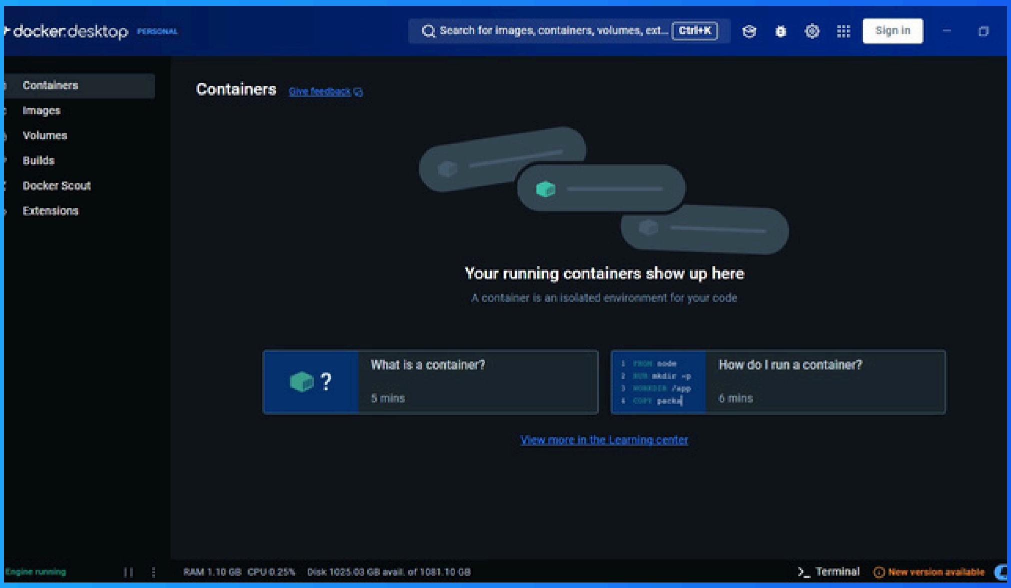
NextWork.org

Containers on Elastic Beanstalk

Hassan Gachoka



Your running containers show up here

A container is an isolated environment for your code

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

RAM 1.10 GB CPU 0.25% Disk 1025.03 GB avail. of 1081.10 GB

Introducing Today's Project!

What is Docker?

Docker is a platform for creating, running, and managing containers, which package applications with everything they need to run. In today's project, I used Docker to build a custom container image for my app and deployed it with Elastic Beanstalk.

One thing I didn't expect...

One thing I didn’t expect in this project was how straightforward deploying Docker applications on Elastic Beanstalk is. It simplifies managing infrastructure, letting me focus on building and packaging my app!

This project took me...

This project took me 2 hours to set up Docker with app code, created a container image, uploaded it to Elastic Beanstalk, configured the environment, deployed the app, tested it live, and confirmed it worked seamlessly online.

Understanding Containers and Docker

Containers

Containers are portable packages that bundle an app and its dependencies. They are useful because they ensure the app runs consistently across different environments, solving the "it works on my machine" problem.

A container image is a blueprint for containers, containing all the code, libraries, dependencies, and files needed to run an application. It ensures consistency and allows identical containers to be created across environments.

Docker

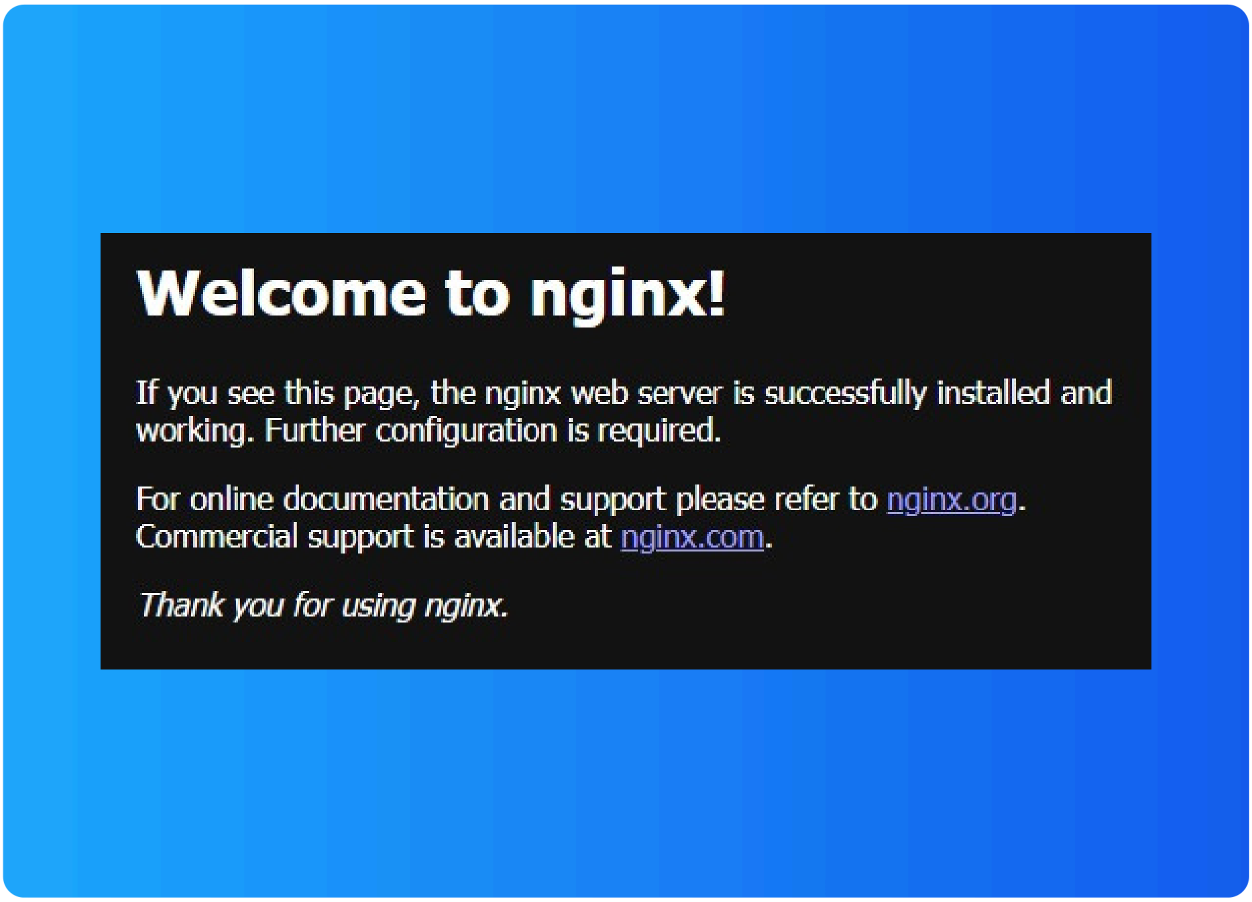
Docker is a tool that helps create, manage, and deploy containers efficiently. Docker Desktop is an app that makes using Docker easier, providing a user-friendly way to build, test, and manage containers directly from your computer.

The Docker daemon is a background process that manages Docker containers. It runs commands, builds, and deploys containers, acting as the engine behind Dockers̓’ operations to ensure everything works smoothly.

Running an Nginx Image

Nginx is a powerful web server used to serve websites and handle high volumes of traffic efficiently. It can also act as a proxy server to balance loads and forward requests, making it a favorite for scalable applications.

The command I ran to start a new container was docker run -d -p 80:80 nginx, which started an Nginx container in detached mode and mapped port 80 of my machine to port 80 in the container.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

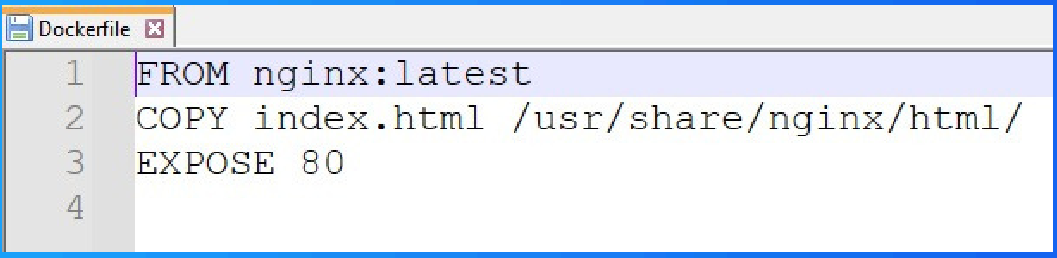
Thank you for using nginx.

Creating a Custom Image

The Dockerfile is a text file containing instructions for building a Docker image. It tells Docker what base image to use, what files to include, and how to configure the environment for the container.

My Dockerfile tells Docker three things: FROM nginx:latest starts with the latest Nginx image as the base. COPY index.html /usr/share/nginx/html/ replaces the default web page with my custom HTML. EXPOSE 80 opens port 80 to receive web traffic.

The command I used to build a custom image with my Dockerfile was docker build -t my-web-app . The . at the end of the command means Docker should look for the Dockerfile in the current directory.



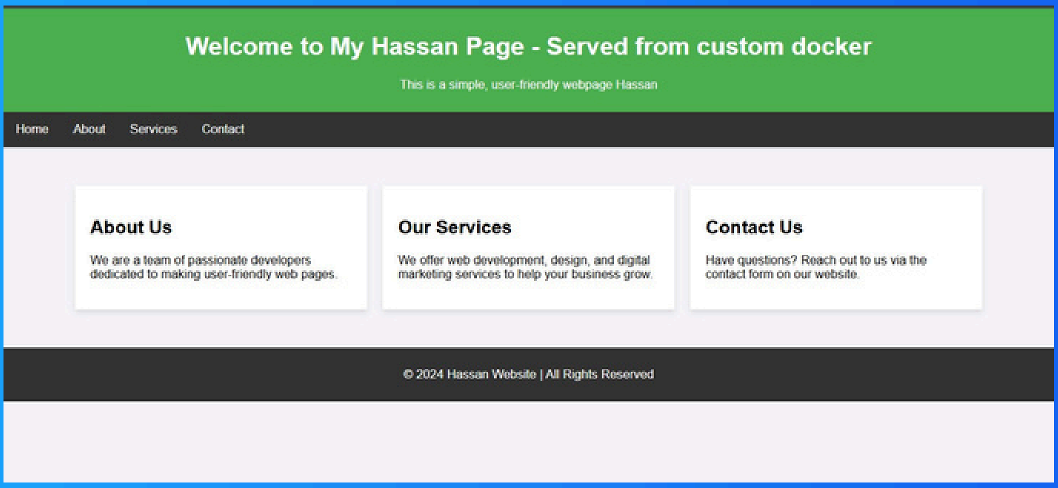
```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80

```

Running My Custom Image

There was an error when I ran my custom image because port 80 was already in use by another container. I resolved this by stopping the existing container using Docker Desktop or the docker stop {container_id} command.

In this example, the container image is the my-web-app image that we built using the Dockerfile. The container is the running instance created from that image, which serves the custom webpage through port 80 on your local machine.



Welcome to My Hassan Page - Served from custom docker

This is a simple, user-friendly webpage Hassan

About Us

We are a team of passionate developers dedicated to making user-friendly web pages.

Our Services

We offer web development, design, and digital marketing services to help your business grow.

Contact Us

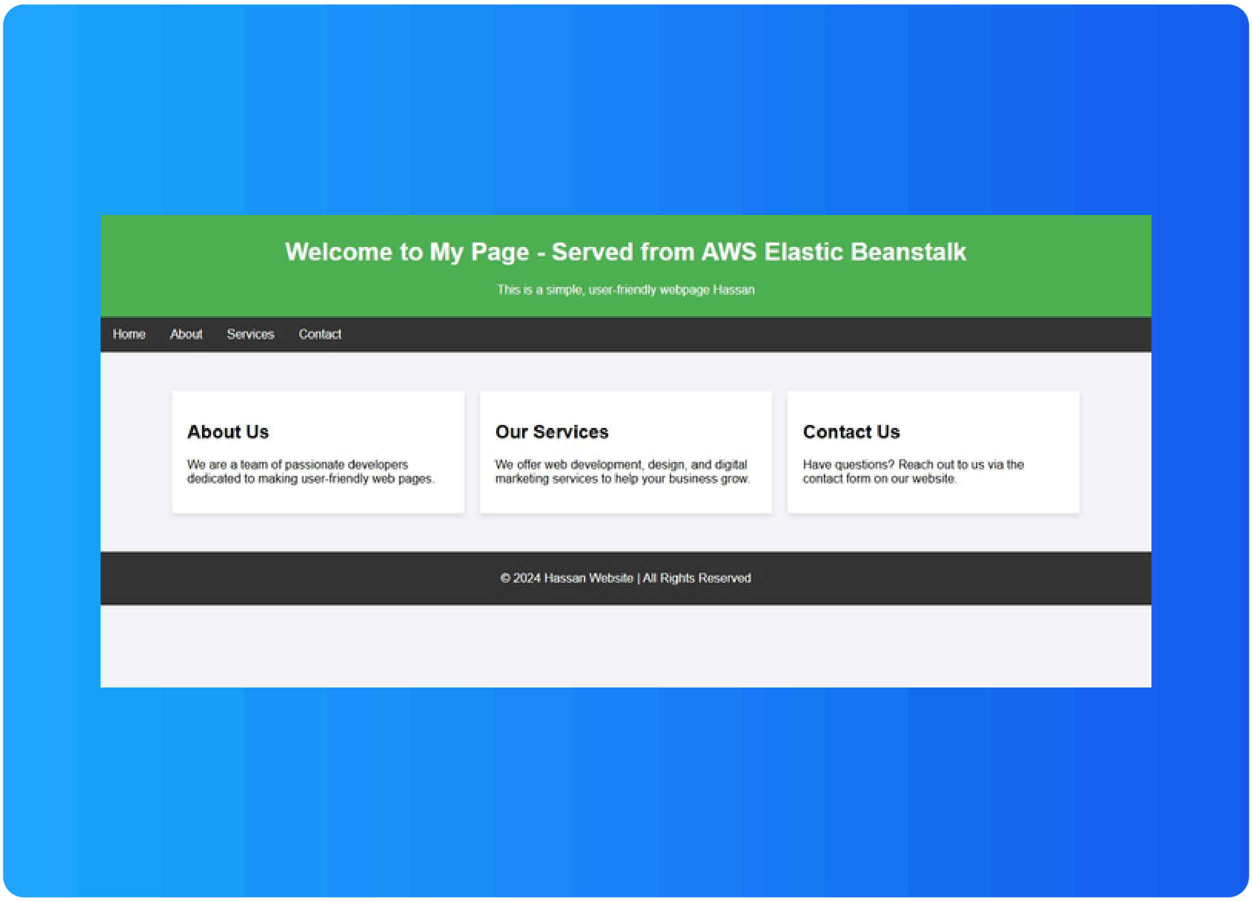
Have questions? Reach out to us via the contact form on our website.

© 2024 Hassan Website | All Rights Reserved

Elastic Beanstalk

Elastic Beanstalk is an AWS service that simplifies deploying and managing web applications. It automatically handles infrastructure tasks like server provisioning, load balancing, scaling, and monitoring, allowing developers to focus on writing code

Deploying my custom image with Elastic Beanstalk took me about 10 minutes. The process included setting up the environment, uploading the Docker image, and waiting for Elastic Beanstalk to configure and launch the application.



Welcome to My Page - Served from AWS Elastic Beanstalk

This is a simple, user-friendly webpage Hassan

About Us

We are a team of passionate developers dedicated to making user-friendly web pages.

Our Services

We offer web development, design, and digital marketing services to help your business grow.

Contact Us

Have questions? Reach out to us via the contact form on our website.

© 2024 Hassan Website | All Rights Reserved