

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.bythewave_bucket.id
  policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Sid    = "PublicReadGetObject",
        Effect = "Allow",
        Principal = "*",
        Action  = ["s3:GetObject"],
        Resource = ["${aws_s3_bucket.bythewave_bucket.arn}/*"]
      }
    ]
  })
}

```

```

output "s3_website_endpoint" {
  value = aws_s3_bucket.bythewave_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 = "AKIAEVXYM5N27AVWKFB"
secret_key = "kuU6YgWlOmtnxG3sbbrYYvzujnUpm+FvcAapI3P8"
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.bythewave_bucket will be created

+ resource "aws_s3_bucket" "bythewave_bucket" {

+ acceleration_status = (known after apply)

+ acl = (known after apply)

+ arn = (known after apply)

+ bucket = "bythewave-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.bythewave_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.bythewave_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/IrJXKsARlxLvE5+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" {