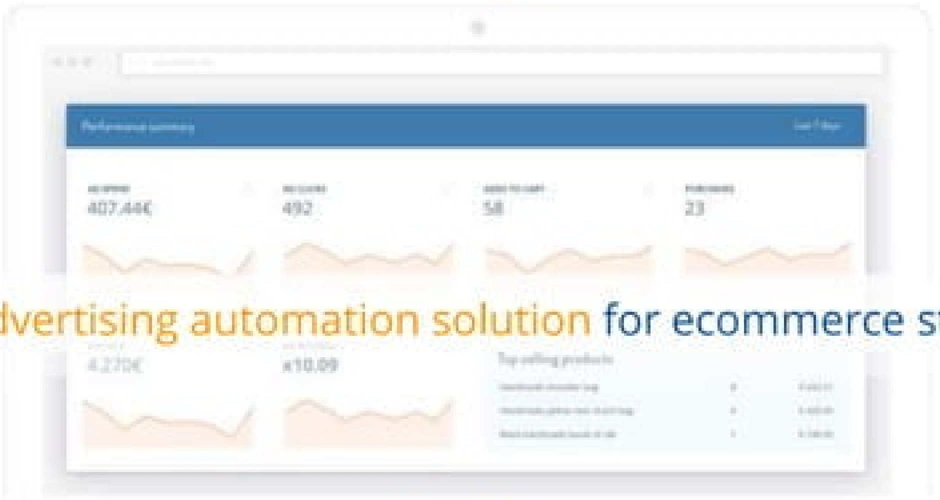
“INTRODUCTION TO AWS LAMBDA WITH PYTHON”

George Zografos
akis@adaplo.com

Software Engineer at

adaplo



"An advertising automation solution for ecommerce stores"

System yellowday Folli Follie mat.

Agenda

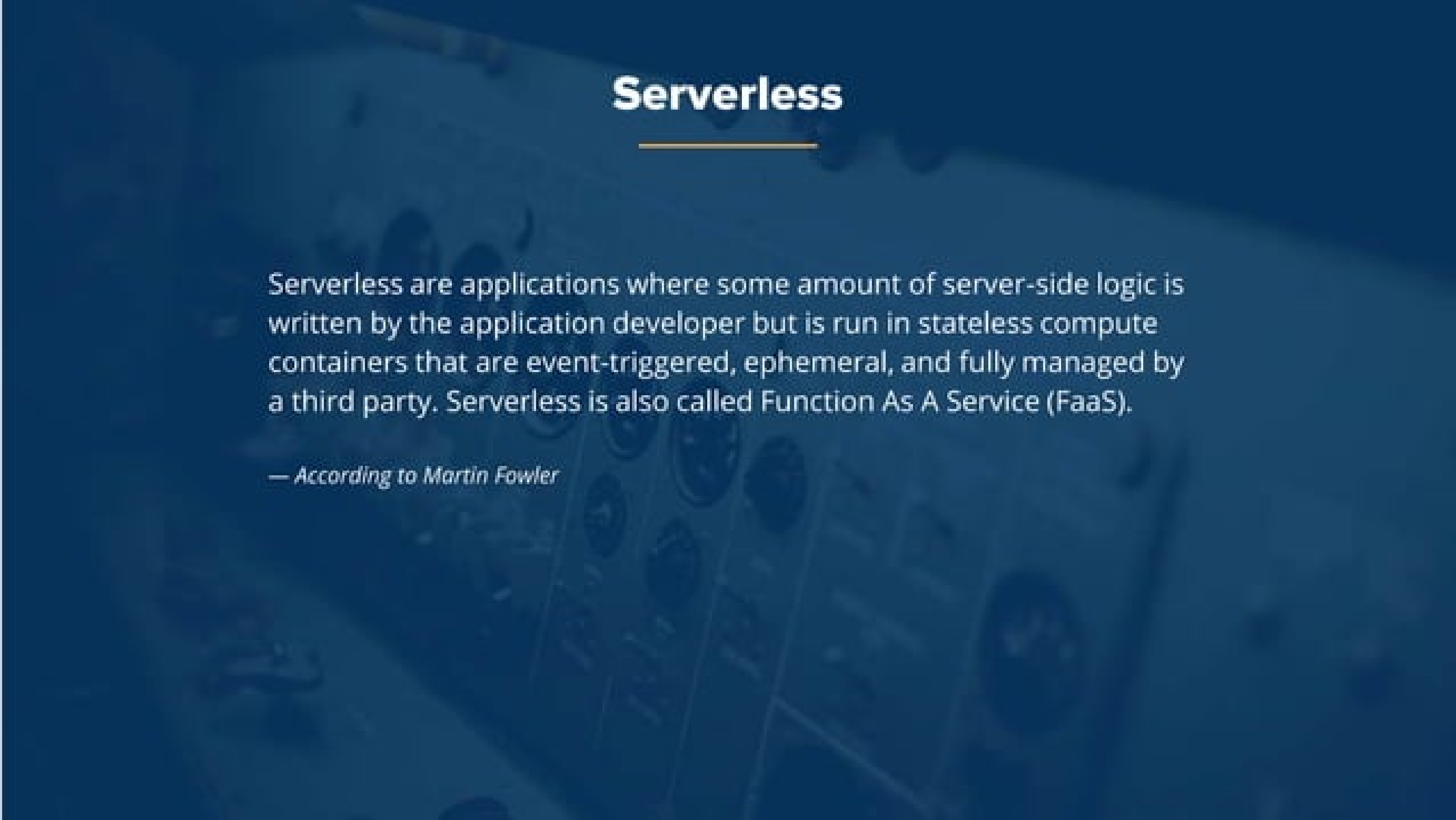
1. About AWS Lambda
2. Getting Started
3. Benefits vs. Drawbacks
4. Use Cases
5. Useful Information
6. Python Demo

About AWS Lambda

AWS Lambda

○ **AWS Lambda** lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration.

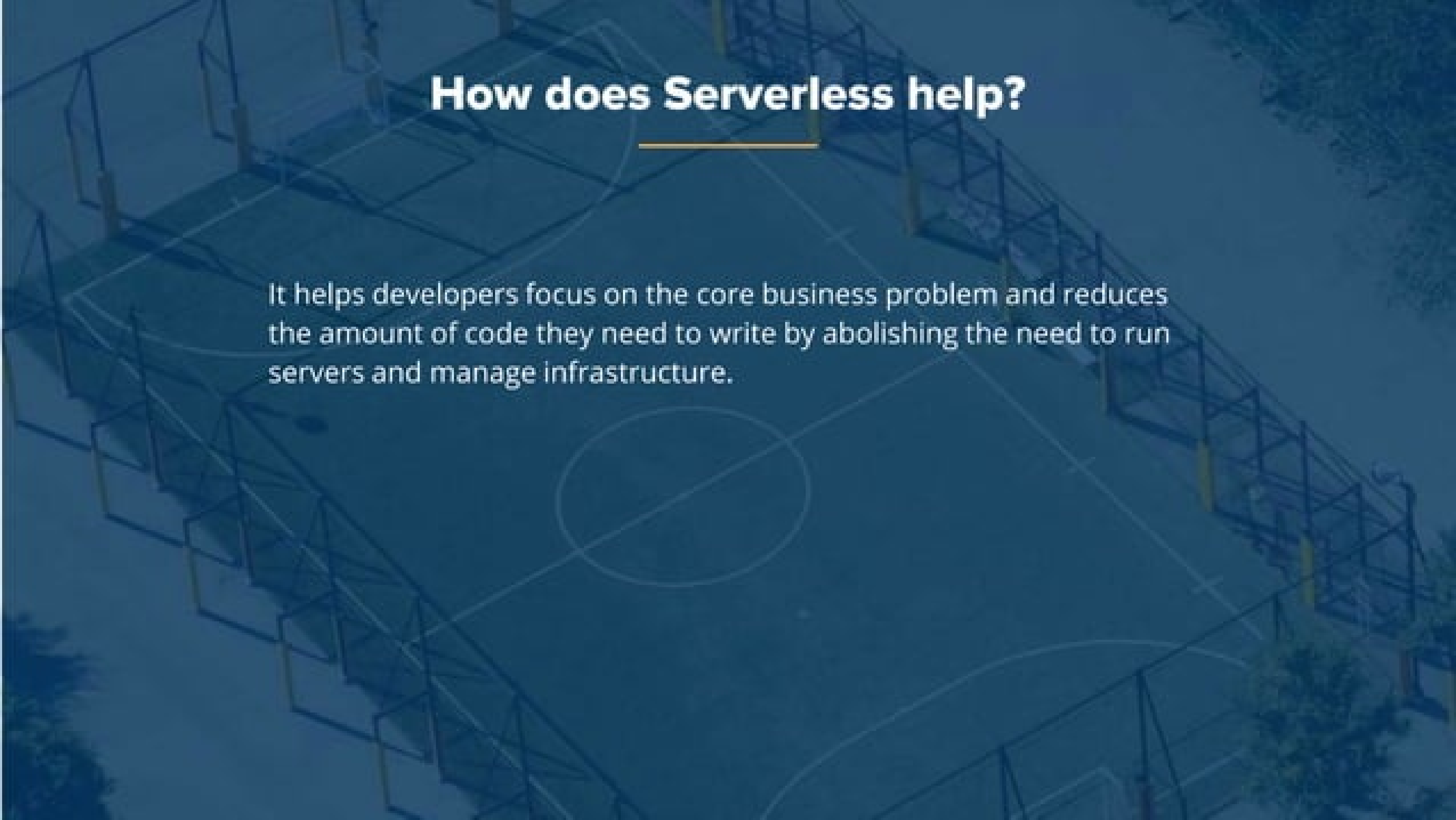
— According to AWS Developer Guide



Serverless

Serverless are applications where some amount of server-side logic is written by the application developer but is run in stateless compute containers that are event-triggered, ephemeral, and fully managed by a third party. Serverless is also called Function As A Service (FaaS).

— According to Martin Fowler

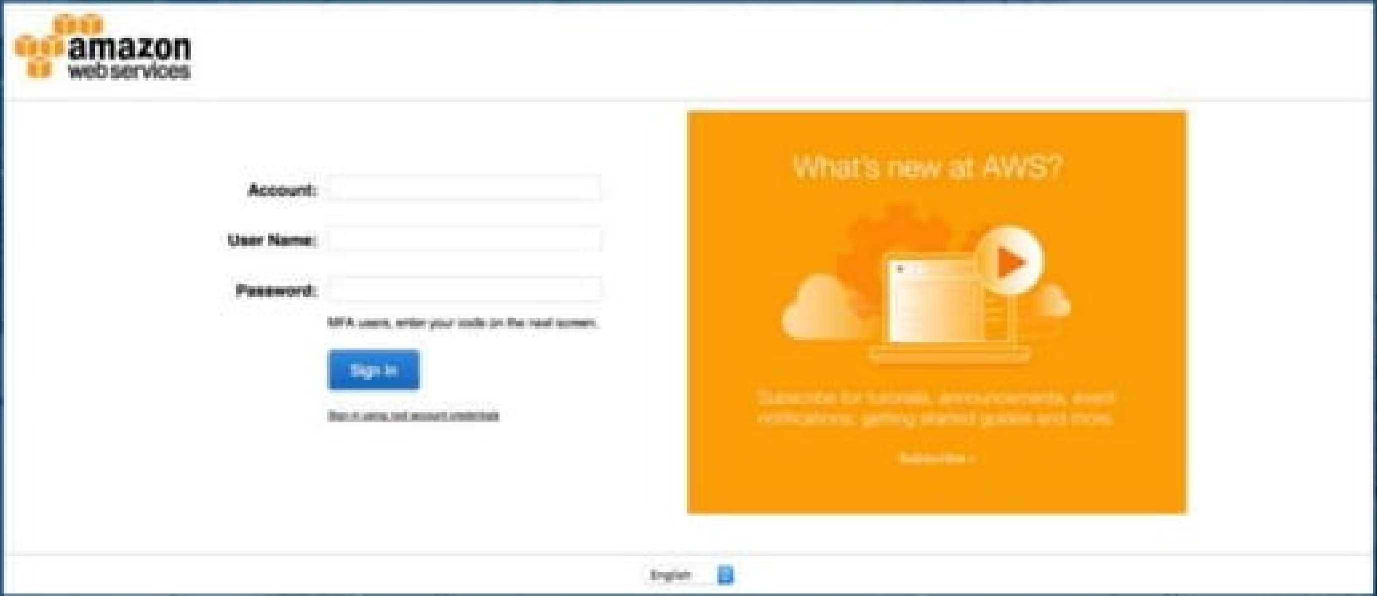


How does Serverless help?

It helps developers focus on the core business problem and reduces the amount of code they need to write by abolishing the need to run servers and manage infrastructure.

Getting Started

Log in to AWS



Account:

User Name:

Password:

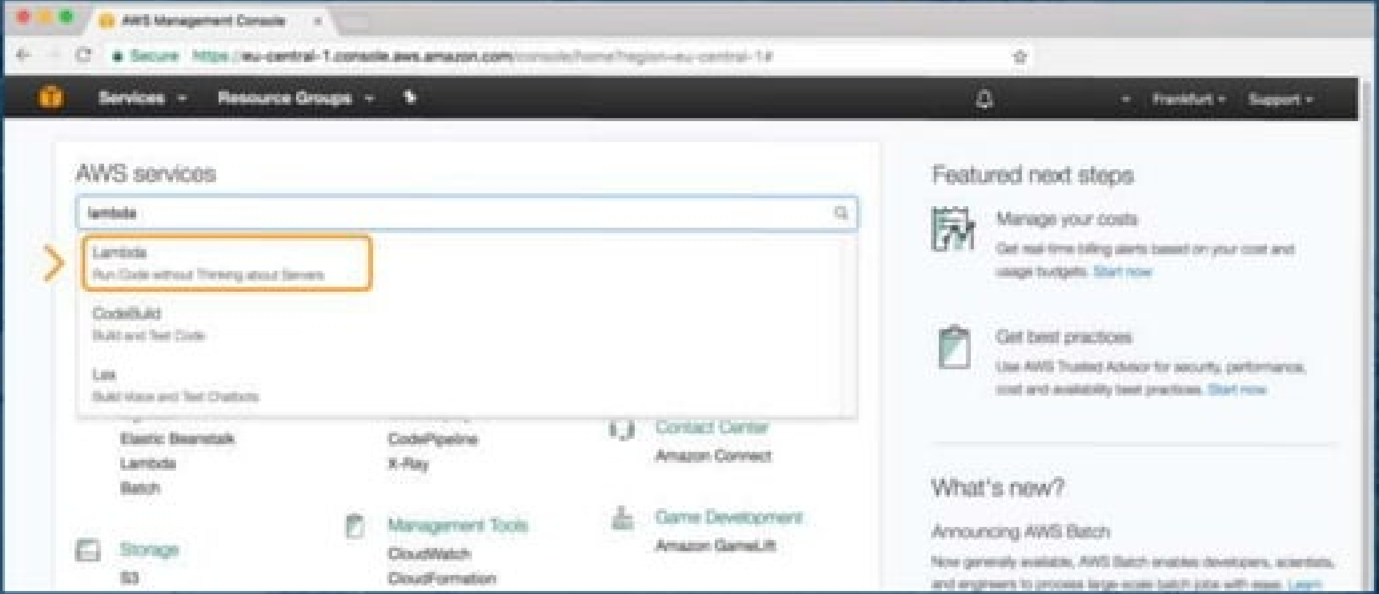
MFA users, enter your code on the next screen.

Sign in using root account credentials

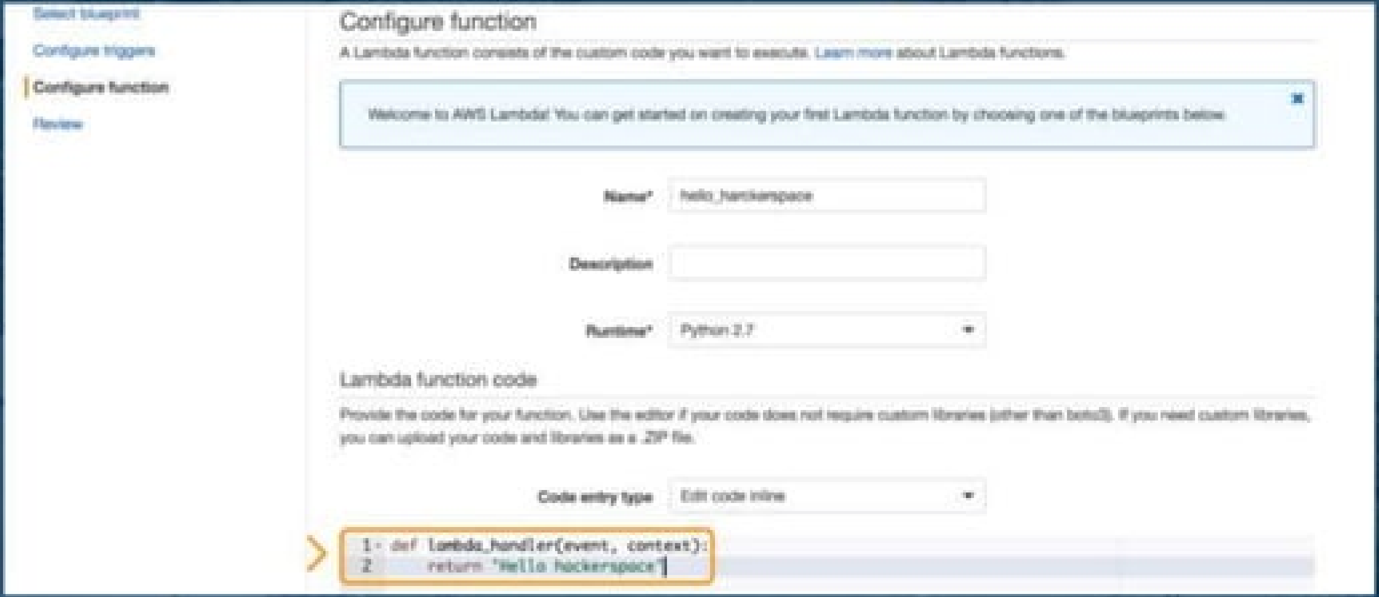
What's new at AWS?

Subscribe for tutorials, announcements, event notifications, getting started guides and more.

Go To Lambda


AWS Management Console
Secure https://eu-central-1.console.aws.amazon.com/console/home?region=eu-central-1#
Services
Resource Groups
Frankfurt
Support
AWS services
lambda
Lambda
Run Code without Thinking about Servers
CodeBuild
Build and Test Code
Lex
Build Voice and Text Chatbots
Elastic Beanstalk
Lambda
Batch
CodePipeline
X-Ray
Contact Center
Amazon Connect
Storage
S3
Management Tools
CloudWatch
CloudFormation
Game Development
Amazon GameLift
Featured next steps
Manage your costs
Get real-time billing alerts based on your cost and usage budgets. Start now
Get best practices
Use AWS Trusted Advisor for security, performance, cost and availability best practices. Start now
What's new?
Announcing AWS Batch
Now generally available, AWS Batch enables developers, scientists, and engineers to process large-scale batch jobs with ease. Learn

Create your function



Select blueprint

Configure triggers

Configure function

Review

Configure function

A Lambda function consists of the custom code you want to execute. Learn more about Lambda functions.

Welcome to AWS Lambda! You can get started on creating your first Lambda function by choosing one of the blueprints below.

Name*: hello_hackerspace

Description:

Runtime*: Python 2.7

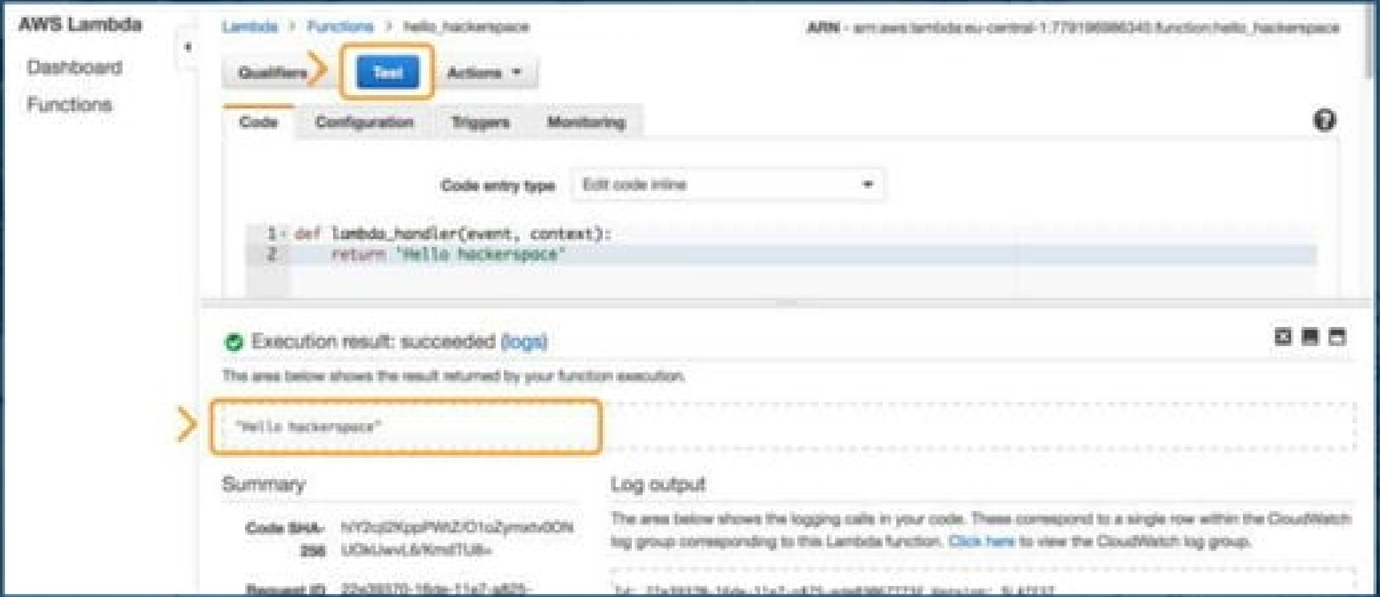
Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than boto3). If you need custom libraries, you can upload your code and libraries as a .ZIP file.

Code entry type: Edit code inline

```python
def lambda_handler(event, context):
    return "Hello hackerspace"
```

Execute the lambda function



ARN - arn:aws:lambda:eu-central-1:779196986340:function:hello_hackerspace

```python
def lambda_handler(event, context):
    return 'Hello hackerspace'
```

Execution result: succeeded (logs)

The area below shows the result returned by your function execution.

```
"Hello hackerspace"
```

Summary

Code SHA-256 NY3cj2KppPWNZ/O1o2ymxtv0ON U0kUwvL6/KmdTU8=

Request ID 22e26570-18de-11e7-a825-

Log output

The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

Lambda function event sources

1. HTTP POST request
2. AWS CLI
3. Python **boto3** library
4. Different types of event sources from other AWS resources (API Gateway, S3, Kinesis, SNS, Cloudwatch Logs/Events)

Benefits vs. Drawbacks

Benefits

Built-in fault **tolerance**

No infrastructure to manage

Scale automatically up or down

Pay only for what you use

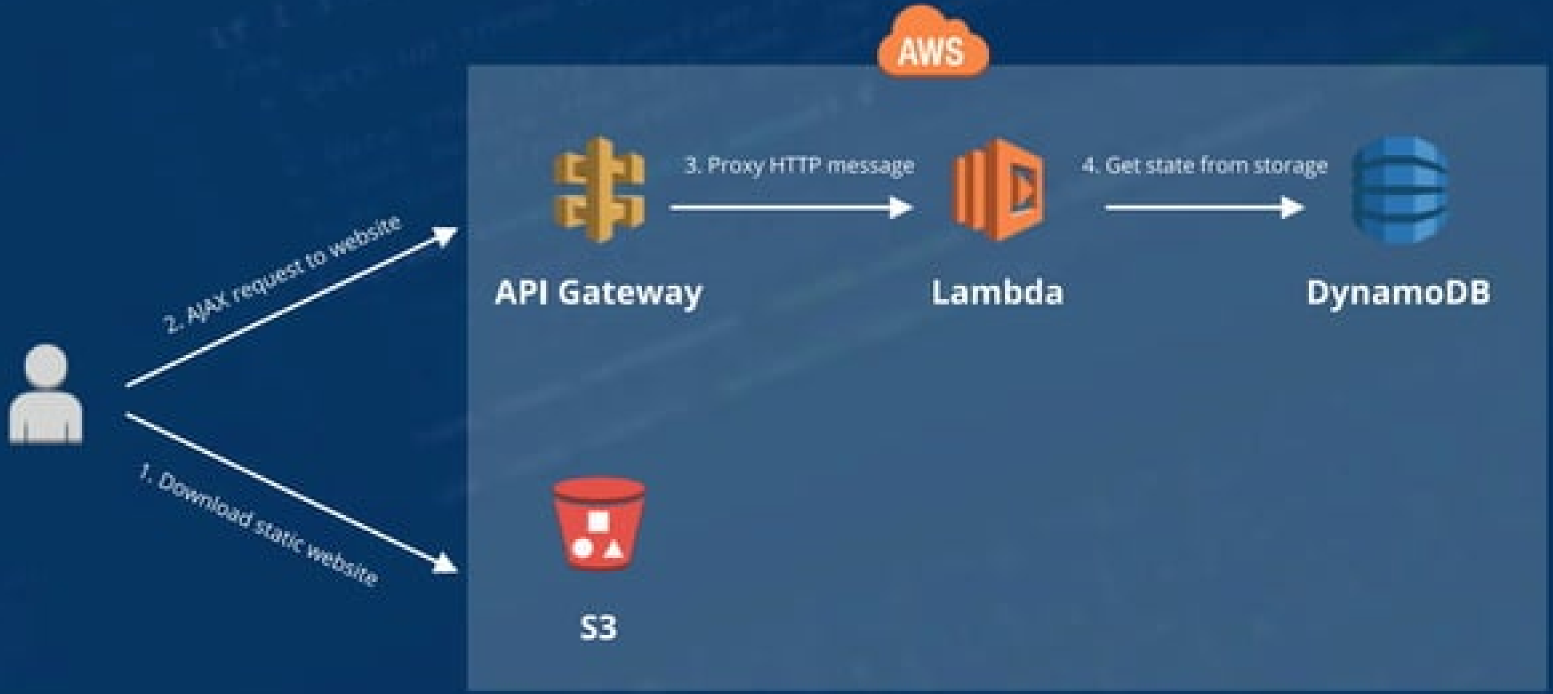
Extend AWS services with custom logic

Drawbacks

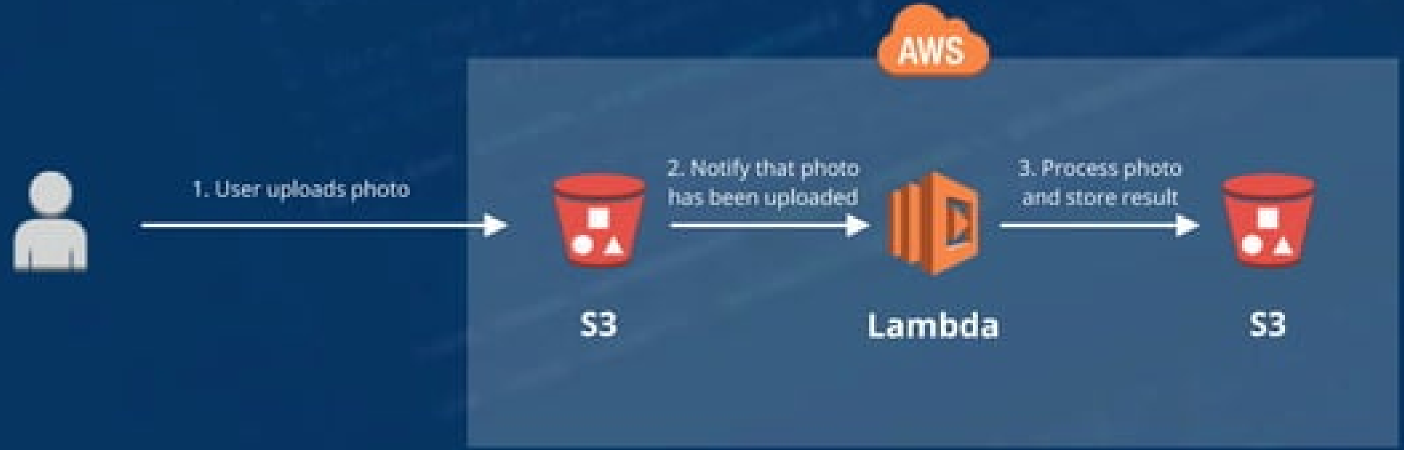
1. Lock into AWS ecosystem
2. New tech that is not battled tested throughout history
3. Long running task that cannot be split into smaller subtasks
4. Non customizable execution environment
5. Not ideal for complex computations with high resource requirements

Use Cases

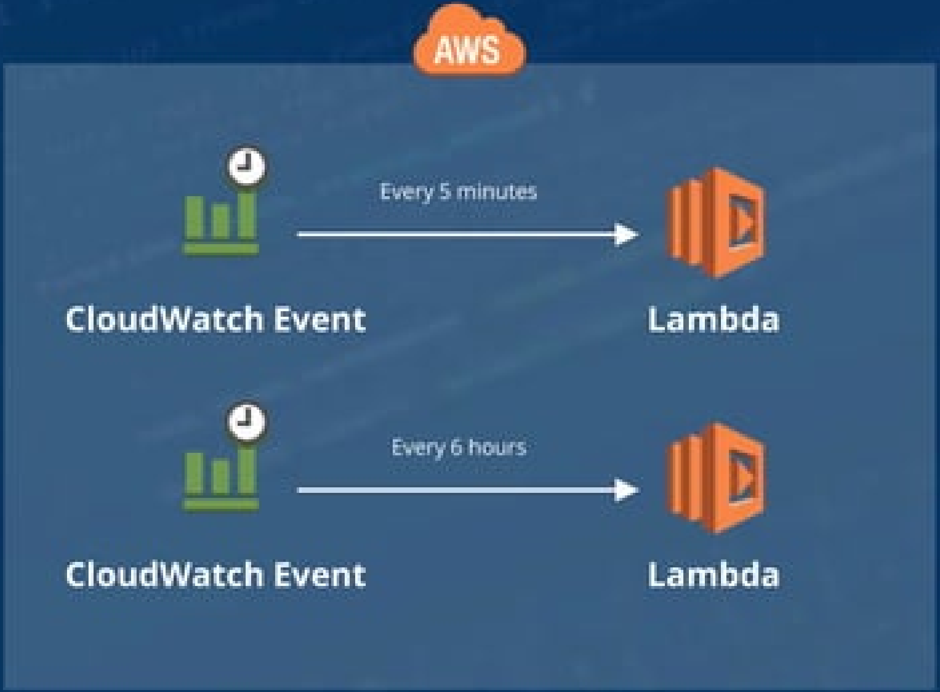
Example 1 - API Backend


AWS
3. Proxy HTTP message
4. Get state from storage
API Gateway
Lambda
DynamoDB
2. AJAX request to website
1. Download static website
S3

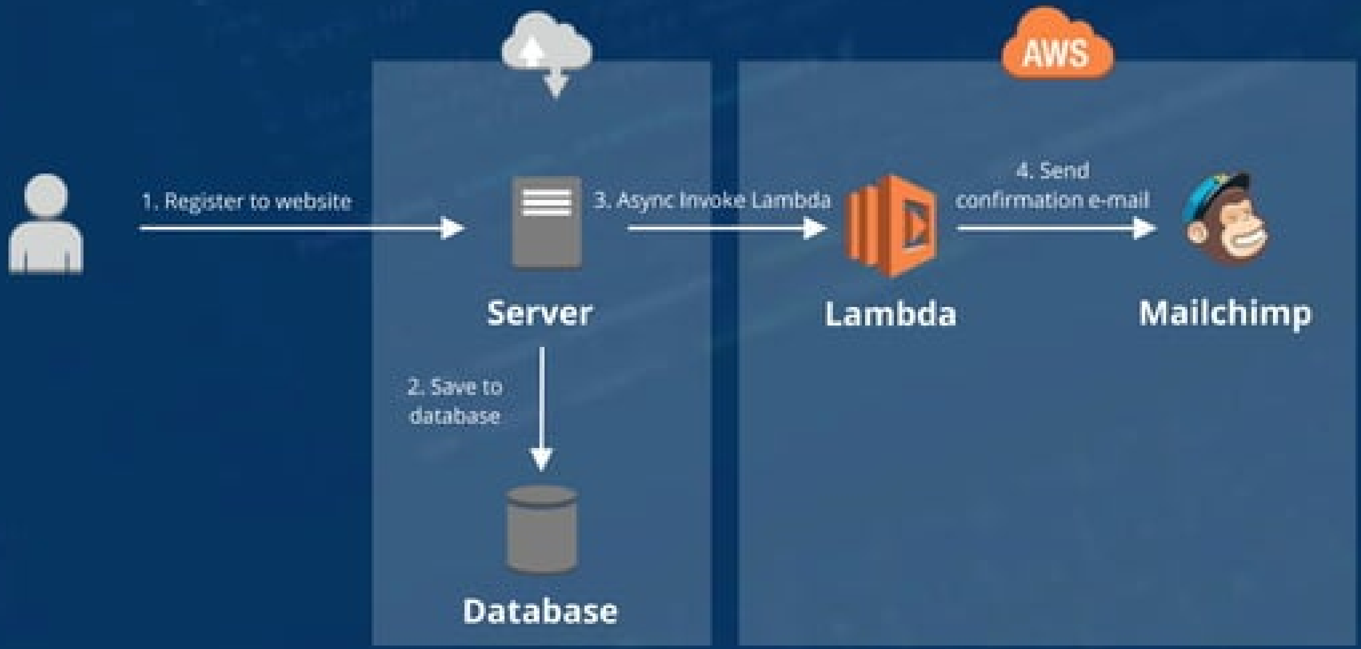
Example 2 - Image Compression


AWS
1. User uploads photo
2. Notify that photo has been uploaded
3. Process photo and store result
S3
Lambda
S3

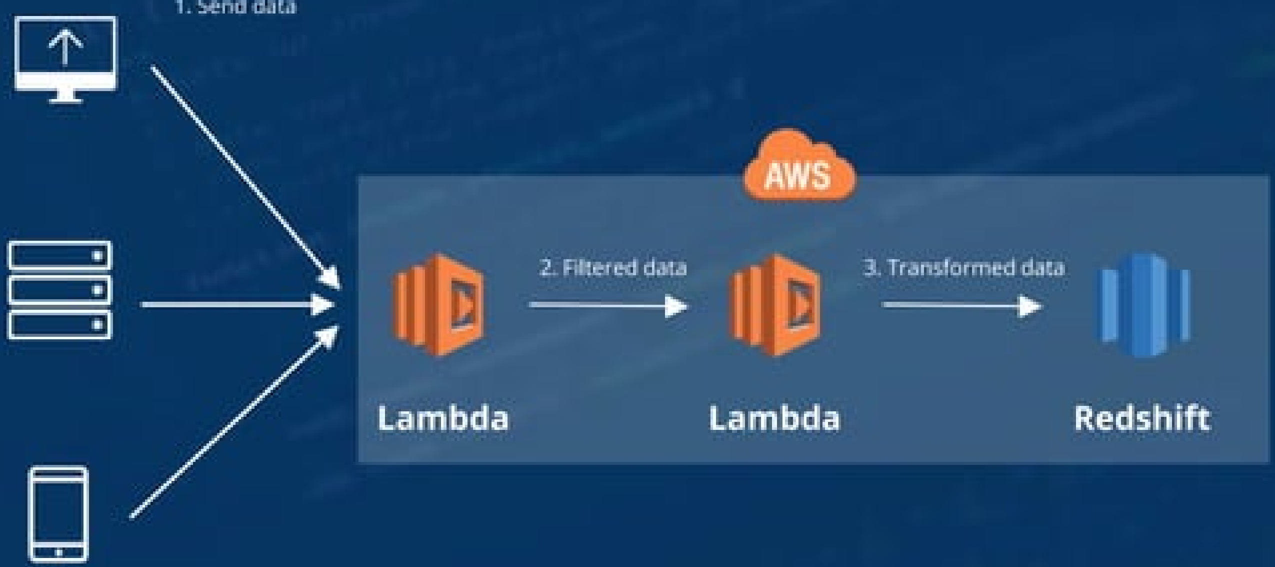
Example 3 - Cron Management



Example 4 - Auxilliary FaaS



Example 5 - Data Pipeline



Useful Information

Case Studies

NETFLIX ConnectWise saturn Aol. Fraud.net BUSTLE

SQUARE ENIX CMP.LY ROYALE International iflix jampp GAMESPARKS

VidRoll Nordstrom NY Summit Localytics OLYMPUSAT VOGUE

ClickTravel Benchling Weather Risk dubsmash InnoVantage

Alternatives

- Google Cloud Functions
- IBM OpenWhisk
- Auth0 webtask.io
- Azure Cloud Functions

CLOUD FUNCTIONS

Create small, single-purpose functions that respond to events in the cloud

APACHE OpenWhisk™

webtask

References

- **AWS Lambda documentation**

  http://docs.aws.amazon.com/lambda/latest/dg/welcome.html?shortFooter=true

- **Awesome curated list of server less resources**

  https://github.com/anaibol/awesome-serverless

- **Another list of awesome server less resources**

  https://github.com/ServerlessHeroes/serverless-resources

  https://www.reddit.com/r/serverless/

  https://martinfowler.com/articles/serverless.html

Questions ?

“PYTHON DEMO”

George Zografos
akis@adaplo.com

Software Engineer @ adaplo

amazon web services