

# Zadanie 1

Na początku na klastrze "potyczki" tworzymy napespace "nginx" oraz uruchamiamy contener **nginx-0** wraz z obrazem **nginx:latest** dodatkowo zmieniamy liczbę replik na 3

The screenshot shows the Kubernetes UI interface. On the left, there's a sidebar with various navigation options like Cluster, Workloads, CronJobs, DaemonSets, Deployments (selected), Jobs, StatefulSets, Pods, Apps, Service Discovery, Storage, Policy, and More Resources. Below the sidebar are icons for Helm, Ingress, Storage, and other cluster tools. At the bottom left, it says 'About v2.8.2'. The main area has a title 'Potyczki młodych adminów '24' and a dropdown 'Only User Namespaces'. It shows a 'Namespace' dropdown set to 'nginx', a 'Name' input field with 'nginx', and a 'Description' text area. A 'Replicas' input field is set to '3'. Below this, tabs for 'Deployment', 'Pod', and 'nginx-0' (selected) are shown, along with a '+ Add Container' button. The 'General' tab is selected in the deployment configuration panel. It contains fields for 'Container Name' (set to 'nginx-0'), 'Image' (set to 'nginx:latest'), 'Pull Policy' (set to 'Always'), and 'Networking' settings. In the networking section, 'Service Type' is set to 'Cluster IP', 'Name' is 'nginx', 'Private Container Port' is '80', 'Protocol' is 'TCP', and there are buttons for 'Add Host' and 'Remove'. At the bottom right are 'Cancel', 'Edit as YAML', and 'Save' buttons.

Kolejnym krokiem jest zapewnienie dostępności usługi na internet. Wchodzimy w zakładkę "Service Discovery" oraz tam tworzymy "Ingress". W polu "Request Host" wpisujemy **nginx.193.187.69.140.sslip.io** potem "Target Service" ustawiamy na "nginx" oraz port na 80

Ingress: Create

Ingresses route incoming traffic from the internet to Services within the cluster based on the hostname and path specified in the request. You can expose multiple Services on the same external IP address and port.

Namespace: nginx Name: nginx Description: Any text you want that better describes this resource

Rules

Request Host: nginx.193.187.69.140.sslip.io

Path	Target Service	Port
Prefix /	nginx	80 Remove

Add Path Add Rule

Cancel Edit as YAML Create

Potyczki młodych adminów '24

Ingress: Create

Ingresses route incoming traffic from the internet to Services within the cluster based on the hostname and path specified in the request. You can expose multiple Services on the same external IP address and port.

Namespace: nginx Name: nginx Description: Any text you want that better describes this resource

Rules

**Default Backend**

Warning: Default backend is used globally for the entire cluster.

Target Service	Port
nginx	e.g. 80 or http

Cancel Edit as YAML Create

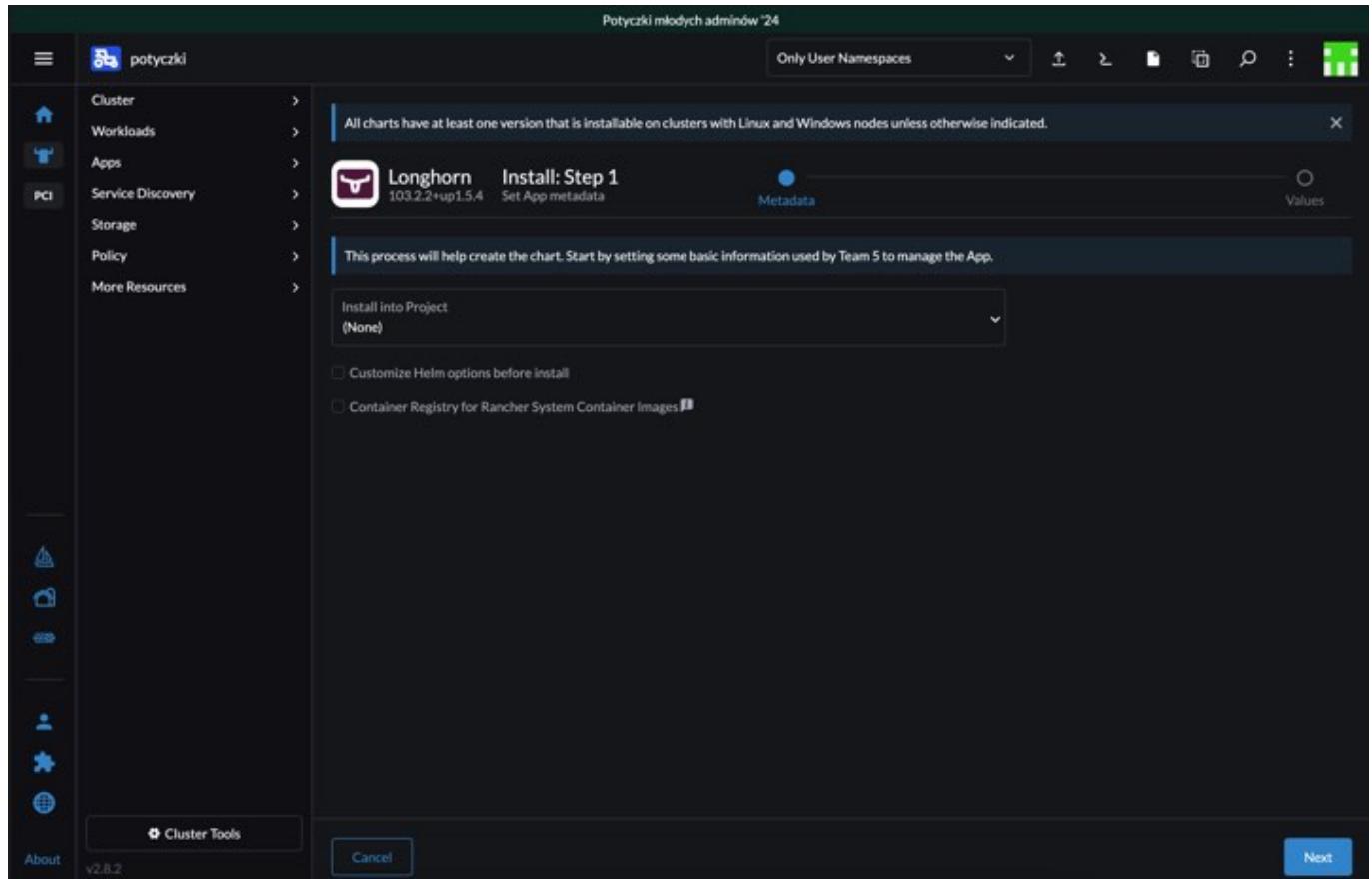
## Zadanie 2

Aby sprawdzić wymagania dla aplikacji "Longhorn" użyliśmy skryptu odpalamy skrypt za pomocą komendy `sudo bash ./enviorment_check.sh`. Wynik działania skryptu to:

```
[INFO] Please install missing dependencies: kubectl jq mktemp sort printf.
```

Doinstalowywujemy potrzebne komponenty za pomocą komend `sudo zypper in kubectl jq mktemp sort printf` oraz `sudo zypper in kubernetes1.20-client`

Kolejnym krokiem jest instalacja "Longhorna" w najnowszej stabilnej wersji na klastrze "potyczki"



Następnie konfigurujemy całą usługę (łącznie z ustawieniem 1 repliki)

All charts have at least one version that is installable on clusters with Linux and Windows nodes unless otherwise indicated. X

 longhorn    Update: Step 2  
103.2.2+up1.5.4    Change how the App works

Metadata    Values

Configure Values used by Helm that help define the App.

Edit Options Edit YAML Compare Changes    View Chart Info

**Longhorn Storage Class Settings**

Default Storage Class    Set as default StorageClass for Longhorn

Storage Class Retain Policy \*    Define reclaim policy (Retain or Delete)

Delete

Default Storage Class Replica Count    Set replica count for Longhorn StorageClass

1

Default Storage Class Data Locality    Set data locality for Longhorn StorageClass

disabled

Enable Storage Class Recurring Job Selector    Enable recurring job selector for Longhorn StorageClass

Enable Storage Class Node Selector    Enable Node selector for Longhorn StorageClass

Default Storage Class Backing Image    Set backing image for Longhorn StorageClass

Default Storage Class Remove Snapshots During Filesystem Trim    Allow automatically removing snapshots during filesystem trim for Longhorn StorageClass

ignored

Longhorn Images  
Private Registry Settings  
Longhorn Default Settings  
Longhorn V2 Data Engine (Preview Feature) Settings  
**Longhorn Storage Class Settings**  
Services and Load Balancing  
Other Settings

I na koniec potwierdzenie wszystkich podów longhorna

State	Name	Image	Ready	Restarts	IP	Node	Age
Namespace: longhorn-system							
<span>Running</span>	csi-attacher-5859bc989f-nw9zj	rancher/mirrored-longhorn-csi-attacher:v4.4.2	1/1	0	10.42.184.246	kubernetes05	1.9 mins
<span>Running</span>	csi-attacher-5859bc989f-pdpbf	rancher/mirrored-longhorn-csi-attacher:v4.4.2	1/1	0	10.42.184.243	kubernetes05	1.9 mins
<span>Running</span>	csi-attacher-5859bc989f-zzt5l	rancher/mirrored-longhorn-csi-attacher:v4.4.2	1/1	0	10.42.184.245	kubernetes05	1.9 mins
<span>Running</span>	csi-provisioner-c7ffcc5cc-72sdg	rancher/mirrored-longhorn-csi-provisioner:v3.6.2	1/1	0	10.42.184.244	kubernetes05	1.9 mins
<span>Running</span>	csi-provisioner-c7ffcc5cc-kq5n2	rancher/mirrored-longhorn-csi-provisioner:v3.6.2	1/1	0	10.42.184.250	kubernetes05	1.9 mins
<span>Running</span>	csi-provisioner-c7ffcc5cc-vhzb2	rancher/mirrored-longhorn-csi-provisioner:v3.6.2	1/1	0	10.42.184.247	kubernetes05	1.9 mins
<span>Running</span>	csi-resizer-777857494b-b9vp9	rancher/mirrored-longhorn-csi-resizer:v1.9.2	1/1	0	10.42.184.249	kubernetes05	1.9 mins
<span>Running</span>	csi-resizer-777857494b-jpdpf	rancher/mirrored-longhorn-csi-resizer:v1.9.2	1/1	0	10.42.184.251	kubernetes05	1.9 mins
<span>Running</span>	csi-resizer-777857494b-xrxpf	rancher/mirrored-longhorn-csi-resizer:v1.9.2	1/1	0	10.42.184.248	kubernetes05	1.9 mins
<span>Running</span>	csi-snapshotter-64cdff5bb4-b2pkn	rancher/mirrored-longhorn-csi-snapshotter:v6.3.2	1/1	0	10.42.184.254	kubernetes05	1.9 mins
<span>Running</span>	csi-snapshotter-64cdff5bb4-mqq87	rancher/mirrored-longhorn-csi-snapshotter:v6.3.2	1/1	0	10.42.184.252	kubernetes05	1.9 mins
<span>Running</span>	csi-snapshotter-64cdff5bb4-wqfkp	rancher/mirrored-longhorn-csi-snapshotter:v6.3.2	1/1	0	10.42.184.253	kubernetes05	1.9 mins
<span>Running</span>	engine-image-ei-933d45a5-rxvbw	rancher/mirrored-longhorn-longhorn-engine:v1.5.4	1/1	0	10.42.184.241	kubernetes05	2.4 mins
<span>Running</span>	instance-manager-beea3ff73e13cb5d6ecb06814aca7ce	rancher/mirrored-longhorn-longhorn-instance-manager:v1.5.4	1/1	0	10.42.184.240	kubernetes05	2.4 mins
7tjw/explorer/pod/longhorn-system/csi-resizer-777857494b-b9vp9		rancher/mirrored-longhorn-csi-node-driver-registrar:v2.9.2	3/3	0	10.42.184.255	kubernetes05	1.9 mins
<span>Running</span>	longhorn-driver-deployer-c77dd9cd6-f8hhm	rancher/mirrored-longhorn-longhorn-manager:v1.5.4	1/1	0	10.42.184.238	kubernetes05	3.2 mins
<span>Running</span>	longhorn-manager-b7zqr	rancher/mirrored-longhorn-longhorn-manager:v1.5.4	1/1	1 (2m9s ago)	10.42.184.236	kubernetes05	3.2 mins
<span>Running</span>	longhorn-ui-77ddf5bdf-lrb9l	rancher/mirrored-longhorn-longhorn-ui:v1.5.4	1/1	0	10.42.184.239	kubernetes05	3.2 mins
<span>Running</span>	longhorn-ui-77ddf5bdf-wwwg8	rancher/mirrored-longhorn-longhorn-ui:v1.5.4	1/1	0	10.42.184.237	kubernetes05	3.2 mins

## Zadanie 3

Tworzymy nowe repozytorium do katalogu aplikacji "Ranchera" o url

<https://rancher.github.io/rodeo>

**Repository: Create**

**Name:** rodeo

**Description:** Any text you want that better describes this resource

**Target**

- http(s) URL to an index generated by Helm
- Git repository containing Helm chart or cluster template definitions

**Index URL:** https://rancher.github.io/rodeo

**Authentication:** None

**Labels**

Key/value pairs that are attached to objects which specify identifying attributes.

**Annotations**

**Create**

Oraz intaluujemy aplikacje "Tetris"

**tetris** Install: Step 1

0.1.9 Set App metadata

Linux only

This process will help create the chart. Start by setting some basic information used by Team 5 to manage the App. To install the app into a new namespace enter it's name in the Namespace field and select it.

**Namespace:** default

**Name:** tetris

Customize Helm options before install

**Next**

Do tego potwierdzenie poprawnej instalacji oraz potwierdzenie działania wszystkich nodeów "Tetrisa"

```
Mon, Apr 8 2024 10:40:46 am
Mon, Apr 8 2024 10:40:46 am -----
Mon, Apr 8 2024 10:40:46 am SUCCESS: helm install --namespace=default --timeout=10m0s --values=/home/shell/helm/values-tetris-0.1.9.yaml --version=0.1.9 --
Mon, Apr 8 2024 10:40:46 am wait=true tetris /home/shell/helm/tetris-0.1.9.tgz
```

All charts have at least one version that is installable on clusters with Linux and Windows nodes unless otherwise indicated. X

 **NeuVector** **Install: Step 1** ○ Values

103.0.2+up2.7.3 Set App metadata Metadata

Linux only

This process will help create the chart. Start by setting some basic information used by Team 5 to manage the App.

Install into Project ▼

(None)

Customize Helm options before install

Container Registry for Rancher System Container Images 

Cancel Next

## Zadanie 4

---

Instalujemy aplikacje "Neu Vector"

All charts have at least one version that is installable on clusters with Linux and Windows nodes unless otherwise indicated. X

 NeuVector **Install: Step 1** 103.0.2+up2.7.3 Set App metadata Metadata  Values

**Linux only**

This process will help create the chart. Start by setting some basic information used by Team 5 to manage the App.

Install into Project  
**(None)** ▼

Customize Helm options before install

Container Registry for Rancher System Container Images ?

Cancel Next

Tutaj potwierdzenie instalacji oraz działania nodeów

Mon, Apr 8 2024 10:43:02 am	SUCCESS: helm upgrade --install=true --namespace=cattle-neuvector-system --timeout=10m0s --values=/home/shell/helm/values-neuvector-103.0.2-up2.7.3.yaml --version=103.0.2+up2.7.3 --wait=true neuvector /home/shell/helm/neuvector-103.0.2-up2.7.3.tgz								
<hr/>									
<input type="checkbox"/> State	Name	Namespace	Image	Ready	Up To Date	Available	Restarts	Age	Health
<input type="checkbox"/> Active	neuvector-controller-pod	cattle-neuvector-system	rancher/mirrored-neuvector-controller:5.3.0	3/3	3	3	0	2.6 mins	<div style="width: 100%; height: 10px; background-color: #2e7131; border-radius: 5px;"></div>
<input type="checkbox"/> Active	neuvector-manager-pod	cattle-neuvector-system	rancher/mirrored-neuvector-manager:5.3.0	1/1	1	1	0	2.6 mins	<div style="width: 100%; height: 10px; background-color: #2e7131; border-radius: 5px;"></div>
<input type="checkbox"/> Active	neuvector-scanner-pod	cattle-neuvector-system	rancher/mirrored-neuvector-scanner	3/3	3	3	0	2.6 mins	<div style="width: 100%; height: 10px; background-color: #2e7131; border-radius: 5px;"></div>

I ostatnim krokiem jest włączenie funkcji "Auto-Scan"

# Containers

Total: 45

Name	Namespace	Node	Applications	State	Scan Status	High	Medium	Scanned at
tigera-operator-767d6f	tigera-operator	kubernetes05		Discover		0	0	
tigera-operator	tigera-operator	kubernetes05		Discover		0	0	
tetris-7f447847d-mw8i	default	kubernetes05	Apache, HTTP	Discover		0	0	
tetris	default	kubernetes05	TCP/80	Discover		0	0	
system-upgrade-contro	cattle-system	kubernetes05		Discover		0	0	
system-upgrade-control	cattle-system	kubernetes05		Discover		0	0	
rke2-snapshot-validat	kube-system	kubernetes05	SSL	Discover		0	0	
rke2-snapshot-validator	kube-system	kubernetes05	TCP/8443	Discover		0	0	
rke2-snapshot-controll	kube-system	kubernetes05	TCP/8080	Discover		0	0	
rke2-snapshot-controller	kube-system	kubernetes05	TCP/8080	Discover		0	0	

Container Details

- ID: 94f6b590bc00d9aaaade3009271532151ccc...
- Service: tigera-operator.tigera-operator
- Service Account: tigera-operator
- Network Mode: host
- Image: docker.io/rancher/pause:3.6
- Image Created At: Aug 25, 2021 22:58:29
- Privileged: false
- Enforcer: neuvvector-enforcer-pod-f54vx

Network Interfaces

IP	MAC	Container
10.0.0.10/24		

## Zadanie 5

Skanujemy sekcje "nvbeta/\*" w rejestrze za pomocą Neu Vector

Edit registry

Test Setting

Registry type

Docker Registry

Name:  
docker-nvbeta

Registry \*:  
https://registry.hub.docker.com/

Sample: https://registry.hub.docker.com/

Password:

.....

User Name:



Filter

nvbeta/\*  Filter...

Sample: gliderlabs/dockerbox:latest, gliderlabs/\*, gliderlabs/\*\*, gliderlabs/n\*:3\*

Rescan after CVE DB  
update

Scan Layers

Periodic  
Scan

Cancel

Submit

Image z największą liczbą podatności to <https://registry.hub.docker.com/nvbeta/node:latest>

# Zadanie 6

Dodalismy Persistent Volume korzystajacy ze sotrage "Longhorna". W pliku .yaml tego deploymentu zmienilismy "storage class" na longhorna aby korzystal z tego Persistent Volume. W usłudze "Longhorn" trzeba dodać nowy volume oraz dodac go do hosta "kubernetes05" Przykładowy kod yaml jest wysłany do **nie-usuwaj.yaml**

Konfiguracja Persistent Volume

**PersistentVolume: Create**

Name* ubuntu-pv	Description Any text you want that better describes this resource
Volume Plugin* Longhorn	Capacity* 5 GiB

**Plugin configuration**

**Customize**

**Plugin configuration**

Filesystem Type ext4	Volume Handle* ubuntu-data
-------------------------	-------------------------------

**Read Only**  
 Yes  No

**Options**

Key	Value	Action
size	2Gi	Remove
numberOfReplicas	3	Remove
staleReplicaTimeout	20	Remove
fromBackup	e.g. bar	Remove

**Add**

**Buttons:** Cancel, Edit as YAML, Create

### PersistentVolume: Create

Name\* **ubuntu-pv**

Description Any text you want that better describes this resource

Volume Plugin\* **Longhorn**

Capacity\* **5** GiB

Plugin configuration

[Customize](#)

**Customize**

**Access Modes**

Single Node Read-Write  
 Many Nodes Read-Only  
 Many Nodes Read-Write

**Mount Options**

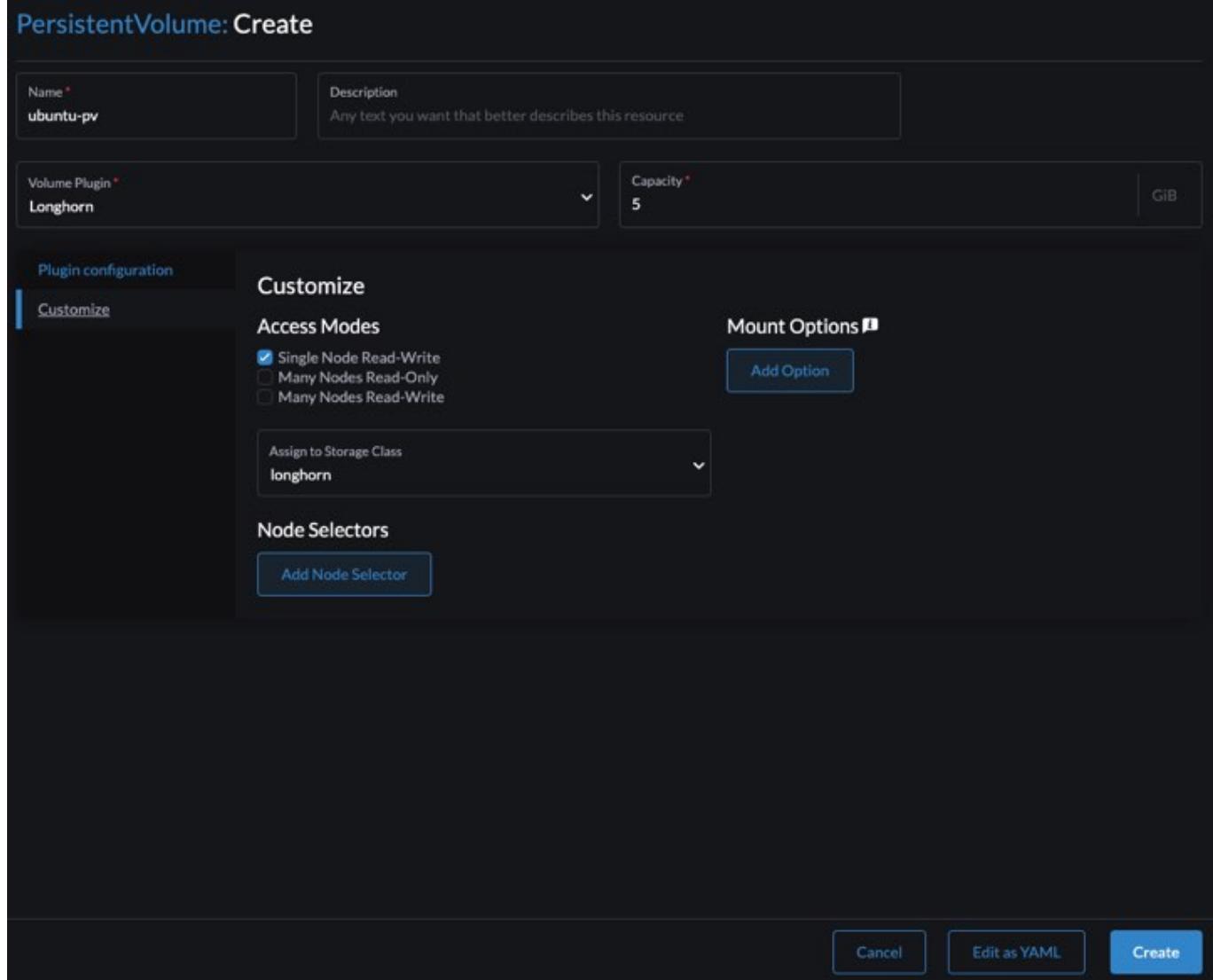
[Add Option](#)

Assign to Storage Class  
**longhorn**

**Node Selectors**

[Add Node Selector](#)

[Cancel](#) [Edit as YAML](#) [Create](#)



Tutaj przykładowy kod yaml z pliku **nie-usuwaj.yaml**

The screenshot shows a 'Import YAML' dialog box. On the left, there is a 'Read from File' button. On the right, a dropdown menu shows 'Default Namespace' and 'default'. The main area contains the following YAML code:

```
1 ---  
2 kind: PersistentVolumeClaim  
3 apiVersion: v1  
4 metadata:  
5   name: ubuntu-pvc  
6 spec:  
7   accessModes:  
8     - ReadWriteOnce  
9   storageClassName: longhorn  
10  resources:  
11    requests:  
12      storage: 1Gi  
13 ---  
14 apiVersion: apps/v1  
15 kind: Deployment  
16 metadata:  
17   labels:  
18     app: ubuntu  
19   name: ubuntu  
20 spec:  
21   replicas: 1  
22   selector:  
23     matchLabels:  
24       app: ubuntu  
25   template:  
26     metadata:  
27       labels:  
28         app: ubuntu  
29     spec:  
30       volumes:  
31         - name: test-vol
```

At the bottom are 'Cancel' and 'Import' buttons.

Następnie dodajemy nowy volume w Longhorn jak wyżej powiedziane

### Create Volume

\* Name: ubuntu-data ✓

\* Size: 5 Gi ✓

\* Number of Replicas: 3 ✓

\* Frontend: Block Device ✓

Data Locality: disabled ✓

Access Mode: ReadWriteOnce ✓

Backing Image: ✓

Backend Data Engine: v1 ✓

Encrypted:

Node Tag: ✓

Disk Tag: ✓

> Advanced Configurations

Cancel OK

Oraz przypisujemy go do hosta i zmieniamy "Update Replicas Count" na 1

### Attach to host

\* Host: kubernetes05 ✓

Maintenance:

Cancel OK

### Update Replicas Count

\* Number Of Replicas:  ✓

Cancel OK

Oraz dodatkowe potwierdzenie że deployment pliku [nie-usuwaj.yaml](#)

State	Name	Size	Actual Size	Created	Data Engine	PV Name	Namespace	Operation
Healthy	ubuntu-data	5 Gi	147 Mi	5 minutes ago	v1	ubuntu-pv	default	

W deployment zmieniam repliki na 0. Storage w "Longhorn" cały czas zostaje tak samo jak Persistant Volume i Persistant Volume Claim.

Deployment: ubuntu Active

Namespace: default Age: 12 mins Pod Restarts: 0

Namespace:  Name:  Description: Any text you want that better describes this resource

Replicas:

Po zmianie deploymentu na 1 replike pod podlaczy sie do poprzedniego Persistent Volume i dane zostana zachowane

## Zadanie 7

Pierwsza metoda: znalezienie resource kliknietie przy nim menu i wybranie opcji "download yaml"

Druga metoda: Z poziomu kubectl mozemy uzyć komendy `kubectl get deploy ubuntu -o yaml` i zmienic typ wyjscia na yaml. Dla przykladu pobranie konfiguracji yaml deploymentu

## Zadanie 7.1

Uruchomienie kontenera MySQL na najprostszych domyslnych ustawieniach moze byc wygodne w poczatkowych fazach rozwoju aplikacji, jednak nie jest to zalecany sposob w produkcji. Domyslne ustawienia moga byc niewystarczajace pod wzgledem wydajnosci, bezpieczenstwa i skalowalnosci. Zaleca sie dostosowanie konfiguracji MySQL do konkretnych potrzeb aplikacji, w tym uwzglednienie optymalizacji

wydajności, zabezpieczeń danych i skalowalności, aby zapewnić stabilność i efektywność działania systemu w dłuższej perspektywie.

Sugestie poprawy: Zmiana deployment na statefull set, przeniesienie secretów do Kubernetes Secret, dodanie persistent volume aby dane były zachowywane, dodanie serwisu dla stabilnej sieci, zmiana wersji z latest na wybraną wersję ze względu na to że w nowych wersjach mogą występuwać zmiany krytyczne które zepsują całą aplikację. Proponowane zmiany wrzucamy jako commit do baza.yaml

Sugestie poprawy:

- zmiana deployment na statefull set(stabilny i unikalny identyfikator sieciowy, uporządkowany deployment i skalowanie uporządkowane "rolling update")
- przeniesienie secretów do "Kubernetes Secret"
- dodanie Persistent Volume aby dane były zachowane
- dodanie serwisu dla stabilnej sieci
- zmiana wersji z latest na wybraną wersję ze względu na to że w nowych wersjach mogą występuwać zmiany krytyczne które mogą zepsuć całą aplikację

Proponowane zmiany przesyłamy jako commit do [baza-secure.yaml](#)

## Zadanie 8

---

Resource o nazwie Gateway w Kubernetes to obiekt, który umożliwia zarządzanie ruchem sieciowym w klastrze Kubernetes. Działa jako punkt wejścia dla zewnętrznego ruchu do aplikacji działających w klastrze, umożliwiając wykonywanie różnych operacji routingu, takich jak przekierowywanie ruchu na konkretne usługi w klastrze na podstawie określonych warunków. Gateway jest używany do definiowania zewnętrznych punktów dostępu do aplikacji w klastrze Kubernetes, zapewniając elastyczność i kontrolę nad ruchem sieciowym.

## Zadanie 9

---

Deployment i ReplicaSet są dwa różne zasoby w Kubernetes, które pomagają w zarządzaniu zestawem identycznych podów, ale mają nieco inne cele i funkcje. W skrócie, Deployment jest używany do zarządzania cyklem życia aplikacji, w tym wdrażania, aktualizacji i skalowania, podczas gdy ReplicaSet jest wykorzystywany do zapewnienia określonej liczby replik podów bez zapewniania funkcji wdrażania i zarządzania cyklem życia aplikacji.

## Zadanie 10

---

Robimy upgrade tylko minor release i w cluster manager zmieniamy wersje.

Cluster: Custom - potyczki Active

Namespace: fleet-default Age: 32 days

Making changes to cluster configuration may result in nodes reprovisioning. For more information see the [documentation](#).

Cluster Name\* potyczki

Cluster Description Any text you want that better describes this cluster

### Cluster Configuration

Basics

Kubernetes Version v1.26.15+rke2r1

Cloud Provider Default - RKE2 Embedded

Show deprecated Kubernetes patch versions

Container Network calico

Security

CIS Profile (None)

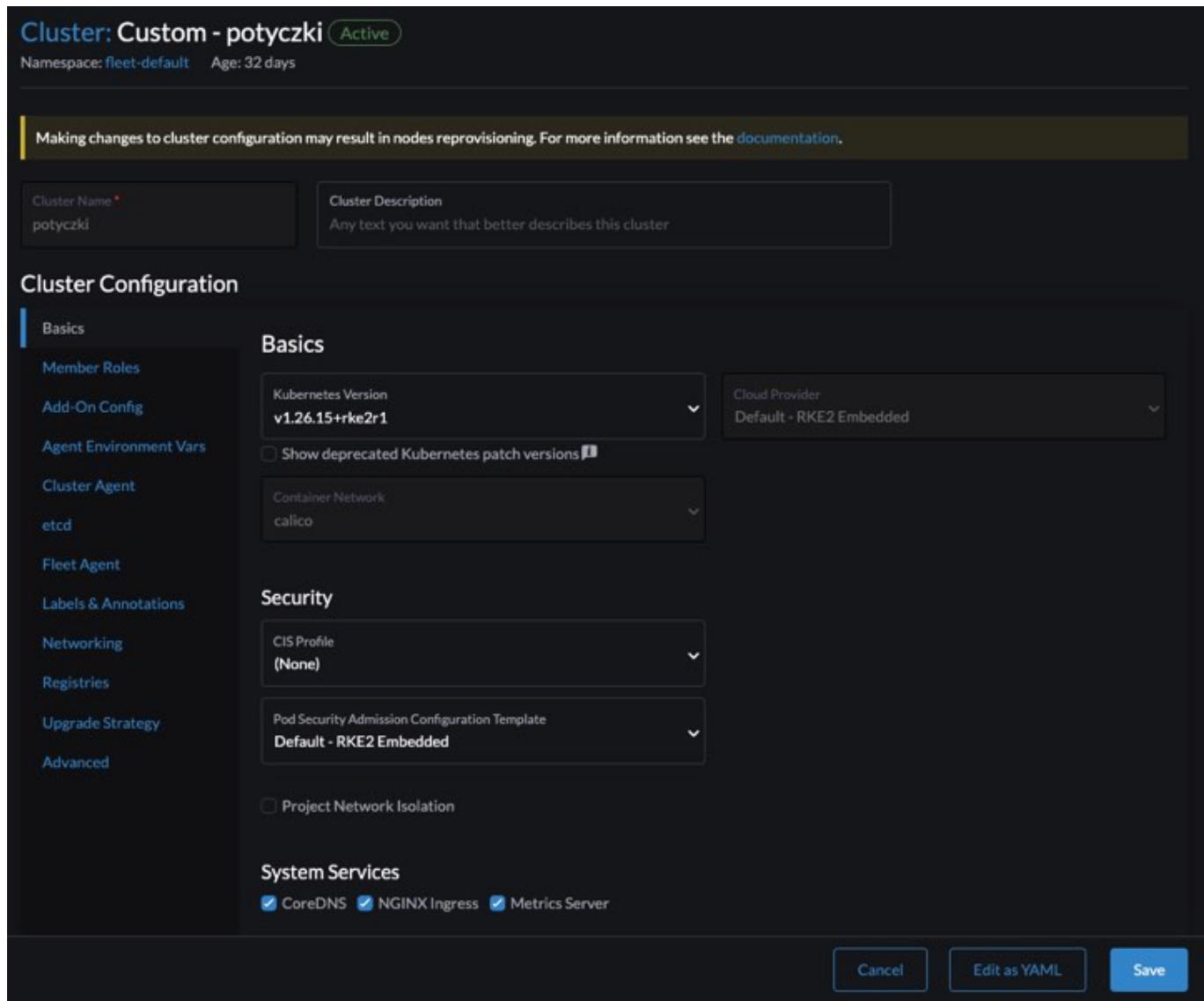
Pod Security Admission Configuration Template Default - RKE2 Embedded

Project Network Isolation

System Services

CoreDNS  NGINX Ingress  Metrics Server

Cancel Edit as YAML Save



Cluster został zaktualizowany 😊

## Zadanie 11

Tworzymy dwóch lokalnych użytkowników Muhammed Yassuff z hasłem **nI7ArT66yTywiWV0**, zmieniamy jego globalne permisje na User-Base oraz tworzymy go

## User: Create

### Credentials

Username \*  
muhammed.yassuff

Display Name  
Muhammed Yassuff

Description  
Dyrektor IT

- Ask user to change their password on next login  
 Generate a random password

Generated Password  
nI7ArT66yTywiWV0

[Copy](#)

[Suggest a password](#)

### Global Permissions

Controls what access the user has to administer the overall Explorer installation.

**Administrator**

Administrators have full control over the entire installation and all resources in all clusters.

**Restricted Administrator**

Restricted Admins have full control over all resources in all downstream clusters but no access to the local cluster.

Warning: The Restricted Administrator role has been deprecated as of Rancher 2.8.0 and will be removed in a future release - Check out the [Release Notes](#)

**Standard User**

Standard Users can create new clusters and manage clusters and projects they have been granted access to.

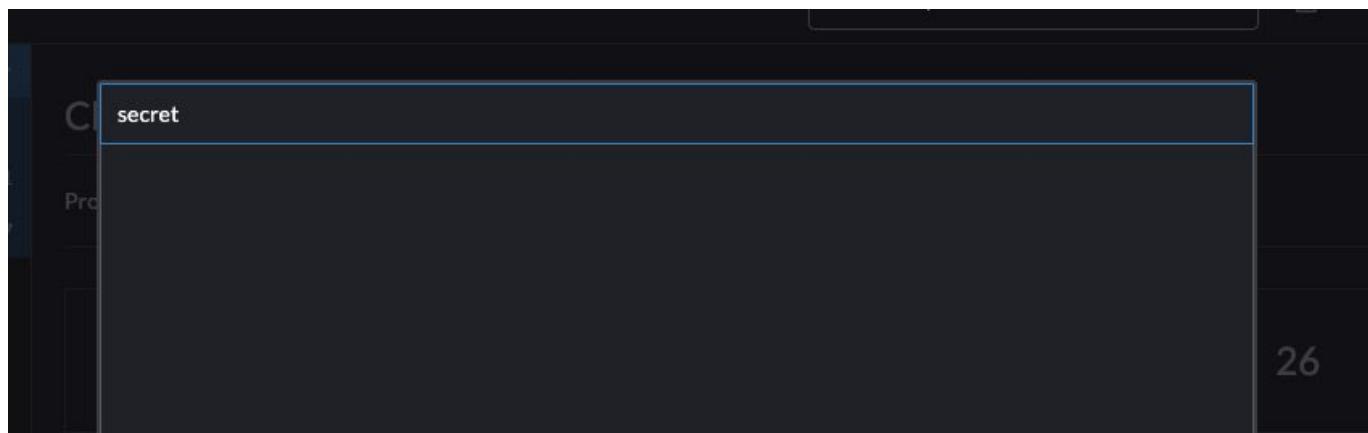
**User-Base**

User-Base users have login-access only.

Musimy dodać uprawnienia na poziomie clustra:

 Muhammed Yassuff (muhammed.yassuff)	<a href="#">View Nodes</a>	<a href="#">Remove</a>
Local		
 Muhammed Yassuff (muhammed.yassuff)	<a href="#">View All Projects</a>	<a href="#">Remove</a>
Local		
 Muhammed Yassuff (muhammed.yassuff)	<a href="#">View Cluster Catalogs</a>	<a href="#">Remove</a>
Local		

Wynik działania nadanych uprawnień pierwszemu użytkownikowi



Tworzymy użytkownika Muhammed Yussuf z hasłem **kEBJprbJyDxSDRbs** tworzymy dokładnie tak samo z permisjami User-Base.

## User: Create

### Credentials

Username\*  
muhammedyussuff

Display Name  
Muhamad Yussuff

Description  
Praktykant

Ask user to change their password on next login

Generate a random password

Generated Password  
kEBJprbjyDxSDRbs

Copy

 Suggest a password

### Global Permissions

Controls what access the user has to administer the overall Explorer installation.

Administrator

Administrators have full control over the entire installation and all resources in all clusters.

Restricted Administrator

Restricted Admins have full control over all resources in all downstream clusters but no access to the local cluster.

Warning: The Restricted Administrator role has been deprecated as of Rancher 2.8.0 and will be removed in a future release - Check out the [Release Notes](#)

Standard User

Standard Users can create new clusters and manage clusters and projects they have been granted access to.

User-Base

User-Base users have login-access only.

### Built-in

Additional roles to define more fine-grain permissions model.

Configure Authentication

Configure Feature Flags

Configure Node Drivers

Cancel

Create

Na poziomie projektu web-server przypisujemy użytkownika nazwa2 i dajemy mu permisje owner

## Add Project Member

### Select Member



Muhamad Yussuff (muhammed.yussuff)

Local



### Project Permissions

Controls what access users have to the Project

Owner

Owners have full control over the Project and all resources inside it.

Member

Members can manage the resources inside the Project but not change the Project itself.

Read Only

Members can only view the resources inside the Project but not change the resources.

Custom

Choose individual roles for this user.

Cancel

Add

## Zadanie 12

W configu deployment mysql limit ramu byl ustawiony na 64mb jest to za malo a po zmianie na 512mb kontener ruszył

**Deployment: mysql** Active

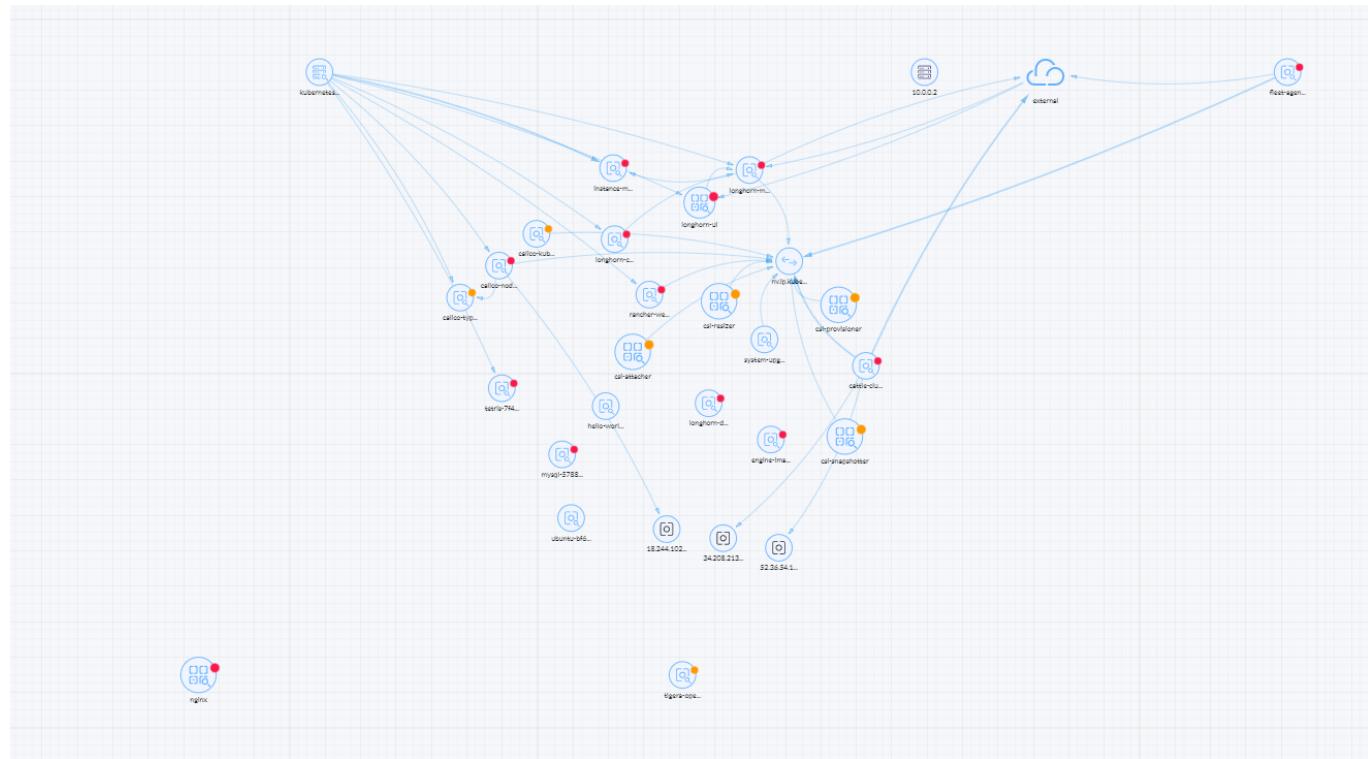
Namespace: default Age: 2.1 hours Pod Restarts: 0

Namespace * default	Name * mysql	Description Any text you want that better describes this resource												
Replicas * 1														
<a href="#">Deployment</a> <a href="#">Pod</a> <a href="#">mysql</a> <a href="#">+ Add Container</a>														
<b>General</b> <a href="#">Health Check</a> <b>Resources</b> <a href="#">Storage</a>														
<b>Resources</b> <table border="1"> <tr> <td>CPU Reservation 100</td> <td>mCPUs</td> <td>Memory Reservation 512</td> <td>MiB</td> </tr> <tr> <td>CPU Limit 100</td> <td>mCPUs</td> <td>Memory Limit 512</td> <td>MiB</td> </tr> <tr> <td colspan="2">NVIDIA GPU Limit/Reservation e.g. 1</td> <td colspan="2">GPUs</td> </tr> </table>			CPU Reservation 100	mCPUs	Memory Reservation 512	MiB	CPU Limit 100	mCPUs	Memory Limit 512	MiB	NVIDIA GPU Limit/Reservation e.g. 1		GPUs	
CPU Reservation 100	mCPUs	Memory Reservation 512	MiB											
CPU Limit 100	mCPUs	Memory Limit 512	MiB											
NVIDIA GPU Limit/Reservation e.g. 1		GPUs												

## Zadanie 13

---

Zrzut ekranu pokazujący połączenia sieciowe w Neu Vector



Opis analizy pakietów:

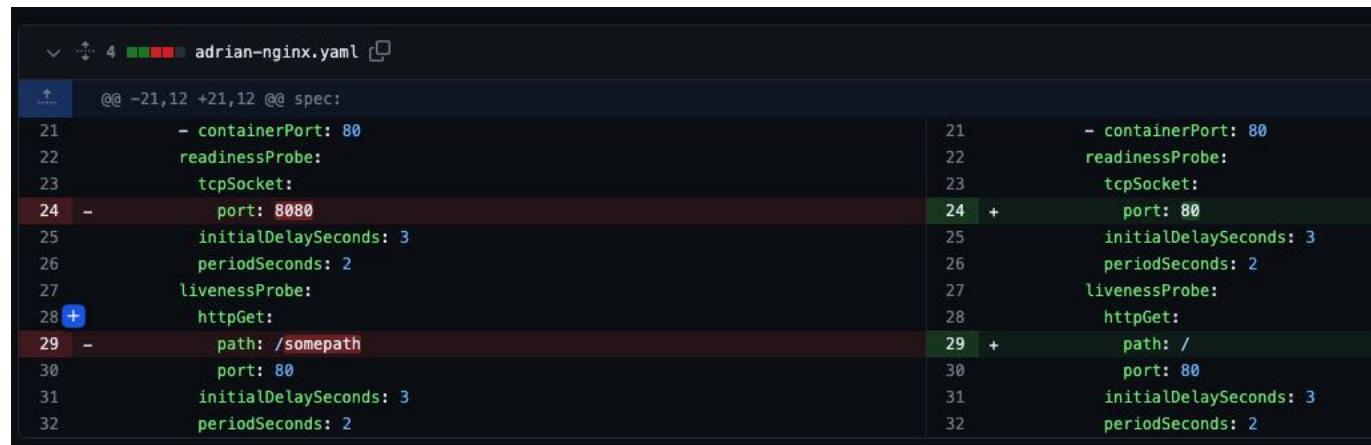
Analiza pakietów sieciowych polega na monitorowaniu, przechwytywaniu i analizowaniu ruchu sieciowego przepływającego przez sieć w celu zrozumienia zachowania sieci, diagnozowania problemów, wykrywania ataków lub optymalizacji wydajności sieciowej. Narzędziem powszechnie używanym do analizy pakietów sieciowych jest Wireshark.

Wireshark to darmowe i otwarte narzędzie do analizy pakietów sieciowych, które działa na wielu platformach, w tym Windows, macOS i Linux. Pozwala ono na przechwytywanie, analizowanie i wizualizowanie ruchu sieciowego w czasie rzeczywistym lub na podstawie zarejestrowanych danych. Wireshark oferuje zaawansowane funkcje filtrowania, dekodowania protokołów, analizy statystycznej oraz generowania raportów, co pozwala na dogłębne zrozumienie ruchu sieciowego i identyfikację potencjalnych problemów lub ataków.

## Zadanie 14

---

Trzeba było zmienić readiness probe na port 80 i ścieżkę liveness probe na "/"



```

@@ -21,12 +21,12 @@ spec:
 21     - containerPort: 80
 22     readinessProbe:
 23       tcpSocket:
 24 -     port: 8080
 25     initialDelaySeconds: 3
 26     periodSeconds: 2
 27   livenessProbe:
 28 +     httpGet:
 29 -       path: /somepath
 30     port: 80
 31     initialDelaySeconds: 3
 32     periodSeconds: 2

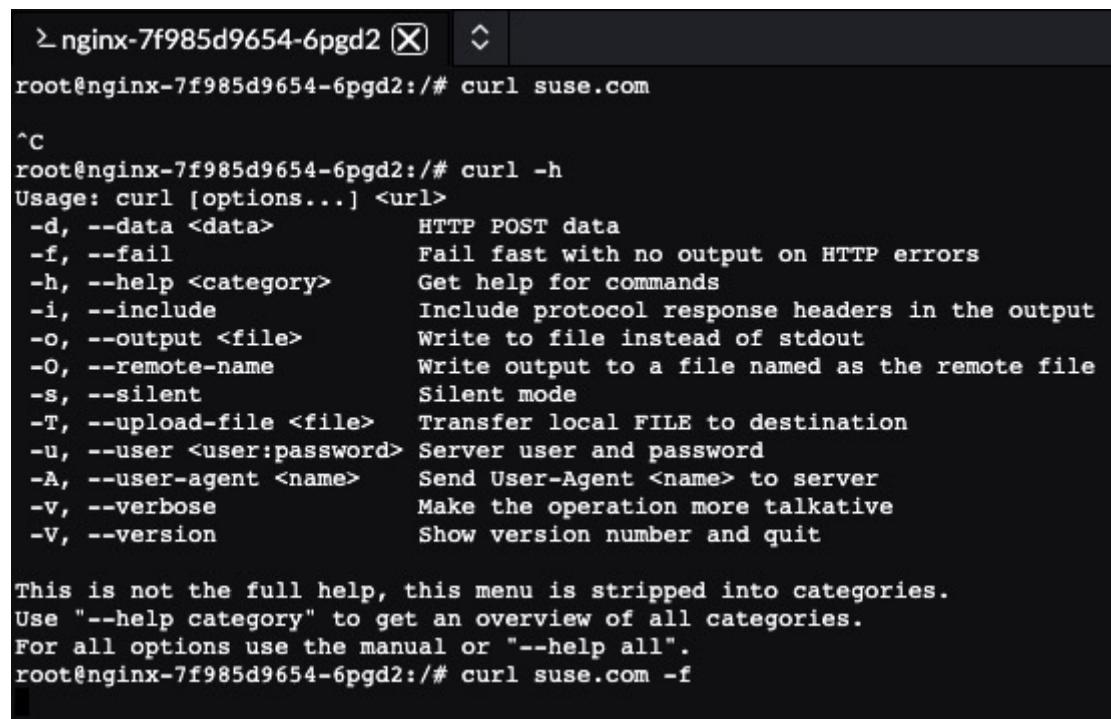
```

W projekcie defualt dodajemy nowy namespace adrian. Tworze w tym projekcie nowy deployment z pliku [adrian-nginx.yaml](#). Wszystkie zmiany są dostępne w commicie do pliku [adrian-nginx.yaml](#)

## Zadanie 15

---

Nie da się wykonać polecenia `curl`



```

root@nginx-7f985d9654-6pgd2:~# curl suse.com
^C
root@nginx-7f985d9654-6pgd2:~# curl -h
Usage: curl [options...] <url>
-d, --data <data>          HTTP POST data
-f, --fail                 Fail fast with no output on HTTP errors
-h, --help <category>      Get help for commands
-i, --include               Include protocol response headers in the output
-o, --output <file>        Write to file instead of stdout
-O, --remote-name          Write output to a file named as the remote file
-s, --silent                Silent mode
-T, --upload-file <file>   Transfer local FILE to destination
-u, --user <user:password> Server user and password
-A, --user-agent <name>    Send User-Agent <name> to server
-v, --verbose               Make the operation more talkative
-V, --version               Show version number and quit

This is not the full help, this menu is stripped into categories.
Use "--help category" to get an overview of all categories.
For all options use the manual or "--help all".
root@nginx-7f985d9654-6pgd2:~# curl suse.com -f

```

Tworze polityke dostępu do sieci w Neu Vector i w ustawieniach zmieniam Network Security Policy na protect

The screenshot shows the 'Network Security Policy Mode' toggle switch turned on (green). Below it, there are three tabs: 'Discover' (grey), 'Monitor' (grey), and 'Protect' (green, indicating it is selected). A note states: 'The Network Security Policy Mode can be set globally, independent of the default group policy mode.'

## Zadanie 16

Dodajemy nowy namespace serwis w projekcie default i tworze deployment. Trzeba było zmienić selektor dla serwisu na app:nginx

The screenshot shows the 'Deployment: nginx' page. It displays the following information:

- Namespace:** servis
- Age:** 16 secs
- Pod Restarts:** 0
- Image:** nginx
- Ready:** 1/1
- Up-to-date:** 1
- Available:** 1
- Labels:** app:nginx
- Annotations:** Show 1 annotation

**Pods by State**

State	Count	Age
Running	1	16 secs

**Services**

State	Name	Namespace	Target	Selector	Type	Age
Active	nginx-svc	servis	10.43.162.154:80	app=nginx	Cluster IP	22 secs

## Zadanie 17

Dodajemy nowy namespace w projekcie default o nazwie baza oraz tworze deployment pliku **baza.yaml** w nowo utworzonym namespace. To co było do poprawy to zmiana literówki zmiennej środowiskowej

The screenshot shows the 'Namespace: baza' page. It displays the following information:

- Namespace:** baza
- Status:** Active
- Age:** 18 secs

## Zadanie 18

W deplymencie trzeba było zmienić architekturę obrazu z arm64 na domyślną

**Deployment: hello-world** Updating

Namespace: default Age: 2.5 hours Pod Restarts: 0

Namespace *	default	Name *	hello-world	Description	Any text you want that better describes this resource
Replicas *	1				
<a href="#">Deployment</a> <a href="#">Pod</a> <a href="#">hello-world</a> <a href="#">+ Add Container</a>					
<div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <p><b>General</b></p> <ul style="list-style-type: none"> <li><a href="#">Health Check</a></li> <li><a href="#">Resources</a></li> <li><a href="#">Security Context</a></li> <li><a href="#">Storage</a></li> </ul> </div> <div style="flex: 3;"> <p><b>General</b></p> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <p>Container Name</p> <input type="text" value="hello-world"/> </div> <div style="flex: 1;"> <p><input type="radio"/> Init Container</p> <p><input checked="" type="radio"/> Standard Container</p> </div> </div> <p><b>Image</b></p> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <p>Container Image</p> <input type="text" value="hello-world"/> </div> <div style="flex: 1;"> <p>Pull Policy</p> <input type="text" value="Always"/> </div> </div> <div style="margin-top: 10px;"> <p>Pull Secrets</p> </div> </div> </div>					
<p><b>Networking</b> ⓘ</p> <p>Define a Service to expose the container, or define a non-functional, named port so that humans will know where the app within the container is expected to run.</p> <p><a href="#">Add Port or Service</a></p>					
<p><b>Command</b></p> <pre>curl -H "Content-Type: application/json" -X POST -d '{"image": "hello-world", "command": "echo Hello World"}' http://127.0.0.1:8080/api/v1/namespaces/default/pods/hello-world</pre>					
<a href="#">Cancel</a> <a href="#">Edit as YAML</a> <a href="#">Save</a>					

Kontener bedzie znajdował się w stanie "crashloopbackoff" ponieważ jest to kontener wykonujący swoje zadanie raz i kończący swoją pracę przez co kubernetes próbuje odpalać go na nowo cały czas.

**Deployment: hello-world** Updating

Namespace: default Age: 2.6 hours Pod Restarts: 4

Deployment does not have minimum availability.

Image: hello-world Ready: 0/1 Up-to-date: 1 Available: 0

Labels: `app: hello-world`

Annotations: [Show 1 annotation](#)

**Pods by State**

State	Count	Actions
Completed	1	<a href="#">Scale</a> <a href="#">-</a> <a href="#">1</a> <a href="#">+</a>

[Download YAML](#) [Delete](#)

State	Name	Image	Ready	Restarts	IP	Node	Age
Completed	hello-world-7b88845654-nwbjv	hello-world	0/1	4 (53s ago)	10.42.184.206	kubernetes05	1.9 mins

Containers with unready status: [hello-world]

Rozwiąza niem dla takiej usługi jest uzycie tzw. "jobu" co wykonalismy w namespace default.

The screenshot shows the Kubernetes UI for a job named 'hello-world'. The top navigation bar includes 'Detail', 'Config', and a three-dot menu. Below the title, it shows the namespace (nginx), age (13 secs), and pod restarts (0). It also displays the start time (13 secs ago), duration (10 secs), image (hello-world), labels (controller-uid: b4bf2d01-39e9-4442-9100-76b1d58ec81b, job-name: hello-world), and annotations (Show 1 annotation).

**Runs**

Successful	Running	Failed
1	0	0

**Pods**

State	Name	Image	Ready	Restarts	IP	Node	Age
Completed	hello-world-ws9tt	hello-world	0/1	0	10.42.184.205	kubernetes05	16 secs