AWS Lambda

workshop

OBJECTIVES OF WORKSHOP

- Be familiar with AWS Lambda
- Be aware of lambda advantage and disadvantage

AGENDA

1. What’s AWS Lambda
2. How it works
3. Hello World & S3EventProcessor
4. Pros & Cons
5. How p10 team uses AWS Lambda
6. Homework

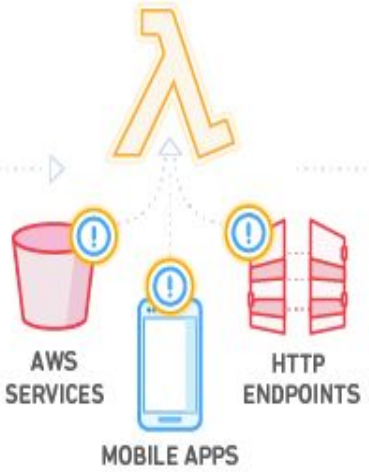
What's AWS LAMBDA

Compute service where you can upload code and run using AWS infrastructure.

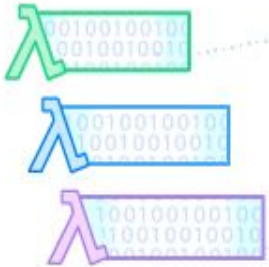
Lambda takes care of provisioning and managing the resources that use to run the code.

How it works

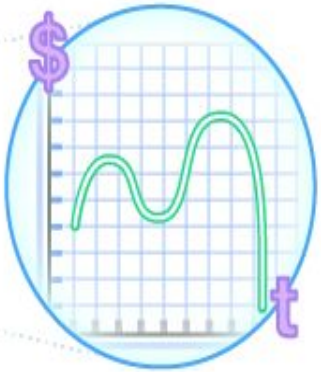
Upload your code to AWS Lambda



Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity



Lambda runs your code only when triggered, using only the compute resources needed



Pay just for the compute time you use

How it works

Lambda can work as follows:

- event-driven compute service
- compute service to run code in response to HTTP request

Lambda executes code only when needed and scales automatically from a few requests per day to thousands per seconds.

Event source

- Amazon S3
- Amazon DynamoDB
- Amazon Kinesis
- Amazon Simple Notification
- Amazon Cognito
- Amazon CloudWatch Logs
- Amazon CloudFormation

AWS RUNTIME VERSIONS

- Node.js v0.10.36
- Java 8
- Python 2.7

Hello World

```java
package recsys.workshop.lambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.LambdaLogger;

public class HelloWorld {

   public String myHandler(int myCount, Context context) {
       LambdaLogger logger = context.getLogger();
       logger.log("received : " + myCount + " \n");
       return String.valueOf(myCount);
   }
}
```

`example.Hello::myHandler - method reference`

S3 EVENT

```
public class S3EventProcessor implements RequestHandler<S3Event, String> {
   @Override
   public String handleRequest(S3Event s3event, Context context) {
       S3EventNotificationRecord record = s3event.getRecords().get(0);
       LambdaLogger logger = context.getLogger();
       String srcBucket = record.getS3().getBucket().getName();
       // Object key may have spaces or unicode non-ASCII characters.
       String srcPath = record.getS3().getObject().getKey().replace('+', ' ');
       try {
           srcPath = URLDecoder.decode(srcPath, "UTF-8");
           logger.log("Bucket: " + srcBucket);
           logger.log("Path: " + srcPath);
           return null;
       } catch (UnsupportedEncodingException e) {
           throw new RuntimeException(e);
       }
   }
}
```

Limits

Ephemeral disk capacity ("/tmp" space) **512 MB**

Number of file descriptors **1,024**

Number of processes and threads (combined total) **1,024**

Maximum execution duration per request **300 seconds**

Invoke request body payload size **6 MB**

Invoke response body payload size **6 MB**

Pros ++++

- autoscaling
- auto management
- easy for development
- integrated with many AWS tools
- push architecture
- error handling - auto retry (default 3x)

CONS ------

- support only java & node & python
- automatic deployment
- security(lack of vpc)
- pricing
- logging & debugging

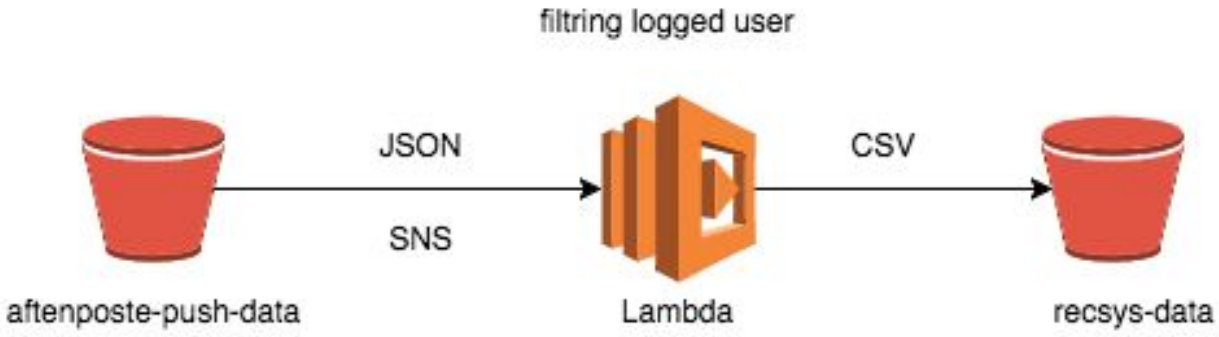
Best PRACTICES

- Stateless style
  - previous run
  - infrastructure
- Minimizing the use of ‘startup’ code not directly related to BL
- use CloudWatch for monitoring Lambda function
- delete unused Lambda functions

What's new

- Python support
- Versioning
- Scheduler
- Increase execution time to five minutes
- Sample event configuration

SPT-PION - LAMBDA USE CASE



Input path: spt-analytics-push-data/aftenposten/behaviour/2015/09/25/1439856008204-8b27600c-2ca2-4cfd-b43e-58c9291d5d98

Output path: spt-analytics-curate-dev/aftenposten/behaviour/2015/09/25/1439856008204-8b27600c-2ca2-4cfd-b43e-58c9291d5d98

USEFUL LINK

- AWS Lambda docs *http://docs.aws.amazon.com/lambda/latest/dg/welcome.html*
- AWS Lambda Blog *https://aws.amazon.com/blogs/compute/aws-lambda-sessions-at-reinvent-2015/*

HOMEWORK

Ex.1 Implement Lambda which prints file content. Lambda should be triggered by the PUT action on S3.