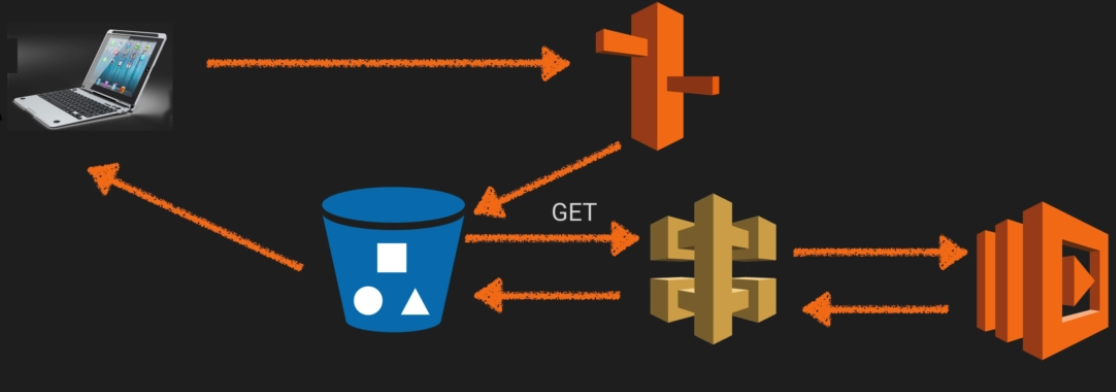
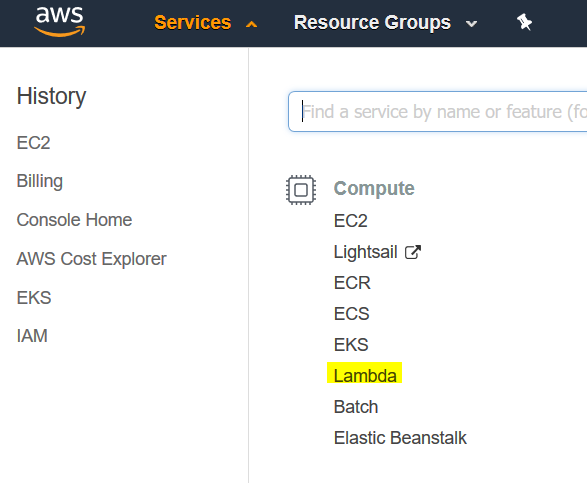
***Building a Serverless Page Using API Gateway and Lambda***

How to build a serverless page on cloud using different component without worrying about datacenters using a native code in the cloud.

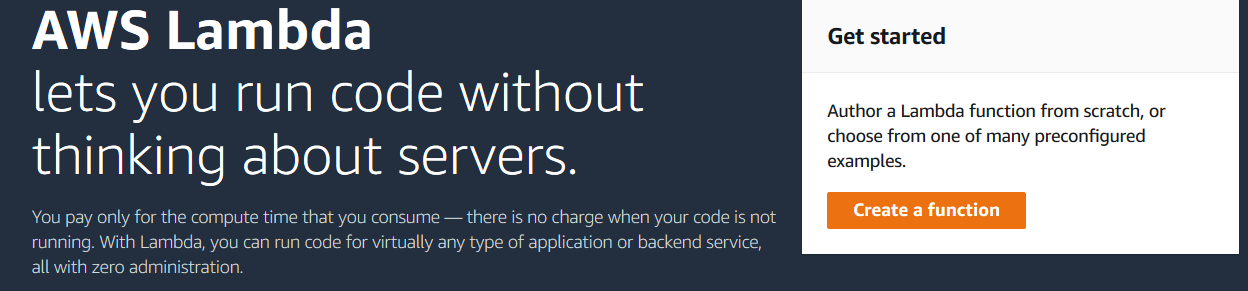


Lets see where we can do the lambda compute.

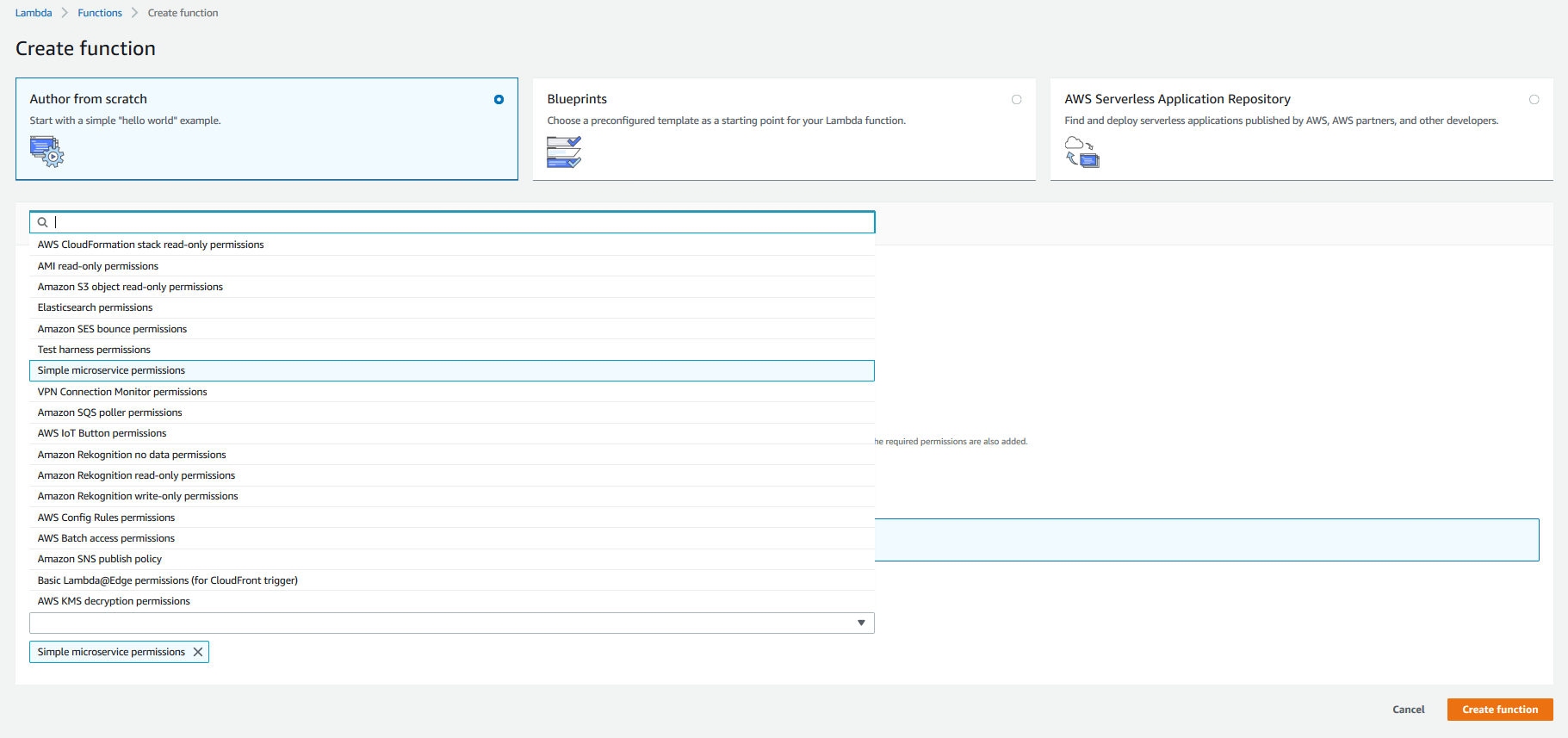


You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

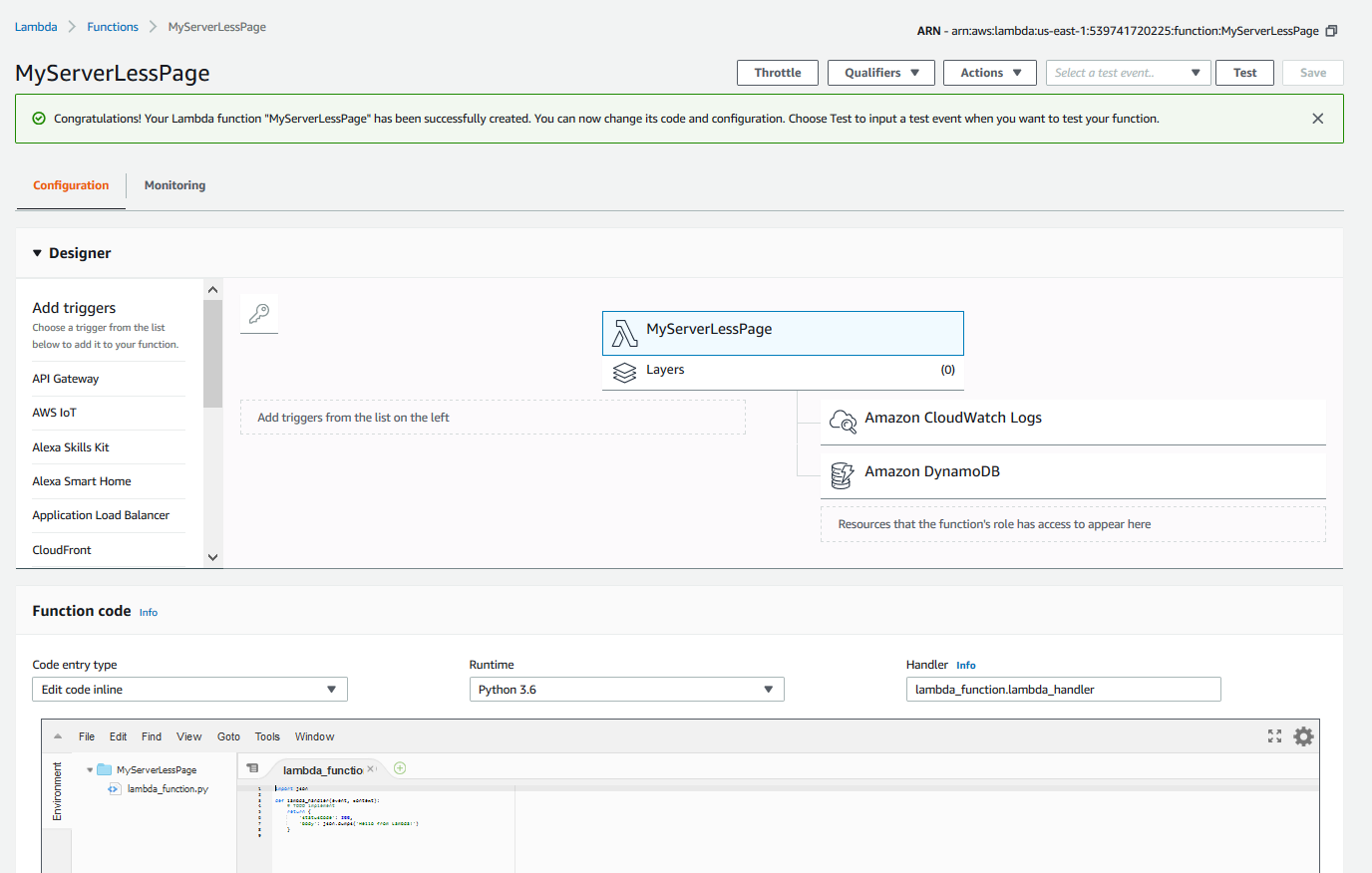
Lets create a function after getting in Lambda compute.

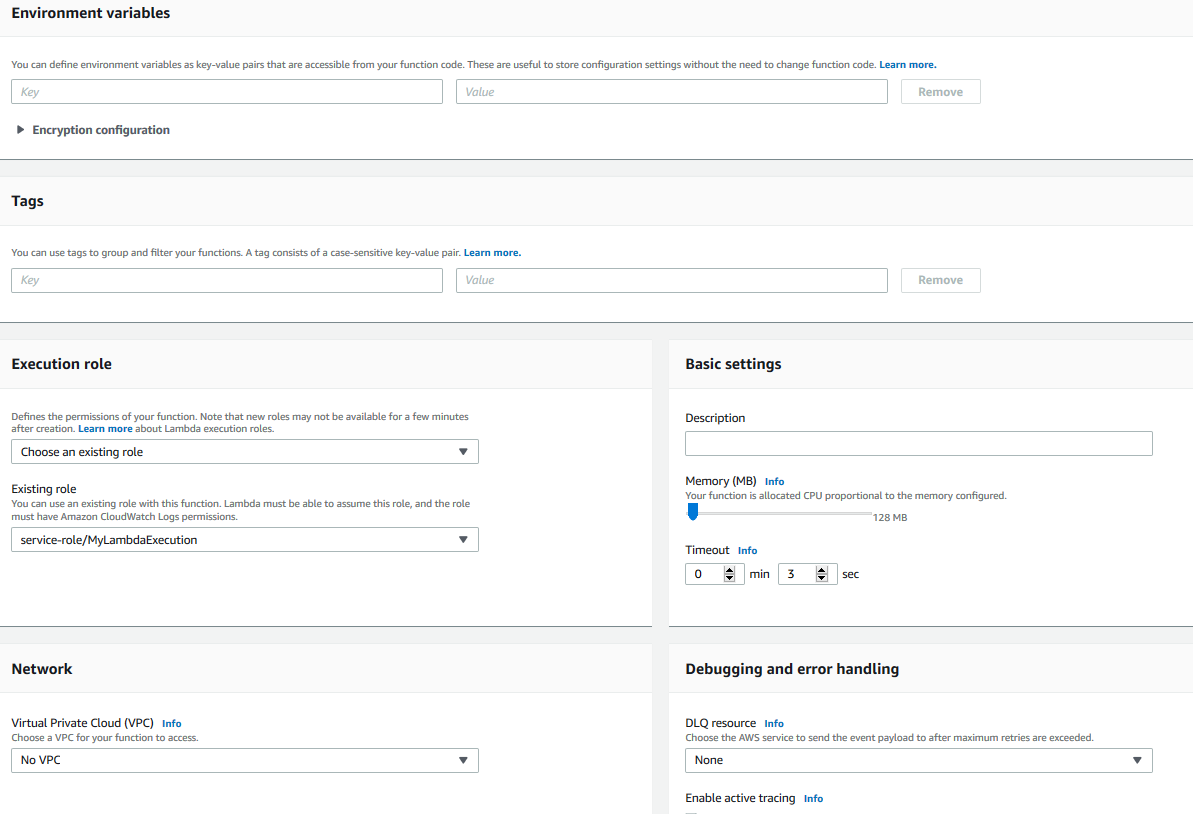


We will be building the page from scratch and fill the form in details as suggested in the image below.

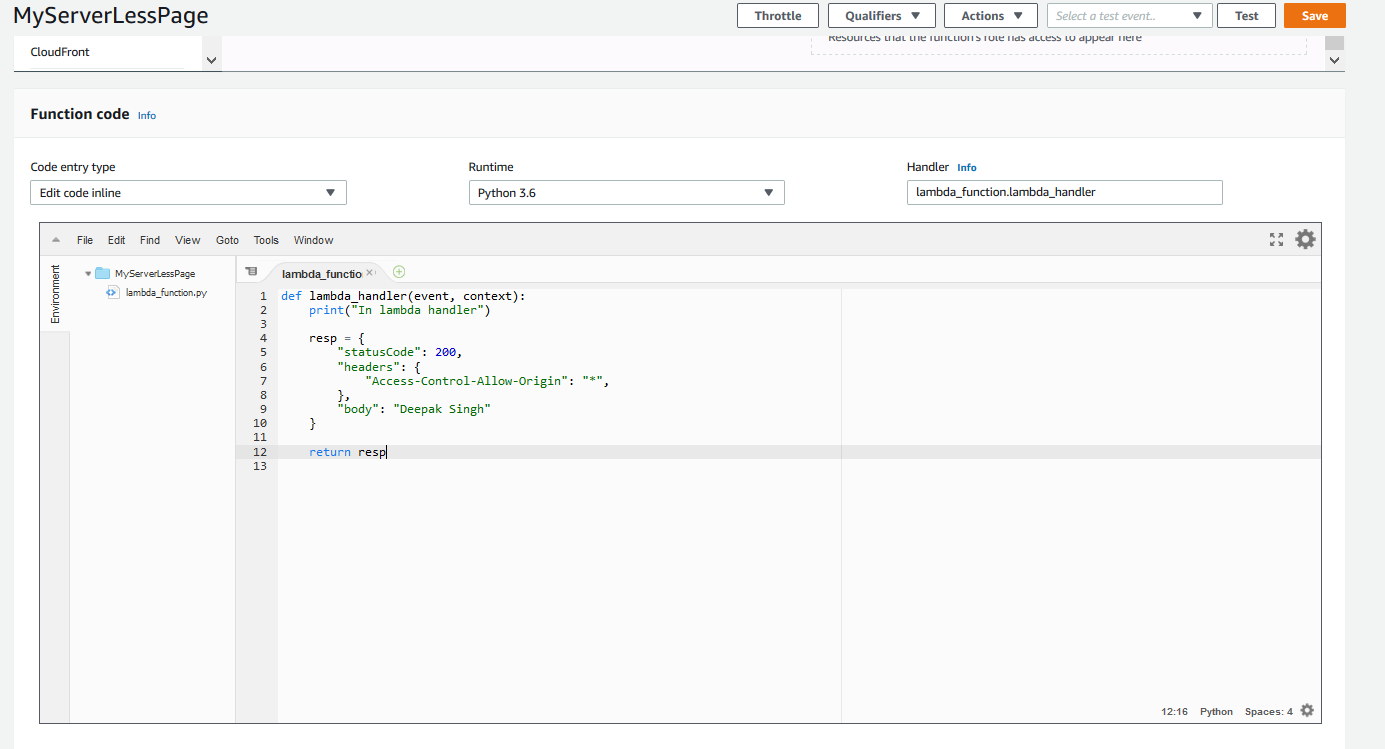


Once filled in the details click on create functions and you will be redirected to the Lambda MyServerLess page.

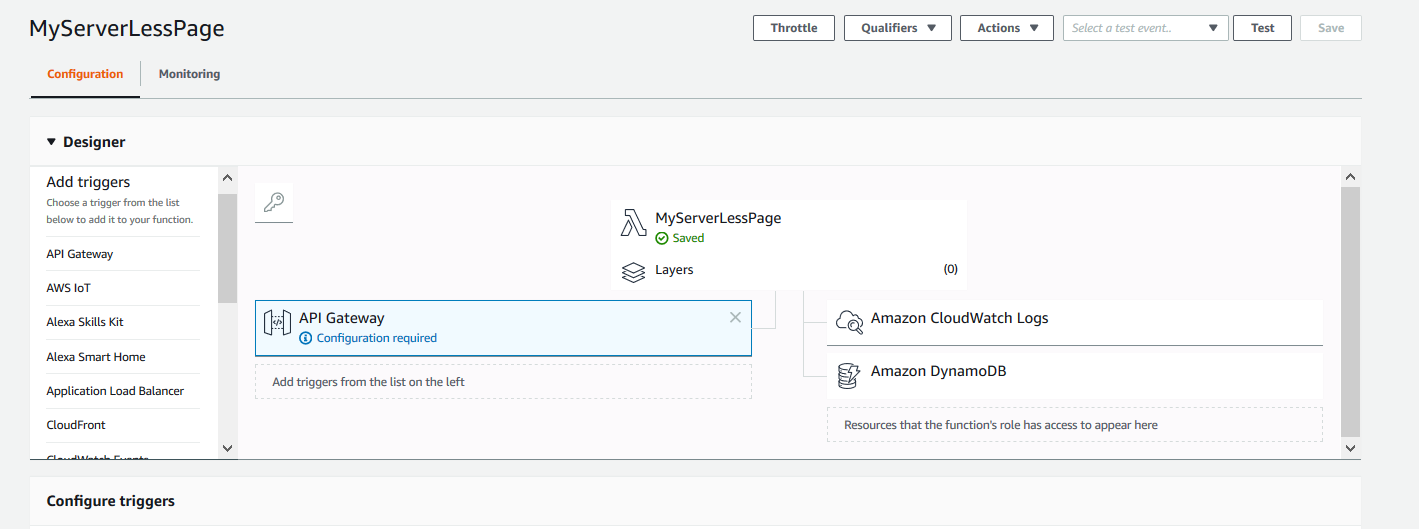




In the basic settings give description and paste the python code shared in the AWS lab folder in the code editor (function code) section and save it.

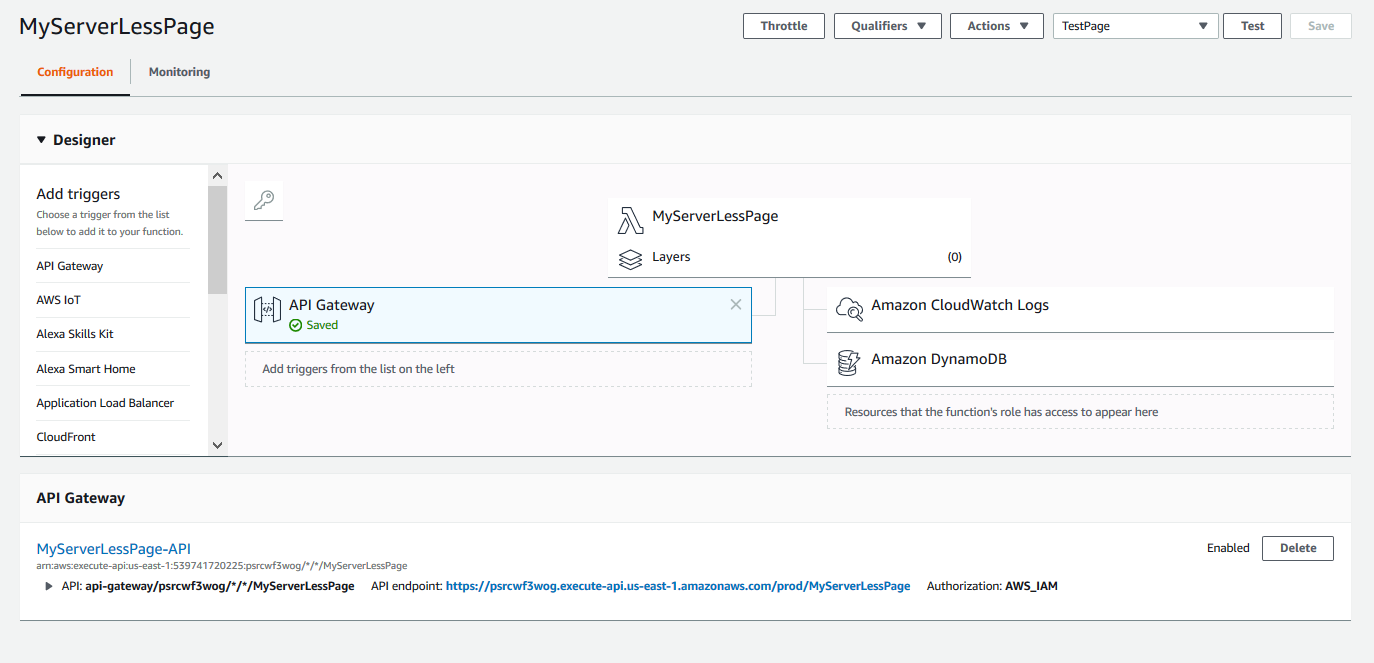


Now since we have save the code we need to trigger. In lambda we need to understand what triggers are. We can use anyone of the triggers specified in the designer section, we are using API gateway as the trigger.



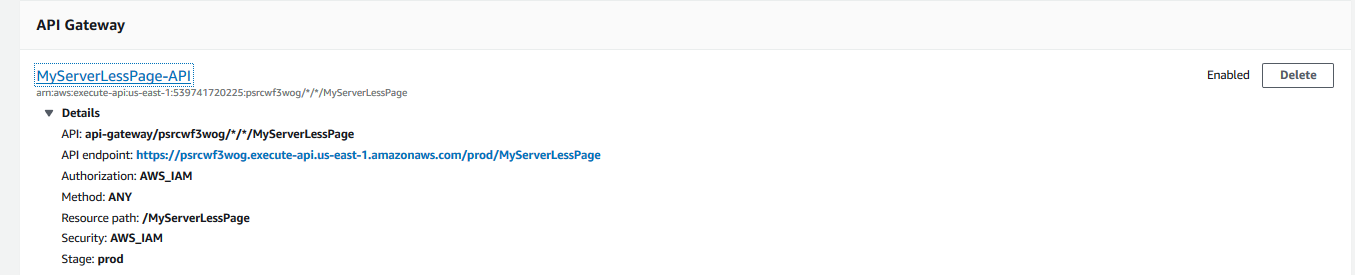
Do the default configuration for triggers.

Select the existing API and select stage as “prod” and security as “AWS IAM” and save the changes.



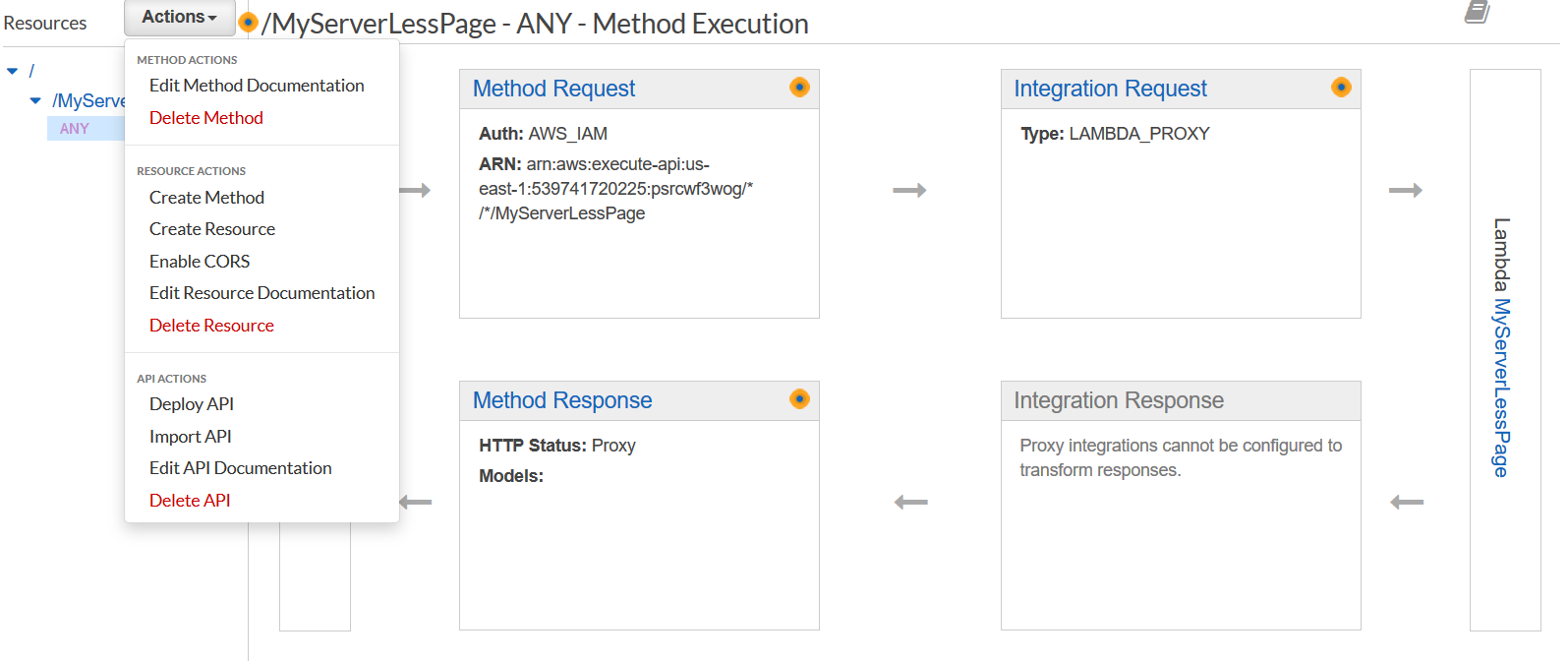
After that you will get an API endpoint.

Now Lets go to API get way section and click the dropdown.

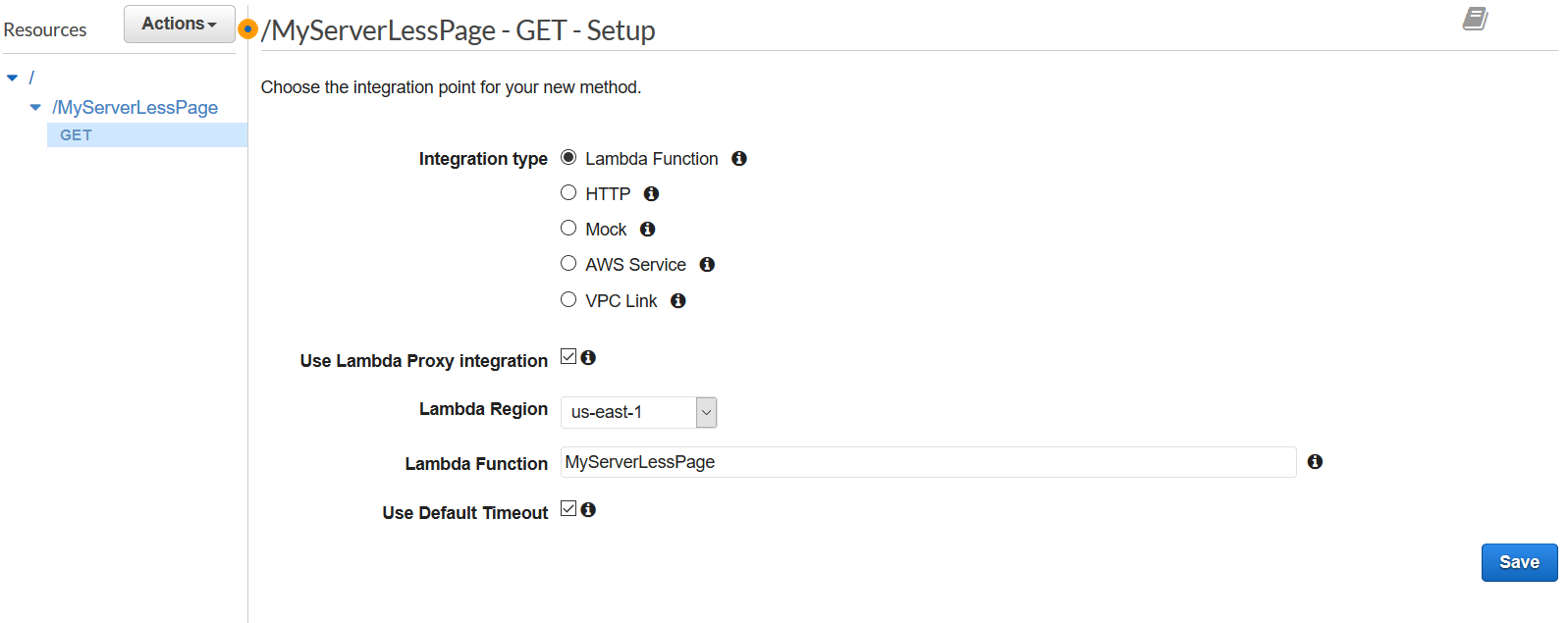


Click on “MyServerLessPage-API” or the name you have given during lambda creation. It will take to API gateway method section.

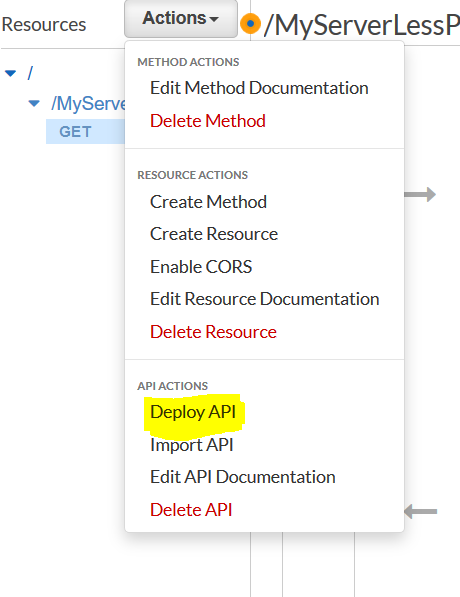
We will not be using method ANY, so we will delete ANY method and create a method and use the GET method.

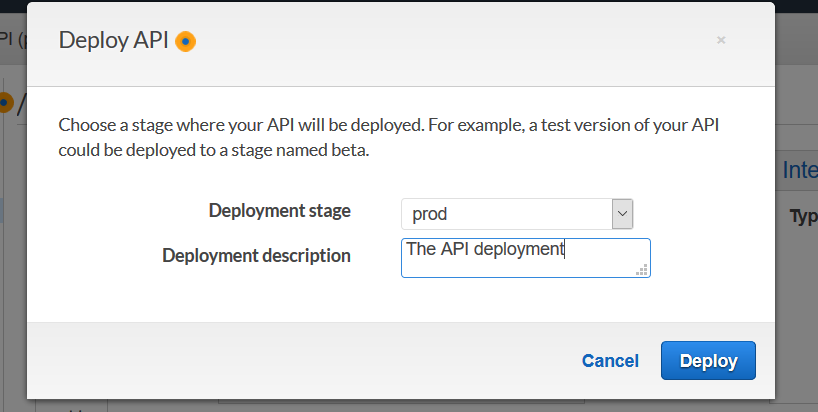


Delete Any and Create method, select GET and fill the configuration.

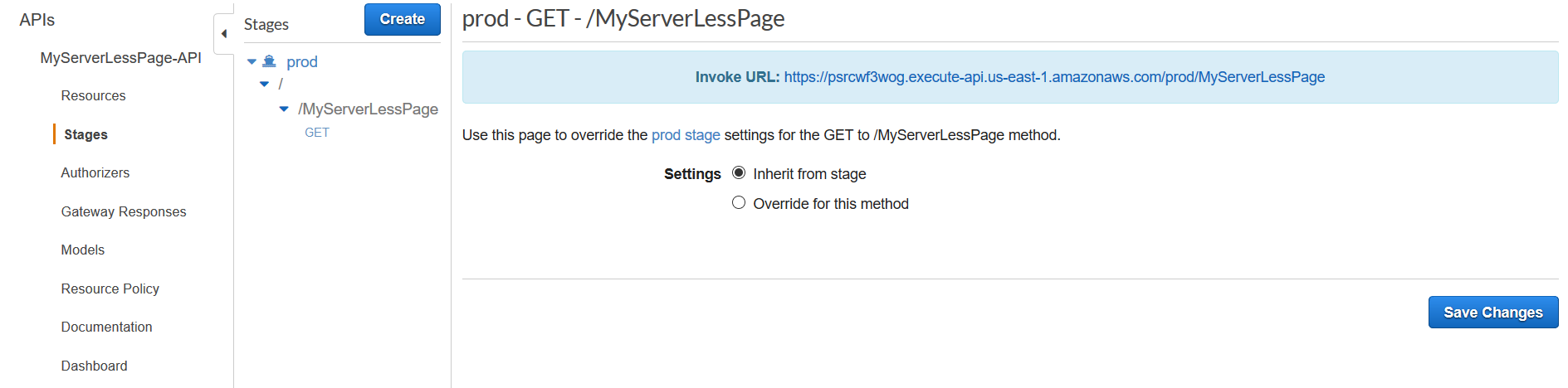


Now since we have defined the method we need to deploy the API in the prod stage. From drop down select deploy API



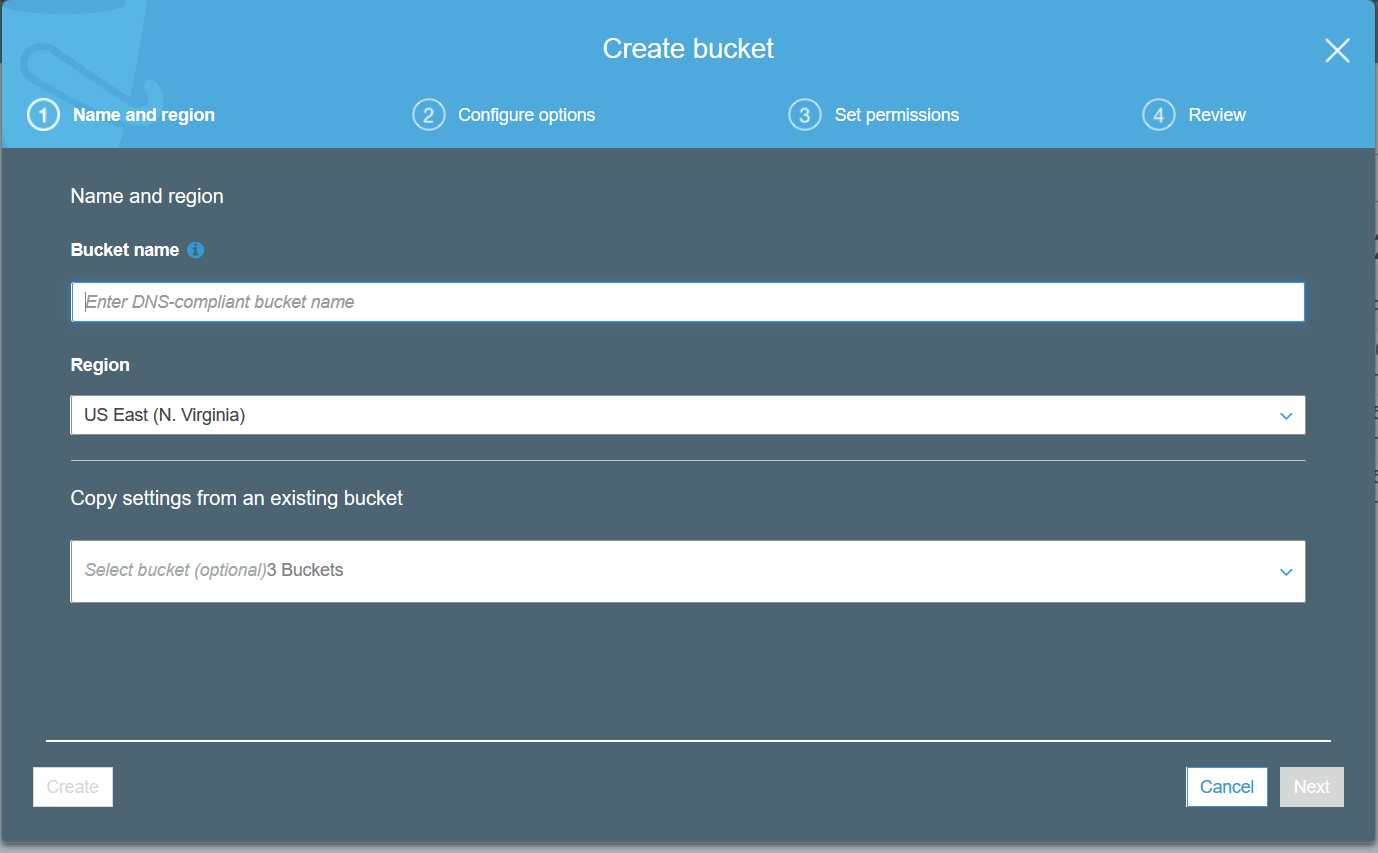


Click on Deploy and you will be on stages section of the page.



Click on the invoke URL and you will be able to see the name written in the script.

Now lets create S3 bucket. Got Services and create bucket.



After creating the bucket Use the index.html that is in dropbox and chage the Invoke URL with yours.

*<html>*

*<script>*

*function myFunction() {*

*var xhttp = new XMLHttpRequest();*

*xhttp.onreadystatechange = function() {*

*if (this.readyState == 4 && this.status == 200) {*

*document.getElementById("my-demo").innerHTML = this.responseText;*

*}*

*};*

*xhttp.open("GET", " https://psrcwf3wog.execute-api.us-east-1.amazonaws.com/prod/MyServerLessPage", true);*

*xhttp.send();*

*}*

*</script>*

*<body><div align="center"><br><br><br><br>*

*<h1>Hello <span id="my-demo">Cloud Gurus!</span></h1>*

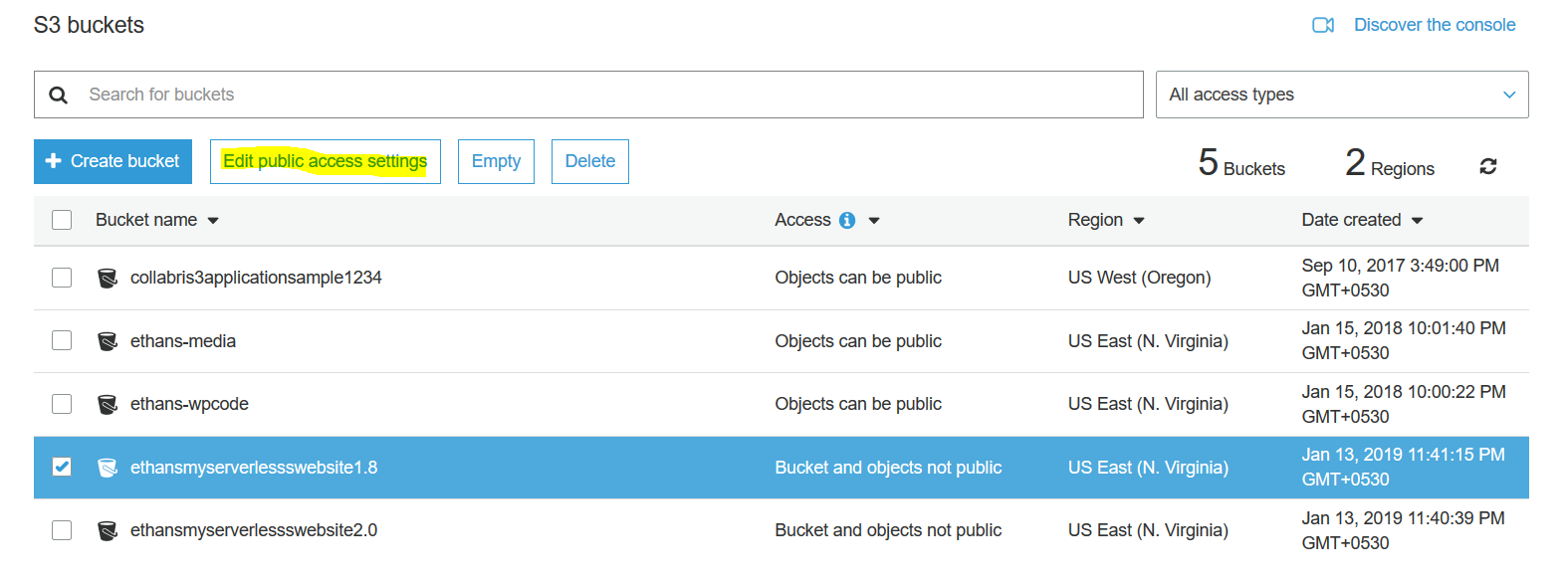
*<button onclick="myFunction()">Click me</button><br>*

*<img src="https://s3.amazonaws.com/acloudguru-opsworkslab-donotdelete/ACG\_Austin.JPG"></div>*

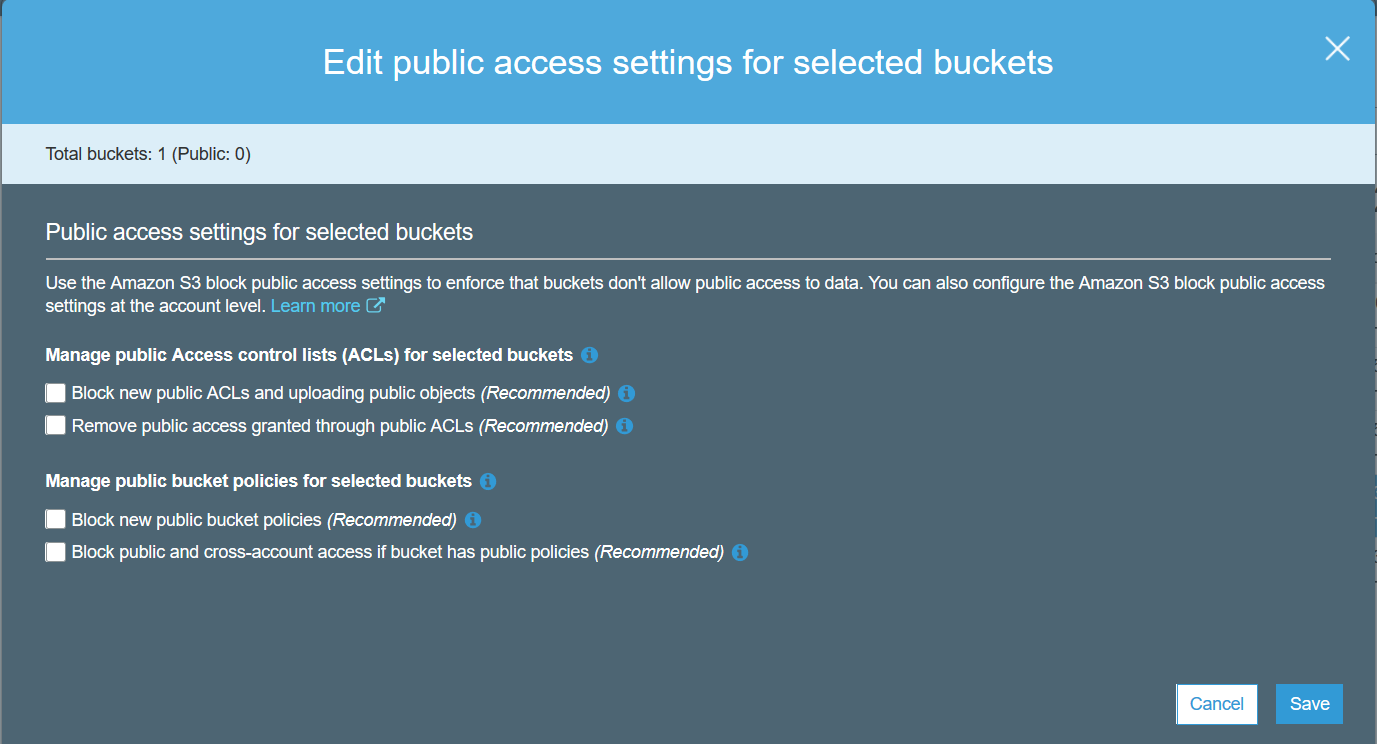
*</body>*

*</html>*

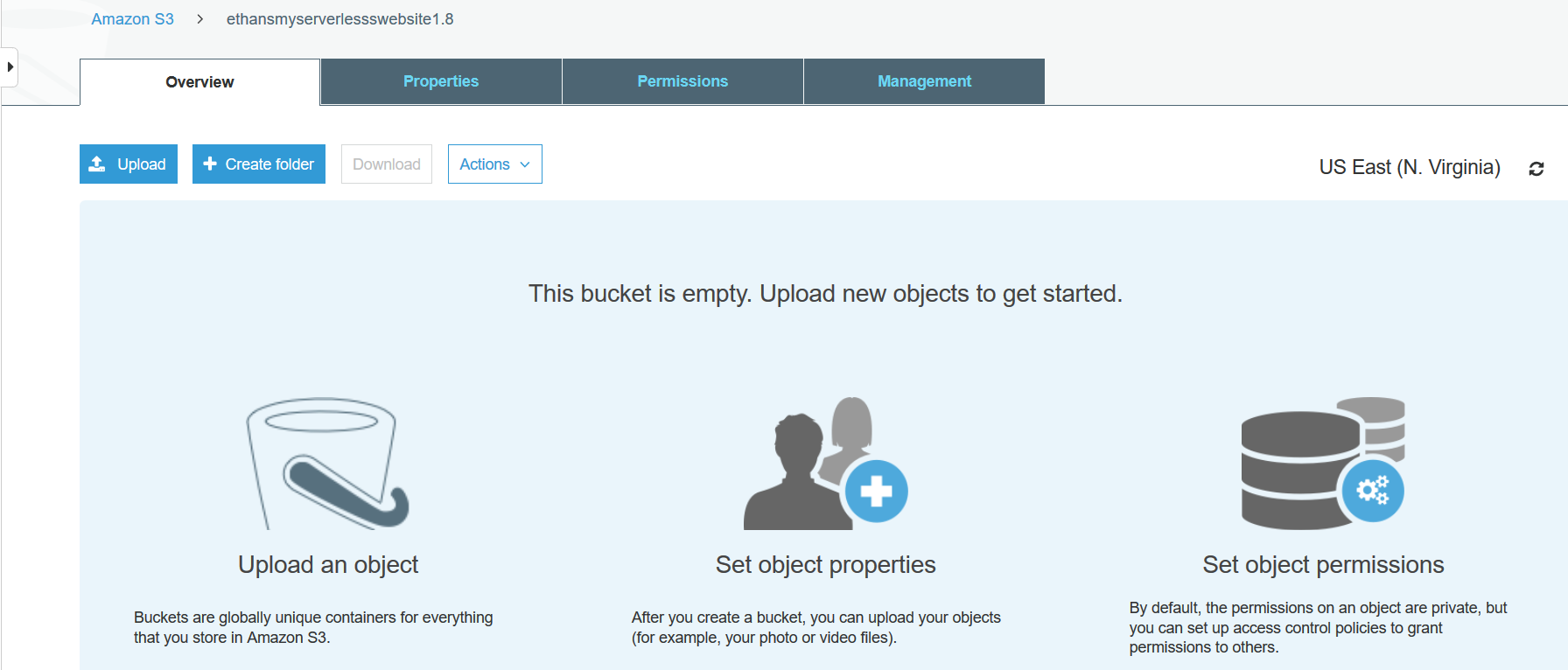
We need to upload this script to S3 bucket, go to S3. And make the bucket public first.



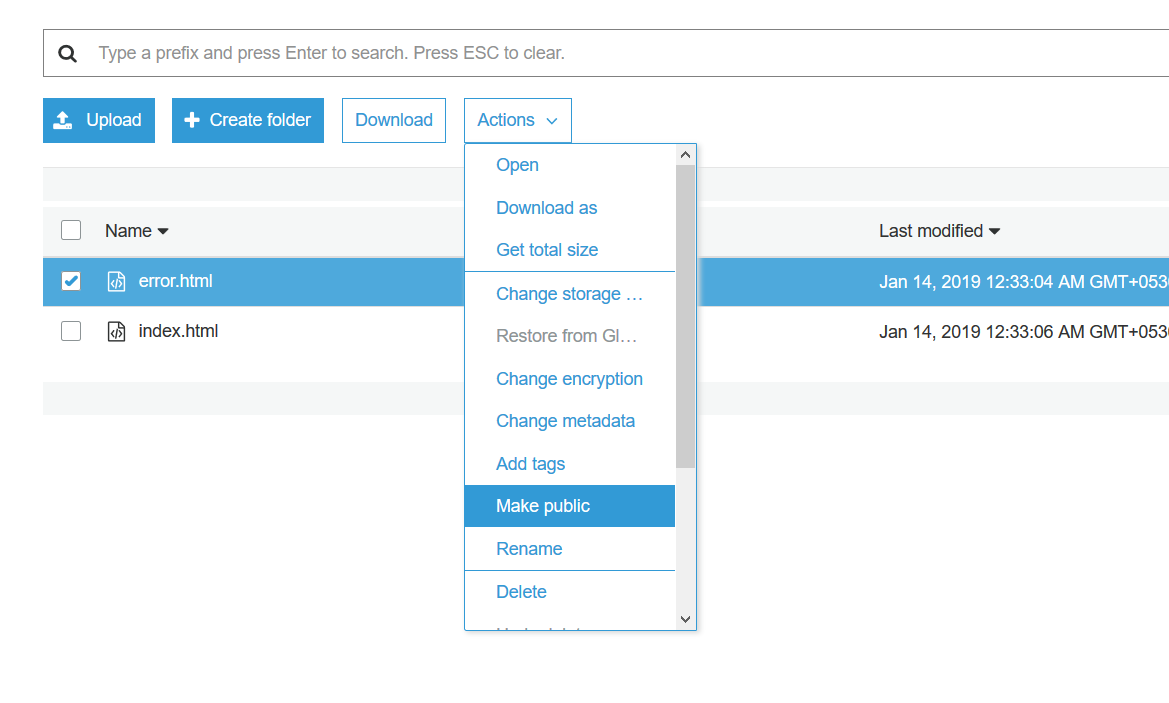
After click on Edit public access setting, in the page do not check any boxes and save and then just write confirm to make the bucket public:



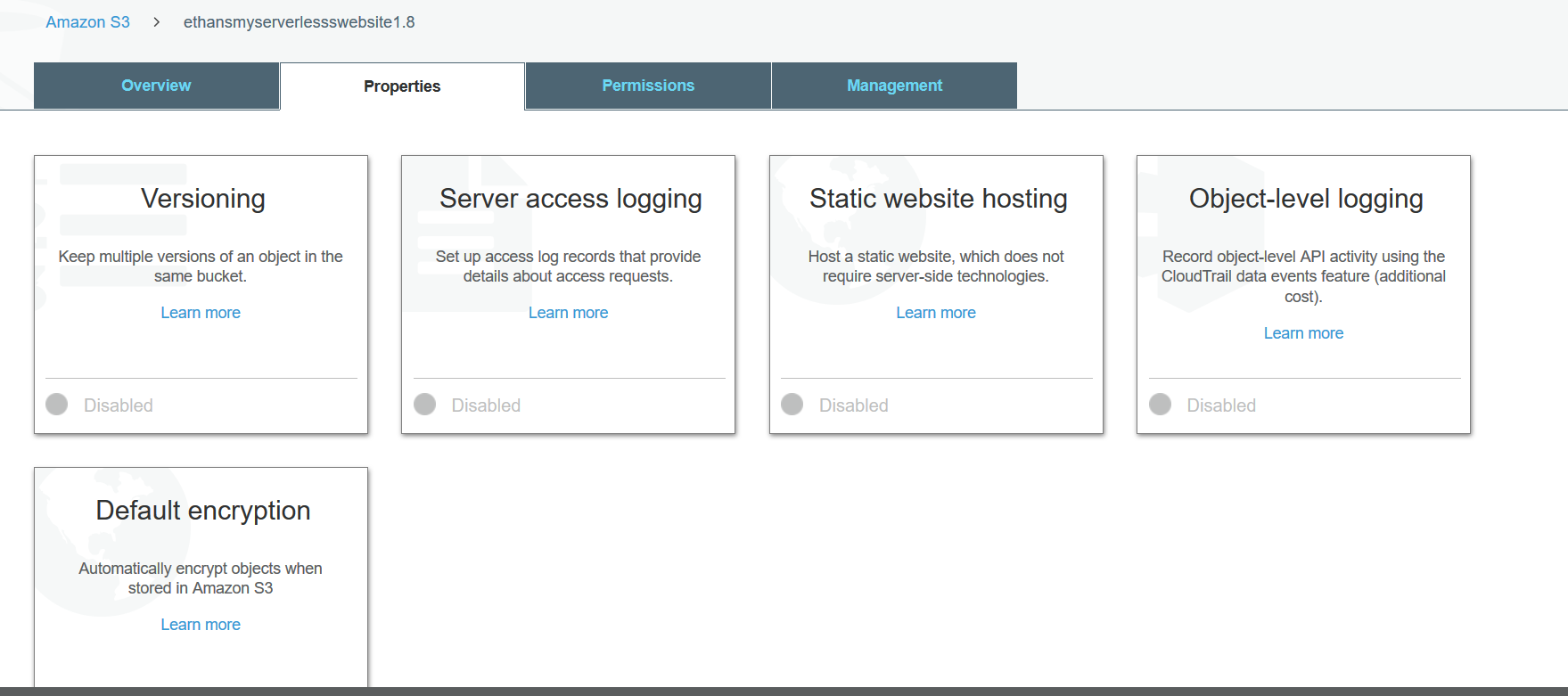
Once public click over the bucket and upload the html files.



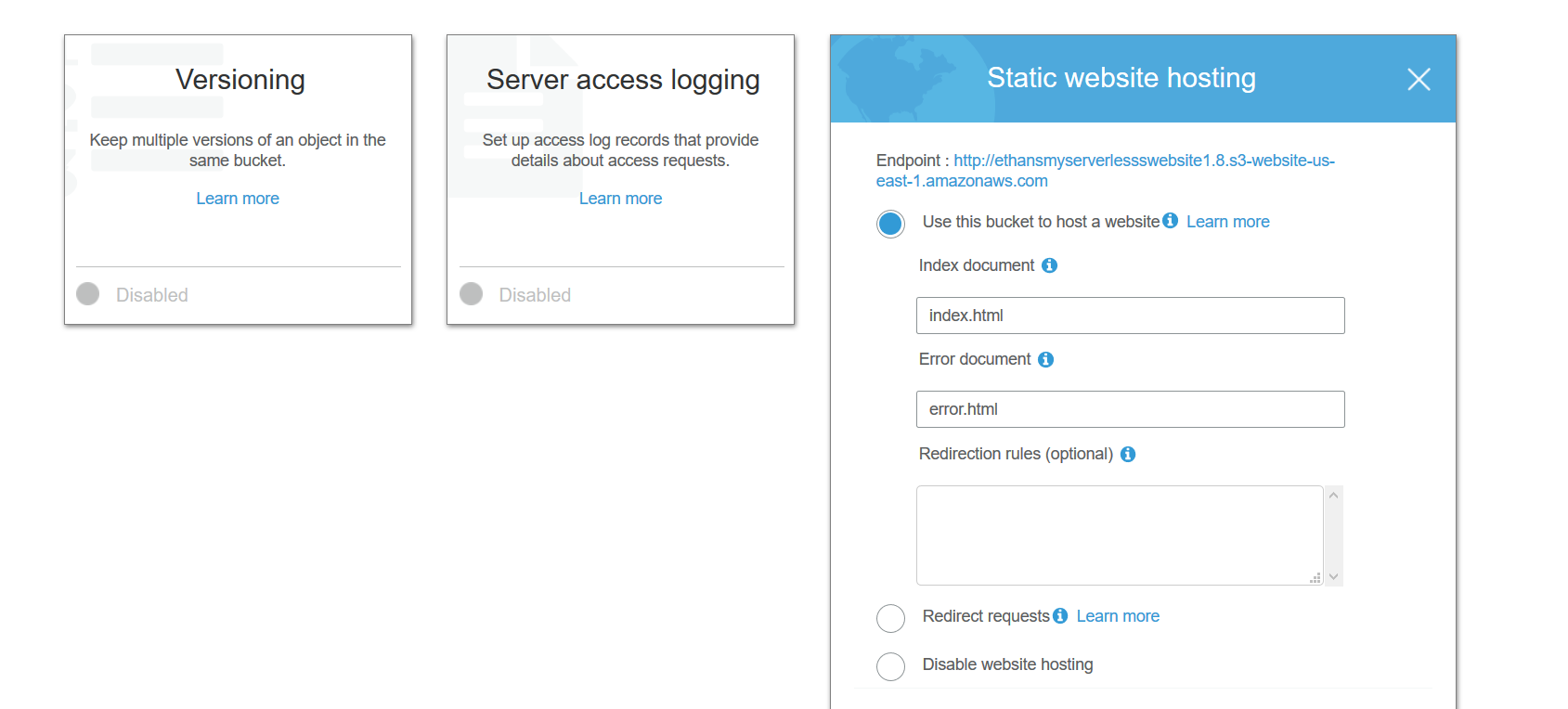
Click on upload to upload the files and then make both the files public one by one.



To make this html pages accessible from a website we need to go to properties and select Static web site hosting.



Configure static website hosting and select use this bucket to host a website and type the name of index.html and error.html.



Once done your serverless url is ready and you can click over to see if it loads the lambda function you have created. If you click on “click me” in the page you will see the lamda function is getting called.

