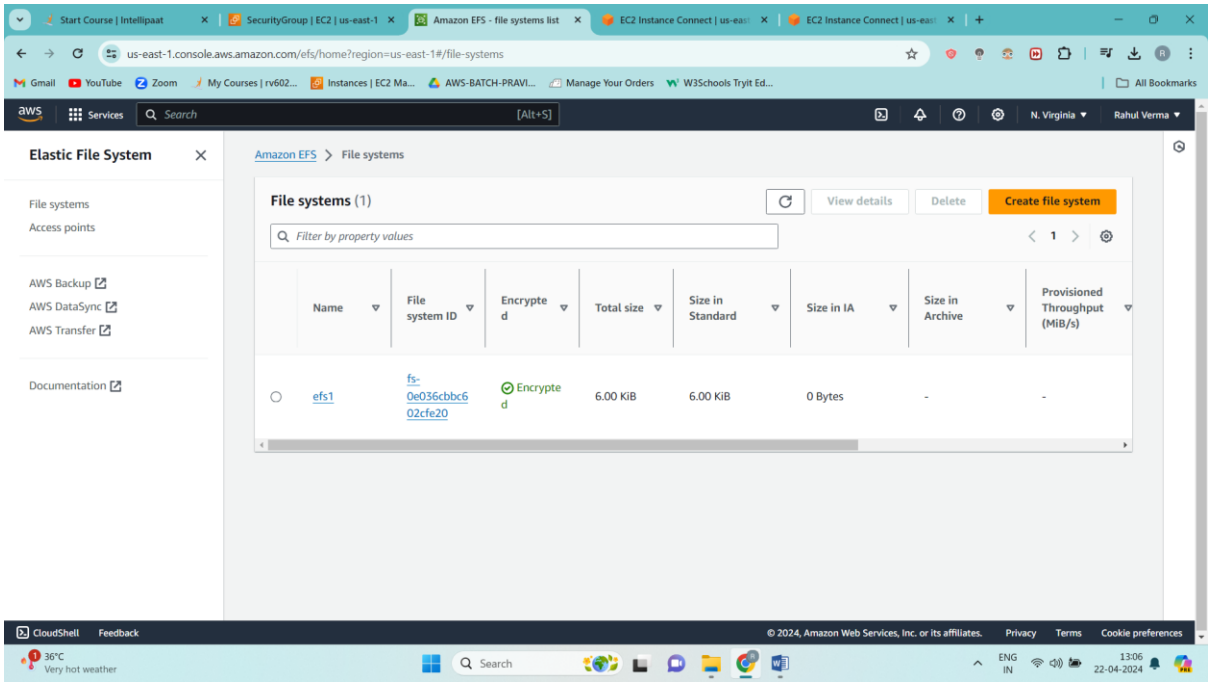
Creating a Persistent volume

Operation 1: create EFS on your aws console



Mount and attach this EFS commands-

sudo apt update

sudo apt-get install nfs-common –y

mkdir efs

ls

Copy NFS client code from EFS and it's done

Operation 2: run the following commands now-

Download the repository

git clone https://github.com/kubernetes-incubator/external-storage

```
ubuntu@ip-172-31-83-79:~/efs$ sudo mkdir file
ubuntu@ip-172-31-83-79:~/efs$ ls
file
ubuntu@ip-172-31-83-79:~/efs$ cd ..
ubuntu@ip-172-31-83-79:~$ git clone https://github.com/kubernetes-incubator/external-storage
Cloning into 'external-storage'...
remote: Enumerating objects: 64319, done.
remote: Total 64319 (delta 0), reused 0 (delta 0), pack-reused 64319
Receiving objects: 100% (64319/64319), 113.79 MiB | 22.39 MiB/s, done.
Resolving deltas: 100% (29663/29663), done.
```

Switch to the deploy directory

cd external-storage/aws/efs/deploy/

```
ubuntu@ip-172-31-83-79:~$ cd external-storage/aws/efs/deploy/
```

Apply rbac permissions

kubectl apply –f rbac.yaml

```
ubuntu@ip-172-31-83-79:~/external-storage/aws/efs/deploy$ kubectl apply -f rbac.yaml
clusterrole.rbac.authorization.k8s.io/efs-provisioner-runner created
clusterrolebinding.rbac.authorization.k8s.io/run-efs-provisioner created
role.rbac.authorization.k8s.io/leader-locking-efs-provisioner created
rolebinding.rbac.authorization.k8s.io/leader-locking-efs-provisioner created
ubuntu@ip-172-31-83-79:~/external-storage/aws/efs/deploy$
```

Operation 3: Create the following manifest file and make changes according to your configurations

nano <file name>.yaml

copy the below content-

---

apiVersion: v1

kind: ConfigMap

metadata:

name: efs-provisioner

data:

file.system.id: fs-0e036cbbc602cfe20

aws.region: us-east-1

provisioner.name: example.com/aws-efs

dns.name: ""

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: efs-provisioner

---

kind: Deployment

apiVersion: apps/v1

metadata:

name: efs-provisioner

spec:

replicas: 1

selector:

matchLabels:

app: efs-provisioner

strategy:

type: Recreate

template:

metadata:

labels:

app: efs-provisioner

spec:

serviceAccount: efs-provisioner

containers:

- name: efs-provisioner

image: quay.io/external_storage/efs-provisioner:latest

env:

- name: FILE_SYSTEM_ID

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: file.system.id

- name: AWS_REGION

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: aws.region

- name: DNS_NAME

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: dns.name

optional: true

- name: PROVISIONER_NAME

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: provisioner.name

volumeMounts:

- name: pv-volume

mountPath: /persistentvolumes

volumes:

- name: pv-volume

nfs:

server: fs-e4413b65.efs.us-east-1.amazonaws.com

path: /

---

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: aws-efs

provisioner: example.com/aws-efs

---

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: efs

annotations:

volume.beta.kubernetes.io/storage-class: "aws-efs"

spec:

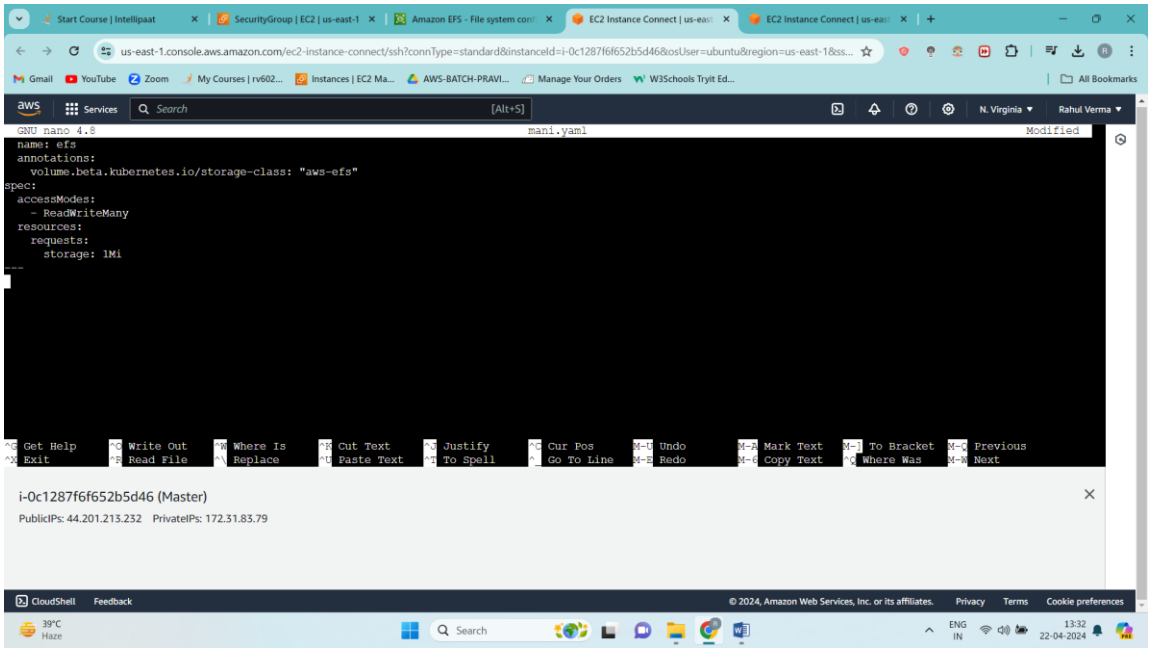
accessModes:

- ReadWriteMany

resources:

requests:

storage: 1Mi



Save and exit by pressing ctrl+s and ctrl+x

Operation 4: now create pod using this yaml file

kubectl create –f <filename>.yaml

```
ubuntu@ip-172-31-83-79:~/external-storage/aws/efs/deploy$ kubectl create -f mani.yaml
configmap/efs-provisioner created
serviceaccount/efs-provisioner created
deployment.apps/efs-provisioner created
storageclass.storage.k8s.io/aws-efs created
Warning: metadata.annotations[volume.beta.kubernetes.io/storage-class]: deprecated since v1.8; use "storageClassName" attribute instead
persistentvolumeclaim/efs created
ubuntu@ip-172-31-83-79:~/external-storage/aws/efs/deploy$
```

To check persistent volume is currently working or not for that just type

kubectl get pvc

```
ubuntu@ip-172-31-83-79:~/external-storage/aws/efs/deploy$ kubectl get pvc
NAME    STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    AGE
efs     Pending                                         aws-efs         7m37s
ubuntu@ip-172-31-83-79:~/external-storage/aws/efs/deploy$
```

It will take some time to bound it

Operation 5: create a deployment calling the claims created and volume

sudo nano pv.yaml

copy the below content and paste it in nano editor-

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pv-deploy
spec:
  replica: 1
  selector:
    matchLabels:
      app: mypv
  template:
```

```
  metadata:
    labels:
      app: mypv
  spec:
    containers:
    - name: shell
      Image: centos:7
      command:
      - "bin/bash"
      - "-c"
      - "sleep 10000"
      volumeMounts:
      - name: mypd
        mountPath: "/tmp/persistent"
    volumes:
    - name: mypd
      persistentVolumesClaim:
        claimName: efs
```

press ctrl+s and ctrl+x to save and exit from nano editor

Now create pod using this file

kubectl create –f <filename>.yaml

write this command to check all the pods

kubectl get pods

Operation 6: Now to get inside for your pod

kubectl exec –it <name_of_your_pod> -- bash

touch /tmp/persistent/data (to create data file in /tmp/persistent location)

ls /tmp/persistent

This will show you file data

Now press ctrl+d to get out of the root

Operation 7: now delete the the deployment pod which you created just now

kubectl delete pod <pod_name>

now write-

kubectl get pods (new deployment pod will be created)

and if you get inside that pod and check it's content

you will see our data file.