**Lab Activity 5**

**Name: Nirja Upendrabhai Patel**

**Stdent ID: 100965379**

**1.resource created on GCP with terraform.**

**GCP-VOC.tf**

**provider "google" {**

**project = "dynamic-moment-441416-m2"**

**region = "us-east1"**

**credentials=("dynamic-moment-441416-m2-9aa348fdd5bc.json")**

**}**

**resource "google\_compute\_network" "patel\_vpc" {**

**name = "patel-vpc-gcp"**

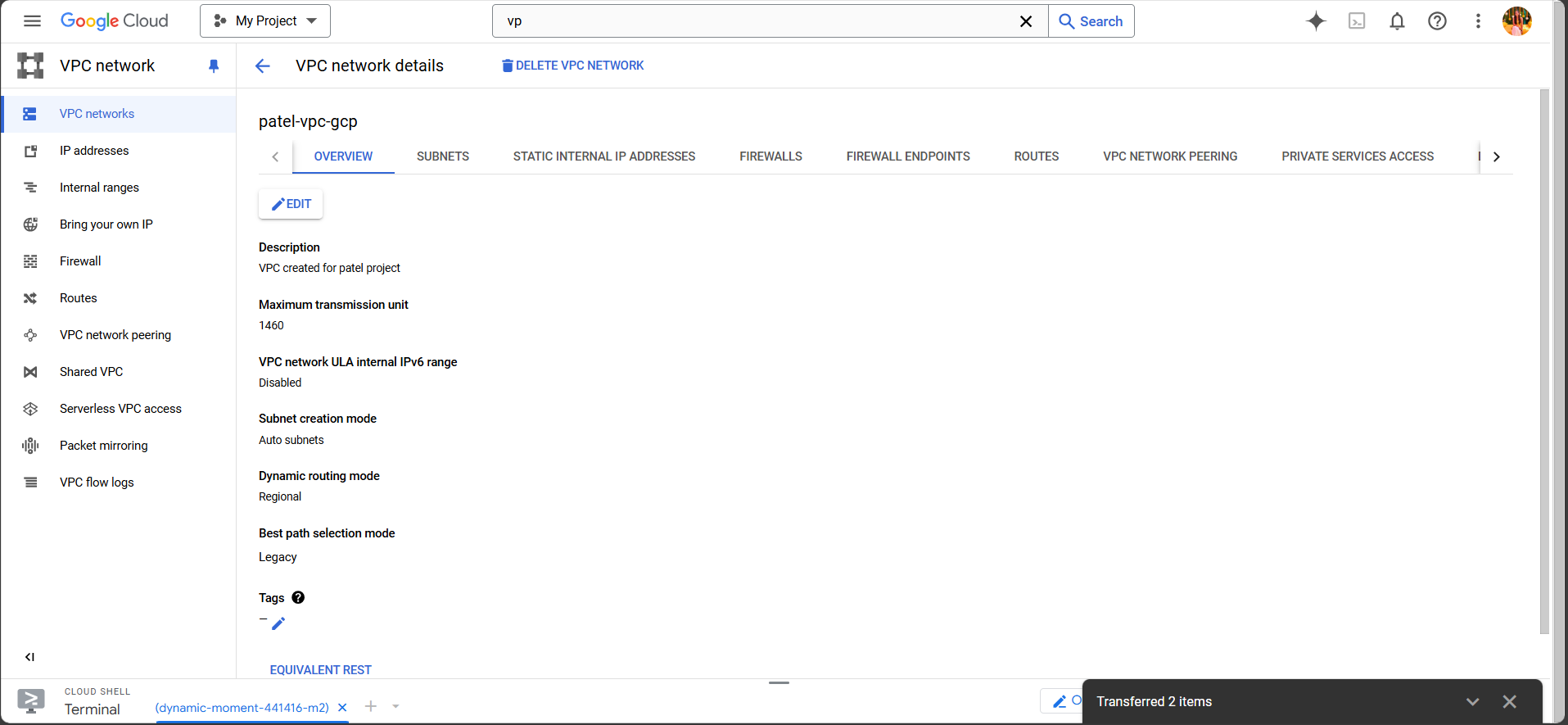
**auto\_create\_subnetworks = true**

**description = "VPC created for patel project"**

**}**

**Output:**

**VPC**

****

**Subnet**

**A screenshot of a computer

Description automatically generated**

**2.resource created on AWS with terraform. Aws-vpc.tf**

**provider "aws" {**

**region = "us-east-1"**

**}**

**resource "aws\_vpc" "patel\_vpc" {**

**cidr\_block = "10.0.0.0/16"**

**tags = {**

**Name = "patel-vpc-aws"**

**}**

**}**

**VPC:**

**A screenshot of a computer

Description automatically generated**

**3. resource created on Azure with terraform**

**Azure-Vnet.tf**

**provider "azurerm" {**

**features {}**

**subscription\_id = "f1dfb7e3-f606-4c89-a299-1286ae7c6052"**

**}**

**resource "azurerm\_resource\_group" "patel\_rg" {**

**name = "patel-resource-group"**

**location = "East US"**

**}**

**resource "azurerm\_virtual\_network" "patel\_vnet" {**

**name = "patel-vnet-azure"**

**address\_space = ["10.0.0.0/16"]**

**location = azurerm\_resource\_group.patel\_rg.location**

**resource\_group\_name = azurerm\_resource\_group.patel\_rg.name**

**}**

**Vnet:**

**A screenshot of a computer

Description automatically generated**

**6. Completed terraform script**

**terraform {**

**required\_providers {**

**aws = {**

**source = "hashicorp/aws"**

**version = "~> 4.16"**

**}**

**}**

**required\_version = ">= 1.2.0"**

**}**

**provider "aws" {**

**region = "ca-central-1"**

**}**

**resource "aws\_s3\_bucket" "mybucket" {**

**bucket = "patel-lab5-bucket"**

**}**

**resource "aws\_s3\_bucket\_public\_access\_block" "mybucket\_access\_block" {**

**bucket = aws\_s3\_bucket.mybucket.id**

**block\_public\_acls = false**

**block\_public\_policy = false**

**ignore\_public\_acls = false**

**restrict\_public\_buckets = false**

**}**

**resource "aws\_s3\_bucket\_website\_configuration" "mybucket\_website" {**

**bucket = aws\_s3\_bucket.mybucket.id**

**index\_document {**

**suffix = "index.html"**

**}**

**error\_document {**

**key = "error.html"**

**}**

**}**

**resource "aws\_s3\_bucket\_ownership\_controls" "mybucket\_ownership" {**

**bucket = aws\_s3\_bucket.mybucket.id**

**rule {**

**object\_ownership = "BucketOwnerPreferred"**

**}**

**}**

**resource "aws\_s3\_bucket\_acl" "mybucket\_acl" {**

**bucket = aws\_s3\_bucket.mybucket.id**

**acl = "public-read"**

**depends\_on = [**

**aws\_s3\_bucket\_ownership\_controls.mybucket\_ownership,**

**aws\_s3\_bucket\_public\_access\_block.mybucket\_access\_block**

**]**

**}**

**resource "aws\_s3\_bucket\_policy" "mybucket\_policy" {**

**bucket = aws\_s3\_bucket.mybucket.id**

**policy = jsonencode({**

**Version = "2012-10-17"**

**Statement = [**

**{**

**Sid = "PublicReadGetObject"**

**Effect = "Allow"**

**Principal = "\*"**

**Action = "s3:GetObject"**

**Resource = [**

**aws\_s3\_bucket.mybucket.arn,**

**"${aws\_s3\_bucket.mybucket.arn}/\*",**

**]**

**},**

**]**

**})**

**depends\_on = [**

**aws\_s3\_bucket\_public\_access\_block.mybucket\_access\_block**

**]**

**}**

**resource "aws\_cloudfront\_distribution" "mycdn" {**

**origin {**

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

**origin\_id = "S3-${aws\_s3\_bucket.mybucket.id}"**

**}**

**default\_cache\_behavior {**

**target\_origin\_id = "S3-${aws\_s3\_bucket.mybucket.id}"**

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

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

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

**default\_ttl = 3600**

**max\_ttl = 86400**

**min\_ttl = 0**

**forwarded\_values {**

**query\_string = false**

**cookies {**

**forward = "none"**

**}**

**}**

**}**

**restrictions {**

**geo\_restriction {**

**restriction\_type = "none"**

**}**

**}**

**viewer\_certificate {**

**cloudfront\_default\_certificate = true**

**}**

**enabled = true**

**}**

**7. S3 section of the script is functional**

**A screenshot of a computer

Description automatically generated**

**8. CDN Section of the script is working**

**A screenshot of a computer

Description automatically generated**

**9. An index.html webpage successfully loads from your CDN when the instructor executes the script and adds an index.html to the created S3 bucket.**

A screenshot of a computer

Description automatically generated