

SubmittedBy:Vedansh Singhal
Submitted To:Mr.Hitesh Kumar Sharmasir
SapID:500061440
RollNo:R171217060
Devops2017-2021
Batch B2

System Provisioning And Configuration

Managment

Assignment -1

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

First create a directory project-terraform and initialise terraform which is installed on your system by following command:

Terraform init

```
Initializing the backend...

Initializing provider plugins...

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, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 3.16"

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.
```

Step 1: Create two T2 Micro EC2 Instances.

Step2: Create a VPN on AWS

Step 3: Create a S3 Bucket

Then using notepad create a file in terraform with .tf extension and add below commands and set the ami from the screen as shown below amd set the instance type as t2 micro and key_name as "os1":

```

provider "aws"{
region="ap-south-1"
profile="VedanshSinghal"
}
resource "aws_security_group" "secgrp" {
name      = "secgrp"
description = "Security for instance"

ingress {
from_port = 443
to_port   = 443
protocol  = "tcp"
cidr_blocks = ["0.0.0.0/0"]
}
ingress {
from_port = 80
to_port   = 80
protocol  = "tcp"
cidr_blocks = ["0.0.0.0/0"]
}

```

```

cidr_blocks = ["0.0.0.0/0"]
}
ingress {
from_port = 80
to_port   = 80
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   = 0
protocol  = "-1"
cidr_blocks = ["0.0.0.0/0"]
}

```

```
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" "singhal" {
  ami = "ami-0a9d27a9f4f5c0efc"
count=2
  instance_type = "t2.micro"
```

```
key_name = "os1"
```

```
security_groups=["${aws_security_group.secgrp.name}"]
tags = {
  Name = "vedansh1"
}
}
```

```
resource "aws_vpc" "vpc" {
  cidr_block = "10.0.0.0/16"
}
```

```
resource "aws_vpn_gateway" "vpn_gateway" {
  vpc_id = aws_vpc.vpc.id
}
```

```
resource "aws_customer_gateway" "customer_gateway" {
  bgp_asn = 65000
  ip_address = "172.0.0.1"
  time_zone = "America/Los_Angeles"
```

```
vpc_id = aws_vpc.vpc.id  
}
```

```
resource "aws_customer_gateway" "customer_gateway" {  
  bgp_asn = 65000  
  ip_address = "172.0.0.1"  
  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_s3_bucket" "singhal" {  
  bucket = "vedansh"  
  acl = "public-read"  
}
```

```
aws_customer_gateway.customer_gateway: Creating...  
aws_vpc.vpc: Creating...  
aws_s3_bucket.singhal: Creating...  
aws_security_group.secgrp: Creating...  
aws_security_group.secgrp: Creation complete after 3s [id=sg-0afce6605b888a81c]  
aws_instance.singhal[1]: Creating...  
aws_instance.singhal[0]: Creating...  
aws_vpc.vpc: Creation complete after 4s [id=vpc-09e41b55789f5789e]  
aws_vpn_gateway.vpn_gateway: Creating...  
aws_s3_bucket.singhal: Creation complete after 5s [id=vedansh]  
aws_customer_gateway.customer_gateway: Still creating... [10s elapsed]  
aws_customer_gateway.customer_gateway: Creation complete after 11s [id=cgw-0e3d0595bb77aa49c]  
aws_instance.singhal[1]: Still creating... [10s elapsed]  
aws_instance.singhal[0]: Still creating... [10s elapsed]  
aws_vpn_gateway.vpn_gateway: Creation complete after 9s [id=vgw-0196f13fb35a4e039]  
aws_vpn_connection.main: Creating...  
aws_instance.singhal[1]: Still creating... [20s elapsed]  
aws_instance.singhal[0]: Still creating... [20s elapsed]  
aws_vpn_connection.main: Still creating... [10s elapsed]  
aws_instance.singhal[1]: Creation complete after 26s [id=i-090f0bcd2c44ff570]  
aws_instance.singhal[0]: Still creating... [30s elapsed]  
aws_vpn_connection.main: Still creating... [20s elapsed]  
aws_instance.singhal[0]: Creation complete after 35s [id=i-06854db028aa3d86a]  
aws_vpn_connection.main: Still creating... [30s elapsed]  
aws_vpn_connection.main: Still creating... [40s elapsed]  
aws_vpn_connection.main: Still creating... [50s elapsed]  
aws_vpn_connection.main: Still creating... [1m0s elapsed]  
aws_vpn_connection.main: Still creating... [1m10s elapsed]  
aws_vpn_connection.main: Still creating... [1m20s elapsed]  
aws_vpn_connection.main: Still creating... [1m30s elapsed]  
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: Creation complete after 3m29s [id=vpn-0e5d1b508d93f1901]  
  
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
```

Step1: Created two T2 Micro EC2 Instances.

The screenshot shows the AWS Management Console 'Instances' page. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The main content area displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Three instances are listed: 'wordpress', 'vedansh1', and 'vedansh1'. All instances are in the 'Running' state and are t2.micro instances. The 'wordpress' instance is in ap-south-1a, while the two 'vedansh1' instances are in ap-south-1b. The 'vedansh1' instance has two different IDs: i-090f0bcd2c44ff570 and i-06854db028aa3d86a. The 'Status check' column shows '2/2 checks ...' for all instances. The 'Alarm status' column shows 'No alarms' for all instances. The 'Availability Zone' column shows 'ap-south-1a' for 'wordpress' and 'ap-south-1b' for the 'vedansh1' instances. The 'Instance state' dropdown is set to 'running'. The 'Launch instances' button is visible in the top right corner.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
wordpress	i-0c7c356234e79db65	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1a
vedansh1	i-090f0bcd2c44ff570	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1b
vedansh1	i-06854db028aa3d86a	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1b

Step2: Create a VPN on AWS


The screenshot shows the AWS Management Console 'Create Customer Gateway' page. The left sidebar contains navigation links for Network ACLs, Security Groups, VIRTUAL PRIVATE NETWORK (VPN), TRANSIT GATEWAYS, and Network Manager. The main content area displays a table of customer gateways with columns: Name, ID, State, Type, IP Address, BGP ASN, and Certificate ARN. One customer gateway is listed: 'cgw-0e3d0595bb77aa49c'. The 'State' is 'available', the 'Type' is 'ipsec.1', the 'IP Address' is '172.0.0.1', and the 'BGP ASN' is '65000'. The 'Certificate ARN' is empty. The 'Create Customer Gateway' button is visible in the top left corner. Below the table, the details of the selected customer gateway are shown, including the ID, Type, BGP ASN, State, IP Address, and Certificate ARN.

Name	ID	State	Type	IP Address	BGP ASN	Certificate ARN
cgw-0e3d0595bb77aa49c	cgw-0e3d0595bb77aa49c	available	ipsec.1	172.0.0.1	65000	

The screenshot shows the AWS Management Console 'Create VPN Connection' page. The left sidebar contains navigation links for Network ACLs, Security Groups, VIRTUAL PRIVATE NETWORK (VPN), TRANSIT GATEWAYS, and Network Manager. The main content area displays a table of VPN connections with columns: Name, VPN ID, State, Virtual Private Gateway, Transit Gateway, and Customer Gateway. One VPN connection is listed: 'vpn-0e5d1b508d93f1901'. The 'State' is 'available', the 'Virtual Private Gateway' is 'vgw-0196f13fb35a4e039', the 'Transit Gateway' is '-', and the 'Customer Gateway' is 'cgw-0e3d0595bb77aa49c'. The 'Create VPN Connection' button is visible in the top left corner. Below the table, the details of the selected VPN connection are shown, including the VPN ID, Virtual Private Gateway, Transit Gateway, State, Customer Gateway, and Customer Gateway Address.

Name	VPN ID	State	Virtual Private Gateway	Transit Gateway	Customer Gateway
vpn-0e5d1b508d93f1901	vpn-0e5d1b508d93f1901	available	vgw-0196f13fb35a4e039	-	cgw-0e3d0595bb77aa49c

And after that I have destroyed my whole infrastructure just by running one command
terraform destroy -auto-approve

 Command Prompt

```
aws_vpc.vpc: Refreshing state... [id=vpc-09e41b55789f5789e]
aws_security_group.secgrp: Refreshing state... [id=sg-0afce6605b888a81c]
aws_s3_bucket.singhal: Refreshing state... [id=vedansh]
aws_instance.singhal[1]: Refreshing state... [id=i-090f0bcd2c44ff570]
aws_instance.singhal[0]: Refreshing state... [id=i-06854db028aa3d86a]
aws_vpn_gateway.vpn_gateway: Refreshing state... [id=vgw-0196f13fb35a4e039]
aws_vpn_connection.main: Refreshing state... [id=vpn-0e5d1b508d93f1901]
aws_instance.singhal[0]: Destroying... [id=i-06854db028aa3d86a]
aws_s3_bucket.singhal: Destroying... [id=vedansh]
aws_instance.singhal[1]: Destroying... [id=i-090f0bcd2c44ff570]
aws_vpn_connection.main: Destroying... [id=vpn-0e5d1b508d93f1901]
aws_s3_bucket.singhal: Destruction complete after 0s
aws_instance.singhal[0]: Still destroying... [id=i-06854db028aa3d86a, 10s elapsed]
aws_instance.singhal[1]: Still destroying... [id=i-090f0bcd2c44ff570, 10s elapsed]
aws_vpn_connection.main: Still destroying... [id=vpn-0e5d1b508d93f1901, 10s elapsed]
aws_vpn_connection.main: Destruction complete after 11s
aws_vpn_gateway.vpn_gateway: Destroying... [id=vgw-0196f13fb35a4e039]
aws_customer_gateway.customer_gateway: Destroying... [id=cgw-0e3d0595bb77aa49c]
aws_vpn_gateway.vpn_gateway: Destruction complete after 1s
aws_vpc.vpc: Destroying... [id=vpc-09e41b55789f5789e]
aws_customer_gateway.customer_gateway: Destruction complete after 1s
aws_vpc.vpc: Destruction complete after 5s
aws_instance.singhal[0]: Still destroying... [id=i-06854db028aa3d86a, 20s elapsed]
aws_instance.singhal[1]: Still destroying... [id=i-090f0bcd2c44ff570, 20s elapsed]
aws_instance.singhal[0]: Still destroying... [id=i-06854db028aa3d86a, 30s elapsed]
aws_instance.singhal[1]: Still destroying... [id=i-090f0bcd2c44ff570, 30s elapsed]
aws_instance.singhal[1]: Destruction complete after 30s
aws_instance.singhal[0]: Destruction complete after 31s
aws_security_group.secgrp: Destroying... [id=sg-0afce6605b888a81c]
aws_security_group.secgrp: Destruction complete after 1s

Destroy complete! Resources: 8 destroyed.
```