

ONE PAGE SUMMARY

IT & Software > Docker > Images and Containers

Introduction to Images and Containers

Master Docker Images and Containers for Efficient Development!

DEFINITION

Docker images are lightweight, standalone packages that contain everything needed to run software. Containers are the running instances of these images, providing an isolated environment for applications.

KEY CONCEPTS

Docker Image

A Docker image is a read-only template used to create containers, including the application code and dependencies.

Docker Container

A container is a runnable instance of a Docker image, providing a lightweight and isolated environment for applications.

Layering

Docker images are built in layers, allowing for efficient storage and reuse of shared components.

Dockerfile

A Dockerfile is a script containing a series of commands to build a Docker image.

Registry

A registry is a storage location for Docker images, such as Docker Hub, where images can be shared and downloaded.

EXAMPLES

- Creating a web server container from an Nginx image.
- Building a custom application image using a Dockerfile.
- Pulling a database image from Docker Hub to run locally.

MEMORY TIPS

- ★ Think of images as blueprints and containers as the buildings constructed from them.
- ★ Remember 'Layer Cake' for image layering in Docker.
- ★ Use 'Dockerfile' as a recipe for building your image.

COMMON MISTAKES

- ✗ Confusing images with containers; remember they are different entities.
- ✗ Neglecting to update images, leading to outdated dependencies.
- ✗ Not using Docker volumes for persistent data storage.

QUICK RECAP

Docker images are templates for creating containers, which run applications in isolated environments. Understanding the relationship between images and containers is crucial for effective Docker usage.