

**In this project i have launched joomla on the top of amazon EKS(Elastic kubernetes Service) cluster. In this project i have used EFS(Elastic file system) as a storage class for persistent volume claim(pvc) or persistent volume.**

**What is EKS?**

Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS without need to install, operate, and maintain your own Kubernetes control plane.

**What is EFS?**

It is a storage service in aws. It provides centralised storage facility unlike EBS which is required in multi-node setup since podes runs on different nodes.

**Note:- First i was launching wix on the top of eks but i was facing some issue in launching wix then i tested this setup for joomla and wordpress and it worked fine. So you will see many palces wix written, don't be confuse with this consider wix as joomla. I will rectify later why this setup is not working for wix.**

**Things you required before this set up.**

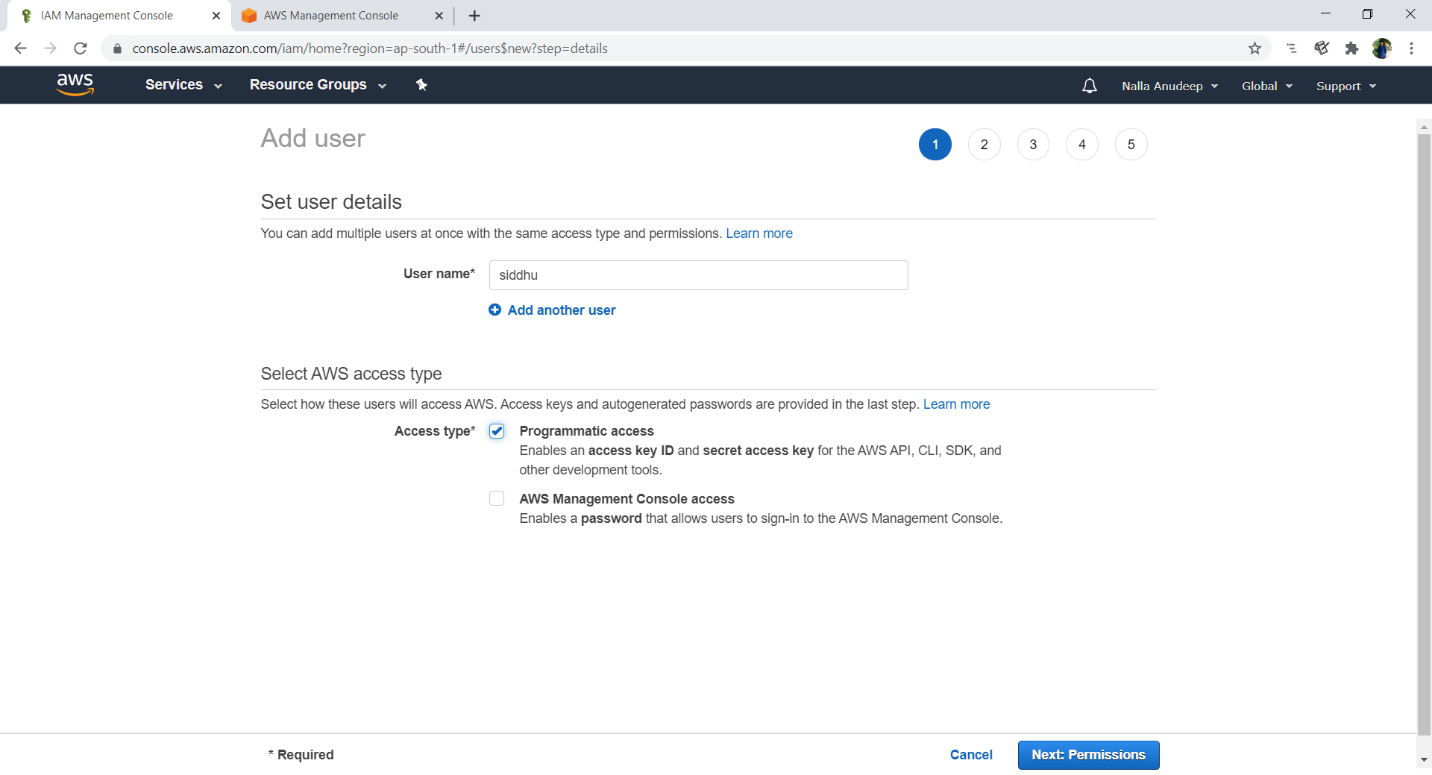
1. AWS account with one IAM user

**Lets start with explanation**

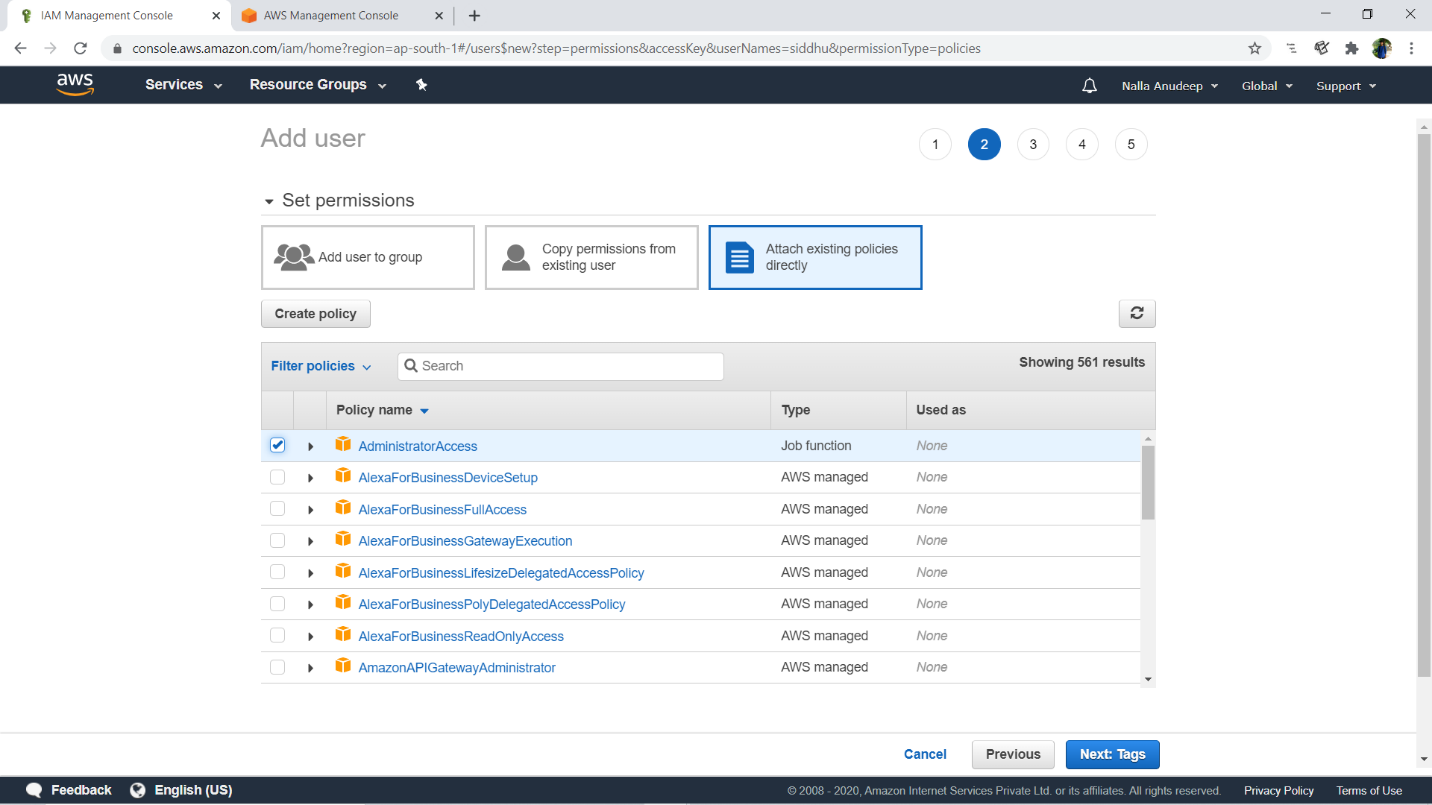
1 create IAM user with administration power

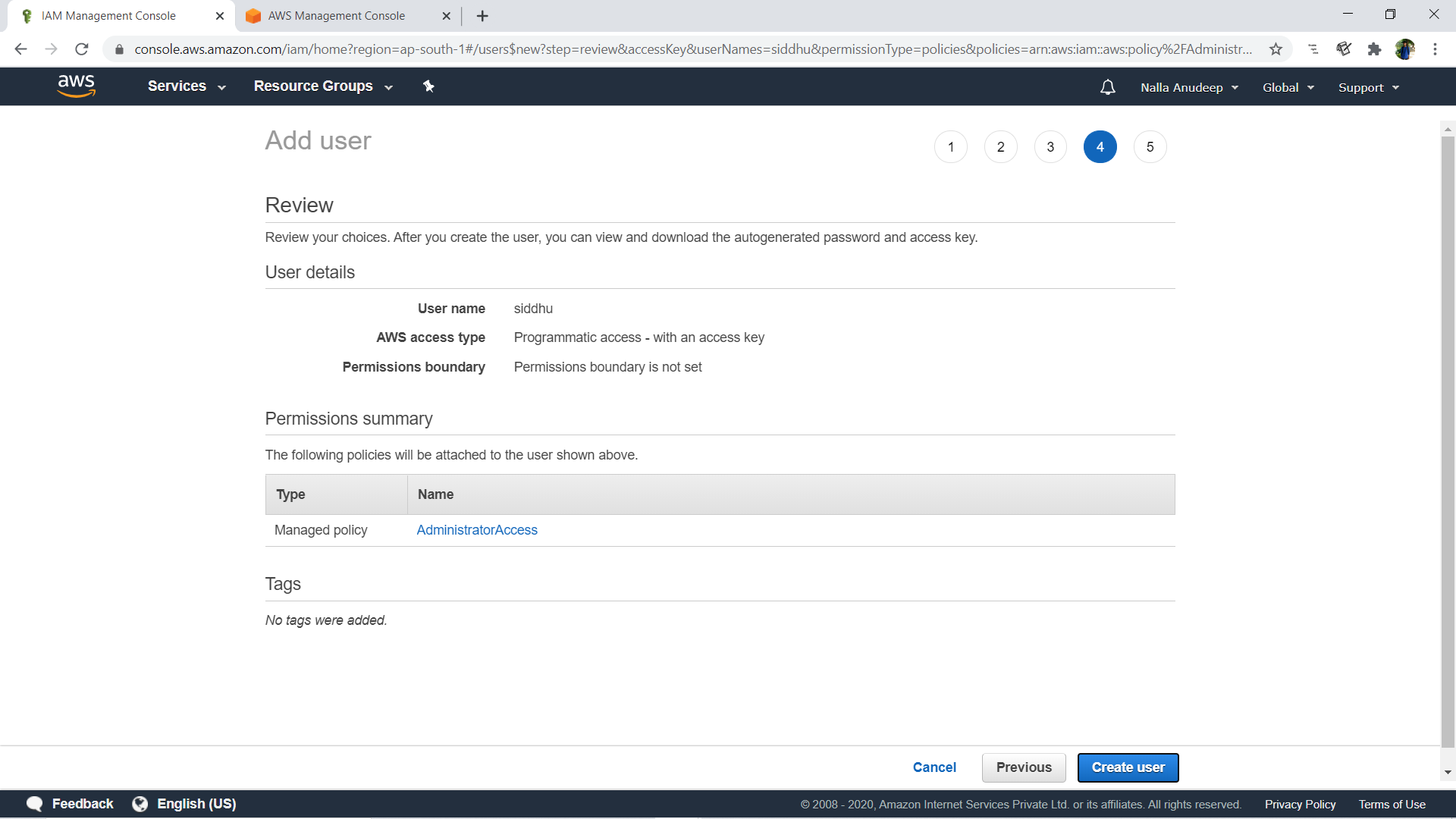
🡪Go to user 🡪 create user

🡪give username and access type with Programmatic access

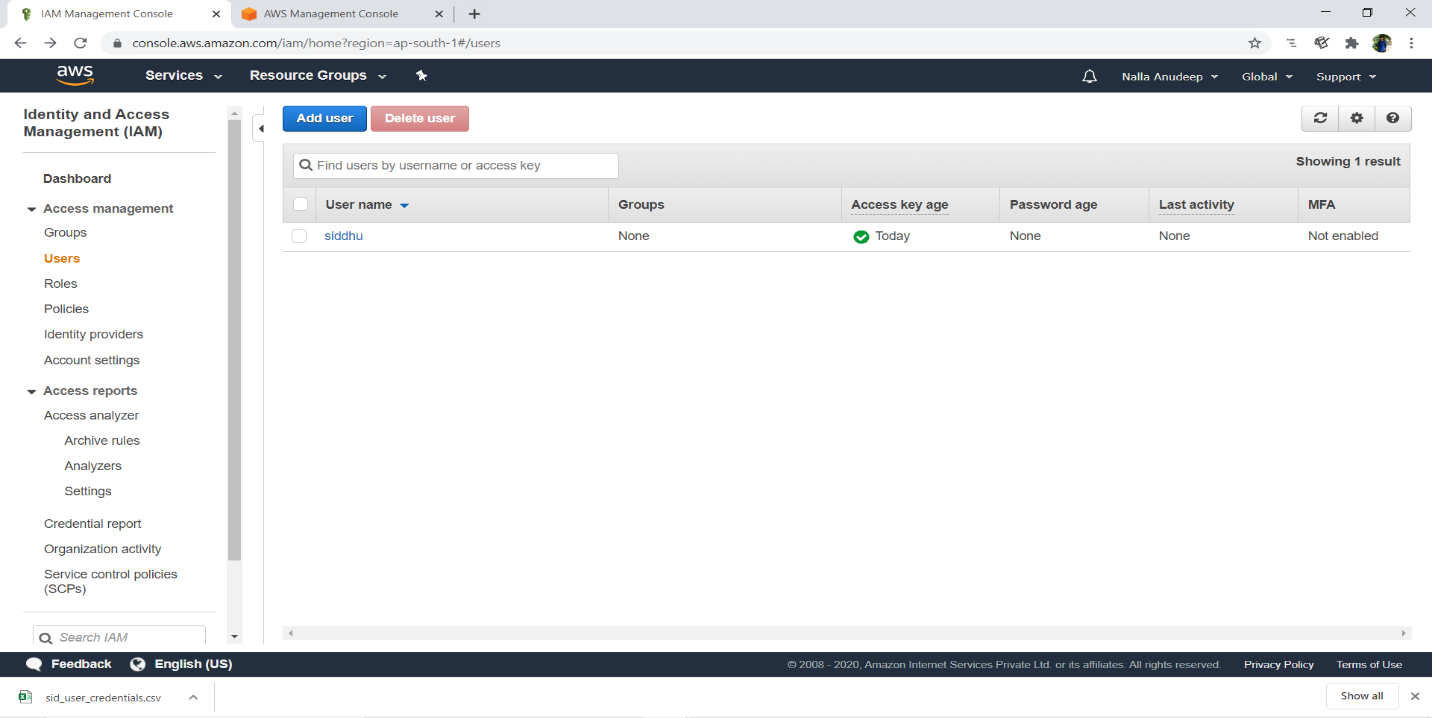


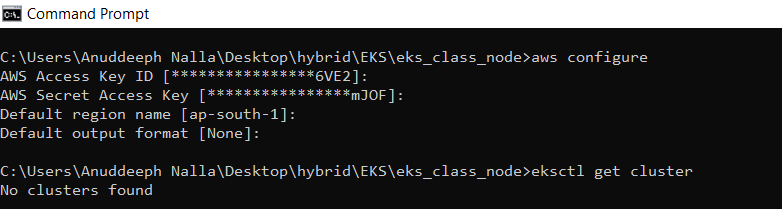
🡪Click on attach existing policy and give administration access to the user





* Finally user is created





Code to create the cluster

apiVersion: eksctl.io/v1alpha5

kind: ClusterConfig

metadata:

name: mycluster

region: ap-south-1

nodeGroups:

- name: ng1

desiredCapacity: 3

instanceType: t2.micro

ssh:

publicKeyName: mykey111222

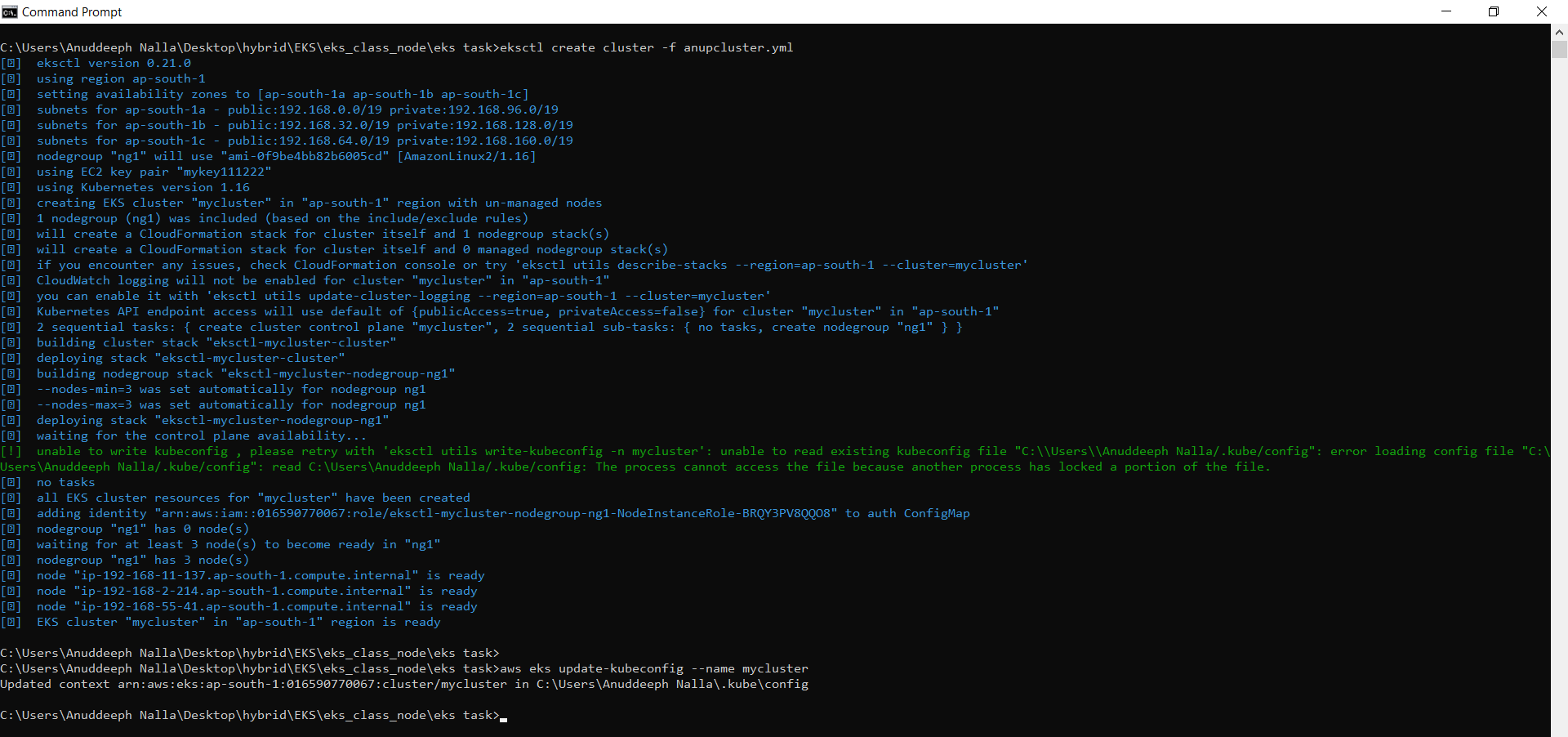
# you have to run following command for creating cluster

eksctl create cluster -f anupcluster.yml

# also you have to run following command for updating kubectl config file after

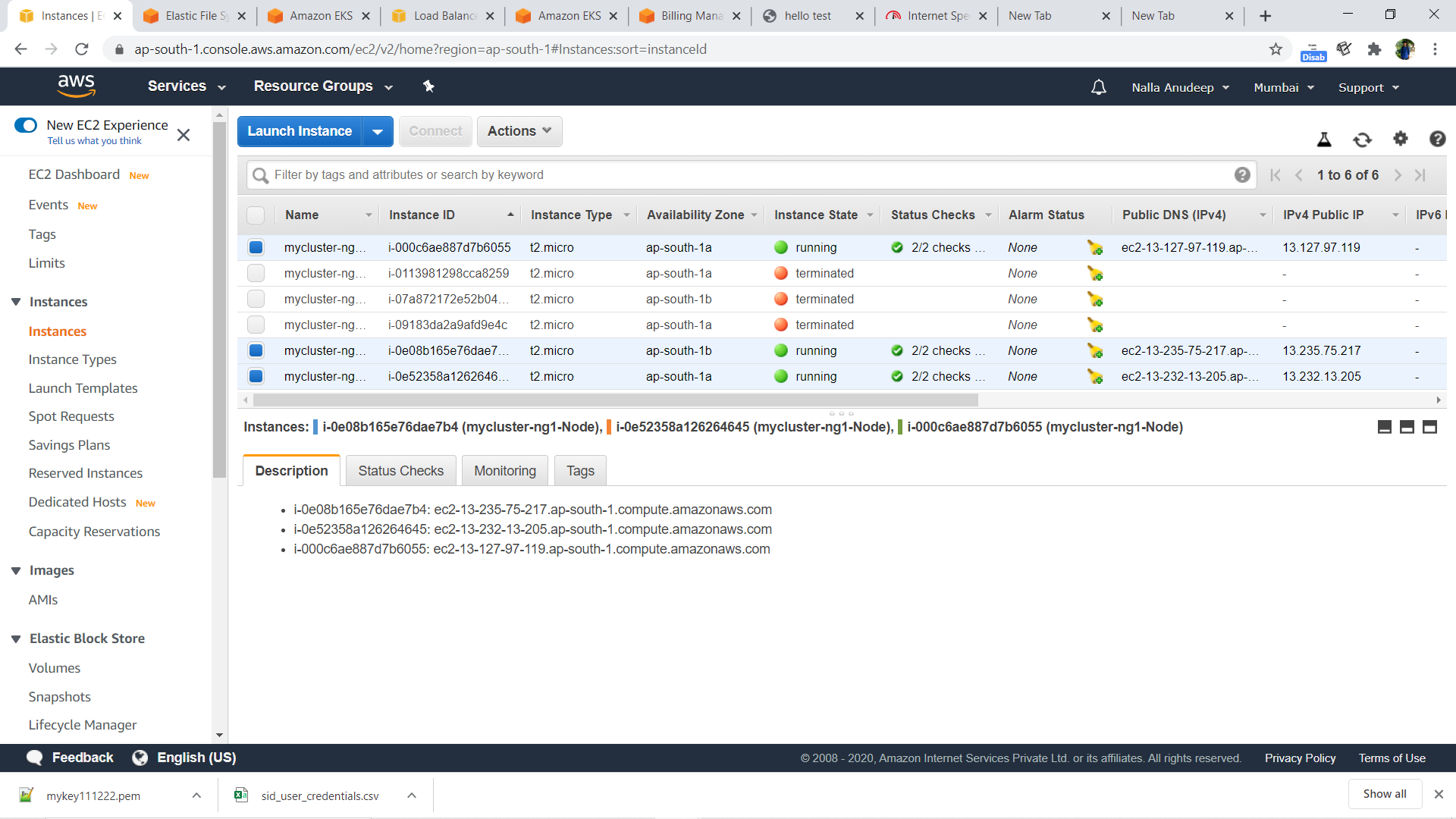
cluster created

aws eks update-kubeconfig --name cluster\_name(i.e mycluster)

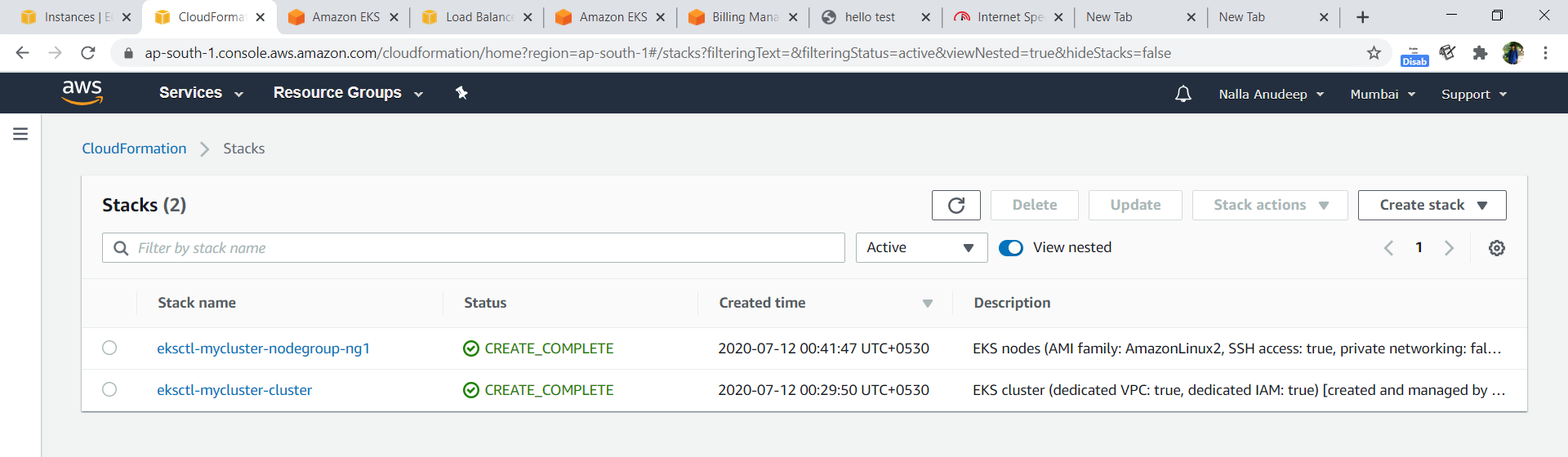


* On AWS portal

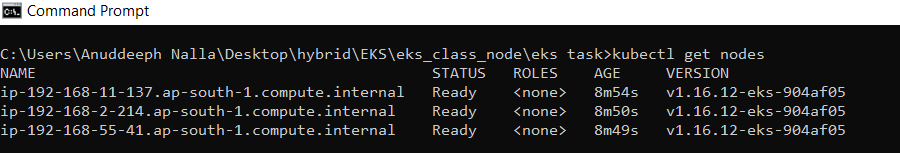
1. These are 3 slave nodes

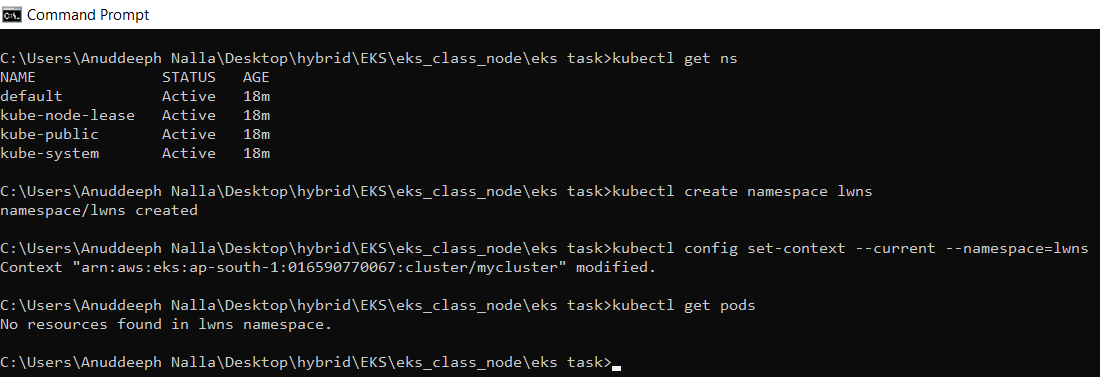


1. These are cloud formation stacks



View nodes using CLI





## Now we have create EFS for working as storage class for eks cluster.

1.First we have to install a software amazon-efs-utils in all the slave nodes created by using following command after login in each node.

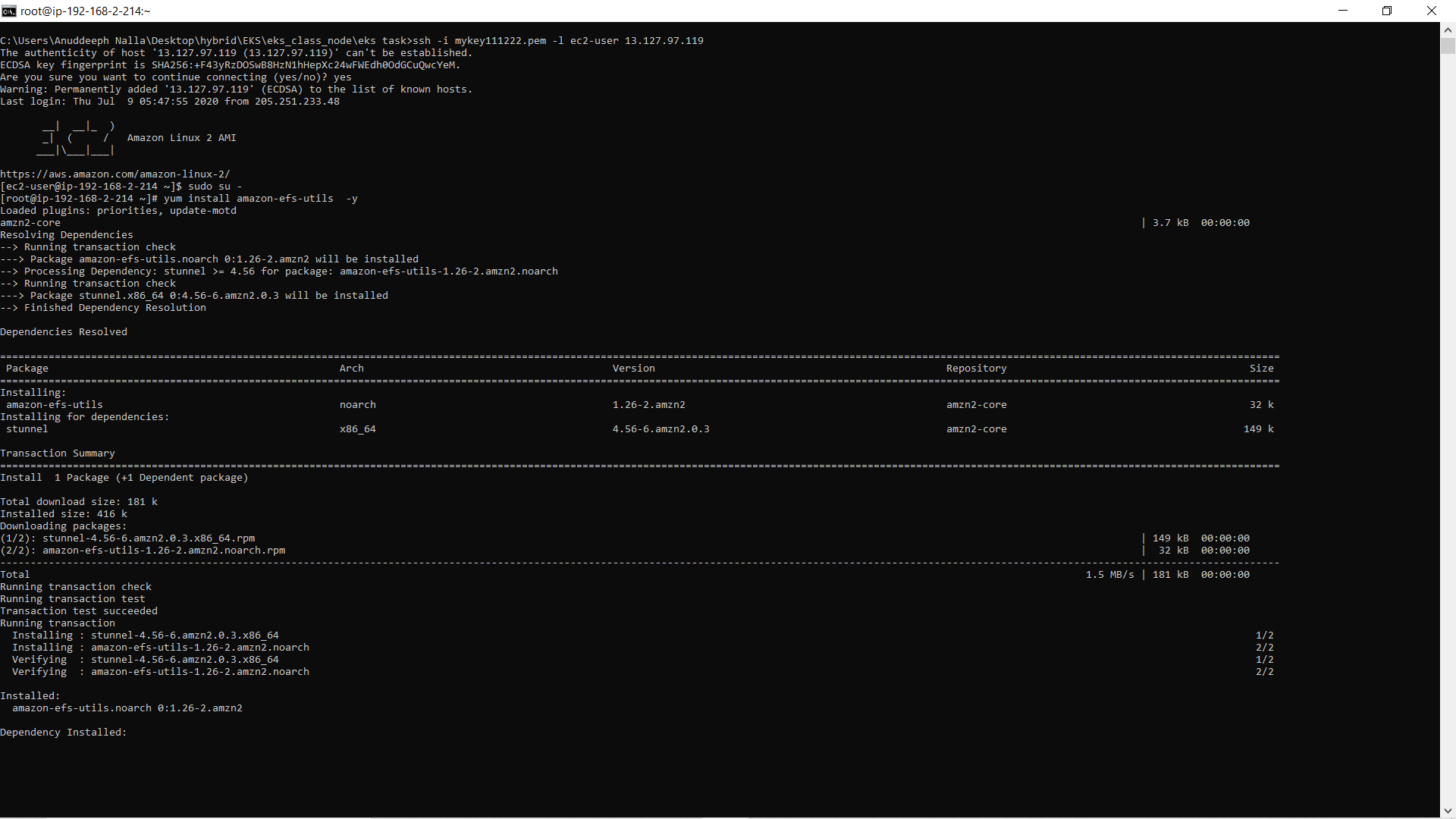
# first login with this command

ssh -i key\_file\_name.pem -l ec2-user IP\_of\_instnaces

#then run this command for installing software

sudo yum install amazon-efs-utils -y

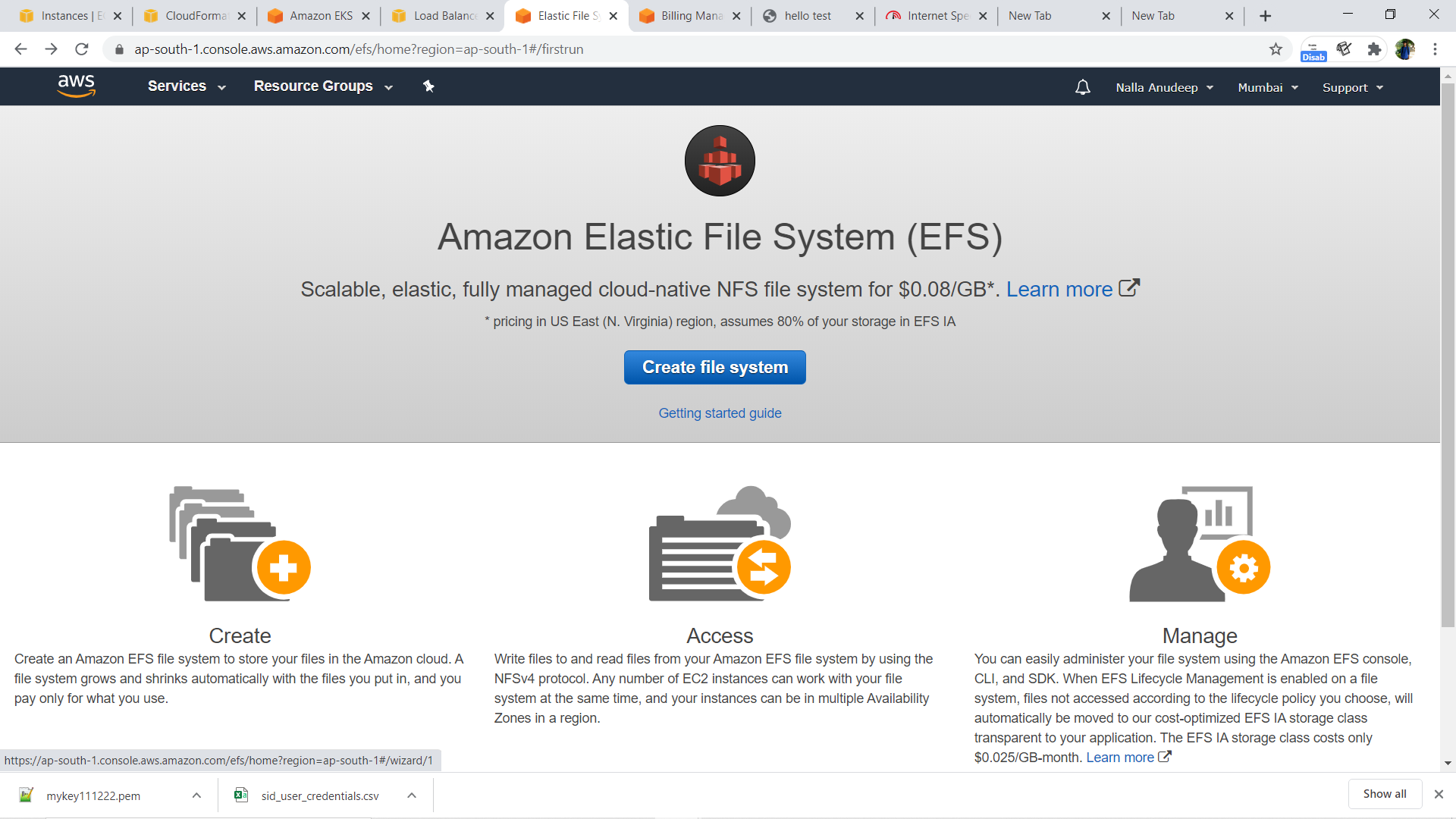
* You will get these screens in this process



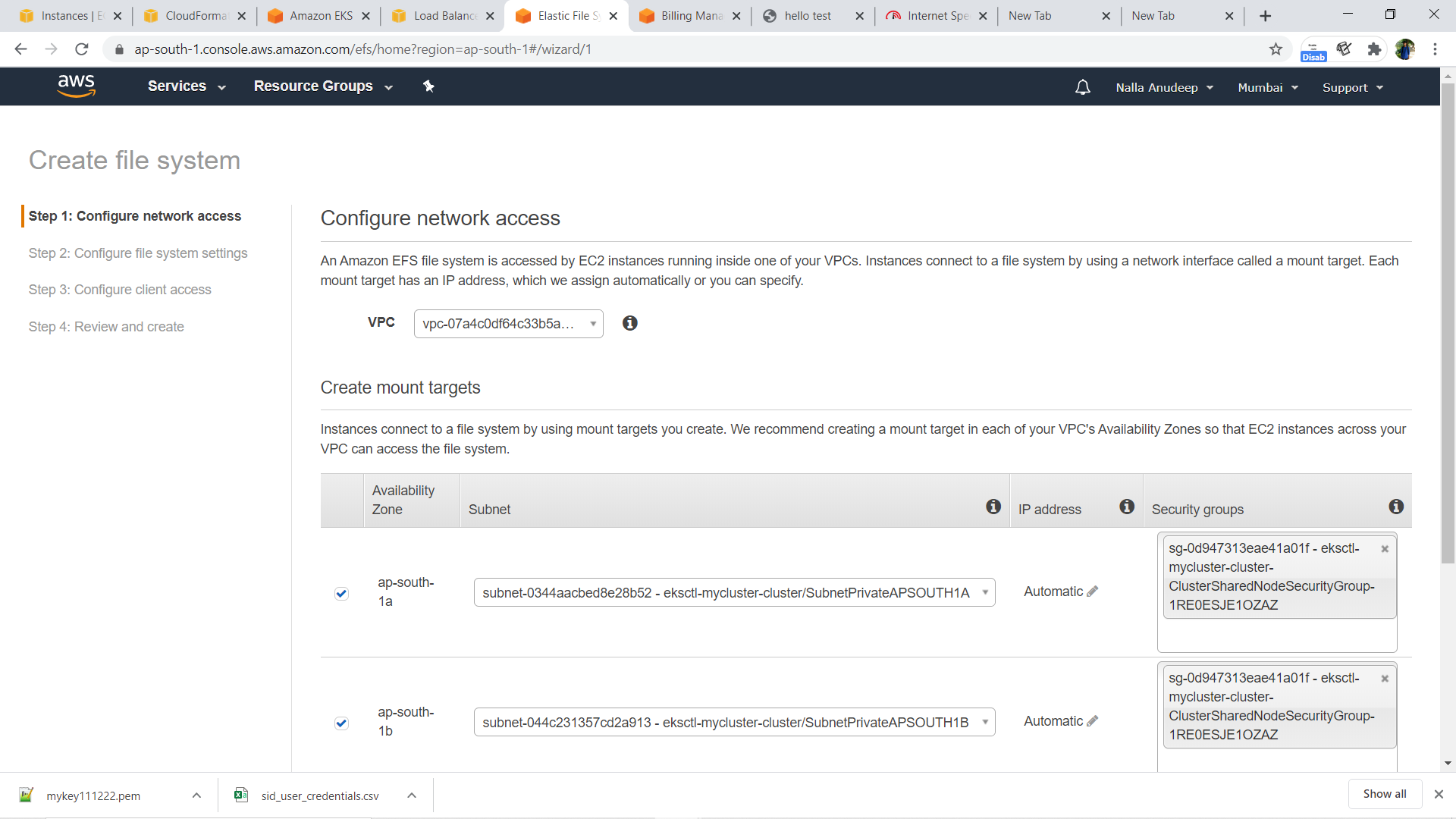
# install amazon-efs-utils in all the nodes

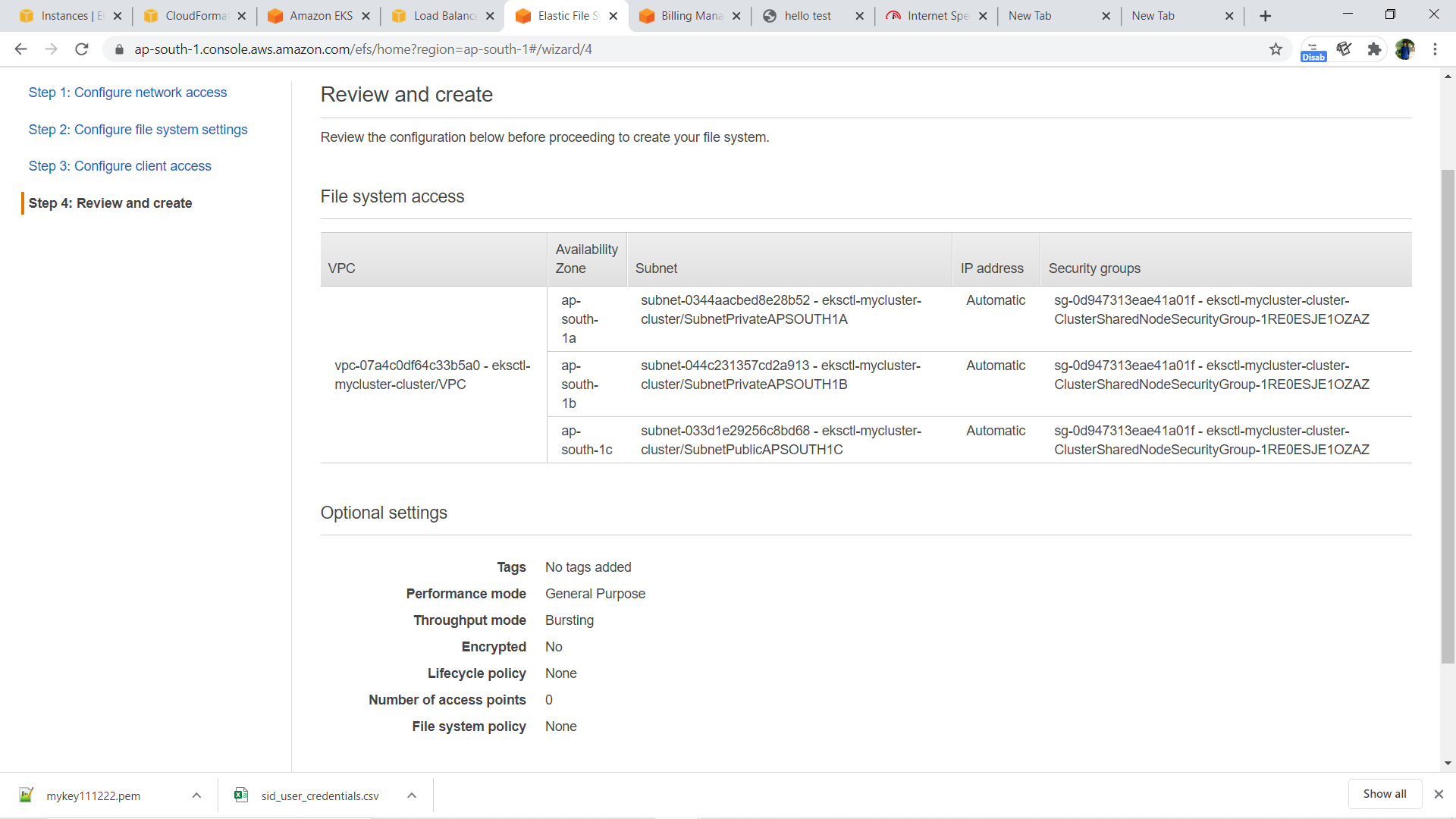
2. After this you must create efs storage, remember one thing at the time of creating efs you have to give same vpc and security groups as worker nodes have.

-Go to efs service and click create



* Change default vpc to eksctl vpc and remove the default security group add the security groups used in instances.





**Now its time to deploy our joomla on eks.**

1. this is efs provisioner file(create-efs-provisioner.yml) for provisioning the EFS. Only thing you must take care of that you have given right DNS and file\_id of efs
2. kind: Deployment
3. apiVersion: apps/v1
4. metadata:
5. name: efs-provisioner2
6. spec:
7. selector:
8. matchLabels:
9. app: efs-provisioner2
10. replicas: 1
11. strategy:
12. type: Recreate
13. template:
14. metadata:
15. labels:
16. app: efs-provisioner2
17. spec:
18. containers:
19. - name: efs-provisioner2
20. image: quay.io/external\_storage/efs-provisioner:v0.1.0
21. env:
22. - name: FILE\_SYSTEM\_ID
23. value: fs-0b78f2da
24. - name: AWS\_REGION
25. value: ap-south-1
26. - name: PROVISIONER\_NAME
27. value: lw-course/aws-efs2
28. volumeMounts:
29. - name: pv-volume
30. mountPath: /persistentvolumes
31. volumes:
32. - name: pv-volume
33. nfs:
34. server:  fs-0b78f2da.efs.ap-south-1.amazonaws.com
35. path: /

## 2.Now, we create Role-Based access control(RBAC). We are giving the role as cluster admin.

apiVersion: rbac.authorization.k8s.io/v1beta1

kind: ClusterRoleBinding

metadata:

  name: anup-nfs-provisioner-role-binding

subjects:

  - kind: ServiceAccount

    name: default

    namespace: lwns

roleRef:

  kind: ClusterRole

  name: cluster-admin

  apiGroup: rbac.authorization.k8s.io

## 3.Now We Create Stoarge for our EFS

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

  name: anup-aws-efs2

provisioner: lw-course/aws-efs2

---

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

  name: anup-efs-wix

  annotations:

    volume.beta.kubernetes.io/storage-class: "anup-aws-efs2"

spec:

  accessModes:

    - ReadWriteMany

  resources:

    requests:

      storage: 10Gi

---

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

  name: anup-efs-mysql

  annotations:

    volume.beta.kubernetes.io/storage-class: "anup-aws-efs2"

spec:

  accessModes:

    - ReadWriteMany

  resources:

    requests:

      storage: 10Gi

## 4.Now we Create Deployment File for our MySQL and joomla

apiVersion: v1

kind: Service

metadata:

  name: wix-mysql

  labels:

    app: wix

spec:

  ports:

    - port: 3306

  selector:

    app: wix

    tier: mysql

  clusterIP: None

---

apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2

kind: Deployment

metadata:

  name: wix-mysql

  labels:

    app: wix

spec:

  selector:

    matchLabels:

      app: wix

      tier: mysql

  strategy:

    type: Recreate

  template:

    metadata:

      labels:

         app: wix

         tier: mysql

    spec:

      containers:

      - image: mysql:5.7

        name: mysql

        env:

        - name: MYSQL\_ROOT\_PASSWORD

          valueFrom:

            secretKeyRef:

              name: mysql-pass

              key: password

        ports:

        - containerPort: 3306

          name: mysql

        volumeMounts:

        - name: mysql-persistent-storage

          mountPath: /var/lib/mysql

      volumes:

      - name: mysql-persistent-storage

        persistentVolumeClaim:

          claimName: anup-efs-mysql

## 5.Next Joomla Deployment

apiVersion: v1

kind: Service

metadata:

  name: wix

  labels:

    app: wix

spec:

  ports:

    - port: 80

  selector:

    app: wix

    tier: frontend

  type: LoadBalancer

---

apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2

kind: Deployment

metadata:

  name: wix

  labels:

    app: wix

spec:

  selector:

    matchLabels:

      app: wix

      tier: frontend

  strategy:

    type: Recreate

  template:

    metadata:

      labels:

          app: wix

          tier: frontend

    spec:

      containers:

      - image:  joomla:3.9.18-php7.4-apache

        name: wix

        env:

        - name: JOOMLA\_DB\_HOST

          value: wix-mysql

        - name: JOOMLA\_DB\_PASSWORD

          valueFrom:

            secretKeyRef:

              name: mysql-pass

              key: password

        ports:

        - containerPort:  80

          name: wix

        volumeMounts:

        - name: wix-persistent-storage

          mountPath: /var/www/html

      volumes:

      - name: wix-persistent-storage

        persistentVolumeClaim:

          claimName: anup-efs-wix

## 6. Now i have Created a kustomization file and add all files there

apiVersion: kustomize.config.k8s.io/v1beta1

kind: Kustomization

secretGenerator:

   - name:  mysql-pass

     literals:

          - password=redhat

resources:

       - anup-efs-provisioner.yaml

       - anup-create-rbac.yaml

       - anup-create-storage.yaml

       - anup-deploy-mysql.yml

       - anup-create-deploy-wix.yml

### -These are all those yml files which i have used in this project.

- anupcluster.yml

- anup-efs-provisioner.yaml

       - anup-create-rbac.yaml

       - anup-create-storage.yaml

       - anup-deploy-mysql.yml

       - anup-create-deploy-wix.yml

- Kustomization.yml

## See all the pods are created and with all svc and pvc

### 

## Finally joomla is launched

## 

## 

## After deleting the pods , it works

## 

## 