AWS S3

Amazon Machine Image (AMI)

Amazon EC2

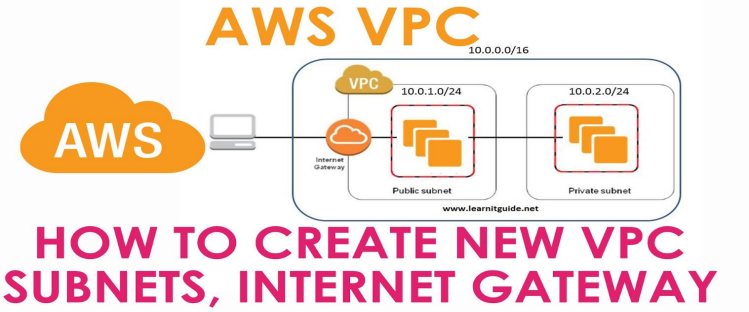
AWS EFS

aws

amazon cloudfront

HashiCorp Terraform

GitHub



Task-2

Perform the task-1 using EFS instead of EBS service on the AWS as,
---->>>> Create/launch Application using Terraform

1. Create Security group which allow the port 80.
2. Launch EC2 instance.
3. In this Ec2 instance use the existing key or provided key and security group which we have created in step 1.
4. Launch one Volume using the EFS service and attach it in your vpc, then 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

Optional
1) Those who are familiar with jenkins or are in devops AL have to integrate jenkins in this task wherever you feel can be integrated

Command Prompt

```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>aws configure --profile mypratik
AWS Access Key ID [****************H5JB]:
AWS Secret Access Key [****************0pRk]:
Default region name [ap-south-1]:
Default output format [json]:

C:\Users\Asus>
```

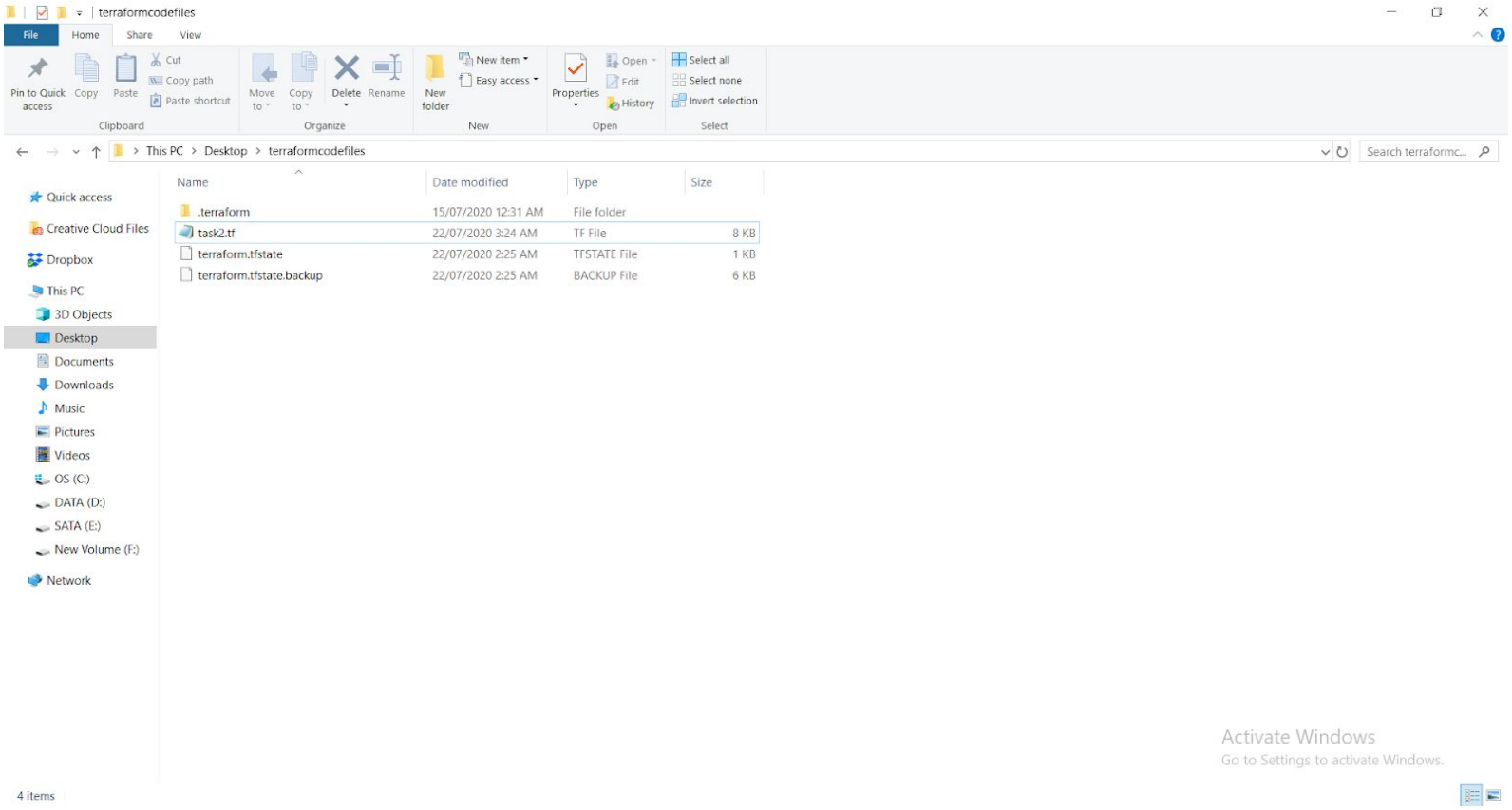
Command Prompt

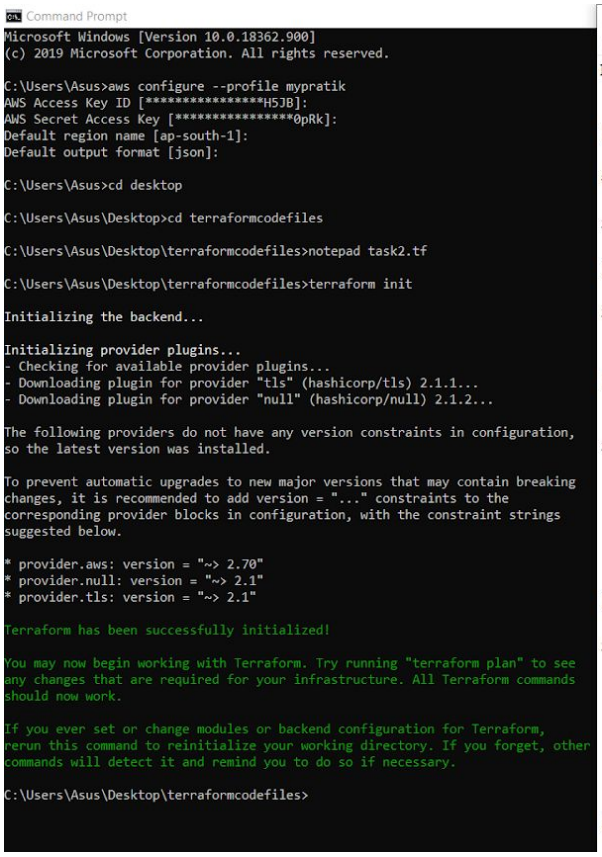
```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>aws configure --profile mypratik
AWS Access Key ID [****************H5JB]:
AWS Secret Access Key [****************0pRk]:
Default region name [ap-south-1]:
Default output format [json]:

C:\Users\Asus>cd desktop

C:\Users\Asus\Desktop>cd terraformcodefiles

C:\Users\Asus\Desktop\terraformcodefiles>notepad task2.tf

C:\Users\Asus\Desktop\terraformcodefiles>
```

task2.tf - Notepad

File Edit Format View Help

```
provider "aws" {
  region     = "ap-south-1"
  profile    = "mypratik"
}

#Creating Key

resource "tls_private_key" "key_form" {
  algorithm = "RSA"
}

resource "aws_key_pair" "task2_key" {
  key_name   = "mykey1112"
  public_key = tls_private_key.key_form.public_key_openssh
}


resource "aws_vpc" "pratik_vpc" {
  cidr_block = "10.0.0.0/16"
  instance_tenancy="default"
  enable_dns_hostnames = true
  tags = {
    Name = "pratik_vpc"
  }
}

resource "aws_subnet" "pratik_subnet" {
  vpc_id            = aws_vpc.myvpc.id
  availability_zone = "ap-south-1a"
  cidr_block        = "10.0.1.0/24"
  map_public_ip_on_launch = true
  tags = {
    Name = "pratik_subnet"
  }
}
```


terraformcodefiles
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC > Desktop > terraformcodefiles
Search terraformc...
Quick access
Creative Cloud Files
Dropbox
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
DATA (D:)
SATA (E:)
New Volume (F:)
Network
Name
Date modified
Type
Size
.terraform
15/07/2020 12:31 AM
File folder
task2.tf
22/07/2020 3:24 AM
TF File
8 KB
terraform.tfstate
22/07/2020 2:25 AM
TFSTATE File
1 KB
terraform.tfstate.backup
22/07/2020 2:25 AM
BACKUP File
6 KB
Activate Windows
Go to Settings to activate Windows.
4 items


Command Prompt
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Asus>aws configure --profile mypratik
AWS Access Key ID [**********H5JB]:
AWS Secret Access Key [**********0pRk]:
Default region name [ap-south-1]:
Default output format [json]:
C:\Users\Asus>cd desktop
C:\Users\Asus\Desktop>cd terraformcodefiles
C:\Users\Asus\Desktop\terraformcodefiles>notepad task2.tf
C:\Users\Asus\Desktop\terraformcodefiles>terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "tls" (hashicorp/tls) 2.1.1...
- Downloading plugin for provider "null" (hashicorp/null) 2.1.2...
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 = "~> 2.70"
* provider.null: version = "~> 2.1"
* provider.tls: 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.
C:\Users\Asus\Desktop\terraformcodefiles>


task2.tf - Notepad
File Edit Format View Help
provider "aws" {
region = "ap-south-1"
profile = "mypratik"
}
#Creating Key
resource "tls_private_key" "key_form" {
algorithm = "RSA"
}
resource "aws_key_pair" "task2_key" {
key_name = "mykey1112"
public_key = tls_private_key.key_form.public_key_openssh
}
resource "aws_vpc" "pratik_vpc" {
cidr_block = "10.0.0.0/16"
instance_tenancy="default"
enable_dns_hostnames = true
tags = {
Name = "pratik_vpc"
}
}
resource "aws_subnet" "pratik_subnet" {
vpc_id = aws_vpc.myvpc.id
availability_zone = "ap-south-1a"
cidr_block = "10.0.1.0/24"
map_public_ip_on_launch = true
tags = {
Name = "pratik_subnet"
}
}

Command Prompt

```
C:\Users\Asus\Desktop\terraformcodefiles>terraform validate
Success! The configuration is valid.


C:\Users\Asus\Desktop\terraformcodefiles>terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_cloudfront_distribution.mycdistribution will be created
  + resource "aws_cloudfront_distribution" "mycdistribution" {
      + active_trusted_signers         = (known after apply)
      + arn                            = (known after apply)
      + caller_reference               = (known after apply)
      + default_root_object            = "index.php"
      + domain_name                    = (known after apply)
      + enabled                        = true
      + etag                           = (known after apply)
      + hosted_zone_id                 = (known after apply)
      + http_version                   = "http2"
      + id                             = (known after apply)
      + in_progress_validation_batches = (known after apply)
      + is_ipv6_enabled                = true
      + last_modified_time             = (known after apply)
      + price_class                    = "PriceClass_All"
      + retain_on_delete               = false
      + status                         = (known after apply)
      + tags                           = {
          + "Name" = "mycdistribution"
        }
      + wait_for_deployment            = true

      + custom_error_response {
          + error_caching_min_ttl = 3000
          + error_code            = 404
          + response_code         = 200
          + response_page_path    = "/1234.jpg"
        }

      + default_cache_behavior {
          + allowed_methods        = [
              + "DELETE",
              + "GET",
              + "HEAD",
              + "OPTIONS",
              + "PATCH",
              + "POST",
```

```
C:\Users\Asus\Desktop\terraformcodefiles>terraform validate
Success! The configuration is valid.
```

Command Prompt

```
              + "PUT",
            ]
          + cached_methods         = [
              + "GET",
              + "HEAD",
            ]
          + compress               = false
          + default_ttl            = 3600
          + max_ttl                = 86400
          + min_ttl                = 0
          + target_origin_id       = "pratik1234"
          + viewer_protocol_policy = "allow-all"

          + forwarded_values {
              + query_string = false

              + cookies {
                  + forward = "none"
                }
            }
        }

      + origin {
          + domain_name = (known after apply)
          + origin_id   = "pratik1234"

          + s3_origin_config {
              + origin_access_identity = (known after apply)
            }
        }

      + restrictions {
          + geo_restriction {
              + restriction_type = "none"
            }
        }

      + viewer_certificate {
          + cloudfront_default_certificate = true
          + minimum_protocol_version       = "TLSv1"
        }
    }

  # aws_cloudfront_origin_access_identity.origin_access_identity will be created
  + resource "aws_cloudfront_origin_access_identity" "origin_access_identity" {
      + caller_reference               = (known after apply)
      + cloudfront_access_identity_path = (known after apply)
      + comment                        = "Sync Cloudfront to s3"
      + etag                           = (known after apply)
      + iam_arn                        = (known after apply)
```

Command Prompt

```
  # aws_efs_file_system.myefs will be created
  + resource "aws_efs_file_system" "myefs" {
      + arn              = (known after apply)
      + creation_token   = "my-efs"
      + dns_name         = (known after apply)
      + encrypted        = (known after apply)
      + id               = (known after apply)
      + kms_key_id       = (known after apply)
      + performance_mode = "generalPurpose"
      + reference_name   = (known after apply)
      + tags             = {
          + "Name" = "myefs"
        }
      + throughput_mode  = "bursting"
    }

  # aws_efs_file_system_policy.policy will be created
  + resource "aws_efs_file_system_policy" "policy" {
      + file_system_id = (known after apply)
      + id             = (known after apply)
      + policy         = (known after apply)
    }

  # aws_efs_mount_target.mymount will be created
  + resource "aws_efs_mount_target" "mymount" {
      + availability_zone_id   = (known after apply)
      + availability_zone_name = (known after apply)
      + dns_name               = (known after apply)
      + file_system_arn        = (known after apply)
      + file_system_id         = (known after apply)
      + id                     = (known after apply)
      + ip_address             = (known after apply)
      + mount_target_dns_name  = (known after apply)
      + network_interface_id   = (known after apply)
      + owner_id               = (known after apply)
      + security_groups        = (known after apply)
      + subnet_id              = (known after apply)
    }

  # aws_instance.myin will be created
  + resource "aws_instance" "myin" {
      + ami                          = "ami-00b494a3f139ba61f"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = "ap-south-1a"
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
```

Command Prompt

```
  # aws_instance.myin will be created
  + resource "aws_instance" "myin" {
      + ami                          = "ami-00b494a3f139ba61f"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = "ap-south-1a"
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = "mykey1112"
      + network_interface_id         = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = (known after apply)
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name" = "myterraOS"
        }
      + tenancy                      = (known after apply)
      + volume_tags                  = (known after apply)
      + vpc_security_group_ids       = (known after apply)

      + ebs_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + snapshot_id           = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }

      + ephemeral_block_device {
          + device_name  = (known after apply)
          + no_device    = (known after apply)
```

Command Prompt

```
      }

    + ephemeral_block_device {
        + device_name  = (known after apply)
        + no_device    = (known after apply)
        + virtual_name = (known after apply)
      }

    + metadata_options {
        + http_endpoint               = (known after apply)
        + http_put_response_hop_limit = (known after apply)
        + http_tokens                 = (known after apply)
      }

    + network_interface {
        + delete_on_termination = (known after apply)
        + device_index          = (known after apply)
        + network_interface_id  = (known after apply)
      }

    + root_block_device {
        + delete_on_termination = (known after apply)
        + device_name           = (known after apply)
        + encrypted             = (known after apply)
        + iops                  = (known after apply)
        + kms_key_id            = (known after apply)
        + volume_id             = (known after apply)
        + volume_size           = (known after apply)
        + volume_type           = (known after apply)
      }
  }

# aws_internet_gateway.mygateway will be created
+ resource "aws_internet_gateway" "mygateway" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "mygateway"
      }
    + vpc_id   = (known after apply)
  }

# aws_key_pair.task2_key will be created
+ resource "aws_key_pair" "task2_key" {
    + arn         = (known after apply)
    + fingerprint = (known after apply)
    + id          = (known after apply)
    + key_name    = "mykey1112"
    + key_pair_id = (known after apply)
```

Command Prompt

```
  }

# aws_route_table.myroutetable will be created
+ resource "aws_route_table" "myroutetable" {
    + id              = (known after apply)
    + owner_id        = (known after apply)
    + propagating_vgws = (known after apply)
    + route           = [
        + {
            + cidr_block                = "0.0.0.0/0"
            + egress_only_gateway_id    = ""
            + gateway_id                = (known after apply)
            + instance_id               = ""
            + ipv6_cidr_block           = ""
            + nat_gateway_id            = ""
            + network_interface_id      = ""
            + transit_gateway_id        = ""
            + vpc_peering_connection_id = ""
          },
      ]
    + tags            = {
        + "Name" = "myroutetable"
      }
    + vpc_id          = (known after apply)
  }

# aws_route_table_association.assoc will be created
+ resource "aws_route_table_association" "assoc" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_s3_bucket.mybucket will be created
+ resource "aws_s3_bucket" "mybucket" {
    + acceleration_status         = (known after apply)
    + acl                         = "public-read"
    + arn                         = (known after apply)
    + bucket                      = "pratik1234"
    + bucket_domain_name          = (known after apply)
    + bucket_regional_domain_name = (known after apply)
    + force_destroy               = true
    + hosted_zone_id              = (known after apply)
    + id                          = (known after apply)
    + region                      = (known after apply)
    + request_payer               = (known after apply)
    + website_domain              = (known after apply)
    + website_endpoint            = (known after apply)

    + cors_rule {
```

Command Prompt

```
    + cors_rule {
        + allowed_headers = [
            + "*",
          ]
        + allowed_methods = [
            + "PUT",
            + "POST",
          ]
        + allowed_origins = [
            + "https://pratik1234",
          ]
        + expose_headers  = [
            + "ETag",
          ]
        + max_age_seconds = 3000
      }

    + versioning {
        + enabled    = (known after apply)
        + mfa_delete = (known after apply)
      }
  }

# aws_s3_bucket_object.myobj will be created
+ resource "aws_s3_bucket_object" "myobj" {
    + acl                    = "public-read"
    + bucket                 = (known after apply)
    + content_type           = (known after apply)
    + etag                   = (known after apply)
    + force_destroy          = false
    + id                     = (known after apply)
    + key                    = "1234.jpg"
    + server_side_encryption = (known after apply)
    + source                 = "C:/Users/Asus/Downloads/1234.jpg"
    + storage_class          = (known after apply)
    + version_id             = (known after apply)
  }

# aws_security_group.mysg will be created
+ resource "aws_security_group" "mysg" {
    + arn                    = (known after apply)
    + description            = "Allow HTTP,ssh for inbound traffic"
    + egress                 = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 0
```

Command Prompt

```
  # aws_security_group.mysg will be created
  + resource "aws_security_group" "mysg" {
      + arn                    = (known after apply)
      + description            = "Allow HTTP,ssh for inbound traffic"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = "Http from VPC"
              + from_port        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
            },
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = "NFS"
              + from_port        = 2049
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 2049
            },
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
```

Command Prompt

```
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = "SSH from VPC"
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
      ]
    + name                   = "mysg"
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "mysecuritygroup"
      }
    + vpc_id                 = (known after apply)
  }

# aws_subnet.pratik_subnet will be created
+ resource "aws_subnet" "pratik_subnet" {
    + arn                             = (known after apply)
    + assign_ipv6_address_on_creation = false
    + availability_zone               = "ap-south-1a"
    + availability_zone_id            = (known after apply)
    + cidr_block                      = "10.0.1.0/24"
    + id                              = (known after apply)
    + ipv6_cidr_block                 = (known after apply)
    + ipv6_cidr_block_association_id  = (known after apply)
    + map_public_ip_on_launch         = true
    + owner_id                        = (known after apply)
    + tags                            = {
        + "Name" = "pratik_subnet"
      }
    + vpc_id                          = (known after apply)
  }

# aws_vpc.pratik_vpc will be created
+ resource "aws_vpc" "pratik_vpc" {
    + arn                              = (known after apply)
    + assign_generated_ipv6_cidr_block = false
    + cidr_block                       = "10.0.0.0/16"
    + default_network_acl_id           = (known after apply)
    + default_route_table_id           = (known after apply)
    + default_security_group_id        = (known after apply)
    + dhcp_options_id                  = (known after apply)
    + enable_classiclink               = (known after apply)
```

Command Prompt

```
  # null_resource.key_pair will be created
  + resource "null_resource" "key_pair" {
      + id = (known after apply)
    }

  # null_resource.null_vol_attach will be created
  + resource "null_resource" "null_vol_attach" {
      + id = (known after apply)
    }

  # tls_private_key.key_form will be created
  + resource "tls_private_key" "key_form" {
      + algorithm                = "RSA"
      + ecdsa_curve              = "P224"
      + id                       = (known after apply)
      + private_key_pem          = (sensitive value)
      + public_key_fingerprint_md5 = (known after apply)
      + public_key_openssh       = (known after apply)
      + public_key_pem           = (known after apply)
      + rsa_bits                 = 2048
    }

Plan: 18 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

tls_private_key.key_form: Creating...
tls_private_key.key_form: Creation complete after 1s [id=584f920a6c60e460a4112d056052f2c52697d31c]
null_resource.key_pair: Creating...
null_resource.key_pair: Provisioning with 'local-exec'...
null_resource.key_pair (local-exec): Executing: ["cmd" "/C" "echo  '-----BEGIN RSA PRIVATE KEY-----\nMIIEogIBAAKCAQEAsw/eCXm1athpVjXH6AcdQu/0Csc2NbdmByySXvKjrpkUSMBf\nzeyZxd9j0MtIzmAcuLosDE+x7ktG7SexuZLkyTGBkW/R
241hyf9cA8ZA13rEPFoE\n+h5AoGHAIIzIww/3+o0zCuoryfnCslzLitH1EVZxqHE4xSKrW3Sz4NUk0p/5+p2r\nhQXhHq01CcWRiOysaSBysFXHGJxL5SE0r1NPSGhScaMABpPlWhVmzhmvTjRU0UCr\n2HhI3uL0W3y0+mx5dqHzKSioP/LyQNh3PbmS+b74Uw0XD7rOoE3fid6ZC
qd3bssO\nD6+6ycUkyUQaOX5kbYAUIEHauEFCzTf/n+iSqwIDAQABAoIBAC++MVORcf+1CBEG\ncfC5xOGOHSGqkBeueLO8/KH4TBBHnJLxSBod1bpen7ySFy1/jujqKsh1j1mcFKMz\nDD4QnpPOcV+X0vqZLdOwFJalC7DoeN4mmmEeI9kfdxy2rOjlnxbdAxkL6vUFdRYd\ni9/N
r2waAmAJIDvfzJSr/zIZYqINtzhbGZ4pLRl4qEOlucKAqdsveeVTCIOQr/0b\nWd8YAwrALYsox7SX+84Mk0lKmvxpqmRrXvWwWf7829os80ReAfORYdWDbuSYwvQw\ni4JtbDXStp31x0Snn4BjotWhHH1ehkpd1MJnM5EJb3h3GmAL+qM/M/qroIH/Vpgp\nC/CSLLECgYEA4xQfo
CUcWpmK0kbcrRqdm3IIS6m3WZ7MmCK68r9e4NdjwYEbAaha\nFSuL4gBhWPgn5udfbrHo4gQwTSpDfeDr+Wc3Xfx73oJ3uAQA80YHrxnVndSBaTN/\n8VHwUfk2gKLc/aLlF4Qhy8l14Dogq2yXjPkep3CHzoBT0QvX7z6Vfv8CgYEAyd4q\niTOgk98mSWHstcfndC30ek1hqRCLs6
yrAw0y1wEhRsoOAJMlvIVC+PQThFLB4Gtc\n7RXtbqhvR04pKqgl9Nvi3ulDzDrzsGwy96Hji6HPU/rN0nNSCf1NmHJFyjHkYe7D\nqi8kAxkVVTuQgcU4QhdS7puD2oAva7oOKskimFUCgYAjIk3zmipU6kTMXCJwthmE\nn5QhANVZjHRjZZw/StTHPQw2OF+RVhb6B47RHPXy6hJ
ia5IccnJx5ewDRHLE2zQn\nloy+gf44Yz9avWPYQBGYSgr/LI4JVOqKz/668DA6nvhKVhXUGaGRAk62rse3PkoE\nidbwWTzp+nj9czOp1KkUyQKBgHC21QdQxVtroOQegl4ppWmwcy6IpLgSnf6P7djB\nWN7GKb9t+I2Hm+18dx2sou0ijFBzgip7MNy1vZSR5tiDlNX4pN+UXZ++
VuzuLTMp\nuHTxv2MDykK001zyi44QWDmCzCOGfsgalEAKlGfH33MmHLiCb3L9cIzjQBclht2K\n+v8VAoGAR31Yo3PF369kHASNHDXzUnT7lYZFYf+2ygKW+vgY3XKxKQXw/1vFuse0\nPbS44Ea2Dvh9DSxymbXmb5L4fKLOajcnOKsEi7X6mp4hIJMNuaC3hnfJrFLhdTp8\n+J4
Udwy+9sEWLRGk239VkB6LQdQsfQJ2G4CBvm414tTNxHm9oww=\n-----END RSA PRIVATE KEY-----\n' > key.pem"]
null_resource.key_pair (local-exec):  '-----BEGIN RSA PRIVATE KEY-----
null_resource.key_pair: Creation complete after 0s [id=1997524165343346184]
aws_cloudfront_origin_access_identity.origin_access_identity: Creating...
aws_efs_file_system.myefs: Creating...
aws_key_pair.task2_key: Creating...
aws_vpc.pratik_vpc: Creating...
```

```
null_resource.null_vol_attach (remote-exec): Total download size: 14 M
null_resource.null_vol_attach (remote-exec): Installed size: 62 M
null_resource.null_vol_attach (remote-exec): Downloading packages:
null_resource.null_vol_attach (remote-exec): (1/21): amazon-efs |  33 kB   00:00
null_resource.null_vol_attach (remote-exec): (2/21): apr-1.6.3- | 118 kB   00:00
null_resource.null_vol_attach (remote-exec): (3/21): apr-util-b |  19 kB   00:00
null_resource.null_vol_attach (remote-exec): (4/21): apr-util-1 |  99 kB   00:00
null_resource.null_vol_attach (remote-exec): (5/21): generic-lo |  19 kB   00:00
null_resource.null_vol_attach (remote-exec): (6/21): git-2.23.3 | 135 kB   00:00
null_resource.null_vol_attach (remote-exec): (7/21): git-core-d | 2.4 MB   00:00
null_resource.null_vol_attach (remote-exec): (8/21): git-core-2 | 5.0 MB   00:00
null_resource.null_vol_attach (remote-exec): (9/21): httpd-2.4. | 1.3 MB   00:00
null_resource.null_vol_attach (remote-exec): (10/21): httpd-fil |  23 kB   00:00
null_resource.null_vol_attach (remote-exec): (11/21): httpd-too |  87 kB   00:00
null_resource.null_vol_attach (remote-exec): (12/21): libzip010 |  30 kB   00:00
null_resource.null_vol_attach (remote-exec): (13/21): mailcap-2 |  31 kB   00:00
null_resource.null_vol_attach (remote-exec): (14/21): mod_http2 | 146 kB   00:00
null_resource.null_vol_attach (remote-exec): (15/21): perl-Erro |  32 kB   00:00
null_resource.null_vol_attach (remote-exec): (16/21): perl-Git- |  47 kB   00:00
null_resource.null_vol_attach (remote-exec): (17/21): perl-Term |  31 kB   00:00
null_resource.null_vol_attach (remote-exec): (18/21): php-5.4.1 | 1.4 MB   00:00
null_resource.null_vol_attach (remote-exec): (19/21): php-commo | 563 kB   00:00
null_resource.null_vol_attach (remote-exec): (20/21): stunnel-4 | 149 kB   00:00
null_resource.null_vol_attach (remote-exec): (21/21): php-cli-5 | 2.8 MB   00:00
null_resource.null_vol_attach (remote-exec): ----------------------------------------
null_resource.null_vol_attach (remote-exec): Total       38 MB/s |  14 MB  00:00
null_resource.null_vol_attach (remote-exec): Running transaction check
null_resource.null_vol_attach (remote-exec): Running transaction test
null_resource.null_vol_attach (remote-exec): Transaction test succeeded
null_resource.null_vol_attach (remote-exec): Running transaction
null_resource.null_vol_attach (remote-exec):   Installing : apr-1.6 [        ]  1/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-1.6 [##      ]  1/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-1.6 [####    ]  1/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-1.6 [######  ]  1/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-1.6 [####### ]  1/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-1.6.3-5.amzn    1/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [        ]  2/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [####### ]  2/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-util-bdb-1.6    2/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [        ]  3/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [##      ]  3/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [####    ]  3/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [######  ]  3/21
null_resource.null_vol_attach (remote-exec):   Installing : apr-uti [####### ]  3/21
```

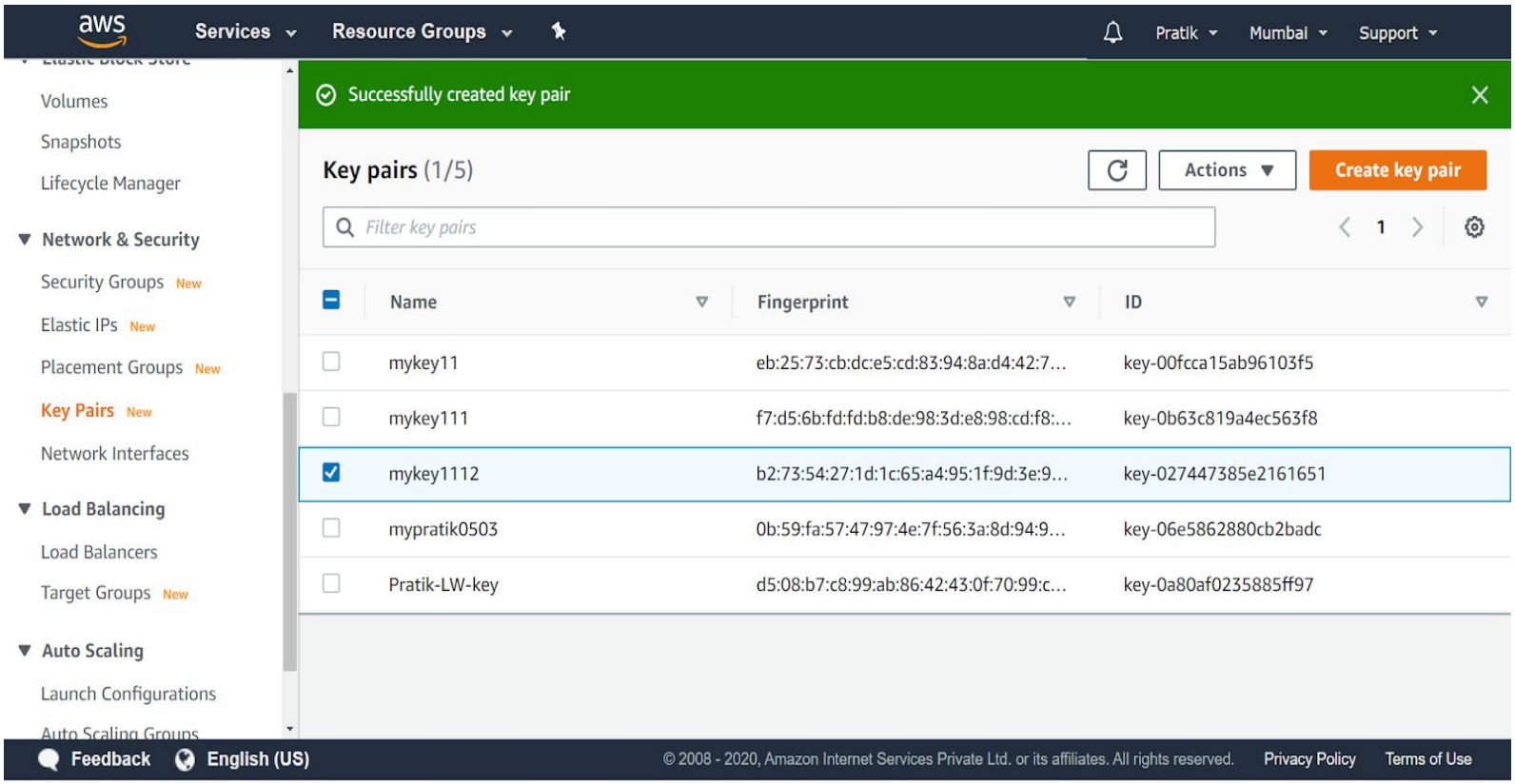
```
Apply complete! Resources: 18 added, 0 changed, 0 destroyed.

Outputs:

cloudfront_ip_addr = d3unb4s4rfnx5d.cloudfront.net
```

#Terraform Code:-

```
provider "aws" {
  region     = "ap-south-1"
  profile    = "mypratik"
}
```

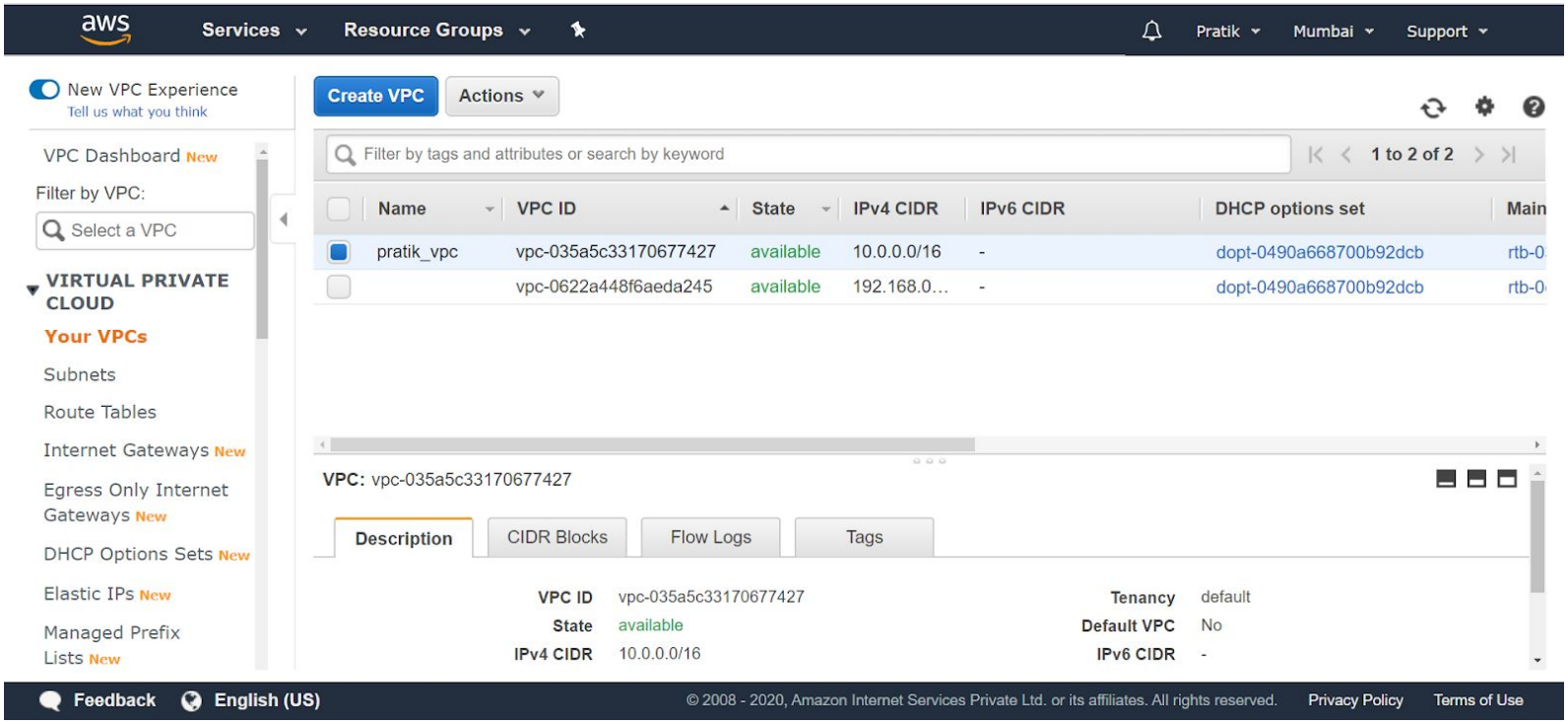
#Creating Key

```
resource "tls_private_key" "key_form" {
  algorithm = "RSA"
}

resource "aws_key_pair" "task2_key" {
  key_name   = "mykey1112"
  public_key = tls_private_key.key_form.public_key_openssh
}
```



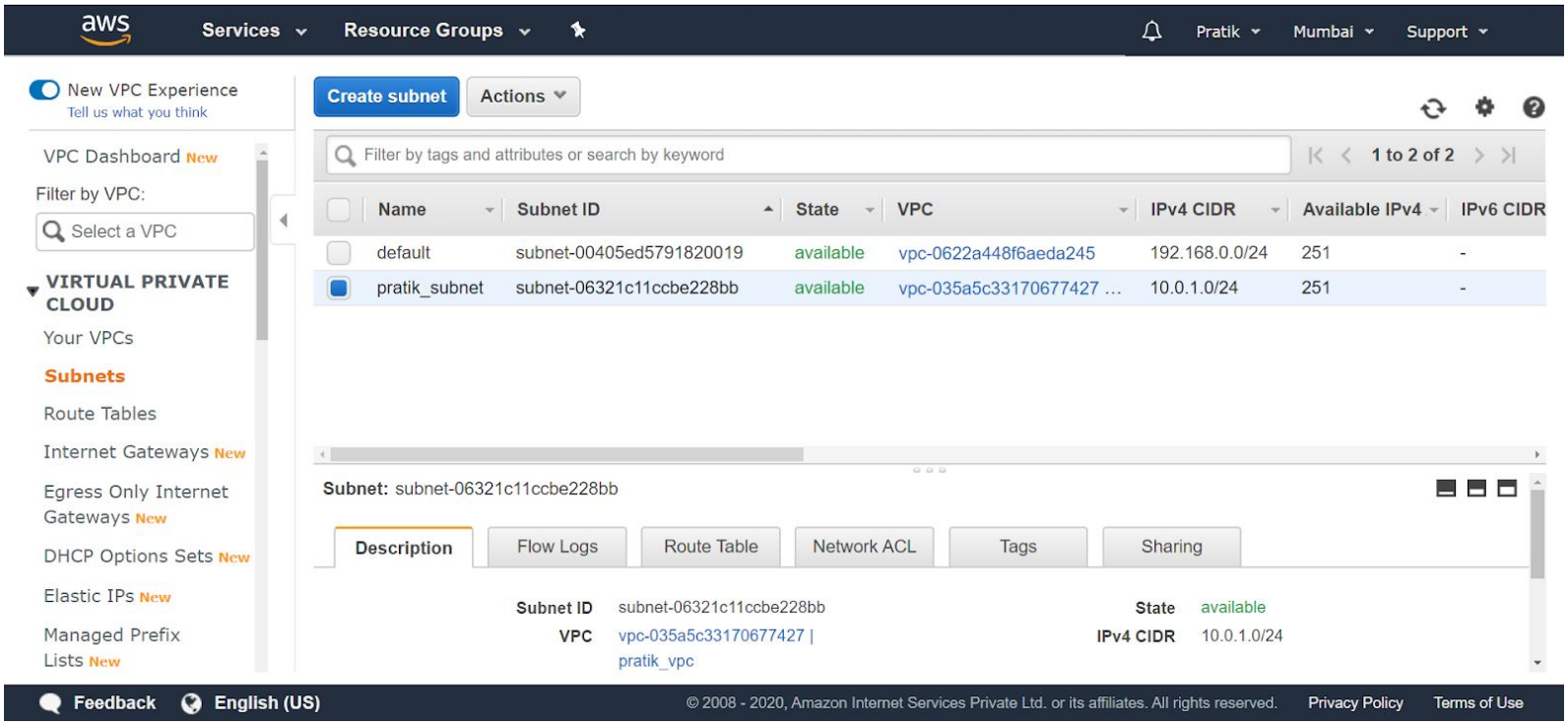
#Creating Virtual Private Cloud

```
resource "aws_vpc" "pratik_vpc" {
  cidr_block = "10.0.0.0/16"
  instance_tenancy="default"
  enable_dns_hostnames = true
  tags = {
    Name = "pratik_vpc"
  }
}
```



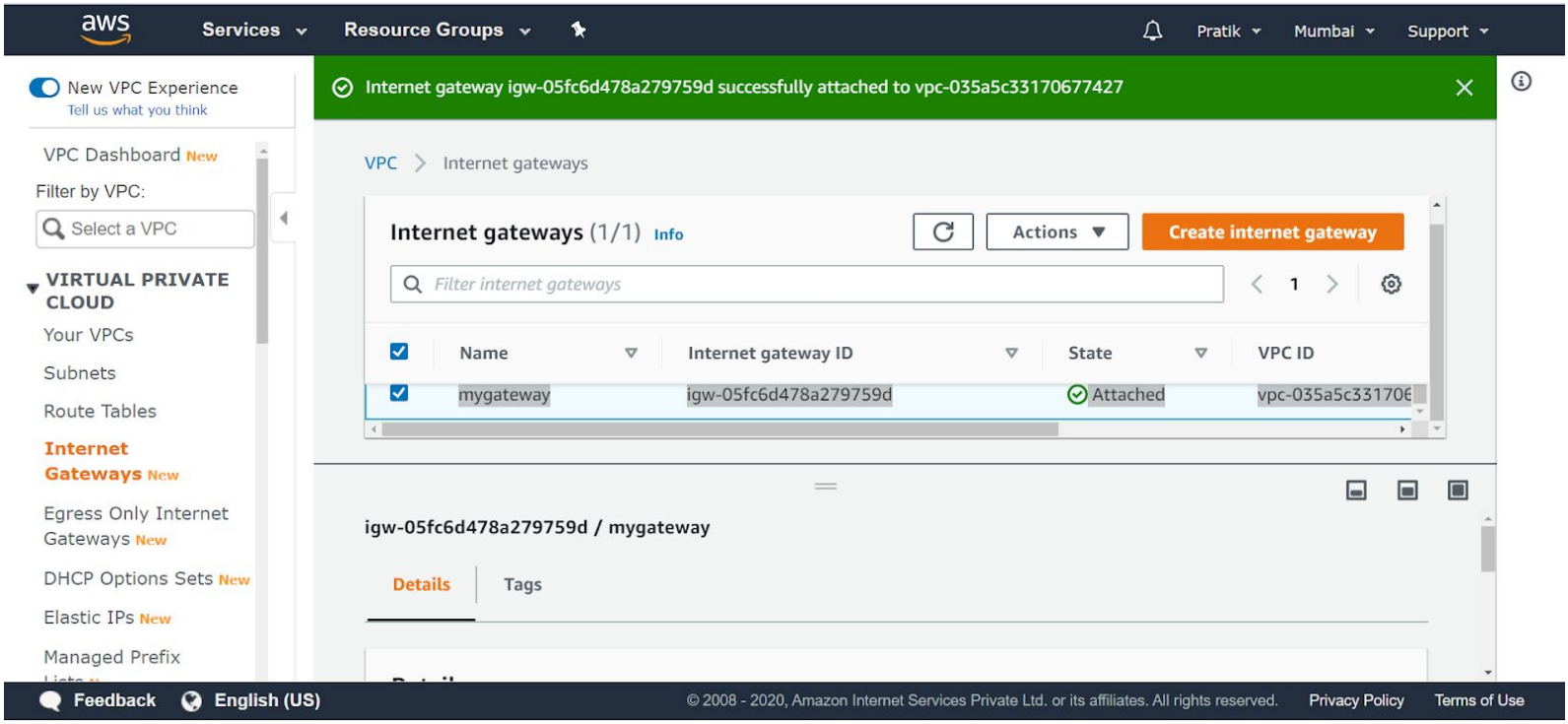
#Creating Subnet

```
resource "aws_subnet" "pratik_subnet" {
  vpc_id            = aws_vpc.pratik_vpc.id
  availability_zone = "ap-south-1a"
  cidr_block        = "10.0.1.0/24"
  map_public_ip_on_launch = true
  tags = {
    Name = "pratik_subnet"
  }
}
```



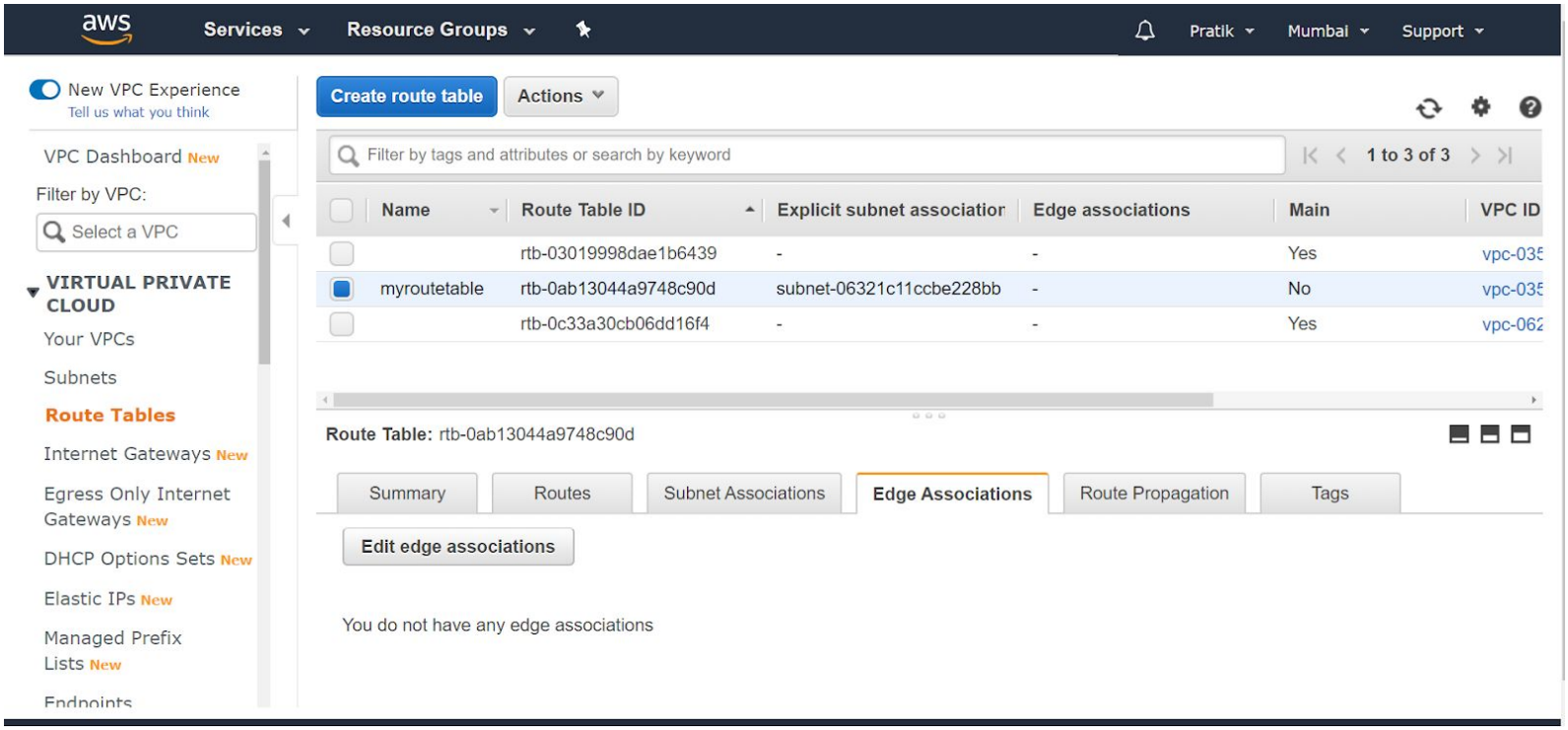
#Creating Internet Gateway

```
resource "aws_internet_gateway" "mygateway" {
  vpc_id = aws_vpc.pratik_vpc.id
  tags = {
    Name = "mygateway"
  }
}
```



#Creating Route Table

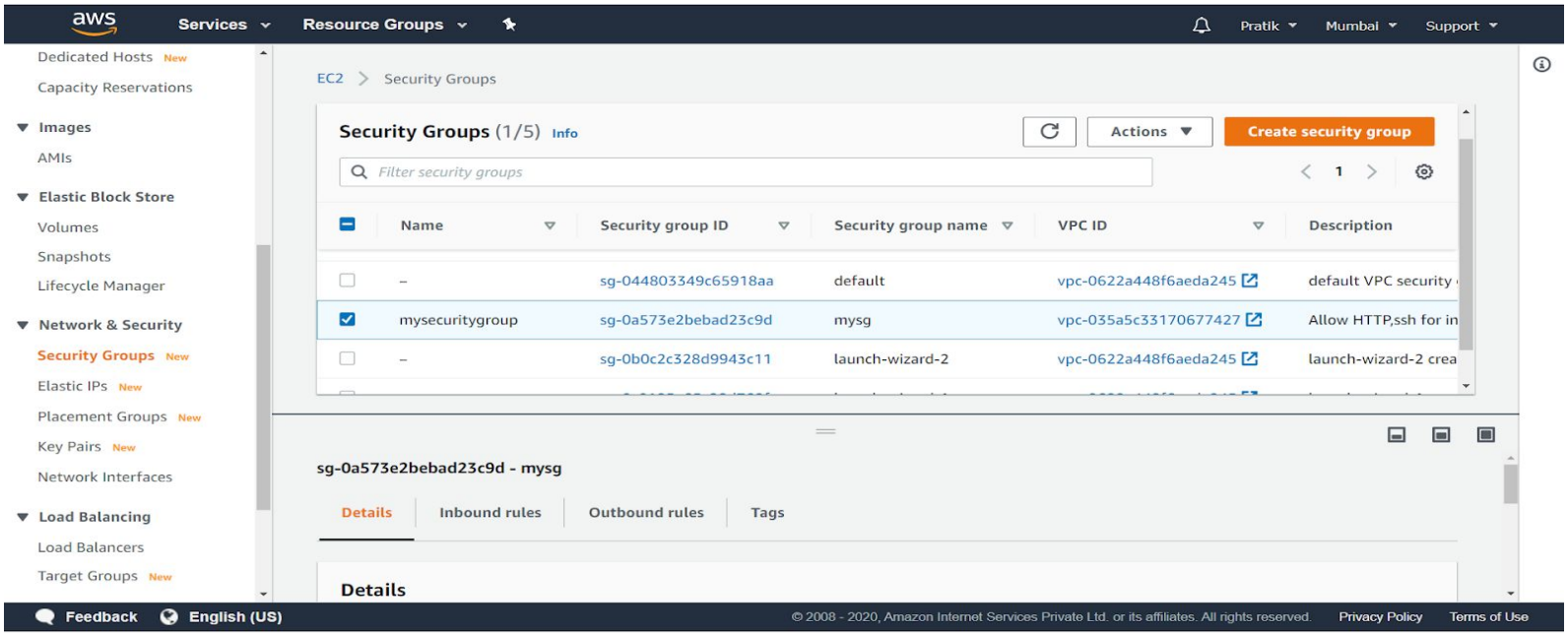
```
resource "aws_route_table" "myroutetable" {
  vpc_id = aws_vpc.pratik_vpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.mygateway.id
  }
  tags = {
    Name = "myroutetable"
  }
}
resource "aws_route_table_association" "assoc" {
  subnet_id      = aws_subnet.pratik_subnet.id
  route_table_id = aws_route_table.myroutetable.id
}
```

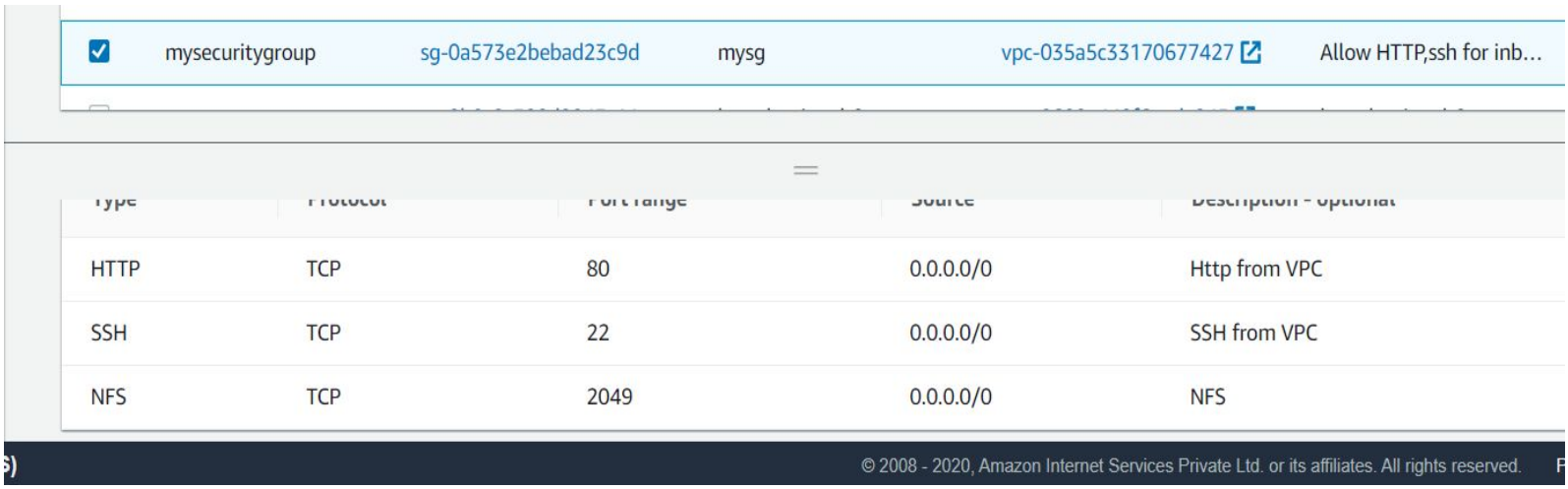


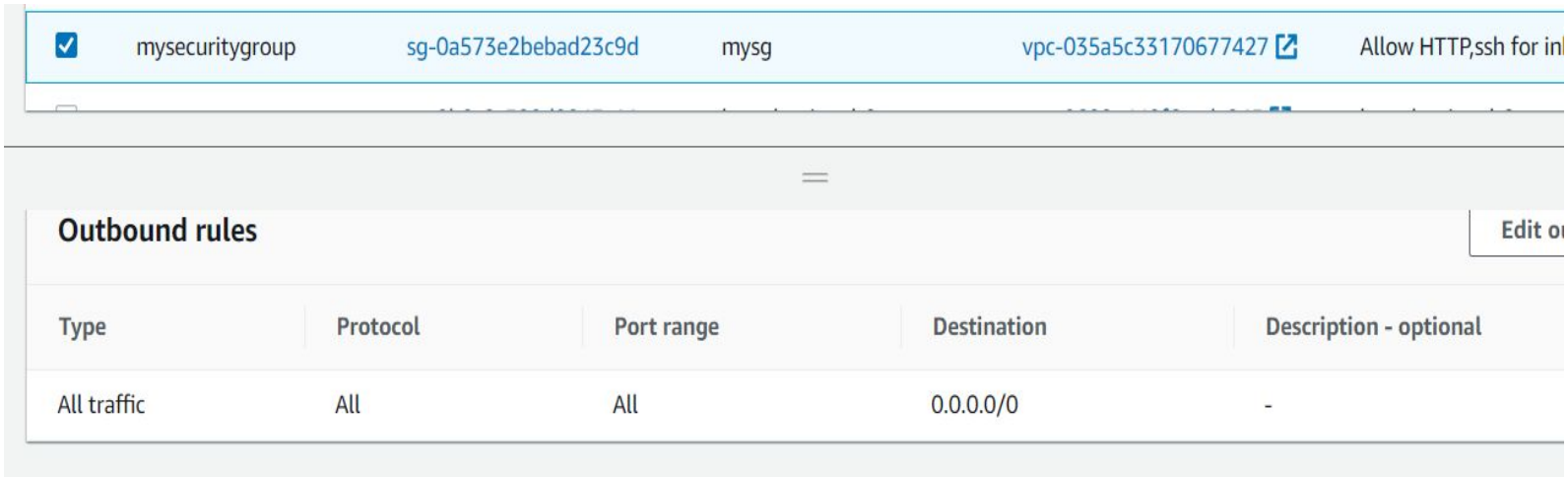
#Creating Security group

```
resource "aws_security_group" "mysg" {
 name        = "mysg"
 description = "Allow HTTP,ssh for inbound traffic"
 vpc_id      = aws_vpc.pratik_vpc.id

 ingress {
  description = "Http from VPC"
  from_port   = 80
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = [ "0.0.0.0/0" ]
 }
 ingress {
  description = "SSH from VPC"
  from_port   = 22
  to_port     = 22
  protocol    = "tcp"
  cidr_blocks = [ "0.0.0.0/0" ]
 }
 ingress {
  description = "NFS"
  from_port   = 2049
  to_port     = 2049
  protocol    = "tcp"
  cidr_blocks = [ "0.0.0.0/0" ]
 }
 egress {
  from_port   = 0
  to_port     = 0
  protocol    = "-1"
  cidr_blocks = ["0.0.0.0/0"]
 }

 tags = {
  Name = "mysecuritygroup"
```

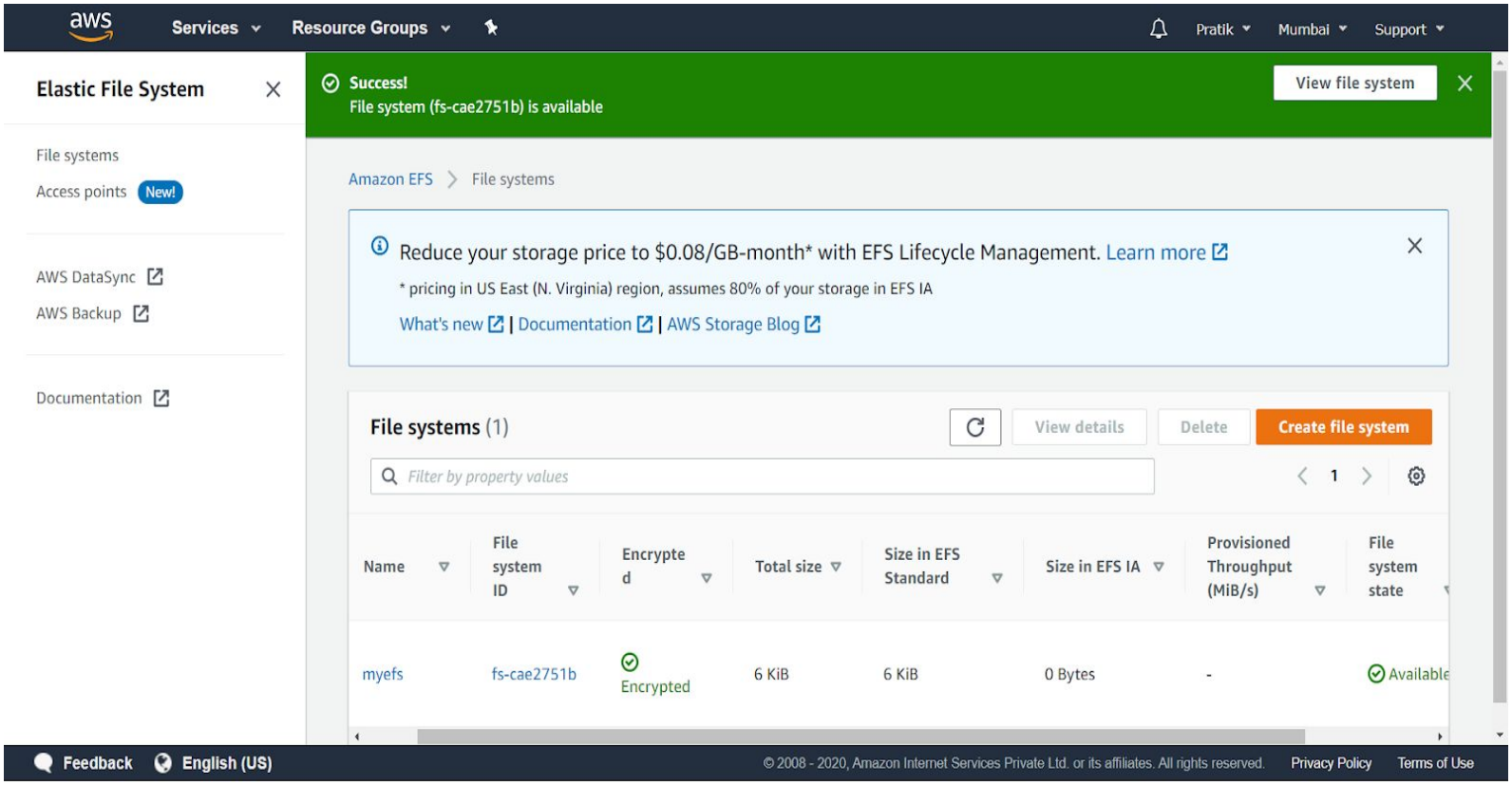
```
  }
}
```


aws
Services
Resource Groups
Pratik
Mumbai
Support
Dedicated Hosts New
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
Load Balancing
Load Balancers
Target Groups New
EC2 > Security Groups
Security Groups (1/5) Info
Actions
Create security group
Filter security groups
Name
Security group ID
Security group name
VPC ID
Description
– sg-044803349c65918aa default vpc-0622a448f6aeda245 default VPC security
mysecuritygroup sg-0a573e2bebad23c9d mysg vpc-035a5c33170677427 Allow HTTP,ssh for in
– sg-0b0c2c328d9943c11 launch-wizard-2 vpc-0622a448f6aeda245 launch-wizard-2 crea
sg-0a573e2bebad23c9d - mysg
Details
Inbound rules
Outbound rules
Tags
Details
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


mysecuritygroup sg-0a573e2bebad23c9d mysg vpc-035a5c33170677427 Allow HTTP,ssh for inb...
HTTP TCP 80 0.0.0.0/0 Http from VPC
SSH TCP 22 0.0.0.0/0 SSH from VPC
NFS TCP 2049 0.0.0.0/0 NFS
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.


mysecuritygroup sg-0a573e2bebad23c9d mysg vpc-035a5c33170677427 Allow HTTP,ssh for in
Outbound rules
Edit o
Type
Protocol
Port range
Destination
Description - optional
All traffic
All
All
0.0.0.0/0
-

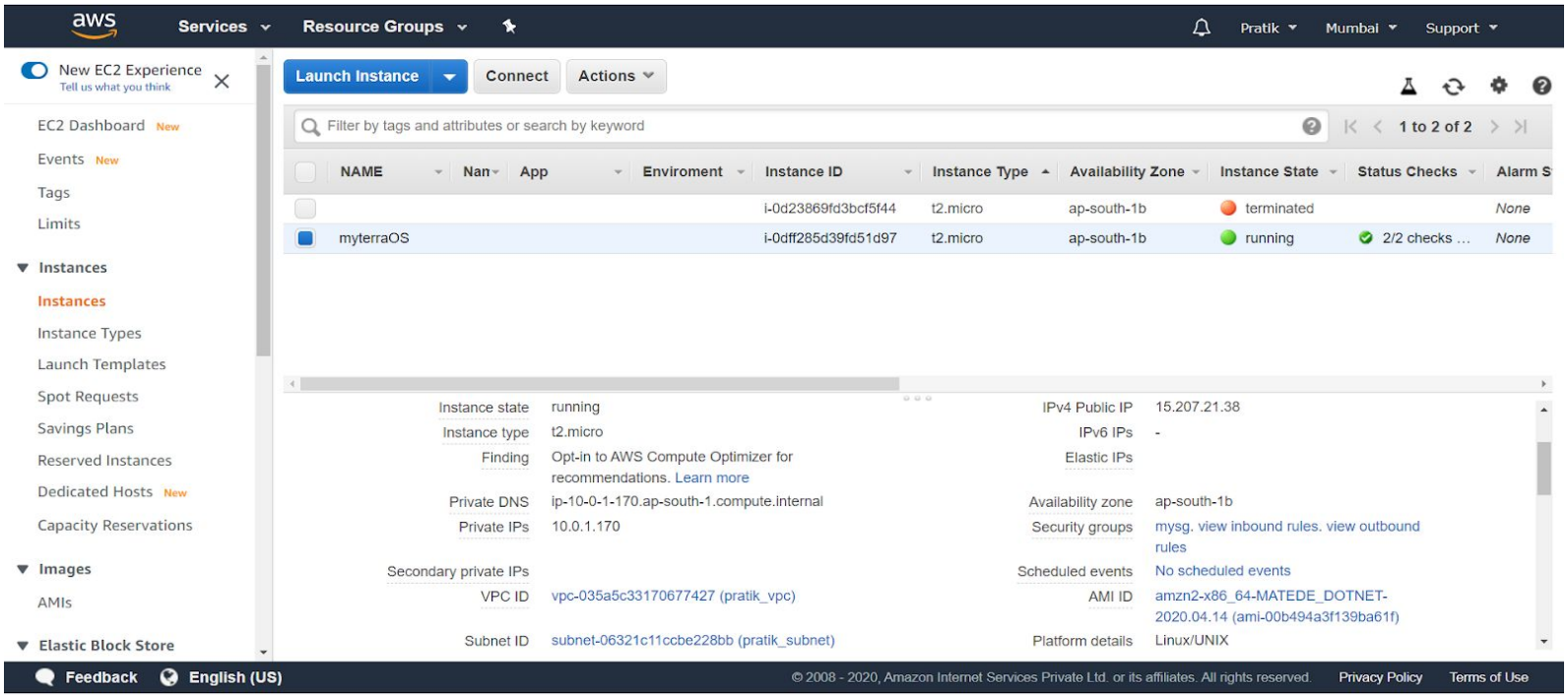
```
#Creating EFS
resource "aws_efs_file_system" "myefs" {
  creation_token = "my-efs"
  performance_mode="generalPurpose"
  tags = {
    Name = "myefs"
  }
}

resource "aws_efs_file_system_policy" "policy" {
  file_system_id = aws_efs_file_system.myefs.id
  policy = <<POLICY
{
    "Version": "2012-10-17",
    "Id": "efs-policy-wizard-c45881c9-af16-441d-aa48-0fbd68ffaf79",
    "Statement": [
        {
            "Sid": "efs-statement-20e4223c-ca0e-412d-8490-3c3980f60788",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Resource": "${aws_efs_file_system.myefs.arn}",
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
POLICY
}

#Mount EFS to EC2 instance
resource "aws_efs_mount_target" "mymount" {
  file_system_id = aws_efs_file_system.myefs.id
  subnet_id      = aws_subnet.pratik_subnet.id
  security_groups = [ "${aws_security_group.mysg.id}" ]
}
```



#Creating ec2 instance

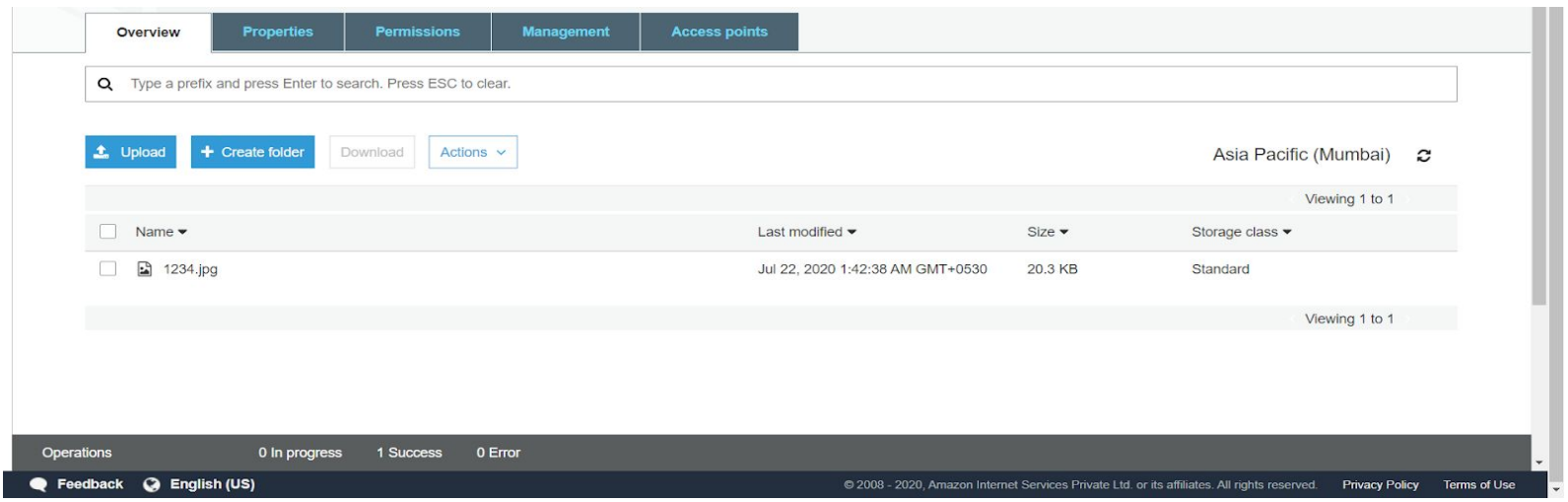
```
resource "aws_instance" "myin" {
  ami           = "ami-00b494a3f139ba61f"
  instance_type = "t2.micro"
  key_name      = "mykey1112"
  availability_zone = "ap-south-1a"
  subnet_id     = aws_subnet.pratik_subnet.id
  security_groups = [ "${aws_security_group.mysg.id}" ]
  tags = {
    Name = "myterraOS"
  }
}
resource "null_resource" "null_vol_attach"  {
  depends_on = [
    aws_efs_mount_target.mymount,
  ]
  connection {
    type     = "ssh"
    user     = "ec2-user"
    private_key = tls_private_key.key_form.private_key_pem
    host     = aws_instance.myin.public_ip
  }
  provisioner "remote-exec" {
    inline = [
      "sleep 30",
      "sudo yum install -y httpd git php amazon-efs-utils nfs-utils",
      "sudo yum install git",
      "sudo systemctl start httpd",
      "sudo systemctl enable httpd",
      "sudo chmod ugo+rw /etc/fstab",
      "sudo echo '${aws_efs_file_system.myefs.id}:/ /var/www/html efs tls,_netdev' >> /etc/fstab",
      "sudo mount -a -t efs,nfs4 defaults",
```

```
    "sudo rm -rf /var/www/html/*",
    "sudo git clone https://github.com/Pratikkohad1999/multicloud.git /var/www/html/"
  ]
 }
}
```



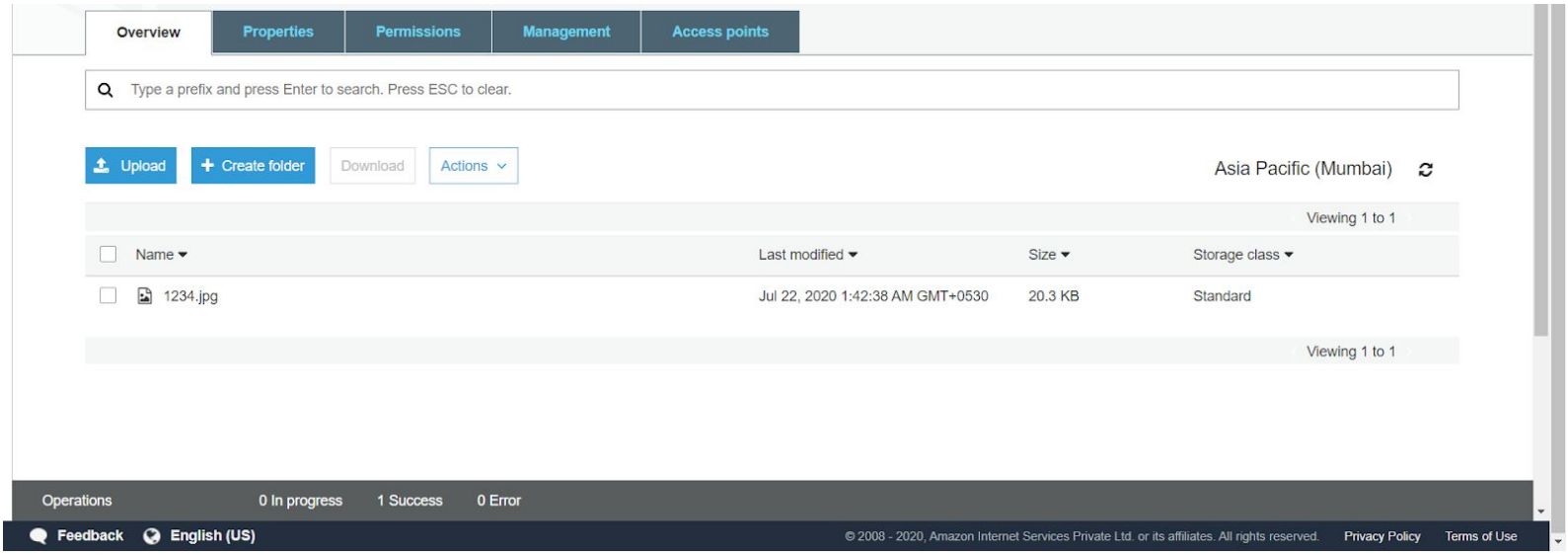
#Creating S3 Bucket

```
resource "aws_s3_bucket" "mybucket" {
  bucket = "pratik1234"
  acl    = "public-read"
  force_destroy  = true
  cors_rule {
    allowed_headers = ["*"]
    allowed_methods = ["PUT", "POST"]
    allowed_origins = ["https://pratik1234"]
    expose_headers  = ["ETag"]
    max_age_seconds = 3000
  }

depends_on = [
   null_resource.null_vol_attach,
  ]
}
```



#putting object inside the s3 bucket

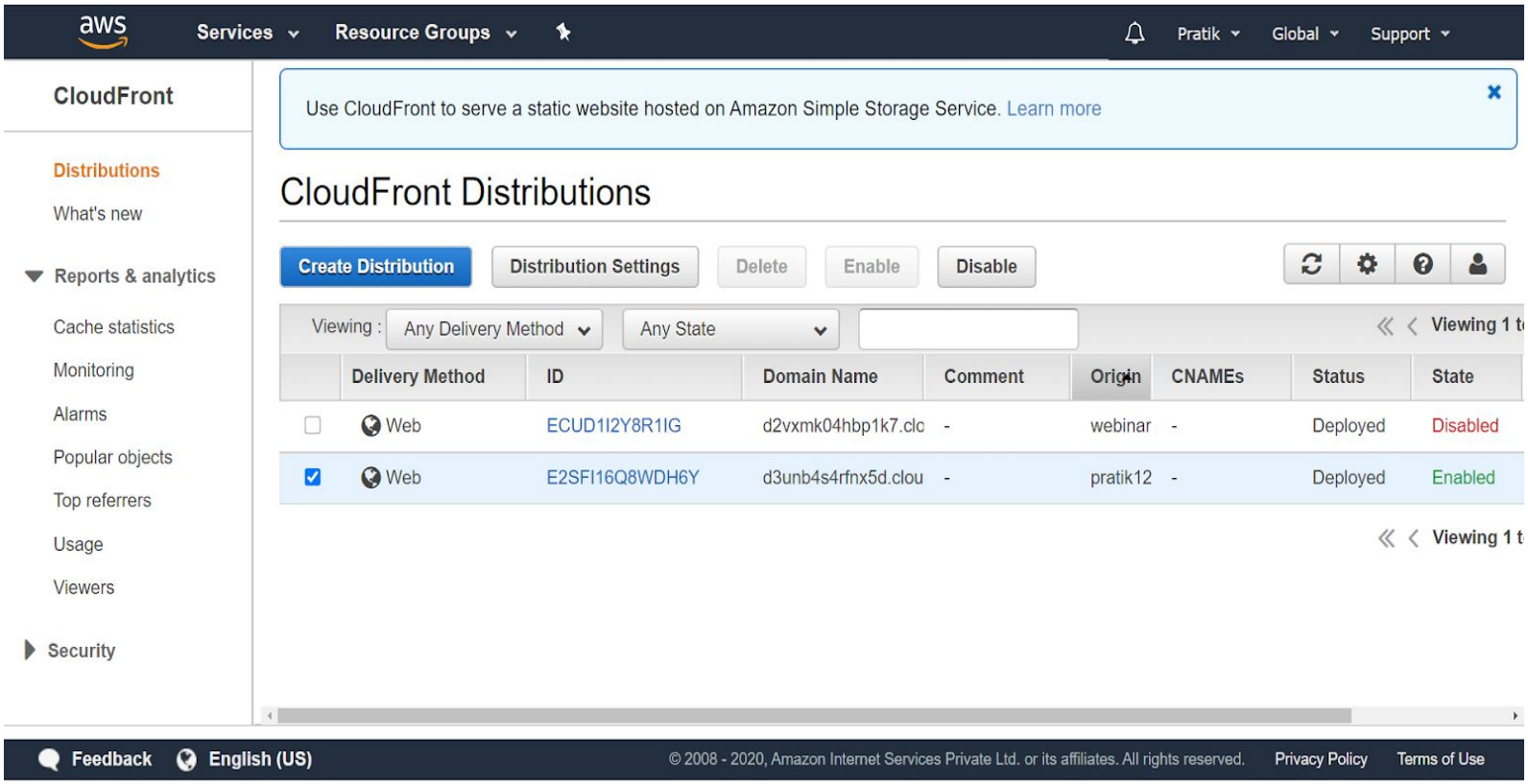
```
resource "aws_s3_bucket_object" "myobj" {
  key = "1234.jpg"
  bucket = aws_s3_bucket.mybucket.id
  source = "C:/Users/Asus/Downloads/1234.jpg"
  acl="public-read"
}
```

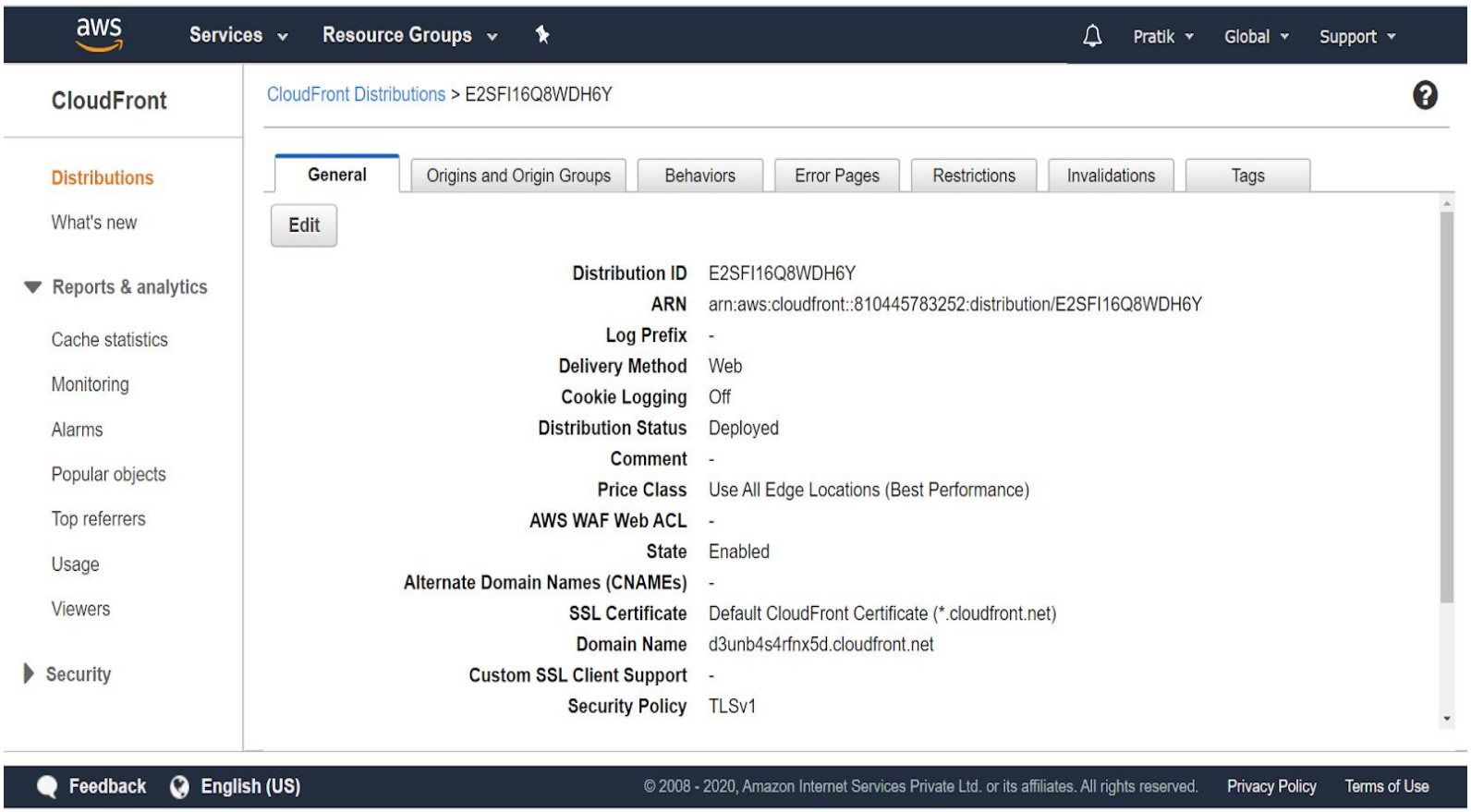


Create Cloudfront distribution

```
locals {
  s3_origin_id=aws_s3_bucket.mybucket.bucket
  image_url="$(aws_cloudfront_distribution.mycdistribution.domain_name)/$(aws_s3_bucket_object.image-pull.key)"
}

resource "aws_cloudfront_origin_access_identity" "origin_access_identity"{
   comment="Sync Cloudfront to s3"
}

resource "aws_cloudfront_distribution" "mycdistribution" {
  origin {
       domain_name=aws_s3_bucket.mybucket.bucket_regional_domain_name
        origin_id=local.s3_origin_id

  s3_origin_config {
     origin_access_identity=aws_cloudfront_origin_access_identity.origin_access_identity.cloudfront_access_identity_path
   }
}

 enabled=true
 is_ipv6_enabled=true
 default_root_object="index.php"

 custom_error_response {
  error_caching_min_ttl=3000
  error_code=404
  response_code=200
  response_page_path="/1234.jpg"
}

default_cache_behavior {
  allowed_methods = ["DELETE", "GET", "HEAD", "OPTIONS", "PATCH", "POST", "PUT"]
```

```
  cached_methods = ["GET", "HEAD"]
  target_origin_id = local.s3_origin_id

  forwarded_values {
     query_string=false
  cookies {
    forward= "none"
 }
}

viewer_protocol_policy= "allow-all"
  min_ttl= 0
  default_ttl= 3600
  max_ttl= 86400
}

 restrictions {
  geo_restriction {
     restriction_type="none"
  }
}

viewer_certificate {
  cloudfront_default_certificate=true
}

tags = {
  Name="mycdistribution"
 }

connection {
    type     = "ssh"
    user     = "ec2-user"
    private_key =  tls_private_key.key_form.private_key_pem
    host     = aws_instance.myin.public_ip
 }


  provisioner "remote-exec" {
    inline = [
       "sudo su << EOF",
       "sudo chmod ugo+rw /var/www/html/",
       "sudo echo \"<img src='http://${aws_cloudfront_distribution.mycdistribution.domain_name}/$(aws_s3_bucket_object.image-pull.key)'>\" >>
/var/www/html/index.html",
       "EOF"
    ]
  }
}


output "cloudfront_ip_addr" {
  value = aws_cloudfront_distribution.mycdistribution.domain_name
}

resource "null_resource" "key_pair"  {
            provisioner "local-exec" {
                command = "echo  '${tls_private_key.key_form.private_key_pem}' > key.pem"
            }
}
```


aws
Services
Resource Groups
Pratik
Global
Support
CloudFront
Use CloudFront to serve a static website hosted on Amazon Simple Storage Service. Learn more
Distributions
What's new
Reports & analytics
Cache statistics
Monitoring
Alarms
Popular objects
Top referrers
Usage
Viewers
Security
CloudFront Distributions
Create Distribution
Distribution Settings
Delete
Enable
Disable
Viewing :
Any Delivery Method
Any State
Viewing 1 t
Delivery Method
ID
Domain Name
Comment
Origin
CNAMEs
Status
State
Web
ECUD1I2Y8R1IG
d2vxmk04hbp1k7.clc
-
webinar
-
Deployed
Disabled
Web
E2SFI16Q8WDH6Y
d3unb4s4rfnx5d.clou
-
pratik12
-
Deployed
Enabled
Viewing 1 t
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Pratik
Global
Support
CloudFront
CloudFront Distributions > E2SFI16Q8WDH6Y
Distributions
What's new
Reports & analytics
Cache statistics
Monitoring
Alarms
Popular objects
Top referrers
Usage
Viewers
Security
General
Origins and Origin Groups
Behaviors
Error Pages
Restrictions
Invalidations
Tags
Edit
Distribution ID E2SFI16Q8WDH6Y
ARN arn:aws:cloudfront::810445783252:distribution/E2SFI16Q8WDH6Y
Log Prefix -
Delivery Method Web
Cookie Logging Off
Distribution Status Deployed
Comment -
Price Class Use All Edge Locations (Best Performance)
AWS WAF Web ACL -
State Enabled
Alternate Domain Names (CNAMEs) -
SSL Certificate Default CloudFront Certificate (*.cloudfront.net)
Domain Name d3unb4s4rfnx5d.cloudfront.net
Custom SSL Client Support -
Security Policy TLSv1
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


```
C:\Users\Asus\Desktop\terraformcodefiles>terraform destroy -auto-approve
tls_private_key.key_form: Refreshing state... [id=584f920a6c60e460a4112d056052f2c52697d31c]
null_resource.key_pair: Refreshing state... [id=1997524165343346184]
null_resource.key_pair: Destroying... [id=1997524165343346184]
null_resource.key_pair: Destruction complete after 0s
```

```
Destroy complete! Resources: 18 destroyed.
```

Thanks for Reading!!!