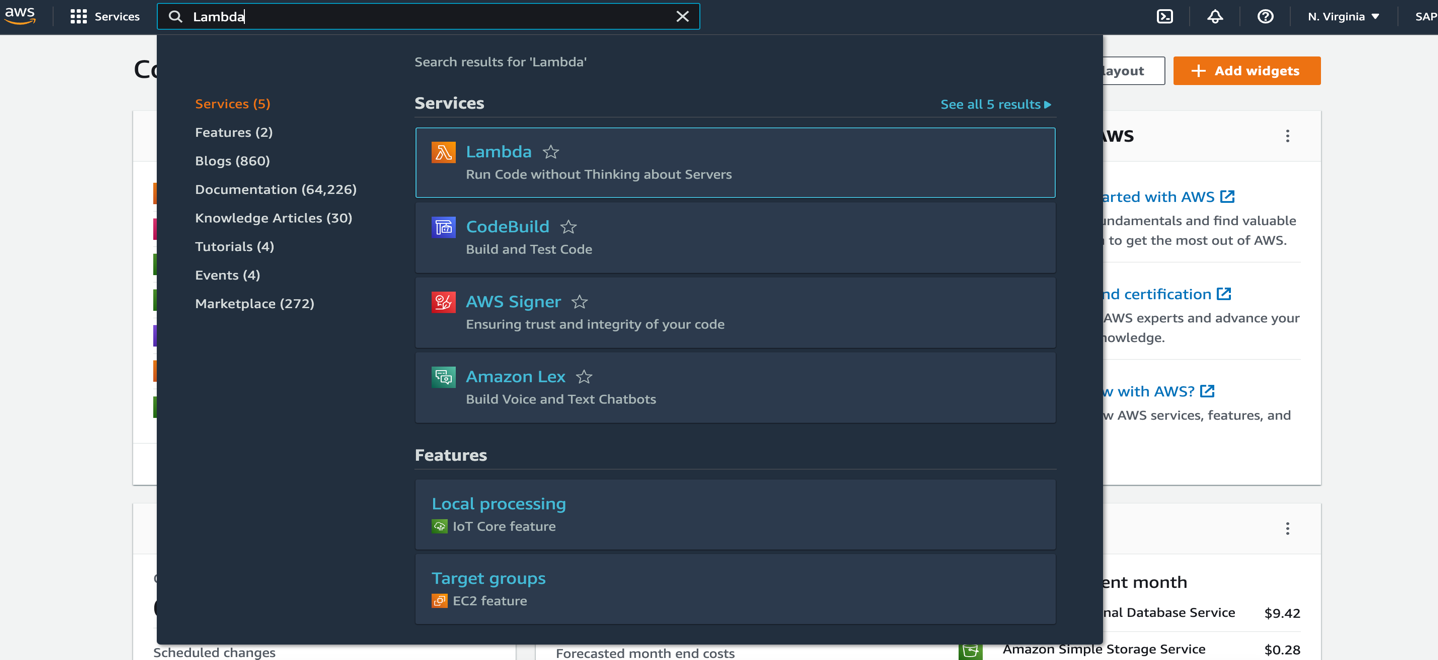
## Module 1

**Assignment 5:** Create the Lambda function

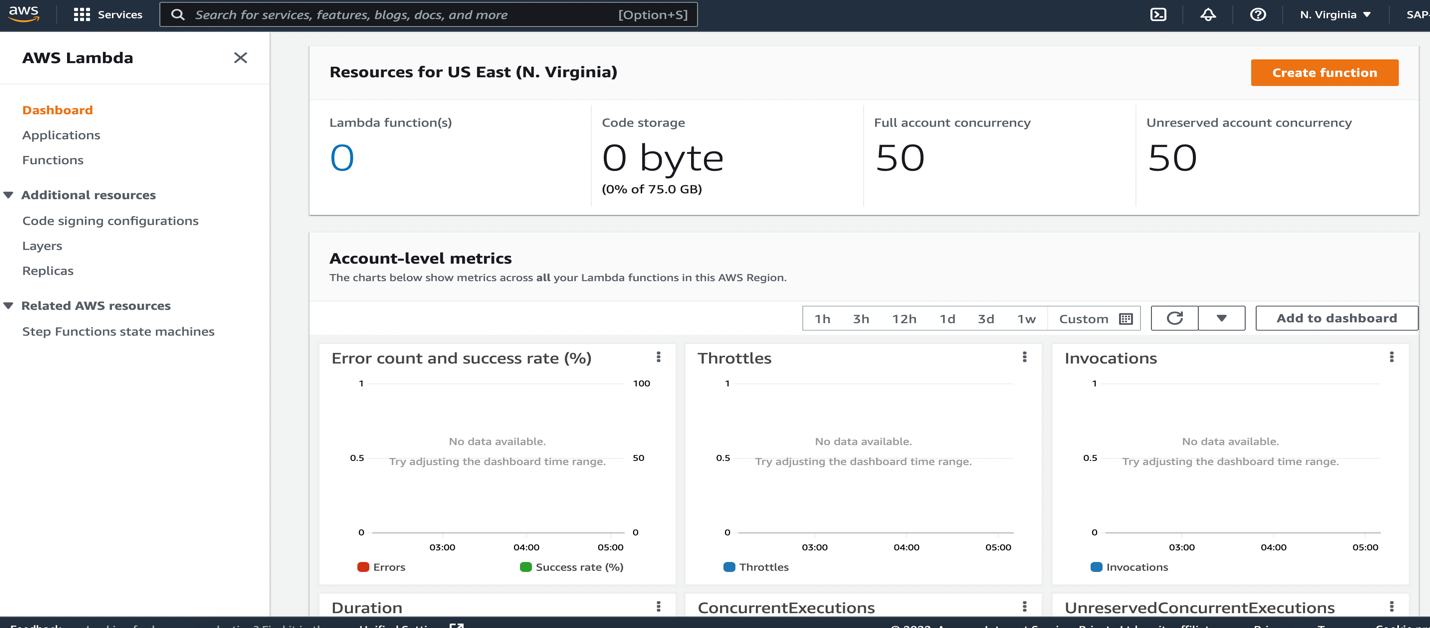
Lambda –

* Lambda is a compute service that lets you run code without provisioning or managing servers.
* With Lambda, you can run code for virtually any type of application or backend service.
* Lambda runs your function only when needed and scales automatically, from a few requests per day to thousands per second.
* You pay only for the compute time that you consume—there is no charge when your code is not running.
* You can invoke your Lambda functions using the Lambda API, or Lambda can run your functions in response to events from other AWS services.

Step 1: Search for Lambda in the search box on AWS Console.



Step 2: On Lambda dashboard, click on ‘Create function.

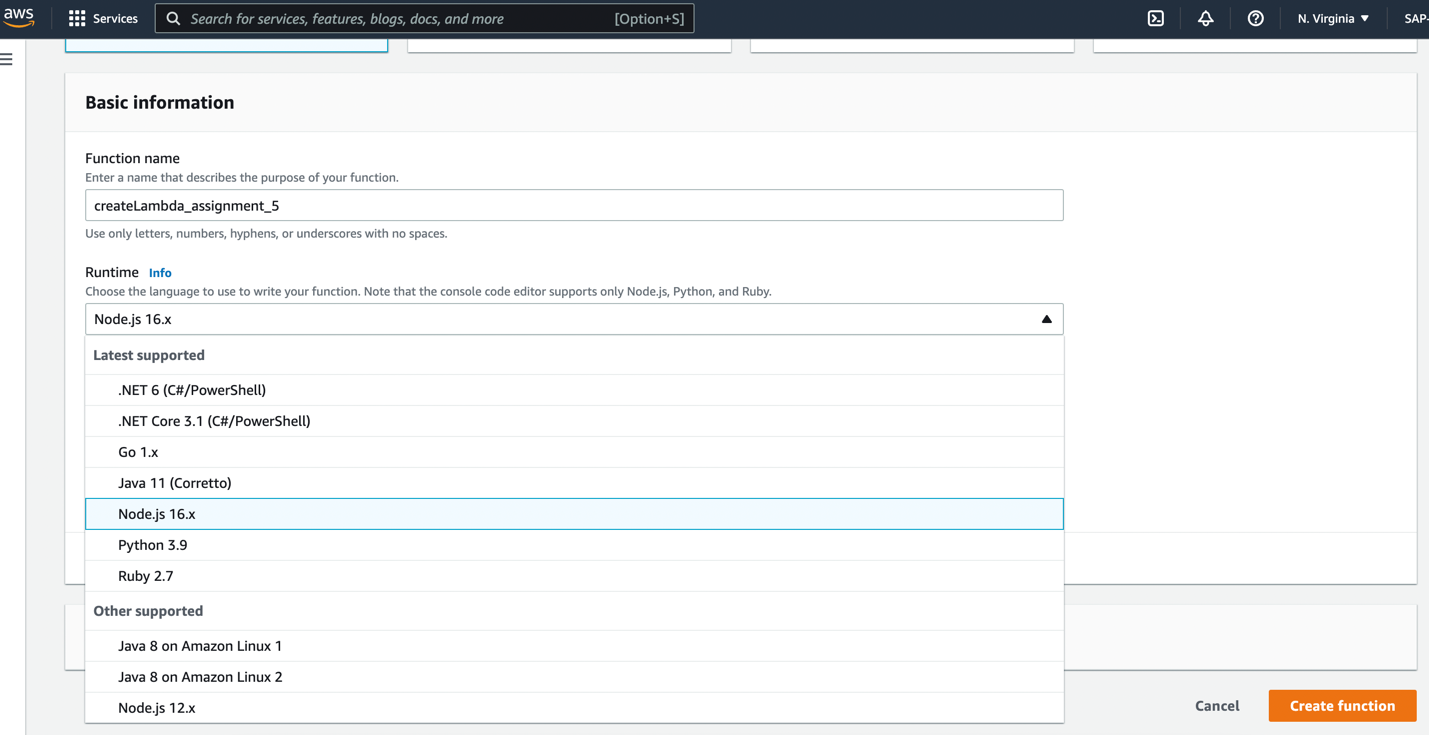


Step 3: We can choose 1 of the 4 options to create our function.

1. Author from scratch – if we want to write the code from scratch.
2. Use a blueprint – common code sample template can be used.
3. Container image – A container image can be used to deploy the code
4. Browse serverless app repository – Sample Lambda AWS Serverless application can be used from Repository.

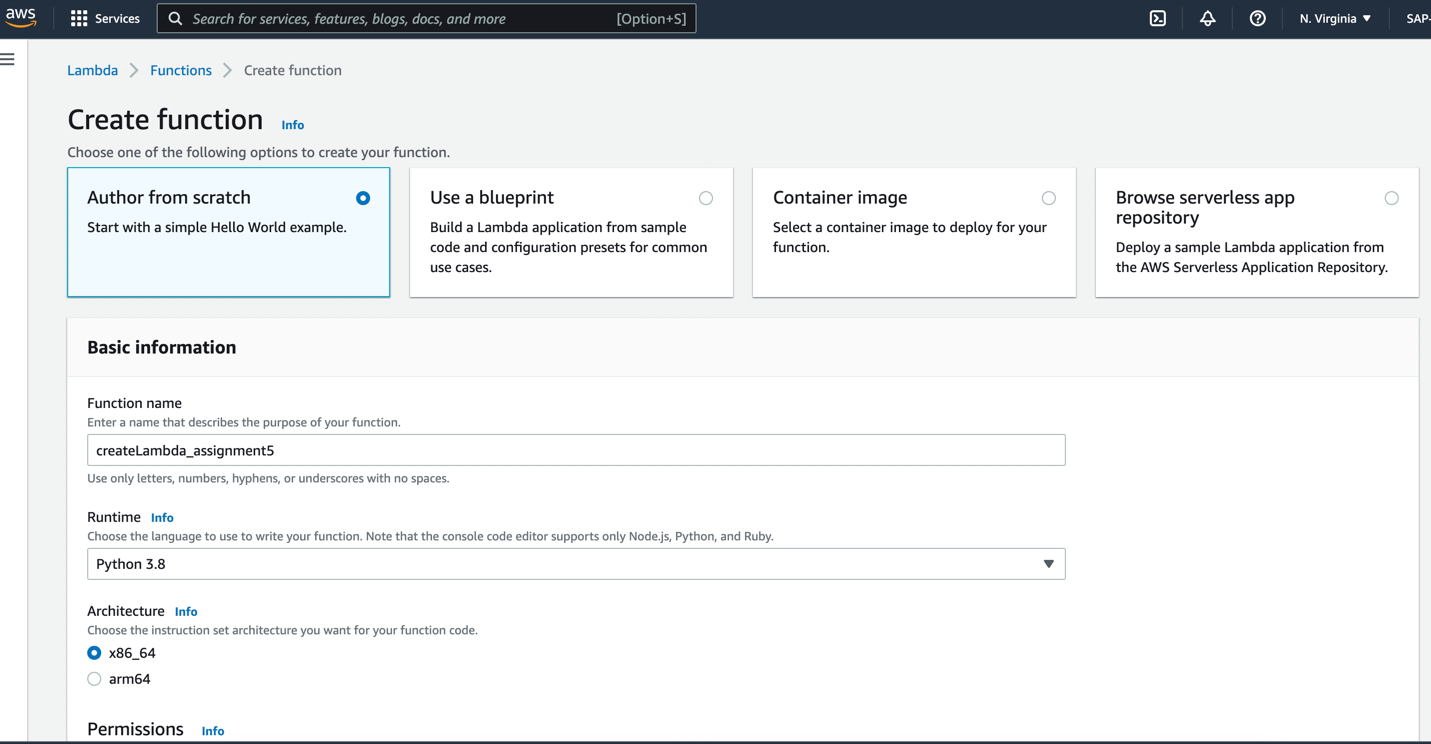
Step 4: complete the below configuration for Lambda creation –

1. **Function name** – Give the name of function as we gave ‘createLambda\_assignment\_5’,
2. **Runtime** – Lambda supports multiple languages through the use of [runtimes](https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-concepts.html#gettingstarted-concepts-runtime). Choose the language to write the code. There are multiple languages available as shown in below image(.NET, Go, Java, Python, Ruby, Node.js).



1. **Architecture** - The instruction set architecture of a Lambda function determines the type of computer processor that Lambda uses to run the function. Lambda provides a choice of instruction set architectures:
   1. arm64 – 64-bit ARM architecture, for the AWS Graviton2 processor.
   2. x86\_64 – 64-bit x86 architecture, for x86-based processors. x86\_64 is the default architecture.

Functions that use arm64 architecture offer lower cost per Gb/s compared with the equivalent function running on an x86-based CPU.

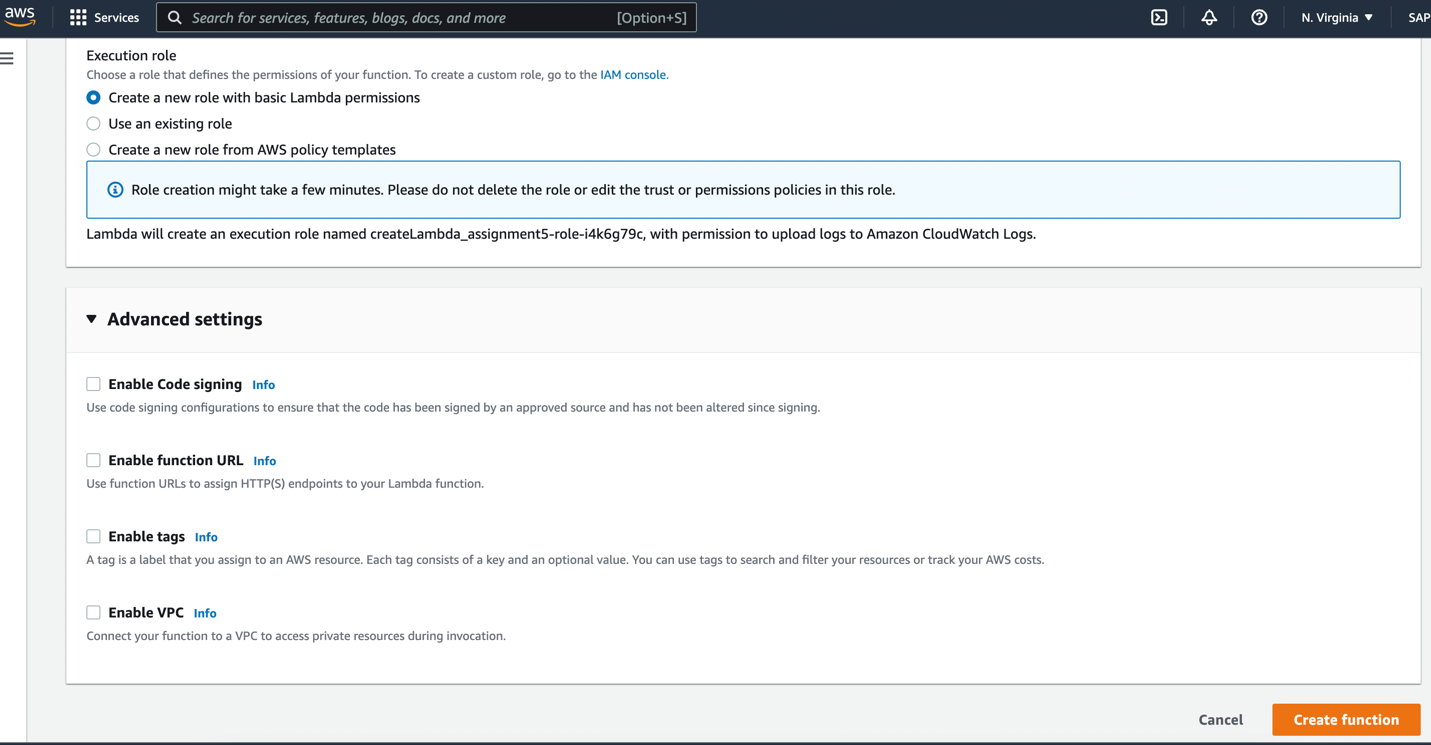


1. **Permissions** - An execution role gives your Lambda function permission to upload logs and access other AWS services. We can add policies to the execution role to give it access to downstream resources, such as DynamoDB tables. Below are the available options for role:
   1. Create a new role with basic Lambda permissions- A new role will be created for this Lambda and will be assigned to it.
   2. Use an existing role – We can attach role which has been created previously and exist in account.
   3. Create a new role from AWS templates – AWS provided templates can be use.
2. **Advanced settings**

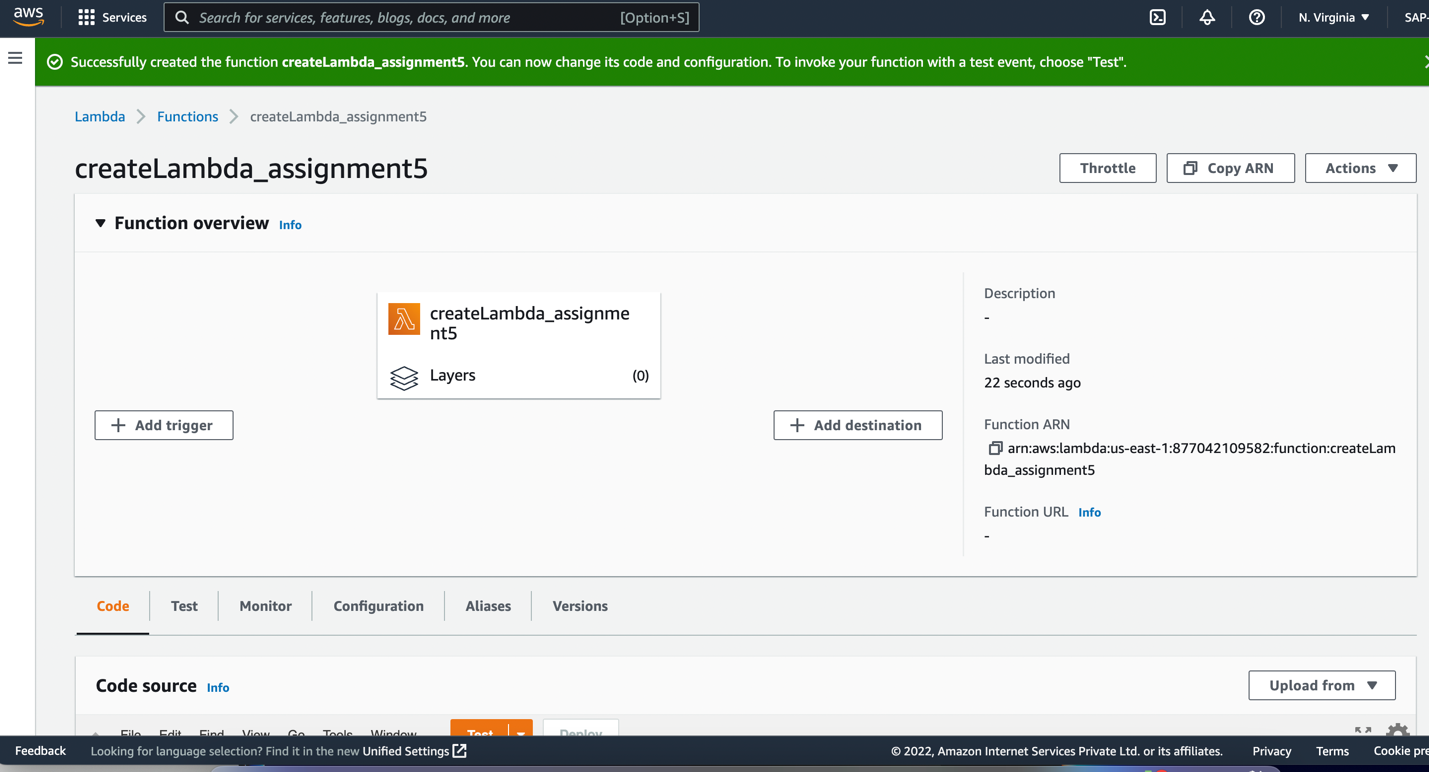
For basic lambda we haven’t used any option available in advance settings but below is short description-

* 1. Enable Code signing - With code signing, you can ensure that the code has been signed by an approved source and has not been altered since signing, and that the code signature has not expired or been revoked. A code signing configuration defines the allowed signing profiles and signature validation policy for the function.
  2. Enable function URL - A function URL is a dedicated HTTP(S) endpoint for your function. When your function URL is configured, you can use it to invoke your function through a browser, curl, Postman, or any HTTP client.
  3. Enable tags - Tags are optional key-value pairs that you can attach to your Lambda function. And categorize the services from functionality, hierarchy, cost wise etc,.
  4. Enable VPC - All Lambda functions run securely inside a default system-managed virtual private cloud (VPC). However, you can also configure your Lambda function to access resources in a custom VPC.

Step 5: Click on ‘Create function’

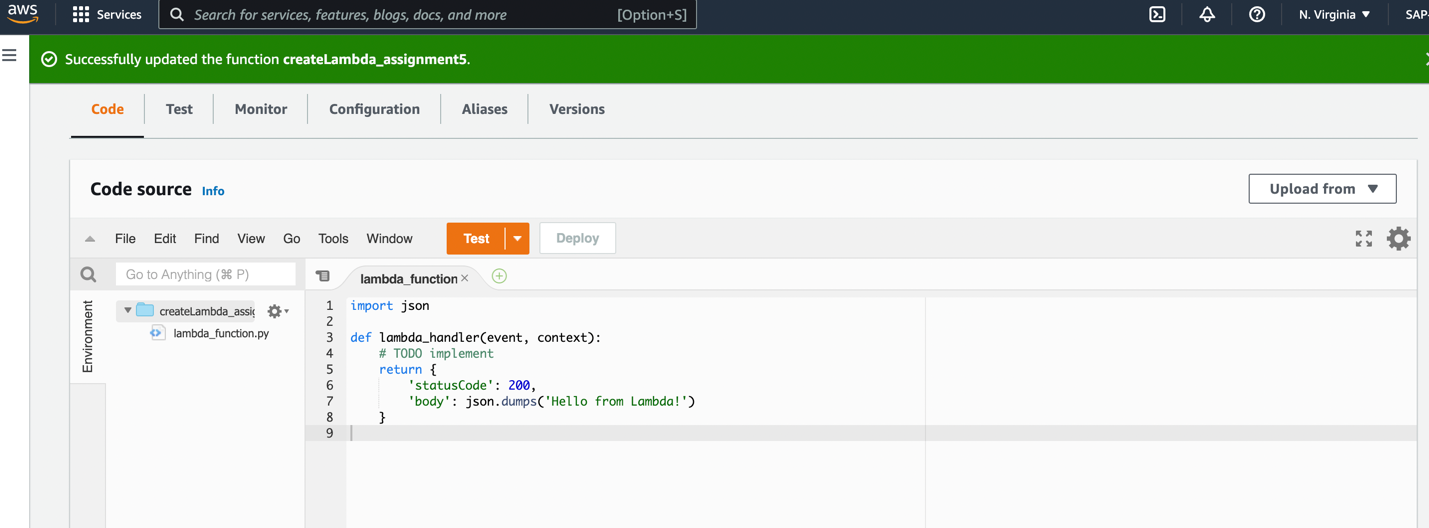


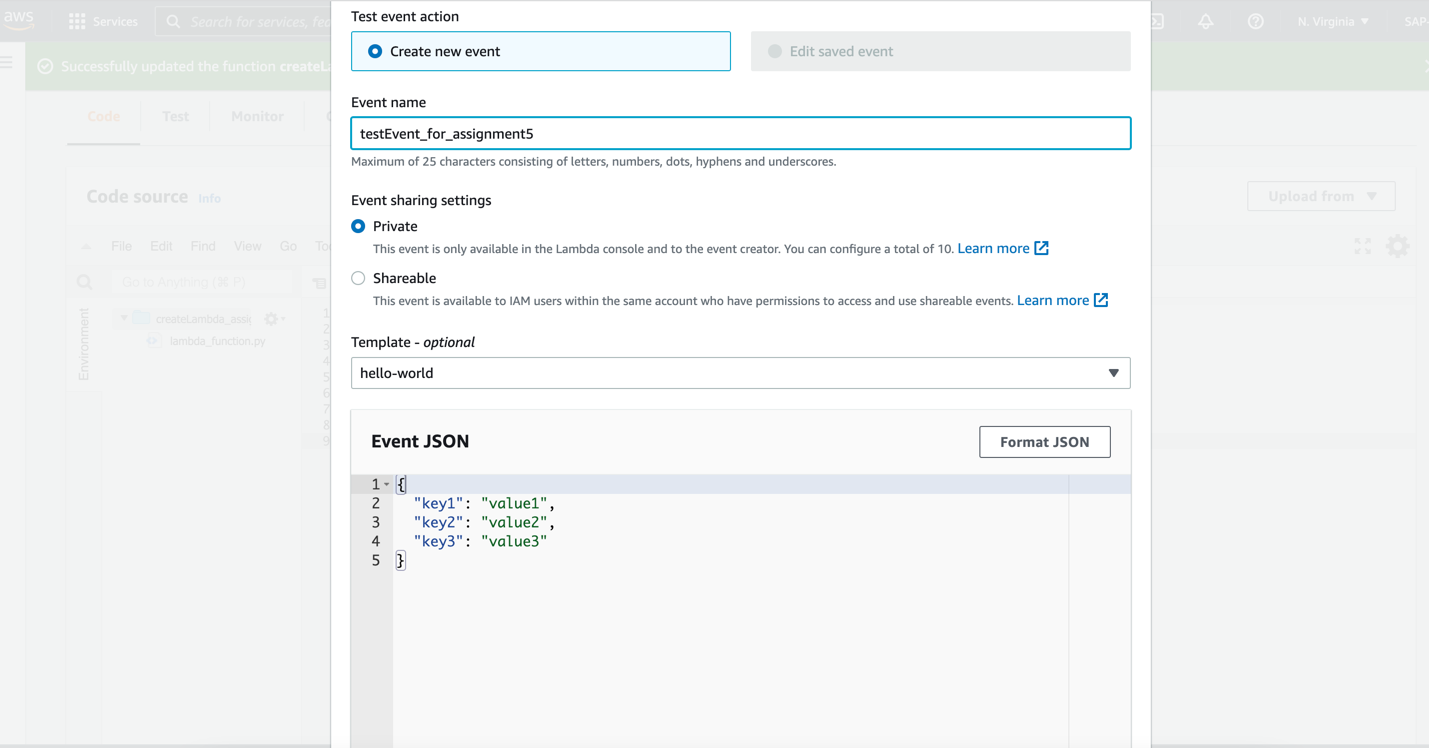
Upon successful lambda creation, a message for successful function creation will be visible as seen in below image.

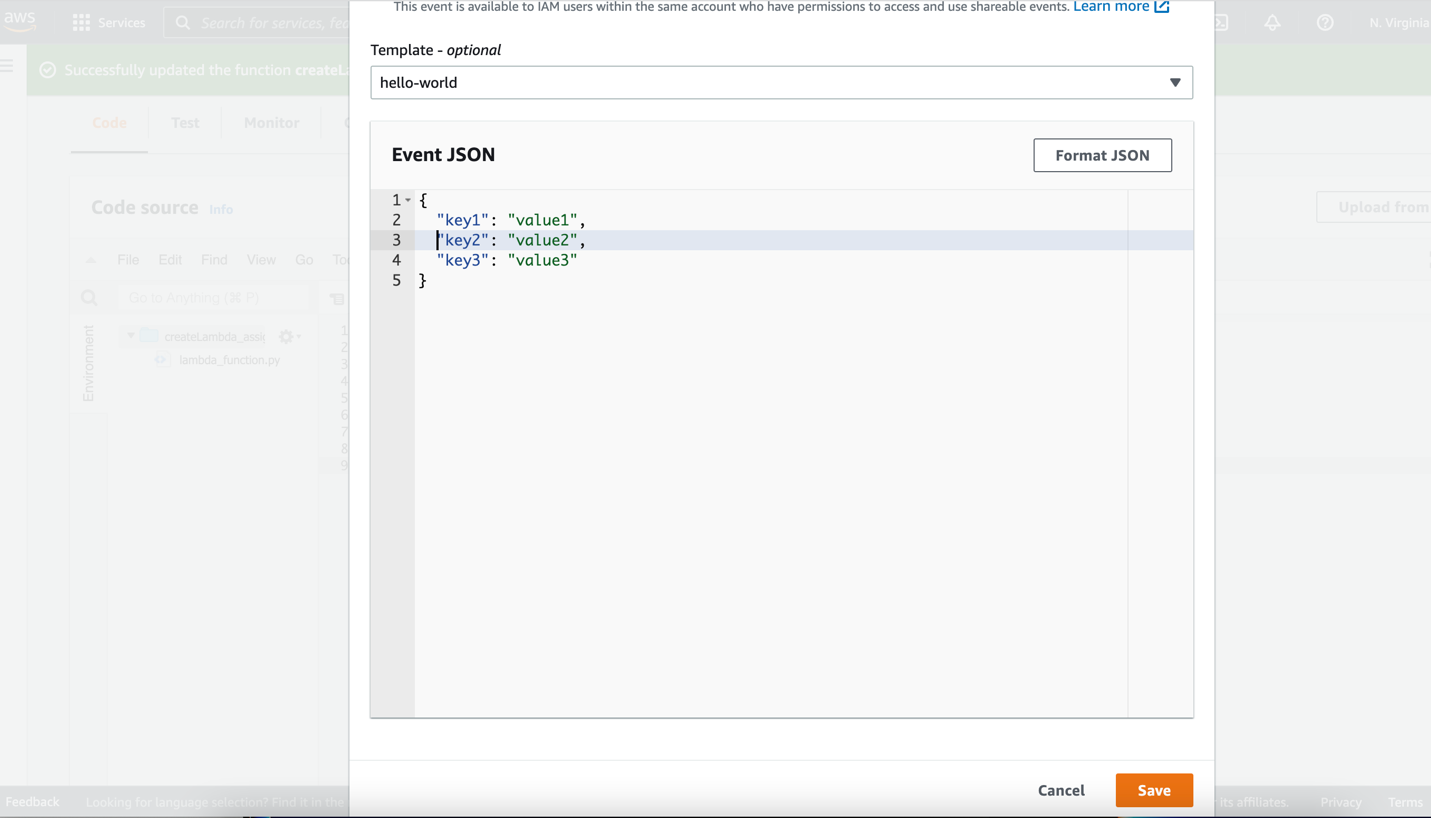


Step 6: Writing/Updating code in Lambda – Code Source:

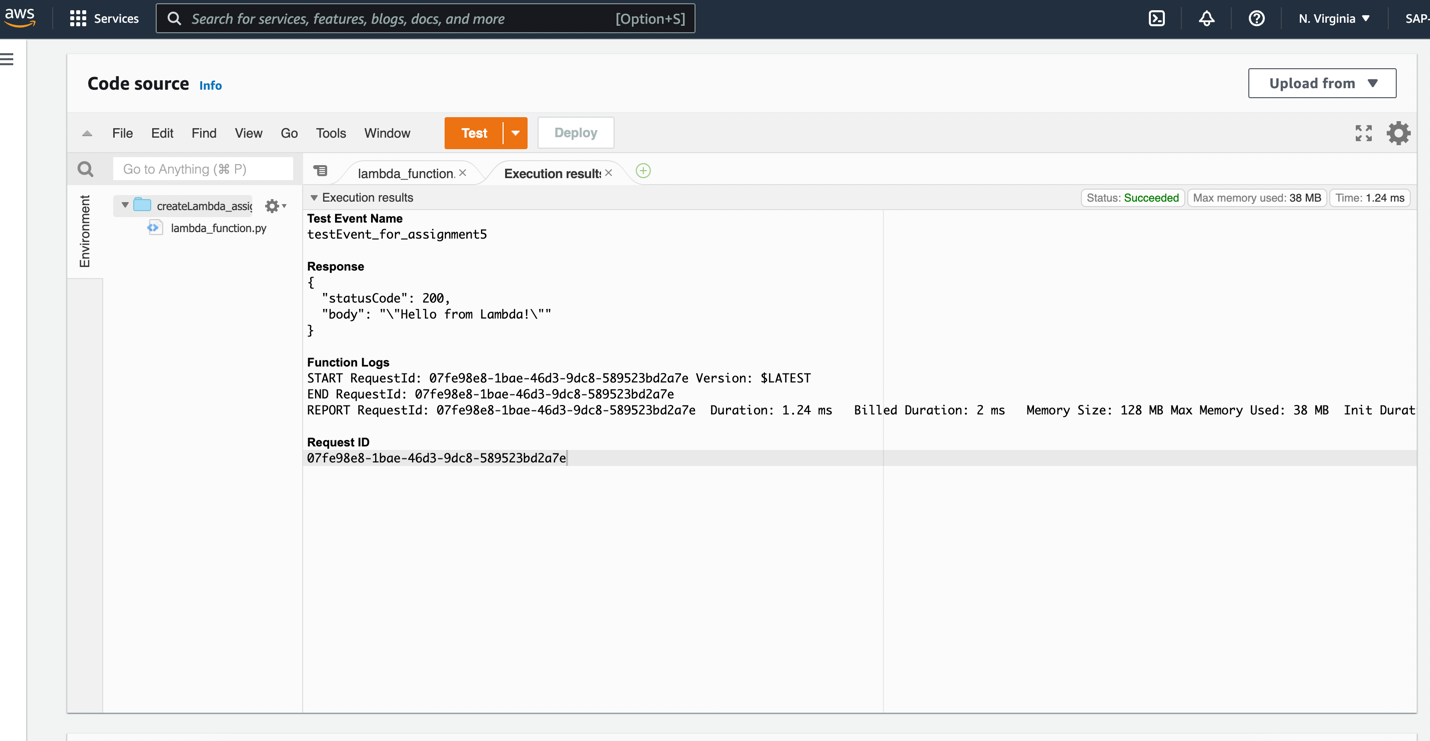
* 1. In Code source lambda\_function will be created by default, update/write the code in the function,
  2. Go to ‘File’ menu and save code.
  3. once code will be saved, deploy option will be enabled, click on deploy.
  4. To test the written code, click on test and configure the test event with required value in JSON format to pass to function code.
  5. Click on ‘Save’ to save the test event.







Click on ‘Test’ after test event configuration code will execute, and a new tab will be open with execution result.



Lambda configuration can be checked in below Code Source section ‘Code properties’, ‘Runtime settings.’

