ONE PAGE SUMMARY

IT & Software › Docker › Docker Basics

# Introduction to Docker Basics

Unlock the Power of Containerization with Docker Basics!

## DEFINITION

> Docker is a platform that allows developers to automate the deployment of applications in lightweight containers. It simplifies software development by ensuring consistent environments across different systems.

## KEY CONCEPTS

### Containers

Containers are lightweight, standalone, and executable packages that include everything needed to run a piece of software.

### Images

Images are read-only templates used to create containers, containing the application code and dependencies.

### Dockerfile

A Dockerfile is a script that contains a series of instructions on how to build a Docker image.

### Docker Hub

Docker Hub is a cloud-based repository where users can share and manage Docker images.

### Volumes

Volumes are used to persist data generated by and used by Docker containers, allowing data to survive container restarts.

## EXAMPLES

→ Creating a Docker container for a Node.js application.

→ Using Docker to run a MySQL database in isolation.

→ Building a custom Docker image from a Dockerfile.

## MEMORY TIPS

★ Think of containers as lightweight boxes that hold everything an app needs.

★ Remember 'Image = Template' for creating containers.

★ Use 'Dockerfile = Recipe' to recall how to build images.

## COMMON MISTAKES

✗ Forgetting to expose necessary ports in the Dockerfile.

✗ Neglecting to manage container data with volumes.

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

## QUICK RECAP

Docker simplifies application deployment through containerization, ensuring consistency across environments. Key components include containers, images, and Dockerfiles, which work together to streamline development.