Moving legacy applications to containers

Introduction

Many organizations have successful initial public cloud projects, but they are primarily new greenfield application projects carefully chosen as good candidates for running in the public cloud. As a result of these successes, IT organizations are attracted to the elasticity, scalability, and speed of deployment that cloud computing offers. By using cloud technology, IT organizations can more quickly respond to developer and line-of-business demands.

Legacy applications are not typically considered for public cloud deployments because of security, regulatory, data locality, or performance concerns. Many legacy apps were written before cloud computing, so it might seem simpler to leave them deployed on existing infrastructure. However, that decision can create bottlenecks for organizations trying to modernize. Efforts to become more responsive while reducing costs cannot succeed without addressing legacy applications, because keeping these applications running often accounts for the majority of IT costs.

Containers are a key technology that makes many of the services offered by public cloud providers possible. The design of containers opens up many possibilities for automation. Containers, combined with a platform that provides cloud-like automation, are an attractive environment for running applications. Migrating legacy applications to containers can remove many of the barriers to modernization.

Reasons for moving legacy applications to containers

Scaling and the need to respond quickly

Legacy applications are often deployed on infrastructure that has fixed and limited resources. Often the resource utilization is low. Yet, when demand increases, it is challenging to scale up without long lead times and high costs. User and business expectations for responsiveness and costs have been changed by the success of Software-as-a-Service (SaaS) applications running in the public cloud. It can be a difficult conversation to explain why internal applications cannot evolve as quickly.

While many legacy applications have had stable and predictable growth in the past, new user-driven demand means that the resources available to the legacy application might need to be scaled up quickly. The user-driven demand is difficult for an IT organization to predict because:

- It is now common for mobile and connected applications to require application programming interface (API)-level access to existing applications.

- The rise of data science and machine learning creates additional demand for data access.

- Some of the demand, as well as the applications that consume data and APIs, can be external to the IT organization.

Because it is difficult to predict growth and control demand, existing applications need to be repositioned to allow the organization to respond quickly. Modern cloud-scale applications address this challenge by running in containers on a platform that increases or decreases the number of containers running, and thus the capacity of the application, in response to demand.

redhat.com

Benefits of running legacy applications in containers

- Portability: Ability to decouple applications from infrastructure and run applications on any platform that supports containers

- Scalability: Ability to scale up (or down) as needed to respond to demand and achieve better resource usage

- Flexibility: Ease in deploying containers to create testing environments when needed, without tying up resources when they are not needed

- Language and technology versatility: Support for a choice of languages, databases, frameworks, and tooling to allow legacy technologies to coexist with more modern technologies, whether the code is decades old or newly written

Positioning legacy systems to enable change

Legacy systems and new greenfield development opportunities are often connected. New applications and services typically need data from legacy apps, or might perform a service by executing a transaction in the legacy system. A common approach to modernization is to put new interfaces and services implemented in newer technologies in front of legacy systems.

Connecting new development on public clouds to internally run legacy applications creates additional complexity and security challenges. Problems, especially network related, are more difficult to trace and diagnose. This issue is more challenging if the legacy application is running on older infrastructure where modern tools are not available.

New applications that depend on legacy systems need to be tested. Modern development methodologies tend to rely on automated testing to improve quality and reliability, so legacy applications likely will need more resources in testing environments. Also, development teams might require access to additional, possibly isolated, legacy application test environments to develop and test their new code.

Deploying legacy applications in containers can remove the barriers to change and provide the flexibility to evolve. The process starts by decoupling applications from old infrastructure and then using the same platform to host legacy applications and new greenfield development. Both can coexist on the same container or cloud platform and can be managed with the same tools. Operational efficiencies can increase once automation and modern management tools are used with legacy applications without the constraints of old infrastructure.

Considerations for moving legacy apps to containers

Persistent storage

Applications that are not cloud-native need persistent storage for data, logs, and sometimes configuration. However, containers are designed to exist for short periods of time. Unless other arrangements are made, anything written inside the container is lost when the container is restarted. Legacy applications can be accommodated by arranging for the container to have access to persistent storage. Because containers are typically run on clusters consisting of multiple machines, the storage for persistent data needs to be available on all of the machines in the cluster that the container could run on. The types of storage available largely depend on the container platform and the infrastructure it runs on.

Container orchestration

Most applications consist of containers that need to run at the same time and connect to each other. For example, the components that make up the tiers of a three-tiered application would run in separate containers. The web or app containers benefit from the ability to dynamically scale out to more machines in the cluster as demand increases. The process of scheduling and managing the containers is referred to as container orchestration, a key responsibility of a container platform.

Networking

Applications often have specific networking requirements that are key to the manner in which they are deployed. Virtual networks might need to be recreated in the container environment. In some cases, physical networking hardware might need to be virtualized in the container environment. As with storage, the virtual network for the app needs to be available on each host the container runs on. The container platform manages the virtual network environment that connects components of an

About Kubernetes

Kubernetes has become the de facto standard container platform. It is an open source platform 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, and replicating containers for use cases like auto-scaling to make sure the desired end state for the application is maintained. Kubernetes works with Linux containers natively, including the popular Docker container format.

Red Hat is the No. 2 contributor to the Kubernetes project, trailing 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.

application running in different containers, and it isolates those components from the other applications running on the container platform.

Building containers for applications

Developers need tools for building the application and any of the necessary dependencies into container images. This process should be repeated for code changes and finished releases. During rollouts, operators or developers also need the ability to deploy the new images in place of the current running container images. While low-level tools exist for performing these tasks, the container platform makes this process much easier.

Building containers to run applications often requires languages, runtimes, frameworks, and application servers to allow the application to run. These can be pulled in during the build process with a base container image as a foundation. While there are a number of sources for base images, the challenge is to acquire them from a known and trusted source. The base images need to be secure, up-to-date, and free of known vulnerabilities. If a vulnerability is discovered, the base images must be updated. Users also need a way to find out if containers are based on out-of-date images.

Public cloud challenges

One of the challenges IT organizations face when adopting the public cloud is that the infrastructure, management, and automation software provided by the public cloud are different from what the IT organization uses in its own datacenters. Many public cloud tools and services are not available to run on-premise, so they cannot be used with applications that run internally.

Many organizations choose to use more than one public cloud for reasons like geographic availability, diversity, and cost. However, each public cloud provider offers vendor-specific interfaces, tools, and services.

Containers and cloud have tremendous potential for improving operational efficiency through automation. Containers are an ideal environment to implement DevOps practices and culture. However, a cloud strategy that uses different platforms everywhere there are hosted applications could overload operators and developers with too much to keep track of and learn.

Red Hat's approach: A cloud experience everywhere

Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform with full-stack automated operations to manage hybrid cloud and multicloud deployments, offering the simplicity and automation of the public cloud. It includes an enterprise-grade Linux® operating system, container runtime, networking, monitoring, registry, and authentication and authorization solutions.

You can deploy Red Hat OpenShift Container Platform on your choice of infrastructure, whether in your on-premise datacenter or in a private cloud. If you prefer not to manage the infrastructure, most public cloud providers offer Red Hat OpenShift as a managed service.

Streamlining operations with a consistent hybrid cloud foundation

Red Hat OpenShift helps address the challenges that arise when legacy applications need to stay on-premise as newer development occurs on cloud platforms. It creates a common application platform by abstracting away the details of the underlying cloud or container platform, easing the transition into hybrid and multicloud deployments.

---

1 *Stackalytics. "Kubernetes," Accessed 6 Dec. 2019. https://www.stackalytics.com/cncf?module=kubernetes.*

Red Hat OpenShift's common operational interface for old and new applications, whether they run internally or externally, streamlines operations. The same tools, consoles, and procedures are used regardless of where the application runs. Operators can be productive faster with a reduced learning curve. They no longer have to remember how things work in different environments, so they can diagnose and resolve problems more quickly.

The common application platform increases application portability and deployment flexibility. Containers do not include all of the deployment details necessary to orchestrate multiple containers into delivering a complete application. Kubernetes uses a number of YAML files to store deployment and configuration details. One of the areas where Red Hat OpenShift adds value over Kubernetes is by providing a graphical user interface (GUI) and deployment templates to eliminate the need for operators and developers to edit YAML files by hand.

Deployment templates streamline the process of deploying applications on Red Hat OpenShift and of moving an application from one OpenShift cluster to another. The templates can be part of the application code or kept separately. Applications can be added to the Red Hat OpenShift service catalog, which allows for point-and-click deployment of applications and software components.

For managing multiple Red Hat OpenShift clusters, Red Hat OpenShift 4 introduced a unified hybrid cloud console. This feature provides centralized management and visualization tools across clusters that can run on-premise or on multiple clouds.

Developing applications in containers

Before applications can migrate to containers, the application code needs to be built into a container image. Red Hat OpenShift gives developers a self-service platform where they can build and run containers without waiting for resources to be provisioned. This is one of the key areas where Red Hat OpenShift adds value over Kubernetes.

Through Red Hat OpenShift, developers can set up automated builds for continuous integration and continuous delivery (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. This feature helps with automated testing and continuous improvement. Red Hat OpenShift has rich functionality for creating sophisticated automated build pipelines. Developers can use familiar tools, like Jenkins, without the complexity of trying to create a build environment from scratch.

IT operations maintains control, and developers can work without administrative access to the cluster. Red Hat OpenShift supports multiple tenants with security. All of the tasks that developers perform, whether running a build or logging in to debug running code, run inside of containers on top of Red Hat OpenShift. Because the development tasks run in containers, they are isolated from other containers and the cluster itself.

Tools for developers

Red Hat offers many tools to help developers build applications to run in containers:

- Red Hat CodeReady Studio is a traditional desktop integrated development environment (IDE) with a broad set of tooling for containers and multiple programming models.

- Red Hat Container Catalog provides a library of tested container images from a trusted source that developers can use as base images.

- Red Hat OpenShift Application Runtimes is a collection of Red Hat OpenShift integrated runtimes covering multiple languages and programming styles to simplify cloud-native development.

- Red Hat Application Migration Toolkit is an assembly of tools that helps developers evaluate code from legacy applications to determine what changes are necessary to run on modern platforms such as current application servers and middleware.

Moving legacy applications into containers

Once the application's containers are built, the next steps for deploying the application are configuring storage and networking. To accommodate the need for permanent storage, applications defined in Red Hat OpenShift can be configured to use persistent storage volumes that are automatically attached to the applications' containers when they run. Developers can manage elastic storage for container-based applications, drawing from storage pools provisioned by operations. Red Hat OpenShift Container Storage can be used to make software-defined persistent storage. It offers block, file, or object access methods to applications running on a Red Hat OpenShift cluster.

Virtual private networking, routing, and load balancing for applications running in containers are built in as part of the platform provided by Kubernetes and Red Hat OpenShift. Networking is specified in a declarative manner as part of the application's deployment configuration. Application-specific network configuration can be stored with the source code to become infrastructure as code. Tying application-specific infrastructure configuration to each application improves reliability when moving, adding, or changing application deployments.

Software-defined routing and load balancing play a key role in enabling applications to automatically scale up or down. Additionally, applications running on Red Hat OpenShift can take advantage of rolling deployments to reduce risk. With Red Hat OpenShift's built-in service routing, strategies for rolling deployments can be used to test new code on subsets of the user population. If something goes wrong, rolling back to a previous version is easier with containers on Red Hat OpenShift.

Finally, Red Hat OpenShift Service Mesh provides increased resilience and performance for distributed applications. OpenShift Service Mesh abstracts the logic of interservice communication into a dedicated infrastructure layer, so communication is more efficient and distributed applications are more resilient. OpenShift Service Mesh incorporates Istio service mesh, Jaeger (for tracing), and Kiali (for visibility) on a security-focused, enterprise platform.

Improving your application landscape

Once your legacy applications are running in containers on Red Hat OpenShift, opportunities for improvement develop. New code releases can occur more frequently with better reliability using CI/CD, build and deployment automation, automated testing, and rolling deployments. The ability to release code more often means your organization can better respond to changing business demands.

A common approach to modernization is to put new interfaces and services implemented in newer technologies in front of legacy systems. This approach is much easier when everything is running in containers, where it does not matter what languages or technologies are running inside each container. The virtual networking capabilities and service mesh in Red Hat OpenShift make it easier to reliably connect application components.

Red Hat OpenShift also makes it easier to deploy the latest middleware alongside your legacy applications. Red Hat offers integration and messaging systems, business process management, and decision management software ready to run on OpenShift clusters in containers. You can use these to connect your applications for agile integration.

Conclusion

Red Hat's approach to hybrid cloud and multicloud provides a common application platform that serves old and new applications, whether they are running on-premise or in the public cloud. The resulting application portability gives organizations the flexibility to run workloads where it makes the most sense. The details of the multiple underlying cloud and container platforms abstract away, making operators and developers more productive regardless of where the application runs.

There are many benefits to containerizing legacy apps and running old and new applications on Red Hat OpenShift. A container-based architecture, orchestrated with Kubernetes and OpenShift, improves application reliability and scalability while decreasing developer and operations overhead. Red Hat OpenShift's full-stack automation, developer self-service, and CI/CD capabilities also provide a foundation for continuous improvement processes.

Learn more about containers and running containers at scale at https://www.redhat.com/en/solutions/hybrid-cloud-infrastructure#scale.

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.