DOCKER

By : LAKSHMIKANT DESHPANDE

Introduction to Docker

- Platform for developing, shipping, and running applications in containers.
- Enables environment consistency, scalability, and portability.
- Key benefit: Run the same app anywhere (developer machine, cloud, production).

Docker vs Virtual Machines

Containers: Lightweight, share OS kernel, fast startup.

VMs: Full OS, heavy, slower to start.

Docker = faster, more efficient.

Key Docker Components

Docker Engine: Runs containers.

Docker Images: Read-only templates for creating containers.

Docker Containers: Running instances of images.

Docker Hub: Central image repository.

Docker Compose: Multi-container app management.

How Docker Works

Images: Build once, run anywhere.

Containers: Isolated environments running applications.

Dockerfile: Defines how to build an image.

Volumes: Store persistent data outside containers.

Docker in Cloud-Based Architectures

Supports **cloud-native** apps (microservices, scalability).

Works with managed services (e.g., **AWS ECS**, **Google GKE**, **Azure AKS**).

Enables **CI/CD** pipelines for automated testing and deployment.

Docker Use Cases

Development/Testing: Replicate production environments locally.

Microservices: Isolated, scalable services.

Scaling: Run and replicate containers to handle traffic spikes.

Serverless: Docker images for serverless functions (e.g., AWS Lambda)

Docker Best Practices

Optimizing Images: Reduce size using multi-stage builds.

Security: Use official images, scan for vulnerabilities.

Networking: Use overlay networks for secure communication.

Monitoring: Leverage tools like **Prometheus** and **Grafana**.

Docker Swarm vs Kubernetes

Docker Swarm: Native orchestration; simple to use, less complex.

Kubernetes: Advanced orchestration, larger ecosystem, more features.

Both support container scaling and management.

Real-World Use Cases

Netflix: Containers for microservices, high scalability.

Spotify: Dockerized backend, simplifies deployment.

Airbnb: Docker for consistent dev/test environments across teams.