

# LSC LAB 6

## 1. Description of running the application

Github repository with application configuration:

<https://github.com/AdMrzyglod/lsc-lab6-kubernetes>.

A Kubernetes application was created in the task:

- A cluster was created and a group of nodes was added to it in EKS:

The screenshot displays the AWS Management Console interface for EKS clusters. At the top, there's a section for 'Clusters (1)' with a search bar and buttons for 'Delete', 'Create cluster', and a dropdown menu. Below this is a table with columns: Cluster name, Status, Kubernetes version, Support period, and Upgrade policy. The table contains one entry: 'lsc-lab-cluster' with status 'Active', version '1.32', and support until 'March 21, 2026'. Below the clusters section, there's a section for 'Node groups (1)' with buttons for 'Edit', 'Delete', and 'Add node group'. It includes a description: 'Node groups implement basic compute scaling through EC2 Auto Scaling groups.' and a table with columns: Group name, Desired size, AMI release version, Launch template, and Status. The table contains one entry: 'lsc-lab-nodegroup' with a desired size of '1', AMI version '1.32.1-20250419', and status 'Active'.

Cluster name	Status	Kubernetes version	Support period	Upgrade policy
lsc-lab-cluster	Active	1.32	Standard support until March 21, 2026	Extended

Group name	Desired size	AMI release version	Launch template	Status
lsc-lab-nodegroup	1	1.32.1-20250419	-	Active

- NFS Server Provisioner was installed using Helm:

```
adrlan@DESKTOP-8R2AAQR:~/lsc$ helm install nfs-server nfs-ganesha-server-and-external-provisioner/nfs-server-provisioner -f nfs-values-data.yaml
NAME: nfs-server
LAST DEPLOYED: Fri Apr 25 06:28:06 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The NFS Provisioner service has now been installed.

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

You can use this storageclass by creating a 'PersistentVolumeClaim' with the
correct storageClassName attribute. For example:

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

- A Persistent Volume Claim was created:

```
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl apply -f ./pvc-data.yaml
persistentvolumeclaim/lsc-data-pvc created
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
lsc-data-pvc	Bound	pvc-ad770526-4c0b-4ae6-9ce8-0af144846c56	1Gi	RWX	nfs-storage

- A deployment was created that creates a pod with an HTTP nginx server:

```
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl apply -f ./deployment-data.yaml
deployment.apps/lsc-nginx-deployment created
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl get deployment
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
lsc-nginx-deployment	1/1	1	1	10s

```
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
lsc-nginx-deployment-85d4d4db8c-s5nqz	1/1	Running	0	29s
nfs-server-nfs-server-provisioner-0	1/1	Running	0	16m

- A service associated with HTTP server Deployment pods was created:

```
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl apply -f ./service-data.yaml
service/lsc-data-service created
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl get service
```

NAME	AGE	TYPE
kubernetes		ClusterIP
lsc-data-service	43m	LoadBalancer
nfs-server-nfs-server-provisioner	7s	ClusterIP

- A job was created that mounts PVC and copies sample content via shared NFS PV:

```
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl apply -f ./job-data.yaml
job.batch/lsc-copy-job created
adrian@DESKTOP-0R2AAQR:~/lsc$ kubectl get job
```

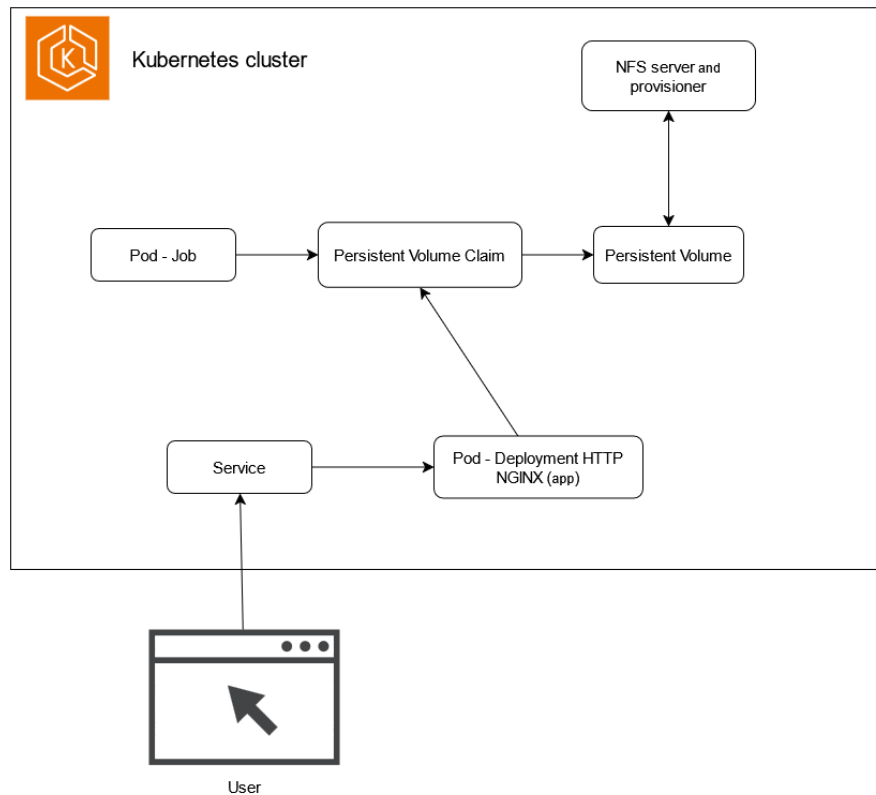
NAME	STATUS	COMPLETIONS	DURATION	AGE
lsc-copy-job	Complete	1/1	4s	11s

- The HTTP server was tested by displaying the content of the website in a browser:

← → ↻ Niezabezpieczone http://a0560445e6abe4635b7ec9d23d837a2f-939931976.us-east-1.elb.amazonaws.com

DATA HTTP SERVER

## 2. Architecture diagram of the created application



### Description:

- 1) Service - LoadBalancer type service, creates a single access point to HTTP applications.
- 2) Deployment HTTP - serves the content of the page, manages application pods, uses created PVCs.
- 3) Job - a task that executes only once, uses PVC, saves data that the server will make available.
- 4) Persistent Volume Claim - This is a request for a persistent volume by the Pod. PVC specifies requirements for disk space and access mode.
- 5) Persistent Volume - Represents a real, persistent storage resource in the cluster. PV is dynamically created by NFS Provisioner on the NFS server.

- 6) NFS Provisioner - This is a component that automatically creates Persistent Volume on request of Persistent Volume Claim, using the NFS server
- 7) NFS Server - This is a Network File System server that shares a remote file system. In a Kubernetes cluster, the NFS server (acting as a Pod) shares volumes to which other Pods can connect.