**Run Scrapy command from AWS Lambda function using Serverless Framework**

**Pre-requisite**:   
  
(Skip if you already have *AWS CLI, Nodejs, npm, Serverless and Docker* installed)

1. Create Access and Secret Access Keys:
   1. For new IAM user:
      1. Login to your AWS account and search for IAM service on the service search filter.
      2. Go to the Users page and click on “Create user”
      3. Now give a user name and click Next.
      4. Now we need to set up permissions for this user. Select on “Attach policies directly”, and provide “AdministratorAccess” to the user. This is going to provide this user full access to your AWS account, you can change if you want. And click Next.
      5. Then on the next page it will ask you to add tags (if required) and click “Create user”.
      6. Once the user is created, follow below steps under (b)
   2. Already have IAM user:
      1. Go to User -> Security Credentials -> Under “Access Keys” click on “Create access key”
      2. Select “Command Line Interface (CLI)” and tick the Confirmation box and click Next.
      3. Type description if required and click “Create access key”.
      4. Click on “Download .csv file” and then click Done. Make sure not to share this file with anyone.
2. Installing AWS CLI on your system
   1. Install [AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html#getting-started-install-instructions) on your system.
   2. After installation go to the command prompt to review your installation type “aws help”
   3. Type “aws configure” and use the Access and Secret Access Keys from Step 1(b)(iv).
3. Setup NodeJS
   1. To install NodeJs go to [nodeJs website](https://nodejs.org/en) and download/install.
   2. Review your installation by checking the node version by typing “node --version” in CMD prompt
4. Installing and configuring AWS Serverless framework
   1. Serverless is provided as npm package. To install serverless open CMD prompt and type “npm install -g serverless”
   2. To review your installation *serverless -v* to check the version
5. Install Docker
   1. Install docker from [Docker website](https://docs.docker.com/engine/install/)
   2. Create account on [docker hub](https://hub.docker.com/) if you do not already have.
   3. Run the docker on local machine and login to Docker hub using above credentials.

**Steps to deploy new application on AWS Lambda using Docker and Serverless Framework**:

1. Create folder named "*MyLambdaFunction*" (or any other name that you feel suitable for this project)
2. cd into that folder from CLI
3. Copy all your application data (after extracting the zip that you shared) into "*MyLambdaFunction*" folder.
4. Create 4 empty files under "*MyLambdaFunction*" directory. The content of all these 4 files is given in step 7 to 10:
   1. Dockerfile
   2. lambda\_function.py
   3. requirements.txt
   4. serverless.yml
5. Create ECR Repository on AWS from [AWS Console](https://us-east-1.console.aws.amazon.com/ecr/) or CLI and name it: “*my-repo-custom*” (or any other name that you feel suitable for this project).
   1. Copy the *URI* that was generated for this ECR repo, we will need it later.
6. **NOTE**: You need to have [docker](https://docs.docker.com/engine/install/) (up and running), [Serverless](https://www.serverless.com/framework/docs/getting-started) and [AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html) installed and configured. Please follow the pre-requisite section that has 5 steps at the start of this documentation. Proceed only if pre-requisite is fulfilled.
7. Edit Dockerfile use below content and save the file. Kindly note that Dockerfile should not have any extension.

FROM public.ecr.aws/lambda/python:3.8

# Required for lxml

RUN yum install -y gcc libxml2-devel libxslt-devel

COPY . ${LAMBDA\_TASK\_ROOT}

RUN pip install --upgrade pip

RUN pip3.8 install -r requirements.txt

WORKDIR ${LAMBDA\_TASK\_ROOT}/newsdata/spiders

CMD [ "lambda\_function.handler" ]

1. Edit lambda\_function.py and use below content and save the file:

import sys

import subprocess

def handler(event, context):

# Run the Scrapy spider

print(event)

print("Scrapy running now")

subprocess.run(["scrapy", "crawl", "fooddive"])

subprocess.run(["scrapy", "crawl", "news"])

print("Scrapy ran successfully")

return {

'statusCode': '200', # a valid HTTP status code

'body': 'Lambda function invoked',

}

1. Edit requirements.txt and use below content and save the file.

attrs==23.1.0

Automat==22.10.0

backports.zoneinfo==0.2.1

boto3==1.28.73

botocore==1.31.73

certifi==2023.7.22

cffi==1.16.0

charset-normalizer==3.3.1

constantly==23.10.4

cryptography==41.0.5

cssselect==1.2.0

dateparser==1.1.8

filelock==3.13.0

hyperlink==21.0.0

idna==3.4

incremental==22.10.0

itemadapter==0.8.0

itemloaders==1.1.0

jmespath==1.0.1

loguru==0.7.2

lxml==4.9.3

mysql-connector-python==8.2.0

packaging==23.2

parsel==1.8.1

Protego==0.3.0

protobuf==4.21.12

pyasn1==0.5.0

pyasn1-modules==0.3.0

pycparser==2.21

PyDispatcher==2.0.7

pyOpenSSL==23.3.0

python-dateutil==2.8.2

pytz==2023.3.post1

queuelib==1.6.2

regex==2023.10.3

requests==2.31.0

requests-file==1.5.1

s3transfer==0.7.0

Scrapy==2.11.0

service-identity==23.1.0

six==1.16.0

tldextract==5.0.1

Twisted==22.10.0

typing\_extensions==4.8.0

tzlocal==5.2

urllib3==1.26.18

w3lib==2.1.2

zope.interface==6.1

1. Edit serverless.yml and use below content and save the file:

service: scrapy-lambda

provider:

name: aws

runtime: python3.8

stage: dev

***region***: *ap-south-1*

functions:

scrapyFunction-py38:

***image***: *<account-id>.dkr.ecr.ap-south-1.amazonaws.com/<repoName>:latest*

timeout: 900

environment:

KEY1: Value1

KEY2: Value2

KEY3: Value3

events:

- eventBridge:

enabled: true

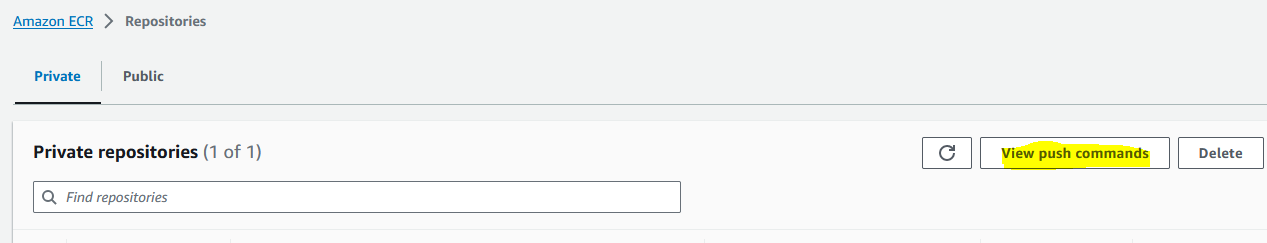
schedule: rate(24 hours)

Kindly replace the ***region*** and ***image*** in above file. The value for ***image*** will be same that was copied in point 5(a). Make sure you add “:latest” at the end of ***image***.

1. Make sure you are under directory “*MyLambdaFunction”* from CLI.
2. Run below command from CLI: (make sure you have [AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html) logged in using “aws configure” command)

aws ecr get-login-password --region ***ap-south-1*** | docker login --username AWS --password-stdin ***<account\_ID>.dkr.ecr.ap-south-1.amazonaws.com/<repoName>***

In above command, please change the **region** as required and use the **URI** from point 5(a)

1. Go to AWS Console for ECR and select the repository and click “*View push commands*”, select the OS that you are using and start from step 2 (building the image) under the commands. Step 1 was performed in previous step. *Please build image (Step 2) using command shared below in case of MacOs with M1 chip.*  
   

The commands contain docker build, docker tag and docker push.

**NOTE**: In case of MACOS, build image using below command

docker buildx build --platform linux/amd64 -f ./Dockerfile -t ***<repo-Name>*** .

Replace the ***<repo-Name>*** with name used in point 5 above, for example in this case, it is “***my-repo-custom***”. Do not miss the “.” at the end of above command.

**NOTE**: Make sure not to change the name from any of these commands unless you know what you are doing.

1. Make sure you are under directory “*MyLambdaFunction”* from CLI and run below command:

serverless deploy --region ap-south-1

1. Wait for the stack to deploy and navigate to AWS Lambda console and test the same.

**Steps to Update**:

1. Update the files as per your requirement (Dockerfile/Serverless.yml/Code etc.) under “*MyLambdaFunction*” directory.
2. Follow steps 11 till 15 from above.