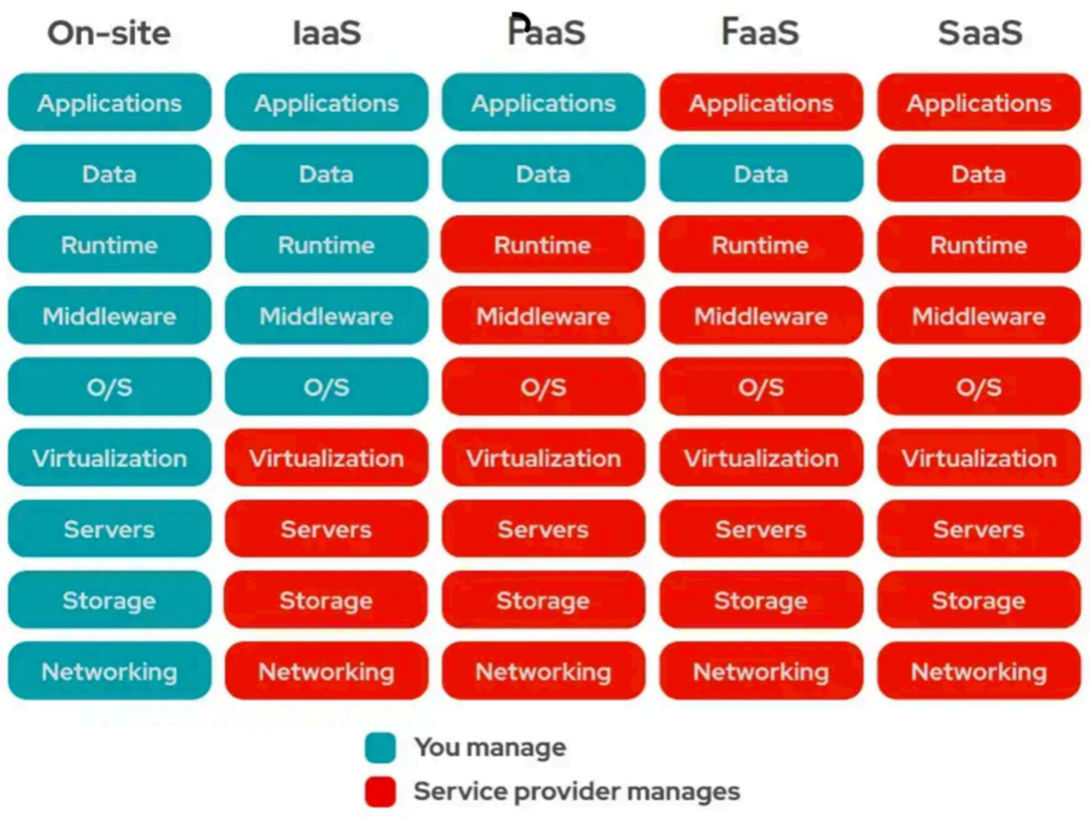
Managed Services in Google Cloud

- We Should understand some terminology used with cloud services:
  - IaaS (Infrastructure as a Service)
  - PaaS (Platform as a Service)
  - FaaS (Function as a Service)
  - CaaS (Container as a Service)
  - SaaS (Software as a Service)



On-site	IaaS	PaaS	FaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage
🟥 Service provider manages

IAAS - Infrastructure As A Service

- IaaS stand for Infrastructure as a Service.Infrastructure as a Service is a form of cloud computing that provides virtualized computing services over the internet to its users.
- So, virtualization uses software to create an abstract layer over a computer hardware.

- Now, with Infrastructure as a Service, as the user you're responsible to manage the application, data, runtime, middleware, and operating system, while the service provider manages the virtualization, servers, storage, and networking.

PAAS - Platform As A Service

- PaaS stand for - PaaS stands for Platform as a Service. Platform as a Service is a service that offers a complete dev and deployment environment for the cloud.
- This provides a model where you only have to worry about deploying your app to the end user while everything else is taken care of.

SAAS Software As A Service

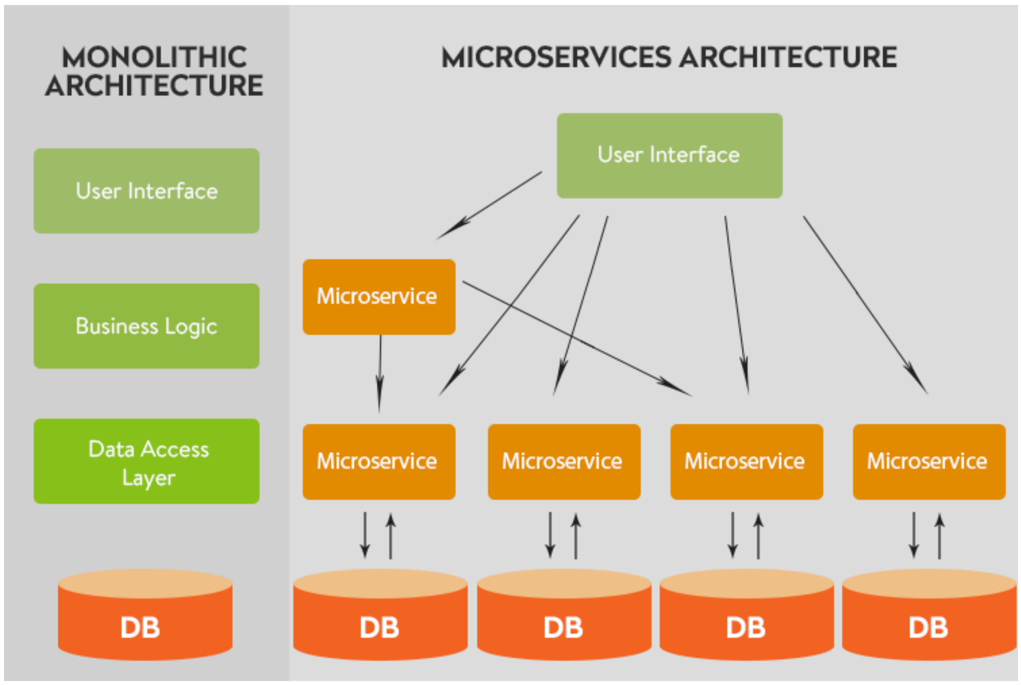
- SaaS stands for a Software as a Service. Software as a Service is a way companies deliver applications or software over the internet to the end user.
- Using a SaaS product, everything is already managed for you. You do not have to manage anything at all.
- The service provider manages everything from the application, the data, the runtime, the middleware, the operating system, the virtualization, servers, storage, and networking.

FAAS - Function As A Service

- FaaS stands for Function as a Service.Function as a Service is a cloud computing service that allows you to execute code in response to events.
- It uses the serverless framework, think of a function in programming, where the function takes data, process it, responds, or returns a result. This same thing is happening here.

Microservices

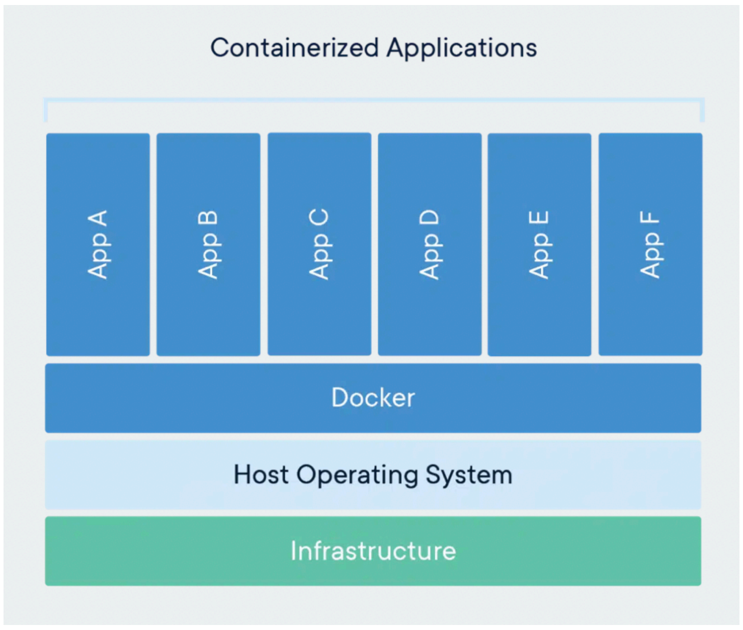
- The microservices architecture enables an organization to deliver large, complex applications rapidly, frequently, reliably and sustainably - a necessity for competing and winning in today’s world.



- A microservices architecture is a variant of the service-oriented architecture structural style.
- Flexibility to innovate and build applications in different programming languages (Go, Java, Python, JavaScript, etc) but deployments become complex. That's where **Containers** comes to rescue.

Containers

- A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.
- Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.
- A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.
- Docker Create Docker images for each microservices. Docker image has all needs of a microservices.
- Docker containers that run on Docker Engine:



•Standard: Docker created the industry standard for containers, so they could be portable anywhere

•Lightweight: Containers share the machine's OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs

- Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

- Advantages
    - Docker containers are light weight
    - Compared to Virtual Machines as they do not have a Guest OS Docker provides isolation for containers
    - Dockers are cloud neutral.

Container Orchestration

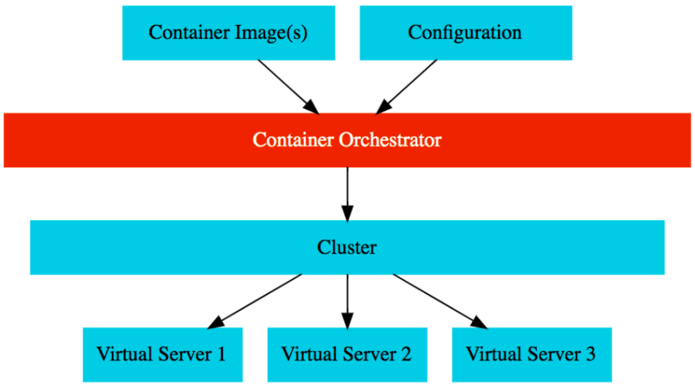
- Container orchestration is the automation of much of the operational effort required to run containerized workloads and services.
- This includes a wide range of things software teams need to manage a container's lifecycle, including provisioning, deployment, scaling (up and down), networking, load balancing and more.

What are the benefits of container orchestration?

Container orchestration is key to working with containers, and it allows organizations to unlock their full benefits. It also offers its own benefits for a containerized environment, including:

- Simplified operations

- Resilience
- Added security
- Portability
- Application development
- Resource utilization and optimization



What is Kubernetes container orchestration?

- Kubernetes is a popular open source platform for container orchestration.
- It enables developers to easily build containerized applications and services, as well as scale, schedule and monitor those containers.
- Kubernetes provides extensive container capabilities, a dynamic contributor community, the growth of cloud-native application development and the widespread availability of commercial and hosted Kubernetes tools.
- Kubernetes is also highly extensible and portable, meaning it can run in a wide range of environments and be used in conjunction with other technologies, such as service meshes.

Container orchestration versus Docker

- Docker is a specific platform for building containers, including the Docker Engine container runtime, whereas container orchestration is a broader term referring to automation of any container's lifecycle.
- Docker also includes Docker Swarm, which is the platform's own container orchestration tool that can automatically start Docker containers.

What is a server?

- The technical definition of a server is a computer or device that provides programmable service to another computer or its user.

- For example if you wanted to connect to your Google account. You're actually connecting to a Google server that could be provisioned in the cloud with a specific CSP (or cloud service provider) like AWS, Azure, or even Google Cloud.
- After connecting to the server, it allows you to have the experience dedicated to the server and what it's trying to provide for you.

What is serverless?

- Serverless is a cloud computing execution model in which the cloud service provider allocates machine resources on demand. In other words, serverless does not mean the absence of servers.
- Serverless architecture refers to delegating server management and maintenance to a third-party cloud provider.
- Serverless is the framework that makes that happen. So the user, can focus on developing products and writing code while the cloud vendor takes care of the server infrastructure.
- The provider will allocate everything you need in response to the activity within your code or function

GCP Managed Services for Compute

Service	Details	Category
Compute Engine	High-performance and general purpose VMs that scale globally	IaaS
Kubernetes Engine	Orchestrate containerized microservices on Kubernetes Needs advanced cluster configuration and monitoring	IaaS
App Engine	Build highly scalable applications on a fully managed platform using open and familiar languages and tools	PaaS
Cloud Run	Develop and deploy highly scalable containerized applications. Basically Serverless!	CaaS(Serverless)
Cloud Functions	Build event driven applications using simple, single-purpose functions	FaaS(Serverless)

Google Cloud - Compute Engine(IAAS)

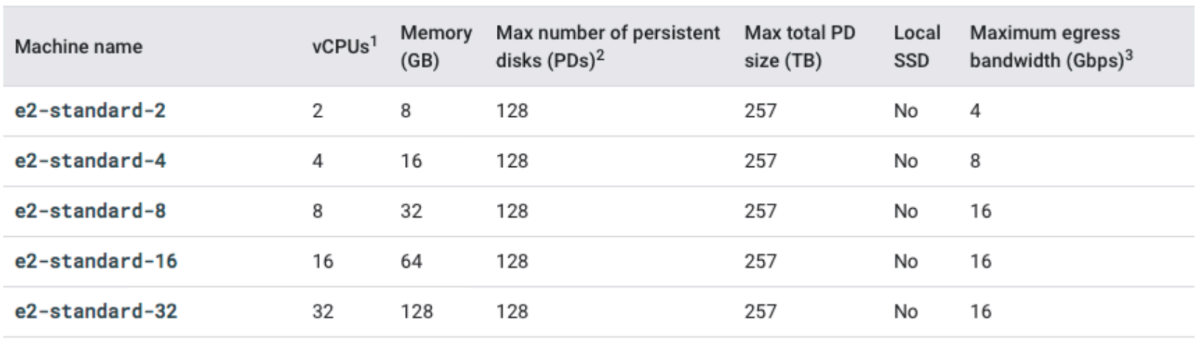
What Are Virtual Machines?

- Virtual Machines is a virtualized instance of a computer that can perform almost all the functions as that of a computer.
- Virtual machines run on a physical machine and access computing resources from the software which is known as a hypervisor.

Compute Engine

- **Google Compute Engine** is a part of Google's IaaS (Infrastructure as a Service) service that provides virtual machines (VMs).
- Compute Engine is a computing and hosting service that lets you create and runs variety of virtual machines on Google infrastructure.

- Compute Engine offers scale, performance, and value that lets you easily launch large compute clusters on Google's infrastructure instead of acquiring and managing server hardware.
- There are no upfront investments, and you can run thousands of virtual CPUs on a system that offers quick, consistent performance.
- Google Compute Engine offers virtual machines running in Google's data centers connected to the worldwide fibre network.
- The tooling and workflow offered by compute engine enable scaling from single instances to global, load-balanced cloud computing.
- VM Instances are comprised of on Operating System and infrastructure resources such as CPU, Memory, Disk, and Networking.
- **Machine Types-** Machines types are templates of virtualized hardware that will be available to the VM instance. These resources include the CPU, Memory, Disk capabilities, and so on.
- Different Machine Families for Different Workloads like Web and application servers, Small-medium databases, Dev environments
    - General Purpose (E2, N2, N2D, N1) : Best price-performance ratio
    - Memory Optimized (M2, M1): Ultra high memory workloads Large in-memory databases and In-memory analytics
    - Compute Optimized (C2): Compute intensive workloads Gaming applications



Machine name	vCPUs[1]	Memory (GB)	Max number of persistent disks (PDs)[2]	Max total PD size (TB)	Local SSD	Maximum egress bandwidth (Gbps)[3]
`e2-standard-2`	2	8	128	257	No	4
`e2-standard-4`	4	16	128	257	No	8
`e2-standard-8`	8	32	128	257	No	16
`e2-standard-16`	16	64	128	257	No	16
`e2-standard-32`	32	128	128	257	No	16

- Memory, disk and networking capabilities increase along with vCPUs

- Disks - The disk you choose will be your single root disk in which your image is loaded during the boot process.
    - Persistent Disks volumes provide high-performance and redundant network storage. Data is durable, meaning the data will remain as you left it after reboots and shutdowns.
    - Local SSDs are physical drives attached directly to the same server as your VM. They can offer better performance, but are ephemeral.
    - Hyperdisk volumes offer the fastest redundant network storage for Compute Engine, with configurable performance and volumes that can be dynamically resized.
    - You can also use Cloud Storage buckets and Filestore with your VMs.
- Images - Images contain a bootloader, Operating System, file system structure, and any software customizations needed for your deployment.The image describes what actually gets loaded onto the root disk.
- Type of Images
    - Public Images: Provided & maintained by Google or Open source communities or third party vendors
    - Custom Images: are available only to your Cloud project.
- Internal and External IP address:
    - External (Public) IP addresses are Internet addressable.
    - Internal (Private) IP addresses are internal to a corporate network
    - All **VM instances** are assigned at least one Internal IP address
    - Creation of external IP address can be enabled for VM instances.
- Preemptibility - Preemptible VM is an affordable, short-lived instance ideal for batch jobs or fault-tolerant workloads.

- They’re up to 80% cheaper than regular instances, so if your application can handle random the termination of VMs at any time, then this is best option.

Applications Of Compute Engine

Some of the use-cases or applications of the Google compute engine:

- Virtual Machine (VM) migration to Compute Engine
- Genomics Data Processing
- BYOL or Bring Your Own License images

Advantages Of Compute Engine

- **Storage Efficiency:** The persistent disks support up to 257 TB of storage which is more than 10 times higher than what Amazon Elastic Block Storage (EBS) can accommodate.
- **Cost:** Within the GCP ecosystem, users pay only for the computing time that they have consumed. The per-second billing plan is used by the Google compute engine.
- **Stability:** It offers more stable services because of its ability to provide live migration of VMs between the hosts.
- **Backups:** Google Cloud Platform has a robust, inbuilt, and redundant backup system. The Compute Engine uses this system for flagship products like Search Engine and Gmail.
- **Scalability:** It makes reservations to help ensure that applications have the capacity they need as they scale.
- **Security:** Google Compute Engine is a more secure and safe place for cloud applications.

Benefits Of Compute Engine

- **Easy Integration:** It allows to easily integrate with other Google Cloud services like AI/ML and data analytics.

- **Compute globally as per requirement:** Rendering reservations for supporting and securing applications having the strength as per measurement and requirement.
- **Gain Infinite Value:** Preventing cost only for executing Compute with sustained-use discounts, and obtaining huge profits while implementing devoted-use discounts.
- **Confidential Computing:** Confidential VMs is a kind of advanced technology that enables users to encode delicate data into the cloud while data is being processed.

Google Cloud - App Engine(PAAS)

- Google App Engine is a Platform as a Service (PaaS) product that allows developers to focus on what they do best: writing code. App Engine automatically scales your app up and down while also balancing the load with access to Google's scalable hosting.
- App Engine makes deployment, maintenance, Scalability easy so that you can focus on Innovation. Provides end-to-end application management
- Especially suitable for building scalable web Applications & Mobile backends.
- Supports - Go, Java, .NET, Node.js, PHP, Python, Ruby using pre-configured runtimes
- Use custom run-time and write code in any language
  Connect to variety of Google Cloud storage products (Cloud SQL etc)
- Google App Engine is free up to a certain amount of resource usage.

Features Of App Engine

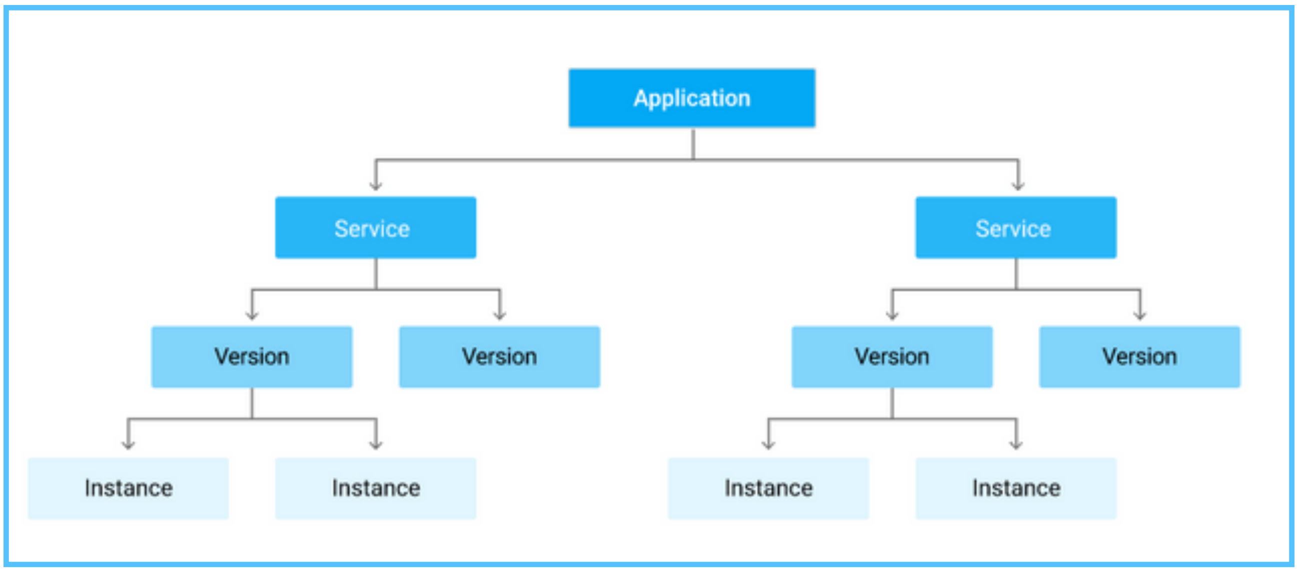
- Automatic load balancing & Auto scaling
- Managed platform updates & Application health monitoring
- Application versioning

- Traffic splitting

App Engine Environments

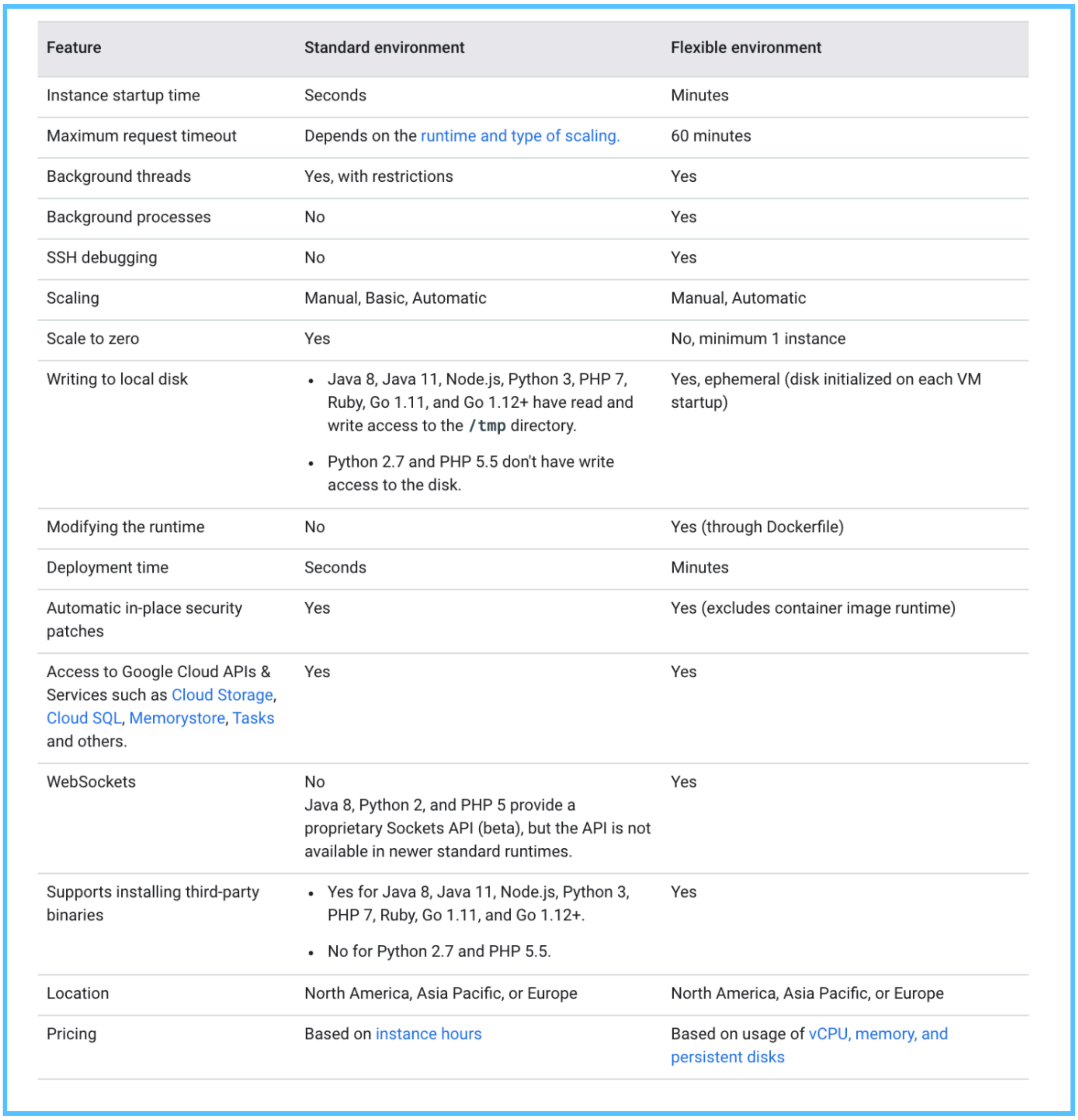
- App Engine **standard environment** is based on specific language sandboxes running on Google's infrastructure.
- App Engine standard environment makes it easy to build and deploy an application that runs reliably even under heavy load and with large amounts of data.
- Standard environment supports the following languages - Java, Python, Node.js, PHP, Ruby, Go.
- **Flexible App Engine** overcome the constraints of Standard App Engine. In flex App Engine, You can customize runtimes or provide your own runtime by supplying a custom Docker image or Docker x`file from the open source community.
- Features of Flexible App Engine
    - Customizable infrastructure
    - Performance options
    - Native feature support
    - Managed virtual machines

App Engine - Application Hierarchy



- Application - An application is like a container for everything.
- Service - Services are the microservices or the app components that you can run on App Engine. A single application can contain multiple services.
- Version - Versions are simply specific versions of your service which are accompanied by code and configuration. At the same time, there can be multiple versions..

App Engine Comparison

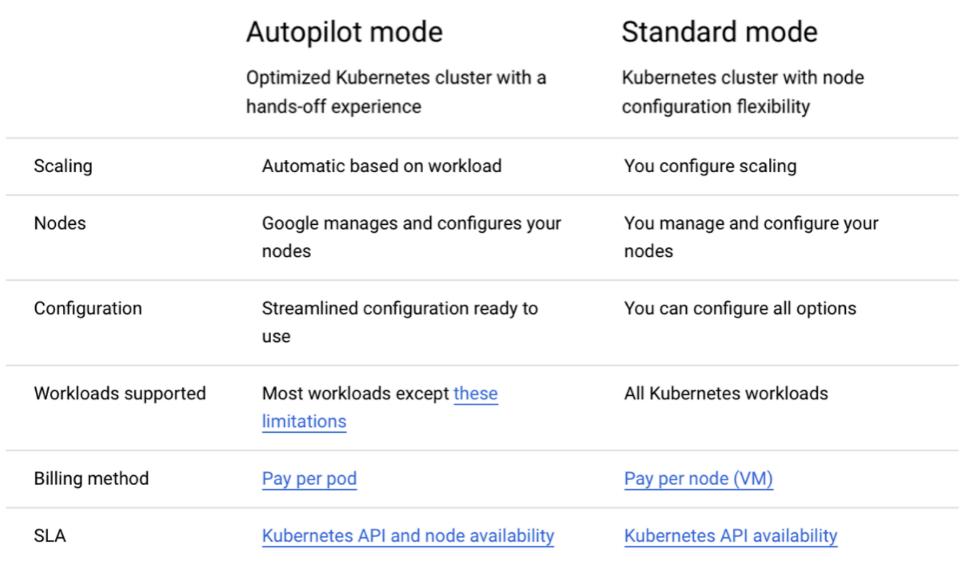


Feature	Standard environment	Flexible environment
Instance startup time	Seconds	Minutes
Maximum request timeout	Depends on the runtime and type of scaling.	60 minutes
Background threads	Yes, with restrictions	Yes
Background processes	No	Yes
SSH debugging	No	Yes
Scaling	Manual, Basic, Automatic	Manual, Automatic
Scale to zero	Yes	No, minimum 1 instance
Writing to local disk	• Java 8, Java 11, Node.js, Python 3, PHP 7, Ruby, Go 1.11, and Go 1.12+ have read and write access to the `/tmp` directory. • Python 2.7 and PHP 5.5 don't have write access to the disk.	Yes, ephemeral (disk initialized on each VM startup)
Modifying the runtime	No	Yes (through Dockerfile)
Deployment time	Seconds	Minutes
Automatic in-place security patches	Yes	Yes (excludes container image runtime)
Access to Google Cloud APIs & Services such as Cloud Storage, Cloud SQL, Memorystore, Tasks and others.	Yes	Yes
WebSockets	No Java 8, Python 2, and PHP 5 provide a proprietary Sockets API (beta), but the API is not available in newer standard runtimes.	Yes
Supports installing third-party binaries	• Yes for Java 8, Java 11, Node.js, Python 3, PHP 7, Ruby, Go 1.11, and Go 1.12+. • No for Python 2.7 and PHP 5.5.	Yes
Location	North America, Asia Pacific, or Europe	North America, Asia Pacific, or Europe
Pricing	Based on instance hours	Based on usage of vCPU, memory, and persistent disks

Google Cloud - Kuberentes Engine(IAAS)

- GKE : **Kubernetes** is an open source solution for managing application containers. It is Managed Kubernetes service
- Kubernetes : Kubernetes is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.
- Kubernetes was created by Google for its own container orchestration purposes.
- GKE - is a cluster manager and orchestration system for running Docker containers in the cloud.
- GKE a production-ready environment with guaranteed uptime, load balancing and included container networking features. It allows you to create multiple-node clusters while also providing access to all Kubernetes' features.
- It comes with four-way auto-scaling and multi-cluster support.
- GKE comes in two types of mode

Autopilot Vs Standard



	Autopilot mode Optimized Kubernetes cluster with a hands-off experience	Standard mode Kubernetes cluster with node configuration flexibility
Scaling	Automatic based on workload	You configure scaling
Nodes	Google manages and configures your nodes	You manage and configure your nodes
Configuration	Streamlined configuration ready to use	You can configure all options
Workloads supported	Most workloads except these limitations	All Kubernetes workloads
Billing method	Pay per pod	Pay per node (VM)
SLA	Kubernetes API and node availability	Kubernetes API availability

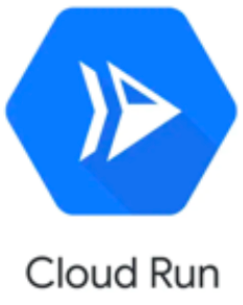
- With Kubernetes, you can decide when your containers should run, increase, or decrease the size of application containers or check the resource consumption of your application deployments.
- Provides Cluster Management and also provides all important container orchestration features:
    - **Load balancing** for Compute Engine instances
    - Node pools to designate subsets of nodes within a cluster for additional flexibility
    - **Automatic scaling** of your cluster's node instance count
    - **Automatic upgrades** for your cluster's node software
    - Node **auto-repair** to maintain node health and availability
    - Logging and Monitoring with Cloud Monitoring for visibility into your cluster
- Minimize operations with auto-repair (repair failed nodes) and auto-upgrade (use latest version of K8S always) features
- Provides Pod and Cluster Autoscaling.
- Enable Cloud Logging and Cloud Monitoring with simple and also Integrated with gcp infrastructure, services, and Kubernetes-specific views.
- A high-availability control plane including multi-zonal and regional clusters. GKE Cluster Types:
    - Zonal Cluster
        - Single zone
        - Multi-zonal
    - Regional cluster
    - Private cluster
    - Alpha cluster
- Vulnerability scanning of container images and data encryption

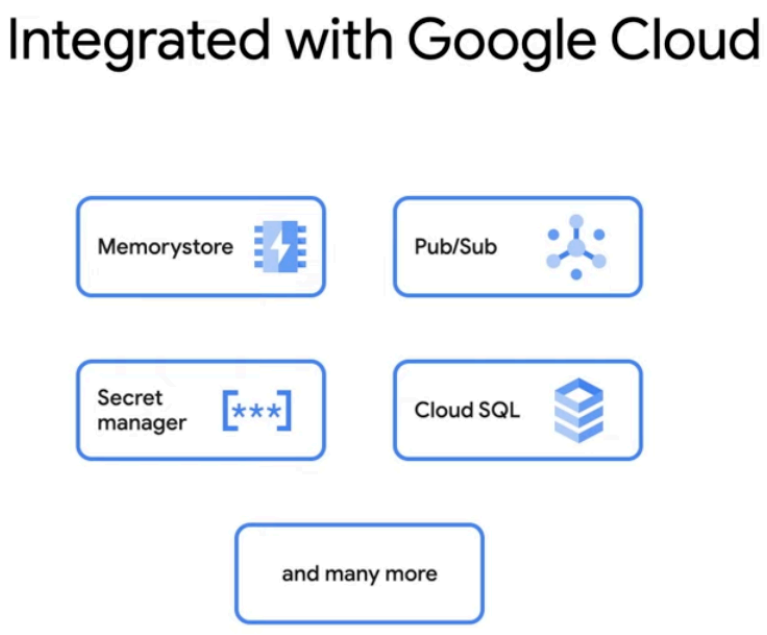
Why Use A Managed GKE Cluster?

- Managed cluster infrastructure
- Minimal Kubernetes knowledge and experience
- Ongoing operations for upgrades, fixes, and security patches
- Micro-service teams

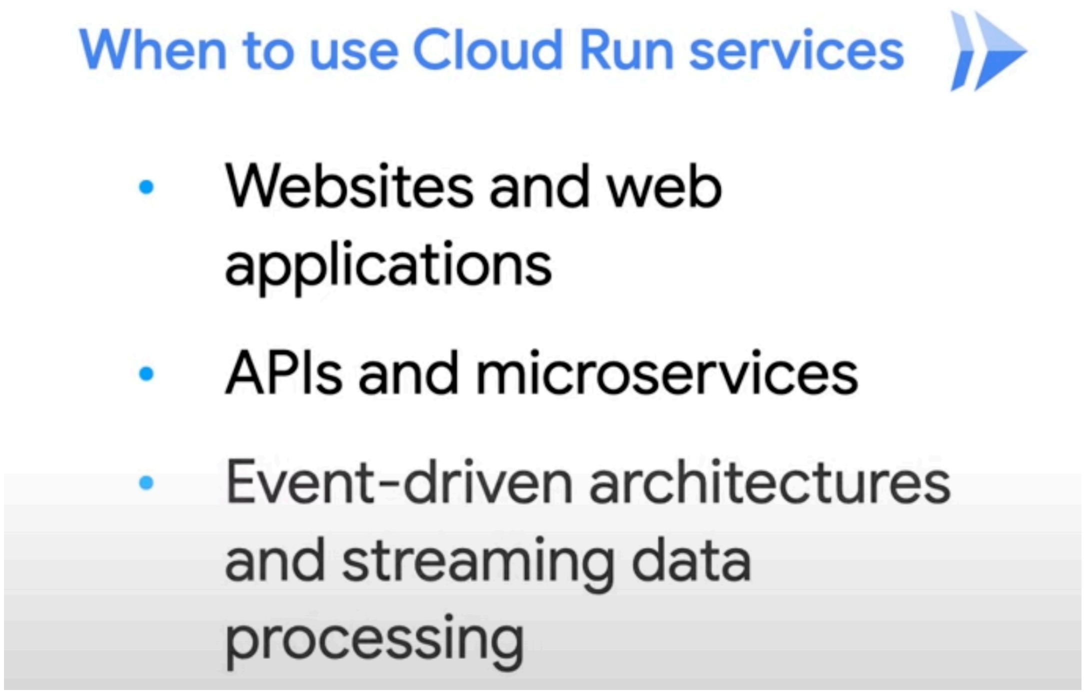
Google Cloud - Cloud Run(CAAS)

- Cloud Run - "Container to Production in Seconds"
- Cloud Run is a fully managed compute platform that lets you run containers directly on top of Google's scalable infrastructure.
- Cloud Run allows developers to spend their time writing their code(Python, Java, Go, C#, .Net and many more) and very little time operating, configuring, and scaling their Cloud Run service.
- You can deploy code written in any programming language on Cloud Run if you can build a container image from it.
- You don't have to create a cluster or manage infrastructure in order to be productive with Cloud Run.





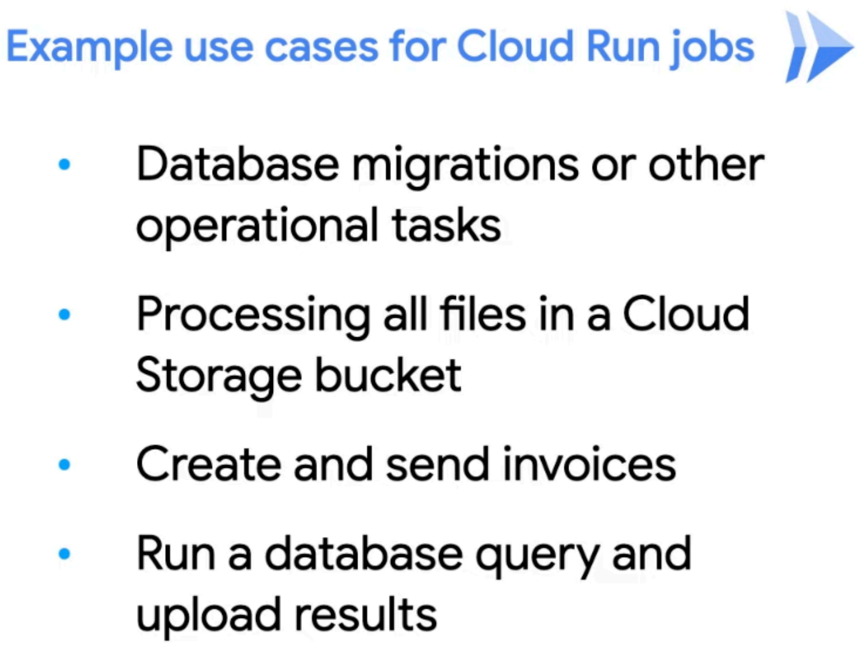
- Built on top of an open standard - **Knative** Fully managed serverless platform for containerized applications.
- Zero infrastructure management Pay-per-use (For used CPU, Memory, Requests and Networking)
- On Cloud Run, your code can either run continuously as a service or as a job.
- Both services and jobs run in the same environment and can use the same integrations with other services on Google Cloud.
  - **Cloud Run services.** Used to run code that responds to web requests, or events.



When to use Cloud Run services

- Websites and web applications
- APIs and microservices
- Event-driven architectures and streaming data processing

- **Cloud Run jobs.** Used to run code that performs work (a job) and quits when the work is done.



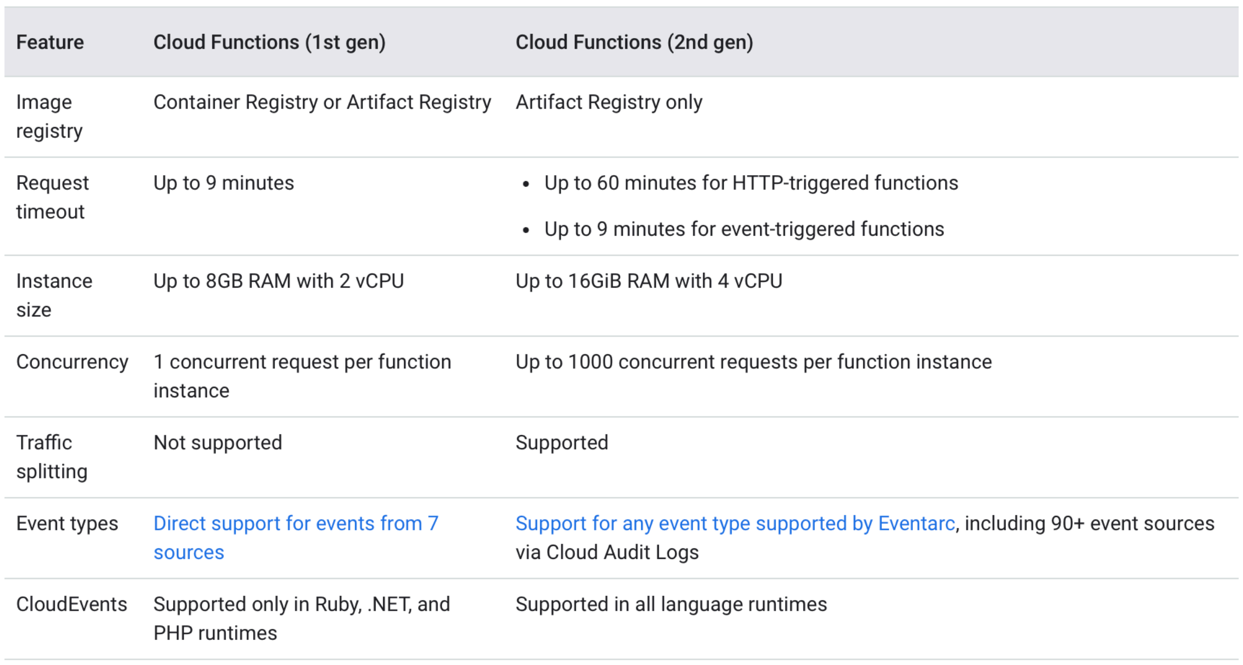
Example use cases for Cloud Run jobs

- Database migrations or other operational tasks
- Processing all files in a Cloud Storage bucket
- Create and send invoices
- Run a database query and upload results

- Fullyintegrated end-to-end developer experience: No limitations in languages, binaries and dependencies
- Easily portable because of container based architecture
- Cloud Code, Cloud Build, Cloud Monitoring & Cloud Logging Integrations
- Leverage your existing Kubernetes investment to run serverless workloads.

Google Cloud - Cloud Function(FAAS)

- Google Cloud Functions is an event-driven serverless compute platform. It automatically spin up and back down in response to events.
- They scale horizontally. It is a piece of code that runs in response to an event.
- Pay only for what you use.
    - Number of invocations
    - Compute Time of the invocations Amount of memory and CPU provisioned
- Cloud functions comes with two versions.



Feature	Cloud Functions (1st gen)	Cloud Functions (2nd gen)
Image registry	Container Registry or Artifact Registry	Artifact Registry only
Request timeout	Up to 9 minutes	• Up to 60 minutes for HTTP-triggered functions • Up to 9 minutes for event-triggered functions
Instance size	Up to 8GB RAM with 2 vCPU	Up to 16GiB RAM with 4 vCPU
Concurrency	1 concurrent request per function instance	Up to 1000 concurrent requests per function instance
Traffic splitting	Not supported	Supported
Event types	Direct support for events from 7 sources	Support for any event type supported by Eventarc, including 90+ event sources via Cloud Audit Logs
CloudEvents	Supported only in Ruby, .NET, and PHP runtimes	Supported in all language runtimes

- Cloud Functions allows you to write your code without worrying about provisioning resources or scaling to handle changing requirements.
- Cloud events are things that happen in your cloud environment. For instances like changes to data in a database, files added to a storage system, or a new virtual machine instance being created.

- Since cloud functions are event-driven, they only run when something happens. This makes them a good choice for tasks that need to be done quickly or that don't need to be running all the time.
- **Event** : Upload object to cloud storage.
- **Trigger:** Respond to event with a Function call
    - **Trigger** - Which function to trigger when an event happens?
    - **Functions** - Take event data and perform action?
- **Events** are triggered from
    - Cloud Storage
    - Cloud Pub/Sub
    - HTTP POST/GET/DELETE/PUT/OPTIONS
    - Firebase - Cloud Firestore
    - Stack driver logging
- For example, you can use a cloud function to:
    - automatically generate thumbnails for images that are uploaded to Cloud Storage.
    - send a notification to a user's phone when a new message is received in Cloud Pub/Sub.
    - process data from a Cloud Firestore database and generate a report.