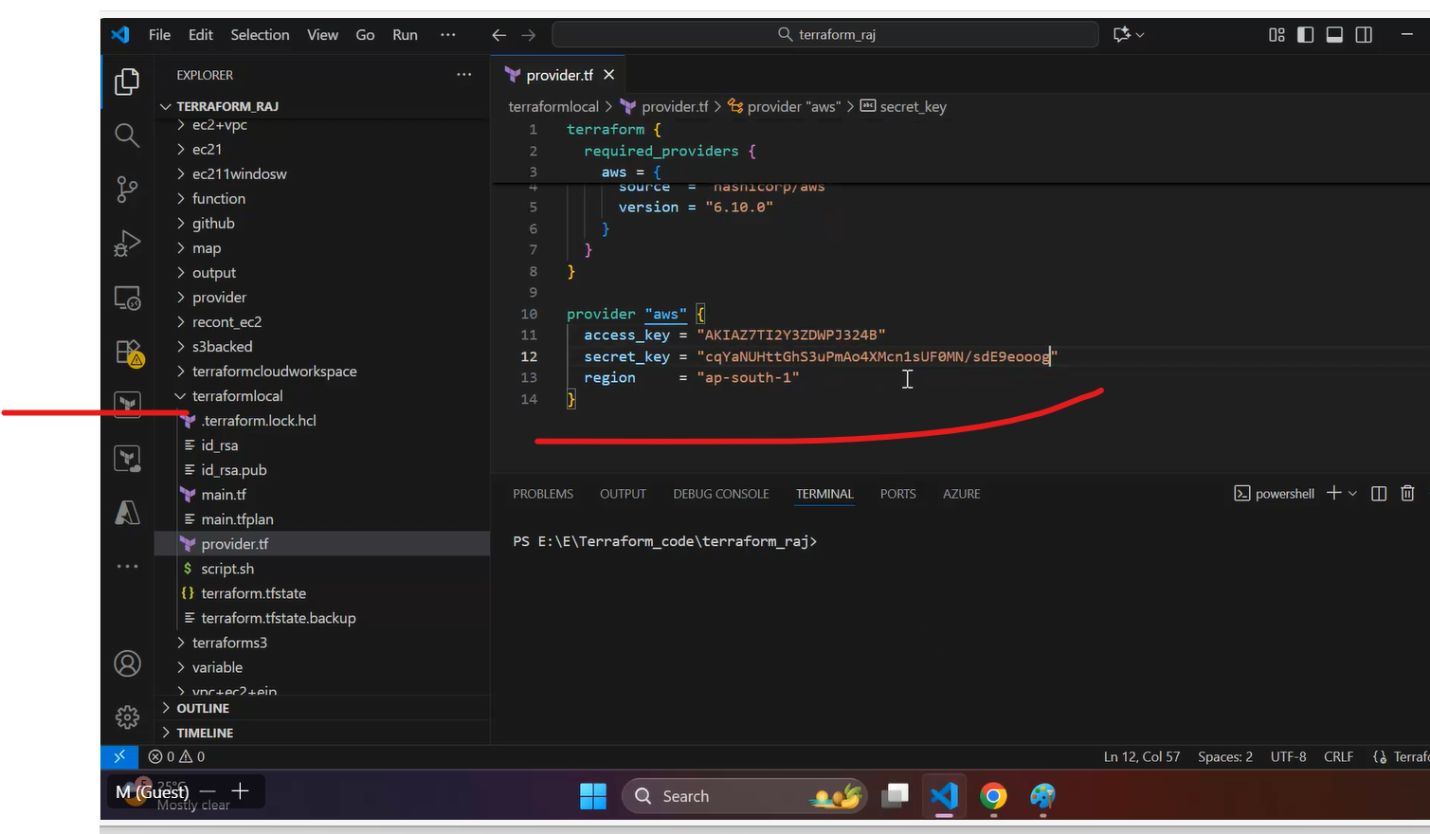
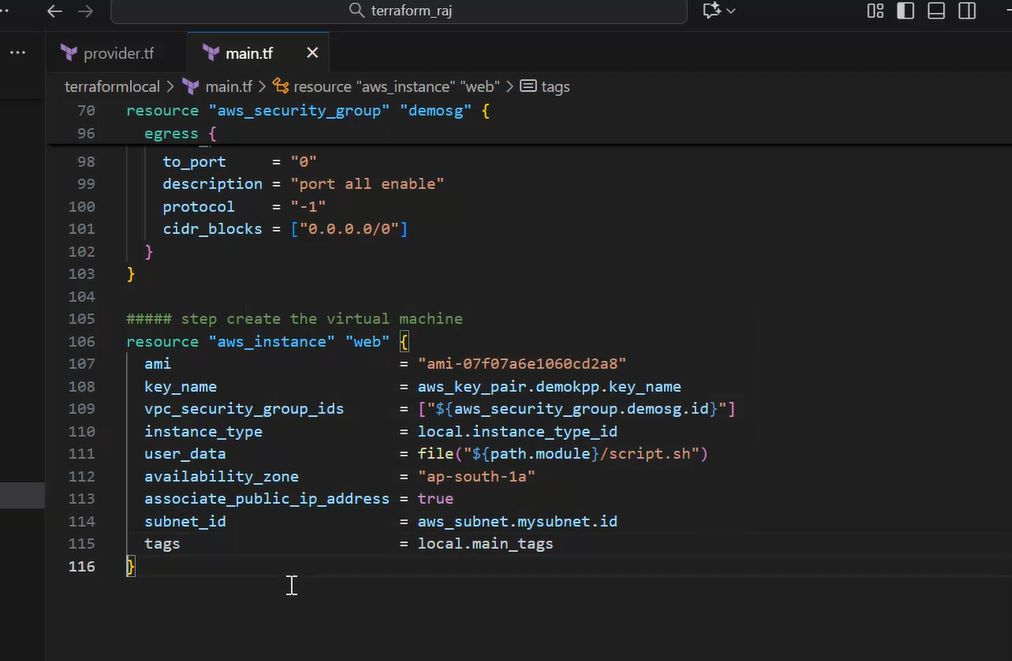
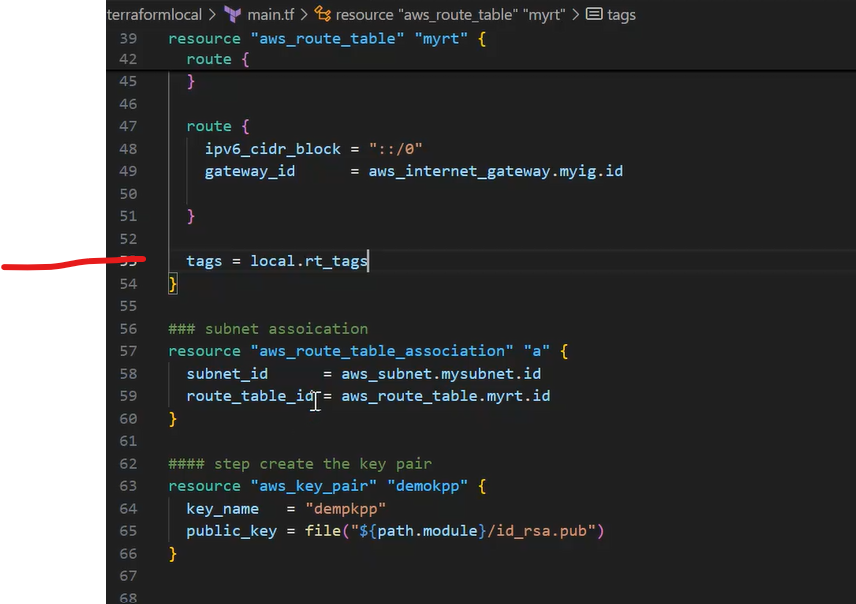
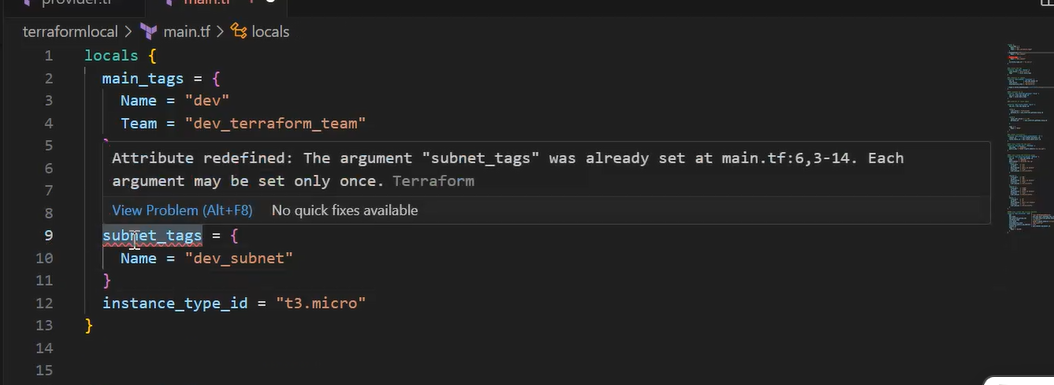
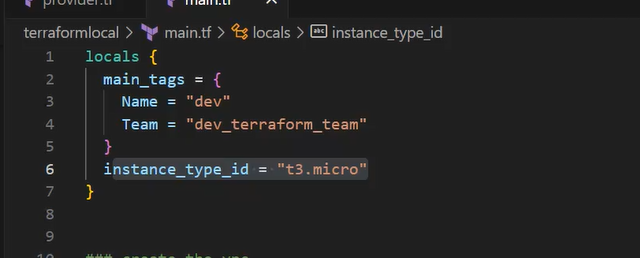
Created new IAM account….{Always creating new IAM account….for security reasons and charges concern}…Terraform Local….Block is the next topic….

****

****locals {

  main\_tags = {

    Name = "dev"

    Team = "dev\_terraform\_team"

  }

  subnet\_tags = {

    Name = "dev\_subnet"

  }

  rt\_tags = {

    Name = "dev\_rt"

  }

  instance\_type\_id   = "t3.micro"

  security\_group\_tag = "dempsgname"

}

### create the vpc

resource "aws\_vpc" "myvpc" {

  cidr\_block = "10.0.0.0/16"

  tags       = local.main\_tags

}

### creation of subnet

resource "aws\_subnet" "mysubnet" {

  vpc\_id            = aws\_vpc.myvpc.id

  cidr\_block        = "10.0.0.0/24"

  availability\_zone = "ap-south-1a"

  tags = local.subnet\_tags

}

#### creation of ig

resource "aws\_internet\_gateway" "myig" {

  vpc\_id = aws\_vpc.myvpc.id

  tags   = local.main\_tags

}

### creation of route table

resource "aws\_route\_table" "myrt" {

  vpc\_id = aws\_vpc.myvpc.id

  route {

    cidr\_block = "0.0.0.0/0"

    gateway\_id = aws\_internet\_gateway.myig.id

  }

  route {

    ipv6\_cidr\_block = "::/0"

    gateway\_id      = aws\_internet\_gateway.myig.id

  }

  tags = local.rt\_tags

}

### subnet assoication

resource "aws\_route\_table\_association" "a" {

  subnet\_id      = aws\_subnet.mysubnet.id

  route\_table\_id = aws\_route\_table.myrt.id

}

#### step create the key pair

resource "aws\_key\_pair" "demokpp" {

  key\_name   = "dempkpp"

  public\_key = file("${path.module}/id\_rsa.pub")

}

#### step create the security group

resource "aws\_security\_group" "demosg" {

  vpc\_id      = aws\_vpc.myvpc.id

  name        = local.security\_group\_tag

  description = "provide the sg"

  ingress {

    from\_port   = "22"

    to\_port     = "22"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    from\_port   = "80"

    to\_port     = "80"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    from\_port   = "3389"

    to\_port     = "3389"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  egress {

    from\_port   = "0"

    to\_port     = "0"

    description = "port all enable"

    protocol    = "-1"

    cidr\_blocks = ["0.0.0.0/0"]

  }

}

##### step create the virtual machine

resource "aws\_instance" "web" {

  ami                         = "ami-07f07a6e1060cd2a8"

  key\_name                    = aws\_key\_pair.demokpp.key\_name

  vpc\_security\_group\_ids      = ["${aws\_security\_group.demosg.id}"]

  instance\_type               = local.instance\_type\_id

  user\_data                   = file("${path.module}/script.sh")

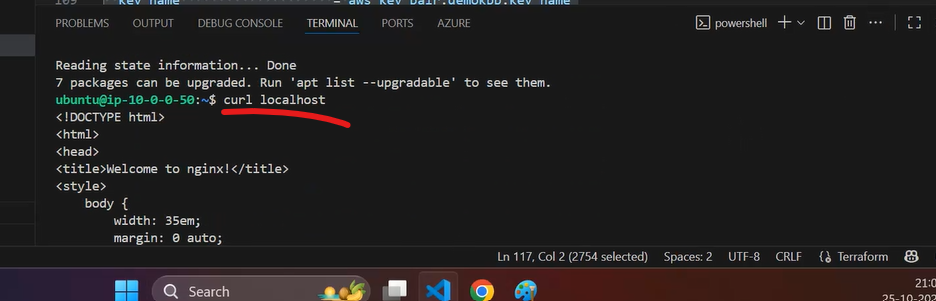
  availability\_zone           = "ap-south-1a"

  associate\_public\_ip\_address = true

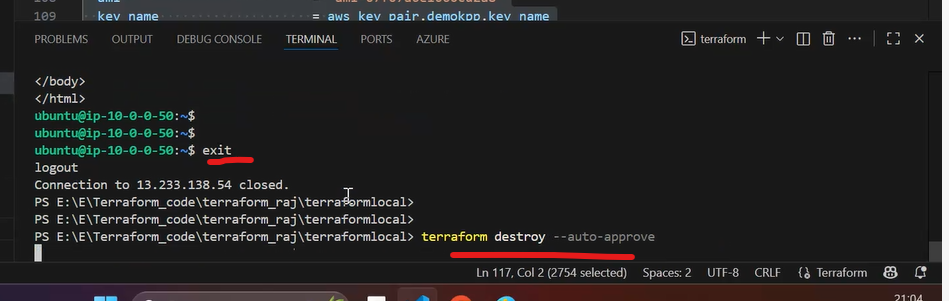
  subnet\_id                   = aws\_subnet.mysubnet.id

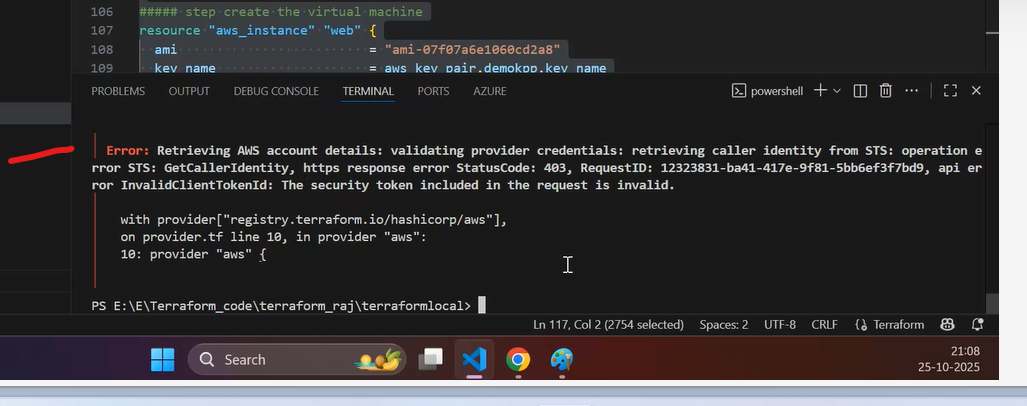
  tags                        = local.main\_tags

}

**Sudo apt update**

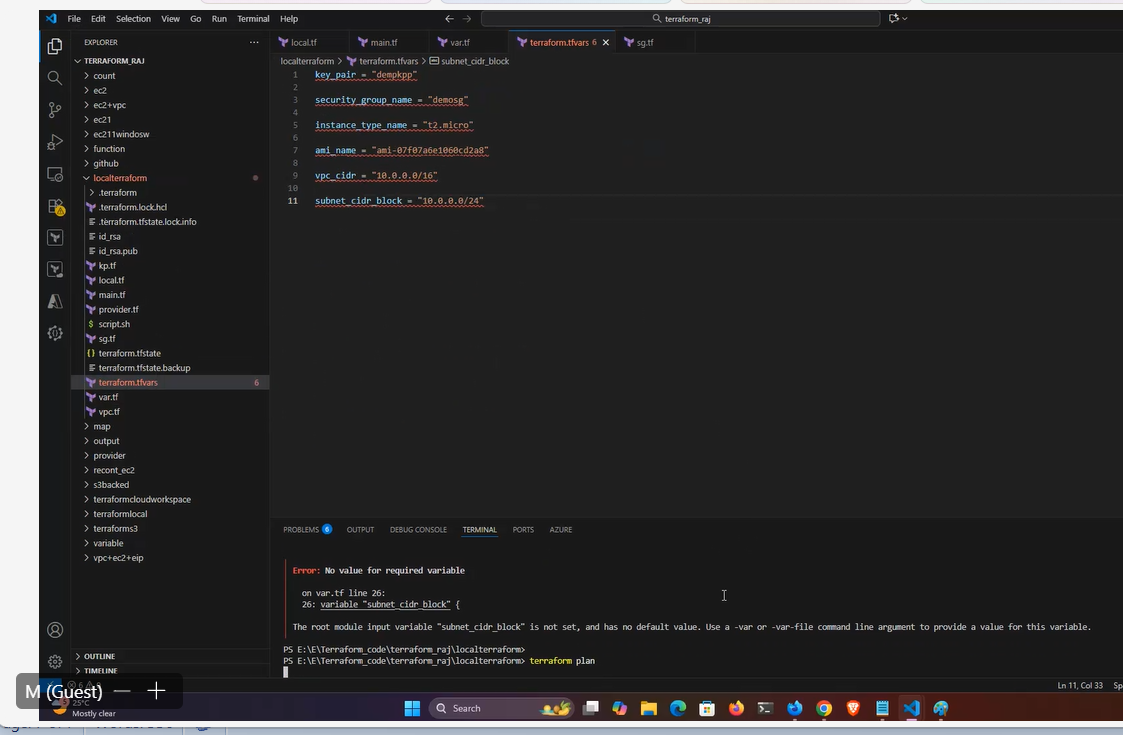
**Access the public IP and noticed nginx is loaded…..**

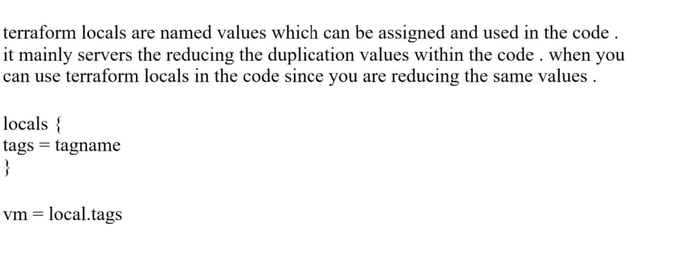
**\**

****

**He removed IAM account….Terraform init and then Terraform validate….he got the above error….**

**Next topic…he distributed the files using variables.tf and tfvars file…**

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

locals {

  main\_tags = {

    Name = "dev"

    Team = "dev\_terraform\_team"

  }

  subnet\_tags = {

    Name = "dev\_subnet"

  }

  rt\_tags = {

    Name = "dev\_rt"

  }

  instance\_type\_id   = "t3.micro"

  security\_group\_tag = "dempsgname"

}

### create the vpc

resource "aws\_vpc" "myvpc" {

  cidr\_block = "10.0.0.0/16"

  tags       = local.main\_tags

}

### creation of subnet

resource "aws\_subnet" "mysubnet" {

  vpc\_id            = aws\_vpc.myvpc.id

  cidr\_block        = "10.0.0.0/24"

  availability\_zone = "ap-south-1a"

  tags = local.subnet\_tags

}

#### creation of ig

resource "aws\_internet\_gateway" "myig" {

  vpc\_id = aws\_vpc.myvpc.id

  tags   = local.main\_tags

}

### creation of route table

resource "aws\_route\_table" "myrt" {

  vpc\_id = aws\_vpc.myvpc.id

  route {

    cidr\_block = "0.0.0.0/0"

    gateway\_id = aws\_internet\_gateway.myig.id

  }

  route {

    ipv6\_cidr\_block = "::/0"

    gateway\_id      = aws\_internet\_gateway.myig.id

  }

  tags = local.rt\_tags

}

### subnet assoication

resource "aws\_route\_table\_association" "a" {

  subnet\_id      = aws\_subnet.mysubnet.id

  route\_table\_id = aws\_route\_table.myrt.id

}

#### step create the key pair

resource "aws\_key\_pair" "demokpp" {

  key\_name   = "dempkpp"

  public\_key = file("${path.module}/id\_rsa.pub")

}

#### step create the security group

resource "aws\_security\_group" "demosg" {

  vpc\_id      = aws\_vpc.myvpc.id

  name        = local.security\_group\_tag

  description = "provide the sg"

  ingress {

    from\_port   = "22"

    to\_port     = "22"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    from\_port   = "80"

    to\_port     = "80"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    from\_port   = "3389"

    to\_port     = "3389"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  egress {

    from\_port   = "0"

    to\_port     = "0"

    description = "port all enable"

    protocol    = "-1"

    cidr\_blocks = ["0.0.0.0/0"]

  }

}

##### step create the virtual machine

resource "aws\_instance" "web" {

  ami                         = "ami-07f07a6e1060cd2a8"

  key\_name                    = aws\_key\_pair.demokpp.key\_name

  vpc\_security\_group\_ids      = ["${aws\_security\_group.demosg.id}"]

  instance\_type               = local.instance\_type\_id

  user\_data                   = file("${path.module}/script.sh")

  availability\_zone           = "ap-south-1a"

  associate\_public\_ip\_address = true

  subnet\_id                   = aws\_subnet.mysubnet.id

  tags                        = local.main\_tags

}

#### step create the key pair

resource "aws\_key\_pair" "demokpp" {

  key\_name   = local.key\_pair\_name

  public\_key = file("${path.module}/id\_rsa.pub")

}

locals {

  main\_tags = {

    Name = "dev"

    Team = "dev\_terraform\_team"

  }

  subnet\_tags = {

    Name = "dev\_subnet"

  }

  rt\_tags = {

    Name = "dev\_rt"

  }

  key\_pair\_name          = var.key\_pair

  security\_group\_name\_id = var.security\_group\_name

  instance\_type\_name\_id  = var.instance\_type\_name

  ami\_name\_id            = var.ami\_name

  vpc\_id                 = var.vpc\_cidr

  subnet\_id              = var.subnet\_cidr\_block

  ##  instance\_type\_id   = "t3.micro"

  ###  security\_group\_tag = "dempsgname"

}

##### step create the virtual machine

resource "aws\_instance" "web" {

  ami                         = local.ami\_name\_id

  key\_name                    = aws\_key\_pair.demokpp.key\_name

  vpc\_security\_group\_ids      = ["${aws\_security\_group.demosg.id}"]

  instance\_type               = local.instance\_type\_name\_id

  user\_data                   = file("${path.module}/script.sh")

  availability\_zone           = "ap-south-1a"

  associate\_public\_ip\_address = true

  subnet\_id                   = aws\_subnet.mysubnet.id

  tags                        = local.main\_tags

}

#### step create the security group

resource "aws\_security\_group" "demosg" {

  vpc\_id      = aws\_vpc.myvpc.id

  name        = local.security\_group\_name\_id

  description = "provide the sg"

  ingress {

    from\_port   = "22"

    to\_port     = "22"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    from\_port   = "80"

    to\_port     = "80"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  ingress {

    from\_port   = "3389"

    to\_port     = "3389"

    description = "port 22 enable"

    protocol    = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

  }

  egress {

    from\_port   = "0"

    to\_port     = "0"

    description = "port all enable"

    protocol    = "-1"

    cidr\_blocks = ["0.0.0.0/0"]

  }

}

key\_pair = "dempkpp"

security\_group\_name = "demosg"

instance\_type\_name = "t2.micro"

ami\_name = "ami-07f07a6e1060cd2a8"

vpc\_cidr = "10.0.0.0/16"

subnet\_cidr\_block = "10.0.0.0/24"

variable "key\_pair" {

  type        = string

  description = "provide the key pair"

}

variable "security\_group\_name" {

  type        = string

  description = "provide the sg name "

}

variable "instance\_type\_name" {

  type        = string

  description = "provide the instance type "

}

variable "ami\_name" {

  type        = string

  description = "provide the ami details"

}

variable "vpc\_cidr" {

  type        = string

  description = "provide the vidr of vpc"

}

variable "subnet\_cidr\_block" {

  type        = string

  description = "provde the subnet"

}

### create the vpc

resource "aws\_vpc" "myvpc" {

  cidr\_block = local.vpc\_id

  tags       = local.main\_tags

}

### creation of subnet

resource "aws\_subnet" "mysubnet" {

  vpc\_id            = aws\_vpc.myvpc.id

  cidr\_block        = local.subnet\_id

  availability\_zone = "ap-south-1a"

  tags = local.subnet\_tags

}

#### creation of ig

resource "aws\_internet\_gateway" "myig" {

  vpc\_id = aws\_vpc.myvpc.id

  tags   = local.main\_tags

}

### creation of route table

resource "aws\_route\_table" "myrt" {

  vpc\_id = aws\_vpc.myvpc.id

  route {

    cidr\_block = "0.0.0.0/0"

    gateway\_id = aws\_internet\_gateway.myig.id

  }

  route {

    ipv6\_cidr\_block = "::/0"

    gateway\_id      = aws\_internet\_gateway.myig.id

  }

  tags = local.rt\_tags

}

### subnet assoication

resource "aws\_route\_table\_association" "a" {

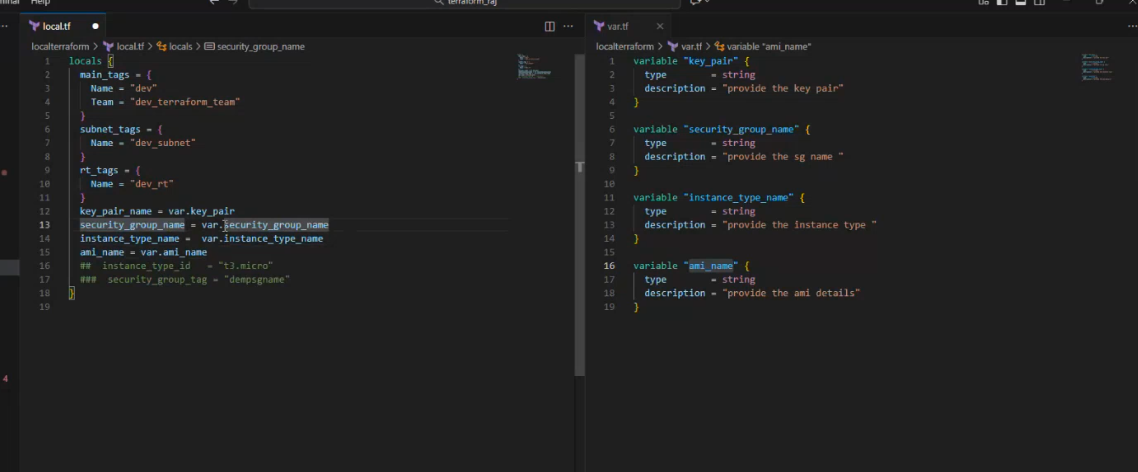
  subnet\_id      = aws\_subnet.mysubnet.id

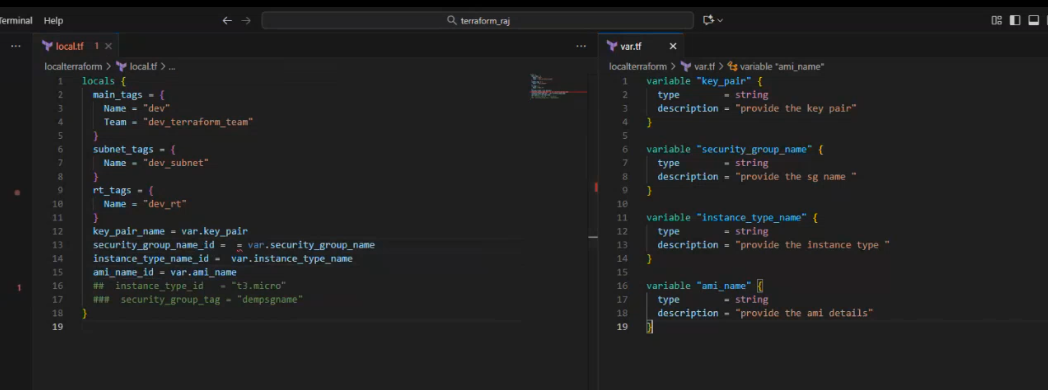
  route\_table\_id = aws\_route\_table.myrt.id

}

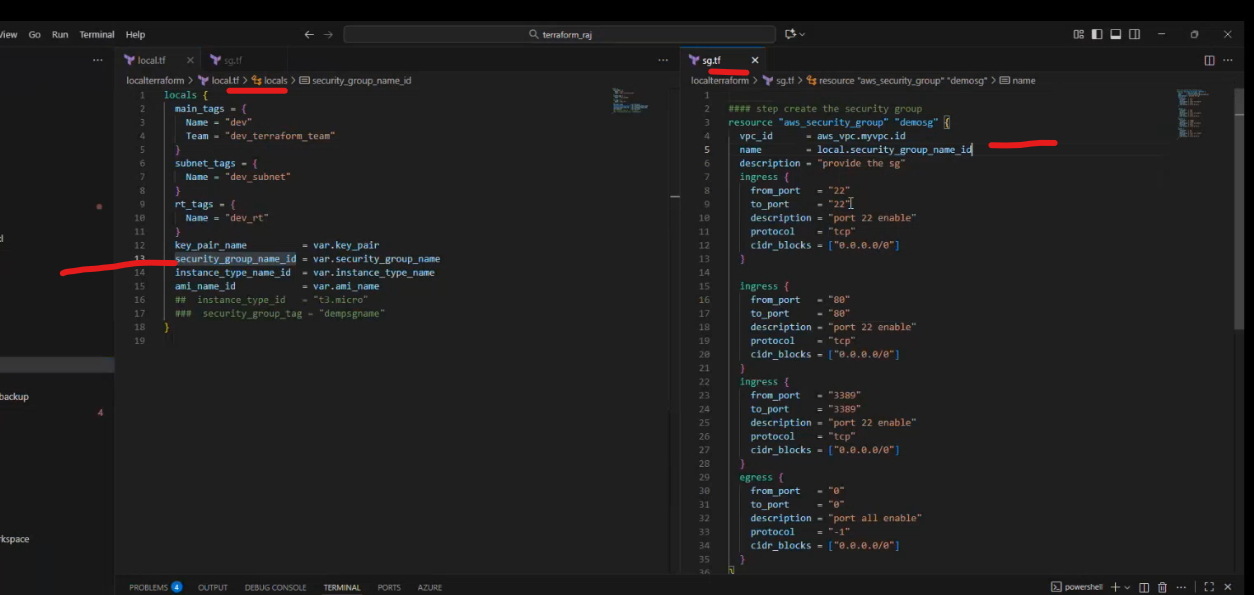
Summary:

In Part3:

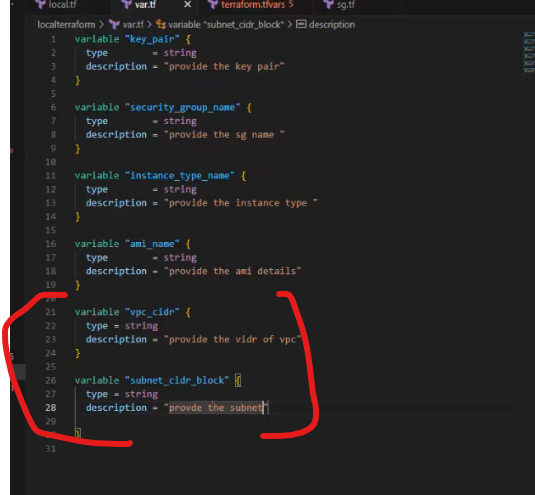




He added 4 variables to local.tf and then used in other tf files as local.[] instead of var.[]



He created 2 new variables for VPC block, Subnet block



He added these 2 to local block…..local block has 6 values now…..

