ONE PAGE SUMMARY

IT & Software › Docker › Images and Containers

Introduction to Images and Containers

Master Docker: Understand Images and Containers!

DEFINITION

> Docker images are templates for creating containers, which are lightweight, portable environments for applications. Containers run applications in isolation, ensuring consistency across different environments.

KEY CONCEPTS

Docker Image

A Docker image is a read-only template used to create containers, containing everything needed to run an application.

Docker Container

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

Layering

Docker images are built in layers, allowing for efficient storage and faster builds by reusing unchanged layers.

Registry

A Docker registry is a repository for storing and sharing Docker images, with Docker Hub being the most popular public registry.

Isolation

Containers provide process isolation, ensuring that applications run independently without interference from one another.

EXAMPLES

→ Creating a web server container from an Nginx image.

→ Running a Python application in a dedicated container.

→ Using a MySQL image to set up a database container.

MEMORY TIPS

★ Think of images as blueprints and containers as the buildings constructed from them.

★ Remember 'Layer, Run, Share' to recall the image creation and container usage process.

★ Visualize containers as separate rooms in a house, each running its own application.

COMMON MISTAKES

✗ Confusing images with containers; images are templates, while containers are instances.

✗ Neglecting to manage container resources, leading to performance issues.

✗ Forgetting to clean up unused images and containers, which can waste disk space.

QUICK RECAP

Docker images are the blueprints for creating containers, which run applications in isolated environments. Understanding the relationship between images and containers is crucial for efficient application deployment and management.