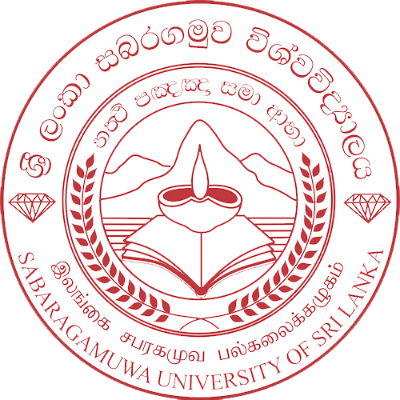
**SE6103 - PARALLEL AND DISTRIBUTED SYSTEMS**

****

**Assignment - 01**

**19APSE4276\_ V. Janarthan**

BSc (Honors) in Software Engineering

Department of Software Engineering

Faculty of Computing

Sabaragamuwa University of Sri Lanka

23rd December 2024

**Question 1**

1. **What is Docker, and why is it used?**

The open-source platform **Docker** allows developers to build, ship and run applications in containers. Containers allow encapsulation of any application with its dependencies and are portable and lightweight, ensuring a seamless experience regardless of the environment.

Docker's Key Features - Scalability, Portability, Isolation and Efficiency

Why you would use Docker -

1. Saves the user from dealing with the problem of this work on his machine. Developers can set up a persistent and identical development, testing and a production environment.
2. Bundling of dependencies and frameworks applications with docker reduces the hassle to set everything up before usage of the application.
3. Testing cycles along with development can be done at a greater speed since integration of one or more Docker containers allows for fast deployment.
4. From a cost perspective, deploying micro services on docker is advantageous as docker containers are cheaper and don't take up as many system resources as virtual machines.
5. Not only can docker be used for easy management of micro services, but it is also helpful in creating micro services since each service will run on its own container.
6. One of docker's key feature is the versioning of containers - It is easy to keep track of containers’ versions and if needed restore the previous version.
7. **Explain the difference between a Docker image and a Docker container.**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Docker Image** | **Docker Container** |
| Definition | A framework or an outline for making enclosures. | A docker image that is currently operating. |
| State | Fixed and permanent. | Fluid, able to be spun up, shut down or frozen at a point in time. |
| Purpose | Contains application logic along with its dependencies and its configurations. | Executes the software described in the specified image. |
| Storage | Is stored on a disk as a file. | Is stored in memory (volatile by nature). |
| Lifecycle | Constructed a single time, however, used many times afterwards. | Constructed via an image, executes then can be removed. |
| Examples | A step-by-step explanation on how to construct a cake. | A cake which has been cooked and is ready to be served. |

1. **What are the benefits of using Docker in software development?**

**1. Portability** - Containers built with Docker can work equally on diverse platforms including deployment and testing which hence clears the issue of “it works on my device”.

**2. Faster Development and Deployment** - Thanks to Docker, developers do not need to worry about the tedious work of packaging all dependencies because installation is faster and more trustworthy.

**3. Consistency and Standardization -** Applications in containers made with Docker work uniformly in different places which in return enhances accuracy and reduces the challenges of deployment.

**4. Isolation -** Since containers designed using Docker are independent the applications along with their dependencies do not interfere in another project.

**5. Resource Efficiency** - Since containers use the core of the source operating system, they are faster and easier to use than virtual machines.

**6. Scalability** - With tools like Kubernetes, scaling applications is simpler and faster, thanks to docker, and makes it possible to run several containers at once or distribute them over clusters.

**7. Simplified Dependency Management** - Docker wraps up all dependencies with the application to ease replication and remove manual configurations across all platforms.

**Question 2**

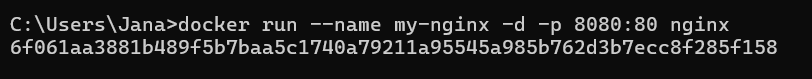
**Task 1: Pull and Run a Container**

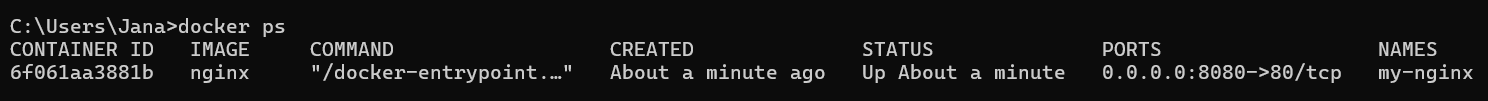
1. Pull the official Nginx image from Docker Hub.

A screenshot of a computer

Description automatically generated

1. Run a container named my-nginx using the pulled Nginx image.
2. Mapport8080 on your host machine to port 80 in the container.





1. Open a browser and verify Nginx is running by navigating to <http://localhost:8080>.

A screenshot of a computer

Description automatically generated

**Task 2: Inspect and Stop the Container**

1. Use appropriate commands to inspect the running container’s:

A black screen with white lines

Description automatically generated

A black screen with a black border

Description automatically generated

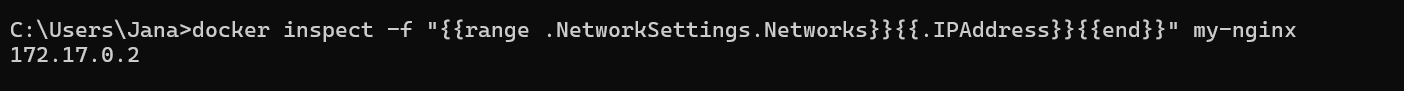
A screenshot of a computer

Description automatically generated

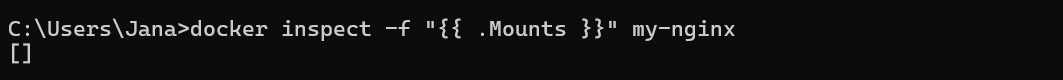
A black and white screen with white text

Description automatically generated

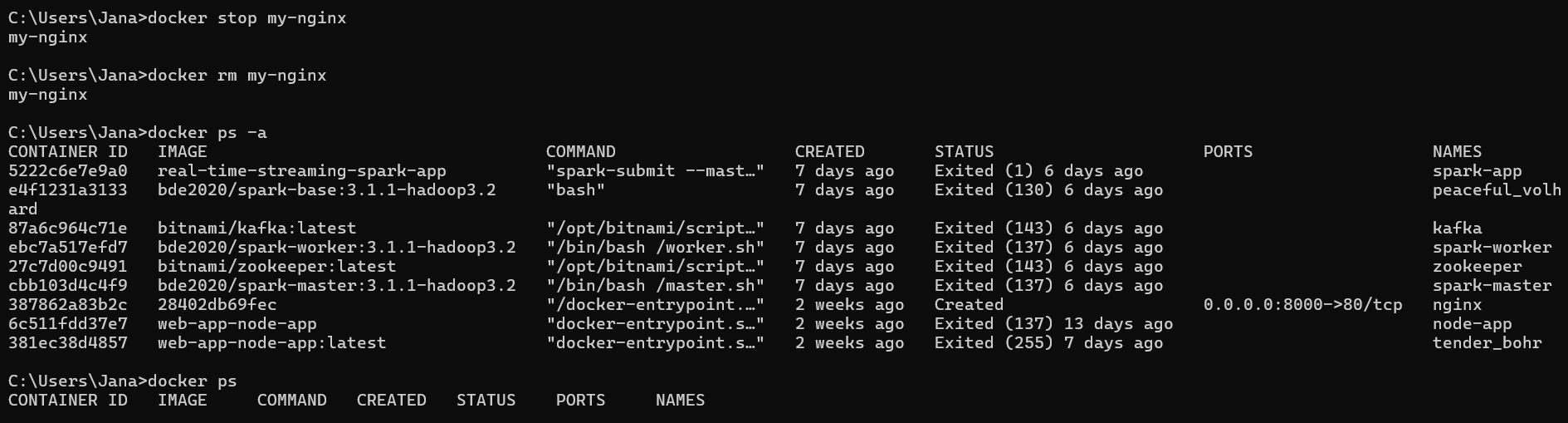
○ IP Address



○ Mountpoints (if any)



1. Stop and remove the my-nginx container.



**Question 3**

Task 3: Create a Custom Image

1. Create a folder named my-app. Inside it, create the following:

A screenshot of a computer

Description automatically generated

○ A file named Dockerfile.

○ An index.html file with the content:

A screenshot of a computer

Description automatically generated

1. Write a Dockerfile that:

○ Uses the official nginx:latest image as the base.

○ Copies your index.html file to the appropriate location inside the container.

A screenshot of a computer

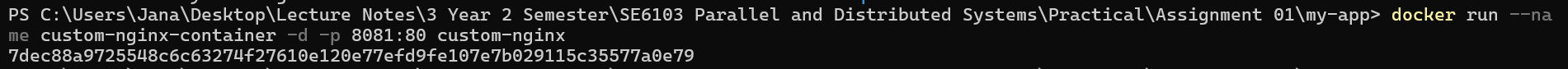
Description automatically generated

1. Build the image with the name custom-nginx.

A screenshot of a computer

Description automatically generated

1. Run a container from this image, mapping port 8081 to port 80 in the container.



1. Verify the container is running and accessible at <http://localhost:8081>

A screenshot of a computer

Description automatically generated