

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$ minikube start
🐳 minikube v1.13.1 on Ubuntu 20.04 (vbox/amd64)
💡 minikube 1.15.0 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.15.0
To disable this notice, run: 'minikube config set WantUpdateNotification false'

🌟 Using the docker driver based on existing profile
👉 Starting control plane node minikube in cluster minikube
🔄 Restarting existing docker container for "minikube" ...
🔧 Preparing Kubernetes v1.19.2 on Docker 19.03.8 ...
🔍 Verifying Kubernetes components...
🔍 Verifying ingress addon...
🌟 Enabled addons: dashboard, default-storageclass, ingress, storage-provisioner
🏁 Done! kubectl is now configured to use "minikube" by default
```

```
kubernetes@kubernetes-VirtualBox:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.3", GitCommit:"1e11e4a2108024935ecfcb2912226cedaafd99df", GitTreeState:"clean", BuildDate:"2020-10-20T14:12:21Z", GoVersion:"go1.15.3", Compiler:"gc", Platform:"linux/amd64"}
Unable to connect to the server: dial tcp 172.17.0.2:8443: connect: no route to host
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl create secret generic empty-secret
secret/empty-secret created
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ kubectl get secret empty-secret
NAME          TYPE          DATA   AGE
empty-secret  Opaque        0       5s
```

```
kubernetes@kubernetes-VirtualBox:~/wordpress_app/config$ cat <<EOF >./kustomization.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 >>./kustomization.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:

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

 Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search Engine
Visibility

☐

Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress



Success!

WordPress has been installed. Thank you, and enjoy!


Username

justkubernetesstuff

Password

Your chosen password.

Log In



Username or Email Address

justkubernetesstuff

Password

••••••••••

☒ Remember Me

Log In

Lost your password?

[← Back to WordpressOnKubernetesLabProject](#)

Dashboard < WordpressOnKubernetesLabProject

127.0.0.1:8001/

Problem loading

WordpressOnKubern

Problem loading

Dashboard < Wordpress

172.17.0.2:31818/wp-admin/

Recommendation

Howdy, justkubernetesstuff

Screen Options

Help

Dashboard

Home

Updates 1

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:

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

Activity

Recently Published

Today, 2:58 pm [Hello world!](#)

Recent Comments

 From A WordPress Commenter on Hello world!

Quick Draft

Title

What's on your mind?

Save Draft

WordPress Events and News

Attend an upcoming event near you. [WordCamp México Online](#)

Wednesdav, 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 *PersistentVolume* 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
wp-pv-claim         Bound        pvc-2e9375fc-cbb6-4d1b-96a6-933e878e1fe8  20Gi
RWO                 standard
```


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 *kubectl describe services/wordpress* and *kubectl 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.yaml
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:"kubernetes-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-4415-81f0-9e02fdb7a209

Type: kubernetes.io/service-account-token

Data
====
ca.crt:      1066 bytes
namespace:   20 bytes
token:       eyJhbGciOiJIUzI1NiIsImtpZCI6Ii1UaEN0Unh5TXExUWdXbG92bHM1SG5Pb3Z0cVY2VhY2NvdW50Iiwia3ViZXJ1ZXR
```

The screenshot shows the Kubernetes Dashboard web interface. The top navigation bar includes the Kubernetes logo and a search bar. The left sidebar contains a menu with sections: Cluster (Cluster Roles, Namespaces, Nodes, Persistent Volumes, Storage Classes), Namespace (anothertry), Overview (selected), Workloads (Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets), and Discovery and Load Balancing (Ingresses, Services). The main content area is titled 'Overview' and features a 'Workload Status' section with three large green circles representing Deployments, Pods, and Replica Sets. Below this, there is a 'Deployments' table with columns: Name, Labels, Pods, Created, and Images. The table lists two deployments: 'wordpress' (4/4 pods, created 6 days ago, using wordpress:4.8-apache) and 'wordpress-mysql' (1/1 pods, created 6 days ago, using mysql:5.6). At the bottom, there is a 'Pods' table with columns: Name, Labels, Node, Status, Restarts, CPU Usage (cores), Memory Usage (bytes), and Created.

Name	Labels	Pods	Created	Images
wordpress	app: wordpress	4 / 4	6 days ago	wordpress:4.8-apache
wordpress-mysql	app: wordpress	1 / 1	6 days ago	mysql:5.6

Name	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
wordpress							