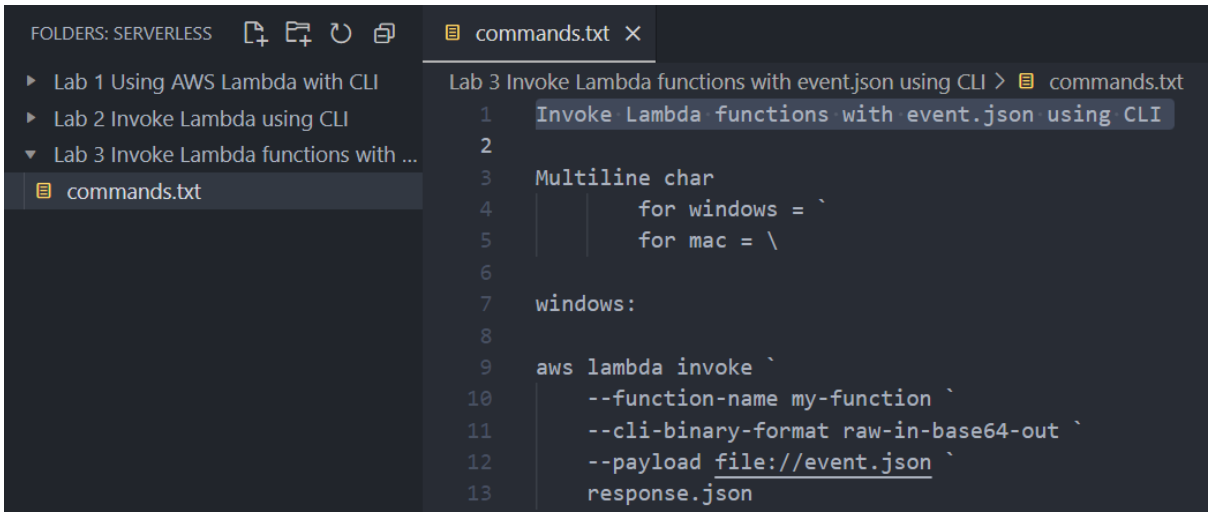
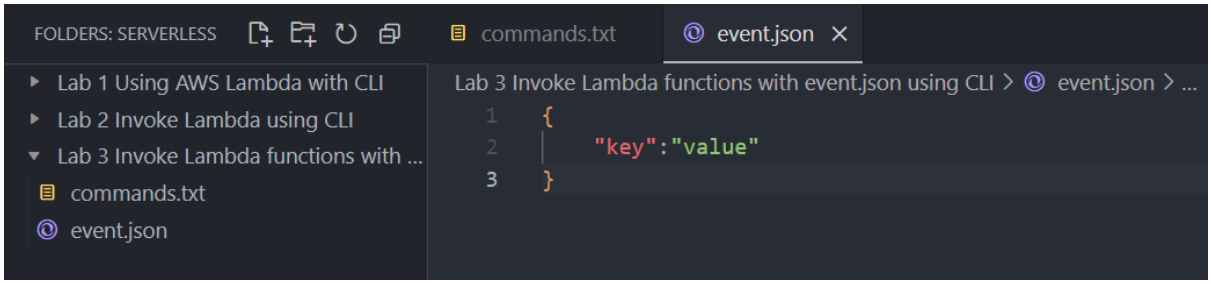
😊 Invoke Lambda functions with event.json using CLI

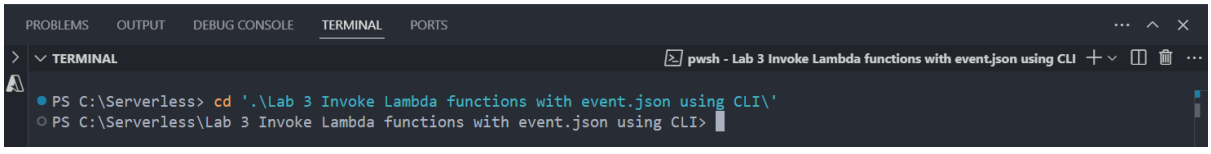
1. In this lab we will invoke our lambda function but this time we will use a JSON file to do that.
2. First, we need to create a new folder for this lab again create a commands file that contains all the commands used in this lab.



3. First, we need to create the event.json file VS Code. As you can see below we have created an events file and wrote the code to invoke the lambda function.



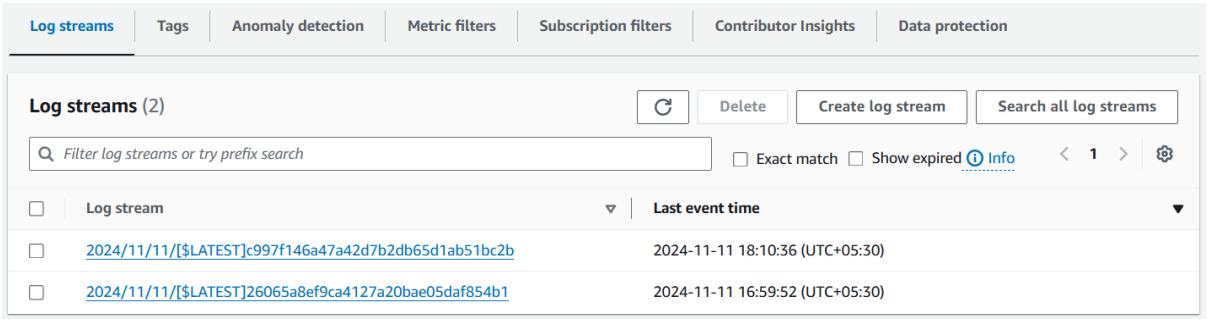
4. Now we need to run the invoke command and this will invoke our lambda function. But before that, you need to come inside the directory then run the invoke command or else you will get the error.



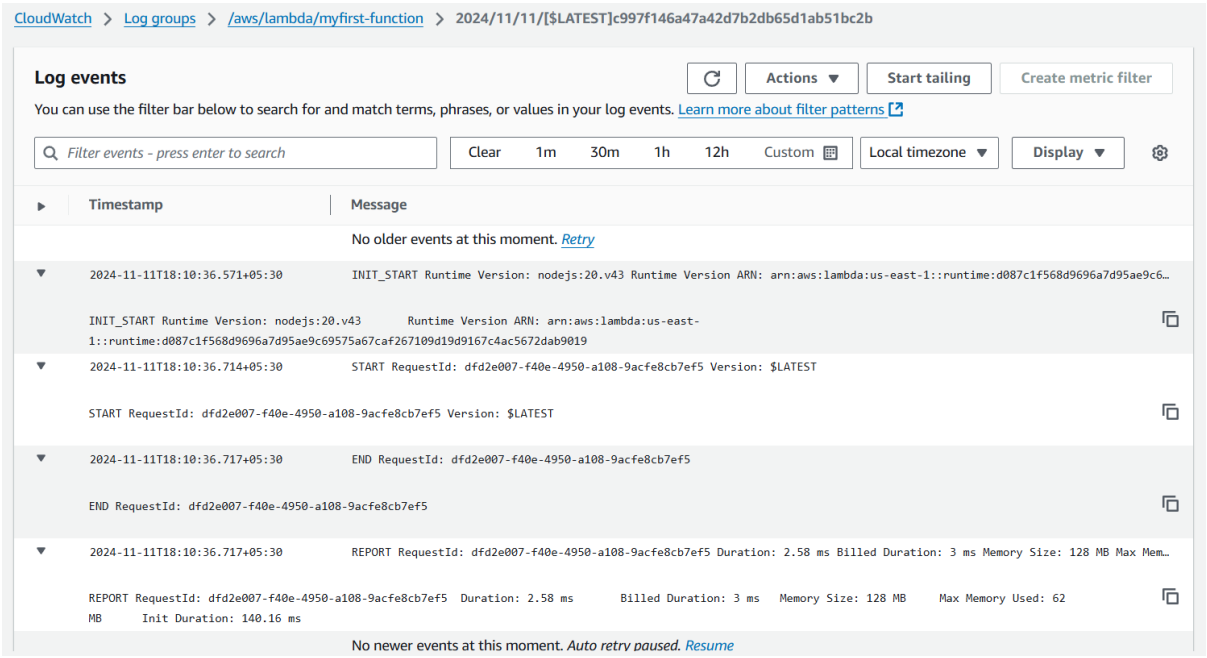
5. Below you can see, that our invoke command works properly.

```
TERMINAL                                                    pwsh - Lab 3 Invoke Lambda functions

PS C:\Serverless> cd '.\Lab 3 Invoke Lambda functions with event.json using CLI\'
PS C:\Serverless\Lab 3 Invoke Lambda functions with event.json using CLI> aws lambda invoke `
>>      --function-name myfirst-function `
>>      --cli-binary-format raw-in-base64-out `
>>      --payload file://event.json `
>>      response.json
ExecutedVersion: $LATEST
StatusCode: 200

PS C:\Serverless\Lab 3 Invoke Lambda functions with event.json using CLI>
```

6. Now if I go back to my function and open CloudWatch logs again. So, this time you will see that two logs generated. Open the latest one.



7. Here you will see that we the logs are generated on behalf of our invocation.



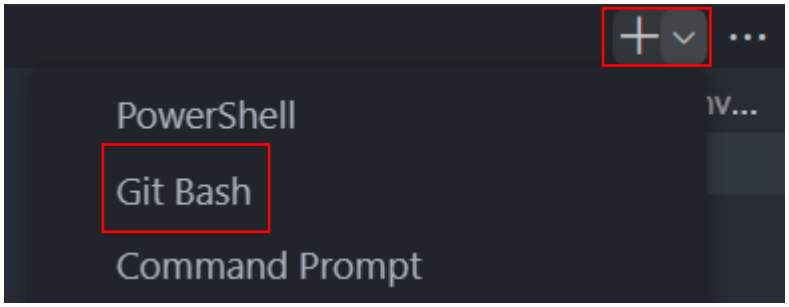
8. Now if we run the command shown below then you will see that this command has given us the log results from the CloudWatch but you will find that this is not in the readable format.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
TERMINAL                pwsh - Lab 3 Invoke Lambda functions with event.json using CLI

PS C:\Serverless\Lab 3 Invoke Lambda functions with event.json using CLI> aws lambda invoke --function-name myfirst-function ou
t --log-type Tail
ExecutedVersion: $LATEST
LogResult: U1RBUlQgUmVxdWVzdElkOiA5YmU2Y2YxNi04ZGI3LTQ4YmMtYmVlMS1iMGY0YTE0NmM4MTYgVmVyc2lvbjogJExBVEVTVApFTkQgUmVxdWVzdElkOiA5
YmU2Y2YxNi04ZGI3LTQ4YmMtYmVlMS1iMGY0YTE0NmM4MTYKUkVQT1JUIFJlcXVlc3RJZDogOWJlNmNmMTYtOGRiNy00OGJjLWJlZTEtYjBmNGExNDZjODE2CUR1cmF
0aW9uOiAxLjY3IG1zCUJpbGxlZCBEdXJhdGlvbjogMiBtcwlNZW1vcnkgU2l6ZTogMTI4IE1CCU1heCBNZW1vcnkgVXNlZDogNjIgTUIJCg==
StatusCode: 200
```

9. So, for that we need to run this command given below. But this thing is we cannot run this command directly on the PowerShell because it is a Linux.
10. So, for this command we need to open the bash terminal inside the VS Code.

    **aws lambda invoke --function-name myfirst-function out --log-type Tail **
    --query 'LogResult' --output text | base64 -d

11. You need to click on the Plus icon and then choose Git Bash from here.



12. After that run this command to get the latest logs for your function and you will see that the logs are there.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /c/Serverless
$ aws lambda invoke --function-name myfirst-function out --log-type Tail \
--query 'LogResult' --output text |  base64 -d
START RequestId: c95c7abd-0697-4809-898e-9d3ea75ee71b Version: $LATEST
END RequestId: c95c7abd-0697-4809-898e-9d3ea75ee71b
REPORT RequestId: c95c7abd-0697-4809-898e-9d3ea75ee71b  Duration: 11.49 ms      Billed Duration: 12 ms  Memory Size: 128
MB   Max Memory Used: 62 MB  Init Duration: 132.07 ms
base64: invalid input

PULKIT@LAPTOP-G2CAKBK8 MINGW64 /c/Serverless
$
```