

eks-cluster.tf

```
module "eks" {
  source      = "terraform-aws-modules/eks/aws"
  version     = "20.8.4"
  cluster_name = local.cluster_name
  cluster_version = var.kubernetes_version
  subnet_ids   = module.vpc.private_subnets

  enable_irs = true

  tags = {
    cluster = "demo"
  }

  vpc_id = module.vpc.vpc_id

  eks_managed_node_group_defaults = {
    ami_type      = "AL2_x86_64"
    instance_types = ["t3.medium"]
    vpc_security_group_ids = [aws_security_group.all_worker_mgmt.id]
  }

  eks_managed_node_groups = {

    node_group = {
      min_size   = 2
      max_size   = 6
      desired_size = 2
    }
  }
}
```

outputs.tf

```
output "cluster_id" {
  description = "EKS cluster ID."
  value       = module.eks.cluster_id
}
```

```

output "cluster_endpoint" {
  description = "Endpoint for EKS control plane."
  value      = module.eks.cluster_endpoint
}

output "cluster_security_group_id" {
  description = "Security group ids attached to the cluster control plane."
  value      = module.eks.cluster_security_group_id
}

output "region" {
  description = "AWS region"
  value      = var.aws_region
}

output "oidc_provider_arn" {
  value = module.eks.oidc_provider_arn
}

#output "zz_update_kubeconfig_command" {
#  value = "aws eks update-kubeconfig --name " + module.eks.cluster_id
#  value = format("%s %s %s %s", "aws eks update-kubeconfig --name", module.eks.cluster_id,
#    "--region", var.aws_region)
#}

```

security-groups.tf

```

resource "aws_security_group" "all_worker_mgmt" {
  name_prefix = "all_worker_management"
  vpc_id      = module.vpc.vpc_id
}

resource "aws_security_group_rule" "all_worker_mgmt_ingress" {
  description      = "allow inbound traffic from eks"
  from_port        = 0
  protocol         = "-1"
  to_port          = 0
  security_group_id = aws_security_group.all_worker_mgmt.id
  type             = "ingress"
  cidr_blocks = [
    "10.0.0.0/8",

```

```
"172.16.0.0/12",
"192.168.0.0/16",
]
}

resource "aws_security_group_rule" "all_worker_mgmt_egress" {
  description    = "allow outbound traffic to anywhere"
  from_port      = 0
  protocol       = "-1"
  security_group_id = aws_security_group.all_worker_mgmt.id
  to_port        = 0
  type           = "egress"
  cidr_blocks    = ["0.0.0.0/0"]
}
```

variables.tf

```
variable "kubernetes_version" {
  default    = 1.27
  description = "kubernetes version"
}

variable "vpc_cidr" {
  default    = "10.0.0.0/16"
  description = "default CIDR range of the VPC"
}

variable "aws_region" {
  default = "us-west-1"
  description = "aws region"
}
```

versions.tf

```
terraform {
  required_version = ">= 0.12"
  required_providers {
    random = {
      source = "hashicorp/random"
      version = "~> 3.1.0"
    }
  }
}
```

```
}
kubernetes = {
  source = "hashicorp/kubernetes"
  version = ">=2.7.1"
}
aws = {
  source = "hashicorp/aws"
  version = ">= 3.68.0"
}
local = {
  source = "hashicorp/local"
  version = "~> 2.1.0"
}
null = {
  source = "hashicorp/null"
  version = "~> 3.1.0"
}
cloudinit = {
  source = "hashicorp/cloudinit"
  version = "~> 2.2.0"
}
}
}
```

vpc.tf

```
provider "aws" {
  region = var.aws_region
}

data "aws_availability_zones" "available" {}

locals {
  cluster_name = "abhi-eks-${random_string.suffix.result}"
}

resource "random_string" "suffix" {
  length = 8
  special = false
}
```

```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"
  version = "5.7.0"

  name          = "abhi-eks-vpc"
  cidr          = var.vpc_cidr
  azs           = data.aws_availability_zones.available.names
  private_subnets = ["10.0.1.0/24", "10.0.2.0/24"]
  public_subnets  = ["10.0.4.0/24", "10.0.5.0/24"]
  enable_nat_gateway = true
  single_nat_gateway = true
  enable_dns_hostnames = true
  enable_dns_support  = true

  tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
  }

  public_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/elb"                      = "1"
  }

  private_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/internal-elb"             = "1"
  }
}
```
