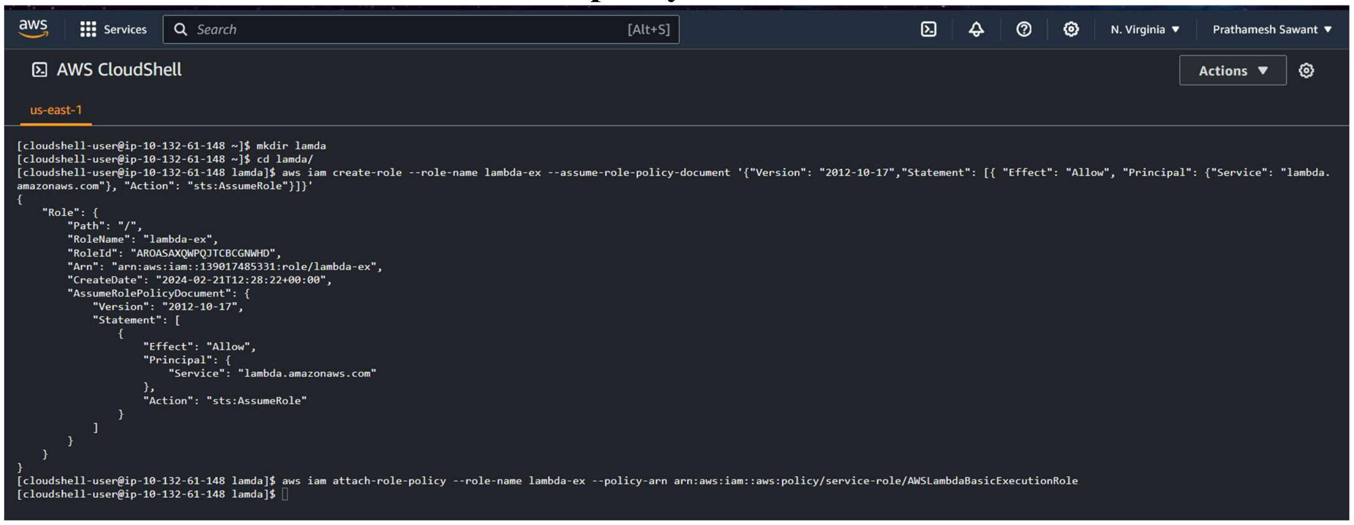
Prathamesh Sawant

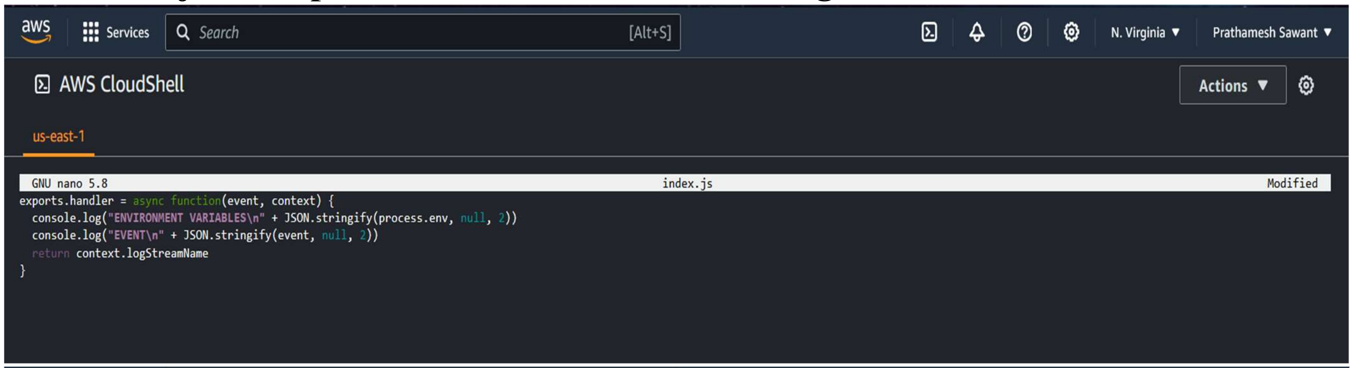
AWS Lambda using CLI

1. Create the execution role and attach policy

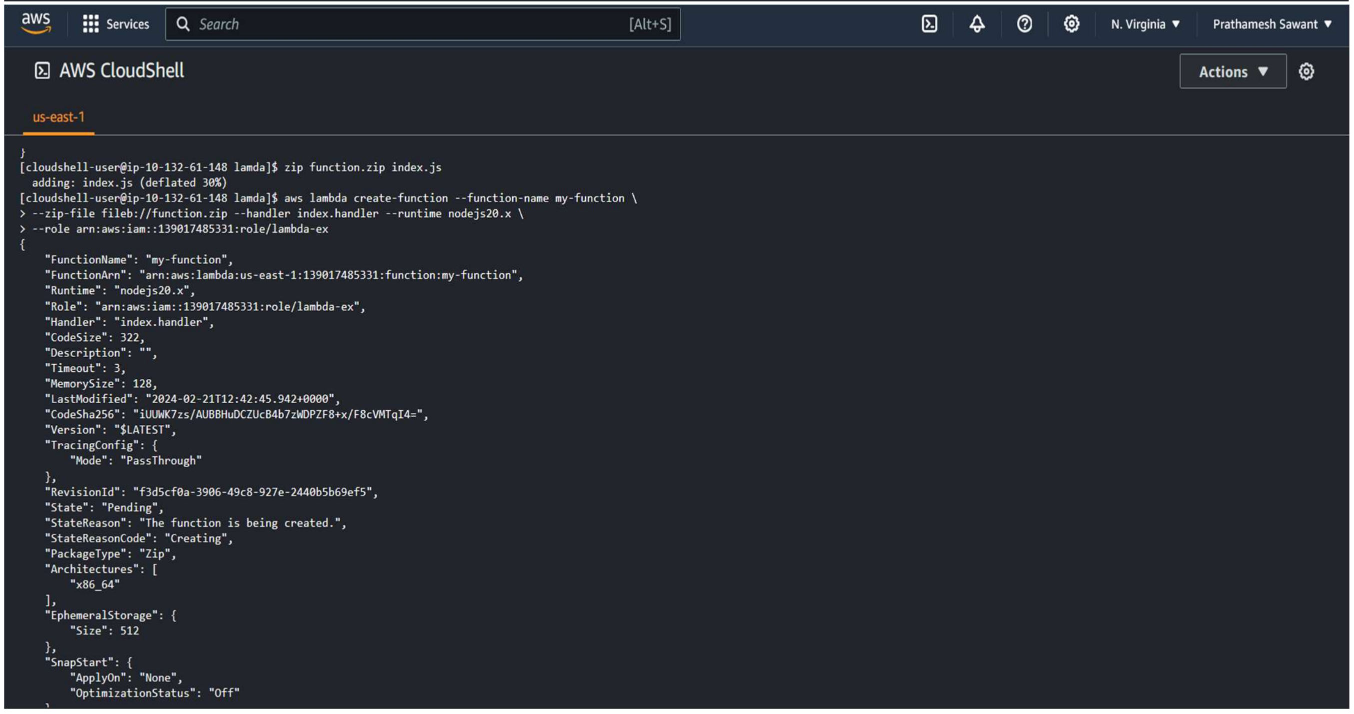


```
[cloudshell-user@ip-10-132-61-148 ~]$ mkdir lamda
[cloudshell-user@ip-10-132-61-148 ~]$ cd lamda/
[cloudshell-user@ip-10-132-61-148 lamda]$ aws iam create-role --role-name lambda-ex --assume-role-policy-document '{"Version": "2012-10-17","Statement": [{ "Effect": "Allow", "Principal": {"Service": "lambda.
amazonaws.com"}, "Action": "sts:AssumeRole"}]}'
{
    "Role": {
        "Path": "/",
        "RoleName": "lambda-ex",
        "RoleId": "AROASAXQWPQJTCBCGNWHD",
        "Arn": "arn:aws:iam::139017485331:role/lambda-ex",
        "CreateDate": "2024-02-21T12:28:22+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "lambda.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
[cloudshell-user@ip-10-132-61-148 lamda]$ aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
[cloudshell-user@ip-10-132-61-148 lamda]$ 
```

2. Add index.js file, zip it and create the function using 'create-function' command

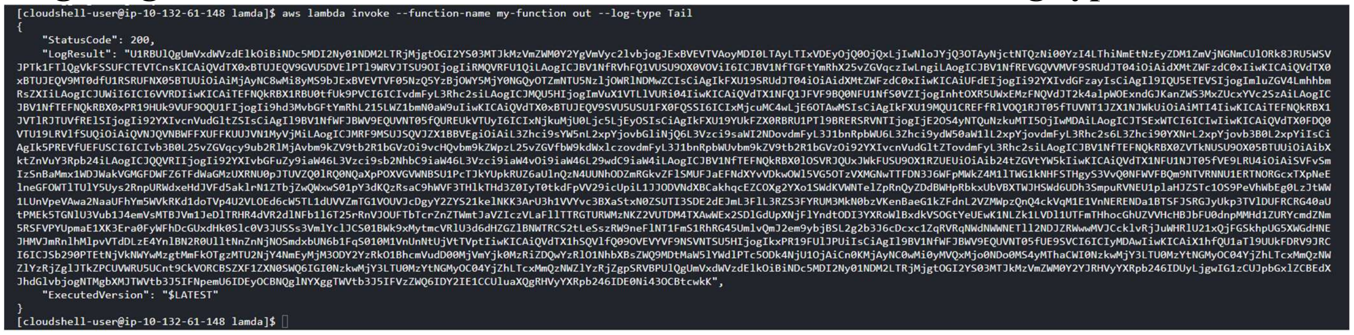


```
  GNU nano 5.8                                  index.js                                  Modified
exports.handler = async function(event, context) {
  console.log("ENVIRONMENT VARIABLES\n" + JSON.stringify(process.env, null, 2))
  console.log("EVENT\n" + JSON.stringify(event, null, 2))
  return context.logStreamName
}
```



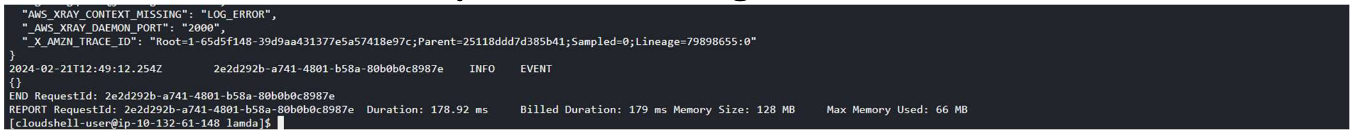
```
}
[cloudshell-user@ip-10-132-61-148 lamda]$ zip function.zip index.js
  adding: index.js (deflated 30%)
[cloudshell-user@ip-10-132-61-148 lamda]$ aws lambda create-function --function-name my-function \
> --zip-file fileb://function.zip --handler index.handler --runtime nodejs20.x \
> --role arn:aws:iam::139017485331:role/lambda-ex
{
    "FunctionName": "my-function",
    "FunctionArn": "arn:aws:lambda:us-east-1:139017485331:function:my-function",
    "Runtime": "nodejs20.x",
    "Role": "arn:aws:iam::139017485331:role/lambda-ex",
    "Handler": "index.handler",
    "CodeSize": 322,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2024-02-21T12:42:45.942+0000",
    "CodeSha256": "iUUWK7zs/AUBBHuDCZUcB4b7zWDPZF8+x/F8cVMTqI4=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "f3d5cf0a-3906-49c8-927e-2440b5b69ef5",
    "State": "Pending",
    "StateReason": "The function is being created.",
    "StateReasonCode": "Creating",
    "PackageType": "Zip",
    "Architectures": [
        "x86_64"
    ],
    "EphemeralStorage": {
        "Size": 512
    },
    "SnapStart": {
        "ApplyOn": "None",
        "OptimizationStatus": "Off"
```

3. **To get logs for an invocation from the command line, use the --log-type**



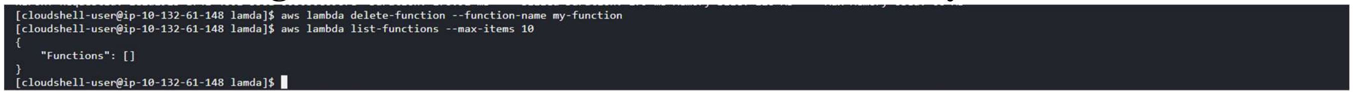
```
[cloudshell-user@ip-10-132-61-148 lamda]$ aws lambda invoke --function-name my-function out --log-type Tail
{
    "StatusCode": 200,
    "LogResult": "U1RBUlQgUmVxdWVzdElkOiBiNDc5MDI2Ny01NDM2LTRjMjgtOGI2YS03MTJkMzVmZWM0Y2YgVmVyc2lvbjogJExBVEVTVAoyMDI0LTAyLTIxVDEyOjQ0OjQxLjIwNloJYjQ3OTAyNjctNTQzNi00YzI4LThiNmEtNzEyZDM1ZmVjNGNmCUlORk8JRU5WSV
JPTk1FTlQgVkFSSUFCTEVTCnsKICAiQVdTX0xBTUJEQV9GVU5DVElPTl9WRVJTSU9OIjogIiRMQVRFU1QiLAogICJBV1NfRVhFQ1VUSU9OX0VOViI6ICJBV1NfTGFtYmRhX25vZGVqczIwLngiLAogICJBV1NfREVGQVVMVF9SRUdJT04iOiAidXMtZWFzdC0xIiwKICAiQVdTX0
xBTUJEQV9MT0dfU1RSRUFNX05BTUUiOiAiMjAyNC8wMi8yMS9bJExBVEVTVF05NzQ5YzBjOWY5MjY0NGQyOTZmNTU5NzljOWRlNDMwZCIsCiAgIkFXU19SRUdJT04iOiAidXMtZWFzdC0xIiwKICAiUFdEIjogIi92YXIvdGFzayIsCiAgIl9IQU5ETEVSIjogImluZGV4Lmhhbm
RsZXIiLAogICJUWiI6ICI6VVRDIiwKICAiTEFNQkRBX1RBU0tfUk9PVCI6ICIvdmFyL3Rhc2siLAogICJMQU5HIjogImVuX1VTLlVURi04IiwKICAiQVdTX1NFQ1JFVF9BQ0NFU1NfS0VZIjogInhtOXR5UWxEMzFNQVdJT2k4alpWOExndGJKanZWS3MxZUcxYVc2SzAiLAogIC
JBV1NfTEFNQkRBX0xPR19HUk9VUF9OQU1FIjogIi9hd3MvbGFtYmRhL215LWZ1bmN0aW9uIiwKICAiQVdTX0xBTUJEQV9SVU5USU1FX0FQSSI6ICIxMjcuMC4wLjE6OTAwMSIsCiAgIkFXU19MQU1CREFfRlVOQ1RJT05fTUVNT1JZX1NJWkUiOiAiMTI4IiwKICAiTEFNQkRBX1
JVTlRJTUVfRElSIjogIi92YXIvcnVudGltZSIsCiAgIl9BV1NfWFJBWV9EQUVNT05fQUREUkVTUyI6ICIxNjkuMjU0Ljc5LjEyOSIsCiAgIkFXU19YUkFZX0RBRU1PTl9BRERSRVNTIjogIjE2OS4yNTQuNzkuMTI5OjIwMDAiLAogICJTSExWTCI6ICIwIiwKICAiQVdTX0FDQ0
VTU19LRVlfSUQiOiAiQVNJQVNBWFFXUFFKUUJVN1MyVjMiLAogICJMRF9MSUJSQVJZX1BBVEgiOiAiL3Zhci9sYW5nL2xpYjovbGliNjQ6L3Vzci9saWI2NDovdmFyL3J1bnRpbWU6L3Zhci9ydW50aW1lL2xpYjovdmFyL3Rhc2s6L3Zhci90YXNrL2xpYjovb3B0L2xpYiIsCi
AgIk5PREVfUEFUSCI6ICIvb3B0L25vZGVqcy9ub2RlMjAvbm9kZV9tb2R1bGVzOi9vcHQvbm9kZWpzL25vZGVfbW9kdWxlczovdmFyL3J1bnRpbWUvbm9kZV9tb2R1bGVzOi92YXIvcnVudGltZTovdmFyL3Rhc2siLAogICJBV1NfTEFNQkRBX0ZVTkNUSU9OX05BTUUiOiAibX
ktZnVuY3Rpb24iLAogICJQQVRIIjogIi92YXIvbGFuZy9iaW46L3Vzci9sb2NhbC9iaW46L3Vzci9iaW4vOi9iaW46L29wdC9iaW4iLAogICJBV1NfTEFNQkRBX0lOSVRJQUxJWkFUSU9OX1RZUEUiOiAib24tZGVtYW5kIiwKICAiQVdTX1NFU1NJT05fVE9LRU4iOiAiSVFvSm
IzSnBaMmx1WDJWakVGMGFDWFZ6TFdWaGMzUXRNU0pJTUVZQ0lRQ0NQaXpPOXVGVWNBSU1PcTJkYUpkRUZ6aUlnQzN4UUNhODZmRGkvZFlSMUFJaEFNdXYvVDkwOWl5VG5OTzVXMGNwTTFDN3J6WFpMWkZ4M1lTWG1kNHFSTHgyS3VvQ0NFWVFBQm9NTVRNNU1ERTNORGcxTXpNeE
lneGFOWTlTUlY5Uys2RnpURWdxeHdJVFd5aklrN1ZTbjZwQWxwS01pY3dKQzRsaC9hWVF3THlkTHd3Z0IyT0tkdFpVV29icUpiL1JJODVNdXBCakhqcEZCOXg2YXo1SWdKVWNTelZpRnQyZDdBWHpRbkxUbVBXTWJHSWd6UDh3SmpuRVNEU1plaHJZSTc1OS9PeVhWbEg0LzJtWW
1LUnVpeVAwa2NaaUFhYm5WVkRKd1doTVp4U2VLOEd6cW5TL1dUVVZmTG1VOUVJcDgyY2ZYS21kelNKK3ArU3h1VVYvc3BXaStxN0ZSUTI3SDE2dEJmL3FlL3RZS3FYRUM3MkN0bzVKenBaeG1kZFdnL2VZMWpzQnQ4ckVqM1E1VnNERENDa1BTSFJSRGJyUkp3TVlDUFRCRG40aU
tPMEk5TGNlU3Vub1J4emVsMTBJVm1JeDlTRHR4dVR2dlNFb1l6T25rRnVJOUFTbTcrZnZTWmtJaVZIczVLaFllTTRGTURWMzNKZ2VUTDM4TXAwWEx2SDlGdUpXNjFlYndtODI3YXRoWlBxdkVSOGtYeUEwK1NLZk1LVDl1UTFmTHhocGhUZVVHcHBJbFU0dnpMMHd1ZURYcmdZNm
5RSFVPYUpmaE1XK3Era0FyWFhDcGUxdHk0Slc0V3JUSSs3VmlYclJCS01BWk9xMytmcVRlU3d6dHZGZlBNWTRCS2tLeSszRW9neFlNT1FmS1RhRG45UmlvQmJ2em9ybjBSL2g2b3J6cDcxc1ZqRVRqNWdNWWNETll2NDJZRWwwMVJCcklvRjJuWHRlU21xQjFGSkhpUG5XWGdHNE
JHMVJmRnlhMlpvVTdDLzE4YnlBN2R0UlltNnZnNjNOSmdxbUN6b1FqS010M1VnUnNtUjVtTVptIiwKICAiQVdTX1hSQVlfQ09OVEVYVF9NSVNTSU5HIjogIkxPR19FUlJPUiIsCiAgIl9BV1NfWFJBWV9EQUVNT05fUE9SVCI6ICIyMDAwIiwKICAiX1hfQU1aTl9UUkFDRV9JRC
I6ICJSb290PTEtNjVkNWYwMzgtMmFkOTgzMTU2NjY4NmEyMjM3ODY2YzRkO1BhcmVudD00MjVmYjk0MzRiZDQwYzRlO1NhbXBsZWQ9MDtMaW5lYWdlPTc5ODk4NjU1OjAiCn0KMjAyNC0wMi0yMVQxMjo0NDo0MS4yMThaCWI0NzkwMjY3LTU0MzYtNGMyOC04YjZhLTcxMmQzNW
ZlYzRjZglJTkZPCUVWRU5UCnt9CkVORCBSZXF1ZXN0SWQ6IGI0NzkwMjY3LTU0MzYtNGMyOC04YjZhLTcxMmQzNWZlYzRjZgpSRVBPUlQgUmVxdWVzdElkOiBiNDc5MDI2Ny01NDM2LTRjMjgtOGI2YS03MTJkMzVmZWM0Y2YJRHVyYXRpb246IDUyLjgwIG1zCUJpbGxlZCBEdX
JhdGlvbjogNTMgbXMJTWVtb3J5IFNpemU6IDEyOCBNQglNYXggTWVtb3J5IFVzZWQ6IDY2IE1CCUluaXQgRHVyYXRpb246IDE0Ni43OCBtcwkK",
    "ExecutedVersion": "$LATEST"
}
[cloudshell-user@ip-10-132-61-148 lamda]$
```

4. **You can use the base64 utility to decode the logs**



```
  "AWS_XRAY_CONTEXT_MISSING": "LOG_ERROR",
  "_AWS_XRAY_DAEMON_PORT": "2000",
  "_X_AMZN_TRACE_ID": "Root=1-65d5f148-39d9aa431377e5a57418e97c;Parent=25118ddd7d385b41;Sampled=0;Lineage=79898655:0"
}
2024-02-21T12:49:12.254Z        2e2d292b-a741-4801-b58a-80b0b0c8987e    INFO    EVENT
{}
END RequestId: 2e2d292b-a741-4801-b58a-80b0b0c8987e
REPORT RequestId: 2e2d292b-a741-4801-b58a-80b0b0c8987e  Duration: 178.92 ms     Billed Duration: 179 ms Memory Size: 128 MB     Max Memory Used: 66 MB
[cloudshell-user@ip-10-132-61-148 lamda]$
```

5. **Run the following delete-function command to delete the my-function function**



```
[cloudshell-user@ip-10-132-61-148 lamda]$ aws lambda delete-function --function-name my-function
[cloudshell-user@ip-10-132-61-148 lamda]$ aws lambda list-functions --max-items 10
{
    "Functions": []
}
[cloudshell-user@ip-10-132-61-148 lamda]$
```