

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:

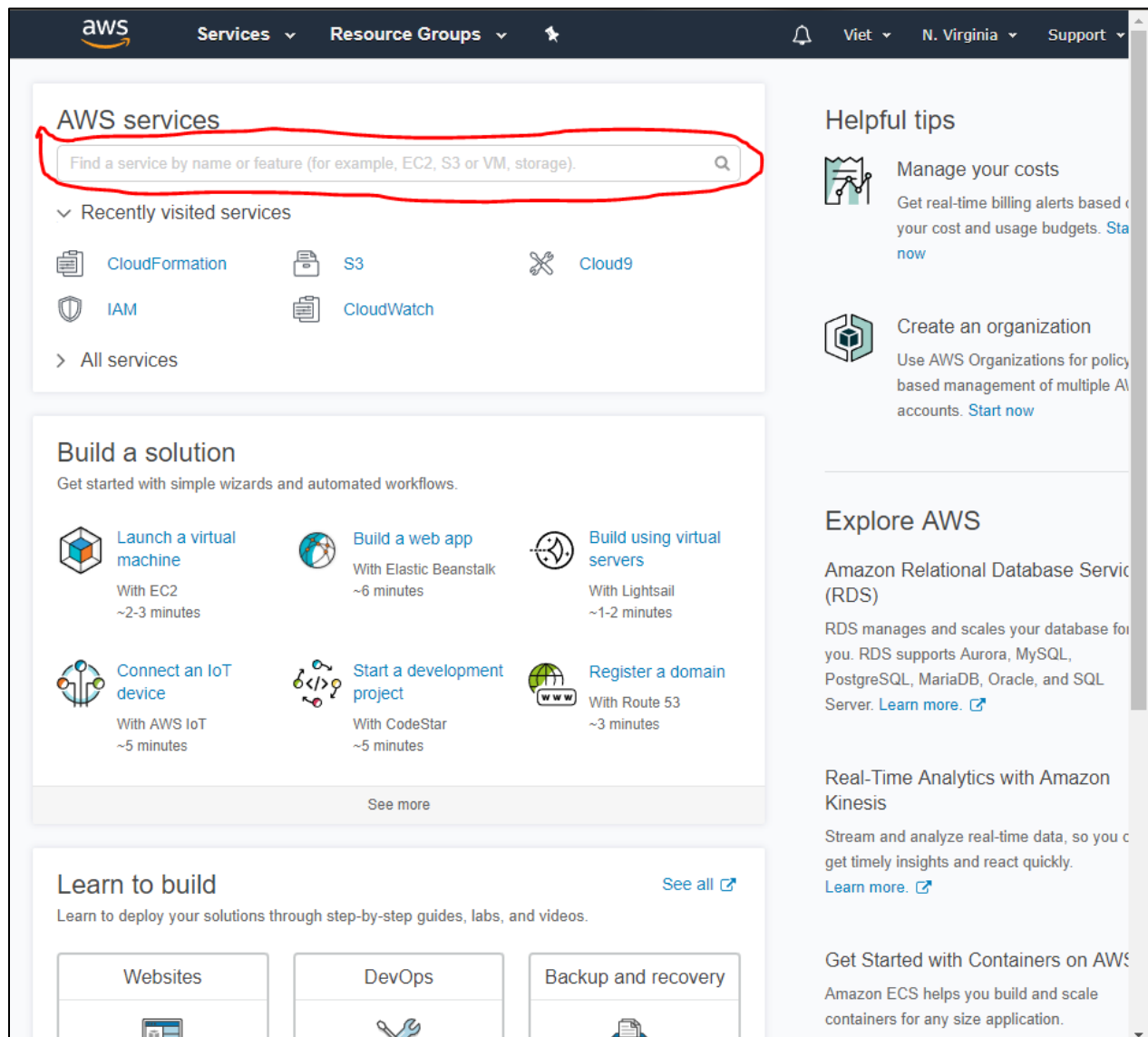


Figure 1 AWS Landing Page

Type in S3 into the search bar circled red.

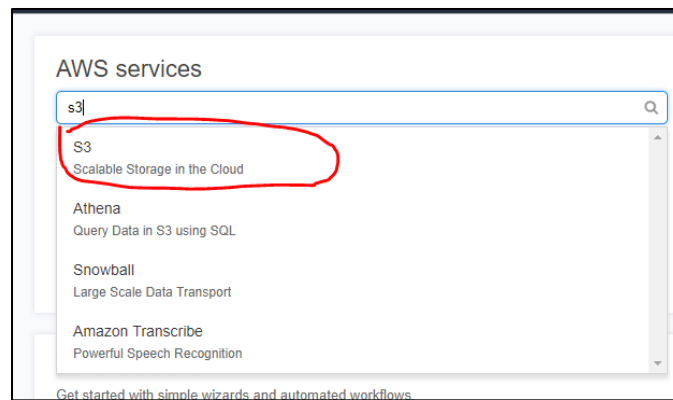


Figure 2 Search Bar

Select **S3**.

Step 1.3 – Create a Bucket:

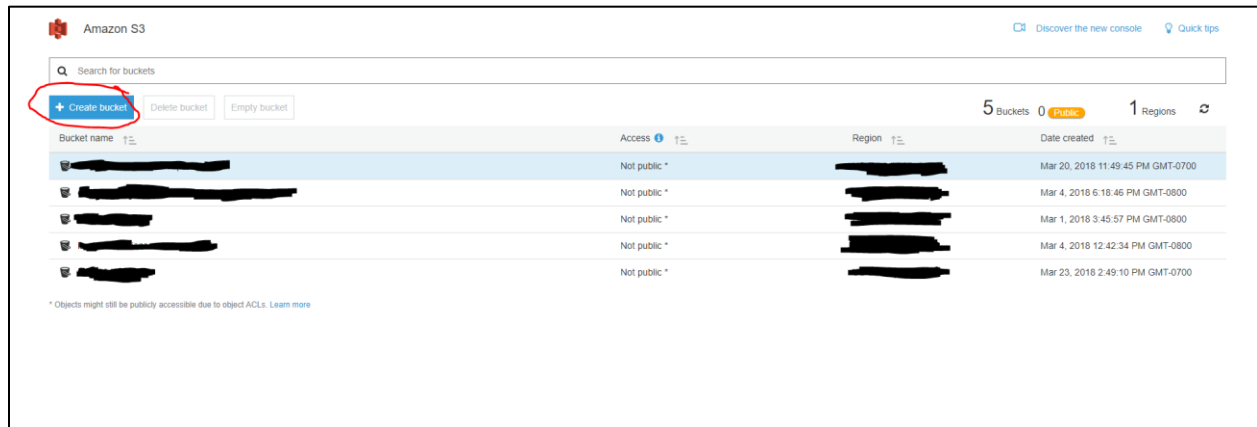


Figure 3 Amazon S3 Page. Personal details removed.

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:

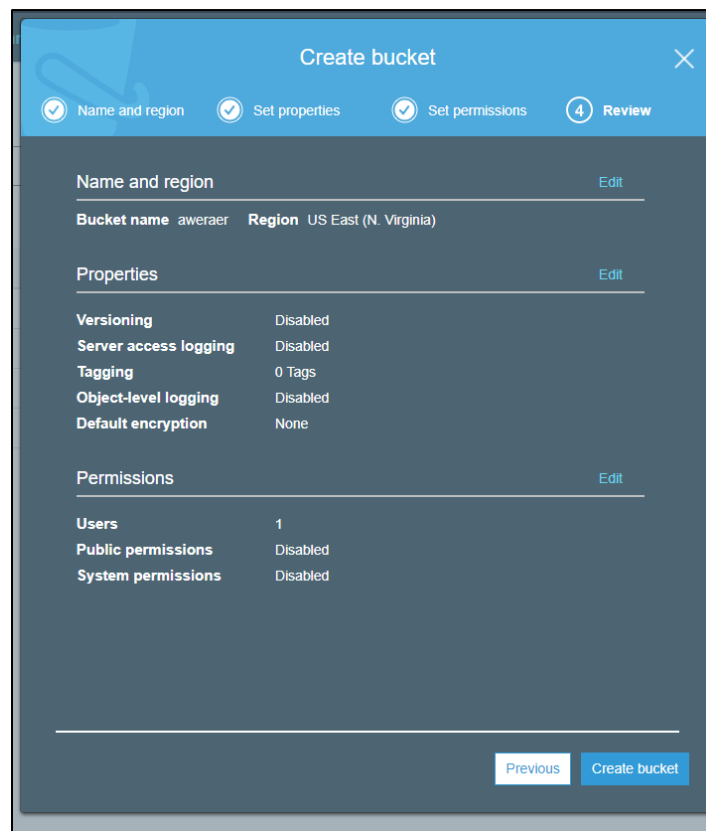


Figure 4 S3 Bucket Summary

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”.

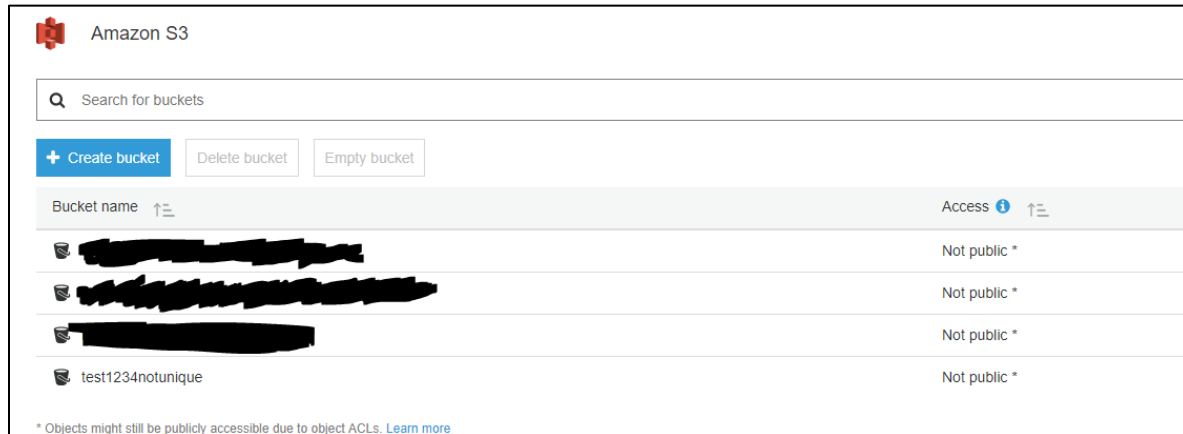


Figure 5

The screen should look like this:

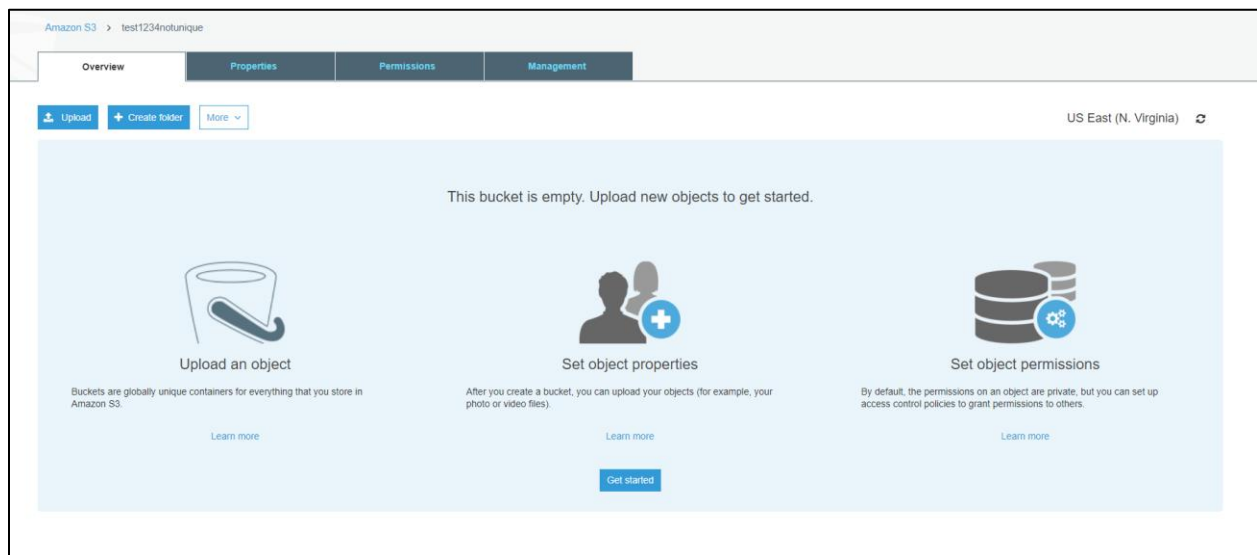


Figure 6

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. It should be the same page as depicted in [Step 1.2, Figure 1](#). Use the search bar to find AWS Cloudformation.

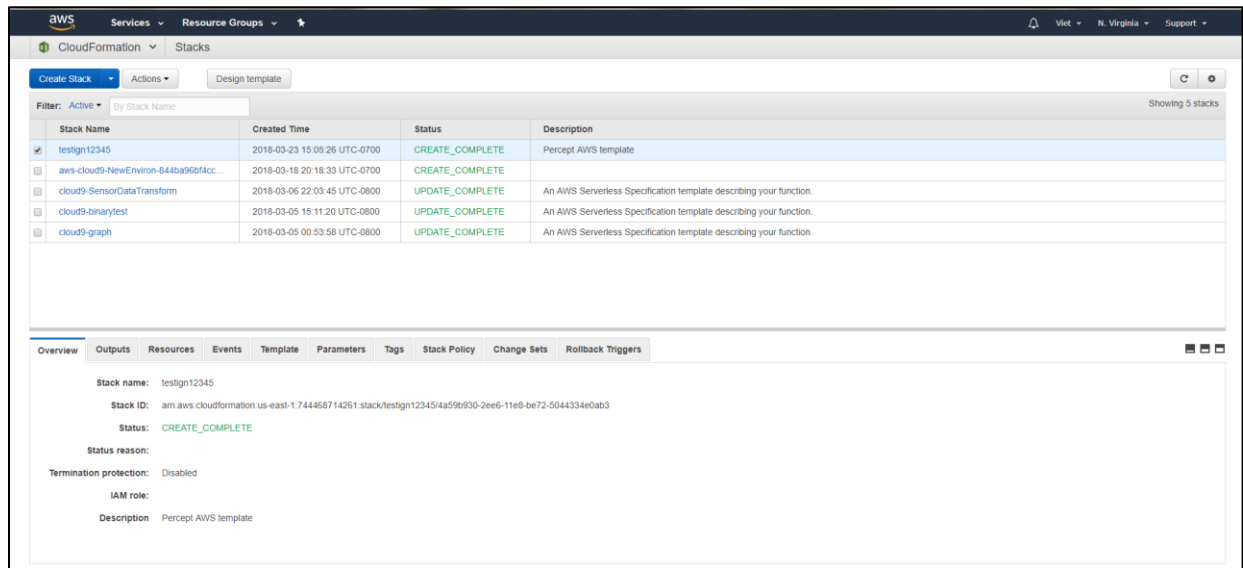


Figure 7 Cloudformation Landing Page

Click “Create Stack”.

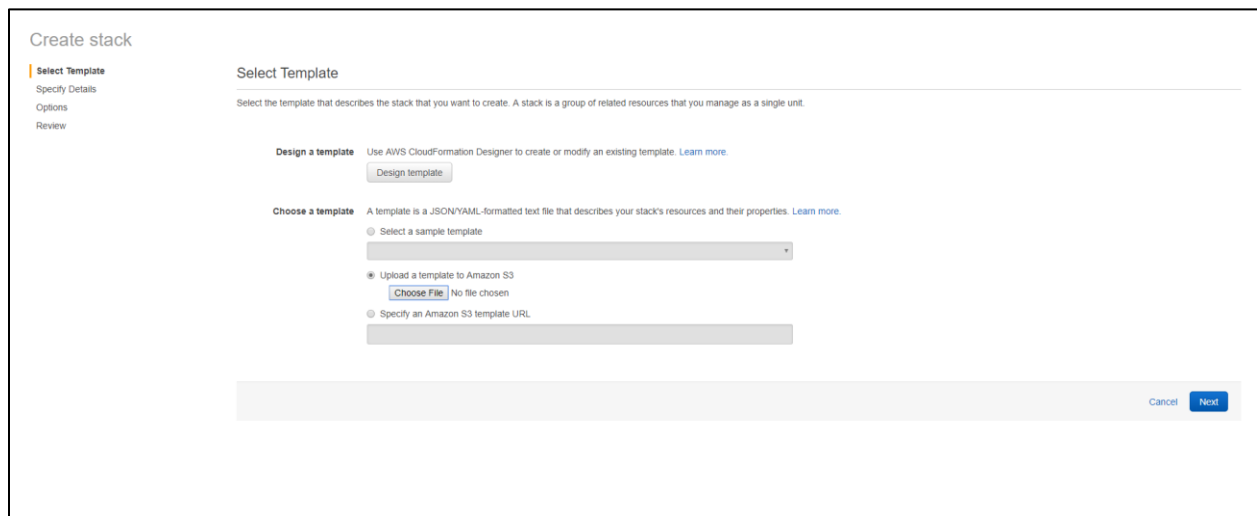


Figure 8 1st Page of Stack Creation

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	

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	

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.

The screenshot shows the AWS CloudFormation console. At the top, there's a navigation bar with 'Services', 'Resource Groups', and a star icon. Below that, the 'CloudFormation' service is selected, and the 'Stacks' view is active. A 'Create Stack' button and an 'Actions' dropdown are visible. A search bar for 'By Stack Name' is present, and it shows 'Showing 5 stacks'.

Stack Name	Created Time	Status	Description
pepperoni	2018-03-27 03:27:30 UTC-0700	CREATE_COMPLETE	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.

Below the stack list, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. The 'Outputs' tab is selected. It shows a table with three outputs:

Key	Value	Description	Export Name
APIEndpoint	https://1xh3j59wmg.execute-api.us-east-1.amazonaws.com/Beta		
Region	us-east-1		
ElasticEndpoint	https://search-sensores-pepperoni-rfiwdyu4jqhmd3ak7ndeylzo1.us-east-1.es.amazonaws.com		

A red circle is drawn around the 'Outputs' tab and the three output rows. At the bottom of the console, there's a footer with 'Feedback', 'English (US)', and copyright information.

Figure 13 Outputs tab on AWS

Keep track of the three keys: APIEndpoint, Region, and ElasticEndpoint. You will need them later.

Step 2 – Post-Stack-Setup:

After setting up the stack, we will need to configure one of the resources, the Elasticsearch cluster. If you're interested in learning more about setting up Elasticsearch (specifically the setup we're doing), consider looking at the following links.

[Elasticsearch Aliasing](#)

[Elasticsearch Mapping](#)

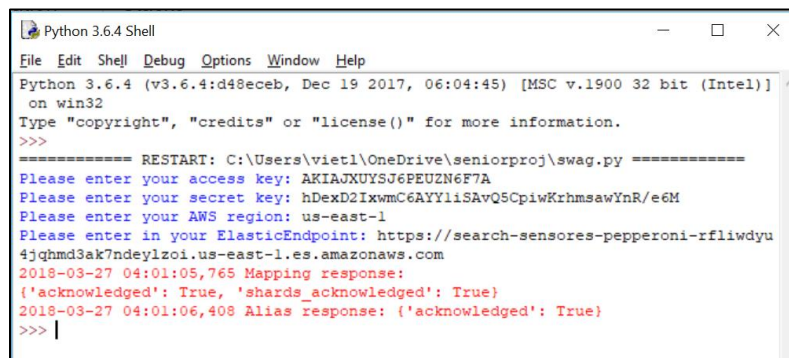
Step 2.1 – Finding Your Access Key and Secret Key

You first need to find your access and secret key for your AWS account. Instructions can be found [here](#).

Step 2.2 – Configuring Elasticsearch Cluster

First, make sure that you have the Python 3.x and the Python library [Requests](#) installed. If you do not have Python, download the latest version from [here](#). Then, assuming you only have Python 3.x installed, run the following command (minus the single quotes): 'pip install requests'.

Run the script via command line or the IDLE, the Python GUI. It will ask you for your access key and secret key, which you should have found in Step 2.1. It will then ask you for your AWS Region, which is the Region key from **your** Cloudformation output in Step 1.5, Figure 12. **DO NOT COPY THE VALUES FROM THE FIGURE.** It will not work if you do so. Finally, it will ask you for your ElasticEndpoint, the key value from Step 1.5, Figure 12. Using the GUI, it should look like this.



```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\viet1\OneDrive\seniorproj\swag.py =====
Please enter your access key: AKIAJXUYSJ6PEUZN6F7A
Please enter your secret key: hDexD2IxwmC6AYYliSAvQ5CpiwKrhmsawYnR/e6M
Please enter your AWS region: us-east-1
Please enter in your ElasticEndpoint: https://search-sensores-pepperoni-rfliwdyu
4jqhmd3ak7ndeylzo1.us-east-1.es.amazonaws.com
2018-03-27 04:01:05,765 Mapping response:
{'acknowledged': True, 'shards_acknowledged': True}
2018-03-27 04:01:06,408 Alias response: {'acknowledged': True}
>>>
```

Figure 13 Expected output from es_setup.py

Step 3 – Running the backend

At this stage, all the setup is done. To POST sensor data, use **your** APIEndpoint key from the Cloudformation Outputs tab in Step 1.5, Figure 13. Append the path **/sensordata** to the URI, and POST away.

Here are the following formats for the corresponding resources. The full path is not shown, just the ending. For example, if **/graph** is referenced, the full url would be **APIEndpoint/graph**, where **APIEndpoint** is the key representing your URI listed in the Cloudformation Outputs tab in Step 1.5, Figure 13.

For all methods, uid can be thought of as the unique identifier of a sensor. **uids must be unique!**

/graph?uid={uid}

This resource only supports the GET method, and requires a query parameter, uid, to be specified.

/sensordata

This resource only supports the POST method. The Content-Type *must* be application/json. The body must follow the JSON format as follows:

```
{
  "uid": string,
  "PowerFactor": integer,
  "CurrentSummationDelivered": integer,
  "InstantaneousDemand": integer,
  "Current": integer,
  "Timestamp": string
}
```

Timestamp must be the ISO8601 format UTC time.

/set-sensor

This resource only supports the POST method. The Content-Type *must* be application/json. The body must follow the JSON format as follows:

```
{
  "uid" : string
  "unit" : string
  "data" : string
  "name": string
  "attribute": string
}
```

Unit is the unit of measurement for the sensor. Attribute is the attribute you wish to display using the **/graph** API. Name is a custom name, attached to that specific uid. Data is a JSON object serialized as a string, representing arbitrary tags you wish to attach to a specific sensor/uid.