

TECHNOLOGY DETAIL

STREAMLINE CLOUD-NATIVE DEVELOPMENT WITH RED HAT OPENSIFT APPLICATION RUNTIMES

KEY BENEFITS:

- Makes it very easy to get started creating cloud-native applications, while transitioning from monoliths to microservices
- Uses existing expertise to solve cloud development challenges with a hand-picked set of runtimes and frameworks
- Runs on any cloud infrastructure or on-premise

INTRODUCTION

While it is easy to see the benefits of building cloud-native applications and microservices, it can take some time between running some initial code and reaching the point of a production-ready delivery. As you move applications to the cloud and migrate to microservices, complexity increases because of the larger number of connected pieces. A slow connection can lead to cascading failures that are difficult to diagnose due to exhaustion of available resources as the number of outstanding requests continually increases. In addition, infrastructure concerns, such as resiliency and security, can take time away from working on code that provides real business value.

New microservice development can be built from the ground up with resiliency and operational readiness in mind. However, the majority of organizations depend on existing applications that would be too expensive to rewrite using the latest framework and language. Also, it is tempting for organizations to adopt different solutions for modernizing existing applications versus creating new microservices. However, this can be inefficient if the two different approaches have nothing in common.

Developers need:

- Runtimes, frameworks, and infrastructure components ready to be built into a container along with application code.
- Working examples that show how to use the runtimes that are buildable and deployable to the cloud.
- The right tool for the job: multiple frameworks and languages with solutions for both existing applications and new microservices.
- Solutions that can be deployed to multiple clouds as well as on-premise.

Red Hat[®] OpenShift Application Runtimes is designed with these needs in mind to streamline cloud-native application development and delivery.

RED HAT OPENSIFT APPLICATION RUNTIMES

OpenShift Application Runtimes provides modern, cloud-native, multicloud application runtimes. These containerized runtimes provide a prescriptive developer experience to help you quickly start creating new cloud-native applications and cloud-enabling existing applications. Multiple frameworks, along with infrastructure components for resiliency, security, and operations, are



facebook.com/redhatinc
[@redhatnews](https://twitter.com/redhatnews)

linkedin.com/company/red-hat

integrated to help you deliver code that is production-ready. A completely streamlined development and deployment platform is possible because OpenShift Application Runtimes is based on Red Hat OpenShift, a container application platform that is designed for open hybrid clouds.

Red Hat OpenShift Application Runtimes is:

- Prescriptive—a set of runtimes and frameworks are carefully selected as recommended solutions for specific development challenges, such as creating microservices or lifting and shifting existing applications to the cloud.
- Multiframework and language-agnostic—provides the flexibility to use the right tool for the job, from providing reactive services for browsers and mobile clients to performing traditional transaction processing.
- Designed to use existing expertise—since many organizations have considerable Java™ EE expertise, several runtime options are provided that allow existing code to be reused, while covering a range of architectural styles from traditional Java EE to microservices.
- Very easy to start using—the Launch service in OpenShift Application Runtimes lets you start a complete, runnable project online from the Red Hat Developers website based on your choice of runtime and sample use case. All of the necessary code is forked into a project under your GitHub account or can be downloaded as a zip file. If you use Maven, the runtimes can be automatically pulled from the Red Hat JBoss® Enterprise Maven Repository by adding a few lines to your project's pom.xml file.
- Tested, verified, and maintained—the components in OpenShift Application Runtimes are tested and verified by Red Hat so you can be confident in your choices. Red Hat is a significant contributor to, and in many cases is leading, the upstream open source projects that are included in the runtimes. Several of the runtimes are based on Red Hat products, including Red Hat JBoss Enterprise Application Platform (JBoss EAP) and Red Hat JBoss Web Server. Red Hat has a comprehensive roadmap for OpenShift Application Runtimes that includes other Red Hat products like Red Hat JBoss Middleware.
- Cloud infrastructure agnostic and DevOps-ready—offers a streamlined path to applications that are ready to run on your choice of public and private clouds, as well as in your own datacenter. A self-service web console lets you build, deploy, and manage your applications. You can use Red Hat OpenShift Online to get started without installing any software or Red Hat Container Development Kit to run a complete single-node Red Hat OpenShift cluster on your laptop.

Using Red Hat OpenShift, Red Hat OpenShift Application Runtimes streamlines deployment and operations in addition to development. Red Hat OpenShift provides a large set of capabilities to take the complexity out of building and running distributed systems at scale. It provides:

- Service discovery, routing, and load balancing to distribute traffic to multiple instances for scaling and rolling deployments.
- Configuration and secrets management to decouple environment-specific information from the code base.
- Centralized logging, metrics, and monitoring for operational efficiency.

Red Hat OpenShift gives developers a self-service platform on which to build and run containerized applications. Using Red Hat OpenShift, you can spin up an environment for a new microservice in minutes. Powerful, automated build and deployment pipelines are available with a few clicks in Red Hat OpenShift for production-grade continuous integration and continuous delivery (CI/CD) and rolling deployments. The ability to automate deployments is critical as you make the move to microservices and start performing more frequent releases.

A key component of Red Hat OpenShift is Kubernetes, the container orchestration platform originally developed by Google. The industry has chosen Kubernetes as a de facto standard cloud operating environment. Kubernetes provides many of the infrastructure services to enable and operate cloud-native applications at cloud scale. Without Red Hat OpenShift, Kubernetes can be daunting to learn with all the YAML files that need to be created. In Red Hat OpenShift, all of the power of Kubernetes is available through a comprehensive web interface that is easy for both developers and operations to use.

By using OpenShift Application Runtimes, within minutes you can build and run code in a container on top of a fully featured, enterprise-grade cloud platform. Figure 1 shows the logical service layers in Red Hat OpenShift Application Runtimes.

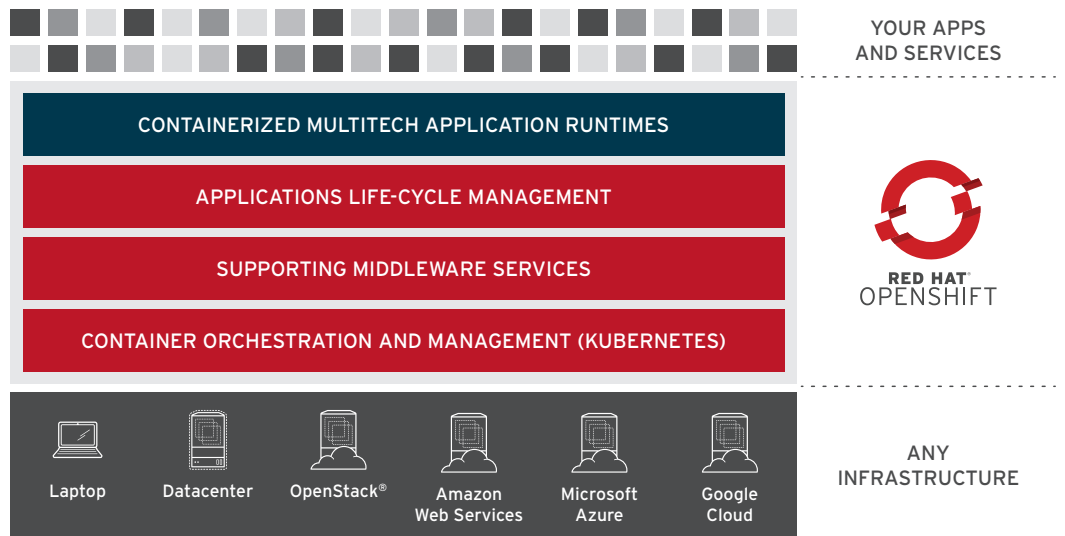


Figure 1. Red Hat OpenShift Application Runtimes service layers

PROVIDED RUNTIMES

Red Hat OpenShift Application Runtimes includes a set of runtimes specifically suited to help developers overcome the challenges of delivering modern, cloud-native applications:

- JBoss EAP is a Java EE 7-certified platform based on a modern, modular, cloud-ready architecture that is ideal for moving existing Java EE and Spring MVC applications to the cloud with minimal changes. The JBoss EAP runtime is tightly integrated to take advantage of the container capabilities in Red Hat OpenShift to simplify dynamic scaling, configuration, and life-cycle management. The capabilities in Red Hat OpenShift let you transition your application to a fast-moving monolith that is easier to deploy and test.
- WildFly Swarm provides a familiar path to microservices for Java EE developers as well as a lighter-weight alternative to a full Java EE runtime. A subset of Java EE is combined with Eclipse MicroProfile, an open, community-driven specification for enterprise Java microservices. The result is a platform that is both standards-based and agile. WildFly Swarm offers an innovative approach to building and running Java EE applications and microservices by packaging them with just enough of the server to create an application that can be run with just the `java -jar` command. This is accomplished by deconstructing the WildFly Application Server into a set of components called fractions. Only the fractions actually needed by the application are assembled into the final JAR file. WildFly Swarm is useful for migrating functionality from existing Java EE monoliths to microservices or developing new microservices while using your existing Java EE expertise.
- Eclipse Vert.x is designed to build reactive microservices on Java Virtual Machine (JVM). The reactive programming model is very popular for building more flexible, loosely coupled, and scalable applications that are responsive and resilient. It is ideal for high-concurrency and low-latency services that are event-driven and use asynchronous, nonblocking I/O. Vert.x lets Java developers use their existing expertise while moving to reactive programming and microservices. Vert.x itself is polyglot, enabling developers to use other languages that run on JVM. However, for the initial release of OpenShift Application Runtimes, only Java is supported.
- Red Hat JBoss Web Server runtime is provided for Spring Boot microservices and Spring framework applications. JBoss Web Server includes Apache Tomcat, Apache HTTPD, Apache CXF, and Hibernate, which are fully supported by Red Hat. This runtime combines Spring, Spring Boot, Spring Cloud Kubernetes, and Netflix OSS (Hystrix and Ribbon). The components are tested and verified to work together by Red Hat. Integration allows native OpenShift/Kubernetes services to be used with the familiar Spring application programming interfaces (APIs) for services like configuration and service discovery. The benefit of using the native services is that they can be managed with Red Hat OpenShift.
- Node.js is a JavaScript runtime. It is an event-driven, server-side framework that is popular for “JavaScript Everywhere,” such as mobile and browser back ends. This runtime gives JavaScript developers access to services that are similar across all of the runtimes. Through Red Hat OpenShift, all of the runtimes share the same management and operational capabilities. For the initial release of OpenShift Application Runtimes, Node.js is considered a technology preview.

Figure 2 shows an overview of the components in Red Hat OpenShift Application Runtimes.

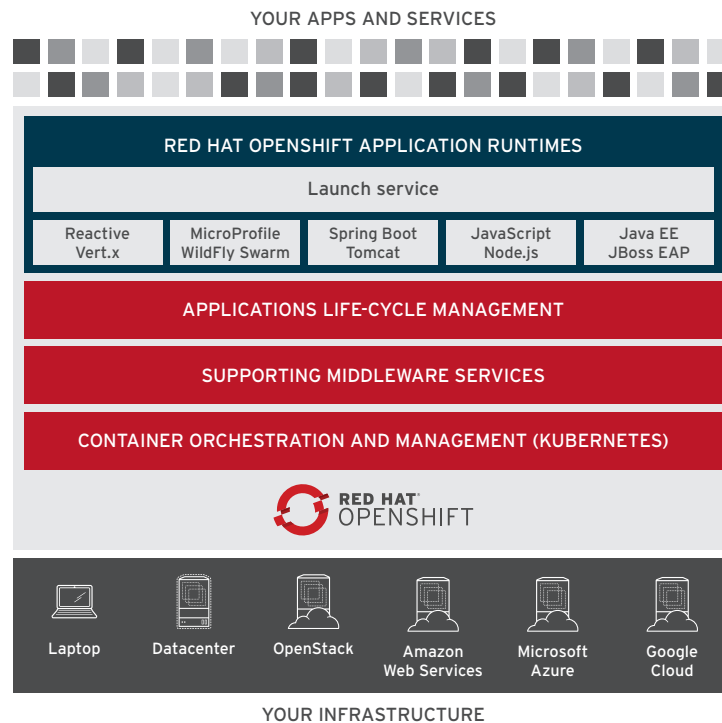


Figure 2. Components of Red Hat OpenShift Application Runtimes

USING RED HAT OPENSIFT APPLICATION RUNTIMES

MODERNIZING EXISTING APPLICATIONS THROUGH CLOUD MIGRATION

It is not always practical or cost-effective to rewrite existing applications using the latest frameworks or architectural methodologies to make them cloud-native. However, these applications still need to be maintained and have a backlog of improvements that are required. Reducing the backlog can be difficult due to lengthy release cycles. Rehosting an existing application in the cloud using OpenShift Application Runtimes makes it much easier to implement CI/CD, along with rolling releases, such as blue/green or canary deployments. This allows you to make more frequent releases with less risk. The cloud-hosted application can become a “fast-moving monolith,” enabling you to address the backlog of improvements.

Rehosting an application in the cloud can also be an effective first step toward implementing microservices. The flexibility of the cloud makes it easier to deploy new microservices in containers alongside the existing application. You can then implement microservices that migrate functionality out of the monolithic application.

TRANSITIONING FROM APPLICATION SERVERS TO CLOUD RUNTIMES

The role of the traditional application server changes when moving to a cloud-native platform. Traditionally, application servers provided a runtime and deployment environment, as well as management services for a central domain running on a cluster of systems. Application servers can be a viable runtime environment for cloud-native applications. However, many of the included operational tools, such as the administration console, are no longer necessary and can interfere with the automated management capabilities provided by OpenShift and Kubernetes.

For migrating Java EE applications, OpenShift Application Runtimes includes a cloud-native runtime based on JBoss EAP. Using the modern, modular architecture of JBoss EAP, this runtime makes it easy to take advantage of OpenShift features such as dynamic scaling and CI/CD. For applications still running on an older, traditional application server, Red Hat Application Migration Toolkit can analyze your project and highlight any areas that might need to change in order to migrate to JBoss EAP.

It is not uncommon for organizations to have applications deployed on application servers today that only use a small subset of Java EE. OpenShift Application Runtimes offers lighter-weight options for those applications. The WildFly Swarm runtime can assemble an application, along with only the components of the application server actually used by the application, into a single executable JAR file. For web applications, or applications that only need a minimal container, the Red Hat JBoss Web Server runtime provides a fully supported cloud-native solution based upon Apache Tomcat and the Apache HTTPD server.

BUILDING MICROSERVICES

OpenShift Application Runtimes includes multiple runtime choices for building microservices.

- For developers with Java EE expertise, the WildFly Swarm runtime is attractive as it blends a subset of familiar Java EE standards with the Eclipse Java Microprofile standard, which is more agile and suited to the evolving world of microservices.
- To build reactive microservices, the Eclipse Vert.x runtime is provided. Additionally, Vert.x is designed to support polyglot development as it allows the use of other languages that run on the JVM.
- Spring Boot microservices can be built using the JBoss Web Server runtime, which includes a supported Apache Tomcat container along with other components, including Spring Cloud Kubernetes. Spring Cloud Kubernetes provides integration between Spring APIs and the management capabilities of Red Hat OpenShift.

Microservices architecture is a very compelling approach for solving many software engineering challenges, but it comes at the cost of increased complexity. Microservices require additional infrastructure services such as service discovery and client-side load balancing. The number of independently deployed services makes monitoring and metrics, including distributed tracing across services, a key operational requirement.

OpenShift Application Runtimes is designed to address these needs. The runtimes include a number of infrastructure components, such as Netflix Hystrix and Ribbon, to robustly handle interprocess communications between microservices while avoiding cascading failures. The OpenShift Application Runtimes Launch service provides runnable code that shows how to use these components to implement patterns like circuit breakers, which prevent problems with slow or unresponsive services from cascading into multiple failures.

AVOIDING UNNECESSARY COMPLEXITY

OpenShift Application Runtimes includes software infrastructure components that provide a number of benefits to developers. The complexity of handling infrastructure concerns is pushed down from the application into the runtime, where it becomes more of a commodity. Developers can use the components and example code provided by Red Hat instead of building their own or searching for other components to use. The maintenance burden and risks of including third-party components and then trying to stay on top of reported bugs and security vulnerabilities are avoided.

THE RIGHT TOOL FOR THE JOB

In the past it might have been practical to use a single framework, language, or architectural style for all development projects. This is no longer the case. When Red Hat recently surveyed customers, nearly half felt it was a priority to choose the right tool for the job. Red Hat OpenShift Application Runtimes includes multiple runtimes covering multiple frameworks and languages.

It is now practical to use multiple frameworks and languages because these runtimes have a lot in common. Applications built with these runtimes can all be configured, deployed, and managed the same way through Red Hat OpenShift, making things easier for both developers and operations. All of the runtimes can be secured through Red Hat Single Sign-On. Additionally, the Launch tool is a valuable learning tool, showing each example implemented with each runtime.

ON-PREMISE AND MULTICLOUD SUPPORT

Most IT organizations prefer not to be limited to a single provider of public cloud infrastructure. Additionally, many organizations believe that even a number of years from now, at least half of their applications will still be running on premise. With Red Hat OpenShift, you have complete deployment flexibility: choose public clouds such as Red Hat OpenShift Online (available on the major cloud infrastructure providers), private clouds such Red Hat OpenShift Dedicated, or your own infrastructure through Red Hat OpenShift Container Platform.

TRY IT NOW

Red Hat OpenShift Application Runtimes includes the Launch service, a getting-started developer experience you can try now. Go to developers.redhat.com/launch, where you can create and deploy a sample runnable project to OpenShift Online. You can also choose to build and deploy locally, downloading the project as a zip file.

Using the Launch service, you first pick a mission—a preconfigured, functioning application that demonstrates a fundamental aspect of modern application development. The missions you can choose from are:

- CRUD—create, read, update, and delete operations to a relational database over an HTTP REST endpoint.
- Circuit breaker—the circuit breaker pattern prevents cascading failures by restricting access to services that are not responding correctly.
- Externalized configuration—an example of using ConfigMap to separate environmental configuration from your application code.
- Health check—the health check pattern checks whether an application is available and is ready and able to service requests. This is one of the steps required when enabling applications and services scale-out via multiple instances.
- REST API Level 0—a basic example of a REST endpoint for remote procedure calls over HTTP.

Once you select a mission, you can choose one of the available runtimes—Eclipse Vert.x, Node.js, Spring Boot, or Wildfly Swarm—to use for your project. Since the same missions are available for the different runtimes, you have the opportunity to explore and compare them. Figure 3 shows the flow for the Launch service.

If you choose to deploy to OpenShift Online, the code is forked to a project in your GitHub account. Your Red Hat OpenShift project is configured to build and deploy on each push to the master branch of your GitHub repository. The benefit of this process is that it handles all of the configuration, building, and deployment steps needed to get your project's code running in Red Hat OpenShift. This lets you experience how Red Hat OpenShift can accelerate your development and delivery processes.

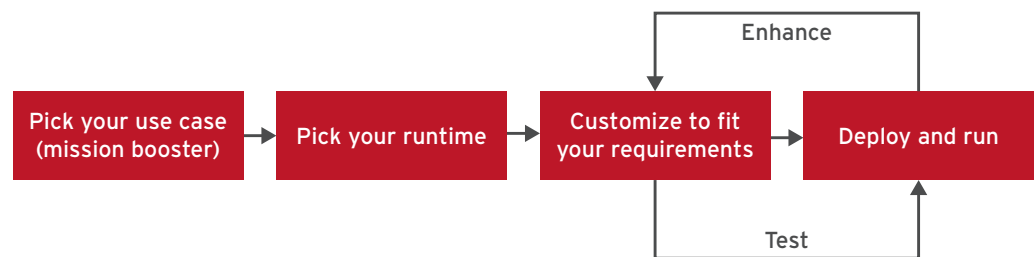


Figure 3. Red Hat OpenShift Application Runtimes Launch service

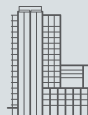
NEXT STEPS

Red Hat has many resources to help you build modern, cloud-native applications:

- Follow the [OpenShift Blog](#) to learn more about Red Hat OpenShift Application Runtimes and Red Hat OpenShift. Watch the OpenShift Commons Briefing videos on Red Hat OpenShift Application Runtimes.
- Download [Red Hat Development Suite](#) from [developers.redhat.com](#) to install a complete development environment on a laptop or desktop. Red Hat Development Suite includes:
 - [Red Hat JBoss Developer Studio](#)—A comprehensive integrated development environment (IDE) based on Eclipse with a broad set of tooling, including OpenShift and container development.
 - [Red Hat Container Development Kit](#)—Run a single-node OpenShift cluster on a local machine even if it is running Windows or macOS.
- Learn how to get a complete cloud-native development environment for you and your team without installing any software through [Red Hat OpenShift.io](#), a developer Software-as-a-Service platform running on OpenShift Online. Note: OpenShift.io is currently in a limited developer preview.
- Get analysis tools to begin migrating your existing applications from the [Red Hat Application Migration Toolkit](#). The toolkit examines application artifacts, including project source directories and application archives, to produce a detailed HTML report that highlights areas that need changes to enable migration. The results are particularly helpful for migrating applications from a legacy Java application server to [JBoss EAP](#), which is one of the runtimes included in OpenShift Application Runtimes.
- Jumpstart modern application development through [Red Hat Open Innovations Labs](#), a focused learning engagement for teams.
- Find out what [Red Hat JBoss Middleware](#) has to offer, including messaging, in-memory data grids, and business process and rules management running on Red Hat OpenShift.
- Read [a free e-book](#) written by Red Hat covering relevant topics like microservices and modern Java EE design patterns.

ABOUT RED HAT

Red Hat is the world's leading provider of open source software solutions, using a community-powered approach to provide reliable and high-performing cloud, Linux, middleware, storage, and virtualization technologies. Red Hat also offers award-winning support, training, and consulting services. As a connective hub in a global network of enterprises, partners, and open source communities, Red Hat helps create relevant, innovative technologies that liberate resources for growth and prepare customers for the future of IT.



[facebook.com/redhatinc](#)
[@redhatnews](#)
[linkedin.com/company/red-hat](#)

[redhat.com](#)
F10490_V1_0118_KVM

NORTH AMERICA
1 888 REDHAT1

EUROPE, MIDDLE EAST,
AND AFRICA
00800 7334 2835
europa@redhat.com

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
info-latam@redhat.com

Copyright © 2018 Red Hat, Inc. Red Hat, Red Hat Enterprise Linux, the Shadowman logo, and JBoss are trademarks of Red Hat, Inc., registered in the U.S. and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks / service marks or trademarks / service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community.