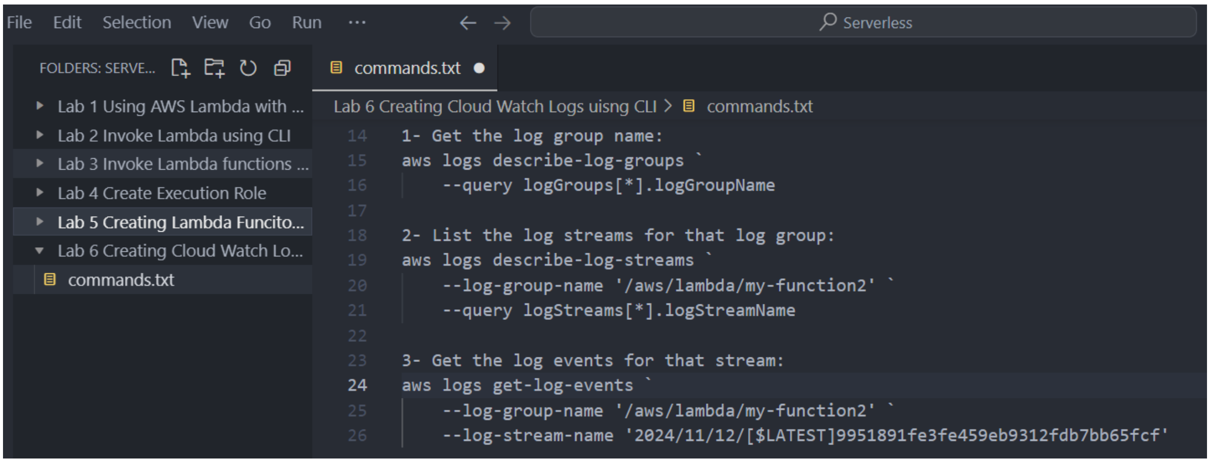
😊 AWS Lambda Cloud Watch logs using CLI

1. In this lab we will get the Cloud Watch logs of our lambda function using AWS CLI.
2. So, first we need to create the lab folder in our VS Code then we need to create the commands text file.



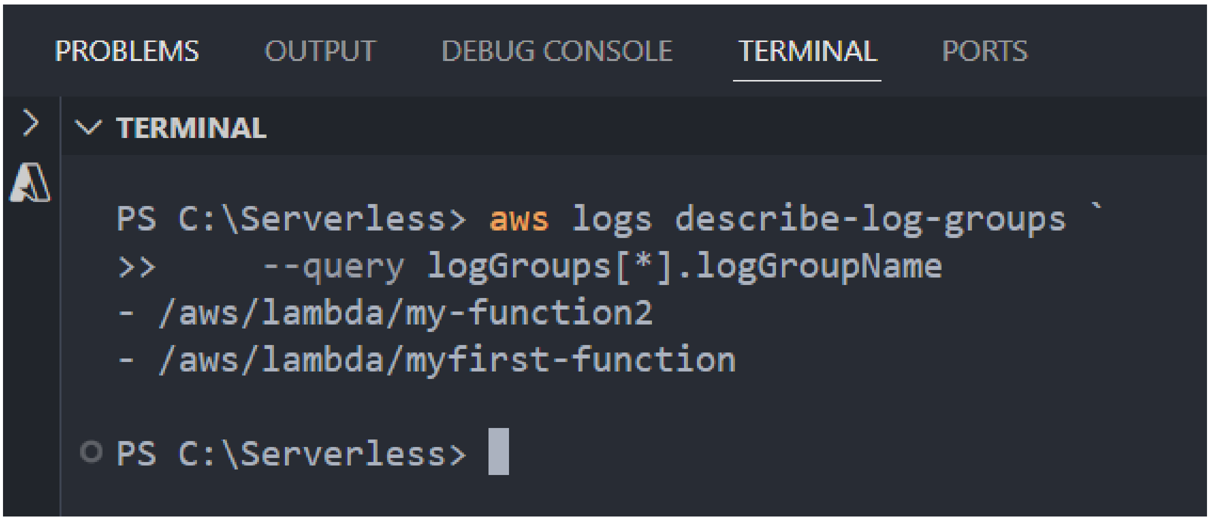
File Edit Selection View Go Run ···
Serverless
FOLDERS: SERVE...
commands.txt ●
Lab 1 Using AWS Lambda with ...
Lab 2 Invoke Lambda using CLI
Lab 3 Invoke Lambda functions ...
Lab 4 Create Execution Role
Lab 5 Creating Lambda Funcito...
Lab 6 Creating Cloud Watch Lo...
commands.txt
Lab 6 Creating Cloud Watch Logs uisng CLI > commands.txt

```
1- Get the log group name:
aws logs describe-log-groups `
    --query logGroups[*].logGroupName

2- List the log streams for that log group:
aws logs describe-log-streams `
    --log-group-name '/aws/lambda/my-function2' `
    --query logStreams[*].logStreamName

3- Get the log events for that stream:
aws logs get-log-events `
    --log-group-name '/aws/lambda/my-function2' `
    --log-stream-name '2024/11/12/[$LATEST]9951891fe3fe459eb9312fdb7bb65fcf'
```

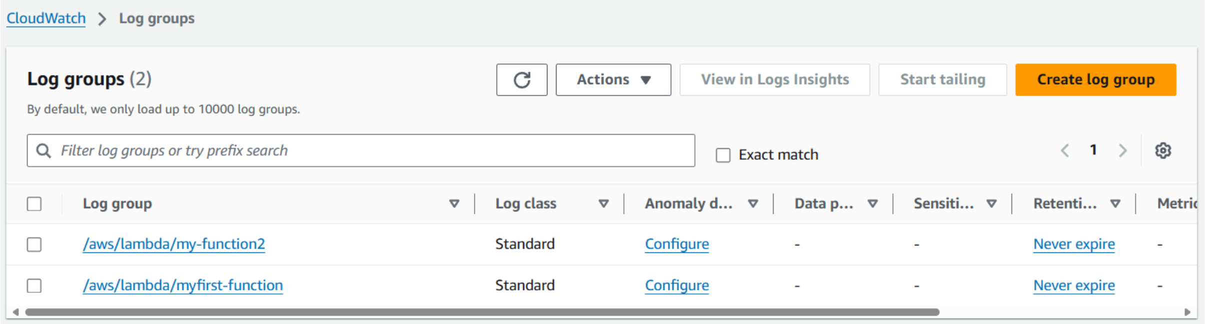
3. Once it is done then we need to run the commands. So, start with the first command. In this command, we are first describing what are the log groups present in the CloudWatch.



```
PS C:\Serverless> aws logs describe-log-groups `
>>      --query logGroups[*].logGroupName
- /aws/lambda/my-function2
- /aws/lambda/myfirst-function

PS C:\Serverless>
```

4. So, if you go to the log groups in the CloudWatch you will see that there are only log groups present which means that the command we ran was perfectly executed.

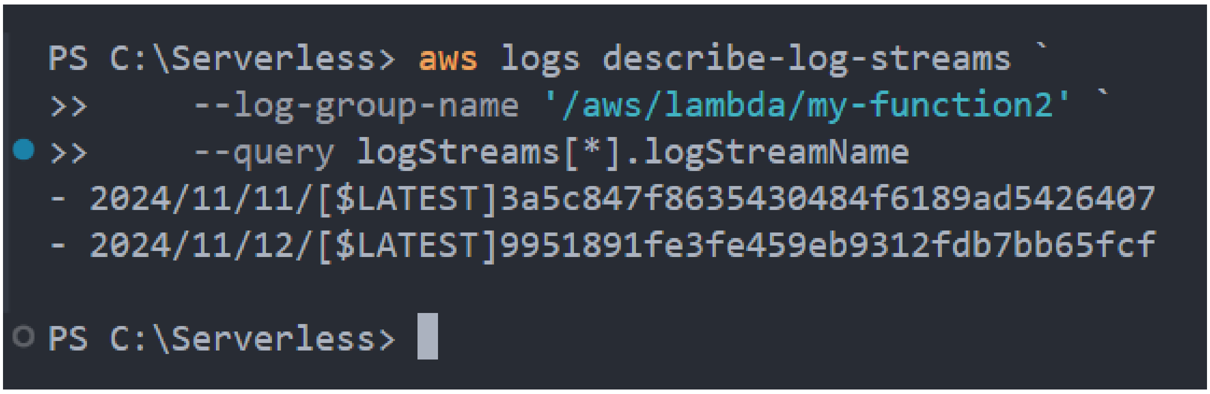


Log groups (2)

By default, we only load up to 10000 log groups.

□	Log group	Log class	Anomaly d...	Data p...	Sensiti...	Retenti...	Metric
□	/aws/lambda/my-function2	Standard	Configure	-	-	Never expire	-
□	/aws/lambda/myfirst-function	Standard	Configure	-	-	Never expire	-

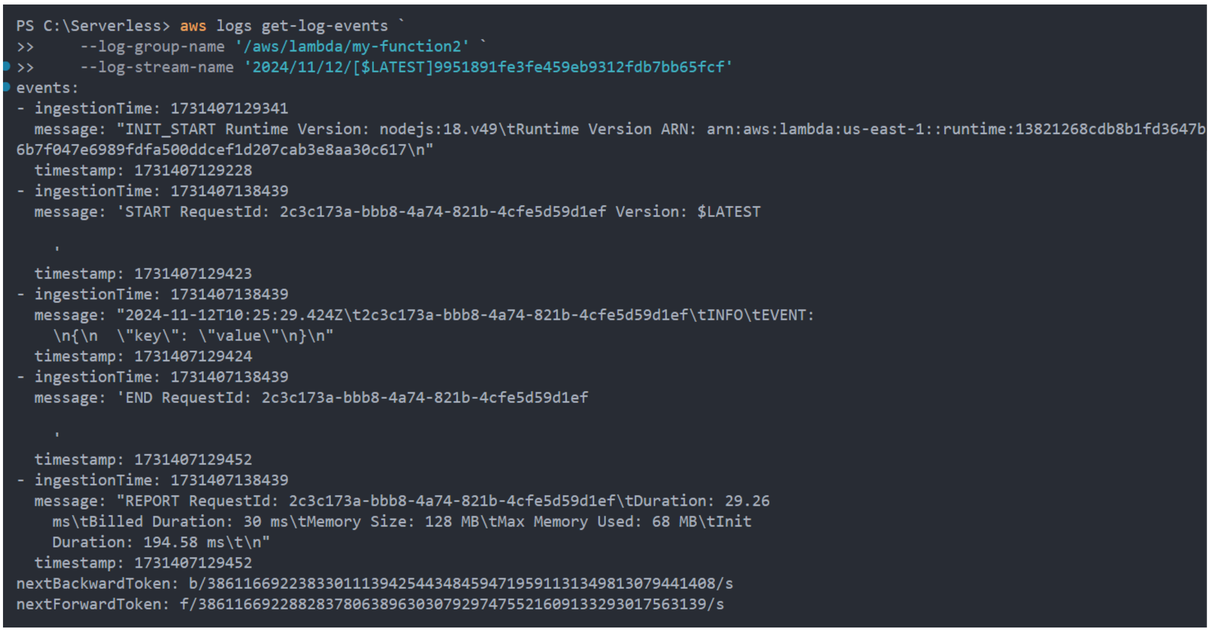
5. After that in the second command we choose a lambda function to give its log group name and then request it to give us the log stream names.



```
PS C:\Serverless> aws logs describe-log-streams `
>>     --log-group-name '/aws/lambda/my-function2' `
>>     --query logStreams[*].logStreamName
- 2024/11/11/[$LATEST]3a5c847f8635430484f6189ad5426407
- 2024/11/12/[$LATEST]9951891fe3fe459eb9312fdb7bb65fcf

PS C:\Serverless> 
```

6. In the last command we choose the latest log stream and put it in the command to get inside the log stream.
7. Below you can see that we get the output as expected and it is in the same format as should be inside the CloudWatch log stream.



```
PS C:\Serverless> aws logs get-log-events `
>>     --log-group-name '/aws/lambda/my-function2' `
>>     --log-stream-name '2024/11/12/[$LATEST]9951891fe3fe459eb9312fdb7bb65fcf'
events:
- ingestionTime: 1731407129341
  message: "INIT_START Runtime Version: nodejs:18.v49\tRuntime Version ARN: arn:aws:lambda:us-east-1::runtime:13821268cdb8b1fd3647b
6b7f047e6989fdfa500ddcef1d207cab3e8aa30c617\n"
  timestamp: 1731407129228
- ingestionTime: 1731407138439
  message: 'START RequestId: 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef Version: $LATEST

    '
  timestamp: 1731407129423
- ingestionTime: 1731407138439
  message: "2024-11-12T10:25:29.424Z\t2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef\tINFO\tEVENT:
    \n{\n  \"key\": \"value\"\n}\n"
  timestamp: 1731407129424
- ingestionTime: 1731407138439
  message: 'END RequestId: 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef

    '
  timestamp: 1731407129452
- ingestionTime: 1731407138439
  message: "REPORT RequestId: 2c3c173a-bbb8-4a74-821b-4cfe5d59d1ef\tDuration: 29.26
    ms\tBilled Duration: 30 ms\tMemory Size: 128 MB\tMax Memory Used: 68 MB\tInit
    Duration: 194.58 ms\t\n"
  timestamp: 1731407129452
nextBackwardToken: b/38611669223833011139425443484594719591131349813079441408/s
nextForwardToken: f/38611669228828378063896303079297475521609133293017563139/s
```