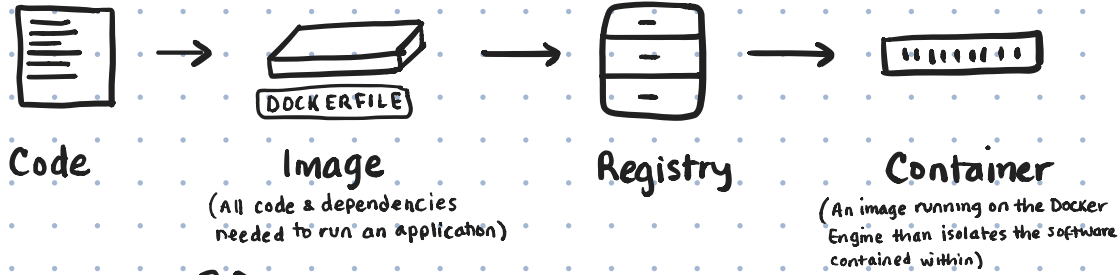




< BASIC DOCKER WORKFLOW >



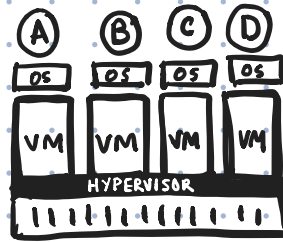
< THE HISTORY - WHY CONTAINERS ? >

1 App Per Server



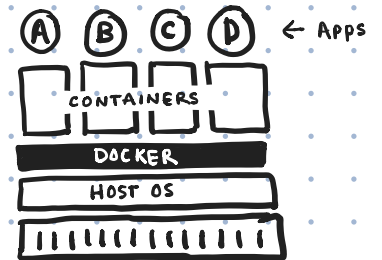
- Expensive
- Underutilized hardware
- Difficult to manage

VMware: Multiple Apps per server



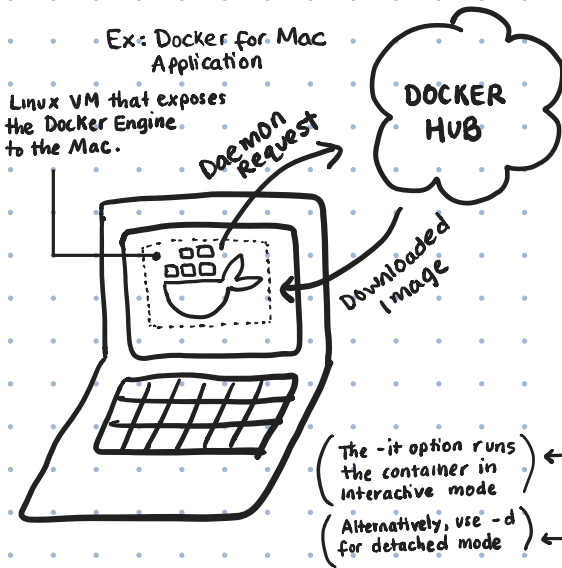
- Requires multiple virtual machines & operating systems = resource hogs!
- ★ VMs are an abstraction at the physical hardware level. ★

Docker: 1 App per container & Multiple containers per server



- Apps start much faster because VMs & OS are already running.
- ★ Containers virtualize the host OS rather than the underlying hardware. ★

< HOW IT WORKS >



- ↳ Users issue commands via CLI to the Docker Client
- ↳ The Docker client controls the Docker Daemon (via a REST API)
- ↳ The Docker daemon listens for API requests & manages images, containers, etc.
- ↳ By default, Docker is configured to use Docker Hub as its default registry.

↓
Repository for public & private images.

< COMMON COMMANDS >

- >> `docker version` → client & server configuration details
- >> `docker info` → System-wide info about the installation, including registry, # of containers & images.
- >> `docker pull image` → Download an image from registry
- >> `docker images` → Show all images on system
- >> `docker rmi tag/id` → Removes an image by tag or id
- >> `docker search foo` → Search Docker Hub for images
- >> `docker start/stop/rm container` → Start, stop or remove a container
- >> `docker build` → Build an image from a Dockerfile
- >> `docker ps -a` → List all containers
- >> `docker attach container` → View & control a running container
- >> `docker exec container command` → Execute a command within a running container
- >> `docker push image` → Share image to registry