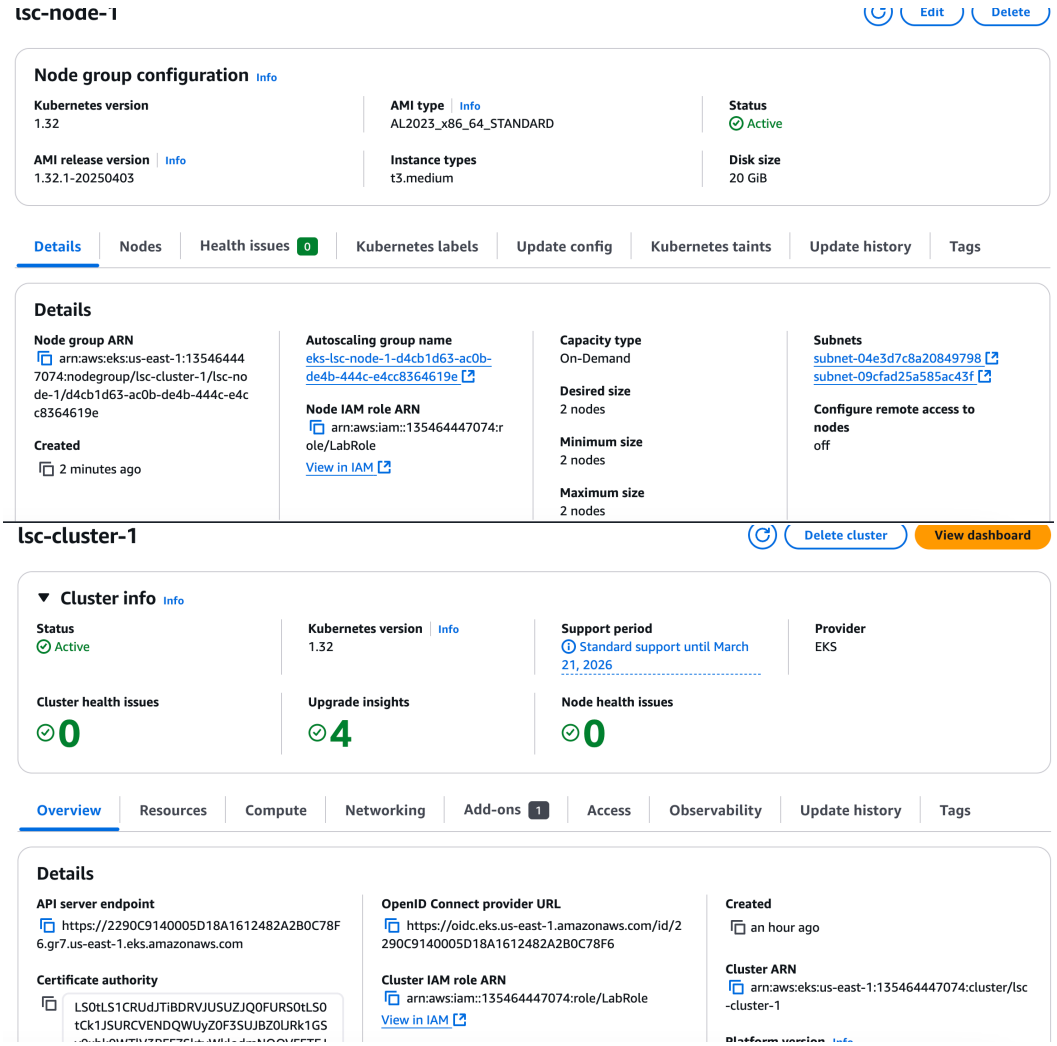
Lab 6 – Kubernetes

Assignments

Create a Kubernetes application (7p)

a) Create a k8s cluster using Amazon Elastic Kubernetes Service (EKS)

I created EKS cluster on AWS



b) Using Helm, install an NFS server and provisioner in the cluster.

```
agh_env) mikolajpajor@MacBookAir ~ % helm install nfs-server stable/nfs-server-provisioner --set storageClass.name=nfs --set persistence.enabled=false --set persistence.size=10Gi
ARNING: This chart is deprecated
AME: nfs-server
AST DEPLOYED: Tue Apr 15 20:51:27 2025
AMESPACE: default
TATUS: deployed
EVISION: 1
EST SUITE: None
OTES:
he NFS Provisioner service has now been installed.

 storage class named 'nfs' has now been created
nd is available to provision dynamic volumes.

ou can use this storageclass by creating a `PersistentVolumeClaim` with the
orrect storageClassName attribute. For example:

  ---
  kind: PersistentVolumeClaim
  apiVersion: v1
  metadata:
    name: test-dynamic-volume-claim
  spec:
    storageClassName: "nfs"
    accessModes:
      - ReadWriteOnce
    resources:
      requests:
        storage: 100Mi
```

c) Create a Persistent Volume Claim which will bind to a NFS Persistent Volume provisioned dynamically by the provisioner installed in the previous step.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc
spec:
  storageClassName: nfs  # Musi pasować do nazwy StorageClass
  accessModes:
    - ReadWriteMany      # NFS obsługuje RWX
  resources:
    requests:
      storage: 1Gi
~
~
Documents       Figure_3.png    Music           config          srv.ya
(agh_env) mikolajpajor@MacBookAir ~ % kubectl apply -f pvc.yaml
persistentvolumeclaim/nfs-pvc created
```

d) Create a Deployment with a HTTP server (e.g., apache or nginx). The web content directory should be mounted as a volume using the PVC created in the previous step.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        ports:
        - containerPort: 80
        volumeMounts:
        - name: nfs-volume
          mountPath: /usr/share/nginx/html
      volumes:
      - name: nfs-volume
        persistentVolumeClaim:
~
```

e) Create a Service associated with the Pod(s) of the HTTP server Deployment.

```
# nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: LoadBalancer
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
~
~
```

f) Create a Job which mounts the PVC and copies a sample content through the shared NFS PV.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: copy-job
spec:
  template:
    spec:
      containers:
      - name: busybox
        image: busybox
        command: ["/bin/sh", "-c", "echo '<h1>Hello from NFS!</h1>' > /mnt/index.html"]
        volumeMounts:
        - name: nfs-volume
          mountPath: /mnt
      volumes:
      - name: nfs-volume
        persistentVolumeClaim:
          claimName: nfs-pvc
      restartPolicy: Never
  backoffLimit: 2
~
~
```

g) Test the HTTP server by showing the sample web content in a browser.

Hello from NFS!

Files and script available on: https://github.com/Pejdzor/large_scale_computing_k8s_nginx