

EVOLUTION

OF RHEL 8 DOCUMENTATION

Jaromír Hradílek

Manager, Content Services

A few words about me

Jaromír Hradílek *Manager, Content Services*

Joined Red Hat in 2010 as an Associate Technical Writer.

Contributed to documentation for RHEL 5, 6, and 7, Red Hat Developer Toolset, and Red Hat Software Collections.

Manage the RHEL documentation team since 2013.



...I also have two cats and too many guitars.

What will we talk about today

How we developed documentation for RHEL 7

What problems do we have with it

What problems do our customers have with it

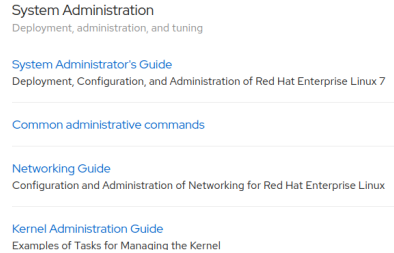
What we think went wrong

What did we do differently with RHEL 8

Why do we like it more

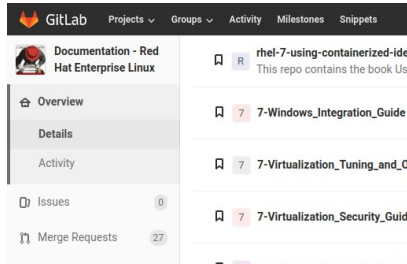
What new problems did we introduce

How is RHEL 7 documentation written



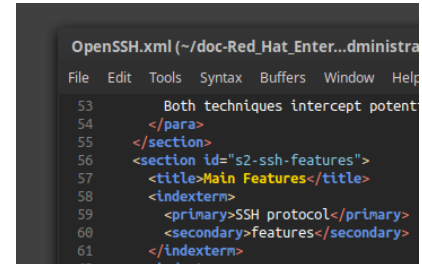
46 books and articles

46 unique books and articles titles categorized according to focus area with generic names like System Administrator's Guide, Networking Guide, Performance Tuning Guide, etc.



38 git repositories

38 individual git repositories with inconsistent names and unpredictable directory structure.



Written in DocBook 4.5

With the exception of two or three, all books written in DocBook 4.5, which is an XML language. Direct contributions from subject matter experts are very rare.

What does a typical RHEL 7 book look like: System Administrator's Guide

English
Multi-page HTML

- 12. OpenSSH
 - 12.1. The SSH Protocol
 - 12.1.1. Why Use SSH?
 - 12.1.2. Main Features
 - 12.1.3. Protocol Versions
 - 12.1.4. Event Sequence of an SSH Connection
 - 12.2. Configuring OpenSSH
 - 12.2.1. Configuration Files
 - 12.2.2. Starting an OpenSSH Server
 - 12.2.3. Requiring SSH for Remote Connections
 - 12.2.4. Using Key-based Authentication
 - 12.3. OpenSSH Clients
 - 12.3.1. Using the ssh Utility
 - 12.3.2. Using the scp Utility
 - 12.3.3. Using the sftp Utility
 - 12.4. More than a Secure Shell
 - 12.4.1. X11 Forwarding
 - 12.4.2. Port Forwarding

12.1.4. Event Sequence of an SSH Connection
The following series of events help protect the integrity of two hosts.

1. A cryptographic handshake is made so that the client communicating with the correct server.
2. The transport layer of the connection between the using a symmetric cipher.
3. The client authenticates itself to the server.
4. The client interacts with the remote host over the

12.1.4.1. Transport Layer

The primary role of the transport layer is to facilitate safe communication between the two hosts at the time of authentication and the transport layer accomplishes this by handling the encryption and by providing integrity protection of data packets as the transport layer also provides compression, speeding the

Once an SSH client contacts a server, key information is exchanged and the client can correctly construct the transport layer. The following

- Keys are exchanged
- The public key encryption algorithm is determined
- The symmetric encryption algorithm is determined
- The message authentication algorithm is determined
- The hash algorithm is determined

Focus on components

Chapters typically document one component, such as OpenSSH, Apache HTTP Server, or Systemd, and explain how to use it in isolation.

Teaching rather than guiding

Despite the fact that books have the word “guide” in their title, most of them teach how to use something.

Large, unfocused books

Some of the books have over 600 pages, are divided in 8 or more parts, and include 3 or more appendices.

What does a typical RHEL 7 book look like: System Administrator's Guide

```

en-US
├── Access_Control_Lists.xml
├── Accessibility.xml
├── Access_Labs.xml
├── Author_Group.xml
├── Automatic_Bug_Reporting_Tool_ABR_T.xml
├── Automating_System_Tasks.xml
├── Book_Info.xml
├── Cloud_Deployment_Planner.xml
├── Configuring_NTP_Using_ntpd.xml
├── Configuring_NTP_Using_the_chrony_Suite.xml
├── Configuring_PTP_Using_ptp4l.xml
├── Configuring_the_Date_and_Time.xml
├── Database_Servers.xml
├── Feedback.xml
├── File_and_Print_Servers.xml
├── FTP.xml
├── Gaining_Privileges.xml
├── Getting_Started.xml
├── icons
├── icon.svg
├── images
├── ABRT-config-events.png
├── ABRT-config-reporting.png
├── ABRT-suit.png
├── Accounts_n.png
├── Accounts_n.xcf
├── Accounts.png
├── Accounts.xcf
├── always_show_univ_access_menu.png
├── apache-mod_ssl-genkey-01.png
├── apache-mod_ssl-genkey-02.png
├── apache-mod_ssl-genkey-03.png
├── apache-mod_ssl-genkey-04.png
├── apache-mod_ssl-genkey-05.png
├── apache-mod_ssl-genkey-06.png
├── apache-mod_ssl-genkey-07.png
├── Cockpit_accounts.png
├── controlling-access-to-services-ntsysv.png
├── controlling-access-to-services-service_configuration.png
├── controlling-access-to-services-service_disabled.png
├── controlling-access-to-services-service_enabled.png
├── controlling-access-to-services-service_running.png
├── controlling-access-to-services-service_selected.png
├── controlling-access-to-services-service_stopped.png
├── controlling-access-to-services-service_unknown.png
├── controlling-access-to-services-service_wrong.png
├── date-and-time-authentication.png
├── date-and-time-date_and_time.png

```

Directory structure

44 XML files (1 unused)

120 images (59 unused)

Git repository

31 unique contributors in master (all writers)

2927 commits in master

63 remote branches

Source code

127 unique DocBook elements used

Uses both tabs and spaces for indentation

No reused content, but struggled with duplication

What problems we have with RHEL 7 documentation



Artificially isolated information

Information presented in one book cannot be **easily** reused elsewhere without duplication. Cross-product documentation is pretty much impossible.

Too much content

We have thousands of pages of content for a product that releases frequently. A lot of our content is rarely accessed.

DocBook XML

Training of new hires to become comfortable with DocBook is time consuming. Contributions from people outside of the documentation team are extremely rare.

But more importantly...

What do customers say about RHEL 7 documentation

“““

We have to spend a lot of time googling to understand how to use features.

“““

It's hard to find the procedure we're looking for.

“““

Too often, we have to reference multiple manuals to get something done.


Obligatory picture of the Airbus A380 cockpit





How was documentation for earlier releases written


Cluster Suite Overview
Cluster Suite Release Notes
Debugging with gdb
Desktop Deployment Guide
DM Multipath
GFS Release Notes
Global File System

RHEL 4

 25 unique books


 25 separate directories in an SVN repository with 100153 revisions


 DocBook XML built with Publican


 no categories


Getting Started	Install the client and register systems	
Installation Guide	Installing Red Hat Enterprise Linux 5 for all architectures	Avail
Deployment Guide	Deployment, configuration and administration of Red Hat Enterprise Linux 5	Avail
System Administration	Deployment, administration, and tuning	
Logical Volume Manager Administration	LVM Administrator's Guide	Avail
...

RHEL 5

 46 unique books


 45 separate directories in an SVN repository and 1 Git repository


 DocBook XML built with Publican


 3 categories


Getting Started	Install the client and register systems	
Common administrative commands		
Installation Guide	Installing Red Hat Enterprise Linux 6.9 for all architectures	Available
Migration Planning Guide	Migrating to Red Hat Enterprise Linux 6	Available
System Administration	Deployment, administration, and tuning	
Deployment Guide	Deployment, Configuration and Administration of Red Hat Enterprise Linux 6	Available

RHEL 6

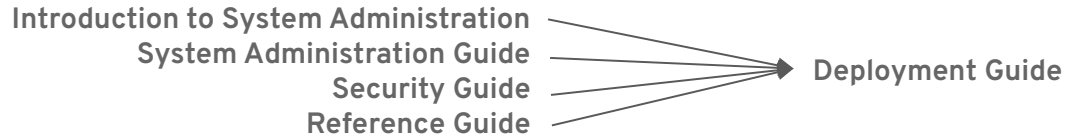
 52 unique books and 1 article

 52 separate directories in an SVN repository or individual Git repositories

 DocBook XML built with Publican

 9 categories

Why we approached documentation for Red Hat Enterprise Linux 7 this way?



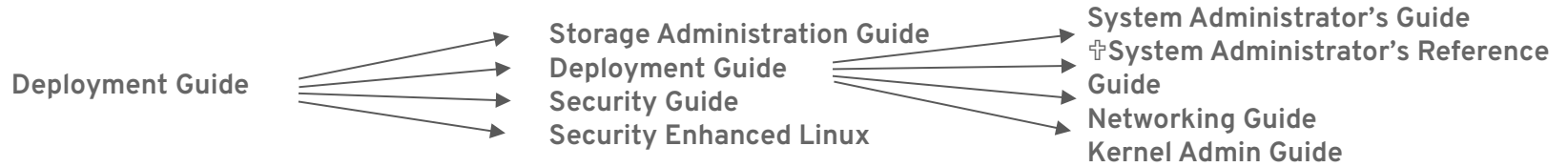
RHEL 4

📖 964 pages in total

RHEL 5

📖 872 pages in total

Why we approached documentation for Red Hat Enterprise Linux 7 this way?



RHEL 5

📖 872 pages in total

RHEL 6

📖 881 pages in the Deployment Guide

📖 1405 pages in total

RHEL 7

📖 592 pages in the System Administrator's Guide

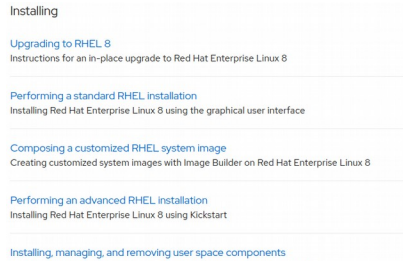
📖 933 pages in total

📖 The System Administrator's Guide got 139 pages of updated content before we decided to remove it

For Red Hat Enterprise Linux 8,
we changed everything...

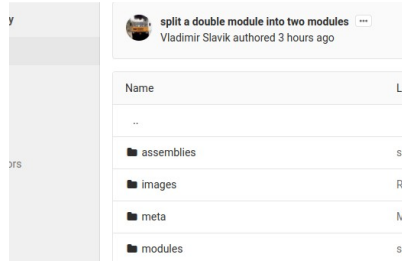
...and we started **from scratch.**

How is RHEL 8 documentation written



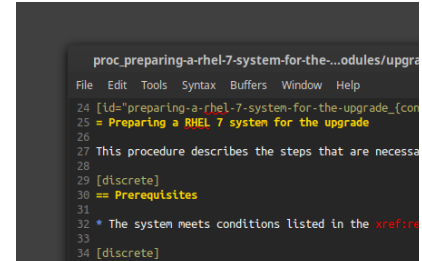
Documentation based on user stories

37 titles carefully crafted for discoverability, each written based on a user story with a specific target audience in mind.



Single git repository

A single git repository written in a modular fashion. Modules are stored separately from assemblies and titles and many are reused.

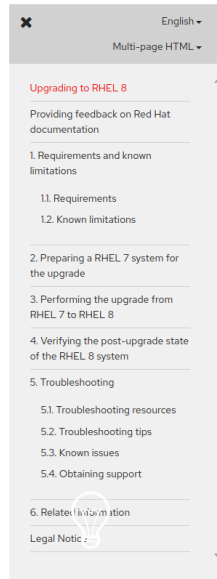


Written in AsciiDoc

All documentation is written in AsciiDoc, which is easy to understand and edit even without studying it. We receive many more patches from subject matter experts.

What does a typical RHEL 8 title look like:

Upgrading to RHEL 8



CHAPTER 2. PREPARING SYSTEM FOR THE UPGRADE

This procedure describes the steps that are necessary for upgrade to RHEL 8 using the **Leapp** utility.

Prerequisites

- The system meets conditions listed in [Chapter 1, Requirements](#).

Procedure

- Make sure your system has been successfully subscribed to Red Hat Enterprise Linux 8 using the Subscription Manager.
- Make sure you have the Red Hat Enterprise Linux 8 system:
 - Automatically assign the Red Hat Enterprise Linux 8 system:


```
# subscription-manager attach --auto
```
 - Verify that RHEL Server is subscribed:

```
# subscription-manager attach --auto
```

```
# subscription-manager list --installed
-----
Installed Product Status
-----
Product Name:   Red Hat Enterprise Linux
Product ID:     69
Version:        7.6
```

Based on a user story

Rather than describing a particular component, documentation gives you specific instructions for all tools that are needed to accomplish the reader's goal.

Task oriented documentation

Focus is on performing a particular task with a single validated method and clearly defined prerequisites. Explanations of concepts are limited to what is needed for understanding of the task and are tailored for a particular experience level.

Short, focused titles

Many titles have less than 50 pages of content, some have less than 20.

We also encourage direct feedback

5.1. Troubleshooting resources

Console output

By default, only error and critical log level messages are shown by the **Leapp** utility. To change the log level, use the `--verbose` option with the `leapp upgrade` command.

- In `verbose` mode, **Leapp** prints info, warning, error, and debug messages. A blue **Add Feedback** button is visible in the top right corner of the console output.

Logs

- The `/var/log/leapp/dnf-debugdata/` directory contains the logs. This directory is present only if **Leapp** is executed.

We use Direct Documentation Feedback

This feature is available to all subscribed customers for all RHEL 8 titles on the multi-page view.

How it works

Highlight text and click **Add Feedback** to add your own, or click **Feedback** in the top right corner to view existing comments.

Where does the feedback go

Every comment creates a bug for the documentation team.

What does our modular repository look like

README: How to contribute to this repository

How is this repository structured?

Simplified contents of the repository displayed in a tree-like format:

```

.
├── community
├── enterprise
│   ├── assemblies
│   │   ├── assembly-title-1.adoc
│   │   └── assembly-title-2.adoc
│   └── modules -> ../modules/
│   ├── images
│   │   ├── image-1.png
│   │   └── image-2.png
│   └── modules
│       ├── core-kernel
│       │   ├── module-title-1.adoc
│       │   └── module-title-2.adoc
│       ├── core-services
│       │   ├── module-title-1.adoc
│       │   └── module-title-2.adoc
│       └── desktop
│           ├── module-title-1.adoc
│           └── module-title-2.adoc
│           └── images -> ../images/
├── titles
└── README.adoc
  
```

Top-level directories are:

- scripts**: A directory for scripts that we develop to help us with the validation and maintenance.
- community**: A directory for upstream-specific assemblies and titles. It is currently empty.
- enterprise**: A directory that stores downstream-specific assemblies and titles. It is currently empty.

Directory structure

Separate directories for **modules**, **assemblies**, and **titles**. Modules are grouped based on how teams are organized (storage, desktop, kernel, ...), but assemblies and titles are not.

Git repository

46 unique contributors in master (including SMEs)
 4312 commits in master
 191 remote branches

Source code

Everything is written in AsciiDoctor
 Uses both tabs and spaces for indentation

We still have problems with RHEL 8 documentation



We need a library of modules

We can and do reuse content with ease, but we need a modular library to keep track of what we have and where. There still is a lot of room for accidental duplication. There's only so much we can do with a directory structure.

We need to improve our planning

Titles and assemblies use modules developed by different teams, but these teams still plan independently and are often unaware of mutual dependencies.


AsciiDoctor

AsciiDoctor is easier to contribute to, but new hires need absorb a lot. Inherited attribute definitions with `ifdef` conditions can get complex quickly.





Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 twitter.com/RedHat

