Moving to cloud–native development

Modernize existing applications to build a path to cloud–native

Benefits of running existing applications in containers:

Portability—decouple applications from infrastructure, run applications on any platform that supports containers.

Scalability—flexibility to scale up (or down) as needed to respond to demand, while achieving better resource utilization.

Flexibility. It is easier to deploy containers to create testing environments when needed, without tying up resources when not needed.

Language- and technology-agnostic. Containers support a choice of languages, databases, frameworks, and tooling. They encapsulate all dependencies while segregating them inside the container—whether it is code that is decades old or something written today. The management is the same, regardless of what is in the container. As a result, containers can run legacy technologies along with more modern technologies.

f 𝕏 in

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

Introduction

Software development is critical for business. Applications differentiate businesses, providing the gateway to new services, revenue streams, and market expansion. To deliver faster, developers need to boost productivity. What limits development speed is often more than old code—it is outdated infrastructure, development practices, and procedures. Releases happen infrequently, require significant manual effort, and testing may not be adequate.

Cloud-native development can help organizations deliver innovative applications faster. However, cloud-native development is more than technology—it also involves people and processes.

Cloud-native development practices

A cloud-native application is built to take advantage of cloud computing models to increase speed, flexibility, and quality, while reducing deployment risks. Despite its name, a cloud-native approach is not focused on where applications are deployed, but instead on how applications are built, deployed, and managed.

Cloud-native development is an application development and deployment approach that combines a modular architectural approach like microservices, a modern container-based and application programming interface (API)-driven infrastructure, and DevOps practices to allow continuous application delivery. While cloud-native applications and microservices are often discussed together, many organizations achieve the benefits of cloud-native approaches by focusing on building better modular monolithic applications using the same principles:

- A loosely coupled service-based architecture.

- Lightweight, technology-agnostic API communication between components.

- A container-based infrastructure that can quickly deploy applications and services, scaling them to meet capacity demands, while providing portability across environments.

- Agile development methods with continuous integration/continuous delivery (CI/CD) and DevOps practices that focus on building and delivering applications collaboratively.

Positioning for change: Move existing applications to containers

Existing applications deployed on outdated infrastructure present challenges that can complicate a move to cloud-native development. In addition, existing infrastructure often has fixed, limited resources that inhibit growth. Current, supported versions of middleware and other software might not run on older infrastructure.

Redeploying existing applications using containers can remove the barriers to change and provide the flexibility to evolve. Moving existing applications into containers provides many benefits:

- Containers are technology agnostic. The age and type of technology used inside of the container becomes much less relevant. Developers are free to use the technology that is the right tool for the right task. Both old and new can be deployed on the same platform in separate containers.

Kubernetes has become the defacto standard container orchestration tool. It is an open source solution that automates deployment and management of containers based on Google's experience running massive numbers of containers at scale. Kubernetes uses automated, self-healing mechanisms such as automatically restarting containers, rescheduling containers on different hosts, or even replicating containers for use cases like auto-scaling. Kubernetes works with Linux® containers natively.

For a number of years, Red Hat has been the number two top contributor to the Kubernetes project behind only Google.[1] Since 2015, the Kubernetes project has been managed by the Cloud Native Computing Foundation. The openness of Kubernetes has led to widespread industry adoption and fueled rapid innovation, spawning additional open source projects that build on top of Kubernetes.

- The latest middleware, application services, and new technology such as artificial intelligence/ machine learning (AI/ML) can be deployed on the container platform, creating opportunities for integration as well as modernization.

- Application portability increases once existing applications are decoupled from their infrastructure. Moving applications or adding resources is simplified.

- Modern development methodologies tend to rely on automated testing to improve quality and reliability. It is much easier to create test environments to support the development of new software when all of the components are running in containers. If the container platform supports self service, developers can create and tear down their own test environments as needed without waiting for someone to provision them.

The Strangler Pattern is an approach that has been used successfully to modernize existing monolithic applications. This method identifies small pieces of functionality and replaces them with microservices, application services, or serverless functions that are cloud-native. Deploying new software components alongside the existing application to modernize becomes straightforward when everything is running on a container platform.

Red Hat OpenShift: A foundation for building cloud-native applications

Kubernetes has become a defacto standard for running containerized applications. Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform that offers the simplicity and automation of the public cloud. However, Kubernetes was built primarily for the needs of IT operations teams, not developers. As a container platform for building and running containers, Red Hat OpenShift is designed for use by both developers and operators. Red Hat OpenShift extends Kubernetes to align with how developers work and aligns with the development life cycle, including automatically building containers from source code to deploy newly built applications into containers.

Developers use Red Hat OpenShift as a self-service platform for building and running containers, creating and tearing down development and test environments as needed. Operators use Red Hat OpenShift to monitor and manage production environments. Recognizing that developers and operators view the same system differently, Red Hat OpenShift includes both a developer and an administrator perspective in the graphical user interface (GUI). A developer-oriented command line interface (CLI) is provided in addition to the administration CLI. DevOps practices are fostered by giving both roles access to the same underlying system and metrics.

With Red Hat OpenShift, developers can deploy applications on Kubernetes—without needing to learn Kubernetes. For example, Kubernetes uses a number of YAML files to store deployment and configuration details, which makes the learning curve fairly steep. Red Hat OpenShift provides a GUI and deployment templates to eliminate the need to edit YAML files by hand. Deployment templates can be stored with the application's source code in version control systems to streamline the deployment of applications and support Infrastructure-as-Code. Using Red Hat OpenShift, sophisticated CI/CD pipelines can easily be created to automate the deployment life cycle.

---

1 *Stackalytics. "Kubernetes," Accessed August 2020.*

While Red Hat OpenShift provides functionality similar to public cloud services, it offers the flexibility to develop and run anywhere, both on-premise and in the public cloud. Developers can use Red Hat OpenShift running in the public cloud as a service, on their organization's own infrastructure, or even on their own laptop. For those who prefer their own self-contained development environment, Red Hat CodeReady Containers can be downloaded to get a minimal preconfigured Red Hat OpenShift environment running locally.

A platform for existing and cloud-native applications across hybrid cloud

A challenge when moving existing applications to containers is that the applications are not cloud-native and need persistent storage for data, logs, and sometimes configuration. However, containers were not designed for stateful applications. To accommodate existing applications' need for permanent storage, Red Hat OpenShift offers flexible persistent storage volumes that are automatically attached to the applications' containers when they run. Red Hat OpenShift Container Storage can be used to make software-defined persistent storage available with block, file, or object access methods.

Applications that need to stay on-premise due to regulatory, data locality, security, or performance concerns can still take advantage of Red Hat OpenShift as an application platform. Organizations can deploy Red Hat OpenShift in their own datacenters. For applications that cannot easily be containerized, Red Hat OpenShift also introduces a method to manage virtual machines and containerized applications in a single interface.

Using Red Hat OpenShift in the public cloud abstracts away the details of the underlying cloud or container platform, benefitting developers and operators as organizations transition to hybrid and multicloud deployments. Developers only need to learn one environment for building and deploying container-based applications.

Red Hat OpenShift provides a common, consistent application platform for both old and new applications. The application portability and deployment flexibility allow organizations to run applications wherever they fit best.

Tools for cloud-native development

Red Hat offers many tools to help developers build applications to run in containers. These tools integrate with and are based on the functionality provided in Red Hat OpenShift.

For a cloud-native development environment, Red Hat CodeReady Workspaces is a Kubernetes-native developer workspace server that runs on top of Red Hat OpenShift. It provides a browser-based integrated development environment (IDE) so that developers do not need to install software or copy code to their local machine.

Red Hat CodeReady Studio is a traditional desktop IDE with a broad set of tooling for containers and multiple programming models. Developers can use their choice of tools. Red Hat offers extensions for Microsoft Visual Studio Code, including OpenShift Connector, and language extensions for Java™, YAML, and XML.

Building containers to run applications requires languages, runtimes, frameworks, and application servers to allow the application to run. While there are a number of sources for base images, the challenge is to acquire them from a known source. Red Hat is a trusted source for base images that are more secure, up-to-date, and free of known vulnerabilities.

Red Hat Runtimes is a collection of Red Hat OpenShift integrated runtimes to simplify cloud-native development. The runtimes include support for Quarkus, Spring Boot, Vert.x, Node.js, and Thorntail to let developers choose the right tool for the right task. Quarkus is a Kubernetes-native Java stack for building fast, lightweight microservices and serverless applications. Red Hat Container Catalog provides an additional library of tested container images that developers can use as base images.

To streamline application delivery, Red Hat Middleware is packaged in containers and ready to run on Red Hat OpenShift. The Red Hat Middleware portfolio includes streaming and broker-based messaging, integration services, single sign-on, API management, process automation, and decision management.

Build and deploy automation with Red Hat OpenShift

With a microservices-based application, tens or even hundreds of services can be released and deployed independently. To deploy all of this code more frequently with greater confidence, automated building, testing, and deployment is critical. Using Red Hat OpenShift, developers can set up automated builds for CI/CD. The builds can be triggered automatically whenever new code is checked into the source code version control system. When the build completes successfully, it can be automatically deployed in place of the previous version. Developers can use familiar tools, like Jenkins, without the complexity of trying to create a build environment from scratch. Powerful build automation pipelines can be created using the Red Hat OpenShift console. The build tasks run as Kubernetes jobs in containers managed by Red Hat OpenShift.

Red Hat OpenShift provides routing and load-balancing functionality for application services running on the platform. This capability allows the use of advanced deployment patterns that can reduce the risk of software releases by allowing new releases to be tested on a controlled subset of users. Patterns like rolling, blue/green, and canary deployments are available.

Moving to cloud-native development

To start cloud-native development:

- Move existing applications into containers running on Red Hat OpenShift—either in the cloud or on-premise. Update the application with modern runtimes and middleware deployed in containers.

- Implement CI/CD using build pipelines and rolling deployments to speed delivery.

- Identify portions of existing applications that can be broken out into new microservices or replaced with application services or serverless functions. Speed new development by using the right tool for the right task, including different frameworks and architectural approaches as well as cloud services.

- Implement DevOps practices and use metrics to learn and improve. Use the networking capabilities of Red Hat OpenShift, like service mesh, to manage, monitor, and trace communications between application components.

Learn more

Red Hat's approach to cloud-native application development helps you accelerate application development and deployment by modernizing existing applications and integrating existing and new applications and data. Read more about Red Hat's cloud-native development solutions.

About Red Hat

Red Hat is the world's leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers integrate new and existing IT applications, develop cloud-native applications, standardize on our industry-leading operating system, and automate, secure, and manage complex environments. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500. As a strategic partner to cloud providers, system integrators, application vendors, customers, and open source communities, Red Hat can help organizations prepare for the digital future.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

North America	Europe, Middle East, and Africa	Asia Pacific	Latin America
1 888 REDHAT1	00800 7334 2835	+65 6490 4200	+54 11 4329 7300
www.redhat.com	europe@redhat.com	apac@redhat.com	info-latam@redhat.com

redhat.com
#F24468_0820