```yaml
# Cloudformation Infrastructure as Code to create EKS cluster
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  WorkerInstanceType:
    Type: String
    Default: t2.micro
    Description: "Worker node instance type"
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: "ID of the VPC to deploy the EKS cluster into"
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
    Description: "List of subnet IDs in different AZs to deploy the EKS cluster into"

Resources:
  # creates eks cluster
  MyCluster:
    Type: AWS::EKS::Cluster
    Properties:
      Name: !Sub ${AWS::StackName}-cluster
      Version: "1.26"
      RoleArn: !GetAtt MyClusterRole.Arn
      ResourcesVpcConfig:
        SecurityGroupIds:
          - !GetAtt MyClusterSecurityGroup.GroupId
        SubnetIds: !Ref SubnetIds
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-cluster

  # creates security group for eks cluster
  MyClusterSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: "My Kubernetes cluster security group"
      VpcId: !Ref VpcId
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 0
          ToPort: 65535
          CidrIp: "0.0.0.0/0"
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 0
          ToPort: 65535
          CidrIp: "0.0.0.0/0"
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-cluster-sg

  # creates role for eks cluster
  MyClusterRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
```

```yaml
            Principal:
              Service: eks.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
        - arn:aws:iam::aws:policy/AmazonEKSServicePolicy
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-cluster-role

  # creates node group for eks cluster
  MyNodeGroup:
    Type: AWS::EKS::Nodegroup
    Properties:
      ClusterName: !Ref MyCluster
      NodeRole: !GetAtt MyNodeRole.Arn
      Subnets: !Ref SubnetIds
      InstanceTypes:
        - !Ref WorkerInstanceType
      ScalingConfig:
        DesiredSize: 1
        MaxSize: 2
        MinSize: 1
      AmiType: AL2_x86_64

  # creates role for node group
  MyNodeRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service: ec2.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-node-role

  # create ecr repository
  MyECR:
    Type: AWS::ECR::Repository
    Properties:
      RepositoryName: !Sub ${AWS::StackName}-ecr
      RepositoryPolicyText:
        Version: "2008-10-17"
        Statement:
          - Sid: AllowPushPull
            Effect: Allow
            Principal:
              AWS: !GetAtt MyNodeRole.Arn
            Action:
              - ecr:GetDownloadUrlForLayer
              - ecr:BatchGetImage
              - ecr:BatchCheckLayerAvailability
```

```
              - ecr:PutImage
              - ecr:InitiateLayerUpload
              - ecr:UploadLayerPart
              - ecr:CompleteLayerUpload

  # roles to let admin user see the pod, service details etc in console

Outputs:
  ECRUrl:
    Description: "ECR URL"
    Value: !GetAtt MyECR.RepositoryUri

```