Silesian University of Technology
Informatics

Politechnika Śląska

Microservice orchestration platforms using Kubernetes

“Implementation of the WordPress system based on the K8S cluster services”

Authors:
Konrad Dębiński
Edyta Hernik
Paweł Sapek
Łukasz Zientek

1. Introduction

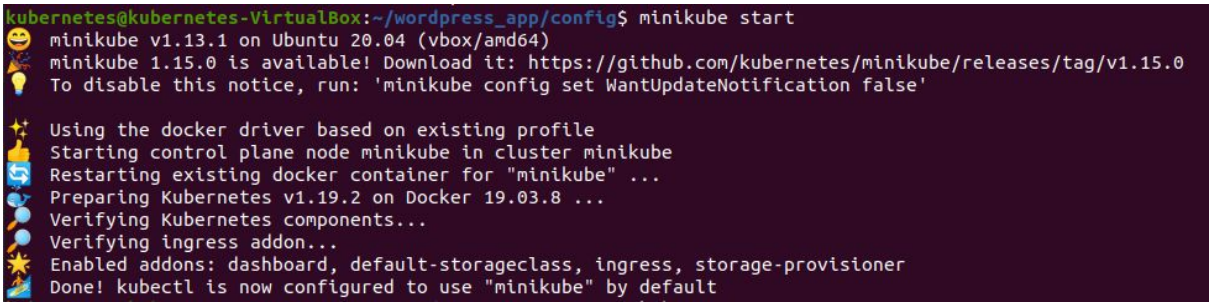
The aim of the project was to implement a WordPress system using Kubernetes cluster services. The task was to design and deploy the system and to ensure its reliability, security, and efficiency. The system should consist of software and images necessary to incorporate basic features as system services, configuration contained in a namespace, database, and cluster monitoring.

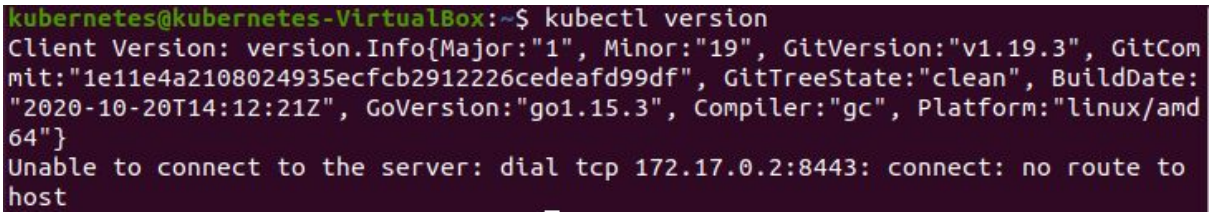
2. Implementation

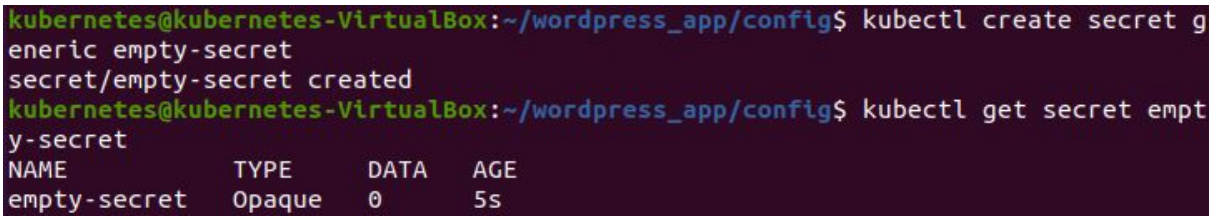
Mandatory features

System services are available at the dedicated DNS name or at least exposed locally (e.g. hosts file)

In this subsection, we will show how the deployment of our application took place. We exposed the services locally.







```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ cat <<EOF >./kustomizat
ion.yaml
> secretGenerator:
> - name: mysql-pass
>   literals:
>   - password=YOUR_PASSWORD
> EOF
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ ls -l
total 4
-rw-rw-r-- 1 kubernetes kubernetes 75 lis 15 21:51 kustomization.yaml
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ sudo vim kustomization.
yaml
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ curl -LO https://k8s.io
/examples/application/wordpress/mysql-deployment.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   185  100   185    0     0    292      0 --:--:-- --:--:-- --:--:--   292
100  1238  100  1238    0     0   1046      0  0:00:01  0:00:01 --:--:--  7074
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ curl -LO https://k8s.io
/examples/application/wordpress/wordpress-deployment.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   185  100   185    0     0    885      0 --:--:-- --:--:-- --:--:--   880
100  1323  100  1323    0     0   3780      0 --:--:-- --:--:-- --:--:--  3780
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ cat <<EOF >>./kustomiza
tion.yaml
> resources:
>   - mysql-deployment.yaml
>   - wordpress-deployment.yaml
> EOF
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl apply -k ./
secret/mysql-pass-tk25d899m9 created
service/wordpress-mysql created
service/wordpress created
deployment.apps/wordpress-mysql created
deployment.apps/wordpress created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
```

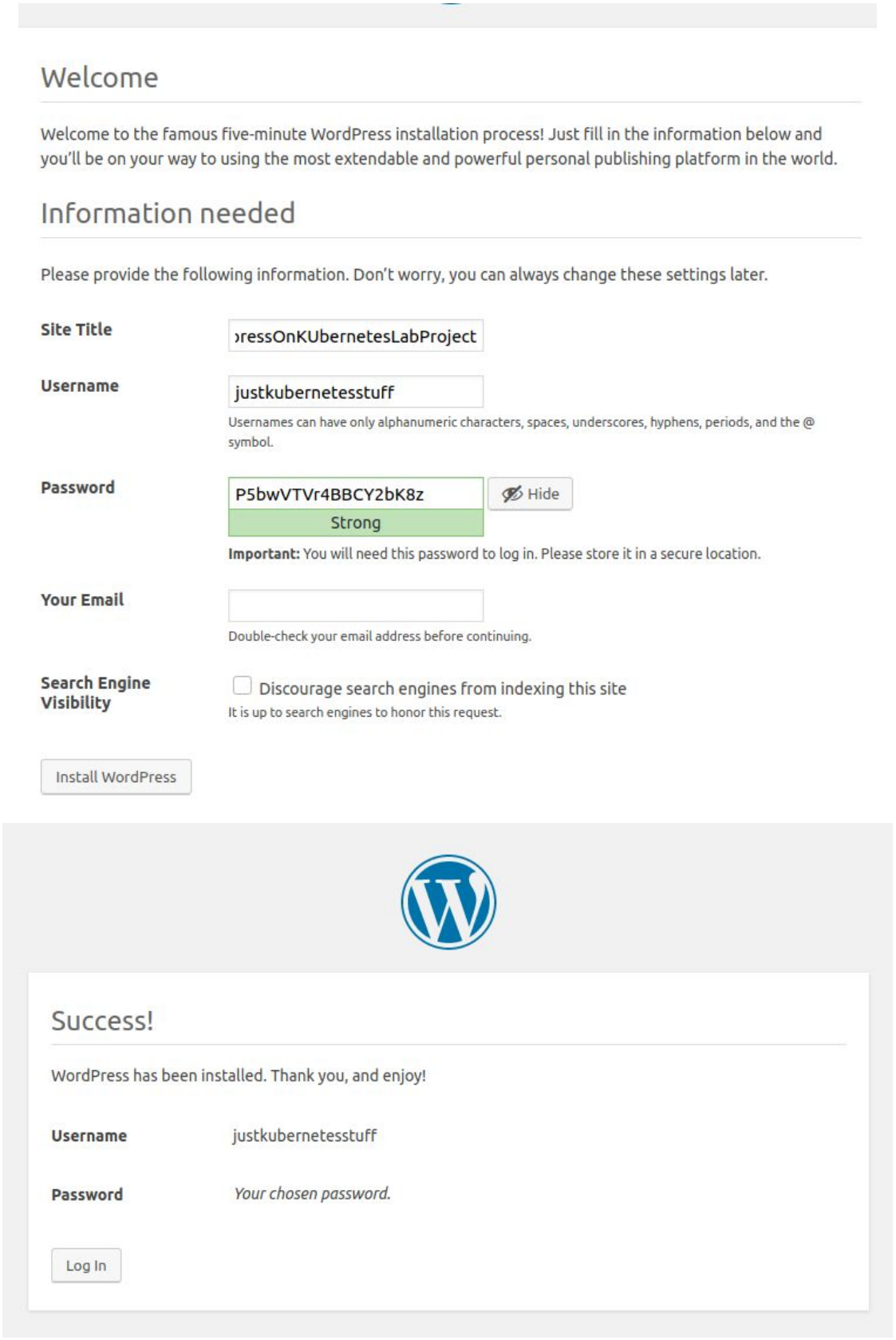
```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get secrets
NAME                  TYPE                                  DATA   AGE
default-token-vmphm   kubernetes.io/service-account-token   3      34d
empty-secret          Opaque                                0      7m32s
mysql-pass-tk25d899m9 Opaque                                1      2m30s
```

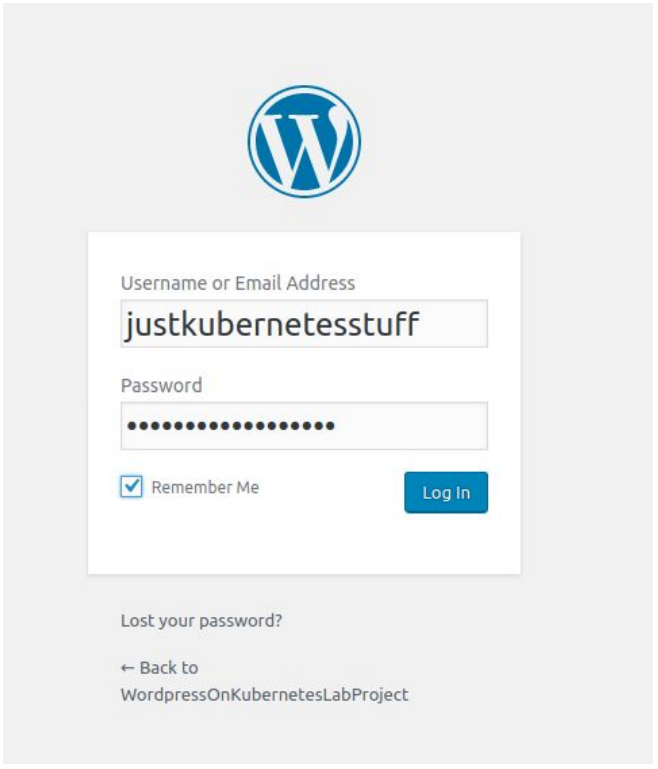
```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get services wordpress
NAME        TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
wordpress   LoadBalancer   10.110.216.162   <pending>     80:31638/TCP   5m4s
```

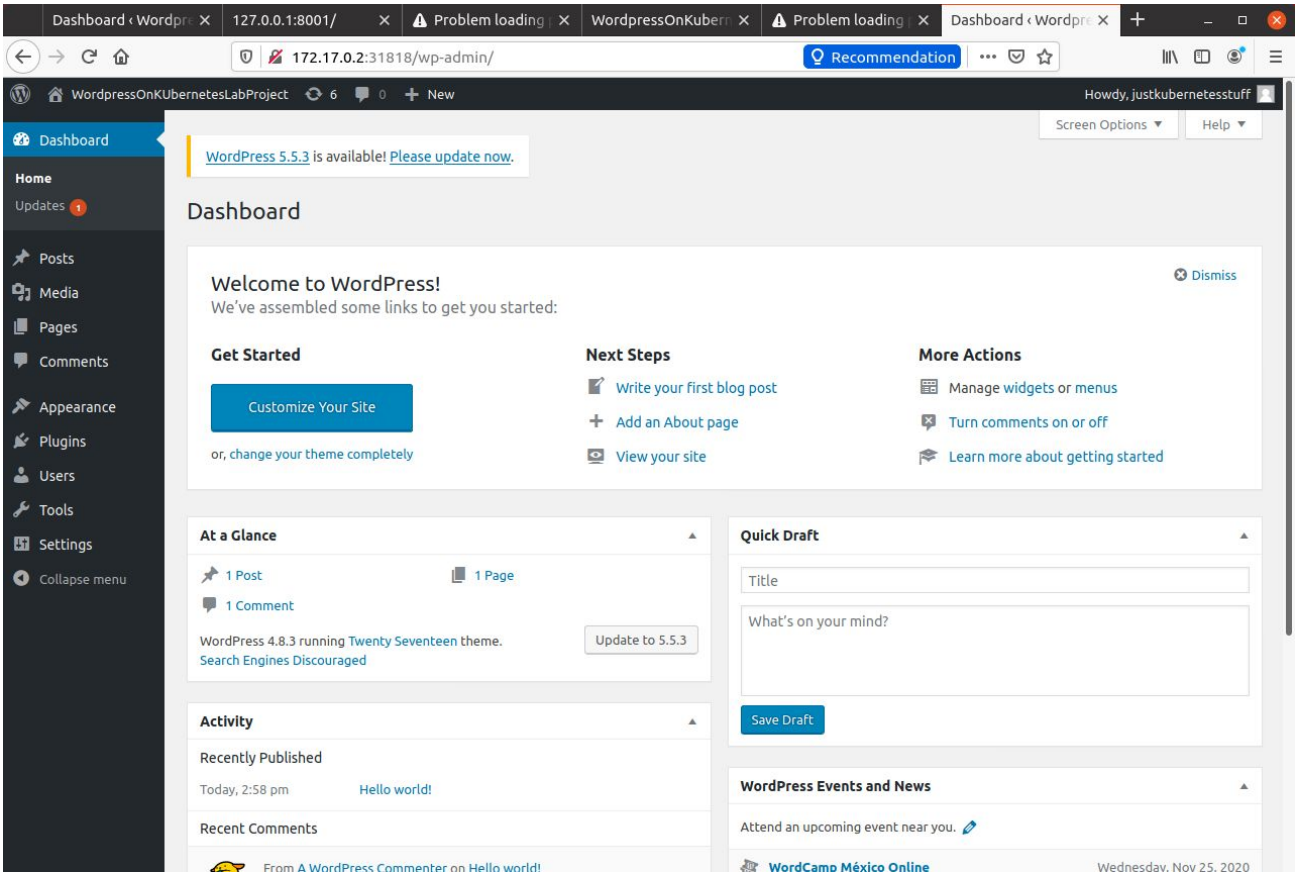
```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
web-79d88c97d6-52nk9               1/1     Running   1          20d
web2-5d47994f45-nxkrs              1/1     Running   1          20d
wordpress-db7f76655-ctbxf          1/1     Running   1          3m51s
wordpress-mysql-67867df58c-hpfjl   1/1     Running   0          3m51s
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ minikube service wordpress --url
http://172.17.0.2:31638
```

Working application:




Username or Email Address
justkubernetesstuff
Password
Remember Me
Log In
Lost your password?
← Back to
WordpressOnKubernetesLabProject


Dashboard ‹ Wordpr
127.0.0.1:8001/
Problem loading
WordpressOnKuberr
Problem loading
Dashboard ‹ Wordpr
172.17.0.2:31818/wp-admin/
Recommendation
WordpressOnKUbernetesLabProject
6
0
New
Howdy, justkubernetesstuff
Screen Options
Help
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
WordPress 5.5.3 is available! Please update now.
Dashboard
Welcome to WordPress!
We've assembled some links to get you started:
Dismiss
Get Started
Customize Your Site
or, change your theme completely
Next Steps
Write your first blog post
Add an About page
View your site
More Actions
Manage widgets or menus
Turn comments on or off
Learn more about getting started
At a Glance
1 Post
1 Page
1 Comment
WordPress 4.8.3 running Twenty Seventeen theme.
Search Engines Discouraged
Update to 5.5.3
Quick Draft
Title
What's on your mind?
Save Draft
Activity
Recently Published
Today, 2:58 pm
Hello world!
Recent Comments
From A WordPress Commenter on Hello world!
WordPress Events and News
Attend an upcoming event near you.
WordCamp México Online
Wednesday, Nov 25, 2020

All configuration is contained in a dedicated namespace

To ensure that all configuration is contained in a dedicated namespace, first, we created our custom namespace *anothertry.* In the next step, we create the kubeconfig files in the created namespace.

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl create namespace anothertry
namespace/anothertry created
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get namespaces
NAME                   STATUS   AGE
anothertry             Active   14s
default                Active   34d
kube-node-lease        Active   34d
kube-public            Active   34d
kube-system            Active   34d
kubernetes-dashboard   Active   34d
private                Active   58m
testnamespace          Active   22m
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get namespaces --show-labels
NAME                   STATUS   AGE    LABELS
anothertry             Active   2m5s   <none>
default                Active   34d    <none>
kube-node-lease        Active   34d    <none>
kube-public            Active   34d    <none>
kube-system            Active   34d    <none>
kubernetes-dashboard   Active   34d    <none>
private                Active   60m    <none>
testnamespace          Active   24m    <none>
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/kubernetes/.minikube/ca.crt
    server: https://172.17.0.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /home/kubernetes/.minikube/profiles/minikube/client.crt
    client-key: /home/kubernetes/.minikube/profiles/minikube/client.key
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config use-context anothertry
error: no context exists with the name: "anothertry"
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config current-context
minikube
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config set-context another --namespace=anothertry --cluster=minikube --user=minikube
Context "another" created.
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/kubernetes/.minikube/ca.crt
    server: https://172.17.0.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    namespace: anothertry
    user: minikube
  name: another
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/kubernetes/.minikube/ca.crt
    server: https://172.17.0.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    namespace: anothertry
    user: minikube
  name: another
- context:
    cluster: minikube
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /home/kubernetes/.minikube/profiles/minikube/client.crt
    client-key: /home/kubernetes/.minikube/profiles/minikube/client.key
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config use-context another
Switched to context "another".
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl config current-context
another
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl apply -k ./
namespace/testnamespace unchanged
secret/mysql-pass-tk25d899m9 created
service/wordpress-mysql created
service/wordpress created
deployment.apps/wordpress-mysql created
deployment.apps/wordpress created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get deployment
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
wordpress         1/1     1            1           12s
wordpress-mysql   1/1     1            1           12s
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get pods -l app=another
No resources found in anothertry namespace.
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get pods -l app=wordpress
NAME                              READY   STATUS    RESTARTS   AGE
wordpress-db7f76655-lwgtc         1/1     Running   1          41s
wordpress-mysql-67867df58c-dt88z  1/1     Running   0          41s
```

Database service is not available from outside the cluster/Data persistence is ensured (i.e. data is stored independently of the system services container(s))

To ensure data persistence we used *PersistentVolumeClaim* and *PersistentVolum*e which are separate storages that work independently to any individual Pods.

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get pvc
NAME             STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
mysql-pv-claim   Bound    pvc-6b809ebf-e242-45fc-ba6a-f44a6f6ee905   20Gi
RWO            standard       3m11s
wp-pv-claim      Bound    pvc-2e9375fc-cbb6-4d1b-96a6-933e878e1fe8   20Gi
RWO            standard       3m11s
```

System services instances are multiplied to achieve basic availability

The multiplication of instances is done by the following command:

kubectl scale deployments/wordpress --replicas=4

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl scale deployments/wordpress --replicas=4
deployment.apps/wordpress scaled
```

We created four replicas of our application. WIth *kubectl get deployments* we can check if the scaling operation was successful. As we can see from the results presented on the screen below, the number of wordpress replicas is four.

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get deployments
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
wordpress         4/4     4            4           21m
wordpress-mysql   1/1     1            1           21m
```

Also, we can observe from the output of *kuectl describe services/wordpress* and *kuectl describe deployments/wordpress* command, that we have four different IP addresses of our pods and also the log registry of creation the replicas.

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl describe services/wordpress
Name:                     wordpress
Namespace:                anothertry
Labels:                   app=wordpress
Annotations:              <none>
Selector:                 app=wordpress,tier=frontend
Type:                     LoadBalancer
IP:                       10.101.132.134
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31818/TCP
Endpoints:                172.18.0.15:80,172.18.0.16:80,172.18.0.17:80 + 1 more...
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl describe deployments/wordpress
Name:               wordpress
Namespace:          anothertry
CreationTimestamp:  Mon, 16 Nov 2020 16:32:40 +0100
Labels:             app=wordpress
Annotations:        deployment.kubernetes.io/revision: 1
Selector:           app=wordpress,tier=frontend
Replicas:           4 desired | 4 updated | 4 total | 4 available | 0 unavailable
StrategyType:       Recreate
MinReadySeconds:    0
Pod Template:
  Labels:  app=wordpress
           tier=frontend
  Containers:
   wordpress:
    Image:      wordpress:4.8-apache
    Port:       80/TCP
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ export NODE_PORT=$(kubectl get services/wordpress -o go-template='{{(index .spec.ports 0).nodePort}}')
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ curl $(minikube ip):$NODE_PORT
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ echo NODE_PORT=$NODE_PORT
NODE_PORT=31818
```

Basic cluster monitoring is deployed (e.g. Kubernetes Dashboard)

Kubernetes offers a convenient graphical UI with a web dashboard. It can be used to monitor and manage clusters.

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yam
l
namespace/kubernetes-dashboard unchanged
serviceaccount/kubernetes-dashboard unchanged
service/kubernetes-dashboard unchanged
secret/kubernetes-dashboard-certs unchanged
secret/kubernetes-dashboard-csrf unchanged
secret/kubernetes-dashboard-key-holder unchanged
configmap/kubernetes-dashboard-settings unchanged
role.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged
deployment.apps/kubernetes-dashboard unchanged
service/dashboard-metrics-scraper unchanged
deployment.apps/dashboard-metrics-scraper unchanged
The ClusterRoleBinding "kubernetes-dashboard" is invalid: roleRef: Invalid value: rbac.RoleRef{APIGroup:"rbac.authorization.k8s.io", Kind:"ClusterRole", Name:"kub
ernetes-dashboard"}: cannot change roleRef
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ cat <<EOF | kubectl apply -f -
> apiVersion: v1
> kind: ServiceAccount
> metadata:
>   name: admin-user
>   namespace: kubernetes-dashboard
> EOF
serviceaccount/admin-user unchanged
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep admin
-user | awk '{print $1}')
Name:         admin-user-token-hfjzs
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: 2b8c1392-d23d-4415-81f0-9e02fbd7a209

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1066 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6Ii1UaENOUnhsTXExUWdXbG92bHM1SG5Pb3ZOcV9YNzhyQ3ZPM1daZ212MjgifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXR
```