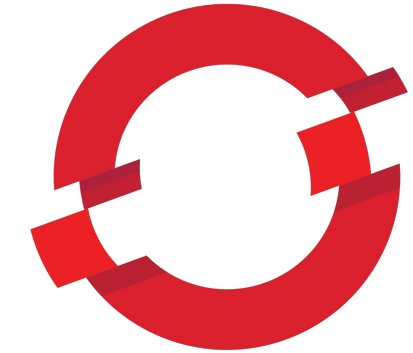



# Grundlagen OpenShift

Tobias Derksen & Norbert Schneider

codecentric

# Agenda

- Übersicht Orchestrierung
- Einführung in OpenShift
- Cluster Aufbau
- Objekte & Ressourcen
- Web Interfaces
- CLI Basics
- Persistent Storage Framework
- OpenShift 4

# Container Orchestrierung

# Aufgaben der Container Orchestrierung

- Starten und Stoppen von Containern
- Konfigurieren von Containern, Netzwerk, Volumes, etc
- Verteilen von Container auf verschiedene Hosts
- Überwachen der Container
- Maßnahmen falls Container abstürzen


&

MESOS

kubernetes

RANCHER®

# Einführung in OpenShift

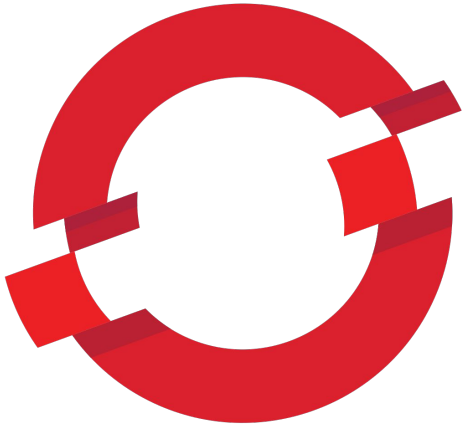
# Was ein Chaos ...

redhat

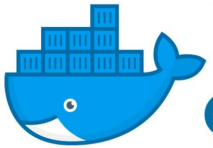
okd

kubernetes

OPENSHIFT origin



cri-o



docker

OPENSHIFT

# OpenShift ist … kubernetes plus

- Routing
- Metriken
- Logging
- Web Oberfläche
- Builds
- Image Registry
- Sicherheitsmaßnahmen
- SDN
- Templates

Mit Red Hat Subscription:

- Trusted Registry
- Security Newsletter
- **Support**

# Begriffe

- Container
- Pod
- Node
- Projekt
- Namespace
- etcd

# Cluster Aufbau

# Verschiedene Node Typen

## Master Nodes

API - Server

ETCD

Web Console

## Infrastructure Nodes

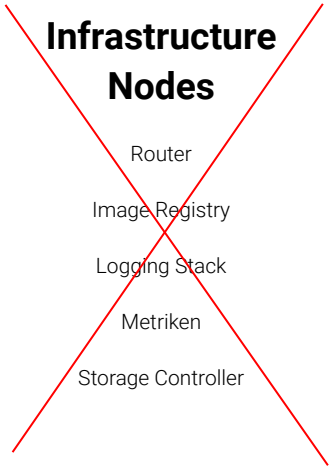
Router

Image Registry

Logging Stack

Metriken

Storage Controller



## Compute Nodes

Applikationen

Services

Datenbanken

Builds

Andere Workloads

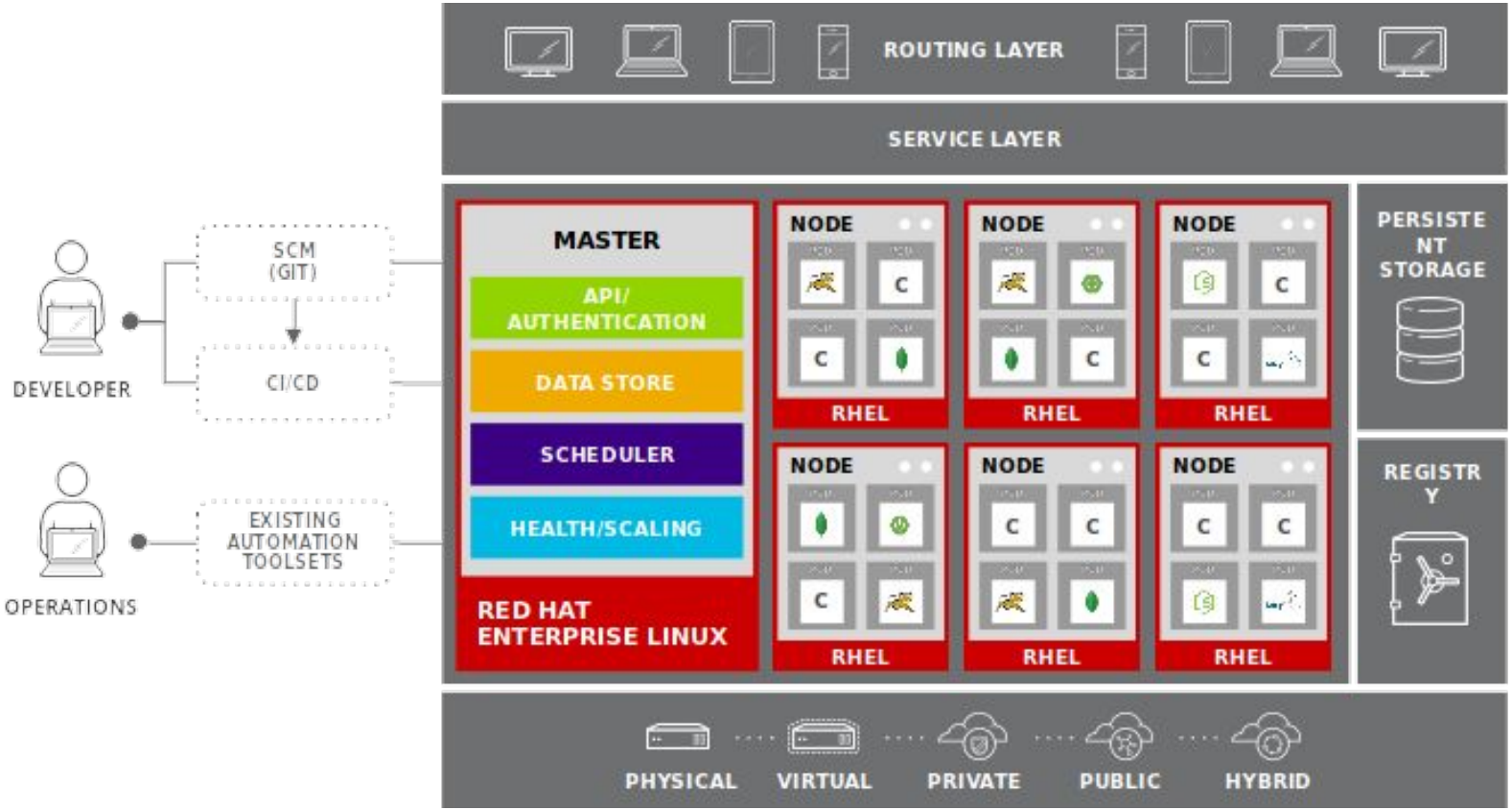
## Storage Nodes

Nur beim EInsatz von Container Storage

Nodes mit physischem Speicher

Können auf mit Compute Nodes kombiniert werden

**Fällt mit OpenShift 4 weg**


ROUTING LAYER
SERVICE LAYER
MASTER
API/ AUTHENTICATION
DATA STORE
SCHEDULER
HEALTH/SCALING
RED HAT ENTERPRISE LINUX
NODE
C
RHEL
PERSISTENT STORAGE
REGISTRY
SCM (GIT)
CI/CD
DEVELOPER
EXISTING AUTOMATION TOOLSETS
OPERATIONS
PHYSICAL
VIRTUAL
PRIVATE
PUBLIC
HYBRID

# Anzahl der Nodes

| | Minimal | Development | Production | Production (HA) |
|---|---|---|---|---|
| **Master** | 1 | 1 | 1 | 3 |
| **Infrastructure** | | | 1+ | 2+ |
| **Compute** | | 2+ | 3+ | 6+ |

**Developer sind teuer, also achtet darauf dass sie nicht warten müssen**

# Objekte & Ressourcen

# Alles nur Objekte ...

- Der **komplette** Zustand des Clusters wird mit Resources abgebildet.

- Cluster Objekte (z.B. Namespaces, Persistent Volumes)
- Projekt Objekte (z.B. Deployments, Builds)

- Die Objekte werden im etcd gespeichert

- Objekte können als JSON oder YAML Format vorliegen
- Alle Objekte unterstützen CRUD Operationen

# Beispiel

- **apiVersion**
- **kind**
- metadata/**name**
- metadata/labels
- metadata/annotations

```yaml
apiVersion: v1
kind: "DeploymentConfig"
metadata:
  name: "instance"
  labels:
    app: test
  annotations:
    logging: true
spec:
  replicas: 5
  selector:
    name: frontend
  template:
    ...
```

# Wichtige Objekt Typen

- Clusterroles
- Rolebindings
- Persistent Volumes
- Persistent Volume Claims
- Template
- Pod

- ConfigMap
- Secret
- Deployment
- DeploymentConfig
- Build
- Route
- Service

# Web Console Basics

# Different GUIs

OpenShift 3.11

- Web Console
- Cluster Console

OpenShift 4:

- Administrator Console
- Developer Console

# OpenShift CLI Basics

# Command Line Interface

- Resource operations (CRUD)
- Administration
- Access Control
- Troubleshooting (logs, debug, portforward, exec, rsync, rsh)

learn.openshift.com

# Persistent Storage

# Persistent Storage Provider

- HostPath
- EmptyDir (Ephemeral Storage)

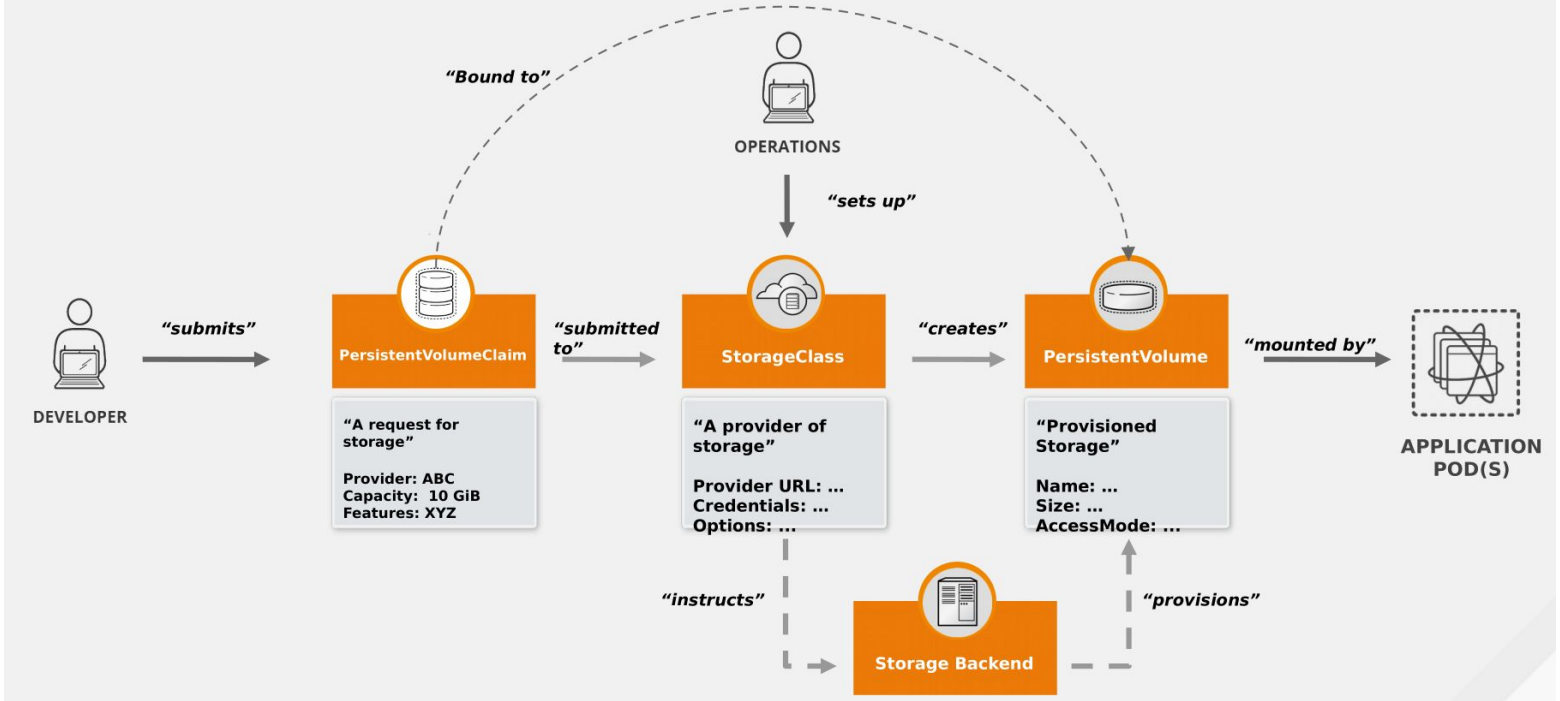
- GlusterFS / OpenShift Container Storage 3
- NFS (unsupported)
- iSCSI
- Ceph / OpenShift Container Storage 4
- Diverse Cloud Mechanismen (AWS, GCP, Azure, etc)

- Dynamic Provisioning

# Access Modes

- Read Only (ROX)
- Read Write Once (RWO)
- Read Write Many (RWX)

# OPENSHIFT PERSISTENT STORAGE FRAMEWORK



# OpenShift 4

# Installation

- Installer provisioned Infrastructure (IPI)
- User provisioned Infrastructure (UPI)

- AWS
- Azure
- GCP
- VMware vSphere
- OpenStack
- IBM Z
- Bare Metal

# What’s new ...

- Neuer Installer
- Over-the-air Updates
- Cluster Autoscaling
- Neues User Interface
- Developer CLI Tools (ODO)
- Service Mesh (Istio)
- Quay Image Registry
- Operators & Operator Hub

# Installation Experiences

## OPENSHIFT CONTAINER PLATFORM

### Full Stack Automated

Simplified opinionated "Best Practices" for cluster provisioning

Fully automated installation and updates including host container OS.

Red Hat Enterprise Linux CoreOS

### Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries

Red Hat Enterprise Linux CoreOS

Red Hat Enterprise Linux

## HOSTED OPENSHIFT

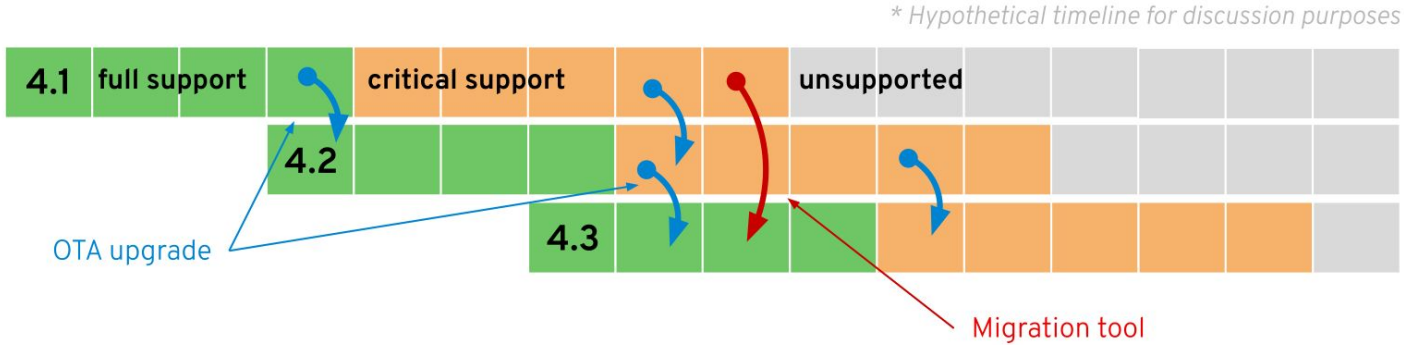
### Azure Red Hat OpenShift

Deploy directly from the Azure console. Jointly managed by Red Hat and Microsoft Azure engineers.

### OpenShift Dedicated

Get a powerful cluster, fully Managed by Red Hat engineers and support.

# OpenShift 4 Upgrades




**OTA Upgrades**
Works between two minor releases in a serial manner.

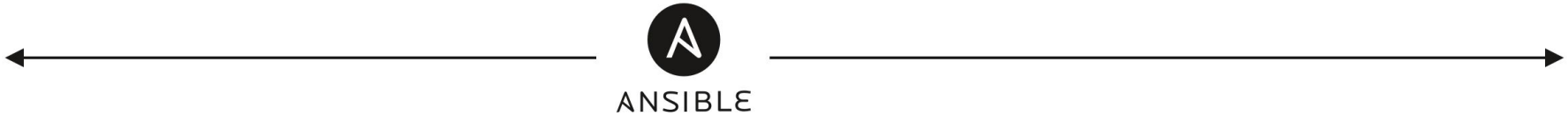
**Happy path = migrate through each version**
On a regular cadence, migrate to the next supported version.

**Optional path = migration tooling**
If you fall more than two releases behind, you must use the application migration tooling to move to a new cluster.

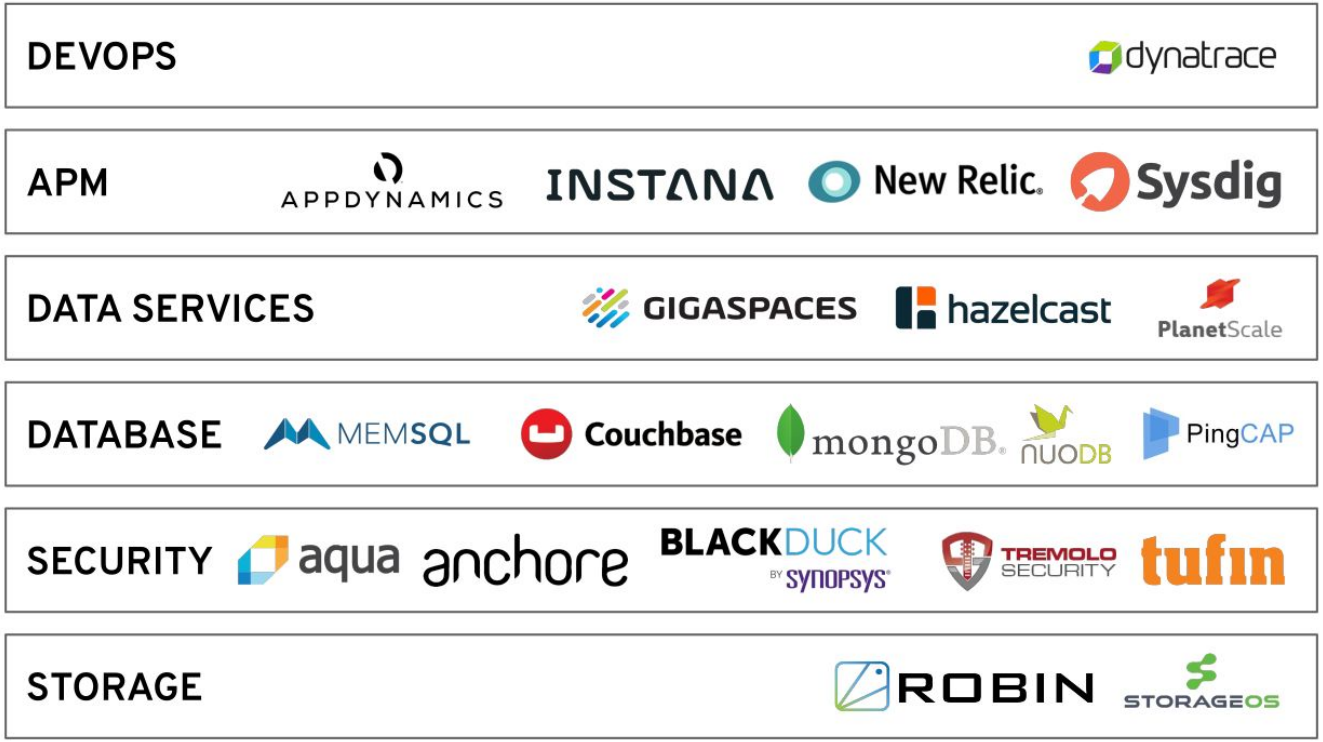
**Current minor release**
Full support for all bugs and security issues
1 month full support overlap with next release to aid migrations

**Previous minor release**
Fixes for critical bugs and security issues for 5 months

| 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 Certified Operators



# You don’t need to know everything, just know where to look

https://docs.openshift.com

https://learn.openshift.com

https://kubernetes.io/docs/