**Lab Assignment 1: Docker Installation & Working with Images and Containers**

**Objective**

* Install Docker and verify the setup.
* Understand how to pull and run Docker images.
* Learn basic container management (start, stop, remove).

**Tasks**

1. **Docker Installation (10 Marks)**
   * **Install and Configure Docker (6 Marks):**
     + Install Docker on your preferred OS (Windows, macOS, or Linux).
     + Verify installation with docker version and docker info.
   * **Post-Installation Steps (4 Marks):**
     + Configure Docker to run without sudo (on Linux) or verify Docker Desktop settings (on Windows/Mac).
     + Document any challenges and solutions during installation.
2. **Pulling and Running Docker Images (10 Marks)**
   * **Pull a Base Image (4 Marks):**
     + Use docker pull to download a popular image (e.g., ubuntu, nginx, or alpine).
     + Verify the pulled image with docker images.
   * **Running Containers (6 Marks):**
     + Start a container from the pulled image (docker run -it ubuntu, for example).
     + Perform a simple task inside the container (e.g., install a package, print a message).
     + Show how to stop and remove the container when finished.
3. **Basic Container Management (10 Marks)**
   * **Inspecting Containers (5 Marks):**
     + Use docker ps, docker ps -a, and docker inspect to gather container details.
     + Show how to list running containers and retrieve specific info (e.g., container IP).
   * **Cleaning Up (5 Marks):**
     + Remove unused containers (docker rm) and images (docker rmi) to free space.
     + Explain best practices for housekeeping.

**Deliverables**

* **Documentation** describing the installation process, commands used, and any troubleshooting steps.
* **Screenshots or terminal logs** showing Docker installation, pulled images, running containers, and container inspection.

**Marks Breakdown (Total 30)**

* **Docker Installation & Configuration:** 10
* **Pulling & Running Images:** 10
* **Container Management & Documentation:** 10

**Lab Assignment 2: Dockerfile & Containerizing Applications**

**Objective**

* Learn how to create and use a Dockerfile.
* Containerize a simple application using Docker.
* Practice building custom images and running them.

**Tasks**

1. **Creating a Dockerfile (10 Marks)**
   * **Simple Application Setup (6 Marks):**
     + Choose a small application (e.g., a simple web server in Python, Node.js, or a static HTML page).
     + Write a Dockerfile that copies your app code into the image and runs it on container startup.
   * **Dockerfile Instructions (4 Marks):**
     + Use at least three instructions (e.g., FROM, COPY, RUN, CMD, or EXPOSE).
     + Explain each instruction in code comments or a separate doc.
2. **Building and Running Custom Images (10 Marks)**
   * **Image Build (6 Marks):**
     + Run docker build -t <your\_image\_name> . to build an image from your Dockerfile.
     + Verify the image creation with docker images.
   * **Container Deployment (4 Marks):**
     + Run the container from the newly built image (docker run -d -p 8080:80 <your\_image\_name>).
     + Confirm the application is accessible (e.g., open http://localhost:8080 in a browser).
3. **Persisting Data and Exploring Container Logs (10 Marks)**
   * **Volume Mount or Basic Persistence (5 Marks):**
     + Demonstrate a simple volume mount or data persistence scenario (e.g., store logs or app data in a volume).
     + Show how to inspect or read data from the host.
   * **Logs & Debugging (5 Marks):**
     + Use docker logs to view container output.
     + Illustrate how logs can be used to troubleshoot issues (e.g., missing dependencies).

**Deliverables**

* **Dockerfile** used for building the image.
* **Documentation** (or well-commented Dockerfile) explaining each instruction.
* **Screenshots or logs** showing the custom image running and any volume mounts.

**Marks Breakdown (Total 30)**

* **Dockerfile Creation & Explanation:** 10
* **Image Build & Container Deployment:** 10
* **Persistence & Logs:** 10

**Lab Assignment 3: Persistent Storage, Networking, and Docker Hub**

**Objective**

* Deepen understanding of Docker’s networking model.
* Implement persistent storage using volumes and port binding.
* Publish images to Docker Hub (or a private registry if preferred).

**Tasks**

1. **Docker Networking & Port Binding (10 Marks)**
   * **Port Binding Basics (5 Marks):**
     + Run a containerized web application with port mapping (-p 8080:80).
     + Verify you can access it externally on http://localhost:8080.
   * **Network Creation (5 Marks):**
     + Create a user-defined network (docker network create <network\_name>).
     + Run two containers on this network (e.g., a web server and a database).
     + Show how they communicate using container names instead of IPs.
2. **Persistent Storage (10 Marks)**
   * **Docker Volumes (6 Marks):**
     + Create and attach a named volume to a container.
     + Demonstrate how data persists after the container is removed.
     + Show how to list and inspect volumes (docker volume ls, docker volume inspect).
   * **Docker Linking or Monitoring (4 Marks):**
     + If desired, use Docker linking (deprecated but still educational) or a monitoring tool (like cAdvisor) to illustrate container relationships or resource usage.
3. **Docker Hub & Publishing Images (10 Marks)**
   * **Docker Hub Account (4 Marks):**
     + Create a Docker Hub account (if not already).
     + Log in via CLI using docker login.
   * **Publishing an Image (6 Marks):**
     + Tag your custom image with your Docker Hub username (e.g., docker tag myapp <username>/myapp:v1).
     + Push the image to Docker Hub (docker push <username>/myapp:v1).
     + Verify it appears in your Docker Hub repository.

**Deliverables**

* **Documentation** detailing the networking setup, volume usage, and Docker Hub steps.
* **Screenshots/Logs** for network creation, volume inspection, and the published image on Docker Hub.

**Marks Breakdown (Total 30)**

* **Networking & Port Binding:** 10
* **Persistent Storage & Volumes:** 10
* **Docker Hub Publishing:** 10

**General Submission Guidelines for All Assignments**

1. **Code and Configuration Files**
   * Provide Dockerfiles, scripts, or any configuration files in a well-organized folder structure.
2. **Documentation**
   * Include a brief write-up or README explaining:
     + The steps taken for each task.
     + Any issues encountered and how you solved them.
3. **Verification and Screenshots**
   * Include terminal outputs or screenshots showing successful container runs, volume mounts, network configs, etc.
4. **Good Practices**
   * Use clear naming conventions for images and containers.
   * Follow best practices for Dockerfile instructions (e.g., minimal base images, proper layering).