

Lab 6 – Kubernetes

Link to github repository: <https://github.com/strzebon/lsc-kubernetes>

Running application

Node group creation in progress
my-node is now being created. This process may take several minutes.

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

Configure node group [Info](#)
A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Step 1 **Configure node group**
Step 2 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.

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](#).

[Create recommended role](#)

Warning: 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](#)

```
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> aws configure set default.region us-east-1
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> aws configure set default.output table
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> aws sts get-caller-identity

-----
|                                     GetCallerIdentity                                     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Account | 120376140394 |
| Arn     | arn:aws:sts::120376140394:assumed-role/voclabs/user3577777=Witold_Strzebonski |
| UserId  | AROARYBX2EJVCJAXW4KPR:user3577777=Witold_Strzebonski |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

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

PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> aws eks --region us-east-1 update-kubeconfig --name my-cluster
Added new context ann:aws:eks:us-east-1:120376140394:cluster/my-cluster to C:\Users\WitoldStrzeboński\.kube\config
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-172-31-6-179.ec2.internal        Ready    <none>    8m18s v1.32.1-eks-5d632ec
ip-172-31-84-175.ec2.internal        Ready    <none>    8m21s v1.32.1-eks-5d632ec
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl --namespace kube-system get pods
NAME                                READY    STATUS    RESTARTS   AGE
aws-node-2xj8h                      2/2      Running   0           8m51s
aws-node-s7mrs                      2/2      Running   0           8m48s
coredns-6b9575c64c-kj6cg            1/1      Running   0           22m
coredns-6b9575c64c-t6pb6            1/1      Running   0           22m
eks-node-monitoring-agent-7gcmb      1/1      Running   0           8m51s
eks-node-monitoring-agent-xgll4      1/1      Running   0           8m48s
eks-pod-identity-agent-6cfqj         1/1      Running   0           8m48s
eks-pod-identity-agent-xl5tr         1/1      Running   0           8m51s
kube-proxy-2zs9z                    1/1      Running   0           8m51s
kube-proxy-wvxsv                     1/1      Running   0           8m48s
metrics-server-6d67d68f67-c5hkj      1/1      Running   0           19m
metrics-server-6d67d68f67-l26xn      1/1      Running   0           19m
```

```
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> helm repo add nfs-ganesha-server-and-external-provisioner https://kubernetes-sigs.github.io/nfs-ganesha-server-and-external-provisioner/
nfs-ganesha-server-and-external-provisioner has been added to your repositories
```

```
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> helm install nfs-server-provisioner nfs-ganesha-server-and-external-provisioner/nfs-server-provisioner
--set storageClass.name=nfs-storage --set storageClass.defaultClass=true
NAME: nfs-server-provisioner
LAST DEPLOYED: Mon Apr 21 21:17:24 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\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl apply -f pvc.yaml
persistentvolumeclaim/nfs-pvc created
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl apply -f deployment.yaml
deployment.apps/web-server created
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl apply -f service.yaml
service/web-service created
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl apply -f job.yaml
job.batch/copy-content created
```

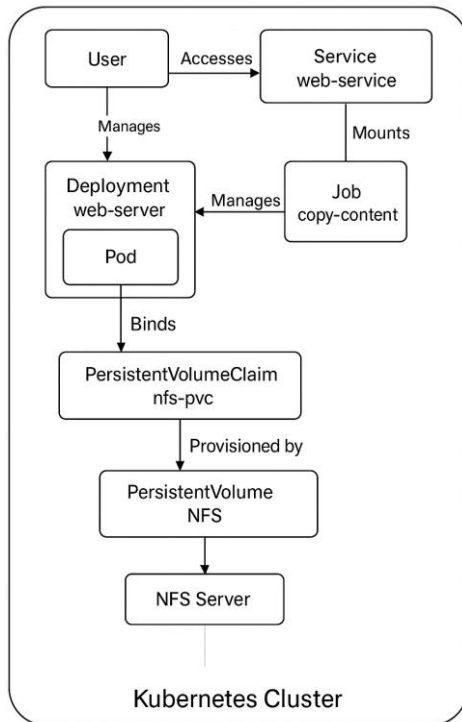
```
PS C:\Users\WitoldStrzeboński\Desktop\lsc-kubernetes> kubectl get svc web-service
NAME    TYPE        CLUSTER-IP    EXTERNAL-IP                                PORT(S)    AGE
web-service  LoadBalancer  10.100.112.52  a19b161d6341747f0967a56800742721-224454825.us-east-1.elb.amazonaws.com  80:30080/TCP  10m
```

← → ↺ a19b161d6341747f0967a56800742721-224454825.us-east-1.elb.amazonaws.com ☆ ⬇️ 🔔 ⋮

📺 | + CANAL+ TVP SPORT Netflix Prime Video Max sportsurge strims Flashscore basket office - YouT...

Hello from NFS!

Architecture diagram



User: Accesses the web app via a browser using the web-service.

Service (web-service): Exposes the HTTP server (nginx) externally via NodePort and routes traffic to pods.

Deployment (web-server): Manages nginx pods that serve content from a shared NFS volume.

Pod: Runs nginx and mounts the shared volume using the PVC.

Job (copy-content): Runs once to add sample HTML content to the shared volume.

PVC (nfs-pvc): Connects the pods and the job to a dynamically provisioned Persistent Volume.

Persistent Volume (PV): Created by the NFS provisioner to fulfill the PVC.

NFS Server: Provides the underlying shared storage for the application.