

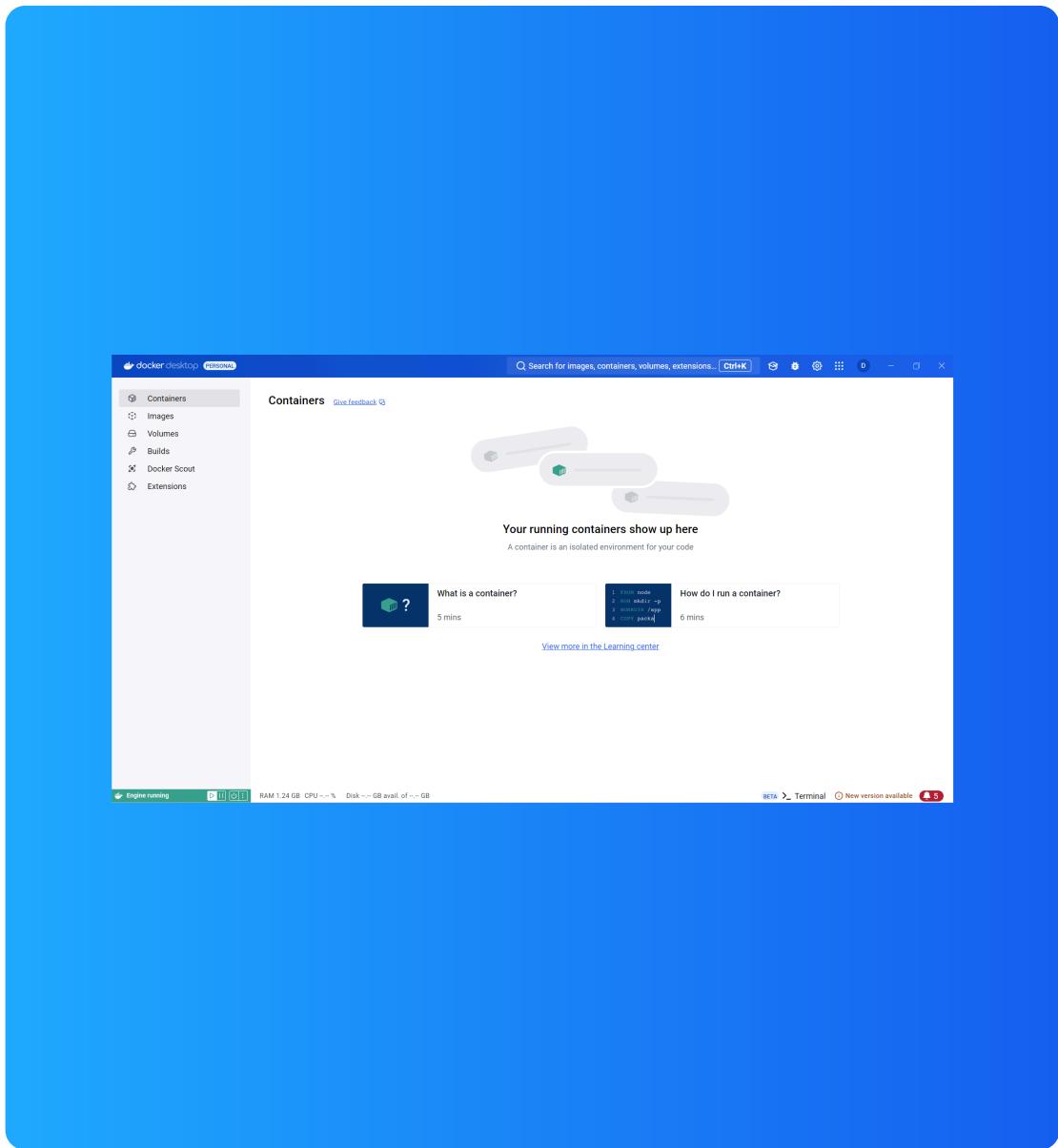


NextWork.org

Containers on Elastic Beanstalk



dineshrajdhanapathy@gmail.com



Introducing Today's Project!

What is Docker?

In this project, we used Docker to create containers based on container images and set up our own container image.

One thing I didn't expect...

one things we didn't expect was seeing how quick it is to deploy an application using Elastic Beanstalk.

This project took me...

This project took us 3 hours including all the demo time.

Understanding Containers and Docker

Containers

Containers are lightweight, portable environments that package an application and its dependencies, isolating them from the underlying system. This ensures consistent performance across different computing environments. Docker manage, deploy, & scale

A container image is a lightweight, standalone, and executable package that includes everything needed to run a piece of software: code, runtime, libraries, and settings.

Docker

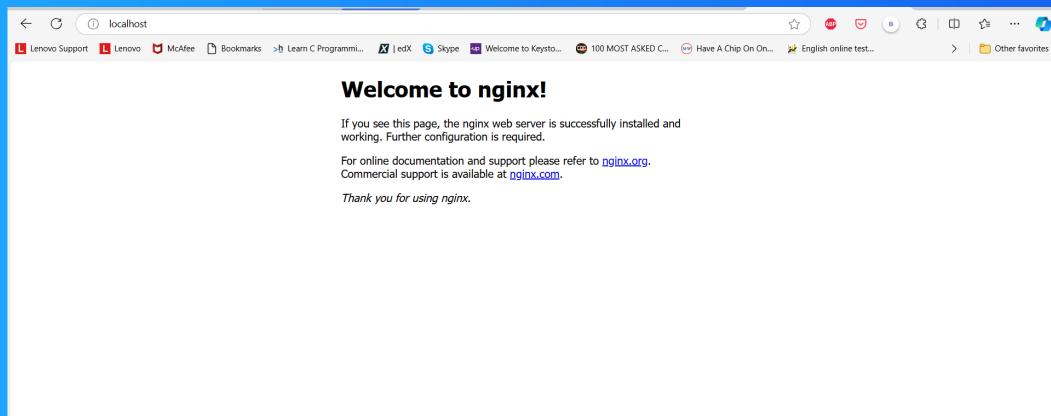
Docker is a platform that enables developers to build, run, and manage containers—isolated environments for applications and their dependencies. Docker Desktop is a GUI tool that provides an easy way to create, test, and manage Docker contain local.

The Docker daemon is a background service that manages Docker objects, such as containers, images, and networks. It listens for API requests, builds and runs containers, & handles tasks like image pulling and container lifecycle management on a host.

Running an Nginx Image

Nginx is a high-performance web server and reverse proxy used for serving static content, load balancing, and caching. Known for its speed and scalability, it's used to handle large volumes of concurrent connections, making it ideal for web apps.

The command I ran to start a new container was 'docker run -d --name <container-name> <image-name>' Here, -d runs the container in detached mode, --name assigns a name, and <image-name> specifies the image to use.

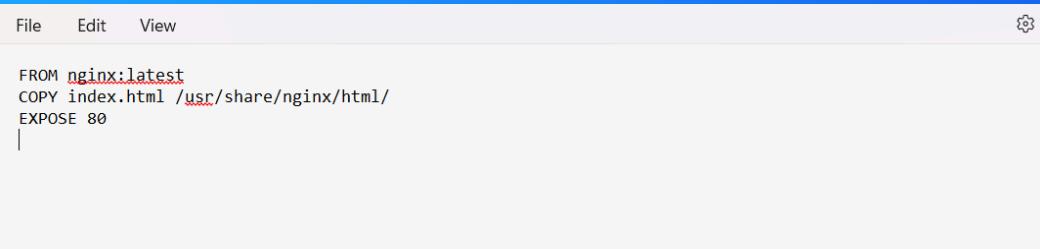


Creating a Custom Image

A Dockerfile is a text file that contains instructions for building a Docker image. It defines the base image, environment variables, dependencies, and commands to configure the application inside the container. It uses it to automate image creation

My Dockerfile tells Docker three things. First, our custom container images uses the (latest version of the Nginx container image at its base. Then, we're modifying this base by replacing the default nginx welcome page with our own custom index.html

The command we used to build a custom image with our Dockerfile was 'docker build'. The '.' at the end of the command means that docker can find the Docker file in the current directory i.e. the compute folder on our desktop.

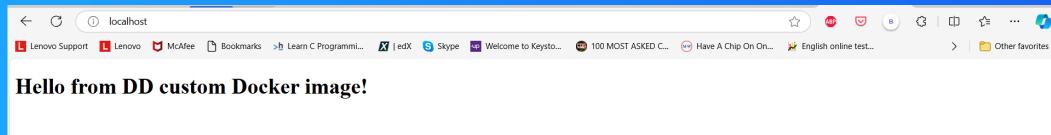


```
File Edit View
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

Running My Custom Image

There was an error when we ran my custom image because we tried to map our port 80 with the new container's port 80, but a running container was already using port 80. We resolved this by stopping the running container so that we can start a new one.

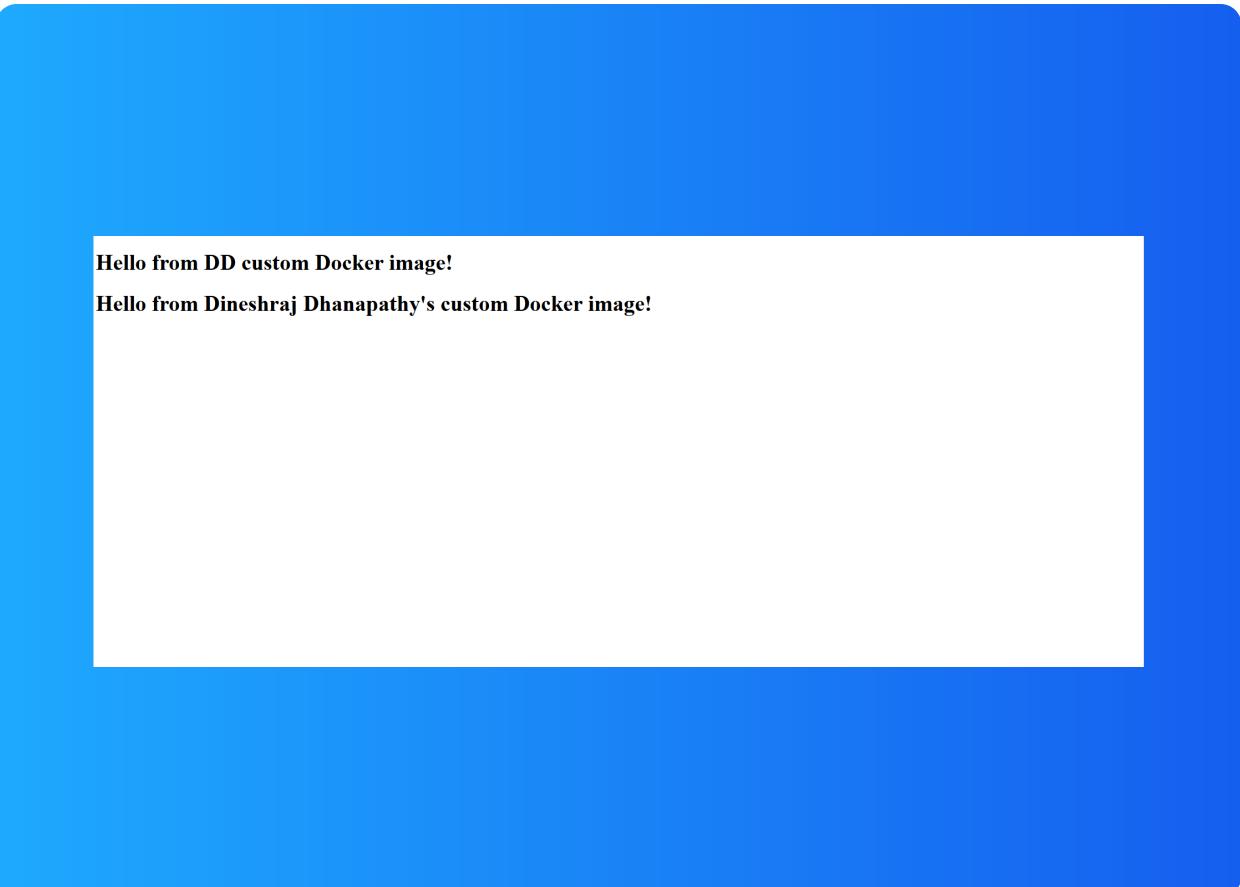
In this example, the container image is template for creating a new container running an Nginx server that's serves our custom index.html file. The container is the actual software that's running an Nginx web server with those customisation.



Elastic Beanstalk

Elastic Beanstalk is an AWS service that simplifies deploying and managing apps by automatically handling infrastructure tasks like load balancing, scaling, and monitoring. You upload your code, and it manages the resources, make deploy fast & easy.

Deploying my custom image with Elastic Beanstalk took us 10 minutes. This includes the time it took to launch the Elastic Beanstalk application.



Hello from DD custom Docker image!
Hello from Dineshraj Dhanapathy's custom Docker image!



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

