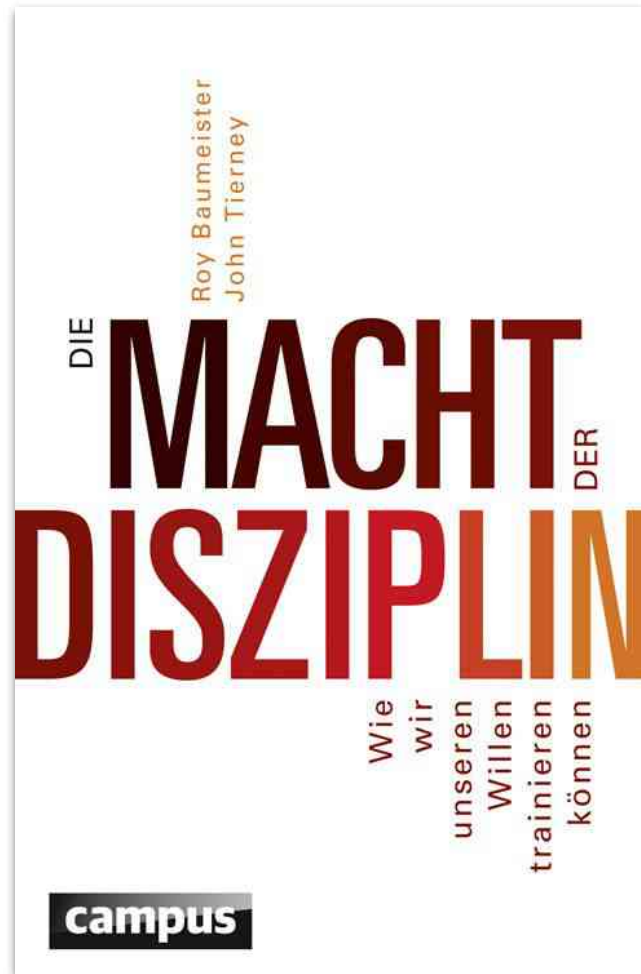


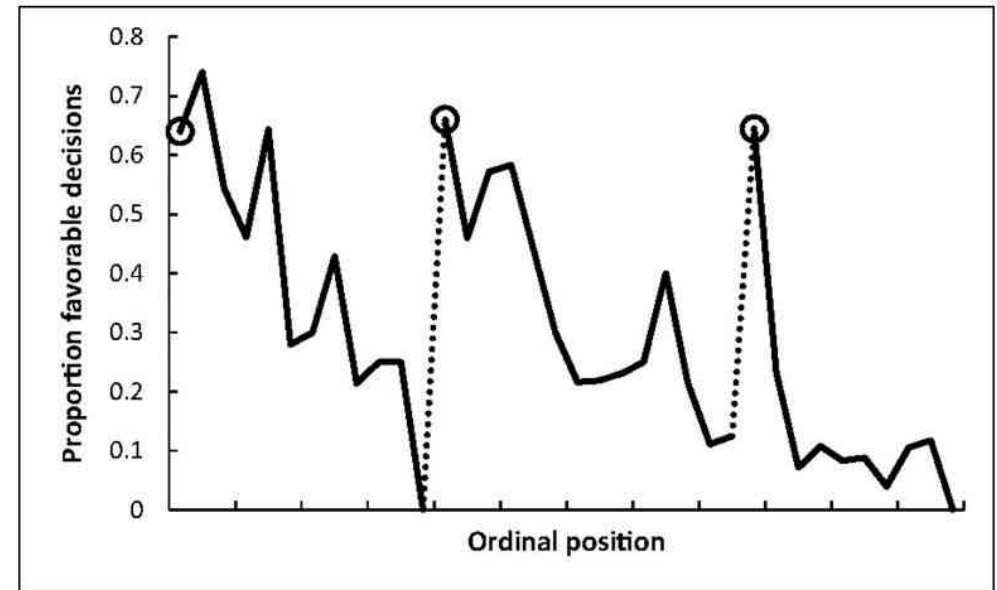


Open.Tour

OpenShift for developer
productivity



Was trennt uns von...



Proportion of rulings in favor of the prisoners by ordinal position. Circled points indicate the first decision in each of the three decision sessions; tick marks on x axis denote every third case; dotted line denotes food break. Because unequal session lengths resulted in a low number of cases for some of the later ordinal positions, the graph is based on the first 95% of the data from each session.

der Mittagspause :)



Karsten Gresch

Specialist Solution Architect (AppDev)

- Background: Lawyer and coder (since age 14)
- Positions: Advocate, developer team manager, business analyst, Java (EE) developer, architect, consultant, systems engineer, solution architect (present)
- In IT since 1998
- Technical interest: Java (from 1.1)/J*EE, Go, Swift, JavaScript (Fullstack), Python, Rust et al.
- Skills: OOP, UML, DDD, UML, RUP, XP, Clean Code, Six Sigma, CI/CD, Reactive, EDA et al.
- Private: Berlin, married, five kids

"To be the catalyst in communities of customers, contributors, and partners creating better technology the open source way."

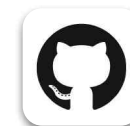


Red Hat's Mission Statement

YouTube Interview (German):
<https://youtu.be/nqxsLkc6bQA>



@gresch



karstengresch

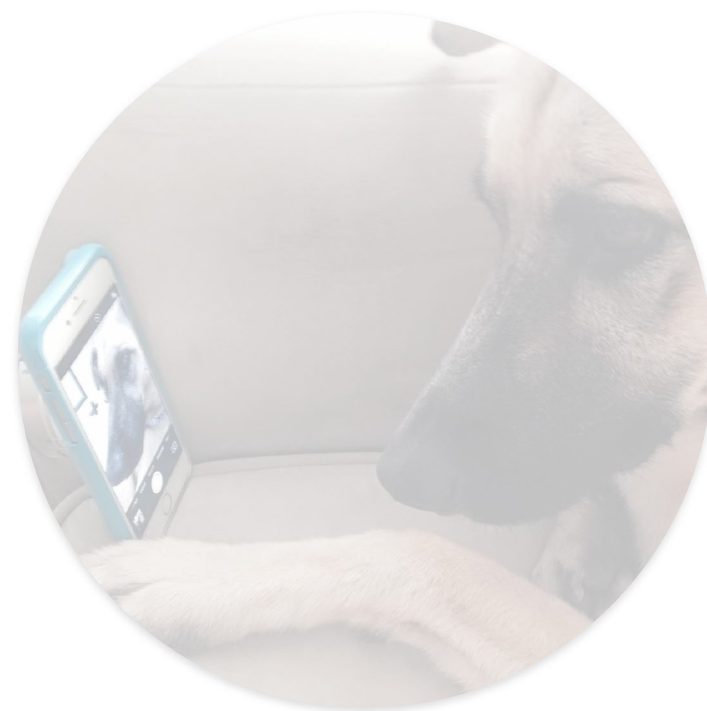
Was?



Warum?



Wie?



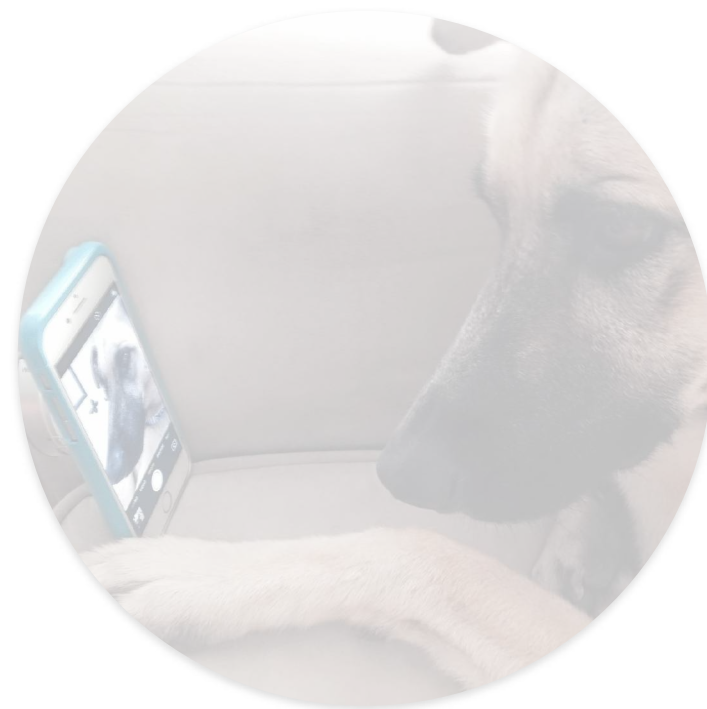
Was?



Warum?



Wie?



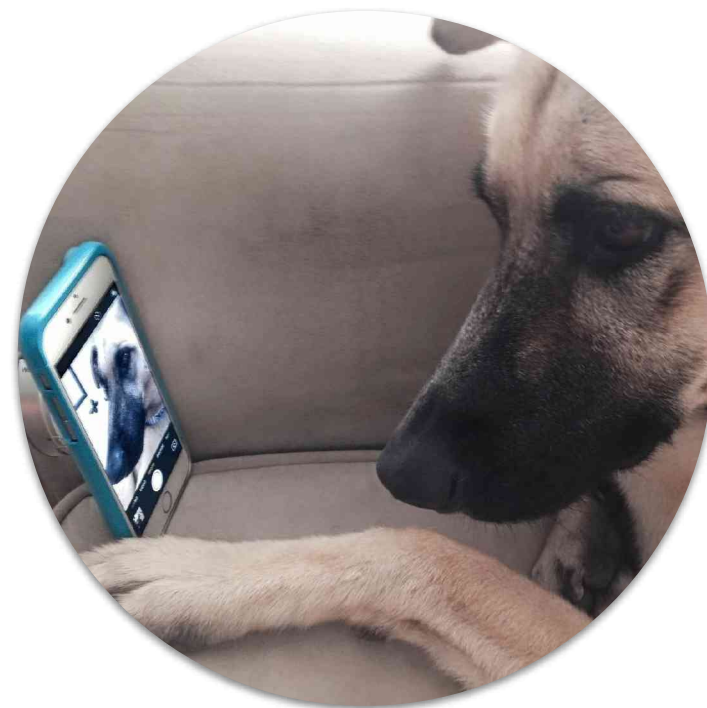
Was?



Warum?



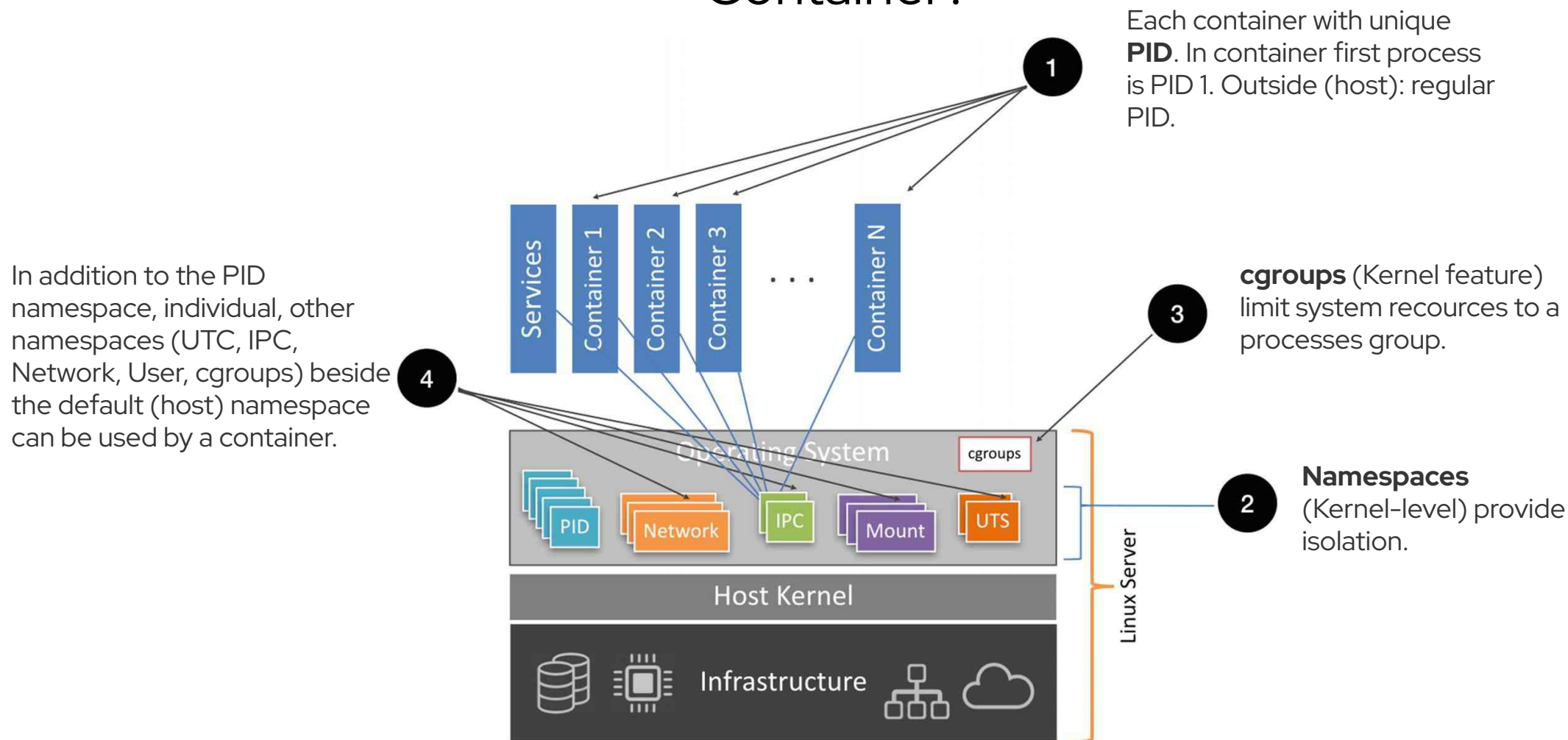
Wie?



Was?



Container?



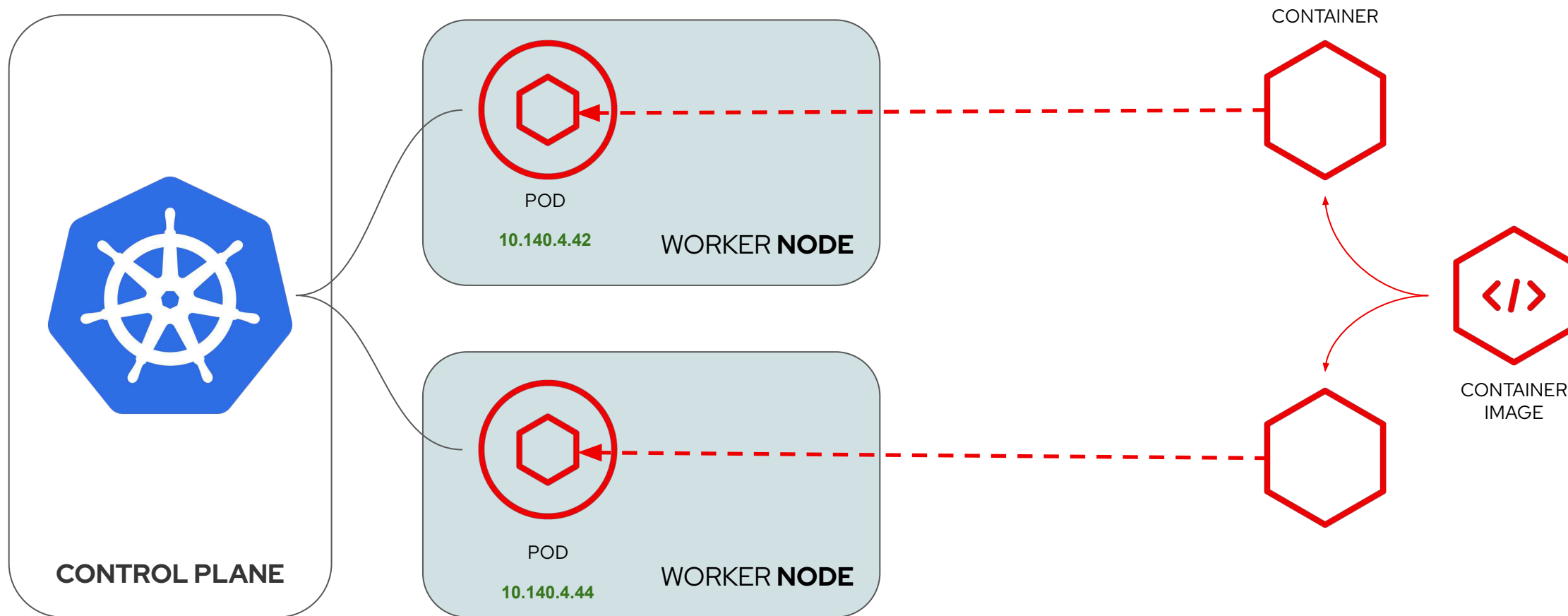


kubernetes?

κυβερνήτης

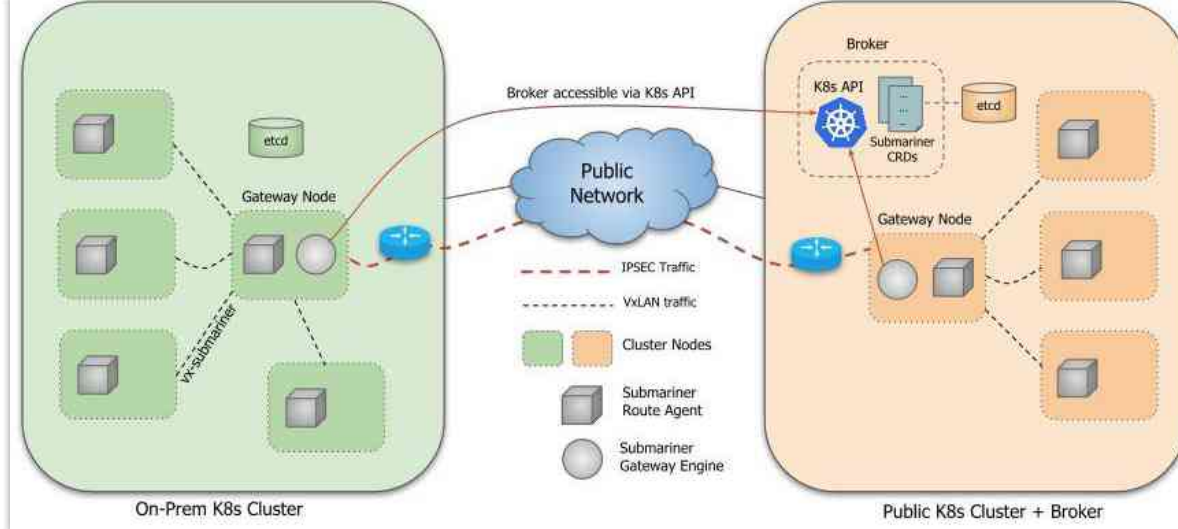
Image by garret parker,
<https://unsplash.com/photos/aYHzEnSEH-w>

Orchestrieren?

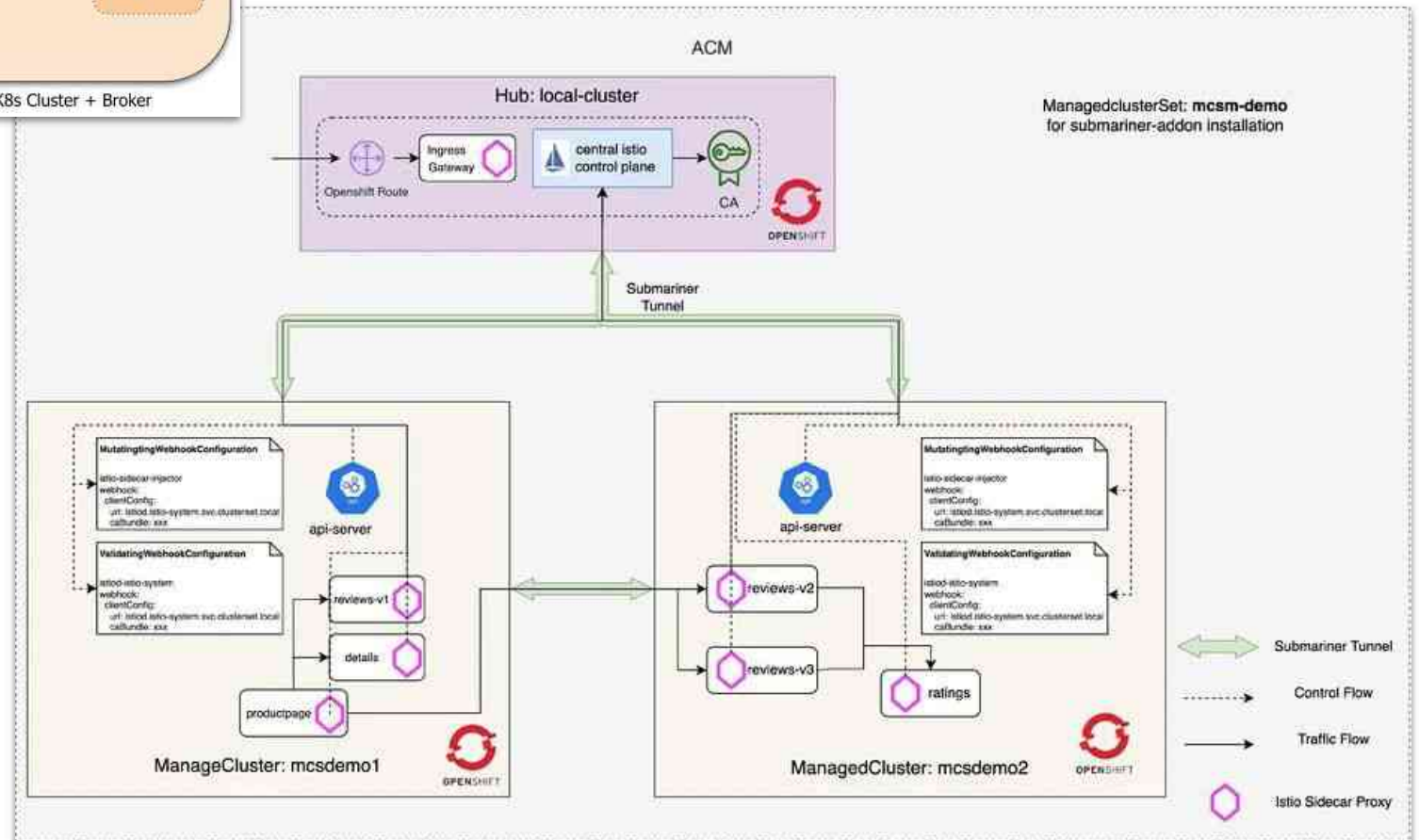


State of the Art

Multi-Hybrid-CloudAnwendungs**betrieb**



Red Hat Integration - AMQ Streams provided by Red Hat	
Provided APIs	
K Kafka Represents a Kafka cluster	KC Kafka Connect Represents a Kafka Connector
KMM Kafka Mirror Maker Represents a Kafka MirrorMaker cluster	KB Kafka Bridge Represents a Kafka Bridge cluster
KT Kafka Topic Represents a topic inside a Kafka cluster	KU Kafka User Represents a user inside a Kafka cluster
KC Kafka Connector Represents a Connector inside a Kafka Connect cluster	KMM Kafka MirrorMaker 2 Represents a Kafka MirrorMaker 2 cluster
KR Kafka Rebalance Triggers rebalance of replicas in a Kafka cluster	SPS Strimzi Pod Set Internal custom resource used to manage Strimzi pods

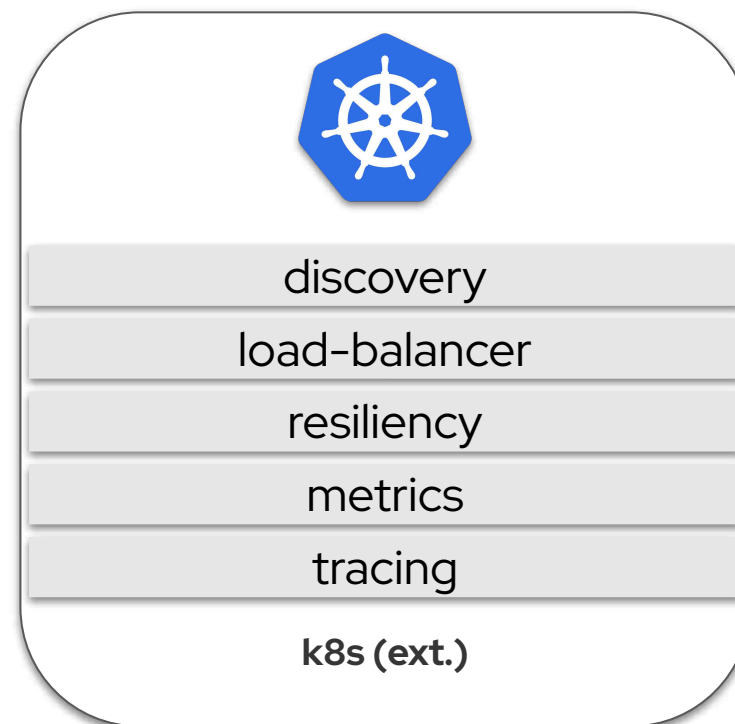
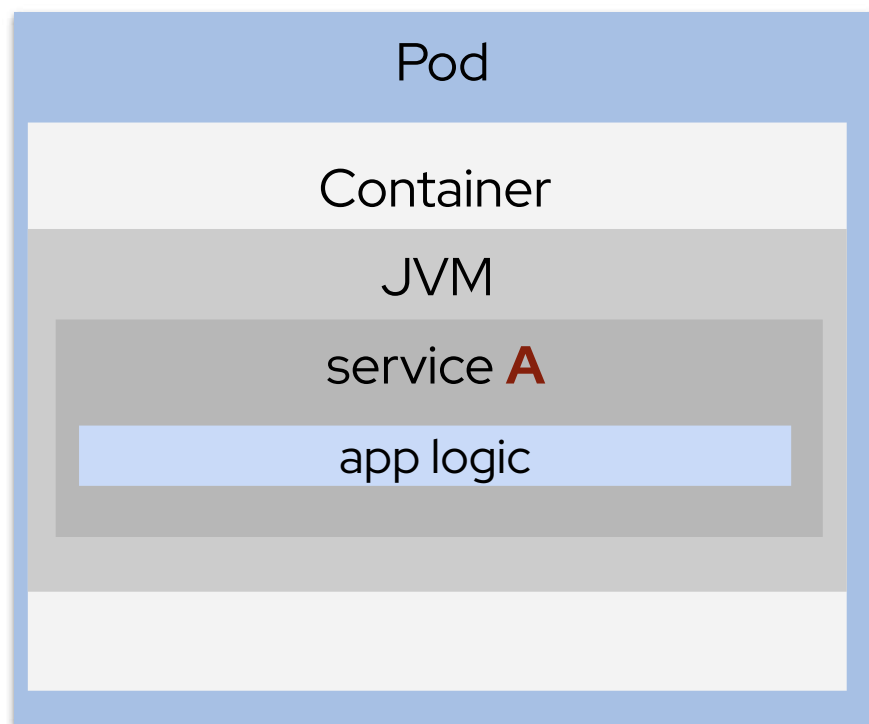


Documentation (Upstream [OSS project]): <https://submariner.io>

Blog Article: <https://cloud.redhat.com/blog/set-up-an-istio-multicluster-service-mesh-with-submariner-in-red-hat-advanced-cluster-management-for-kubernetes>

State of the Application Runtime

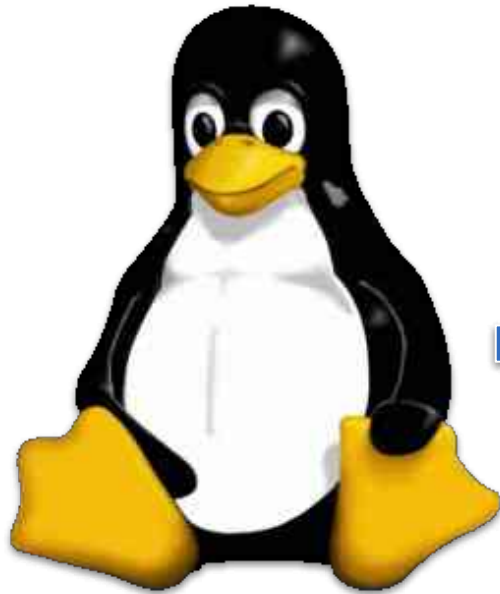
k8s (extended)



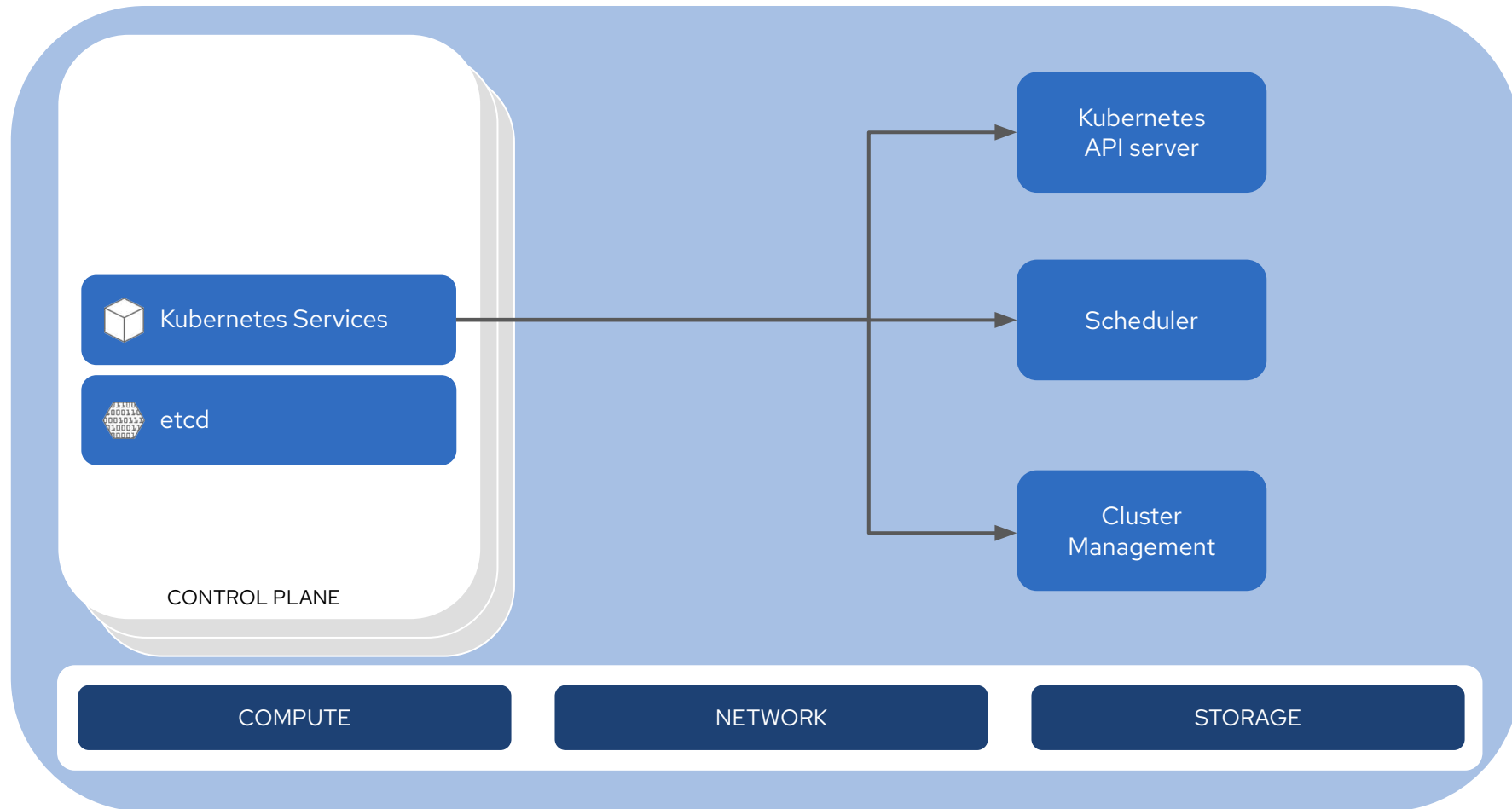
Vanilla k8s – for Developers?



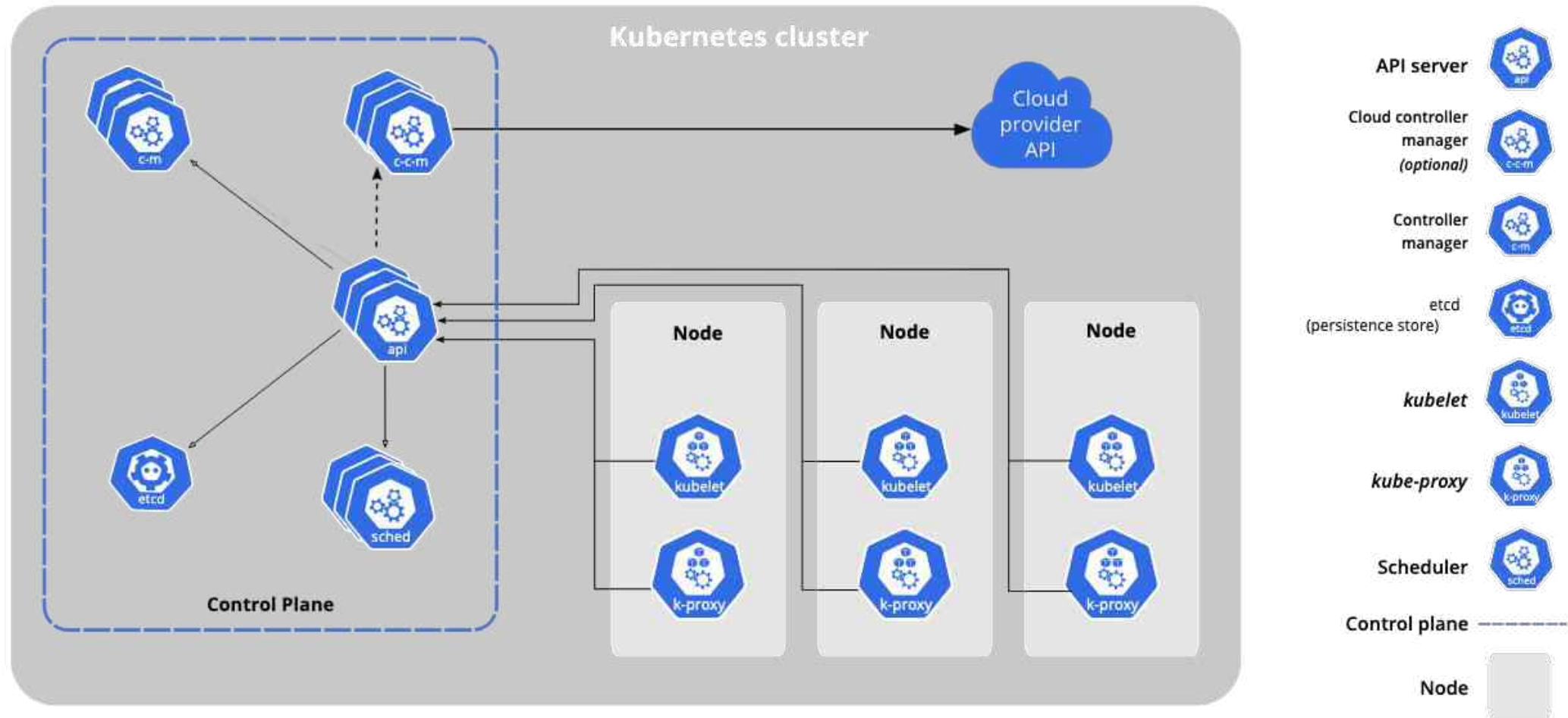
Vanilla Linux auf die Server?



Kubernetes Kernkomponenten



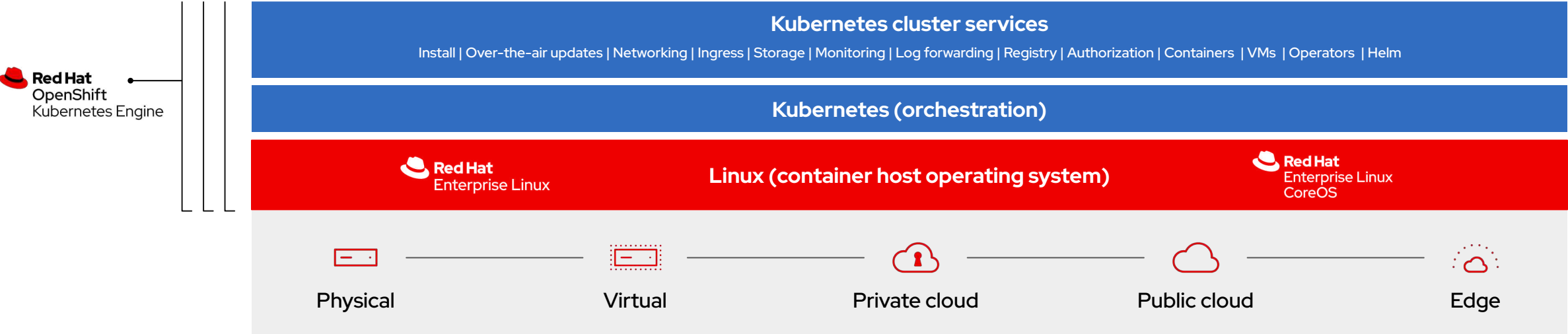
Kubernetes Kernkomponenten



Vanilla k8s auf die Server?



Red Hat OpenShift



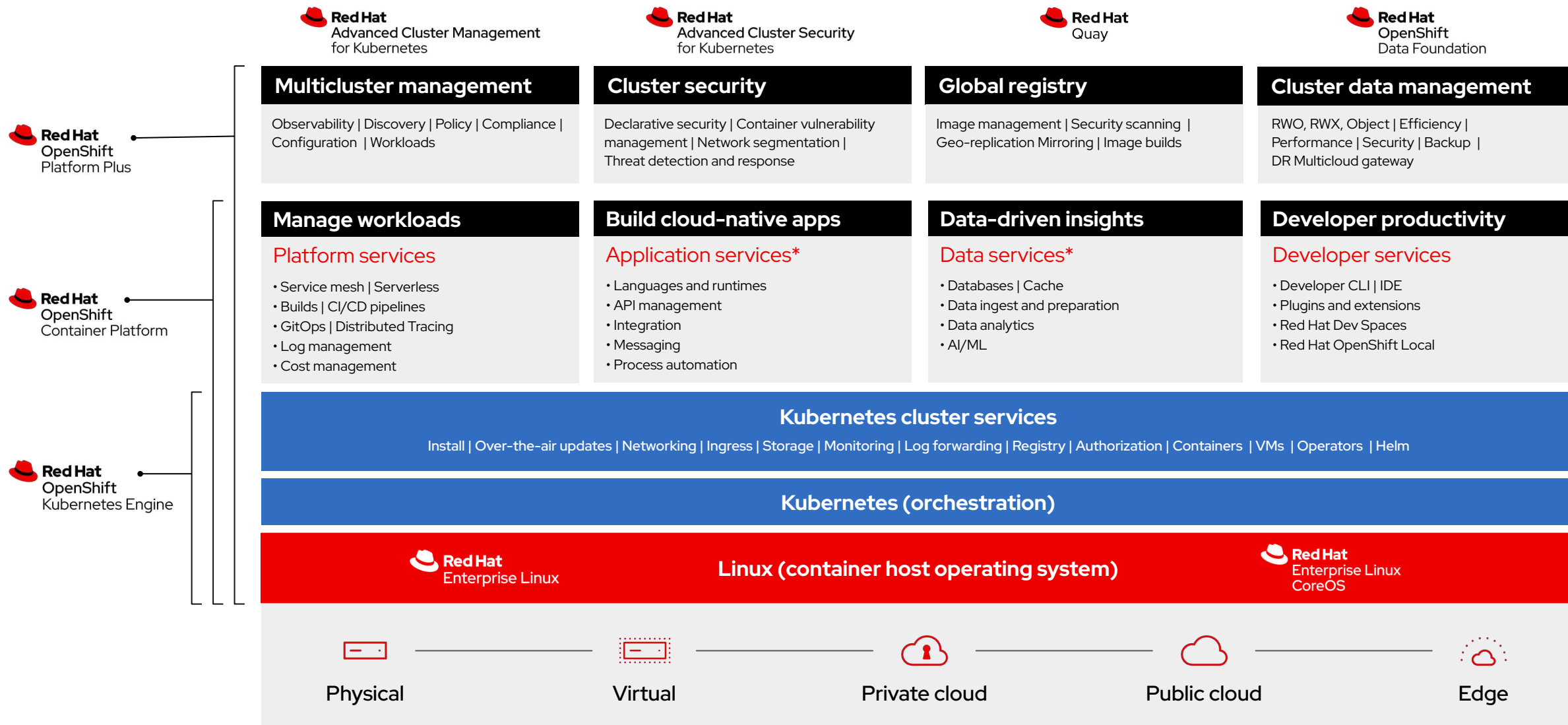
Kein Dilemma mit OpenShift

The Moving Target Platform Dilemma

How Do You Ensure Target Platform Consistency?



Red Hat OpenShift



* Red Hat OpenShift® includes supported runtimes for popular languages/frameworks/databases. Additional capabilities listed are from the Red Hat Application Services and Red Hat Data Services portfolios.

** Disaster recovery, volume and multicloud encryption, key management service, and support for multiple clusters and off-cluster workloads requires OpenShift Data Foundation Advanced

OpenShift == certified kubernetes!

CLOUD NATIVE LANDSCAPE

CNCF Cloud Native Interactive Landscape

The cloud native landscape (png, pdf), serverless landscape (png, pdf), and member landscape (png, pdf) are dynamically generated below. Please open a pull request to correct any issues. Greyed logos are not open source. Last Updated: 2022-10-04T09:16:08.

You are viewing 68 cards with a total of 70,300 stars, market cap of \$3.8T and funding of \$1.3B.

Platform - Certified Kubernetes - Distribution (68)

Reset Filters

Grouping

Category

Sort By

Project Started (earlier to later)

Category

Certified Kubernetes - Distribution

Project

Any

License

Any

Organization

Any

Headquarters

Any

Company Type

Any

Industry

Any

Download as CSV

Example filters

Cards by age

Open source landscape

Member cards

Cards by stars

Cards from China

Certified K8s/KCSP/KTP

Cards by MCap/Funding

Cards without bestpractices.dev

Red Hat OpenShift

KUBERNATIC

CANONICAL

Microsoft Azure

Typhoon

Deckhouse Kubernetes Platform

RANCHER

Kubic

METALK8S

KUBESPHERE

MicroK8s

WINDRIVER

ALVISTACK

K3S

AGORA KUBE

Flexkube

kurl

FURY

RKE GOVERNMENT

KOS

elastisys

PaaS-TA OpenPaaS

VMware Tanzu Community Edition

aws

desktop kubernetes

vcluster

KubeCube

Intel Smart Edge Open

Kamaji

ACCORDION

AiStation

alouda Container Platform

ASUS Cloud Infra

bizkube

博云 BoCloud

铁钟院 CARS

磐基平台

中国东信 China-ASEAN Information Hub

CISCO

Cloudbooster

Cocktail Cloud

Constellation

D2 IQ

DaoCloud

DIAMANTI

易捷行云 EasyStack

ERICSSON

Giant Swarm

中电金信 GienTech

H3C

谐云科技 HARMONY CLOUD

Hewlett Packard Enterprise

inspur 浪潮

JD.COM

K8sPlus

kublr

MIRANTIS

网易数帆 NETEASE DIGITALS

ORACLE

palette

PLATFORM9

ROBIN

时速云 tenscloud.com

TmaxA&C

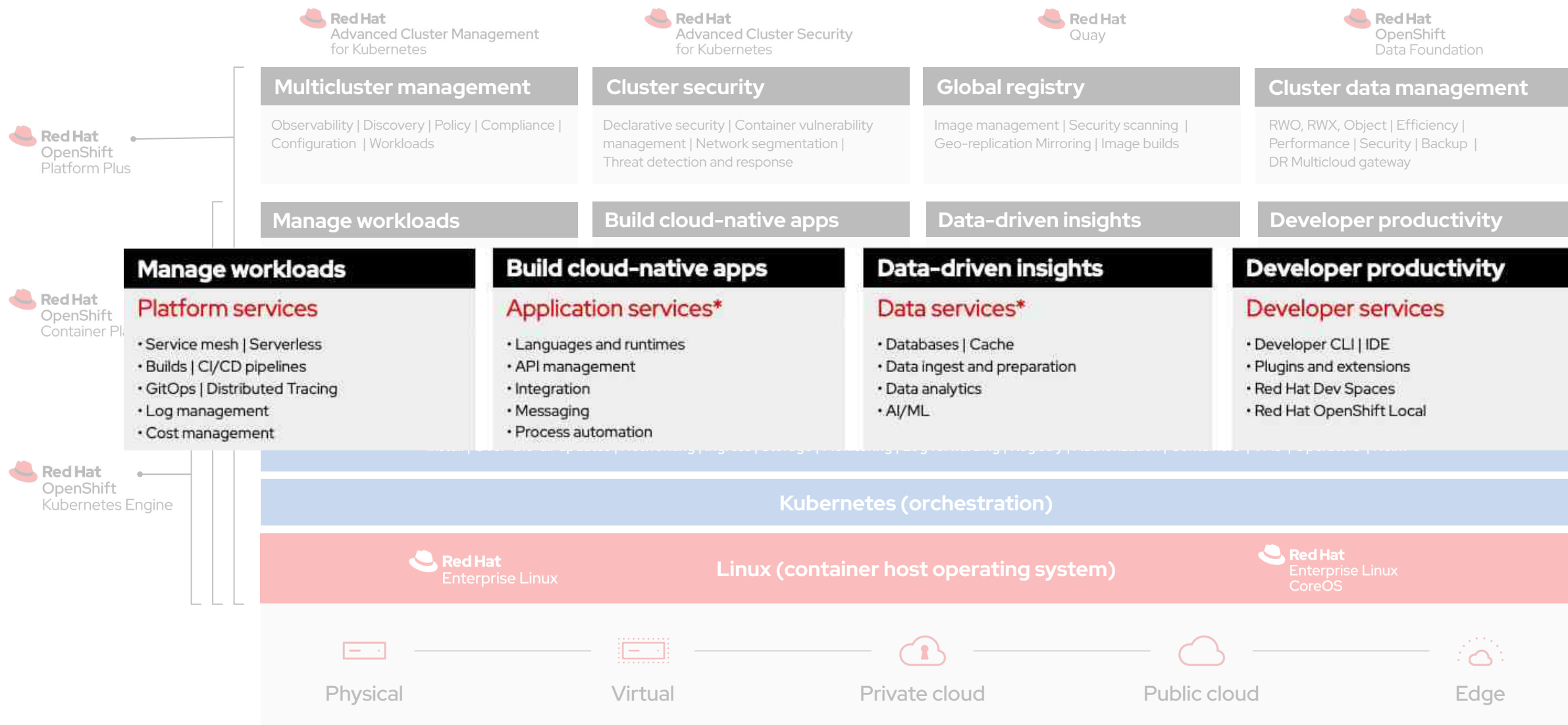
VMware Tanzu

VMware Tanzu

whitestack

ise2c

Red Hat OpenShift

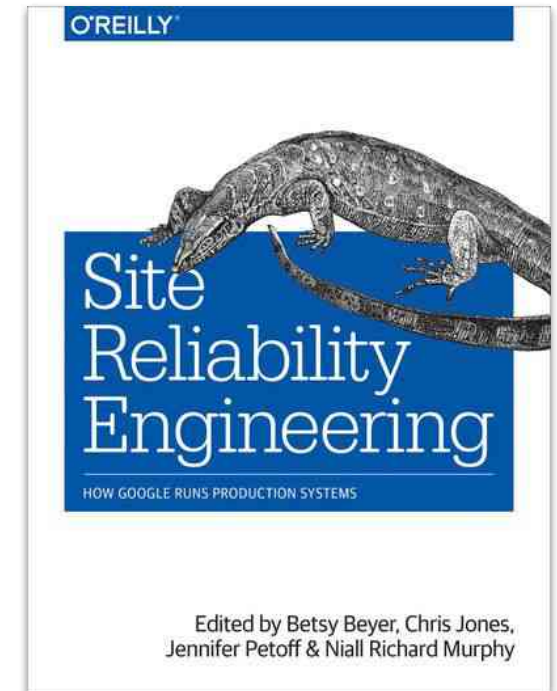
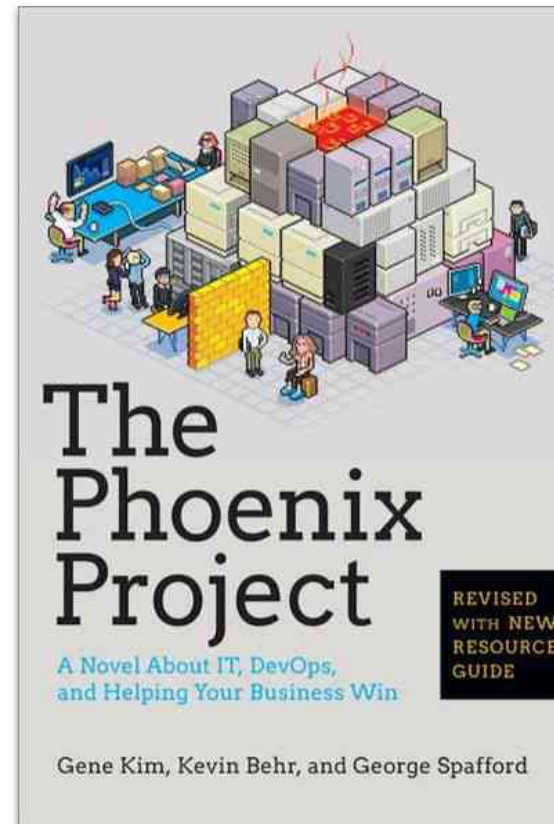


Warum?



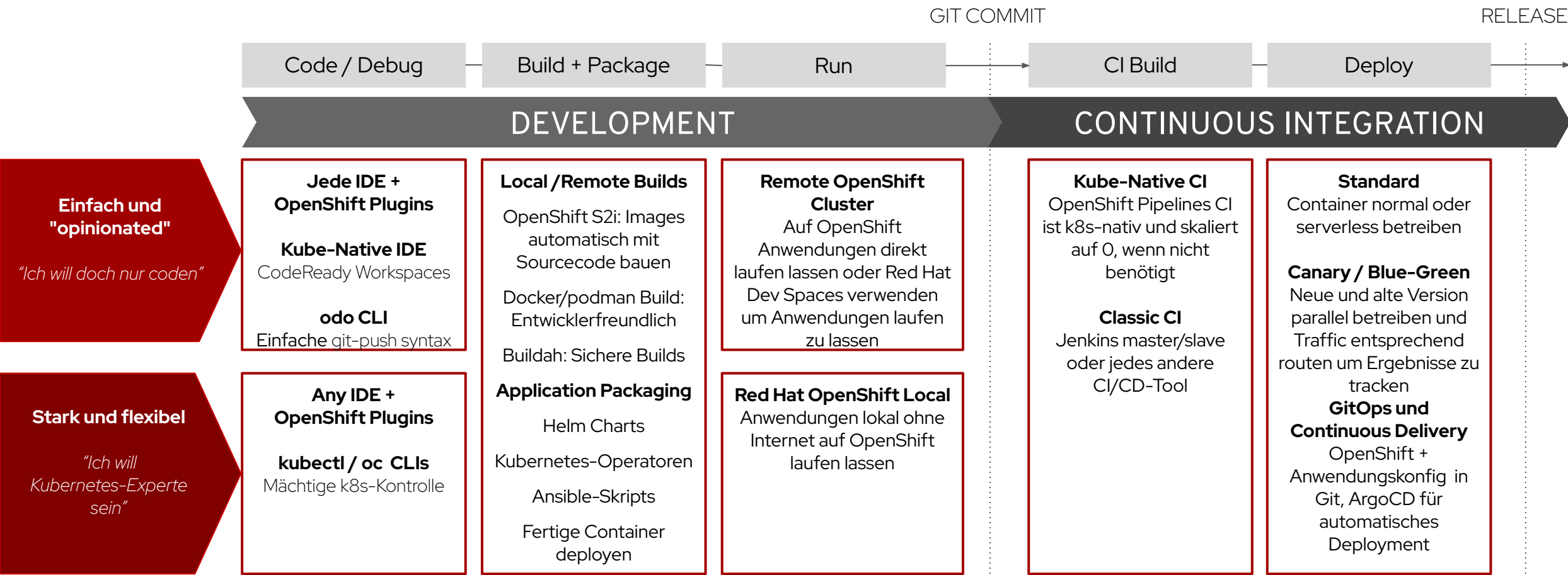
Container-Image

Cloud-nativ entwickeln geht nicht ohne Verzahnung

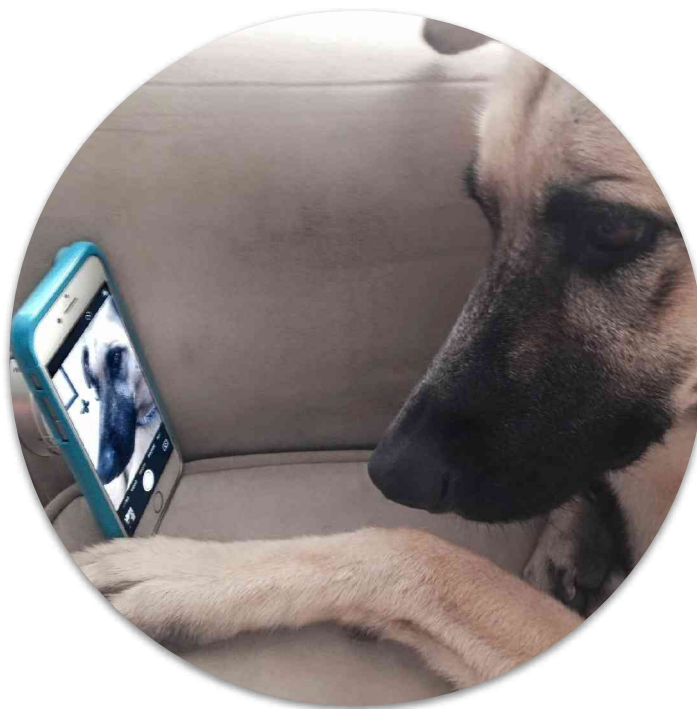


developer != developer

Unterschiedliche Annäherungsgrade



Wie?



Software **Development** Lifecycle

Disciplines, **not** formal, e.g. ISO/IEEE...

DESIGN

PLAN
REQUIREMENTS
ARCHITECTURE

DEVELOP

VCS
PROTOTYPING
CODING

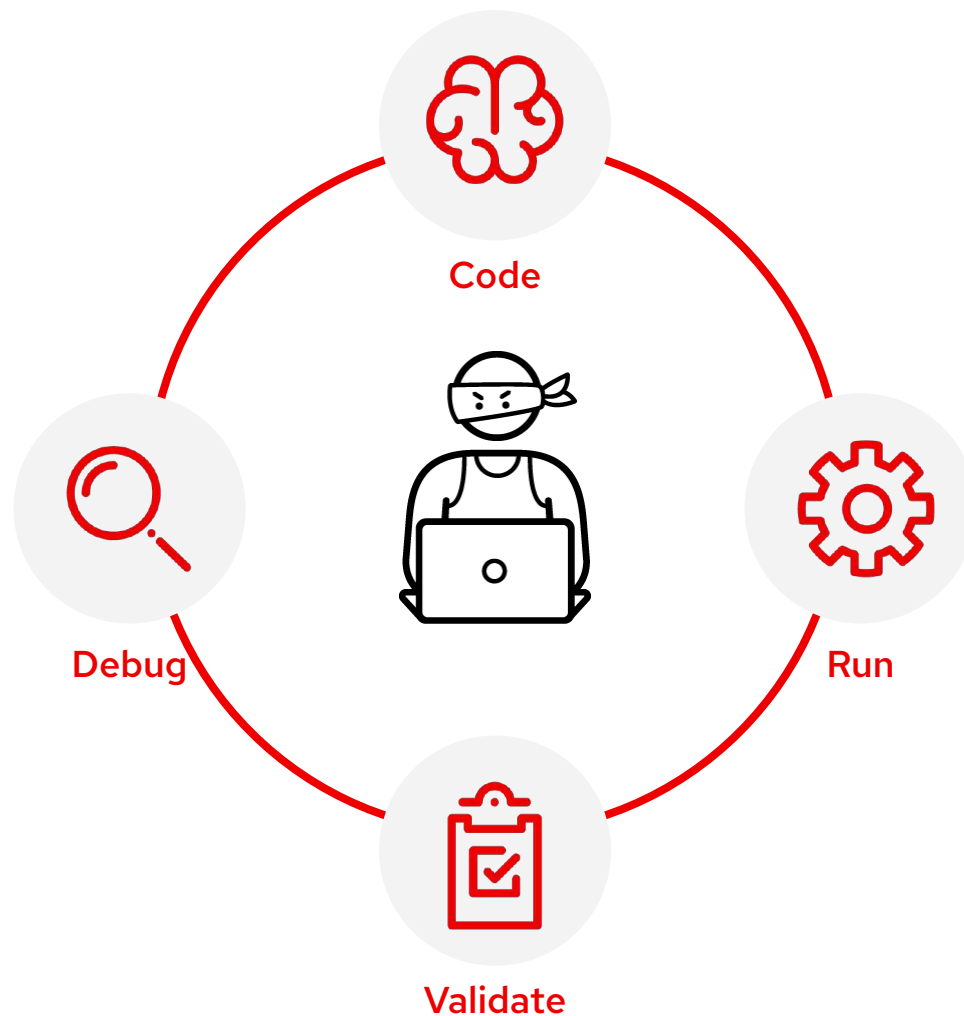
TEST

DEBUGGING
UNIT TESTS
INTEGRATION

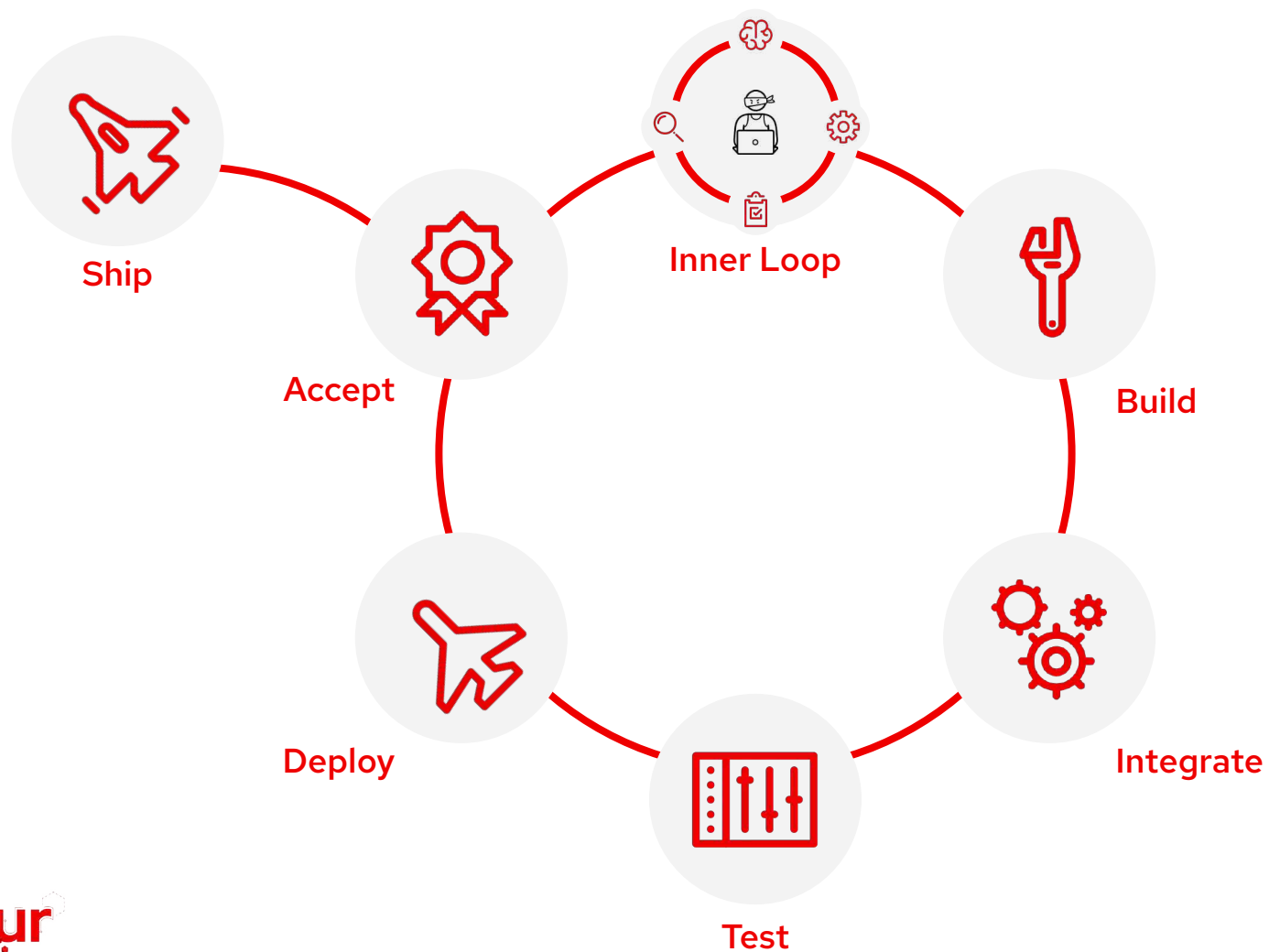
DEPLOY

STAGING
CI/CD/GITOPS
APP OPS

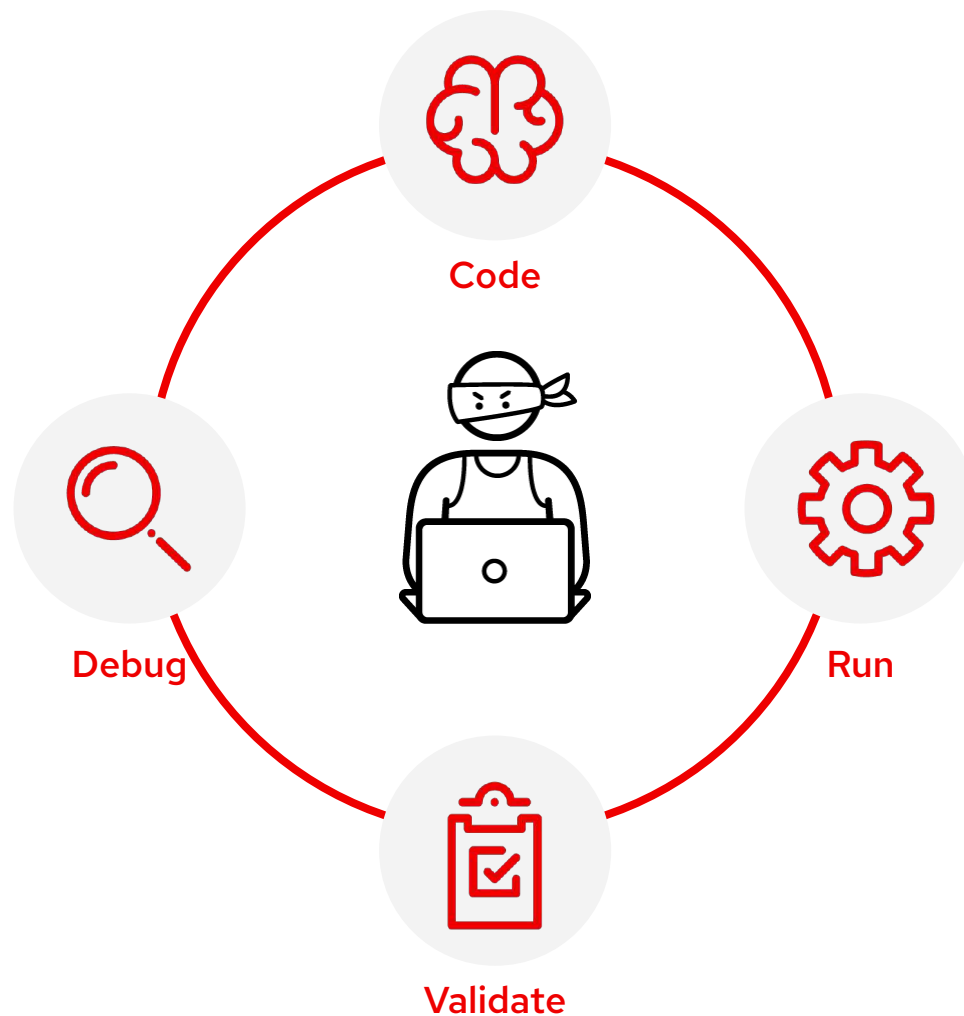
Developer INNER “Loop”



Developer OUTER Loop



Developer **INNER** "Loop"



Quarkus – Teil von OpenShift!

Cloud-Native Java-Runtime von Grund auf



QUARKUS

- ▶ konfigurationsfrei, live-reload
- ▶ Standardbasiert, doch erweiterbar
- ▶ Einheitliche Konfiguration
- ▶ Streamlined Code für 80% der Anwendungsfälle, flexibel bei den anderen 20%
- ▶ Imperativ und reaktiv coden
- ▶ Native Anwendungen einfach erstellen



odo - "OpenShift do!"

Anwendungen nach k8s
bringen ohne Spezialistentum

```
~/dev/ws/mbp02/rhcli/odo $ cd ~/dev/ws/mbp02/rhcli/odo
~/dev/ws/mbp02/rhcli/odo $ odo project create odospringjava
✓ Project 'odospringjava' is ready for use
✓ New project created and now using project: odospringjava
~/dev/ws/mbp02/rhcli/odo $ mkdir odospringjava
~/dev/ws/mbp02/rhcli/odo $ cd odospringjava
~/dev/ws/mbp02/rhcli/odo/odospringjava $ odo create java-spring

Devfile Object Validation
✓ Checking devfile existence [43820ns]
✓ Creating a devfile component from registry: DefaultDevfileR
Validation
✓ Validating if devfile name is correct [86855ns]

Starter Project
✓ Downloading starter project springbootproject from https://

Please use `odo push` command to create the component with source deployed
~/dev/ws/mbp02/rhcli/odo/odospringjava $ odo push

Validation
✓ Validating the devfile [131632ns]

Creating Kubernetes resources for component odospringjava
✓ Added storage m2 to odospringjava
✓ Waiting for component to start [6s]
✓ Waiting for component to start [29ms]
```

```
$ odo dev
```



Developing using the my-nodejs-app Devfile
Namespace: default
odo version: v3.0.0-alpha2

⇒ Deploying to the cluster in developer mode

- ✓ Waiting for Kubernetes resources [3s]
- ✓ Syncing files into the container [330ms]
- ✓ Building your application in container on cluster [4s]
- ✓ Executing the application [1s]

Your application is now running on the cluster

- Forwarding from 127.0.0.1:40001 -> 3000

Watching for changes in the current directory /Users/user/express
Press Ctrl+c to exit `odo dev` and delete resources from the cluster

Operator Framework

The screenshot displays the OpenShift OperatorHub interface. The top navigation bar includes the Red Hat OpenShift Container Platform logo and a user profile 'karsten'. The left sidebar contains a menu with 'Administrator', 'Home', 'Operators', 'Workloads', 'Serverless', 'Networking', 'Storage', 'Builds', 'Monitoring', and 'Compute'. The 'Operators' section is expanded, showing 'OperatorHub' and 'Installed Operators'. The main content area is titled 'OperatorHub' and includes a description: 'Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through Red Hat Marketplace. You can install Operators on your clusters to provide optional add-ons and shared services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experience.' Below this, there is a filter bar with 'All Items' and a search input 'Filter by keyword...'. A list of operators is displayed in a grid, each with a logo, name, provider, and description. The operators shown are: 3scale API Management (provided by Red Hat), Akka Cluster Operator (provided by Lightbend, Inc.), Alvearie Imaging Ingestion Operator (provided by Alvearie), Apache Spark Operator (provided by radanalytics.io), API Operator for Kubernetes (provided by WSO2), APIcast (provided by Red Hat), Apicurio Registry Operator (provided by Apicurio), Apicurito Operator (provided by Apicurio Project), and several others. The bottom of the interface shows a 'Provider type' filter with 'Red Hat (58)' selected.

IDE Tooling - Products



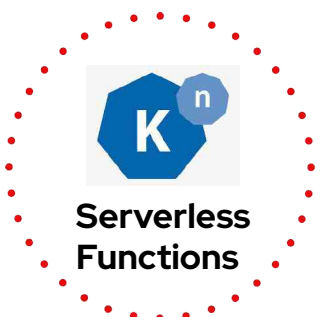
OpenShift Connector

Supports OpenShift Local 2.5.1
Supports odo 2.5.1
Components using Default Devfile Registry
Improved Get Started Workflow



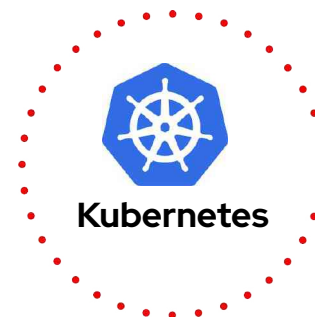
Tekton Pipelines Extension

Support latest Tekton Pipelines 0.35.0
Support latest Tekton Triggers 0.20.0
Support latest tkn cli 0.24.0



Knative/Functions Extension

Support Serverless Function Workflow
Support Node, Quarkus, Go, Python language
Support s2i builder images
Support on-build cluster (in progress)



IntelliJ K8s Extension

Continue Kubernetes support on IntelliJ
Cluster resource error management
Diff viewer for local and remote K8s resource

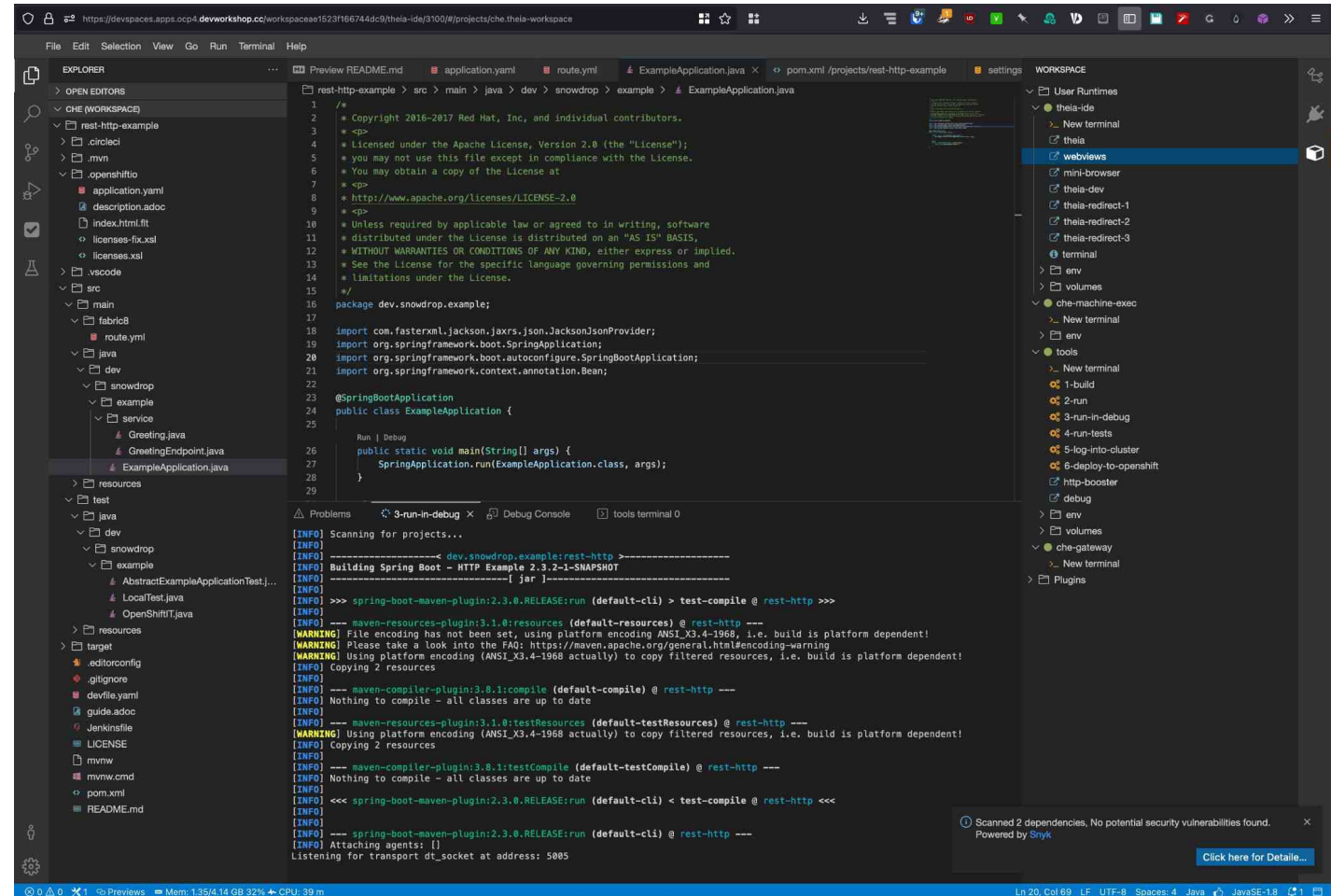


Red Hat OpenShift Dev Spaces (Formerly known as CodeReady Workspaces)

- ▶ Workspaces containerisiert
- ▶ web-IDE (Theia) basiert auf Monaco-Framework (wie VSCode)
- ▶ Debugging, Language-Support containerisiert
- ▶ Anbindung wie Snyk zur Code-Kontrolle

Resources

- [Detailed Blog on Whats in V3.0](#)
- [Revamped Documentation](#)
- [How to Install](#) - Red Hat OpenShift Dev Spaces is available with OpenShift 4.10 onwards
- [How to Upgrade](#)



devfile

```
schemaVersion: 2.2.0
metadata:
  name: java-springboot
  version: 1.1.0
  attributes:
    alpha.dockerimage-port: 8081
  displayName: Java Spring Boot
  description: Java Spring Boot using Maven
  tags: ["Java", "Spring"]
  projectType: "springboot"
  language: "java"
  provider: Red Hat
parent:
  id: java-springboot
  registryUrl: "https://registry.devfile.io"
components:
  - name: outerloop-build
```

```
- name: outerloop-build
  image:
    imageName: java-springboot-image:latest
    dockerfile:
      uri: docker/Dockerfile
      buildContext: .
      rootRequired: false
- name: outerloop-deploy
  attributes:
    deployment/replicas: 1
    deployment/cpulimit: "100m"
  kubernetes:
    uri: outerloop-deploy.yaml
```

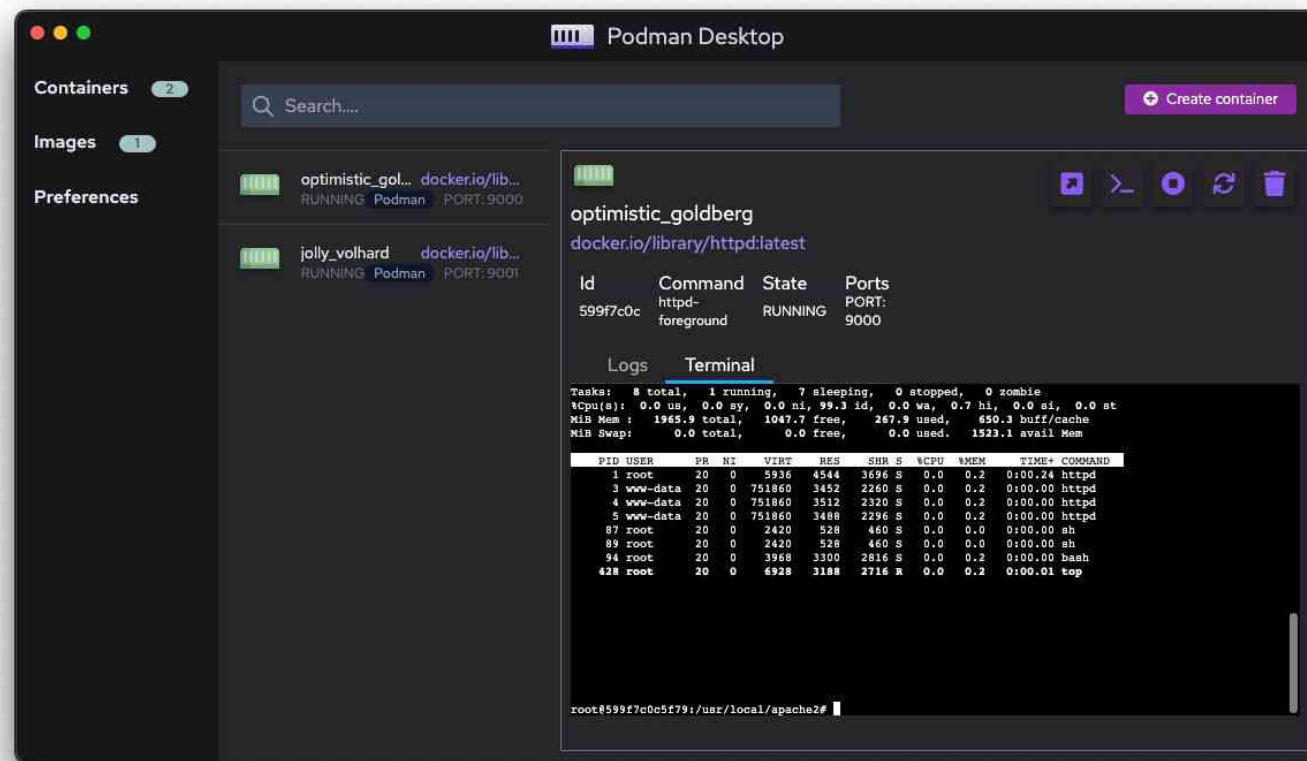
```
commands:
  - id: build-image
    apply:
      component: outerloop-build
  - id: deployk8s
    apply:
      component: outerloop-deploy
  - id: deploy
    composite:
      commands:
        - build-image
        - deployk8s
      group:
        kind: deploy
        isDefault: true
```

Podman Desktop

Early Stage

Containers and Kubernetes for application developers

- Run anywhere: Windows, Mac and Linux
- Install and keep-up-to date Podman Engine
- Build, test, run and inspect containers
- Configure and work with container registries
- Readiness for enterprise: Proxy, VPN configuration
- Deploy containers to Kubernetes
- Run and work with OpenShift/Kubernetes Local
- Extensible container engine provider and plugins

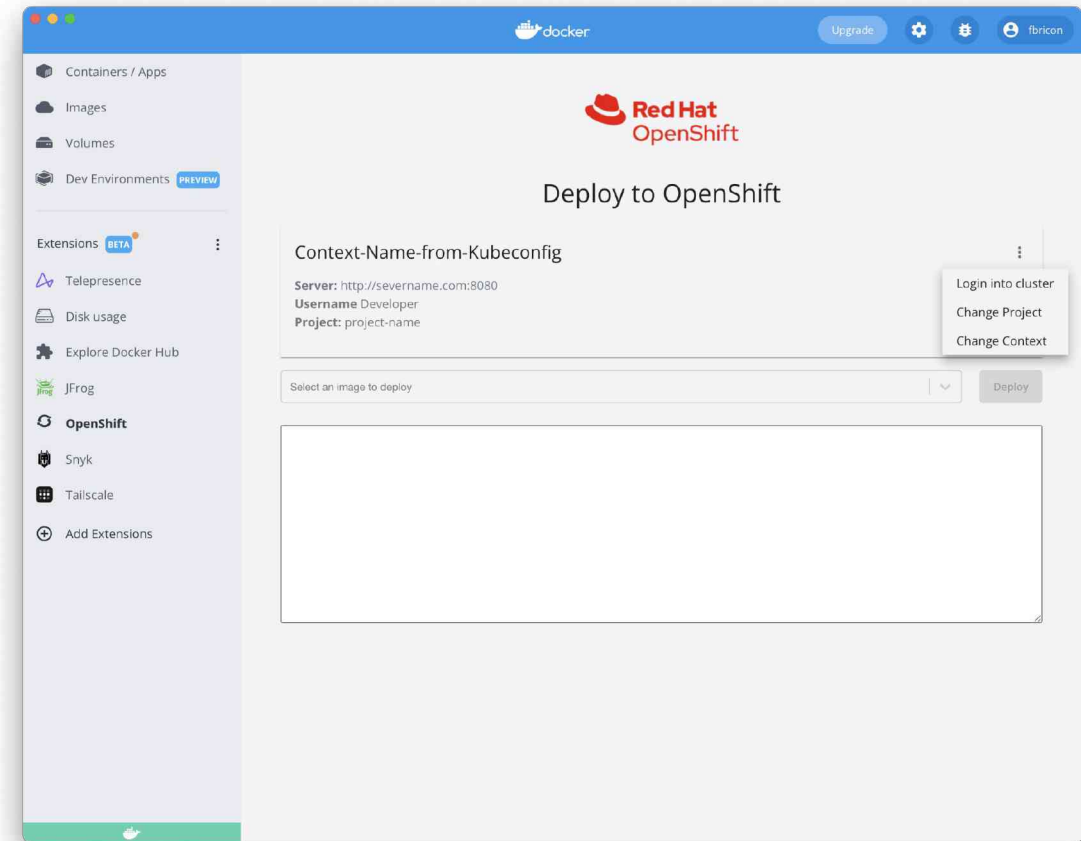


OpenShift Extension for Docker Desktop

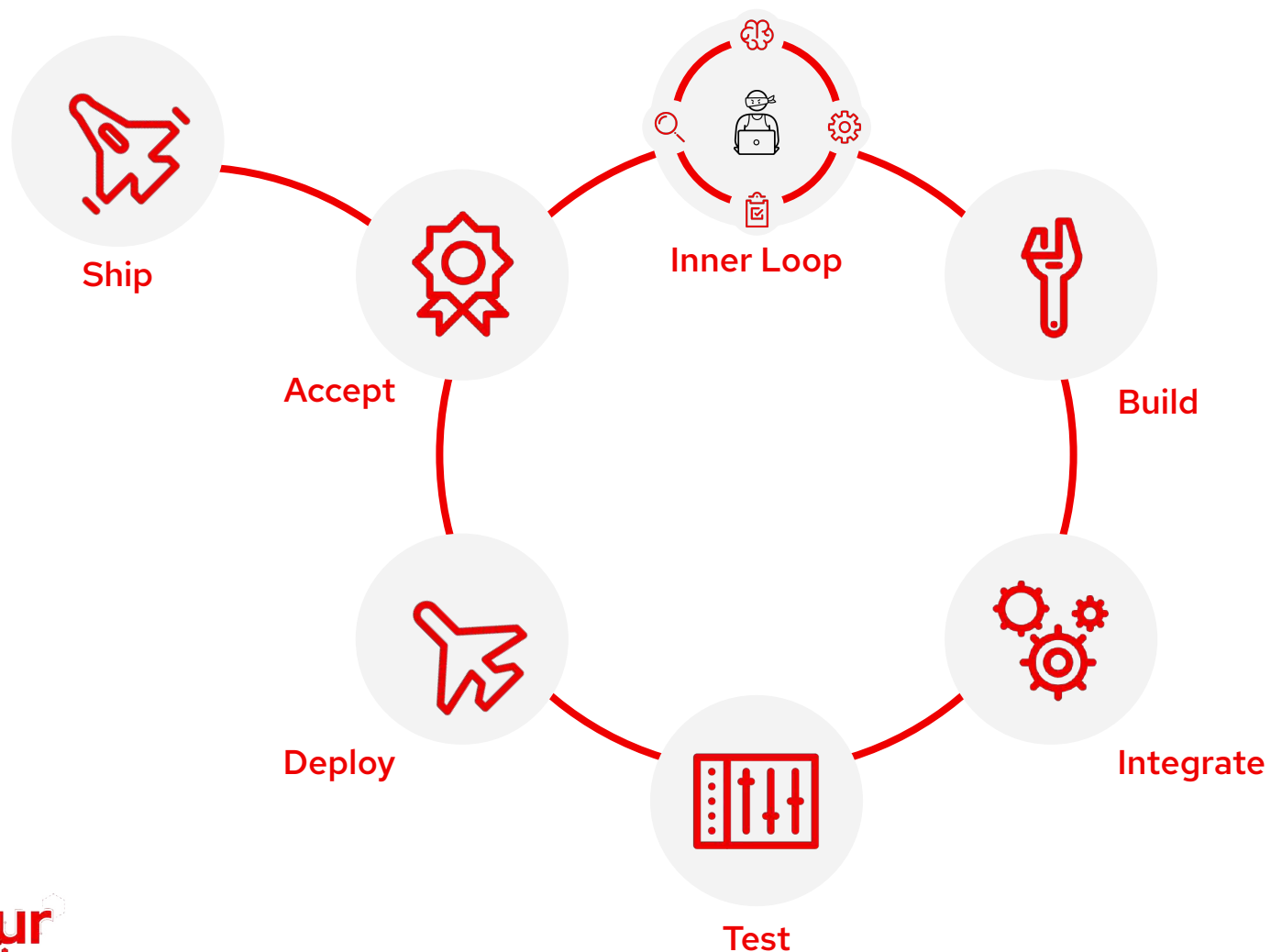
Easily Build and Deploy Containers to OpenShift

10k Pulls

- Detect Kubernetes environments
- Login into cluster
- Namespace selection
- Selection of local containers
- Push image to Registry
- Deploy to cluster
- Easy access to Kubernetes with DevSandbox



Developer OUTER Loop



Developer Console

Ressourcen festlegen!

No resources selected

Ins Menü setzen!

Red Hat OpenShift Container Platform

Developer

+Add

Topology

Monitoring

Search

Project: mdprea

Search

Resources 0

Name my-resource

Project: mdprea

Search

Resources 1

Name quarkus

Resource CM ConfigMap

Name quarkus

Create ConfigMap

Name	Size
CM quarkus-quickstarts-1-ca	1
CM quarkus-quickstarts-1-global-ca	1
CM quarkus-quickstarts-1-sys-config	0

Name	Namespace	Size	Created
CM quarkus-quickstarts-1-ca	NS mdprea	1	Jul 5, 2021, 11:42 AM
CM quarkus-quickstarts-1-global-ca	NS mdprea	1	Jul 5, 2021, 11:42 AM
CM quarkus-quickstarts-1-sys-config	NS mdprea	0	Jul 5, 2021, 11:42 AM

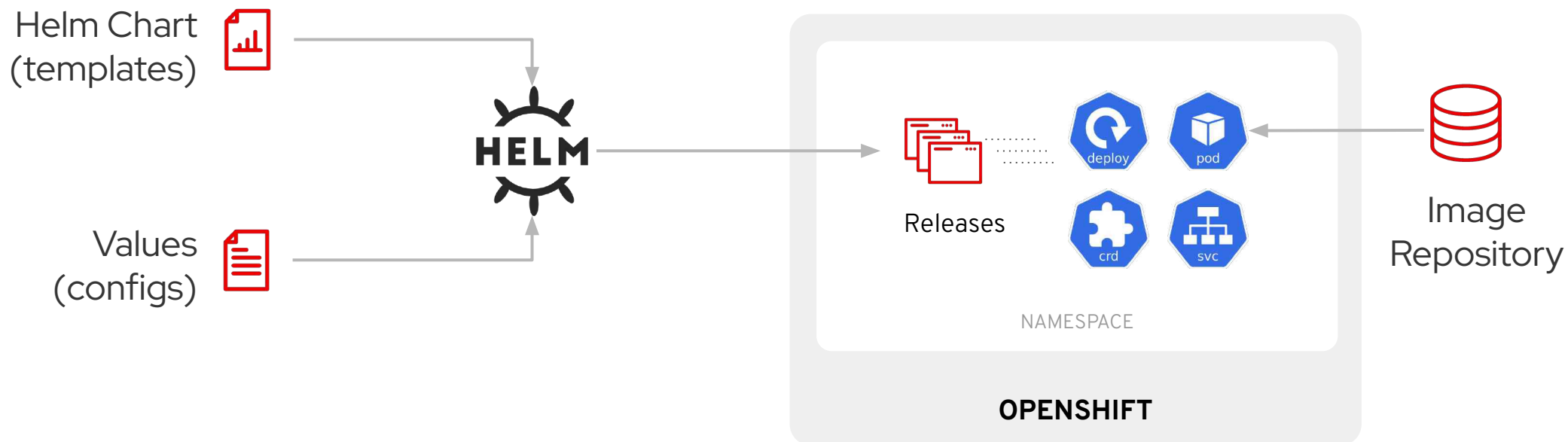
Deployments apps/v1

Create Deployment

Add to navigation

Name	Namespace	Status	Labels	Pod selector
ID quarkus-quickstarts	NS mdprea	1 of 1 pods	a...=quarkus-quic... app.kube...=quar... app.kub...=quark...	Q app=quarkus-quickstarts

Helm on OpenShift



Key Features

- ▶ Integration with the developer console to enable self-service consumption of Helm Charts from the developer catalog
- ▶ Simpler way to extend available Helm Charts available in the Developer Catalog with a form based approach which is getting more discoverable too.
 - ▶ Configure custom Helm Chart Repositories at cluster level
 - ▶ Configure custom Helm Chart Repository at namespace level, which requires less permissions

Service Binding Operator

Key Features & Updates

- ▶ Blog posts and demos leveraging Service Binding Operator along with ACK Controller from AWS.
- ▶ Support of the latest version of Service Binding Specification
- ▶ Improvements in OpenShift Developer Console and `odo`, on leveraging Service Binding

```
import com.nebhale.bindings.Binding;
import com.nebhale.bindings.Bindings;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

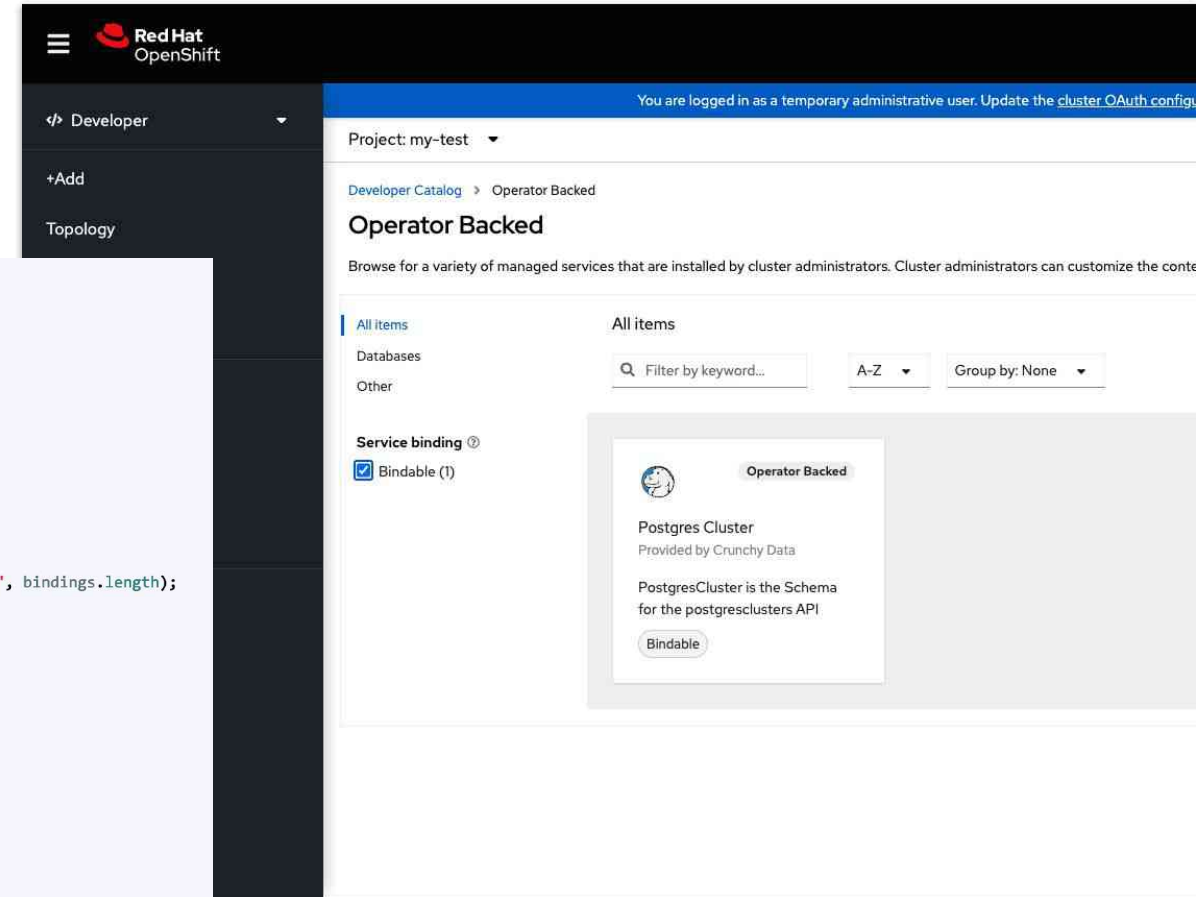
public class Application {

    public static void main(String[] args) {
        Binding[] bindings = Bindings.fromServiceBindingRoot();
        bindings = Bindings.filter(bindings, "postgresql");
        if (bindings.length != 1) {
            System.err.printf("Incorrect number of PostgreSQL drivers: %d\n", bindings.length);
            System.exit(1);
        }

        String url = bindings[0].get("url");
        if (url == null) {
            System.err.println("No URL in binding");
            System.exit(1);
        }

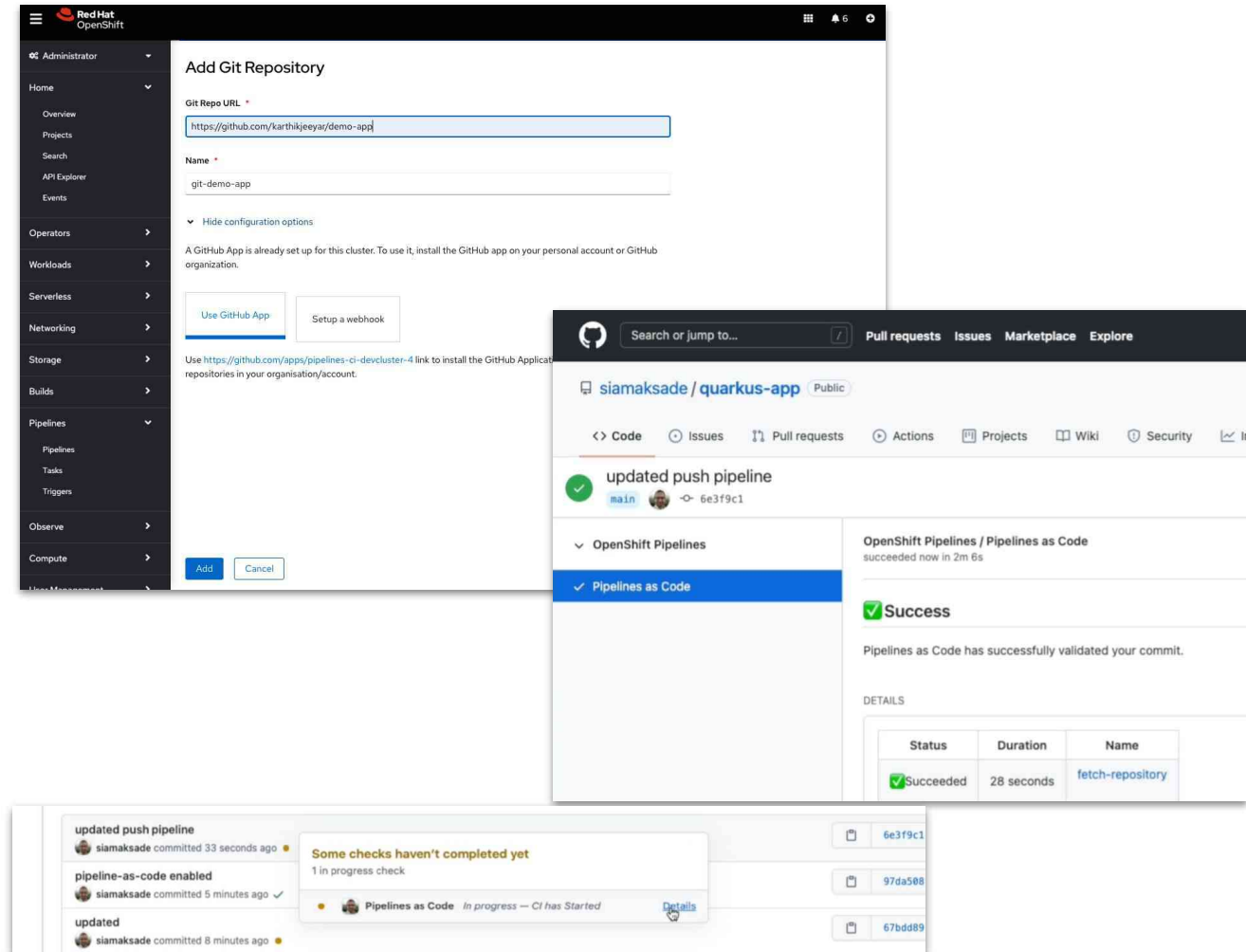
        Connection conn;
        try {
            conn = DriverManager.getConnection(url);
        } catch (SQLException e) {
            System.err.printf("Unable to connect to database: %s", e);
            System.exit(1);
        }

        // ...
    }
}
```



OpenShift Pipelines

- ▶ OpenShift Pipelines 1.8
- ▶ External database support in Tekton Hub
- ▶ Pipelines on Arm architecture (Tech Preview)
- ▶ Pipelines as code enhancements
 - ▶ Trigger multiple pipelines for Git event
 - ▶ GitLab and BitBucket support
 - ▶ CLI commands for configuring webhooks
 - ▶ Manual and third-party triggers
- ▶ Dev Console enhancements
 - ▶ Configure Git repositories with pipelines as code
 - ▶ Create GitHub App for pipelines as code



OpenShift GitOps

- ▶ OpenShift GitOps 1.6
- ▶ Argo CD 2.4
- ▶ ApplicationSets (General Availability)
- ▶ Notifications (Tech Preview)
- ▶ Secret management guide
- ▶ Custom plugins in Argo CD
- ▶ Single sign on updates to Dex syntax
- ▶ Support for running on IBM Power and IBM Z
- ▶ Encrypted comms with Redis
- ▶ Dev console deployment history

The screenshot displays the OpenShift GitOps interface. At the top, there are tabs for 'LIVE MANIFEST', 'DIFF', and 'DESIRED MANIFEST'. The 'LIVE MANIFEST' tab is active, showing a YAML configuration for an ApplicationSet. The configuration includes annotations for sync-wave and last-applied-configuration, a creation timestamp of '2022-03-31T10:00:00Z', and a generation of 3. Below the manifest, there is a section for triggers, including 'on-sync-status-unknown' and 'on-sync-succeeded'. A notification overlay is visible on the right side, stating 'Application my-app-2 has been successfully synced.' with a date of 'Jun 23' and a 'to me' dropdown menu.

```

1  apiVersion: argoproj.io/v1alpha1
2  kind: ApplicationSet
3  metadata:
4    annotations:
5      argocd.argoproj.io/sync-wave: 1
6      kubectrl.kubernetes.io/last-applied-configuration: '{"apiVersion":"argoproj.io/"
7    creationTimestamp: "2022-03-31T10:00:00Z"
8    generation: 3
9    labels:
10     app.kubernetes.io/instance: my-app-2
11
12  triggers:
13    - on-sync-status-unknown:
14      - description: Application status is 'Unknown'
15      send:
16        - app-sync-status-unknown
17      when: app.status.sync.status == 'Unknown'
18    - on-sync-succeeded:
19      - description: Application syncing has succeeded
20      send:
21        - app-sync-succeeded
22      when: app.status.operationState.phase in ['Succeeded']
23  kind: ConfigMap
24  metadata:

```


Demo Time 😁





Red Hat



Red Hat
Developer

Join Red Hat Developer.
Build here. Go anywhere.



youtube.com/RedHatDevelopers



linkedin.com/showcase/red-hat-developer



facebook.com/RedHatDeveloperProgram



twitter.com/rhdevelopers



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

Danke!

"To be the catalyst in communities of
customers, contributors, and partners
**creating better technology the
open source way.**"

Red Hat's Mission Statement



twitter.com/RedHat

