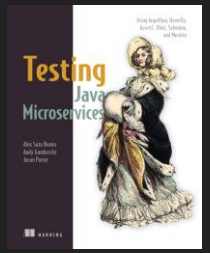
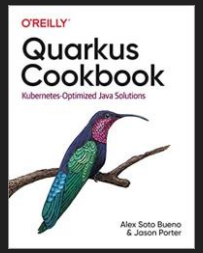

DevNation
Kubernetes II Deep Dive
Homework dn.dev/kube-tutorial
Red Hat
Developer
dn.dev/kubemaster2

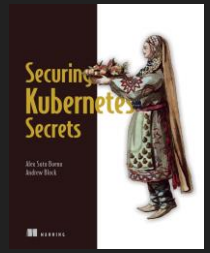
Red Hat Developer | DevNation

Alex Soto (lordofthejars.com)

- @alexsotob
- asotobue@redhat.com
- Currently Red Hat's Director of Developer Experience
- Featured speaker at technology events around the globe
- A Java Champion since 2017
- Writer, University Professor, Radio collaborator
- A big fan of testing and continuous delivery in 21st century







dn.dev/kubemaster2

Agenda

Red Hat Developer | DevNation

Part I

- Why Kubernetes
- What is Kubernetes
- Installation
- kubectl
- Pod
- ReplicaSet
- Deployment
- logs, stern
- Apps as a Service

Part II

- Building Images
- Resource Limits
- Rolling Updates
- Liveness & Readiness
- Env & ConfigMap

Red Hat Developer | DevNation

Quick Recap

Kubernetes Deep Dive I

dn.dev/kubemaster2

DevOps Challenges for Multiple Containers

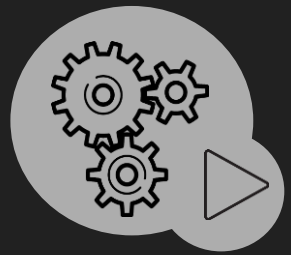
Red Hat Developer | DevNation

- How to scale?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
- How to keep them running?
- How to update them?
- Rebuild Container Images?

Kubernetes Terms

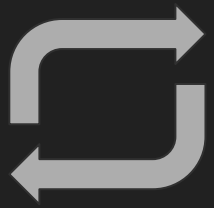
Red Hat Developer | DevNation

Pod



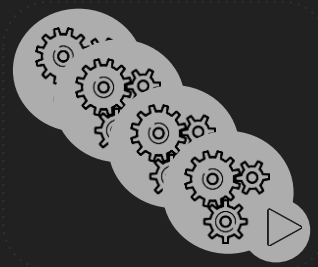
- ✓ 1+ containers
- ✓ Shared IP
- ✓ Shared storage (ephemeral)
- ✓ Shared resources
- ✓ Shared lifecycle

Replicaset/ Deployment



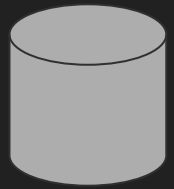
- ✓ The Desired State - replicas, pod template: health checks, resources, image

Service



- ✓ Grouping of pods (acting as one) has stable virtual IP and DNS name

Persistent Volume



- ✓ Network available storage
- ✓ PVs and PVCs

Label

- ✓ Key/Value pairs associated with Kubernetes objects (env=production)

dn.dev/kubemaster2


Red Hat
Developer
DevNation
Building Images
dn.dev/kubemaster2

Running an image on Kubernetes

Red Hat Developer | DevNation

1. Find a base Image: Docker Hub, quay.io, gcr.io, access.redhat.com/containers
2. Craft your `Dockerfile`
3. Build your Image: `docker build -t mystuff/myimage:v1`
    a. If remote, do "docker push"
4. `kubectl apply -f myDeployment.yml`
5. `kubectl apply -f myService.yml`
6. Expose a URL via your Kubernetes distribution's load-balancer

Image Builders

Red Hat Developer | DevNation

Options Include:

A. **docker build** then kubectl run or kubectl create -f deploy.yml
~~B. Fabric8 maven plugin (fabric8.io)~~ (Eclipse JKube)
C. Jib - Maven/Gradle plugin
D. s2i - source to image
~~E. Knative Build~~ - source to image, source to url (Tekton.dev)
F. Instead of docker...
    a. Red Hat's podman/buildah, Google's kaniko, Uber's makisu
G. Buildpacks - similar to Heroku & Cloud Foundry

Red Hat Developer | DevNation

Building Exercises

View exercises

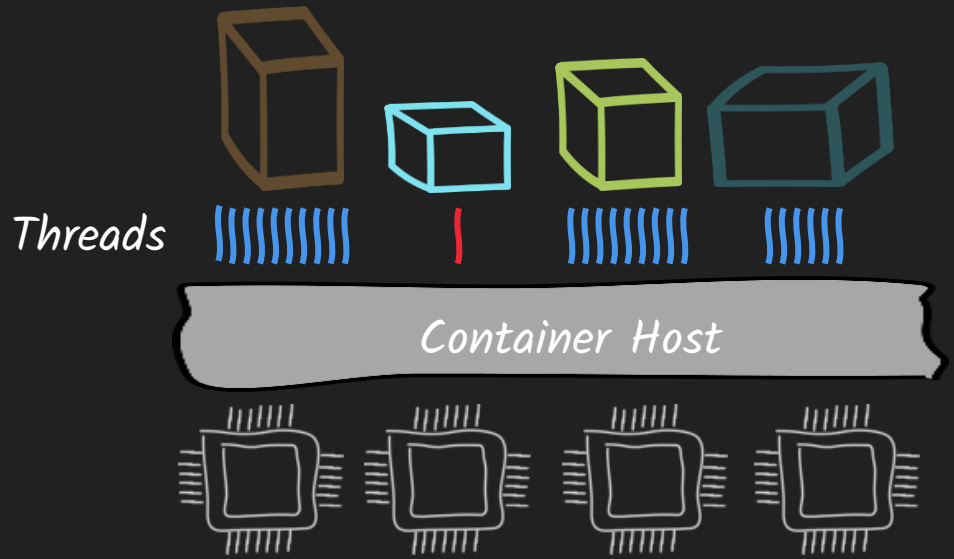
dn.dev/kubemaster2


Red Hat
Developer
DevNation
Resource Limits
dn.dev/kubemaster2

The hidden truth of containers

Red Hat Developer | DevNation

Containers are about **sharing** and **deployment density**



costs memory

cost lots of memory

cost CPU cycles

\+ quotas = BOOM

Resource Requests/Limits

Red Hat Developer | DevNation

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

Red Hat Developer | DevNation

Resource Limits Exercises

View exercises

dn.dev/kubemaster2


Red Hat
Developer
DevNation
Rolling Updates
dn.dev/kubemaster2

"Rolling updates allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones."

https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/

Red Hat Developer | DevNation

Rolling Update Exercises

View exercises

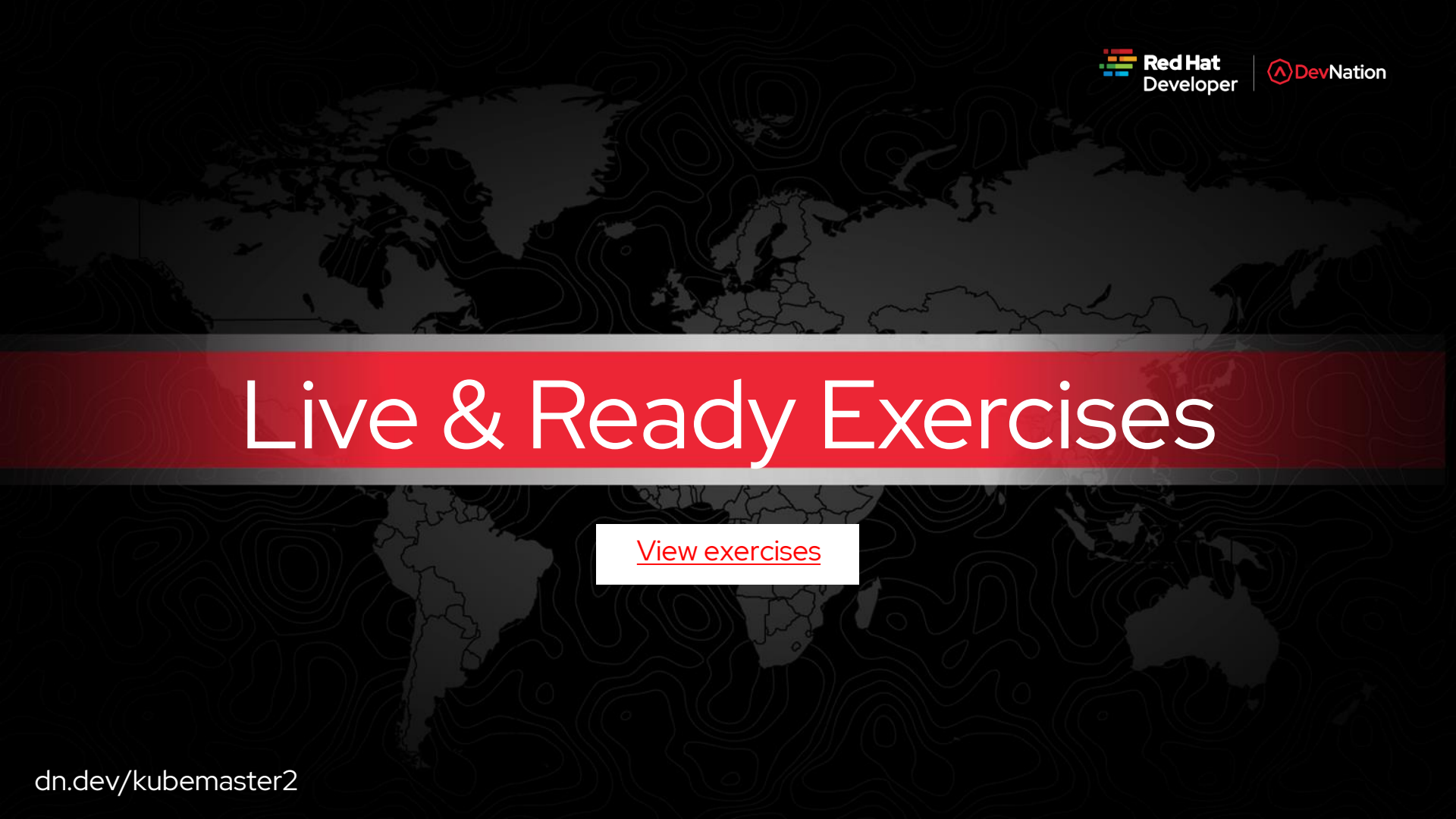
dn.dev/kubemaster2


Red Hat
Developer
DevNation
Liveness & Readiness
dn.dev/kubemaster2

“The kubelet uses liveness probes to know when to restart a container.”

“The kubelet uses readiness probes to know when a container is ready to start accepting traffic.”

https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/


Red Hat Developer
DevNation
Live & Ready Exercises
View exercises
dn.dev/kubemaster2


Red Hat
Developer
DevNation
Environment & ConfigMap
dn.dev/kubemaster2

env vars & configmaps

Red Hat Developer | DevNation

An app's config is everything that is likely to vary between deploys (staging, production, developer environments, etc). *12 Factor Apps*

```
kubectl set env deployment/myboot DBCONN="jdbc:sqlserver://45.91.12.123:1443;user=MyUserName;password=*****;"

kubectl create cm my-config --from-env-file=config/some.properties
```

ConfigMaps

Red Hat Developer | DevNation

```
kubectl create cm my-config --from-env-file=config/some.properties
```

some.properties

```
GREETING=jambo
LOVE=Amour
```

Partial deployment.yml

```
spec:
 containers:
 - name: myboot
   image: 9stepsawesome/myboot:v1
   ports:
     - containerPort: 8080
   envFrom:
   - configMapRef:
       name: my-config
```

Partial .java

```
@Autowired

private Environment environment;

String greeting =
environment.getProperty("GREETING","Default");

String love =
environment.getProperty("LOVE","Default");
```

dn.dev/kubemaster2

Red Hat Developer | DevNation

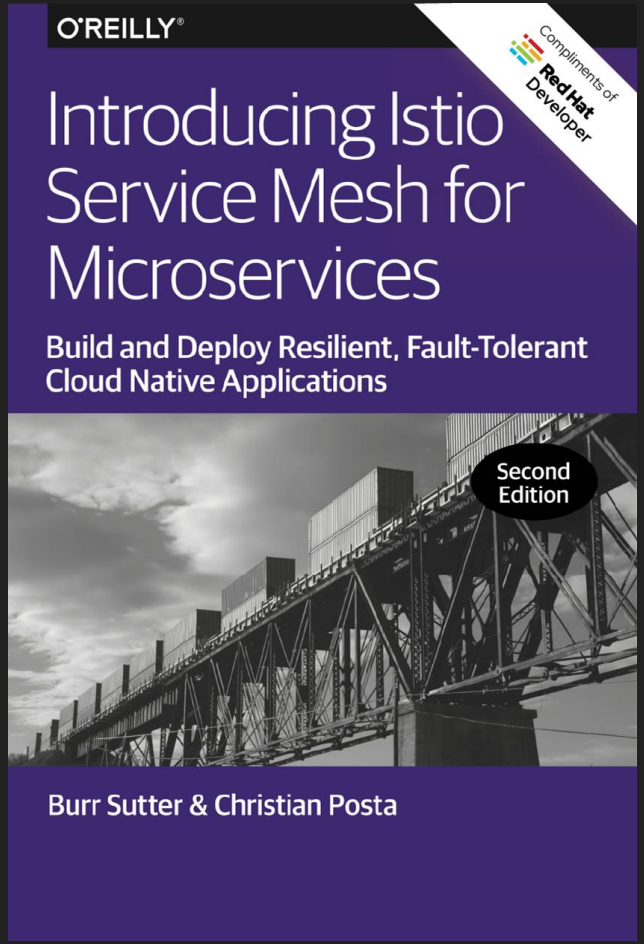
Exercises

External Configuration

Secrets exercise

dn.dev/kubemaster2


Red Hat
Developer
DevNation
Free Resources
dn.dev/kubemaster2

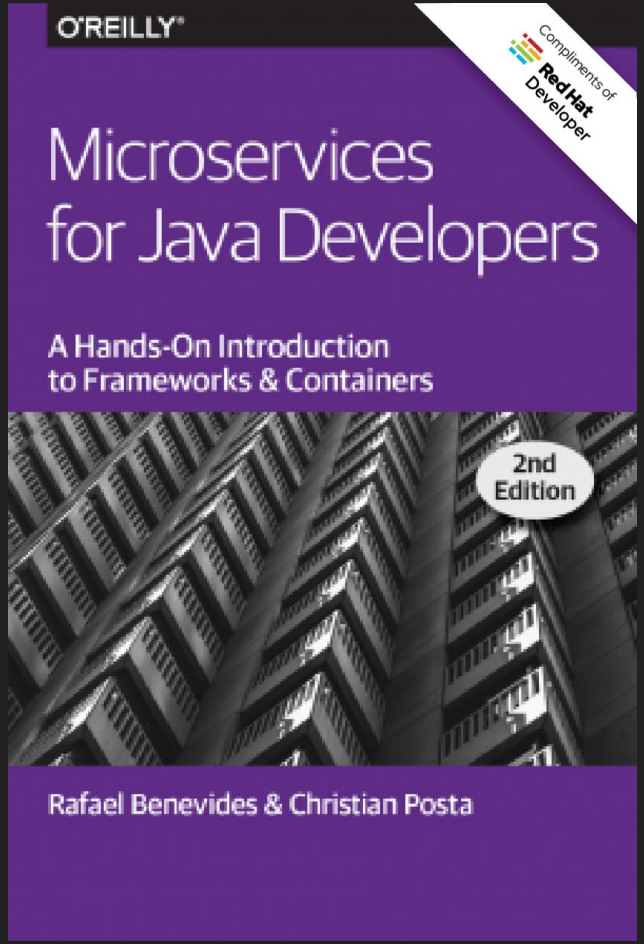


Red Hat Developer | DevNation

Download

bit.ly/istiobook

dn.dev/kubemaster2

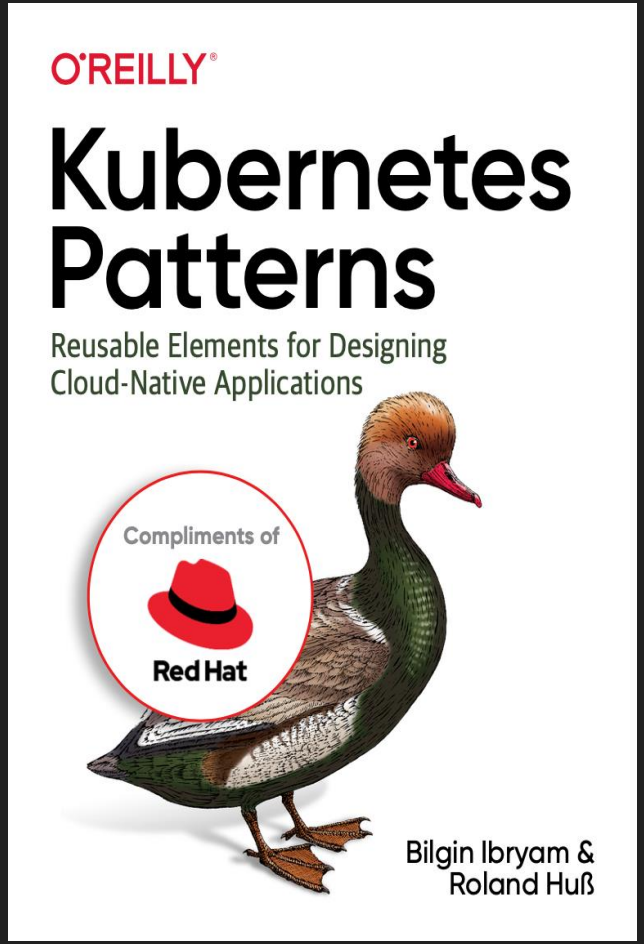


Red Hat Developer | DevNation

Download

bit.ly/javamicroservicesbook

dn.dev/kubemaster2

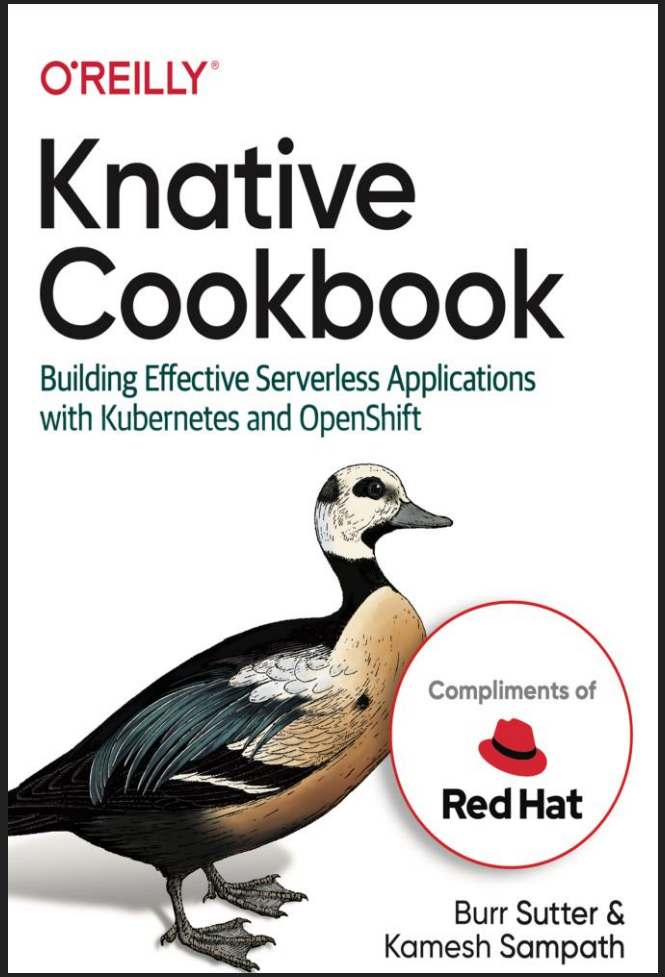


Red Hat Developer | DevNation

Download

dn.dev/k8spatterns1

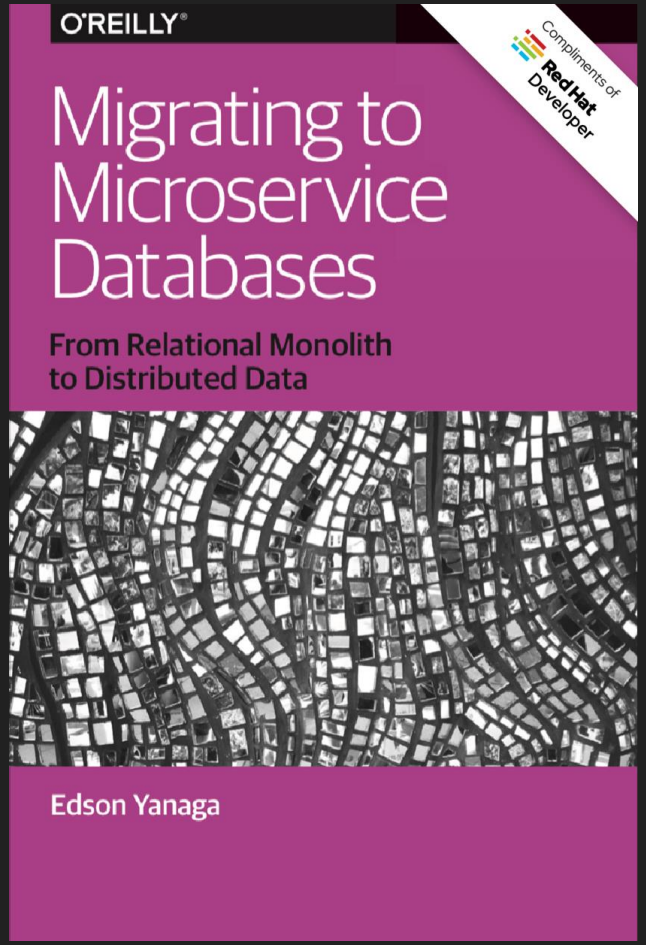
dn.dev/kubemaster2



Red Hat Developer | DevNation

Download

dn.dev/knative-cookbook

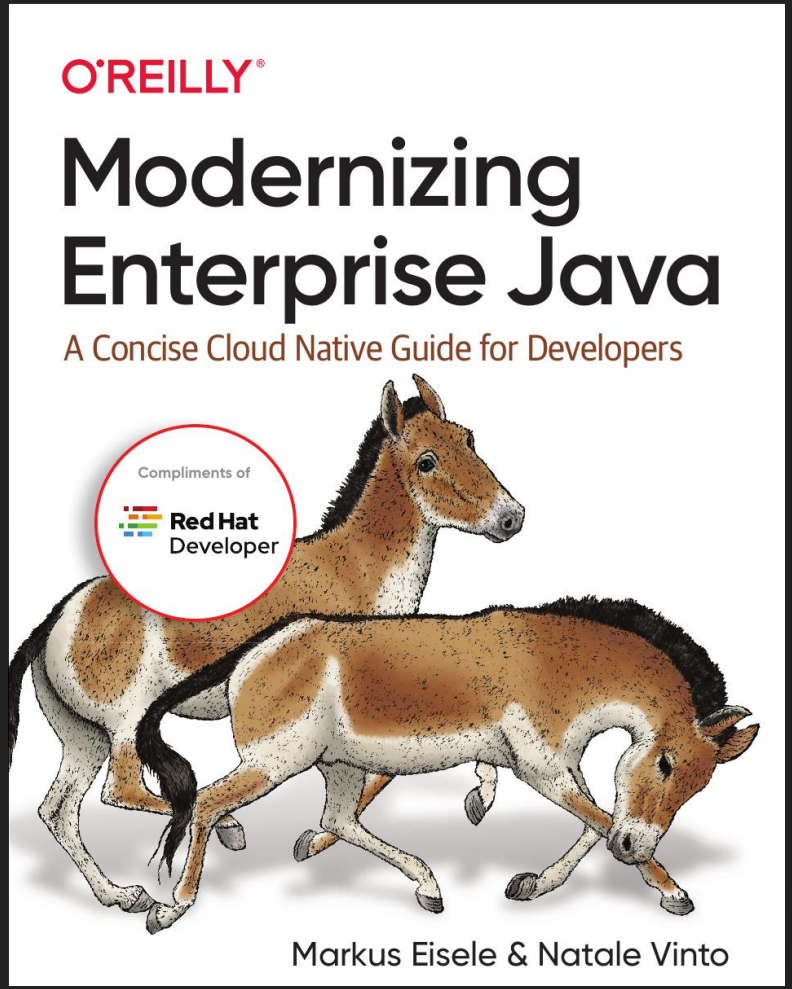


Red Hat Developer | DevNation

Download

bit.ly/mono2microdb

dn.dev/kubemaster2

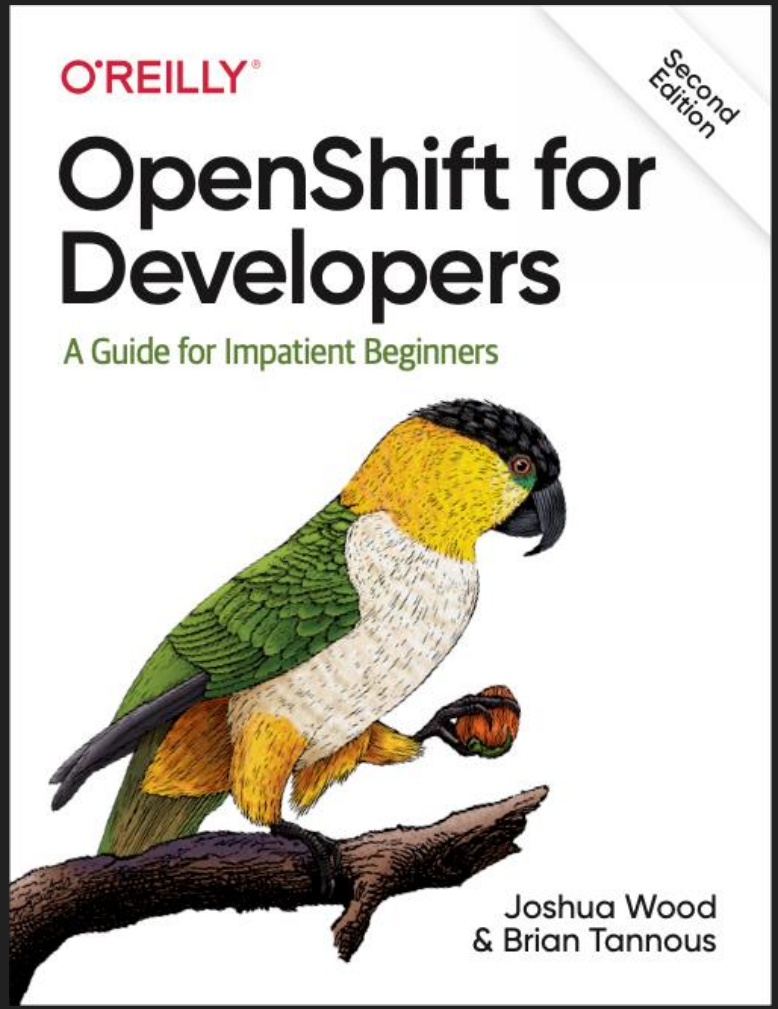


Red Hat Developer | DevNation

Download

https://red.ht/modernize-enterprise-java

dn.dev/kubemaster2



Red Hat Developer | DevNation

Download

https://red.ht/3IxJCzY

dn.dev/kubemaster2


Red Hat
Developer
DevNation
The End
dn.dev/kubemaster2