



Red Hat Enterprise Linux Server for ARM 7.1 Release Notes

Release Notes for Red Hat Enterprise Linux Server for ARM

Red Hat Customer Content Services

Red Hat Enterprise Linux Server for ARM 7.1 Release Notes

Release Notes for Red Hat Enterprise Linux Server for ARM

Red Hat Customer Content Services

Legal Notice

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

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

All other trademarks are the property of their respective owners.

Abstract

The Release Notes document major features implemented in Red Hat Enterprise Linux Server for ARM.

Table of Contents

Introduction	3
Chapter 1. Architecture	4
Chapter 2. Installation and Booting	5
2.1. Installer	5
2.2. Boot Loader	8
Chapter 3. Storage	9
LVM Cache	9
LVM Application Programming Interface	9
Chapter 4. File Systems	10
Support of XFS File System	10
Support of Btrfs File System	10
Fast Block Devices Caching Slower Block Devices	10
Chapter 5. Kernel	11
Chapter 6. Hardware Enablement	12
Chapter 7. System and Services	13
systemd	13
Chapter 8. Compiler and Tools	14
8.1. GCC Toolchain	14
8.2. GLIBC	14
8.3. GDB	15
8.4. Performance Tools	15
8.5. Programming Languages	17
Chapter 9. Networking	19
NetworkManager	19
Networking Team Driver	19
Precision Time Protocol	19
chrony Suite	19
Dynamic Firewall Daemon, firewalld Suite	19
DNSSEC	20
DDoS Protection	20
Network Namespaces	20
Chapter 10. Resource Management	21
Control Groups	21
Chapter 11. Security	22
OpenSSH chroot Shell Logins	22
OpenSSH - Multiple Required Authentications	22
GSS Proxy	22
Changes in NSS	22
New Boolean Names	22
Chapter 12. Subscription Management	23
Certificate-Based Entitlements	23
Chapter 13. Web Servers and Services	24
Apache HTTP Server 2.4	24

Apache HTTP Server 2.4	24
MariaDB 5.5	24
PostgreSQL 9.2	24
Chapter 14. Supportability and Maintenance	25
ABRT 2.1	25
Additional Information on ABRT	25
Revision History	26

Introduction

Red Hat is pleased to announce the availability of Red Hat Enterprise Linux Server for ARM 7.1.

In this release, Red Hat brings together improvements across the server, systems, and the overall Red Hat open source experience. Red Hat Enterprise Linux Server for ARM 7.1 introduces:

- » support for the ARMv8-A architecture
- » **XFS** as the default file system;
- » a new boot loader and a fully redesigned graphical installer;
- » the **systemd** system and service manager.

Capabilities and limits of Red Hat Enterprise Linux Server for ARM are currently undefined. While the theoretical limits are essentially the same as with the AMD64 and Intel 64 architectures, the actual limits can differ depending on the selected platform, which will be listed in later releases.

Chapter 1. Architecture

Red Hat Enterprise Linux Server for ARM is available as a single kit on the 64-bit ARM (AArch64) architecture. Note that the Red Hat Enterprise Linux Server for ARM installation is only supported on 64-bit hardware.

Chapter 2. Installation and Booting

2.1. Installer

The Red Hat Enterprise Linux Server for ARM installer (known as **anaconda**) assists in the installation of Red Hat Enterprise Linux Server for ARM. This section of the *Release Notes* provides an overview of the new features implemented in the installer for Red Hat Enterprise Linux Server for ARM.

The new installer in Red Hat Enterprise Linux Server for ARM features a wide range of bug fixes and enhancements, including: a fully redesigned graphical installer and major updates to the storage configuration tools.

2.1.1. Installation Methods

The installer provides three main interfaces to install Red Hat Enterprise Linux Server for ARM:

- the graphic installer,
- the text-based installer,
- and **kickstart**.

2.1.1.1. Graphical Installer

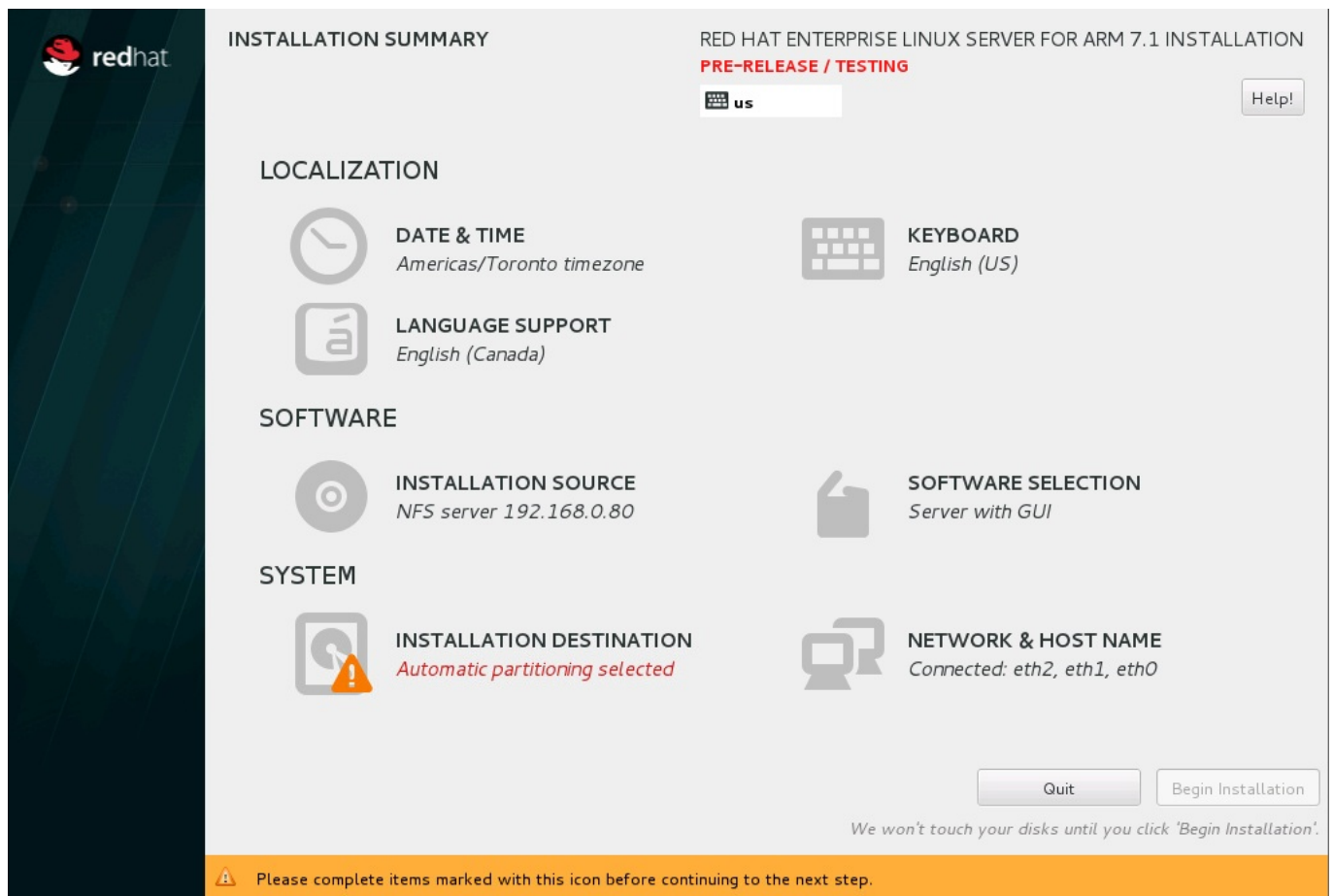
The Red Hat Enterprise Linux Server for ARM graphical installer provides an intuitive graphical user interface to prepare a system for installation. The Red Hat Enterprise Linux Server for ARM graphical installer introduces a brand new user interface designed to make installation quicker and easier.



Important

Normally, only the text mode installation interface is available during installation of Red Hat Enterprise Linux Server for ARM. To access the graphical user interface, you must connect to the system remotely using **VNC**. See Red Hat Enterprise Linux Server for ARM 7.1 Installation Guide for information about performing an installation over **VNC**.

Previously, the installer was a series of wizard-style screens that required the user to review the settings and click to get to the next screen. The new installer interface provides a central hub that lists groups of configuration options for an installation; the user clicks on the options that need changing, changes them, then initiates the installation.



The new graphical installer also generates automatic default settings where applicable. For example, if the installer detects a network connection, the user's general location is determined with GeoIP and sane suggestions are made for the default keyboard layout, language and timezone.

Additionally, the graphical installer processes some tasks concurrently (for example, storage layout detection) allowing the user to continue configuring the installation using the GUI while the processor-intensive tasks are processed in the background.

2.1.1.2. Text-Based Installer

The text-based installer is provided primarily for systems with limited resources. Red Hat Enterprise Linux Server for ARM features a completely rewritten text-mode installer that provides better support for serial consoles and other limited display interfaces. The text-based installer utilizes the **tmux** utility, making multiple shell terminals available for all installation methods, not just those supporting Linux virtual consoles.

2.1.1.3. Kickstart

Kickstart is an automated installation method that system administrators can use to install Red Hat Enterprise Linux Server for ARM. Using **kickstart**, a single file is created, containing the answers to all the questions that would normally be asked during a typical installation. **Kickstart** in Red Hat Enterprise Linux Server for ARM supports Active Directory host enrollment using the **kickstart** service **realmd**.

2.1.2. Plug-In Architecture

The installer in Red Hat Enterprise Linux Server for ARM supports the development of plug-ins that can provide site-specific extensions or customization. Plug-ins can be developed to add additional screens and options to the graphical installer. The plug-in architecture also allows developers to add new **kickstart** commands for system administrators to utilize.

2.1.3. Storage Features and Enhancements

2.1.3.1. Custom Partitioning

MANUAL PARTITIONING RED HAT ENTERPRISE LINUX SERVER FOR ARM 7.1 INSTALLATION
PRE-RELEASE / TESTING

Done us Help!

New Red Hat Enterprise Linux Server for ARM 7.1 Installation

DATA

/home 639.89 GiB
rhelsfa_mustang-home

SYSTEM

/boot 500 MiB
sda2

/boot/efi 200 MiB
sda1

/ 50 GiB >
luks-rhelsfa_mustang-root

swap 8192 MiB
rhelsfa_mustang-swap

+ - ↺ 🗑

AVAILABLE SPACE 57.83 MiB **TOTAL SPACE** 698.64 GiB

[1 storage device selected](#)

luks-rhelsfa_mustang-root

Mount Point:

Device(s): ATA WDC WD7500BPKX-2 (sda)
Modify...

Desired Capacity:

Device Type: LVM ☒ Encrypt

Volume Group: rhelsfa_mustang (4096 KiB free)
Modify...

File System: xfs ☒ Reformat

Label:

Name:

Update Settings

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

Reset All

2.1.3.2. Rescanning Storage

The installer does not expose all possible storage tunables in the user interface. In order to accommodate users who require very low-level configuration of their storage, the user can exit the installer to perform their storage configuration. The user can then return to the installer and have it rescan the storage to detect their configuration and present it in the graphical interface.

2.1.3.3. Automatic Partitioning

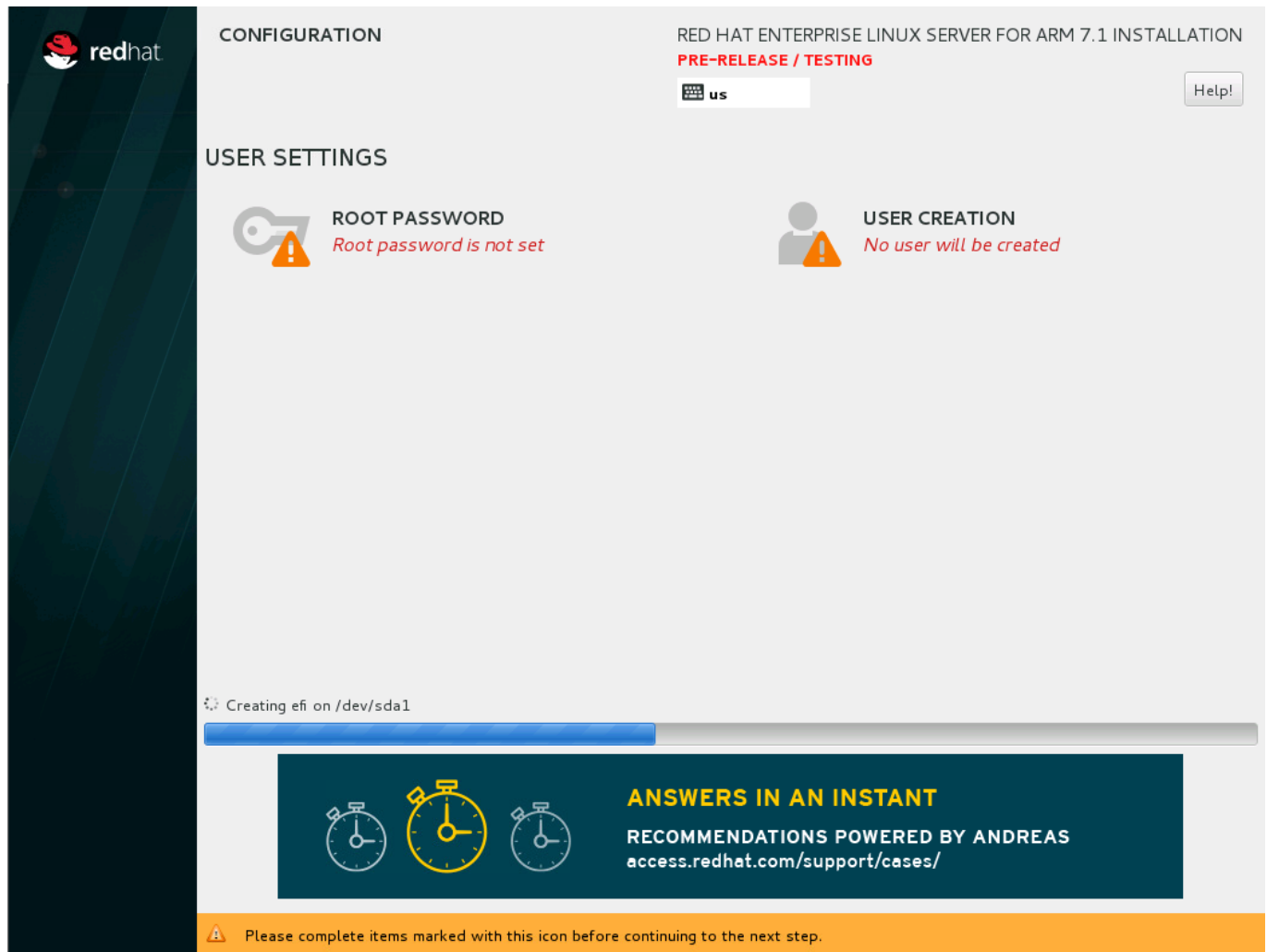
The Red Hat Enterprise Linux Server for ARM installer offers more automatic partitioning choices; for example, LVM, LVM with thin provisioning, BTRFS, or standard partitions.

2.1.3.4. Installation Environment

Previously, the installation environment implemented its own initialization and device discovery tools that were different from the tools used to boot the installed system. In Red Hat Enterprise Linux Server for ARM, the installer utilizes the same initialization (**systemd**) and device discovery tools (**dracut**) as the installed system.

2.1.3.5. Initial System Configuration

After installation, the initial system configuration screens allow for further configuration of Red Hat Enterprise Linux Server for ARM installation. The Initial System Configuration screens in Red Hat Enterprise Linux Server for ARM are also redesigned to match the user experience of the new installer graphical interface. Additionally, some tasks that were traditionally only configurable post-installation (for example, creating the initial user) can now be configured in the installer while the system is being installed.



2.2. Boot Loader

GRUB 2

Red Hat Enterprise Linux Server for ARM includes a new boot loader, GRUB 2, which is more robust, portable, and powerful than its predecessor, GRUB. GRUB 2 provides a number of features and improvements, the most notable of which are:

- » GRUB 2 supports EFI firmware on the 64-bit ARM (AArch64) architecture.
- » In addition to supporting Master Boot Record (MBR) partition tables, GRUB 2 supports GUID Partition Tables (GPT).
- » GRUB 2 also supports the ext2, ext3, ext4, and XFS Linux file systems.

Chapter 3. Storage

LVM Cache

Red Hat Enterprise Linux Server for ARM introduces LVM cache as a Technology Preview. This feature allows users to create logical volumes with a small fast device performing as a cache to larger slower devices. Please refer to the **lvmd(8)** manual page for information on creating cache logical volumes.

Note that the following commands are not currently allowed on cache logical volumes:

- ✦ **pvmove**: will skip over any cache logical volume;
- ✦ **lvresize**, **lvreduce**, **lvextend**: cache logical volumes cannot be resized currently;
- ✦ **vgsplit**: splitting a volume group is not allowed when cache logical volumes exist in it.

LVM Application Programming Interface

Red Hat Enterprise Linux Server for ARM features the new LVM application programming interface (API) as a Technology Preview. This API is used to query and control certain aspects of LVM.

Refer to the **lvmdapp.h** header file for more information.

Chapter 4. File Systems

Support of XFS File System

The default file system for an **Anaconda**-based installation of Red Hat Enterprise Linux Server for ARM is **XFS**. The **ext4**, **ext3** and **ext2** file systems can be used as alternatives to **XFS**.

XFS is a highly scalable, high-performance file system which was originally designed at Silicon Graphics, Inc. It was created to support file systems up to 16 exabytes (approximately 16 million terabytes), files up to 8 exabytes (approximately 8 million terabytes) and directory structures containing tens of millions of entries. **XFS** supports metadata journaling, which facilitates quicker crash recovery. **XFS** file system can also be defragmented and expanded while mounted and active. Note that it is not possible to shrink XFS file system.

For information about changes between commands used for common tasks in **ext4** and **XFS**, see the Reference Table in the [Installation Guide](#).

Support of Btrfs File System

The **Btrfs** (B-Tree) file system is supported as a Technology Preview in Red Hat Enterprise Linux Server for ARM. This file system offers advanced management, reliability, and scalability features. It enables users to create snapshots, it allows for compression and integrated device management.

For more information about the Btrfs Technology Preview, see [Storage Administration Guide](#).

Fast Block Devices Caching Slower Block Devices

LVM provides the ability to have fast block devices act as a cache for slower block devices. This feature is introduced as a Technology Preview in Red Hat Enterprise Linux Server for ARM and allows a PCIe SSD device to act as a cache for direct-attached storage (DAS) or storage area network (SAN) storage, which improves file system performance.

For more information, refer to the *LVM Cache* entry in [Chapter 3, Storage](#) and the **lvmm(8)** manual page.

Chapter 5. Kernel

Red Hat Enterprise Linux Server for ARM 7.1 includes the *kernel* 3.19.0 with the Advanced Configuration and Power Interface (ACPI) enabled by default.

Chapter 6. Hardware Enablement

Red Hat Enterprise Linux Server for ARM only supports hardware designs based on 64-bit ARM v8-A specification that are also compliant with Server Base System Architecture (SBSA) and Server Base Boot Requirements (SBBR) standards. Red Hat Enterprise Linux Server for ARM 7.1 includes support for certified hardware platforms that are based on Applied Micro X-Gene system-on-chip (SoC) and AMD Opteron A1100-Series SoC.

Chapter 7. System and Services

systemd

systemd is a system and service manager for Linux, and replaces SysV and Upstart used in previous releases of Red Hat Enterprise Linux Server for ARM. systemd is compatible with SysV and Linux Standard Base init scripts.

systemd offers, among others, the following capabilities:

- » Aggressive parallelization capabilities;
- » Use of socket and D-Bus activation for starting services;
- » On-demand starting of daemons;
- » Managing of control groups;
- » Creating of system state snapshots and restoring of the system state.

For detailed information about systemd and its configuration, see [System Administrator's Guide](#).

Chapter 8. Compiler and Tools

8.1. GCC Toolchain

In Red Hat Enterprise Linux Server for ARM, the GCC toolchain is based on the gcc-4.8.x release series and includes numerous enhancements and bug fixes. Similarly, Red Hat Enterprise Linux Server for ARM includes binutils-2.23.52.x.

Notable highlights of the Red Hat Enterprise Linux Server for ARM toolchain are the following:

- ✧ Experimental support for building applications compliant with C++11 (including full C++11 language support) and some experimental support for C11 features;
- ✧ Improved support for programming parallel applications, including OpenMP v3.1, C++11 Types and GCC Built-ins for Atomic Memory Access and experimental support for transactional memory (including Intel RTM/HLE intrinsics, built-ins, and code generation);
- ✧ A new local register allocator (LRA), improving code performance;
- ✧ DWARF4 is now used as the default debug format;
- ✧ A variety of new architecture-specific options;
- ✧ Link-time optimization support;
- ✧ Enhanced warnings and diagnostics;
- ✧ A variety of new Fortran features.

8.2. GLIBC

In Red Hat Enterprise Linux Server for ARM, the **glibc** libraries (**libc**, **libm**, **libpthread**, NSS plug-ins, and others) are based on the **glibc** 2.17 release, which includes numerous enhancements and bug fixes.

Notable highlights of the Red Hat Enterprise Linux Server for ARM glibc libraries are the following:

- ✧ Experimental ISO C11 support;
- ✧ New Linux interfaces: **prlimit**, **prlimit64**, **fanotify_init**, **fanotify_mark**, **clock_adjtime**, **name_to_handle_at**, **open_by_handle_at**, **syncfs**, **setns**, **sendmmsg**, **process_vm_readv**, **process_vm_writev**;
- ✧ Checking versions of the FD_SET, FD_CLR, FD_ISSET, poll, and ppoll file descriptors added;
- ✧ Caching of the netgroup database is now supported in the **nscd** daemon;
- ✧ The new function **secure_getenv()** allows secure access to the environment, returning NULL if running in a SUID or SGID process. This function replaces the internal function **__secure_getenv()**;
- ✧ The **crypt()** function now fails if passed salt bytes that violate the specification for those values. On Linux, the **crypt()** function will consult the **/proc/sys/crypto/fips_enabled** file to determine if FIPS mode is enabled, and fail on encrypted strings using the Message-Digest algorithm 5 (MD5) or Data Encryption Standard (DES) algorithm when the mode is enabled;
- ✧ The **clock_*** suite of functions (declared in **<time.h>**) is now available directly in the main C library. Previously it was necessary to link with **-lrt** to use these functions. This change has the effect that a single-threaded program that uses a function such as **clock_gettime()** (and is not linked with **-lrt**)

will no longer implicitly load the pthreads library at runtime and so will not suffer the overheads associated with multi-thread support in other code such as the C++ runtime library;

- New header `<sys/auxv.h>` and function **getauxval()** allow easy access to the `AT_*` key-value pairs passed from the Linux kernel. The header also defines the `HWCAP_*` bits associated with the `AT_HWCAP` key;

8.3. GDB

In Red Hat Enterprise Linux Server for ARM, the GDB debugger is based on the *gdb-7.6.1* release, and includes numerous enhancements and bug fixes.

This version corresponds to GDB in Red Hat Developer Toolset 2.1.

Notable new features of **GDB** included in Red Hat Enterprise Linux Server for ARM are the following:

- Faster loading of symbols using the new **.gdb_index** section and the new **gdb-add-index** shell command.
- **gdbserver** now supports standard input/output (STDIO) connections, for example: **(gdb) target remote | ssh myhost gdbserver - hello;**
- Improved behavior of the **watch** command using the **-location** parameter;
- Virtual method tables can be displayed by a new command, **info vtbl;**
- Control of automatic loading of files by new commands **info auto-load**, **set auto-load**, and **show auto-load;**
- Displaying absolute path to source file names using the **set filename-display absolute** command;
- Control flow recording with hardware support by a new command, **record btrace.**

In addition, Red Hat Enterprise Linux Server for ARM provides a new package, *gdb-doc*, which contains the GDB Manual in PDF, HTML, and info formats. The GDB Manual was part of the main RPM package in previous versions of Red Hat Enterprise Linux Server for ARM.

8.4. Performance Tools

Red Hat Enterprise Linux Server for ARM includes updates to the most recent versions of several performance tools, such as **oprofile**, **papi**, and **elfutils**, bringing performance, portability, and functionality improvements.

Moreover, Red Hat Enterprise Linux Server for ARM premieres:

- Support for Performance Co-Pilot;
- SystemTap support for (DynInst-based) instrumentation that runs entirely in unprivileged user space, as well as efficient (Byteman-based) pinpoint probing of Java applications;
- Valgrind support for hardware transactional memory and improvements in modeling vector instructions.

8.4.1. Performance Co-Pilot

Performance Co-Pilot (PCP) has not yet been implemented for Red Hat Enterprise Linux Server for ARM.

8.4.2. SystemTap

Red Hat Enterprise Linux Server for ARM 7.1 includes *systemtap* version 2.7. Currently, only tracepoints have been implemented. Other features will be added in later releases.

8.4.3. Valgrind

Red Hat Enterprise Linux Server for ARM includes **Valgrind**, an instrumentation framework that includes a number of tools to profile applications. This version is based on the **Valgrind** 3.10.0.

Notable new features of **Valgrind** included in Red Hat Enterprise Linux Server for ARM are the following:

- ✧ The default size of the translation cache has been increased to 16 sectors, reflecting the fact that large applications require instrumentation and storage of huge amounts of code. For similar reasons, the number of memory mapped segments that can be tracked has been increased by a factor of 6. The maximum number of sectors in the translation cache can be controlled by the new flag **--num-transtab-sectors**;
- ✧ **Valgrind** no longer temporarily creates a mapping of the entire object to read from it. Instead, reading is done through a small fixed sized buffer. This avoids virtual memory spikes when **Valgrind** reads debugging information from large shared objects;
- ✧ The list of used suppressions (displayed when the **-v** option is specified) now shows, for each used suppression, the file name and line number where the suppression is defined;
- ✧ A new flag, **--sigill-diagnostics** can now be used to control whether a diagnostic message is printed when the just-in-time (JIT) compiler encounters an instruction it cannot translate. The actual behavior — delivery of the SIGILL signal to the application — is unchanged.
- ✧ The **Memcheck** tool has been improved with the following features:
 - Improvements in handling of vector code, leading to significantly fewer false error reports. Use the **--partial-loads-ok=yes** flag to get the benefits of these changes;
 - Better control over the leak checker. It is now possible to specify which kind of leaks (definite, indirect, possible, and reachable) should be displayed, which should be regarded as errors, and which should be suppressed by a given leak suppression. This is done using the options **--show-leak-kinds=kind1,kind2,...**, **--errors-for-leak-kinds=kind1,kind2,...** and an optional **match-leak-kinds:** line in suppression entries, respectively;

Note that generated leak suppressions contain this new line and are therefore more specific than in previous releases. To get the same behavior as previous releases, remove the **match-leak-kinds:** line from generated suppressions before using them;

- Reduced **possible leak** reports from the leak checker by the use of better heuristics. The available heuristics provide detection of valid interior pointers to `std::string`, to new[] allocated arrays with elements having destructors, and to interior pointers pointing to an inner part of a C++ object using multiple inheritance. They can be selected individually using the **--leak-check-heuristics=heur1,heur2,...** option;
- Better control of stacktrace acquisition for heap-allocated blocks. Using the **--keep-stacktraces** option, it is possible to control independently whether a stack trace is acquired for each allocation and deallocation. This can be used to create better "use after free" errors or to decrease Valgrind's resource consumption by recording less information;

- Better reporting of leak suppression usage. The list of suppressions used (shown when the **-v** option is specified) now shows, for each leak suppression, how many blocks and bytes it suppressed during the last leak search.
- ✧ The Valgrind GDB server integration has been improved with the following monitoring commands:
 - A new monitor command, **v.info open_fds**, that gives the list of open file descriptors and additional details;
 - A new monitor command, **v.info execontext**, that shows information about the stack traces recorded by Valgrind;
 - A new monitor command, **v.do expensive_sanity_check_general**, to run certain internal consistency checks.

8.5. Programming Languages

Ruby 2.0.0

Red Hat Enterprise Linux Server for ARM 7.1 provides the latest Ruby version, 2.0.0. The most notable of the changes between version 2.0.0 and 1.8.7 are the following:

- ✧ New interpreter, YARV (yet another Ruby VM), which significantly reduces loading times, especially for applications with large trees or files;
- ✧ New and faster "Lazy Sweep" garbage collector;
- ✧ Ruby now supports string encoding;
- ✧ Ruby now supports native threads instead of green threads.

For more information about Ruby 2.0.0, consult the upstream pages of the project: <https://www.ruby-lang.org/en/>.

Python 2.7.5

Red Hat Enterprise Linux Server for ARM includes Python 2.7.5, which is the latest Python 2.7 series release. This version contains many improvements in performance and provides forward compatibility with Python 3. The most notable of the changes in Python 2.7.5 are the following:

- ✧ An ordered dictionary type;
- ✧ A faster I/O module;
- ✧ Dictionary comprehensions and set comprehensions;
- ✧ The sysconfig module.

For the full list of changes, see <http://docs.python.org/dev/whatsnew/2.7.html>

Java 7 and Multiple JDKs

Red Hat Enterprise Linux Server for ARM 7.1 features OpenJDK7 as the default Java Development Kit (JDK) and Java 7 as the default Java version. All Java 7 packages allow installation of multiple versions in parallel, similarly to the kernel.

The ability of parallel installation allows users to try out multiple versions of the same JDK simultaneously, to tune performance and debug problems if needed. The precise JDK is selectable through `/etc/alternatives/` as before.



Important

Although installing multiple versions of Java in parallel is supported, no other vendors currently provide Java for the 64-bit ARM (AArch64) architecture. Versions provided by other vendors can become available at a later date.

Chapter 9. Networking

NetworkManager

NetworkManager has been significantly enhanced to configure and monitor all the networking features for enterprise class servers and for desktop applications.

For the enterprise data centers, **NetworkManager** can be used for tasks such as basic networking configuration, network teaming, configuring virtual LANs, bridges, bonds, IPv6, VPNs, assigning interfaces to firewall zones, and others. For desktop servers it can manage wired and wireless networks and VPNs.

NetworkManager now comes with three types of interfaces:

- a robust CLI interface that allows users and scripts to interact with NetworkManager;
- NetworkManager TUI that is a text-based highlight-and-select type of interface;
- NetworkManager GUI that is more suitable for GUI desktop environments.

NetworkManager can also work side by side with initscripts if the system administrators prefer a mixed environment. NetworkManager also has full support for D-Bus as well as OpenLMI interfaces.

Networking Team Driver

In the past, the bonding driver was used for all types of link aggregation, which created various challenges. Network Teaming has been introduced as an alternative to bonding for link aggregation. The Team driver offers performance and flexibility improvements. Unlike with bonding, the control and management interface is located in user space and the fast data path is in kernel space. The Team driver supports all of the features supported by the bonding driver. A migration tool, **bond2team**, to assist with migration from bonding to teaming is also available.

Precision Time Protocol

Precision Time Protocol, or PTP, as defined in the IEEE 1588 standard, is fully supported in Red Hat Enterprise Linux Server for ARM. PTP can be used for precisely synchronizing distributed system clocks. It is capable of achieving clock accuracy in the sub-microsecond range when used in conjunction with PTP-enabled hardware devices. When used in combination with **ntpd** or **chrony**, it can be used to accurately synchronize time from the host to virtual machines. PTP also has the capability to use clock signals from GPS satellites, thus providing the same exact sub-microsecond accuracy across the globe.

chrony Suite

The **chrony** suite of utilities is available to update the system clock on systems that do not fit into the conventional permanently networked, always on, dedicated server category. The **chrony** suite should be considered for all systems which are frequently suspended or otherwise intermittently disconnected and reconnected to a network. Mobile and virtual systems for example.

Dynamic Firewall Daemon, firewalld Suite

Red Hat Enterprise Linux Server for ARM includes the dynamic firewall daemon, **firewalld**, which provides a dynamically managed firewall with support for network "zones" to assign a level of trust to a network and its associated connections and interfaces. It has support for **IPv4** and **IPv6** firewall settings. It supports Ethernet bridges and has a separation of runtime and permanent configuration options. It also has an interface for services or applications to add firewall rules directly.

DNSSEC

DNSSEC is a set of Domain Name System Security Extensions (DNSSEC) that allow s clients to determine origin authentication of **DNS** data, authenticated denial of existence and data integrity. DNSSEC prevents man-in-the-middle attacks in w hich active eavesdropping or intercepted communication occurs betw een tw o systems.

DDoS Protection

Distributed Denial of Service (DDoS) attacks are increasing, and becoming commonplace, as more and more products and services become dependent on delivering services over the Internet. The **SYNPROXY** module is designed to protect the system against common SYN-floods and ACK-floods, but can also be adjusted to protect against SYN-ACK floods. The **SYNPROXY** module filters out false SYN-ACK and ACK packets before the socket enters the "listen" state lock.

Network Namespaces

Netw ork namespaces provides a lightw eight container-based virtualization that allow s virtual netw ork stacks to be associated w ith a process group. It creates an isolated copy of the netw orking data structures such as the interface list, sockets, routing table, the **/proc/net/** directory, port numbers, and so on. Netw ork namespaces is managed through the **ip** interface (sometimes also referred to as **iproute2**), namely by the **ip netns** command.

Chapter 10. Resource Management

Control Groups

Red Hat Enterprise Linux Server for ARM features control groups, cgroups, which is a concept for organizing processes in a tree of named groups for the purpose of resource management. They provide a way to hierarchically group and label processes and a way to apply resource limits to these groups. In Red Hat Enterprise Linux Server for ARM, control groups are exclusively managed through **systemd**. Control groups are configured in systemd unit files and are managed with systemd's command line interface (CLI) tools.

Control groups and other resource management features are discussed in detail in the [Resource Management and Linux Containers Guide](#).

Chapter 11. Security

OpenSSH chroot Shell Logins

Generally, each Linux user is mapped to an SELinux user using SELinux policy, enabling Linux users to inherit the restrictions placed on SELinux users. There is a default mapping in which Linux users are mapped to the SELinux `unconfined_u` user.

In Red Hat Enterprise Linux 7, the **ChrootDirectory** option for chrooting users can be used with unconfined users without any change, but for confined users, such as `staff_u`, `user_u`, or `guest_u`, the SELinux `selinuxuser_use_ssh_chroot` variable has to be set. Administrators are advised to use the `guest_u` user for all chrooted users when using the **ChrootDirectory** option to achieve higher security.

OpenSSH - Multiple Required Authentications

Red Hat Enterprise Linux Server for ARM supports multiple required authentications in SSH protocol version 2 using the **AuthenticationMethods** option. This option lists one or more comma-separated lists of authentication method names. Successful completion of all the methods in any list is required for authentication to complete. This enables, for example, requiring a user to have to authenticate using the public key or GSSAPI before they are offered password authentication.

GSS Proxy

GSS Proxy is the system service that establishes GSS API Kerberos context on behalf of other applications. This brings security benefits; for example, in a situation when the access to the system keytab is shared between different processes, a successful attack against that process leads to Kerberos impersonation of all other processes.

Changes in NSS

The `nss` packages have been upgraded to upstream version 3.15.2. Message-Digest algorithm 2 (MD2), MD4, and MD5 signatures are no longer accepted for online certificate status protocol (OCSP) or certificate revocation lists (CRLs), consistent with their handling for general certificate signatures.

Advanced Encryption Standard Galois Counter Mode (AES-GCM) Cipher Suite (RFC 5288 and RFC 5289) has been added for use when TLS 1.2 is negotiated. Specifically, the following cipher suites are now supported:

- ✧ `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`;
- ✧ `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`;
- ✧ `TLS_DHE_RSA_WITH_AES_128_GCM_SHA256`;
- ✧ `TLS_RSA_WITH_AES_128_GCM_SHA256`.

New Boolean Names

Several SELinux boolean names have been changed to be more domain-specific. The old names can still be used, however, only the new names will appear in the lists of booleans.

The old boolean names and their respective new names are available from the `/etc/selinux/<policy_type>/booleans.subs_dist` file.

Chapter 12. Subscription Management

Red Hat Enterprise Linux Server for ARM is available using the Red Hat Subscription Management services. The following [Knowledge Base article](#) provides a brief overview and instructions on how to register your Red Hat Enterprise Linux Server for ARM system with Red Hat Subscription Management.

Certificate-Based Entitlements

Red Hat Enterprise Linux Server for ARM supports new certificate-based entitlements through the **subscription-manager** tool.

Chapter 13. Web Servers and Services

Apache HTTP Server 2.4

Version 2.4 of the Apache HTTP Server (**httpd**) is included in Red Hat Enterprise Linux Server for ARM, and offers a range of new features:

- an enhanced version of the "Event" processing module, improving asynchronous request process and performance;
- native FastCGI support in the **mod_proxy** module;
- support for embedded scripting using the Lua language.

More information about the features and changes in **httpd** 2.4 can be found at http://httpd.apache.org/docs/2.4/new_features_2_4.html. A guide to adapting configuration files is also available: <http://httpd.apache.org/docs/2.4/upgrading.html>.

MariaDB 5.5

MariaDB is the default implementation of MySQL in Red Hat Enterprise Linux Server for ARM. MariaDB is a community-developed fork of the MySQL database project, and provides a replacement for MySQL. MariaDB preserves API and ABI compatibility with MySQL and adds several new features; for example, a non-blocking client API library, the Aria and XtraDB storage engines with enhanced performance, better server status variables, and enhanced replication.

Detailed information about MariaDB can be found at <https://mariadb.com/kb/en/what-is-mariadb-55/>.

PostgreSQL 9.2

PostgreSQL is an advanced Object-Relational database management system (DBMS). The *postgresql* packages include the PostgreSQL server package, client programs, and libraries needed to access a PostgreSQL DBMS server.

Red Hat Enterprise Linux Server for ARM features version 9.2 of PostgreSQL. For a list of new features, bug fixes and possible incompatibilities against version 8.4 packaged in Red Hat Enterprise Linux 6, please refer to the upstream release notes:

- <http://www.postgresql.org/docs/9.2/static/release-9-0.html>
- <http://www.postgresql.org/docs/9.2/static/release-9-1.html>
- <http://www.postgresql.org/docs/9.2/static/release-9-2.html>

Or the PostgreSQL wiki pages:

- http://wiki.postgresql.org/wiki/What's_new_in_PostgreSQL_9.0
- http://wiki.postgresql.org/wiki/What's_new_in_PostgreSQL_9.1
- http://wiki.postgresql.org/wiki/What's_new_in_PostgreSQL_9.2

Chapter 14. Supportability and Maintenance

ABRT 2.1

Red Hat Enterprise Linux Server for ARM includes the **Automatic Bug Reporting Tool (ABRT)** 2.1, which features an improved user interface and the ability to send *μReports*, lightweight anonymous problem reports suitable for machine processing, such as gathering crash statistics. The set of supported languages, for which **ABRT** is capable of detecting problems, has been extended with the addition of Java and Ruby in **ABRT 2.1**.

In order to use **ABRT**, ensure that the *abrt-desktop* or the *abrt-cli* package is installed on your system. The *abrt-desktop* package provides a graphical user interface for **ABRT**, and the *abrt-cli* package contains a tool for using **ABRT** on the command line. You can also install both.

To install the package containing the graphical user interface for **ABRT**, run the following command as the **root** user:

```
~]# yum install abrt-desktop
```

To install the package that provides the command line **ABRT** tool, use the following command:

```
~]# yum install abrt-cli
```

Note that while both of the above commands cause the main **ABRT** system to be installed, you may need to install additional packages to obtain support for detecting crashes in software programmed using various languages. See the *Automatic Bug Reporting Tool (ABRT)* chapter of the [Red Hat Enterprise Linux 7 System Administrator's Guide](#) for information on additional packages available with the **ABRT** system.

Upon installation, the **abrt-d** daemon, which is the core of the **ABRT** crash-detecting service, is configured to start at boot time. You can use the following command to verify its current status:

```
~]$ systemctl is-active abrt-d.service
active
```

In order to discover as many software bugs as possible, administrators should configure **ABRT** to automatically send reports of application crashes to Red Hat. To enable the autoreporting feature, issue the following command as **root**:

```
~]# abrt-auto-reporting enabled
```

Additional Information on ABRT

- ✦ [Red Hat Enterprise Linux 7 System Administrator's Guide](#) — The *Automatic Bug Reporting Tool (ABRT)* chapter of the *Administrator's Guide* for Red Hat Enterprise Linux 7 contains detailed information on installing, configuring, and using the **ABRT** service.

Revision History

Revision 0.0-8	Fri Apr 3 2015	Milan Navrátil
Update of the Red Hat Enterprise Linux Server for ARM 7.1 Release Notes.		
Revision 0.0-7	Tue Feb 3 2015	Milan Navrátil
Release of the Red Hat Enterprise Linux Server for ARM 7.1 Release Notes.		
Revision 0.0-5	Mon Dec 8 2014	Milan Navrátil
Release of the Red Hat Enterprise Linux Server for ARM Development Preview 1.6 (snapshot 6) Release Notes		
* Hardw are Enablement: SBSA and SBBR standards compliance		
* Hardw are Enablement: Applied Micro (APM) X-Gene SoC and AMD Seattle SoC fully supported		
* Kernel: the kernel 3.17.3 included		
* Kernel: ACPI enabled by default		
Revision 0.0-4	Wed Oct 29 2014	Milan Navrátil
Release of the Red Hat Enterprise Linux Server for ARM Development Preview 1.5 (snapshot 5) Release Notes		
Revision 0.0-3	Tue Sep 30 2014	Milan Navrátil
Release of the Red Hat Enterprise Linux Server for ARM Development Preview 1.4 (snapshot 4) Release Notes		
Revision 0.0-2	Wed Aug 27 2014	Milan Navrátil
Release of the Red Hat Enterprise Linux Server for ARM Development Preview 1.3 (snapshot 3) Release Notes		
* Hardw are Enablement: 64-bit AMD Seattle supported		
* Kernel: the kernel version 3.16.0 included		
Revision 0.0-1	Wed Jul 30 2014	Milan Navrátil
Release of the Red Hat Enterprise Linux Server for ARM Development Preview 1.2 Release Notes		