Name → Siddharth Sutar

PRN → 22010517

ROLL NO → 323054

SUBJECT → CC

Batch → C3

Assignment 7

AIM

Deploying node js / python app using docker

Theory

Docker is an open source platform that enables developers to build, deploy, run, update and manage containers

Containers are standardized, executable components that combine application source code with the operating system libraries and dependencies required to run that code in any environment. Docker allows you to separate your applications from your infrastructure so you can deliver software quickly and reliably. Docker also provides a comprehensive end to end platform that includes UIs, CLIs, APIs and security tools for developing, shipping and running applications

There are many benefits of using Docker, such as

Return on investment and cost savings: Docker can help reduce infrastructure costs by allowing you to run more applications on the same hardware, and by simplifying the deployment and maintenance processes

Consistency and reliability: Docker ensures that your applications run the same way in any environment, whether it is a developer's laptop, a testing server, or a production cloud platform. Docker also helps you avoid compatibility issues and dependency conflicts by packaging your applications with all the necessary components

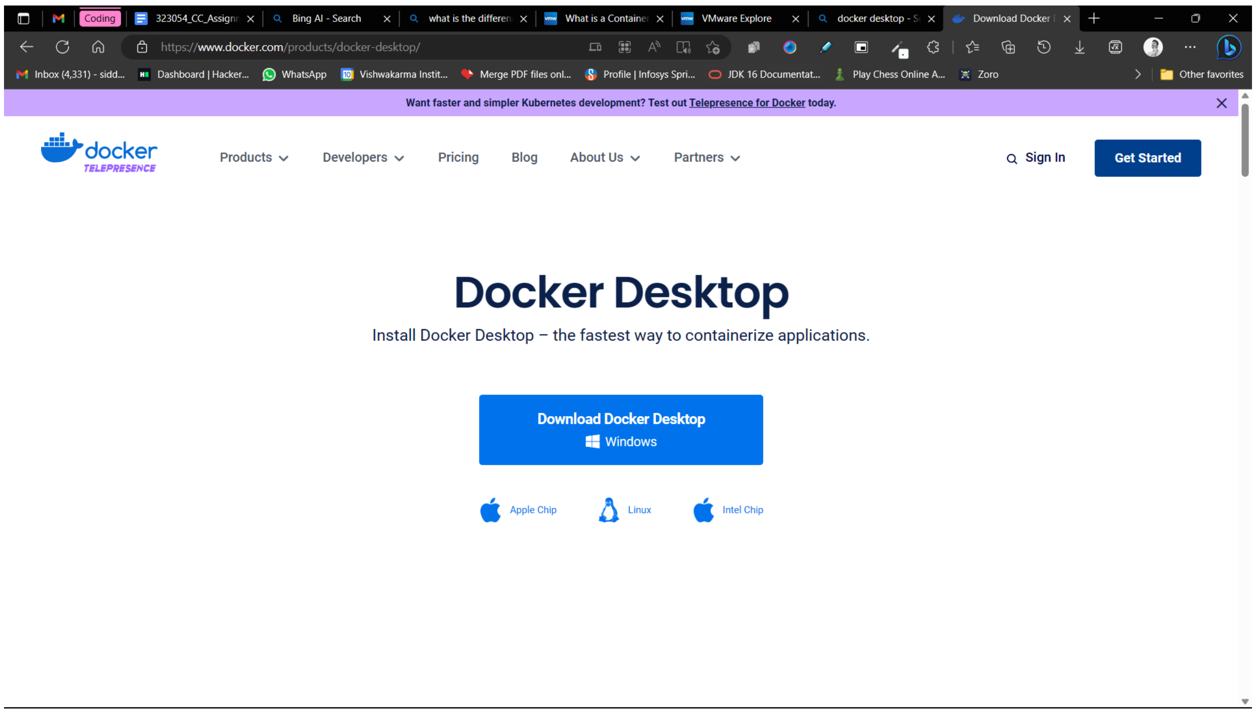
Rapid deployment and scalability: Docker allows you to quickly start and stop containers, which makes it easy to deploy new versions of your applications or scale them up or down according to the demand

Isolation and security: Docker provides a layer of isolation between your applications and the underlying system, which reduces the risk of malicious attacks or accidental errors

Flexibility and portability: Docker supports a wide range of languages, frameworks, and platforms, which gives you more freedom and choice in developing your applications

Installation

Just install the [docker app] then you are good to go!

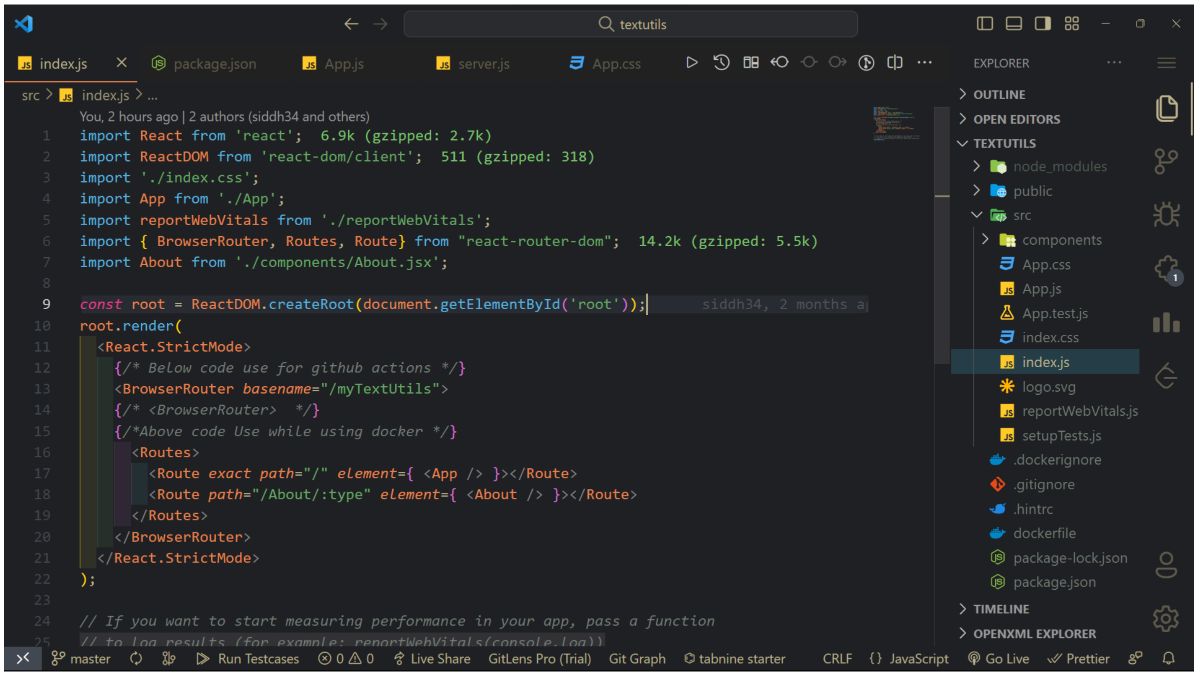

https://www.docker.com/products/docker-desktop/
Want faster and simpler Kubernetes development? Test out Telepresence for Docker today.
docker TELEPRESENCE
Products
Developers
Pricing
Blog
About Us
Partners
Sign In
Get Started
Docker Desktop
Install Docker Desktop – the fastest way to containerize applications.
Download Docker Desktop
Windows
Apple Chip
Linux
Intel Chip

Note:

Make sure that you keep the docker desktop app open all the time while doing this assignment

Steps

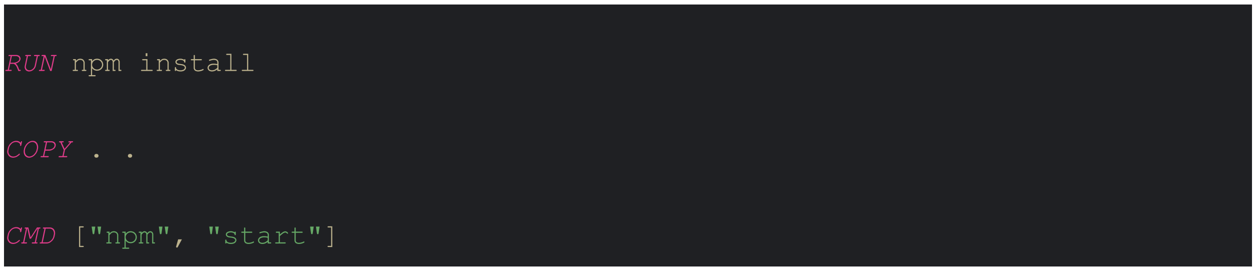
1. You have to have node js application ready. Here I will be using a already done product we will be just seeing it's deployment in docker container



```javascript
import React from 'react';  6.9k (gzipped: 2.7k)
import ReactDOM from 'react-dom/client';  511 (gzipped: 318)
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { BrowserRouter, Routes, Route} from "react-router-dom";  14.2k (gzipped: 5.5k)
import About from './components/About.jsx';

const root = ReactDOM.createRoot(document.getElementById('root'));        siddh34, 2 months a
root.render(
  <React.StrictMode>
    {/* Below code use for github actions */}
    <BrowserRouter basename="/myTextUtils">
    {/* <BrowserRouter>  */}
    {/*Above code Use while using docker */}
      <Routes>
        <Route exact path="/" element={ <App /> }></Route>
        <Route path="/About/:type" element={ <About /> }></Route>
      </Routes>
    </BrowserRouter>
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
```

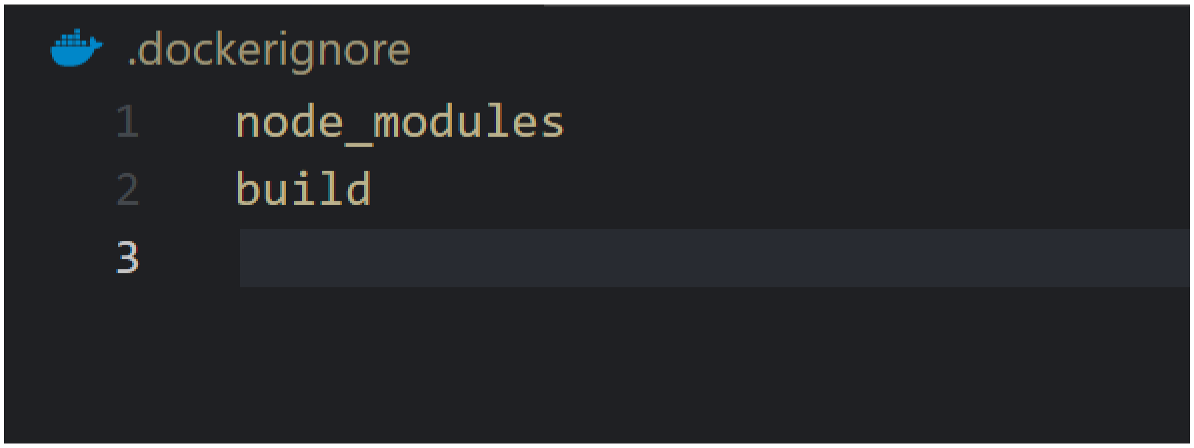
2. Write a docker file

```dockerfile
# The command FROM node:18 Grabs node js version 18

# WORKDIR is the directory where we will be save our application files in docker container

# RUN npm install means it installs node js

# COPY use to copy specific files into /app

# CMD ["npm", "start"]: runs the npm start command

FROM node:18

WORKDIR /app

COPY ["package.json", "package-lock.json*", "./"]
```



```
RUN npm install

COPY . .

CMD ["npm", "start"]
```

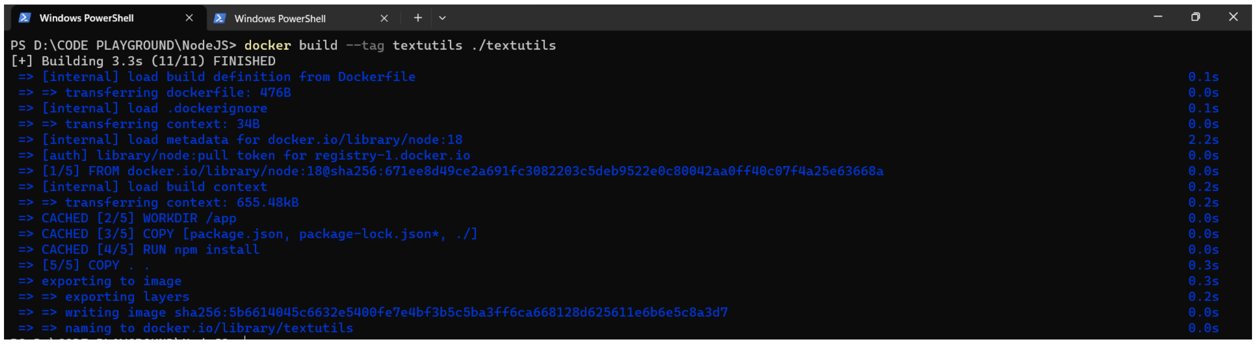
3. Write a dockerignore file (It is similar to gitignore)



```
.dockerignore
node_modules
build

```

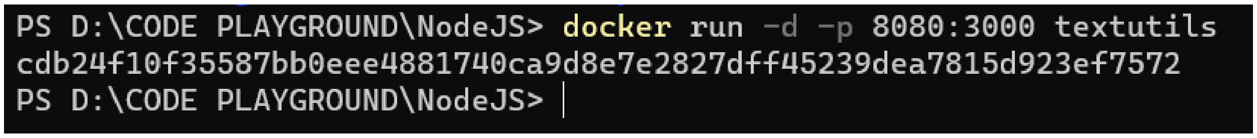
4. Open terminal where you have kept your project file

    a. Use docker build command along with tag



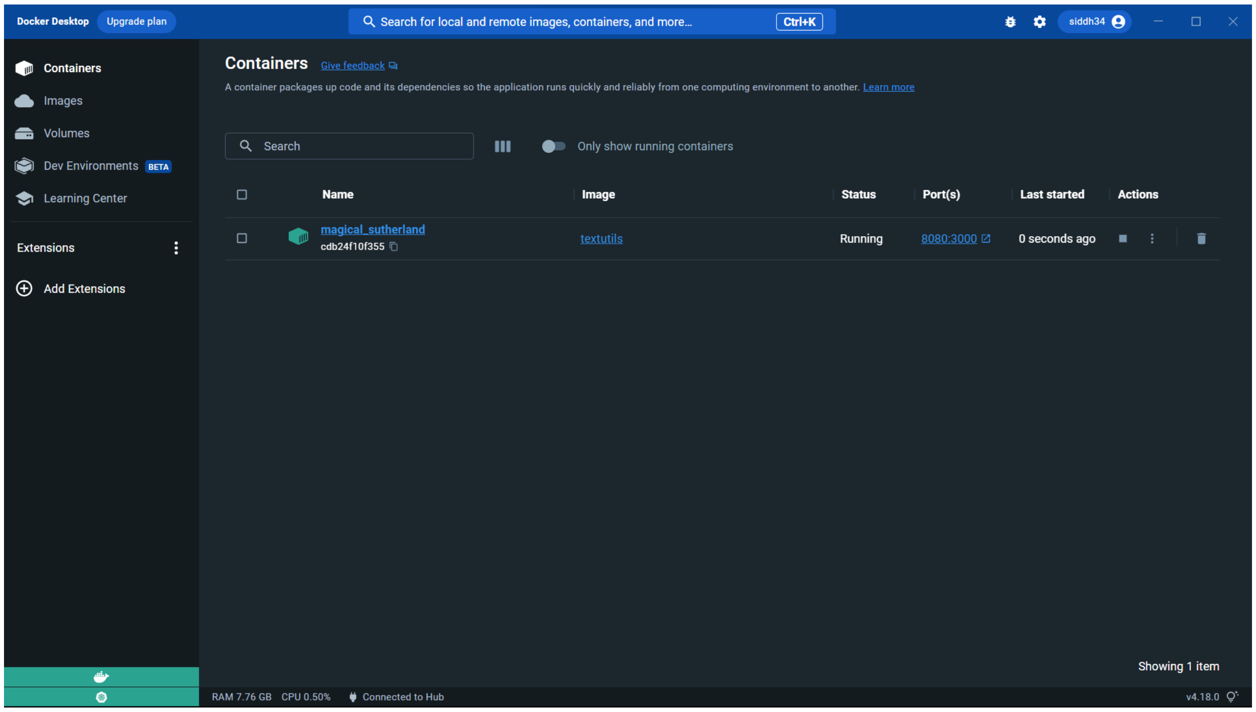
```
PS D:\CODE PLAYGROUND\NodeJS> docker build --tag textutils ./textutils
[+] Building 3.3s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                       0.1s
 => => transferring dockerfile: 476B                                                                       0.0s
 => [internal] load .dockerignore                                                                          0.1s
 => => transferring context: 34B                                                                           0.0s
 => [internal] load metadata for docker.io/library/node:18                                                 2.2s
 => [auth] library/node:pull token for registry-1.docker.io                                                0.0s
 => [1/5] FROM docker.io/library/node:18@sha256:671ee8d49ce2a691fc3082203c5deb9522e0c80042aa0ff40c07f4a25e63668a  0.0s
 => [internal] load build context                                                                          0.2s
 => => transferring context: 655.48kB                                                                      0.2s
 => CACHED [2/5] WORKDIR /app                                                                              0.0s
 => CACHED [3/5] COPY [package.json, package-lock.json*, ./]                                               0.0s
 => CACHED [4/5] RUN npm install                                                                           0.0s
 => [5/5] COPY . .                                                                                         0.3s
 => exporting to image                                                                                     0.3s
 => => exporting layers                                                                                    0.2s
 => => writing image sha256:5b6614045c6632e5400fe7e4bf3b5c5ba3ff6ca668128d625611e6b6e5c8a3d7               0.0s
 => => naming to docker.io/library/textutils                                                               0.0s
```

5. Use docker run command specify the ports as 8080:3000 (3000 is localhost as application is running on 8080 & displayed on localhost)



```
PS D:\CODE PLAYGROUND\NodeJS> docker run -d -p 8080:3000 textutils
cdb24f10f35587bb0eee4881740ca9d8e7e2827dff45239dea7815d923ef7572
PS D:\CODE PLAYGROUND\NodeJS>
```

Now let's see whether our container is running or not, firstly open the docker desktop app



Containers

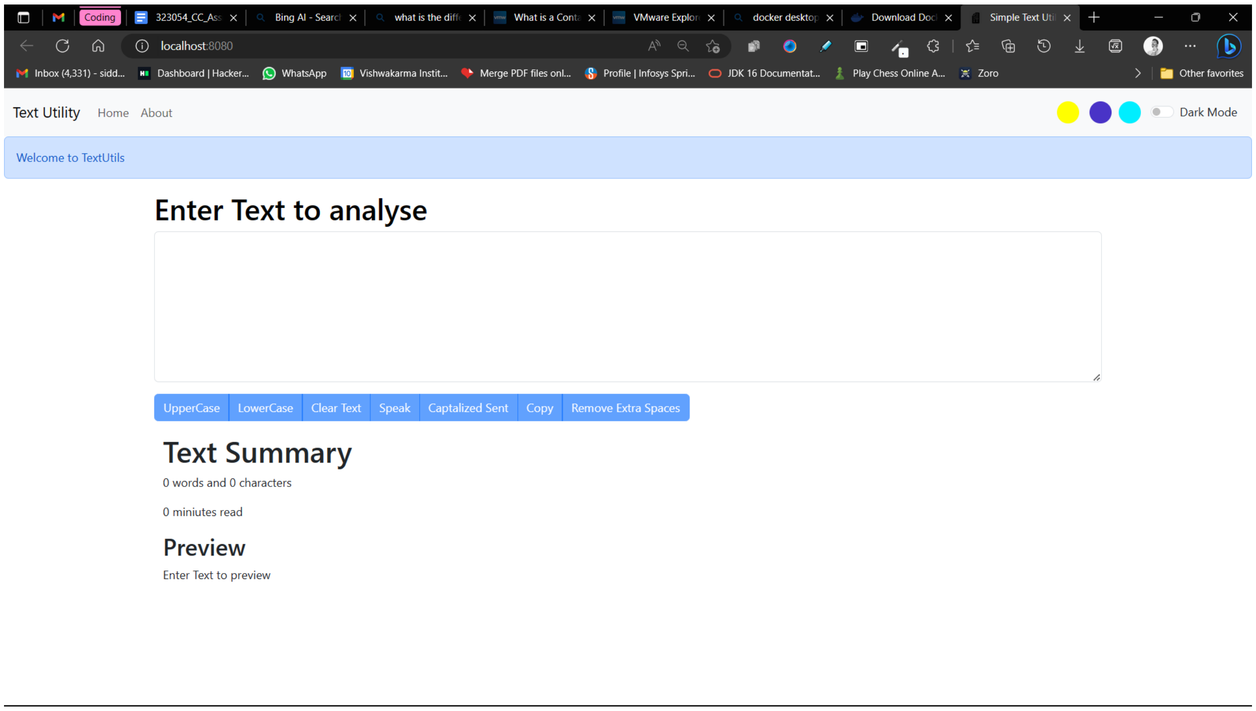
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Name	Image	Status	Port(s)	Last started	Actions
magical_sutherland cdb24f10f355	textutils	Running	8080:3000	0 seconds ago	

Showing 1 item

RAM 7.76 GB CPU 0.50% Connected to Hub v4.18.0

Click on ports here! Your application is running in a container



Welcome to TextUtils

Enter Text to analyse

Text Summary

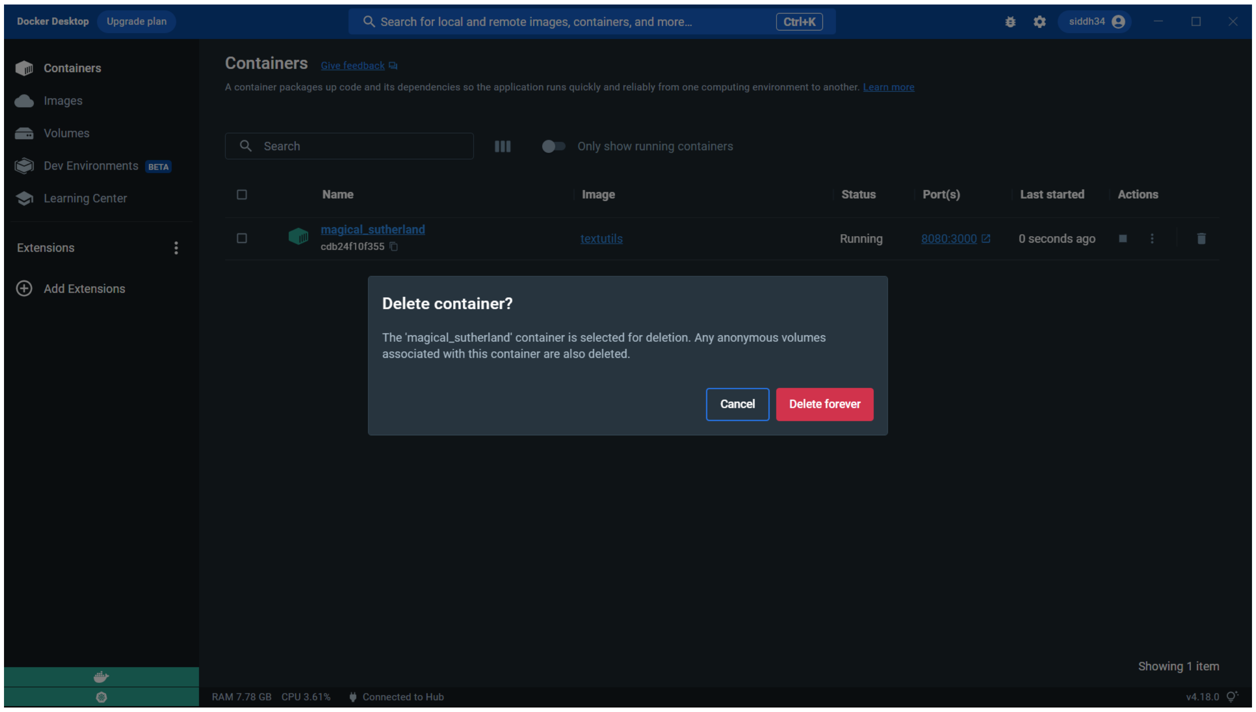
0 words and 0 characters

0 miniutes read

Preview

Enter Text to preview

Delete the container after using it!



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Name	Image	Status	Port(s)	Last started	Actions
magical_sutherland cdb24f10f355	textutils	Running	8080:3000	0 seconds ago	

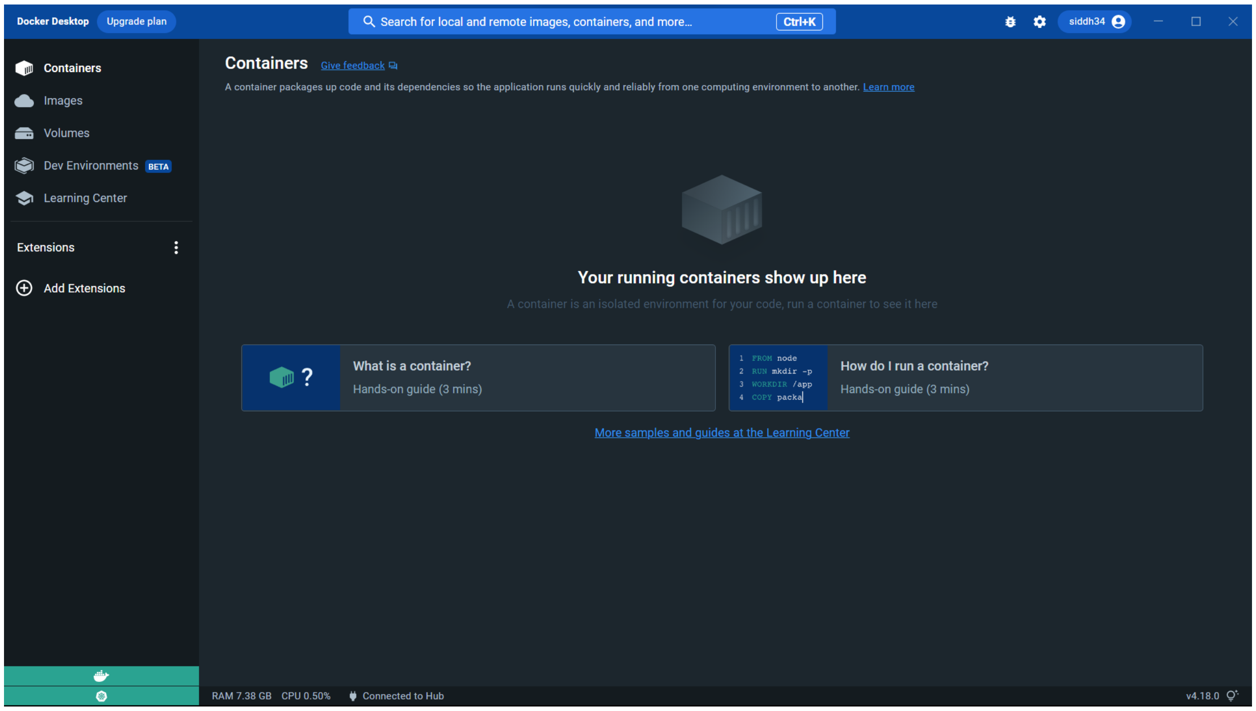
Delete container?

The 'magical_sutherland' container is selected for deletion. Any anonymous volumes associated with this container are also deleted.

Showing 1 item

RAM 7.78 GB CPU 3.61% Connected to Hub v4.18.0

Done!



Containers Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Your running containers show up here

A container is an isolated environment for your code, run a container to see it here

What is a container?
Hands-on guide (3 mins)

```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

How do I run a container?
Hands-on guide (3 mins)

More samples and guides at the Learning Center

RAM 7.38 GB CPU 0.50% Connected to Hub v4.18.0

Conclusion

Docker is understood