



# National Textile University

## Department of Computer Science

Subject:

Operating System

---

Submitted to:

Sir Nasir Mehmood

---

Submitted by:

Akasha Fatima

---

Reg. number:

23-NTU-CS-FL-1132

---

Semester:

5<sup>th</sup> - A

---

# **LAB\_12**

## **Task\_01:**

```
PS C:\Users\abc> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
node lts-alpine3.23 c921b97d4b74 33 hours ago 226MB
nginx trixie fb01117203ff 9 days ago 225MB
postgres 15.15-trixie 697ff7029514 10 days ago 627MB
ubuntu resolute-20251130 901617b8bedb 2 weeks ago 133MB
ubuntu latest c35e29c94501 2 months ago 117MB
PS C:\Users\abc> docker run -it --name test-container -v my-volume:/home/ubuntu/my-new-volume ubuntu:resolute-20251130 bash
root@cdb0ae9e819c:/# cd home/ubuntu/
root@cdb0ae9e819c:/home/ubuntu# ls
my-new-volume
root@cdb0ae9e819c:/home/ubuntu# cd my-new-volume/
root@cdb0ae9e819c:/home/ubuntu/my-new-volume# echo "Hello World" > File.txt
root@cdb0ae9e819c:/home/ubuntu/my-new-volume# cat File.txt
Hello World
root@cdb0ae9e819c:/home/ubuntu/my-new-volume#
```

## **Task\_02:**

```
PS C:\Users\abc> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
node lts-alpine3.23 c921b97d4b74 33 hours ago 226MB
nginx trixie fb01117203ff 9 days ago 225MB
postgres 15.15-trixie 697ff7029514 10 days ago 627MB
ubuntu resolute-20251130 901617b8bedb 2 weeks ago 133MB
ubuntu latest c35e29c94501 2 months ago 117MB
PS C:\Users\abc> docker run -it --name test-container -v D:\NTU_LIFE\5th_Smester\Operating_System\LABS\LAB_12_1132:/home/ ubuntu:resolute-20251130 bash
root@e8f4862dda56:/#
```

## **Task\_03:**

```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. Re-enable it, run 'Import-Module PSReadLine'.
PS D:\NTU_LIFE\5th_Smester\Docker> Import-Module PSReadLine
PS D:\NTU_LIFE\5th_Smester\Docker> npm init -y
Wrote to D:\NTU_LIFE\5th_Smester\Docker\package.json:

{
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "license": "ISC",
  "type": "commonjs"
}

PS D:\NTU_LIFE\5th_Smester\Docker>
PS D:\NTU_LIFE\5th_Smester\Docker> npm i express

● added 65 packages, and audited 66 packages in 15s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
PS D:\NTU_LIFE\5th_Smester\Docker> docker images
node          lts-alpine3.23   c921b97d4b74   34 hours ago  226MB
nginx         trixie          fb01117203ff   9 days ago   225MB
● postgres     15.15-trixie   697ff7029514   10 days ago  627MB
ubuntu        resolute-20251130 901617bbbedb   2 weeks ago   133MB
ubuntu        latest          c35e29c94501   2 months ago  117MB
PS D:\NTU_LIFE\5th_Smester\Docker> docker build -t my-first-server .
[+] Building 1.2s (1/1) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 31B
ERROR: failed to build: failed to solve: the Dockerfile cannot be empty
PS D:\NTU_LIFE\5th_Smester\Docker> docker build -t my-first-server .
[+] Building 0.4s (1/1) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 31B
PS D:\NTU_LIFE\5th_Smester\Docker> ls

●

Directory: D:\NTU_LIFE\5th_Smester\Docker

Mode                LastWriteTime       Length Name
----                -----          ----
```

The screenshot shows the VS Code interface with the Docker extension active. The terminal tab is selected, displaying the output of a Docker build process. The log shows the following steps:

```
[+] Building 43.9s (9/9) FINISHED
  => [internal] load build definition from Dockerfile
  => transferring dockerfile: 192B
  => [internal] load metadata for docker.io/library/node:lts-alpine3.23
  => [internal] load .dockerignore
  => resolve docker.io/library/node:lts-alpine3.23@sha256:c921b97d4b74f51744057454b306b418cf69386e73b8100559189605f6955b8
  3s
  => [internal] load build context
  5s
  => -> transferring context: 558B
  1s
  => [2/4] COPY index.js .
  2s
  => [3/4] COPY package.json .
  2s
  => [4/4] RUN npm i
  5s
  => exporting to image
  7s
  => -> exporting layers
  2s
  => -> exporting manifest sha256:34094ed793424ae6132011bd98c7724d6bc5a068d77090469643822efc674c25
  1s
  => -> exporting config sha256:3b31b49f11e795acf11e5505cff0b0c9a3e1f817db89ca747337f2337ad9d974
  1s
  => -> exporting attestation manifest sha256:7cdaf2f25125856bbfe91121f57eb6c182c021f9e2dc4e620aaa00060c2fd64ee
  1s
  => -> exporting manifest list sha256:409d8240399ce97697952a87afbcb11148aea7ba41f01125b135589d86c01ba8
  1s
  => -> naming to docker.io/library/my-first-server:latest
  0s
  => -> unpacking to docker.io/library/my-first-server:latest
  6s
```

The status bar at the bottom indicates the file is 'Dockerfile' and the time is '3:50 PM 12/19/2025'.

## Task\_04: Running a Simple Command on Web Request: Index.js

The screenshot shows the VS Code interface with the Docker extension active. The terminal tab is selected, displaying the content of the 'index.js' file. The code is as follows:

```
import express from 'express';
const app = express();
function handler(request, response) {
  response.send('Hello World!');
}
app.get('/', handler);
app.listen(3000);
```

The status bar at the bottom indicates the file is 'index.js' and the time is '4:00 PM 12/19/2025'.

## Dockerfile:

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists files and folders, including a 'DOCKER' folder containing 'node\_modules', 'Public', 'index.html', 'Dockerfile', 'index.js', 'package-lock.json', and 'package.json'. The main editor area displays a Dockerfile with the following content:

```
#Base Image
FROM node:lts-alpine3.23
# Copy files
COPY index.js .
COPY package.json .
# intall package
RUN npm i
CMD ["node", "index.js"]
```

The status bar at the bottom shows the file is 'Dockerfile', has 21 lines, 4 spaces, and is saved in UTF-8 with CRLF line endings. The date and time are 12/19/2025 4:01 PM.

## Task\_05: Passing An HTML File:

### DockerFile:

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists files and folders, including a 'DOCKER' folder containing 'node\_modules', 'Public', 'index.html', 'Dockerfile', 'index.js', 'package-lock.json', and 'package.json'. The main editor area displays a Dockerfile with the following content:

```
#Base Image
FROM node:lts-alpine3.23
# Copy files
COPY index.js .
COPY Public ./Public
# intall package
RUN npm i
CMD ["node", "index.js"]
```

A small 'Windows PowerShell' window is overlaid on the status bar, showing the command 'cd .\Public' being run. The status bar at the bottom shows the file is 'Dockerfile', has 13 lines, 4 spaces, and is saved in UTF-8 with CRLF line endings. The date and time are 12/19/2025 4:29 PM.

### Index.html:

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists files in a 'DOCKER' folder: package.json, index.js, Dockerfile, index.html, package-lock.json, and package.json. The 'index.html' file is selected and open in the main editor area. The code in index.html is:

```
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>Hello World</title>
7   </head>
8   <body>
9     <h1>Hello From the Docker Container</h1>
10    <p>This is a simple web page served using Express.js in a Docker container.</p>
11
12 </body>
13 </html>
```

The status bar at the bottom shows: Ln 10, Col 84, Spaces: 4, UTF-8, CRLF, HTML, Go Live, 4:29 PM, 12/19/2025.

## Index.JS:

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists files in a 'DOCKER' folder: package.json, index.js, Dockerfile, index.html, package-lock.json, and package.json. The 'index.js' file is selected and open in the main editor area. The code in index.js is:

```
1 import express from 'express';
2
3 const app = express();
4
5 app.use(express.static('Public'));
6
7 app.listen([3000])
```

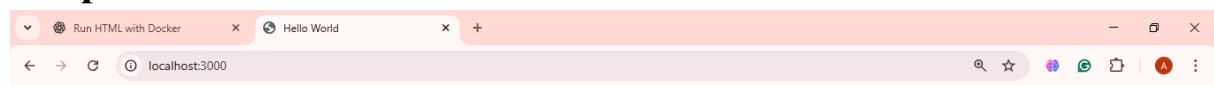
The status bar at the bottom shows: Ln 7, Col 17, Spaces: 4, UTF-8, CRLF, JavaScript, Go Live, 4:29 PM, 12/19/2025.

## Container Execution:

The screenshot shows the VS Code interface with the Docker extension open. The Explorer sidebar on the left lists files and folders related to the project, including `Dockerfile`, `index.js`, `package-lock.json`, and `package.json`. The Terminal tab is active, displaying the command `docker build -t my-first-server .` and its output. The output shows the build process for the `Dockerfile`, including copying files, building the image, and exporting it. The status bar at the bottom indicates the current file is `index.js`.

```
PS D:\WITU_LIFE\5th_Semester\ Docker> docker build -t my-first-server .
[+] Building 29.6s (10/10) FINISHED
-> [internal] load build definition from Dockerfile
-> transferring dockerfile: 214B
-> [internal] load metadata for docker.io/library/node:lts-alpine3.23
-> [internal] load .dockerignore
-> transferring context: 2B
-> resolve docker.io/library/node:lts-alpine3.23@sha256:c921b97d4b74f51744057454b306b418cf693865e73b8100559189605f6955b8
-> [internal] load build context
-> transferring context: 124B
-> CACHED [2/5] COPY index.js .
-> CACHED [3/5] COPY Public ./Public
-> CACHED [4/5] COPY package.json .
-> CACHED [5/5] RUN npm i
-> exporting to image
-> exporting layers
-> exporting manifest sha256:6aaef493d0d60784e64cb26a89fc2bc63d455f3b49fe66949f9d5266528362f5
-> exporting config sha256:597096757e4fca4a7afecc932f1a12899193b256ff427edf5d5e6be49a0e8648bd7
-> exporting attestation manifest sha256:7de0518599a49f13830398ce3ea34757a0f005612e49d10e7bd8e7287cf680
-> exporting manifest list sha256:aeee0784aa89a76fc0be3ae82ac0c63bf96c88790f82341d33918a340229e41
-> naming to docker.io/library/my-first-server:latest
-> unpacking to docker.io/library/my-first-server:latest
PS D:\WITU_LIFE\5th_Semester\ Docker> docker run -d -p 3000:3000 my-first-server
● 4b6310c7695adfe8c8bcccd750829586e36c8f96de567d3cac5914a3d4d63
PS D:\WITU_LIFE\5th_Semester\ Docker>
```

## Output:



# Hello From the Docker Container

This is a simple web page served using Express.js in a Docker container.

