

Containers

- **What is a Container?**

- Lightweight and portable.
- OS-level virtualization (compared to VMs).
- Runs applications and dependencies in an isolated environment.

- **Container Runtime:**

- OCI standards for container image format.
- Examples: `containerd`, `CRI-O`, `runc`, Docker.
- How Kubernetes interacts with container runtimes through the Container Runtime Interface (CRI).

- **Container Images:**

- Layers and image composition.
- Building images using Dockerfile.
- Storing and retrieving images from registries (Docker Hub, Artifact Registry, etc.).

- **Container Lifecycle:**

- Creation, running, stopping containers.
- Restart policies in Kubernetes (Always, OnFailure, Never).

- **Security:**

- Image scanning and vulnerability management.
- Running containers with least privilege (user ID and group).
- Container isolation using namespaces and cgroups.

- **Networking:**

- Container Network Interface (CNI) plugins.
- Port mapping and exposing services.
- Container-to-container and container-to-host communication.

- **Storage:**

- Ephemeral storage (temporary storage for containers).
- Persistent storage (using volumes in Kubernetes).
- Mounting and sharing volumes between containers.

Revision #2

Created 18 November 2024 21:51:01 by Admin

Updated 21 November 2024 19:56:36 by Admin