

# TERRAFORM CODE DEPLOYMENT

This is a mini-project (task1) of “Hybrid Multi Cloud Computing” training which is assigned by our instructor Mr. Vimal Daga.

## AIM OF THE ASSIGNED TASK:-

To create an infrastructure on cloud (aws) by a piece of code using terraform to deploy my website. I have pushed my website code on github from where the code will download inside /var/www/html folder of my instance on cloud (aws).

The code will create the following services which will lessen the latency for deploying of code on client's system.

1. Created Security Group and Key which will provide inbound rule for port 80(webserver) and give authentication to our instance.
2. Then using this key and security group launched instance and used provisioner block to run program inside the instance i.e to install httpd, php and git etc.
3. And to create a persistent storage for our website code,i have created an EBS volume and mount it with /var/www/html
4. Then the code will download the github code and copy it in /var/www/html
5. Created S3 bucket, and copy/deploy the images (static data) from github repo into the s3 bucket and change the permission to public readable.
6. Then Created a Cloudfront (CDN) using s3 bucket(which contains Static Content of website) and use the Cloudfront URL to update in code in /var/www/html

## SOFTWARE / TOOLS REQUIRED:-

1. **Aws cli software:-** We are working on our cloud providers ([aws](#) in this project) using Terraform therefore we have to create a credential block called “profile” which contain the user/ pass in case of aws access-key / secret-key.

```
C:\Users\user\Desktop\teraform\mypro1>aws configure --profile akanksha
AWS Access Key ID [*****N5HD]:
AWS Secret Access Key [*****nJfE]:
Default region name [None]: ap-south-1
Default output format [None]:
```

2. **Terraform Software:-** We write terraform code in declarative language [HCL](#) (HashiCorp Configuration Language) in file having extension (.tf). Terraform is a tool which work on IAC (Infrastructure As Code). These codes could be referred as document of our infrastructure that could be share in a team to create the same infrastructure on Cloud at different levels like production, testing etc.

```
C:\Users\user>terraform -version
Terraform v0.12.26
```

3. **Git Software:-** We have created our repository using DVCS (git) at own system then push it on SCM (github) so that anyone could access it on internet and download the code from repository to deploy on our cloud instance.

```
MINGW64:/c/Users/user/Desktop
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop (master)
$ git version
git version 2.26.2.windows.1
```

### DVCS (GIT) REPOSITORY:-

Using GIT I have created a repository of my website code and content files. So that the code could be manageable and if any change occur in code then git will track the changes. Firstly, I have initialized the workspace to convert it into repository.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git init
Initialized empty Git repository in C:/Users/user/Desktop/Hybrid_task_1/webpagecode/.git/
```

After that if we see the status of repository then we could see all the untrack file.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ ls
assets/  css/  fonts/  index.html  js/  OwlCarousel2-2.3.4/  webfonts/

user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    OwlCarousel2-2.3.4/
    assets/
    css/
    fonts/
    index.html
    js/
    webfonts/

nothing added to commit but untracked files present (use "git add" to track)
```

Then using the following command I have added the code file and contents inside the repository staging area from where git will keep on tracking my files.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git add .
warning: LF will be replaced by CRLF in css/all.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/owl.carousel.min.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/owl.theme.default.min.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/owl.carousel.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in webfonts/fa-brands-400.svg.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in webfonts/fa-regular-400.svg.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in webfonts/fa-solid-900.svg.
The file will have its original line endings in your working directory

user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   OwlCarousel2-2.3.4/.gitattributes
    new file:   OwlCarousel2-2.3.4/.gitignore
```

Commit the files giving appropriate messages along which will create a backup of files.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ ls -a
./      .git/   css/    index.html  OwlCarousel2-2.3.4/
../     assets/ fonts/  js/         webfonts/

user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git commit index.html -m "my photography website: Valley Times"
[master (root-commit) 888fafa] my photography website: Valley Times
1 file changed, 171 insertions(+)
create mode 100644 index.html

user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git commit css -m "css code for beautiful background and theme"
warning: LF will be replaced by CRLF in css/all.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/owl.carousel.min.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/owl.theme.default.min.css.
The file will have its original line endings in your working directory
[master c9a35b0] css code for beautiful background and theme
5 files changed, 4821 insertions(+)
create mode 100644 css/all.css
create mode 100644 css/fonts.css
```

The logs will describe the commit id of master node. With the author and date of commit. Through this rolling back of files to this backup point could be done.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git status
On branch master
nothing to commit, working tree clean

user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git log
commit 77454f3c9ddd8afa4237fde57833194e7439358c (HEAD -> master)
Author: Uni-wv <akanksha.6698@gmail.com>
Date: Sat Jun 13 23:01:32 2020 +0530

    web fonts which i have used in title and heading

commit b35151c21f1ed2bdbb12a2174d0b798a9014ac98
Author: Uni-wv <akanksha.6698@gmail.com>
Date: Sat Jun 13 23:00:11 2020 +0530

    javascript code for dynamic webpage

commit 6ed5df88a2c1beba1a580d8fc9a6745e9ac23b82
Author: Uni-wv <akanksha.6698@gmail.com>
Date: Sat Jun 13 22:58:53 2020 +0530

    For fonts of different shapes and sizes

commit 66b250f88620befc507ab966fdff76df1b2cc38b
Author: Uni-wv <akanksha.6698@gmail.com>
Date: Sat Jun 13 22:58:03 2020 +0530

    my photograph folder

commit 2f3dd528143b525257dbc8b14df23c5e335e9477
Author: Uni-wv <akanksha.6698@gmail.com>
```

Now create a remote url with a reference name generally we use “origin”.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git remote -v

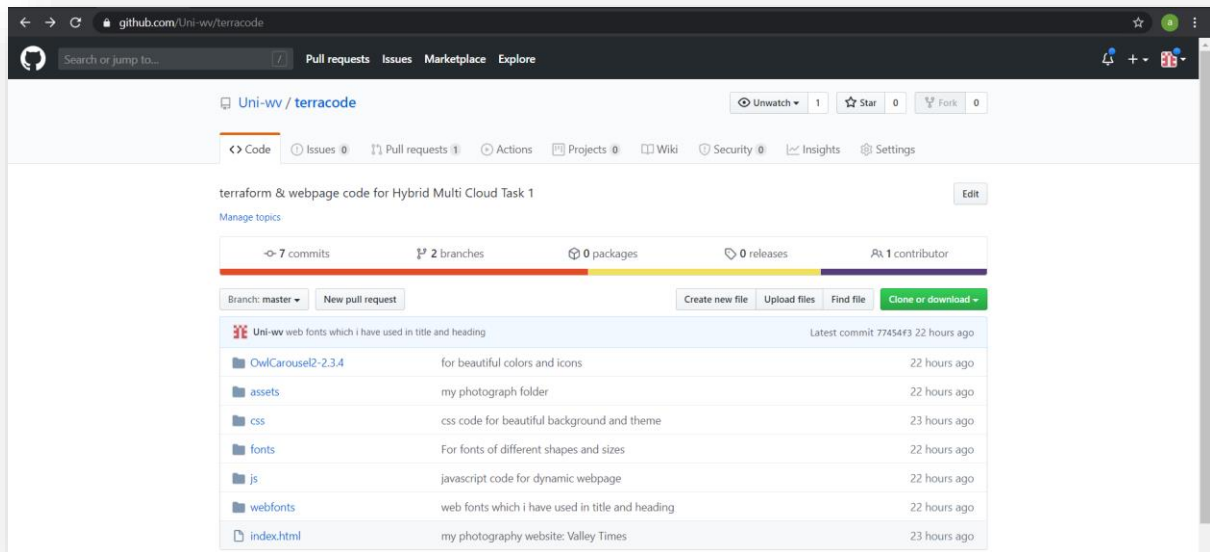
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git remote add origin https://github.com/Uni-wv/terrancode.git

user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git remote -v
origin https://github.com/Uni-wv/terrancode.git (fetch)
origin https://github.com/Uni-wv/terrancode.git (push)
```

Now I push the code to github repository using the following command.

```
user@DESKTOP-4Q0TPOI MINGW64 ~/Desktop/Hybrid_task_1/webpagecode (master)
$ git push -u origin master
Enumerating objects: 475, done.
Counting objects: 100% (475/475), done.
Delta compression using up to 4 threads
Compressing objects: 100% (470/470), done.
Writing objects: 65% (309/475), 233.32 MiB | 758.00 KiB/s
Writing objects: 100% (475/475), 415.37 MiB | 365.00 KiB/s, done.
Total 475 (delta 57), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (57/57), done.
To https://github.com/Uni-wv/terrancode.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

You may see the website code on github (<https://github.com/Uni-wv/terrancode.git>) .



I have also push the Terraform code on GitHub to share with my connections so that you all could create the same infrastructure and checkout my project by deploying the website on it.

```
C:\Users\user\Desktop\teraform\mypro1>git init
Initialized empty Git repository in C:/Users/user/Desktop/teraform/mypro1/.git/

C:\Users\user\Desktop\teraform\mypro1>git add .
warning: LF will be replaced by CRLF in .terraform/plugins/windows_amd64/lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in terraform.tfstate.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in terraform.tfstate.backup.
The file will have its original line endings in your working directory

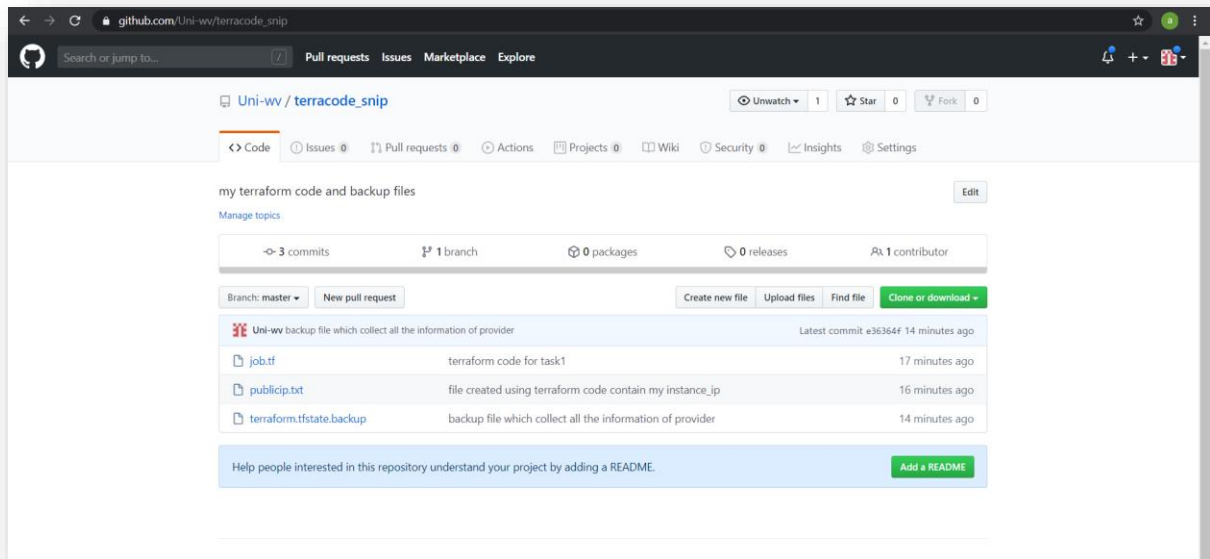
C:\Users\user\Desktop\teraform\mypro1>git commit job.tf -m "terraform code for task1"
[master (root-commit) d6bbdea] terraform code for task1
1 file changed, 211 insertions(+)
create mode 100644 job.tf
```

Similarly, I have created a remote url name and push the repository using follwing command.

```
C:\Users\user\Desktop\teraform\mypro1>git remote -v
origin https://github.com/Uni-wv/terrancode_snip.git (fetch)
origin https://github.com/Uni-wv/terrancode_snip.git (push)

C:\Users\user\Desktop\teraform\mypro1>git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 4.78 KiB | 489.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Uni-wv/terrancode_snip.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

You may see the code on GitHub ([https://github.com/Uni-wv/terraform\\_snip.git](https://github.com/Uni-wv/terraform_snip.git))



## TERRAFORM:-

Terraform which is a product of HashiCorp which work on Infrastructure As Code (IAC) i.e. terraform code made the infrastructure As A Service (IAAS) provider e.g.- AWS as a code. The benefits of Terraform are:-

- Terraform saves the information about the service i.e id,ami,ip etc. inside **terraform.tfstate** file.
- Automatically connect to respected service API by using plugins.
- Could create or destroy the infrastructure from one command using the code. > terraform destroy
- One Could Share code easily.

## TERRAFORM CODE SNIPPET:-



Above is VPC (Virtual Private Cloud) of my Amazon Web Services which I will use in the code to provide Networking As Service (NAS) to the resources / instances.

In below code I have written code block to login into provider “aws” and some inbound rules for security groups / firewall permissions.

```
1 provider "aws" {
2     region = "ap-south-1"
3     profile = "akanksha"
4 }
5
6 //security-group permissions
7 resource "aws_security_group" "mysg_allow_ssh_http" {
8     name = "aws_security_group"
9     description = "gives ssh & http permissions to client"
10    vpc_id = "vpc-79978a11"
11
12    ingress{
13        from_port = 80
14        to_port = 80
15        protocol = "tcp"
16        cidr_blocks = ["0.0.0.0/0"]
17    }
18
19    ingress{
20        from_port = 22
21        to_port = 22
22        protocol = "tcp"
23        cidr_blocks = ["0.0.0.0/0"]
24    }
25
26    egress{
27        from_port = 0
28        to_port = 0
29        protocol = "-1"
30        cidr_blocks = ["0.0.0.0/0"]
31    }
32
33    tags = {
34        Name = "mysg_allow_ssh_http"
35    }
36 }
37
```

Follwing blocks will create instance and ebs volume for persistent storage of data.

```
39 //creating instance
40 resource "aws_instance" "myinstance" {
41     ami = "ami-052c08d70def0ac62"
42     instance_type = "t2.micro"
43     key_name = "key123"
44     availability_zone = "ap-south-1a"
45     security_groups = [ aws_security_group.mysg_allow_ssh_http.name ]
46
47     connection {
48         type = "ssh"
49         user = "ec2-user"
50         private_key = file("C:/Users/user/Downloads/key123.pem")
51         host = aws_instance.myinstance.public_ip
52     }
53
54     provisioner "remote-exec" {
55         inline = [
56             "sudo yum install httpd php git -y" ,
57             "sudo systemctl restart httpd" ,
58             "sudo systemctl enable httpd" ,
59         ]
60     }
61
62     tags = {
63         Name = "myinstance"
64     }
65 }
66
67 //creating volume
68 resource "aws_ebs_volume" "ebs_volume" {
69     availability_zone = aws_instance.myinstance.availability_zone
70     size = 1
71     tags = {
72         Name = "ebs_volume"
73     }
74 }
```



Attaching the ebs volume to instance. Recommended :- not to use force\_detach.

```
75 //attaching volume
76 resource "aws_volume_attachment" "ebs_attach" {
77   device_name = "/dev/sdh"
78   volume_id   = aws_ebs_volume.ebs_volume.id
79   instance_id = aws_instance.myinstance.id
80   force_detach = true
81 }
82
83
84 //printing ip
85 output "myinstance_ip" {
86   value = aws_instance.myinstance.public_ip
87 }
88 //printing port
89 output "web_port" {
90   value = 80
91 }
92 //storing ip in a file
93 resource "null_resource" "nulllocal" {
94   provisioner "local-exec" {
95     command = "echo ${aws_instance.myinstance.public_ip} > publicip.txt"
96   }
97 }
98
99 resource "null_resource" "nullremote" {
100
101   depends_on = [
102     aws_volume_attachment.ebs_attach,
103   ]
104
105   connection {
106     type = "ssh"
107     user = "ec2-user"
108     private_key = file("C:/Users/user/Downloads/key123.pem")
109     host = aws_instance.myinstance.public_ip
110   }
111 }
```

Now after attaching volume create partition and then format and mount it with folder then store the GitHub code into it.

```
97 }
98 //to prioritise the terraform creation
99 resource "null_resource" "nullremote" {
100
101   depends_on = [
102     aws_volume_attachment.ebs_attach,
103   ]
104
105   connection {
106     type = "ssh"
107     user = "ec2-user"
108     private_key = file("C:/Users/user/Downloads/key123.pem")
109     host = aws_instance.myinstance.public_ip
110   }
111 }
112
113 provisioner "remote-exec" {
114   inline = [
115     "sudo mkfs.ext4 /dev/xvdb",
116     "sudo mount /dev/xvdb /var/www/html",
117     "sudo rm -rf /var/www/html/*",
118     "sudo git clone https://github.com/Uni-wv/terracode.git /var/www/html/" ,
119   ]
120 }
121 }
122 }
```



The code block would create a s3 bucket in which static content of website could be put like images, videos etc. And give public readable permissions through s3 policy.

```
123 //to create s3 bucket
124 resource "aws_s3_bucket" "test123akadb" {
125     bucket = "test123akadb"
126     acl = "private"
127     region = "ap-south-1"
128
129     tags = {
130         Name = "test123akadb"
131     }
132 }
133
134 locals {
135     s3_origin_id = "myS3Origin"
136 }
137
138 //to upload object to bucket
139 resource "aws_s3_bucket_object" "object" {
140     depends_on = [
141         aws_s3_bucket.test123akadb,
142     ]
143     bucket = "test123akadb"
144     key = "my_bucket"
145     source = "C:/Users/user/Desktop/html page/website/assets/web5.jpg"
146 }
147
148 //to give policy of bucket
149 resource "aws_s3_bucket_public_access_block" "public_access" {
150     depends_on = [
151         aws_s3_bucket.test123akadb,
152     ]
153     bucket = "test123akadb"
154     block_public_acls = false
155     block_public_policy = false
156 }
```

Distribute the content through CDN edges to lessen the latency of accessing the content

```
155 }
156 //cloudfront distribution
157 resource "aws_cloudfront_distribution" "cloud_distribute" {
158     origin {
159         domain_name = aws_s3_bucket.test123akadb.bucket_regional_domain_name
160         origin_id = local.s3_origin_id
161     }
162     enabled = true
163     is_ipv6_enabled = true
164     comment = "my cloud front access"
165     default_root_object = "index.html"
166     default_cache_behavior {
167         allowed_methods = ["DELETE", "PATCH", "OPTIONS", "POST", "PUT", "GET", "HEAD"]
168         cached_methods = ["GET", "HEAD"]
169         target_origin_id = local.s3_origin_id
170
171         forwarded_values {
172             query_string = false
173             cookies {
174                 forward = "none"
175             }
176         }
177         min_ttl = 0
178         default_ttl = 3600
179         max_ttl = 86400
180         compress = true
181         viewer_protocol_policy = "redirect-to-https"
182     }
183
184     restrictions {
185         geo_restriction {
186             restriction_type = "whitelist"
187             locations = ["US"]
188         }
189     }
190
191     tags = {
192         Environment = "production"
193     }
194 }
```

```

194
195     viewer_certificate {
196         cloudfront_default_certificate = true
197     }
198 }
199
200 resource "null_resource" "nullocal1" {
201
202
203     depends_on = [
204         null_resource.nullremote,
205     ]
206
207     provisioner "local-exec" {
208         command = "curl ${aws_instance.myinstance.public_ip}"
209     }
210 }
211

```

Now to will validate and initialize our terraform file which will give code accuracy and download respective plugins for providers (in my case aws).

```

C:\Users\user\Desktop\teraform\mypro1>terraform validate
Success! The configuration is valid.

C:\Users\user\Desktop\teraform\mypro1>terraform init

Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.65"
* provider.null: version = "~> 2.1"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

```

Plugins it download behind the seen for aws provider

```

C:\Users\user\Desktop\teraform\mypro1>cd .terraform
C:\Users\user\Desktop\teraform\mypro1\.terraform>dir
Volume in drive C is OS
Volume Serial Number is A870-F449

Directory of C:\Users\user\Desktop\teraform\mypro1\.terraform

11-06-2020  06:37    <DIR>        .
11-06-2020  06:37    <DIR>        ..
11-06-2020  06:37    <DIR>        plugins
               0 File(s)                0 bytes
               3 Dir(s)  147,627,638,784 bytes free

```

The plugins executable files are stored inside windows\_amd64 folder (windows OS).

```
Command Prompt
C:\Users\user\Desktop\teraform\mypro1\.terraform\plugins>dir
Volume in drive C is OS
Volume Serial Number is A870-F449

Directory of C:\Users\user\Desktop\teraform\mypro1\.terraform\plugins

11-06-2020 06:37 <DIR>      .
11-06-2020 06:37 <DIR>      ..
14-06-2020 08:43 <DIR>      windows_amd64
                0 File(s)      0 bytes
                3 Dir(s)  147,627,569,152 bytes free

C:\Users\user\Desktop\teraform\mypro1\.terraform\plugins>cd windows_amd64
C:\Users\user\Desktop\teraform\mypro1\.terraform\plugins\windows_amd64>dir
Volume in drive C is OS
Volume Serial Number is A870-F449

Directory of C:\Users\user\Desktop\teraform\mypro1\.terraform\plugins\windows_amd64

14-06-2020 08:43 <DIR>      .
14-06-2020 08:43 <DIR>      ..
14-06-2020 11:21             157 lock.json
11-06-2020 06:37      156,011,008 terraform-provider-aws_v2.65.0_x4.exe
14-06-2020 08:43      20,375,040 terraform-provider-null_v2.1.2_x4.exe
                3 File(s)    176,386,205 bytes
                2 Dir(s)  147,627,569,152 bytes free

C:\Users\user\Desktop\teraform\mypro1\.terraform\plugins\windows_amd64>notepad lock.json
```

To run the Terraform code we will use command -> terraform apply -auto-approve #to run the code without any interruption of asking any questions to the user.

```
C:\Users\user\Desktop\teraform\mypro1>terraform apply -auto-approve
aws_security_group.mysg_allow_ssh_http: Refreshing state... [id=sg-021cc87f36bf8ce21]
aws_instance.myinstance: Refreshing state... [id=i-01585ea18a08e30ed]
null_resource.nulllocal: Refreshing state... [id=6746408244146925452]
aws_ebs_volume.ebs_volume: Refreshing state... [id=vol-0d6861909aae47570]
aws_volume_attachment.ebs_attach: Refreshing state... [id=vai-2892169574]
null_resource.nullremote: Refreshing state... [id=4505649951120701549]
null_resource.nulllocal1: Creating...
aws_s3_bucket.test123akadb: Creating...
null_resource.nulllocal1: Provisioning with 'local-exec'...
null_resource.nulllocal1 (local-exec): Executing: [cmd " /c "curl 13.235.132.32"]
null_resource.nulllocal1 (local-exec): % Total % Received % Xferd Average Speed Time Time Time Current
null_resource.nulllocal1 (local-exec): Dload Upload Total Spent Left Speed
null_resource.nulllocal1 (local-exec): 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0
null_resource.nulllocal1 (local-exec): 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0
null_resource.nulllocal1 (local-exec): 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0
null_resource.nulllocal1 (local-exec): 100 3985 100 3985 0 0 1992 0 0:00:02 0:00:02 0:00:00 1545
null_resource.nulllocal1 (local-exec): <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/D
TD/xhtml11.dtd">
null_resource.nulllocal1 (local-exec): <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
null_resource.nulllocal1 (local-exec): <head>
null_resource.nulllocal1 (local-exec): <title>Test Page for the Apache HTTP Server on Red Hat Enterprise Linux<
/title>
null_resource.nulllocal1 (local-exec): <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

In my case I have added some code for cloudFront Distribution later to the terraform code file (job.tf) and when I run the code again. I was amazed that it ran so quick. Terraform check the services already present and only created the newer one.

```

C:\Users\user\Desktop\teraform\mypro1>terraform apply -auto-approve
aws_security_group.mysg_allow_ssh_http: Refreshing state... [id=sg-021cc87f36bf8ce21]
aws_s3_bucket.test123akadb: Refreshing state... [id=test123akadb]
aws_instance.myinstance: Refreshing state... [id=i-01585ea18a08e30ed]
aws_s3_bucket_public_access_block.public_access: Refreshing state... [id=test123akadb]
aws_s3_bucket_object.object: Refreshing state... [id=my_bucket]
aws_ebs_volume.ebs_volume: Refreshing state... [id=vol-0d6861909aae47570]
null_resource.nulllocal: Refreshing state... [id=6746408244146925452]
aws_volume_attachment.ebs_attach: Refreshing state... [id=vai-2892169574]
null_resource.nullremote: Refreshing state... [id=4505649951120701549]
null_resource.nulllocal1: Refreshing state... [id=2776803249537781145]
aws_cloudfront_distribution.cloud_distribute: Creating...
aws_cloudfront_distribution.cloud_distribute: Still creating... [10s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [20s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [30s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [40s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [50s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [1m0s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [1m10s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [1m20s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [1m30s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [1m40s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [1m50s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [2m0s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [2m10s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [2m20s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [2m30s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [2m40s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [2m50s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [3m0s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [3m10s elapsed]
aws_cloudfront_distribution.cloud_distribute: Still creating... [3m20s elapsed]
aws_cloudfront_distribution.cloud_distribute: Creation complete after 3m24s [id=E23ZR954V2Q5GV]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
myinstance_ip = 13.235.132.32
web_port = 80

```

After running the code terraform automatically creates a documentary (statefile) collecting information about all the service's credentials like ip, rules etc. The file has default name terraform.tfstate from which terraform match the service exists or not if not exist then create and if already exist then update the services with new rule.

```

C:\Users\user\Desktop\teraform\mypro1>dir
Volume in drive C is OS
Volume Serial Number is A870-F449

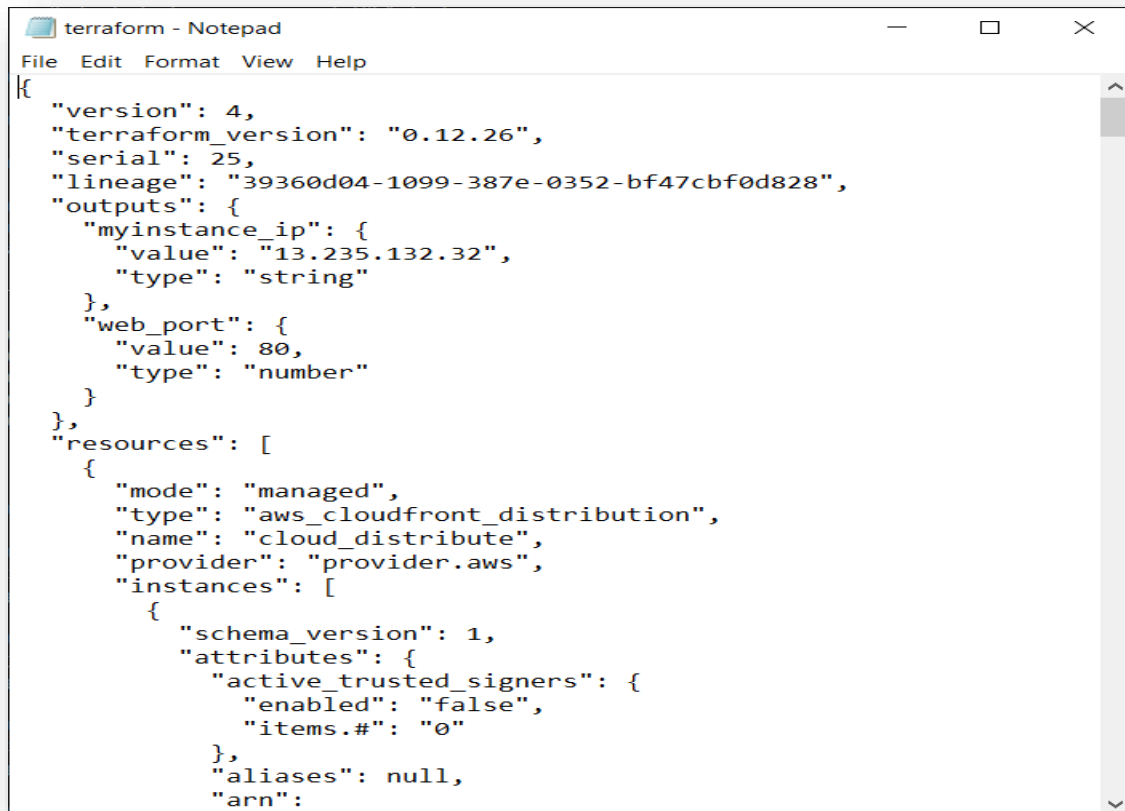
Directory of C:\Users\user\Desktop\teraform\mypro1

14-06-2020  11:29    <DIR>          .
14-06-2020  11:29    <DIR>          ..
11-06-2020  06:37    <DIR>          .terraform
14-06-2020  11:46             4,373  job.tf
14-06-2020  11:17             17  publicip.txt
14-06-2020  11:29            16,484  terraform.tfstate
14-06-2020  11:29            12,598  terraform.tfstate.backup
               4 File(s)          33,472 bytes
               3 Dir(s)  149,858,639,872 bytes free

C:\Users\user\Desktop\teraform\mypro1>

```

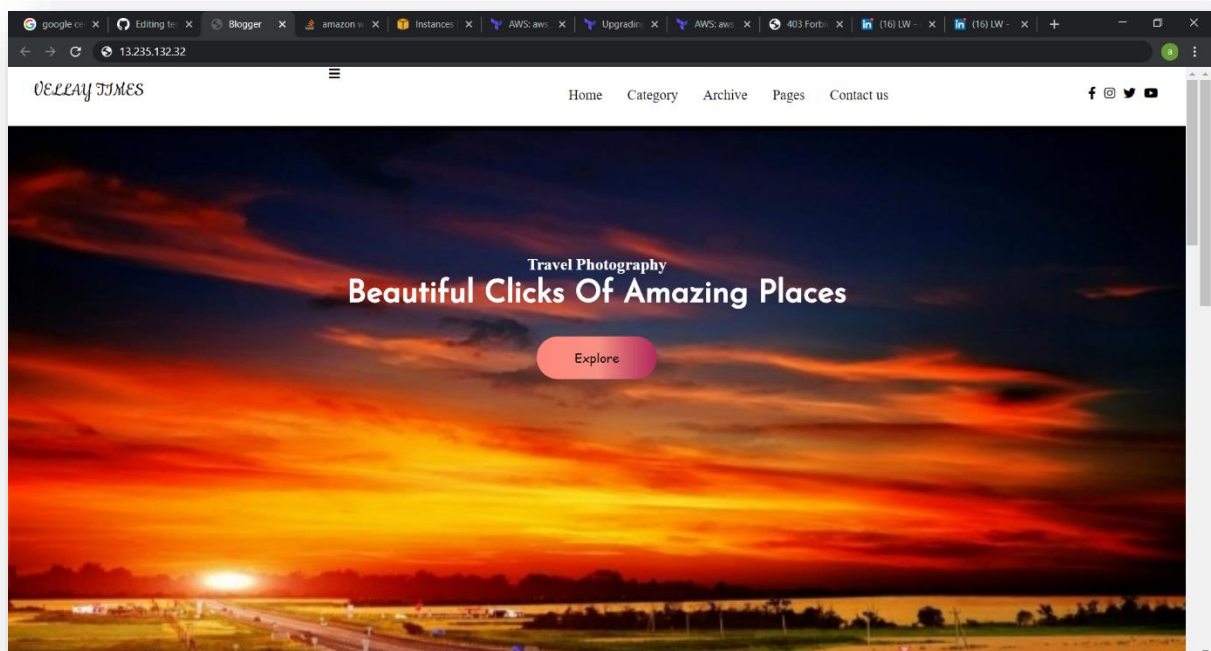
Following is the state file terraform created in my case.



```
{
  "version": 4,
  "terraform_version": "0.12.26",
  "serial": 25,
  "lineage": "39360d04-1099-387e-0352-bf47cbf0d828",
  "outputs": {
    "myinstance_ip": {
      "value": "13.235.132.32",
      "type": "string"
    },
    "web_port": {
      "value": 80,
      "type": "number"
    }
  },
  "resources": [
    {
      "mode": "managed",
      "type": "aws_cloudfront_distribution",
      "name": "cloud_distribute",
      "provider": "provider.aws",
      "instances": [
        {
          "schema_version": 1,
          "attributes": {
            "active_trusted_signers": {
              "enabled": "false",
              "items.#": "0"
            },
            "aliases": null,
            "arn":

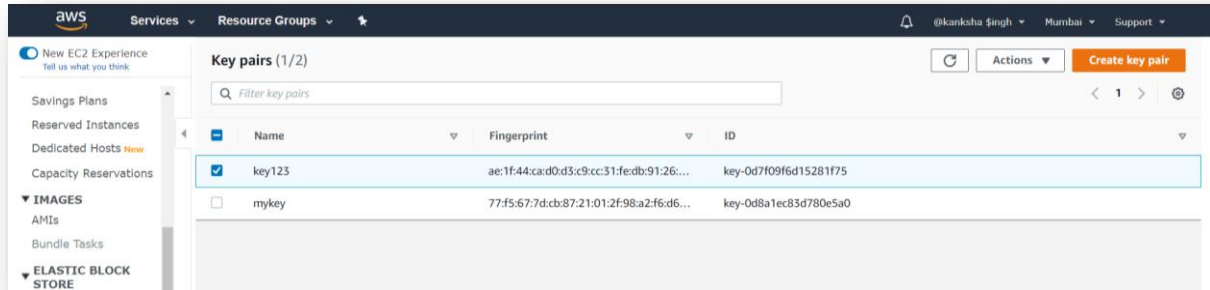
```

Now using the instance ip one may check the website on respective browser.

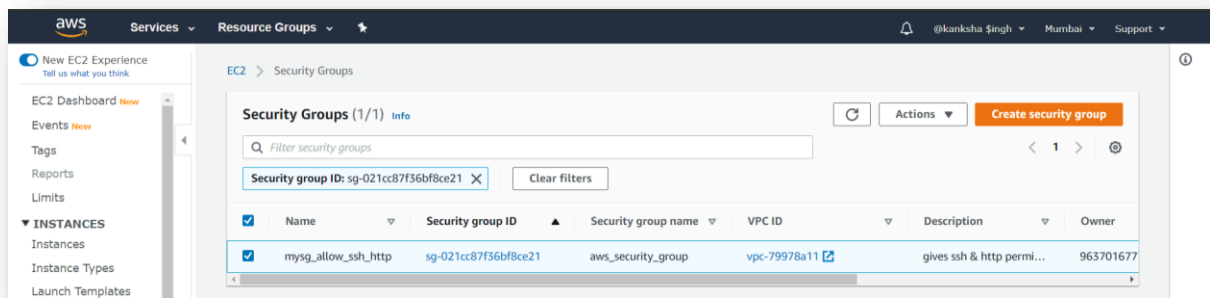


Let's watch the services created by Terraform inside my AWS management console (GUI).

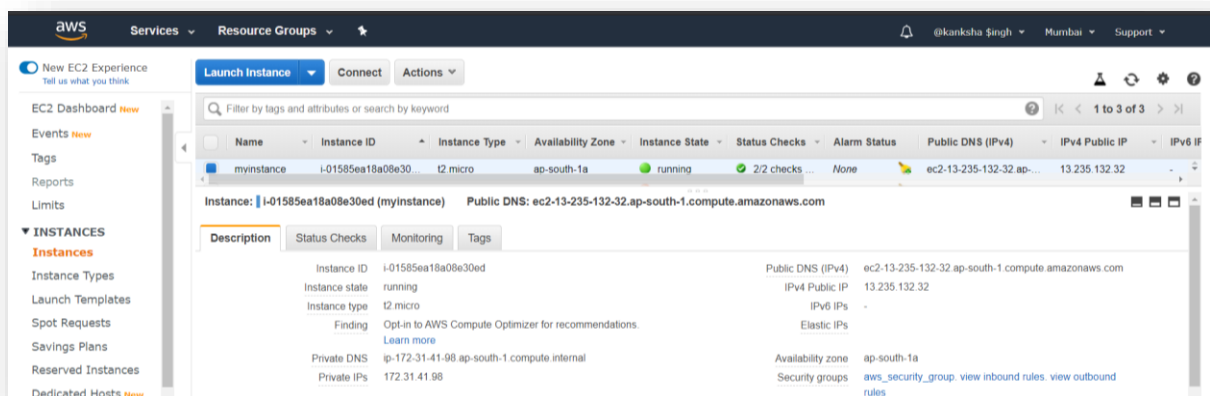
### ➤ Key-pair



### ➤ Security Group



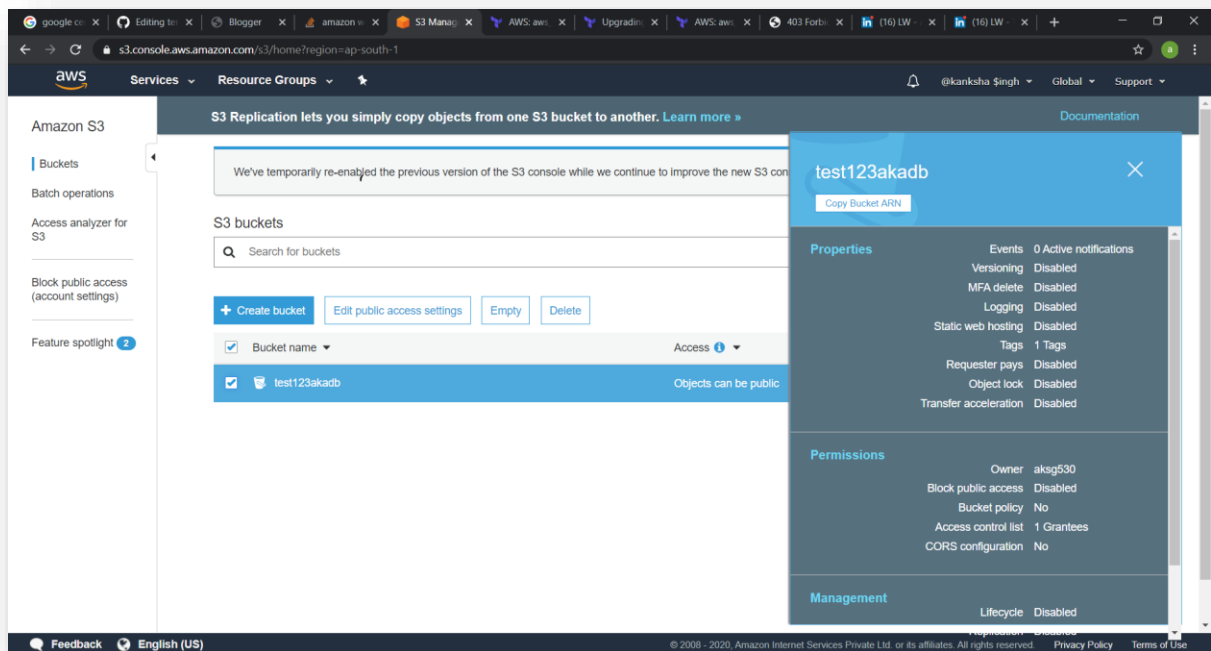
### ➤ Instance



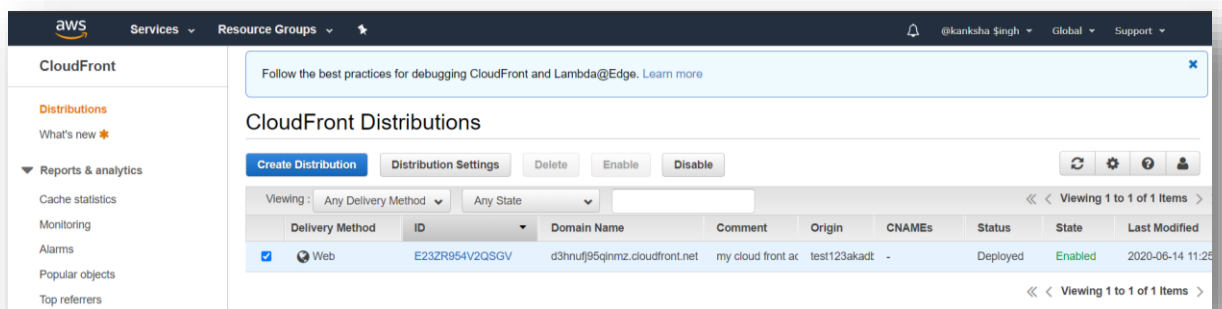
## ➤ EBS Volume



## ➤ S3 bucket

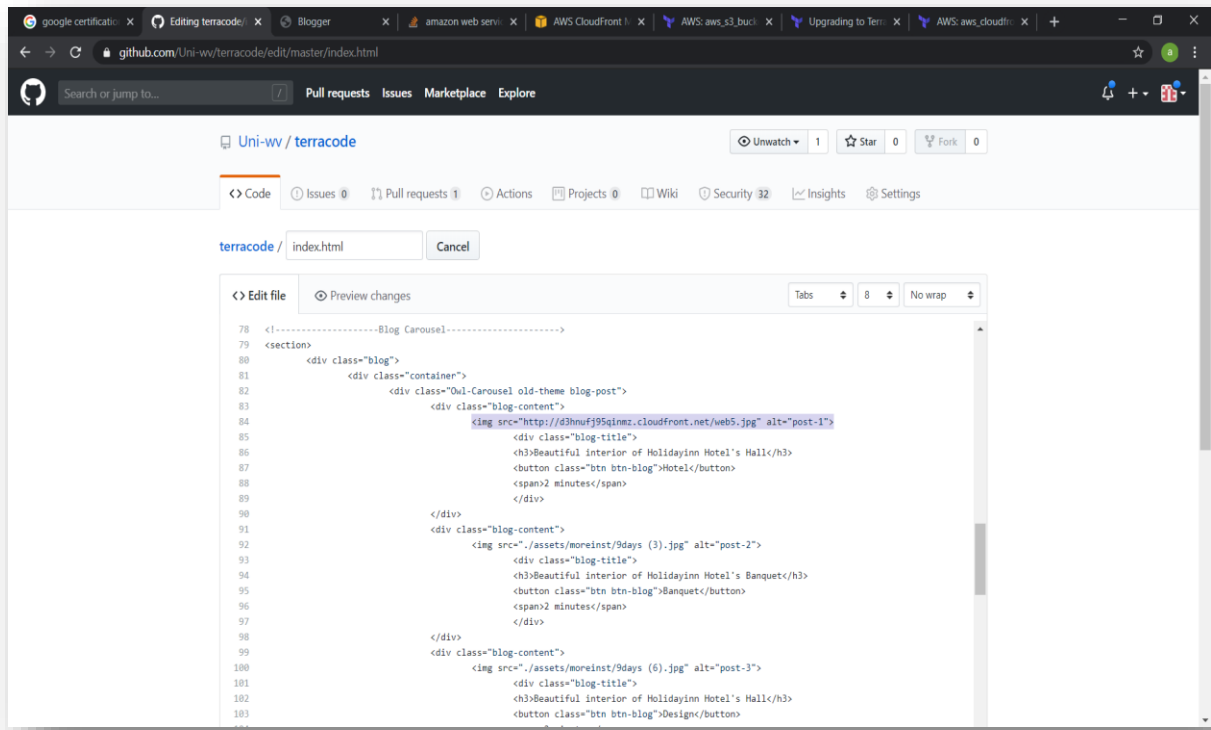


## ➤ Cloud Front Distribution

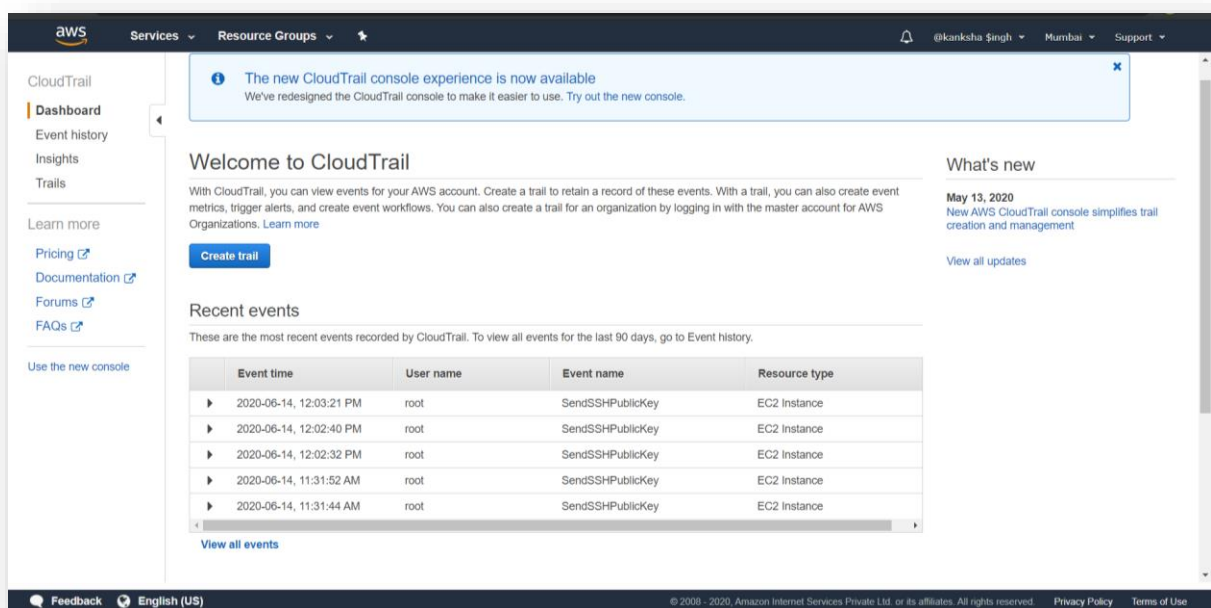




Have used the cloud front URL to get the image from the CDN edge location through which accessing the website is easy.



Now checkout the cloud Trail to trace the activities code have performed in AWS console with specific time.



## ERRORS:-

I also want to discuss the error generated while writing my terraform code and how I resolved them.

- ❖ Interpolation Warning: Due to syntax of variable occur due to the latest version of terraform software (Terraform v0.12.26). Resolved by removing "\${}" from variable name.

```
C:\Users\user\Desktop\teraform\myproj>terraform validate

Warning: Interpolation-only expressions are deprecated

  on job.tf line 73, in resource "aws_ebs_snapshot" "example_snapshot":
  73:   volume_id = "${aws_ebs_volume.ebs_volume.id}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${" sequence from the start and the "}"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.
```

- ❖ EOF error: End Of File error occur due to excessive spaces provided while writing blocks of code. Resolved by removing the extra spaces.

```
aws_instance.myinstance (remote-exec): Password: false
aws_instance.myinstance (remote-exec): Private key: true
aws_instance.myinstance (remote-exec): Certificate: false
aws_instance.myinstance (remote-exec): SSH Agent: false
aws_instance.myinstance (remote-exec): Checking Host Key: false
aws_instance.myinstance (remote-exec): Connected!

Error: Failed to upload script: Error chmodding script file to 0777 in remote machine: EOF
```

- ❖ Locals error inside s3 block:- Due to silly mistake, I tried to change the local name while writing origin\_id inside s3 block.

```
C:\Users\user\Desktop\teraform\myproj>terraform validate

Error: Reserved block type name in resource block

  on job.tf line 147, in resource "aws_s3_bucket" "aka_website":
 147:   locals {

The block type name "locals" is reserved for use by Terraform in a future
version.
```

- ❖ Error in viewer protocol policy in Cloud Front:- Solved using “viewer\_protocol\_policy = redirect-to-https” (in my case).

```
aws_s3_bucket_public_access_block.public_access: Creation complete after 1s [id=test123akadb]
aws_s3_bucket_object.object: Creation complete after 2s [id=my_bucket]

Error: error creating CloudFront Distribution: MalformedXML: 1 validation error detected: Value 'allow_all' at 'distributionConfigWithTags.distributionConfig.defaultCacheBehavior.viewerProtocolPolicy' failed to satisfy constraint: Member must satisfy enum value set: [redirect-to-https, https-only, allow-all]
    status code: 400, request id: 427d6e5f-d231-446e-9778-5a2c2b105297

on job.tf line 153, in resource "aws_cloudfront_distribution" "cloud_distribute":
153: resource "aws_cloudfront_distribution" "cloud_distribute" {
```

- ❖ Chrome error:- As Linux instance which terraform code created on cloud doesn't have chrome configured. To resolve it I have used curl command.

```
Error: Error running command 'chrome 13.235.132.32': exit status 1. Output: 'chrome' is not recognized as an internal or external command,
operable program or batch file.
```

## RESULT:-

Terraform code created all the services required to create an infrastructure on which one could deploy website. Have used Edge Location (Cloud Front) for low latency.

