



Apex를 사용한 Lambda function 관리 및 배포

MooYeol Prescott Lee
Software Engineer at Law&Company



발표자 소개

- 이무열 (@mooyoul at Github)
- Software Engineer at Law&Company
- Member of Apex organization

Apex

Apex lets you **build, deploy, and manage AWS Lambda functions with ease**. With Apex you can use languages that are not natively supported by AWS Lambda, such as **Golang**, through the use of a Node.js shim injected into the build. A variety of workflow related tooling is provided for testing functions, **rolling back deploys, viewing metrics, tailing**

설치

설치

```
curl https://raw.githubusercontent.com/  
apex/apex/master/install.sh | sudo sh
```

설치

```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
[Prescott@Prescottui-Macbook-Pro /p/tmp]> curl https://raw.githubusercontent.com/apex/apex/master/install.sh | sudo sh
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload   Total Spent   Left  Speed
100 1646 100 1646    0     0  3668      0 --:--:-- --:--:-- --:--:--  3674
[Prescott@Prescottui-Macbook-Pro /p/tmp]> apex

Usage:
  apex [command]

Available Commands:
  build      Build a function
  delete     Delete functions
  deploy     Deploy functions and config
  docs       Output documentation
  infra      Infrastructure management
  init       Initialize a project
  invoke     Invoke functions
  list       Output functions list
  logs       Output function logs
  metrics    Output function metrics
  rollback   Rollback functions
  upgrade   Upgrade apex to the latest stable release
  version   Print version of Apex

Flags:
  -C, --chdir string      Working directory
  -D, --dry-run           Perform a dry-run
  -e, --env string        Infrastructure environment (default: "defaultEnvironment" from project.json)
  -h, --help              help for apex
  -l, --log-level string Log severity level (default "info")
  -p, --profile string   AWS profile
  -r, --region string    AWS region

Use "apex [command] --help" for more information about a command.
[Prescott@Prescottui-Macbook-Pro /p/tmp]> apex version
Apex version 0.10.2
[Prescott@Prescottui-Macbook-Pro /p/tmp]>
```



참 쉽죠?

AWS credentials

- Via environment variables
 - AWS_ACCESS_KEY_ID
 - AWS_SECRET_ACCESS_KEY
 - AWS_REGION
- Via `~/.aws` files
 - `~/.aws/credentials`
 - `~/.aws/config`
- Via profile flag
 - (profile `myapp-prod` configured on `~/.aws/config`)
 - `$ apex --profile myapp-prod deploy`
- Via project configuration

```
{  
  "profile": "myapp-prod"  
}
```

AWS credentials

AWS CLI가 구성되어 있다면 바로 사용할 수 있다!

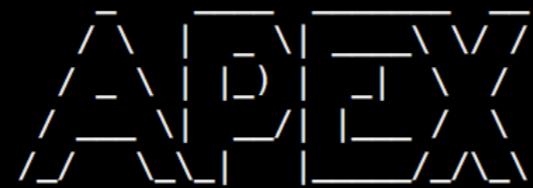
빠르게 시작하기

프로젝트 생성

- \$ apex init
- 새로운 Apex 프로젝트를 만듬
 - IAM lambda function role 생성
 - IAM lambda logs policy 생성 및 적용
 - 프로젝트 default configuration file (project.json) 생성
 - 프로젝트 스캐폴딩

프로젝트 생성

```
[Prescott@Prescottui-Macbook-Pro /p/tmp]> mkdir awsseminar  
[Prescott@Prescottui-Macbook-Pro /p/tmp]> cd awsseminar/  
[Prescott@Prescottui-Macbook-Pro /p/t/awsseminar]> apex init
```



Enter the name of your project. It should be machine-friendly, as this is used to prefix your functions in Lambda.

Project name: awsseminar

Enter an optional description of your project.

Project description: An example project of AWSKRUG seminar

```
[+] creating IAM awsseminar_lambda_function role  
[+] creating IAM awsseminar_lambda_logs policy  
[+] attaching policy to lambda_function role.  
[+] creating ./project.json  
[+] creating ./functions
```

Setup complete, deploy those functions!

```
$ apex deploy
```

```
Prescott@Prescottui-Macbook-Pro /p/t/awsseminar>
```

배포

- \$ apex deploy
현재 프로젝트 내 모든 람다 함수 배포
- \$ apex deploy foo
현재 프로젝트 내 foo 이름을 가진 함수 배포
- \$ apex deploy foo bar
현재 프로젝트 내 foo 혹은 bar 이름을 가진 함수 배포
- \$ apex deploy awsome-*
현재 프로젝트 내 awesome-* 으로 매치되는 함수 배포
- \$ apex deploy foo —alias beta
현재 프로젝트 내 foo 함수의 별명을 beta로 설정하여 배포

호출

- \$ apex invoke foo
현재 프로젝트 내 foo 이름을 가진 함수 실행
- (당연하지만) 함수가 배포 된 상태여야 함 (Local Invoke가 안되므로)

호출

```
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar] apex deploy
  • creating function           function=hello
  • created alias current     function=hello version=1
  • function created          function=hello name=awskrug-seminar_hello version=1
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar] apex invoke hello
{"hello":"world"}
Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar]
```

풀백

우리가 한 것들

- 프로젝트 생성 (apex init)
 - IAM Role/Policy create/attach
 - Project scaffolding
- 함수 배포 (apex deploy)
 - Archive each functions into ZIP Archive
 - Upload built function zip archive to AWS
- 함수 호출 (apex invoke)

롤백

- \$ apex rollback
현재 프로젝트 내 모든 람다 함수를 이전버전으로 롤백
- \$ apex rollback foo
현재 프로젝트 내 foo 이름을 가진 함수를 이전 버전으로 롤백
- \$ apex rollback foo -v 3
현재 프로젝트 내 foo 이름을 가진 함수를 버전 3으로 롤백
- \$ apex rollback awsome-*
현재 프로젝트 내 awesome-* 으로 매치되는 함수를 이전 버전으로 롤백
- deploy command와 거의 비슷함

로백

```
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex list
  hello
    runtime: nodejs4.3
    memory: 128mb
    timeout: 5s
    role: arn:aws:iam::586568971265:role/awskrug-seminar_lambda_function
    handler: index.handle
    current version: 1

[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar>
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar>
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> vim functions/hello/index.js
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar>
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> cat functions/hello/index.js
  console.log('starting function')
  exports.handle = function(e, ctx, cb) {
    console.log('processing event: %j', e)
    cb(null, { hello: 'awskrug' })
  }
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex deploy
  • config unchanged          function=hello
  • updating function         function=hello
  • updated alias current    function=hello version=2
  • function updated          function=hello name=awskrug-seminar_hello version=2
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex invoke hello
  {"hello": "awskrug"}
```

version 2로 업데이트 후

롤백

```
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex list
hello
  runtime: nodejs4.3
  memory: 128mb
  timeout: 5s
  role: arn:aws:iam::586568971265:role/awskrug-seminar_lambda_function
  handler: index.handle
  current version: 2

[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex invoke hello
{"hello":"awskrug"}
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex rollback
  • rolling back          function=hello
  • rollback to version: 1 function=hello
  • function rolled back  current version=1 function=hello
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex invoke hello
{"hello":"world"}
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex rollback -v 2
  • rolling back          function=hello
  • rollback to version: 2 function=hello
  • function rolled back  current version=2 function=hello
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar> apex invoke hello
{"hello":"awskrug"}
Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar>
```

전체 함수 롤백, 호출,
다시 새로 배포했던 버전 (2) 으로 롤백, 호출

로깅

로깅

- apex logs
전체 함수에 대한 로그 출력
- apex logs -f
전체 함수에 대한 로그 출력 및 스트리밍 (tail -F 옵션과 비슷!)
- apex logs foo
foo 함수에 대한 로그 출력
- apex logs foo -f
foo 함수에 대한 로그 출력 및 스트리밍
- apex logs foo —since 5m
5분 전 시점부터 현재까지 foo 함수에 대한 로그 출력

로깅

```
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar] apex logs hello --since 1h
/aws/lambda/awskrug-seminar_hello 2016-09-21T08:56:45.724Z undefined starting function
/aws/lambda/awskrug-seminar_hello START RequestId: 514d74b0-7fd9-11e6-9dda-bdf812005149 Version: 1
/aws/lambda/awskrug-seminar_hello 2016-09-21T08:56:45.728Z 514d74b0-7fd9-11e6-9dda-bdf812005149 processing event: {}
/aws/lambda/awskrug-seminar_hello END RequestId: 514d74b0-7fd9-11e6-9dda-bdf812005149
/aws/lambda/awskrug-seminar_hello REPORT RequestId: 514d74b0-7fd9-11e6-9dda-bdf812005149 Duration: 3.91 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 34 MB
/aws/lambda/awskrug-seminar_hello 2016-09-21T09:05:45.292Z undefined starting function
/aws/lambda/awskrug-seminar_hello START RequestId: 9470ff06-7fda-11e6-bcb2-e90989e7057a Version: 2
/aws/lambda/awskrug-seminar_hello 2016-09-21T09:05:45.307Z 9470ff06-7fda-11e6-bcb2-e90989e7057a processing event: {}
/aws/lambda/awskrug-seminar_hello END RequestId: 9470ff06-7fda-11e6-bcb2-e90989e7057a
/aws/lambda/awskrug-seminar_hello REPORT RequestId: 9470ff06-7fda-11e6-bcb2-e90989e7057a Duration: 14.61 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 7 MB
/aws/lambda/awskrug-seminar_hello START RequestId: 4dd2e149-7fdb-11e6-88a3-e789502b6d48 Version: 1
/aws/lambda/awskrug-seminar_hello 2016-09-21T09:10:56.247Z 4dd2e149-7fdb-11e6-88a3-e789502b6d48 processing event: {}
/aws/lambda/awskrug-seminar_hello END RequestId: 4dd2e149-7fdb-11e6-88a3-e789502b6d48
/aws/lambda/awskrug-seminar_hello REPORT RequestId: 4dd2e149-7fdb-11e6-88a3-e789502b6d48 Duration: 14.20 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 34 MB
/aws/lambda/awskrug-seminar_hello START RequestId: 600d0489-7fdb-11e6-b0e9-f3f9f5218dc6 Version: 2
/aws/lambda/awskrug-seminar_hello 2016-09-21T09:11:26.767Z 600d0489-7fdb-11e6-b0e9-f3f9f5218dc6 processing event: {}
/aws/lambda/awskrug-seminar_hello END RequestId: 600d0489-7fdb-11e6-b0e9-f3f9f5218dc6
/aws/lambda/awskrug-seminar_hello REPORT RequestId: 600d0489-7fdb-11e6-b0e9-f3f9f5218dc6 Duration: 7.82 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 7 MB
/aws/lambda/awskrug-seminar_hello START RequestId: 646d2aec-7fdb-11e6-ad1b-6debac536a48 Version: 1
/aws/lambda/awskrug-seminar_hello 2016-09-21T09:11:34.105Z 646d2aec-7fdb-11e6-ad1b-6debac536a48 processing event: {}
/aws/lambda/awskrug-seminar_hello END RequestId: 646d2aec-7fdb-11e6-ad1b-6debac536a48
/aws/lambda/awskrug-seminar_hello REPORT RequestId: 646d2aec-7fdb-11e6-ad1b-6debac536a48 Duration: 0.70 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 34 MB
/aws/lambda/awskrug-seminar_hello START RequestId: 67f20b60-7fdb-11e6-b29d-f5323b464027 Version: 2
/aws/lambda/awskrug-seminar_hello 2016-09-21T09:11:39.999Z 67f20b60-7fdb-11e6-b29d-f5323b464027 processing event: {}
/aws/lambda/awskrug-seminar_hello END RequestId: 67f20b60-7fdb-11e6-b29d-f5323b464027
/aws/lambda/awskrug-seminar_hello REPORT RequestId: 67f20b60-7fdb-11e6-b29d-f5323b464027 Duration: 0.81 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 7 MB
[Prescott@Prescottui-Macbook-Pro /p/t/awskrug-seminar]
```

하
우

훅

- Apex에서는 함수에 대한 세가지 Hook을 제공
 - build - ZIP 아카이브를 만들기 전
 - deploy - 함수 배포 전
 - clean - 배포 후

후

- 배포 전 npm script로 등록된 build script 실행

The screenshot shows a code editor with two tabs open:

- analysis/package.json**: This tab displays the main project configuration. It includes fields for name, version, description, main file, scripts (with build, clean, lint, and test commands), repository, author, license, bugs, homepage, and dependencies.
- analysis/function.json**: This tab displays configuration for a function. It includes environment variables (DEBUG set to analysis:*) and hooks for build and clean operations, along with memory and timeout settings.

```
analysis/package.json
{
  "name": "train",
  "version": "1.0.0",
  "description": "OpenQna keyword classification analyser",
  "main": "lib/index.js",
  "scripts": {
    "build": "babel -d lib src",
    "clean": "rm -rf lib",
    "lint": "eslint src",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/lawcompany/lawtalk-ml.git"
  },
  "author": "MooYeol Prescott Lee <mooyoul@gmail.com>",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/lawcompany/lawtalk-ml/issues"
  },
  "homepage": "https://github.com/lawcompany/lawtalk-ml#readme",
  "dependencies": {}
}

analysis/function.json
{
  "environment": {
    "DEBUG": "analysis:*
```

Resources

- Apex Official Website
<https://apex.run>
- Apex Github Repository
<https://github.com/apex/apex>
- Apex Slack
<https://apex-dev.azurewebsites.net/>

Common Issues

Q&A

감사합니다.