- **EKS -> PV,PVC,POD**

root@DESKTOP-8OOG2HF:~# eksctl create cluster -f eks-cube-cluster.yaml

```
2024-02-28 11:21:06 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-02"
2024-02-28 11:21:26 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-01"
2024-02-28 11:22:12 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-02"
2024-02-28 11:22:25 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-01"
2024-02-28 11:23:43 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-01"
2024-02-28 11:24:12 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-02"
2024-02-28 11:24:12 [ℹ]  waiting for the control plane to become ready
2024-02-28 11:24:13 [✔]  saved kubeconfig as "/root/.kube/config"
2024-02-28 11:24:13 [ℹ]  no tasks
2024-02-28 11:24:13 [✔]  all EKS cluster resources for "eks-cube-dev" have been created
2024-02-28 11:24:13 [ℹ]  nodegroup "eks-cube-dev-ng-01" has 2 node(s)
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-28-234.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-47-206.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  waiting for at least 2 node(s) to become ready in "eks-cube-dev-ng-01"
2024-02-28 11:24:13 [ℹ]  nodegroup "eks-cube-dev-ng-01" has 2 node(s)
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-28-234.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-47-206.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  nodegroup "eks-cube-dev-ng-02" has 2 node(s)
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-11-179.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-35-225.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  waiting for at least 2 node(s) to become ready in "eks-cube-dev-ng-02"
2024-02-28 11:24:13 [ℹ]  nodegroup "eks-cube-dev-ng-02" has 2 node(s)
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-11-179.ap-south-1.compute.internal" is ready
2024-02-28 11:24:13 [ℹ]  node "ip-192-168-35-225.ap-south-1.compute.internal" is ready
2024-02-28 11:24:14 [ℹ]  kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
2024-02-28 11:24:14 [✔]  EKS cluster "eks-cube-dev" in "ap-south-1" region is ready
```

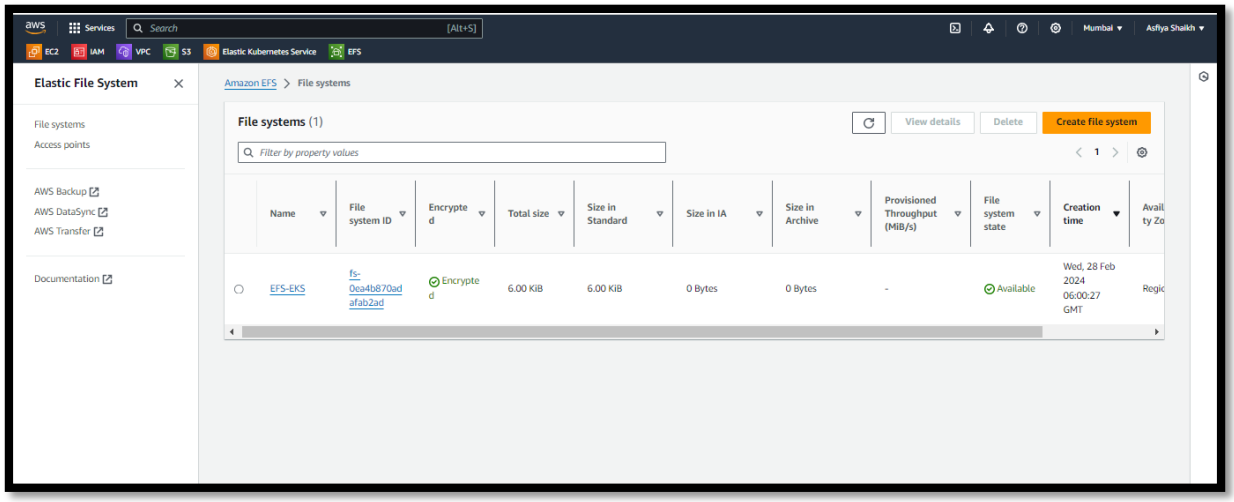
root@DESKTOP-8OOG2HF:~# aws eks --region ap-south-1 update-kubeconfig --name eks-cube-dev

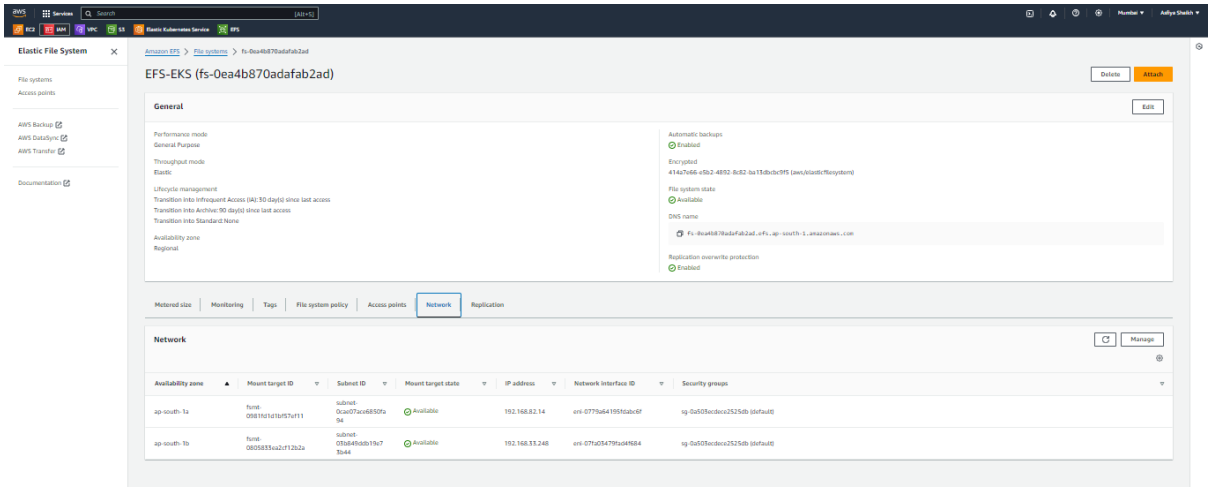
Updated context arn:aws:eks:ap-south-1:313897473255:cluster/eks-cube-dev in /root/.kube/config

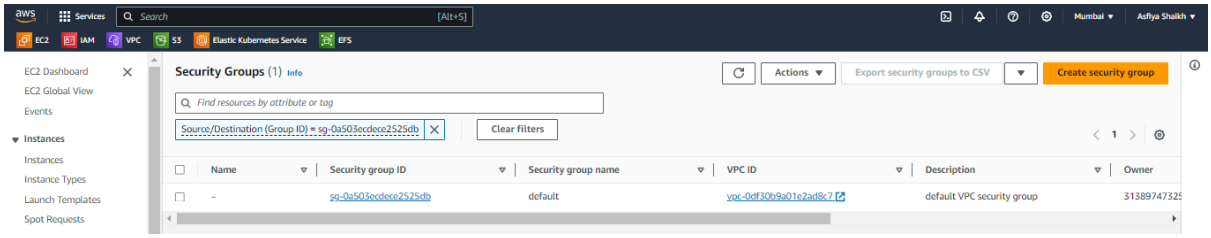
root@DESKTOP-8OOG2HF:~# kgn

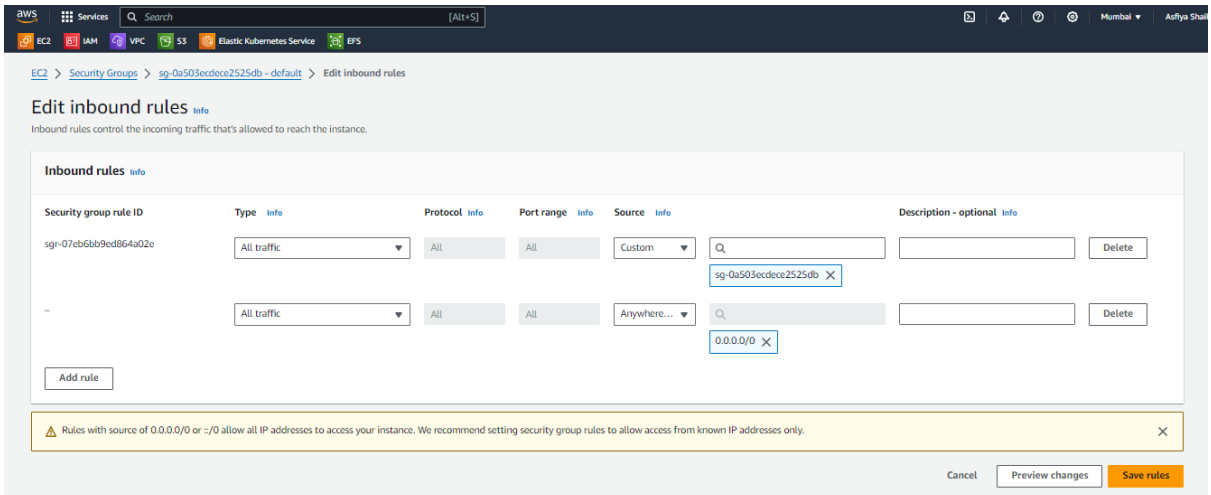
```
root@DESKTOP-8OOG2HF:~# aws eks --region ap-south-1 update-kubeconfig --name eks-cube-dev
Updated context arn:aws:eks:ap-south-1:313897473255:cluster/eks-cube-dev in /root/.kube/config
root@DESKTOP-8OOG2HF:~# kgn
NAME                                           STATUS   ROLES    AGE     VERSION
ip-192-168-11-179.ap-south-1.compute.internal   Ready    <none>   3m34s   v1.27.9-eks-5e0fdde
ip-192-168-28-234.ap-south-1.compute.internal   Ready    <none>   3m32s   v1.27.9-eks-5e0fdde
ip-192-168-35-225.ap-south-1.compute.internal   Ready    <none>   3m33s   v1.27.9-eks-5e0fdde
ip-192-168-47-206.ap-south-1.compute.internal   Ready    <none>   3m50s   v1.27.9-eks-5e0fdde
```

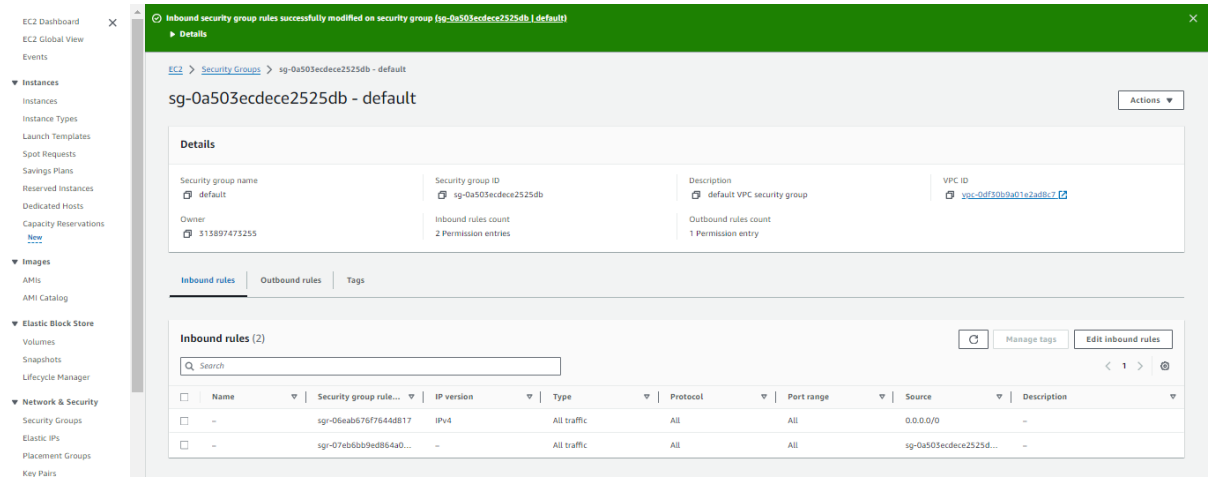
- ➔ Now Go to AWS
- ➔ Click on EFS Service
- ➔ and Create File Stystem.

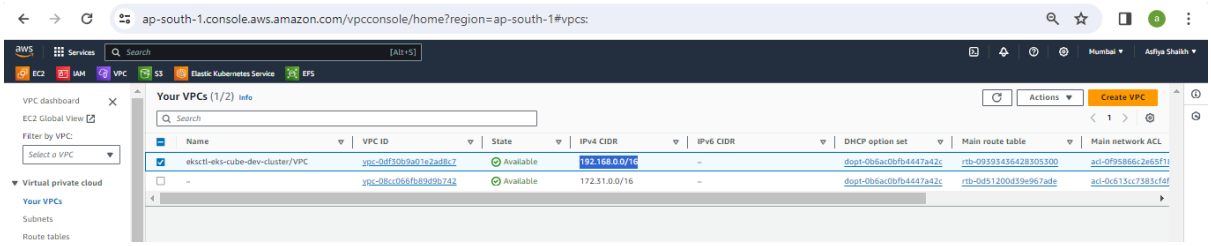


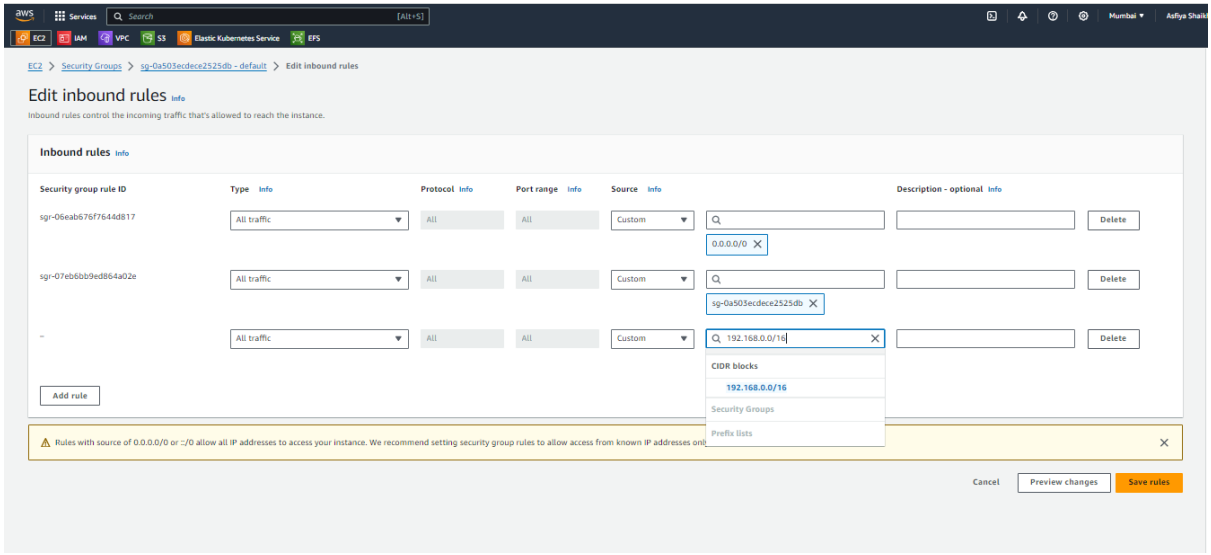

Elastic File System
EFS-EKS (fs-0ea4b870adafab2ad)
General
Network

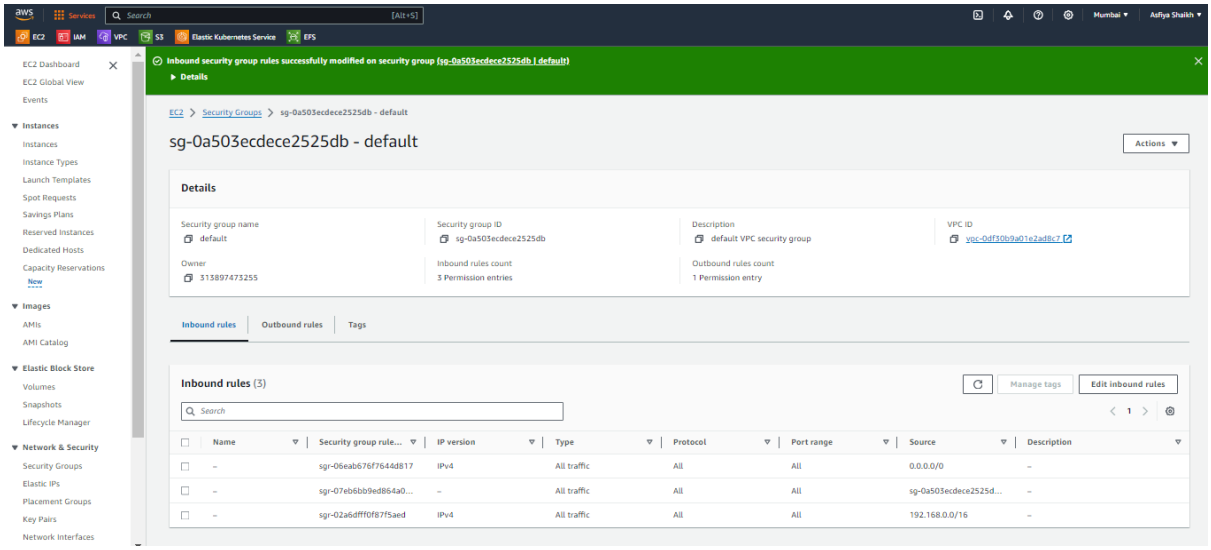

Security Groups (1) Info
Actions
Export security groups to CSV
Create security group
Clear filters
Name
Security group ID
Security group name
VPC ID
Description
Owner
default
default VPC security group


Edit inbound rules
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Security group rule ID
Type
Protocol
Port range
Source
Description - optional
sgr-07eb6bb9ed864a02e
All traffic
Custom
sg-0a503ecdece2525db
Delete
All traffic
Anywhere...
0.0.0.0/0
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Preview changes
Save rules


Inbound security group rules successfully modified on security group (sg-0a503ecdece2525db | default)
sg-0a503ecdece2525db - default
Actions
Details
Security group name
default
Security group ID
sg-0a503ecdece2525db
Description
default VPC security group
VPC ID
Owner
313897473255
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules (2)
Manage tags
Edit inbound rules







root@DESKTOP-8OOG2HF:eks-pv# vim PV.yaml

root@DESKTOP-8OOG2HF:eks-pv# cat PV.yaml

```
#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-0ea4b870adafab2ad
```

root@DESKTOP-8OOG2HF:eks-pv# vim PVC.yaml

root@DESKTOP-8OOG2HF:eks-pv# cat PVC.yaml

```
#pvc.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: efs-claim
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: ""
  resources:
    requests:
      storage: 5Gi
```

root@DESKTOP-8OOG2HF:eks-pv# vim Pod.yaml

root@DESKTOP-8OOG2HF:eks-pv# cat Pod.yaml

```
#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-8OOG2HF:eks-pv# ll

total 20

drwxr-xr-x 2 root root 4096 Feb 28 12:06 ./

drwx------ 40 root root 4096 Feb 28 12:06 ../

-rw-r--r-- 1 root root 308 Feb 28 12:05 PV.yaml

-rw-r--r-- 1 root root 195 Feb 28 12:06 PVC.yaml

-rw-r--r-- 1 root root 391 Feb 28 12:06 Pod.yaml

root@DESKTOP-8OOG2HF:eks-pv# k apply -f PV.yaml

persistentvolume/efs-pv created

root@DESKTOP-8OOG2HF:eks-pv# kubectl get pv efs-pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE

efs-pv 5Gi RWO Retain Available 92s

root@DESKTOP-8OOG2HF:eks-pv# k apply -f PVC.yaml

persistentvolumeclaim/efs-claim created

root@DESKTOP-8OOG2HF:eks-pv# kubectl get pvc efs-claim

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

efs-claim Bound efs-pv 5Gi RWO 20s

root@DESKTOP-8OOG2HF:eks-pv# k apply -f Pod.yaml

pod/efs-app created

root@DESKTOP-8OOG2HF:eks-pv# kubectl get pods

NAME READY STATUS RESTARTS AGE

efs-app 0/1 ContainerCreating 0 2m33s

root@DESKTOP-8OOG2HF:eks-pv# k logs efs-app

Error from server (BadRequest): container "app" in pod "efs-app" is waiting to start: ContainerCreating

==========

EFS DIVER INSTALLATION:

==========

root@DESKTOP-8OOG2HF:eks-pv# curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3

root@DESKTOP-8OOG2HF:eks-pv# chmod 700 get_helm.sh

root@DESKTOP-8OOG2HF:eks-pv# ./get_helm.sh

Downloading https://get.helm.sh/helm-v3.14.2-linux-amd64.tar.gz

Verifying checksum... Done.

Preparing to install helm into /usr/local/bin

helm installed into /usr/local/bin/helm

root@DESKTOP-8OOG2HF:eks-pv# helm version

version.BuildInfo{Version:"v3.14.2", GitCommit:"c309b6f0ff63856811846ce18f3bdc93d2b4d54b", GitTreeState:"clean", GoVersion:"go1.21.7"}

---

1. **Download the IAM policy document**

root@DESKTOP-8OOG2HF:eks-pv# curl -S https://raw.githubusercontent.com/kubernetes-sigs/aws-efs-csi-driver/v1.2.0/docs/iam-policy-example.json -o iam-policy.json

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   732  100   732    0     0   5580      0 --:--:-- --:--:-- --:--:--  5587
```

---

2. **Create an IAM policy**

root@DESKTOP-8OOG2HF:eks-pv# aws iam create-policy --policy-name EFSCSIControllerIAMPolicy --policy-document file://iam-policy.json


```
{
    "Policy": {
        "PolicyName": "EFSCSIControllerIAMPolicy",
        "PolicyId": "ANPAUSFN7KTTTJQ5WRSHS",
        "Arn": "arn:aws:iam::313897473255:policy/EFSCSIControllerIAMPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2024-02-28T06:59:55Z",
        "UpdateDate": "2024-02-28T06:59:55Z"
    }
}
```


root@DESKTOP-8OOG2HF:eks-pv# kubectl config get-contexts

```
root@DESKTOP-8OOG2HF:eks-pv# kubectl config get-contexts
CURRENT   NAME                                                  CLUSTER                                               AUTHINFO                                              NAM
ESPACE
*         arn:aws:eks:ap-south-1:313897473255:cluster/eks-cube-dev   arn:aws:eks:ap-south-1:313897473255:cluster/eks-cube-dev   arn:aws:eks:ap-south-1:313897473255:cluster/eks-cube-dev
          asfi@eks-cube-dev.ap-south-1.eksctl.io                eks-cube-dev.ap-south-1.eksctl.io                     asfi@eks-cube-dev.ap-south-1.eksctl.io
```

root@DESKTOP-8OOG2HF:eks-pv# eksctl utils associate-iam-oidc-provider --region=ap-south-1 --cluster=eks-cube-dev --approve

```
root@DESKTOP-8OOG2HF:eks-pv# eksctl utils associate-iam-oidc-provider --region=ap-south-1 --cluster=eks-cube-dev --approve
2024-02-28 12:41:38 [ℹ]  will create IAM Open ID Connect provider for cluster "eks-cube-dev" in "ap-south-1"
2024-02-28 12:41:39 [✔]  created IAM Open ID Connect provider for cluster "eks-cube-dev" in "ap-south-1"
```

---

3. **Create a Kubernetes service account**

root@DESKTOP-8OOG2HF:eks-pv# 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::313897473255:policy/EFSCSIControllerIAMPolicy –approve

```
root@DESKTOP-8OOG2HF:eks-pv# eksctl create iamserviceaccount --cluster=eks-cube-dev --region=ap-south-1 --namespace=kube-system --name=efs-csi-controller-sa --override-existing-serviceaccoun
ts --attach-policy-arn=arn:aws:iam::313897473255:policy/EFSCSIControllerIAMPolicy --approve
2024-02-28 12:41:46 [ℹ]  1 iamserviceaccount (kube-system/efs-csi-controller-sa) was included (based on the include/exclude rules)
2024-02-28 12:41:46 [!]  metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2024-02-28 12:41:46 [ℹ]  1 task: {
    2 sequential sub-tasks: {
        create IAM role for serviceaccount "kube-system/efs-csi-controller-sa",
        create serviceaccount "kube-system/efs-csi-controller-sa",
    } }2024-02-28 12:41:46 [ℹ]  building iamserviceaccount stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-28 12:41:47 [ℹ]  deploying stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-28 12:41:47 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-28 12:42:17 [ℹ]  waiting for CloudFormation stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-28 12:42:17 [ℹ]  created serviceaccount "kube-system/efs-csi-controller-sa"
```

---

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

root@DESKTOP-8OOG2HF:eks-pv# eksctl get iamserviceaccount --cluster eks-cube-dev --name efs-csi-controller-sa --namespace kube-system

NAMESPACE NAME ROLE ARN

kube-system efs-csi-controller-sa arn:aws:iam::313897473255:role/eksctl-eks-cube-dev-addon-iamserviceaccount-k-Role1-lxbvgh7ZVjYX

5. **Now install AWS EFS Storage Controller driver.**

root@DESKTOP-8OOG2HF:eks-pv# 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!⎈

root@DESKTOP-8OOG2HF:eks-pv# 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

Error: Kubernetes cluster unreachable: exec plugin: invalid apiVersion "client.authentication.k8s.io/v1alpha1"

root@DESKTOP-8OOG2HF:eks-pv# curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2

```
root@DESKTOP-8OOG2HF:eks-pv# curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  6875  100  6875    0     0   4884      0  0:00:01  0:00:01 --:--:--  4884
Helm v3.8.2 is available. Changing from version .
Downloading https://get.helm.sh/helm-v3.8.2-linux-amd64.tar.gz
Preparing to install helm and tiller into /usr/local/bin
helm installed into /usr/local/bin/helm
info: tiller binary was not found in this release; skipping tiller installation
Run 'helm init' to configure helm.
```

root@DESKTOP-8OOG2HF:eks-pv# 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

```
root@DESKTOP-8OOG2HF:eks-pv# curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  6875  100  6875    0     0   4884      0  0:00:01  0:00:01 --:--:--  4884
Helm v3.8.2 is available. Changing from version .
Downloading https://get.helm.sh/helm-v3.8.2-linux-amd64.tar.gz
Preparing to install helm and tiller into /usr/local/bin
helm installed into /usr/local/bin/helm
info: tiller binary was not found in this release; skipping tiller installation
Run 'helm init' to configure helm.
root@DESKTOP-8OOG2HF:eks-pv# 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.na
me=efs-csi-controller-sa
Release "aws-efs-csi-driver" does not exist. Installing it now.
NAME: aws-efs-csi-driver
LAST DEPLOYED: Wed Feb 28 13:12:03 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
To verify that aws-efs-csi-driver has started, run:

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

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

root@DESKTOP-8OOG2HF:eks-pv# kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-efs-csi-driver,app.kubernetes.io/instance=aws-efs-csi-driver"

```
root@DESKTOP-8OOG2HF:eks-pv# kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-efs-csi-driver,app.kubernetes.io/instance=aws-efs-csi-dri
ver"
NAME                                  READY   STATUS    RESTARTS   AGE
efs-csi-controller-598bf64f56-5wb7x   3/3     Running   0          33s
efs-csi-controller-598bf64f56-z8txc   3/3     Running   0          33s
efs-csi-node-79wqg                    3/3     Running   0          33s
efs-csi-node-gzpn2                    3/3     Running   0          33s
efs-csi-node-mjxqn                    3/3     Running   0          33s
efs-csi-node-qvv2r                    3/3     Running   0          33s
```

root@DESKTOP-8OOG2HF:eks-pv# k delete -f Pod.yaml

pod "efs-app" deleted

root@DESKTOP-8OOG2HF:eks-pv# k apply -f Pod.yaml

pod/efs-app created

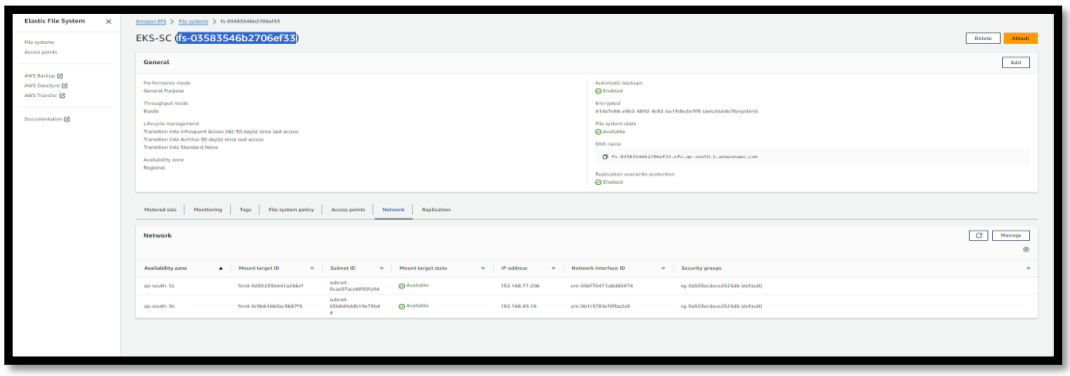
root@DESKTOP-8OOG2HF:eks-pv# kgp

NAME READY STATUS RESTARTS AGE

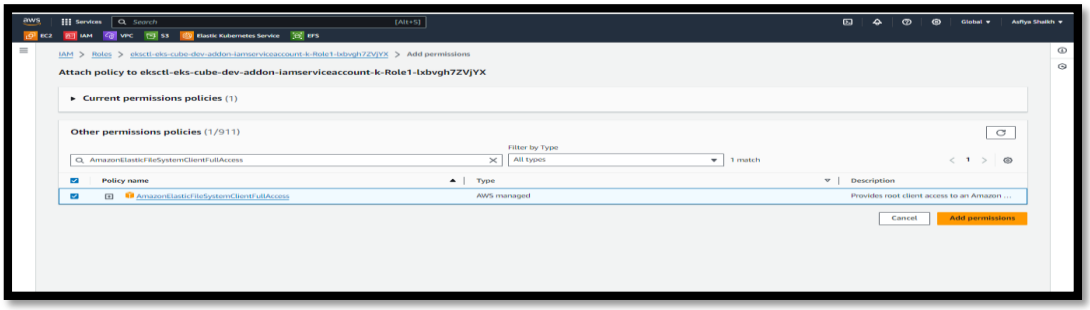
efs-app 1/1 Running 0 70s

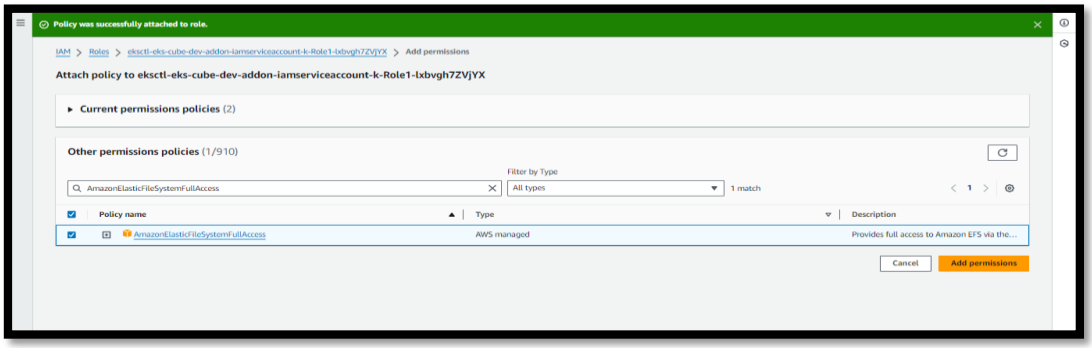
```
root@DESKTOP-8OOG2HF:eks-pv# k get pv,pvc,po
NAME                      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM               STORAGECLASS   REASON   AGE
persistentvolume/efs-pv   5Gi        RWO            Retain           Bound    default/efs-claim                           68m

NAME                                STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
persistentvolumeclaim/efs-claim     Bound    efs-pv   5Gi        RWO                           66m

NAME          READY   STATUS    RESTARTS   AGE
pod/efs-app   1/1     Running   0          115s
```

➢ EKS -> SC,PVC,POD



Then, we want to go into IAM service -> ROLES .

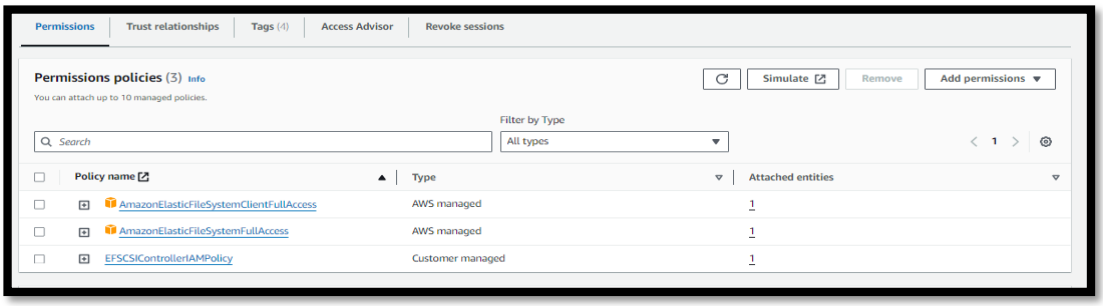




Attached Two Policies In

eksctl-eks-cube-dev-addon-iamserviceaccount-k-Role1-lxbvgh7ZVjYX

1. **AmazonElasticFileSystemClientFullAccess**
2. **AmazonElasticFileSystemFullAccess**



root@DESKTOP-8OOG2HF:eks-sc# vim sc.yaml

root@DESKTOP-8OOG2HF:eks-sc# cat sc.yaml

```
# sc.yaml
---
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: efs-sc
provisioner: efs.csi.aws.com
parameters:
  provisioningMode: efs-ap
  fileSystemId: fs-03583546b2706ef33
  directoryPerms: "700"
```

root@DESKTOP-8OOG2HF:eks-sc# vim pvc.yaml

root@DESKTOP-8OOG2HF:eks-sc# cat pvc.yaml

```
root@DESKTOP-8OOG2HF:eks-sc# cat pvc.yaml
#pvc.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: efs-claim-1
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: efs-sc
  resources:
    requests:
      storage: 5Gi
```

root@DESKTOP-8OOG2HF:eks-sc# vim pod.yaml

root@DESKTOP-8OOG2HF:eks-sc# cat pod.yaml

```
#pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: efs-app-1
spec:
  containers:
    - name: app
      image: centos
      command: ["/bin/sh"]
      args: ["-c", "while true; do echo $(date -u) >> /data/out; sleep 5; done"]
      volumeMounts:
        - name: persistent-storage
          mountPath: /data
  volumes:
    - name: persistent-storage
      persistentVolumeClaim:
        claimName: efs-claim-1
```

root@DESKTOP-8OOG2HF:eks-sc# kubectl apply -f sc.yaml

storageclass.storage.k8s.io/efs-sc created

```
root@DESKTOP-8OOG2HF:eks-sc# k get sc
NAME            PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
efs-sc          efs.csi.aws.com         Delete          Immediate              false                  34m
gp2 (default)   kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   false                  3h52m
```

root@DESKTOP-8OOG2HF:eks-sc# kubectl apply -f pvc.yaml

persistentvolumeclaim/efs-claim-1 created

```
root@DESKTOP-8OOG2HF:eks-sc# k get pvc
NAME          STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
efs-claim-1   Bound    pvc-6c272555-b6ff-4b40-80ec-d39b7a32f448   5Gi        RWX            efs-sc         33m
```

root@DESKTOP-8OOG2HF:eks-sc# k apply -f pod.yaml

pod/efs-app-1 created

```
root@DESKTOP-8OOG2HF:eks-sc# k get po
NAME         READY   STATUS    RESTARTS   AGE
efs-app-1    1/1     Running   0          19m
```

root@DESKTOP-8OOG2HF:eks-sc# k get sc,pvc,po

```
root@DESKTOP-8OOG2HF:eks-sc# k get sc,pvc,po
NAME                                    PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
storageclass.storage.k8s.io/efs-sc      efs.csi.aws.com         Delete          Immediate              false                  36m
storageclass.storage.k8s.io/gp2 (default)   kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   false                  3h53m

NAME                                  STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
persistentvolumeclaim/efs-claim-1     Bound    pvc-6c272555-b6ff-4b40-80ec-d39b7a32f448   5Gi        RWX            efs-sc         35m

NAME            READY   STATUS    RESTARTS   AGE
pod/efs-app-1   1/1     Running   0          21m
```

root@DESKTOP-8OOG2HF:eks-sc# kubectl get pv | grep efs-sc

pvc-6c272555-b6ff-4b40-80ec-d39b7a32f448 5Gi RWX Delete Bound default/efs-claim-1 efs-sc 33m

root@DESKTOP-8OOG2HF:eks-sc# kubectl get pods

NAME READY STATUS RESTARTS AGE

efs-app-1 1/1 Running 0 32m

root@DESKTOP-8OOG2HF:eks-sc# kubectl exec -ti efs-app-1 -- tail -f /data/out

Wed Feb 28 09:49:43 UTC 2024

Wed Feb 28 09:49:48 UTC 2024

Wed Feb 28 09:49:53 UTC 2024

Wed Feb 28 09:49:58 UTC 2024

Wed Feb 28 09:50:03 UTC 2024

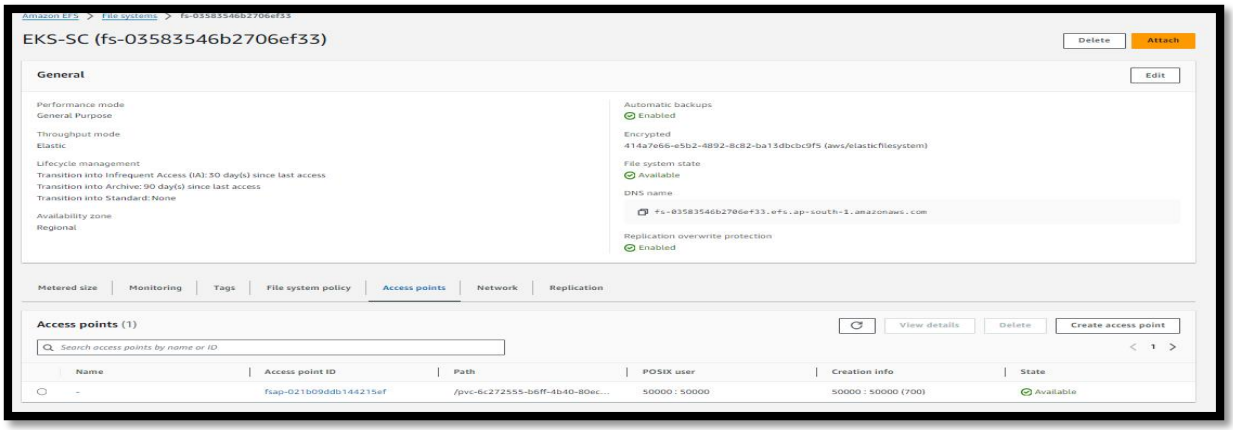
Wed Feb 28 09:50:08 UTC 2024

Wed Feb 28 09:50:13 UTC 2024

Wed Feb 28 09:50:18 UTC 2024

Wed Feb 28 09:50:23 UTC 2024

Wed Feb 28 09:50:28 UTC 2024


Amazon EFS > File systems > fs-03583546b2706ef33
EKS-SC (fs-03583546b2706ef33)
Delete
Attach
General
Edit
Performance mode
General Purpose
Throughput mode
Elastic
Lifecycle management
Transition into Infrequent Access (IA): 30 day(s) since last access
Transition into Archive: 90 day(s) since last access
Transition into Standard: None
Availability zone
Regional
Automatic backups
Enabled
Encrypted
414a7e66-e5b2-4892-8c82-ba13dbcbc9f5 (aws/elasticfilesystem)
File system state
Available
DNS name
fs-03583546b2706ef33.efs.ap-south-1.amazonaws.com
Replication overwrite protection
Enabled
Metered size
Monitoring
Tags
File system policy
Access points
Network
Replication
Access points (1)
View details
Delete
Create access point
Search access points by name or ID
1
Name
Access point ID
Path
POSIX user
Creation info
State
-
fsap-021b09ddb144215ef
/pvc-6c272555-b6ff-4b40-80ec...
50000 : 50000
50000 : 50000 (700)
Available