O'REILLY®

# Google Cloud Fundamentals

# Prereqs for Course

- No previous experience with Google Cloud
- Some exposure to working on the cloud recommended
- Basic understanding of deploying software on-premises

# Prereqs for Hands-on Demos

- Create a free Google Cloud account
- https://console.cloud.google.com/
- Enable billing on that account
- Please watch the getting set up video linked here:
- https://drive.google.com/drive/folders/130rcJUmsy4LANX-7iWasu7KmuvFULkSf?usp=sharing

# Introductions

I have experience with the Google Cloud Platform:

1. No experience at all
2. 0-1 years of experience
3. 2-3 years of experience
4. 3+ years of experience

# Introductions

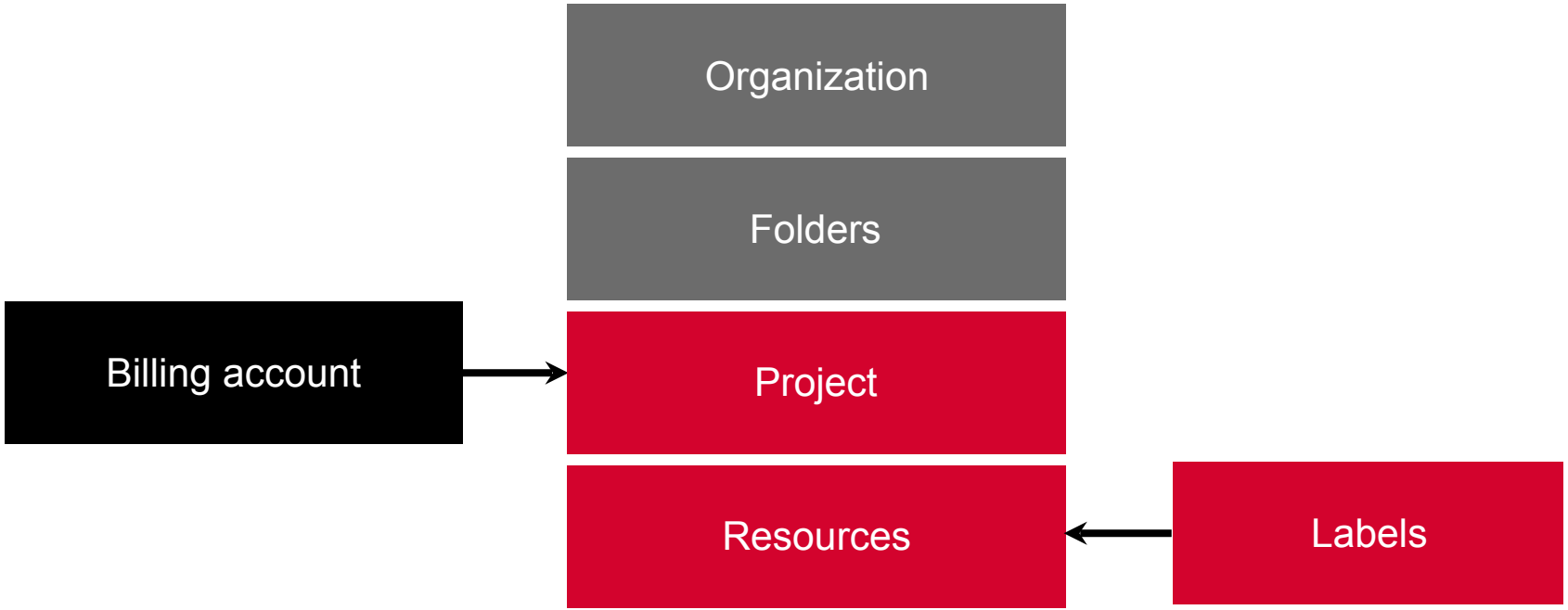
I have worked on other cloud platforms:

1. Mostly AWS
2. Mostly Azure
3. Other cloud platforms

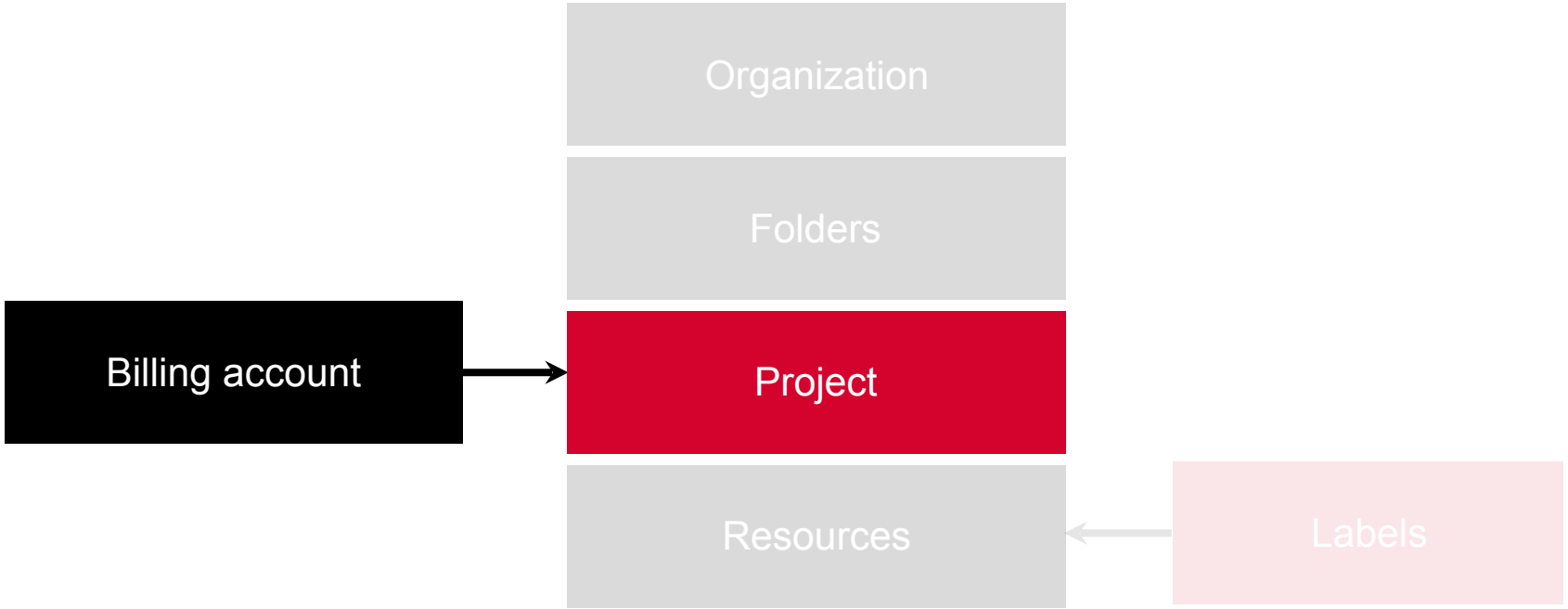
O'REILLY®

# Basic Structure of Resources on the Google Cloud

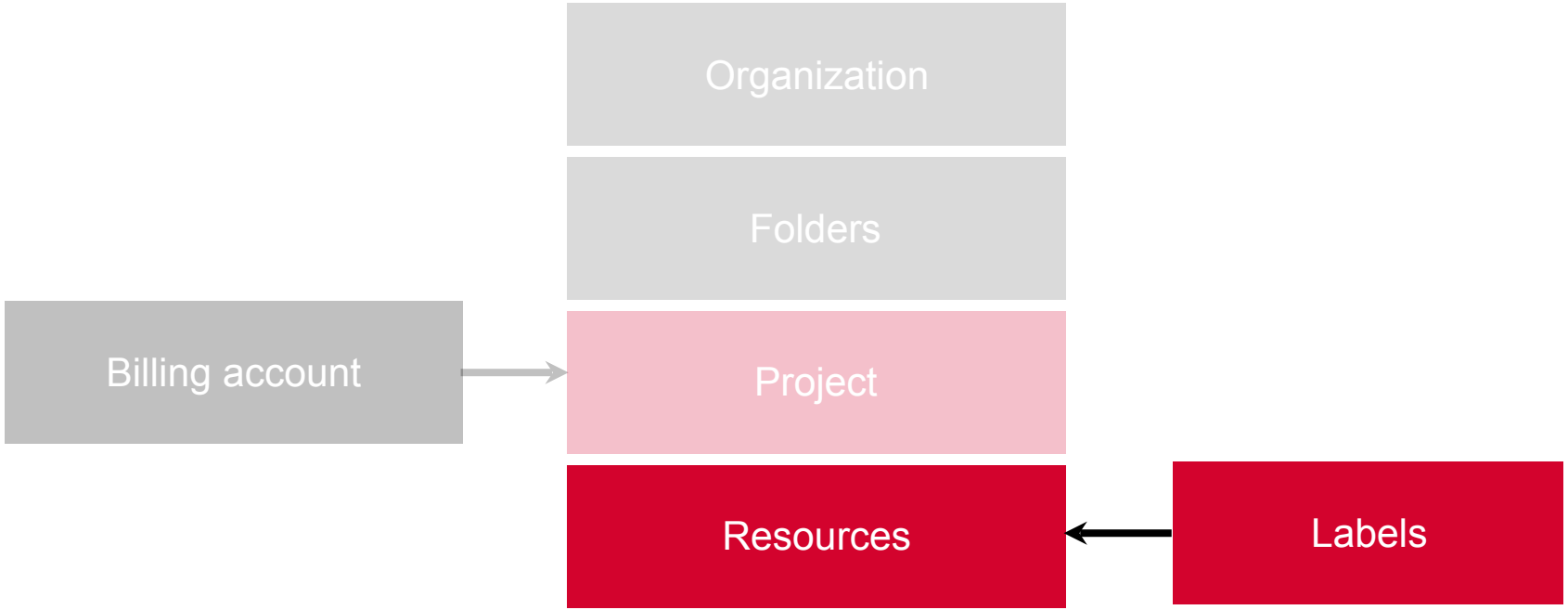
# Resource Hierarchy of Components



# Billing Accounts Are Associated with Projects



# Labels Are Applied to Resources



# Organization

- Top of resource hierarchy
- Contains projects and folders
- Identities come from G Suite or a Cloud Identity account
- IAM policies are inherited down into projects and resources
- Central control for all resources
- Projects belong to the organization, not employees
- Can grant organization level roles

# Folders

- Grouping mechanism within an organization
- Logical group of projects
- Can set IAM policies to administer multiple projects
- Model legal entities, departments, and teams

# Projects

- Container for billable resources
- Some resources can be used for free
- For all others, billing account needs to be linked
- Required resource for using Google Cloud services

# Resources

- Any component that incurs billing
- Must exist within project
- Can set resource-level IAM
- Additional IAM policies inherited from organization, folder, project
- Lowest level of the hierarchy

# Using Google Cloud Resources

Cloud Console

Cloud Shell

Command-line Tools

APIs and Client Libraries

O'REILLY

# Hands On Demos - Projects and Cloud Shell

# Projects

Which of the following best describes a project on the GCP?

1. Logical grouping of resources based on labels
2. Root node in the resource hierarchy
3. Used to group GCP networks
4. Logical grouping for resources, associated with billing

# Projects

Which of the following best describes a project on the GCP?

1. Logical grouping of resources based on labels
2. Root node in the resource hierarchy
3. Used to group GCP networks
4. **Logical grouping for resources, associated with billing**

# Cloud Shell

Which of the following best describes Cloud Shell?

1. Command-line utility used to work with the GCP services
2. Ephemeral VM which offers a terminal on the browser
3. PaaS offering on the GCP for hosted applications
4. IaaS offering on the GCP

# Cloud Shell

Which of the following best describes Cloud Shell?

1. Command-line utility used to work with the GCP services
2. **Ephemeral VM which offers a terminal on the browser**
3. PaaS offering on the GCP for hosted applications
4. IaaS offering on the GCP

O'REILLY

# Infra-as-a-Service on the Google Cloud

# Choices in Computing

**Compute**

Where is code executed and how?

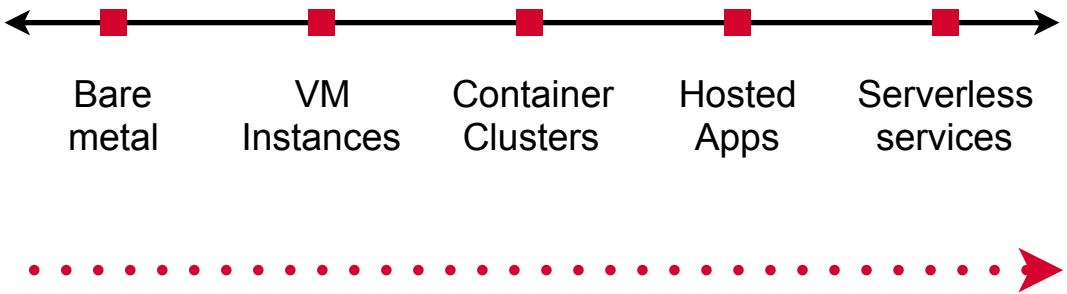
**Storage**

Where is data stored?

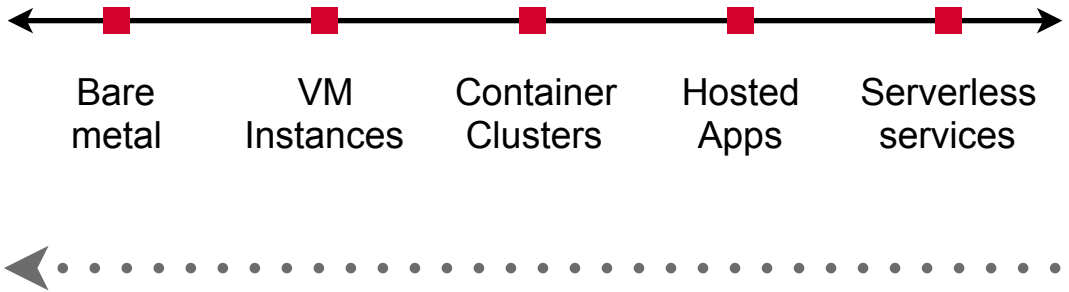
Networking, logging, are choices made after this fundamental decision

# Compute Choices

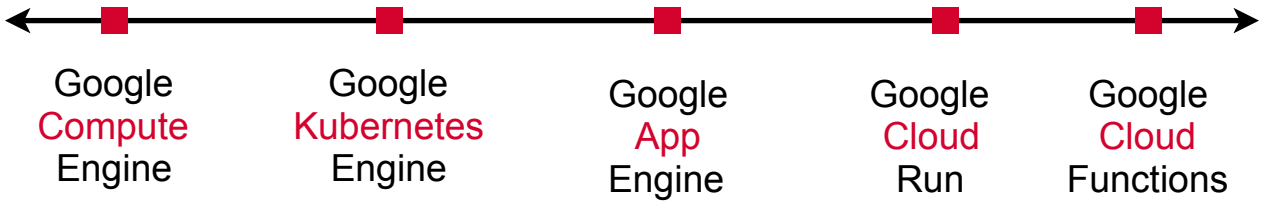
# Compute Choices



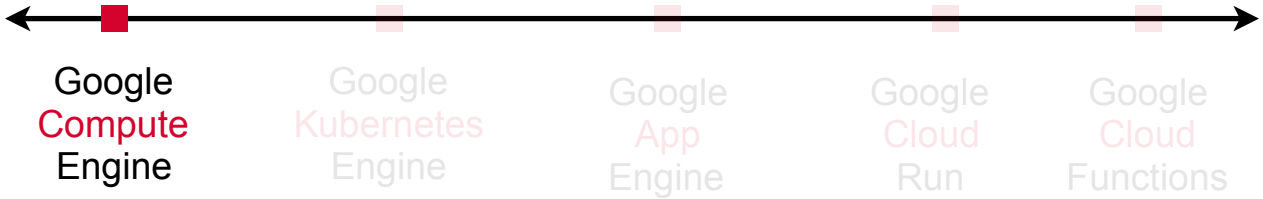
# Compute Choices



# Google Cloud Compute Choices



# Google Cloud Compute Choices



# IaaS vs. Bare Metal

## Bare Metal

- Apps run on OS which runs on hardware
- Less portable
- CPUs
- Full burden of ops and admin

## IaaS

- Hypervisor between apps and hardware
- More portable
- vCPUs
- Much of ops burden managed by service provider

# Google Cloud Internals

**Zone**

Availability zone (similar to a datacenter)

**Region**

Set of zones with high-speed network links

**Network**

User-controlled IP addresses, subnets and firewalls

# Google Cloud Internals

**Zone**

“us-central1-a”

**Region**

“us-central1”

**Network**

“default”

# Google Cloud Internals

**Zone**

“asia-south1-a”

**Region**

“asia-south1”

**Network**

“default”

# Configuration Choices

## Machine Family

General purpose, compute optimized, memory optimized, accelerator-optimized

## Machine Series

Machines have generation numbers where higher generations have newer features

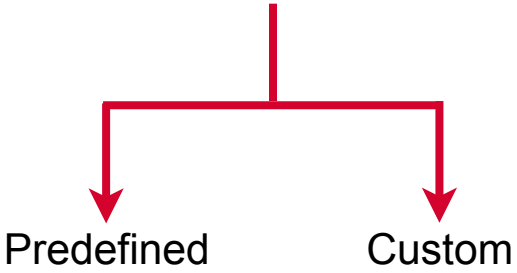
## Machine Type

vCPUs count, memory capacity, and storage capacity

## Base Image

Public (free or premium), custom, snapshots from boot disks

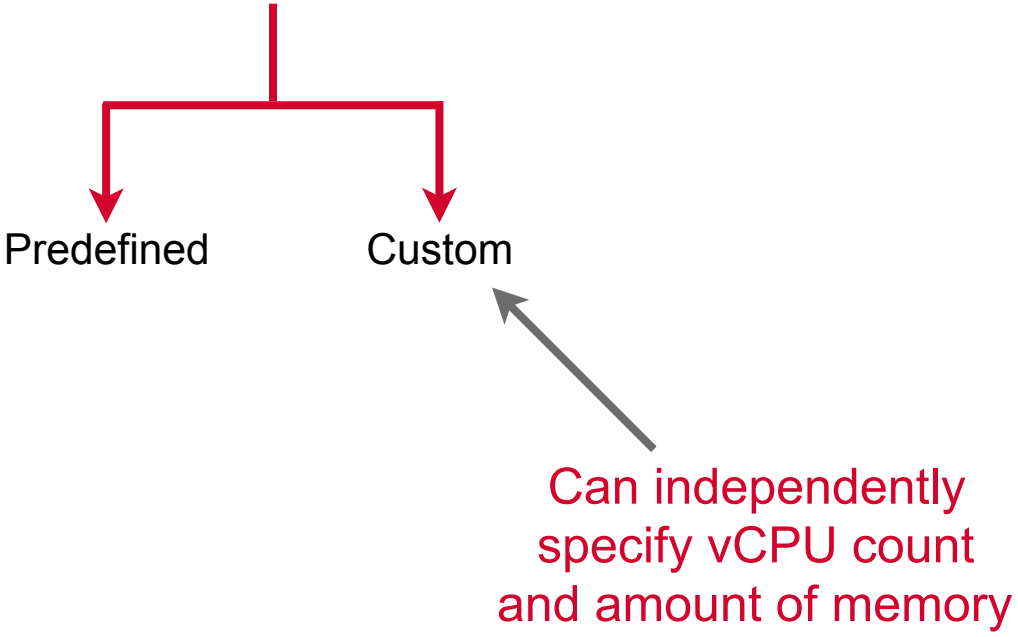
# Machine Type



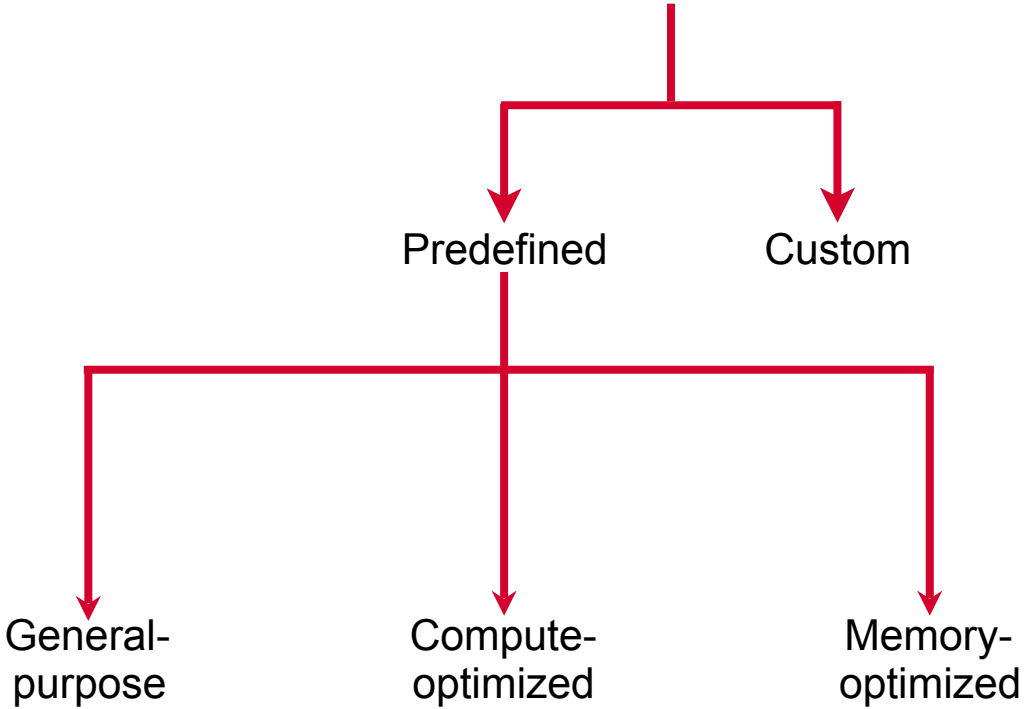
# Machine Type


Fixed set of types with fixed ratios of memory to vCPU count

# Machine Type



# Machine Type



# General Purpose Machines

- Day to day computing for known workloads
- **Best price-performance ratio**
- N1 first generation: 6.5GB of memory per vCPU
- N2 second generation: 8GB of memory per vCPU
  - More heavy duty workloads such as web serving, databases, applications use N2
- Can customize machine types
- Come in high-memory and high-cpu variants

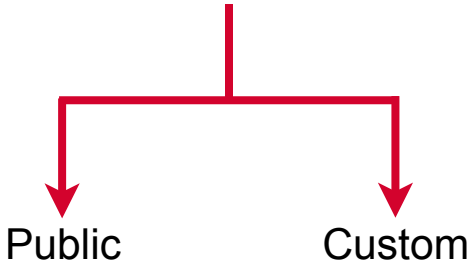
# Compute-optimized Machines

- Compute intensive workloads
- Offer the **highest performance per core**
- C2 machine types
- Gaming, single-threaded applications, electronic design automation
- Custom machine types not supported

# Memory-optimized Machines

- Memory-intensive workloads
- Offer the **highest memory per core**
- Custom machine types not supported

# Base Images



# Base Images

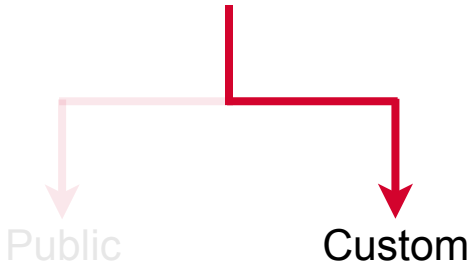

Provided and maintained by Google, open-source communities, and third-party vendors

All projects have access to these images and can use them to create instances

# Base Images


Linux, Windows, Container-optimized OS,
SQL Server

# Base Images




Available only to your project

First, create a custom image from boot disks and other images; then, use the custom image to create an instance

# Spot VM Instances

An instance that you can create and run at a much lower price than normal instances. However, **GCE might terminate (preempt)** these instances if it requires access to those resources for other tasks.

May not always be available

Not covered by SLAs

# Preemptible VM Instances

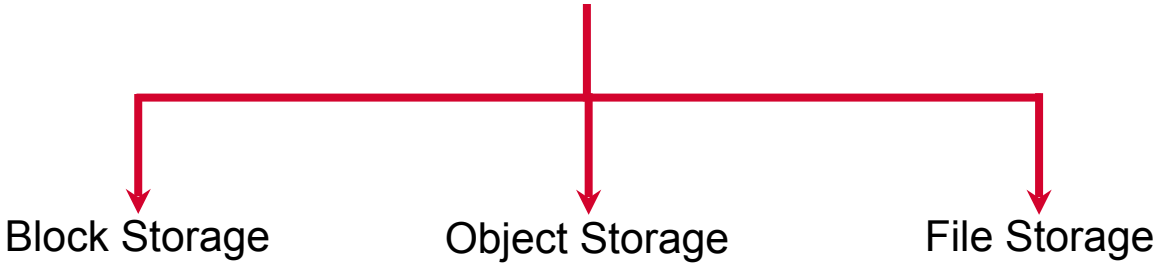
Similar to Spot VMs (older product and will have fewer features than Spot VMs)

**Will definitely be preempted every 24 hours**

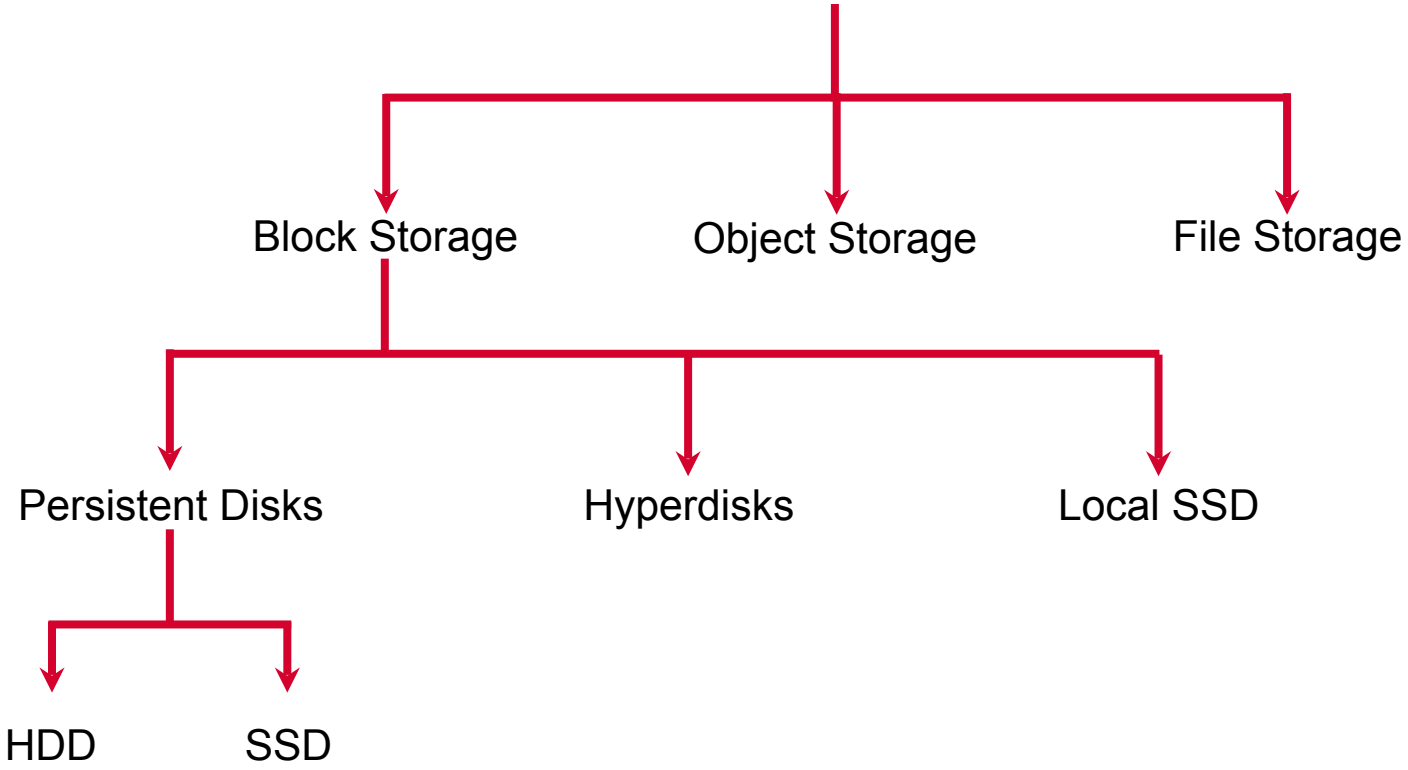
May not always be available

Not covered by SLAs

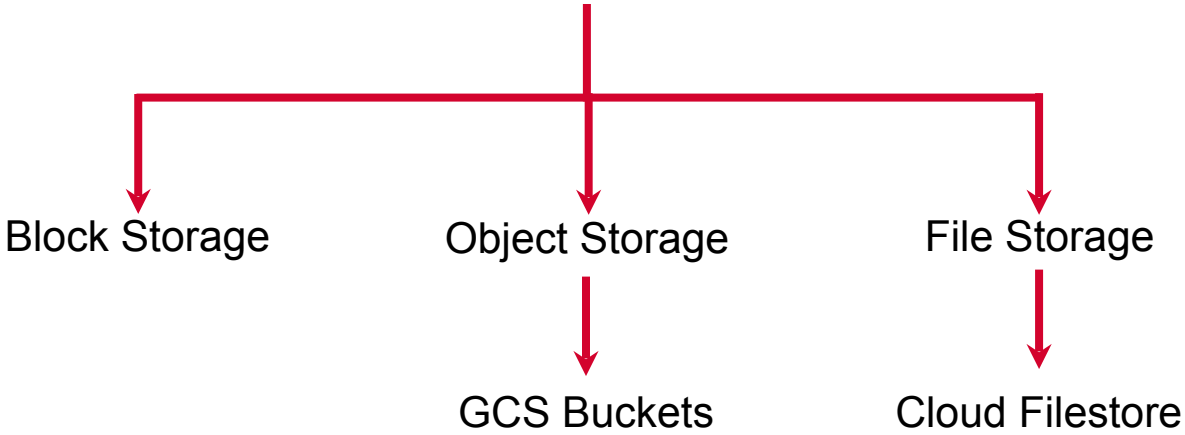
# Accessing Storage from VMs



# Accessing Storage from VMs



# Accessing Storage from VMs



# Persistent Disks vs. Buckets

## Persistent Disks

- Block storage
- Max 64TB in size
- Pay what you allocate
- Tied to GCE VMs
- Zonal (or regional) access

## Buckets

- Object storage
- Infinitely scalable
- Pay what you use
- Independent of GCE VMs
- Global access

O'REILLY®

# Hands On Demos - Google Compute Engine

# Region

Which of the following best describes a region on the GCP?

1. A logical area that may be spread across countries
2. A single datacenter on the GCP
3. A geographical area with multiple datacenters
4. Physically connected hardware devices in a datacenter

# Region

Which of the following best describes a region on the GCP?

1. A logical area that may be spread across countries
2. A single datacenter on the GCP
3. **A geographical area with multiple datacenters**
4. Physically connected hardware devices in a datacenter

# Persistent Disks

What is the pricing mechanism for Persistent Disks?

1. If the you create a 100GB disk but you use just 5GB you pay for the entire 100GB
2. If you create a 100GB disk but you use just 5GB you pay for only 5GB
3. If you create a 100GB disk but you use just 5GB you pay for only 5GB + a little extra

# Persistent Disks

What is the pricing mechanism for Persistent Disks?

1. **If the you create a 100GB disk but you use just 5GB you pay  for the entire 100GB**
2. If you create a 100GB disk but you use just 5GB you pay for only 5GB
3. If you create a 100GB disk but you use just 5GB you pay for only 5GB + a little extra


O'REILLY®
Storage:
Exploring Storage
Resources on the
Google Cloud

# Choices in Computing

**Compute**

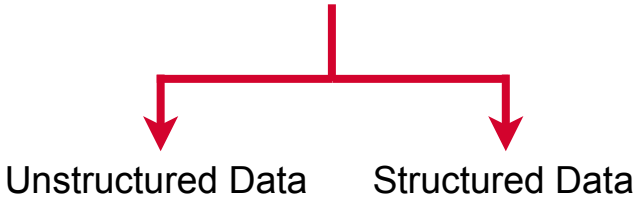
Where is code executed and how?

**Storage**

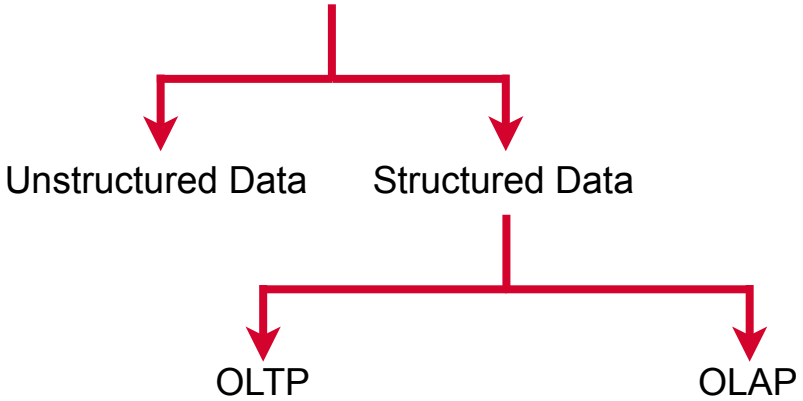
Where is data stored?

Networking, logging, are choices made after this fundamental decision

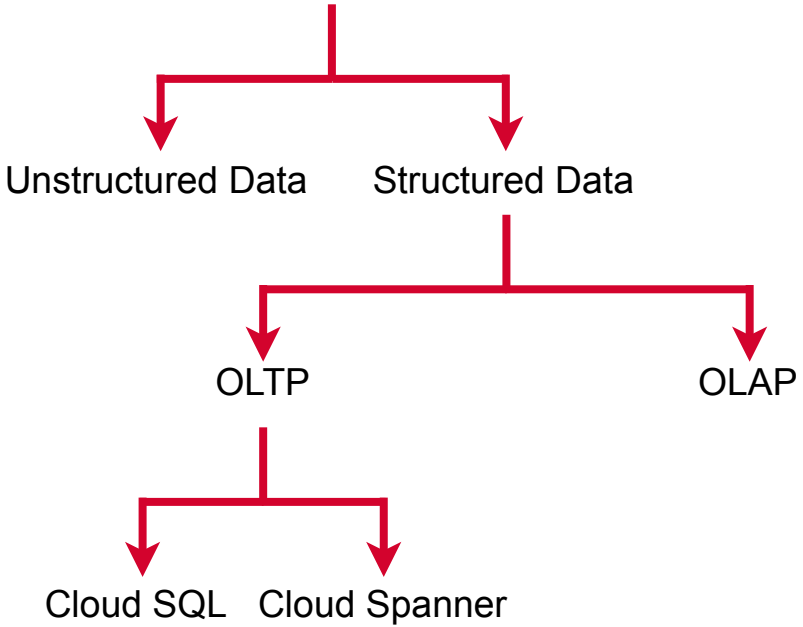
# Storage Technologies



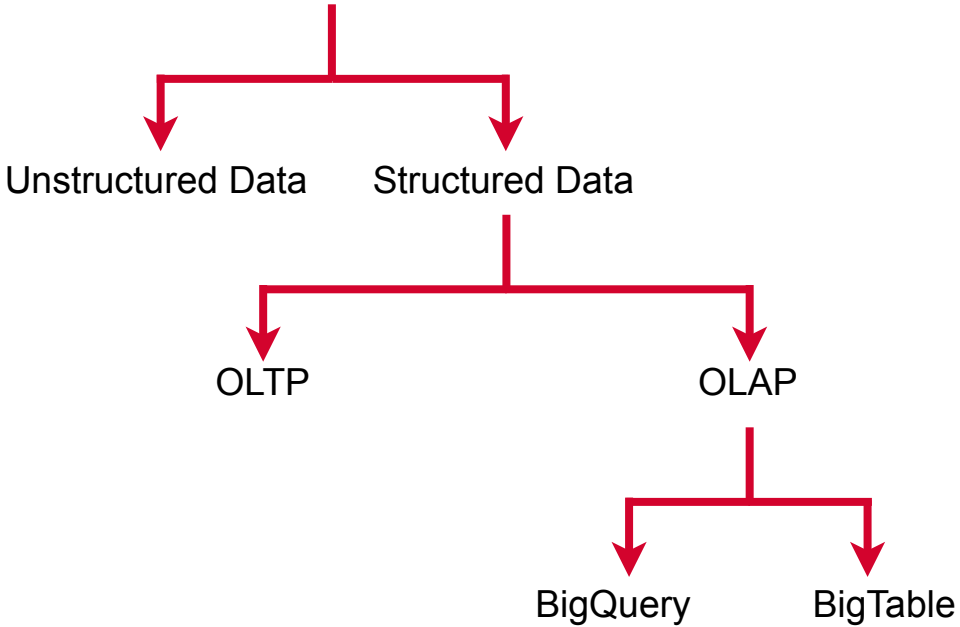
# Storage Technologies



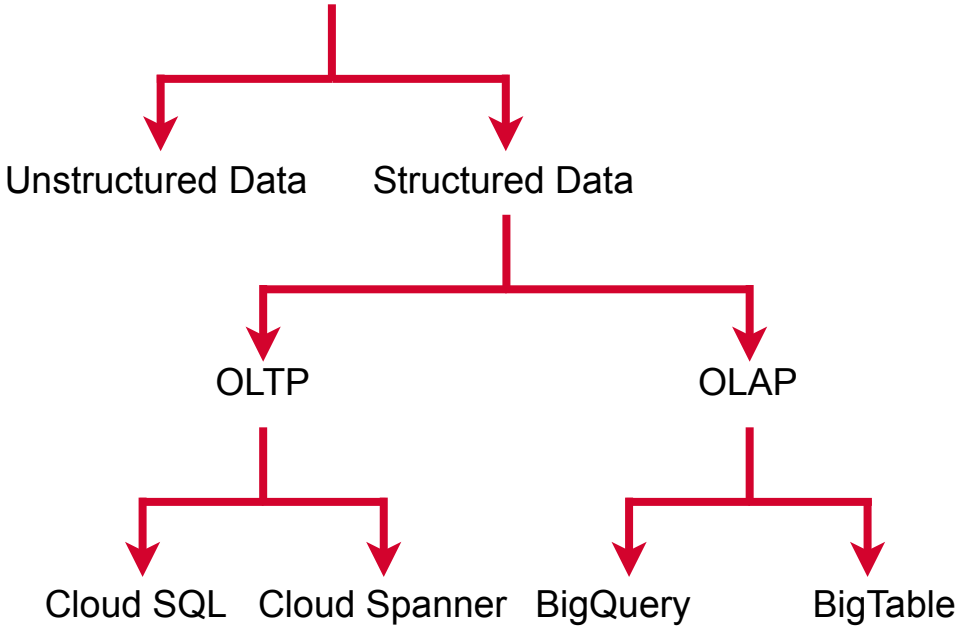
# Storage Technologies



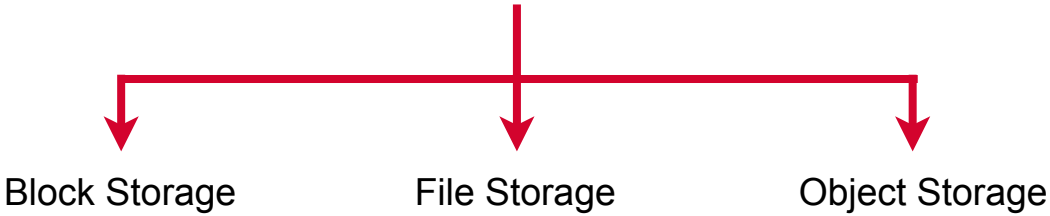
# Storage Technologies



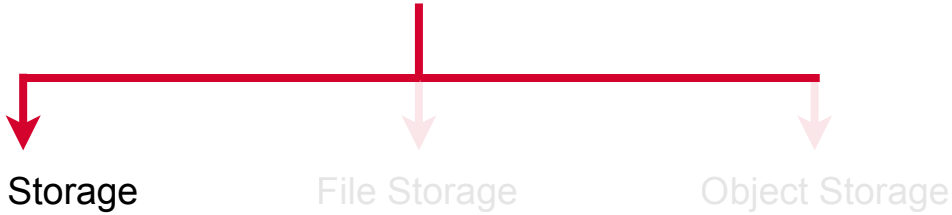
# Storage Technologies



# Unstructured Data



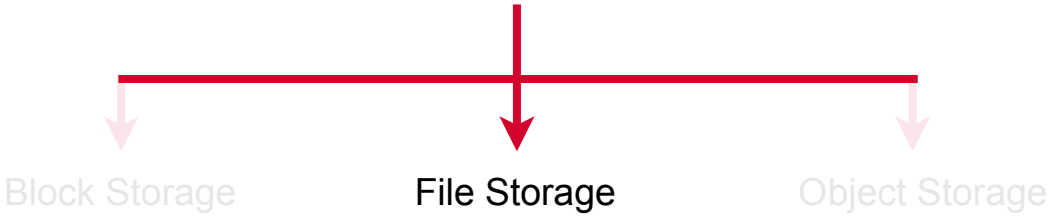
# Unstructured Data




Physically addressable storage accessed from compute - data split into uniform blocks

High performance read and write access at the block level

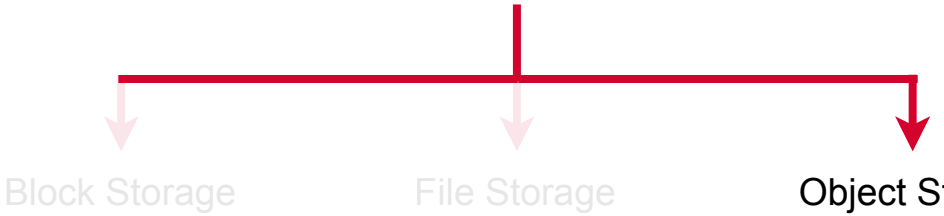
# Unstructured Data




Stores data as a hierarchy of files within directories

Shared concurrent access from multiple machines

# Unstructured Data




Logically addressable storage accessed from compute or by human users

# Persistent Disks vs. Buckets

## Persistent Disks

- Block storage
- Max 64TB in size
- Pay what you allocate
- Tied to GCE VMs
- Zonal (or regional) access

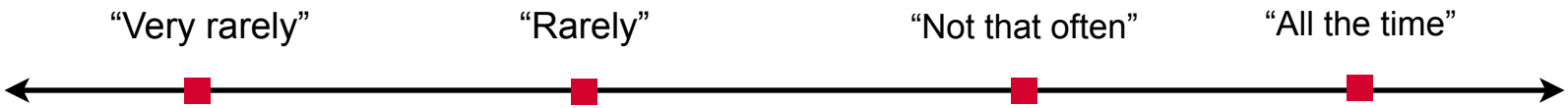
## Buckets

- Object storage
- Infinitely scalable
- Pay what you use
- Independent of GCE VMs
- Global access

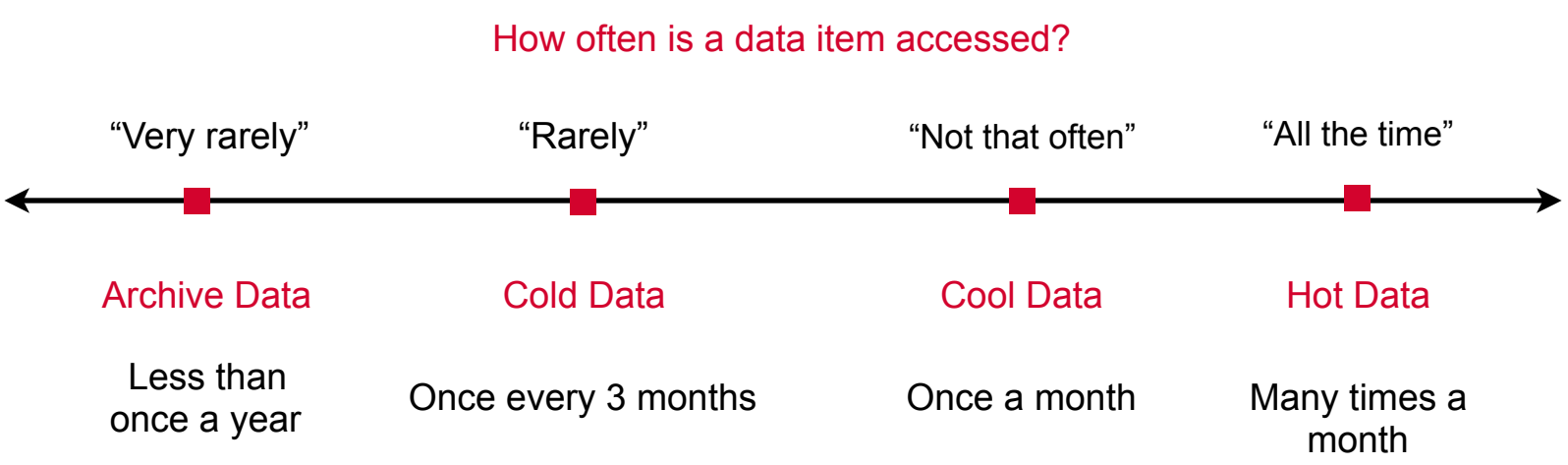

O'REILLY®
Google
Cloud
Storage
Insert subtitle here...

# GCS Storage Classes

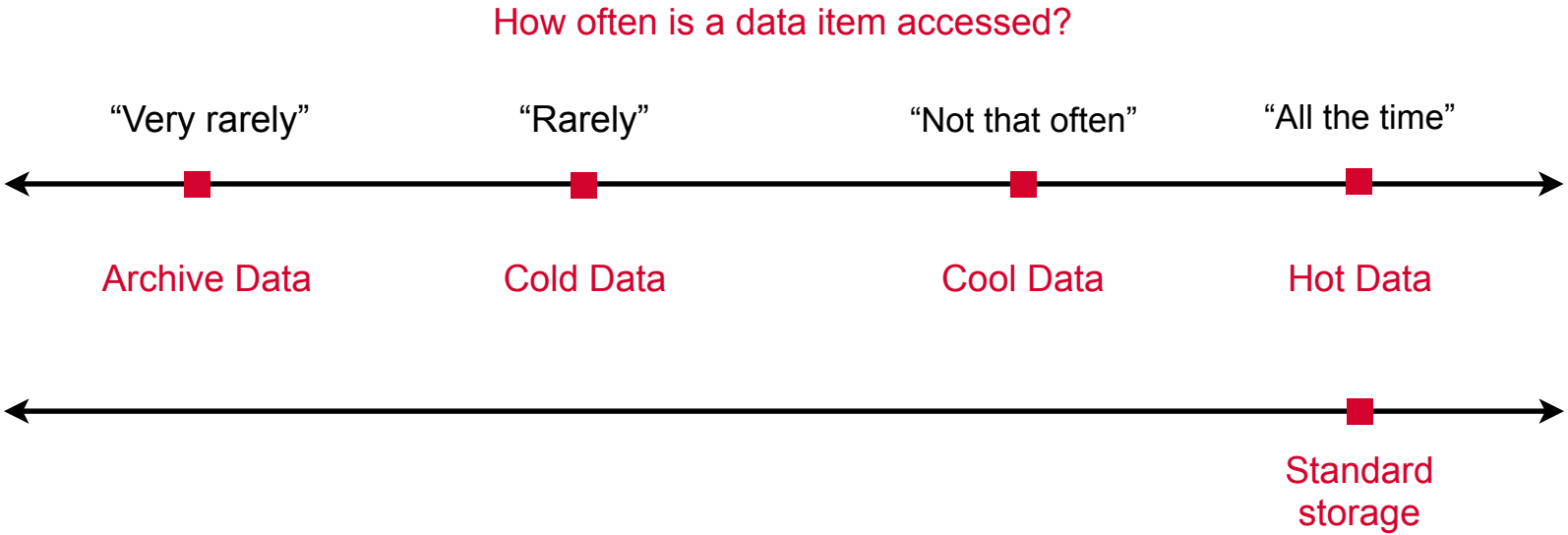
How often is a data item accessed?



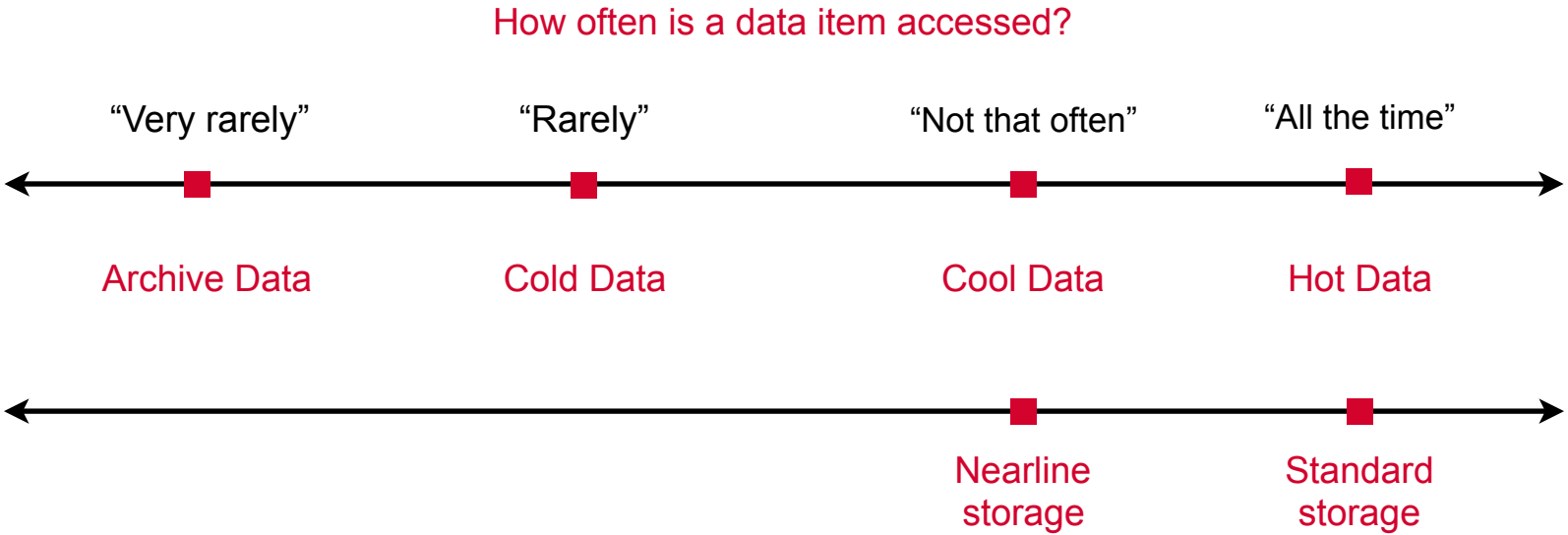
# GCS Storage Classes



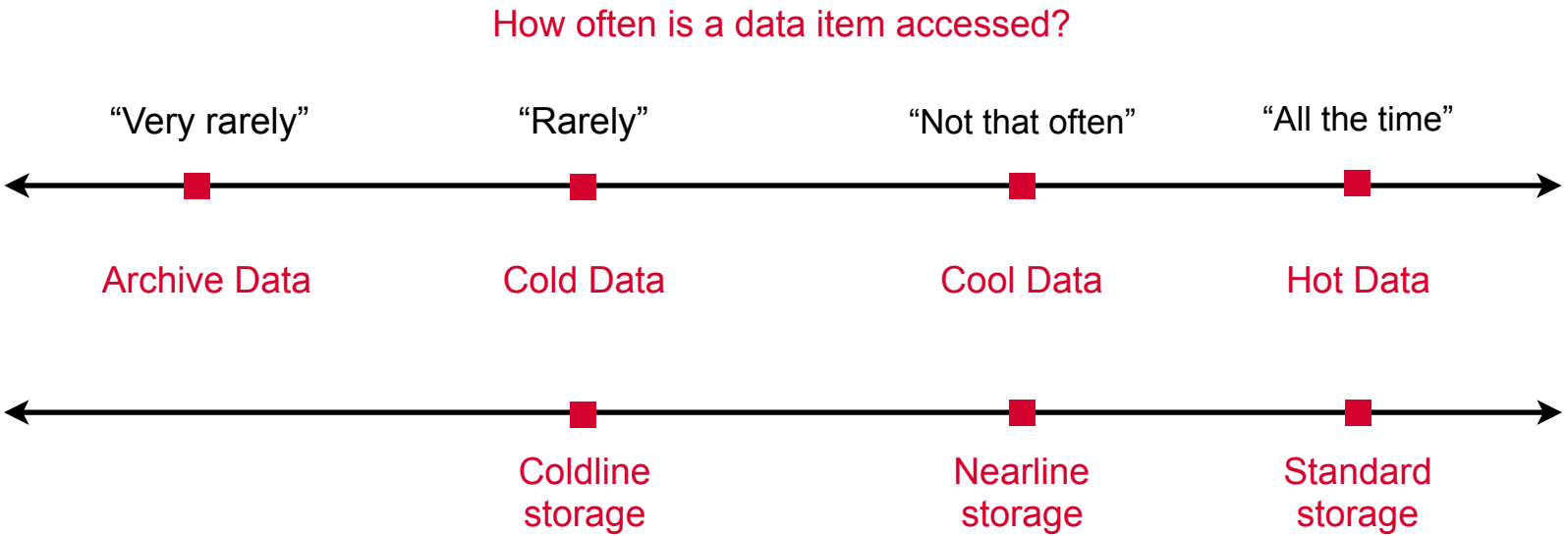
# GCS Storage Classes



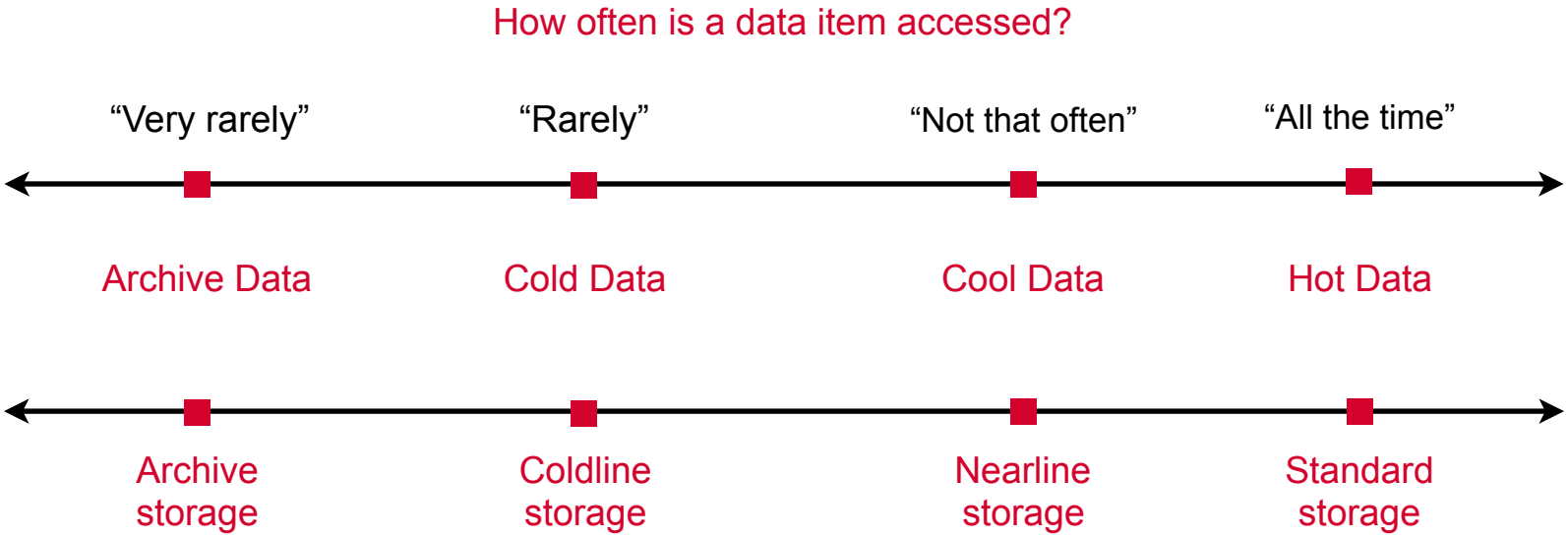
# GCS Storage Classes



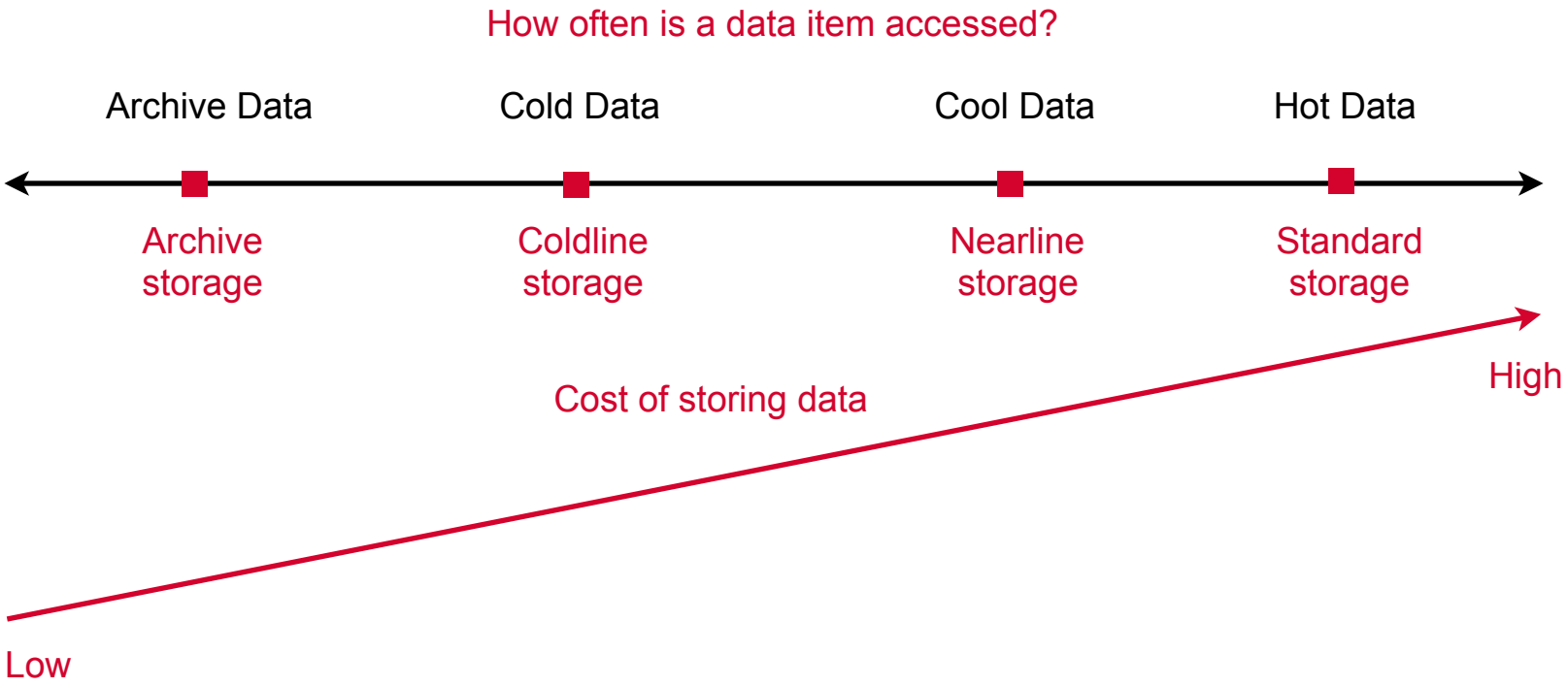
# GCS Storage Classes



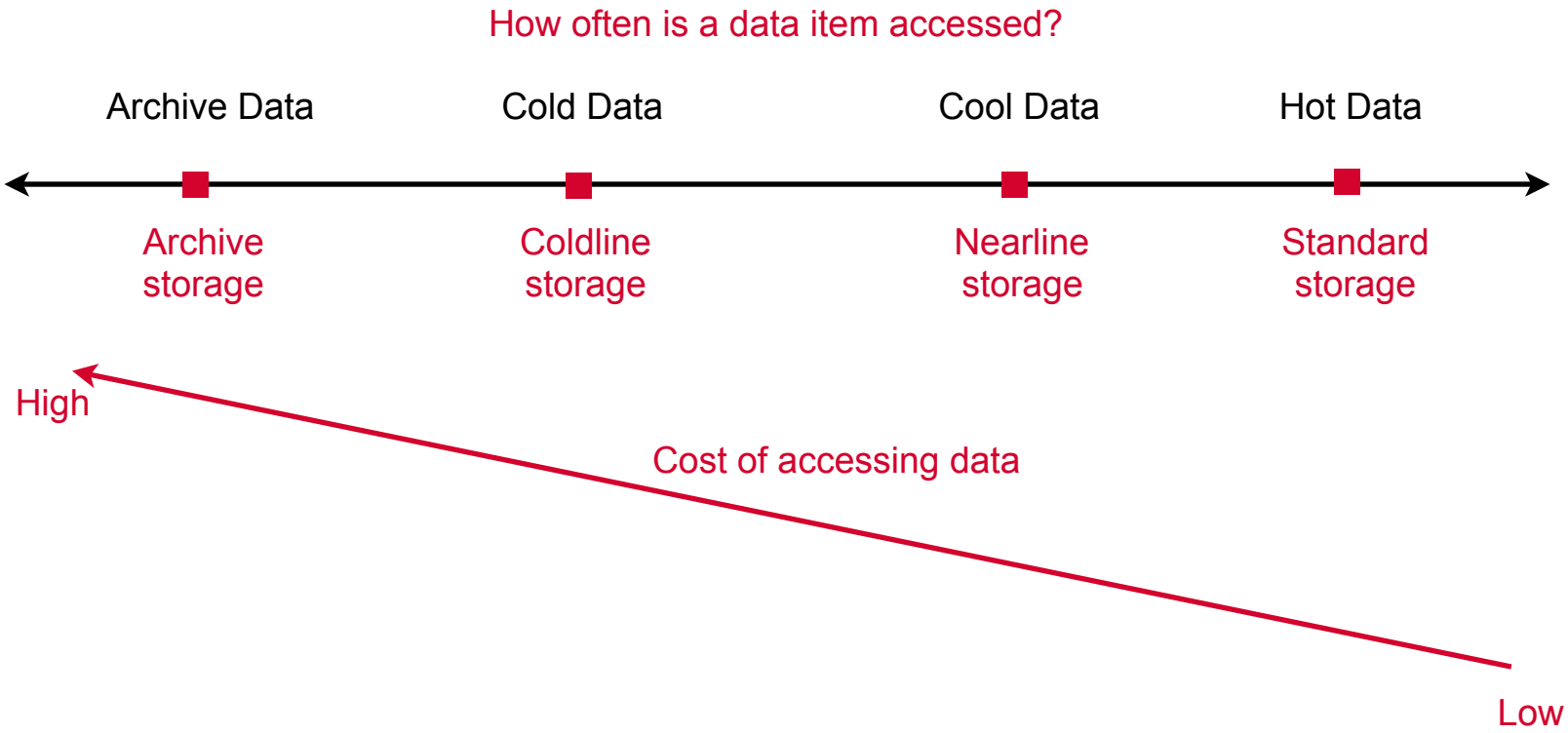
# GCS Storage Classes



# GCS Storage Classes



# GCS Storage Classes



# All Storage Classes


# Autoclass

Moves data that is not accessed to colder storage classes to reduce cost

Moves data that is accessed to standard storage to optimize cost of future access

Coldline and Archive has about the same speed of access as other storage classes (different from AWS Glacier and S3)

- Availability
- Storage Costs
- Retrieval Costs
- Durability
- Access Frequency
- Use Cases

Different storage classes represent different trade-offs

Several parameters along which to compare

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Availability |
|---|---|
| Standard storage (dual and multi-regional) | 99.95% |
| Standard storage (regional) | 99.9% |
| Nearline (regional) | 99.0% |
| Coldline (regional) | 99.0% |

Dual-region and multi-region buckets are tied to multi-regional locations: US, EU and Asia

Helps adhere to data storage regulations in the US and EU

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Storage Cost (cents/GB/month) |
|---|---|
| Standard | 2.6 |
| Nearline | 1.0 |
| Coldline | 0.7 |
| Archive | 0.24 |

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Retrieval Cost (cents/GB) |
|---|---|
| Standard | **None** |
| Nearline | 1.0 |
| Coldline | 2.0 |
| Archive | 5.0 |

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Minimum Commitment |
|---|---|
| Standard | **None** |
| Nearline | 30 days* |
| Coldline | 90 days* |
| Archive | 365 days* |

*Early deletion will incur charges

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Durability |
| --- | --- |
| Standard | 99.999999999% |
| Nearline | 99.999999999% |
| Coldline | 99.999999999% |
| Archive | 99.999999999% |

“11 nines”

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Access Frequency |
|---|---|
| Standard | Daily |
| Nearline | Monthly |
| Coldline | Quarterly |
| Archive | Less than once a year |

Availability

Storage Costs

Retrieval Costs

Durability

Access Frequency

Use Cases

| Storage Class | Access Frequency |
| --- | --- |
| Standard storage (dual and multi-regional) | Serving websites, interactive workloads, mobile and gaming applications |
| Standard storage (regional) | Access from Compute Engine VMs or Dataproc cluster |
| Nearline | Data backup, disaster recovery, archival storage |
| Coldline/Archive | Legal or regulatory needs; also disaster recovery where recovery time is important |

# Autoclass

Moves data that is not accessed to colder storage classes to reduce cost

Moves data that is accessed to standard storage to optimize cost of future access

# Object Versioning

- Needs to be enabled for bucket
- Once enabled, bucket creates archived versions of each object
- Whenever live object is overwritten or deleted
- Version with unique generation number is created
- Each copy charged separately

# Object Lifecycle Management

- Can automatically specify changes to object storage class
  - “Change from regional to nearline after 30 days”
  - “Delete all data created before 1/8/2018”
  - “Delete all but 2 most recent versions”

# Encryption

- Encrypted even at rest
- Default: Google generates keys
- Can use CSEK
  - *C*ustomer *S*upplied *E*ncryption *K*ey

O'REILLY®

# Hands on Demos - Google Cloud Storage

# Storage Class

Which of the following is true for coldline storage?

1. Low cost of storage, high cost of retrieval
2. Low cost of storage, low cost of retrieval
3. High cost of storage, low cost of retrieval
4. High cost of storage, high cost of retrieval

# Storage Class

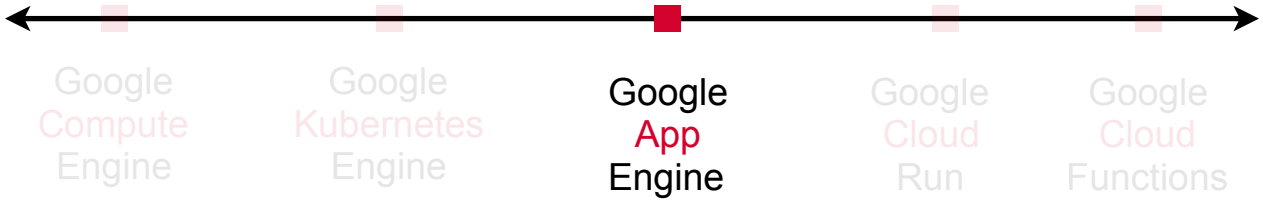
Which of the following is true for coldline storage?

1. **Low cost of storage, high cost of retrieval**
2. Low cost of storage, low cost of retrieval
3. High cost of storage, low cost of retrieval
4. High cost of storage, high cost of retrieval


O'REILLY®
Platform-as-a-Service on the Google Cloud


O'REILLY®
Google
App Engine

# Google Cloud Compute Choices



# Google App Engine

Web framework and platform for hosting web applications on the Google Cloud Platform

Support for Go, PHP, Java, Python, Node.js, .NET, Ruby and other languages

# Google App Engine

Web framework and platform for hosting web applications on the Google Cloud Platform

Support for Go, PHP, Java, Python, Node.js, .NET, Ruby and other languages

Focus on development and code

Infrastructure and scaling taken care of by the platform

# App Engine Environments

| Standard Environment | Flexible Environment |
| --- | --- |

# App Engine Environments

## Standard

- App runs in a proprietary sandbox
- Instances start up in seconds
- Code in few languages/versions only
- No other runtimes possible
- Apps cannot access Compute Engine resources
- Can install 3rd party binaries for selected runtimes

# App Engine Environments

## Standard

- App runs in a proprietary sandbox
- Instances start up in seconds
- Code in few languages/versions only
- No other runtimes possible
- Apps cannot access Compute Engine resources
- Can install 3rd party binaries for selected runtimes

## Flexible

- Runs in Docker container on GCE VM
- Instance start up in minutes
- Code in far more languages/versions
- Custom runtimes possible
- Apps can access Compute Engine resources, some OS packages
- Can install and access third-party binaries

# App Engine Environments

## Standard

- Apps that experience traffic spikes
- Usually stateless HTTP web apps

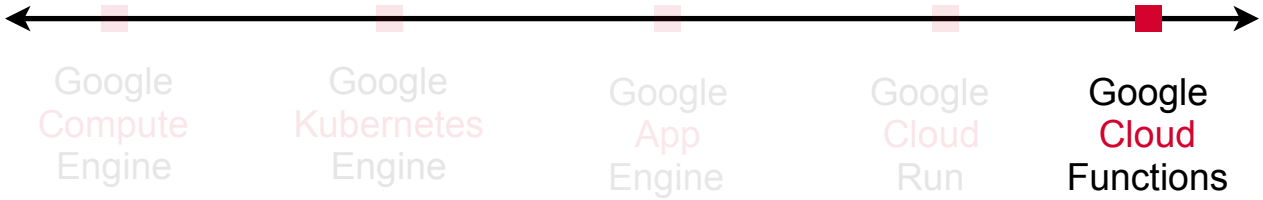
## Flexible

- Apps that experience consistent traffic
- General purpose apps

O'REILLY®

# Google Cloud Functions

# Google Cloud Compute Choices



# Cloud Functions

Event-driven serverless compute platform

# Event-driven Serverless Compute

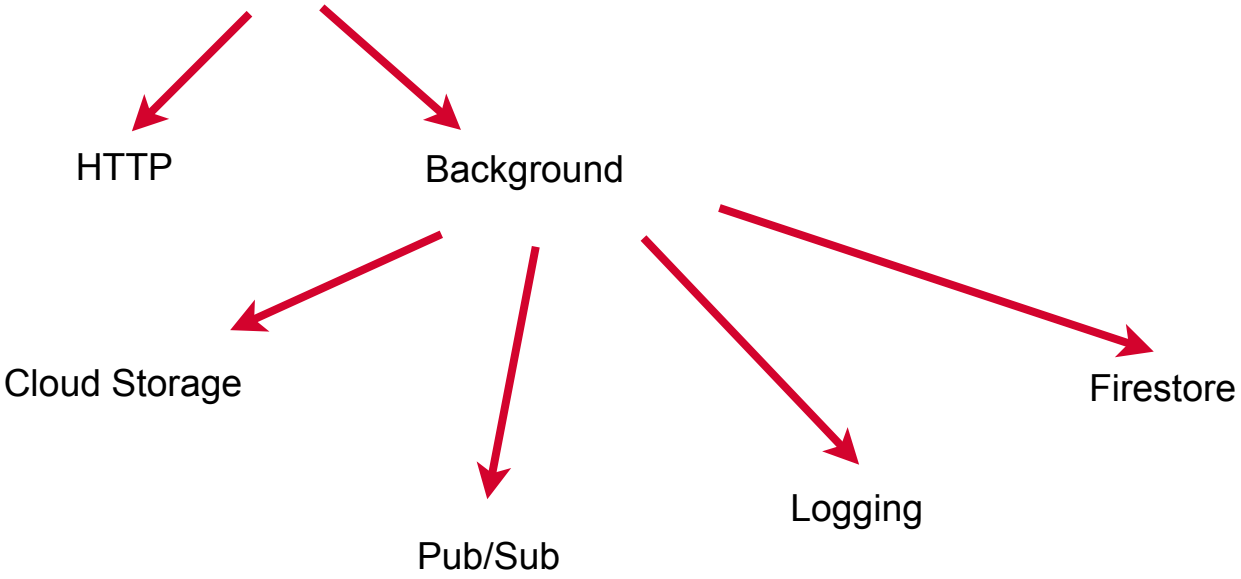
Event occurs

Platform triggers execution

Cloud Function code runs

Invokes other Google Cloud services

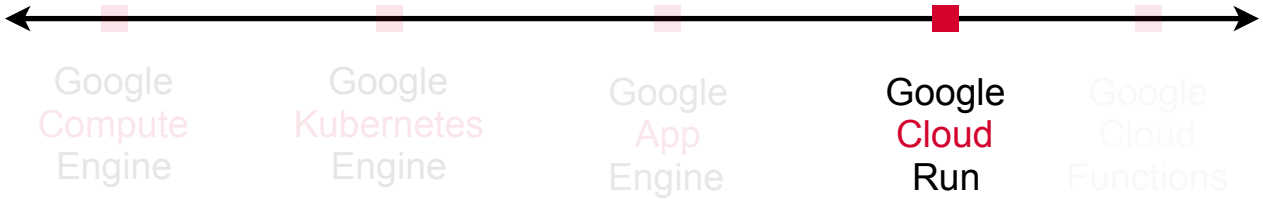
# Types of Events



# Concurrency and Scale

- Spin up function instances based on current load
- Functions receive event parameters from platform
- Functions do not share memory or variables
- An instance processes a single request (generation 1)
- Function concurrency supported (generation 2)
- Functions should be stateless

# Google Cloud Compute Choices



# Cloud Run

Serverless, managed platform that lets you run containers directly on top of Google’s scalable architecture

# Cloud Run

- Write your code in any programming language
- Create a container image (or use source-based deployment option - Google Cloud will build container image for you)
- Register the container with the artifact registry
- Deploy your container directly using Cloud Run
- No cluster creation no infrastructure management
- Request-based pricing and instance-based pricing

# Running Code Using Cloud Run


Both use the same environment and have the same integrations with other Google Cloud services

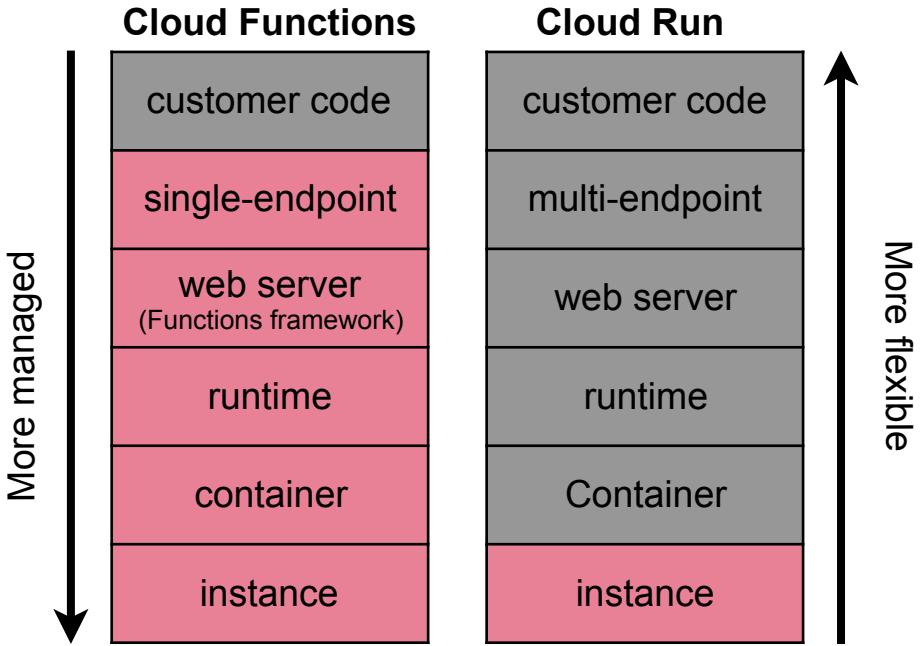
# Cloud Run Services

- Used to run code that responds to web requests or events
- Each service located in a Google Cloud region
- Replicated across zones in the region
- Exposes an endpoint
- Automatically scales underlying infrastructure to handle incoming requests
- Version management, rollbacks, traffic management - all handled by the platform

# Cloud Run Jobs

- Used to run code that performs work (a job) and quits when the job is done
- Each service located in a Google Cloud region and executes one or more containers to completion
- A job comprises of many tasks executing in parallel - each container runs one task

# Cloud Functions vs. Cloud Run




How managed do you want to be?

# Cloud Functions vs. Cloud Run

## Cloud Functions

- Specific limited runtimes supported
- Can be triggered based on platform events
- No support for running jobs
- 2nd generation functions support concurrency

## Cloud Run

- All runtimes that can be run using containers
- Expose endpoints and invoked using HTTP requests
- Support for running jobs
- Great support for concurrent requests

# Cloud Functions vs. Cloud Run

## Cloud Functions

- Choose Cloud Functions if you primarily want to connect to other cloud services on Google Cloud

## Cloud Run

- Choose Cloud Run if you want a simple way to scale and maintain services using containers

O'REILLY®

# Hands on Demos - Google Cloud Functions

# Serverless Applications

When would you choose to use Cloud Functions over Cloud Run?

1. When you need to run a containerized application.
2. When you need to run a function in response to events.
3. When you require fine-grained control over application resources.
4. When you need to deploy a long-running application.

# Serverless Applications

When would you choose to use Cloud Functions over Cloud Run?

1. When you need to run a containerized application.
2. **When you need to run a function in response to events.**
3. When you require fine-grained control over application resources.
4. When you need to deploy a long-running application.

O'REILLY®

# Networking on the Google Cloud

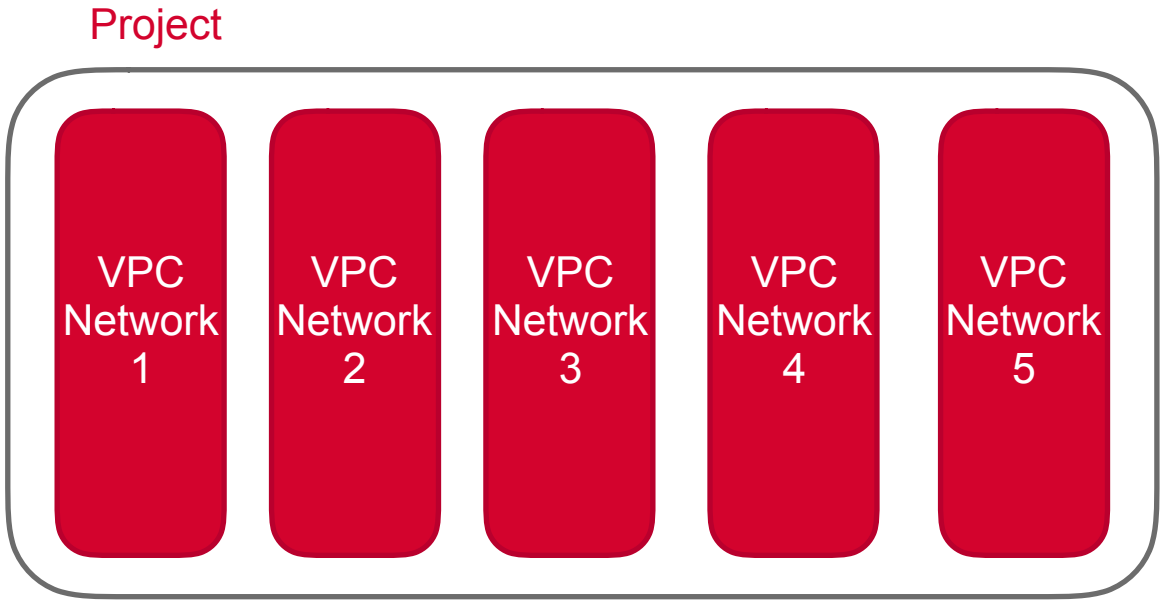
# Google Virtual Private Cloud

A VPC network, or just network, is a global, private, isolated virtual network partition that provides managed network functionality on the Google Cloud

# Google Virtual Private Cloud

A VPC network, often just called a network, is a global, private, isolated virtual network partition that provides managed network functionality on Google Cloud

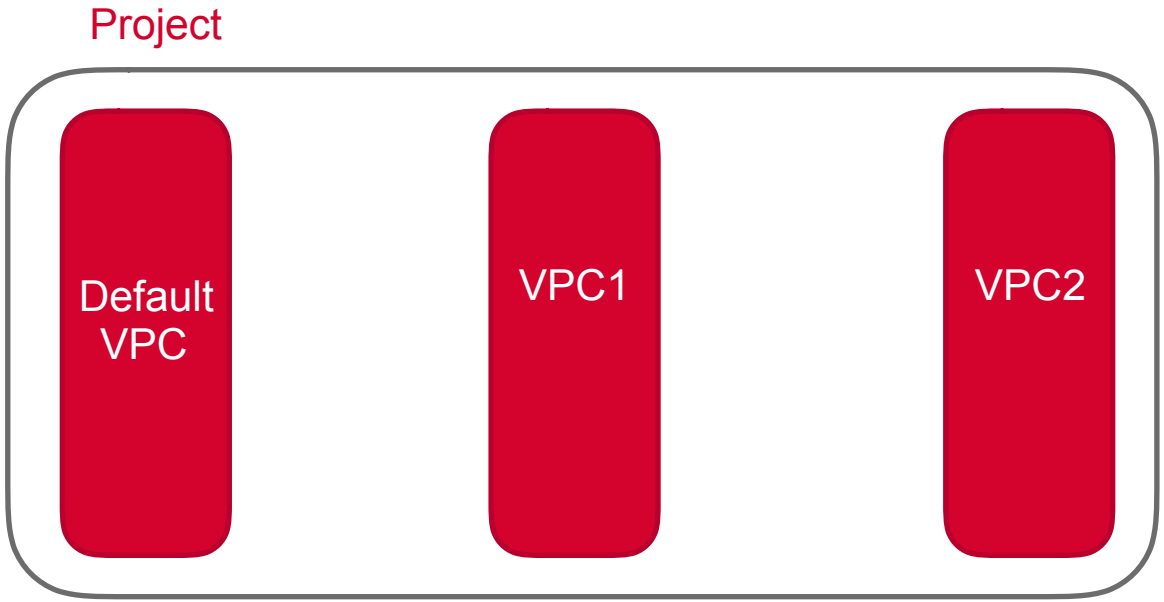
# Multiple VPCs in a Project



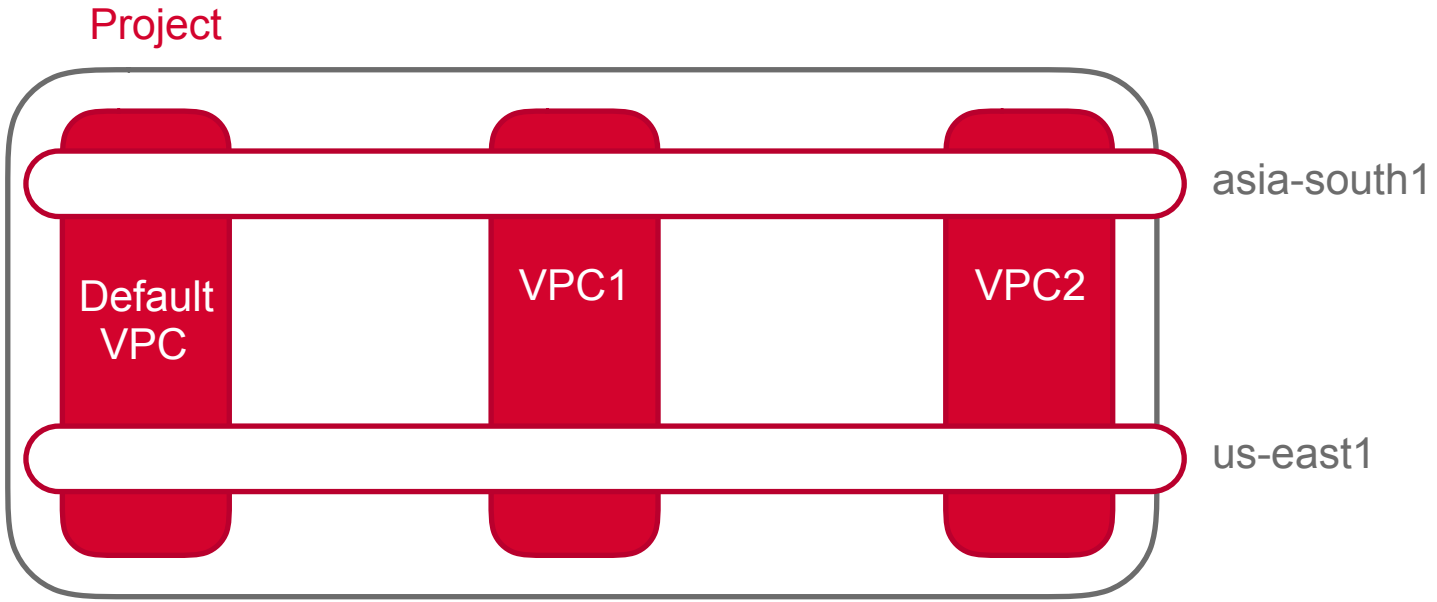
# Projects and VPCs

- VPCs are global resources on Google Cloud
- Each VPC must exist inside a project
- Default VPC pre-created in each project
- Can add additional VPCs
  - Auto Mode
  - Custom Mode

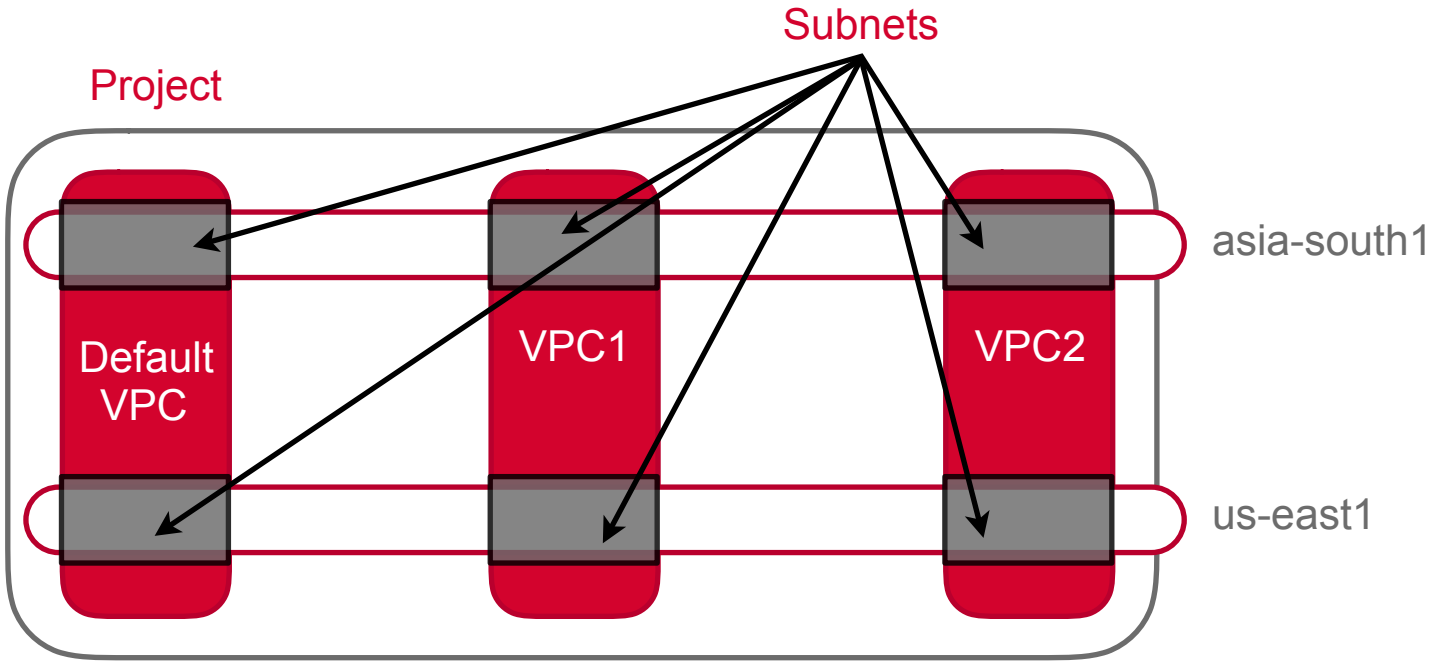
# VPCs Are Global



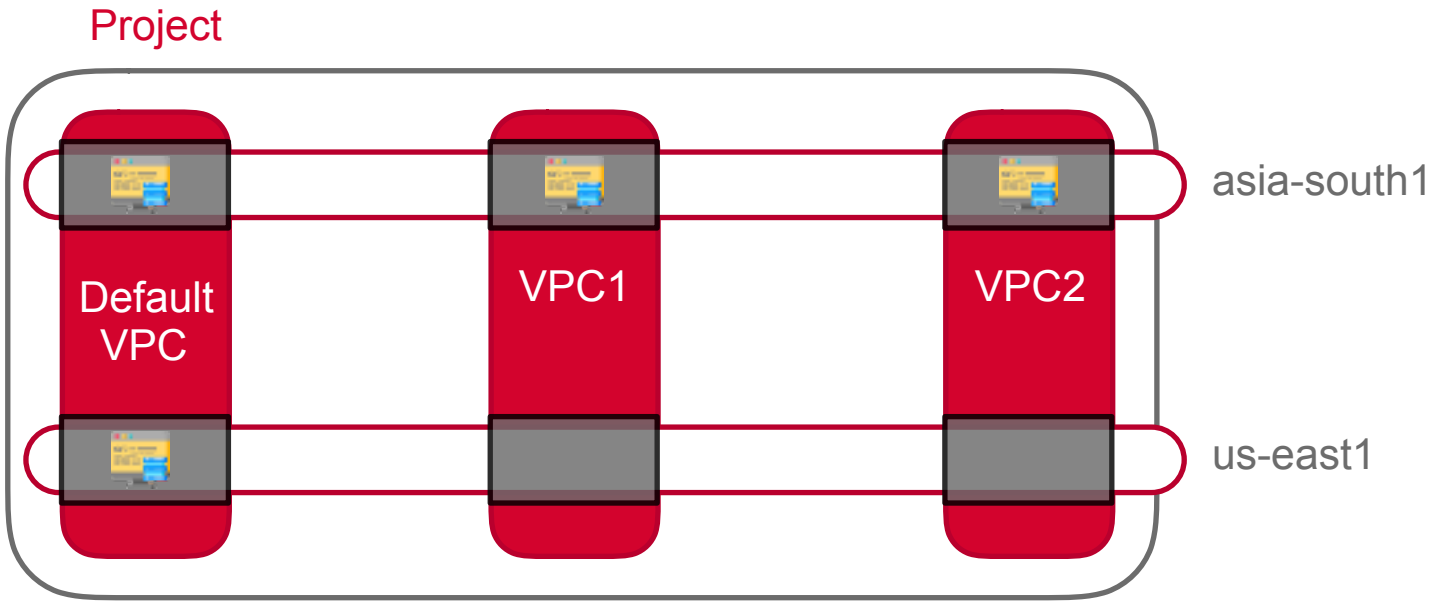
# VPCs Are Global



# Subnets in Each Region



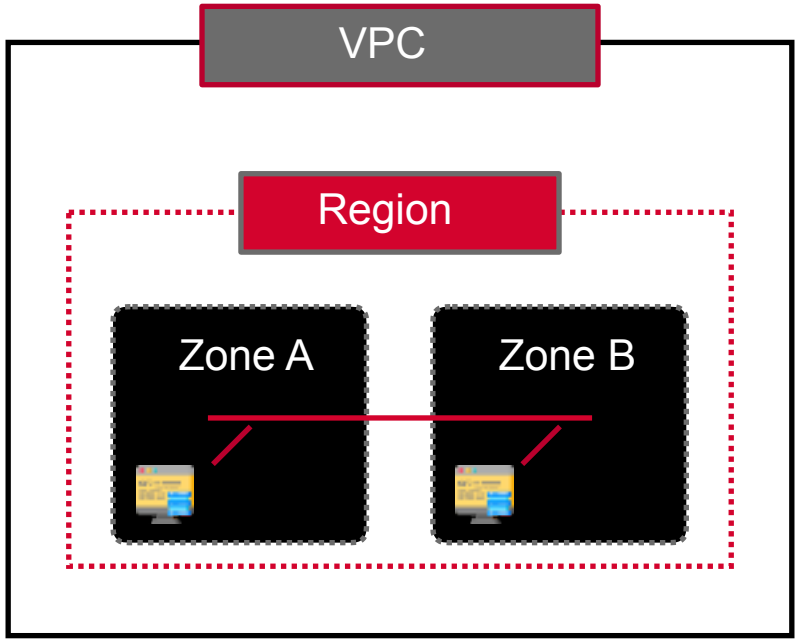
# Resources Provisioned on Subnets



# Subnets

- **IP range partitions** within global VPCs
- VPCs have no IP ranges
- Subnets are regional - can span zones inside a region
- Network has to have at least one subnet before you can use it

# Subnets Span Zones



# Subnets and IP Ranges

- Each subnet must have primary address range
- Valid RFC 1918 CIDR block
- Subnet ranges in **same network cannot overlap**
- Subnet ranges in **different networks can overlap**

# AutoMode and CustomMode VPCs

## Auto Mode

Subnets automatically created in each region, default firewall rules

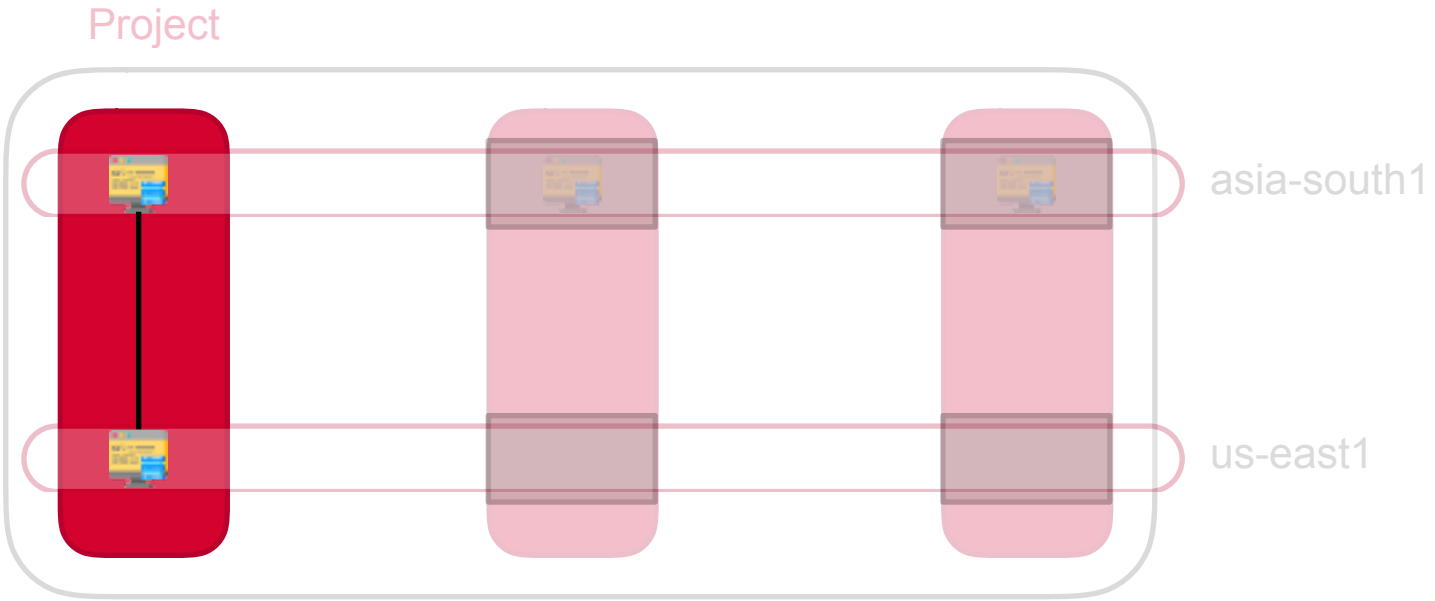
## Custom Mode

Manually create subnets in regions, no defaults preconfigured

# AutoMode and CustomMode VPCs

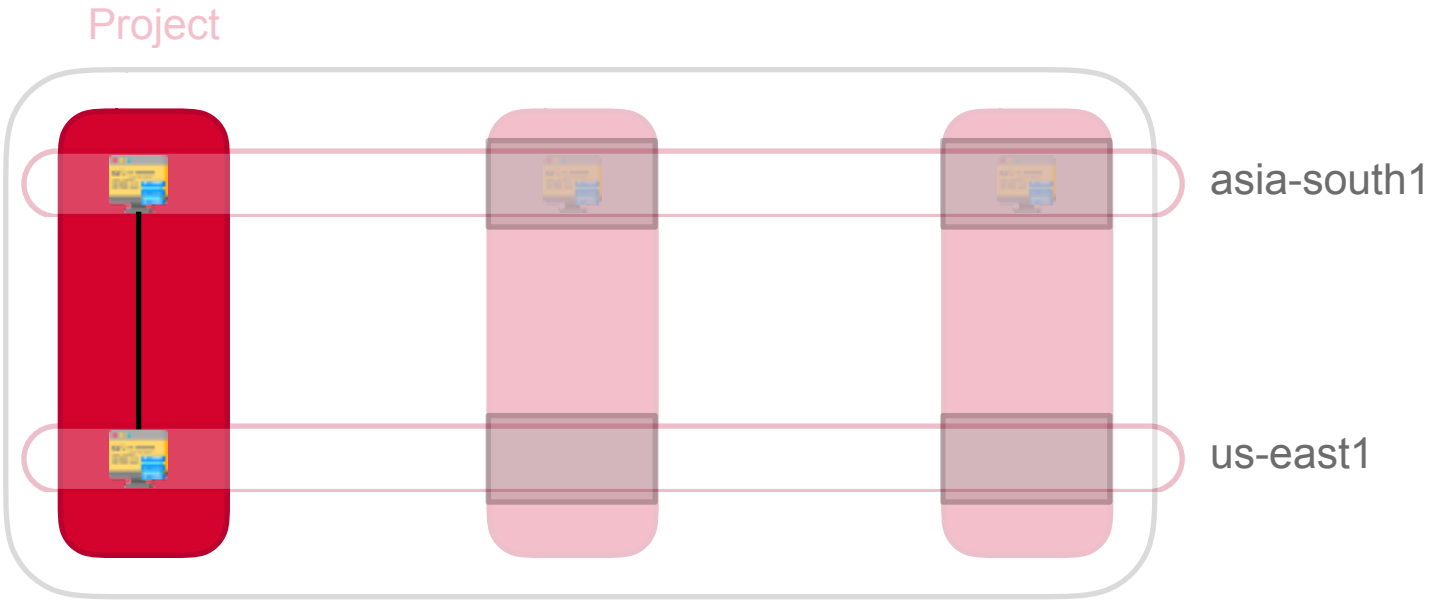
- Auto Mode VPCs have pre-created subnets
  - One in each Google Cloud region
- Custom Mode VPCs start with no subnets
  - Full control over which regions have subnets
  - Can create multiple subnets in a region

# Communication on VPCs



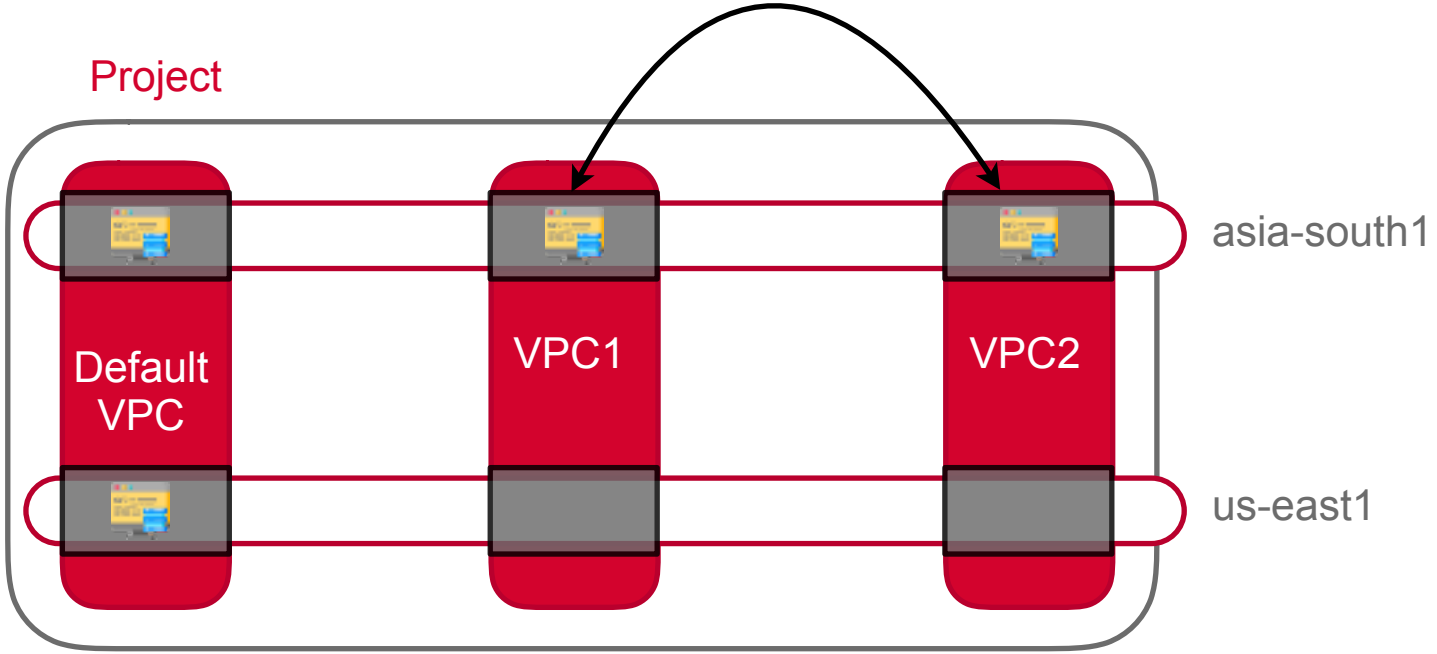

Resources within a VPC communicate using private IP addresses

# Communication on VPCs



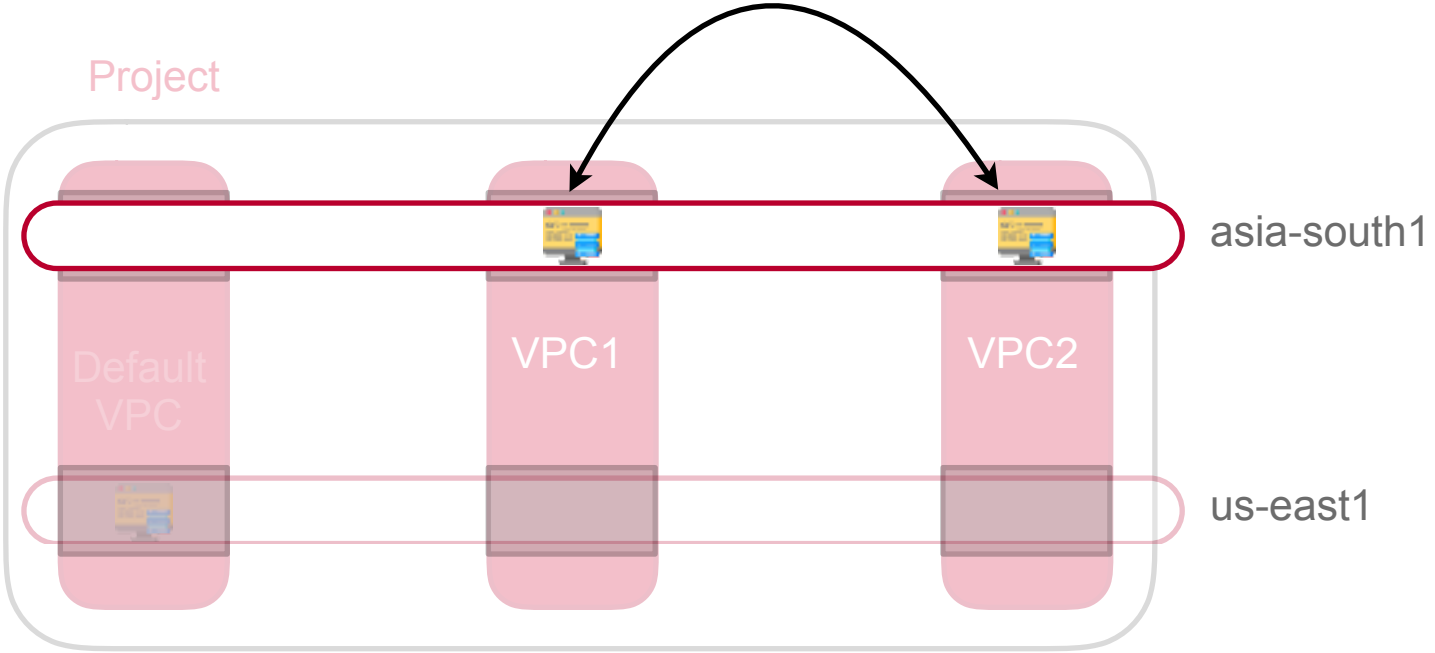

Wherever they are located in the world -
irrespective of physical location

# Communication on VPCs




Resources on different VPCs communicate over the internet using external IPs

# Communication on VPCs




Even though they are in the same region - they may even be in the same zone on the same physical hardware

# Default VPC

- Pre-created on every project
- Includes subnet for each Google Cloud region
- New subnets added when new regions are created
- Resources created here by default

# Default VPC

- Includes routes for all resources
- All VMs on the default VPC can talk to each other
- Default gateway to internet
- Includes several firewall rules

# Firewall Rules

- Every VPC is a distributed firewall
- Firewall rules defined in VPC
- Are applied on per-instance basis
- Can also regulate internal traffic

# Firewall Rules

- Every VPC has two permanent rules
  - Implied allow egress
  - Implied deny ingress
- Can be overridden by more specific rules
- In addition, default VPC has several rules

# Additional Rules in Default VPC

- default-allow-internal
- default-allow-ssh
- default-allow-rdp
- Default-allow-icmp

O'REILLY®

# Hands on Demos – VPC Networks


O'REILLY®
Connecting Networks
Insert subtitle here...

# Shared VPC

- Share VPC across projects on GCP
- One VPC shared across projects
- Projects must be in the same organization
- Host project, guest resources
- Shared VPC admin to administer the shared VPC

# VPC Peering

- Two or more VPCs shared across projects
- Projects need not be in the same organization
- Allows resources on different VPC networks to communicate using internal IP addresses
- Resources on the network use Google infrastructure to communicate
- Reduced latency, higher security and lower cost as compared with using external IPs

# Shared VPCs vs. Network Peering

## Shared VPCs

- Only within **same organization**
- One VPC used across projects
- Host and service projects not peers
- Only single level of  sharing possible

## Network Peering

- Across **organization boundaries**
- Multiple VPCs share resources
- Connected VPCs are peers
- Multiple levels of peering possible

# Interconnecting Networks

**GCP-to-GCP**

VPC Network Peering

**Enterprise connectivity**

Peering and interconnect options

# Interconnecting Networks

GCP-to-GCP

VPC Network Peering

**Enterprise connectivity**

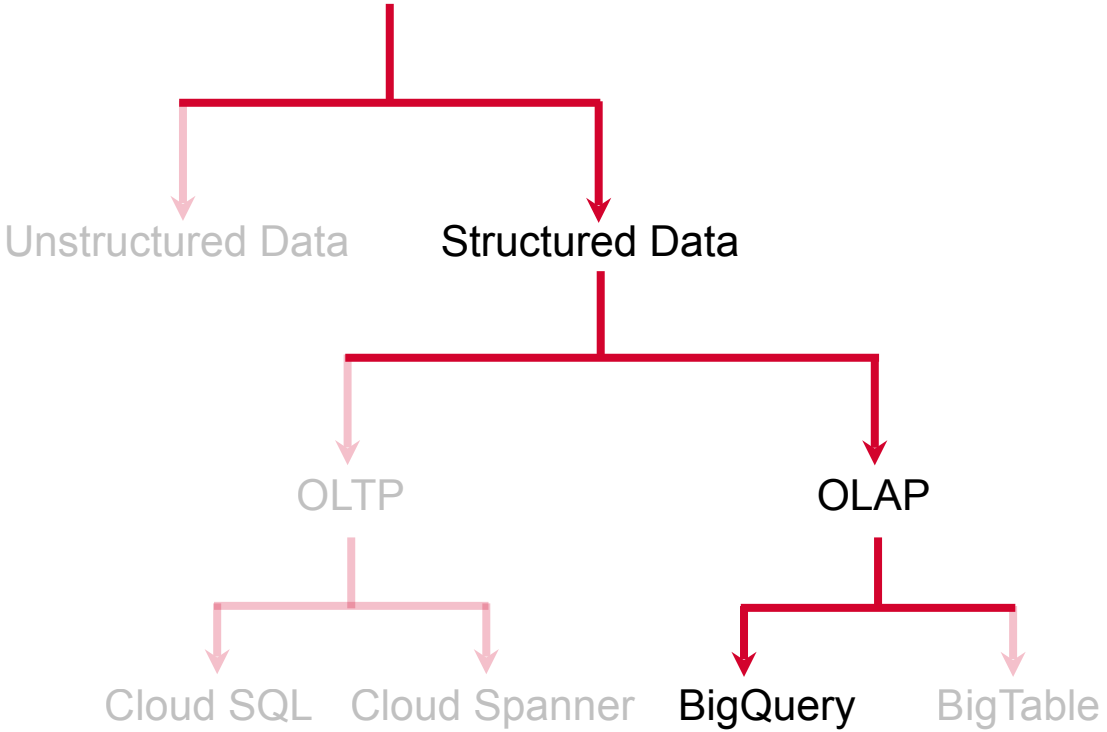
Peering and interconnect options

Connect a cloud network with an on-premise network using private or public IP addresses - VPNs, Cloud Interconnect

O'REILLY®

# BigQuery

# Storage Technologies



# BigQuery is a Data Warehouse that is hard to tell apart from an RDBMS

# BigQuery vs. Traditional Data Warehouses

## BigQuery

- Complex analytical queries
- Scales to Petabytes
- Both reads and updates
- Real-time or batch access
- Multiple data sources
- Streaming as well as batch

## Traditional Data Warehouse

- Complex analytical queries
- Scales to Petabytes
- Mostly reads
- Long running jobs
- Multiple data sources
- Often more focus on batch

# BigQuery Features

- Serverless: No cluster, no provisioning
- Autoscaling
- Automatic high availability

# Support for the 4Vs

- Volume: Scales to Petabytes
- Variety: Federated data sources
  - Cloud storage
  - BigTable
  - Google Drive spreadsheets
- Velocity
  - Streaming ingestion
  - Real-time queries
- Variability
  - Schema auto-detection

# SQL Support

- Standard SQL
  - ANSI:2011 compliant
  - Extensions for nested/repeated fields

# BigQuery

**Datasets**

**Tables**

**Views**

# BigQuery Dataset

Top-level container used to organize and control access to tables and views. A table or view must belong to a dataset.

# BigQuery Table

Contains individual records organized in rows. Each record is composed of columns (also called fields).

# BigQuery View

Virtual table defined by a SQL query. Whenever a user queries the view, the underlying view-query is executed.

# Advantages of Views

- Reduce query complexity
- Restrict access to data
- Construct different logical tables from the same physical table


# Hands on Demos - BigQuery

Insert subtitle here...

# BigQuery

Which of the following statements about BigQuery is true?

1. BigQuery tops out after storing terabyte size data
2. BigQuery does not need cluster provisioning and set up
3. BigQuery offers transaction support at scale
4. BigQuery does not allow partitioning of data

# BigQuery

Which of the following statements about BigQuery is true?

1. BigQuery tops out after storing terabyte size data
2. **BigQuery does not need cluster provisioning and set up**
3. BigQuery offers transaction support at scale
4. BigQuery does not allow partitioning of data