

# Docker Basics

## Overview

This documentation demonstrates the basics of Docker containerization using a simple Node.js note-taking application. The project showcases how to create, build, and run Docker containers, as well as push images to Docker registries.

## Prerequisites

- Ubuntu 22.04 LTS
- Node.js and npm installed
- Docker installed and configured
- Docker Hub account

## Dockerfile Explanation

```
FROM node:18-alpine          # Base image with Node.js 18 on Alpine Linux
WORKDIR /usr/src/app         # Set working directory inside container
COPY package*.json ./        # Copying all package files
RUN npm install               # Install Node.js dependencies
COPY . .                     # Copy all project files to container
EXPOSE 3000                  # Document that app uses port 3000
CMD ["npm", "start"]         # Command to start the application
```

## Step-by-Step Instructions

### 1. Environment Setup

**Command:** Install required software

```
sudo apt update && sudo apt upgrade -y
sudo apt install nodejs npm -y
sudo apt install docker.io -y
sudo usermod -aG docker $USER
```

**Screenshot Required:** Terminal showing version outputs

- `node --version`
- `npm --version`
- `docker --version`

### 2. Project Initialization

**Command:** Create and initialize project

```
mkdir note-taking-app
cd note-taking-app
npm init -y
npm install express
```

### 3. Create Application Files

**Command:** Create main application file

```
nvim app.js
# Copy the provided application code
```

## 4. Test Local Application

**Command:** Run application locally

```
npm start
```

```
→ npm start
> simple-notes-app@1.0.0 start
> node app.js

Notes App running on http://localhost:3000
Create, edit, and delete your notes!
```

## 5. Create Docker Configuration

**Command:** Create Dockerfile

```
nvim Dockerfile
```

```
File: Dockerfile
1 FROM node:18-alpine
2
3 WORKDIR /usr/src/app
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 COPY . .
10
11 EXPOSE 3000
12
13 CMD ["npm", "start"]
14
15
```

## 6. Build Docker Image

**Command:** Build the Docker image

```
docker build -t simple-notes-app .
```

```
→ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
note-taking-app latest 254dcbca1ba 6 seconds ago 135MB
note-taking-app v1 18ffa1a702 14 minutes ago 135MB
```

## 7. Run Docker Container

**Command:** Run the containerized application

```
docker run -p 3000:3000 simple-notes-app
```

## 8. Docker Container Management

**Command:** Various container management operations

```
# Run in detached mode
docker run -d -p 3000:3000 --name notes-container simple-notes-app
```

```
→ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
08cf7a5326bb note-taking-app:latest "docker-entrypoint.s..." 24 minutes ago Up 3 seconds 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp note-container
```

```
# View running containers
docker ps
```

```
# View container logs
```

```
docker logs notes-container
```

```
→ docker logs note-container
> simple-notes-app@1.0.0 start
> node app.js

Notes App running on http://localhost:3000
Create, edit, and delete your notes!
npm error path /usr/src/app
npm error command failed
npm error signal SIGTERM
npm error command sh -c node app.js
npm error A complete log of this run can be found in: /root/.npm/_logs/2025-08-12T06_18_00_006Z-debug-0.log
> simple-notes-app@1.0.0 start
> node app.js

Notes App running on http://localhost:3000
Create, edit, and delete your notes!
```

```
# Stop container
docker stop notes-container
```

```
# Remove container
docker rm notes-container
```

## 9. Docker Registry Operations

**Command:** Login and push to Docker Hub

```
# Login to Docker Hub
```

```
docker login
```

```
→ docker login -u prash3
Info → A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password:
WARNING! Your credentials are stored unencrypted in '/home/prashant/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded

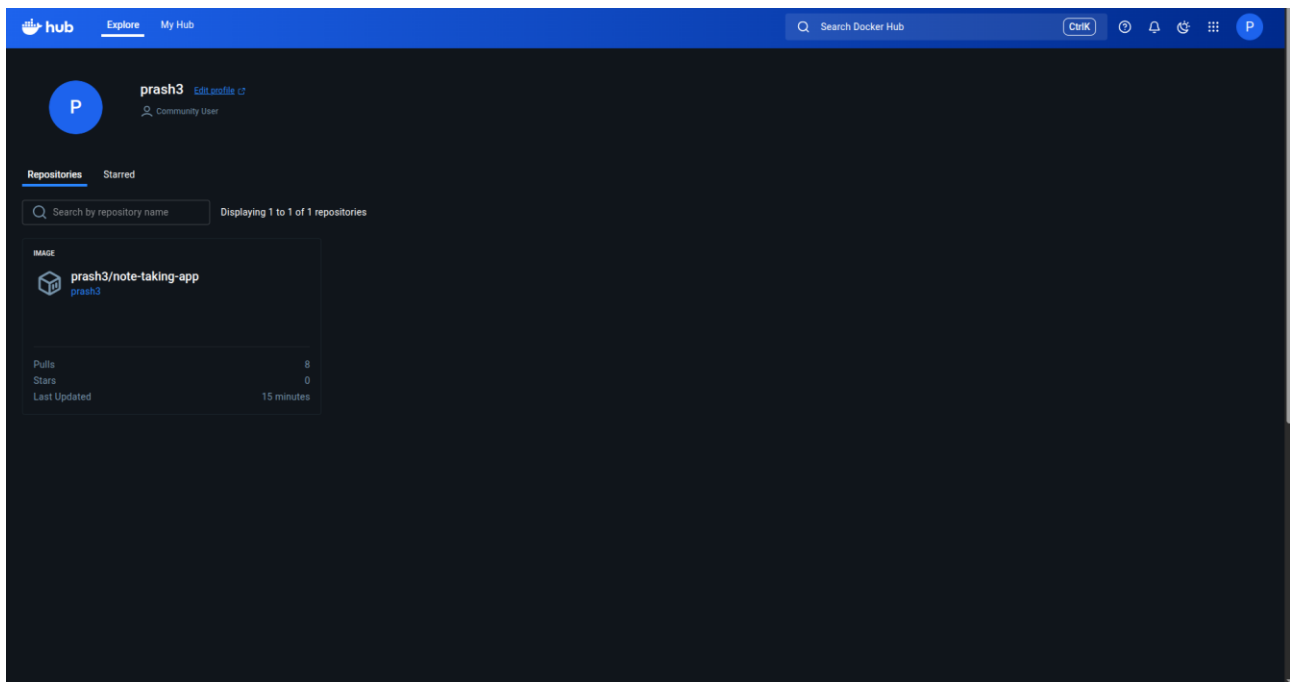
Prashant/node_project/note-taking-app via v12.22.9 took 9s
→ docker push prash3/note-taking-app:latest
The push refers to repository [docker.io/prash3/note-taking-app]
An image does not exist locally with the tag: prash3/note-taking-app

Prashant/node_project/note-taking-app via v12.22.9
→ docker tag note-taking-app prash3/note-taking-app:latest

Prashant/node_project/note-taking-app via v12.22.9
→ docker push prash3/note-taking-app:latest
The push refers to repository [docker.io/prash3/note-taking-app]
b6f4e5308b17: Pushed
993879eae434: Pushed
088c23f478bd: Pushed
3252d0e9e48: Pushed
82140d9a70a7: Mounted from library/node
f2b40b0cdbc1: Mounted from library/node
001f2e057bd0: Mounted from library/node
00000c18d1d: Mounted from library/node
latest: digest: sha256:18aa47e9925bf259e96699ba7b94b84d4bf61a9a972195d290ed0f554a68be56 size: 1995
```

```
# Tag the image
docker tag simple-notes-app prash3/simple-notes-app:latest
```

```
# Push to registry
docker push prash3/simple-notes-app:latest
```



## Key Docker Commands Used

Command	Purpose
<code>docker build -t &lt;name&gt; .</code>	Build image from Dockerfile
<code>docker run -p &lt;host&gt;:&lt;container&gt; &lt;image&gt;</code>	Run container with port mapping
<code>docker ps</code>	List running containers
<code>docker images</code>	List available images
<code>docker logs &lt;container&gt;</code>	View container logs
<code>docker stop &lt;container&gt;</code>	Stop running container
<code>docker rm &lt;container&gt;</code>	Remove container
<code>docker rmi &lt;image&gt;</code>	Remove image
<code>docker login</code>	Login to Docker registry
<code>docker tag &lt;image&gt; &lt;new-tag&gt;</code>	Tag image for registry
<code>docker push &lt;image&gt;</code>	Push image to registry
<code>docker pull &lt;image&gt;</code>	Pull image from registry

## Docker Concepts Demonstrated

### 1. Containerization

- Packaging application with all dependencies
- Ensuring consistent runtime environment
- Isolation from host system

### 2. Image Layers

- Base image (node:18-alpine)
- Application layer (copied files)
- Dependencies layer (npm install)

### 3. Port Mapping

- Container port 3000 mapped to host port 3000
- Multiple containers on different host ports

### 4. Registry Operations

- Image versioning and tagging
- Push and pull operations
- Public image distribution

### 5. Container Lifecycle

- Build → Run → Stop → Remove
- Persistent vs ephemeral containers
- Container management commands

## Troubleshooting Common Issues

Issue	Solution
Port already in use	Use different port: <code>-p 3001:3000</code>
Permission denied	Add user to docker group: <code>sudo usermod -aG docker \$USER</code>
Build fails	Check Dockerfile syntax and file paths
Container won't start	Check logs: <code>docker logs &lt;container-name&gt;</code>
Push denied	Verify login: <code>docker login</code>

## Port already in use error

```
→ npm start
> simple-notes-app@1.0.0 start
> node app.js
events.js:291
    throw er; // Unhandled 'error' event
    ^
Error: listen EADDRINUSE: address already in use :::3000
    at Server.setupListenHandle [as _listen2] (net.js:1310:10)
    at listenInCluster (net.js:1338:12)
    at Server.listen (net.js:1464:7)
    at Function.listen (/home/prashantn/Prashant/node_project/note-taking-app/node_modules/express/lib/application.js:635:24)
    at Object.<anonymous> (/home/prashantn/Prashant/node_project/note-taking-app/app.js:361:5)
    at Module._compile (internal/modules/cjs/loader.js:999:38)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
    at Module.load (internal/modules/cjs/loader.js:863:32)
    at Function.Module._load (internal/modules/cjs/loader.js:768:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:60:12)
Emitted 'error' event on Server instance at:
    at emitErrorNT (net.js:1337:8)
    at processTicksAndRejections (internal/process/task_queues.js:84:21) {
  code: 'EADDRINUSE',
  errno: 'EADDRINUSE',
  syscall: 'listen',
  address: ':::',
  port: 3000
}
```

# Running application

