Connect the Cloud9 IDE & CICD VPC to the EKS VPC

```
cd ~/environment/tfekscode/c9net
```

Run the script to Initialize the Terraform

```
terraform init -upgrade
Initializing the backend...

** OUTPUT TRUNCATED FOR BREVITY  as similar to previous examples **
```

Validate the Terraform code

```
terraform validate
Success! The configuration is valid.
```

Plan the deployment:

```
terraform plan -out tfplan
data.aws_ssm_parameter.rtb-isol: Reading...
data.aws_ssm_parameter.sub-p1: Reading...
....
data.aws_iam_instance_profile.c9ip: Read complete after 0s [id=AIPAWM4ZLYGW4L62AAQFN]

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
  + create

Terraform will perform the following actions:
  # aws_route.rt-cicd will be created
  + resource "aws_route" "rt-cicd" {
      + destination_cidr_block    = (sensitive value)
      + id                        = (known after apply)
      + instance_id               = (known after apply)
      + instance_owner_id         = (known after apply)
      + network_interface_id      = (known after apply)
      + origin                    = (known after apply)
      + route_table_id            = "rtb-05357448e65d6c821"
      + state                     = (known after apply)
      + vpc_peering_connection_id = (known after apply)
    }

  # aws_route.rt-cicd-isol will be created
  + resource "aws_route" "rt-cicd-isol" {
      + destination_cidr_block    = "172.30.0.0/24"
      + id                        = (known after apply)
      + instance_id               = (known after apply)
      + instance_owner_id         = (known after apply)
      + network_interface_id      = (known after apply)
      + origin                    = (known after apply)
      + route_table_id            = (sensitive value)
      + state                     = (known after apply)
      + vpc_peering_connection_id = (known after apply)
    }


** OUTPUT TRUNCATED FOR BREVITY **


  # aws_vpc_peering_connection.def-peer will be created
  + resource "aws_vpc_peering_connection" "def-peer" {
      + accept_status = (known after apply)
      + auto_accept   = true
      + id            = (known after apply)
      + peer_owner_id = (known after apply)
      + peer_region   = (known after apply)
      + peer_vpc_id   = (sensitive value)
      + tags_all      = (known after apply)
      + vpc_id        = "vpc-01d527302b583115a"
    }

Plan: 28 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + c9lab      = " true "
  + c9role     = "arn:aws:iam::440018911661:role/eksworkshop-admin"
  + cicdpeerid = (known after apply)
  + peerid     = (known after apply)


Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan"
```

You can see from the plan the following resources will be created to link the Cloud9 IDE to the EKS VPC

- VPC peering.
- VPC route table entries for the default and EKS subnets:
  - A route to 172.31 via the VPC peering for the EKS VPC.
  - A route to 10.0 via the VPC peering for the Default VPC.
- VPC route table entries for the CICD subnets and EKS subnets:
  - A route to 172.30 via the VPC peering for the EKS VPC.
  - A route to 10.0 via the VPC peering for the VPC.
- A security group rule for the Cloud9 Security group allowing traffic from 10.0.x.x.

- A security group rule for the EKS Worker Nodes Security Group allowing traffic in port 22 from the Cloud9 IDE Security Group.
- A security group rule for the EKS Cluster Security Group allowing traffic in port 443 from the Cloud9 IDE Security Group

Build the environment:

```
terraform apply tfplan
aws_security_group_rule.eks-all-cicd: Creating...
aws_security_group_rule.sg-cicd-self: Creating...
aws_security_group_rule.eks-node-self: Creating...
aws_vpc_peering_connection.def-peer: Creating..

** OUTPUT TRUNCATED FOR BREVITY **

aws_route.rt-cicd-isol: Creation complete after 0s [id=r-rtb-0c1fd9874373efefc3963506374]
aws_route.rt-cicd: Creation complete after 1s [id=r-rtb-05357448e65d6c821653952448]
aws_route.rt-def-cicd: Creation complete after 1s [id=r-rtb-02e7d960f968c63fa653952448]
aws_route.rt-eks6: Creation complete after 1s [id=r-rtb-0e6663bb3e496e6163963506374]
aws_security_group_rule.eks-all-egress: Creation complete after 2s [id=sgrule-3294423239]
aws_route.rt-eks5: Creation complete after 1s [id=r-rtb-0efdd4f535b00c0b43963506374]
aws_route.rt-eks4: Creation complete after 1s [id=r-rtb-053c87ea4ac5f33fe3963506374]

Apply complete! Resources: 28 added, 0 changed, 0 destroyed.

Outputs:

c9lab = " true "
c9role = "arn:aws:iam::440018911661:role/eksworkshop-admin"
cicdpeerid = "pcx-0bce1dd259fb6bf0d"
peerid = "pcx-0115cd6f8504827c3"
```

The above links the Cloud9 IDE and the CICD VPC to the EKS private VPC and allows the required traffic to flow so you can communicate with the cluster from the Cloud9 IDE in the default VPC and the CodeBuild ENI that will be provisioned in the private subnet of the CICD VPC.