AWS Lambda

Introduction

- It is a compute service where you can upload your code to AWS Lambda and the service can run the code on your behalf using AWS infrastructure
- ***It is Serverless Compute***: Run code without managing servers, code only runs when it needs to run (event driven)
- Everything is dynamically auto-scaled so if you have 1 user or 1 billion you pay for usage.
- Lambda handles: Capacity, Scaling, Deployment, Monitoring, Logging, Web Service frontend, Security Patching.

Some Use Cases:

- Send Customized messages to different users (SNS + Lambda)
- Send an offer when a user lives run out of lives in my game (Cognito + Lambda + SNS)
- Transform the records in a click stream or an IoT data stream (Kinesis + Lambda)
- Take an action on a github pull (Github + SNS + Lambda)
- Scan files to added to an s3 bucket and index them (S3 + Lambda + DynamoDB)
- Audit all AWS API usage (CloudTrail + S3 + Lambda)

Terminology

Term	**Description**
Lambda	lets you run code without provisioning or managing servers.
blueprint	Sample configurations of event sources and Lambda functions.
Trigger	adding triggers for executing functions w.r.t schedule / condition
Alias	Points to the specified Lambda function version
Version	allows you to better manage your in-production Lambda function code by enabling you to publish one or more versions of your Lambda function.
ARN	Amazon Resource Name. Unique and immutable (it can't be changed) Qualified ARN -The function ARN with the version suffix. Unqualified ARN – The function ARN without the version suffix.

Objectives

- Creating Hello world function **(Inline / Zip)**
- Managing Versioning Using the AWS Management Console
- Adding Triggers to Lambda function

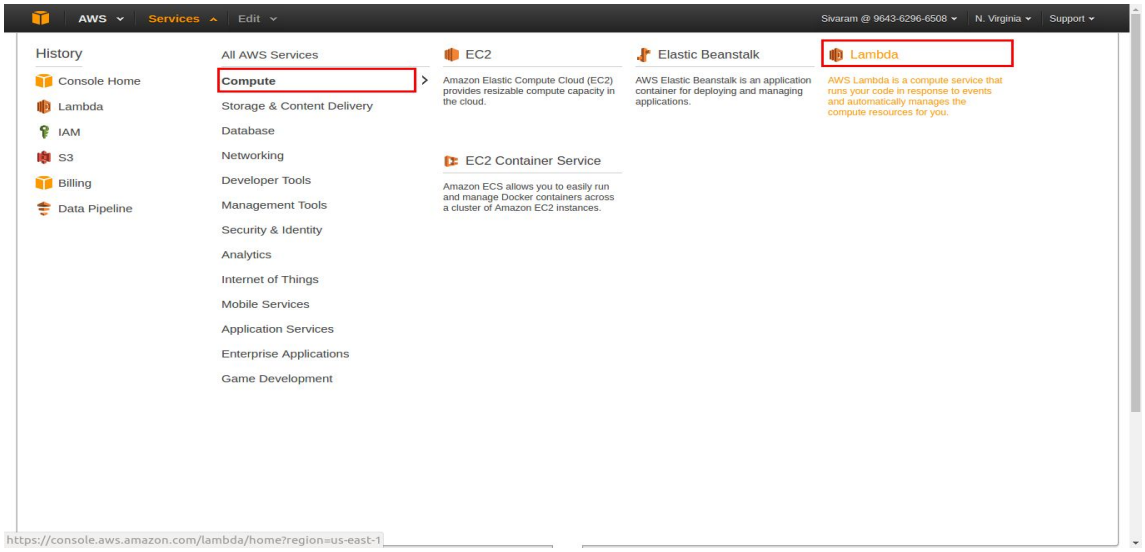
Pre-Requisites

- AWS Account
- Basic understand on AWS Services and Practical knowledge on AWS CLI / AWS Console

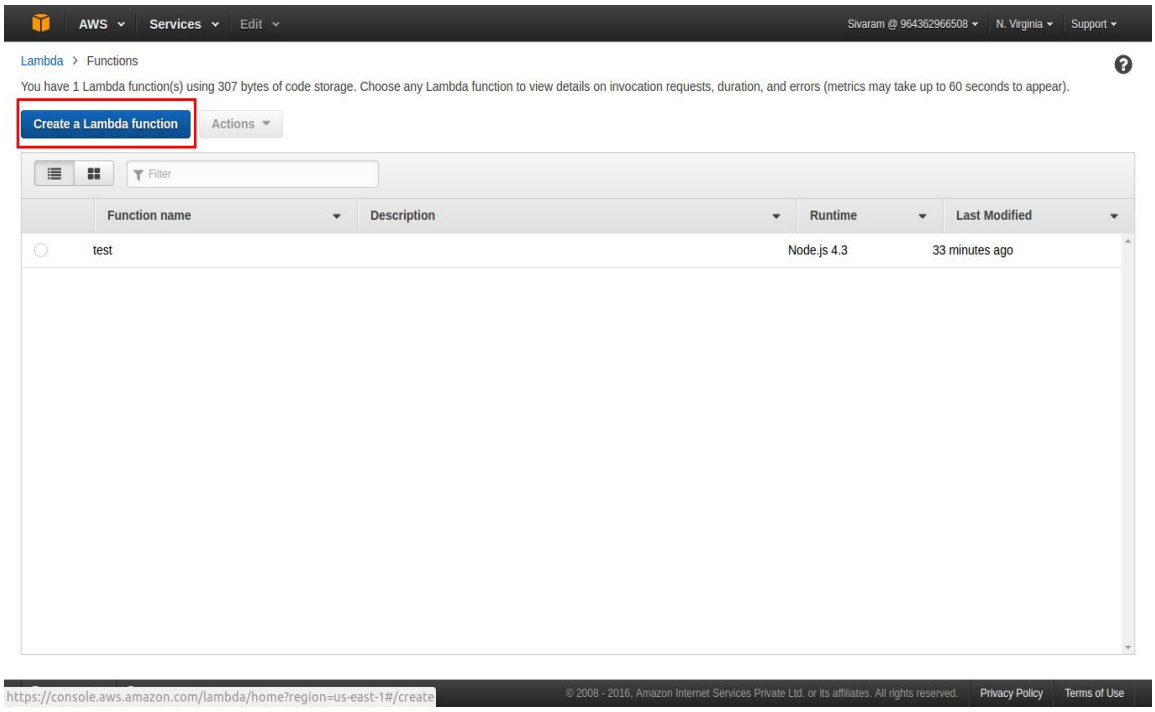
Instructions

Creating Hello world function

1. Go to Services -> Select Compute -> Click on Lambda

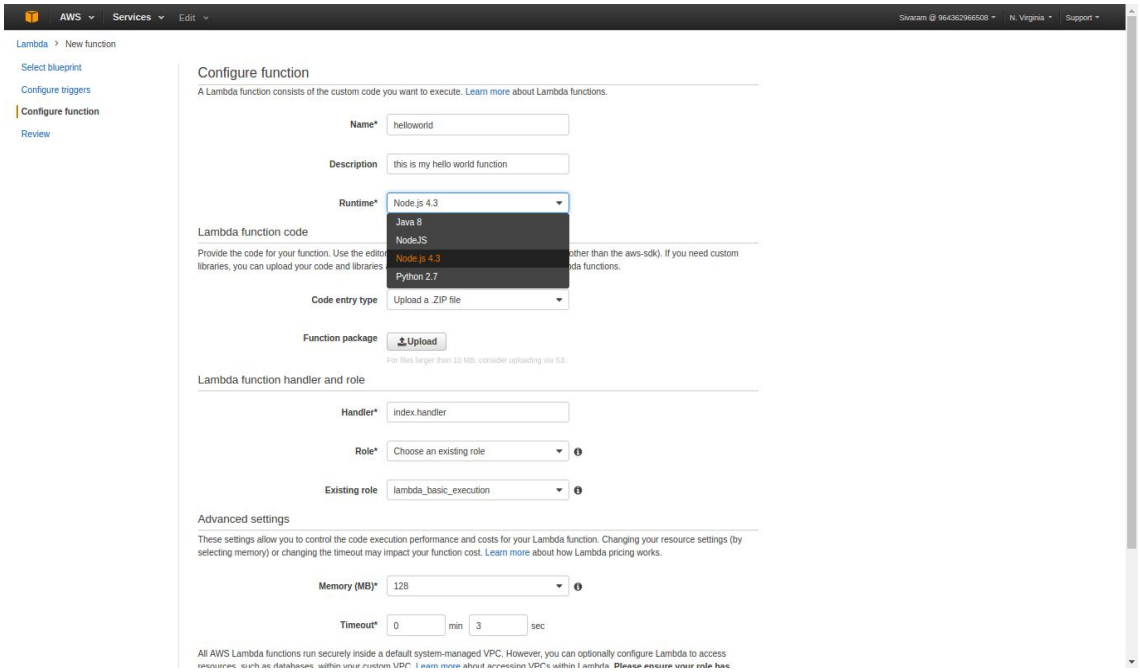


2. Click on "**Create a Lambda function**"

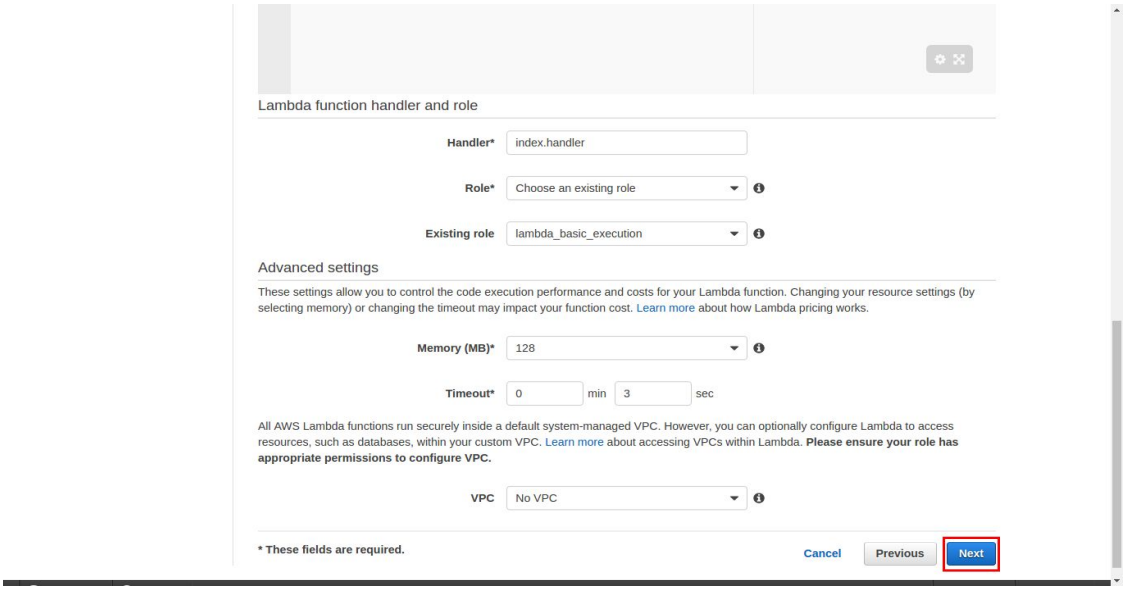


3. Click on **Configure function**

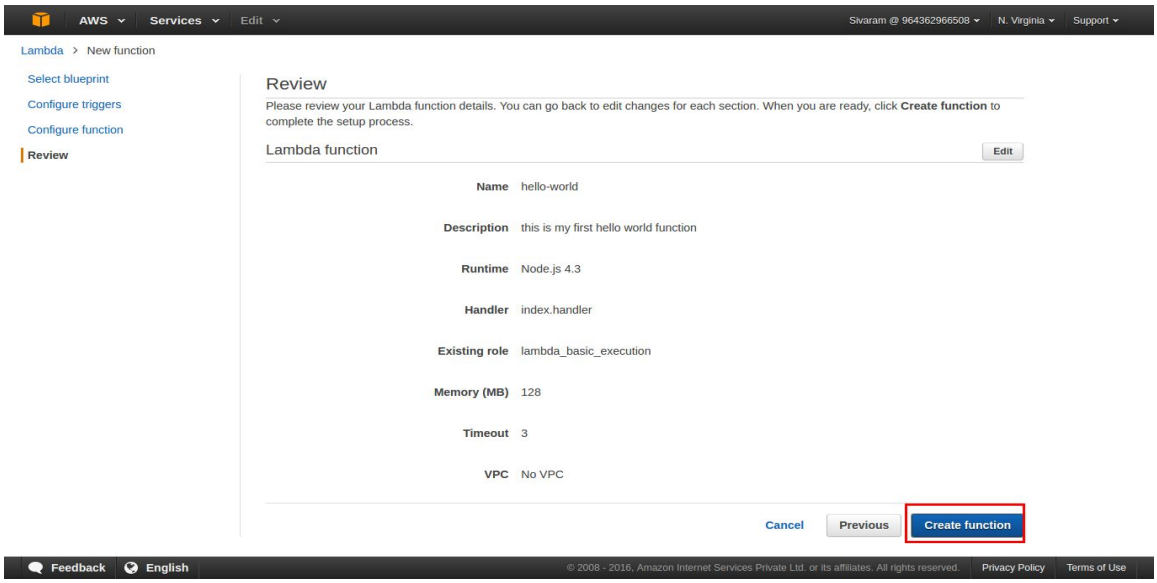
a. Configure function: Give the name of function, description and Runtime variable(Node.js / Java / Python)
b. Lambda function Code: Select Code entry type as either inline / **Upload a .ZIP file** / Upload a file from s3
c. Select "**Upload a .zip file"** from dropdown and upload the hello-world.zip from pc
d. Lambda function handler and role: give the handler name select the role for the Lambda function
e. Advance Settings: Memory: 128 MB, Timeout: 3 Sec



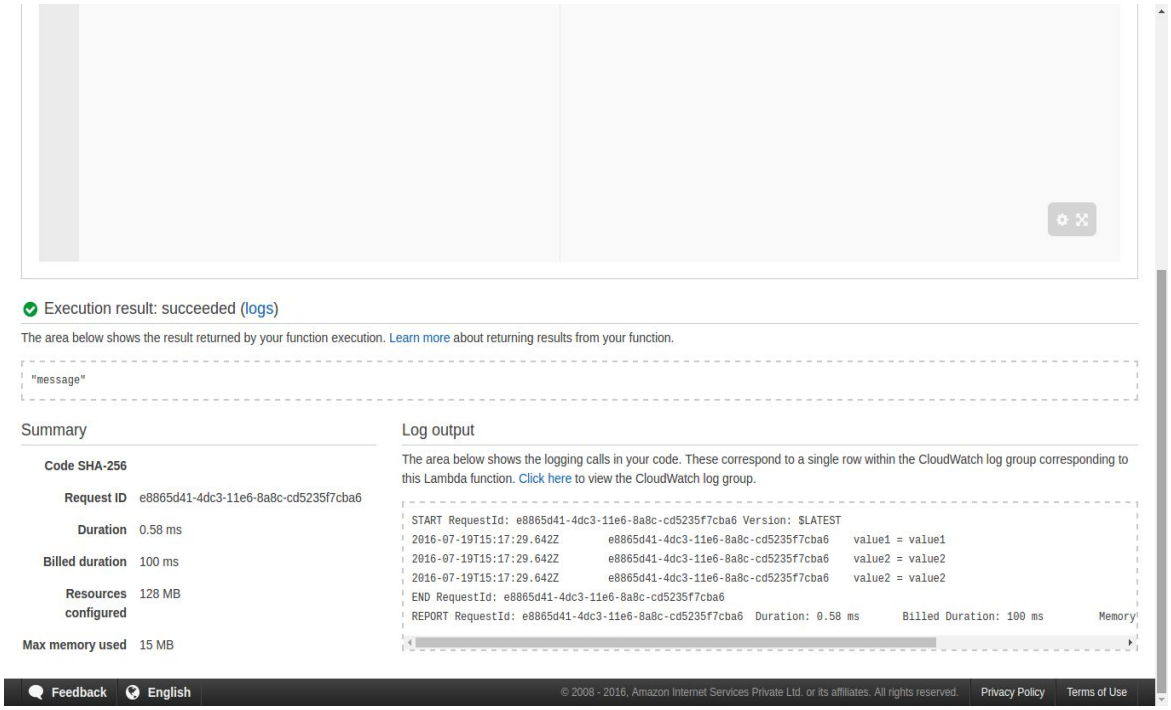
4. Click on **Next**



5. Review the function details and click on "**Create function**"

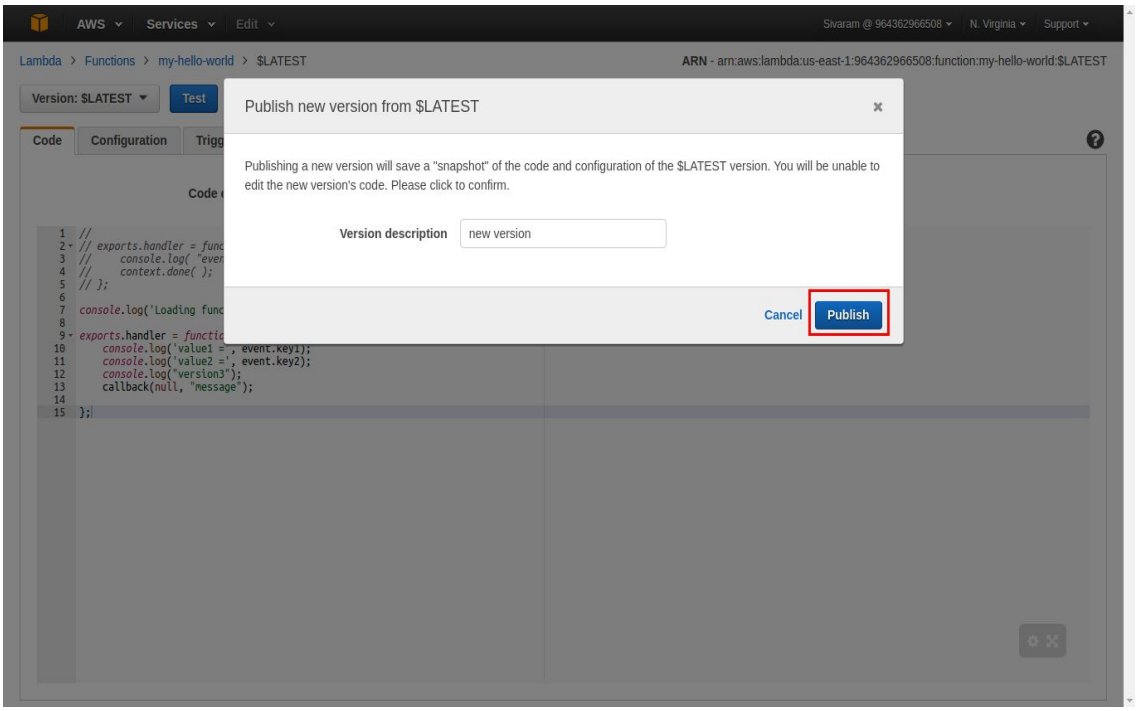


6. Click on "**Test**" to run the function and Check the execution results.

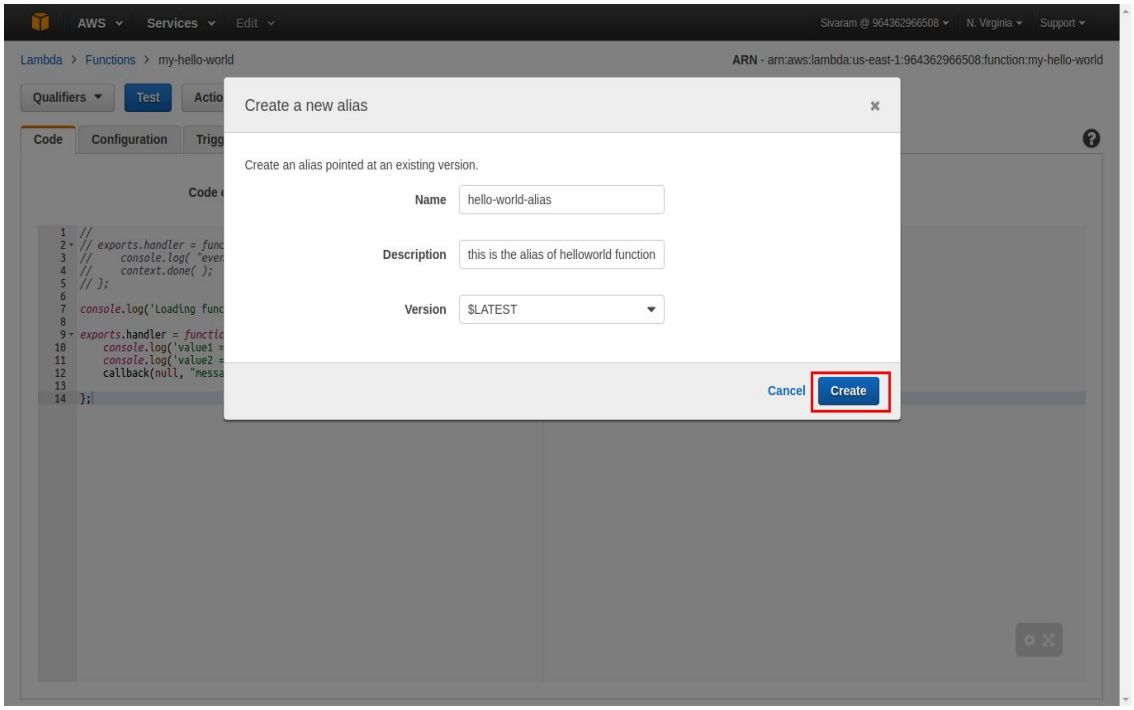


Managing Versioning Using the AWS Management Console

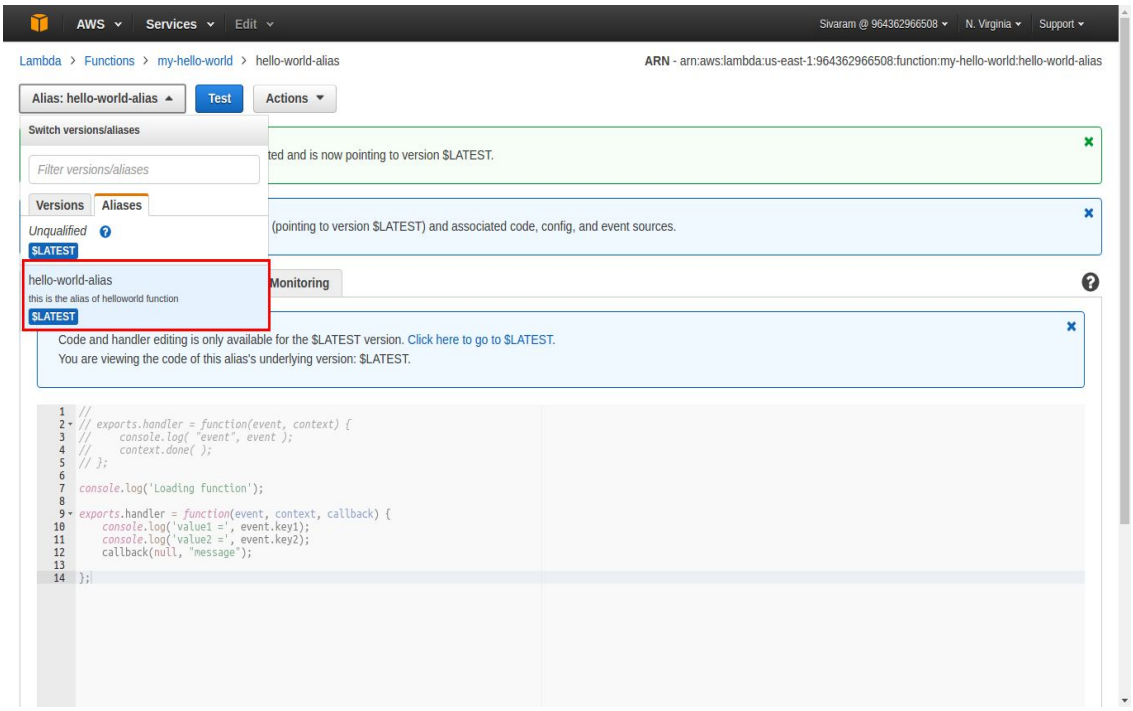
1. Select the existing lambda function and choose **"Qualifiers"**
2. It has two options as Versions: **$LATEST and** Aliases
3. Modify the code with any changes and Choose Actions and "**Publish new Version**"



4. It creates the Version(s)1 (immutable)
5. And also Choose Actions and Click on "**Create Alias**" (and map to specific version)

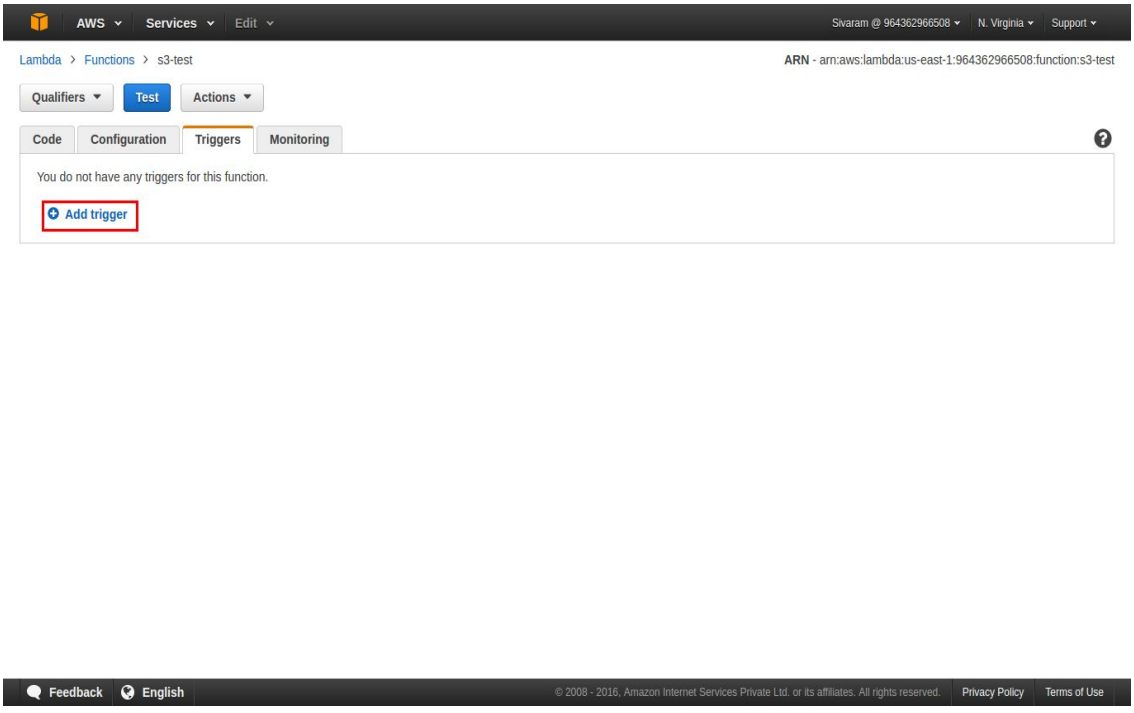


6. The below screen displays the existing versions and alias

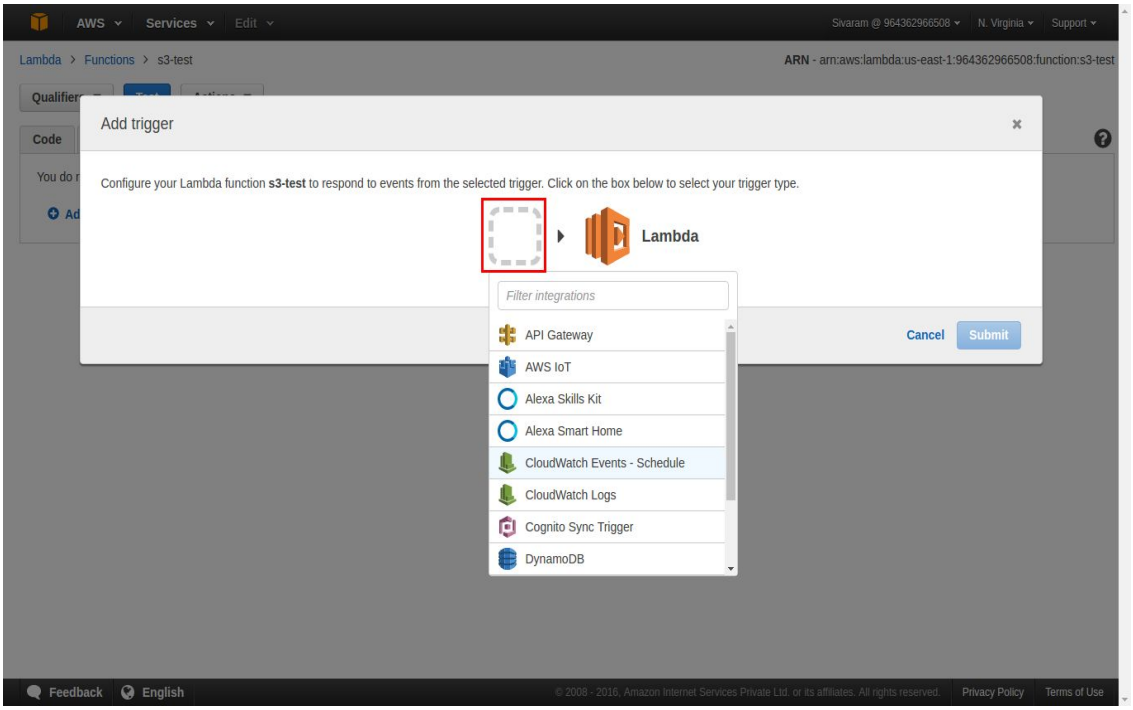


Adding Triggers to Lambda function

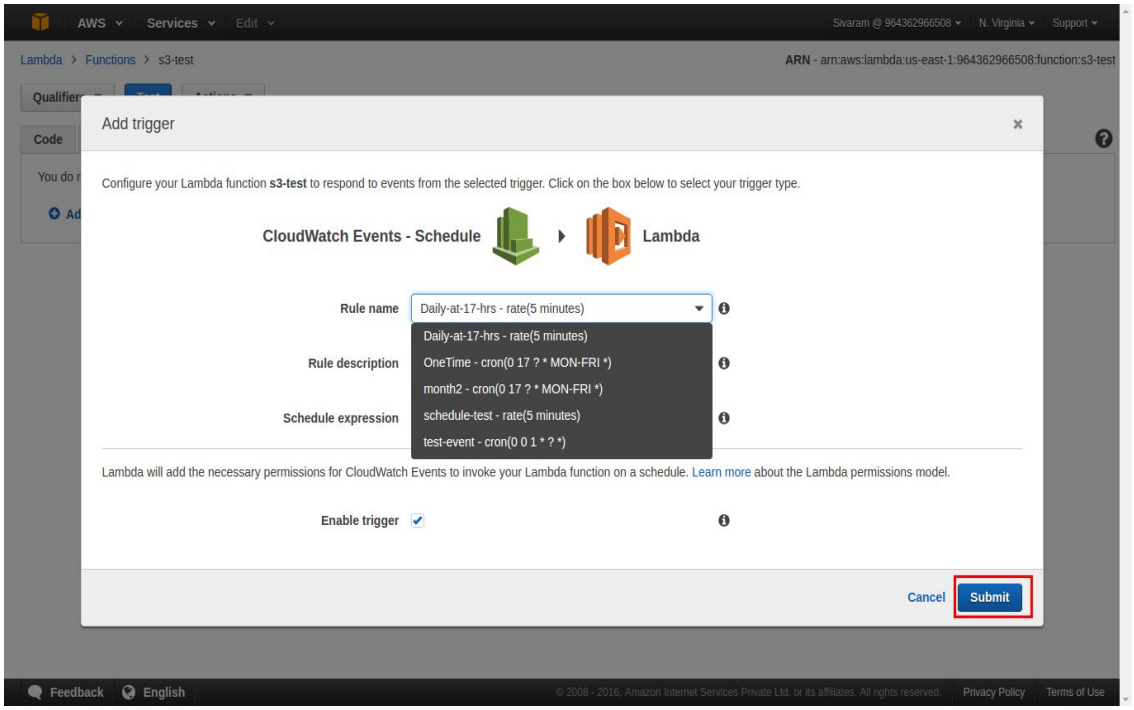
1. Double Click on any Lambda function
2. Select the 'Triggers' tab and Click on "**Add trigger"**



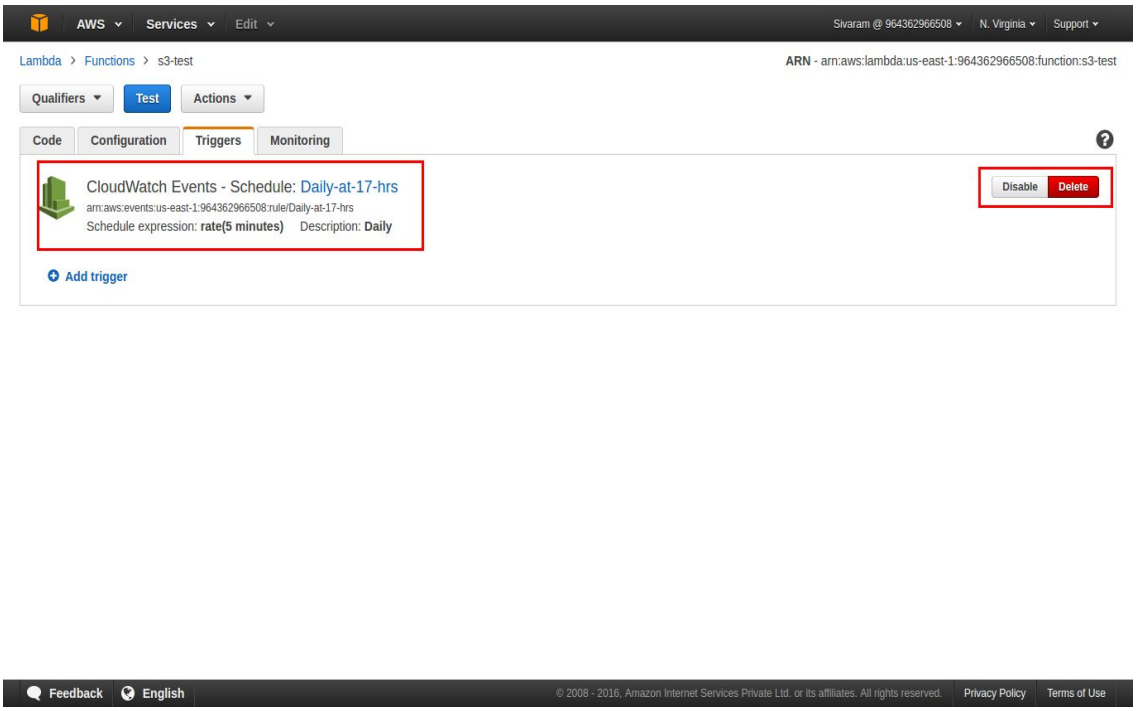
3. Configure the event in below screen



4. Select any integration (event), give the Rule name, schedule expression and schedule and click on Submit



5. Check the created Trigger



Note: We can run the Lambda function by adding triggers, we need to disable, when we have multiple versions, because triggers run automatically in backend. We can monitor the lambda function through cloud-watch.

Sample function Lambda function in Node.js

Helloworld.js
 	
exports.handler = function(event, context, callback) {
  console.log('value1 =', event.key1);
  console.log('value2 =', event.key2);
  console.log('value3 =', event.key3);
  callback(null, "message");
};
 	
Handler: helloworld.handler
	node-lambda
	(https://www.npmjs.com/package/node-lambda)
	npm install -g node-lambda // Install
	
	node-lambda setup // to setup lambda app in local env
	node-lambda run // to run lambda app in local env
	node-lambda package // to zip the package
	node-lambda deploy // to deploy the zip to remote (AWS)

References:

- http://docs.aws.amazon.com/lambda/latest/dg/use-cases.html
- https://github.com/awslabs/aws-lambda-zombie-workshop
- https://github.com/awslabs/aws-lambda-rdbms-integration

***** **Thank You** ******