// Define an IAM role for the EKS cluster control plane

resource "aws\_iam\_role" "eks\_cluster" {

name = "eks-cluster"

// Specify the permissions for assuming this role

assume\_role\_policy = <<POLICY

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "eks.amazonaws.com"

},

"Action": "sts:AssumeRole"

}

]

}

POLICY

}

// Attach AmazonEKSClusterPolicy to the IAM role created for EKS cluster

resource "aws\_iam\_role\_policy\_attachment" "AmazonEKSClusterPolicy" {

policy\_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"

role = aws\_iam\_role.eks\_cluster.name

}

// Attach AmazonEKSServicePolicy to the IAM role created for EKS cluster

resource "aws\_iam\_role\_policy\_attachment" "AmazonEKSServicePolicy" {

policy\_arn = "arn:aws:iam::aws:policy/AmazonEKSServicePolicy"

role = aws\_iam\_role.eks\_cluster.name

}

// Create an EKS cluster

resource "aws\_eks\_cluster" "aws\_eks" {

name = "eks\_cluster\_demo"

role\_arn = aws\_iam\_role.eks\_cluster.arn

// Configure VPC for the EKS cluster

vpc\_config {

subnet\_ids = ["subnet-4b808830", "subnet-d728d9bf"]

}

// Add tags to the EKS cluster for identification

tags = {

Name = "EKS\_demo"

}

}

// Define an IAM role for EKS worker nodes

resource "aws\_iam\_role" "eks\_nodes" {

name = "eks-node-group-demo"

// Specify the permissions for assuming this role

assume\_role\_policy = <<POLICY

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "ec2.amazonaws.com"

},

"Action": "sts:AssumeRole"

}

]

}

POLICY

}

// Attach AmazonEKSWorkerNodePolicy to the IAM role created for EKS worker nodes

resource "aws\_iam\_role\_policy\_attachment" "AmazonEKSWorkerNodePolicy" {

policy\_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"

role = aws\_iam\_role.eks\_nodes.name

}

// Attach AmazonEKS\_CNI\_Policy to the IAM role created for EKS worker nodes

resource "aws\_iam\_role\_policy\_attachment" "AmazonEKS\_CNI\_Policy" {

policy\_arn = "arn:aws:iam::aws:policy/AmazonEKS\_CNI\_Policy"

role = aws\_iam\_role.eks\_nodes.name

}

// Attach AmazonEC2ContainerRegistryReadOnly to the IAM role created for EKS worker nodes

resource "aws\_iam\_role\_policy\_attachment" "AmazonEC2ContainerRegistryReadOnly" {

policy\_arn = "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"

role = aws\_iam\_role.eks\_nodes.name

}

// Create an EKS node group

resource "aws\_eks\_node\_group" "node" {

cluster\_name = aws\_eks\_cluster.aws\_eks.name

node\_group\_name = "node\_demo"

node\_role\_arn = aws\_iam\_role.eks\_nodes.arn

subnet\_ids = ["subnet-4b808830", "subnet-d728d9bf"]

// Configure scaling options for the node group

scaling\_config {

desired\_size = 1

max\_size = 1

min\_size = 1

}

// Ensure that the creation of the node group depends on the IAM role policies being attached

depends\_on = [

aws\_iam\_role\_policy\_attachment.AmazonEKSWorkerNodePolicy,

aws\_iam\_role\_policy\_attachment.AmazonEKS\_CNI\_Policy,

aws\_iam\_role\_policy\_attachment.AmazonEC2ContainerRegistryReadOnly,

]

}

Replace the Subnet ID’s with your default VPC’s subnet ID’s in both the**“aws\_eks\_node\_group”** and **“aws\_eks\_cluster”** resource blocks.

* To do so, open the **AWS console** and go to **VPC**
* Open Subnet, make sure the **eu-west-3** region is selected as in the above code we have used the **eu-west-3** region. If you are using a different region, use the subnet ID according to your region.
* Copy the subnet IDs and paste them in the terraform.

You can create an output block to display cluster name

output "eks\_cluster\_name" {

value = aws\_eks\_cluster.aws\_eks.name

}

Save the file and run