

# + Deploying a Web Server Using Docker and Kubernetes on Ubuntu

## 1. Create a Dockerfile:

web-app-project > Dockerfile

```
1 FROM nginx:latest
2 COPY index.html /usr/share/nginx/html/index.html
```

## 2. Create an index.html file:

- Index.html code

web-app-project > index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>DevOps Hub</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 0;
11      padding: 0;
12      background-color: #f4f4f4;
13      color: #333;
14    }
15    header {
16      background: #333;
17      color: white;
18      text-align: center;
19      padding: 20px;
20      font-size: 24px;
21    }
22    nav {
23      text-align: center;
24      padding: 15px;
25      background: #008080;
26    }
27    nav a {
```

## 3. Build the Docker image:

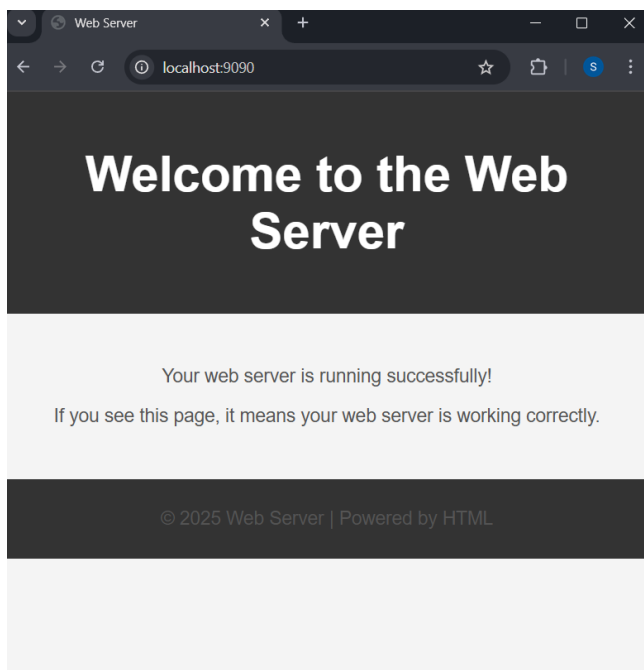
```
50
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Docker\web-app-project> docker build -t web-server-image .
[+] Building 20.5s (8/8) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 104B                             0.0s
=> [internal] load metadata for docker.io/library/nginx:latest  3.9s
=> [auth] library/nginx:pull token for registry-1.docker.io     0.0s
=> [internal] load .dockerignore                                0.1s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 1.27kB                               0.0s

PS C:\Docker\web-app-project> docker run -d -p 9090:80 web-server-image
b246ce16eeee542126c83e8384f94cfda488ef1a5ba29669ee2cd5a0f2711759
```

### 3.1 Access on web browser:

- http://localhost:9090



## 4. Push the Docker Image to Docker Hub:

### 4.1 Tag the docker image & login to dockerhub-

```
PS C:\Docker\web-app-project> docker tag web-server-image shrutikadewarde/web-server-image:latest
PS C:\Docker\web-app-project> docker login
Authenticating with existing credentials...
Login Succeeded
```



### 4.2



```
PS C:\Docker\web-app-project> docker push shrutikadewarde/web-server-image:latest
The push refers to repository [docker.io/shrutikadewarde/web-server-image]
3468150c9614: Pushed
```

### 4.3 Verify the Image on Docker Hub-

## shrutikadewarde/web-server-image

Last pushed 6 minutes ago • Repository size: 68.8 MB

Add a description  

Add a category  

General Tags Image Management **BETA** Collaborators Webhooks Settings

### Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	less than 1 day	7 minutes

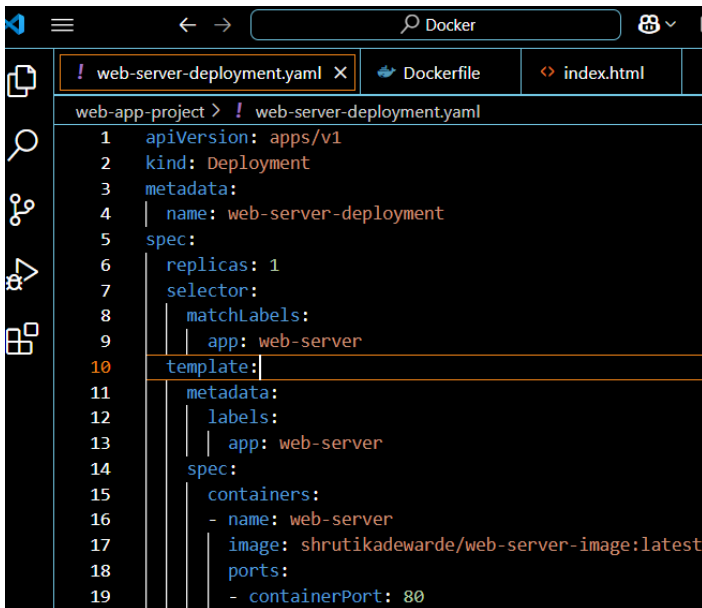
[See all](#)

## 5. Set Up a Kubernetes Cluster

```
-minikube start
```

Deploy the Web Server on Kubernetes

### 5.1 Create a Kubernetes Deployment Configuration (web-server-deployment.yaml):



```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: web-server-deployment
5 spec:
6   replicas: 1
7   selector:
8     matchLabels:
9       app: web-server
10  template:
11    metadata:
12      labels:
13        app: web-server
14    spec:
15      containers:
16      - name: web-server
17        image: shrutikadewarde/web-server-image:latest
18        ports:
19        - containerPort: 80
```

### 5.2 Apply and verify the Deployment Configuration:

```
- kubectl apply -f web-server-deployment.yaml
```

```
- kubectl get deployments
```

### 5.3 Create a Kubernetes Service configuration:

```
web-app-project > ! web-server-service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    | name: web-server-service
5  spec:
6    | selector:
7      | app: web-server
8    | ports:
9      - protocol: TCP
10     | port: 80
11     | targetPort: 80
12    | type: NodePort
```

Apply and verify:

6 Access the Web Server Application:

6.1 Retrieve the Node port:

- `kubect1 get service web-server-service`

6.2 Access the application:

- `Minikube service web-server-service`

```
PS C:\Docker\web-app-project> kubect1 get service web-server-service
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
web-server-service  NodePort    10.102.2.180  <none>         80:32237/TCP     2m56s
PS C:\Docker\web-app-project> minikube service web-server-service
|-----|
| NAMESPACE | NAME           | TARGET PORT | URL                |
|-----|
| default   | web-server-service | 80          | http://192.168.49.2:32237 |
|-----|
* Starting tunnel for service web-server-service.
|-----|
| NAMESPACE | NAME           | TARGET PORT | URL                |
|-----|
| default   | web-server-service |             | http://127.0.0.1:53058 |
|-----|
```

7 Cleaning up resources:

-`kubect1 delete -f web-server-service.yaml`

`kubect1 delete -f web-server-deployment.yaml`

