Using Terraform to create the IAM Roles and Policies for EKS

Create the Required EKS Roles

1

```
cd ~/environment/tfekscode/iam
```

Initialize Terraform

1

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

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

Validate the Terraform code

1

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

Plan the deployment:

1

```
terraform plan -out tfplan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
```

```
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
```

Terraform will perform the following actions:

```
  # aws_iam_role.eks-cluster-ServiceRole will be created
  + resource "aws_iam_role" "eks-cluster-ServiceRole" {
      + arn                   = (known after apply)
      + assume_role_policy    = jsonencode(
          {
            + Statement = [
              + {
                  + Action    = "sts:AssumeRole"
                  + Effect    = "Allow"
                  + Principal = {
                      + Service = [
                          + "eks-fargate-pods.amazonaws.com",
                          + "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                  = (sensitive value)
      + name_prefix           = (known after apply)
      + path                  = "/"
      + tags                  = {
          + "Name" = (sensitive value)
      }
      + tags_all              = {
          + "Name" = "f23f39c2e634f9ae-eks-cluster/ServiceRole"
```

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

** OUTPUT TRUNCATED FOR BREVITY **

 # aws_ssm_parameter.nodegroup_role_arn will be created
  + resource "aws_ssm_parameter" "nodegroup_role_arn" {
    + arn           = (known after apply)
    + data_type     = (known after apply)
    + description   = "node grpup role arn"
    + id            = (known after apply)
    + insecure_value = (known after apply)
    + key_id        = (known after apply)
    + name          = "/workshop/tf-eks/nodegroup_role_arn"
    + tags          = {
      + "workshop" = "tf-eks-workshop"
    }
    + tags_all      = {
      + "workshop" = "tf-eks-workshop"
    }
    + tier          = (known after apply)
    + type          = "String"
    + value         = (sensitive value)
    + version       = (known after apply)
  }

Plan: 23 to add, 0 to change, 0 to destroy.
```

You can see from the plan the following resources will be created - open the corresponding files to see the Terraform HCL code that details the configuration

- A Cluster Service Role (**aws_iam_role__cluster-ServiceRole.tf**)
- A Node Group Service Role (**aws_iam_role__nodegroup-NodeInstanceRole.tf**)

- Various policy definitions that EKS needs eg: (**aws_iam_role_policy__nodegroup-NodeInstanceRole-PolicyAutoScaling.tf**)
- Policy attachments to the cluster and node group roles eg: (**aws_iam_role_policy_attachment__cluster-ServiceRole-AmazonEKSClusterPolicy.tf**)

Build the Roles, Policies etc.:

1

terraform apply tfplan

aws_iam_role.eks-nodegroup-ng-ma-NodeInstanceRole: Creating...

aws_iam_role.eks-cluster-ServiceRole: Creating...

aws_key_pair.eksworkshop: Creating...

aws_key_pair.eksworkshop: Creation complete after 0s [id=f23f39c2e634f9ae-eksworkshop]

aws_ssm_parameter.key_name: Creating...

aws_ssm_parameter.key_name: Creation complete after 0s [id=/workshop/tf-eks/key_name]

** OUTPUT TRUNCATED FOR BREVITY **

aws_iam_role_policy_attachment.eks-cluster-ServiceRole__AmazonEKSVPCResourceController: Creation complete after 0s [id=f23f39c2e634f9ae-eks-cluster-ServiceRole-20230416133548140700000006]

aws_iam_role_policy.eks-cluster-ServiceRole__eks-cluster-PolicyCloudWatchMetrics: Creation complete after 0s [id=f23f39c2e634f9ae-eks-cluster-ServiceRole:f23f39c2e634f9ae-eks-cluster-PolicyCloudWatchMetrics]

aws_iam_role_policy.eks-cluster-ServiceRole-HUIGIC7K7HNJ__eks-cluster-PolicyELBPermissions: Creation complete after 0s [id=f23f39c2e634f9ae-eks-cluster-ServiceRole:f23f39c2e634f9ae-eks-cluster-PolicyELBPermissions]

aws_iam_role_policy_attachment.eks-cluster-ServiceRole__AmazonEKSClusterPolicy: Creation complete after 0s [id=f23f39c2e634f9ae-eks-cluster-ServiceRole-20230416133548170400000007]

Apply complete! Resources: 23 added, 0 changed, 0 destroyed.

The above creates the necessary Roles with attached Policies needed by EKS Examine the results in AWS console (IAM section)