Chapter-1

Container is an encapsulated process that includes the required runtime dependencies for a program to run. In container application-specific libraries are independent of the host operating system libraries.

immutable : unchanging over time
ephermal : lasting for a very short time.

Containers use cgroups for resource management, such as CPU time allocation and system memory

Containers use namespaces to isolate processes within containers from each other and from the host system.

Envirorment within a container is linux-based. regardless of the host O.S.

linux-specific features are often virtualized by container engine.

OCI => Open Container Initiative.

Container image: ① immutable data of app
② libraries.

One C. image can create multiple C. instances / containers.

Containers have smaller footprint and start and stops faster.

Virtual Machines	Containers
Machine level functionality → Hypervisor	Container engine
Management → VM management interface	Container engine/ Orchestration software
Virtulization level → fully virtulized env.	Only relevant parts
→ Size in gb	Size im mb
Portability → Only same hypervisor	→ Any OCI-compliant engine.

Red Hat OpenShift Container Platform (RHOCP)

→ Manages both containers & virtual Machines from a common interface.

Software applications requires specific versions of libraries to be available for deployment, which can result in dependency issues or specific OS requirements.
As Container image includes libraries no worry of dependencies are there during deployment.

Kubernetes is an orchestration service that simplifies the deployment, management and scaling of containerized applications.
Manages complex pools of resources such as CPU, RAM, storage & networking.
Kubernetes provides high uptime & fault tolerance for container app.

Smallest unit in kubernetes ~~in~~ is a pod
Pod may consist one or more containers