National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Divya Balasaheb Thorat
Student ID: x18191878

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Divya Balasaheb Thorat
Student ID:	x18191878
Programme:	Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	17/08/2020
Project Title:	Configuration Manual
Word Count:	1000
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	17th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Divya Balasaheb Thorat
x18191878

1 Introduction

1.1 Motivation of the document

This presented manual describes requirement, deployment instruction, and validation for MapReduce application with a genetic algorithm on the serverless platform, according to National College of Ireland's project module handbook. The purpose of the configuration manual to define the prior requirement to implement MapReduce application with a genetic algorithm on a serverless platform.

1.2 Config Module Structure

Section	**Purpose**
Review	This section gives the information about implementation summary and amazon services required to implement the research question.
Configuration required for development	To implement the research question this model describes the setup description.
Deployment for solution	This section describes the implementing procedure of deployment.
Validations	This section describes the experiment we have performed to validate the research question

2 Basic Information

2.1 Objective

The research question aims to reduce the total completion time of map-reduce application on serverless infrastructure. lambda function has better scalability and low cold start delay, a combination of slow and fast storage gives a better performance along with low cost, so we have used AWS services Pu (Feb 2019).

2.2 Implementation Summary

This model shows the implementation summary of project.
- Initially, the s3 bucket is use to store the static and dynamic data, coordinator lambda function use for making the connection between the mapper phase and reducer phase to

process the data. To fetch data from s3 to lambda function, we need to give some permission to the IAM role. The IAM role permissions include S3 full access, Cloud watch full access, s3 read-only access, VPC full access, elastic file system full access, lambda role, lambda VPC access executive role, and cloud watch events full access. The S3 trigger event is use to trigger the lambda function when data is inserted into a bucket.
- In this project, the coordinator function takes the input and distribute the data to two mapper functions with some conditions according to the data size.
- Mapper function is invoked from the coordinator function and discover good answer for the available task with the help of a genetic algorithm. Each mapper's data will store into Redis in-memory data storage to provide the excellent grained elasticity Pu (Feb 2019).
- Then the invocation of reducer is perform, and aggregated output is stored in another s3 bucket. To provide security for the applications, user processes the data with VPC by configuring the lambda function And creating the VPC endpoint for AWS S3 service.

2.3 Architecture prerequisite

This section explains the services required to implement the research question. we are using the AWS services as the lambda service provides the low latency and scalability Wang et al. (2018).

2.3.1 Amazon Web Services console account

In this it is necessity to create the console account to access the services.

2.3.2 AWS Lambda

This service enables the user to run the code without provisioning and monitoring the servers. We just have to pay for compute the time of the service. It also provides continuous scaling whenever it will require.In this project, lambda service is used to create the coordinator, mapper, and reducer function. Coordinator function is used to distribute the data across the mapper function, each mapper function is used to find out the best output solution with the help of a genetic algorithm, and reducer function is used to aggregate the output of the mapper function. configuration done with the role, trigger event, network topology, security groups, and python language with the latest version 3.8 has been added at run time [1] Giménez-Alventosa et al. (2019).

2.3.3 AWS S3

It is an object storage service used by customers of all sizes and companies for storing and protecting any amount of data for multiple applications. It gives the 99.99999999999% of durability to store data for multiple application of companies. In this project, two s3 buckets need to create to store the Input and output data. Input buckets name is AWS-lambda-trigger1, and one trigger event is created so that as soon as the data is coming into an Input s3 bucket, it will invoke the coordinator lambda function for execution. The output bucket is used for storing the output data coming out of the redis function. [2]

[1] https://aws.amazon.com/lambda/
[2] https://aws.amazon.com/s3

2.3.4 AWS VPC

To access internet in AWS Lambda and elastic cache it is necessary to create the virtual private network. subnet, security group and route table are added in vpc[3].

2.3.5 AWS Cloud Watch

The AWS Cloud Watch service is used for analysing monitoring, error handling, analysing logs and so on. This service provides all the information required for executing Map-reduce tasks at a given time ([4].

2.3.6 AWS Elastic Cache

To store the intermediate data generated through mapper we have used the elastic cache for redis[5].

2.4 Run time Programming language

The run programming language used is Python 3.8 to create the functions. In this PyGAD is an open-source python 3 libraries for implementing the genetic algorithm. PyGAD provides assistance for distinct types of operators such as mutation, cross over, and parent selection. It allows a distinct type of problem is optimized using the genetic algorithm by the fitness function..

3 Configuration required for development

3.1 Code Repository

The zip file of the code is attach.

3.2 Required Run Time Programming Languages

The AWS lambda service gives platform for running code written in multiple language but we have used the python 3.8. The below packages has been imported for implementing the research question:

- boto3 - 1.10.28
- redis - 5.0.6
- csv -13.1.1
- random
- json

[3]https://aws.amazon.com/vpc/
[4]https://aws.amazon.com/cloudwatch)
[5]https://aws.amazon.com/elasticache

3.3 Creation of IAM role

Before creating Lambda function, it is necessary create the IAM role which has below policies attach to it as shown in figure 1. so, it help the lambda function to create the execution environment.

- AmazonElasticCacheFullAccess
- AmazonS3FullAccess
- CloudWatchFullAccess
- AmazonVPCFullAccess
- AWSLambdaVPCAccessExecutionRole
- AWSLambdaRole
- CloudWatchEventsFullAccess
- AWSLambdaENIManagementAccess

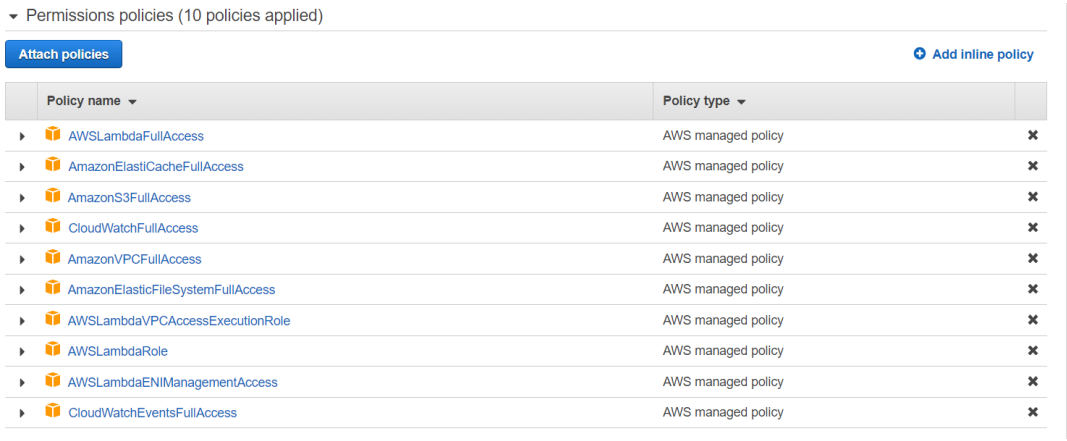

Figure 1: Policies attach to a IAM role

3.4 Creating VPC

we are creating VPC (server vpc) with 192.168.0.0/16 ipv4 CIDR block as shown in figure 2.

1. create public subnet with 192.168.10.0/24 v4 CIDR block as shown in figure 3
2. create private subnet with 192.168.20.0/24 ipv4 CIDR block as shown in figure 4.
3. create internet gateway to assign the public subnet shown in figure 5.
4. create NAT gateway assign to assign private subnet as shown in figure 6
5. create public route table and assign public subnet to it along with internet gateway as shown in figure 7.

6. create private route table and assign private subnet to it along with nat gateway and vpc end point as shown in figure 8

7. create vpc endpoint as shown in figure 9

8. create the security group which act as virtual firewall so it allows the incoming traffic and outgoing traffic. In this security group it allows the traffic from anywhere to Redis and other services with the help of HTTP, HTTPs, SSH, custom TCP, and All ICMP – Ipv4 protocol as shown in figure 10.

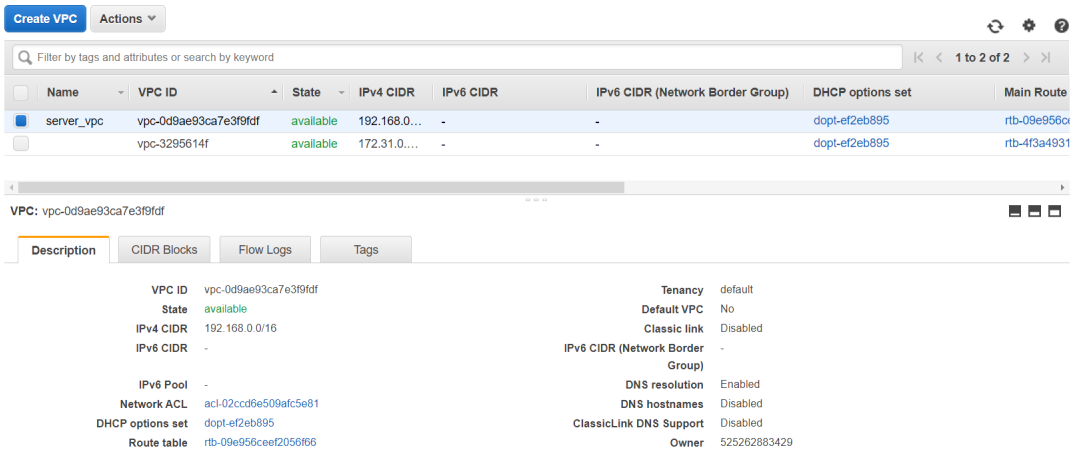

Figure 2: creation of a vpc

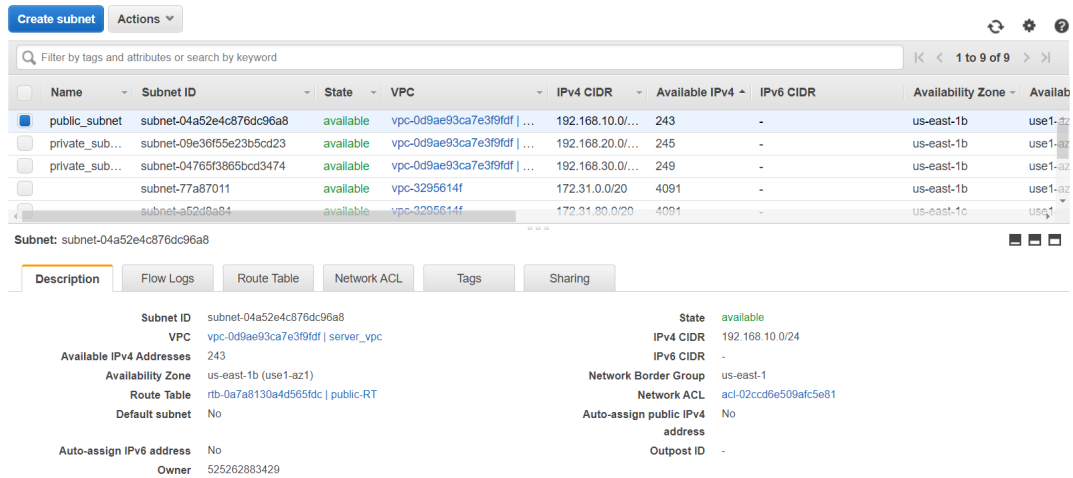

Figure 3: creation of a public subnet

3.5 create s3 bucket

creating the s3 bucket to store the incoming data 11.

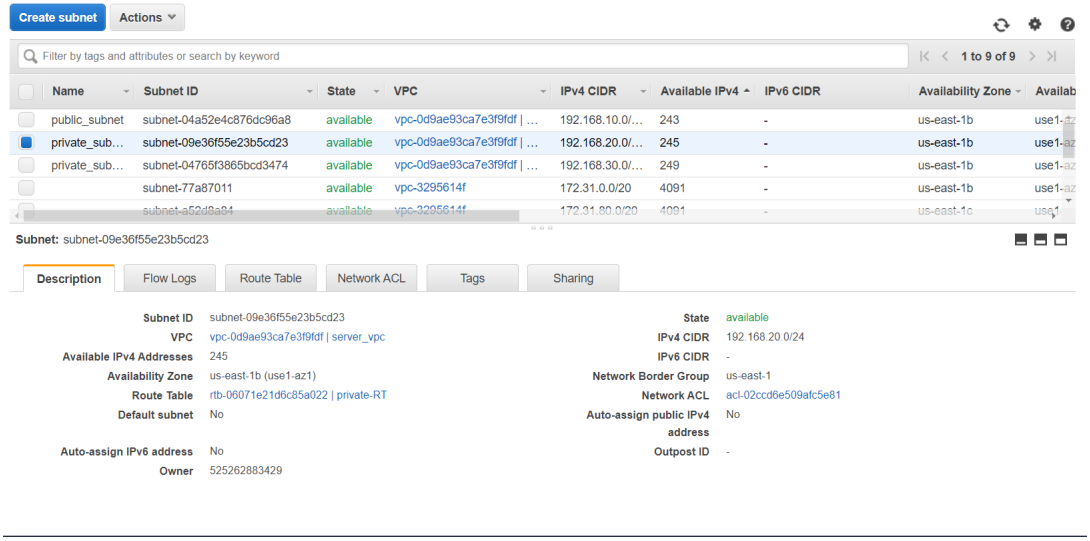

Figure 4: creation of a private subnet

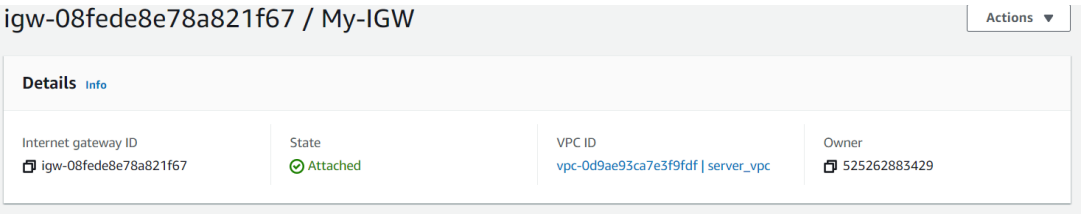

Figure 5: creation of a internet gateway

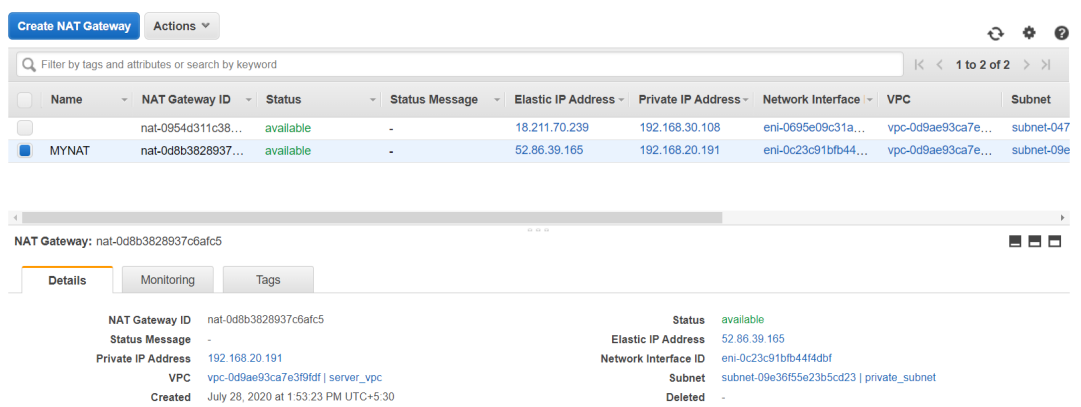

Figure 6: creation of a NAT gateway

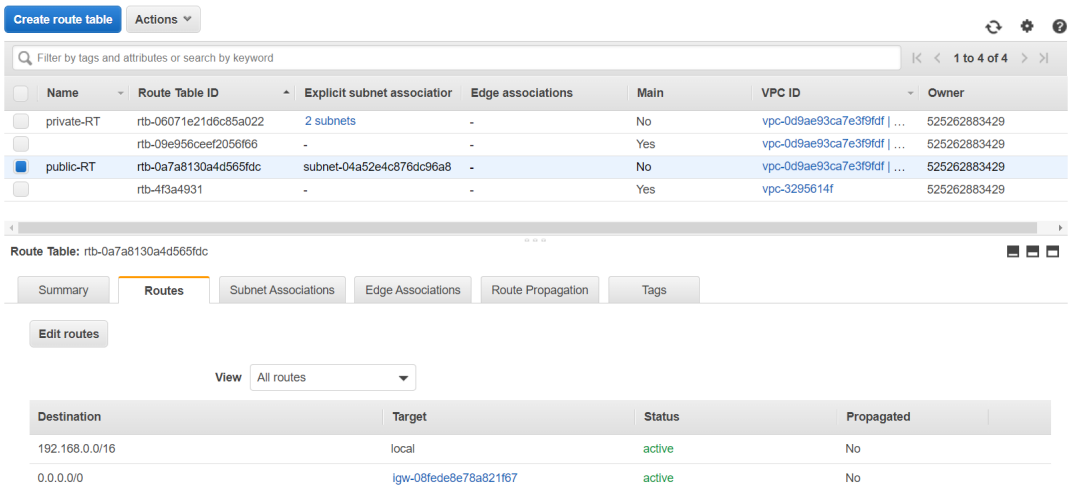

Figure 7: creation of a public route table

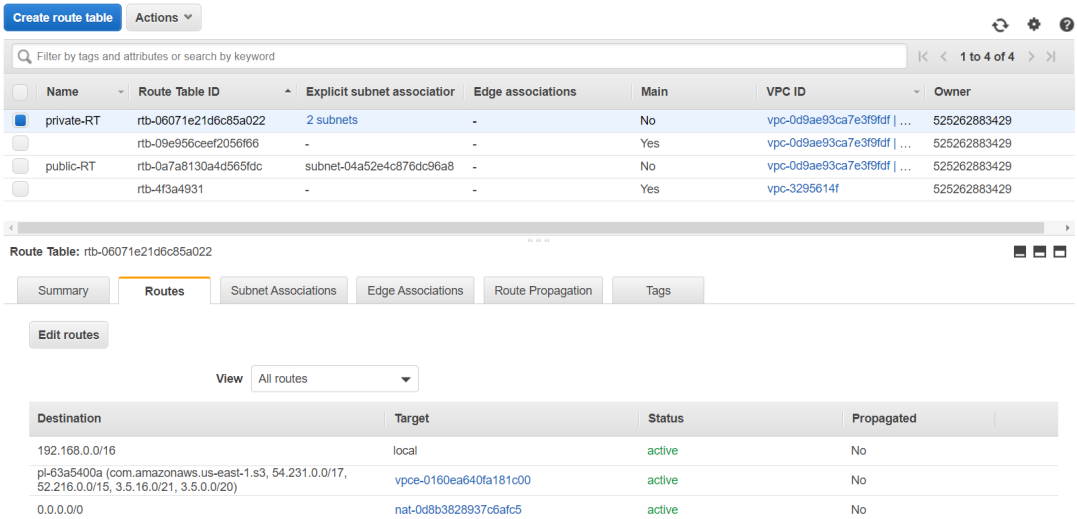

Figure 8: creation of a private route table

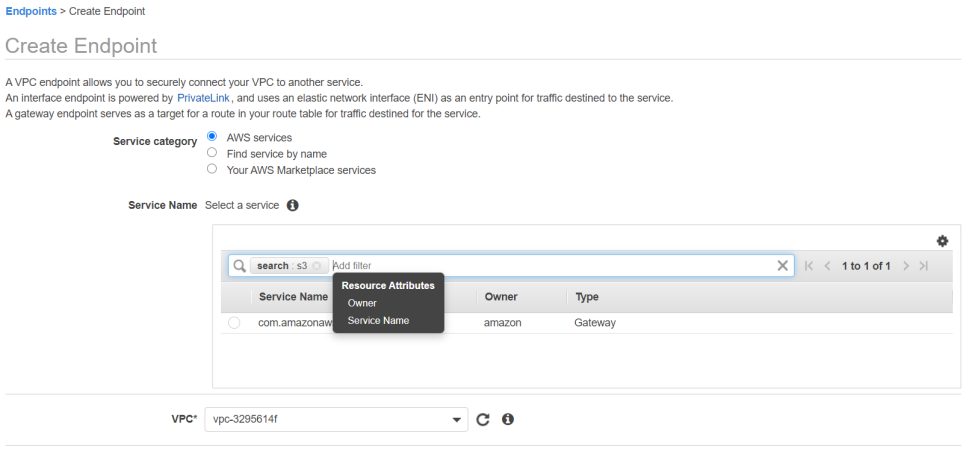

Figure 9: creation of a vpc endpoint

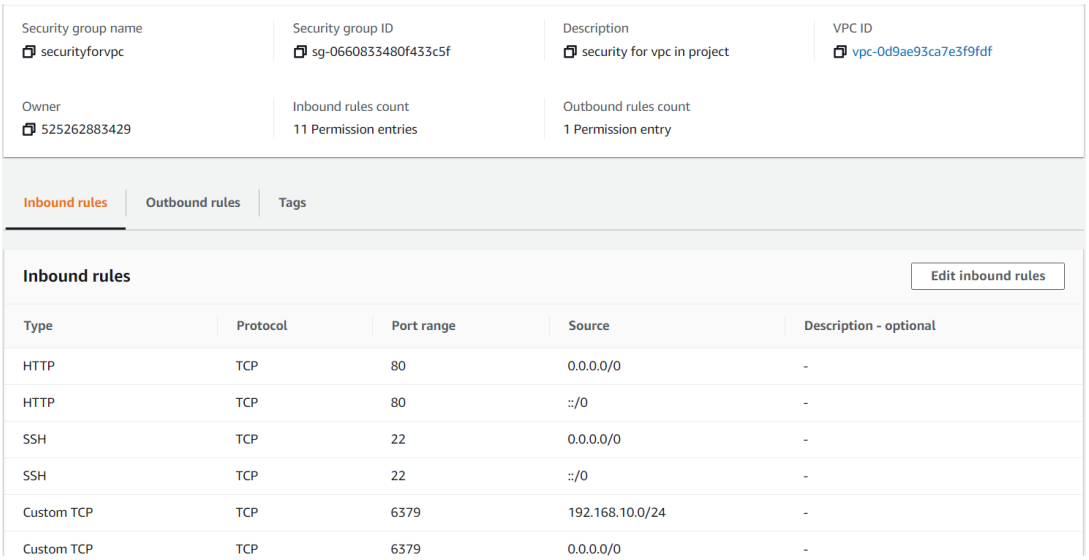

Figure 10: creation of a security group

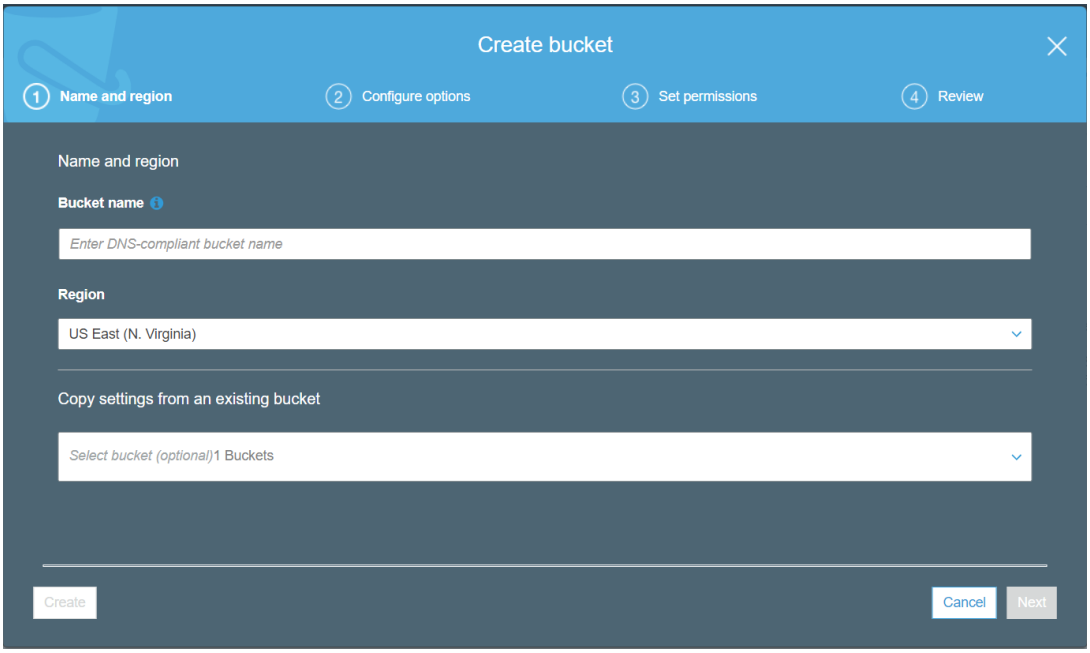

Figure 11: creation of a s3 bucket

3.6 Creation of Lambda function

After creating a IAM role, creating the lambda function for mapper, coordinator and reducer.

- To create lambda function go to AWS lambda and click on create a function as shown in figure 12.
- After selecting a create function now select the blueprint as shown in figure 13.
- now in a basic information section choose the function name, run-time language and permission i.e. IAM role with all policies that is required. as shown in figure 14]
- After creating the project lambda function now edit the vpc section and add the network configuration as shown in figure 15
- when there is input data in s3 bucket it should be notified to lambda function for that we have to create the s3 trigger event in lambda function 16

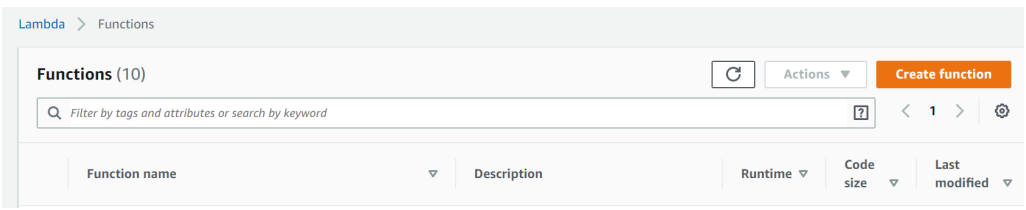

Figure 12: Creation of function

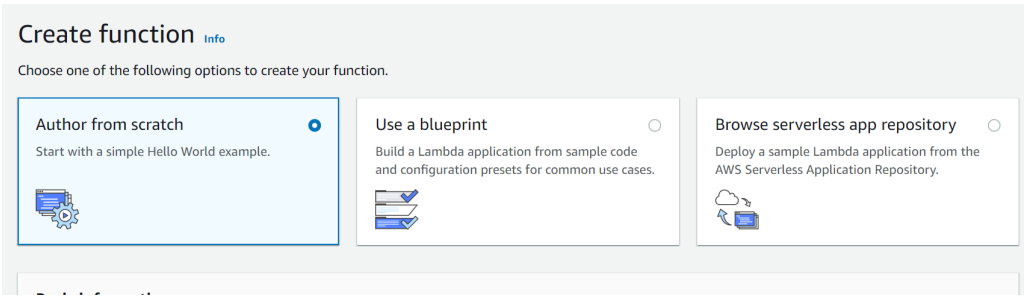

Figure 13: Selection of blueprint

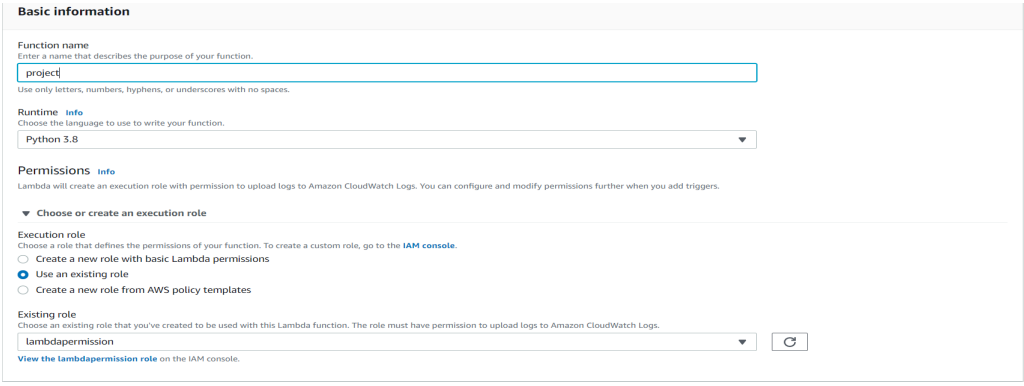

Figure 14: Configuration details

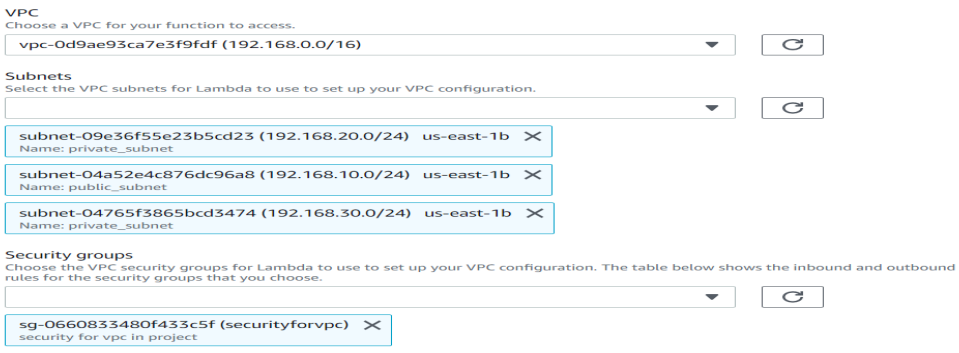

Figure 15: Network Configuration

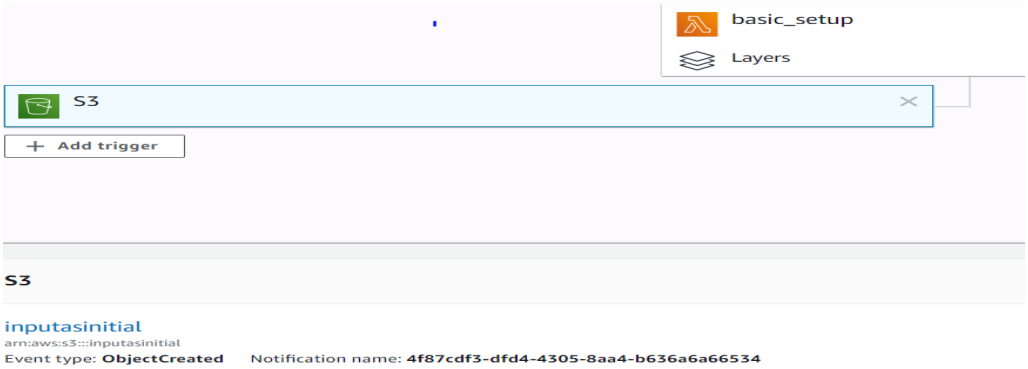

Figure 16: creation of s3 event trigger

3.7 Create Cache Memory

create the elastic cache memory for redis to store the intermediate data generated from mapper function.

1. select the cluster engine as redis as shown in figure 17
2. then configure a redis setting in which select the engine version, port, node type and number of replica as shown in figure 18
3. in advance setting need to select the subnet group, security group, backup and maintenance window as shown in figure19

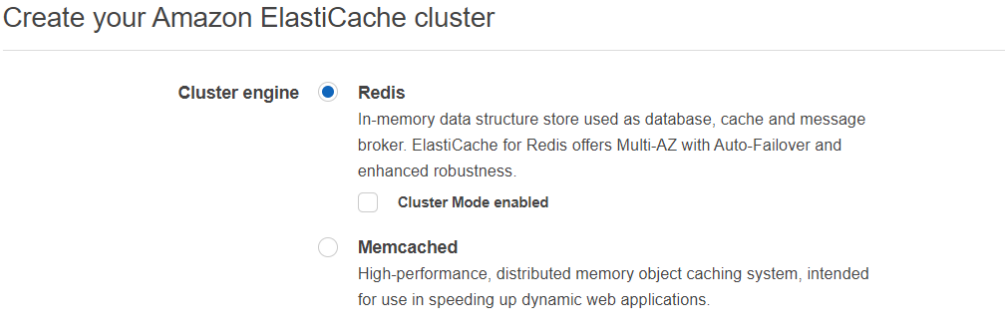

Figure 17: Redis Configuration

Figure 18: Redis setting

4 Validation

create a lambda function and inside that function configure a mapper,coordinator and a reducer function. now prepare the code and deploy it into a function using a upload zip

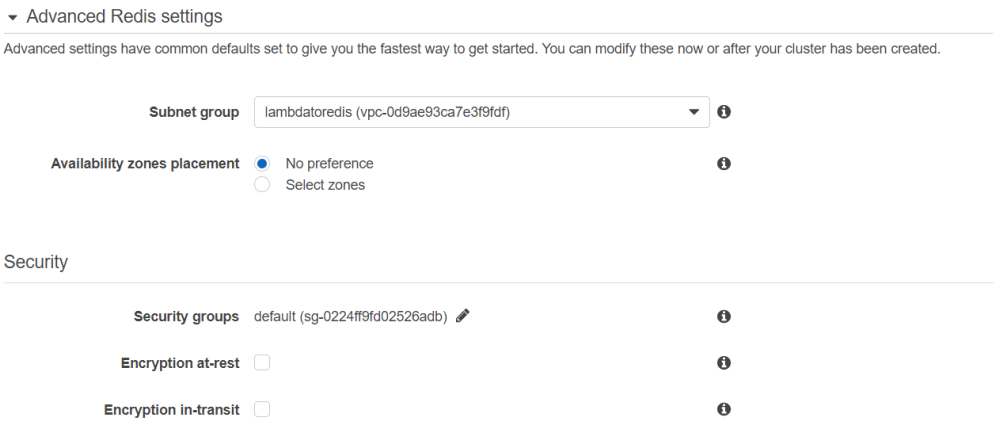

Figure 19: Advance redis setting

file option. Now configure the network configuration inside vpc section. after creating all the services now test the configuration. Testing of the configuration done in two ways as shown below.

- first way is to create test inside a function as shown in a figure 20. Run the test and the results are displayed in figure 21. An error is thrown when there is no correct configuration.
- second way is to add the input in s3 bucket as shown in figure 22. so whenever there is input at s3 bucket s3 event trigger is happen. If there is proper connection between s3 trigger event and lambda function then it will execute automatically, the output is displayed in cloud watch service as shown in figure 23.

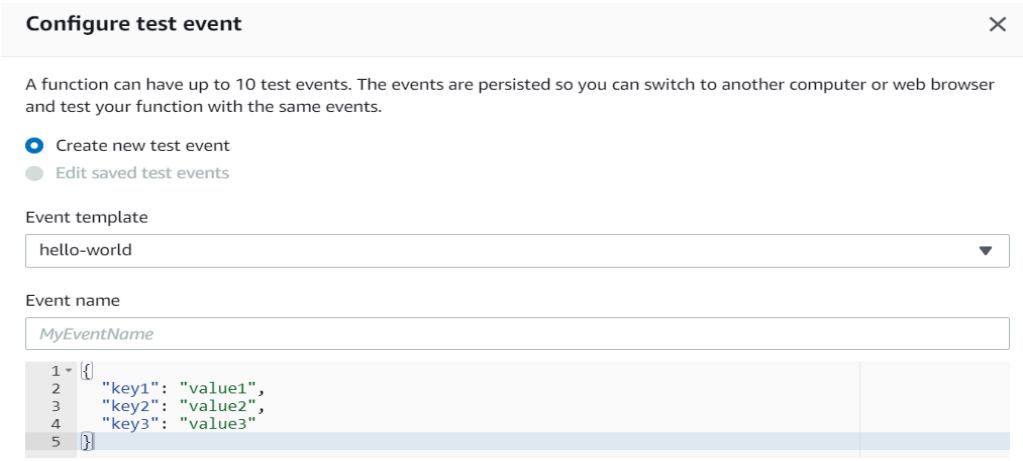

Figure 20: Creation of test event

Figure 21: Lambda output result

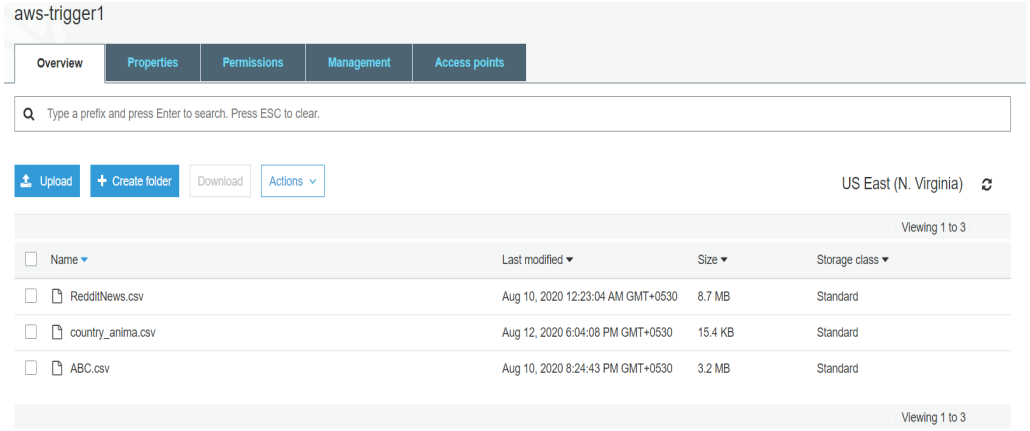

Figure 22: S3 Trigger event

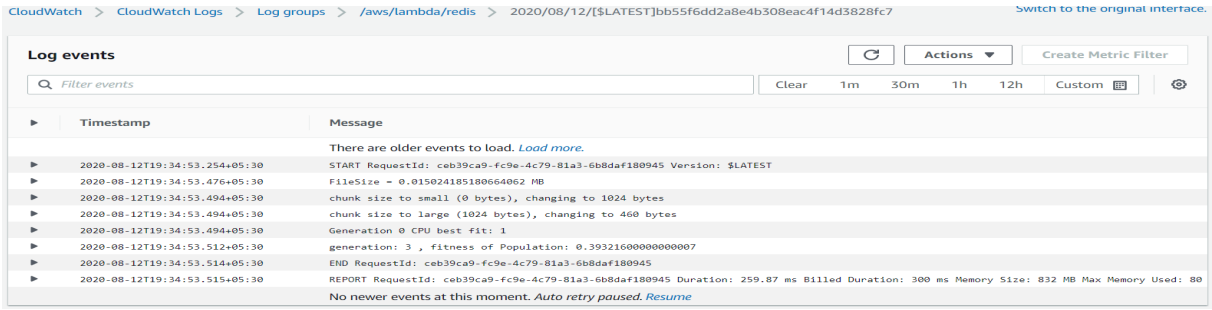

Figure 23: Cloud Watch logs

References

Giménez-Alventosa, V., Moltó, G. and Caballer, M. (2019). A framework and a performance assessment for serverless MapReduce on AWS lambda, **97**: 259–274. journal Impact Factor: 4.639.
URL: *http://www.sciencedirect.com/science/article/pii/S0167739X18325172*

Pu, Q. (Feb 2019). Shuffling, fast and slow: Scalable analytics on serverless infrastructure, p. 15.
URL: *https://www.usenix.org/system/files/nsdi19-pu.pdf*

Wang, L., Li, M., Zhang, Y., Ristenpart, T. and Swift, M. (2018). Peeking behind the curtains of serverless platforms, p. 13.
URL: *https://www.usenix.org/system/files/conference/atc18/atc18-wang-liang.pdf*