Sicat, Sean Russel F. IV-ACSAD

Study Plan: Introduction to Docker and Containerization

Goal

By the end of this study plan, the student should understand:

- The concept of containerization and how it differs from virtualization.
- Docker's history and role in modern DevOps.
- How to build, deploy, and manage Docker containers.
- Best practices in using Docker for development and production.

Day 1: Introduction to Containerization

Topics:

- What are images and containers?
- Virtual Machines vs Containers
- Why containers are important in modern software deployment

Activities:

- Watch: "Containers Explained in 5 Minutes" (YouTube)
- Read: *Docker Docs → What is a Container?*
- Compare system resource usage between VMs and containers.

Outcome:
Understand the concept and advantages of containers.

Day 2: History of Docker and Containerization

Topics:

- Evolution of container technology:
    - **Chroot (1979) → FreeBSD Jails (2000) → LXC (2008) → Docker (2013)**
- How Docker popularized containerization.

- Docker's impact on CI/CD, DevOps, and cloud computing.

Activities:

- Read: *"A Brief History of Containers" (Docker Blog or Medium)*
- Write a short summary: *"How Docker changed software deployment."*

Outcome:
Be able to explain the historical evolution and Docker's key innovations (images, registry, portability).

Day 3: Docker Basics

Topics:

- Installing Docker (Windows, macOS, or Linux)
- Key components:
    - Docker Engine
    - Docker Images
    - Docker Containers
    - Docker Hub

Activities:

- Install Docker Desktop.
- Run your first container:
- docker run hello-world
- Explore Docker Hub for available images.

Outcome:
Successfully run and understand a basic Docker container.

Day 4: Working with Docker Images and Containers

Topics:

- Creating custom Docker images using a **Dockerfile**
- Docker commands:
    - docker build, docker run, docker ps, docker stop, docker rm
- Understanding container lifecycle

Activities:

- Build a simple Node.js or Python app and containerize it.
- Run multiple containers simultaneously.

Outcome:
Be able to build and run custom images.

Day 5 : Docker Networking and Volumes

Topics:

- Docker networks: bridge, host, overlay
- Data persistence using volumes and bind mounts

Activities:

- Create a container that saves data to a volume.
- Connect two containers using Docker network.

Outcome:
Understand how containers communicate and store persistent data.

Day 6: Docker Compose and Multi-Container Apps

Topics:

- Introduction to **Docker Compose**
- Defining services in docker-compose.yml
- Running multi-container environments

Activities:

- Build a simple app with a backend and database using Compose.
- Commands: docker-compose up, docker-compose down

Outcome:
Deploy a multi-container app using Docker Compose.

Day 7: Best Practices and Final Review

Topics:

- **Best Practices for Docker and Containerization:**
    1. Use small, efficient base images (e.g., alpine).
    2. Minimize layers in Dockerfile.
    3. Use .dockerignore to reduce image size.
    4. Tag images properly (avoid using latest in production).
    5. Use environment variables for configuration.
    6. Scan images for vulnerabilities regularly.
    7. Implement logging and monitoring for containers.
    8. Avoid running containers as root.

Activities:

- Review key Docker commands and concepts.
- Create a short presentation or report summarizing:
    - History of Docker
    - How it works
    - Best practices

Outcome:
Be confident in explaining Docker concepts and applying best practices.

References

- **https://docs.docker.com/**