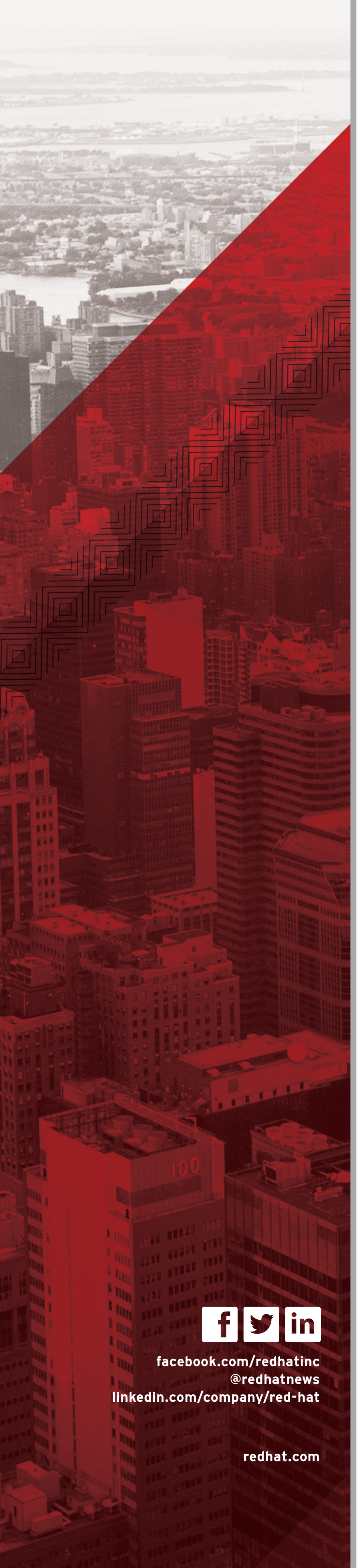


facebook.com/redhatinc
@redhatnews
linkedin.com/company/red-hat

redhat.com

TECHNOLOGY DETAIL

RED HAT CONTAINER IMAGE AND HOST GUIDE

Selecting the right container images and hosts

TABLE OF CONTENTS

AUGUST 2017

100 East Davie Street
Raleigh NC 27601 USA
Phone: +1 919 754 4950
Phone: 888 733 4281
Fax: +1 919 800-3804

The information contained herein is subject to change without notice. Red Hat, Inc. shall not be liable for technical or editorial errors or omissions contained herein.

INTRODUCTION

Linux® containers make it easy to deploy and run your applications across different environments—from a developer's laptop to production clusters. They decouple the upgrade and downgrade of applications from that of the servers on which they run, allowing systems administrators and developers to move at different speeds. Containers also provide transactional upgrades with all of the application dependencies, so if something does not work, it takes seconds to roll back (so no more eight-hour change windows starting at 2 a.m.)

Containers provide all of these capabilities by allowing the operating system to be managed as two separate parts—the container host and the container image. The underlying container host provides the kernel and related tools, which enable the hardware (or virtual hardware), schedule resources, and provide security. The container images, which run on top of a container host, include the operating system libraries, language runtimes, and the application itself. Managing the operating system as two pieces makes it easy for operations teams to focus on managing the container hosts, but there is still shared responsibility for what is inside the container image.

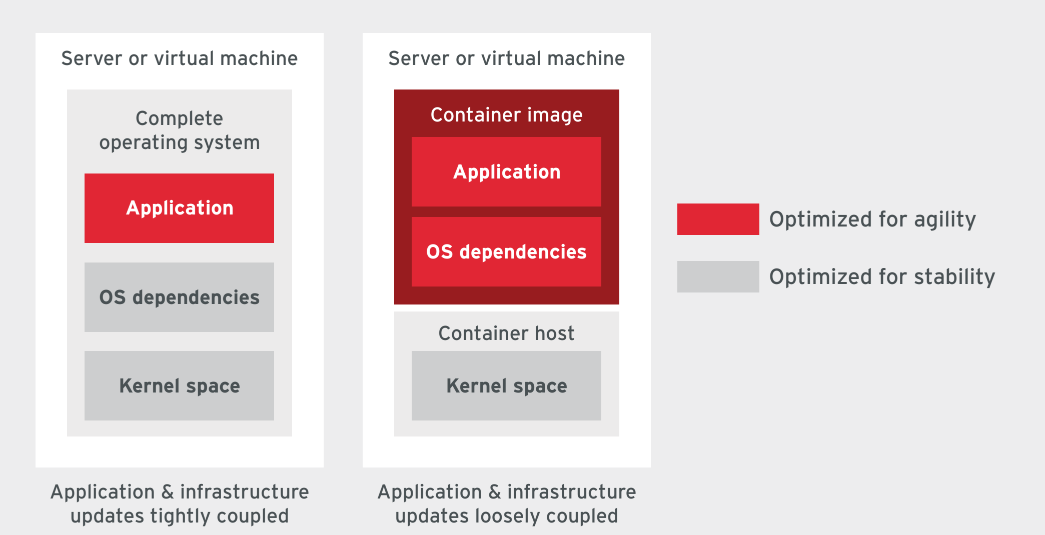

Server or virtual machine
Complete operating system
Application
OS dependencies
Kernel space
Application & infrastructure updates tightly coupled
Server or virtual machine
Container image
Application
OS dependencies
Container host
Kernel space
Application & infrastructure updates loosely coupled
Optimized for agility
Optimized for stability

Figure 1. Infrastructure and application coupling before and after containers

Containers provide application portability. This gives teams better control over the time and place where they deploy code, but portability and compatibility are not the same thing. Development and operations teams can work at different speeds, upgrade and downgrade servers and applications independently of each other, and easily move applications between different environments, but the importance of performance, security, and compatibility between the container hosts and container images does not change. This paper will provide technical guidance on how to select the right container images and hosts for your applications.

CONTAINER IMAGES

TECHNOLOGY

Applications need language runtimes, interpreters, and libraries, so it is important to understand that the final container image that will be deployed in production is typically made up of multiple layers. The first image is called a base image, and it includes everything necessary to get started. Teams often work by building their own layers on these base images. Users can also build on more advanced application images, which not only have an operating system but also include language runtimes, debugging tools, and libraries.

Base images are built from the same utilities and libraries included in an operating system. A good base image provides a secure and stable platform on which to build applications. Red Hat provides base images for Red Hat® Enterprise Linux. These images can be used like a normal operating system. Users can customize them as necessary for their applications, installing packages and enabling services to start up just like a normal Red Hat Enterprise Linux Server.

Red Hat also provides more advanced application images that already include extra software. Examples include language runtimes, databases, middleware, storage, and independent software vendor (ISV) software. Software included in these images is tested, preconfigured and certified to work out of the box. These prebuilt images make it easy to deploy multitier applications from existing components without the work of building your own application images.

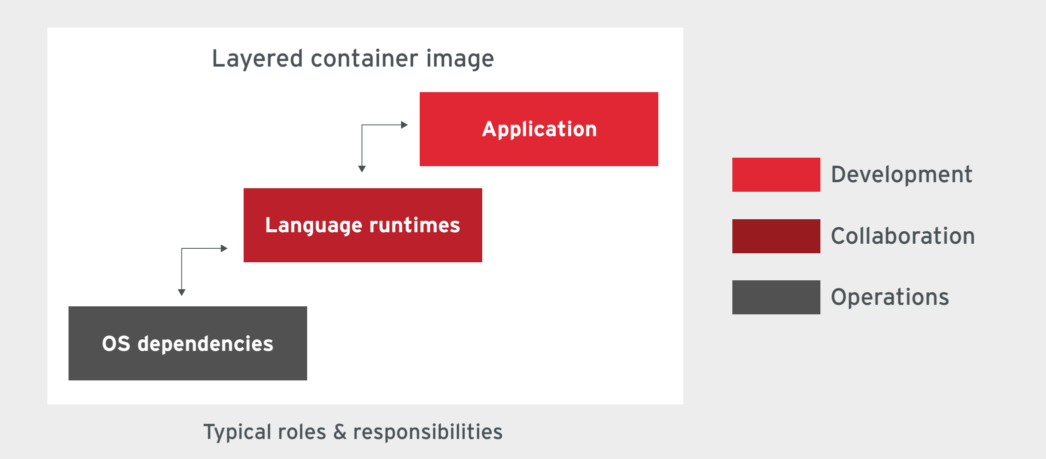


Figure 2. Typical roles and responsibilities for layers within a container image

Linux containers are often described as portable, but portability does not guarantee compatibility. Portability is often believed to mean that you can take a Red Hat Enterprise Linux container image and run it on any container host built from any Linux distribution, but this is not technically accurate. Linux container images are collections of files, including libraries and binaries, which are specific to the hardware architecture and operating system. When the container image is run, the binaries inside the image run just like they would on a normal Linux operating system. There must be compatibility between the container image and container host.

For example, you cannot run 64-bit binaries on a 32-bit host. Nor can you run ARM containers on x86_64 hosts. The same operating system rules apply. Containers are not like virtualization—they do not provide any kind of translation or guarantees that a binary can run on different platforms.

It is sometimes possible to run a binary built for one x86_64 Linux distribution on another without problems. It is also sometimes possible to launch and run container images between a wide variety of Linux distributions. Users can experience inconsistencies when mixing and matching distributions of container images and hosts because of the binaries in Linux container images.

These inconsistencies can range from sporadic failure, which is hard to debug, to complete failure where the container will not run at all. Worse, incompatibilities can lead to performance degradation or security issues. Portability of Linux containers is not absolute—using different Linux distributions, or widely differing versions of the same Linux distribution, can lead to serious problems.

It is best practice to run a consistent version and distribution of Linux in all containers and on all container hosts in a particular cluster. This guarantees compatibility both horizontally and vertically.

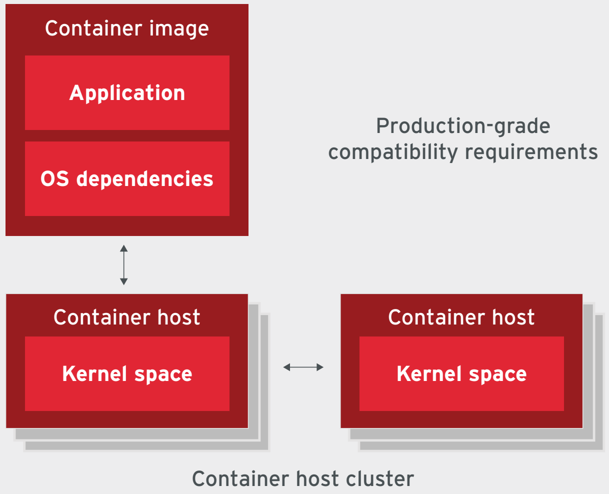


Figure 3. Container host cluster compatibility requirements

Applications and their dependent libraries within the container image often have expectations about the container engine and kernel features that can be exposed when running them on a different Linux distribution. Examples of problems that can occur when mixing Linux distributions, versions, or configurations within the cluster include:

- Running binaries in the container that are compiled with the expectation of Security-Enhanced Linux (SELinux) support on hosts that do not support SELinux.[1]
- Running container images that expect specific kernel capabilities (CAP_PTRACE, etc.) on a container host, with a container engine, configured to deny the behavior. [2]

[1] "Docker, SELinux and the myth of kernel independence | Fewbytes." 21 Nov. 2014, http://www.fewbytes.com/docker-selinux-and-the-myth-of-kernel-indipendence/. Accessed 22 Mar. 2017.

[2] "CentOS as a Docker host causes different container behavior ...." 7 Jul. 2016, http://www.centos.org/forums/viewtopic.php?t=58409. Accessed 22 Mar. 2017.

- Running binaries that require specific input/output control (ioctl) calls, or specific layouts of /proc and /sys, which are incompatible with, and determined by, the version of the underlying container host kernel. All of these interfaces can change through versions of the kernel, user space tools, or libraries.

- Incompatibilities in the underlying filesystem backing the container engine. This can lead to problems that are very difficult to debug. For example, if the underlying filesystem does not support POSIX or extended attributes, arbitrary failures can occur in containerized applications.[3] Never mix and match Overlay2, device mapper, or filesystems within a cluster.

- Container host and container image version mismatches. The larger the version mismatch between a user space and the kernel, the more likely there will be incompatibilities.

  - Newer binaries or libraries, older kernel: Running binaries or libraries that depend on newer kernel features not available on the container host could cause problems. This can happen when running a newer kernel in development and an older kernel in production. With a mismatch like this, glibc will check for a minimum kernel version. If the minimum version is not satisfied, the program will exit. [4]

  - Older binaries or libraries, newer kernel: Running binaries or libraries on a container host with a newer kernel than used during testing and development could result in behavioral differences. This is particularly true when glibc enables new runtime-detected features. Always be careful to ship what you test.

- Differences in hardware, kernel, and libraries. For example, glibc is optimized for specific versions of hardware—some accelerated routines are selected based on hardware availability. This is a combination of kernel detection and glibc detection. Inconsistencies between nodes can lead to behavioral changes in your application; for example, routines may change from hardware accelerated to software only or vice versa. This could unexpectedly slow your application down or speed it up, depending on which node it is scheduled. If your application is made up of multiple containers serving a single service or application programming interface (API), all requests may not perform the same.

- Running containers that manage the environment. This is common in distributed system administration tasks, which may need to interact with Kubernetes masters, Red Hat OpenShift Container Platform masters, the docker daemon, or the Linux kernel directly. Managing the environment is a common use case with super-privileged containers and can require complete compatibility between all APIs—from the orchestration layer down to the kernel. Furthermore, these tasks can be critical to restoring service to a cluster; as a result, they carry a bigger risk than any individual application.

Complete portability of containers is critical when a container host or containerized process fails. Failed containers need to be restarted quickly on an alternate host. The risk of having a container fail to restart because of compatibility issues is increased when mixing and matching container hosts and container images based on different versions and different distributions of Linux. The worst time to find out about incompatibility problems is during a production outage.

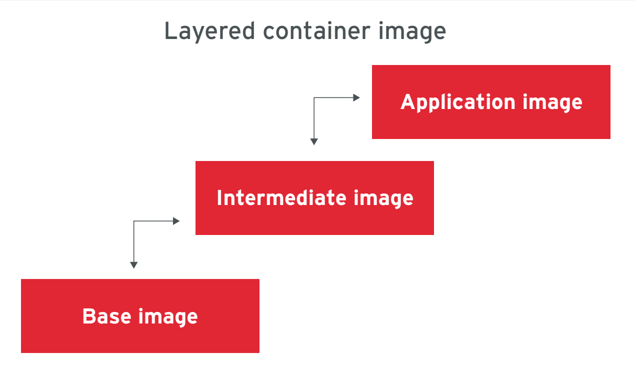
[3] *"Bug 1268347 – posix_fallocate emulation on NFS ... - Red Hat Bugzilla." 2 Oct. 2015, **https://bugzilla.redhat.com/show_bug.cgi?id=1268347**. Accessed 22 Jun. 2017.*

[4] *"Getting the minimum version of glibc for a binary - Alan's Web Ramblings." 12 Dec. 2015, **http://agardner.me/golang/cgo/c/dependencies/glibc/kernel/linux/2015/12/12/c-dependencies.html**. Accessed 22 Mar. 2017.*

To meet production-grade standards of compatibility, three specific requirements should be met:

1. All container images within a cluster should be based off of the same base image. The versions of the programs and libraries (user space) in the container image must be compatible with the container host kernel.
2. All of the container hosts in the cluster must have compatible hardware, kernels, and libraries—ideally, they should be identical. Red Hat Enterprise Linux, Red Hat Enterprise Linux Atomic Host, and all minor versions provide this level of compatibility by default with the API/ABI compatibility guidelines.[5]
3. All of the container hosts must be configured identically, especially network and storage. All configuration and access to shared storage (NFS, Gluster, iSCSI, etc.), local storage (overlay2 or devicemapper) time servers, container runtime (docker), kernel parameters, SELinux, seccomp, etc. must be identical. The simplified configuration of Red Hat Enterprise Linux Atomic Host helps to remove many of these worries.

Only when all of these requirements are met can the highest levels of portability be guaranteed. Red Hat only tests Red Hat-provided container images and container hosts together to maintain production-grade compatibility requirements. This is simplified by Red Hat's API/ABI (application binary interface) guarantee, which becomes even more critical in a cloud environment.[5]



Using layers

Figure 4. Red Hat offers certified container images that are useable for all layers of your application containers

FEATURES AND CAPABILITIES

Red Hat offers a range of images for application development and deployment.

Base images built by Red Hat are a secure and stable foundation for your applications.

- **Red Hat Enterprise Linux standard image.** Full-featured base image that can be used for any application. Includes yum, systemd, and access to any package that is included in Red Hat Enterprise Linux. Follows normal Red Hat Enterprise Linux major and minor release life cycles. Red Hat rigorously tests Red Hat Enterprise Linux 6 and 7-based versions.

[5] "Red Hat Enterprise Linux 7: Application Compatibility GUIDE - Red Hat Customer Portal" 12 Jan. 2017, https://access.redhat.com/articles/rhel-abi-compatibility. Accessed 22 Jun. 2017.

- **Red Hat Enterprise Linux atomic image.** Small and lean base image targeted toward fast-moving development environments. To reduce image size, it uses microdnf and does not include systemd. It is always based off the latest version of Red Hat Enterprise Linux 7, and only the latest minor release is supported.

Intermediate images designed and built by Red Hat provide you with a preconfigured starting point to build your applications, allowing you to focus on your application instead of the infrastructure.

- Red Hat software collections provide images for language runtimes, including node.js, perl, php, python, ruby, and rails.
- .Net Core container images give you quick access to the only commercially supported .Net Core distribution on Linux.

Application images designed and built by Red Hat help you quick-start your application by giving you access to prebuilt components, such as database, caches, ISV images, and more.

- Red Hat software collection images provide quick and easy access to Apache HTTPD, MariaDB, MySQL, Nginx, Passenger, PostgreSQL, and Redis.
- Red Hat also delivers container images from partner ISVs, ranging from NoSQL databases and messaging platforms to load balancers and security scanning tools.

All Red Hat container images are built on Red Hat Enterprise Linux. To offer users the best experience, Red Hat Enterprise Linux base images (standard and atomic) conform to these standards:

Architecture and build process

- Follow the same engineering, quality assurance, and testing as Red Hat Enterprise Linux to ensure stability and compatibility.
- Ship with core, kernel-sensitive libraries, like glibc, libseccomp, and libselinux. These libraries follow a tight integration cycle with Red Hat Enterprise Linux kernel and ensure compatibility throughout the life cycle of your container images and container hosts.
- Tightly integrate with tools like System Tap, GDB, ABRT, sosreport, tcpdump, and kdump.[6] This integration ensures that operations teams have access to all of the tools they trust to debug production clusters and allows Red Hat support to reproduce and repair problems that may arise between a container and the kernel of the host operating system.

Security

- Are protected by the the same product security team, which tracked and fixed 1,346 vulnerabilities in 2016.[7] Tracking data (Errata and OVAL) are produced for RPMs within all Red Hat images.
- Follow the same security-hardening and errata process that ensure security compliance can be met in production container clusters.

[6] *"[atomic-devel] How to handle crashes - Project Atomic." **https://lists.projectatomic.io/projectatomic-archives/atomic-devel/2015-February/msg00026.html**. Accessed 23 Mar. 2017.*

[7] *"Red Hat Product Security Risk Report 2016 - Red Hat Customer Portal." 7 Mar. 2017, **https://access.redhat.com/blogs/766093/posts/2957221**. Accessed 25 Jul. 2017.*

Performance and testing

- Are used in production, at scale within the Red Hat OpenShift Online[8] and Red Hat OpenShift Dedicated[9] environments. This makes these images battle-hardened and tested at scale in a large, multitenant Kubernetes environment.
- Have binaries built with GCC and glibc that ensure performance, stability, and security. Red Hat contributes heavily to the development and maintenance of both.
- Include access to GCC. With 332 optimization passes and 330,000 tests, Red Hat is constantly contributing code and expanding test coverage. Tests validate syntax, verify optimization passes, and check that compiled code can run and be stepped through with a debugger.[10]
- Are linked against glibc, an extremely stable and well-tested C library.
- Use the glibc DNS resolver that ensures network utilities and servers perform exactly as expected in containers.
- Are built and tested in conjunction with the Red Hat Enterprise Linux kernel to ensure compatibility between container images and host kernels for the following:
  - syscalls
  - ioctls
  - /proc and /sys (super-privileged containers)
  - glibc and the underlying host kernel (C code)

Support and certification

- Are supported at the same level as underlying Red Hat Enterprise Linux container host subscriptions (standard, premium, etc.).[11]
- For convenience, subscriptions management for containers is handled transparently. Entitlements are automatically passed from a registered container host into running containers. This makes it convenient and easy to build and run Red Hat Enterprise Linux containers on Red Hat hosts. Using subscription passthrough, users can easily install and update packages inside of Red Hat Enterprise Linux containers, whether from Red Hat Satellite or directly from the Red Hat Customer Portal. Bypassing Red Hat Subscription management, or manually passing subscription keys into Red Hat containers running on non-Red Hat hosts, is not supported.

[8] *"OpenShift Online (Next Gen) Developer Preview." https://www.openshift.com/devpreview/. Accessed 23 Mar. 2017.*

[9] *"Learn more about Red Hat OpenShift Dedicated." https://www.openshift.com/dedicated/. Accessed 23 Mar. 2017.*

[10] *"Testing... Testing... GCC – RHD Blog - Red Hat Developers." 13 Feb. 2017, https://developers.redhat.com/blog/2017/02/13/testing-testing-gcc/. Accessed 23 Mar. 2017.*

[11] *"Red Hat Container Support Policy - Red Hat Customer Portal." 23 Feb. 2017, https://access.redhat.com/articles/2726611. Accessed 23 Mar. 2017.*

CONTAINER HOSTS

TECHNOLOGY

Applications are built on container images that provide the necessary language runtimes and libraries. These containerized application images are then turned into running containers, but they have to run on a host. Running containers are really just isolated processes utilizing kernel technologies, such as SELinux, cgroups, and namespace. These kernel technologies provide layers of security and performance isolation, but do not provide the same level of separation as virtualization. This means that the version and quality of the host kernel determines the security and performance of the containerized processes. It also means that the physical architecture of the underlying hardware determines which containers can be run on a host (x86, ARM, POWER, etc.).

The value of a container host is that it is built, tested, and certified with all of the components necessary to run these containers. The container host is a major component contributing to the security, performance, and supportability of your containerized applications. The support policies and life cycle of a container host has a critical effect on the time and money required to maintain the environment over time and during upgrades.

A container host includes a tested and certified Linux kernel, a compatible container engine, and all of the libraries and tooling necessary to run, troubleshoot, diagnose, and resolve issues in a production environment—in short, end-to-end support. All of these factors should be considered when selecting the right container host for the environment.

FEATURES AND CAPABILITIES

Red Hat provides two container hosts: Red Hat Enterprise Linux Server and its variant, Red Hat Enterprise Linux Atomic Host. Both variants are built, tested, and certified to run container images based on Red Hat Enterprise Linux. Both container hosts are tested and certified with a large ecosystem of storage, network, hardware, software, hypervisors, and cloud providers. On the surface, both variants look quite similar, but the difference is really in how they are managed.

Red Hat Enterprise Linux Server is designed as a general purpose operating system for all of your applications; some customers are running thousands of instances in production. Red Hat Enterprise Linux Server provides flexibility to configure each instance as necessary to support its workload. This flexibility comes at a cost. Since a wide variety of package combinations are possible, management is inherently different and handled with more traditional tools.

Red Hat Enterprise Linux Atomic Host is designed around a different deployment, operational, and management paradigm that makes it manageable at hyperscale. All Red Hat Enterprise Linux Atomic Hosts are designed to be configured upon deployment and managed identically as immutable hosts. It is designed to be easily configured with automation, which is useful in a completely ephemeral environment, such as private cloud, public cloud, or even as a virtual machine on a developer's laptop.

In many environments, operations creates a standard operating environment (aka core build or golden image) with an ideal set of packages for the required workloads. Red Hat Enterprise Linux Atomic Host extends the concept of a core build by providing an opinionated set of packages optimized for a container host. This make Red Hat Enterprise Linux Atomic Host a lightweight option for deploying in cloud environments.

Red Hat Enterprise Linux Atomic Host is updated using a specialized technology called rpm-ostree. This makes updates on Red Hat Enterprise Linux Atomic Host very different than Red Hat Enterprise Linux Server—the entire host is updated in a single, declarative, atomic transaction. Rather than calculating which specific packages need to be updated, the atomic model provides consistency at scale and can be rolled back if needed. The immutable ostree layers and transactional updates makes Red Hat Enterprise Linux Atomic Host manageable in large, distributed system environments where control over any individual host is limited. This simplifies deployment whether you have an instance running on a developer's laptop or thousands of hosts in production.[12]

Trusted storage components are particularly important for containerized applications that require access to persistent data. Integration, testing, and the ability to support the entire software stack, from container orchestration (Kubernetes) down to the container host kernel (Linux) is critical to ensuring data integrity and security.

Red Hat employs many of the key developers at each layer of our storage stack. We contribute directly to the upstream Kubernetes, Gluster, and Linux kernel communities (and many, many others), as well as capitalizing on that in-house expertise when designing, building, and maintaining our container hosts.

Red Hat carefully certifies storage components. Each layer is tuned, secured, and carefully protects the integrity of customers' data—from data structures in the container platform (Kubernetes Persistent Volumes), to the supporting filesystems (XFS, Overlay2, etc.) and block storage components (devicemapper, iscsi, fiber channel) in the container host.

Architecture and build process

To offer users the best experience and full support,[13] Red Hat Enterprise Linux Server and Red Hat Enterprise Linux Atomic Host:

- Are built from the same set of RPM content, which enables a large ecosystem of monitoring, management agents, and support tooling.
- Ship with core, kernel-sensitive libraries, like glibc, libseccomp, and libselinux. These libraries follow a tight integration cycle with the Red Hat Enterprise Linux container images and ensure compatibility throughout the independent life cycle of your container images and container hosts.
- Tightly integrate with tools like System Tap, GDB, ABRT, sosreport, tcpdump, and kdump.[14] This ensures that operations teams have access to all of the tools they trust to debug production clusters. This allows Red Hat support to reproduce and repair problems that may arise when running containers. This includes compatibility problems between a container and the kernel of the host operating system.

[12] *"Differences between RHEL Server and RHEL Atomic Host" 31 May, 2017, https://access.redhat.com/articles/2772861.*

[13] *"Red Hat Container Support Policy - Red Hat Customer Portal." 8 Jun. 2017, https://access.redhat.com/articles/2726611. Accessed 25 Jul. 2017.*

[14] *"How to configure kexec/kdump on Atomic Host - Red Hat Customer Portal" 5 December 2016, https://access.redhat.com/solutions/2792901. Accessed 20 Mar. 2017.*

Security

- Use SELinux, cgroups, and namespaces to provide kernel-level isolation between containerized processes.
- Are protected by the the same product security team that tracked and fixed 1,346 vulnerabilities in 2016.[15] Tracking data (Errata and OVAL) are produced for RPMs within Red Hat Enterprise Linux and Red Hat Enterprise Linux Atomic Host.
- Follow the same security-hardening and errata process that ensures security compliance can be met and easily verified in production container clusters.

Performance and testing

- Follow the same engineering, quality assurance, and testing between user space and kernel.
- Include binaries built with compatible versions of GCC and glibc to ensure performance, stability, and security. Red Hat contributes heavily to the development and maintenance of both.
- Are used in production, at scale within the Red Hat OpenShift Online[16] and Red Hat OpenShift Dedicated[17] environments. This makes these container hosts hardened and tested at scale in a large, multitenant Kubernetes environment.

Storage

- Provide specific storage testing and optimizations to ensure a supportable environment:
  - Are tested with persistent volumes in Red Hat OpenShift Container Platform.
  - Support devicemapper and Overlay2 graph drivers (OverlayFS requires use of XFS).
  - Provide convenient tools to manage the dockerpool and select the graph driver for your environment.
  - Enhanced XFS error handling controls. These ensure if a thin pool becomes full, the container can be gracefully stopped.
  - Red Hat Enterprise Linux Atomic Host defaults to devicemapper backed by an lvm thin pool for optimal performance.
  - Red Hat Enterprise Linux includes the latest version of OverlayFS and ensures some historic OverlayFS defects are not in the distribution, such as:
    - Prevents Overlay whiteouts from becoming visible.
    - Fixes hardlink to unix domain socket creation issue.
    - SELinux support for OverlayFS is coming soon.

[15] *"Red Hat Product Security Risk Report 2016 - Red Hat Customer Portal." 7 Mar. 2017, https://access.redhat.com/blogs/766093/posts/2957221. Accessed 25 Jul. 2017.*

[16] *OpenShift Online Developer Preview https://www.openshift.com/devpreview/index.html*

[17] *OpenShift Dedicated: https://www.openshift.com/dedicated/index.html*

Support and certification

- Are supported at the same level as the underlying Red Hat Enterprise Linux subscription (standard, premium, etc.).
- Follow the same support matrix that ensures stability and compatibility with the same ecosystem of bare metal servers, hypervisors, private cloud platforms (OpenStack®), and public cloud providers.
- For convenience, subscriptions management for containers is handled transparently. Entitlements are automatically passed from a registered container host into running containers. This makes it convenient and easy to build and run Red Hat Enterprise Linux containers on Red Hat hosts. Using subscription passthrough, users can easily install and update packages inside Red Hat Enterprise Linux containers whether from Red Hat Satellite or directly from the Red Hat Customer Portal. Bypassing Red Hat subscription management, or manually passing subscription keys into Red Hat containers running on non-Red Hat hosts, is not supported.

CONCLUSION

While containers create the appearance of virtualization-like isolation and compartmentalization, their dependencies on the underlying operating system and support profile are still similar to those of a conventional user application. For this reason, Red Hat supports Red Hat container images as applications, with all the usual support and certification requirements for the underlying container host. This protects Red Hat customers, ensuring that they can enjoy the same level of reliability, responsiveness, and support that they expect from a Red Hat product.

Red Hat container images

Customers who have a Red Hat Enterprise Linux subscription today already have access to supported and certified Linux container images. All official Red Hat images are delivered through the Red Hat Container Catalog, giving you:

- A trusted source for official container images.
- Secure access.
- Simple, yet sophisticated, security tooling to help architects quickly decide which repositories, images, and tags to use.
- The ability to get software up and running quickly and easily.
- Access to enhanced container-focused documentation.

Red Hat container hosts

Customers who have a Red Hat Enterprise Linux subscription today already have access to supported and certified container hosts. Customers who prefer to simplify their operations at scale, or need a simplified container host for development, can take advantage of Red Hat Enterprise Linux Atomic Host, which is also available as part of your Red Hat Enterprise Linux subscription. Both Red Hat Enterprise Linux and Red Hat Enterprise Linux Atomic Host can be run on any certified bare-metal server, hypervisor, cloud platform (OpenStack), or cloud provider host, providing a number of deployment options for Red Hat customers who want to run their containers in a supported fashion.

Get access today

Visit the Red Hat Container Catalog to learn more about all of the images provided by Red Hat: http://red.ht/2qrkW8K

Visit the Red Hat Portal to learn more about the container hosts provided by Red Hat: http://red.ht/2q3g092

Visit the Red Hat Developers site to get started with a no-cost developer subscription: http://red.ht/2qkiM7H

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.

NORTH AMERICA
1 888 REDHAT1

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

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

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

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

redhat.com
#8326_0817