

ANIRUDH CHAUDHARY

500061147

R171217008

SPCM ASSIGNMENT 1

TASK DESCRIPTION

Write Terraform script to do perform following tasks on AWS cloud Platform

- Step 1: Create two T2 Micro EC2 Instances.
- Step2: Create a VPN on AWS
- Step 3: Create a S3 Bucket
- Step 4: Write the code for step 1,2 and 3 in a IaC terraform file and run terraform commands to execute these steps.

THEORY:

- AWS EC2 Instance: An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. EC2 is a service that allows business subscribers to run application programs in the computing environment. Instances are created from Amazon Machine Images (AMI). The machine images are like templates that are configured with an operating system and other software, which determine the user's operating environment. Users can select an AMI provided by AWS, the user community, or through the AWS Marketplace. Users can also create their own AMIs and share them.
- AWS VPN: AWS Client VPN is a fully-managed, elastic VPN service that automatically scales up or down based on user demand. Because it is a cloud VPN solution, you don't need to install and manage hardware or software-based solutions or try to estimate how many remote users to support at one time.
- AWS S3: Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface. Amazon S3 uses the same scalable storage infrastructure that Amazon.com uses to run its global ecommerce network.

TERRAFORM SCRIPTS

```
es Text Editor
Fri 08:58
*main.tf
Save

l.txt x *main.tf x

provider "aws" {
  region = "ap-south-1"
  profile = "Anirudh"
}

resource "aws_instance" "basic0s" {
  count      = "2"
  ami       = "ami-0447a12f28fddb066"
  instance_type = "t2.micro"
  key_name   = "Key_Anirudh"
  security_groups = ["basicSecurity"]
  tags = {
    Name = "first0s -${ count.index + 1}"
  }
}

resource "aws_s3_bucket" "mytestbucket125" {
  bucket = "mytestbucket456789"
  acl    = "public-read"
  tags = {
    Name = "testbucket456789"
  }
  versioning {
    enabled = true
  }
}

Plain Text Tab Width: 8 Ln 11, Col 38 INS
```

VPN CREATION SCRIPT

```
es Text Editor
Fri 08:59
*main.tf
Save

l.txt x *main.tf x

variable "vpc_private_subnets" {
  type     = list(string)
  default = ["10.10.11.0/24", "10.10.12.0/24", "10.10.13.0/24"]
}

module "vpn_gateway" {
  source = "../.."

  create_vpn_connection = false

  vpn_gateway_id      = module.vpc.vgw_id
  customer_gateway_id = aws_customer_gateway.main.id

  vpc_id                = module.vpc.vpc_id
  vpc_subnet_route_table_ids = module.vpc.private_route_table_ids
  vpc_subnet_route_table_count = length(var.vpc_private_subnets)
}

resource "aws_customer_gateway" "main" {
  bgp_asn      = 65000
  ip_address   = "172.83.124.12"
  type         = "ipsec.1"

  tags = {
    Name = "minimal-vpn-gateway"
  }
}
```

```

module "vpc" {
  source = "terraform-aws-modules/vpc/aws"
  version = "~> 2.0"

  name = "minimal-vpn-gateway"

  cidr = "10.10.0.0/16"

  azs          = ["eu-west-1a", "eu-west-1b", "eu-west-1c"]
  public_subnets = ["10.10.1.0/24", "10.10.2.0/24", "10.10.3.0/24"]
  private_subnets = var.vpc_private_subnets

  enable_nat_gateway = false

  enable_vpn_gateway = true

  tags = {
    Owner       = "user"
    Environment = "staging"
    Name        = "complete"
  }
}

```

EXECUTING THE SCRIPTS

Terraform init will tell Terraform to scan the code, figure out which providers you're using, and download the code for them. By default, the provider code will be downloaded into a terraform folder, which is Terraform's scratch directory.

```

always2k@ubuntu:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.15.0...
- Installed hashicorp/aws v3.15.0 (signed by HashiCorp)

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, we recommend adding version constraints in a required_providers block
in your configuration, with the constraint strings suggested below.

* hashicorp/aws: version = "~> 3.15.0"

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.

```

To actually create the Instance, run the terraform apply command. Adding `--autoapprove` will eliminate the requirement of entering “yes” after command execution.

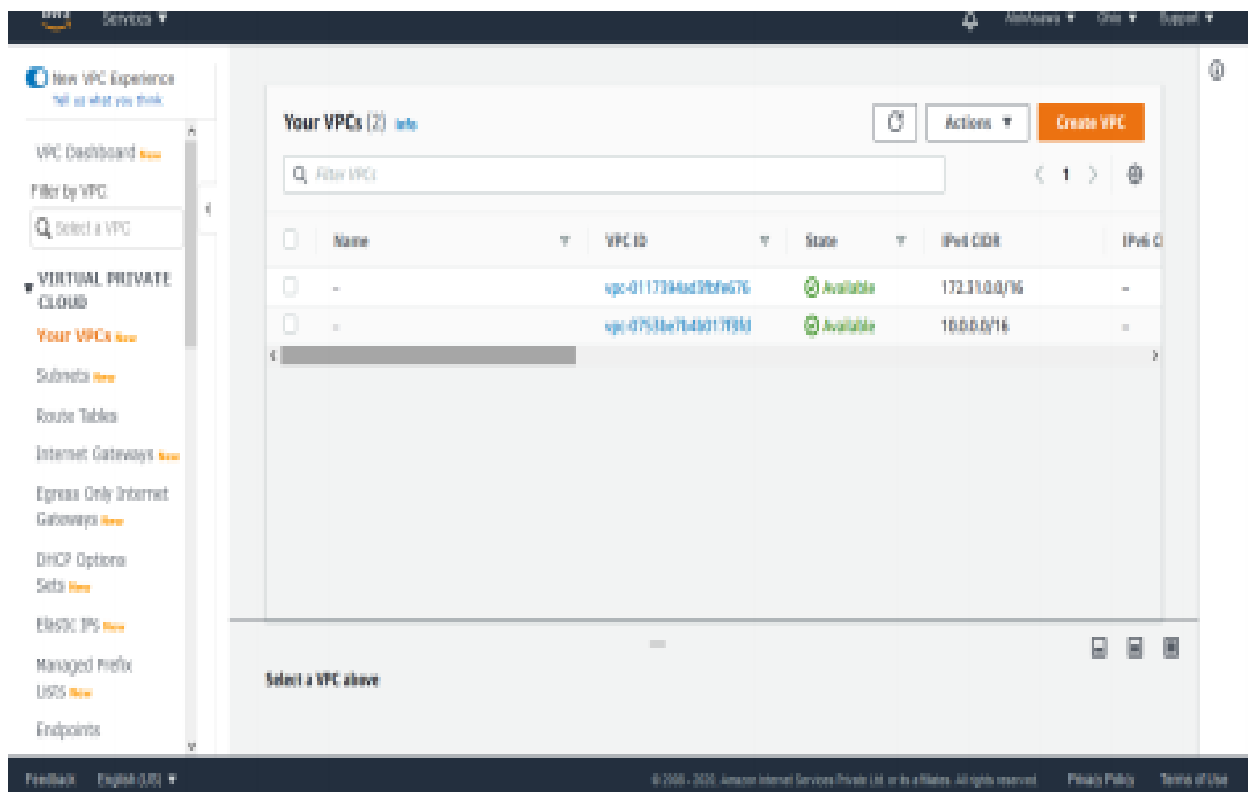
```
alwais2k@ubuntu:~$ terraform apply --auto-approve
aws_customer_gateway.customer_gateway: Creating...
aws_vpc.vpc: Creating...
aws_instance.FirstEc2Instance: Creating...
aws_s3_bucket.bucket: Creating...
aws_instance.SecondEc2Instance: Creating...
aws_instance.FirstEc2Instance: Still creating... [10s elapsed]
aws_vpc.vpc: Still creating... [10s elapsed]
aws_customer_gateway.customer_gateway: Still creating... [10s elapsed]
aws_s3_bucket.bucket: Still creating... [10s elapsed]
aws_instance.SecondEc2Instance: Still creating... [10s elapsed]
aws_customer_gateway.customer_gateway: Creation complete after 17s [id=cgw-0c43b1ca4e3738f]
aws_vpc.vpc: Creation complete after 17s [id=vpc-0207a49b6da427839]
aws_vpn_gateway.vpn_gateway: Creating...
aws_instance.FirstEc2Instance: Still creating... [20s elapsed]
aws_s3_bucket.bucket: Still creating... [20s elapsed]
aws_instance.SecondEc2Instance: Still creating... [20s elapsed]
aws_s3_bucket.bucket: Creation complete after 22s [id=darshbucket7899]
aws_vpn_gateway.vpn_gateway: Still creating... [10s elapsed]
aws_instance.FirstEc2Instance: Still creating... [30s elapsed]
aws_instance.SecondEc2Instance: Still creating... [30s elapsed]
aws_vpn_gateway.vpn_gateway: Creation complete after 16s [id=vgw-06ad6ec93c15795dc]
aws_vpn_connection.vpn: Creating...
aws_instance.FirstEc2Instance: Creation complete after 40s [id=i-0877949be9a90cbd5]
aws_instance.SecondEc2Instance: Still creating... [40s elapsed]
aws_vpn_connection.vpn: Still creating... [10s elapsed]
aws_instance.SecondEc2Instance: Still creating... [50s elapsed]
aws_instance.SecondEc2Instance: Creation complete after 51s [id=i-0a2b156bb39f3adfc]
aws_vpn_connection.vpn: Still creating... [10s elapsed]
aws_instance.SecondEc2Instance: Still creating... [50s elapsed]
aws_instance.SecondEc2Instance: Creation complete after 51s [id=i-0a2b156bb39f3adfc]
aws_vpn_connection.vpn: Still creating... [20s elapsed]
aws_vpn_connection.vpn: Still creating... [30s elapsed]
aws_vpn_connection.vpn: Still creating... [40s elapsed]
aws_vpn_connection.vpn: Still creating... [50s elapsed]
aws_vpn_connection.vpn: Still creating... [1n0s elapsed]
aws_vpn_connection.vpn: Still creating... [1n10s elapsed]
aws_vpn_connection.vpn: Still creating... [1n20s elapsed]
aws_vpn_connection.vpn: Still creating... [1n30s elapsed]
aws_vpn_connection.vpn: Still creating... [1n40s elapsed]
aws_vpn_connection.vpn: Still creating... [1n50s elapsed]
```

Now the resources are deployed to the AWS account using Terraform.

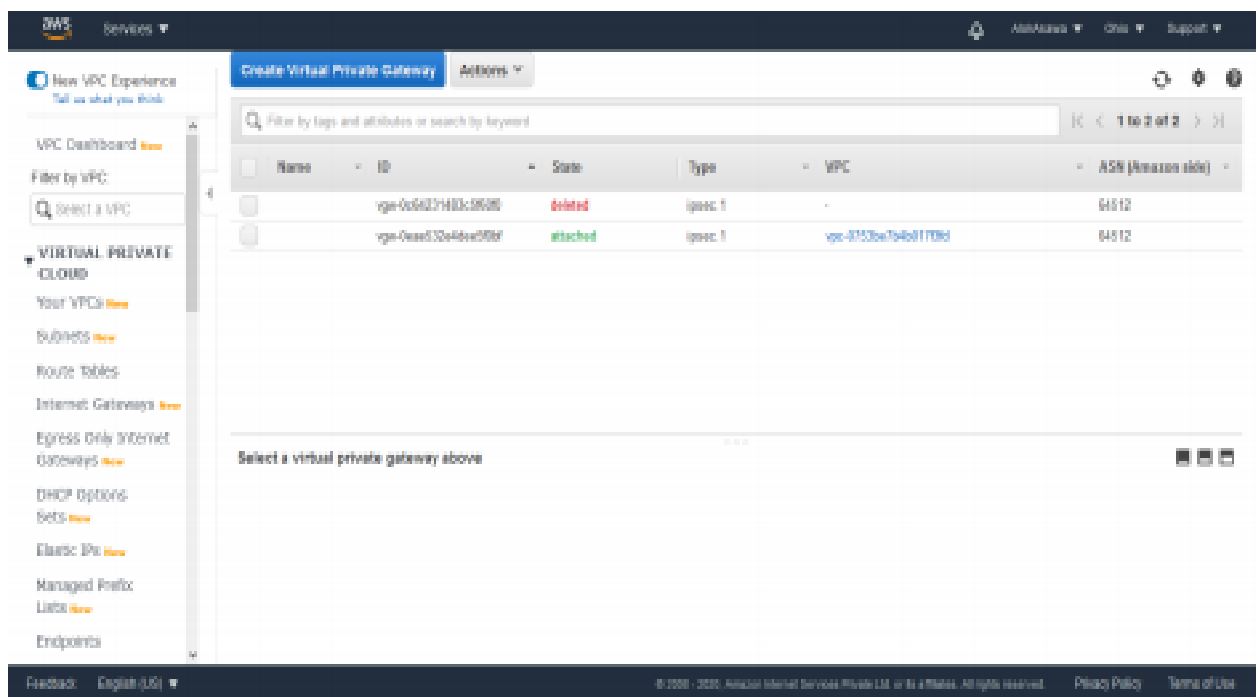
The screenshot shows the AWS Management Console interface. On the left is a navigation menu with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'AMIs'. The main area displays a 'Welcome to the new instances experience!' message. Below this, there's a section for 'Instances (2)' with a search bar and a filter dropdown set to 'Instance state: running'. A table lists two EC2 instances, both in a 'Running' state. The footer of the console shows the year 2020 and various legal links.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone
<input type="checkbox"/>	EC2_Instance_2	i-06d780e954c5e440d	Running	t2.micro	2/2 checks ...	No alarms	us-east-2c
<input type="checkbox"/>	EC2_Instance_1	i-0a0b2edcd8903ae96	Running	t2.micro	2/2 checks ...	No alarms	us-east-2c

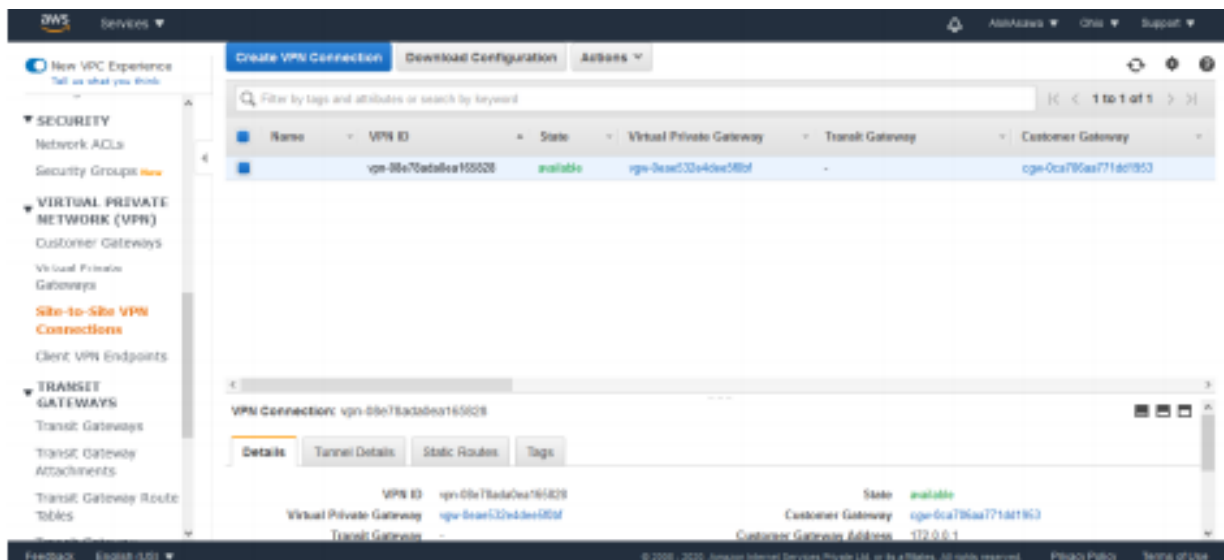
EC2 Instances



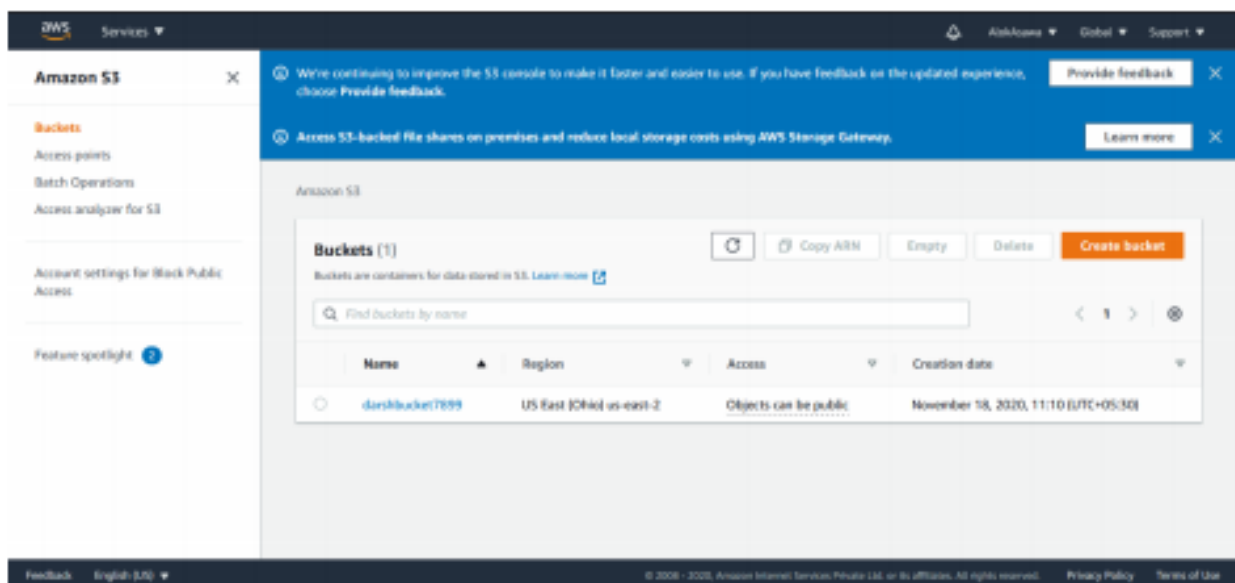
VPC DEPLOYMENT



VIRTUAL PRIVATE GATEWAY



VIRTUAL PRIVATE NETWORK CONNECTION



S3 Bucket