**Terraform**

Terraform is an open-source infrastructure as code software tool created by HashiCorp. It enables users to define and provision a datacenter infrastructure using a high-level configuration language known as Hashicorp Configuration Language (HCL), or optionally JSON.

**Need of Terraform**

Terraform is a really handy tech tool that lets you build, change, and version infrastructure safely and efficiently. It's specifically designed to support and manage the lifecycle of a wide range of resources, including physical servers, networking, and SaaS products.

Terraform gives an opportunity to multiple cloud (AZURE, GCP, AWS and Many More)

**Basic Command of Terraform**

**1 Terraform init**

The terraform init command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times.

**2 Terraform plan**

The terraform plan command is used to create an execution plan. Terraform performs a refresh, unless explicitly disabled, and then determines what actions are necessary to achieve the desired state specified in the configuration files.

**3 Terraform apply**

The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.

**4 Terraform destrory**

Resources can be destroyed using the terraform destroy command, which is similar to terraform apply but it behaves as if all of the resources have been removed from the configuration.

**TASK 1**

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

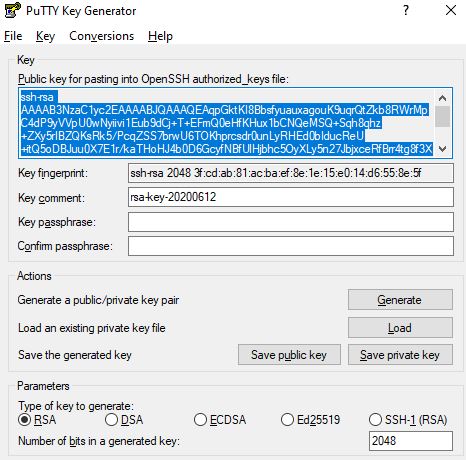
7. Create S3 bucket, and copy/deploy the images from github repo into the s3 bucket and change the permission to public readable.

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

(Here I am doing for windows)

How to create the key and security group which allow the port 80

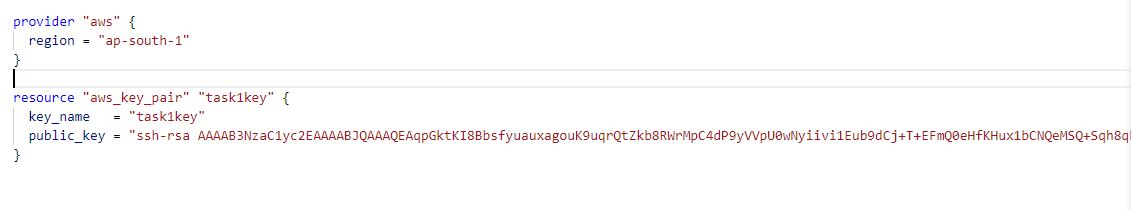
Start -> Puttygen->Generate(Drag your pointer for few seconds to create some randomness)



Click on “Save public key”.

Or Copy the key

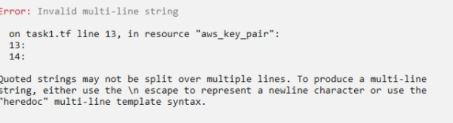
Open a new notepad file



Here “key\_name” , “tags” are optional.

“Public\_key” required

\*\*When you paste your key in notepad file make sure it should be paste in single line, otherwise you will get this error



Open CMD/PowerShell run “terraform apply”.

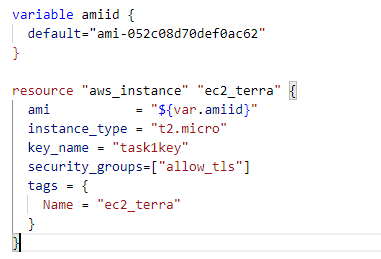
For creating security group gone through with this following



In networking, Ingress(In-Bound) & Egress(Out-Bound) can be considered as network boundary. All the network traffic filtered by these two components.

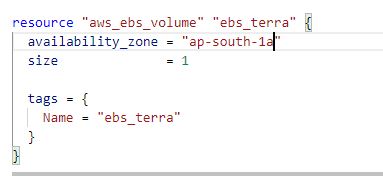
Open CMD/PowerShell run “terraform apply”.

 For launching EC2 instance, in this Ec2 instance we use the key and security group which we have created

.

Open CMD/PowerShell run “terraform apply”.

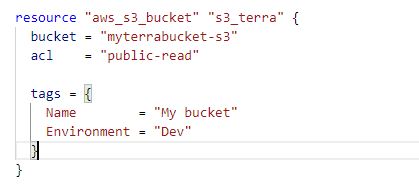
Launch one Volume (EBS) and mount that volume into /var/www/html



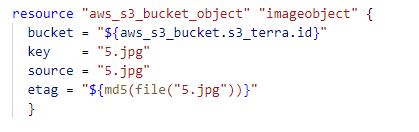


Open CMD/PowerShell run “terraform apply”.

S3 Bucket



Upload an image into a bucket



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



And Finally

