

Assignment

SPCM

Rachit Chauhan

500060009

R171217049

Btech. CS-DevOps

1. Create a folder/directory in which you want to work.
2. Go to that folder and inside that create a file with name main.tf.
3. Now, setup a connection to aws using the access key and secret key which you can create and download from your aws management console by clicking on your name> security credentials>access key

Inside the main.tf file write the code

```
provider "aws" {  
  region= "us-west-2"  
  access_key= "AKIAIGOKUAPY3EMYUKIQ"  
  secret_key= "rI8a/FyoZ6pDEZBP9AY5yQ//Gg7tXATZERqyaFyc"  
}  
  
resource "aws_instance" "myFirstInstance" {  
  ami      = "ami-07dd19a7900a1f049"  
  count=2  
  key_name = "keypair"  
  instance_type = "t2.micro"  
  security_groups= [ "rachit"]  
  tags= {  
    Name = "rachit_instance"  
  }  
}  
  
resource "aws_vpc" "vpc" {  
  cidr_block = "10.0.0.0/24"  
}  
  
resource "aws_security_group" "rachit" {  
  name      = "rachit"  
  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 = "rachit"
  }
}
```

4. Then Initialize it with the help of terraform init command.

```

vagrant@ubuntu-xenial: ~/Spcm_project
vagrant@ubuntu-xenial:~/Spcm_project$ nano main.tf
vagrant@ubuntu-xenial:~/Spcm_project$ 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.
vagrant@ubuntu-xenial:~/Spcm_project$

```

5. Then terraform plan and then terraform apply.
6. This will create your instance

```

vagrant@ubuntu-xenial: ~/Spcm_project
+ cidr_block              = "10.0.0.0/24"
+ 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)
}

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

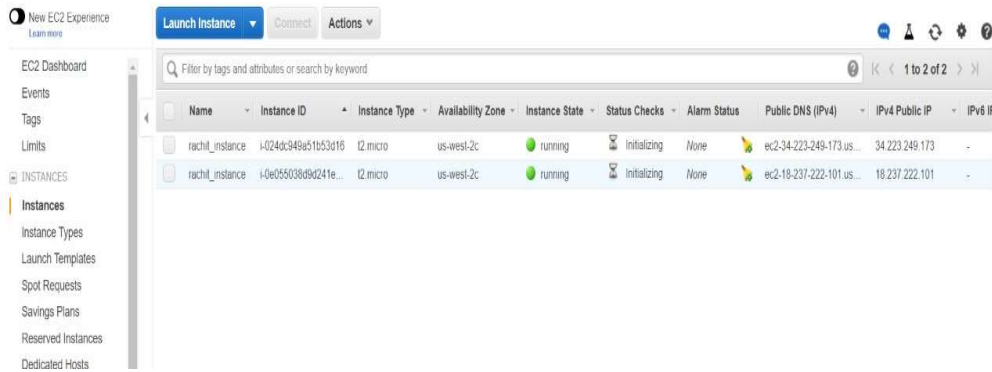
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.myFirstInstance[1]: Creating...
aws_vpc.vpc: Creating...
aws_instance.myFirstInstance[0]: Creating...
aws_security_group.rachit: Creating...
aws_vpc.vpc: Still creating... [10s elapsed]
aws_instance.myFirstInstance[1]: Still creating... [10s elapsed]
aws_security_group.rachit: Still creating... [10s elapsed]
aws_instance.myFirstInstance[0]: Still creating... [10s elapsed]
aws_security_group.rachit: Creation complete after 12s [id=sg-0338b1dbe409aad47]
aws_vpc.vpc: Creation complete after 15s [id=vpc-0db002bb8aa7c8c76]
aws_instance.myFirstInstance[0]: Still creating... [20s elapsed]
aws_instance.myFirstInstance[1]: Still creating... [20s elapsed]
aws_instance.myFirstInstance[0]: Still creating... [30s elapsed]
aws_instance.myFirstInstance[1]: Still creating... [30s elapsed]
aws_instance.myFirstInstance[1]: Creation complete after 37s [id=i-024dc949a51b53d16]
aws_instance.myFirstInstance[0]: Creation complete after 37s [id=i-0e055038d9d241e2b]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
vagrant@ubuntu-xenial:~/Spcm_project$

```



For Creating Buckets add this change in the main.tf code file as:

```
provider "aws" {
  region= "us-west-2"

  access_key= "AKIAIEFPXQPZPRG725BA"

  secret_key= "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"
}

resource "aws_s3_bucket" "rachit" {
  bucket = "rachit"
}
```

```
vagrant@ubuntu-xenial: ~/Spcm_project

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_s3_bucket.rachitchauhan will be created
+ resource "aws_s3_bucket" "rachitchauhan" {
  + acceleration_status = (known after apply)
  + acl                 = "private"
  + arn                 = (known after apply)
  + bucket              = "rachitchauhan"
  + bucket_domain_name = (known after apply)
  + bucket_regional_domain_name = (known after apply)
  + force_destroy       = false
  + hosted_zone_id      = (known after apply)
  + id                  = (known after apply)
  + region              = (known after apply)
  + request_payer       = (known after apply)
  + website_domain      = (known after apply)
  + website_endpoint    = (known after apply)
  + versioning {
    + enabled = (known after apply)
    + mfa_delete = (known after apply)
  }
}

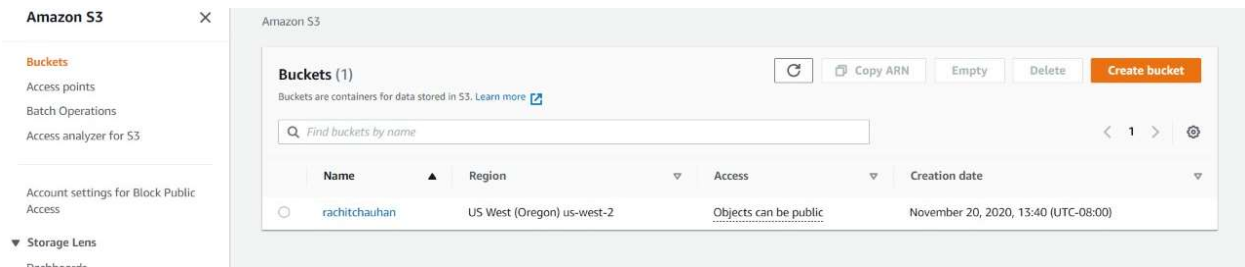
Plan: 1 to add, 0 to change, 0 to destroy.

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.rachitchauhan: Creating...
aws_s3_bucket.rachitchauhan: Still creating... [10s elapsed]
aws_s3_bucket.rachitchauhan: Still creating... [20s elapsed]
aws_s3_bucket.rachitchauhan: Creation complete after 20s [id=rachitchauhan]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
vagrant@ubuntu-xenial:~/Spcm_project$
```



For creating VPN

main.tf

```
provider "aws" {  
  region= "us-west-2"  
  
  access_key= "AKIAIGOKUAPY3EMYUKIQ"  
  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 = "172.0.0.12"  
  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" "rachit123" {
  name        = "rachit123"
  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 = "rachit123"
}
}

```

```

vagrant@ubuntu-xenial:~/Spcm_project$ terraform apply
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   = "172.0.0.12"
  + type        = "ipsec.1"
}

# aws_security_group.rachit123 will be created
+ resource "aws_security_group" "rachit123" {
  + arn          = (known after apply)
  + description   = "security group "
  + egress       = [
    + {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + description = ""
      + from_port   = 0
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
      + protocol     = "tcp"
      + security_groups = []
      + self         = false
    }
  ]
}

```


vagrant@ubuntu-xenial: ~/Spcm_project

```
} + owner_id = (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_cgw_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_cgw_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: 5 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.

vagrant@ubuntu-xenial: ~/Spcm_project

```
aws_customer_gateway.customer_gateway: Creation complete after 16s [id=cgw-0ccac3c55fd886551]
aws_vpn_gateway.vpn_gateway: Still creating... [10s elapsed]
aws_vpn_gateway.vpn_gateway: Creation complete after 14s [id=vgw-0b912f5ded5427635]
aws_vpn_connection.main: Creating...
aws_vpn_connection.main: Still creating... [10s elapsed]
aws_vpn_connection.main: Still creating... [21s elapsed]
aws_vpn_connection.main: Still creating... [31s elapsed]
aws_vpn_connection.main: Still creating... [41s elapsed]
aws_vpn_connection.main: Still creating... [51s elapsed]
aws_vpn_connection.main: Still creating... [1m1s elapsed]
aws_vpn_connection.main: Still creating... [1m11s elapsed]
aws_vpn_connection.main: Still creating... [1m21s elapsed]
aws_vpn_connection.main: Still creating... [1m31s elapsed]
aws_vpn_connection.main: Still creating... [1m41s elapsed]
aws_vpn_connection.main: Still creating... [1m51s elapsed]
aws_vpn_connection.main: Still creating... [2m1s elapsed]
aws_vpn_connection.main: Still creating... [2m11s elapsed]
aws_vpn_connection.main: Still creating... [2m21s elapsed]
aws_vpn_connection.main: Still creating... [2m31s elapsed]
aws_vpn_connection.main: Still creating... [2m41s elapsed]
aws_vpn_connection.main: Still creating... [2m51s elapsed]
aws_vpn_connection.main: Still creating... [3m1s elapsed]
aws_vpn_connection.main: Still creating... [3m11s elapsed]
aws_vpn_connection.main: Still creating... [3m21s elapsed]
aws_vpn_connection.main: Still creating... [3m31s elapsed]
aws_vpn_connection.main: Still creating... [3m41s elapsed]
aws_vpn_connection.main: Still creating... [3m51s elapsed]
aws_vpn_connection.main: Still creating... [4m1s elapsed]
aws_vpn_connection.main: Still creating... [4m11s elapsed]
aws_vpn_connection.main: Still creating... [4m21s elapsed]
aws_vpn_connection.main: Still creating... [4m31s elapsed]
aws_vpn_connection.main: Still creating... [4m41s elapsed]
aws_vpn_connection.main: Still creating... [4m51s elapsed]
aws_vpn_connection.main: Still creating... [5m1s elapsed]
aws_vpn_connection.main: Still creating... [5m11s elapsed]
aws_vpn_connection.main: Still creating... [5m21s elapsed]
aws_vpn_connection.main: Still creating... [5m31s elapsed]
aws_vpn_connection.main: Still creating... [5m41s elapsed]
aws_vpn_connection.main: Still creating... [5m51s elapsed]
aws_vpn_connection.main: Still creating... [6m1s elapsed]
aws_vpn_connection.main: Creation complete after 6m8s [id=vpn-0067322261c0c79d0]
```

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

vagrant@ubuntu-xenial: ~/Spcm_project\$

☒ New VPC Experience
Tell us what you think

Endpoint Services

NAT Gateways [New](#)

Peering Connections

▼ SECURITY

Network ACLs

Security Groups [New](#)

▼ AWS NETWORK

FIREWALL

Firewalls

Firewall policies

Create Virtual Private Gateway

Actions ▼

Filter by tags and attributes or search by keyword

< > 1 to 1 of 1 > >

Name	ID	State	Type	VPC	ASN (Amazon side)
	vgw-0b912f5ded5427635	attached	ipsec.1	vpc-044b8d71/e448868e	64512

☒ New VPC Experience
Tell us what you think

Endpoint Services

NAT Gateways [New](#)

Peering Connections

▼ SECURITY

Network ACLs

Security Groups [New](#)

▼ AWS NETWORK

FIREWALL

Firewalls

Firewall policies

Network Firewall rule groups

Create Customer Gateway

Actions ▼

Filter by tags and attributes or search by keyword

< > 1 to 1 of 1 > >

Name	ID	State	Type	IP Address	BGP ASN	Certificate ARN
	cgw-0ccac3c55fd886551	available	ipsec.1	172.0.0.12	65000	