

## Experience Report

Prior to embarking on this MERN project, my approach to web development and deployment was predominantly monolithic. This assignment was a revelation to discover that not all projects are bound to such a singular approach. This assignment exposed me to the professional deployment of MERN projects, revealing the flexibility and efficiency of microservices architecture—a glaring contrast to the monolithic structure I was accustomed to.

Working with Docker, I experienced firsthand the advantages of microservices. This architecture not only streamlines the development process but also enhances the application's scalability and maintainability. It was gratifying to delineate the services within the Docker Compose file, clearly defining the roles of the frontend, backend, and database. Each service was a cog in a larger, well-oiled machine, seamlessly interacting yet operating independently. The satisfaction of successfully configuring persistent storage with MongoDB's volume feature affirmed my belief in the necessity of data durability.

Configuring Nginx to act as a reverse proxy was a defining moment. By directing traffic to the backend service and listening on port 80, the setup highlighted the elegance and simplicity with which Docker handles networking. The feeling of contentment I got from knowing I could navigate complex networking configurations was empowering. The process of bringing down containers using Docker Compose in the command prompt was equally gratifying. Ensuring a clean state to work from not only prevented potential issues but also imparted a sense of control and professionalism in managing the application's lifecycle.

Deploying the application locally gave me a surge of accomplishment. Registering, logging in, and interacting with the application on my local machine felt like watching my creation come to life—a true testament to the effectiveness of the Docker environment. When it came time to deploy to an EC2 instance, the steps taken to log in through PowerShell and rebuild the Docker images from scratch were

a true application of theory into practice. Avoiding the use of cache ensured that my deployment was fresh and up-to-date, mirroring the meticulous nature of my approach.

Launching the containers in detached mode using Docker Compose rounded off the experience. Knowing that the services were running in the background, ready to scale and update as needed, instilled a profound sense of achievement. This journey from monolithic apprehension to microservices mastery has been transformative. It's not just about understanding the technology—it's about grasping the architectural philosophy that underpins modern, efficient, and cost-effective web application development. The feeling of elation from successfully configuring the project for both local development and production can't be overstated—it's a milestone that signifies growth, understanding, and the ability to adapt to the ever-evolving landscape of software development.