ONE PAGE SUMMARY

IT & Software › Docker › Docker Best Practices

Introduction to Docker Best Practices

Master Docker with Essential Best Practices for Efficiency.

DEFINITION

> Docker best practices are guidelines to optimize containerized applications for performance and maintainability. They help developers create efficient, secure, and scalable Docker images and containers.

KEY CONCEPTS

Minimal Images

Use the smallest base images possible to reduce size and improve security.

Layer Caching

Organize Dockerfile commands to leverage layer caching, speeding up builds.

Environment Variables

Utilize environment variables for configuration to enhance flexibility and security.

Multi-Stage Builds

Use multi-stage builds to keep final images lightweight by separating build and runtime dependencies.

Health Checks

Implement health checks to monitor container status and ensure reliability.

EXAMPLES

→ FROM alpine:latest for a minimal base image.

→ RUN apt-get update && apt-get install -y package for efficient layer caching.

→ ENV NODE_ENV=production to set environment variables.

MEMORY TIPS

★ Think 'Less is More' for minimal images.

★ Remember 'Check Your Health' for health checks.

★ Use 'Stages for Success' to recall multi-stage builds.

COMMON MISTAKES

✗ Neglecting to clean up unused images and containers.

✗ Not using .dockerignore files, leading to bloated images.

✗ Hardcoding sensitive information instead of using environment variables.

QUICK RECAP

Docker best practices enhance application performance and security. Focus on minimal images, layer caching, and health checks for optimal results.