

## SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT (ASSIGNMENT 1) Prajjawal Banati

Terraform scripts to perform following tasks on AWS cloud Platform

1. Creating two T2 micro ec2 instances
2. Creating a VPN on AWS
3. Creating a S3 bucket

### What is Terraform?

Terraform is an open source tool for infrastructure provisioning created by HashiCorp. It provides Infrastructure as code allowing you to automate and manage your infrastructure, platform and your services that run on the platform. Terraform can manage existing and popular service providers(aws, azure, GCP etc).

You do not have to prepare infrastructure like private network space, ec2 server instances, installing docker and other tools and security. Terraform does all that for you by preparing the whole infrastructure using terraform scripts. Thus, it is a software tool that provides Infrastructure as code. Terraform is declarative which means you define what you want.

Run the following commands

### # terraform init

Initializes working directory containing terraform configuration files. It is safe to run this command multiple times.

```
~ /COLLEGE/SEM VII/System-Provisioning/Assignment-1  ✓< base  at 07:16:43 PM
terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 3.16.0...

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.
```

## # terraform plan

To create execution plan that helps you check whether execution plan matches your expectations.

```

❏ > ~\COLLEGE\SEM VII\System-Provisioning\Assignment-1
❏ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

-----

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_customer_gateway.customer_gateway will be created
+ resource "aws_customer_gateway" "customer_gateway" {
+   arn          = (known after apply)
+   bgp_asn      = "65000"
+   id           = (known after apply)
+   ip_address   = "172.0.0.1"
+   type         = "ipsec.1"
}

# aws_instance.name[0] will be created
+ resource "aws_instance" "name" {
+   ami              = "value"
+   arn              = (known after apply)
+   associate_public_ip_address = true
+   availability_zone = (known after apply)
+   cpu_core_count   = (known after apply)
+   cpu_threads_per_core = (known after apply)
+   get_password_data = false
+   host_id          = (known after apply)
+   id               = (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          = "My Key"
+   outpost_arn       = (known after apply)
+   password_data     = (known after apply)
+   placement_group   = (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)
+   secondary_private_ips = (known after apply)
+   security_groups    = (known after apply)
+   source_dest_check  = true
+   subnet_id         = (known after apply)
+   tags              = {
+     "Name" = "Instance"
+   }
+   tenancy      = (known after apply)
}

# aws_vpn_connection.main will be created
+ resource "aws_vpn_connection" "main" {
+   arn                  = (known after apply)
+   customer_gateway_configuration = (known after apply)
+   customer_gateway_id   = (known after apply)
+   id                   = (known after apply)
+   routes                = (known after apply)
+   static_routes_only    = true
+   transit_gateway_attachment_id = (known after apply)
+   tunnel1_address       = (known after apply)
+   tunnel1_bgp_asn       = (known after apply)
+   tunnel1_bgp_holdtime  = (known after apply)
+   tunnel1_cgwr_inside_address = (known after apply)
+   tunnel1_inside_cidr   = (known after apply)
+   tunnel1_preshared_key = (sensitive value)
+   tunnel1_vgw_inside_address = (known after apply)
+   tunnel2_address       = (known after apply)
+   tunnel2_bgp_asn       = (known after apply)
+   tunnel2_bgp_holdtime  = (known after apply)
+   tunnel2_cgwr_inside_address = (known after apply)
+   tunnel2_inside_cidr   = (known after apply)
+   tunnel2_preshared_key = (sensitive value)
+   tunnel2_vgw_inside_address = (known after apply)
+   type                  = "ipsec.1"
+   vgw_telemetry          = (known after apply)
+   vpn_gateway_id        = (known after apply)
}

# aws_vpn_gateway.vpn_gateway will be created
+ resource "aws_vpn_gateway" "vpn_gateway" {
+   amazon_side_asn = (known after apply)
+   arn             = (known after apply)
+   id              = (known after apply)
+   vpc_id          = (known after apply)
}

Plan: 13 to add, 0 to change, 0 to destroy.

-----

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.

```

### # terraform apply -auto-approve

To apply the changes to reach the desired state of the configuration

```
❏ > ~/COLLEGE/SEM VII/System-Provisioning/Assignment-1
terraform apply -auto-approve
aws_customer_gateway.customer_gateway: Creating...
aws_key_pair.my-ssh-key: Creating...
aws_vpc.my-vpc: Creating...
aws_s3_bucket.my-bucket: Creating...
aws_key_pair.my-ssh-key: Creation complete after 2s [id=My Key]
aws_customer_gateway.customer_gateway: Still creating... [10s elapsed]
aws_vpc.my-vpc: Still creating... [10s elapsed]
aws_s3_bucket.my-bucket: Still creating... [10s elapsed]
aws_customer_gateway.customer_gateway: Creation complete after 16s [id=cgw-0458b2bfedf94642b]
aws_vpc.my-vpc: Creation complete after 19s [id=vpc-06c77adb72dba69aa]
aws_vpn_gateway.vpn_gateway: Creating...
aws_internet_gateway.my-gateway: Creating...
aws_subnet.my-subnet: Creating...
aws_security_group.my-group: Creating...
aws_s3_bucket.my-bucket: Still creating... [20s elapsed]
aws_s3_bucket.my-bucket: Creation complete after 22s [id=prajjawalmybucket]
aws_subnet.my-subnet: Creation complete after 5s [id=subnet-006d560e6e3eab1a6]
aws_internet_gateway.my-gateway: Creation complete after 7s [id=igw-0b9d0acb2e097cd6b]
aws_route_table.my-route-table: Creating...
aws_vpn_gateway.vpn_gateway: Still creating... [10s elapsed]
aws_security_group.my-group: Still creating... [10s elapsed]
aws_security_group.my-group: Creation complete after 11s [id=sg-004ceabf1fab9a275]
aws_instance.name[0]: Creating...
aws_instance.name[1]: Creating...
aws_route_table.my-route-table: Creation complete after 6s [id=rtb-05312599030ec00fa]
aws_route_table_association.my-route-table-association: Creating...
aws_route_table_association.my-route-table-association: Creation complete after 2s [id=rtbassoc-09b8b2b4b4d1c2a56]
aws_vpn_gateway.vpn_gateway: Still creating... [20s elapsed]
aws_instance.name[0]: Still creating... [10s elapsed]
aws_instance.name[1]: Still creating... [10s elapsed]
aws_vpn_gateway.vpn_gateway: Creation complete after 26s [id=vgw-093f82e688c5bb191]
aws_vpn_connection.main: Creating...
aws_instance.name[0]: Still creating... [20s elapsed]
aws_instance.name[1]: Still creating... [20s elapsed]
aws_vpn_connection.main: Still creating... [10s elapsed]
aws_instance.name[0]: Still creating... [30s elapsed]
aws_instance.name[1]: Still creating... [30s elapsed]
aws_vpn_connection.main: Still creating... [20s elapsed]
aws_instance.name[1]: Creation complete after 39s [id=i-083513d152b1a5b03]
aws_instance.name[0]: Still creating... [40s elapsed]
aws_vpn_connection.main: Still creating... [30s elapsed]
aws_instance.name[0]: Still creating... [50s elapsed]
aws_instance.name[0]: Creation complete after 51s [id=i-03b325dc82b5a40db]
aws_vpn_connection.main: Still creating... [40s elapsed]
aws_vpn_connection.main: Still creating... [7m50s elapsed]
aws_vpn_connection.main: Still creating... [8m0s elapsed]
aws_vpn_connection.main: Creation complete after 8m5s [id=vpn-0b0e323e7d7ddb3c4]

Apply complete! Resources: 13 added, 0 changed, 0 destroyed.
```

Now you can check the instances, VPN and S3 bucket have been created on your AWS cloud. Both t2-micro ec2-instances are created in N. Virginia region.

**Welcome to the new instances experience!**  
We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

**Instances (1/2)** Info

Filter instances

Instance state: running X Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/>	Instance	i-083513d152b1a5b03	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c	-	3.94.102.220	-
<input type="checkbox"/>	Instance	i-03b325dc82b5a40db	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c	-	18.215.181.144	-

**Instance: i-083513d152b1a5b03 (Instance)**

Details | **Security** | Networking | Storage | Status Checks | Monitoring | Tags

**Security details**

IAM Role: - Owner ID: 593856011819 Launch time: Fri Nov 20 2020 21:58:04 GMT+0530 (India Standard Time)

Security groups: sg-004ceabf1fab9a275 (terraform-20201120160750940400000001)

**Inbound rules**

Filter rules

Port range	Protocol	Source	Security groups
22	TCP	10.0.0.0/16	terraform-20201120160750940400000001

**Outbound rules**

## VPN

**Create VPN Connection** Download Configuration Actions

Filter by tags and attributes or search by keyword

	Name	VPN ID	State	Virtual Private Gateway	Transit Gateway	Customer Gateway	Customer Gateway Address	Inside Ip Version	Type	Categ
<input type="checkbox"/>		vpn-078832e1671db63a	deleted	vpgw-0c7880ab9b8e86a6	-	cgw-0473e794e4e550d17	172.0.0.1	IPv4	ipse.1	VPN
<input checked="" type="checkbox"/>		vpn-0b0e323e7d7db3c4	available	vpgw-09382e688c5b191	-	cgw-0458b2bfed94642b	172.0.0.1	IPv4	ipse.1	VPN

**VPN Connection: vpn-0b0e323e7d7db3c4**

Details | Tunnel Details | Static Routes | Tags

VPN ID: vpn-0b0e323e7d7db3c4 State: available

Virtual Private Gateway: vpgw-09382e688c5b191

Transit Gateway: -

Type: ipsec.1

VPC: vpc-06c77adb72bba9aa

Acceleration Enabled: false

Local IPv4 Network Cidr: 0.0.0.0/0

Local IPv6 Network Cidr: -

Customer Gateway: cgw-0458b2bfed94642b

Customer Gateway Address: 172.0.0.1

Category: VPN

Routing: Static

Authentication Type: Pre Shared Key

Remote IPv4 Network Cidr: 0.0.0.0/0

Remote IPv6 Network Cidr: -



Feedback English (US) ▼ © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use