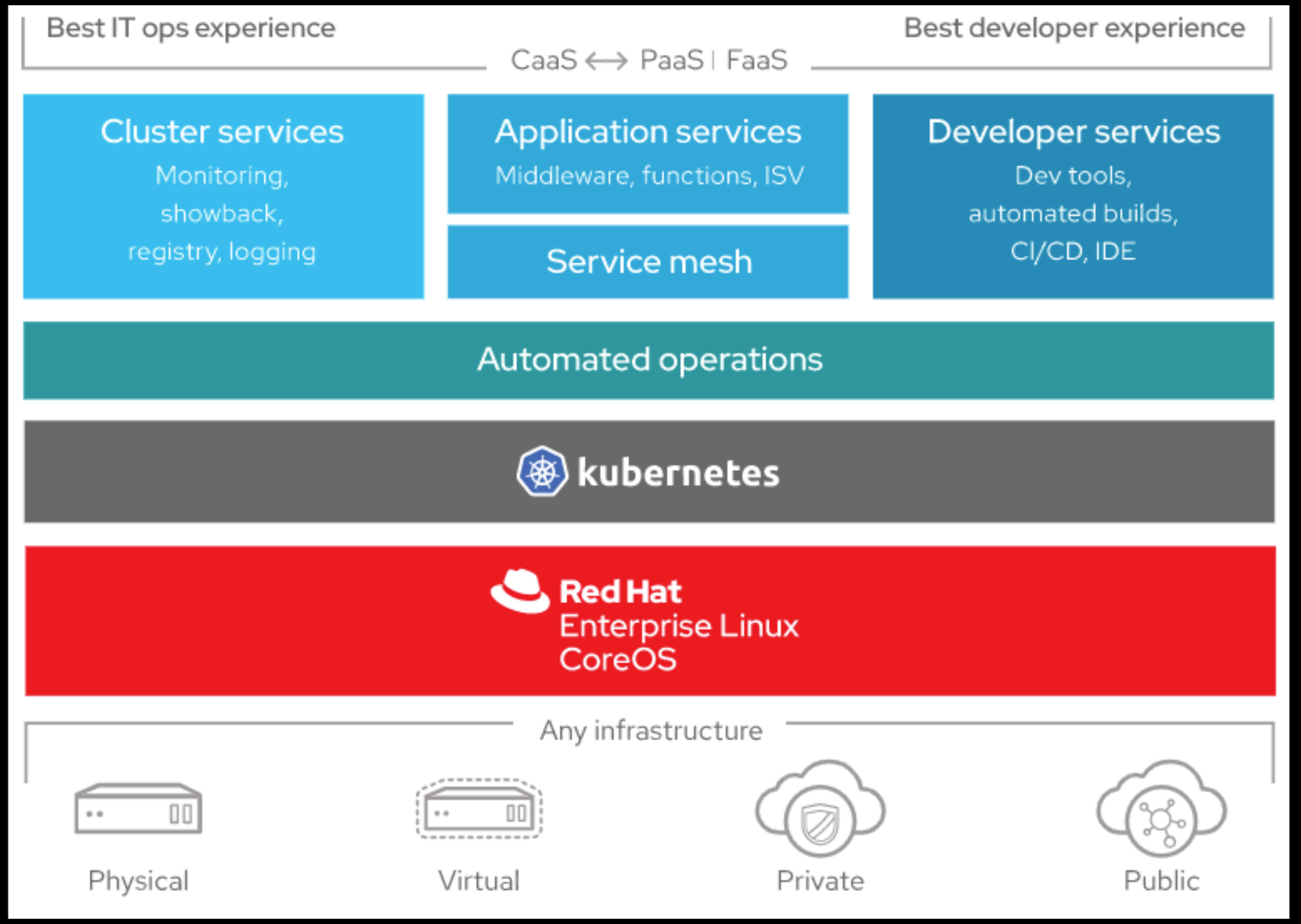
What is Red Hat OpenShift?

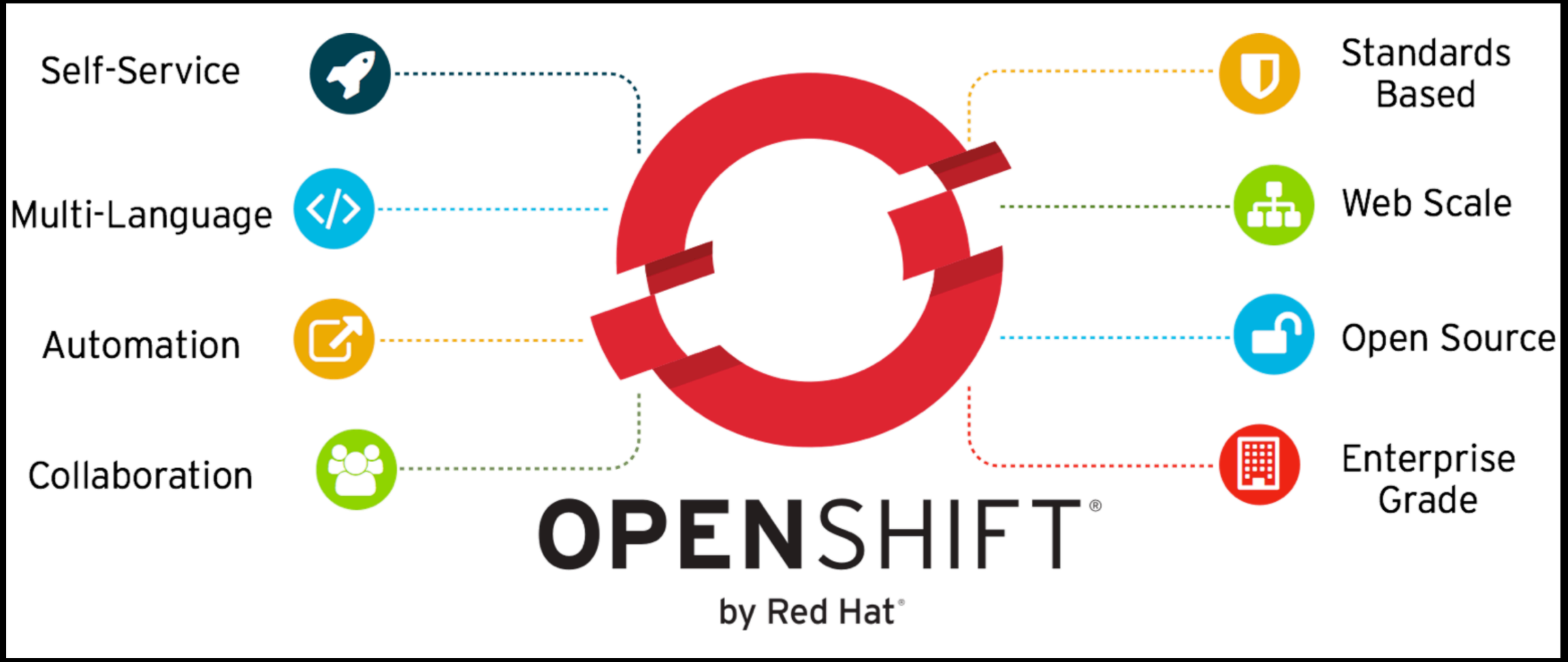
Fundamentals

What is Red Hat OpenShift?

Red Hat OpenShift is a hybrid cloud, enterprise Kubernetes application platform.



Value of OpenShift



For developers...

Code. Shift. Push.

Build with speed, deploy at scale

No two developers work in exactly the same way. Red Hat OpenShift helps application developers build with speed, agility, confidence, and choice. Code in production mode, anywhere you choose to build. Get back to doing work that matters.

Learn Kubernetes in your browser. No downloads required.

Use what you love

Red Hat OpenShift provides commercial support for the languages, databases, and tooling you already use, while providing easy access to services, including public cloud services through the service broker.

Streamlined delivery

Red Hat OpenShift includes everything you need to manage your development lifecycle: standardized workflows, support for multiple environments, continuous integration, release management, and more.

Use new technology

Red Hat OpenShift is extensible to emerging Kubernetes-based frameworks and technologies, making it easier for teams to do more with containers. Benefit from greater productivity with Knative, Istio, and machine-learning / artificial intelligence.

For operations...

Stable. Scalable. Enterprise security.

Consistency and stability, everywhere you operate

Increase efficiency with automated operations, without sacrificing stability. Accelerate developer productivity and deliver application portability on a consistent foundation across the hybrid cloud, commercially supported by the open source enterprise software leader and with all the benefits of enterprise security.

Try Red Hat OpenShift Container Platform, free for 30 days.

Continuous Security

Control, defend, and extend the application platform throughout the application's lifecycle. Red Hat OpenShift monitors security throughout the software supply chain to make applications more stable without reducing developer productivity.

Unified Control

Control clusters, services, and roles for multiple teams from a centralized administrative console, along with metering and management capabilities.

Automated Operations

Red Hat OpenShift includes streamlined installation and automated updates for the container host, the cluster, and applications and services. Install Red Hat OpenShift once and consume Kubernetes as a service, anywhere your organization runs.

For business...

Unite your teams, shift your business

Stay ahead of your competition

Red Hat® OpenShift® is a platform for long-term innovation. Power business transformation and unite your teams on a cost-effective, single platform to quickly deliver the exceptional experiences your customers expect, anywhere they are.

Learn how Red Hat drives innovation and efficiencies with containers and Kubernetes.

Faster deployments, lower costs

Red Hat and Red Hat OpenShift have helped customers like UPS, BBVA cut development times for new applications from months to mere weeks, reduce platform deployments to hours, and cut application build costs by up to 60%.

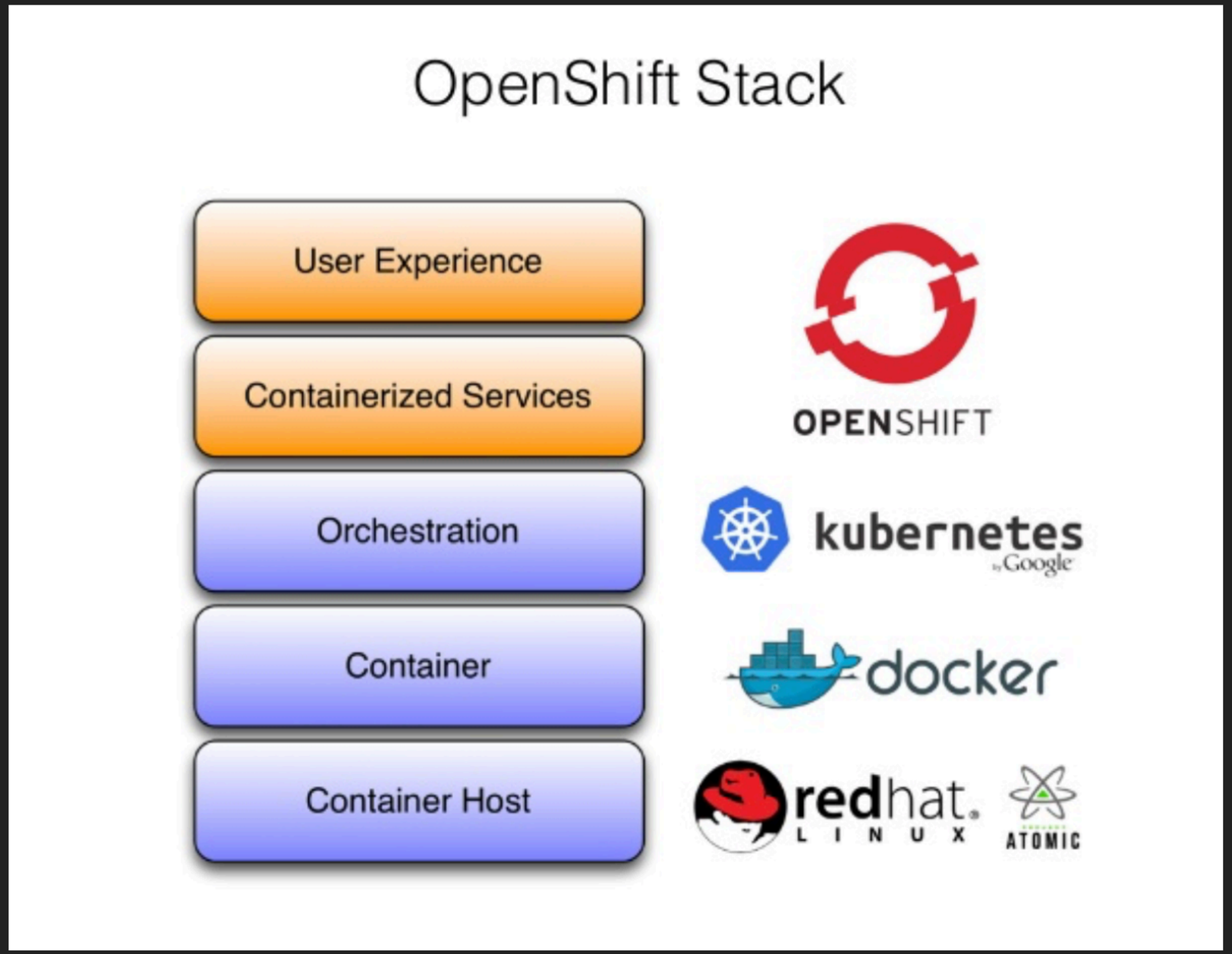
Hybrid cloud flexibility

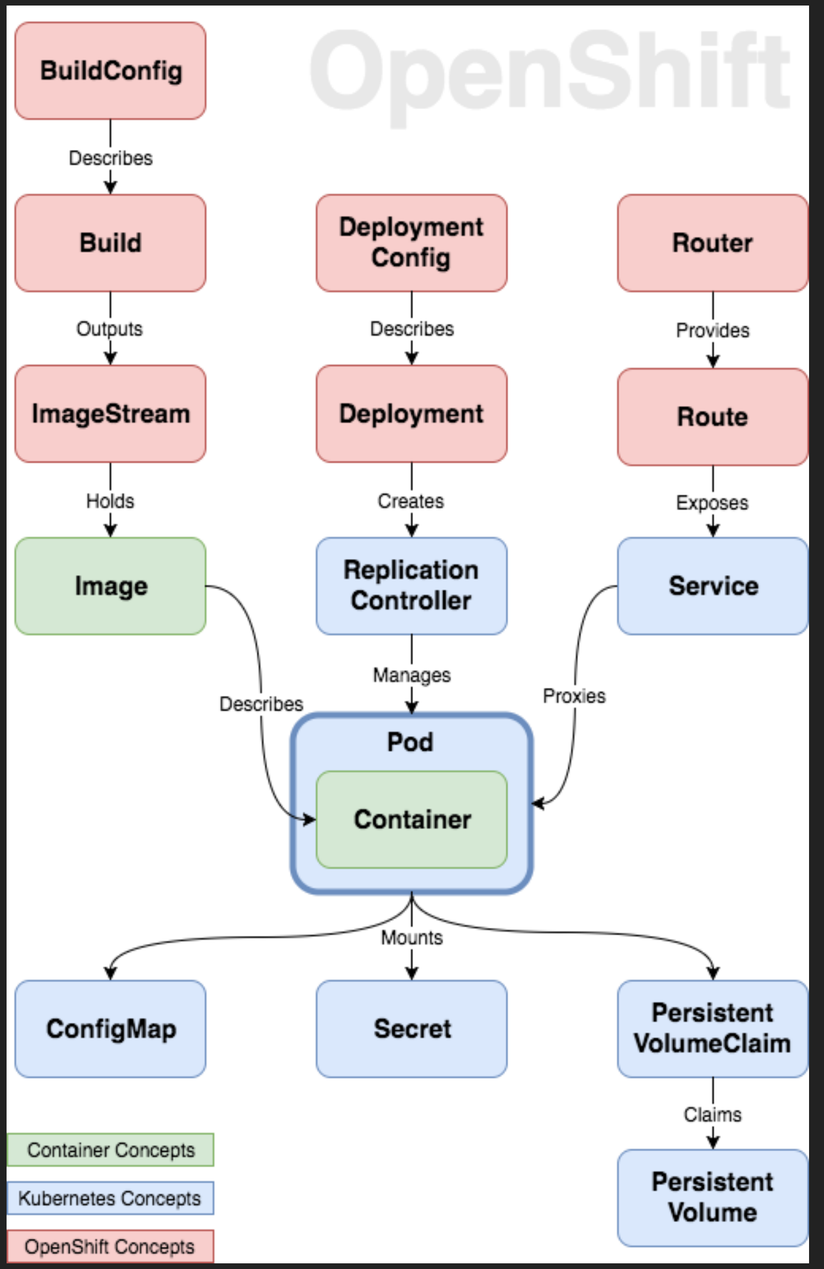
Red Hat OpenShift gives organizations greater flexibility across on-premises and public cloud infrastructure. It provides a consistent Linux and Kubernetes foundation across any infrastructure, including managed solutions like Red Hat OpenShift Dedicated and Red Hat OpenShift on Azure.

Enterprise expertise

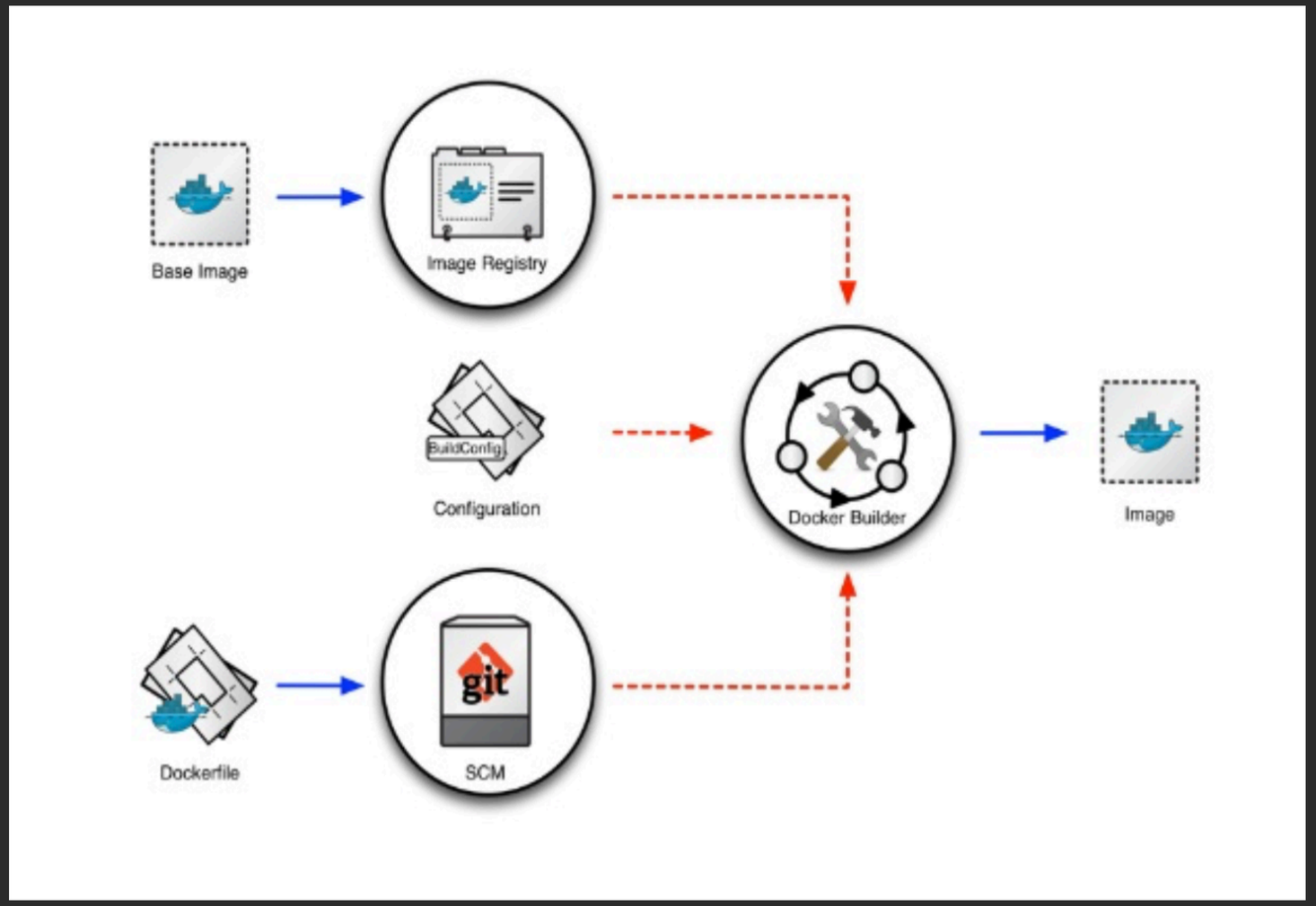
Red Hat has years of experience helping customers operationalize Kubernetes, whether it's on-premises with Red Hat OpenShift Container Platform or as a managed service with Red Hat OpenShift Online or Red Hat OpenShift Dedicated.

- Openshift stack




OpenShift
BuildConfig
Describes
Build
Outputs
ImageStream
Holds
Image
Deployment Config
Describes
Deployment
Creates
Replication Controller
Manages
Router
Provides
Route
Exposes
Service
Describes
Proxies
Pod
Container
Mounts
ConfigMap
Secret
Persistent VolumeClaim
Claims
Persistent Volume
Container Concepts
Kubernetes Concepts
OpenShift Concepts

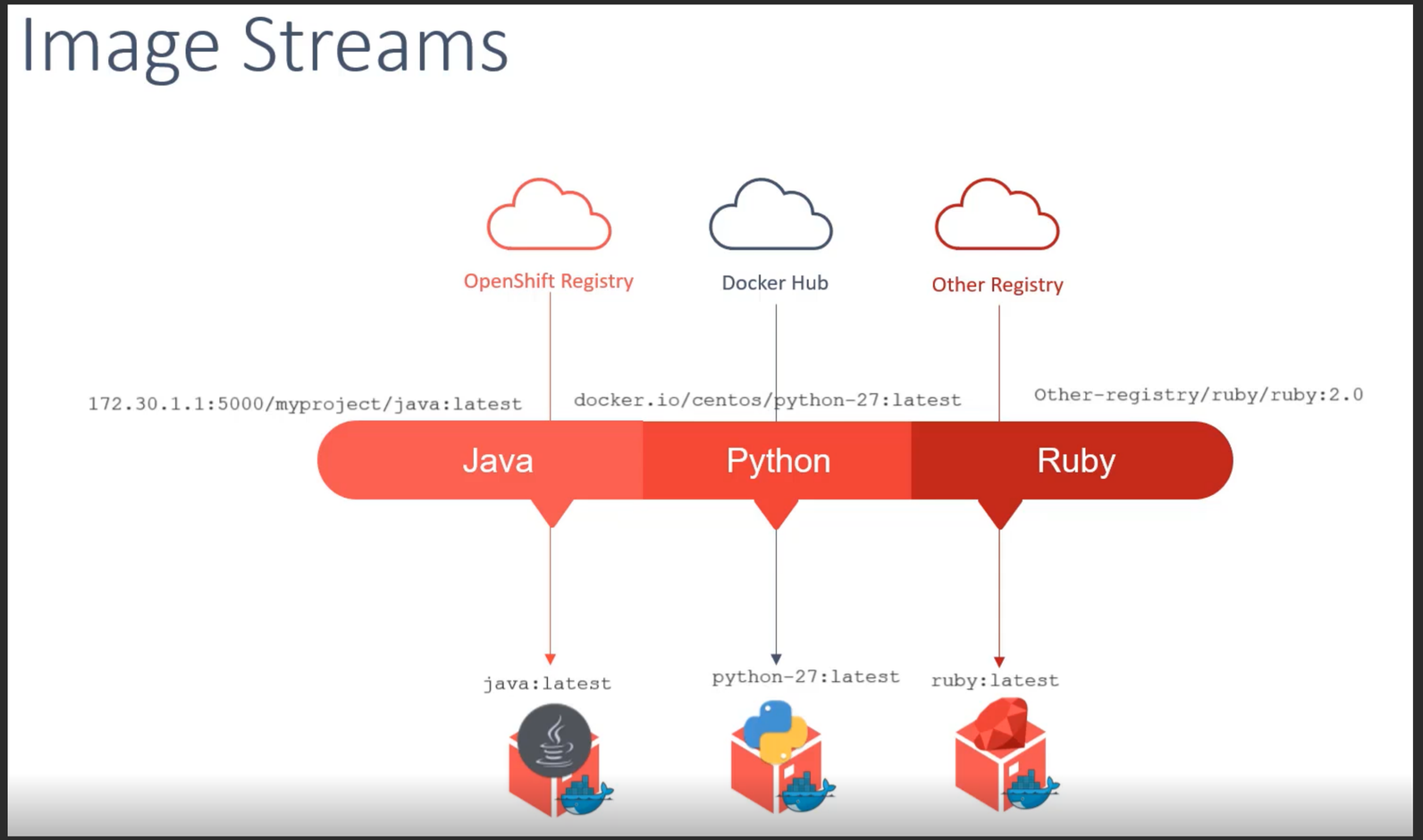
Build Config



- A build is used to transform source code into a runnable container image.
- A build configuration describes the definition and set of triggers that are to be created when a new build is created.

```
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "ruby-sample-build" (1)
spec:
  runPolicy: "Serial" (2)
  triggers: (3)
    -
      type: "GitHub"
      github:
        secret: "secret101"
    - type: "Generic"
      generic:
        secret: "secret101"
    -
      type: "ImageChange"
  source: (4)
    git:
      uri: "https://github.com/openshift/ruby-hello-world"
  strategy: (5)
    sourceStrategy:
      from:
        kind: "ImageStreamTag"
        name: "ruby-20-centos7:latest"
  output: (6)
    to:
      kind: "ImageStreamTag"
      name: "origin-ruby-sample:latest"
  postCommit: (7)
      script: "bundle exec rake test"
```

Image Streams

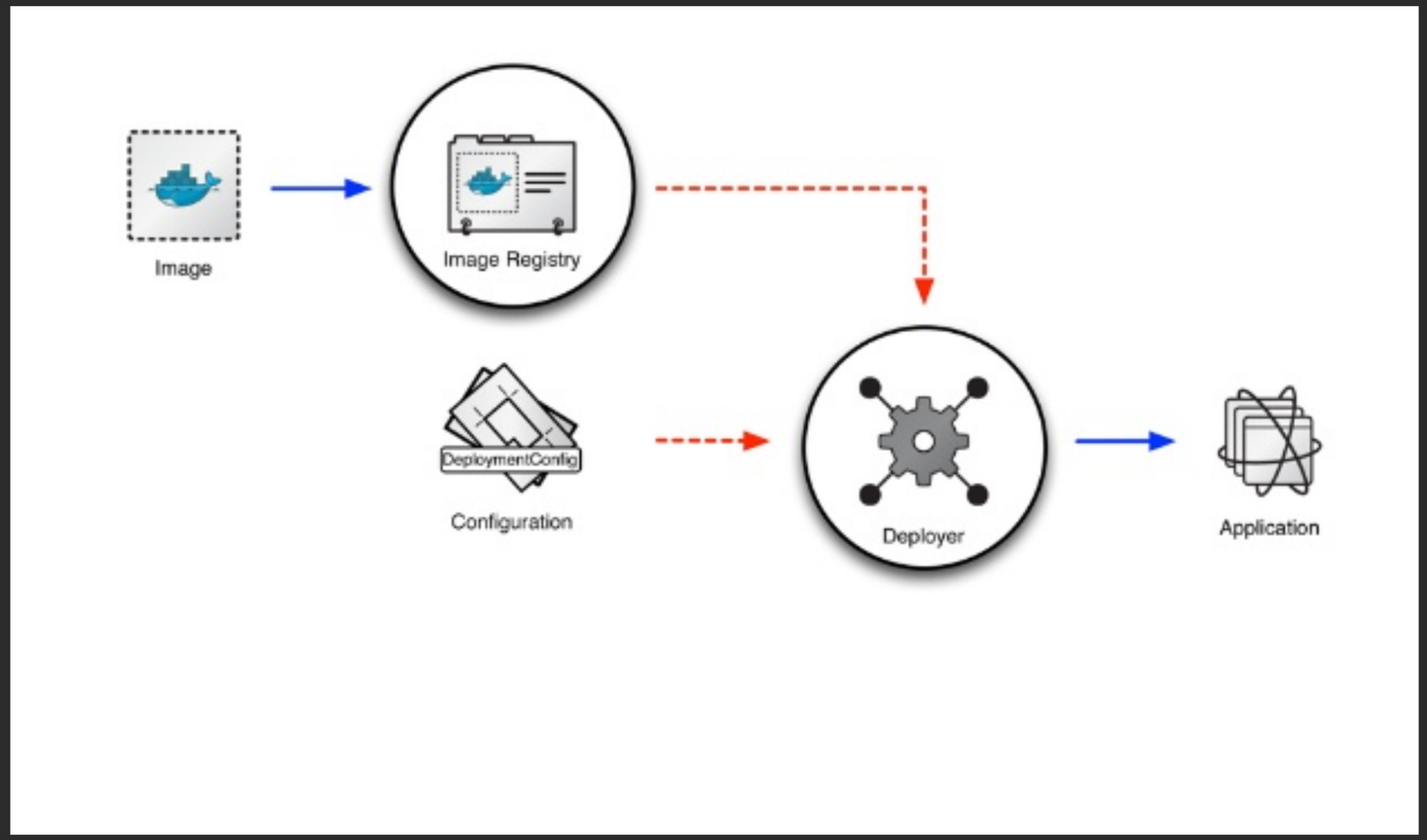


- An image stream comprises one or more Docker images identified by tags. It presents a single virtual view of related images, similar to a Docker image repository, and may contain images from any of the following:

- Its own image repository in OpenShift's integrated Docker Registry


```
{
  "kind": "ImageStream",
  "apiVersion": "v1",
  "metadata": {
    "name": "origin-ruby-sample",
    "namespace": "p1",
    "selfLink": "/osapi/v1/namesapces/p1/imageStreams/origin-ruby-sample",
    "uid": "480dfe73-f340-11e4-97b5-001c422dcd49",
    "resourceVersion": "293",
    "creationTimestamp": "2015-05-05T16:03:34Z",
    "labels": {
      "template": "application-template-stibuild"
    }
  },
  "spec": {},
  "status": {
    "dockerImageRepository": "172.30.30.129:5000/p1/origin-ruby-sample",
    "tags": [
      {
        "tag": "latest",
        "items": [
          {
            "created": "2015-05-05T16:05:47Z",
            "dockerImageReference": "172.30.30.129:5000/p1/origin-ruby-sample@sha256:4d3a646b
58685449179a0c61ad4baa19a8df8ba668e0f0704b9ad16f5e16e642",
            "image": "sha256:4d3a646b58685449179a0c61ad4baa19a8df8ba668e0f0704b9ad16f5e16e642
"
          }
        ]
      }
    ]
  }
}
```

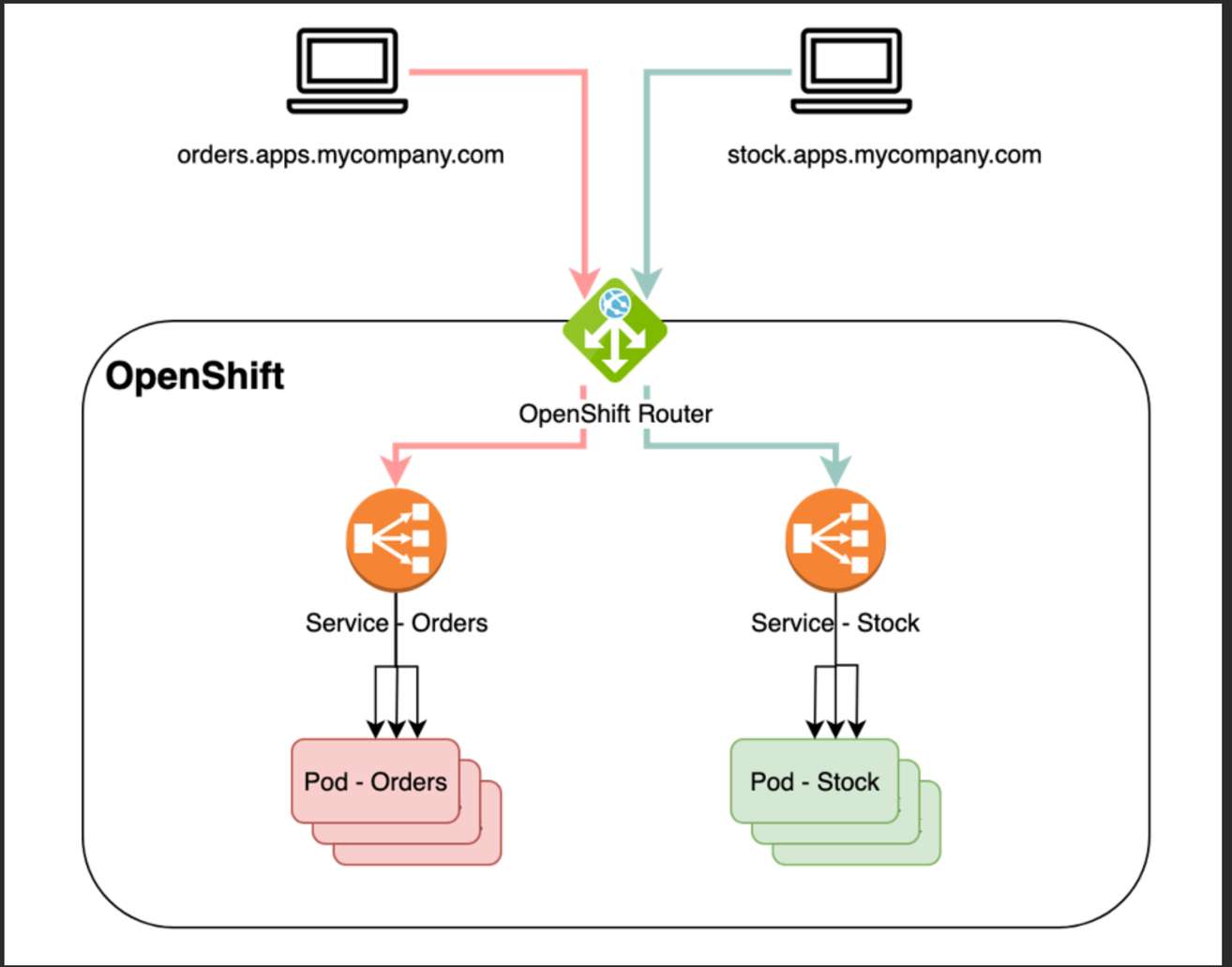
Deployment Config



- A deployment configuration describes the desired state of a particular component of the application as a pod template.

```
kind: "DeploymentConfig"
apiVersion: "v1"
metadata:
  name: "frontend"
spec:
  template: (1)
    metadata:
      labels:
        name: "frontend"
    spec:
      containers:
        - name: "helloworld"
          image: "openshift/origin-ruby-sample"
          ports:
            - containerPort: 8080
              protocol: "TCP"
  replicas: 5 (2)
  triggers:
    - type: "ConfigChange" (3)
    - type: "ImageChange" (4)
      imageChangeParams:
        automatic: true
        containerNames:
          - "helloworld"
        from:
          kind: "ImageStreamTag"
          name: "origin-ruby-sample:latest"
  strategy: (5)
    type: "Rolling"
  paused: false (6)
  revisionHistoryLimit: 2 (7)
  minReadySeconds: 0 (8)
```

Routes



- An OpenShift route is a way to expose a service by giving it an externally-reachable hostname like www.example.com.

```yaml
apiVersion: v1
kind: Route
metadata:
  name: host-route
spec:
  host: www.example.com (1)
  to:
    kind: Service
    name: service-name
```

```yaml
apiVersion: v1
kind: Route
metadata:
  name: no-route-hostname
spec:
  to:
    kind: Service
    name: service-name
```

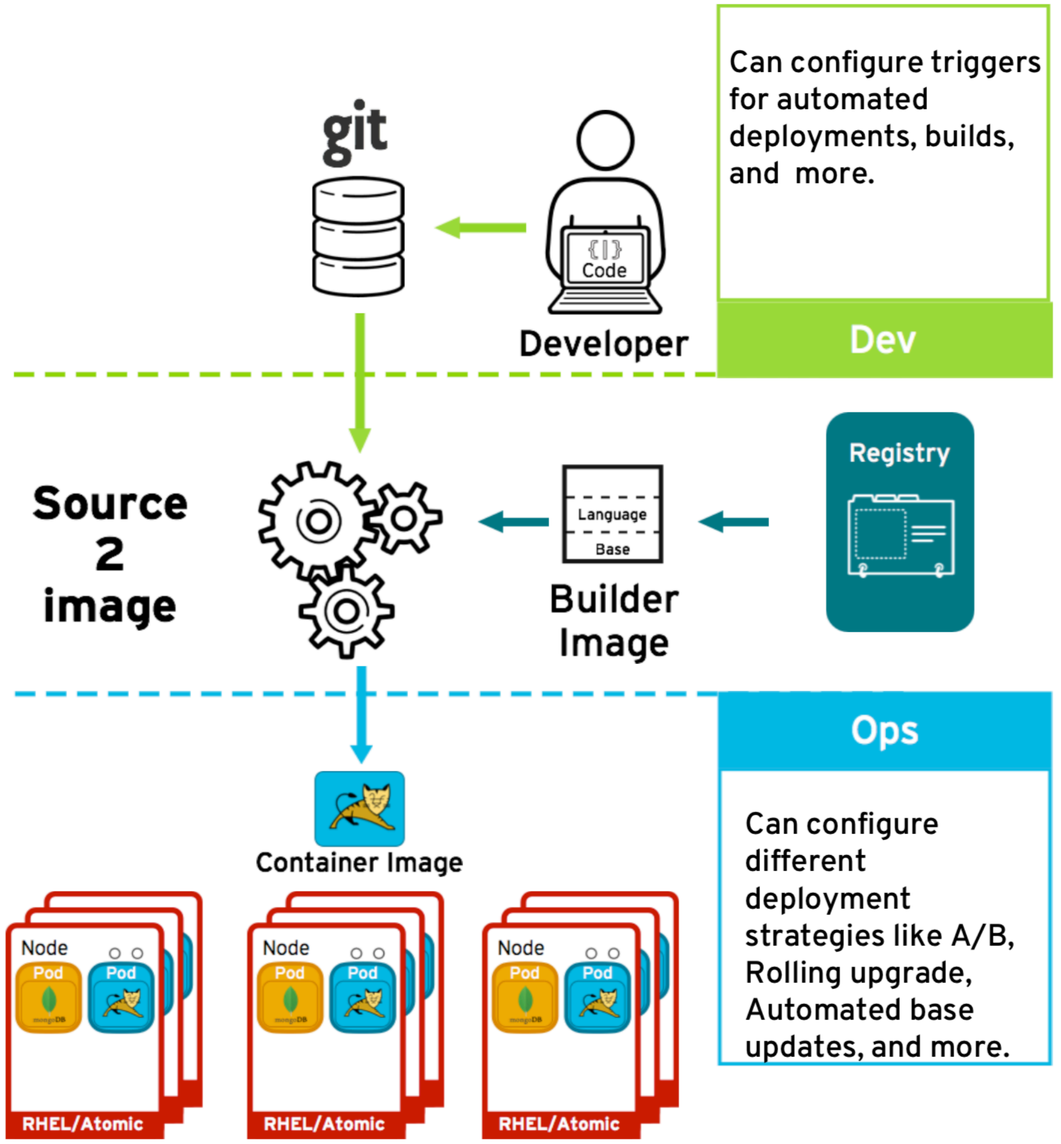
Source 2 Image

Source-to-Image (S2I) is a toolkit and workflow for building reproducible container images from source code. S2I produces ready-to-run images by injecting source code into a container image and letting the container prepare that source code for execution. By creating self-assembling **builder images**, you can version and control your build environments exactly like you use container images to version your runtime environments.

Source 2 Image Walkthrough

Deploy

OpenShift automates the deployment of application containers across multiple Node hosts via the Kubernetes scheduler. Users can automatically trigger deployments on application changes and do rollbacks, configure A/B deployments & other custom deployment types.



KUBERNETES VS OPENSHIFT

Well suited for Kubernetes cluster types

- ✓ Client has a mature Kubernetes platform or an existing preference for Kubernetes
- ✓ Client depends on middleware that is better supported by Kubernetes than OpenShift
- ✓ Client values access to the latest open source features and a broad selection of alternatives
- ✓ Client has unique and specific requirements for their cloud platform
- ✓ Client prefers to leverage public cloud services over cluster hosted services
- ✓ Client does not see the cost/benefit value associated with the additional OpenShift expenses
- ✓ Client expects to run the majority of their workload in the public IBM Cloud
- ✓ Client would prefer to keep their CI/CD orchestration outside of their clusters

Well suited for OpenShift cluster types

- ✓ Client has an existing investment in OpenShift or an existing preference for OpenShift
- ✓ Client depends on middleware that is produced or supported by Red Hat and/or IBM
- ✓ Client values hand-picked, security hardened, pre-integrated open source solutions
- ✓ Client values a user friendly, turn-key cloud platform that “just works”
- ✓ Client anticipates self-hosting most services to maintain consistency in every environment
- ✓ Client wants the highest security measures possible throughout their development lifecycle
- ✓ Client needs to manage hybrid-cloud, multi-cloud and/or multi-tenant cloud environments
- ✓ Client would benefit from a cloud-native CI/CD pipeline designed for speed and consistency