



Lambda, Serverless & Serverless Microservices

Author: **Than Dao**

Date: **9/3/2019**



Content

- Lambda functions
- API Gateway
- Typescript
- Serverless framework
- Serverless Microservices
- Some AWS services and Demo session



Lambda functions

- Virtual functions
- Based on Serverless architecture
- Responsible only for your code
- Limited by time execution (15mins)
- Scale automatically



Lambda functions

- Languages: NodeJS, Java, Python, C#, ...
- Stores code in Amazon S3
- Integration: API gateway, S3, SQS,...
- Can be triggered from AWS services
- Pay on demand

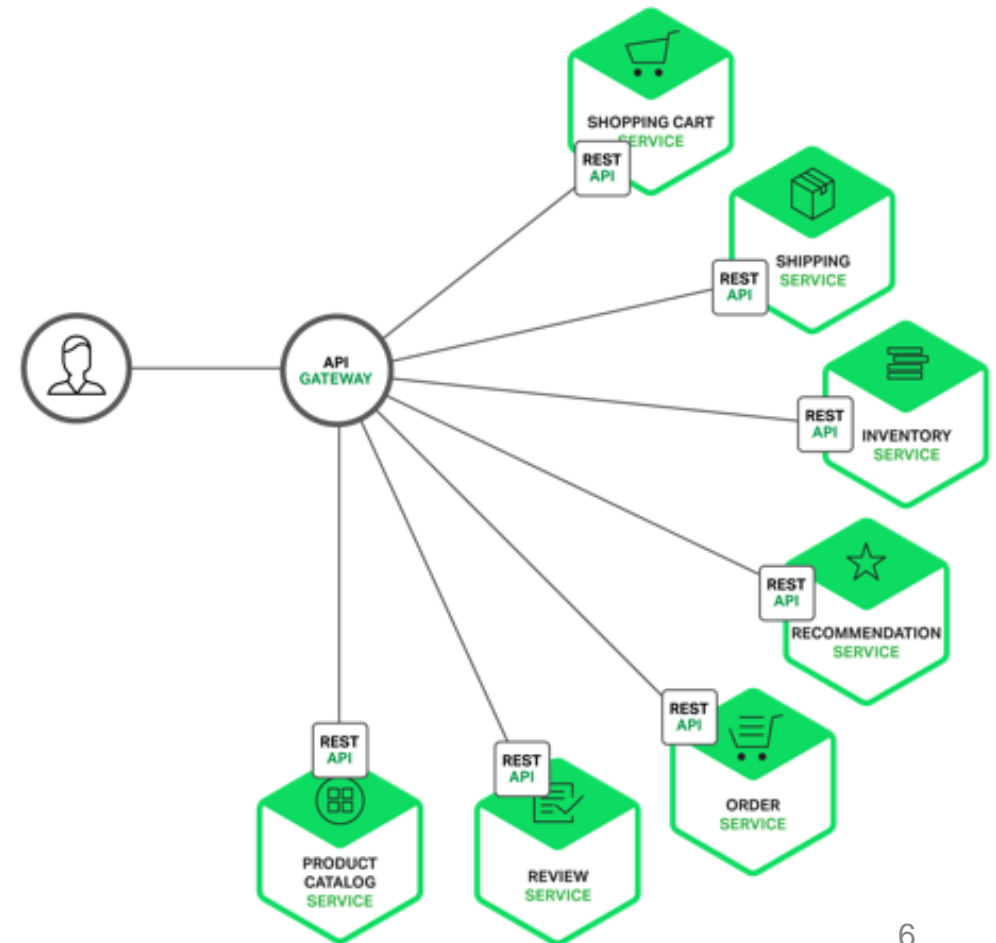
Lambda functions

Pros	Cons
- Run apps without managing server	- 1000 concurrent executions
- Not running your app for 24/7	- Can not control environment
- Pay on demand	- Limit function execution time: 15mins
- Auto scaling	
- Trigger by multiple AWS services	

API Gateway

- Sit in front of API, single entryway into a system
- Takes all API call, routes to appropriate

Microservices, APIs



API Gateway

Pros	Cons
- Hide your system's structure	- Affect performance due to lots of request
- Client can get data by single hit	- Can become single point of failure
- Logging, authentication	- Increase the complexity of the system
- Can handle partial failure	

Typescript

- A superset of JavaScript
- Created by Microsoft in 2010, release in 2012
- Open source, written by Typescript itself
- Using static typing, and is optional
- Inheritance – OOP principles

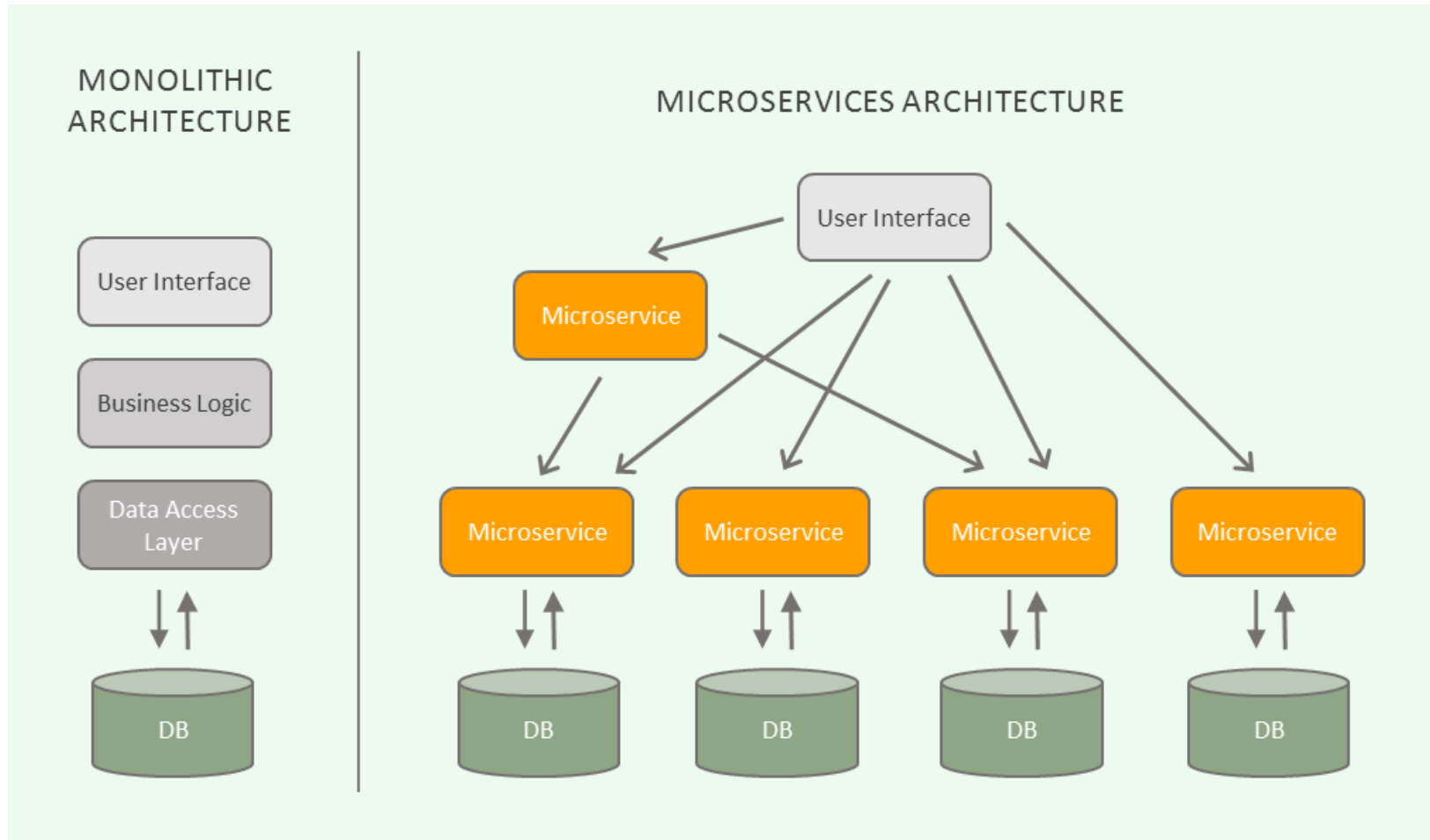


Serverless Framework

- Easy way to manage Lambda functions
- Integrating well with CI/CD tools
- Having stack's Cloudformation supports
- Support multiple language: NodeJS, Java,..
- Commands: deploy, create,..



Microservices or Microservice Architecture



Serverless Microservices

- Based on Microservice architecture
- Using API Gateway
- Only run when they are needed by applications
- Can be FaaS

Serverless Microservices

Pros	Cons
- Independently develop and deploy services	- Testing can become complicated
- Different services can use different languages	- Managing whole products can become complicated
- Easy integration and automatic deployment	
- Update services separately	
- Better fault isolation	

Demo

- Some AWS services: S3, Lambda, API Gateway, Cloudformation, Cloudwatch Logs
- Example: Lambda functions, Serverless Framework, Typescript
- Run on local by serverless offline
- Deploy to AWS

Thanks for listening!

Q&A?