**Task 1 : Have to create/launch Application using Terraform**

**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**

**CODE:-**

provider "aws" {

region = "ap-south-1"

access\_key = "#####################"

secret\_key = "#######################"

}

**####### Creating key-pair ########**

resource "aws\_key\_pair" "my\_first\_key" {

key\_name = "my\_first\_key"

public\_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCqV9yHvZYKclXQvdZdtHIfgKFuhSUECU1axtZciDkSfuzPHc4efR4l21WVkZENKPqlfORHpjyEaJm6eX4S7QBOhKc42iOT2ijBXaVW2vUBeu3FrOCMnv4xErku5+aXsOseODwf4U6njR+wOB2EushhhtKjt2JiOjMQAL67vd95bgNu+s7XXe8bxcK4oc+ltrie5tn4XrWzZQCMYinSEgv4W/BkfiEEMtdoFCC2PAxf/f0/ysCsEdEIqfcSzbrymU5rNO5iDHY48VN2a597kqYWFV+F5TADiUZ9sq3wT4TuAcXvbz1uOa9LkNO3r70EpYBLpJb/SsmZGvxVOxnPdmF root@ip-172-31-7-29.ap-south-1.compute.internal"

}

**###### Security group creation ######**

resource "aws\_security\_group" "MYSG" {

vpc\_id = "vpc-deeef3b6"

name = "MYSG"

description = "allow ssh and http traffic"

ingress {

cidr\_blocks = ["0.0.0.0/0"]

from\_port = 80

to\_port = 80

protocol = "tcp"

}

ingress {

cidr\_blocks = ["0.0.0.0/0"]

from\_port = 22

to\_port = 22

protocol = "tcp"

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

}

**####### create instance ########**

resource "aws\_instance" "myinstance" {

ami = "ami-0447a12f28fddb066"

instance\_type = "t2.micro"

key\_name = "my\_first\_key"

availability\_zone = "ap-south-1b"

vpc\_security\_group\_ids = [

"${aws\_security\_group.MYSG.id}", ]

user\_data = <<-EOF

#!/bin/bash

sudo su

yum -y install httpd php git

cd /var/www/html

git clone <https://github.com/Yashsharma99/task>

cp task/myimg.jpg /var/www/html/

echo "<img src="donald-teel-wCIDfKBMDHE-unsplash.jpg" width="400" height="500" title="My site" alt=" My public site" >" >> /var/www/html/index.php

sudo systemctl enable httpd

sudo systemctl start httpd

EOF

tags = {

Name = "myos"

}

}

**######## create volume ######**

resource "aws\_ebs\_volume" "myvol" {

availability\_zone = "ap-south-1b"

tags = {

Name = "data volume"

}

}

**####### Attach EBS Volume ######**

resource "aws\_volume\_attachment" "gud-mrg-vol" {

device\_name = "/dev/sdh"

volume\_id = "${aws\_ebs\_volume.myvol.id}"

instance\_id = "${aws\_instance.myinstance.id}"

}

**####### create s3 bucket ###**

resource "aws\_s3\_bucket" "b" {

bucket = "mybucket.buzz"

acl = "public-read"

tags = {

Name = "My bucket"

Environment = "Dev"

}

}

**####### put object ######**

resource "aws\_s3\_bucket\_object" "obj" {

bucket = "mybucket.buzz"

key = "donald-teel-wCIDfKBMDHE-unsplash.jpg"

content\_type = "image/jpeg"

source = "C:/donald-teel-wCIDfKBMDHE-unsplash.jpg"

acl = "public-read"

}

**####### create cloudfront #######################**

resource "aws\_cloudfront\_distribution" "www\_distribution" {

// origin is where CloudFront gets its content from.

origin {

// We need to set up a "custom" origin because otherwise CloudFront won't

// redirect traffic from the root domain to the www domain, that is from

// runatlantis.io to www.runatlantis.io.

custom\_origin\_config {

// These are all the defaults.

http\_port = "80"

https\_port = "443"

origin\_protocol\_policy = "http-only"

origin\_ssl\_protocols = ["TLSv1", "TLSv1.1", "TLSv1.2"]

}

// Here we're using our S3 bucket's URL!

domain\_name = "${aws\_s3\_bucket.b.bucket\_regional\_domain\_name}"

// This can be any name to identify this origin.

origin\_id = "mydistribution"

}

enabled = true

default\_root\_object = "index.html"

// All values are defaults from the AWS console.

default\_cache\_behavior {

viewer\_protocol\_policy = "redirect-to-https"

compress = true

allowed\_methods = ["GET", "HEAD"]

cached\_methods = ["GET", "HEAD"]

// This needs to match the `origin\_id` above.

target\_origin\_id = "mydistribution"

min\_ttl = 0

default\_ttl = 86400

max\_ttl = 31536000

forwarded\_values {

query\_string = false

cookies {

forward = "none"

}

}

}

// Here we're ensuring we can hit this distribution using www.runatlantis.io

// rather than the domain name CloudFront gives us.

//aliases = ["${var.www\_domain\_name}"]

restrictions {

geo\_restriction {

restriction\_type = "none"

}

}

viewer\_certificate {

// acm\_certificate\_arn = "${aws\_acm\_certificate.certificate.arn}"

ssl\_support\_method = "sni-only"

cloudfront\_default\_certificate = true

}

}

**#Terraform command:-**

C:\Users\yash sharma.DESKTOP-9UMDR1O\Desktop\terra\task>terraform apply

aws\_ebs\_volume.myvol: Refreshing state... [id=vol-0dd12ab08f89c2cdc]

aws\_key\_pair.my\_first\_key: Refreshing state... [id=my\_first\_key]

aws\_security\_group.MYSG: Refreshing state... [id=sg-03e5869cda40161a1]

aws\_s3\_bucket\_object.obj: Refreshing state... [id=myimg.jpg]

aws\_s3\_bucket.b: Refreshing state... [id=mybucket.buzz]

aws\_instance.myinstance: Refreshing state... [id=i-03b6ffdaf3f11ed97]

aws\_volume\_attachment.gud-mrg-vol: Refreshing state... [id=vai-404537395]

aws\_cloudfront\_distribution.www\_distribution: Refreshing state... [id=E33N9U1WV2R8K]

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

~ update in-place

Terraform will perform the following actions:

# aws\_cloudfront\_distribution.www\_distribution will be updated in-place

~ resource "aws\_cloudfront\_distribution" "www\_distribution" {

active\_trusted\_signers = {

"enabled" = "false"

"items.#" = "0"

}

aliases = []

arn = "arn:aws:cloudfront::728503498903:distribution/E33N9U1WV2R8K"

caller\_reference = "terraform-20200614085446779400000001"

default\_root\_object = "index.html"

domain\_name = "dykrtevu9uhs9.cloudfront.net"

enabled = true

etag = "EOWL0TH1DMCR2"

hosted\_zone\_id = "Z2FDTNDATAQYW2"

http\_version = "http2"

id = "E33N9U1WV2R8K"

in\_progress\_validation\_batches = 0

is\_ipv6\_enabled = false

last\_modified\_time = "2020-06-14 18:00:32.708 +0000 UTC"

price\_class = "PriceClass\_All"

retain\_on\_delete = false

status = "Deployed"

tags = {}

wait\_for\_deployment = true

default\_cache\_behavior {

allowed\_methods = [

"GET",

"HEAD",

]

cached\_methods = [

"GET",

"HEAD",

]

compress = true

default\_ttl = 86400

max\_ttl = 31536000

min\_ttl = 0

smooth\_streaming = false

target\_origin\_id = "mydistribution"

trusted\_signers = []

viewer\_protocol\_policy = "redirect-to-https"

forwarded\_values {

headers = []

query\_string = false

query\_string\_cache\_keys = []

cookies {

forward = "none"

whitelisted\_names = []

}

}

}

origin {

domain\_name = "mybucket.buzz.s3.ap-south-1.amazonaws.com"

origin\_id = "mydistribution"

custom\_origin\_config {

http\_port = 80

https\_port = 443

origin\_keepalive\_timeout = 5

origin\_protocol\_policy = "http-only"

origin\_read\_timeout = 30

origin\_ssl\_protocols = [

"TLSv1",

"TLSv1.1",

"TLSv1.2",

]

}

}

restrictions {

geo\_restriction {

locations = []

restriction\_type = "none"

}

}

~ viewer\_certificate {

cloudfront\_default\_certificate = true

minimum\_protocol\_version = "TLSv1"

+ ssl\_support\_method = "sni-only"

}

}

Plan: 0 to add, 1 to change, 0 to destroy.

Warning: Interpolation-only expressions are deprecated

on main.tf line 102, in resource "aws\_volume\_attachment" "gud-mrg-vol":

102: volume\_id = "${aws\_ebs\_volume.myvol.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.

(and 2 more similar warnings elsewhere)

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

aws\_cloudfront\_distribution.www\_distribution: Modifying... [id=E33N9U1WV2R8K]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 10s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 20s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 30s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 40s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 50s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 1m0s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 1m10s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 1m20s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 1m30s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 1m40s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 1m50s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 2m0s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 2m11s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 2m21s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 2m31s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 2m41s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 2m51s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 3m1s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 3m11s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 3m21s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 3m31s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Still modifying... [id=E33N9U1WV2R8K, 3m41s elapsed]

aws\_cloudfront\_distribution.www\_distribution: Modifications complete after 3m41s [id=E33N9U1WV2R8K]

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

C:\Users\yash sharma.DESKTOP-9UMDR1O\Desktop\terra\task>