

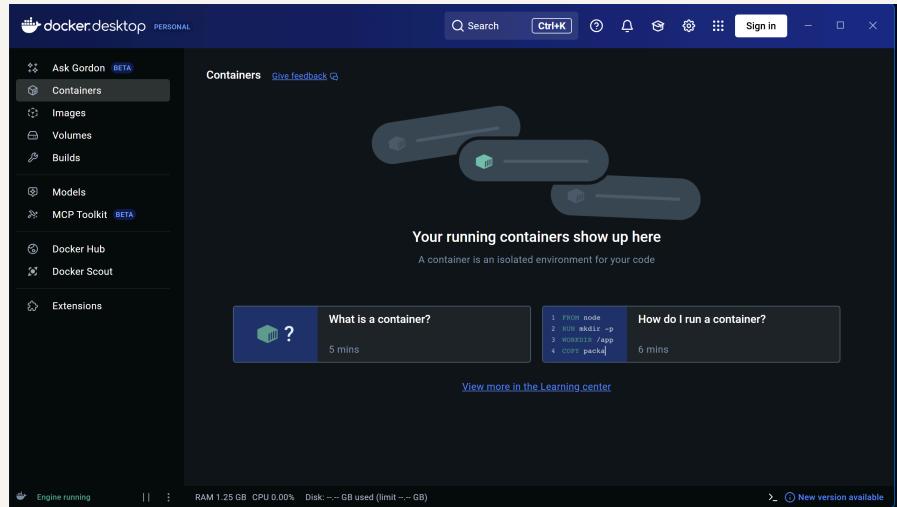


[nextwork.org](https://nextwork.org)

# Deploy an App with Docker

A

Ashmit Rajput



# Introducing Today's Project!

## What is Docker?

Docker is a containerization tool and it is useful because it stores your code, tools, environment variables all in one box, so that box can be used by all user and all devices to use the application. It is faster and more efficient than using VM's.

## One thing I didn't expect...

One thing I didn't expect in this project was for it to be complicated yet so fast and efficient, Docker has various images and is easy to use, its integration with Elastic Beanstalk is very functional and cuts down manual efforts a lot.

## This project took me...

This project took me approximately 3 hours to finish, mostly because I learned about all the topics in detail by reading documentations online, I also tried different configurations while setting up my Environment, all this was a learning experience.

# Understanding Containers and Docker

## Containers

Containers are like those big shipping containers, stacked on cargo ships. Each container has everything an app needs to run, its own code, libraries, and settings. You can load a bunch of these containers onto your computer, and they work smoothly.

A container image is a blueprint or template for making containers. It tells Docker exactly what to put inside each container, things like your app's code, the libraries it needs, and any other required files

## Docker

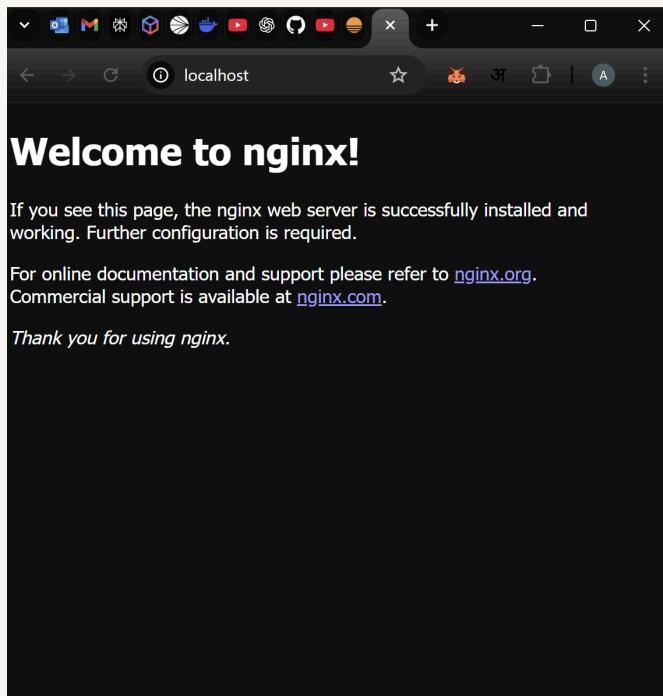
Docker helps you create, manage, and deploy containers with everything needed for your apps efficiently. Docker Desktop lets you manage everything about your containers. You can create new containers, adjust their settings, or monitor how they run.

the Docker daemon is a background process that manages the Docker containers on your computer. It takes commands from the Docker client (i.e. commands you type into the terminal, or clicks you make through the Docker Desktop), it does all the work.

# Running an Nginx Image

Nginx is a web server, which is a program you use to run websites and web apps. If you want people to visit your site, you're going to need a web server to deliver your website's files to their browsers.

I ran `docker run -d -p 80:80 nginx` to start a new container, `docker run` starts a new container and we use a pre-existing image of 'Nginx', and `-d` means in detached mode, and `-p 80:80` maps port 80 of host machine to port 80 in the container.

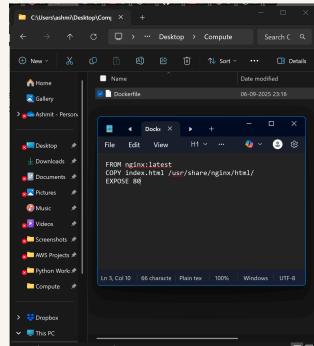


# Creating a Custom Image

A Dockerfile is a document with all the instructions for building your Docker image. Docker would read a Dockerfile to understand how to set up your application's environment and which software packages it should install.

My Dockerfile tells Docker three things FROM nginx:latest, to copy Nginx and then continue ahead, COPY index.html /usr/share/nginx/html/, copy html file after Nginx file, use both these files and wrok to, EXPOSE 80, show it on Port 80.

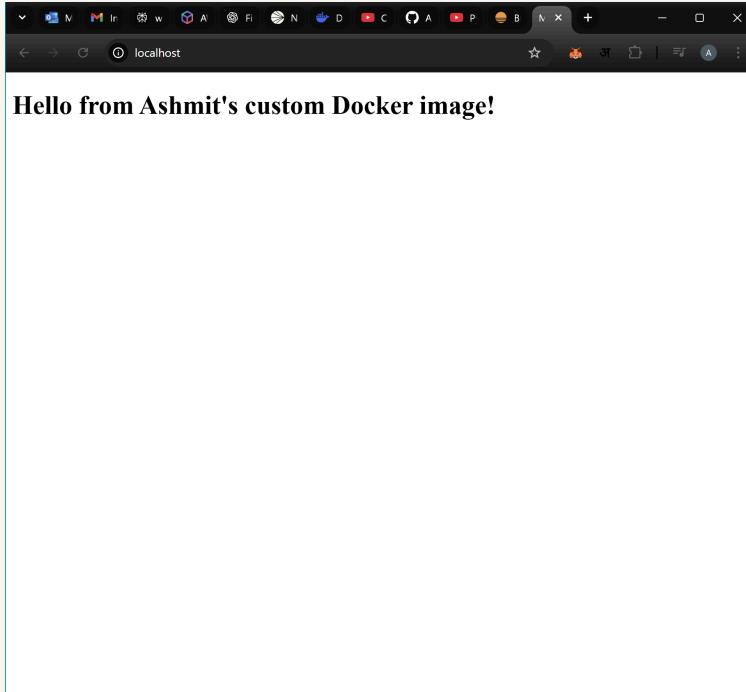
The command I used to build a custom image with my Dockerfile was docker build -t my-web-app . Which creates an Image called "my-web-app", The '!' at the end of the command means that Docker will find the dockerfile in the current directory.



# Running My Custom Image

There was an error when I ran my custom image because my NginX image was already running on port 80, which I started when I was testing out docker. I resolved the issue by stopping the Nginx image and then reassigning port 80 to our my-web-app image

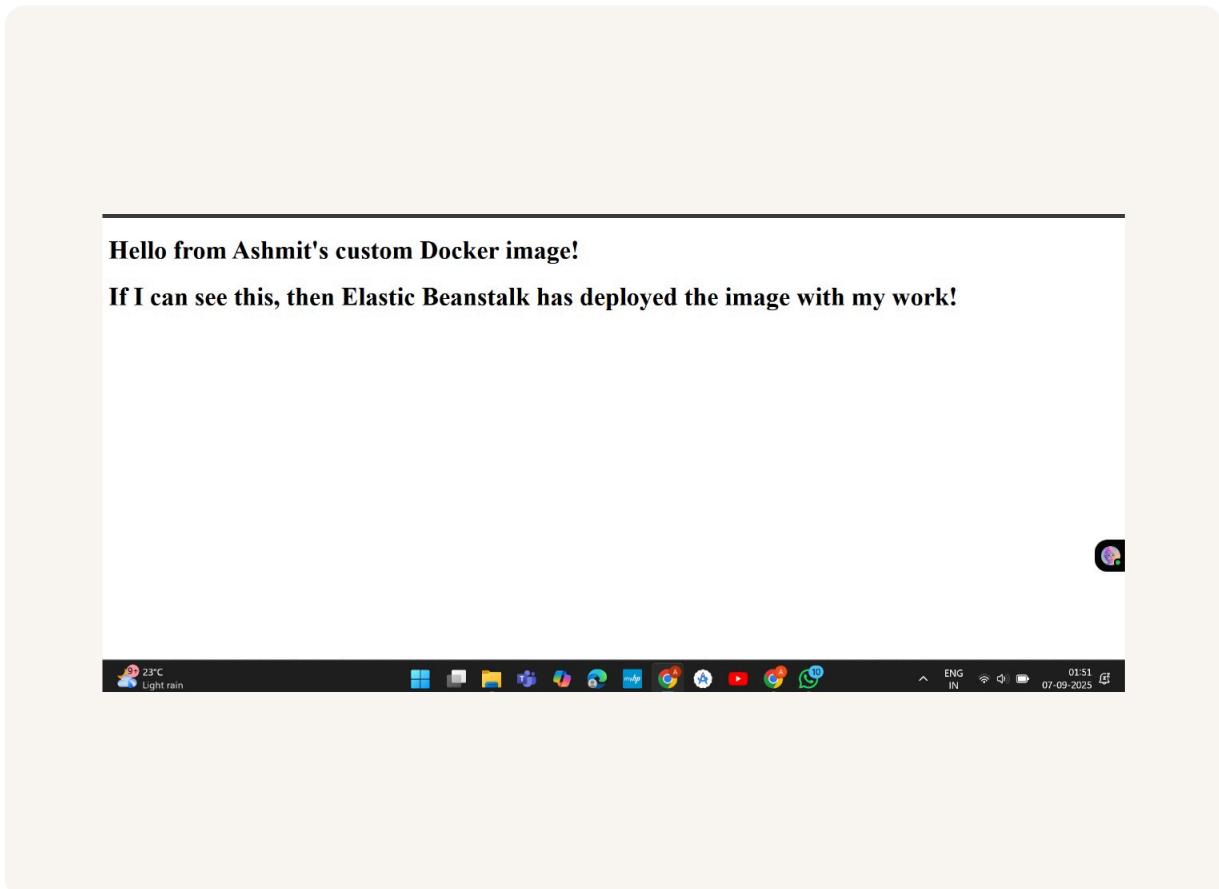
In this example, the container image is the blueprint that tells Docker the application code, dependencies, libraries etc that should go into a container. The container is the actual software that's created from this image and running on the web.



# Elastic Beanstalk

Elastic Beanstalk is AWS's way of running your Docker containers without the setup headaches. Build and test your app locally, wrap it into a Docker image, then hand that image to Elastic Beanstalk. It spins up the servers, launches your container.

Deploying my custom image with Elastic Beanstalk is a comparatively easier way to do as I don't have to manage my own code and initiate EC2 instances on my own, it all gets configured on its own. Also easily integrates with many platforms e.g Docker





[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

