1. INSTALL EKSCTL ON LINUX.

```
root@DESKTOP-VIDGOBF:KUBERNETES# eksctl create cluster -f eks-cube-cluster.yaml
2024-0-27 12:33:02 | eksctl version 0.171.0
2024-0-27 12:33:03 | eksctl passibility zones to [ap-south-la ap-south-lb]
2024-0-27 12:33:03 | eksting availability zones to [ap-south-la ap-south-lb]
2024-0-27 12:33:03 | eksting availability zones to [ap-south-la ap-south-la ap-sou
```

```
NAME
                                                  STATUS
                                                           ROLES
                                                                     AGE
                                                                             VERSION
ip-192-168-14-80.ap-south-1.compute.internal
                                                                     7m44s
                                                                             v1.27.9-eks-5e0fdde
                                                  Ready
                                                           <none>
ip-192-168-38-174.ap-south-1.compute.internal
                                                  Ready
                                                           <none>
                                                                     7m59s
                                                                             v1.27.9-eks-5e0fdde
                                                                             v1.27.9-eks-5e0fdde
ip-192-168-4-61.ap-south-1.compute.internal
                                                  Ready
                                                           <none>
                                                                     8m6s
ip-192-168-61-72.ap-south-1.compute.internal
                                                                     8m9s
                                                  Ready
                                                           <none>
                                                                             v1.27.9-eks-5e0fdde
```

EFS Plugin Install REF Link

----> https://github.com/haneefshaikh/cubejs on eks/tree/main

TO CREATE AN IAM OIDC IDENTITY PROVIDER FOR YOUR CLUSTER WITH EKSCTL

Determine whether you have an existing IAM OIDC provider for your cluster. Retrieve your cluster's OIDC provider ID and store it in a variable.

oidc_id=\$(aws eks describe-cluster --name eks-cube-test --query "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5)

```
root@DESKTOP-VIDGD8F:task4# oidc_id=$(aws eks describe-cluster --name eks-cube-dev --query "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5) root@DESKTOP-VIDGD8F:task4#
```

Determine whether an IAM OIDC provider with your cluster's ID is already in your account.

aws iam list-open-id-connect-providers | grep \$oidc id | cut -d "/" -f4

```
root@DESKTOP-VIDGD8F:task4#
root@DESKTOP-VIDGD8F:task4# aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
root@DESKTOP-VIDGD8F:task4#
```

Create an IAM OIDC identity provider for your cluster with the following command. Replace eks-cube-dev with your own value.

eksctl utils associate-iam-oidc-provider --cluster eks-cube-test --approve

```
root@DESKTOP-VIDGD8F:task4# eksctl utils associate-iam-oidc-provider --cluster eks-cube-dev --approve
2024-02-27 14:34:44 [m] will create IAM Open ID Connect provider for cluster "eks-cube-dev" in "ap-south-1"
2024-02-27 14:34:45 [.] created IAM Open ID Connect provider for cluster "eks-cube-dev" in "ap-south-1"
root@DESKTOP-VIDGD8F:task4#
```

Check again whether an IAM OIDC provider with your cluster's ID is created in your account.

aws iam list-open-id-connect-providers | grep \$oidc id | cut -d "/" -f4

```
root@DESKTOP-VIDGD8F:task4# aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4 563845675DCFF584D760A0B0D815B504" root@DESKTOP-VIDGD8F:task4#
```

AMAZON EFS CSI DYNAMIC PROVISIONING

Download the IAM policy document

curl -S https://raw.githubusercontent.com/kubernetes-sigs/aws-efs-csi-driver/v1.2.0/docs/iam-policy-example.json -o iam-policy.json

Create an IAM policy

aws iam create-policy \

- --policy-name EFSCSIControllerIAMPolicy \
- --policy-document file://iam-policy.json

```
root@DESKTOP-VIDGD8F:KUBERNETES# aws iam create-policy --policy-name EFSCSIControllerIAMPolicy --policy-document file://iam-policy.json
{
    "Policy": {
         "PolicyName": "EFSCSIControllerIAMPolicy",
         "PolicyId": "ANPATXEPQB7UJYVPEZTSN",
         "Arn": "anriaws:iam::255851499496:policy/EFSCSIControllerIAMPolicy",
         "Path": '/",
         "DefaultVersionId": "v1",
         "AtachmentCount": 0,
         "PermissionsBoundaryUsageCount": 0,
         "IsAttachable": true,
         "CreateDate": "2024-02-27T08:13:42Z",
         "UpdateDate": "2024-02-27T08:13:42Z"
}
}
root@DESKTOP-VIDGD8F:KUBERNETES#
```

Create a Kubernetes service account

eksctl create iamserviceaccount \

- --cluster=eks-cube-test \setminus
- --region=us-east-2 \
- --namespace=kube-system \
- --name=efs-csi-controller-sa \
- --override-existing-serviceaccounts \
- --attach-policy-arn=arn:aws:iam::<AWS account ID>:policy/EFSCSIControllerIAMPolicy \
- --approve

```
root@DESKTOP-VIDGO8F:KUBERNETES# eksctl create iamserviceaccount --cluster=eks-cube-dev --region=ap-south-1 --namespace=kube-system --name=efs-csi-controller-sa --override-existing-serviceaccounts --attach-policy-arn=arn:aws:iam::255851499496:policy/EFSCSIControllerIAMPolicy --approve 2024-02-27 21:36:9 [s] 1 iamserviceaccount (kube-system/efs-csi-controller-sa) was included (based on the include/exclude rules) 2024-02-27 21:36:99 [s] 1 task: {
2024-02-27 21:36:99 [s] 1 task: {
2 sequential sub-tasks: {
    create 1AM role for serviceaccount "kube-system/efs-csi-controller-sa",
    create 1AM role for serviceaccount "kube-system/efs-csi-controller-sa",
    create serviceaccount "kube-system/efs-csi-controller-sa",
    2024-02-27 21:36:99 [s] building iamserviceaccount stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa" 2024-02-27 21:36:99 [s] deploying stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa" 2024-02-27 21:36:90 [s] waiting for CloudFormation stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa" 2024-02-27 21:36:90 [s] created serviceaccount "kube-system/efs-csi-controller-sa" 2024-02-27 21:36:90 [s] created serviceaccou
```

To verify that the new service role is created, run one of the following commands:

eksctl get iamserviceaccount --cluster eks-cube-test --name efs-csi-controller-sa --namespace kubesystem

```
efs-csi-controller-sa
                       arn:aws:iam::255851499496:role/eksctl-eks-cube-dev-addon-iamserviceaccount-k-Role1-tcyY1eyt4vul
```

Now install AWS EFS Storage Controller driver.

helm repo add aws-efs-csi-driver https://kubernetes-sigs.github.io/aws-efs-csi-driver

```
onfiguration file is world-readable. This is insecure. Location: /mnt/c/Users/ADMIN/Desktop/My_Notes/KUBERNETES/k8sCluster_kubeadm_terraform/suit
```

helm repo update

```
root@DESKTOP-VIDGD8F:KUBERNETES# helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "aws-efs-csi-driver" chart repository
Update Complete. ∗Happy Helming!∗
```

helm upgrade -i aws-efs-csi-driver aws-efs-csi-driver/aws-efs-csi-driver \

- --namespace kube-system \
- --set image.repository=602401143452.dkr.ecr.us-west-2.amazonaws.com/eks/aws-efs-csi-driver \
- --set controller.serviceAccount.create=false \
- --set controller.serviceAccount.name=efs-csi-controller-sa

```
TATUS: deployed
EVISION: 1
EST SUITE: None
 kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-efs-csi-driver,app.kubernetes.io/instance=aws-efs-csi-driver"
t@DESKTOP-VIDGD8F:-##
```

To verify that aws-efs-csi-driver has started, run:

kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-efs-csidriver,app.kubernetes.io/instance=aws-efs-csi-driver"

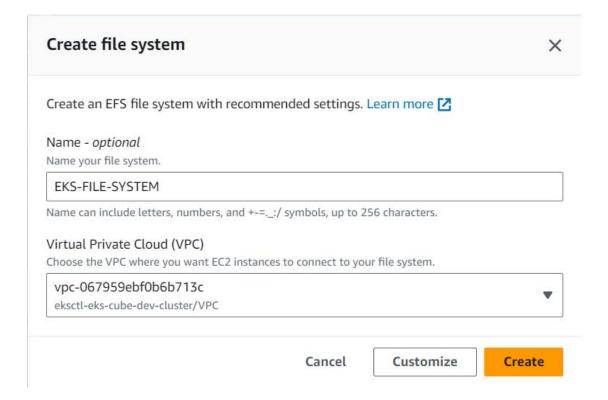
```
-csi-controller-598bf64f56-dgrmg
-csi-node-dfbrr
-csi-node-fwt9l
```

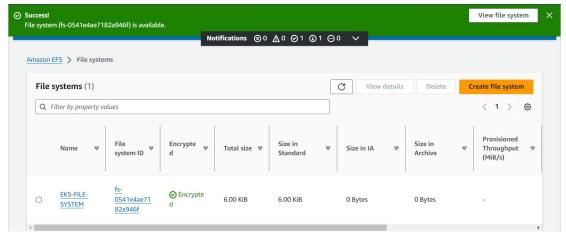
To create iamserviceaccount.

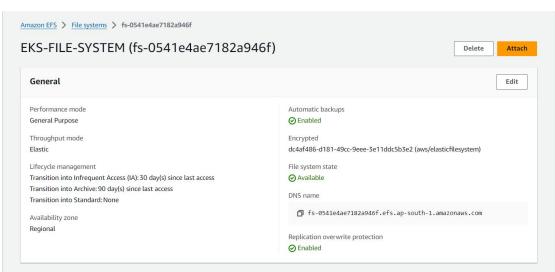
eksctl create iamserviceaccount \

- --cluster=eks-cube-test \
- --namespace=kube-system \setminus
- --name=aws-load-balancer-controller \setminus
- --override-existing-service accounts $\$
- --approve

root@DESKTOP-VIDGD8F:KUBERNETES# eksctl create iamserviceaccount --cluster=eks-cube-dev --namespace=kube-system --name=aws-load-balancer-control ler --attach-policy-arn=arn:aws:iam::258851499496:policy/AWSLoadBalancerControllerIAMPolicy --override-existing-serviceaccounts --approve 2024-02-27 18:59:40 [*] 2 existing iamserviceaccount(s) (kube-system/aws-load-balancer-controller, kube-system/efs-csi-controller-sa) will be excluded 2024-02-27 18:59:40 [*] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) was excluded (based on the include/exclude rules) 2024-02-27 18:59:40 [*] motadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set 2024-02-27 18:59:40 [*] no tasks coot@DESKTOP-VIDGD8F EXBERNETES#







```
root@DESKTOP-VIDGD8F:task4# vim pv.yaml
root@DESKTOP-VIDGD8F:task4# cat pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: efs-pv
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  storageClassName:
  persistentVolumeReclaimPolicy: Retain
  csi:
    driver: efs.csi.aws.com
    volumeHandle: fs-0541e4ae7182a946f
root@DESKTOP-VIDGD8F:task4#
```

root@DESKTOP-VIDGD8F:task4# kubectl apply -f .
persistentvolume/efs-pv created
root@DESKTOP-VIDGD8F:task4#

```
root@DESKTOP-VIDGD8F:task4# kubectl get pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE
efs-pv 5Gi RWO Retain Available 70s
root@DESKTOP-VIDGD8F:task4#
```

```
root@DESKTOP-VIDGD8F:task4# vim pvc.yaml
root@DESKTOP-VIDGD8F:task4# cat pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
   name: efs-claim
spec:
   accessModes:
    - ReadWriteOnce
   storageClassName: ""
   resources:
     requests:
     storage: 5Gi
root@DESKTOP-VIDGD8F:task4#
```

root@DESKTOP-VIDGD8F:task4# kubectl apply -f pvc.yaml
persistentvolumeclaim/efs-claim created
root@DESKTOP-VIDGD8F:task4#

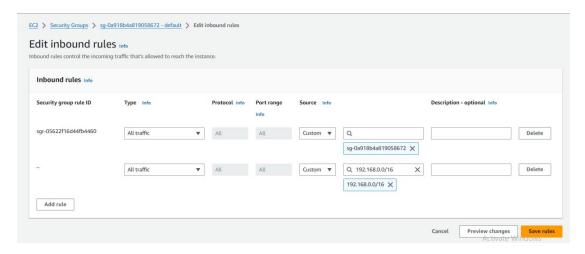
```
root@DESKTOP-VIDGD8F:task4# kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
efs-claim Bound efs-pv 5Gi RWO 6s
root@DESKTOP-VIDGD8F:task4#
```

root@DESKTOP-VIDGD8F:task4# kubectl get pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE
efs-pv 5Gi RWO Retain Bound default/efs-claim 5m17:
root@DESKTOP-VIDGD8F:task4#

root@DESKTOP-VIDGD8F:task4# cat pod.yaml apiVersion: v1 kind: Pod metadata: name: efs-app spec: containers: - name: app image: centos command: ["/bin/sh"] args: ["-c", "while true; do echo \$(date -u) >> /data/out.txt; sleep 2; done"] - name: persistent-storage mountPath: /data volumes: - name: persistent-storage persistentVolumeClaim: claimName: efs-claim root@DESKTOP-VIDGD8F:task4#

root@DESKTOP-VIDGD8F:task4# kubectl apply -f pod.yaml pod/efs-app created root@DESKTOP-VIDGD8F:task4#



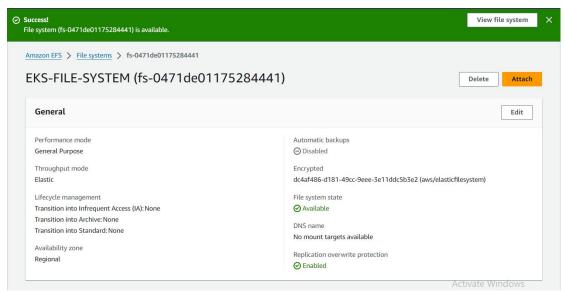
root@DESKTOP-VIDGD8F:task4# kubectl apply pod/efs-app created persistentvolume/efs-pv created persistentvolumeclaim/efs-claim created

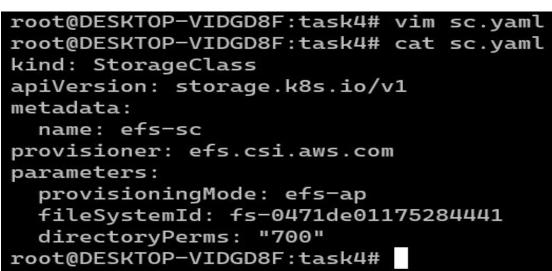
root@DESKTOP-VIDGD8F:task4# kgp NAME READY STATUS RESTARTS AGE Running efs-app 1/1 61s

root@DESKTOP-VIDGD8F:task4# k get pv NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE default/efs-claim Retain Bound root@DESKTOP-VIDGD8F:task4# k get pvc STATUS VOLUME CAPACITY ACCESS MODES **STORAGECLASS** AGE RWO efs-claim Bound 5Gi 69s

efs-pv

Next, we need to create a Storage Class and provide the FileSystemId of the newly created file system:





```
root@DESKTOP-VIDGD8F:task4# vim pvc.yaml
root@DESKTOP-VIDGD8F:task4# cat pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
   name: efs-claim
spec:
   accessModes:
    - ReadWriteMany
   storageClassName: efs-sc
   resources:
     requests:
     storage: 5Gi
root@DESKTOP-VIDGD8F:task4#
```

```
root@DESKTOP-VIDGD8F:task4# cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
 name: efs-app
spec:
 containers:
  - name: app
    image: centos
   command: ["/bin/sh"]
   args: ["-c", "while true; do echo $(date -u) >> /data/out.txt; sleep 2; done"]
   volumeMounts:
    - name: persistent-storage
     mountPath: /data
 volumes:
  - name: persistent-storage
    persistentVolumeClaim:
     claimName: efs-claim
root@DESKTOP-VIDGD8F:task4#
```

```
root@DESKTOP-VIDGD8F:task4# kubectl apply -f sc.yaml
storageclass.storage.k8s.io/efs-sc created
root@DESKTOP-VIDGD8F:task4# kubectl apply -f pvc.yaml
persistentvolumeclaim/efs-claim created
root@DESKTOP-VIDGD8F:task4# kubectl apply -f pod.yaml
pod/efs-app created
root@DESKTOP-VIDGD8F:task4#
```

```
root@DESKTOP-VIDGO8F:task4# k get sc,pvc,po
NAME
storageclass.storage.k8s.io/efs-sc efs.csi.aws.com Delete
storageclass.storage.k8s.io/gp2 (default)

NAME
NAME
Persistentvolumeclaim/efs-claim
RECLAIMPOLICY
NOLUME
PROVISIONER
PROVISIONER
Sets.csi.aws.com Delete
RECLAIMPOLICY
Immediate
WaitForFirstConsumer
False
NAME
Pocapacity
ROOK
Pocapacity
Pocapacity
ROOK
RECLAIMPOLICY
NAME
RECLAIMPOLICY
NAME
Pocapacity
Rook
RECLAIMPOLICY
ROUNG
ROUNG
ROUNG
ROUNG
RECLAIMPOLICY
ROUNG
ROUN
```

```
root@DESKTOP-VIDGD8F:KUBERNETES# kubectl exec -ti efs-app -- tail -f /data/out.txt
Wed Feb 28 08:01:25 UTC 2024
Wed Feb 28 08:01:27 UTC 2024
Wed Feb 28 08:01:31 UTC 2024
Wed Feb 28 08:01:31 UTC 2024
Wed Feb 28 08:01:33 UTC 2024
Wed Feb 28 08:01:35 UTC 2024
Wed Feb 28 08:01:37 UTC 2024
Wed Feb 28 08:01:37 UTC 2024
Wed Feb 28 08:01:39 UTC 2024
Wed Feb 28 08:01:41 UTC 2024
Wed Feb 28 08:01:41 UTC 2024
Wed Feb 28 08:01:43 UTC 2024
Wed Feb 28 08:01:45 UTC 2024
```