

**NAME : UDDHAV BAPU GUND | DESIGNATION: JR.AWS DEVELOPER**

## **TASK : Docker Basics**

- Create a simple Dockerfile for a Node.js/Java app.
  - Build and run the container.
  - Push image to DockerHub.
- 

### **1. Introduction**

This report demonstrates the completion of the Docker Basics task using a simple Node.js application. The task included creating a Dockerfile, building and running the container, and pushing the image to Docker Hub.

### **2. Prerequisites**

- Docker Desktop installed on Windows(IT'S NOT NEEDED IF WE USE DOCKER HUB)
- Docker Hub account (Username: uddhav345)
- Command Prompt/PowerShell for running Docker commands

### **3. Project Setup**

USE `docker login` command to login at my account

Created a folder named '`node-docker`' and added the following files in folder:

Open **PowerShell** and create a folder:

```
mkdir node-docker  
cd node-docker
```

The screenshot shows a Windows PowerShell window titled 'Windows PowerShell'. The command `mkdir node-docker` is entered and executed, followed by `cd node-docker`. The current directory is then shown as `PS D:\devops_docker>`. The user then runs `docker login`, which prompts for existing credentials for the username `uddhav345`. A message indicates that the login succeeded. Finally, the user runs `cd node-docker` again to change the directory to the newly created folder.

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
PS D:\devops_docker> docker login  
Authenticating with existing credentials... [Username: uddhav345]  
  
[ Info → To login with a different account, run 'docker logout' followed by 'docker login'  
  
Login Succeeded  
PS D:\devops_docker> mkdir node-docker  
  
Directory: D:\devops_docker  
  
Mode LastWriteTime Length Name  
---- ----- ---- --  
d---- 13-09-2025 10:37 node-docker  
  
PS D:\devops_docker> cd node-docker
```

- package.json

The screenshot shows the VS Code interface with the package.json tab selected. The code editor displays the following JSON configuration:

```
1  {
2    "name": "node-hello",
3    "version": "1.0.0",
4    "main": "index.js",
5    "scripts": {
6      "start": "node index.js"
7    },
8    "dependencies": {
9      "express": "^4.18.2"
10   }
11 }
12 }
```

The status bar at the bottom shows various extensions and settings, including "BLACKBOX Agent", "Open Website", "Spaces: 4", "UTF-8", "CRLF", "JSON", "Go Live", "AI Code Chat", and "Prettier".

- index.js

The screenshot shows the VS Code interface with the index.js tab selected. The code editor displays the following Express.js application code:

```
1  const express = require('express');
2  const app = express();
3  const port = process.env.PORT || 3000;
4
5  app.get('/', (req, res) => res.send('Hello from Node in Docker!'));
6
7  app.listen(port, () => console.log(`App listening on ${port}`));
8  |
```

The status bar at the bottom shows various extensions and settings, including "BLACKBOX Agent", "Open Website", "Spaces: 4", "UTF-8", "CRLF", "JavaScript", "Go Live", "AI Code Chat", and "Prettier".

- .dockerignore

A screenshot of the Visual Studio Code interface. The title bar shows the project name "node-docker". The tabs at the top are "Welcome", "Dockerfile", "Welcome", ".dockerignore", and "index.js". The ".dockerignore" tab is active, highlighted with a yellow background. The main editor area displays the contents of the ".dockerignore" file:

```
1 node_modules
2 npm-debug.log
3
```

The left sidebar contains icons for "File", "Search", "Problems", "... (three dots)", "User", and "Settings". The bottom status bar shows various icons and text, including "BLACKBOX Agent", "Open Website", "Ignore", "Go Live", "AI Code Chat", "Prettier", and a notification bell.

- Dockerfile

A screenshot of the Visual Studio Code interface, identical to the previous one but with the "Dockerfile" tab active. The title bar shows the project name "node-docker". The tabs at the top are "Welcome", "Dockerfile", "Welcome", ".dockerignore", and "index.js". The "Dockerfile" tab is active, highlighted with a yellow background. The main editor area displays the contents of the Dockerfile:

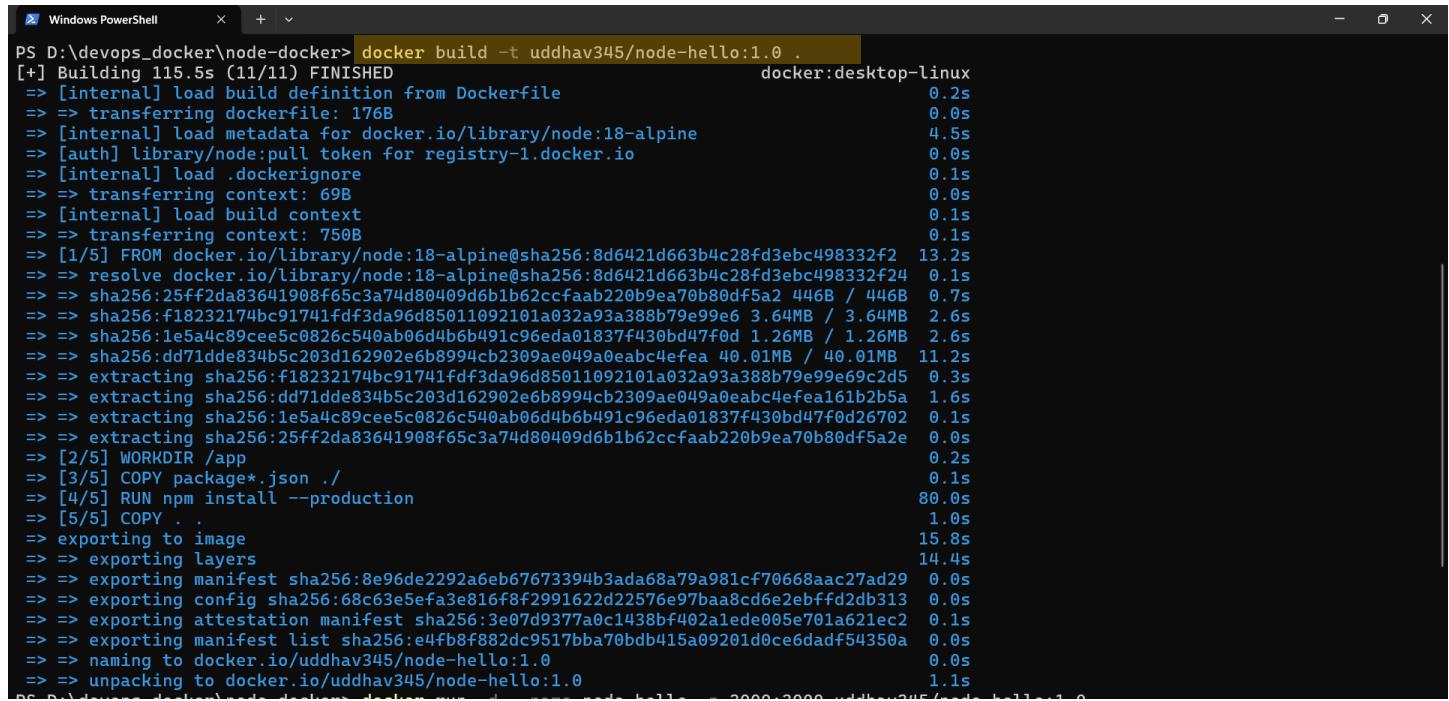
```
1 FROM node:18-alpine
2 WORKDIR /app
3 COPY package*.json .
4 RUN npm install --production
5 COPY .
6 EXPOSE 3000
7 CMD ["node", "index.js"]
8
```

The left sidebar contains icons for "File", "Search", "Problems", "... (three dots)", "User", and "Settings". The bottom status bar shows various icons and text, including "BLACKBOX Agent", "Open Website", "Dockerfile", "Ignore", "Go Live", "AI Code Chat", "Prettier", and a notification bell.

## 4. Build Docker Image

Executed the following command to build the Docker image:

```
docker build -t <YOUR_DOCKERHUB_USERNAME>/node-hello:1.0 .
```

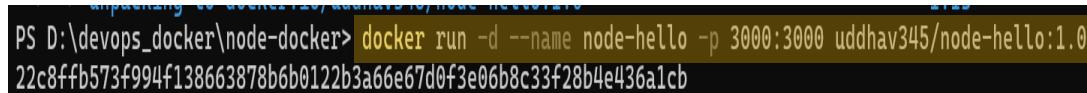


```
PS D:\devops_docker\node-docker> docker build -t uddhav345/node-hello:1.0 .
[+] Building 115.5s (11/11) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from Dockerfile                           0.2s
=> => transferring dockerfile: 176B                                         0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine           4.5s
=> [auth] library/node:pull token for registry-1.docker.io                  0.0s
=> [internal] load .dockerrcignore                                         0.1s
=> => transferring context: 69B                                           0.0s
=> [internal] load build context                                         0.1s
=> => transferring context: 750B                                         0.1s
=> [1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f2 13.2s
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f24 0.1s
=> => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2 446B / 446B 0.7s
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e6 3.64MB / 3.64MB 2.6s
=> => sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d 1.26MB / 1.26MB 2.6s
=> => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4eфеа 40.01MB / 40.01MB 11.2s
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5 0.3s
=> => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4eфеа161b2b5a 1.6s
=> => extracting sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702 0.1s
=> => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e 0.0s
=> [2/5] WORKDIR /app                                                 0.2s
=> [3/5] COPY package*.json ./                                         0.1s
=> [4/5] RUN npm install --production                            80.0s
=> [5/5] COPY . .                                                 1.0s
=> exporting to image                                              15.8s
=> => exporting layers                                             14.4s
=> => exporting manifest sha256:8e96de2292a6eb67673394b3ada68a79a981cf70668aac27ad29 0.0s
=> => exporting config sha256:68c63e5efa3e816f8f2991622d22576e97baa8cd6e2ebffd2db313 0.0s
=> => exporting attestation manifest sha256:3e07d9377a0c1438bf402a1ede005e701a621ec2 0.1s
=> => exporting manifest list sha256:e4fb8f882dc9517bba70bdb415a09201d0ce6dadf54350a 0.0s
=> => naming to docker.io/uddhav345/node-hello:1.0                0.0s
=> => unpacking to docker.io/uddhav345/node-hello:1.0              1.1s
PS D:\devops_docker\node-docker>
```

## 5. Run Docker Container

Executed the following command to run the container:

```
docker run -d --name node-hello -p 3000:3000 <YOUR_DOCKERHUB_USERNAME>/node-hello:1.0
```



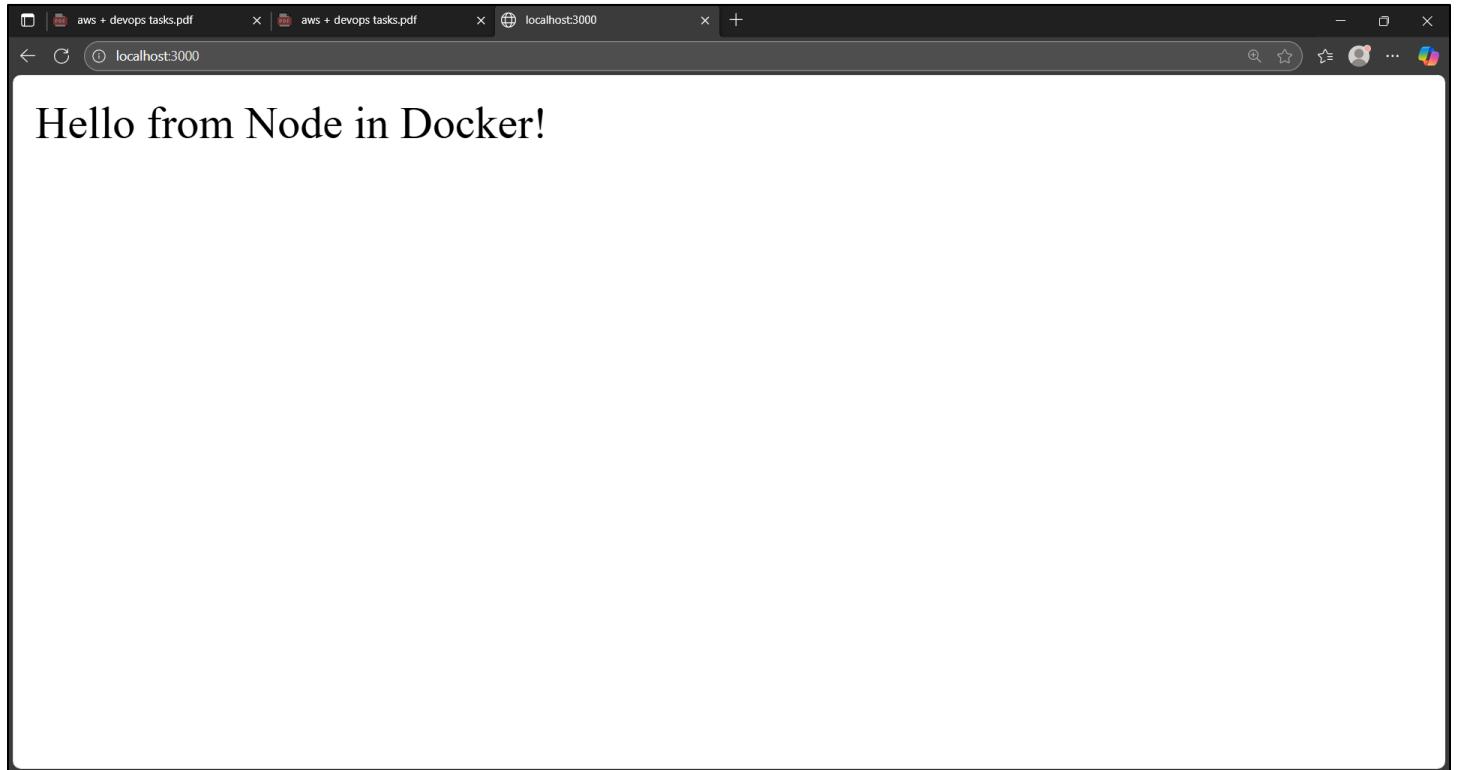
```
PS D:\devops_docker\node-docker> docker run -d --name node-hello -p 3000:3000 uddhav345/node-hello:1.0
22c8ffb573f994f138663878b6b0122b3a66e67d0f3e06b8c33f28b4e436a1cb
PS D:\devops_docker\node-docker>
```

Verified with the following command:

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
22c8ffb573f9	uddhav345/node-hello:1.0	"docker-entrypoint.s..."	19 seconds ago	Up 18 seconds	0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
node-hello					

The application was then tested in the browser at <http://localhost:3000> and successfully displayed:  
'Hello from Node in Docker!'



## 6. Push Image to Docker Hub

The image was pushed to Docker Hub with the following command:

```
docker push <YOUR_DOCKERHUB_USERNAME>/node-hello:1.0
```

```
PS D:\devops_docker\node-docker> docker push uddhav345/node-hello:1.0
The push refers to repository [docker.io/uddhav345/node-hello]
9d82dee22622: Pushed
48876fad9838: Pushed
9cd23d69112c: Pushed
25ff2da83641: Pushed
2890141656f5: Pushed
4b0c85d4e85e: Pushed
1e5a4c89cee5: Pushed
dd71dde834b5: Pushed
f18232174bc9: Pushed
1.0: digest: sha256:e4fb8f882dc9517bba70bdb415a09201d0ce6dadf54350a6db8ebb758e2b69e3 size: 856
PS D:\devops_docker\node-docker> |
```

The image is now available on Docker Hub at: <https://hub.docker.com/r/uddhav345/node-hello>  
And image also show in Docker Desktop.

A screenshot of a web browser displaying the Docker Hub interface. The URL in the address bar is [hub.docker.com/repositories/uddhav345](https://hub.docker.com/repositories/uddhav345). The page shows the 'My Hub' section for the user 'uddhav345'. On the left, there is a sidebar with options like 'Repositories', 'Collaborations', 'Settings', 'Default privacy', 'Notifications', 'Billing', 'Usage', 'Pulls', and 'Storage'. The main area is titled 'Repositories' and shows a single repository: 'uddhav345/node-hello'. This repository was last pushed 27 minutes ago, contains an image, and is public. A 'Create a repository' button is visible in the top right of the main content area.

## DOCKER DESKTOP

A screenshot of the Docker Desktop application window. The sidebar on the left includes 'Ask Gordon (BETA)', 'Containers' (which is selected), 'Images', 'Volumes', 'Builds', 'Models', 'MCP Toolkit (BETA)', 'Docker Hub', 'Docker Scout', and 'Extensions'. The main pane is titled 'Containers' and shows one container named 'node-hello' with the ID '22c8fffb573f9'. The container is using 0% CPU and 16.12MB memory. It was last started 32 minutes ago. The status bar at the bottom shows 'Selected 1 of 1', 'Engine running', 'RAM: 0.84 GB CPU: 0.00%', 'Disk: 1.41 GB used (limit 1006.85 GB)', and 'Terminal'.

## 7. Conclusion

The Docker Basics task was successfully completed. A simple Node.js application was containerized using Docker, the container was built and tested locally, and the image was pushed to Docker Hub under the username 'uddhav345'.