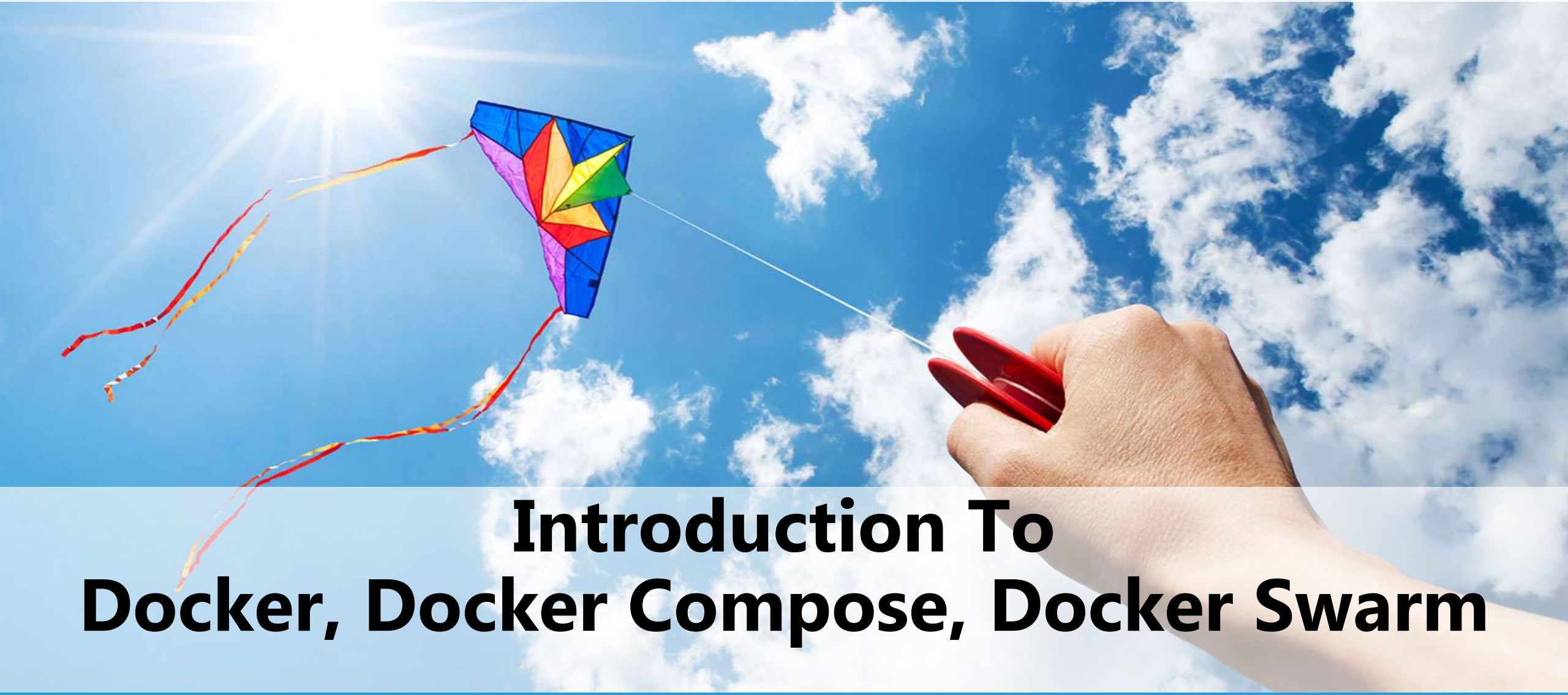


Introduction To Docker, Docker Compose, Docker Swarm

About me

An Nguyen

Email: nthienan@tma.com.vn

4 years of experience in software industry

Java, Python developer, DevOps engineer at TMA Solutions

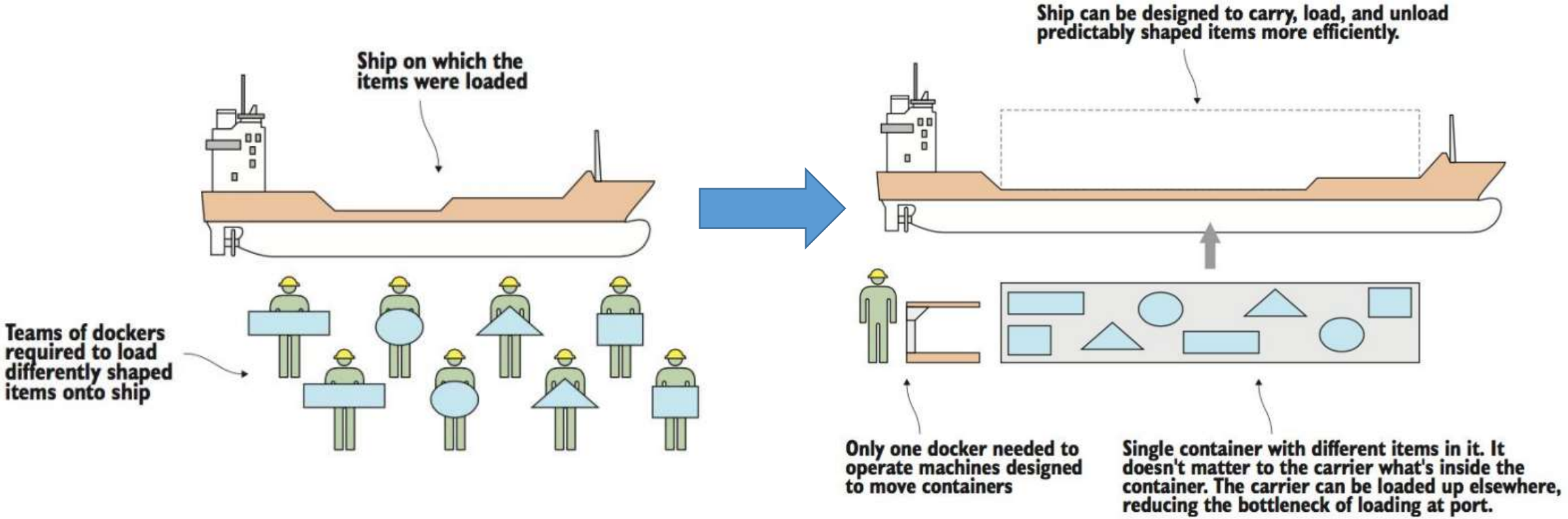
Agenda

- Docker
- Docker Compose
- Docker Swarm

- Who knows about Docker?
- Who uses Docker for development?
- Who uses Docker in production?
- Who tried but could not do it?

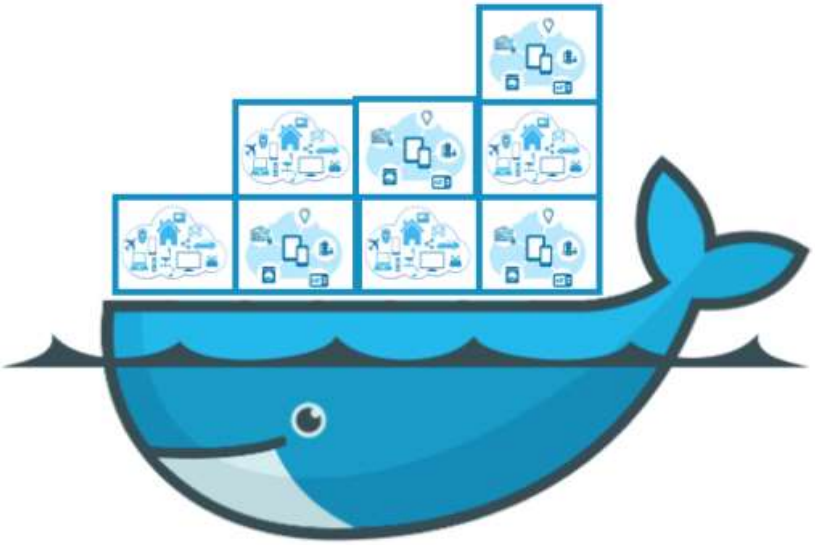
What is Docker?

- Docker is container engine



What is Docker? (cont.)

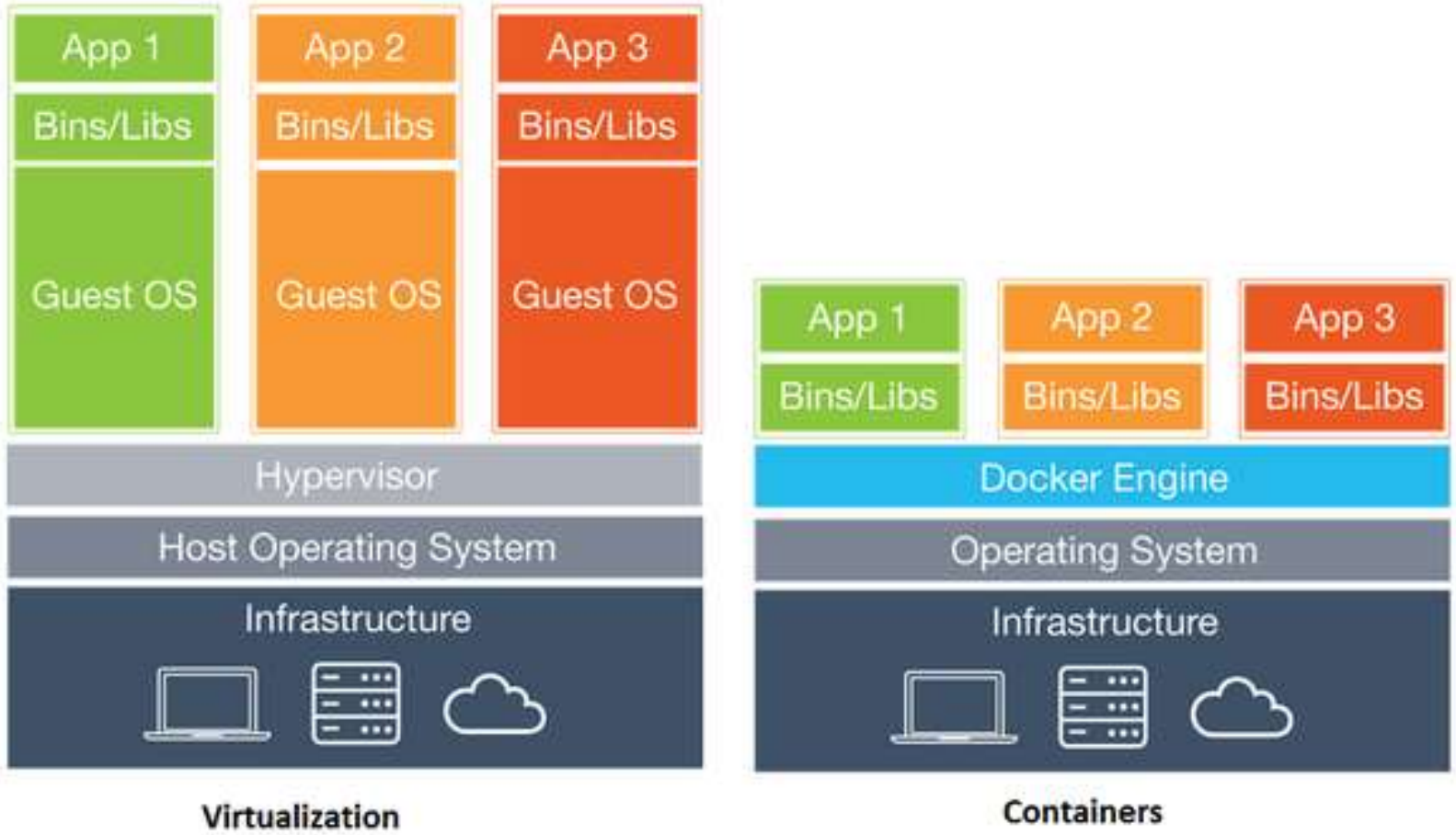
- Docker is an open platform for developing, shipping, and running applications.
- Docker allows you to package an application with all of its dependencies into a standardized unit for software development.



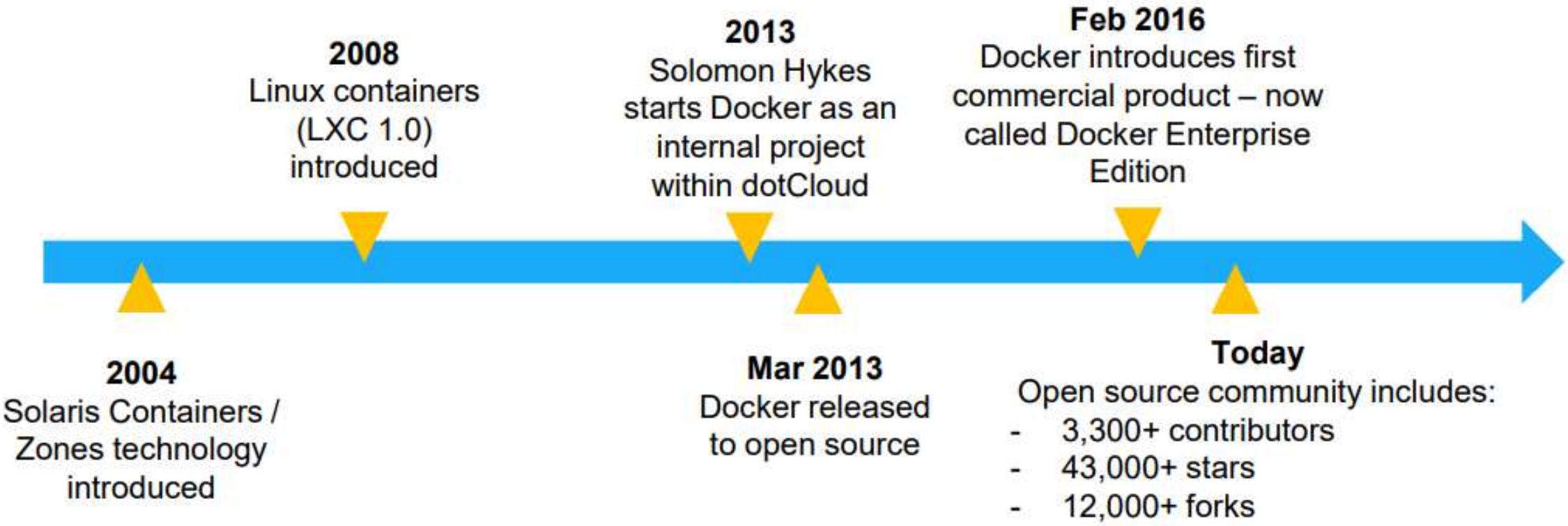
What is Docker? (cont.)

- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works for all major Linux distributions
- Containers native to Windows Server 2016

Virtual Machine vs Container



History of Docker



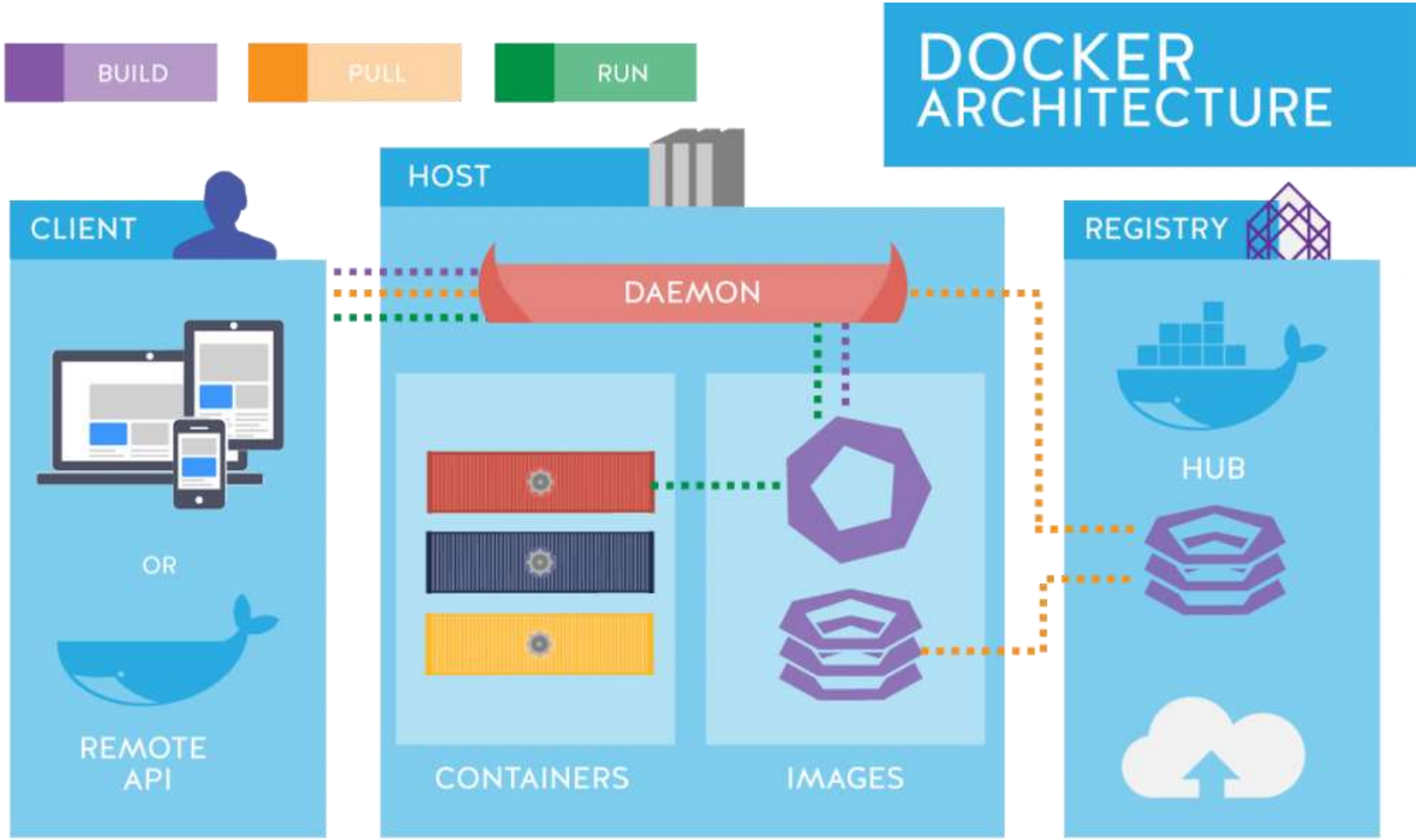
Benefits of Docker

- Fast (deployment, migration, restarts)
- Secure
- Lightweight (save disk & CPU)
- Open Source
- Portable software
- Micro-services and integrations (APIs)
- Simplify DevOps
- Version control capabilities

Technologies behind Docker

- Linux x86-64
- Go language
- Client - Server (deamon) architecture
- Union file systems (UnionFS: AUFS, btrfs, vfs etc)
- Namespaces (pid, net, ipc, mnt, uts)
- Control Groups (cgroups)
- Container format (libcontainer)

Docker architecture



https://docs.docker.com/engine/docker-overview/#docker-architecture

Docker Vocabulary

Docker client

- Is the primary user interface to Docker.
- Accepts commands from the user and communicates back and forth with a Docker daemon

Docker daemon

- Runs on a host machine.
- Users do not directly interact with the daemon, but instead through the Docker client with the RESTful API or sockets

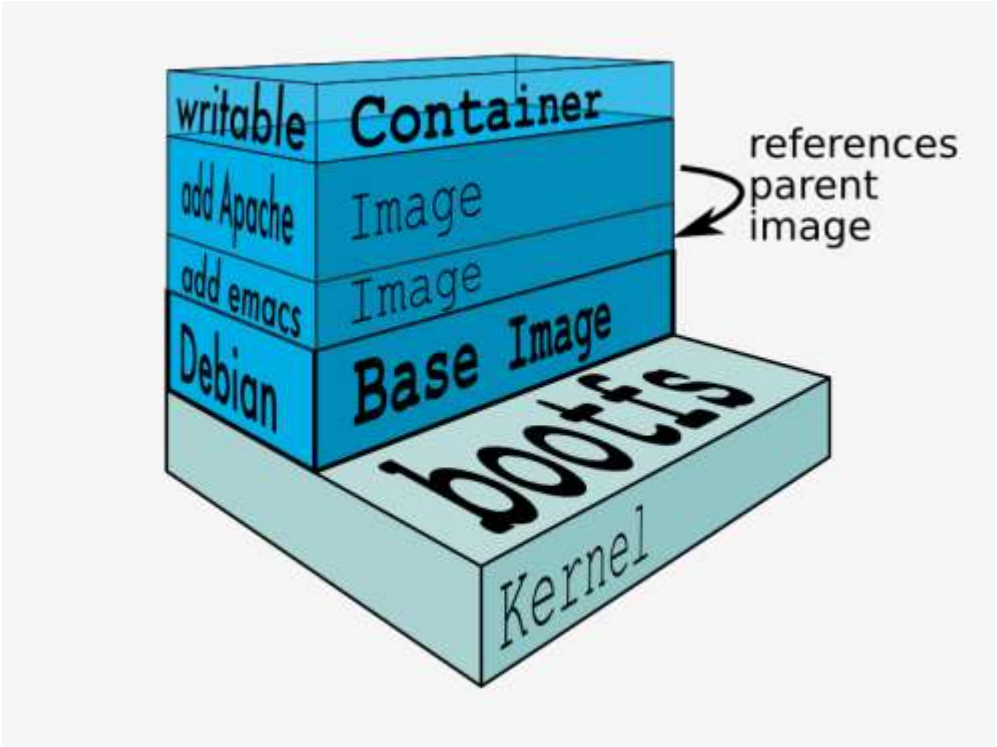
Docker Vocabulary (cont.)

❑ Image

- Is a **read-only** template with instructions for creating a Docker container.
- An image is based on another image, with some additional customization

❑ Container

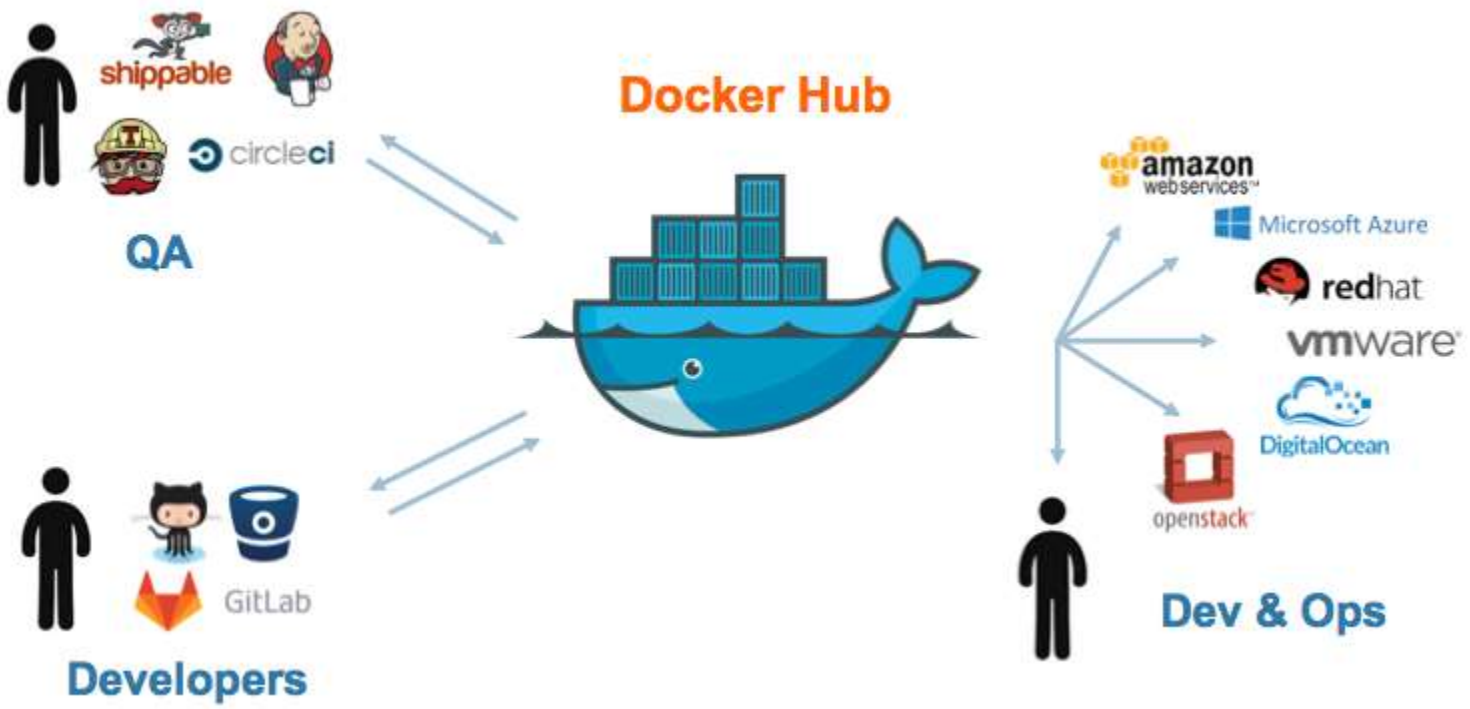
- Is a runnable instance of an image.
- A container is removed, any changes to its state that are not stored in persistent storage disappear



Docker Vocabulary (cont.)

❑ Registry

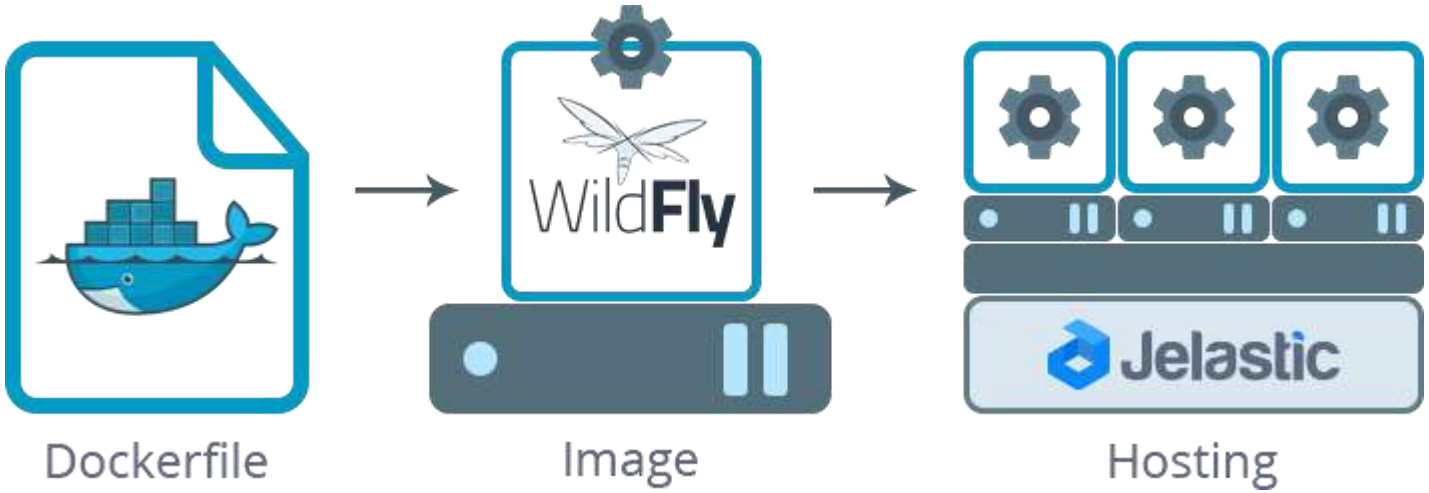
- A Docker *registry* stores Docker images.
- Docker Hub and Docker Cloud are public registries that anyone can use
- You can run your own private registry



Docker Vocabulary (cont.)

❑ Dockerfile

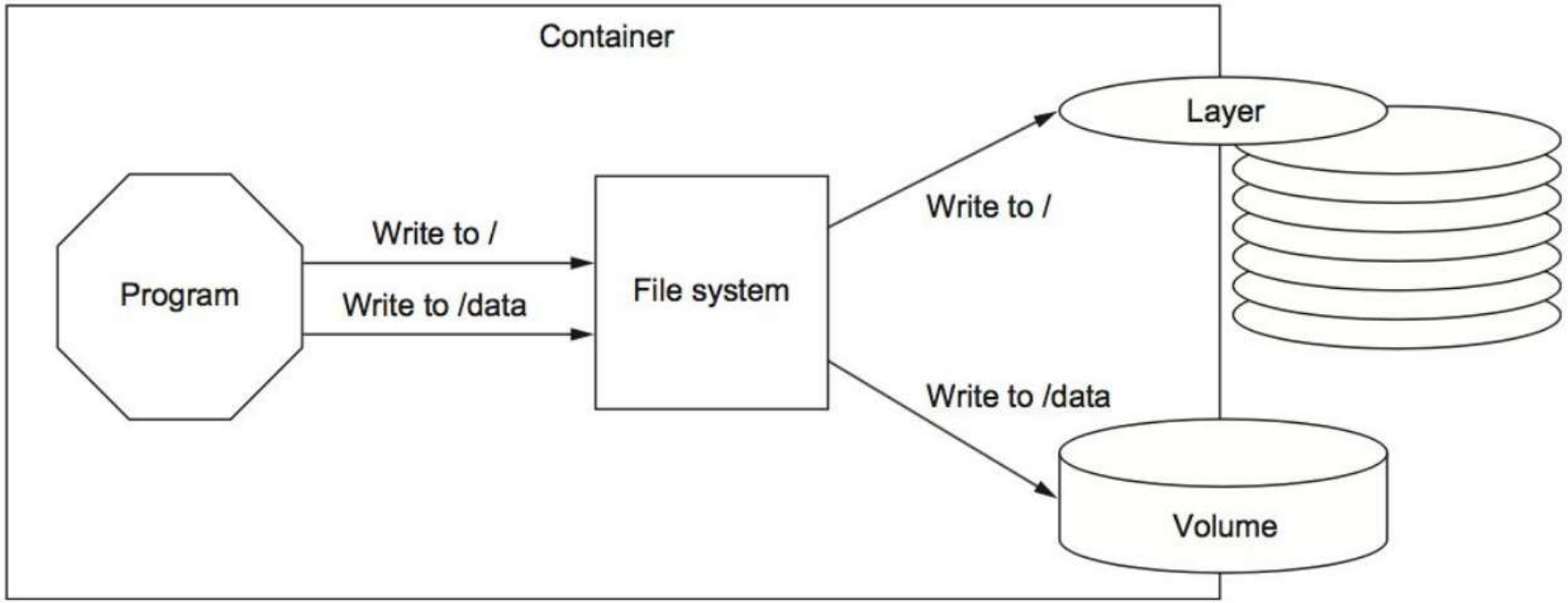
- Is a text document that contains all the commands a user could call on the command line to create an image
- Dockerfile reference on docker docs



Docker Vocabulary (cont.)

❑ Volume

- Is a mount point on the container's directory tree where a portion of the host directory tree has been mounted



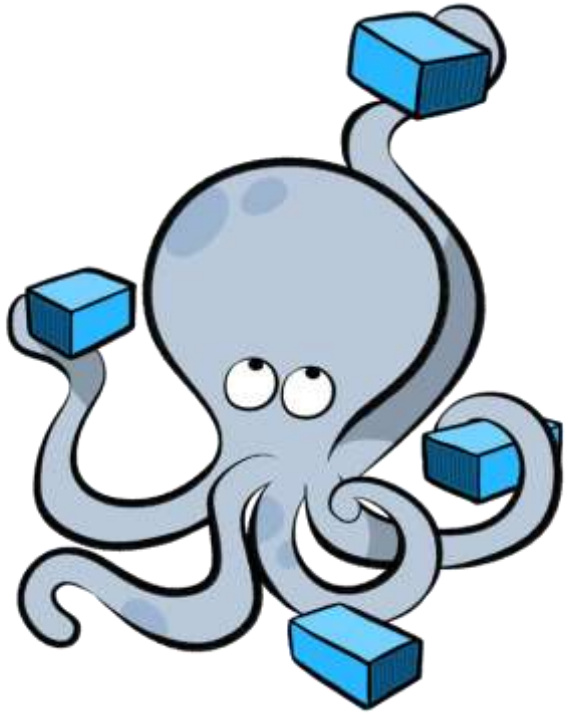
Demo Time

Common Docker usages

- Sandbox environment (develop, test, debug, educate)
- Continuous Integration & Deployment
- Scaling apps
- Development collaboration
- Infrastructure configuration
- Local development
- Multi-tier applications
- PaaS, SaaS

Useful resources

- Awesome Docker (list of Docker resources & projects)
- Docker cheat sheet
- Docker in Practice, The Docker Book (books)
- Docker aliases/shortcuts
- Docker case studies
- Self-learning courses



Docker Compose

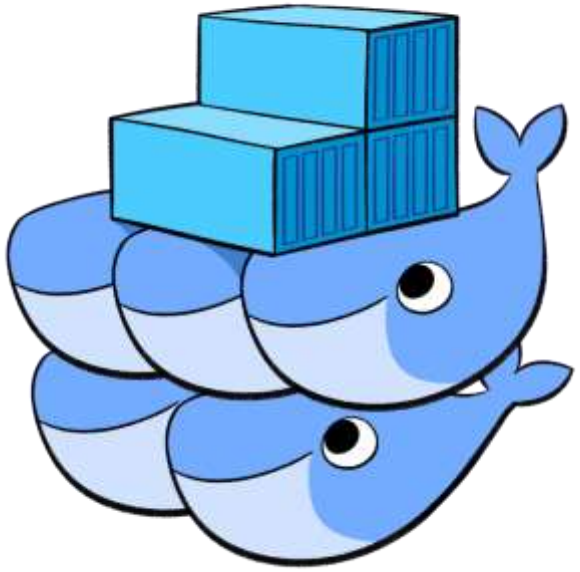
Real life applications

- One application consists of multiple containers.
- One container is dependent on another.
- Mutual dependency/ startup order.
- Process involves building containers and then deploy them
- Long docker run commands
- Complexity is proportional to the number of containers involved

What is Docker compose?

- Tool for defining and running multi-container Docker application.
- It is a YML file.
- Compose contains information about how to build the containers and deploy containers.
- Integrated with Docker Swarm

Demo Time

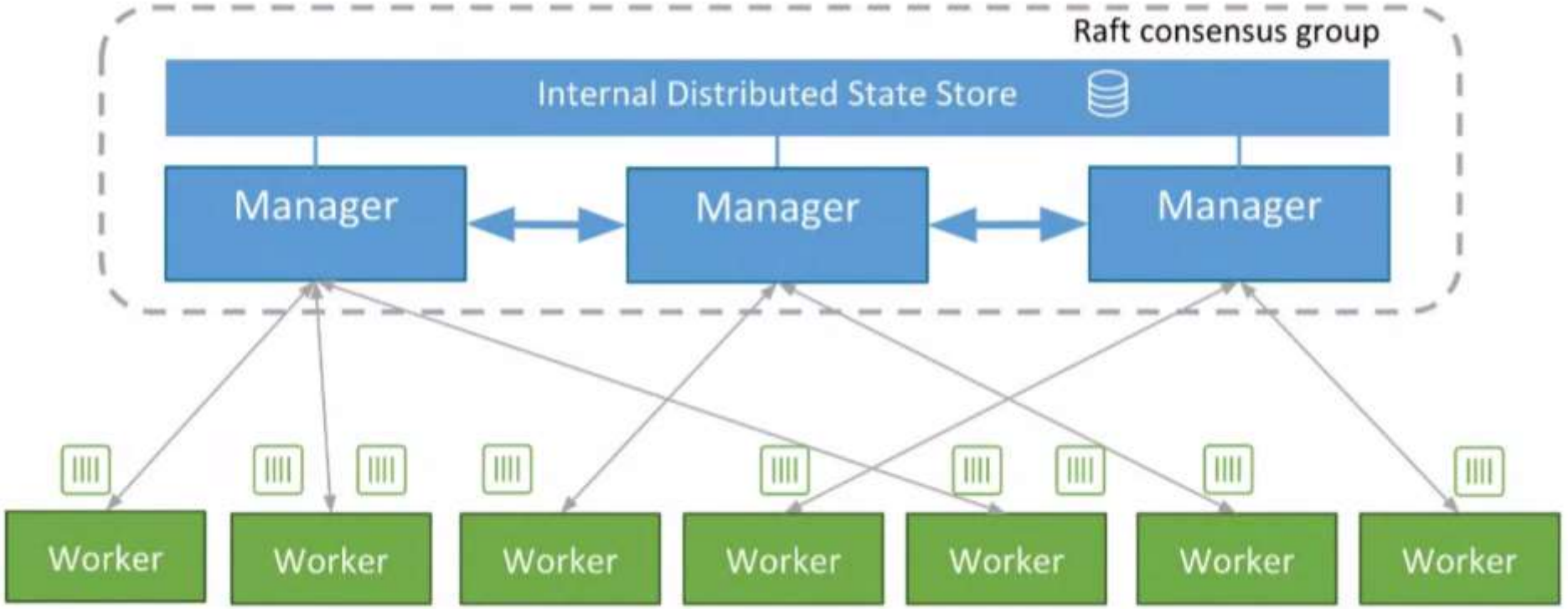


Docker Swarm

What is Docker swarm?

- Is a clustering and scheduling tool for Docker containers. With Swarm, IT administrators and developers can establish and manage a cluster of Docker nodes as a single virtual system
- A swarm is a group of machines that are running Docker and joined into a cluster

Swarm architecture



Feature highlights

Cluster management integrated with Docker Engine

- Use the Docker Engine CLI to create a swarm of Docker Engines where you can deploy application services

Declarative service model

- Uses a declarative approach to let you define the desired state of the various services in your application stack

Scaling

- When you scale up or down, the swarm manager automatically adapts by adding or removing tasks to maintain the desired state

Feature highlights (cont.)

Desired state reconciliation

- Swarm manager node constantly monitors the cluster state and reconciles any differences between the actual state and your expressed desired state

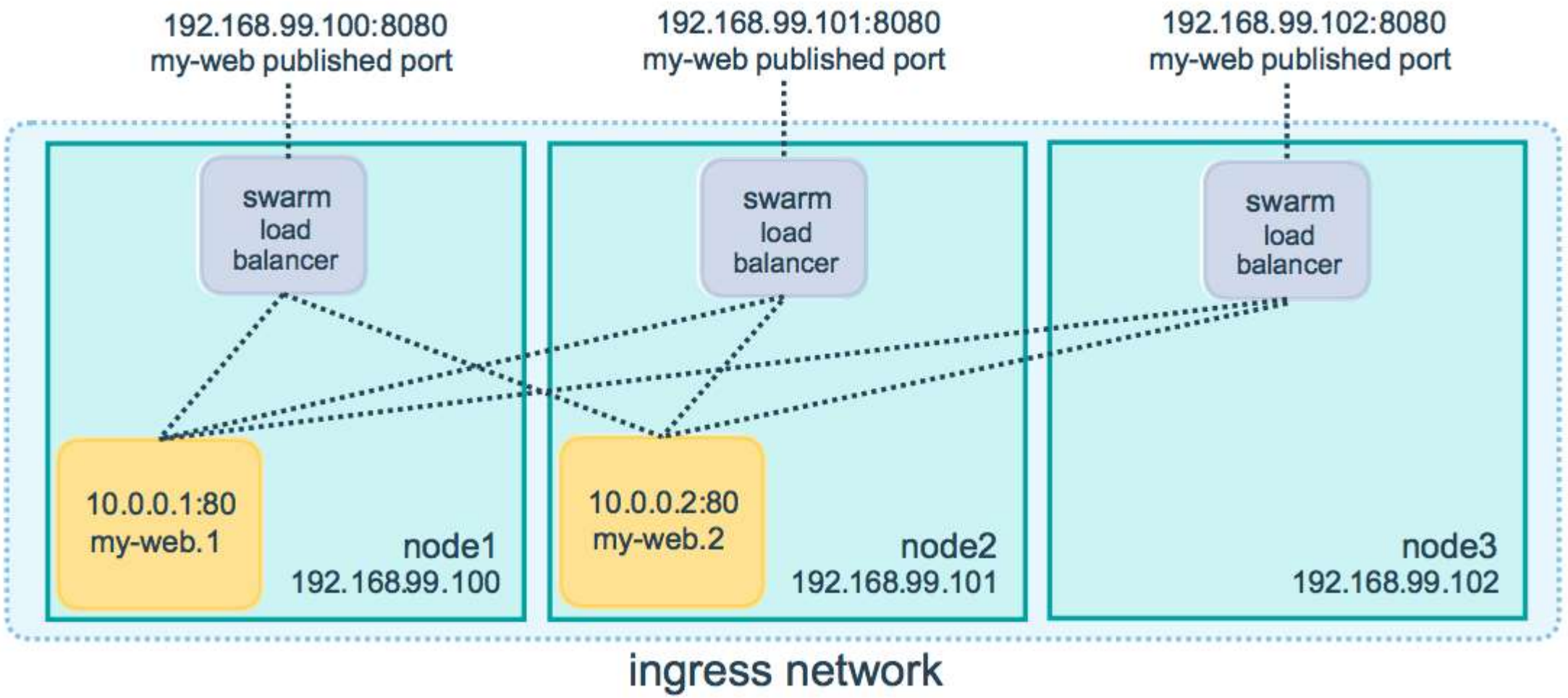
Service discovery

- Swarm manager nodes assign each service in the swarm a unique DNS name and load balances running containers

Rolling updates

- At rollout time you can apply service updates to nodes incrementally. If anything goes wrong, you can roll-back a task to a previous version of the service

Routing mesh



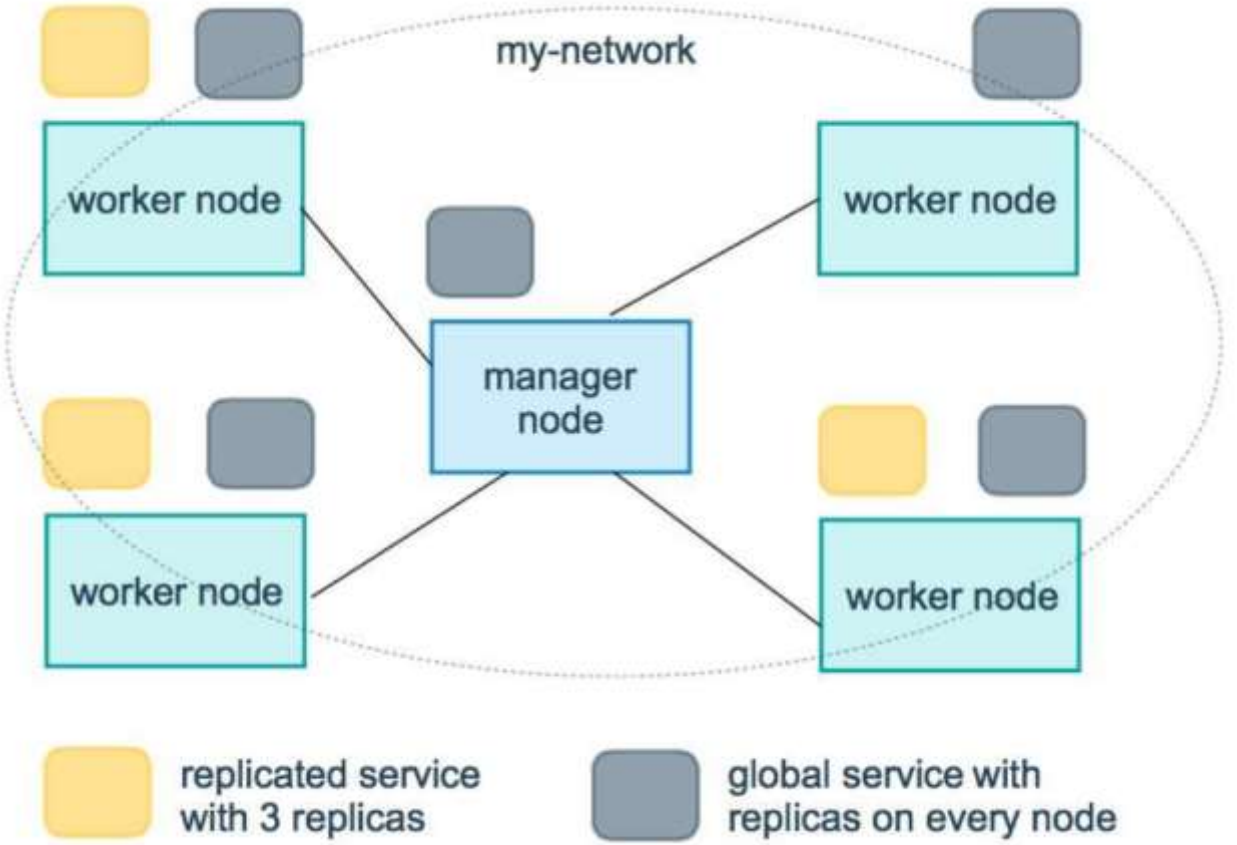
Core concepts

- **Service**:
  - Services are really just “containers in production”
  - A service only runs one image, but it codifies the way that image runs—what ports it should use, how many replicas of the container should run so the service has the capacity it needs, and so on
  - Scaling a service changes the number of container instances running that piece of software

Core concepts (cont.)

Two type of service:

- **Replicated**
  - Specify the number of identical tasks you want
- **Global**
  - Service that runs one task on every node



Core concepts (cont.)

- **Stack**:
  - A stack is a group of interrelated services that share dependencies, and can be orchestrated and scaled together
  - A single stack is capable of defining and coordinating the functionality of an entire application (though very complex applications may want to use multiple stacks)

Demo Time

Q&A

Thank You!

Tel:	+84 8 3997-8000	North America number:	+ 1 802-735-1392
Mobile:	+84 908-676-212	Australia number:	+ 61 414-734-277
Fax:	+84 8 3990-3303	Japan number:	+81 3-6432-4994
Email:	sales@tmasolutions.com	Website:	www.tmasolutions.com