Creating an EKS Cluster Using Terraform ⭐

In this guide, we'll use Terraform to create an Amazon Elastic Kubernetes Service (EKS) cluster. The focus is on simplicity and accuracy while leveraging Terraform's modular approach.

What is Terraform?
Terraform is an open-source tool by HashiCorp for defining, provisioning, and managing cloud infrastructure using Infrastructure-as-Code (IaC).

How Terraform Works

1. **Write:** Define your resources in .tf files.
2. **Plan:** Preview changes before applying them.
3. **Apply:** Provision resources as defined in the configuration.
4. **Creating an EKS Cluster Using Terraform** 🌟

   In this guide, we'll use Terraform to create an **Amazon Elastic Kubernetes Service (EKS)** cluster. The focus is on simplicity and accuracy while leveraging Terraform's modular approach.

   ## What is Terraform?

   Terraform is an open-source tool by HashiCorp for defining, provisioning, and managing cloud infrastructure using Infrastructure-as-Code (IaC).

   ### How Terraform Works

   1. **Write**: Define your resources in `.tf` files.
   2. **Plan**: Preview changes before applying them.
   3. **Apply**: Provision resources as defined in the configuration.
   4. **Manage**: Maintain infrastructure with a state file that tracks the resources.

   ## Why Terraform for EKS?

   - **Automation**: Handles complex configurations effortlessly.
   - **Consistency**: Guarantees reproducible environments.
   - **Scalability**: Makes scaling clusters easy.

   ## Prerequisites

   1. **AWS CLI** installed and configured with appropriate permissions.
   2. **kubectl** installed for Kubernetes management.
   3. **Terraform CLI** installed on your machine.
5. Maintain infrastructure with a state file that tracks the resources.

Why Terraform for EKS?

- Automation: Handles complex configurations effortlessly.
- Consistency: Guarantees reproducible environments.
- Scalability: Makes scaling clusters easy.

Prerequisites

- AWS CLI installed and configured with appropriate permissions.
- kubectl installed for Kubernetes management.
- Terraform CLI installed on your machine.

Step-by-Step Guide to Create an EKS Cluster

Step 1: Set Up the Project

1. **Create a new directory for the project**:
2. **Create the required Terraform files**:
   - `main.tf`
   - `variables.tf`
   - `providers.tf`
   - `vpc.tf`
   - `eks.tf`

Step 2: Terraform Configuration

1. **providers.tf**
   Configure the AWS provider:

```
provider "aws" {
 region = "eu-west-1"
 default_tags {
    tags = local.tags
  }
}

terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.67.0"
    }
  }

  required_version = ">= 1.4.2"
}
```

2. main.tf

```
locals {
  tags = {
    created-by = "eks-demo"
    env        = var.cluster_name
  }
}
```

3. vpc.tf

```
data "aws_availability_zones" "available" {
  state = "available"
}

module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "~> 5.1"

  name = var.cluster_name
  cidr = var.vpc_cidr

  azs                   = local.azs
  public_subnets        = local.public_subnets
  private_subnets       = local.private_subnets
  public_subnet_suffix  = "SubnetPublic"
  private_subnet_suffix = "SubnetPrivate"

  enable_nat_gateway   = true
  create_igw           = true
  enable_dns_hostnames = true
  single_nat_gateway   = true

  # Manage so we can name
  manage_default_network_acl    = true
  default_network_acl_tags      = { Name = "${var.cluster_name}-default" }
  manage_default_route_table    = true
  default_route_table_tags      = { Name = "${var.cluster_name}-default" }
  manage_default_security_group = true
  default_security_group_tags   = { Name = "${var.cluster_name}-default" }

  public_subnet_tags = merge(local.tags, {
    "kubernetes.io/role/elb" = "1"
  })
  private_subnet_tags = merge(local.tags, {
    "karpenter.sh/discovery"          = var.cluster_name
    "kubernetes.io/role/internal-elb" = "1"
  })

  tags = local.tags
}
```

4. eks.tf

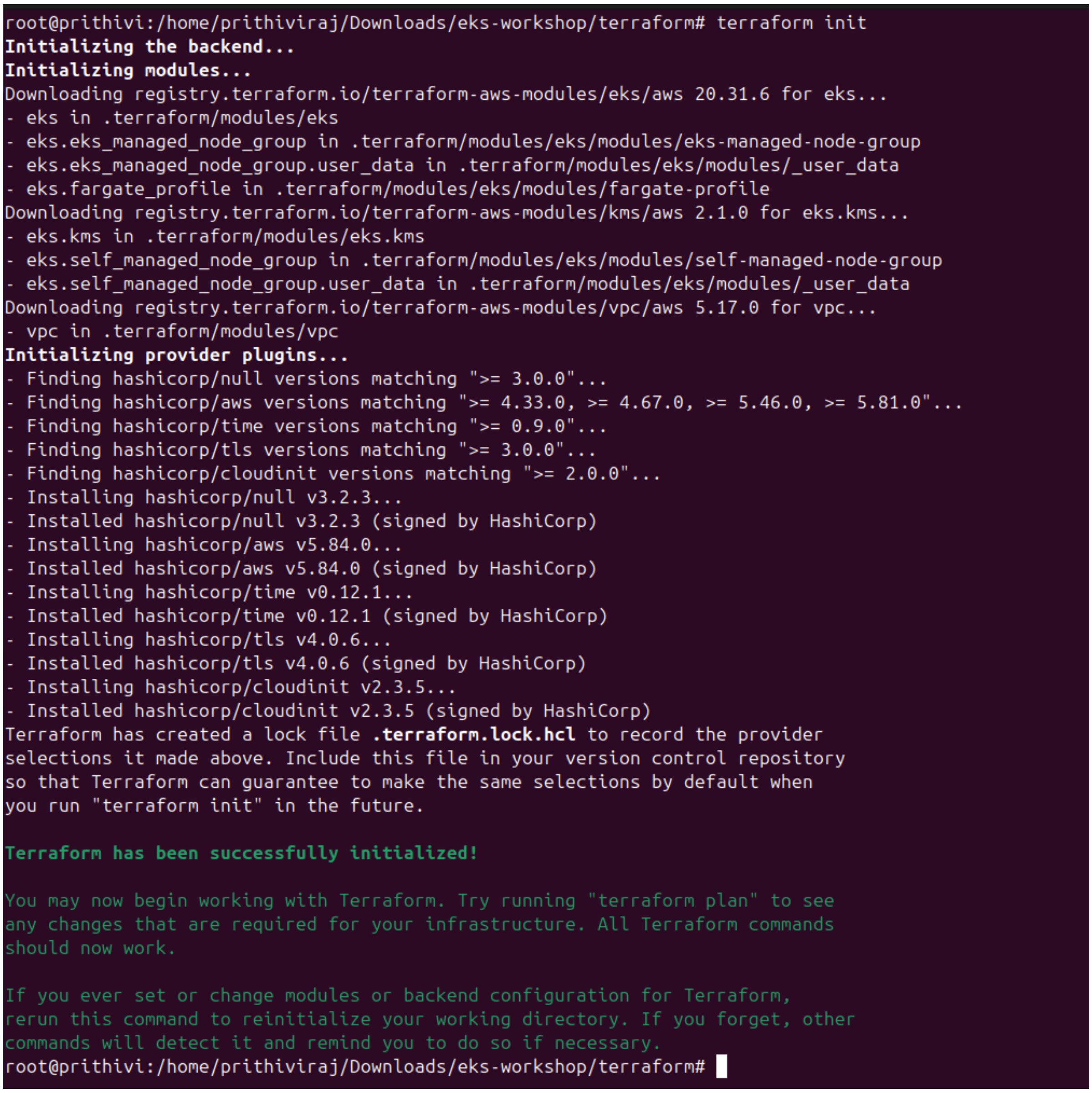
```
module "eks" {
  source  = "terraform-aws-modules/eks/aws"
  version = "~> 20.0"

  cluster_name                             = var.cluster_name
  cluster_version                          = var.cluster_version
  cluster_endpoint_public_access           = true
  enable_cluster_creator_admin_permissions = true

  cluster_addons = {
    vpc-cni = {
      before_compute = true
      most_recent    = true
      configuration_values = jsonencode({
        env = {
          ENABLE_POD_ENI                    = "true"
          ENABLE_PREFIX_DELEGATION          = "true"
          POD_SECURITY_GROUP_ENFORCING_MODE = "standard"
        }
        nodeAgent = {
          enablePolicyEventLogs = "true"
        }
        enableNetworkPolicy = "true"
      })
    }
  }

  vpc_id     = module.vpc.vpc_id
  subnet_ids = module.vpc.private_subnets

  create_cluster_security_group = false
  create_node_security_group    = false

  eks_managed_node_groups = {
    default = {
      instance_types           = ["m5.large"]
      force_update_version     = true
      release_version          = var.ami_release_version
      use_name_prefix          = false
      iam_role_name            = "${var.cluster_name}-ng-default"
      iam_role_use_name_prefix = false

      min_size     = 3
      max_size     = 6
      desired_size = 3

      update_config = {
        max_unavailable_percentage = 50
      }

      labels = {
        workshop-default = "yes"
      }
    }
  }

  tags = merge(local.tags, {
    "karpenter.sh/discovery" = var.cluster_name
  })
}
```

5.variables.tf

```
variable "cluster_name" {
  description = "EKS cluster"
  type        = string
  default     = "eks-demo"
}

variable "cluster_version" {
  description = "EKS cluster version."
  type        = string
  default     = "1.30"
}

variable "ami_release_version" {
  description = "Default EKS AMI release version for node groups"
  type        = string
  default     = "1.30.0-20240625"
}

variable "vpc_cidr" {
  description = "Defines the CIDR block used on Amazon VPC created for Amazon EKS."
  type        = string
  default     = "10.42.0.0/16"
}
```

Step 3: Initialize Terraform
Run the following to initialize Terraform:

```
terraform init
```

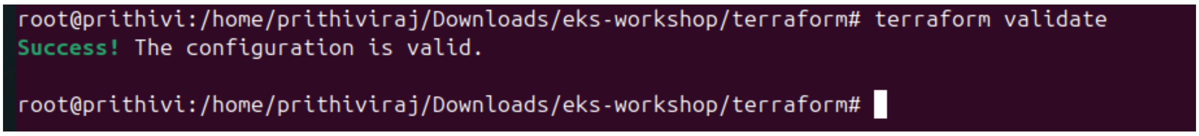


```
root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform# terraform init
Initializing the backend...
Initializing modules...
Downloading registry.terraform.io/terraform-aws-modules/eks/aws 20.31.6 for eks...
- eks in .terraform/modules/eks
- eks.eks_managed_node_group in .terraform/modules/eks/modules/eks-managed-node-group
- eks.eks_managed_node_group.user_data in .terraform/modules/eks/modules/_user_data
- eks.fargate_profile in .terraform/modules/eks/modules/fargate-profile
Downloading registry.terraform.io/terraform-aws-modules/kms/aws 2.1.0 for eks.kms...
- eks.kms in .terraform/modules/eks.kms
- eks.self_managed_node_group in .terraform/modules/eks/modules/self-managed-node-group
- eks.self_managed_node_group.user_data in .terraform/modules/eks/modules/_user_data
Downloading registry.terraform.io/terraform-aws-modules/vpc/aws 5.17.0 for vpc...
- vpc in .terraform/modules/vpc
Initializing provider plugins...
- Finding hashicorp/null versions matching ">= 3.0.0"...
- Finding hashicorp/aws versions matching ">= 4.33.0, >= 4.67.0, >= 5.46.0, >= 5.81.0"...
- Finding hashicorp/time versions matching ">= 0.9.0"...
- Finding hashicorp/tls versions matching ">= 3.0.0"...
- Finding hashicorp/cloudinit versions matching ">= 2.0.0"...
- Installing hashicorp/null v3.2.3...
- Installed hashicorp/null v3.2.3 (signed by HashiCorp)
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (signed by HashiCorp)
- Installing hashicorp/time v0.12.1...
- Installed hashicorp/time v0.12.1 (signed by HashiCorp)
- Installing hashicorp/tls v4.0.6...
- Installed hashicorp/tls v4.0.6 (signed by HashiCorp)
- Installing hashicorp/cloudinit v2.3.5...
- Installed hashicorp/cloudinit v2.3.5 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform#
```

Step 4: Validate,Plan and Apply

1. Validate the deployment

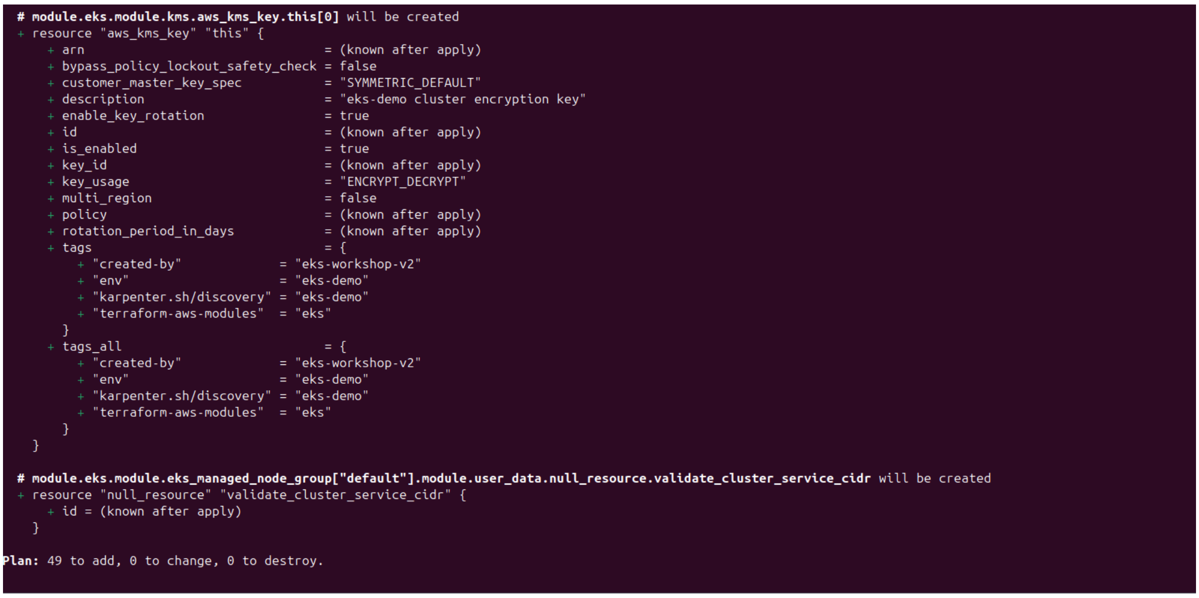
```
terraform validate
```



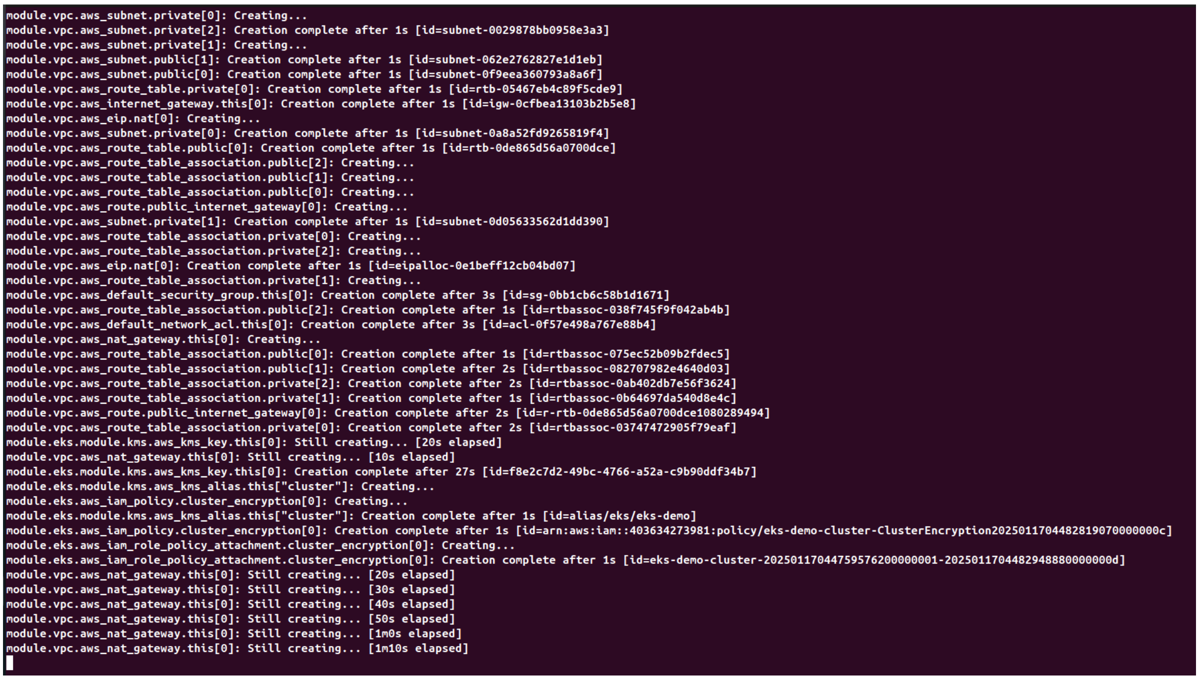
```
root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform# terraform validate
Success! The configuration is valid.

root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform#
```

2.Plan the deployment

```
terraform plan
```



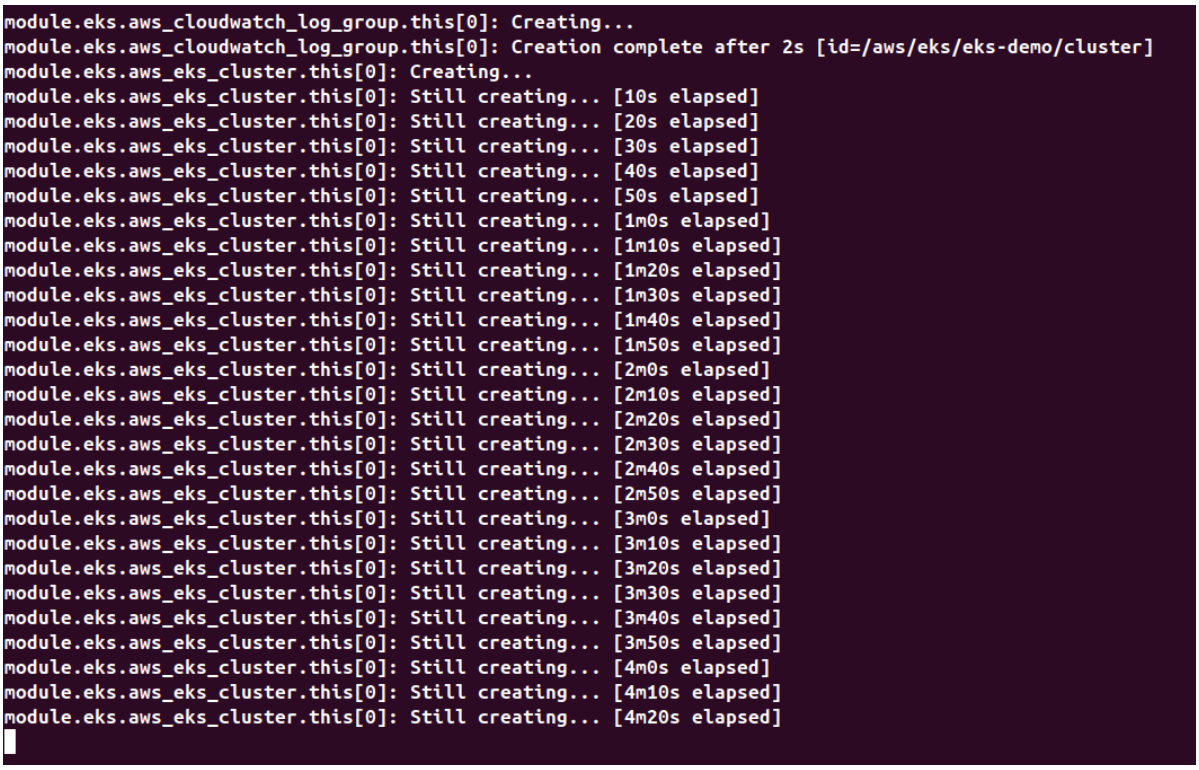
```
  # module.eks.module.kms.aws_kms_key.this[0] will be created
  + resource "aws_kms_key" "this" {
      + arn                                = (known after apply)
      + bypass_policy_lockout_safety_check = false
      + customer_master_key_spec           = "SYMMETRIC_DEFAULT"
      + description                        = "eks-demo cluster encryption key"
      + enable_key_rotation                = true
      + id                                 = (known after apply)
      + is_enabled                         = true
      + key_id                             = (known after apply)
      + key_usage                          = "ENCRYPT_DECRYPT"
      + multi_region                       = false
      + policy                             = (known after apply)
      + rotation_period_in_days            = (known after apply)
      + tags                               = {
          + "created-by"             = "eks-workshop-v2"
          + "env"                    = "eks-demo"
          + "karpenter.sh/discovery" = "eks-demo"
          + "terraform-aws-modules"  = "eks"
        }
      + tags_all                           = {
          + "created-by"             = "eks-workshop-v2"
          + "env"                    = "eks-demo"
          + "karpenter.sh/discovery" = "eks-demo"
          + "terraform-aws-modules"  = "eks"
        }
    }

  # module.eks.module.eks_managed_node_group["default"].module.user_data.null_resource.validate_cluster_service_cidr will be created
  + resource "null_resource" "validate_cluster_service_cidr" {
      + id = (known after apply)
    }

Plan: 49 to add, 0 to change, 0 to destroy.
```



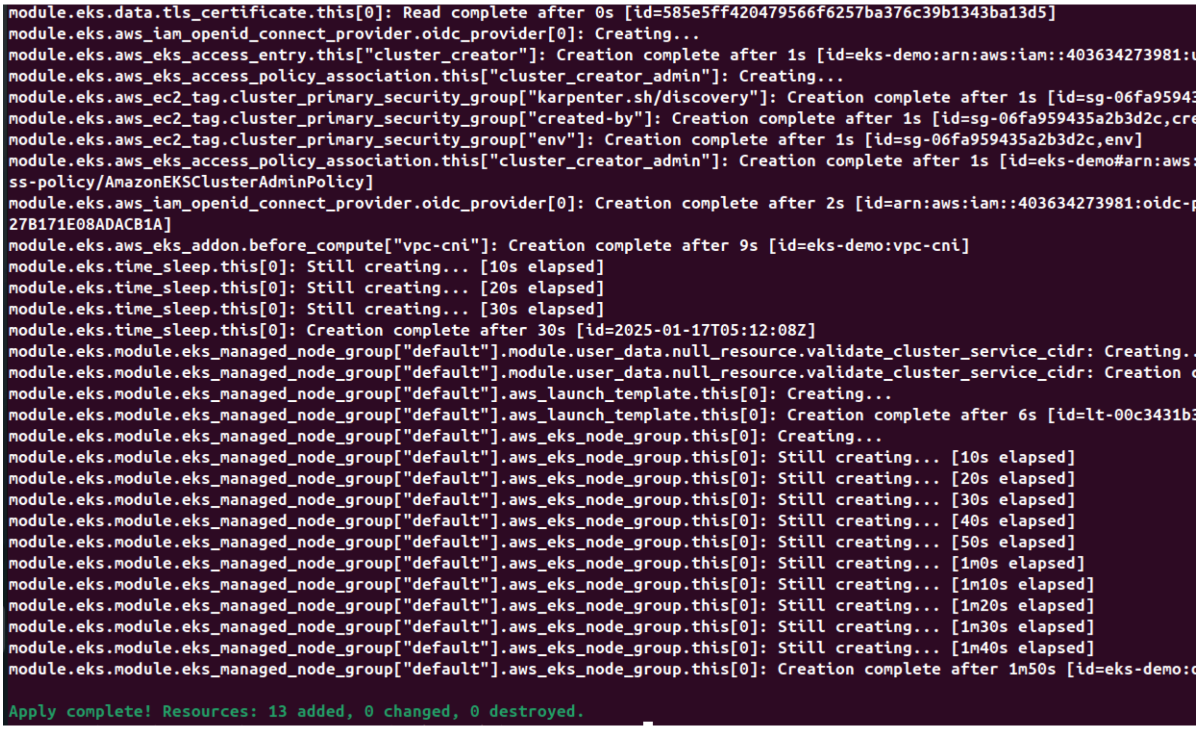
```
module.vpc.aws_subnet.private[0]: Creating...
module.vpc.aws_subnet.private[2]: Creation complete after 1s [id=subnet-0029878bb0958e3a3]
module.vpc.aws_subnet.private[1]: Creating...
module.vpc.aws_subnet.public[1]: Creation complete after 1s [id=subnet-062e2762827e1d1eb]
module.vpc.aws_subnet.public[0]: Creation complete after 1s [id=subnet-0f9eea360793a8a6f]
module.vpc.aws_route_table.private[0]: Creation complete after 1s [id=rtb-05467eb4c89f5cde9]
module.vpc.aws_internet_gateway.this[0]: Creation complete after 1s [id=igw-0cfbea13103b2b5e8]
module.vpc.aws_eip.nat[0]: Creating...
module.vpc.aws_subnet.private[0]: Creation complete after 1s [id=subnet-0a8a52fd9265819f4]
module.vpc.aws_route_table.public[0]: Creation complete after 1s [id=rtb-0de865d56a0700dce]
module.vpc.aws_route_table_association.public[2]: Creating...
module.vpc.aws_route_table_association.public[1]: Creating...
module.vpc.aws_route_table_association.public[0]: Creating...
module.vpc.aws_route.public_internet_gateway[0]: Creating...
module.vpc.aws_subnet.private[1]: Creation complete after 1s [id=subnet-0d05633562d1dd390]
module.vpc.aws_route_table_association.private[0]: Creating...
module.vpc.aws_route_table_association.private[2]: Creating...
module.vpc.aws_eip.nat[0]: Creation complete after 1s [id=eipalloc-0e1beff12cb04bd07]
module.vpc.aws_route_table_association.private[1]: Creating...
module.vpc.aws_default_security_group.this[0]: Creation complete after 3s [id=sg-0bb1cb6c58b1d1671]
module.vpc.aws_route_table_association.public[2]: Creation complete after 1s [id=rtbassoc-038f745f9f042ab4b]
module.vpc.aws_default_network_acl.this[0]: Creation complete after 3s [id=acl-0f57e498a767e88b4]
module.vpc.aws_nat_gateway.this[0]: Creating...
module.vpc.aws_route_table_association.public[0]: Creation complete after 1s [id=rtbassoc-075ec52b09b2fdec5]
module.vpc.aws_route_table_association.public[1]: Creation complete after 2s [id=rtbassoc-082707982e4640d03]
module.vpc.aws_route_table_association.private[2]: Creation complete after 2s [id=rtbassoc-0ab402db7e56f3624]
module.vpc.aws_route_table_association.private[1]: Creation complete after 1s [id=rtbassoc-0b64697da540d8e4c]
module.vpc.aws_route.public_internet_gateway[0]: Creation complete after 2s [id=r-rtb-0de865d56a0700dce1080289494]
module.vpc.aws_route_table_association.private[0]: Creation complete after 2s [id=rtbassoc-03747472905f79eaf]
module.eks.module.kms.aws_kms_key.this[0]: Still creating... [20s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [10s elapsed]
module.eks.module.kms.aws_kms_key.this[0]: Creation complete after 27s [id=f8e2c7d2-49bc-4766-a52a-c9b90ddf34b7]
module.eks.module.kms.aws_kms_alias.this["cluster"]: Creating...
module.eks.aws_iam_policy.cluster_encryption[0]: Creating...
module.eks.module.kms.aws_kms_alias.this["cluster"]: Creation complete after 1s [id=alias/eks/eks-demo]
module.eks.aws_iam_policy.cluster_encryption[0]: Creation complete after 1s [id=arn:aws:iam::403634273981:policy/eks-demo-cluster-ClusterEncryption2025011704482819070000000c]
module.eks.aws_iam_role_policy_attachment.cluster_encryption[0]: Creating...
module.eks.aws_iam_role_policy_attachment.cluster_encryption[0]: Creation complete after 1s [id=eks-demo-cluster-20250117044759576200000001-2025011704482948880000000d]
module.vpc.aws_nat_gateway.this[0]: Still creating... [20s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [30s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [40s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [50s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [1m0s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [1m10s elapsed]
```

3.Apply the deployment

```
terraform apply
```



```
module.eks.aws_cloudwatch_log_group.this[0]: Creating...
module.eks.aws_cloudwatch_log_group.this[0]: Creation complete after 2s [id=/aws/eks/eks-demo/cluster]
module.eks.aws_eks_cluster.this[0]: Creating...
module.eks.aws_eks_cluster.this[0]: Still creating... [10s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [20s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [30s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [40s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [50s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [1m0s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [1m10s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [1m20s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [1m30s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [1m40s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [1m50s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [2m0s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [2m10s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [2m20s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [2m30s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [2m40s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [2m50s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [3m0s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [3m10s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [3m20s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [3m30s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [3m40s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [3m50s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [4m0s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [4m10s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [4m20s elapsed]
```



```
module.eks.data.tls_certificate.this[0]: Read complete after 0s [id=585e5ff420479566f6257ba376c39b1343ba13d5]
module.eks.aws_iam_openid_connect_provider.oidc_provider[0]: Creating...
module.eks.aws_eks_access_entry.this["cluster_creator"]: Creation complete after 1s [id=eks-demo:arn:aws:iam::403634273981:
module.eks.aws_eks_access_policy_association.this["cluster_creator_admin"]: Creating...
module.eks.aws_ec2_tag.cluster_primary_security_group["karpenter.sh/discovery"]: Creation complete after 1s [id=sg-06fa9594
module.eks.aws_ec2_tag.cluster_primary_security_group["created-by"]: Creation complete after 1s [id=sg-06fa959435a2b3d2c,cre
module.eks.aws_ec2_tag.cluster_primary_security_group["env"]: Creation complete after 1s [id=sg-06fa959435a2b3d2c,env]
module.eks.aws_eks_access_policy_association.this["cluster_creator_admin"]: Creation complete after 1s [id=eks-demo#arn:aws:
ss-policy/AmazonEKSClusterAdminPolicy]
module.eks.aws_iam_openid_connect_provider.oidc_provider[0]: Creation complete after 2s [id=arn:aws:iam::403634273981:oidc-
27B171E08ADACB1A]
module.eks.aws_eks_addon.before_compute["vpc-cni"]: Creation complete after 9s [id=eks-demo:vpc-cni]
module.eks.time_sleep.this[0]: Still creating... [10s elapsed]
module.eks.time_sleep.this[0]: Still creating... [20s elapsed]
module.eks.time_sleep.this[0]: Still creating... [30s elapsed]
module.eks.time_sleep.this[0]: Creation complete after 30s [id=2025-01-17T05:12:08Z]
module.eks.module.eks_managed_node_group["default"].module.user_data.null_resource.validate_cluster_service_cidr: Creating..
module.eks.module.eks_managed_node_group["default"].module.user_data.null_resource.validate_cluster_service_cidr: Creation
module.eks.module.eks_managed_node_group["default"].aws_launch_template.this[0]: Creating...
module.eks.module.eks_managed_node_group["default"].aws_launch_template.this[0]: Creation complete after 6s [id=lt-00c3431b
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Creating...
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [10s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [20s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [30s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [40s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [50s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m0s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m10s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m20s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m30s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m40s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Creation complete after 1m50s [id=eks-demo:

Apply complete! Resources: 13 added, 0 changed, 0 destroyed.
```

Step 5: Configure `kubectl`

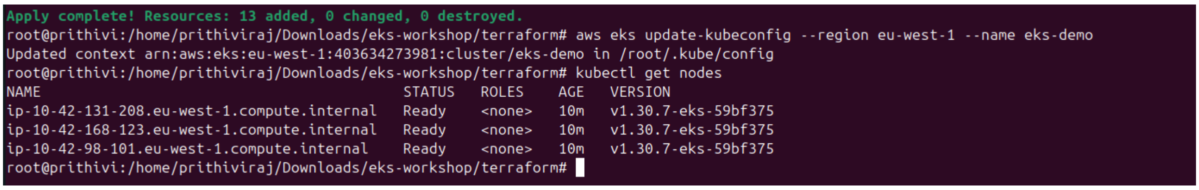
Update the Kubernetes configuration to connect to the new cluster:

```
aws eks --region eu-west-1 update-kubeconfig --name eks-demo
```

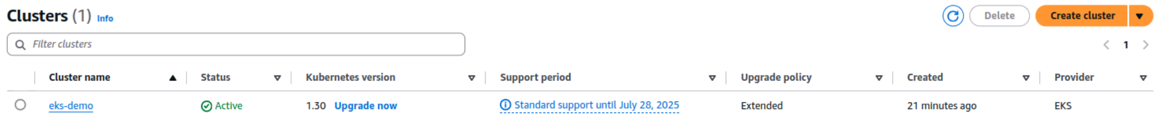
Step 6: Verify Your EKS Cluster

List the nodes:

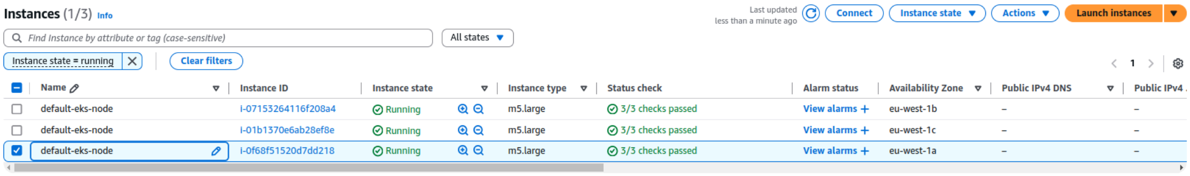
```
kubectl get nodes
```



```
Apply complete! Resources: 13 added, 0 changed, 0 destroyed.
root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform# aws eks update-kubeconfig --region eu-west-1 --name eks-demo
Updated context arn:aws:eks:eu-west-1:403634273981:cluster/eks-demo in /root/.kube/config
root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform# kubectl get nodes
NAME                                         STATUS   ROLES    AGE   VERSION
ip-10-42-131-208.eu-west-1.compute.internal   Ready    <none>   10m   v1.30.7-eks-59bf375
ip-10-42-168-123.eu-west-1.compute.internal   Ready    <none>   10m   v1.30.7-eks-59bf375
ip-10-42-98-101.eu-west-1.compute.internal    Ready    <none>   10m   v1.30.7-eks-59bf375
root@prithivi:/home/prithiviraj/Downloads/eks-workshop/terraform# 
```



	Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
○	eks-demo	Active	1.30 Upgrade now	Standard support until July 28, 2025	Extended	21 minutes ago	EKS



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 …
□	default-eks-node	i-07153264116f208a4	Running	m5.large	3/3 checks passed	View alarms +	eu-west-1b	–	–
□	default-eks-node	i-01b1370e6ab28ef8e	Running	m5.large	3/3 checks passed	View alarms +	eu-west-1c	–	–
☑	default-eks-node	i-0f68f51520d7dd218	Running	m5.large	3/3 checks passed	View alarms +	eu-west-1a	–	–

Benefits of Using Terraform Modules

- **Reusable Components**: Use pre-built modules for consistency.
- **Simplified Configurations**: Modules handle complex resource dependencies.
- **Scalable Architecture**: Makes scaling infrastructure seamless.

Conclusion

Deploying an EKS cluster with Terraform streamlines the process of setting up and managing Kubernetes workloads on AWS. By leveraging Terraform modules, you ensure a reliable, scalable, and efficient setup.

Happy Learning

Prithiviraj Rengarajan

DevOps Engineer