

Kubernetes: From 0 to Production-Grade with Java



Kevin Dubois
Senior Principal Developer Advocate
Red Hat

 @kevindubois.com

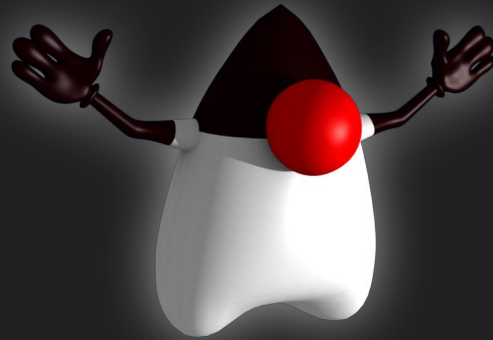
Kevin Dubois



- ★ Sr. Principal Developer Advocate at Red Hat 
- ★ Java Champion 
- ★ From Belgium 
- ★ Professional Software Engineer for ~20 years
- ★ Working with Java and Cloud Native for ~10 years
- ★  Speak English, Dutch, French, Italian

Java™
Champions @kevindubois.com youtube.com/@thekevindubois linkedin.com/in/kevindubois github.com/kdubois

“Kube Native” Java !?

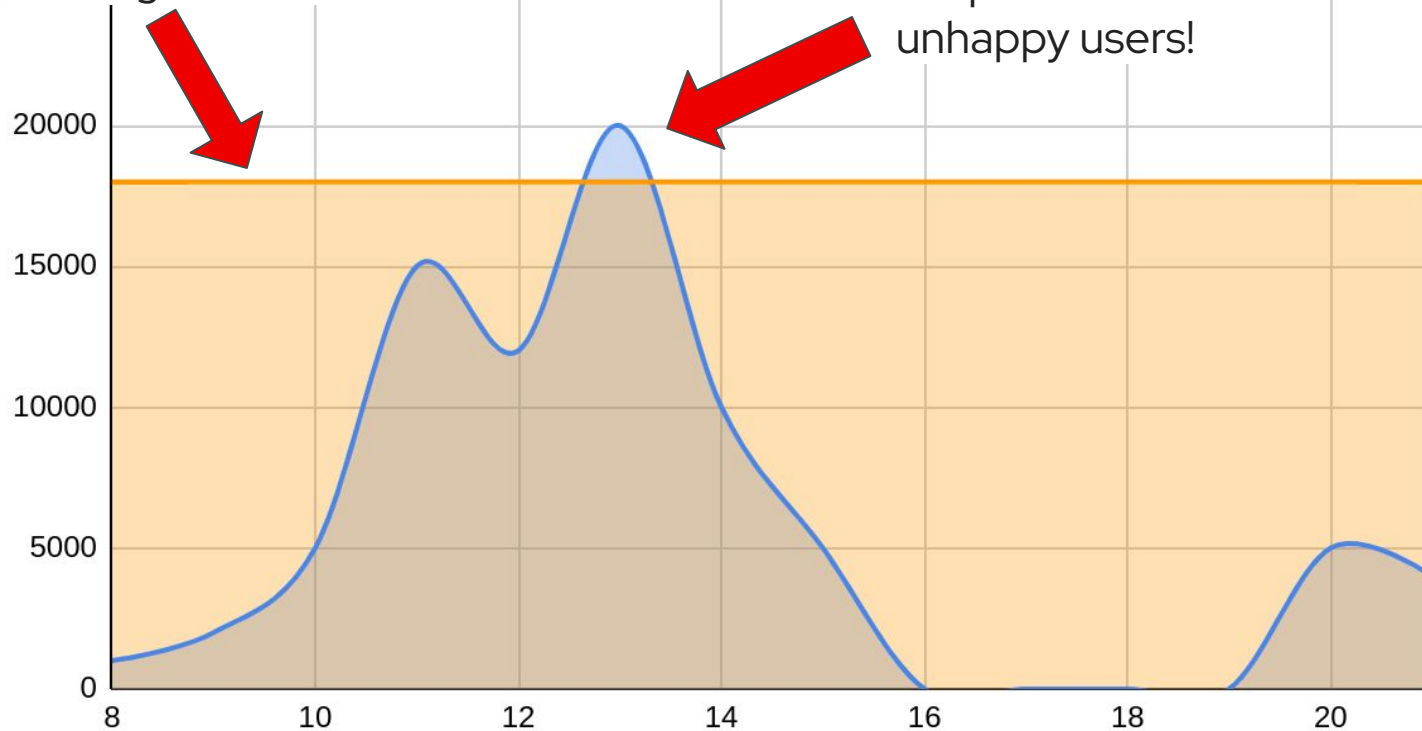


Traditional Deployments

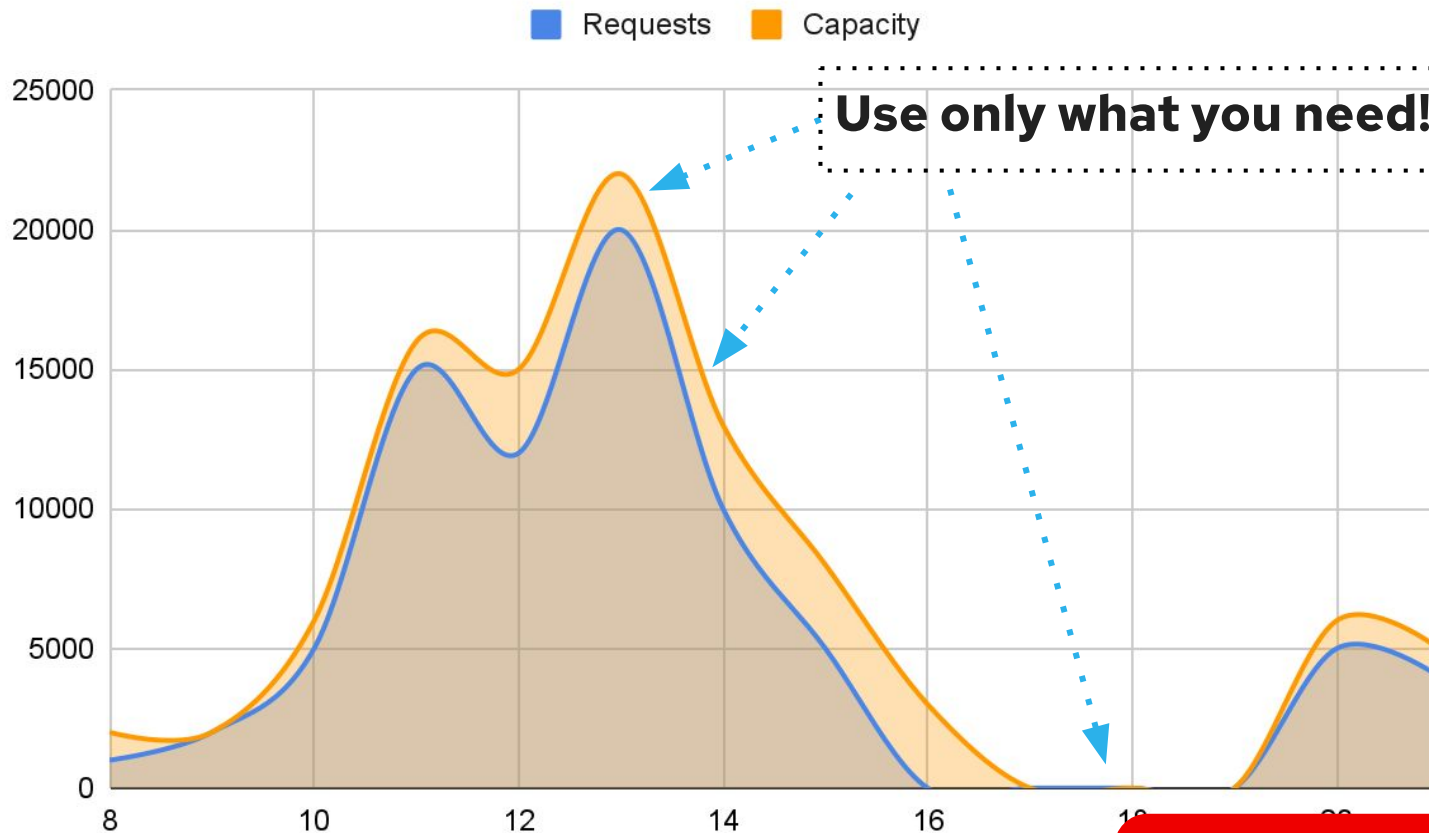
overprovisioned &
wasting resources

Requests Capacity

underprovisioned &
unhappy users!



Cloud Computing



with Serverless

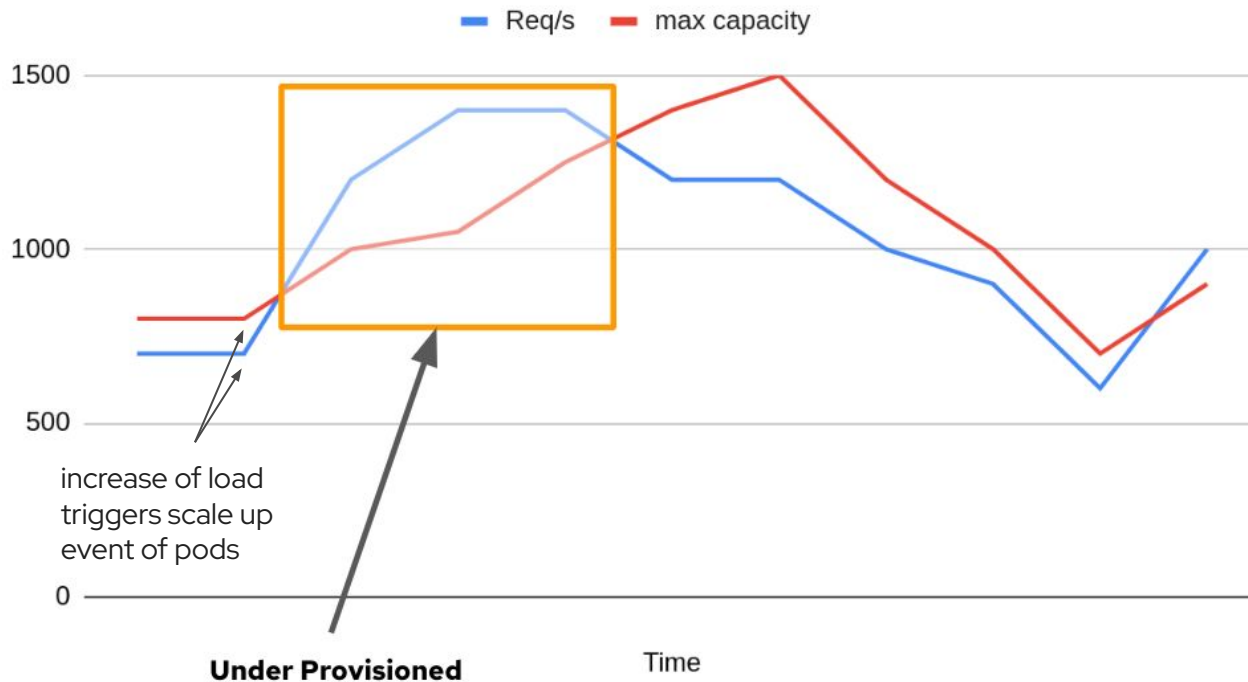
When you can't scale fast enough...



© CEN

JVM Scaling

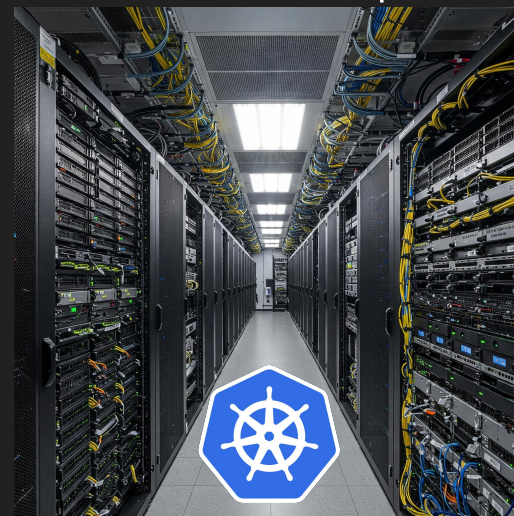
Req/s vs max capacity





"Traditional" Java

- Throughput at the expense of footprint
- Long running at expense of startup speed
- Rich, dynamic behaviour for mutable systems



"Kube Native" Java

- Throughput solved by scaling
- Footprint and startup speed matter
- Ephemeral, immutable systems



CRaC



MICRONAUT®





QUARKUS

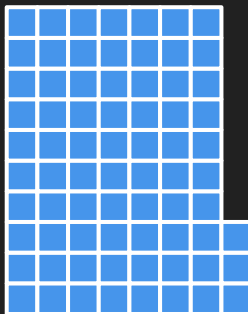
Red Hat
Developer

DevNation

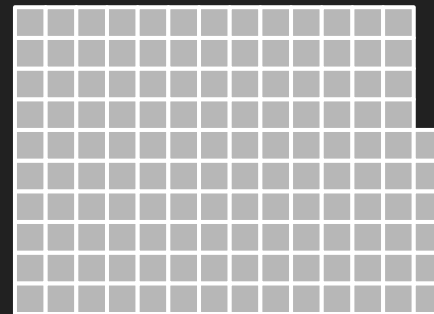
Supersonic. Subatomic. Java



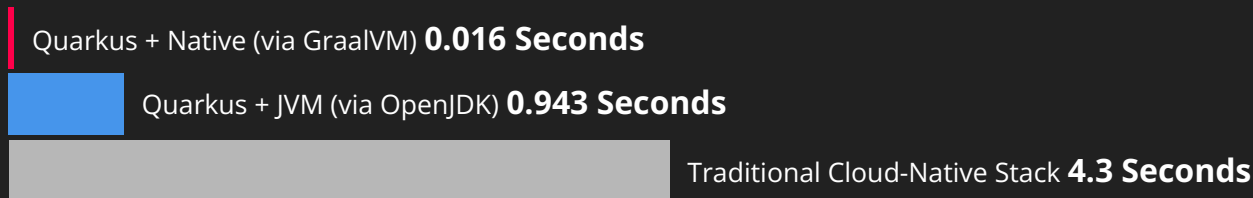
Quarkus + Native
(via GraalVM)
12 MB



Quarkus + JVM
(via OpenJDK)
73 MB



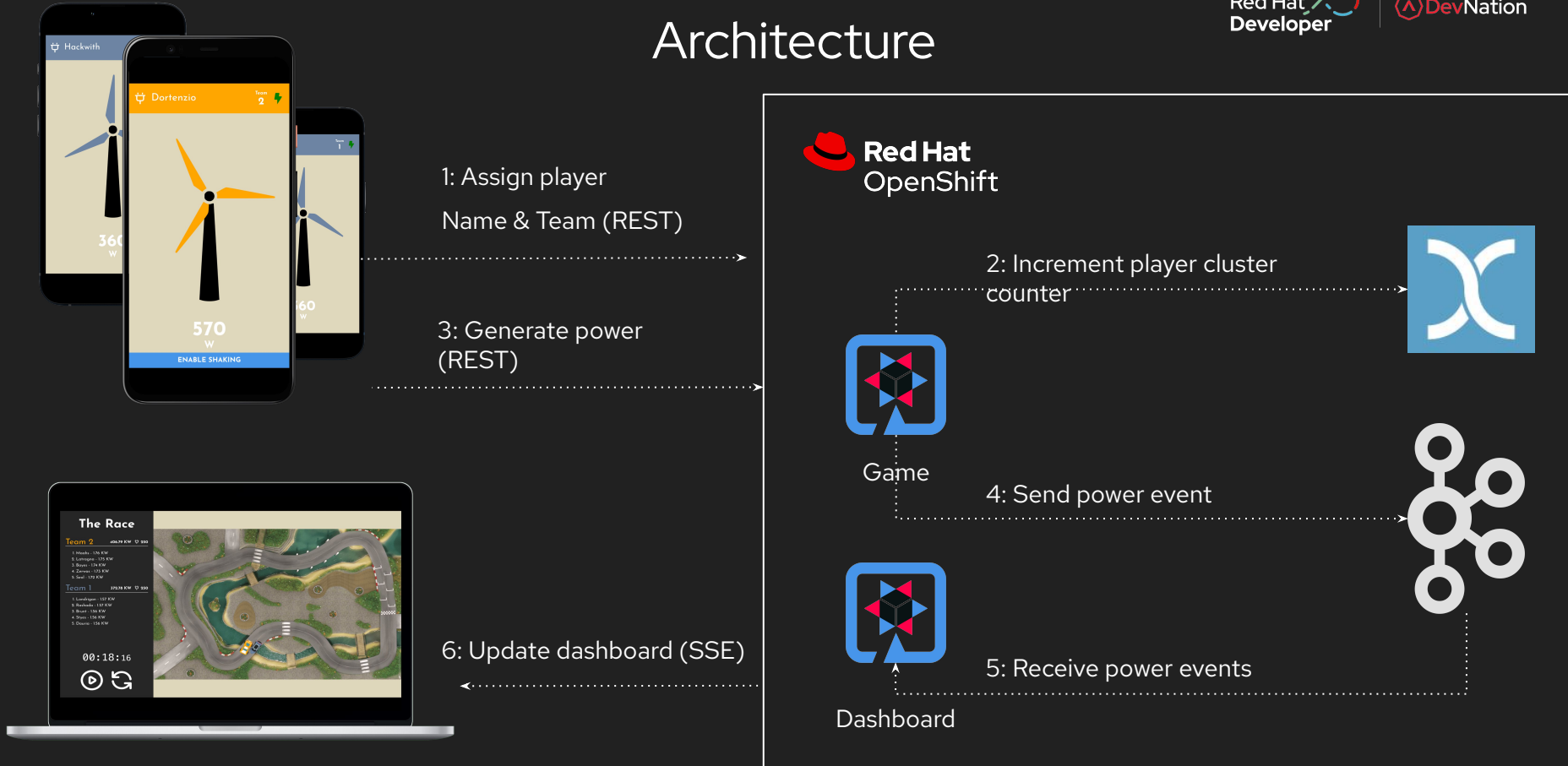
Traditional
Cloud-Native Stack
136 MB



From 0 ... To "Production-Grade"



Architecture



Architecture



3: Generate power
(REST)



Game

4: Send power event



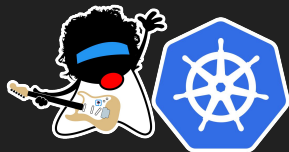




```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kubenative
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: kubenative
  template:
    spec:
      containers:
        - image: quay.io/kubekrnl/kubekrnl
          name: kubenative
          ports:
            - containerPort: 8080
              name: http
              protocol: TCP
```



```
kind: Deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: kubenative
  template:
    spec:
      containers:
        - image: quay.io/kubekrnl/kubekrnl
          name: kubenative
          ports:
            - containerPort: 8080
              name: http
              protocol: TCP
```



```
# Container image settings
quarkus.container-image.registry=quay.io
quarkus.container-image.group=kevindubois
quarkus.container-image.name=demo

quarkus.kubernetes.image-pull-policy=IfNotPresent

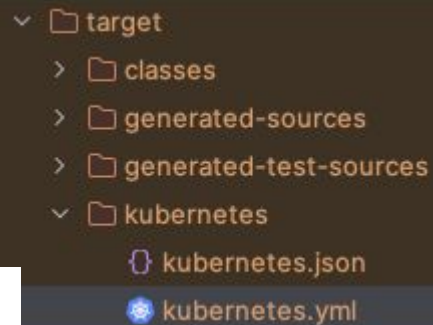
# secrets
%prod.quarkus.kubernetes-config.secrets=postgresql
%prod.quarkus.kubernetes-config.secrets.enabled=true
# configmaps
%prod.quarkus.kubernetes-config.enabled=true
%prod.quarkus.kubernetes-config.config-maps=properties

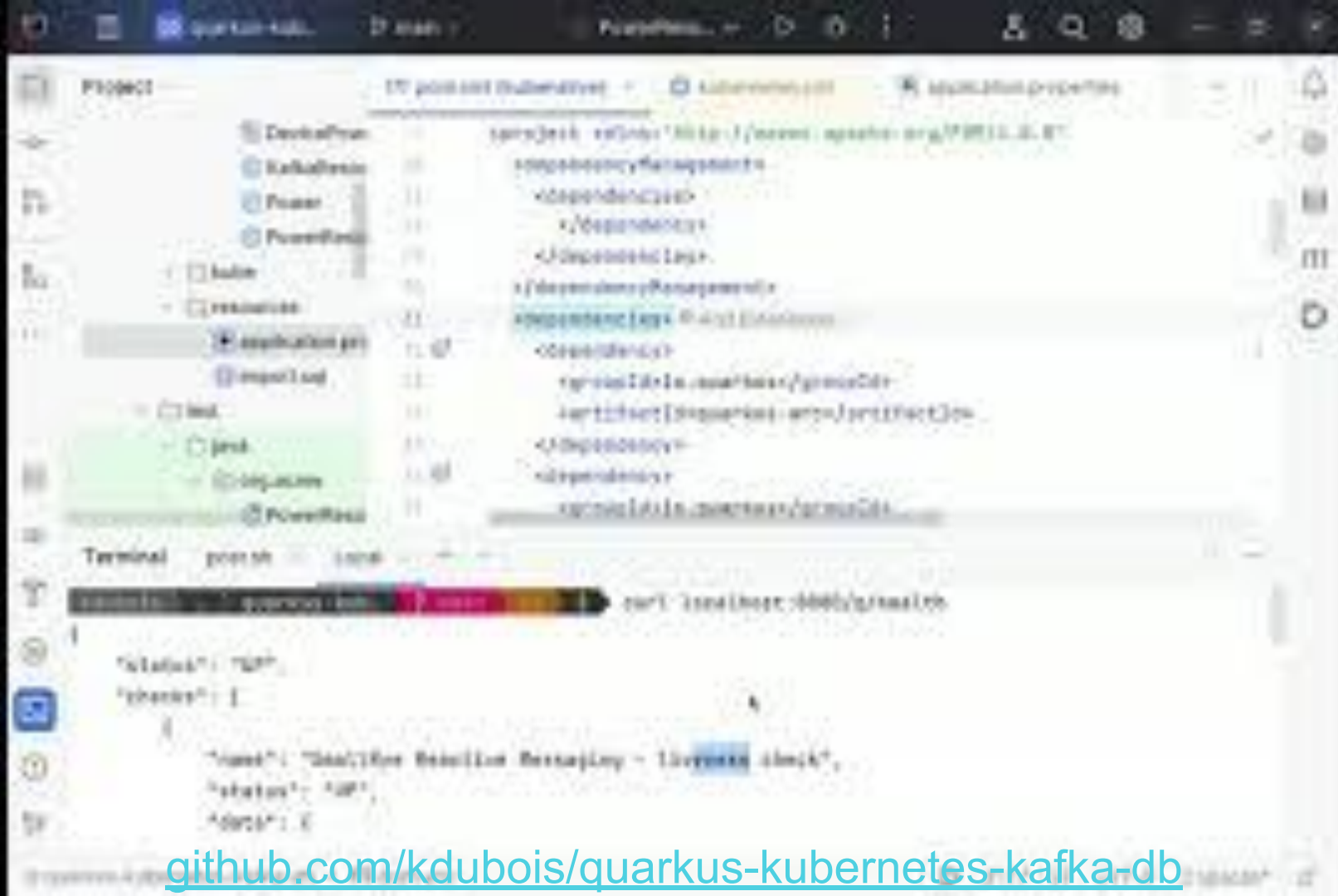
# limits
quarkus.kubernetes.resources.limits.cpu=2
quarkus.kubernetes.resources.limits.memory=400Mi

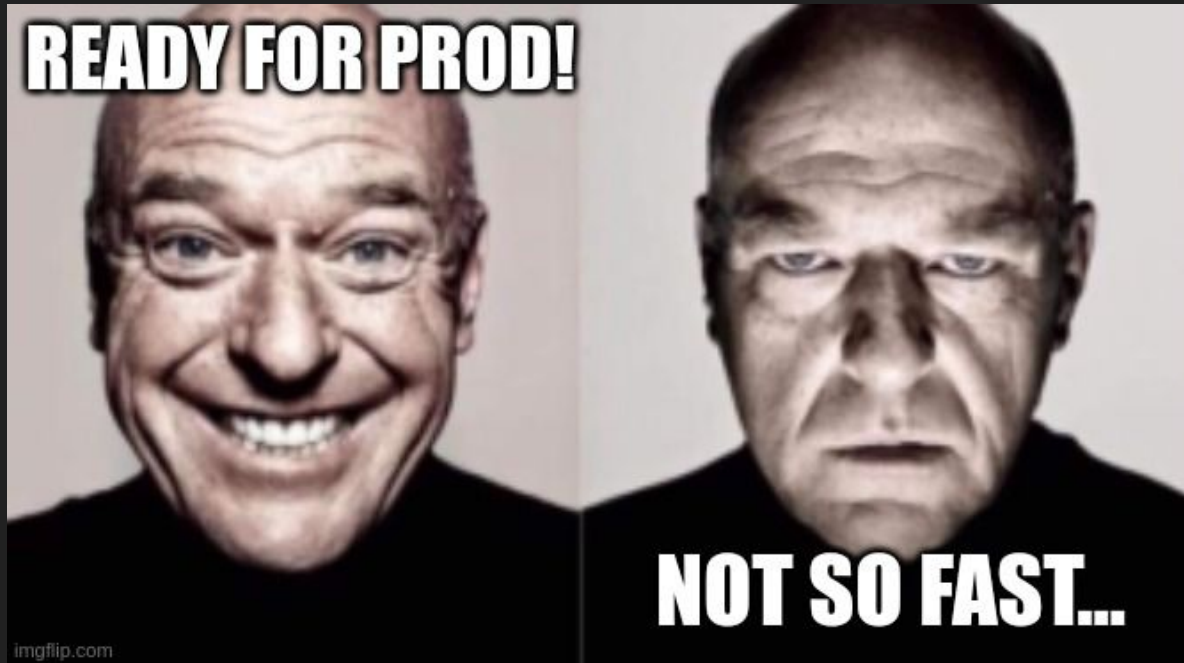
# requests
quarkus.kubernetes.resources.requests.cpu=200m
quarkus.kubernetes.resources.requests.memory=200Mi

# env variables
quarkus.kubernetes.env.vars.JDK_JAVA_OPTIONS.value=-XX:+UseParallelGC -XX:MaxRAMPercentage=75
```

mvn package







Container Running
Sends traffic
App not ready

[illegible]

Container Running
Sends traffic
App not ready

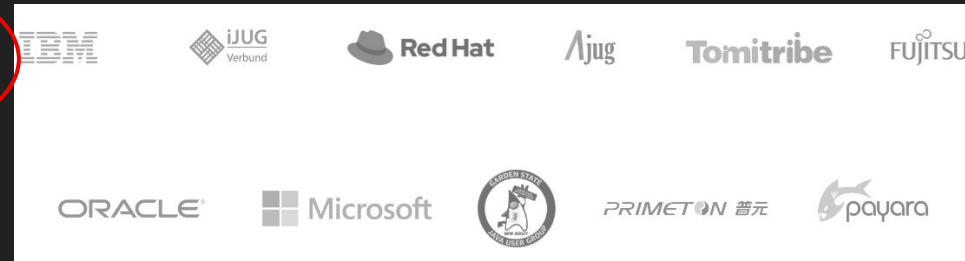
After 2.6s (Kafka, DB, Caching)
- App is ready to receive traffic

**Downtime
of 2.6s**

READY!



```
readinessProbe:
  failureThreshold: 3
  httpGet:
    path: /q/health/ready
    port: 8080
    scheme: HTTP
  initialDelaySeconds: 5
livenessProbe:
  failureThreshold: 3
  httpGet:
    path: /q/health/live
    port: 8080
    scheme: HTTP
  initialDelaySeconds: 5
```



Kubernetes Secrets & ConfigMaps



```
containers:
- name: nginx-container
  image: nginx:latest
  ports:
  - containerPort: 80
volumes:
- name: secret-volume
  secret:
    secretName: your-secret-name
```

Resource Limits



Aw, Snap!

Something went wrong while displaying this webpage.

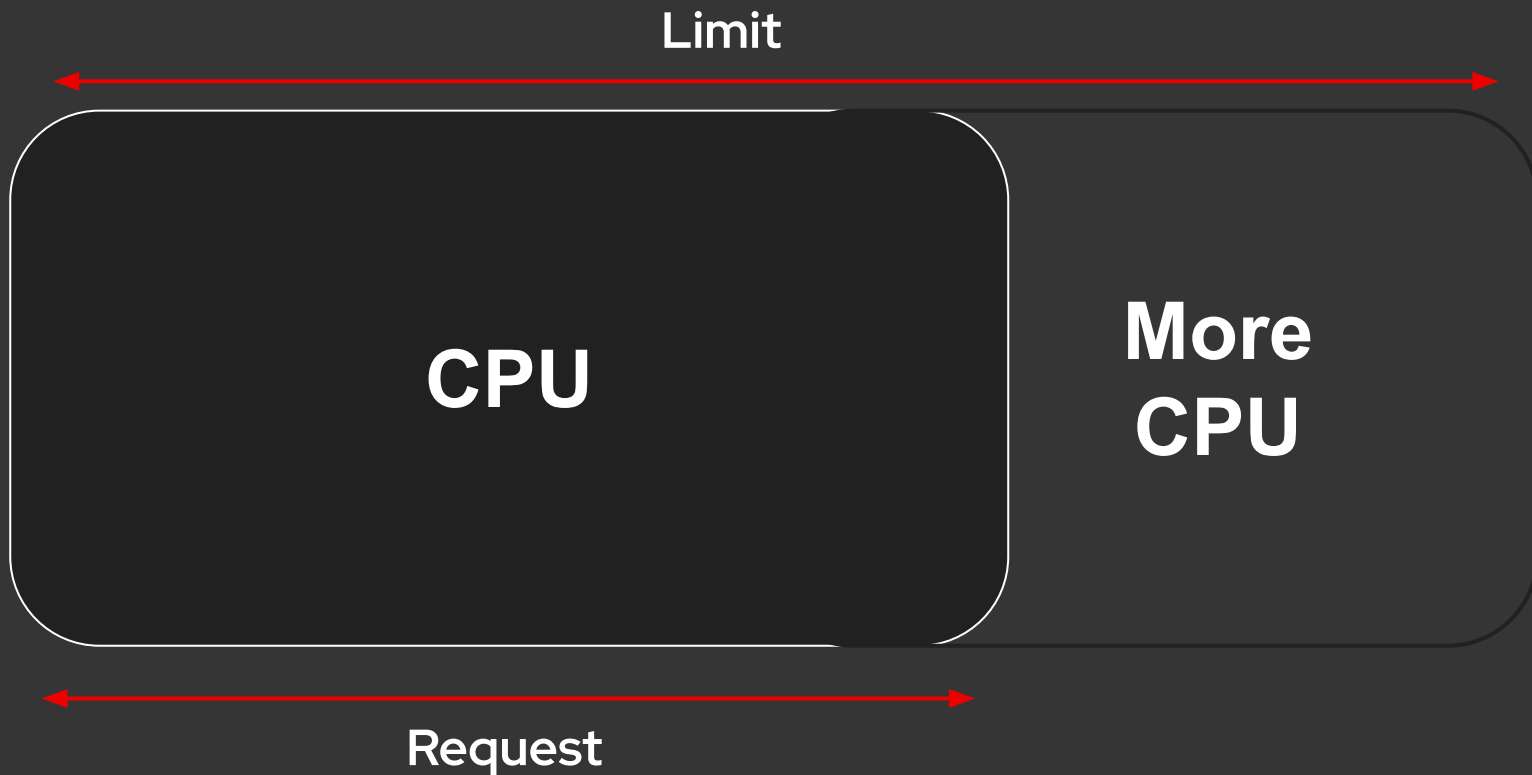
Error code: Out of Memory

[Learn more](#)

Reload

```
resources:
  requests:
    memory: "300Mi"
    cpu: "250m" # 1/4 core
  limits:
    memory: "400Mi"
    cpu: "1000m" # 1 core
```

State your intentions



Burstable

Limit

Memory

More
Memory

Request



Guaranteed

Limit

Memory



Request

Some sensible starting numbers

vCPU cores	Memory	Heap Size	GC	Replicas
2	4Gb	75%	Parallel	2

Sensible Initial Values (JVM)

```
resources:
  requests:
    memory: "4000Mi"
    cpu: "2000m"
  limits:
    memory: "4000Mi"
    cpu: "2000m"

- env:
  - name: JDK_JAVA_OPTIONS
    value: "-XX:+UseParallelGC -XX:MaxRAMPercentage=75"
```



Observability & Fault Tolerance



MICROMETER
application observability

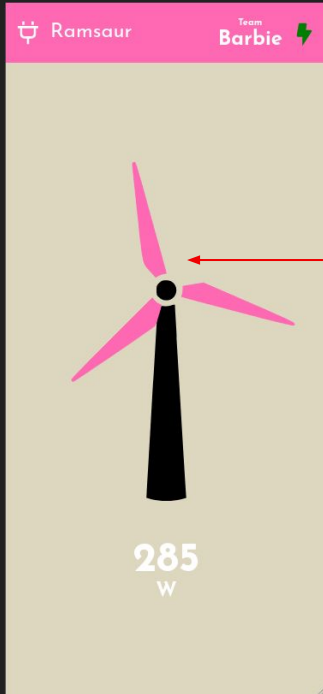
Ready for Production!



Dashboard: Green Energy

Nickname

Team



Two teams competing
(top 5 players)



Cars that
need energy

First team
wins



Scan the QR Code with your phone to play



V1

WHAT IS GITOPS?



Git Repository

Git is the single source of truth



Infrastructure as Code

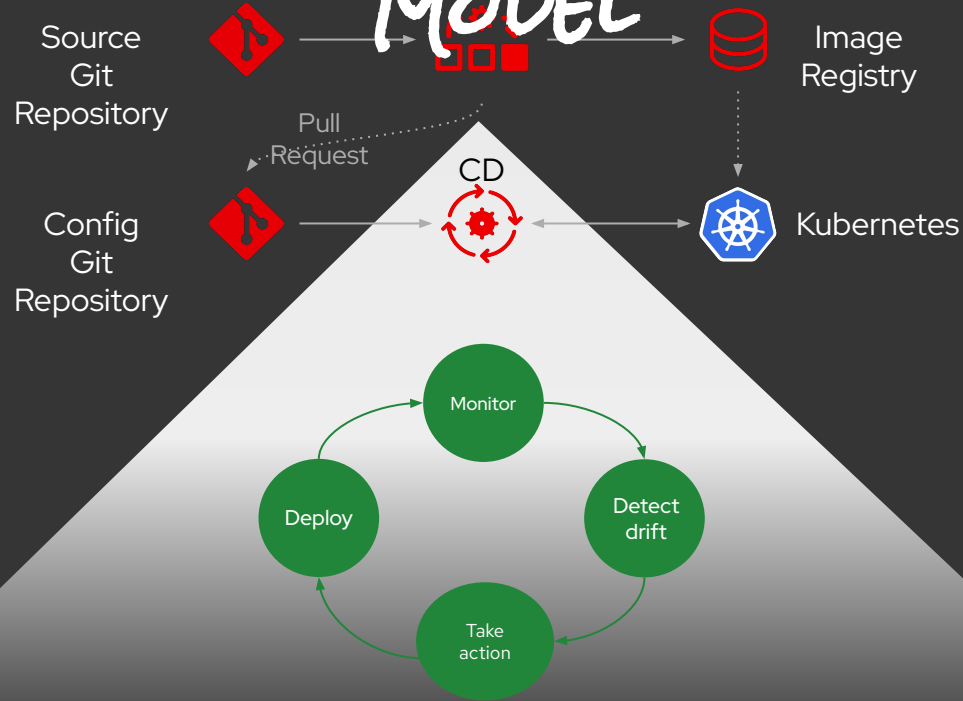
Treat everything as code



Git Commands

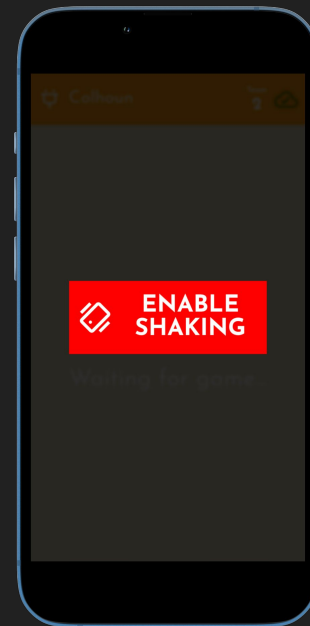
Operations through Git workflows

GitOps APPLICATION DELIVERY MODEL





Scan the QR Code with your phone to play



V2

github.com/kdubois/quarkus-kubernetes-kafka-db

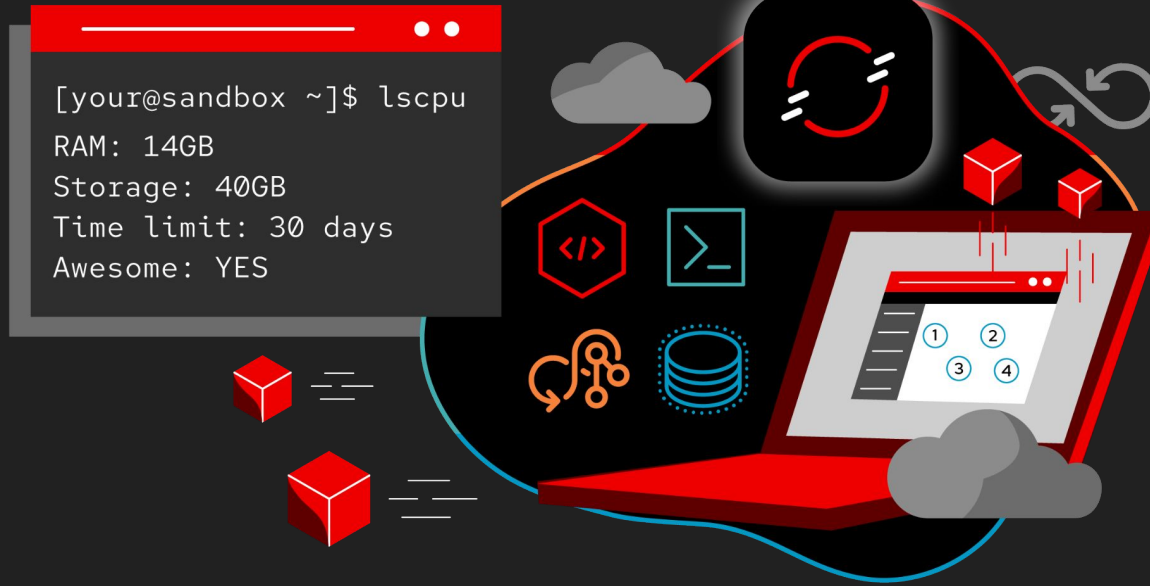
github.com/redhat-developer-demos/quinoa-wind-turbine



Free Developer e-Books & Tutorials!

developers.redhat.com/eventtutorials





developers.redhat.com/developer-sandbox

Learn containers, Kubernetes, and OpenShift in your browser.

Start exploring in the OpenShift Sandbox.



Try Red Hat's products and technologies
without setup or configuration.



Red Hat

Red Hat
Developer



@kevindubois



youtube.com/@thekevindubois



linkedin.com/in/kevindubois



github.com/kdubois

Red Hat
Developer

DevNation



slides



Thank you!