

Kubernetes w praktyce – podręcznik labowy.

Spis treści

Lab 1: Instalacja klastra k8s i zapoznanie z pojęciami namespace i context	3
Task 1: Zapoznanie ze środowiskiem i kubectl get	3
Task 2: Tworzenie namespace i zarządzanie kontekstami	4
Lab 2 - Tworzenie przykładowej złożonej aplikacji	8
Lab 3: Zapoznanie z pojęciem POD	11
Task1: Tworzenie PODa „on-fly” przy użyciu kubectl run	11
Task 2: Dostęp do PODa	12
Task 3: Kolejny POD z YAML	13
Task 4: Ręczna modyfikacja deklaracji PODa	14
Task 5: Kubectl exec i pod z 2 kontenerami	16
Task 6: Kasowanie na podstawie label	20
Task 7: Domyślne zachowanie podów (...i nie tylko)	21
Lab 4: Zapoznanie z obiektem deployment	24
Task 1. Stworzenie obiektu Deployment (a także ReplicaSet i Pod)	24
Task 2. Skalowanie obiektu Deployment 2 metodami	27
Task 3. Deployment – strategia RollingUpdate	30
Task 5. Deployment – historia zmian	35
Task 6. Deployment – historia zmian część 2	41
Task 7. Deployment – struktura	44
Lab 6: Rozmieszczanie Podów na klastrze Kubernetes.	46
Task 1: nodeName w specyfikacji Podów	46
Task 2: nodeSelector w specyfikacji Podów	47
Task 3: Affinity (lub anti-Affinity) w specyfikacji Podów	48
Task 4: Affinity (lub anti-Affinity) w specyfikacji Podów	49
Lab 7: Obiekty w Kubernetes	51
Task 1: Obiekt Job	51
Task 2: Obiekt CronJob	55
Task 3: Obiekt Deployment w klastrze	57

Task 4: Obiekt DaemonSet w klastrze	62
Task 5: Obiekt DaemonSet – wykorzystanie taints i tolerations	66
Task 6: Obiekt Deployment – obiekt Service	70
Task 7: Obiekt Deployment – obiekt NodePort	75
Task 8: Obiekt Deployment – obiekt LoadBalancer	78
Task 9: Obiekt StatefulSet i Headless Service	80
Lab 8: Obiekt Secret	84
Task 1. Tworzenie obiektu typu Secret	84
Task 2: Podmontowanie obiektu Secret	90
Lab 9: Obiekt ConfigMap	93
Lab 10: Sposoby montowania zasobów – volumeny	95
Task 1: Podmontowanie volumenu	95
Task 2: Volumeny efemeryczne	98
Task 3: Przykład obiektów PersistentVolume	100
i PersistentVolumeClaim	100
Lab 11: Limitowanie zasobów w Kubernetes	108
Dodatek A.	116
Task 1: Helm	116
Task 2: Wdrożenie Dashboard za pomocą Helm	123
.....	129

Lab 1: Instalacja klastra k8s i zapoznanie z pojęciami namespace i context

Cele:

instalacja klastra

zaznajomienie z get, get -A

przykłady obiektów nie widocznych po get all

tworzenie namespace (z dry-run)

zarządzanie kontekstami

Task 1: Zapoznanie ze środowiskiem i kubectl get

Wszystkie aktywności powinny być wykonywane na koncie użytkownika student. Jeśli jesteś na innym koncie wykonaj polecenie:

```
[root@base ~]# su - student
```

Aby tabulator podpowiadał nazwy komend i obiektów w kubectl:

uwaga - jedna linia

```
[student@master ~]$ kubectl completion bash >/home/student/kubectl-completion
```

```
[student@master ~]$ source /home/student/kubectl-completion
```

Podstawowy przegląd:

```
[student@master ~]$ kubectl get nodes
```

```
master Ready control-plane,master 14d v1.20.2
```

```
[student@master ~]$ kubectl get namespaces
```

NAME	STATUS	AGE
default	Active	14d
kube-node-lease	Active	14d
kube-public	Active	14d
kube-system	Active	14d

Wersja skrócona:

```
[student@master ~]$ kubectl get ns
```

NAME	STATUS	AGE
default	Active	14d
kube-node-lease	Active	14d
kube-public	Active	14d
kube-system	Active	14d

[student@master ~]\$ **kubectl get all**

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default	service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	14d

Wszystkie ns:

[student@master ~]\$ **kubectl get all -A**

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	pod/coredns-74ff55c5b-z2hzk	1/1	Running	2	14d
kube-system	pod/etcd-minikube	1/1	Running	2	14d
kube-system	pod/kube-apiserver-minikube	1/1	Running	2	14d
kube-system	pod/kube-controller-manager-minikube	1/1	Running	2	14d
kube-system	pod/kube-proxy-h75sn	1/1	Running	2	14d
kube-system	pod/kube-scheduler-minikube	1/1	Running	2	14d
kube-system	pod/storage-provisioner	1/1	Running	4	14d

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default	service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	14d
kube-system	service/kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP,53/TCP,9153/TCP	14d

NAMESPACE	NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
kube-system	daemonset.apps/kube-proxy	1	1	1	1	1	kubernetes.io/os=linux	14d

NAMESPACE	NAME	READY	UP-TO-DATE	AVAILABLE	AGE
kube-system	deployment.apps/coredns	1/1	1	1	14d

NAMESPACE	NAME	DESIRED	CURRENT	READY	AGE
kube-system	replicaset.apps/coredns-74ff55c5b	1	1	1	14d

Obiekt typu secrets należy do wielu, które nie są pokazywane przez get all:

[student@master ~]\$ **kubectl get secrets**

NAME	TYPE	DATA	AGE
default-token-h4cdb	kubernetes.io/service-account-token	3	3h26m

Task 2: Tworzenie namespace i zarządzanie kontekstami

Tworzymy nowy namespace, tak, aby pozostały pliki konfiguracyjne umożliwiające odtworzenie.

```
[student@master ~]$ kubectl create namespace lab1 --dry-run=server -o yaml
```

```
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: "2021-03-16T10:05:08Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:status:
        f:phase: {}
    manager: kubectl-create
    operation: Update
    time: "2021-03-16T10:05:08Z"
  name: lab1
  uid: 04b596ad-1a5c-49e7-b1cf-ac200b2eea03
spec:
  finalizers:
  - kubernetes
status:
  phase: Active
```

```
[student@master ~]$ mkdir -p k8s/lab1
```

uwaga - jedna linia

```
[student@master ~]$ kubectl create namespace lab1 --dry-run=server -o
yaml >/home/student/k8s/lab1/1-namespace.yaml
```

```
[student@base ~]$ kubectl get ns
```

...

nie został stworzony

uwaga - jedna linia

```
[student@master ~]$ kubectl apply -f /home/student/k8s/lab1/1-namespace.yaml
```

```
[student@master ~]$ kubectl get ns
```

NAME	STATUS	AGE
default	Active	127d
kube-node-lease	Active	127d
kube-public	Active	127d
kube-system	Active	127d
lab1	Active	2s

Różnica między --dry-run=server a --dry-run=client: spróbujmy stworzyć jeszcze raz to co już istnieje:

```
[student@master ~]$ kubectl create namespace lab1 --dry-run=server -o yaml
```

Error from server (AlreadyExists): namespaces "lab1" already exists

```
[student@master ~]$ kubectl create namespace lab1 --dry-run=client -o yaml
```

```
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: null
  name: lab1
spec: {}
status: {}
```

Praca z kontekstami

```
[student@master ~]$ kubectl config view
```

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://10.10.1.10:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
    name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

uwaga - jedna linia

```
[student@master ~]$ kubectl config set-context moj-lab1 --namespace=lab1 --cluster=kubernetes --user=kubernetes-admin
```

Context "moj-lab1" created.

```
[student@master ~]$ kubectl config get-contexts
```

CURRENT	NAME	CLUSTER	AUTHINFO	NAMESPACE
*	kubernetes-admin@kubernetes	kubernetes	kubernetes-admin	
	moj-lab1	kubernetes	kubernetes-admin	lab1

```
[student@master ~]$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/lab-pod	1/1	Running	0	96m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	127d

```
[student@master ~]$ kubectl config use-context moj-lab1
```

Switched to context "moj-lab1".

```
[student@master ~]$ kubectl get all
```

No resources found in lab1 namespace.

```
[student@master ~]$ kubectl config view
```

```
...
- context:
  cluster: kubernetes
  namespace: lab1
  user: kubernetes-admin
  name: moj-lab1
  current-context: moj-lab1
...
```

Lab 2 - Tworzenie przykładowej złożonej aplikacji

Cele:

Zaznajomienie z tworzeniem kilku-tierowej aplikacji z gotowych deklaracji

Dokładniejsze zaznajomienie z przeglądaniem obiektów k8s

```
[student@master lab2]$ kubectl config use-context moj-lab1
```

```
[student@master]$ mkdir k8s/lab2/
```

```
[student@master]$ cd /home/student/k8s/lab2/
```

```
[student@master lab2]$ git clone https://github.com/lamw/yelb.git
```

```
[student@master lab2]$ kubectl get all
```

No resources found in lab1 namespace.

uwaga - jedna linia

```
[student@master lab2]$ kubectl apply -f  
/home/student/k8s/lab2/yelb/deployments/platformdeployment/Kubernetes/yaml/yelb-k8s-minikube-  
nodeport.yaml
```

Wykonaj następujące polecenia:

```
kubectl get all
```

```
kubectl get pods
```

```
kubectl get pods -w
```

```
kubectl get pods --show-labels
```

```
kubectl get pods -w -o wide
```

```
kubectl get pods -w -o wide --show-labels
```

```
kubectl get events
```

Odpowiedz na pytania: co daje opcja -w? Co daje opcja -o wide?

Listujemy tylko tier frontend:


```
[student@master lab2]$ kubectl get all -l tier=frontend
```

NAME	READY	STATUS	RESTARTS	AGE
pod/yelb-ui-798667d648-bg4fd	1/1	Running	0	11m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/yelb-ui	NodePort	10.110.15.197	<none>	80:31438/TCP	11m

Więcej informacji o PODzie (identyfikatory będą się różnić):

```
[student@master lab2]$ kubectl describe pod yelb-ui-798667d648-bg4fd
```

```
Name:      yelb-ui-798667d648-bg4fd
Namespace: lab1
Priority:   0
Node:      node1/10.10.1.10
```

...

```
[student@master lab2]$ kubectl logs yelb-ui-798667d648-bg4fd
```

...

Pojęcie serwisu, ich rodzaj będzie omawiane później, teraz jednak zobaczmy deklarację:

```
[student@master lab2]$ kubectl describe service yelb-ui
```

```
Name:      yelb-ui
Namespace: lab1
Labels:    app=yelb-ui
           tier=frontend
Annotations: <none>
Selector:   app=yelb-ui,tier=frontend
Type:       NodePort
IP Families: <none>
IP:         10.110.15.197
IPs:        10.110.15.197
Port:       <unset> 80/TCP
TargetPort: 80/TCP
NodePort:   <unset> 31438/TCP
Endpoints:  172.17.0.4:80
Session Affinity: None
External Traffic Policy: Cluster
Events:     <none>
```

Sprawdź IP na jakim wystawiony jest service yelb-ui (pole Ips), oraz nr portu – NodePort. Zobacz czy dostęp do aplikacji jest możliwy wykorzystując te informacje.

```
student@master:~$ curl 10.102.86.206
```

<!doctype html>

```

<html>
<head>
  <script src="env.js"></script>
  <meta charset="utf-8">
  <title>Yelb</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico?v=2">
</head>
<body>
<yelb>Loading...</yelb>
<script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="styles.bundle.js"></script><script
type="text/javascript" src="scripts.bundle.js"></script><script type="text/javascript" src="vendor.bundle.js"></script><script
type="text/javascript" src="main.bundle.js"></script></body>
</html>

```

student@master:~\$ curl 10.10.1.20:32719

```

<!doctype html>
<html>
<head>
  <script src="env.js"></script>
  <meta charset="utf-8">
  <title>Yelb</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico?v=2">
</head>
<body>
<yelb>Loading...</yelb>
<script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="styles.bundle.js"></script><script
type="text/javascript" src="scripts.bundle.js"></script><script type="text/javascript" src="vendor.bundle.js"></script><script
type="text/javascript" src="main.bundle.js"></script></body>
</html>

```

student@master:~\$ curl 10.10.1.30:32719

```

<!doctype html>
<html>
<head>
  <script src="env.js"></script>
  <meta charset="utf-8">
  <title>Yelb</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico?v=2">
</head>
<body>
<yelb>Loading...</yelb>
<script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="styles.bundle.js"></script><script
type="text/javascript" src="scripts.bundle.js"></script><script type="text/javascript" src="vendor.bundle.js"></script><script
type="text/javascript" src="main.bundle.js"></script></body>
</html>

```

Lab 3: Zapoznanie z pojęciem POD

Czynności:

- Jak utworzyć POD
- Jak sprawdzić logi i zdarzenia z PODa
- Jak dostać się do powłoki wewnątrz PODa
- Jak przekierować port bezpośrednio do PODa
- Jak pracować z PODem z więcej niż jednym kontenerem

Task1: Tworzenie PODa „on-fly” przy użyciu kubectl run

uwaga - jedna linia

```
[student@master ~]$ kubectl run lab3-pod -l lab=pod,task=1 --image=httpd --port 80
```

```
pod/lab3-pod created
```

```
[student@master ~]$ kubectl get all -l lab=pod
```

NAME	READY	STATUS	RESTARTS	AGE
pod/lab3-pod	1/1	Running	0	45s

```
[student@master ~]$ kubectl describe pod/lab3-pod
```

```
Name:      lab3-pod
Namespace:  lab1
Priority:    0
Node:       minikube/192.168.39.3
Start Time: Tue, 06 Jul 2021 19:11:55 +0200
Labels:     lab=pod
            task=1
Annotations: <none>
Status:     Running
IP:         172.17.0.8
IPs:
  IP: 172.17.0.8
Containers:
  lab3-pod:
    Container ID:  docker://0d76f980500feaf41a909e322e4f14e9797b6e3e583170f0bd481fffde4cdf1e
    Image:         httpd
    Image ID:      docker-pullable://httpd@sha256:317cc1a2ded5e96225e4181323737f6d29b4fda58a6cc840a5752af6493a231f
    Port:         80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Tue, 06 Jul 2021 19:11:58 +0200
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
```

```

/var/run/secrets/kubernetes.io/serviceaccount from default-token-mphxp (ro)
Conditions:
  Type      Status
  Initialized   True
  Ready        True
  ContainersReady  True
  PodScheduled   True
Volumes:
  default-token-mphxp:
    Type:      Secret (a volume populated by a Secret)
    SecretName: default-token-mphxp
    Optional:   false
QoS Class:     BestEffort
Node-Selectors: <none>
Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason      Age    From          Message
  ----    -
Normal   Scheduled   100s   default-scheduler Successfully assigned lab1/lab3-pod to minikube
Normal   Pulling     99s    kubelet       Pulling image "httpd"
Normal   Pulled      97s    kubelet       Successfully pulled image "httpd" in 1.461198191s
Normal   Created     97s    kubelet       Created container lab3-pod
Normal   Started     97s    kubelet       Started container lab3-pod

```

Task 2: Dostęp do PODa

Spróbujmy uruchomić shell w kontenerze:

```
[student@master ~]$ kubectl exec -it lab3-pod -- /bin/sh
```

```
# hostname
lab3-pod
```

```
# pwd
/usr/local/apache2
```

```
# exit
```

```
[student@master ~]$ kubectl exec -it lab3-pod -- /bin/bash
```

```
root@lab3-pod:/usr/local/apache2# hostname
lab3-pod
```

```
root@lab3-pod:/usr/local/apache2# pwd
/usr/local/apache2
```

```
root@lab3-pod:/usr/local/apache2# exit
```

Spróbujmy przekierować port bezpośrednio do POD'a aby przekonać się czy działa serwis www:

```
[student@master ~]$ kubectl port-forward pod/lab3-pod 9090:80
```

Forwarding from 127.0.0.1:9090 -> 80

Handling connection for 9090

Przy pomocy innej konsoli możemy sprawdzić odpowiedź:

```
[student@master ~]$ curl localhost:9090
```

```
<html><body><h1>It works!</h1></body></html>
```

Aby przerwać przekierowanie należy nacisnąć [Ctrl]+[c]

Task 3: Kolejny POD z YAML

Utwórzmy plik YAML opisujący dwa nowe PODy:

```
[student@master ~]$ mkdir -p k8s/lab3
```

```
[student@master ~]$ cd k8s/lab3/
```

uwaga - jedna linia

```
[student@master lab3]$ kubectl run pod-lab3 --image=nginx --labels "lab=pod,task=3a" --dry-run=client -o yaml > /home/student/k8s/lab3/pod-lab3.yaml
```

i drugi z obrazem apache:

uwaga - jedna linia

```
[student@master lab3]$ kubectl run httpd-from-file --image=httpd --labels "lab=pod,task=3b" --dry-run=client -o yaml > /home/student/k8s/lab3/httpd-pod.yaml
```

Utwórzmy PODy na podstawie plików:

```
[student@master lab3]$ ls
```

httpd-pod.yaml pod-lab3.yaml

Dodaj do konfiguracji kontenera imagePullPolicy: IfNotPresent

```
[student@master lab3]$ kubectl apply -f pod-lab3.yaml
```

pod/echo-from-file created

Dodaj do konfiguracji kontenera imagePullPolicy: IfNotPresent

```
[student@master lab3]$ kubectl apply -f httpd-pod.yaml
```

pod/httpd-from-file created

```
[student@master lab3]$ kubectl get pods -l lab=pod
```

NAME	READY	STATUS	RESTARTS	AGE
echo-from-file	1/1	Running	0	13s
lab3-pod	1/1	Running	0	69m
httpd-from-file	1/1	Running	0	10s

Skasujmy pierwszego z nich, używając pliku jako źródła informacji o kasowanych obiektach:

```
[student@master lab3]$ kubectl delete -f pod-lab3.yaml
```

pod "echo-from-file" deleted

```
[student@master lab3]$ kubectl get pods -l lab=pod
```

NAME	READY	STATUS	RESTARTS	AGE
echo-lab-pod	1/1	Running	0	71m
httpd-from-file	1/1	Running	0	2m8s

Task 4: Ręczna modyfikacja deklaracji PODa

```
[student@master lab3]$ cp httpd-pod.yaml combo.yaml
```

```
[student@master lab3]$ vi combo.yaml
```

Wprowadź poniższe modyfikacje w pliku :

```
[student@master lab3]$ cat combo.yaml
```

```
apiVersion: v1
kind: Pod
```

```
metadata:
  creationTimestamp: null
  labels:
    lab: pod
    task: "4"
  name: nowy-httpd
spec:
  containers:
  - image: httpd
    name: nowy-httpd-container
    imagePullPolicy: IfNotPresent
    ports:
    - containerPort: 80
    resources: {}

  dnsPolicy: ClusterFirst
  restartPolicy: Always
```

```
[student@master lab3]$ kubectl apply -f combo.yaml
```

```
pod/nowy-httpd created
```

```
[student@master lab3]$ kubectl describe pod nowy-httpd
```

```
Name:      nowy-httpd
Namespace:  lab1
Priority:    0
Node:       minikube/192.168.39.3
Start Time: Tue, 06 Jul 2021 19:42:13 +0200
Labels:     lab=pod
            task=4
Annotations: <none>
Status:      Running
IP:          172.17.0.8
IPs:
  IP: 172.17.0.8
Containers:
  nowy-httpd-container:
    Container ID:  docker://486bef653fb0bd3cff82fc38d8658f9bd44730f7bebb80aae492b95440b7b45a
    Image:         httpd
    Image ID:      docker-pullable://httpd@sha256:317cc1a2ded5e96225e4181323737f6d29b4fda58a6cc840a5752af6493a231f
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Tue, 06 Jul 2021 19:42:16 +0200
    Ready:         True
    Restart Count: 0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-mpxhp (ro)
Conditions:
```

```

Type          Status
Initialized    True
Ready          True
ContainersReady True
PodScheduled   True
Volumes:
default-token-mphxp:
  Type:      Secret (a volume populated by a Secret)
  SecretName: default-token-mphxp
  Optional:  false
QoS Class:    BestEffort
Node-Selectors: <none>
Tolerations:  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
               node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
Type Reason Age From Message
----
Normal Scheduled 24s default-scheduler Successfully assigned lab1/nowy-httpd to minikube
Normal Pulling 24s kubelet Pulling image "httpd"
Normal Pulled 22s kubelet Successfully pulled image "httpd" in 1.445828582s
Normal Created 22s kubelet Created container nowy-httpd-container
Normal Started 22s kubelet Started container nowy-httpd-container

```

```
[student@master lab3]$ kubectl get pods -l lab=pod
```

```

NAME          READY STATUS RESTARTS AGE
httpd-from-file 1/1 Running 0      10m
nowy-httpd     1/1 Running 0      6m23s

```

Task 5: Kubectl exec i pod z 2 kontenerami

Najpierw „zwykły POD”

```
[student@master lab3]$ kubectl exec -it httpd-from-file -- /bin/bash
```

```
root@httpd-from-file:/usr/local/apache2# ls /bin
```

```

bash dash dnsdomainname findmnt ln mktemp pidof run-parts sync uname zcat zgrep
cat date domainname grep login more pwd sed tar uncompress zcmp zless
chgrp dd echo gunzip ls mount rbash sh tempfile vdir zdiff zmore
chmod df egrep gzexe lsblk mountpoint readlink sleep touch wdctl zegrep znew
chown dir false gzip mkdir mv rm stty true which zfgrep
cp dmesg fgrep hostname mknod nisdomainname rmdir su umount ypdomainname zforce

```

```
root@httpd-from-file:/usr/local/apache2# apt list | head
```

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Listing...


```
adduser/now 3.118 all [installed,local]
apt/now 1.8.2.3 amd64 [installed,local]
base-files/now 10.3+deb10u10 amd64 [installed,local]
base-passwd/now 3.5.46 amd64 [installed,local]
bash/now 5.0-4 amd64 [installed,local]
bsdutils/now 1:2.33.1-0.1 amd64 [installed,local]
coreutils/now 8.30-3 amd64 [installed,local]
dash/now 0.5.10.2-5 amd64 [installed,local]
debconf/now 1.5.71 all [installed,local]
```

```
root@httpd-from-file:/usr/local/apache2# cat /etc/debian_version
```

```
10.10
```

W przypadku PODa z dwoma kontenerami można użyć describe, aby ustalić nazwy kontenerów do których chcemy się dostać (zauważ, że oba kontenery będą miały wspólny adres IP poda). Stwórz poda z podwójnym kontenerem a następnie znajdź nazwy poszczególnych kontenerów:

```
[student@master lab3]$ vim podwojny.yaml
```

```
[student@master lab3]$ cat podwojny.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: podwojny
spec:
  containers:
    - name: busy1
      image: busybox
      command: ["sh", "-c", "sleep 240"]
      imagePullPolicy: IfNotPresent

    - name: busy2
      image: busybox
      command: ["sh", "-c", "sleep 360"]
      imagePullPolicy: IfNotPresent
```

```
[student@master lab3]$ kubectl apply -f podwojny.yaml
```

```
pod/podwojny created
```

```
[student@master lab3]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
mynginx-5b686ccd46-q5z7d	1/1	Running	0	145m
podwojny	2/2	Running	0	13s
redis-server-74556bbcb7-bncdm	1/1	Running	0	3h38m
second-pod	0/1	Completed	0	90m
yelb-appserver-d584bb889-xh95q	1/1	Running	0	3h38m

yelb-db-694586cd78-86db6	1/1	Running	0	3h38m
yelb-ui-798667d648-wcsjq	1/1	Running	0	3h38m

[student@master lab3]\$ **kubectl describe pod podwojny**

Name: podwojny
Namespace: lab1
Priority: 0
Node: minikube/192.168.39.3
Start Time: Tue, 06 Jul 2021 22:38:06 +0200
Labels: <none>
Annotations: <none>
Status: Running
IP: 172.17.0.9
IPs:
IP: 172.17.0.9

Containers:

busy1:

Container ID: docker://36ecc827c4884540b03d8ca4164631c5d9b36b5edd631d12bdad05455842cdf
Image: busybox
Image ID: docker-pullable://busybox@sha256:930490f97e5b921535c153e0e7110d251134cc4b72bbb8133c6a5065cc68580d
Port: <none>
Host Port: <none>

Command:

sh
-c
sleep 240

State: Running
Started: Tue, 06 Jul 2021 22:38:09 +0200
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
/var/run/secrets/kubernetes.io/serviceaccount from default-token-mpxhp (ro)

busy2:

Container ID: docker://38e66e659ecc9daa909fa35fc0e390b5b01a790ec6483ee0a05b8092c7fd495b
Image: busybox
Image ID: docker-pullable://busybox@sha256:930490f97e5b921535c153e0e7110d251134cc4b72bbb8133c6a5065cc68580d
Port: <none>
Host Port: <none>

Command:

sh
-c
sleep 360

State: Running
Started: Tue, 06 Jul 2021 22:38:12 +0200
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
/var/run/secrets/kubernetes.io/serviceaccount from default-token-mpxhp (ro)

Conditions:

Type	Status
Initialized	True
Ready	True

```
ContainersReady True
PodScheduled True
Volumes:
default-token-mphxp:
  Type: Secret (a volume populated by a Secret)
  SecretName: default-token-mphxp
  Optional: false
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type Reason Age From Message
  ----
Normal Scheduled 41s default-scheduler Successfully assigned lab1/podwojny to minikube
Normal Pulling 40s kubelet Pulling image "busybox"
Normal Pulled 38s kubelet Successfully pulled image "busybox" in 1.524292288s
Normal Created 38s kubelet Created container busy1
Normal Started 38s kubelet Started container busy1
Normal Pulling 38s kubelet Pulling image "busybox"
Normal Pulled 36s kubelet Successfully pulled image "busybox" in 2.685702166s
Normal Created 35s kubelet Created container busy2
Normal Started 35s kubelet Started container busy2
```

```
[student@master lab3]$ kubectl exec -it podwojny -- /bin/sh
```

```
/ # ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
76: eth0@if77: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
    link/ether 02:42:ac:11:00:08 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.8/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
```

```
/ # top
```

```
Mem: 2267984K used, 121056K free, 628976K shrd, 104556K buff, 1268416K cached
CPU:  4.0% usr  2.5% sys  0.0% nic 93.0% idle  0.0% io  0.0% irq  0.1% irq
Load average: 0.98 0.68 0.48 2/727 15
PID PPID USER STAT VSZ %VSZ CPU %CPU COMMAND
  9   0 root  S   1324 0.0  0 0.0 /bin/sh
 15   9 root  R   1320 0.0  1 0.0 top
   1   0 root  S   1308 0.0  0 0.0 sleep 240
```

```
/ # exit
```

```
[student@master lab3]$ kubectl exec -it -c busy2 podwojny -- /bin/sh
```

```

/ #
/ # ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
76: eth0@if77: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
    link/ether 02:42:ac:11:00:08 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.8/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever

```

```

/ # top

```

```

Mem: 2268228K used, 120812K free, 628976K shrd, 105068K buff, 1268816K cached
CPU:  4.5% usr  2.6% sys  0.0% nic 92.3% idle  0.1% io  0.0% irq  0.2% sirq
Load average: 0.25 0.37 0.39 2/723 14
PID  PPID USER  STAT  VSZ %VSZ CPU %CPU COMMAND
  7   0 root   S    1328  0.0  1  0.0 /bin/sh
 14   7 root   R    1320  0.0  0  0.0 top
  1   0 root   S    1308  0.0  0  0.0 sleep 360
/ #

```

Task 6: Kasowanie na podstawie label

```

[student@master lab3]$ kubectl get all --show-labels

```

NAME	READY	STATUS	RESTARTS	AGE	LABELS
pod/first-pod	1/1	Running	7	13m	<none>
pod/httpd-from-file	1/1	Running	0	95m	lab=pod,task=3b
pod/mynginx-5b686ccd46-q5z7d	1/1	Running	0	60m	app=mynginx,pod-template-hash=5b686ccd46
pod/nowy-httpd	1/1	Running	0	91m	lab=pod,task=4
pod/redis-server-74556bbcb7-bncdm	1/1	Running	0	133m	app=redis-server,pod-template-hash=74556bbcb7,tier=cache
pod/second-pod	0/1	Completed	0	5m13s	<none>
pod/yelb-appserver-d584bb889-xh95q	1/1	Running	0	133m	app=yelb-appserver,pod-template-hash=d584bb889,tier=middletier
pod/yelb-db-694586cd78-86db6	1/1	Running	0	133m	app=yelb-db,pod-template-hash=694586cd78,tier=backenddb
pod/yelb-ui-798667d648-wcsjq	1/1	Running	0	133m	app=yelb-ui,pod-template-hash=798667d648,tier=frontend

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	LABELS
service/redis-server	ClusterIP	10.99.236.225	<none>	6379/TCP	133m	app=redis-server,tier=cache
service/yelb-appserver	ClusterIP	10.96.144.180	<none>	4567/TCP	133m	app=yelb-appserver,tier=middletier
service/yelb-db	ClusterIP	10.99.4.117	<none>	5432/TCP	133m	app=yelb-db,tier=backenddb
service/yelb-ui	NodePort	10.104.19.5	<none>	80:31585/TCP	133m	app=yelb-ui,tier=frontend

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	LABELS
deployment.apps/mynginx	1/1	1	1	60m	app=mynginx
deployment.apps/redis-server	1/1	1	1	133m	<none>
deployment.apps/yelb-appserver	1/1	1	1	133m	<none>

```
deployment.apps/yelb-db      1/1   1      1      133m <none>
deployment.apps/yelb-ui      1/1   1      1      133m <none>
```

```
NAME                                DESIRED  CURRENT  READY  AGE   LABELS
replicaset.apps/mynginx-5b686ccd46    1        1        1     60m  app=mynginx,pod-template-hash=5b686ccd46
replicaset.apps/redis-server-74556bbcb7 1        1        1    133m  app=redis-server,pod-template-hash=74556bbcb7,tier=cache
replicaset.apps/yelb-appserver-d584bb889 1        1        1    133m  app=yelb-appserver,pod-template-hash=d584bb889,tier=middletier
replicaset.apps/yelb-db-694586cd78    1        1        1    133m  app=yelb-db,pod-template-hash=694586cd78,tier=backenddb
replicaset.apps/yelb-ui-798667d648    1        1        1    133m  app=yelb-ui,pod-template-hash=798667d648,tier=frontend
```

```
[student@master lab3]$ kubectl get all -l lab=pod
```

```
NAME                READY  STATUS   RESTARTS  AGE
pod/httpd-from-file 1/1    Running  0         97m
pod/nowy-httpd      1/1    Running  0         92m
```

```
[student@master lab3]$ kubectl delete all -l lab=pod
```

```
pod "httpd-from-file" deleted
pod "nowy-httpd" deleted
```

Przykład bardziej zaawansowanego użycia etykiet:

```
[student@master lab3]$ kubectl get pods -l 'tier in (frontend, backenddb)'
```

```
NAME                READY  STATUS   RESTARTS  AGE
yelb-db-694586cd78-86db6 1/1    Running  0         137m
yelb-ui-798667d648-wcsjq 1/1    Running  0         137m
```

Task 7: Domyślne zachowanie podów (...i nie tylko)

Stwórz plik deklaracyjny opisujący typowego poda:

```
[student@master lab3]$ cat restart-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
```

```
name: first-pod
spec:
  containers:
  - name: container1
    image: busybox
    command: ['sh', '-c', 'sleep 20']
    imagePullPolicy: IfNotPresent
```

Uruchom poda na podstawie yamla i sprawdź jego zachowanie

```
[student@master lab3]$ kubectl apply -f restart-pod.yml
```

pod/first-pod created

```
[student@master lab3]$ watch kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
first-pod	0/1	Completed	3	2m7s
httpd-from-file	1/1	Running	0	83m
mynginx-5b686ccd46-q5z7d	1/1	Running	0	48m
nowy-httpd	1/1	Running	0	79m
redis-server-74556bbcb7-bncdm	1/1	Running	0	121m
yelb-appserver-d584bb889-xh95q	1/1	Running	0	121m
yelb-db-694586cd78-86db6	1/1	Running	0	121m
yelb-ui-798667d648-wcsjq	1/1	Running	0	121m

Sprawdź dokumentację Kubernetesa i spróbuj znaleźć opcję restartPolicy. Sprawdź jaka opcja jest domyślna i jakie są możliwości.

```
[student@master lab3]$ kubectl explain pod.spec
```

```
[student@master lab3]$ kubectl explain pod.spec.restartPolicy
```

```
KIND: Pod
VERSION: v1
```

```
FIELD: restartPolicy <string>
```

DESCRIPTION:

Restart policy for all containers within the pod. One of Always, OnFailure, Never. Default to Always. More info:
<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy>

Zmodyfikuj kod poda i uruchom go z inną nazwą. Sprawdź jak zachowa się z opcją restartPolicy: Never

```
[student@master lab3]$ vim restart-pod.yml
```

```
[student@master lab3]$ cat restart-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: second-pod
spec:
  containers:
  - name: container1
    image: busybox
    imagePullPolicy: IfNotPresent
    command: ['sh', '-c', 'sleep 20']
  restartPolicy: Never
```

```
[student@master lab3]$ kubectl apply -f restart-pod.yml
```

Sprawdź teraz zachowanie poda:

```
[student@master lab3]$ watch kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
first-pod	0/1	CrashLoopBackOff	6	9m54s
httpd-from-file	1/1	Running	0	91m
mynginx-5b686ccd46-q5z7d	1/1	Running	0	56m
nowy-httpd	1/1	Running	0	87m
redis-server-74556bbcb7-bncdm	1/1	Running	0	129m
second-pod	0/1	Completed	0	75s
yelb-appserver-d584bb889-xh95q	1/1	Running	0	129m
yelb-db-694586cd78-86db6	1/1	Running	0	129m
yelb-ui-798667d648-wcsjq	1/1	Running	0	129m

Lab 4: Zapoznanie z obiektem deployment

W celu wygodniejszej pracy z plikami yaml możesz zmodyfikować zachowanie vima ustawiając tabulację na 2 znaki spacji (zgodnie ze specyfikacją języka yaml):

```
[student@master ~]$ pwd
```

```
/home/student
```

```
[student@master ~]$ echo "autocmd FileType yaml setlocal ai ts=2 sw=2 et" > .vimrc
```

Czynności:

- Jak utworzyć deployment
- Jak sprawdzić logi i zdarzenia obiektu deployment
- Jak zmienić podstawowe parametry deploymentów

Task 1. Stworzenie obiektu Deployment (a także ReplicaSet i Pod)

uwaga - jedna linia

```
[student@master k8s]$ kubectl create deployment www --image=httpd --dry-run=client -o yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: www
  name: www
spec:
  replicas: 1
  selector:
    matchLabels:
      app: www
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: www
    spec:
      containers:
      - image: httpd
        name: httpd
        resources: {}
status: {}
```

```
[student@master k8s]$
```


uwaga - jedna linia

```
[student@master k8s]$ kubectl create deployment www --image=httpd --dry-run=client -o yaml > deploy.yml
```

Dodaj w definicji kontenera imagePullPolicy: IfNotPresent

```
[student@master k8s]$ kubectl apply -f deploy.yml
```

deployment.apps/www created

```
[student@master k8s]$ kubectl get deployments.apps
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
mynginx	1/1	1	1	3h1m
redis-server	1/1	1	1	4h14m
www	1/1	1	1	79s
yelb-appserver	1/1	1	1	4h14m
yelb-db	1/1	1	1	4h14m
yelb-ui	1/1	1	1	4h14m

```
[student@master k8s]$ kubectl describe deployments.apps www
```

Name: www
Namespace: lab1
CreationTimestamp: Tue, 06 Jul 2021 23:12:59 +0200
Labels: app=www
Annotations: deployment.kubernetes.io/revision: 1
Selector: app=www
Replicas: 1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
Labels: app=www
Containers:
httpd:
Image: httpd
Port: <none>
Host Port: <none>
Environment: <none>
Mounts: <none>
Volumes: <none>
Conditions:
Type Status Reason

Available True MinimumReplicasAvailable
Progressing True NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet: www-6d9f88f6b5 (1/1 replicas created)
Events:
Type Reason Age From Message

Normal ScalingReplicaSet 96s deployment-controller Scaled up replica set www-6d9f88f6b5 to 1

```
[student@master ~]$ kubectl get replicaset.apps
```

NAME	DESIRED	CURRENT	READY	AGE
mynginx-5b686ccd46	1	1	1	3h3m
redis-server-74556bbcb7	1	1	1	4h16m
www-6d9f88f6b5	1	1	1	3m37s
yelb-appserver-d584bb889	1	1	1	4h16m
yelb-db-694586cd78	1	1	1	4h16m
yelb-ui-798667d648	1	1	1	4h16m

```
[student@master k8s]$ kubectl get rs
```

NAME	DESIRED	CURRENT	READY	AGE
mynginx-5b686ccd46	1	1	1	3h3m
redis-server-74556bbcb7	1	1	1	4h16m
www-6d9f88f6b5	1	1	1	3m37s
yelb-appserver-d584bb889	1	1	1	4h16m
yelb-db-694586cd78	1	1	1	4h16m
yelb-ui-798667d648	1	1	1	4h16m

```
[student@master k8s]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
mynginx-5b686ccd46-q5z7d	1/1	Running	0	3h3m
podwojny	2/2	Running	9	26m
redis-server-74556bbcb7-bncdm	1/1	Running	0	4h16m
second-pod	0/1	Completed	0	128m
www-6d9f88f6b5-2vvzq	1/1	Running	0	3m45s
yelb-appserver-d584bb889-xh95q	1/1	Running	0	4h16m
yelb-db-694586cd78-86db6	1/1	Running	0	4h16m
yelb-ui-798667d648-wcsjq	1/1	Running	0	4h16m

```
[student@master k8s]$ kubectl get all -l app=www
```

NAME	READY	STATUS	RESTARTS	AGE
pod/www-6d9f88f6b5-2vvzq	1/1	Running	0	5m17s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www	1/1	1	1	5m17s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www-6d9f88f6b5	1	1	1	5m17s

Task 2. Skalowanie obiektu Deployment 2 metodami

Wykonaj skalowanie ilości podów w danym deploymentie za pomocą kubectl scale:

```
[student@master k8s]$ kubectl scale --replicas=3 deployment/www
```

deployment.apps/www scaled

```
[student@master k8s]$ kubectl describe deployments.apps www
```

```
Name:          www
Namespace:     lab1
CreationTimestamp: Tue, 06 Jul 2021 23:12:59 +0200
Labels:        app=www
Annotations:    deployment.kubernetes.io/revision: 1
Selector:      app=www
Replicas:      3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=www
  Containers:
    httpd:
      Image:      httpd
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:      <none>
      Volumes:     <none>
  Conditions:
    Type      Status Reason
    ----      -
    Progressing True  NewReplicaSetAvailable
    Available  True  MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet:  www-6d9f88f6b5 (3/3 replicas created)
Events:
  Type    Reason      Age    From          Message
  ----    -
  Normal  ScalingReplicaSet  7m40s  deployment-controller  Scaled up replica set www-6d9f88f6b5 to 1
  Normal  ScalingReplicaSet  10s    deployment-controller  Scaled up replica set www-6d9f88f6b5 to 3
```

```
[student@master k8s]$ kubectl get all -l app=www
```

NAME	READY	STATUS	RESTARTS	AGE
pod/www-6d9f88f6b5-257n4	1/1	Running	0	2m14s
pod/www-6d9f88f6b5-2vvzq	1/1	Running	0	9m44s
pod/www-6d9f88f6b5-qpm6g	1/1	Running	0	2m14s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www	3/3	3	3	9m45s

NAME	DESIRED	CURRENT	READY	AGE
------	---------	---------	-------	-----

```
replicaset.apps/www-6d9f88f6b5 3 3 3 9m45s
```

```
[student@master k8s]$ kubectl describe rs www-6d9f88f6b5
```

```
Name:      www-6d9f88f6b5
```

```
Namespace: lab1
```

```
Selector:   app=www,pod-template-hash=6d9f88f6b5
```

```
Labels:     app=www
```

```
            pod-template-hash=6d9f88f6b5
```

```
Annotations: deployment.kubernetes.io/desired-replicas: 3
```

```
              deployment.kubernetes.io/max-replicas: 4
```

```
              deployment.kubernetes.io/revision: 1
```

```
Controlled By: Deployment/www
```

```
Replicas:   3 current / 3 desired
```

```
Pods Status: 3 Running / 0 Waiting / 0 Succeeded / 0 Failed
```

```
Pod Template:
```

```
  Labels: app=www
```

```
          pod-template-hash=6d9f88f6b5
```

```
Containers:
```

```
  httpd:
```

```
    Image:      httpd
```

```
    Port:       <none>
```

```
    Host Port:  <none>
```

```
    Environment: <none>
```

```
    Mounts:      <none>
```

```
    Volumes:      <none>
```

```
Events:
```

```
  Type    Reason      Age   From          Message
```

```
-----
```

```
Normal SuccessfulCreate 11m   replicaset-controller Created pod: www-6d9f88f6b5-2vvzq
```

```
Normal SuccessfulCreate 3m48s replicaset-controller Created pod: www-6d9f88f6b5-257n4
```

```
Normal SuccessfulCreate 3m48s replicaset-controller Created pod: www-6d9f88f6b5-qpm6g
```

Wykonaj skołowanie ilości podów w deploymentie za pomocą edycji pliku yaml:

```
[student@master k8s]$ vim deploy.yml
```

```
[student@master k8s]$ cat deploy.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  creationTimestamp: null
```

```
  labels:
```

```
    app: www
```

```
    name: www
```

```
spec:
```

```
  replicas: 5
```

```
  selector:
```

```
    matchLabels:
```

```
      app: www
```

```
  strategy: {}
```

```

template:
  metadata:
    creationTimestamp: null
  labels:
    app: www
  spec:
    containers:
    - image: httpd
      name: httpd
      imagePullPolicy: IfNotPresent
      resources: {}
status: {}

```

[student@master k8s]\$ kubectl get pods -l app=www

NAME	READY	STATUS	RESTARTS	AGE
www-6d9f88f6b5-257n4	1/1	Running	0	18m
www-6d9f88f6b5-2vvzq	1/1	Running	0	26m
www-6d9f88f6b5-qpm6g	1/1	Running	0	18m

[student@master k8s]\$ kubectl apply -f deploy.yml

deployment.apps/www configured

[student@master k8s]\$ kubectl get pods -l app=www

NAME	READY	STATUS	RESTARTS	AGE
www-6d9f88f6b5-257n4	1/1	Running	0	18m
www-6d9f88f6b5-2vvzq	1/1	Running	0	26m
www-6d9f88f6b5-qpm6g	1/1	Running	0	18m
www-6d9f88f6b5-vqqp7	0/1	ContainerCreating	0	1s
www-6d9f88f6b5-vqx2h	0/1	ContainerCreating	0	1s

[student@master ~]\$ kubectl get pods -l app=www

NAME	READY	STATUS	RESTARTS	AGE
www-6d9f88f6b5-257n4	1/1	Running	0	30m
www-6d9f88f6b5-2vvzq	1/1	Running	0	38m
www-6d9f88f6b5-qpm6g	1/1	Running	0	30m
www-6d9f88f6b5-vqqp7	1/1	Running	0	11m
www-6d9f88f6b5-vqx2h	1/1	Running	0	11m

Task 3. Deployment – strategia RollingUpdate

Sprawdź jakie są opcje aktualizowania deploymentu:

[student@master k8s]\$ kubectl explain deployment.spec.strategy

KIND: Deployment

VERSION: apps/v1

RESOURCE: strategy <Object>

DESCRIPTION:

The deployment strategy to use to replace existing pods with new ones.

DeploymentStrategy describes how to replace existing pods with new ones.

FIELDS:

rollingUpdate <Object>

Rolling update config params. Present only if DeploymentStrategyType = RollingUpdate.

type <string>

Type of deployment. Can be "Recreate" or "RollingUpdate". Default is

RollingUpdate.

uwaga - jedna linia

[student@master k8s]\$ kubectl explain deployment.spec.strategy.rollingUpdate

KIND: Deployment

VERSION: apps/v1

RESOURCE: rollingUpdate <Object>

DESCRIPTION:

Rolling update config params. Present only if DeploymentStrategyType = RollingUpdate.

Spec to control the desired behavior of rolling update.

FIELDS:

maxSurge <string>

The maximum number of pods that can be scheduled above the desired number of pods. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 25%. Example: when this is set to 30%, the new ReplicaSet can be scaled up immediately when the rolling update starts, such that the total number of old and new pods do not exceed 130% of desired pods. Once old pods have been killed, new ReplicaSet can be scaled up further, ensuring that total number of pods running at any time during the update is at most 130% of desired pods.

maxUnavailable <string>

The maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). Absolute number is calculated from percentage by rounding down. This can not be 0 if MaxSurge is 0. Defaults to 25%. Example: when this is set to 30%, the old ReplicaSet can be scaled down to 70% of desired pods

immediately when the rolling update starts. Once new pods are ready, old ReplicaSet can be scaled down further, followed by scaling up the new ReplicaSet, ensuring that the total number of pods available at all times during the update is at least 70% of desired pods.

Dokonaj odpowiednich zmian w pliku deploymentu (sekcja strategy), a następnie zastosuj zmiany:

```
[student@master k8s]$ vim deploy.yml
```

```
[student@master k8s]$ cat deploy.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: www
  name: www
spec:
  replicas: 5
  selector:
    matchLabels:
      app: www
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: www
    spec:
      containers:
      - image: httpd:2.4
        name: httpd
        imagePullPolicy: IfNotPresent
        resources: {}
  status: {}
```

```
[student@master ~]$ kubectl apply -f deploy.yml
deployment.apps/www configured
```

```
[student@master k8s]$ kubectl get pods -l app=www
```

NAME	READY	STATUS	RESTARTS	AGE
www-6d9f88f6b5-257n4	0/1	Terminating	0	63m
www-6d9f88f6b5-2vvzq	0/1	Terminating	0	71m
www-6d9f88f6b5-qpm6g	0/1	Terminating	0	63m

www-6d9f88f6b5-vqx2h	0/1	Terminating	0	45m
www-7dc75659f-bw8xf	1/1	Running	0	11s
www-7dc75659f-d8gbn	1/1	Running	0	17s
www-7dc75659f-jzrd5	1/1	Running	0	17s
www-7dc75659f-kb6wv	1/1	Running	0	13s
www-7dc75659f-wj5qv	1/1	Running	0	17s

[student@master k8s]\$ **kubectl get pods -l app=www**

NAME	READY	STATUS	RESTARTS	AGE
www-7dc75659f-bw8xf	1/1	Running	0	2m31s
www-7dc75659f-d8gbn	1/1	Running	0	2m37s
www-7dc75659f-jzrd5	1/1	Running	0	2m37s
www-7dc75659f-kb6wv	1/1	Running	0	2m33s
www-7dc75659f-wj5qv	1/1	Running	0	2m37s

Task 4. Deployment – strategia Recreate

Dokonaj odpowiednich zmian w pliku deploymentu (sekcja strategy), a następnie zastosuj zmiany:

```
[student@master k8s]$ vim deploy.yml
```

```
[student@master k8s]$ cat deploy.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: www
  name: www
spec:
  replicas: 5
  selector:
    matchLabels:
      app: www
  strategy:
    type: Recreate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: www
    spec:
      containers:
      - image: httpd
        name: httpd
        imagePullPolicy: IfNotPresent
      resources: {}
  status: {}
```

Sprawdź jak teraz zachowuje się dokonywanie zmian w deploymentcie:

```
[student@master ~]$ kubectl apply -f deploy.yml
deployment.apps/www configured
```

Zachodzące zmiany możesz śledzić za pomocą komendy:

```
[student@master ~]$ watch kubectl get pods -l app=www
```

lub kilka razy uruchamiając:

```
[student@master ~]$ kubectl get pods -l app=www
```

NAME	READY	STATUS	RESTARTS	AGE
www-7dc75659f-gbfkm	0/1	Terminating	0	3m8s
www-7dc75659f-hklsf	0/1	Terminating	0	3m12s
www-7dc75659f-j4w7m	0/1	Terminating	0	3m12s
www-7dc75659f-mhwsz	0/1	Terminating	0	3m12s
www-7dc75659f-sm89w	0/1	Terminating	0	3m5s

[student@master ~]\$ **kubectl get pods -l app=www**

NAME	READY	STATUS	RESTARTS	AGE
www-6d9f88f6b5-55vlc	1/1	Running	0	9s
www-6d9f88f6b5-56j8h	0/1	ContainerCreating	0	9s
www-6d9f88f6b5-gchf4	0/1	ContainerCreating	0	9s
www-6d9f88f6b5-qhrv6	1/1	Running	0	9s
www-6d9f88f6b5-wg7s9	1/1	Running	0	9s

Task 5. Deployment – historia zmian

```
[student@master ~]$ kubectl rollout history deployment
```

```
deployment.apps/www
REVISION CHANGE-CAUSE
2      <none>
3      <none>
```

```
[student@master ~]$ kubectl edit deployments.apps www
```

```
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "3"
    kubectl.kubernetes.io/last-applied-configuration: |
{"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{"creationTimestamp":null,"labels":{"app":"www"},"name":"www","namespace":"default"},"spec":{"replicas":5,"selector":{"matchLabels":{"app":"www"},"strategy":{"type":"Recreate"},"template":{"metadata":{"creationTimestamp":null,"labels":{"app":"www"},"spec":{"containers":[{"image":"httpd","name":"httpd"},"resources":{}}}}},"status":{"creationTimestamp":"2021-07-07T17:40:51Z"
generation: 5
labels:
  app: www
  name: www
  namespace: default
  resourceVersion: "60569"
  uid: f8f2bac9-fb3c-4708-adaa-710556704195
spec:
  progressDeadlineSeconds: 600
  replicas: 5
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: www
  strategy:
    type: Recreate
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: www
spec:
  containers:
    - image: httpd:2.4
      imagePullPolicy: IfNotPresent
      name: httpd
      resources: {}
      terminationMessagePath: /dev/termination-log
```

```
    terminationMessagePolicy: File
    dnsPolicy: ClusterFirst
    restartPolicy: Always
    schedulerName: default-scheduler
    securityContext: {}
    terminationGracePeriodSeconds: 30
status:
  availableReplicas: 5
  conditions:
  - lastTransitionTime: "2021-07-07T17:40:51Z"
    lastUpdateTime: "2021-07-07T17:52:43Z"
    message: ReplicaSet "www-7dc75659f" has successfully progressed.
    reason: NewReplicaSetAvailable
    status: "True"
    type: Progressing
  - lastTransitionTime: "2021-07-07T17:56:00Z"
    lastUpdateTime: "2021-07-07T17:56:00Z"
    message: Deployment has minimum availability.
    reason: MinimumReplicasAvailable
    status: "True"
    type: Available
  observedGeneration: 5
  readyReplicas: 5
  replicas: 5
  updatedReplicas: 5
```

deployment.apps/www edited

[student@master ~]\$ kubectl rollout history deployment

```
deployment.apps/www
REVISION  CHANGE-CAUSE
3         <none>
4         <none>
```

[student@master ~]\$ kubectl get rs

NAME	DESIRED	CURRENT	READY	AGE
www-6d9f88f6b5	0	0	0	43m
www-7dc75659f	5	5	5	31m

[student@master ~]\$ kubectl describe deployments.apps www

```
Name:          www
Namespace:     default
CreationTimestamp: Wed, 07 Jul 2021 19:40:51 +0200
Labels:       app=www
Annotations:  deployment.kubernetes.io/revision: 4
Selector:     app=www
Replicas:     5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType: Recreate
MinReadySeconds: 0
Pod Template:
  Labels: app=www
```

Containers:

httpd:

Image: httpd:2.4

Port: <none>

Host Port: <none>

Environment: <none>

Mounts: <none>

Volumes: <none>

Conditions:

Type	Status	Reason
------	--------	--------

Progressing	True	NewReplicaSetAvailable
-------------	------	------------------------

Available	True	MinimumReplicasAvailable
-----------	------	--------------------------

OldReplicaSets: <none>

NewReplicaSet: www-7dc75659f (5/5 replicas created)

Events:

Type	Reason	Age	From	Message
Normal	ScalingReplicaSet	45m	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 1
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 3
Normal	ScalingReplicaSet	33m	deployment-controller	Scaled up replica set www-7dc75659f to 2
Normal	ScalingReplicaSet	33m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 4
Normal	ScalingReplicaSet	33m	deployment-controller	Scaled up replica set www-7dc75659f to 3
Normal	ScalingReplicaSet	33m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 3
Normal	ScalingReplicaSet	33m	deployment-controller	Scaled up replica set www-7dc75659f to 4
Normal	ScalingReplicaSet	33m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 2
Normal	ScalingReplicaSet	30m (x4 over 33m)	deployment-controller	(combined from similar events): Scaled down replica set www-7dc75659f to 0
Normal	ScalingReplicaSet	30m (x2 over 38m)	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 5
Normal	ScalingReplicaSet	5m24s	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 0
Normal	ScalingReplicaSet	5m16s	deployment-controller	Scaled up replica set www-7dc75659f to 5

[student@master ~]\$ kubectl rollout history deployment www

deployment.apps/www

REVISION CHANGE-CAUSE

3 <none>

4 <none>

[student@master ~]\$ kubectl rollout history deployment www --revision=3

deployment.apps/www with revision #3

Pod Template:

Labels: app=www

pod-template-hash=6d9f88f6b5

Containers:

httpd:

Image: httpd

Port: <none>

Host Port: <none>

Environment: <none>

Mounts: <none>

Volumes: <none>

```
[student@master ~]$ kubectl rollout history deployment www --revision=4
```

deployment.apps/www with revision #4

Pod Template:

Labels: app=www

pod-template-hash=7dc75659f

Containers:

httpd:

Image: httpd:2.4

Port: <none>

Host Port: <none>

Environment: <none>

Mounts: <none>

Volumes: <none>

```
[student@master ~]$ kubectl describe deployments.apps www
```

Name: www

Namespace: default

CreationTimestamp: Wed, 07 Jul 2021 19:40:51 +0200

Labels: app=www

Annotations: deployment.kubernetes.io/revision: 4

Selector: app=www

Replicas: 5 desired | 5 updated | 5 total | 5 available | 0 unavailable

StrategyType: Recreate

MinReadySeconds: 0

Pod Template:

Labels: app=www

Containers:

httpd:

Image: httpd:2.4

Port: <none>

Host Port: <none>

Environment: <none>

Mounts: <none>

Volumes: <none>

Conditions:

Type	Status	Reason
------	--------	--------

----	-----	-----
------	-------	-------

Progressing	True	NewReplicaSetAvailable
-------------	------	------------------------

Available	True	MinimumReplicasAvailable
-----------	------	--------------------------

OldReplicaSets: <none>

NewReplicaSet: www-7dc75659f (5/5 replicas created)

Events:

Type	Reason	Age	From	Message
------	--------	-----	------	---------

----	-----	----	-----	-----
------	-------	------	-------	-------

Normal	ScalingReplicaSet	52m	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 1
Normal	ScalingReplicaSet	47m	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 3
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled up replica set www-7dc75659f to 2
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 4
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled up replica set www-7dc75659f to 3
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 3
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled up replica set www-7dc75659f to 4
Normal	ScalingReplicaSet	40m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 2

Normal ScalingReplicaSet 37m (x4 over 40m) deployment-controller (combined from similar events): Scaled down replica set www-7dc75659f to 0

Normal ScalingReplicaSet 37m (x2 over 45m) deployment-controller Scaled up replica set www-6d9f88f6b5 to 5

Normal ScalingReplicaSet 12m deployment-controller Scaled down replica set www-6d9f88f6b5 to 0

Normal ScalingReplicaSet 12m deployment-controller Scaled up replica set www-7dc75659f to 5

[student@master ~]\$ kubectl rollout undo deployment www --to-revision=3

deployment.apps/www rolled back

[student@master ~]\$ kubectl describe deployments.apps www

Name: www
Namespace: default
CreationTimestamp: Wed, 07 Jul 2021 19:40:51 +0200
Labels: app=www
Annotations: deployment.kubernetes.io/revision: 5
Selector: app=www
Replicas: 5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType: Recreate
MinReadySeconds: 0
Pod Template:
Labels: app=www
Containers:
httpd:
Image: httpd
Port: <none>
Host Port: <none>
Environment: <none>
Mounts: <none>
Volumes: <none>
Conditions:
Type Status Reason

Progressing True NewReplicaSetAvailable
Available True MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet: www-6d9f88f6b5 (5/5 replicas created)

Events:

Type	Reason	Age	From	Message
Normal	ScalingReplicaSet	55m	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 1
Normal	ScalingReplicaSet	50m	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 3
Normal	ScalingReplicaSet	43m	deployment-controller	Scaled up replica set www-7dc75659f to 2
Normal	ScalingReplicaSet	43m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 4
Normal	ScalingReplicaSet	43m	deployment-controller	Scaled up replica set www-7dc75659f to 3
Normal	ScalingReplicaSet	43m	deployment-controller	Scaled up replica set www-7dc75659f to 4
Normal	ScalingReplicaSet	43m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 3
Normal	ScalingReplicaSet	43m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 2
Normal	ScalingReplicaSet	40m (x4 over 43m)	deployment-controller	(combined from similar events): Scaled down replica set www-7dc75659f to 0
Normal	ScalingReplicaSet	14m	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 0
Normal	ScalingReplicaSet	14m	deployment-controller	Scaled up replica set www-7dc75659f to 5
Normal	ScalingReplicaSet	34s	deployment-controller	Scaled down replica set www-7dc75659f to 0
Normal	ScalingReplicaSet	18s (x3 over 48m)	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 5

```
[student@master ~]$ kubectl rollout history deployment www
```

```
deployment.apps/www  
REVISION  CHANGE-CAUSE  
4         <none>  
5         <none>
```

```
[student@master ~]$ kubectl set image deployment/www httpd=httpd:2.4
```

```
deployment.apps/www image updated
```

```
[student@master ~]$ kubectl rollout history deployment www
```

```
deployment.apps/www  
REVISION  CHANGE-CAUSE  
5         <none>  
6         <none>
```


Task 6. Deployment – historia zmian część 2

Historię zmian we wdrożeniu aplikacji można nieco ulepszyć:

```
[student@master ~]$ kubectl describe deployments.apps www | grep -i image
```

```
Image:      httpd:2.4
```

uwaga - jedna linia

```
[student@master ~]$ kubectl set image deployment/www httpd=httpd --record=true
```

```
deployment.apps/www image updated
```

```
[student@master ~]$ kubectl rollout history deployment www
```

```
deployment.apps/www
REVISION  CHANGE-CAUSE
6         <none>
```

```
7         kubectl set image deployment/www httpd=httpd --record=true
```

```
[student@master ~]$ kubectl describe deployments.apps www | grep -i image
```

```
      kubernetes.io/change-cause: kubectl set image deployment/www httpd=httpd --record=true
Image:      httpd
```

Dokonaj kolejnych zmian w obrazie:

```
[student@master ~]$ vim deploy.yml
```

```
[student@master ~]$ cat deploy.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: www
  name: www
spec:
  replicas: 5
  selector:
    matchLabels:
      app: www
  strategy:
    type: Recreate
  template:
```

```
metadata:
  creationTimestamp: null
  labels:
    app: www
spec:
  containers:
  - image: httpd:2.4
    name: httpd
    imagePullPolicy: IfNotPresent

  resources: {}
status: {}
```

```
[student@master ~]$ kubectl apply -f deploy.yml --record=true
```

deployment.apps/www configured

```
[student@master ~]$ kubectl rollout history deployment www
```

deployment.apps/www

REVISION CHANGE-CAUSE

7 kubectl set image deployment/www httpd=httpd --record=true

8 kubectl apply --filename=deploy.yml --record=true

```
[student@master ~]$ kubectl describe deployments.apps www
```

```
Name:          www
Namespace:     default
CreationTimestamp: Wed, 07 Jul 2021 19:40:51 +0200
Labels:        app=www
Annotations:    deployment.kubernetes.io/revision: 8
                kubernetes.io/change-cause: kubectl apply --filename=deploy.yml --record=true
Selector:      app=www
Replicas:      5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:   Recreate
MinReadySeconds: 0
Pod Template:
  Labels: app=www
  Containers:
    httpd:
      Image:      httpd:2.4
      Port:       <none>
      Host Port:   <none>
      Environment: <none>
      Mounts:      <none>
      Volumes:     <none>
Conditions:
  Type           Status  Reason
  ----           -
  Progressing    True    NewReplicaSetAvailable
  Available      True    MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet:  www-7dc75659f (5/5 replicas created)
```

Events:

Type	Reason	Age	From	Message
----	-----	-----	-----	-----
Normal	ScalingReplicaSet	22m (x2 over 36m)	deployment-controller	Scaled down replica set www-7dc75659f to 0
Normal	ScalingReplicaSet	21m (x4 over 83m)	deployment-controller	Scaled up replica set www-6d9f88f6b5 to 5
Normal	ScalingReplicaSet	53s (x3 over 50m)	deployment-controller	Scaled down replica set www-6d9f88f6b5 to 0
Normal	ScalingReplicaSet	39s (x3 over 50m)	deployment-controller	Scaled up replica set www-7dc75659f to 5

Task 7. Deployment – struktura

```
[student@master ~]$ kubectl get deployments.apps
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
www	5/5	5	5	153m

```
[student@master ~]$ kubectl get deployments.apps --show-labels
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	LABELS
www	5/5	5	5	153m	app=www

```
[student@master ~]$ kubectl get all -l app=www
```

NAME	READY	STATUS	RESTARTS	AGE
pod/www-7dc75659f-6c7mn	1/1	Running	0	63m
pod/www-7dc75659f-g9q74	1/1	Running	0	63m
pod/www-7dc75659f-hlv7p	1/1	Running	0	63m
pod/www-7dc75659f-jzrpr	1/1	Running	0	63m
pod/www-7dc75659f-wl2jv	1/1	Running	0	63m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www	5/5	5	5	153m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www-6d9f88f6b5	0	0	0	153m
replicaset.apps/www-7dc75659f	5	5	5	142m

```
[student@master ~]$ kubectl delete pod/www-7dc75659f-6c7mn
```

```
pod "www-7dc75659f-6c7mn" deleted
```

```
[student@master ~]$ kubectl get all -l app=www
```

NAME	READY	STATUS	RESTARTS	AGE
pod/www-7dc75659f-g9q74	1/1	Running	0	67m
pod/www-7dc75659f-hlv7p	1/1	Running	0	67m
pod/www-7dc75659f-jzrpr	1/1	Running	0	67m
pod/www-7dc75659f-vr86w	1/1	Running	0	42s
pod/www-7dc75659f-wl2jv	1/1	Running	0	67m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www	5/5	5	5	157m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www-6d9f88f6b5	0	0	0	157m
replicaset.apps/www-7dc75659f	5	5	5	145m

```
[student@master ~]$ kubectl get rs -l app=www
```

NAME	DESIRED	CURRENT	READY	AGE
------	---------	---------	-------	-----

www-6d9f88f6b5	0	0	0	164m
www-7dc75659f	5	5	5	152m

[student@master ~]\$ **kubectl get all -l app=www**

NAME	READY	STATUS	RESTARTS	AGE
pod/www-7dc75659f-g9q74	1/1	Running	0	74m
pod/www-7dc75659f-hlv7p	1/1	Running	0	74m
pod/www-7dc75659f-jzrpr	1/1	Running	0	74m
pod/www-7dc75659f-vr86w	1/1	Running	0	7m41s
pod/www-7dc75659f-wl2jv	1/1	Running	0	74m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www	5/5	5	5	164m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www-6d9f88f6b5	0	0	0	164m
replicaset.apps/www-7dc75659f	5	5	5	152m

[student@master ~]\$ **kubectl delete rs www-7dc75659f**

replicaset.apps "www-7dc75659f" deleted

[student@master ~]\$ **kubectl get rs -l app=www**

NAME	DESIRED	CURRENT	READY	AGE
www-6d9f88f6b5	0	0	0	165m
www-7dc75659f	5	5	0	4s

[student@master ~]\$ **kubectl get all -l app=www**

NAME	READY	STATUS	RESTARTS	AGE
pod/www-7dc75659f-j5jf8	1/1	Running	0	11s
pod/www-7dc75659f-l9t79	1/1	Running	0	11s
pod/www-7dc75659f-r27cx	1/1	Running	0	11s
pod/www-7dc75659f-v8m2m	0/1	ContainerCreating	0	11s
pod/www-7dc75659f-xxdhj	1/1	Running	0	11s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www	4/5	5	4	165m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www-6d9f88f6b5	0	0	0	165m
replicaset.apps/www-7dc75659f	5	5	4	11s

Lab 6: Rozmieszczanie Podów na klastrze Kubernetes.

Cele:

- poznanie sposobów rozmieszczania Podów na poszczególnych nodach w klastrze.

Task 1: nodeName w specyfikacji Podów

Zdefiniuj nowy obiekt typu Pod z opcją nodeName:

```
[student@master ~]$ vim pod-nodename.yml  
[student@master ~]$ cat pod-nodename.yml
```

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: pod-nodename  
  labels:  
    app: v4  
spec:  
  containers:  
    - name: www  
      image: httpd  
      imagePullPolicy: IfNotPresent  
  nodeName: node1
```

```
[student@master ~]$ kubectl apply -f pod-nodename.yml
```

pod/pod-nodename created

```
[student@master ~]$ kubectl get po -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod-nodename	1/1	Running	0	8s	10.36.0.3	gdansk.domain1.local	<none>	<none>

Task 2: nodeSelector w specyfikacji Podów


```
pod-selector 1/1 Running 0 9s 10.36.0.2 node1 <none> <none>
```

Task 3: Affinity (lub anti-Affinity) w specyfikacji Podów

Zdefiniuj nowy obiekt typu Pod z opcją affinity, który podczas wdrażania na klaster będzie musiał wybrać nod za pomocą requiredDuringSchedulingIgnoredDuringExecution:

```
[student@master ~]$ vim pod-affinity1.yml
```

```
[student@master ~]$ cat pod-affinity1.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-affinity-required
  labels:
    app: v4
spec:
  containers:
    - name: www
      image: httpd
      imagePullPolicy: IfNotPresent
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: moc
                operator: In
            values:
              - full
```

```
[student@master ~]$ kubectl apply -f pod-affinity1.yml
```

```
pod/pod-affinity-required created
```

```
[student@master ~]$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod-affinity-required	1/1	Running	0	3s	10.36.0.1	node1	<none>	<none>
pod-nodename	1/1	Running	0	37m	10.36.0.3	node1	<none>	<none>

pod-selector 1/1 Running 0 39m 10.36.0.2 node1 <none> <none>

Task 4: Affinity (lub anti-Affinity) w specyfikacji Podów

Zdefiniuj nowy obiekt typu Pod z opcją affinity, który podczas wdrażania na klaster będzie musiał wybrać nod za pomocą preferredDuringSchedulingIgnoredDuringExecution:

```
[student@master ~]$ vim pod-affinity2.yml
```

```
[student@master ~]$ cat pod-affinity2.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-affinity-preferred-node1
  labels:
    app: v4
spec:
  containers:
    - name: www
      image: httpd
      imagePullPolicy: IfNotPresent
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            matchExpressions:
              - key: kubernetes.io/hostname
                operator: In
                values:
                  - node1
```

```
[student@master ~]$ kubectl apply -f pod-affinity2.yml
```

pod/pod-affinity-preferred-gdansk created

```
[student@master ~]$ kubectl get po -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod-affinity-required	1/1	Running	0	3s	10.36.0.1	gdansk.domain1.local	<none>	<none>
pod-affinity-preferred-gdansk	1/1	Running	0	82s	10.36.0.4	node1	<none>	<none>

Spróbuj zmienić wartość preferredDuringSchedulingIgnoredDuringExecution na nazwę hosta, który nie jest podłączony do klastra:

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-affinity-preferred-warszawa
  labels:
    app: v4
spec:
  containers:
    - name: www
      image: httpd
      imagePullPolicy: IfNotPresent
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            matchExpressions:
              - key: kubernetes.io/hostname
                operator: In
                values:
                  - warszawa.domain1.local
```

[student@master ~]\$ kubectl apply -f pod.yml

pod/pod-affinity-preferred-warszawa created

[student@master ~]\$ kubectl get po -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod-affinity-preferred	1/1	Running	0	4m18s	10.44.0.2	poznan.domain1.local	<none>	<none>
pod-affinity-preferred-warszawa	1/1	Running	0	3s	10.44.0.4	node2	<none>	<none>
pod-affinity-required	1/1	Running	0	16m	10.36.0.1	gdansk.domain1.local	<none>	<none>

Lab 7: Obiekty w Kubernetes

Cele:

- zaznajomienie z najczęściej używanymi obiektami
- przegląd dostępnych obiektów i ich opcjonalnych zachowań

Task 1: Obiekt Job

Sprawdź, które API jest potrzebne do zdefiniowania obiektu Job:

```
[student@master ~]$ kubectl api-resources | grep -i job
```

```
cronjobs    cj      batch/v1beta1    true    CronJob
jobs        batch/v1          true    Job
```

Sprawdź dostępne opcje dla obiektu Job:

```
[student@master ~]$ kubectl explain job.spec.template
```

KIND: Job

VERSION: batch/v1

RESOURCE: template <Object>

DESCRIPTION:

Describes the pod that will be created when executing a job. More info:

<https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/>

PodTemplateSpec describes the data a pod should have when created from a template

FIELDS:

metadata <Object>

Standard object's metadata. More info:

<https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>

spec <Object>

Specification of the desired behavior of the pod. More info:

<https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

```
[student@master ~]$ vim job.yml
```

```
[student@master ~]$ cat job.yml
```

```
apiVersion: batch/v1
```

```
kind: Job
```

```
metadata:
```

```
  name: testjob
```

```
spec:
```

```
  template:
```

```
    spec:
```

```
      containers:
```

```
        - name: testjob-pod
```

```
          image: busybox
```

```
            imagePullPolicy: IfNotPresent
```

```
            command: ['sh', '-c', 'sleep 30']
```

```
          restartPolicy: OnFailure
```

```
[student@master ~]$ kubectl apply -f job.yml
```

```
job.batch/testjob created
```

```
[student@master ~]$ kubectl get jobs
```

NAME	COMPLETIONS	DURATION	AGE
testjob	0/1	6s	6s

```
[student@master ~]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
2in1	1/2	CrashLoopBackOff	107	8h
initcont	1/1	Running	84	9h
lab-pod	1/1	Running	1	30h
mywww	1/1	Running	0	10h
pod1	1/1	Running	0	7h41m
pod2	1/1	Running	0	7h40m
testjob-tkm6q	1/1	Running	0	20s
ubuntu1	0/1	CrashLoopBackOff	126	10h
volcont	1/1	Running	59	8h
www-7dc75659f-j5jf8	1/1	Running	0	51m
www-7dc75659f-l9t79	1/1	Running	0	51m
www-7dc75659f-r27cx	1/1	Running	0	51m
www-7dc75659f-v8m2m	1/1	Running	0	51m
www-7dc75659f-xxdhj	1/1	Running	0	51m
www2	1/1	Running	0	9h

```
[student@master ~]$ kubectl get jobs
```

NAME	COMPLETIONS	DURATION	AGE
testjob	1/1	2m4s	7m27s

Sprawdź jeszcze raz status poda uruchomionego przez Joba po 30s odkąd wystartował. Jeśli Job zakończył się pomyślnie powinien mieć status Completed.

Sprawdź jak wygląda status Joba za pomocą komendy (szczególnie sekcja Events):

```
[student@master ~]$ kubectl describe job/testjob
```

Events:

Type	Reason	Age	From	Message
Normal	SuccessfulCreate	8m2s	job-controller	Created pod: testjob-kz5j2
Normal	Completed	7m38s	job-controller	Job completed

Skasuj obiekt Job. Dokonaj odpowiednich zmian w definicji Joba:

```
apiVersion: batch/v1
kind: Job
metadata:
  name: testjob
labels:
  jobgroup: joby
```

spec:

parallelism: 1 # ile replik joba wystartuje
completions: 1 # ile podow ma sie zakonczyc powodzeniem
activeDeadlineSeconds: 20 # ile sekund max moze trwac job

template:

spec:

containers:

- name: testjob-pod

image: busybox

imagePullPolicy: IfNotPresent

command: ["sh", "-c", "sleep 30"]

restartPolicy: Never

backoffLimit: 4 # ile razy moze sie wysypac

**Uruchom obiekt Job z pliku yaml. Sprawdź jeszcze raz stan obiektów Pods, Jobs.
Ponownie wykonaj komendę:**

[student@master ~]\$ **kubectl describe job/testjob**

Tym razem Job powinien się wykonać błędnie po przekroczeniu limitu czasu:

Events:

Type	Reason	Age	From	Message
----	-----	----	-----	-----
Normal	SuccessfulCreate	100s	job-controller	Created pod: job1-dzbhh
Normal	SuccessfulDelete	70s	job-controller	Deleted pod: job1-dzbhh
Warning	DeadlineExceeded	70s (x2 over 70s)	job-controller	Job was active longer than specified deadline

Task 2: Obiekt CronJob

Ustawiamy zadanie Kubernetesa tak, żeby wykonywało się co 1 minutę (zapis w opcji schedule jest zgodny ze specyfikacją crond systemów linuxowych).

```
[student@master ~]$ vim cronJob.yml
```

```
[student@master ~]$ cat cronJob.yml
```

```
apiVersion: batch/v1beta1
```

```
kind: CronJob
```

```
metadata:
```

```
  name: cron1
```

```
spec:
```

```
  schedule: "*/1 * * * *"
```

```
  jobTemplate:
```

```
    spec:
```

```
      template:
```

```
        spec:
```

```
          restartPolicy: Never
```

```
          containers:
```

```
            - name: cronjob-cont
```

```
              image: busybox
```

```
                imagePullPolicy: IfNotPresent
```

```
              command: ['sh', '-c', 'echo CronJobs w akcji']
```

```
[student@master ~]$ kubectl apply -f cronJob.yml
```

```
cronjob.batch/cron1 created
```

Uwaga: istnieje CronJob do wykonania, ale czy istnieje pod związany z tym obiektem ?

```
[student@master ~]$ kubectl get cronjobs
```

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
cron1	10 2 * * *	False	0	<none>	10s

[student@master ~]\$ **kubectl get pods**

NAME	READY	STATUS	RESTARTS	AGE
mywww	1/1	Running	0	11h
pod1	1/1	Running	0	8h
pod2	1/1	Running	0	8h
testjob-tkm6q	0/1	Completed	0	28m
www-7dc75659f-j5jf8	1/1	Running	0	79m
www-7dc75659f-l9t79	1/1	Running	0	79m
www-7dc75659f-r27cx	1/1	Running	0	79m
www-7dc75659f-v8m2m	1/1	Running	0	79m
www-7dc75659f-xxdhj	1/1	Running	0	79m
www2	1/1	Running	0	10h

Sprawdź po upływie 1 minuty czy pojawił się Pod związany z CronJobem (jego nazwa powinna się zaczynać od cron1-...).

Po pojawieniu się Pod sprawdź jego logi za pomocą polecenia:

kubectl logs pod/cron1-...

Po upływie następnych paru minut sprawdź ile jest obiektów Pod związanych z obiektem CronJob, który utworzyłeś. 1? 2? 3? 4? 5?

Za pomocą polecenia:

kubectl explain cronjobs.spec

odszukaj opcji, która odpowiada za ilość przechowywanych Podów związanych z obiektem CronJob.

Task 3: Obiekt Deployment w klastrze.

uwaga - jedna linia

```
[student@master ~]$ kubectl create deployment www1 --image=nginx --dry-run=client -o yaml > www1.yml
```

```
[student@master ~]$ cat www1.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  creationTimestamp: null
```

```
  labels:
```

```
    app: www1
```

```
  name: www1
```

```
spec:
```

```
  replicas: 1
```

```
  selector:
```

```
    matchLabels:
```

```
      app: www1
```

```
  strategy: {}
```

```
  template:
```

```
    metadata:
```

```
      creationTimestamp: null
```

```
      labels:
```

```
        app: www1
```

```
    spec:
```

```
      containers:
```

```
        - image: nginx
```

```
          name: nginx
```

```
          imagePullPolicy: IfNotPresent
```

```
          resources: {}
```

```
status: {}
```

```
[student@master ~]$ kubectl apply -f www1.yml
```

```
deployment.apps/www1 created
```

```
[student@master ~]$ kubectl get deploy
```

```
NAME READY UP-TO-DATE AVAILABLE AGE
www1 1/1 1 1 12s
```

```
[student@master ~]$ kubectl get deploy -o wide
```

```
NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR
www1 1/1 1 1 40s nginx nginx app=www1
```

```
[student@master ~]$ kubectl get pod -o wide
```

```
NAME          READY STATUS RESTARTS AGE IP      NODE          NOMINATED NODE READINESS GATES
demo1-hr6jk    1/1 Running 0      46m 10.44.0.1 poznan.domain1.local <none> <none>
demo1-tj78j    1/1 Running 0      46m 10.47.0.1 gdansk.domain1.local <none> <none>
www1-7dd78bb58-qb2qr 1/1 Running 0      52s 10.44.0.2 poznan.domain1.local <none>
<none>
```

```
[student@master ~]$ ssh root@poznan free -h
```

```
root@poznan's password:
```

```
total    used    free   shared buff/cache available
Mem:    3.8Gi  555Mi  1.7Gi   17Mi   1.6Gi   3.1Gi
Swap:   0B    0B    0B
```

```
[student@master ~]$ ssh root@gdansk free -h
```

root@gdansk's password:

	total	used	free	shared	buff/cache	available
Mem:	2.9Gi	520Mi	789Mi	17Mi	1.6Gi	2.1Gi
Swap:	0B	0B	0B			

```
[student@master ~]$ ssh root@gdansk cat /proc/cpuinfo | grep process
```

root@gdansk's password:

```
processor           : 0
processor           : 1
```

uwaga - jedna linia

```
[student@master ~]$ ssh root@poznan cat /proc/cpuinfo | grep
```

```
process
```

root@poznan's password:

```
processor           : 0
processor           : 1
```

```
[student@master ~]$ vim www1.yml
```

```
[student@master ~]$ cat www1.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  creationTimestamp: null
```

```
  labels:
```

```
    app: www1
```

```
  name: www1
```

```
spec:
```

```
  replicas: 2
```

```
  selector:
```

```
matchLabels:
  app: www1
strategy: {}
template:
  metadata:
    creationTimestamp: null
  labels:
    app: www1
  spec:
    containers:
      - image: nginx
        name: nginx
        imagePullPolicy: IfNotPresent
        resources: {}
    status: {}
```

```
[student@master ~]$ kubectl apply -f www1.yml
```

deployment.apps/www1 configured

```
[student@master ~]$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
demo1-hr6jk	1/1	Running	0	51m	10.44.0.1	node1	<none>	<none>		
demo1-tj78j	1/1	Running	0	51m	10.47.0.1	node1	<none>	<none>		
www1-7dd78bb58-9fjwc	1/1	Running	0	7s	10.44.0.3	node2	<none>	<none>		
www1-7dd78bb58-qb2qr	1/1	Running	0	5m24s	10.44.0.2	node2	<none>	<none>		

```
[student@master ~]$ vim www1.yml
```

```
[student@master ~]$ cat www1.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  creationTimestamp: null
```

```
  labels:
```

```
    app: www1
```

```
  name: www1
```

```
spec:
```

```
  replicas: 4
```

```
  selector:
```

```
    matchLabels:
```

```
      app: www1
```

```
  strategy: {}
```

```
  template:
```

```
    metadata:
```

```
      creationTimestamp: null
```

```
      labels:
```

```
        app: www1
```

```
    spec:
```

```
      containers:
```

```
        - image: nginx
```

```
          name: nginx
```

```
          imagePullPolicy: IfNotPresent
```

```
          resources: {}
```

```
status: {}
```

```
[student@master ~]$ kubectl apply -f www1.yml
```

deployment.apps/www1 configured

```
[student@master ~]$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
demo1-hr6jk	1/1	Running	0	83m	10.44.0.1	node2	<none>	<none>
demo1-tj78j	1/1	Running	0	83m	10.47.0.1	node1	<none>	<none>
www1-7dd78bb58-45pwb	0/1	ContainerCreating	0	3s		<none>	node2	<none>
www1-7dd78bb58-9f4w6	0/1	ContainerCreating	0	3s		<none>	node1	<none>
www1-7dd78bb58-9fjwc	1/1	Running	0	32m	10.44.0.3	node2	<none>	<none>
www1-7dd78bb58-qb2qr	1/1	Running	0	37m	10.44.0.2	node1	<none>	<none>

Task 4: Obiekt DaemonSet w klastrze.

```
[student@master ~]$ vim demo1.yml
```

```
[student@master ~]$ cat demo1.yml
```

apiVersion: apps/v1

kind: DaemonSet

metadata:

name: demo1

labels:

app: nginx

spec:

selector:

matchLabels:

app: nginx

template:

metadata:

name: demo1-pod

labels:

app: nginx

spec:

containers:

- name: demo1-pod-container

image: nginx

imagePullPolicy: IfNotPresent

```
[student@master ~]$ kubectl apply -f demo1.yml
```

```
[student@master ~]$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
demo1-hr6jk	1/1	Running	0	83m	10.44.0.1	poznan.domain1.local	<none>	<none>
demo1-tj78j	1/1	Running	0	83m	10.47.0.1	gdansk.domain1.local	<none>	<none>

```
[student@master ~]$ kubectl label nodes poznan.domain1.local cpu=4
```

node/poznan.domain1.local labeled

```
[student@master ~]$ kubectl get nodes --show-labels
```

NAME	STATUS	ROLES	AGE	VERSION	LABELS
------	--------	-------	-----	---------	--------

```
gdansk.domain1.local Ready <none> 47m v1.21.2
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=gdansk.domain1.local,kubernetes.io/os=linux

katowice.domain1.local Ready control-plane,master 79m v1.21.2
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=katowice.domain1.local,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=node-role.kubernetes.io/master=,node.kubernetes.io/exclude-from-external-load-balancers=

poznan.domain1.local Ready <none> 61m v1.21.2
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,cpu=4,kubernetes.io/arch=amd64,kubernetes.io/hostname=poznan.domain1.local,kubernetes.io/os=linux
```

Dokonaj odpowiednich zmian w pliku definiującym DaemonSet demo1.yml.

Zauważ, że wdrożenie aplikacji odbędzie się tylko na nodach, które zostały oznaczone odpowiednią etykietą:

```
[student@master ~]$ vim demo1.yml
```

```
[student@master ~]$ cat demo1.yml
```

```
apiVersion: apps/v1
```

```
kind: DaemonSet
```

```
metadata:
```

```
name: demo2
```

```
labels:
```

```
  app: nginx
```

```
spec:
```

```
  selector:
```

```
    matchLabels:
```

```
      app: nginx
```

```
  template:
```

```
    metadata:
```

```
      name: demo1-pod
```

```
      labels:
```


app: nginx

spec:

nodeSelector:

cpu: "4"

containers:

- name: demo1-pod-container

image: nginx

imagePullPolicy: IfNotPresent

[student@master ~]\$ kubectl apply -f demo1.yml

daemonset.apps/demo2 created

[student@master ~]\$ kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
demo2-j7gn7	1/1	Running	0	6s
www1-7dd78bb58-45pwb	1/1	Running	0	22m
www1-7dd78bb58-9f4w6	1/1	Running	0	22m
www1-7dd78bb58-9fjwc	1/1	Running	0	54m
www1-7dd78bb58-qb2qr	1/1	Running	0	59m

[student@master ~]\$ kubectl get pods -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
demo2-j7gn7	1/1	Running	0	13s	10.44.0.5	poznan.domain1.local	<none>	<none>
www1-7dd78bb58-45pwb	1/1	Running	0	22m	10.44.0.4	poznan.domain1.local	<none>	<none>
www1-7dd78bb58-9f4w6	1/1	Running	0	22m	10.47.0.2	gdansk.domain1.local	<none>	<none>
www1-7dd78bb58-9fjwc	1/1	Running	0	54m	10.44.0.3	poznan.domain1.local	<none>	<none>
www1-7dd78bb58-qb2qr	1/1	Running	0	59m	10.44.0.2	poznan.domain1.local	<none>	<none>

Task 5: Obiekt DaemonSet – wykorzystanie taints i tolerations.

Oznacz jeden z nodów tak, by scheduler nie rozmieszczał na nim podów:

uwaga - jedna linia

```
[student@master ~]$ kubectl taint nodes gdansk.domain1.local key1=value1:NoSchedule
```

```
node/gdansk.domain1.local tainted
```

```
[student@master ~]$ kubectl describe node gdansk.domain1.local
```

Name: gdansk.domain1.local

Roles: <none>

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

kubernetes.io/arch=amd64

kubernetes.io/hostname=gdansk.domain1.local

kubernetes.io/os=linux

noc=full

Annotations: kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock

node.alpha.kubernetes.io/ttl: 0

volumes.kubernetes.io/controller-managed-attach-detach: true

CreationTimestamp: Fri, 19 Nov 2021 23:09:40 +0100

Taints: key1=value1:NoSchedule

Zdefiniuj nowy obiekt DaemonSet, który powinien standardowo na każdym z nodów umieścić po jednym podzie:

apiVersion: apps/v1

kind: DaemonSet

metadata:

name: demo1

labels:

```
  app: httpd1
spec:
  selector:
    matchLabels:
      app: httpd1
  template:
    metadata:
      name: demo1-pod
    labels:
      app: httpd1
  spec:
    containers:
      - name: demo1-pod-container
        image: httpd
        imagePullPolicy: IfNotPresent
```

```
[student@master ~]$ kubectl apply -f ds1.yml
```

daemonset.apps/demo1 created

Sprawdź w jaki sposób obiekt DaemonSet rozmieścił pody (standardowo na każdym z nodów powinien się znaleźć jeden pod zdefiniowany w DaemonSet):

```
[student@master ~]$ kubectl get ds
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
demo1	1	1	1	1		<none>	5s

```
[student@katowice ~]$ kubectl get po -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
demo1-wh9pb	1/1	Running	0	14s	10.44.0.6	poznan.domain1.local	<none>	<none>

Zmień specyfikację obiektu DaemonSet – wykorzystaj tolerations w specyfikacji poda:

```
[student@master ~]$ vim ds1.yml
```

```
[student@master ~]$ cat ds1.yml
```

```
apiVersion: apps/v1
```

```
kind: DaemonSet
```

```
metadata:
```

```
  name: demo2
```

```
  labels:
```

```
    app: httpd1
```

```
spec:
```

```
  selector:
```

```
    matchLabels:
```

```
      app: httpd1
```

```
  template:
```

```
    metadata:
```

```
      name: demo1-pod
```

```
      labels:
```

```
        app: httpd1
```

```
    spec:
```

```
      containers:
```

```
        - name: demo1-pod-container
```

```
          image: httpd
```

```
          imagePullPolicy: IfNotPresent
```

```
      tolerations:
```

```
        - key: "key1"
```

```
          operator: "Exists"
```

effect: "NoSchedule"

```
[student@master ~]$ kubectl apply -f ds1.yml
```

daemonset.apps/demo2 created

Sprawdź jak teraz zostały rozrzucone pody w klastrze (czy tolerations zadziałało?):

```
[student@master ~]$ kubectl get po -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
demo1-wh9pb	1/1	Running	0	18m	10.44.0.6	poznan.domain1.local	<none>	<none>
demo2-dgrtz	1/1	Running	0	17s	10.36.0.6	gdansk.domain1.local	<none>	<none>
demo2-ncjnh	1/1	Running	0	17s	10.44.0.7	poznan.domain1.local	<none>	<none>

Zdejmij ograniczenia na gdansku:

uwaga - jedna linia

```
[student@master ~]$ kubectl taint nodes gdansk.domain1.local key1=value1:NoSchedule-
```

node/gdansk.domain1.local untainted

```
[student@master ~]$ kubectl describe node gdansk.domain1.local
```

Name: gdansk.domain1.local

Roles: <none>

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

kubernetes.io/arch=amd64

kubernetes.io/hostname=gdansk.domain1.local

kubernetes.io/os=linux

moc=full

Annotations: kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock

node.alpha.kubernetes.io/ttl: 0

volumes.kubernetes.io/controller-managed-attach-detach: true

CreationTimestamp: Fri, 19 Nov 2021 23:09:40 +0100

Taints: <none>

Task 6: Obiekt Deployment – obiekt Service.

Popraw wcześniejszy deployment tworząc nowe wdrożenie:

```
[student@master ~]$ vim www1.yml
```

```
[student@master ~]$ cat www1.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  creationTimestamp: null
```

```
  labels:
```

```
    app: www2
```

```
  name: www2
```

```
spec:
```

```
  replicas: 2
```

```
  selector:
```

```
    matchLabels:
```

```
      app: www2
```

```
  strategy: {}
```

```
  template:
```

```
    metadata:
```

```
      creationTimestamp: null
```

```
      labels:
```

```
        app: www2
```

spec:

containers:

- image: nginx

name: nginx

imagePullPolicy: IfNotPresent

ports:

- containerPort: 80

name: http-port

resources: {}

status: {}

[student@master ~]\$ kubectl apply -f www1.yml

deployment.apps/www2 created

[student@master ~]\$ kubectl get deploy

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
------	-------	------------	-----------	-----

www1	4/4	4	4	3h48m
------	-----	---	---	-------

www2	2/2	2	2	21s
------	-----	---	---	-----

Nasza aplikacja działa na 2 osobnych nodach i adresach IP:

[student@master ~]\$ kubectl get pods -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
demo1-csqm8	1/1	Running	0	169m	10.44.0.1	poznan.domain1.local	<none>	<none>
demo2-j7gn7	1/1	Running	0	169m	10.44.0.5	poznan.domain1.local	<none>	<none>
www1-7dd78bb58-45pwb	1/1	Running	0	3h11m	10.44.0.4	poznan.domain1.local	<none>	<none>
www1-7dd78bb58-9f4w6	1/1	Running	0	3h11m	10.47.0.2	gdansk.domain1.local	<none>	<none>
www1-7dd78bb58-9fjwc	1/1	Running	0	3h43m	10.44.0.3	poznan.domain1.local	<none>	<none>
www1-7dd78bb58-qb2qr	1/1	Running	0	3h49m	10.44.0.2	poznan.domain1.local	<none>	<none>
www2-7fbf7b8894-dv6x8	1/1	Running	0	34s	10.47.0.1	gdansk.domain1.local	<none>	<none>
www2-7fbf7b8894-x9xnb	1/1	Running	0	34s	10.47.0.3	gdansk.domain1.local	<none>	<none>

```
[student@master ~]$ curl http://10.47.0.1
```

```
<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

  body {

    width: 35em;

    margin: 0 auto;

    font-family: Tahoma, Verdana, Arial, sans-serif;

  }

</style>

</head>

<body>

<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to

<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at

<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>

</body>

</html>
```

```
[student@master ~]$ curl http://10.47.0.3
```

```
<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>
```



```

<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```

```
[student@master ~]$ kubectl expose deployment www2
```

```
service/www2 exposed
```

```
[student@master ~]$ kubectl get all -l app=www2
```

NAME	READY	STATUS	RESTARTS	AGE
pod/www2-7fbf7b8894-dv6x8	1/1	Running	0	11m
pod/www2-7fbf7b8894-x9xnb	1/1	Running	0	11m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/www2	ClusterIP	10.101.80.161	<none>	80/TCP	85s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www2	2/2	2	2	11m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www2-7fbf7b8894	2	2	2	11m

```
[student@master ~]$ curl http://10.101.80.161
```

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
```

</html>

Task 7: Obiekt Deployment – obiekt NodePort.

```
[student@master ~]$ vim www1.yml
```

```
[student@master ~]$ cat www1.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  creationTimestamp: null
```

```
  labels:
```

```
    app: www3
```

```
  name: www3
```

```
spec:
```

```
  replicas: 2
```

```
  selector:
```

```
    matchLabels:
```

```
      app: www3
```

```
  strategy: {}
```

```
  template:
```

```
    metadata:
```

```
      creationTimestamp: null
```

```
      labels:
```

```
        app: www3
```

```
spec:
  containers:
  - image: httpd
    name: httpd
    imagePullPolicy: IfNotPresent
  ports:
  - containerPort: 80
    name: http-port
  resources: {}
status: {}
```

```
[student@master ~]$ kubectl apply -f www1.yml
```

deployment.apps/www3 created

```
[student@master ~]$ kubectl get deploy
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
www1	4/4	4	4	4h9m
www2	2/2	2	2	20m
www3	2/2	2	2	3m58s

```
[student@master ~]$ kubectl expose deployment www3 --type=NodePort
```

service/www3 exposed

```
[student@katowice ~]$ kubectl get all -l app=www3
```

NAME	READY	STATUS	RESTARTS	AGE
pod/www3-7f5bcd4cd5-6w6ll	1/1	Running	0	6m22s
pod/www3-7f5bcd4cd5-ns4b6	1/1	Running	0	6m22s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/www3	NodePort	10.107.167.1	<none>	80:31024/TCP	109s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www3	2/2	2	2	6m22s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www3-7f5bcd4cd5	2	2	2	6m22s

```
[student@master ~]$ curl http://10.107.167.1
```

```
<html><body><h1>It works!</h1></body></html>
```

```
[student@master ~]$ curl localhost:31024
```

```
<html><body><h1>It works!</h1></body></html>
```

```
[student@master ~]$
```

```
[student@master ~]$ curl localhost:80
```

```
curl: (7) Failed to connect to localhost port 80: Connection refused
```

Task 8: Obiekt Deployment – obiekt LoadBalancer.

Popraw wcześniejszy deployment tworząc nowe wdrożenie:

```
[student@master ~]$ vim www1.yml
```

```
[student@master ~]$ cat www1.yml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

creationTimestamp: null

labels:

app: **www4**

name: **www4**

spec:

replicas: 2

selector:

matchLabels:

app: **www4**

strategy: {}

template:

metadata:

creationTimestamp: null

labels:

app: **www4**

spec:

containers:

- image: httpd

name: httpd

imagePullPolicy: IfNotPresent

ports:

- containerPort: 80

name: http-port

resources: {}

status: {}

[student@master ~]\$ kubectl apply -f www1.yml

deployment.apps/www4 created

```
[student@master ~]$ kubectl get deployments
```

	NAME	READY	UP-TO-DATE	AVAILABLE	AGE
www1	4/4	4	4	4h42m	
www2	2/2	2	2	53m	
www3	2/2	2	2	36m	
www4	2/2	2	2	19s	

```
[student@master ~]$ kubectl expose deployment www4 --type=LoadBalancer
```

service/www4 exposed

```
[student@master ~]$ kubectl get all -l app=www4
```

	NAME	READY	STATUS	RESTARTS	AGE
pod/www4-cc7f4c6cb-28fs6	1/1	Running	0	2m41s	
pod/www4-cc7f4c6cb-fvd9k	1/1	Running	0	2m41s	

	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/www4	LoadBalancer	10.107.104.200	<pending>	80:32161/TCP	111s	

	NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/www4	2/2	2	2	2m41s	

	NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/www4-cc7f4c6cb	2	2	2	2m41s	

```
[student@master ~]$ curl 10.107.104.200
```

```
<html><body><h1>It works!</h1></body></html>
```

```
[student@master ~]$ curl localhost:80
```

```
curl: (7) Failed to connect to localhost port 80: Connection refused
```

```
[student@master ~]$ curl localhost:32161
```

```
<html><body><h1>It works!</h1></body></html>
```

Task 9: Obiekt StatefulSet i Headless Service.

Stwórz definicję obiektu StatefulSet:

```
[student@master ~]$ vim state.yml
```

```
[student@master ~]$ cat state.yml
```

```
apiVersion: "apps/v1"
```

```
kind: StatefulSet
```

```
metadata:
```

```
  name: redis
```

```
  labels:
```

```
    app: redis
```

```
spec:
```

```
  serviceName: redis-service #obiekt Headless Service
```

```
  replicas: 5
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: redis
```

```
    spec:
```

```
      containers:
```

```
        - name: redis-cont
```


image: redis:5.0.1-alpine

imagePullPolicy: IfNotPresent

selector:

matchLabels:

app: redis

```
[student@master ~]$ kubectl apply -f state.yml
```

statefulset.apps/redis created

```
[student@master ~]$ kubectl get statefulsets
```

NAME	READY	AGE
------	-------	-----

redis	1/5	12s
-------	-----	-----

```
[student@master ~]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
------	-------	--------	----------	-----

redis-0	1/1	Running	0	17s
---------	-----	---------	---	-----

redis-1	1/1	Running	0	9s
---------	-----	---------	---	----

redis-2	0/1	ContainerCreating	0	2s
---------	-----	-------------------	---	----

Stwórz definicję obiektu Headless. Pamiętaj o opcji clusterIP: None

w specyfikacji obiektu:

```
[student@master ~]$ vim headless.yml
```

```
[student@master ~]$ cat headless.yml
```

apiVersion: "v1"

kind: Service

metadata:

name: redis-service

spec:

clusterIP: None

ports:

- port: 6379

targetPort: 6379

name: client

selector:

app: redis

```
[student@master ~]$ kubectl apply -f headless.yml
```

service/redis-service created

```
[student@master ~]$ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16h
redis-service	ClusterIP	None	<none>	6379/TCP	8s

```
[student@master ~]$ kubectl get svc -o wide
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16h	<none>
redis-service	ClusterIP	None	<none>	6379/TCP	77s	app=redis

Skasuj jeden z podów należących do StatefulSet, zaobserwuj odbudowywanie się nowego Poda – jaką ma nazwę? Czy taka sama sytuacja występowała

np. w obiekcie Deployment ?

```
[student@master ~]$ kubectl get po -l app=redis
```

NAME	READY	STATUS	RESTARTS	AGE
redis-0	1/1	Running	0	31m
redis-1	1/1	Running	0	31m
redis-2	1/1	Running	0	31m
redis-3	1/1	Running	0	31m
redis-4	1/1	Running	0	31m

```
[student@master ~]$ kubectl delete pod redis-1
```

pod "redis-1" deleted

```
[student@master ~]$ kubectl get po -l app=redis
```

NAME	READY	STATUS	RESTARTS	AGE
redis-0	1/1	Running	0	32m
redis-1	1/1	Running	0	3s
redis-2	1/1	Running	0	31m
redis-3	1/1	Running	0	31m
redis-4	1/1	Running	0	31m

Lab 8: Obiekt Secret

Task 1. Tworzenie obiektu typu Secret.

Cele:

- sprawdzenie istniejących w Kubernetesie sekretów
- tworzenie obiektu Sekret z poufnymi danymi

***Lab'y możesz wykonać na Minikube lub na stworzonym klastrze.
Poniższe ćwiczenia zostały wykonane po uruchomieniu Minikube:***

```
[student@master ~]$ kubectl get secrets
```

NAME	TYPE	DATA	AGE
default-token-hgrc6	kubernetes.io/service-account-token	3	134d

```
[student@master ~]$ kubectl get secrets --all-namespaces
```

NAMESPACE	NAME	TYPE	DATA	AGE
default	default-token-hgrc6	kubernetes.io/service-account-token	3	134d
kube-node-lease	default-token-mqwpc	kubernetes.io/service-account-token	3	134d

kube-public	default-token-4kjwx	kubernetes.io/service-account-token	3	134d
kube-system	attachdetach-controller-token-xkjwn	kubernetes.io/service-account-token	3	134d
kube-system	bootstrap-signer-token-dhz5s	kubernetes.io/service-account-token	3	134d
kube-system	certificate-controller-token-nlpzr	kubernetes.io/service-account-token	3	134d
kube-system	clusterrole-aggregation-controller-token-kzm59	kubernetes.io/service-account-token	3	134d

[student@master ~]\$ **kubect! explain secret.data**

KIND: Secret

VERSION: v1

FIELD: data <map[string]string>

DESCRIPTION:

Data contains the secret data. Each key must consist of alphanumeric characters, '-', '_' or '.'. The serialized form of the secret data is a base64 encoded string, representing the arbitrary (possibly non-string) data value here. Described in <https://tools.ietf.org/html/rfc4648#section-4>

[student@master ~]\$ **kubect! explain secret.stringData**

KIND: Secret

VERSION: v1

FIELD: stringData <map[string]string>

DESCRIPTION:

stringData allows specifying non-binary secret data in string form. It is provided as a write-only convenience method. All keys and values are merged into the data field on write, overwriting any existing values. It is never output when reading from the API.

Stwórz szablon naszego obiektu Secret. Nie każda zmienna musi być kodowana:

[student@master ~]\$ **vim secret.yml**

[student@master ~]\$ **cat secret.yml**

apiVersion: v1

kind: Secret

metadata:

name: tajne-dane

data:

login: ""

haslo: ""

```
stringData:  
  ip: "10.10.1.1"
```

Wygeneruj login i hasło zakodowane za pomocą base64. Opcja -n w komendzie echo nie zapisuje znaku końca linii w zakodowanym tekście (może to być istotne z punktu widzenia Kubernetesa):

```
[student@master ~]$ echo -n "root" | base64
```

```
cm9vdA==
```

```
[student@master ~]$ echo -n "TajneHaslo" | base64
```

```
VGFqbmlVYXNsbw==
```

```
[student@master ~]$
```

Uzupełnij nasz szablon Sekretu zakodowanymi wartościami:

```
[student@master ~]$ vim secret.yml
```

```
[student@master ~]$ cat secret.yml
```

```
apiVersion: v1  
kind: Secret  
metadata:  
  name: tajne-dane  
data:  
  login: "cm9vdA=="  
  haslo: "VGFqbmlVYXNsbw=="  
stringData:  
  ip: "10.10.1.1"
```

```
[student@master ~]$ kubectl apply -f secret.yml
```

```
secret/tajne-dane created
```

```
[student@master ~]$ kubectl get secrets
```

NAME	TYPE	DATA	AGE
default-token-hgrc6	kubernetes.io/service-account-token	3	134d
tajne-dane	Opaque	3	5s

```
[student@master ~]$ kubectl describe secrets tajne-dane
```

```
Name:      tajne-dane
Namespace:  default
Labels:     <none>
Annotations: <none>
```

```
Type: Opaque
```

```
Data
```

```
====
```

```
haslo: 10 bytes
```

```
ip: 9 bytes
```

```
login: 6 bytes
```

Przygotuj deklarację poda, który odwołuje się do obiektu Secret tajne-dane:

```
[student@master ~]$ vim pod-secret.yml
```

```
[student@master ~]$ cat pod-secret.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: mysql-pod
spec:
  containers:
    - name: mysql-cont
      image: mysql
      imagePullPolicy: IfNotPresent
  env:
    - name: MYSQL_ROOT_PASSWORD
      valueFrom:
        secretKeyRef:
          name: tajne-dane
          key: haslo
```

Uruchom poda mysql-pod, poczekaj aż uzyska status Running i sprawdź czy baza danych została utworzona:

```
[student@master ~]$ kubectl apply -f pod-secret.yml
```

pod/mysql-pod created

```
[student@master ~]$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
mysql-pod	0/1	ContainerCreating	0	9s

```
[student@master ~]$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
mysql-pod	1/1	Running	0	2m18s

```
[student@master ~]$ kubectl exec -it mysql-pod -- /bin/bash
```

```
root@mysql-pod:/# mysql -u root -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 8

Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.01 sec)
```

```
mysql> exit
```

Bye

root@mysql-pod:/# env

KUBERNETES_SERVICE_PORT_HTTPS=443

KUBERNETES_SERVICE_PORT=443

MYSQL_MAJOR=8.0

HOSTNAME=mysql-pod

PWD=/

MYSQL_ROOT_PASSWORD=TajneHaslo

HOME=/root

KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443

MYSQL_VERSION=8.0.25-1debian10

GOSU_VERSION=1.12

TERM=xterm

SHLVL=1

KUBERNETES_PORT_443_TCP_PROTO=tcp

KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1

KUBERNETES_SERVICE_HOST=10.96.0.1

KUBERNETES_PORT=tcp://10.96.0.1:443

KUBERNETES_PORT_443_TCP_PORT=443

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

_=/usr/bin/env

root@mysql-pod:/# exit

exit

[student@master ~]\$

Task 2: Podmontowanie obiektu Secret.

Znajdź opcję montowania obiektów typu Secret i stwórz poda z podmontowanym obiektem Secret:

```
[student@master ~]$ kubectl explain pod.spec.volumes | grep -A 1 secret
```

Items for all in one resources secrets, configmaps, and downward API

--

secret <Object>

Secret represents a secret that should populate this volume. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#secret>

```
[student@master ~]$ kubectl explain pod.spec.volumes.secret
```

KIND: Pod

VERSION: v1

RESOURCE: secret <Object>

DESCRIPTION:

Secret represents a secret that should populate this volume. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#secret>

Adapts a Secret into a volume.

The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

FIELDS:

defaultMode <integer>

Optional: mode bits used to set permissions on created files by default.

Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

items <[]Object>

If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will

error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

optional <boolean>

Specify whether the Secret or its keys must be defined

secretName <string>

Name of the secret in the pod's namespace to use. More info:

<https://kubernetes.io/docs/concepts/storage/volumes#secret>

```
[student@master ~]$ vim pod-mount-secret.yml
```

```
[student@master ~]$ cat pod-mount-secret.yml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: sekrety
```

```
spec:
```

```
  containers:
```

```
    - name: sekrety-cont
```

```
      image: httpd
```

```
        imagePullPolicy: IfNotPresent
```

```
  volumeMounts:
```

```
    - name: tajneprzezpoufne
```

```
      mountPath: /sekrety
```

```
      readOnly: true
```

```
volumes:
```

```
  - name: tajneprzezpoufne
```

```
    secret:
```

```
      secretName: tajne-dane
```

```
[student@master ~]$ kubectl apply -f pod-mount-secret.yml
```

pod/sekrety created

```
[student@master ~]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
mysql-pod	1/1	Running	0	33m
sekrety	0/1	ContainerCreating	0	4s

```
[student@master ~]$ kubectl get pods
```

```
0
```

```
NAME      READY STATUS  RESTARTS  AGE
mysql-pod 1/1   Running  0         33m
sekrety   1/1   Running  0         31s
```

```
[student@master ~]$ kubectl exec -it sekrety -- /bin/bash
```

```
root@sekrety:/usr/local/apache2# ls /
```

```
bin boot dev  etc home lib  lib64 media mnt opt  proc root run sbin sekrety  srv sys tmp usr var
```

```
root@sekrety:/usr/local/apache2# ls /sekrety
```

```
haslo ip login
```

```
root@sekrety:/usr/local/apache2# cat /sekrety/haslo
```

```
TajneHasloroot@sekrety:/usr/local/apache2#
```

```
root@sekrety:/usr/local/apache2# cat /sekrety/ip
```

```
10.10.1.1root@sekrety:/usr/local/apache2#
```

```
root@sekrety:/usr/local/apache2# cat /sekrety/login
```

```
marcinroot@sekrety:/usr/local/apache2#
```

Sprawdź czy można nadpisać dane z obiektu Secret:

```
root@sekrety:/usr/local/apache2# echo "192.168.10.1" > /sekrety/ip
```

```
bash: /sekrety/ip: Read-only file system
```

```
root@sekrety:/usr/local/apache2# exit
```

```
exit
```

```
command terminated with exit code 1
```

Lab 9: Obiekt ConfigMap.

Stwórz plik w formacie YAML zawierający 2 definicje obiektów: ConfigMap oraz korzystającego z ConfigMap Pod. Jeśli w jednym pliku znajdują się 2 definicje, to każda z nich zaczyna się trzema myślinikami (znak '---').

```
[student@master ~]$ vim cm.yml
```

```
[student@master ~]$ cat cm.yml
```

```
---  
  
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: mapa1  
data:  
  napis: Hurra Kubernetes
```

```
---  
  
apiVersion: v1  
kind: Pod  
metadata:  
  name: pod-mapa1  
spec:  
  containers:  
    - name: pod-mapa1-kont1
```

image: busybox

imagePullPolicy: IfNotPresent

command: ["/bin/sh", "-c", "echo \$(POKAZ)"]

env:

- name: POKAZ

valueFrom:

configMapKeyRef:

name: mapa1

key: napis

Sprawdź obecność na klastrze obiektów typu ConfigMap i ewentualnie Pod.

[student@master ~]\$ **kubectl get configmaps**

NAME	DATA	AGE
kube-root-ca.crt	1	252d

[student@master ~]\$ **kubectl get cm**

NAME	DATA	AGE
kube-root-ca.crt	1	252d

[student@master ~]\$ **kubectl get po**

No resources found in default namespace.

[student@master ~]\$ **kubectl apply -f cm.yml**

configmap/mapa1 created

pod/pod-map1 created

Sprawdź czy ConfigMapa została stworzona i czy zmienna napis została przekazana do Poda.

[student@master ~]\$ **kubectl get cm**

NAME	DATA	AGE
------	------	-----

kube-root-ca.crt 1 252d

mapa1 1 7s

```
[student@master ~]$ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
------	-------	--------	----------	-----

pod-mapa1	0/1	Completed	2	17s
-----------	-----	-----------	---	-----

```
[student@master ~]$ kubectl logs pod-mapa1
```

Hurra Kubernetes

Lab 10: Sposoby montowania zasobów – volumeny.

Task 1: Podmontowanie volumenu.

Montowanie katalogu znajdującego się na dysku noda. Zdefiniuj deklarację poda z montowaniem zasobu:

```
[student@master ~]$ vim k8s/lab3/volume.yml
```

```
[student@master ~]$ cat k8s/lab3/volume.yml
```

apiVersion: v1

kind: Pod

metadata:

name: volcont

spec:

volumes:

- name: vol1

hostPath:

path: /data

containers:

- name: volcont-pod

image: busybox

imagePullPolicy: IfNotPresent

command: ['sh', '-c', 'sleep 240']

volumeMounts:

- mountPath: /folderPod

name: vol1

Zaloguj się na hosty node1 i node2. Stwórz na nich folder, który ma być montowany w Podzie.

[student@master ~]\$ ssh student@node1

[student@node1 ~]\$ sudo mkdir /folderPod

[student@node1 ~]\$ ls /

bin dev folderPod init lib64 linuxrc mnt proc run srv tmp var

data etc home lib libexec media opt root sbin sys usr

[student@master ~]\$ kubectl apply -f k8s/lab3/volume.yml

pod/volcont configured

[student@master ~]\$ kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
cron1-1625710200-nd6kr	0/1	Completed	0	83m
lab-pod	1/1	Running	3	41h
lab3-pod	1/1	Running	0	77m

```
lab4-pod      1/1   Running   0    77m
mywww         1/1   Running   1    21h
pv-pod        1/1   Running   0    19m
testjob-tkm6q 0/1   Completed 0    11h
volcont       1/1   Running   0    20s
www-7dc75659f-j5jf8 1/1   Running   1    12h
www-7dc75659f-l9t79 1/1   Running   1    12h
www-7dc75659f-r27cx 1/1   Running   1    12h
www-7dc75659f-v8m2m 1/1   Running   1    12h
www-7dc75659f-xxdhj 1/1   Running   1    12h
```

```
[student@master ~]$ kubectl exec -it volcont -- /bin/sh
```

```
/ # ls
```

```
bin    dev    etc    folderPod home   proc   root   sys    tmp    usr    var
```

```
/ # touch folderPod/plik1
```

```
/ # ls folderPod/
```

```
plik1
```

```
/ # echo "Witaj w Kubernetesie" > folderPod/plik1
```

```
/ # exit
```

```
[student@master ~]$
```

Zaloguj się na host node1 lub node2. Sprawdź czy plik utworzony ze środka Poda istnieje.

```
[student@master ~]$ ssh student@node1
```

```
$ ls /
```

```
bin  dev  folderPod  init  lib64  linuxrc  mnt  proc  run  srv  tmp  var
```

```
data  etc  home  lib  libexec  media  opt  root  sbin  sys  usr
```

```
$ sudo rmdir /folderPod
```

```
$ ls /data
```

```
plik1
```

```
$ cat /data/plik1
```


Witaj w Kubernetesie

\$ exit

logout

[student@base ~]\$

Task 2: Volumeny efemeryczne.

Zdefiniuj i uruchom Poda z dwoma kontnerami wewnątrz Poda. Wprowadź nową sekcję volumes: definiującą dołączane volumeny (w tym przypadku EmptyDir służący do wymiany danych pomiędzy kontenerami wewnątrz Poda).

W definicji poszczególnych kontenerów podmontuj volumen.

apiVersion: v1

kind: Pod

metadata:

name: kat-share

spec:

containers:

- name: first-cont

image: alpine

imagePullPolicy: IfNotPresent

volumeMounts:

- name: shared-data

mountPath: /tmp

command: ["sh", "-c", "echo hello", ">", "/data/plik1"]

- name: second-cont

```
image: busybox
imagePullPolicy: IfNotPresent
volumeMounts:
- name: shared-data
  mountPath: /data
command: ["sh", "-c", "ls", "/data"]
```

```
volumes:
- name: shared-data
  emptyDir: {}
```

```
[student@master ~]$ kubectl apply -f empty-dir.yml
```

```
pod/kat-share created
```

```
[student@master ~]$ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
kat-share	0/2	Completed	0	6s
pv-pod	0/1	ContainerCreating	0	4h40m
redis-0	1/1	Running	0	46m
redis-1	1/1	Running	0	14m
redis-2	1/1	Running	0	45m
redis-3	1/1	Running	0	45m
redis-4	1/1	Running	0	45m

```
[student@master ~]$ kubectl logs -c second-cont kat-share
```

```
bin
```

```
data
```

dev
etc
home
proc
root
sys
tmp
usr
var

Task 3: Przykład obiektów PersistentVolume i PersistentVolumeClaim.

*Na serwerze base przygotuj udział NFS w katalogu /exampleshare (dostępny dla każdego hosta – opcja *(rw)), który będzie podmontowywany w klastrze Kubernetes. Możesz utworzyć w katalogu /exampleshare jakiś plik lub folder.*

```
[root@base ~]# systemctl status nfs-server
```

- nfs-server.service - NFS server and services

Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor preset: disabled)

Drop-In: /run/systemd/generator/nfs-server.service.d

└─order-with-mounts.conf

Active: active (exited) since Sun 2021-10-24 07:49:47 CEST; 2h 41min ago

Main PID: 1648 (code=exited, status=0/SUCCESS)

Tasks: 0 (limit: 62106)

Memory: 0B

CGroup: /system.slice/nfs-server.service

paź 24 07:49:47 base.domain1.local systemd[1]: Starting NFS server and services...

paź 24 07:49:47 base.domain1.local systemd[1]: Started NFS server and services.

Dokonaj zmian w pliku /etc/exports:

[root@base ~]# **vim /etc/exports**

[root@base ~]# **cat /etc/exports**

/exampleshare *(rw)

/var/www/html 10.10.0.0/16(ro)

/remotehomes 10.10.0.0/16(rw)

[root@base ~]# **exportfs -rv**

exporting 10.10.0.0/16:/remotehomes

exporting 10.10.0.0/16:/var/www/html

exporting */:/exampleshare

Na serwerze master sprawdź czy zasób NFS z serwera base jest dostępny:

[student@master ~]\$ **sudo apt install nfs-common**

[sudo] password for student:

Failed to set locale, defaulting to C.UTF-8

Last metadata expiration check: 0:41:43 ago on Sun Oct 24 08:50:26 2021.

Dependencies resolved.

=====

Package	Architecture	Version	Repository	Size
---------	--------------	---------	------------	------

=====

Installing:

nfs-utils	x86_64	1:2.3.3-41.el8_4.2	baseos	498 k
-----------	--------	--------------------	--------	-------

Installing dependencies:

gssproxy	x86_64	0.8.0-19.el8	baseos	119 k
keyutils	x86_64	1.5.10-6.el8	baseos	63 k
libverto-libevent	x86_64	0.3.0-5.el8	baseos	16 k
rpcbind	x86_64	1.2.5-8.el8	baseos	70 k

Transaction Summary

=====

Install 5 Packages

Total download size: 766 k

Installed size: 2.0 M

Is this ok [y/N]: **y**

Downloading Packages:

(1/5): libverto-libevent-0.3.0-5.el8.x86_64.rpm	1.1 MB/s 16 kB	00:00
(2/5): keyutils-1.5.10-6.el8.x86_64.rpm	1.5 MB/s 63 kB	00:00
(3/5): rpcbind-1.2.5-8.el8.x86_64.rpm	1.1 MB/s 70 kB	00:00
(4/5): nfs-utils-2.3.3-41.el8_4.2.x86_64.rpm	1.0 MB/s 498 kB	00:00
(5/5): gssproxy-0.8.0-19.el8.x86_64.rpm	193 kB/s 119 kB	00:00

Total	962 kB/s 766 kB	00:00
-------	-------------------	-------

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing	:	1/1
Running scriptlet:	rpcbind-1.2.5-8.el8.x86_64	1/5
Installing	: rpcbind-1.2.5-8.el8.x86_64	1/5
Running scriptlet:	rpcbind-1.2.5-8.el8.x86_64	1/5
Installing	: libverto-libevent-0.3.0-5.el8.x86_64	2/5
Installing	: gssproxy-0.8.0-19.el8.x86_64	3/5
Running scriptlet:	gssproxy-0.8.0-19.el8.x86_64	3/5
Installing	: keyutils-1.5.10-6.el8.x86_64	4/5
Running scriptlet:	nfs-utils-1:2.3.3-41.el8_4.2.x86_64	5/5
Installing	: nfs-utils-1:2.3.3-41.el8_4.2.x86_64	5/5
Running scriptlet:	nfs-utils-1:2.3.3-41.el8_4.2.x86_64	5/5
Verifying	: gssproxy-0.8.0-19.el8.x86_64	1/5
Verifying	: keyutils-1.5.10-6.el8.x86_64	2/5
Verifying	: libverto-libevent-0.3.0-5.el8.x86_64	3/5
Verifying	: nfs-utils-1:2.3.3-41.el8_4.2.x86_64	4/5
Verifying	: rpcbind-1.2.5-8.el8.x86_64	5/5

Installed:

gssproxy-0.8.0-19.el8.x86_64 keyutils-1.5.10-6.el8.x86_64 libverto-libevent-0.3.0-5.el8.x86_64 nfs-utils-1:2.3.3-41.el8_4.2.x86_64
rpcbind-1.2.5-8.el8.x86_64

Complete!

[student@master ~]\$ showmount -e 10.10.1.1

Export list for base:

/exampleshare *

/remotehomes 10.10.0.0/16

/var/www/html 10.10.0.0/16

Zdefiniuj i uruchom definicję obiektu PersistentVolume. Poszczególne opcje specyfikacji NFS możesz zobaczyć za pomocą polecenia:

```
[student@master]$ kubectl explain pv.spec.nfs
```

```
[student@master]$ vim persistent-volume-nfs.yml
```

```
[student@master]$ cat persistent-volume-nfs.yml
```

```
apiVersion: v1
```

```
kind: PersistentVolume
```

```
metadata:
```

```
  name: pv1
```

```
spec:
```

```
  nfs:
```

```
    path: /exampleshare
```

```
    server: base
```

```
    readOnly: false
```

```
capacity:
```

```
  storage: 1Gi
```

```
accessModes:
```

```
  - ReadWriteMany
```

```
persistentVolumeReclaimPolicy: Retain
```

```
[student@master ~]$ kubectl apply -f persistent-volume-nfs.yml
```

```
persistentvolume/pv1 created
```

```
[student@master ~]$ kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
pv1	1Gi	RWX	Retain	Available				8s

Zdefiniuj i uruchom definicję obiektu PersistentVolumeClaim:

uwaga - jedna linia

```
[student@master persistent-volume]$ vim persistent-volume-nfs-claim.yml
```

uwaga - jedna linia

```
[student@master persistent-volume]$ cat persistent-volume-nfs-claim.yml
```

```
apiVersion: v1
```

```
kind: PersistentVolumeClaim
```

```
metadata:
```

```
  name: pvc1
```

```
spec:
```

```
  accessModes:
```

```
    - ReadWriteMany
```

```
  resources:
```

```
    requests:
```

```
      storage: 200Mi
```

```
[student@master ~]$ kubectl apply -f persistent-volume-nfs-claim.yml
```

```
persistentvolumeclaim/pvc1 created
```

```
[student@master ~]$ kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc1	Bound	pv1	1Gi	RWX		70m

Spójrz na kolumnę CLAIM w dostępnym PersistentVolume:

```
[student@master ~]$ kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
pv1	1Gi	RWX	Retain	Bound	default/pvc1			84m

Zdefiniuj i uruchom obiekt Deployment wykorzystujący skonfigurowany PersistentVolumeClaim:

```
[student@master ~]$ vim deployment.yml
```

```
[student@master ~]$ cat deployment.yml
```


apiVersion: apps/v1

kind: Deployment

metadata:

name: pv-deploy

labels:

app: pv-deploy-httpd

spec:

selector:

matchLabels:

app: pv-deploy-httpd

template:

metadata:

labels:

app: pv-deploy-httpd

spec:

containers:

- name: pv-deploy-cont

image: httpd

imagePullPolicy: IfNotPresent

volumeMounts:

- name: nfs-vol

mountPath: /tmp

ports:

- containerPort: 80

protocol: TCP

volumes:

- name: nfs-vol

persistentVolumeClaim:

claimName: pvc1

```
[student@master ~]$ kubectl apply -f deployment.yml
```

```
deployment.apps/pv-deploy created
```

```
[student@master ~]$ kubectl get deploy
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
pv-deploy	1/1	1	1	65m

Zbadaj czy w opisie deploymentu widać zamontowane zasoby:

```
[student@master ~]$ kubectl describe deploy pv-deploy
```

Name: pv-deploy

Namespace: default

CreationTimestamp: Sun, 24 Oct 2021 09:54:36 +0200

Labels: app=pv-deploy-httpd

Annotations: deployment.kubernetes.io/revision: 1

Selector: app=pv-deploy-httpd

Replicas: 1 desired | 1 updated | 1 total | 1 available | 0 unavailable

StrategyType: RollingUpdate

MinReadySeconds: 0

RollingUpdateStrategy: 25% max unavailable, 25% max surge

Pod Template:

Labels: app=pv-deploy-httpd

Containers:

pv-deploy-cont:

Image: httpd

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts:

/tmp from nfs-vol (rw)

Volumes:

nfs-vol:

Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)

ClaimName: pvc1

ReadOnly: false

Conditions:

Type	Status	Reason
------	--------	--------

----	-----	-----
------	-------	-------

Available	True	MinimumReplicasAvailable
-----------	------	--------------------------

Progressing	True	NewReplicaSetAvailable
-------------	------	------------------------

OldReplicaSets: <none>

NewReplicaSet: pv-deploy-5c87f5dc5b (1/1 replicas created)

Events: <none>

Lab 11: Limitowanie zasobów w Kubernetes.

Cele:

- uruchamianie obiektów Kubernetesa z limitem dostępu do CPU i RAM

[student@master ~]\$ **kubectl get nodes**

NAME	STATUS	ROLES	AGE	VERSION
minikube	Ready	control-plane,master	134d	v1.20.2

Przygotuj deklarację pierwszego poda wykorzystującego obraz progrium/stress do testowania obciążenia:

```
[student@master ~]$ vim pod-stress.yml
```

```
[student@master ~]$ cat pod-stress.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: stress-pod1
spec:
  containers:
  - name: stress-cont1
    image: progrium/stress
    imagePullPolicy: IfNotPresent
    command: ["stress", "--cpu", "2"]
```

```
[student@master ~]$ kubectl apply -f pod-stress.yml
```

pod/stress-pod1 created

Sprawdź obciążenie procesora w trakcie startu poda (jeszcze nie pracuje w pełni):

```
[student@master ~]$ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
stress-pod1	0/1	ContainerCreating	0	6s

```
[student@master:~$] ssh student@node1
```

student@node1's password:

Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-121-generic x86_64)

* Documentation: <https://help.ubuntu.com>

* Management: <https://landscape.canonical.com>

* Support: <https://ubuntu.com/advantage>

System information disabled due to load higher than 2.0

0 updates can be applied immediately.

Last login: Tue Jul 5 22:03:31 2022 from 10.10.1.1

```
[student@node1:~$] top
```

top - 16:01:01 up 7:13, 1 user, load average: 2.09, 1.56, 0.75

Tasks: 149 total, 3 running, 146 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.7 us, 0.3 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3931.7 total, 2002.5 free, 455.6 used, 1473.6 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 3299.5 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12790	root	20	0	7312	100	0	R	99.3	0.0	5:49.83	stress
12791	root	20	0	7312	100	0	R	98.7	0.0	5:50.14	stress
692	root	20	0	1929580	100544	64656	S	1.0	2.5	7:17.08	kubelet
3698	root	20	0	750832	42856	32520	S	0.3	1.1	0:32.81	coredns
1	root	20	0	104232	13200	8492	S	0.0	0.3	0:05.79	systemd
1	root	20	0	245652	14396	9268	S	0.0	0.1	0:02.66	systemd

**Po pewnym czasie, kiedy nasz pod będzie działał w stanie „Running”
sprawdź jeszcze raz obciążenie:**

student@master:~\$ **kubectl get po -o wide**

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
pod1	1/1	Running	0	3h26m	10.44.0.2	node1	<none>		<none>	
stress-pod1	1/1	Running	0	2m28s	10.44.0.3	node1	<none>		<none>	

[student@node1 ~]\$ **top**

top - 20:44:00 up 13 min, 2 users, load average: 1.16, 0.46, 0.23
Tasks: 260 total, 1 running, 259 sleeping, 0 stopped, 0 zombie
%Cpu(s): 50.8 us, 1.5 sy, 0.0 ni, 47.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 9735.1 total, 4507.8 free, 3252.9 used, 1974.4 buff/cache
MiB Swap: 4047.0 total, 4047.0 free, 0.0 used. 6203.1 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2546	qemu	20	0	5159320	2.4g	20352	S	206.7	25.1	3:51.85	qemu-kvm
1	root	20	0	245652	14396	9268	S	0.0	0.1	0:02.68	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp

Skasuj poda stress-pod1:

```
[student@master ~]$ kubectl delete pod stress-pod1  
pod "stress-pod1" deleted
```

Przygotuj wersję poda z limitowaniem zasobów:

```
[student@master ~]$ vim pod-stress2.yml
```

```
[student@master ~]$ cat pod-stress2.yml
```

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: stress-pod2  
spec:  
  containers:  
    - name: stress-cont1  
      image: progrium/stress  
      imagePullPolicy: IfNotPresent  
      command: ["stress", "--cpu", "2"]  
      resources:  
        requests:  
          cpu: "500m"  
          memory: "512Mi"  
        limits:  
          cpu: "500m"  
          memory: "512Mi"
```

```
[student@master ~]$ kubectl apply -f pod-stress2.yml
```

```
pod/stress-pod2 created
```

```
[student@node1 ~]$ top
```

```
top - 21:22:56 up 52 min, 2 users, load average: 0.43, 0.30, 0.28  
Tasks: 257 total, 1 running, 256 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 17.1 us, 0.8 sy, 0.0 ni, 81.7 id, 0.1 wa, 0.3 hi, 0.0 si, 0.1 st  
MiB Mem : 9735.1 total, 4486.6 free, 3254.2 used, 1994.3 buff/cache  
MiB Swap: 4047.0 total, 4047.0 free, 0.0 used, 6201.4 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2546	qemu	20	0	5028184	2.4g	20352	S	70.8	25.1	20:00.48	qemu-kvm
1	root	20	0	245652	14396	9268	S	0.0	0.1	0:02.81	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp

Skasuj poda stress-pod2 i sprawdź czy obciążenie wróciło do normy:

[student@master ~]\$ kubectl get po

NAME	READY	STATUS	RESTARTS	AGE
stress-pod2	1/1	Running	0	6m34s

[student@master ~]\$ kubectl delete pod stress-pod1

pod "stress-pod1" deleted

[student@node1 ~]\$ top

top - 21:28:12 up 57 min, 2 users, load average: 0.58, 0.70, 0.50
 Tasks: 258 total, 1 running, 257 sleeping, 0 stopped, 0 zombie
 %Cpu(s): 6.0 us, 0.9 sy, 0.0 ni, 92.9 id, 0.0 wa, 0.3 hi, 0.0 si, 0.0 st
 MiB Mem : 9735.1 total, 4346.7 free, 3254.0 used, 2134.4 buff/cache
 MiB Swap: 4047.0 total, 4047.0 free, 0.0 used. 6201.4 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2546	qemu	20	0	5019988	2.4g	20352	S	28.2	25.1	23:38.72	qemu-kvm
1379	apache	20	0	2536296	12728	7276	S	0.3	0.1	0:00.75	httpd
1770	gdm	20	0	3557656	258464	107156	S	0.3	2.6	0:04.30	gnome-shell

Dla ambitnych: Popraw deklarację użycia procesora na wartość maksymalną cpu: „1000m” i sprawdź czy pod się uruchomi oraz jakie będzie obciążenie procesora.

Uruchom teraz poda z przydzieloną ilością pamięci RAM ponad możliwości zasobów serwera. Sprawdź status poda :

[student@master ~]\$ vim pod-stress3.yml

[student@master ~]\$ cat pod-stress3.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: stress-pod1
spec:
  containers:
  - name: stress-cont1
    image: progrium/stress
    imagePullPolicy: IfNotPresent
    command: ["stress", "--cpu", "2"]
    resources:
      requests:
        cpu: "500m"
        memory: "5000Mi"
      limits:
        cpu: "500m"
        memory: "5000Mi"
```

```
[student@master ~]$ kubectl apply -f pod-stress3.yml
```

pod/stress-pod1 created

```
[student@master ~]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
stress-pod1	0/1	Pending	0	3s

Skasuj uruchomionego poda ze statusem Pending. Zmień obraz poda tak, by ustawić wysokie żądania użycia pamięci oraz procesora. Ogranicz limity dostępne dla poda w sekcji limits:

```
[student@master ~]$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
stress-pod1	0/1	Pending	0	5m13s

```
[student@master ~]$ kubectl delete pod stress-pod1
```


pod "stress-pod1" deleted

```
[student@master ~]$ cp pod-stress3.yml pod-stress4.yml
```

```
[student@master ~]$ vim pod-stress4.yml
```

```
[student@master ~]$ cat pod-stress4.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: stress-pod1
spec:
  containers:
  - name: stress-cont1
    image: progrium/stress
    imagePullPolicy: IfNotPresent
    command: ["stress", "--cpu", "4"]
    resources:
      requests:
        cpu: "300m"
      limits:
        cpu: "500m"
```

```
[student@node1 ~]$ top
```

```
top - 22:08:36 up 1:38, 2 users, load average: 0.25, 0.31, 0.34
Tasks: 258 total, 2 running, 256 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.7 us, 0.9 sy, 0.0 ni, 94.2 id, 0.0 wa, 0.3 hi, 0.0 si, 0.0 st
MiB Mem : 9735.1 total, 4302.9 free, 3257.7 used, 2174.4 buff/cache
MiB Swap: 4047.0 total, 4047.0 free, 0.0 used. 6197.4 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2546	qemu	20	0	5019988	2.4g	20352	S	22.6	25.1	36:53.75	qemu-kvm
5038	student	20	0	65540	5216	4328	R	0.3	0.1	0:00.01	top
1	root	20	0	245652	14396	9268	S	0.0	0.1	0:03.01	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd

```
[student@master ~]$ kubectl apply -f pod-stress4.yml
```

pod/stress-pod1 created

```
[student@master ~]$ kubectl get pods
```

```
NAME      READY  STATUS   RESTARTS  AGE
stress-pod1 1/1    Running  0         3m26s
```

```
[student@node1 ~]$ top
```

```
top - 22:09:07 up 1:38, 2 users, load average: 0.25, 0.31, 0.34
Tasks: 258 total, 1 running, 257 sleeping, 0 stopped, 0 zombie
%Cpu(s): 18.3 us, 0.9 sy, 0.0 ni, 80.3 id, 0.1 wa, 0.3 hi, 0.1 si, 0.0 st
MiB Mem : 9735.1 total, 4300.2 free, 3257.6 used, 2177.3 buff/cache
MiB Swap: 4047.0 total, 4047.0 free, 0.0 used. 6197.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2546	qemu	20	0	5019988	2.4g	20352	S	75.7	25.1	37:13.85	qemu-kvm
5095	student	20	0	65540	5248	4356	R	0.3	0.1	0:00.02	top
1	root	20	0	245652	14396	9268	S	0.0	0.1	0:03.01	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd

```
[student@master ~]$ kubectl get pods
```

```
NAME      READY  STATUS   RESTARTS  AGE
stress-pod1 1/1    Running  0         3m26s
```

```
[student@master ~]$ kubectl delete -f pod-stress4.yml
```

```
pod "stress-pod1" deleted
```

Dodatek A.

Task 1: Helm.

Helm pozwala na wdrożenia aplikacji o bardziej złożonej konfiguracji. Zazwyczaj takie aplikacje składają się z kilku lub więcej modułów. Helm pozwala na wdrożenie całości w jednym kroku. Do tego celu używane są tzw. Charts.

Aby skorzystać z Helma należy go wcześniej zainstalować, gdyż nie jest on częścią samego Kubernetesa:

```
[student@master ~]$ wget https://get.helm.sh/helm-v3.6.0-linux-amd64.tar.gz
```

```
--2021-11-09 19:17:42-- https://get.helm.sh/helm-v3.6.0-linux-amd64.tar.gz
```

```
Resolving get.helm.sh (get.helm.sh)... 152.199.21.175, 2606:2800:233:1cb7:261b:1f9c:2074:3c
```

```
Connecting to get.helm.sh (get.helm.sh)|152.199.21.175|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 14168950 (14M) [application/x-tar]
```

```
Saving to: 'helm-v3.6.0-linux-amd64.tar.gz'
```

```
helm-v3.6.0-linux-amd64.tar.gz 100%[=====>] 13.51M
6.52MB/s in 2.1s
```

```
2021-11-09 19:17:45 (6.52 MB/s) - 'helm-v3.6.0-linux-amd64.tar.gz' saved [14168950/14168950]
```

```
[student@master ~]$ ls
```

```
cm.yml demon.yml helm-v3.6.0-linux-amd64.tar.gz pod pod1.yml rs.yml
```

```
[student@master ~]$ tar -xvf helm-v3.6.0-linux-amd64.tar.gz
```

```
linux-amd64/
```

```
linux-amd64/helm
```

```
linux-amd64/LICENSE
```

```
linux-amd64/README.md
```

```
[student@master ~]$ sudo cp linux-amd64/helm /usr/local/bin/helm
```

[sudo] password for student:

Wyświetl dostępne repozytoria:

```
[student@master ~]$ helm list
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
------	-----------	----------	---------	--------	-------	-------------

Wyszukaj repozytoriów z pakietem apache:

```
[student@master ~]$ helm search hub apache
```

URL	CHART VERSION	APP VERSION	DESCRIPTION
https://artifacthub.io/packages/helm/bitnami/ap...	8.9.1	2.4.51	Chart for Apache HTTP Server
https://artifacthub.io/packages/helm/bitnami-ak...	8.8.3	2.4.50	Chart for Apache HTTP Server
https://artifacthub.io/packages/helm/cloudposse...	0.1.3		Apache Helm chart for Kubernetes

Pierwsze testowe uruchomienie aplikacji w klastrze za pomocą Helm. Dodaj repozytorium ealenn:

Uwaga – jedna linia

```
[student@master ~]$ helm repo add ealenn https://ealenn.github.io/charts
```

"ealenn" has been added to your repositories

```
[student@master ~]$ helm repo update
```

Hang tight while we grab the latest from your chart repositories...

...Successfully got an update from the "ealenn" chart repository

Update Complete. ✨Happy Helming!✨

```
[student@master ~]$ helm upgrade -i tester ealenn/echo-server --debug
```

history.go:56: [debug] getting history for release tester

Release "tester" does not exist. Installing it now.

install.go:173: [debug] Original chart version: ""

install.go:190: [debug] CHART PATH: /home/student/.cache/helm/repository/echo-server-0.3.1.tgz

[student@master ~]\$ kubectl get po

NAME	READY	STATUS	RESTARTS	AGE
tester-echo-server-786768d9f4-499zg	0/1	ContainerCreating	0	6s

[student@master ~]\$ kubectl get deploy

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
tester-echo-server	1/1	1	1	19s

[student@master ~]\$ kubectl get rs

NAME	DESIRED	CURRENT	READY	AGE
tester-echo-server-786768d9f4	1	1	1	22s

[student@master ~]\$ kubectl get svc

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d22h
tester-echo-server	ClusterIP	10.109.161.173	<none>	80/TCP	54s

[student@master ~]\$ curl 10.109.161.173

```
{"host":{"hostname":"10.109.161.173","ip":"","ffff:10.32.0.1","ips":[]},"http":{"method":"GET","baseUrl":"","originalUrl":"/","protocol":"http"},"request":{"params":{"0":"","/"},"query":{"},"cookies":{"},"body":{"},"headers":{"host":"10.109.161.173","user-agent":"curl/7.61.1","accept":"*/*"},"environment":{"PATH":"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","HOSTNAME":"tester-echo-server-786768d9f4-499zg","LOGS__IGNORE__PING":"false","PORT":"80","ENABLE__ENVIRONMENT":"true","ENABLE__FILE":"true","ENABLE__HEADER":"true","ENABLE__HOST":"true","ENABLE__HTTP":"true","ENABLE__REQUEST":"true","KUBERNETES_PORT_443_TCP":"tcp://10.96.0.1:443","KUBERNETES_PORT_443_TCP_PORT":"443","TESTER_ECHO_SERVER_SERVICE_HOST":"10.109.161.173","KUBERNETES_SERVICE_PORT":"443","KUBERNETES_SERVICE_PORT_HTTPS":"443","TESTER_ECHO_SERVER_SERVICE_PORT":"80","TESTER_ECHO_SERVER_SERVICE_PORT_HTTP":"80","KUBERNETES_PORT_443_TCP_ADDR":"10.96.0.1","TESTER_ECHO_SERVER_PORT_8
```

```
O_TCP":"tcp://10.109.161.173:80","TESTER_ECHO_SERVER_PORT_80_TCP_PROTO":"tcp","TESTER_ECHO_SERVER_PORT_80_TCP_PORT":"80","TESTER_ECHO_SERVER_PORT_80_TCP_ADDR":"10.109.161.173","KUBERNETES_SERVICE_HOST":"10.96.0.1","KUBERNETES_PORT_443_TCP_PROTO":"tcp","KUBERNETES_PORT":"tcp://10.96.0.1:443","TESTER_ECHO_SERVER_PORT":"tcp://10.109.161.173:80","NODE_VERSION":"12.16.3","YARN_VERSION":"1.22.4","HOME":"/root"}}
```

Sprawdź listę dostępnych pakietów helma:

```
[student@master ~]$ helm list
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
tester	default	1	2021-11-24 21:12:33.323756703 +0100 CET	deployed	echo-server-0.3.1	0.4.0

Odinstaluj pakiet tester, odinstalowane będą również poszczególne obiekty wdrożone przez Helma:

```
[student@master ~]$ helm uninstall tester
```

```
release "tester" uninstalled
```

```
[student@master ~]$ kubectl get deploy
```

```
No resources found in default namespace.
```

```
[student@master ~]$ kubectl get rs
```

```
No resources found in default namespace.
```

```
[student@master ~]$ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
tester-echo-server-786768d9f4-499zg	1/1	Terminating	0	6m45s

Dodaj repozytorium bitnami, w którym znajduje się m.in. apache:

Uwaga – jedna linia

```
[student@master ~]$ helm repo add bitnami https://charts.bitnami.com/bitnami
```

```
"bitnami" has been added to your repositories
```

Rozpakuj chart z apache:

```
[student@master ~]$ helm fetch bitnami/apache --untar
```

```
[student@master ~]$ tree apache/
```

```
apache/
```

```
|-- Chart.lock
```

```
|-- Chart.yaml
```

```
|-- README.md
```

```
|-- charts
```

```
|  `-- common
```

```
|    |-- Chart.yaml
```

```
|    |-- README.md
```

```
|    |-- templates
```

```
|      |-- _affinities.tpl
```

```
|      |-- _capabilities.tpl
```

```
|      |-- _errors.tpl
```

```
|      |-- _images.tpl
```

```
|      |-- _ingress.tpl
```

```
|      |-- _labels.tpl
```

```
|      |-- _names.tpl
```

```
|      |-- _secrets.tpl
```

```
|      |-- _storage.tpl
```

```
|      |-- _tplvalues.tpl
```

```
|      |-- _utils.tpl
```

```
|      |-- _warnings.tpl
```

```
|      `-- validations
```

```
|          |-- _cassandra.tpl
```

```
|          |-- _mariadb.tpl
```

```
|          |-- _mongodb.tpl
```

```
|          |-- _postgresql.tpl
```

```
|          |-- _redis.tpl
```

```
|          `-- _validations.tpl
```

```
|  `-- values.yaml
```

```
|-- ci
```

```
| `-- ct-values.yaml
|-- files
| |-- README.md
| `-- vhosts
|   `-- README.md
|-- templates
| |-- NOTES.txt
| |-- _helpers.tpl
| |-- configmap-vhosts.yaml
| |-- configmap.yaml
| |-- deployment.yaml
| |-- extra-list.yaml
| |-- hpa.yaml
| |-- ingress.yaml
| |-- pdb.yaml
| |-- svc.yaml
| `-- tls-secrets.yaml
|-- values.schema.json
`-- values.yaml
```

8 directories, 40 files

Zainstaluj apache z repozytorium bitnami/apache:

```
[student@master ~]$ cd apache
```

```
[student@master apache]$ helm install web1 .
```

NAME: web1

LAST DEPLOYED: Wed Nov 24 21:23:55 2021

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

CHART NAME: apache

CHART VERSION: 8.9.6

APP VERSION: 2.4.51

**** Please be patient while the chart is being deployed ****

1. Get the Apache URL by running:

**** Please ensure an external IP is associated to the web1-apache service before proceeding ****

**** Watch the status using: kubectl get svc --namespace default -w web1-apache ****

```
export SERVICE_IP=$(kubectl get svc --namespace default web1-apache --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}"
```

```
echo URL      : http://$SERVICE_IP/
```

WARNING: You did not provide a custom web application. Apache will be deployed with a default page. Check the README section "Deploying your custom web application" in <https://github.com/bitnami/charts/blob/master/bitnami/apache/README.md#deploying-your-custom-web-application>.

[student@master apache]\$ kubectl get deploy

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
web1-apache	0/1	1	0	12s

[student@master apache]\$ kubectl get rs

NAME	DESIRED	CURRENT	READY	AGE
web1-apache-94585df89	1	1	0	18s

[student@master apache]\$ kubectl get svc

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d22h

web1-apache LoadBalancer 10.96.2.68 <pending> 80:30670/TCP,443:31271/TCP 22s

[student@master apache]\$ **kubectl get po**

NAME	READY	STATUS	RESTARTS	AGE
pod-nodename	1/1	Running	0	123m
pod-selector	1/1	Running	0	125m
web1-apache-94585df89-gjj4s	0/1	Running	0	29s

[student@master apache]\$ **kubectl get svc**

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4d22h
web1-apache	LoadBalancer	10.96.2.68	<pending>	80:30670/TCP,443:31271/TCP	37s

[student@master apache]\$ **curl 10.96.2.68**

<html><body><h1>It works!</h1></body></html>

[student@master apache]\$ **helm --help**

Task 2: Wdrożenie Dashboard za pomocą Helm

Uwaga – jedna linia

[student@master ~]\$ **helm search hub --max-col-width 100 kubernetes-dashboard**

URL	CHART VERSION	APP VERSION	DESCRIPTION
https://artifacthub.io/packages/helm/k8s-dashboard/kubernetes-dashboard	5.0.4	2.4.0	General-purpose web UI for Kubernetes clusters
https://artifacthub.io/packages/helm/cloudnativeapp/kubernetes-dashboard	1.5.2	1.10.1	General-purpose web UI for Kubernetes clusters
https://artifacthub.io/packages/helm/wenerme/kubernetes-dashboard	5.0.4	2.4.0	General-purpose web UI for Kubernetes clusters
https://artifacthub.io/packages/helm/banzaicloud-stable/kubernetes-dashboard	0.9.2	1.10.0	General-purpose web UI for Kubernetes clusters
https://artifacthub.io/packages/helm/mesosphere/kubernetes-dashboard	2.0.0	2.0.0-beta6	General-purpose web UI for Kubernetes clusters

Dodaj repozytorium helma z dashboardem:

Uwaga – jedna linia

```
[student@master ~]$ helm repo add kubernetes-dashboard https://kubernetes.github.io/dashboard/
```

"kubernetes-dashboard" has been added to your repositories

```
[student@master ~]$ helm repo update
```

Hang tight while we grab the latest from your chart repositories...

...Successfully got an update from the "kubernetes-dashboard" chart repository

...Successfully got an update from the "ingress-nginx" chart repository

...Successfully got an update from the "bitnami" chart repository

Update Complete. ✨Happy Helming!✨

```
[student@master ~]$ helm fetch kubernetes-dashboard/kubernetes-dashboard --untar
```

```
[student@master ~]$ ls
```

```
daemonset.yml  helm-v3.6.0-linux-amd64.tar.gz  ingress.yml      linux-amd64  rs.yml
ds.yml         ingress-nginx                    kubernetes-dashboard  pod.yml     token
```

```
[student@master ~]$ ls kubernetes-dashboard/
```

```
Chart.lock  Chart.yaml  README.md  charts  templates  values.yaml
```

Zmień typ obiektu Service na NodePort:

```
[student@master ~]$ vim kubernetes-dashboard/values.yaml
```

Uwaga – jedna linia

```
[student@master kubernetes-dashboard]$ grep -A 4 -i ^service: values.yaml
```

service:

type: NodePort

Dashboard service port

externalPort: 443

```
[student@master ~]$ cd kubernetes-dashboard/
```

```
[student@master kubernetes-dashboard]$ helm install dashboard .
```

NAME: dashboard

LAST DEPLOYED: Thu Nov 25 20:37:51 2021

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

*** PLEASE BE PATIENT: kubernetes-dashboard may take a few minutes to install ***

Get the Kubernetes Dashboard URL by running:

```
export NODE_PORT=$(kubectl get -n default -o jsonpath="{.spec.ports[0].nodePort}" services dashboard-kubernetes-dashboard)
```

```
export NODE_IP=$(kubectl get nodes -o jsonpath="{.items[0].status.addresses[0].address}")
```

```
echo https://$NODE_IP:$NODE_PORT/
```

```
[student@master kubernetes-dashboard]$ kubectl get serviceaccounts
```

NAME	SECRETS	AGE
dashboard-kubernetes-dashboard	1	46s
default	1	5d21h

```
[student@katowice kubernetes-dashboard]$ kubectl create clusterrolebinding dashaccess --clusterrole=cluster-admin \
> --serviceaccount=default:dashboard-kubernetes-dashboard
clusterrolebinding.rbac.authorization.k8s.io/dashaccess created
```

Uwaga – jedna linia

```
[student@master kubernetes-dashboard]$ kubectl create \ clusterrolebinding dashaccess --  
clusterrole=cluster-admin \
```

```
--serviceaccount=default:dashboard-kubernetes-dashboard
```

```
clusterrolebinding.rbac.authorization.k8s.io/dashaccess created
```

Uwaga – jedna linia

```
[student@master kubernetes-dashboard]$ kubectl get clusterrolebinding | grep dashboard
```

```
dashboard-kubernetes-dashboard-metrics      ClusterRole/dashboard-kubernetes-dashboard-metrics  
3m39s
```

Uwaga – jedna linia

```
[student@master kubernetes-dashboard]$ kubectl get secrets | grep dashboard
```

```
dashboard-kubernetes-dashboard-certs      Opaque          0    28m  
dashboard-kubernetes-dashboard-token-kwfxp  kubernetes.io/service-account-token  3    28m  
kubernetes-dashboard-csrf                Opaque          1    28m  
kubernetes-dashboard-key-holder          Opaque          2    28m  
sh.helm.release.v1.dashboard.v1         helm.sh/release.v1  1    28m
```

Uwaga – jedna linia

```
[student@master kubernetes-dashboard]$ kubectl describe secrets dashboard-kubernetes-dashboard-  
token-kwfxp
```

```
Name:      dashboard-kubernetes-dashboard-token-kwfxp
```

```
Namespace: default
```

```
Labels:    <none>
```

```
Annotations: kubernetes.io/service-account.name: dashboard-kubernetes-dashboard
```

```
kubernetes.io/service-account.uid: 3e52af1e-dc81-4e1e-81b1-028fe676d5b6
```

```
Type: kubernetes.io/service-account-token
```

```
Data
```

```
====
```

ca.crt: 1099 bytes

namespace: 7 bytes

token:

```
eyJhbGciOiJSUzI1NiIsImtpZCI6IjBKT1dfX2xyZ3dRdy1pUHhwdUdhZDRaaHN0dmQ4ZnppWGRqRkcySURXZTgifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNIYWVWb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNIYWVWb3VudC9zZWNyZXQubmFtZSI6ImRhc2hib2FyZC1rdWJlcm5ldGVzLWRhc2hib2FyZC10b2t1bi1rd2Z4cCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWJfY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJkYXNoYm9hcmQta3ViZXJuZXRlcy1kYXNoYm9hcmQiLCJrdWJlcm5ldGVzLmVhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJkYXNoYm9hcmQta3ViZXJuZXRlcy1kYXNoYm9hcmQifQ.jlA83B1fwDv9E_5YaTK__KuZfMimgThvmtVhOjTjDrsH9OSGmlwAnXaBekaPbuEDuROjhqXDJxPbml-1rTXVy0nflTXuH6G6B9iutwUn02u42zQRoQcmJfjvewED9B8ACIfC74XlaBzy-0NWrl0MuqfcZMDhPQC6XqpwI3Dcbki_fqWIMnDL8qqIw1UUnObNNbpgH25H_vKxGCjoq2pukpfuKILZT5FquhbS6x3sCwd7kCRNR2KE5Z3DE3CnHtomhBUXuAnXtNnPW97flkQdgEN15tv_TljsdiKUgYE37uALwMf5qzMBxF_VasCKxOYCbQoep7r5RaICMmKelaxfJQ
```

Utwórz plik token_dashboard.txt i skopiuj do niego powyższy token:

```
[student@master kubernetes-dashboard]$ vim token_dashboard.txt
```

```
[student@master kubernetes-dashboard]$ cat token_dashboard.txt
```

token:

```
eyJhbGciOiJSUzI1NiIsImtpZCI6IjBKT1dfX2xyZ3dRdy1pUHhwdUdhZDRaaHN0dmQ4ZnppWGRqRkcySURXZTgifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNIYWVWb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNIYWVWb3VudC9zZWNyZXQubmFtZSI6ImRhc2hib2FyZC1rdWJlcm5ldGVzLWRhc2hib2FyZC10b2t1bi1rd2Z4cCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWJfY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJkYXNoYm9hcmQta3ViZXJuZXRlcy1kYXNoYm9hcmQiLCJrdWJlcm5ldGVzLmVhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJkYXNoYm9hcmQta3ViZXJuZXRlcy1kYXNoYm9hcmQifQ.jlA83B1fwDv9E_5YaTK__KuZfMimgThvmtVhOjTjDrsH9OSGmlwAnXaBekaPbuEDuROjhqXDJxPbml-1rTXVy0nflTXuH6G6B9iutwUn02u42zQRoQcmJfjvewED9B8ACIfC74XlaBzy-0NWrl0MuqfcZMDhPQC6XqpwI3Dcbki_fqWIMnDL8qqIw1UUnObNNbpgH25H_vKxGCjoq2pukpfuKILZT5FquhbS6x3sCwd7kCRNR2KE5Z3DE3CnHtomhBUXuAnXtNnPW97flkQdgEN15tv_TljsdiKUgYE37uALwMf5qzMBxF_VasCKxOYCbQoep7r5RaICMmKelaxfJQ
```

Prześlij token do katalogu domowego użytkownika student na maszynie base. Będzin potrzebny do wklejenia przez przeglądarkę w celu uwierzytelnienia:

Uwaga – jedna linia

```
[student@master kubernetes-dashboard]$ scp token_dashboard.txt
```

```
student@base:/home/student/token
```

The authenticity of host 'base (10.10.1.1)' can't be established.

ECDSA key fingerprint is SHA256:/bytdKejNsPZlDyt6cRs9rylvQnOSi8YUNRj9ASeD8Q.

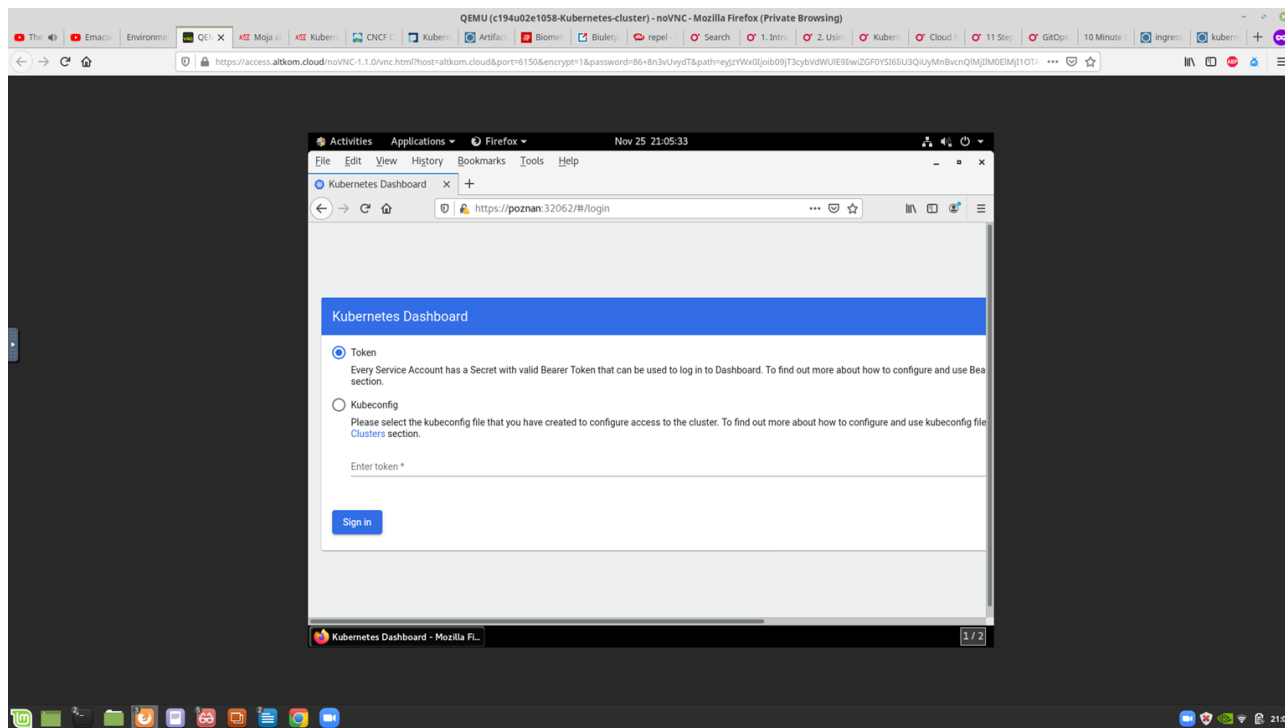
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

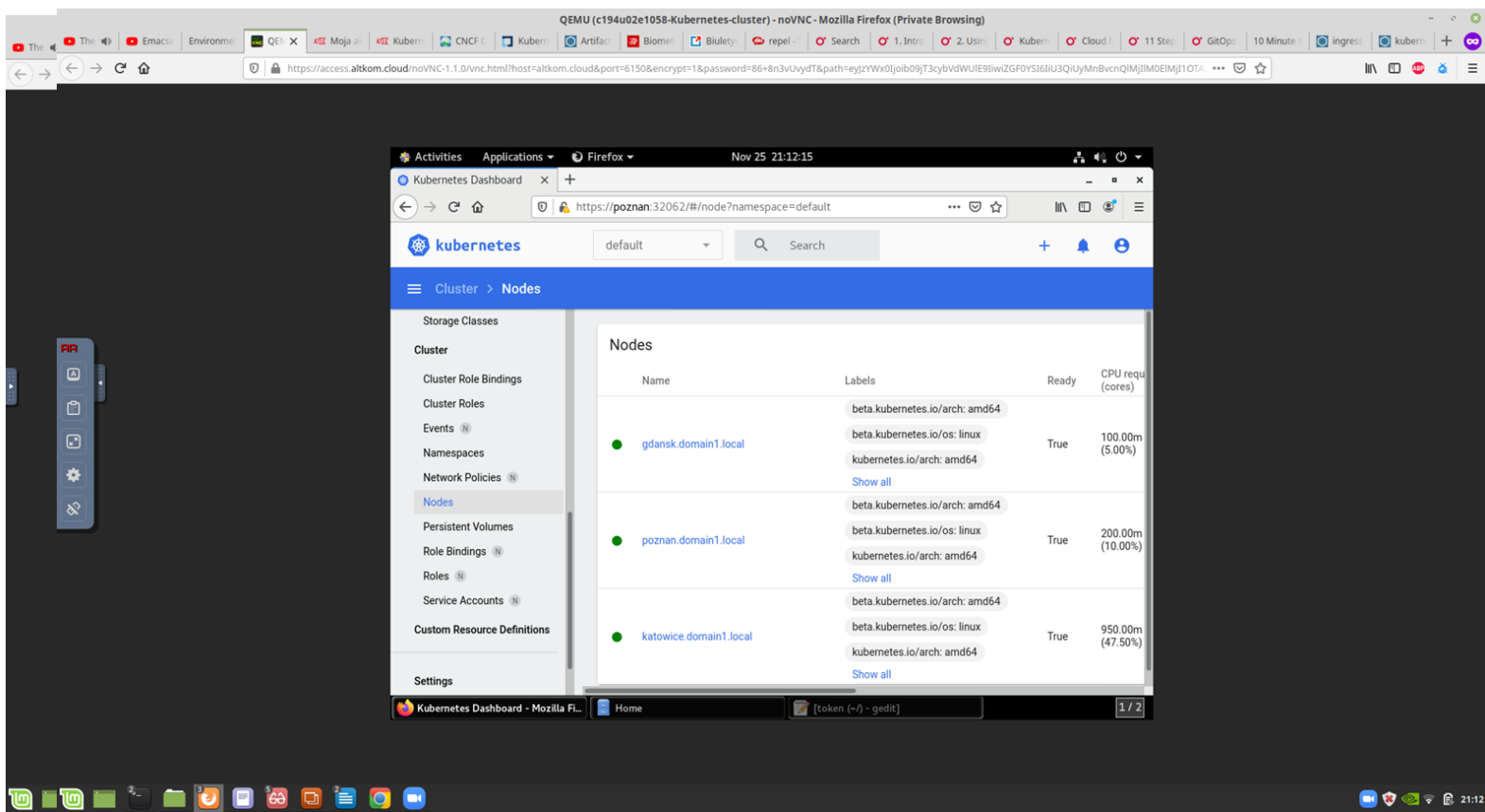
Warning: Permanently added 'base,10.10.1.1' (ECDSA) to the list of known hosts.

student@base's password:

token_dashboard.txt

Otwórz plik `/home/student/token` i przeklej token do pola "Enter token" i kliknij przycisk „Sign in”:





Miłej zabawy Dashboardem !