Tanzu Cloud Native webinars 1/6

Modernize your applications

Robert Jensen

Lead Systems Engineer @Vmware

@rhjensen / jensenr@vmware.com

Agenda

- What does it mean to modernize your applications.
- Application Design
- How does Containers fit in ?
- How does Kubernetes fit in ?

The purpose of this Webinar

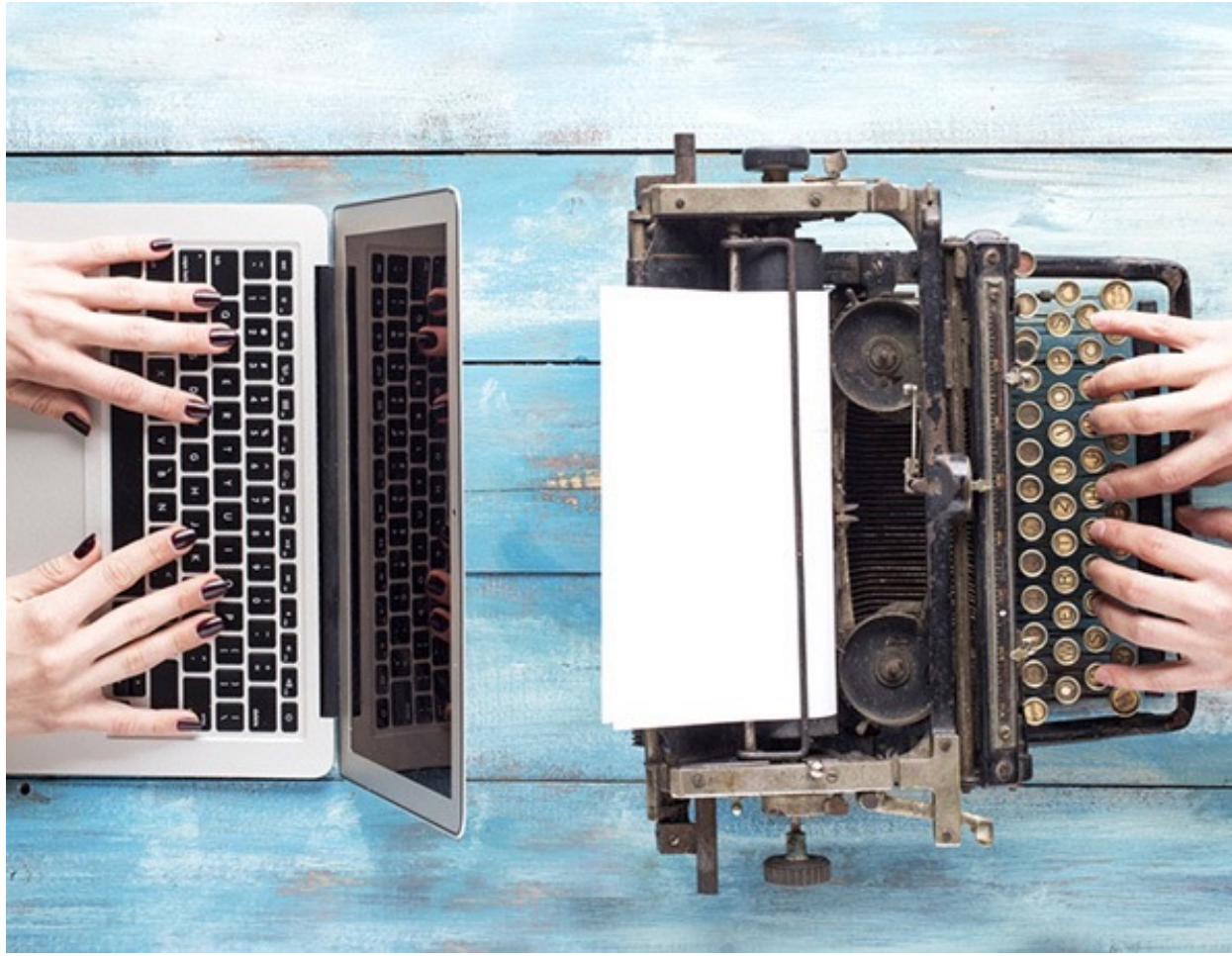
Is to give an introduction to Cloud Native concepts and technologies.

We start with the basics, and try to take it to the next level.

Questions : Please use the Q/A function. We will look at them In the end.

What does it mean to modernize your applications

Application modernization is the practice of updating older software for newer computing approaches, including newer languages, frameworks and infrastructure platforms



Why modernize legacy applications?

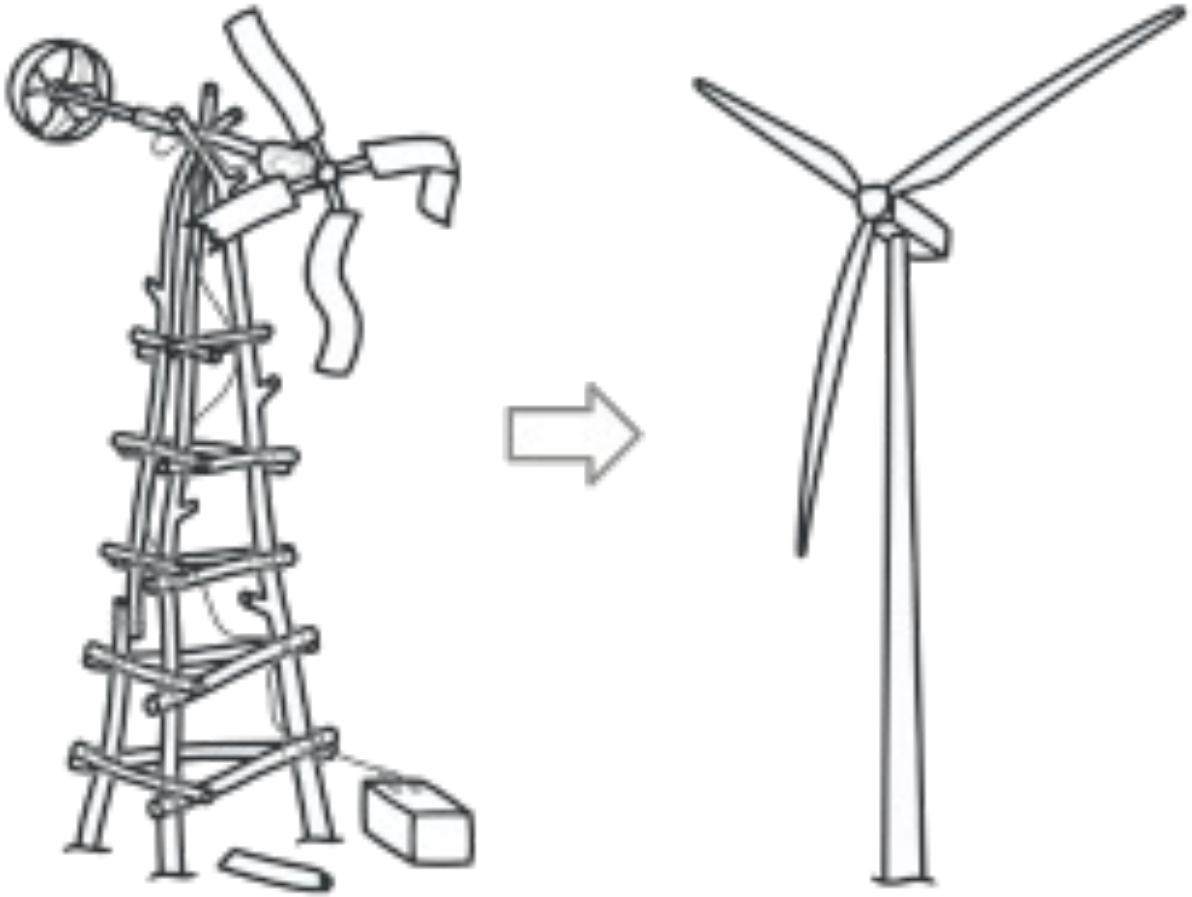
Application modernization enables an organization to protect its investments and refresh its software portfolio to take advantage of contemporary infrastructure, tools, languages and other technology progress.

A robust application modernization strategy can **reduce** the **resources** required to run an application, **increase** the **frequency** and **reliability** of deployments, and **improve uptime** and **resiliency**, among other benefits

Application modernization patterns

Lift and shift: Sometimes called rehosting, the phrase “lift and shift” has become software development lingo for taking an existing application and moving it from a legacy environment (such as an on-premises server) to newer infrastructure, such as a public cloud platform

Application modernization patterns

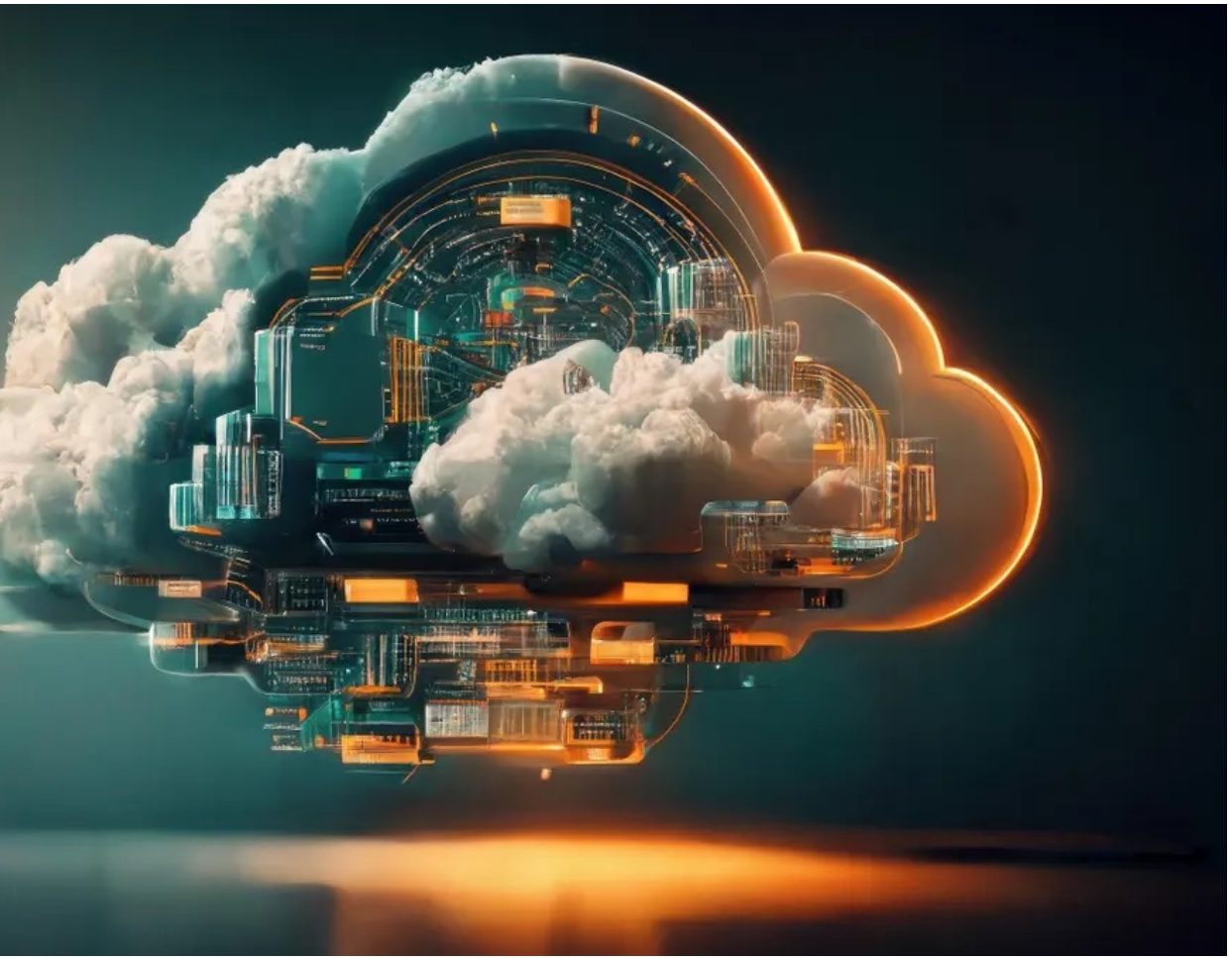


Refactoring: is essentially another way of saying “rewriting” or “restructuring.” This approach to application modernization entails taking a legacy application and retooling significant chunks of its underlying code to better run in a new environment, usually cloud infrastructure

Application modernization patterns

Replatforming: This pattern can be viewed as a middle ground or compromise between the lift-and-shift and refactoring approaches. It does not require major changes in code or architecture, as with refactoring, but entails complementary updates that enable the legacy app to take advantage of a modern cloud platform, such as modifying or replacing the application's backend database.

Key technologies



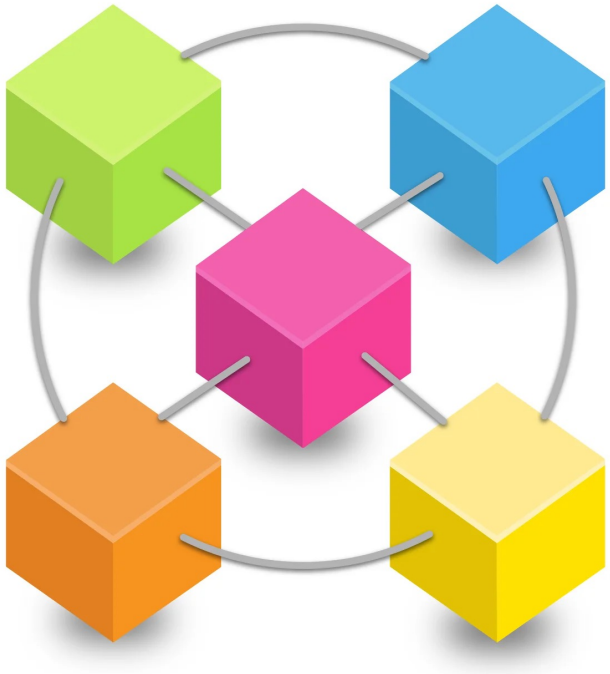
Cloud Computing: When people discuss application modernization, they are typically referring to the process of migrating traditional applications to run in modern cloud environments. These include public cloud platforms, private clouds and hybrid clouds (which usually refer to public and/or private clouds integrated with on-premises environments.)

Key technologies

Containers: Containers are a cloud-centric method for packaging, deploying and operating applications and workloads. The big-picture benefits associated with containerization include greater scalability, portability and operational efficiency that is well-suited for cloud infrastructure, and especially multi-cloud and hybrid cloud environments.

Key technologies

Microservices

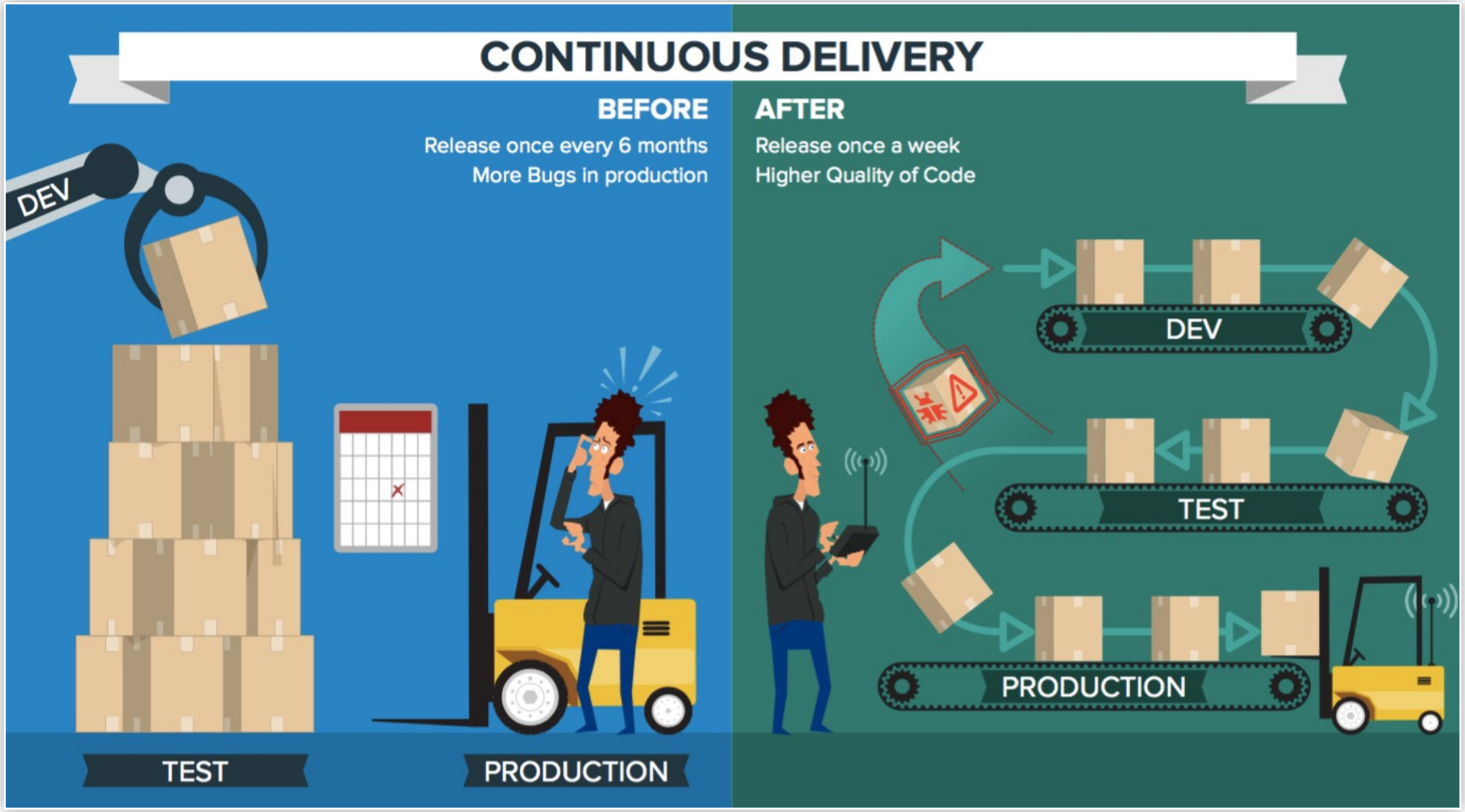


Microservices: This is not so much a technology as an architectural choice. Instead of building and operating an application as a single, complete codebase—usually called a monolith, or monolithic development—you decouple different components into smaller, discrete pieces that can be deployed, updated and operated independently.

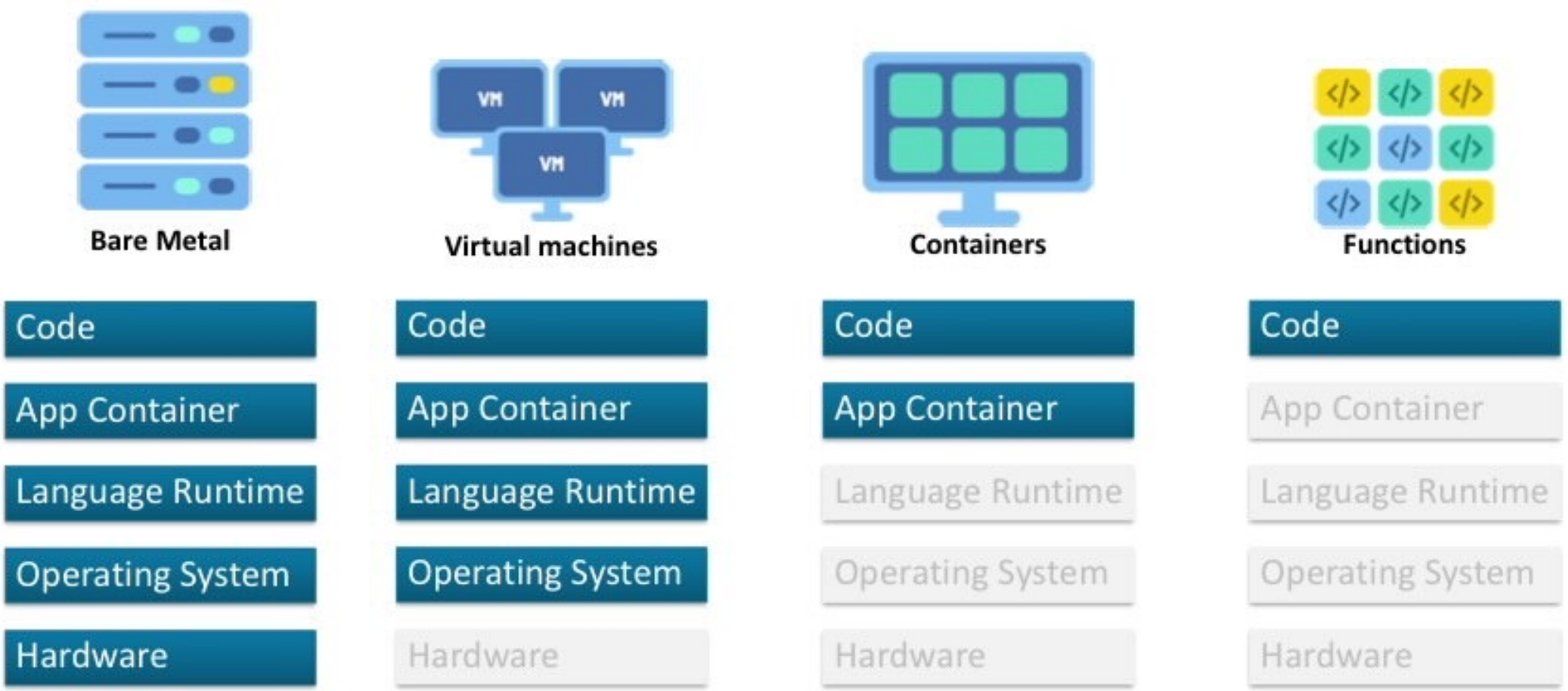
Key technologies

Orchestration and Automation: Orchestration in software development refers to the automation of many of the operational tasks associated with containers, including deployment, scaling and networking. Automation in general is an important principle and technology, as it is increasingly necessary to ensure that development, operations and security teams can sustainably manage their modern apps at scale.

Application Design

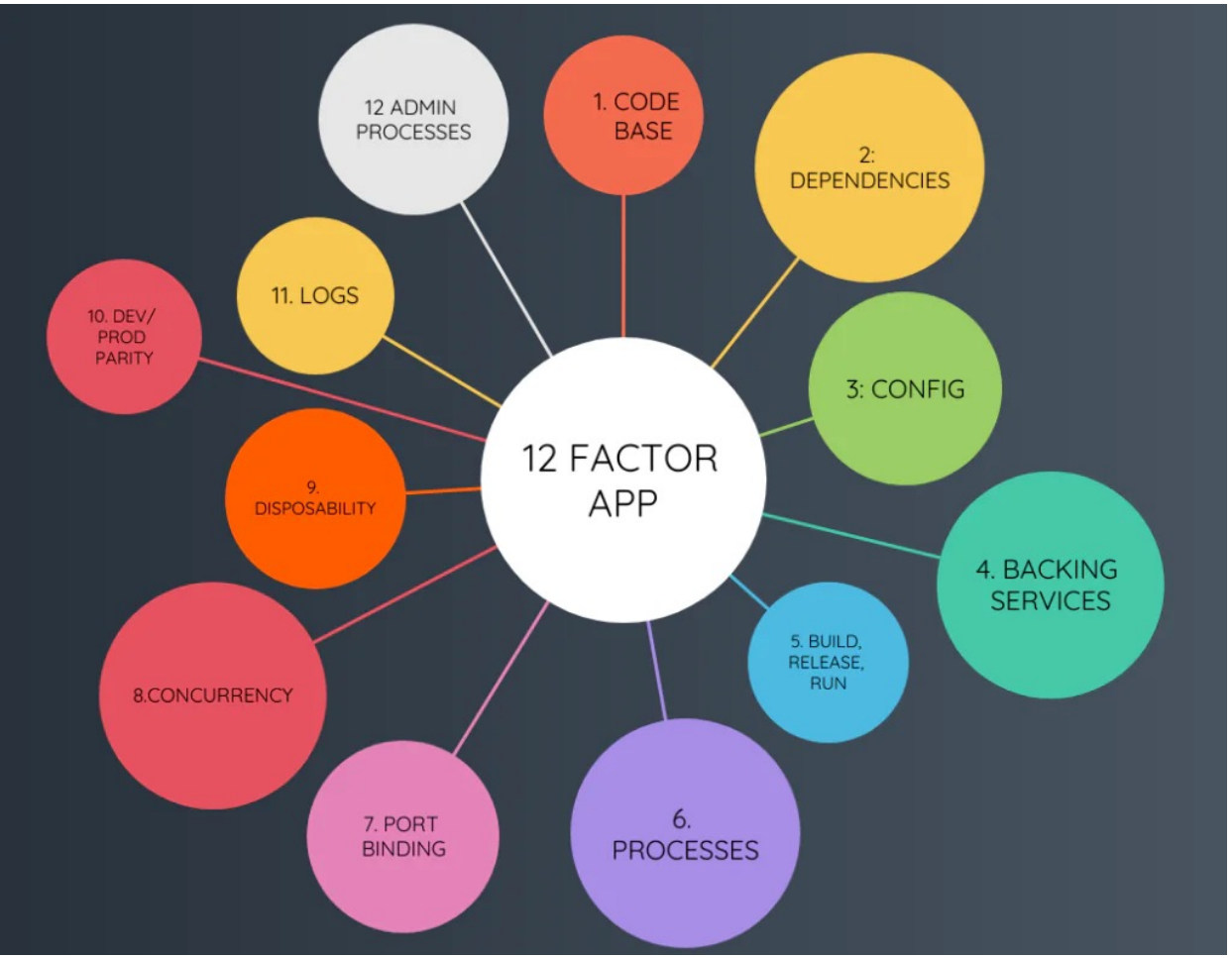


Chose your architecture



Source : https://blogs.oracle.com/developers/post/functions-as-a-service-evolution-use-cases-and-getting-started

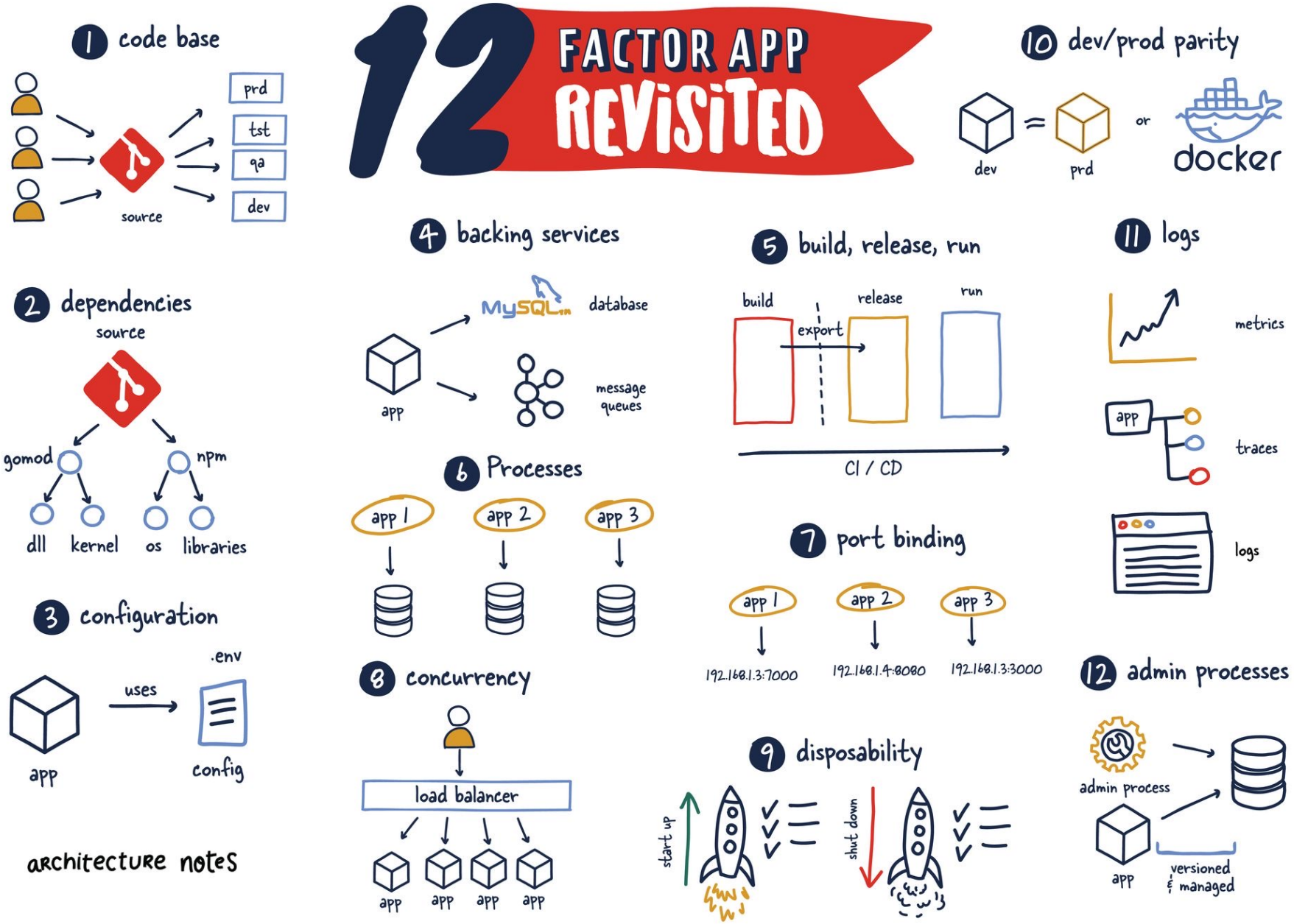
Application Design



12 factor app : is a methodology for building distributed applications, published in 2012 by Heroku.

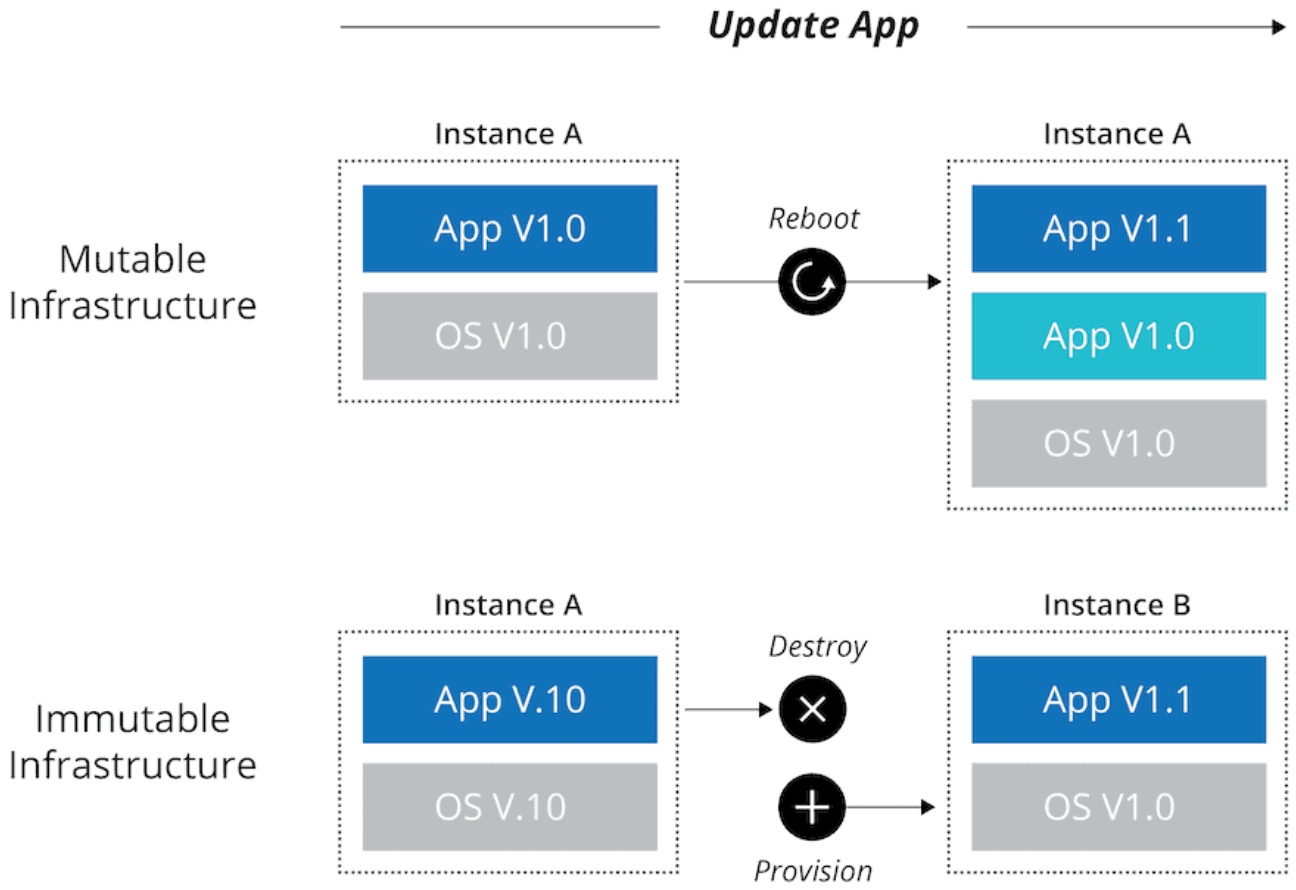
The goal of these 12-factors is to teach developers how to build cloud-ready applications using declarative formats for automating setup, had a clean contract with underlying operating system and were prepared for dynamic scaling

https://12factor.net



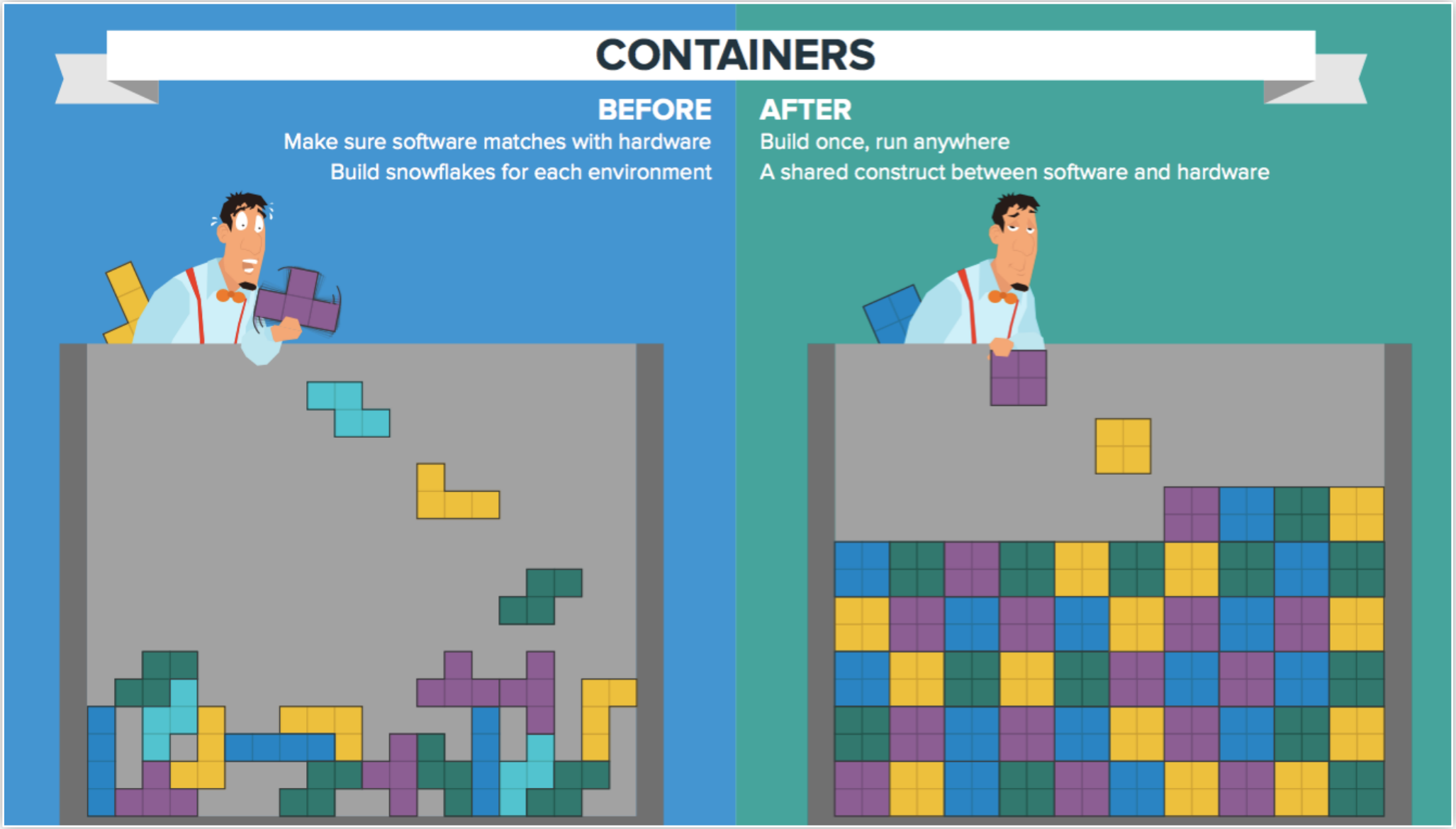
Source : https://architecturenotes.co/12-factor-app-revisited/

Mutable vs Immutable Infrastructure

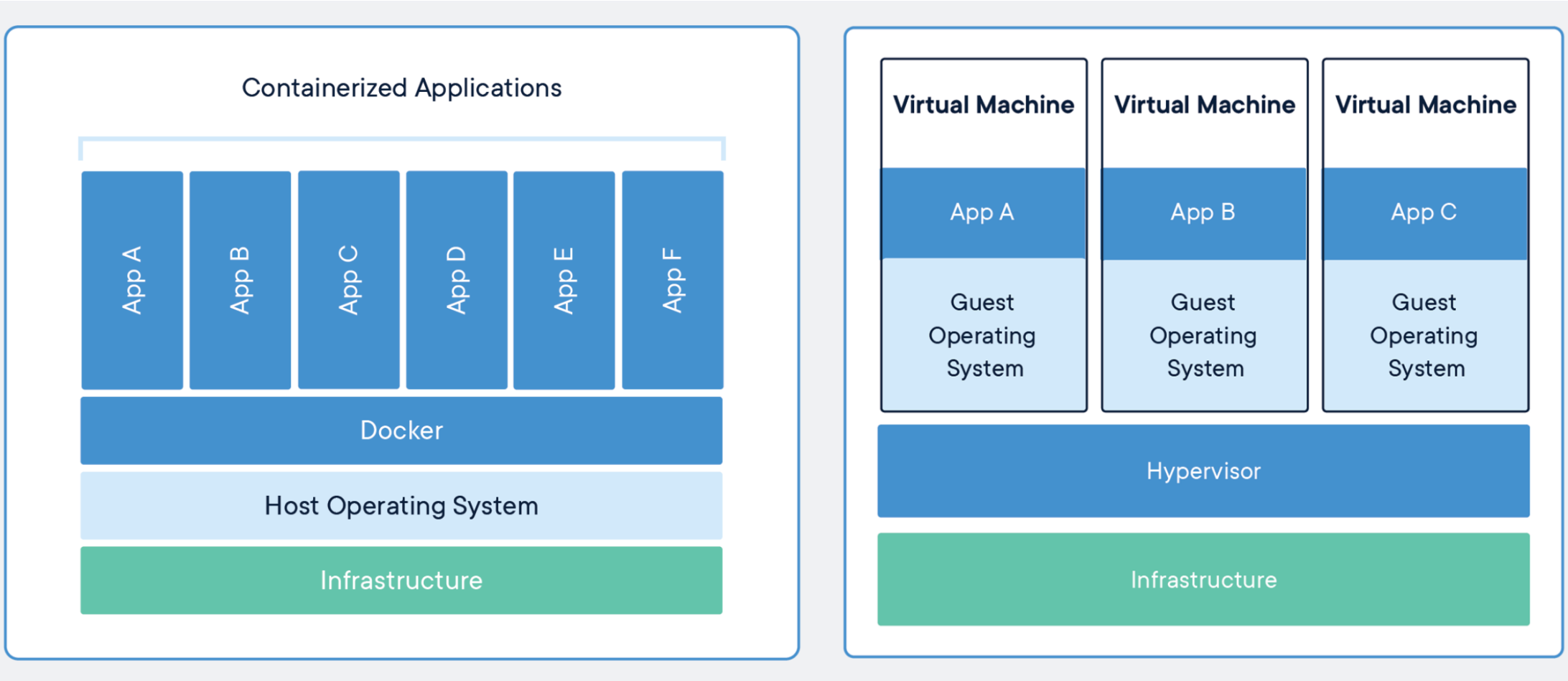


Source : https://k21academy.com/terraform-iac/why-terraform-not-chef-ansible-puppet-cloudformation/

How does Containers fit in ?



Docker vs VM's



VM vs Containers – Initial setup

VM Operations

1. Install or deploy OS
2. Configure OS
3. Patch OS
4. Install tools (if not included in base image)
5. Install dependencies
6. Deploy App
7. Configure App

Container Operations

1. Create Dockerfile
2. Build container from Dockerfile
3. Deploy container

Red = Automated task

VM vs Containers – Update

VM Operations

1. Patch OS
2. Update dependencies
3. Update App

Container Operations

1. Push new code
2. Build new container
3. Deploy new container

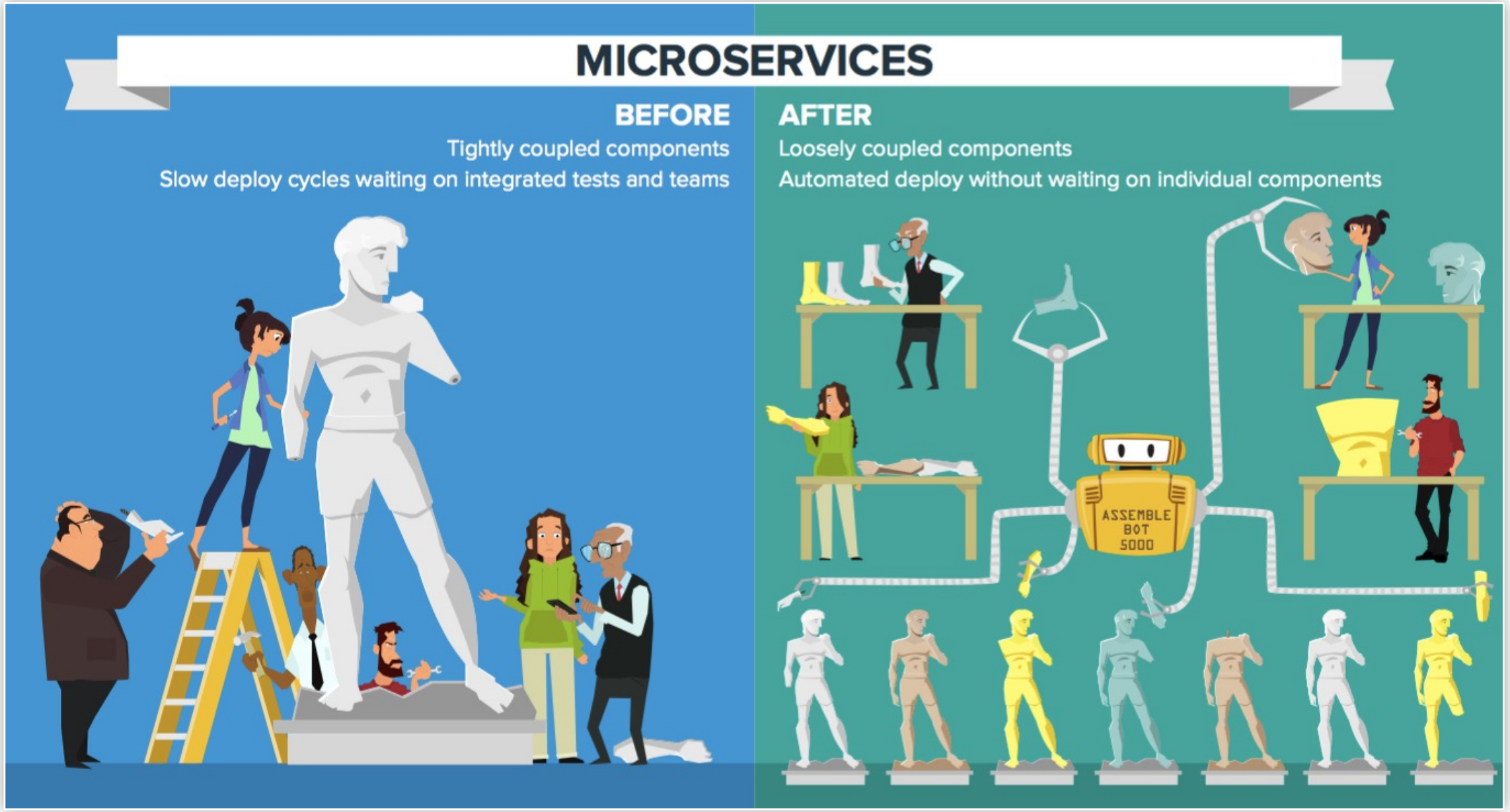
Red = Automated task

What is a Dockerfile

```
FROM python:3.12.0b4-slim-bullseye
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt --no-cache-dir
CMD ["python3", "app.py"]
```

Demo

How does Kubernetes fit in ?



What is Kubernetes (K8s)

Kubernetes, is an open-source platform for managing, automating deployment, scaling, and operating containerized applications across a cluster of worker nodes.

Capabilities:

- Deploy your applications quickly and predictably
- Scale your applications on the fly
- Seamlessly roll out new features
- Optimize use of your hardware by using only the resources you need

Role:

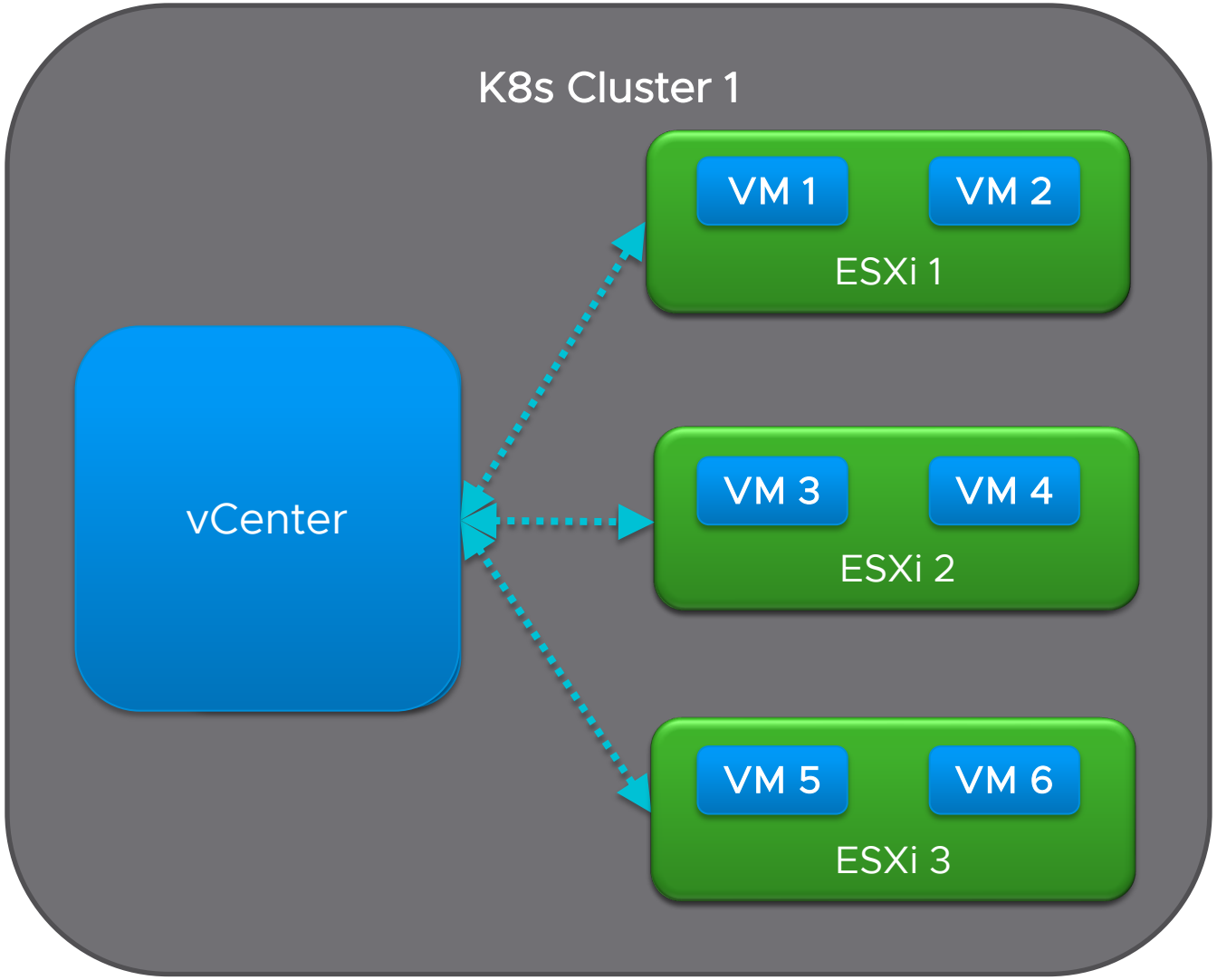
- K8s sits in the Container as a Service (CaaS) or Container orchestration layer

kubernetes

K8s introduces a lot new concepts

10,000 ft. View

- Cluster
- vCenter
- ESXi (Hosts/nodes)
- VMs



Important notes about Kubernetes

- Kubernetes can be hard, but It does not have to be https://github.com/kelseyhightower/kubernetes-the-hard-way
- Easy as in building your own cloud
- Do not treat k8s as vm's (Cat's vs Cattle)
- Everything is done using the Kubernetes API.
- Kubernetes is not "just" about Containers (CRD's)

Demo

Next times Agenda

- How to get started with Gitops, DevOps or DevSecOps
- What new skills and tools are needed ?
- What about Databases ?
- How to monitor your applications ?

Until next time & Q&A

Look at the following

Robert Jensen
Lead Systems Engineer @Vmware

@rhjensen / jensenr@vmware.com

12 Factor app video : https://youtu.be/1OhmRmMsGdQ

12 factor app website : https://12factor.net

Docker : https://www.docker.com

Docker Hub : https://hub.docker.com

Kubernetes : https://kubernetes.io

CNCF : https://www.cncf.io

Tanzu : https://tanzu.vmware.com

Register for next event on
https://webinars.tanzu.dk

Recording / Slides will also be available there.