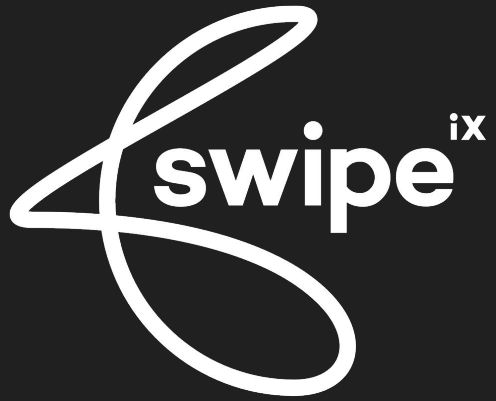
swipe
ix

Welcome

PHP as a custom runtime environment on Lambda @ Swipe

01

Modern Monolithic vs
Microservice Architectures

02

AWS Supports both
Monolithic and Microservice

03

PHP Lambda Demo

04

Further Reading

Swipe iX is a specialist software development company with a reputation for building scalable and secure technology..

Despite having a logically modular architecture (MVC), the application is packaged and deployed as a monolith.

- Presentation Layer (User Interface / Rest API)
- Business Logic (login, different CRUDs)
- Data Interface (database communication)
- Database

Modern Monolithic Pros:
- Easy to: Develop / Test / Deploy
- Simple to Scale: horizontally, just add another copy behind LB
- Less management: Logging, debug, caching and performance monitoring, it’s all one unit.

Modern Monolithic Cons:
- Understanding: complex system can be hard manage
- Scalability: Individual components can't be scaled, only as a whole
- New Technology: Mostly required a complete rewrite
- Making change: downstream effects your not aware of, single bug can bring down the entire stack

The entire stack is split up into independently deployable modules, each service covers its own scope and can be updated, deployed, and scaled independently.

- Presentation Layer: static S3 bucket
- Presentation Layer: API gateway (Rest API) made public for mobile / web / desktop apps
- Business Logic & Data store: Each endpoint contains its own business logic & data source

Microservice Pros:
- Individuality: Services updated & deployed individually.
- Understanding: Don't have to understand entire application.
- Scalability: Each service can be scaled individually to requirement.
- New Technology: Complete rewrites is not required.

Microservice Cons:
- Comlex: All the services together that make up your application
- Management: Logging, debugging, health checks all over the place
- Testing: Single unit test will not test the entire stack

Classic Monolithic:

- Load Balancer
- EC2 (containers, auto-scaling)
- RDS (MySQL, PostgreSQL, Oracle, Microsoft SQL, MariaDB)

Common Microservice Components:

- S3/Static S3 (Extraction of presentation layer)
- API Gateway (Public & Private)
  comms to your stack or between services
- Lambda (application logic)
  .Net, Go, Java, NodeJs, Python, Ruby and Custom runtimes
- Application Integration
  SQS, SNS, MQ, SWF, Step-functions
- Data Layer
  RDS, DynamoDB (NoSql db), Aurora

AWS Supports Monolithic and Microservice

Migrating to Microservice architecture does not mean you have to replace everything at once or event that you have to choose between the two when building your application.

You can start replacing piece by piece, or make use of a combination of the two architectures, whatever suits your business requirements.

Lambda lets you listen to API gateway events and many other service's events, and in response trigger a function call lambda, per HTTP request.

Enabling you to build a web service completely event driven. No server needed!

With lambda custom environments we are now able to build these services in PHP even though they are not natively supported.

PHP on Lambda

Lambda:

- Lambda is a AWS service branding as Function-as-a-Service (FaaS)
- It lets you run code as a function in response to other AWS service's events

Lambda Layers:

- A function can use up to 5 layers at a time. (total unzip size of 250 MB)
- Common layers between lambda functions don't get invoked each time

PHP as a custom runtime environment on Lambda demo

Don’t be lazy

Further Reading

My Example was based of:
https://aws.amazon.com/blogs/apn/aws-lambda-custom-runtime-for-php-a-practical-example/
https://aws.amazon.com/blogs/compute/scripting-languages-for-aws-lambda-running-php-ruby-and-go/
https://akrabat.com/serverless-php-on-aws-lamda/

Bref - Laravel & Symfony
- https://bref.sh/docs/

Vapor (paid for service)
- https://vapor.laravel.com/

JFGI - Endless supply of online articles and examples

Thank You.

Theo Ferreira
theo@swipeix.com

Twitter:
Facebook:
TikTok:
I'm a developer don't bother me.

www.swipeix.com