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

**Initializing: -**

C:\Users\HP\Desktop\tera\extra>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 = "~> 2.66"

Warning: Interpolation-only expressions are deprecated

 on finall.tf line 48, in resource "aws\_volume\_attachment" "taskattach":

 48: volume\_id = "${aws\_ebs\_volume.taskebs.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 8 more similar warnings elsewhere)

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.

**Planning: -**

C:\Users\HP\Desktop\tera\extra>terraform plan

Refreshing Terraform state in-memory prior to plan...

The refreshed state will be used to calculate this plan, but will not be

persisted to local or remote state storage.

------------------------------------------------------------------------

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.s3\_distribution will be created

 + resource "aws\_cloudfront\_distribution" "s3\_distribution" {

     + active\_trusted\_signers        = (known after apply)

     + arn                           = (known after apply)

     + caller\_reference              = (known after apply)

     + comment                       = "Some comment"

     + 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                          = {

         + "Environment" = "production"

       }

     + wait\_for\_deployment           = true

     + default\_cache\_behavior {

         + allowed\_methods       = [

             + "DELETE",

             + "GET",

             + "HEAD",

             + "OPTIONS",

             + "PATCH",

             + "POST",

             + "PUT",

           ]

         + cached\_methods        = [

             + "GET",

             + "HEAD",

           ]

         + compress              = false

         + default\_ttl           = 86400

         + max\_ttl               = 31536000

         + min\_ttl               = 0

         + target\_origin\_id      = "myS3Origin"

         + viewer\_protocol\_policy = "allow-all"

         + forwarded\_values {

             + query\_string = false

             + cookies {

                 + forward = "none"

               }

           }

       }

     + origin {

         + domain\_name = (known after apply)

         + origin\_id  = "myS3Origin"

         + s3\_origin\_config {

             + origin\_access\_identity = (known after apply)

           }

       }

     + restrictions {

         + geo\_restriction {

             + locations       = [

                 + "CA",

                 + "GB",

                 + "IN",

                 + "US",

               ]

             + restriction\_type = "whitelist"

           }

       }

     + 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                        = "Some comment"

     + etag                           = (known after apply)

     + iam\_arn                        = (known after apply)

     + id                             = (known after apply)

     + s3\_canonical\_user\_id           = (known after apply)

   }

 # aws\_codepipeline.codepipeline will be created

 + resource "aws\_codepipeline" "codepipeline" {

     + arn     = (known after apply)

     + id      = (known after apply)

     + name    = "srishti-pipeline"

     + role\_arn = (known after apply)

     + artifact\_store {

         + location = "srishtibalti"

         + region  = (known after apply)

         + type    = "S3"

       }

     + stage {

         + name = "Source"

         + action {

             + category        = "Source"

             + configuration   = {

                 + "Branch"    = "master"

                 + "OAuthToken" = "a962d286561fb7eeae716f2ecee9d258ac141042"

                 + "Owner"     = "srish072"

                 + "Repo"      = "task1"

               }

             + name            = "Source"

             + output\_artifacts = [

                 + "SourceArtifacts",

               ]

             + owner           = "ThirdParty"

             + provider        = "GitHub"

             + region          = (known after apply)

             + run\_order       = (known after apply)

             + version         = "1"

           }

       }

     + stage {

         + name = "Deploy"

         + action {

             + category       = "Deploy"

             + configuration  = {

                 + "BucketName" = "srishtibalti"

                 + "Extract"   = "true"

               }

             + input\_artifacts = [

                 + "SourceArtifacts",

               ]

             + name           = "Deploy"

             + owner          = "AWS"

             + provider       = "S3"

             + region         = (known after apply)

             + run\_order      = (known after apply)

             + version        = "1"

           }

       }

   }

 # aws\_ebs\_volume.taskebs will be created

 + resource "aws\_ebs\_volume" "taskebs" {

     + arn              = (known after apply)

     + availability\_zone = "ap-south-1a"

     + encrypted        = (known after apply)

     + id               = (known after apply)

     + iops             = (known after apply)

     + kms\_key\_id       = (known after apply)

     + size             = 1

     + snapshot\_id      = (known after apply)

     + tags             = {

         + "Name" = "taskebs"

       }

     + type             = (known after apply)

   }

 # aws\_iam\_role.codepipeline\_role will be created

 + resource "aws\_iam\_role" "codepipeline\_role" {

     + arn                  = (known after apply)

     + assume\_role\_policy   = jsonencode(

           {

             + Statement = [

                 + {

                     + Action   = "sts:AssumeRole"

                     + Effect   = "Allow"

                     + Principal = {

                         + Service = "codepipeline.amazonaws.com"

                       }

                   },

               ]

             + Version  = "2012-10-17"

           }

       )

     + create\_date          = (known after apply)

     + force\_detach\_policies = false

     + id                   = (known after apply)

     + max\_session\_duration = 3600

     + name                 = "task"

     + path                 = "/"

     + unique\_id            = (known after apply)

   }

 # aws\_iam\_role\_policy.codepipeline\_policy will be created

 + resource "aws\_iam\_role\_policy" "codepipeline\_policy" {

     + id    = (known after apply)

     + name  = "codepipeline\_policy"

     + policy = (known after apply)

     + role  = (known after apply)

   }

 # aws\_instance.taskinst will be created

 + resource "aws\_instance" "taskinst" {

     + ami                         = "ami-0447a12f28fddb066"

     + 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                    = "task-key1"

     + 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             = [

         + "task-sg1",

       ]

     + source\_dest\_check           = true

     + subnet\_id                   = (known after apply)

     + tags                        = {

         + "Name" = "task-inst"

       }

     + tenancy                     = (known after apply)

     + user\_data                   = "aef23c3c751f0c2435c08554a2edd4358746f557"

     + 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)

         + 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\_key\_pair.taskkey will be created

 + resource "aws\_key\_pair" "taskkey" {

     + fingerprint = (known after apply)

     + id         = (known after apply)

     + key\_name   = "task-key1"

     + key\_pair\_id = (known after apply)

     + public\_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC2ThbwkFgnI/aAzOVSBsY2TjaLRUnGnyzylxG+tp46vxJInDcFn83TRjY5WojaA4A8eHjJFtpJdNYQw8Z8wMznjDSnefZZOqVYVAhl61j4C0YVhlCY+74rJCu7ms8Jhn7heMIwUUxWHdSr4eL+JoUg7E4+XUH4olvDtghLn/ddBabGIk3dQYlsOME4Ri1xFkaggKvMBOXw/2NMBqgsTG65duUG5kzYMf/zL69ZAxUxtgpVhlsoiqOAV9Vh70CQq8nZDRJ59EaQ1knBwxOFbcRczRvyCWna6MFdaVrIAfqJC/H2la3qcLKqMbYRrZAjHLubuAcC4kqUg6aldiQrVVyP"

   }

 # aws\_s3\_bucket.srishtibalti will be created

 + resource "aws\_s3\_bucket" "srishtibalti" {

     + acceleration\_status        = (known after apply)

     + acl                        = "public-read"

     + arn                        = (known after apply)

     + bucket                     = "srishtibalti"

     + bucket\_domain\_name         = (known after apply)

     + bucket\_regional\_domain\_name = (known after apply)

     + force\_destroy              = false

     + hosted\_zone\_id             = (known after apply)

     + id                         = (known after apply)

     + region                     = (known after apply)

     + request\_payer              = (known after apply)

     + tags                       = {

         + "Name" = "srishtibalti"

       }

     + website\_domain             = (known after apply)

     + website\_endpoint           = (known after apply)

     + versioning {

         + enabled   = (known after apply)

         + mfa\_delete = (known after apply)

       }

   }

 # aws\_security\_group.tasksg will be created

 + resource "aws\_security\_group" "tasksg" {

     + arn                   = (known after apply)

     + description           = "Allow TLS 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\_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     = "SSH"

             + from\_port       = 22

             + ipv6\_cidr\_blocks = []

             + prefix\_list\_ids = []

             + protocol        = "tcp"

             + security\_groups = []

             + self            = false

             + to\_port         = 22

           },

       ]

     + name                  = "task-sg1"

     + owner\_id              = (known after apply)

     + revoke\_rules\_on\_delete = false

     + tags                  = {

         + "Name" = "tasksg"

       }

     + vpc\_id                = "vpc-33465a5b"

   }

 # aws\_volume\_attachment.taskattach will be created

 + resource "aws\_volume\_attachment" "taskattach" {

     + device\_name = "/dev/sdf"

     + id         = (known after apply)

     + instance\_id = (known after apply)

     + volume\_id  = (known after apply)

   }

Plan: 11 to add, 0 to change, 0 to destroy.

Warning: Interpolation-only expressions are deprecated

 on finall.tf line 48, in resource "aws\_volume\_attachment" "taskattach":

 48: volume\_id = "${aws\_ebs\_volume.taskebs.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 8 more similar warnings elsewhere)

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform

can't guarantee that exactly these actions will be performed if

"terraform apply" is subsequently run.

**Running terraform: -**

C:\Users\HP\Desktop\tera\extra>terraform apply -auto-approve

aws\_ebs\_volume.taskebs: Creating...

aws\_key\_pair.taskkey: Creating...

aws\_cloudfront\_origin\_access\_identity.origin\_access\_identity: Creating...

aws\_iam\_role.codepipeline\_role: Creating...

aws\_instance.taskinst: Creating...

aws\_security\_group.tasksg: Creating...

aws\_s3\_bucket.srishtibalti: Creating...

aws\_key\_pair.taskkey: Creation complete after 2s [id=task-key1]

aws\_iam\_role.codepipeline\_role: Creation complete after 3s [id=task]

aws\_cloudfront\_origin\_access\_identity.origin\_access\_identity: Creation complete after 3s [id=E77KI2VUEXKOS]

aws\_security\_group.tasksg: Creation complete after 5s [id=sg-0559409c9adbf562b]

aws\_s3\_bucket.srishtibalti: Creation complete after 9s [id=srishtibalti]

aws\_iam\_role\_policy.codepipeline\_policy: Creating...

aws\_codepipeline.codepipeline: Creating...

aws\_cloudfront\_distribution.s3\_distribution: Creating...

aws\_ebs\_volume.taskebs: Still creating... [10s elapsed]

aws\_instance.taskinst: Still creating... [10s elapsed]

aws\_iam\_role\_policy.codepipeline\_policy: Creation complete after 3s [id=task:codepipeline\_policy]

aws\_ebs\_volume.taskebs: Creation complete after 12s [id=vol-03868d3c94a9cc889]

aws\_codepipeline.codepipeline: Creation complete after 5s [id=srishti-pipeline]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [10s elapsed]

aws\_instance.taskinst: Still creating... [20s elapsed]

aws\_instance.taskinst: Creation complete after 27s [id=i-000e8118c8ff6dfa7]

aws\_volume\_attachment.taskattach: Creating...

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [20s elapsed]

aws\_volume\_attachment.taskattach: Still creating... [10s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [30s elapsed]

aws\_volume\_attachment.taskattach: Still creating... [20s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [40s elapsed]

aws\_volume\_attachment.taskattach: Creation complete after 24s [id=vai-139154386]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [50s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [1m0s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [1m10s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [1m20s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [1m30s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [1m40s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [1m50s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [2m0s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [2m10s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [2m20s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [2m30s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [2m40s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [2m50s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [3m0s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [3m10s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [3m20s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Still creating... [3m30s elapsed]

aws\_cloudfront\_distribution.s3\_distribution: Creation complete after 3m38s [id=E15QAYT6BYK6ID]

Warning: Interpolation-only expressions are deprecated

 on finall.tf line 48, in resource "aws\_volume\_attachment" "taskattach":

 48: volume\_id = "${aws\_ebs\_volume.taskebs.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 8 more similar warnings elsewhere)

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.

**Outputs: -**



















