A CONTAINERS STATE OF MIND

Chris Van Tuin
Chief Technologist, West
cvantuin@redhat.com

redhat.

促进软件开发领域知识与创新的传播

InfoQueue

ArchSummit
全球架构师峰会

[深圳] 2015年7月17日-18日

QCon
全球软件开发大会

[上海] 2015年10月15-17日

关注InfoQ官方微信
及时获取QCon演讲视频信息

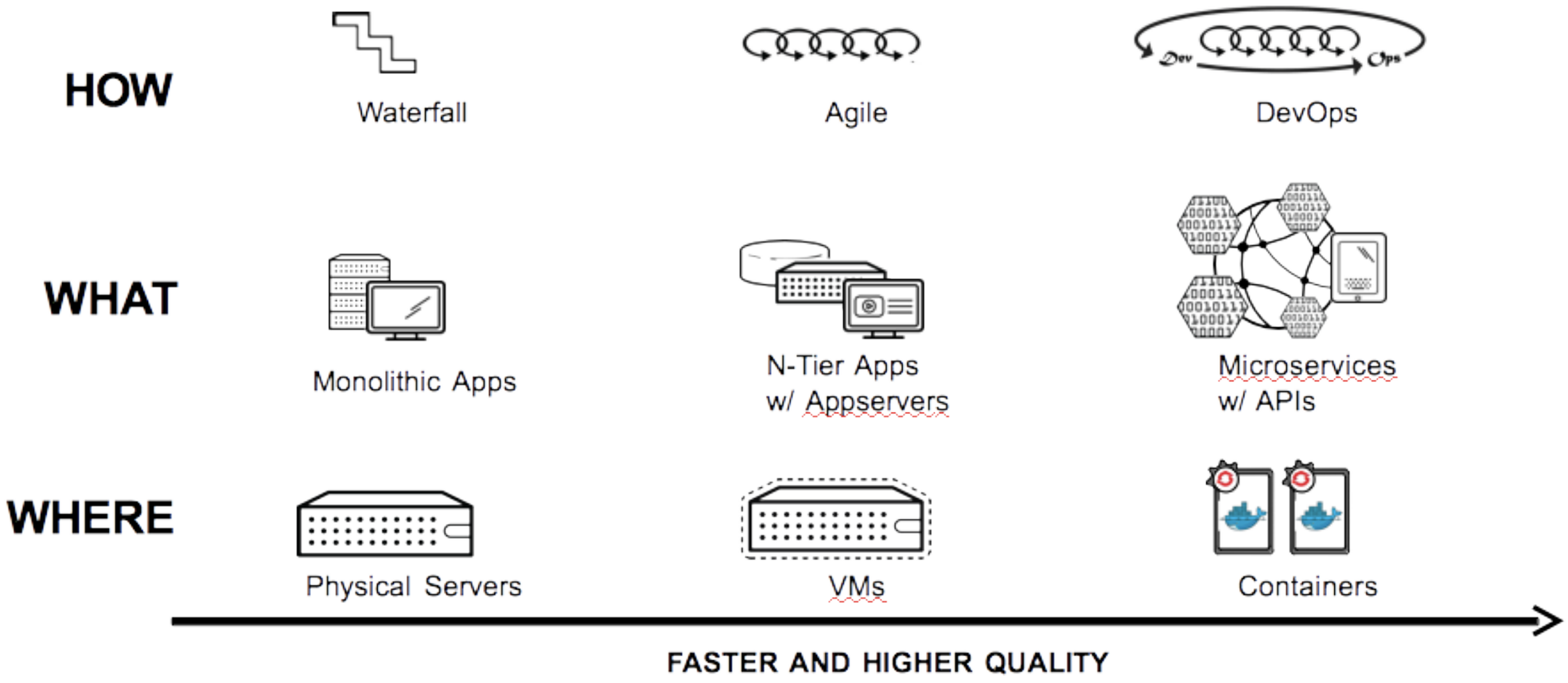
THERE ARE THE WINDS OF CHANGE
AND THEN THERE IS ACCELERATING CHANGE

“…every strategic inflection point [is] characterized by a ’10x’ change… There’s wind and then there is a typhoon”

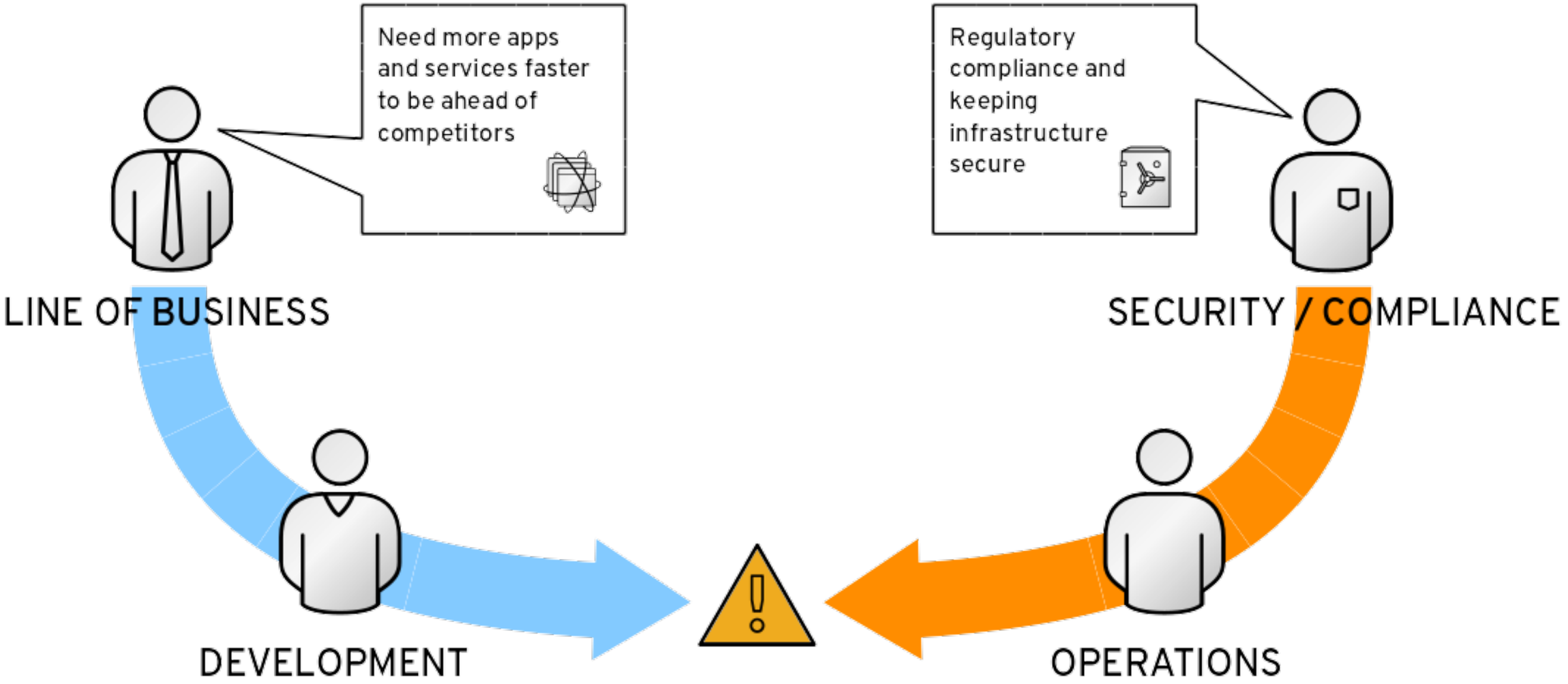
-Andy Grove

THE NEED FOR SPEED

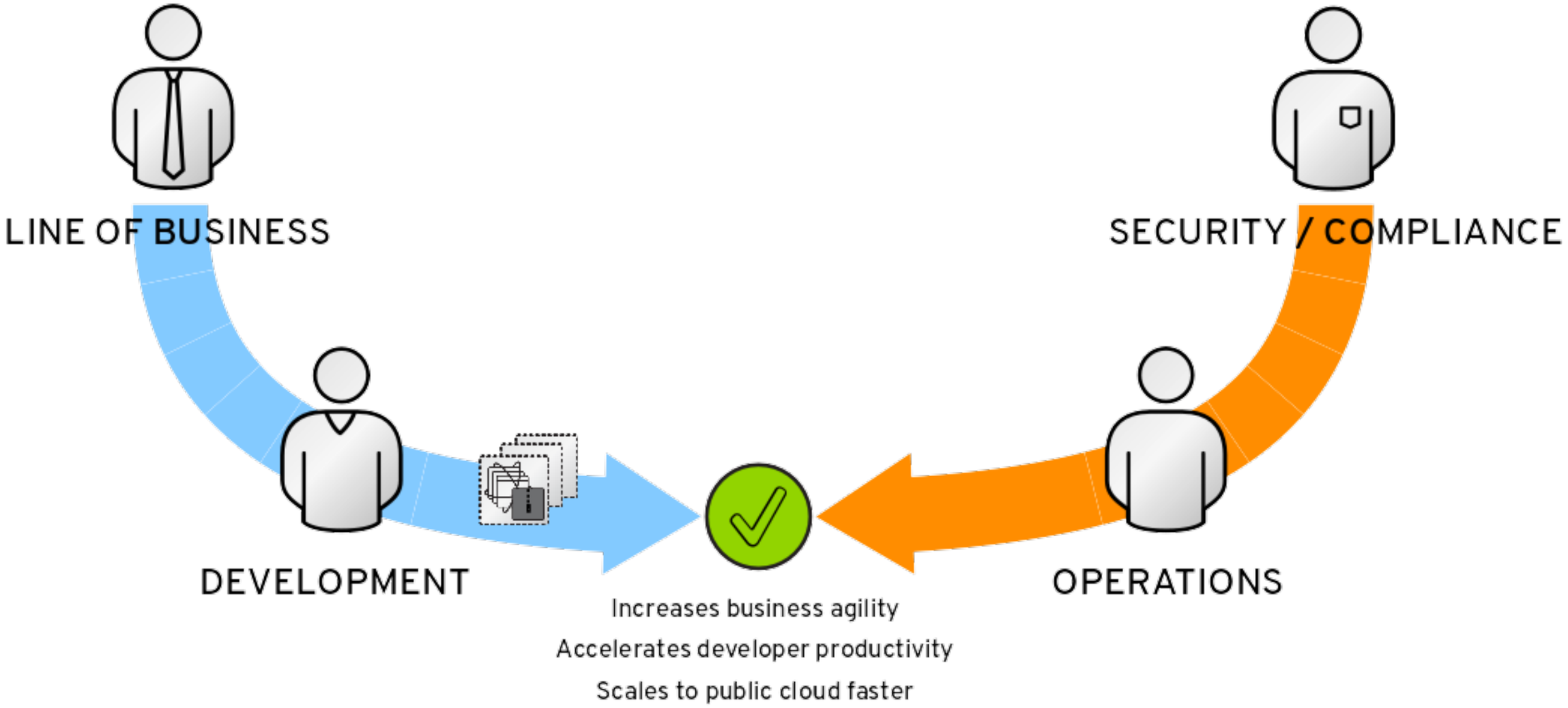
THE ACCELERATION OF APPLICATION DELIVERY FOR THE BUSINESS



THE PROBLEM: FRICTION

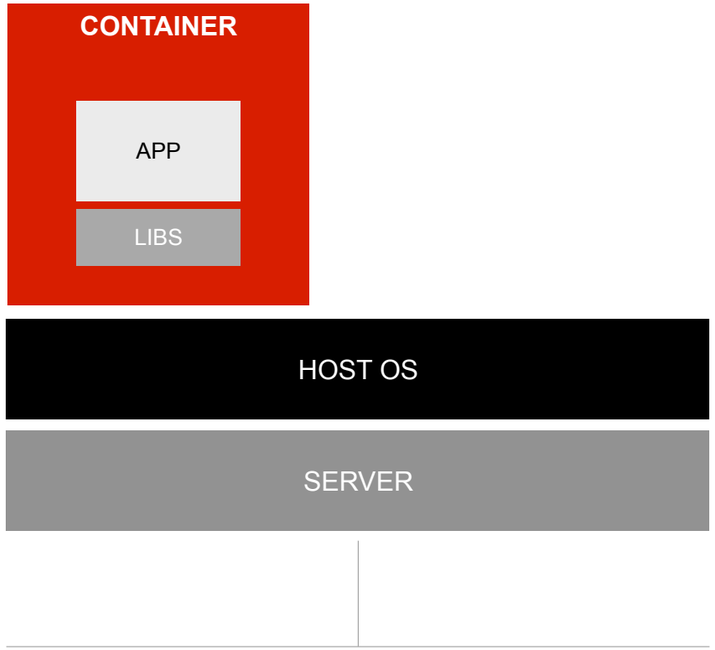


APPLICATION DELIVERY VIA CONTAINERS

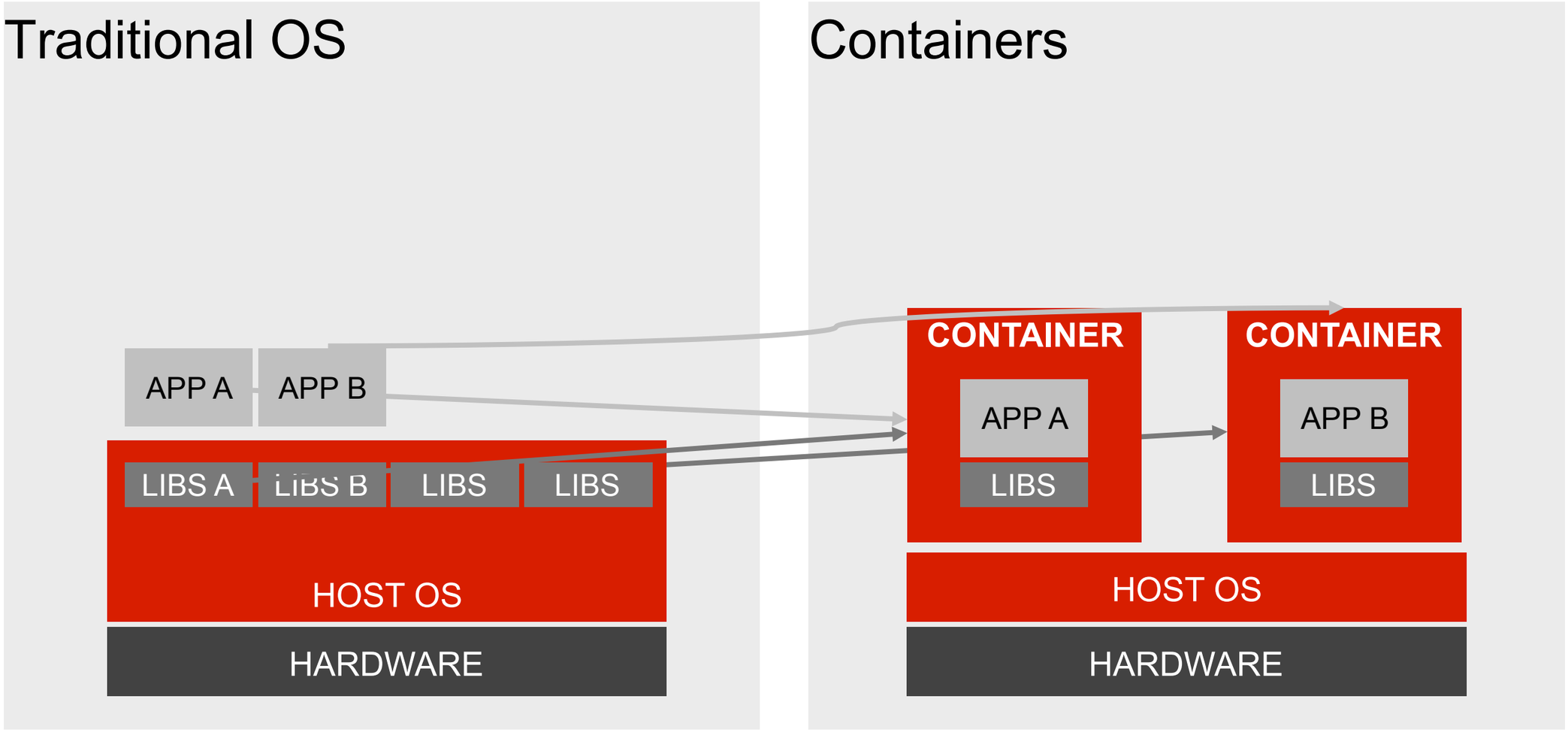


WHAT ARE LINUX CONTAINERS?

- Used to create containers for software applications / microservices
- Containers provide lightweight isolation of process, network, filesystem spaces
- Package Once Deploy Anywhere
- Docker builds on Linux containers, adds an API, image format, runtime, and a delivery and sharing model

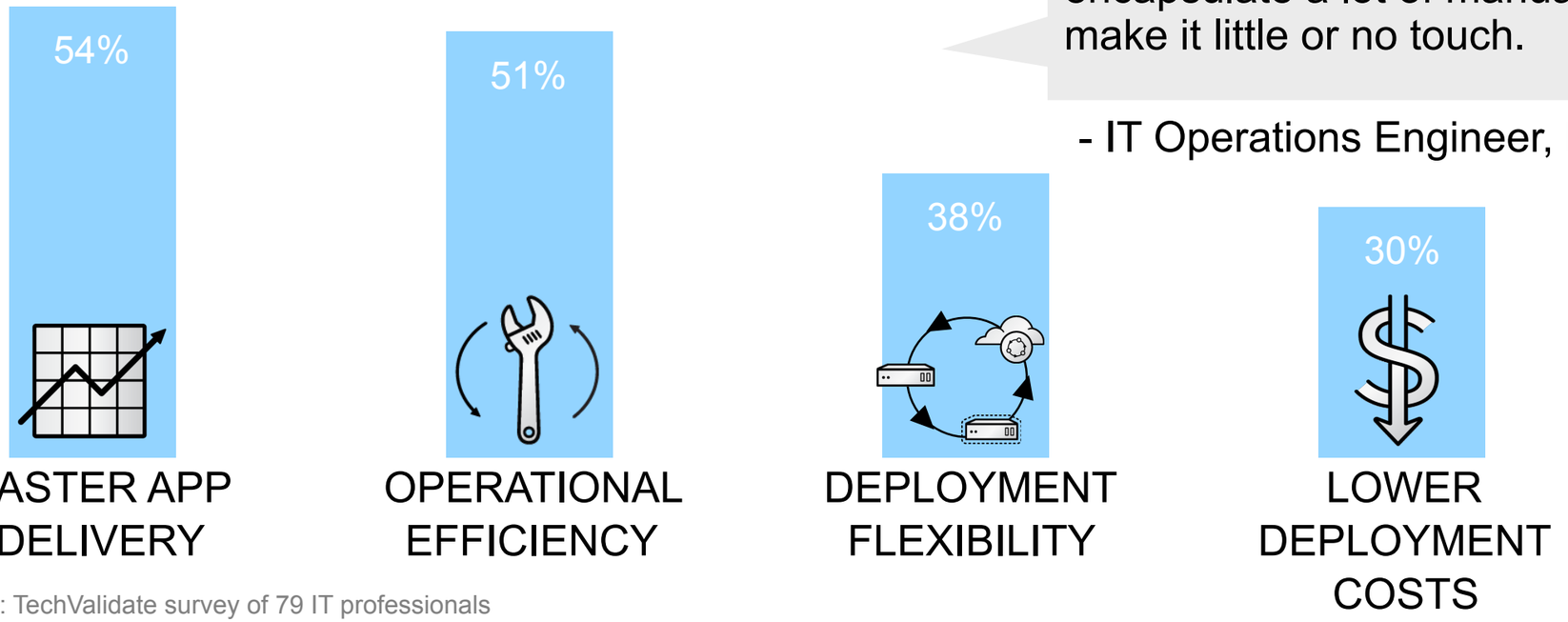


TRADITIONAL OS VS CONTAINERS



MANY SEE CONTAINERS AS THE UTOPIA OF APPLICATION DELIVERY

"What top benefits do you see with containers?"

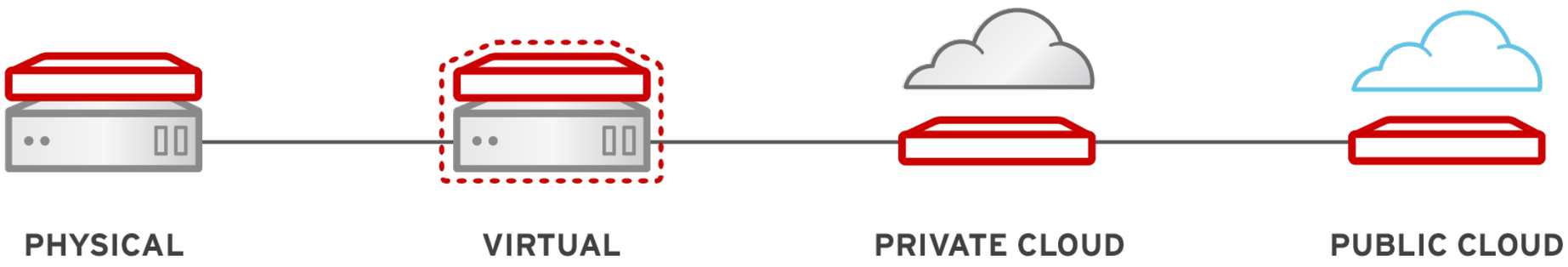


Source: TechValidate survey of 79 IT professionals

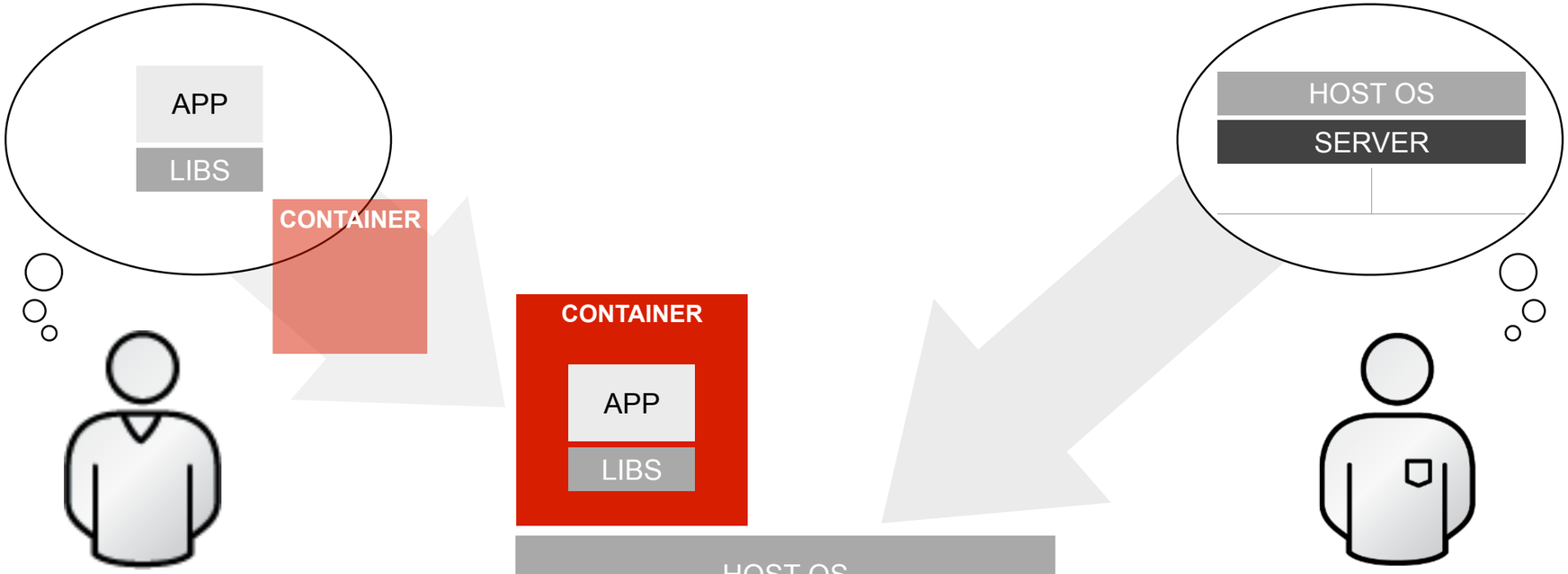
> Containers potentially offer the ability to encapsulate a lot of manual processes and make it little or no touch.
>
> - IT Operations Engineer, Financial Services

Containers transform the way you deliver applications to accelerate innovation.

PORTABILITY and FLEXIBILITY



FASTER APP DELIVERY and EFFICIENCY



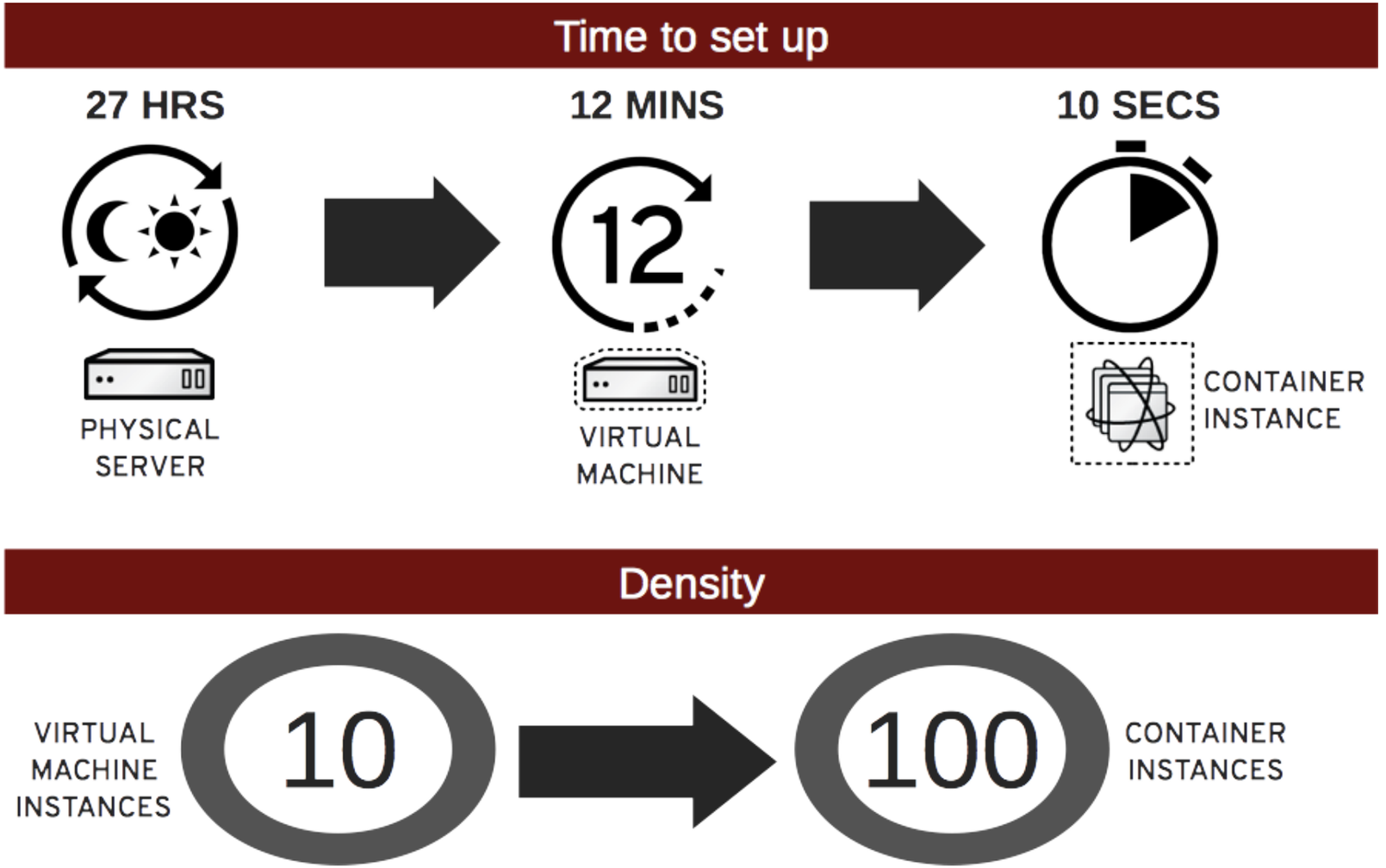
DEVELOPMENT FOCUS

- CODE
- APPLICATIONS
- LIBRARIES AND OTHER DEPENDENCIES
- DATA STORE

OPERATIONS FOCUS

- HARDWARE
- OPERATING SYSTEM
- NETWORKING
- MONITORING

VELOCITY and DENSITY



BENEFITS OF CONTAINERS (cont)

- *Portability*
- *Faster app deployment*
- *Velocity*
- *Density*
- Isolation
- Snapshotting
- Security sandbox
- Limit resource usage
- Simplified dependency
- Sharing

UNDERLYING TECHNOLOGY

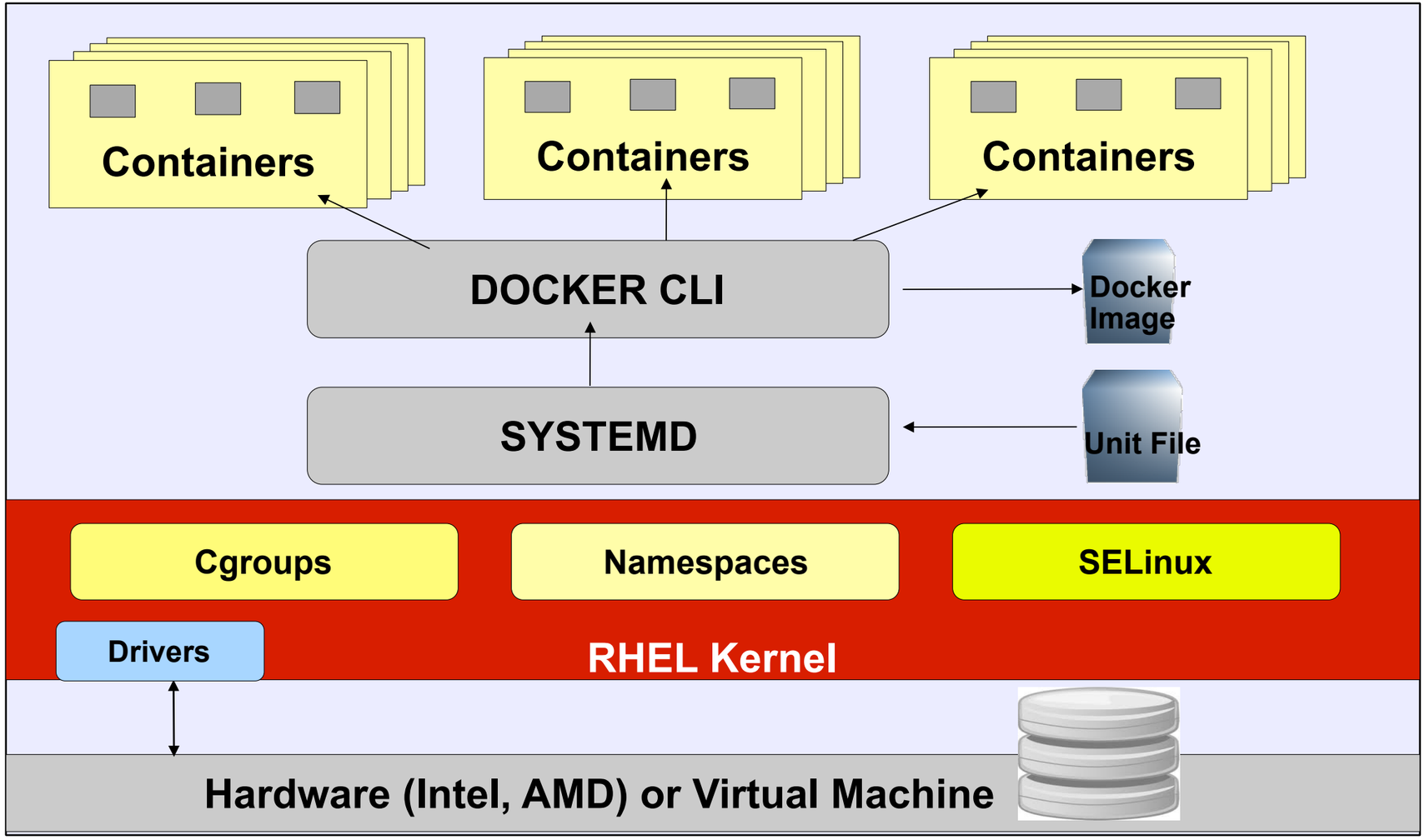
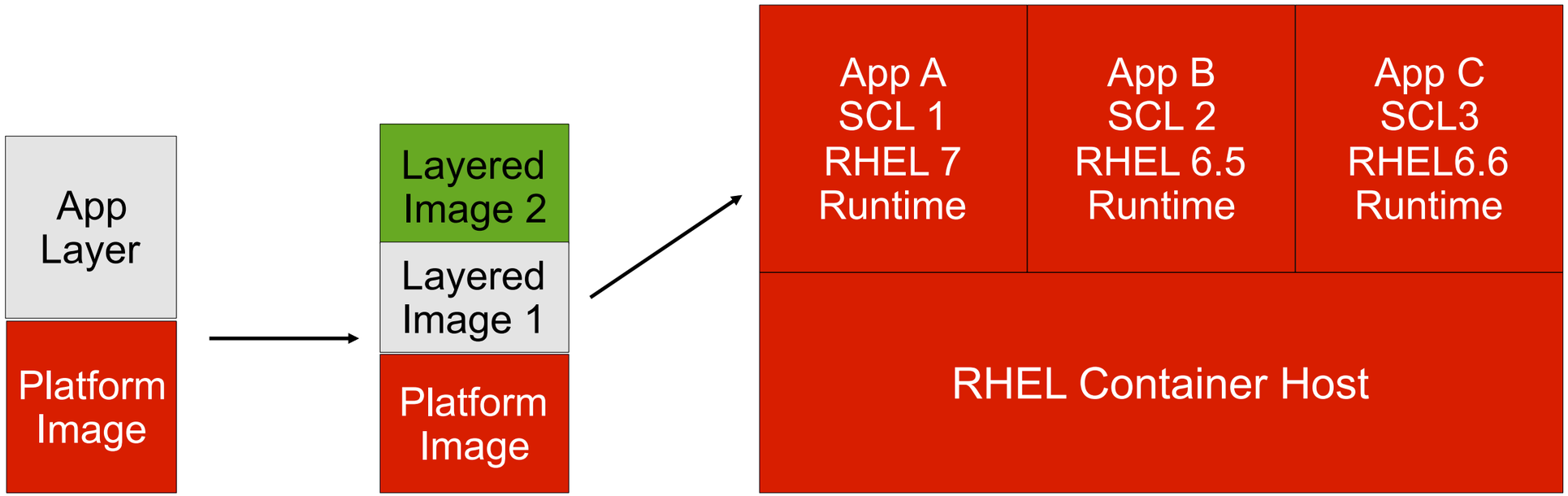


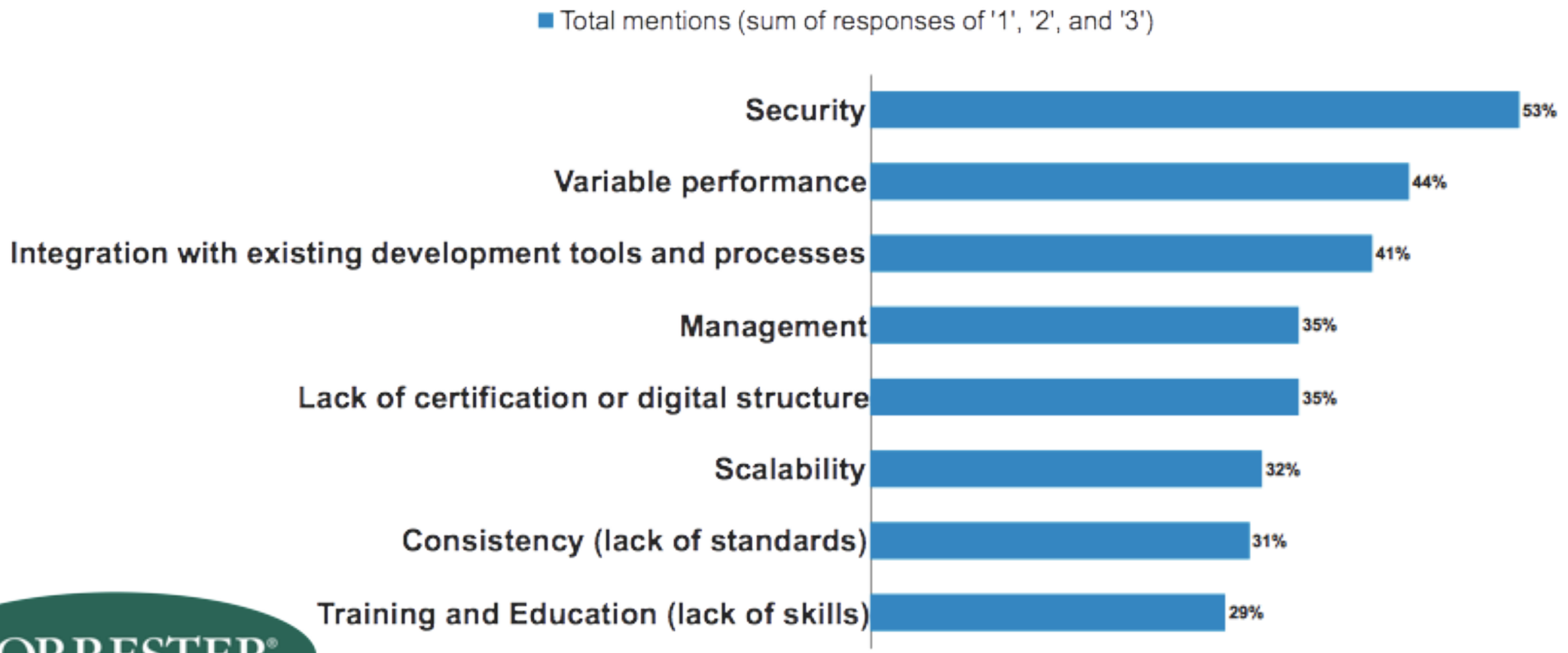
IMAGE -BASED CONTAINERS WITH DOCKER TECHNOLOGY



RHEL 7 Container Host provides the capability to run images built with Docker format for content distribution

TOP CURRENT CONTAINER CHALLENGES

What are the top three challenges your organization has experienced so far in its use of containers?



FORRESTER®

Base: 1711 IT and Developer/programmer decision-makers at companies with 500+ employees in APAC, EMEA, and NA
Source: A commissioned study conducted by Forrester Consulting on behalf of Red Hat, January 2015

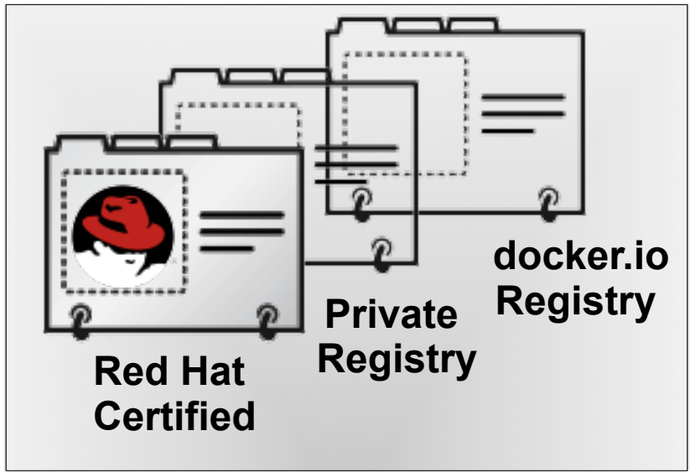
redhat.

CONTAINERS

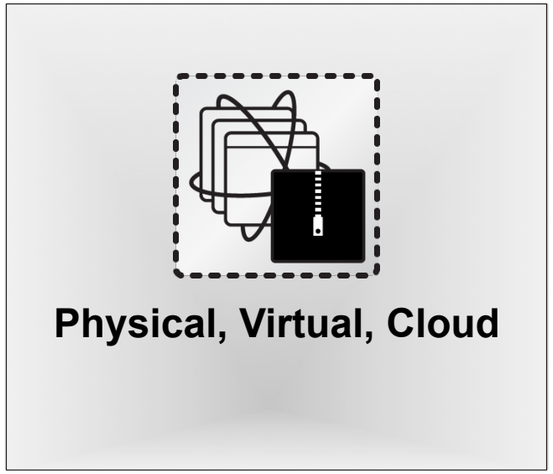
BUILD, SHIP, RUN

Dockerfile

Image



Container



Build

“docker build or commit”

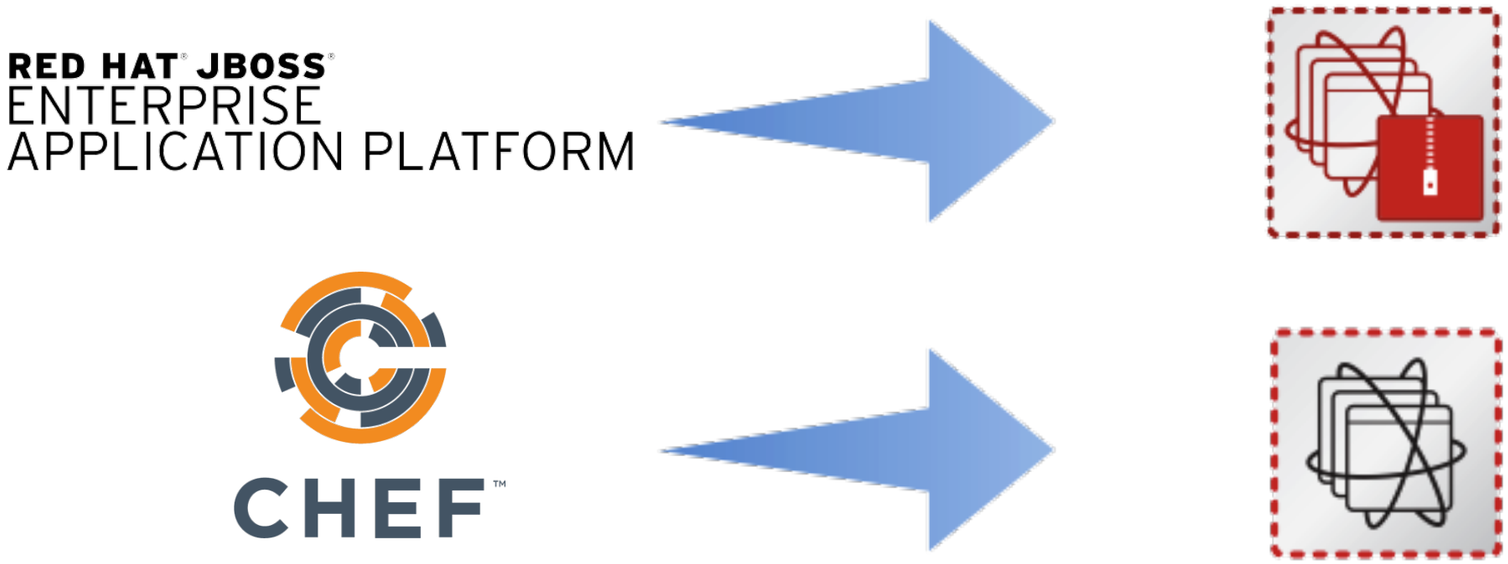
Ship

“docker push or pull <IMAGE_ID>”

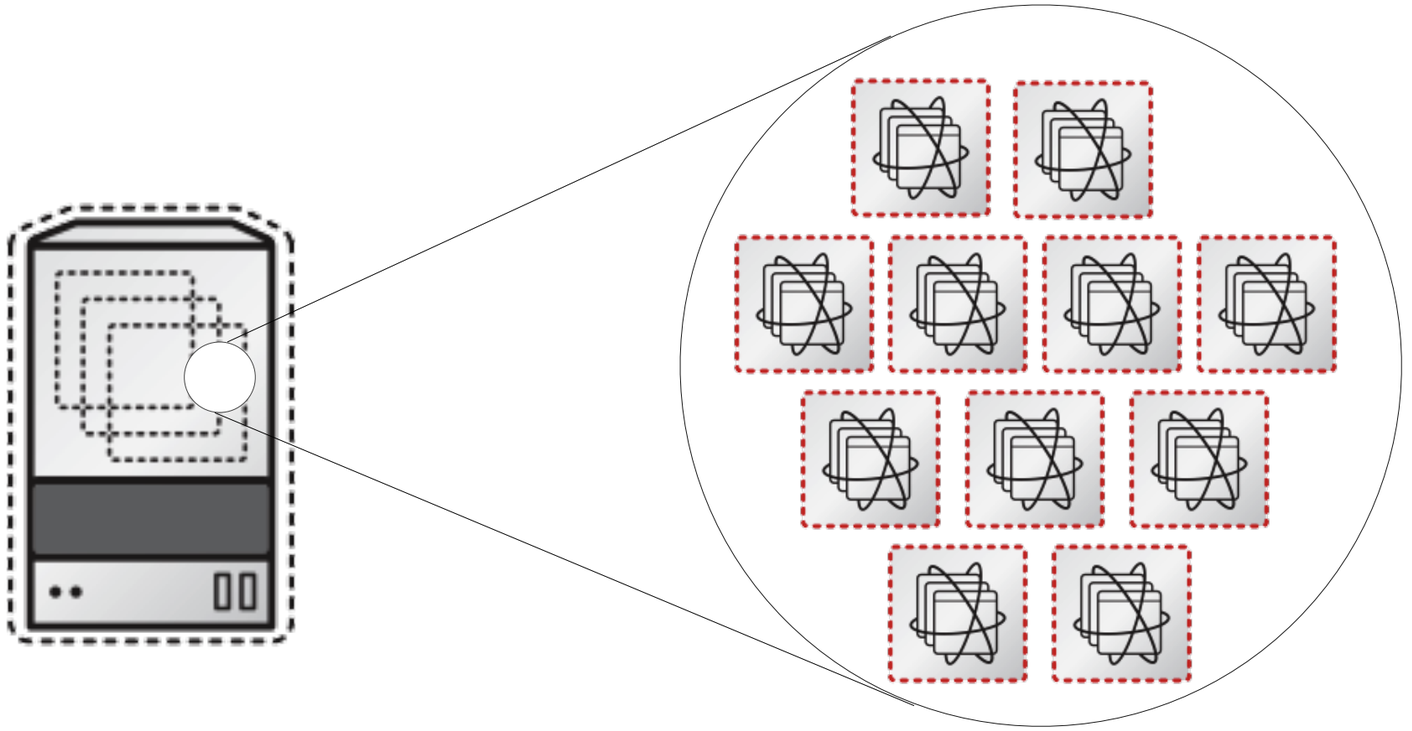
Run

“docker run <IMAGE_ID>”

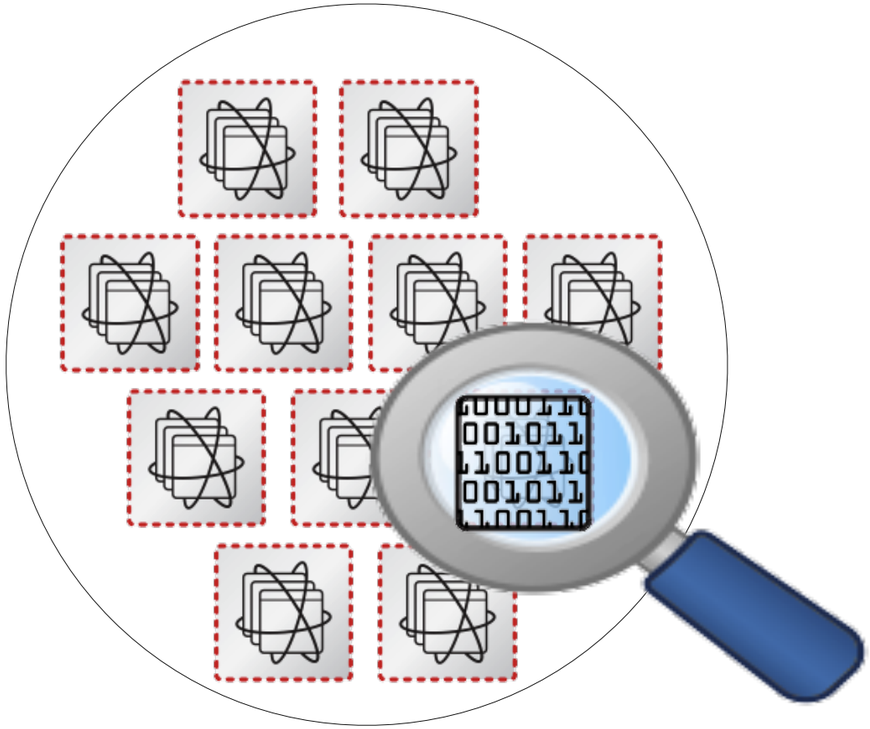
Containers provide a discrete package mechanism for application components or microservices



Several containers can live on a virtual or bare metal system



These containers are easily created and controlled by Docker tools



Finding and running containers is easy

```
-bash-4.2# docker run -P fedora/apache
Unable to find image 'fedora/apache:latest' locally
Pulling repository registry.access.redhat.com/fedora/apache
Pulling repository fedora/apache
2e11d8fd18b3: Download complete
511136ea3c5a: Download complete
ff75b0852d47: Download complete
0dae8c30a0b2: Download complete
84f33df93401: Download complete
24b116bb2956: Download complete
a7f290a6f21d: Download complete
eb86e2be11d4: Download complete
c06d2cba0d4a: Download complete
f0b140ef8cdd: Download complete
b05601b61180: Download complete
Status: Downloaded newer image for fedora/apache:latest
```

This container is exposing httpd on port 49156 as noted in 'docker ps'

```
-bash-4.2# docker ps
CONTAINER ID        IMAGE                    COMMAND             CREATED             STATUS              PORTS                     NAMES
7ca95d1b0114        fedora/apache:latest     "/run-apache.sh"    2 minutes ago       Up 2 minutes        0.0.0.0:49156->80/tcp     suspicious_colden
```

Rollbacks are also easy because image history is preserved

```
-bash-4.2# docker history fedora/apache
IMAGE               CREATED             CREATED BY                                      SIZE
b05601b61180        5 months ago        /bin/sh -c chmod -v +x /run-apache.sh           249 B
f0b140ef8cdd        5 months ago        /bin/sh -c #(nop) ADD file:d16ad02a7a4176bbff   249 B
c06d2cba0d4a        5 months ago        /bin/sh -c #(nop) EXPOSE map[80/tcp:{}]         0 B
eb86e2be11d4        5 months ago        /bin/sh -c echo "Apache" >> /var/www/html/ind   7 B
a7f290a6f21d        5 months ago        /bin/sh -c yum -y install httpd && yum clean    15.87 MB
24b116bb2956        5 months ago        /bin/sh -c yum -y update && yum clean all       164.1 MB
84f33df93401        5 months ago        /bin/sh -c #(nop) MAINTAINER "Scott Collier"    0 B
ff75b0852d47        5 months ago        /bin/sh -c #(nop) MAINTAINER Lokesh Mandvekar   0 B
511136ea3c5a        21 months ago                                                       0 B
```

redhat.

RPM still exists in the container so its features and tools can still be used, even in a containerized environment

```
[root@7ca95d1b0114 /]# rpm -qa | wc -l
160

[root@7ca95d1b0114 /]# rpm -qi httpd
Name        : httpd
Version     : 2.4.10
Release     : 1.fc20
Architecture: x86_64
Install Date: Mon 29 Sep 2014 12:28:44 PM UTC
Group       : System Environment/Daemons
License     : ASL 2.0
Signature   : RSA/SHA256, Wed 23 Jul 2014 01:23:23 PM UTC, Key ID 2eb161fa246110c1
Source RPM  : httpd-2.4.10-1.fc20.src.rpm
Build Date  : Wed 23 Jul 2014 10:32:07 AM UTC
Build Host  : buildvm-22.phx2.fedoraproject.org
Vendor      : Fedora Project
URL         : http://httpd.apache.org/
Summary     : Apache HTTP Server
Description :
The Apache HTTP Server is a powerful, efficient, and extensible
web server.

[root@7ca95d1b0114 /]# rpm -qV httpd
missing     /run/httpd/htcacheclean
```

Containers cannot look outside of their namespace but admins can easily peak in, giving operational flexibility

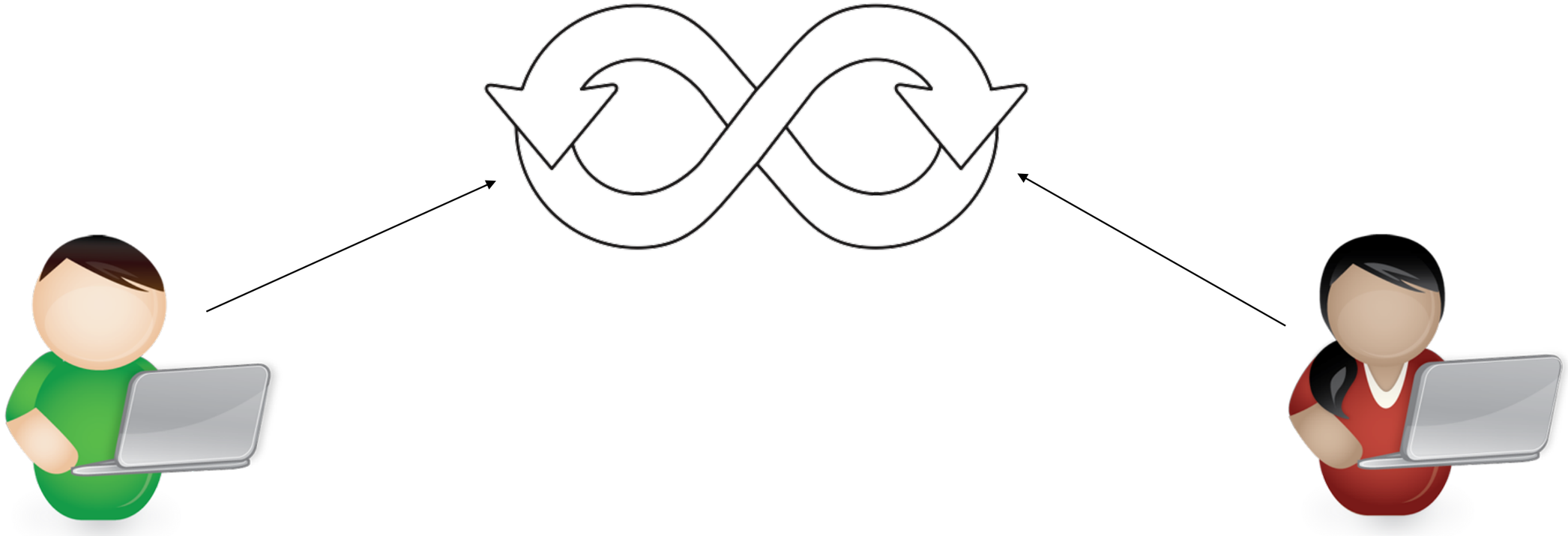
```
-bash-4.2# docker exec -it suspicious_colden /bin/bash

[root@7ca95d1b0114 /]# ps -ax
   PID TTY      STAT   TIME COMMAND
     1 ?        Ss     0:00 /bin/sh /usr/sbin/apachectl -D FOREGROUND
     9 ?        S      0:00 /usr/sbin/httpd -D FOREGROUND
    10 ?        S      0:00 /usr/sbin/httpd -D FOREGROUND
    11 ?        S      0:00 /usr/sbin/httpd -D FOREGROUND
    12 ?        S      0:00 /usr/sbin/httpd -D FOREGROUND
    13 ?        S      0:00 /usr/sbin/httpd -D FOREGROUND
    14 ?        S      0:00 /usr/sbin/httpd -D FOREGROUND
    33 ?        S      0:00 /bin/bash
    50 ?        R+     0:00 ps -ax
```

Builds are reproducible via Dockerfiles

~/my-app/Dockerfile

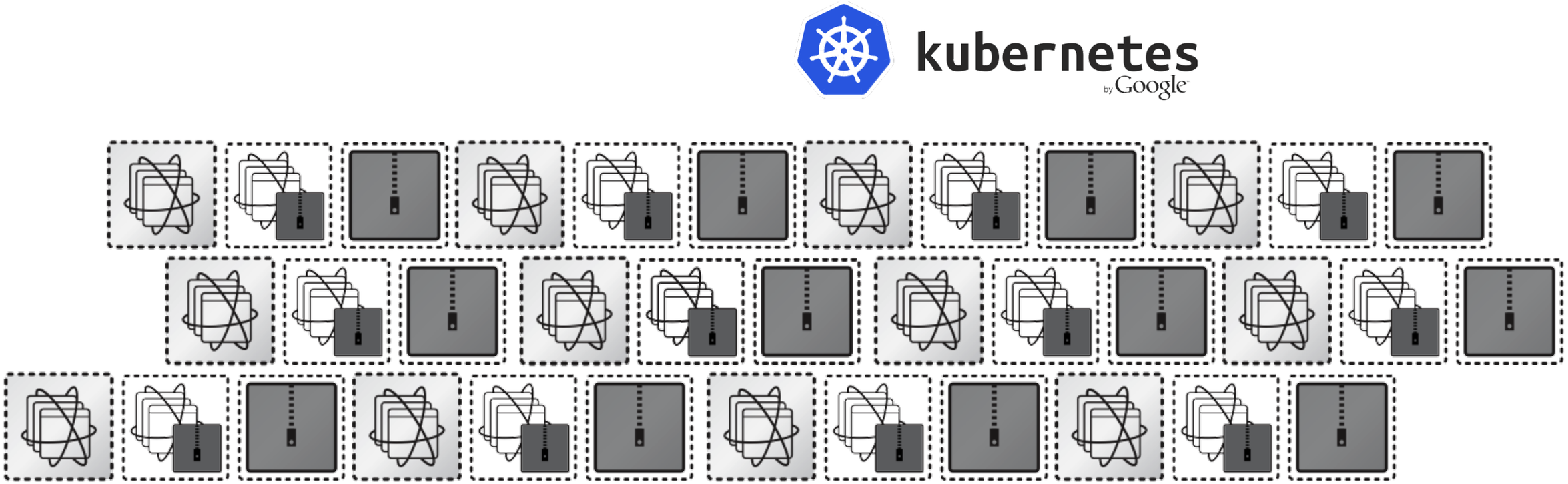
```
FROM fedora
MAINTAINER Mrs Developer <mydev@example.com>

ENV GOPATH /go/src
RUN yum install -y golang git hg && yum clean all
RUN mkdir -p $GOPATH && echo $GOPATH >> ~/.bash_profile

ADD . /my-stuff
```

Docker image delivery and consumption can happen via your operations work flow needs, devops or not



ORCHESTRATION

More serious workloads require orchestration like Kubernetes to offload management overhead

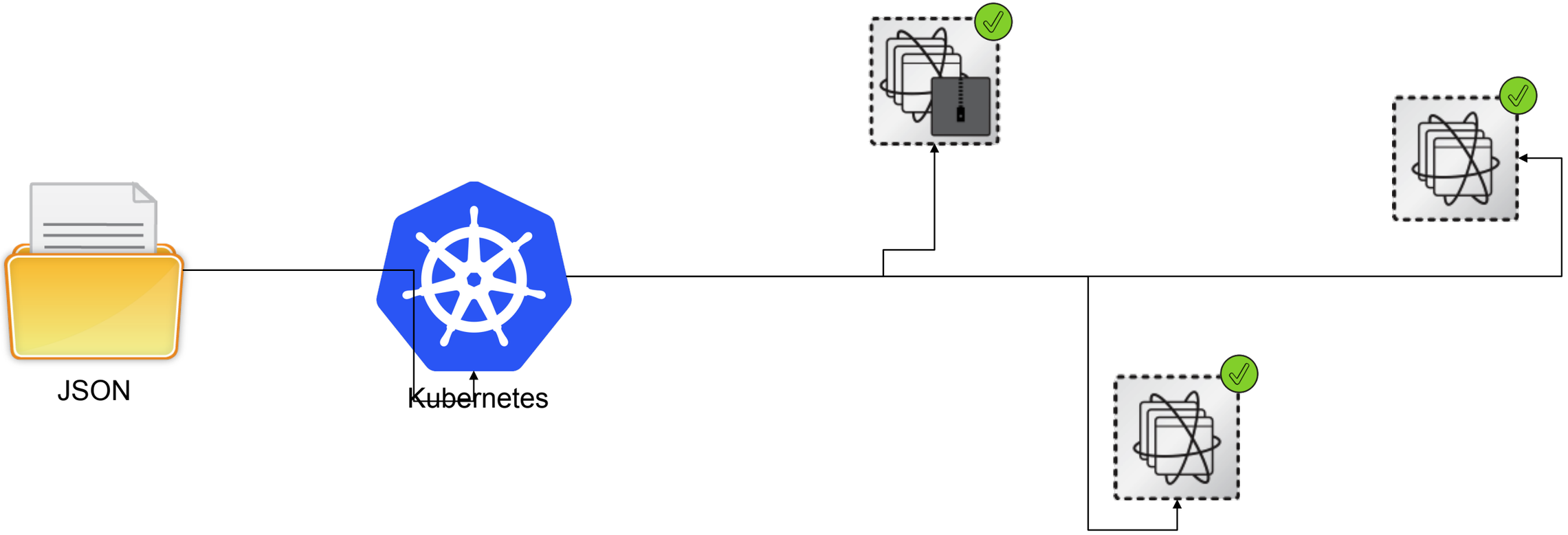


Kubernetes allow operations teams to describe an application via json

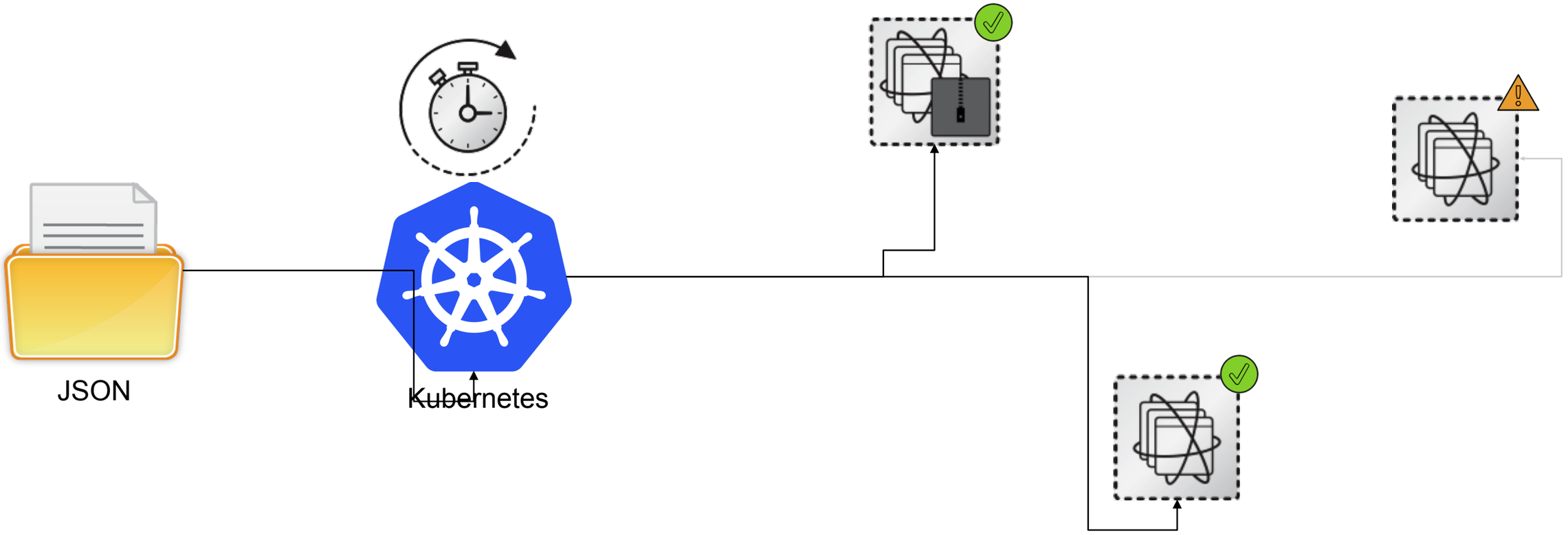
~/MyApp/kubernetes/prod_description.json

```
{
  "id": "frontend-controller",
  "kind": "ReplicationController",
  "apiVersion": "v1beta1",
  "desiredState": {
    "replicas": 3,
    "replicaSelector": {"name": "frontend"},
    "podTemplate": {
      "desiredState": {
         "manifest": {
           "version": "v1beta1",
           "id": "frontend",
           "containers": [{
             "name": "php-redis",
             "image": "kubernetes/example-guestbook-php-redis",
             "cpu": 100,
             "memory": 50000000,
             "ports": [{"name": "http-server", "containerPort": 80}]
...
```

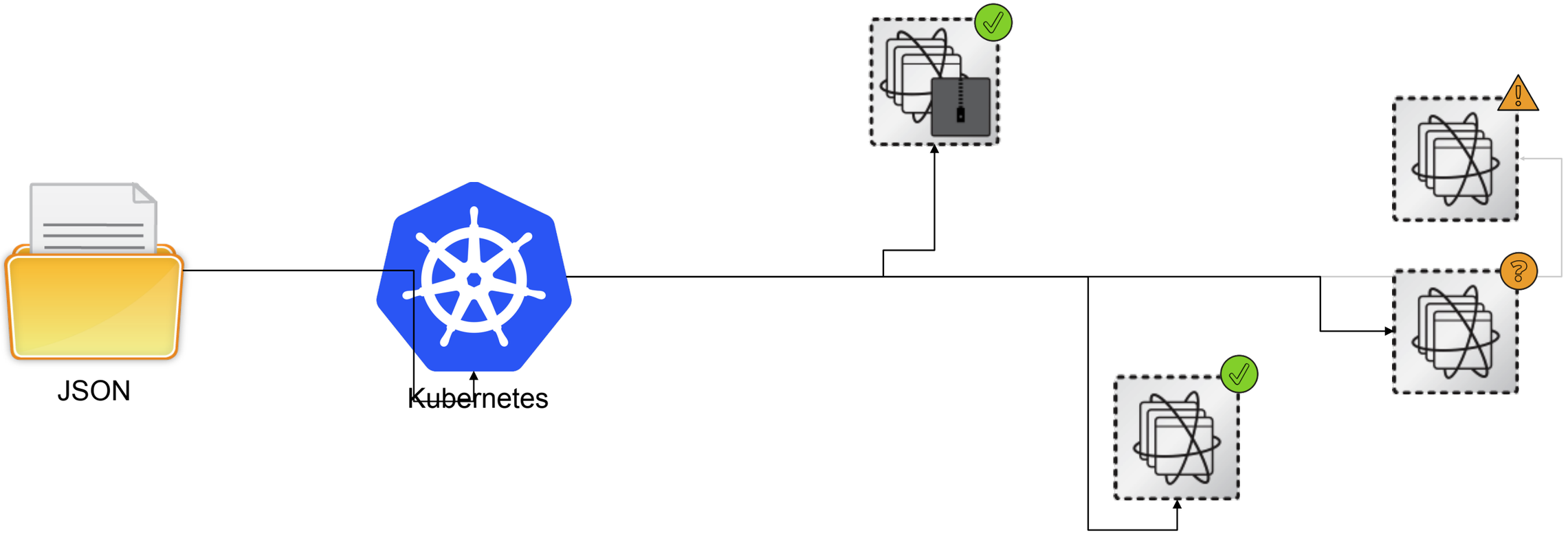
Once created, Kubernetes will keep the environment online as described in the json file



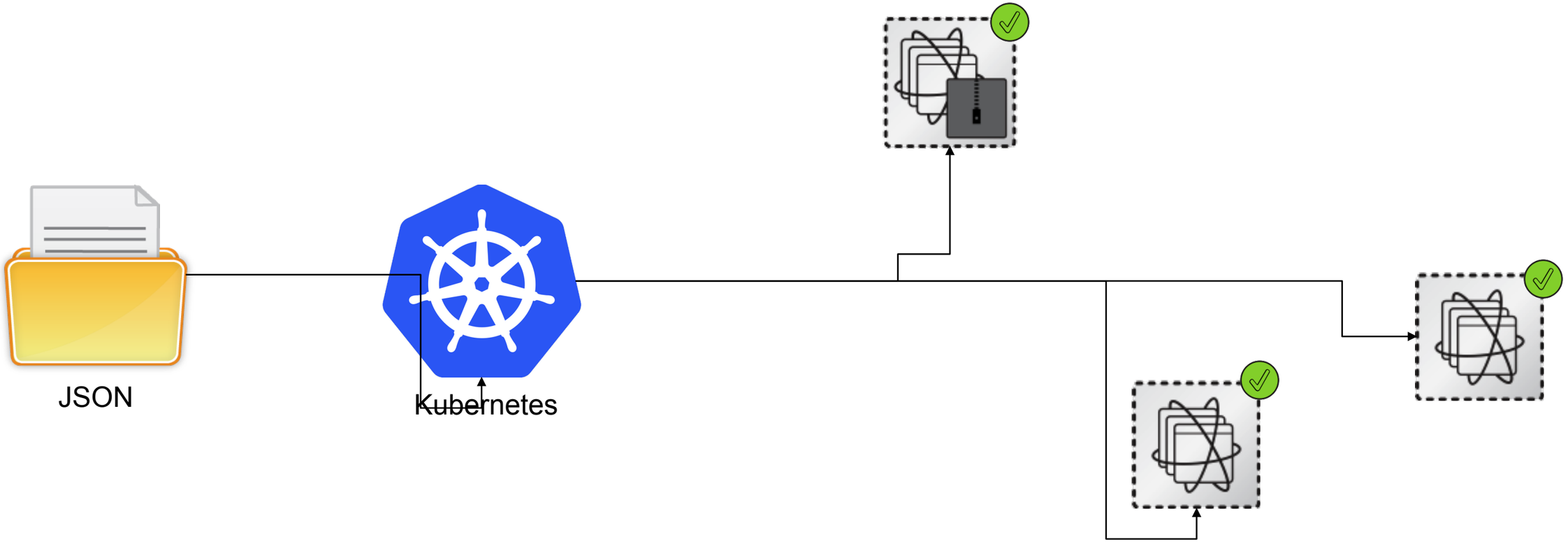
Once created, Kubernetes will keep the environment online as described in the json file



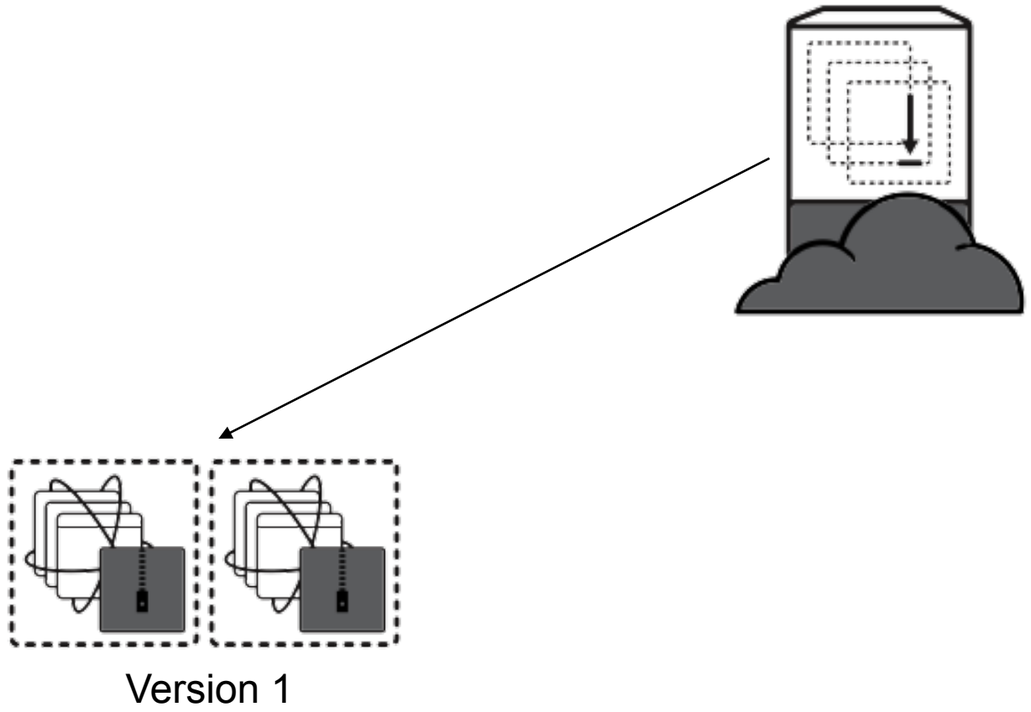
Once created, Kubernetes will keep the environment online as described in the json file



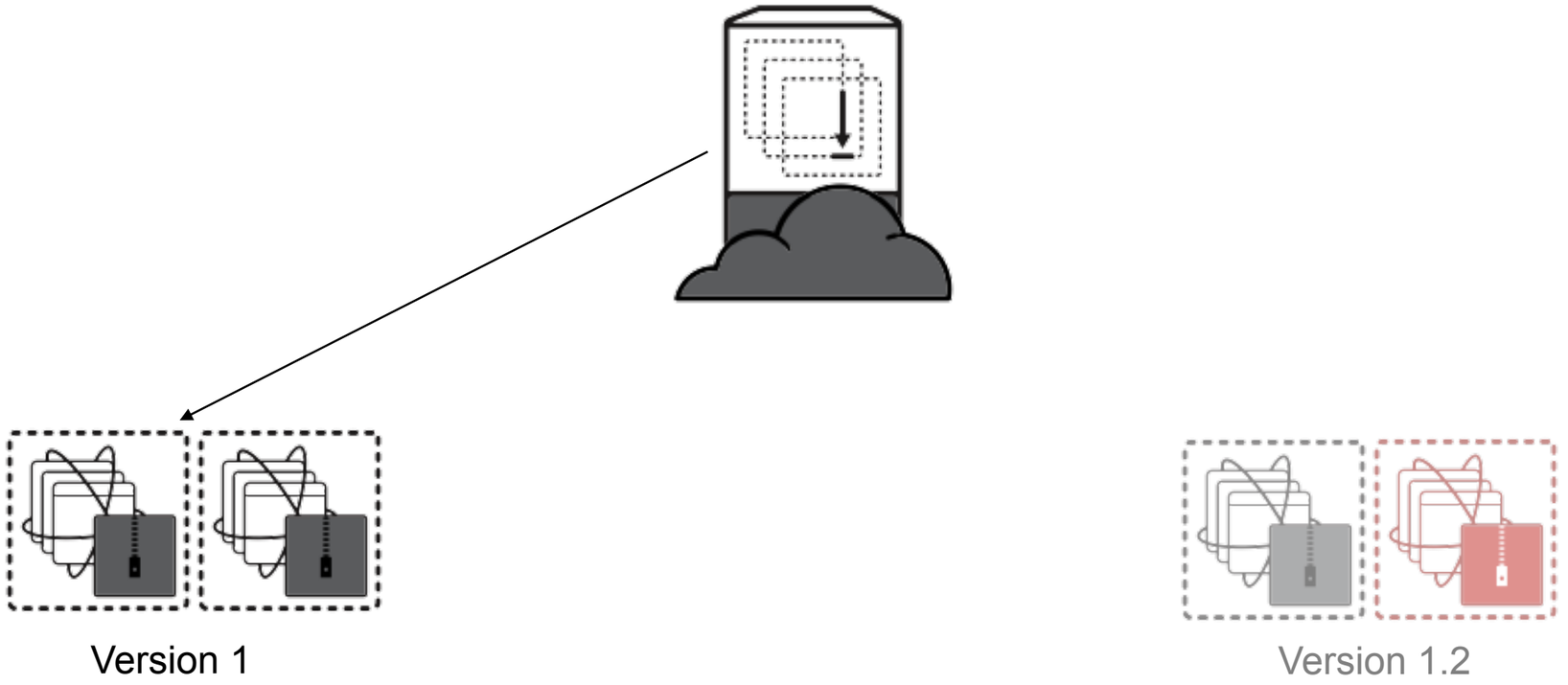
Once created, Kubernetes will keep the environment online as described in the json file



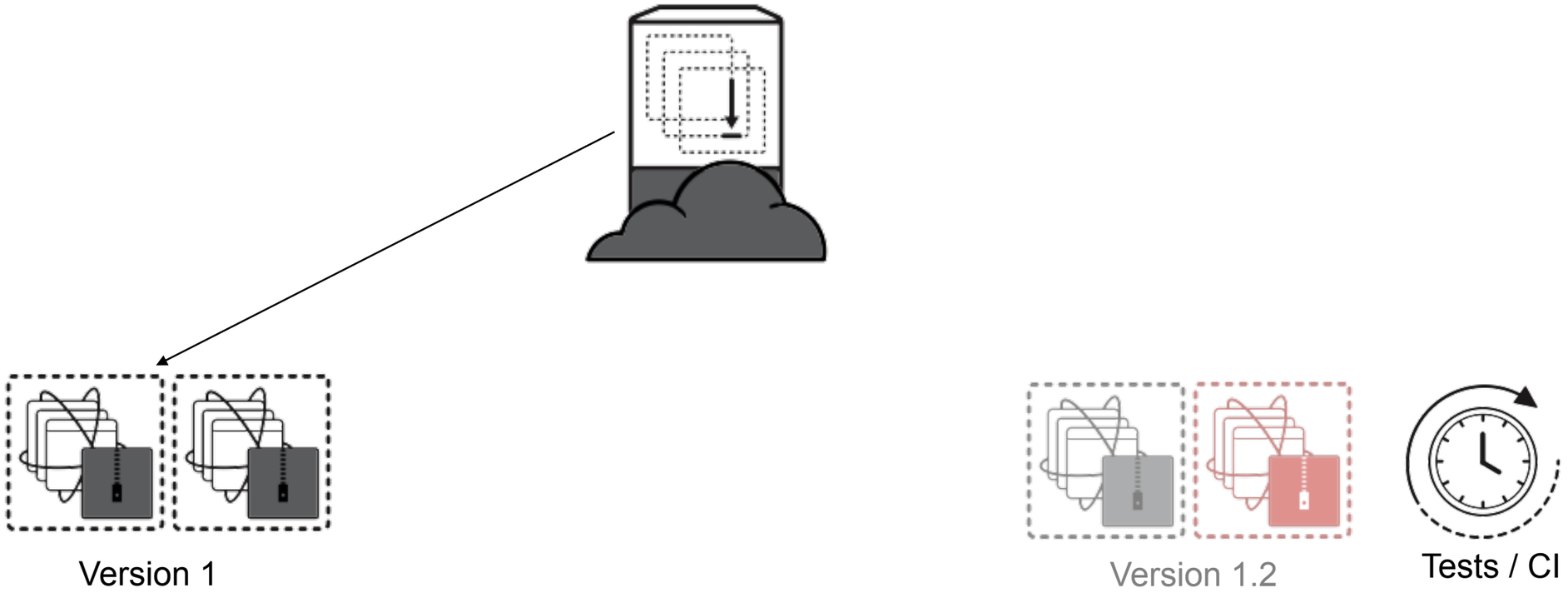
New applications can be spun up and tested before old applications are removed, lowering risk for upgrades



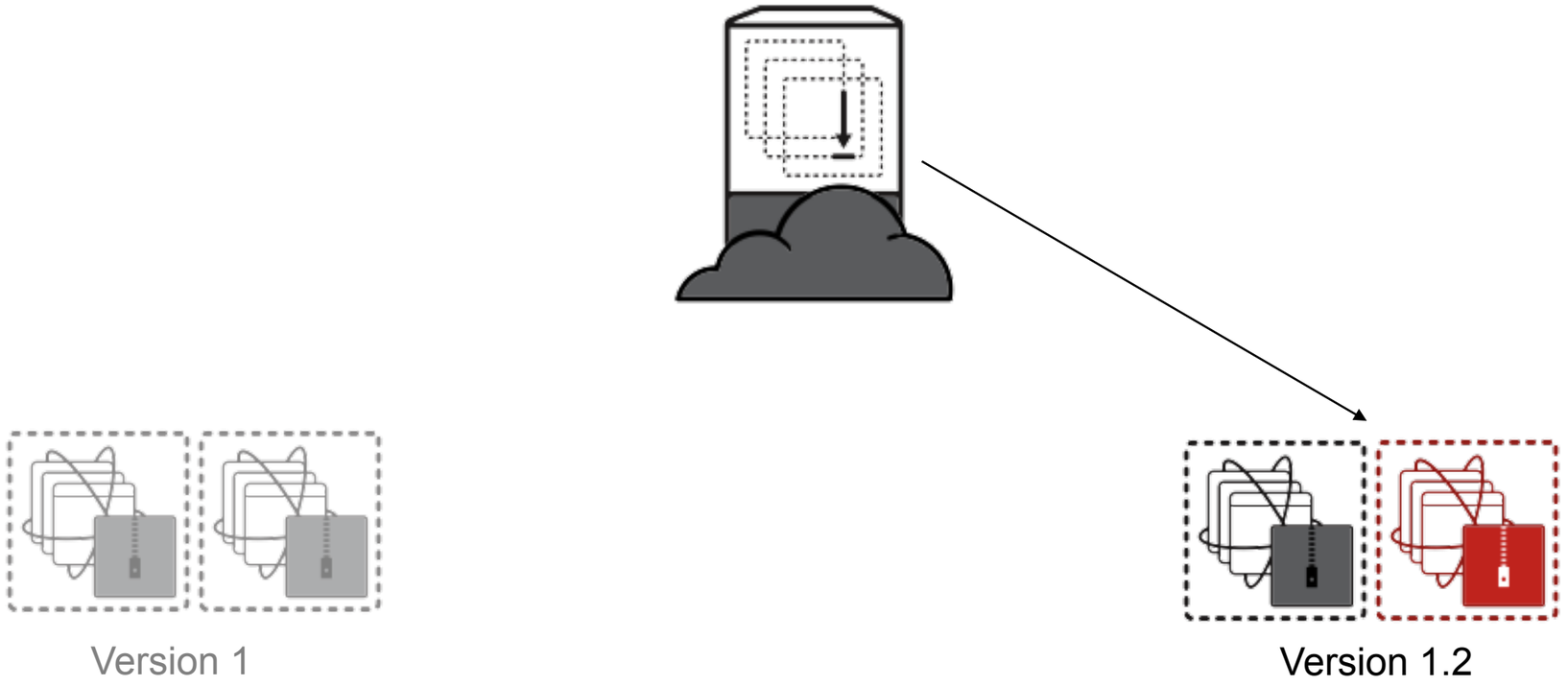
Some call this method of deployment red-black deployment, admins won't get stuck in the middle of an upgrade



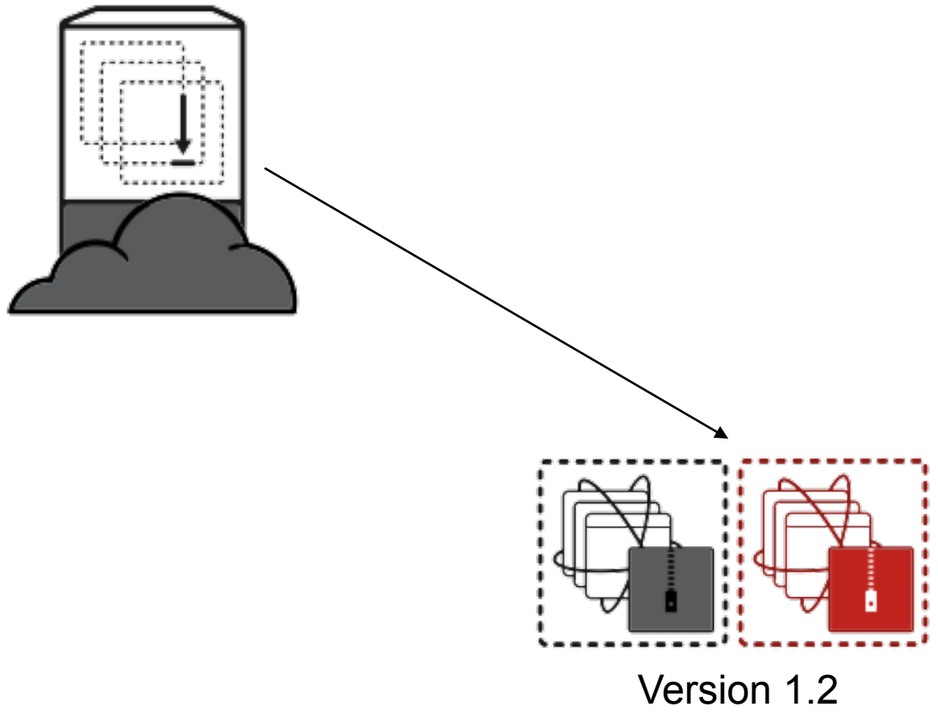
Tests and certification can be done before customers access it



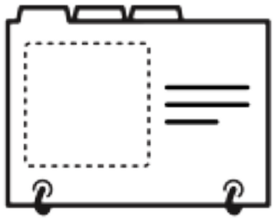
Once ready, the new version is used and the old version can be removed



Rollbacks can be done using the same method if desired



Docker images can be pushed to and pulled from a registry, though not all registries are created equal



docker.io
Registry

Red Hat
Certified

Private
Registry

THANK YOU

BIMODAL IT

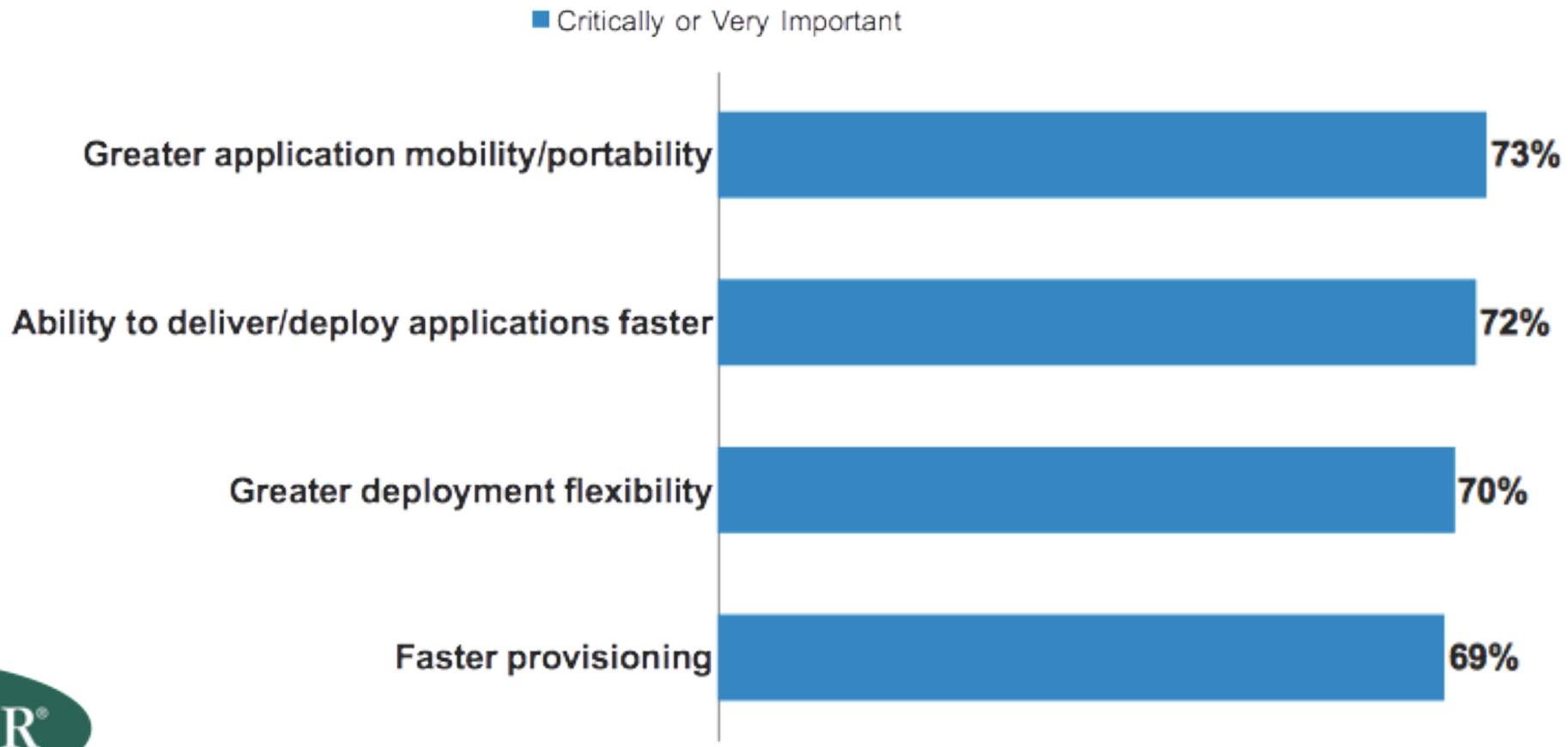
To capture digital opportunities, CIOs need to deal with speed, innovation, and uncertainty. This requires operating two modes of enterprise IT: conventional and "nonlinear."

Gartner

Executive Summary:
Taming the Digital Dragon: The 2014 CIO Agenda

CONTAINERS DELIVER MANY BENEFITS

How important are the following benefits of containers to your organization?



FORRESTER®

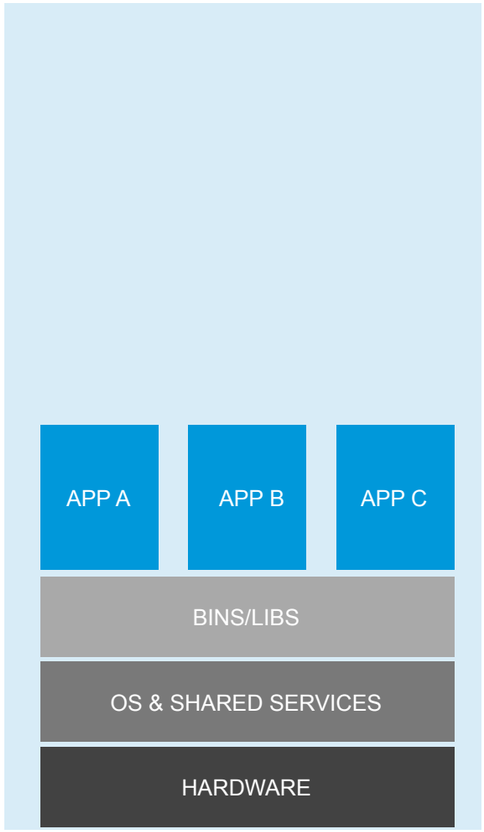
Base: 194 IT and Developer/programmer decision-makers at companies with 500+ employees in APAC, EMEA, and NA
Source: A commissioned study conducted by Forrester Consulting on behalf of Red Hat, January, 2015

redhat.

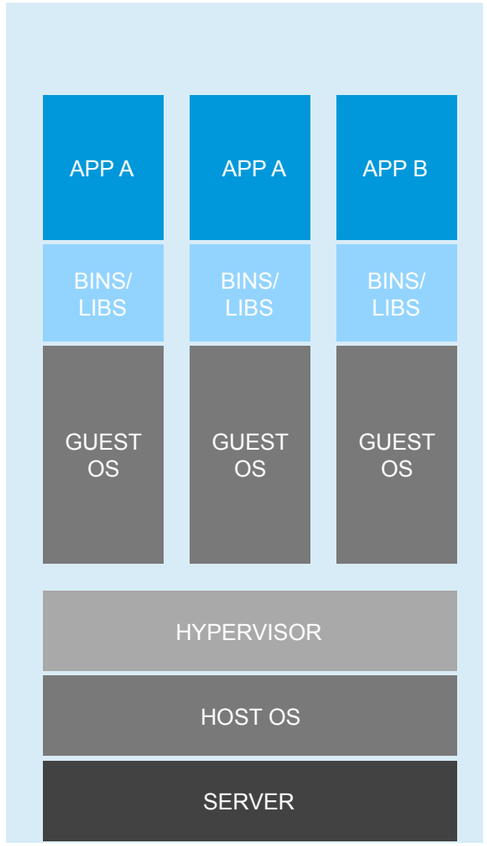
TOP 5 MISCONCEPTIONS OF CONTAINERS

1. Containers are new.
2. Containers equal virtualization.
3. Containers are universally portable.
4. Containers are secure by default.
5. Containers are not enterprise-ready.

EVOLUTION TO CONTAINERS



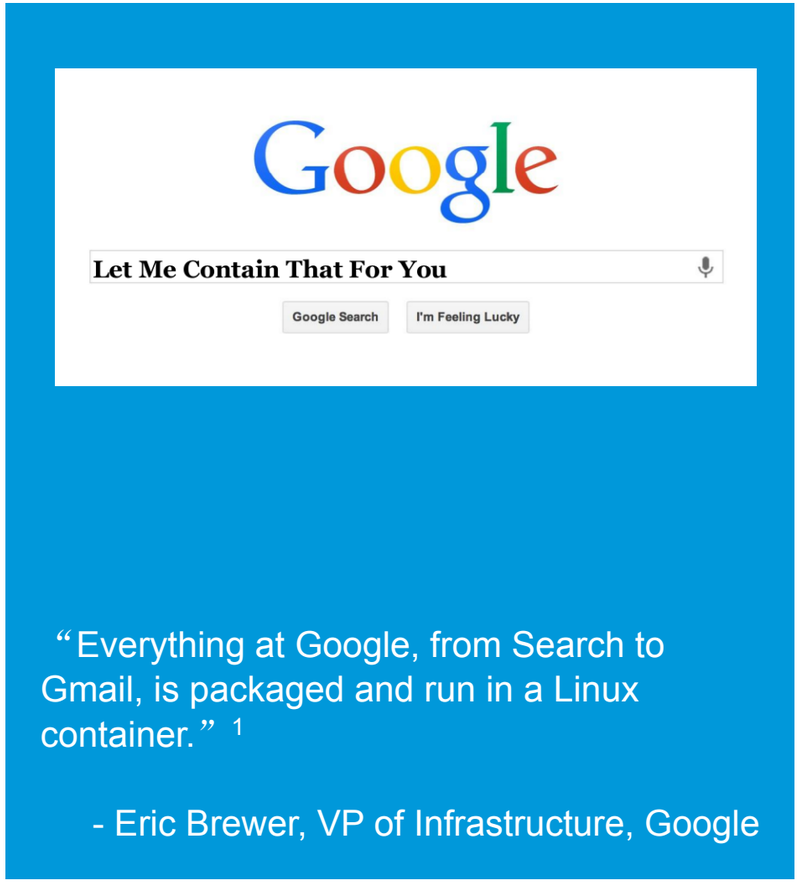
PHYSICAL



VM

CONTAINER

SOME OF THE MOST ADVANCED INFRASTRUCTURES RUN ON CONTAINERS



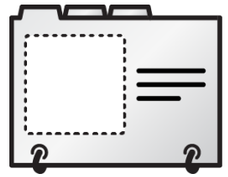
[1] Source: http://googlecloudplatform.blogspot.com/2014/06/an-update-on-container-support-on-google-cloud-platform.html

ESTABLISHING STANDARDS AROUND…

ISOLATION WITH LINUX CONTAINERS

CONTAINER FORMAT WITH DOCKER

ORCHESTRATION WITH KUBERNETES



REGISTRY / CONTAINER DISCOVERY