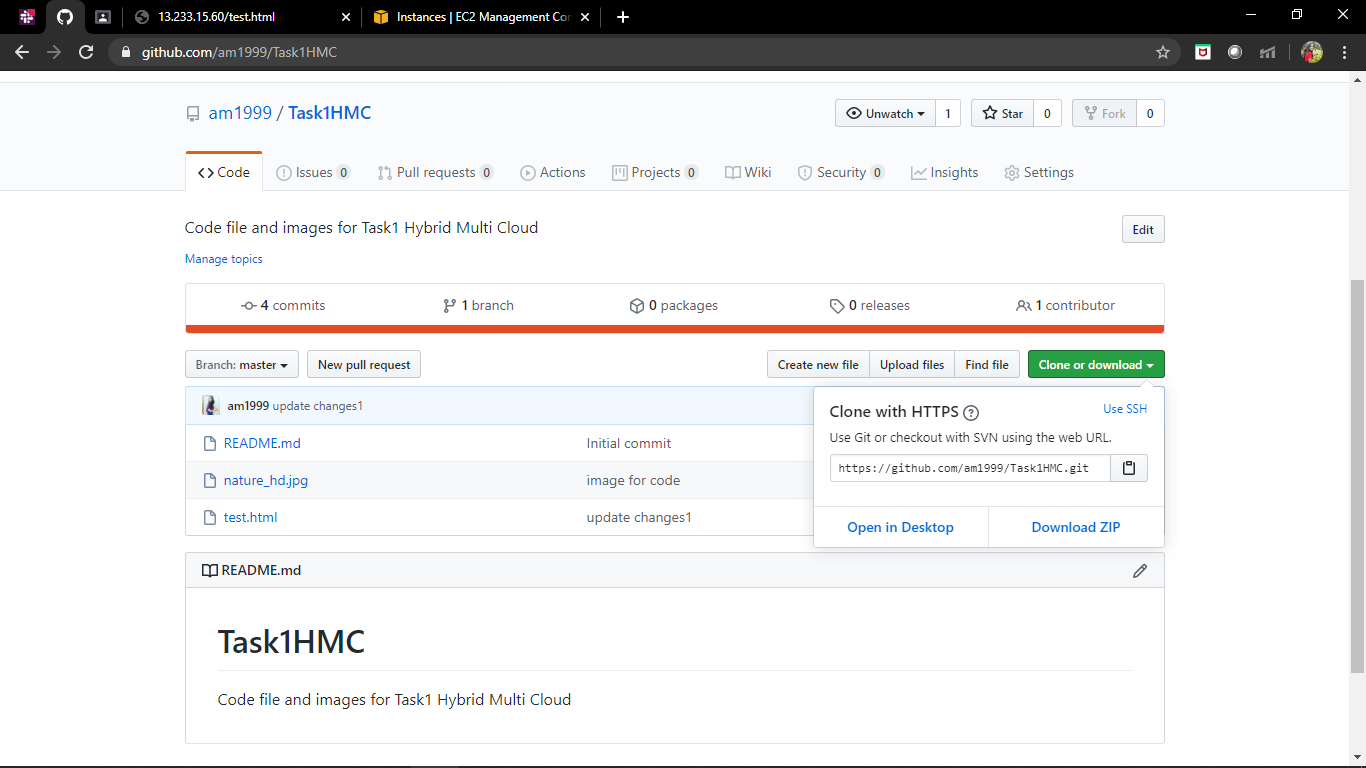
**TERRAFORM-AWS TASK 1** Date: 6th June 2020, Saturday

TASK description: Create and Launch Application using Terraform.

1. Create the key and security group which allow the port 80.
2. Launch EC2 instance.
3. In this Ec2 instance use the key and security group which we have created in step 1.
4. Launch one Volume (EBS) and mount that volume into /var/www/html
5. Developer have uploded the code into github repo also the repo has some images.
6. Copy the github repo code into /var/www/html
7. Create S3 bucket, and copy/deploy the images from github repo into the s3 bucket and change the permission to public readable.
8. Create a Cloudfront using s3 bucket (which contains images) and use the Cloudfront URL to update in code in /var/www/html.
9. Link to Github Repository: <https://github.com/am1999/Task1HMC>

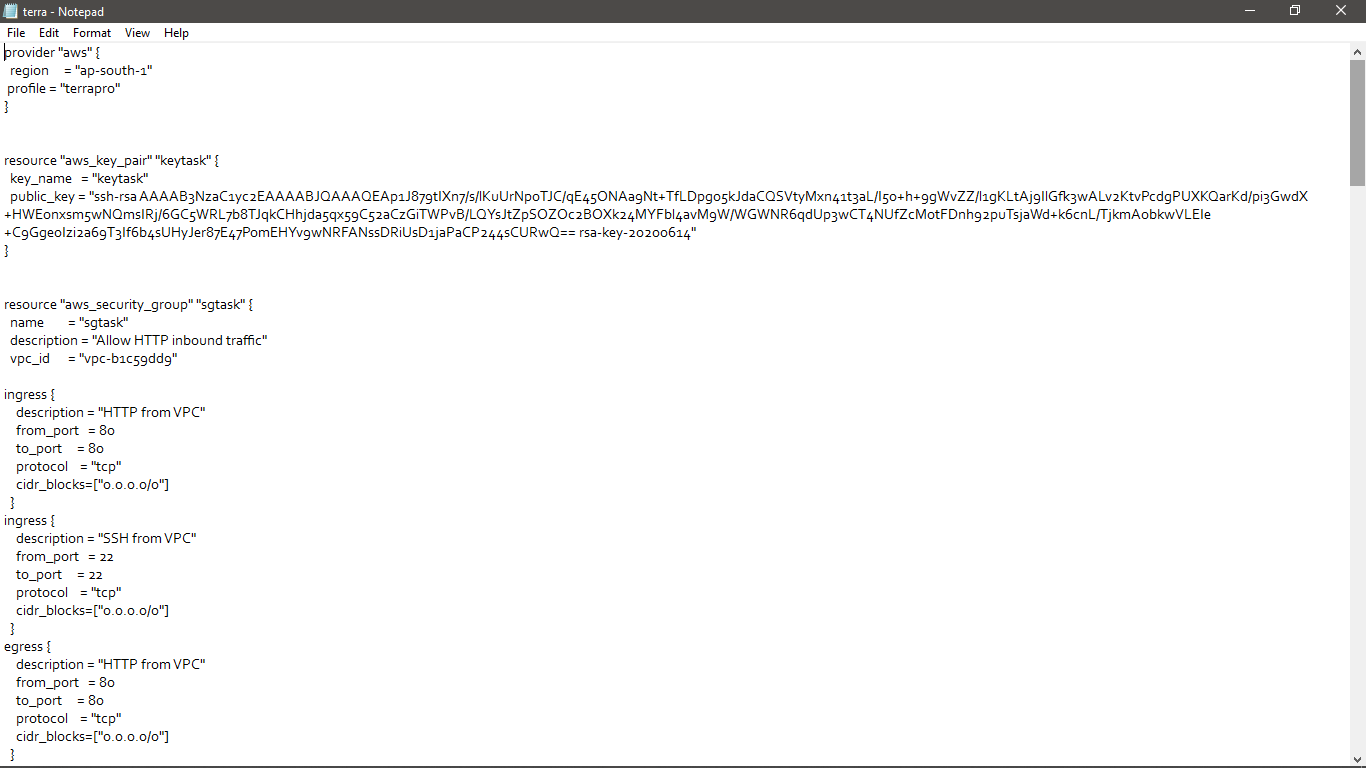
STEP 0: Create a repository and upload your webpage files and images.

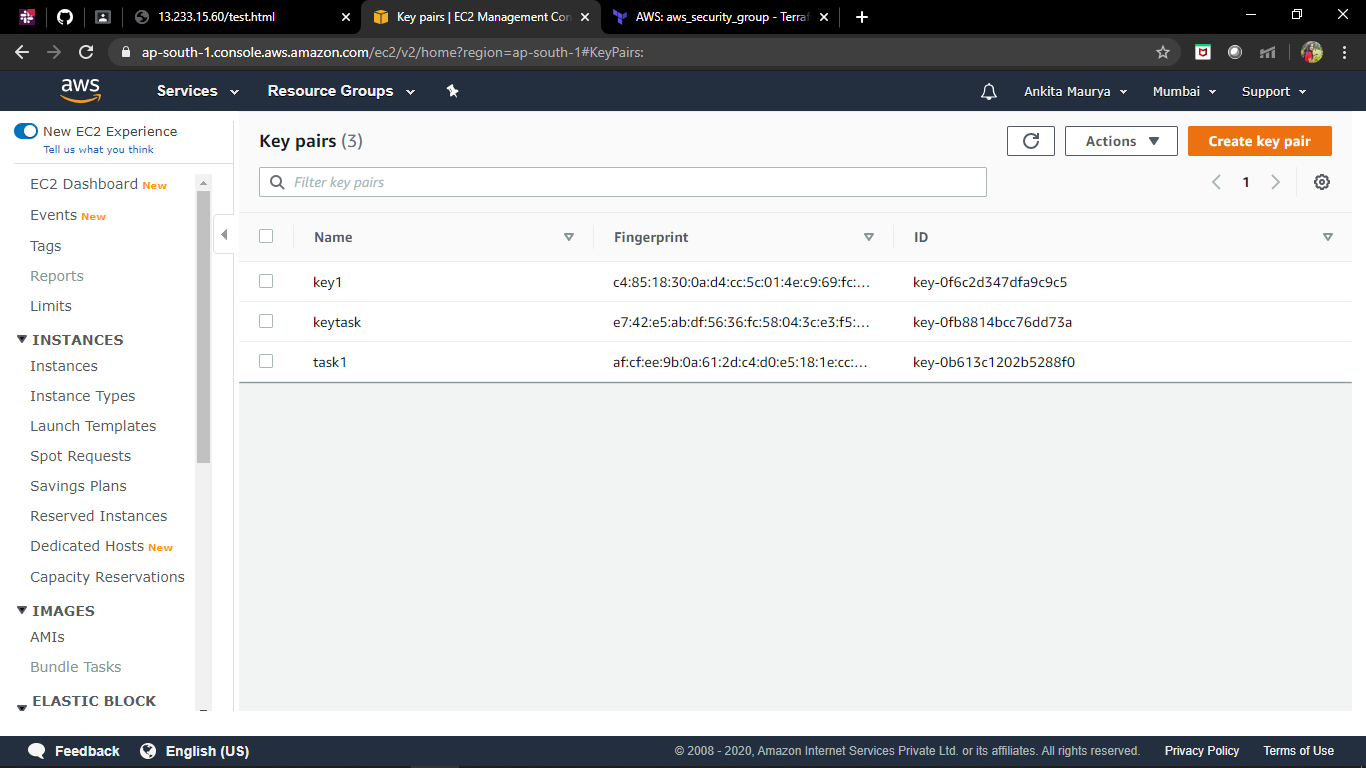


**Terraform Code with corresponding results of AWS Web UI:**

STEP 1: Configure the Provider and Create the keypair.

The Amazon Web Services (AWS) provider is used to interact with the many resources supported by AWS. The provider needs to be configured with the proper credentials before it can be used.

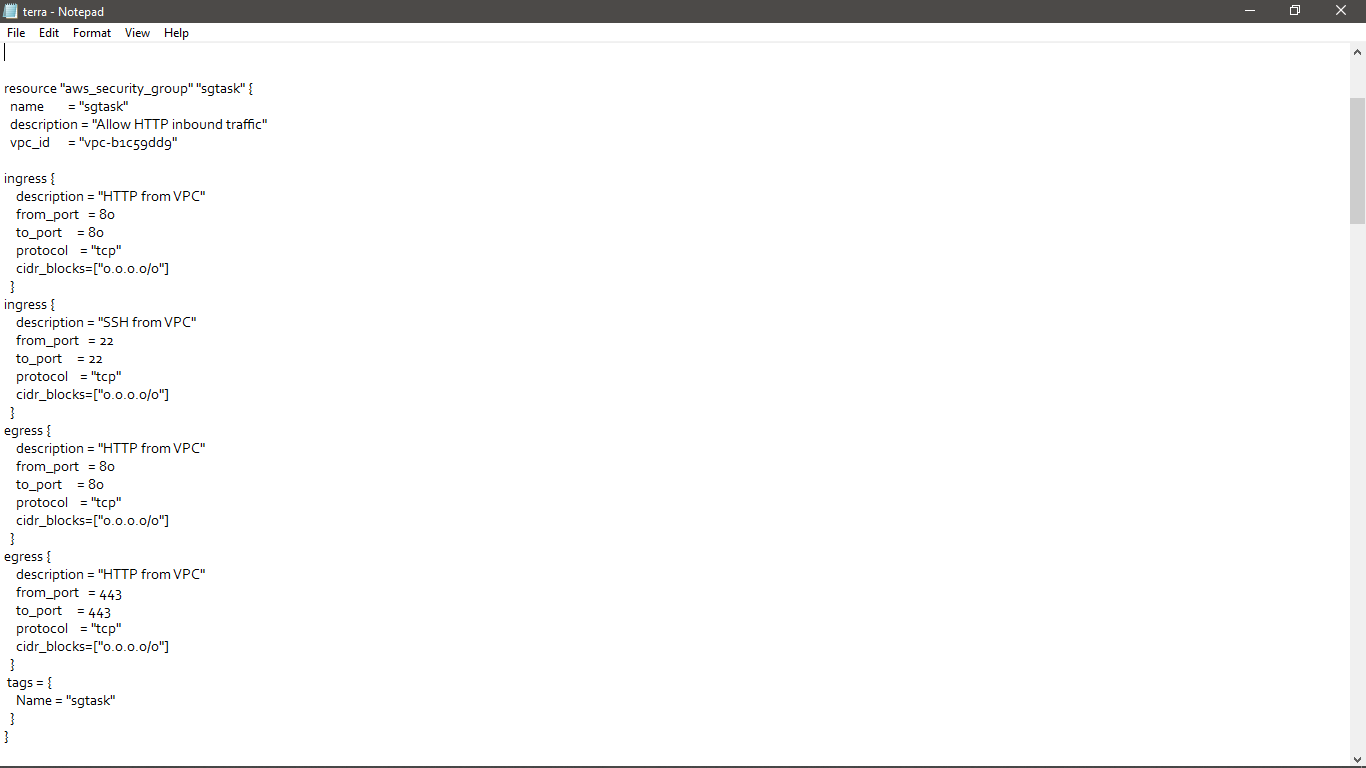


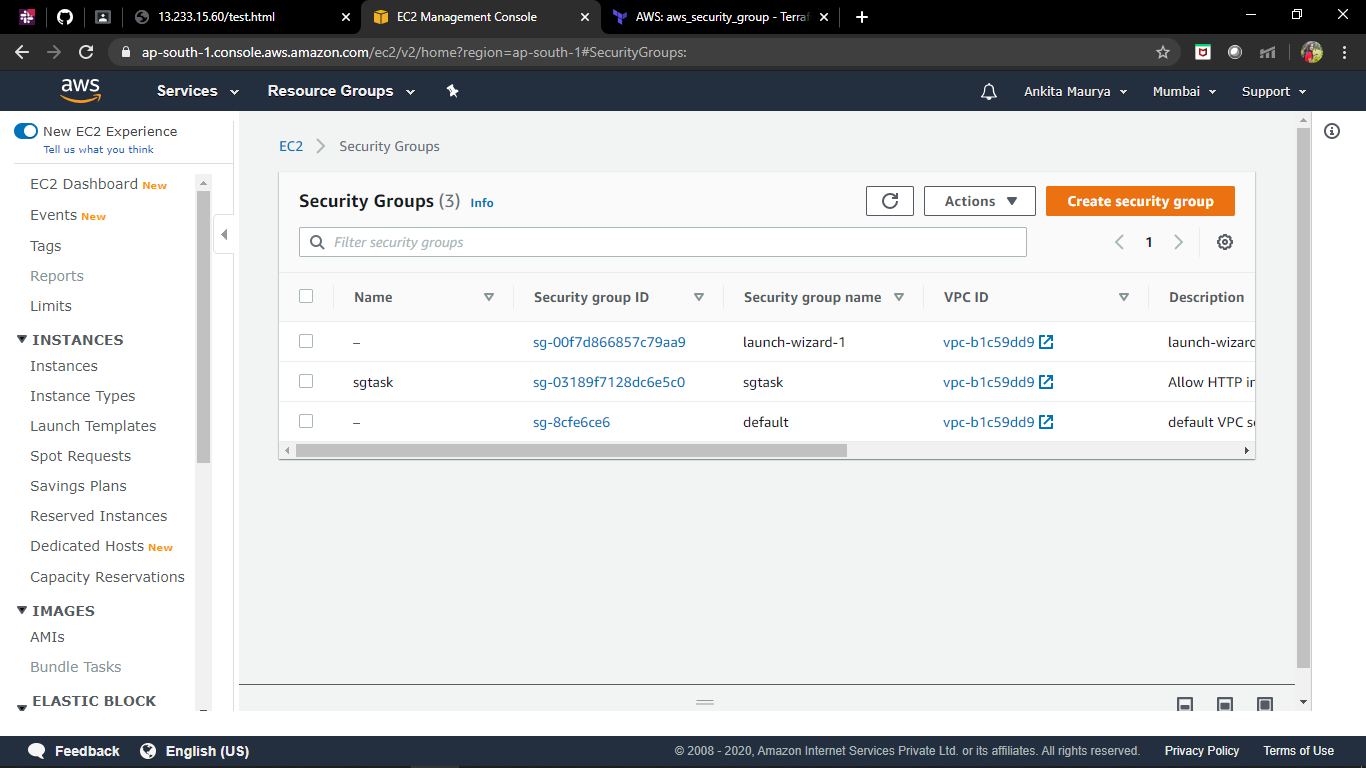


Step 2: Create Security Group

aws\_security\_group provides details about a specific Security Group.

Allow port 80 (HTTP).



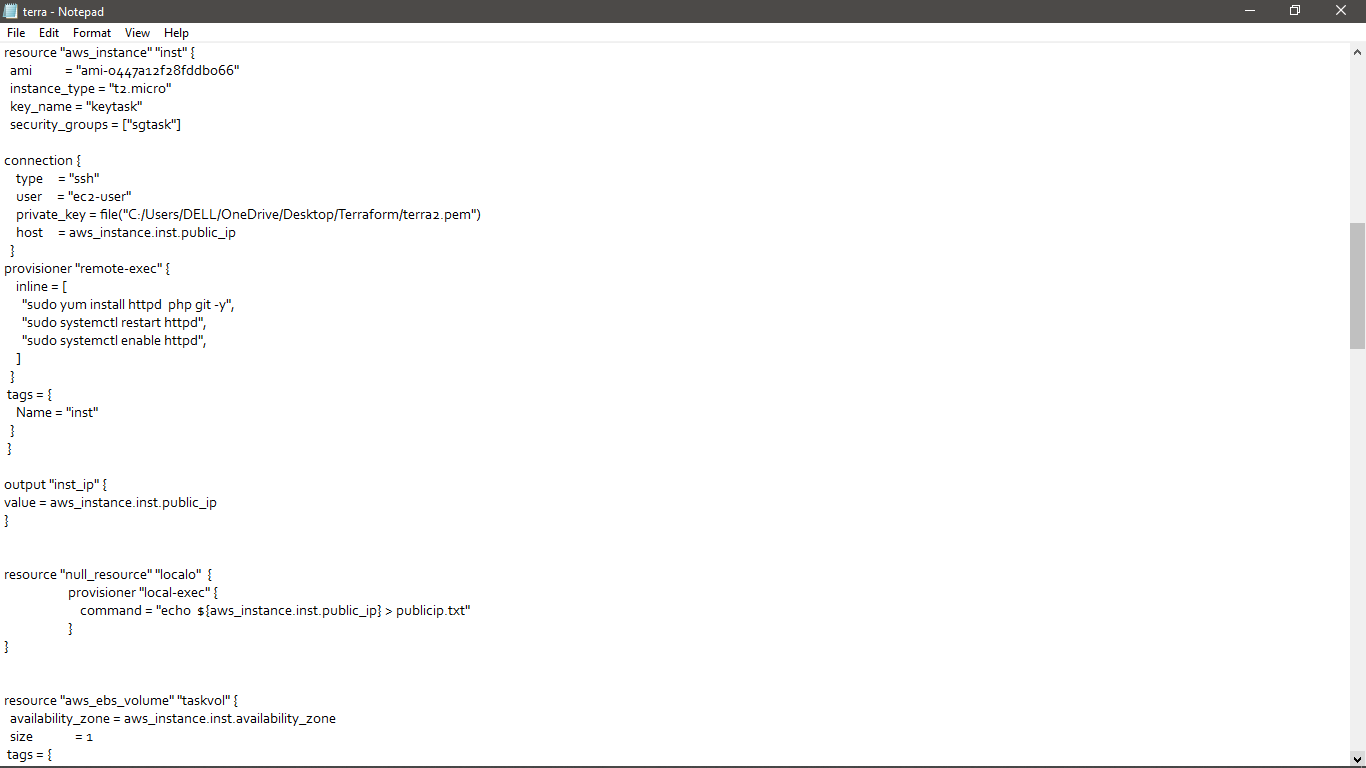


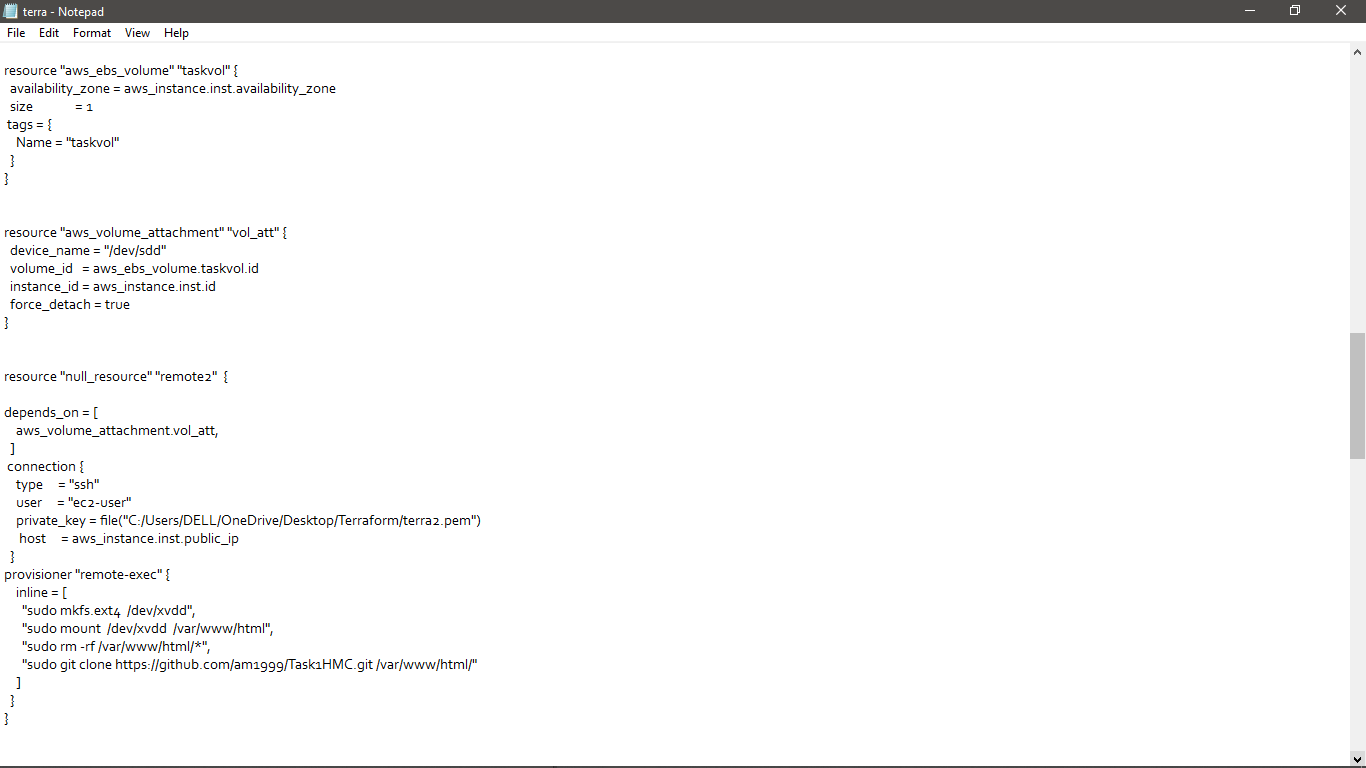
Step 3 : Launch Instance and ebs volume and attach ebs volume to Instance.

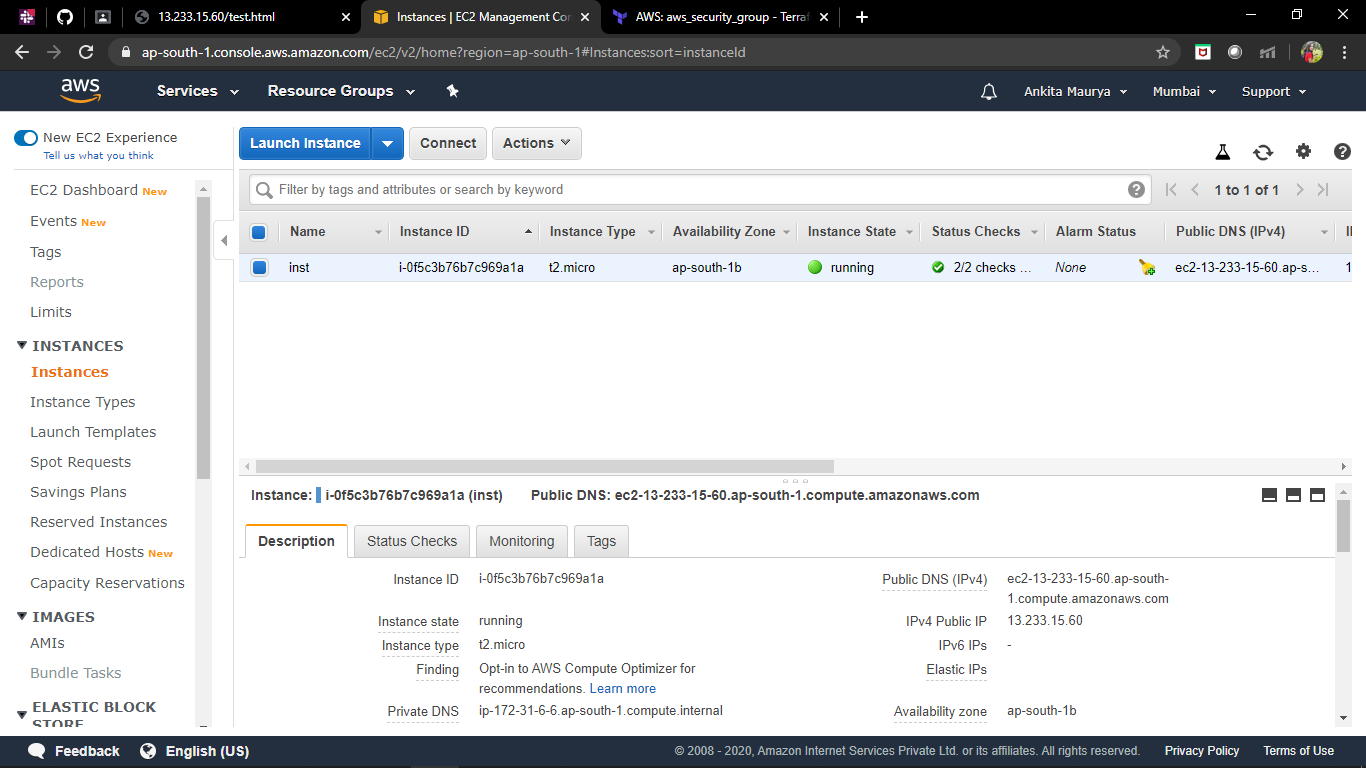
Provides an EC2 instance resource. This allows instances to be created, updated, and deleted. Instances also support [provisioning](https://www.terraform.io/docs/provisioners/index.html).

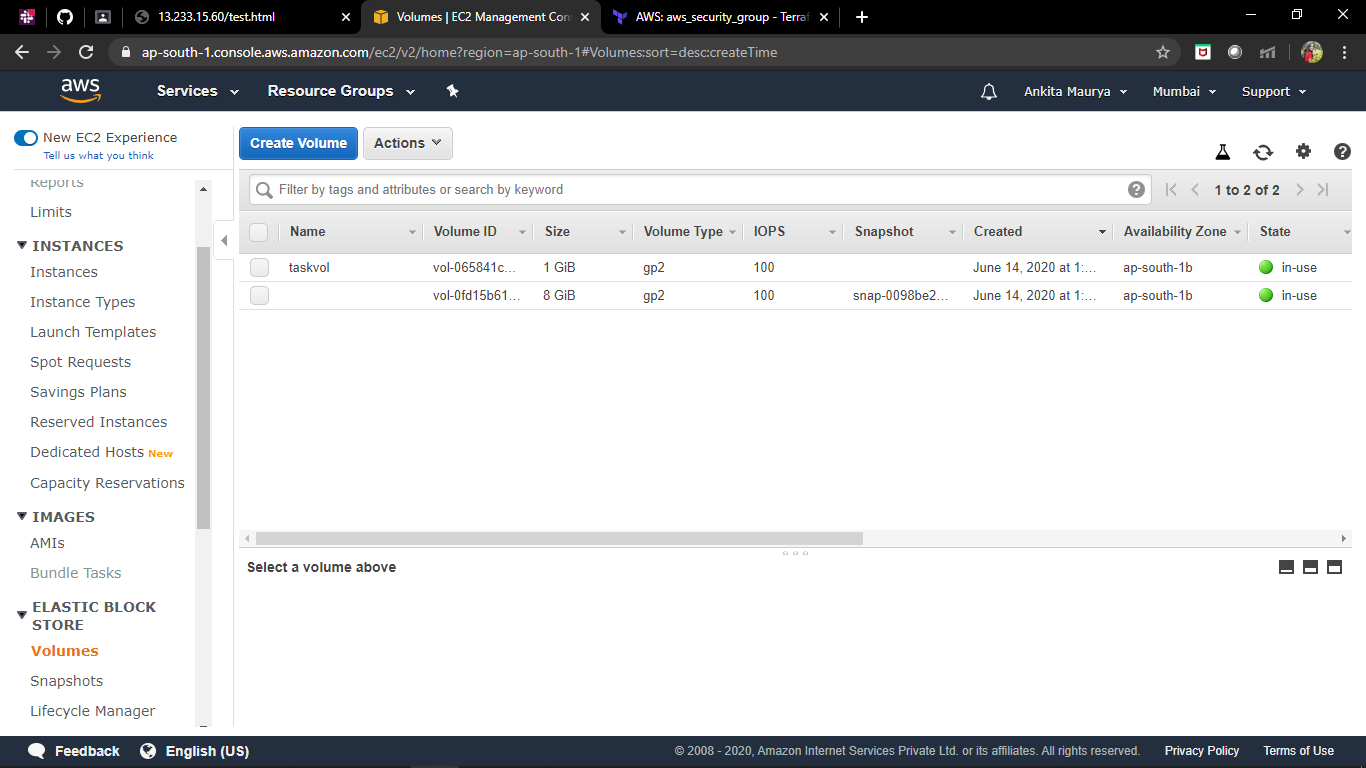
Manages a single EBS volume.

Establish connection and provisions.



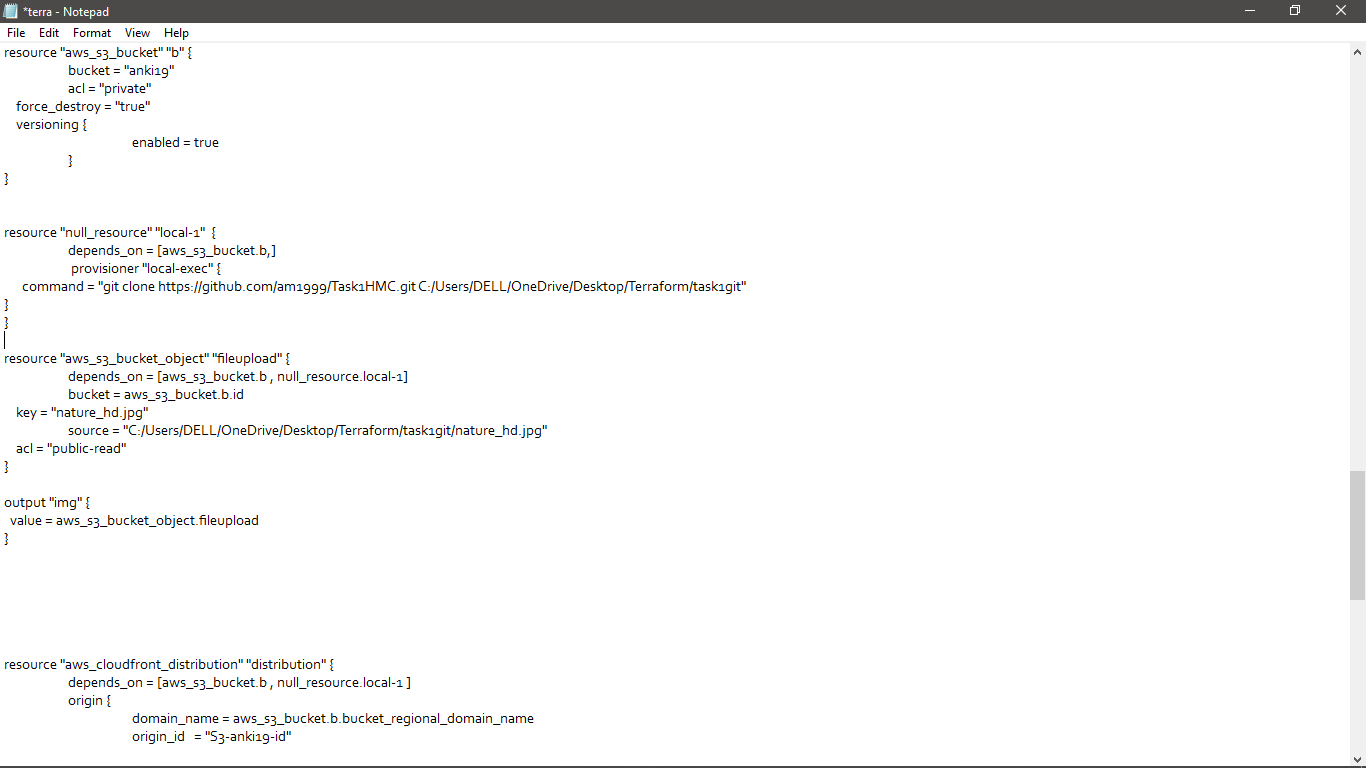


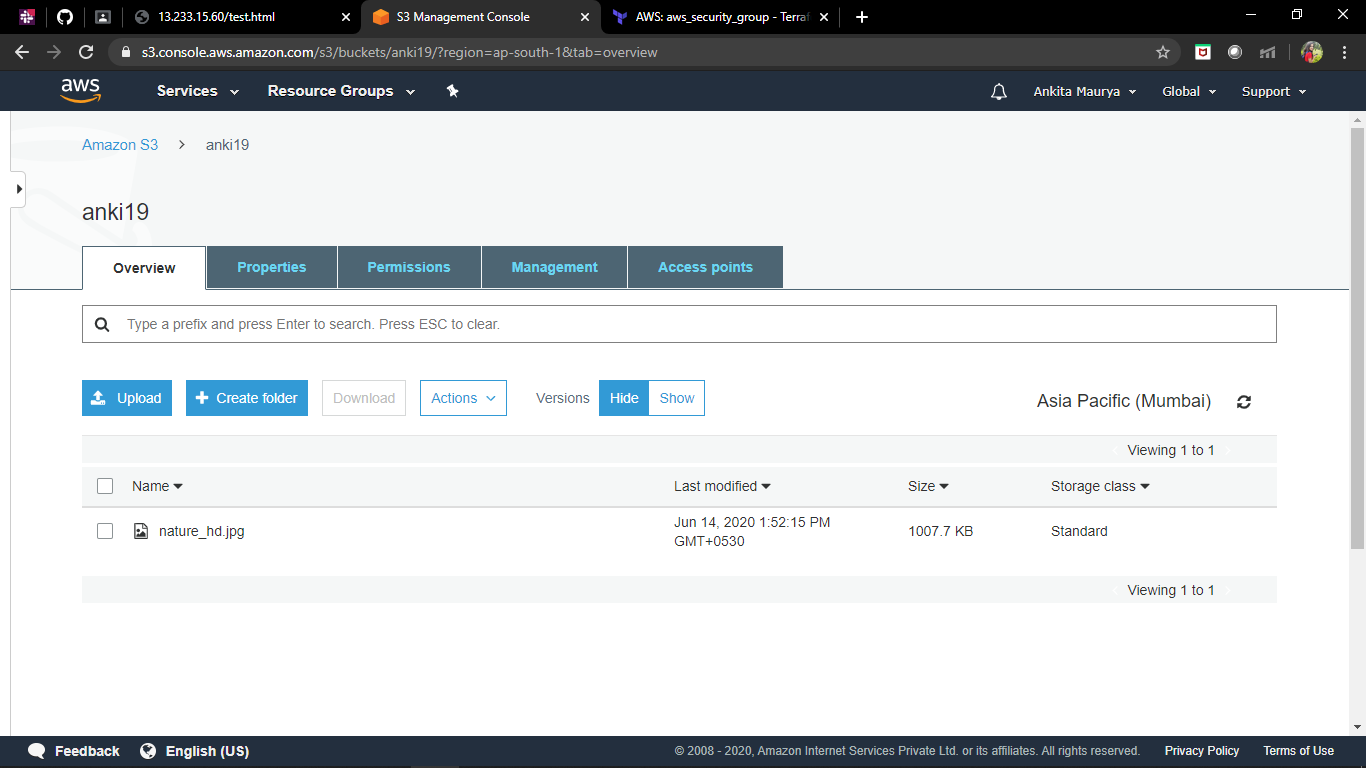




Step 4: Create S3 bucket, and copy/deploy the images from git hub repository into the s3 bucket and change the permission to public readable.

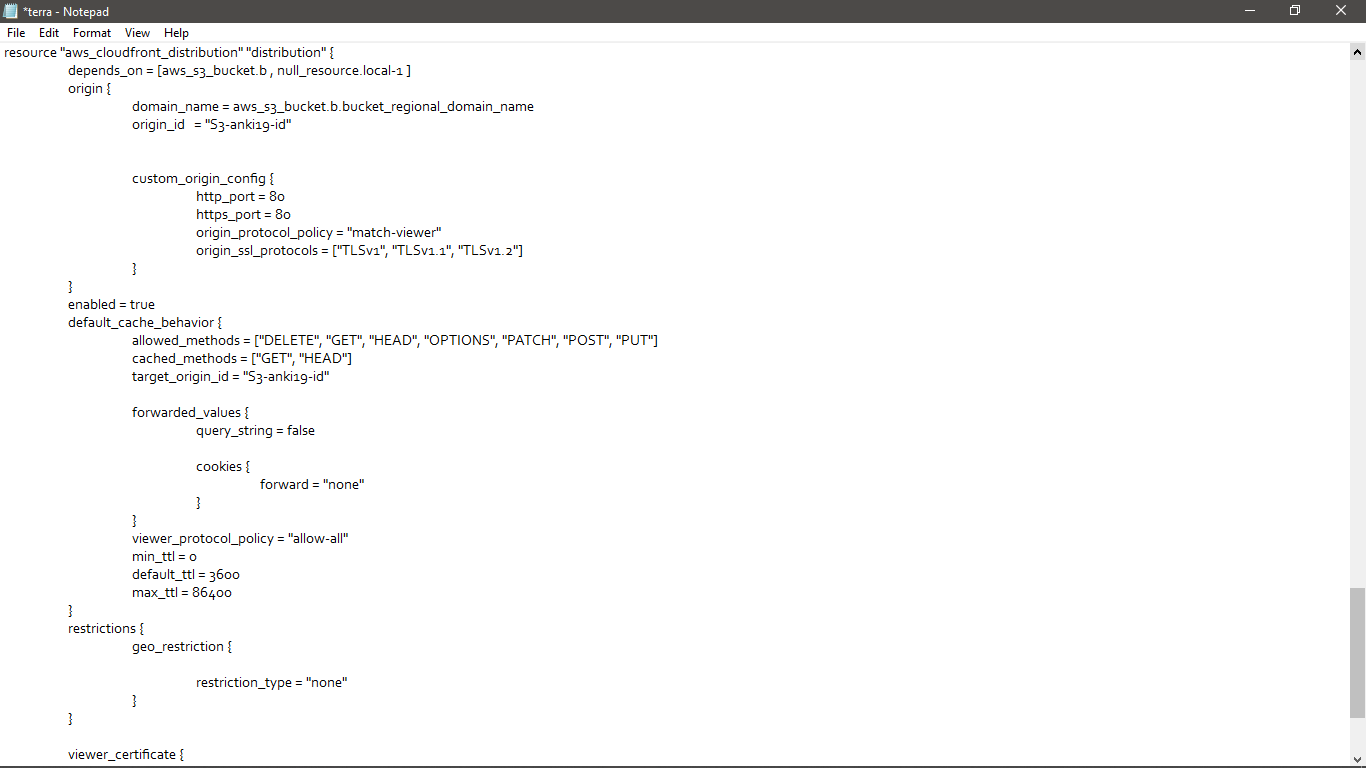
Provides a S3 bucket resource.

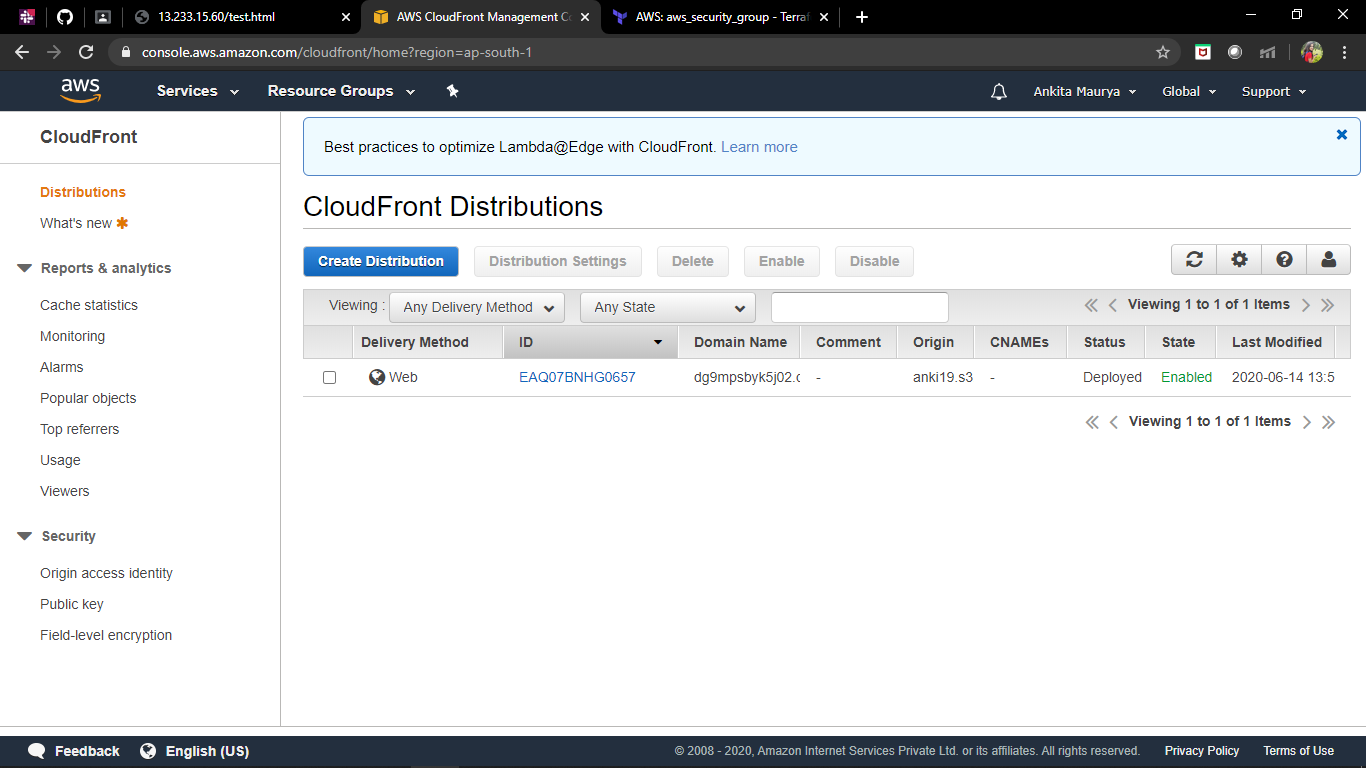




Step 5: Create a Cloudfront using s3 bucket (which contains images) and use the Cloudfront URL to update in code in /var/www/html.

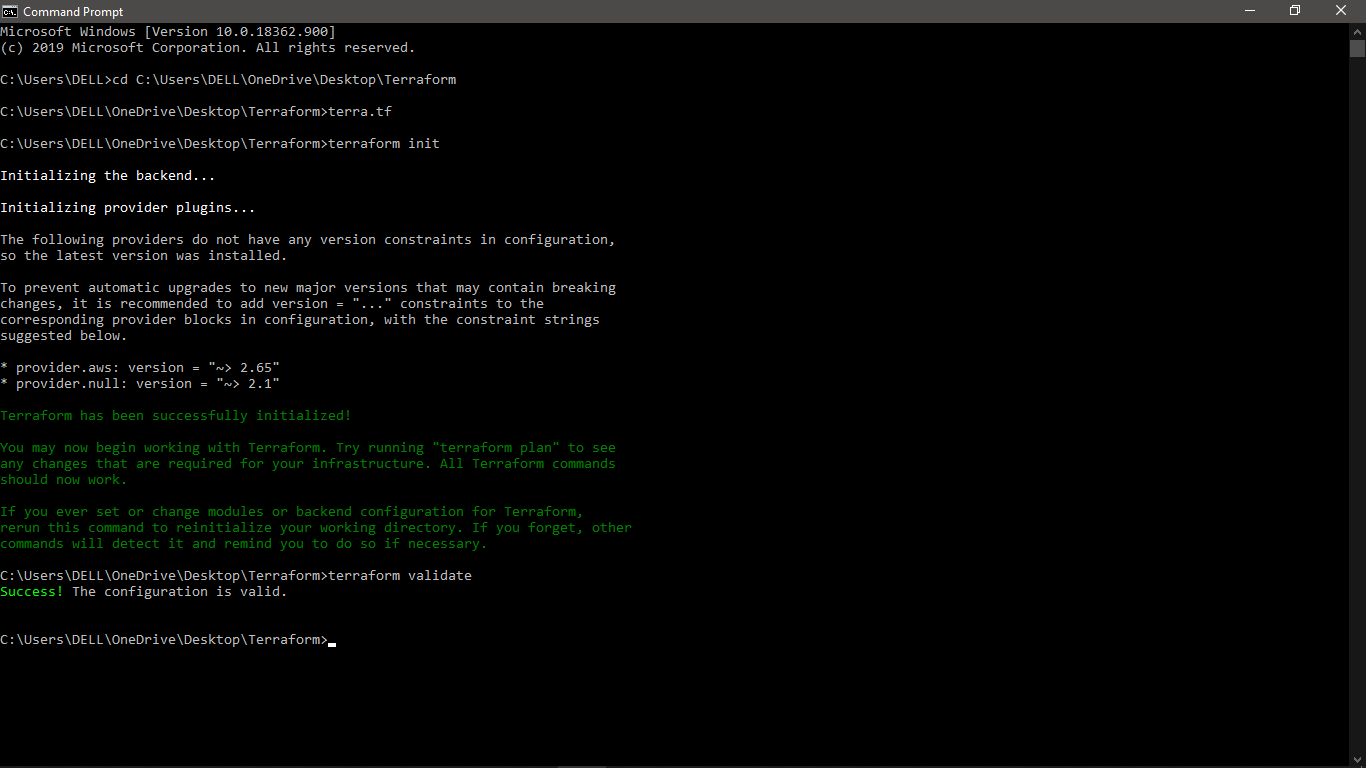
 CloudFront distributions take about 15 minutes to a deployed state after creation or modification.



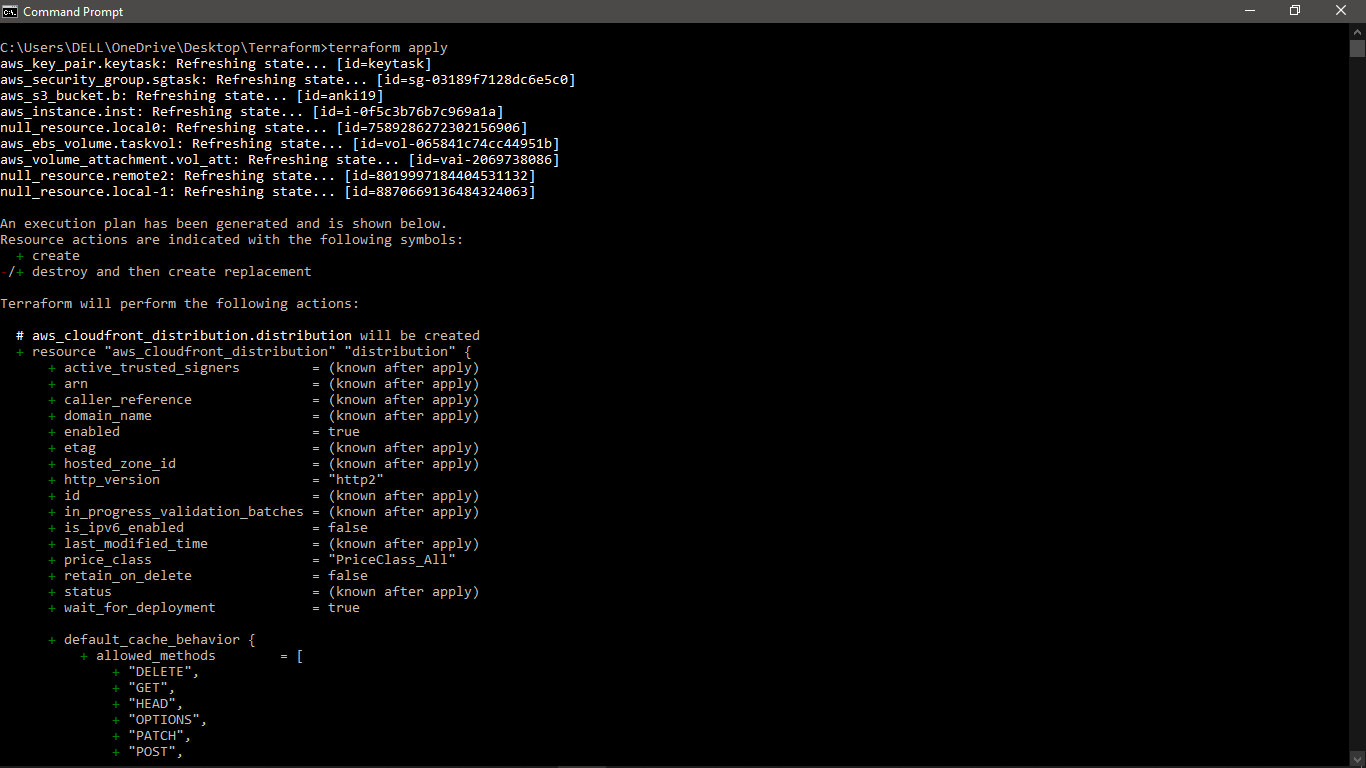


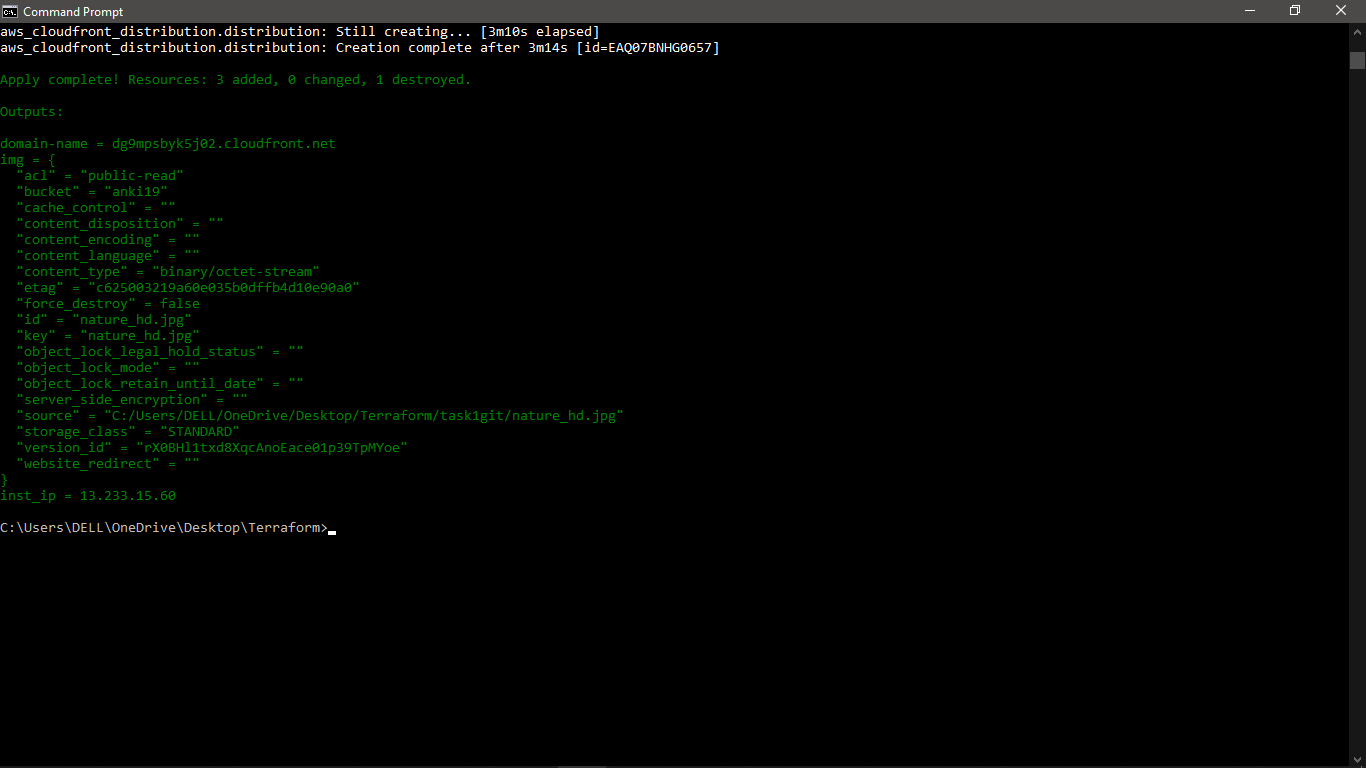
After writing the whole terraform code. Execute the following Commands:

1. terraform init
2. terraform validate
3. terraform apply



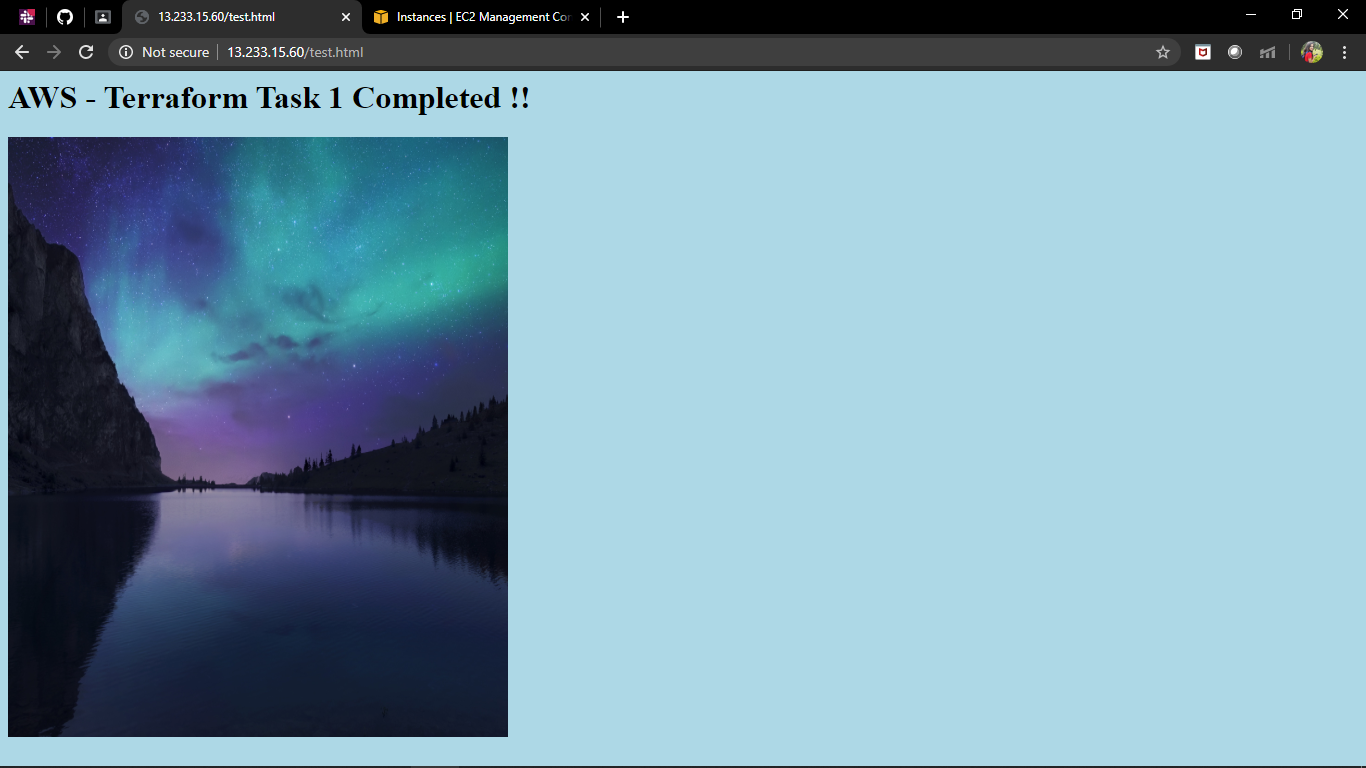
Hence, init and validation successfull!





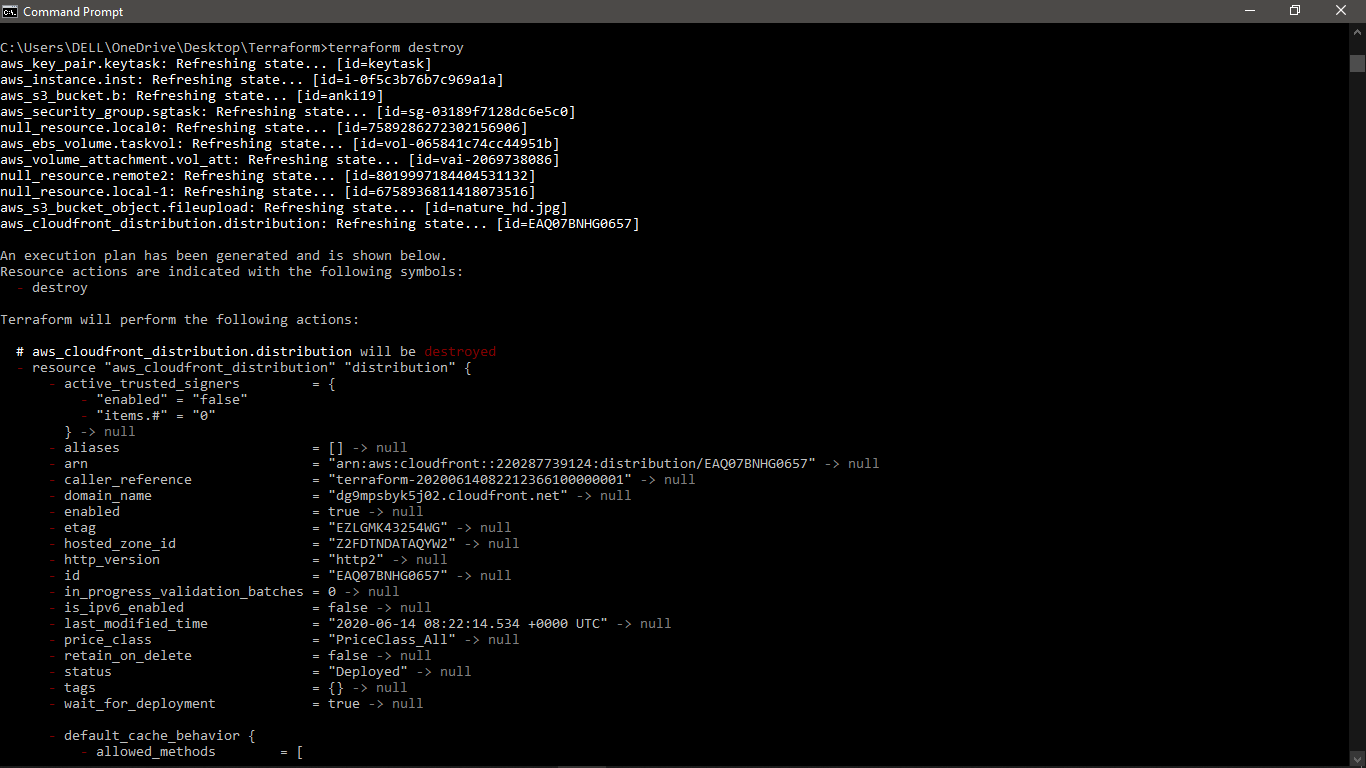
Hence, Apply Successful.

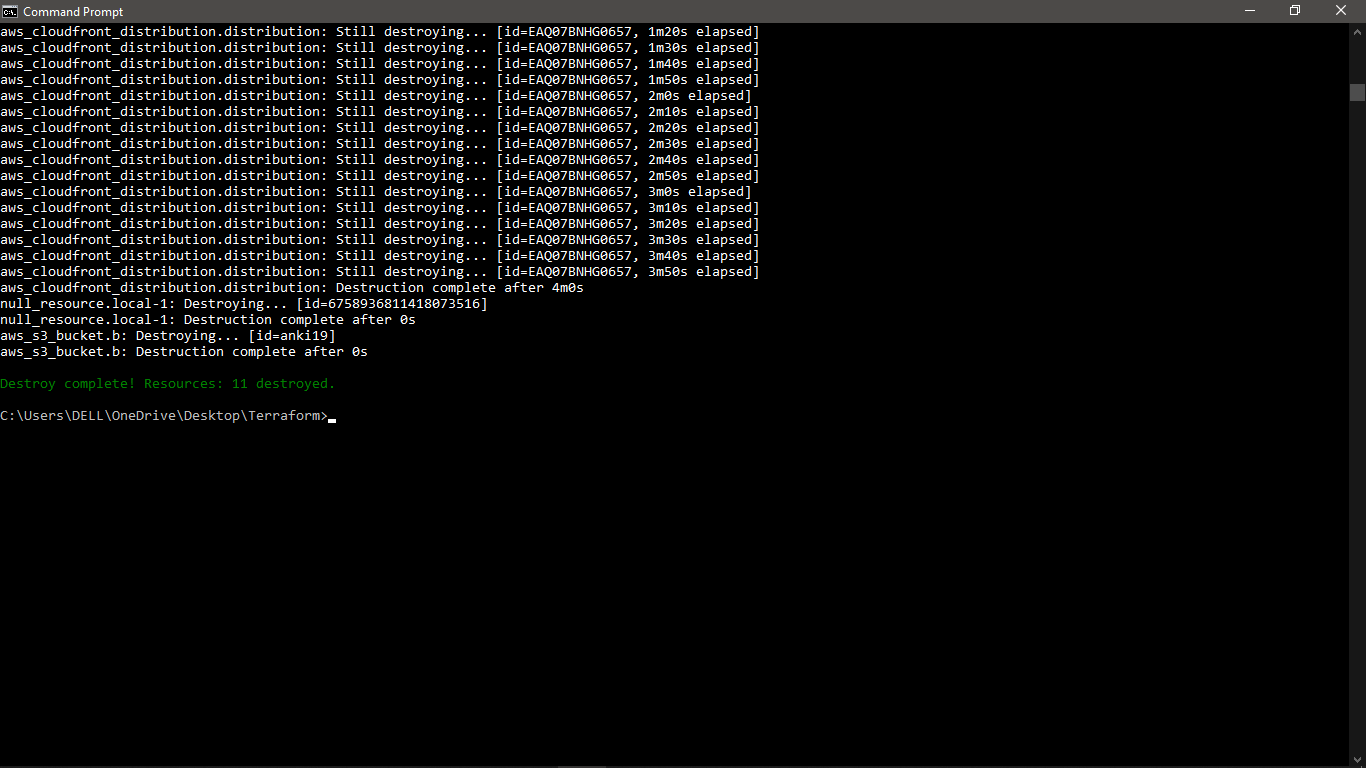
Hence, Task completed!!



Last Remove all the Infrastructure deployed through terraform code in one go.

Use: terraform destroy





Conclusion:

Deployed an AWS infrastructure through code using Terraform. Further using this code more such related infrastructure can be deployed in just one go in future. Terraform makes working with almost every cloud service provider feasible. Here, we have successfully written the terraform code and therefore deployed webpage/application using Terraform-AWS.