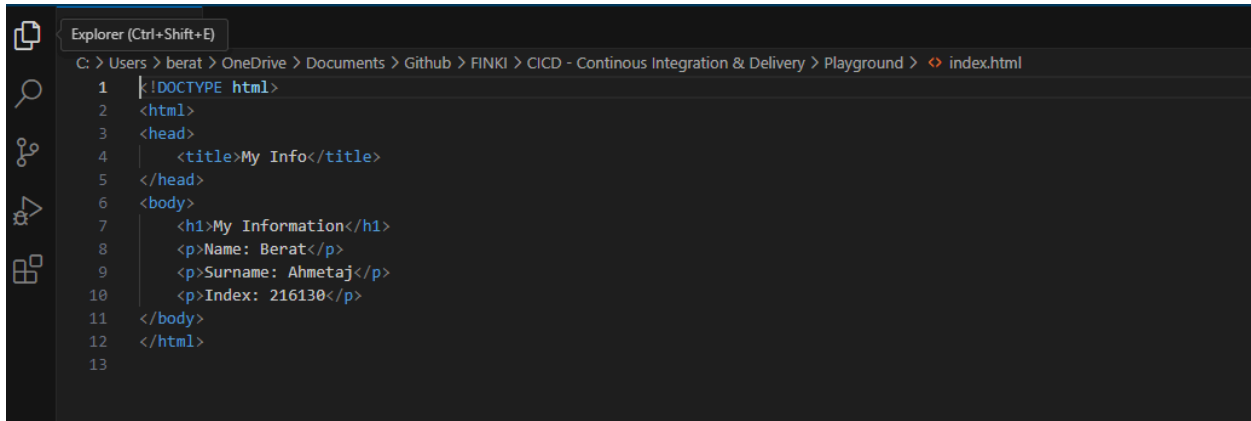


КИИИ Лаб 8

Class Assignment / Homework

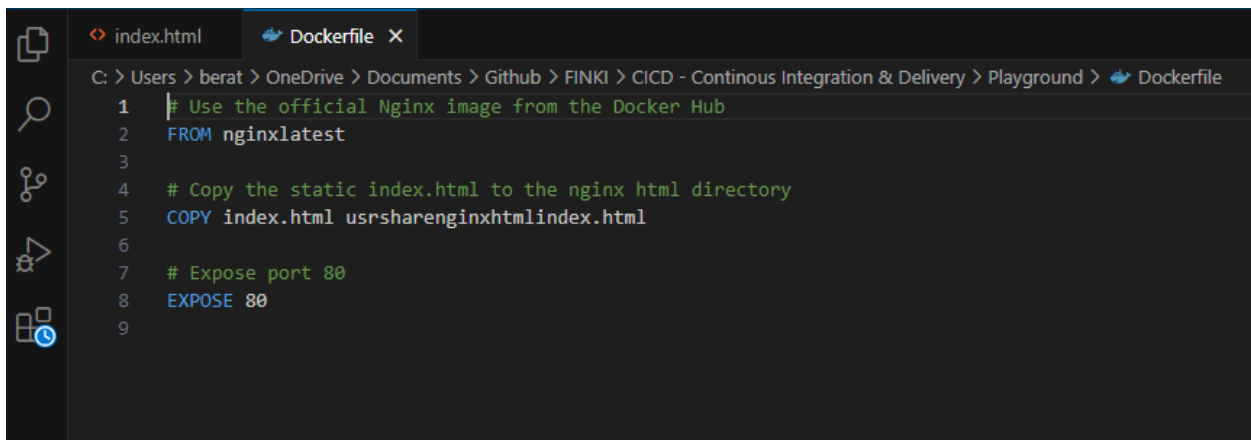
- Create a static index.html with your info in it
 - name, surname, index
- Dockerize it with nginx
- Deploy it to dockerhub with tag 1.0
- Write a deployment manifest
 - 5 replicas, 2 max surge, 1 max unavailable
- Deploy it to your cluster
- Refactor index.html with current year in it - tag 2.0
- Make a rollout with 2.0 image tag
- Revert to 1.0 version

Step 1: Create a index.html with my info

A screenshot of the Visual Studio Code Explorer window. The file explorer on the left shows the path: C: > Users > berat > OneDrive > Documents > Github > FINKI > CICD - Continuous Integration & Delivery > Playground > index.html. The main editor area displays the content of index.html, which is an HTML document with a title 'My Info' and three paragraphs of personal information.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My Info</title>
5 </head>
6 <body>
7   <h1>My Information</h1>
8   <p>Name: Berat</p>
9   <p>Surname: Ahmetaj</p>
10  <p>Index: 216130</p>
11 </body>
12 </html>
13
```

Step 2: Create dockerfile

A screenshot of the Visual Studio Code editor showing two tabs: index.html and Dockerfile. The Dockerfile tab is active, displaying a Dockerfile configuration for an Nginx web server. The file path in the Explorer is: C: > Users > berat > OneDrive > Documents > Github > FINKI > CICD - Continuous Integration & Delivery > Playground > Dockerfile.

```
1 # Use the official Nginx image from the Docker Hub
2 FROM nginx:latest
3
4 # Copy the static index.html to the nginx html directory
5 COPY index.html /usr/share/nginx/html/index.html
6
7 # Expose port 80
8 EXPOSE 80
9
```

Step 3: Build and deploy to docker hub the first build

Command:

```
docker build -t beratahmetaj/my-static-site:1.0 .
```

Result:

```
View build details: docker-desktop://dashboard/build/default/default/z03r7bjabx68ned672qtc2zmm
PS C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continous Integration & Delivery\Playground> docker build -t be
ratahmetaj/my-static-site:1.0 .
[+] Building 0.0s (0/0)  docker:default
2024/05/24 20:38:05 http2: server: error reading preface from client //./pipe/docker_engine: file has already been close
[+] Building 18.6s (7/7) FINISHED  docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 251B 0.0s
=> [internal] load metadata for docker.io/library/nginx:latest 2.3s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 243B 0.0s
=> [1/2] FROM docker.io/library/nginx:latest@sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72a 14.6s
=> => resolve docker.io/library/nginx:latest@sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72af 0.0s
=> => sha256:e784f4560448b14a66f55c26e1b4dad2c2877cc73d001b7cd0b18e24a700a070 7.16kB / 7.16kB 0.0s
=> => sha256:a11fc495baf95699c7cb83ca0878f63f94e34c28837c1da8ae7c9879343604f 41.83MB / 41.83MB 12.5s
=> => sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72af059c 10.27kB / 10.27kB 0.0s
=> => sha256:e688fed0b0c7513a63364959e7d389c37ac8ecac7a6c6a31455eca2f5a71ab8b 2.29kB / 2.29kB 0.0s
=> => sha256:09f376ebb190216b0459f470e71bec7b5dfa611d66bf008492b40dcc5f1d8eae 29.15MB / 29.15MB 7.3s
=> => sha256:933cc84705771a00b189769c48d4a39ba2652e416e1a6f4dd189c21d2b4f8266 628B / 628B 0.7s
=> => sha256:999643392fb738ac637978a3661798ac5f8df2b8ec469817bdef8ac8952b7644 954B / 954B 1.0s
=> => sha256:971bb7f4fb12b17328d6c3461914ea12bf59013c11fba8c72af461e99d716538 394B / 394B 1.3s
=> => sha256:45337c09cd57ea39ca1de37bcf39129b4b650b8a76521ac1fbf88f8183620e80 1.21kB / 1.21kB 1.7s
=> => sha256:de3b062c0af7ef7ffea0dcd16ea18075e2cb2f2e10df1fb8eb44ce7da5921b10 1.40kB / 1.40kB 2.0s
=> => extracting sha256:09f376ebb190216b0459f470e71bec7b5dfa611d66bf008492b40dcc5f1d8eae 2.1s
=> => extracting sha256:a11fc495baf95699c7cb83ca0878f63f94e34c28837c1da8ae7c9879343604f 1.4s
=> => extracting sha256:933cc84705771a00b189769c48d4a39ba2652e416e1a6f4dd189c21d2b4f8266 0.0s
=> => extracting sha256:999643392fb738ac637978a3661798ac5f8df2b8ec469817bdef8ac8952b7644 0.0s
=> => extracting sha256:971bb7f4fb12b17328d6c3461914ea12bf59013c11fba8c72af461e99d716538 0.0s
=> => extracting sha256:45337c09cd57ea39ca1de37bcf39129b4b650b8a76521ac1fbf88f8183620e80 0.0s
=> => extracting sha256:de3b062c0af7ef7ffea0dcd16ea18075e2cb2f2e10df1fb8eb44ce7da5921b10 0.0s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html 1.4s
=> exporting to image 0.1s
=> => exporting layers 0.0s
=> => writing image sha256:4e63083325524025e1fb613df78906e07e720ca5dfb88f90e6c749657ca65901 0.0s
=> => naming to docker.io/beratahmetaj/my-static-site:1.0 0.0s

View build details: docker-desktop://dashboard/build/default/default/z03r7bjabx68ned672qtc2zmm

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continous Integration & Delivery\Playground> |
```

Then we push the build to DockerHub

Command:

```
docker push beratahmetaj/my-static-site:1.0
```

Step 4: Deployment Manifest

Deployment.yaml

```
C: > Users > berat > OneDrive > Documents > Github > FINKI > CICD - Continuous Integration & Delivery > Playground > ! deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: my-static-site-deployment
5  spec:
6    replicas: 5
7    strategy:
8      rollingUpdate:
9        maxSurge: 2
10       maxUnavailable: 1
11   selector:
12     matchLabels:
13       app: my-static-site
14   template:
15     metadata:
16       labels:
17         app: my-static-site
18     spec:
19       containers:
20       - name: my-static-site
21         image: beratametaj/my-static-site:1.0
22         ports:
23         - containerPort: 80
```

Kubectl deployment.yaml

```
PS C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continuous Integration & Delivery\Playground> k
ubectl apply -f deployment.yaml
deployment.apps/my-static-site-deployment created
PS C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continuous Integration & Delivery\Playground>
```

Step 5: Refactor Index.html and deploy again

Index.html

```
index.html
C: > Users > berat > OneDrive > Documents > Github > FINKI > CICD - Continuous Integration & Delivery > Playground > index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>My Info</title>
5  </head>
6  <body>
7    <h1>My Information</h1>
8    <p>Name: Berat</p>
9    <p>Surname: Ahmetaj</p>
10   <p>Index: 216130</p>
11   <p>Year: <span id="currentYear"></span></p>
12
13   <script>
14     document.getElementById("currentYear").innerText = new Date().getFullYear();
15   </script>
16 </body>
17 </html>
18
```

New docker build

```
docker build -t berat Ahmetaj/my-static-site:2.0 .
```

Then push to the DockerHub as v2.0

```
docker push berat Ahmetaj/my-static-site:2.0
```

Proof of these steps:

```
P5 C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continuous Integration & Delivery\Playground>
docker build -t berat Ahmetaj/my-static-site:2.0 .
[+] Building 0.0s (0/0) docker:default
2020/05/20 21:32:05 http2: server: error reading preface from client //./pipe/docker_engine: file has
[+] Building 2.1s (0/0) FIM19M0
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 281B 0.0s
=> [internal] load metadata for docker.io/library/nginx:latest 1.7s
=> [auth] library/nginx:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.1s
=> [internal] load build context 0.1s
=> => transferring context: 0B 0.1s
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:a48d19ebd6211f5299934ac80f6ad81b06 0.0s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html 0.1s
=> exporting to image 0.1s
=> => exporting layers 0.0s
=> writing image sha256:b1172a07152eb8b0e218f363bb161502c6c5682a197f0f6336f37173c6dc3d 0.0s
=> naming to docker.io/berat Ahmetaj/my-static-site:2.0 0.0s
View build details: docker-desktop://dashboard/build/default/default/9accjyqubunplyzedheg5ii
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
P5 C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continuous Integration & Delivery\Playground>
docker push berat Ahmetaj/my-static-site:2.0
The push refers to repository [docker.io/berat Ahmetaj/my-static-site]
80712ba569: Pushed
1077370806d: Layer already exists
762f6b9c368c: Layer already exists
56f6f6e6d0d: Layer already exists
9f4d72e635f1: Layer already exists
707b79baeb8: Layer already exists
f4c5c6a3139: Layer already exists
8d4d2766acc: Layer already exists
2.0: digest: sha256:2e1ec8379509e7a00ebd1f630211u11u4f09f0e55d34f05356e97f052 size: 1905
P5 C:\Users\berat\OneDrive\Documents\Github\FINKI\CICD - Continuous Integration & Delivery\Playground>
```

Deployment change to v2.0

```
app: my-static-site
spec:
  containers:
  - name: my-static-site
    image: beratahmetaj/my-static-site:2.0 # VERZIJA 2.0 TAG
    ports:
    - containerPort: 80
```


Run kubectl with deployment.yaml now

kubectl apply -f deployment.yaml



FINAL STEP: REVERT BACK TO 1.0



To Revert we just change the deployment.yaml to 1.0 and run

kubectl apply -f deployment.yaml again

beratahmetaj/my-static-site 

Updated 3 minutes ago

This repository does not have a description   INCOMPLETE

This repository does not have a category   INCOMPLETE

Docker commands





To push a new tag to this repository:

```
docker push beratahmetaj/my-static-site:tagname
```

[Public View](#)

Tags


This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
 2.0		Image	2 minutes ago	3 minutes ago
 1.0		Image	an hour ago	an hour ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#) .

[Upgrade](#)