



RED HAT DEVELOPERS

Kubernetes Introduction

Rafael Benevides
benevides@redhat.com
@rafabene
<http://bit.ly/kubernetes-intro>

Rafael Benevides

Director of Developer Experience at Red Hat
Apache DeltaSpike P.M.C



✉ benevides@redhat.com

🐦 @rafabene

Java Certifications:

SCJA / SCJP / SCWCD / SCBCD / SCEA

JBoss Certifications:

JBCD / JBCAA

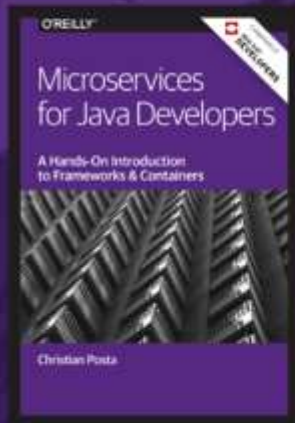
Red Hat Certifications:

OpenShift / Containers / Ansible

Other Certifications:

SAP Netweaver / ITIL / IBM Software Quality





MICROSERVICES FOR JAVA DEVELOPERS:

A hands-on introduction to frameworks and containers.

[DOWNLOAD NOW](#)

READ MORE ON MICROSERVICES

- 🔗 [Tear Down Data Silos with Microservices](#)
- 🔗 [Different types of Microservices?](#)
- 🔗 [Scalable Microservices through messaging](#)

**READY. SET.
CODE!** 7

RED HAT JOSS
ENTERPRISE
APPLICATION PLATFORM

Join Red Hat Developers and
try it now

.NET
on Red Hat
Enterprise Linux

Start using .NET on Linux
today

DEVNATION

June 26 - 29, 2016 • San Francisco, CA

Event Recap

Didn't make it to DevNation?
Watch Sessions OnDemand



OpenJDK from Red Hat
for Windows and you.

Join Red Hat Developers and
try OpenJDK

**MongoDB
Shell Cheat
Sheet**

Getting started, collections,
indexes, and dangers.
Download now.

Why do you want to run your application inside containers?

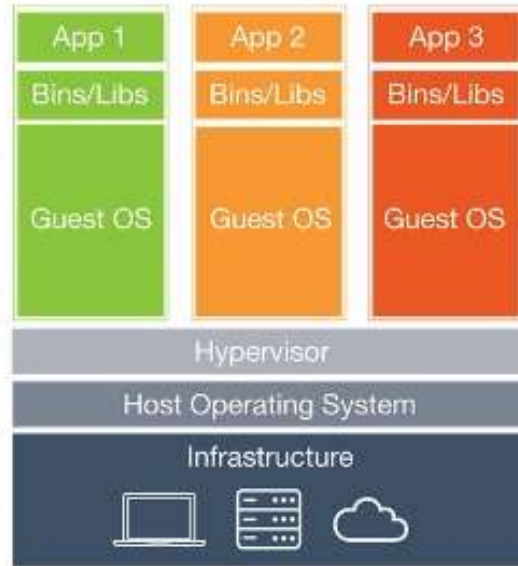


```
// A small Java program
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

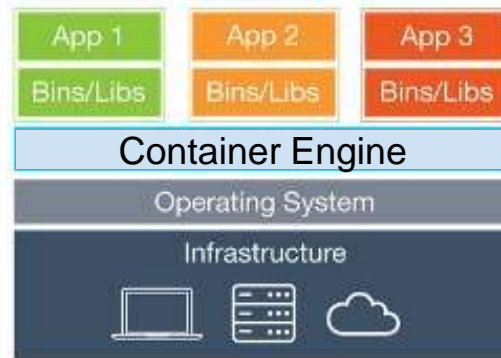


Container Advantages

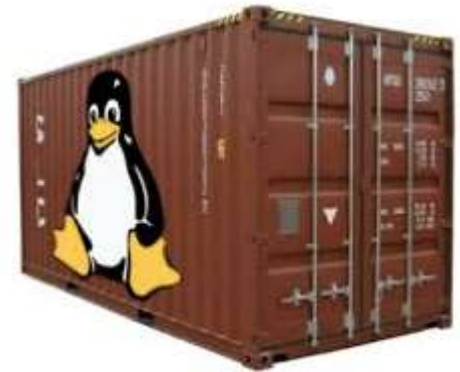
- Lightweight footprint and minimal overhead,
- Portability across machines,
- Simplify DevOps practices,
- Speeds up Continuous Integration,
- Empower Microservices Architectures.
- Isolation



Virtual Machines



Containers



A way to run a Linux container:

```
$ docker run -d <image-name>
```

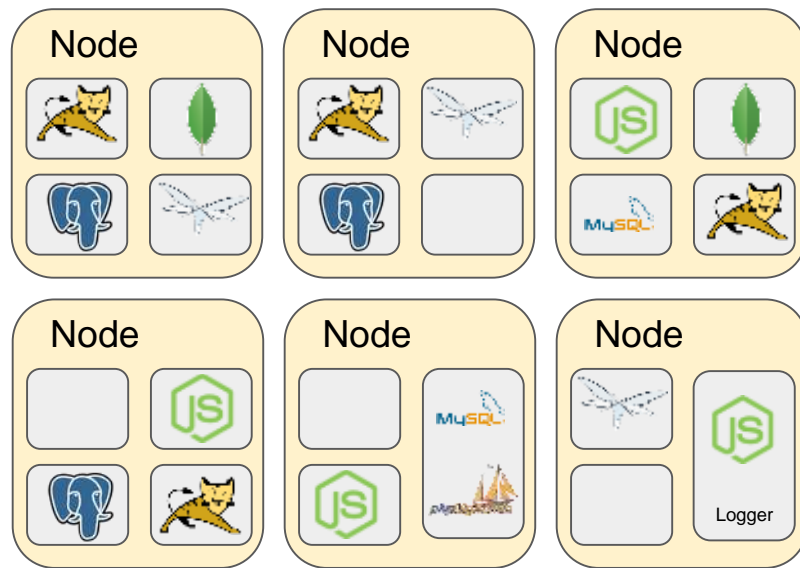
Linux Containers



A single and isolated Linux process
running in a single machine

DevOps challenges for multiple containers

- How to scale?
- How to avoid port conflicts?
- How to manage them in multiple hosts?
- What happens if a host has a trouble?
- How to keep them running?
- How to update them?
- Where are my containers?



Meet Kubernetes

Greek for “*Helmsman*”; also the root of the word “*Governor*” (from latin: *gubernator*)

- Container **orchestrator**
- Supports **multiple cloud** and **bare-metal** environments
- Inspired by Google’s experience with containers
- **Open source**, written in **Go**

Manage **applications**, not machines



Open Source community

Project

Version 1.7

Hosted on GitHub

1000+ contributors

52,000+ commits

25,000+ GitHub stars

Partners

Red Hat

CoreOS

HP

Pivotal

IBM

SaltStack

Mesosphere VMWare

Microsoft

<http://kubernetes.io/>

<https://github.com/kubernetes/kubernetes>



Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

Registry

Master

API Server

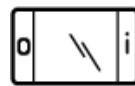
Kubernetes

OpenShift

- Deployments
- Builds
- ImageStreams

Controllers

- Scheduler
- Replication
- Services
- Builds
- Routes
- Deployment



Routing Layer

Node



Node

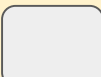


Node

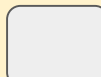


SDN Overlay Network

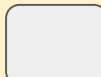
Node



Node



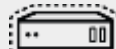
Node



Service Layer



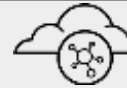
Physical



Virtual



Private



Public

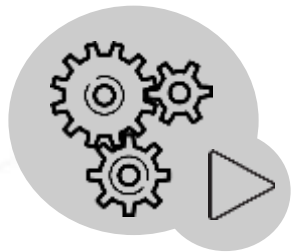


Persistent
Storage



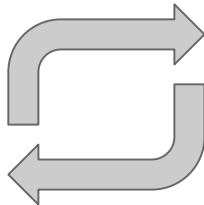
Kubernetes Concepts

Pod



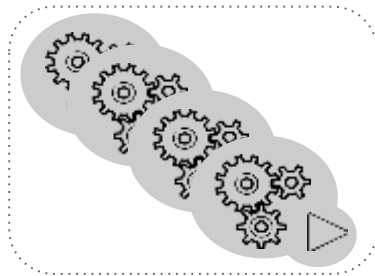
One or More Containers
Shared IP
Shared Storage Volume
Shared Resources
Shared Lifecycle

**Replication
Controller /
Deployment**



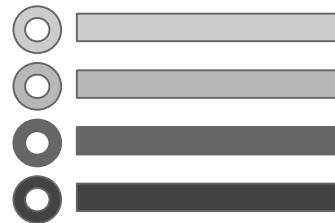
Ensures that a specified
number of pod replicas are
running at any one time

Service



Grouping of pods, act as
one, has stable virtual IP
and DNS name

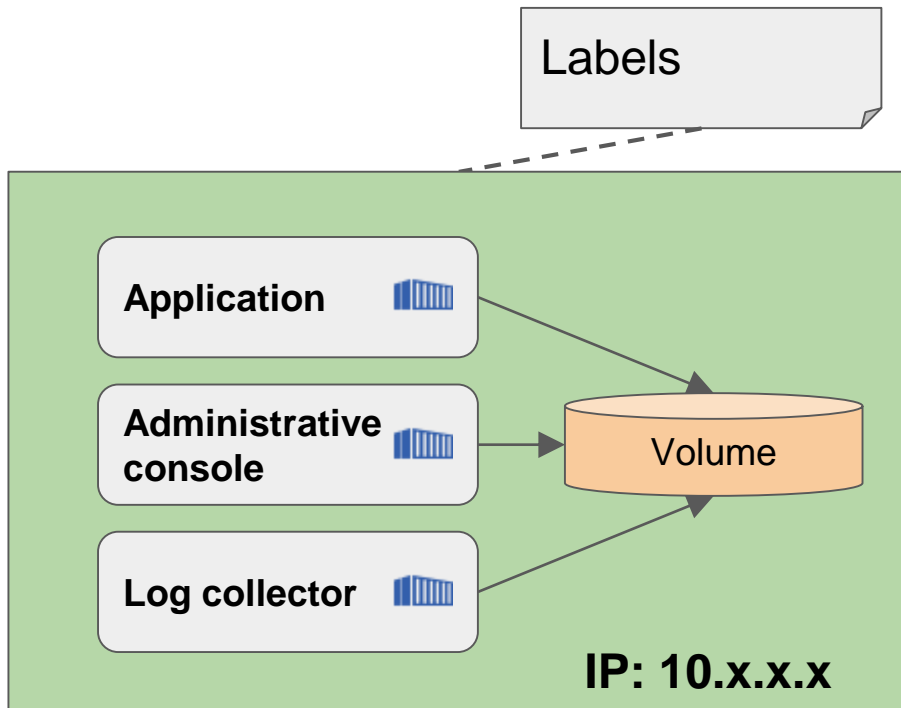
Label



Key/Value pairs associated
with Kubernetes objects
(e.g. env=production)

Concept: Pod

- Group of containers
- Live and die together
- Share:
 - IP
 - Secrets
 - Labels *
 - Volumes *



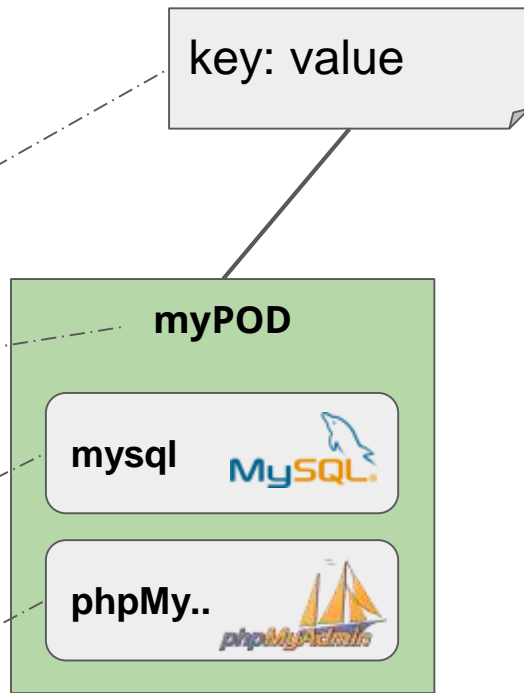
** we will talk about these concepts later*



Concept: POD

Defining a POD as YAML:

```
apiVersion: v1
kind: Pod
metadata:
  name: myPod
  labels:
    key: value
spec:
  containers:
    - name: mysql
      image: username/image
    - name: phpMyAdmin
      image: username/image2
```



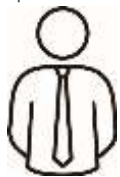


Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

Registry

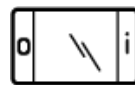
Master

API Server

Kubernetes

OpenShift
- Deployments
- Builds
- ImageStreams

Controllers
- Scheduler
- Replication
- Services
- Builds
- Routes
- Deployment



Routing Layer

Node



Node

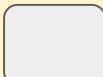


Node



SDN Overlay Network

Node



Node



Node

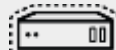


Logger

Service Layer



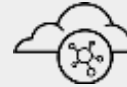
Physical



Virtual



Private



Public



Persistent
Storage



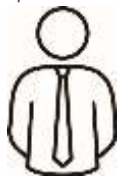


Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

Registry

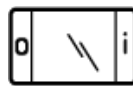
Master

API Server

Kubernetes

OpenShift
- Deployments
- Builds
- ImageStreams

Controllers
- Scheduler
- Replication
- Services
- Builds
- Routes
- Deployment



Routing Layer

Node



Node

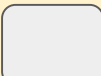


Node



SDN Overlay Network

Node



Node



Node

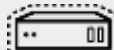


Logger

Service Layer



Physical



Virtual



Private



Public



Persistent
Storage





Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

Registry

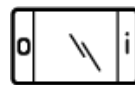
Master

API Server

Kubernetes

OpenShift
- Deployments
- Builds
- ImageStreams

Controllers
- Scheduler
- Replication
- Services
- Builds
- Routes
- Deployment



Routing Layer

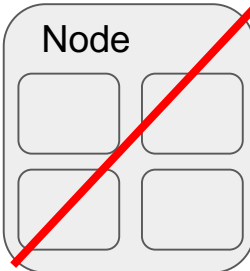
Node



Node



Node



SDN Overlay Network

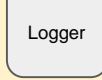
Node



Node



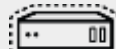
Node



Service Layer



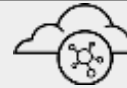
Physical



Virtual



Private



Public



Persistent
Storage



Concept: Replication Controllers / Deployment

Defining a Deployment as YAML:

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: myDeployment

spec:

replicas: 4

template:

metadata:

spec:

apiVersion: v1

kind: Pod

metadata:

name: myPod

labels:

key: value

spec:

containers:

- name: myPod

image: username/image

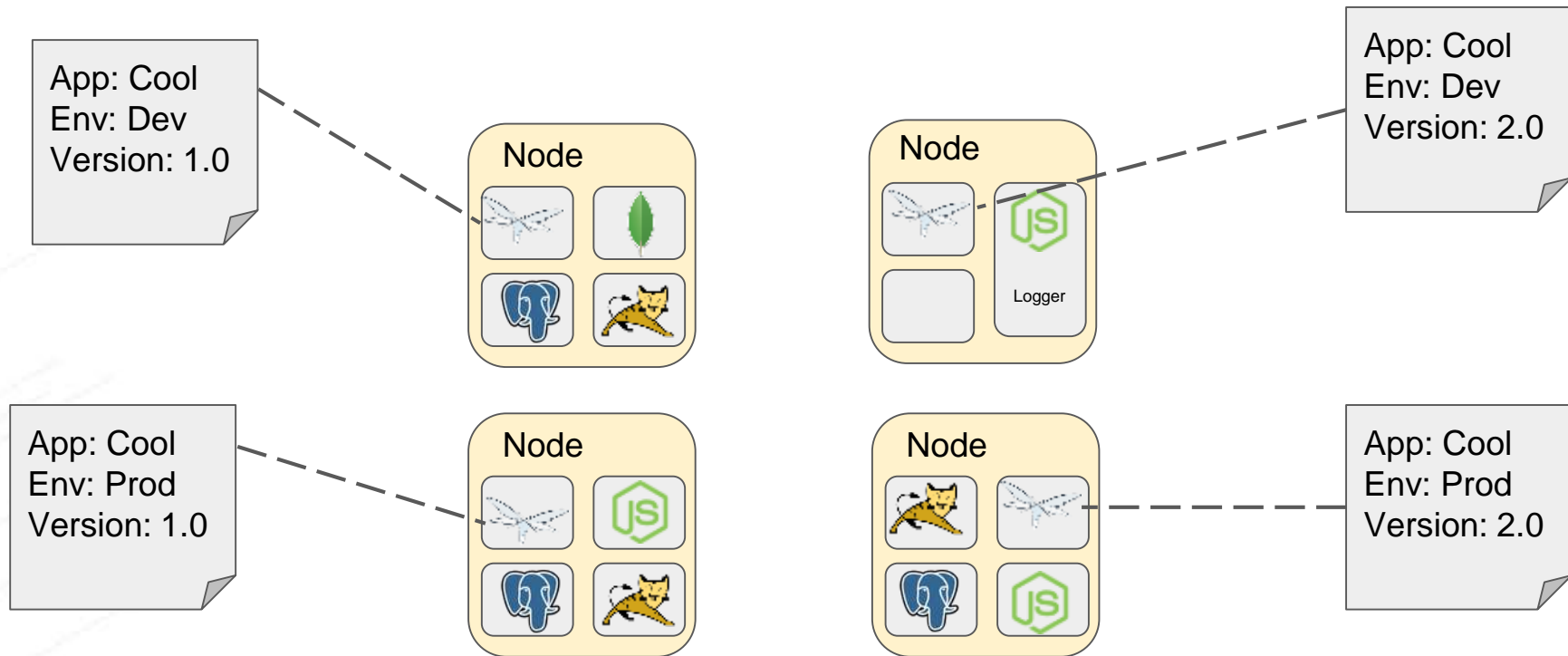
ports:

- name: http

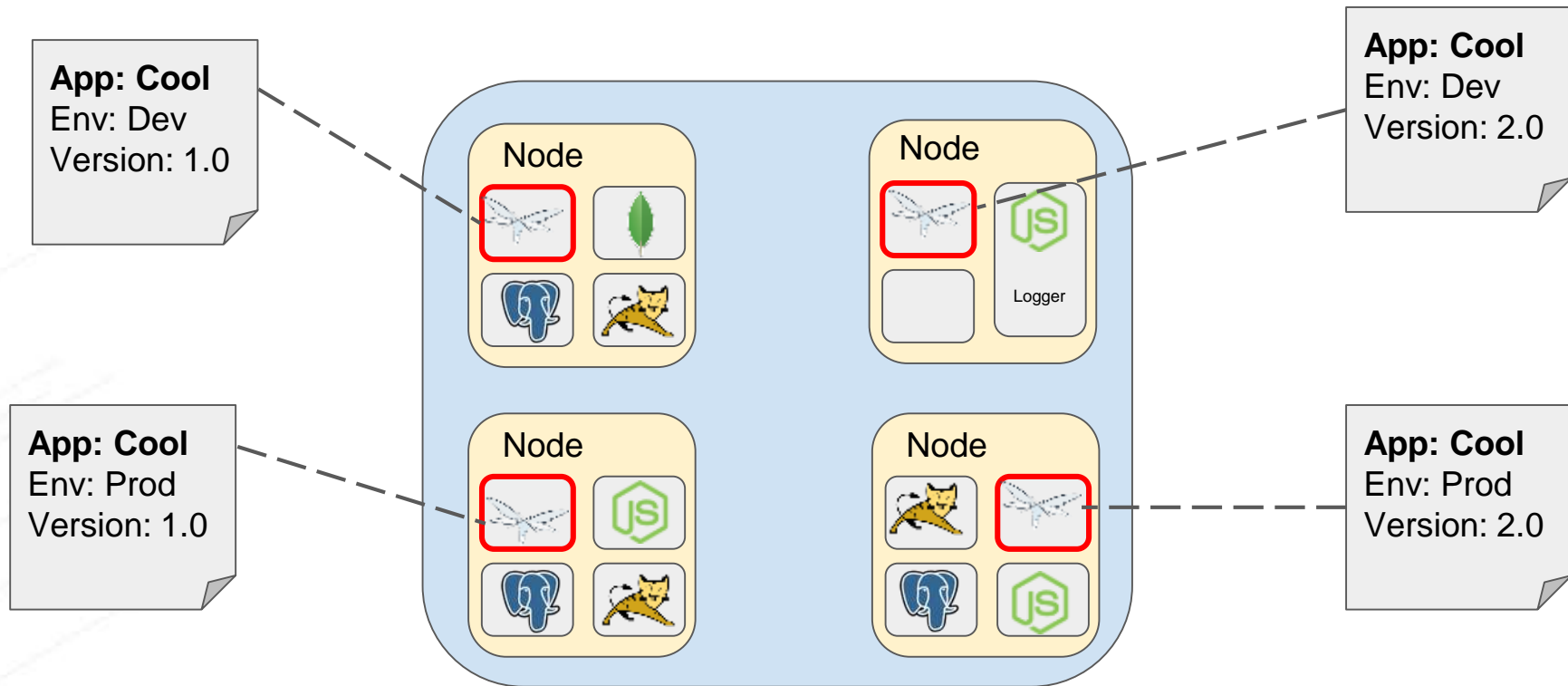
containerPort: 8080

Concept: Labels

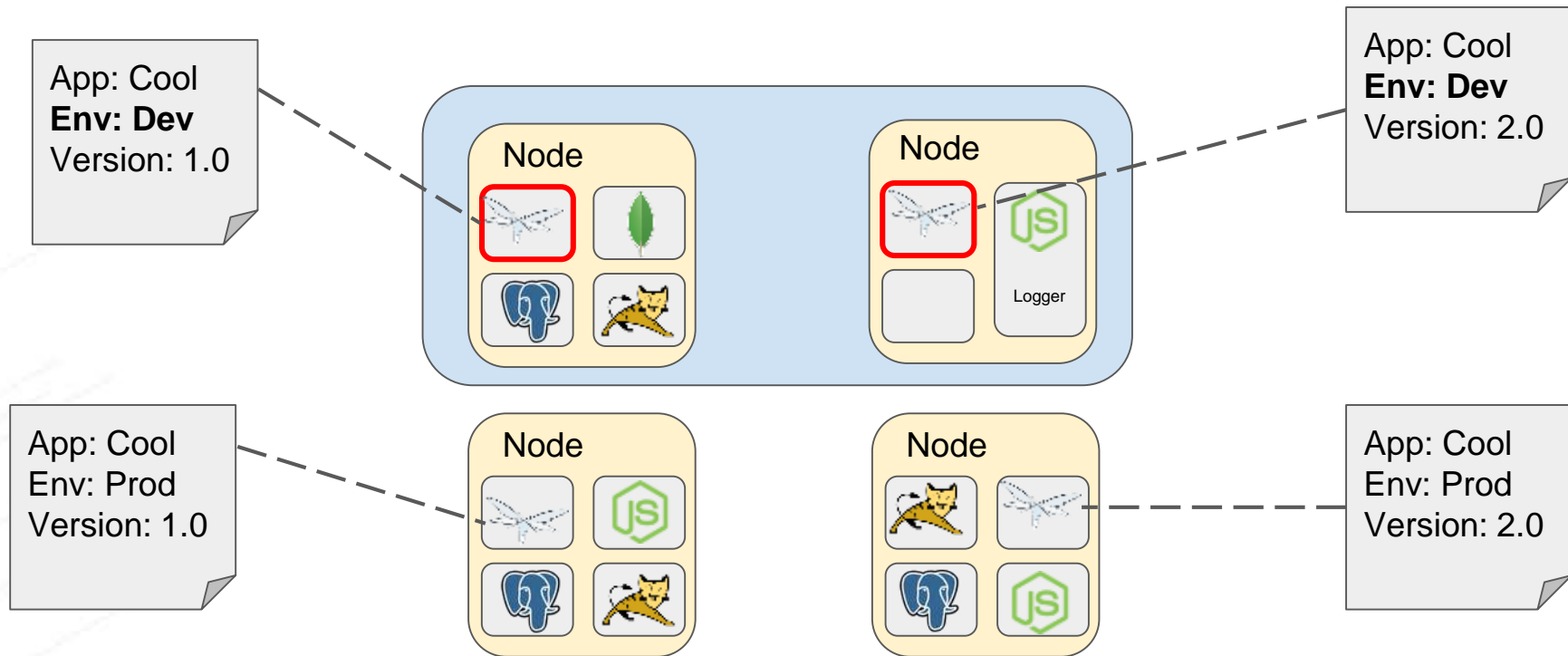
Everything in Kubernetes can have a label



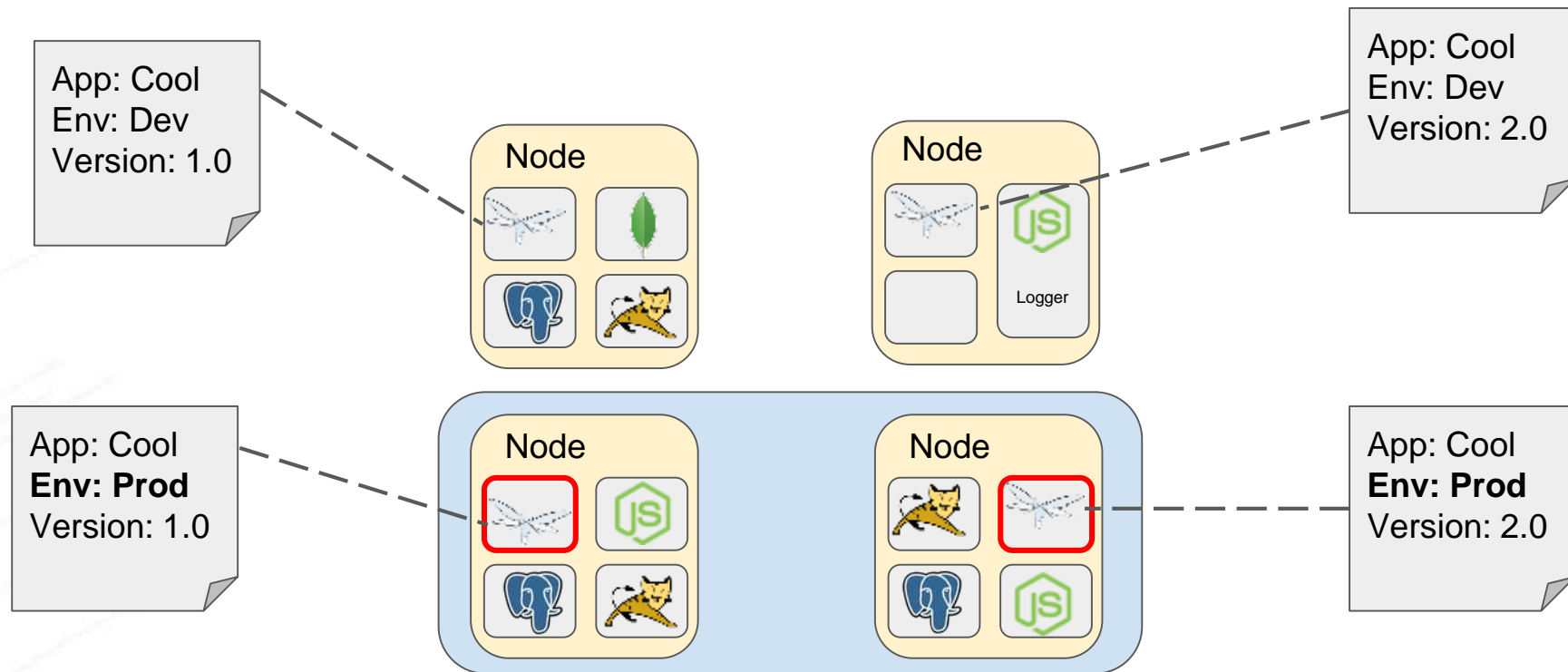
Concept: Labels



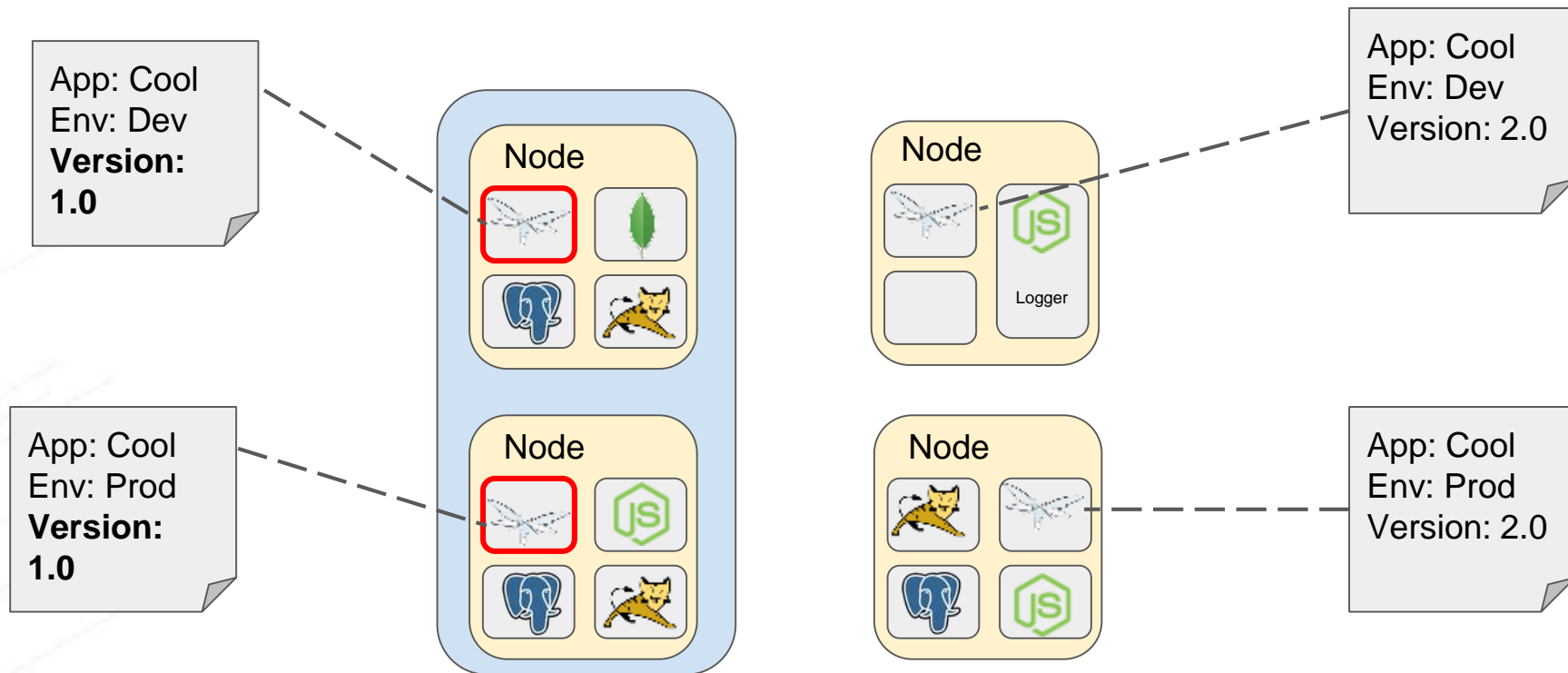
Concept: Labels



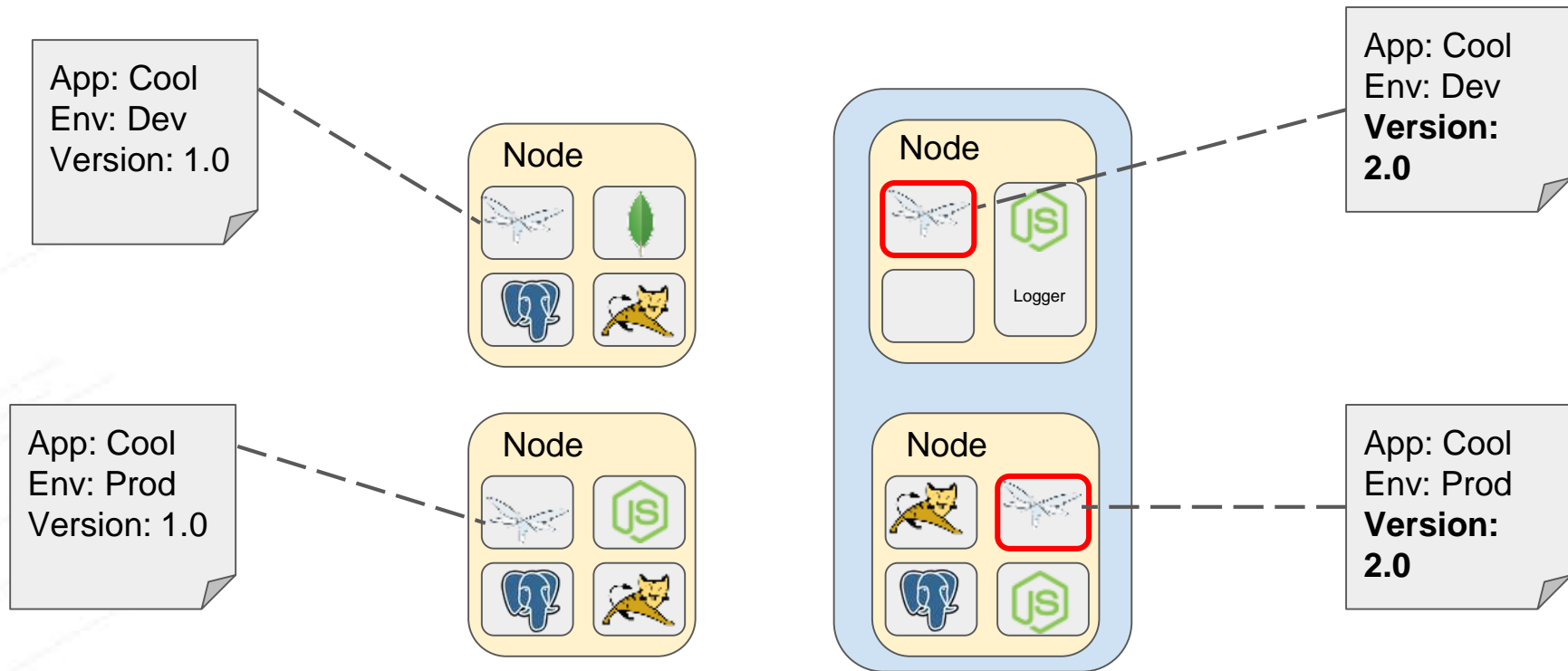
Concept: Labels



Concept: Labels



Concept: Labels



Concept: Labels

Defining Labels as YAML:
(can be placed in any object metadata)

metadata:


name: objectName

labels:

App: Cool

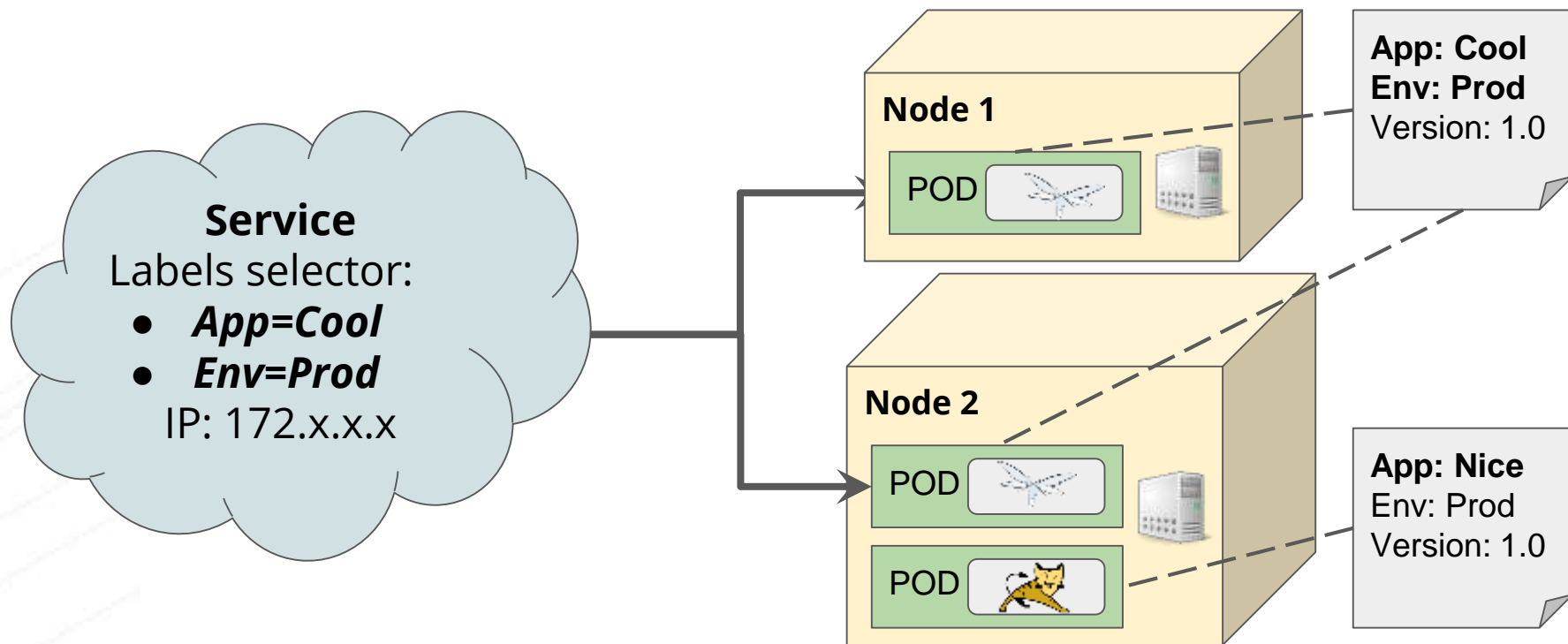
Env: Dev

Version: 1.0



App: Cool
Env: Dev
Version: 1.0

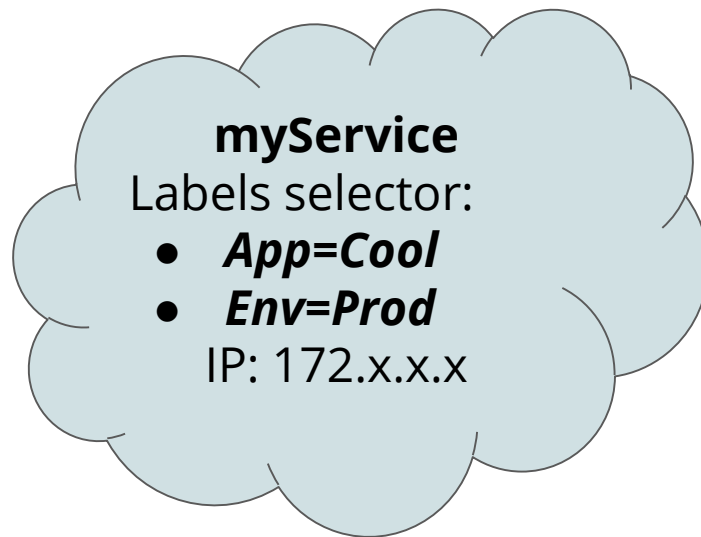
Concept: Services



Concept: Services

Defining a Service as YAML:

```
apiVersion: v1
kind: Service
metadata:
  name: myService
  labels:
    ...
spec:
  ports:
    - port: 80
      targetPort: 80
  selector:
    App: Cool
    Env: Prod
```



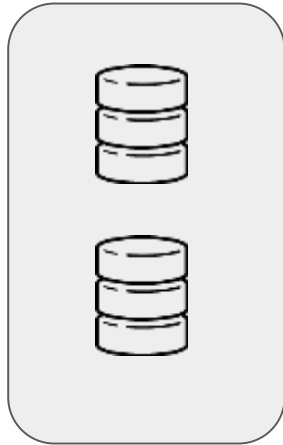
Service discovery inside Kubernetes

Using Environment variables:

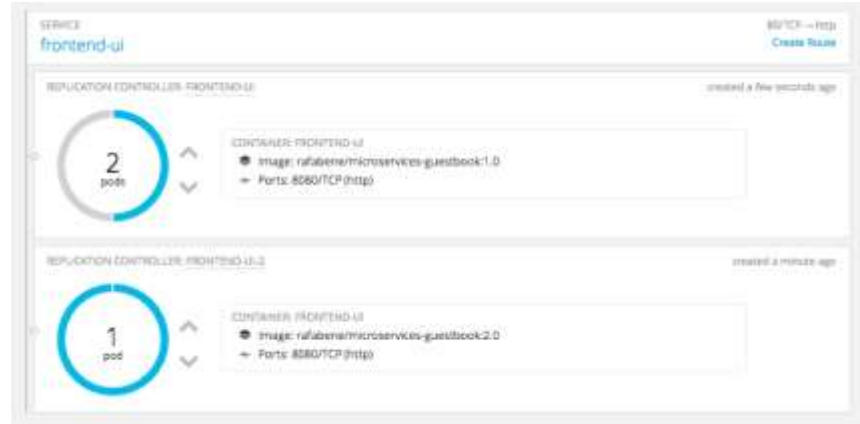
```
sh-4.2$ set|grep MYSQL
MYSQL_PORT=tcp://172.30.154.164:3306
MYSQL_PORT_3306_TCP=tcp://172.30.154.164:3306
MYSQL_PORT_3306_TCP_ADDR=172.30.154.164
MYSQL_PORT_3306_TCP_PORT=3306
MYSQL_PORT_3306_TCP_PROTO=tcp
MYSQL_SERVICE_HOST=172.30.154.164
MYSQL_SERVICE_PORT=3306
```

Using internal DNS: `$ ping mysql`

Other concepts



Persistent Volumes

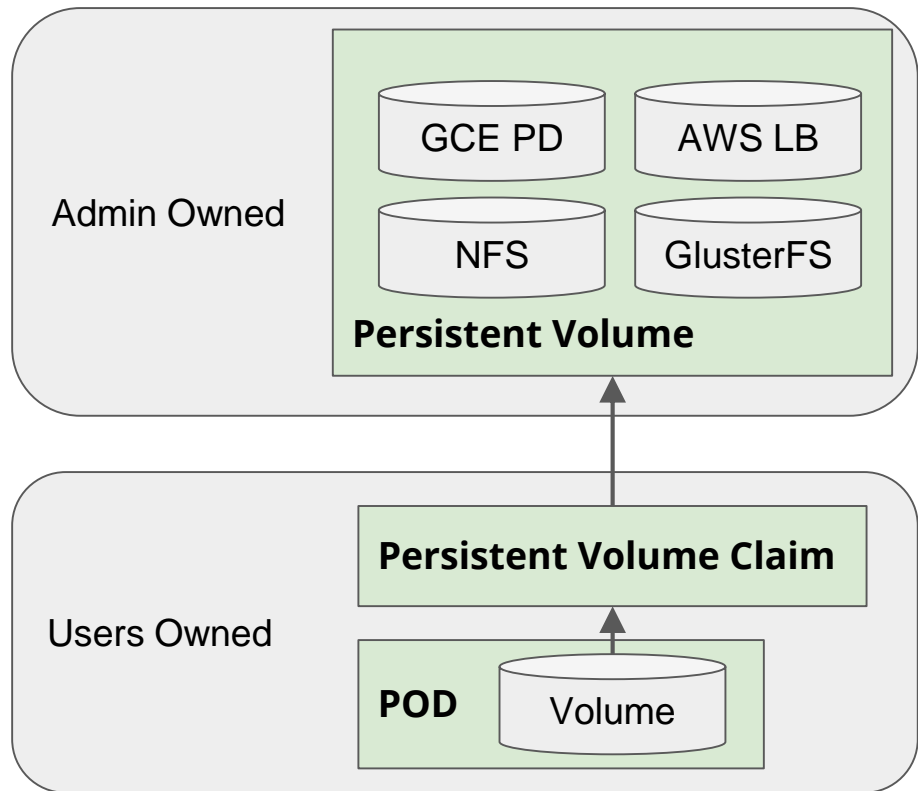


Rolling updates

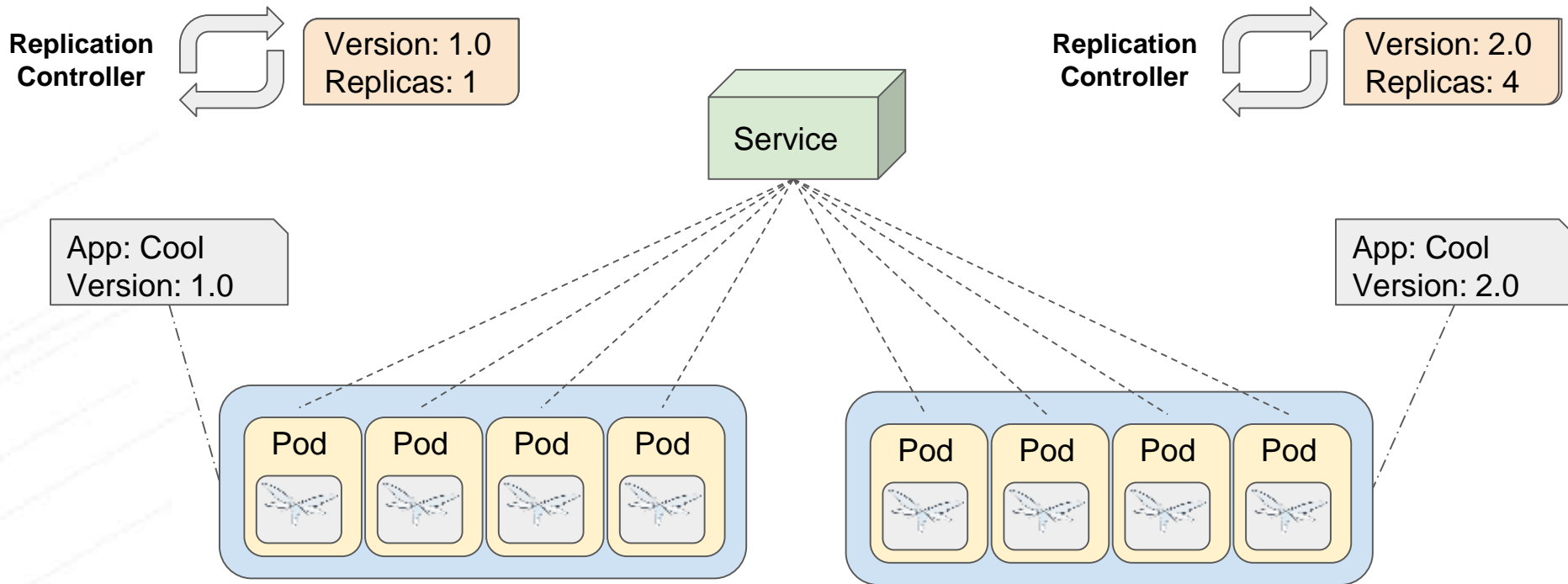
Concept: Persistent Volumes

- Admin provisions them, Users claim them
- High-level abstraction
- Pods can mount PVCs as Volumes

```
volumeMounts:  
  # name must match the volume name below  
  - name: mysql-persistent-volume  
    # mount path within the container  
    mountPath: /var/lib/mysql/data  
volumes:  
  - name: mysql-persistent-volume  
    persistentVolumeClaim:  
      claimName: mysql-pvc
```

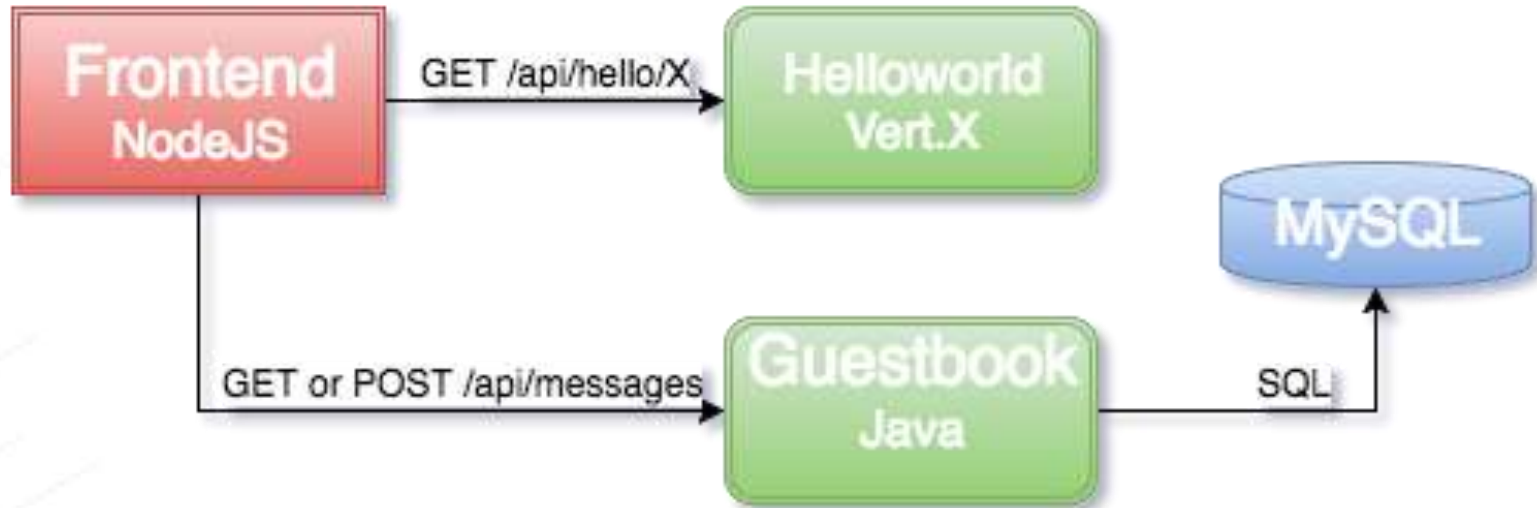


Deployment Concept: Rolling Updates



Kubernetes lab

Application Overview



Say Hi - Rafael

Your name

Message

Submit



Hello Rafael from helloworld-service-vertx-375244497-5zgtz with 1.0 from config file modified

Name	Message
Rafael	Hello DevNation!!!!
Rafael	teste1
Benevides	Hello World

Say Hi - Rafael

Guestbook Service -
Create

Your name

Rafael

Message

Submit

Hello World Service
- Greet



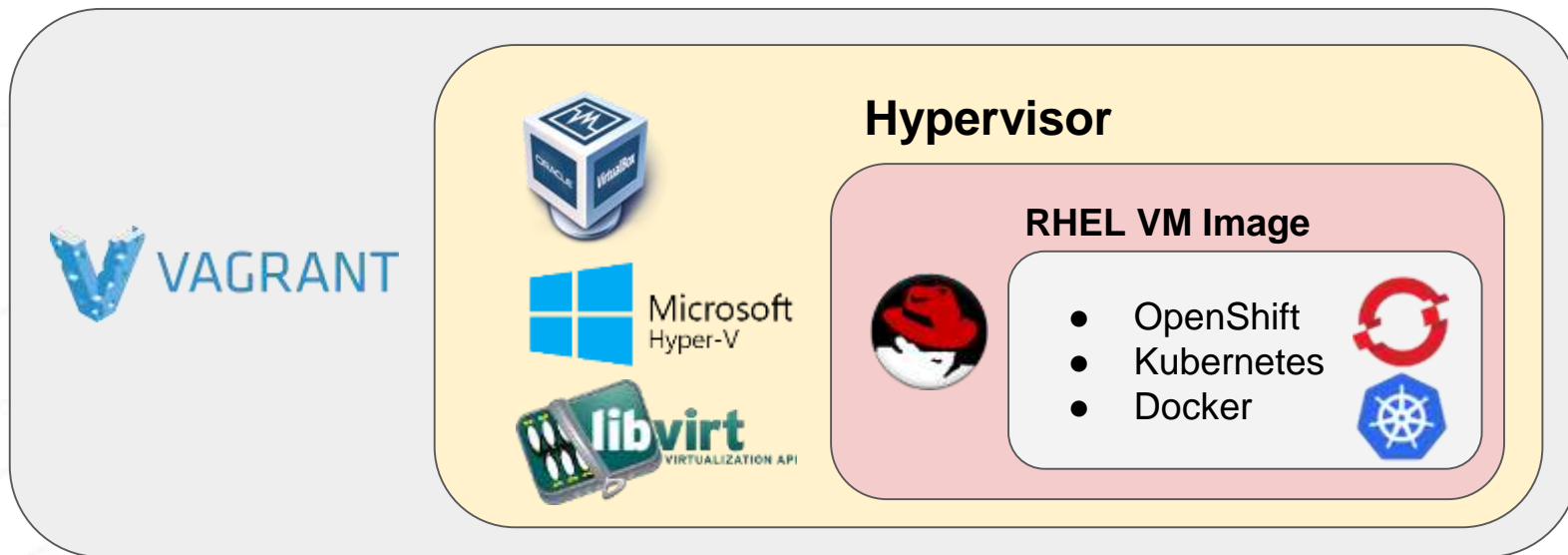
Hello Rafael from helloworld-service-vertx-375244497-5zgtz with 1.0 from config file modified

Name	Message
Rafael	Hello DevNation!!!!
Rafael	teste1
Benevides	Hello World

Guestbook Service -
Retrieve

Lab infrastructure

Container Development Kit



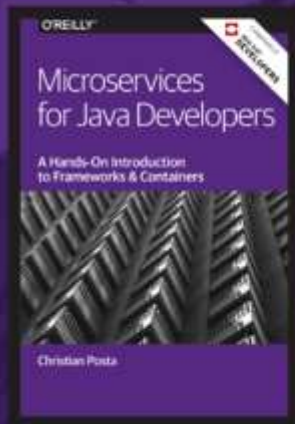
<http://developers.redhat.com/products/cdk/download/>

Kubernetes lab

VERY IMPORTANT

<http://bit.ly/kubernetes-lab>

Follow me on the Setup environment section!



MICROSERVICES FOR JAVA DEVELOPERS:

A hands-on introduction to frameworks and containers.

[DOWNLOAD NOW](#)

READ MORE ON MICROSERVICES

- 🔗 [Tear Down Data Silos with Microservices](#)
- 🔗 [Different types of Microservices?](#)
- 🔗 [Scalable Microservices through messaging](#)

**READY. SET.
CODE!** 7

RED HAT JOSS
ENTERPRISE
APPLICATION PLATFORM

Join Red Hat Developers and
try it now

.NET
on Red Hat
Enterprise Linux

Start using .NET on Linux
today

DEVNATION
June 26 - 29, 2016 • San Francisco, CA

Event Recap

Didn't make it to DevNation?
Watch Sessions OnDemand



OpenJDK from Red Hat
for Windows and you.

Join Red Hat Developers and
try OpenJDK

**MongoDB
Shell Cheat
Sheet**

Getting started, collections,
indexes, and dangers.
Download now.



@RAFABENE