Google Cloud Computing Foundation Course
Jimmy Tran
SMB Growth Program Manager
Google Cloud

Lecture-39
Using Apigee

(Refer Slide Time: 00:08)

Apigee Edge: A platform for developing and managing APIs

Design, Secure, Publish, Analyze, Monitor, and Monetize APIs
Manage microservices, Leverage developer portal
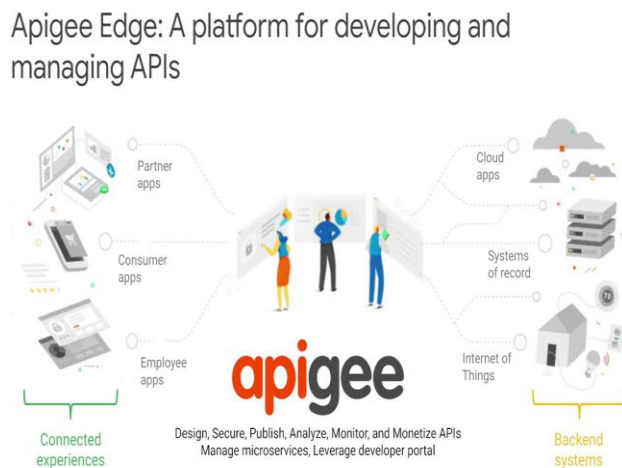
In this topic you're going to learn about Apigee edge another platform for managing api's. Apigee edge allows you to front your services with a proxy layer. An API proxy is your interface to developers that want to use your back-end services. Rather than having them consume those services directly they acts as an edge api proxy that you create. With the proxy you can provide value-added features such as security, rate limiting, quotas, caching and persistence, analytics transformations, fault handling and much more.

Many users of apigee edge are provided a software service to other companies and those features come in handy because the back-end services for Apigee edge need not be in GCP engineers also often use it when they are working to take a legacy application apart. Instead of replacing a monolithic application in one risky move they can instead use Apigee edge to peel off its services one-by-one. Standing up micro services to implement each intern until the legacy application can finally be retired.

(Refer Slide Time: 01:17)

An API gateway can retrieve data from multiple services with a single request
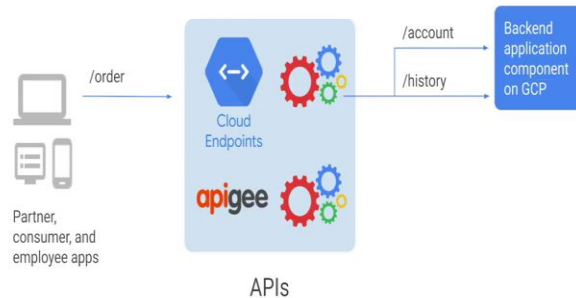
An API gateway creates a layer of abstraction and insulates the clients from the partitioning of the application into micro services. You can use Cloud endpoints to implement API gateways additionally the API for your application can run on backends such as App Engine, Google kubernetes Engine or compute engine. If you have legacy applications that cannot be refactored and moved to the cloud consider implementing api's as a facade or adapter layer.

Each consumer can then invoke these modern api's to retrieve information from the backend instead of implementing functionality to communicate using outdated protocols and disparate interfaces.