LS Lab 1: Containerization and application layer load balancing

In this lab, we will focus mainly on Docker engine and L7 load balancer, which will be covering the below topics:

- Docker images
- Docker containers
- Docker volumes
- Docker network
- Docker-compose
- Nginx Application Layer Load balancing

Task 1: Get familiar with Docker Engine

1. Pull *Nginx v1.23.3* image from *dockerhub* registry and confirm it is listed in local images
2. Run the pulled *Nginx* as a container with the below properties
    a. Map the port to 8080.
    b. Name the container as *nginx-<stX>.*
    c. Run it as daemon .

3. Confirm port mapping.
    a. List open ports in host machine.
    b. List open ports inside the running container.
    c. Access the page from your browser.
4. Create a Dockerfile similar to the below properties (let's call it container A).
    a. Image tag should be *Nginx v1.23.3.*
    b. Create a custom *index.html* file and copy it to your docker image to replace the Nginx default web page.
    c. Build the image from the *Dockerfile*, tag it during build as *nginx:<stX>*, check/validate local images, and run your custom made docker image.
    d. Access via browser and validate that your custom page is hosted.

Task 2: Work with multi-container environment

1. Create another Dockerfile similar to step 1.4 (Let's call it container B), and an *index.html* with different content.
2. Write a docker-compose file with the below properties
    a. Multi-build: Builds both Dockerfiles and runs both images.
    b. Port mapping: Container A should listen to port 8080 and container B should listen to port 9090. (They host two different web pages)
    c. confirm both websites are accessible
    d. Volumes: Mount (bind) a directory from the host file system to Nginx containers and update the contents of index.html in the host file system, re-deploy and confirm in the browser that the web page's content is updated.

3. **Configure L7 Loadbalaner**
    a. Install Nginx in the host machine or add a third container in the docker-compose that will act as loadbalancer, and configure it in front of two containers in a manner that it should distribute the load in a Weighted Round Robin approach.
    b. Access the page of Nginx ALB and validate, it is load-balancing the traffic (you see two different content per page reload).