Mateusz Waga
LSC-lab6

1) Link to repository: https://github.com/wlpinjlte/LSC-kubernetes
2) Short description of running the application:

```
Use "helm [command] --help" for more information about a command.
PS C:\Users\admin> helm version
version.BuildInfo{Version:"v3.17.3", GitCommit:"e4da49785aa6e6ee2b86efd5dd9e43400318262b", GitTreeState:"clean", GoVersi
on:"go1.23.7"}
PS C:\Users\admin> aws configure set default.region us-east-1
PS C:\Users\admin>
PS C:\Users\admin> aws configure set default.output table
PS C:\Users\admin>
PS C:\Users\admin> aws sts get-caller-identity
------------------------------------------------------------------
|                         GetCallerIdentity                       |
+----------+-----------------------------------------------------+
|  Account |  382266566128                                       |
|  Arn     |  arn:aws:sts::382266566128:assumed-role/voclabs/user3577778=Mateusz_Waga |
|  UserId  |  AROAVSA0A4XYX4X6JF3NX:user3577778/Mateusz_Waga     |
+----------+-----------------------------------------------------+

PS C:\Users\admin> aws eks describe-cluster --region us-east-1 --name lsc --query cluster.status
----------------
|DescribeCluster|
+--------------+

PS C:\Users\admin> aws eks --region us-east-1 update-kubeconfig --name lsc
Added new context arn:aws:eks:us-east-1:382266566128:cluster/lsc to C:\Users\admin\.kube\config
PS C:\Users\admin> kubectl get nodes
NAME                         STATUS   ROLES    AGE     VERSION
ip-172-31-23-51.ec2.internal   Ready    <none>   4m24s   v1.32.1-eks-5d632ec
ip-172-31-88-112.ec2.internal  Ready    <none>   4m50s   v1.32.1-eks-5d632ec
PS C:\Users\admin> kubectl get nodes
NAME                         STATUS   ROLES    AGE     VERSION
ip-172-31-23-51.ec2.internal   Ready    <none>   10m     v1.32.1-eks-5d632ec
ip-172-31-88-112.ec2.internal  Ready    <none>   10m     v1.32.1-eks-5d632ec
```

```
PS C:\Users\admin> helm install nfs-server-provisioner nfs-ganesha-server-and-external-provisioner/nfs-server-provisio
r  --set storageClass.name=nfs-storage  --set storageClass.defaultClass=true
NAME: nfs-server-provisioner
LAST DEPLOYED: Mon Apr 21 21:17:32 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
PS C:\Users\admin> kubectl create namespace lsc
namespace/lsc created
PS C:\Users\admin> kubectl apply -f pvc.yaml
error: the path "pvc.yaml" does not exist
PS C:\Users\admin> cd ../..
PS C:\> cd E:\projekty\LSC-kubernetes\yamls
PS E:\projekty\LSC-kubernetes\yamls> kubectl apply -f pvc.yaml -n lsc
persistentvolumeclaim/nfs-pvc created
PS E:\projekty\LSC-kubernetes\yamls> kubectl apply -f nginx-deployment.yaml -n lsc
deployment.apps/nginx-server created
PS E:\projekty\LSC-kubernetes\yamls> kubectl apply -f nginx-service.yaml -n lsc
service/nginx-service created
PS E:\projekty\LSC-kubernetes\yamls> kubectl apply -f content-job.yaml -n lsc
job.batch/content-upload-job created
```

Hello from NFS!

☰ Amazon Elastic Kubernetes Service 〉 Clusters 〉 lsc 〉 Add node group   ⓘ ⚙

○ Set compute and scaling configuration

Step 3
○ Specify networking

Step 4
○ Review and create

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

[node]

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console 🗗.

[LabRole   ▼]   (⟳)   (Create recommended role 🗗)

ⓘ The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more 🗗

Launch template Info

These properties cannot be changed after the node group is created.

⊙ Use launch template
Configure this node group using an EC2 launch template.

Kubernetes labels Info

This node group does not have any labels.

(Add label)

Remaining labels available to add: 50

Kubernetes taints Info

This node group does not have any taints.

(Add taint)

Remaining taints available to add: 50

☰ Amazon Elastic Kubernetes Service 〉 Create EKS cluster   ⓘ

Step 1
⦿ Configure cluster

Step 2
○ Specify networking

Step 3
○ Configure observability

Step 4
○ Select add-ons

Step 5
○ Configure selected add-ons settings

Step 6
○ Review and create

Configure cluster

Configuration options - *new* Info

Choose how you would like to configure the cluster.

○ **Quick configuration (with EKS Auto Mode) - new**
Quickly create a cluster with production-grade default settings. The configuration uses EKS Auto Mode to automate infrastructure tasks like creating nodes and provisioning storage.

⦿ **Custom configuration**
To change default settings prior to creation, choose this option. This configuration gives the option to use EKS Auto Mode and customize the cluster's configuration.

EKS Auto Mode - *new* Info

Choose if you would like to use EKS's Auto Mode.

⊙ Use EKS Auto Mode
EKS automates routine cluster tasks for compute, storage, and networking. When a new pod can't fit onto existing nodes, EKS creates a new node. EKS combines cluster infrastructure managed by AWS with integrated Kubernetes capabilities to meet application compute needs. View pricing 🗗

▶ Included capabilities

Cluster configuration Info

Name

Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

[lsc]

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.

Cluster IAM role | Info

Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. To create a new custom role, follow the instructions in the Amazon EKS User Guide 🗗.

[LabRole   ▼]   (⟳)   (Create recommended role 🗗)

Kubernetes version settings

Kubernetes version | Info

Select Kubernetes version for this cluster.

[1.32   ▼]

Upgrade policy | Info

Choose one of the following options. You can switch the setting later while the standard support period is in effect.

⦿ **Standard**
This option supports the Kubernetes version for 14 months after the release date. There is no additional cost. When standard support ends, your cluster will be auto upgraded to the next version.
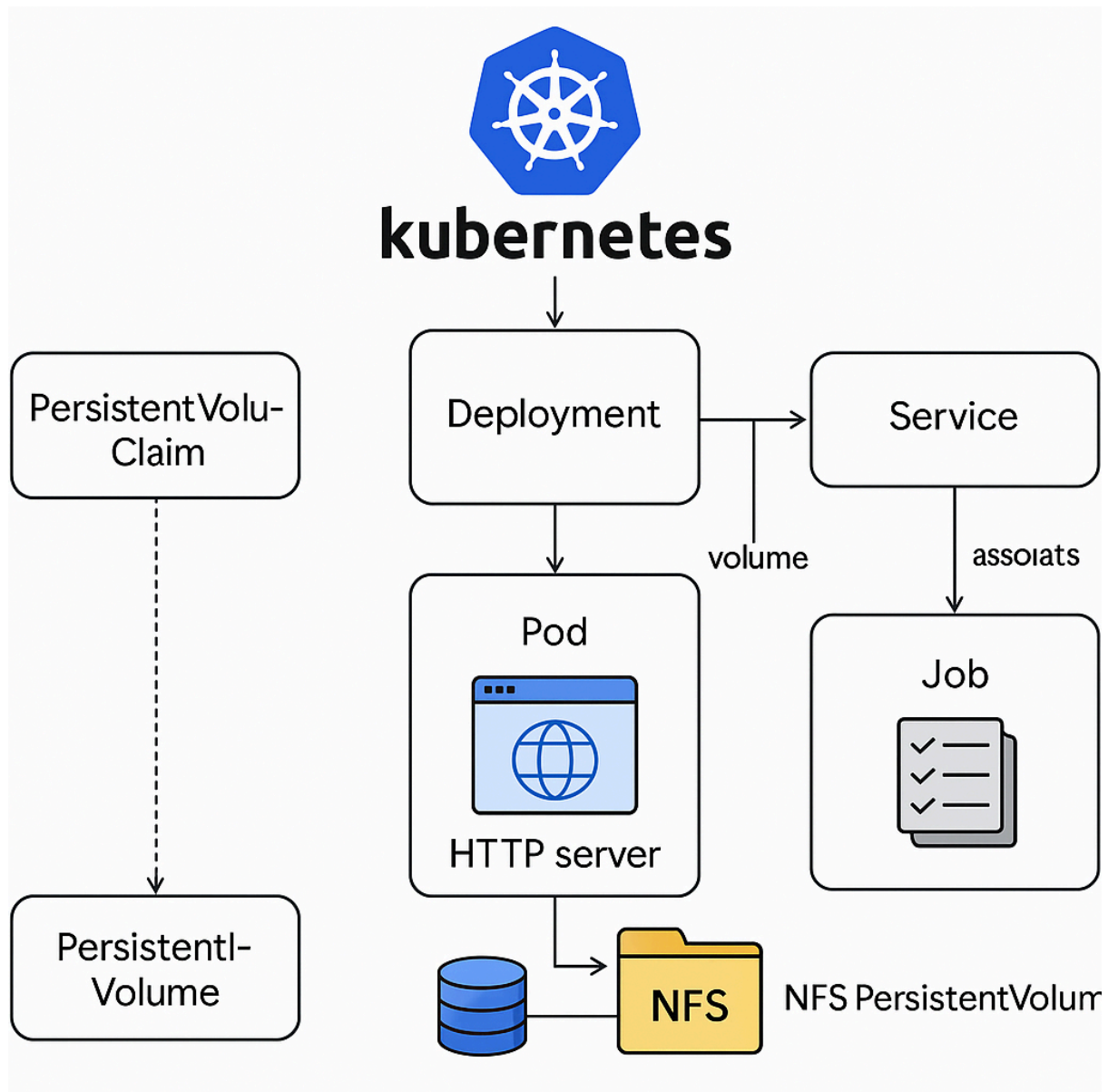
○ **Extended**
This option supports the Kubernetes version for 26 months after the release date. The extended support period has an additional hourly cost that begins after the standard support period ends. When extended support ends, your cluster will be auto upgraded to the next version.

Auto Mode Compute - *new* Info

Configure node management for your EKS cluster. EKS offers four compute options: EKS Auto Mode, EC2 Managed Node Groups, Fargate, and hybrid nodes. Node groups, Fargate profiles, and hybrid nodes are configured after cluster creation. You can also create self-managed nodes.

ⓘ **Compute configuration**   (View documentation 🗗)

3) Architecture diagram of the created application with a description explaining the role of the components and their connections:

kubernetes

PersistentVolu-Claim	Deployment	Service

Pod

HTTP server

volume

assoıats

Job

Persistentl-Volume

NFS

NFS PersistentVolum

This diagram illustrates a Kubernetes architecture integrating an NFS-based persistent storage solution with a web server application.

- A **PersistentVolumeClaim (PVC)** dynamically binds to an **NFS PersistentVolume** via a provisioner.

- A **Deployment** manages a **Pod** running an HTTP server (e.g., Nginx), which mounts the PVC to serve web content.

- A **Service** exposes the HTTP server Pod, making it accessible within or outside the cluster.

- A **Job** mounts the same PVC to upload or generate sample content into the shared NFS volume.

- The NFS volume ensures shared, persistent access to data between the Pod and the Job.