Carnegie Mellon University
Master of
Software Engineering

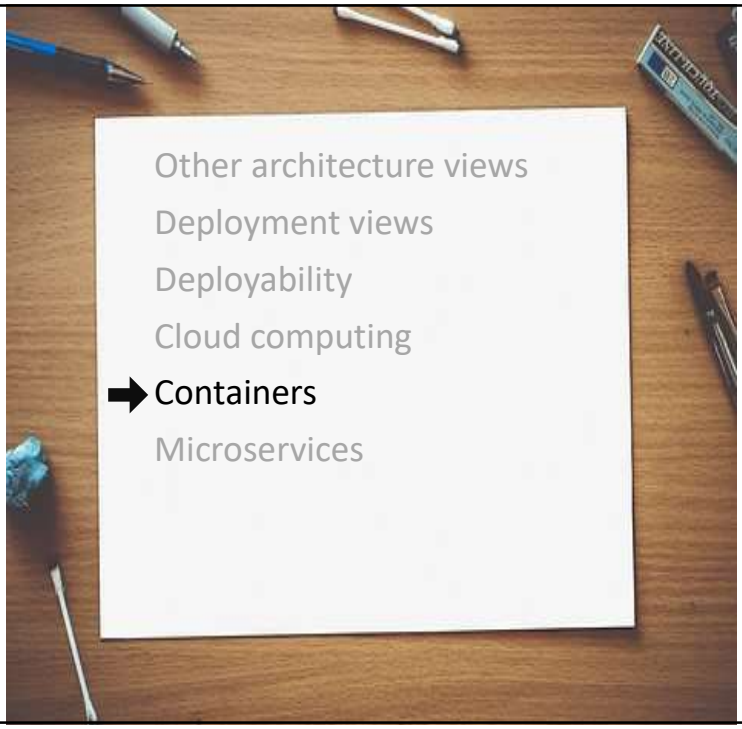
Deployment views – containers

LG Architecture Training Program

Paulo Merson

1

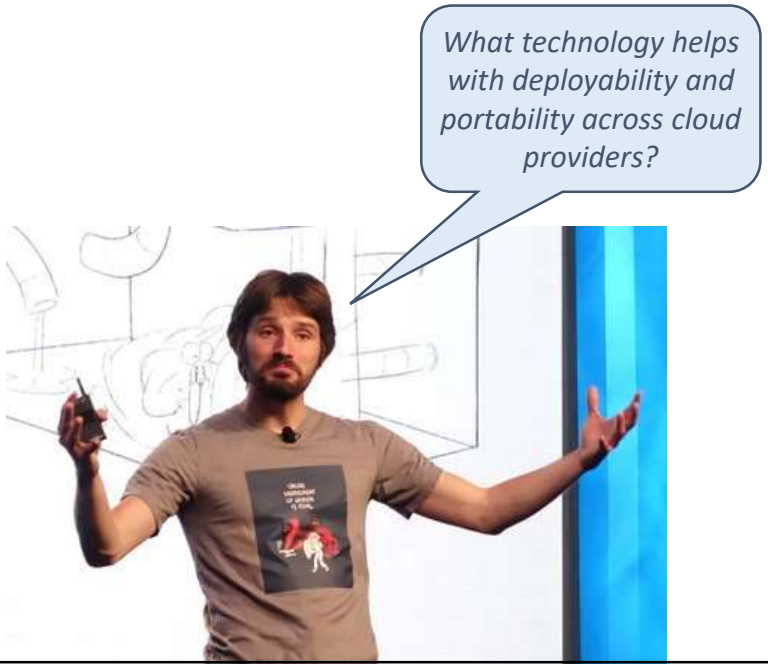
Agenda



- Other architecture views
- Deployment views
- Deployability
- Cloud computing
- ➡ Containers
- Microservices

2 Carnegie Mellon University

2



3

Software containers

- Docker was initially inspired by Linux containers (LXC)
- It quickly became the *de facto* standard from which *de jure* standards were derived
  - OCI image-spec
  - OCI runtime-spec

Docker is a tool to create, run and manage lightweight virtual machines called containers. [Ruka14]

docker

OPEN CONTAINER INITIATIVE

4

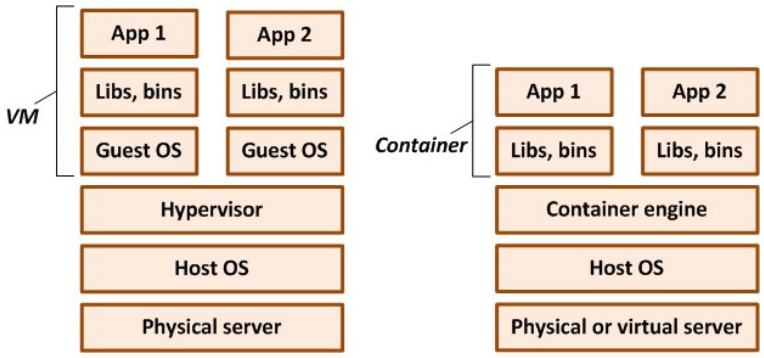
Docker container

- A docker container is a component that contains:
  - Application code + libraries
  - File system
  - Its own subnet (host and ports)
  - OS kernel (shared with other containers)
  - Runtime engine/VM, tools, middleware, and all else needed to run the application

Typically each application runs in its own container

Does the container really have everything needed to run the application?

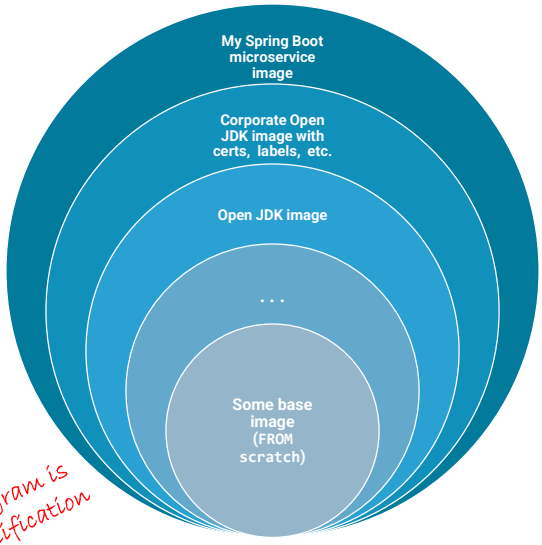
Virtualization vs containerization



A key difference is that in general VMs are perennial and containers are ephemeral

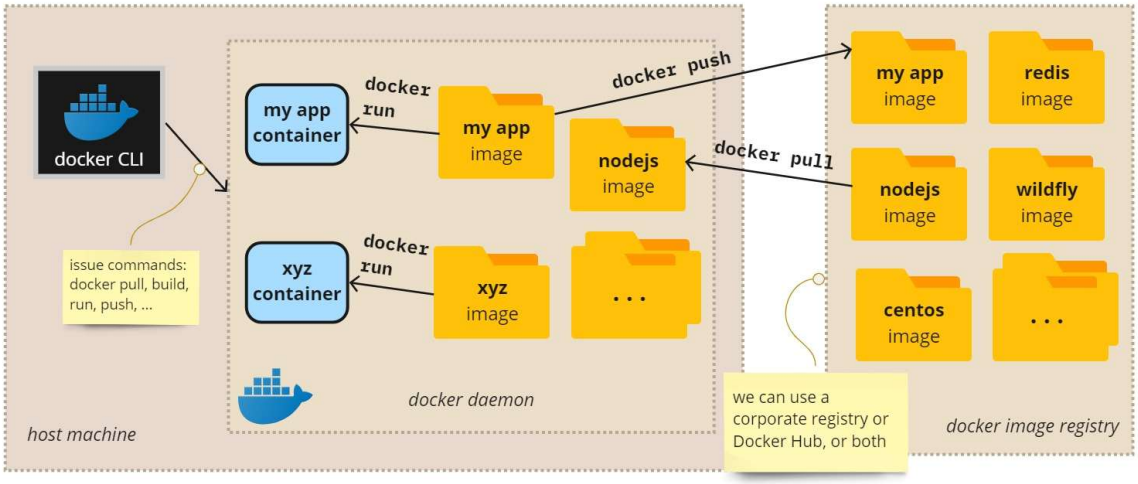
Docker layers

- Docker images are created in layers
- Typically, you create the image for your microservice from a parent image
- Often the parent image comes from docker hub (was not created by you)

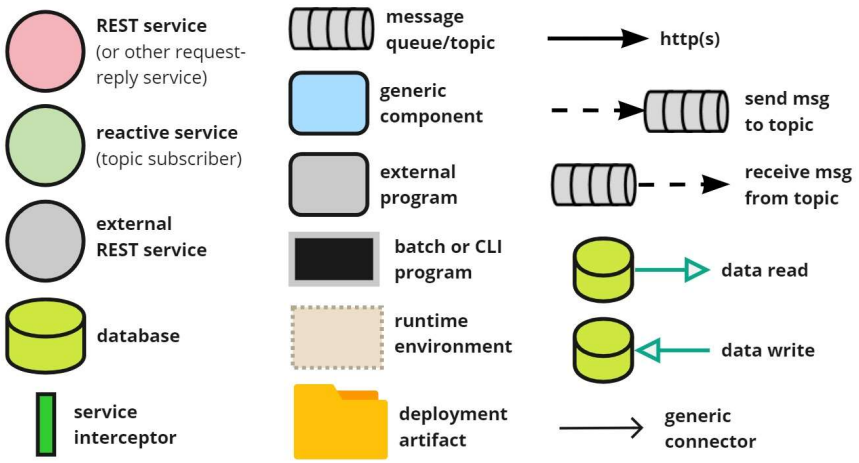


The diagram is a simplification

Docker overview



Notation key for (most) diagrams showing containerization and microservices



Containers – benefits

- Portability across different platforms and cloud providers
- Scalability due to improved ability to apply passive redundancy
- Low resource consumption (compared to VMs)
- Deployability
  - Docker registry makes distribution easier
  - Embedded versioning support
  - The application is packaged with its runtime environment

Containerization is a DevOps enabler

Containers – challenges

- Creating slim images becomes a new requirement
- Access to some host machine resources is limited
- There can be difficulties if your application does not run on Linux
- We need to coordine a myriad containers and container instances
  - Access to shared resources requires synchronization
  - Container lifecycle and scalability should be automatic
  - Communication between tightly-coupled containers should be optimized

What technology extends containers to deal with these coordination challenges?

Container orchestration

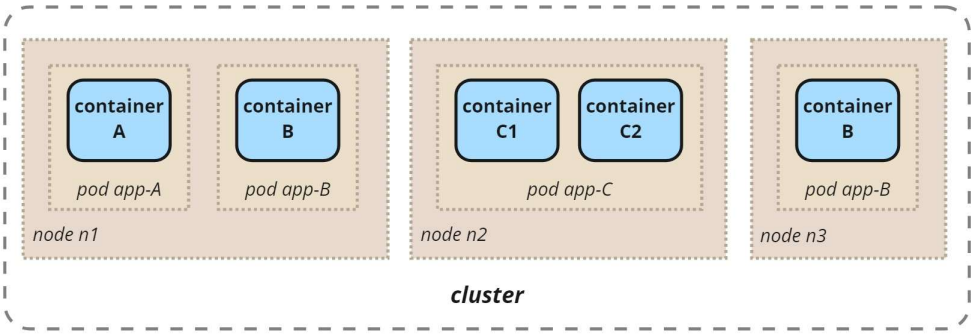
- There might be hundreds of components published as containers
- A container orchestration platform
  - can redeploy a container as soon as its image is updated in the registry
  - allows configuring URL routing, replication, security, and other execution parameters
  - oversees the health of containers
  - can automatically scale in and out microservices and load-balance requests
  - can provision shared resources to containers (e.g., persistent storage)

kubernetes

Kubernetes (K8S) has become the de facto standard for container orchestration

Containers in pods

- Container orchestration allows a group of containers to run together in a "pod"
- Pods are assigned to nodes in a cluster
- Nodes can be VMs or physical host machines



Questions?

Paulo Merson
pmerson@cmu.edu