Setting Up Angular in Docker

A step-by-step guide for setting up Docker and using Angular inside Docker.

Rebekah Williams

RPW 335: Instruction Writing

Winter 2022

Table of Contents

Introduction

Welcome! This manual aims to explain and walk through how to use Angular in Docker. Inside you'll find the set-up process, how to update the software, and troubleshooting notes. Angular is a framework designed to make web development easier to use and Docker can make that even easier. If you've been hard coding websites in HTML and CSS and are ready for something easier you've come to the right place. In this manual you will learn how to set up and "kick start" your Angular environment in Docker. **This manual does not contain information on how to build websites in** Angular, but it will get your first application running.

It is recommended users of this manual have experience with front end development and general programming knowledge before they attempt to use this software. If you are completely new to programming, there will be concepts and terms that may confuse you, but you are welcome to walk through it and see if it fits what you are looking for.

What is Docker?

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.

Recommended Prerequisites:

1. HTML
2. JavaScript
3. Experience with Atom or visual Studio Code
4. Experience using the Command Prompt/Terminal

If this is the first software you've installed on your computer, that's okay, this is a great place to start. The process is simple, but you must read the directions carefully and correctly.

This user manual is intended to get you started but does not hold all the details for what you can accomplish with Angular in Docker.

Anytime in this manual that you see text with this formatting `cd .\Desktop\My-docker-app\`, that means that the text is being inputted into your terminal. The color and formatting might seem odd, but this is standard practice for text in coding documents.

Troubleshooting note…

Installing software and using that software is one of the most difficult challenges you can embark on. Throughout the installation process you are sure to encounter issues this manual does not address. These might be specific to your laptop, settings and more. If there is nothing in the troubleshooting section for your issue you can always Google it.

Setting Up Docker

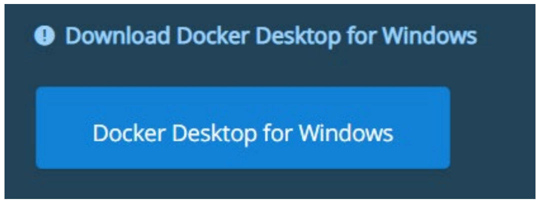
Installing docker is simple and easy to do. Follow the steps below to get it set up.

Step 1

Navigate to https://docs.docker.com/desktop/windows/install/ on a browser.

Step 2

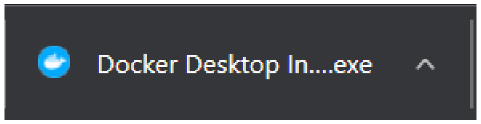
Once the page loads, click on the blue button labeled **Docker Desktop for Windows.**


Download Docker Desktop for Windows
Docker Desktop for Windows

Step 3

When it's done downloading, it will appear like the image below at the bottom of your browser.

Click on the download



What Browser should I use?

Chrome is recommended for this manual, but you can use whatever browser you are most comfortable with. This manual was tested using Chrome so if you encounter errors particular to that browser they will not be included in the manual.

Step 4

A message will appear saying,

“Do you want to allow this app to make changes to your device?”

Click YES on the box!

Step 5

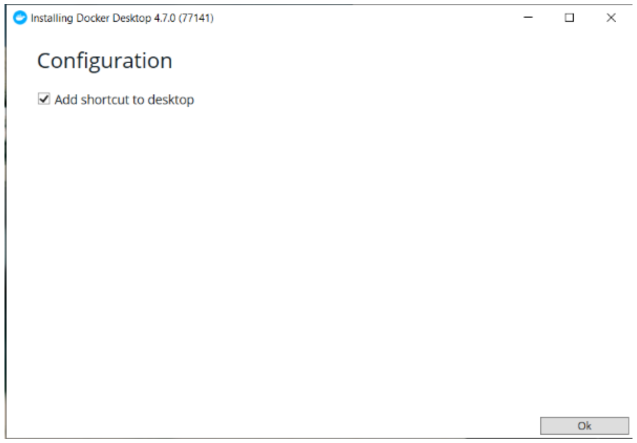
Wait for it to download onto your device. **(This will take several minutes)**

Error – Step 5

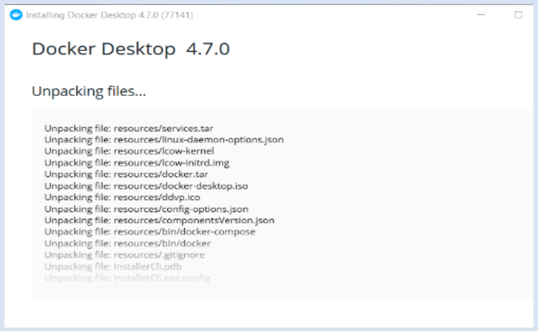
If you receive any errors during this step, go here.

Step 6

When this window pops up, you can check yes and select ok if you want to add a shortcut to the desktop. (We recommend this)



During the download process your screen will look like this…

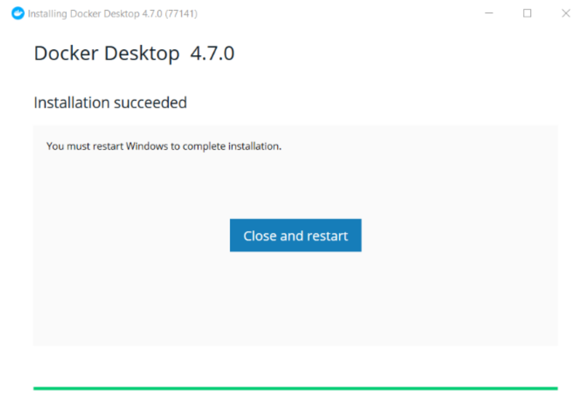


```
Unpacking files...

Unpacking file: resources/services.tar
Unpacking file: resources/linux-daemon-options.json
Unpacking file: resources/lcow-kernel
Unpacking file: resources/lcow-initrd.img
Unpacking file: resources/docker.tar
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
```

Step 6

Once the download is complete it will say Docker Desktop and the version number.

The blue button in the center of the screen will say close and restart. Click the **close and restart** button



Installing Docker Desktop 4.7.0 (77141)

Docker Desktop 4.7.0

Installation succeeded

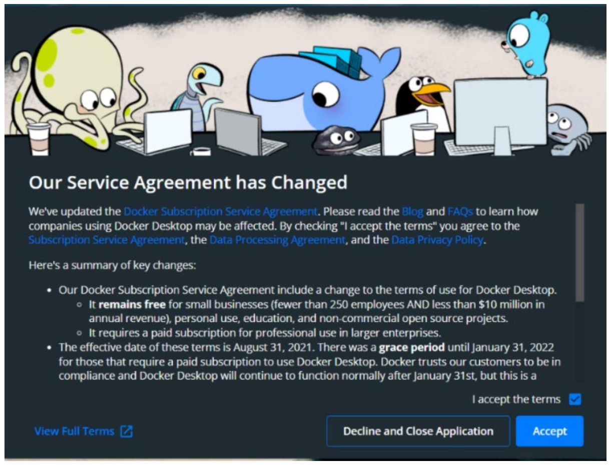
You must restart Windows to complete installation.

Step 7

Once the computer restarts, open Docker. (You'll see a message appear)

Step 8

Click the blue **Accept** button.



Our Service Agreement has Changed

We've updated the Docker Subscription Service Agreement. Please read the Blog and FAQs to learn how companies using Docker Desktop may be affected. By checking "I accept the terms" you agree to the Subscription Service Agreement, the Data Processing Agreement, and the Data Privacy Policy.

Here's a summary of key changes:

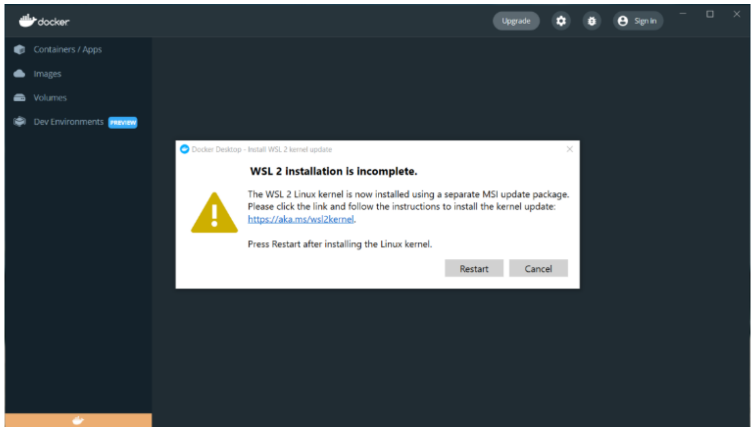
- Our Docker Subscription Service Agreement include a change to the terms of use for Docker Desktop.
  - It **remains free** for small businesses (fewer than 250 employees AND less than $10 million in annual revenue), personal use, education, and non-commercial open source projects.
  - It requires a paid subscription for professional use in larger enterprises.
- The effective date of these terms is August 31, 2021. There was a **grace period** until January 31, 2022 for those that require a paid subscription to use Docker Desktop. Docker trusts our customers to be in compliance and Docker Desktop will continue to function normally after January 31st, but this is a

I accept the terms ✓

Step 9

Click the **blue link** in the WSL 2 Installation is incomplete (Once it pops up).

This will open a link in the web browsers to a list of steps.



WSL 2 installation is incomplete.

The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: https://aka.ms/wsl2kernel.

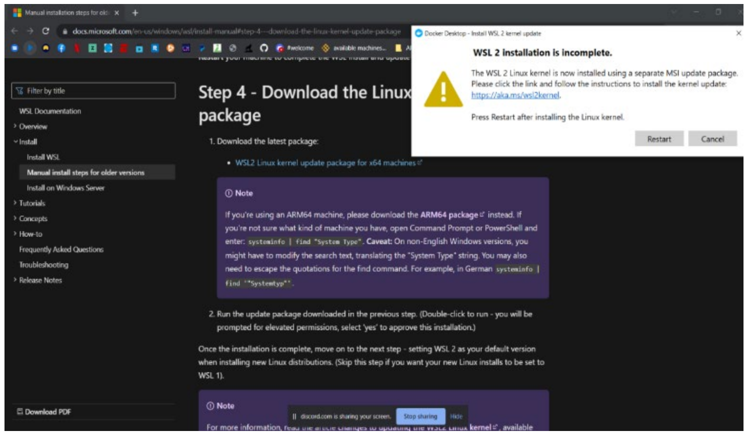
Press Restart after installing the Linux kernel.

Didn't receive the WSL Installation is incomplete message?

That means you've already got the other software and permissions Docker needs. If that's the case, you can skip ahead to step 22.

Step 10

Once the link opens, scroll down to step 4.



Step 4 - Download the Linux package

1. Download the latest package:
   - WSL2 Linux kernel update package for x64 machines

Note

If you're using an ARM64 machine, please download the **ARM64 package** instead. If you're not sure what kind of machine you have, open Command Prompt or PowerShell and enter: `systeminfo | find "System Type"`. **Caveat:** On non-English Windows versions, you might have to modify the search text, translating the "System Type" string. You may also need to escape the quotations for the find command. For example, in German `systeminfo | find '"Systemtyp"'`.

2. Run the update package downloaded in the previous step. (Double-click to run - you will be prompted for elevated permissions, select 'yes' to approve this installation.)

Once the installation is complete, move on to the next step - setting WSL 2 as your default version when installing new Linux distributions. (Skip this step if you want your new Linux installs to be set to WSL 1).

WSL 2 installation is incomplete.

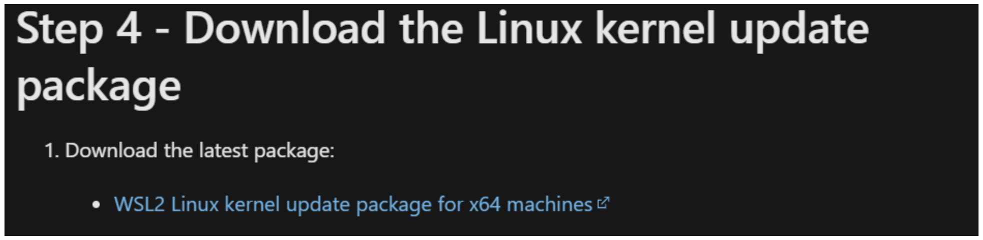
The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: https://aka.ms/wsl2kernel.

Press Restart after installing the Linux kernel.

Manual installation steps for old
docs.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package
#welcome
available machines...
Filter by title
WSL Documentation
Overview
Install
Install WSL
Manual install steps for older versions
Install on Windows Server
Tutorials
Concepts
How-to
Frequently Asked Questions
Troubleshooting
Release Notes
Download PDF
Note
discord.com is sharing your screen.
Stop sharing
Hide
Docker Desktop - Install WSL 2 kernel update
Restart
Cancel

Step 11

Click on the WSL2 Linux Kernel update package for x64 machines link or click this link to download the software.



Step 4 - Download the Linux kernel update package

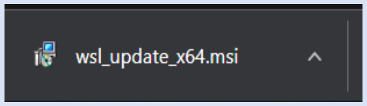
1. Download the latest package:

- WSL2 Linux kernel update package for x64 machines

Step 12

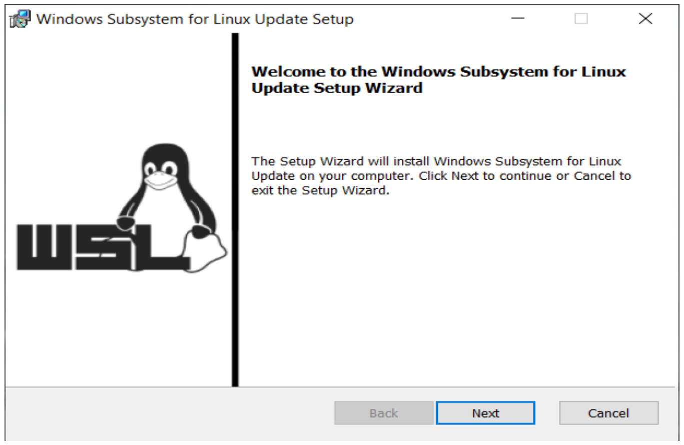
Click on the **wsl_update_x64.msi** that appeared at the bottom of your browser.

The download will appear at the bottom of your screen like the image below.



Step 13

Click next on the penguin popup.



Windows Subsystem for Linux Update Setup

Welcome to the Windows Subsystem for Linux Update Setup Wizard

The Setup Wizard will install Windows Subsystem for Linux Update on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

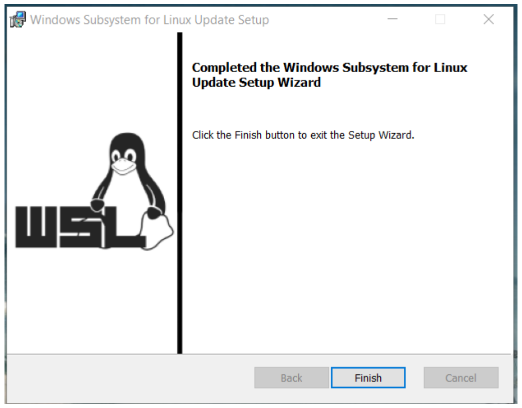
Step 14

Click YES when this message appears as a pop-up on the screen.

"Do you want to allow this app to make changes to your device?"

Step 15

Click **finish** when the finish button appears on the penguin image.



Windows Subsystem for Linux Update Setup

Completed the Windows Subsystem for Linux Update Setup Wizard

Click the Finish button to exit the Setup Wizard.

Step 16

Search for your Microsoft store on your computer.

Step 17

In the Microsoft store search bar type, *Ubuntu 20.04.4* and hit enter.

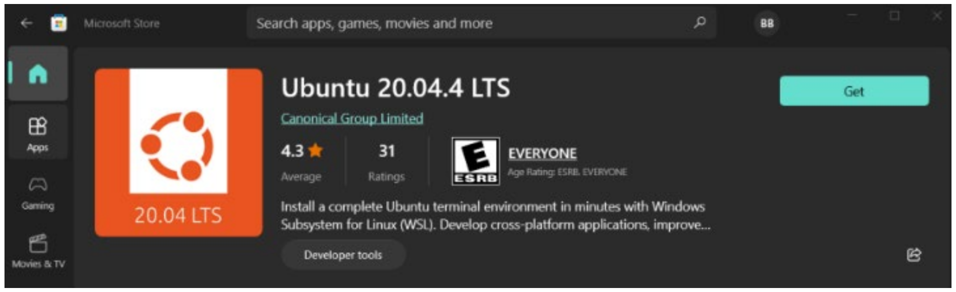
Step 18

Click on the **Ubuntu 20.04.4 LTS** app.

The Microsoft store should look something like this!

Step 19

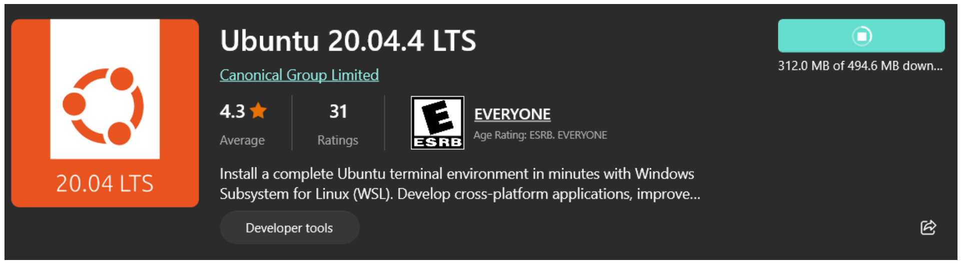
Once it opens, click the **get** button on Ubuntu.



Ubuntu 20.04.4 LTS
Canonical Group Limited
4.3 Average · 31 Ratings · EVERYONE, Age Rating: ESRB. EVERYONE
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform applications, improve...

Step 20

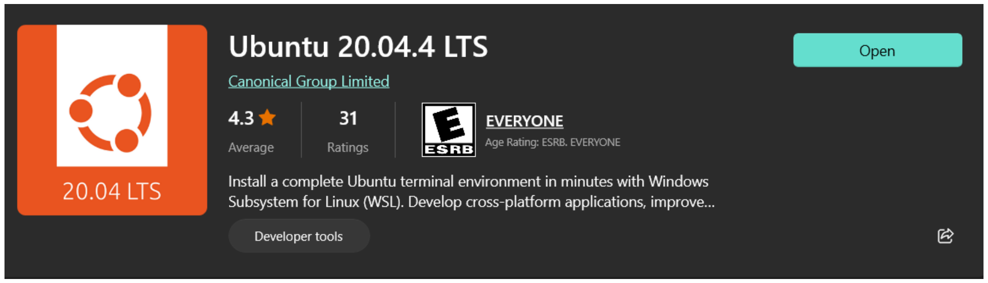
Waiting for it to download. It will look like the image below.



Ubuntu 20.04.4 LTS
Canonical Group Limited
4.3 Average · 31 Ratings · EVERYONE, Age Rating: ESRB. EVERYONE
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform applications, improve...

Step 21

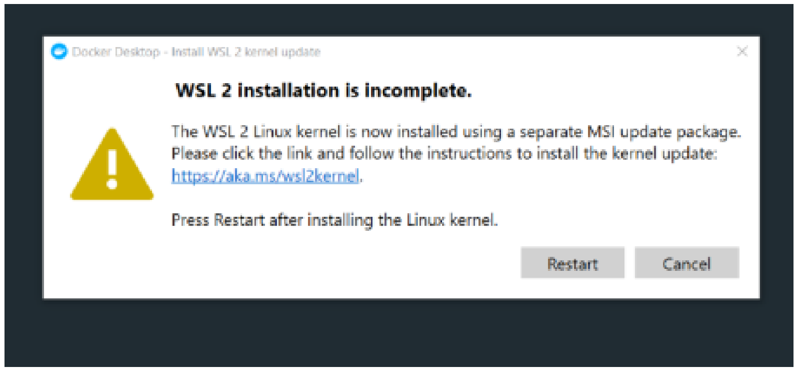
Click the **open** button on Ubuntu.



Ubuntu 20.04.4 LTS
Canonical Group Limited
4.3 Average · 31 Ratings · EVERYONE, Age Rating: ESRB. EVERYONE
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform applications, improve...

Step 22

Click on WSL2 **restart** button, which will restart Docker.



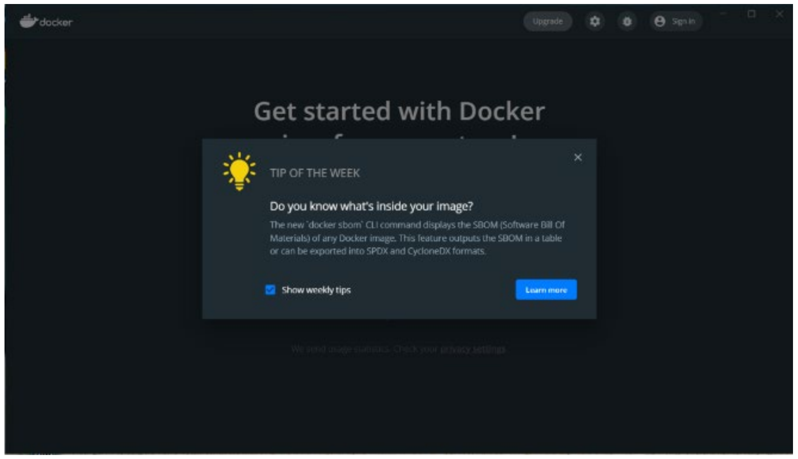
WSL 2 installation is incomplete.

The WSL 2 Linux kernel is now installed using a separate MSI update package.
Please click the link and follow the instructions to install the kernel update:
https://aka.ms/wsl2kernel.

Press Restart after installing the Linux kernel.

Step 23

Once Docker reopens, click the **X** if the tip of the week window appears.

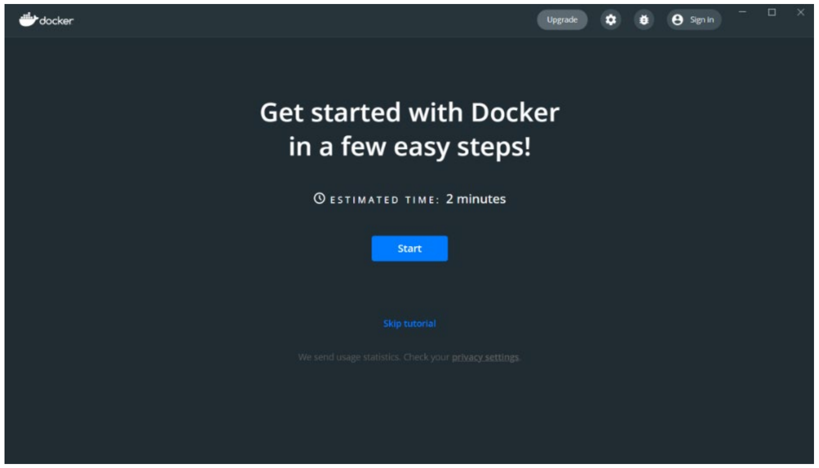


Do you know what's inside your image?

The new `docker sbom` CLI command displays the SBOM (Software Bill Of Materials) of any Docker image. This feature outputs the SBOM in a table or can be exported into SPDX and CycloneDX formats.

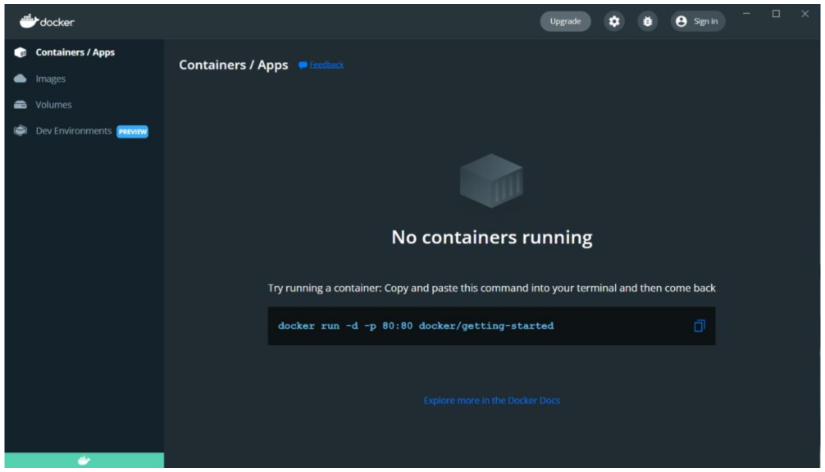
Step 24

Click **skip tutorial**.


docker
Upgrade
Sign in
Get started with Docker in a few easy steps!
ESTIMATED TIME: 2 minutes
Start
Skip tutorial
We send usage statistics. Check your privacy settings.

Step 25

Docker will open and will look like the image below.



Try running a container: Copy and paste this command into your terminal and then come back

```
docker run -d -p 80:80 docker/getting-started
```

Want to take the tutorial?

You are welcome to take the tutorial, but the tutorial is not covered in this manual.

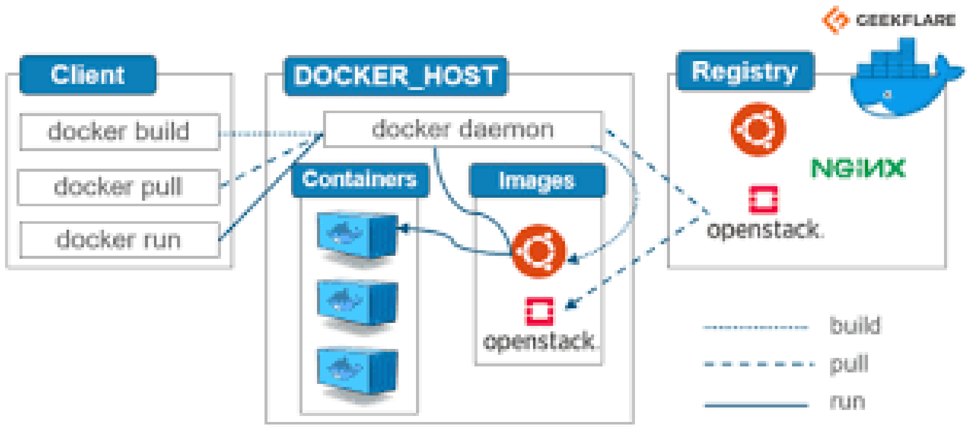
Making a Docker Container

Building a container in docker is easy, but understanding the process is a bit trickier. The image below shows what's happening.

We are the Client in the image, running the commands in the terminal. When we run "docker build", "docker pull" our device looks at the docker host and pulls from the registry to get the image to host for container.

What is a Docker Container?

A Docker container is a running instance of a Docker image.



Now that we understand the process, let's try it out together.

Hello World Image

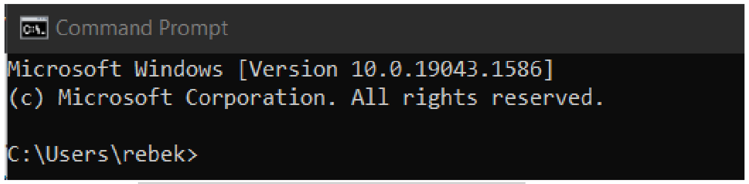
Step 1

Open your terminal app.

Computers have different one's but mine looks like this. If you've never used the terminal before you can find it by searching "Command Prompt" in the search bar of your laptop.

Inside your command prompt it should look something like this.

Where it says, “rebek” it should say the name that you registered with your computer.



```
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rebek>
```

If we type `docker run hello-world` in our terminal it won’t find the image locally, but it will pull from the Docker library. Let’s try it!

Step 2

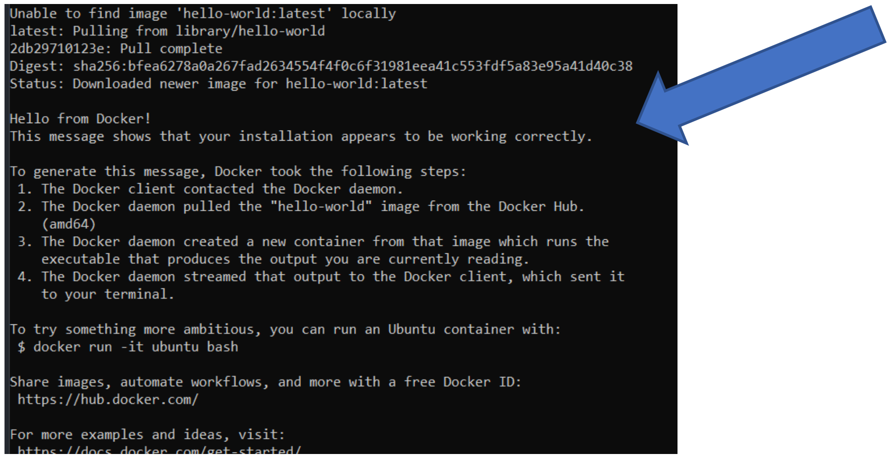
Inside your terminal type the command `docker run hello-world`.

You’ll get a message saying, Unable to find image 'hello-world:latest' locally.

Then another message saying, latest: Pulling from library/hello-world.

Since you don’t have the image, it will pull from the library. After its done it should look like the image below in your terminal. Look for the line

“This message shows that your installation appears to be working correctly to verify that its been installed correctly.”



```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:bfea6278a0a267fad2634554f4f0c6f31981eea41c553fdf5a83e95a41d40c38
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Now you have pulled your first image! Let’s make a container.

Angular Container with Nginx Image

In this section we are going to make a Docker website using a nginx image in Docker. There are a lot of complex pieces in this section but essentially, you're making two files that will execute when you run the two terminal commands. These two files tell Docker how to make the image, what volumes to use, what ports to host the website on and more.

What is nginx?

Nginx, stylized as NGIИX, is a web server. It can be used for lots of things but for us it will be what our container is referencing.

Creating Files in Atom or Visual Studio Code

You'll need Atom or Visual Studio Code for this part. If you don't have Atom or Visual Studio Code, go here.

Step 1

On your desktop make a new folder called *My-docker-app*.

Step 2

Open Atom or Visual Studio code on your computer.

Step 3

1. Click file
2. Click add project folder
3. Add the "My-docker-app" folder

Step 4

Create a new file and name it Dockerfile.

Step 5

Copy in the text on the right to your Dockerfile.

```
FROM node:14

# Create app directory
WORKDIR /usr

# Install app dependencies
# A wildcard is used to ensure both package.json AND
package-lock.json are copied
# where available (npm@5+)


RUN npm install -g @angular/cli
RUN ng new angularTest
WORKDIR /usr/angularTest
RUN ng build

EXPOSE 8080
CMD [ "ng", "serve", "--host", "0.0.0.0", "--port",
"8080" ]
```

Step 6

Save your file (CTRL S).

Step 7

Create a new file named docker-compose.yml.

Step 8

Copy in the text on the right to your docker-compose file.

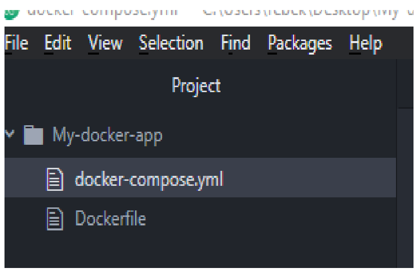
```
# Use root/example as user/password credentials
version: '3.1'

services:

  angularContainer:
    build:
      context: .
    image: angulardockertest
    container_name: angularContainer
    volumes:
      - sharedspace:/usr/angularTest/dist/angular-test
    restart: always
    healthcheck:
      test: "exit 0"
    stdin_open: true
    tty: true
    ports:
      - 5011:8080

  nginxContainer:
     image: nginx
     depends_on:
       angularContainer:
          condition: service_healthy
     container_name: nginxContainer
     volumes:
       - sharedspace:/usr/share/nginx/html
     ports:
        - 5012:80


volumes:
  sharedspace:
```

Step 9

Save your file (CTRL S).

Step 10

Verify that your file set up looks like the image below



Step 11

Now that you've verified your files are correct, let's go run them here.

Creating Files in Notepad

If you don't have Atom or Visual studio code installed, we will make our two files in notepad.

For notepad files…

If you use Notepad to make your files you MUST make sure you copy the file text from the creating files in notepad section otherwise your files will not run properly.

Step 1

On your desktop make a new folder called *My-docker-app*.

Step 2

Open Notepad.

Step 4

Copy in the text on the right to your file.

Step 5

Save the file as Dockerfile with .txt filetype in the My-docker-app folder.

```
FROM node:14

# Create app directory
WORKDIR /usr

# Install app dependencies
# A wildcard is used to ensure both package.json AND
package-lock.json are copied
# where available (npm@5+)


RUN npm install -g @angular/cli
RUN ng new angularTest
WORKDIR /usr/angularTest
RUN ng build

EXPOSE 8080
CMD [ "ng", "serve", "--host", "0.0.0.0", "--port",
"8080" ]
```

Step 6

Open a new notepad file.

Step 7

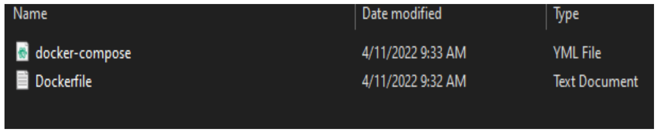
Copy in the text on the right to your new file.

Step 8

Save the file as docker-compmose.yml with the All Files type selected in the My-docker-app folder.

Step 9

Verify your files look like the image below.



Name	Date modified	Type
docker-compose	4/11/2022 9:33 AM	YML File
Dockerfile	4/11/2022 9:32 AM	Text Document

Step 10

Now that you've verified your files are correct, let's go run them here.

```
# Use root/example as user/password credentials
version: '3.1'

services:

  angularContainer:
    build:
      context: .
      dockerfile: Dockerfile.txt
    image: angulardockertest
    container_name: angularContainer
    volumes:
      - sharedspace:/usr/angularTest/dist/angular-test
    restart: always
    healthcheck:
      test: "exit 0"
    stdin_open: true
    tty: true
    ports:
      - 5011:8080

  nginxContainer:
     image: nginx
     depends_on:
       angularContainer:
          condition: service_healthy
     container_name: nginxContainer
     volumes:
       - sharedspace:/usr/share/nginx/html
     ports:
        - 5012:80


volumes:
  sharedspace:
```

Running the files

You've got your files and you're ready to run them. Follow the directions closely and watch for errors as you go.

Step 11

Open the terminal.

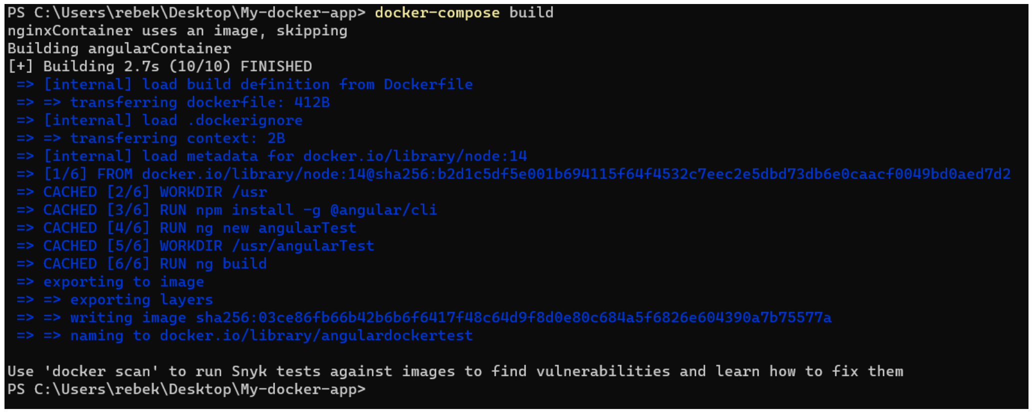
Step 12

Type `cd .\Desktop\My-docker-app\` into the terminal.

Step 13

Type `docker-compose build` into the terminal.

When the screen below appears, you can move to the next step.



```
PS C:\Users\rebek\Desktop\My-docker-app> docker-compose build
nginxContainer uses an image, skipping
Building angularContainer
[+] Building 2.7s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 412B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/node:14
 => [1/6] FROM docker.io/library/node:14@sha256:b2d1c5df5e001b694115f64f4532c7eec2e5dbd73db6e0caacf0049bd0aed7d2
 => CACHED [2/6] WORKDIR /usr
 => CACHED [3/6] RUN npm install -g @angular/cli
 => CACHED [4/6] RUN ng new angularTest
 => CACHED [5/6] WORKDIR /usr/angularTest
 => CACHED [6/6] RUN ng build
 => exporting to image
 => => exporting layers
 => => writing image sha256:03ce86fb66b42b6b6f6417f48c64d9f8d0e80c684a5f6826e604390a7b75577a
 => => naming to docker.io/library/angulardockertest

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS C:\Users\rebek\Desktop\My-docker-app>
```

Step 14

Type `docker-compose up` into the terminal

When you see the Compiled successfully with the green checkmark appear at the bottom of your terminal you know it built the angularContainer.

What is cd?

Cd means to change directory.

Error -Step 13

If you're repeating this tutorial and get an error that says, cannot create container for service nginxContainer: Conflict. The container name "/nginxContainer" is already in use Navigate to the Container In Use section in the Troubleshooting page.

Error -Step 13 or Step 14

If you get a timeout error navigate to the Terminal timeout section in the Troubleshooting page.

```
angularContainer     | ✓ Compiled successfully.
```

When you see the nginxCOntainer come across your screen and see these messages you know it's working!

```
nginxContainer       | 2022/04/10 22:28:37 [notice] 1#1: start worker process 32
nginxContainer       | 2022/04/10 22:28:37 [notice] 1#1: start worker process 33
nginxContainer       | 2022/04/10 22:28:37 [notice] 1#1: start worker process 34
nginxContainer       | 2022/04/10 22:28:37 [notice] 1#1: start worker process 35
```

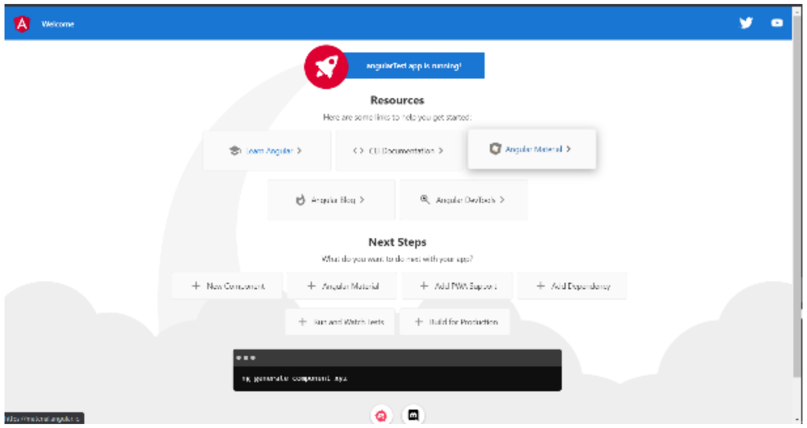
Step 15 - Let's check to see if it worked!

Open your Chrome browser.

Step 16

Type this into your browser window `localhost:5012`

You'll see this image in your browser which means you made your angular site with a nginx image in Docker!



Docker Terminal Commands

This is a list of Docker Terminal commands. If you type `docker -help` into your terminal, you can find this same list. If you want more specific details on one command type `docker COMMAND -help` into the terminal.

Terminal Command	Explanation
`docker attach`	Attach local standard input, output, and error streams to a running container
`docker build`	Build an image from a Dockerfile
`Docker commit`	Create a new image from a container's changes
`docker cp`	Copy files/folders between a container and the local filesystem
`Docker create`	Create a new container
`Docker diff`	Inspect changes to files or directories on a container's filesystem
`Docker events`	Get real time events from the server
`Dokcer exec`	Run a command in a running container
`docker export`	Export a container's filesystem as a tar archive
`Docker history`	Show the history of an image
`docker images`	List images
`docker import`	Import the contents from a tarball to create a filesystem image
`docker info`	Display system-wide information
`docker inspect`	Return low-level information on Docker objects
`docker kill`	Kill one or more running containers
`docker  load`	Load an image from a tar archive or STDIN
`docker login`	Log in to a Docker registry
`docker  logout`	Log out from a Docker registry
`docker logs`	Fetch the logs of a container
`docker unpause`	Unpauses all processes within one or more containers
`docker  port`	List port mappings or a specific mapping for the container
`docker ps`	Shows what containers are running
`docker pull`	Pull an image or a repository from a registry
`docker  push`	Push an image or a repository to a registry
`docker rename`	Rename a container
`docker restart`	Restart one or more containers

What are terminal commands?

Terminal commands allow us to complete tasks on a computer in the command prompt, meaning we don't need a Graphical User Interface.

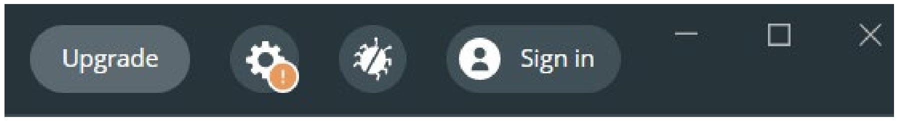
`docker  rm`	Remove one or more containers
`docker rmi`	Remove one or more images
`docker run`	Run a command in a new container
`docker  save`	Save one or more images to a tar archive (streamed to STDOUT by default)
`docker search`	Search the Docker Hub for images
`docker start`	Start one or more stopped containers
`docker  stats`	Display a live stream of container(s) resource usage statistics
`docker stop`	Stops the container
`Docker tag`	tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
`docker top`	Display the running processes of a container
`docker pause`	Pause all processes within one or more containers
`docker update`	Update configuration of one or more containers
`docker version`	Show the Docker version information
`docker wait`	Block until one or more containers stop, then print their exit codes

Maintenance

Maintenance might not be a word you normally hear when it comes to software, but it is very important. When software's get out of date, they get slow, can't run the newest features and eventually stop working all together.

Docker

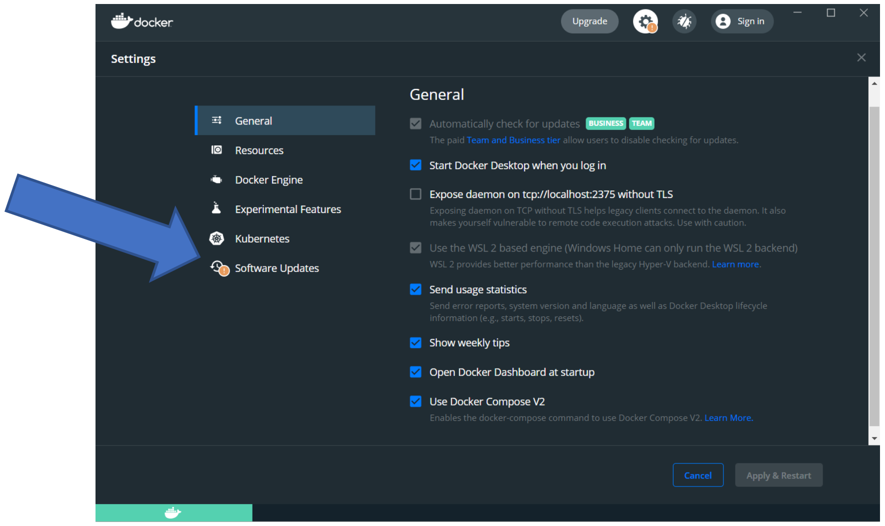
Docker must be checked for updates. If you open docker on your device, you will find the settings tag. There will be an orange exclamation point if there is something important you need to know about.



Step One

Click on the **settings** button.

You will see the settings tab open on the left and the right will contain the general settings. Here the orange exclamation point is located under the software settings tab, so we know we have an update to install.

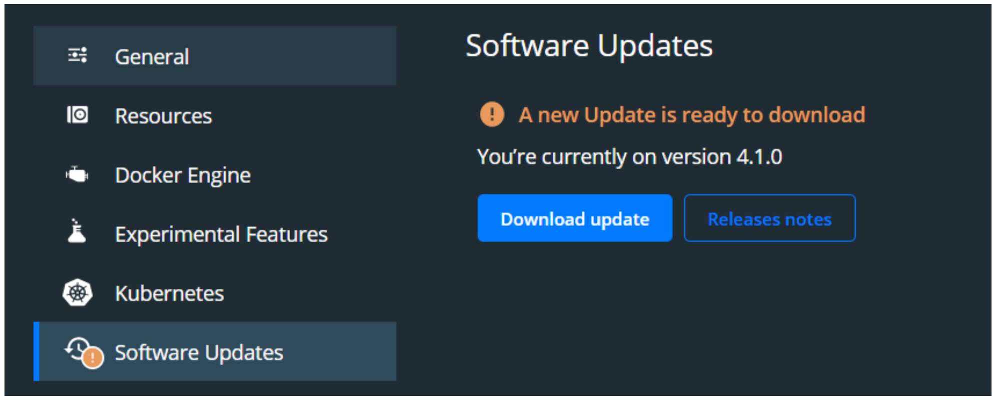


General

- [✓] Automatically check for updates BUSINESS TEAM
  The paid Team and Business tier allow users to disable checking for updates.
- [✓] Start Docker Desktop when you log in
- [ ] Expose daemon on tcp://localhost:2375 without TLS
  Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.
- [✓] Use the WSL 2 based engine (Windows Home can only run the WSL 2 backend)
  WSL 2 provides better performance than the legacy Hyper-V backend. Learn more.
- [✓] Send usage statistics
  Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).
- [✓] Show weekly tips
- [✓] Open Docker Dashboard at startup
- [✓] Use Docker Compose V2
  Enables the docker-compose command to use Docker Compose V2. Learn More.

Step 2

Click on the **Software Updates** icon. Now, you can see what version you are on before updating.



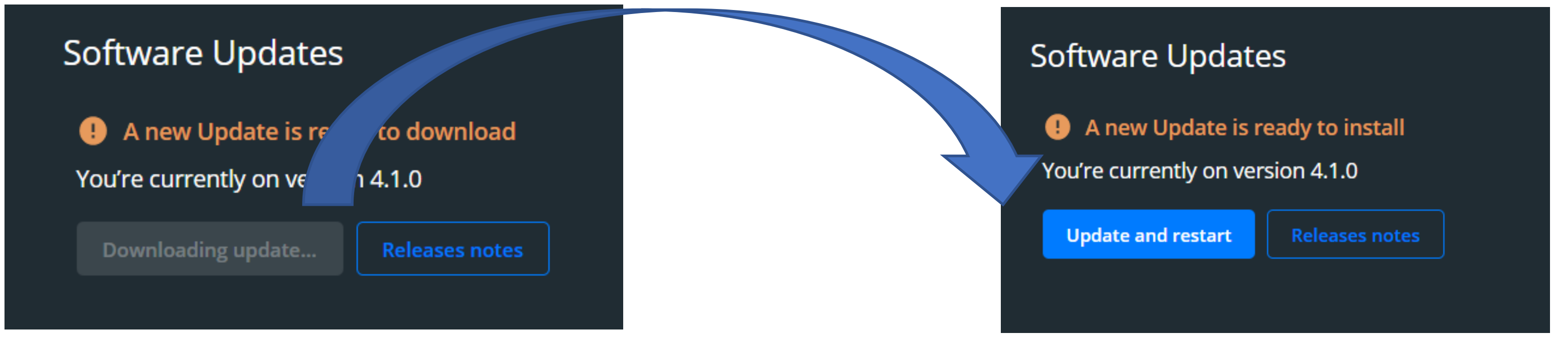
A new Update is ready to download

You're currently on version 4.1.0

Step 3

Click on the **download update** button.

The download update button will change to downloading update, so you know it is working on downloading the update.



A new Update is re to download

You're currently on ve n 4.1.0

A new Update is ready to install

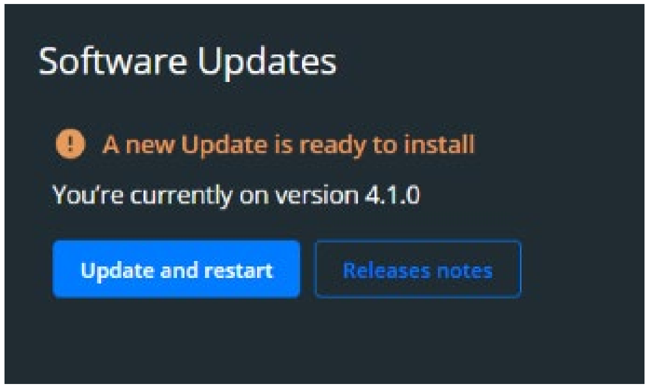
You're currently on version 4.1.0

Step 4

Click the blue **update and restart** button.

Once you do this the application will update and restart.

Now you’ve got the newest version of Docker on your device.


Software Updates

A new Update is ready to install

You’re currently on version 4.1.0

Troubleshooting

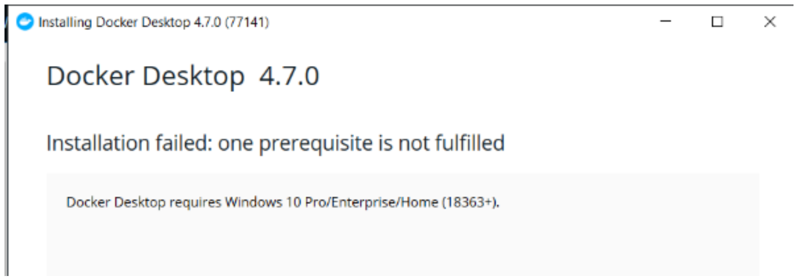
Programming is all about problem-solving. If you've encountered an error or issue that isn't covered in the manual here are some tips and tricks that might help you with the troubleshooting process.

Recommended troubleshooting sources and methodologies

1. **Work Backwards** – Sometimes when you encounter a programming problem it's because of an error you made. (Like a logical or syntax error). Get back to where you were before when the code/software was working. Then redo the steps and make sure after each step that everything is working correctly. If you get back to where the problem was but this time it's working, you probably had a typo or logic error along the way.
2. **Stack Overflow** – Stack overflow is place where people can ask questions and get help. If you're encountering an error, there is a good chance someone else has as well. Look through last posts and see if you find a solution.
3. **Google the exact error** – Sometimes errors are very specific, try googling the error and seeing what you find. Someone somewhere has likely encountered the error before you.

Docker Installation

While installing Docker you might get an error like this. If that occurs navigate to this link on the web and follow the steps to resolve the issues.



Installing Docker Desktop 4.7.0 (77141)

Docker Desktop 4.7.0

Installation failed: one prerequisite is not fulfilled

```
Docker Desktop requires Windows 10 Pro/Enterprise/Home (18363+).
```

Most likely you aren't on the correct version of Windows for the software

Container In Use

If you're receiving this error you are trying to build a container that already has that name, meaning its already in use. You have two options to resolve this, you can delete your previous container in Docker or rename the new one in your files. If you choose to delete, you'll have to delete the information from the containers/App section, images section, and volumes section. If you're still using the same names this manual suggests you'll need to delete these exact items:

1. In the Docker App go to Containers/App open the my-docker-manual and delete the **angularContainer** and **nginxContainer**
2. In the Docker App go to Volumes and delete the **my-docker-app_sharedspace**.
3. In the Docker App go to Images and delete the **angulardockertest** image.

If you don't want to delete these items because you want to keep that container running, you'll need to go into you code and change the names of angularContainer and nginxContainer to something else and make sure the folder you're running your new app from has a different title than the other ones you have made.

Terminal Timeout

When you're running commands in the terminal that are complex (Like docker-compose up) you can encounter a timeout. Usually this means it is taking too long to run the command and so the computer cancels the command. If this happens, try running the command again. Sometimes you'll have to run the command multiple times before it will run it successfully.

If you have a slow internet connection the timeout error is likely to occur.

Usability Report

Rebekah Williams

RPW 335: Instruction Writing

Winter 2022

Introduction

This report outlines the item Docker, the audience, and the manual that was created for it and testing results completed on the manual. This report is meant to act as a summary of the research and usability testing on the manual and device but does not contain extensive detail on every research endeavor.

Item Description

The item itself is a software called Docker that is used for development. To use the software multiple conditions must be met, including having a functional computer with the other necessary elements like Angular to run and host a website. (These must be updated) Additionally for operational specifications, the object needs internet connection and a functional web browser. For manufacturing purposes, the hardware and equipment must be up to handle the newest version of the software, meaning the object must be on a recent/new version of Windows.

Audience Description

The manual was designed for those that have programming and software experience but do not have Angular or Docker on their device. (For example, the glossary does not contain what coding is but would need to explain terms like nginx and Docker). The audience needs to have certain "prerequisites met" like the manual lists in the introduction.

Research Methods

For this project I used an object analysis, an audience analysis, and a task analysis for the manual to test its usability. I have used these methods both before in both Usability Studies, Advanced Problem Solving and Rhetorical Theory for projects and papers.

Method Qualifications

These methods can be used to fully understand the users perspective and to sort and organize feedback from my object analysis (the research and observation questions for the software) into corresponding sections.

Research and Observation Questions

Docker (The Object)

Research Questions

1. What does the object communicate as a stand-alone mechanism? (Signifier)
2. What does the object map out for the user? (Mapping)
3. What type of feedback is received for task completion verification? (Feedback)
4. What does the object have in terms of constraints? (Constraints)
5. Does the mental model relate to the above questions?
6. Does the mental model remain true? (Conceptual Model)

The object communicates the basic buttons and settings of an application on a computer. It maps out these signifiers with the bar on the left side and continues that into the settings section. When tasks are completed, like updating Docker it provides a notification when it needs to be done and reinforces that the change has been made after the update by removing the notification. The object is constraining since the users can't tell from its signifiers all the things it can do. The mental model does not remain true since there is so much more the object can do than the mental model shows. The conceptual pieces of this object are more abstract than it appears when you look at the object which is confusing to users, especially new ones.

Observation Questions

1. Is the object simple, intuitive, consistent, coherent, and useful?
2. Is the feedback clear or unclear about what happened?
3. How are the controls grouped?
4. Is the item efficient based on its signifiers?
5. In what ways will those with disabilities have trouble using the software?
6. Does the user encounter anything expected or unexpected?

The object has many properties and is complex. It is relatively intuitive, consistent, coherent, and useful. While the purpose might be known to a user who just opened it, because of its intuitive design the user can navigate the basics. The feedback is very clear, and the object always confirms to the user when somethings has been done, whether that's correctly or incorrectly done. The controls are grouped together on the home screen. With all the containers together, settings together and other options on the left we can see how the grouping works. The way the controls are grouped by topic and choice makes the object intuitive. The item is efficient based on the signifiers it provides. But since there are lots of things the object does not signify the object is not 100 percent efficient. If the user knew about all the things it could do, then the object would be more efficient. People with disabilities will have a hard time using the software because they can't see everything they can do on the screen. People who have hand issues will have trouble using the computer to access the software. Additionally, those who have trouble with seeing their colors or color blindness will have trouble using the command prompt because of the harsh color tones. The user will encounter lots of unexpected things while using this object. There are so many errors that can occur while using a programming software that the user needs to be comfortable with problem solving and encountering new things along the way.

Audience

Research Questions

1. What is the connection between the object and the user?
2. What does the object communication to the user?

The object connects to the user via the computer. The user is using this object to complete tasks, objectives, and goals. For the user it is a "means to and end". The object communicates good signifiers and feedback for task competition. For example, when there is an update needed for Docker, the object notifies you. (See the manual for the image.)

Observation Questions

1. What are the user's weaknesses?
2. What does the user need to know?
3. How can the user be informed about what they need?
4. What does the user bring before encountering this object?
5. What problems are they expected to face?
6. Who are the users?

The user's weaknesses include lack of programming experience and lack of problem-solving abilities. The user needs to know how to install the software and use it, but this cannot be achieved by the object alone. Documentation or a manual is required for the installation process and the incorporating Angular part or the manual. The user can be informed by good directions, and explanations. There are a lot of confusing terms and issues in this project, so it must be explained well. The user brings their experience or lack of when they come to use this product. They are expected to face problems related to a Windows environment and must troubleshoot the issues. The user is someone knew to programming but has some basic experience.

Manual

Research Questions

1. What is needed to create a useful and usable manual?
2. What is simplest way for the user to complete the tasks required? (Meaning there could more than one way to install Angular, so which is the most effective)

To create a useful manual for the object, it must have a table of contents, good headings, and detailed instructions since some of things the object does are quite complex and hard to understand. I need to make sure the images have explanations above them with the step associated with them.

Observation Questions

1. What is most findable on the manual?
2. What can be created to make things, error free, memorable (easy to remember), and efficient for using?
3. How to create (subjective) satisfaction for users? (Meaning the user is happy and understands the manual)

On the manual the images are the easiest thing to find along with the headers. They provide the user with a way to section the information and decide if the section has the information that they need. To create an error free manual, it needs to be proof-read but also tested by real users to make sure the directions work to complete the tasks. It needs to be memorable it terms of its design so the user knows and remembers where things are at in the manual so they can go back for reference. It needs to be efficient, meaning it doesn't contain repetitive information and is precise in its language.

Research Methods on Manual

For my research methods on the manual, I completed task analysis with three users at different experience levels. Working with each of them helped me gather research for the audience analysis as well.

Participant One - Task Analysis

User Type: New to programming.

Say: Thanks for joining me today. Everything you say and do will be recorded for this report. Your performance is not about you but rather the interaction between you and the manual and you and the object.

Task	Notes
Download Docker onto Device	- User was able to download the software successful

	- They liked the layout - Asked about adding in better notes about the time it will take to download and install - Add a note about having to restart the device after installing and downloading it - Said the instructions need to be a bit more detailed for the new users - Link a note to troubleshooting
Make A Container	- User was able to make a container by following the instructions - Needed font a bit bigger for seeing everything

Follow Up Questions

1. What was confusing about the manual?
2. What would have helped you?
3. Did you have the correct signifiers for completing a task both from the manual and the object?
4. What are your overall thoughts?

Responses Received:

- Thought the manual was overall clear.
- Had the information they needed to complete the tasks.
- Manual signified what they needed to know.
- Overall, they liked the feel of the manual.
- They wanted to see some the information under each task laid out better.

Say: Thanks for participating in this task analysis!

Participant Two- Task Analysis

User Type: Never programmed.

Say: Thanks for joining me today. Everything you say and do will be recorded for this report. Your performance is not about you but rather the interaction between you and the manual and you and the object.

Task	Notes
Make A Docker Container	-User was able to make the container

	- No comments about the process
See if Docker needs to be updated	-User was able to see if Docker needed to be updated - Said that the instructions for the maintenance section of the manual were exactly what they needed to complete the task - Needed more information on what the “terminal” and “command prompt” are

Follow Up Questions

1. What was confusing about the manual?
2. What would have helped you?
3. Did you have the correct signifiers for completing a task both from the manual and the object?
4. What are your overall thoughts?

Responses Received:

- Wanted to see better spacing.
- Docker and Angular should be capitalized.
- Would have been easier if the image had been on the right.
- Had the correct signifiers to complete the assigned tasks.
- Overall, they liked the manual, especially the colors.

Say: Thanks for participating in this task analysis!

Participant Three- Task Analysis

User Type: Experienced programmer.

Say: Thanks for joining me today. Everything you say and do will be recorded for this report. Your performance is not about you but rather the interaction between you and the manual and you and the object.

Task	Notes
Install Docker on your device	-User had a brand-new computer which was perfect for testing this type of software - Ran into some issues that I had not encountered before so I added those things to the troubleshooting page

Create the Angular Container and view the website in the browser	-The user was able to create the Angular container with the nginx image -They ran into some other issues along the way that were incorrect in the manual I made -We made changes to the manual, ran it again and it worked fine - Added a note about rerunning commands (Something I hadn't thought of)

Follow Up Questions

5. What was confusing about the manual?
6. What would have helped you?
7. Did you have the correct signifiers for completing a task both from the manual and the object?
8. What are your overall thoughts?

Responses Received:

- Wanted to have things explained a bit more throughout.
- Like the overall design and layout of the manual.
- Ran into issues not in the manual but found the troubleshooting section easy to navigate.
- Gave feedback about a few typo/clarification items.
- Felt the steps were clearly labeled.

Note: This user had previously installed Docker and used angular but felt the process was extremely confusing before. They loved the new manual and the images that show what feedback should be received after each step.

Say: Thanks for participating in this task analysis!

Why a Manual is Needed

A manual is needed because there a lot of steps to install Docker and use Angular inside Docker. Additionally, there is not a manual for how to set up an Angular environment in Docker. There is some document on how to install Docker and Angular but not both together. The documentation that does exist for installing Docker is hardly “documentation” at all.

The audience needs to have things laid out at their basic programming understanding level and then go from there. In the above sections you can find the analysis of the object, audience, and the manual. From the object analysis and the audience analysis it is clear what needs to be done for the user to understand the object. The directions need to be clear; somethings have settings and signifiers, but others don’t so communicating them in the manual is key for the user to understand. The manual can then address what the user needs to know about the object, like key terms that are not key clear and key visuals that are not included in the Docker set up link. In the new manual there are key terms explained on the side for users that are still learning.

Summary

Based on the research questions conducted at the beginning of this project and the user testing on the manual/object I was able to take that information and incorporate it into my usable manual for an audience with limited programming knowledge. The manual meets the needs of the user based on them having some programming experience but not a lot. The third test participant is an experienced programmer and he noted things that might have gone over a beginner's head that should be changed. Based on the testing results we know the manual is clear and concise and even works properly on a brand-new computer.

Conclusion

Thank you for reading my usability report for this project. If anything is confusing and unclear or if you have questions and comments, please feel free to reach out via email at rswilli1@svsu.edu.