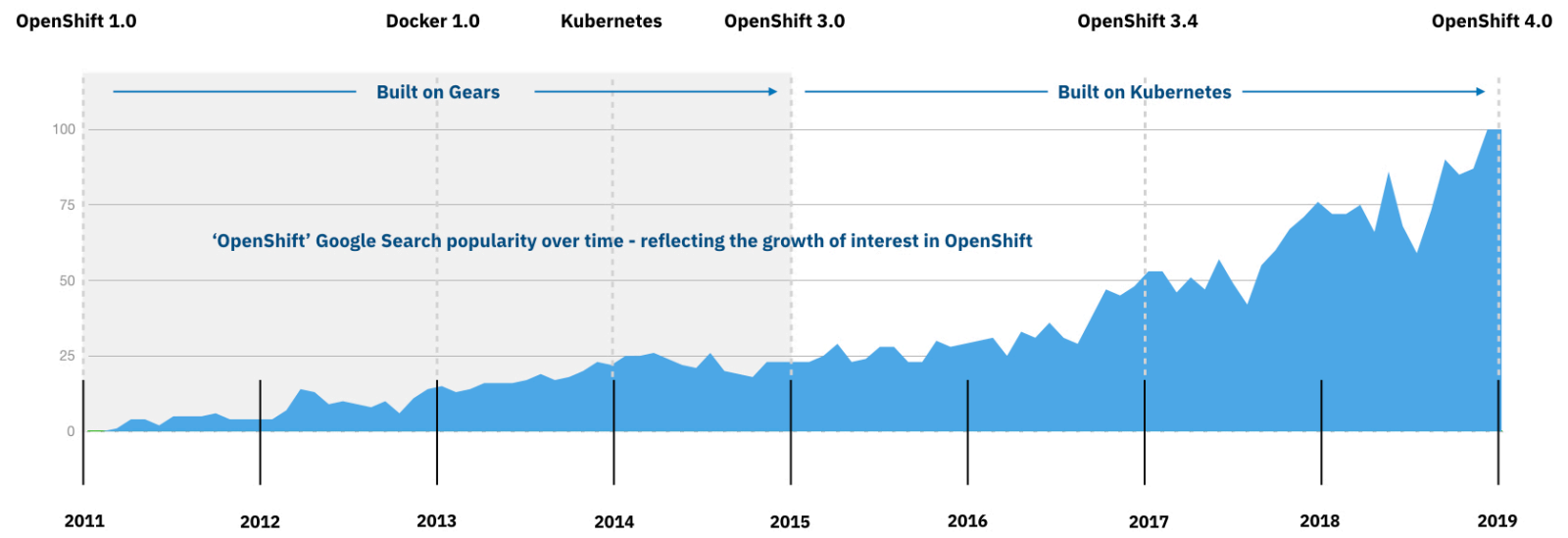
What is OpenShift

- A leading hybrid cloud, enterprise Kubernetes application platform trusted by 1,700+ organizations
- Vendor-neutral Kubernetes platform (AWS, Azure, GCP, VMware vSphere, IBM Cloud, OpenStack, bare metal, etc.)
  - AWS first ☺
  - Azure next
- Tags for OpenShift
  - Red Hat
  - Kubernetes
  - Container
  - Cloud
  - PaaS

History of OpenShift



Linux, building on Red Hat's heritage and deep expertise in Linux and the reliability of **Red Hat Enterprise Linux** which served as the foundation for OpenShift 3

RedHat and Docker announced a collaboration around

Fedora RedHat Enterprise Linux and OpenShift

IBM begin contributing code to the Docker Open Source Project

Kubernetes (κυβερνήτης, Greek for "governor", "helmsman" or "captain")[3] was founded by Joe Beda, Brendan Burns and Craig McLuckie,[6] who were quickly joined by other Google engineers including Brian Grant and Tim Hockin, and was first announced by Google in mid-2014.[7] Its development and design are heavily influenced by Google's Borg system.

IBM contributing code to the Kubernetes Open Source Project

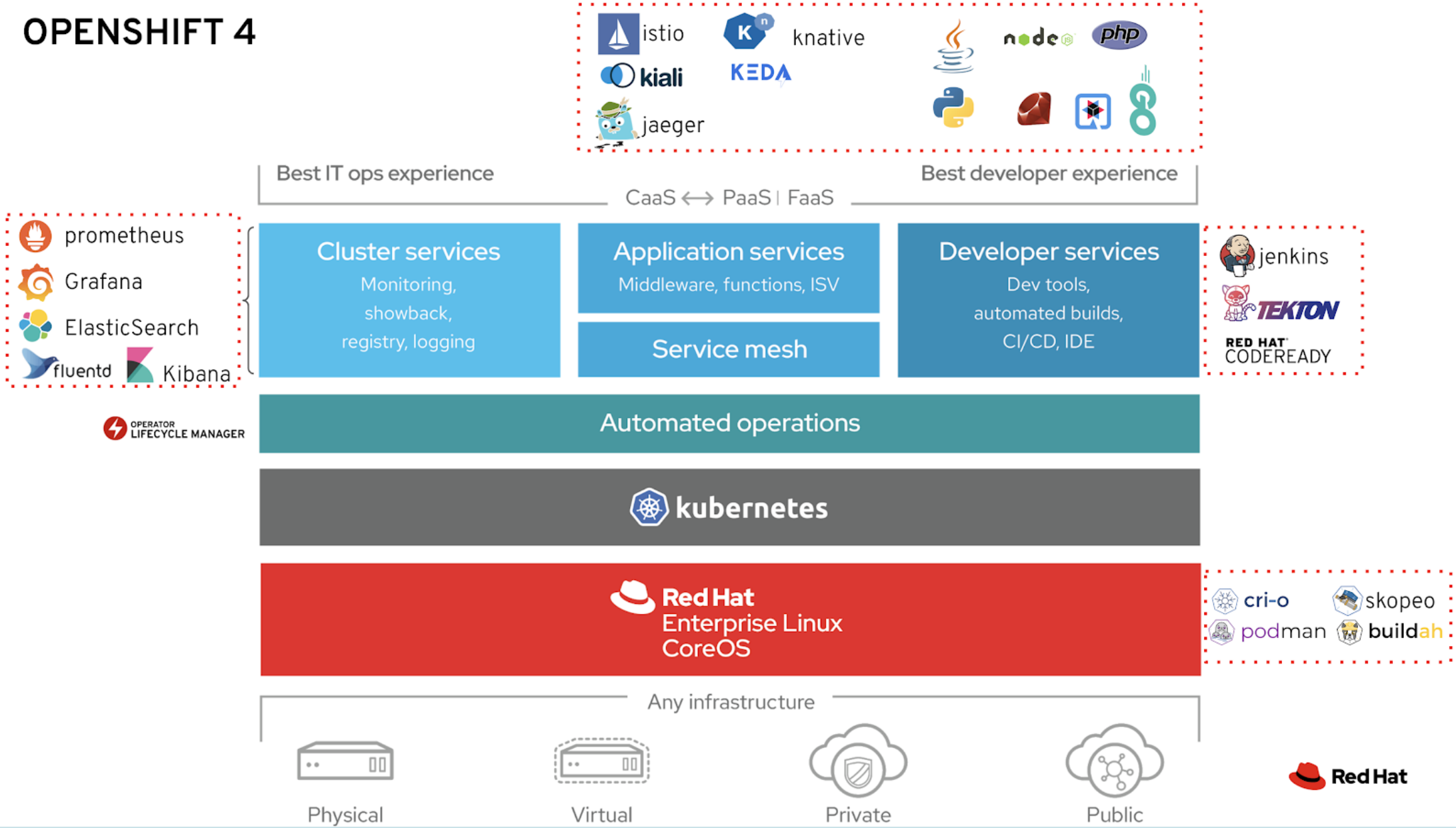
Kubernetes for the Enterprise

includes Kubernetes core services, along with Prometheus, Grafana, Elasticsearch, software defined networking, storage, registry and other components that make up the OpenShift Kubernetes platform.

OpenShift vs Kubernetes: What's the Difference?

- Some analogies
  - Car vs Engine
    - Kubernetes is the engine (framework, project) that drives OpenShift
    - OpenShift is the complete car (product) that gets you where you want to
  - Linux kernel vs Linux distribution
    - Kubernetes is the 'kernel' of distributed systems
    - OpenShift is the distribution that focuses on the experience of both developers and administrators
      - Kubernetes plus many other upstream projects => project OKD, upstream of OpenShift
      - OKD plus some other projects and RH support => product OpenShift

OPENSHIFT 4


istio
knative
kiali
KEDA
jaeger
node
php
Best IT ops experience
Best developer experience
CaaS ⟷ PaaS | FaaS
prometheus
Grafana
ElasticSearch
fluentd
Kibana
Cluster services
Monitoring, showback, registry, logging
Application services
Middleware, functions, ISV
Service mesh
Developer services
Dev tools, automated builds, CI/CD, IDE
jenkins
TEKTON
RED HAT CODEREADY
OPERATOR LIFECYCLE MANAGER
Automated operations
kubernetes
Red Hat Enterprise Linux CoreOS
cri-o
skopeo
podman
buildah
Any infrastructure
Physical
Virtual
Private
Public
Red Hat

OpenShift vs Kubernetes: Major OpenShift 4 benefits

- Trusted OS foundation: RHEL CoreOS or RHEL
  - OpenShift 4 control panel can only be deployed on Red Hat CoreOS (RHCOS) – immutable, lower footprint, optimized for running containers and managing Kubernetes clusters at scale
  - Workers can be deployed on RHCOS or RHEL
  - RHEL CoreOS and RHEL make the perfect foundation for OpenShift to run anywhere from bare-metal to private and public clouds with the same experience
- Automated operations
  - Automated installation and day-2 Operations which makes it easier to administrate, upgrade, and provide an enterprise container platform
  - Ubiquitous usage of operators to make all possible
  - Built-in Operator Lifecyle Manager (OLM) and rich ecosystem of operators

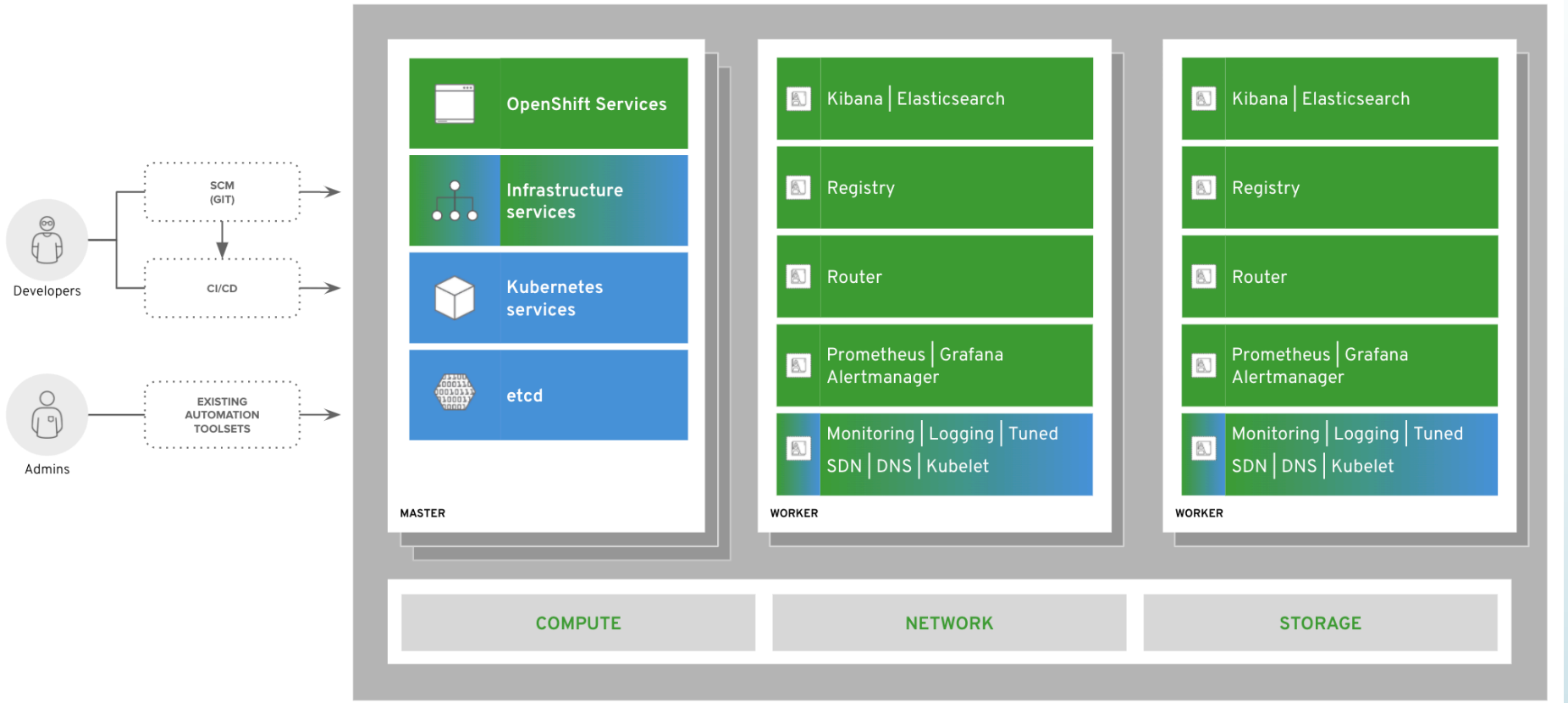
OpenShift vs Kubernetes: Major OpenShift 4 benefits

- Cluster services
  - Centralized logs: EFK (ELK) stack
  - Metrics and Monitoring: Prometheus, Grafana
  - Built-in OAuth provider
  - Strict security model, integral RBAC
- Application services
  - Support microservices and serverless architectures
    - OpenShift Service Mesh provides Istio, Kiali, and Jaeger out-of-the-box to support microservices adoption
    - OpenShift Serverless includes Knative and Keda (for Azure functions)
  - Allow to migrate legacy VMs to OpenShift by using Container Native Virtualization (tech preview)

OpenShift vs Kubernetes: Major OpenShift 4 benefits

- Developer services
  - Built-in OperatorHub which offers a catalog of over 180 services delivered through Operators
  - Developer perspective of web console dedicated to developers
  - Allow easy application deployment from different sources (git source, binary, external registry, Dockerfile, …)
  - Application topology view
  - CodeReady Containers: OpenShift 4 running on laptop
  - Out-of-the-box CI/CD features: Jenkens, Tekton
- Deep expertise Red Hat has in Kubernetes ecosystem
  - Comprehensive QA process in OpenShift release cycle
  - Professional support
    - Critical bugs fixed earlier than Kubernetes and supported/patched on much longer time frame

OpenShift 4 Architecture



More Detailed Comparison between OpenShift and Kubernetes

	OpenShift	Kubernetes
Product or project	Product with paid support	Open-source project
Supported OS	RHCOS and RHEL	Any Linux
Installation	Simply installation by dedicated operator (OpenShift 4)	The tool of your choice
Security	Very strict, built-in authentication and authorization model, integral RBAC for long	Less strict
Service provisioning	Operator, templates	Helm
Deployment approach	DeploymentConfig and Deployment	Deployment
Router vs Ingress	Router (and support Ingress to Router translation)	Ingress

More Detailed Comparison between OpenShift and Kubernetes

	OpenShift	Kubernetes
Container image management	ImageStream	
Integrated CI/CD	Jenkins, Tekton (tech preview), source-to-image (S2I)	
Project vs namespace	Projects are Kubernetes namespaces with additional features	Namespace
User experience – CLI	oc (kubectl plus some convenient features)	kubectl
User experience – web intreface	Nice-looking web console with admin and developer perspectives, supporting SSO to access external services	Primitive dashboard UI
Networking	Native networking solution	Some components require 3rd party plugins

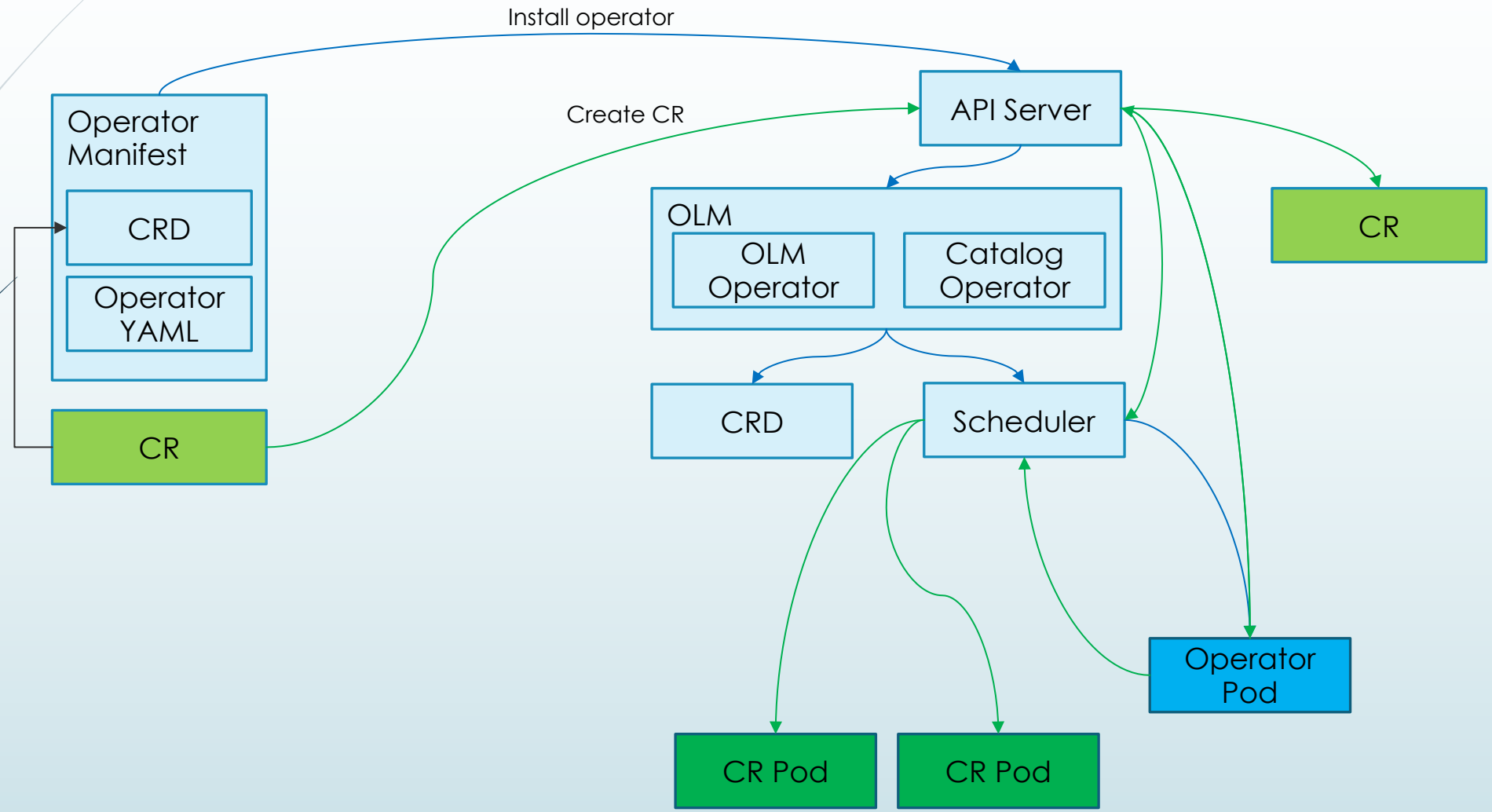
Operators

- Infrastructure as Code (IaC)
  - Probably the most important benefit Kubernetes has enabled
  - Each Kubernetes resource (pod, service, deployment, etc.) is described in machine readable YAML format (a.k.a., desired status)
  - When the YAML is 'applied' to Kubernetes cluster, Kubernetes will 'make it happen' (actual status)
  - If the 'desired' status is updated, Kubernetes will make the 'actual' status matched
  - YAML files can be source-controlled as application artifacts
  - Resources described in YAML format can be easily moved from one place to the other

Operators

- Introduced by CoreOS in 2016
- Operators are powerful extension of Kubernetes' IaC enablement
- Innovative approach to automate infrastructure and application management tasks using Kubernetes as the automation engine
- Kubernetes Operator
  - Kubernetes-native application
    - Puts all operational knowledge into Kubernetes primitives
    - Administrators, shell scripts, automation software (e.g. Ansible®) now in Kubernetes pods
    - Integrates natively with Kubernetes concepts and APIs
  - Are pods with operator code that interact with Kubernetes API server
  - Run "reconciliation loops" to check on application service
    - Make sure user-specified state of objects is achieved
  - Manage all deployed resources and your application
  - Act as application-specific controllers
  - Extend Kubernetes API with Custom Resource Definition (CRD)

Operators: My Understanding



Operators: Custom Resource Definition (CRD)

- A CRD extends Kubernetes API by defining the schema of a new custom resource (CR)
  - Kubernetes API server then has new endpoints for the CR

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  creationTimestamp: null
  name: tomcats.tomcat.apache.org
spec:
  group: tomcat.apache.org
  names:
    kind: Tomcat
    listKind: TomcatList
    plural: tomcats
    singular: tomcat
  scope: Namespaced
  subresources:
    status: {}
  validation:
    openAPIV3Schema:
      properties:
        apiVersion:
          type: string
        kind:
          type: string
        metadata:
          type: object
        spec:
          type: object
        status:
          type: object
  version: v1alpha1
  versions:
  - name: v1alpha1
    served: true
    storage: true
```

Operators: Custom Resource (CR)

- Custom Resources (CRs) can be created based on CRD
- Operator watches for creation of CR and reacts by creating all resources CR represents

```
apiVersion: tomcat.apache.org/v1alpha1
kind: Tomcat
metadata:
  name: mytomcat
spec:
  replicas: 2
  image: tomcat:latest
  imagePullPolicy: IfNotPresent
  webArchiveImage: sampleapp/webarchive:1.0
  deployDirectory: /usr/local/tomcat/webapps
```

Operators: Custom Resource Creation and Management

- Create CR in OpenShift
  - `oc create -f mytomcat.yaml`
- Manipulate and examine CR
  - `oc get tomcats`
  - `oc describe tomcat mytomcat`
- Delete CR
  - `oc delete tomcat mytomcat`

Operators in OpenShift 4 are Ubiquitous

- Installation of OpenShift 4 itself
- Cluster configuration
- Cluster upgrade (components, host OS)
- Cluster autoscaling by provisioning or destroying nodes
- OLM
- OperatorHub
- “Everything as Code”

Operators

- Operator Framework
  - Operator SDK
    - Developers build, package, test operator
    - No knowledge of Kubernetes API complexities required
  - Operator Lifecycle Manager (OLM)
    - Helps install, update, manage life cycle of all operators in cluster
  - Operator Metering
    - Usage reporting for Operators and resources within Kubernetes

Operators

- OperatorHub.io
  - Kubernetes Internet community for sharing Operators
  - Works for any Kubernetes environment
  - Packages Operators for easy deployment and management
  - Publicizes Operators and enables adoption
  - Uses OLM to install, manage, update Operators

Operators vs Helm

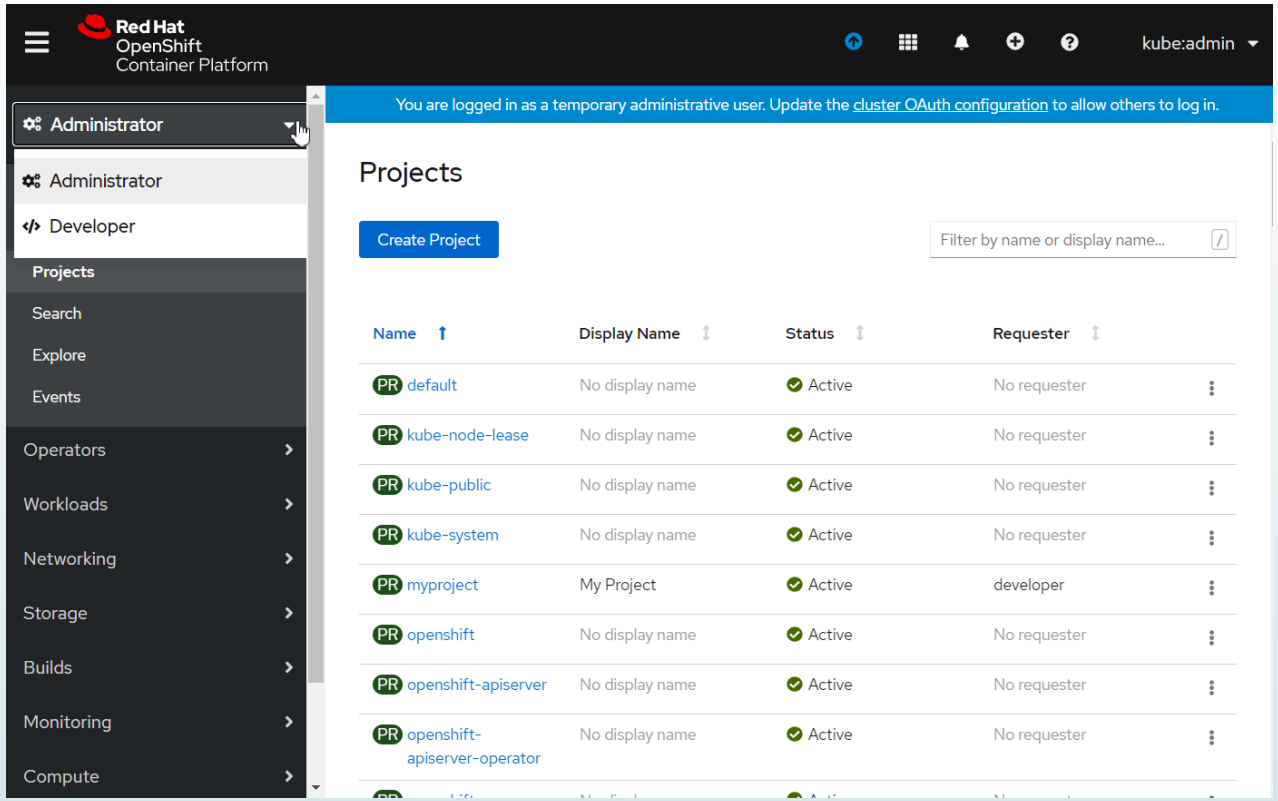
	Helm Chart	Operator
Packaging	✓	✓
App Installation	✓	✓
App Update (kubernetes manifests)	✓	✓
App Upgrade (data migration, adaption, etc)	-	✓
Backup & Recovery	-	✓
Workload & Log Analysis	-	✓
Intelligent Scaling	-	✓
Auto tuning	-	✓

CLI (oc vs kubectl)

- OpenShift CLI tool ‘oc’ is command compatible with ‘kubectl’
- ‘oc’ offers extra features and simplicity
  - ‘oc’ has support of logging in to OpenShift cluster (oc login)
  - ‘oc’ lets you switch your context between projects/namespaces (oc project myproject)
  - ‘oc’ allows you to list out **your** namespaces easily (oc projects)
  - ‘oc’ creates default RoleBindings alongside with a new project (oc new-project myproject)
  - ‘oc’ allows you to build container images and deploy applications from source code or binaries (a.k.a., S2I) with one single command (oc new-app)

Web Console

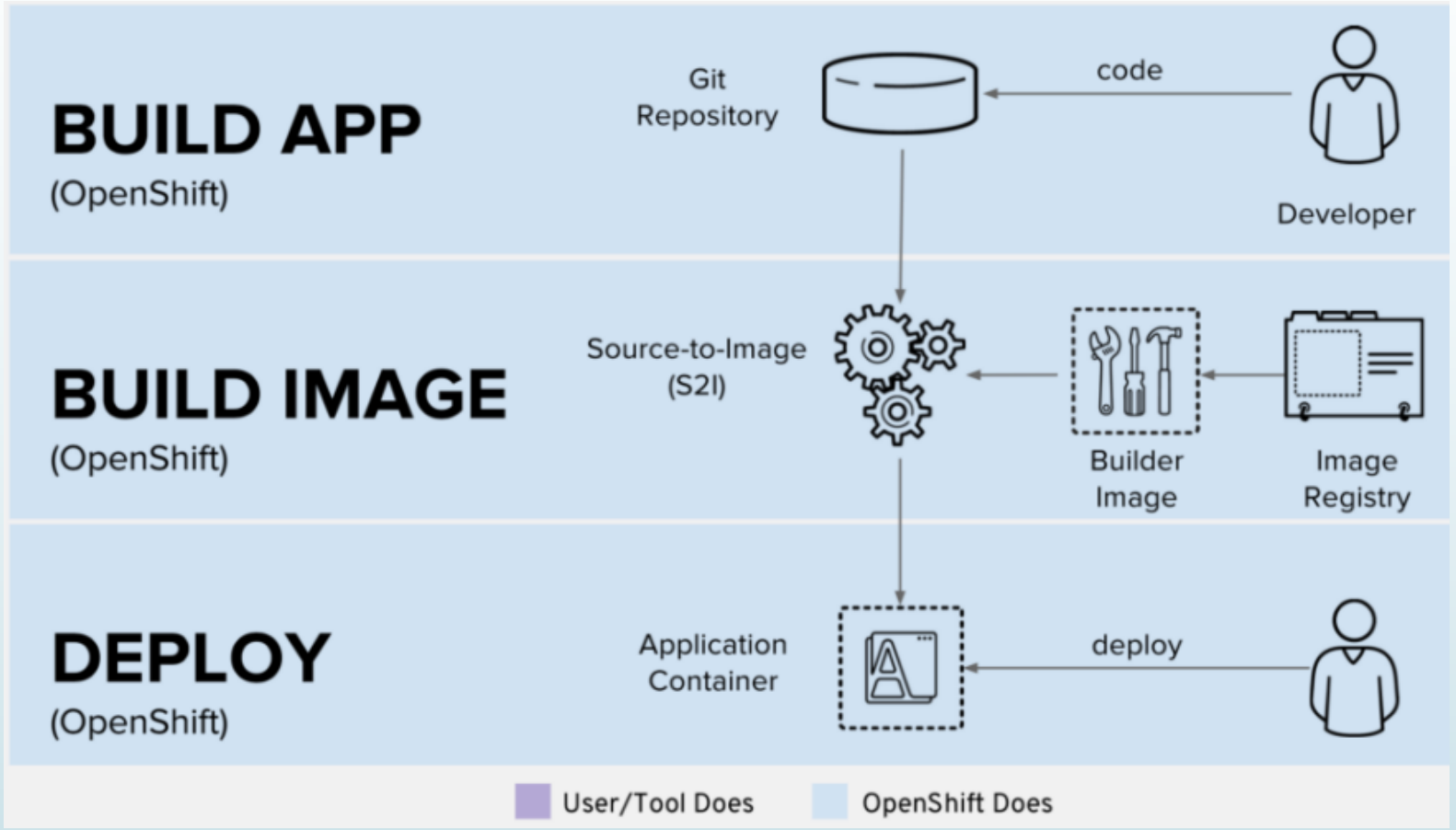
- Two perspectives
  - Administrator
  - Developer
- Run as pods



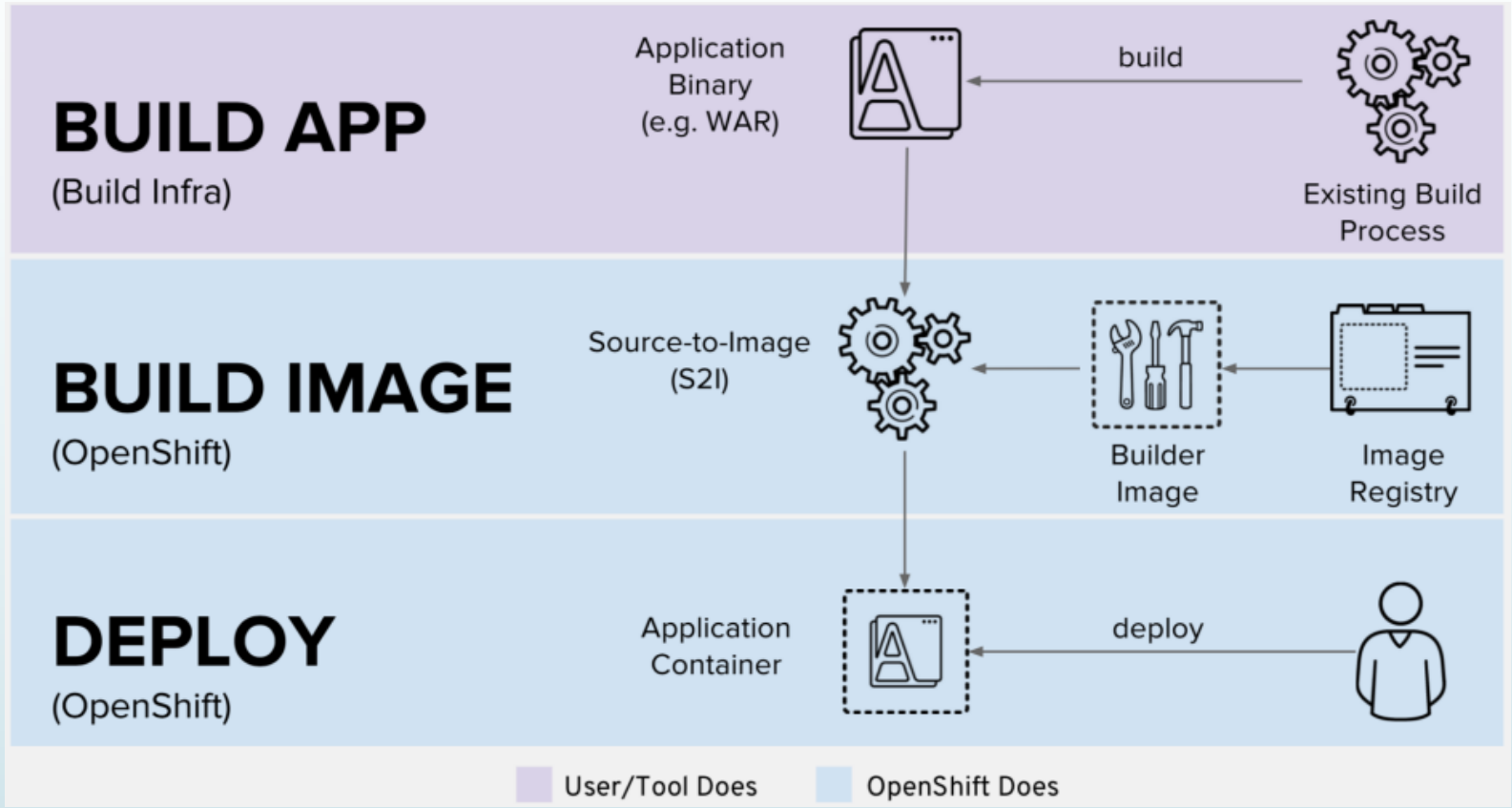
Build and Deploy Container Images in OpenShift

- Deploy source code
- Deploy application binary
- Deploy container image

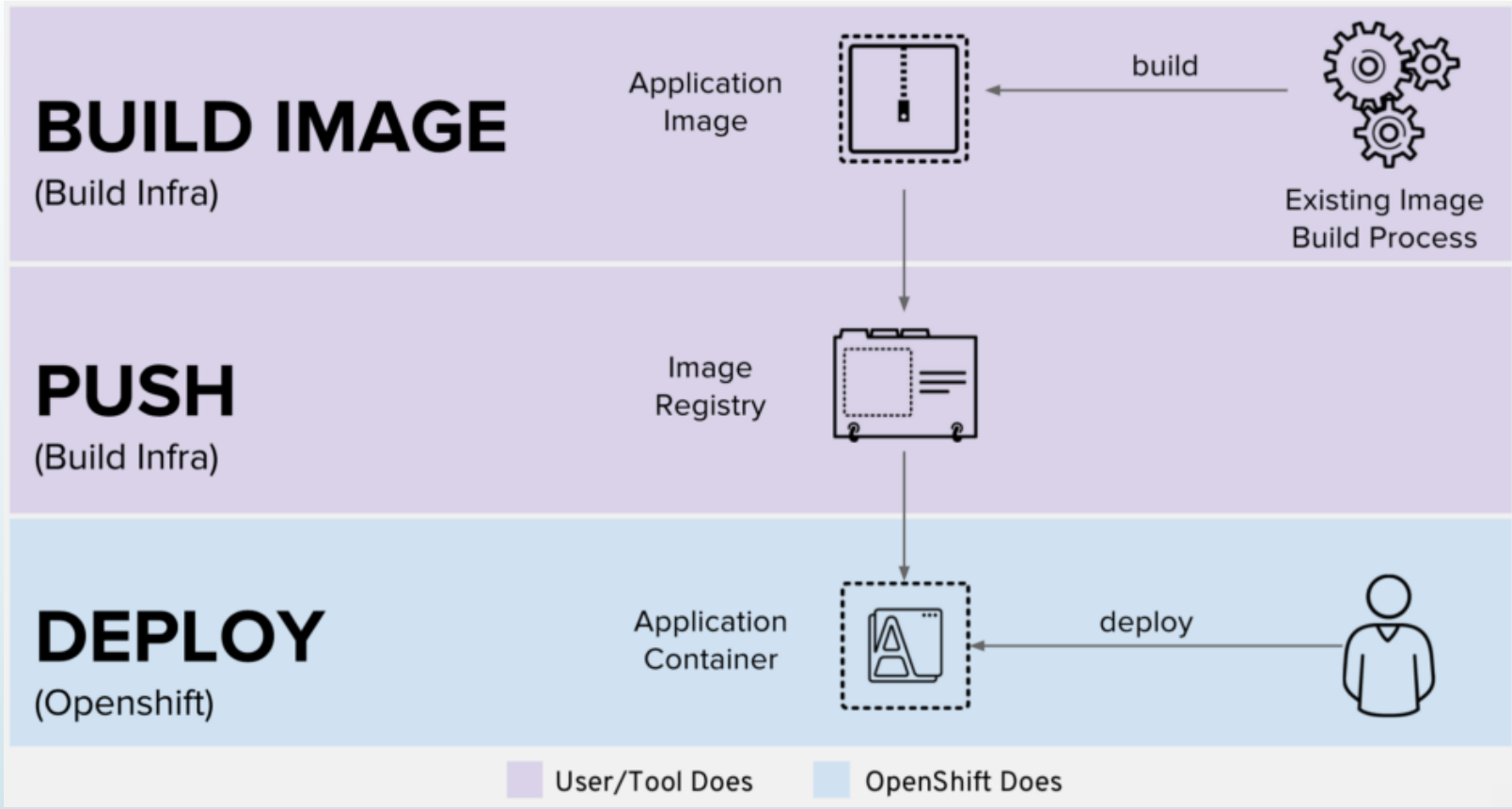
Deploy Source Code with S2I



Deploy App Binary with S2I



Deploy Container Image



References

- Red Hat: https://en.wikipedia.org/wiki/Red_Hat
- List of mergers and acquisitions by IBM: https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions_by_IBM
- Red Hat CEO Paul Cormier Talks about IBM and His Vision for the Future: https://www.datacenterknowledge.com/open-source/red-hat-ceo-paul-cormier-talks-about-ibm-and-his-vision-future
- Jim Whitehurst: https://en.wikipedia.org/wiki/Jim_Whitehurst
- James Whitehurst: https://www.redhat.com/en/about/company/management/james-whitehurst
- Jim Whitehurst Becomes President of IBM. Why? He Gets Culture. https://www.forbes.com/sites/jonobacon/2020/01/31/jim-whitehurst-becomes-president-of-ibm-why-he-gets-culture/#253469c36394
- Why IBM Choosing Arvind Krishna Over Jim Whitehurst Was Wise: https://www.itprotoday.com/linux/why-ibm-choosing-arvind-krishna-over-jim-whitehurst-was-wise

References

- What is OpenShift: https://www.openshift.com/learn/what-is-openshift
- What is Red Hat OpenShift? (31-second video) https://www.openshift.com/blog/what-is-red-hat-openshift
- OpenShift: https://en.wikipedia.org/wiki/OpenShift
- OpenShift success stories: https://www.openshift.com/learn/success-stories/
- A brief history of Kubernetes, OpenShift, and IBM: https://developer.ibm.com/technologies/containers/blogs/a-brief-history-of-red-hat-openshift/
- Red Hat OpenShift Container Platform Life Cycle Policy: https://access.redhat.com/support/policy/updates/openshift
- Kubernetes commit contribution: https://www.stackalytics.com/cncf?module=kubernetes&date=all

References

- OpenShift and Kubernetes: What's the difference? https://www.redhat.com/en/blog/openshift-and-kubernetes-whats-difference
- Enterprise Kubernetes with OpenShift (Part one): https://www.openshift.com/blog/enterprise-kubernetes-with-openshift-part-one
- OpenShift vs Kubernetes: What are the differences? https://www.whizlabs.com/blog/openshift-vs-kubernetes/
- 10 most important differences between OpenShift and Kubernetes: https://cloudowski.com/articles/10-differences-between-openshift-and-kubernetes/
- OpenShift vs. Kubernetes: Are they really in competition? https://www.padok.fr/en/blog/kubernetes-openshift

References

- Kubernetes Operators: https://www.redhat.com/cms/managed-files/cl-oreilly-kubernetes-operators-ebook-f21452-202001-en_2.pdf
- Operators: https://www.openshift.com/learn/topics/operators
- OperatorHub.io: https://operatorhub.io/
- Honest review of OpenShift 4: https://cloudowski.com/articles/honest-review-of-openshift-4/
- Tomcat-operator: https://github.com/kube-incubator/tomcat-operator
- Red Hat OpenShift 4 on your laptop: https://haralduebele.blog/2019/09/13/red-hat-openshift-4-on-your-laptop/
- Product Documentation for Red Hat CodeReady Containers 1.11: https://access.redhat.com/documentation/en-us/red_hat_codeready_containers/1.11/

Thank you!

Q&A