

Lambda CLI:

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running. In Lambda free tier, First 1 million requests per month are free. Even after that it's only \$0.20 per million requests.

AWS Lambda has 2 components. An event triggers the Lambda function. An event can be image being uploaded to S3 or a SNS notification or API gateway call etc.

When the event trigger our Lambda function, the logic or code in the function does something for us. We can write any kind of code. In our case we will stop our running EC2 instances.

So to review, an event triggers a Lambda function. Then the function will run without us having to worry about the servers. Hence it's called server less.

Creating a role for our function:

```
$ aws iam create-role --role-name Test-role-Lambda --assume-role-policy-document file://lambdaTrustPolicy.json
```

**copy the role arn

Find policy with EC2 Access:

```
$ aws iam list-policies --output table | grep EC2Full
```

Copy the arn:

```
arn:aws:iam::aws:policy/AmazonEC2FullAccess
```

Attach the policy to the role:

```
aws iam attach-role-policy --role-name Test-role-Lambda --  
policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
```

Make a file with a python code (stop.py).

**** file will be included in the resource section.**

ZIP the file with command line or with GUI :

```
$ zip -r lambda stop.py
```

Create a lambda function (handler should be filename.handlerfunctionname)

```
aws lambda create-function --function-name CLI-lambda-test --  
runtime python2.7 --role  
arn:aws:iam::77554000355:role/lambda_full --handler  
stop.lambda_handler --zip-file fileb://lambda.zip
```

**** make sure you paste Arn of your role**

Start 3 EC2 instances with this command:

```
aws ec2 run-instances --image-id ami-6057e21a --instance-type  
t2.micro --count 3
```

Invoke the function Asynchronously:

```
aws lambda invoke --invocation-type Event --function-name CLI-  
lambda-test output.txt
```

Alternately invote synchronously with

```
aws lambda invoke --invocation-type RequestResponse --  
function-name CLI-lambda-test output.txt
```

If you need to pass certain values add this to the end of your invoke command:

```
--payload '{"key1":"value1", "key2":"value2", "key3":"value3"}'
```
