

# Instructions for Setting up AWS Backend for Percept

## Part 1: Create stack resources

### Step 1.1 – Log In:

You will first need to create the stack of resources for the backend. Go to <https://aws.amazon.com/> and sign in. If you do not have an account, you will need to create one.

### Step 1.2 – Visit S3:

Once you log in, your web page should look like this:

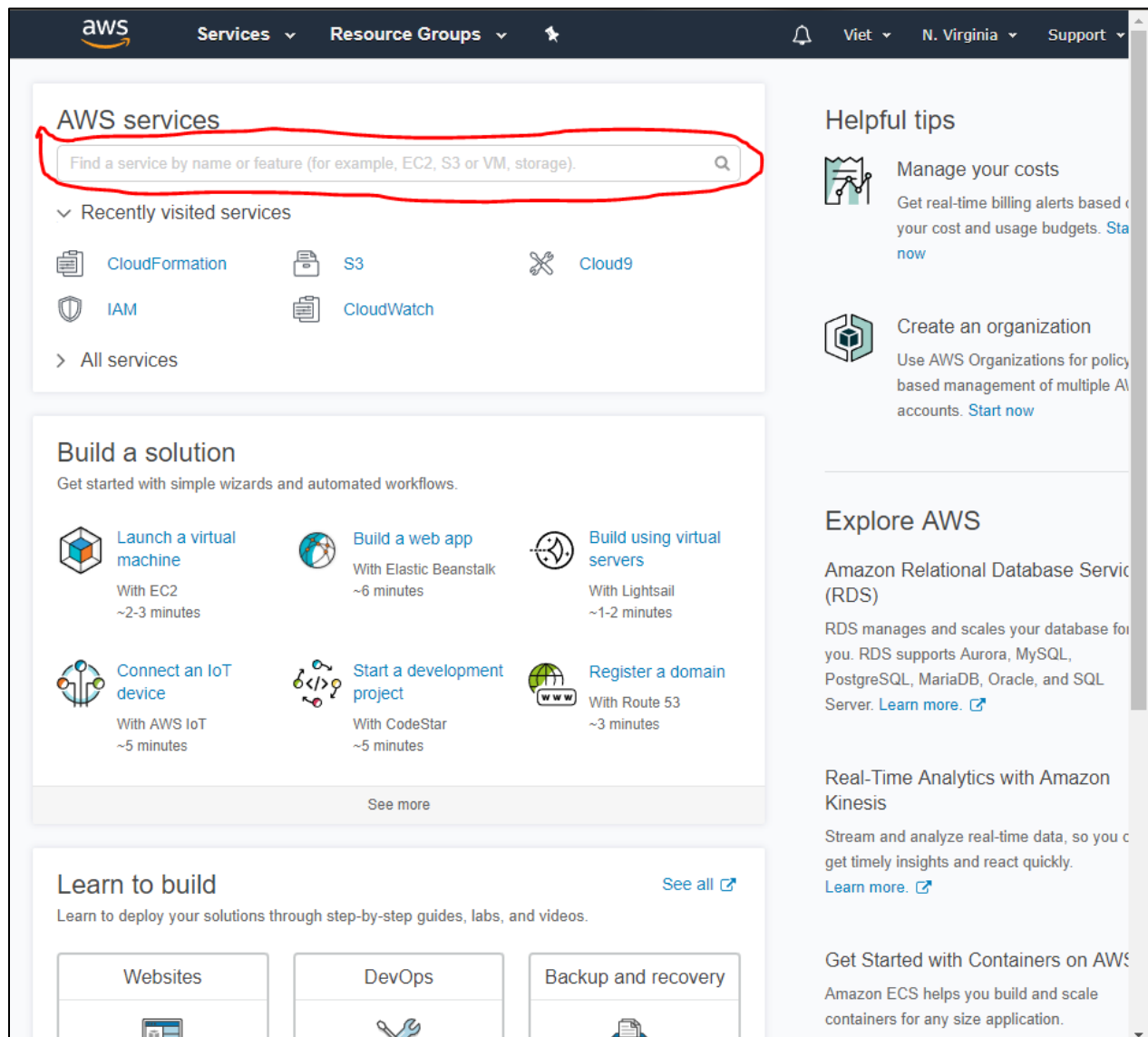
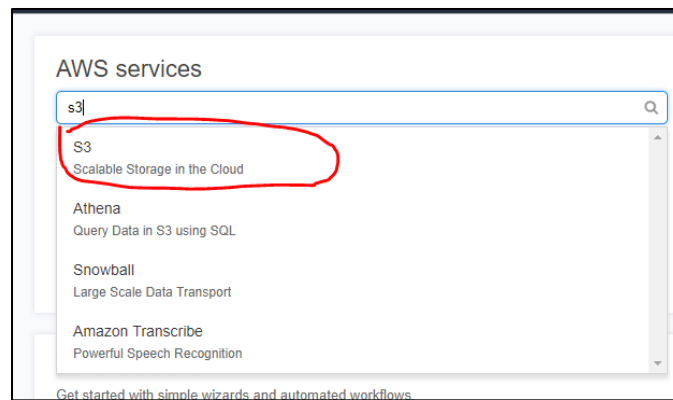


Figure 1 AWS Landing Page

Type in S3 into the search bar circled red.



*Figure 2 Search Bar*

Select **S3**.

### Step 1.3 – Create a Bucket:

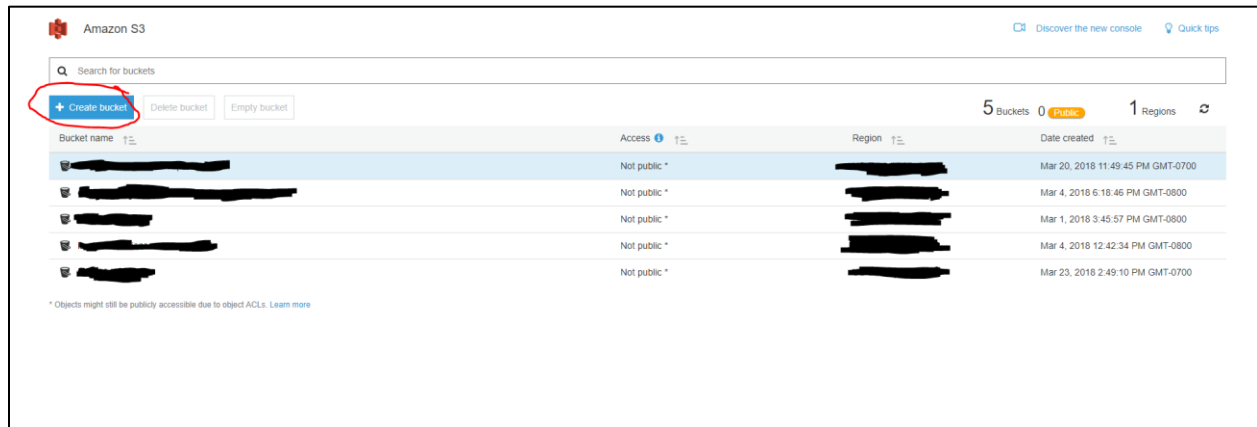


Figure 3 Amazon S3 Page. Personal details removed.

Click the button labeled “Create Bucket”, then choose a unique name for that bucket. **WRITE THE NAME DOWN. YOU WILL NEED IT LATER.** Keep pressing “Next” until you are at the summary, which should look like this:

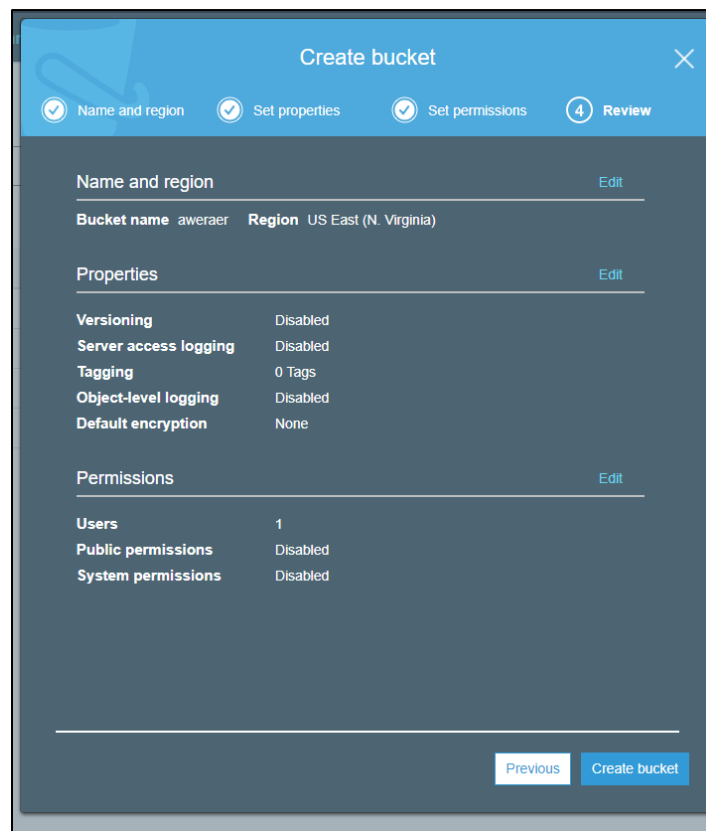


Figure 4 S3 Bucket Summary

Click “Create Bucket”.

### Step 1.4 – Upload Code into Bucket:

Click on the bucket that you created. In my case, the bucket's name was “test1234notunique”.

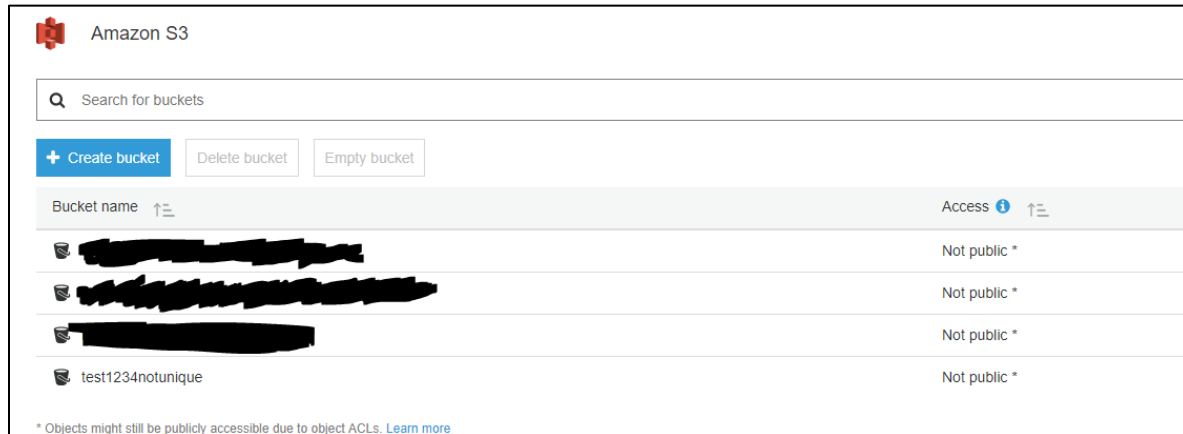


Figure 5

The screen should look like this:

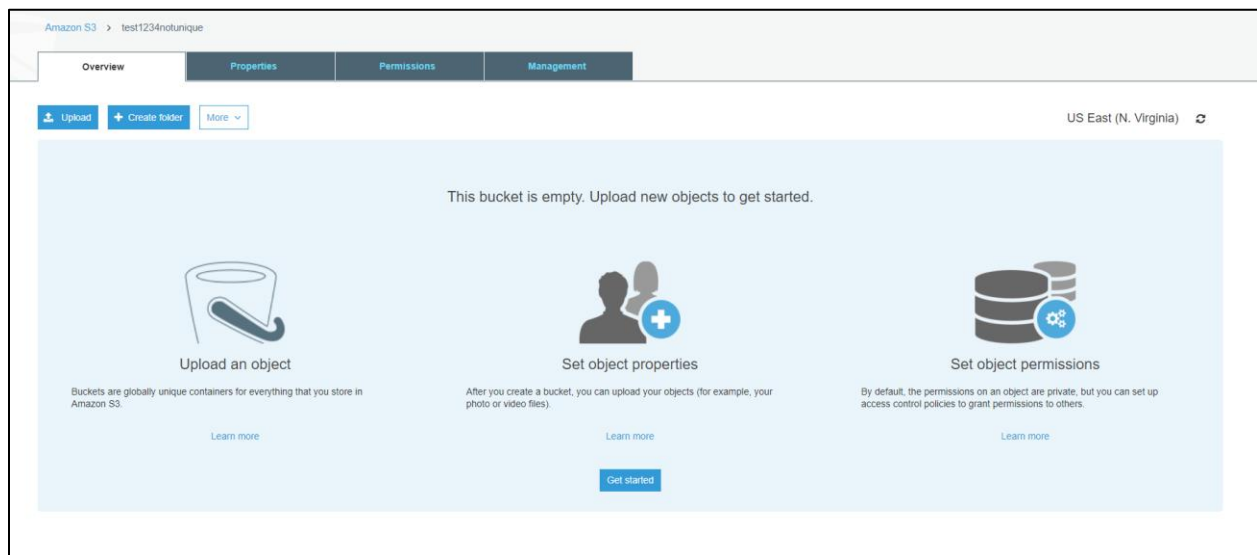


Figure 6

Drag the three folders in the Github repo labeled **graphcreator.zip**, **sd2db.zip**, and **sd2es.zip** into the bucket to upload them. Alternatively, you can upload them using the “Upload” button.

## Step 1.5 – Generate stack using AWS Cloudformation

Go back to the landing page of [aws.amazon.com](https://aws.amazon.com). It should be the same page as depicted in [Step 1.2, Figure 1](#). Use the search bar to find AWS Cloudformation.

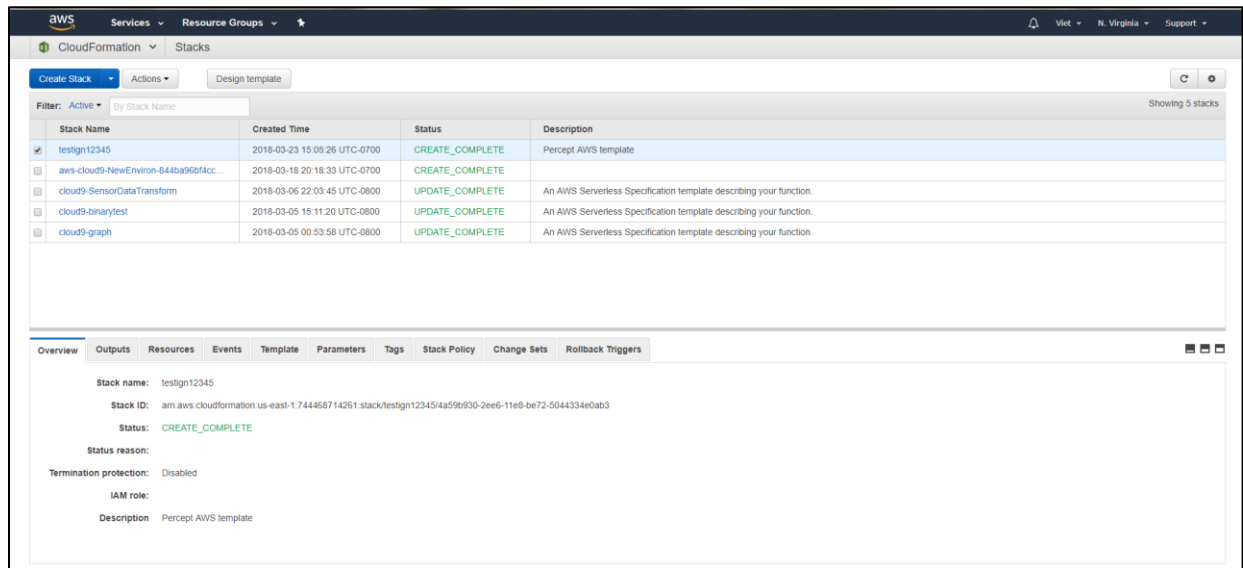


Figure 7 Cloudformation Landing Page

Click “Create Stack”.

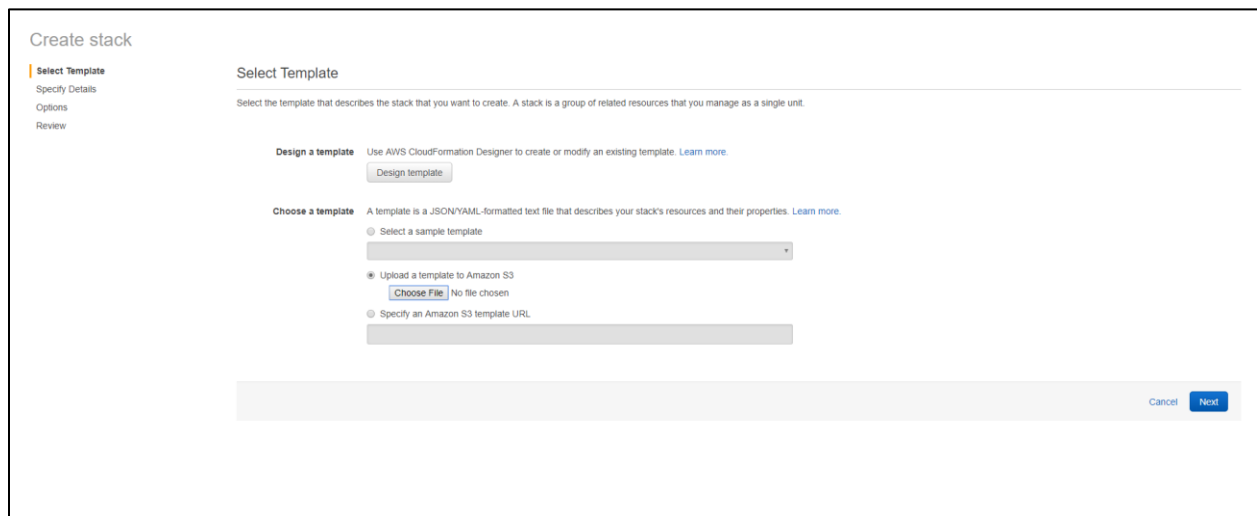


Figure 8 1st Page of Stack Creation

Choose “Upload a template to Amazon S3”. Upload **cloudformation.yaml** from the Github repo. Press “Next”.

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

Parameters

S3BucketName

Cancel Previous Next

Figure 9 Stack Parameters

Choose a stack name. It must follow the regex pattern `[a-z][a-z0-9\-\-]+`. For “S3BucketName”, input the bucket name you created in [Step 1.3 – Create a Bucket](#). Press “Next”.

Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. [Learn more.](#)

Key (127 characters maximum)	Value (255 characters maximum)
1 <input type="text"/>	<input type="text"/>

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more.](#)

IAM Role
Choose a role (optional)
Enter role arn

Rollback Triggers

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more](#)

Monitoring Time
0-180 Minutes
Minimum value of 0. Maximum value of 180.

Type	ARN (Amazon Resource Name)
1 AWS: CloudWatch: Alarm	<input type="text"/>

Available triggers remaining: 5

Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

Cancel Previous Next

Figure 10

Press “Next”.

**Details**

Stack name: asdf34123  
 S3BucketName asdf123454

**Options**

Tags  
 No tags provided

Rollback Triggers  
 No monitoring time provided  
 No rollback triggers provided

**Advanced**

Notification  
 Termination Protection Disabled  
 Timeout none  
 Rollback on failure Yes

**Capabilities**

**i** The following resource(s) require capabilities: [AWS::IAM::Role]  
 This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions.  
[Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources.

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

[Cancel](#) [Previous](#) [Create](#)

Figure 11

Check the box labeled “I acknowledge that AWS Cloudformation might create IAM resources. Press “Create”.

Filter: Active ▾ By Stack Name				
	Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/>	asdf34123	2018-03-23 16:09:16 UTC-0700	CREATE_IN_PROGRE...	Percept AWS template

Figure 12

Wait until “CREATE\_IN\_PROGRESS” switches to “CREATE\_COMPLETE”. You may need to refresh the page.



Filter: Active By Stack Name				
Stack Name	Created Time	Status	Description	
testign12345	2018-03-23 15:05:26 UTC-0700	UPDATE_IN_PROGRE...	Percept AWS template	
aws-cloud9-NewEnviron-844ba96bf4cc...	2018-03-18 20:18:33 UTC-0700	CREATE_COMPLETE		
cloud9-SensorDataTransform	2018-03-06 22:03:45 UTC-0800	UPDATE_COMPLETE	An AWS Serverless Specification template describing your function.	
cloud9-binarytest	2018-03-05 15:11:20 UTC-0800	UPDATE_COMPLETE	An AWS Serverless Specification template describing your function.	
cloud9-graph	2018-03-05 00:53:58 UTC-0800	UPDATE_COMPLETE	An AWS Serverless Specification template describing your function.	

Overview Outputs Resources Events Template Parameters Tags Stack Policy Change Sets Rollback Triggers		
Key	Value	Description
APIGatewayRole	arn:aws:iam::744468714261:role/testign12345-APIGatewayRole-16AXPDM025 DGT	
GraphCreatorName	graphcreator-testign12345	
SensorData2DBName	sensordata2db-testign12345	
SensorDataDBName	SensorData-testign12345	
SensorMetadataDBName	SensorMetadata-testign12345	
SensorAssociationDBName	SensorAssociation-testign12345	
SensorData2ESName	sensordata2es-testign12345	

Figure 13

Select that stack and click on the “Outputs” tab on the bottom page. **Keep track of all the Keys and their corresponding Values. You will need them later.**

## Part 2 – Set up REST API

### Step 2.1 – Go to API Gateway:

Go back to the landing page of [aws.amazon.com](https://aws.amazon.com). It should be the same page as depicted in [Step 1.2, Figure 1](#). Use the search bar to find API Gateway.

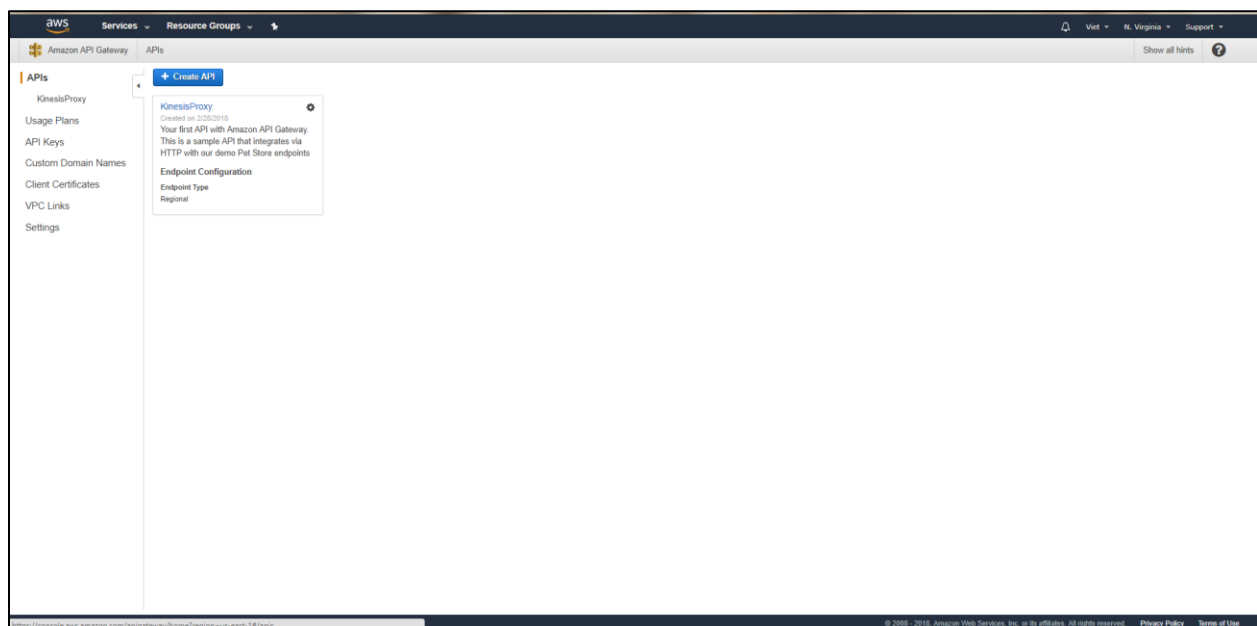
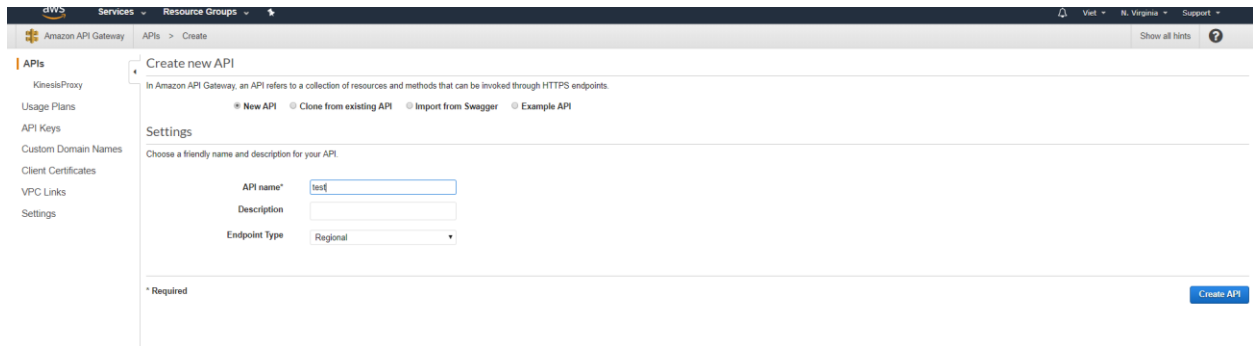


Figure 14

## Step 2.2 – Create a new API:

Create a new API using the “Create API” button.



The screenshot shows the Amazon API Gateway console interface. At the top, there's a navigation bar with 'Services' and 'Resource Groups'. Below it, the breadcrumb 'Amazon API Gateway > APIs > Create' is visible. On the left, a sidebar lists various API Gateway features: APIs, KinesisProxy, Usage Plans, API Keys, Custom Domain Names, Client Certificates, VPC Links, and Settings. The main content area is titled 'Create new API' and includes a sub-header: 'In Amazon API Gateway, an API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.' Below this, there are four radio buttons for selection: 'New API' (which is selected), 'Clone from existing API', 'Import from Swagger', and 'Example API'. Under the 'Settings' section, there's a prompt 'Choose a friendly name and description for your API'. This section contains three input fields: 'API name\*' with the value 'test', 'Description', and 'Endpoint Type' with a dropdown menu set to 'Regional'. At the bottom left, there's a note '\* Required'. At the bottom right, there is a blue button labeled 'Create API'.

Figure 15

Choose “New API” and a API name. Click “Create API”.

### Step 2.3 – Create Resources:

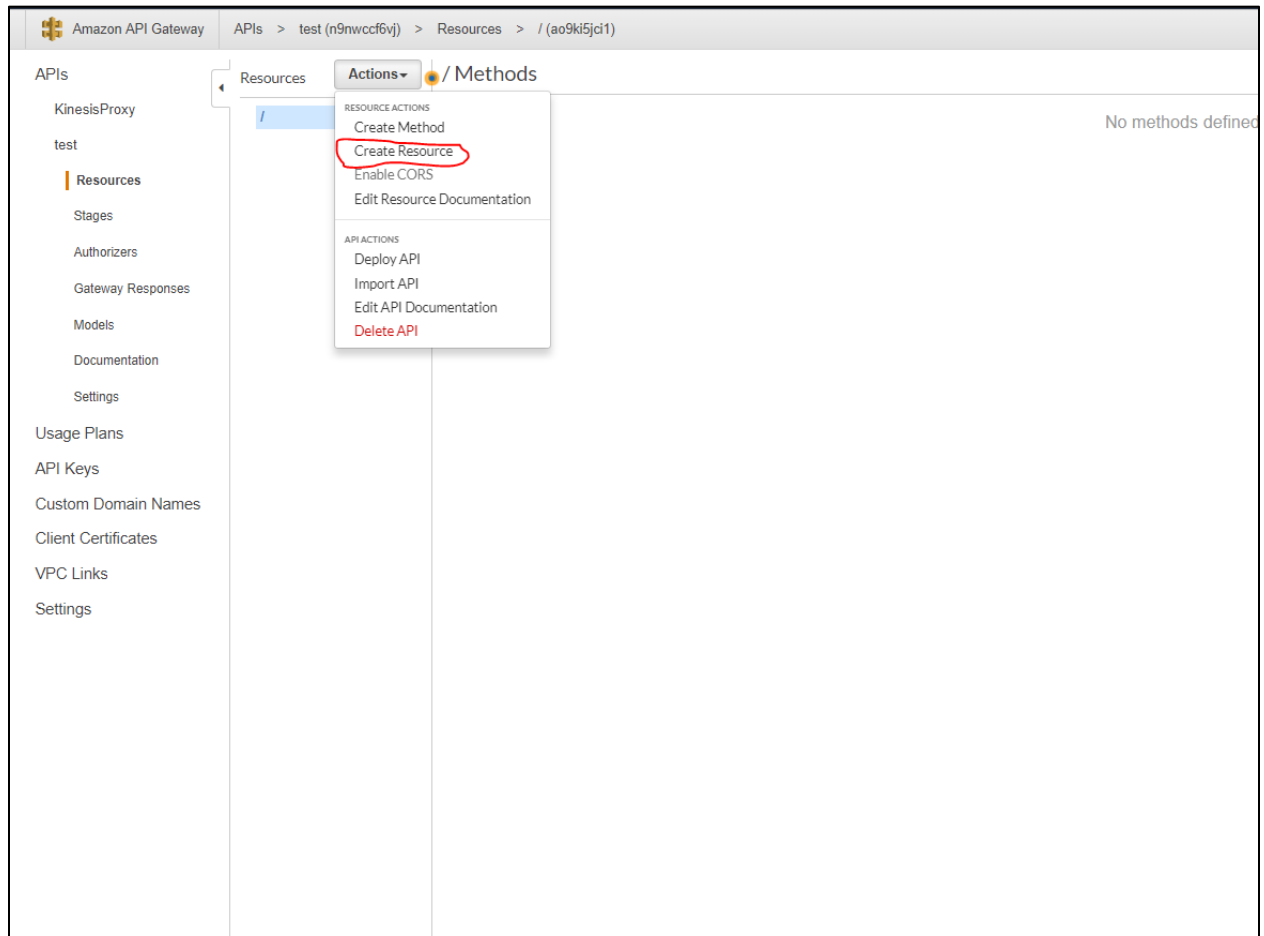


Figure 16

Select Create Resource. Make sure the existing resource “/” is selected.

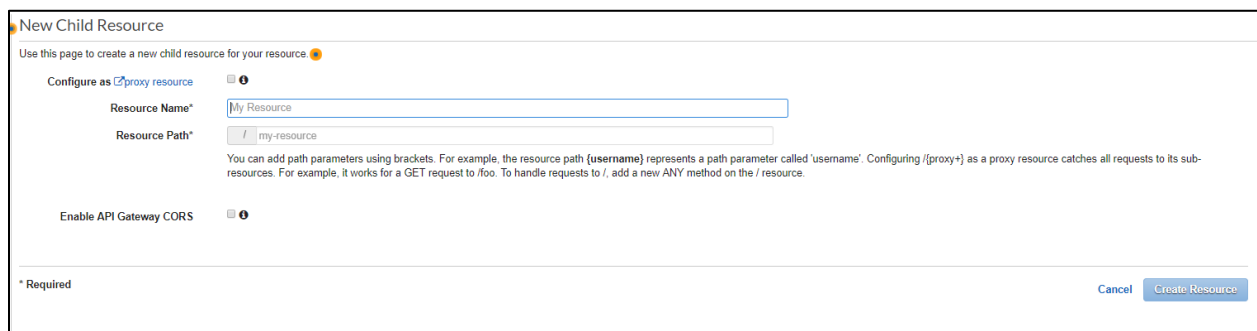


Figure 17

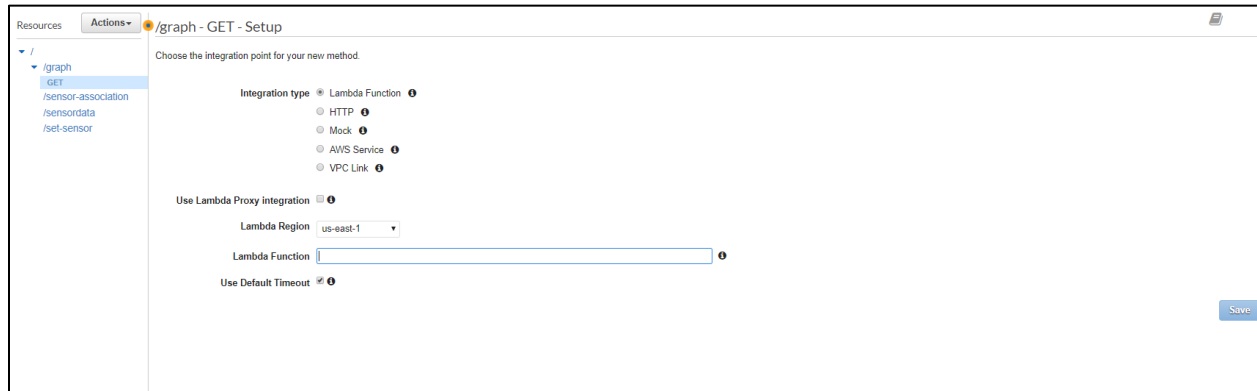
For resource name, input “graph”. Press “Create Resource”. Repeat this step to create three other resources: “set-sensor”, “sensor-association”, and “sensordata”. It should look like this now:



*Figure 18 Resource Creation finished*

## Step 2.4 – Create Resource Methods for /graph

Select **/graph**. Create a GET method.



The screenshot shows the AWS API Gateway console interface for creating a new method. On the left, a sidebar lists resources: `/`, `/graph`, `/sensor-association`, `/sensordata`, and `/set-sensor`. The `/graph` resource is selected, and a new GET method is being created. The main panel is titled `/graph - GET - Setup`. It prompts the user to "Choose the integration point for your new method." Under "Integration type", `Lambda Function` is selected. Other options include `HTTP`, `Mock`, `AWS Service`, and `VPC Link`. Below this, the "Use Lambda Proxy integration" checkbox is unchecked. The "Lambda Region" is set to `us-east-1`. The "Lambda Function" field is empty. The "Use Default Timeout" checkbox is checked. A "Save" button is located in the bottom right corner.

Figure 19

For the lambda function name, input the **GraphCreatorName** key from the Outputs in Step 1.5, Figure 13. Press Save.

## Step 2.5 – Create Resource Methods for /set-sensor

Select **/set-sensor**. Create a POST method. Select **AWS Service** for Integration Type. Choose the AWS region in which you deployed your stack. Select **DynamoDB** for **AWS Service**. Select **POST** for the **HTTP Method**. For **Action**, input "PutItem". Copy and paste the **APIGatewayRole** value you got from the stack in Step 1.5. Press Save.



The screenshot shows the AWS API Gateway console interface for creating a new method. The top bar shows the breadcrumb `/sensor-association - POST - Setup`. The main panel prompts the user to "Choose the integration point for your new method." Under "Integration type", `AWS Service` is selected. Other options include `Lambda Function`, `HTTP`, `Mock`, and `VPC Link`. Below this, the "AWS Region" is set to `us-east-1`. The "AWS Service" is set to `DynamoDB`. The "AWS Subdomain" field is empty. The "HTTP method" is set to `POST`. Under "Action Type", `Use action name` is selected. The "Action" field is set to `PutItem`. The "Execution role" is set to `arn:aws:iam::744468714261:role/testsign12345-APIGatewayRole-16AXPDM025DGT`. The "Content Handling" is set to `Passthrough`. The "Use Default Timeout" checkbox is checked. A "Save" button is located in the bottom right corner.

### Step 2.6 – Create Method for /sensordata

Select **/sensordata**. Create a POST method. For the AWS Lambda function, input the **SensorData2ESName** value from Step 1.5.

### Step 2.7 – Create Resource Methods for /sensor-association

Select **/sensor-association**. Create a POST method. Select **AWS Service** for Integration Type. Choose the AWS region in which you deployed your stack. Select DynamoDB for **AWS Service**. Select POST for the **HTTP Method**. For **Action**, input “PutItem”. Copy and paste the **APIGatewayRole** value you got from the stack in Step 1.5. Press Save.

Great! You’re all done. Go ahead and use some of the items in the test folder to test it out.