

Practical 9

AIM: Demonstration of virtualization using Docker Container

DOCKER

- Docker is a popular platform for virtualization using containerization.
- It allows you to create, deploy, and manage lightweight, portable containers that run applications and their dependencies consistently across different environments
- Docker containers include all dependencies (frameworks, libraries, etc.) to run an application in an efficient and bug-free manner.
- Docker Containers have the following benefits:
 1. Light-weight
 2. Applications run in isolation
 3. Occupies less space
 4. Easily portable and highly secure
 5. Short boot-up time


Step-by-Step demonstration of virtualization using Docker

Step 1: Install Docker First, ensure you have Docker installed on your system. You can download and install Docker Desktop for Windows or macOS from the official Docker website. For Linux users, you can follow the instructions for your specific distribution.

Step 2: Pull a Docker Image Docker images are the blueprints for creating containers. Let's start by pulling a simple image, like the official Nginx web server image:

Open a terminal (or command prompt) and run the following command to pull the Nginx image:

```
bash
```

 Copy code

```
docker pull nginx
```

Step 3: Run a Docker Container Now that you have the Nginx image, you can create a container based on it. Run the following command to start an Nginx container:

```
bash Copy code  
  
docker run -d -p 8080:80 nginx
```

This command tells Docker to run the Nginx container in the background (-d), and it maps port 8080 of your host system to port 80 of the container (-p 8080:80).

Step 4: Access the Container You can access the Nginx web server running inside the Docker container by opening your web browser and navigating to <http://localhost:8080>.

Step 5: View Running Containers To see the list of running containers, run:

```
bash Copy code  
  
docker ps
```

Step 6: Stop and Remove the Container When you are done, you can stop the Nginx container by running:

```
bash Copy code  
  
docker stop <CONTAINER_ID>
```

Replace <CONTAINER_ID> with the actual ID of the running container (you can get the ID from the docker ps command).

To remove the container, run:

```
bash Copy code  
  
docker rm <CONTAINER_ID>
```

Step 7: Clean Up

If you don't plan to use the Nginx image anymore, you can remove it:

```
bash Copy code  
  
docker rmi nginx
```

These are the basic steps to demonstrate virtualization using Docker containers. Docker offers much more functionality, such as creating custom images, managing volumes for persistent data, and orchestration using tools like Docker Compose or Kubernetes. It's a powerful tool for building, shipping, and running applications in a consistent and isolated environment.