**Task4:**

# Terraform Configuration for Task-4

# Activities: Static Website (ByteWave Solutions) + EC2 Dev Server (CloudNova Inc.)

provider "aws" {

  region     = "us-east-1"

  access\_key = var.access\_key

  secret\_key = var.secret\_key

}

###############################################

# Activity 1: S3 Static Website – ByteWave

###############################################

resource "aws\_s3\_bucket" "bytewave\_bucket" {

  bucket = "bytewave-website-as"  # Replace 'as' with your initials if needed

}

resource "aws\_s3\_bucket\_website\_configuration" "website\_config" {

  bucket = aws\_s3\_bucket.bytewave\_bucket.bucket

  index\_document {

    suffix = "index.html"

  }

}

resource "aws\_s3\_bucket\_public\_access\_block" "block\_public" {

  bucket = aws\_s3\_bucket.bytewave\_bucket.id

  block\_public\_acls       = true

  block\_public\_policy     = false

  ignore\_public\_acls      = true

  restrict\_public\_buckets = false

}

resource "aws\_s3\_bucket\_policy" "public\_read\_policy" {

  bucket = aws\_s3\_bucket.bytewave\_bucket.id

  policy = jsonencode({

    Version = "2012-10-17",

    Statement = [

      {

        Sid       = "PublicReadGetObject",

        Effect    = "Allow",

        Principal = "\*",

        Action    = ["s3:GetObject"],

        Resource  = ["${aws\_s3\_bucket.bytewave\_bucket.arn}/\*"]

      }

    ]

  })

}

output "s3\_website\_endpoint" {

  value = aws\_s3\_bucket.bytewave\_bucket.website\_endpoint

  description = "S3 static website URL"

}

###############################################

# Activity 2: EC2 Dev Server – CloudNova Inc.

###############################################

resource "aws\_security\_group" "dev\_sg" {

  name        = "dev-server-sg"

  description = "Allow SSH access"

  ingress {

    from\_port   = 22

    to\_port     = 22

    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"]

  }

}

resource "aws\_instance" "dev\_server" {

  ami           = "ami-0c02fb55956c7d316"  # Amazon Linux 2 (us-east-1)

  instance\_type = "t2.micro"

  key\_name      = var.key\_name

  security\_groups = [aws\_security\_group.dev\_sg.name]

  user\_data = <<-EOF

              #!/bin/bash

              yum update -y

              yum install -y python3 git

              EOF

  tags = {

    Name = "CloudNovaDevServer"

  }

}

output "ec2\_public\_ip" {

  value       = aws\_instance.dev\_server.public\_ip

  description = "Public IP of the Dev EC2 instance"

}

Variables.tf

variable "access\_key" {}

variable "secret\_key" {}

variable "key\_name" {}

terraform.tfvars

access\_key = "AKIAXEVXYM5N27AVWKFB"

secret\_key = "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"

key\_name   = "anjali-key"

# aws\_instance.dev\_server will be created

+ resource "aws\_instance" "dev\_server" {

+ ami = "ami-0c02fb55956c7d316"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

+ disable\_api\_stop = (known after apply)

+ disable\_api\_termination = (known after apply)

+ ebs\_optimized = (known after apply)

+ enable\_primary\_ipv6 = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ host\_resource\_group\_arn = (known after apply)

+ iam\_instance\_profile = (known after apply)

+ id = (known after apply)

+ instance\_initiated\_shutdown\_behavior = (known after apply)

+ instance\_lifecycle = (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 = "anjali-key"

+ monitoring = (known after apply)

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ placement\_partition\_number = (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)

+ region = "eu-north-1"

+ secondary\_private\_ips = (known after apply)

+ security\_groups = [

+ "dev-server-sg",

]

+ source\_dest\_check = true

+ spot\_instance\_request\_id = (known after apply)

+ subnet\_id = (known after apply)

+ tags = {

+ "Name" = "CloudNovaDevServer"

}

+ tags\_all = {

+ "Name" = "CloudNovaDevServer"

}

+ tenancy = (known after apply)

+ user\_data = <<-EOT

#!/bin/bash

yum update -y

yum install -y python3 git

EOT

+ user\_data\_base64 = (known after apply)

+ user\_data\_replace\_on\_change = false

+ vpc\_security\_group\_ids = (known after apply)

+ capacity\_reservation\_specification (known after apply)

+ cpu\_options (known after apply)

+ ebs\_block\_device (known after apply)

+ enclave\_options (known after apply)

+ ephemeral\_block\_device (known after apply)

+ instance\_market\_options (known after apply)

+ maintenance\_options (known after apply)

+ metadata\_options (known after apply)

+ network\_interface (known after apply)

+ private\_dns\_name\_options (known after apply)

+ root\_block\_device (known after apply)

}

# aws\_s3\_bucket.bytewave\_bucket will be created

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

+ acceleration\_status = (known after apply)

+ acl = (known after apply)

+ arn = (known after apply)

+ bucket = "bytewave-website-as"

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

+ bucket\_prefix = (known after apply)

+ bucket\_region = (known after apply)

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

+ force\_destroy = false

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

+ id = (known after apply)

+ object\_lock\_enabled = (known after apply)

+ policy = (known after apply)

+ region = "eu-north-1"

+ request\_payer = (known after apply)

+ tags\_all = (known after apply)

+ website\_domain = (known after apply)

+ website\_endpoint = (known after apply)

+ cors\_rule (known after apply)

+ grant (known after apply)

+ lifecycle\_rule (known after apply)

+ logging (known after apply)

+ object\_lock\_configuration (known after apply)

+ replication\_configuration (known after apply)

+ server\_side\_encryption\_configuration (known after apply)

+ versioning (known after apply)

+ website (known after apply)

}

# aws\_s3\_bucket\_policy.public\_read\_policy will be created

+ resource "aws\_s3\_bucket\_policy" "public\_read\_policy" {

+ bucket = (known after apply)

+ id = (known after apply)

+ policy = (known after apply)

+ region = "eu-north-1"

}

# aws\_s3\_bucket\_public\_access\_block.block\_public will be created

+ resource "aws\_s3\_bucket\_public\_access\_block" "block\_public" {

+ block\_public\_acls = true

+ block\_public\_policy = false

+ bucket = (known after apply)

+ id = (known after apply)

+ ignore\_public\_acls = true

+ region = "eu-north-1"

+ restrict\_public\_buckets = false

}

# aws\_s3\_bucket\_website\_configuration.website\_config will be created

+ resource "aws\_s3\_bucket\_website\_configuration" "website\_config" {

+ bucket = "bytewave-website-as"

+ id = (known after apply)

+ region = "eu-north-1"

+ routing\_rules = (known after apply)

+ website\_domain = (known after apply)

+ website\_endpoint = (known after apply)

+ index\_document {

+ suffix = "index.html"

}

+ routing\_rule (known after apply)

}

Plan: 5 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ ec2\_public\_ip = (known after apply)

+ s3\_website\_endpoint = (known after apply)

╷

│ Warning: Deprecated attribute

│

│ on main.tf line 51, in output "s3\_website\_endpoint":

│ 51: value = aws\_s3\_bucket.bytewave\_bucket.website\_endpoint

│

│ The attribute "website\_endpoint" is deprecated. Refer to the provider documentation for details.

│

│ (and one more similar warning elsewhere)

Plan: 5 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ ec2\_public\_ip = (known after apply)

+ s3\_website\_endpoint = (known after apply)

╷

│ Warning: Deprecated attribute

│

│ on main.tf line 51, in output "s3\_website\_endpoint":

│ 51: value = aws\_s3\_bucket.bytewave\_bucket.website\_endpoint

│

│ The attribute "website\_endpoint" is deprecated. Refer to the provider documentation for details.

│

│ (and one more similar warning 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\_s3\_bucket.bytewave\_bucket: Creating...

aws\_instance.dev\_server: Creating...

aws\_instance.dev\_server: Still creating... [00m10s elapsed]

aws\_instance.dev\_server: Still creating... [00m20s elapsed]

aws\_instance.dev\_server: Still creating... [00m30s elapsed]

aws\_instance.dev\_server: Creation complete after 37s [id=i-0173842008ef4214d]

╷

│ Error: creating S3 Bucket (bytewave-website-as): operation error S3: CreateBucket, https response error StatusCode: 409, RequestID: WC2RW7VVTKZP1SCQ, HostID: YT44ATBALH1p/IrJXKsARlxLvvE5+2ioBxjU75y156/t/6ZwqG6Ab7j8xn6VZla/i0DSIGf0oMA=, BucketAlreadyExists:

│

│ with aws\_s3\_bucket.bytewave\_bucket,

│ on main.tf line 13, in resource "aws\_s3\_bucket" "bytewave\_bucket":

│ 13: resource "aws\_s3\_bucket" "bytewave\_bucket" {