

Large Scale Computing - Lab 6 – Kubernetes

Report

Wiktor Satora 411502

```
> nano ~/.aws/credentials
> aws sts get-caller-identity
{
  "UserId": "ARO3AZQSWBU7BA0Z6MBP2:user3577538=Wiktor_Satora",
  "Account": "810714271038",
  "Arn": "arn:aws:sts::810714271038:assumed-role/voclabs/user3577538=Wiktor_Satora"
}
```

Cluster creation:

```
aws eks create-cluster --region us-east-1 --name lsc-cluster --role-arn
arn:aws:iam::810714271038:role/LabRole --resources-vpc-config subnetIds=subnet-
010b83e6853537b57,subnet-08066cc83ffba6d0a
```

```
{
  "cluster": {
    "name": "lsc-cluster",
    "arn": "arn:aws:eks:us-east-1:810714271038:cluster/lsc-cluster",
    "createdAt": "2025-04-10T18:58:55.572000+02:00",
    "version": "1.32",
    "roleArn": "arn:aws:iam:810714271038:role/LabRole",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-010b83e6853537b57",
        "subnet-08066cc83ffba6d0a"
      ],
      "securityGroupIds": [],
      "vpcId": "vpc-0476320ba27d88155",
      "endpointPublicAccess": true,
      "endpointPrivateAccess": false,
      "publicAccessCidrs": [
        "0.0.0.0/0"
      ]
    },
    "kubernetesNetworkConfig": {
      "serviceIpv4Cidr": "10.100.0.0/16",
      "ipFamily": "ipv4",
      "elasticLoadBalancing": {
        "enabled": false
      }
    },
    "logging": {
      "clusterLogging": [
        {
          "types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ],
          "enabled": false
        }
      ]
    }
  }
}
```

Amazon Elastic Kubernetes Service > Clusters > lsc-cluster

Amazon Elastic Kubernetes Service

Clusters

Settings

Console settings

Amazon EKS Anywhere

Enterprise Subscriptions

Related services

Amazon ECR

AWS Batch

Documentation

lsc-cluster

[Delete cluster](#) [View dashboard](#)

Cluster info

Status [Info](#)

Active

Cluster health issues

0

Kubernetes version [Info](#)

1.32

Upgrade insights

4

Support period [Info](#)

Standard support until March 21, 2026

Node health issues

0

Provider

EKS

[Overview](#) [Resources](#) [Compute](#) [Networking](#) [Add-ons](#) [Access](#) [Observability](#) [Update history](#) [Tags](#)

Details

API server endpoint

<https://7109965E6FFDCDF58BE4C41F75174082.gr7.us-east-1.eks.amazonaws.com>

OpenID Connect provider URL

<https://oidc.eks.us-east-1.amazonaws.com/id/7109965E6FFDCDF58BE4C41F75174082>

Created

31 minutes ago

Certificate authority

LS0LS1CRUdJTiB0RVVJU5UzQ0FUR5OLSOICK1JSURCVENDQWUyZD F3SU8ZDUWGH1UVR3eINDa2B3RFFZSKevWk0dmN0QVFfTEJRQ2Q GVEVUTUJFR0E5VUJkQWNSZ2ZvmlawEP1WbS6GN6QWVGd8B3T1

Cluster IAM role ARN

[arn:aws:iam:810714271038:role/LabRole](#) [View in IAM](#)

Cluster ARN

[arn:aws:eks:us-east-1:810714271038:cluster/lsc-cluster](#)

Platform version [Info](#)

eks.6

EKS Auto Mode [Info](#)

EKS automates routine cluster tasks for compute, storage, and networking to meet application compute needs.

EKS Auto Mode

Disabled

[Manage](#)

Nodes creation:

```
aws eks create-nodegroup --region us-east-1 --cluster-name lsc-cluster --nodegroup-name lsc-ng --node-role arn:aws:iam::810714271038:role/LabRole --subnet subnet-010b83e6853537b57 subnet-08066cc83ffba6d0a --instance-types t3.medium --scaling-config minSize=1,maxSize=2,desiredSize=1 --disk-size 20 --ami-type AL2_x86_64
```

```
{
  "nodegroup": {
    "nodegroupName": "lsc-ng",
    "nodegroupArn": "arn:aws:eks:us-east-1:810714271038:nodegroup/lsc-cluster/lsc-ng/a8cb1056-b652-6579-203d-db37a6ba7e2b",
    "clusterName": "lsc-cluster",
    "version": "1.32",
    "releaseVersion": "1.32.1-20250403",
    "createdAt": "2025-04-10T19:09:19.628000+02:00",
    "modifiedAt": "2025-04-10T19:09:19.628000+02:00",
    "status": "CREATING",
    "capacityType": "ON_DEMAND",
    "scalingConfig": {
      "minSize": 1,
      "maxSize": 2,
      "desiredSize": 1
    },
    "instanceTypes": [
      "t3.medium"
    ],
    "subnets": [
      "subnet-010b83e6853537b57",
      "subnet-08066cc83ffba6d0a"
    ],
    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::810714271038:role/LabRole",
    "diskSize": 20,
    "health": {
      "issues": []
    },
    "updateConfig": {
      "maxUnavailable": 1
    },
    "tags": {}
  }
}
```

Amazon Elastic Kubernetes Service > Clusters > lsc-cluster > lsc-ng

Amazon Elastic Kubernetes Service

Clusters

Settings

Console settings

Amazon EKS Anywhere

Enterprise Subscriptions

Related services

Amazon ECR

AWS Batch

Documentation

lsc-ng

Node group configuration

Kubernetes version	1.32	AMI type	AL2_x86_64	Status	Active
AMI release version	1.32.1-20250403	Instance types	t3.medium	Disk size	20 GiB

Details Nodes Health issues Kubernetes labels Update config Kubernetes taints Update history Tags

Details

Node group ARN arn:aws:eks:us-east-1:810714271038:nodegroup/lsc-cluster/lsc-ng/a8cb1056-b652-6579-203d-db37a6ba7e2b	Autoscaling group name eks-lsc-ng-a8cb1056-b652-6579-203d-db37a6ba7e2b	Capacity type On-Demand	Subnets subnet-010b83e6853537b57 subnet-08066cc83ffba6d0a
Created 3 minutes ago	Node IAM role ARN arn:aws:iam::810714271038:role/LabRole	Desired size 1 node	Configure remote access to nodes off
		Minimum size 1 node	
		Maximum size 2 nodes	

Node auto repair configuration

The node auto repair feature reacts to the Ready condition of the kubelet and any node object manual deletions. It can detect more node conditions for repair when the node monitoring agent is also installed. [Go to Add-ons](#)

Node auto repair

Kubectrl set-up:

```
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/darwin/arm64/kubectl
```

```
chmod +x kubectl
```

```
sudo mv kubectl /usr/local/bin/
```

Check cluster status and add new context for kubectl:

```
> aws eks describe-cluster --region us-east-1 --name lsc-cluster --query cluster.status
"ACTIVE"
> aws eks --region us-east-1 update-kubeconfig --name lsc-cluster
Added new context arn:aws:eks:us-east-1:810714271038:cluster/lsc-cluster to /Users/PCTR0126773/.kube/config
```

Added:

```
> kubectl config get-contexts
CURRENT NAME CLUSTER AUTHINFO
* arn:aws:eks:us-east-1:810714271038:cluster/lsc-cluster arn:aws:eks:us-east-1:810714271038:cluster/lsc-cluster arn:aws:eks:us-east-1:810714271038:cluster/lsc-cluster
```

Check if nodes are ready:

```
> kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-87-61.ec2.internal Ready <none> 12m v1.32.1-eks-5d632ec
```

Helm installation:

```
> curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 > get_helm.sh
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 11913 100 11913 0 0 72678 0 --:--:-- --:--:-- --:--:-- 72640
> chmod 700 get_helm.sh
> ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.17.2-darwin-arm64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
Password:
helm installed into /usr/local/bin/helm
```

NFS installation:

```
helm repo add nfs-ganesha-server-and-external-provisioner https://kubernetes-sigs.github.io/nfs-ganesha-server-and-external-provisioner/
> helm install nfs-server-provisioner nfs-ganesha-server-and-external-provisioner/nfs-server-provisioner -f config_files/nfs-config.yaml
NAME: nfs-server-provisioner
LAST DEPLOYED: Thu Apr 10 19:28:33 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The NFS Provisioner service has now been installed.

A storage class named 'nfs' 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"
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

Kubernetes setup repository: <https://github.com/wiksat/lsc-kubernetes>

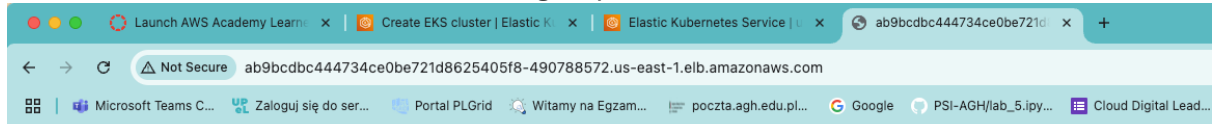
Setup files application:

```
> kubectl apply -f ./config_files/persistent_volume_claim.yaml
persistentvolumeclaim/lsc-pvc created
> kubectl apply -f ./config_files/deployment.yaml
deployment.apps/lsc-http-server created
> kubectl apply -f ./config_files/service.yaml
service/lsc-service created
> kubectl apply -f ./config_files/job.yaml
job.batch/lsc-job created
```

Check external-ip of http-service:

```
kubectl get service lsc-service
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
lsc-service  LoadBalancer  10.100.3.37      ab9bcdbc444734ce0be721d8625405f8-490788572.us-east-1.elb.amazonaws.com  80:31163/TCP  15m
```

Screenshot from web browser – running http-server:



Welcome to the HTTP server

Screenshot from final cluster workloads:

Workloads: Pods (7)

Pod is the smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster. [Learn more](#)

All Namespaces

Name	Created
aws-node-jcn8c	an hour ago
coredns-6b9575c64c-pxzzc	an hour ago
coredns-6b9575c64c-tkvc4	an hour ago
kube-proxy-ltc5v	an hour ago
lsc-http-server-769b7567cf-25cjq	22 minutes ago
lsc-job-c2fmk	9 minutes ago
nfs-server-provisioner-0	33 minutes ago

Architecture diagram:

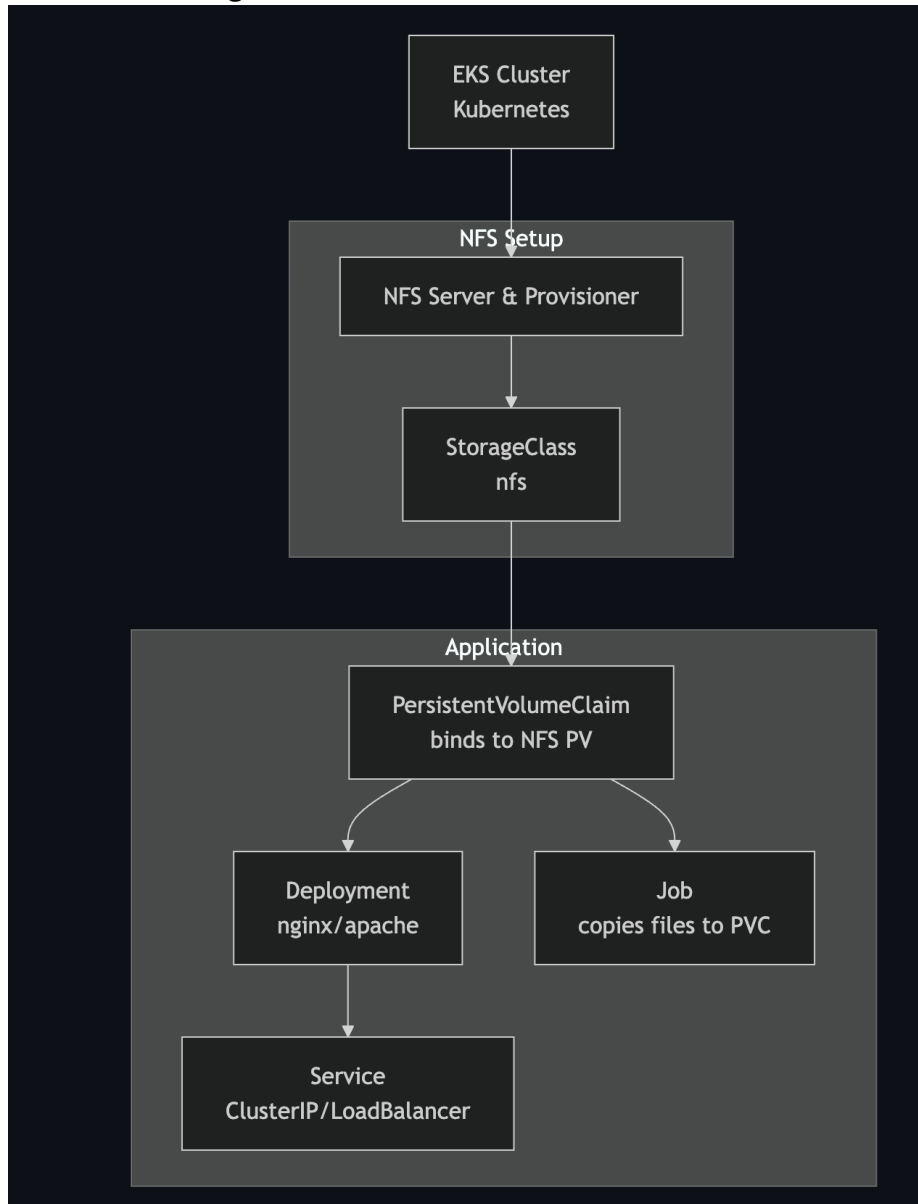


Diagram Explanation:

- **EKS Cluster:** Base Kubernetes environment.
- **NFS server & provisioner:** The NFS Server provides shared network storage for pods using the NFS protocol, while the NFS Provisioner dynamically creates PersistentVolumes (PVs)
- **StorageClass (nfs):** Defines dynamic provisioning parameters for NFS.
- **PersistentVolumeClaim:** Requests storage bound to NFS-backed persistent volume.
- **Deployment:** Runs HTTP server (nginx) with PVC-mounted web root.
- **Job:** Copies sample (index.html) file to the shared PVC (accessible by HTTP server).
- **Service:** Exposes the Deployment (ClusterIP for internal access or LoadBalancer for external).