

Lab3. Lambda API로 함수 실행하기

목적

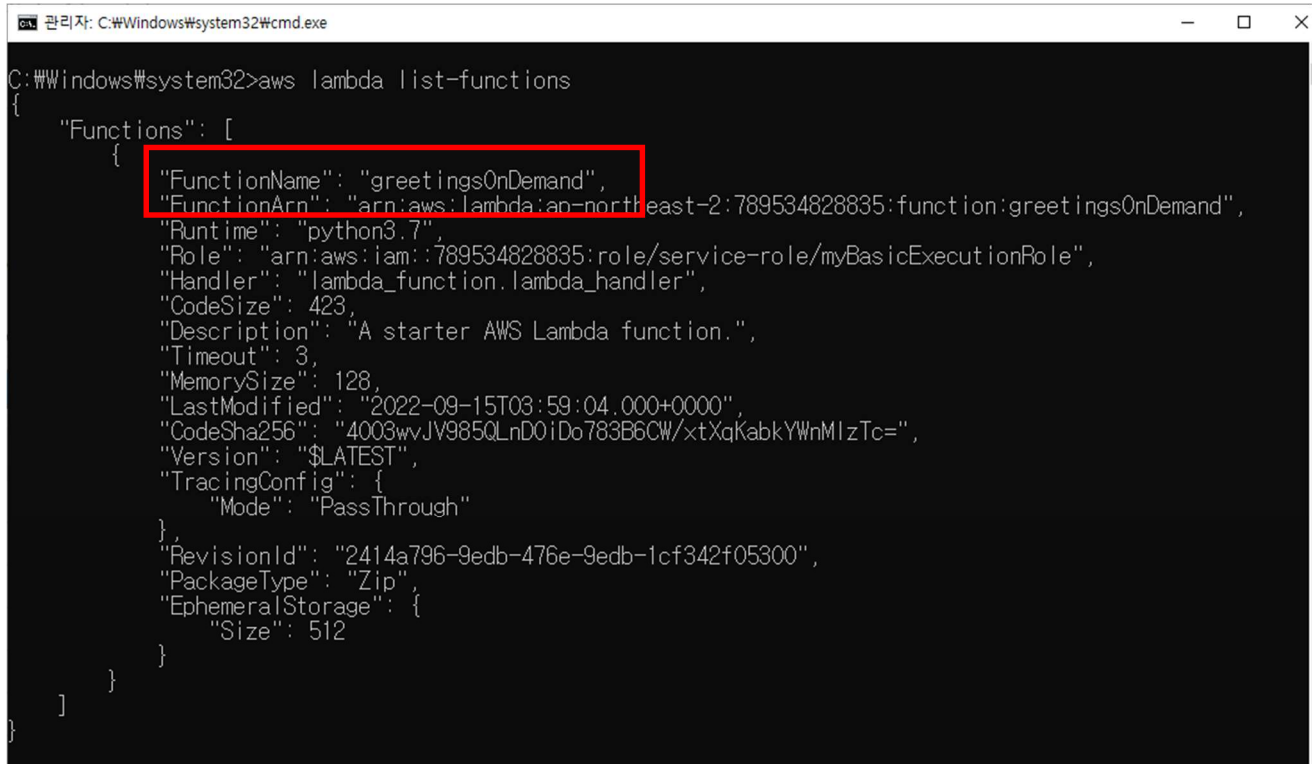
AWS Lambda Invoke API 호출을 통한 Lambda 함수 실행하기

사전 준비물

AWS Free-Tier 계정

1. **AWS CLI**가 정상적으로 설치되고 설정되었는지 확인해 보자. 다음의 명령을 통해 확인할 수 있다.

```
$ aws lambda list-functions
```



```
C:\Windows\system32>aws lambda list-functions
{
  "Functions": [
    {
      "FunctionName": "greetingsOnDemand",
      "FunctionArn": "arn:aws:lambda:ap-northeast-2:789534828835:function:greetingsOnDemand",
      "Runtime": "python3.7",
      "Role": "arn:aws:iam::789534828835:role/service-role/myBasicExecutionRole",
      "Handler": "lambda_function.lambda_handler",
      "CodeSize": 423,
      "Description": "A starter AWS Lambda function.",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2022-09-15T03:59:04.000+0000",
      "CodeSha256": "4003wvJV985QLnD0iDo783B6CW/xtXqKabkYWnMlztC=",
      "Version": "$LATEST",
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "2414a796-9edb-476e-9edb-1cf342f05300",
      "PackageType": "Zip",
      "EphemeralStorage": {
        "Size": 512
      }
    }
  ]
}
```

2. 위의 결과에서 확인할 수 있듯이 앞 Lab에서 생성했던 **greetingsOnDemand** 함수를 확인할 수 있고, 또한 Memory Size나 Timeout, Role을 확인할 수 있다.
3. 함수를 호출해보자.

```
$ aws lambda invoke --function-name greetingsOnDemand --payload '{"name" : "Sujan"}' output.txt
```



```
C:\Windows\system32>aws lambda invoke --function-name greetingsOnDemand --payload '{"name" : "Sujan"}' output.txt

usage: aws [options] <command> [<subcommand> ...] [parameters]
To see help text, you can run:

    aws help
    aws <command> help
    aws <command> <subcommand> help

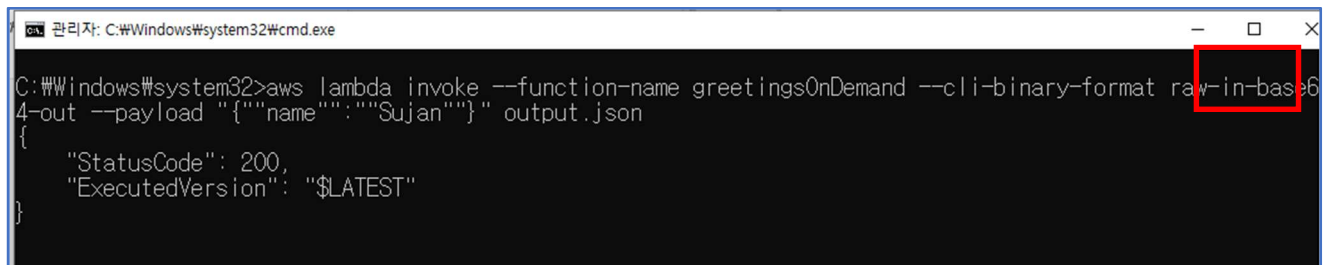
Unknown options: output.txt, Sujan'
```

4. Windows Command 창에서 실행할 때 주의할 점은 작은 따옴표를 사용하면 위의 그림 그리고 아래의 그림과 같이 오류가 발생한다. 또한 Invalid base64 오류는 --payload 파라미터를 사용하여 Lambda 함수 호출 시, 자주 발생하는 오류이다.

```
C:\Windows\system32>aws lambda invoke --function-name greetingsOnDemand --payload '{"name":"Sujan"}' output.txt
Invalid base64: '{"name:Sujan}'
```

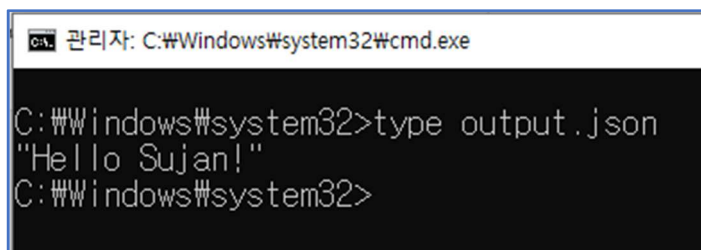
5. 그래서 **Windows Command** 창에서는 `'{"name":"Sujan"}'` 을 `"{"name":"Sujan"}"` 을 사용해야 한다. 또한 Invalid base64 오류를 해결하기 위해 --cli-binary-format 의 값에 raw-in-base64-out 을 추가해야 한다.

```
$ aws lambda invoke --function-name greetingsOnDemand --cli-binary-format raw-in-base64-out --payload '{"name":"Sujan"}' output.json
```



```
C:\Windows\system32>aws lambda invoke --function-name greetingsOnDemand --cli-binary-format raw-in-base64-out --payload '{"name":"Sujan"}' output.json
{
  "StatusCode": 200,
  "ExecutedVersion": "$LATEST"
}
```

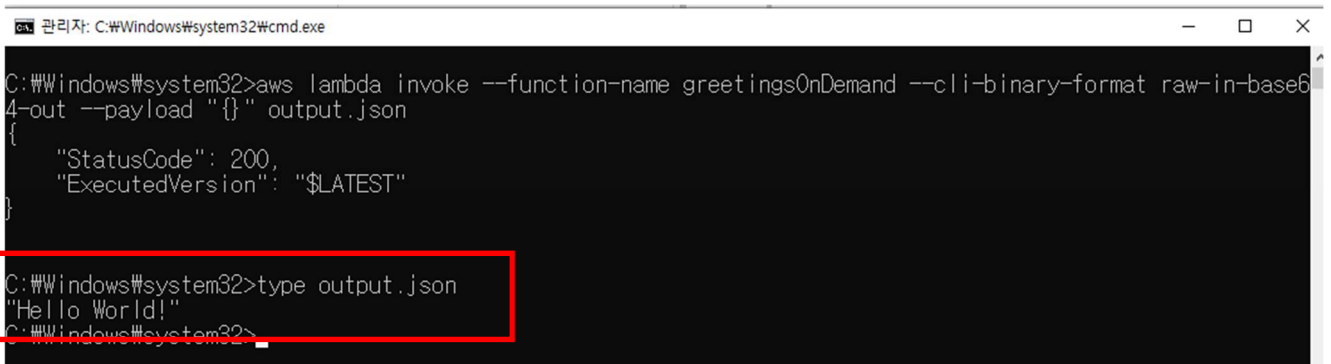
6. 결과를 확인해 보자. **output.json** 파일을 열어보면 다음과 같다.



```
C:\Windows\system32>type output.json
"Hello Sujan!"
C:\Windows\system32>
```

7. 이번에는 이름을 입력하지 않고 실행했다.

```
$ aws lambda invoke --function-name greetingsOnDemand --cli-binary-format raw-in-base64-out --payload "{}" output.json
```



```
C:\Windows\system32>aws lambda invoke --function-name greetingsOnDemand --cli-binary-format raw-in-base64-out --payload "{}" output.json
{
  "StatusCode": 200,
  "ExecutedVersion": "$LATEST"
}
C:\Windows\system32>type output.json
"Hello World!"
C:\Windows\system32>
```