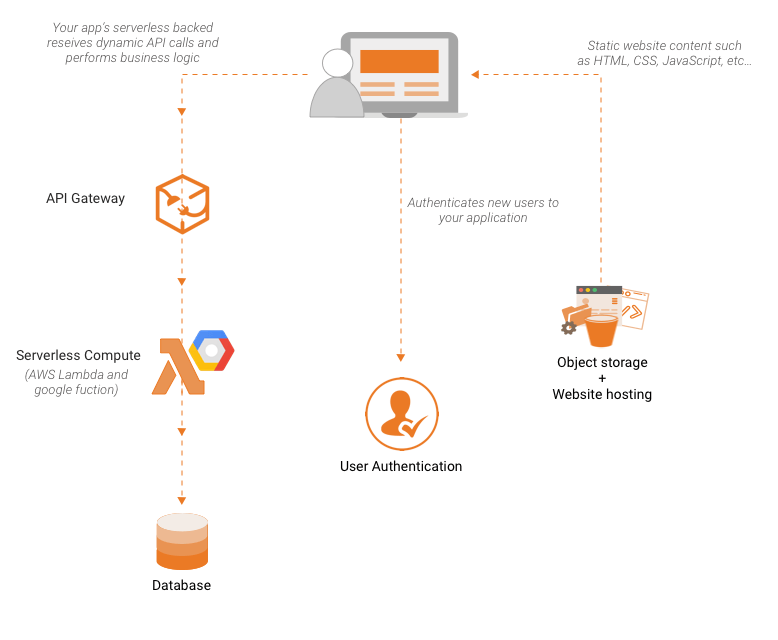
**BUILD A SERVERLESS WEB APPLICATION**

**What is a serverless Web Application?**

**Serverless** refers to an application framework for building **web applications** without going into the detailing of servers. The servers are managed by cloud provider, taking care of its provisioning and allocation. This makes the application to run in a stateless compute containers that are ephemeral and event-triggered.



With serverless, application development is dependent on a combination of third-party services, client-side logic and cloud-hosted remote procedure calls, and is hence referred to as **Functions as a Service.**

FaaS refers to an implementation of serverless architecture where a piece of business logic processes individual requests. It is an independent, server-side, logical functions which are small, separate, units of logic that take input arguments, operate on them and return the result, such as lambda. It is stateless, which means, any two invocations of the same function could run on completely different containers.

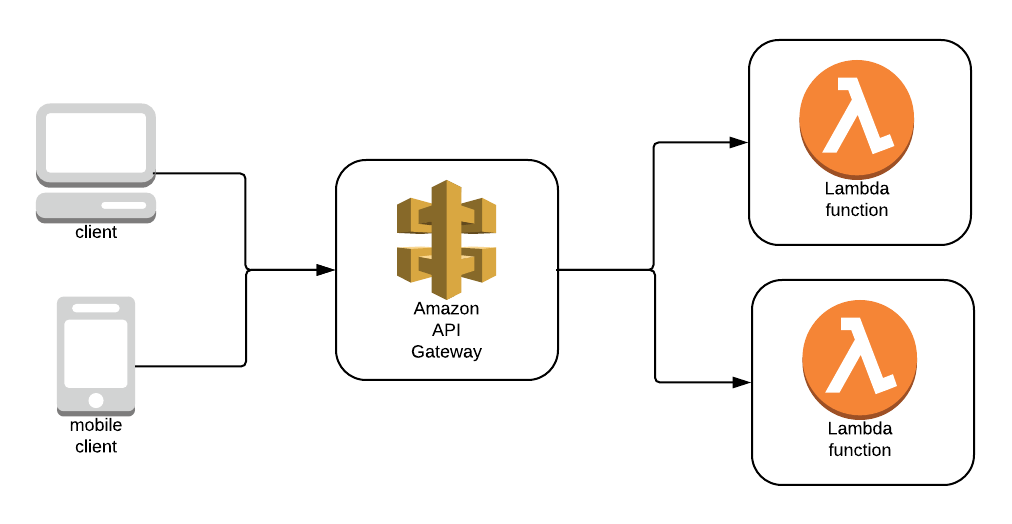
AWS Lambda, Azure Functions, IBM OpenWhisk and Google Cloud Functions are most well-known FaaS solutions available, supporting a range of languages and runtimes e.g. Node.js, Python, etc.

**Steps To Build A Serverless Web Application:**

Here I have divided all the steps in order of ‘Creation’, ‘Coding’, ‘Deployment’, ‘testing’ and ‘cleanup’.

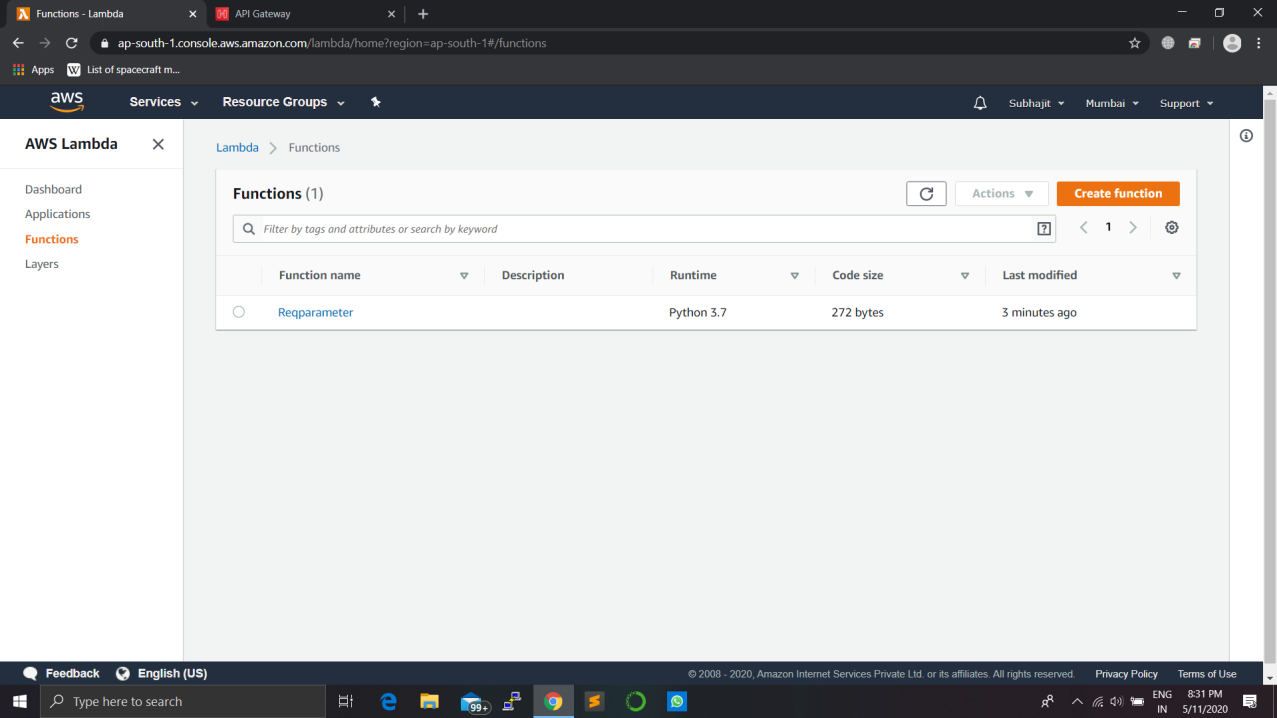
**STEPS:**

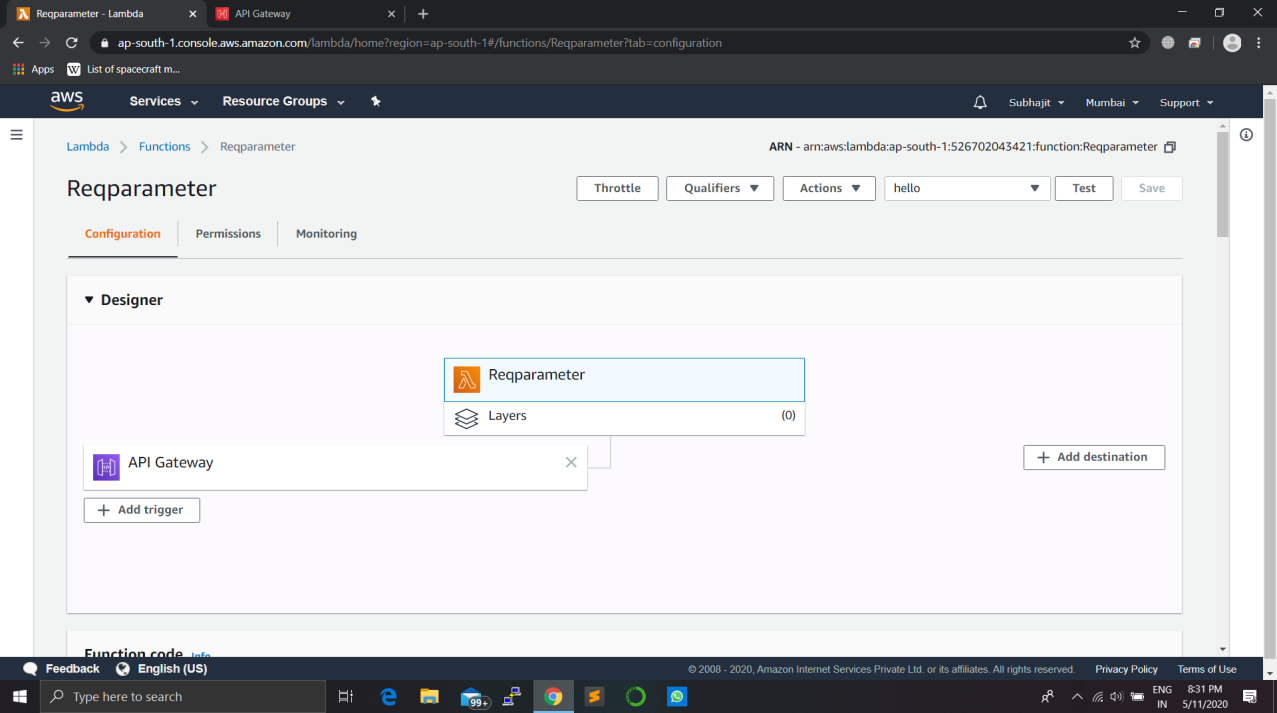
* Creation**:**
* Creation of Aws Lambda function
* Creation of Api-gateway
* Creation of IAM role for lambda
* Creation of a S3 bucket
* Coding:
* Writing the code for lambda function
* Writing your own html page
* Deployment:
* Setup your api-gateway and deploy it.
* Testing:
* Test the above framework by giving some actual value.
* Cleanup:
* Finally delete all the created services after your project is complete.

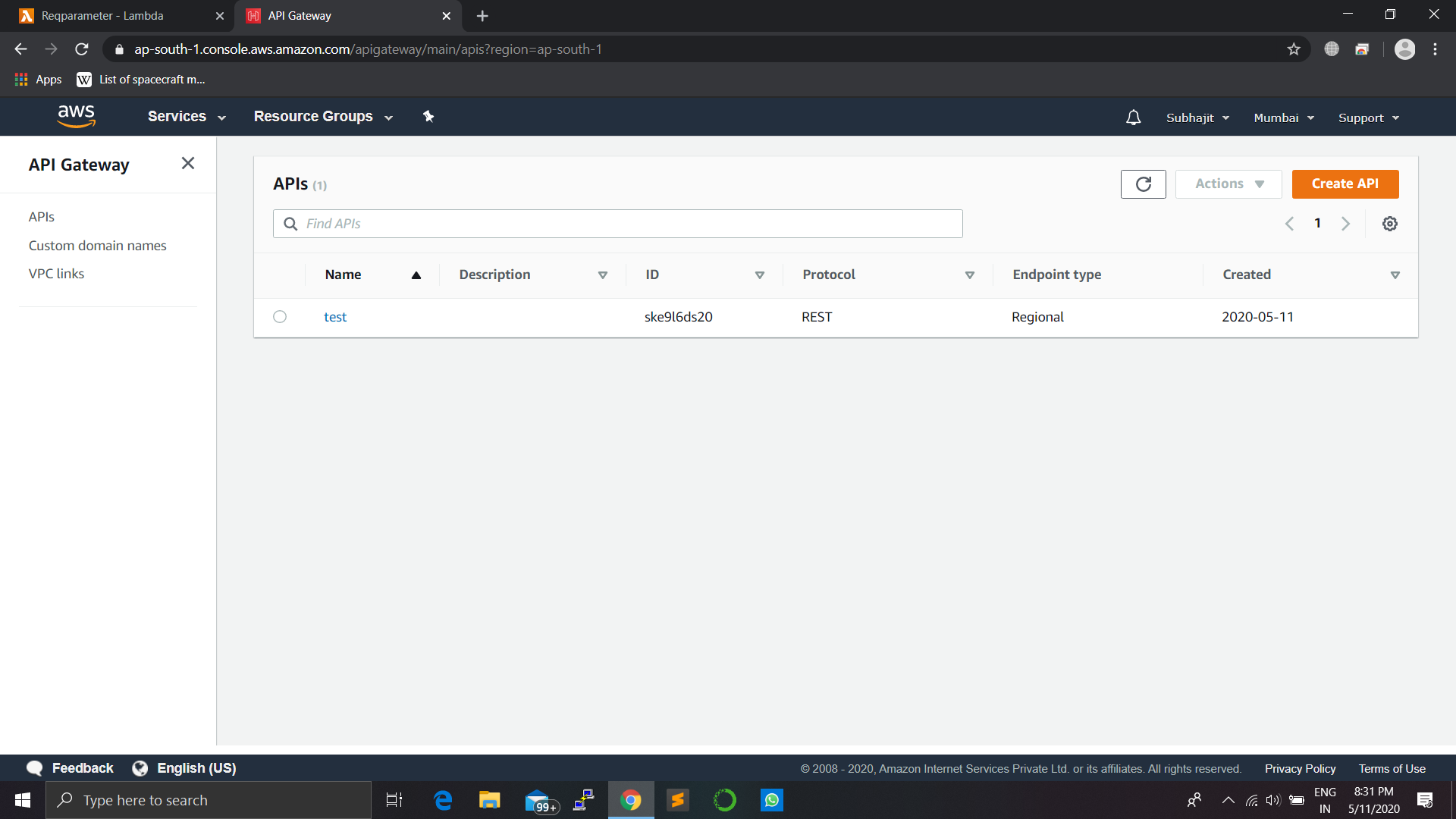


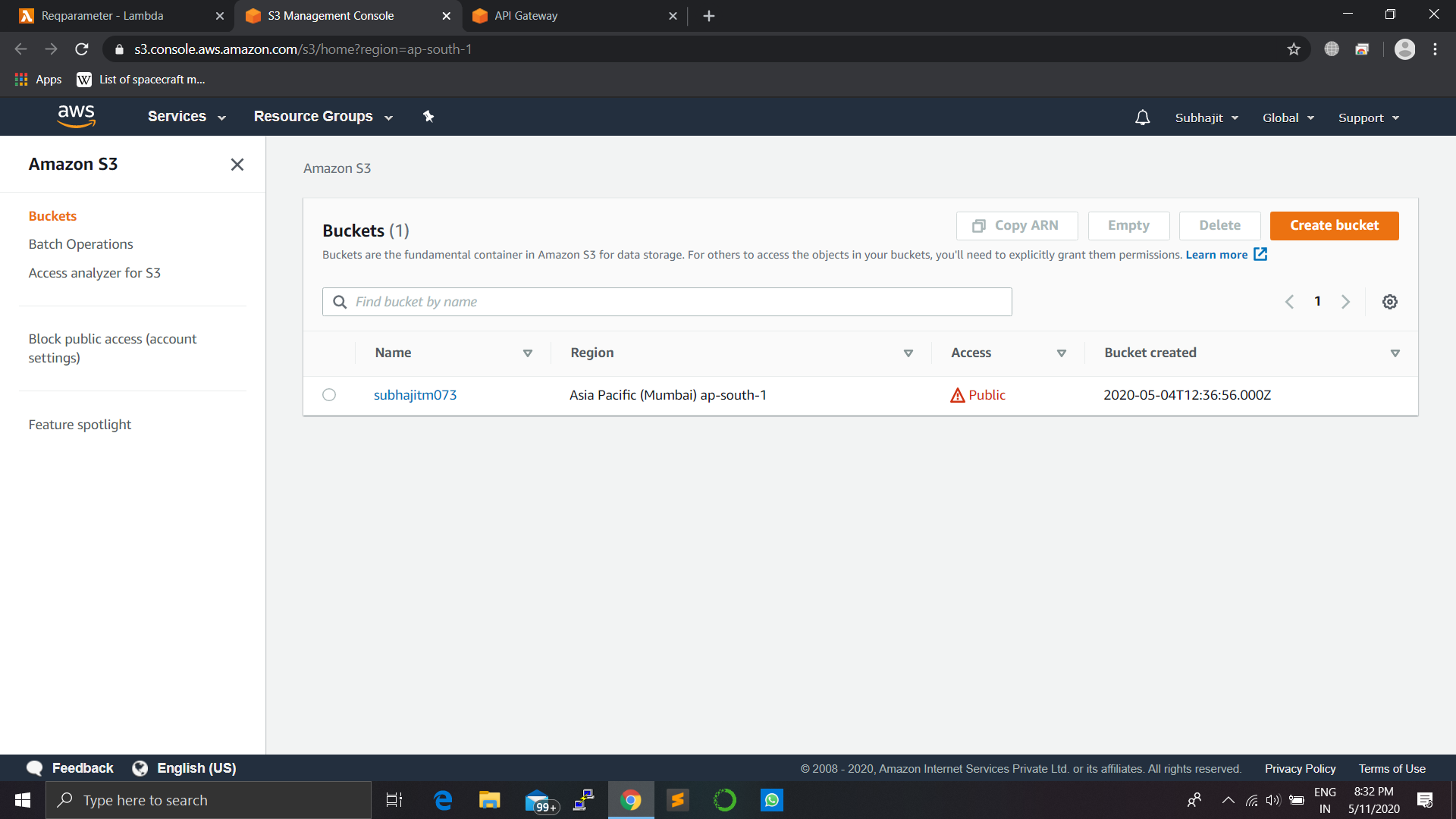
Step1: Creation

* In this step we will be creating framework for our project.
* First create a lambda function.
* Then create an iam role to give permission to your lambda function.
* Create a api-gateway for your framework.
* Create a S3 bucket to store the html file.



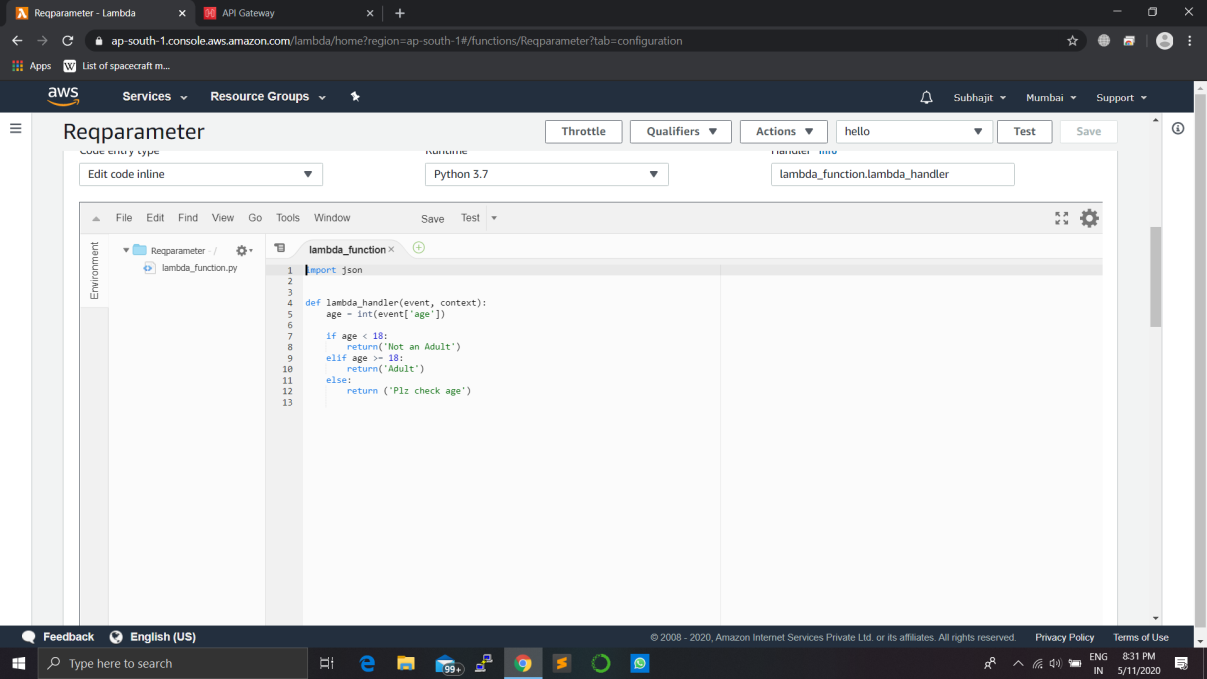






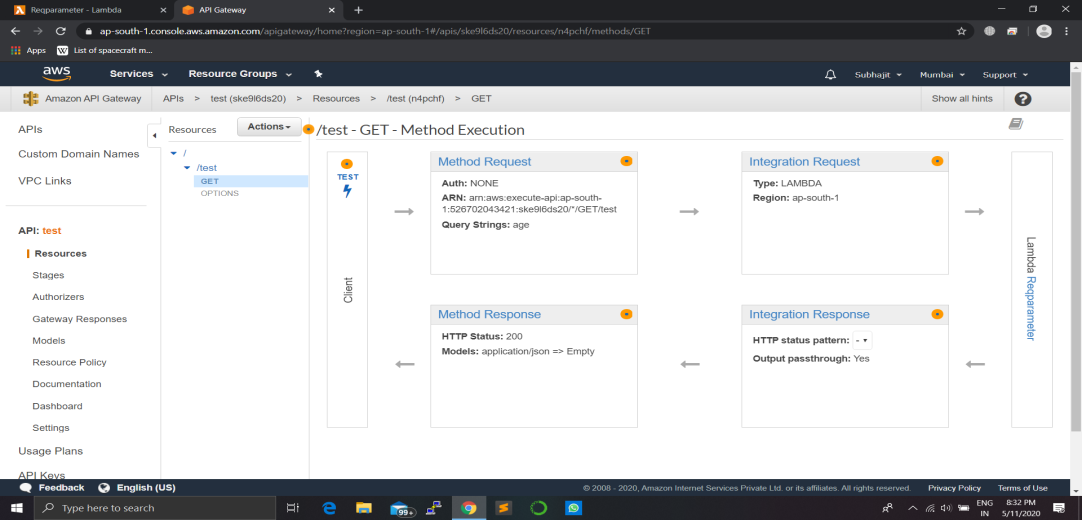
Step2:Coding

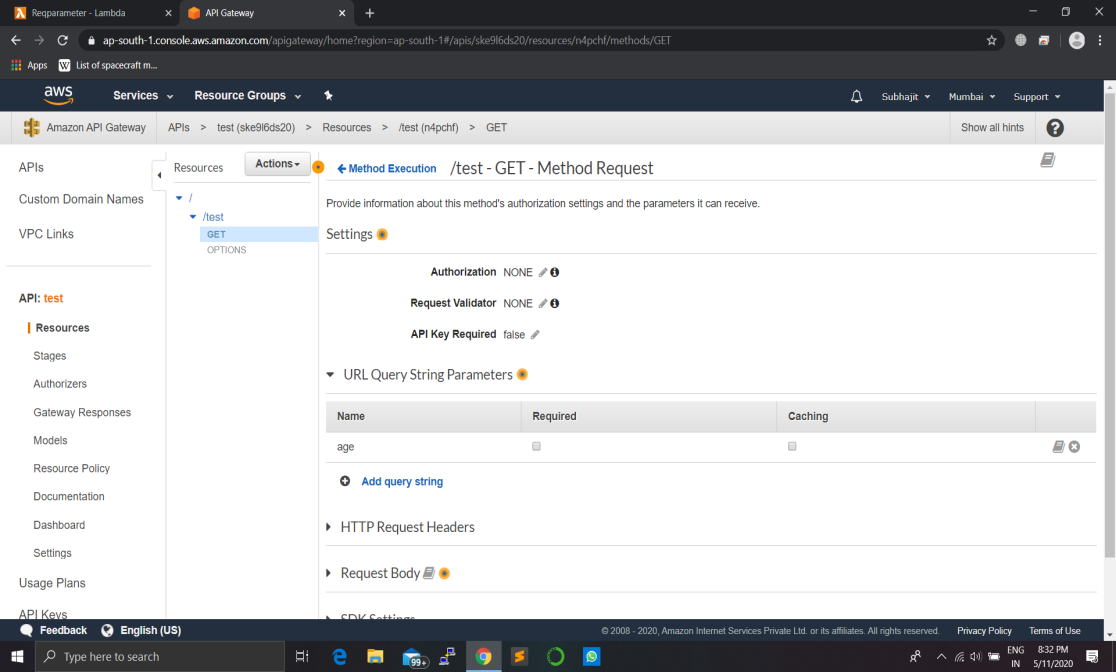
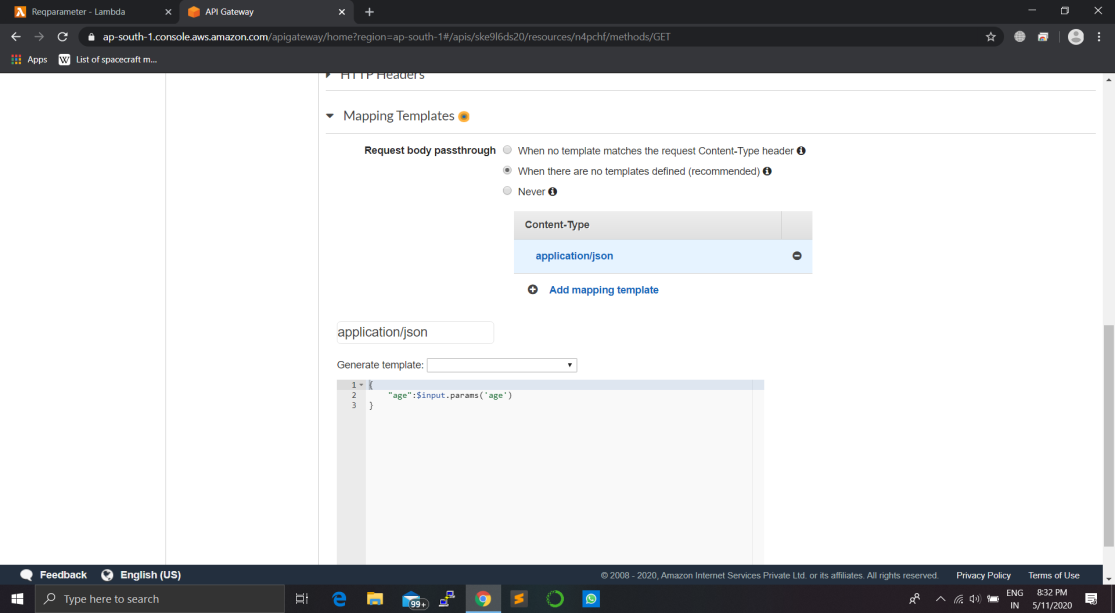
* In this step we will write code for our lambda function.

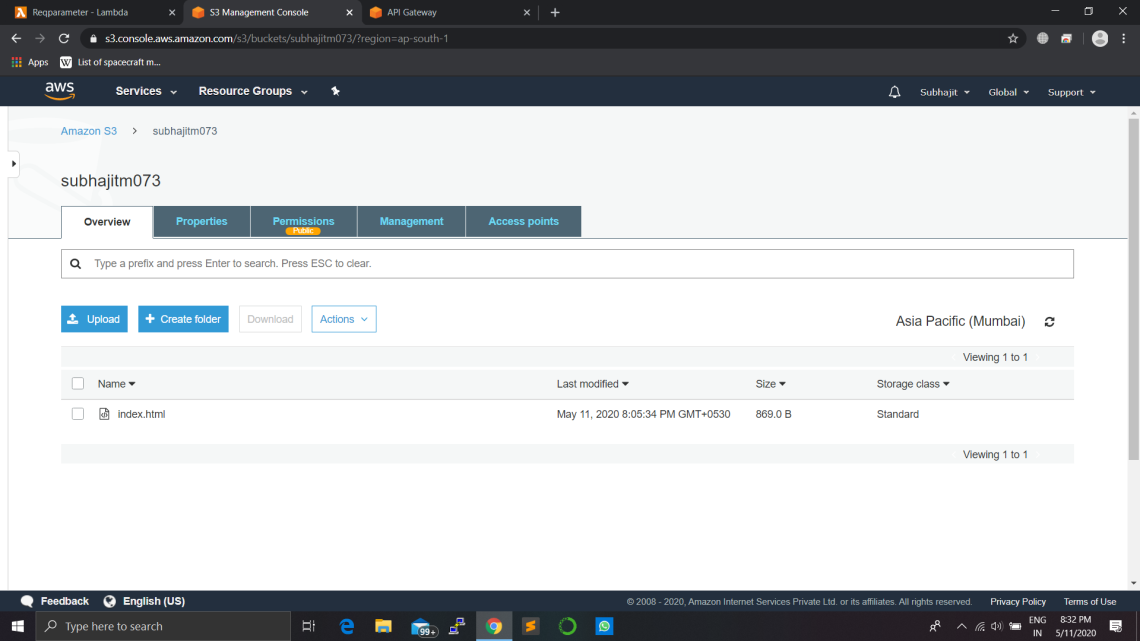


Step3:Deployment

* In this step we will be setting up out api-gateway to act as a middle man between our webpage and lambda function.
* Here we will be attaching lambda function to the api.
* Then we will be writing a application/json script to transfer our request.
* After the steps are done we will deploy our api.
* After deploying our api we will get the ARN to our api which we will use in our html code.
* Also we will upload our page to s3 bucket.

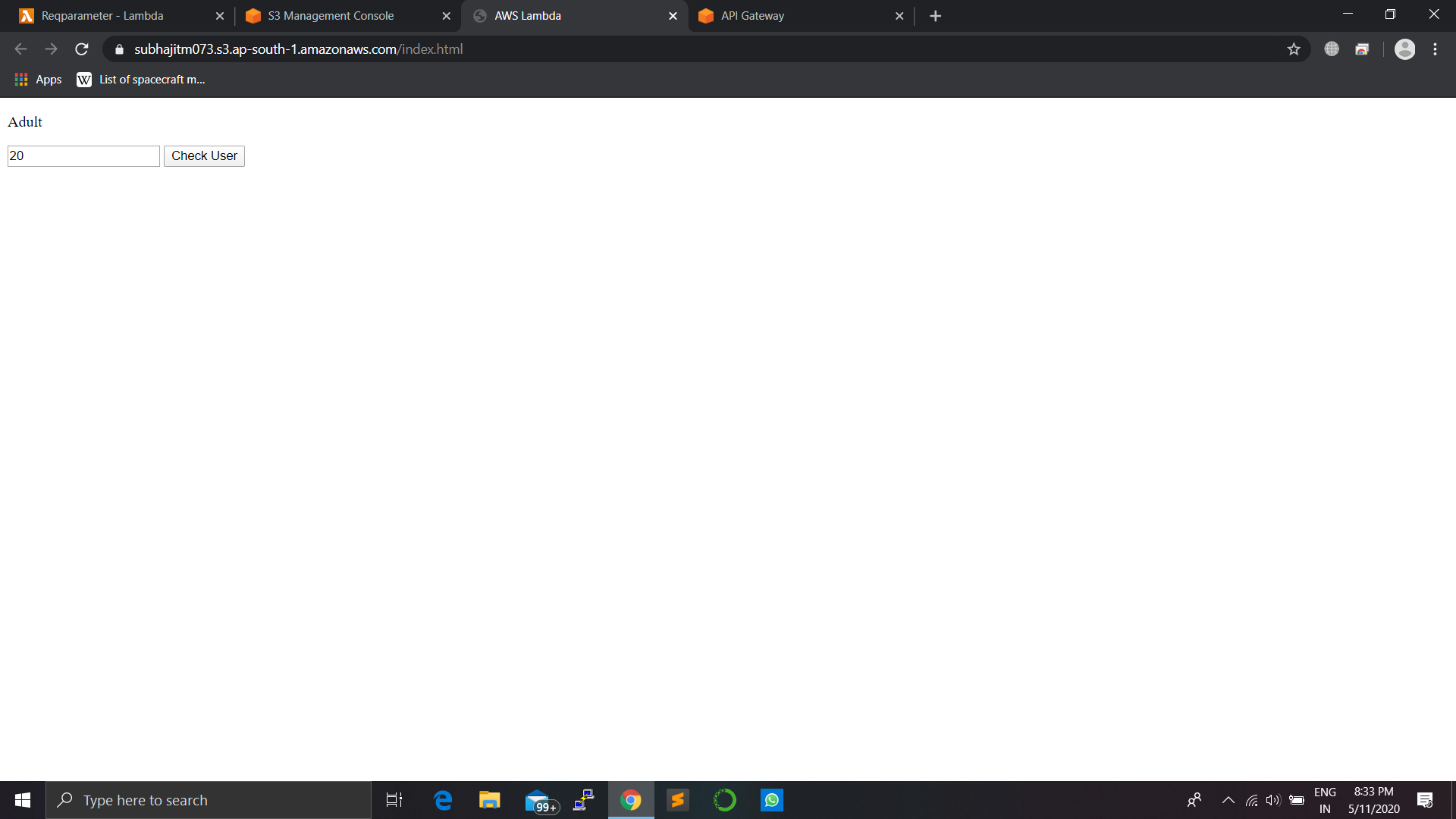
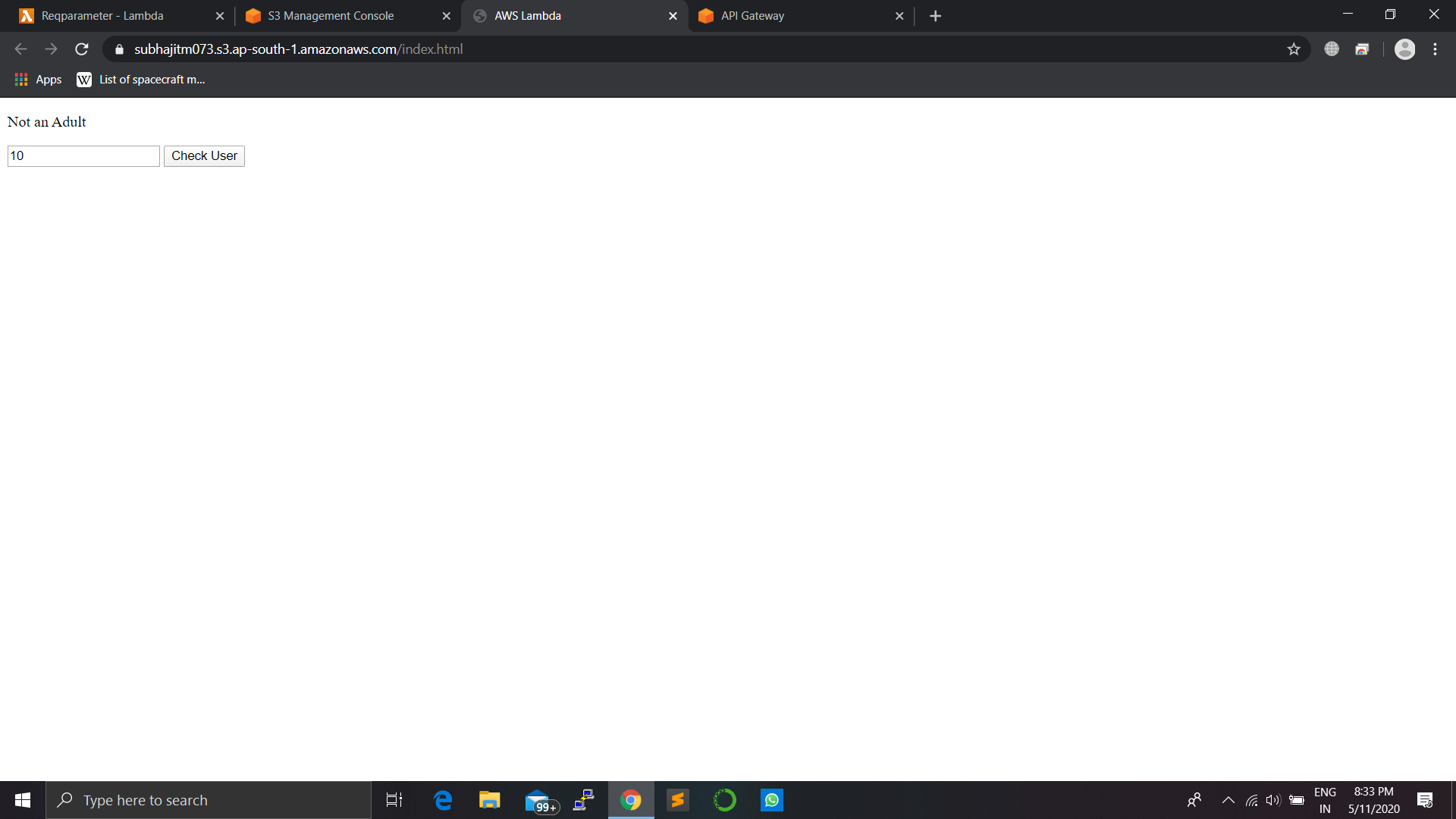






Step4:Testing

* In this stage we will be testing our project.



Step5:Clean up

* In this step we will delete all the services created like api-gateway, lambda function, S3 bucket, IAM role.

