IT4090

Cloud Computing

4th Year, 2nd Semester

Assignment 1 – Report

Lakshan wijewardana W.M.W

IT18132588

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

2021.11.07

Contents

1. Architecture

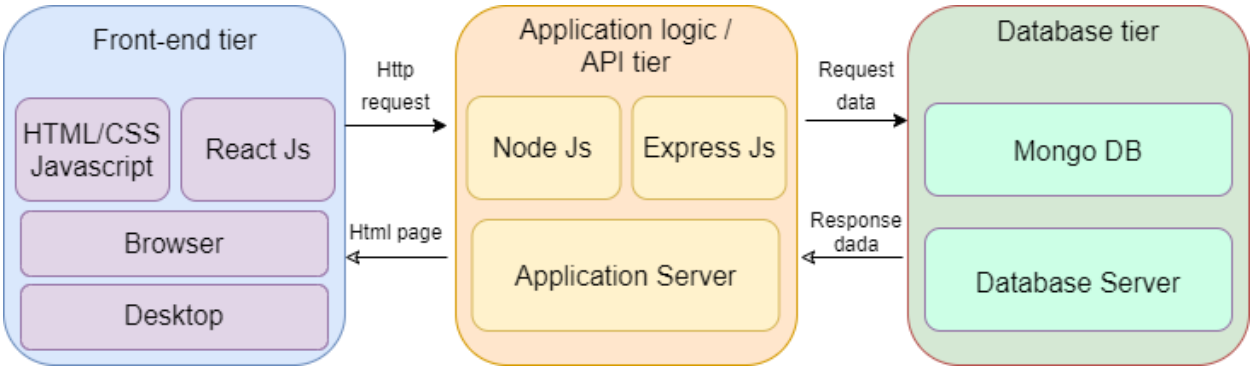

Figure 1:3 Tier Architecture

This application's architecture is built on a standard MVC model. The MVC architectural pattern is a popular paradigm for implementing this model. The logic, data, and visualization are split into three sorts of objects in the MVC paradigm, each performing its own duty. The View is in charge of the visual aspect, as well as user interaction. The Controller reacts to system and user events by instructing the Model and View to modify. The Model manipulates data, responding to information requests, and modifying its state in accordance with the Controller's instructions. Our front-end tier (View) will be developed in Javascript, HTML, and CSS, with the framework ReactJS. This is the level of architecture with which the user will interface in order to access the functionality of our application.

The Application Logic Tier (Controller)/API will be developed in NodeJS and ExpressJS, and it will represent the Application Server that will act as a communication bridge between the Front-end Tier and the Database Tier. This tier will provide HTML pages to the user's device, accept HTTP requests from the user, and answer properly.

MongoDB (Database Tier (Model)) is where we will save all of the critical data that our application need to work.

2. Application Logic

This is Student details management application. User can add students' name, address and course name and user can delete, update, view students' details. Student details are stored in Mongodb database.

3. Screenshots of Docker environment

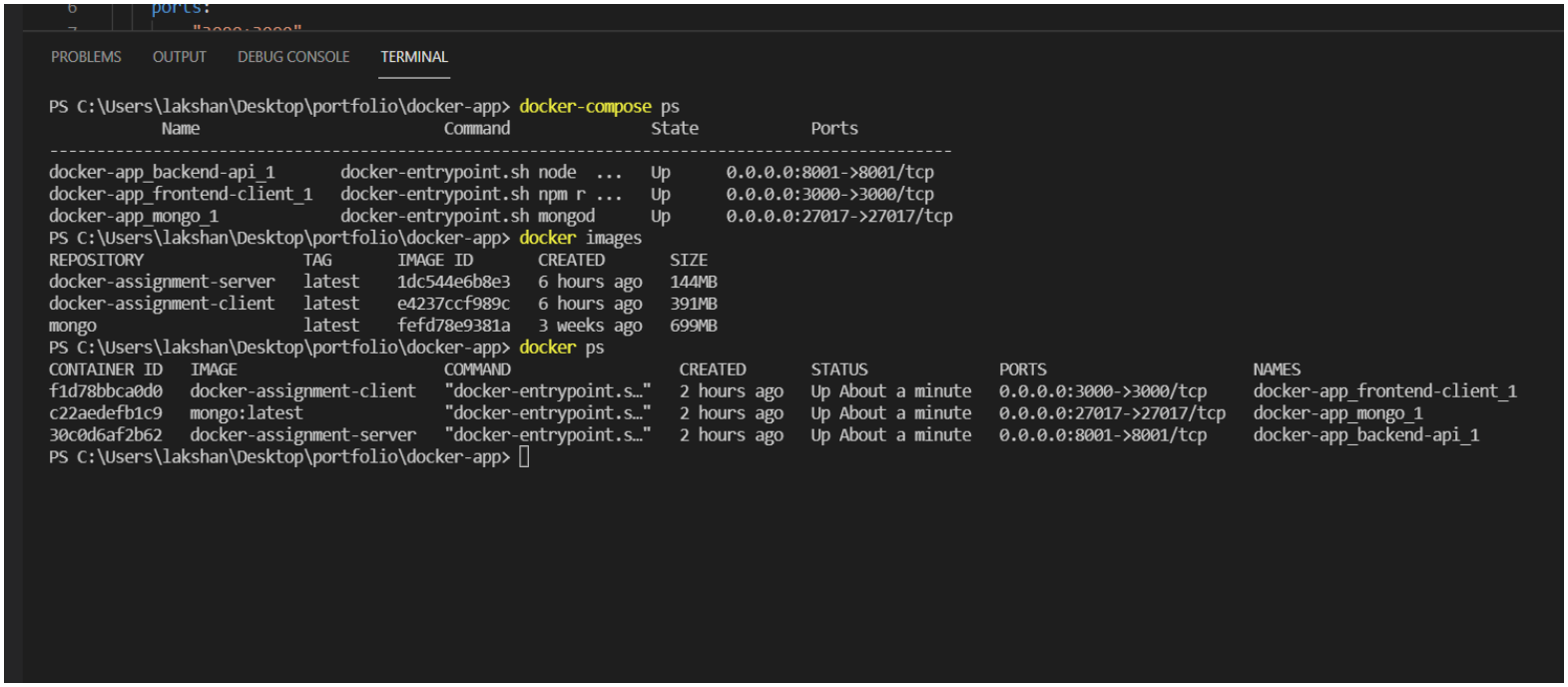

Figure 2: Docker images

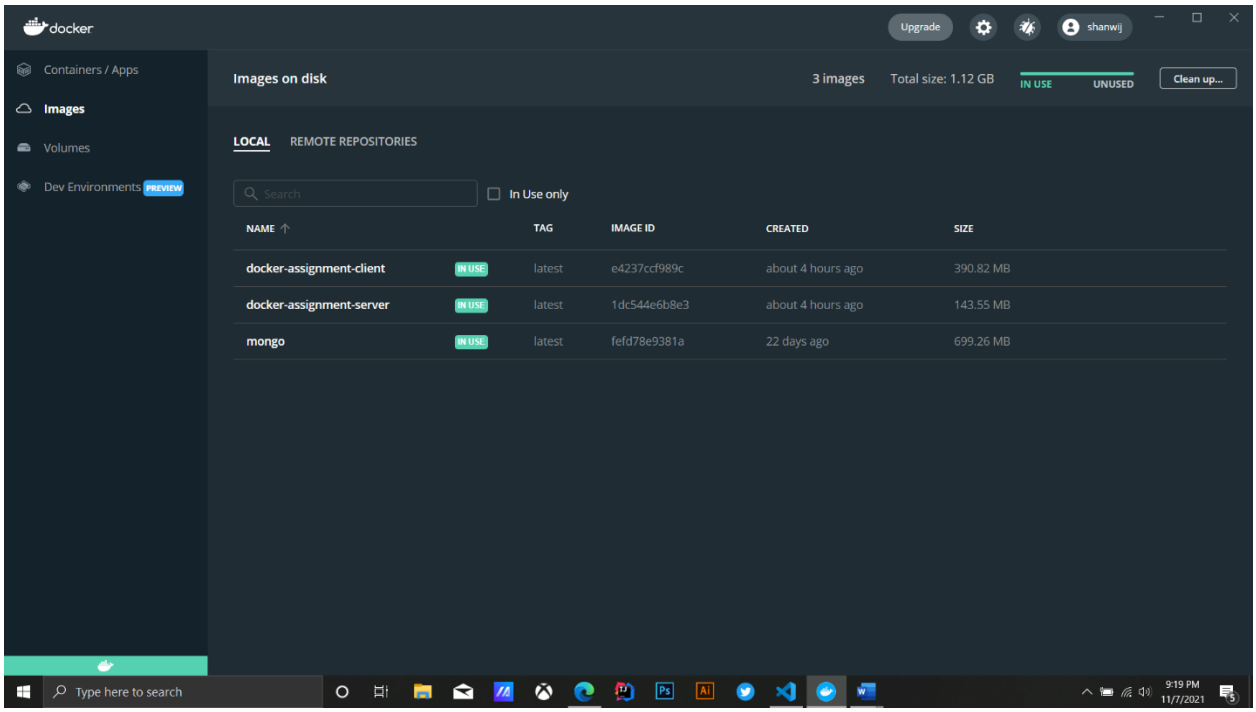

Figure 3: Docker environment

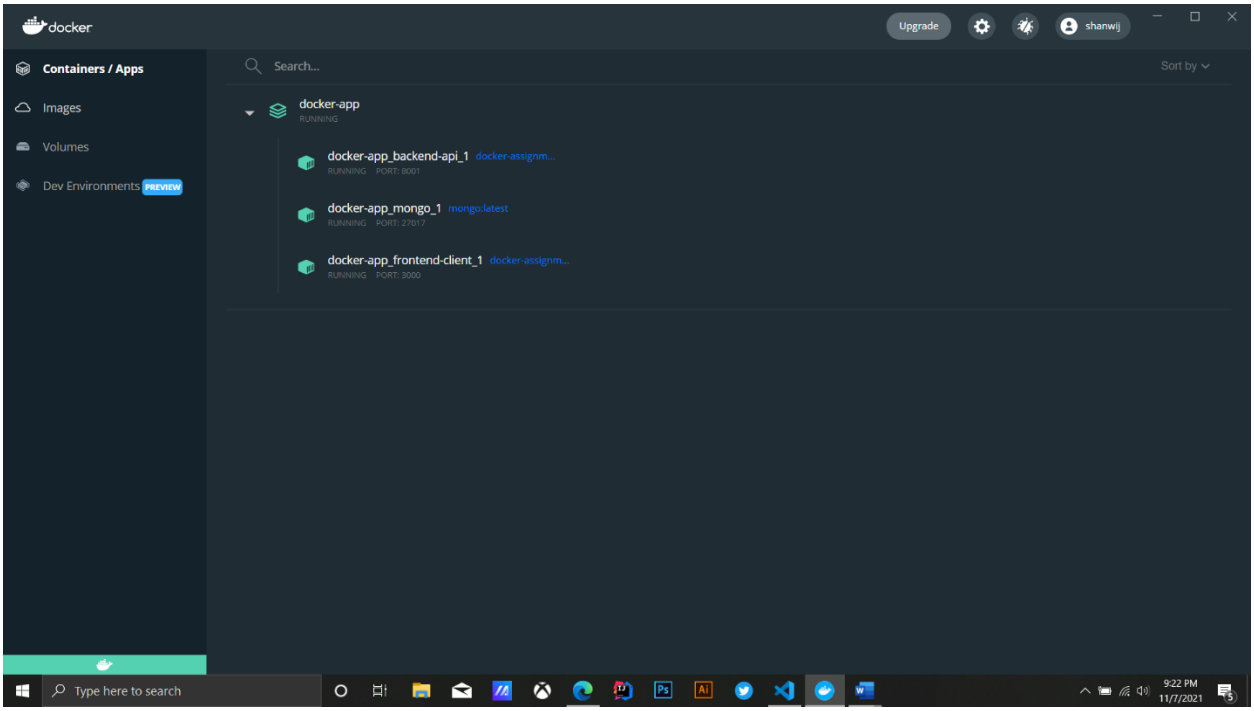

Figure 4:Docker Container

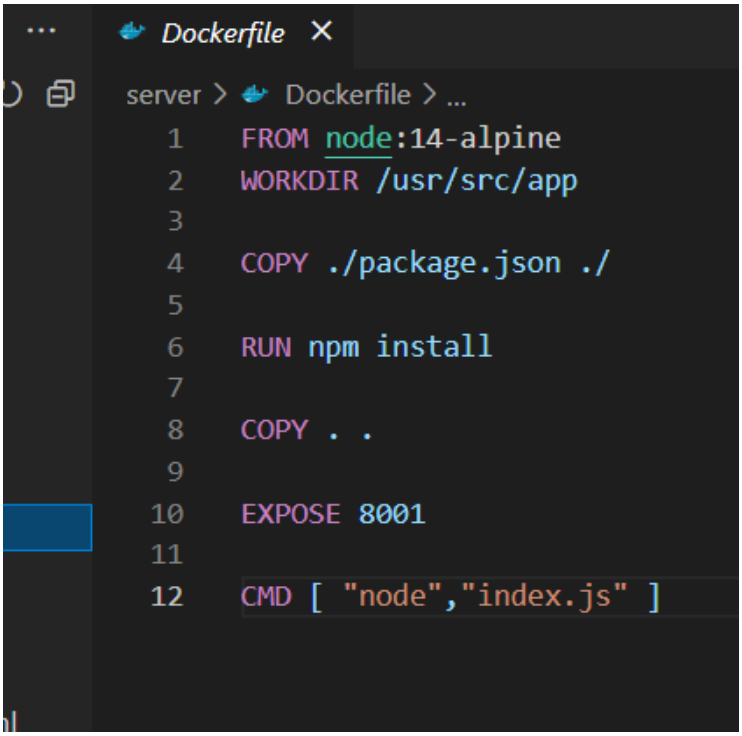

Figure 5: Dockerfile - server

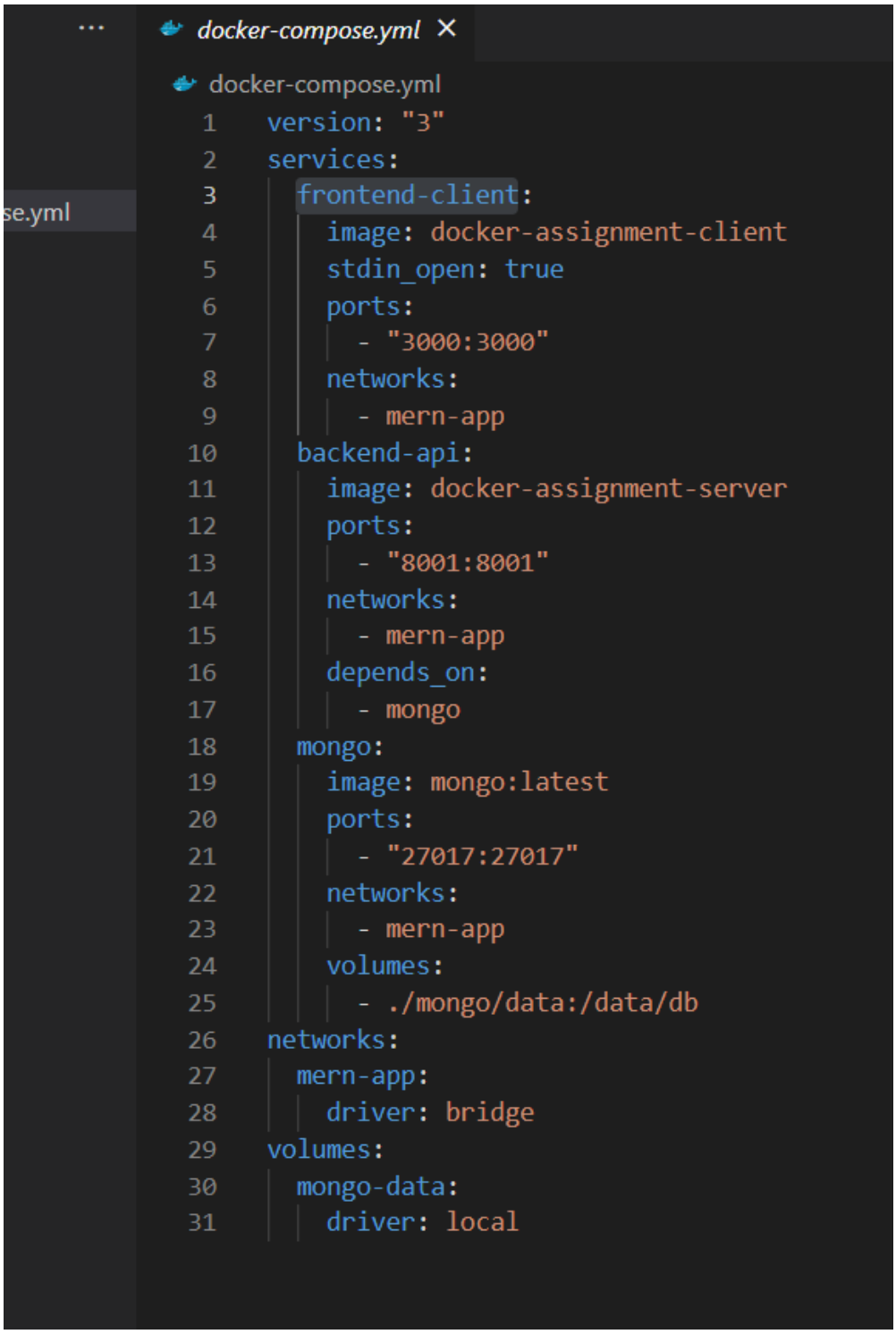

Figure 6: Docker-compose file

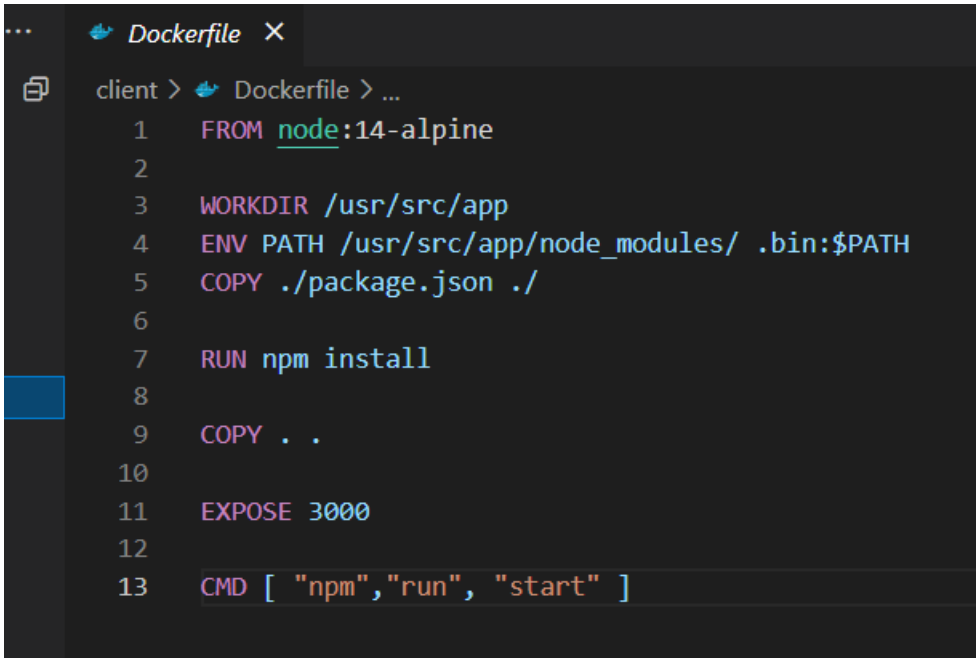

Figure 7: Dockerfile - Client