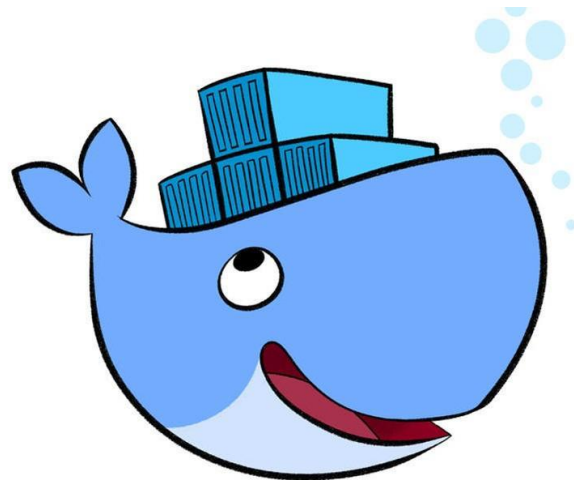


Docker Containers for Beginners

Steen Dybboe

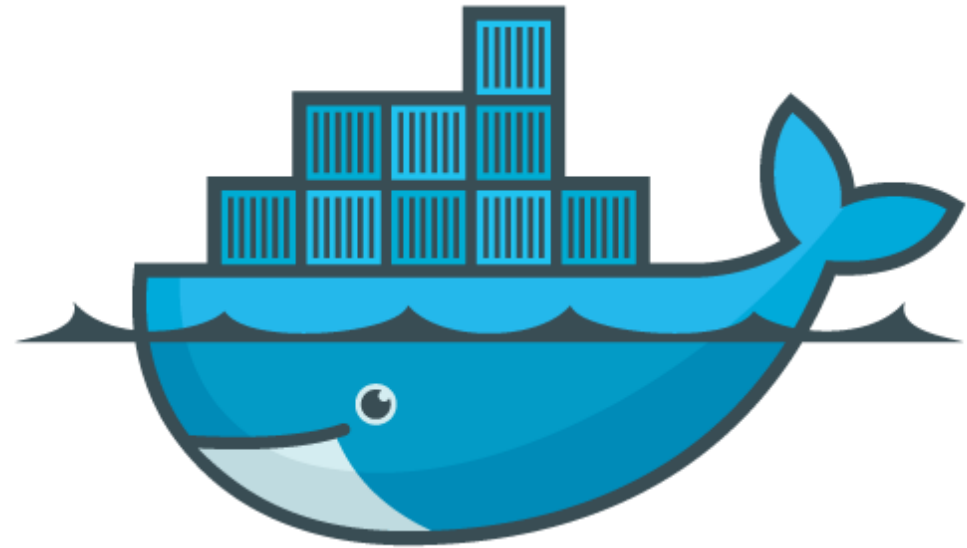


Pragmatic BI



Agenda

- The Challenge
- VM and Container
- Docker
- MsSQL in Docker Container
- Kubernetes

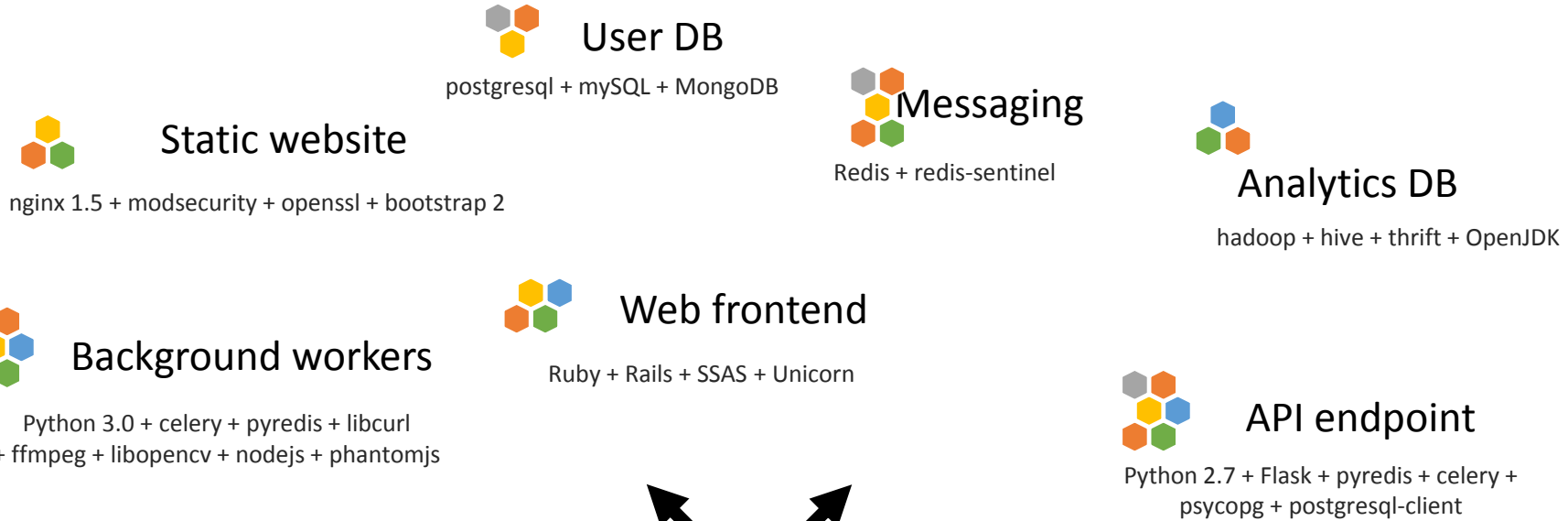


docker

The Challenge

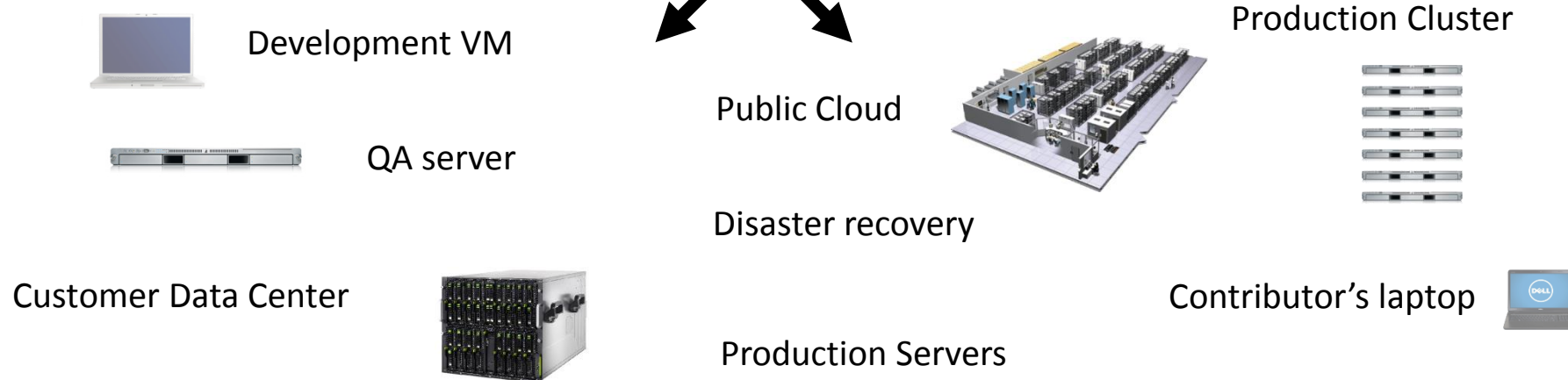


Multiplicity of Stacks



Do services and apps interact appropriately ?

Multiplicity of hardware environments



Can I migrate smoothly and quickly ?

The Matrix From Hell



	Static website	?	?	?	?	?	?	?
	Web frontend	?	?	?	?	?	?	?
	Background workers	?	?	?	?	?	?	?
	User DB	?	?	?	?	?	?	?
	Analytics DB	?	?	?	?	?	?	?
	Messaging	?	?	?	?	?	?	?
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers



Cargo Transport Pre-1960

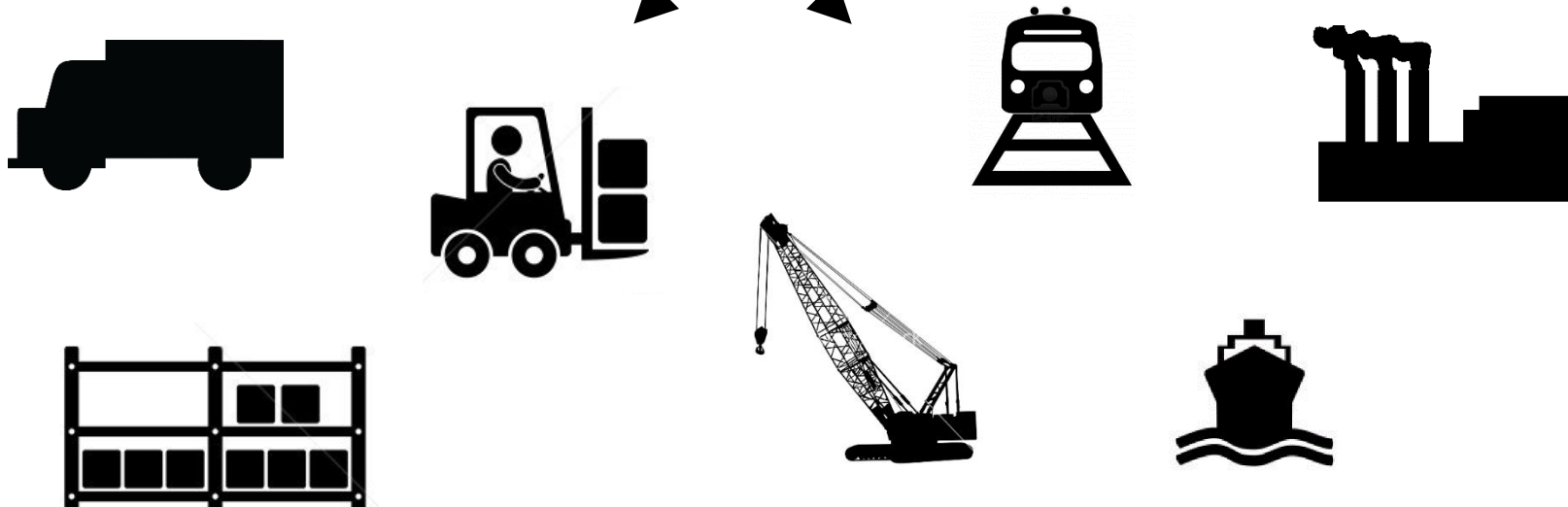


**Multiplicity
of Goods**



**Do I worry about
how goods
interact (e.g.
coffee beans
next to spices)**








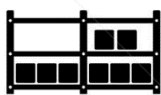





**Multiplicity
of methods
for
transporting
or storing**



**Can I transport
quickly and
smoothly
(e.g. from boat
to train to truck)**

Also a matrix from hell



	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

Solution: Intermodal Shipping Container



Multiplicity of Goods



A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

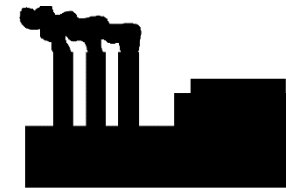
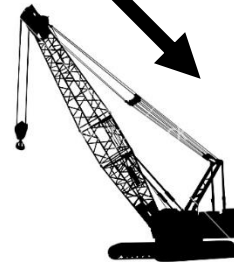
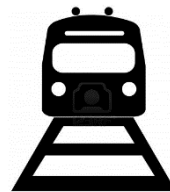
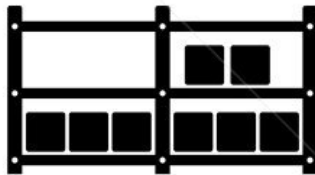


...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another

Do I worry about how goods interact (e.g. coffee beans next to spices)

Can I transport quickly and smoothly (e.g. from boat to train to truck)

Multiplicity of methods for transporting/storing



Docker is a shipping container system for code



Multiplicity of Stacks

Static website User DB Web frontend Messaging Analytics DB

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...



Do services and apps interact appropriately?

...that can be manipulated using standard operations and run consistently on virtually any hardware platform

Can I migrate smoothly and quickly

Multiplicity of hardware environments

Development VM

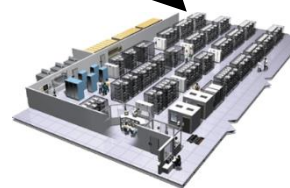
QA server

Customer Data Center

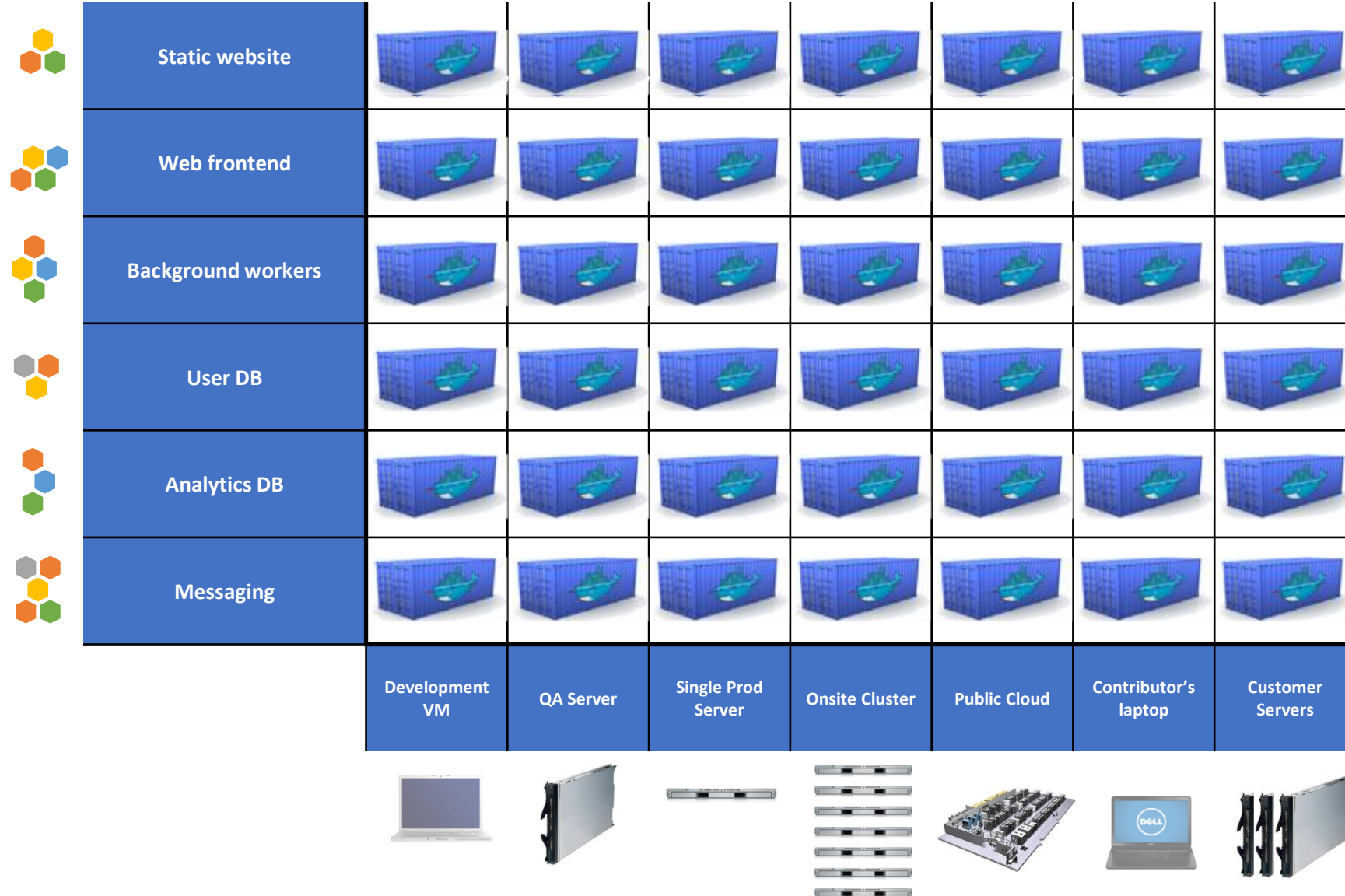
Public Cloud

Production Cluster

Contributor's laptop



Docker eliminates the matrix from Hell





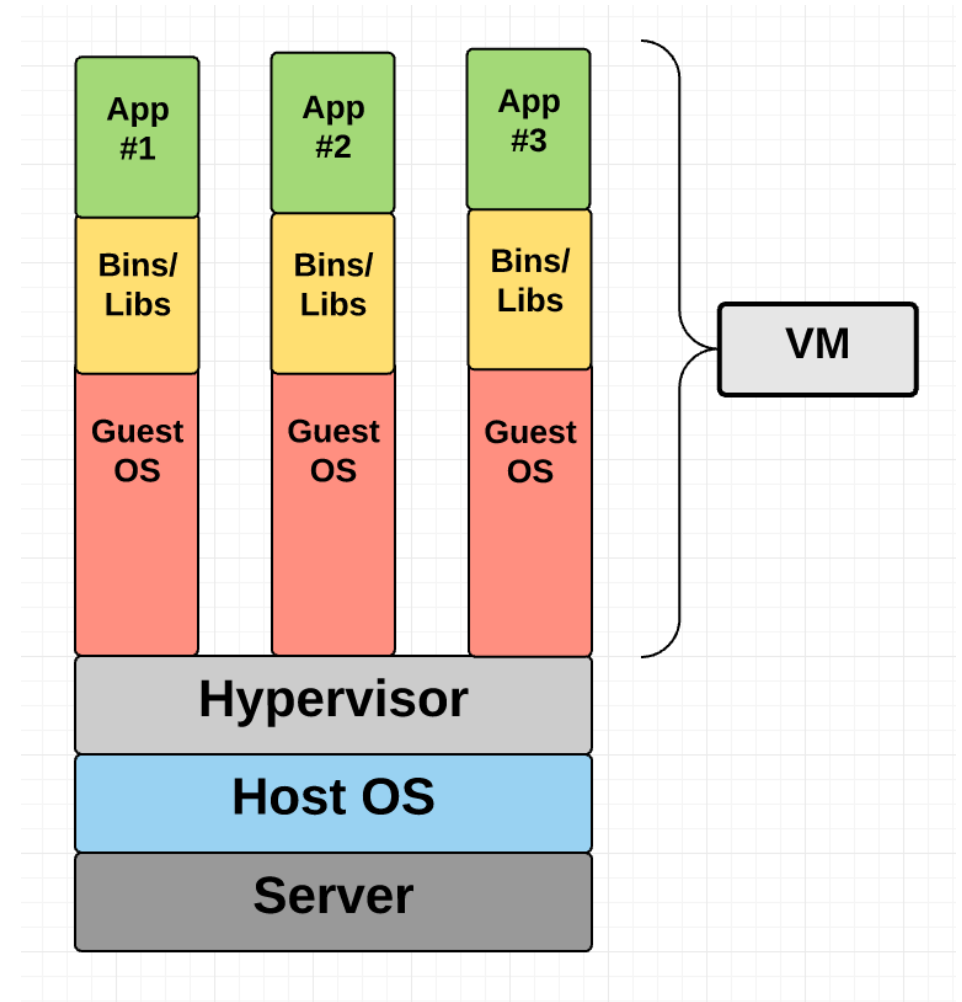
Purpose of VM og Containerere

- To **isolate** an application and its dependencies into a self-contained unit that can run anywhere.
- Containers and VMs **remove the need for physical hardware**
- The main difference between containers and VMs is in their architectural approach



Virtual Machines

- Emulation of a real computer that executes programs like a real computer.
- VMs run on top of a physical machine using a “*hypervisor*”.
 - Piece of software, firmware, or hardware that VMs run on top of.
- Can run on either a **hosted hypervisor** or a **bare-metal hypervisor**



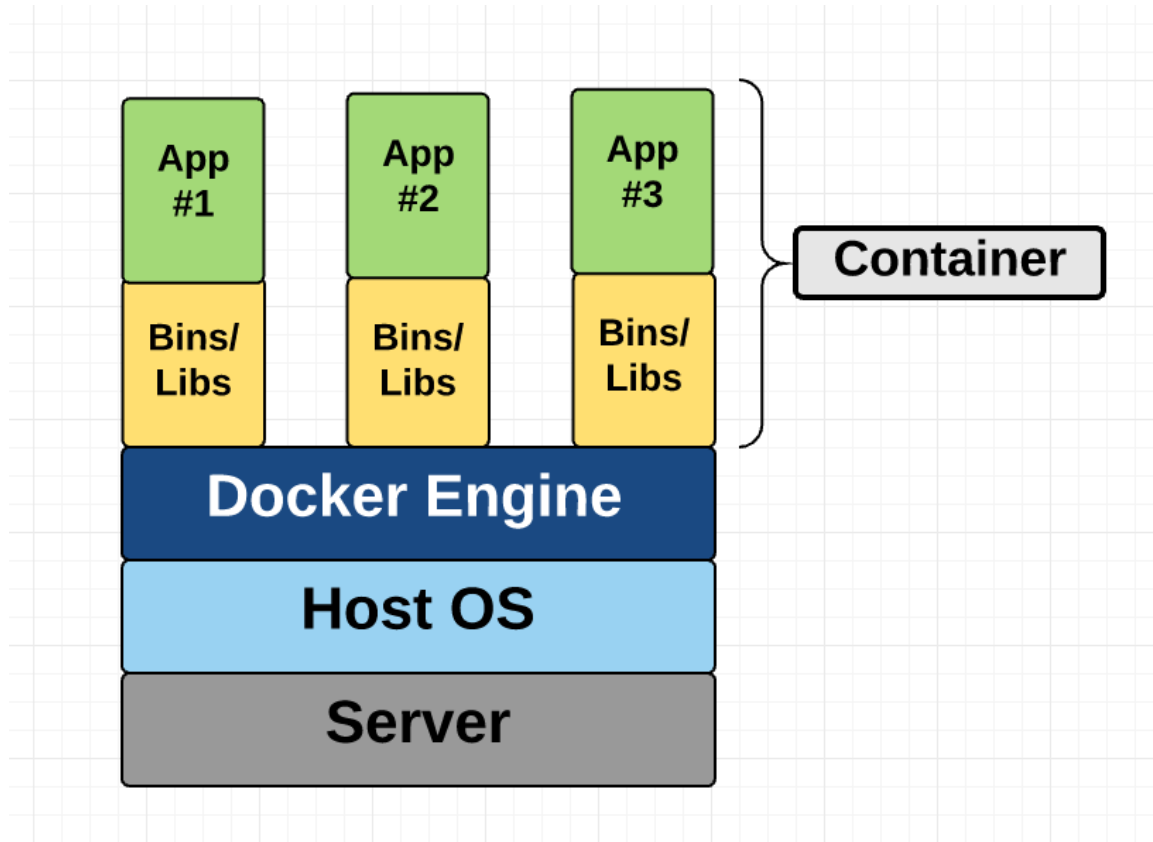


Container

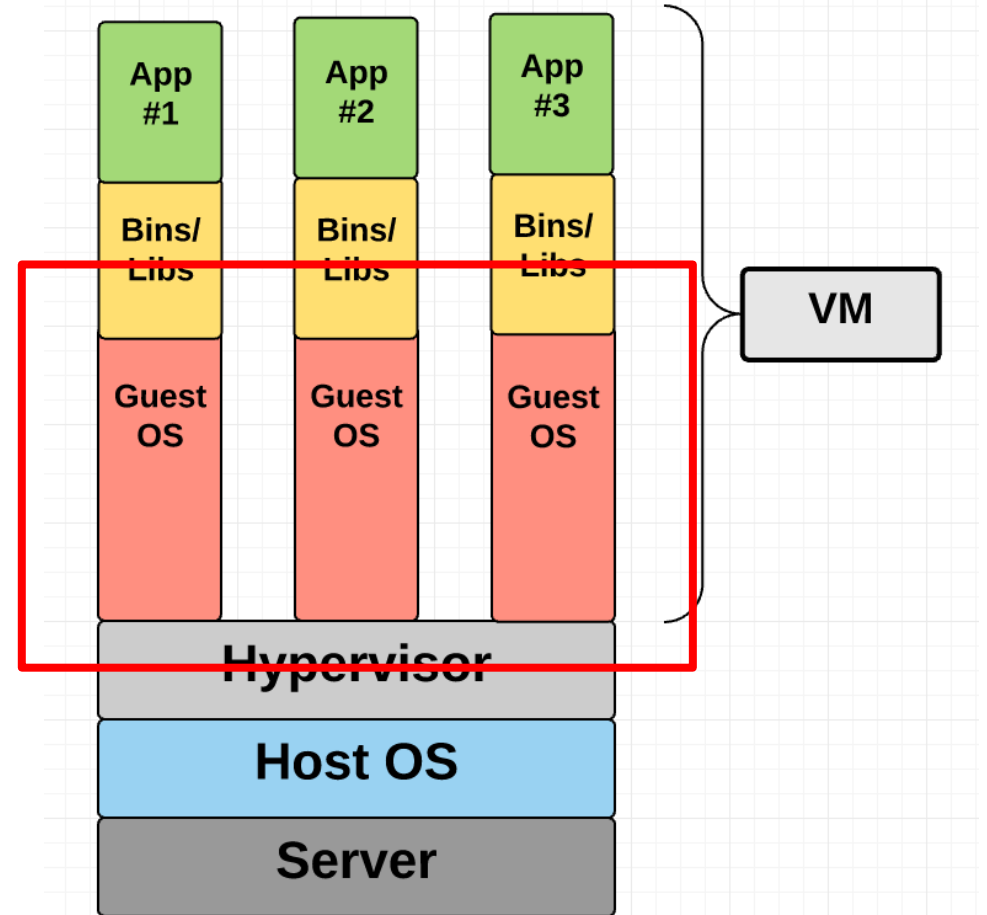
- “Old” technology
- Provides **operating-system-level** virtualization
- Look like a VM.
 - They have private space for processing
 - Can execute commands as root
 - Have a private network interface and IP address
 - Can mount file systems, and etc.
- The one big difference between containers and VMs is that containers “share” the host system’s kernel with other containers.



Container



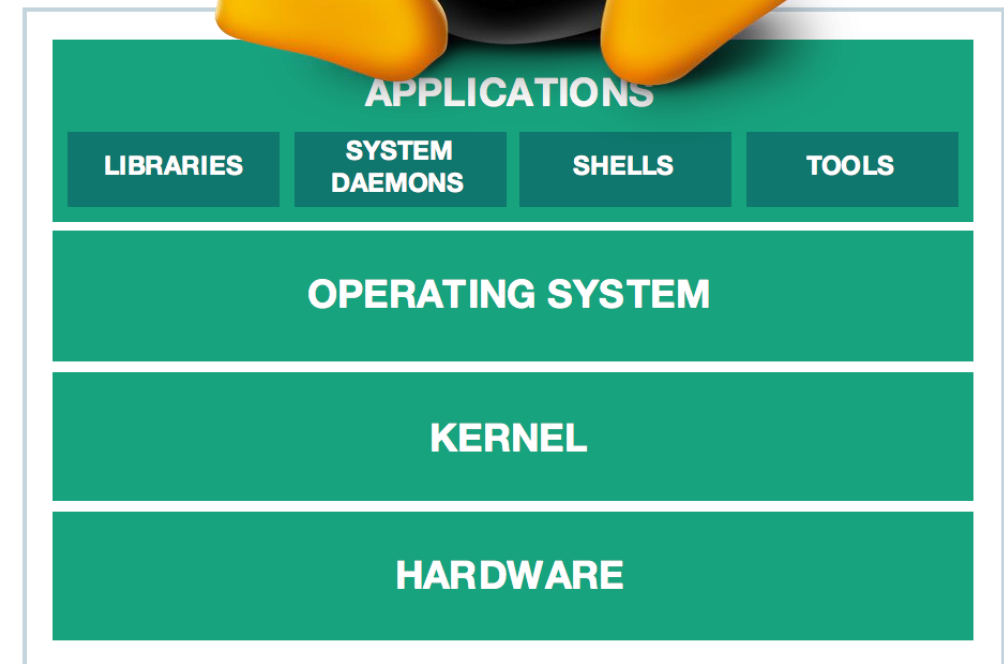
Virtual Machine





Linux

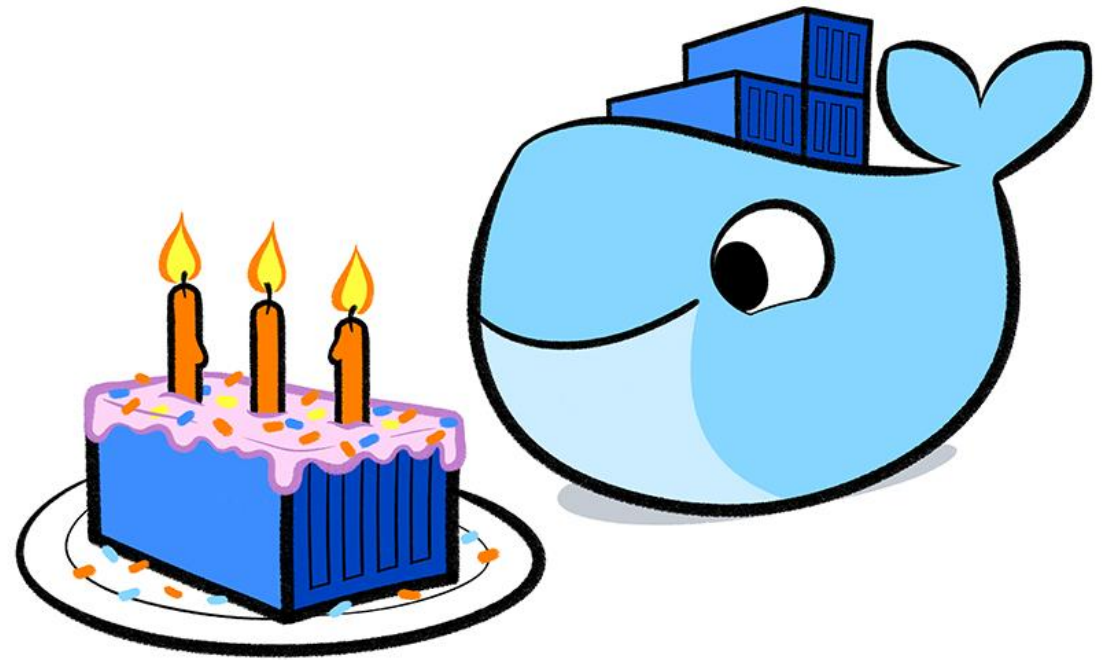
- Linux Kernel
- Linux Operating System Family
 - Ubuntu
 - Red Hat Enterprise Linux (RHEL)
 - Suse
 - CentOS
 - Debian
 - + Many, many more
- MacOS – A distant cousin
 - But a different kernel





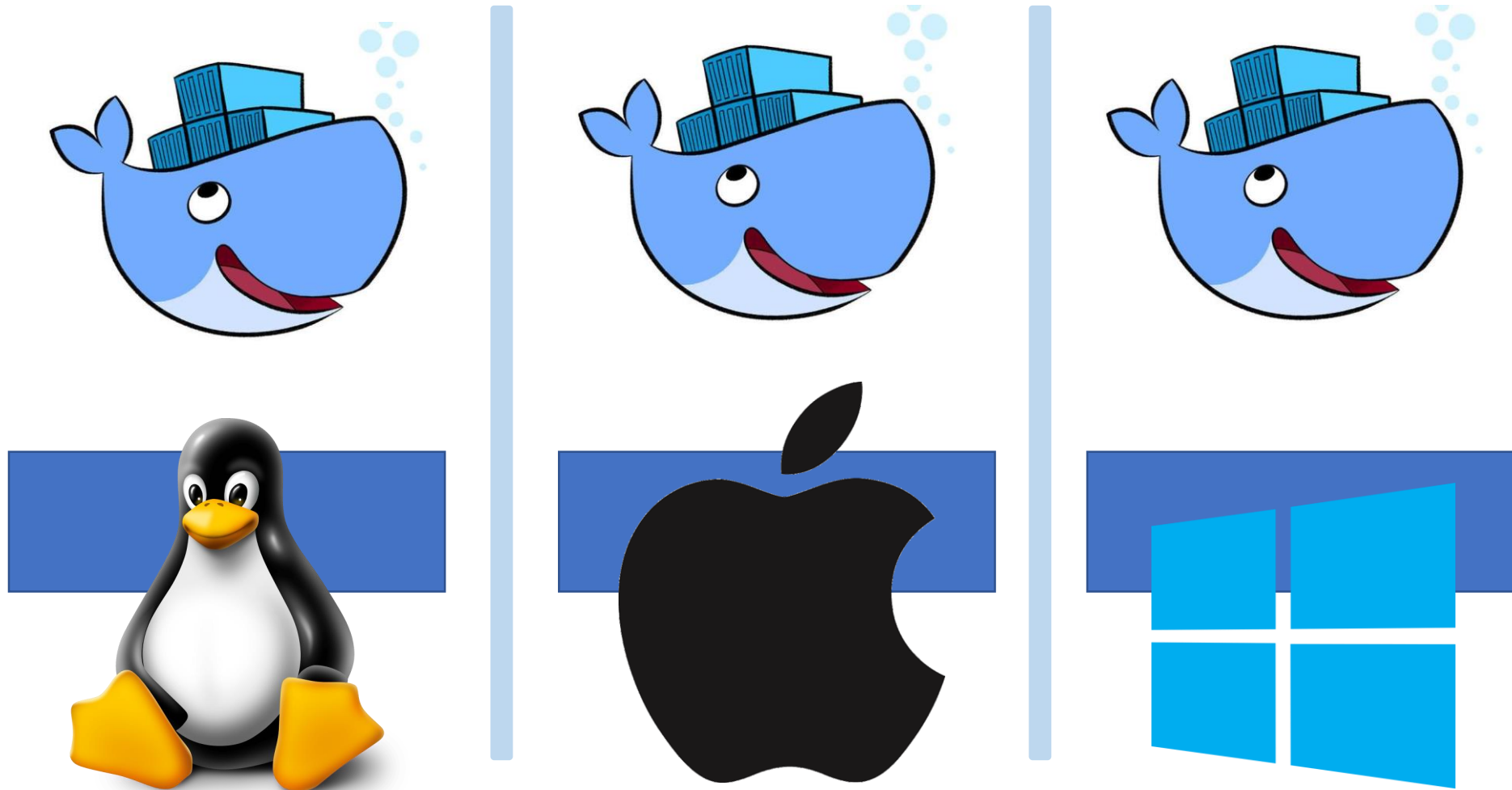
Docker

- Open-source project based on Linux containers
- Gaining popularity because of
 - Ease of use
 - Speed
 - Docker Hub
 - Modularity and Scalability



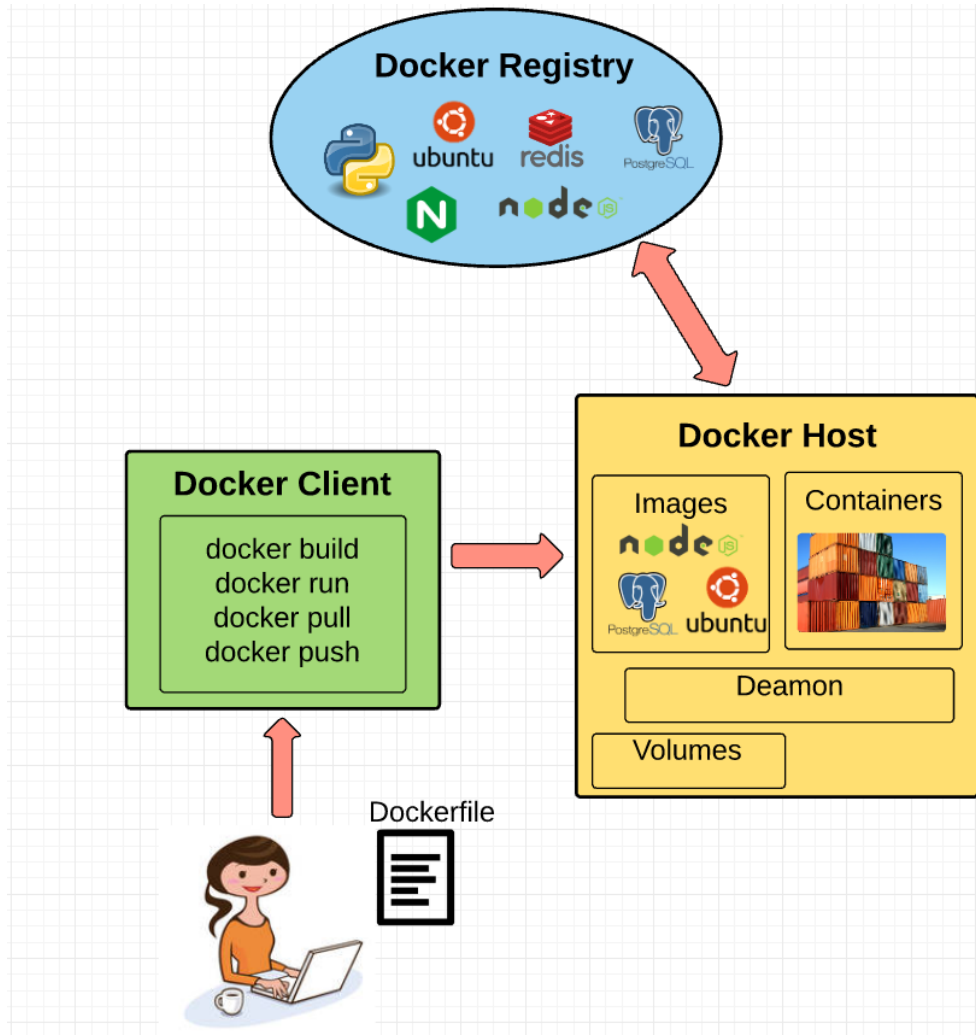


Kernels and containers





Concepts

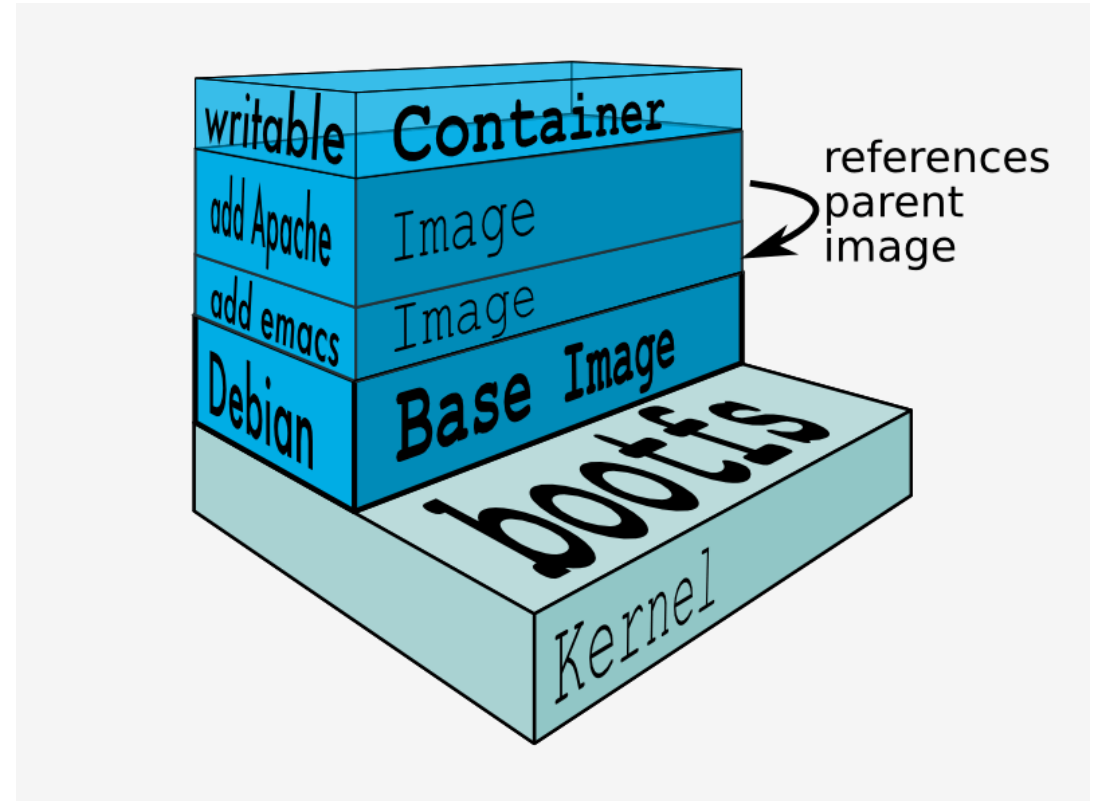


- Docker Registry
- Docker Engine
 - Docker Deamon
 - Docker Images
 - Docker File
 - **Docker Containers**
 - Volumes
- Docker Client



Docker Containers

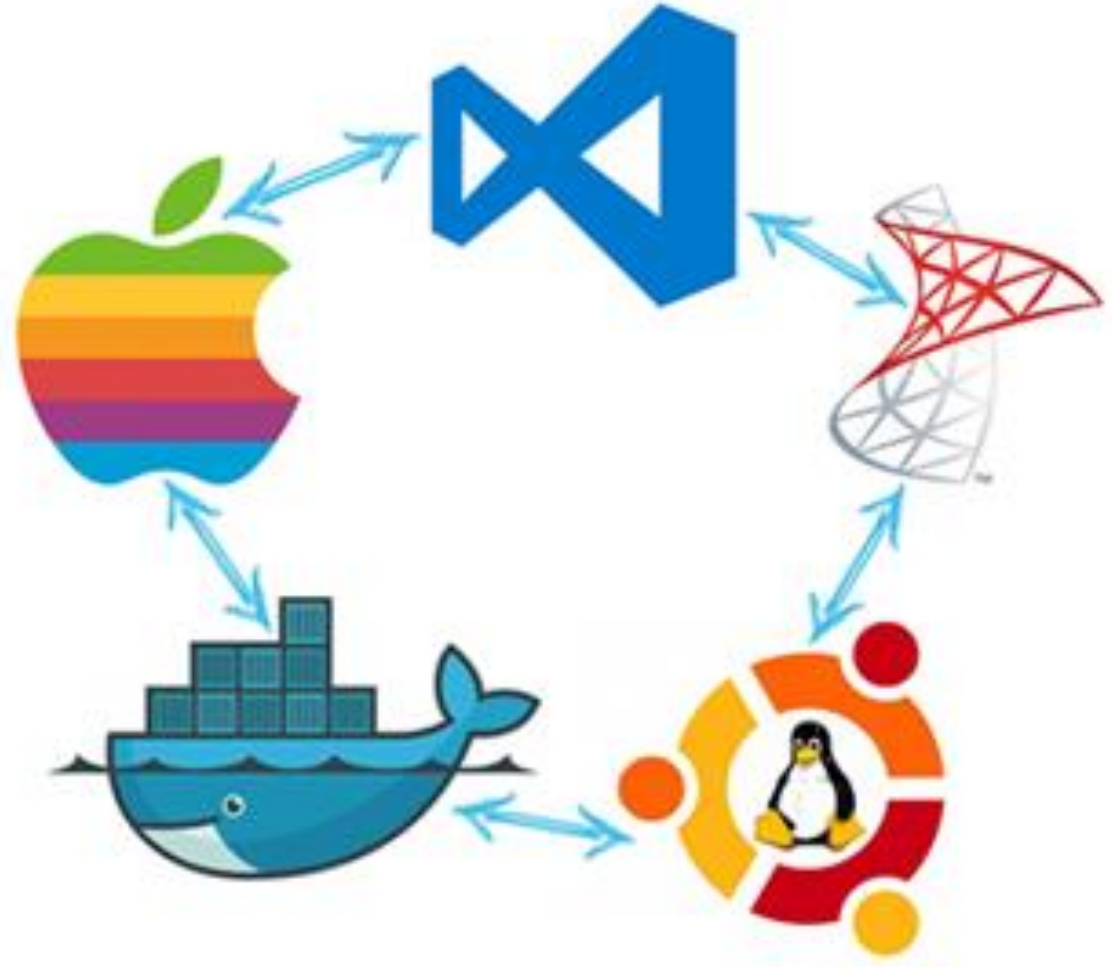
- Docker containers are built off Docker images.
- Box with an application's software
- Everything the application needs to run:
 - “operating system”
 - application code,
 - runtime, system tools,
 - system libraries, and etc.





Demo

The most simple WEB Server
in the world



MS SQL Server in Linux

- This is huge
- Microsoft isn't going open source with its server products
- This is a slap at Oracle
- MySQL/MariaDB and PostgreSQL are in no danger
- All editions will be supported
- There's a lot in SQL Server to like
- The economics of the cloud made this all but inevitable
- This is only the beginning



The SQL Server Image

- Public Registries
 - Docker Hub (hub.docker.com)
 - Red Hat Container Catalog (<https://access.redhat.com/containers>)
 - The Microsoft Container Registry (mcr.microsoft.com)
- SQL Server 2017
 - Ubuntu based Images
 - RTM plus each CU has separate image
 - Includes mssql-server and mssql-tools packages
- SQL Server 2019 Preview images
 - RHEL based images
 - Includes mssql-server and mssql-tools



Running a SQL Server Container

docker run

```
-e 'ACCEPT_EULA=Y'  
-e 'MSSQL_SA_PASSWORD=PragInCloud01'  
-p 1401:1433  
-v sqlvolume:/var/opt/mssql  
--name sql1  
-d  
mcr.microsoft.com/mssql/server:2017-CU8
```



Data Volume Mapping here
MacOS Limitation



Demo

Exploring SQL Server and Containers





Update a SQL Server Container

Interacting with Data

Use your normal T-SQL and app to modify data

Use volumes or your changes will be lost if container removed

Configuration changes for SQL Server

Modify the container directly

Startup a new container with your configuration changes but same user db

```
sp_configure  
ALTER SERVER CONFIGURATION
```

Updating the build of SQL Server

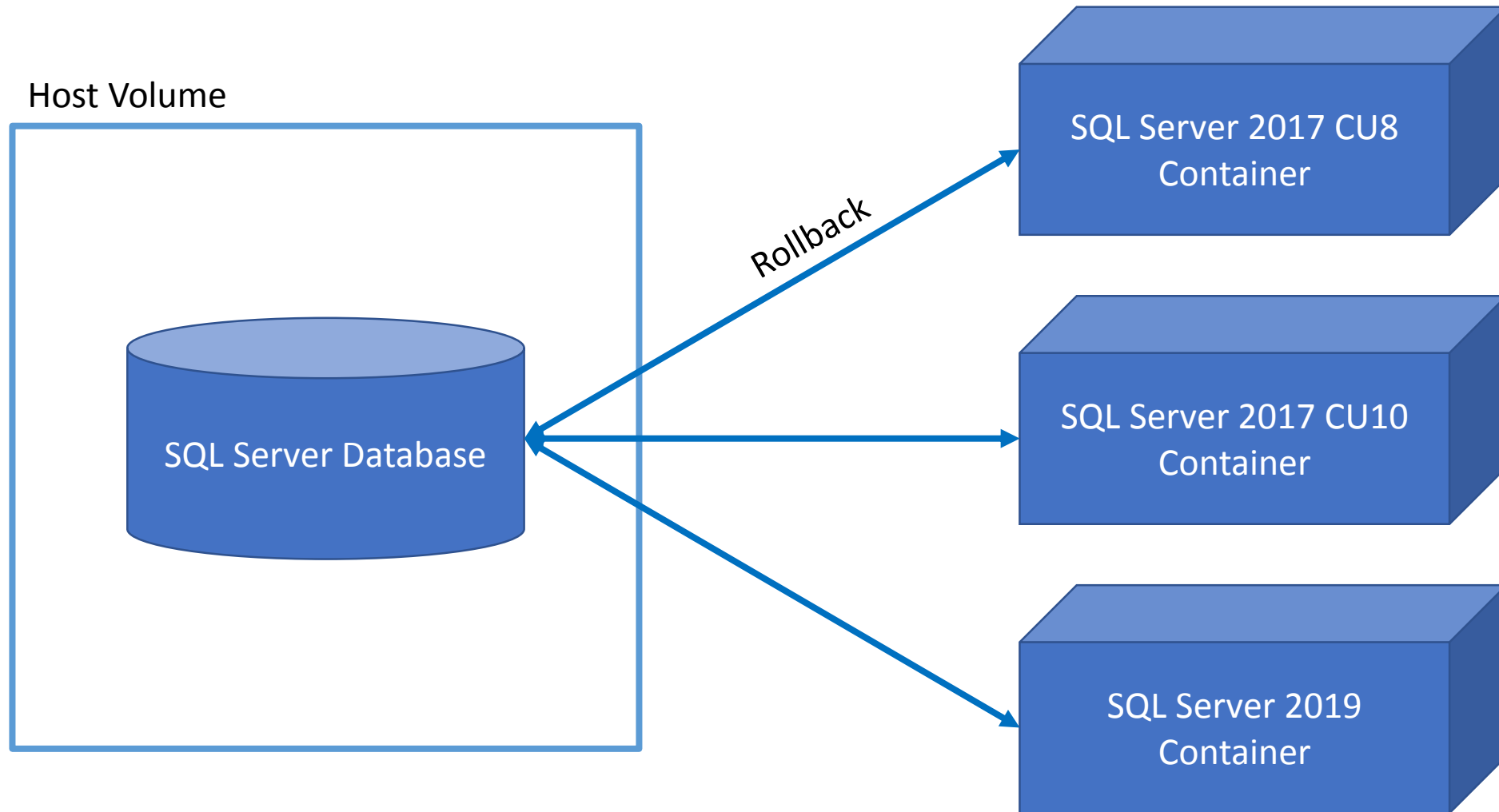
Shutdown current container

Start a new container with new updated image pointing to same volume

Rollback to previous container if on same major release of SQL Server

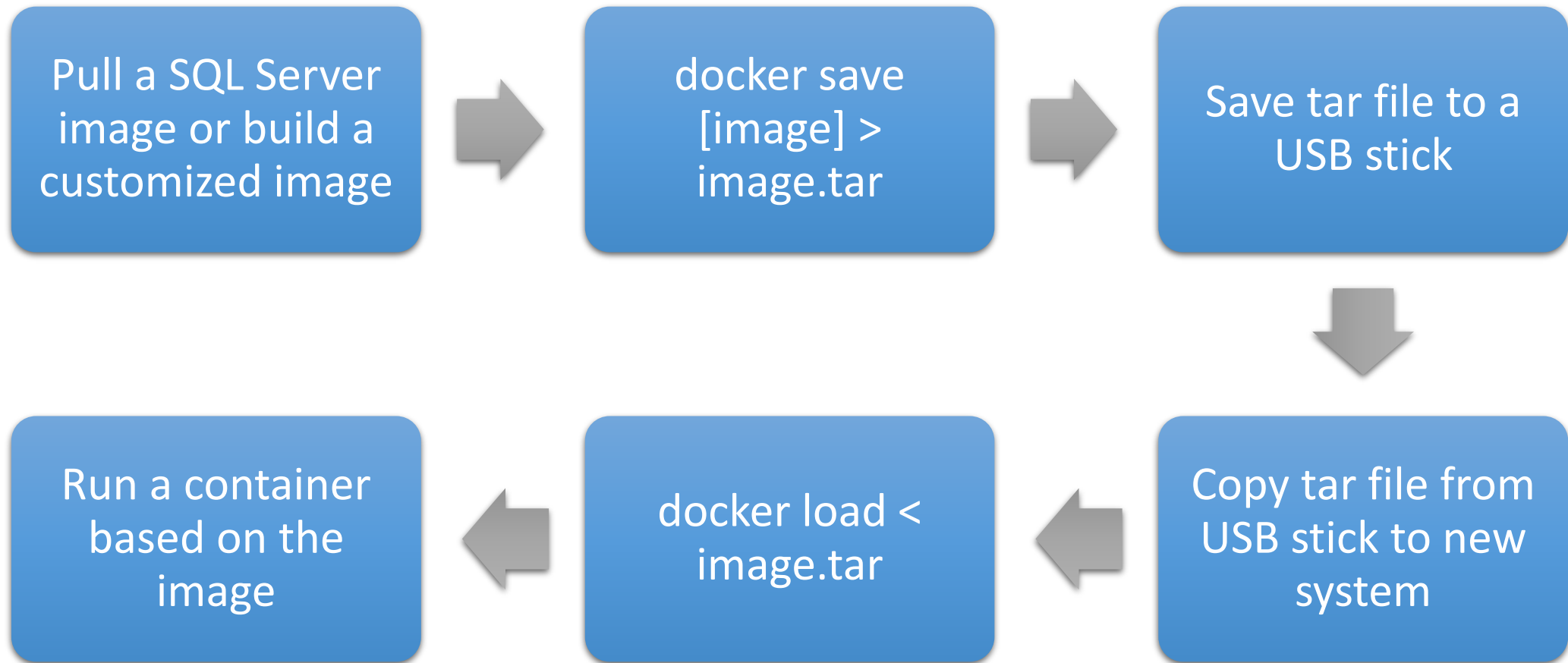


Upgrade a SQL Server





SQL Server on a Stick





Demo

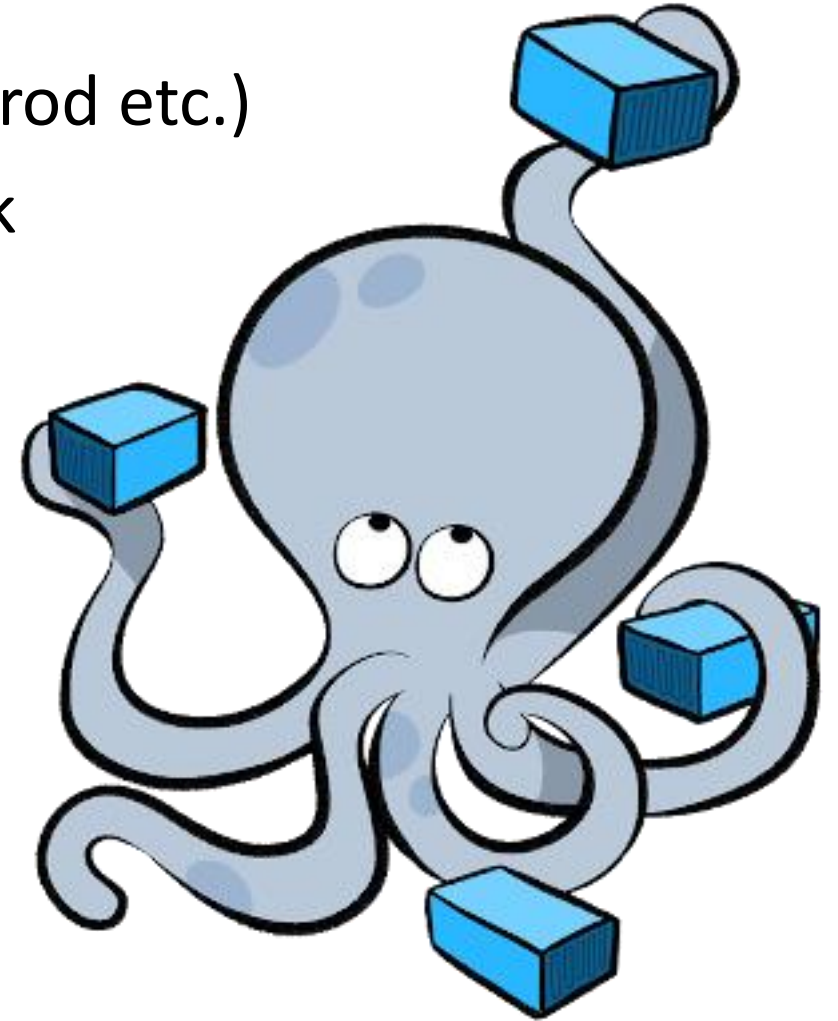
Updating and Upgrading
SQL Server Containers





Container Orchestration

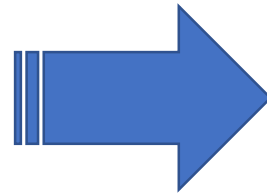
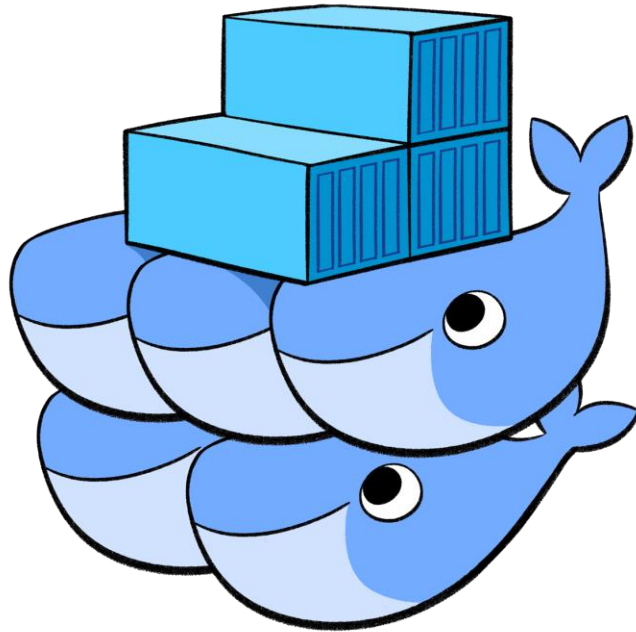
- Need for multiple environments (dev, test, prod etc.)
- Need for multiple servers (nodes) in network
- Need for Orchestration
 - Best machine to deploy to
 - Failover on Containers
 - Balance if new machines are added
 - Failover on Nodes





Container Orchestration

Docker Swarm



Kubernetes





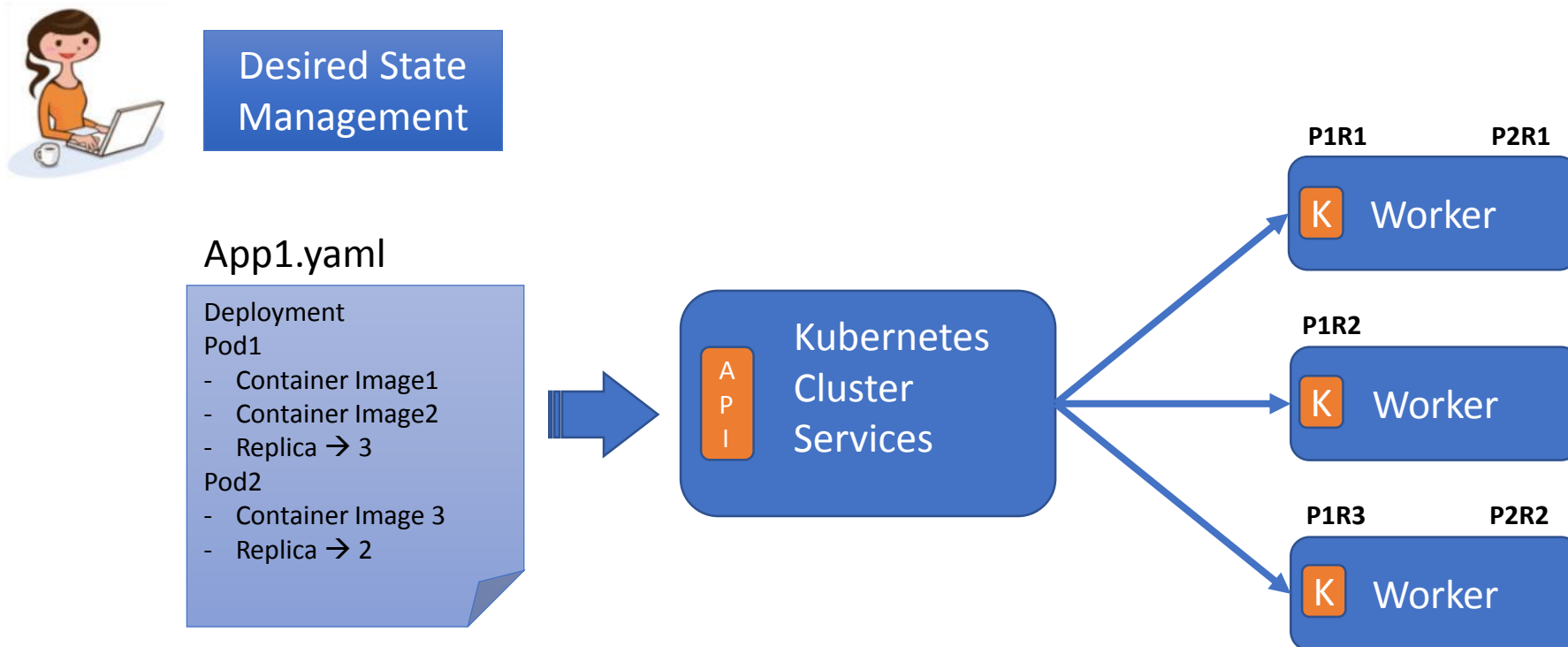
Kubernetes



- Orchestration
- Automate
 - Deploying, Scaling, Monitoring
- K8s is a platform for working with **Containers** (not only Dockers)
- Helm and Helm Charts
 - Help to define, package, install, upgrade, roll-back, delete
- Azure Kubernetes Service (AKS)
 - Builds on Helm Charts – operation assistance
 - Azure Context



Kubernetes Basics



K = Kubelets



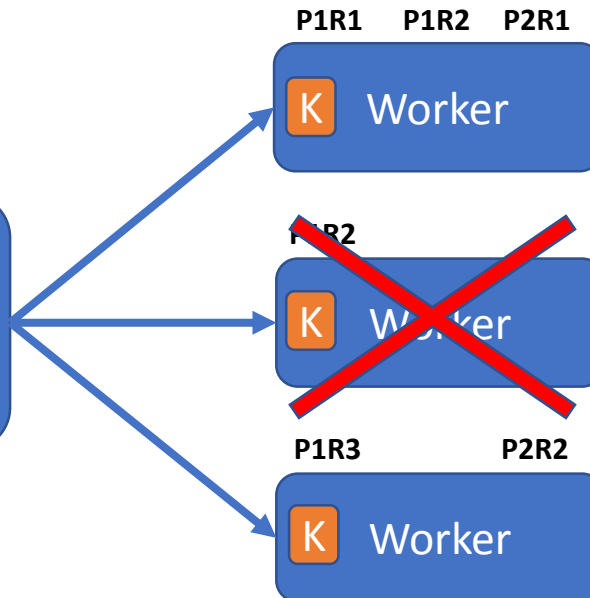
Kubernetes Basics (fail)



Desired State Management

App1.yaml

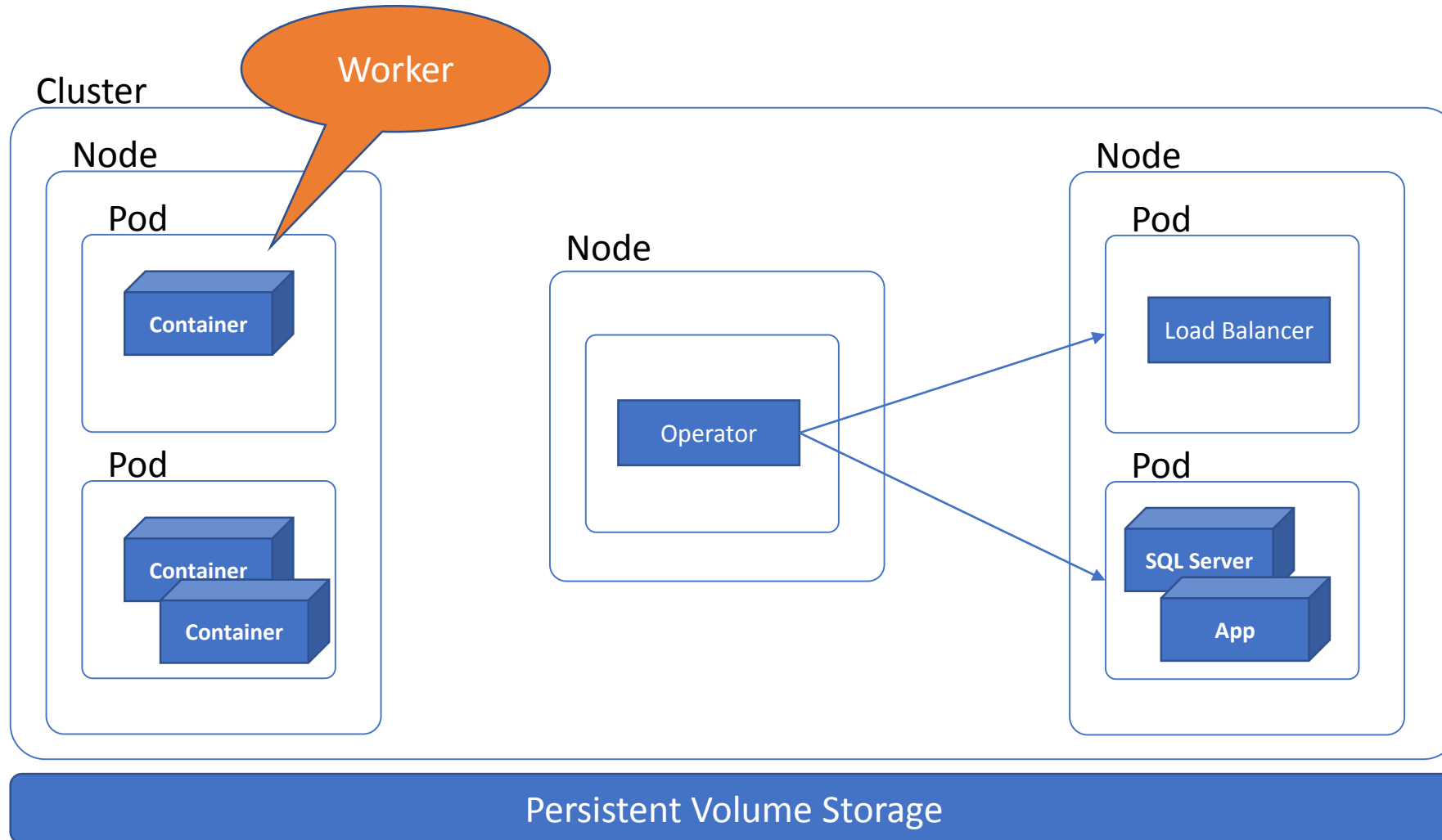
```
Deployment
Pod1
- Container Image1
- Container Image2
- Replica → 3
Pod2
- Container Image 3
- Replica → 2
```



K = Kubelets

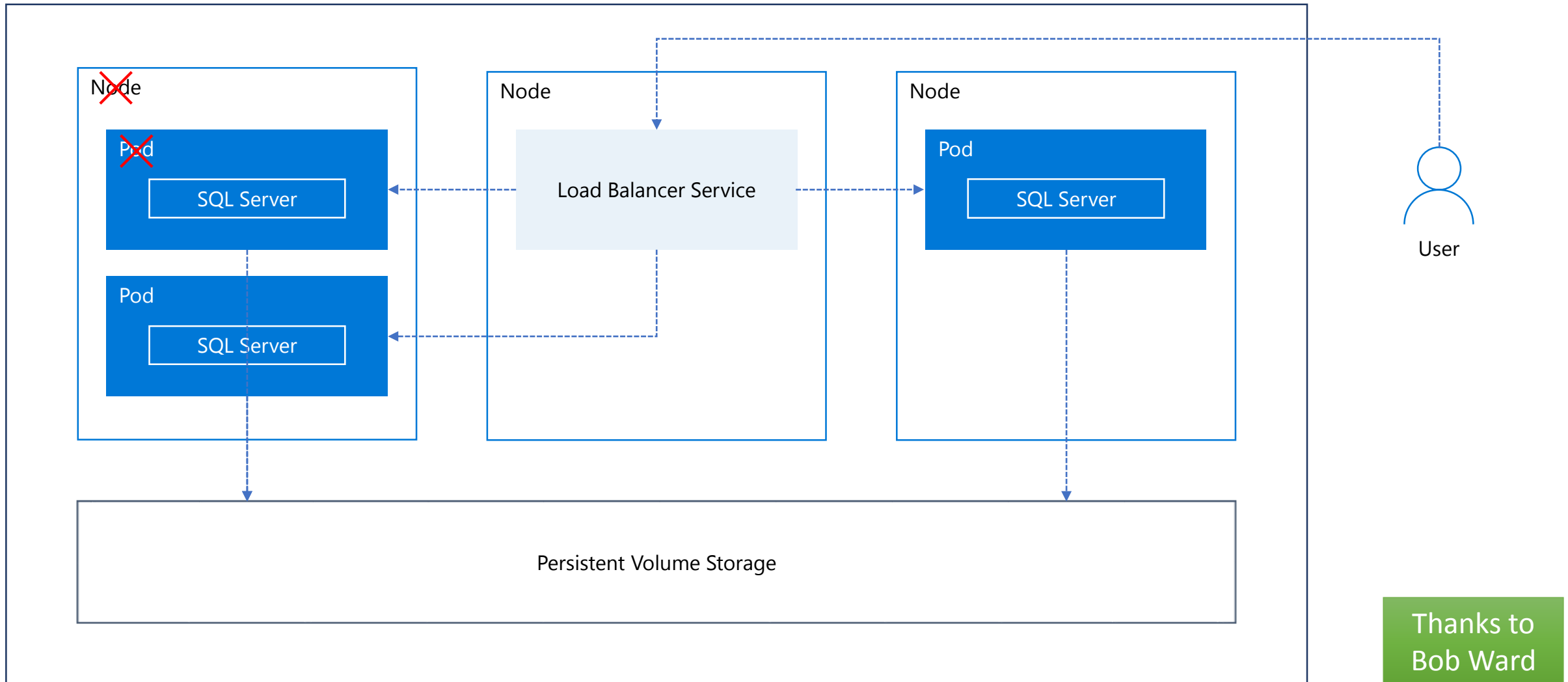


K8s Architectural structure





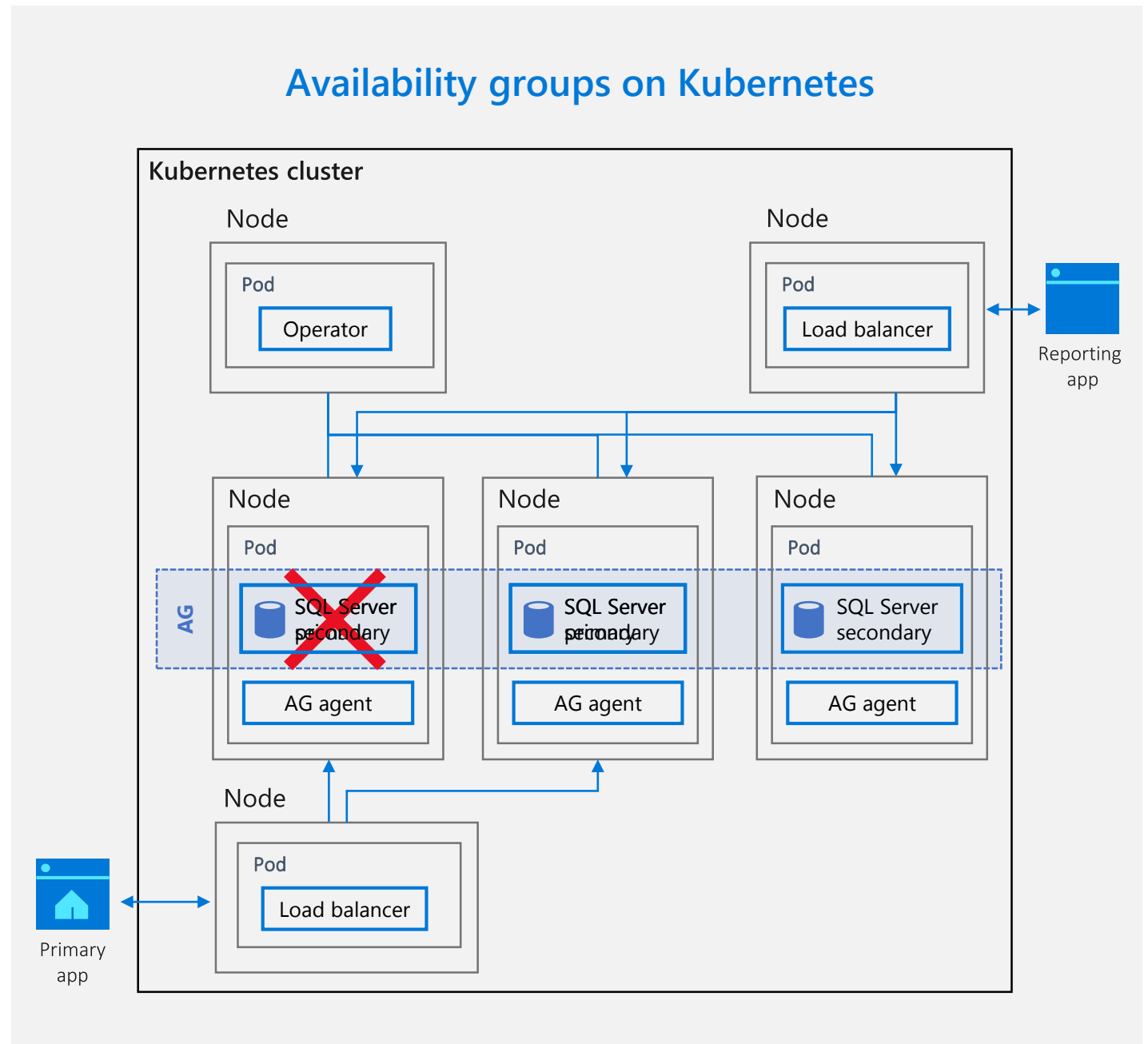
SQL Server 2017 – HA in K8s



Thanks to
Bob Ward

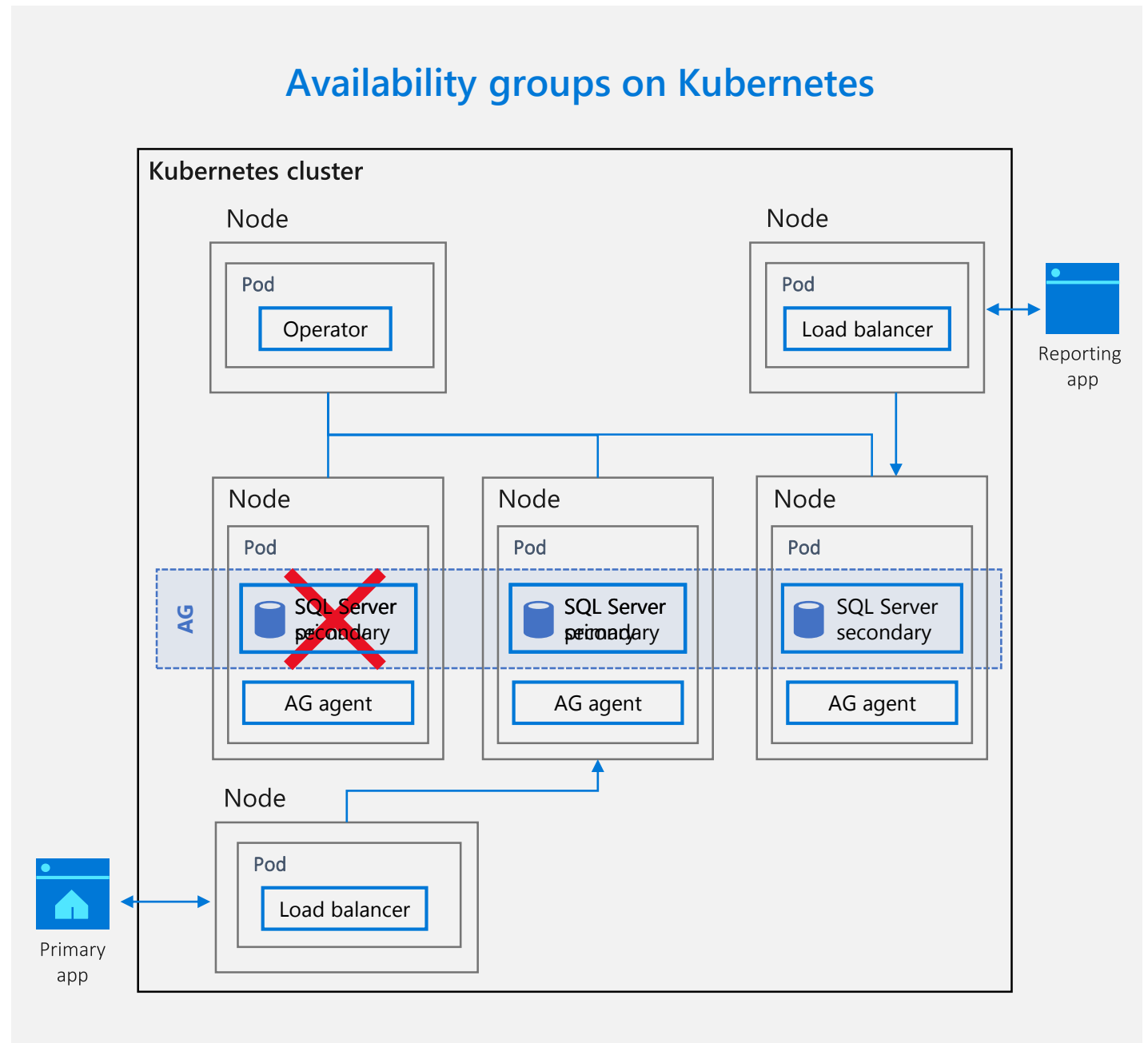
SQL Server 2019 Always On Availability Groups on Kubernetes

- Operator orchestrates and deploys
- AG concepts all apply
- Load Balancer for Primary App
- Load Balancer for Secondary Replica Readers



SQL Server 2019 Always On Availability Groups on Kubernetes

- Operator orchestrates and deploys
- AG concepts all apply
- Load Balancer for Primary App
- Load Balancer for Secondary Replica Readers



Ressources

- <https://aka.ms/sqlcontainers>
- <https://github.com/Microsoft/sqllinuxlabs>
- <https://hub.docker.com/r/microsoft/mssql-server/>
- <https://docs.docker.com/engine/docker-overview/>
- <https://docs.docker.com/reference/>
- <https://docs.docker.com/engine/docker-overview/#docker-architecture>
- <https://kubernetes.io/>





Questions





Pragmatic BI

Pragmatic BI Aps

Åhusene 4,9,3

8000 Aarhus C

Telefon +45 42 606 202

E-mail steen@pragbi.dk

Internet www.pragbi.dk

CVR nr. 36974133