

Cloud project report

Shailendra Kushwaha

 Cloud Service Technology

16 May 2019

|  |  |
| --- | --- |
| **Author(s)**  Shailendra Kushwaha | |
| **Degree programme**  Business Information Technology | |
| **Report**  Cloud Service Technology Project (ICT4TF024-3002) | **Number of pages and appendix pages**  22 + 2 |
| The Report is based on both the theoretical as well as the practical implementation about a solution architect in AWS Cloud Service. The acknowledgement of this research gives the enough experience to the reader who wants to host their website on AWS cloud.  . | |
| **Keywords**  Cloud computing, | |

Table of contents

[1 Introduction 1](#_Toc8936196)

[2 AWS Overview 1](#_Toc8936197)

[2.1 The History of AWS 1](#_Toc8936198)

[2.2 Market Share by AWS 2](#_Toc8936199)

[2.3 Services by AWS 2](#_Toc8936200)

[2.4 AWS Global Infrastructure 3](#_Toc8936201)

[3 Identity Access Management & S3 3](#_Toc8936202)

[3.1 Identity Access Management (IAM) 3](#_Toc8936203)

[3.2 S3 4](#_Toc8936204)

[3.2.1 Creating S3 Bucket 4](#_Toc8936205)

[3.2.2 S3 Version Control 8](#_Toc8936206)

[3.2.3 S3 Replication 8](#_Toc8936207)

[3.2.4 S3 Lifecycle Management and Glacier 8](#_Toc8936208)

[3.2.5 Cloud Front 8](#_Toc8936209)

[4 EC2 9](#_Toc8936210)

[4.1 EC2 Instance Types 9](#_Toc8936211)

[4.2 EBS 10](#_Toc8936212)

[4.3 Launch First EC2 Instance 10](#_Toc8936213)

[4.4 Using ssh key generated with Putty for using generated ssh keys 15](#_Toc8936214)

[4.5 How to update the EC2 Instance? 16](#_Toc8936215)

[5 Lambda 17](#_Toc8936216)

[6 API GateWay 18](#_Toc8936217)

[7 Build A Serverless Webpage with API GateWay & Lambda 19](#_Toc8936218)

[8 References 29](#_Toc8936219)

# Introduction

As we started this course by getting familiar with cloud technologies and why now a day cloud is one of the highly growing technology as business companies are moving their infrastructure to the cloud and this will increase more and more. We found many companies are there providing cloud services like AWS (Amazon Web Services) by Amazon, Google Cloud Platform by Google, Microsoft Azure by Microsoft and many more other companies. But here we decided to carry our project by using AWS (Amazon Web Services) as our platform as it is one of dominant cloud service providing company and its demand is increasing more and more along with businesses moving their infrastructure to the cloud. This has created demand for AWS certified manpower as well so, being familiar will support in professional growth as well.

# AWS Overview

## The History of AWS

Before AWS and other clouds, the situation was in order to provide services by any small companies either they had to use platform from some limited platform providing companies which were very expensive to get and stay with them for 3-5-year contract even though any collateral damage occurs during experiments. Other option was to run businesses we had to rent or buy services for which should have enough money to invest to build our infrastructure. But along with many cloud services like AWS, Microsoft Azure, GCP, IBM cloud the situation has changed. If we want to run our businesses serverless easily we can use different clouds and build platform as per our need and pay only for the services that we’ll use and those platforms scales automatically with demand for little to no running costs. And beauty of these cloud services is in our businesses we can try new ideas and do a lots of experiments and if it don’t work we can just terminate to used resources and we can save our costs just by using virtual machines for using them for couple of hours.

Since, we are doing our project in Amazon Web Services (AWS) so, we will look at how AWS started and until now where it is, where it is going and why it’s important to learn AWS. AWS, the pioneering cloud computing platform provided by Amazon.com, emerged from internal initiatives at Amazon over 15years ago to both aid developers and also improve the efficiency of the company’s own infrastructure (Alec Rojas, 2017).

## Market Share by AWS

A screenshot of a map

Description automatically generated

It seems clear that in the case of different major cloud providing companies market share AWS is leading one. Bezos has said, “AWS had the unusual advantage of a seven-year head start before facing like-minded competition. As a result, the AWS services are by far the most evolved and most functionality-rich.” (Katy Stalcup, 2019)

## Services by AWS

Amazon web services currently until 2019 offering more than 1600 including high- and low-level services. These are high level services offered by Amazon Web Services until today and which contains so many low level server in them.

A screenshot of a cell phone

Description automatically generated

## AWS Global Infrastructure

In this section we have looked at what makes AWS truly global that is serving its customers to more than 190 countries (About, s. 2019). The AWS Global Infrastructure is designed and built to deliver the most flexible, reliable, scalable, and secure cloud computing environment with the highest quality global network performance available today. Every component of the AWS infrastructure is design and built for redundancy and reliability, from regions to networking links to load balancers to routers and firmware. AWS Global Infrastructure consists of Regions (20+), Availability Zones (AZ) (60+) and Edge locations (155+). Here, we have differentiated Regions, Availability Zones and Edge locations.

# Identity Access Management & S3

## Identity Access Management (IAM)

**What is IAM?**

The concepts of IAM comes when we are needed to manage the users and their level of access to the AWS console. It allows us to set up users’ group’s permissions and their roles. So, it is important to understand IAM and how it works, for administrating a company’s AWS account in real life.

The main key features of IAM can be listed as below:

* Centralize control of our AWS account.
* Shared access to our AWS account.
* Granular permissions (defined users’ permissions with the services for their access)
* Identity Federation (including Active directory, Facebook, Linkedin etc.) for accessing the console.
* Multifactor Authentication: It means along with username and password needed special code to access.
* Provide temporary access for users/developers and services where necessary like by using mobile app
* Allows us to set up our own password rotation policy which means we can make users to rotate their passwords every three months or three months for extra securing.
* Integrates with many different AWS services.
* Support PCI DSS Compliance: It is basically a complaint framework that if we’re taking credit card details, we need to be complaint with the framework.

## S3

S3 means **Simple Storage Services**. S3 provides developers and IT teams with secure, durable, highly scalable object storage. Amazon S3 is easy to use with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

S3 is a safe place to store our file. It is an object-based storage. The data is spread across multiple devices and facilities.

**The basics of S3 are as follows:**

* S3 is Object-based – i.e. allows us to upload files.
* Files can be from 0 Bytes to 5 TB.
* There is unlimited storage.
* Files are stored in Buckets (means folders).
* S3 is a universal namespace. That is, names must be unique globally.

S3 has the following features:

* Tired Storage Available
* Lifecycle Management: We can move our objects to different tiers defining different conditions e.g. we can define when the file is 30 days old move it to this tier when it’s 90 days old archive it off to Glacier and so on.
* Versioning: So, we can have multiple versions of object in our S3 Buckets.
* Encryption
* MFA Delete
* Secure our data using Access Control Lists and Bucket Policies

### Creating S3 Bucket

Here, we will create bucket, for which first we will go to AWS Services then from there form Storage section which is high level service containing many low and core service we’ll choose S3 where we are going to create of first storage bucket also upload some content to it.

Once we click on S3 we can see this in our console.

A screenshot of a social media post

Description automatically generated

Here, we can see “Create bucket” option from where we can start also here we can see in region we have “Global” which we’ll specify once we’ll create our bucket.

There are four main steps through which we have to go to create bucket for our content storage in S3.

**Step 1: Choose name and region**

Here, we will give bucket name and specify the region for our bucket and also can copy settings from existing bucket if we want.

**A screenshot of a cell phone

Description automatically generated**

**Step 2: Configure Open**

Here, we can configure different settings related to our bucket that we want specify with all the requirements.

**A screenshot of a cell phone

Description automatically generated**

**Step:3 Set Permissions**

In this section we’ll set all the permissions that we want to define for our bucket.

A screenshot of a cell phone

Description automatically generated

**Step 4: Review**

In this part we’ll review all the configurations and permissions that we defined for our bucket.

A screenshot of a computer screen

Description automatically generated

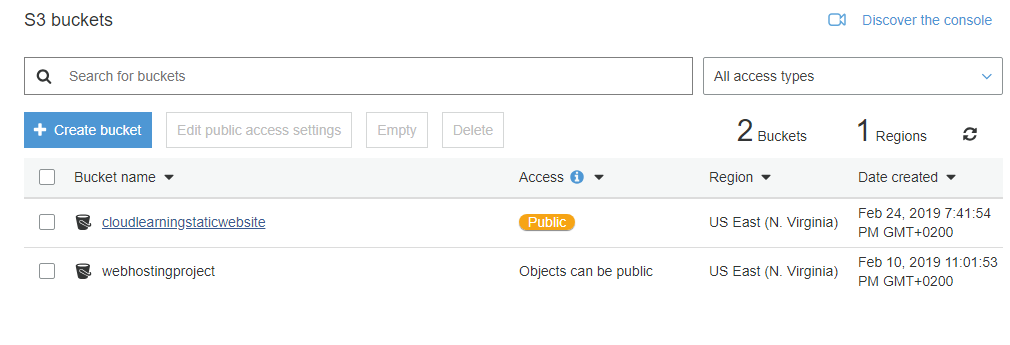
**Our Result:**

So, following these 4 steps we can create our bucket.

In our first experiment of creation of first bucket, the exact URL is:

<https://s3-ew-west-1.amazonaws.com/cloudlearningstaticwebsite/>

(The green marked text represents the region whereas the yellow marked text represent the name of bucket in S3.)



### S3 Version Control

This facilities in AWS Cloud we serve to keep multiple version of object in one bucket. For eg: if we are stored a file file.txt as version 1 and on later when we will update the version from 1 to 1.1, both the files of different versions can keep in the same buckets.

### S3 Replication

This service allows you to replicate the objects from one bucket to another buckets. It gives the options all about which files you want to replicate where, which version to which region everything else.

The replicated files/objects don’t automatically transfer. For this, we need to use the command line tool bundled installer.

### S3 Lifecycle Management and Glacier

This service is about to set the rules of the lifecycle of your objects/files which allows you to automatically expires on your attention period. Let's say, you uploaded a file and if you set to expire after 3 months if it is not modified within 3 months, then the objects can be removed from your buckets after 3 months. For resolve this, you can set the rules for object after a certain period to move from one to another S3 version.

The following diagram can clear the concepts of Lifecycle management.

Initial state after 30 days after 90 days

V1.txt

V2.txt

V1.txt

### Cloud Front

What is CDN? Why do we need CDN?

A **Content Delivery Network**(CDN) is a system of distributed servers(network) that deliver webpages and other web content to a user based on the geographic locations of the user, the origin of the webpages and a content delivery server. The main purpose of CDN is to make load the content fast and to reduce the content loading time.

Key Terminologies:

* Edge locations: This is the location when content will be cached. This is separate to an AWS Region/AZ.
* Origin: This is the origin of all the files that the CDN will distribute. This can be either an S3 bucket, an EC2 instance, an Elastic load balancer or Route53.
* Distribution: This is the name given the CDN which consists of collection of Edge Locations.

Amazon Cloud Front can be used to deliver your entire website including dynamic, static, streaming and interactive content using a global network of edge locations. Requests for your content are automatically routed to the nearest edge location. So, content is delivered with the best possible performance.

Amazon cloud front is optimized to work with other Amazon Web Services like Amazon Simple Storage Service (Amazon S3), Amazon Elastic compute cloud (Amazon EC2), Amazon Elastic load Balancing and Amazon Route53. Amazon Cloud Front also works seamlessly with any non-AWS origin server which stores the original definitive versions of your files. There are mainly two types of Distribution in Cloud Front as CDN.

* Web Distribution: Typically used for websites
* RTMP (Real Time Messaging Protocol): Used for media streaming

# EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down as your computing requirements change. EC2 is a main cause of the economic billing as your cloud storage used. There are different EC2 options to effects on your billing.

* On Demand- allows you to pay a fixed rate by the hour (or by the second) with no commitment.
* Reserved- provides you with a capacity reservation and offer a significant discount on the hourly charge for an instance. 1 year of 3 year Terms.
* Spot- enables you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times.
* Dedicated Hosts- Physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses.

## EC2 Instance Types

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. It comprises varying combination of hardware and software as well as the better performance of different resources for your applications. Some instance types, specialty and their use case are listed as below.

|  |  |  |
| --- | --- | --- |
| **Family** | **Speciality** | **Use Case** |
| F1 | Field programmanle Gate Array | Genomics research, financial analytics, real time video processing, big data etc. |
| I3 | High speed storage | NoSQL DBs, Data warehousing etc. |
| G3 | Graphics Intensive | Video Encoding/3D Application Streaming |
| H1 | High Disk Throughput  (Throughput optimized is an EBS volume types which is used for Big Data, log processing, Data warehouse.) | MapReduce. Based workloads, distributed file systems such as HDFS and mapR-FS |
| M5 | General Purpose | Application Servers |
| T2 | Lowest cost, General Purpose | Web servers/small DBs |
| X1 | Memory Optimized | SAP HANA/Apache Spark etc. |

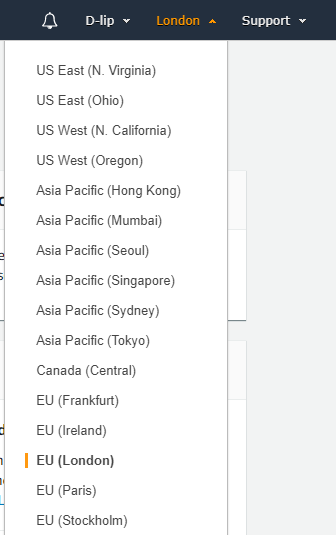
## EBS

Amazon EBS (Elastic Block Store) allows you to create storage volumes and attach them to Amazon EC2 instances. Once attached, you can create a file system on top of these volumes, run a database or use them in any other way you would use a block device. Amazon EBS volumes are placed in a specific Availability Zone, where they are automatically replicated to protect you from the failure of a single component.

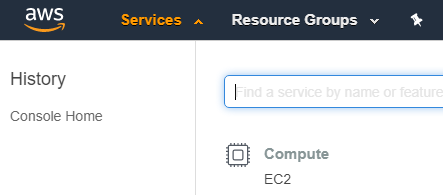
## Launch First EC2 Instance

**Before Starting to Launch our EC2 first we need to setup something:**

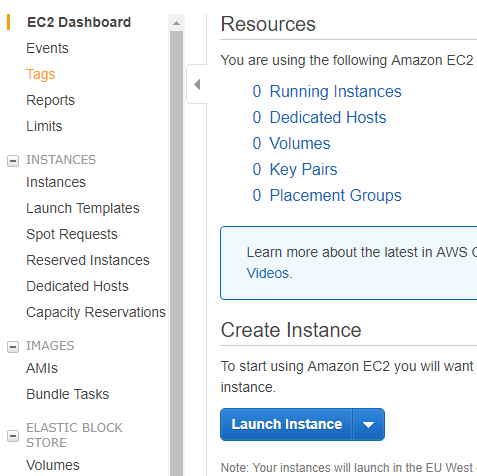
**Choose region**



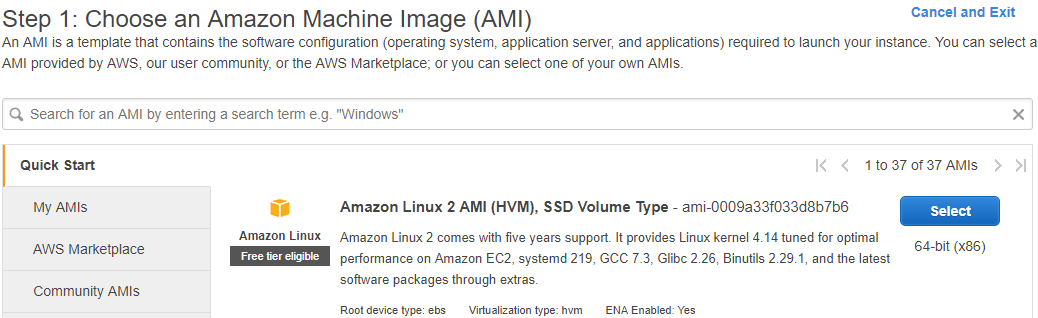
**Select the service EC2 from AWS management Console Dashboard**



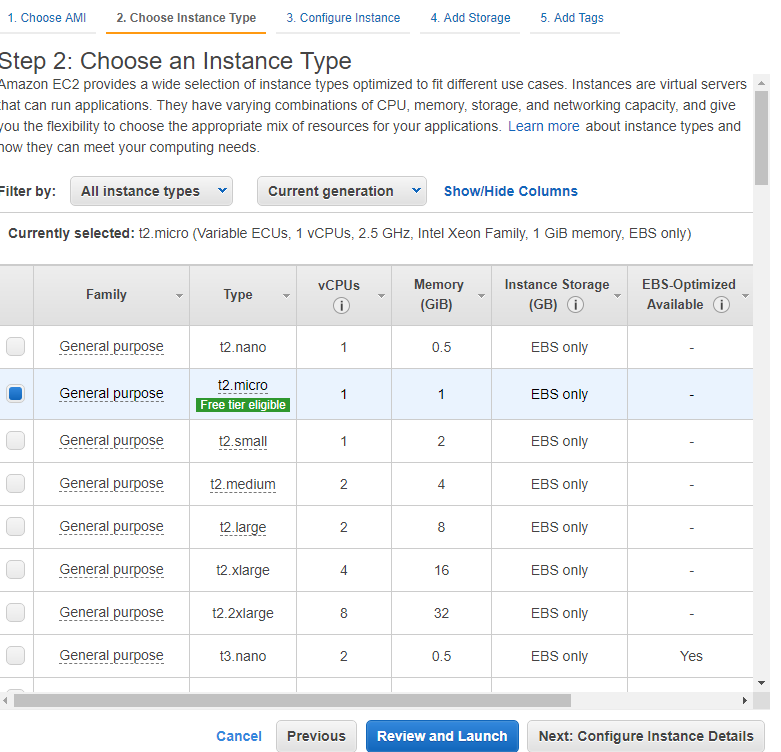
**Launch the instance**



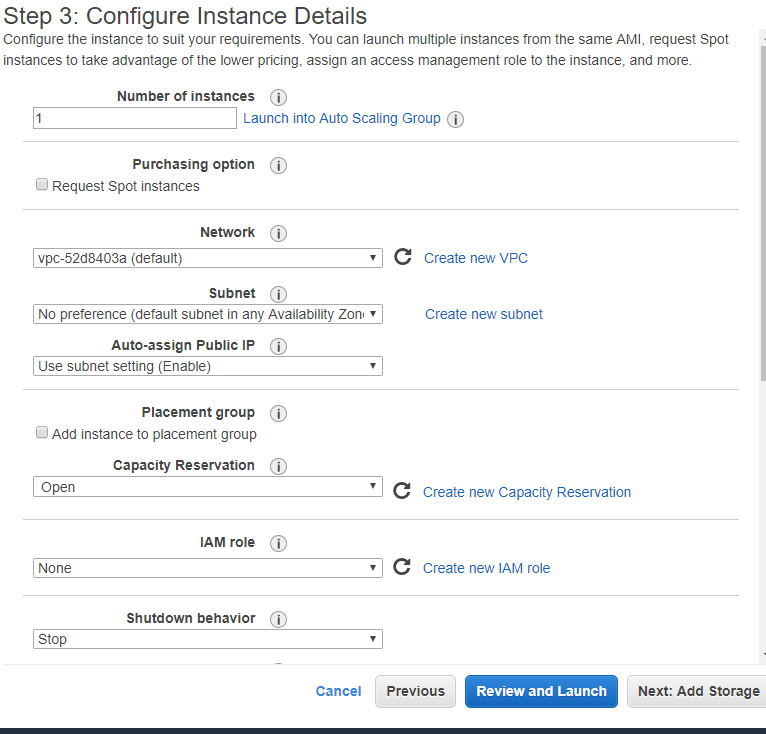
**Step 1: Choose the first Amazon Machine Image (AMI) for Linux**



**Step 2: Choose the instance type**

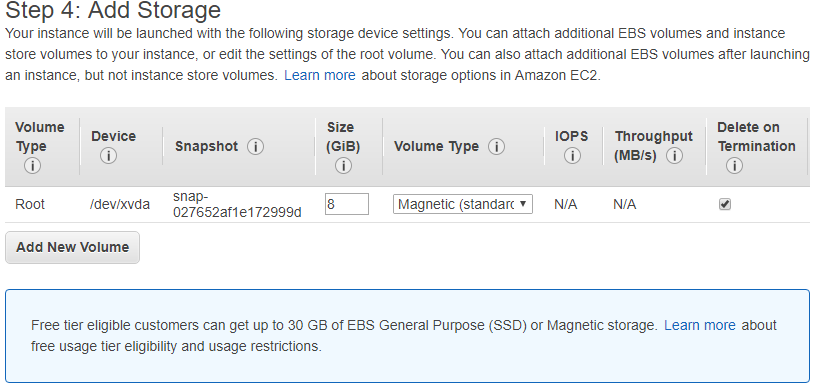


**Step 3: Configure Instance**

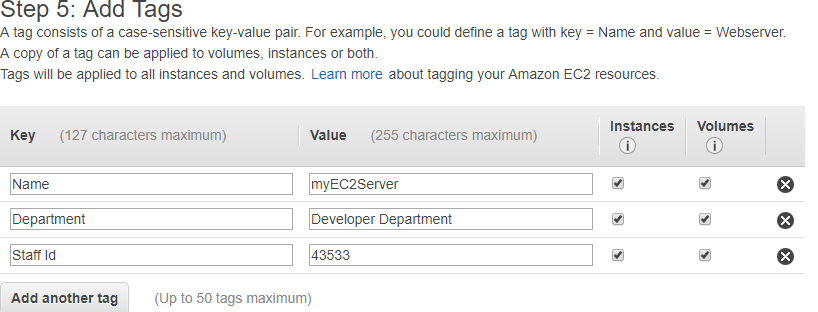


**Step 4: Add Storage**

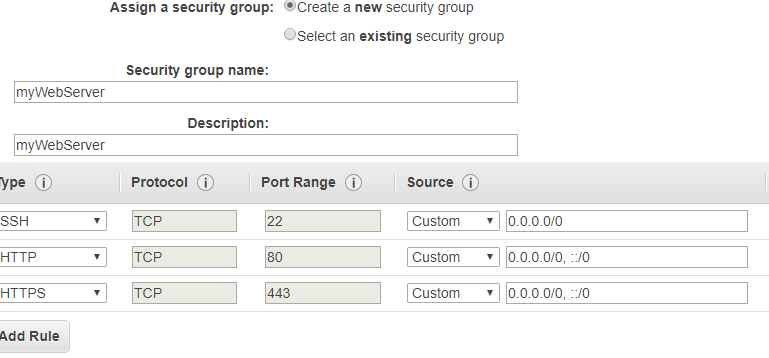
In this step, we choose the Magnetic volume type. We can add and delete the new volume if needed.



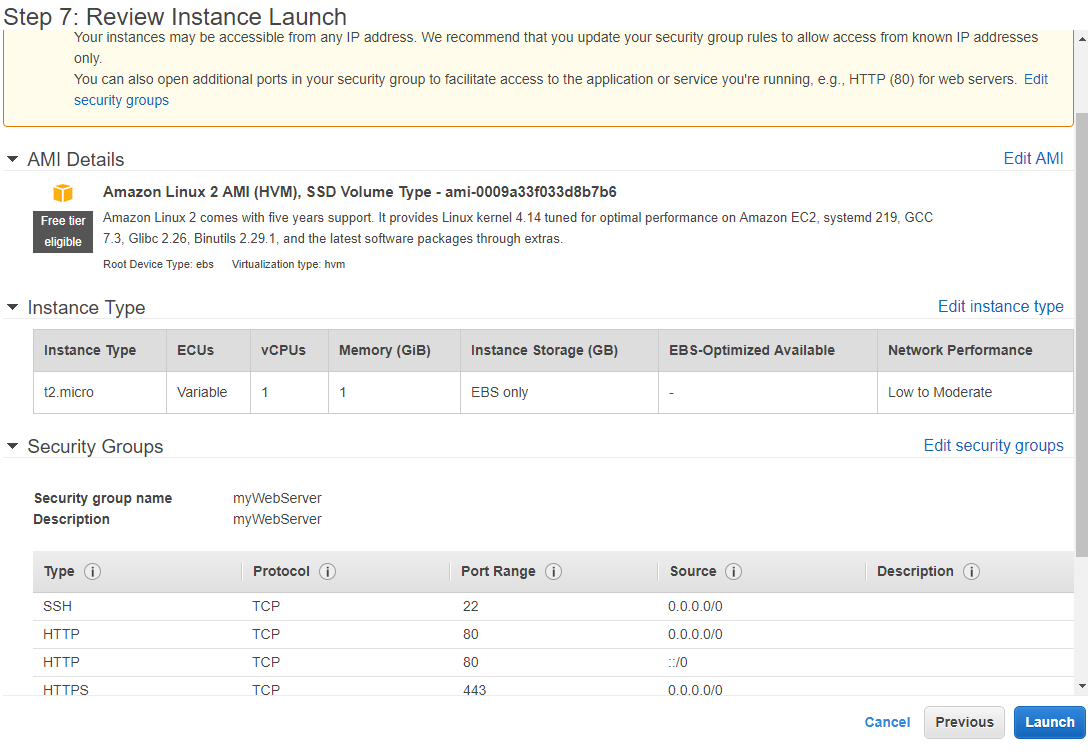
**Step 8: Add tag to EC2 Instance**



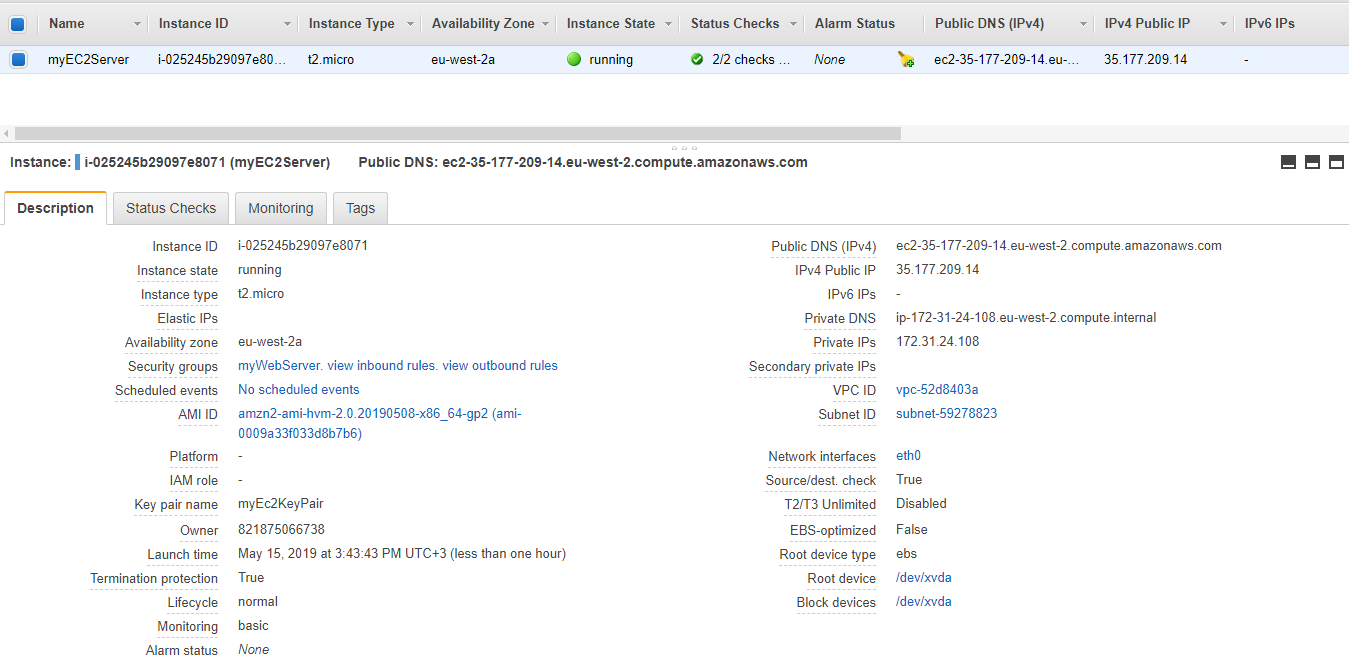
**Step 9: Configure security Group**



**Step 10: Review**



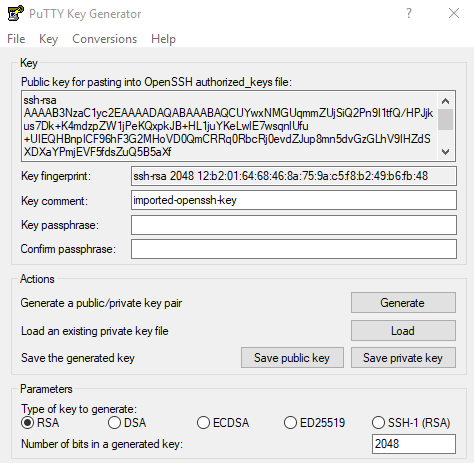
**Finally launch an EC2 Instances for Demo**

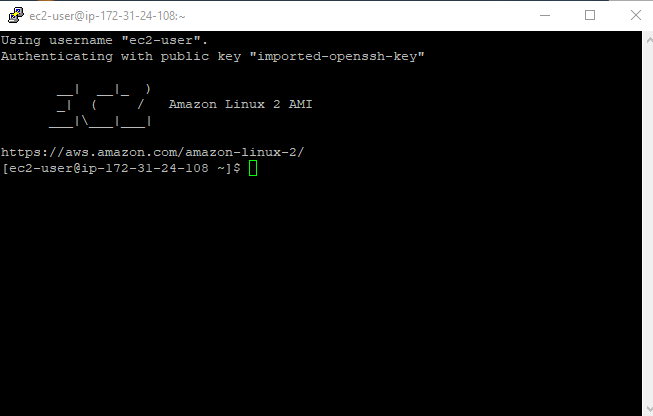


## Using ssh key generated with Putty for using generated ssh keys

**How to use Putty to use the generated ssh key from EC2 instance to protect your IP**

* First generate and download the ssh key
* Open the puttygen app in windows and load it





## How to update the EC2 Instance?

**To update the EC2 instance**

sudo su

yum update – y

# Lambda

What is Lambda?

Aws lambda is a compute service where you can upload your code and create a lambda function. AWS Lambda takes care of provisioning and managing the servers that you use to run the code. Lambda functions are independent, 1 event = 1 function. Lambda is serverless. You don’t have to worry about operating systems, patching, etc. Basically, Lambda refers to the encapsulates of:

* Data centers (of your code)
* Hardware
* Assembly Code/Protocols
* High level language
* Operating Systems
* Application Layer / AWS APIs
* AWS Lambda

You can use the lambda in the following ways:

* As an event-driven compute service where AWS Lambda runs your code in response to events. The events could be changes to data in an Amazon S3 bucket or an Amazon DynamoDB table.
* As a compute service to run your code in response to HTTP requests using Amazon API Gateway or API calls made using AWS SDKs. This is what we use at our project.

The language which are supported with Lambda are Node.js, Java, Python, C#.

**How the Lamda is priced?**

* Number of requests: first 1 million request are free. after that $0.20 per million requests.
* Duration: Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. You are charged $0.00001667 for every GB-second used.

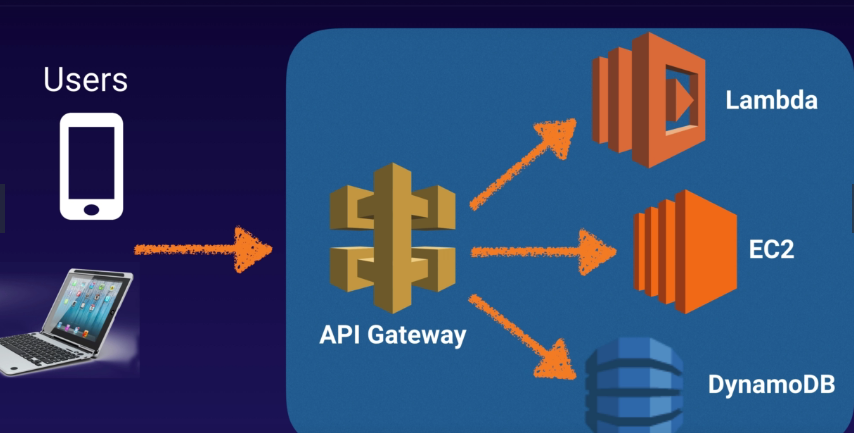
By using Lambda, we don’t need to worry about the servers. Because of this, there is no need of Database administrator and network administrator, just need to focus on code. The next benefit of Lambda is continuous Scaling. It means that when millions of users hitting API gateway, it will scale automatically to handle that load as well as lambda will scale automatically. This system is super cheap.

# API GateWay

Amazon API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor and secure APIs at any scale. With a few click in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic or functionality from your back-end services, such as applications running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda or any web applications.

**How API Gateway works?**

When the request is sent to AWS cloud, it dispatches to the Lambda or EC2 of DynamoDB base on request type.

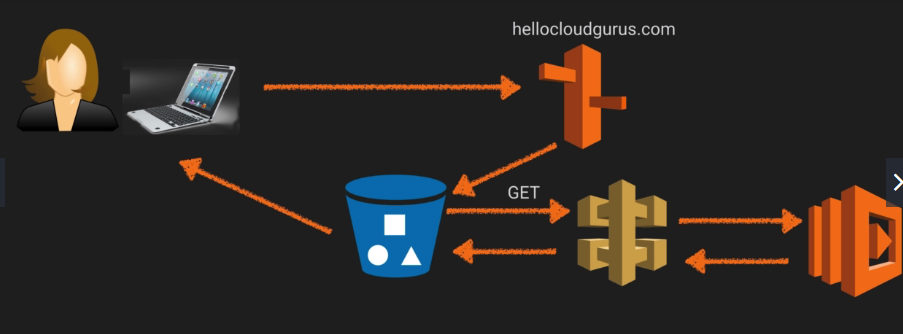
****

**What can API Gateway Do?**

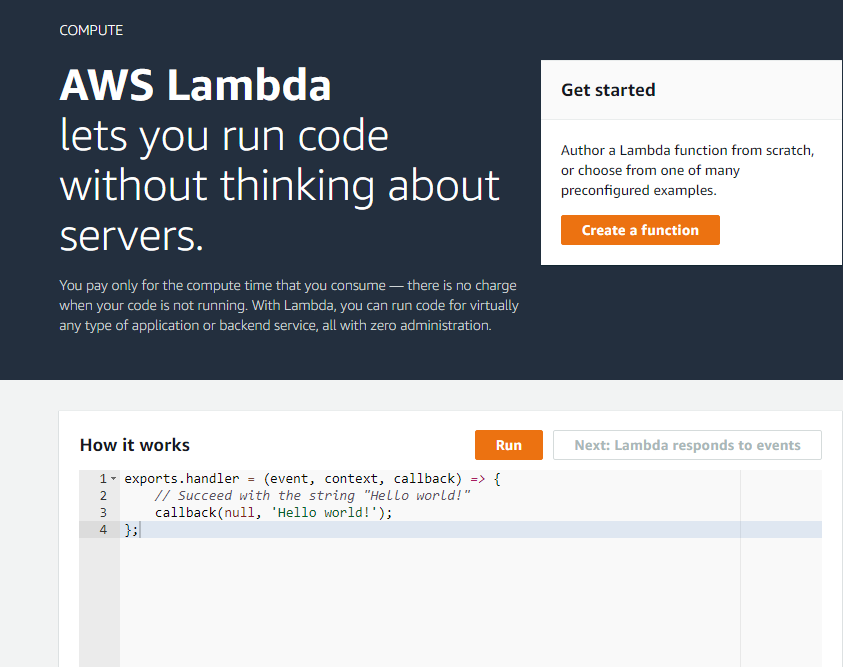
* Expose HTTPS endpoints to define RESTful API.
* Serverless-ly connect to services like Lambda & DynamoDB
* Send each API endpoint to a different target.
* Run efficiently with low cost
* Scale effortlessly
* Track and control by API key
* Throttle requests to prevent attacks
* Connect to CloudWatch to log all requests for monitoring
* Maintain multiple version of API

# Build A Serverless Webpage with API GateWay & Lambda

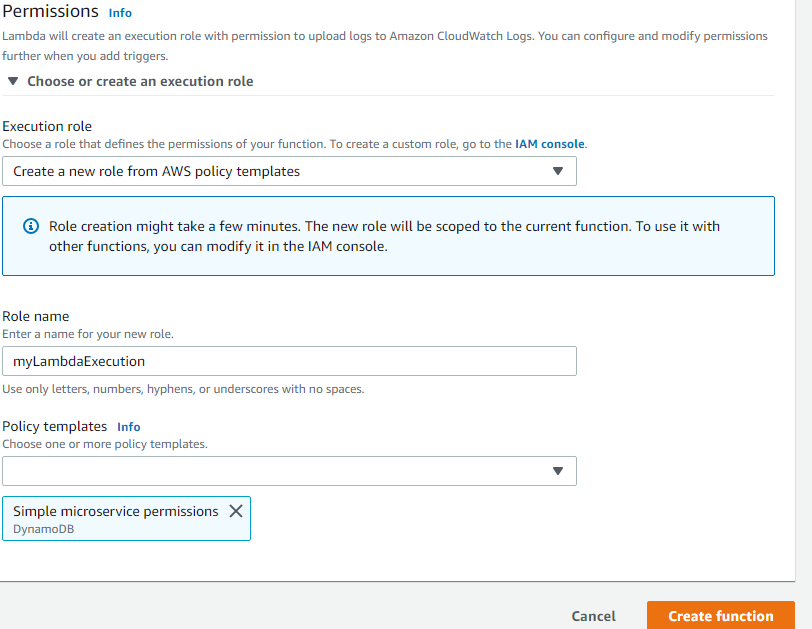
In our project of serverless webpage, we consider that we have users and they are going to make a get request all they are trying to resolve the URL name which will be. We haven’t registered the domain name yet so that we are going to use the DNS name for the free buckets. In the page, we have got a button that you’re going to push when the button is going to trigger a get request to API gateway and then that is going to trigger a lambda function and that lambda function simply print out your name so you just change your name in there and that will be passed back through API gateway to bucket and return to the user.

****

**Creating of Lambda functions:**

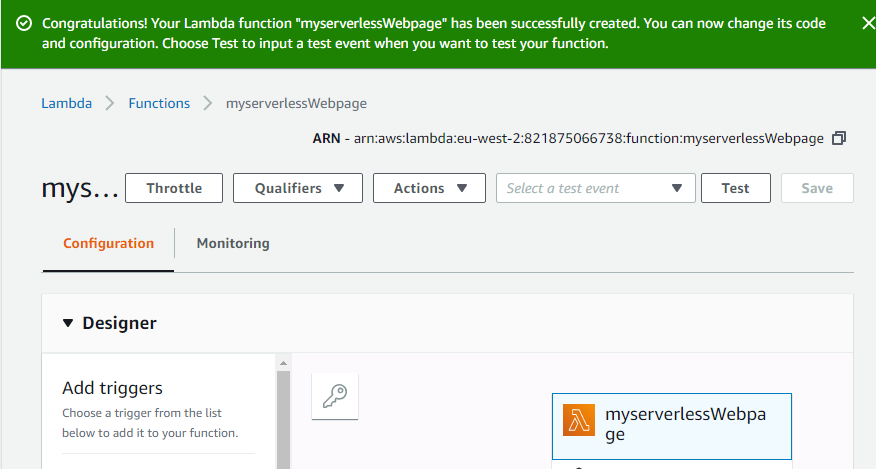


To create a new lambda function and click on the button “Create a function” and select the Author from scratch and fill the field as

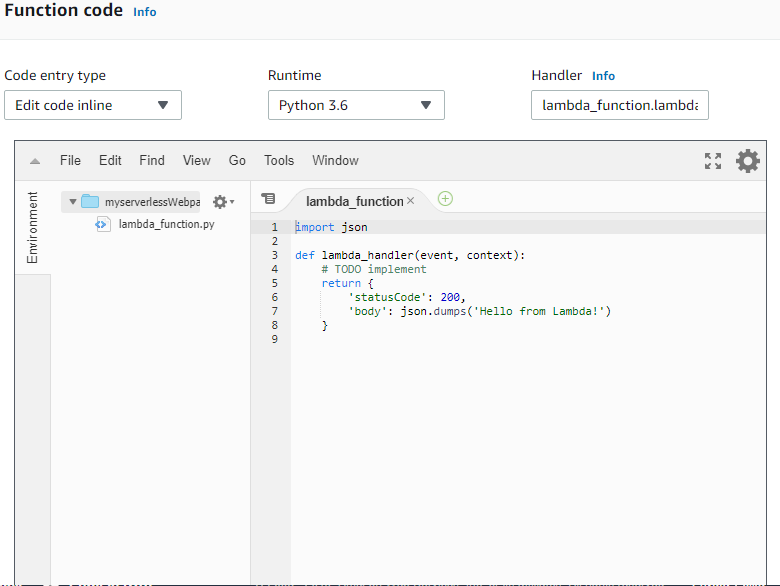


**Function Name: (as your choice)**

select runtime environment which you want to run your code and give a role name and select the “Simple Microservice permission” as policy templates and finally click on “**create function”**. After successively created the Lambda function, we will have the following window.

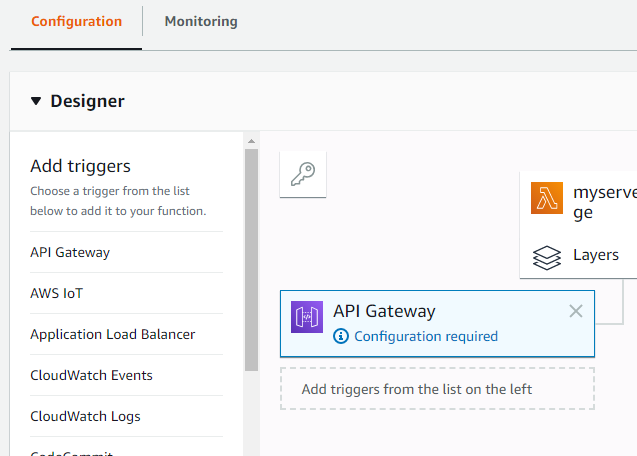
****

**The default code of our lambda function is as:**

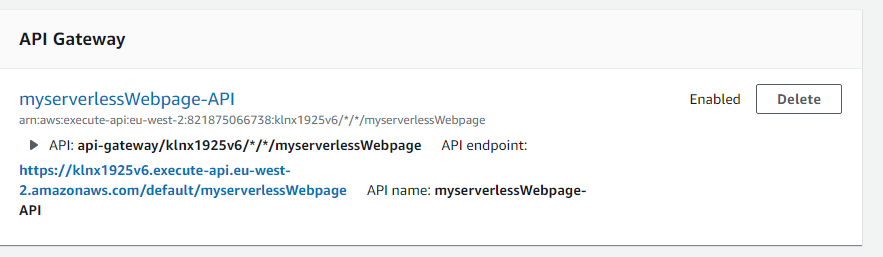
****

**Adding API Gateway Configure**

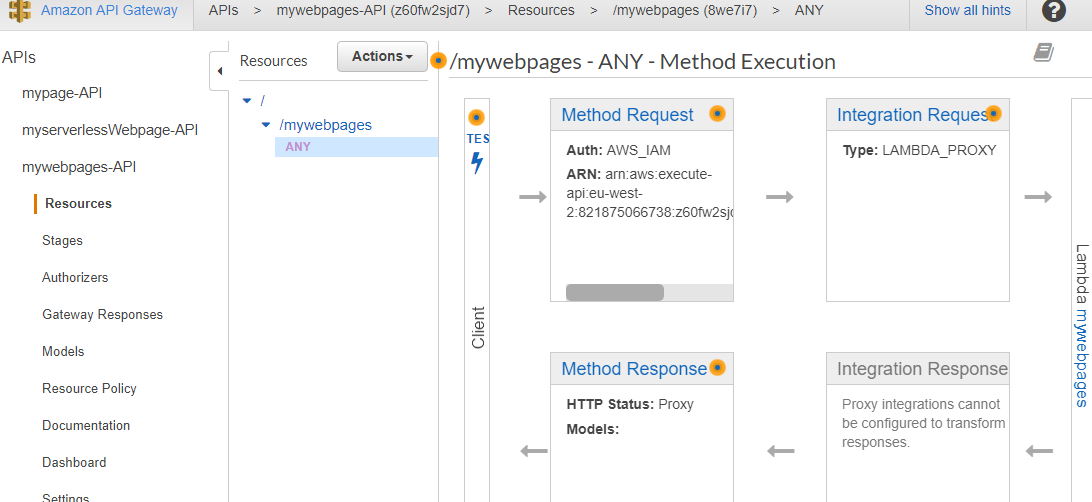
It works as a front door of our application to access in app.

****

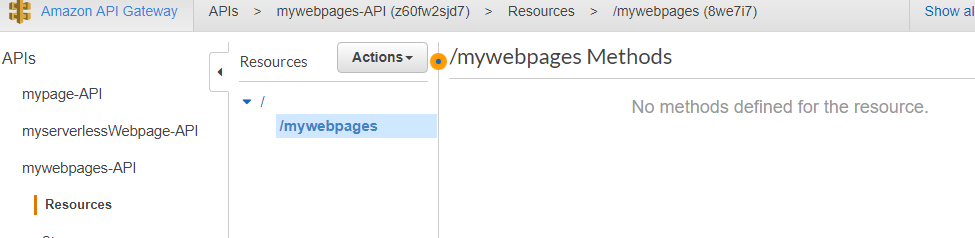
**After triggered:**

****

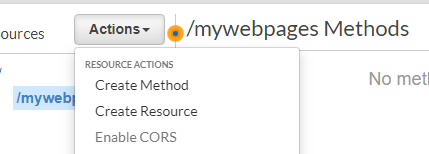
the default given URL doesn’t work yet. So, we need to execute the method which we can do by clicking myservlessWebpage-API (Note, we have changed our API name to “**/mywebpages**”).



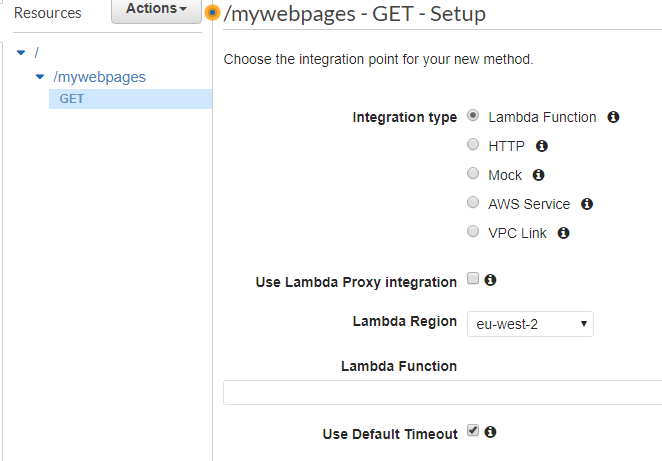
the method any is nor secure. so, we delete the method and create the new method called get method. Now we don’t have any method.



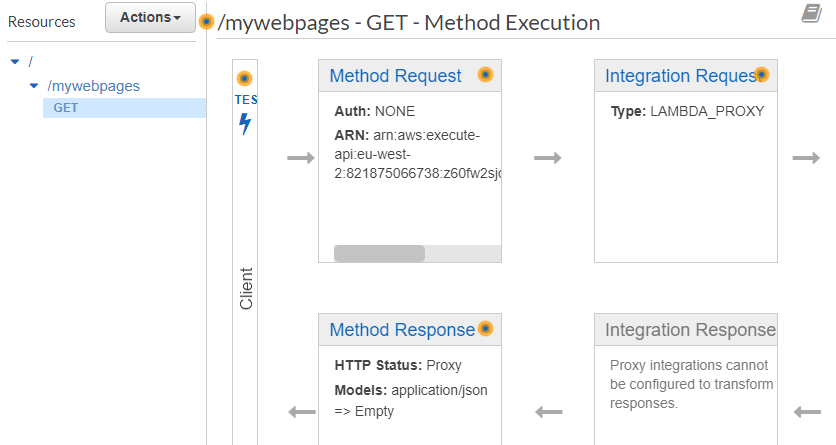
**Now we create GET Method.**



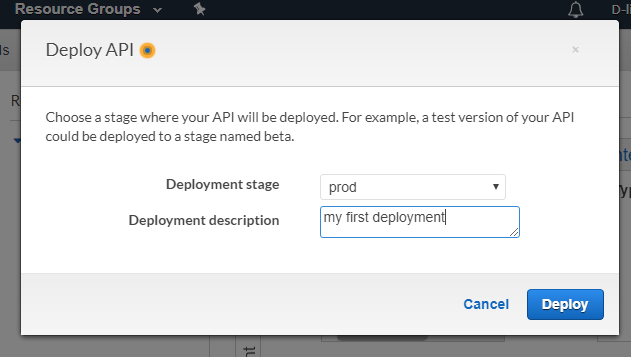
**After selecting Get method, settings are executed, and we got this:**



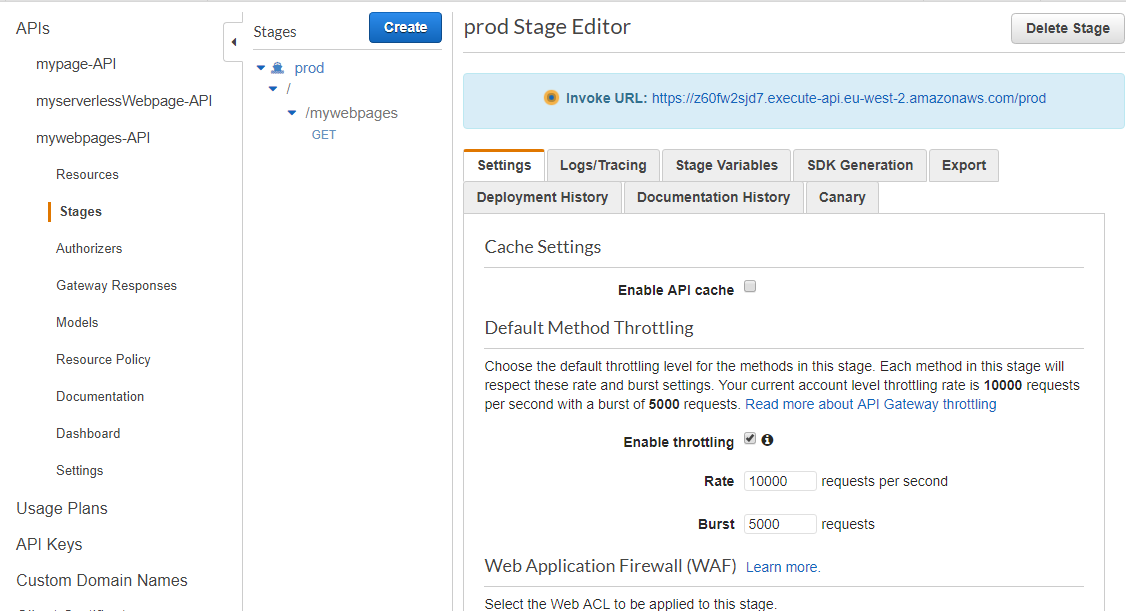
**Now after declared the Get Method,**



**Now we are going to deploy the API:**



**After deployed the API**



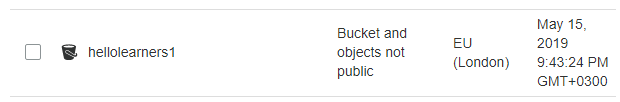
**Finally, we got the link.**

<https://z60fw2sjd7.execute-api.eu-west-2.amazonaws.com/prod/mywebpages>

Now we are going to host our serverless website using bucket.

so, lets create the bucket first:

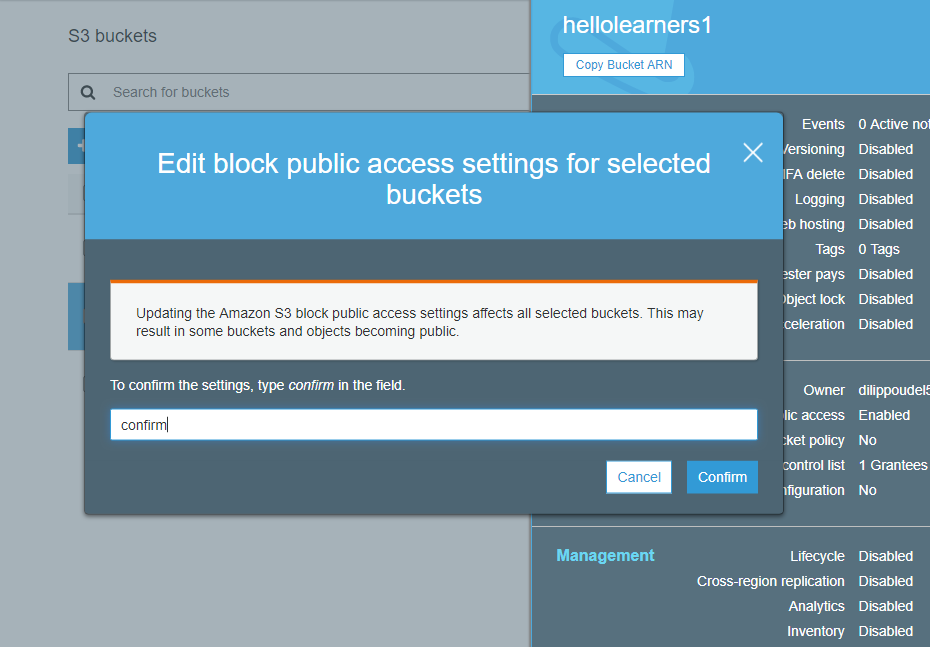
**Our newly created bucket name is hellolearners1.**



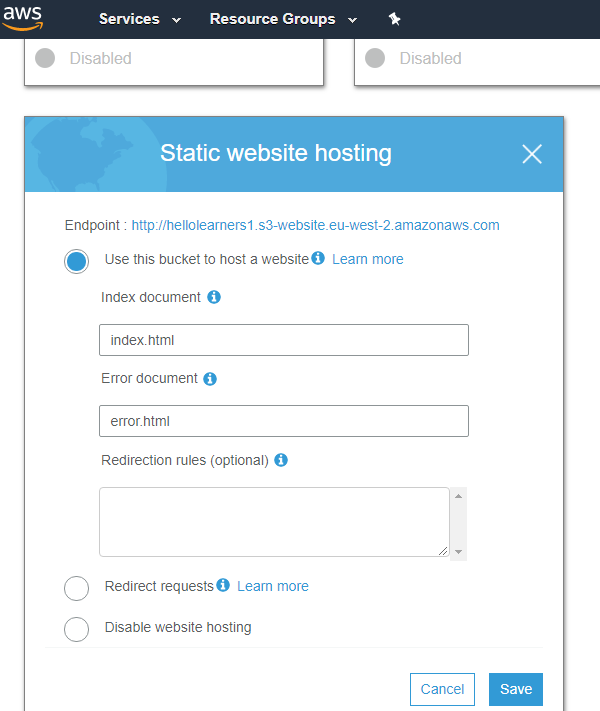
first, we created a index.html file to upload in our bucket with following snippets. In this code, we have used the URL link to run on browser and the image is pulled from another bucket which was already created during webhosting project.



The next step to use our newly created bucket, we must make it public.



Now, its turn to host the website.



This is our Final Product!!!

<http://hellolearners1.s3-website.eu-west-2.amazonaws.com/>



# References

*About*. (n.d.). Retrieved from Amazon documentation: www.docs.aws.amazon.com

Alec Rojas. (2017, August 31). *news blog*. Retrieved from media temple: https://mediatemple.net

Katy Stalcup. (2019, April 30). *blog*. Retrieved from ParkMyCloud: https://www.parkmycloud.com