

AWS LocalStack - Lambda

LocalStack AWS

~ .

DataLake ETL .

Jupyter Notebook .

ETL (Extract), (Transform), (Load) .

- Dockercompose

```
version: '3'
services:
  localstack:
    image: localstack/localstack:3.0.2
    ports:
      - "4567:4566"
    environment:
      - SERVICES=s3:4566,iam,lambda
      - HOSTNAME_EXTERNAL=localstack
      - AWS_ACCESS_KEY_ID=test
      - AWS_SECRET_ACCESS_KEY=test
      - AWS_DEFAULT_REGION=us-east-1
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock"
    labels:
      io.rancher.scheduler.affinity:host_label: server=lake01
      io.rancher.container.hostname_override: container_name
```

(WSL2) .

AWS TestKey

```
aws configure
- AWS_ACCESS_KEY_ID=test
- AWS_SECRET_ACCESS_KEY=test
- AWS_DEFAULT_REGION=us-east-1
```

AWS CLI .

endpoint AWS Cloud Stack .

- --endpoint-url=http://localhost:4567

IAM Role

AWS , RoleBase .

Function .

```
aws iam create-role --role-name lambda-ex --assume-role-policy-document '{"Version": "2012-10-17","Statement":
[{"Effect": "Allow", "Principal": {"Service": "lambda.amazonaws.com"}, "Action": "sts:AssumeRole"}]}' --
endpoint-url=http://localhost:4567

aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role
/AWSLambdaBasicExecutionRole --endpoint-url=http://localhost:4567
```

node.js .

index.js -

```
exports.handler = async function(event, context) {
  console.log("ENVIRONMENT VARIABLES\n" + JSON.stringify(process.env, null, 2))
  console.log("EVENT\n" + JSON.stringify(event, null, 2))
  return context.logStreamName
}
```

OpenStack .

: https://docs.aws.amazon.com/ko_kr/lambda/latest/dg/python-handler.html

() , .

```
zip function.zip index.js

aws lambda create-function --function-name my-function \
--zip-file fileb://function.zip --handler index.handler --runtime nodejs20.x \
--role arn:aws:iam::123456789012:role/lambda-ex --endpoint-url=http://localhost:4567
```

```
aws lambda invoke --function-name my-function out --log-type Tail --endpoint-url=http://localhost:4567 \
--query 'LogResult' --output text | base64 -d
```

~

vscode code-server .

.

```

version: "2"
services:
  code-server:
    image: lscr.io/linuxserver/code-server:latest
    container_name: code-server
    # stop
    restart: always
    ports:
      - "8443:8443"
    environment:
      # Container
      TZ: Asia/Seoul
      # code-space
      PASSWORD: webnoridev
      # Workspace (Container )
      DEFAULT_WORKSPACE: /config/workspace
    volumes:
      # VSCode ->
      - codespace-config:/config
      # Workspace Host
      - /opt/vscode:/config/workspace
    labels:
      io.rancher.scheduler.affinity:host_label: server=home01
      io.rancher.container.hostname_override: container_name

volumes:
  codespace-config:

```

OpenStack Ec2 AWS Stack

EC2 Vagrant VM .

- IDC(10kg) > VM(Vagrant) / AWS EC2 (OS 1GB) > DockerContainer(200mb) > / AWS Lambda(100kb)