

INSTALLATION OF EFS ON EKS CLUSTER

1. INSTALL EKSCTL ON LINUX.

```
root@DESKTOP-VIDGD8F:~# eksctl create cluster -f eks-cube-cluster.yaml
2024-02-27 12:32:02 [i] eksctl version 0.171.0
2024-02-27 12:32:02 [i] using region ap-south-1
2024-02-27 12:32:02 [i] skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.medium,t2.small
2024-02-27 12:32:02 [i] setting availability zones to [ap-south-1a ap-south-1b]
2024-02-27 12:32:02 [i] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2024-02-27 12:32:02 [i] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2024-02-27 12:32:03 [i] nodegroup "eks-cube-dev-ng-01" will use "ami-0ee9bbeeb96514b3b" [AmazonLinux2/1.27]
2024-02-27 12:32:03 [i] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-01-4f:d5:d7:98:8d:8b:6b:06:1c:ef:67:cf:a9:a9:4c:e2"
2024-02-27 12:32:03 [i] nodegroup "eks-cube-dev-ng-02" will use "ami-0ee9bbeeb96514b3b" [AmazonLinux2/1.27]
2024-02-27 12:32:03 [i] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-eks-cube-dev-nodegroup-eks-cube-dev-ng-02-4f:d5:d7:98:8d:8b:6b:06:1c:ef:67:cf:a9:a9:4c:e2"
2024-02-27 12:32:03 [i] using Kubernetes version 1.27
2024-02-27 12:32:03 [i] creating EKS cluster "eks-cube-dev" in "ap-south-1" region with un-managed nodes
2024-02-27 12:32:03 [i] 2 nodegroups (eks-cube-dev-ng-01, eks-cube-dev-ng-02) were included (based on the include/exclude rules)
2024-02-27 12:32:03 [i] will create a CloudFormation stack for cluster itself and 2 nodegroup stack(s)
2024-02-27 12:32:03 [i] will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
2024-02-27 12:32:03 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=eks-cube-dev'
2024-02-27 12:32:03 [i] Kubernetes API endpoint access will use default of [publicAccess=true, privateAccess=false] for cluster "eks-cube-dev" in "ap-south-1"
2024-02-27 12:32:03 [i] CloudWatch logging will not be enabled for cluster "eks-cube-dev" in "ap-south-1"
2024-02-27 12:32:03 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=eks-cube-dev'
2024-02-27 12:32:03 [i]
2 sequential tasks: { create cluster control plane "eks-cube-dev",
  2 sequential sub-tasks: {
    wait for control plane to become ready,
    2 parallel sub-tasks: {
      create nodegroup "eks-cube-dev-ng-01",
      create nodegroup "eks-cube-dev-ng-02",
    },
  },
}
2024-02-27 12:32:03 [i] building cluster stack "eksctl-eks-cube-dev-cluster"
2024-02-27 12:32:04 [i] deploying stack "eksctl-eks-cube-dev-cluster"
```

```
root@DESKTOP-VIDGD8F:~# kgn
NAME                                     STATUS    ROLES    AGE      VERSION
ip-192-168-14-80.ap-south-1.compute.internal Ready    <none>    7m44s    v1.27.9-eks-5e0fdde
ip-192-168-38-174.ap-south-1.compute.internal Ready    <none>    7m59s    v1.27.9-eks-5e0fdde
ip-192-168-4-61.ap-south-1.compute.internal Ready    <none>    8m6s     v1.27.9-eks-5e0fdde
ip-192-168-61-72.ap-south-1.compute.internal Ready    <none>    8m9s     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 [✓] 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
```

```
root@DESKTOP-VIDGD8F:KUBERNETES# 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
           % Done    Dload  Upload  Total   Spent    Left   Speed
100  732  100  732    0     0   846      0 --:--:-- --:--:-- --:--:--   846
root@DESKTOP-VIDGD8F:KUBERNETES#
```

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": "ANPATXEPQB7UJVPEZTSN",
    "Arn": "arn:aws:iam::255851499496:policy/EFSCSIControllerIAMPolicy",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 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 \
--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-VIDGD8F: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 21:36:09 [✓] 1 iamserviceaccount (Kube-system/efs-csi-controller-sa) was included (based on the include/exclude rules)
2024-02-27 21:36:09 [!] metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2024-02-27 21:36:09 [✓] 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"
  } }2024-02-27 21:36:09 [✓] building iamserviceaccount stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-27 21:36:09 [✓] deploying stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-27 21:36:40 [✓] waiting for CloudFormation stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-27 21:36:40 [✓] waiting for CloudFormation stack "eksctl-eks-cube-dev-addon-iamserviceaccount-kube-system-efs-csi-controller-sa"
2024-02-27 21:36:42 [✓] created serviceaccount "Kube-system/efs-csi-controller-sa"
Activata Windows
```

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 kube-system`

```
root@DESKTOP-VIDGD8F:task4# 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::255851499496:role/eksctl-eks-cube-dev-addon-iamserviceaccount-k-Role1-tcyY1eyt4vuN
root@DESKTOP-VIDGD8F:task4#
```

Now install AWS EFS Storage Controller driver.

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

```
root@DESKTOP-VIDGD8F:k8sCluster_kubeadm_terraform# helm repo add aws-efs-csi-driver https://kubernetes-sigs.github.io/aws-efs-csi-driver
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /mnt/c/Users/ADMIN/Desktop/My_Notes/KUBERNETES/k8sCluster_kubeadm_terraform/suitable
-ant.conf
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /mnt/c/Users/ADMIN/Desktop/My_Notes/KUBERNETES/k8sCluster_kubeadm_terraform/suitable
-ant.conf
"aws-efs-csi-driver" has been added to your repositories
root@DESKTOP-VIDGD8F:k8sCluster_kubeadm_terraform#
```

`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`

```
root@DESKTOP-VIDGD8F:~# 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
Release "aws-efs-csi-driver" does not exist. Installing it now.
NAME: aws-efs-csi-driver
LAST DEPLOYED: Tue Feb 27 18:39:02 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"
root@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-csi-driver,app.kubernetes.io/instance=aws-efs-csi-driver"`

```
root@DESKTOP-VIDGD8F:~# kubectl get pod -n kube-system -l "app.kubernetes.io/name=aws-efs-csi-driver,app.kubernetes.io/instance=aws-efs-csi-driver"
NAME                                READY   STATUS    RESTARTS   AGE
efs-csi-controller-598bf64f56-9dpbp 3/3     Running   0           15m
efs-csi-controller-598bf64f56-dgrmg 3/3     Running   0           15m
efs-csi-node-dfbrn                   3/3     Running   0           15m
efs-csi-node-fwt9l                   3/3     Running   0           15m
efs-csi-node-x8l7l                   3/3     Running   0           15m
efs-csi-node-s5slj                   3/3     Running   0           15m
root@DESKTOP-VIDGD8F:~#
```

To create iamserviceaccount.

```
eksctl create iamserviceaccount \  
  
--cluster=eks-cube-test \  
  
--namespace=kube-system \  
  
--name=aws-load-balancer-controller \  
  
--attach-policy-arn=arn:aws:iam::509002973204:policy/AWSLoadBalancerControllerIAMPolicy \  
  
--override-existing-serviceaccounts \  
  
--approve
```

```
root@DESKTOP-VIDGD8F:KUBERNETES# eksctl create iamserviceaccount --cluster=eks-cube-dev --namespace=kube-system --name=aws-load-balancer-controller --attach-policy-arn=arn:aws:iam::255851499496: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 [!] metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set  
2024-02-27 18:59:40 [■] no tasks  
root@DESKTOP-VIDGD8F:KUBERNETES#
```

Create file system

Create an EFS file system with recommended settings. [Learn more](#)

Name - optional
Name your file system.

EKS-FILE-SYSTEM

Name can include letters, numbers, and +-=._:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system.

vpc-067959ebf0b6b713c
eksctl-eks-cube-dev-cluster/VPC

Cancel

Customize

Create

Success!
File system (fs-0541e4ae7182a946f) is available.

Notifications 0 0 1 1 0 0

Amazon EFS > File systems

File systems (1) [View details](#) [Delete](#) [Create file system](#)

Filter by property values

	Name	File system ID	Encryption	Total size	Size in Standard	Size in IA	Size in Archive	Provisioned Throughput (MiB/s)
<input type="radio"/>	EKS-FILE-SYSTEM	fs-0541e4ae7182a946f	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes	-

Amazon EFS > File systems > fs-0541e4ae7182a946f

EKS-FILE-SYSTEM (fs-0541e4ae7182a946f) [Delete](#) [Attach](#)

General [Edit](#)

Performance mode General Purpose	Automatic backups Enabled
Throughput mode Elastic	Encrypted dc4af486-d181-49cc-9eee-3e11ddc5b3e2 (aws/elasticfilesystem)
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	File system state Available
Availability zone Regional	DNS name fs-0541e4ae7182a946f.efs.ap-south-1.amazonaws.com
	Replication overwrite protection Enabled

```
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  5m17s
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 pod.yaml
pod/efs-app created
root@DESKTOP-VIDGD8F:task4#

```

EC2 > Security Groups > sg-0a918b4a819058672 - default > Edit inbound rules

Edit inbound rules [info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type info	Protocol info	Port range info	Source info	Description - optional info	
sg-05622f16d44fb4460	All traffic	All	All	Custom	Q	sg-0a918b4a819058672 X
-	All traffic	All	All	Custom	Q 192.168.0.0/16 X	192.168.0.0/16 X

[Add rule](#)

Cancel [Preview changes](#) [Save rules](#)

```

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

```

```

root@DESKTOP-VIDGD8F:task4# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
efs-app       1/1     Running   0           61s

```

```

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    standard      66s

```

```

root@DESKTOP-VIDGD8F:task4# kubectl get pvc
NAME          STATUS   VOLUME    CAPACITY   ACCESS MODES   STORAGECLASS   AGE
efs-claim     Bound   efs-pv    5Gi        RWO            standard      69s

```

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

Success!

File system (fs-0471de01175284441) is available.

View file system

[Amazon EFS](#) > [File systems](#) > fs-0471de01175284441

EKS-FILE-SYSTEM (fs-0471de01175284441)

DeleteAttach

General

Edit

Performance mode

General Purpose

Throughput mode

Elastic

Lifecycle management

Transition into Infrequent Access (IA): None

Transition into Archive: None

Transition into Standard: None

Availability zone

Regional

Automatic backups

Disabled

Encrypted

dc4af486-d181-49cc-9eee-3e11ddc5b3e2 (aws/elasticfilesystem)

File system state

Available

DNS name

No mount targets available

Replication overwrite protection

Enabled

Activate Windows

```
root@DESKTOP-VIDGD8F:task4# vim sc.yaml
root@DESKTOP-VIDGD8F:task4# cat sc.yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: efs-sc
provisioner: efs.csi.aws.com
parameters:
  provisioningMode: efs-ap
  fileSystemId: fs-0471de01175284441
  directoryPerms: "700"
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:
    - 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-VIDGD8F:task4# k get sc,pvc,po

```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
storageclass.storage.k8s.io/efs-sc	efs.csi.aws.com	Delete	Immediate	false	8m10s
storageclass.storage.k8s.io/gp2 (default)	kubernetes.io/aws-efs	Delete	WaitForFirstConsumer	false	113m

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
persistentvolumeclaim/efs-claim	Bound	pvc-04082afd-552c-402f-88af-797a172ada48	5Gi	RWX	efs-sc	8m4s

NAME	READY	STATUS	RESTARTS	AGE
pod/efs-app	1/1	Running	0	7m55s

```

root@DESKTOP-VIDGD8F:task4# █

```

```
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 08:01:29 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 08:01:39 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
```