



Global Knowledge®

Expert Reference Series of White Papers

# New RHEL 7.1 Features Ease Business and IT Adoption

# New RHEL 7.1 Features Ease Business and IT Adoption

Kerry Doyle, MA, CPL

## Introduction

A range of new tools and features in the most recent version of Red Hat Enterprise Linux (RHEL) 7, and the incremental milestone update, RHEL 7.1, offer key advantages for business-class and IT users. Some of these capabilities include in-place upgrades, dynamic patching, support for Ceph block storage, and container technology for application development, to name a few. For example, the integration of Docker Project version 1.x with RHEL 7 now provides the ability to abstract and isolate applications. These upgraded features provide IT with tools to increase agility and offer developers the means to build high-level applications more quickly.

Linux containers are particularly helpful when an app involves multiple phases of development. The container feature can be used in daily development and in long-term operations. The same holds true for a number of other new features. Earlier adoption of the Linux 3.0.x kernel (now version 4.x) has enabled better process management, reliability, improved overall performance, and new levels of scalability.

For instance, the ability to perform dynamic patches provides greater control of uptime, increasingly important in 24/7 environments that must ensure unfettered access and dependability. In this white paper, we'll explore a number of updated features and performance improvements in RHEL 7 overall and in the latest 7.1 release. We will also examine benefits of the Linux kernel upgrade from the previous version and how it makes new management and ease-of-use capabilities possible.

## In-Place Upgrades, Performance Co-Pilot, and Livemedia-Creator

The official release of RHEL 7 has been recognized as a substantial revision aimed at architects, system administrators, and DevOps. The 7.1 upgrade further refines these new features, which tie-in to major trends currently taking place in IT and enterprise data centers in general.

For those organizations and end users committed to enterprise Linux, one important change is in-place upgrades, which help improve overall maintenance. While RHEL version 4 to version 5 upgrades were perfectly possible via DVD boot, in-place upgrades from RHEL 6 to RHEL 7 now have full support. The upgrade tool prepares a system to upgrade RPM packages, download packages, reconfigure the boot-up sequence, and then reboot. Enterprises that have standardized on the RHEL 6 platform now have the flexibility to keep their current operations intact as they explore and transition to the current container-based application infrastructure.

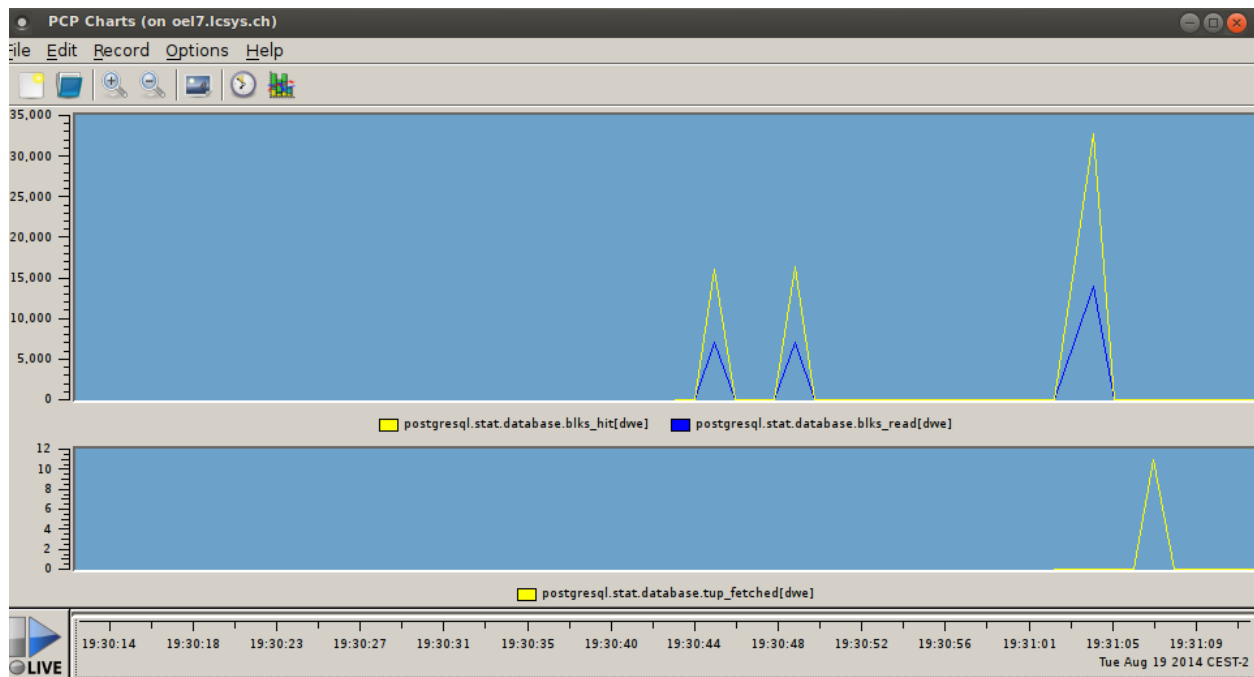
With the in-place upgrade, administrators gain better insight into overall system improvements. When upgrading, it's important to minimize risk by removing unnecessary 32-bit packages from multiarch systems before upgrading. In addition, administrators may need to add extra repository specifications to the `redhat-upgrade-tool` depending on what channels were used to install software. Example specs would include the following:

- `--addrepo=REPOID=URL`

- `--addrepo=REPOID=@MIRRORLIST`

Administrators can also easily access a uniform set of management tools for networking, storage, file systems, and performance. While overall system improvements have been a critical part of the new upgrade, measuring performance levels is also important. Performance Co-Pilot offers a useful means for system-wide monitoring.

Users can analyze and record performance via an application programming interface (API) that imports and exports sampled and fixed data. Tools for interrogating, retrieving, and then processing the collected data are also included. Performance Co-Pilot transmits this data across a network and integrates with subsystems, such as `syslogd`, `sar/sysstat`, and `systemd`. It provides both a common GUI for browsing through the collected data as well as interactive text interfaces.



## Performance Co-Pilot

Easier configurations are another key aspect of RHEL 7.1. For example, `livemedia-creator` enables the building of customized installation media from a kickstart file for different deployment use cases. The tool eases overall management of configurations on multiple machines across an enterprise, whether they are standardized corporate desktops, servers, virtual machines (VMs), or hyperscale deployments.

Essentially, `livemedia-creator` employs the Anaconda installation program (updated and improved in version 7.1), `kickstart`, and `Lorax` to create bootable media that uses the same path as a normal system install. The utility enables users to simply parse the kickstart file, extract the URL option, and then pass this as a command line argument to Anaconda. `Livemedia-creator` can be used to make live ISOs, bootable (partitioned) disk images and file system images for use with virtualization.

New kickstart options that have been added include the following:

- specifying the configuration profile name to use with thin logical volumes:  
`logvol --profile=default/thin-performance`
- creating a network bridge with various properties:  
`network --device=bro --bridgeslaves=etho --bridgeopts=stp/priority/forward-delay/hello-time/max-age/ageing-time`
- changing the default file system when using automatic partitioning:  
`autopart --fstype=ext4/xfs`
- providing the specified repository during the installation process:  
`epo --install=/path/to/repository`
- ensuring that the boot loader is not installed when using Docker:  
`bootloader --disabled`

In terms of initiating upgrades to the current 7.1 version, the new pre-upgrade package is especially helpful. An administrator can run the tool to evaluate the current RHEL system for upgrade suitability and any possible risks. Once a report run is successful, the utility uploads all reports to a convenient web app for review. The interface is useful for estimating the level of effort required for moving to RHEL 7.1 because it gauges the severity of possible issues and provides detailed analysis of potential problems.

The tool lists the pre-modified configuration files and identifies current user-modified files. It also recommends files and settings that should be manually checked. These features enable administrators to examine configuration changes and to make final decisions on completing an upgrade. Administrators can use the command line option and web app to store, examine, and compare reports from multiple hosts.

Similar to deploying workloads, it's critical to test and re-test before working on production systems. In general, there's a certain amount of evaluation that's required to upgrade an RHEL 6 system, and most systems administrators are sufficiently capable to handle the task. However, to provide further support, instructions on the Red Hat Customer Portal also include a Supported Use Cases document which offers key information for transitioning. The package defines what can be upgraded in-place and what must be performed manually.

## Benefits of the Linux 3.x and 4.x Kernels

As mentioned previously, in-place upgrades substantially ease the process of moving from previous versions to the current RHEL 7.1. That ability is partly due to the new capabilities made possible by comprehensive development of the Linux 3.x kernel. The most recent transition to the 4.x kernel was prompted by overall developer demand and the decision to avoid long version numbers. While the 3.x kernel was responsible for comprehensive improvements, the 4.x kernel does provide driver updates, performance tweaks, and bug fixes. It also introduces live kernel rebooting, a feature that was introduced earlier in RHEL 7 via kpatch.

Previous versions of the RHEL operating system came with the older, yet reliable, 2.5.x and 2.6.x kernels. In general, the Unix kernel acts as a mediator for programs and hardware. It ensures that each of the processor's cycles is distributed evenly. It also provides an interface through which programs can talk to hardware. Stable kernel updates, such as the current 4.x version, offer the ability to control more types of hardware by increasing the number of enabled device drivers.

In addition to offering new capabilities that streamline and automate installation and deployment, the 4.x kernel helps make a range of management and ease-of-use features possible. The new kernel provides increased process management and has shown performance improvements in CPU-intensive benchmarks. This is partly due to the move from the RHEL 6-modified Linux 2.6.x kernel to the current version with its Intel P-State driver and various other enhancements. The update improves the performance of Btrfs, as the default file system for the root partition, and now supports XFS file systems that are up to 500TB in size.

Additional hardware improvements, new drivers, and performance tweaks include:

- improvements to Intel 'Skylake' platform
- the merging of Intel Quark SoC support
- new ARM hardware support
- improved Toshiba laptop power settings driver, which adds USB sleep/charge functionality and other enhancements
- file system tweaks, including F2FS, Btrfs, etc.

## Dynamic Patching, OpenLMI, And Ceph Block Storage Support

While newer kernels offer the ability to talk to more types of hardware, this also increases complexity, making uptime critical. Dynamic patching provides protections against vulnerabilities in the form of patches that are applied immediately without the need to reboot, which normally requires an urgent downtime stage. It also means that critical 24/7 infrastructures don't need to sacrifice access or dependability.

There are several steps in the new Dynamic Kernel Patching capability, including the Patch Generation process and the Patch Application process. Generating a patch requires converting a source-level kernel patch into a kernel module. This module will apply the source-level change to the running system. Building a module is performed with a simple `kpatch build` command whose only input is the file containing the patch against the currently running kernel version.

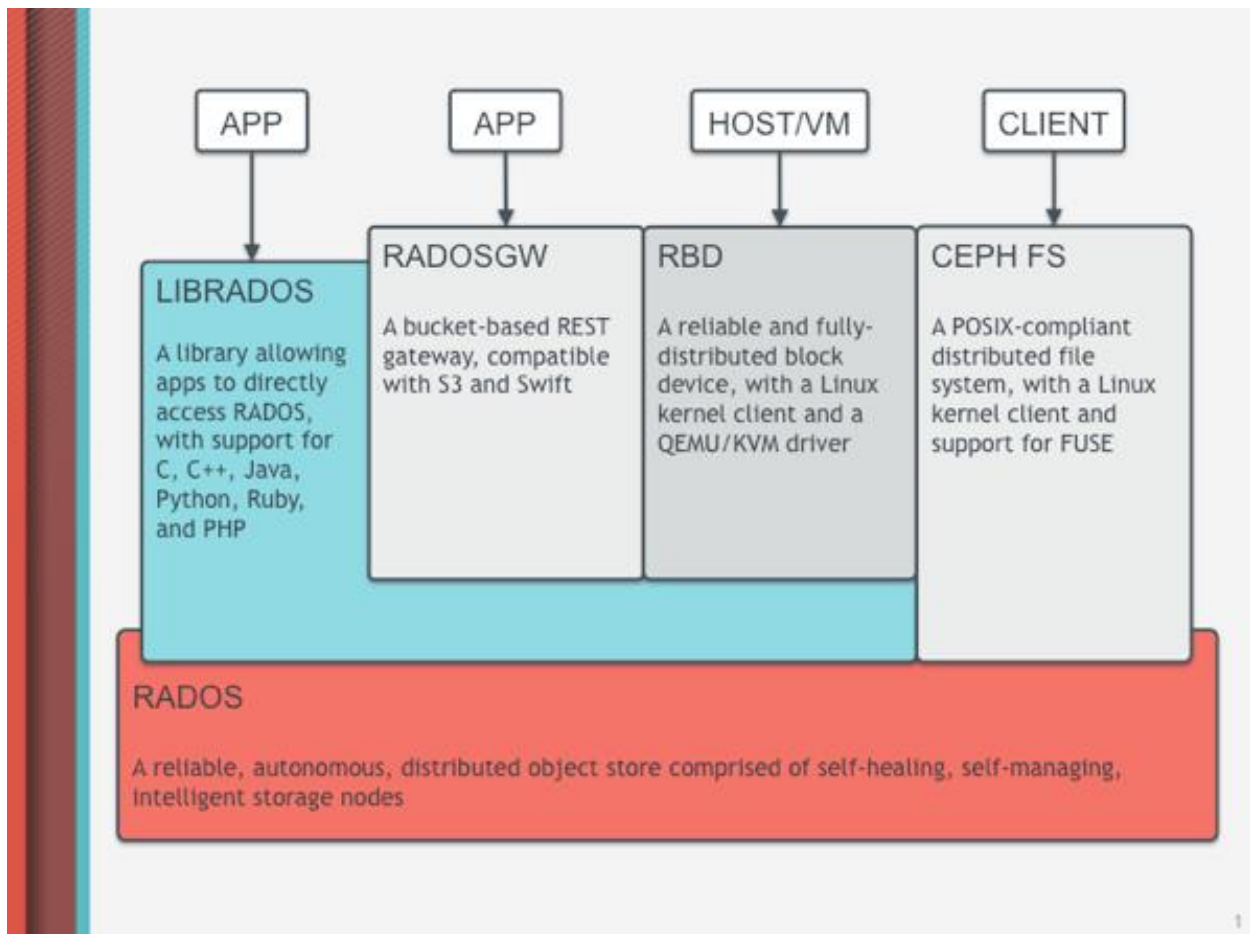
Dynamic Kernel Patching replaces a specific function in the kernel. Now available under General Public License (GPL) v.2, `kpatch`'s core module uses the kernel `ftrace` subsystem to enable administrators to more easily replace old functions with new ones, further simplifying tasks. It's 100 percent self-contained in its own module and doesn't rely on changing all the `kthreads`. A patch is applied atomically using `stop_machine()`, so it's safe with respect to data semantic changes.

Installing the hot patch module into a running system is accomplished with the `kpatch install` command. The module can also be easily removed with the `kpatch unload` command. It's important to note that `kpatch` is supported to run only on AMD64 and Intel 64 architectures. To further simplify configuration and administration, Red Hat introduced OpenLMI. The benefits of OpenLMI extend to both networking and overall infrastructure management of storage, file systems, performance, identities, and security. For example, administrators can use scripting and APIs to automate management.

Built on a model of interactive query, modify, and notify against a single system, the interface simplifies control and the standardized APIs make it easier to build custom tools. OpenLMI solves a common problem of changing volume labels that occur when hardware is reconfigured. With OpenLMI, administrators can address volumes by labels, Universal Unique Identifiers (UUID), or Device ID. In addition to simplifying configuration of network bridging and bonding, OpenLMI provides notifications of changes in network devices.

For systems with multiple drives, administrators can more easily configure and manage storage capabilities. The most recent version, RHEL 7.1, introduces support for thin provisioning in Logical Volume Manager (LVM). It also includes support for Ceph block storage, the open source DIY scale-out storage system that is able to expand linearly without the need for painful forklift upgrades. The inclusion of Ceph user space components and the Ceph RADOS Block Devices (RBD) kernel module eases access to Ceph block storage devices. Ceph is designed to have no single point of failure; therefore, it can scale to an infinite number of nodes.

Built using simple servers, with no shared component between them, Ceph ensures that each server has some amount of local storage replicating to each other via network connections. Such replication guarantees redundancy. Should a node fail, the cluster identifies the blocks with only one copy and creates a second copy somewhere else in the cluster. Thus, storage can be dynamically expanded by adding nodes to the cluster, which are immediately rebalanced by the Crush algorithm.



## Ceph Block Storage

The significance of RHEL 7.1 support is that it integrates and improves client-side functionality to communicate with Ceph block storage. Moreover, the new upgrade ensures that logical units can now be recognized with dynamic logical unit number (LUN) detection, resulting in fewer reboots and less downtime. Standardized APIs, dynamic detection, and persistent device names all help to ensure storage consistency, even when hardware and

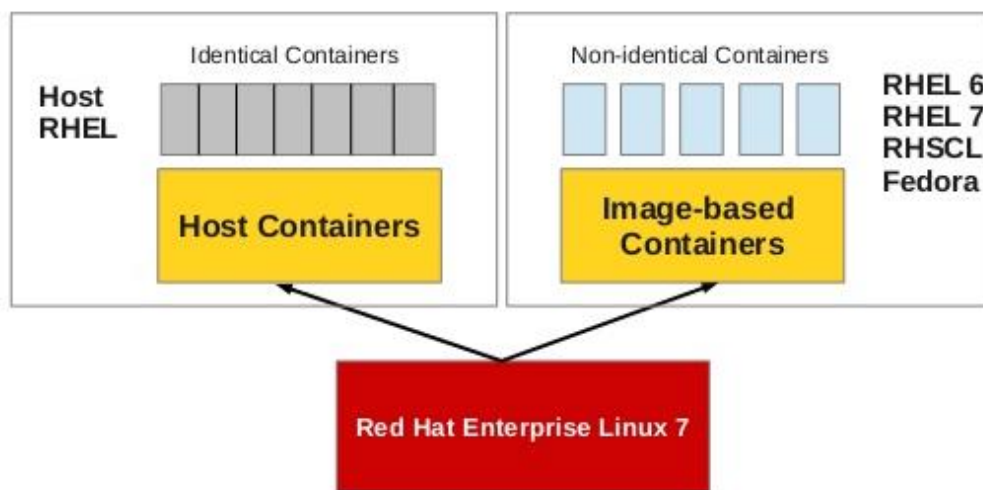
software change. Such standardized APIs are also key for remote network management to query and configure the network hardware.

## App Isolation in RHEL 7: Containers

RHEL 7 now provides the ability to abstract and isolate applications through strong integration with Docker. The base value for this container engine enables it to run on a number of different platforms and environments. This means that developers can build an app within the Linux container and subsequently run it, for example, on OpenStack or another cloud platform. But first, it's useful to know just how a container functions.

By providing a segregated section of host-based resources within the kernel, a container eliminates the need to allocate simulated hardware. It provides a filtered view of resources to the app based on kernel namespaces and capabilities, effectively simulating a separate system. An administrator can then provide discreet users, file systems, networking, and process tables, as necessary. The same goes for applications: code, libraries, binaries, and other operating services are running inside the container.

## RHEL Containers



This process of isolating applications can be critical for resolving deployment issues that plague the application development lifecycle — applications that work in development will work in production. Container technology is helpful especially when an app needs to go through multiple phases of development as well as for distribution or collaboration. RHEL 7.1 implements containers using core technologies such as control groups (cGroups) for resource management, namespaces for process isolation, and SELinux for security, reducing the potential and eliminating the conditions that contribute to security exploits. It's also a version-control system that can be used in daily development and ongoing operations practices. By packaging applications in isolated containers, developers can keep apps from competing for resources, easily choose the version of Java to employ, or resolve other development issues.

Isolating a process and simulating its environment in a host container also helps ensure that the application is safe from attack, infection, and compromise. This latest inclusion represents a significant step forward for any deployment of applications in the cloud or in straight RHEL systems. Further, RHEL 7 containers are combined with Docker and cGroups to ensure security and easy deployment. Linux containers effectively set the stage for a



new class of applications enabled through isolation and thorough testing. In addition, Red Hat container certification ensures that application containers built using RHEL will operate seamlessly across certified container hosts.

As with the adoption of any new technology, using Linux containers will require modifications. For example, container technology requires a change in application configuration and packaging on the part of the developer and independent software vendors (ISVs). The inclusion of container technology in this newest release of RHEL represents a long-term strategy to meet the needs of both developers and IT administrators.

## OpenStack and RHEL

As a key contributor to the Linux platform, Red Hat has focused on code contribution, development, and efficient distribution. In the process, RHEL has become the primary basis for commercial open source virtualization and cloud deployment. RHEL 7.1 leverages key Linux features and KVM (Kernel-based Virtual Machine) to run the OpenStack cloud infrastructure. In the most recent version, such features as database management via MariaDB, container-based app isolation, the XFS file system, and OpenLMI are all critical to a high-functioning open source cloud environment. These controls and tools help unify management, increase scalability, and reduce the administrative burden.

For example, organizations must contend with ever-increasing amounts of data. The switch to XFS as the default file system for RHEL 7.1 offers a scalable, high-performance storage and retrieval process over the earlier Btrfs format. The role of RHEL as a foundation for OpenStack means simplified workload migrations and the porting of applications between diverse providers and from on-premise environments to the cloud.

The OpenStack cloud environment effectively represents the extension of the RHEL platform into the cloud. The combination of OpenStack and RHEL functions as a single server platform that can run in either private cloud deployments, on-premise data centers, or as a public cloud. Such flexibility and scalability are also key elements for attaining true hybrid cloud deployment. One reason OpenStack has become increasingly attractive to companies is due to the levels of innovation, as well as the platform's feature velocity. OpenStack APIs provide mechanisms for extensions of value-add features including Group-Based Policy (GBP).

Such a framework defines the configuration of network policies directly to an application. GBP also designates how an application can signify its intent to an infrastructure as well as how that infrastructure can be configured by the application in a scalable, automated, and resilient fashion. The ease of working with an open source API is another key characteristic of the platform. For a developer who might be writing a recipe in Chef to deploy an application to an OpenStack cloud, API upgrading goes on behind the scenes between the configuration platform and OpenStack communities. Then, if there's a version change, the configuration platform in the middle, such as Chef or Ansible, becomes the target a developer writes to instead of the OpenStack API, further simplifying the development process. There's no reprogramming involved or need to learn new APIs because the configuration management tool has already been updated.

In its journey toward easier cross-cloud provisioning, OpenStack has already begun enabling bare metal server provisioning and application containers, key elements of workload portability and increased scalability. The latest OpenStack release, Ironic, along with the Nova compute controller, not only allows hypervisor and VM management, but also enables the deployment of workloads to physical servers while distinguishing between different types of servers.

Ultimately, more automated deployments and operations for business applications enables IT to be more agile as well as to scale more efficiently. The ability to move to a fully productized, open source private cloud that has public cloud features for end users can free up IT staff to take on other challenges. It gets companies out of proprietary cloud lock-in, while providing much more flexibility.



## Conclusion

Whether the primary goal is to build network-intensive applications or administer an enterprise-level platform that services a range of users, RHEL 7.1 offers the functionality and features to meet diverse needs. Recent changes to the operating system in this new upgrade reflect major trends critical to IT. These include increased data center automation, incorporating cloud services and big data, and reliance on software defined infrastructures, to name a few.

In addition to the new features covered here, additional refinements to RHEL 7 are of similar value. Some of these include the move to MariaDB for database management, formatting with the XFS file system, support for large storage arrays and 40GB Ethernet, as well as the inclusion of performance profiles.

Such improvements and additions represent the latest efforts designed to increase the growth of open source adoption. They're geared to easing administrative tasks, streamlining the development process, providing greater stability to end users, and expanding the platform to better accommodate newer initiatives, from open source cloud (OpenStack) and big data (Apache Hadoop) to Software Defined Networking (SDN). As these initiatives and technologies become more firmly established, expect RHEL to remain a key open source alternative.

## Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

[Red Hat System Administration I](#)

[Red Hat System Administration II](#)

[RHCSA Rapid Track Course](#)

Visit [www.globalknowledge.com](http://www.globalknowledge.com) or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

## About the Author

Kerry Doyle writes for a diverse group of companies based in technology and business. As an educator, editor at ZDNet/CNet.com, and GigaOm analyst, he has written extensively on high tech issues for over 15 years. His reports appear in a range of technology and industry publications, including *IDG/ComputerWorld*, *TechTarget*, and *UBMTechWeb*.