BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Backend as a Service (BaaS)

Chandan ravandur N

BaaS

- is a platform that
  - automates backend side development
  - takes care of the cloud infrastructure

- App teams
  - outsource the responsibilities of running and maintaining servers to a third party
  - focus on the frontend or client-side development

- Provides a set of tools to help developers to create a backend code speedily
- with help of ready to use features such as
  - scalable databases
  - APIs
  - cloud code functions
  - social media integrations
  - file storage
  - push notifications

Parse
Firebase
kinvey
backendless
Back4App

Apps suitable for BaaS

- Social media apps
  - alike Facebook, Instagram
- Real-time chat applications
  - alike WhatsApp
- Taxi apps
  - alike Uber, Ola
- Video and music streaming apps
  - similar to Netflix
- Mobile games
- Ecommerce apps

Why Backend as a service?

- A BaaS platform solves two problems:
  - Manage and scale cloud infrastructure
  - Speed up backend development

- Business reasons to use BaaS:
  - Reduce time to market
  - Save money and decrease the cost of development
  - Assign fewer backend developers to a project
  - Outsource cloud infrastructure management

Technical reasons to use BaaS:

- Focus on frontend development
- Excludes redundant stack setup
- No need to program boilerplate code
- Standardize the coding environment
- Let backend developers program high-value lines of code
- Provides ready to use features like authentication, data storage, and search

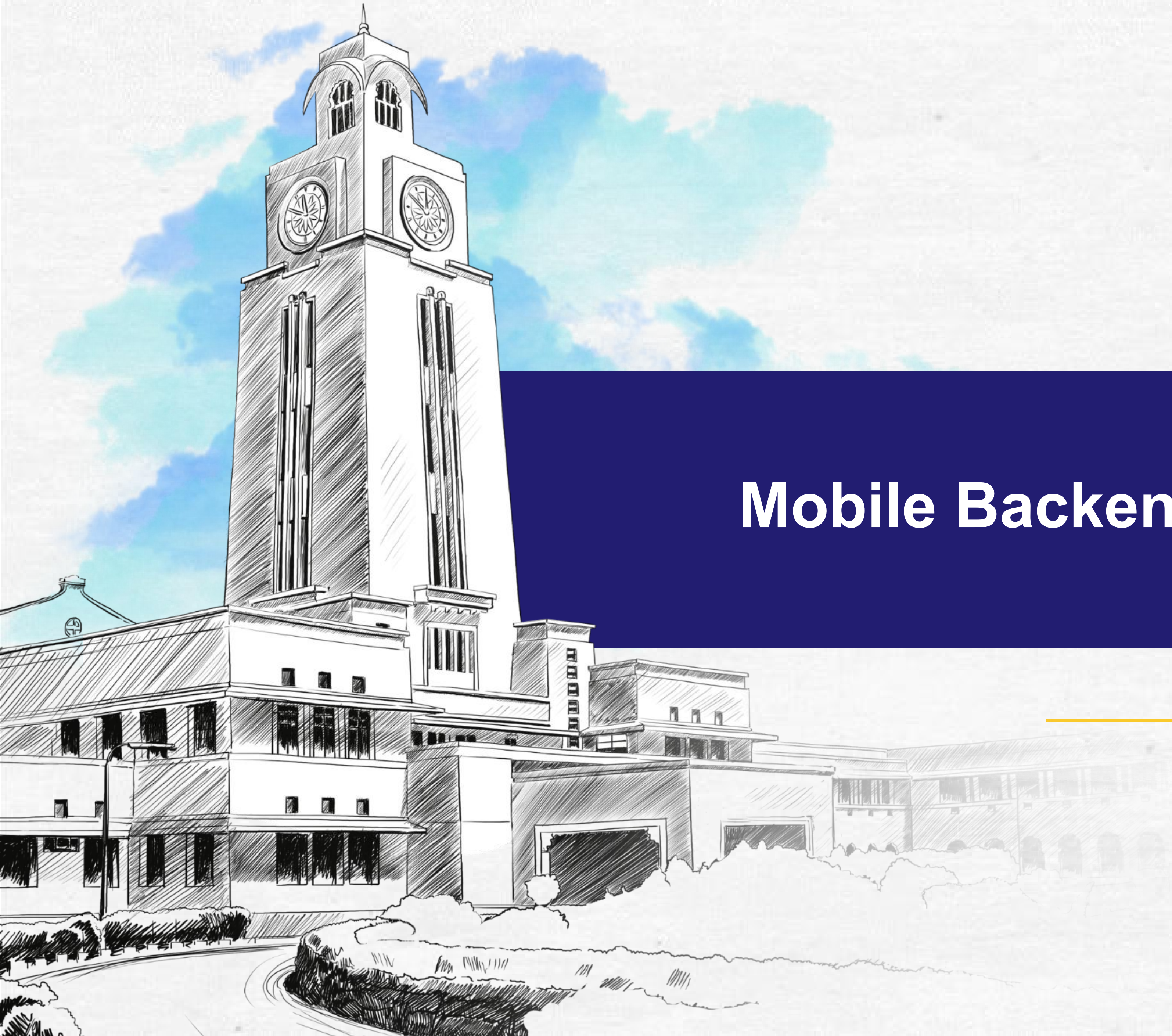
Pros-Cons

- Advantages
  - Speedy Development
  - Reduced Development price
  - Serverless, and no need to manage infrastructure

- Disadvantages
  - Less flexible as compared to custom coding / deployments
  - Less customization in comparison to a custom backend
  - Vendor lock-in possible

Thank You!

In our next session:



Mobile Backend as a Service MBaaS

Chandan Ravandu N

MBaaS

- Pretty much similar to BaaS!
  - BaaS can be used for web projects or mobile projects
  - Termed as Mobile backend as a service (MBaaS) when used for mobile development

- Allows you to use pre-developed backend stored in the cloud

- Backend includes unified functionality most applications use:
  - push notifications
  - social networks integrations
  - cloud storage
  - messaging
  - analytics etc.

MBaaS / BaaS Architecture

Three Tier

- The first layer - Database
  - Is the foundation and contains the database servers
  - A database cluster has at least two servers to replicate data and a backup routine to retrieve data
- The second layer - Application
  - Is the application cluster and contains multiple servers to process requests
  - Quantity of servers fluctuates throughout the time of the day
  - Auto-scaling procedures are necessary to fulfill the correct number of servers
- The third layer - Gateway
  - Connects the application servers to the Internet
  - Composed of load balancers and CDNs

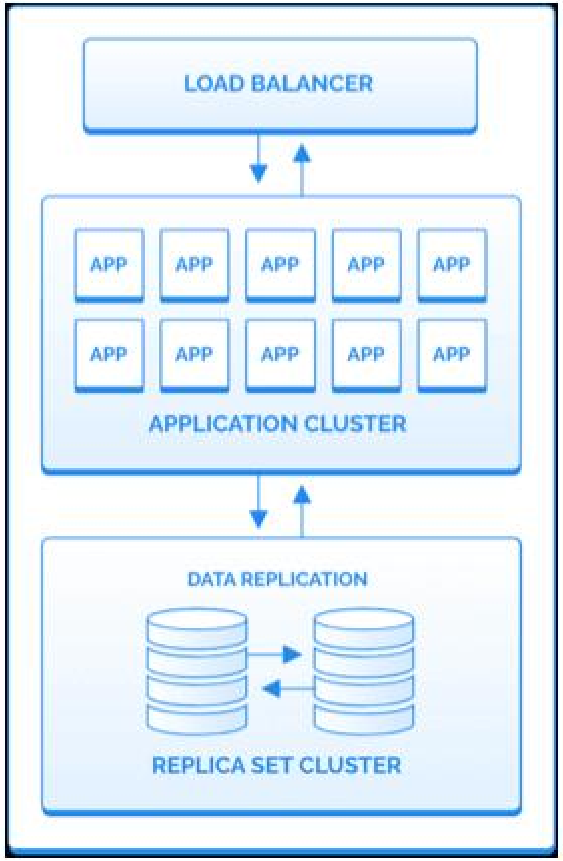


Image source : Back4App

MBaaS Providers

Back4App

- Offers a comprehensive product that uses several open-source technologies
  - NodeJS, Parse Server, and MongoDB
- Uses a simple approach
  - Help developers with resources needed to create scalable applications without the hassles of reinventing the wheel
- Features: The platform features include data model, push-notifications, REST and GraphQL APIs, login, authentication data backup, multiple SDK, non-technical administration panel, cloud functions, reset passwords, and CLI,
- Open Source
- Cloud-Hosting or self hosting possible
- Private Cloud and on-premises deployment possible
- Professional Services: Solutions Architect, Consulting, and enterprise plans

MBaaS Providers

Parse

- Excellent framework for expediting mobile application development
- Facebook converted the project to open source in 2016

- Features: JSON-like data management console, social-login (Facebook, Apple, Twitter, WeChat, GitHub, Google, etc.), password rest, integration with AWS file storage, push-notifications, SDKs (GraphQL, Rest, Javascript, iOS, Android, etc.)
- Open Source
- Cloud-Hosting Not possible, but Self-Hosting possible
- Remarks: This framework has over 16k stars and 4k forks on Github

MBaaS Providers

Firebase

- Versatile platform for mobile and web application development from Google
- Developers can create serverless apps and send push notifications to connected mobile devices
- Highly secure and efficient platform for scalability
- Unique feature of Firebase is its realtime database - makes it an excellent choice for developing apps with live chat

- Features: The main features include realtime database, phone authentication, storage, cloud functions, hosting, ML kit, and many other excellent features
- Cloud-Hosting possible
- Pay as you go pricing model
- Support Services: The platform supports case submission.
- Remarks: Firebase does neither offer private clouds or enterprise plans.

Thank You!

In our next session:

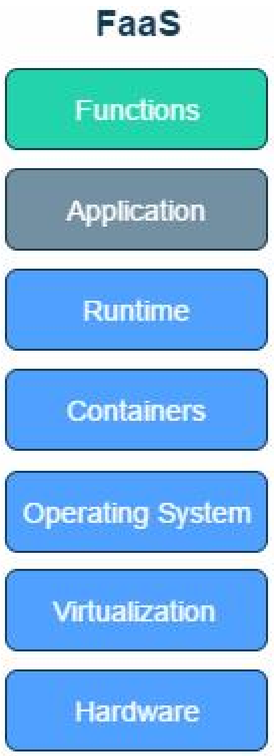
Function as a Service

Chandan Ravandur n

What is FaaS (Function-as-a-Service)?

- A type of cloud-computing service that
  - ✓ allows to execute code
  - ✓ in response to events
  - ✓ without the complex infrastructure
  - ✓ typically associated with building and launching microservices applications

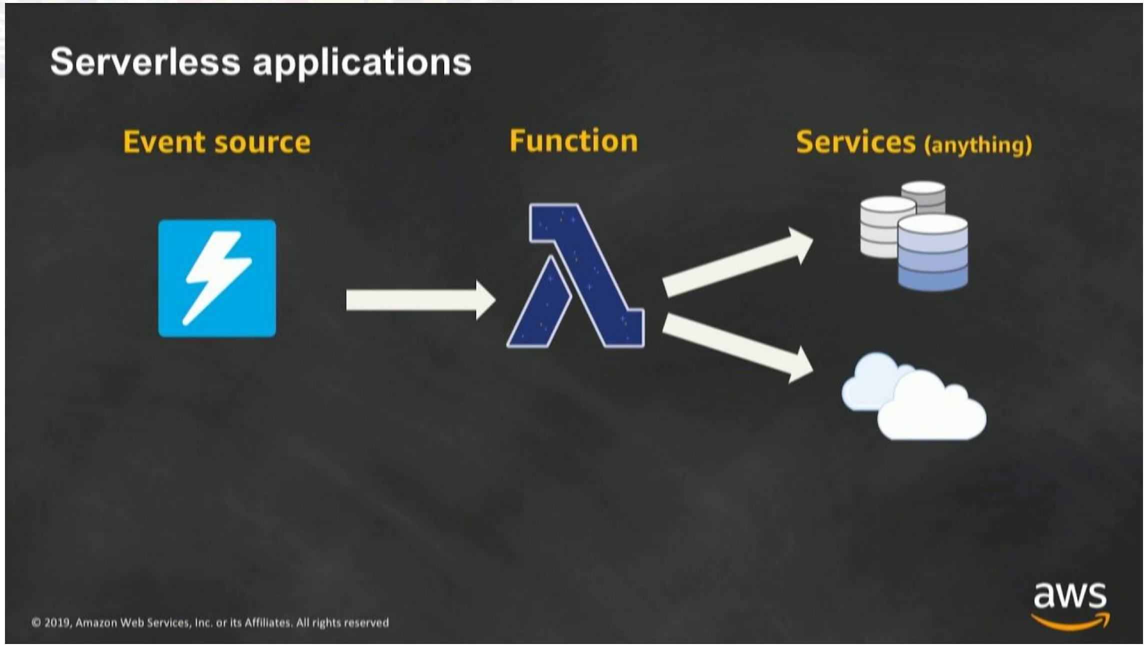
- Hosting a software application on the internet typically requires
  - ✓ provisioning and managing a virtual or physical server
  - ✓ managing an operating system and web server hosting processes



- With FaaS, the physical hardware, virtual machine operating system, and web server software management are all handled automatically by cloud service provider
  - ✓ allowing developers to focus solely on individual functions in application code

Characteristics of FaaS

- No server management or maintenance needed
- Stateless
- Automatic scaling, fine grained to number of requests
- Runs only when needed (triggered by events such as requests)



Benefits of FaaS

FaaS is a valuable tool if you're looking to efficiently and cost-effectively migrate applications to the cloud.

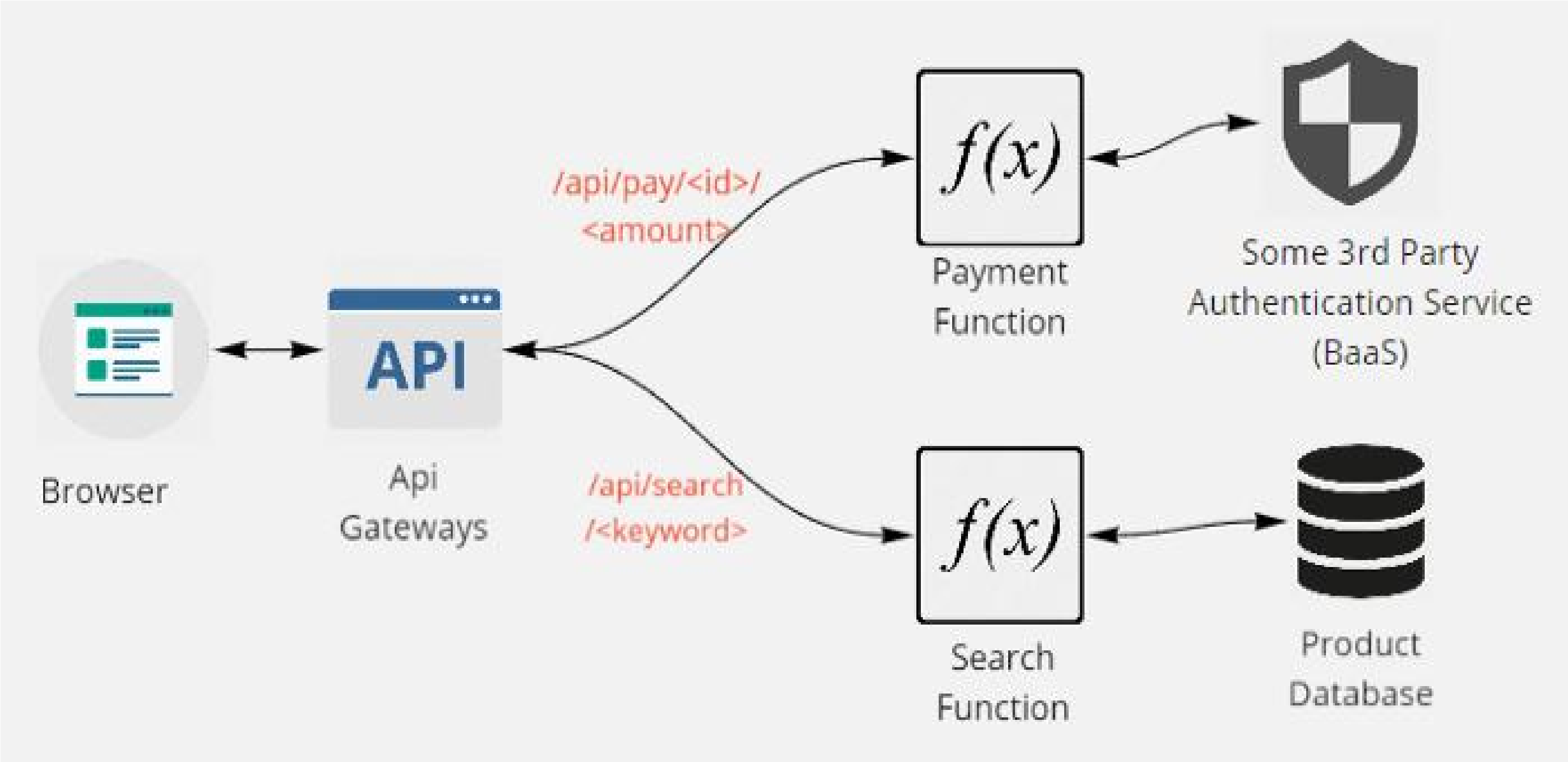
- Focus more on code, not infrastructure:
  - ✓ With FaaS, can divide the server into functions that can be scaled automatically and independently so don't have to manage infrastructure
  - ✓ This allows to focus on the app code and can dramatically reduce time-to-market

- Pay only for the resources you use, when you use them:
  - ✓ With FaaS, you pay only when an action occurs
  - ✓ When the action is done, everything stops—no code runs, no server idles, no costs are incurred
  - ✓ FaaS is, therefore, cost-effective, especially for dynamic workloads or scheduled tasks
  - ✓ FaaS also offers a superior total-cost-of-ownership for high-load scenarios

- Scale up or down automatically:
  - ✓ With FaaS, functions are scaled automatically, independently, and instantaneously, as needed
  - ✓ When demand drops, FaaS automatically scales back down

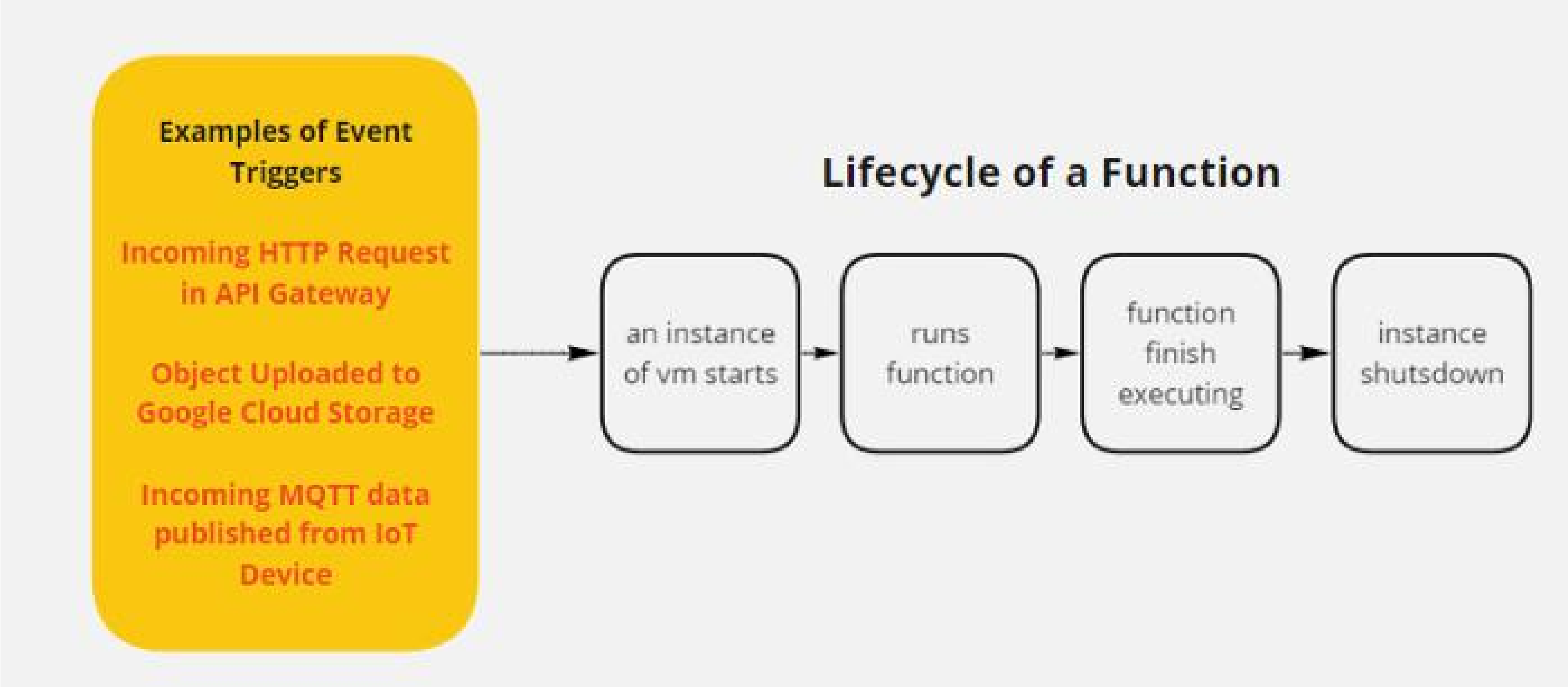
- Get all the benefits of robust cloud infrastructure:
  - ✓ FaaS offers inherent high availability because it
  - ✓ is spread across multiple availability zones per geographic region
  - ✓ can be deployed across any number of regions without incremental costs

Serverless Architecture



Source : medium

How it works?



Examples of Event Triggers
Incoming HTTP Request in API Gateway
Object Uploaded to Google Cloud Storage
Incoming MQTT data published from IoT Device
Lifecycle of a Function
an instance of vm starts
runs function
function finish executing
instance shutsdown

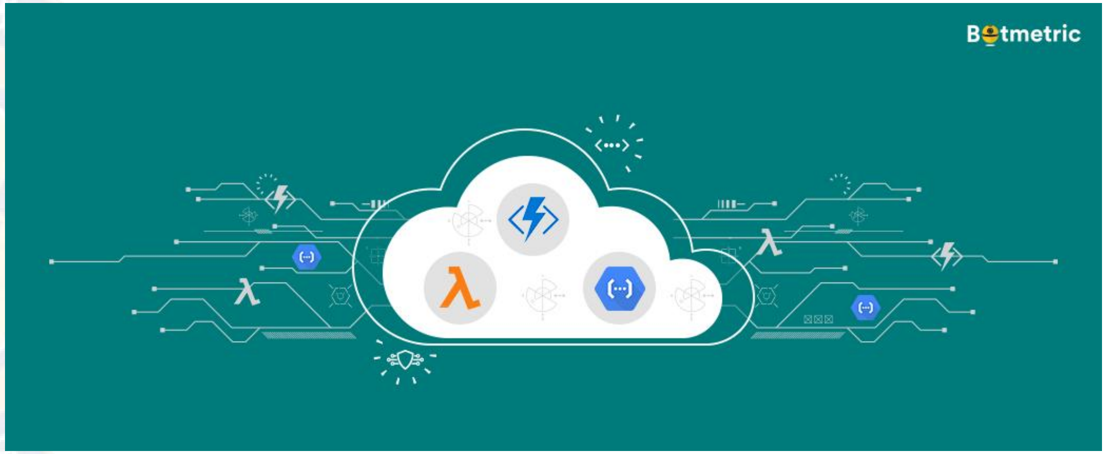
Source : medium

FaaS vs Serverless

- Serverless and Functions-as-a-Service (FaaS) are often conflated with one another
  - ✓ but the truth is that FaaS is actually a subset of serverless

- Serverless is focused on any service category
  - ✓ be it compute, storage, database, messaging, API gateways, etc.
  - ✓ where configuration, management, and billing of servers are invisible to the end user

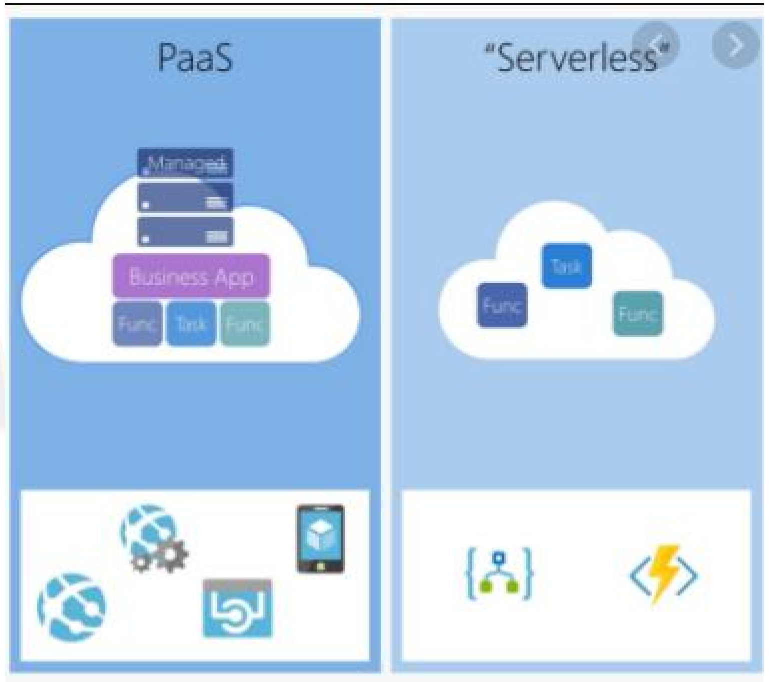
- FaaS, is focused on the event-driven computing paradigm
  - ✓ while perhaps the most central technology in serverless architectures
  - ✓ wherein application code, or containers, only run in response to events or requests



PaaS vs FaaS

- PaaS is very similar to FaaS
  - ✓ except that that it does not triggered by event
  - ✓ is typically always on
  - ✓ This does not suit the serverless framework

- Characteristics
  - ✓ No server management or maintenance needed
  - ✓ Stateless
  - ✓ Automatic scaling, adjustable to number of instances running
  - ✓ Typically runs 24/7



Source : stackify

- Sounds very similar to FaaS right?
  - ✓ However there is one key operational difference between the both of them:
  - ✓ Scalability, which affects operating cost.

Thank You!

In our next session:

Serverless on AWS

Chandan Ravandur N

Serverless on AWS

Build and run applications without thinking about servers

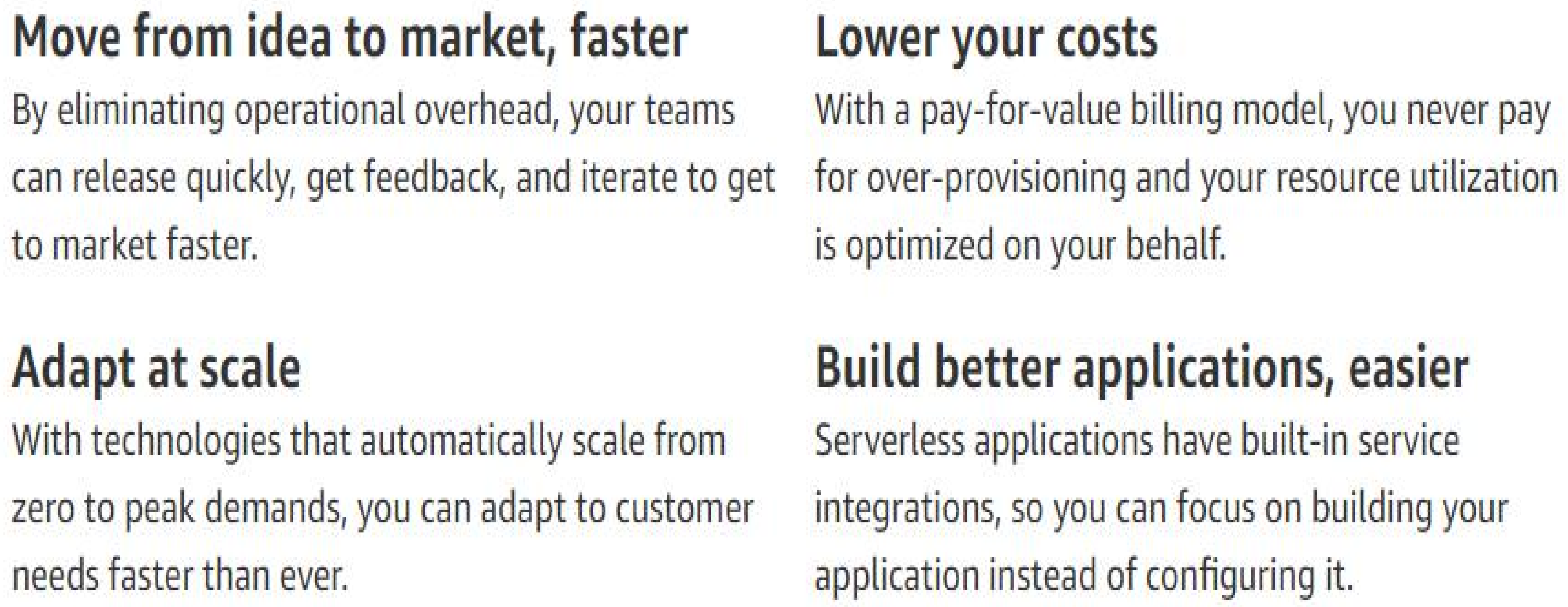
- Serverless is a way to describe the services, practices, and strategies that enables building more agile applications
  - ✓ to foster innovations and responses to the changes faster

- With serverless computing, infrastructure management tasks like capacity provisioning and patching are handled by AWS
  - ✓ developer can focus on only writing code that serves customers

- Serverless services like AWS Lambda come with
  - ✓ automatic scaling
  - ✓ built-in high availability
  - ✓ and a pay-for-value billing model

- Lambda is an event-driven compute service that enables to run code in response to events from over 150 natively-integrated AWS and SaaS sources
  - ✓ all without managing any servers

Benefits



Move from idea to market, faster

By eliminating operational overhead, your teams can release quickly, get feedback, and iterate to get to market faster.

Lower your costs

With a pay-for-value billing model, you never pay for over-provisioning and your resource utilization is optimized on your behalf.

Adapt at scale

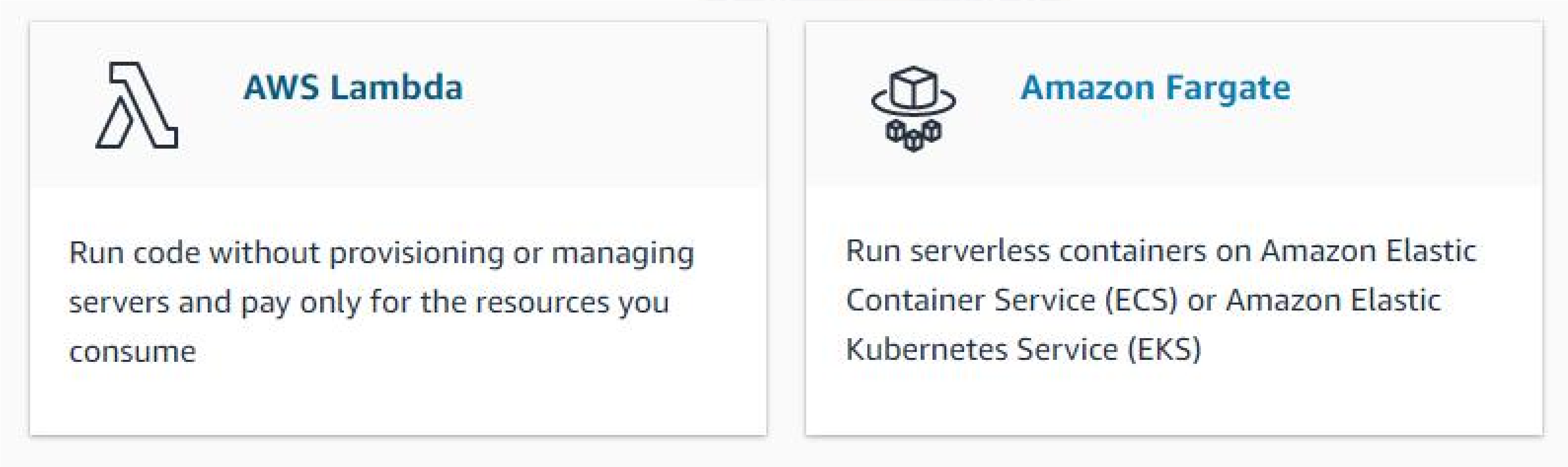
With technologies that automatically scale from zero to peak demands, you can adapt to customer needs faster than ever.

Build better applications, easier

Serverless applications have built-in service integrations, so you can focus on building your application instead of configuring it.

Serverless Services on AWS

Compute



AWS Lambda

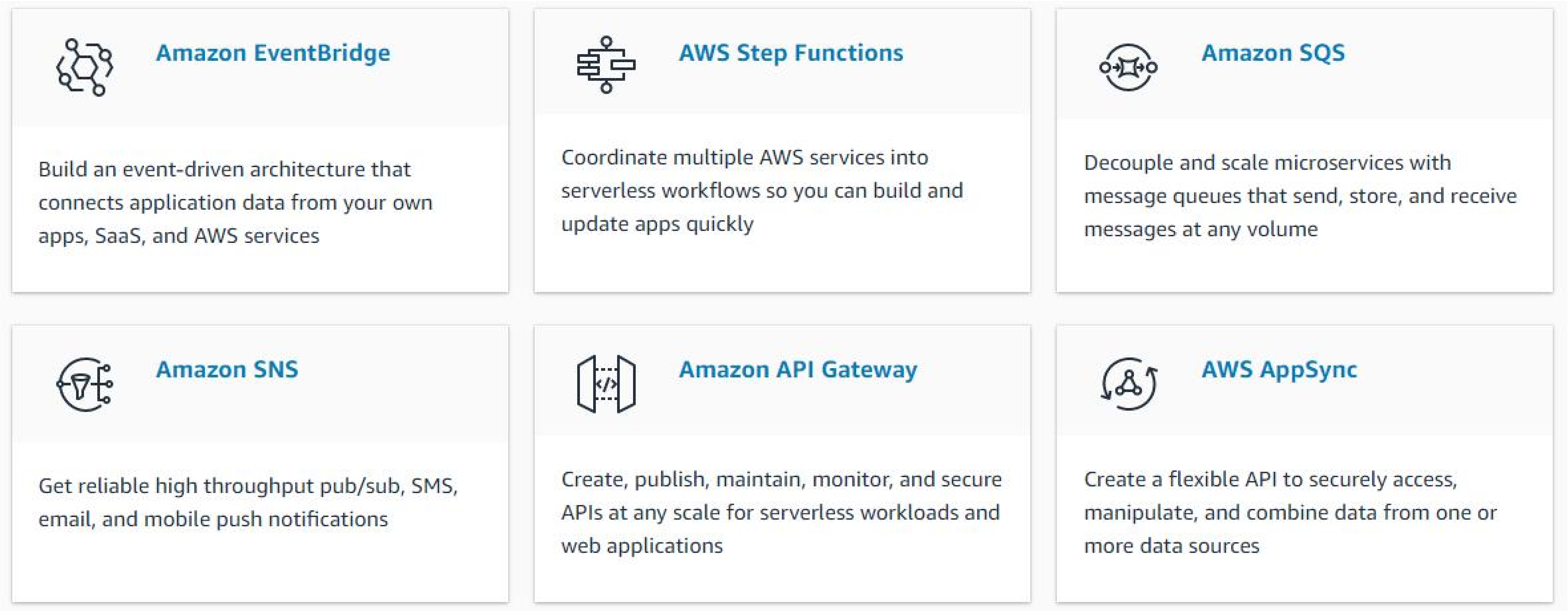
Run code without provisioning or managing servers and pay only for the resources you consume

Amazon Fargate

Run serverless containers on Amazon Elastic Container Service (ECS) or Amazon Elastic Kubernetes Service (EKS)

Serverless Services on AWS(2)

Application Integration



Amazon EventBridge

Build an event-driven architecture that connects application data from your own apps, SaaS, and AWS services

AWS Step Functions

Coordinate multiple AWS services into serverless workflows so you can build and update apps quickly

Amazon SQS

Decouple and scale microservices with message queues that send, store, and receive messages at any volume

Amazon SNS

Get reliable high throughput pub/sub, SMS, email, and mobile push notifications

Amazon API Gateway

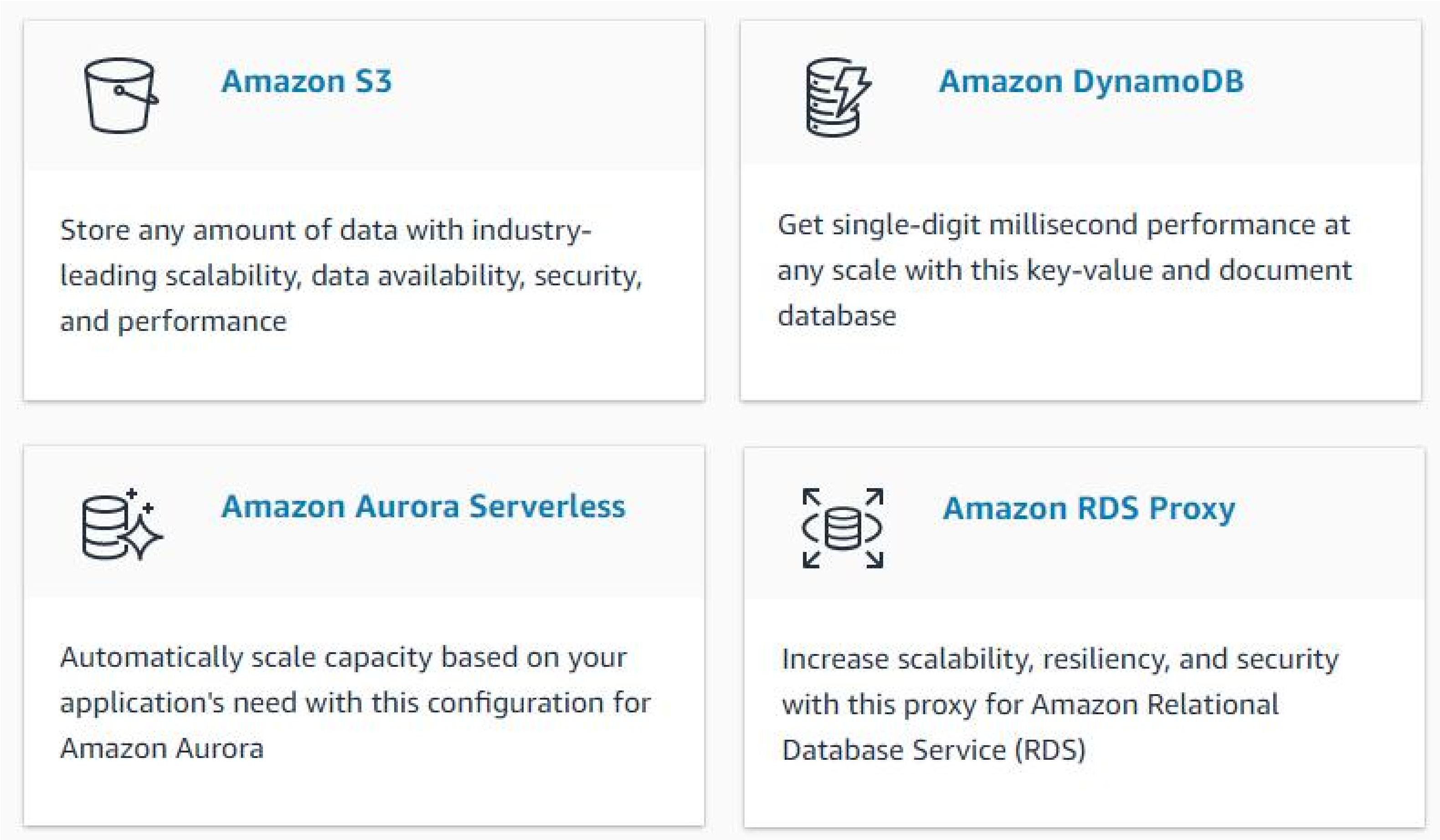
Create, publish, maintain, monitor, and secure APIs at any scale for serverless workloads and web applications

AWS AppSync

Create a flexible API to securely access, manipulate, and combine data from one or more data sources

Serverless Services on AWS(3)

Data Store



Amazon S3

Store any amount of data with industry-leading scalability, data availability, security, and performance

Amazon DynamoDB

Get single-digit millisecond performance at any scale with this key-value and document database

Amazon Aurora Serverless

Automatically scale capacity based on your application's need with this configuration for Amazon Aurora

Amazon RDS Proxy

Increase scalability, resiliency, and security with this proxy for Amazon Relational Database Service (RDS)

Reference :

AWS Serverless

Thank You!

In our next session:

AWS Lambda

Chandan Ravandur n

AWS Lambda

Run code without thinking about servers or clusters. Only pay for what you use.

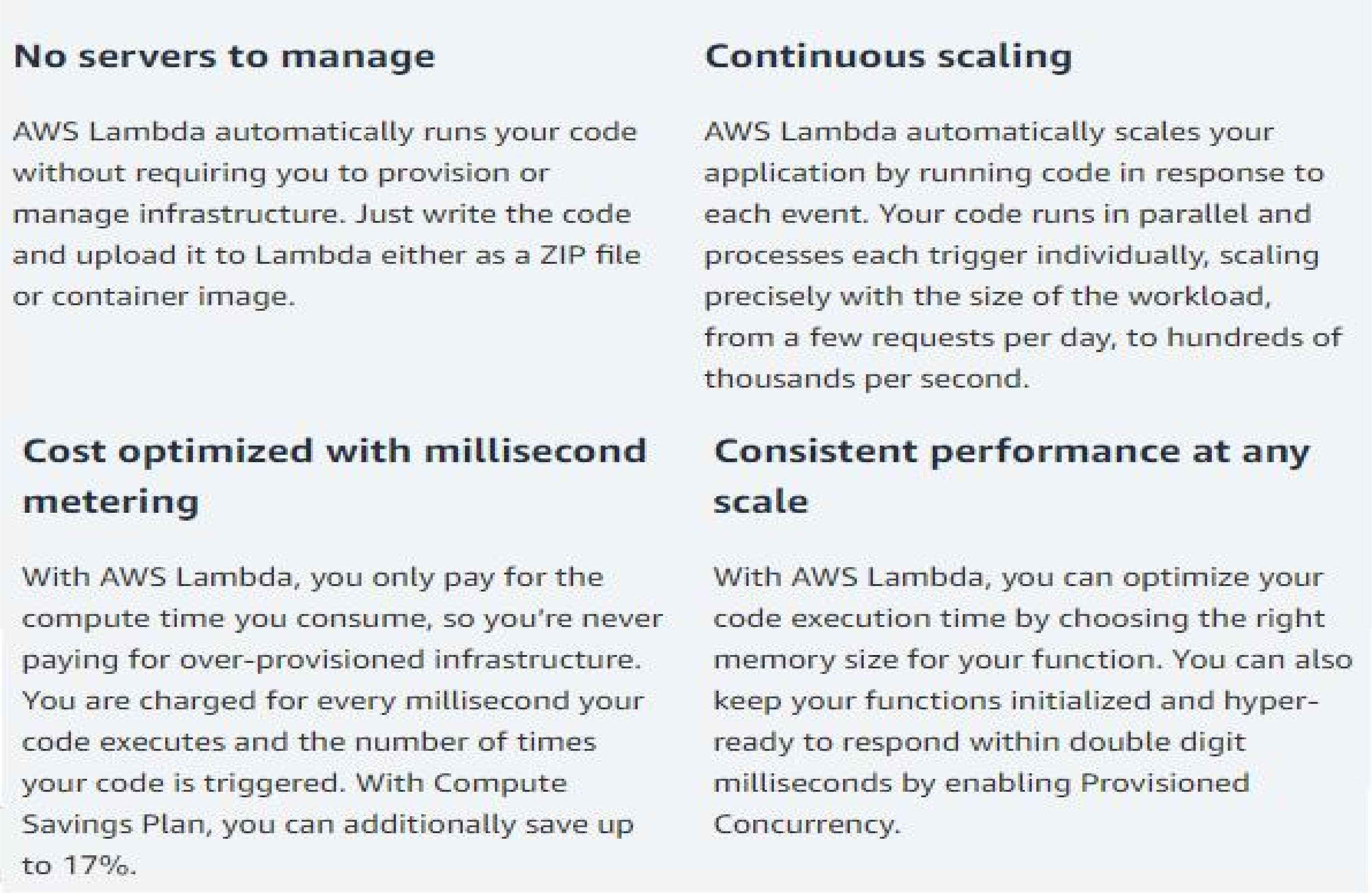
- AWS Lambda is a serverless compute service that allows to run code
  - ✓ without provisioning or managing servers
  - ✓ creating workload-aware cluster scaling logic
  - ✓ maintaining event integrations
  - ✓ or managing runtimes

- With Lambda, one can run code for virtually any type of application or backend service
  - ✓ all with zero administration

- Just upload code as a ZIP file or container image, and Lambda
  - ✓ automatically and precisely allocates compute execution power
  - ✓ runs code based on the incoming request or event, for any scale of traffic

- Can write Lambda functions in favorite language (Node.js, Python, Go, Java, and more)
  - ✓ use both serverless and container tools, such as AWS SAM or Docker CLI, to build, test, and deploy functions

Benefits



No servers to manage

AWS Lambda automatically runs your code without requiring you to provision or manage infrastructure. Just write the code and upload it to Lambda either as a ZIP file or container image.

Continuous scaling

AWS Lambda automatically scales your application by running code in response to each event. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload, from a few requests per day, to hundreds of thousands per second.

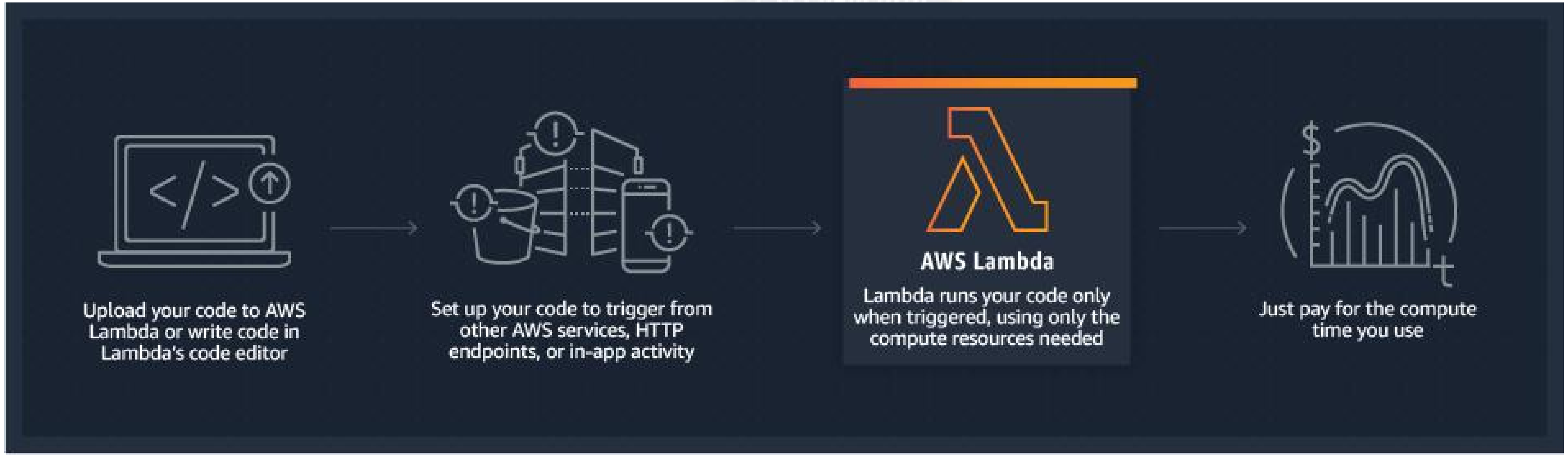
Cost optimized with millisecond metering

With AWS Lambda, you only pay for the compute time you consume, so you're never paying for over-provisioned infrastructure. You are charged for every millisecond your code executes and the number of times your code is triggered. With Compute Savings Plan, you can additionally save up to 17%.

Consistent performance at any scale

With AWS Lambda, you can optimize your code execution time by choosing the right memory size for your function. You can also keep your functions initialized and hyper-ready to respond within double digit milliseconds by enabling Provisioned Concurrency.

How it works?



Upload your code to AWS Lambda or write code in Lambda's code editor

Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

AWS Lambda

Lambda runs your code only when triggered, using only the compute resources needed

Just pay for the compute time you use

Use cases

What can you build with AWS Lambda?

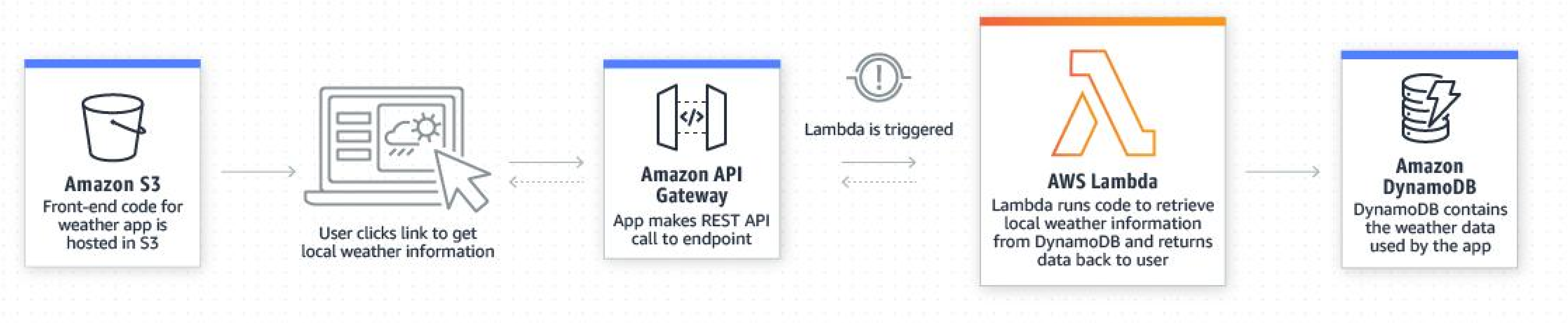
- Backends
  - ✓ Can build serverless backends using AWS Lambda to handle
    - ❖ web
    - ❖ mobile
    - ❖ Internet of Things (IoT)
    - ❖ and 3rd party API requests
  - ✓ Can take advantage of Lambda's consistent performance controls, such as multiple memory configurations and Provisioned Concurrency
    - ❖ for building latency-sensitive applications at any scale

- Data processing
  - ✓ Can use AWS Lambda to execute code in response to triggers such as
    - ❖ changes in data
    - ❖ shifts in system state
    - ❖ or actions by users
  - ✓ Lambda
    - ❖ can be directly triggered by AWS services such as S3, DynamoDB, Kinesis, SNS, and CloudWatch
    - ❖ can connect to existing EFS file systems, or it can be orchestrated into workflows by AWS Step Functions
    - ❖ allows to build a variety of real-time serverless data processing systems

Use case

Backends - Web applications

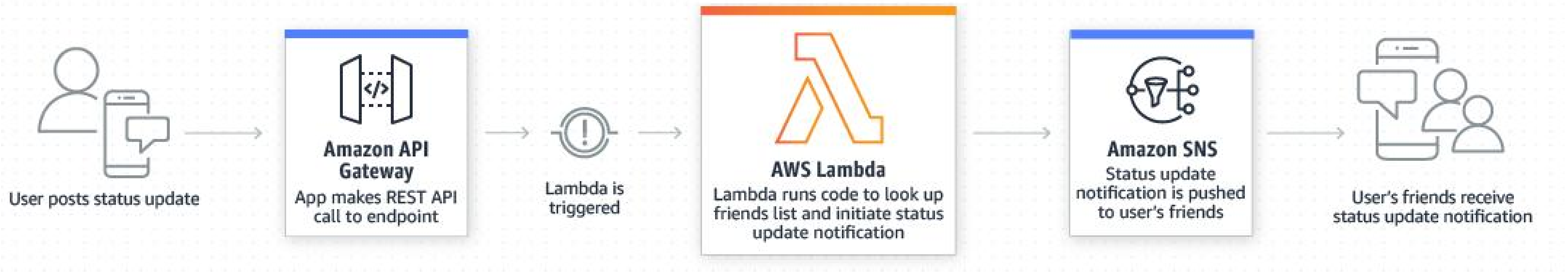
- By combining AWS Lambda with other AWS services, developers can build powerful web applications
  - ✓ that automatically scale up and down and run in a highly available configuration across multiple data centers
  - ✓ with zero administrative effort required for scalability, back-ups or multi-data center redundancy


Amazon S3
Front-end code for weather app is hosted in S3
User clicks link to get local weather information
Amazon API Gateway
App makes REST API call to endpoint
Lambda is triggered
AWS Lambda
Lambda runs code to retrieve local weather information from DynamoDB and returns data back to user
Amazon DynamoDB
DynamoDB contains the weather data used by the app

Use case (2)

Backends - Mobile backends

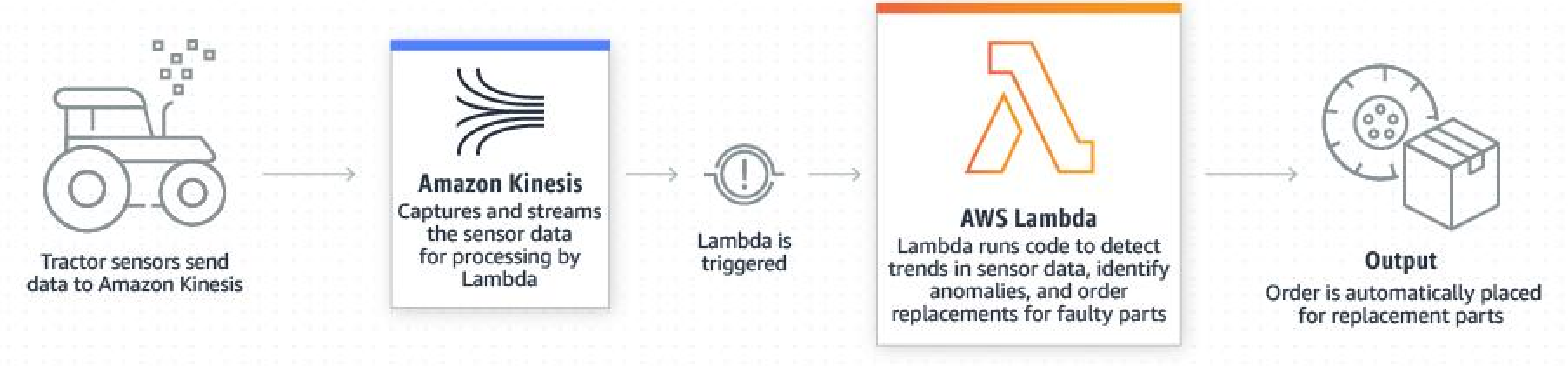
- AWS Lambda makes it easy to create rich, personalized app experiences
- Can build backends using AWS Lambda and Amazon API Gateway to authenticate and process API requests
- Can use AWS Amplify to easily integrate backend with your iOS, Android, Web, and React Native frontends


User posts status update
Amazon API Gateway
App makes REST API call to endpoint
Lambda is triggered
AWS Lambda
Lambda runs code to look up friends list and initiate status update notification
Amazon SNS
Status update notification is pushed to user's friends
User's friends receive status update notification

Use case (3)

Backends - IoT backends

- Can build serverless backends using AWS Lambda to handle
  - ✓ web
  - ✓ mobile
  - ✓ Internet of Things (IoT)
  - ✓ and 3rd party API requests



Tractor sensors send data to Amazon Kinesis

Amazon Kinesis
Captures and streams the sensor data for processing by Lambda

Lambda is triggered

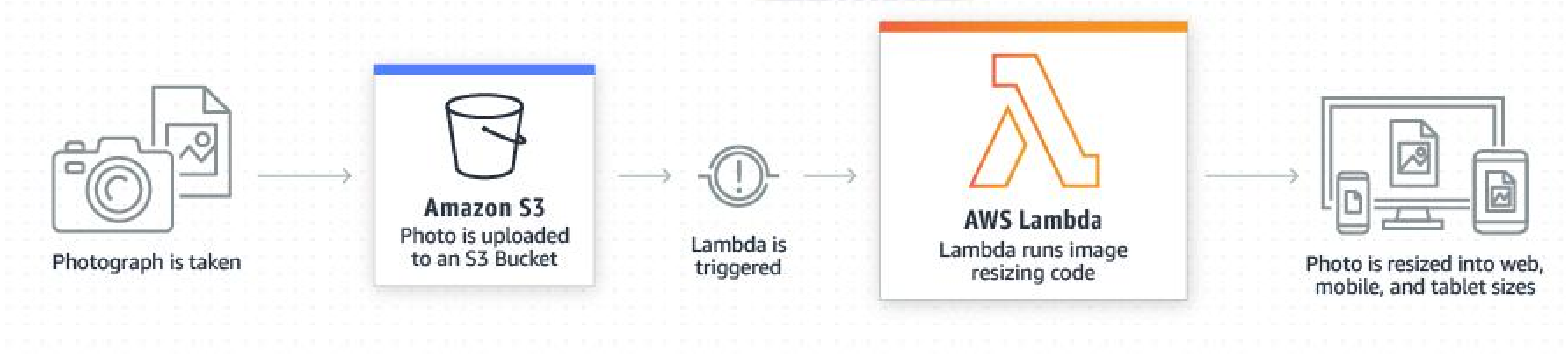
AWS Lambda
Lambda runs code to detect trends in sensor data, identify anomalies, and order replacements for faulty parts

Output
Order is automatically placed for replacement parts

Use case (4)

Data processing - Real-time file processing

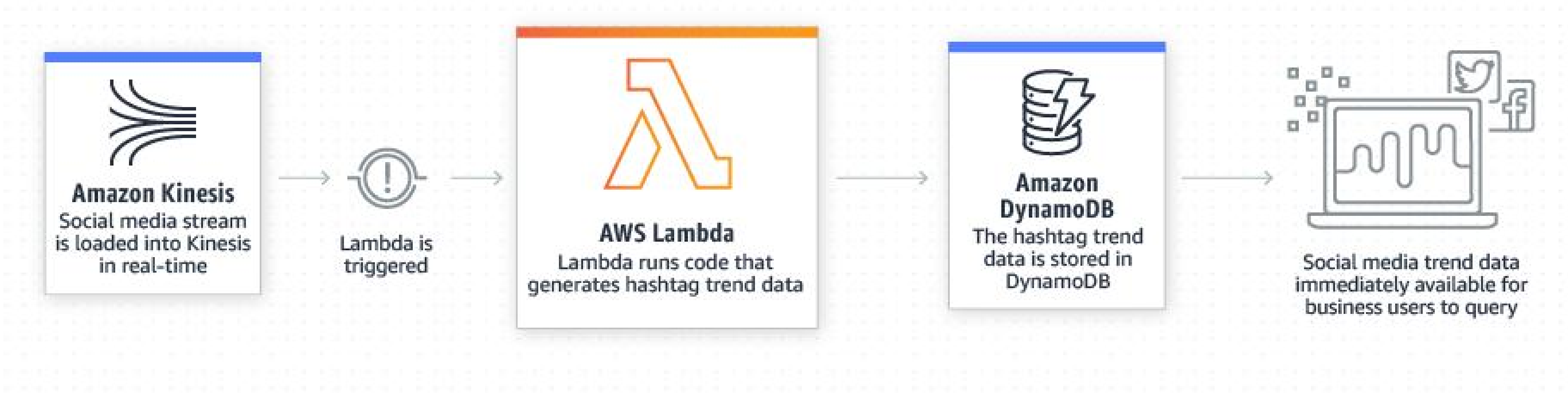
- Can use Amazon S3 to trigger AWS Lambda to process data immediately after an upload
- Can also connect to an existing Amazon EFS file system directly
- enabling massively parallel shared access for large scale file processing
- Can use Lambda to
  - ✓ thumbnail images
  - ✓ transcode videos
  - ✓ index files
  - ✓ process logs
  - ✓ validate content
  - ✓ and aggregate and filter data in real-time


Photograph is taken
Amazon S3
Photo is uploaded to an S3 Bucket
Lambda is triggered
AWS Lambda
Lambda runs image resizing code
Photo is resized into web, mobile, and tablet sizes

Use case (5)

Data processing - Real-time stream processing

- Can use AWS Lambda and Amazon Kinesis to process real-time streaming data for
  - ✓ application activity tracking
  - ✓ transaction order processing
  - ✓ click stream analysis
  - ✓ data cleansing
  - ✓ metrics generation
  - ✓ log filtering
  - ✓ indexing, social media analysis
  - ✓ and IoT device data telemetry and metering


Amazon Kinesis
Social media stream is loaded into Kinesis in real-time
Lambda is triggered
AWS Lambda
Lambda runs code that generates hashtag trend data
Amazon DynamoDB
The hashtag trend data is stored in DynamoDB
Social media trend data immediately available for business users to query

Reference :
AWS Serverless

Thank You!

In our next session:

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

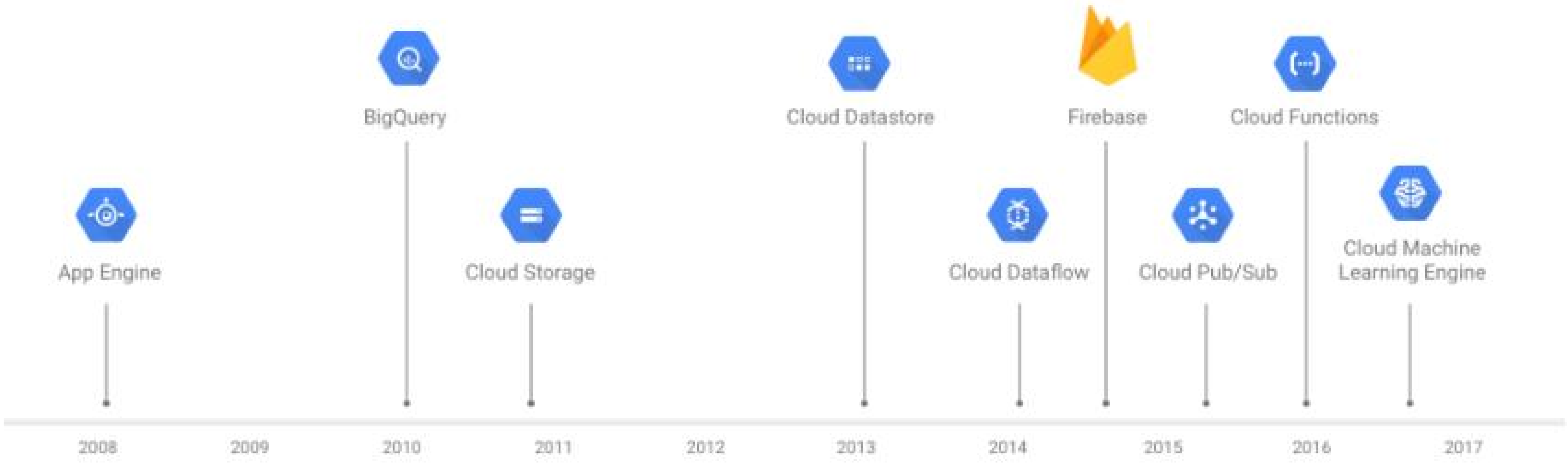
GCP Serverless computing

Chandan Ravandur N

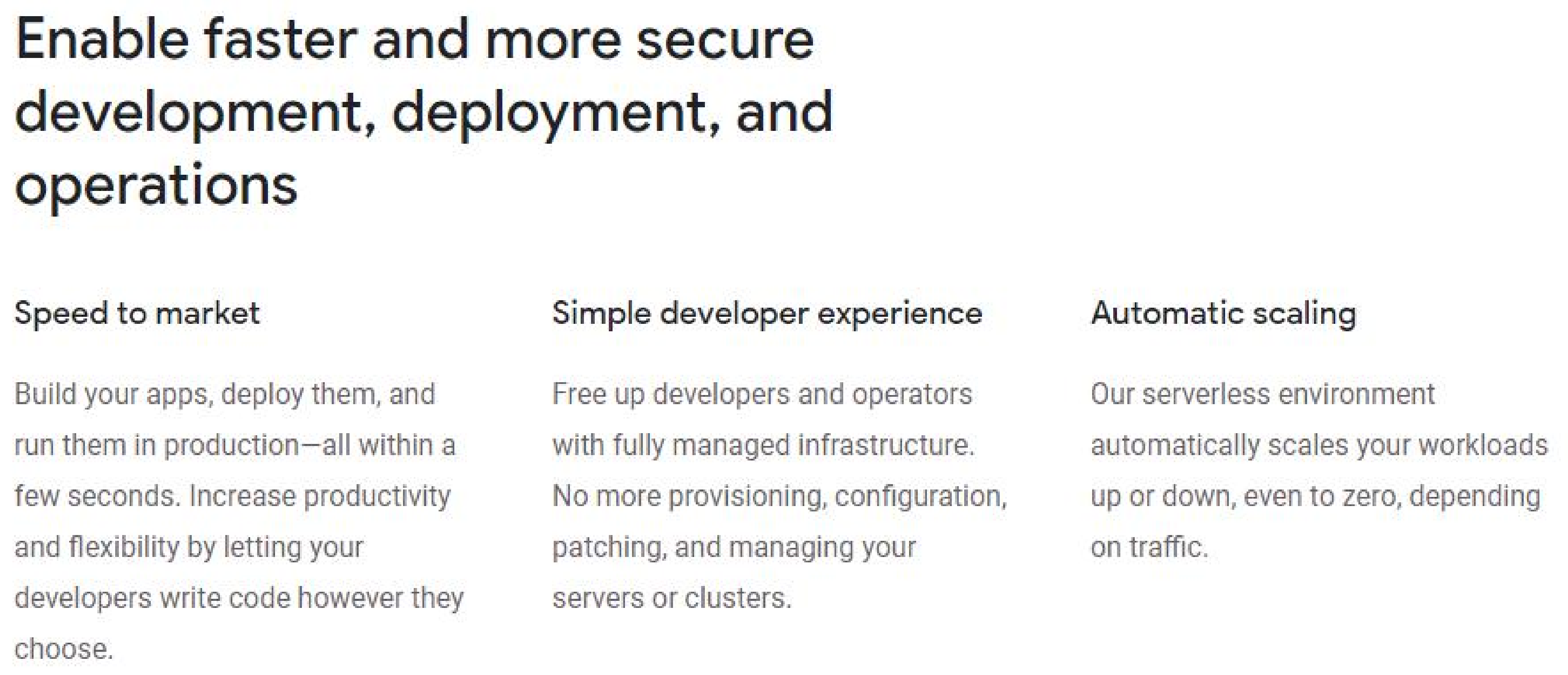
Google Cloud Platform

Serverless

- Google Cloud’s serverless platform allows to
  - ✓ build, develop, and deploy functions and applications
  - ✓ as source code or containers
  - ✓ while simplifying the developer experience by eliminating all infrastructure management



Benefits



Enable faster and more secure development, deployment, and operations

Speed to market

Build your apps, deploy them, and run them in production—all within a few seconds. Increase productivity and flexibility by letting your developers write code however they choose.

Simple developer experience

Free up developers and operators with fully managed infrastructure. No more provisioning, configuration, patching, and managing your servers or clusters.

Automatic scaling

Our serverless environment automatically scales your workloads up or down, even to zero, depending on traffic.

Common use cases for serverless compute products

- Build scalable, secure web apps
  - ✓ Code, build, and deploy scalable applications in a fully managed environment
  - ✓ designed to help developers to succeed with
    - ❖ built-in security
    - ❖ auto scaling
    - ❖ ops management for faster deployment

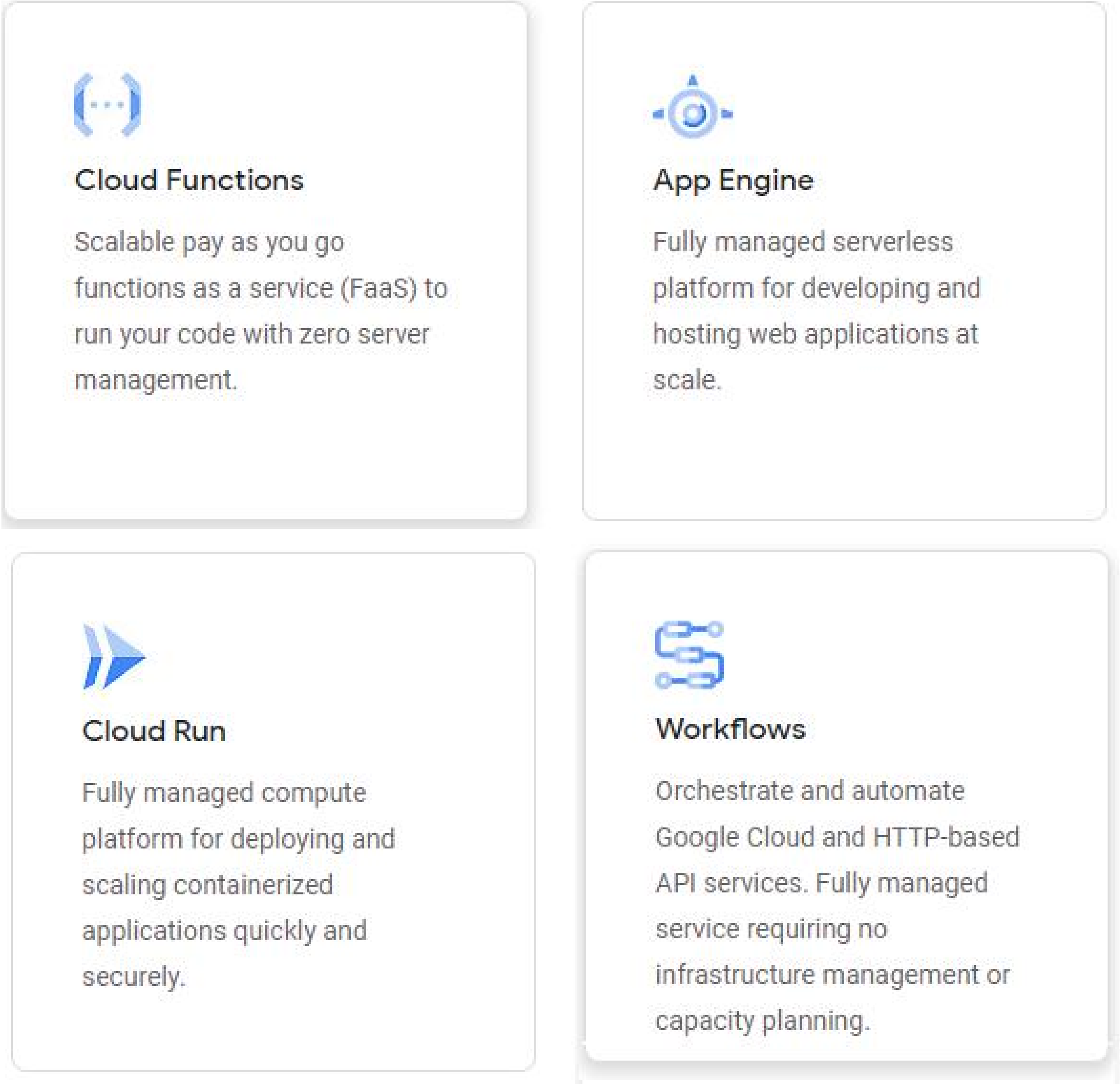
- Develop, deploy, and manage APIs
  - ✓ Build scalable APIs in an environment built for developers to succeed
  - ✓ Can develop REST APIs for web and mobile backends
  - ✓ Manage the connection between different parts of application and internal cloud services

Common use cases for serverless compute products(2)

- Build apps with data processing in mind
  - ✓ Serverless computing environment manages the infrastructure workloads need, in order to handle
    - ❖ auto scaling
    - ❖ Authorization
    - ❖ and event triggers
  - ✓ The pub/sub model of communication makes it easy to ingest and transform large amounts of data and build complex, scalable data pipelines
    - ✓ while saving time on backend confusion

- Automate event orchestration
  - ✓ Automatically validate policies or configurations and perform other scripted automation using event triggers
  - ✓ Serverless computing products can listen to events from other clouds, handle webhooks, and manage distributing events and workloads to other components
  - ✓ This built-in ability makes it straightforward for application to handle complex event needs

GCP Serverless products



Cloud Functions

Scalable pay as you go functions as a service (FaaS) to run your code with zero server management.

App Engine

Fully managed serverless platform for developing and hosting web applications at scale.

Cloud Run

Fully managed compute platform for deploying and scaling containerized applications quickly and securely.

Workflows

Orchestrate and automate Google Cloud and HTTP-based API services. Fully managed service requiring no infrastructure management or capacity planning.

Reference :
GCP Serverless Computing Product pages

Thank You!

In our next session:

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

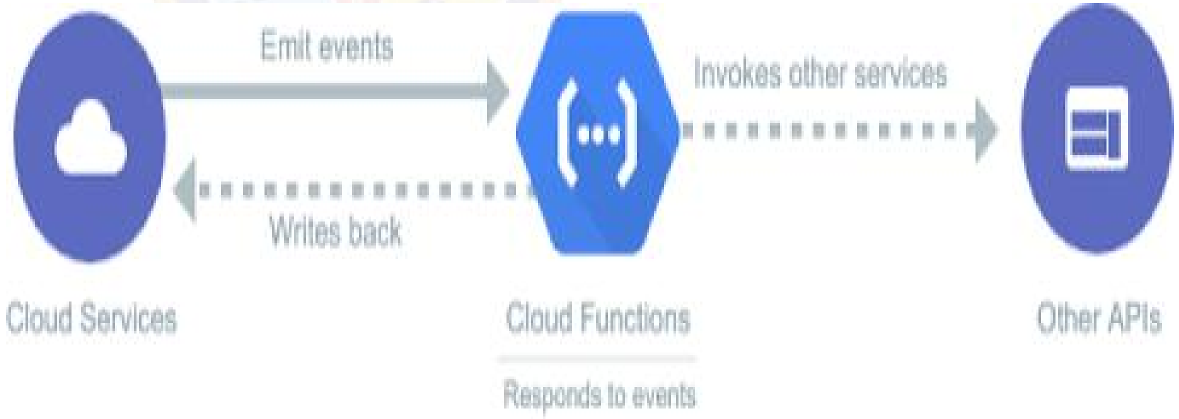
Google Cloud Functions

Chandan Ravandur N

Google Cloud Functions

Scalable pay-as-you-go functions as a service (FaaS) to run your code with zero server management

- No servers to provision, manage, or upgrade
- Automatically scale based on the load
- Integrated monitoring, logging, and debugging capability
- Built-in security at role and per function level based on the principle of least privilege
- Key networking capabilities for hybrid and multi-cloud scenarios



Key features

- Simplified developer experience and increased developer velocity
  - ✓ Cloud Functions has a simple and intuitive developer experience
  - ✓ Just write code and let Google Cloud handle the operational infrastructure
  - ✓ Develop faster by writing and running small code snippets that respond to events
  - ✓ Connect to Google Cloud or third-party cloud services via triggers to streamline challenging orchestration problems

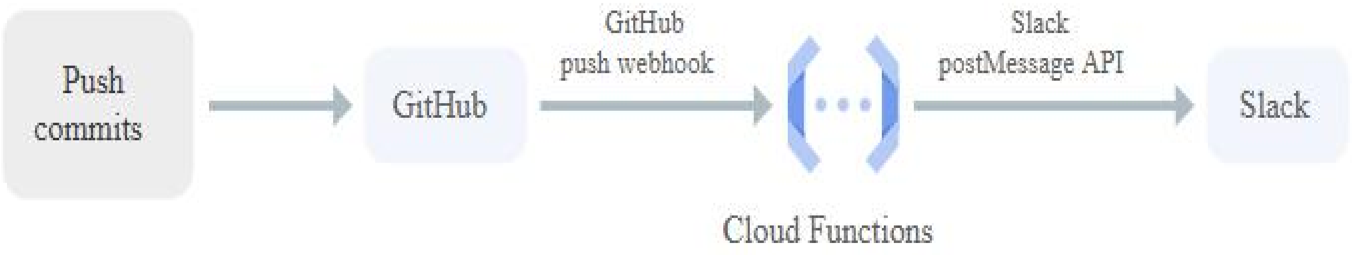
- Pay only for what you use
  - ✓ Only billed for function’s execution time, metered to the nearest 100 milliseconds
  - ✓ Pay nothing when function is idle
  - ✓ Cloud Functions automatically spins up and backs down in response to events

- 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, and other Knative-based serverless environments

Use cases

Integration with third-party services and APIs

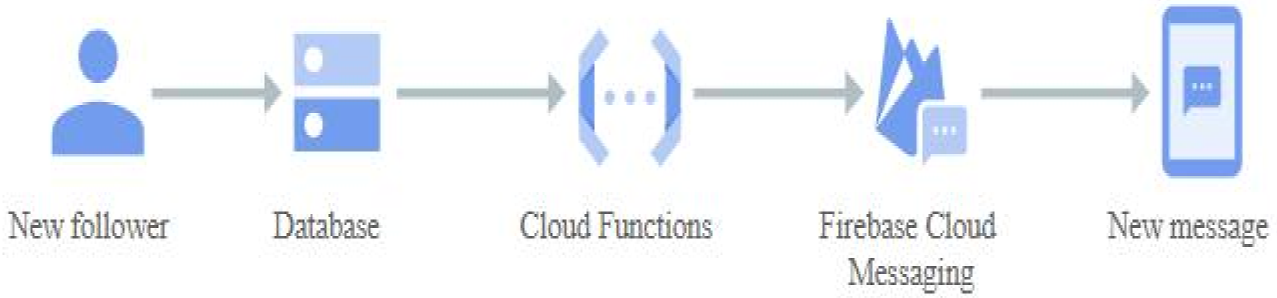
- Use Cloud Functions to
  - ✓ surface out microservices via HTTP APIs
  - ✓ or integrate with third-party services that offer webhook integrations
  - ✓ to quickly extend application with powerful capabilities such as
    - ❖ sending a confirmation email after a successful Stripe payment
    - ❖ or responding to Twilio text message events



Use cases (2)

Serverless mobile back ends

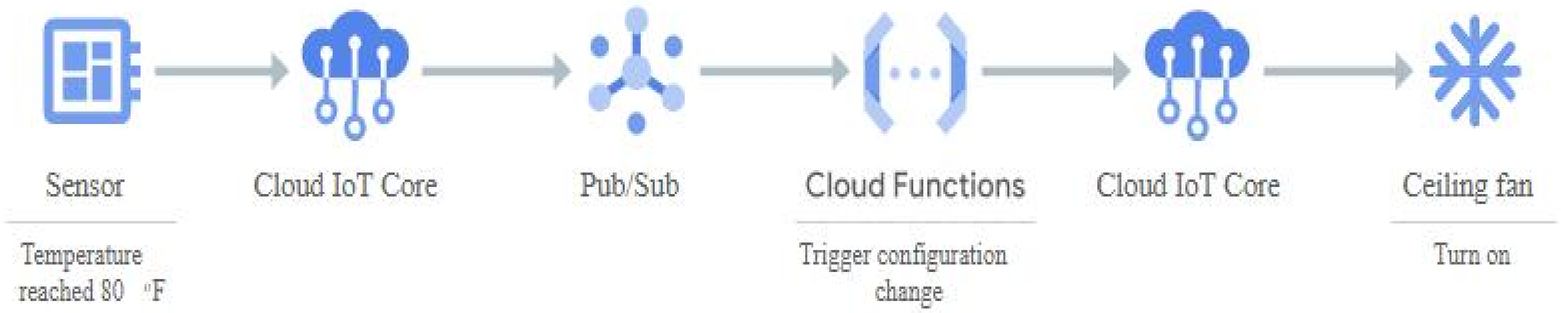
- Use Cloud Functions directly from Firebase to extend application functionality without spinning up a server
- Run code in response to user actions, analytics, and authentication events
  - ✓ to keep users engaged with event-based notifications
  - ✓ to offload CPU- and networking-intensive tasks to Google Cloud



Use cases (3)

Serverless IoT back ends

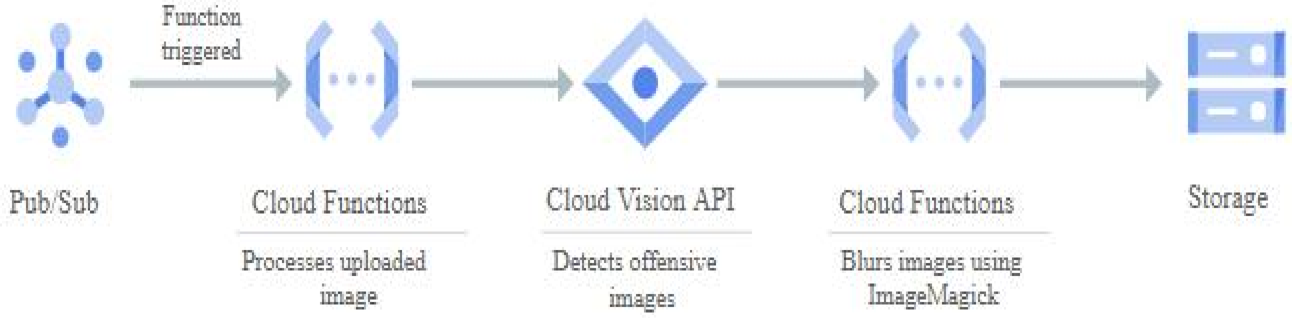
- Use Cloud Functions with Cloud IoT Core and other fully managed services to build back ends for
  - ✓ Internet of Things (IoT) device telemetry data collection
  - ✓ real-time processing
  - ✓ and analysis
- Cloud Functions allows to apply custom logic to each event as it arrives



Use cases (4)

Real-time stream processing

- Use Cloud Functions to respond to events from Pub/Sub to process, transform, and enrich streaming data in
  - ✓ transaction processing
  - ✓ click-stream analysis
  - ✓ application activity tracking
  - ✓ IoT device telemetry
  - ✓ social media analysis
  - ✓ and other types of applications



Reference :
GCP Serverless Computing Product pages

Thank You!

In our next session: