

# Documentation of serverless assignment

## 1<sup>st</sup> Step

- Creating the AWS account for access the Lambda service.
- Creating access key.

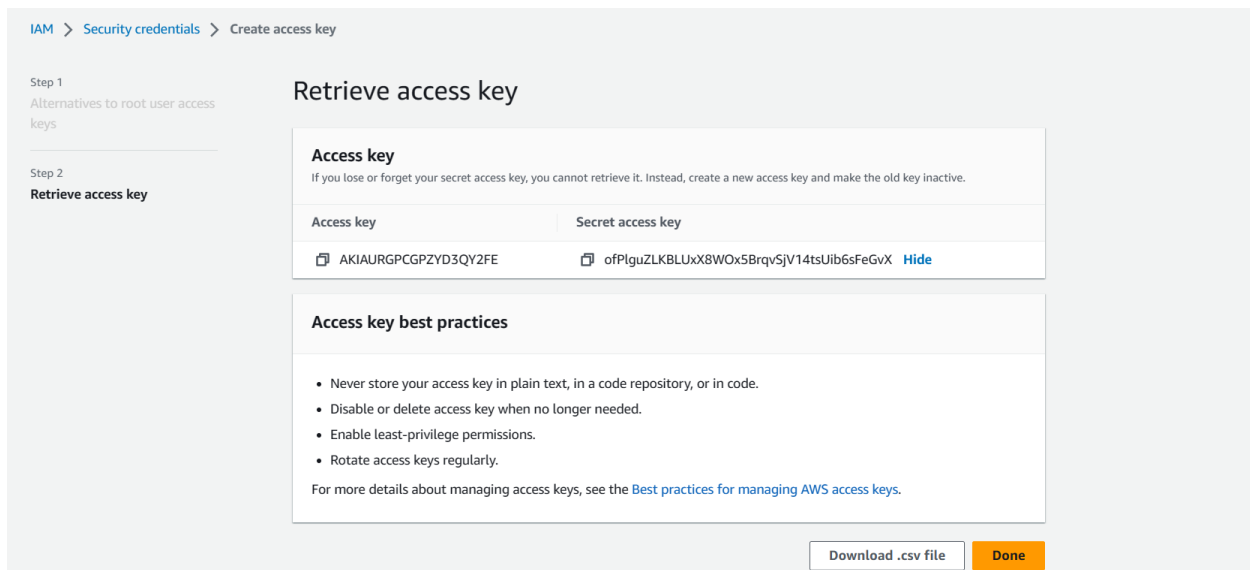


Figure 1: Retrieve access key

## 2<sup>nd</sup> Step

- Installing terraform to the environment (Vscode).  
v1.4.2 (latest version)

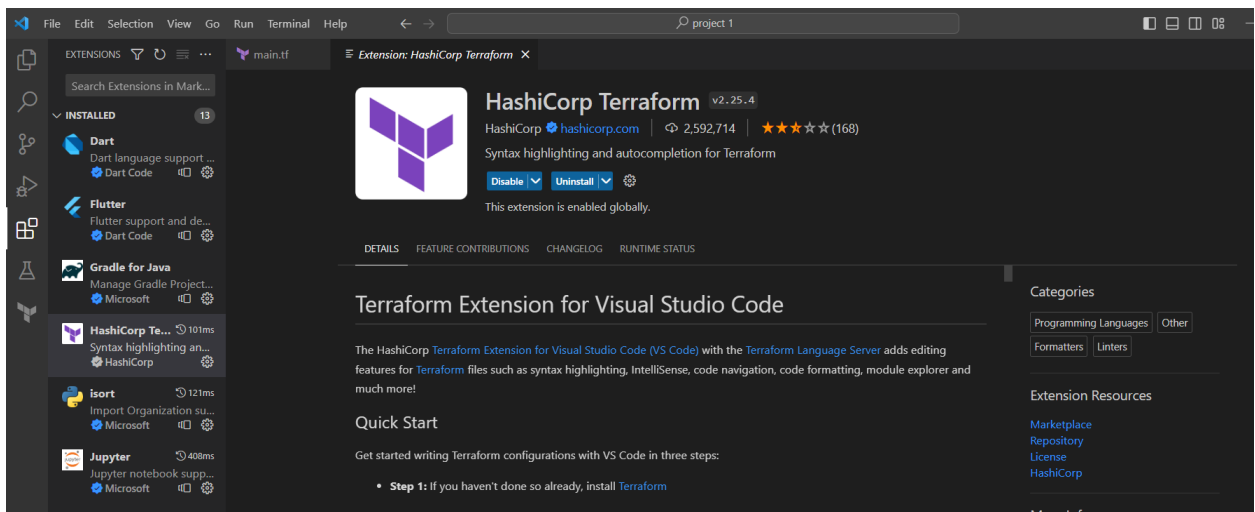


Figure 2: HashiCorp Terraform

### 3<sup>rd</sup> Step

- Installing AWS provider to the working environment.
- Installed AWS provider latest version 4.59.0v

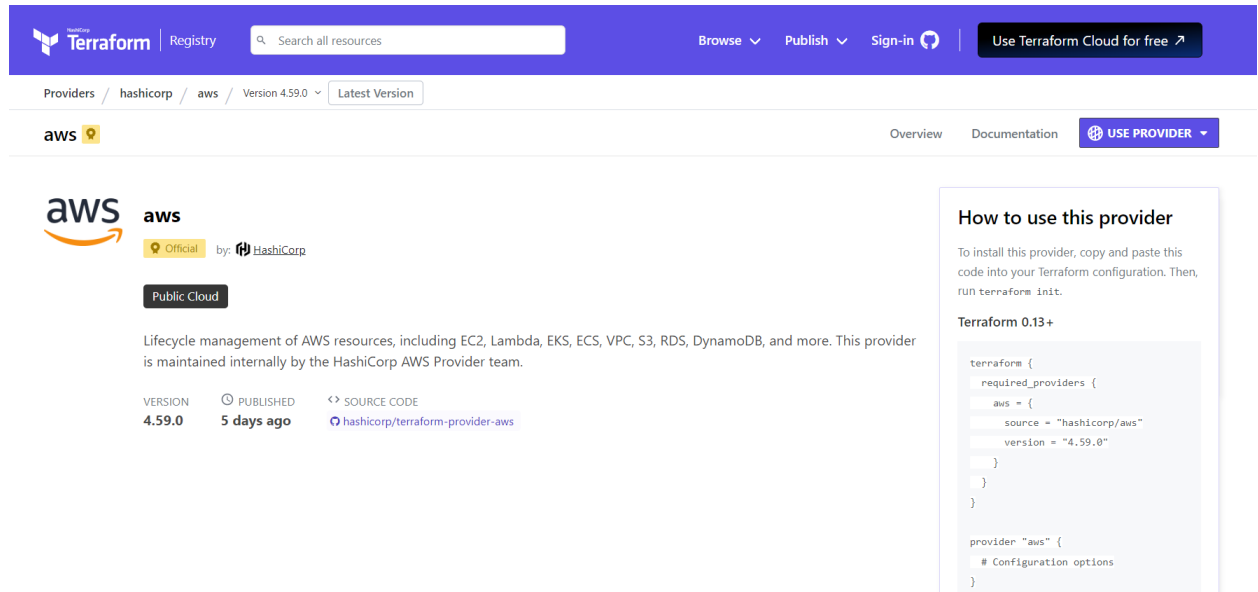


Figure 3: version 4.59.0v

- But when using the latest version of AWS provider got an error.

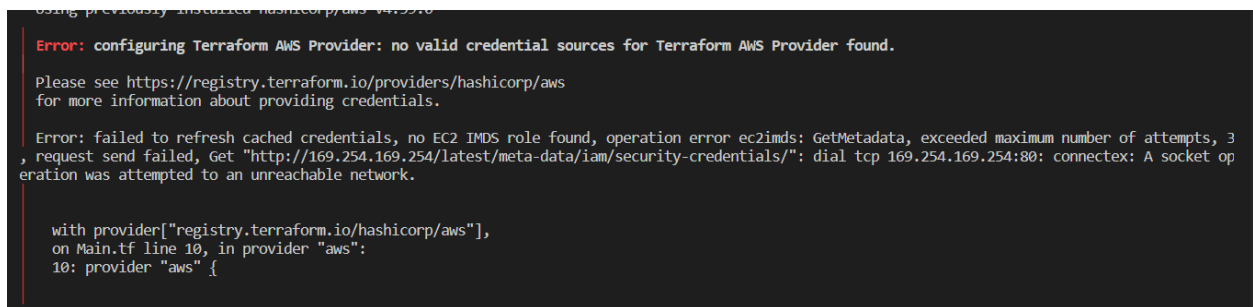


Figure 4: Error to connect with AWS console by my credentials

Solution:

When searching for a solution for this error I discover that there is a problem with latest versions of AWS provider 4.0.0+v. Therefore, I have reinstalled AWS provider Version 3.7.0

Source- <https://github.com/hashicorp/terraform-provider-aws/issues/13057>

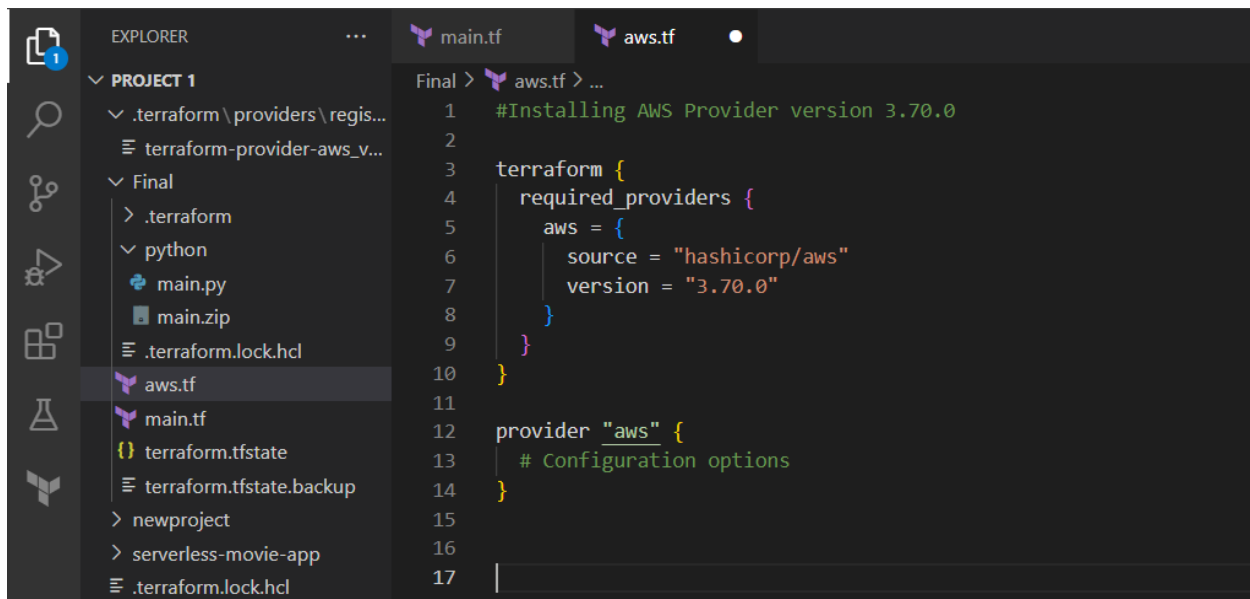


Figure 5: Code for Installing AWS provider

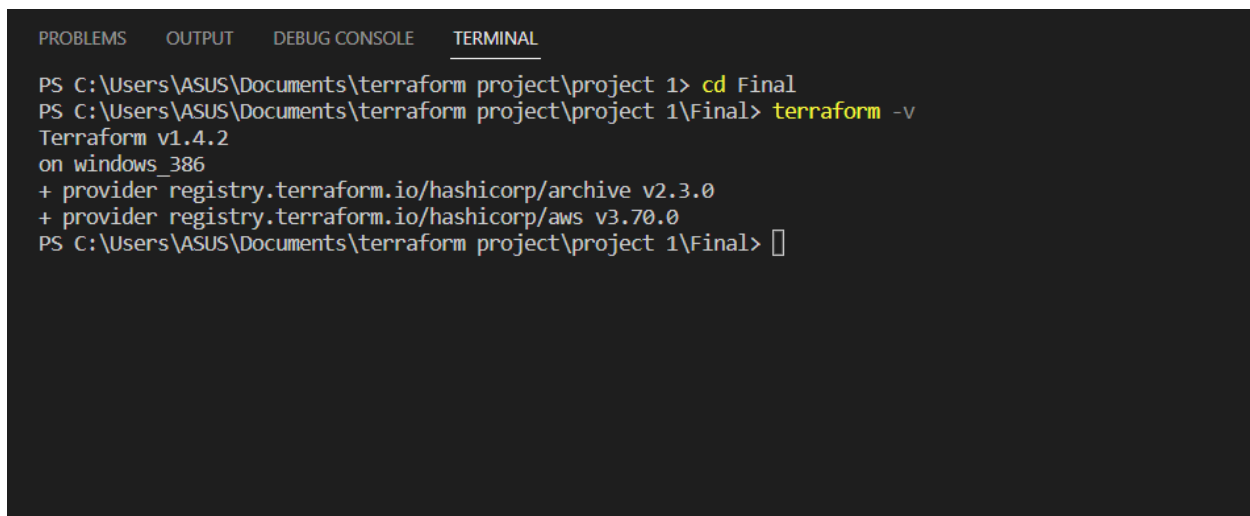


Figure 6: Checking the version

#### 4<sup>th</sup> Step

- Develop a terraform code to manage AWS Lambda resources.
- Develop a lambda function using python.

- Successfully added AWS Lambda resources through Terraform.

```
PS C:\Users\ASUS\Documents\terraform project\project 1\Final> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/archive from the dependency lock file
- Using previously-installed hashicorp/aws v3.70.0
- Using previously-installed hashicorp/archive v2.3.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\ASUS\Documents\terraform project\project 1\Final> |
```

Figure 7: Initializes Terraform Code

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

+ created_date          = (known after apply)
+ description           = "REST API for movie serverless application"
+ disable_execute_api_endpoint = (known after apply)
+ execution_arn         = (known after apply)
+ id                    = (known after apply)
+ minimum_compression_size = -1
+ name                  = "serverless_rest_api"
+ policy                = (known after apply)
+ root_resource_id      = (known after apply)
+ tags_all              = (known after apply)
}

Plan: 6 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ api_gateway_url = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly t
PS C:\Users\ASUS\Documents\terraform project\project 1\Final> |
```

Figure 8: preview the actions of the Terraform Code

```
17 resource "aws_lambda_function" "movie_serverless" {  
  ...  
}  
  
Enter a value: yes  
  
aws_api_gateway_rest_api.serverless_rest_api: Creating...  
aws_api_gateway_rest_api.serverless_rest_api: Creation complete after 3s [id=t8vpsymkva]  
aws_api_gateway_resource.serverless_post: Creating...  
aws_api_gateway_resource.serverless_post: Creation complete after 2s [id=2u6x9t]  
aws_api_gateway_method.serverless_post_method: Creating...  
aws_api_gateway_method.serverless_post_method: Creation complete after 1s [id=agm-t8vpsymkva-2u6x9t-POST]  
aws_api_gateway_integration.serverless_lambda_integration: Creating...  
aws_api_gateway_integration.serverless_lambda_integration: Creation complete after 2s [id=agi-t8vpsymkva-2u6x9t-POST]  
aws_api_gateway_deployment.serverless_deployment: Creating...  
aws_api_gateway_deployment.movie_serverless: Creating...  
aws_api_gateway_deployment.movie_serverless: Creation complete after 2s [id=9rxo8z]  
aws_api_gateway_deployment.serverless_deployment: Creation complete after 5s [id=3ziifa]  
  
Apply complete! Resources: 6 added, 0 changed, 0 destroyed.  
  
Outputs:  
  
api_gateway_url = "https://t8vpsymkva.execute-api.us-east-1.amazonaws.com/prod"  
PS C:\Users\ASUS\Documents\terraform project\project 1\Final>
```

Figure 9: Successfully adding the AWS Lambda resources

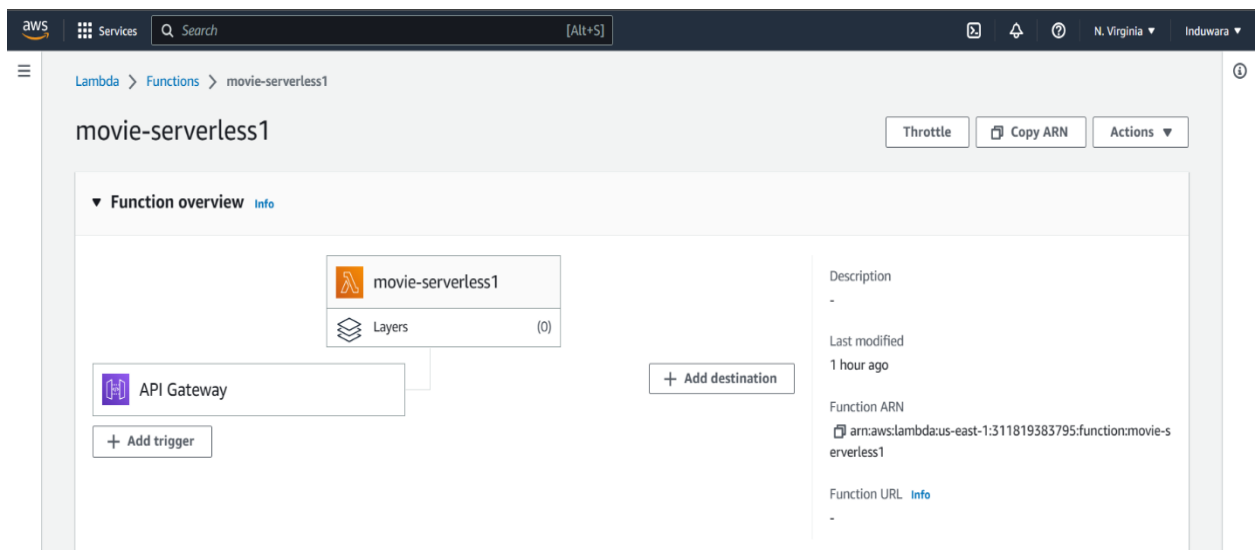


Figure 10: Overview of the AWS Lambda console after adding the resources.

## 5<sup>th</sup> Step

- Trigger the Rest API.

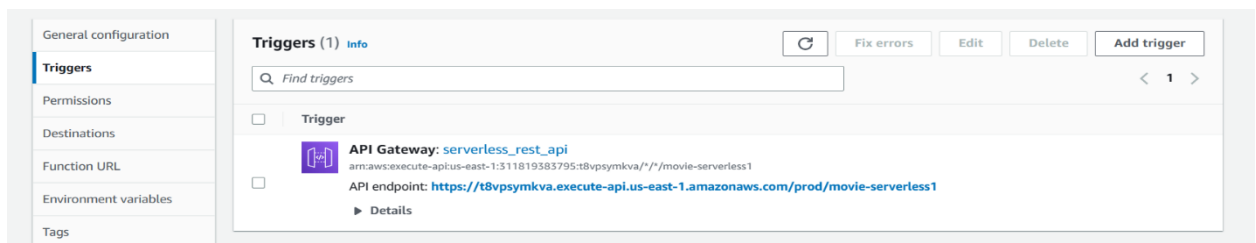
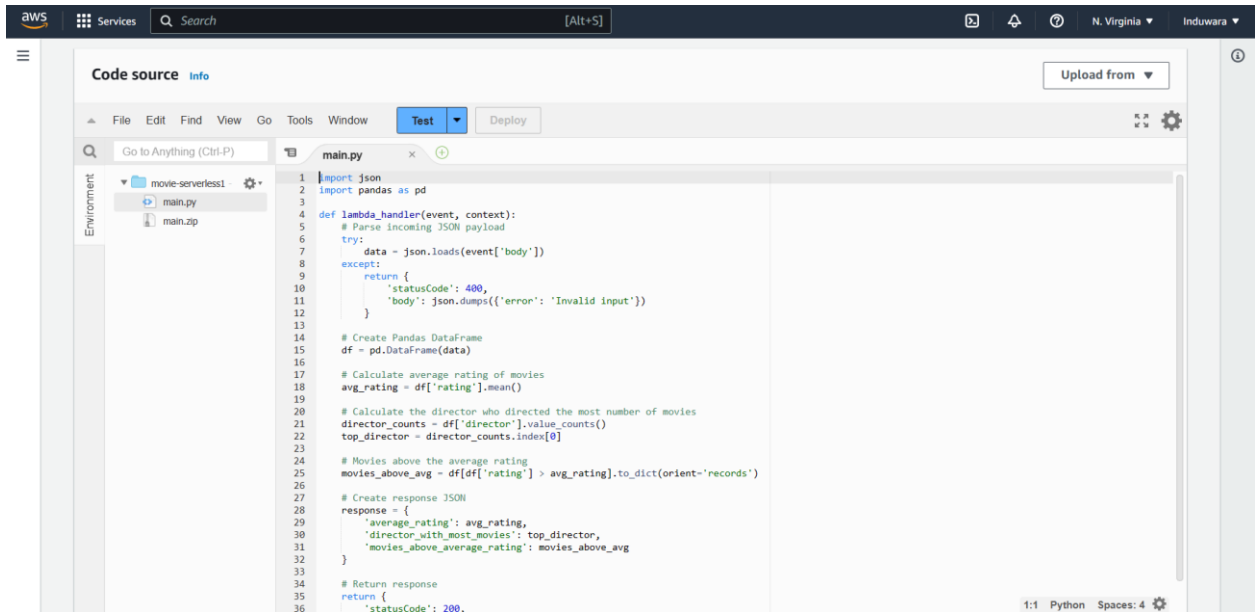


Figure 11: Added Rest API

- Testing the Lambda Function.



The screenshot shows the AWS Lambda console interface. The 'Code source' tab is selected, displaying the Python code for the 'main.py' file. The code imports 'json' and 'pandas as pd'. It defines a 'lambda\_handler' function that takes 'event' and 'context' as arguments. The function first parses the incoming JSON payload. If successful, it loads the data and calculates the average rating of movies using 'df['rating'].mean()'. It also identifies the director with the most movies using 'df['director'].value\_counts()'. Finally, it filters movies above the average rating and returns a JSON response containing the average rating, the director with the most movies, and the list of movies above the average rating. The code is written in Python 3.7 and is ready for deployment.

Figure 12: Successfully Lambda function Code have delivered

- Submitting JSON request to the Lambda function

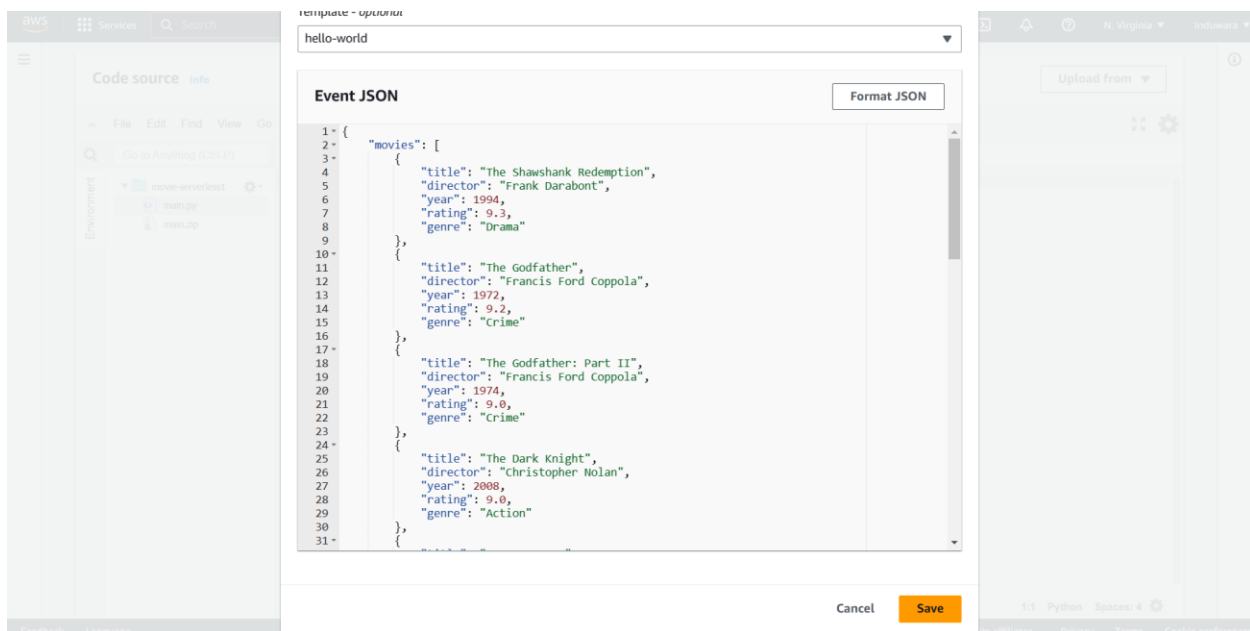


Figure 13: Saving the JSON request

## 6<sup>th</sup> Step

- Running the Lambda function code in the AWS Lambda console
- Error on Running the Lambda function code

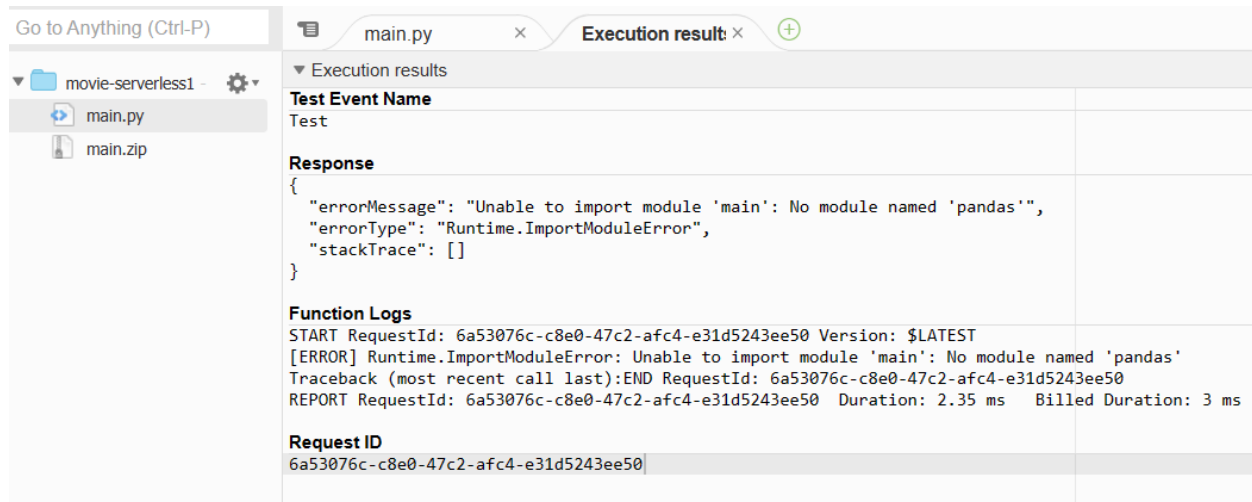


Figure 14: Error on Pandas library

As a solution,

I tried to do “pip install pandas” on Lambda function code but still the error is receiving.

Then I tried to install Pandas by adding a layer to the function.

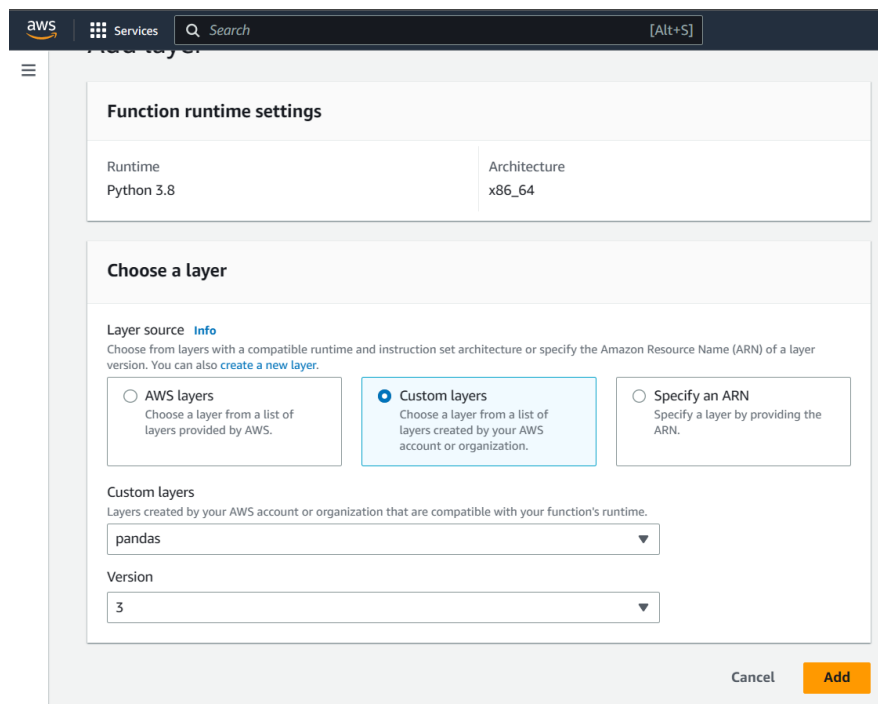


Figure 15: Adding Pandas as a custom layer to the function

Also I tried to overcome this error by adding extra libraries to the function as layers.

Added Libraries,

- Pandas v1.5.3 Linux
- Numpy v1.24.2
- Pytz v2022.7.2

But still the AWS Lambda environment not identifying the Pandas Library.

These are the resources that I used,

- How to use pandas in AWS Lambda

<https://www.gcptutorials.com/post/how-to-use-pandas-in-aws-lambda>

- Lambda layer to use Numpy and Pandas in AWS Lambda Function

<https://medium.com/@shimo164/lambda-layer-to-use-numpy-and-pandas-in-aws-lambda-function-8a0e040faa18>