Deploy Static Website on CloudFront using Terraform

- We will provision the CloudFront for Static Website Delivery to users more quickly and securely.
- We will connect the CloudFront to ACM for secure web access.
- We will create S3 bucket to store the *static-website* files.
- We will also create a Route53 Hosted Zone and add a Route53 Record to access the website using the domain name.
- We will create all these resources using Terraform as an Infrastructure as Code.

Prerequisites

1. AWS Account with IAM User Access Keys
2. Terraform installed
3. Website repository
4. Domain name

Write Terraform Configuration files

First, we will write Terraform configuration files for AWS resources using predefined modules available on the internet.

Steps

1. Create the **cloudfront-website-terraform** project directory.
2. The folder structure for the above-created directory is as follows:

```
cloudfront-website-terraform
├──.terraform.lock.hcl
├──locals.tf
├──main.tf
├──providers.tf
├──terraform.tfstate
├──terraform.tfstate.backup
└──.terraform
```

> We need to only create *providers.tf*, *main.tf*, & *locals.tf* file. Other files are generated while initiating terraform.

3. Create a *providers.tf* file inside the above-created directory.
4. Inside it, define the following:
    - terraform
        - required_providers
    - provider

        - aws
5. Click code for reference.
6. The definition of *providers.tf* file is complete.
7. Now create the *main.tf* file.
8. Define the following modules inside it:
    - module.s3
    - module.acm-route53
    - module. cloudfront
    - module.route53-record
9. Click code for reference.
10. The definition of *main.tf* file is complete.
11. Now we will create *locals.tf* file.
12. Inside it, define the following variables:
    - local.aws-region
    - local.s3-properties
    - local.acm-properties
    - local.route53-zone-properties
    - local.cloudfront-properties
    - local.route53-record-properties
13. Click code for reference.
14. The definition of *locals.tf* file is complete.

> Ensure you give the appropriate values to the variables defined in *locals.tf* file.
> Also, ensure you set *aws-region* to **us-east-1**, as ACM will work with Cloudfront only & only if it is deployed in that region.

Provisioning the Infrastructure

Now we will provision the AWS infrastructure by applying the above-created Terraform configuration files.

> Ensure AWS CLI is configured with appropriate IAM User Access Keys with enough permissions.

Steps:

1. Open the PowerShell Window.
2. Change the directory to the above-created cloudfront-website-terraform directory using the `cd` command.
3. Run the `terraform fmt -recursive` command to format the syntax of the files.
4. Run the `terraform init` command to initialize the terraform.
5. Run the `terraform validate` command to validate the configuration files.
6. Run the `terraform plan` command to plan the resources to be created.
7. Run the `terraform apply` command and if prompted, type `yes` to provision the infrastructure.
8. Apply will take time, till then
    1. Login to the AWS console and search for the **Route-53** service.
    2. Click open the Route-53 console.
    3. In the left plane of the window, click on `Hosted zones`.

4. Terraform will create a new hosted zone for your domain e.g. `example.com`.
5. On completing, two records of type `NS` and `SOA` gets created here.
6. Copy all four values from column `Value/Route traffic to` of `NS` record.
7. Go to your domain provider's website and add these copied nameservers in place of your domain's original nameservers. This will dedicate your domain to AWS.
8. Now, head to the AWS Console and search for the **S3** service.
9. Click open the **S3** service, select the terraform provisioned bucket, and upload the static website files to it.
10. This will store all the website files and deliver them once the CloudFront requests.

9. Once the terraform provisioning is completed, try accessing the static website on the browser.

Destroy the provisioned infrastructure

Lastly, we will destroy the above-created resources.

Steps

1. To destroy infrastructure, open the Powershell Window and change the directory to the above-created **cloudfront-website-terraform** directory using the `cd` command.
2. Run `terraform destroy` & if prompted, type `yes`.
3. Infrastructure will be destroyed.
