

Sprawozdanie z "deploymentu Webaplikacji w Springu w środowisku k8s"

1. Konfiguracja i instalacja **Dockera** i **Kubernetesa** (wraz z *minikube* oraz *kubectl*).
2. *Minikube* to lokalny **Kubernetes**, skupiający się na ułatwianiu nauki i rozwijania **Kubernetesa**. Uruchomiłam klaster komendą:

```
PS C:\Windows\system32> minikube start
* minikube v1.28.0 na Microsoft Windows 10 Pro 10.0.19044 Build 19044
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
  > gcr.io/k8s-minikube/kicbase: 0 B [_____] ?% ? p/s 43s
* docker "minikube" container is missing, will recreate.
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Przygotowywanie Kubernetesa v1.25.3 na Docker 20.10.20...
  - Generating certificates and keys ...
  - Uruchamianie płaszczyzny kontrolnej ...
  - Konfigurowanie zasad RBAC ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

3. Aby upewnić się, że zrobiłam to prawidłowo wpisałam "*minikube status*".
4. Bardzo ważne, aby konkretnie w konsoli PowerShell wpisać komendę odpowiadającą "*eval \$(minikube docker-env)*" czyli "*minikube docker-env | Invoke-Expression*". Jest ona kluczowa, gdyż łączy Dockera z Docker Daemon wewnątrz maszyny wirtualnej, przez co później utworzone "podsy" będą działały poprawnie, gdyż będą miały dostęp do obrazu.

```
Run 'minikube --help' for usage.
PS C:\Windows\system32> minikube status
minikube
type: Control Plane
host: Running
kubeleet: Running
apiserver: Running
kubeconfig: Configured

PS C:\Windows\system32> minikube docker-env | Invoke-Expression
```

5. Przeszłam do katalogu, w którym mam swoją webaplikację, która wykorzystuje technologię Spring:

```
PS C:\Windows\system32> cd "D:\SGGW\Semestr VII\fakultet-k8s"
```

6. Tworzę Dockerfile:

```
k8s-deploy.yaml × service.yaml × Dockerfile × SpringbootK8sDemoApplication.java ×  
FROM openjdk:8  
EXPOSE 8080  
ADD target/springboot-k8s-demo.jar springboot-k8s-demo.jar  
ENTRYPOINT ["java","-jar","/springboot-k8s-demo.jar"]
```

7. Tworzę plik konfiguracyjny swój deployment:

```
k8s-deploy.yaml × service.yaml × Dockerfile × SpringbootK8sDemoApplication.java ×  
1  apiVersion: apps/v1  
2  kind: Deployment  
3  metadata:  
4    name: spring-boot-k8s  
5  spec:  
6    selector:  
7      matchLabels:  
8        app: spring-boot-k8s  
9    replicas: 3  
10   template:  
11     metadata:  
12       labels:  
13         app: spring-boot-k8s  
14     spec:  
15       containers:  
16         - name: spring-boot-k8s  
17           image: fakultet-k8s:1.0  
18           imagePullPolicy: IfNotPresent  
19           ports:  
20             - containerPort: 8080
```

8. Sprawdzam czy pliki są widoczne dla PowerShella:

```
PS D:\SGGW\Semestr VII\fakultet-k8s> ls  
  
Directory: D:\SGGW\Semestr VII\fakultet-k8s  
  
Mode                LastWriteTime         Length Name  
----                -  
d-----          13.12.2022    00:30           .idea  
d-----          12.12.2022    21:28           src  
d-----          12.12.2022    23:36          target  
-a-----          12.12.2022    21:28          142 Dockerfile  
-a-----          13.12.2022    00:35          430 k8s-deploy.yaml  
-a-----          12.12.2022    21:28         1347 pom.xml  
-a-----          12.12.2022    21:28          21 README.md  
-a-----          12.12.2022    21:28          437 service.yaml
```

9. Tworzę obraz z konfiguracji Dockerfile:

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> docker build -t fakultet-k8s:1.0 .
Sending build context to Docker daemon 17.76MB
Step 1/4 : FROM openjdk:8
8: Pulling from library/openjdk
001c52e26ad5: Pull complete
d9d4b9b6e964: Pull complete
2068746827ec: Pull complete
9daef329d350: Pull complete
d85151f15b66: Pull complete
52a8c426d30b: Pull complete
8754a66e0050: Pull complete
Digest: sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8
Status: Downloaded newer image for openjdk:8
--> b273004037cc
Step 2/4 : EXPOSE 8080
--> Running in 76affe24ebb9
Removing intermediate container 76affe24ebb9
--> 62ffffe0851b5
Step 3/4 : ADD target/springboot-k8s-demo.jar springboot-k8s-demo.jar
--> 7cbc7cf8e5d1
Step 4/4 : ENTRYPOINT ["java","-jar","/springboot-k8s-demo.jar"]
--> Running in a0bf4c2eb2f2
Removing intermediate container a0bf4c2eb2f2
--> 20a9b5c5a737
Successfully built 20a9b5c5a737
Successfully tagged fakultet-k8s:1.0
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All file
s and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double c
heck and reset permissions for sensitive files and directories.
```

10. Sprawdzam czy prawidłowo się utworzył:

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
fakultet-k8s	1.0	20a9b5c5a737	24 seconds ago	544MB
registry.k8s.io/kube-apiserver	v1.25.3	0346dbd74bcb	2 months ago	128MB
registry.k8s.io/kube-controller-manager	v1.25.3	603999231275	2 months ago	117MB
registry.k8s.io/kube-scheduler	v1.25.3	6d23ec0e8b87	2 months ago	50.6MB
registry.k8s.io/kube-proxy	v1.25.3	beaaf00edd38	2 months ago	61.7MB
openjdk	8	b273004037cc	4 months ago	526MB
registry.k8s.io/pause	3.8	4873874c08ef	6 months ago	711kB
registry.k8s.io/etcd	3.5.4-0	a8a176a5d5d6	6 months ago	300MB
registry.k8s.io/coredns/coredns	v1.9.3	5185b96f0bec	6 months ago	48.8MB
k8s.gcr.io/pause	3.6	6270bb605e12	15 months ago	683kB
gcr.io/k8s-minikube/storage-provisioner	v5	6e38f40d628d	20 months ago	31.5MB

11. Tworzę deployment za pomocą *kubectl* i sprawdzam czy “podsy” działają prawidłowo

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> ls
```

Directory: D:\SGGW\Semestr VII\ fakultet-k8s

Mode	LastWriteTime	Length	Name
d----	13.12.2022 00:45		.idea
d----	12.12.2022 21:28		src
d----	12.12.2022 23:36		target
-a----	12.12.2022 21:28	142	Dockerfile
-a----	13.12.2022 00:49	419	k8s-deploy.yaml
-a----	12.12.2022 21:28	1347	pom.xml
-a----	12.12.2022 21:28	21	README.md
-a----	12.12.2022 21:28	437	service.yaml

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl apply -f k8s-deploy.yaml
deployment.apps/spring-boot-k8s created
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
spring-boot-k8s	3/3	3	3	11s

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
spring-boot-k8s-68c749564d-5k6cs	1/1	Running	0	59s
spring-boot-k8s-68c749564d-gvm6l	1/1	Running	0	59s
spring-boot-k8s-68c749564d-h7787	1/1	Running	0	59s

12. Przykład prawidłowego działania jednego "poda"

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl logs spring-boot-k8s-68c749564d-5k6cs

:: Spring Boot ::
      (v2.6.1)

2022-12-12 23:51:06.929 INFO 1 --- [           main] c.j.k8s.SpringbootK8sDemoApplication : Starting
SpringbootK8sDemoApplication v0.0.1-SNAPSHOT using Java 1.8.0_342 on spring-boot-k8s-68c749564d-5k6cs with
PID 1 (/springboot-k8s-demo.jar started by root in /)
2022-12-12 23:51:06.932 INFO 1 --- [           main] c.j.k8s.SpringbootK8sDemoApplication : No active
profile set, falling back to default profiles: default
2022-12-12 23:51:12.717 INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat in
ialized with port(s): 8080 (http)
2022-12-12 23:51:12.810 INFO 1 --- [           main] o.apache.catalina.core.StandardService : Starting
service [Tomcat]
2022-12-12 23:51:12.811 INFO 1 --- [           main] org.apache.catalina.core.StandardEngine : Starting
Servlet engine: [Apache Tomcat/9.0.55]
2022-12-12 23:51:13.212 INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializ
ing Spring embedded WebApplicationContext
2022-12-12 23:51:13.212 INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebA
pplicationContext: initialization completed in 5985 ms
2022-12-12 23:51:24.633 INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat st
arted on port(s): 8080 (http) with context path ''
2022-12-12 23:51:24.714 INFO 1 --- [           main] c.j.k8s.SpringbootK8sDemoApplication : Started S
pringbootK8sDemoApplication in 22.48 seconds (JVM running for 24.595)
```

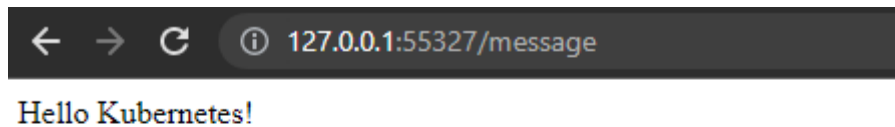
13. Tworzę konfigurację do serwisu Kubernetesa:

```
apiVersion: v1
kind: Service
metadata:
  name: spring-boot-k8s-service
spec:
  selector:
    app: spring-boot-k8s
  ports:
    - protocol: "TCP"
      port: 8080
      targetPort: 8080
  type: NodePort
```

14. Tworzę serwis według powyższej konfiguracji:

```
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl apply -f service.yaml
service/spring-boot-k8s-service created
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl get service
NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
spring-boot-k8s-service            NodePort    10.102.61.159 <none>         8080:31176/TCP   7s
PS D:\SGGW\Semestr VII\ fakultet-k8s> kubectl get nodes -o wide
NAME                                STATUS    ROLES    AGE   VERSION   INTERNAL-IP    EXTERNAL-IP    OS-IMAGE             KERN
minikube Ready        control-plane 17m   v1.25.3   192.168.49.2    <none>         Ubuntu 20.04.5 LTS   5.10
.16.3-microsoft-standard-WSL2      docker://20.10.20
PS D:\SGGW\Semestr VII\ fakultet-k8s> minikube ip
192.168.49.2
PS D:\SGGW\Semestr VII\ fakultet-k8s> minikube service spring-boot-k8s-service --url
http://127.0.0.1:55327
! Z powodu użycia sterownika dockera na systemie operacyjnym windows, terminal musi zostać uruchomiony.
```

15. Wyżej uzyskałam url do działającego serwisu, więc sprawdzam działanie:



Działanie jest prawidłowe, gdyż wyświetla się wiadomość, która wzięła się z webaplikacji dla danego endpoint'u:

