STEP-1 Provision VPCs

* Create 3 VPCs
  + VPC-1 has the CIDR block - 10.0.0.0/16
  + VPC-2 has the CIDR block - 20.0.0.0/16
  + VPC-3 has the CIDR block - 30.0.0.0/16

Graphical user interface, text, application, email

Description automatically generated

* Create 3 subnet and attach it to its respective VPC.
  + subnet1 has CIDR block 10.0.0.0/24
  + subnet2 has CIDR block 20.0.0.0/24
  + subnet3 has CIDR block 30.0.0.0/24

Graphical user interface, application, Teams

Description automatically generated

* Provision a Internet gateway
  + connect to VPC-1

Graphical user interface, text, application, Teams

Description automatically generated

* Rename VPC's default route table.

Graphical user interface, text, application, email

Description automatically generated

* Change both the default route tables to add a route to its IGW.

STEP-2 Provision Ec2 instances and security group.

* Security Groups
  + create 3 security groups
  + add inbound rule to open port 22 for ssh 0.0.0.0/0

* Ec2 inances
  + Provision three t2.micro linux ec2 instance
  + ec2-1 with public ip
  + ec2-2 and ec2-3 without public ip
  + Associate them with respective sg

STEP-3 Provision of transit Gateway

* Provision a TGW

Graphical user interface, text, application, Teams

Description automatically generated

* create attachments.

Graphical user interface, application, Teams

Description automatically generated

STEP-4 Modify security groups.

* modify SG to open inbound rule for telnet i.e, port 23
* port 23 should be opened to cidr block of vpc-1 and vpc-2
  + for vpc1- it should open cidr block of vpc2 and vpc3 -> 20.0.0.0/16, 30.0.0.0/16

Graphical user interface, application

Description automatically generated

STEP-5 Modify VPC route table

* Modify route table of both vpc to route traffic to tg
  + for vpc1-it should route traffic for vpc2 ie, 20.0.0.0/16 to transit gateway
  + for vpc2-it should route traffic for vpc1 ie,.10.0.0.0/16 to transit gateway

STEP-6 ssh and telnet

* ssh into ec2-one using public ip
* run
  + sudo yum update
  + sudo yum install telnet
  + telnet <ec2-2 private ip> 22

Provisioning of infrastructure through Terraform.

 provider "aws" {

  region = "us-east-1"

}

#Provisioning VPC-one

resource "aws\_vpc" "slyth-vpc-one" {

  cidr\_block = "10.0.0.0/16"

  tags = {

    Name = "slyth-one"

  }

}

#Provisioning VPC-TWO

resource "aws\_vpc" "slyth-vpc-two" {

  cidr\_block = "20.0.0.0/16"

  tags = {

    Name = "slyth-two"

  }

}

#Provisioning VPC-three

resource "aws\_vpc" "slyth-vpc-three" {

  cidr\_block = "30.0.0.0/16"

  tags = {

    Name = "slyth-three"

  }

}

# Provisioning Subnet-ONE

resource "aws\_subnet" "slyth-subnet-one" {

  vpc\_id     = aws\_vpc.slyth-vpc-one.id

  cidr\_block = "10.0.0.0/24"

  availability\_zone = "us-east-1b"

  tags = {

    Name = "slyth-one"

  }

}

# Provisioning Subnet-TWO

resource "aws\_subnet" "slyth-subnet-two" {

  vpc\_id     = aws\_vpc.slyth-vpc-two.id

  cidr\_block = "20.0.0.0/24"

  availability\_zone = "us-east-1b"

  tags = {

    Name = "slyth-two"

  }

}

# Provisioning Subnet-THREE

resource "aws\_subnet" "slyth-subnet-three" {

  vpc\_id     = aws\_vpc.slyth-vpc-three.id

  cidr\_block = "30.0.0.0/24"

  availability\_zone = "us-east-1b"

  tags = {

    Name = "slyth-three"

  }

}

# Provisioning IGW-ONE

resource "aws\_internet\_gateway" "slyth-igw-one" {

  vpc\_id = aws\_vpc.slyth-vpc-one.id

  tags = {

    Name = "slyth-one"

  }

}

#Provisioning SG-ONE

resource "aws\_security\_group" "slyth-sg-one" {

  name        = "slyth-sg-one"

  description = "Allow TLS inbound traffic"

  vpc\_id      = aws\_vpc.slyth-vpc-one.id

  ingress {

    from\_port        = 22

    to\_port          = 22

    protocol         = "tcp"

    cidr\_blocks      = ["0.0.0.0/0"]

  }

  ingress {

    from\_port        = 23

    to\_port          = 23             #for telnet

    protocol         = "tcp"

    cidr\_blocks      = ["20.0.0.0/16"]

  }

  ingress {

    from\_port        = 23

    to\_port          = 23             #for telnet

    protocol         = "tcp"

    cidr\_blocks      = ["30.0.0.0/16"]

  }

  ingress {

    from\_port        = -1

    to\_port          = -1          #for icmp

    protocol         = "icmp"

    cidr\_blocks      = ["20.0.0.0/16"]

  }

  ingress {

    from\_port        = -1

    to\_port          = -1          #for icmp

    protocol         = "icmp"

    cidr\_blocks      = ["30.0.0.0/16"]

  }

  egress {

    from\_port        = 0

    to\_port          = 0

    protocol         = "-1"

    cidr\_blocks      = ["0.0.0.0/0"]

  }

  tags = {

    Name = "slyth-one"

  }

}

#Provisioning SG-TWO

resource "aws\_security\_group" "slyth-sg-two" {

  name        = "slyth-sg-two"

  description = "Allow TLS inbound traffic"

  vpc\_id      = aws\_vpc.slyth-vpc-two.id

  ingress {

    from\_port        = 22

    to\_port          = 22

    protocol         = "tcp"

    cidr\_blocks      = ["0.0.0.0/0"]

  }

  ingress {

    from\_port        = -1

    to\_port          = -1          #for icmp

    protocol         = "icmp"

    cidr\_blocks      = ["10.0.0.0/16"]

  }

  ingress {

    from\_port        = -1

    to\_port          = -1          #for icmp

    protocol         = "icmp"

    cidr\_blocks      = ["30.0.0.0/16"]

  }

  egress {

    from\_port        = 0

    to\_port          = 0

    protocol         = "-1"

    cidr\_blocks      = ["0.0.0.0/0"]

  }

  tags = {

    Name = "slyth-two"

  }

}

#Provisioning SG-THREE

resource "aws\_security\_group" "slyth-sg-three" {

  name        = "slyth-sg-two"

  description = "Allow TLS inbound traffic"

  vpc\_id      = aws\_vpc.slyth-vpc-three.id

  ingress {

    from\_port        = 22

    to\_port          = 22

    protocol         = "tcp"

    cidr\_blocks      = ["0.0.0.0/0"]

  }

  ingress {

    from\_port        = -1

    to\_port          = -1          #for icmp

    protocol         = "icmp"

    cidr\_blocks      = ["10.0.0.0/16"]

  }

  ingress {

    from\_port        = -1

    to\_port          = -1          #for icmp

    protocol         = "icmp"

    cidr\_blocks      = ["20.0.0.0/16"]

  }

  egress {

    from\_port        = 0

    to\_port          = 0

    protocol         = "-1"

    cidr\_blocks      = ["0.0.0.0/0"]

  }

  tags = {

    Name = "slyth-three"

  }

}

# Create a Transit Gateway

resource "aws\_ec2\_transit\_gateway" "slyth-tgw" {

  tags = {

    Name = "slyth-tgw"

  }

}

# Attachment one

resource "aws\_ec2\_transit\_gateway\_vpc\_attachment" "slyth-attach-one" {

  transit\_gateway\_id = aws\_ec2\_transit\_gateway.slyth-tgw.id

  vpc\_id = aws\_vpc.slyth-vpc-one.id

  subnet\_ids = [aws\_subnet.slyth-subnet-one.id]

  tags = {

    Name = "slyth-attachment-one"

  }

}

# Attachment two

resource "aws\_ec2\_transit\_gateway\_vpc\_attachment" "slyth-attach-two" {

  transit\_gateway\_id = aws\_ec2\_transit\_gateway.slyth-tgw.id

  vpc\_id = aws\_vpc.slyth-vpc-two.id

  subnet\_ids = [aws\_subnet.slyth-subnet-two.id]

  tags = {

    Name = "slyth-attachment-two"

  }

}

# Attachment three

resource "aws\_ec2\_transit\_gateway\_vpc\_attachment" "slyth-attach-three" {

  transit\_gateway\_id = aws\_ec2\_transit\_gateway.slyth-tgw.id

  vpc\_id = aws\_vpc.slyth-vpc-three.id

  subnet\_ids = [aws\_subnet.slyth-subnet-three.id]

  tags = {

    Name = "slyth-attachment-three"

  }

}

# Provisioning first EC2 INSTANCE

resource "aws\_instance" "slyth-ec2-one" {

  ami = "ami-0fa1de1d60de6a97e"

  instance\_type = "t2.micro"

  subnet\_id = aws\_subnet.slyth-subnet-one.id

  vpc\_security\_group\_ids = [aws\_security\_group.slyth-sg-one.id]

  key\_name = "slytherin"

  associate\_public\_ip\_address = true

  tags = {

    Name = "slyth-one"

  }

}

# Provisioning SECOND EC2-instance

resource "aws\_instance" "slyth-ec2-two" {

  ami = "ami-0fa1de1d60de6a97e"

  instance\_type = "t2.micro"

  subnet\_id = aws\_subnet.slyth-subnet-two.id

  vpc\_security\_group\_ids = [aws\_security\_group.slyth-sg-two.id]

  key\_name = "slytherin"

  tags = {

    Name = "slyth-two"

  }

}

# Provisioning THIRD EC2-instance

resource "aws\_instance" "slyth-ec2-three" {

  ami = "ami-0fa1de1d60de6a97e"

  instance\_type = "t2.micro"

  subnet\_id = aws\_subnet.slyth-subnet-three.id

  vpc\_security\_group\_ids = [aws\_security\_group.slyth-sg-three.id]

  key\_name = "slytherin"

  tags = {

    Name = "slyth-three"

  }

}