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
```

5. Create Docker Configuration

Command: Create Dockerfile

```
nvim Dockerfile
```

6. Build Docker Image

Command: Build the Docker image

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

7. Run Docker Container

Command: Run the containerized application

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

8. Docker Container Management

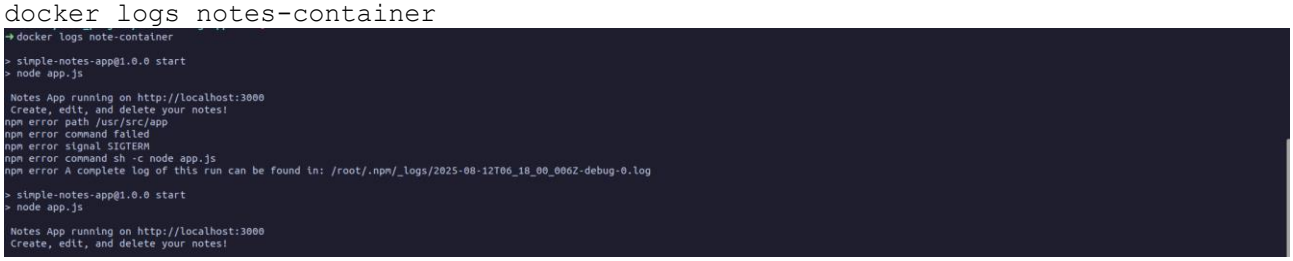
Command: Various container management operations

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


```
# View running containers
docker ps

# View container logs
```

```
docker logs notes-container
```




```
# Stop container
docker stop notes-container

# Remove container
docker rm notes-container
```

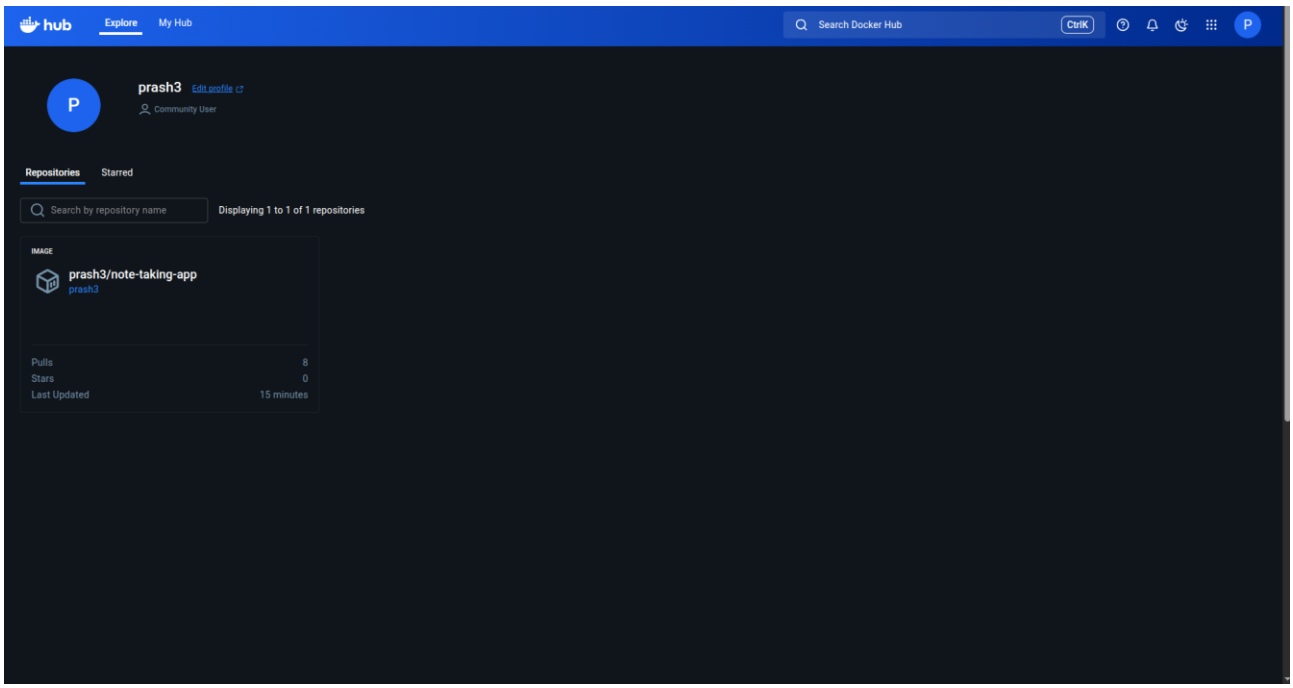
9. Docker Registry Operations

Command: Login and push to Docker Hub

```
# Login to Docker Hub
docker login
```


```
# 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
`docker build -t <name> .`	Build image from Dockerfile
`docker run -p <host>:<container> <image>`	Run container with port mapping
`docker ps`	List running containers
`docker images`	List available images
`docker logs <container>`	View container logs
`docker stop <container>`	Stop running container
`docker rm <container>`	Remove container
`docker rmi <image>`	Remove image
`docker login`	Login to Docker registry
`docker tag <image> <new-tag>`	Tag image for registry
`docker push <image>`	Push image to registry
`docker pull <image>`	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: `-p 3001:3000`
Permission denied	Add user to docker group: `sudo usermod -aG docker $USER`
Build fails	Check Dockerfile syntax and file paths
Container won't start	Check logs: `docker logs <container-name>`
Push denied	Verify login: `docker login`

Port already in use error

Running application