**AWS CloudFormation Template creation using Troposphere**

The Troposphere library allows easier creation of AWS CloudFormation JSON template. Using Python code troposphere describes the AWS resources. Currently other than AWS CloudFormation, Troposphere has also some support for OpenStack using Heat.

**Benefits of troposphere over AWS CloudFormation**

The main benefit of Troposphere is that it can create dynamic CloudFormation templates. For example, if we need to get some user input (Region, AMI, OS types etc.) which can be dynamic and based on that we need to create CloudFormation template, in this scenario troposphere is very handy. We can create reusable Python code using troposphere and AWS Python SDK (boto3) to generate dynamic CloudFormation template based on user input. Other than this, troposphere has some other benefits over CloudFormation. For example:

* Looping over user input / returned object is possible when we use troposphere to create CloudFormation template.
* Writing Python code is relatively easier that writing JSON.
* It provides some inbuilt error checking which help to identify the problem way before deploying the stack.

Other than Troposphere, HashiCorp Terraform also does the same job. Terraform uses HCL (HashiCorp Configuration Language) as its template language while troposphere uses Python. So, while both does the same job efficiently, troposphere get some edge over Terraform as it uses very well-known scripting language and well defined CloudFormation documentation.

**Installation and pre-requisite for CloudFormation template generation**

Troposphere can be installed by Python, pip distribution system as below:



To run troposphere to generate CloudFormation template some basic requirements need to be fulfilled:

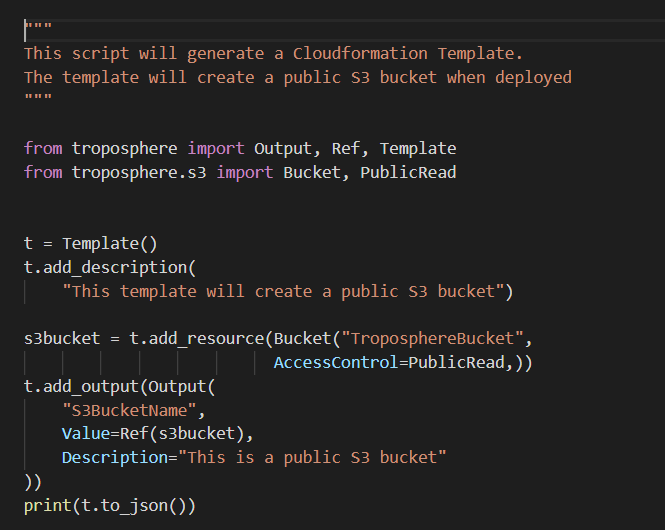
* Python3 should be installed and configured
* AWSCLI should be installed and AWS configuration should be done
* Boto3 should be installed

Once all these requirements are fulfilled and troposphere is available we can go ahead create a script to generate CloudFormation template.

**Example**

In the following section we will see a simple example where we will write a troposphere code which will create a CloudFormation template to generate a public S3 Bucket.

*Troposphere Code:*



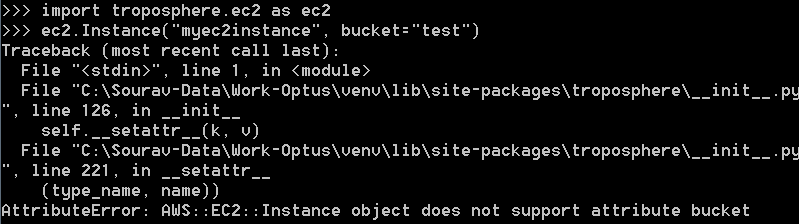
*Generated CloudFormation Template:*



You can find more examples [here](https://github.com/cloudtools/troposphere/tree/master/examples).

**Example of error checking**

If we try to set incorrect property to AWS resources, troposphere identifies that and provides error message. In the current scenario, we are trying to add invalid attribute “bucket” to EC2 instance and we get the following error message.



As EC2 instance don’t support bucket attribute, we get the AttributeError message.

**Summary**

Overall troposphere is an excellent Python library for generating dynamic CloudFormation template. Though some of the newest AWS services are still not available and it lacks some good documentation, it is very handy and reusable when creating CloudFormation template.

**References**

<https://github.com/cloudtools/troposphere>

<https://groups.google.com/forum/#!forum/cloudtools-dev>

<https://packetlost.com/blog/2017/09/04/dynamic-cloudformation-templates-troposphere-and-boto3/>

**Tags**

#AWS, #CloudFormation #InfrastructureAsCode #Troposphere #Boto3 #Python #CloudComputing #IaaS #PaaS #SaaS #AmazonWebServices #Cloud