Red Hat
Enterprise Linux

KuberTENes is now a teenager, how did we get here?

Michael Bang
Snr Solutions Architect

Henrik Løvborg
Tech Sales Leader

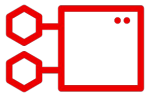
Intro

The birth of Kubernetes

Where did it all begin?

Containers

First expressed as a concept in 2006 by Google Engineer Rohit Seth. Popularised by Docker from 2013



Microservices

Driven by the possibilities of containerisation and the challenges that come with ever growing monoliths, microservice architecture is becoming more and more popular.

Cloud

Led by AWS, cloud is a central theme in the time kubernetes is born.

Leads to Kubernetes announced by Google in 2014

A lightweight container orchestration engine

Open sourced with the help of Red Hat

Google engineers realise the potential of their project and that they will need a community to scale it, which ends up with Google first open sourcing of a project

First commit June 6 2014

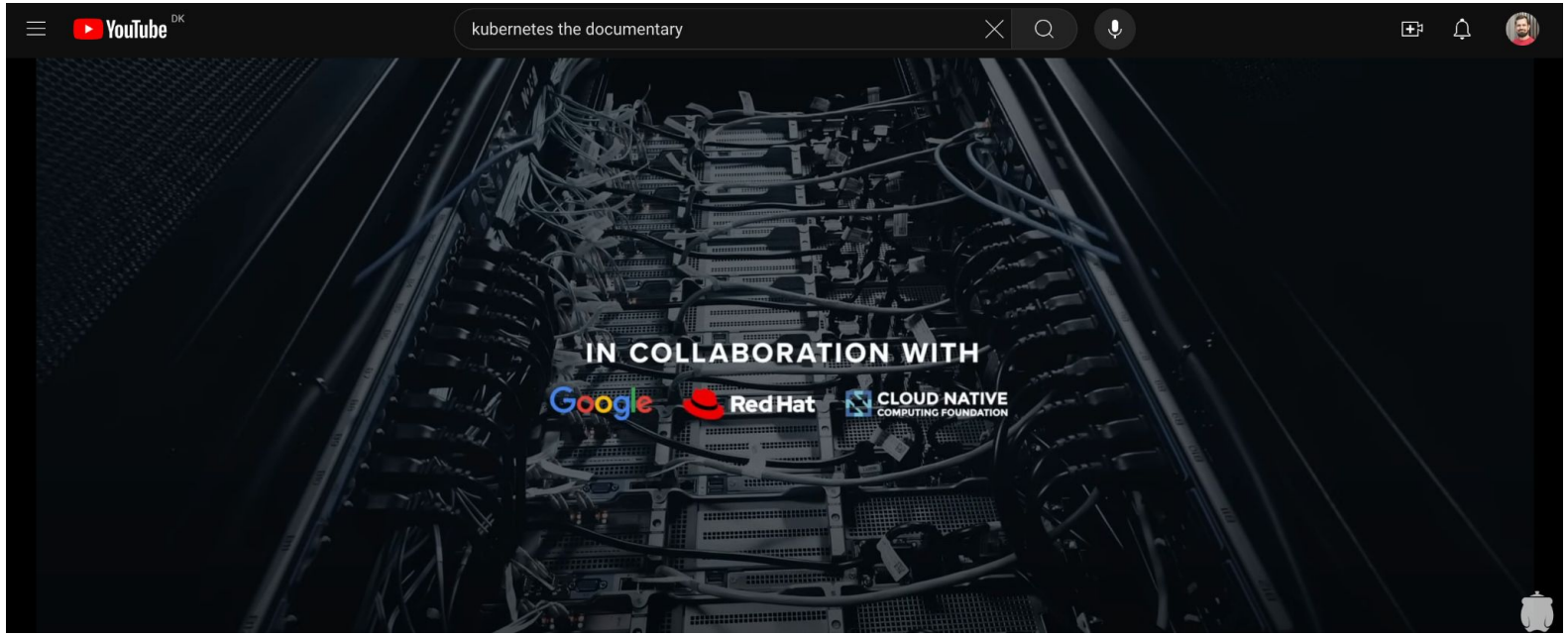
250 files and 47.501 lines of Go, Bash and Markdown make up the first commit of Kubernetes, by Google

Announced at Dockercon June 10 2014

Google VP of Infrastructure, Eric Brewer announces the open sourcing of Kubernetes in his keynote speech at Dockercon 2014 in San Francisco

Kubernetes: The Documentary

Available on YouTube



Source:
https://www.youtube.com/watch?v=BE77h7dmoQU&t=1s

The Origin Story

Released on YouTube and tells the details of the story of kubernetes with Google and Red Hat engineers involved in the process.
A thrilling story in 2 parts.

Red Hat

One click install?

Seems not - a new guide becomes the sys admins best friend

kelseyhightower/
kubernetes-the-hard-way

Bootstrap Kubernetes the hard way. No scripts.

52 Contributors

18 Used by

40k Stars

14k Forks

Source:
Insert source data here
Insert source data here

The final platform?

Or just the beginning

Source:
Insert source data here
Insert source data here

One company to rule it all?

Cloud Native Compute Foundation established

Major companies join forces for governing organ

Founding members include Google, CoreOS, Mesosphere, Red Hat, Twitter, Huawei, Intel, RX-M, Cisco, IBM, Docker, Univa, and VMware.

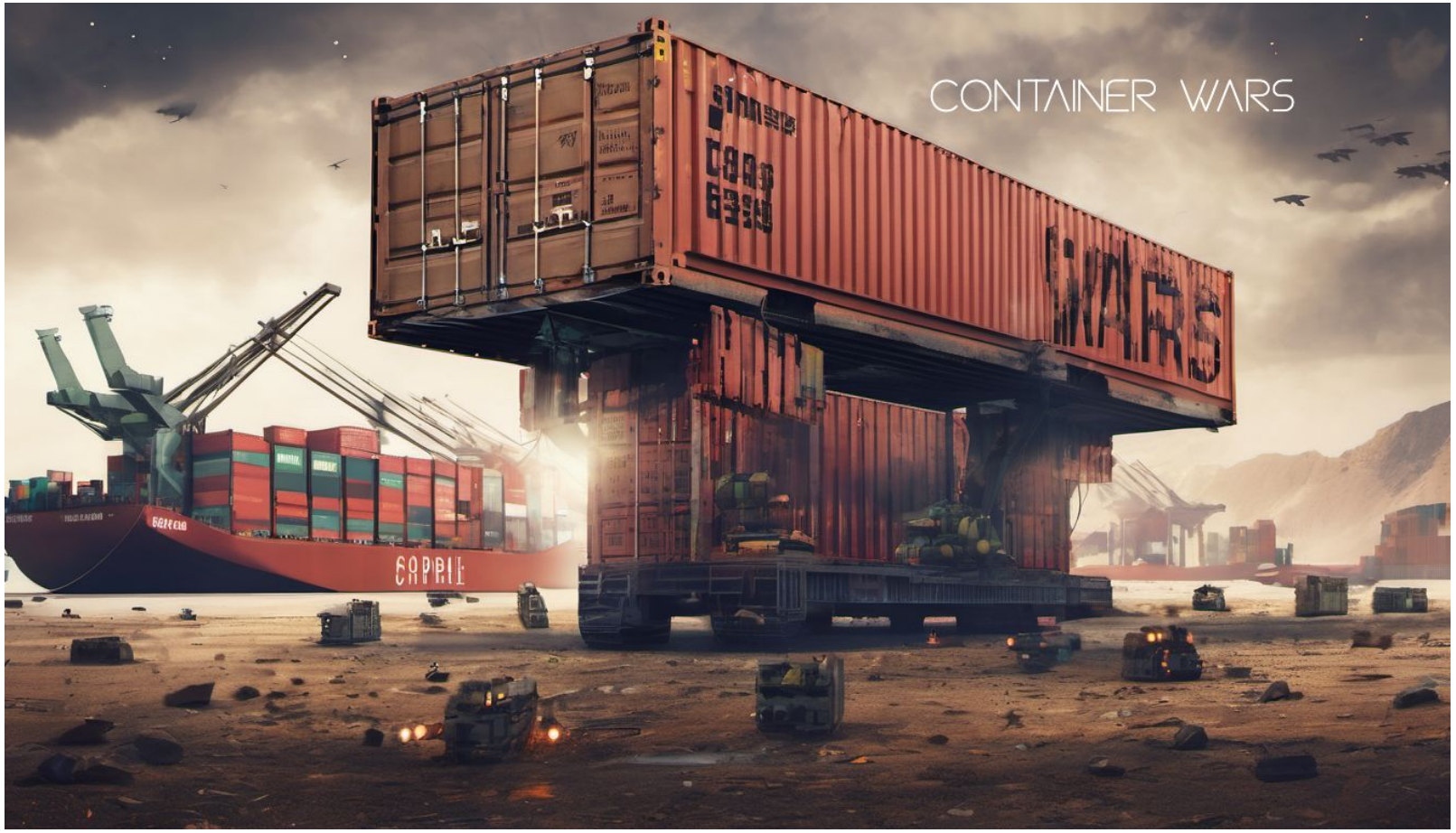
Establishes Working Groups and CNCF Landscape

With SIGs, TAGs and working groups, all relevant projects are governed and the direction is set by a multitude of actors in true open source fashion

Maturity Model, Certifications and The CNCF Landscape

Establishes important maturity components like the lifecycle model for projects, certifications for a range of skills and the notorious CNCF landscape to help navigate the many options

The Container Orchestration Wars



Red Hat

The Breakthrough

Why did Kubernetes break through ?

- Everything as code
  - Automation - GitOps
- Desired state
- Scalability
- Extensibility
- Resilience
- Technology agnostic
- Microservices
- DevOps
- Community

Burr Sutter: 9 steps to awesome

Available on YouTube

Source:
https://www.youtube.com/watch?v=ZpbXSdzp_vo

9 steps to awesome ...

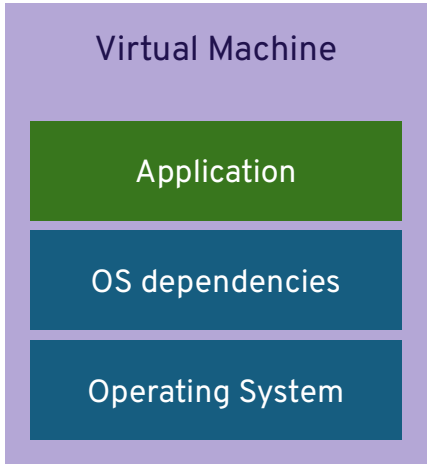
3 hours of showcasing the cool functions in Kubernetes. The video is 5 years old, but still highly relevant.

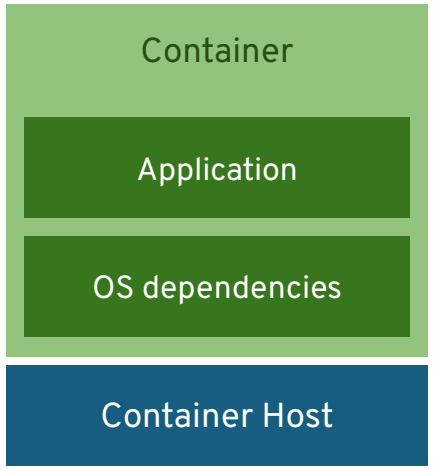
Red Hat

A Brief History of Kubernetes and OpenShift

- Kubernetes
  - Makes up the core of OpenShift
  - An open-source project introduced by Google in 2014
  - Based on Google's extensive experience running very large scale distributed applications in containers
- Red Hat, IBM, and CoreOS joined the Kubernetes Community that same year
- Kubernetes v1.0 and OpenShift 3.0 were released in mid 2015

VIRTUAL MACHINES AND CONTAINERS





- ✚ VM Isolation
- ➖ Complete OS
- ➖ Static Compute
- ➖ Static Memory
- ➖ High Resource Usage

- ✚ Container Isolation
- ✚ Shared Kernel
- ✚ Burstable Compute
- ✚ Burstable Memory
- ✚ Low Resource Usage

Inject data

- Environment Variables
- ConfigMaps
- Secrets

Demo

- Deployment of an application
- Desired state (restart of pod)
- Inject an environment variable

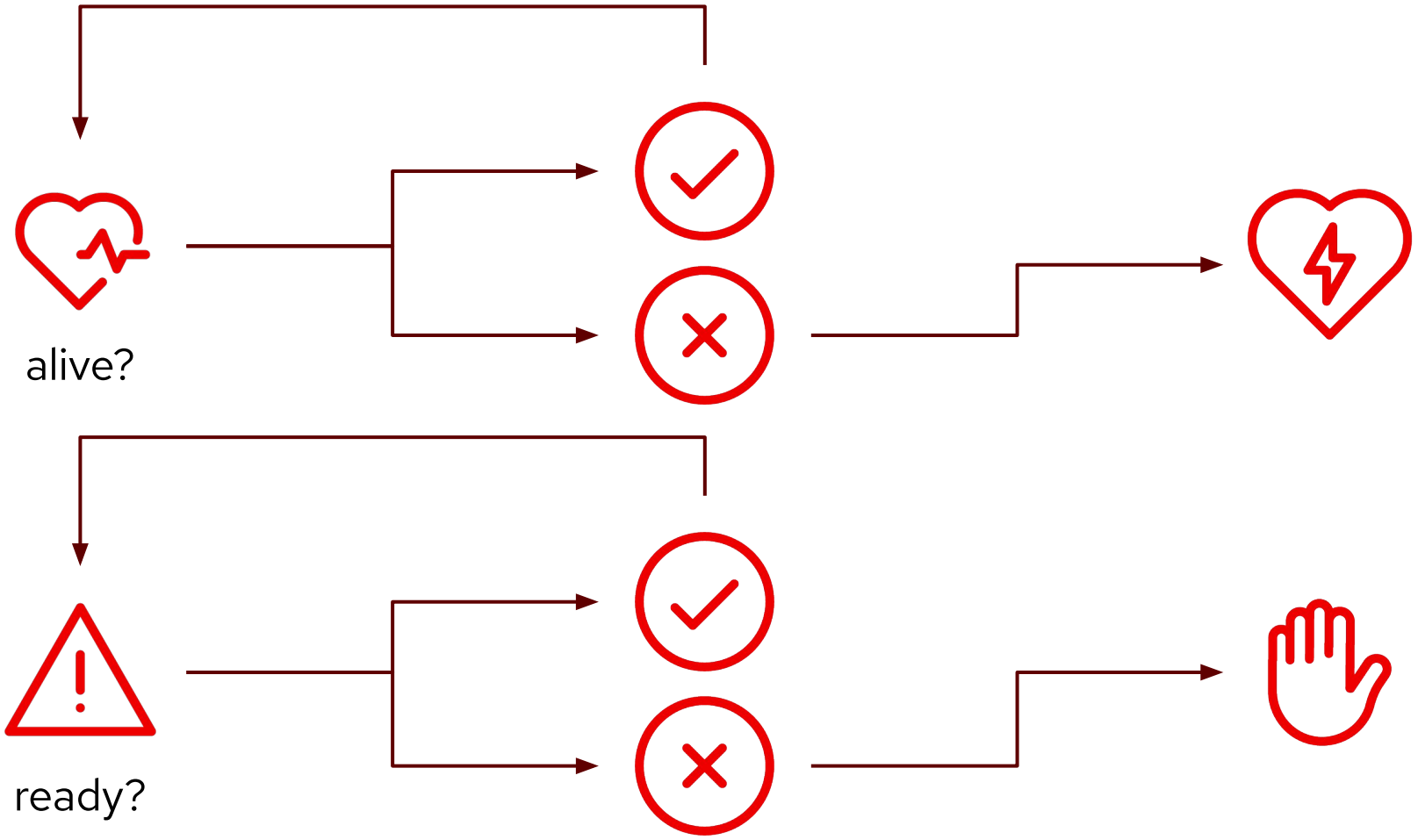
Scaling

- Without Kubernetes
  - Setting up a load balancer
  - Setting up certificates
- With Kubernetes
  - Batteries included (Services, Ingress, Routes...)

Scaling – Demo

- Scale a deployment

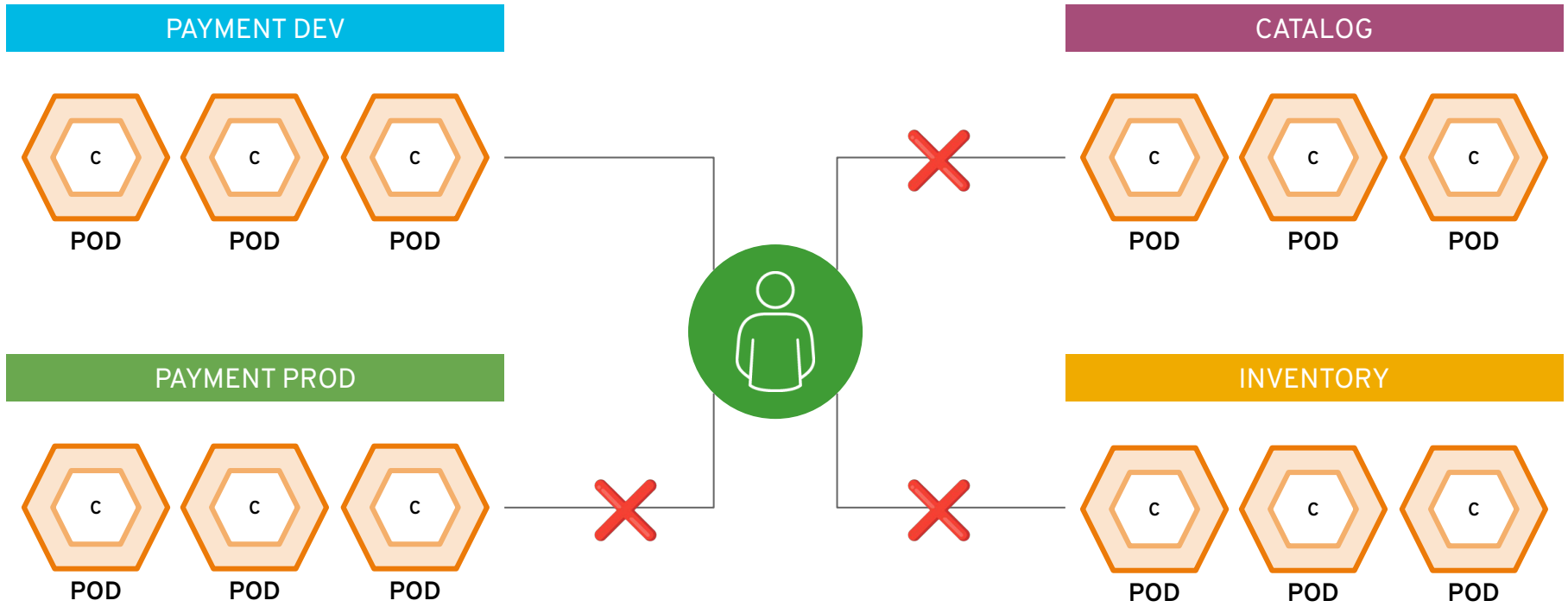
Liveness and Readiness



Readiness – Demo

- Add readiness and liveness probe to deployment
- Scale deployment (real ready state reflected)

projects isolate apps across environments, teams, groups and departments



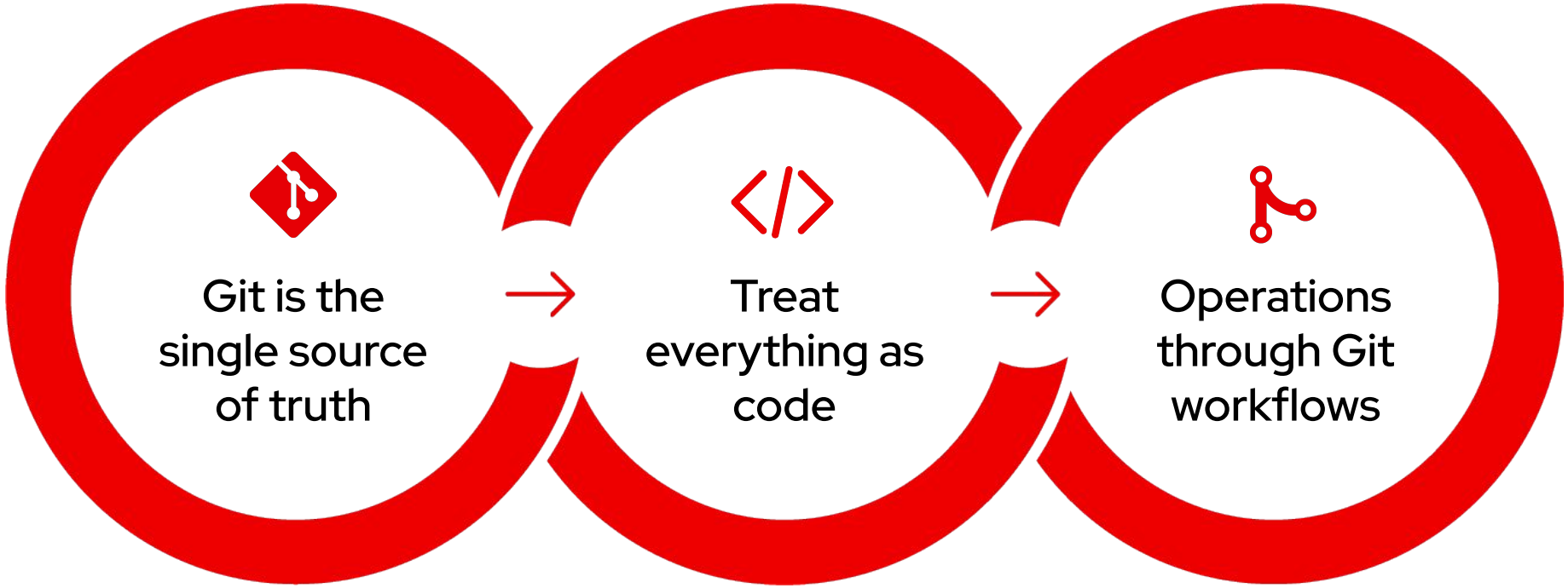
Red Hat

Pets vs Cattle

Pets vs. Cattle

What is GitOps?

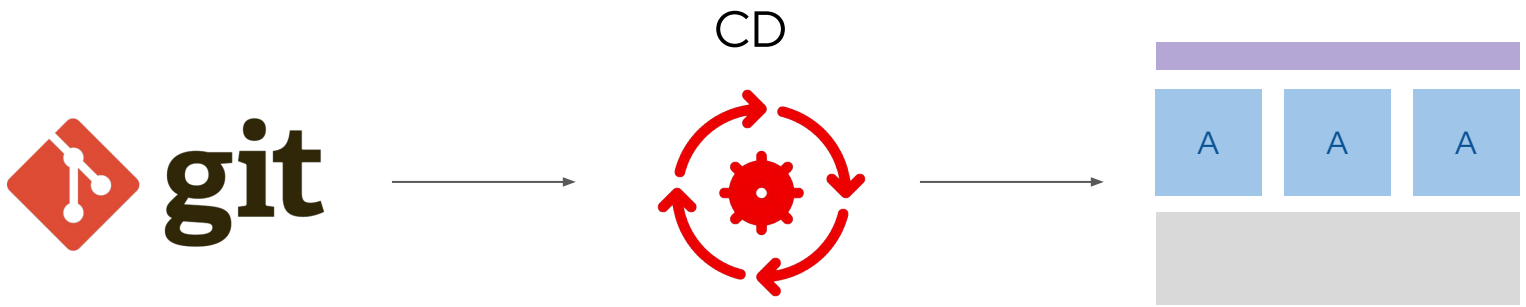
An developer-centric approach to Continuous Delivery and infrastructure operation



Red Hat

GitOps Workflow

a declarative approach to application delivery



What you want
(desired state)

What you have
(current state)

Red Hat

Java and .NET in containers

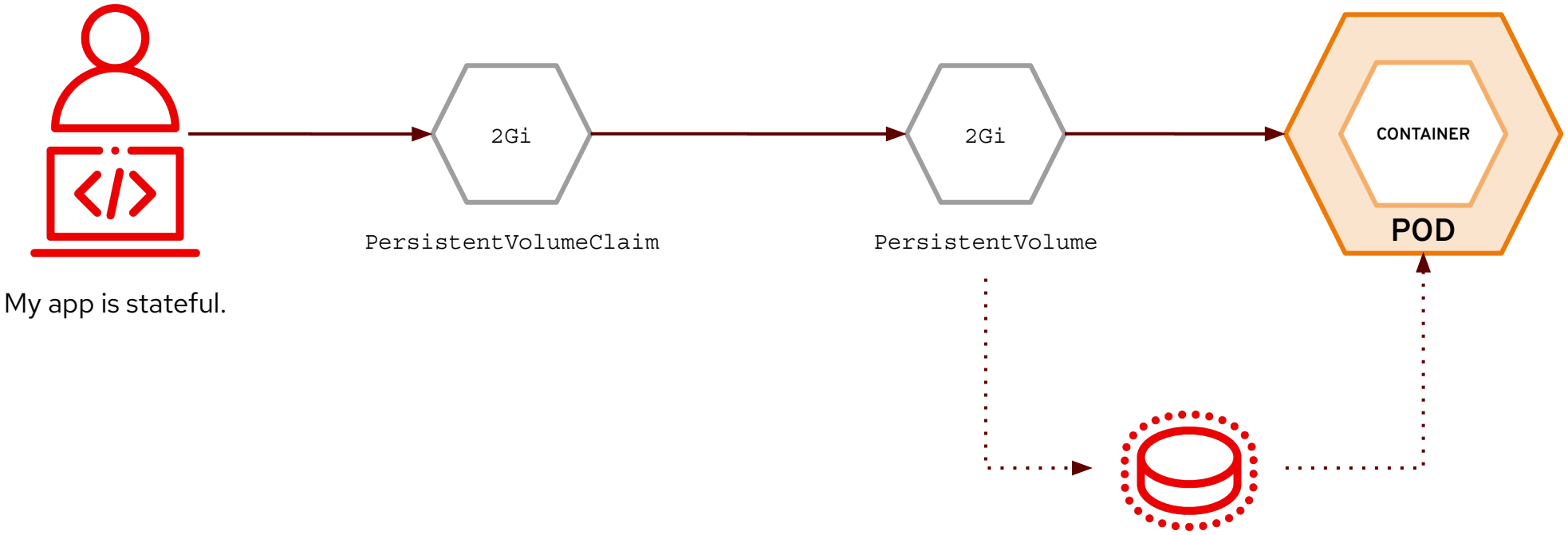

Designed for
Throughput
At the expense
of **footprint**


Designed to be **long-running**
At the expense of **startup speed**


Rich dynamic behavior
built for **mutable**
systems
Yet
containers are
primarily

Stateful?

Persistent Volume and Claims



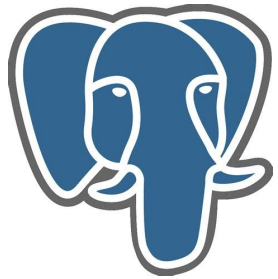
Red Hat

Databases on Kubernetes ?

Databases for non-production environments

Pointing to external databases for production

Full RDBMS in production on Kubernetes



Red Hat

Virtualization native to Kubernetes

- Operators are a Kubernetes-native way to introduce new capabilities
- New CustomResourceDefinitions (CRDs) for native VM integration, for example:
  - `VirtualMachine`
  - `VirtualMachineInstance`
  - `VirtualMachineInstanceMigration`
  - `DataVolume`

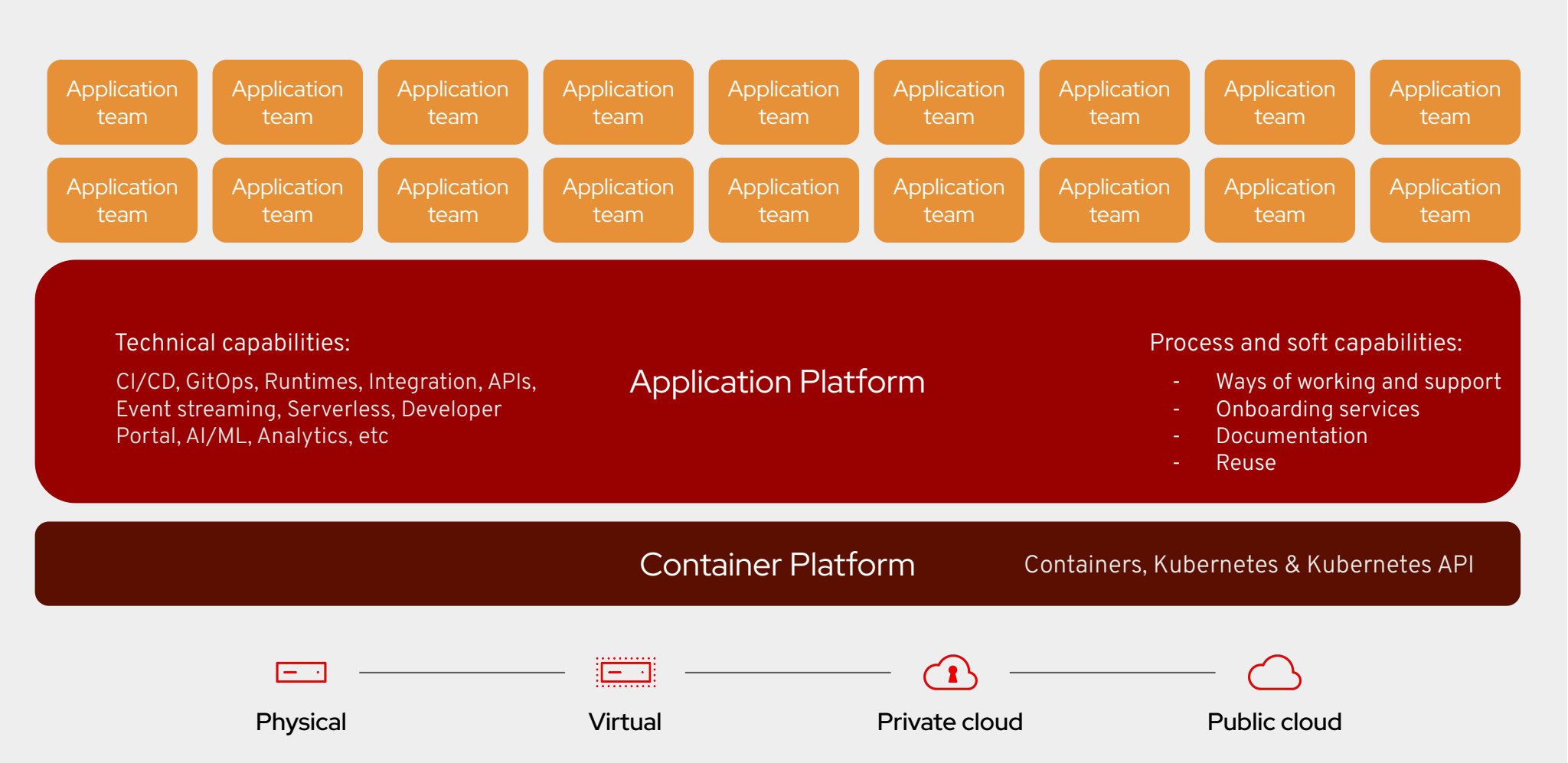
```yaml
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
  labels:
    app: demo
    flavor.template.kubevirt.io/small: "true"
  name: rhel
spec:
  dataVolumeTemplates:
  - apiVersion: cdi.kubevirt.io/v1alpha1
    kind: DataVolume
    metadata:
      creationTimestamp: null
      name: rhel-rootdisk
    spec:
      pvc:
        accessModes:
        - ReadWriteMany
        resources:
          requests:
            storage: 20Gi
        storageClassName: managed-nfs-storage
        volumeMode: Filesystem
```

Red Hat

Ways of working

Application and Platform model

Collaborating on shared problems

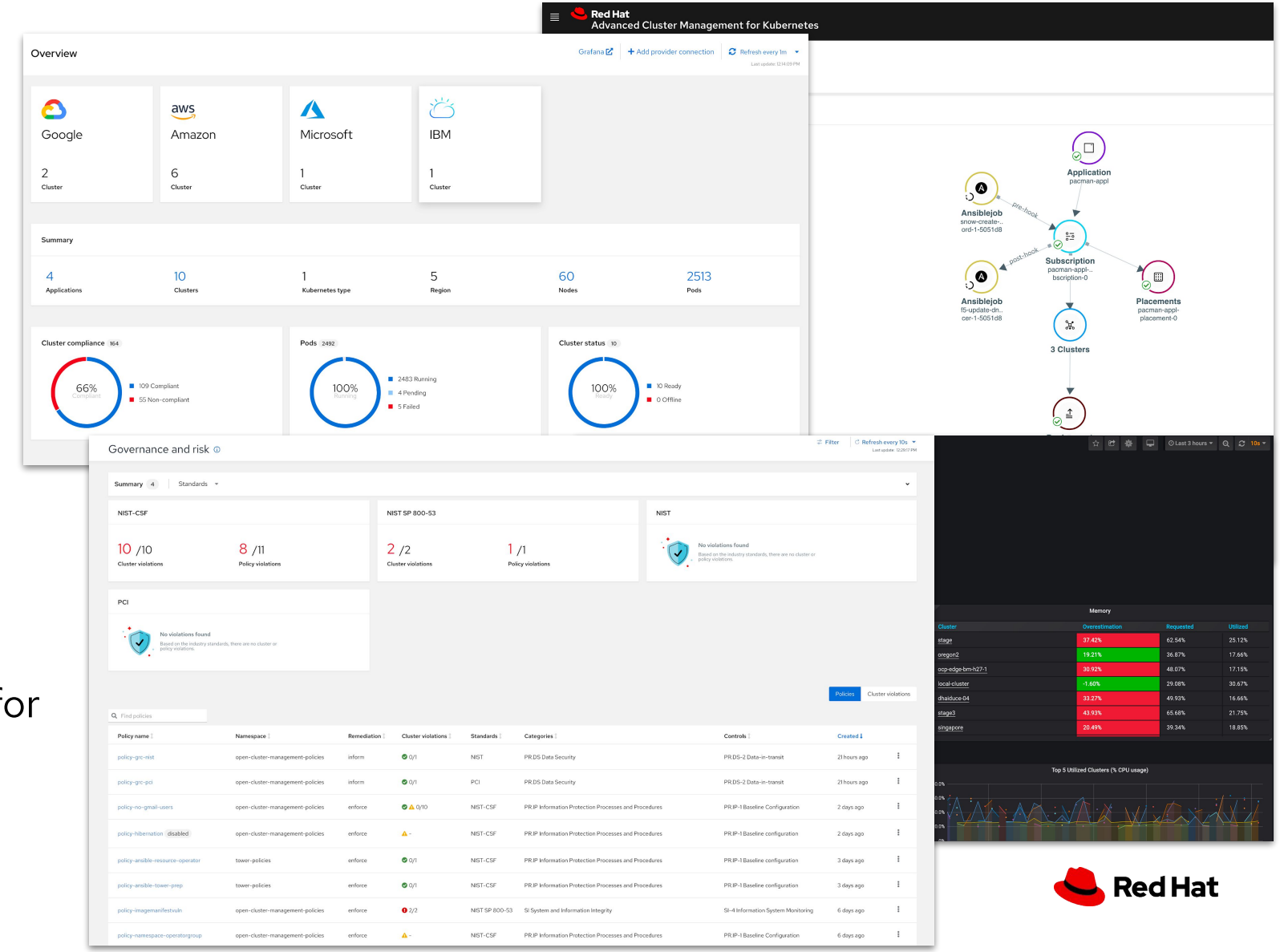


Red Hat

The Future

Robust. Proven. Award winning.

- Multicluster lifecycle management
- Policy driven governance, risk, and compliance
- Advanced application lifecycle management
- Multicluster observability for health and optimization



WHAT IS THE EDGE

Edge Tiers

Red Hat's focus

End-User Premises Edge

Provider Edge

Provider/Enterprise Core

Device or Sensor

Edge Endpoint

Edge Gateway

Edge Server

Provider Far Edge

Provider Access Edge

Provider Aggregation Edge

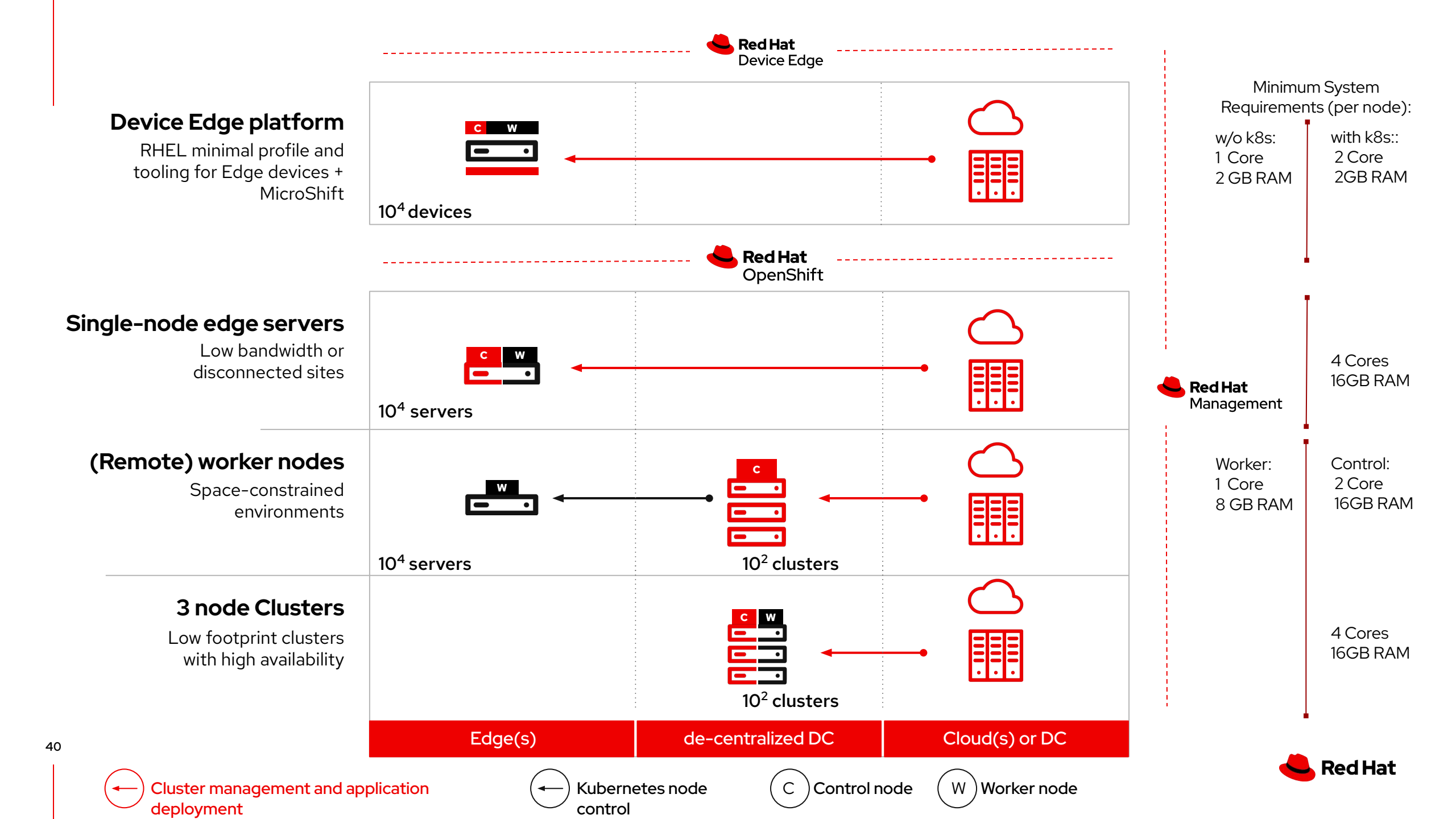
Regional Data Center

Core Data Center

Partners

"last mile"

* Edge computing == Fog computing (there is no real difference other t


Red Hat
Device Edge
Device Edge platform
RHEL minimal profile and tooling for Edge devices + MicroShift
C
W
10^4 devices
Red Hat
OpenShift
Single-node edge servers
Low bandwidth or disconnected sites
10^4 servers
(Remote) worker nodes
Space-constrained environments
10^4 servers
10^2 clusters
3 node Clusters
Low footprint clusters with high availability
10^2 clusters
Edge(s)
de-centralized DC
Cloud(s) or DC
Minimum System Requirements (per node):
w/o k8s:
1 Core
2 GB RAM
with k8s::
2 Core
2GB RAM
Red Hat
Management
4 Cores
16GB RAM
Worker:
1 Core
8 GB RAM
Control:
2 Core
16GB RAM
4 Cores
16GB RAM
Cluster management and application deployment
Kubernetes node control
C
Control node
W
Worker node
Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat