Cluster scaling
Seamless integration
Identity and access management
Integrated logging and monitoring
Integrated networking
Cloud Console

# So how to decide?



# Comparing Google Cloud computing solutions



# Compute Engine

- Fully customizable virtual machines
- Persistent disks and optional local SSDs
- Global load balancing and autoscaling
- Per-second billing

# App Engine

- Provides a fully managed, code-first platform.
- Streamlines application deployment and scalability.
- Provides support for popular programming languages and application runtimes.
- Supports integrated monitoring, logging, and diagnostics.
- Simplifies version control, canary testing, and rollbacks.

# Google Kubernetes Engine

- Fully managed Kubernetes platform.
- Supports cluster scaling, persistent disks, automated upgrades, and auto node repairs.
- Built-in integration with Google Cloud services.
- Portability across multiple environments
  - Hybrid computing
  - Multi-cloud computing

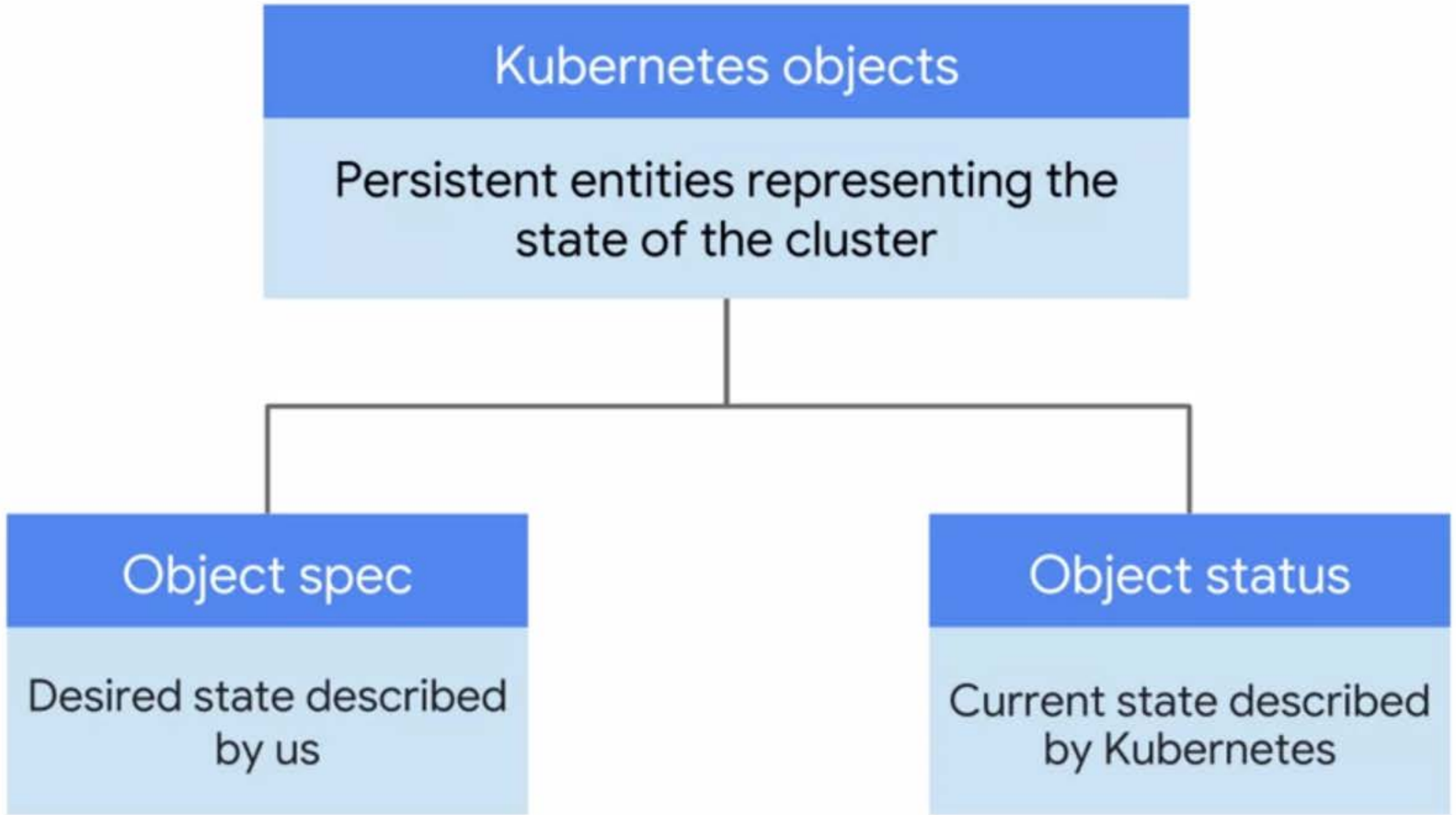
# Cloud Run

- Enables stateless containers.
- Abstracts away infrastructure management.
- Automatically scales up and down.
- Open API and runtime environment.

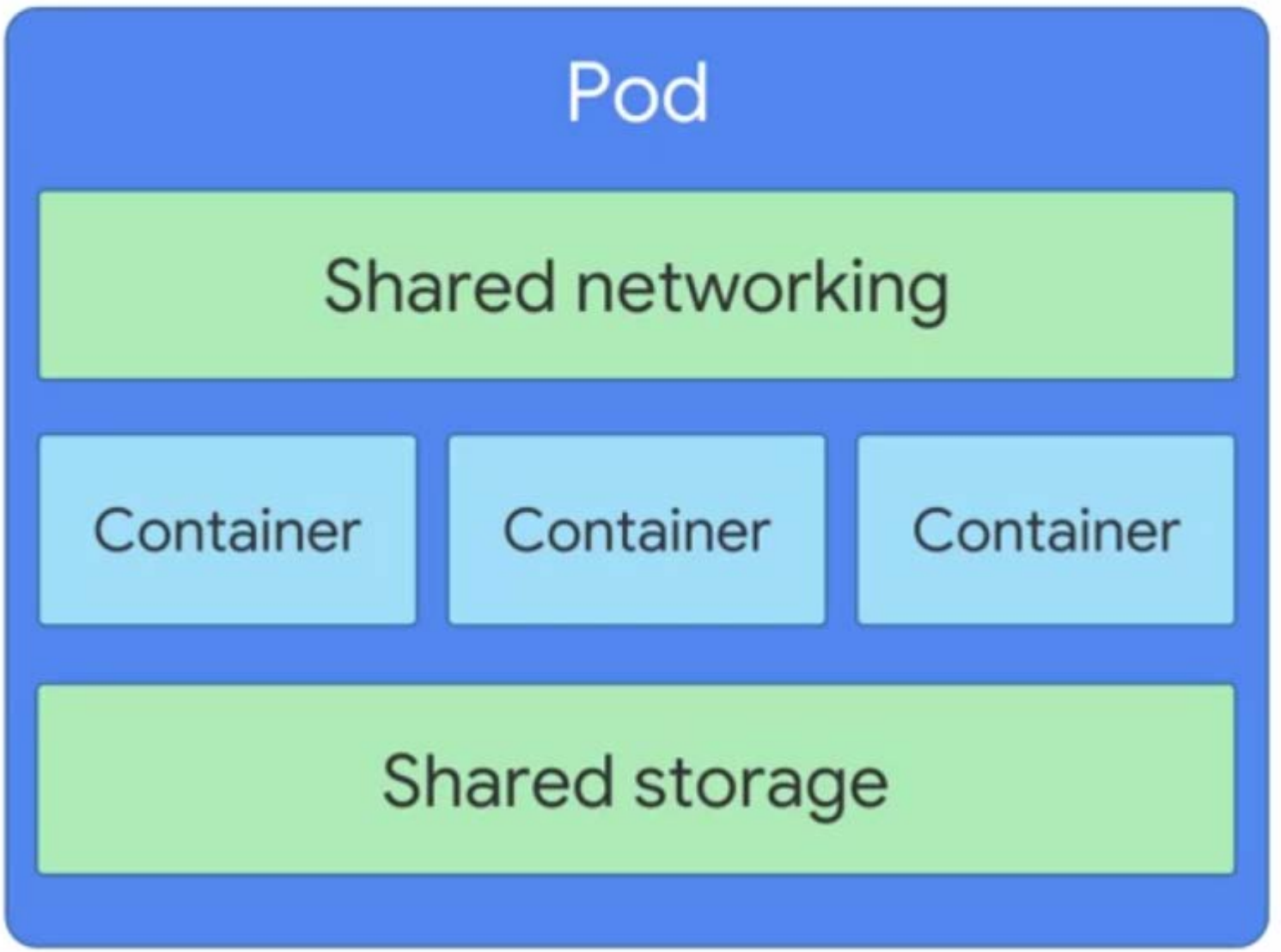
# Cloud Functions

- Event-driven, serverless compute service.
- Automatic scaling with highly available and fault-tolerant design.
- Charges apply only when your code runs.
- Triggered based on events in Google Cloud services, HTTP endpoints, and Firebase.

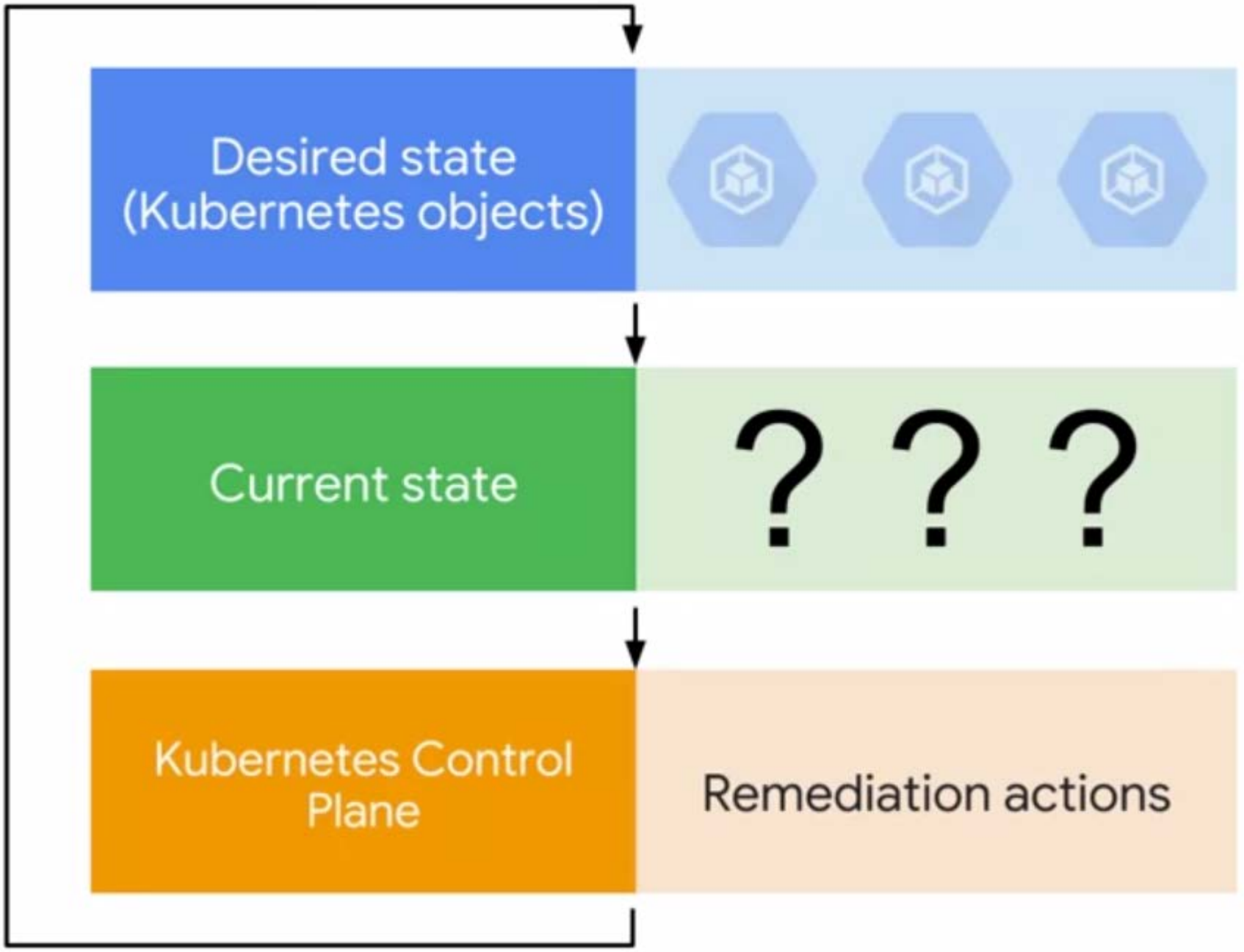
# There are two elements to Kubernetes objects

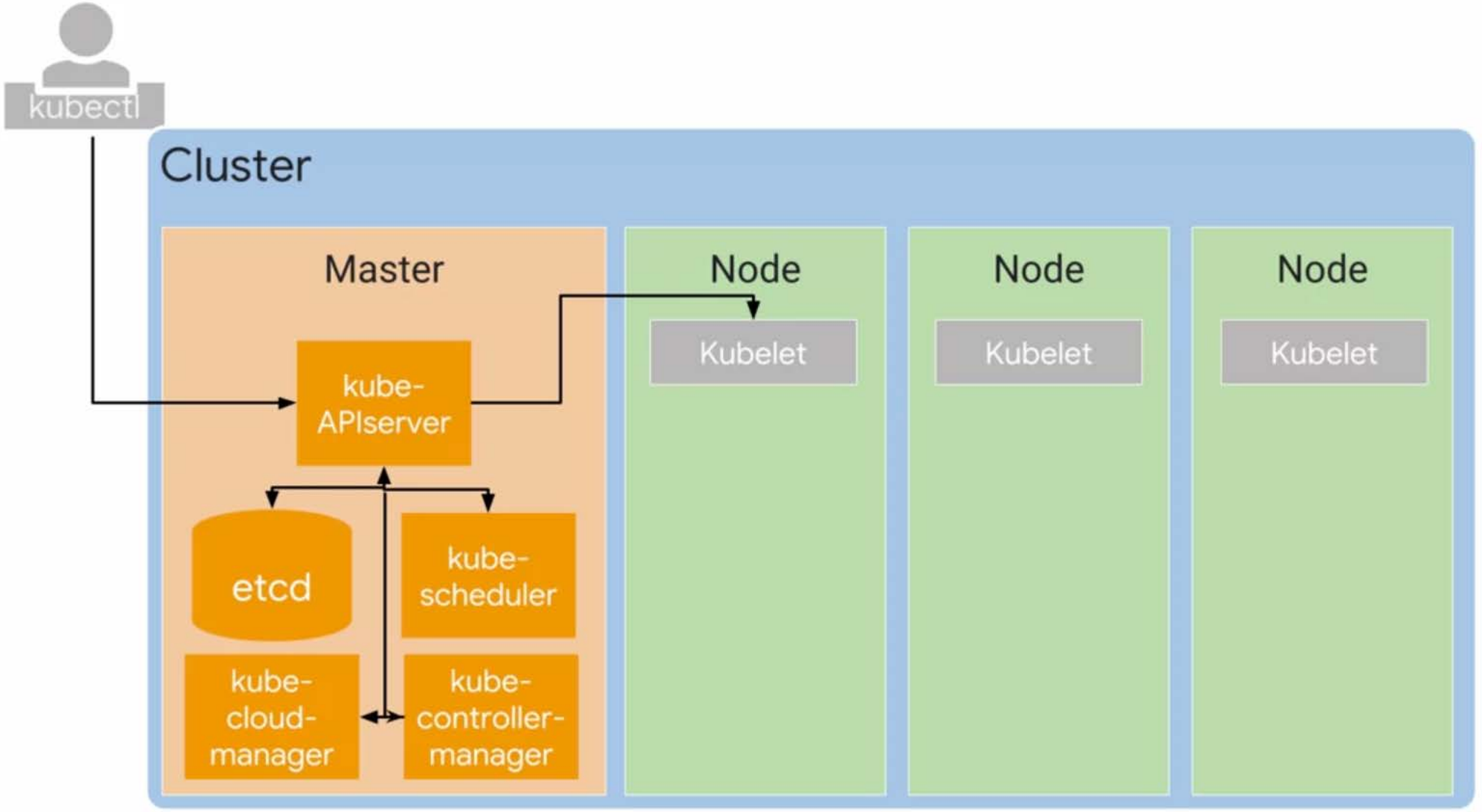


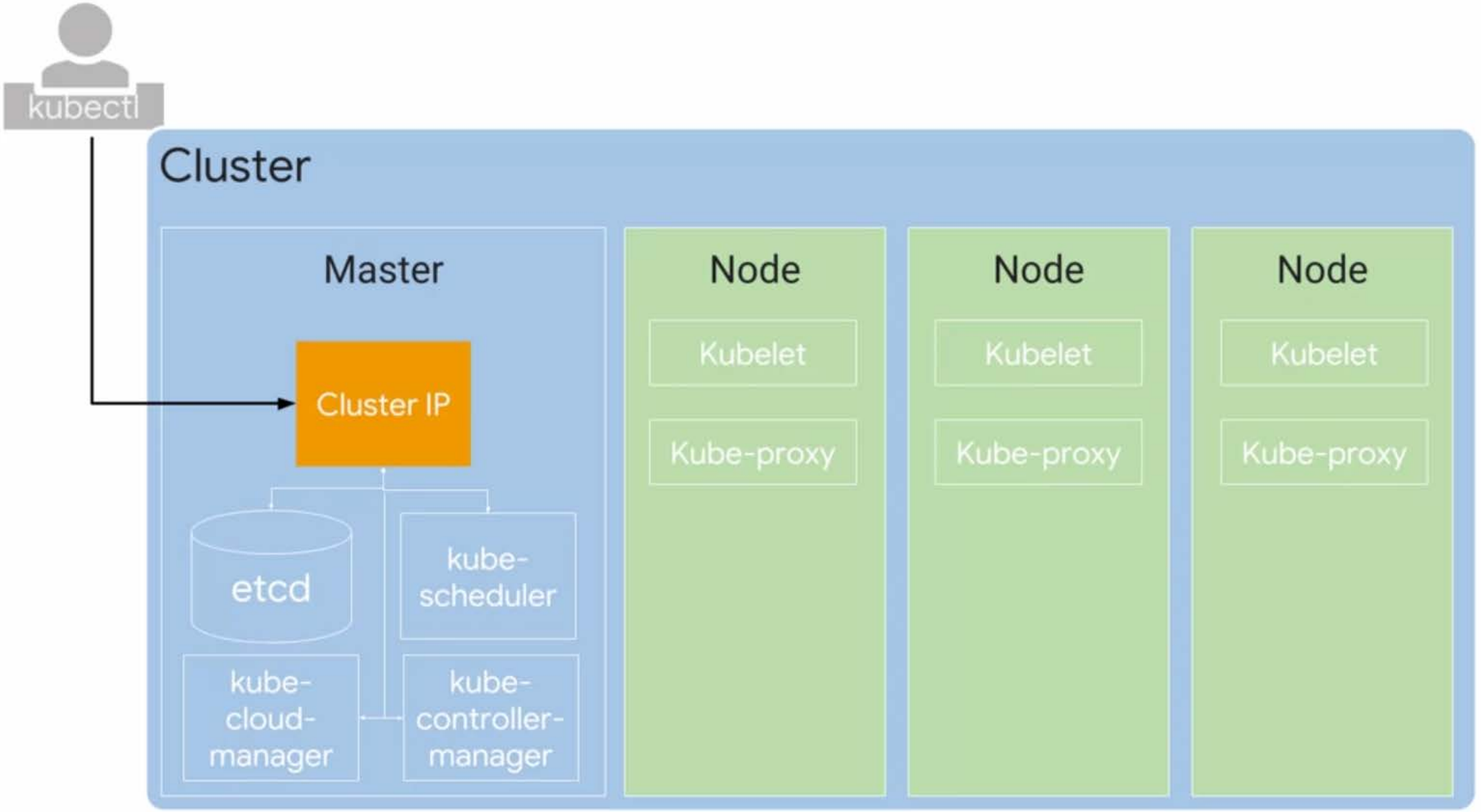
# Containers in a Pod share resources



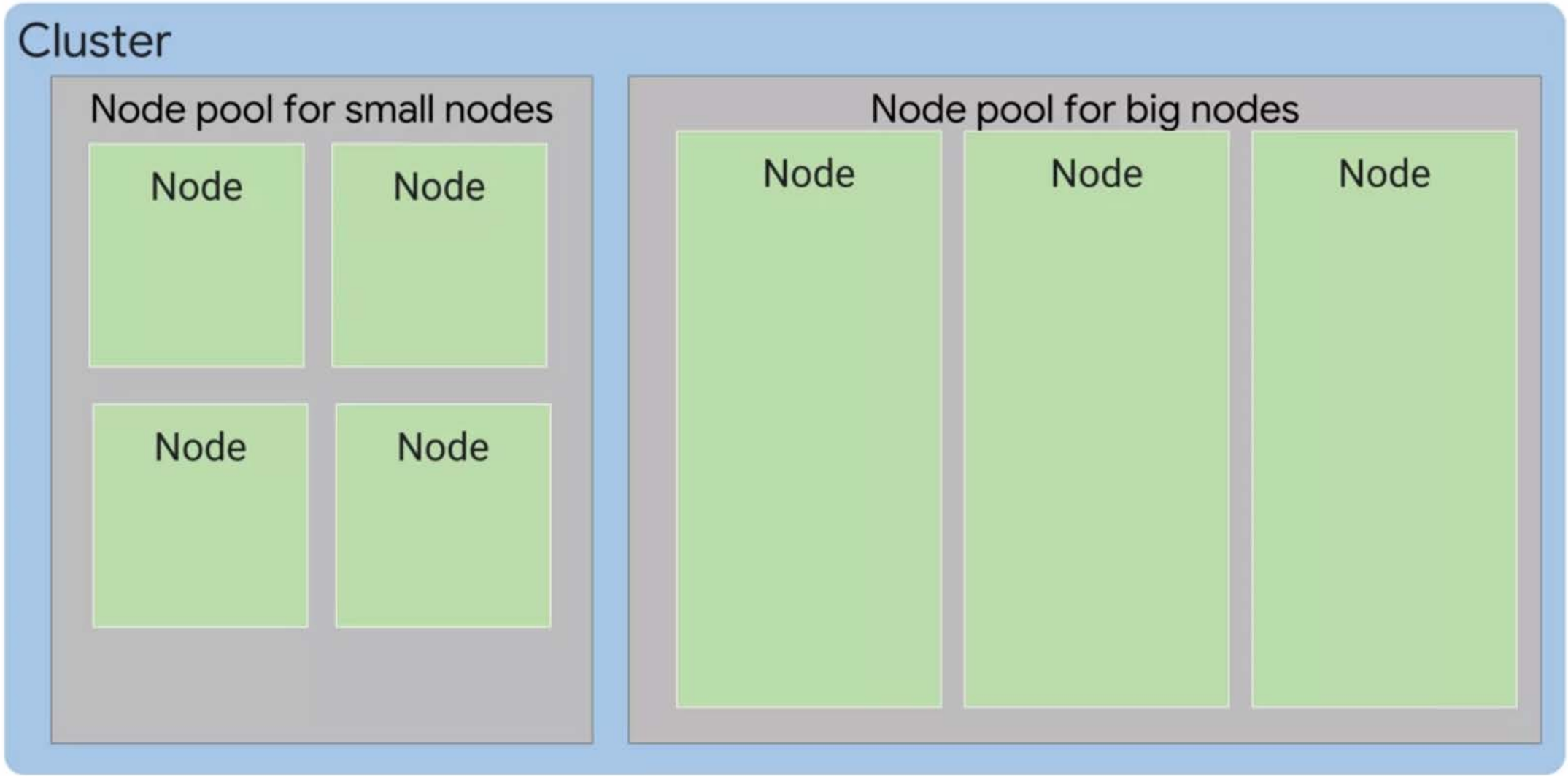
# Example: Desired state compared to current state



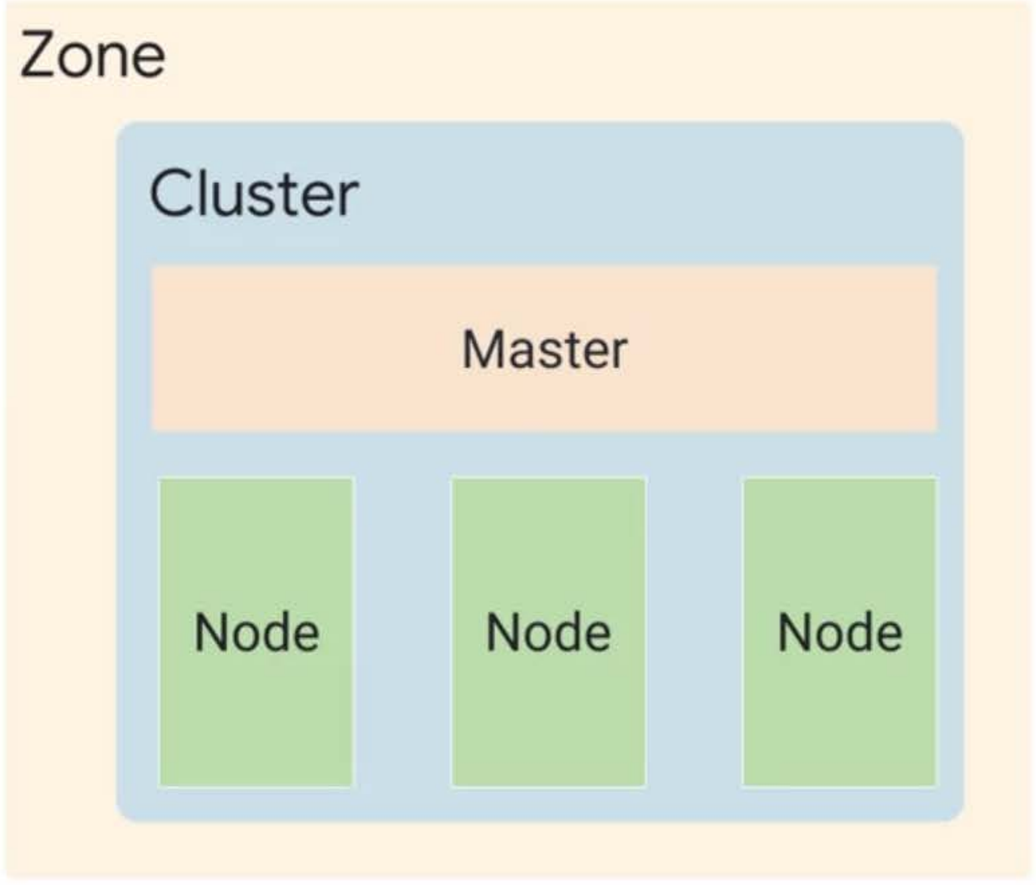

kubectl
Cluster
Master
kube-
APIserver
etcd
kube-
scheduler
kube-
cloud-
manager
kube-
controller-
manager
Node
Kubelet
Node
Kubelet
Node
Kubelet


kubectl
Cluster
Master
Cluster IP
etcd
kube-scheduler
kube-cloud-manager
kube-controller-manager
Node
Kubelet
Kube-proxy
Node
Kubelet
Kube-proxy
Node
Kubelet
Kube-proxy

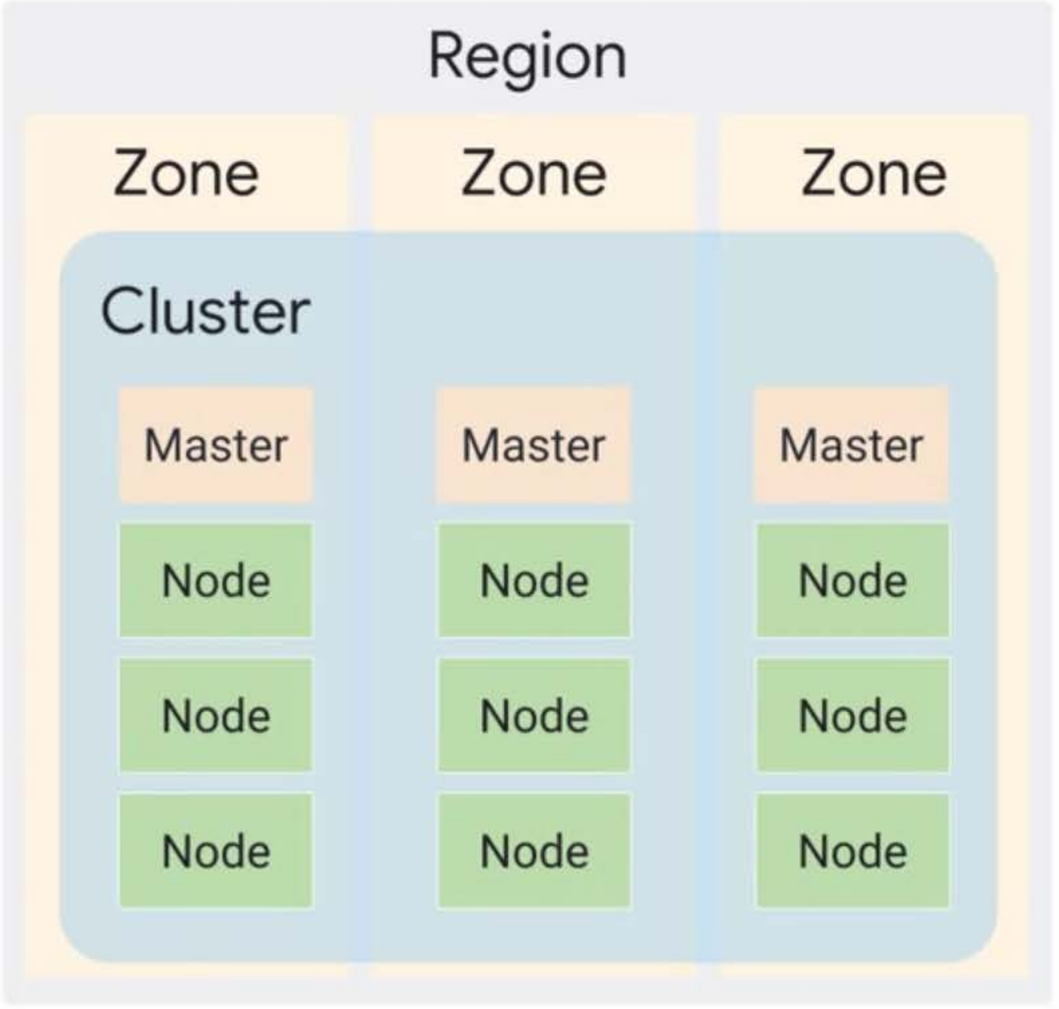
# Use node pools to manage different kinds of nodes



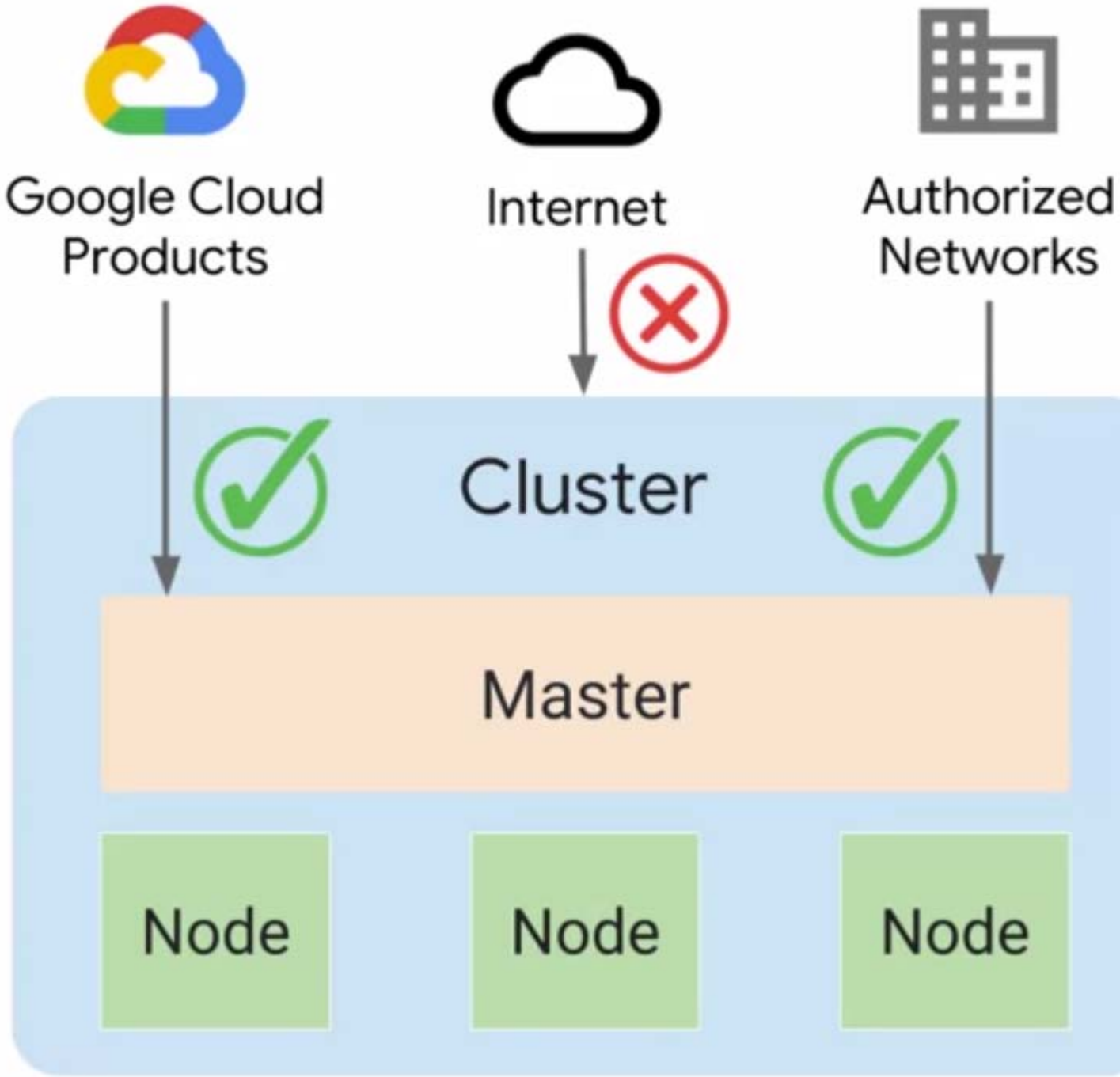
## Zonal cluster




## Regional cluster



# Private cluster



# Objects are defined in a YAML file

```yaml
apiVersion: apps/v1
kind: Pod
metadata:
   name: nginx
   labels:
      app: nginx
spec:
   containers:
   - name:  nginx
     image: nginx:latest
```

# Object names

All objects are identified by a name.

```
apiVersion: apps/v1
kind: Pod
metadata:
   name: nginx
[...]
```

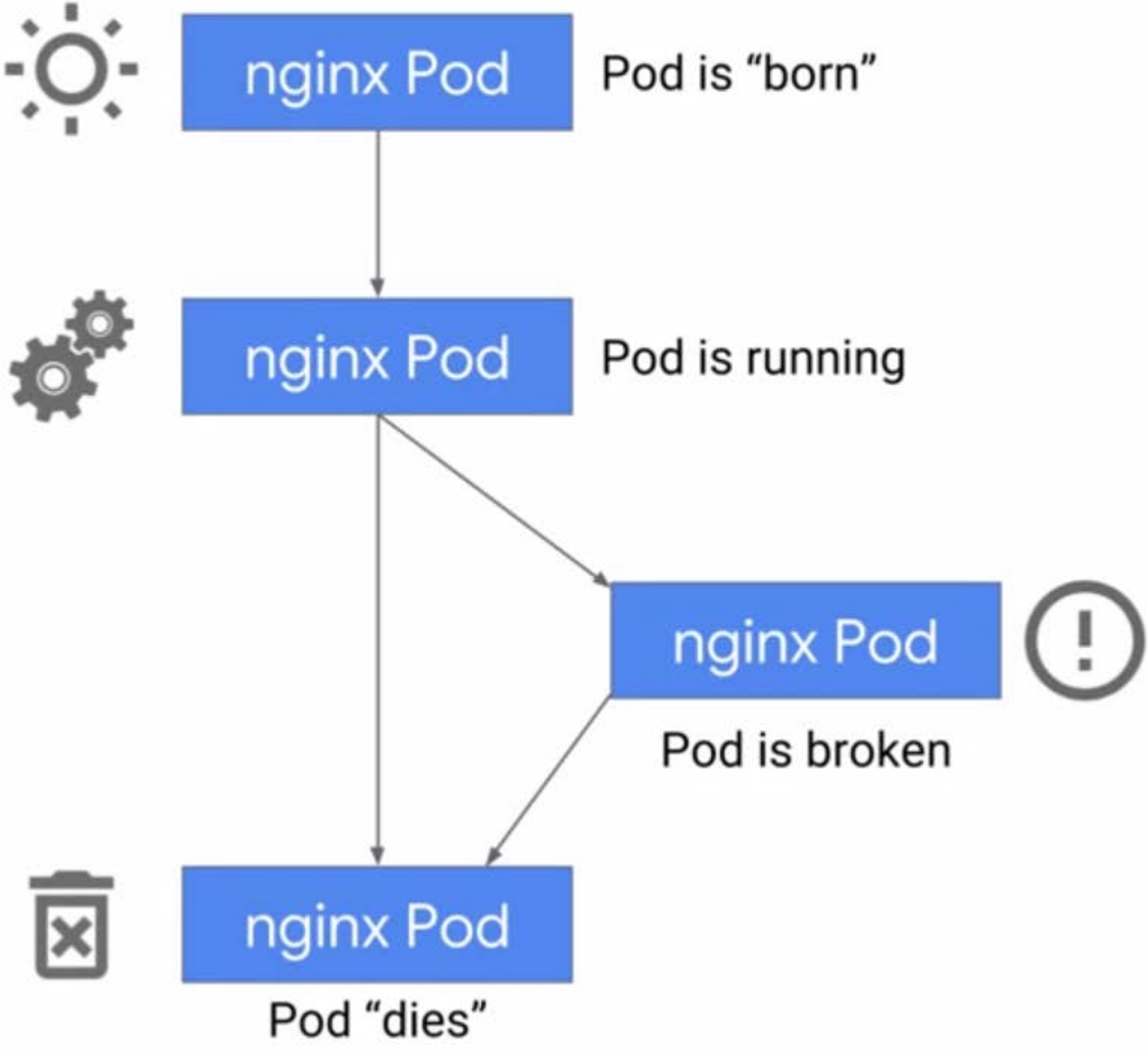
```
apiVersion: apps/v1
kind: Pod
metadata:
   name: nginx
[...]
```

Cannot have two of the same object types with same names

# Labels

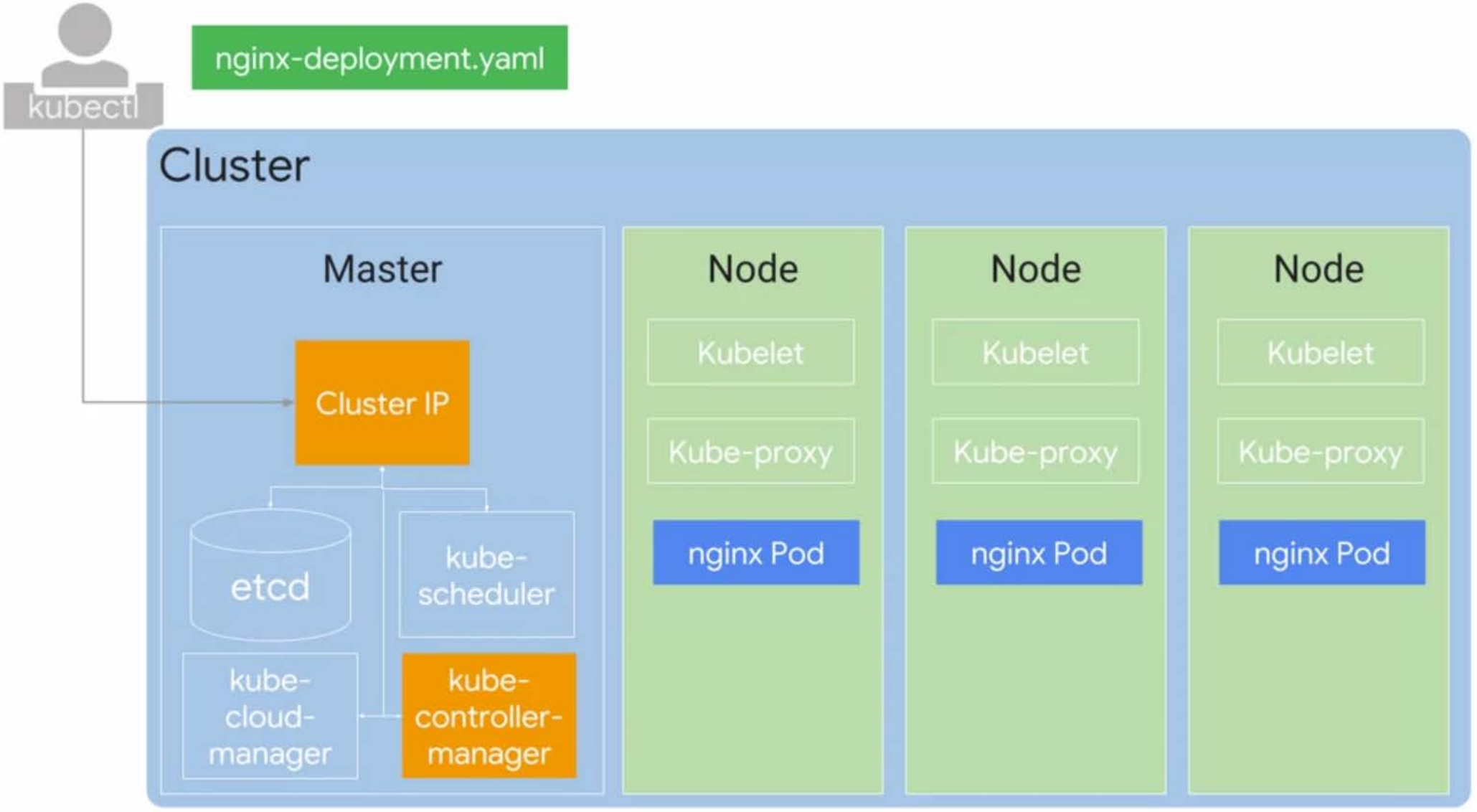
```
apiVersion: apps/v1
kind: Deployment
metadata:
   name: nginx
   labels:
      app: nginx
      env: dev
      stack: frontend
spec:
   replicas: 3
   selector:
      matchLabels
      app: nginx
```

# Pods have a life cycle

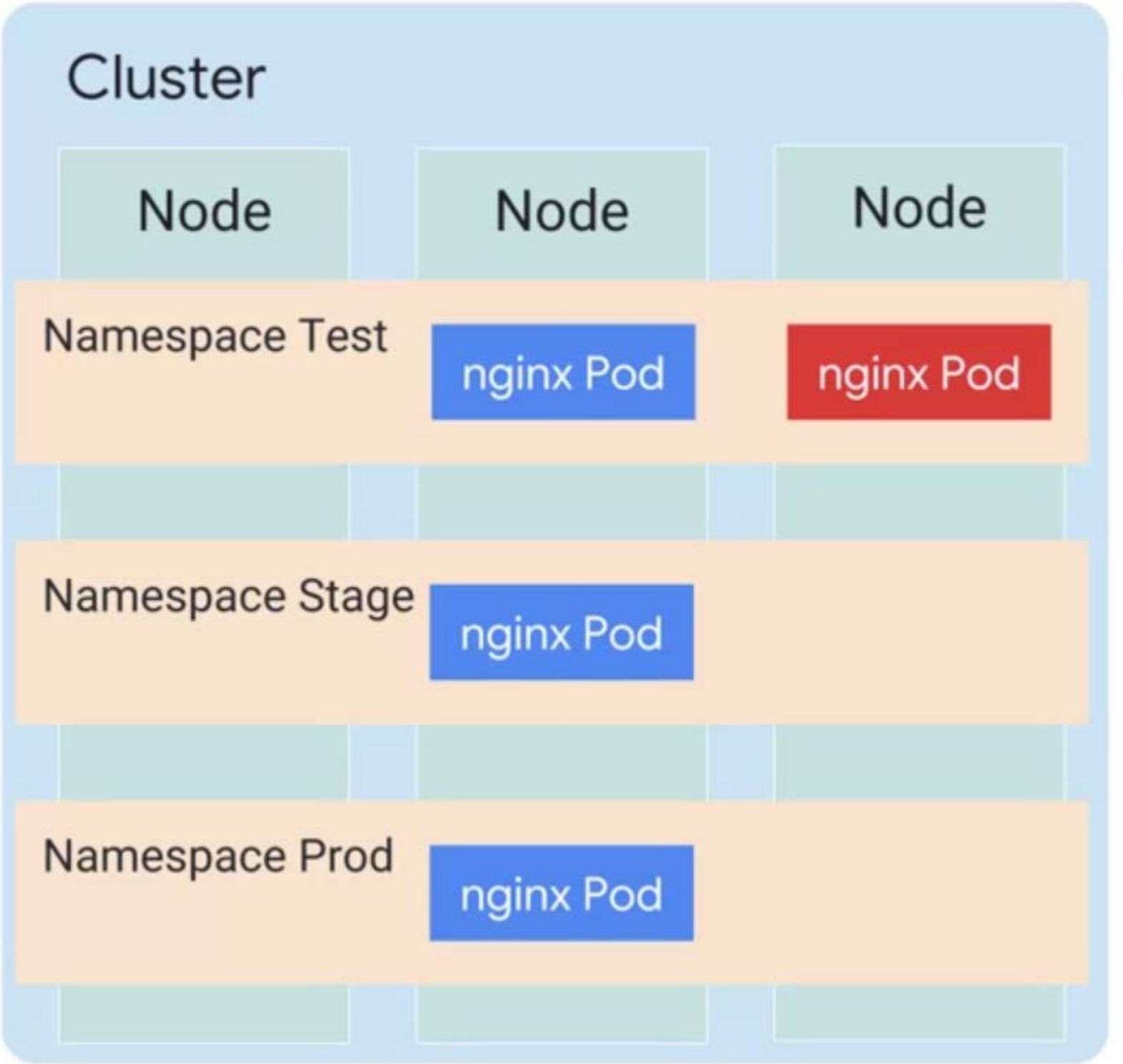


# Pods and Controller Objects


| Controller object types | • Deployment<br>• StatefulSet<br>• DaemonSet<br>• Job |
| --- | --- |


kubectl
nginx-deployment.yaml
Cluster
Master
Cluster IP
etcd
kube-scheduler
kube-cloud-manager
kube-controller-manager
Node
Kubelet
Kube-proxy
nginx Pod
Node
Kubelet
Kube-proxy
nginx Pod
Node
Kubelet
Kube-proxy
nginx Pod

# Namespaces


Cluster
Node
Node
Node
Namespace Test
nginx Pod
nginx Pod
Namespace Stage
nginx Pod
Namespace Prod
nginx Pod

# Best practice tip: namespace-neutral YAML

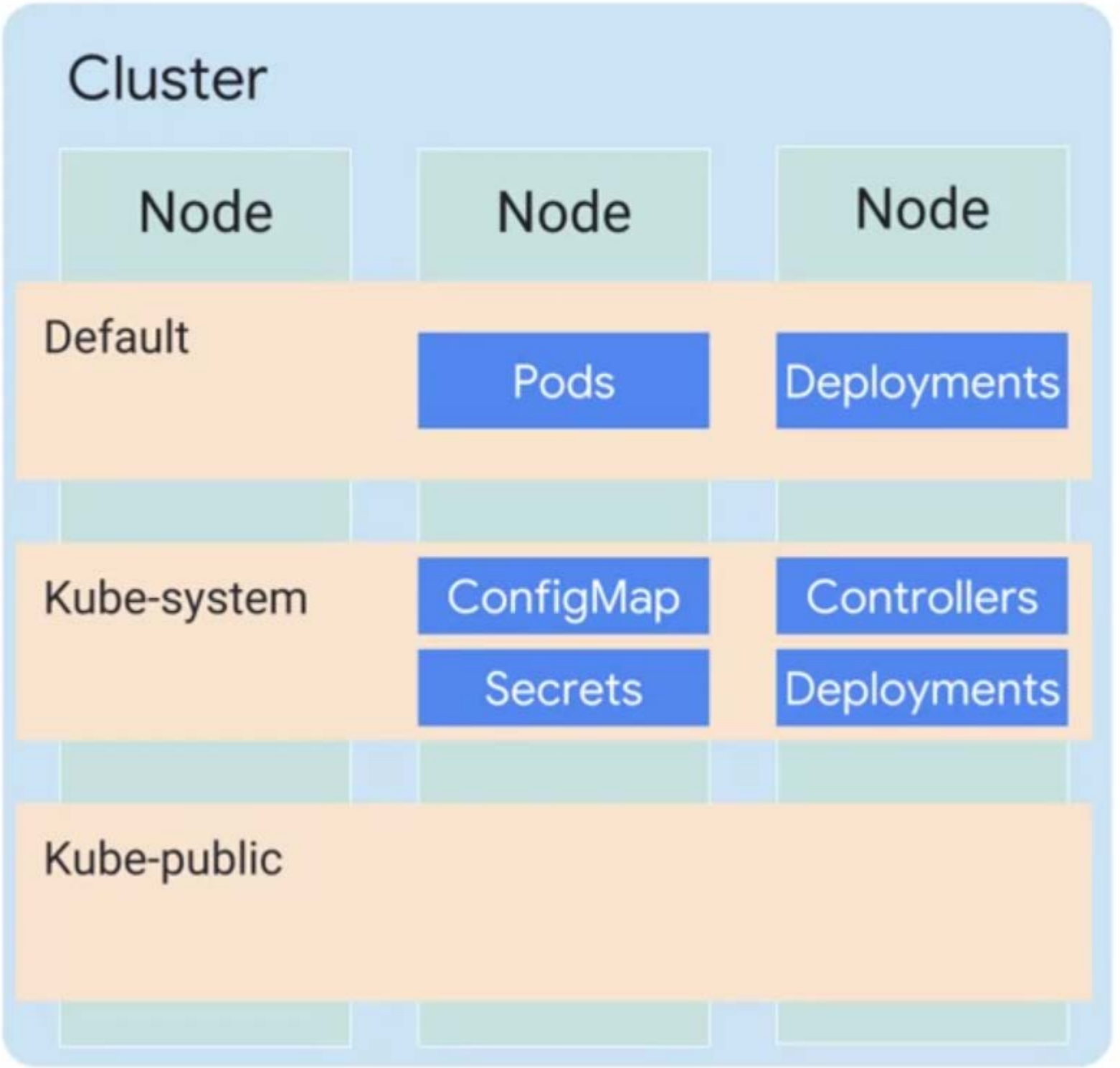
Most flexible:

```
kubectl -n demo apply -f mypod.yaml
```

Legal but less flexible:

```
apiVersion: v1
kind: Pod
metadata:
   name: mypod
   namespaces: demo
```

# Namespaces

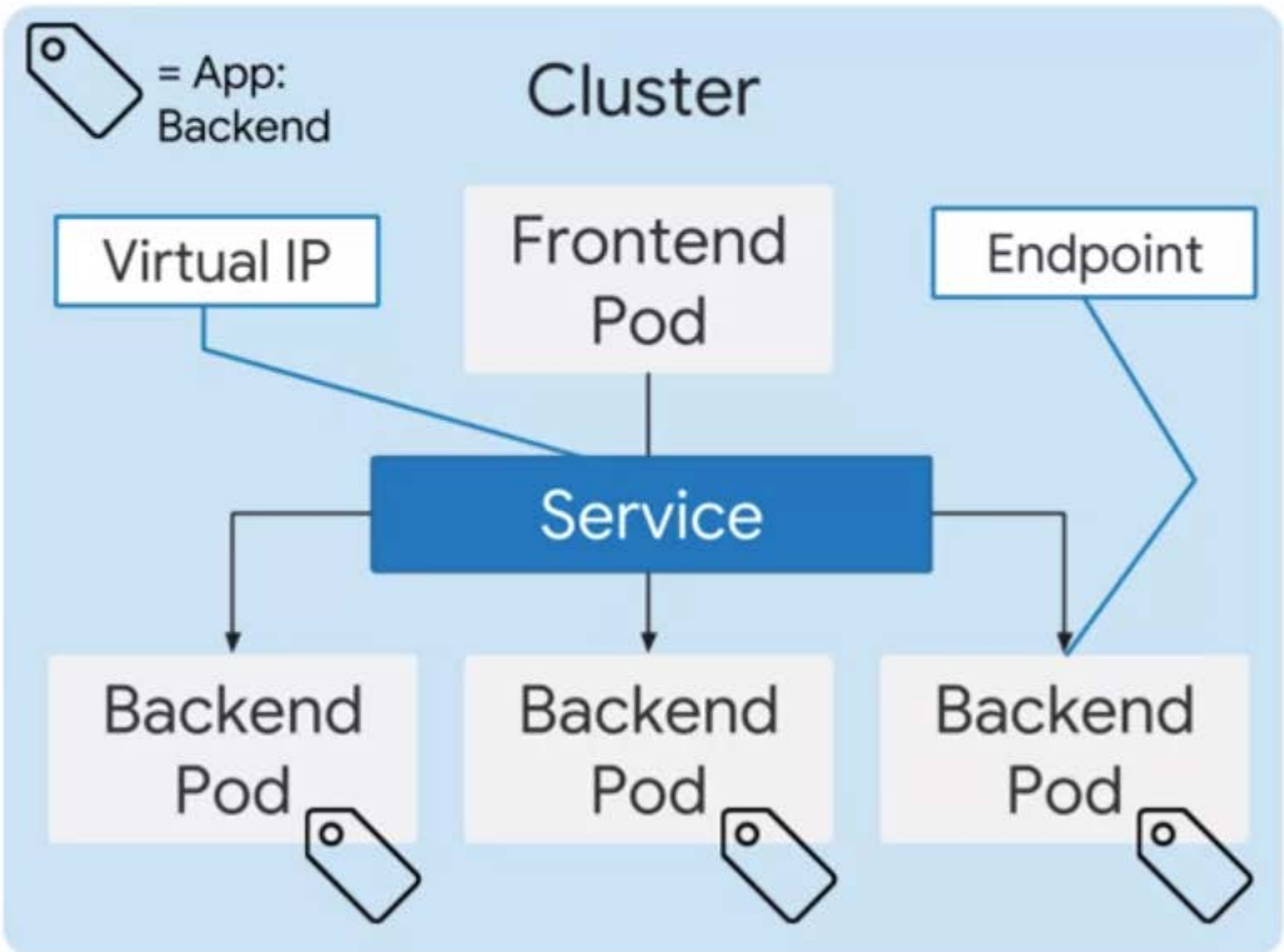

Cluster
Node
Node
Node
Default
Pods
Deployments
Kube-system
ConfigMap
Controllers
Secrets
Deployments
Kube-public

# Advanced objects: Volume

- A directory that is accessible to all containers in a Pod
- Requirements of the Volume can be specified using Pod specification
- You must mount these Volumes specifically on each container within a Pod
- Set up Volumes using external storage outside of your Pods to provide durable storage

# Advanced objects: Service

Service is a set of Pods and a policy to access them with



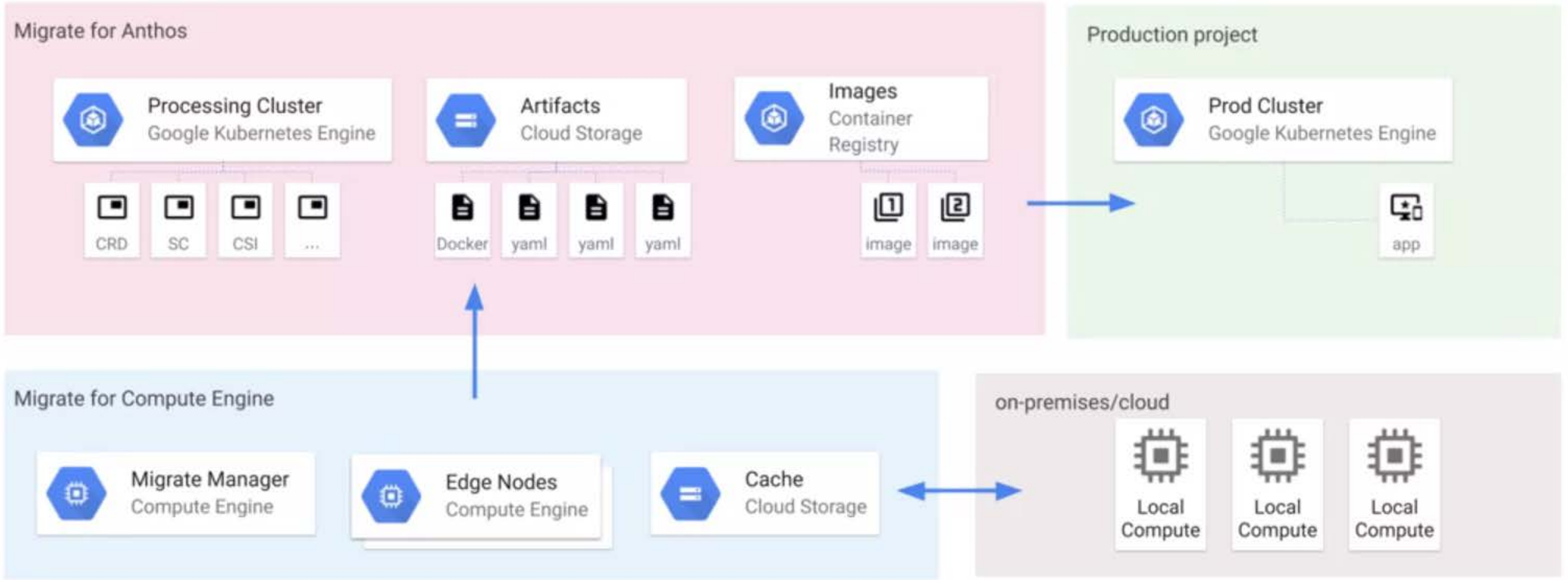
# Labels can be matched by label selectors

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
    env: dev
    stack: frontend
spec:
  replicas: 3
  selector:
    matchLabels
    app: nginx
```

# Migrate for Anthos moves VMs to containers

- Move and convert workloads into containers.
- Workloads can start as physical servers or VMs.
- Moves workload compute to container immediately (<10 min).
- Data can be migrated all at once or "streamed" to the cloud until the app is live in the cloud.

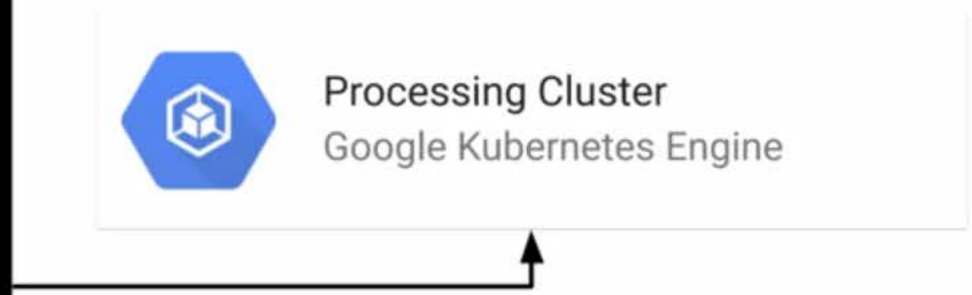
# A migration requires an architecture to be built



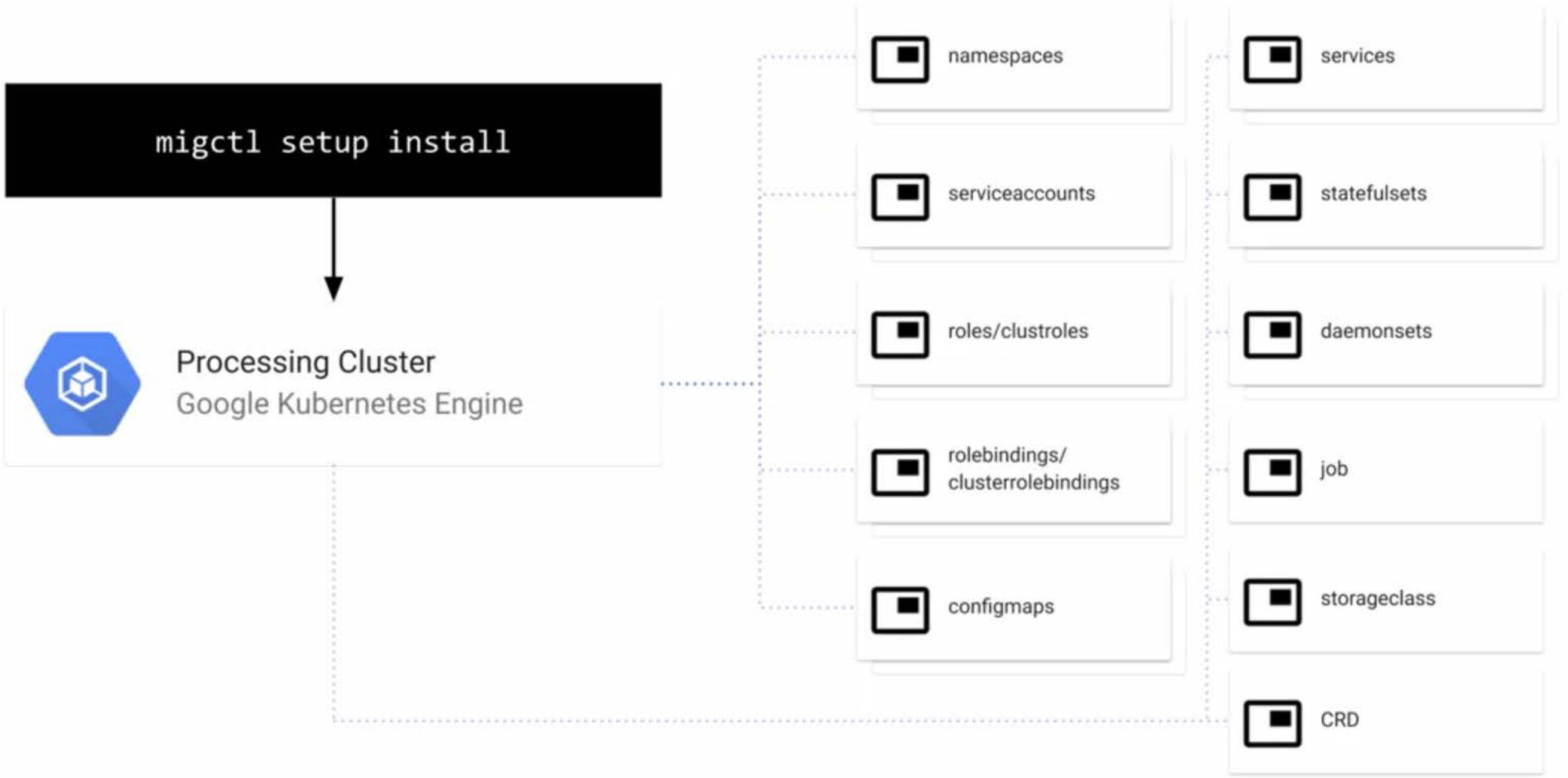
# A Migration is a multi-step process

# Migrate for Anthos requires a processing cluster

```
gcloud container --project $PROJECT_ID \
clusters create $CLUSTER_NAME \
--zone $CLUSTER_ZONE \
--username "admin" \
--cluster-version 1.14 \
--machine-type "n1-standard-4" \
--image-type "UBUNTU" \
--num-nodes 1 \
--enable-stackdriver-kubernetes \
--scopes "cloud-platform" \
--enable-ip-alias \
--tags="http-server"
```



# Installing Migrate for Anthos uses `migctl`

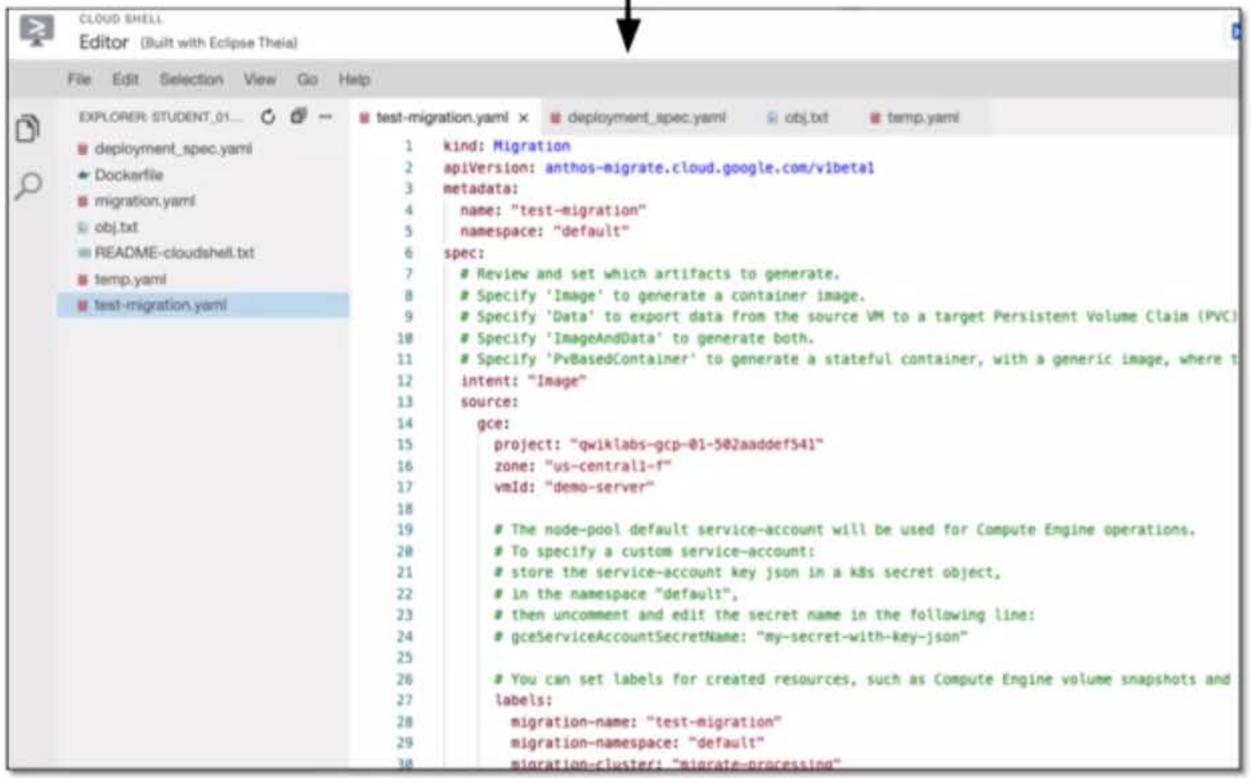


# Adding a source enables migrations from a specific environment

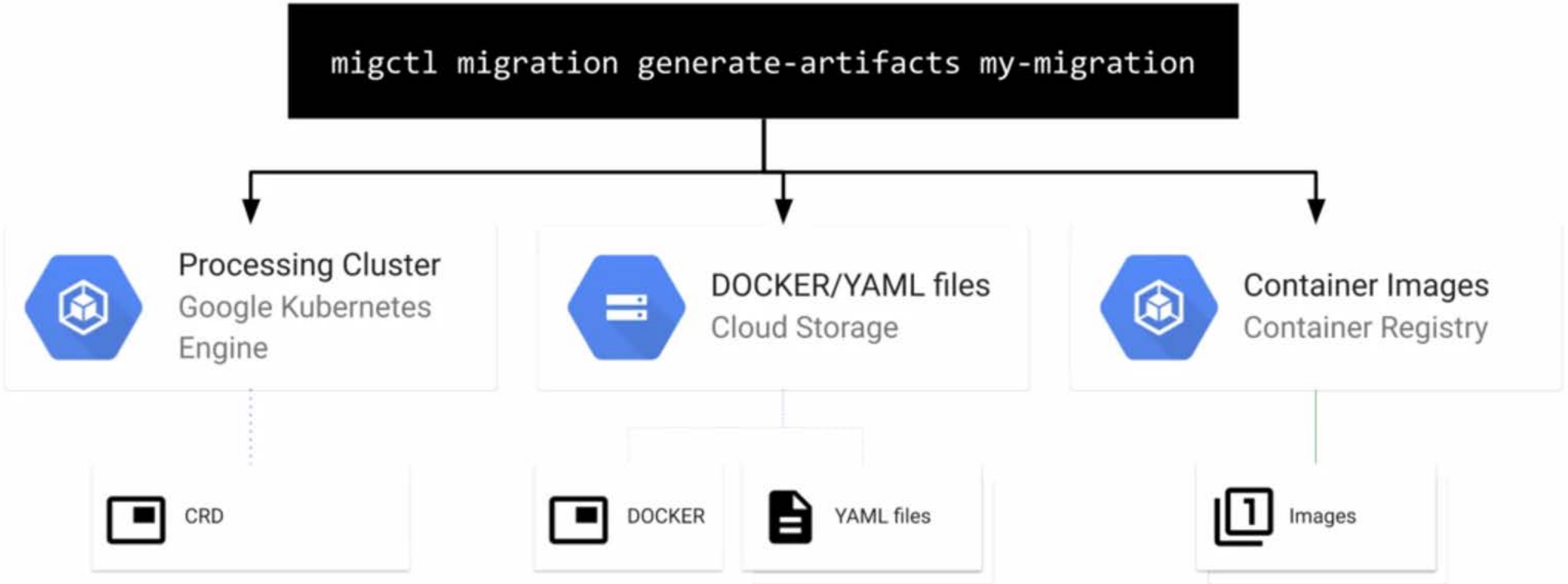
```
migctl source create ce my-ce-src --project my-project --zone zone
```

# Creating a migration generates a migration plan

```
migctl migration create test-migration --source my-ce-src --vm-id my-id --intent
Image
```

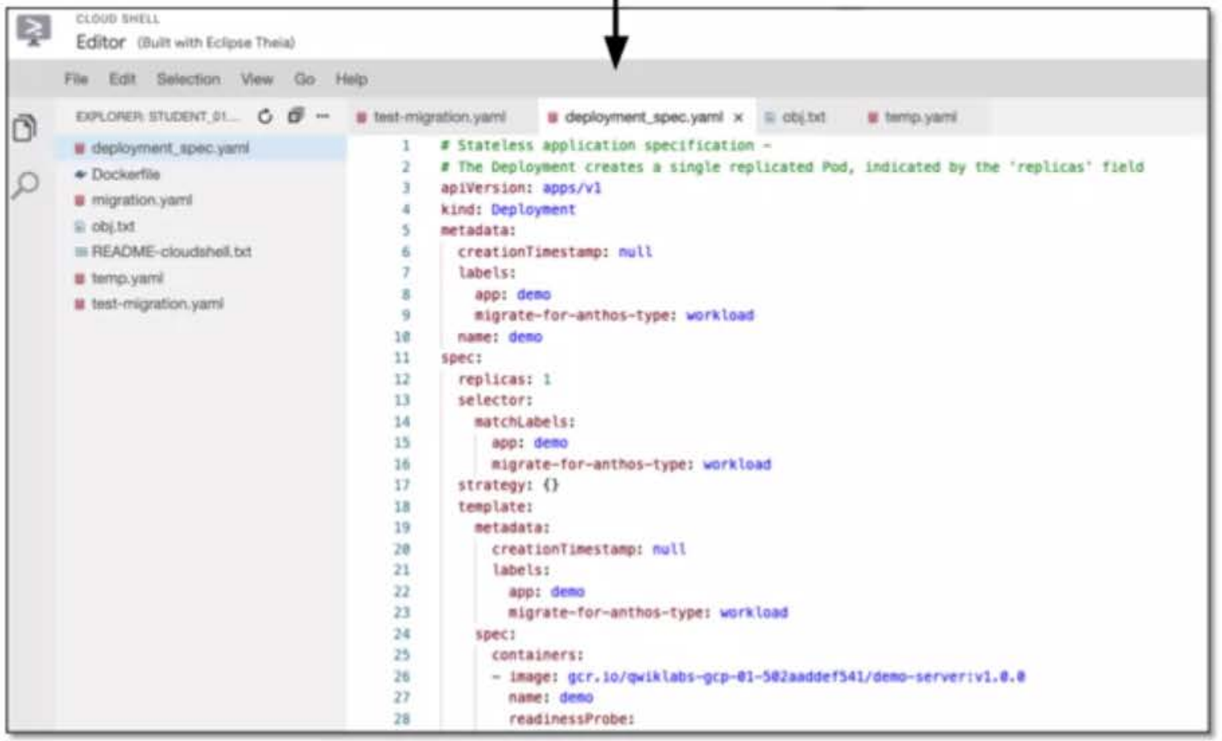


# Executing a migration generates resources and artifacts



# Deployment files typically need modification

```
migctl migration get-artifacts test-migration
```



# Apply the configuration to deploy the workload

```
kubectl apply -f deployment_spec.yaml
```