Summary:

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 uploaded 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 image) and use the Cloudfront URL to update in code in /var/www/html

```
provider "aws" {

region ="ap-south-1"

profile = "my_profile"



}



#Create Security group

resource "aws_security_group" "allow_tls2" {

name = "allow_tls2"

description = "Allow TLS inbound traffic"

vpc_id = "vpc-d7e8f5bf"



ingress {

description = "SSH"

from_port = 22

to_port = 22
```

```
protocol = "tcp"

cidr_blocks = ["0.0.0.0/0"]



}



ingress {

description = "TLS from VPC"

from_port = 80

to_port = 80

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 = "allow_tls2"

}

}
```

```
#Create EBS volume

resource "aws_ebs_volume" "MyVol1" {

availability_zone = "${aws_instance.myin2.availability_zone}"

size = 1

tags = {

Name = "MyVolume"

}

}


#Create EC2 instance

resource "aws_instance" "myin2" {

ami = "ami-0447a12f28fddb066"

instance_type = "t2.micro"

key_name = "mytask1Key"

security_groups = [ "allow_tls2" ]

connection {

type = "ssh"

user = "ec2-user"

private_key = file("C:/Users/ashoka/Downloads/mytask1Key.pem")

host = aws_instance.myin2.public_ip

}

provisioner "remote-exec" {

inline = [

"sudo yum install httpd php git -y",

"sudo systemctl restart httpd",

"sudo systemctl enable httpd",
```

```
]

}


tags = {

Name = "Terra 1"

}

}


#Used for configuration and mounting

resource "null_resource" "nullremote3" {


depends_on = [

aws_volume_attachment.AttachVol,

]


connection {

type = "ssh"

user = "ec2-user"

private_key = file("C:/Users/ashoka/Downloads/mytask1Key.pem")

host = aws_instance.myin2.public_ip

}

provisioner "remote-exec" {

inline = [

"sudo mkfs.ext4 /dev/xvdh",

"sudo mount /dev/xvdh /var/www/html",
```

```
"sudo rm -rf /var/www/html/*",
"sudo git clone https://github.com/ashokasmg99/terra.git /var/www/html/"

]
}
}



#Attaching EBS with EC2
resource "aws_volume_attachment" "AttachVol" {
device_name = "/dev/sdh"
volume_id = "${aws_ebs_volume.MyVol1.id}"
instance_id = "${aws_instance.myin2.id}"
depends_on = [
aws_ebs_volume.MyVol1,
aws_instance.myin2
]
}


#Creating S3 bucket
resource "aws_s3_bucket" "MyTerraformBucket" {
bucket = "bucket"
acl = "public-read"
}


#Uploading file to S3 bucket
```

```
resource "aws_s3_bucket_object" "object1" {

bucket = "bucket"

key = "image.jpg"

source = "image.jpg"

acl = "public-read"

content_type = "image/jpg"

depends_on = [

aws_s3_bucket.MyTerraformBucket

]

}


#Creating Cloud-front and attaching S3 bucket to it

resource "aws_cloudfront_distribution" "myCloudfront1" {

origin {

domain_name = "bucket.s3.amazonaws.com"

origin_id = "S3-bucket"


custom_origin_config {

http_port = 80

https_port = 80

origin_protocol_policy = "match-viewer"

origin_ssl_protocols = ["TLSv1", "TLSv1.1", "TLSv1.2"]

}

}


enabled = true
```

```
default_cache_behavior {

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

cached_methods = ["GET", "HEAD"]

target_origin_id = "S3-bucket"


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

}

depends_on = [

aws_s3_bucket_object.object1

]

 }
```