**TaskCat**

The process of automating the process of checking and deploying the templates can be done using an opensource tool, i.e., **Taskcat**, a tool for testing CloudFormation templates.

**Taskcat** is one of the tools which can optimize the run/test of the CF template.

Taskcat deploys the CF template in regions and availability zones and generates a pass/fail grade for the region which helps in testing without actually launching the resources the in the cloud and just returning the pass/fail grade.

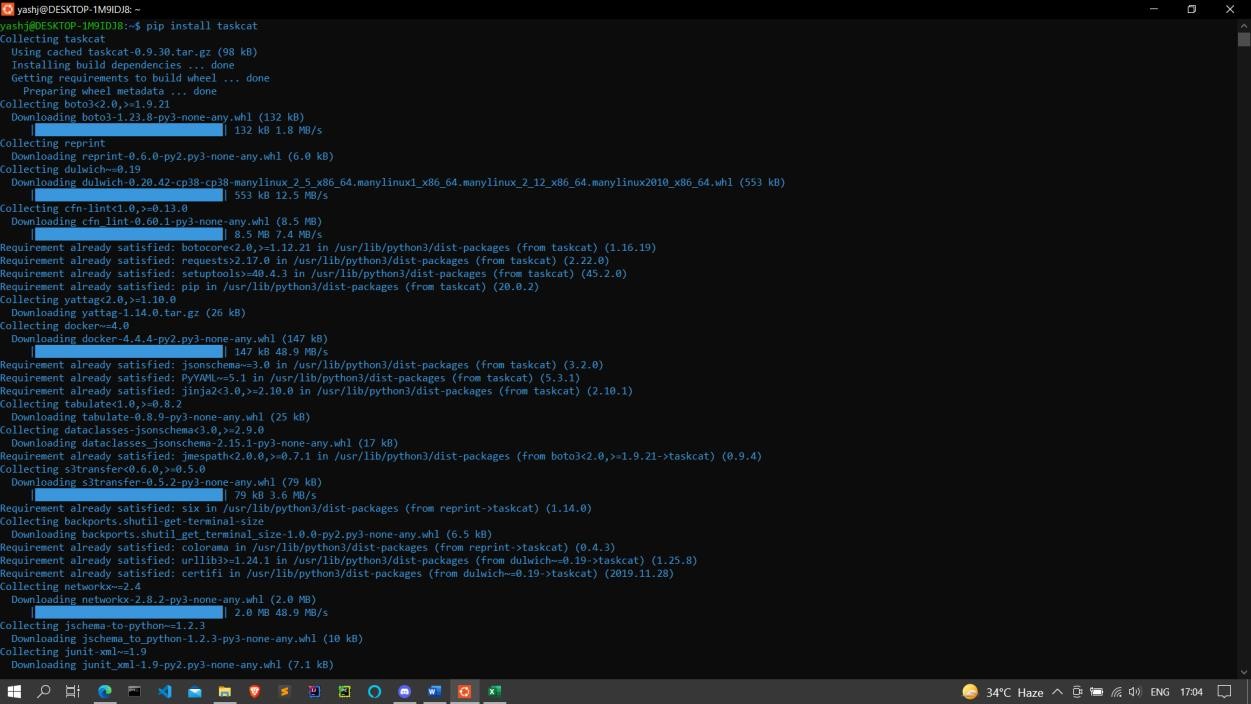
# Taskcat Installation 

Requirements 

* Linux, as windows is not supported by Taskcat, for windows we can use WSL.
* Docker
* Git
* Python3 – versions below 3.10
* AWS CLI Installation –

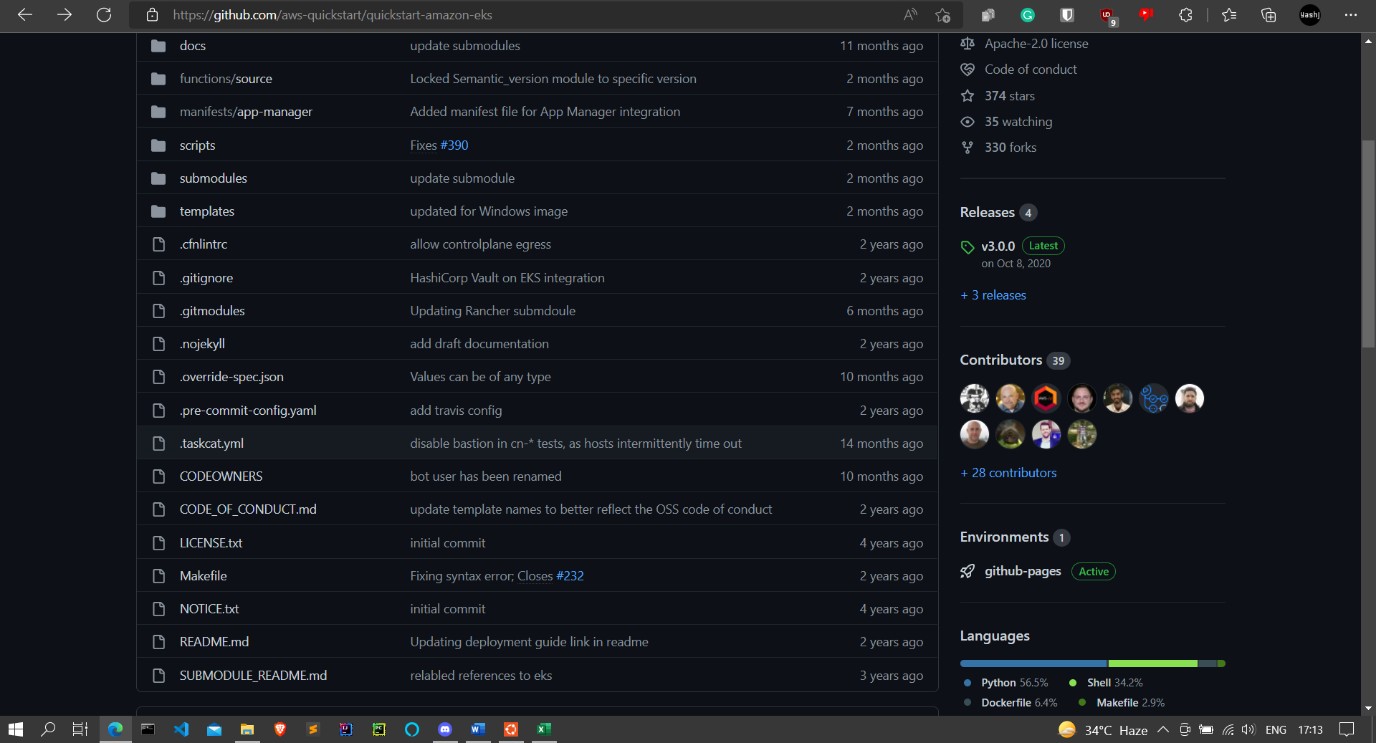
After resolving all the requirements, we run 

# pip install taskcat



CF Template to test on -- [EKS](https://github.com/aws-quickstart/quickstart-amazon-eks).

Taskcat requires the project to contain a .taskcat.yml file to run the test and link to the CF templates.



.taskcat.yml has been provided in the [EKS](https://github.com/aws-quickstart/quickstart-amazon-eks) repository.

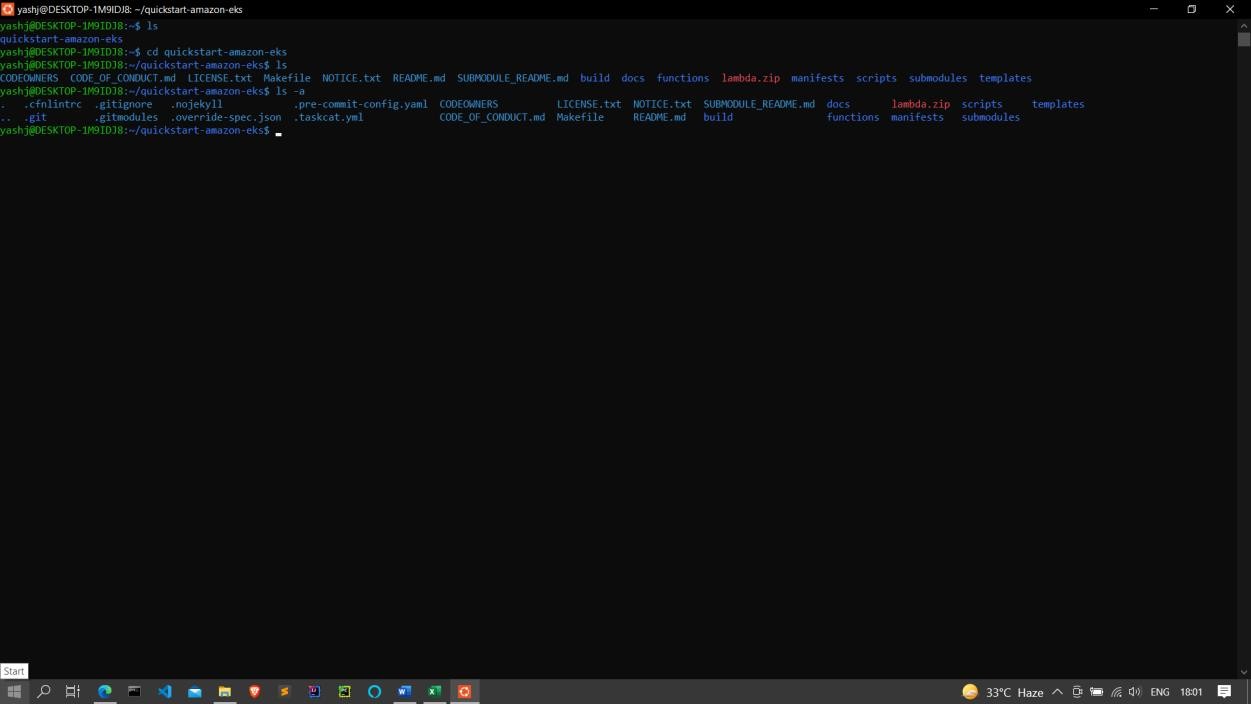
# Running taskcat to test the CF templates 

Configure the aws cli & check the credentials 

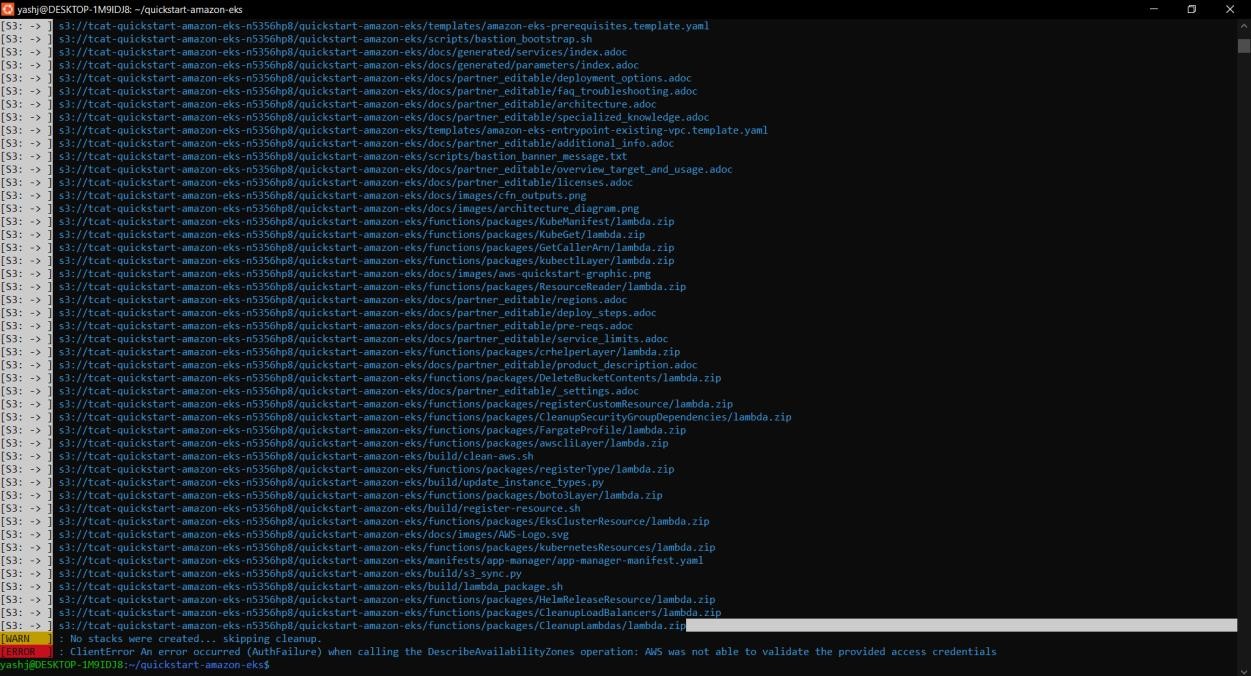
# aws configure

**aws sts get-caller-identity**

We have to clone the repository and navigate into it and run taskcat to test the templates.



# taskcat test run

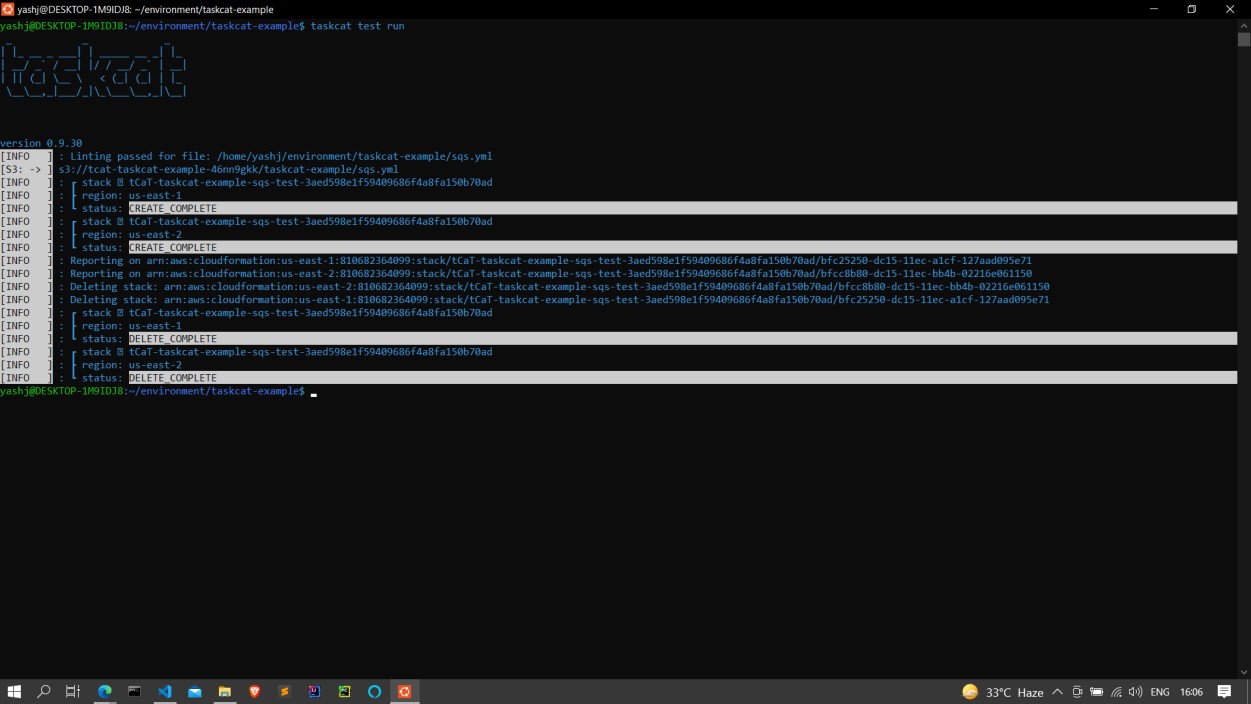


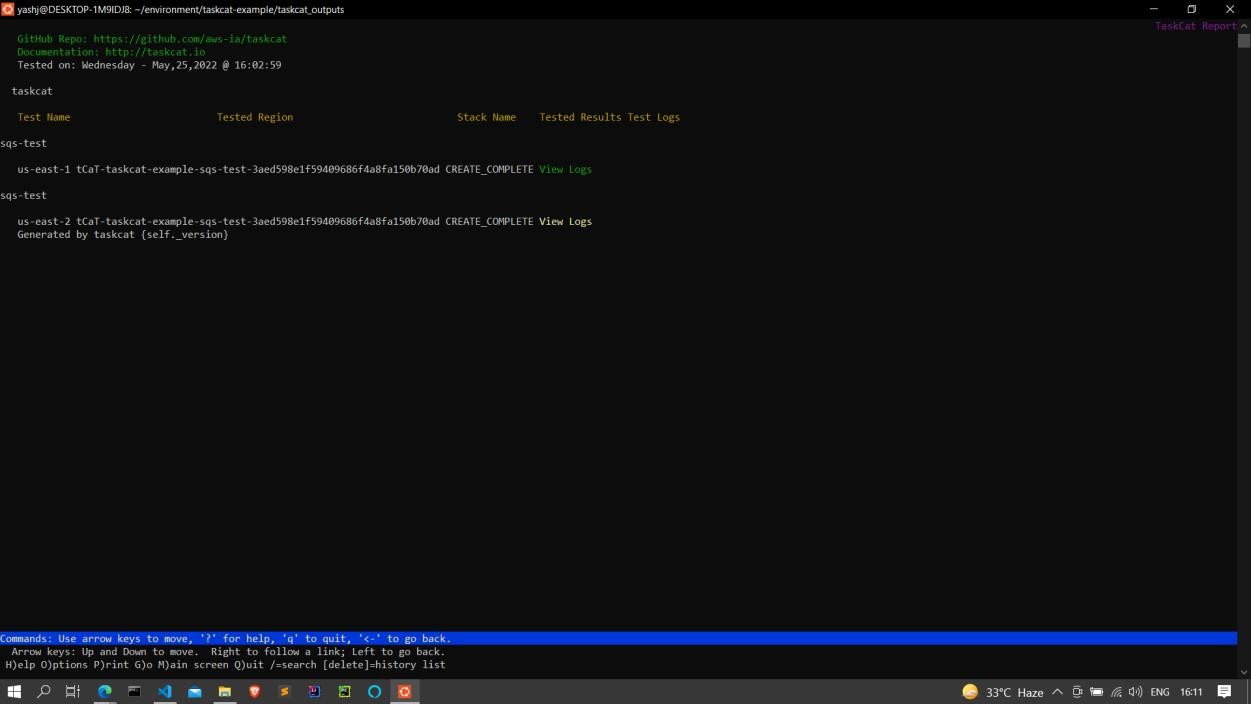
Here in this example, my AWS account is not authorised to use certain regions defined in the taskcat.yml file, thus is giving AZ error.

The result of the test is stored in taskcat\_outputs folder.

Taskcat can be integrated to the CI/CD pipeline, where when the CF template is pushed along with taskcat.yml file, the test can be automated and cf template can be sent forward without any issues upon successfully passing from the taskcat tests.

The above solutions will help to test the CF template with incurring 0 costs.

Sample working of taskcat on sqs.yml template 

Output in lynx cmd browser 