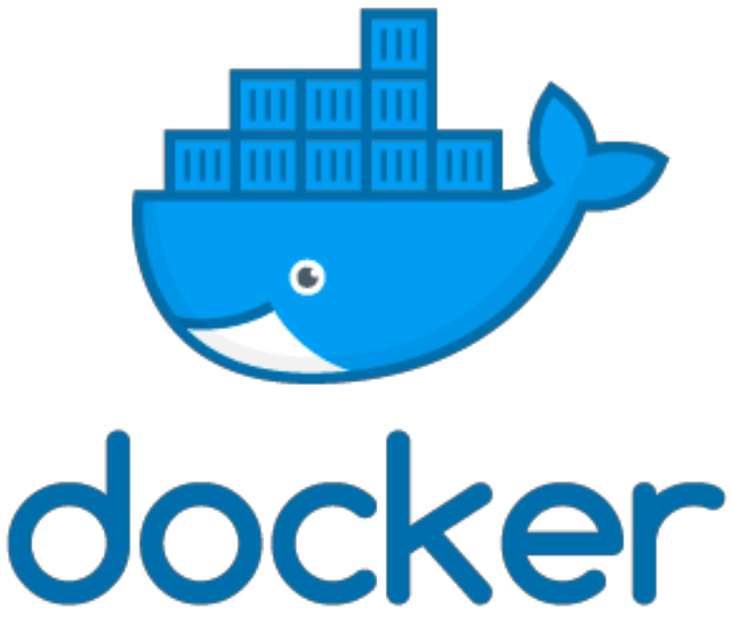
Docker

Nebojsa Krtolica
outside collaborator @pannonia-expertise.com


docker

Docker – Motivation

Automated Builds for all environments

“Build once run anywhere” - portability mantra

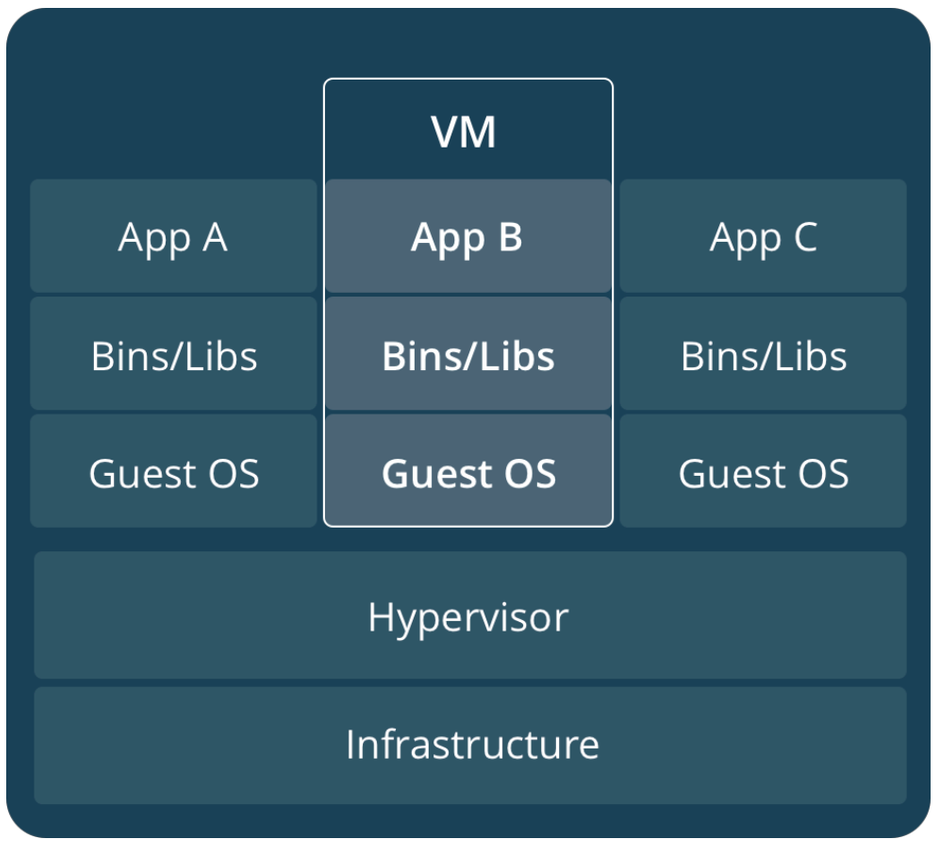
Combined strengths from VM and Bare Metal

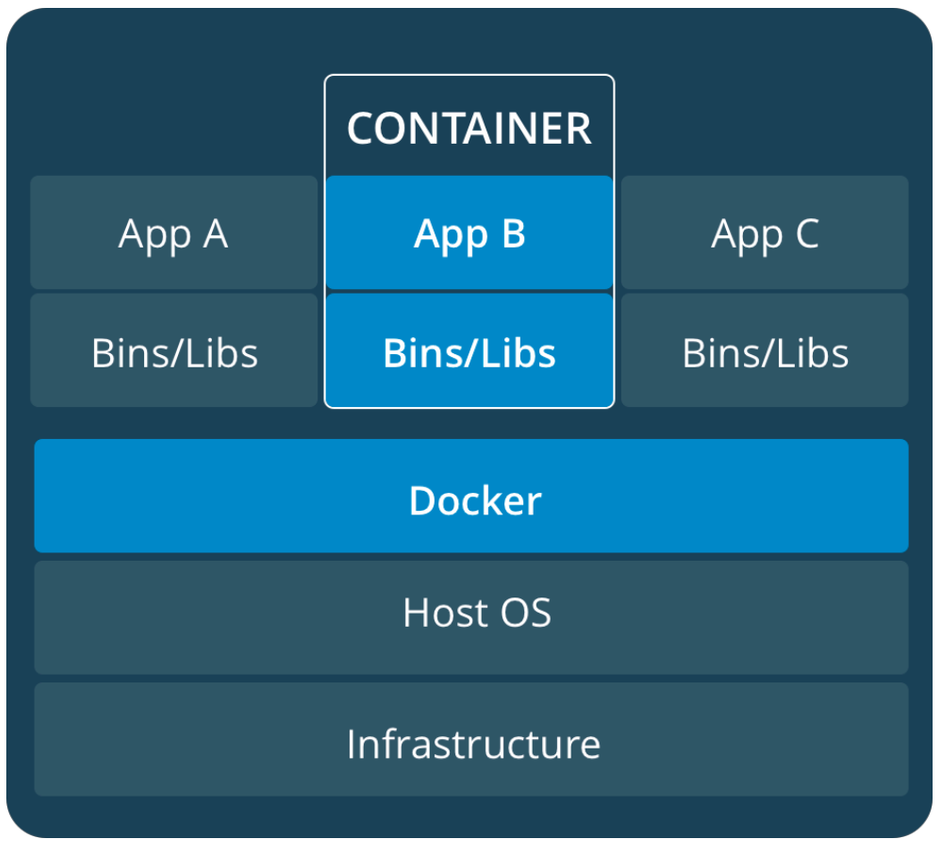
Promotes stable configurations and homogenous environments

Infrastructure as Code - this one is fancy :)

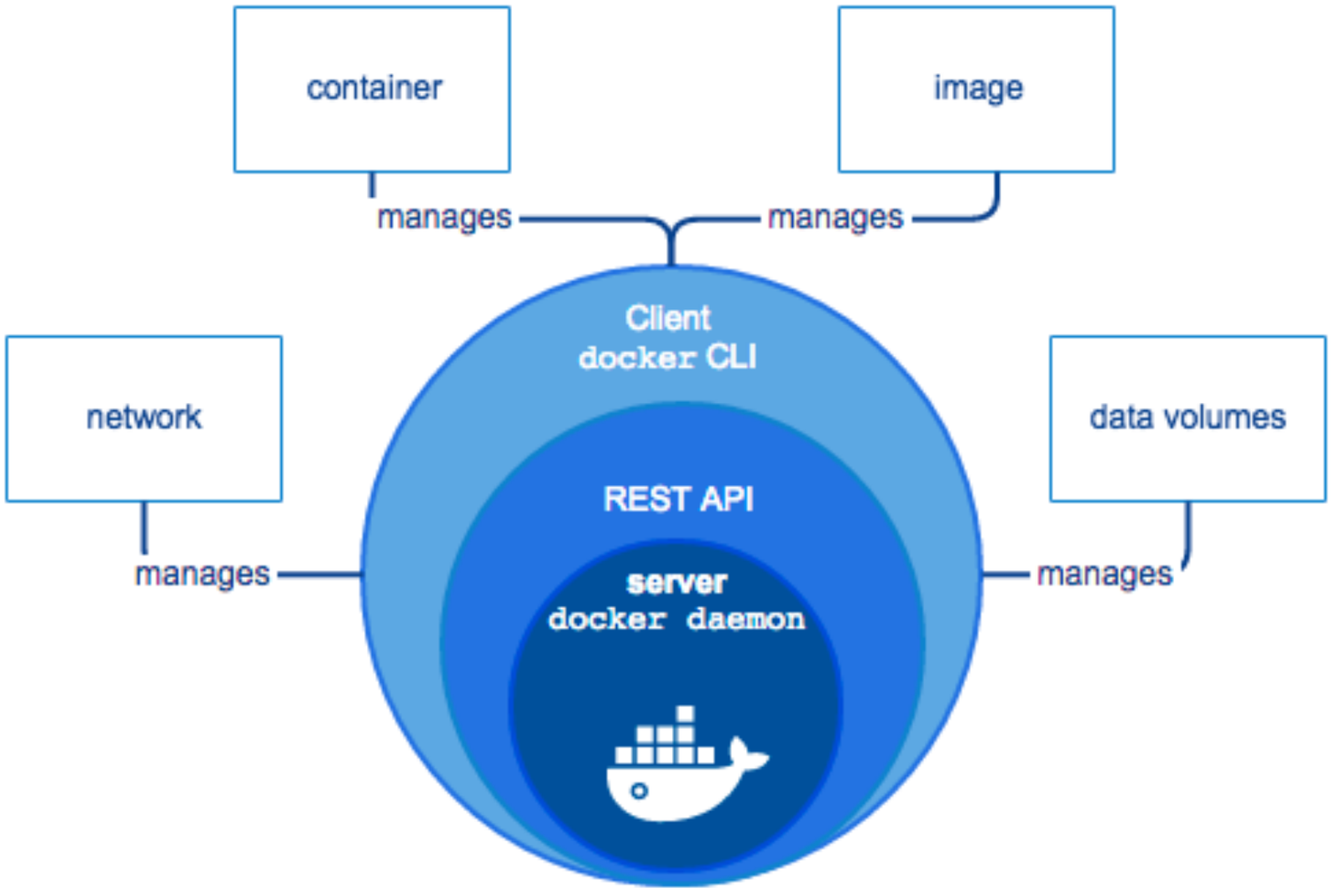
History

VM vs Container

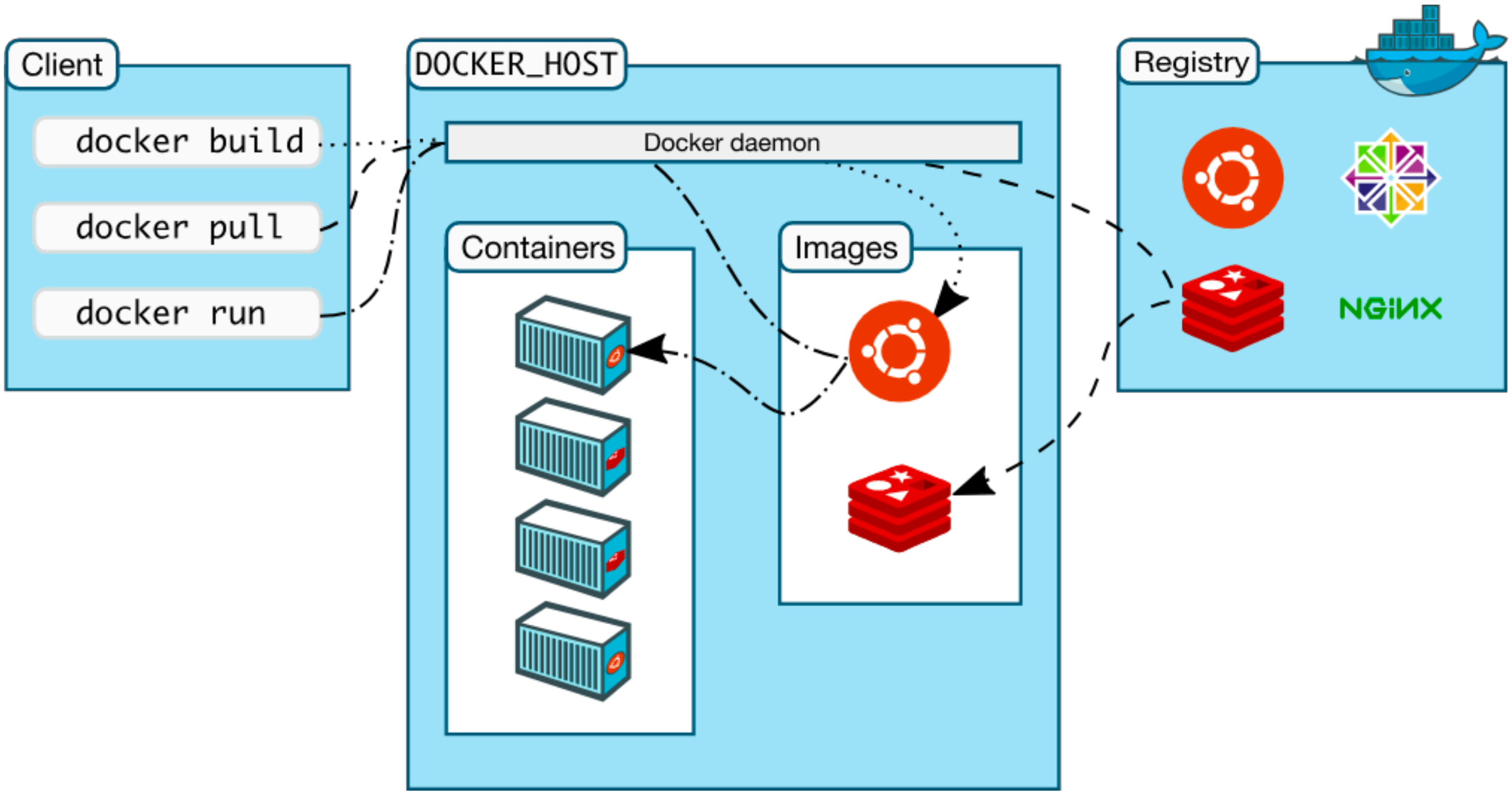

VM
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Guest OS
Hypervisor
Infrastructure


CONTAINER
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker
Host OS
Infrastructure

Docker ‘stack’ overview


container
image
manages
manages
Client
docker CLI
network
data volumes
REST API
manages
server
docker daemon
manages

‘Architecture’


Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

Dockerfile

Dockerfile

…defines what goes on in the environment inside your container.

https://docs.docker.com/get-started/part2/#dockerfile

Image

Image

…An **image** is an executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables, and configuration files.

https://docs.docker.com/get-started/#images-and-containers

Container

Container

…in the docker world:
A running instance of an image

- Flexible: Even the most complex applications can be containerized.
- Lightweight: Containers leverage and share the host kernel. :(
- Interchangeable: You can deploy updates and upgrades on-the-fly.
- Portable: You can build locally, deploy to the cloud, and run anywhere.
- Scalable: You can increase and automatically distribute container replicas.
- Stackable: You can stack services vertically and on-the-fly.

Image Registry

Image Registry

…A registry is a collection of repositories, and a repository is a collection of docker images—sort of like a GitHub repository, except the code is already built…

https://hub.docker.com/

https://docs.docker.com/get-started/part2/#share-your-image

Volumes

Volumes

…Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.

https://docs.docker.com/storage/volumes/

- Volumes are easier to back up or migrate than bind mounts.
- You can manage volumes using Docker CLI commands or the Docker API.
- Volumes work on both Linux and Windows containers.
- Volumes can be more safely shared among multiple containers.
- Volume drivers let you store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
- New volumes can have their content pre-populated by a container.

Networking

Networking

…One of the reasons Docker containers and services are so powerful is that you can connect them together, or connect them to non-Docker workloads

https://docs.docker.com/network/

Network driver summary

- **User-defined bridge networks** are best when you need multiple containers to communicate on the same Docker host.
- **Host networks** are best when the network stack should not be isolated from the Docker host, but you want other aspects of the container to be isolated.
- **Overlay networks** are best when you need containers running on different Docker hosts to communicate, or when multiple applications work together using swarm services.
- **Macvlan networks** are best when you are migrating from a VM setup or need your containers to look like physical hosts on your network, each with a unique MAC address.
- **Third-party network plugins** allow you to integrate Docker with specialized network stacks.

Hands-On