```
[stan@Stans-MacBook-Air-2 dev % terraform plan
module.eks.data.aws_availability_zones.available: Reading...
module.eks.data.aws_availability_zones.available: Read complete after 1s [id=us-east-1]

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:

  # module.eks.aws_eks_cluster.eks will be created
  + resource "aws_eks_cluster" "eks" {
      + arn                       = (known after apply)
      + bootstrap_self_managed_addons = true
      + certificate_authority     = (known after apply)
      + cluster_id                = (known after apply)
      + created_at                = (known after apply)
      + endpoint                  = (known after apply)
      + id                        = (known after apply)
      + identity                  = (known after apply)
      + name                      = "dev-eks"
      + platform_version          = (known after apply)
      + region                    = "us-east-1"
      + role_arn                  = (known after apply)
      + status                    = (known after apply)
      + tags_all                  = (known after apply)
      + version                   = (known after apply)

      + access_config (known after apply)

      + kubernetes_network_config (known after apply)

      + upgrade_policy (known after apply)

      + vpc_config {
          + cluster_security_group_id = (known after apply)
          + endpoint_private_access   = false
          + endpoint_public_access    = true
          + public_access_cidrs       = (known after apply)
          + subnet_ids                = (known after apply)
          + vpc_id                    = (known after apply)
        }
    }

  # module.eks.aws_eks_node_group.node_group will be created
  + resource "aws_eks_node_group" "node_group" {
      + ami_type              = (known after apply)
      + arn                   = (known after apply)
      + capacity_type         = (known after apply)
      + cluster_name          = "dev-eks"
      + disk_size             = (known after apply)
      + id                    = (known after apply)
      + instance_types        = (known after apply)
      + node_group_name       = "dev-eks-node-group"
      + node_group_name_prefix = (known after apply)
      + node_role_arn         = (known after apply)
      + region                = "us-east-1"
      + release_version       = (known after apply)
      + resources             = (known after apply)
      + status                = (known after apply)
      + subnet_ids            = (known after apply)
      + tags_all              = (known after apply)
      + version               = (known after apply)
```

```
    + node_repair_config (known after apply)

    + scaling_config {
        + desired_size = 2
        + max_size     = 4
        + min_size     = 1
      }

    + update_config (known after apply)
  }

# module.eks.aws_iam_role.eks_cluster will be created
+ resource "aws_iam_role" "eks_cluster" {
    + arn                   = (known after apply)
    + assume_role_policy    = jsonencode(
          {
            + Statement = [
                + {
                    + Action    = "sts:AssumeRole"
                    + Effect    = "Allow"
                    + Principal = {
                        + Service = "eks.amazonaws.com"
                      }
                  },
              ]
            + Version   = "2012-10-17"
          }
      )
    + create_date           = (known after apply)
    + force_detach_policies = false
    + id                    = (known after apply)
    + managed_policy_arns   = (known after apply)
    + max_session_duration  = 3600
    + name                  = "dev-eks-cluster-role"
    + name_prefix           = (known after apply)
    + path                  = "/"
    + tags                  = {
        + "Name" = "dev-eks-cluster-role"
      }
    + tags_all              = {
        + "Name" = "dev-eks-cluster-role"
      }
    + unique_id             = (known after apply)

    + inline_policy (known after apply)
  }

# module.eks.aws_iam_role.eks_node_group will be created
+ resource "aws_iam_role" "eks_node_group" {
    + arn                   = (known after apply)
    + assume_role_policy    = jsonencode(
          {
            + Statement = [
                + {
                    + Action    = "sts:AssumeRole"
                    + Effect    = "Allow"
                    + Principal = {
                        + Service = "ec2.amazonaws.com"
                      }
                  },
              ]
            + Version   = "2012-10-17"
```

```
      )
    + create_date           = (known after apply)
    + force_detach_policies = false
    + id                    = (known after apply)
    + managed_policy_arns   = (known after apply)
    + max_session_duration  = 3600
    + name                  = "dev-eks-nodegroup-role"
    + name_prefix           = (known after apply)
    + path                  = "/"
    + tags                  = {
        + "Name" = "dev-eks-nodegroup-role"
      }
    + tags_all              = {
        + "Name" = "dev-eks-nodegroup-role"
      }
    + unique_id             = (known after apply)

    + inline_policy (known after apply)
  }

# module.eks.aws_iam_role_policy_attachment.AmazonEC2ContainerRegistryReadOnly will be created
+ resource "aws_iam_role_policy_attachment" "AmazonEC2ContainerRegistryReadOnly" {
    + id         = (known after apply)
    + policy_arn = "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"
    + role       = "dev-eks-nodegroup-role"
  }

# module.eks.aws_iam_role_policy_attachment.AmazonEKSWorkerNodePolicy will be created
+ resource "aws_iam_role_policy_attachment" "AmazonEKSWorkerNodePolicy" {
    + id         = (known after apply)
    + policy_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
    + role       = "dev-eks-nodegroup-role"
  }

# module.eks.aws_iam_role_policy_attachment.AmazonEKS_CNI_Policy will be created
+ resource "aws_iam_role_policy_attachment" "AmazonEKS_CNI_Policy" {
    + id         = (known after apply)
    + policy_arn = "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
    + role       = "dev-eks-nodegroup-role"
  }

# module.eks.aws_iam_role_policy_attachment.eks_cluster_AmazonEKSClusterPolicy will be created
+ resource "aws_iam_role_policy_attachment" "eks_cluster_AmazonEKSClusterPolicy" {
    + id         = (known after apply)
    + policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
    + role       = "dev-eks-cluster-role"
  }

# module.eks.aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + region   = "us-east-1"
    + tags     = {
        + "Name" = "dev-igw"
      }
    + tags_all = {
        + "Name" = "dev-igw"
      }
    + vpc_id   = (known after apply)
  }
```

```
# module.eks.aws_route_table.public will be created
+ resource "aws_route_table" "public" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + region           = "us-east-1"
    + route            = [
        + {
            + cidr_block                = "0.0.0.0/0"
            + gateway_id                = (known after apply)
              # (11 unchanged attributes hidden)
          },
      ]
    + tags             = {
        + "Name" = "dev-public-rt"
      }
    + tags_all         = {
        + "Name" = "dev-public-rt"
      }
    + vpc_id           = (known after apply)
  }

# module.eks.aws_route_table_association.public[0] will be created
+ resource "aws_route_table_association" "public" {
    + id             = (known after apply)
    + region         = "us-east-1"
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# module.eks.aws_route_table_association.public[1] will be created
+ resource "aws_route_table_association" "public" {
    + id             = (known after apply)
    + region         = "us-east-1"
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# module.eks.aws_subnet.private[0] will be created
+ resource "aws_subnet" "private" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.3.0/24"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = false
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + region                                         = "us-east-1"
    + tags                                           = {
        + "Name" = "dev-private-1"
        + "Tier" = "private"
      }
    + tags_all                                       = {
        + "Name" = "dev-private-1"
        + "Tier" = "private"
```

```
      }
    + vpc_id                                       = (known after apply)
  }

# module.eks.aws_subnet.private[1] will be created
+ resource "aws_subnet" "private" {
    + arn                                          = (known after apply)
    + assign_ipv6_address_on_creation              = false
    + availability_zone                            = "us-east-1b"
    + availability_zone_id                         = (known after apply)
    + cidr_block                                   = "10.0.4.0/24"
    + enable_dns64                                 = false
    + enable_resource_name_dns_a_record_on_launch  = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                           = (known after apply)
    + ipv6_cidr_block_association_id               = (known after apply)
    + ipv6_native                                  = false
    + map_public_ip_on_launch                      = false
    + owner_id                                     = (known after apply)
    + private_dns_hostname_type_on_launch          = (known after apply)
    + region                                       = "us-east-1"
    + tags                                         = {
        + "Name" = "dev-private-2"
        + "Tier" = "private"
      }
    + tags_all                                     = {
        + "Name" = "dev-private-2"
        + "Tier" = "private"
      }
    + vpc_id                                       = (known after apply)
  }

# module.eks.aws_subnet.public[0] will be created
+ resource "aws_subnet" "public" {
    + arn                                          = (known after apply)
    + assign_ipv6_address_on_creation              = false
    + availability_zone                            = "us-east-1a"
    + availability_zone_id                         = (known after apply)
    + cidr_block                                   = "10.0.1.0/24"
    + enable_dns64                                 = false
    + enable_resource_name_dns_a_record_on_launch  = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                           = (known after apply)
    + ipv6_cidr_block_association_id               = (known after apply)
    + ipv6_native                                  = false
    + map_public_ip_on_launch                      = true
    + owner_id                                     = (known after apply)
    + private_dns_hostname_type_on_launch          = (known after apply)
    + region                                       = "us-east-1"
    + tags                                         = {
        + "Name" = "dev-public-1"
        + "Tier" = "public"
      }
    + tags_all                                     = {
        + "Name" = "dev-public-1"
        + "Tier" = "public"
      }
    + vpc_id                                       = (known after apply)
  }

# module.eks.aws_subnet.public[1] will be created
+ resource "aws_subnet" "public" {
    + arn                                          = (known after apply)
```

```
  }

  # module.eks.aws_subnet.public[1] will be created
  + resource "aws_subnet" "public" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1b"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.2.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "us-east-1"
      + tags                                           = {
          + "Name" = "dev-public-2"
          + "Tier" = "public"
        }
      + tags_all                                       = {
          + "Name" = "dev-public-2"
          + "Tier" = "public"
        }
      + vpc_id                                         = (known after apply)
    }

  # module.eks.aws_vpc.main will be created
  + resource "aws_vpc" "main" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = true
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "us-east-1"
      + tags                                 = {
          + "Name" = "dev-vpc"
        }
      + tags_all                             = {
          + "Name" = "dev-vpc"
        }
    }

Plan: 17 to add, 0 to change, 0 to destroy.
```