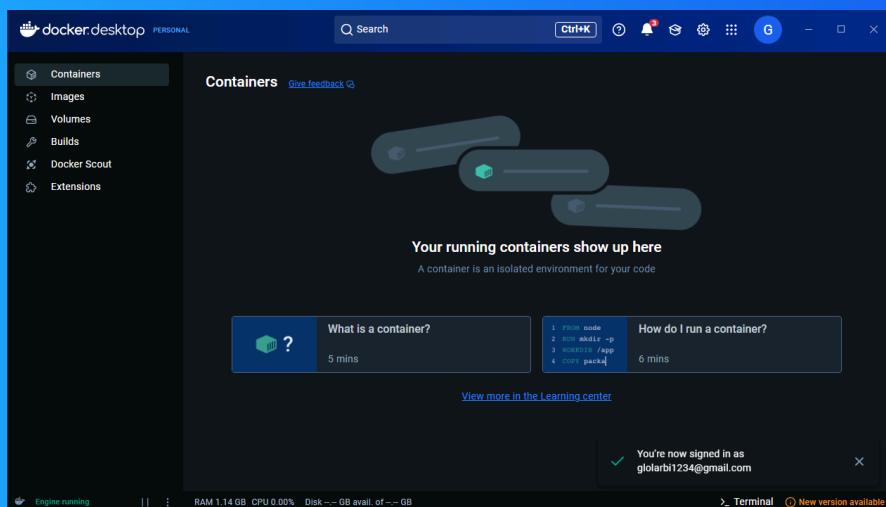




Containers on Elastic Beanstalk

G Gloria





Introducing Today's Project!

What is Docker?

Docker is a tool that creates containers, which are lightweight, portable environments for running applications. I used Docker to build a container image for my app, run it locally, and deploy it to AWS Elastic Beanstalk to make it available online.

One thing I didn't expect...

One thing I didn't expect in this project was how simple and easy Elastic Beanstalk makes the deployment process. It handled the infrastructure setup, scaling, and management, allowing me to focus more on my application and less on infrastructure.

This project took me...

This project took me 30 minutes to complete. It involved building a Docker image, deploying it to AWS Elastic Beanstalk, and seeing the application live on the internet, all within a straightforward and efficient workflow.



Understanding Containers and Docker

Containers

Containers are lightweight, portable packages that bundle applications and their dependencies, ensuring consistent behavior across different environments. They solve the "it works on my machine" problem, enabling faster development, and deployment.

A container image is a template with everything needed to run an app, like code, libraries, and files. It helps create identical containers that work the same everywhere, making it easy for teams to share and run applications without setup issues.

Docker

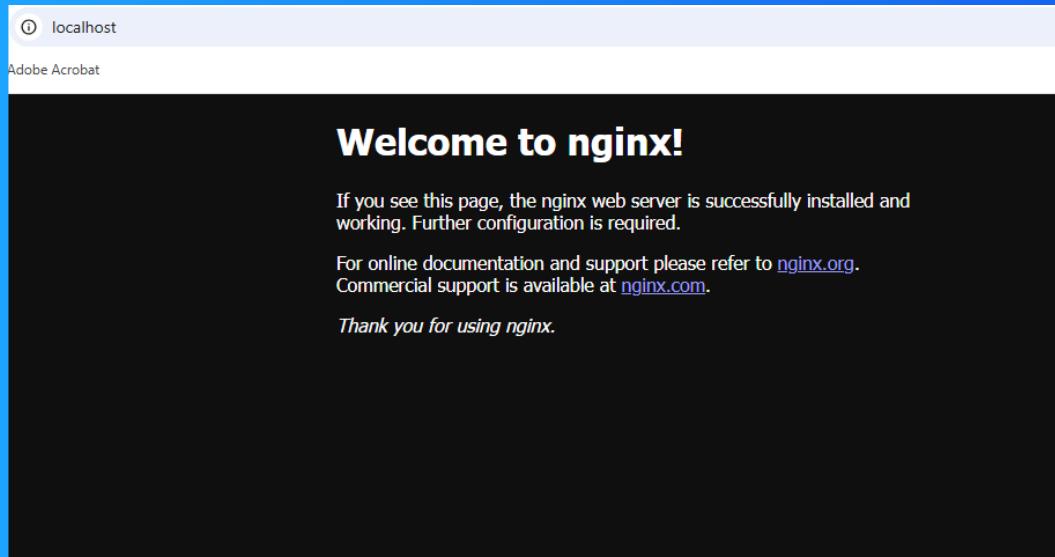
Docker is a tool that helps package apps and everything they need to run into containers, making them work the same everywhere. Docker Desktop is an easy-to-use program for your computer that lets you create, test, and run containers locally.

The Docker daemon is the core service of Docker that runs in the background. It manages Docker containers, images, and networks. It listens for commands from the Docker client, processes them, and handles tasks like building, running, containers.

Running an Nginx Image

Nginx is a web server that delivers web pages to users. It also works as a reverse proxy, load balancer, and caching tool. It's fast, reliable, and used to handle high traffic for websites and apps efficiently.

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



Creating a Custom Image

The Dockerfile is a text file with instructions that tell Docker how to build a custom container image. It includes steps like selecting a base image, adding files, installing software, and setting commands to run inside the container.

My Dockerfile uses the latest Nginx image as the base (FROM nginx:latest), copies my `index.html` into the default Nginx directory (COPY index.html /usr/share/nginx/html/), and exposes port 80 (EXPOSE 80) to allow the container to serve the webpage.

The command I used to build a custom image with my Dockerfile was docker build -t my-web-app . .The -t my-web-app assigns a name to the image, and the `.` at the end specifies the current directory as the location of the Dockerfile and related files.

```
GNU nano 2.0
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

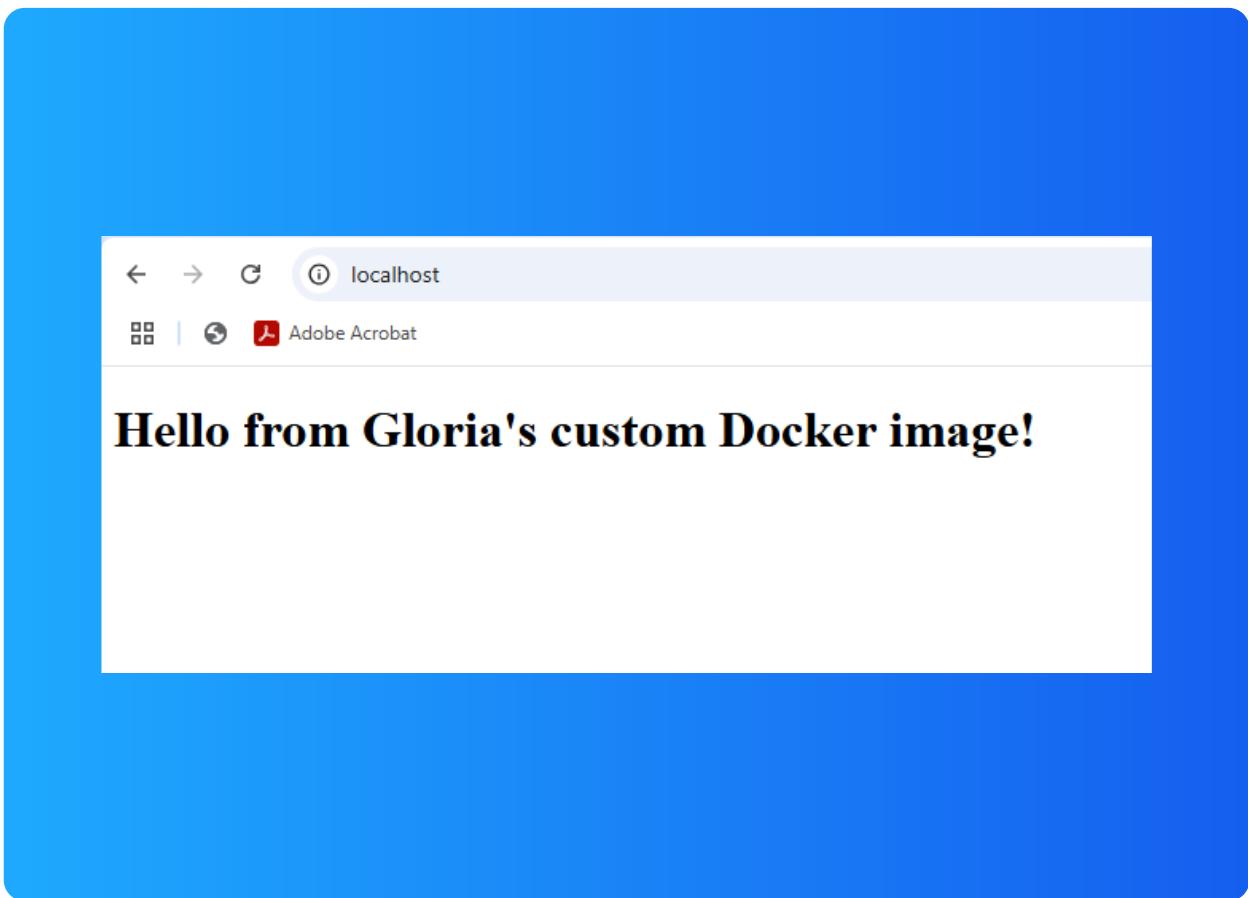
Read 3 lines

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo ^X Exit ^R Read File ^V Replace ^O Paste ^J Justify ^Y Go To Line M-E Redo M-A Set Mark M-G Copy

Running My Custom Image

There was an error because another container was using port 80. I fixed it by finding the running container with `docker ps --filter "publish=80"` and stopping it using docker stop {container_id}. Then, I ran my custom image again without any issues.

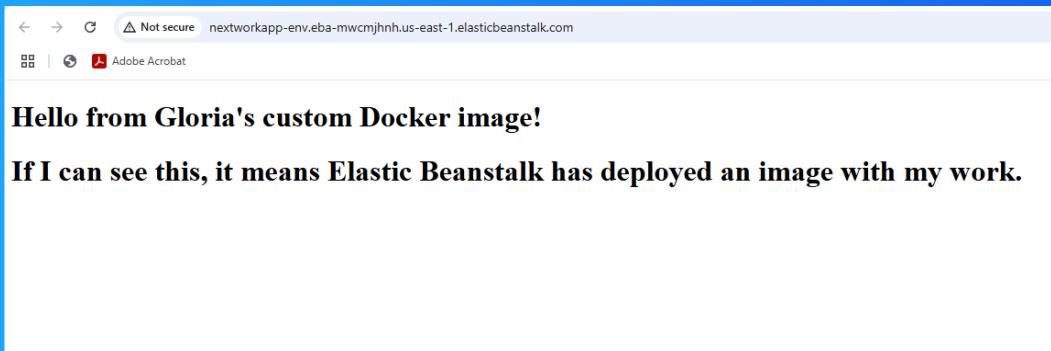
In this example, the container image is the template (`my-web-app`) that has all the files and settings. The container is what runs from the image, acting like a working app that shows the webpage and uses the computer's resources.



Elastic Beanstalk

Elastic Beanstalk is a service by AWS that helps deploy and manage applications in the cloud. It handles tasks like setting up servers, scaling, and monitoring, so developers can focus on their code instead of managing infrastructure.

Deploying my custom image with Elastic Beanstalk took me 10 minutes. It was a quick process that involved setting up the application, uploading the Docker image, and letting Elastic Beanstalk handle the infrastructure setup automatically.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

