

# docker basics

---

## various definitions

---

flashcards:

- **title:** "What are Containers"  
**definition:** "Containers are lightweight, executable units that package up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another."  
**code\_example:** |  
# No direct code example for the concept of containers.
- **title:** "Need for Containers"  
**definition:** "Containers provide a consistent environment for applications to run in isolation from other processes. They are portable, more resource-efficient than virtual machines, and ensure that software runs reliably when moved from one environment to another."
- **title:** "Bare Metal vs VM vs Containers"  
**definition:** |  
"Bare Metal refers to the physical hardware. VMs (Virtual Machines) are software emulations of physical computers that include a full copy of an operating system. Containers, on the other hand, share the host system's kernel and isolate the application processes from the rest of the system."  
**code\_example:** |  
# No direct code example for comparison.
- **title:** "Docker and OCI"  
**definition:** "Docker is a set of platform-as-a-service products that use OS-level virtualization to deliver software in packages called containers. OCI (Open Container Initiative) is a project under the Linux Foundation to design open standards for containers, to ensure interoperability."  
**code\_example:** |  
# No direct code example for Docker and OCI.
- **title:** "Namespaces"  
**definition:** "Namespaces are a feature of the Linux kernel that partitions kernel resources so that one set of processes sees one set of resources while another set of processes sees a different set of resources."  
**code\_example:** |  
# Namespaces are used by Docker internally and typically not manipulated directly via code.
- **title:** "cgroups"  
**definition:** "Control Groups (cgroups) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes."  
**code\_example:** |  
# Cgroups are used by Docker internally and usually not handled directly in user code.
- **title:** "Union Filesystems"

**definition:** "Union filesystems operate by creating layers, making them very lightweight and fast. Docker uses union filesystems to provide the building blocks for containers, images, and storage components."

**code\_example:** |

```
# Union Filesystems are part of Docker's internal mechanics.  
FROM ubuntu:18.04  
RUN touch /example.txt  
# This creates layers in a union filesystem.
```

- **title:** "Docker Engine"

**definition:** "Docker Engine is the underlying client-server technology that builds and runs containers using Docker's components and services."

**code\_example:** |

```
# Install Docker Engine  
$ curl -fsSL https://get.docker.com -o get-docker.sh  
$ sh get-docker.sh
```

- **title:** "Data Persistence in Docker"

**definition:** "Data persistence in Docker refers to the mechanism that allows data to be stored in a way that it persists beyond the life of the container, typically using volumes or bind mounts."

**code\_example:** |

```
# Create a volume for persistence  
$ docker volume create my-vol
```

- **title:** "Ephemeral FS"

**definition:** "An ephemeral filesystem in Docker is a temporary storage that is created when a container starts and destroyed when the container stops. It's used for data that doesn't need to persist after the container is gone."

**code\_example:** |

```
# Container with an ephemeral filesystem  
$ docker run --rm -d my-image
```

- **title:** "Volume Mounts"

**definition:** "Volume mounts in Docker are a way to persist data stored in volumes which are managed by Docker and exist independently of the active life of containers."

**code\_example:** |

```
# Running a container with a volume mount  
$ docker run -d -v my-vol:/data my-image
```

- **title:** "Bind Mounts"

**definition:** "Bind mounts are a type of mount that allows you to store data on the host system outside the Docker managed volume system. They are often used for development purposes."

**code\_example:** |

```
# Running a container with a bind mount  
$ docker run -d -v /my/data:/data my-image
```

