**Prerequisites:**

You should be ready with EKS cluster and node

**Steps:**

1. **Create IAM Policy for EBS:**

Policy details: Amazon\_EBS\_CSI\_Driver

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:AttachVolume",

"ec2:CreateSnapshot",

"ec2:CreateTags",

"ec2:CreateVolume",

"ec2:DeleteSnapshot",

"ec2:DeleteTags",

"ec2:DeleteVolume",

"ec2:DescribeInstances",

"ec2:DescribeSnapshots",

"ec2:DescribeTags",

"ec2:DescribeVolumes",

"ec2:DetachVolume"

],

"Resource": "\*"

}

]

}

1. **Get the IAM role Worker Nodes using and Associate this policy to that role**you can get the role using below command  
     
   kubectl -n kube-system describe configmap aws-auth  
     
   o/p be like : rolearn: arn:aws:iam::180789647333:**role/eksctl-eksdemo1-nodegroup-eksdemo-NodeInstanceRole-IJN07ZKXAWNN**you can attached the above policy to highlighted role
2. **Deploy Amazon EBS CSI Driver**

You can install the EBS CSI Driver using below command

kustomize build "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.41" | kubectl apply -f –

Verify ebs-csi pods running

kubectl get pods -n kube-system

1. **Below are the manifest file for Storage class, Persistent Volume, Postgress db and Sonarqube**storage-class.yml  
     
   apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

  name: ebs-sc

provisioner: ebs.csi.aws.com  
volumeBindingMode: WaitForFirstConsumer  
  
persistent-volume-claim.yml  
  
apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: postgres-pv-claim

  namespace: sonarqube

spec:

  accessModes:

    - ReadWriteOnce

  storageClassName: ebs-sc

  resources:

    requests:

      storage: 4Gi

postgress.yaml

apiVersion: v1

kind: ConfigMap

metadata:

  name: postgres-config

  namespace: sonarqube

data:

  POSTGRES\_DB: sonarqube

  POSTGRES\_USER: sonar

  POSTGRES\_PASSWORD: sonarpassword

---

apiVersion: apps/v1

kind: Deployment

metadata:

  name: postgres

  namespace: sonarqube

spec:

  replicas: 1

  selector:

    matchLabels:

      app: postgres

  template:

    metadata:

      labels:

        app: postgres

    spec:

      containers:

        - name: postgres

          image: postgres:15

          envFrom:

            - configMapRef:

                name: postgres-config

          ports:

            - containerPort: 5432

          volumeMounts:

            - name: postgres-storage

              mountPath: /var/lib/postgresql/data

              subPath: postgres

      volumes:

        - name: postgres-storage

          persistentVolumeClaim:

            claimName: postgres-pv-claim

---

apiVersion: v1

kind: Service

metadata:

  name: postgres

  namespace: sonarqube

spec:

  ports:

    - port: 5432

      targetPort: 5432

  selector:

    app: postgres

  type: ClusterIP

sonar.yaml  
  
  
apiVersion: apps/v1

kind: Deployment

metadata:

  name: sonarqube

  namespace: sonarqube

spec:

  replicas: 1

  selector:

    matchLabels:

      app: sonarqube

  template:

    metadata:

      labels:

        app: sonarqube

    spec:

      containers:

        - name: sonarqube

          image: sonarqube:community

          env:

            - name: SONAR\_JDBC\_URL

              value: "jdbc:postgresql://postgres:5432/sonarqube"

            - name: SONAR\_JDBC\_USERNAME

              value: "sonar"

            - name: SONAR\_JDBC\_PASSWORD

              value: "sonarpassword"

          ports:

            - containerPort: 9000

---

apiVersion: v1

kind: Service

metadata:

  name: sonarqube

  namespace: sonarqube

spec:

  type: NodePort

  selector:

    app: sonarqube

  ports:

    - protocol: TCP

      port: 9000

      targetPort: 9000

      nodePort: 32085