



# Creating Lambda Function using CLI

1. In this lab we will create the Lambda function using CLI. For that first, we will create the folder in VS Code then inside the folder we need to create the Commands text file with that we also need to create the index.js file.
2. The code you can see in the index.js file is taken from the lambda function itself. When you create a normal lambda function using node.js then you get this hello lambda code in your function.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a tree view of files and folders. The current folder is "Lab 5 Creating Lambda Funciton using CLI". Inside, there are "commands.txt", "trust-policy.json", and another "index.js" file.
- Editor:** The active tab is "index.js". The code content is as follows:

```
1 exports.handler = async (event) => {
2     console.log("EVENT: \n" + JSON.stringify(event, null, 2))
3
4     // TODO implement
5     const response = {
6         statusCode: 200,
7         body: JSON.stringify('Hello from Lambda!'),
8     };
9     return response;
10 };
```

3. Once we have created our index.js file now we need to make a zip file out of it so that we can upload that zip file while creating our lambda function.
4. First you need to get inside the lab 5 folder then you need to run the command.

The screenshot shows the VS Code terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

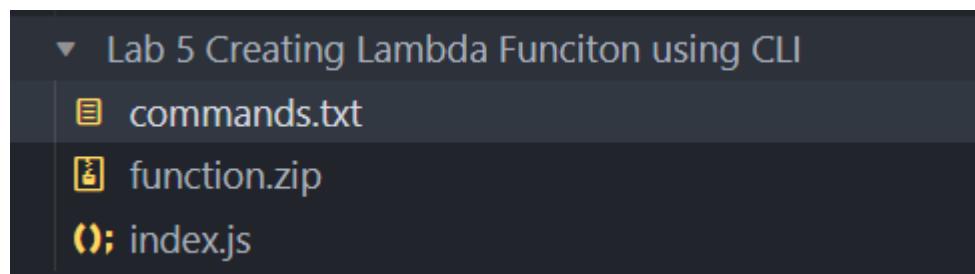
> ▾ TERMINAL
A ● PS C:\Serverless> cd '.\Lab 5 Creating Lambda Funciton using CLI\'
```

5. Below you can see that our command was executed successfully and we have the zip file as expected.

The screenshot shows the VS Code terminal window with the following output:

```
▼ TERMINAL
● PS C:\Serverless> cd '.\Lab 5 Creating Lambda Funciton using CLI\'
```

At the end of the command, there is a prompt for further input, indicated by three greater than signs (>>>).



6. Then we run this command to get the ARN of our role so that we can use this to create our lambda function and attach the role directly to our function.

```
PS C:\Serverless\Lab 5 Creating Lambda Funciton using CLI> aws iam get-role --role-name lambda-exec
Role:
  Arn: arn:aws:iam::878893308172:role/lambda-exec
  AssumeRolePolicyDocument:
    Statement:
      - Action: sts:AssumeRole
        Effect: Allow
        Principal:
```

7. Now we are going to run the create lambda command and our function will get created.
8. Below you can see that our lambda function has been created.

```
aws lambda create-function \
--function-name my-function2 \
--runtime nodejs18.x \
--zip-file fileb://function.zip \
--handler index.handler \
--role arn:aws:iam::878893308172:role/lambda-exec
```

```
Statement:
PS C:\Serverless\Lab 5 Creating Lambda Funciton using CLI> aws lambda create-function
>>   --function-name my-function2 \
>>   --runtime nodejs18.x \
>>   --zip-file fileb://function.zip \
>>   --handler index.handler \
>>   --role arn:aws:iam::878893308172:role/lambda-exec
Architectures:
- x86_64
CodeSha256: nbEnam00xLRnkRUCAoPvhYgnsa80ypGKDpnbZJuxjt4=
CodeSize: 301
Description: ''
EphemeralStorage:
  Size: 512
FunctionArn: arn:aws:lambda:us-east-1:878893308172:function:my-function2
FunctionName: my-function2
Handler: index.handler
LastModified: 2024-11-12T10:20:45.588+0000
LoggingConfig:
  LogFormat: Text
  LogGroup: /aws/lambda/my-function2
-- More --
```

9. If we go to the AWS Console and open our Lambda function page, we will be able to see our new function.

Functions (2)					
Last fetched 0 seconds ago					
Actions ▾ Create function					
<input type="checkbox"/> Filter by tags and attributes or search by keyword					
Function name	Description	Package type	Runtime	Last modified	⋮
<a href="#">myfirst-function</a>	-	Zip	Node.js 20.x	23 hours ago	
<a href="#">my-function2</a>	-	Zip	Node.js 18.x	50 seconds ago	

10. Our function is created successfully now we will invoke our function and we will see the Cloud Watch logs.

11. Now before you invoke the function you have to create the event.json file.

```

FOLDERS: SERVE...  ↗️ 🚫 ⏪ ⏴
commands.txt  ●  event.json Lab 3 Invoke Lambda functions
▶ Lab 1 Using AWS Lambda with ...
▶ Lab 2 Invoke Lambda using CLI
▶ Lab 3 Invoke Lambda functions ...
▶ Lab 4 Create Execution Role
▼ Lab 5 Creating Lambda Funciton using CLI > event.json > ...
  commands.txt
  event.json
  function.zip
  index.js

```

12. Now just run the command to invoke your function and below you can see that our function has been invoked.

```

TERMINAL
PS C:\Serverless\Lab 5 Creating Lambda Funciton using CLI> aws lambda invoke `>>   --function-name my-function2 `>>   --cli-binary-format raw-in-base64-out `>>   --payload file://event.json `>>   --response.json
ExecutedVersion: $LATEST
StatusCode: 200

```

13. Open your function then view the Cloud Watch logs. In the log stream you can see that your logs has been generated for the invocation of your function.

Log events	
You can use the filter bar below to search for and match terms, phrases, or values in your log events. <a href="#">Learn more about filter patterns</a>	
<input type="text"/> Filter events - press enter to search	<a href="#">Clear</a> <a href="#">1m</a> <a href="#">30m</a> <a href="#">1h</a> <a href="#">12h</a> <a href="#">Custom</a> <a href="#">Local timezone</a> <a href="#">Display</a> 
▶ Timestamp	Message
No older events at this moment. <a href="#">Retry</a>	
▶ 2024-11-12T15:55:29.228+05:30	INIT_START Runtime Version: nodejs:18.v49 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:13821268cdb8b1fd3647b6b7f047e...
▶ 2024-11-12T15:55:29.423+05:30	START RequestId: 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef Version: \$LATEST
▶ 2024-11-12T15:55:29.424+05:30	2024-11-12T10:25:29.424Z 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef INFO EVENT: { "key": "value" }
▶ 2024-11-12T15:55:29.452+05:30	END RequestId: 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef
▶ 2024-11-12T15:55:29.452+05:30	REPORT RequestId: 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef Duration: 29.26 ms Billed Duration: 30 ms Memory Size: 128 MB Max Mem...
No newer events at this moment. <a href="#">Auto retry paused</a> . <a href="#">Resume</a>	