ASSIGNMENT-1

System Provisioning & Configuration Management

Terraform scripts to perform following tasks on AWS cloud Platform

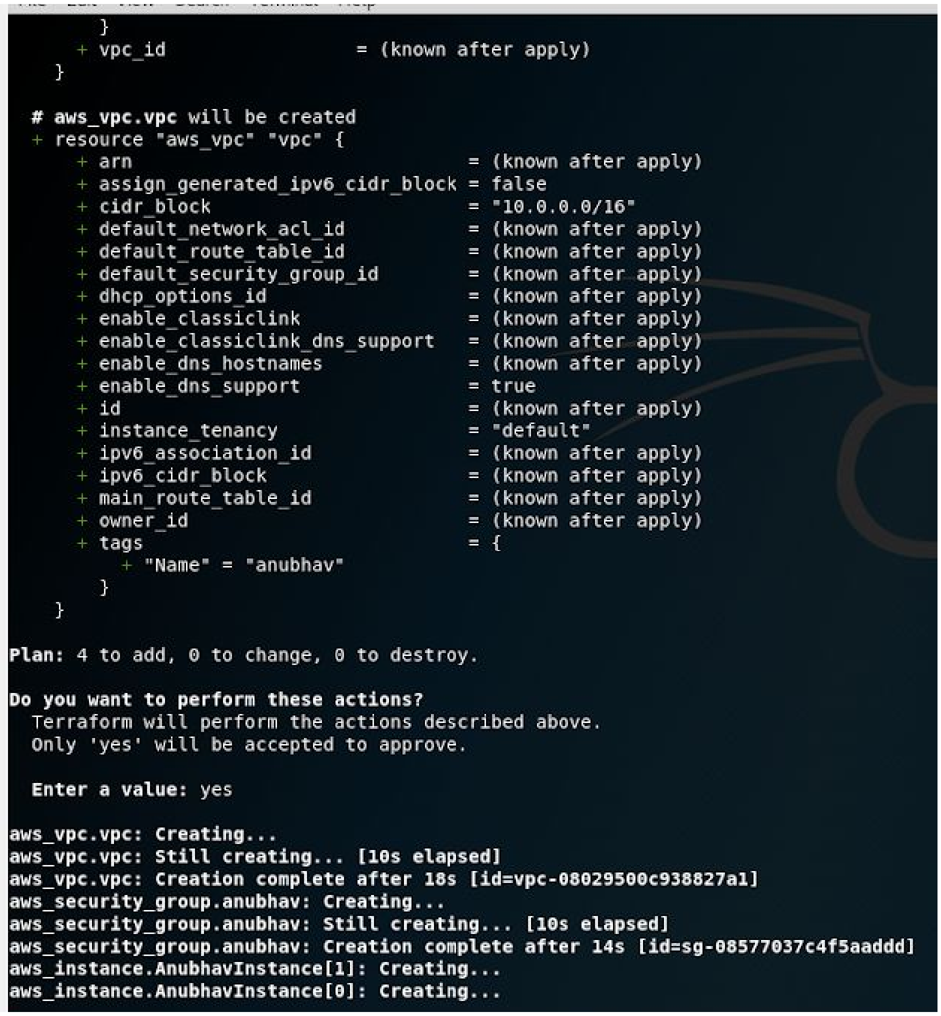
1. Creating two T2.micro ec2 instances

Code:

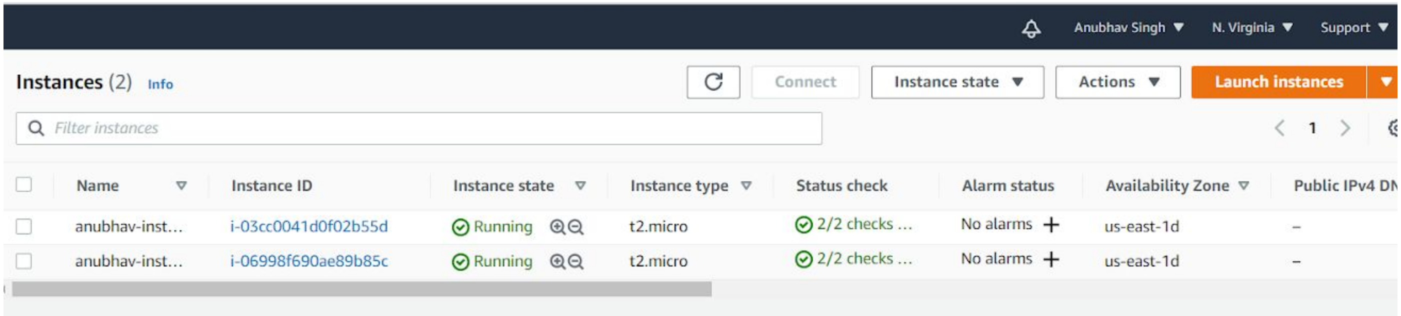
```
provider "aws"  {
 region= "us-east-1"
 access_key= "AKIAI3LLBDKEXYIXHXFQ"
 secret_key= "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"
}
resource "aws_instance" "myFirstInstance" {
  ami           = "ami-07dd19a7900a1f049"
  count=2
  key_name = "keypair"
  instance_type = "t2.micro"
  security_groups= [ "anubhav"]
  tags= {
    Name = "anubhav_instance"
  }
}
```

```
resource "aws_vpc" "vpc" {
  cidr_block = "10.0.0.0/24"
}

resource "aws_security_group" "anubhav" {
  name        = "anubhav"
  description = "security group "

  ingress {
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

 ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags= {
    Name = "anubhav"
  }
}
```

Output:



```
      }
    + vpc_id                 = (known after apply)
  }

# aws_vpc.vpc will be created
+ resource "aws_vpc" "vpc" {
    + arn                              = (known after apply)
    + assign_generated_ipv6_cidr_block = false
    + cidr_block                       = "10.0.0.0/16"
    + default_network_acl_id           = (known after apply)
    + default_route_table_id           = (known after apply)
    + default_security_group_id        = (known after apply)
    + dhcp_options_id                  = (known after apply)
    + enable_classiclink               = (known after apply)
    + enable_classiclink_dns_support   = (known after apply)
    + enable_dns_hostnames             = (known after apply)
    + enable_dns_support               = true
    + id                               = (known after apply)
    + instance_tenancy                 = "default"
    + ipv6_association_id              = (known after apply)
    + ipv6_cidr_block                  = (known after apply)
    + main_route_table_id              = (known after apply)
    + owner_id                         = (known after apply)
    + tags                             = {
        + "Name" = "anubhav"
      }
  }

Plan: 4 to add, 0 to change, 0 to destroy.

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_vpc.vpc: Creating...
aws_vpc.vpc: Still creating... [10s elapsed]
aws_vpc.vpc: Creation complete after 18s [id=vpc-08029500c938827a1]
aws_security_group.anubhav: Creating...
aws_security_group.anubhav: Still creating... [10s elapsed]
aws_security_group.anubhav: Creation complete after 14s [id=sg-08577037c4f5aaddd]
aws_instance.AnubhavInstance[1]: Creating...
aws_instance.AnubhavInstance[0]: Creating...
```



Instances (2) Info

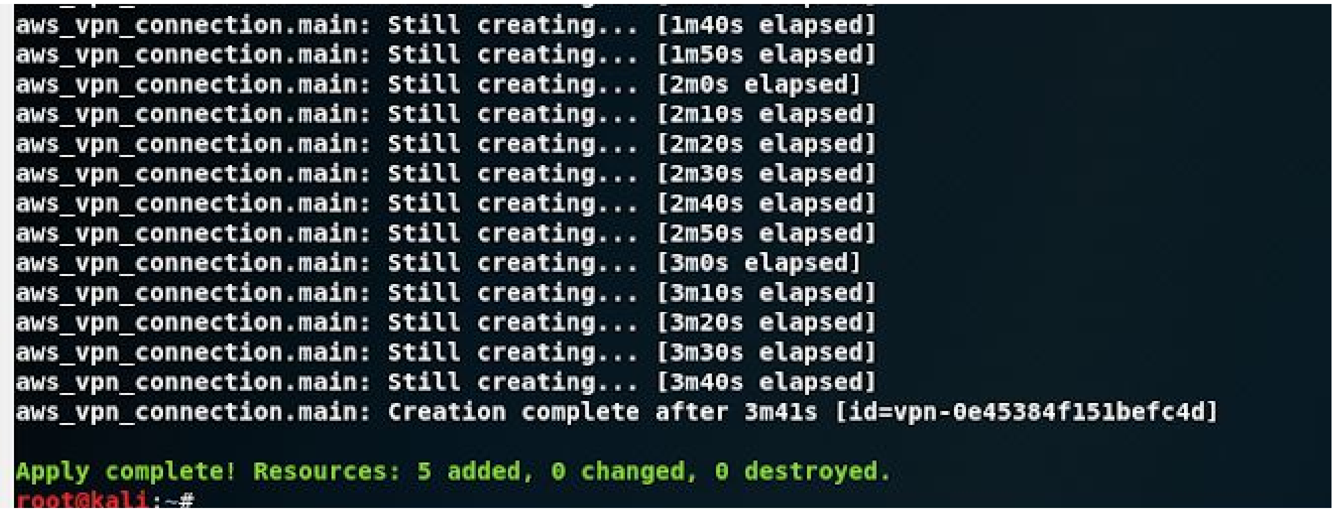
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
□	anubhav-inst...	i-03cc0041d0f02b55d	✓ Running	t2.micro	✓ 2/2 checks ...	No alarms +	us-east-1d	–
□	anubhav-inst...	i-06998f690ae89b85c	✓ Running	t2.micro	✓ 2/2 checks ...	No alarms +	us-east-1d	–

2. Creating a VPN on AWS

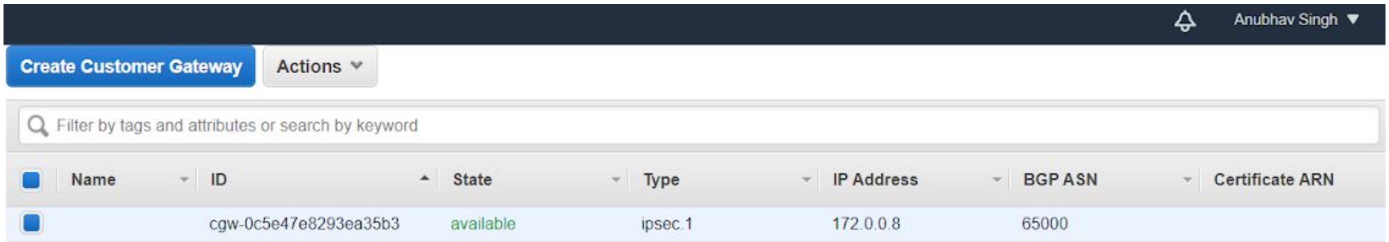
Code:

```
provider "aws"  {
 region= "us-east-1"
 access_key= "AKIAI3LLBDKEXYIXHXFQ"
 secret_key= "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"
}
resource "aws_vpc" "vpc" {
  cidr_block = "10.0.0.0/24"
}
resource "aws_vpn_gateway" "vpn_gateway" {
  vpc_id = aws_vpc.vpc.id
}
resource "aws_customer_gateway" "customer_gateway" {
  bgp_asn    = 65000
  ip_address = "172.0.0.8"
  type       = "ipsec.1"
}

resource "aws_vpn_connection" "main" {
  vpn_gateway_id      = aws_vpn_gateway.vpn_gateway.id
  customer_gateway_id = aws_customer_gateway.customer_gateway.id
  type                = "ipsec.1"
  static_routes_only  = true
}

resource "aws_security_group" "anubhav" {
  name        = "anubhav"
  description = "security group "

  ingress {
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

 ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
 egress {
  from_port   = 0
  to_port     = 65535
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }
 tags= {
  Name = "anubhav"
 }
}
```

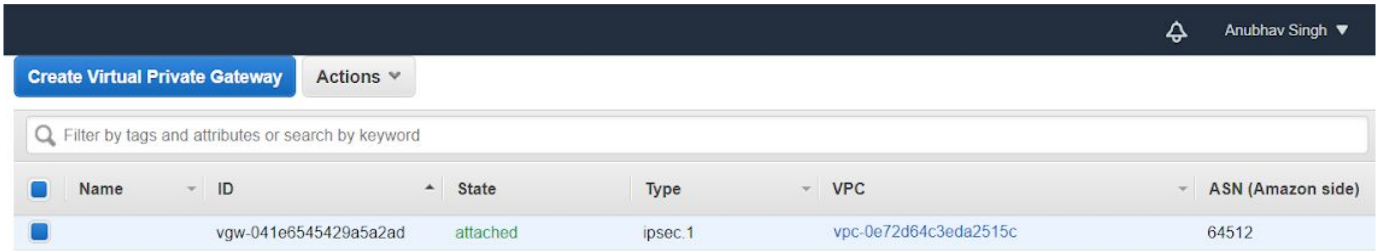
Output:



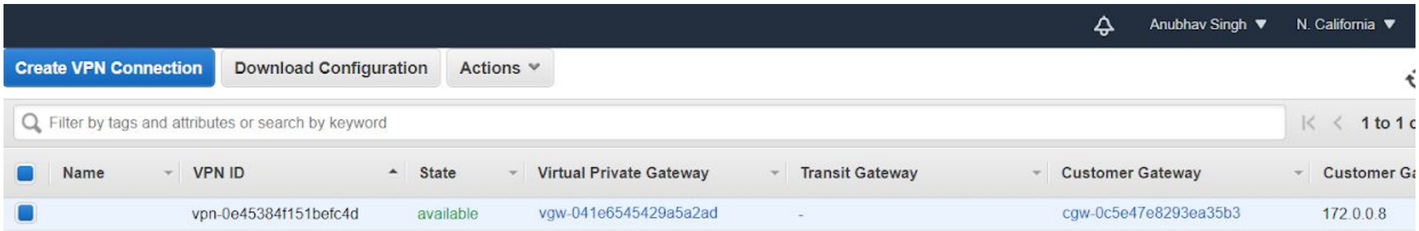
```
aws_vpn_connection.main: Still creating... [1m40s elapsed]
aws_vpn_connection.main: Still creating... [1m50s elapsed]
aws_vpn_connection.main: Still creating... [2m0s elapsed]
aws_vpn_connection.main: Still creating... [2m10s elapsed]
aws_vpn_connection.main: Still creating... [2m20s elapsed]
aws_vpn_connection.main: Still creating... [2m30s elapsed]
aws_vpn_connection.main: Still creating... [2m40s elapsed]
aws_vpn_connection.main: Still creating... [2m50s elapsed]
aws_vpn_connection.main: Still creating... [3m0s elapsed]
aws_vpn_connection.main: Still creating... [3m10s elapsed]
aws_vpn_connection.main: Still creating... [3m20s elapsed]
aws_vpn_connection.main: Still creating... [3m30s elapsed]
aws_vpn_connection.main: Still creating... [3m40s elapsed]
aws_vpn_connection.main: Creation complete after 3m41s [id=vpn-0e45384f151befc4d]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
root@kali:~#
```



☐	Name	ID	State	Type	IP Address	BGP ASN	Certificate ARN
☐		cgw-0c5e47e8293ea35b3	available	ipsec.1	172.0.0.8	65000	


Anubhav Singh
Create Virtual Private Gateway
Actions
Filter by tags and attributes or search by keyword

☐	Name	ID	State	Type	VPC	ASN (Amazon side)
☐		vgw-041e6545429a5a2ad	attached	ipsec.1	vpc-0e72d64c3eda2515c	64512



☐	Name	VPN ID	State	Virtual Private Gateway	Transit Gateway	Customer Gateway	Customer Ga
☐		vpn-0e45384f151befc4d	available	vgw-041e6545429a5a2ad	-	cgw-0c5e47e8293ea35b3	172.0.0.8

3. Creating a S3 bucket

Script :

```
provider "aws"  {

 region= "us-east-1"

 access_key= "AKIAI3LLBDKEXYIXHXFQ"

 secret_key= "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"
```

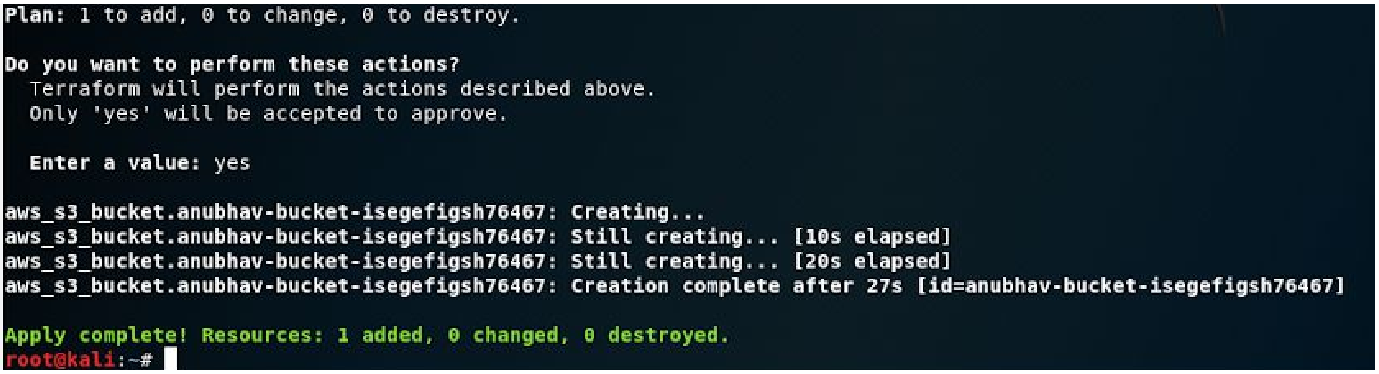
}

resource "aws_s3_bucket" "anubhav-bucket-isegefigsh76467" {

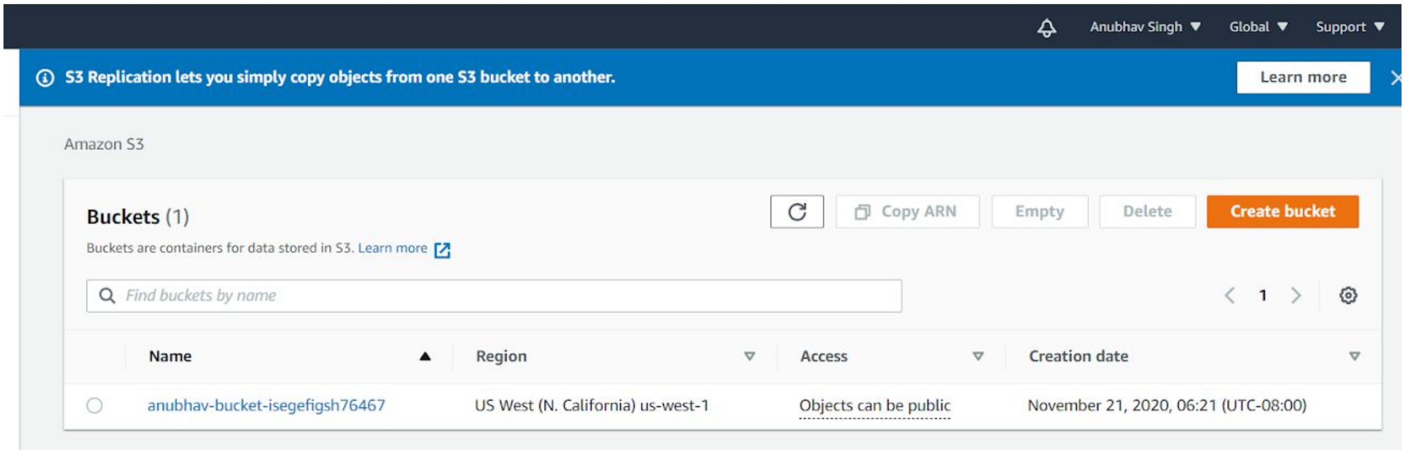
  bucket = "anubhav-bucket-isegefigsh76467"

}

Output :



```
Plan: 1 to add, 0 to change, 0 to destroy.

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.anubhav-bucket-isegefigsh76467: Creating...
aws_s3_bucket.anubhav-bucket-isegefigsh76467: Still creating... [10s elapsed]
aws_s3_bucket.anubhav-bucket-isegefigsh76467: Still creating... [20s elapsed]
aws_s3_bucket.anubhav-bucket-isegefigsh76467: Creation complete after 27s [id=anubhav-bucket-isegefigsh76467]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@kali:~#
```



S3 Replication lets you simply copy objects from one S3 bucket to another.

Buckets (1)

Buckets are containers for data stored in S3. Learn more

	Name	Region	Access	Creation date
○	anubhav-bucket-isegefigsh76467	US West (N. California) us-west-1	Objects can be public	November 21, 2020, 06:21 (UTC-08:00)