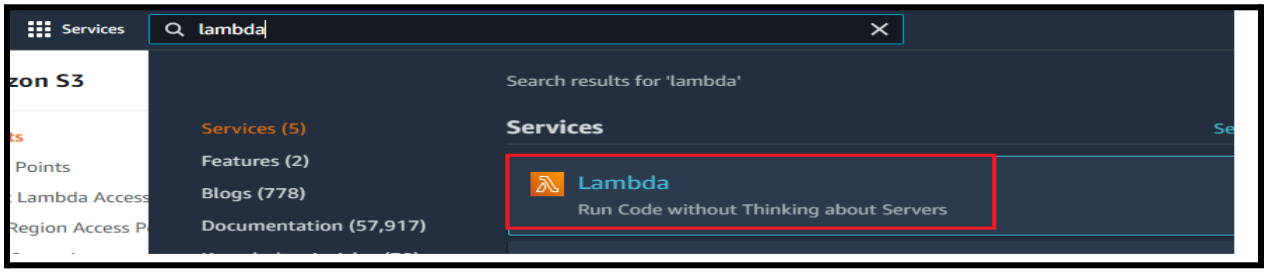
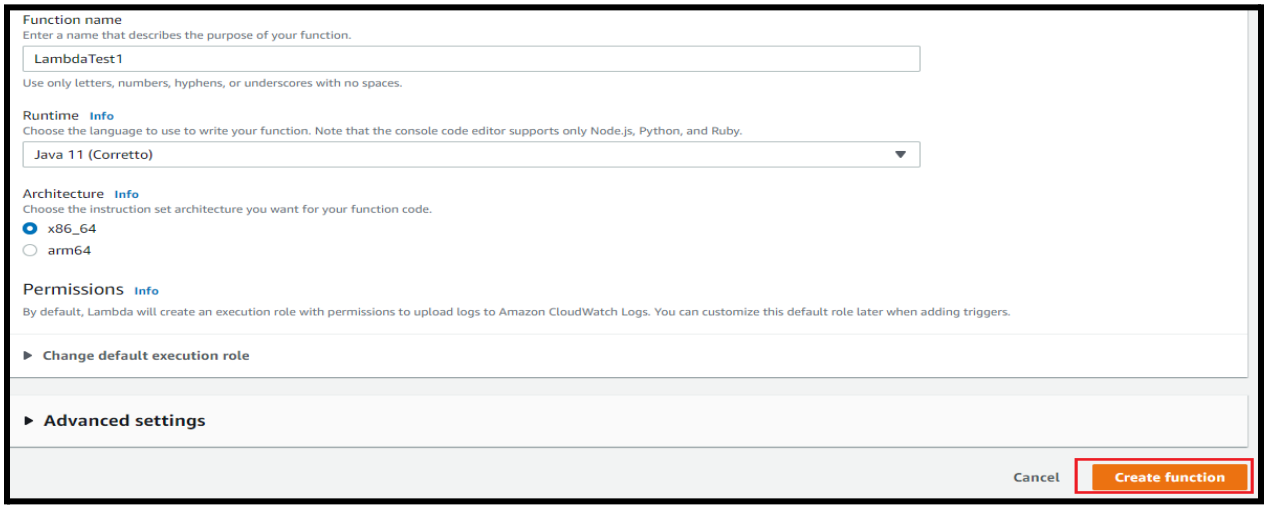
Create Lambda Function:

- Login to AWS Console. Click on Services, Search for Lambda, and open Lambda service.

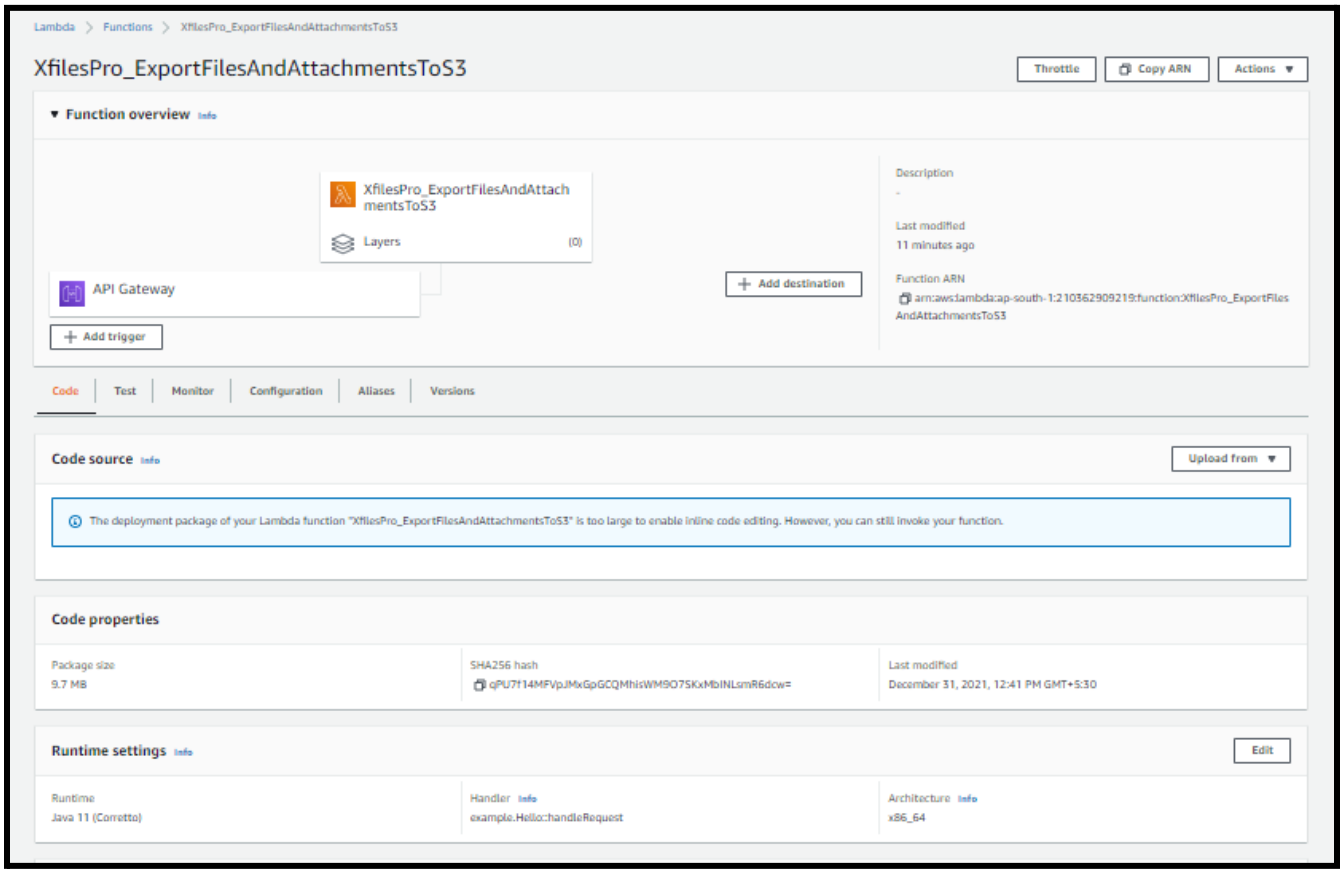


- Click on the **"Create Function"** button

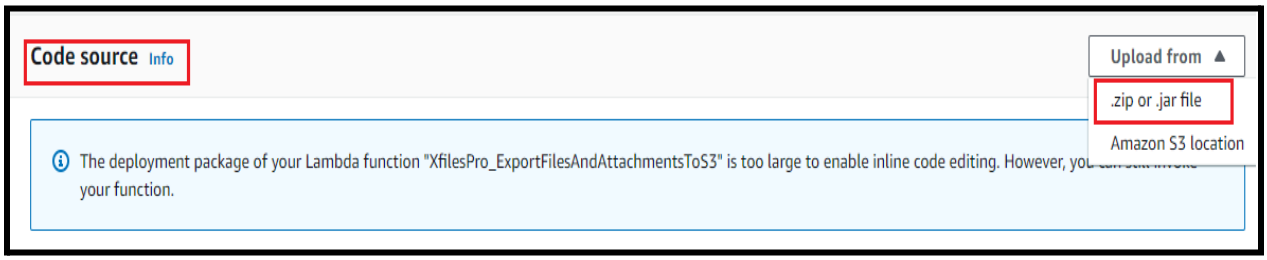
- Provide the Basic Information:
  - **Function Name** - Provide any name
  - **Runtime** - Select Java 11(Corretto) from the dropdown.
  - Click on the **'Create Function'** button.



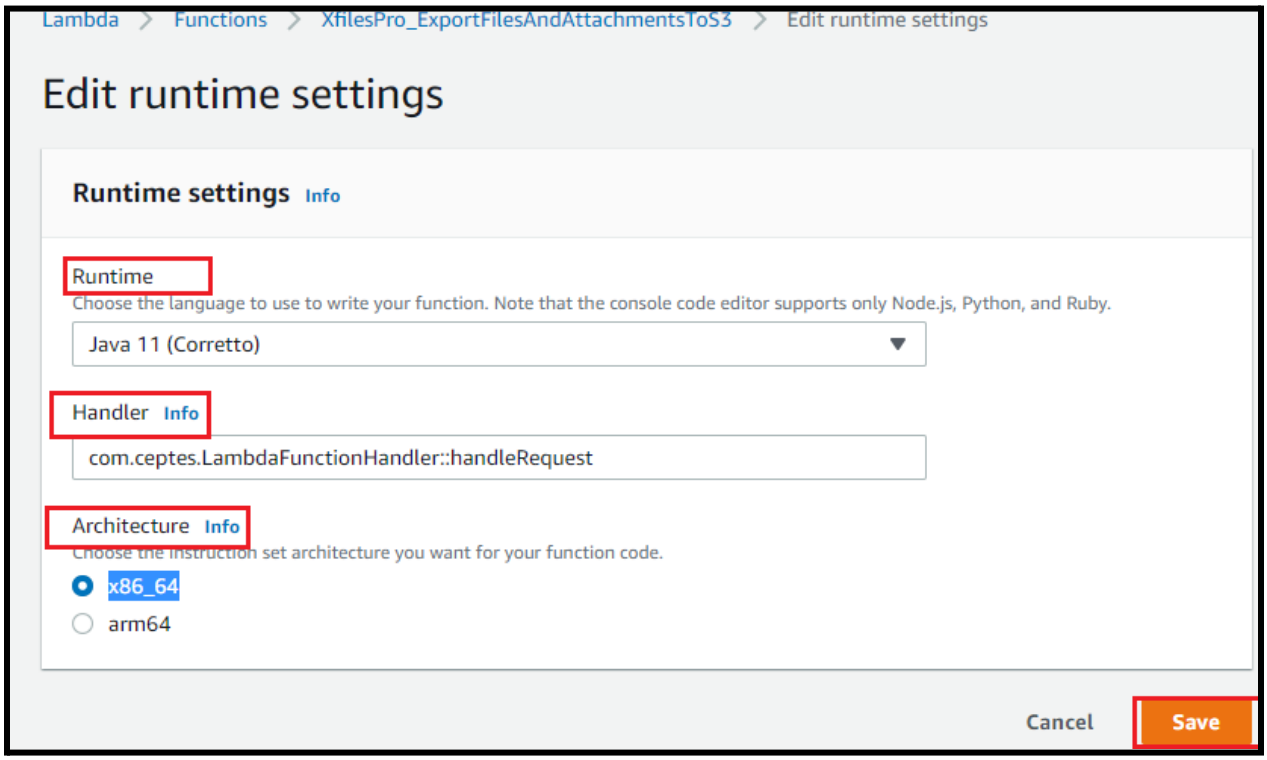
- Function gets created and the below screen appears on the same screen.



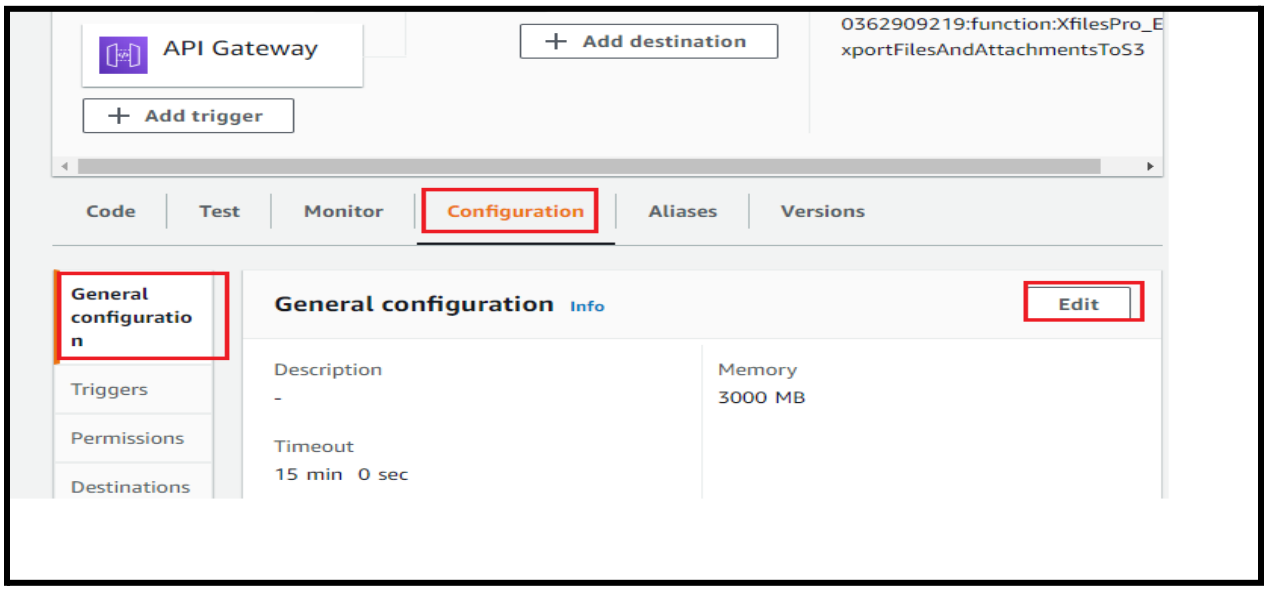
- From the **Code Source** section, click on **Upload from** (the dropdown button) and select **.zip or . jar file**

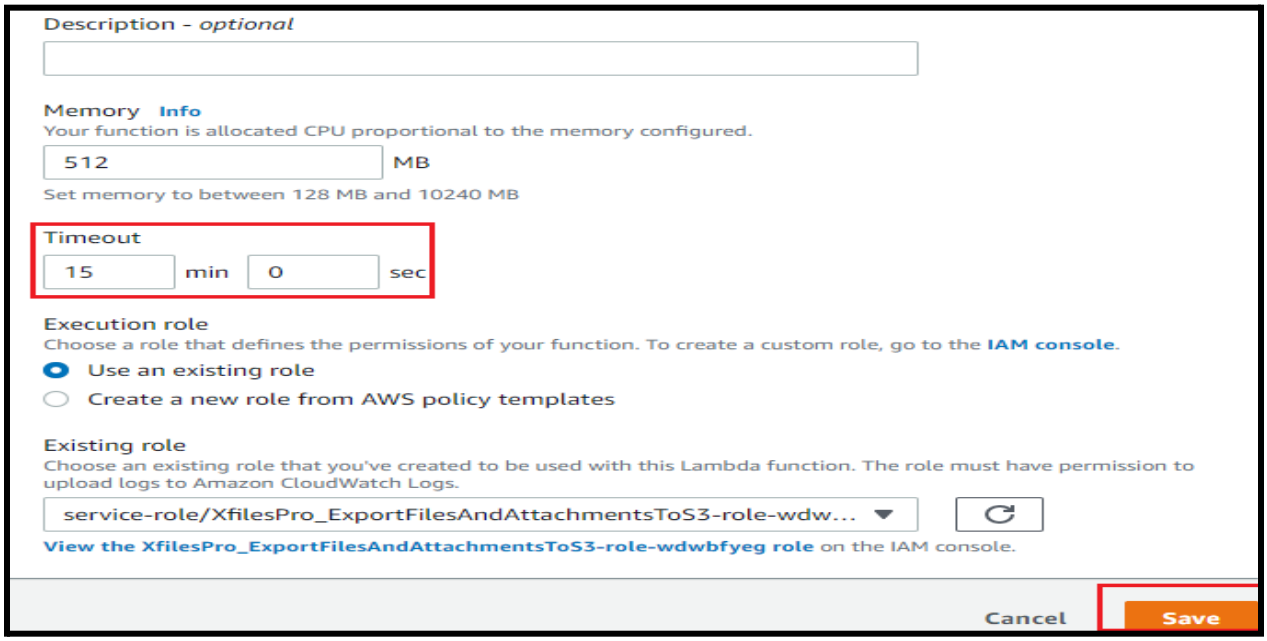


- From the **Runtime Settings** section, click on the **Edit** button.
    - Select the Runtime as **Java 11 (Corretto)**
    - The handler should be given as **"com.ceptes.LambdaFunctionHandler"**
    - Architecture as **x86_64**
    - Click on the **Save** button.



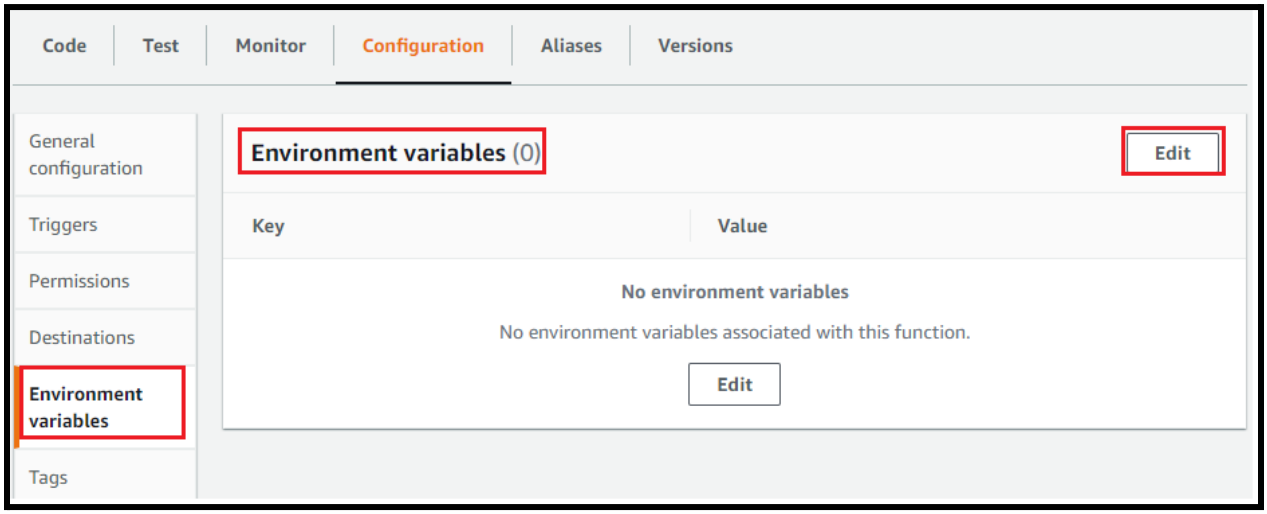
- Click on **Configuration tab** and from the navigation section select **General Configuration** and Click on **Edit** button.
    - Let the memory be 512MB
    - Provide the Timeout as **15 min**
    - Click on **Save** button



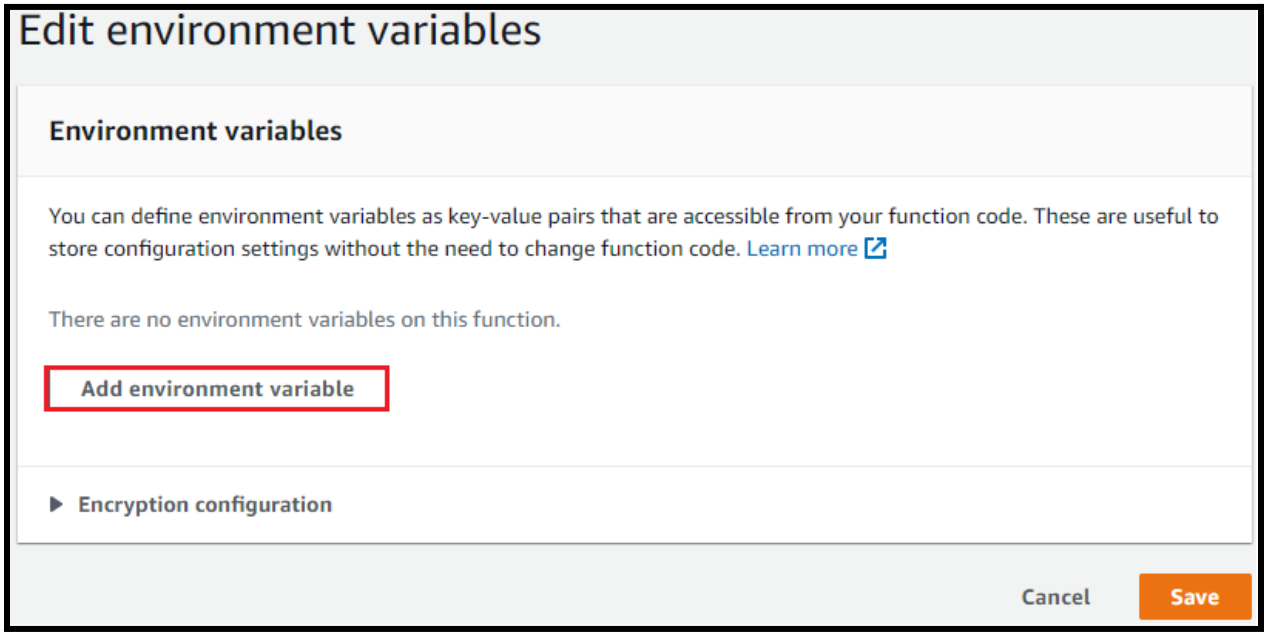


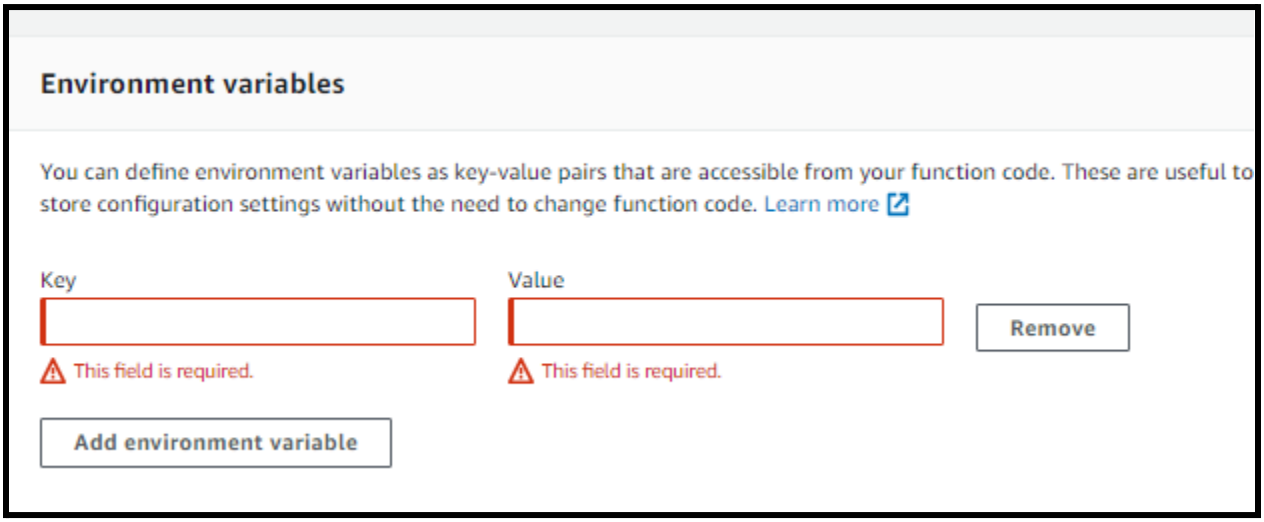
Add Environment Variables

- From the same **Configuration** tab, in the navigation bar select **Environment Variables** and click on **Edit** button.



- Click On **Add environment variable** and add the following variables (listed in the table).





Please provide the **Keys** and **Values** for the following environment variables:

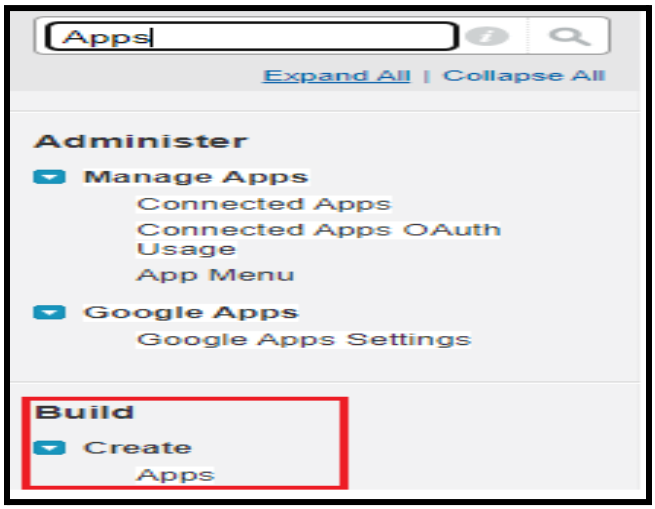
Key	Value
XFILES_ConnectedApp_ConsumerKey	Consumer Key of the connected App (see steps below to get the value)
XFILES_ConnectedApp_ConsumerSecret	ConsumerSecret of the Connected App (see steps below to get the value)
XFILES_SFDC_UserName	Salesforce User Name
XFILES_SFDC_Password	Salesforce Password and Security Token (see steps below to get the value)
XFILES_SFDC_Environment	Salesforce login URL
XFILES_S3_AccessKey	S3 Access Key
XFILES_S3_SecretAccessKey	S3 Secret Access Key
XFILES_SFDC_Namespace	XFILES

Key	Value	
XFILES_ConnectedApp_ConsumerKey	[redacted]	Remove
XFILES_ConnectedApp_ConsumerSecret	[redacted]	Remove
XFILES_S3_AccessKey	[redacted]	Remove
XFILES_S3_SecretAccessKey	[redacted]	Remove
XFILES_SFDC_Environment	https://login.salesforce.com	Remove
XFILES_SFDC_Namespace	XFILES	Remove
XFILES_SFDC_Password	[redacted]	Remove
XFILES_SFDC_UserName	xfpreprodqa@xfiles.com	Remove

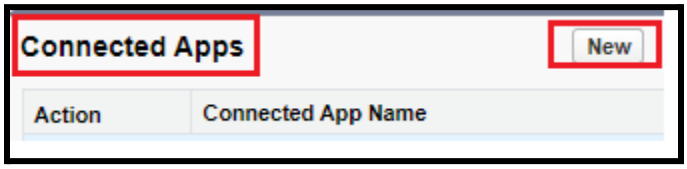
- Click On **Save** Button.

How to get the **ConsumerKey** and **ConsumerSecret**

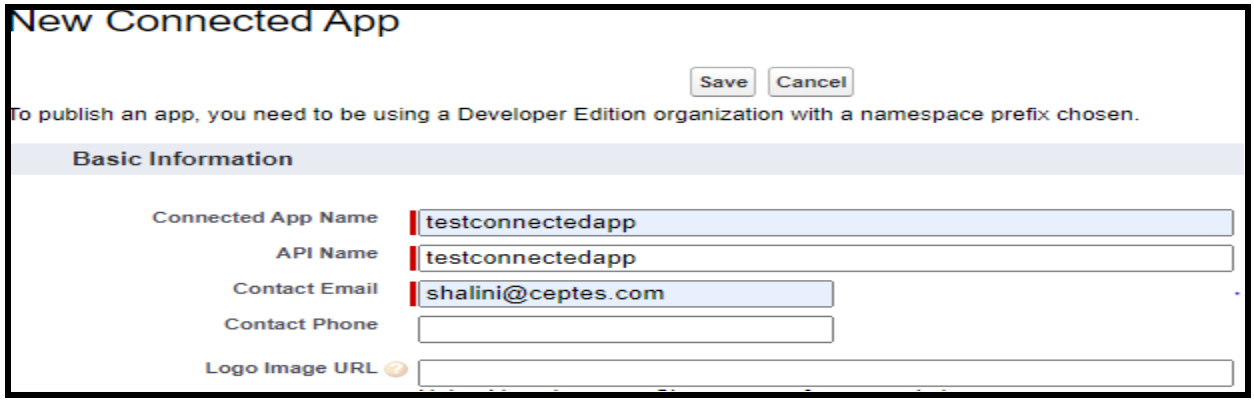
1. In Quick find search for **Create** and click on **Apps**



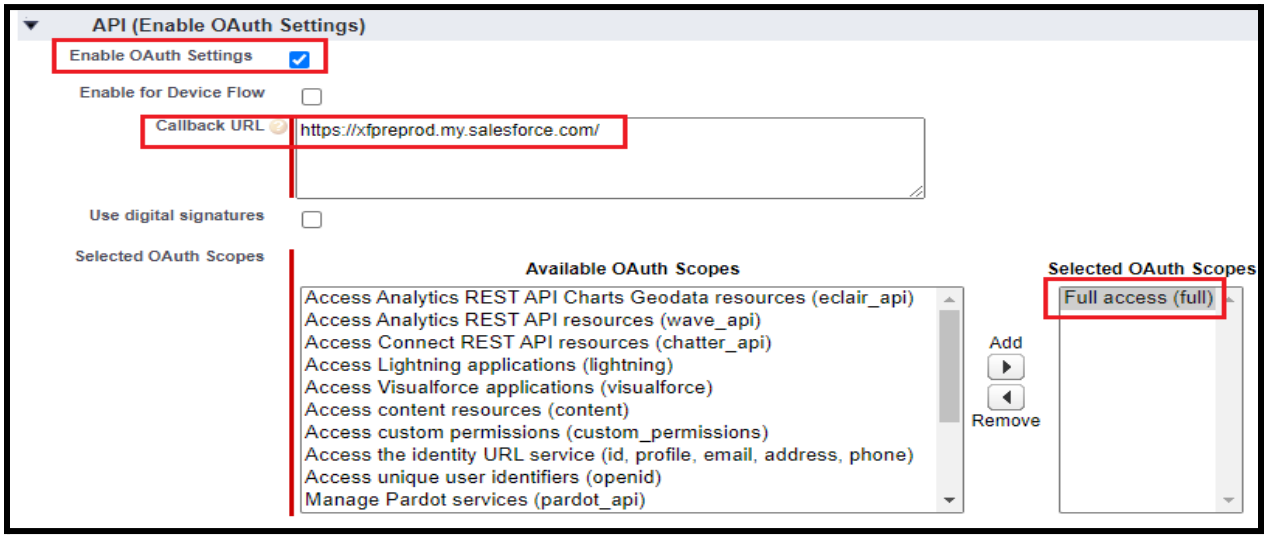
2. From the **Connected Apps** section, click on the **New** button.



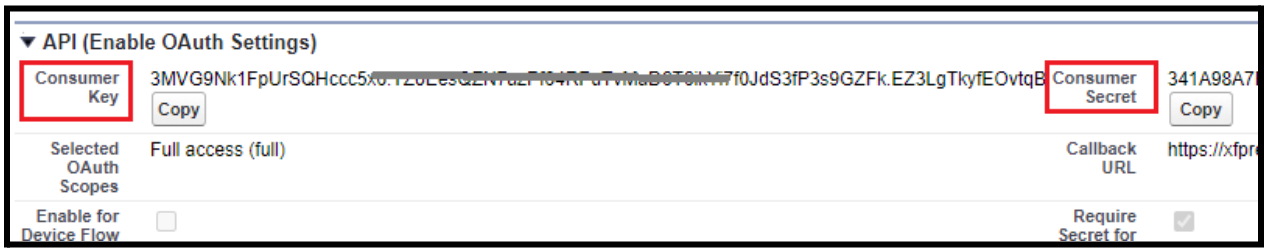
3. **Basic Information** section, Provide the mandatory fields - Connected App Name, API Name and Contact Email.



4. API (Enable OAuth Settings) section.
   a. **Enable OAuth Settings**: should be checked
   b. **Callback URL**: should be the org domain url.
   c. **Selected OAuth Scopes**: provide **Access Connect REST API resources (chatter_api).**
   d. Click on **Save** button and again click on **Continue**.

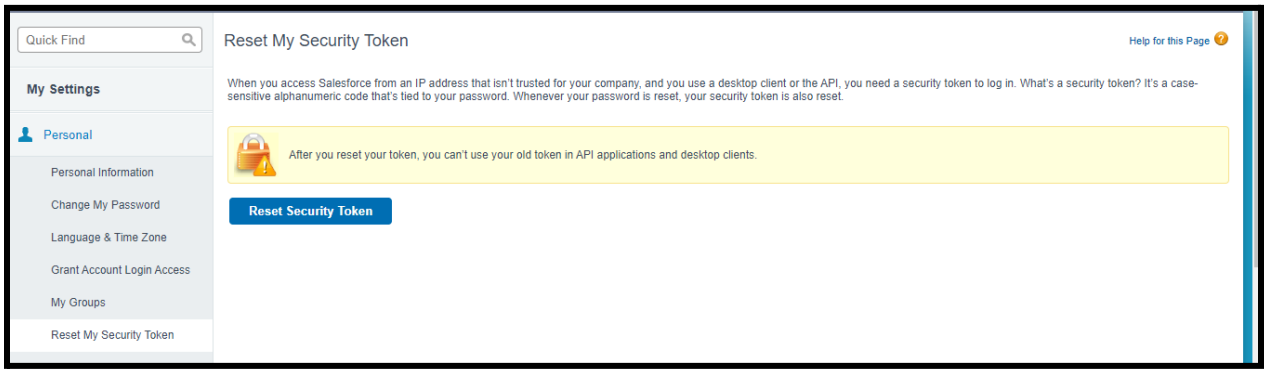


Connected App gets created with **Consumer Key** and **Consumer Secret** (Click on Click to reveal button)



How to get the **Security Token** (if you don't have the token already)

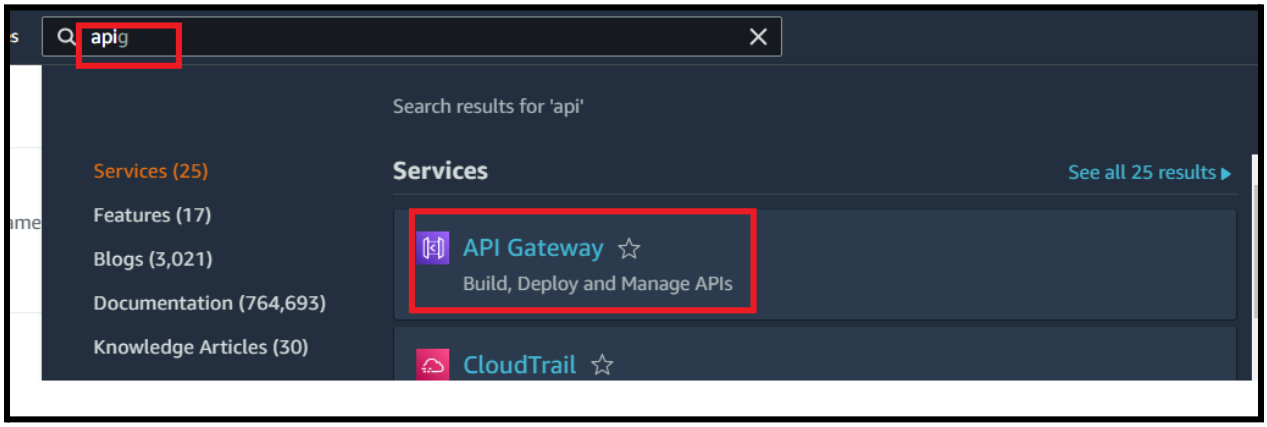
1. In the salesforce setup go to **My Setting**.
2. Click on **Personal.**
3. Click on **Reset My Security Token** and Click On **Reset Security Token** (see screenshot below).
4.



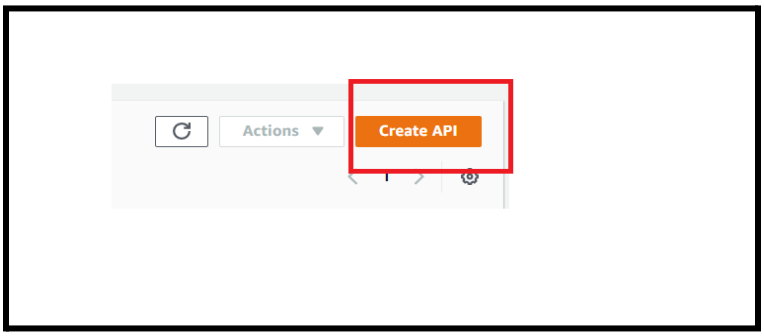
5. Please check your email and copy the token.
6. Add the environment variable value in the below format.
   For example: if your password is *ceptes123* and the security token is *etrtyqtewryey* then add the value as **ceptes123etrtyqtewryey.**
   Note: Here we are not supporting passwords with special characters.

Create API Gateway

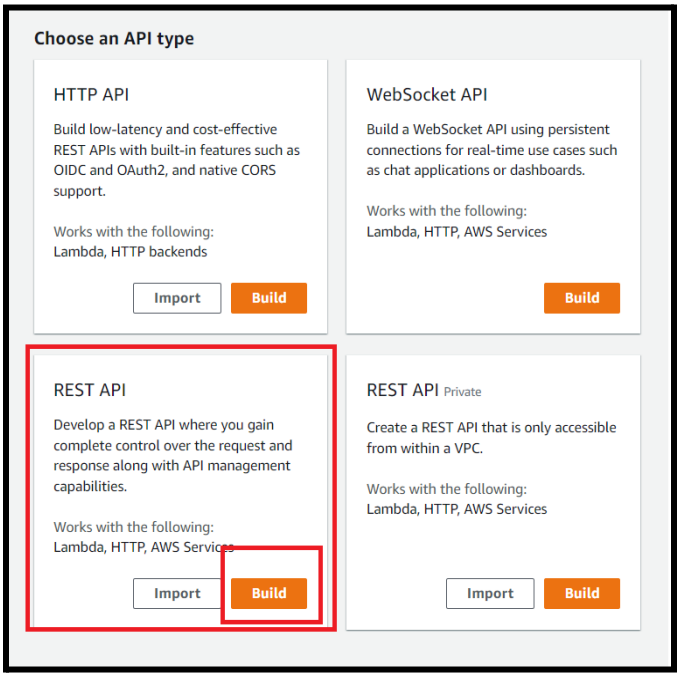
- Click on Services, open **API Gateway** service.



- Click on Create API

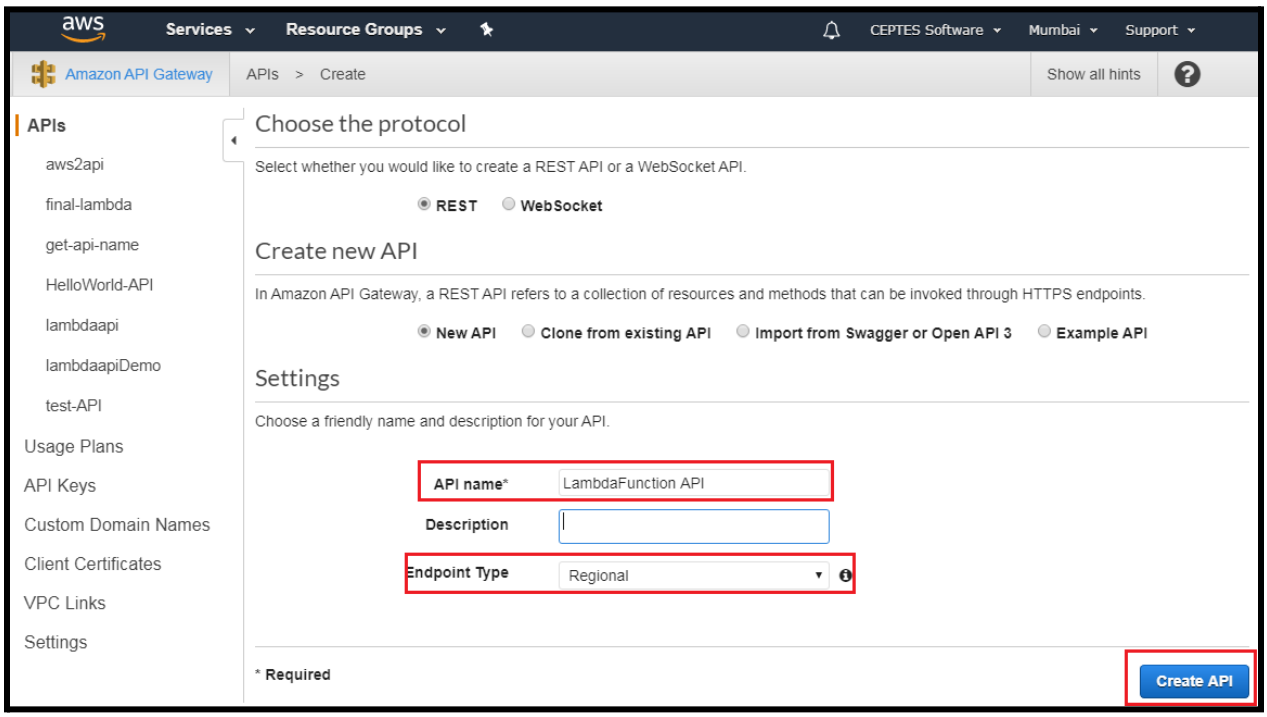


- Click on **Build**

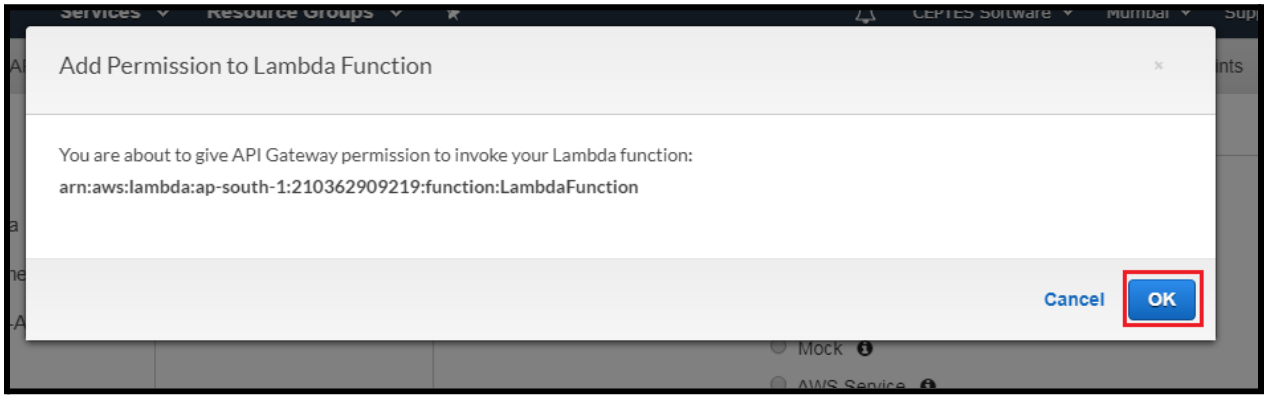


- From the Settings section, please enter the following:
    - Provide API name (any)
    - Description (if required)
    - Endpoint Type as Regional

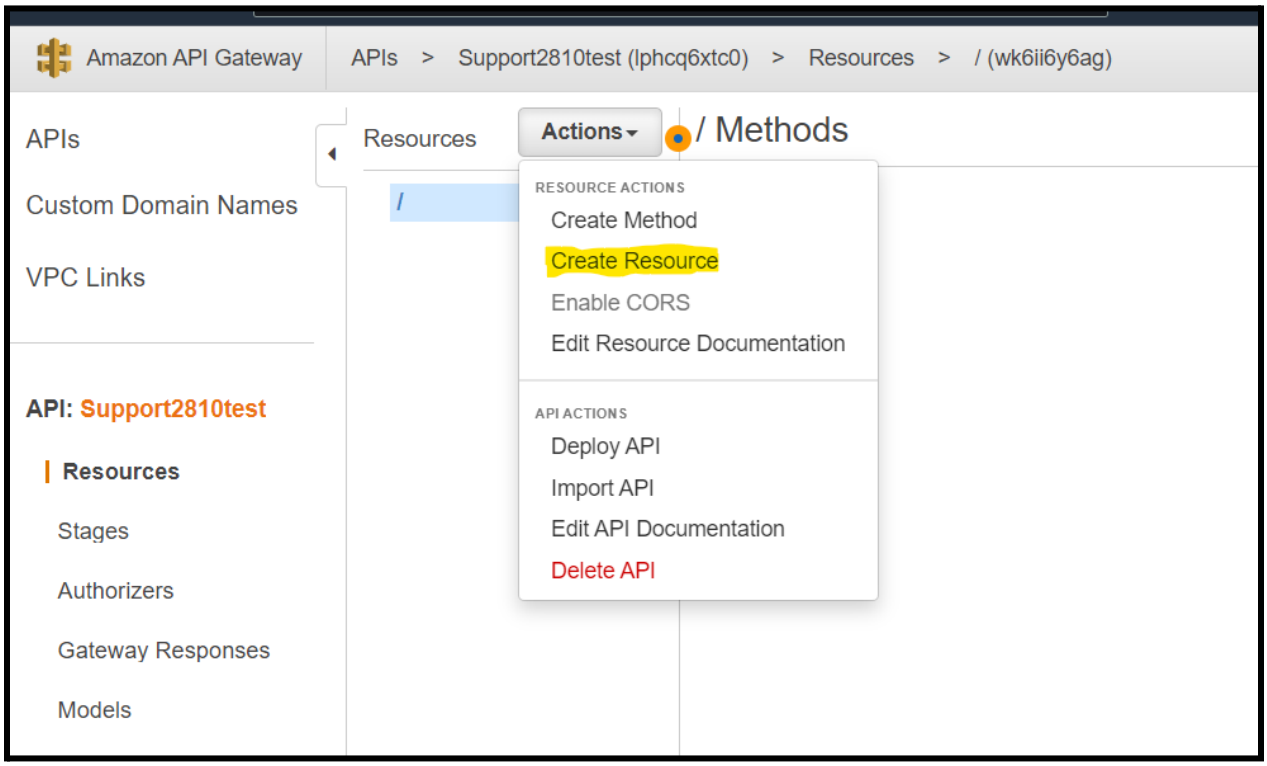
- Click on the **'Create API'** button.



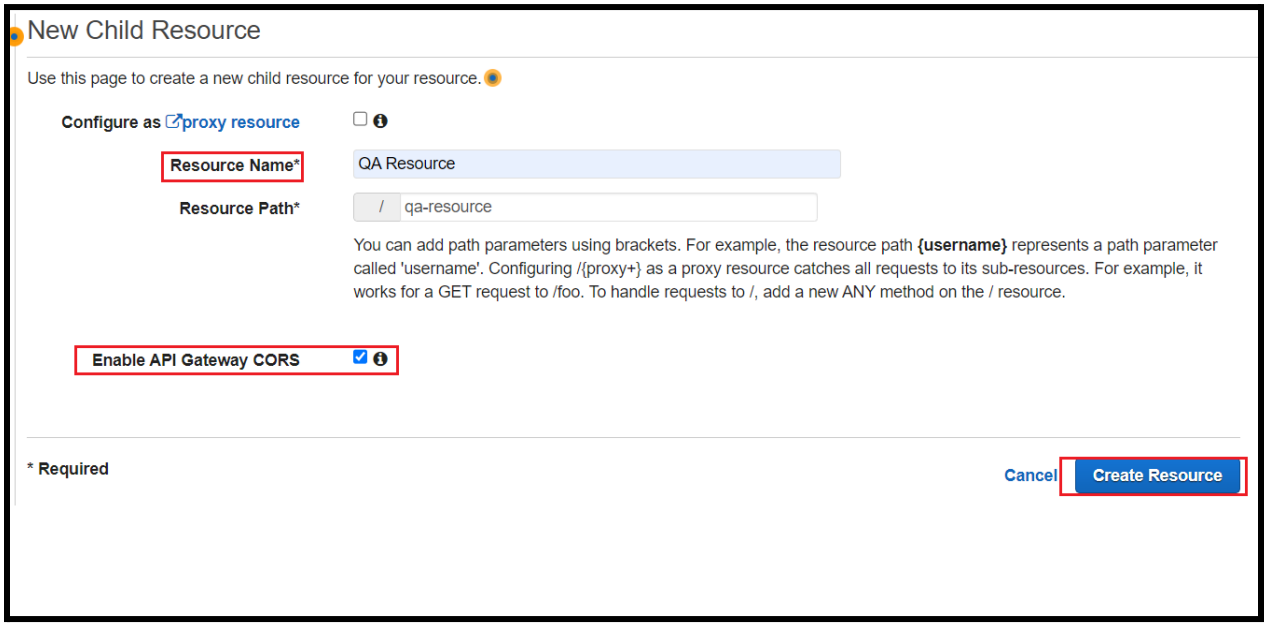
- Click on the **'Ok'** button.

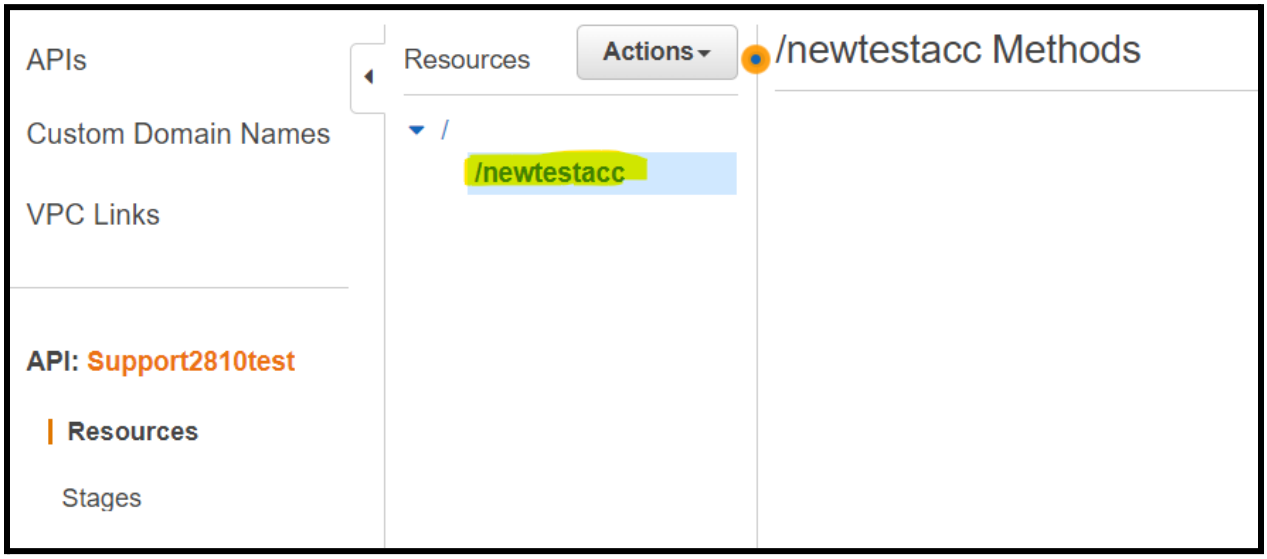


- The below screen appears. Now Click on the '**Action'** dropdown and select '**Create Resource'**

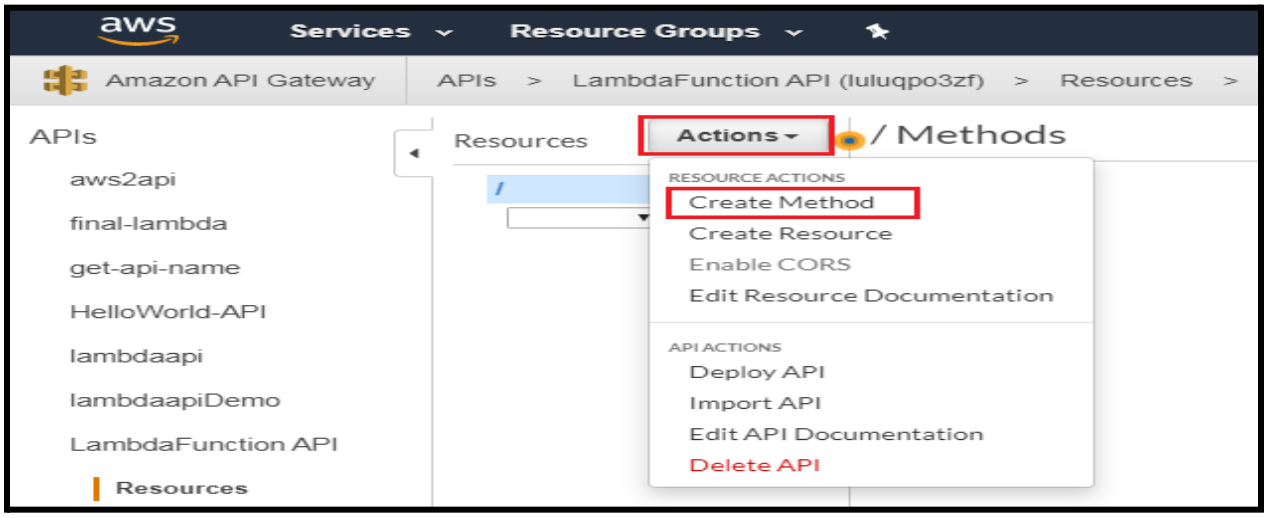


Now, Give the **Resource Name, Resource path** will be filled automatically.
Enable API Gateway CORS and Click on **Create Resources**.

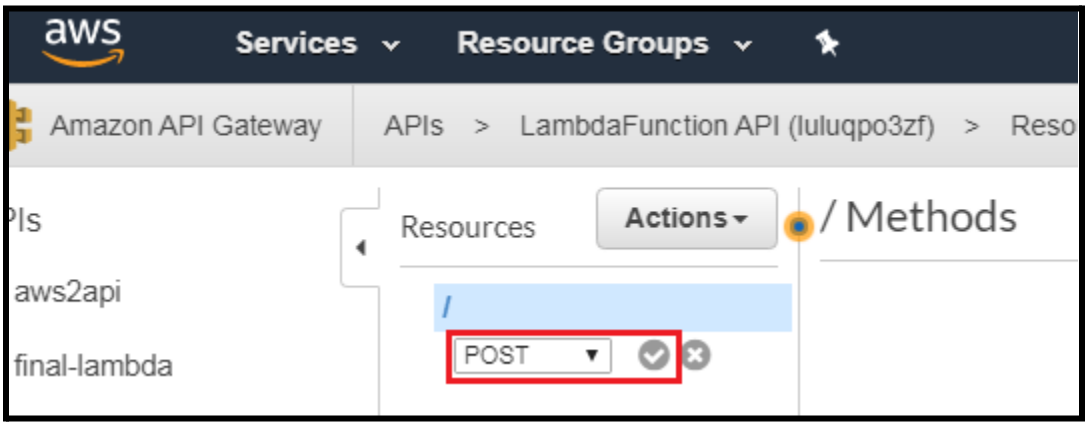

New Child Resource
Use this page to create a new child resource for your resource.
Configure as proxy resource
Resource Name*
QA Resource
Resource Path*
/
qa-resource
You can add path parameters using brackets. For example, the resource path {username} represents a path parameter called 'username'. Configuring /{proxy+} as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.
Enable API Gateway CORS
* Required
Cancel
Create Resource


APIs
Custom Domain Names
VPC Links
API: Support2810test
Resources
Stages
Resources
Actions
/
/newtestacc
/newtestacc Methods

- The below screen appears. Now Click on the '**Action'** dropdown and select '**Create Method'**

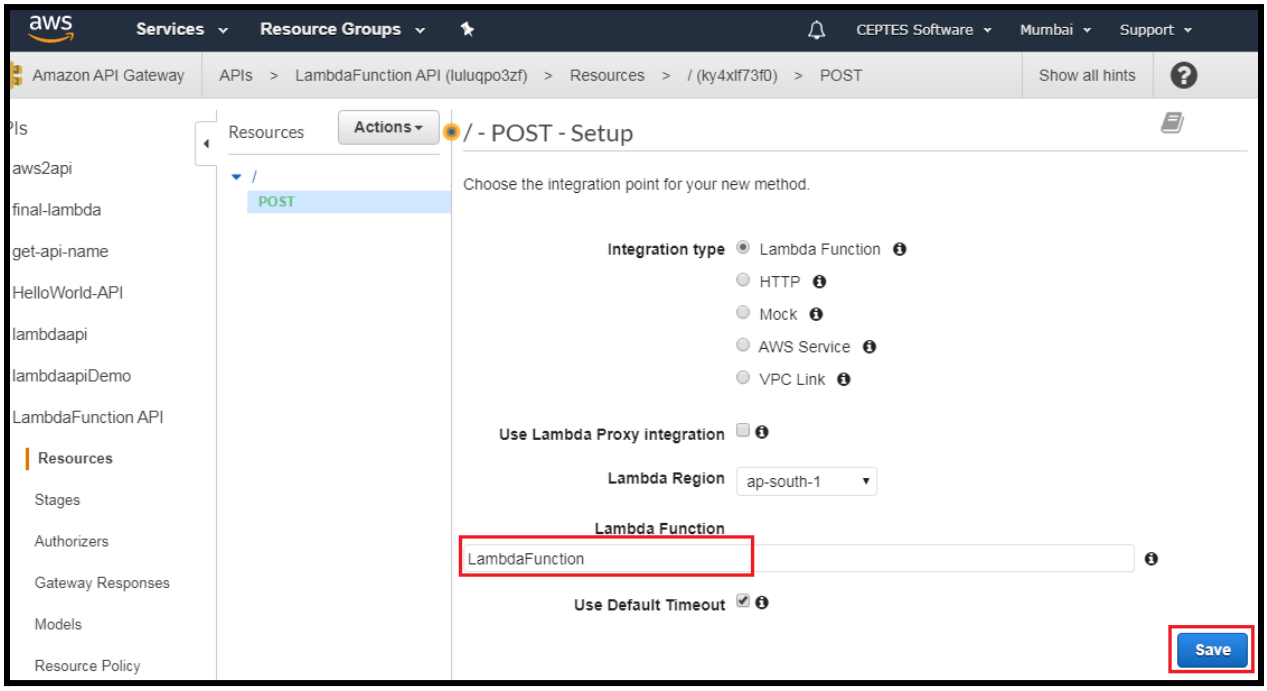


- Now select **'Post'** from the dropdown and click on **'Right Mark'**



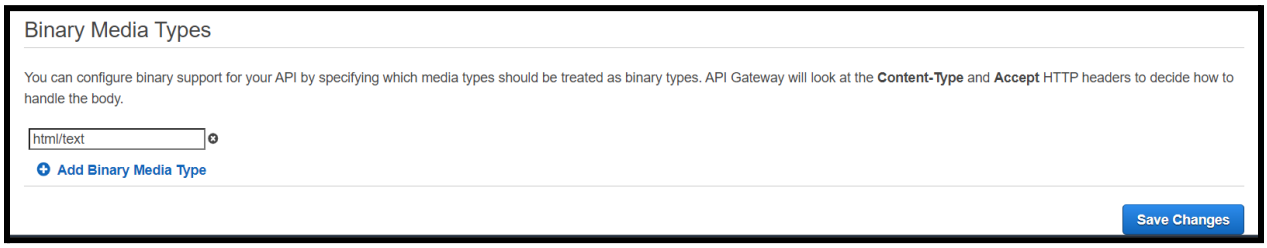
- The below screen appears. Now give the '**Lambda Function'** name (previously created Lambda Function)

And Click on **'Save'** button.

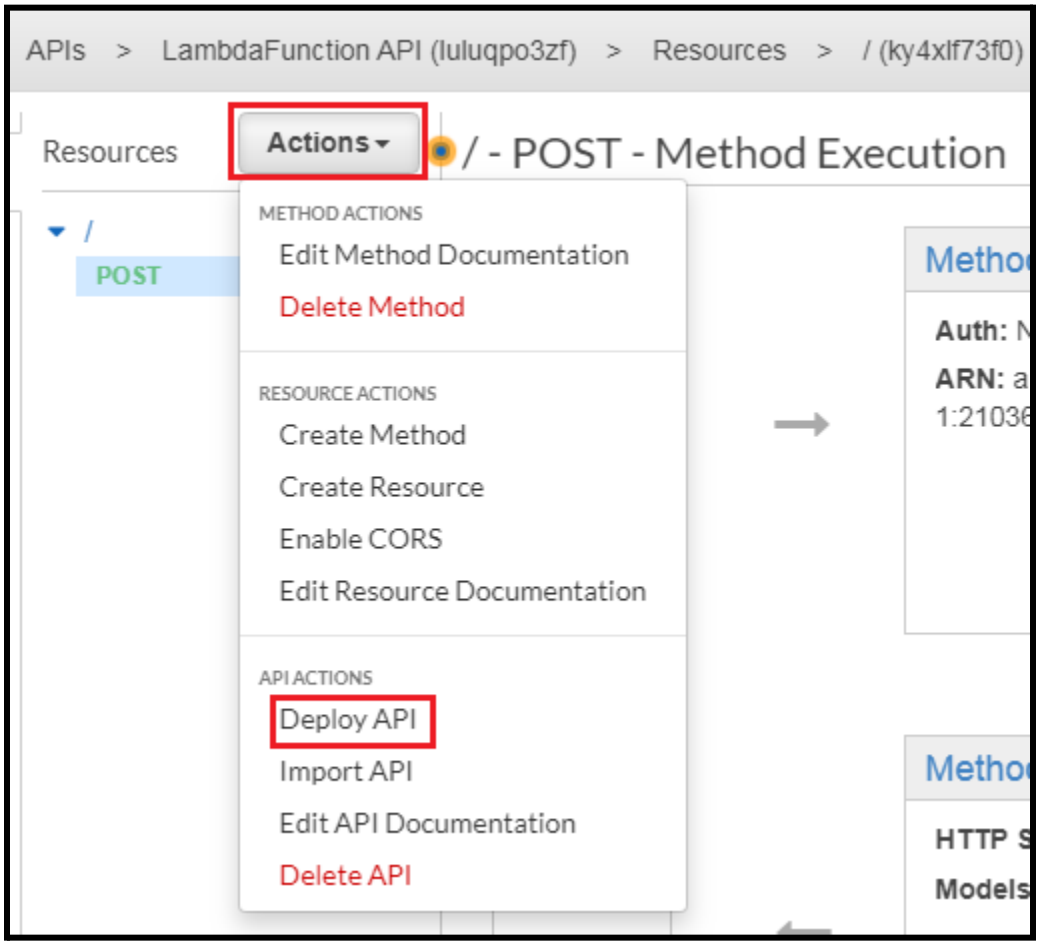


Encode the parameters from Gateway (optional)

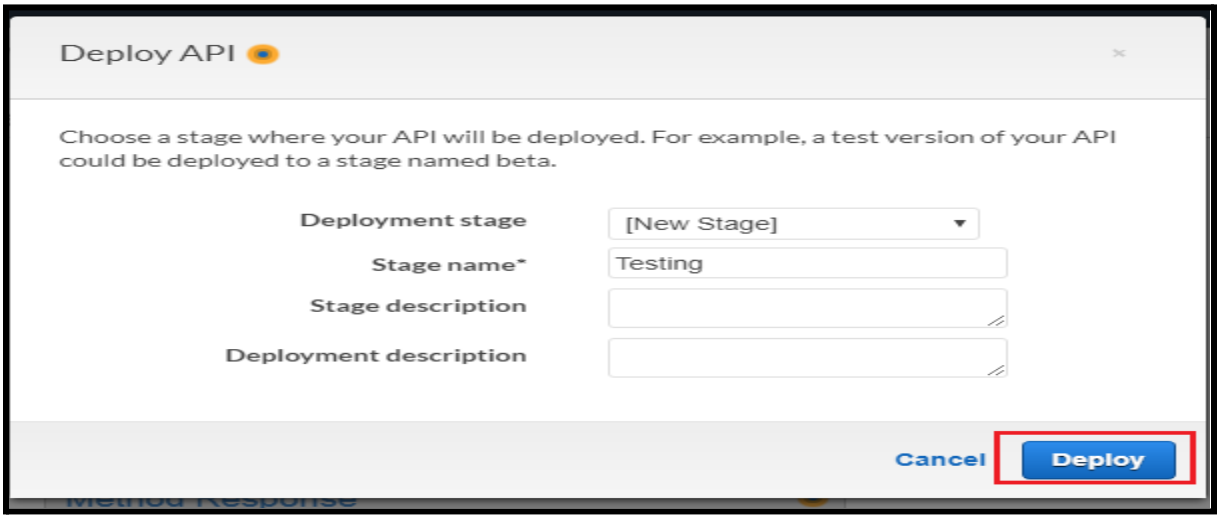
- Click on the **Settings** tab from the left panel.
- Scroll down to **Binary Media Types** section and add **html/text**
- Click on **Save Changes**



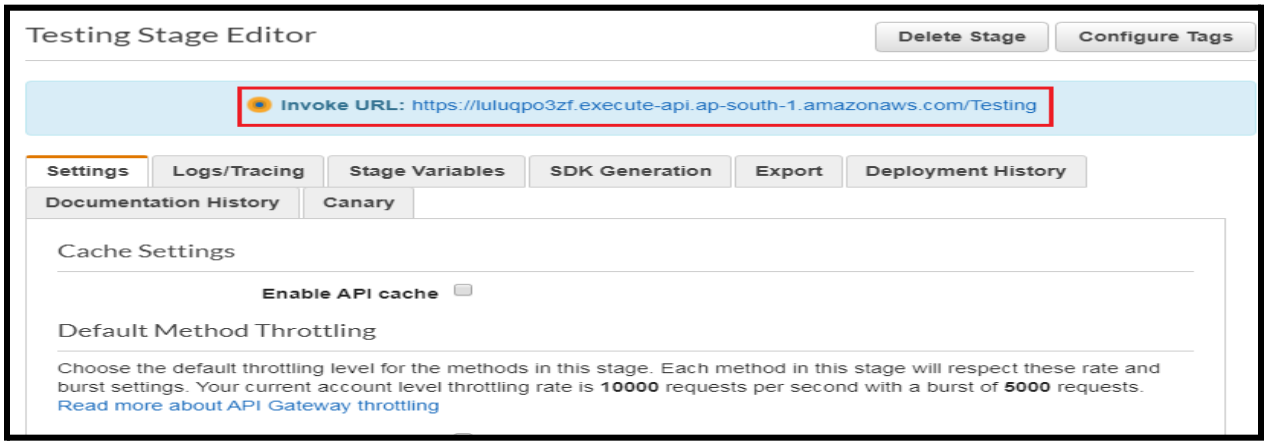
- The below screen appears. Click on the **'Actions'** dropdown and Click on **'Deploy API'**



- Provide the Deployment Stage as **'New Stage'** and Provide the **'Stage name'** and Click on the **'Deploy'** button



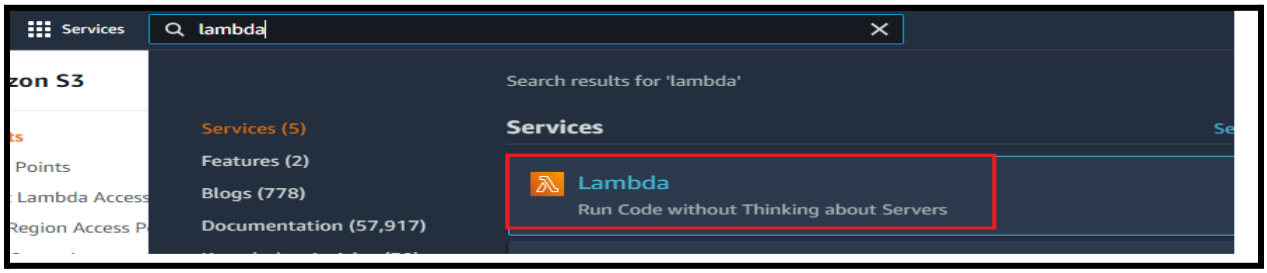
- The below page appears with the Invoke URL and keep this url (this is required to be added in the Salesforce's custom settings)



Add Authorizer

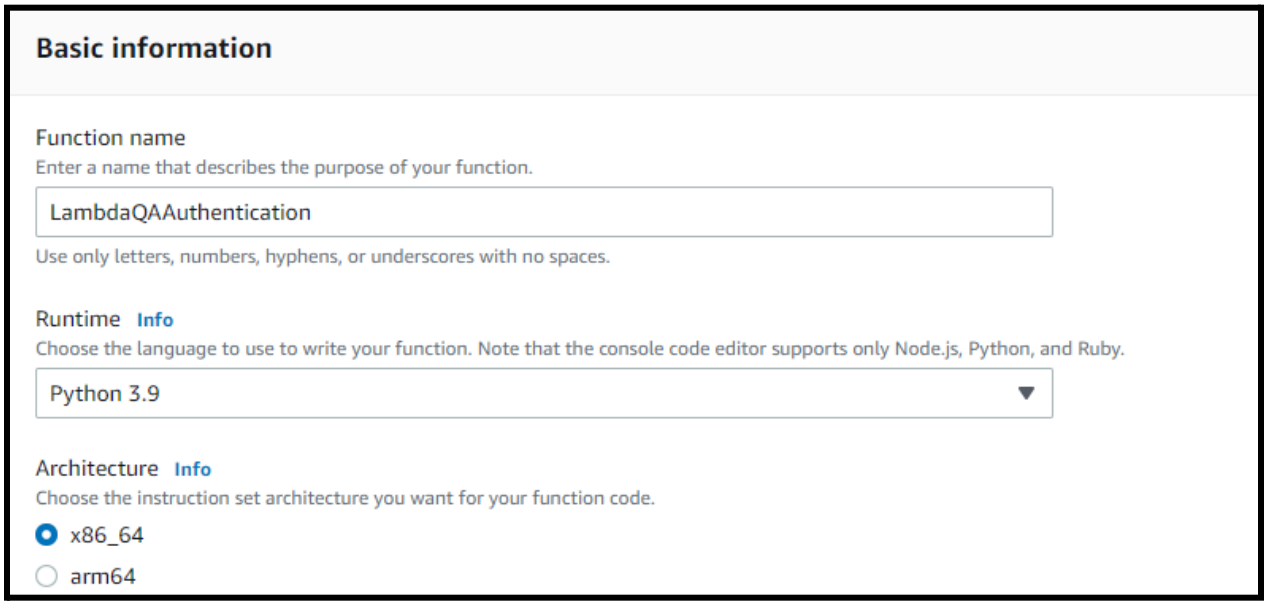
Create Authorizer Lambda Function

- Go to AWS Console. Click on Services, Search for Lambda, and open Lambda service.



- Click on the **"Create Function"** button

- **Provide the Basic Informations**
  - **Function Name** - Provide any name
  - **Runtime** - Select **Python 3.9** from the dropdown And click on the **Create Function** button.

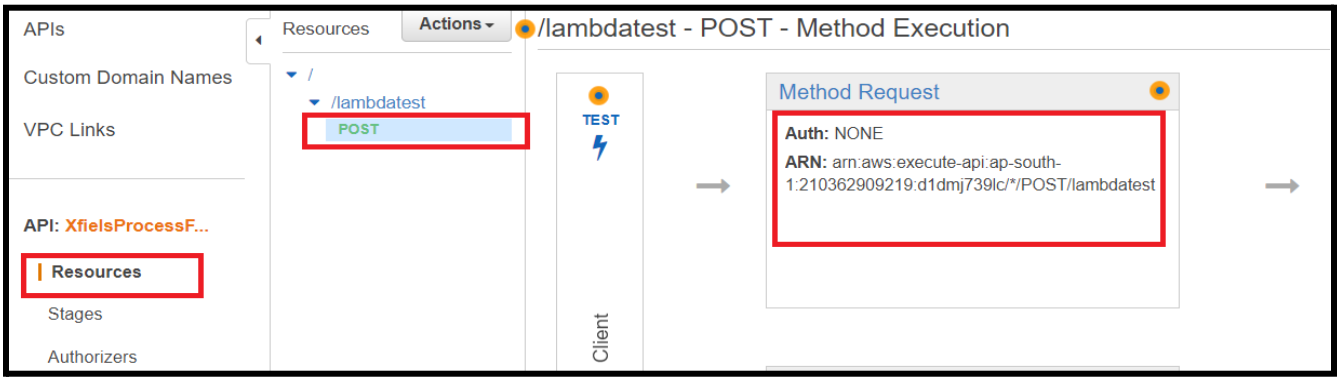


- Add the below python code snippet and make a note of the `authorizationToken` (in the below code it is `'abc123'`), this **authorization token** should be added in the Salesforce.

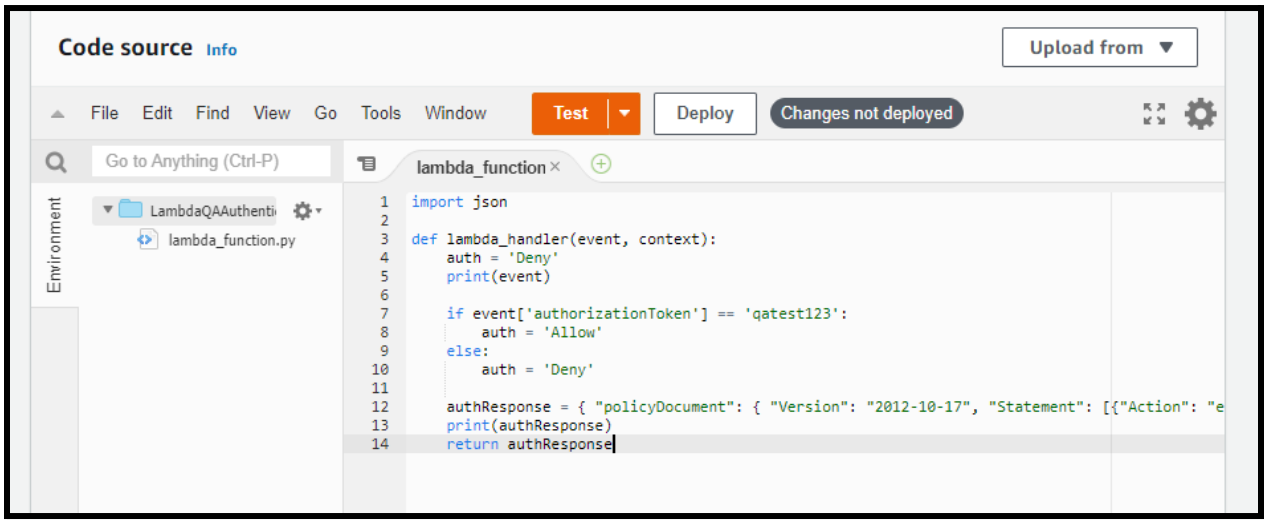
```
import json

def lambda_handler(event, context):
    auth = 'Deny'

    if event['authorizationToken'] == abc123:
        auth = 'Allow'
    else:
        auth = 'Deny'

    authResponse = { "policyDocument": { "Version": "2012-10-17", "Statement":
[{"Action": "execute-api:Invoke", "Resource":
["arn:aws:execute-api:ap-south-1:210362909219:y0p9k245f8/*/*"], "Effect":
auth}] }}
    return authResponse
```

<u>Update above python code with your lambda function's ARN</u>
Get the ARN from the earlier created API Gateway as shown in the below screenshot.

Example: "arn:aws:execute-api:ap-south-1:210362909219:gz8ek9a24c"

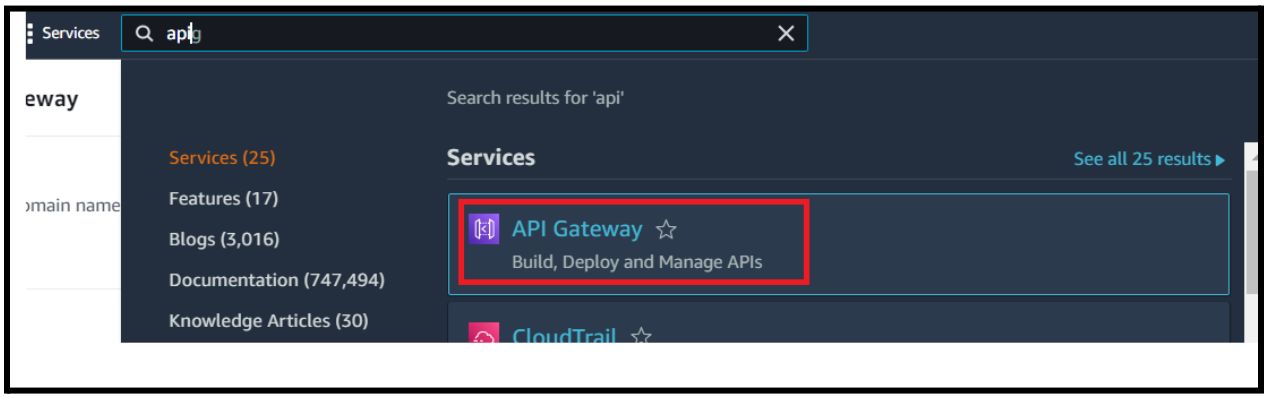


- Click On the **Deploy** button

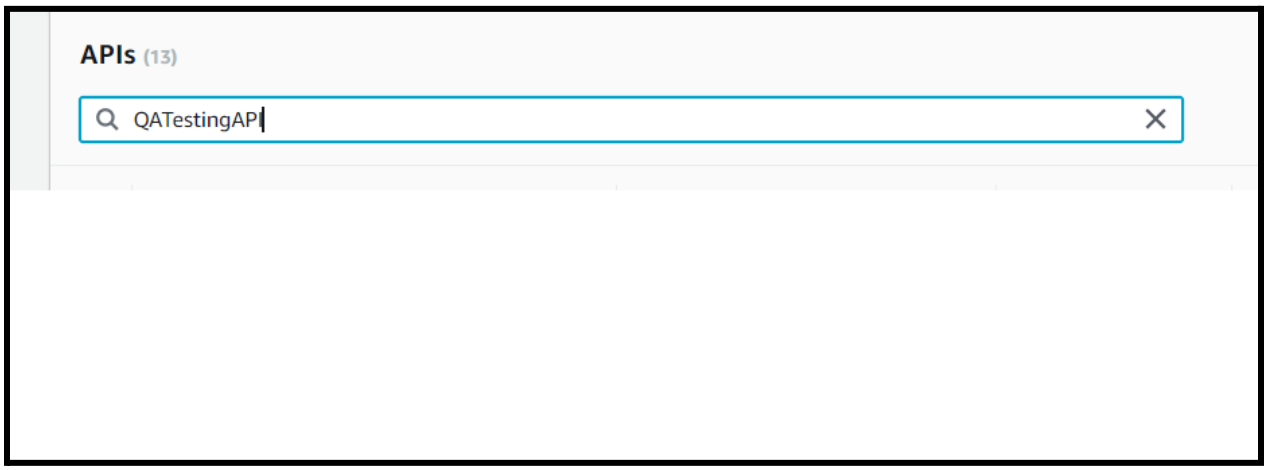


Create Authorizer for Gateway API

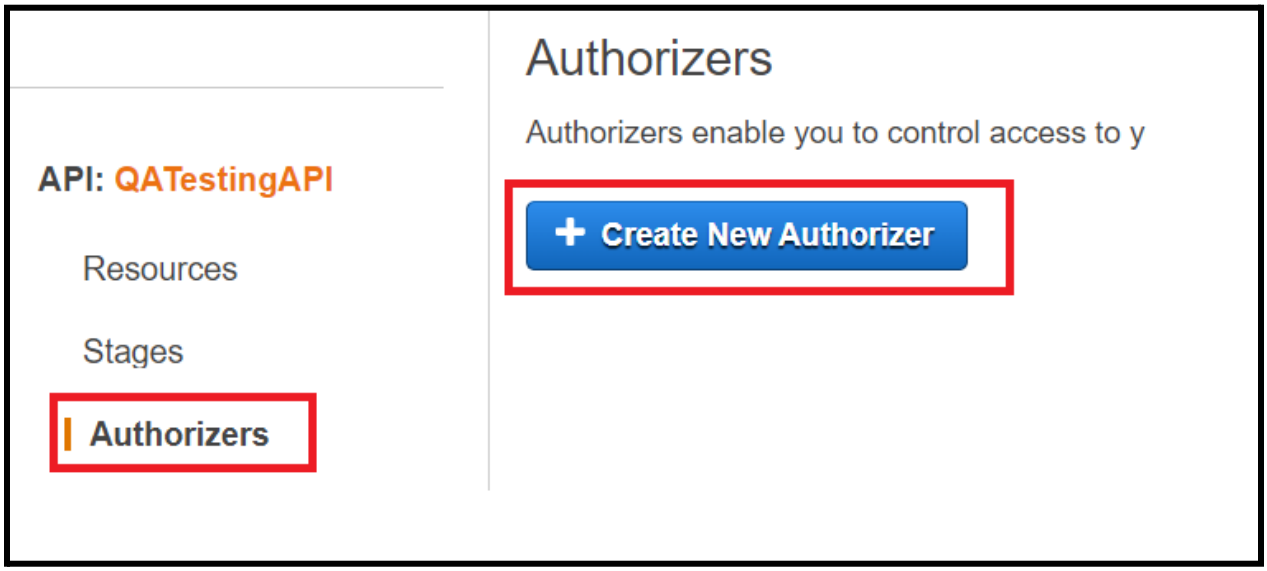
- Search for **API Gateway** and open it.



- Open the Gateway API which was created earlier.

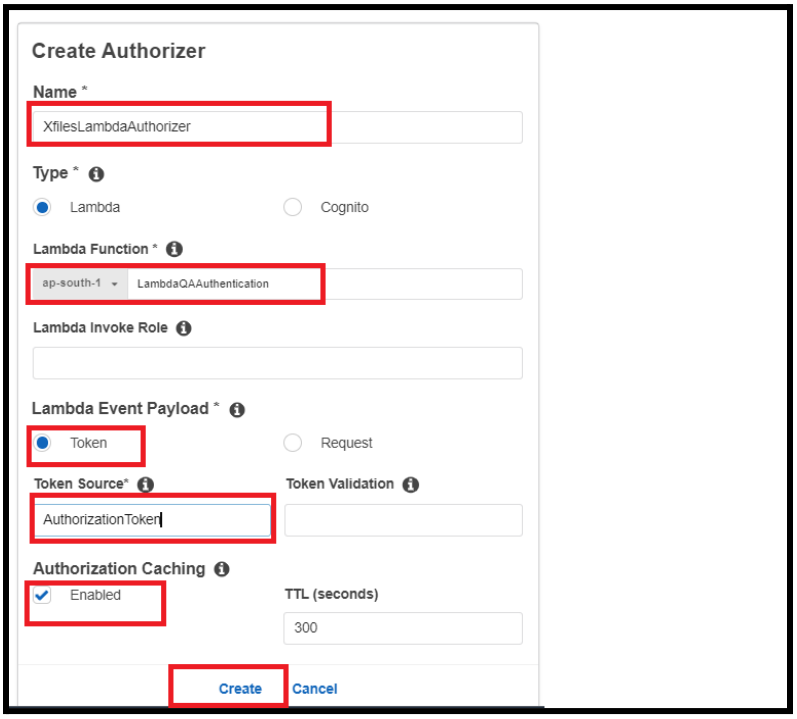


- Click on **Authorizers** and select **Create New Authorization**

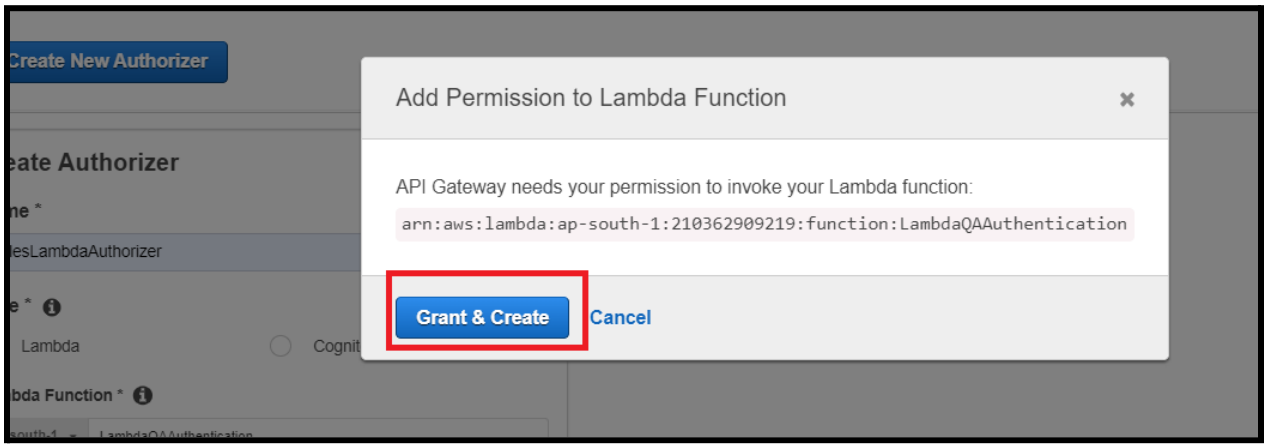


- Provide the Information:
  - **Function Name** - provide any name
  - **Type**: Lambda
  - **Lambda Function Name**: Select Authorizer Lambda Function (second lambda function having authorization python code).
  - **Token Source:** AuthorizationToken

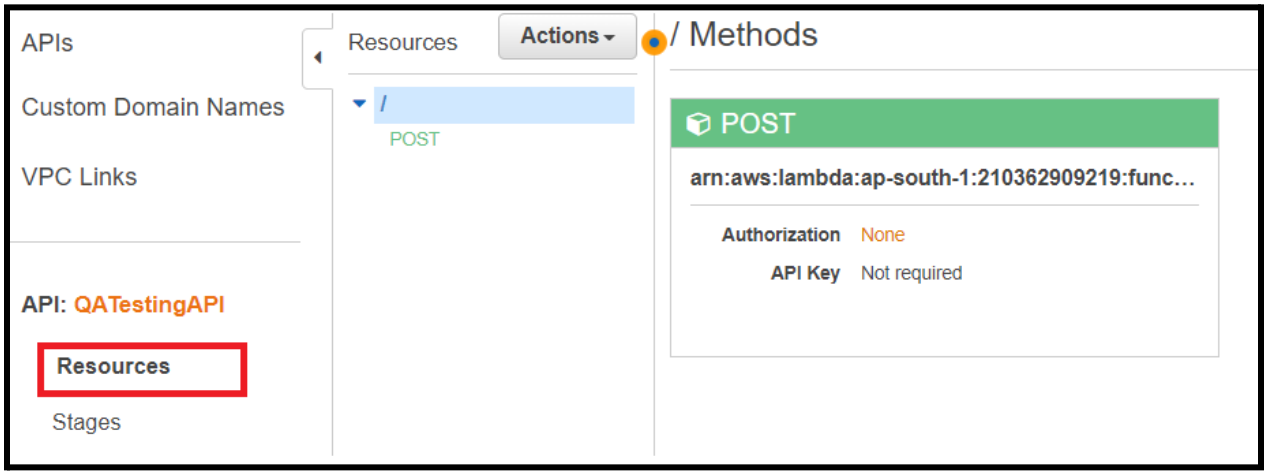
- Click on the **Create** link.



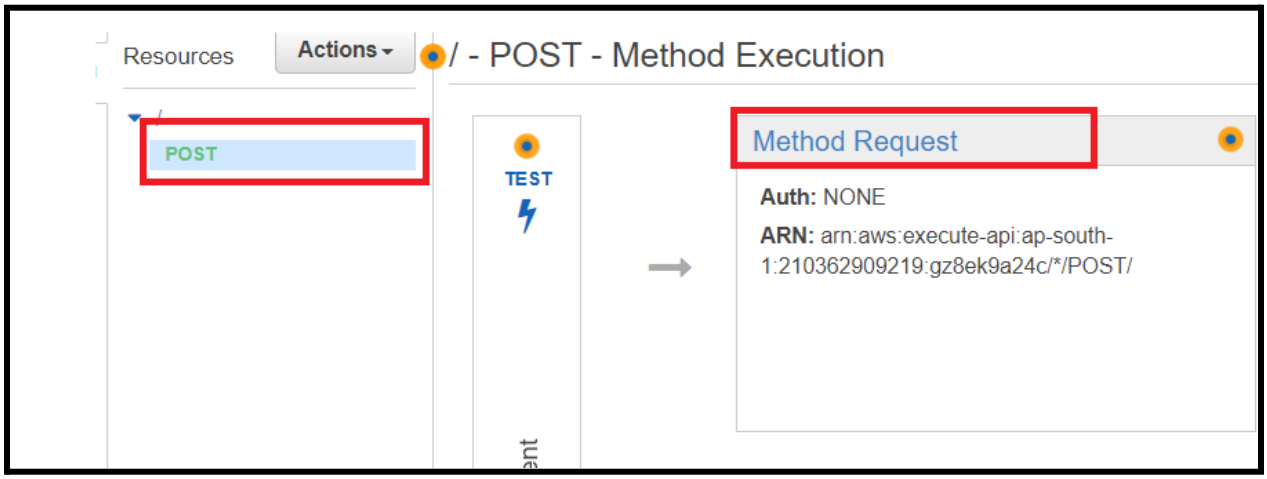
- Click on **Grant & Create.**



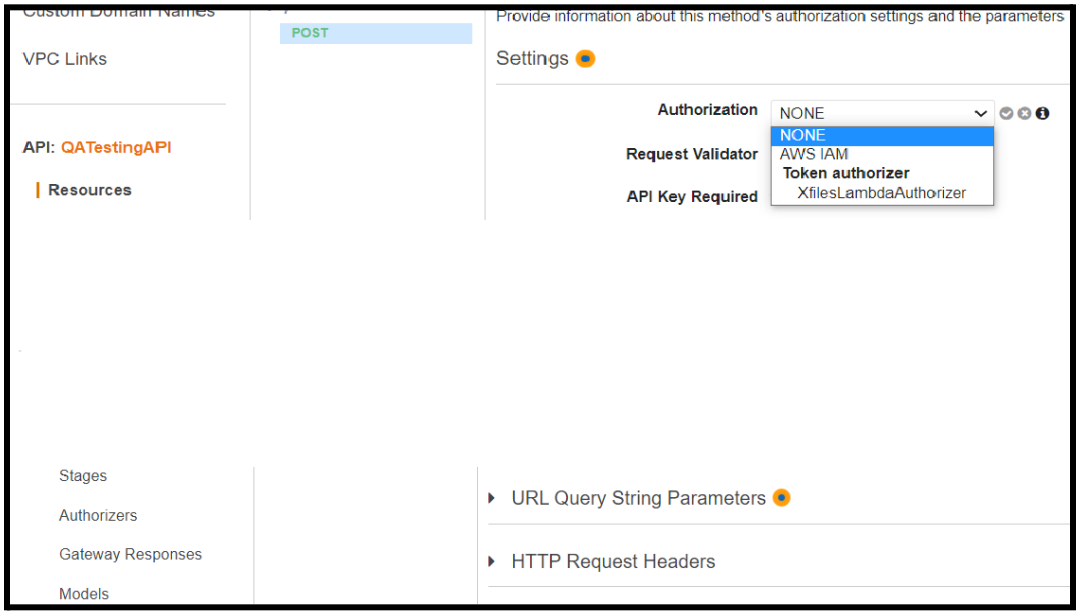
- Then click on **Resources.**



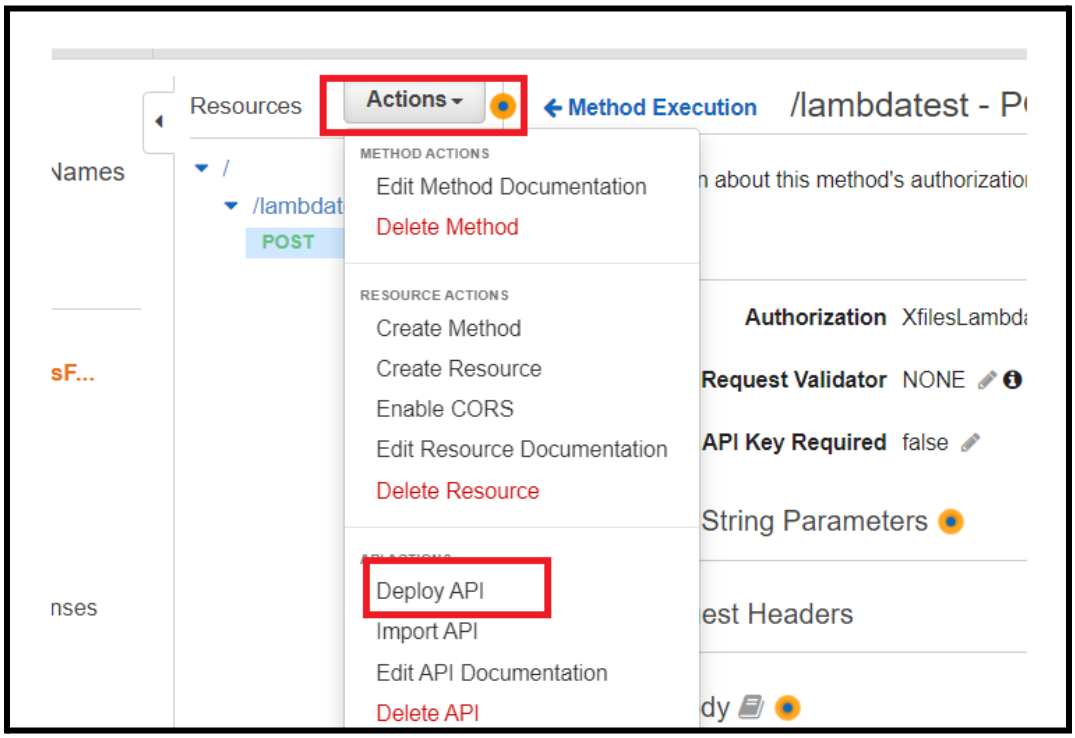
- Click on Post and then click on **Method Request.**



- In the **Authorization** field, select the authorizer Lambda function and click on the right mark.

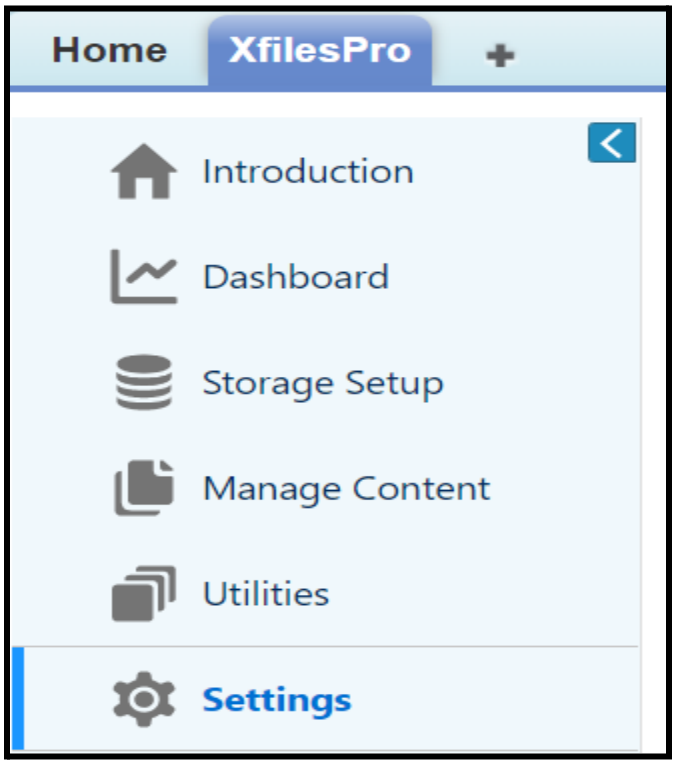


- Then click on the Action button select **Deploy API**

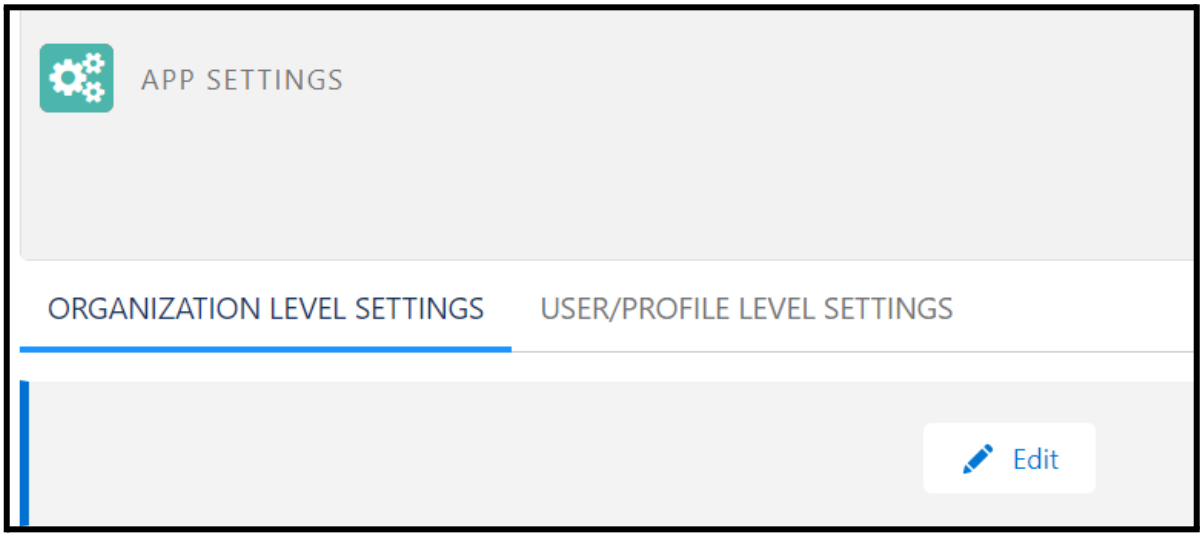


Add the Gateway API invoke URL and Authorization Token

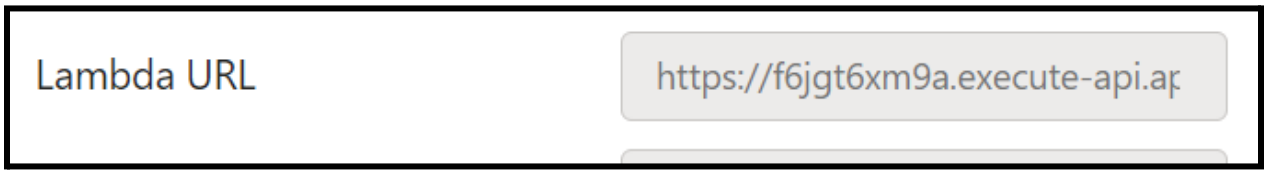
1. Go to **XfilesPro Setting**



2. Click On **Edit**



3. In the **Lambda URL** field, paste the Gateway API invoke URL (copied from AWS gateway)



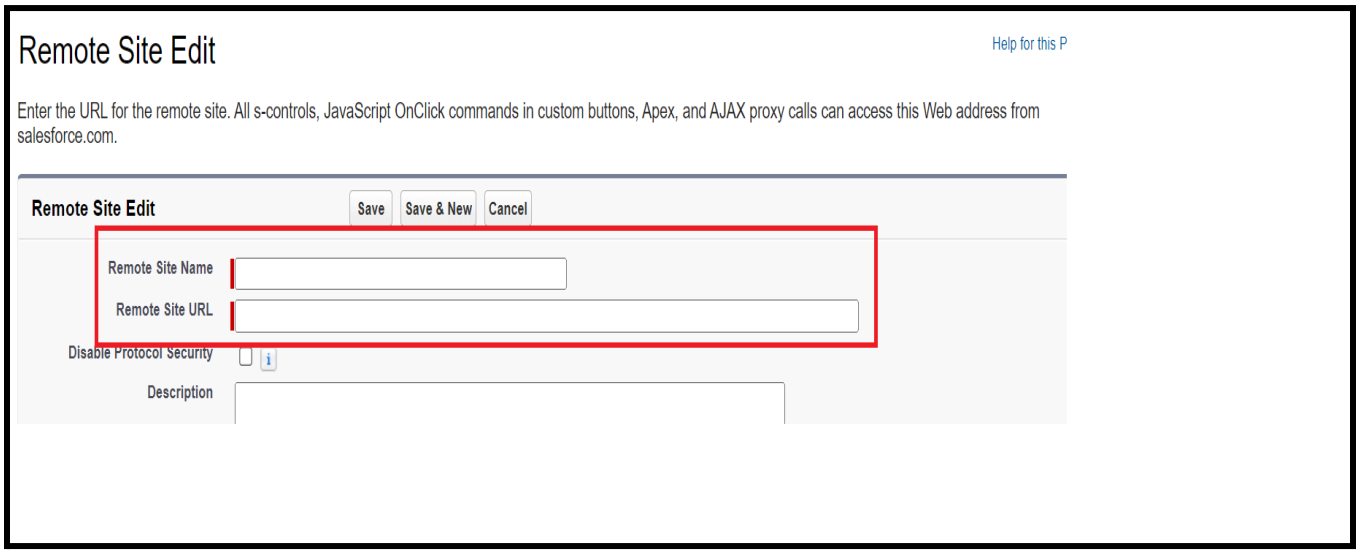
4. In the **AWS Authorization Token** field, enter the Authorization Token (Which was given in python code)

Note: For security reasons, the Authorization Token will be saved in the encrypted form.

5. Click On **Save**

Remote Site Settings

1. Go to Salesforce setup.
2. Search for Remote Site and open **Remote Site Settings**.
3. Click on **New Remote Site**.
4. Provide the Information:
    - **Remote Site Name**: provide any name.
    - **Remote Site URL**: Add Gateway API invoke URL (from AWS API Gateway).



CSP Trusted Site

5. Go to Salesforce setup.
6. Search for CSP and open **CSP Trusted Site**.
7. Click on **New Trusted Site**.
8. Provide the Information:
    - **Trusted Site Name**: provide any name.'
    - **Trusted Site URL**: Add Gateway API invoke URL (from aws API Gateway).