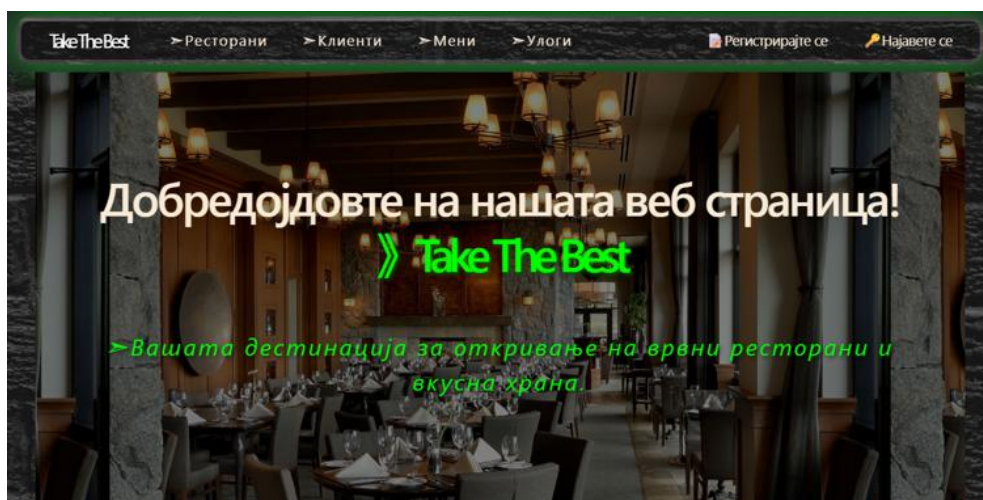




Универзитет „Св. Кирил и Методиј“ во Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**



Факултет: **Факултет за информатички науки и компјутерско инженерство**

Предмет: **Континуирана интеграција и испорака**

Проект: **Take The Best - Containerized Deployment Using Kubernetes and Docker**

Студент: **Иван Пупиноски 223260**

Наставници: **Проф. д-р Панче Рибарски, Асист. д-р Стефан Андонов**

Датум: **01.07.2025**

GitHub link: <https://github.com/iSqloud7/TakeTheBestDockerizedApp>

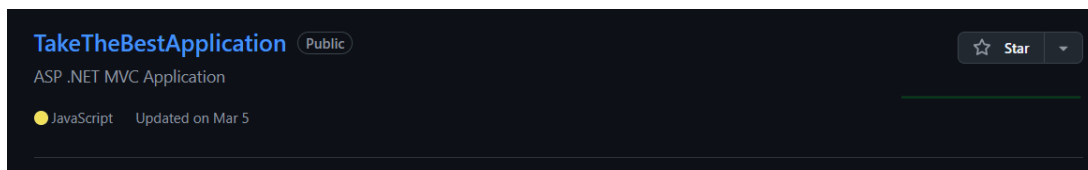
Краток опис

Проектот „TakeTheBest“ вклучува:

- ASP.NET MVC веб-апликација за преглед на ресторани и достапност на маси во реално време
- Докеризација на апликацијата и SQL Server база
- Оркестрација со Docker Compose
- Конфигурација на Kubernetes манифести (deployment, service, ingress, stateful set)
- CI/CD pipeline со GitHub Actions за автоматска изградба и пуштање на Docker image на DockerHub
- Локално тестирање преку Docker Desktop со вклучен Kubernetes кластер

Практични примери и докази за извршување

- Поставување на апликацијата на јавен git repositiorium.



- Докеризација на апликацијата.

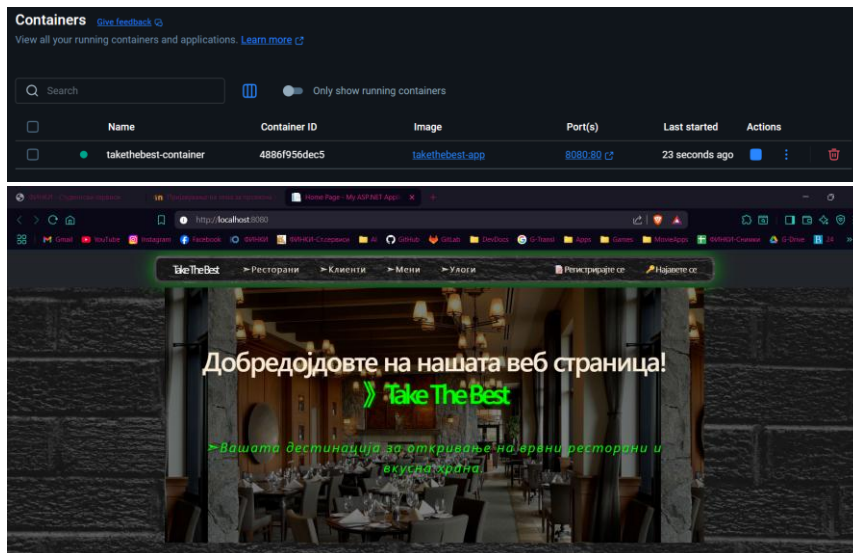
```
1 FROM mcr.microsoft.com/dotnet/framework/aspnet:4.8
2
3 WORKDIR /inetpub/wwwroot
4
5 COPY obj/Docker/publish/ .
6
7 EXPOSE 80
```

docker build -t takethebest-app .

```
PS D:\TakeTheBestApplication\TakeTheBest-Project\TakeTheBest-Project> docker build -t takethebest-app .
Sending build context to Docker daemon 162.4MB
Step 1/4 : FROM mcr.microsoft.com/dotnet/framework/aspnet:4.8
--> 48ad0b50f449
Step 2/4 : WORKDIR /inetpub/wwwroot
--> Using cache
--> a847105b90a8
Step 3/4 : COPY obj/Docker/publish/ .
--> d133c6355882
Step 4/4 : EXPOSE 80
--> Running in 87ce5ddaa218
--> Removed intermediate container 87ce5ddaa218
--> f694c82c300b
Successfully built f694c82c300b
Successfully tagged takethebest-app:latest
```

docker run -d -p 8080:80 --name takethebest-container takethebest-app

```
PS D:\TakeTheBestApplication\TakeTheBest-Project\TakeTheBest-Project> docker run -d -p 8080:80 --name takethebest-container takethebest-app
4886f956dec521d7634e498a31bf8755b41fd34f619c18daf5ffa8fe5831aa13
```



- Оркестрација на апликацијата и базата со Docker Compose.

```

1  version: '3.8'
2
3  services:
4    webapp:
5      image: take-the-best-app:latest
6      ports:
7        - "8080:80"

```

docker-compose up --build -d

PS D:\TakeTheBestApplication\TakeTheBest-Project\TakeTheBest-Project> docker-compose up --build -d
time="2025-06-30T21:27:57+02:00" level=warning msg="D:\TakeTheBestApplication\TakeTheBest-Project\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 1/1
✔ Container take-the-best-project-webapp-1 Started 9.9s

Containers Give feedback

View all your running containers and applications. [Learn more](#)

Name	Container ID	Image	Port(s)	Last started	Actions
take-the-best-container	77b1b9b656d6	take-the-best-app	8080:80	2 minutes ago	[Stop] [Refresh] [Delete]
take-the-best-project	-	-	-	41 seconds ago	[Stop] [Refresh] [Delete]
webapp-1	b8f3965606aa	take-the-best-app:latest	8081:80	41 seconds ago	[Stop] [Refresh] [Delete]

- Користење на GitHub Actions и сетирање pipeline за CI/CD решение.

git add .

git commit -m "Dockerized web app in ASPT .NET MVC!"

git push origin master

```
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project> git add .
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project> git commit -m "Dockerized web app in ASP .NET MVC!"
[master (root-commit) d5680dd] Dockerized web app in ASP .NET MVC!
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project> git push origin master
Enumerating objects: 969, done.
Counting objects: 100% (969/969), done.
Delta compression using up to 8 threads
Compressing objects: 100% (922/922), done.
Writing objects: 100% (969/969), 61.57 MiB | 8.26 MiB/s, done.
Total 969 (delta 277), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (277/277), done.
To https://github.com/iSqloud7/TakeTheBestDockerizedApp.git
 * [new branch]      master -> master
```

TakeTheBestDockerizedApp Public
☆ Star

A web app in ASP .NET MVC that is dockerized on the Continuous Integration and Delivery course.

JavaScript Updated now

Repository secrets			New repository secret
Name	Last updated		
DOCKERHUB_TOKEN	now		
DOCKERHUB_USERNAME	2 minutes ago		

Files
Collapse file tree
master
Go to file
.github/workflows
docker-publish.yml
TakeTheBest-Project

TakeTheBestDockerizedApp / .github / workflows / docker-publish.yml
View Runs
7b32a48 · now
History
Code
38 lines (29 loc) · 1.06 KB
1
name: Build and Push Docker Image
2
3
on:
4
push:
5
branches:
6
- master
7
8
jobs:
9
build-and-push:
10
runs-on: windows-latest
11
12
steps:
13
- name: Checkout repo
14
uses: actions/checkout@v3
15
16
- name: Setup MSBuild
17
uses: microsoft/setup-msbuild@v1.1
18
19
- name: Build and publish
20
run: msbuild TakeTheBest-Project\TakeTheBest-Project.sln /p:Configuration=Release /p:DeployOnBuild=true /p:PublishProfile=FolderProfile /p:PublishDir=obj/Docker/publish/
21
22
- name: Log in to DockerHub
23
uses: docker/login-action@v2
24
with:
25
username: \${ secrets.DOCKERHUB_USERNAME }
26
password: \${ secrets.DOCKERHUB_TOKEN }
27
28
- name: Build Docker image
29
run: |
30
docker build `
31
-t \${ secrets.DOCKERHUB_USERNAME }/takethebest-app:latest `
32
-f TakeTheBest-Project\TakeTheBest-Project\Dockerfile `
33
TakeTheBest-Project\TakeTheBest-Project
34
35
- name: Push Docker image to DockerHub
36
run: |
37
docker push \${ secrets.DOCKERHUB_USERNAME }/takethebest-app:latest

TakeTheBestDockerizedApp / .github / workflows /

Add file

iSqloud7

Update docker-publish.yml

7b32a48 · 8 minutes ago

History

Name	Last commit message	Last commit date
..		
docker-publish.yml	Update docker-publish.yml	8 minutes ago

iSqloud7 / TakeTheBestDockerizedApp

Type to search

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Build and Push Docker Image

Update docker-publish.yml #9

Re-run all jobs

Summary

Jobs

Run details

Usage

Workflow file

build-and-push

succeeded 2 minutes ago in 5m 35s

Search logs

Set up job

Checkout repo

Setup MSBuild

Build and publish

Log in to DockerHub

Build Docker image

Push Docker image to DockerHub

Post Log in to DockerHub

Post Checkout repo

Complete job

1s

8s

1s

1m 36s

1s

14s

1m 28s

0s

2s

0s

dockerhub

Explore

My Hub

Search Docker Hub

CtrlK

ivkeex

Docker Personal

Repositories

Collaborations

Settings

Default privacy

Notifications

Repositories

All repositories within the ivkeex namespace.

Search by repository name

All content

Create a repository

Name	Last Pushed	Contains	Visibility	Scout
ivkeex/takethebest-app	3 minutes ago	IMAGE	Public	Scout

ivkeex/takethebest-app

Last pushed 4 minutes ago · Repository size: 2.3 GB

Add a description

Add a category

General

Tags

Image Management

Collaborators

Webhooks

Settings

Tags

DOCKER SCOUT INACTIVE

Activate

This repository contains 0 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	4 minutes

git pull

```
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project> git pull
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 45 (delta 8), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (45/45), 10.19 KiB | 56.00 KiB/s, done.
From https://github.com/iSqloud7/TakeTheBestDockerizedApp
d5680dd..7b32a48  master    -> origin/master
Updating d5680dd..7b32a48
Fast-forward
 .github/workflows/docker-publish.yml | 38 ++++++
 1 file changed, 38 insertions(+)
 create mode 100644 .github/workflows/docker-publish.yml
```

- Kubernetes deployment.

```
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad .\namespace.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: take-the-best

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad .\app-configmap.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\app-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
  namespace: take-the-best
data:
  LOG_LEVEL: "info"

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad .\app-secret.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\app-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secret
  namespace: take-the-best
type: Opaque
data:
  SA_PASSWORD: WW91c\N0cm9uZyFQYXNzdzByZA==

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad .\app-deployment.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\app-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: take-the-best-app
  namespace: take-the-best
spec:
  replicas: 2
  selector:
    matchLabels:
      app: take-the-best-app
  template:
    metadata:
      labels:
        app: take-the-best-app
    spec:
      containers:
        - name: webapp
          image: ivkeex/takethebest-app:latest
          ports:
            - containerPort: 80
          env:
            - name: ConnectionStrings__DefaultConnection
              value: "Server=sqlserver;Database=TakeTheBestDatabase;User Id=sa;Password=$(SA_PASSWORD);TrustServerCertificate=True;"
            - name: SA_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: app-secret
                  key: SA_PASSWORD
            - name: LOG_LEVEL
              valueFrom:
                configMapKeyRef:
                  name: app-config
                  key: LOG_LEVEL

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad .\app-service.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\app-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: take-the-best-app-svc
  namespace: take-the-best
spec:
  selector:
    app: take-the-best-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: ClusterIP

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad .\app-ingress.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\app-ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: take-the-best-ingress
  namespace: take-the-best
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
    - host: take-the-best.local
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: take-the-best-app-svc
                port:
                  number: 80
```



```

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\sqlserver-statefulset.yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: sqlserver
  namespace: take-the-best
spec:
  selector:
    matchLabels:
      app: sqlserver
  serviceName: "sqlserver"
  replicas: 1
  template:
    metadata:
      labels:
        app: sqlserver
    spec:
      containers:
        - name: mssql
          image: mcr.microsoft.com/mssql/server:2019-latest
          ports:
            - containerPort: 1433
          env:
            - name: ACCEPT_EULA
              value: "Y"
            - name: SA_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: app-secret
                  key: SA_PASSWORD
          volumeMounts:
            - name: mssql-data
              mountPath: /var/opt/mssql
      volumeClaimTemplates:
        - metadata:
            name: mssql-data
          spec:
            accessModes: ["ReadWriteOnce"]
            resources:
              requests:
                storage: 5Gi
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> notepad sqlserver-service.yaml
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> cat .\sqlserver-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: sqlserver
  namespace: take-the-best
spec:
  ports:
    - port: 1433
      targetPort: 1433
  clusterIP: None
  selector:
    app: sqlserver

```

kubectl apply -f ...

```

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project> cd .\k8s-manifests\
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f namespace.yaml
namespace/take-the-best created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f app-configmap.yaml
configmap/app-config created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f app-secret.yaml
secret/app-secret created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f sqlserver-service.yaml
service/sqlserver created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f sqlserver-statefulset.yaml
statefulset.apps/sqlserver created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f app-deployment.yaml
deployment.apps/take-the-best-app created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f app-service.yaml
service/take-the-best-app-svc created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl apply -f app-ingress.yaml
ingress.networking.k8s.io/take-the-best-ingress created

```

kubectl get all -n take-the-best

```

PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectl get all -n take-the-best
NAME                                READY    STATUS              RESTARTS   AGE
pod/sqlserver-0                    0/1      ContainerCreating   0           117s
pod/take-the-best-app-65fff9cccd-bj5wk  0/1      ContainerCreating   0           117s
pod/take-the-best-app-65fff9cccd-mn769  0/1      ContainerCreating   0           117s

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/sqlserver                   ClusterIP      None           <none>          1433/TCP   118s
service/take-the-best-app-svc       ClusterIP      10.96.141.235 <none>          80/TCP     116s

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/take-the-best-app    0/2      2              0            117s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/take-the-best-app-65fff9cccd  2          2          0        117s

NAME                                READY    AGE
statefulset.apps/sqlserver           0/1      117s

```

Setting up ingress configurations with minikube.

```
PS C:\WINDOWS\system32> minikube start --driver=docker --cpu=4 --memory=7800
* minikube v1.36.0 on Microsoft Windows 11 Pro 10.0.26100.4484 Build 26100.4484
* Using the docker driver based on user configuration
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.47 ...
* Creating docker container (CPU=4, Memory=7800MB) ...
* Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Donel kubectll is now configured to use "minikube" cluster and "default" namespace by default
PS C:\WINDOWS\system32> minikube addons enable ingress
* Ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
After the addon is enabled, please run "minikube tunnel" and your Ingress resources would be available at "127.0.0.1"
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.5.3
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.5.3
  - Using image registry.k8s.io/ingress-nginx/controller:v1.12.2
* Verifying ingress addon...
* The "ingress" addon is enabled
PS C:\WINDOWS\system32> minikube tunnel
* Tunnel successfully started

* NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-ro
ot-permission
* Starting tunnel for service take-the-best-ingress.
```

Containers

[Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage

12.68% / 800% (8 CPUs available)

Container memory usage

1.01GB / 7.48GB

Show charts

Q Search

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	Last started	Actions	CPU (%)	Memory usage...	Memor
<input type="checkbox"/>	<div><div></div>minikube</div>	13f15f1a3d1d	k8s-miniku	51070:22 Show all ports (5)	7 minutes ago	<div> </div>	12.81%	1.01GB / 7.62GB	13

kubectll get ingress -n take-the-best

```
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectll create namespace take-the-best
namespace/take-the-best created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectll apply -f .\app-ingress.yaml
ingress.networking.k8s.io/take-the-best-ingress created
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectll get ingress -n take-the-best
NAME          CLASS  HOSTS          ADDRESS      PORTS      AGE
take-the-best-ingress  nginx  take-the-best.local  80           8s
```

kubectll describe ingress take-the-best-ingress -n take-the-best

```
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectll describe ingress take-the-best-ingress -n t
ake-the-best
Name:          take-the-best-ingress
Labels:        <none>
Namespace:     take-the-best
Address:       192.168.49.2
Ingress Class: nginx
Default backend: <default>
Rules:
  Host      Path      Backends
  ---      -
take-the-best.local  /  take-the-best-app-svc:80 (<error: services "take-the-best-app-svc" not found>)
Annotations:  nginx.ingress.kubernetes.io/rewrite-target: /
Events:
  Type     Reason     Age          From          Message
  ----     -
Normal    Sync       15s (x2 over 53s)  nginx-ingress-controller  Scheduled for sync
```

kubectll get pods -n take-the-best -w

```
PS C:\Users\Ivan\Desktop\TakeTheBestDockerizedApp\TakeTheBest-Project\TakeTheBest-Project\k8s-manifests> kubectll get pods -n take-the-best -w
NAME          READY   STATUS             RESTARTS   AGE
sqlserver-0   0/1     ContainerCreating   0           65s
take-the-best-app-f87558d4f-qf4d4  0/1     ContainerCreating   0           65s
take-the-best-app-f87558d4f-trm76  0/1     ContainerCreating   0           65s
```


Ограничување со Windows Containers и Kubernetes Ingress

При реализација на проектот, апликацијата „TakeTheBest“ беше докеризирана со користење на **Windows Containers**, бидејќи апликацијата е развиена со **ASP.NET MVC на .NET Framework 4.8**, што бара Windows-средината. Ова значи дека Docker image-от кој се објавува на DockerHub е изграден со **Windows-базиран base image**: mcr.microsoft.com/dotnet/framework/aspnet:4.8.

При преминување на делот со **Kubernetes**, наиде на клучно ограничување:

Kubernetes (Minikube) не поддржува Windows Containers. Kubernetes околината, особено Minikube искористено за локално тестирање, работи исклучиво со **Linux Containers**, а не со Windows. Тоа е проблем бидејќи:

- Windows-based images не можат да се стартуваат во Minikube или да се оркестрираат преку Deployment или StatefulSet.
- **Ingress Controller-ите (NGINX Ingress)**, кои овозможуваат пристап до сервисите преку домен (take-the-best.local), работат **исклучиво со Linux-базирани кластер јазли**.

Поради тоа, иако беше креиран Ingress ресурс со соодветен hostname, **не беше можно да се пристапи до апликацијата преку <http://take-the-best.local>**, бидејќи:

- Има несоодветност помеѓу типот на image (Windows) и кластерот (Linux).
- NGINX Ingress Controller не може да рутира кон Windows pods, бидејќи тие не можат да се креираат.

Дополнително:

- **ASP.NET Framework (4.8)** е зависен од Windows API и не работи на Linux. Затоа за оваа апликација мора да се користат **Windows Containers**.
- **ASP.NET Core** е крос-платформна технологија која работи и на Windows и на Linux. Ако апликацијата беше ASP.NET Core, може да се користи **Linux базирана Docker слика** (mcr.microsoft.com/dotnet/aspnet:7.0), што е многу полесно за работење во Kubernetes, Minikube и останатите оркестрациони алатки.

Во овој случај (класичен ASP.NET Framework 4.8) **не може да користи Linux контејнер, само Windows**. Но ако беше ASP.NET Core — тогаш Linux е опција.

Заклучок

Преку овој проект се прикажа практична примена на Docker и Kubernetes за поставување на ASP.NET MVC апликација со SQL Server. Со користење на CI/CD преку GitHub Actions и локално тестирање, се постигна автоматизирана, стабилна и полесна инфраструктура за развој и deployment на апликацијата.
