How to speed up development productivity with OpenShift

Magnus Eklund

Solution Architect AppDev

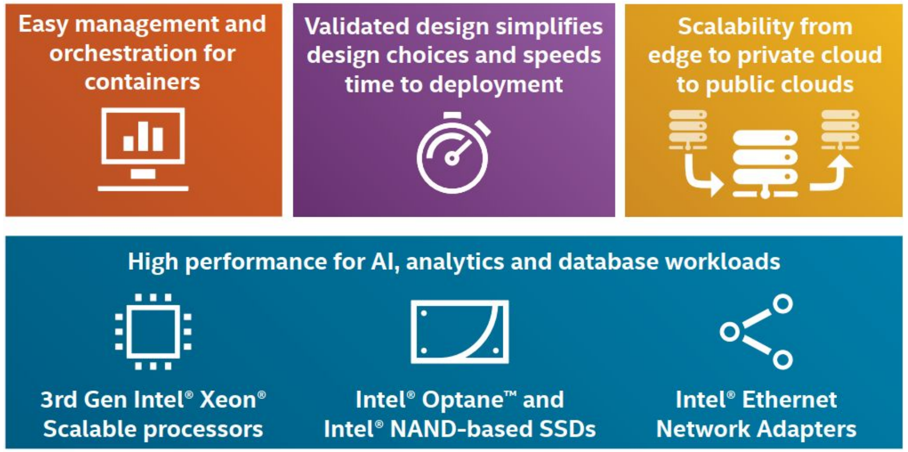
Red Hat OpenShift Reference Architecture

Joint Red Hat and Intel OpenShift Reference Architecture

Solution overview

Summary: The RA enables deployment of performant and low-latency container-based workloads onto different footprints, such as bare metal, virtual, private cloud, public cloud, or a combination of these, in either a centralized data center or at the edge

Purpose: A general purpose OpenShift reference architecture to showcase the best of Intel and Red Hat products with key workloads


Easy management and orchestration for containers

Validated design simplifies design choices and speeds time to deployment

Scalability from edge to private cloud to public clouds

High performance for AI, analytics and database workloads

3rd Gen Intel® Xeon® Scalable processors

Intel® Optane™ and Intel® NAND-based SSDs

Intel® Ethernet Network Adapters

Solution ecosystem

ISVs

- 🟩 IBM Cloud Pak for Data
- 🟩 No SQL DBs – OPEN DATA HUB
- ⬜ splunk>

OEMs / CSPs

- 🟩 DELL, Hewlett Packard Enterprise, Lenovo, CISCO, SUPERMICRO
- 🟩 aws, Microsoft Azure, IBM Cloud

SIs

- 🟩 World Wide Technology
- 🟨 accenture

Key 🟩 Complete 🟨 In Progress ⬜ In Plan

Intel enabling status

- Intel® Xeon (2nd Gen – Cascade Lake, 3rd Gen – Ice Lake)
- Intel Optane (PMEM, SSD); Columbiaville

Collateral

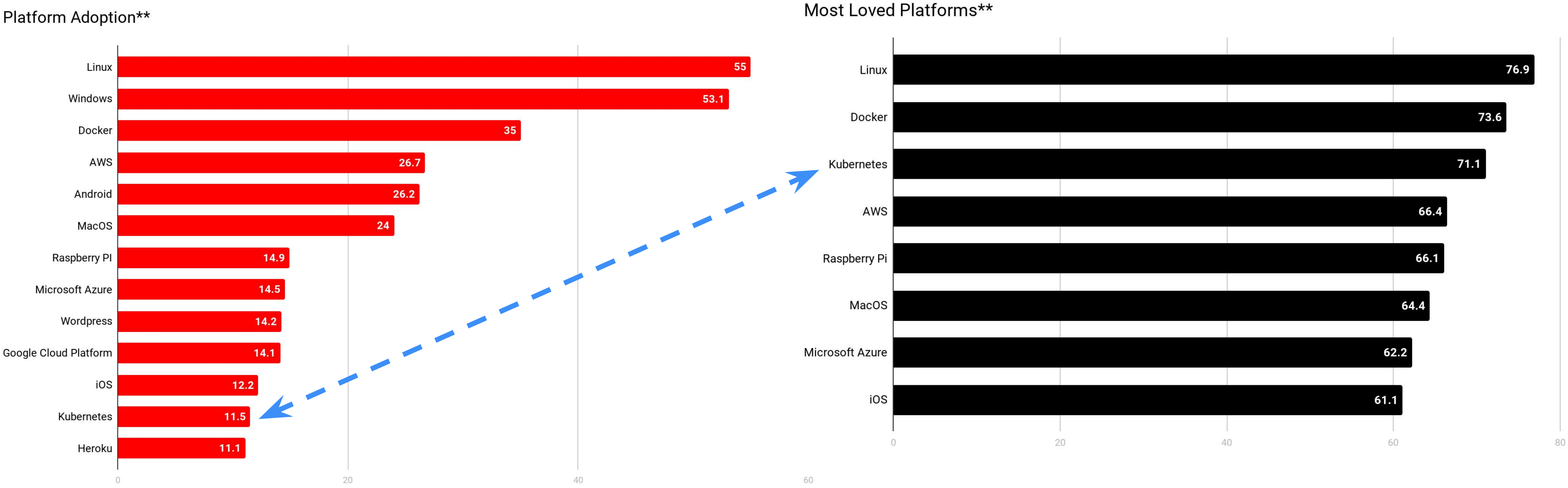
- Intel OpenShift RA for 4.6
- Intel OpenShift Solution Brief for 4.6
- Red Hat: OpenShift Ref Arch – Multiple OEMs
- Dell: OpenShift Offering
- HPE: OpenShift Offering
- Cisco: OpenShift Offering
- Lenovo: OpenShift Offering
- Supermicro: OpenShift Offering
- Penguin Computing: OpenShift Offering

Agenda

- Kubernetes solves all your problems!?
- Spend more time developing
- Easily navigate the many components of your applications.
- Get advanced services, without the hassle.
- Recap

Kubernetes solves all your problems!?

Kubernetes and Developers



Kubernetes tried by most businesses

** Source: Stackoverflow Developer Survey 2020 , 65000 developers.

Complexities preventing adoption...

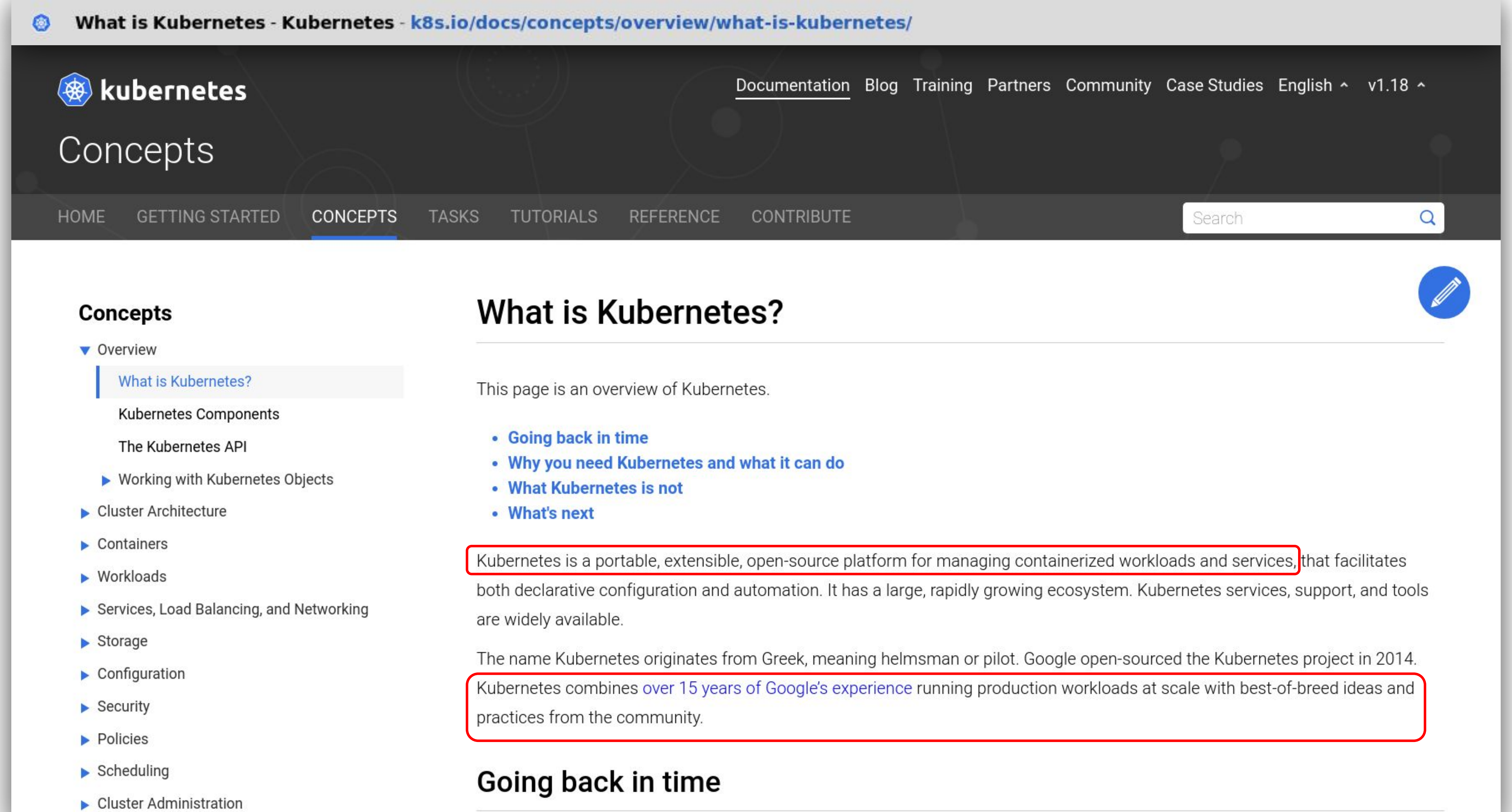
Ask a Developer

"What do you think about when you hear the word **Kubernetes** ?

- Containers
- YAML
- Virtualization
- Infrastructure
- Abstraction
- Complexity

DIY K8s strategy - maintenance problems

Feeling Lucky?



What is Kubernetes?

This page is an overview of Kubernetes.

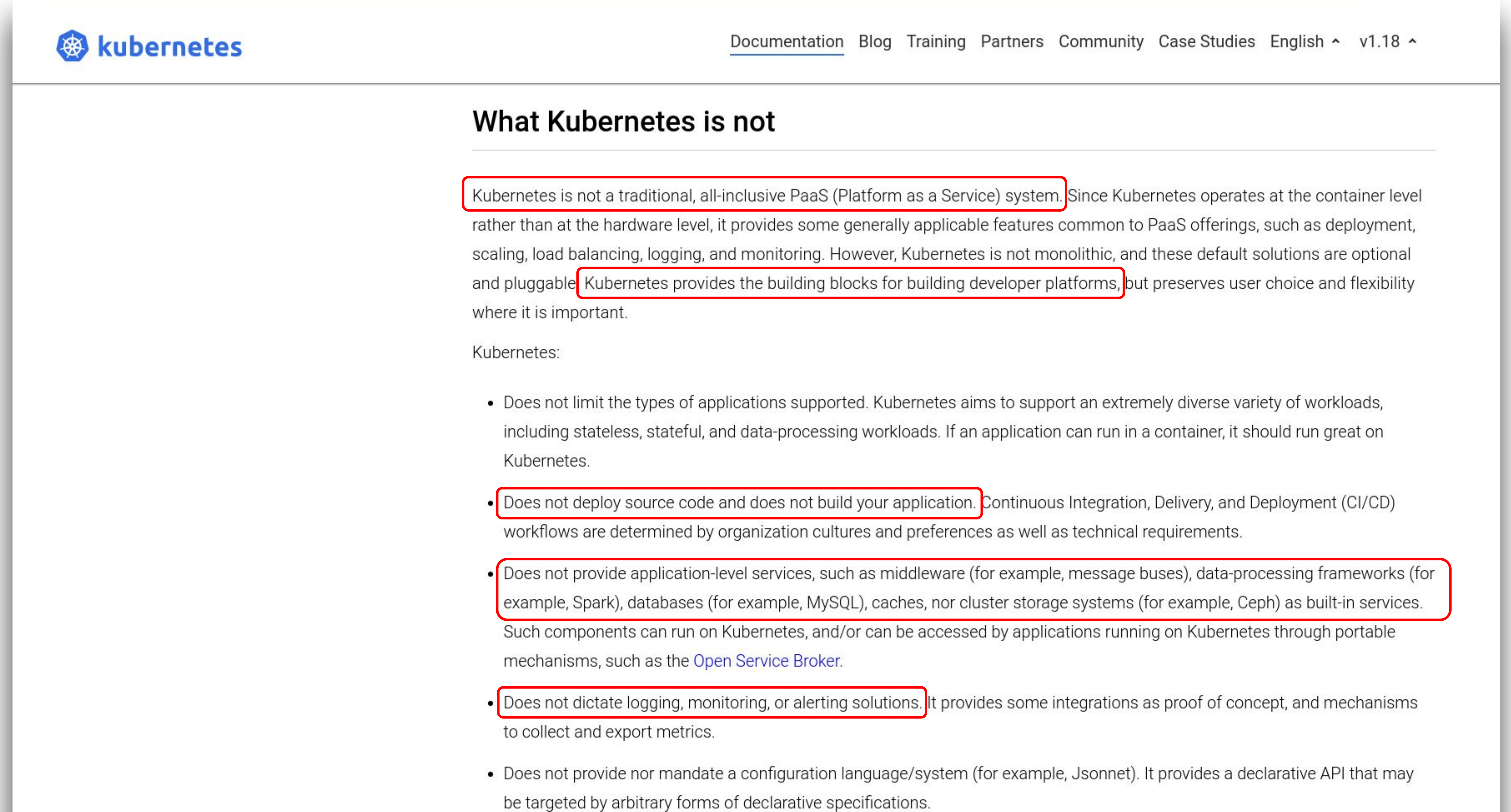
- Going back in time
- Why you need Kubernetes and what it can do
- What Kubernetes is not
- What's next

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

The name Kubernetes originates from Greek, meaning helmsman or pilot. Google open-sourced the Kubernetes project in 2014. Kubernetes combines over 15 years of Google's experience running production workloads at scale with best-of-breed ideas and practices from the community.

Going back in time

Out of Scope



What Kubernetes is not

Kubernetes is not a traditional, all-inclusive PaaS (Platform as a Service) system. Since Kubernetes operates at the container level rather than at the hardware level, it provides some generally applicable features common to PaaS offerings, such as deployment, scaling, load balancing, logging, and monitoring. However, Kubernetes is not monolithic, and these default solutions are optional and pluggable. Kubernetes provides the building blocks for building developer platforms, but preserves user choice and flexibility where it is important.

Kubernetes:

- Does not limit the types of applications supported. Kubernetes aims to support an extremely diverse variety of workloads, including stateless, stateful, and data-processing workloads. If an application can run in a container, it should run great on Kubernetes.
- Does not deploy source code and does not build your application. Continuous Integration, Delivery, and Deployment (CI/CD) workflows are determined by organization cultures and preferences as well as technical requirements.
- Does not provide application-level services, such as middleware (for example, message buses), data-processing frameworks (for example, Spark), databases (for example, MySQL), caches, nor cluster storage systems (for example, Ceph) as built-in services. Such components can run on Kubernetes, and/or can be accessed by applications running on Kubernetes through portable mechanisms, such as the Open Service Broker.
- Does not dictate logging, monitoring, or alerting solutions. It provides some integrations as proof of concept, and mechanisms to collect and export metrics.
- Does not provide nor mandate a configuration language/system (for example, Jsonnet). It provides a declarative API that may be targeted by arbitrary forms of declarative specifications.

Complexities preventing adoption...

Ask a Developer

"What do you think about when you hear the word **Kubernetes** ?

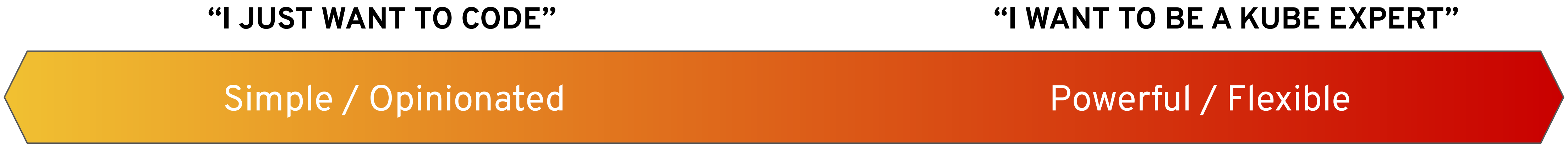
- Containers
- YAML
- Virtualization
- Infrastructure
- Abstraction
- Complexity

What do you really want from **Kubernetes** ?

- Infrastructure as Code
- Innovation & Cloud Native Tools
- Flexibility
- Extensibility
- Portability
- Community

Not much focus and support yet

Developer interest



“I JUST WANT TO CODE”

“I WANT TO BE A KUBE EXPERT”

Simple / Opinionated

Powerful / Flexible

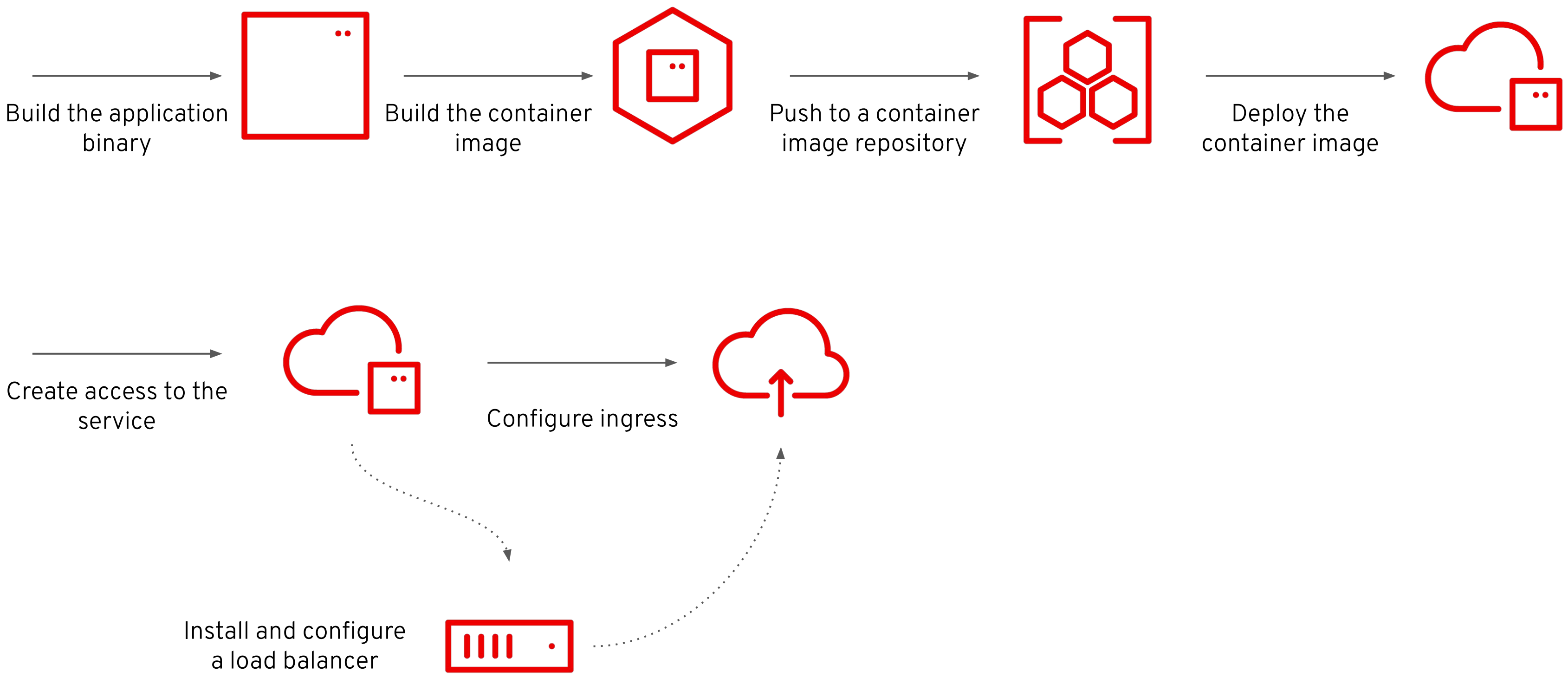
...but all developers want:

- ➔ Choice of tools, not “corporate dictated” tools
- ➔ Ability to consume cloud services and deploy anywhere
- ➔ Kubernetes...but easier
- ➔ To be able to switch contexts instantly

How to easily deploy things on OpenShift

Deploy an application on Kubernetes

Common process



Build the application binary

Build the container image

Push to a container image repository

Deploy the container image

Create access to the service

Configure ingress

Install and configure a load balancer

https://medium.com/swlh/im-so-sorry-openshift-i-ve-taken-you-for-granted-f36fb47ea4d9

S2I demo

oc new-app demo

What happened? Source-to-Image explained

- Created a build pod for building the app
- Created an OpenShift Build config
- Pulled the builder image into OpenShift’s internal docker registry.
- Cloned the application repo locally
- Analysis showed there’s a maven pom, so compiled the application using maven
- Created a new container image with the compiled java application and pushed this container image into the internal container registry
- Created a Kubernetes Deployment with specifications around pods, service spec etc.
- Kicked off a deploy of the container image.
- Removed the build pod.
- Exposed the service in the route (ingress)

Red Hat

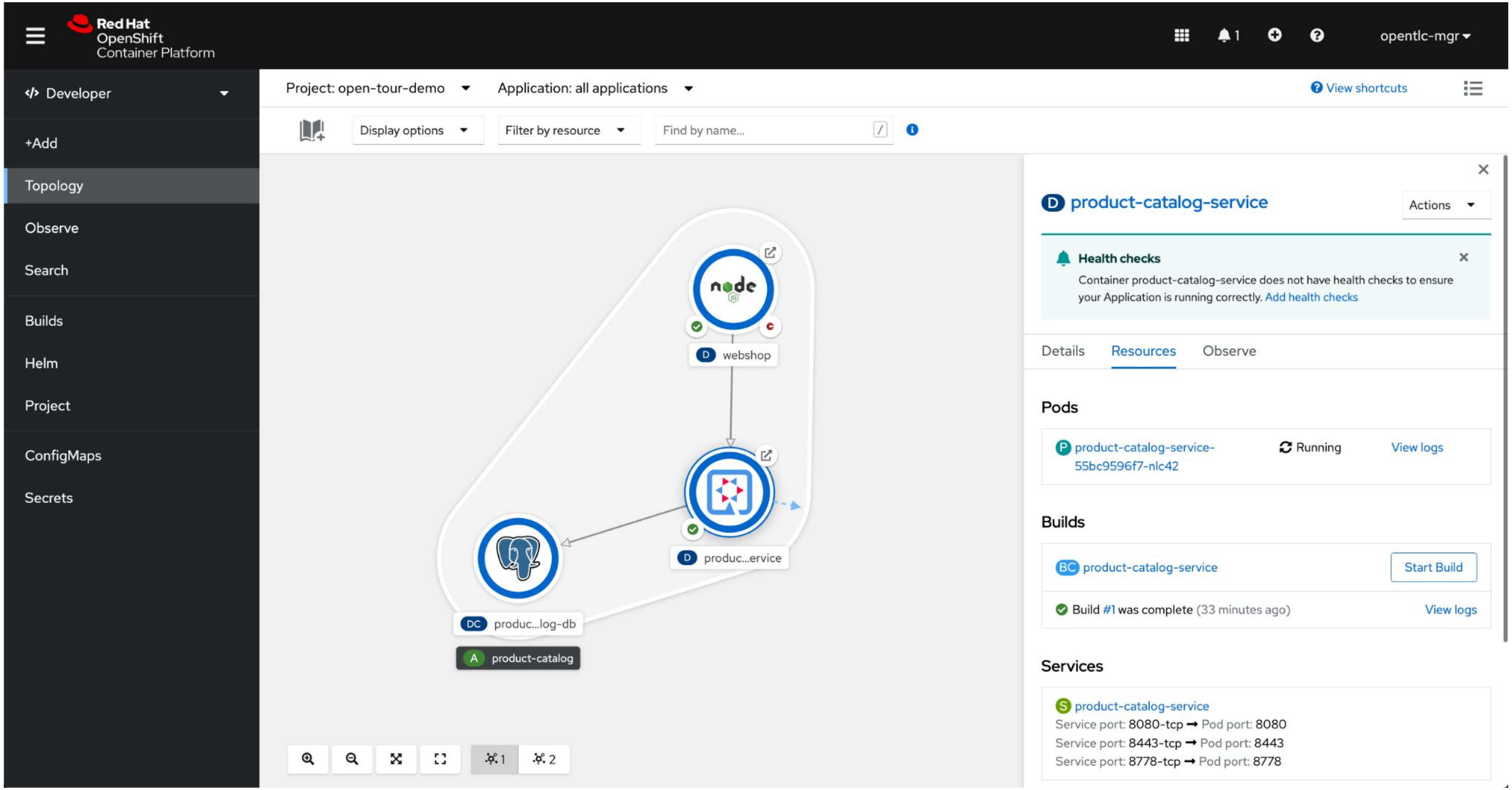
Recap -
how to easily deploy things on OpenShift
So *that you can spend more time developing*

Red Hat

How to use the Developer Perspective

Red Hat

Topology view



D product-catalog-service

Health checks

Container product-catalog-service does not have health checks to ensure your Application is running correctly. Add health checks

Pods

P product-catalog-service-55bc9596f7-nlc42 Running View logs

Builds

BC product-catalog-service Start Build

Build #1 was complete (33 minutes ago) View logs

Services

S product-catalog-service

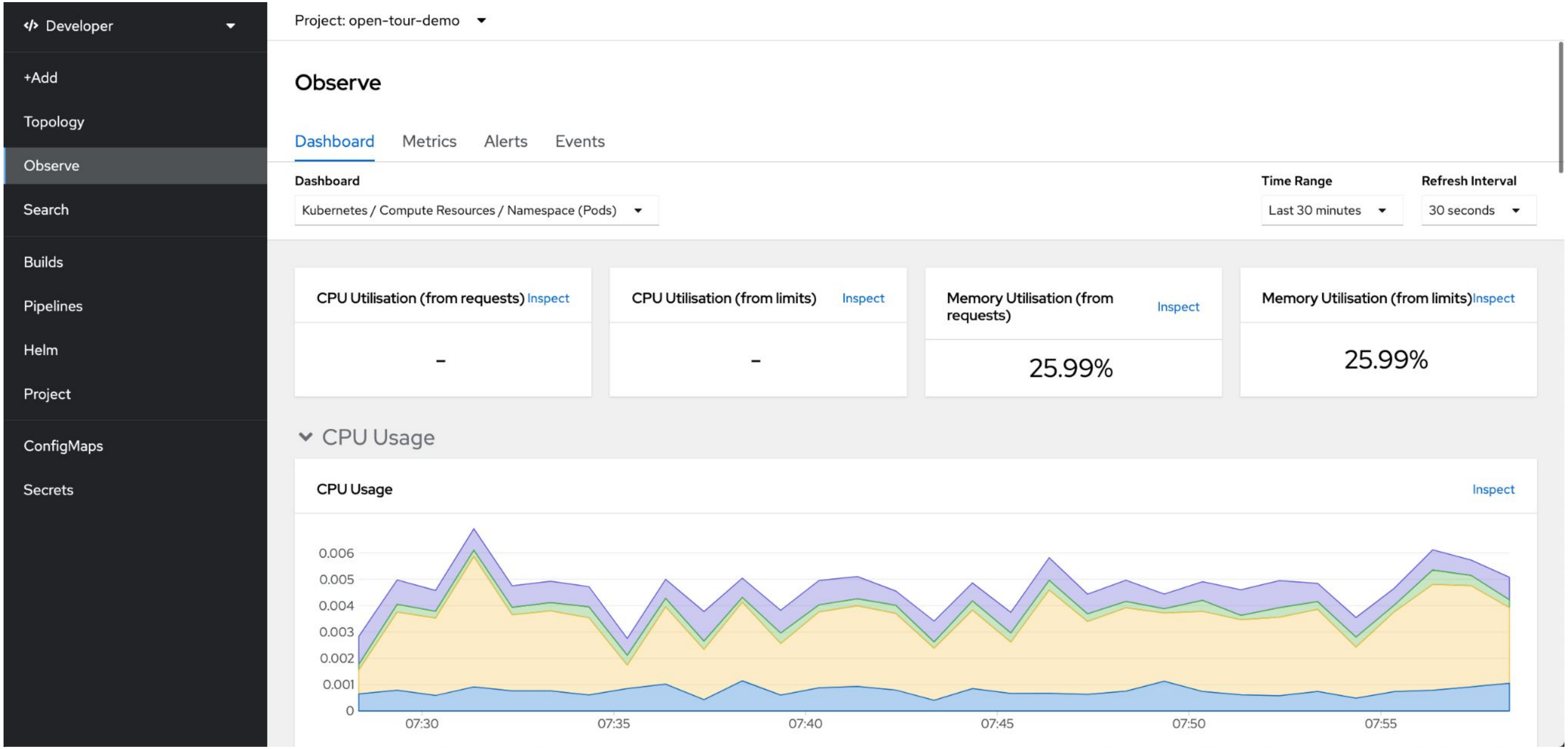
Service port: 8080-tcp → Pod port: 8080

Service port: 8443-tcp → Pod port: 8443

Service port: 8778-tcp → Pod port: 8778

Dev console demo - add from git

Observe



Project: open-tour-demo

Observe

Dashboard: Kubernetes / Compute Resources / Namespace (Pods)

Time Range: Last 30 minutes

Refresh Interval: 30 seconds

CPU Utilisation (from requests) Inspect

-

CPU Utilisation (from limits) Inspect

-

Memory Utilisation (from requests) Inspect

25.99%

Memory Utilisation (from limits) Inspect

25.99%

CPU Usage

CPU Usage Inspect

Dev console demo - view project

Recap – How to use the Developer Perspective

So you can easily navigate the many components of your applications

Easily deploy developer services on OpenShift

Dev catalogue - add capabilities - kafka etc

Recap –
Easily deploy developer services on OpenShift

So you can use modern day features, without the hassle

We showed you how to easily deploy applications on OpenShift

So you can spend more time developing

We showed you how to use the developer perspective

So you can easily overview and manage the many components of your applications.

We showed you how to easily deploy developer services on OpenShift

So you can use modern day features, without the hassle

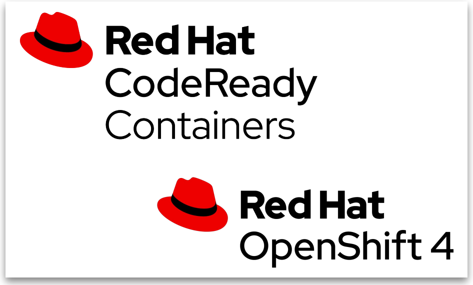
Developer Productivity in Scope

	Kubernetes	OpenShift
Developer Productivity	Out of scope	✓ Developer Perspective (app-centric topology view, live terminals, logs, and stats), Routes (easy hostname records), Developer Catalog (Operator-backed, Admin configurable)
Inner Loop	Out of scope	✓ CodeReady Workspaces (hosted IDE w/ VScode plugin support), **odo** ("cf push" style code promotion between commits)
Builds, Pipelines	Out of scope	✓ Tekton pipelines, Source-to-Image S2I, Buildah (runs builds on cluster, producing container images that **do not require root privs**)
Container Registry	Out of scope	✓ Quay w/ Clair for static image analysis (on-cluster)
Logging	Out of scope	✓ Prometheus (on-cluster, Developer Perspective dashboard includes PromQL and live data)
Service Mesh	Out of scope	✓ Istio, Kiali, Jaeger (on-cluster)
Serverless	Out of scope	✓ Knative (on-cluster, Developer Perspective dashboard integration)
Advanced Workloads & Marketplace	CRDs (v1.12+ recommended)	✓ Dashboard support for CRDs, Disk storage (volumes and claims), Operator Marketplace (on-cluster), Helm3, KubeVirt (VMs), GPUs

Red Hat

The Red Hat Developer Program

Technical developer content that educates; access to free software for development



Red Hat CodeReady Containers

Red Hat OpenShift 4

Free Products

Free access to downloadable and SaaS software for use in development and pre-production environments.

Red Hat DevNation

Technical talks and hands-on labs delivered virtually and physically. Book a private session for your organization.

RED HAT DEVELOPER

Containers

CHEAT SHEET

Bachir Chihani, Ph.
and Rafael Benevid

O'REILLY

Compliments of RED HAT DEVELOPERS

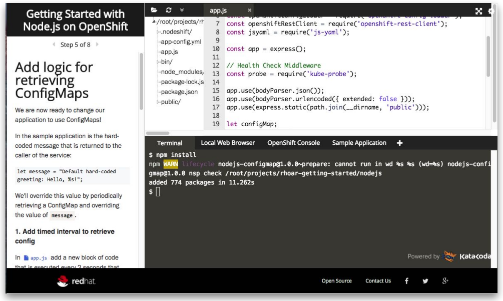
Microservices for Java Developers

A Hands-On Introduction to Frameworks & Containers

Christian Posta

Industry Leading Experts

E-books, quick reference guides, cheat sheets, and how-to articles and videos. Learn new technologies from the experts.



Getting Started with Node.js on OpenShift

Step 5 of 8

Add logic for retrieving ConfigMaps

We are now ready to change our application to use ConfigMaps!

In the sample application is the hard-coded message that is returned to the caller of the service:

```
let message = "Default hard-coded greeting: Hello, %s!";
```

We'll override this value by periodically retrieving a ConfigMap and overriding the value of `message`.

1. Add timed interval to retrieve config

In app.js add a new block of code

```
const openshiftRestClient = require('openshift-rest-client');
const jsyaml = require('js-yaml');

const app = express();

// Health Check Middleware
const probe = require('kube-probe');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(express.static(path.join(__dirname, 'public')));

let configMap;
```

```
$ npm install
npm WARN lifecycle nodejs-configmap@1.0.0~prepare: cannot run in wd %s %s (wd=%s) nodejs-confi
gmap@1.0.0 nsp check /root/projects/rhoar-getting-started/nodejs
added 774 packages in 11.262s
$ 
```

Labs and Trials

Faster, easier product evaluation and adoption through step-by-step tutorials and labs. Free trials for dev teams.

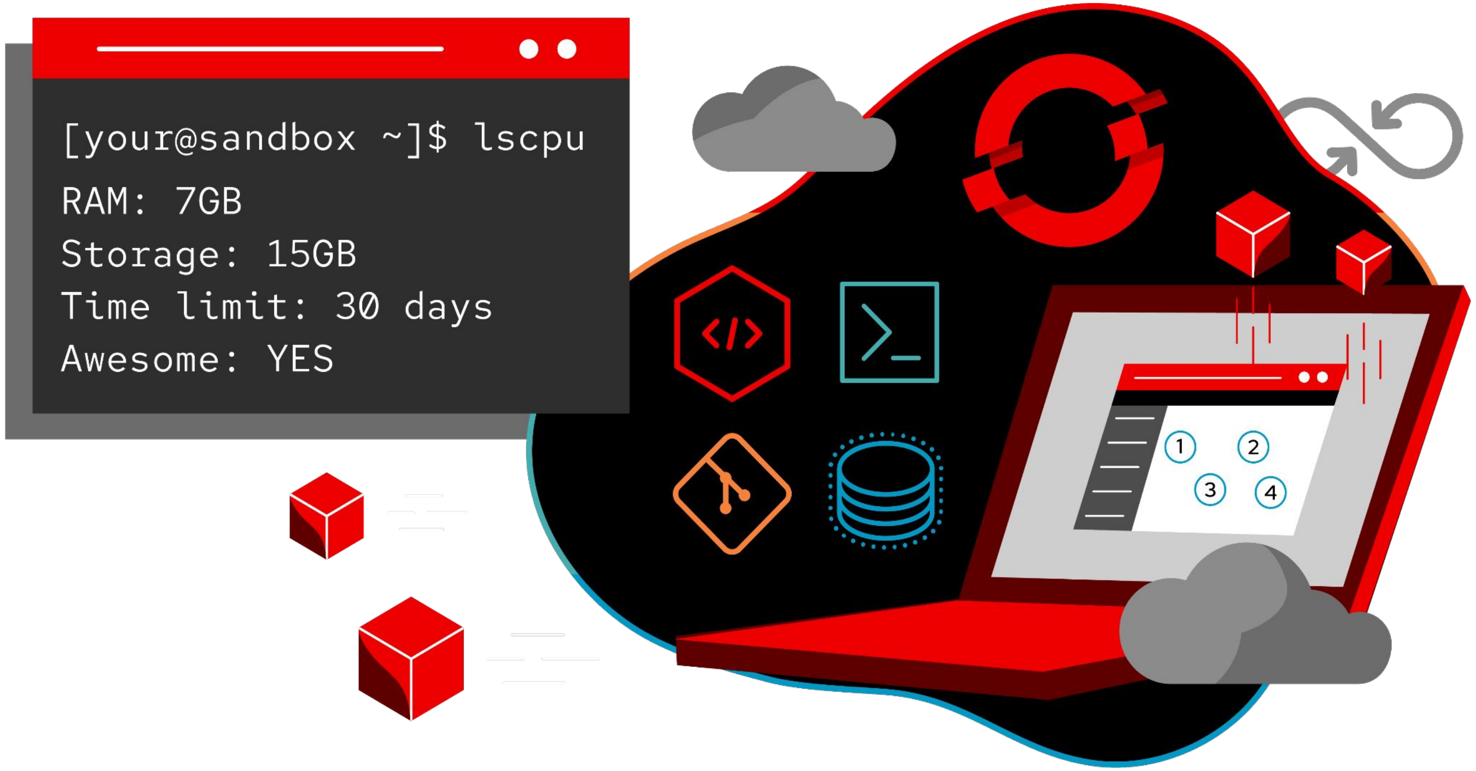
Sign up for free at developers.redhat.com

Red Hat
Developer

Developer Sandbox

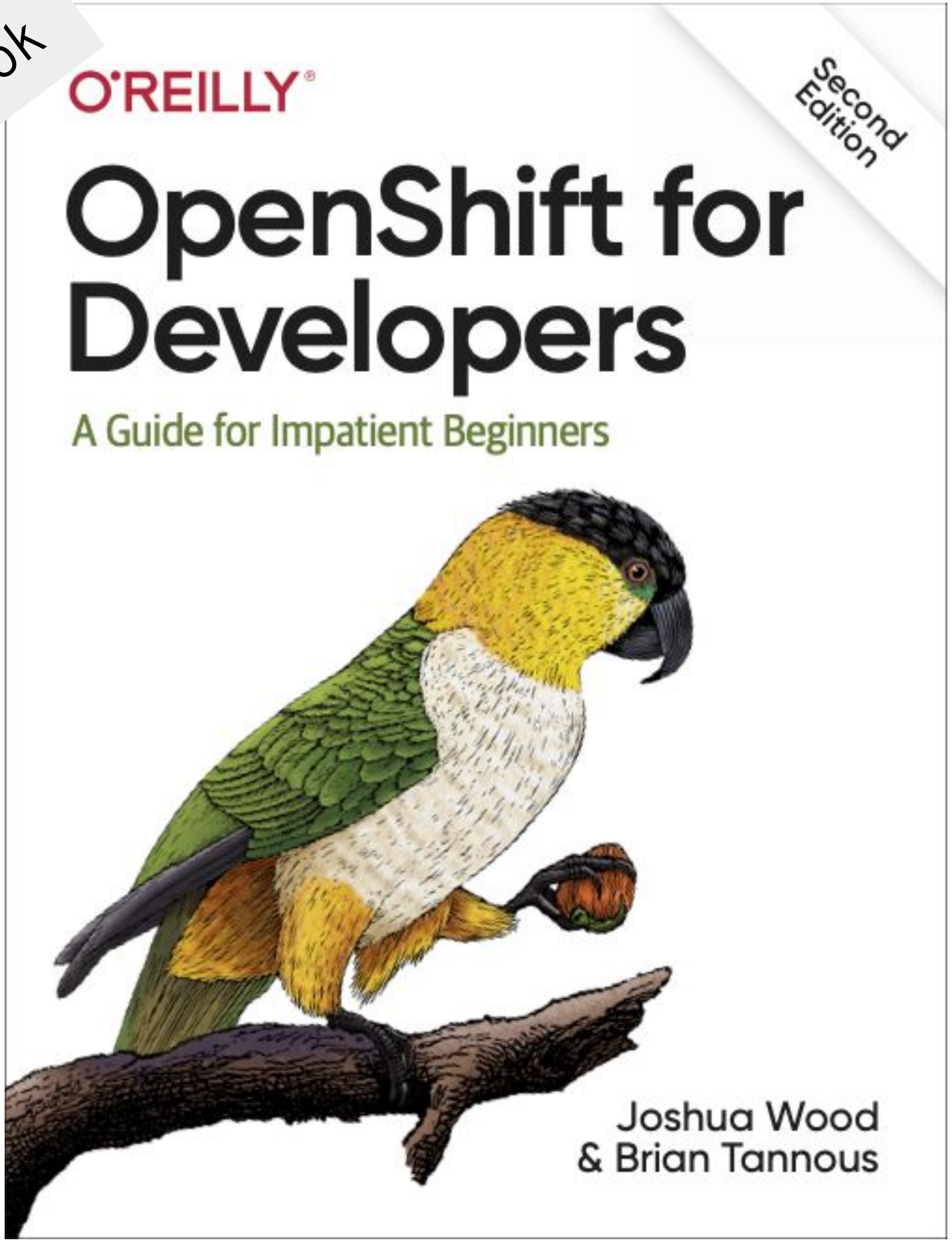
Get **free access** for renewable **30 days** to a self-service, cloud-hosted **Kubernetes** experience with **Developer Sandbox** for **Red Hat OpenShift**.

https://developers.redhat.com/developer-sandbox



```
[your@sandbox ~]$ lscpu
RAM: 7GB
Storage: 15GB
Time limit: 30 days
Awesome: YES
```

free e-book

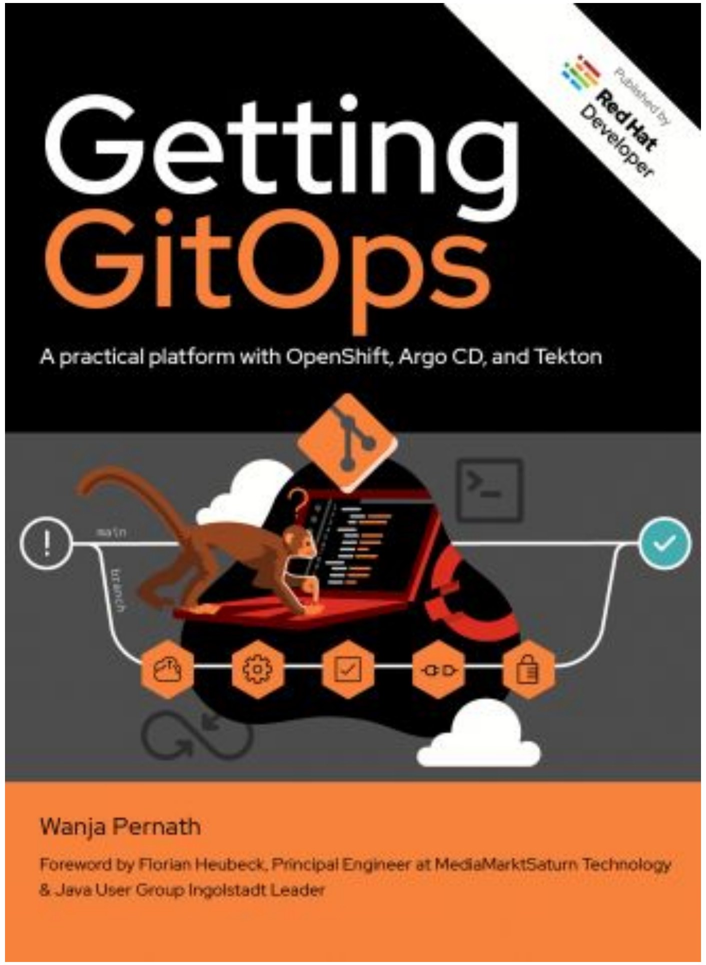


Download

https://red.ht/3IxJCzY

free e-book

Getting GitOps: A practical platform with OpenShift, Argo CD, and Tekton



Published by Red Hat Developer

Getting GitOps

A practical platform with OpenShift, Argo CD, and Tekton

Wanja Pernath

Foreword by Florian Heubeck, Principal Engineer at MediaMarktSaturn Technology & Java User Group Ingolstadt Leader

This book covers:

- How to install and use Quarkus for Java development
- How to configure an application to use a PostgreSQL database in a Kubernetes environment
- Basic Kubernetes files
- OpenShift Templates
- The Kustomize configuration management tool
- The Docker, Podman, Buildah, and Skopeo build tools
- Basic Helm charts and subcharts
- Kubernetes Operators
- CI/CD with Tekton
- CI/CD with OpenShift Pipelines
- GitOps with Argo CD
- Tekton security

https://developers.redhat.com/e-books/getting-gitops-practical-platform-openshift-argo-cd-and-tekton

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.

Corporate: linkedin.com/company/red-hat
Developer: linkedin.com/showcase/red-hat-developer/

Corporate: youtube.com/user/RedHatVideos
Developer:
youtube.com/channel/UC7noUdfWp-ukXUIAsJnSm-Q

Corporate: @RedHat
Developer: @rhdevelopers