

# THE TWELVE FACTORS

## I. Codebase

One codebase tracked in revision control, many deploys

## II. Dependencies

Explicitly declare and isolate dependencies

## III. Config

Store config in the environment

## IV. Backing services

Treat backing services as attached resources

## V. Build, release, run

Strictly separate build and run stages

## VI. Processes

Execute the app as one or more stateless processes

## VII. Port binding

Export services via port binding

## VIII. Concurrency

Scale out via the process model

## IX. Disposability

Maximize robustness with fast startup and graceful shutdown

## X. Dev/prod parity

Keep development, staging, and production as similar as possible

## XI. Logs

Treat logs as event streams

## XII. Admin processes

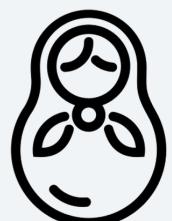
Run admin/management tasks as one-off processes

<https://12factor.net/>

# Red Hat OpenShift



OPENSIFT



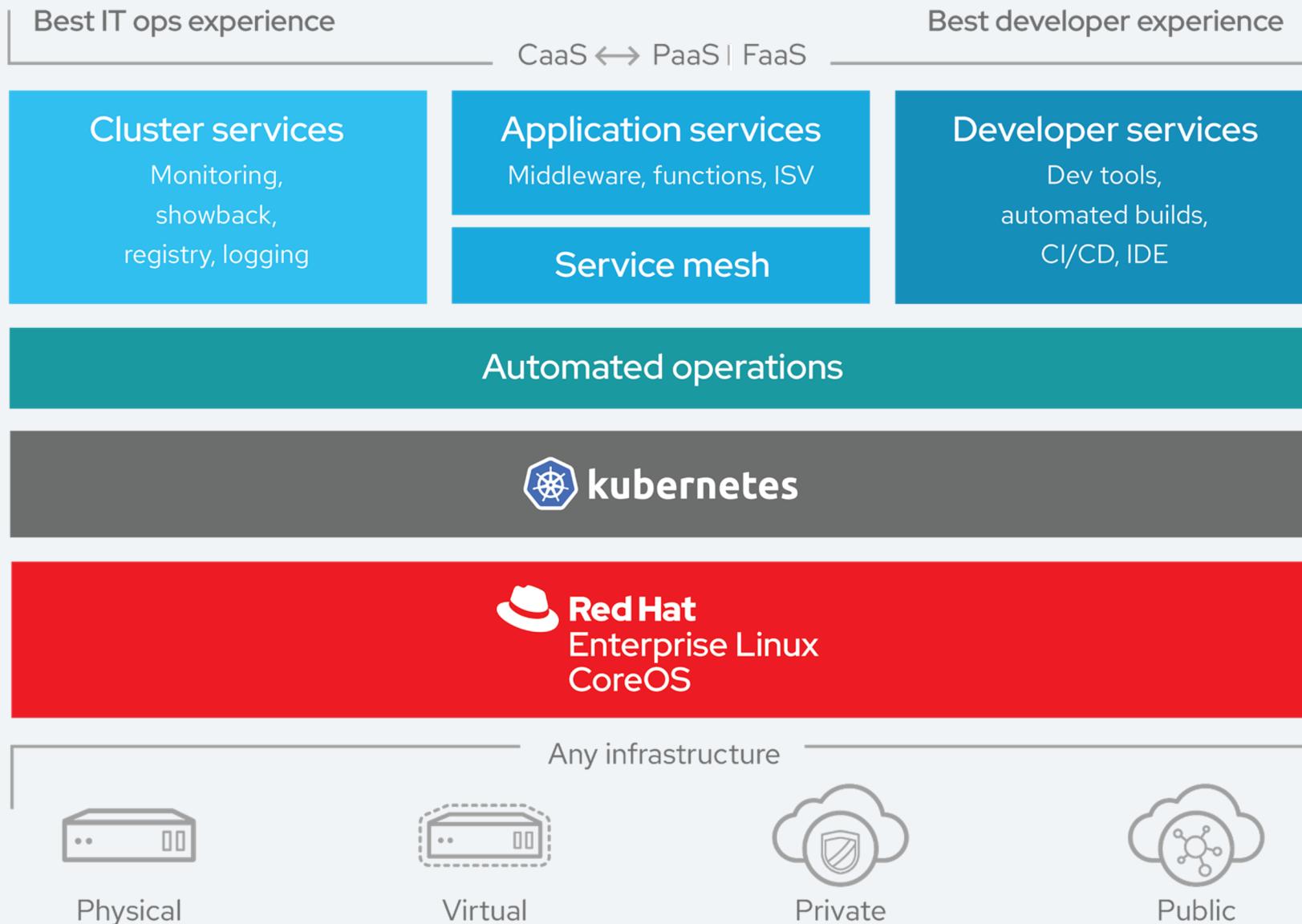
Red Hat  
OpenShift Container Platform

Project: adevconsole1 Application: all applications

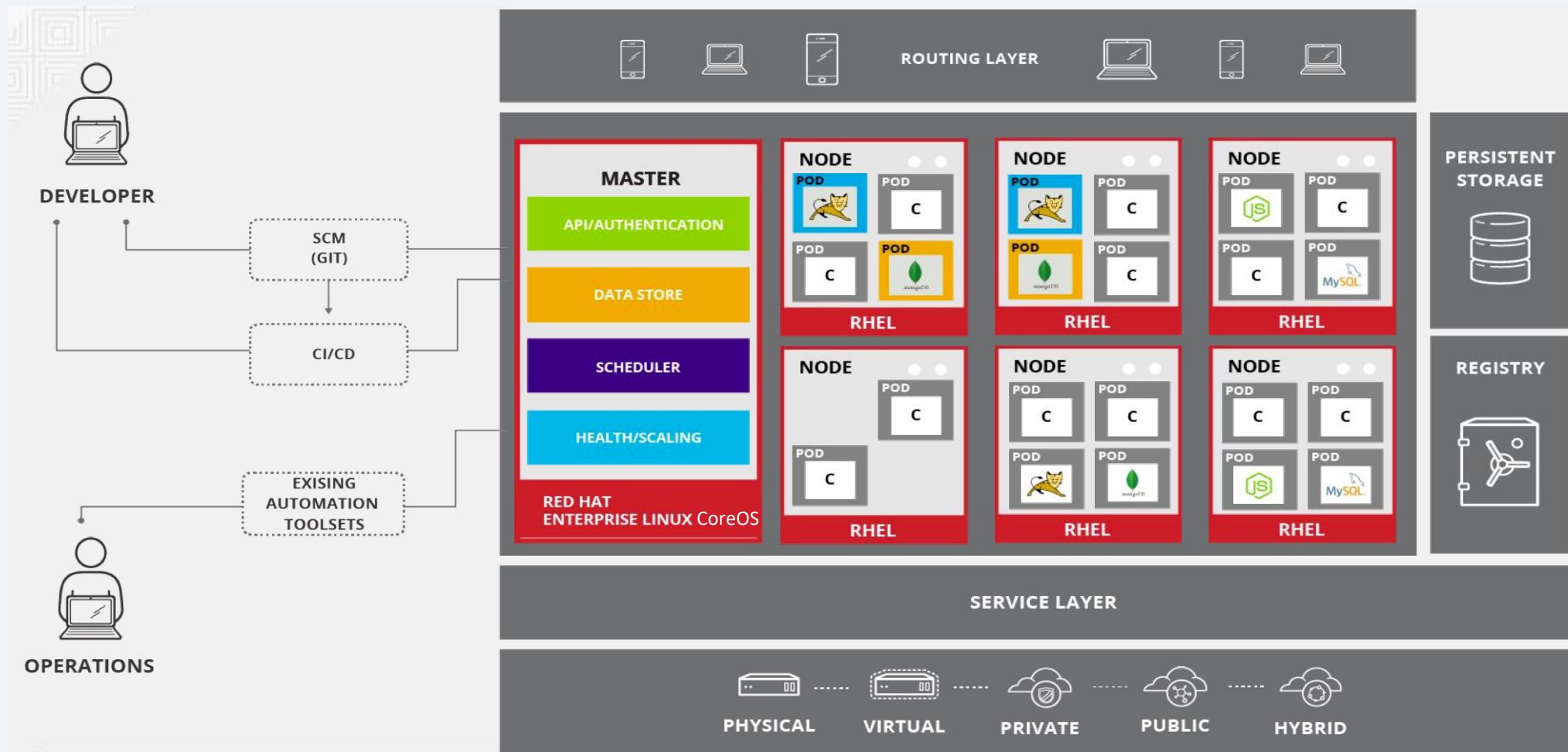
The screenshot shows the Red Hat OpenShift web interface. The top navigation bar includes the Red Hat logo, the text "Red Hat OpenShift Container Platform", and dropdown menus for "Project: adevconsole1" and "Application: all applications". On the left, a sidebar menu is open under the "Developer" section, showing options like "Administrator", "Developer" (which is selected), "Topology", "Builds", "Pipelines", and "Advanced". Under "Advanced", there are links for "Projects", "Events", and "Search". The main content area displays a "Topology" view with several service icons: a "caching-service..." pod with a teal 'M' icon, a "backend" pod with a cartoon character icon, a "frontend-b..." pod with a green 'node.js' icon, and two other partially visible pods with red 'K' and blue 'K' icons. A black arrow points from the "backend" pod towards the "frontend-b..." pod.

# What is Red Hat OpenShift?

Red Hat OpenShift is Kubernetes for business, in Hybrid Cloud mode.



# OpenShift Architecture



# Kubernetes vs OpenShift

## What does Kubernetes need to put it into production?

- Operating system
- Container Runtime
- Image registry
- Advanced networking
- Log Management
- Metrics and monitoring
- ...

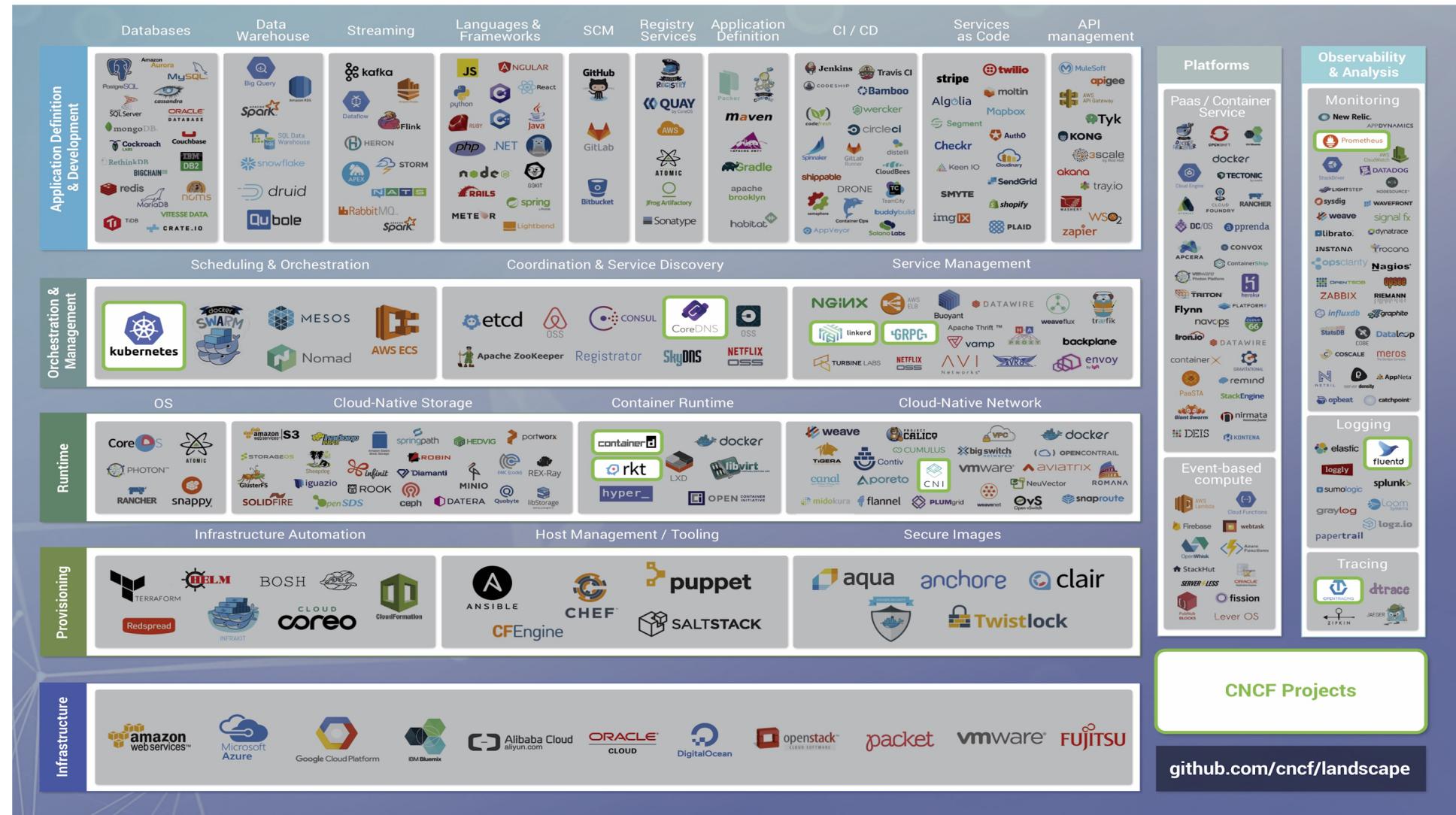


The customer (or third-party) must configure, integrate, operate and support additional components to be fully operational.



You need to choose from in order to run containers in production.

Then you will have to vet, test, and select individual technologies throughout the stack.



**THIS IS A MAP OF ALL THE CLOUD-NATIVE COMPONENTS YOU NEED TO CHOOSE FROM IN ORDER TO RUN CONTAINERS IN PRODUCTION.** It covers everything from infrastructure through application / developer services, as well as observability tools for the whole stack. It gets bigger and more complicated every few weeks, when a new revision rolls out. (Side note: this does not include serverless frameworks - that's an entirely separate map). Latest version of this map is always here: <https://github.com/cncf/landscape>



# OpenShift

## Cloud Platform Agnostic



### kubernetes



Azure Kubernetes Service (AKS)



IBM Cloud  
Kubernetes Service



Amazon  
EKS



Google Kubernetes Engine

Vendor-specific  
Architecture/Configuration



**RED HAT<sup>®</sup>**  
**OPENSHIFT**  
Container Platform



IBM Cloud Google Cloud

Identical Architecture  
/Unified Experience



# kubernetes

## Production-Grade Open Source Project

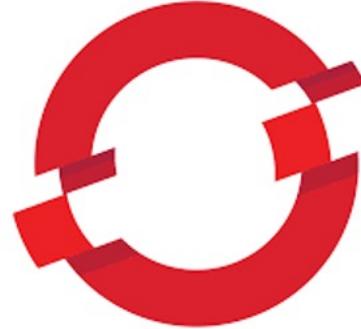
Quarterly minor releases, no long-term Support

Community support

Platform certification: (AKS, EKS, GKE, IKS)

core framework / limited security

platform or user responsible to integrate beyond core



## OPENSIFT

## Production-Grade Open Source based Product

Quarterly releases, support for major release 3+ years

Enterprise support

Ecosystem certification: platform and app containers

Kubernetes core plus abstractions / console / security

Opinionated integration of common features

## Project vs. Product



# OpenShift

## CLI vs. Console

**Workload Status**

Deployment	Pods	Replica Sets
3	3	3

**Deployments**

Name	Namespace	Labels	Pods	Created	Images
myapp-deploy	default	app: myapp-deploy	4 / 4	11 hours ago	myapp:v3

**Pods**

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
myapp-deploy-7445f-7t4qk	default	app: myapp-deploy	minion-1	Running	0	0m	0B	11 hours ago



Dashboard

**Add**

Select a way to create an application, component or service from one of the options.

<b>From Git</b> Import code from your git repository to be built and deployed	<b>Container Image</b> Deploy an existing image from an image registry or image stream tag
<b>From Catalog</b> Browse the catalog to discover, deploy and connect to services	<b>From Dockerfile</b> Import your Dockerfile from your git repo to be built & deployed
<b>YAML</b> Create resources from their YAML or JSON definitions	<b>Database</b> Browse the catalog to discover database services to add to your application



vs.

Console



# OpenShift

## Key Features Over Kubernetes

### I. Automation

**Automated installation, upgrades, and lifecycle management** throughout the container stack—the operating system, Kubernetes and cluster services, and applications—**on any cloud.**

### II. Agility

**Helps teams build with speed, agility, confidence, and choice. Code in production mode anywhere** you choose to build.

### III. Security

**Security at every level** of the container stack and throughout the application lifecycle.

# Red Hat OpenShift provides a comprehensive approach to securing containers

## Linux Host Security

- RHCOS Immutable user space
- SELinux+
- LUKS volume encryption / FIPS mode
- Non-root containers

## Configuration & Lifecycle Management

- OpenShift operators manage drift
- OLM manages operator privileges & dependencies
- RH supply chain (backport fixes)
- One maintenance window for the full stack
- Upgrades with zero application downtime
- ArgoCD integration

## Authentication & Authorization

- Built-in token based authentication
- Supports 9 Identity Providers including AD/LDAP
- Pre-configured RBAC with Multi-Level Access Control
- Secrets and certificate management

## Networking Isolation

- Ingress / Egress control
- Multus CNI plugin
- Network microsegmentation

## Integrated Audit, Logging, Monitoring

- Host and Kubernetes event audit on by default
- Monitoring on by default
- Applications can use cluster monitoring

## Developer Tools

- IDE plugins for dependency analysis
- Code Ready Workspaces
- Jenkins / Tekton Pipelines

## Image Security

- RH Trusted Content with Health Index
- ImageStreams track changes to external images
- Image Scanning (Quay with Clair)
- Deployment policies (admission controllers)

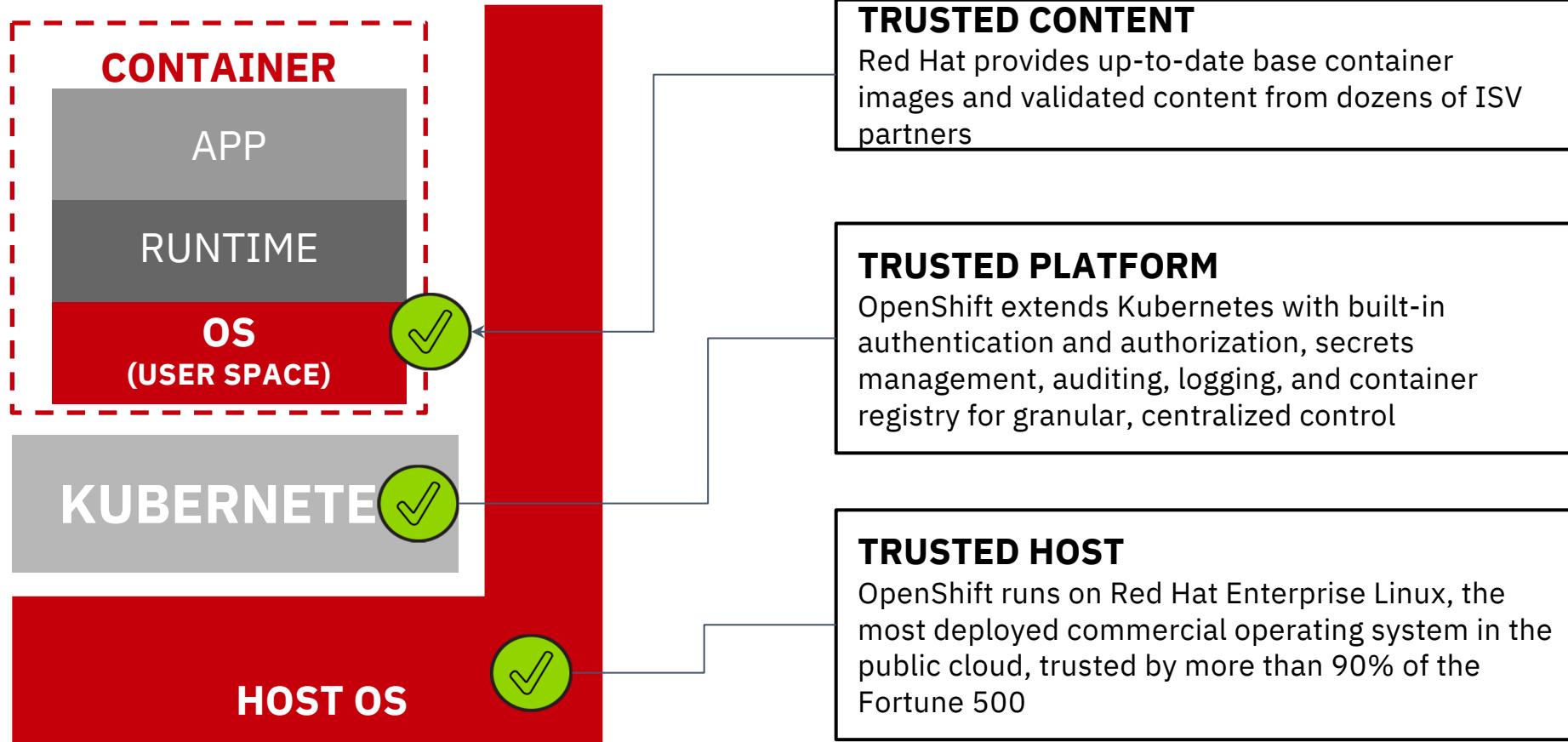
## Runtime protection

- SCC (Security Context Controls)
- No privileged containers by default
- Projects with SELinux annotations control Access to Resources
- Automate Compliance Audit and Remediation

## Data Protection

- Encrypt secrets at rest (etcd datastore)
- All traffic to master nodes is encrypted by default
- Configure cipher suites
- Encrypt east / west traffic (Service Mesh)

# Trusted Host, Content, Platform



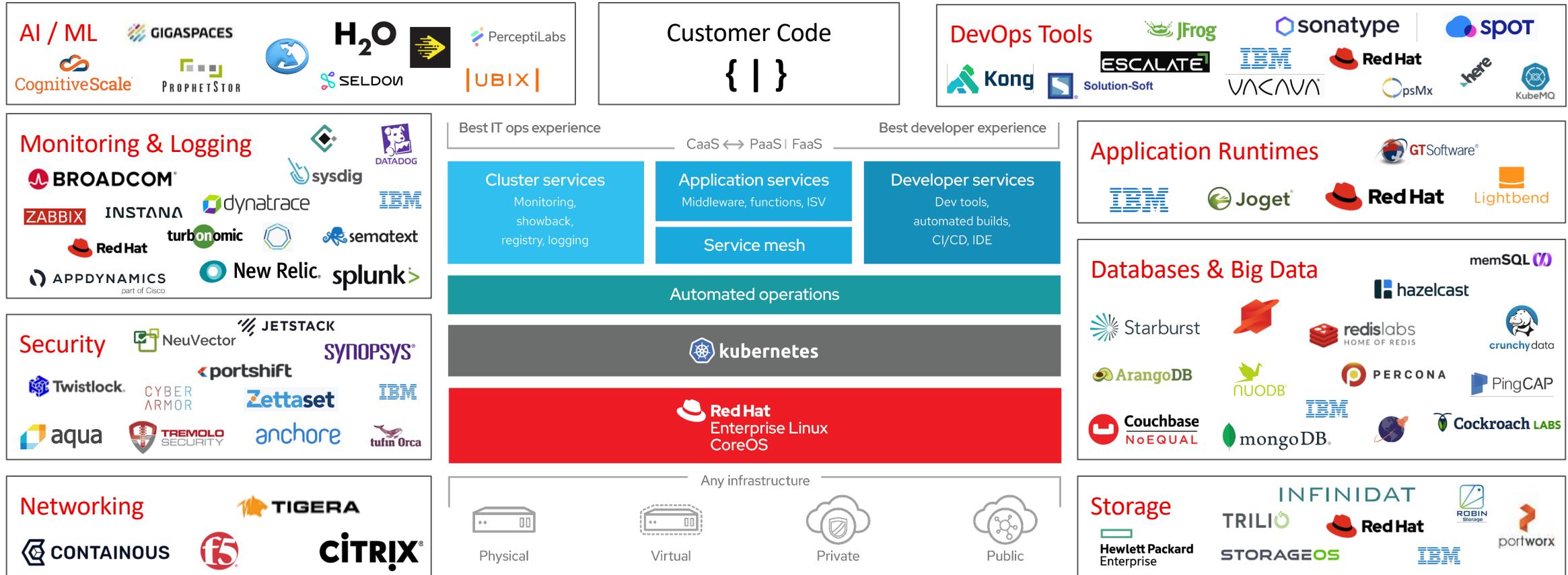
# OCP vs DIY Kubernetes

	 Red Hat OpenShift 4	 kubernetes
Container orchestration	✓ Kubernetes	✓ Kubernetes
Container image	✓ OCI-compliant/docker	✓ BYO OCI-compliant
Container runtime	✓ CRIOS/docker	✓ BYO OCI-compliant engine
Container build	✓ RHCC/S2I/dockerfile	✗
Container registry	✓ Quay/OSS docker registry	✗
Container scanner	✓ Clair	✗
CI/CD automation	✓ OCP Pipelines/Tekton	✗
IDE	✓ Che/Code Ready workspaces	✗
Web UX	✓ Web console (admin and developer perspectives)	✓ Web console
CLI UX	✓ oc/odo/kubectl	✓ kubectl
Service catalog	✓ Operators	✗
Secrets management	✓ Kubernetes Secrets	✓ Kubernetes Secrets
Supported/preferred runtime	✓ RHOAR (EAP, Spring, vert.x, node.js)	✗

# OCP vs DIY Kubernetes

	 Red Hat OpenShift 4	 kubernetes
Service mesh	✓ Istio, Jaeger, Kiali, Prometheus&Grafana	✗
Logging	✓ EFK	✗
Metrics	✓ Prometheus/Grafana	✗
Storage	✓ OpenShift Container Storage	✗
Network	✓ OVN -(TechPreview)	✗
Ingress	✓ Kubernetes Ingress/Routes	✓ Kubernetes Ingress
Ingress controller	✓ HA Proxy	✗
Egress	✓ Egress Router	✗
Authentication	✓ Kubernetes Auth/RH-IdM	✓ Kubernetes Auth
App isolation	✓ Kubernetes scheduler	✓ Kubernetes Scheduler
Infrastructure	✓ Bare metal, vSphere, KVM, OpenStack, AWS, GCP, Azure	✓ BYO Linux
Infra automation	✓ Ansible/Operators	✗
Infra management	✓ Admin Console	✗
Operating system	✓ RHEL or RHEL CoreOS	✗
CNCF Certified Kubernetes	✓	✗

# 100+ Red Hat OpenShift certified operators



# The Red Hat Marketplace difference



Any cloud. On-prem. Anywhere OpenShift runs.

Build once, deploy to any environment with software that allows for workload portability across clouds.



Certified enterprise software

All software is certified for Red Hat OpenShift and uses Kubernetes operators for built-in management logic.



Continuous support

Central support portal offers a single point of entry for any software purchased in the marketplace.



Consolidated usage tracking & spend optimization

Pool spending across clouds. Monitor license usage, expiration, and renewals on a single dashboard.

The screenshot shows the Red Hat Marketplace homepage. At the top, there's a navigation bar with links for 'Red Hat Marketplace', 'Workspace', 'Docs', 'Support', and 'About'. On the right side of the header are 'Log in' and 'Create account' buttons. Below the header, there's a search bar with the placeholder 'Find certified software for Red Hat OpenShift' and a magnifying glass icon. To the right of the search bar is a decorative illustration of a white hot air balloon with a red basket floating above clouds. On the left, there's a sidebar titled 'Product Categories' with a list of 63 products categorized into: AI/Machine Learning (4), Application Runtime (10), Big Data (3), Database (2), Developer Tools (1), Integration & Delivery (1), Logging & Tracing (1), Monitoring (1), Networking (1), Security (1), Storage (1), and Streaming & Messaging (1). The main content area displays a grid of software packages, each with a thumbnail, name, provider, description, and rating. Some examples include 'Coretex Certified' by Coretex, 'Turbonomic' by Turbonomic Inc., 'IBM Cloud Pak for Data' by IBM, 'Hazelcast In Memory C...' by Hazelcast Inc., 'Jogjet' by jogjet inc., 'Jogjet DX Open Source' by jogjet inc., 'IBM Cloud Pak for App...' by IBM, 'Lightbend Platform Op...' by lightbend inc., 'Starburst Enterprise PR...' by Starburst Data, 'Gigaspaces InsightEdge' by Gigaspaces Technologies Inc., 'Robin Storage' by Robin Storage, 'IBM Cloud Pak for Data' by IBM, 'Crunchy PostgreSQL L...' by Crunchy Data Solutions Inc., 'memSQL' by MemSQL Inc., 'CockroachDB' by Cockroach Lab, and 'PlanetScaleDB Operator' by PlanetScale. Each package card includes a 'View All' link and a star rating. At the bottom of the page is a dark footer bar with the text 'Sell on Red Hat Marketplace' and 'Showcase your product to millions of potential clients, customers, sellers and developers'. On the far right of the footer is a 'Get started' button.

# Operators enable a cloud-like experience everywhere you run Kubernetes



|



PostgreSQL

- Containerized



Cloud database

- Containerized
- Cloud storage ready
- Replicated
- Backup
- Automated updates



|



OPERATOR  
FRAMEWORK

- Containerized
- Container storage ready
- Replicated
- Backup
- Automated updates
- Enhanced observability
- Customization
- Local development
- Fully open source
- Any Kubernetes
- Certified on OpenShift

