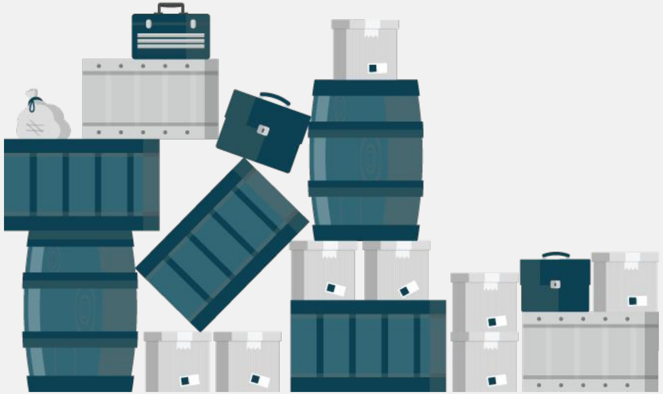
OpenShift Architecture Workshop

Intro

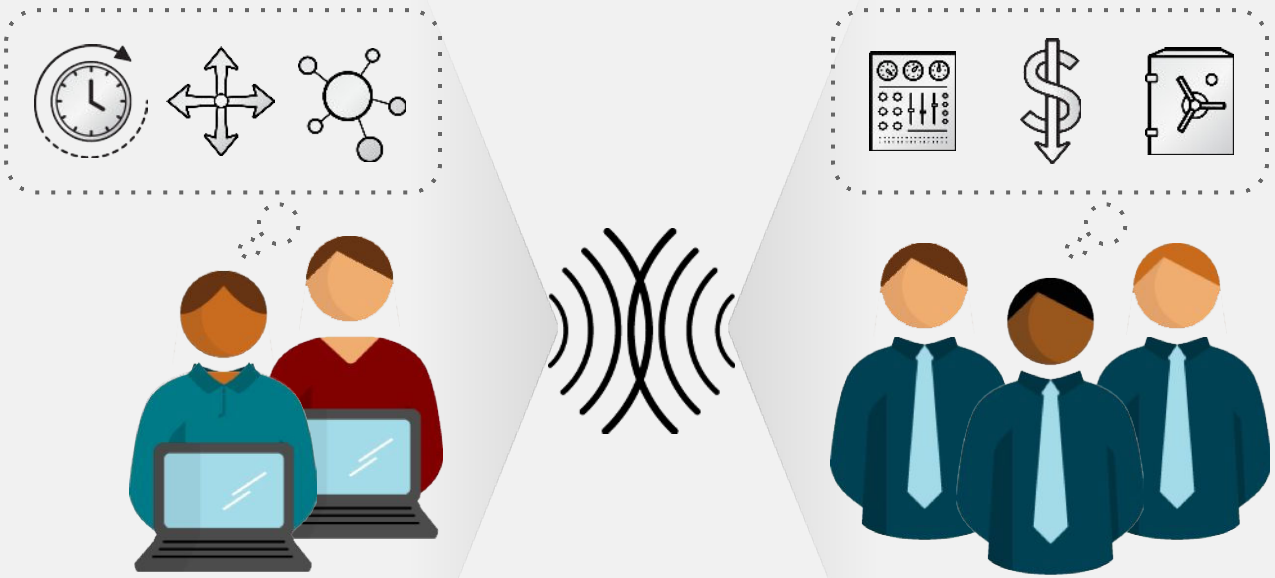
Wanja Pernath
EMEA Partner Enablement Manager, Middleware & OpenShift
12. September 2017

The Problem

Applications require complicated installation and integration every time they are deployed



THE PROBLEM



DEVELOPERS

I.T. OPERATIONS

DEVOPS

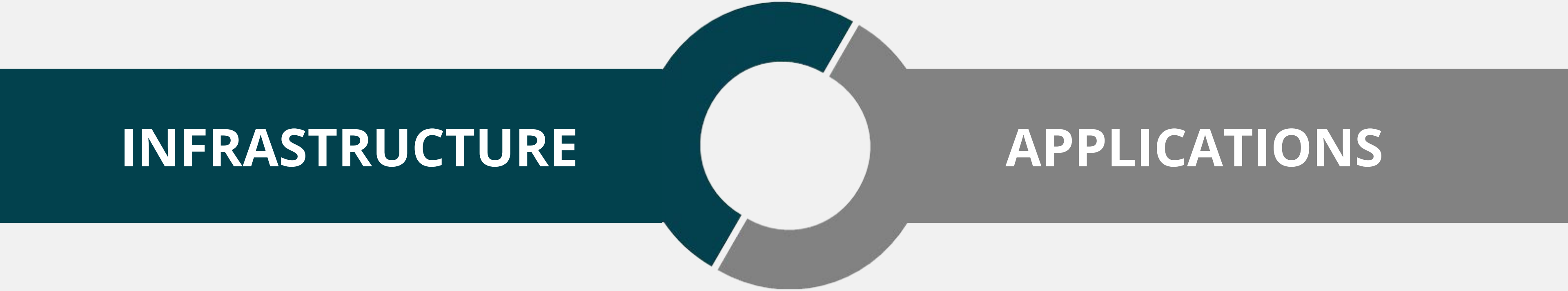
- Everything as code
- Automate everything
- Continuous Integration/Delivery
- Application is always “releaseable”
- Application monitoring
- Rapid feedback
- Rebuild vs. Repair
- Delivery pipeline

The Solution

Adopting a container strategy will allow applications to be easily shared and deployed.

WHAT ARE CONTAINERS?

It Depends Who You Ask



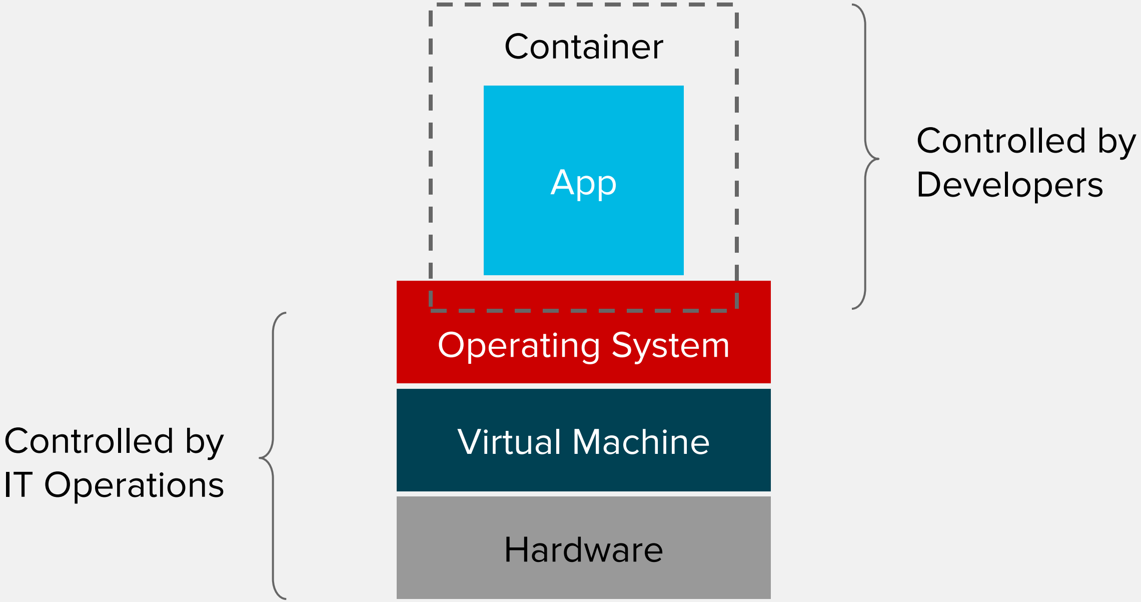
INFRASTRUCTURE

- Sandboxed application processes on a shared Linux OS kernel
- Simpler, lighter, and denser than virtual machines
- Portable across different environments

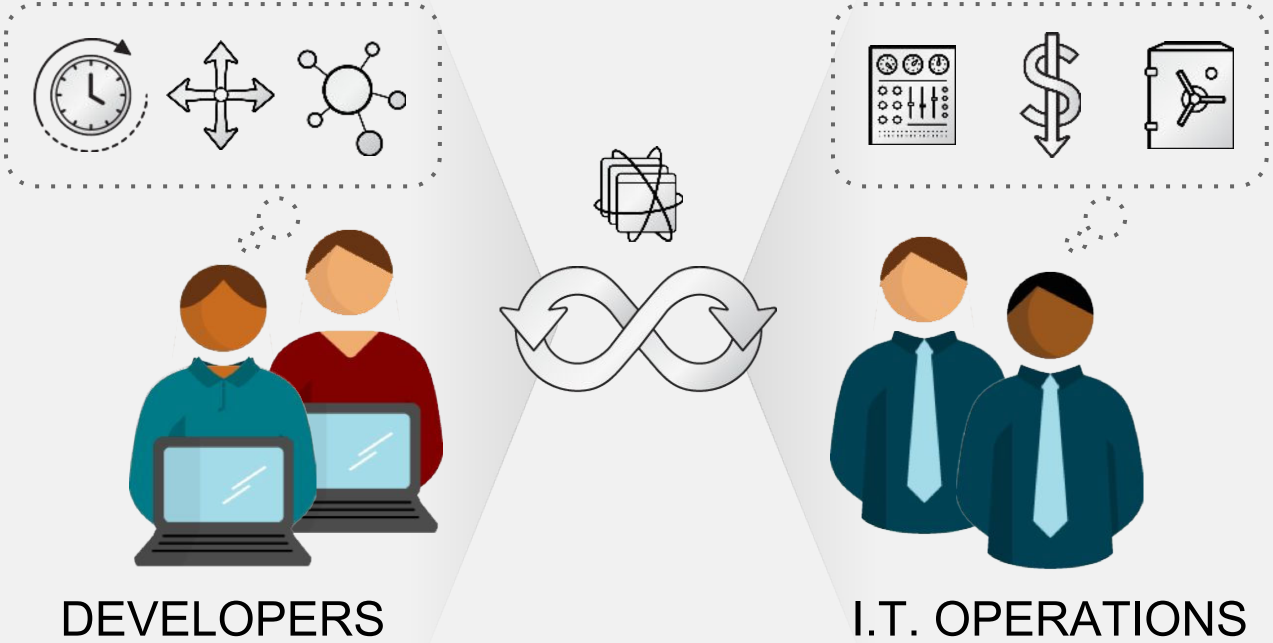
APPLICATIONS

- Package my application and all of its dependencies
- Deploy to any environment in seconds and enable CI/CD
- Easily access and share containerized components

THE SOLUTION

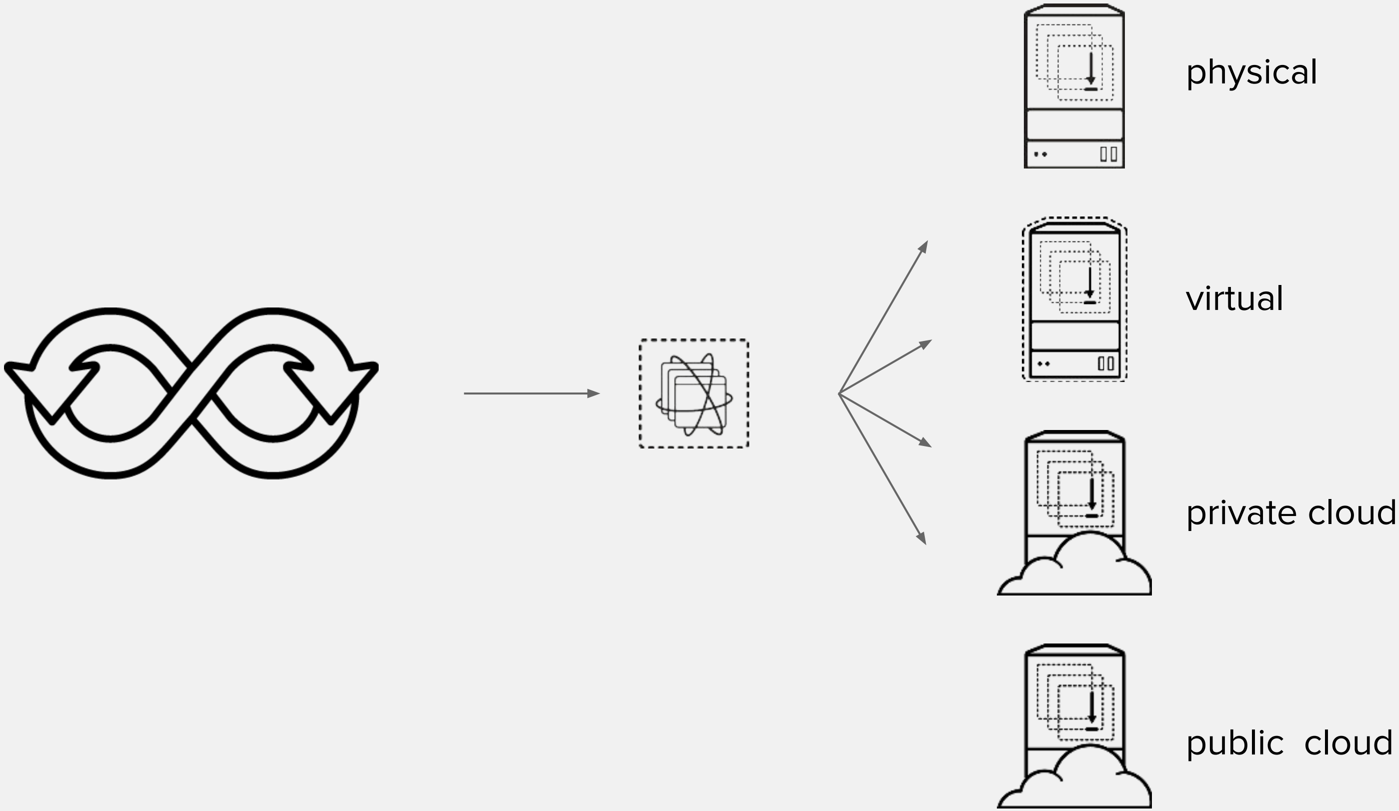


THE SOLUTION



```
$ docker build -t app:v1 .

$ docker run app:v1
```



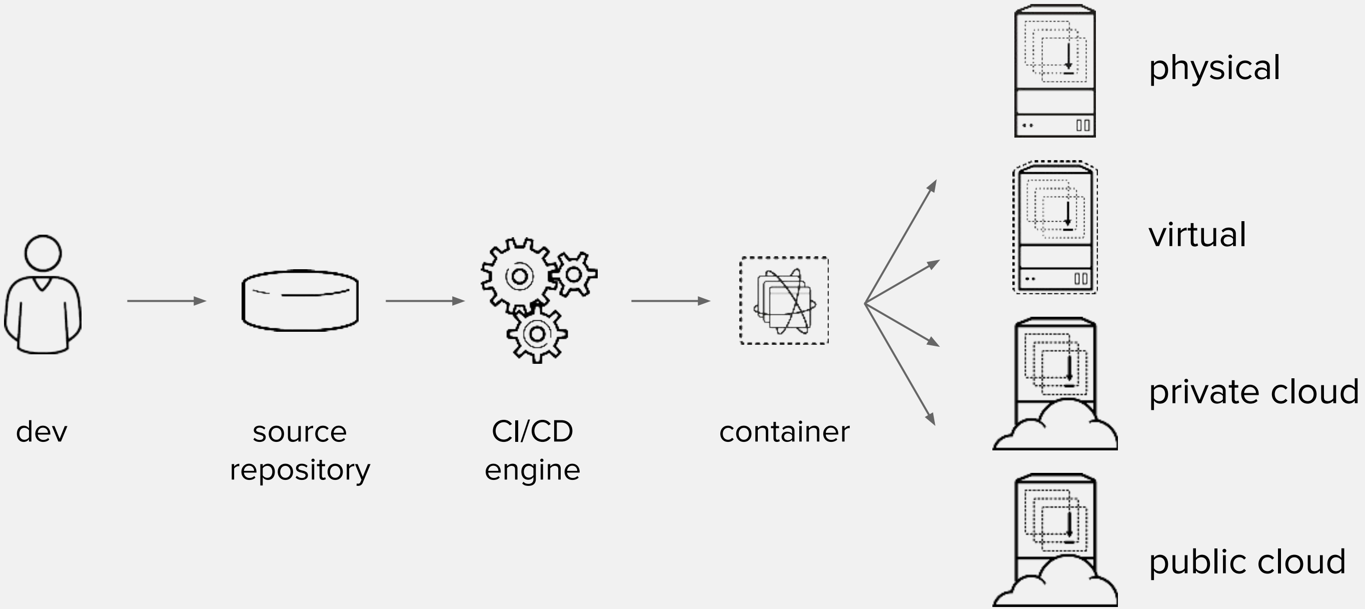
physical

virtual

private cloud

public cloud

DEVOPS WITH CONTAINERS



dev

source repository

CI/CD engine

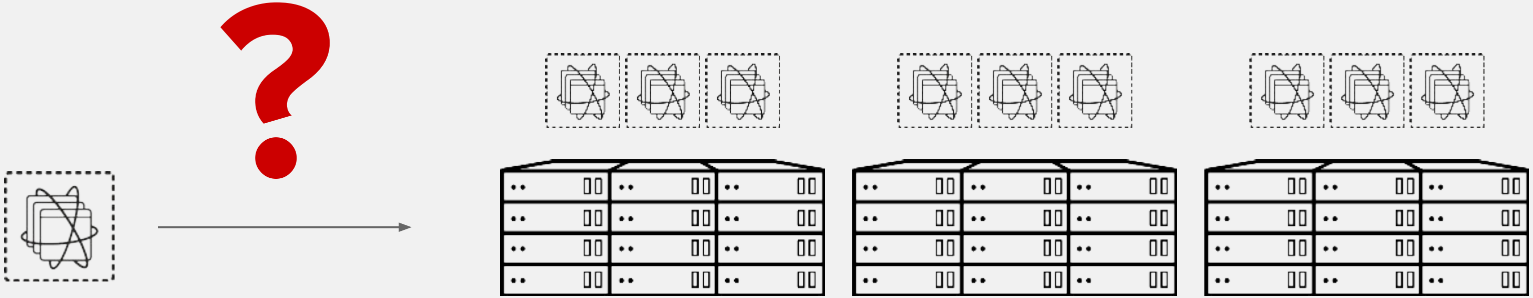
container

physical

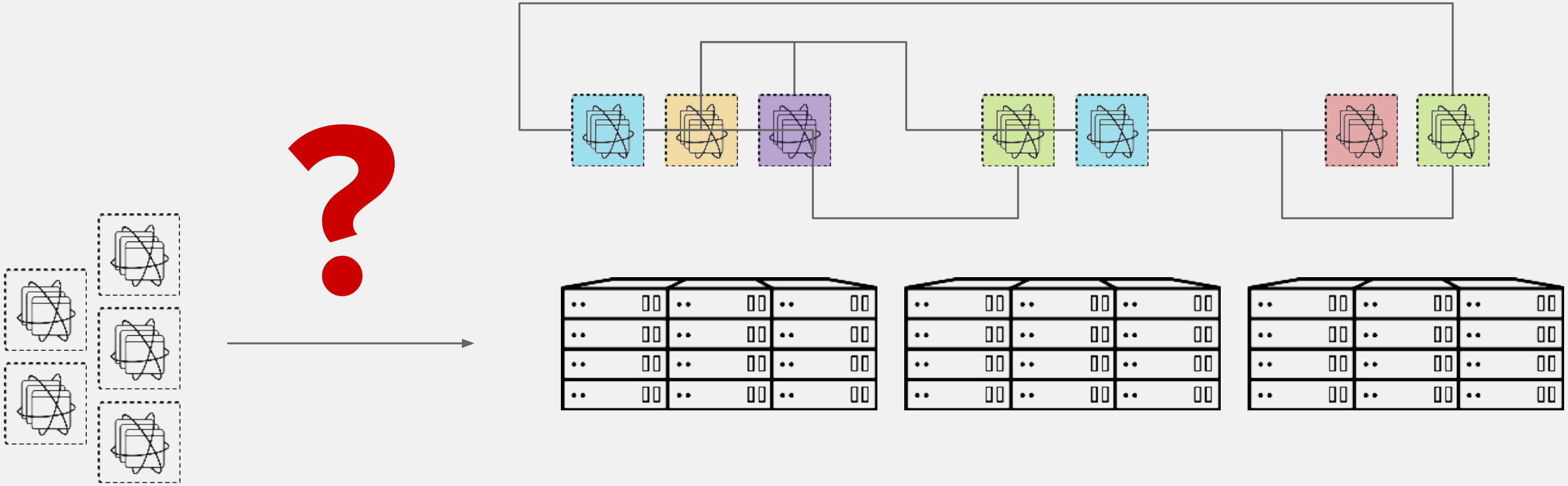
virtual

private cloud

public cloud



```
$ docker build -t app/frontend:v1 .
$ docker build -t app/backend:v1 .
$ docker build -t app/database:v1 .
$ docker build -t app/cache:v1 .
$ docker build -t app/messaging:v1 .
$ docker run app/frontend:v1 link-to-backend
$ docker run app/frontend:v1 link-to-backend
$ docker run app/backend:v1 link-to-db-cache-messaging
$ docker run app/backend:v1 link-to-db-cache-messaging
$ docker run app/database:v1
$ docker run app/cache:v1 link-to-db
$ docker run app/messaging:v1
```



WE NEED MORE THAN JUST CONTAINERS

Scheduling

Decide where to deploy containers

Lifecycle and health

Keep containers running despite failures

Discovery

Find other containers on the network

Monitoring

Visibility into running containers

Security

Control who can do what

Scaling

Scale containers up and down

Persistence

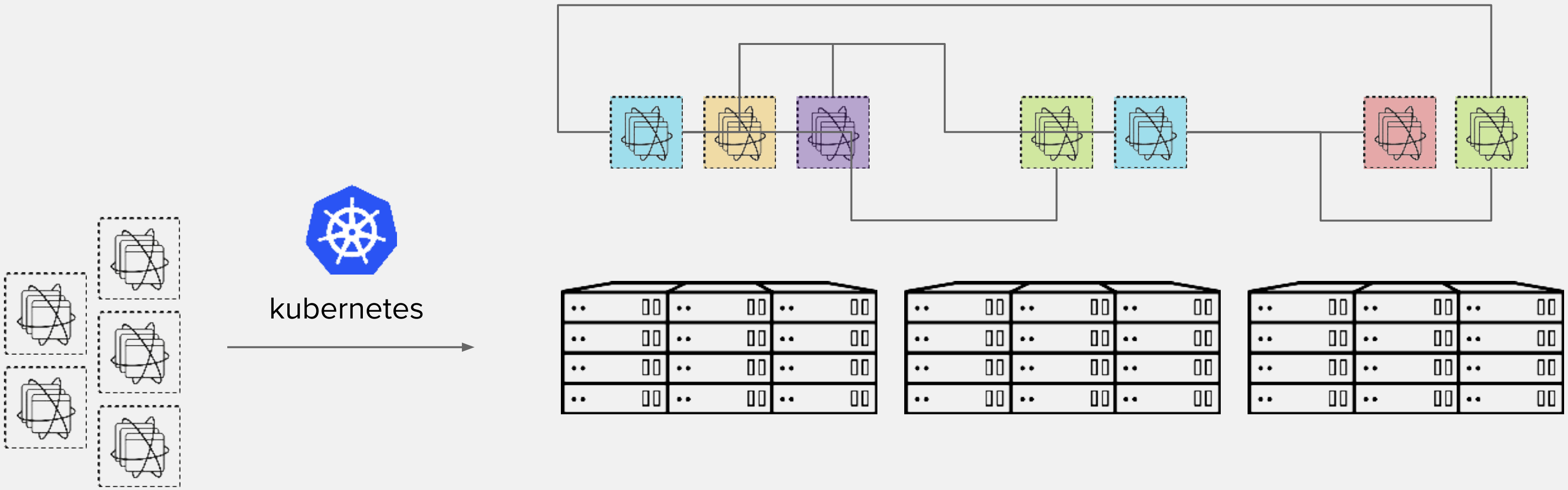
Survive data beyond container lifecycle

Aggregation

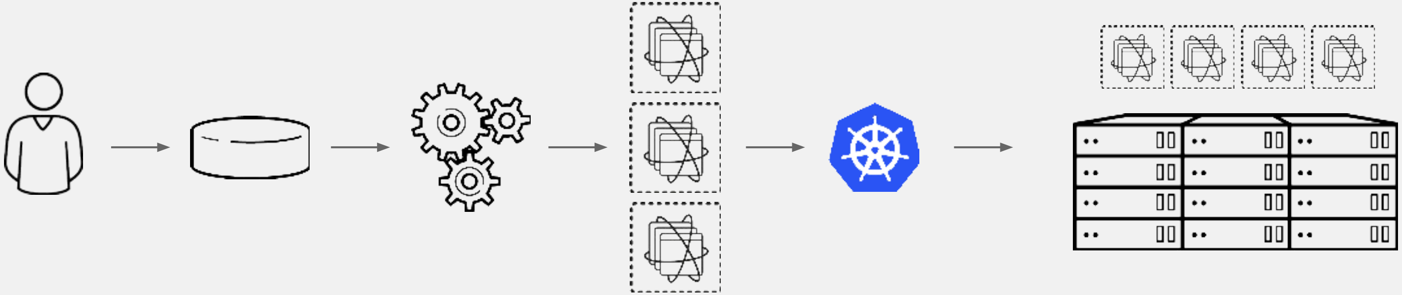
Compose apps from multiple containers

Kubernetes is an open-source system for automating deployment, operations, and scaling of containerized applications across multiple hosts

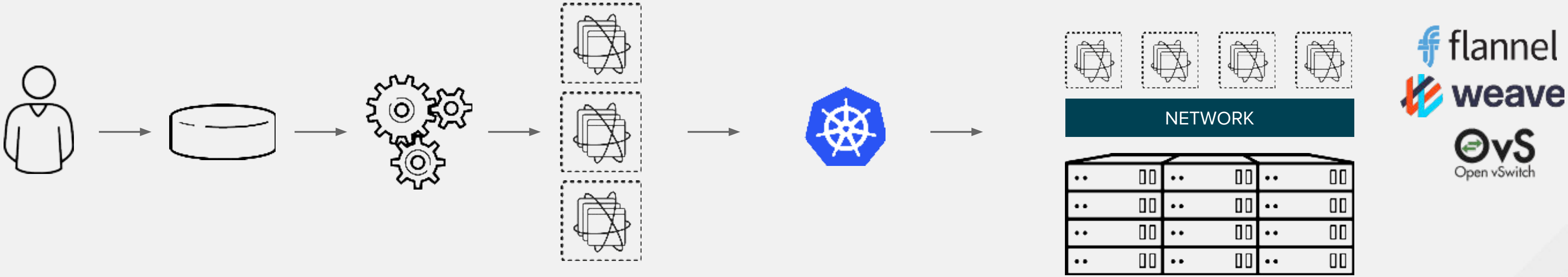
kubernetes


kubernetes

DEVOPS WITH
CONTAINERS AND KUBERNETES

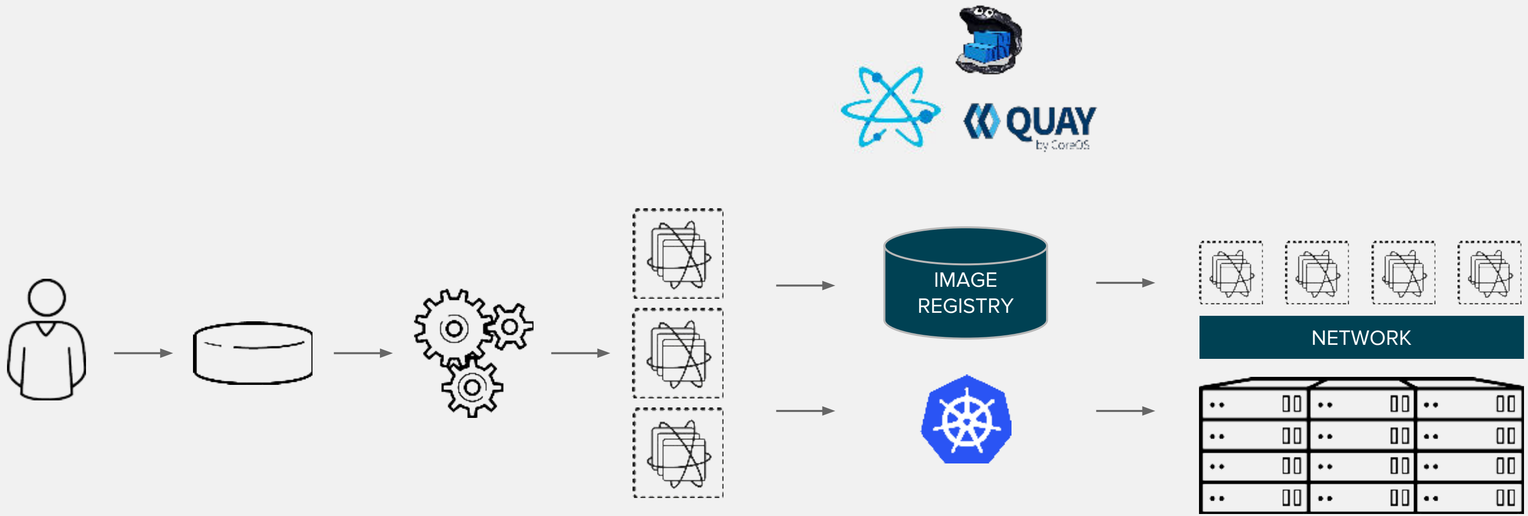


DEVOPS WITH
CONTAINERS AND KUBERNETES



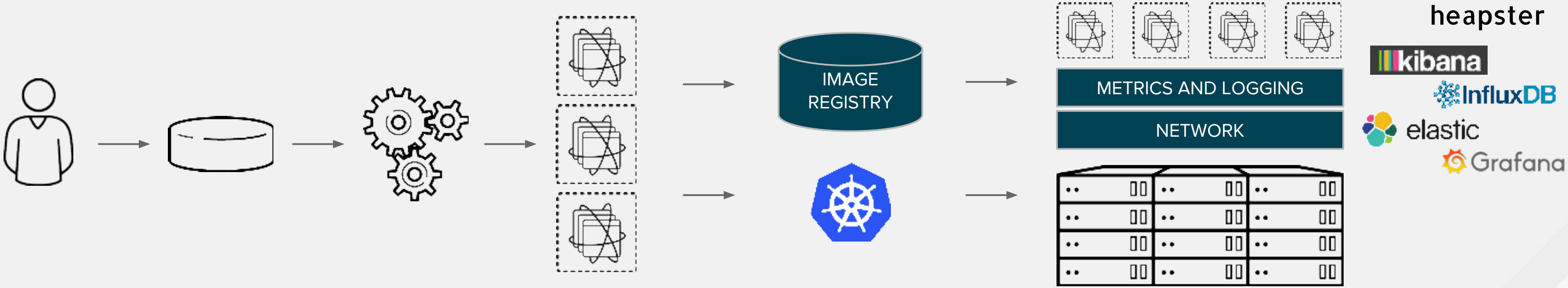
Not enough! Need networking

DEVOPS WITH CONTAINERS AND KUBERNETES



Not enough! Need an image registry

DEVOPS WITH CONTAINERS AND KUBERNETES



Not enough! Need metrics and logging

DEVOPS WITH
CONTAINERS AND KUBERNETES

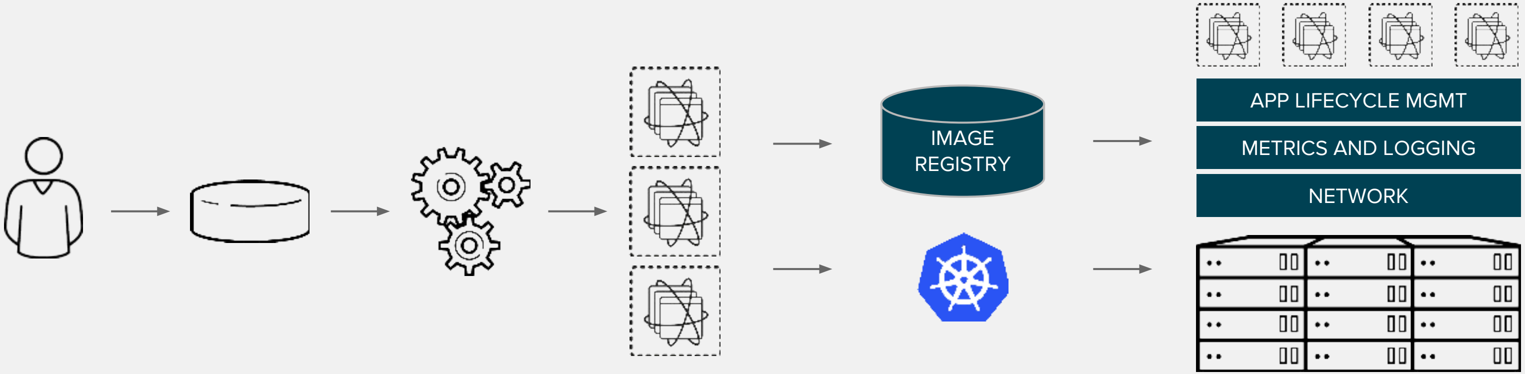


IMAGE REGISTRY

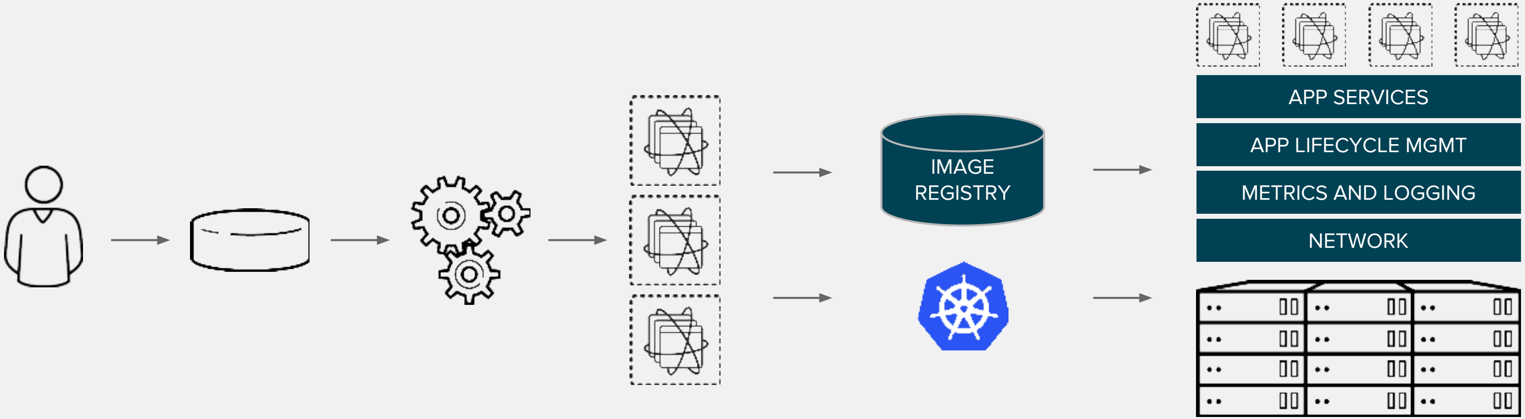
APP LIFECYCLE MGMT

METRICS AND LOGGING

NETWORK

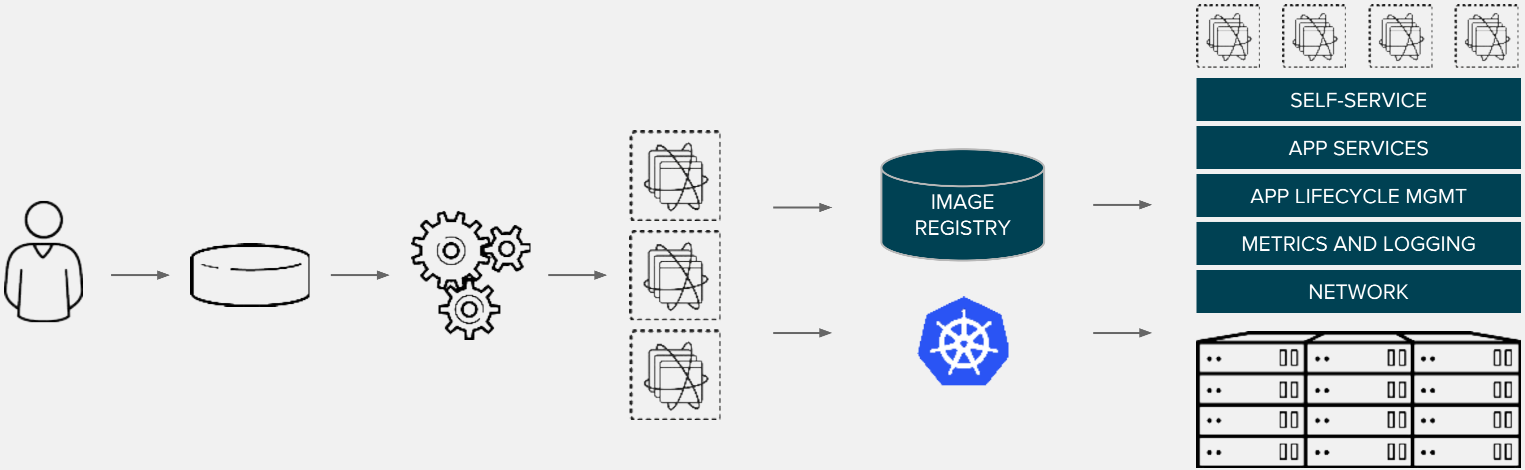
Not enough! Need application lifecycle management

DEVOPS WITH CONTAINERS AND KUBERNETES



Not enough! Need application services e.g. database and messaging

DEVOPS WITH
CONTAINERS AND KUBERNETES



Not enough! Need self-service portal

NOT ENOUGH, THERE IS MORE!

- Multi-tenancy
- Routing & Load Balancing
- CI/CD Pipelines
- Role-based Authorization
- Capacity Management
- Infrastructure Visibility
- Teams and Collaboration
- Quota Management
- Image Build Automation
- Container Isolation
- Vulnerability Scanning
- Chargeback

Container application platform based on Docker and Kubernetes for building, distributing and running containers at scale

RED HAT® OPENSHIFT

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos