Serverless (Web)Apps with AWS API Gateway and Lambda

Soenke Ruempler
@s0enke
ruempler.eu

PHP Unconference 2015-09-20

Any non german speaking folks here?

Disclaimer: “alpha” content

Hello.

I am Soenke.

I work for Jimdo.

I do a lot of
ops and infrastructure
management.

I don’t like
ops and infrastructure
management.

When you run web app with scale, you have to care about many things.

to name a few ...

Racks, Server, Router, Switches, Networks, Flood/DoS protection, OS updates, OS upgrades, Server provisioning, App Deployment, Inventory Management, Development environment, Infrastructure (perimeter) security, Staging environments, Data storage+backup+restore, Database scaling+upgrades, API provisioning, load balancing, scaling , caching(+invalidation), API versioning, API documentation, API SDK generation, Identity management, Login management, User Authentication, API Authorization, Legacy API Facade, Application Logging, Application Metrics, Application Monitoring, Feature Lifecycle Management, Circiut breakers

But I don’t want to care about all this stuff.

So let’s get rid of maintaining this stuff by ourselves.

Step by step.

Level 1: Infrastructure as Service (Virtual Servers and Networks)

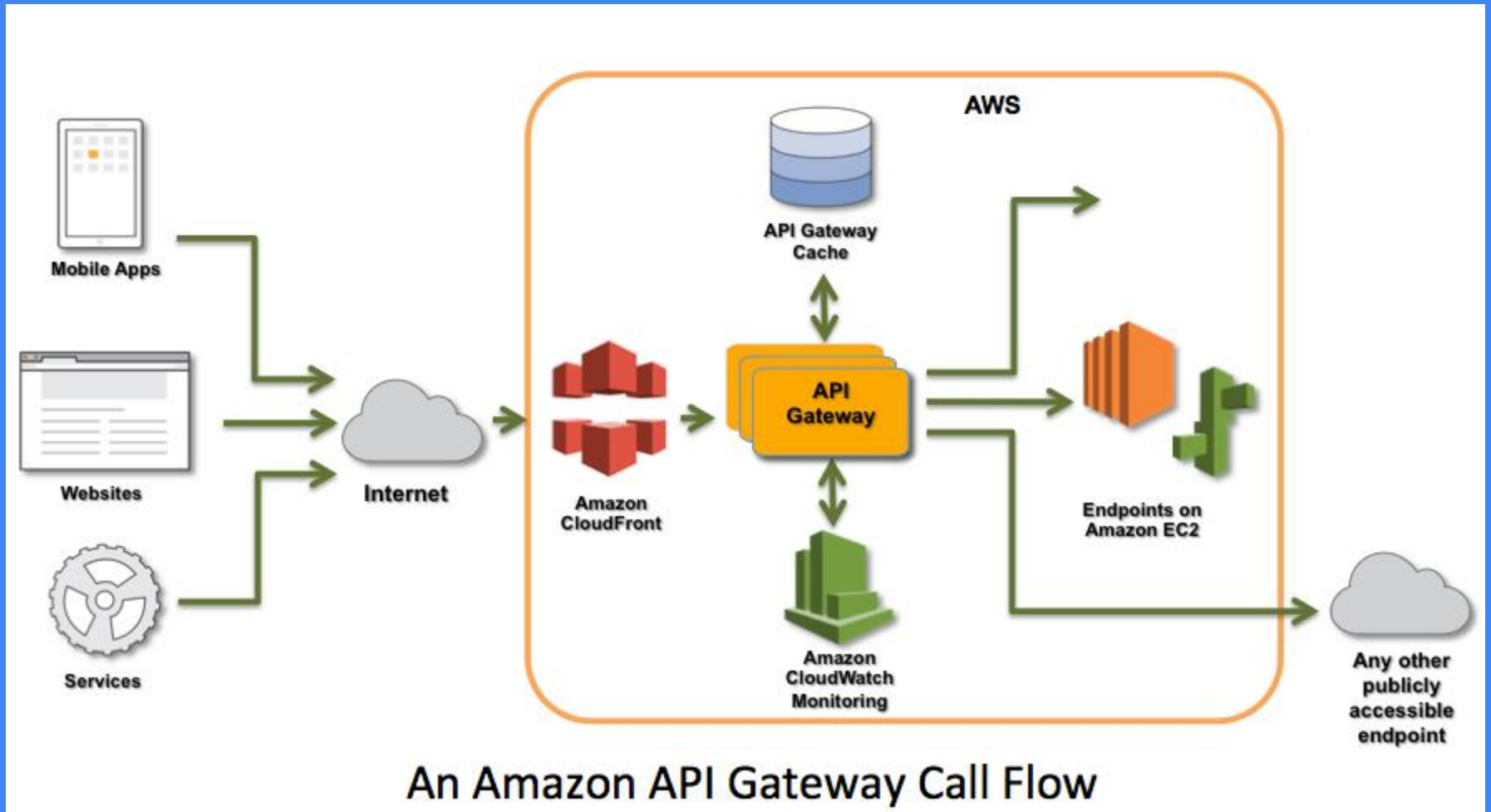
Level 1a: Database as Service (e.g. Google Cloud SQL, AWS RDS/DynamoDB)

Racks, Server, Router, Switches, Networks, Flood/DoS protection, OS updates, OS upgrades, Server provisioning, App Deployment, Inventory Management, Development environment, Infrastructure (perimeter) security, Staging environments, Data storage+backup+restore, Database scaling+upgrades, API provisioning, load balancing, scaling , caching(+invalidation), API versioning, API documentation, API SDK generation, Identity management, Login management, User Authentication, API Authorization, Legacy API Facade, Application Logging, Application Metrics, Application Monitoring, Feature Lifecycle Management, Circiut breakers

Level 2: Platform as a Service (e.g. Heroku, Docker …)

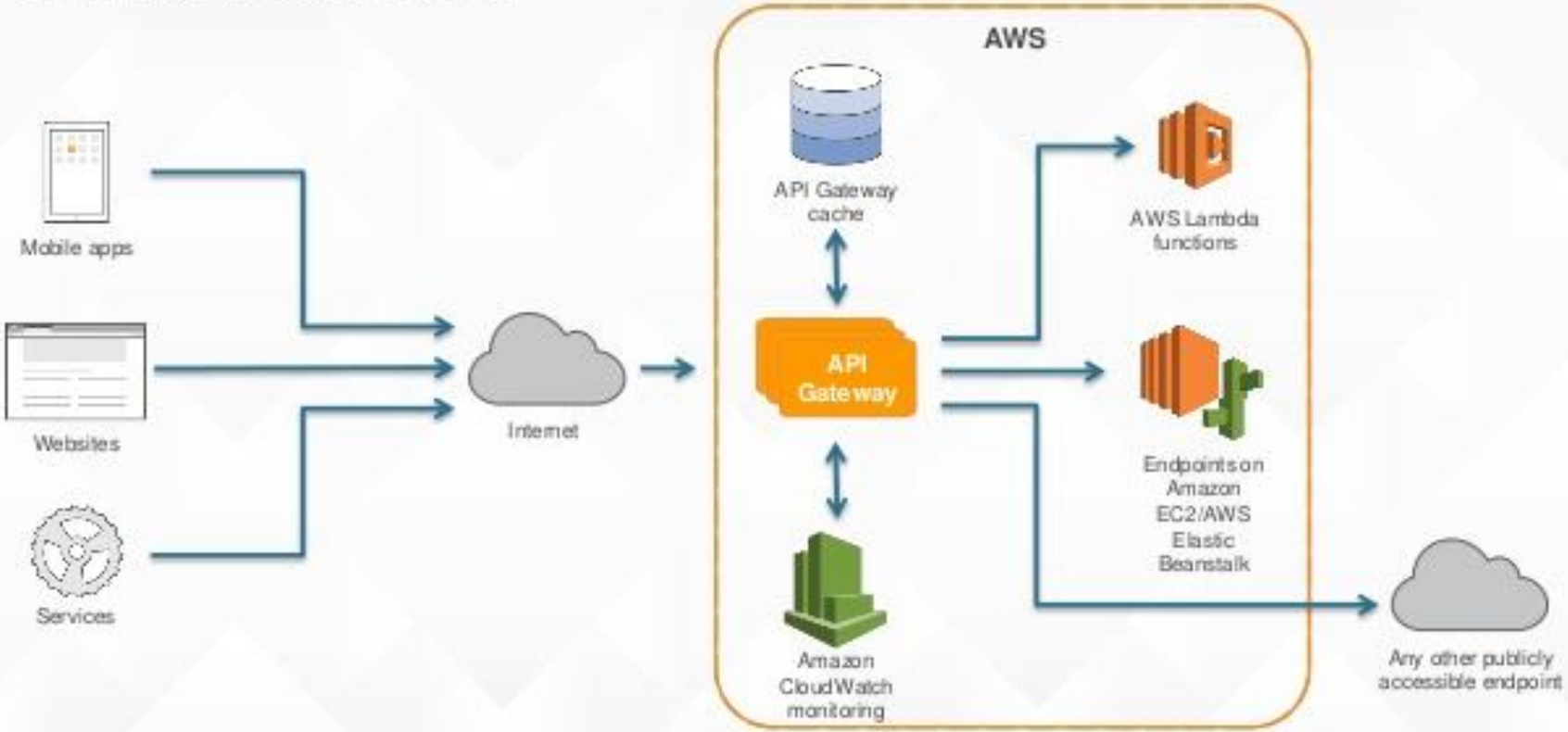
Racks, Server, Router, Switches, Networks, Flood/DoS protection, OS updates, OS upgrades, Server provisioning, App Deployment, Inventory Management, Development environment, Infrastructure (perimeter) security, Staging environments, Data storage+backup+restore, Database scaling+upgrades, API provisioning, load balancing, scaling , caching(+invalidation), API versioning, API documentation, API SDK generation, Identity management, Login management, User Authentication, API Authorization, Legacy API Facade, Application Logging, Application Metrics, Application Monitoring, Feature Lifecycle Management, Circiut breakers

Level 3: APIs as a service (e.g. API Gateway)



An Amazon API Gateway Call Flow

An API call flow



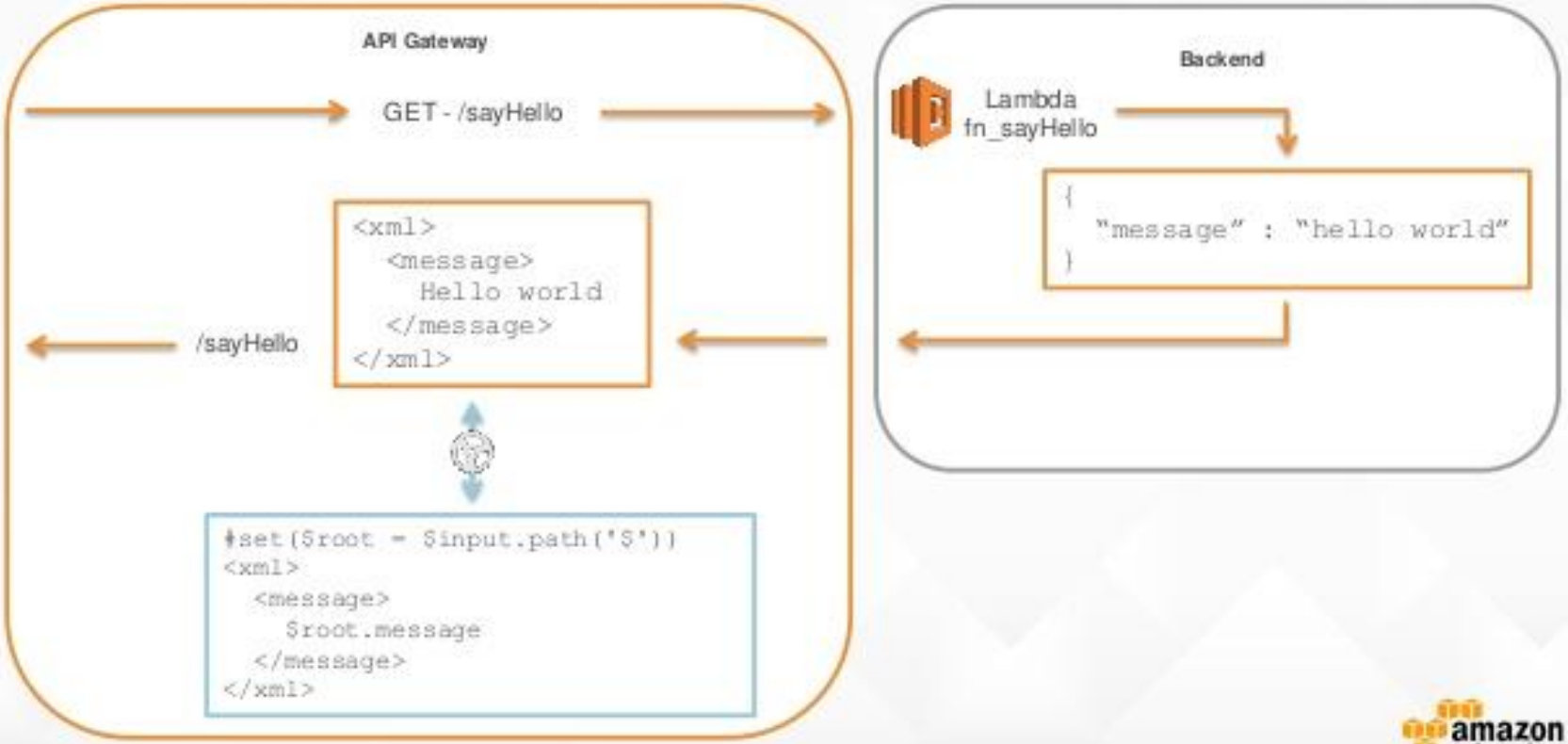
amazon web services

Swagger.io -> API GW

https://github.com/awslabs/aws-apigateway-swagger-importer

SDK generator
for iOS, Android, JS

Transform Example: JSON to XML



amazon web services

Racks, Server, Router, Switches, Networks, Flood/DoS protection, OS updates, OS upgrades, Server provisioning, App Deployment, Inventory Management, Development environment, Infrastructure (perimeter) security, Staging environments, Data storage+backup+restore, Database scaling+upgrades, API provisioning, load balancing, scaling , caching(+invalidation), API versioning, API documentation, API SDK generation, Identity management, Login management, User Authentication, API Authorization, Legacy API Facade, Application Logging, Application Metrics, Application Monitoring, Feature Lifecycle Management, Circiut breakers

Level 4: Program execution as a service
(e.g. AWS Lambda)

“AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources for you.”

First 1 million requests per month are free

$0.20 per 1 million requests thereafter ($0.0000002 per request)

Currently node.js and JVM runtimes supported

AWS Lambda minimal example

```
exports.handler = function (event, context)
{
    context.done(null, {"Hello":"World"});
};
```

Racks, Server, Router, Switches, Networks, Flood/DoS protection, OS updates, OS upgrades, Server provisioning, App Deployment, Inventory Management, Development environment, Infrastructure (perimeter) security, Staging environments, Data storage+backup+restore, Database scaling+upgrades, API provisioning, load balancing, scaling , caching(+invalidation), API versioning, API documentation, API SDK generation, Identity management, Login management, User Authentication, API Authorization, Legacy API Facade, Application Logging, Application Metrics, Application Monitoring, Feature Lifecycle Management, Circiut breakers

Level 5: Identity as a Service (e.g. AWS Cognito)

Racks, Server, Router, Switches, Networks, Flood/DoS protection, OS updates, OS upgrades, Server provisioning, App Deployment, Inventory Management, Development environment, Infrastructure (perimeter) security, Staging environments, Data storage+backup+restore, Database scaling+upgrades, API provisioning, load balancing, scaling , caching(+invalidation), API versioning, API documentation, API SDK generation, Identity management, Login management, User Authentication, API Authorization, Legacy API Facade, Application Logging, Application Metrics, Application Monitoring, Feature Lifecycle Management, Circiut breakers

“The essence of LEAN is doing nothing and still earning money”

- someone at devopsdays Barcelona 2012

Ok, Live Demo!

Resources

- http://www.kinvey.com/blog/3194/what-is-the-motivation-of-baas-adopters
- http://www.slideshare.net/AmazonWebServices/build-and-manage-your-apis-with-amazon-api-gateway
- http://www.slideshare.net/adriancockcroft/dockercon-state-of-the-art-in-microservices
- https://github.com/awslabs/api-gateway-secure-pet-store
- http://swagger.io/
-