



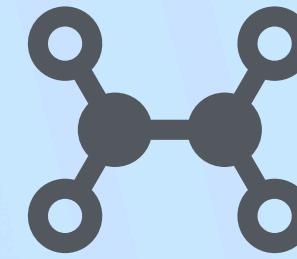
POLITECHNIKA  
LUBELSKA  
WYDZIAŁ ELEKTROTECHNIKI  
I INFORMATYKI

# Programowanie Full-Stack w Chmurze Obliczeniowej

## Instrukcja nr 11

Obszar zastosowania i konfiguracja wdrożeń opartych o obiekty typu StatefulSet. Konfiguracja aplikacji za pomocą ConfigMap. Zasady tworzenia ConfigMaps wykorzystujących wolumeny.

Dr inż. Sławomir Przyłucki  
[s.przylucki@pollub.pl](mailto:s.przylucki@pollub.pl)



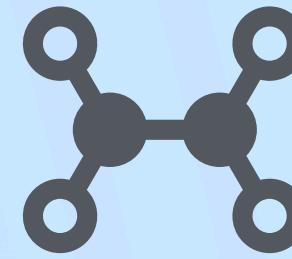
## Wdrażanie aplikacji w oparciu o obiekty StatefulSet

Obiekty StatefulSet **POWINNY** być wykorzystywane w przypadku aplikacji, które wymagają:

- stabilnych i unikalnych identyfikatorów sieciowych (w ramach struktury wewnętrznych połączeń K8s),
- stabilnego mechanizmu replikacji danych w systemach trwałego przechowywania danych,
- uporządkowanego (uwzględniającego kolejność działań) wdrażania i skalowania,
- uporządkowanej realizacji procesu rolling-updates (i bardziej rozbudowanych schematów uaktualniania)

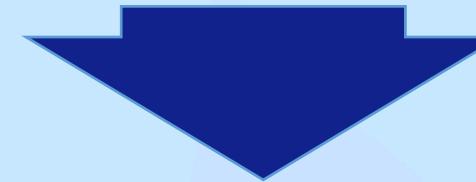


Jeśli powyższe wymagania nie dotyczą wdrażanej aplikacji - należy stosować metodę wdrażania aplikacji opartą o obiekty Deployment.



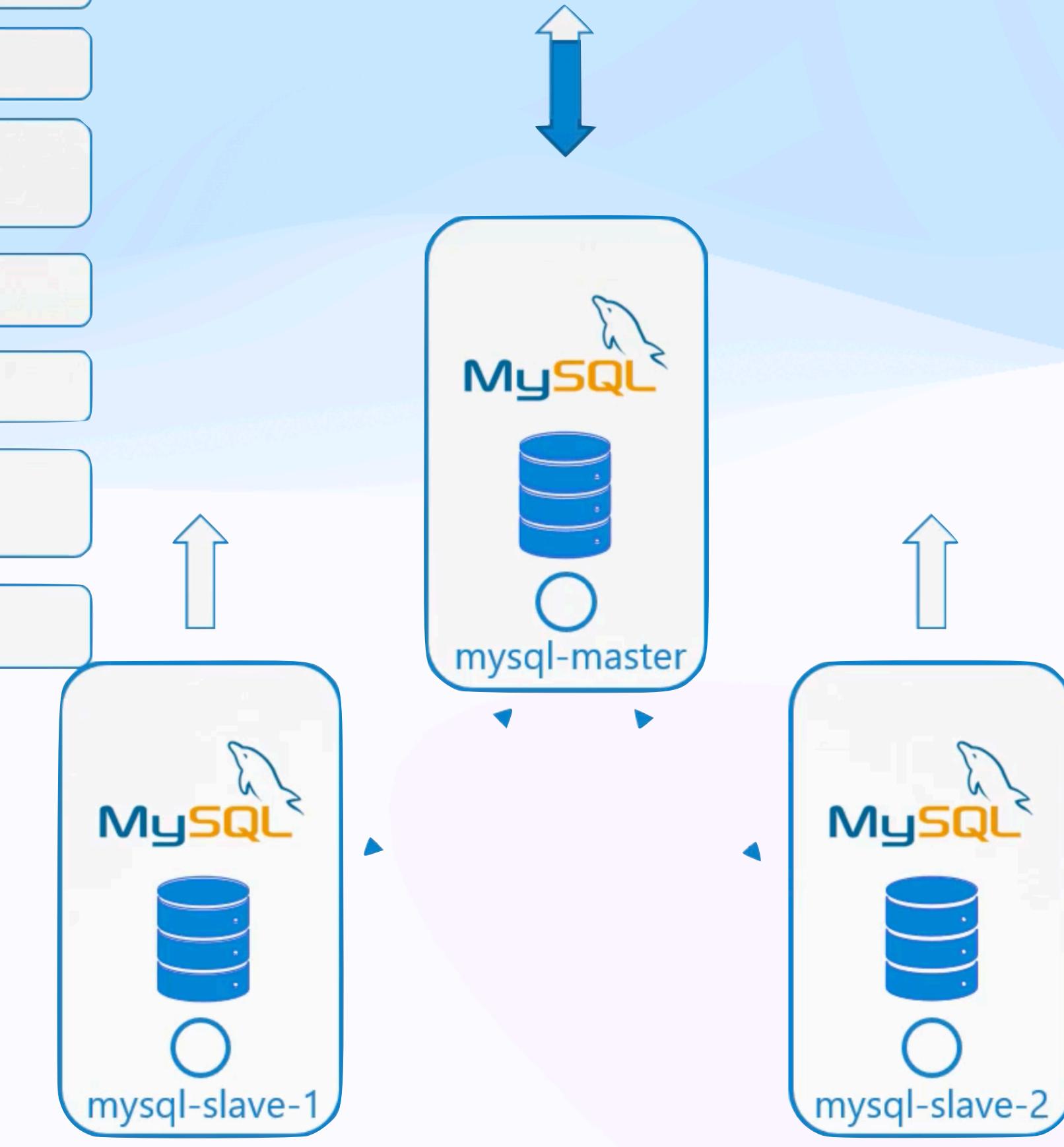
## Podstawy korzystania z obiektów StatefulSet - cz. I

Idea tworzenia i wykorzystania replik bazy danych (np. MySQL)



<https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/>

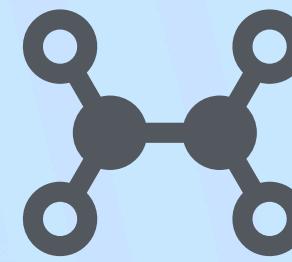
1. Setup master first and then slaves
2. Clone data from the master to slave-1
3. Enable continuous replication from master to slave-1
4. Wait for slave-1 to be ready
5. Clone data from slave-1 to slave-2
6. Enable continuous replication from master to slave-2
7. Configure Master Address on Slave



Niektóre elementy składowe aplikacji full-stack wymagają metod wdrażania oraz nadzoru nad cyklem życia odmiennych niż te, oferowane podczas typowego wykorzystania Deployment

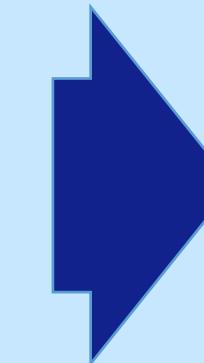
MASTER\_HOST=mysql-master

MASTER\_HOST=mysql-master



## Podstawy korzystania z obiektów StatefulSet - cz. II

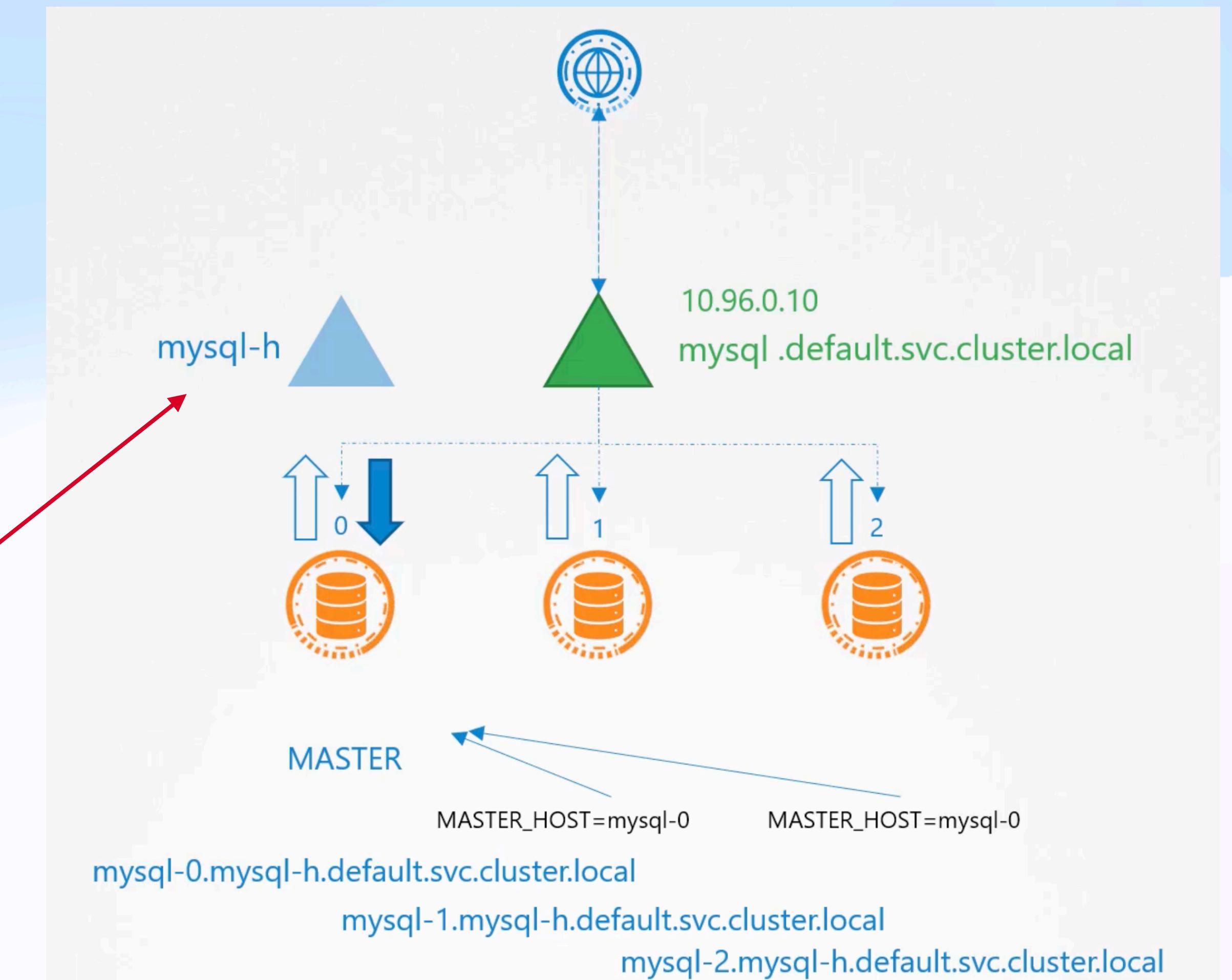
Idea usług (ang. Services) typu headless



<https://kubernetes.io/docs/concepts/services-networking/service/#headless-services>

```
headless-service.yml

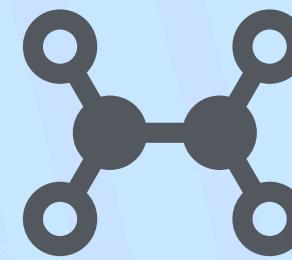
apiVersion: v1
kind: Service
metadata:
  name: mysql-h
spec:
  ports:
    - port: 3306
  selector:
    app: mysql
  clusterIP: None
```



Schemat budowy nazwy usług headless

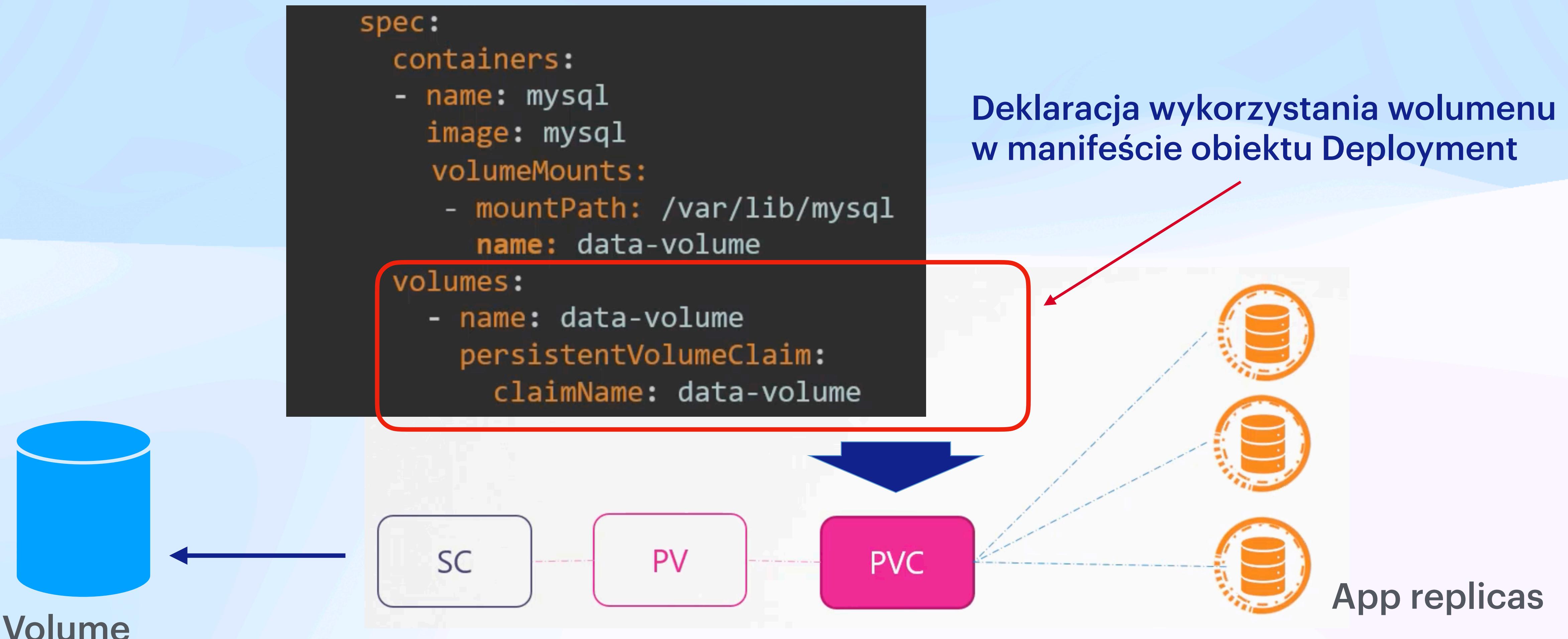


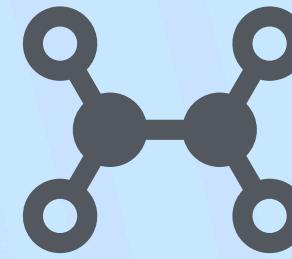
podname.headless-servicename.  
namespace.svc.cluster-domain



## Podstawy korzystania z obiektów StatefulSet - cz. III

Przypomnienie - klasyczna metoda wykorzystania wolumenów w Deployment (nie powinna być wykorzystywana w przypadku baz danych)



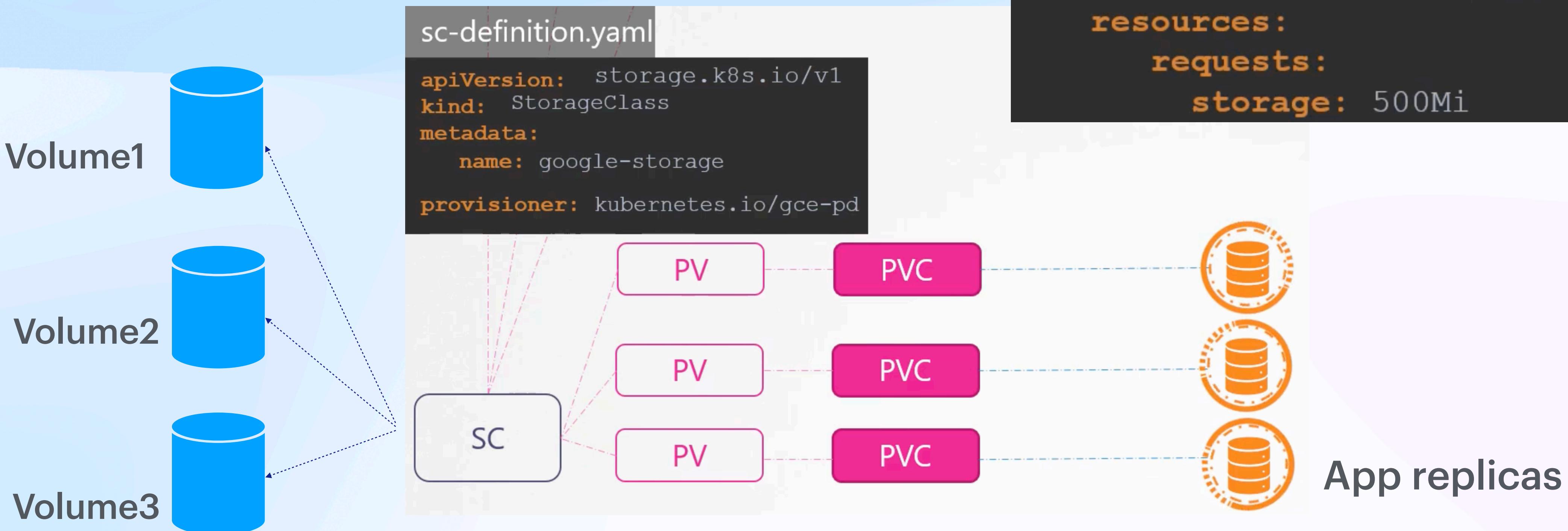


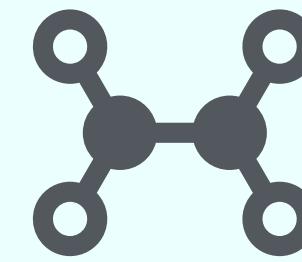
## Podstawy korzystania z obiektów StatefulSet - cz. IV

### Wykorzystanie wolumenów w StatefulSet

W manifeście obiektu StatefulSet należy umieścić dodatkową sekcję **volumeClaimTemplates**

<https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/#volume-claim-templates>





## Cechy obiektu StatefulSet - podsumowanie

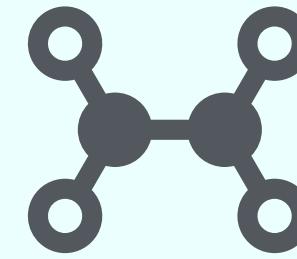
Głównym celem obiektów StatefulSet jest zapewnienie trwałej tożsamość (nazewnictwa) dla Pod-ów, a także realizację dedykowanej metody przechowywania danych dla tych Pod-ów.

Każdy Pod w ramach StatefulSet ma przypisywany identyfikator, który nie ulega zmianie podczas realizacji skalowania. Jednocześnie te identyfikatory uwzględniają kolejność tworzenia lub usuwania (ustawienie domyślne, które można wyłączyć)



### Korzystanie z obiektów StatefulSet WYMAGA:

- **Utworzenia usługi typu headless,**
- **Wcześniejszego skonfigurowania w klastrze StorageClass**



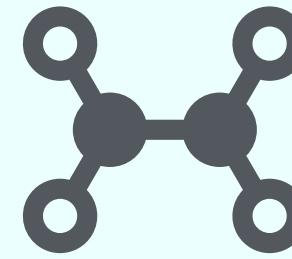
## Cechy obiektu StatefulSet - uwagi praktyczne

- Po usunięciu StatefulSet powiązane woluminy nie zostaną usunięte.
- Nie ma gwarancji, że Pody zostaną zatrzymane podczas usuwania StatefulSet i dlatego zaleca się wcześniejsze skalowanie w dół do zera Podów przed ostatecznym usunięciem obiektu StatefulSet.
- W trakcie wykonywania aktualizacji aplikacji (ang. Rolling Updates) z ustawieniem domyślnym polityki zarządzania Pod-ami (OrderedReady) nie można wykluczyć, że Pod-y znajdą się w stanie zatrzymania (zgłoszą błąd) i naprawa tego stanu nie jest możliwa automatycznie - konieczne jest ręczne usunięcie problemu.

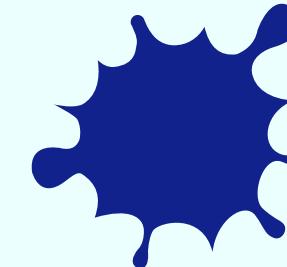


Proszę KONIECZNIE zapoznać się z przykładami konfigurowania i wykorzystania StatefulSet, jakie są dostępne pod adresem:

<https://kubernetes.io/docs/tutorials/stateful-application/basic-stateful-set/>



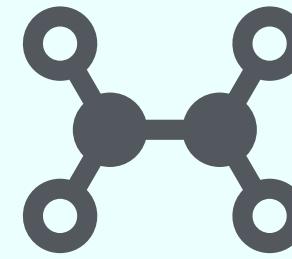
## Przykład konfiguracji StatefulSet - cz. I



Pliki manifestów yaml wykorzystywane w dalszych przykładach dostępne są na MSTeams, plik: **lab11\_examples.zip**

- 1 Zgodnie z informacjami z poprzednich slajdów, konieczne jest sprawdzenie czy zdefiniowany jest StorageClass. Jeśli tak, można skonfigurować StatefulSet.
- 2 Należy zapoznać się z treścią pliku yaml o nazwie **stateful.yaml**. Deklaruje on usługę headless oraz obiekt StatefulSet na bazie obrazu nginx. Następnie należy uruchomić w/w manifest yaml.
  - Proszę pamiętać, że zasoby uruchamiane są na jednowęzłowym klastrze, dlatego należy rozsądnie gospodarować dostępnymi zasobami (ilość replik, wielkość wolumenu).
  - W praktycznych rozwiązaniach wymagane jest dysponowanie klastrem z rozproszonym (wielowęzłowym) systemem przechowywania danych, np. Ceph, Longhorn itd. i skonfigurowanym dla nich StorageClass.
  - Przy usuwaniu obiektu StatefulSet proszę KONIECZNIE pamiętać, że utworzone PV nie są automatycznie usuwane i należy je usunąć ręcznie.





## Przykład konfiguracji StatefulSet - cz. II

3

Należy sprawdzić poprawność utworzenia usługi headless.

```
~/lab13
> kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes  ClusterIP  10.96.0.1   <none>       443/TCP   2d18h
nginx     ClusterIP  None        <none>       80/TCP    12m

~/lab13
> kubectl get endpoints -l app=nginx
NAME    ENDPOINTS   AGE
nginx  10.244.120.88:80,10.244.120.97:80  16m
```

4

Należy sprawdzić poprawność utworzenia aplikacji typu StatefulSet.

```
~/lab13
> kubectl get statefulset
NAME  READY  AGE
web   2/2   19m
```

```
~/lab13
> kubectl get pods
NAME  READY  STATUS  RESTARTS  AGE
mypad  1/1   Running  17 (9m43s ago)  2d18h
mysapod 1/1   Running  15 (9m44s ago)  2d16h
web-0   1/1   Running  0          20m
web-1   1/1   Running  0          19m
```

5

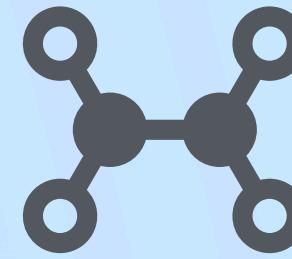
Należy sprawdzić konfigurację wolumenów skojarzonych (ang. Bind) z Pod-ami utworzonymi przez obiekt Statefulset



```
~/lab13
> kubectl get pv,pvc
NAME                                     CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM  STORAGECLASS  REASON  AGE
persistentvolume/pvc-40e392b6-2a5a-430f-8ea0-a51eed08881c  500Mi   RWO            Delete        Bound  default/www-web-0 standard  25m
persistentvolume/pvc-8bec43f2-7dbf-4b90-80ad-a80f179b66bb  500Mi   RW0            Delete        Bound  default/www-web-1 standard  59m

NAME           STATUS  VOLUME
persistentvolumeclaim/www-web-0  Bound   pvc-40e392b6-2a5a-430f-8ea0-a51eed08881c
persistentvolumeclaim/www-web-1  Bound   pvc-8bec43f2-7dbf-4b90-80ad-a80f179b66bb
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
www-web-0	Bound	pvc-40e392b6-2a5a-430f-8ea0-a51eed08881c	500Mi	RWO	standard	25m
www-web-1	Bound	pvc-8bec43f2-7dbf-4b90-80ad-a80f179b66bb	500Mi	RW0	standard	59m



## Rola ConfigMaps w środowisku Kubernetes

Podstawowym paradrygmatem w K8s jest **oddzielenie informacji specyficznych dla danego środowiska (klastra) od kodu aplikacji**. Innymi słowy, kod powinien być statyczny, co czyni go przenośnym, a zmienne środowiskowe opisują dane wdrożenia i powinny być dostarczane z zewnątrz, w trakcie danego wdrożenia.

**Obiekty ConfigMap są implementacją w/w paradrygmatu.**

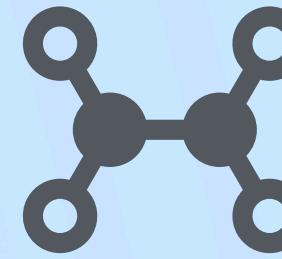


Podstawowe informacje na temat ConfigMap są dostępne w dokumentacji Kubernetes:  
<https://kubernetes.io/docs/concepts/configuration/configmap/>

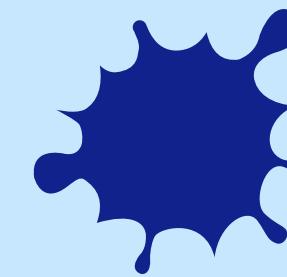


### BARDZO WAŻNA UWAGA:

**Caution:** ConfigMap does not provide secrecy or encryption. If the data you want to store are confidential, use a Secret rather than a ConfigMap, or use additional (third party) tools to keep your data private.



## ConfigMap - przykład wykorzystania - cz. I



Wykorzystanie ConfigMap wiąże się z dwoma zadaniami:

- utworzeniem obiektu ConfigMap,
- wywołaniem utworzonego ConfigMap w spec pod-a



Jeśli aplikacja odwołuje się do ConfigMap, ConfigMap **MUSI** istnieć w klastrze przed uruchomieniem tej aplikacji

### Tworzenie ConfigMap - polecenie imperatywne

```
MacBook-Pro:~ user$ kubectl create configmap -h  
Create a config map based on a file, directory, or specified literal value.
```

A single config map may package one or more key/value pairs.

```
minikube on minikube (default)
```

When creating a config map based on a file, the key will default to the basename of the file, and the value will default to the file content. If the basename is an invalid key, you may specify an alternate key.

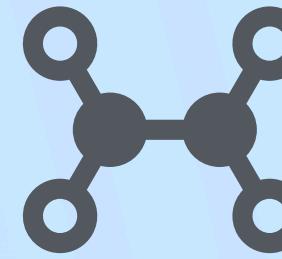
When creating a config map based on a directory, each file whose basename is a valid key in the directory will be packaged into the config map. Any directory entries except regular files are ignored (e.g. subdirectories, symlinks, devices, pipes, etc).

Aliases:  
configmap, cm

III-cia metoda

II-ga metoda

I-sza metoda



## ConfigMap - przykład wykorzystania - cz. II

1

Należy zwolnić zasoby klastra: `kubectl delete all --all`

Następnie należy utworzyć plik o nazwie `vars.file`, który zawiera następujące deklaracje zmiennych:

```
MySQL_ROOT_PASSWORD=password  
MySQL_USER=student
```

2

Na podstawie pliku `vars.file` należy utworzyć ConfigMap o nazwie `mydbvars`.

3

Należy sprawdzić poprawność utworzenia obiektu ConfigMaps `mydbvars`

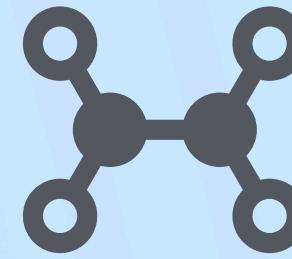
4

Należy wyświetlić zawartość utworzonej ConfigMaps

5

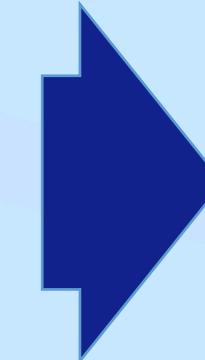
Należy utworzyć Deployment o nazwie `mydb` wykorzystujący obraz `mariadb` oraz zawierający 3 repliki

```
~  kubectl describe cm mydbvars  
Name:          mydbvars  
Namespace:     default  
Labels:        <none>  
Annotations:   <none>  
  
Data  
====  
MySQL_ROOT_PASSWORD:  
-----  
password  
MySQL_USER:  
-----  
student  
BinaryData  
====  
  
Events:  <none>
```



## ConfigMap - przykład wykorzystania - cz. III

Deployment uruchomiony w p. 5 nie został uruchomiony poprawnie  
(przyczyna powinna być oczywista)

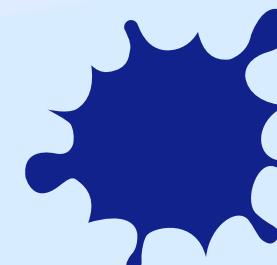


kubectl get all --selector app=mydb					
NAME	READY	STATUS	RESTARTS	AGE	
pod/mydb-7bbb588c85-nvgd7	0/1	CrashLoopBackOff	3 (14s ago)	62s	
pod/mydb-7bbb588c85-qnflp	0/1	CrashLoopBackOff	3 (15s ago)	62s	
pod/mydb-7bbb588c85-ttchv	0/1	Error	3 (40s ago)	62s	
NAME	READY	UP-TO-DATE	AVAILABLE	AGE	
deployment.apps/mydb	0/3	3	0	62s	
NAME	DESIRED	CURRENT	READY	AGE	
replicaset.apps/mydb-7bbb588c85	3	3	0	62s	

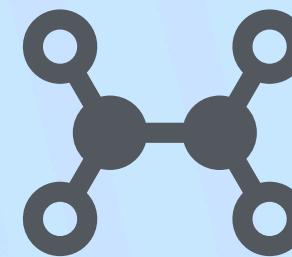
6 Należy przypisać zmienne zawarte w ConfigMap *mydbvars* do Deployment *mydb*

kubectl set env deploy mydb --from=configmap/mydbvars

7 Należy sprawdzić czy update Deployment-u *mydb* przebieg bez problemów i czy aplikacja działa poprawnie.



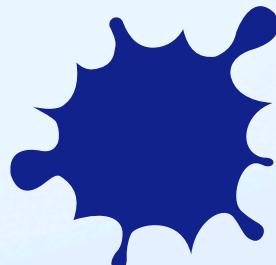
**WNIOSEK PRAKTYCZNY:** Wykorzystanie metody imperatywnej jest jak najbardziej możliwe i poprawne ale w praktyce deklaracje zmiennych/pliki konfiguracyjne deklarowane są w postaci manifestów yaml



## ConfigMap - przykład wykorzystania - cz. IV (metoda deklaratywna)

```
config-map.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  APP_COLOR: blue
  APP_MODE: prod
```

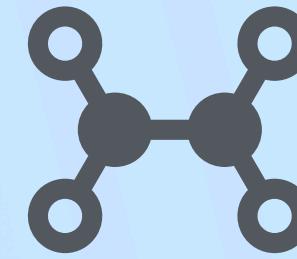
```
pod-definition.yaml
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp-color
  labels:
    name: simple-webapp-color
spec:
  containers:
  - name: simple-webapp-color
    image: simple-webapp-color
    ports:
    - containerPort: 8080
      envFrom:
        configMapRef:
          name: app-config
```



**DOBRA PRAKTYKA:** Proszę manify dla obiektów ConfigMaps powinny być nazywane w taki sposób by było jasne ich powiązanie z daną aplikacją oraz typem wdrożenia.

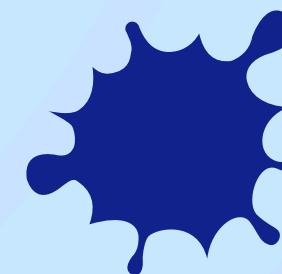
8

Proszę wygenerować plik yaml dla uruchomionego Deployment *mydb*

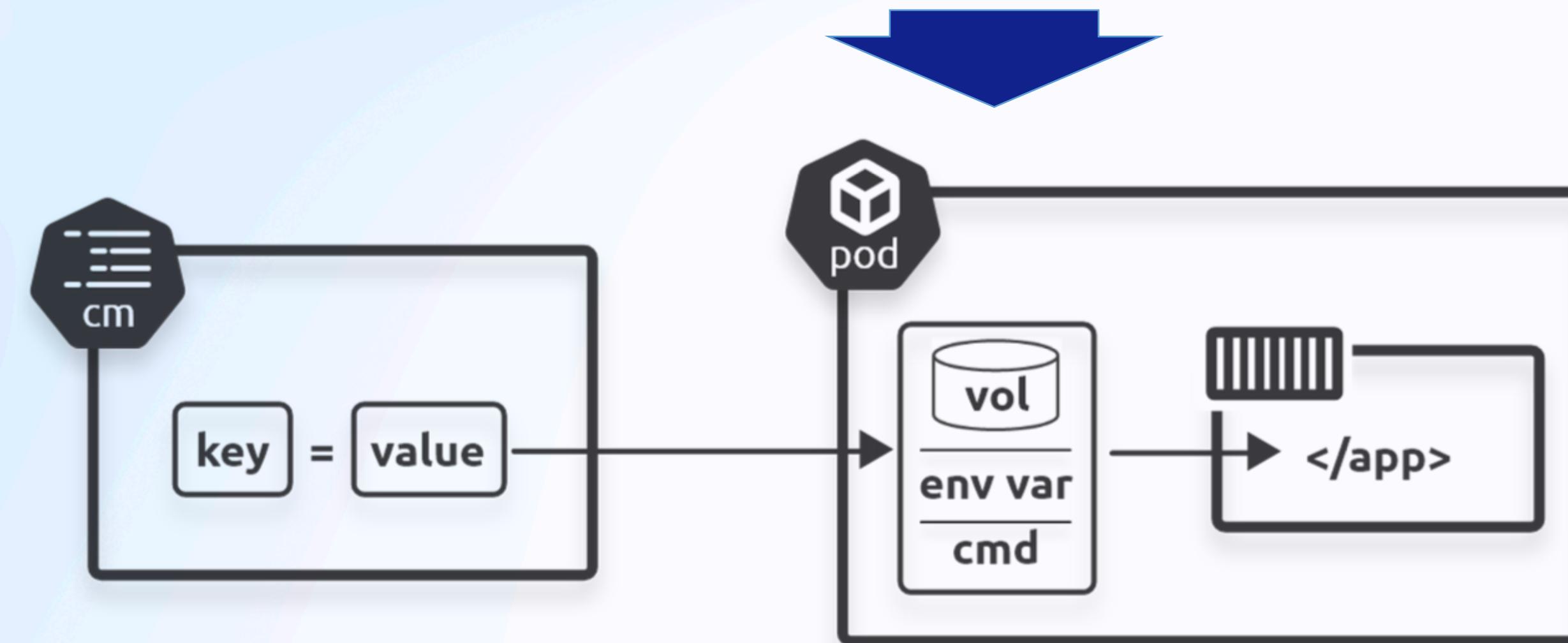


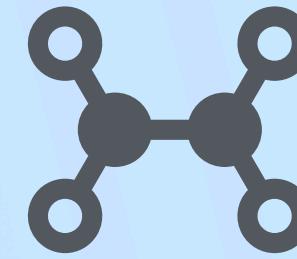
## ConfigMap - wykorzystanie wolumenów - cz. I

Poprzednie przykłady dotyczą częstego przypadku dołączania do aplikacji zmiennych środowiskowych. **PROSZĘ PAMIĘTAĆ**, że zmienne te są zapisywane w bazie etcd i w tym przypadku jest narzucone ograniczenie na wielkość tworzonych obiektów ConfigMap.



W przypadku większej liczby zmiennych lub gdy niezbędny jest plik (pliki) konfiguracyjne dla danej aplikacji to należy wykorzystać metodę tworzenia ConfigMap z pliku lub z katalogu. **WAŻNE:** W takim przypadku ConfigMap są przechowywane na dedykowanym wolumenie





## ConfigMap - wykorzystanie wolumenów - cz. II

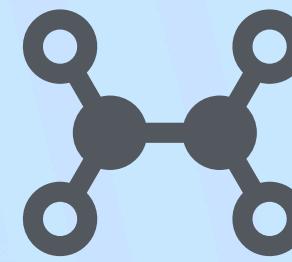


- Jeśli ConfigMap jest tworzony z pliku lub wszystkich plików w danym katalogu, to aby móc go wykorzystać w aplikacji, w/w wolumen musi być „zamountowany” poprzez odpowiednie deklaracje w manifeście aplikacji.
- Nie ma metody imperatywnej do montowania wolumenu do istniejącego Deployment-u/Pod-a. Wobec tego należy używać metody deklaratywnej (manifesty yaml).



**BARDZO przydatny fragment dokumentacji:**

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/>



## ConfigMap - wykorzystanie wolumenów - cz. II

1

Należy utworzyć plik o nazwie *index.html* zawierający tekst: *Kubernetes jest fajny*  
Następnie, na podstawie tego pliku należy utworzyć ConfigMap o nazwie *myindex*

2

Wygenerujmy plik yaml dla Deployment o nazwie *myweb* wykorzystujący obraz *nginx* oraz jedną replikę - proszę użyć opcji *dry-run=client* (nazwa pliku manifestu: *myweb.yaml*)

3

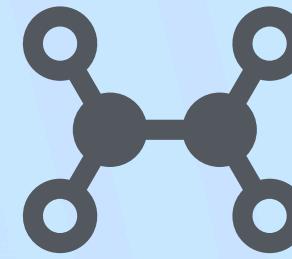
Wykorzystując przykład z poniższego linku:

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#populate-a-volume-with-data-stored-in-a-configmap>

należy:

- (A) w template dla pod-a zdeklarować użycie wolumenu o nazwie *index-vol* i przypisać go do utworzonego ConfigMap o nazwie *myindex*
- (B) „zamontować” ten wolumen w ścieżce /usr/share/nginx/html

```
~/Labs/lab13
> echo "Kubernetes jest fajny" > index.html
~/Labs/lab13
> kubectl create cm myindex --from-file=index.html
configmap/myindex created
~/Labs/lab13
> kubectl get cm
NAME          DATA   AGE
kube-root-ca.crt  1    133m
mydbvars        2    74m
myindex         1    17s
```



PFSwChO laboratorium 11

## ConfigMap - wykorzystanie wolumenów - cz. III

Plik `myweb.yaml` po wykonaniu edycji z p. 3 na poprzednim slajdzie

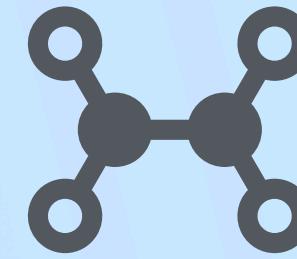
4

Wykorzystując poznane polecenia, proszę sprawdzić poprawność dołgczenia pliku konfiguracyjnego w postaci ConfigMap o nazwie `myindex`.

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   creationTimestamp: null
5   labels:
6     app: myweb
7     name: myweb
8   spec:
9     replicas: 1
10    selector:
11      matchLabels:
12        app: myweb
13    strategy: {}
14    template:
15      metadata:
16        creationTimestamp: null
17        labels:
18          app: myweb
19      spec:
20        volumes:
21          - name: index-vol
22            configMap:
23              name: myindex
24        containers:
25          - image: nginx
26            name: nginx
27            volumeMounts:
28              - name: index-vol
29                mountPath: /usr/share/nginx/html
30            resources: {}
31 status: {}
```

(A)

(B)



## Zadanie - NIEOBOWIĄZKOWE - cz. I

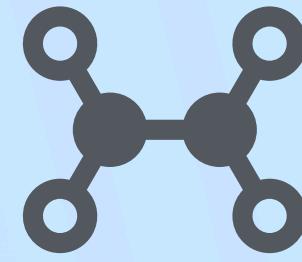
Obiekty StatefulSet, podobnie jak obiekty Deployment można skalować. Różnica polega na uwzględnieniu specyficznych zasad tworzenia kolejnej kopii źródłowej aplikacji (np. bazy danych) opartej o uporządkowaną kolejność uruchamiania i synchronizacji kolejnych podów i ich zależności. Wyjaśnienie poparte przykładami można znaleźć w dokumentacji Kubernetes, np. :

<https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/#deployment-and-scaling-guarantees>

<https://kubernetes.io/docs/tutorials/stateful-application/basic-stateful-set/#scaling-a-statefulset>

Jak również w niezählonych zasobach internetowych, np:

<https://medium.com/@walissonscd/scaling-statefulsets-managing-your-pods-efficiently-d5f934605b8a>



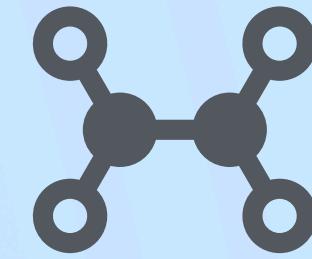
## Zadanie - NIEOBOWIĄZKOWE - cz. II

W sprawozdaniu z wykonania zadania należy:

- (A) Na podstawie dokumentacji wyjaśnić b. krótko jaka jest różnica pomiędzy skalowaniem StatefulSet a Deployment (odpowiedź powinna zawierać odniesienie do skalowania „w górę” jak i „w dół”)
- (B) Utworzyć zestaw manifestów niezbędny do uruchomienia bazy danych MySQL (lub dowolnej innej) metodą deklaratywną.
- (C) Na podstawie wdrożenia aplikacji z punktu B, w oparciu o samodzielnie dobrane przykłady, należy potwierdzić zasady skalowania opisane w punkcie A

Preferowana forma sprawozdania, repozytorium git (np. Github) z opracowanymi plikami manifestów oraz plikiem README.md zawierającym niezbędne opisy. Alternatywnie, plik pdf/docx.

**VERTE !!!!!**



PFSwChO laboratorium 11

## Zadanie - NIEOBOWIĄZKOWE - cz. III

Wykonanie sprawozdania jest **NIEOBOWIĄZKOWE**, termin realizacji poda prowadzący.  
Ocena rozwiązania zostanie uwzględniona w propozycji oceny końcowej z laboratorium.

### ALTERNATYWNIĘ:

*W przypadku gdy propozycja oceny końcowej będzie niezadowalająca dla studenta lub (szczególnie istotne) dla prowadzącego a zadanie nie zostało wcześniej wykonane to w trakcie ostatniego laboratorium można będzie podwyższyć ocenę końcową poprzez zaprezentowanie i omówienie rozwiązania tego zadania.*