How to speed up development productivity with OpenShift

Eero Arvonen
Architect

Red Hat

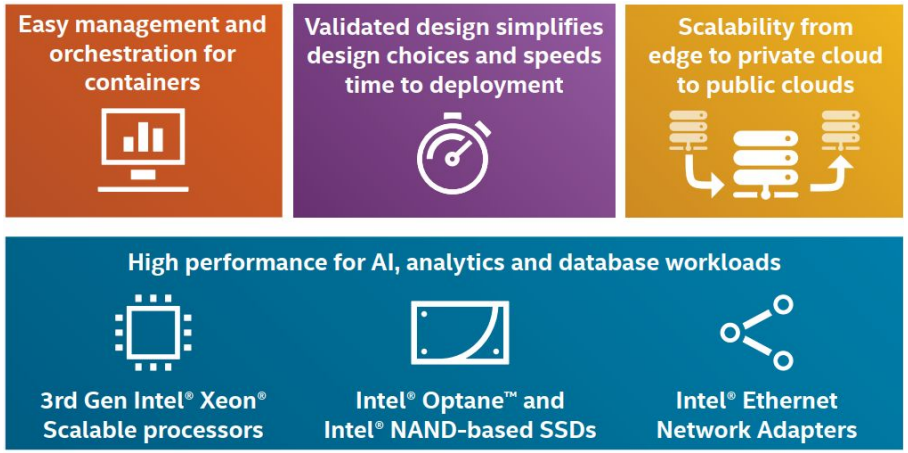
Red Hat OpenShift Reference Architecture

Joint Red Hat and Intel OpenShift Reference Architecture

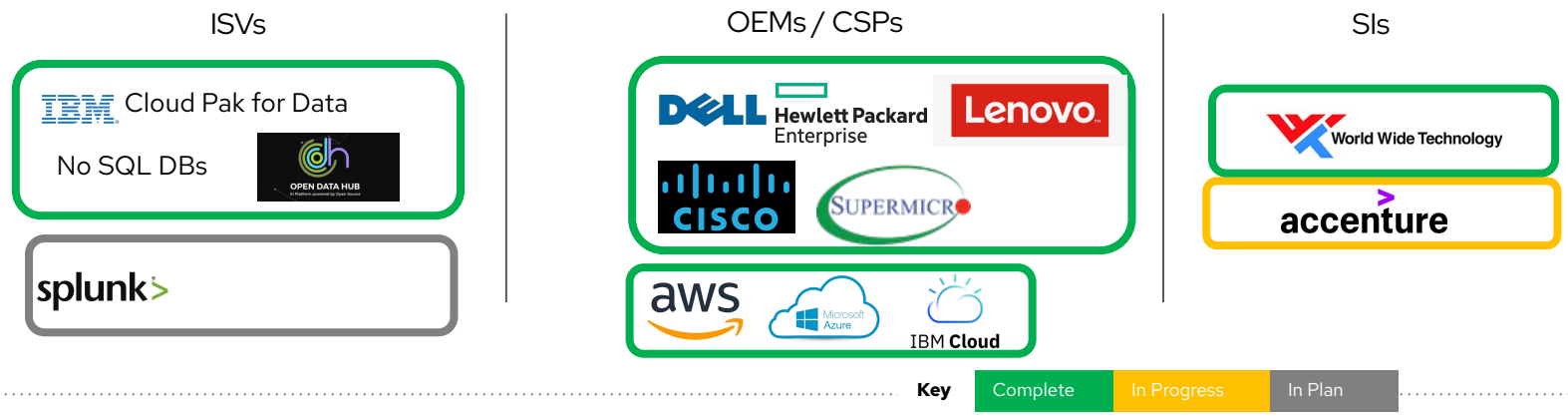
Solution overview

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



Solution ecosystem



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

Red Hat | intel.

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

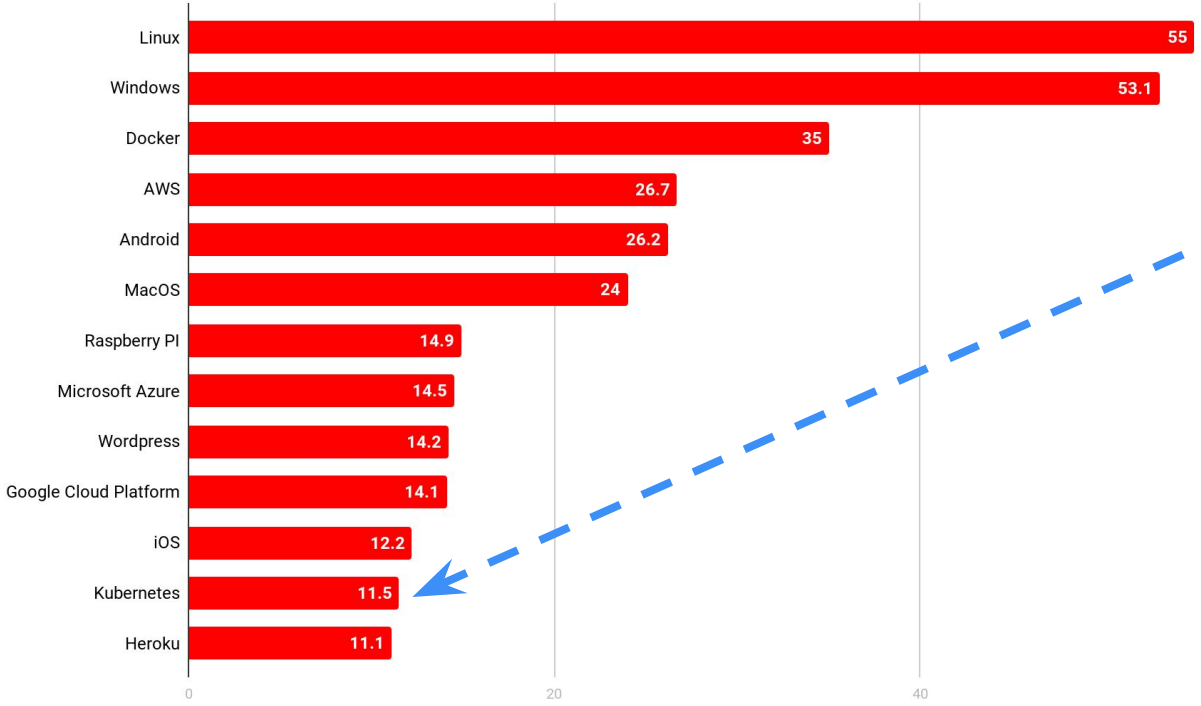
Red Hat

Kubernetes solves all your problems!?

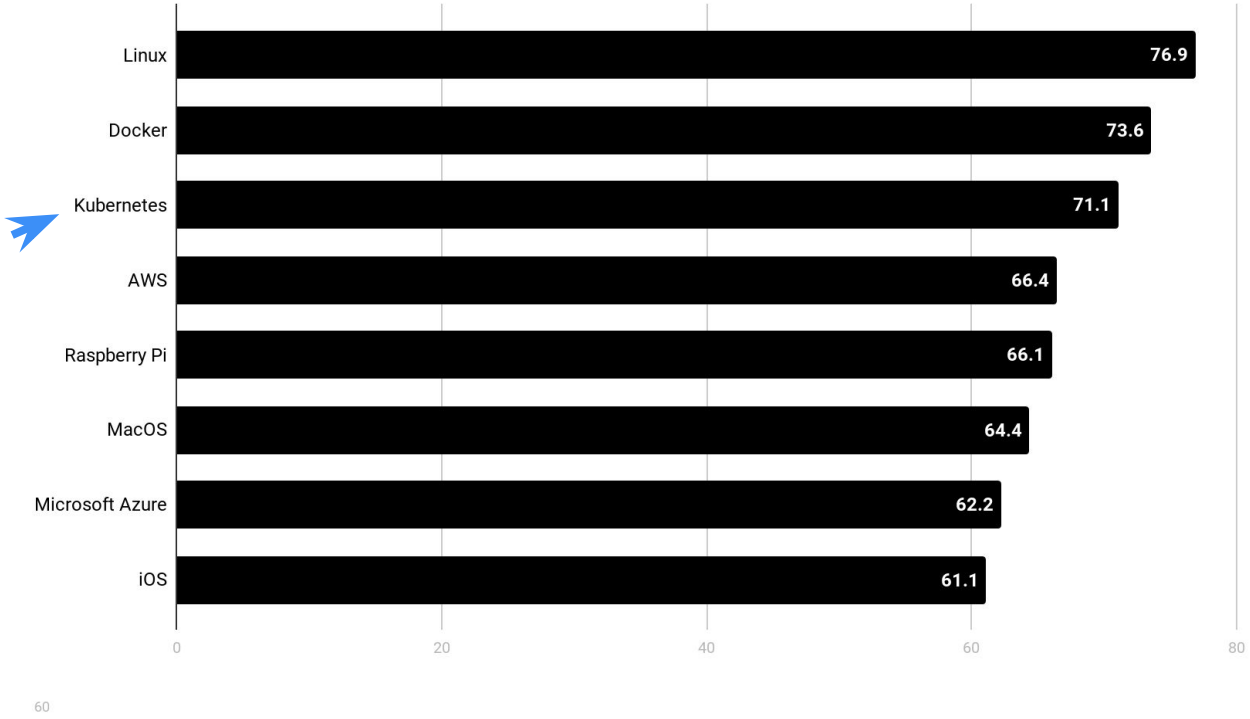
Red Hat

Kubernetes and Developers

Platform Adoption**



Most Loved Platforms**



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

Red Hat

Unintended complexities of adoption...

Ask a Developer

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

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

Feeling Lucky?

What is Kubernetes - Kubernetes - k8s.io/docs/concepts/overview/what-is-kubernetes/

kubernetes

Documentation Blog Training Partners Community Case Studies English v1.18

Concepts

HOME GETTING STARTED CONCEPTS TASKS TUTORIALS REFERENCE CONTRIBUTE

Search

Concepts

- Overview
  - What is Kubernetes?
  - Kubernetes Components
  - The Kubernetes API
  - Working with Kubernetes Objects
- Cluster Architecture
- Containers
- Workloads
- Services, Load Balancing, and Networking
- Storage
- Configuration
- Security
- Policies
- Scheduling
- Cluster Administration

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

kubernetes

Documentation Blog Training Partners Community Case Studies English v1.18

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.

Unintended complexities of 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

okd
falco
HELM
CERT MANAGER
kubernetes
CODENVY
Red Hat

...And Red Hat provides developer tools that support your journey to cloud-native application development

"I JUST WANT TO CODE" — Simple / Opinionated

"I WANT TO BE A KUBE EXPERT" — 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

Red Hat

KUBERNETES DONE RIGHT IS HARD

INSTALL

- Templating
- Validation
- OS Setup

DEPLOY

- Identity & Security Access
- App Monitoring & Alerts
- Storage & Persistence
- Egress, Ingress & Integration
- **Host Container Images**
- **Build/Deploy Methodology**

HARDEN

- Platform Monitoring & Alerts
- Metering & Chargeback
- Platform Security Hardening
- Image Hardening
- Security Certifications
- Network Policy
- Disaster Recovery
- Resource Segmentation

OPERATE

- OS Upgrade & Patch
- Platform Upgrade & Patch
- Image Upgrade & Patch
- App Upgrade & Patch
- Security Patches
- Continuous Security Scanning
- Multi-environment Rollout
- Enterprise Container Registry
- Cluster & App Elasticity
- Monitor, Alert, Remediate
- Log Aggregation

75%

of enterprise users identify complexity of implementation and operations as the top blocker to adoption

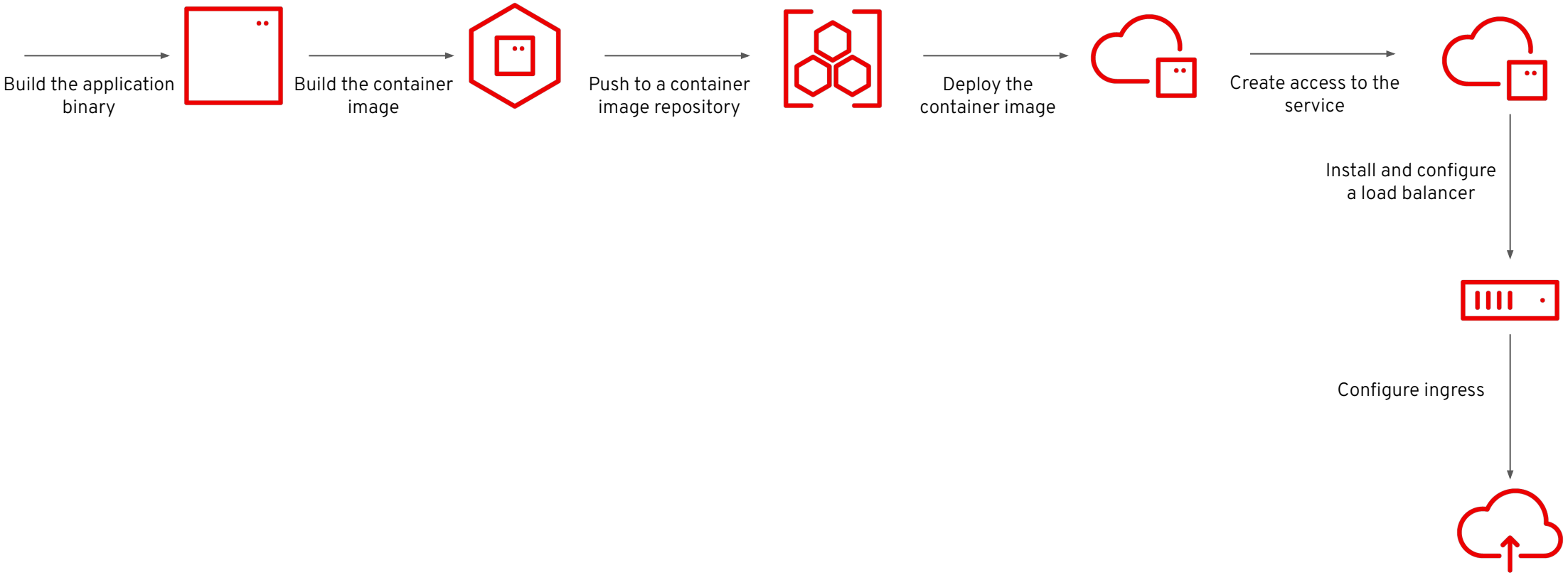
Source: The New Stack, The State of the Kubernetes Ecosystem, August 2017

Red Hat

How to easily deploy things on OpenShift

Red Hat

Deploy an application on Kubernetes



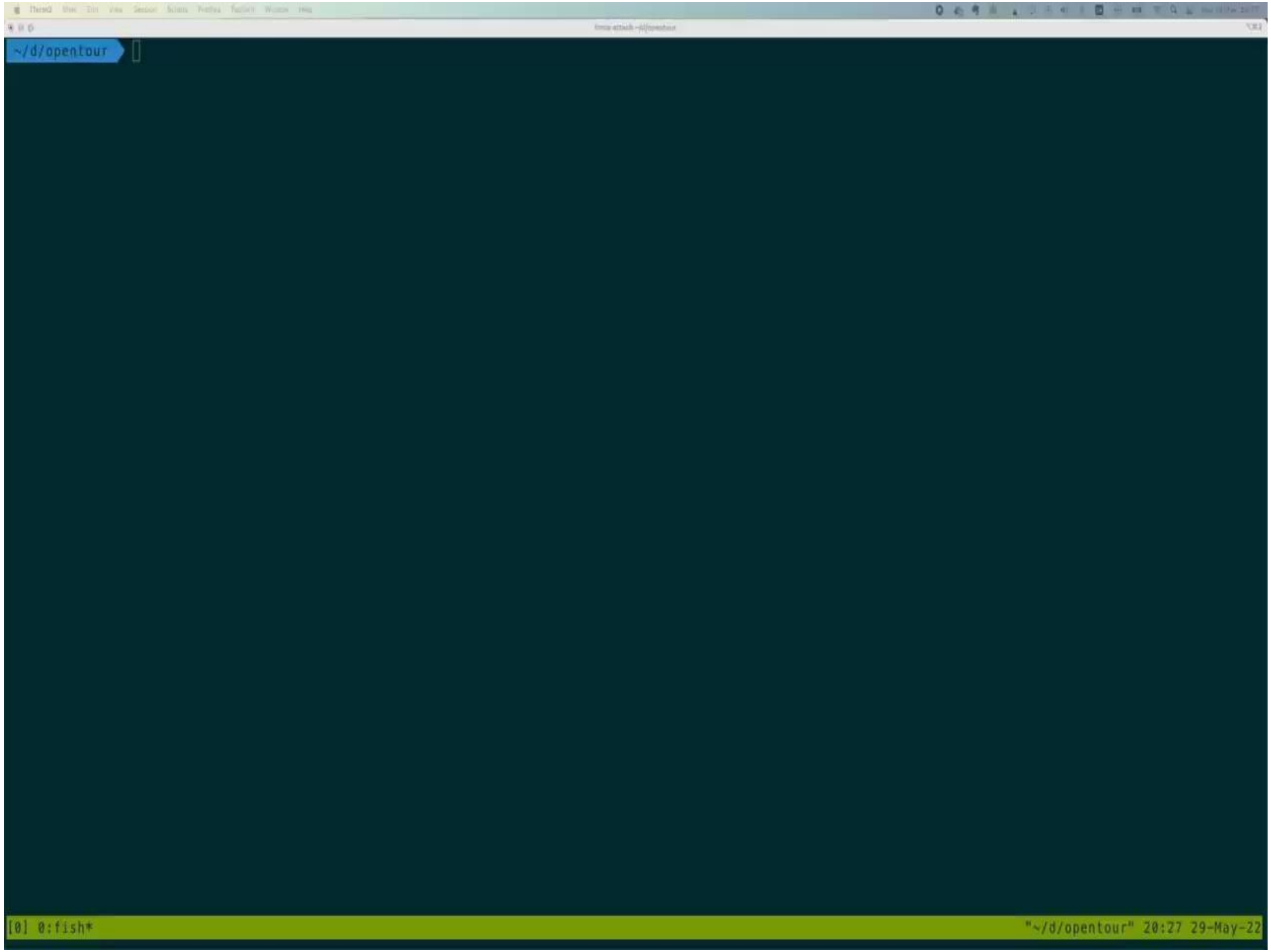
https://medium.com/swlh/im-so-sorry-openshift-i-ve-taken-you-for-granted-f36fb47ea4d9

Red Hat

Deploy an application on Kubernetes

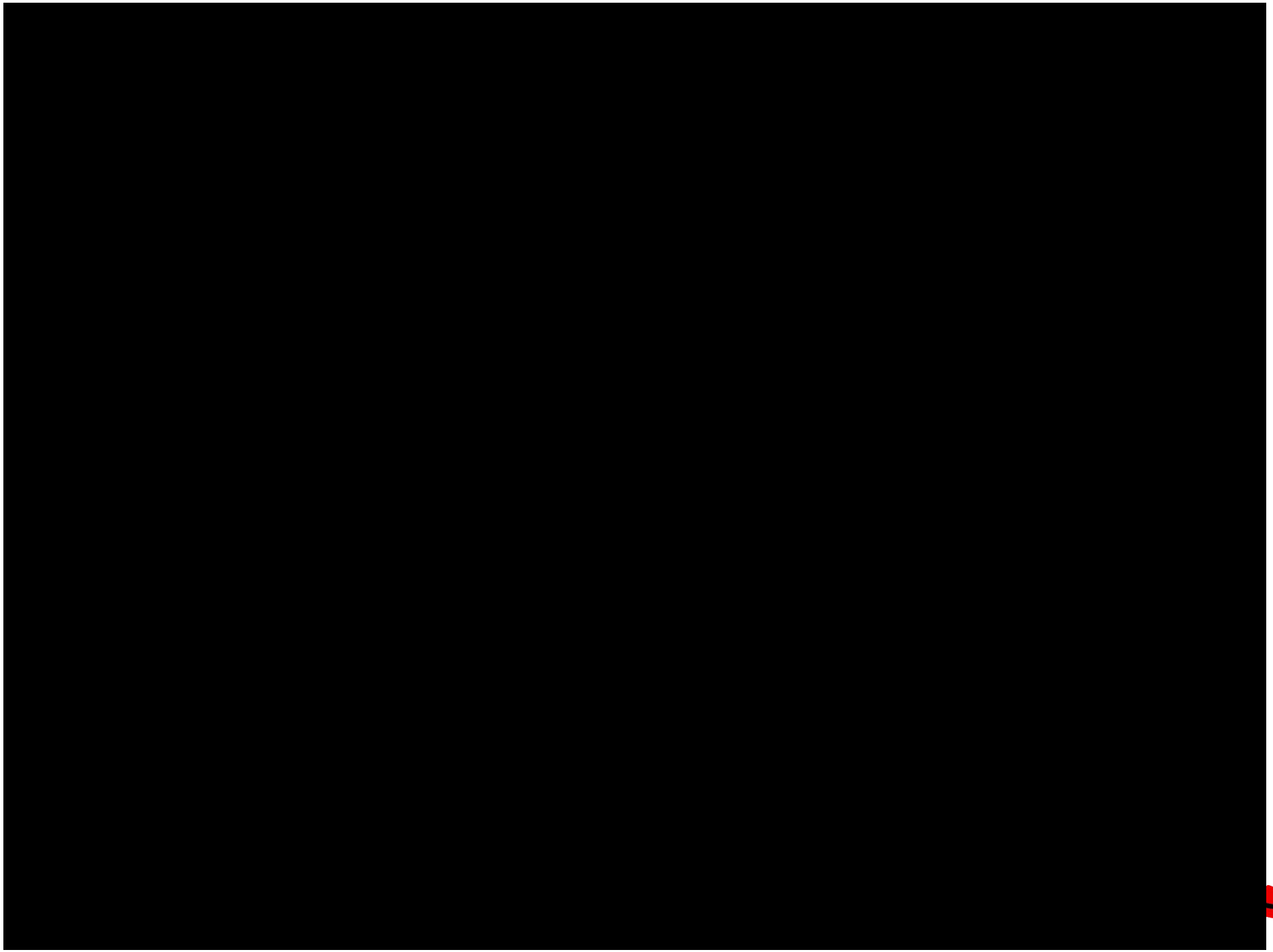
1. Package the application binary
2. Build the container image (Dockerfile, base image)
3. Install or sign up for a public container image repository
4. Push the container to the container image repository
5. Deploy the container image
6. Create access to the service
7. Install and configure a load balancer
8. Configure ingress

S2I demo



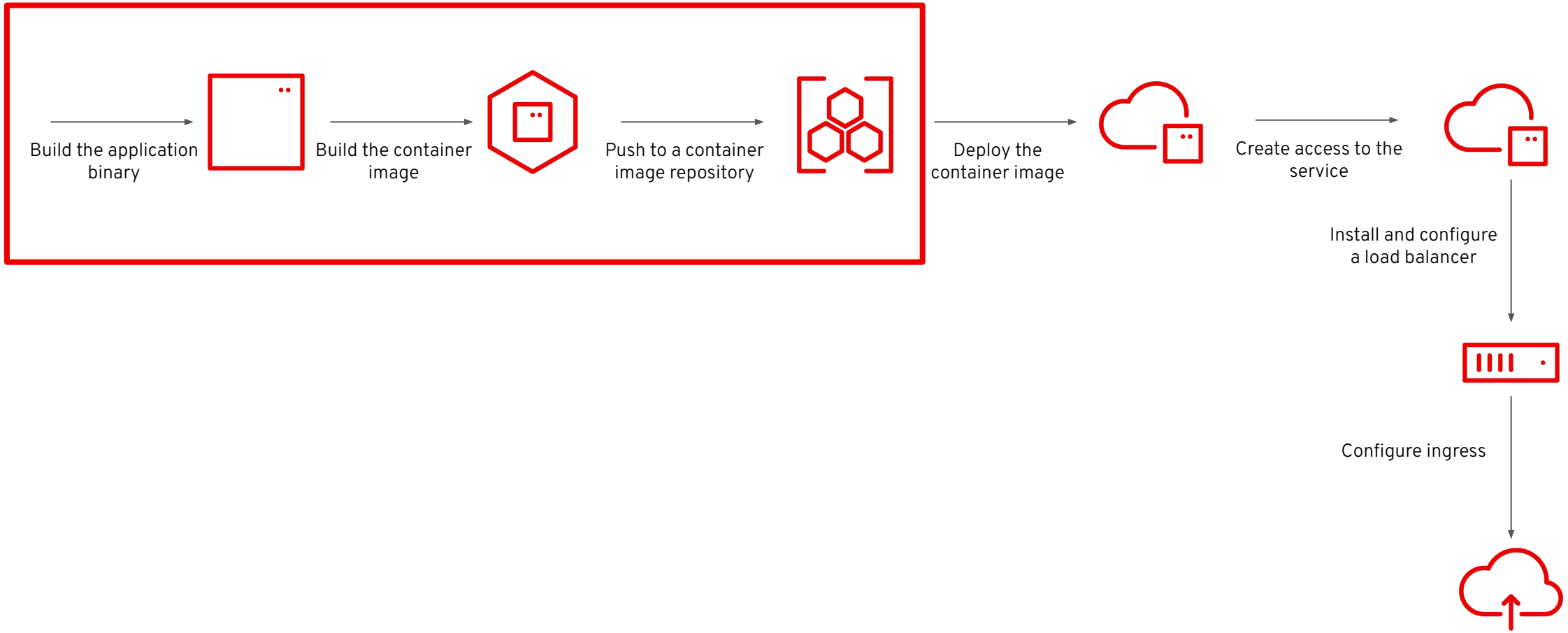
Red Hat

oc new-app demo



Red Hat

Deploy an application on OpenShift



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

Red Hat

Automating - Source-to-Image(S2I)

```
s2i build https://github.com/sclorg/django-ex centos/python-35-centos7 hello-python
```

- Creates a container based on the build image using the application source in src
- Sets the environment variables from .s2i/environment (optional)
- Starts the container and runs its assemble script
- When done, commits the container, setting the CMD for the output image to be the run script and tagging the image with the name provided.

```
oc new-app  registry.access.redhat.com/ubi8/openjdk-8:latest~https://github.com/mostmark/hello-spring-boot.git
oc expose svc/hello-spring-boot
```

Red Hat

What happened? Source-to-Image explained

- Created a build pod to do “stuff” for building the app
- Created an OpenShift Build config
- Pulled the builder image into OpenShift’s internal docker registry.
- Cloned the “Hello World” repo locally
- Seen that 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 pod spec, service spec etc.
- Kicked off a deploy of the container image.
- Removed the build pod.
- Exposed the service in the route (ingress)

Red Hat

Demo script

Local build with s2i

```
docker images

s2i build https://github.com/mostmark/hello-spring-boot.git registry.access.redhat.com/ubi8/openjdk-8:latest hello-spring-boot

docker images

docker run -p 8080:8080 hello-spring-boot
```

Build and deploy to OpenShift with oc new-app

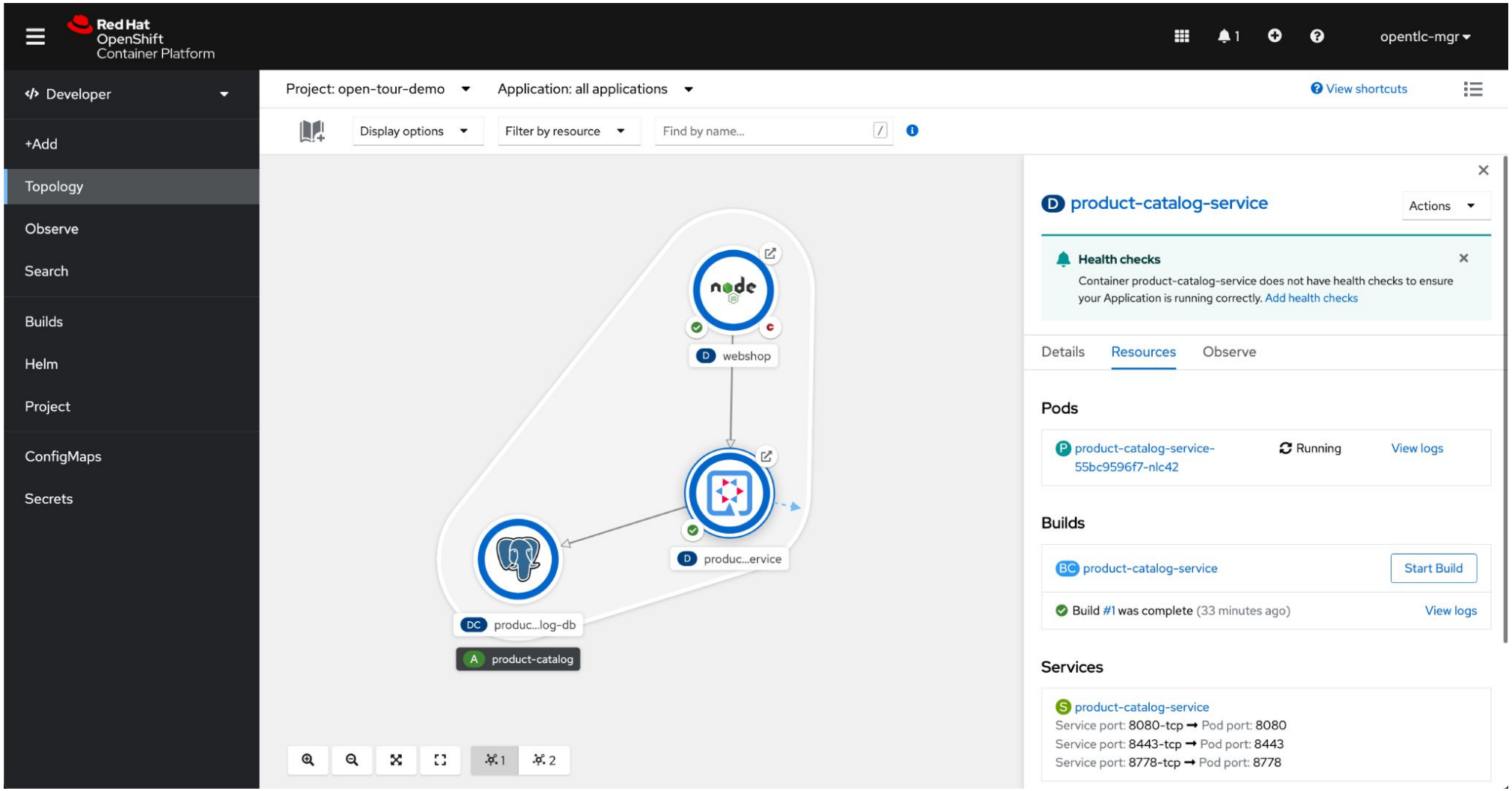
```
oc new-app registry.access.redhat.com/ubi8/openjdk-8:latest~https://github.com/mostmark/hello-spring-boot.git

oc expose service/hello-spring-boot

oc get all

oc logs -f buildconfig/hello-spring-boot

oc start-build hello-spring-boot
```


Red Hat

How to use the Developer Perspective

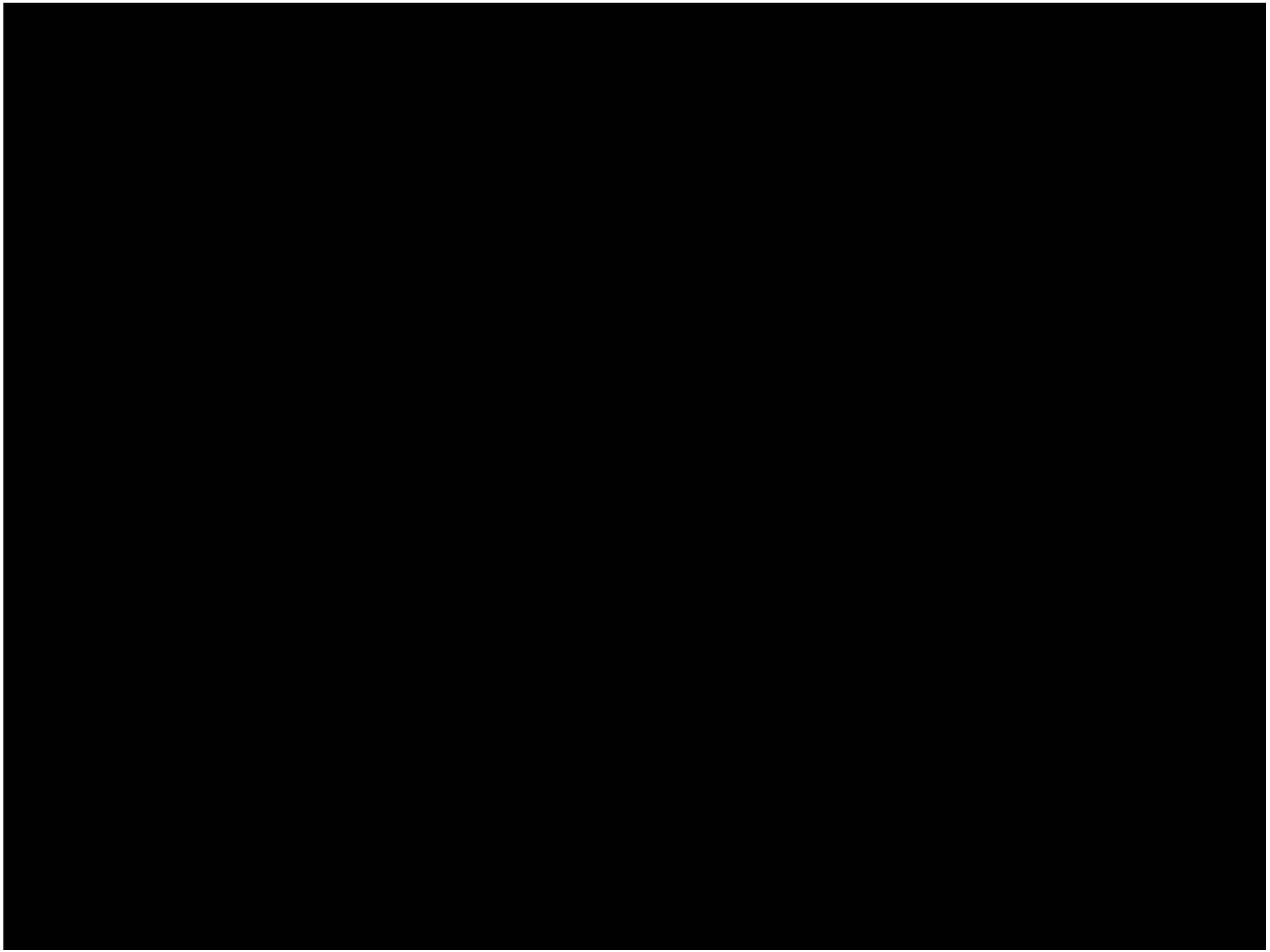
Red Hat

Topology view



Red Hat

Add from git



Observe

Developer

+Add

Topology

Observe

Search

Builds

Pipelines

Helm

Project

ConfigMaps

Secrets

Project: open-tour-demo

Observe

Dashboard Metrics Alerts Events

Dashboard

Kubernetes / Compute Resources / Namespace (Pods)

Time Range

Last 30 minutes

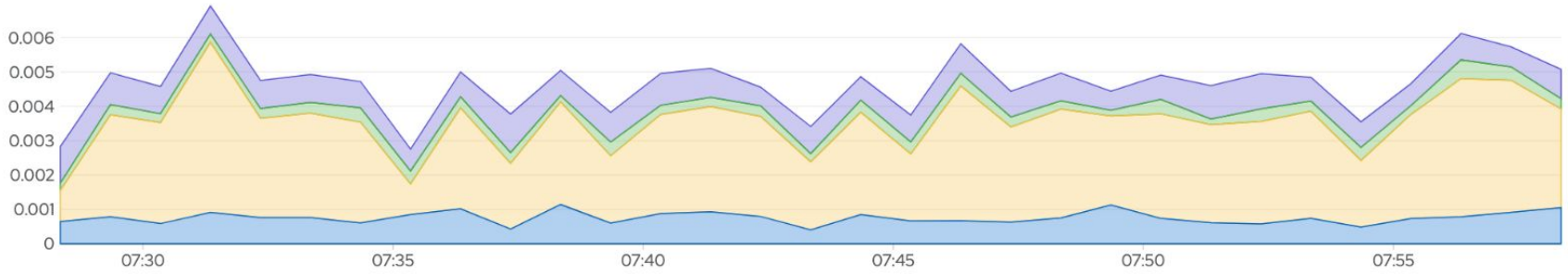
Refresh Interval

30 seconds

CPU Utilisation (from requests) Inspect	CPU Utilisation (from limits) Inspect	Memory Utilisation (from requests) Inspect	Memory Utilisation (from limits) Inspect
-	-	25.99%	25.99%

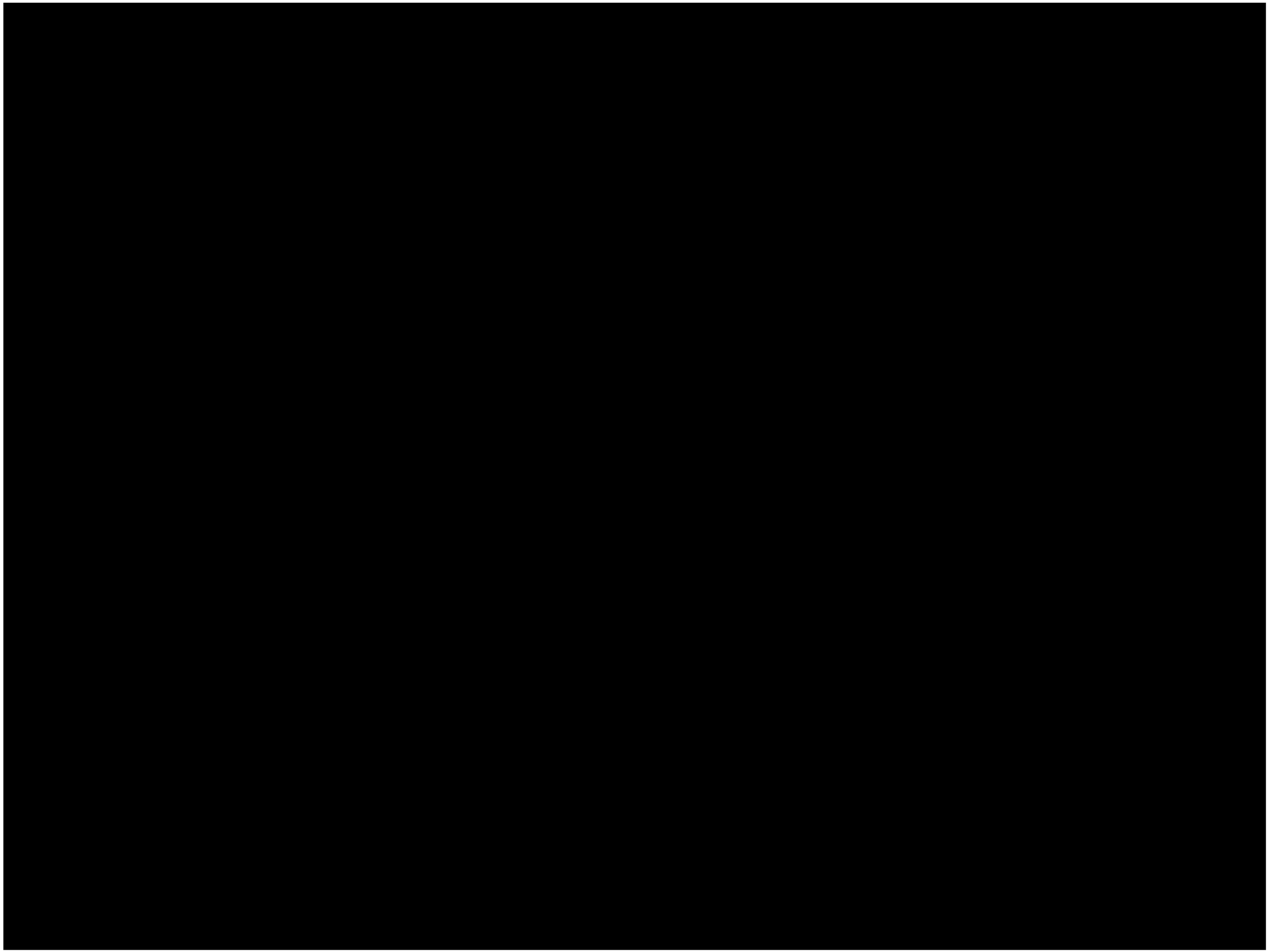
CPU Usage

CPU Usage Inspect



Red Hat

Project



Red Hat

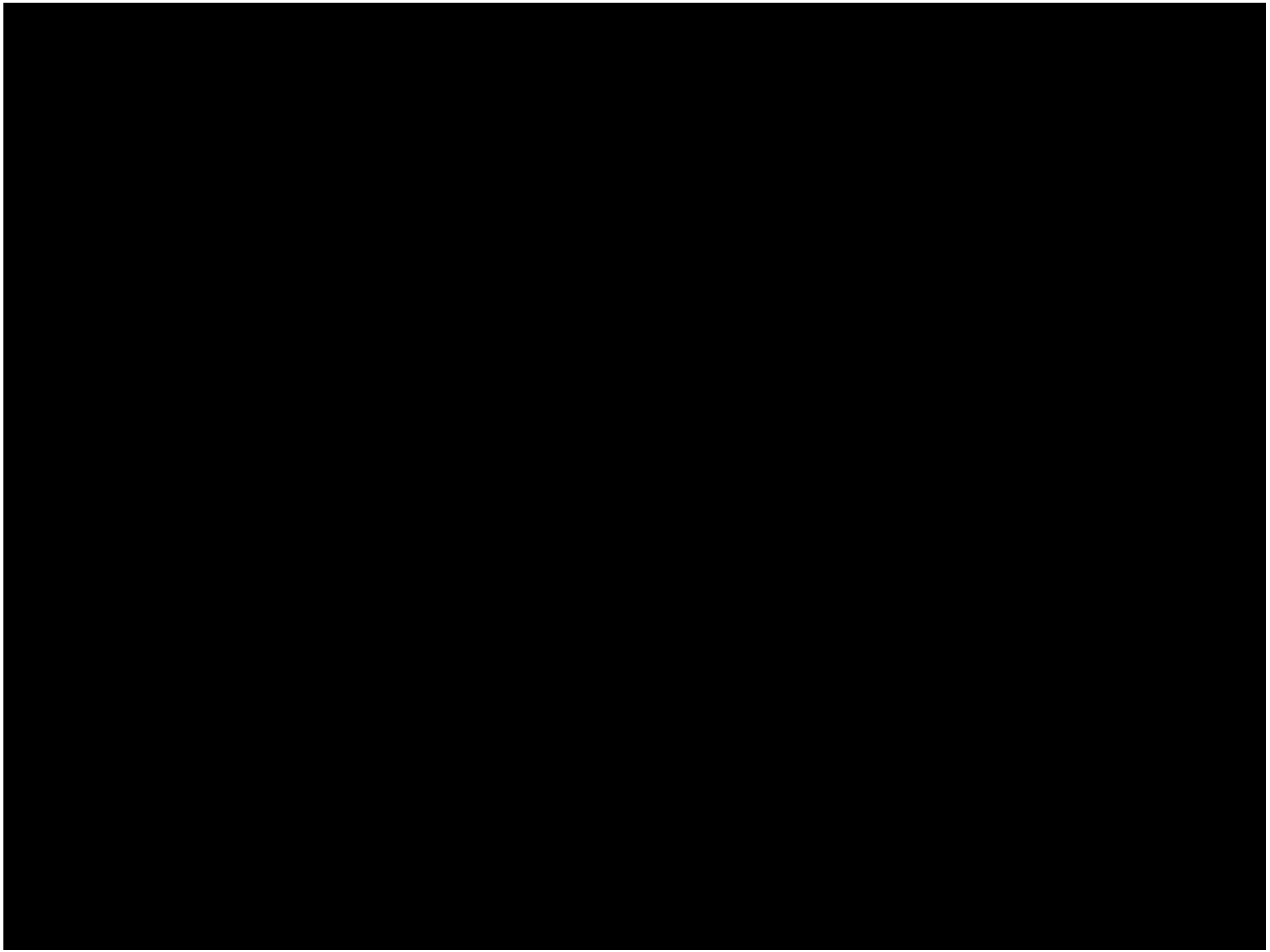
How to use the Developer Perspective

So you can easily navigate the many components of your applications

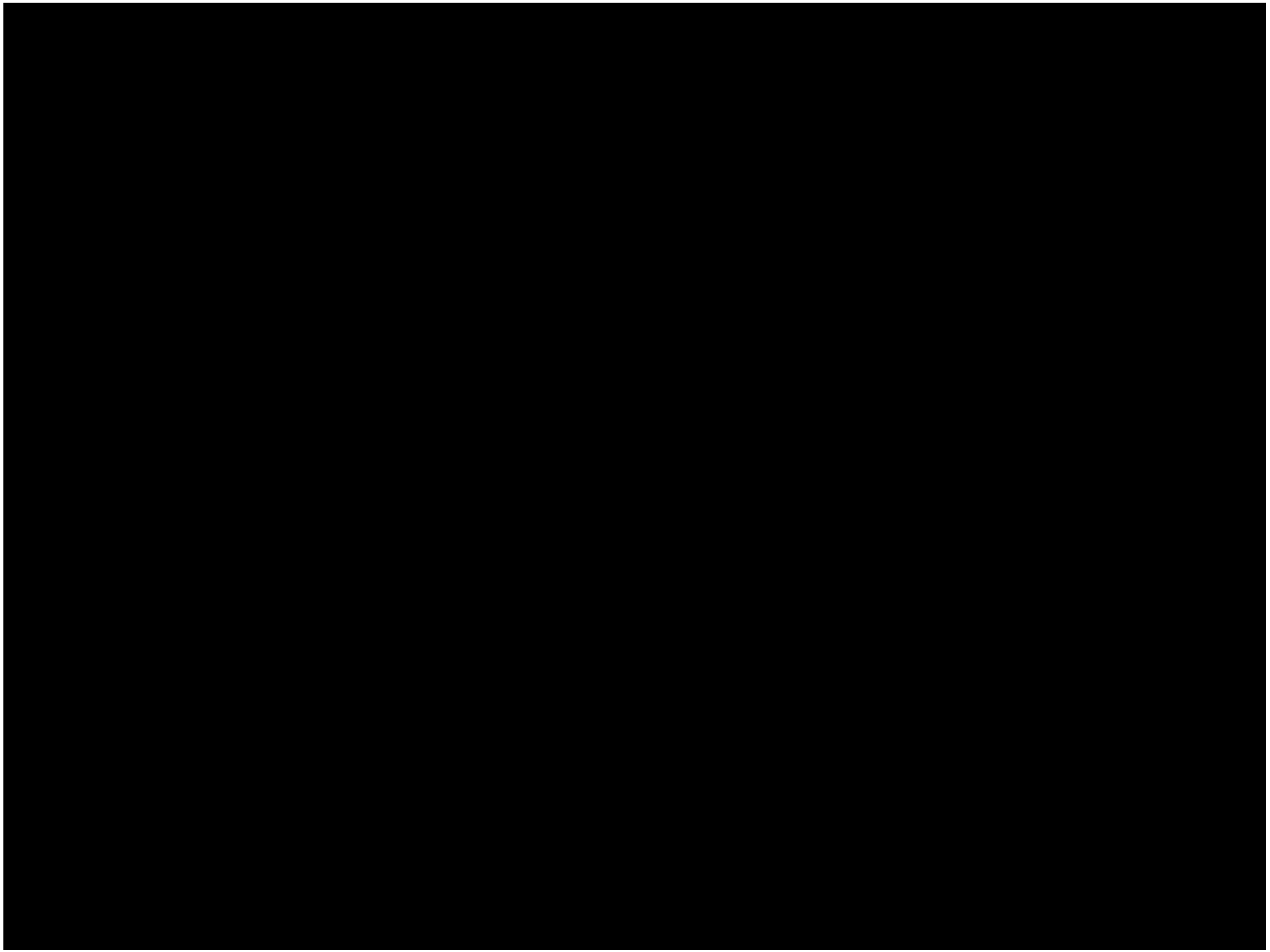
Red Hat

Easily deploy developer services on OpenShift

Red Hat



Red Hat



Red Hat

Easily deploy developer services on OpenShift

So you can use modern day features, without the hassle

Red Hat

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

Summary

Red Hat

We showed you how to easily deploy applications on OpenShift

So you can spend more time developing

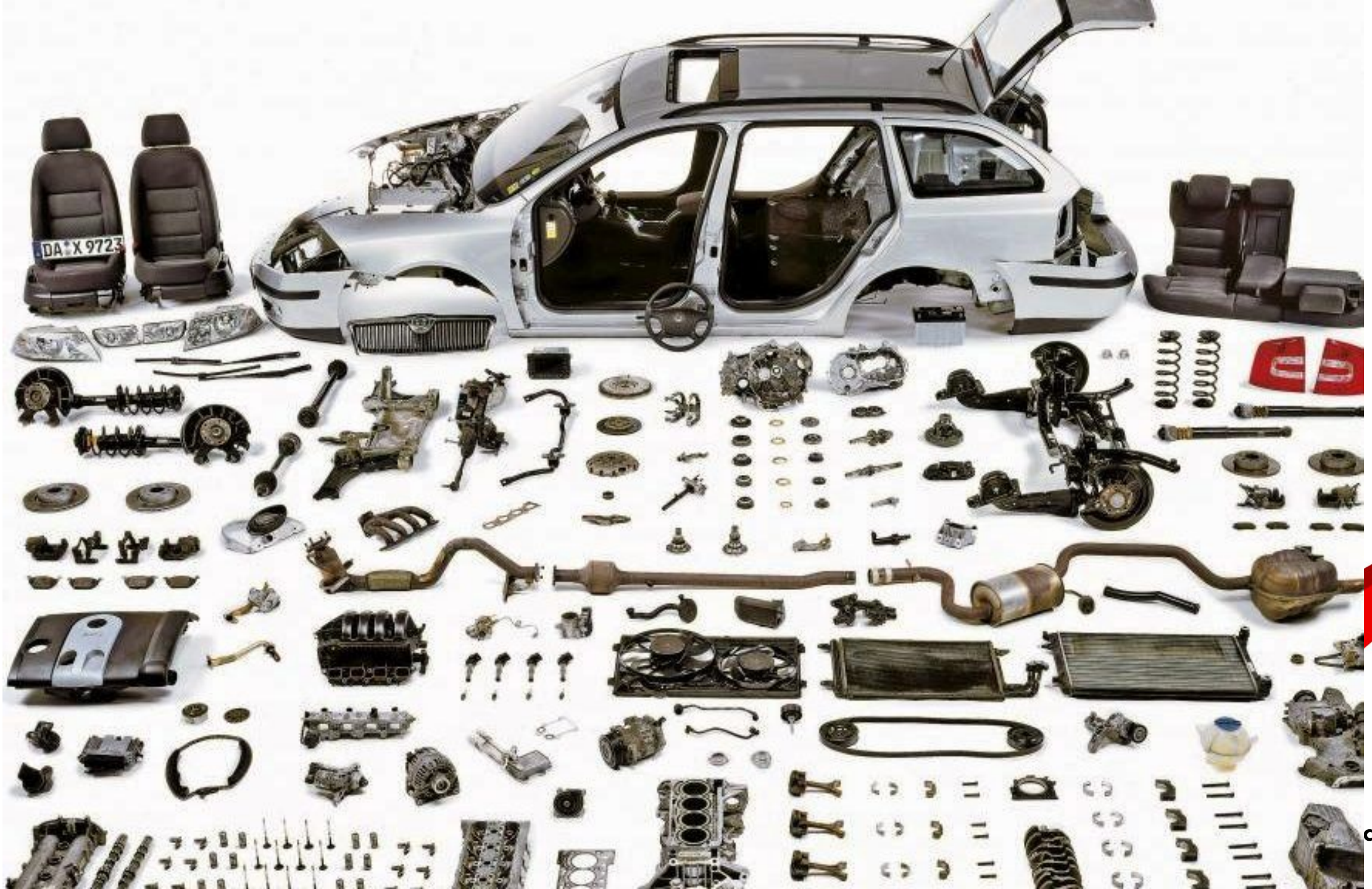
We showed you how to use the developer perspective

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

Red Hat



d Hat

Red Hat

Lufthansa Technik

Anticipates significant reductions in MRO costs for participating airlines—equating to **millions of euros** per airline

Challenge

Improve airlines' technical operations.

Solution

Lufthansa Technik built and operated AVIATAR using a hybrid cloud infrastructure based on enterprise open source software from Red Hat.

Why Red Hat

Lufthansa Technik needed a flexible, scalable environment that could run multiple applications using a shared repository of industry data.

Results

- Achieved faster application workflows
- Created greater flexibility through improved integration between internal infrastructure and third-party solutions
- Anticipates significant savings for participating airlines on maintenance, repair, and overhaul (MRO)

Products and services

Red Hat® Enterprise Linux®
Red Hat Gluster® Storage
Red Hat Data Grid
Red Hat Fuse
Red Hat OpenShift® Container Platform
Red Hat JBoss® Enterprise Application Platform
Red Hat 3scale API Management
Red Hat Ansible® Automation Platform

Sources: Red Hat success story page. "Lufthansa Technik builds a digital foundation with Red Hat," May 2018.
Microsoft Azure video. "Red Hat and Microsoft provide the foundation for Lufthansa Technik's hybrid cloud," Oct. 2018.

Red Hat

Red Hat's application portfolio, and cloud vendor and partner integrations make your development teams more agile

Schiphol
Amsterdam Airport

50%

Reduction in application development time*

- Development pipeline, source-to-image builds and flexible developer tooling speed application development
- 3Scale API management and Fuse service integration platform allows development teams who are not "coders" to create business applications.
- OpenShift global technical support to run thousands of containers and mission critical workloads.

Schiphol Airport Case Study: https://www.redhat.com/en/success-stories/amsterdam-airport-schiphol

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

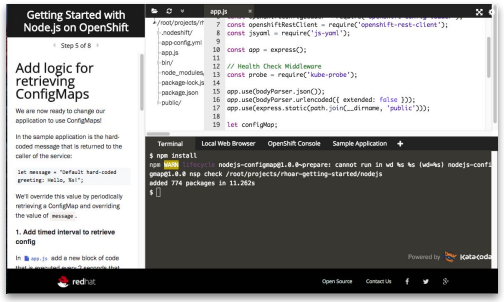
Free access to downloadable and SaaS software for use in development and pre-production environments.

Red Hat DevNation

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

Industry Leading Experts

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



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

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

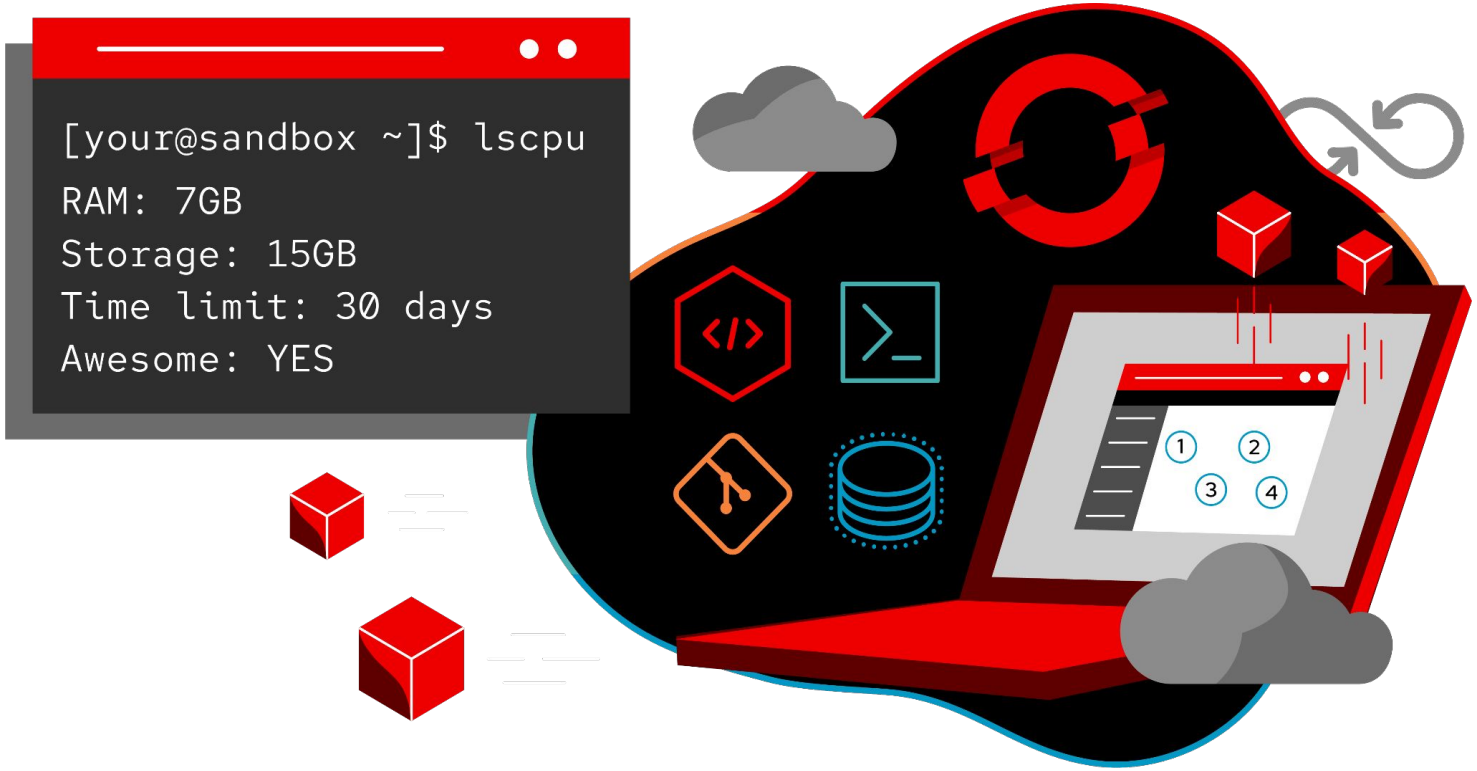
Red Hat

Red Hat Developer

Developer Sandbox

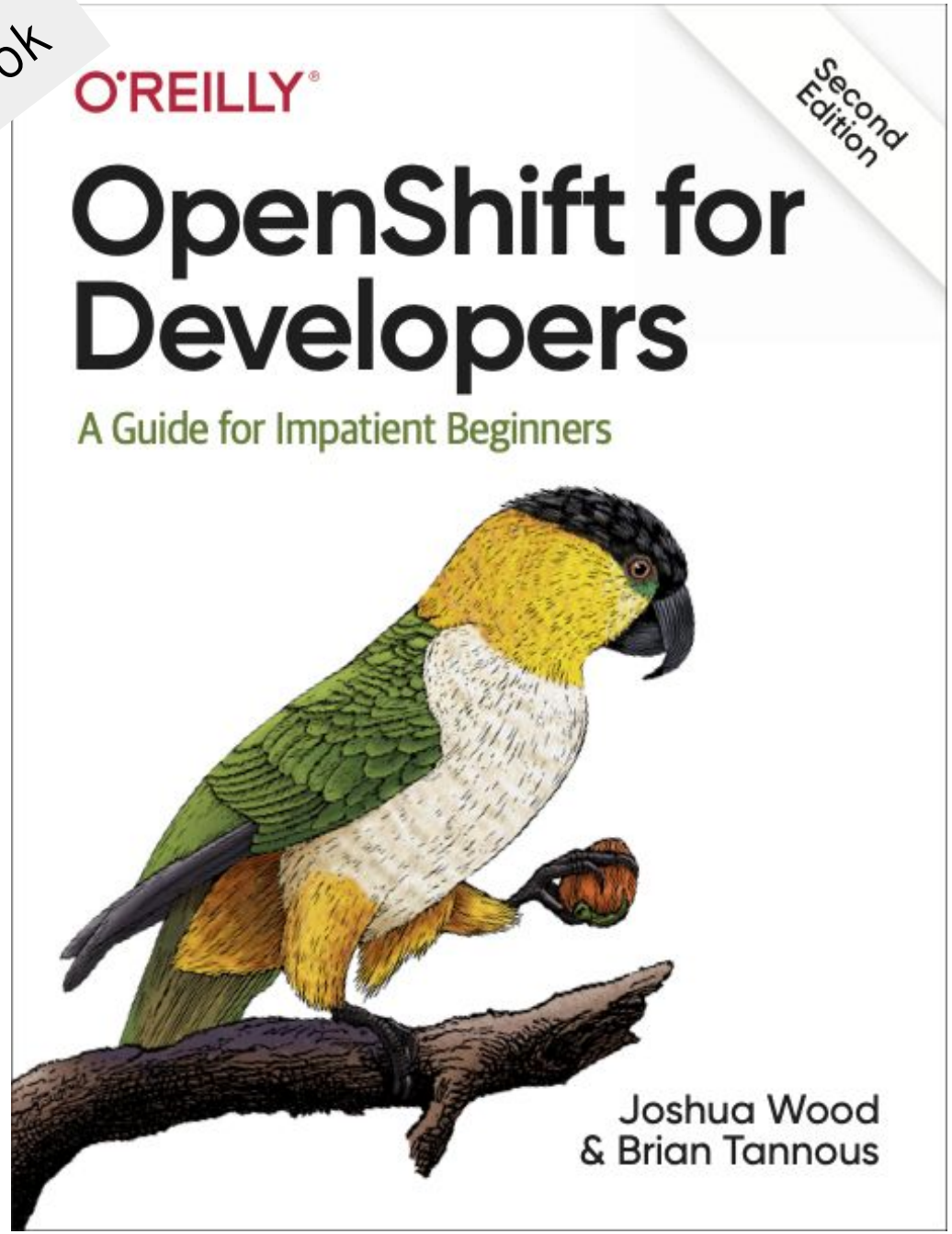
Get **free access** for renewable **30 days** to a self-service, cloud-hosted **Kubernetes** experience with **Developer Sandbox** for **Red Hat OpenShift**.

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



Red Hat

free e-book



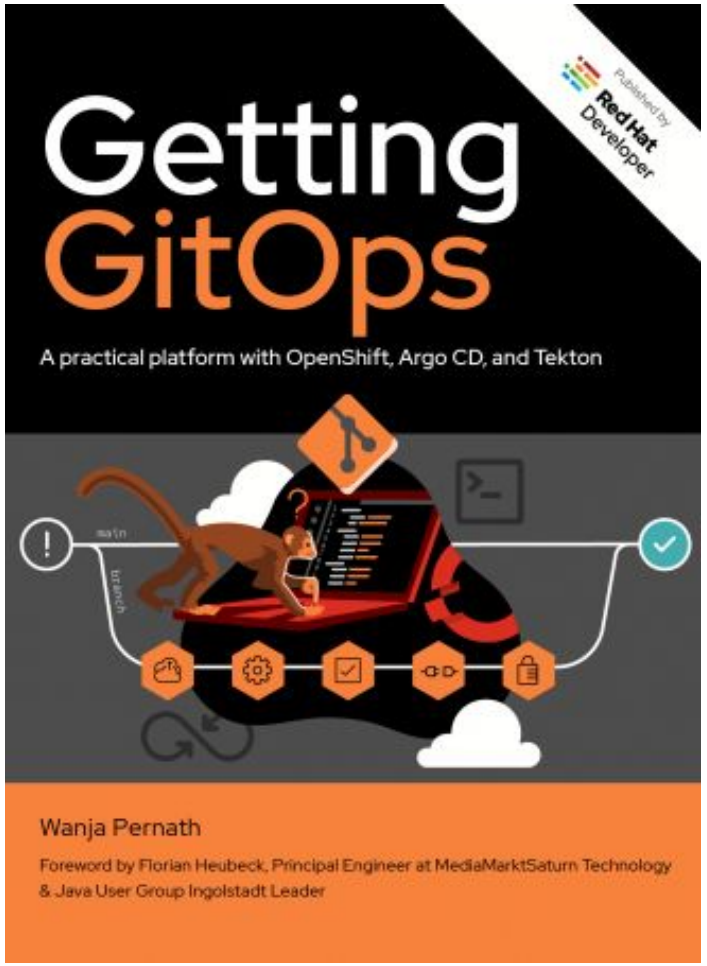
[Download](https://red.ht/3lxJCzY)

https://red.ht/3lxJCzY

Red Hat

free e-book

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



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

Red Hat

CONFIDENTIAL
Internal use only

Alex Groom

Based in UK

Specialist Solution Architect, EMEA

Focus on OpenShift Developer Adoption

+25 years in Software Development

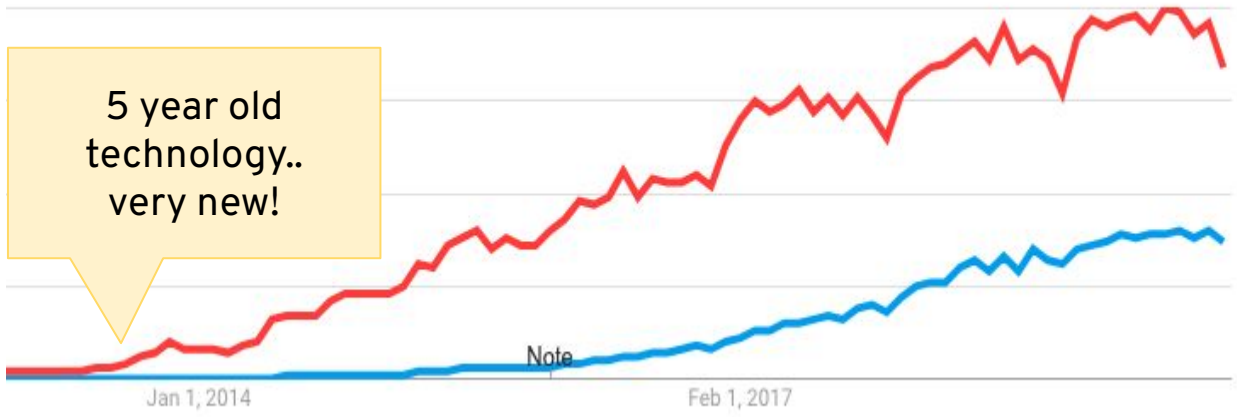
">80% of my development teams aren't Kubernetes and container experts"

Challenge

- Today development happens on laptops and only sometimes in containers
- IT is moving to Kubernetes in production
- Devs have programming knowledge, but little experience with containers and Kubernetes
- Training every developer to become a container and Kube expert isn't viable: too hard, too long

Need a way to make devs productive on Kube now, but enable their learning journey going forward

Google Search Trend: Kubernetes and Linux Containers



Red Hat

Building a Platform is not YOUR business focus

Service Mesh	App-Services	DB-Services
CI/CD	DNS	Authentication
Monitoring	xKS Service	Automation
Logging	Registry	Security
Compute	Storage	Network

OpenShift allows your technology teams to focus on building business value, not focusing on building technology platforms

xKS services are NOT free, and not simple

- xKS Services are Kubernetes Cloud lock-in
- 10-20+ individual services needed to make a platform
- Customer must do integrations and maintenance
- All non-xKS Services priced differently
- All non-xKS Services supported individually

Additional costs and integrations

Red Hat

A simpler developer experience

Inner Loop

Ability for code, debug, build and test app code prior to committing to git

Outer Loop

Seamless enablement of git repos and app into CI/CD leveraging DevOps and GitOps principles

Observe

Complete app observability to allow for alerting, metrics, tracking, reporting and dashboards

helvetia

“We looked at alternatives, but we felt that Red Hat OpenShift Container Platform was simpler than it’s competitors, especially in developer experience”

Inner & Outer Loop

End-to-End Secure Software Delivery Lifecycle

GIT COMMIT

RELEASE

Code / Debug — Build and Package — Run → CI Build/Validate — Deploy →

DEVELOPMENT

CONTINUOUS DELIVERY

Simple and Opinionated
"I just want to code"

Powerful and Flexible
"Get out of my way"

Secure

Kube-Native IDE
CodeReady Workspaces Up and coding quickly

Dev-focused CLI
odo Kube support and flexible stacks with devfiles
Web Terminal – browser access to CLIs

Any IDE + OpenShift Plugins

`kubectl` / `oc` / `kustomize` **CLIs**
Powerful Kube control

Kube-Native Build
Provide container build capability with OpenShift S2I, buildah, buildpacks

Application Packaging
Operator Framework provide full automated operational support
Helm – packaging format for simple install handling

OpenShift Serverless
Knative Serverless, Eventing and Functions

OpenShift Service Mesh
Connect, secure, control & observe your services

Health Checks
Easily enable k8s probes and monitor in webui
CodeReady Containers
Dev and run apps locally

Kube-Native CI
OpenShift Pipelines
Tekton - webhooks, persistent workspaces and cred management for Git & image repos, VS Code and Tekton Hub

CI Plugins
- GitHub Actions
- Microsoft Azure DevOps
- GitLab Runner
- Jenkins

GitOps and Continuous Delivery
Kubernetes leading GitOps / CD open source project ArgoCD
Provide opinionated E2E flows

ShiftLeft Security with CodeReady Dependency Analytics

DevSecOps KubeLinter

Physical | Virtual | Private cloud | Public cloud | Managed cloud (Azure, AWS, GCP, IBM, Red Hat) | Edge cloud

OpenShift integrates into your organization's preferred toolchain

GIT COMMIT

RELEASE

Code / Debug → Build and Package → Run → CI Build → Deploy

DEVELOPMENT

CONTINUOUS INTEGRATION

JET BRAINS

Visual Studio Code

eclipse

Eclipse Che

HELM

ANSIBLE

docker

OPERATOR FRAMEWORK

Maven

Gradle

kubernetes

Red Hat OpenShift

okd

docker

TC

Bamboo

Travis CI

circleci

GitLab

TEKTON

argo

weaveworks

snyk

cloudbees

Bitbucket

Microsoft Visual Studio Team Foundation

GitHub

GitLab

QUAY by CoreOS

JFrog Artifactory

Sonatype Nexus

Red Hat

Now I'm gonna show you how to use the developer perspective,

So that you keep the overview and easily navigate the many components of your applications.

SEE SPEAKER NOTES

Red Hat

Options for developing with OpenShift using a desktop IDE

Desktop IDE + CodeReady Containers

Ideal for power users who are already comfortable with Kubernetes.

Coding:

- Eclipse, JetBrains or VSCode IDE
- Add plugins to interact with OpenShift, Kafka, Camel, Quarkus, Kubernetes YAML, etc…

Run, build and test:

- Run CodeReady Containers on the laptop to have full admin control of your own local OpenShift cluster

Desktop IDE + secure local containers

Ideal for transitioning to secure containers built and run locally on the laptop.

Coding:

- Eclipse, JetBrains or VSCode IDE
- Add plugins to interact with OpenShift, Kafka, Camel, Quarkus, Kubernetes YAML, etc…

Run, build and test:

- Run Podman, Builah and Skopeo for secure build and run, or docker for unsecure builds

Desktop IDE + remote OpenShift

Ideal for eliminating "it works on my machine" by using OpenShift during dev.

Coding:

- Eclipse, JetBrains or VSCode IDE
- Add plugins to interact with OpenShift, Kafka, Camel, Quarkus, Kubernetes YAML, etc…

Run, build and test:

- Run, build and test in a private sandbox in a connected OpenShift cluster, or using CodeReady Workspaces

Red Hat

Opportunity: Speed Project Onboarding

Organizations suffer because of how long it takes for consultants, offshore developers and others to become contributors to a project

Instant onboarding for consultants and remote developers means increased velocity and market impact

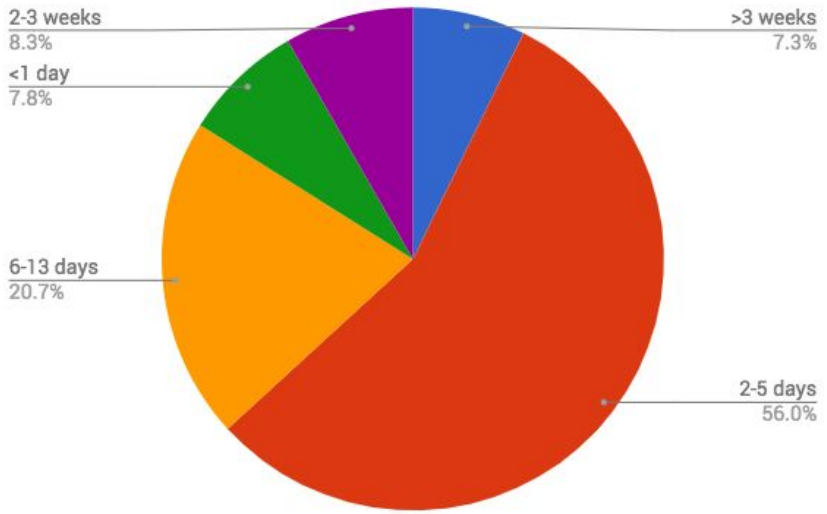
Challenges

- **Consultants often take 5-10 days to get setup** on internal systems
  - Get approved laptop
  - Get access to relevant tools
  - Get access to shared dev clusters
  - Struggle with slow VDI speeds
- **Remote development offices are inefficient** due to high turnover and ongoing setup and ramp-up
  - VDI solutions are slow and painful for developers: makes them less agile and effective

Goals should be:

- **Near-instant provisioning**
- **Quickly reproduce production environment in development**

Average Time To First Project Contribution*



* Red Hat survey of ~300 enterprise customers (using a mix of VMs, containers, and bare metal in clouds and data centers)

Red Hat

CodeReady Workspaces: reduce onboarding from weeks to minutes

Teams can write container-native apps in OpenShift,
with zero Linux and Kubernetes knowledge.

DevOps Integrations

Reference developer workspaces from any issue, failed build, or git notification.

Accelerate developer onboarding

Protect Source Code

Full access to source code without any of it landing on hard-to-secure laptops.

Keep your IP safe on IT-controlled servers

Container Workspaces

Workspace replicas to end “works on my machine” and enable team collaboration.

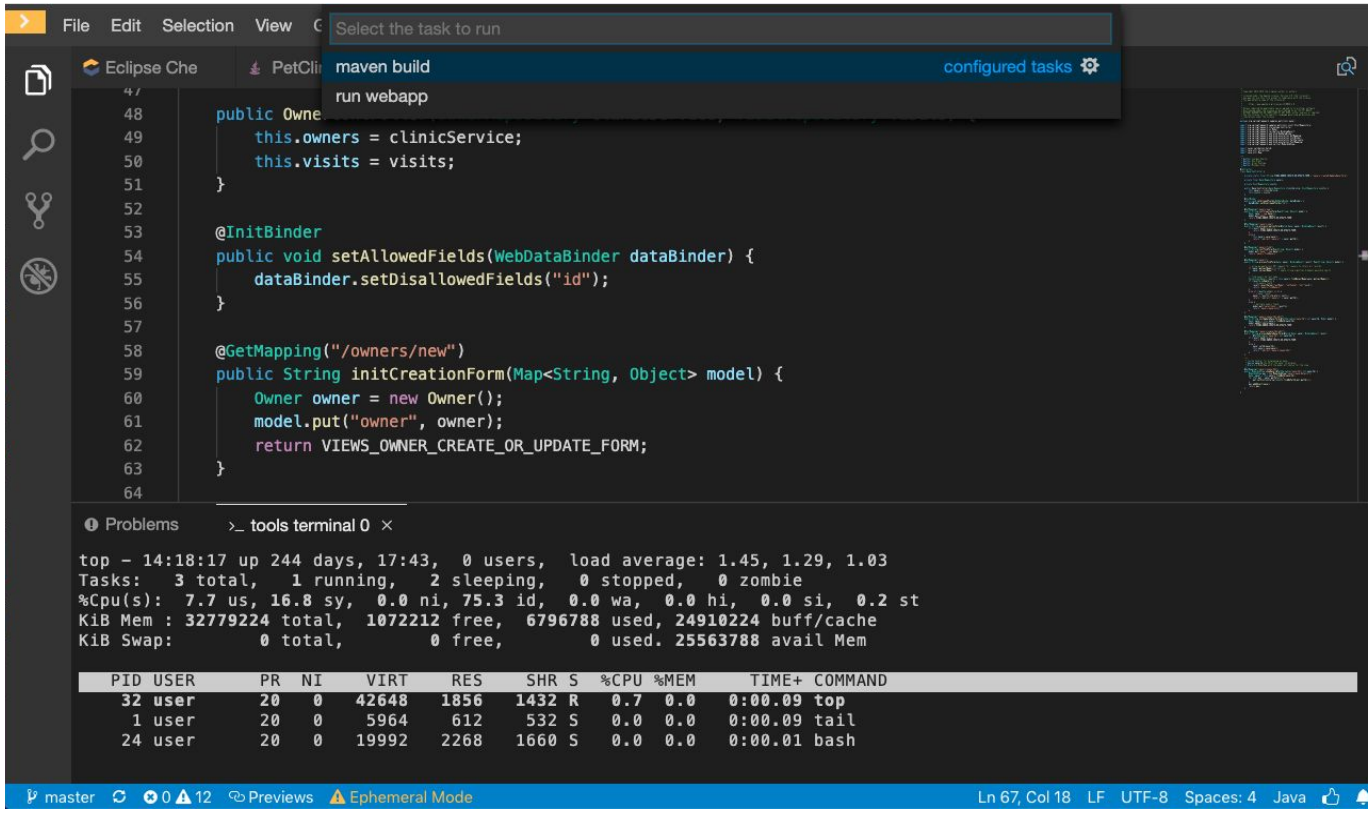
Remove laptop-to-laptop inconsistencies

Red Hat

CodeReady Workspaces creates a containerized developer environment in Kubernetes - requires no Kube knowledge

Project sources
Dependencies
Developer Tools
Commands
Build and packaging tools
Terminal
Operating system
Web server / application server
Database
(All other runtime components)

Everything a developer needs is managed in a personal Workspace hosted in an IT-Managed OpenShift cluster.



1. Accelerates projects and onboarding of developers.
2. Removes inconsistencies between dev and prod.
3. Protects source code by keeping it off laptops.

Available for OpenShift on x86, and OpenShift on IBM Z mainframe

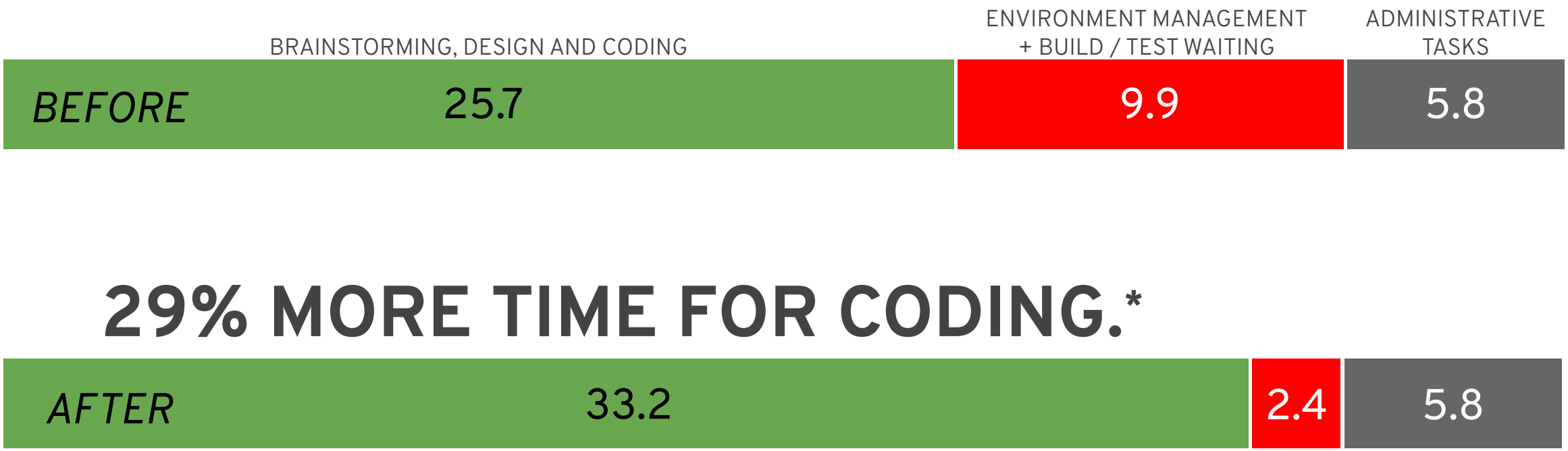
Red Hat

Red Hat CodeReady Workspaces offerings for IBM Z & Power

	Red Hat CodeReady Workspaces (on Z)	**IBM Wazi for Red Hat CodeReady Workspaces**
Quick description…	CodeReady Workspaces running on z/Linux (and later Power)	CodeReady Workspaces for developing Z apps
Built from…	Eclipse Che project	CodeReady Workspaces product
Runs on…	OpenShift (x86, z/Linux, Power)	OpenShift (x86, z/Linux, Power)
Targeted at…	Cloud Pak for Apps customers who want to run CRW on z/Linux	IBM customers who want to use CodeReady Workspaces to develop apps for Z
What's included?	Identical offering to CRW, just running on z/Linux	CRW plus proprietary extensions for COBOL and PL/I language services, debugging and execution
Pricing	Free with OpenShift (and ICP4A)	Charged add-on
Sales channel	Sold by Red Hat or IBM channels	Sold by IBM channel
Target Availability	July 2020	Announced May 2020 (Targeted GA 19-June-2020)

Red Hat

OpenShift and Red Hat CodeReady Workspaces allow teams to spend time coding, not maintaining their workbench



29% MORE TIME FOR CODING.*

** Time tracking of internal Red Hat teams over 6 month period (3 months before using CodeReady Workspaces, 3 months after)*

Red Hat

Education opportunities for developers

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

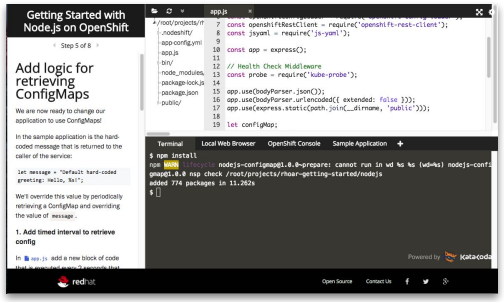
Free access to downloadable and SaaS software for use in development and pre-production environments.

Red Hat DevNation

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

Industry Leading Experts

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



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

DevNation

Dev to the power of Ops.

DevNation brings the expertise of top open source contributors to topics like Java, Kubernetes, Tekton, Kafka, Istio, Quarkus and more. It includes articles, TechTalks, Master Courses and full-day in-person events.

DevNation has delivered over 30 TechTalks to over 20K attendees and live events around the world.

DevNation is going fully virtual for 2020; check out the link above for added offerings every month.

Dev to the power of Ops

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

Red Hat

Appendix:
Details on Red Hat's Developer Tools

Red Hat

Developer Services

OpenShift “Dev” console for self-service

`odo` : OpenShift’s dev-focused CLI

IDE plugins

CVE checking in the IDE (DevSecOps)

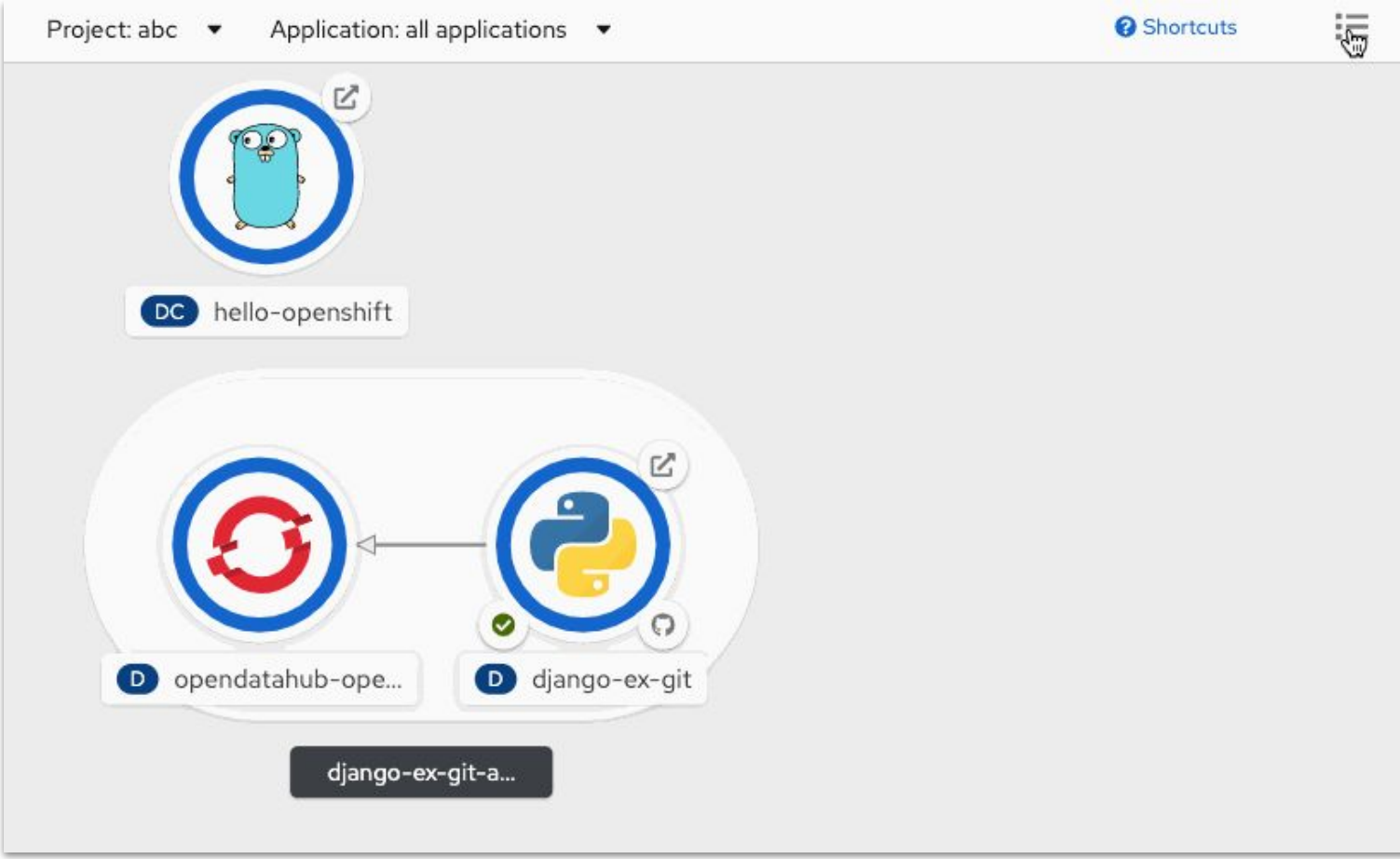
Red Hat

OpenShift Console: Developer perspective and DevOps capabilities

A DevOps console with developer and admin perspectives:

- Create apps from git, images, etc…
- See live application topologies
- Create and track CI pipelines
- Scale up/down in a single click
- Monitor app health and metrics
- Link to more detailed admin views

A single, integrated UI to facilitate the move to DevOps.



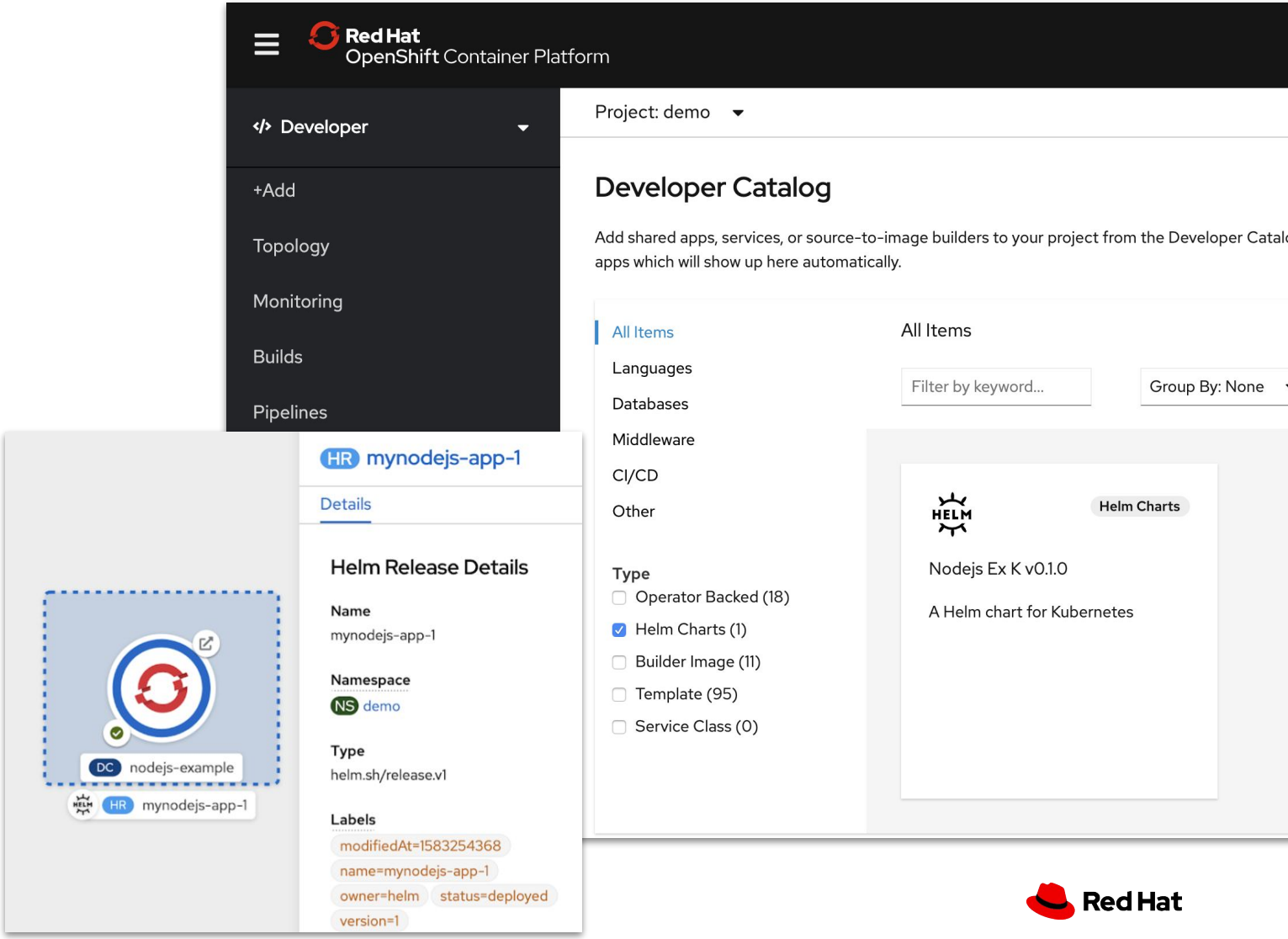
Helm v3: First Class Support in OpenShift

OpenShift 4.3

- Helm 3 CLI in Tech Preview
- Built and shipped with OpenShift
- Available in Console CLI menu

OpenShift 4.4+

- Helm 3 in the OpenShift console
  - Charts in developer catalog
  - Releases in dev console
  - Update / rollback / delete
- Helm developer guides



Red Hat

Helm and Operators

Package and Install

Automated Day-2 Operations

Helm

Operator

Phase I	Phase II	Phase III	Phase IV	Phase V
Basic Install	Seamless Upgrades	Full Lifecycle	Deep Insights	Auto Pilot
Automated application provisioning and configuration management	Patch and minor version upgrades supported	App lifecycle, storage lifecycle (backup, failure recovery)	Metrics, alerts, log processing and workload analysis	Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

Red Hat

OpenShift's developer-focused CLI: "odo"

A developer-focused command-line tool for rapid development iterations on OpenShift.

Simplifies building of microservices applications on OpenShift.

Uses a familiar "git push" style syntax that makes it an easier transition for ex-Cloud Foundry users.

```
$ odo create wildfly backend
Component 'backend' was created.

$ odo push
Pushing changes to component: backend

$ odo create php frontend
Component 'frontend' was created.
To push source code to the component run 'odo push'

$ odo push
Pushing changes to component: frontend

$ odo url create
frontend - http://frontend-myapp.192.168.99.100.nip.io

$ odo watch
Waiting for something to change in /dev/frontend
```


Red Hat

>6M developers use Red Hat IDE extensions

Java, Kubernetes, Knative, Tekton, Quarkus, Camel, and more…

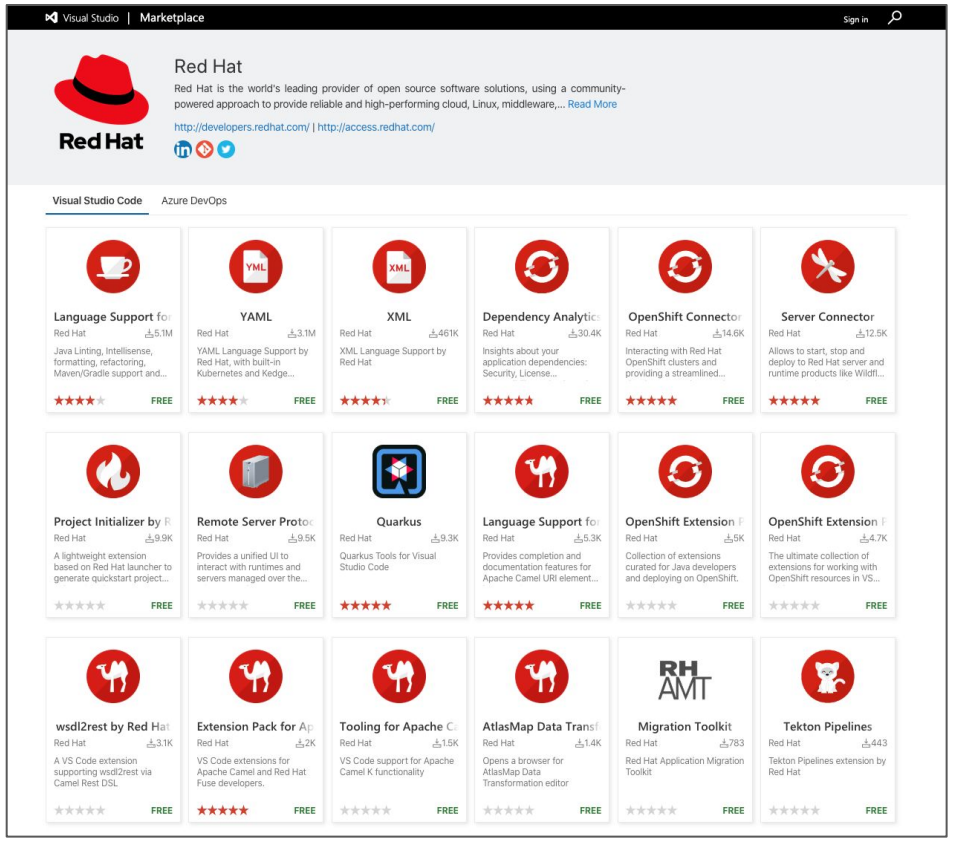
Creating a great experience for Red Hat projects and products across VSCode, JetBrains and Eclipse, builds our affinity and reputation with this critical audience.

New → Quarkus: Write Java with Quarkus to build low memory, fast-startup functions and microservices that are as lightweight as NodeJS or Golang. Over 15k users already!

New → Tekton: Build, preview, and control tasks and pipelines from the IDE. Use it with OpenShift Pipelines or Tekton.

New → Knative: Create and interact with Knative and OpenShift Serverless containers and deployment artifacts.

Coming Soon → Operators: In-IDE assistance, skaffolding and error checking for developers building L1 to L5 Kube Operators.



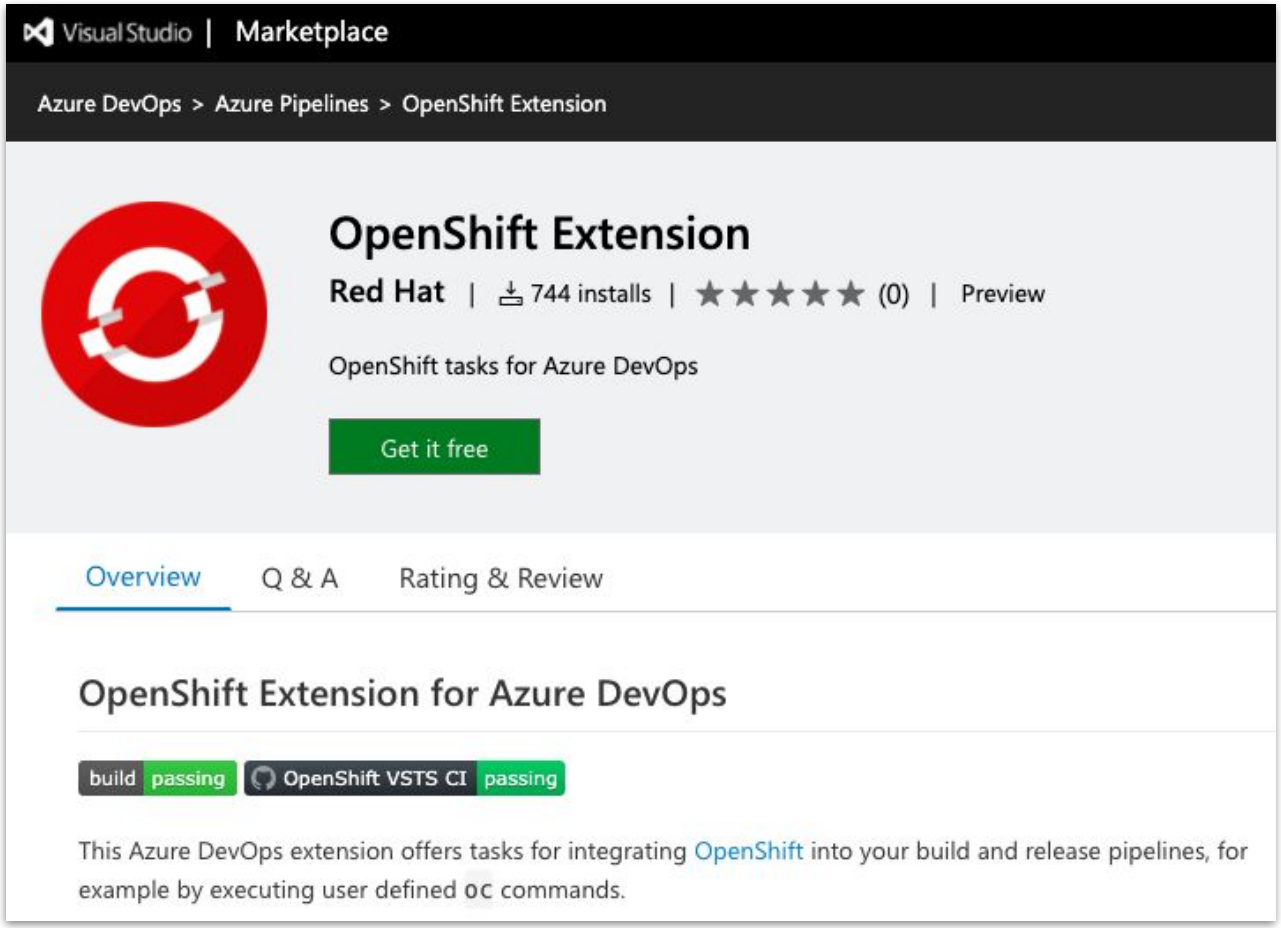
Red Hat

OpenShift plugin for Microsoft Azure Devops and Team Foundation Server

An extensions that offers tasks for integrating OpenShift into your build and release pipelines whether you're using Azure DevOps (cloud) or Team Foundation Server (on-premises).

Developers can deploy to any OpenShift:

- On-premises
- In the cloud
- Hosted by Red Hat
- On the Azure Red Hat OpenShift service
- etc…



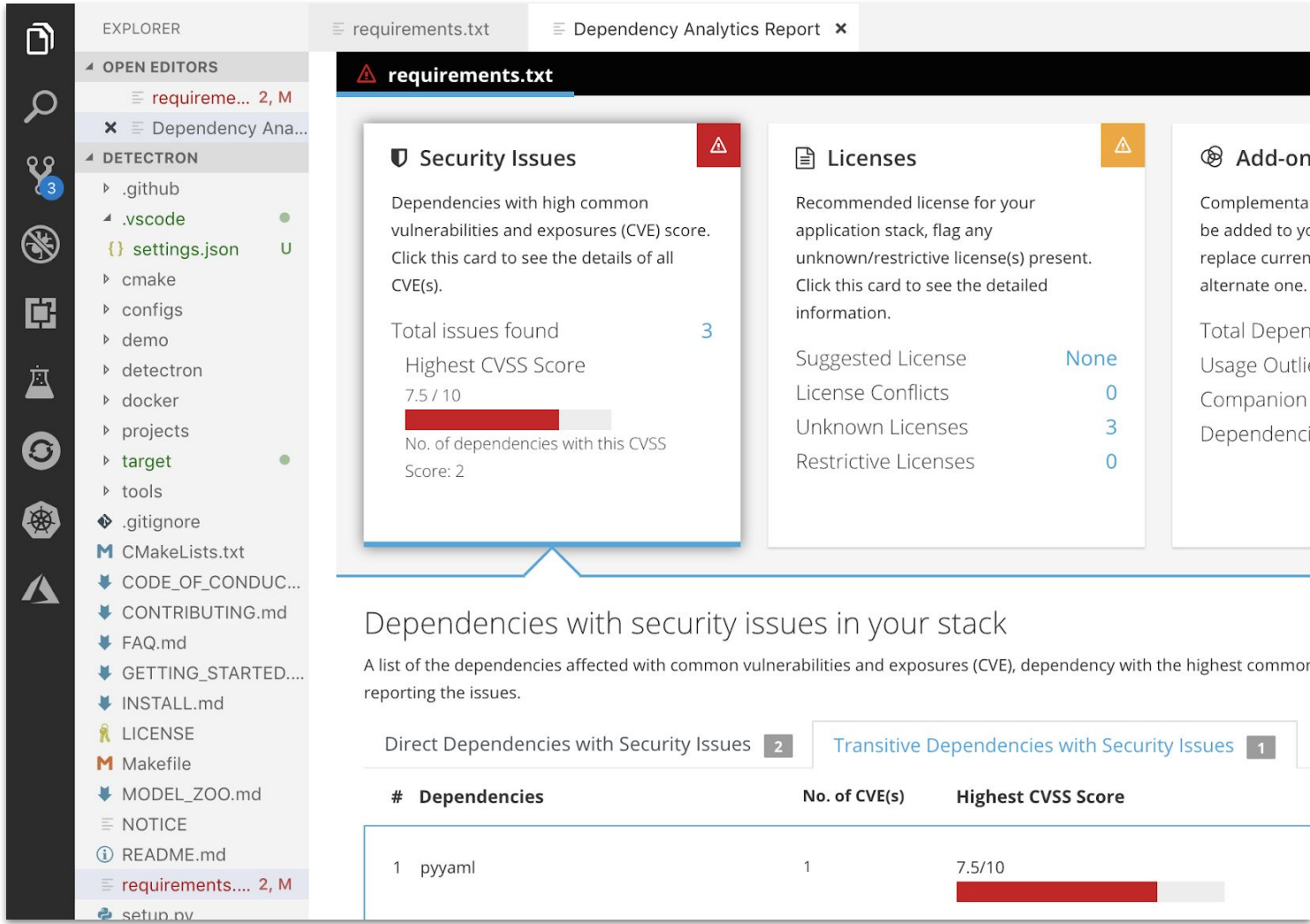
Red Hat

Enhancing Secure Application Development and DevSecOps

“Shift Left” - find CVEs and license issues during development

Red Hat Dependency Analytics IDE plugins provide security and license warnings for any project dependency:

- Be notified of CVEs in any package or sub-package
- Remediation advice (upgrade / downgrade)
- Uses open source and Snyk CVE databases
- Supported for Java, Node, Python



Red Hat CodeReady Toolchain

A single OpenShift Operator that configures a suite of developer tools and services, to support DevOps for applications on OpenShift

App Dev Experience	Dev Tools & Services	Open Integration
Streamlined application delivery natively with Dev Console and IDE plug-ins	Set of leading open source tools & services for rapid dev of cloud native applications	Plug-in 3rd party dev tools to customize the toolchain for your needs
Focus on delivering business value	Create, build, test, deploy cloud-native applications	Leverage existing investment in tools and processes

Based on OpenShift

Product Manager: Parag Dave

OpenShift Platform Services for Applications

OpenShift Serverless and Service Mesh

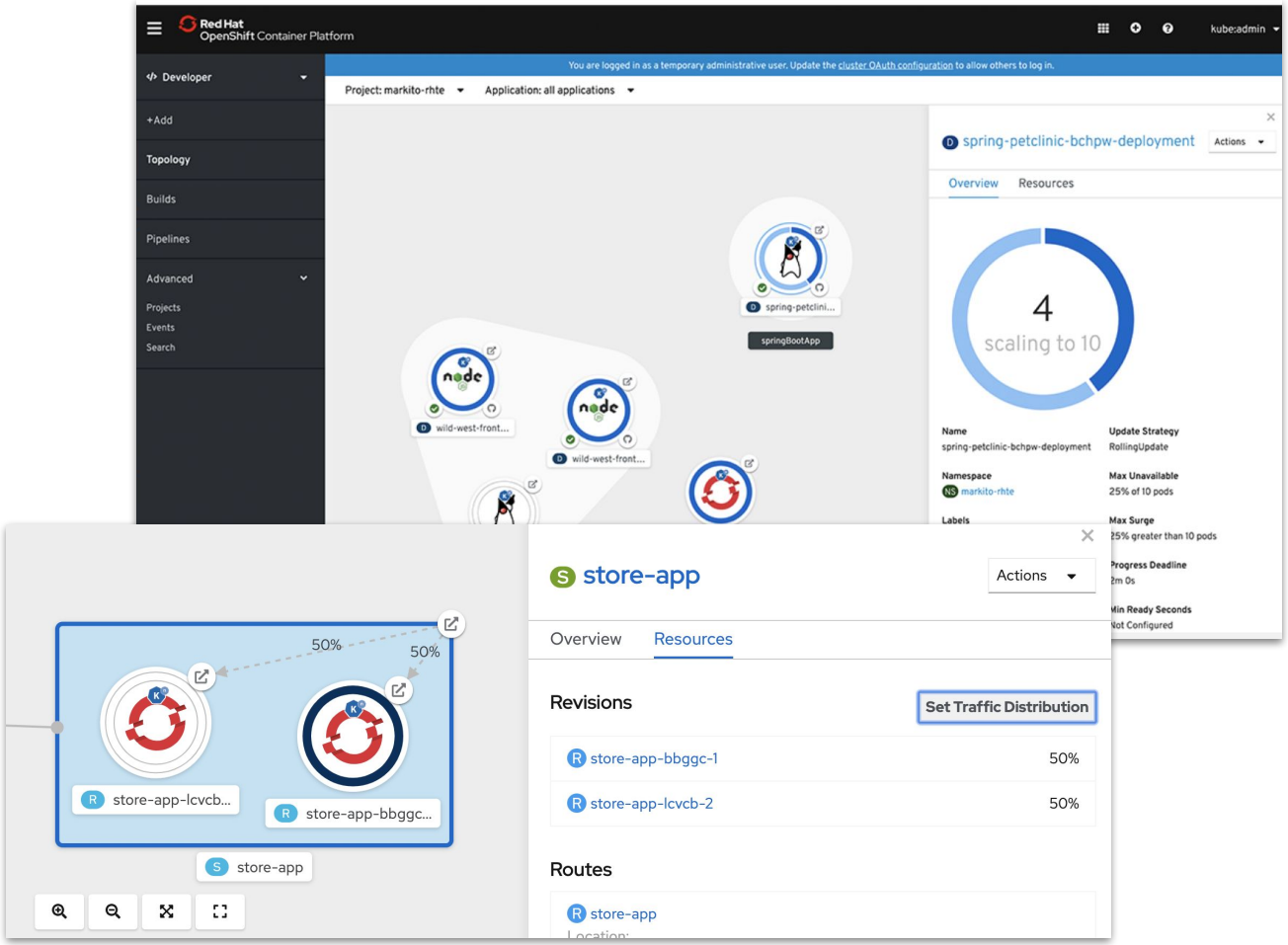
OpenShift Builds

OpenShift Pipelines

Red Hat

OpenShift Serverless: Scale to zero, fewer resources, and traffic splitting

- Based on **Knative**: familiar to Kube users; no vendor lock-in
- Deploy any container workload as serverless (applications and functions)
- Scale to 0, and autoscale to N
- Multiple event sources for flexibility
- Visualize event triggers and relationships
- Split traffic between app versions
- Simple to install through a Kubernetes Operator on OperatorHub

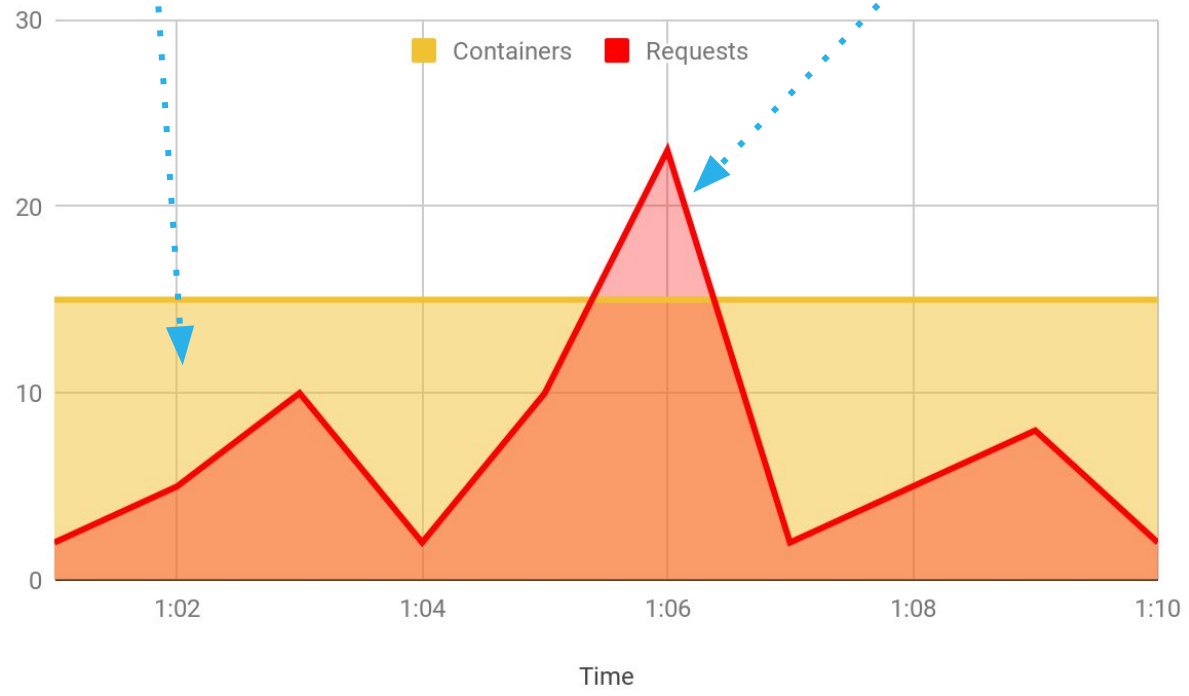


Red Hat

Serverless Operational Benefits

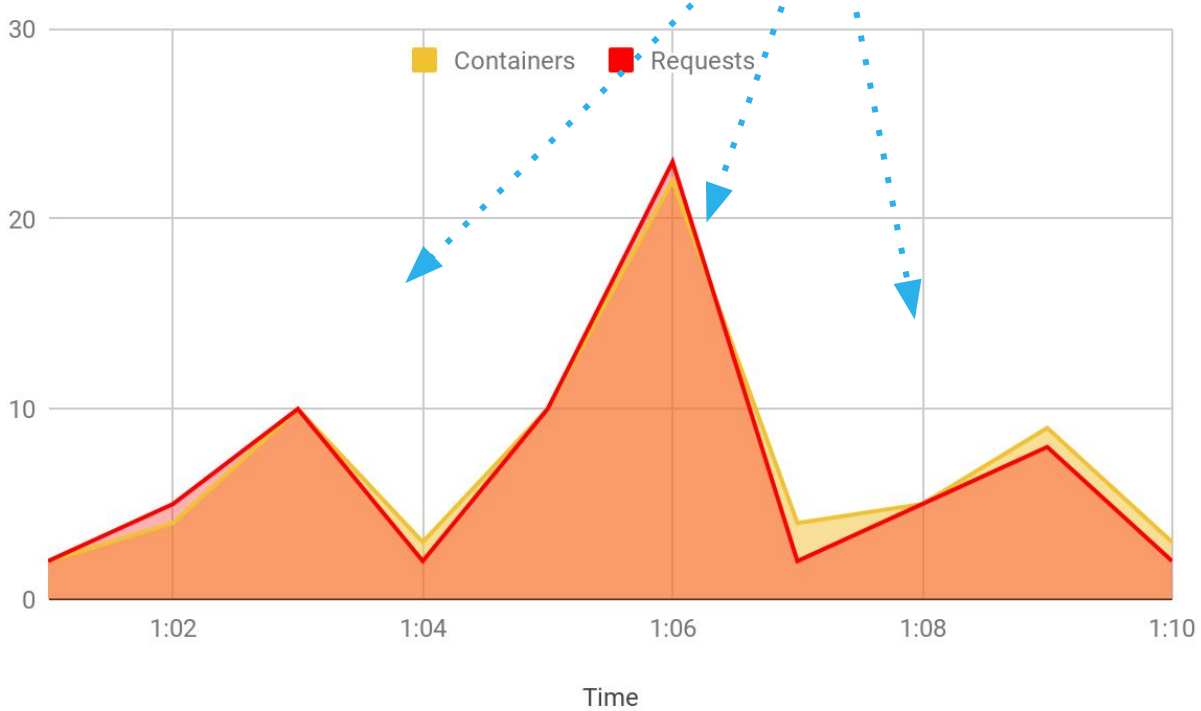
Over provisioning
Time in capacity planning
IT cost of idle resources

Under provisioning
Lost business revenue
Poor quality of service



NOT Serverless

More applications
Direct line between IT costs & business revenue



with Serverless

Red Hat

OpenShift Serverless

Red Hat OpenShift Container Platform

William Oliveira

Developer

+Add

Topology

Builds

Pipelines

Advanced

Project: serverless-demo

Application: all applications

Shortcuts

CJS my-cron

100%

REV kiosk-kghyl-2

KSVC kiosk

50% 50%

REV frontend-ycfbg-2

REV frontend-hr7jf

KSVC frontend

kiosk-app

kn service create store-app --image=markito/store-app:v1

```
markito@anakin  /tmp
$ kn service create store-app --image=markito/store-app:v1
```

KSVC frontend

Actions

Overview Resources

Revisions

Set Traffic Distribution

REV frontend-hr7jf 50%

D frontend-hr7jf-deployment 0

REV frontend-ycfbg-2 50%

D frontend-ycfbg-2-deployment 0

REV frontend-rjlnt-3

D frontend-rjlnt-3-deployment 0

Routes

RT frontend

Location:

http://frontend.serverless-demo.apps.devint.openshiftknativedemo.org

Red Hat

OpenShift Builds

Build lean images from application source code and binary using Kubernetes tools on OpenShift

Build slim runtime images without the build dependencies

Use Kubernetes build tools (e.g. buildah, S2I, CNB[1], Kaniko, etc)

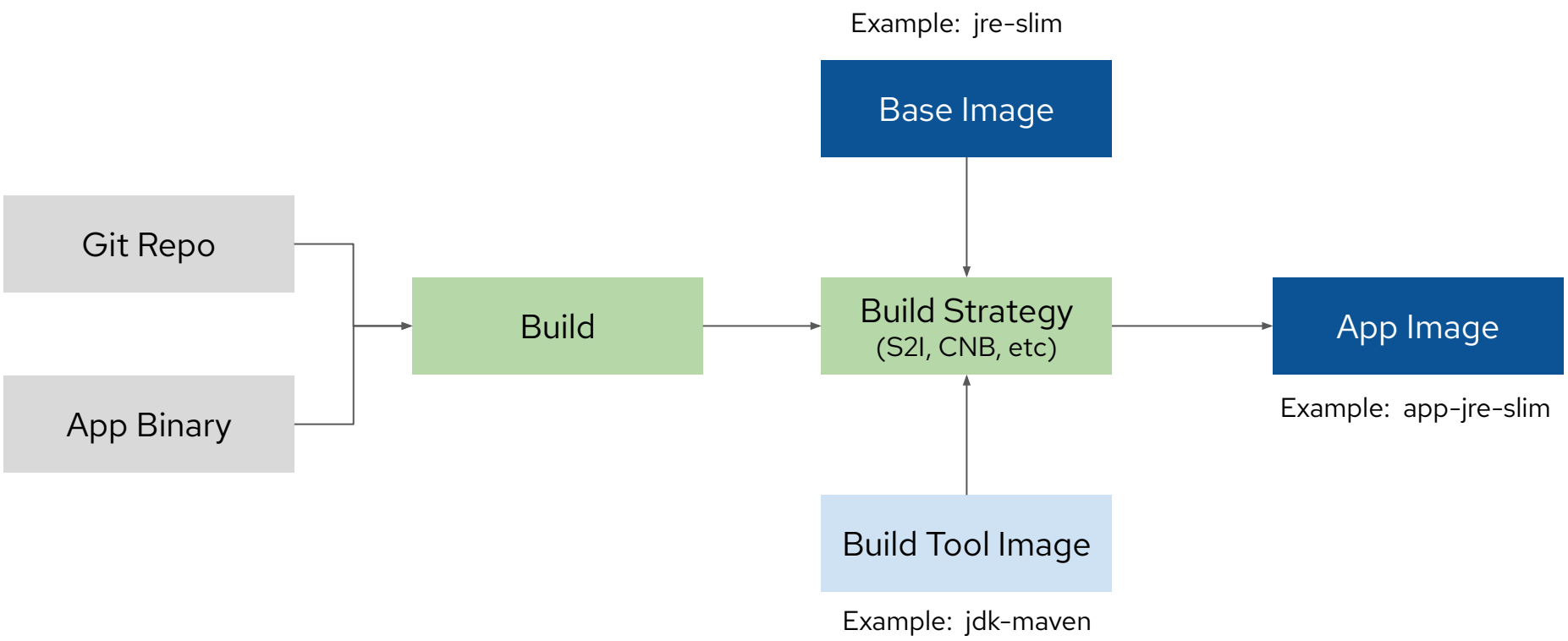
Extensible and customizable with your own build tools

Portable builds to any Kubernetes platform

[1] Cloud-Native Buildpacks

Red Hat

OpenShift Builds



OpenShift Builds

Cloud-Native Buildpacks

```yaml
kind: Build
metadata:
  name: myapp-cnb-build
spec:
  source:
    url: https://github.com/myorg/myapp
  strategy:
    name: buildpacks-v3
  builder:
    image: heroku/buildpacks:18
  output:
    image: quay.io/myorg/myapp:v1
```

Source-to-Image (S2I)

```yaml
kind: Build
metadata:
 name: myapp-s2i-build
spec:
 source:
   url: https://github.com/myorg/myapp
 strategy:
   name: source-to-image
 builder:
   image: registry.redhat.io/openjdk/openjdk-11-rhel8
 output:
   base: docker.io/openjdk:11-jre-slim
   image: quay.io/myorg/myapp:v1
```


Red Hat

OpenShift Pipelines

Cloud-Native CI/CD with Tekton on on OpenShift

Kubernetes-native declarative Pipelines with Tekton

Serverless CI/CD with no single server to share and maintain

Run pipelines in isolated containers with all required dependencies

Standard and portable to any Kubernetes platform

Web, CLI, and Visual Studio Code and IDE plugins

Red Hat

OpenShift Pipelines

Red Hat OpenShift Container Platform

kube:admin

Developer

+Add

Topology

Monitoring

Builds

Pipelines

More

Project: demo

Pipeline Runs > Pipeline Run Details

Dev Preview

Actions

PLR petclinic-deployment-pipeline-vivla6 Running

Details YAML Logs

Pipeline Run Details

unit-tests

build-image

deploy

slow-unit-te...

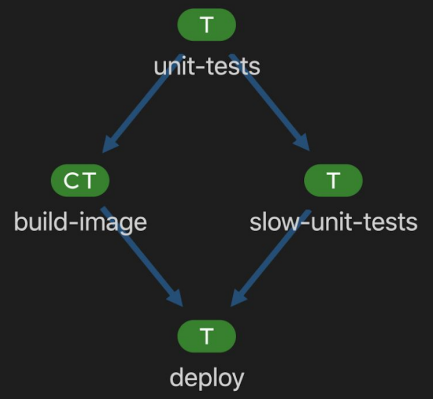
Name

pipeline.yaml — pipelines-examples

console-categories.txt | simple-task.yml | pipeline.yaml | Preview pipeline.yaml

private > tmp > pipeline.yaml > {} spec > [ ] tasks > {} 2 > [ ] params > {} 0 > [ ] value

```yaml
    resources:
      inputs:
      - name: source
        resource: app-git
      outputs:
      - name: image
        resource: app-image
  - name: slow-unit-tests
    taskRef:
      name: mvn
    runAfter:
      - unit-tests
    resources:
      inputs:
      - name: source
        resource: app-git
    params:
    - name: GOALS
      value:
        - verify
  - name: deploy
    taskRef:
      name: oc-redeploy
    runAfter:
      - build-image
      - slow-unit-tests
    params:
    - name: DEPLOYMENT
      value: spring-petclinic
```



Red Hat OpenShift Container Platform

Developer

+Add

Topology

Monitoring

Builds

Pipelines

More

Project: demo

Pipeline Runs > Pipeline Run Details

PLR petclinic-deployment-pipeline-vivla6 Running

Details YAML Logs

unit-tests

slow-unit-tests

build-image

build-image

```
STEP-GENERATE

  {"level":"info","ts":1585670155.8098009,"logger":"fallback-logger","caller":"logging/config.g
  Application dockerfile generated in /gen-source/Dockerfile.gen

STEP-BUILD

  {"level":"info","ts":1585670174.8911083,"logger":"fallback-logger","caller":"logging/config.g
  STEP 1: FROM registry.access.redhat.com/redhat-openjdk-18/openjdk18-openshift
  Getting image source signatures
  Copying blob sha256:6c85ac87d44df4b64d7c273886fc5aed55a28422df33dcb641884ffa419db218
  Copying blob sha256:51e9f237b750efcda2d5755785cdb8dd080d51585ae35d368e4f9b29a11b1994
  Copying blob sha256:41bc10dbb03d6e2f00d82a33d2fdfca62e3a50cd17b8efdfa59cbeb650e278af
  Copying config sha256:fc8de886c1471711ed33a0c1ab46143e433e02870d342b00683be0175a92a64f
  Writing manifest to image destination
  Storing signatures
```

Red Hat

Jenkins CI/CD in OpenShift

Jenkins is still the most used CI/CD platform in enterprises and can be used from inside OpenShift.

An intuitive pipeline visualization makes it simple for users to see how builds are progressing.

The full Jenkins UI is also available.

Red Hat

OpenShift Service Mesh

Enhanced Application Flexibility through Istio, Kiali, and Jaeger

Enhanced gRPC support for efficient microservice development

Managed Updates to the Control Plane via Operator Driven Deployment

Enhanced Authorization Traffic Controls

Increased visibility with new Kiali and Jaeger functionality

Increased Traffic Management Capabilities

Red Hat

Time-range selection in Kiali

OpenShift Service Mesh

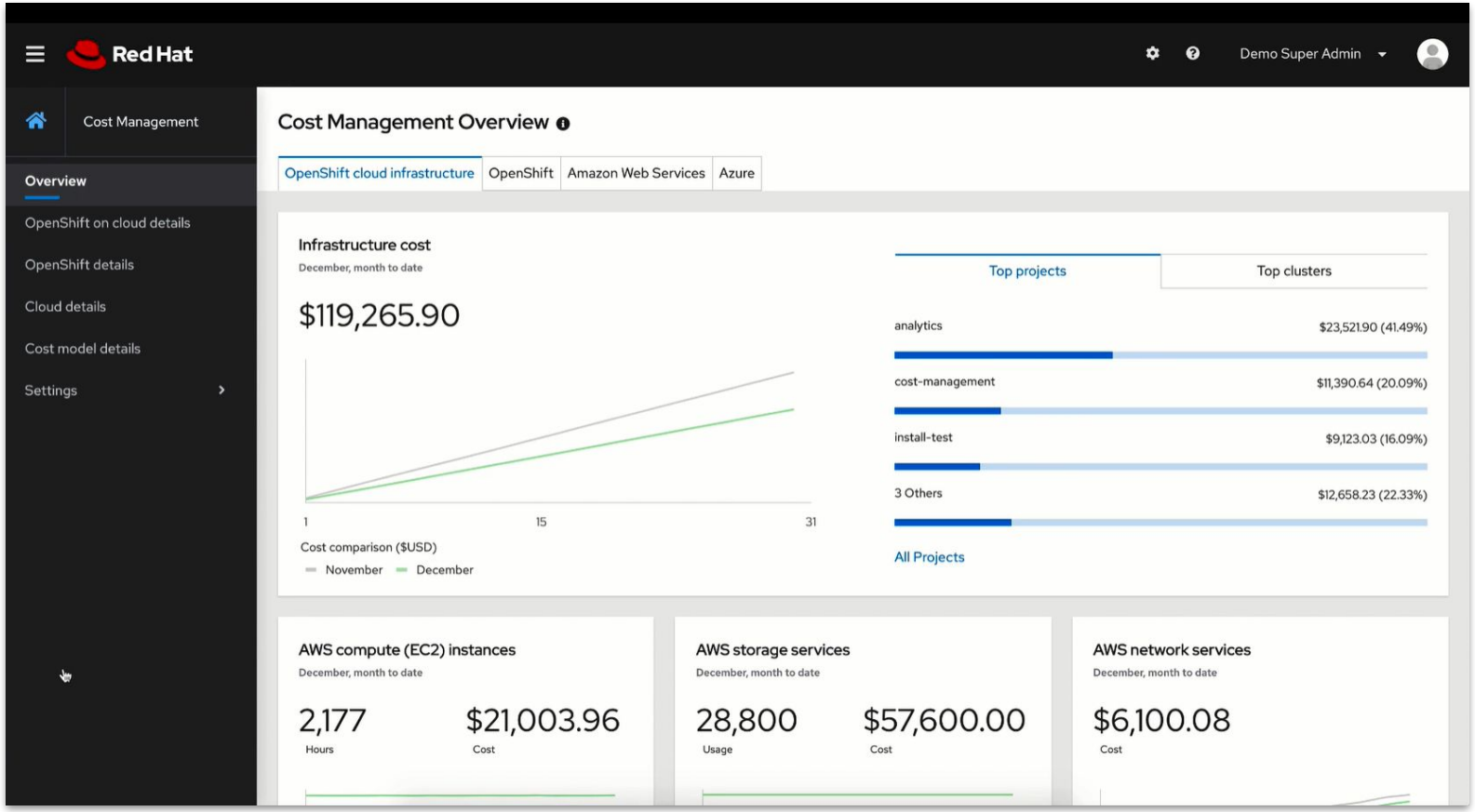
Istio Enhancements in Detail

- Streamlined mutual TLS setup
- Improved troubleshooting with istioctl analyze
- Mirroring a percentage of traffic
- Automatic rotation of expired root certificates managed by Citadel
- Authorization Policy graduating to beta
- Adoption of OpenAPI v3 schemas for Istio CRDs
- More efficient Regular Expression handling for header checks

Cost Management for OpenShift

Provide cost visibility across applications and clusters, both on-premises and in clouds

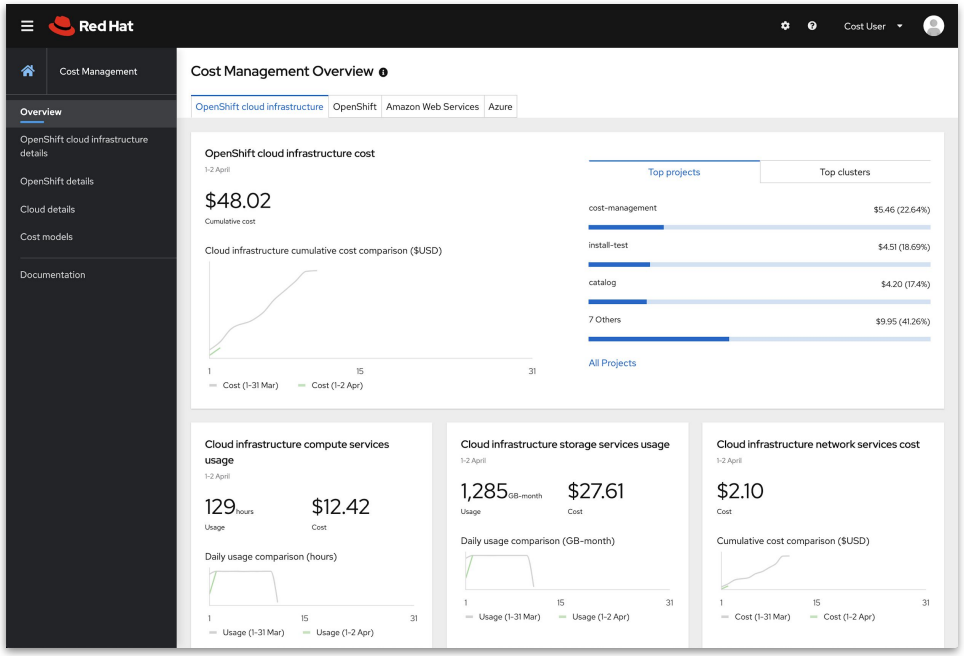
- Visualize costs across hybrid cloud infrastructure
- Track cost trends
- Map charges to projects and organizations -- the most expensive applications may be ripe for modernization!
- Normalize data and add markups with cost models
- Generate showback and chargeback information



Red Hat

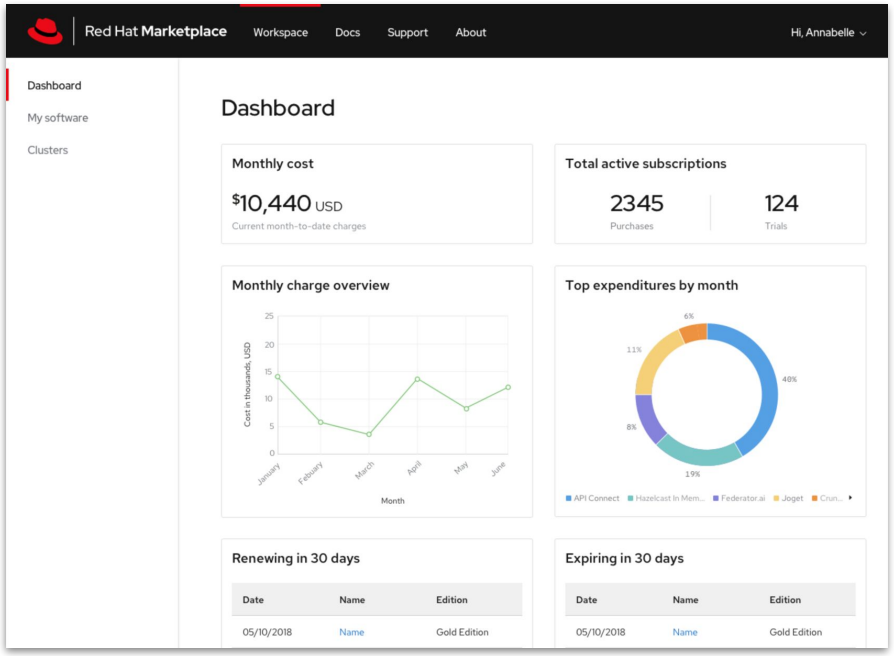
Metering & Chargeback

Red Hat Cost Management



- Cost visibility for all stakeholders
- Cost mapping to teams
- Combines cloud IaaS costs and Red Hat subscription usage

Red Hat Marketplace



- Install and track software usage across clusters
- Developers can find the right certified software solutions quickly in one place
- Simplify approvals, delegation—with visibility and control over software use

Red Hat