provider "aws" {

region = "us-east-1"  
}  
resource "aws\_vpc" "redshift\_vpc" {  
  
 cidr\_block = "10.0.0.0/16"  
  
 instance\_tenancy = "default"  
  
 tags = {  
  
 Name = "redshift-vpc"  
  
 }  
  
}  
  
  
resource "aws\_internet\_gateway" "redshift\_vpc\_igw" {  
  
 vpc\_id = aws\_vpc.redshift\_vpc.id  
  
 depends\_on = [  
  
 aws\_vpc.redshift\_vpc  
  
 ]  
  
}  
  
resource "aws\_default\_security\_group" "redshift\_security\_group" {  
  
 vpc\_id = aws\_vpc.redshift\_vpc.id  
  
 ingress {  
  
 from\_port = 5439  
  
 to\_port = 5439  
  
 protocol = "tcp"  
  
 cidr\_blocks = ["0.0.0.0/0"]  
  
 }  
  
  
 tags = {  
  
 Name = "redshift-sg"  
  
 }  
  
 depends\_on = [  
  
 aws\_vpc.redshift\_vpc  
  
 ]  
  
}  
  
resource "aws\_subnet" "redshift\_subnet\_1" {  
  
 vpc\_id = aws\_vpc.redshift\_vpc.id  
  
 cidr\_block = "10.0.1.0/24"  
  
 availability\_zone = "us-east-1a"  
  
 map\_public\_ip\_on\_launch = "true"  
  
 tags = {  
  
 Name = "redshift-subnet-1"  
  
 }  
  
 depends\_on = [  
  
 aws\_vpc.redshift\_vpc  
  
 ]  
  
}  
  
resource "aws\_subnet" "redshift\_subnet\_2" {  
  
 vpc\_id = aws\_vpc.redshift\_vpc.id  
  
 cidr\_block = "10.0.2.0/24"  
  
 availability\_zone = "us-east-1b"  
  
 map\_public\_ip\_on\_launch = "true"  
  
 tags = {  
  
 Name = "redshift-subnet-2"  
  
 }  
  
 depends\_on = [  
  
 aws\_vpc.redshift\_vpc  
  
 ]  
  
}  
  
resource "aws\_redshift\_subnet\_group" "redshift\_subnet\_group" {  
  
 name = "redshift-subnet-group"  
  
 subnet\_ids = [aws\_subnet.redshift\_subnet\_1.id,  
  
aws\_subnet.redshift\_subnet\_2.id]  
  
 tags = {  
  
 environment = "dev"  
  
 Name = "redshift-subnet-group"  
  
 }  
  
}  
  
  
resource "aws\_iam\_policy" "redshift\_iam\_policy" {  
 name = "redshift\_iam\_policy"  
 path = "/"  
 description = "Allow "  
  
 policy = jsonencode({  
 "Version" : "2012-10-17",  
 "Statement" : [  
 {  
 "Sid" : "VisualEditor0",  
 "Effect" : "Allow",  
 "Action" : [  
 "s3:PutObject",  
 "s3:GetObject",  
 "s3:ListBucket",  
 "s3:DeleteObject"  
 ],  
 "Resource" : [  
 "arn:aws:s3:::\*/\*",  
 "arn:aws:s3:::my-bucket-name"  
 ]  
 }  
 ]  
 })  
}  
  
resource "aws\_iam\_role" "redshift\_iam\_role" {  
 name = "redshift\_iam\_role"  
  
 assume\_role\_policy = jsonencode({  
 Version = "2012-10-17"  
 Statement = [  
 {  
 Action = "sts:AssumeRole"  
 Effect = "Allow"  
 Sid = ""  
 Principal = {  
 Service = "ec2.amazonaws.com"  
 }  
 },  
 ]  
 })  
}  
  
resource "aws\_redshift\_cluster" "default" {  
  
 cluster\_identifier = "sample-cluster"  
  
 database\_name = "database\_1"  
  
 master\_username = "sample"  
  
 master\_password = "Sample12345678"  
  
 node\_type = "dc2.large"  
  
 cluster\_type = "single-node"  
  
 cluster\_subnet\_group\_name = aws\_redshift\_subnet\_group.redshift\_subnet\_group.id  
  
 skip\_final\_snapshot = true  
  
 iam\_roles = [aws\_iam\_role.redshift\_iam\_role.arn]  
  
 depends\_on = [  
  
 aws\_vpc.redshift\_vpc,  
  
 aws\_default\_security\_group.redshift\_security\_group,  
  
 aws\_redshift\_subnet\_group.redshift\_subnet\_group,  
  
 aws\_iam\_role.redshift\_iam\_role  
  
 ]  
  
}