

# Use Docker commands to build, tag, and push Docker images to a registry

---

## Table of Contents

---

- [Description](#)
- [Problem Statement](#)
- [Prerequisites](#)
  - [Software Requirement](#)
  - [Hardware Requirement](#)
- [Implementation Steps](#)
  - [Step-1: Build the Docker Image](#)
  - [Step-2: Tag the Docker Image](#)
  - [Step-3: Push the Docker Image to a Registry](#)
- [References](#)

## Description

---

This guide explains how to build, tag, and push Docker images for your **ToDoApp** using Docker commands. After creating a Dockerfile for the web application, you'll need to build the image, assign a tag, and push it to a remote registry (such as Docker Hub or a private registry).

## Problem Statement

---

To deploy your **ToDoApp** in different environments or share it with others, you must publish the Docker image to a registry. This requires:

- Building a Docker image from the application code.
- Tagging the image with a unique identifier (such as a version number).
- Pushing the image to a Docker registry for distribution.

## Prerequisites

---

Completion of all previous lab guides (up to Lab Guide-01) is required before proceeding with Lab Guide-02.

### Software Requirement

- **Docker Desktop:** Installed on your Windows machine.
- **Docker Hub Account** (or any other Docker-compatible registry).
- **ToDoAPP\_MYSQL:** To download the source folder [click here](#)

### Hardware Requirement

- Minimum of 4 GB RAM.

- At least 2 cores in the processor.

## Implementation Steps

### Step-1: Build the Docker Image

1. **Create a Dockerfile** for your TodoApp (if not done already). Here is an example Dockerfile for a simple Java-based web application:

```
FROM openjdk:11.0.15-jre
ADD target/*.jar app.jar
ENTRYPOINT ["java", "-jar", "app.jar"]
```

2. **Build the Docker image:**

```
docker build -t todoapp .
```

In this example:

- **todoapp:latest** is the tag you are assigning to the image. The **latest** tag indicates that this is the most current version.
- The **.** specifies that Docker should look for the Dockerfile in the current directory.

```
C:\Users\Administrator\Documents\TodoApp_MySQL-main\todoapp>docker build -t todoapp .
[+] Building 4.2s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 125B                                              0.1s
=> [internal] load metadata for docker.io/library/openjdk:11.0.15-jre          2.5s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 83B                                                 0.0s
=> [1/2] FROM docker.io/library/openjdk:11.0.15-jre@sha256:b90104c2eec246d8b6aec962456499f0163a5b58fcfc10fe8027 0.1s
=> => resolve docker.io/library/openjdk:11.0.15-jre@sha256:b90104c2eec246d8b6aec962456499f0163a5b58fcfc10fe8027 0.1s
=> CACHED [2/2] ADD target/*.jar app.jar                                         0.0s
=> exporting to image                                                            1.1s
=> => exporting layers                                                         0.0s
=> => exporting manifest sha256:0775913eef0155d3b95f98522e9b3f89aa753f769c5f02f966afe16ebc0d4091          0.0s
=> => exporting config sha256:7d62a3ce7ab07ec4a7c028ad9c709265b78037fa2711f8e6cf1268646410eebe          0.0s
=> => exporting attestation manifest sha256:dcba0493d44dc55df0060cfc3934b2da7ffb932a0e7ecba94bdc6a55a9f5295b 0.1s
=> => exporting manifest list sha256:24fc48428bd35ce6595a2f597daf2c05730ee5c8cdc42052fd2d3dd0ba4b8e7e      0.0s
=> => naming to docker.io/library/todoapp:latest                             0.0s
=> => unpacking to docker.io/library/todoapp:latest                           0.9s

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\Users\Administrator\Documents\TodoApp_MySQL-main\todoapp>
```

3. Verify that the image was built successfully:

```
docker images
```

This command will list all available Docker images. You should see **todoapp** in the list.

```
C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
todoapp       latest    24fc48428bd3   29 hours ago   557MB

C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>_
```

## Step-2: Tag the Docker Image

Next, assign a tag to the image so that it can be pushed to a registry. Tags generally include the registry URL, the repository name, and a version number.

```
docker tag todoapp:latest <your-dockerhub-username>/todoapp:v1.0
```

In this example:

- **<your-dockerhub-username>** is your Docker Hub username.
- **todoapp:v1.0** assigns a version (**v1.0**) to the image.

Now, when you list images using **docker images**, you should see the newly tagged image.

```
C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>docker tag todoapp sooryav/todoapp:v1.0
C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>_
```

```
C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
sooryav/todoapp  v1.0      24fc48428bd3   29 hours ago   557MB
todoapp       latest    24fc48428bd3   29 hours ago   557MB

C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>_
```

## Step-3: Push the Docker Image to a Registry

To make the image accessible to others, push it to a Docker registry. For this example, we will use Docker Hub, but you can push to any compatible registry.

### 1. Login to Docker Hub:

```
docker login
```

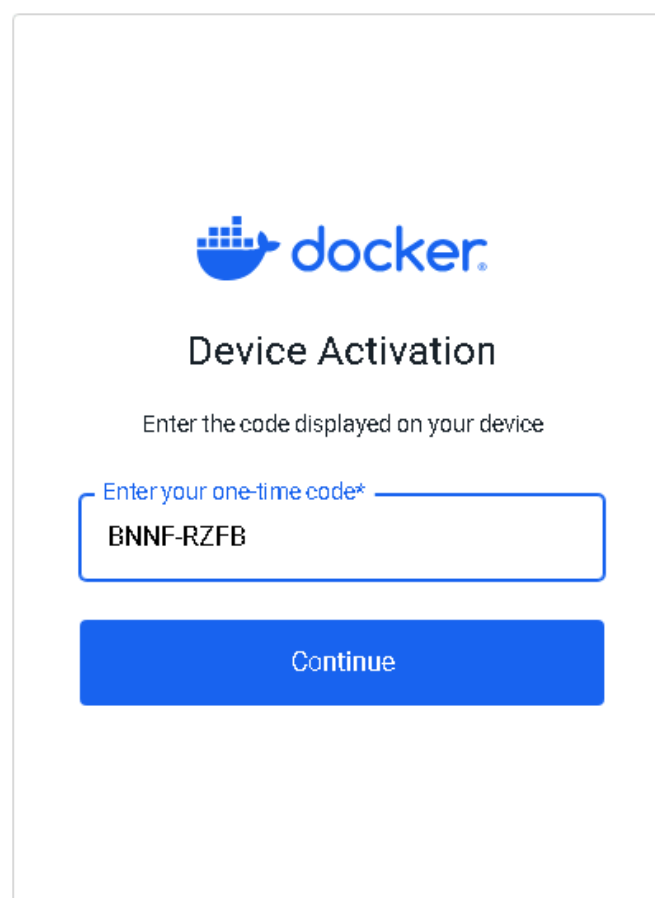
```
C:\Users\Administrator\Documents\TodoApp_MySQL-main\todoapp>docker login


USING WEB BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: BNNF-RZFB
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...
-
```

- Copy the device confirmation code and submit the device code here: <https://login.docker.com/activate>



 **docker**


## Device Activation

Enter the code displayed on your device

Enter your one-time code\*

[Continue](#)

- You'll be prompted to enter your Docker Hub username and password.





## Sign in

Using Docker for work? We recommend signing in with your work email address.

Continue

OR

 Continue with Google

 Continue with GitHub

Don't have an account? [Sign Up](#)

- Once the device is connected, you should should an output like this:



**Congratulations, you're all  
set!**

Your device is now connected.

```
C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>docker login

USING WEB BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: PZMV-DWFD
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...
Login Succeeded
```

2. **Push the tagged image** to Docker Hub:

```
docker push <your-dockerhub-username>/todoapp:v1.0
```

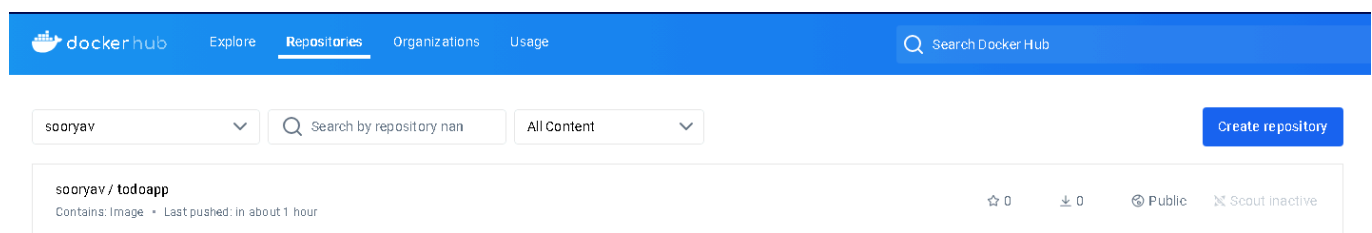
Docker will upload the image to your Docker Hub repository.

```
C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>docker push sooryav/todoapp:v1.0
The push refers to repository [docker.io/sooryav/todoapp]
d1989b6e74cf: Pushed
19babefbed06: Pushed
529e95713a10: Pushed
72f9b148e5a2: Pushed
66a9e63c657a: Pushed
d836772a1c1f: Pushed
09bd66433199: Pushed
cc8fcf8b61a2: Pushed
v1.0: digest: sha256:24fc48428bd35ce6595a2f597daf2c05730ee5c8cdc42052fd2d3dd0ba4b8e7e size: 856

C:\Users\Administrator\Documents\ToDoApp_MySQL-main\todoapp>_
```

### 3. Verify the image:

Once the push is complete, visit [Docker Hub](https://hub.docker.com/) and verify that your image is available in your repository.



## References

- Docker Hub Documentation: <https://docs.docker.com/docker-hub/>
- Docker Build and Push: <https://docs.docker.com/engine/reference/commandline/push/>
- Tagging Docker Images: <https://docs.docker.com/engine/reference/commandline/tag/>