

Laboratorium nr.:	10	Data laboratorium:	1.12.2021
Temat:	Składowanie danych w klastrze aplikacyjnym		
Autorraportu:	Kacper Guzik		
Grupa:	185IC		
Grupa lab.	A1	Data raportu:	1.12.2021
Subiektywna ocena trudności laboratorium (nie wpływa na ocenę pracy!) [1-łatwe, 10-trudne]			10

Q-1. Na podstawie dokumentacji projektu kubernetes (kubernetes.io/docs) odpowiedz na pytania:

- **hostPath** - Jest realizowany w taki sposób że montuje plik lub katalog z host node's filesystem do naszego poda. Nie jest to coś co większość podów potrzebuje ale zapewnia fajne obejście dla niektórych aplikacji. Aby skorzystać z tego PV musimy wcześniej ten katalog z plikiem stworzyć, inaczej pod nie wystartuje poprawnie. Rozwiązanie tego typu nie jest w ogóle bezpieczne i należy go nie używać dlatego że można w ten sposób ujawnić ważne dane uwierzytelniające. Ograniczenia są takie że pody z identyczną konfiguracją mogą zadziałać inaczej na innych węzłach. Typy dostępu: ReadWriteOnce
- **Fc** – FC jest realizowany przez 'fibre channel block channel', specyfikując jeden lub więcej WWN's. Aby użyć PV trzeba wpierw skonfigurować strefę FC SAN. Aby uzyskać dostęp persistence volume trzeba być w tej samej przestrzeni nazw użytkownika. Typy dostępu: ReadWriteOnce, ReadOnlyMany
- **vsphereVolume** – vsphereVolume montuje wolumen w podzie. Do korzystania jest wymagane skonfigurowane KubernetesvSphere oraz utworzyć wolumen vSphere VMDK. Typy dostępu: ReadWriteOnce
- **awsElasticBlockStore(EBS)** - montuje wolumen AWS do naszego poda. Kiedy pod jest usuwany, EBS volume wciąż istnieje a jest jedynie odnotowywany od poda. Dzięki temu EBS może być wstępnie wypełniony danymi i udostępniać je między podami. Aby użyć tego PV wcześniej musimy utworzyć wolumin na AWS. Ograniczenia EBS, węzły na których działają pody muszą być instancjami AWS EC2, instancje muszą się znajdować w tym samym regionie i strefie dostępu co PV EBS, EBS obsługuje tylko jedną instancję EC2 montującą wolumen. Typy dostępu: ReadWriteOnce
- **Nfs** - Wolumin nfs umożliwia zamontowanie istniejącego udziału NFS (Network File System) w pode. W przeciwieństwie do emptyDir, który jest wymazywany po usunięciu Poda, zawartość woluminu nfs jest zachowywana, a wolumin jest po prostu odmontowany. Oznacza to, że wolumin NFS może być wstępnie wypełniony danymi i że dane mogą być udostępniane między zasobnikami. NFS może być montowany jednocześnie przez wielu autorów. Wymagane jest posiadanie własnego NFS serwera. Typy dostępu: ReadWriteOnce, ReadOnlyMany, ReadWriteMany

Q-2. W raporcie umieść wyniki sprawdzania dostępności pliku i katalogu w podzie PIERWSZYM, DRUGIM i TRZECIM.

```
student@env4-WKSnode-lab: ~/lab10
-ld /dane/
-bash: nazwa_PIERWSZEGO_poda: No such file or directory
student@env4-WKSnode-lab:~/lab10$ kubectl exec it bb-nostorage-57c5b799db-t8sgz
-- ld -ld /dane/
Error from server (NotFound): pods "it" not found
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz
-- ld -ld /dane/
[ERROR] exec failed: container_linux.go:380: starting container process cause
d: exec: "ld": executable file not found in $PATH
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it <nazwa_PIERWSZEGO_poda> -- ls -ld /dane/
-bash: nazwa_PIERWSZEGO_poda: No such file or directory
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz -- ls -ld /dane/
drwxr-xr-x 2 root root 23 Dec 20 15:39 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz
bb-nostorage-57c5b799db-t8sgz: command not found
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz -- sh
/ # mkdir /dane
mkdir: can't create directory '/dane': File exists
/ # echo "Dane testowe" > /dane/plik.txt
/ # exit
student@env4-WKSnode-lab:~/lab10$ echo "Dane testowe" > /dane/plik.txt
-bash: /dane/plik.txt: No such file or directory
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz -- cat /dane/plik.txt
Dane testowe
student@env4-WKSnode-lab:~/lab10$ kubectl scale --replicas=2 deployment bb-nostorage
deployment.apps/bb-nostorage scaled
student@env4-WKSnode-lab:~/lab10$ kubectl get pod
NAME READY STATUS RESTARTS AGE
bb-nostorage-57c5b799db-g2g57 1/1 Running 0 16s
bb-nostorage-57c5b799db-t8sgz 1/1 Running 0 7m47s
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz -- ls -ld /dane/
drwxr-xr-x 2 root root 39 Dec 20 15:44 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-t8sgz -- cat /dane/plik.txt
Dane testowe
student@env4-WKSnode-lab:~/lab10$ kubectl delete bb-nostorage-57c5b799db-t8sgz
error: the server doesn't have a resource type "bb-nostorage-57c5b799db-t8sgz"
student@env4-WKSnode-lab:~/lab10$ kubectl delete pod bb-nostorage-57c5b799db-t8sgz
pod "bb-nostorage-57c5b799db-t8sgz" deleted
student@env4-WKSnode-lab:~/lab10$ kubectl get pod
NAME READY STATUS RESTARTS AGE
bb-nostorage-57c5b799db-cntc4 1/1 Running 0 50s
bb-nostorage-57c5b799db-g2g57 1/1 Running 0 2m13s
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-cntc4 -- ls -ld /dane/
ls: /dane/: No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-cntc4 -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-g2g57 -- ls -ld /dane/
ls: /dane/: No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-nostorage-57c5b799db-g2g57 -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$
```

opis zaobserwowanej sytuacji – W pierwszym podzie był dostępny katalog jak i zawartość pliku. W podzie drugim i trzecim katalog jak i plik nie istniał.

Wnioski – Umieszczając bezpośrednio danę wewnątrz poda, dane te podą dostępne tylko i wyłącznie w tym podzie, nie ma to wpływu na inne pody. Po usunięciu tego poda tracimy je na zawsze.

Q-3. . W raporcie umieść wyniki sprawdzania dostępności pliku i katalogu w podzie PIERWSZYM, DRUGIM i TRZECIM.

```

student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- ls -ld /dane/
drwxrwxrwx 2 root root 6 Dec 20 16:20 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-jtzbq -- sh
$ ls -ld /dane/
drwxrwxrwx 2 root root 22 Dec 20 16:19 /dane/
$ mkdir /dane
mkdir: can't create directory '/dane': File exists
$ echo "Dane testowe" > /dane/plik.txt
$ exit
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-jtzbq -- sh
$ echo "Dane testowe" > /dane/plik.txt
$ exit
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- cat /dane/plik.txt^C
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-jtzbq -- cat /dane/plik.txt
Dane testowe
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- ls -ld /dane/
Error: unknown shorthand flag: 'l' in -ld
See 'kubectl exec --help' for usage.
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- ls -ld /dane/
drwxrwxrwx 2 root root 6 Dec 20 16:20 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl get pod
NAME READY STATUS RESTARTS AGE
bb-emptydir-78cb4d95ff-8zcj1 1/1 Running 0 5m17s
bb-emptydir-78cb4d95ff-jtzbq 1/1 Running 0 8m52s
bb-nostorage-57c5b799db-cntc4 1/1 Running 0 39m
bb-nostorage-57c5b799db-g2g57 1/1 Running 0 40m
student@env4-WKSnode-lab:~/lab10$ kubectl delete pod ^C
student@env4-WKSnode-lab:~/lab10$ kubectl delete pod bb-emptydir-78cb4d95ff-jtzbq
pod "bb-emptydir-78cb4d95ff-jtzbq" deleted
student@env4-WKSnode-lab:~/lab10$
student@env4-WKSnode-lab:~/lab10$ kubectl get pod
NAME READY STATUS RESTARTS AGE
bb-emptydir-78cb4d95ff-8zcj1 1/1 Running 0 6m12s
bb-emptydir-78cb4d95ff-zjj4q 1/1 Running 0 32s
bb-nostorage-57c5b799db-cntc4 1/1 Running 0 40m
bb-nostorage-57c5b799db-g2g57 1/1 Running 0 41m
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-zjj4q -- ls -ld /dane/
drwxrwxrwx 2 root root 6 Dec 20 16:26 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-zjj4q -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- ls -ld /dane/
drwxrwxrwx 2 root root 6 Dec 20 16:20 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-emptydir-78cb4d95ff-8zcj1 -- cat /dane/plik.txt
cat: can't open '/dane/plik.txt': No such file or directory
command terminated with exit code 1
student@env4-WKSnode-lab:~/lab10$

```

- opis zaobserwowanej sytuacji – Tworząc katalogi i plik w pierwszym podzie. Katalog będzie widoczny dla poda 2 i 3 ale plik już nie, tylko pod 1 będzie mógł odczytać plik.

- wnioski – Folder jest widoczny dla wszystkich 3 podów dlatego że został przygotowany w pliku.yaml. Ale plik który został już stworzony tylko w podzie 1, nie będzie widoczny dla reszty podów. EmptyDir daje możliwość współdzielenia plików tylko w tym samym podzie.

Q-4. W raporcie przytocz linie z odpowiedzi platformy wskazujące na naturę problemu z którym tu mamy do czynienia.

```

student@env4-WKSnode-lab:~/lab10$ kubectl describe replicaset -l app=bb-hostpath
Name:         bb-hostpath-78d78656f5
Namespace:    labproj04
Selector:     app=bb-hostpath,pod-template-hash=78d78656f5
Labels:       app=bb-hostpath
              pod-template-hash=78d78656f5
Annotations:  deployment.kubernetes.io/desired-replicas: 1
              deployment.kubernetes.io/max-replicas: 2
              deployment.kubernetes.io/revision: 1
Controlled By: Deployment/bb-hostpath
Replicas:     0 current / 1 desired
Pods Status:  0 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=bb-hostpath
           pod-template-hash=78d78656f5
  Containers:
    busybox:
      Image:      image-registry.openshift-imageregistry.svc:5000/default/busybox:latest
      Port:       <none>
      Host Port:  <none>
      Command:
        sleep
        9000
      Environment: <none>
      Mounts:
        /dane from test-volume (rw)
  Volumes:
    test-volume:
      Type:      HostPath (bare host directory volume)
      Path:      /dane-lab
      HostPathType: Directory
Conditions:
  Type           Status  Reason
  ----           -
  ReplicaFailure  True    FailedCreate

Events:
  Type    Reason      Age    From          Message
  ----    -
  Warning  FailedCreate 38s (x15 over 2m) replicaset-controller Error creating pods "bb-hostpath-78d78656f5-": is forbidden: unable to validate against a security context constraint: [spec.volumes[0]: Invalid value: "hostPath": hostPath volumes are not allowed to be used, provider "nonroot": Forbidden: not usable by user or serviceaccount, provider "noobaa": Forbidden: not usable by user or serviceaccount, provider "noobaa-endpoint": Forbidden: not usable by user or serviceaccount, provider "hostmount-anyuid": Forbidden: not usable by user or serviceaccount, provider "machine-api-termination-handler": Forbidden: not usable by user or serviceaccount, provider "hostnetwork": Forbidden: not usable by user or serviceaccount, provider "hostaccess": Forbidden: not usable by user or serviceaccount, provider "rook-ceph": Forbidden: not usable by user or serviceaccount, provider "node-exporter": Forbidden: not usable by user or serviceaccount, provider "privileged": Forbidden: not usable by user or serviceaccount, provider "rook-ceph-csi": Forbidden: not usable by user or serviceaccount]

```

hostPath volumes are not allowed to be used, provider 'nonroot'

Providerem może być tylko root?

Q-5. Na podstawie pola PROVISIONER spróbuj ustalić mechanizm implementujący daną StorageClass dla dwóch S.C. spośród wylistowanych powyżej

openshift-storage.rbd.csi.ceph.com


```

apiVersion: v1
kind: PersistentVolume
metadata:
  name: ceph-pv ❶
spec:
  capacity:
    storage: 2Gi ❷
  accessModes:
    - ReadWriteOnce ❸
  rbd: ❹
    monitors: ❺
      - 192.168.122.133:6789
    pool: rbd
    image: ceph-image
    user: admin
    secretRef:
      name: ceph-secret ❻
    fsType: ext4 ❼
    readOnly: false
  persistentVolumeReclaimPolicy: Retain

```

- ❶ The name of the PV that is referenced in pod definitions or displayed in various `oc volume` commands.
- ❷ The amount of storage allocated to this volume.
- ❸ `accessModes` are used as labels to match a PV and a PVC. They currently do not define any form of

kubernetes.io/no-provisioner

Local

FEATURE STATE: `Kubernetes v1.14` [stable]

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer

```

Local volumes do not currently support dynamic provisioning, however a StorageClass should still be created to delay volume binding until Pod scheduling. This is specified by the `WaitForFirstConsumer` volume binding mode.

Delaying volume binding allows the scheduler to consider all of a Pod's scheduling constraints when choosing an appropriate PersistentVolume for a PersistentVolumeClaim.

Q-6. Na podstawie powyższego polecenia umieść w raporcie sekcję „Events” pokazującą przebieg procesu tworzenia przestrzeni dyskowej.

```

Events:
  Type      Reason      Message      Age      From
  ----      -
Normal ExternalProvisioning 78s (x2 over 78s) persistentvolume-controller
Waiting for a volume to be created, either by external provisioner "k8s-sigs.io/nfs-subdir-external-provisioner" or manually created by
em administrator
Normal Provisioning 78s k8s-sigs.io/nfs-subdir-external-provisioner_nfs-client-provisioner-69b9f5cbf4-c5jqc_81b4cab4-6b14-4f6
ba-7c165d2b11a8 External provisioner is provisioning volume for claim "labproj04/pvc-bb"
Normal ProvisioningSucceeded 78s k8s-sigs.io/nfs-subdir-external-provisioner_nfs-client-provisioner-69b9f5cbf4-c5jqc_81b4cab4-6b14-4f6
ba-7c165d2b11a8 Successfully provisioned volume pvc-10134071-5bce-44bc-9c4d-1c76aee382ac
student@env4-WKSnode-lab:~/lab10$

```

Q-7. Za pomocą polecenia „kubectl describe” na utworzonym automatycznie PV ustal adres serwera NFS realizującego składowanie danych oraz ścieżkę utworzonego katalogu.

```

student@env4-WKSnode-lab:~/lab10$ kubectl describe pv pvc-10134071-5bce-44bc-9c4d-1c76aee382ac
Name:          pvc-10134071-5bce-44bc-9c4d-1c76aee382ac
Labels:        <none>
Annotations:   pv.kubernetes.io/provisioned-by: k8s-sigs.io/nfs-subdir-external-provisioner
Finalizers:    [kubernetes.io/pv-protection]
StorageClass:  managed-nfs-storage
Status:        Bound
Claim:         labproj04/pvc-bb
Reclaim Policy: Delete
Access Modes:  RWX
VolumeMode:    Filesystem
Capacity:      1Gi
Node Affinity: <none>
Message:
Source:
  Type:        NFS (an NFS mount that lasts the lifetime of a pod)
  Server:      192.168.241.30
  Path:        /nfs/labproj04-pvc-bb-pvc-10134071-5bce-44bc-9c4d-1c76aee382ac
  ReadOnly:    false
Events:        <none>
student@env4-WKSnode-lab:~/lab10$

```

Adres: 192.168.241.30

Path: /nfs/labproj04-pvc-bb-pvc-10134071-5bce-44bc-9c4d-1c67aee382ac

Q-8. . W raporcie umieść wyniki sprawdzania dostępności pliku i katalogu w podzie PIERWSZYM, DRUGIM i TRZECIM.

```

student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-clustervolume-6f8578f6f8-bt95p -- sh
/ # ls -ld /dane/
drwxrwxrwx 2 nobody nobody 4096 Dec 20 16:37 /dane/
/ # mkdir /dane
mkdir: can't create directory '/dane': File exists
/ # echo "dane testowe" > /dane/plik.txt
/ # exit
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-clustervolume-6f8578f6f8-bt95p -- cat /dane/plik.txt
dane testowe
student@env4-WKSnode-lab:~/lab10$ kubectl scale --replicas=2 deployment bb-clustervolume
deployment.apps/bb-clustervolume scaled
student@env4-WKSnode-lab:~/lab10$ kubectl get pod
NAME READY STATUS RESTARTS AGE
bb-clustervolume-6f8578f6f8-bt95p 1/1 Running 0 3m17s
bb-clustervolume-6f8578f6f8-r4f97 1/1 Running 0 7s
bb-emptydir-78cb4d95ff-8zcyj 1/1 Running 0 27m
bb-emptydir-78cb4d95ff-zjj4q 1/1 Running 0 21m
bb-nostorage-57c5b799db-cntc4 1/1 Running 0 61m
bb-nostorage-57c5b799db-g2g57 1/1 Running 0 62m
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-clustervolume-6f8578f6f8-r4f97 -- ls -ld /dane/
drwxrwxrwx 2 nobody nobody 4096 Dec 20 16:46 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-clustervolume-6f8578f6f8-r4f97 -- cat /dane/plik.txt
dane testowe
student@env4-WKSnode-lab:~/lab10$ kubectl delete pod bb-clustervolume-6f8578f6f8-bt95p
pod "bb-clustervolume-6f8578f6f8-bt95p" deleted

student@env4-WKSnode-lab:~/lab10$
student@env4-WKSnode-lab:~/lab10$ kubectl get pod
NAME READY STATUS RESTARTS AGE
bb-clustervolume-6f8578f6f8-bzqvj 1/1 Running 0 35s
bb-clustervolume-6f8578f6f8-r4f97 1/1 Running 0 2m
bb-emptydir-78cb4d95ff-8zcyj 1/1 Running 0 29m
bb-emptydir-78cb4d95ff-zjj4q 1/1 Running 0 23m
bb-nostorage-57c5b799db-cntc4 1/1 Running 0 63m
bb-nostorage-57c5b799db-g2g57 1/1 Running 0 64m
student@env4-WKSnode-lab:~/lab10$ ^C
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-clustervolume-6f8578f6f8-bzqvj -- ls -ld /dane/
drwxrwxrwx 2 nobody nobody 4096 Dec 20 16:46 /dane/
student@env4-WKSnode-lab:~/lab10$ kubectl exec -it bb-clustervolume-6f8578f6f8-bzqvj -- cat /dane/plik.txt
dane testowe
student@env4-WKSnode-lab:~/lab10$

```

- opis zaobserwowanej sytuacji – Wszystkie pody miały dostęp do katalogu i jak i pliku.

-wnioski - Użyliśmy PV typu NFS, który nie znika po usunięciu poda a jedynie jest odłączany od niego. Każdy pod który powstał na podstawie deploymentu z PV będzie miał dostęp do katalogu i pliku. Jeśli w podzie do PV dopiszemy dane a następnie usuniemy tego poda, to każdy pod kolejny pod który powstanie na tym deploymentu będzie miał dostęp do zmienionych danych.

Q-9. W raporcie podaj katalog który został automatycznie utworzony na serwerze nfs aby udostępnić przestrzeń dla aplikacji.

Path: /nfs/labproj04-pvc-nginx-pvc-10134071-5bce-44bc-9c4d-1c67aee382ac

Q-10. W raporcie zamieść komendy którymi utworzyłeś testowego poda, wykonaną komendę testującą i efekt jej działania

```

NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE                                NOMINATED NODE   READINESS GATES
bb-cluster-volume-6f9578f6f8-bzqvj  1/1    Running   0           44m   10.254.8.202    worker-5.ocp.lab.cloudpak.site     <none>            <none>
bb-cluster-volume-6f9578f6f8-r4f97  1/1    Running   0           46m   10.254.6.120    worker-1.ocp.lab.cloudpak.site     <none>            <none>
bb-emptydir-78cb4d95ff-8scj1        1/1    Running   0           73m   10.254.6.101    worker-1.ocp.lab.cloudpak.site     <none>            <none>
bb-emptydir-78cb4d95ff-z1j4q        1/1    Running   0           67m   10.254.8.182    worker-5.ocp.lab.cloudpak.site     <none>            <none>
bb-nostorage-57c8b799db-cncc4       1/1    Running   0           107m   10.254.7.126    worker-4.ocp.lab.cloudpak.site     <none>            <none>
bb-nostorage-57c8b799db-g2q57       1/1    Running   0           108m   10.254.7.123    worker-4.ocp.lab.cloudpak.site     <none>            <none>
bb-ww-7758f6495-9wa1c              1/1    Running   0           55s   10.254.7.197    worker-4.ocp.lab.cloudpak.site     <none>            <none>
student@env4-WKNode-lab:~/lab10$ kubectl exec bb -- wget -O - 10.254.7.197:8081
Error from server (NotFound): pods "bb" not found
student@env4-WKNode-lab:~/lab10$ kubectl run bb --restart=Never --image=default-route-openshift-image-registry.apps.ocp.lab.cloudpak.site/default/busybox:latest --command sleep 7200
pod/bb created
student@env4-WKNode-lab:~/lab10$ kubectl exec bb -- wget -O - 10.254.7.197:8081
Connecting to 10.254.7.197:8081 (10.254.7.197:8081)
!DOCTYPE html>
<html>
<body>

<h1>Kacper Guzik AMW</h1>

</body>
</html>
Writing to stdout
100% |*****| 74 0:00:00 ETA
Written to stdout
student@env4-WKNode-lab:~/lab10$

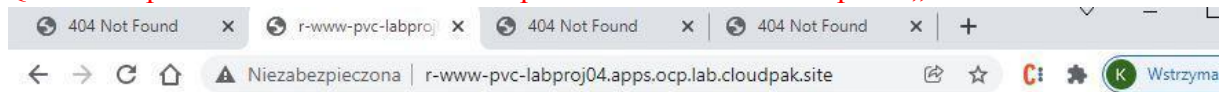
```

- Kubectl get pod -o wide
- Kubectl run bb ...
- Kubectl exec bb -- wget -o - 10.254.67.197:8081

Q-13. W raporcie typ filesystemu (FSType) użytego w naszym automatycznie utworzonym obiekcie PV

Odp: cluster-volume

Q-11. W raporcie zamieść screenshot odpowiedzi serwera na endpoint „/info”



Kacper Guzik AMW

Q-14. W raporcie zamieść komendy którymi utworzyłeś testowego poda, wykonaną komendę testującą i efekt jej działania

```

NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE                                NOMINATED NODE   READINESS GATES
pvc-nano-ceph-75587886b8-f8trv8     1/1    Running   0           4m9s   10.254.8.60     worker-5.ocp.lab.cloudpak.site     <none>            <none>
student@env4-WKNode-lab:~/lab10$ kubectl exec bb -- wget -O - http://10.254.8.60
Connecting to 10.254.8.60 (10.254.8.60:80)
Writing to stdout
!DOCTYPE html>
<html>
<body>

<h1>Kacper Guzik AMW</h1>

</body>
</html>
100% |*****| 74 0:00:00 ETA
Written to stdout
student@env4-WKNode-lab:~/lab10$
student@env4-WKNode-lab:~/lab10$ <body>
bash: syntax error near unexpected token `newline'
student@env4-WKNode-lab:~/lab10$ cp pvc-ceph.yaml d-nano-ceph.yaml /home/student/lab10/results/
student@env4-WKNode-lab:~/lab10$ kubectl get all > /home/student/lab10/results/ver-2.txt
student@env4-WKNode-lab:~/lab10$ kubectl get deploy
No resources found.
student@env4-WKNode-lab:~/lab10$ kubectl get pvc
NAME      STATUS   VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS          AGE
pvc-ceph  Bound   pvc-ee9e4975-4918-4cbd-8265-549b785512a4  50Mi        RWO            ocs-storagecluster-ceph-rbd  19m
student@env4-WKNode-lab:~/lab10$ kubectl delete pvc pvc-ceph
persistentvolumeclaim "pvc-ceph" deleted

```

- Kubectl get pod -o wide
- Kubectl exec -- wget -o - http://10.254.8.60

Q-15. Jakie zaobserwowałeś różnice w działaniu provisioningu opartego o nfs i ceph?

Odp: Niestety nie znalazłem żadnych różnic