Name: Rohit Harawade

Roll No: A09

ID: TU4F2122009

EXPERIMENT NO: 7

# Aim: To create Docker containers of any web application image.

**Theory:**

# What is Docker?

Docker is a virtual machine, but unlike virtual machines that create a completely separate operating system, Docker allows the applications to use the Linux kernel of the same machine on which it is installed and by taking this benefit, it can make the applications ready to ship to other machines that are running the same Linux OS with somewhat different configurations. In this way, application size is reduced significantly and at the time of shipping, application performance also gets improved.

# Docker container:

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

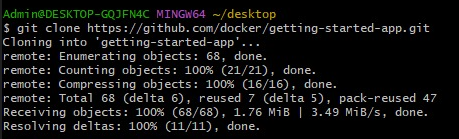
# Docker containers that run on Docker Engine:

* Standard: Docker created the industry standard for containers, so they could be portable anywhere
* Lightweight: Containers share the machine’s OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs
* Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

**Steps to create docker containers of any web application image:**

1. Clone the repository using the following command:

git clone <https://github.com/docker/getting-started-app.git>



1. View the contents of the cloned repository. You should see the following files and sub-directories.

├── getting-started-app/

│ ├── package.json

│ ├── README.md

│ ├── spec/

│ ├── src/

│ └── yarn.lock

In the Windows Command Prompt, run the following commands.

1. Make sure you're in the getting-started-app directory. Replace \path\to\getting-started-app with the path to your getting-started-app directory.

cd \path\to\getting-started-app



1. Create an empty file named Dockerfile.

type nul > Dockerfile

1. Using a text editor or code editor, add the following contents to the Dockerfile:

# syntax=docker/dockerfile:1

FROM node:18-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]

EXPOSE 3000

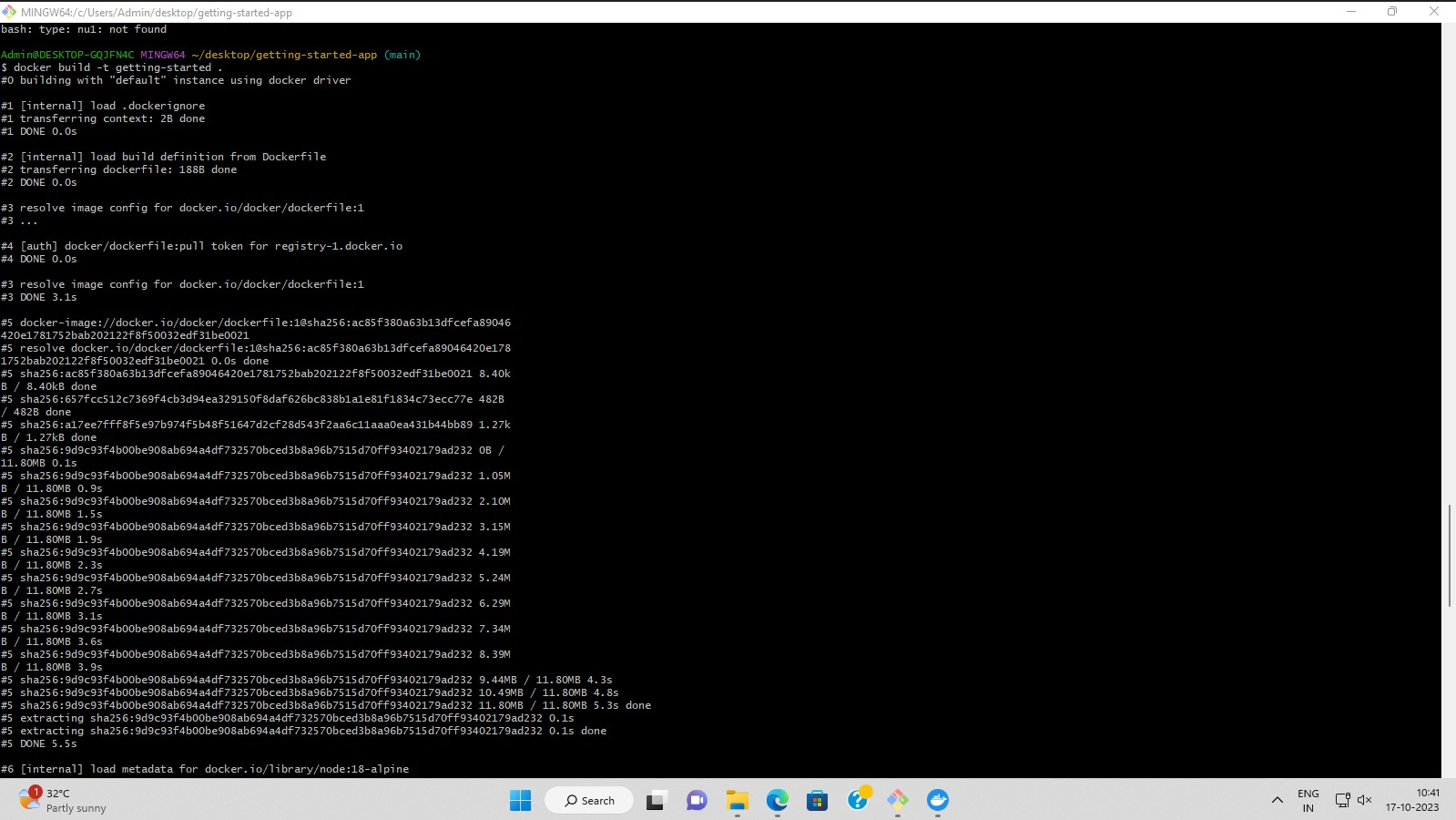
1. Build the image using the following commands:

In the terminal, make sure you're in the getting-started-app directory. Replace /path/to/getting-started-app with the path to your getting-started-app directory.

cd /path/to/getting-started-app

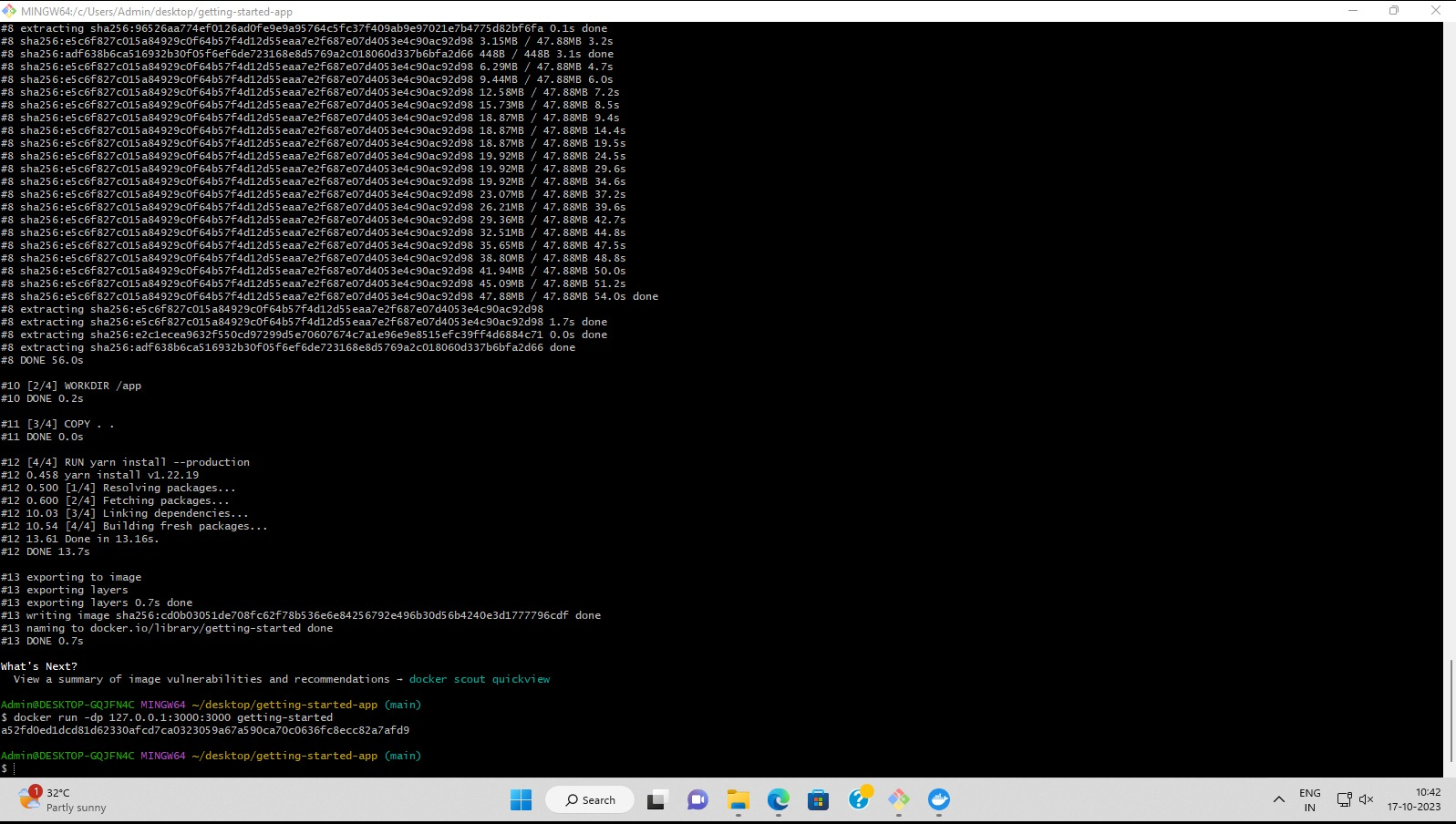
Build the image:

docker build -t getting-started .

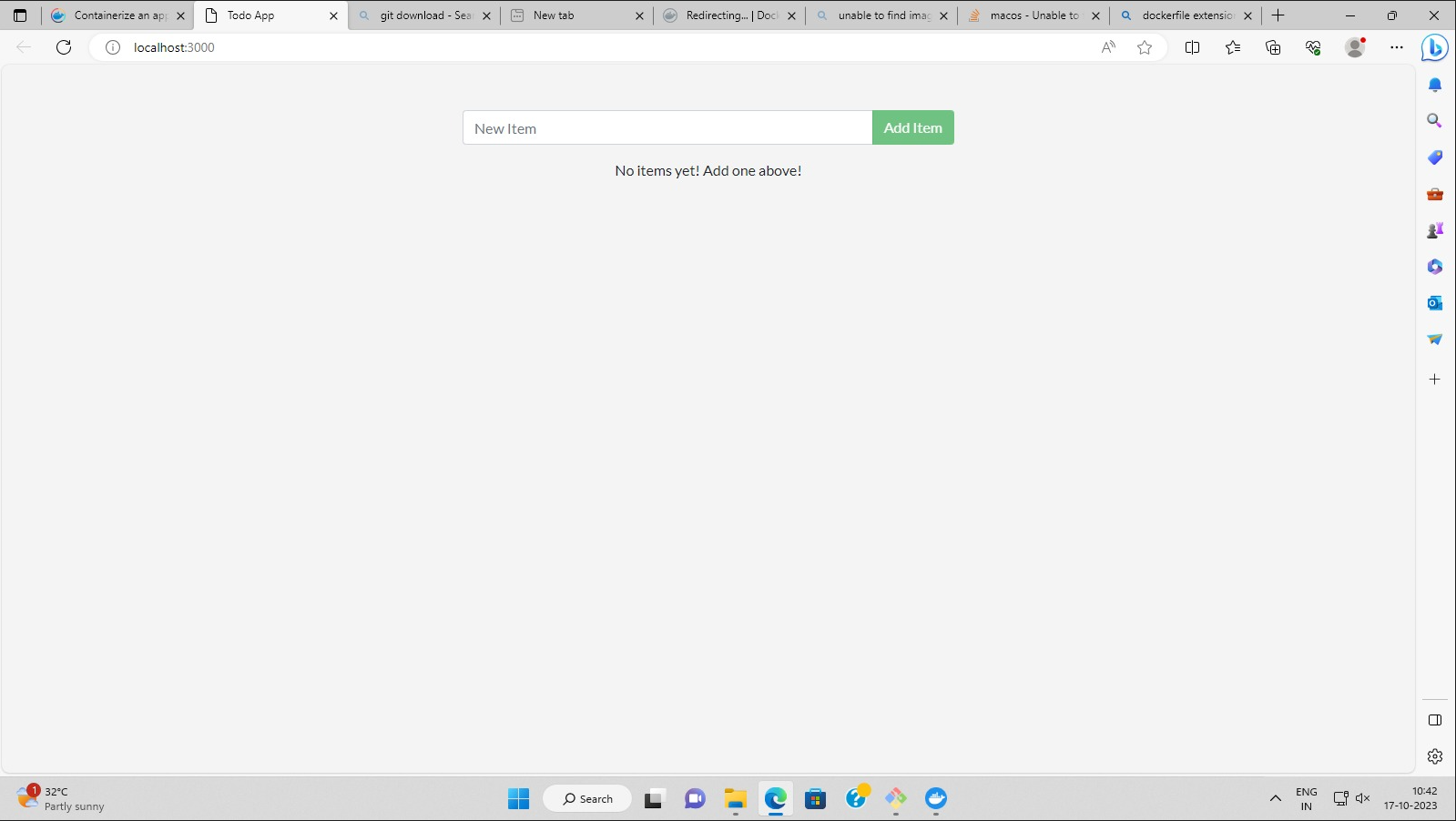


1. Run your container using the docker run command and specify the name of the image you just created:

docker run -dp 127.0.0.1:3000:3000 getting-started



After a few seconds, open your web browser to [http://localhost:3000](http://localhost:3000/). You should see your app.



**Conclusion:** Thus we have studied about how to build an image for a sample web application using Dockerfile.