

ĐIỆN TOÁN ĐÁM MÂY (Cloud Computing)

PRACTICES Practice 7 – Cloud-App Environment

Presenter: **Dr. Nguyen Dinh Long**

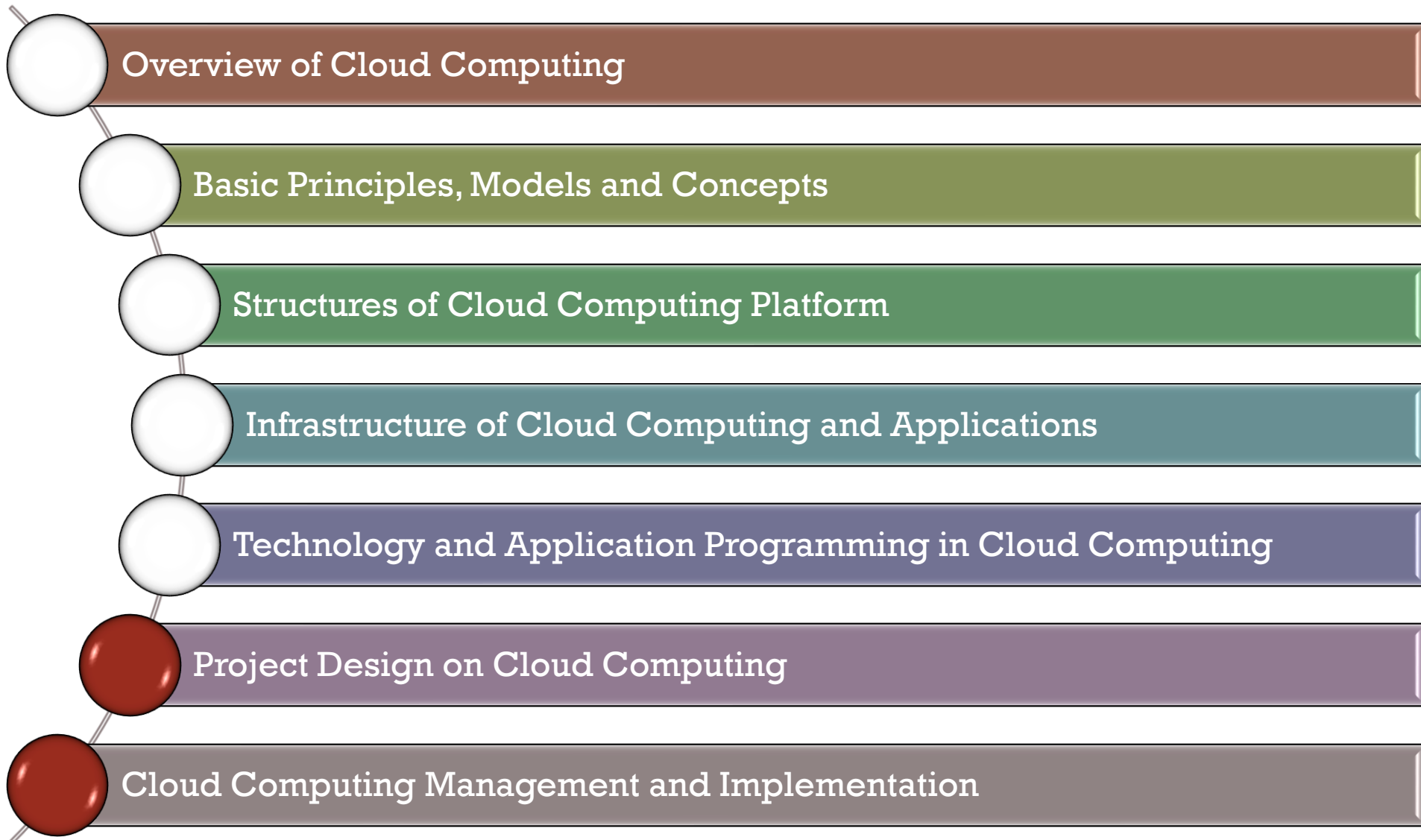
Email: dinhlonghcmut@gmail.com

Phone: +84 947 229 599

Google-site: <https://sites.google.com/view/long-dinh-nguyen>

Dec. 2022

Outline



References

Main:

- Thomas Erl, Zaigham Mahmood, and Ricardo Puttini. 2013. *Cloud Computing Concepts, Technology & Architecture*. Prentice Hall.
- Michael J. Kavis. 2014. *Architecting the Cloud: Design Decisions for Cloud Computing Service Models*. Wiley
- Arshdeep Bahga, and Vijay Madisetti. 2013. *Cloud Computing: A Hands-On Approach*. CreateSpace Independent Publishing Platform

More:

- Rajkuma Buyya, Jame Broberg and Andrzej Goscinski. 2011. *Cloud Computing –Principles and paradigms*, Wiley
- Nick Antonopoulos, and Lee Gillam. 2010. *Cloud Computing - Principles, Systems and Applications*, Springer-Verlag London Limited.
- Slides here are modified from several sources in Universities and Internet.

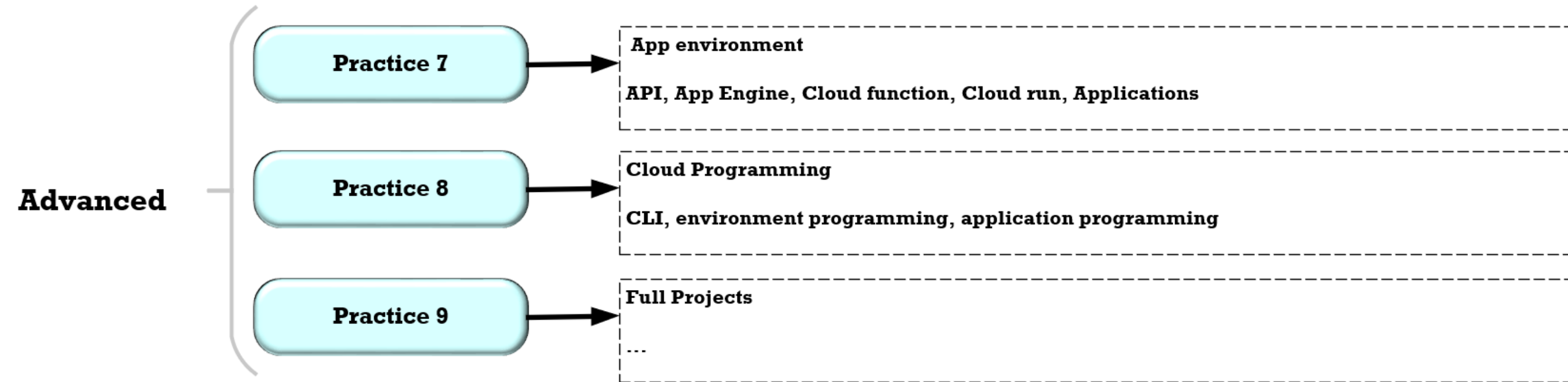
Cloud Computing: Practices

Levels: Beginning (3 weeks) – Intermediate (3 weeks) – **Advanced (3 weeks)**

Groups: 9 with 5 person/group

Practice: submit a report for each group, submit to our Google Classroom

Cloud Computing: Practices

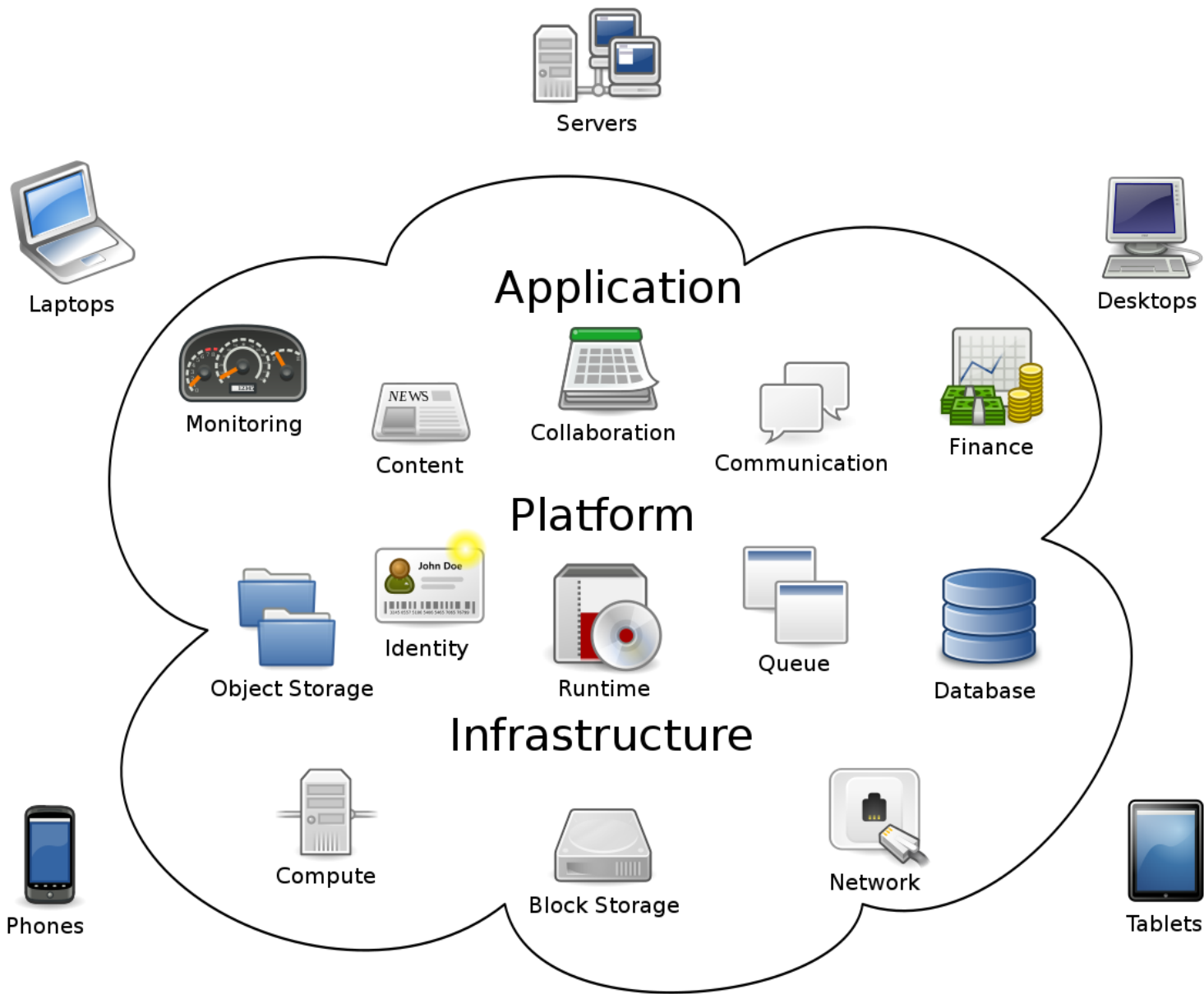
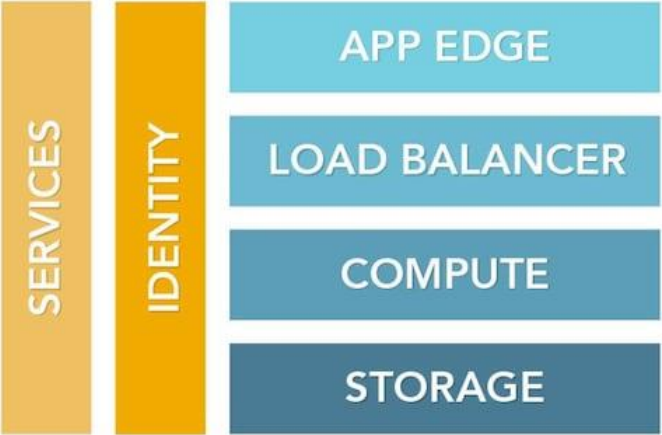


Content of Practice 7

1. What is App environment
2. App Engine
3. Cloud Run
4. Cloud Functions
5. Standard vs Flexible environment: Choose an App environment
6. Applications

Cloud App Environment

Why and What:



App Engine vs Cloud Run vs Cloud Functions

❑ Cloud app environment:

- Serverless computing lets the developer focus on what's most important – development – and not have to worry about the underlying details of infrastructure and maintenance.
- Other benefits such as autoscaling, pay-as-you-go and on-demand provisioning make this a no-brainer if you are building stateless HTTP applications, web and mobile apps, IoT and sensor apps, data processing systems, chatbots and more.
- There are three offerings in Google Cloud Platform that should be considered, so let's dive in. The three offerings we will be looking at are GCP App Engine, Google Cloud Run, and Cloud Function.



App Engine



Cloud Run



Google Cloud Functions

App Engine vs Cloud Run vs Cloud Functions

❑ GCP App Engine:

- GCP app engine is for building a serverless app with [a web or API backend](#) and supports several development languages without needing to worry about infrastructure support.
- When you have an application that needs to communicate with multiple services like a web app or API's, Google Cloud App Engine is a suitable solution.
- The relationship among those services means the application is treated as a managed entity.



App Engine vs Cloud Run vs Cloud Functions

❑ GCP App Engine:

- Build monolithic server-side rendered websites. App Engine supports popular development languages with a range of developer tools.

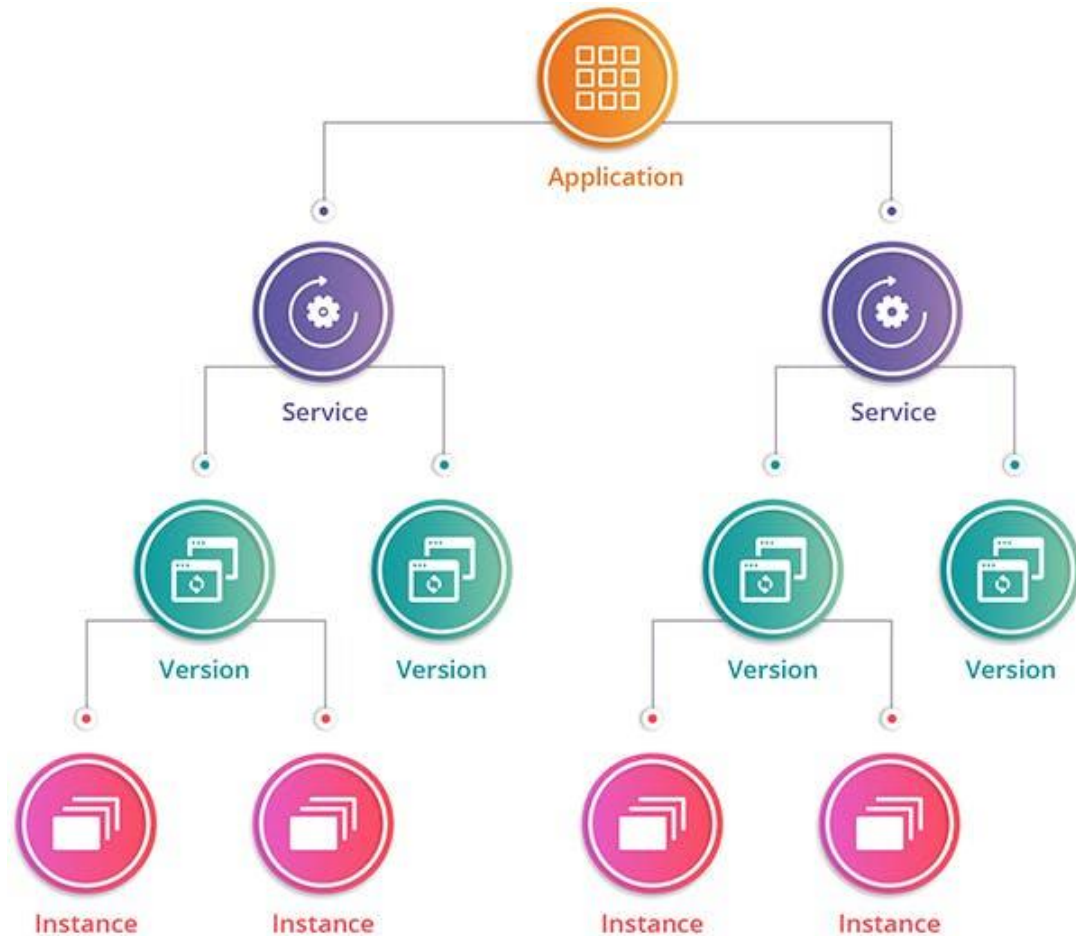
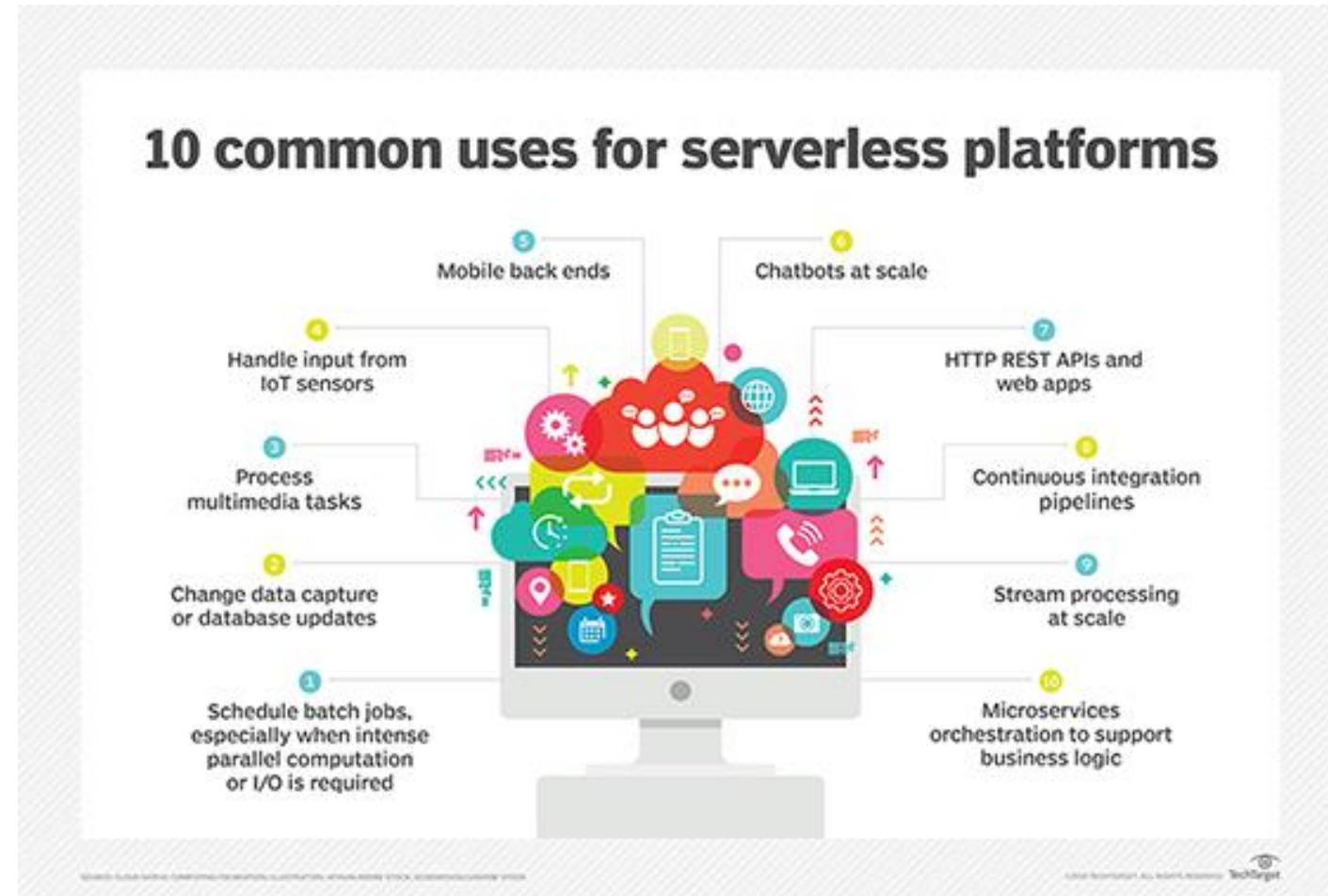


Image 2: Hierarchy Structure of App Engine

(Source: Cloud.google.com)





App Engine

#GCPsketchnote

@PVERGADIA THECLOUDGIRL.DEV 4.21.2021

What is App Engine?

FULLY MANAGED SERVERLESS COMPUTE

to build & deploy low latency, highly scalable applications



USE APP ENGINE WHEN....

- Web traffic requires minimal latency
- Web framework that supports routes
 - HTTP methods
 - Custom domains

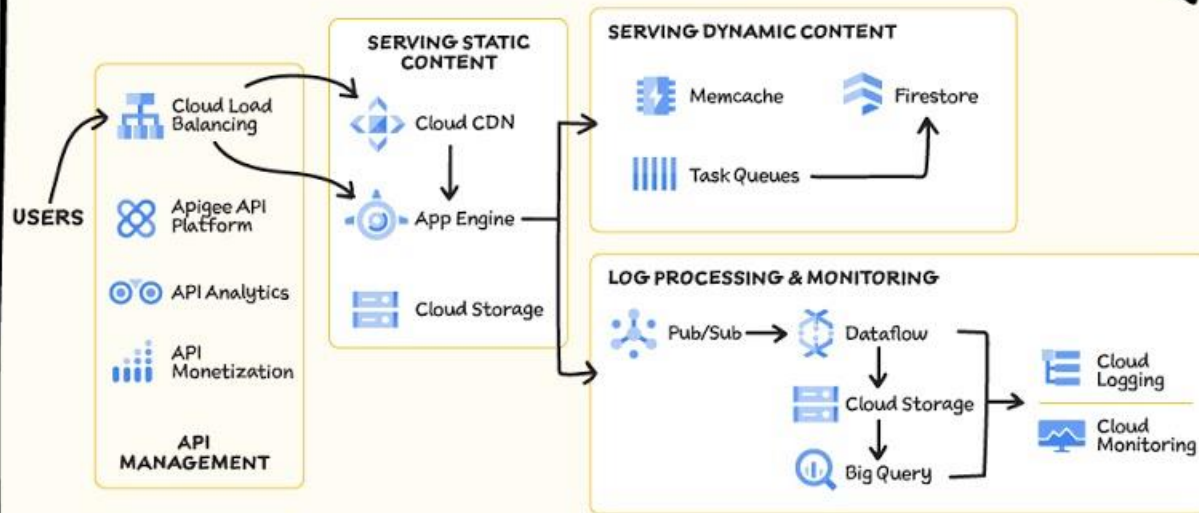
FEATURES

- Autoscaling from 0-many
- Application versioning
- Multiple language support
- Supports traffic splitting across versions for A/B testing
- Container support for custom runtimes
- Custom Domain & managed TLS certificate
- Static file serving

How does it Work?



Modern web applications & mobile back ends



* Choosing App Engine Environment



STANDARD

FLEX

Runtime environment for supported language

Application instances run in a sandbox

App with sudden & extreme traffic spikes

Scale from 0-many

Seconds

Depends on runtime

No

RUNTIME

TRAFFIC PATTERNS

SCALE TO ZERO

DEPLOYMENT TIME

MAX REQUEST TIMEOUT

WEBSOCKET SUPPORT

Custom runtimes any library/framework

Application instances run within Docker containers on Compute Engine

Apps with consistent traffic, regular fluctuations

Scale from 1 to many

Minutes

60 mins

Yes

App Engine vs Cloud Run vs Cloud Functions

❏ GCP App Engine:

- Key features
 - Popular programming languages
 - ✓ Build your application in Node.js, Java, Ruby, C#, Go, Python, or PHP.
- Fully managed
 - A fully managed environment lets you focus on code while App Engine manages infrastructure concerns.
- ✓ Free up your developers with zero server management and zero configuration deployments
- ✓ Stay agile with support for popular development languages and a range of developer tools
- ✓ Explore more products in our serverless portfolio

Popular languages	Build your application in Node.js, Java, Ruby, C#, Go, Python, or PHP.
Fully managed	A fully managed environment lets you focus on code while App Engine manages infrastructure concerns.
Powerful application diagnostics	Use Cloud Monitoring and Cloud Logging to monitor the health and performance of your app and Cloud Debugger and Error Reporting to diagnose and fix bugs quickly.
Application versioning	Easily host different versions of your app, and easily create development, test, staging, and production environments.
Application security	Help safeguard your application by defining access rules with App Engine firewall and leverage managed SSL/TLS certificates by default on your custom domain at no additional cost.
Services ecosystem	Tap a growing ecosystem of Google Cloud services from your app including an excellent suite of cloud developer tools.

App Engine vs Cloud Run vs Cloud Functions

❑ GCP App Engine: Pricing

- App Engine has competitive cloud pricing that scales with your app's usage. There are a few basic components you will see in the App Engine billing model such as standard environment instances, flexible environment instances, and App Engine APIs and services. To get an estimate of your bill, please refer to our pricing calculator.

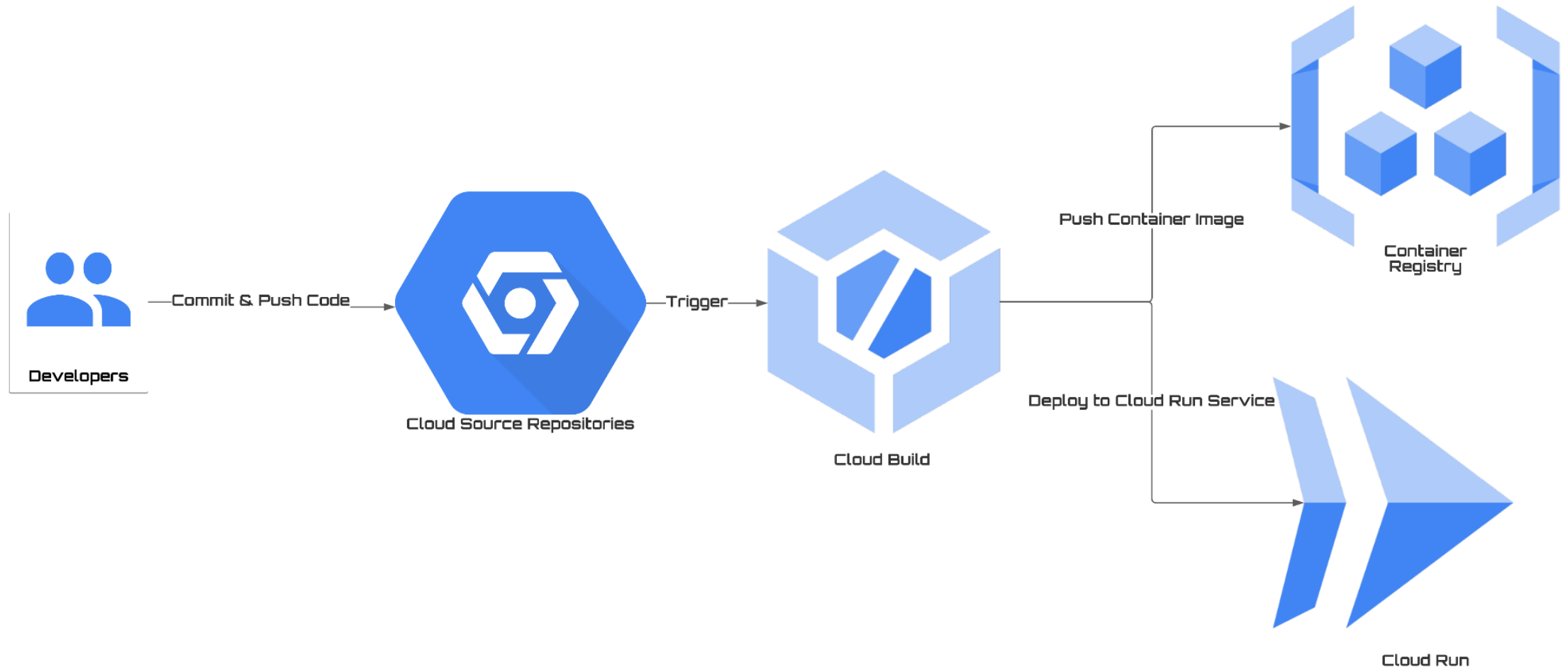
<https://cloud.google.com/appengine/pricing>

- Pricing is different for apps in the standard environment and the flexible environment. Apps in both environments may also incur charges from other Google Cloud products, depending on which products they use in addition to App Engine. Also, apps in both environments are subject to certain usage quotas and limits.

App Engine vs Cloud Run vs Cloud Functions

Google Cloud Run:

- Cloud Run is a serverless compute platform that is completely managed by Google, so you don't have to worry about backend application support.

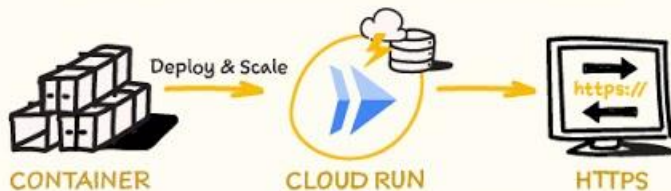




Cloud Run

What is Cloud Run??

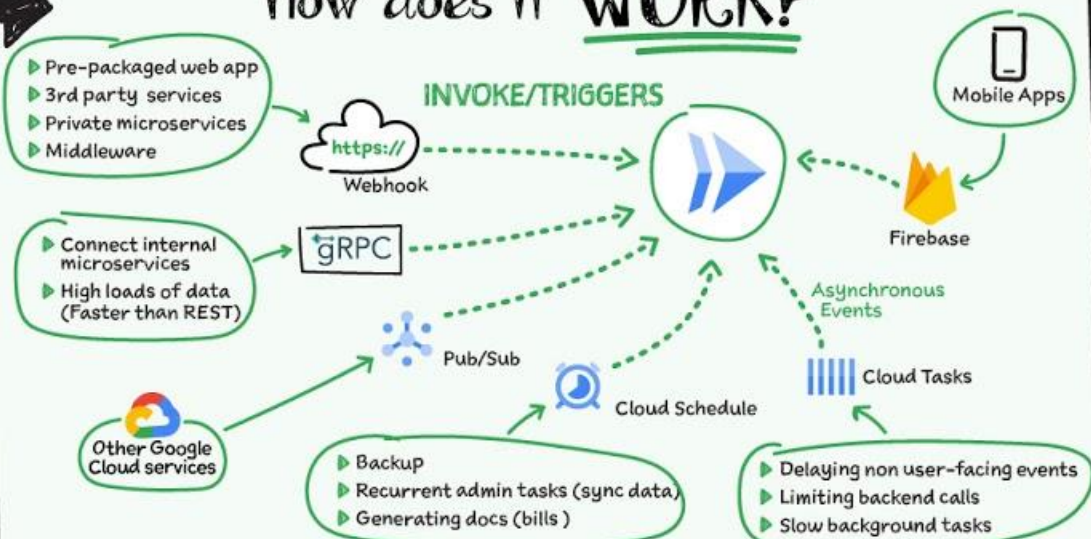
FULLY MANAGED COMPUTE ENVIRONMENT
TO DEPLOY & SCALE CONTAINERIZED APPS



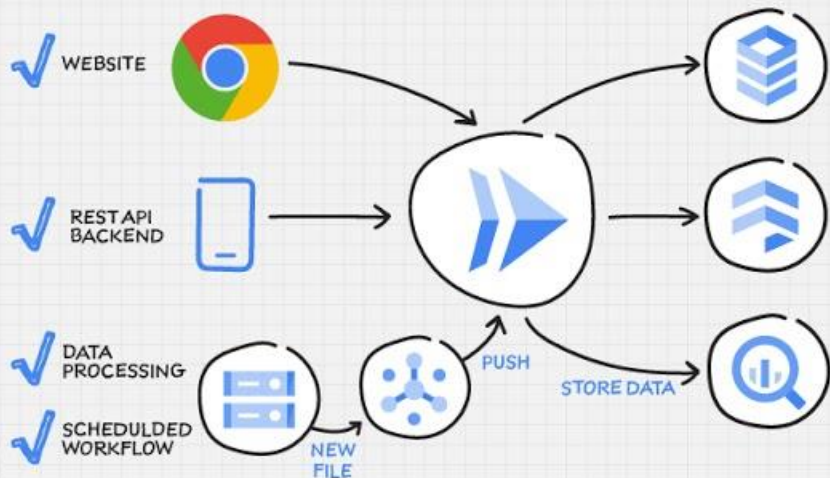
USE CLOUD RUN OVER CLOUD FUNCTIONS IF:

- ▶ Languages not supported in Cloud Functions
- ▶ Longer request timeout (upto 15 mins)
- ▶ Large volume
- ▶ Large concurrency

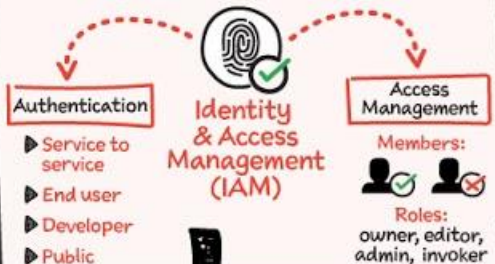
How does it WORK?



Cloud Run Use case (example)



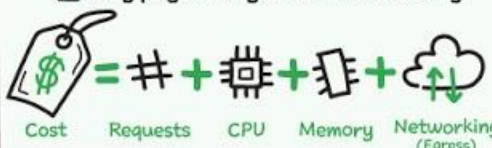
SECURITY for Cloud Run



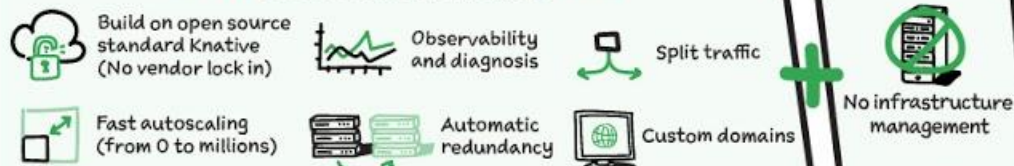
Cloud Run PRICING

PAY PER USE

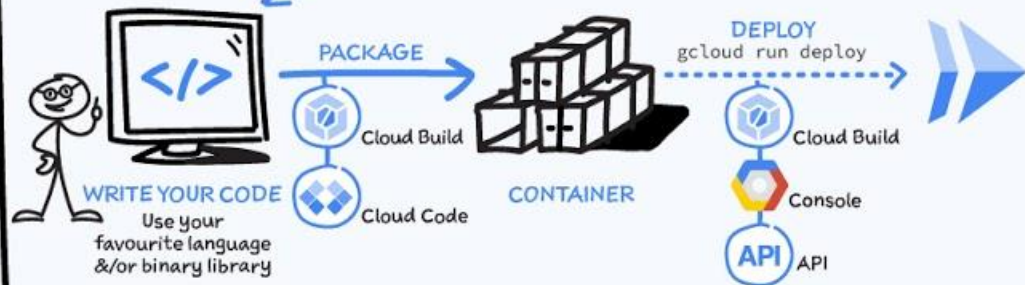
- ✓ Comes with always free tier
- ✓ Only pay when your code is running



CLOUD RUN FEATURES



HOW TO USE Cloud Run



App Engine vs Cloud Run vs Cloud Functions

❑ Google Cloud Run:

- Build and deploy scalable containerized apps written in any language (including Go, Python, Java, Node.js, .NET, and Ruby) on a fully managed platform.
- ✓ Deploy a sample container that responds to incoming web requests with this Quickstart.
- ✓ Building from source? Deploy a sample application to Cloud Run from source with this guide.
- ✓ Run database migrations, nightly reports, or batch data transformation with Cloud Run jobs

Key features

- Any language, any library, any binary
 - Use the programming language of your choice, any language or operating system libraries, or even bring your own binaries.
- Leverage container workflows and standards
 - Containers have become a standard to package and deploy code and its dependencies. Cloud Run pairs great with the container ecosystem: Cloud Build, Cloud Code, Artifact Registry, and Docker.
- Pay-per-use
 - Only pay when your code is running, billed to the nearest 100 milliseconds.

App Engine vs Cloud Run vs Cloud Functions

Google Cloud Run:

Any language, any library, any binary

Leverage container workflows and standards

Enhanced developer experience

Fully managed

Per-instance concurrency

Fast autoscaling

Redundancy

Security

Ephemeral and persistent storage

Integrated logging and monitoring

Process web traffic

Process asynchronous events

Portability

HTTPS URLs

Custom domains

HTTP/2, WebSockets, and gRPC

App Engine vs Cloud Run vs Cloud Functions

❑ Google Cloud Run: Pricing

- Pay-per-use, with an always-free tier, rounded up to the nearest 100 millisecond.
- Total cost is the sum of used CPU, Memory, Requests and Networking.
- Use the Google Cloud Pricing Calculator for an estimate.

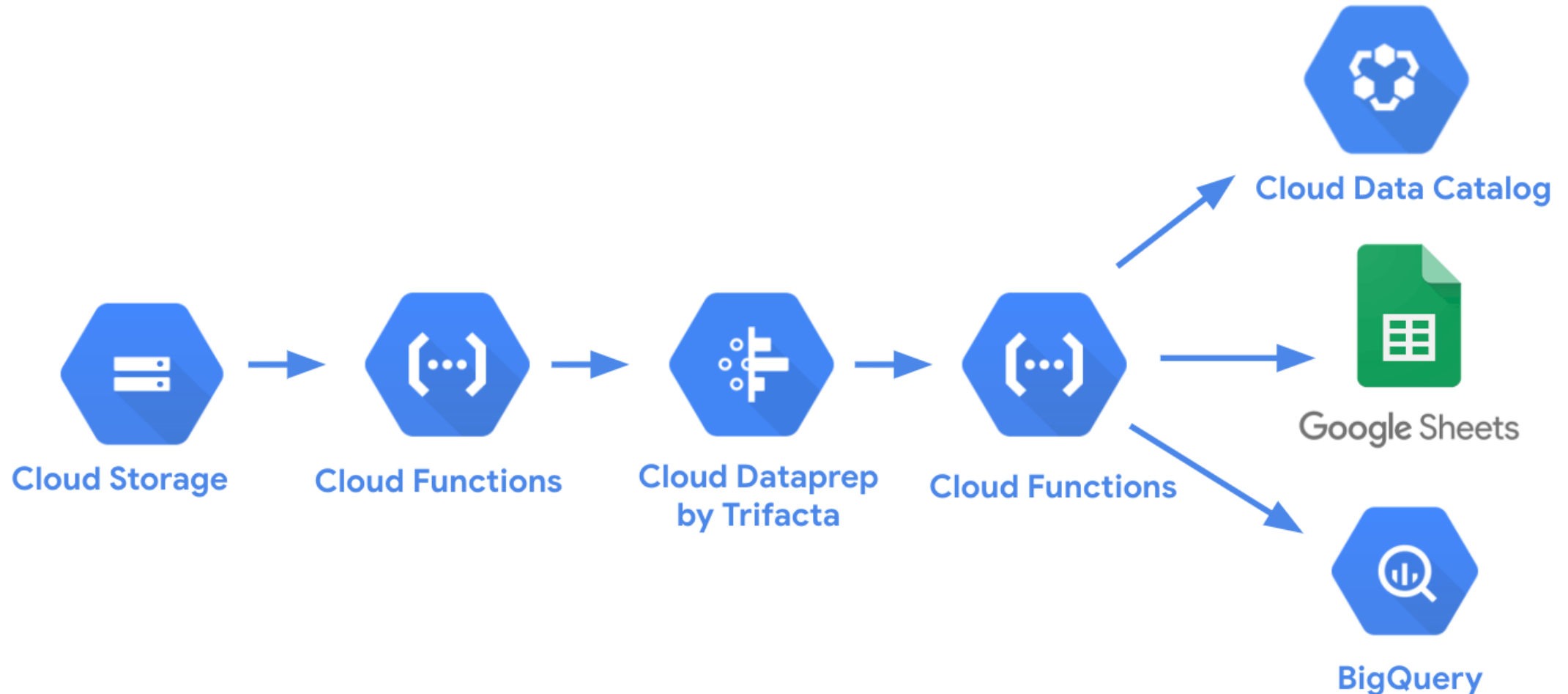
Cloud Run charges you only for the resources you use, rounded up to the nearest 100 millisecond. Note that each of these resources have a free tier. Your total Cloud Run bill will be the sum of the resources in the pricing table.

	CPU	MEMORY	REQUESTS
Price	\$0.00002400 per vCPU-second	\$0.00000250 per GiB-Second	\$0.40 per million requests
Always free	180,000 vCPU-seconds per month	360,000 GiB-seconds per month	2 million requests per month

App Engine vs Cloud Run vs Cloud Functions

Google Cloud Function:

- Event-driven solutions that extend to Google and 3rd party services are a good fit for cloud functions, as well as ones that need to scale quickly.





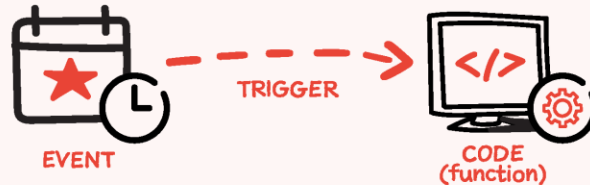
Cloud Functions

#GCPsketchnote

@PVERGADIA THECLOUDGIRL.DEV
8.17.2020

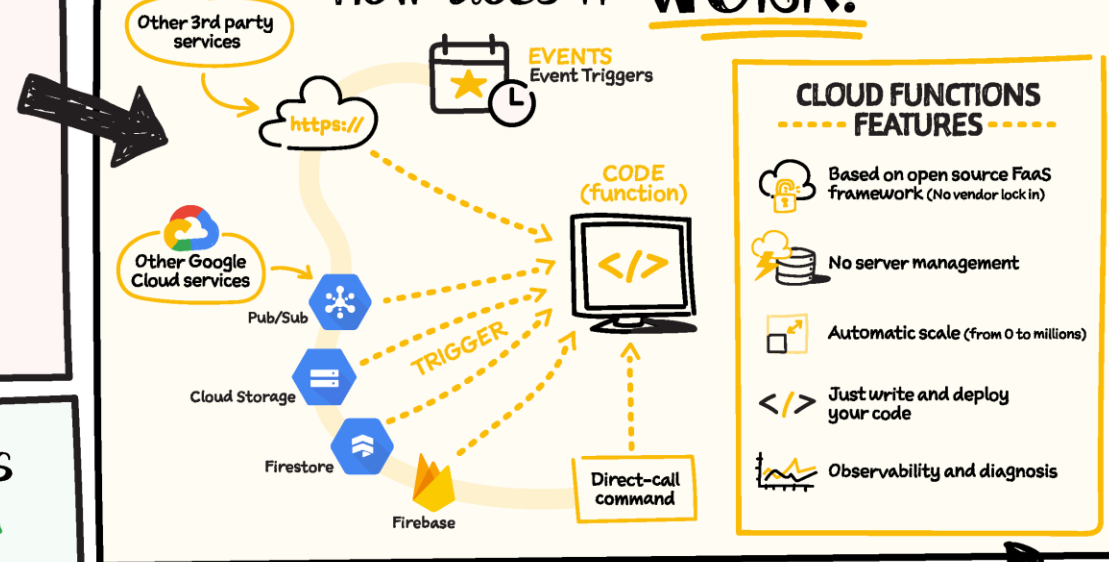
What is Cloud Functions?

EVENT-DRIVEN SERVERLESS
FUNCTION-AS-A-SERVICE (FAAS)



RUN CODE IN RESPONSE TO AN EVENT

How does it WORK?

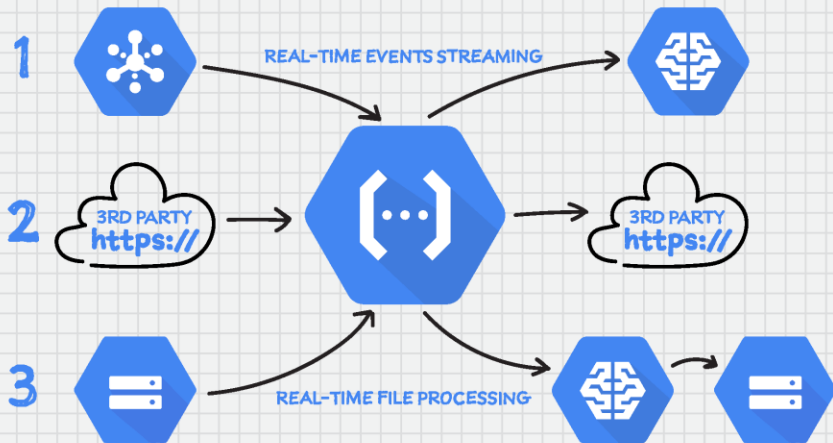


CLOUD FUNCTIONS FEATURES

- Based on open source FaaS framework (No vendor lock in)
- No server management
- Automatic scale (from 0 to millions)
- Just write and deploy your code
- Observability and diagnosis

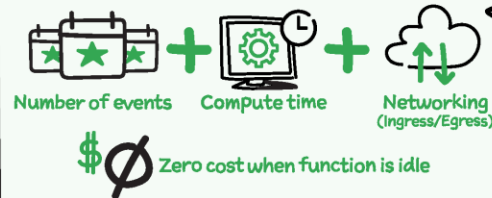
Cloud Functions Use case (example)

- Integration with third-party services and APIs
- IoT processing
- Real-time data processing - streaming
- Real-time file processing
- Serverless mobile back-ends
- Virtual assistance
- Real time ML



Cloud Functions PRICING

PAY AS YOU GO



SECURITY for Cloud Functions

NETWORK-BASED



VPC controls

IDENTITY-BASED

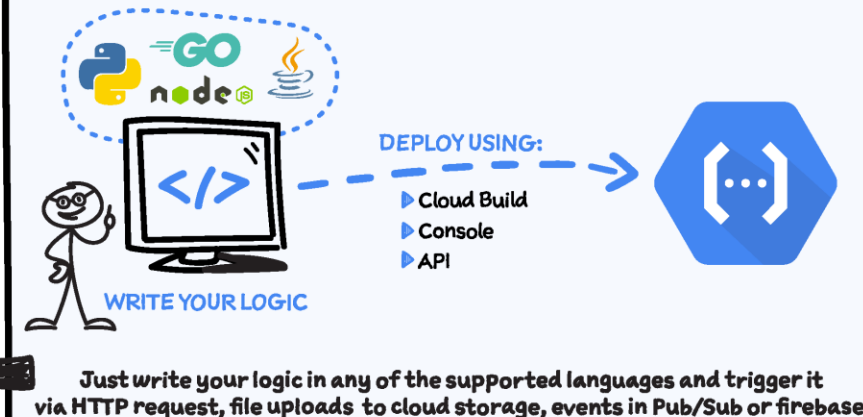


Identity & Access Management (IAM)

Function

Role

HOW TO USE Cloud Functions



App Engine vs Cloud Run vs Cloud Functions

❑ Google Cloud Function:

- Run your code in the cloud with no servers or containers to manage with our scalable, pay-as-you-go functions as a service (FaaS) product.
- ✓ Build and deploy your first Cloud Function using only your web browser with this Quickstart
- ✓ Serve users from zero to planet-scale without even thinking about any infrastructure.
- ✓ See how customers design applications using event-driven architectures with Cloud Functions

Simplified developer experience and increased developer velocity

Cloud Functions has a simple and intuitive developer experience. Just write your code and let Google Cloud handle the operational infrastructure. Develop faster by writing and running small code snippets that respond to events.

Pay only for what you use

You are only billed for your function's execution time, metered to the nearest 100 milliseconds. You pay nothing when your function is idle.

Avoid lock-in with open technology

Use open source FaaS (function as a service) framework to run functions across multiple environments and prevent lock-in. Supported environments include Cloud Functions, local development environment, on-premises, Cloud Run, and other serverless environments.

App Engine vs Cloud Run vs Cloud Functions

☐ Google Cloud Function:

• Connects and extends services to build complex applications	Cloud Functions lets you treat all Google and third-party cloud services as building blocks. Connect and extend them with code, and rapidly move from concept to production with end-to-end solutions and complex workflows. Further, integrate with third-party services that offer webhook integrations to quickly extend your application with powerful capabilities.
End-to-end development and diagnosability	Go from code to deploy, with integrated monitoring. Get full observability and diagnosability for your application with Cloud Trace and Cloud Debugger . Additionally, get support for local and disconnected development/debugging using open sourced functions framework.
Develop locally, scale globally	Serve users from zero to planet-scale without even thinking about any infrastructure. Cloud Functions automatically manages and scales underlying infrastructure with the size of workload.
No server management	Deploy your code and let Google run and scale it for you. Cloud Functions abstracts away all the underlying infrastructure, so that you can focus on your code and build applications faster than ever before.

Runs code in response to events

Cloud Functions allows you to trigger your code from Google Cloud, Firebase, and [Google Assistant](#), or call it directly from any web, mobile, or backend application via HTTP.

Pay only for what you use

You are only billed for your function's execution time, metered to the nearest 100 milliseconds. You pay nothing when your function is idle. Cloud Functions automatically spins up and backs down in response to events.

Avoid lock-in with open technology

Use open source FaaS (function as a service) framework to run functions across multiple environments and prevent lock-in. Supported environments include Cloud Functions, local development environment, on-premises, [Cloud Run](#), [Cloud Run for Anthos](#), and other Knative-based serverless environments.

App Engine vs Cloud Run vs Cloud Functions

❑ Google Cloud Function: Pricing

- Cloud Functions are priced according to how long your function runs, how many times it is invoked and how many resources you provision for the function. If your function makes an outbound network request, there are also additional data transfer fees.
- Cloud Functions uses Cloud Build to convert your function source code into an executable image.
- The images for Cloud Functions are stored in either Container Registry or Artifact Registry.
- Cloud Functions (2nd gen) exclusively uses Artifact Registry.
- If your function uses Container Registry, you will incur charges for storing your function's image, even when your use of Cloud Functions falls within the free tier.
- Artifact Registry has its own free tier of storage.
- If you deploy your functions to Artifact Registry and exceed the Artifact Registry free tier usage, you will incur charges for deploying your functions

App Engine vs Cloud Run vs Cloud Functions

❑ Comparison of App Environment:

▪ Deployment Type

The delivery method to deploy the stateless application

▪ Scaling

Identify if scaling is possible and if it is, what method of scaling is enabled. Cloud Run for Anthos scales differently from a standard app.

▪ Runtime environment support

Language support varies between services in GCP. Cloud Run and Cloud Run/Anthos flavors support virtually all common languages, but App Engine and Cloud Functions support more limited language support.

▪ Access controls

There are different approaches to handling security and identity management for stateless applications in GCP and they each handle things a little differently.

▪ HTTP/gRPC

Only Cloud Run with Anthos handles Cloud Endpoints management due to the mixed topology requirements

App Engine vs Cloud Run vs Cloud Functions

❑ Comparison of App Environment:

- Custom domains

Standalone and container-based applications support custom domain name mapping to your app, but the Cloud Functions platform does not.

- GPU/TPU support

Extended GPU capability using either graphics processing units (GPUs) or tensor processing units (TPUs) is only supported with Cloud Run for Anthos.

- Timeout (minutes)

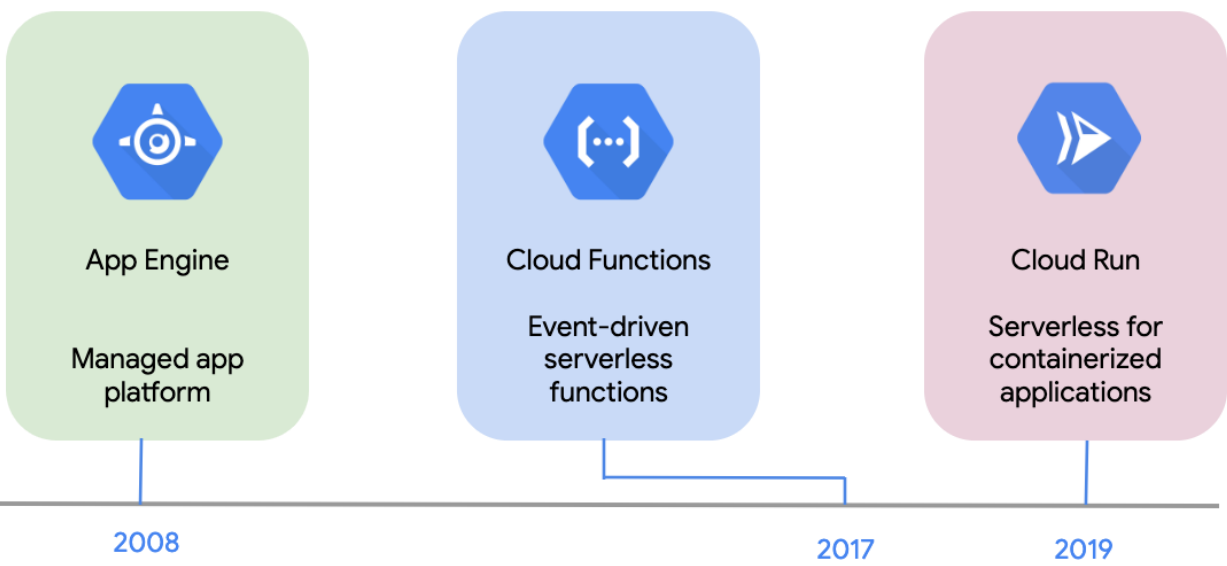
The number of minutes before timeout with no activity is different between stateless deployment types. Cloud Run flavors both offer 15-minute timeout windows; Cloud Functions times out at 9 minutes, and App Engine at 1 minute.

- VPC Networking

All stateless applications except for Cloud Run allow for VPC networking. As of this writing, GCP does not have VPC networking in general availability.

App Engine vs Cloud Run vs Cloud Functions

Comparison of App Environment:



Comparing Stateless Application Types			
	App Engine	Cloud Functions	Cloud Run
Deployment Type	App	Function	Container
Scaling	Yes	Yes	Yes
Runtime support	Java Node.js Python Go PHP	Node.js Python Go	All available
Access controls	Auth 2.0 CICP Firebase auth Google Sign-in Users API	IAM permissions	IAM permissions CICP Google Sign-in Firebase auth
HTTP/gRPC	No	No	No
Custom domains	Yes	No	Yes
GPU/TPU support	No	No	No
Timeout (mins)	1	9	15
VPC networking	Yes	Yes	No

App Engine vs Cloud Run vs Cloud Functions

❑ GCP App Engine vs Cloud Functions:

- While App Engine supports many different services within a single application, Cloud Functions support individualized services. It's an important detail when comparing Google App Engine vs Cloud Functions. If your requirements don't include multiple services then Cloud Functions is a great choice.
- For example, if you are saving or extracting data from a database, posting a file, or doing simple data validation, then using Cloud Functions is an appropriate choice.
- Functions operate in their own runtime environment and run independently; when a function is invoked it runs in a separate instance from other function calls.
- Unlike in GCP App Engine where data is shared among instances, once a function is invoked with Cloud Functions, it is on its own, so if you need to keep track of data when using Cloud Functions, you'll need to use a database or writable file in Cloud Storage. Currently, GCP supports Node.js, Python and Go..

App Engine vs Cloud Run vs Cloud Functions

❑ Cloud Run vs App Engine vs Cloud Functions:

- Unlike Google App Engine and Cloud Functions, Cloud Run can scale stateless containers and leverages Google Kubernetes Engine. This is a defining feature when comparing Cloud Run vs App Engine vs Cloud Functions
- If you need a serverless option that needs an application to run in a stateless container, Cloud Run may be the best choice for this kind of deployment. It is fully managed, and the pricing is based only on resources consumed. Also if you have a mixed architecture with on-premises and cloud resources, Anthos, the GKE (Google Kubernetes Engine) serverless container option is available for Cloud Run, so it's a great choice for these scenarios.

App Engine vs Cloud Run vs Cloud Functions

❑ **Standard vs Flexible environment: Choose an App environment**

- You can run your applications in App Engine by using the App Engine flexible environment or the App Engine standard environment. You can also choose to simultaneously use both environments for your application and allow your services to take advantage of each environment's individual benefits.

▪ **App Engine environments**

App Engine is well suited to applications that are designed using a microservice architecture, especially if you decide to utilize both environments. Use the following sections to learn and understand which environment best meets your application's needs.

App Engine vs Cloud Run vs Cloud Functions

❑ Standard vs Flexible environment: Choose an App environment

▪ When to choose the standard environment

Application instances run in a sandbox, using the runtime environment of a supported language listed below.

Applications that need to deal with rapid scaling.

The standard environment is optimal for applications with the following characteristics:

- Source code is written in specific versions of the supported many programming languages.
- Intended to run for free or at very low cost, where you pay only for what you need and when you need it. For example, your application can scale to 0 instances when there is no traffic.
- Experiences sudden and extreme spikes of traffic which require immediate scaling..

App Engine vs Cloud Run vs Cloud Functions

❑ Standard vs Flexible environment: Choose an App environment

▪ Standard environment

Standard environment is a mode where Google Cloud has pre-defined instance classes like B1, B2...F4_1G. This is more of a managed VM where the appengine has pre-defined vm configurations and we will be charged based on the instance running hours.

Advantages

- Zero downscaling is possible.
- Automatic security patches for managed VMs
- Very sophisticated development environment which mimics the actual live environment setup for deployment. Can deploy to PROD even from Eclipse :)
- Deployment happens within seconds
- VM Instance comes up in seconds in case of auto scaling

Disadvantages

- No direct way to consume compute engine resources from app-engine.
- No SSH access. Yes. read it again. No SSH access to VMs from standard environment. This means lesser flexibility in integration with third party systems
- No background process can be run.
- With projects involving multi threading and projects where performance is not a luxury, standard environment can be a killer.
- For each instance, there is an initial start up cost of 15 minutes instance time. Google will also charge you for 15mins after the last thread has completed the execution.
- Standard environment applications are single-homed, meaning that all instances of the application live in a single availability zone.

App Engine vs Cloud Run vs Cloud Functions

❑ Standard vs Flexible environment: Choose an App environment

▪ When to choose the flexible environment

Application instances run within Docker containers on Compute Engine virtual machines (VM).

Applications that receive consistent traffic, experience regular traffic fluctuations, or meet the parameters for scaling up and down gradually.

The flexible environment is optimal for applications with the following characteristics:

- Source code that is written in a version of any of the supported programming languages:
- Python, Java, Node.js, Go, Ruby, PHP, or .NET
- Runs in a Docker container that includes a custom runtime or source code written in other programming languages.
- Uses or depends on frameworks that include native code.
- Accesses the resources or services of your Google Cloud project that reside in the Compute Engine network.

App Engine vs Cloud Run vs Cloud Functions

❑ Standard vs Flexible environment: Choose an App environment

▪ Flexible environment

Flexible environment is a mode of usage with google app engine where the virtual machines are managed by the team using the environment. Runtime customization (Java8, Go, Ruby, etc. any runtime that u need), Infrastructure customization (Docker, SSH access, etc.) & Performance (guaranteed with choice of power packed VMs suiting your needs). Unlike standard environment, clientss will be charged based on the resource usage.

Advantages

- SSH access and support for docker . This is a key difference which opens a lot of possibilities.
- Access to compute engine resources as the VMs will be handled by the compute engines to efficiently execute client requests.
- Instances are health-checked, healed as necessary, and co-located with other services within the project.
- Critical, backwards compatible updates are automatically applied to the underlying operating system.
- Support for background threads and found to be very scalable when used even with multi threaded architecture

Disadvantages

- Instance startup time in minutes rather than seconds when compared to standard environment
- Deployment time in minutes rather than seconds when compared to standard environment
- NO ACCESS to following services : App Engine APIs & Services such as NDB, Users API, Memcache, Images API and others.

Cloud App Environment - Report

TASK I: Cloud App Environment

<https://cloud.google.com/appengine>

<https://cloud.google.com/functions>

<https://cloud.google.com/run>

Phân tích các đặc trưng (All features) của 3 loại App Environment:

- Google App Engine
- Google Cloud Functions
- Google Cloud Run

(lưu ý: bằng tiếng Việt!!!)

Cloud App Environment - Report

TASK II: Google App Environment Applications

<https://cloud.google.com/appengine>

<https://cloud.google.com/functions>

<https://cloud.google.com/run>

- ☺ Liệt kê và Giới thiệu một số ứng dụng thực tế của Google App Engine
- ☺ Liệt kê và Giới thiệu một số ứng dụng thực tế của Google Cloud Function
- ☺ Liệt kê và Giới thiệu một số ứng dụng thực tế của Google Cloud Run

Cloud App Environment - Report

TASK III: App Environment Pricing

How to find a good price for deploying App Environment

<https://cloud.google.com/appengine/pricing#app-engine-pricing>

App Engine standard environment pricing (Zone: Sydney (australia-southeast1))	
<i>Monthly cost</i>	
Instance class	Two B2 instance (20 hours/day), one B4 instance (12 hours/day)
Network resources	Outgoing network traffic (100 Gigabytes)
App Engine resources	5 Terabytes per month 500 Gigabytes Search API: 5000 Queries/day

Cloud App Environment - Report

TASK III: App Environment Pricing

How to find a good price for deploying App Environment

<https://cloud.google.com/appengine/pricing#app-engine-pricing>

App Engine flexible environment pricing (Zone: Taiwan (asia-east1))	
<i>Monthly cost</i>	
vCPU	4 core/20 hours/day
Memory	16 GB/20 hours/day
Persistent disk	Standard provisioned space: 2 TB
Outgoing network traffic	Network (Egress) Worldwide Destinations: 1 TB
Incoming network traffic	2 TB

Cloud App Environment - Report

TASK IV: Choose an App Engine environment

<https://cloud.google.com/appengine/docs/flexible/flexible-for-standard-users>

So sánh sự giống nhau và khác nhau (Similarities and key differences) giữa App Engine Standard and Flexible Environment → *bằng Tiếng Việt!*

- Application execution
- Accessing external services
- Local development
- Scaling characteristics
- Health checks
- Dropping requests when overloaded
- Instance sizes
- Maximum request timeout
- Traffic migration
- Cost comparisons