



Red Hat Hardware Certification 2024

红帽硬件认证测试套件用户指南

用于红帽硬件认证

用于红帽硬件认证

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, the Red Hat logo, JBoss, OpenShift, 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 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.

摘要

红帽硬件认证测试套件用户指南介绍了在 Red Hat Enterprise 软件上认证硬件所需的步骤。它概述整个认证流程，说明如何设置认证环境、测试已认证系统或组件，并将结果提交到红帽以进行验证。本指南还提供背景信息，包括测试方法和结果评估。版本 9.0 和 8.80 更新了 2024 年 5 月 28 日。

目录

使开源包含更多	5
第 1 章 红帽硬件认证计划简介	6
1.1. RED HAT 认证计划概述	6
1.2. 认证 workflow	6
1.3. 获取支持和提供反馈	6
第 2 章 合作伙伴加入	8
2.1. 创建红帽帐户	8
2.2. 加入硬件认证计划	8
第 3 章 使用 RED HAT CERTIFICATION TOOL 创建新的认证问题单	10
第 4 章 设置测试环境	11
4.1. 在测试下设置主机	11
4.2. 设置测试服务器	12
第 5 章 从红帽客户门户网站下载测试计划	14
第 6 章 使用 COCKPIT 配置系统并运行测试	15
6.1. 设置 COCKPIT 服务器	15
6.2. 将主机测试和测试服务器添加到 COCKPIT	15
6.3. 在 RED HAT SSO 网络中获取授权	16
6.4. 从红帽客户门户网站下载 COCKPIT 中的测试计划	16
6.5. 使用测试计划准备主机进行测试	17
6.6. 使用测试计划准备测试服务器	17
6.7. 使用 COCKPIT 运行认证测试	18
6.8. 检查并下载测试结果文件	18
6.9. 将测试结果从 COCKPIT 提交到红帽客户门户网站	19
6.10. 将测试结果文件上传到 RED HAT CERTIFICATION TOOL	19
第 7 章 使用 RHCERT CLI 工具配置系统并运行测试	20
7.1. 使用测试计划准备主机进行测试	20
7.2. 使用测试计划准备测试服务器	20
7.3. 使用 CLI 运行认证测试	21
7.4. 提交测试结果文件	21
第 8 章 认证 workflow	22
8.1. 在之前认证的硬件中添加认证	22
8.2. 在现有认证中更改功能或硬件	22
8.3. 使用现有规格文件创建系统直通证书	22
8.4. 创建并发布组件直通认证	23
8.5. 在产品认证中添加缺少的数据	24
8.6. 认证 64K 内核	25
8.7. 在测试执行过程中下载客户机镜像	25
第 9 章 层次产品认证	27
9.1. 认证层次产品	27
9.2. RED HAT ENTERPRISE LINUX FOR REAL-TIME	28
9.3. RED HAT VIRTUALIZATION	28
9.4. RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM COMPUTE	28
9.5. RED HAT OPENSTACK PLATFORM FOR REAL-TIME APPLICATIONS	29
9.6. RED HAT OPENSIFT CONTAINER PLATFORM	29

第 10 章 利用	31
10.1. 同一供应商的系统认证规则	31
10.2. 为不同供应商利用系统认证的规则	31
10.3. 生成用于利用系统认证的测试结果 ID	31
10.4. 利用现有组件	32
第 11 章 查看测试结果并完成认证	34
11.1. 红帽检查测试结果	34
11.2. 完成认证	34
附录 A. 测试	35
A.1. ACPI 密钥	35
A.2. AUDIO	36
A.3. BACKLIGHT	37
A.4. 电池	38
A.5. 蓝牙	39
A.6. BLURAY	39
A.7. CD ROM	41
A.8. CORE	42
A.9. CPU 扩展	42
A.10. DVD	44
A.11. ETHERNET	45
A.12. EXPRESSCARD	47
A.13. FINGERPRINTREADER	48
A.14. FIRMWARE	48
A.15. FV_CORE	49
A.16. FV_CPU_PINNING	50
A.17. FV_LIVE_MIGRATION	51
A.18. FV_MEMORY	52
A.19. FV_PCIE_STORAGE_PASSTHROUGH	52
A.20. FV_USB_NETWORK_PASSTHROUGH	53
A.21. FV_USB_STORAGE_PASSTHROUGH	54
A.22. FV_PCIE_NETWORK_PASSTHROUGH	55
A.23. INFINIBAND 连接	56
A.24. INTEL_SST	58
A.25. IPXE	59
A.26. IWARP 连接	60
A.27. KDUMP	63
A.28. LID	64
A.29. MEMORY	65
A.30. NETWORK	67
A.31. NETWORKMANAGEABLECHECK	72
A.32. 使用 FABRIC 测试的 NVME	73
A.33. OMNIPATH 连接	80
A.34. POWER_STOP	83
A.35. PROFILER	84
A.36. REALTIME	88
A.37. REBOOT	90
A.38. ROCE 连接	91
A.39. SATA	94
A.40. SATA_SSD	95
A.41. M2_SATA	96
A.42. U2_SATA	97

A.43. SAS	98
A.44. SAS_SSD	99
A.45. PCIE_NVME	100
A.46. M2_NVME	101
A.47. U2_NVME	102
A.48. NVDIMM	103
A.49. SR-IOV	105
A.50. STORAGE	107
A.51. 特殊键	109
A.52. SUPPORTABLE	110
A.53. SUSPEND	113
A.54. 磁带	115
A.55. THUNDERBOLT3	116
A.56. THUNDERBOLT4	117
A.57. USB_STORAGE	118
A.58. USB2	119
A.59. USB3	120
A.60. USB4	122
A.61. VIDEO	123
A.62. VIDEO_PORTS	125
A.63. VIDEO_DRM	128
A.64. VIDEO_DRM_3D	129
A.65. WIRELESSG	131
A.66. WIRELESSN	131
A.67. WIRELESSAC	132
A.68. WIRELESSAX (SUPERSED BY WIFI6)	133
A.69. WIFI6	133
A.70. WIFI6E	134
A.71. 手动添加并运行测试	134

使开源包含更多

红帽承诺替换我们的代码和文档中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于这一努力的精力，这些更改将在即将发布的版本中逐渐实施。[有关让我们的语言更加包含的更多详情，请参阅我们的CTO Chris Wright 信息。](#)

第1章 红帽硬件认证计划简介

使用本指南认证您公司的硬件产品，以运行一个或多个红帽产品。

1.1. RED HAT 认证计划概述

红帽认证计划确保了与 Red Hat Enterprise Linux、Red Hat OpenStack Platform、Red Hat Enterprise Linux for Real Time 和其他红帽软件产品在硬件平台上的红帽软件产品的兼容性。<https://connect.redhat.com/partner-with-us/hardware-certification-overview> 程序有三个主要元素：

- **测试套件**：完成硬件或软件应用程序测试。
- **红帽认证生态系统**：帮助探索和查找认证产品，包括硬件、软件、云和服务提供商。
- **支持**：与您与红帽之间的共同支持关系。

1.2. 认证 workflow

硬件认证包括测试服务器、桌面、工作站、笔记本电脑和个别组件，以运行 Red Hat Enterprise Linux、Red Hat OpenStack Platform Compute 和 Red Hat Enterprise Linux for Real Time。

先决条件

1. 与红帽建立认证关系。
2. 设置由合作伙伴的产品以及要认证的红帽产品组合组成的测试环境。
3. 进行初步测试以确保此组合正常工作。
4. 安装 redhat-certification 工具。

流程

1. 使用红帽认证工具为特定软件或硬件组件创建认证请求。
2. 红帽认证团队对硬件规格应用认证策略，以创建官方测试计划。测试计划为 RHEL 8 和 RHEL 9 认证系统或组件，包括根据已识别的组件及其提交给红帽的规格发布的测试和功能。
3. 运行官方测试计划中指定的测试，并通过红帽认证工具向红帽认证团队提交结果进行分析。<mailto:cert-ops@redhat.com>
4. 认证团队分析测试结果，并告知任何需要重新进行的测试。
5. 为红帽提供一个代表硬件示例，涵盖认证项目。
6. 当所有测试都有可比性的结果时，认证已完成，并通过红帽生态系统目录提供经过认证的产品。

其他资源

- 有关红帽硬件认证要求和策略的更多信息，请参阅 [红帽硬件认证政策指南](#)

1.3. 获取支持和提供反馈

具有专用支持资源、被分配的工程合作伙伴经理、工程客户经理或大客户经理的合作伙伴可使用它们用来请求对其他红帽产品提供支持的相同工具创建一个支持问题单。

没有专用支持资源的合作伙伴可在以下实例 [下使用红帽客户门户网站创建一个支持问题单](#)：

- 报告问题并获得认证过程的帮助
- 在认证工具集和文档中提交反馈和请求增强
- 要获取有关认证您的产品或应用程序的红帽产品的帮助。要接收红帽产品协助，必须拥有所需的产品权利和订阅，这些授权与认证特定权利和订阅分开

要使用红帽客户门户网站界面创建支持问题单，请完成以下步骤：

1. 使用红帽帐户凭证登录 [红帽客户门户网站](#)，这些凭证也用于访问其他红帽资产，如 [Red Hat Connect for Technology Partners](#) 和软件订阅。
2. 点红帽客户门户网站主页中的 **Open a Support Case**。
3. 特别注意以下字段，请完成 Support Case Form：
 - 在 **Product** 字段中，根据以下详情选择您产品/应用程序认证的红帽产品名称：
 - 对于 Red Hat OpenStack Platform 认证，请选择 **Red Hat OpenStack Platform**。
 - 对于经认证的云和服务提供程序(CCSP)认证，请选择 **Red Hat Enterprise Linux**。
 - 对于 Red Hat Container Certification，请选择 **Red Hat Enterprise Linux**。
 - 对于红帽硬件认证，请选择 **Red Hat Enterprise Linux**。
 - 在 **Product Version** 字段中，选择产品的版本。
 - 在 **Problem Statement** 字段中，使用以下格式输入问题语句/问题或反馈：
{partner Certification} (问题/咨询或反馈)

用认证流程 或红帽产品或文档的反馈替换（问题/ 反馈或反馈）。

For example: {Partner Certification} Error occurred while submitting certification test results using the Red Hat Certification application.

使用 [客户门户网站上如何打开和管理支持问题单的详细信息完成剩余的表格](#)？



注意

红帽建议您在开始认证过程前具有红帽认证工程师或具有同等经验。

第 2 章 合作伙伴加入

使用红帽客户门户网站创建新帐户并加入硬件认证计划。

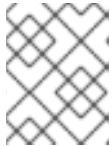
如果您在认证过程中遇到任何问题，可以通过以下方式联系我们以获取支持：

- 在 **产品认证** 类别下，创建 **合作伙伴加速台(PAD)** 票据。
- 将红帽认证操作(cert-ops)团队发送邮件至 cert-ops@redhat.com。
- 如果您已被分配，请联系您的专用生态系统合作伙伴管理(EPM)。

2.1. 创建红帽帐户

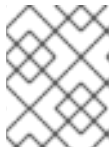
流程

1. 打开 [红帽客户门户网站](#)，点页面右上角的 **Register**。
此时会显示 **Register for a Red Hat account** 页面。



注意

请确定您使用您的公司电子邮件 ID 而不是您的个人电子邮件。您输入的电子邮件 ID 将用于您与红帽的所有通信。



注意

红帽建议使用唯一的登录 ID，它与电子邮件 ID 分开，以避免在以后出现与帐户相关的问题。创建后无法更改登录 ID。

2. 输入您的 **登录信息** 和 **个人信息**。
3. 选择 **Corporate** 作为帐户类型。
4. 输入您公司的 **联系信息**。
5. 单击 **Create My Account**。创建了一个新的红帽帐户。
6. 请求在同一帐户上启用红帽合作伙伴订阅(RHPS)。您必须有一个管理员机构（机构）。admins）发出请求的权限。
具体步骤请查看 [红帽合作伙伴订阅\(RHPS\)](#)。

验证

- 验证您的帐户号是否已分配给您的帐户。
为此，请登录 [红帽客户门户](#)，然后单击右上角的头像以确认帐户详细信息。

2.2. 加入硬件认证计划

流程

1. 登录到 [Red Hat Partner Connect 门户](#) 以加入硬件认证计划。
2. 点 **Accept Conditions and Conditions**。

3. 提供有关您公司和产品的一些详细信息，**然后单击提交。**
4. 点您在邮箱中接收的链接验证您的电子邮件地址。
5. 在 **Red Hat Conditions and Conditions** 页面中，选择所有 **我已阅读并同意条款** 复选框，然后单击 **Submit**。
上会显示一条成功成为硬件合作伙伴的消息。
6. 从 **How will use this subscription?** 下拉列表中选择一个选项，然后单击 **Request partner subscription**。
7. 在 **Red Hat Conditions and Conditions** 页面中，选择所有 **我已阅读并同意条款** 复选框，然后单击 **Submit**。
成功接收免费合作伙伴订阅后会显示一条消息。

创建 vendor 配置集，并且会自动添加到供应商用户中的单点登录(SSO)。



注意

创建后您的供应商个人资料将保留在红帽数据库中。但是，您的 Red Hat 合作伙伴订阅 (RHPS) 帐户在一年不活跃后就不活跃。因此，如果您是返回合作伙伴，请在开始硬件认证前请求重新激活 RHPS 帐户。

激活 RHPS 帐户后，cert-ops 团队会收到您的详细信息通知，包括分配给您的帐户的单点登录(SSO)信息。

后续步骤

[使用 Red Hat Certification Tool 创建新的认证问题单](#)

第 3 章 使用 RED HAT CERTIFICATION TOOL 创建新的认证问题单

作为合作伙伴，您可以在红帽认证工具中打开一个新认证案例，开始认证流程。完成此任务后，红帽会根据您提供的产品规格准备测试计划。

先决条件

- 您已与红帽建立了认证关系。
- 有用户登录凭证。
- 您有与用户登录关联的供应商和产品。

流程

1. 登录红帽客户门户 <https://rhcrt.connect.redhat.com/#/home>。
2. 在主页上，单击 **Open Certification**。
此时会打开 **Open a New Certification Case** 对话框。
3. 点 **Next**。
4. 从 **Partner and Product** 列表中选择一个选项。
如果没有显示您的产品，请在 **Product** 字段中输入其名称来创建它。然后，选择它。
5. **在此小节中**，选择适用于您的产品的复选框。
您的产品可能对多个生态系统具有资格。
6. 点 **Next**。
7. 输入 **Make**。
Model 会根据之前输入的信息显示。
8. 选中 **Which category best describes your product?** 中的适用复选框？
9. 可选：输入 **产品 URL**、**支持 URL** 和 **规格 URL**。
10. 点 **Next**。
根据您的输入，合作伙伴产品列表中 **会创建一个新产品**。
11. 从 **Red Hat Certification** 列表选择一个选项，然后点 **Next**。
12. 检查您提供的信息，**然后单击下一步**。

验证

如果您已成功为产品创建了新的认证问题单，则会显示一条信息。

后续步骤

当红帽为产品准备测试计划时，您可以设置 [测试环境](#)，以便为运行测试准备系统。

第 4 章 设置测试环境

设置可运行测试的测试环境。

测试环境至少由两个系统组成：

- 系统 1：在测试下作为主机执行(HUT)
- 系统 2：将 Acts 作为测试服务器

4.1. 在测试下设置主机

安装或配置需要认证的产品的系统称为测试下的主机(HUT)。

先决条件

- HUT 安装了 RHEL 版本 8 或 9。为方便起见，红帽提供了 [kickstart](#) 文件来安装 HUT 的操作系统。在启动安装过程前，请按照适合您的系统文件中的说明进行操作。



注意

红帽硬件认证需要使用 Red Hat Enterprise Linux 版本正式发行(GA)内核进行硬件认证。

安装 RHEL 时，请下载并使用 Binary DVD 离线安装镜像，而不是引导 ISO 镜像。Boot ISO 镜像需要一个网络连接，并将自动安装当前内核，而不是所需的 GA 内核。

不要在安装过程中使用 Red Hat Subscription Management (RHSM)注册系统。仅在安装过程完成后使用 RHSM 注册系统。

流程

1. 配置 [红帽认证](#) 存储库。
使用您的 RHN 凭证，使用 Red Hat Subscription Management 注册您的系统：

```
# subscription-manager register
```

2. 显示您的系统可用订阅列表：

```
# subscription-manager list --available*
```

3. 搜索提供红帽认证（适用于 RHEL 服务器）存储库的订阅，并记录订阅及其池 ID。
4. 将订阅附加到您的系统。将 pool_ID 替换为订阅的池 ID。

```
# subscription-manager attach --pool=<pool_ID>
```



注意

如果您为 Red Hat Subscription Management 启用选项 **Simple content access**，则不必将订阅附加到您的系统。如需了解更多详细信息，请[参阅如何为红帽订阅管理启用简单内容访问？](#)

5. 订阅红帽认证频道：

- 对于 RHEL 8:

```
# subscription-manager repos --enable=cert-1-for-rhel-8-_  
_<HOSTTYPE>_-rpms
```

将 HOSTTYPE 替换为系统架构。要查找系统架构，请运行

```
uname -m
```

Example:

```
# subscription-manager repos --enable=cert-1-for-rhel-8-x86_64-rpms
```

- 在 RHEL 9 中：

```
# subscription-manager repos --enable=cert-1-for-rhel-9-_  
_<HOSTTYPE>_-rpms
```

将 HOSTTYPE 替换为系统架构。要查找系统架构，请运行

```
uname -m
```

Example:

```
# subscription-manager repos --enable=cert-1-for-rhel-9-x86_64-rpms
```

6. 安装硬件测试套件软件包：

```
# yum install redhat-certification-hardware
```

4.2. 设置测试服务器

在 test (HUT) 下的主机上运行的一些测试需要第二个系统才能通过。第二个系统称为测试服务器。

例如，测试检查带宽，将数据从一个系统传输到另一个系统以便传递。

先决条件

- 测试服务器安装了 RHEL 版本 8 或 9。红帽提供了 [kickstart](#) 文件来安装 HUT 的操作系统，但您也可以使用它们来安装测试服务器。在启动安装过程前，请按照适合您的系统文件中的说明进行操作。

流程

1. 配置 *红帽认证* 存储库。
使用您的 RHN 凭证，使用 Red Hat Subscription Management 注册您的系统：

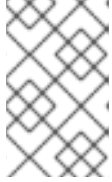
```
# subscription-manager register
```

2. 显示您的系统可用订阅列表：

```
# subscription-manager list --available*
```


- 3. 搜索提供红帽认证（适用于 RHEL 服务器）存储库的订阅，并记录订阅及其池 ID。
- 4. 将订阅附加到您的系统。将 pool_ID 替换为订阅的池 ID。

```
# subscription-manager attach --pool=<pool_ID>
```



注意

如果您为 Red Hat Subscription Management 启用选项 **Simple content access**，则不必将订阅附加到您的系统。如需了解更多详细信息，[请参阅如何为红帽订阅管理启用简单内容访问？](#)

- 5. 订阅红帽认证频道：

- 对于 RHEL 8:

```
# subscription-manager repos --enable=cert-1-for-rhel-8-<HOSTTYPE>-rpms
```

将 HOSTTYPE 替换为系统架构。要查找系统架构，请运行

```
uname -m
```

Example:

```
# subscription-manager repos --enable=cert-1-for-rhel-8-x86_64-rpms
```

- 在 RHEL 9 中：

```
# subscription-manager repos --enable=cert-1-for-rhel-9-<HOSTTYPE>-rpms
```

将 HOSTTYPE 替换为系统架构。要查找系统架构，请运行

```
uname -m
```

Example:

```
# subscription-manager repos --enable=cert-1-for-rhel-9-x86_64-rpms
```

- 6. 安装硬件测试套件软件包。

```
#yum install redhat-certification-hardware
```

后续步骤

请参阅[从红帽客户门户网站下载测试计划](#)。

第 5 章 从红帽客户门户网站下载测试计划

流程

1. 登录到 [红帽认证门户](#)。
2. 搜索与您的产品认证相关的问题单号，并复制它。
3. 点 **Cases** → 输入产品问题单号。
4. 可选：要列出测试运行期间测试的组件，请点 **Test Plans**。
5. 点 **Download Test Plan**。

后续步骤

如果您计划使用 Cockpit 运行测试，请参阅使用 [Cockpit 配置系统并运行测试](#)。

如果您计划使用 CLI 运行测试，请参阅使用 [CLI 配置系统并运行测试](#)。

第 6 章 使用 COCKPIT 配置系统并运行测试

要完成认证过程，您必须配置 cockpit，在 test (HUT)和测试服务器下准备主机，运行测试并检索测试结果。

6.1. 设置 COCKPIT 服务器

Cockpit 是一个 RHEL 工具，可让您更改系统配置，并从用户友好的 Web 界面监控其资源。



注意

- 您必须在一个新系统上设置 Cockpit，它与 test server 下的主机分开。
- 确保 Cockpit 能够访问 test 和 test server 下的主机。

有关安装和配置 Cockpit 的更多信息，请参阅在 RHEL 8 上使用 [RHEL web 控制台入门](#)，使用 RHEL 9 上的 [RHEL web 控制台](#)和 [Introducing Cockpit](#)。

前提条件

- Cockpit 服务器安装了 RHEL 版本 8 或 9。
- 您已在系统上安装了 Cockpit 插件。
- 您已启用了 Cockpit 服务。

流程

1. 登录到安装 Cockpit 的系统。
2. 安装由红帽认证团队提供的 Cockpit RPM。

```
# yum install redhat-certification-cockpit
```

您必须在端口 9090 上运行 Cockpit。

6.2. 将主机测试和测试服务器添加到 COCKPIT

将主机添加到 test (HUT)，并将测试服务器添加到 Cockpit 可让两个系统使用免密码 SSH 进行通信。

重复此步骤，为一个系统添加一个系统。

先决条件

- 您有 HUT 和测试服务器的 IP 地址或主机名。

流程

1. 在浏览器中输入 `http://<Cockpit_system_IP>:9090/` 来启动 Cockpit Web 应用程序。
2. 输入用户名和密码，然后点 Login。
3. 点 logged-in cockpit 用户名上的下箭头→Add new host。

此时将显示对话框。

4. 在 **Host** 字段中，输入系统的 IP 地址或主机名。
5. 在 **User name** 字段中输入您要分配给此系统的名称。
6. 可选：选择预定义的颜色，或为添加的主机选择一个新颜色。
7. 点 **Add**。
8. 点 **Accept key 并连接**，让 Cockpit 通过免密码 SSH 与系统通信。
9. 输入密码。
10. 选中 **Authorize SSH Key** 复选框。
11. 点 **登录**。

验证

在左侧面板中，点 **Tools → Red Hat Certification**，验证您刚刚添加的系统是否在右侧的 **Hosts** 部分下显示。

6.3. 在 RED HAT SSO 网络中获取授权

流程

1. 在浏览器的地址栏中输入 http://<Cockpit_system_IP>:9090/ 以启动 Cockpit Web 应用程序。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。
4. 在 Cockpit 主页上，单击 **Authorize**，以建立与红帽系统的连接。显示 **Log in to your Red Hat account** 页面。
5. 输入您的凭证并点 **Next**。
此时会显示对 **rhcert-cwe** 页面的授予访问权限。
6. 点 **Grant access**。确认消息显示成功设备登录。您现在已连接到 Cockpit Web 应用。

6.4. 从红帽客户门户下载 COCKPIT 中的测试计划

对于非授权或有限的访问用户：

- 要下载测试计划，请参阅[从红帽客户门户网站下载测试计划](#)。

对于授权用户：

流程

1. 在浏览器的地址栏中输入 http://<Cockpit_system_IP>:9090/ 以启动 Cockpit Web 应用程序。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。

4. 点 **Test Plans** 选项卡。将出现 **Recent Certification Support Cases** 列表。
5. 点 **Download Test Plan**。这时将显示一条消息，确认已成功添加测试计划。
6. 下载的计划将在 **Test Plan Files** 的 **File Name** 部分下列出。

6.5. 使用测试计划准备主机进行测试

在测试下置备主机执行很多操作，如设置与 cockpit 的免密码 SSH 通信、根据您的认证类型安装所需的软件包，以及创建最终测试计划来运行，这是从红帽提供的测试计划中进行的常用测试列表，并针对发现系统要求而生成的测试。

例如，如果测试计划旨在认证硬件产品，则安装所需的硬件软件包。

先决条件

- 您已下载红帽提供的测试计划。

流程

1. 在浏览器地址栏中输入 `http:// <Cockpit_system_IP > :9090/` 来启动 Cockpit Web 应用。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。
4. 单击 **Hosts** 选项卡，然后单击测试下要在其上运行测试的主机。
5. 单击 **Provision**。
此时会出现一个对话框。
 - a. 点 **Upload**，然后选择新的 test plan .xml 文件。然后，单击 **Next**。此时会显示成功上传信息。
另外，如果要重复使用之前上传的测试计划，请再次选择它来重新上传。



注意

在认证过程中，如果您收到持续产品认证的重新设计测试计划，您可以在上一步中上传它。但是，您必须在 **Terminal** 选项卡中运行 **rhcert-clean all**，然后才能继续。

- b. 在 **Role** 字段中，选择 **test** 下的 **Host**，再单击 **Submit**。
- c. 默认情况下，该文件上传到路径 `/var/rhcert/plans/<testplanfile.xml >`。

6.6. 使用测试计划准备测试服务器

运行 `Provision Host` 命令会启用并启动 `rhcertd` 服务，该服务在测试服务器上配置测试套件中指定的服务，如 `iperf` 以进行网络测试，以及 `kdump` 测试中使用的 `nfs` 挂载点。

先决条件

- 您已下载红帽提供的测试计划。

流程

1. 在浏览器地址栏中输入 `http:// <Cockpit_system_IP > :9090/` 来启动 Cockpit Web 应用。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。
4. 单击 **Hosts** 选项卡，然后单击测试下要在其上运行测试的主机。
5. 单击 **Provision**。
此时会出现一个对话框。
 - a. 点 **Upload**，然后选择新的 test plan .xml 文件。然后，单击 **Next**。此时会显示成功上传信息。
另外，如果要重复使用之前上传的测试计划，请再次选择它来重新上传。



注意

在认证过程中，如果您收到持续产品认证的重新设计测试计划，您可以在上一步中上传它。但是，您必须在 Terminal 选项卡中运行 `rhcert-clean all`，然后才能继续。

- b. 在 **Role** 字段中，选择 **Test server**，再单击 **Submit**。默认情况下，该文件上传到 `/var/rhcert/plans/<testplanfile.xml>` 路径。

6.7. 使用 COCKPIT 运行认证测试

先决条件

- 您已在测试下准备了主机。
- 您已准备了测试服务器。

流程

1. 在浏览器地址栏中输入 `http:// <Cockpit_system_IP > :9090/` 来启动 Cockpit Web 应用。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。
4. 单击 **Hosts** 选项卡，再单击要在其上运行测试的主机。
5. 点 **Terminal** 选项卡，然后选择 **Run**。
此时会显示基于测试计划上传的推荐测试列表。运行的最终测试计划是从红帽提供的测试计划中的常见测试列表，并在发现系统要求时生成的测试。
6. 出现提示时，选择是否通过键入 **yes** 或 **no** 运行每个测试。
您还可以通过键入 **select**，从列表中选择特定的测试。

6.8. 检查并下载测试结果文件

流程

1. 在浏览器地址栏中输入 `http://<Cockpit_system_IP>:9090/` 来启动 Cockpit Web 应用。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。
4. 点 **Result Files** 选项卡查看生成的测试结果。
 - a. 可选：点 **Preview** 查看每个测试结果。
 - b. 点结果文件旁的 **Download**。默认情况下，结果文件保存为 `/var/rhcert/save/hostname-date-time.xml`。

6.9. 将测试结果从 COCKPIT 提交到红帽客户门户网站

流程

1. 在浏览器的地址栏中输入 `http://<Cockpit_system_IP>:9090/` 以启动 Cockpit Web 应用程序。
2. 输入用户名和密码，然后点 **Login**。
3. 在左侧面板中选择 **Tools → Red Hat Certification**。
4. 点 **Result Files** 选项卡，然后从显示的列表中选择问题单号。
 - a. 对于授权用户，请单击 **Submit**。这时将显示一条消息，确认已成功上传测试结果文件。
 - b. 对于非授权用户，请参阅[将已执行测试计划的结果文件上传到红帽客户门户网站](#)。

已执行的测试计划的测试结果文件将上传到红帽认证门户网站中。

6.10. 将测试结果文件上传到 RED HAT CERTIFICATION TOOL

使用红帽认证工具向红帽认证团队提交所执行测试计划的测试结果文件。

先决条件

- 您已从 Cockpit 或 HUT 下载测试结果文件。

流程

1. 登录到 [红帽认证工具](#)。
2. 在主页上，在搜索栏中输入产品问题单号。
从显示的列表中选择问题单号。
3. 在 **Summary** 选项卡中，在 Files 部分下点 **Upload**。

后续步骤

红帽会审查您提交的结果文件并建议后续步骤。如需更多信息，请访问红帽 [认证工具](#)。

第 7 章 使用 RHCERT CLI 工具配置系统并运行测试

要使用 CLI 完成认证流程，您必须在测试(HUT)和测试服务器下准备主机，运行测试并检索测试结果。

7.1. 使用测试计划准备主机进行测试

运行 `provision` 命令执行很多操作，如设置与测试服务器的免密码 SSH 通信、根据您的认证类型安装所需的软件包，以及创建最终测试计划以运行的最终测试计划，这是从红帽提供的测试计划中进行的常用测试列表，并针对发现系统要求而生成的测试计划进行测试。

例如，如果测试计划旨在认证硬件产品，将安装所需的硬件软件包。

先决条件

- 您有测试服务器的主机名或 IP 地址。

流程

1. 以任何一种方式运行 `provision` 命令。测试计划将自动下载到您的系统。

- 如果您已经下载了测试计划：

```
# rhcert-provision <path_to_test_plan_document>
```

将 `<path_to_test_plan_document>` 替换为系统中保存的测试计划文件。

按照屏幕说明进行操作。

- 如果您还没有下载测试计划：

```
# rhcert-provision
```

按照屏幕说明操作，并在提示时输入您的 认证 ID。

2. 出现提示时，提供测试服务器的主机名或 IP 地址来设置免密码 SSH。只有在您第一次添加新系统时，才会提示您。

7.2. 使用测试计划准备测试服务器

运行 `Provision` 命令会启用并启动 `rhcertd` 服务，该服务配置测试服务器上的测试套件中指定的服务，如 `iperf` 用于网络测试，以及 `kdump` 测试中使用的 `nfs` 挂载点。

先决条件

- 在 `test` 下具有主机的主机名或 IP 地址。

流程

1. 通过在您要添加的系统中定义角色 "test server" 来运行 `provision` 命令。这只在置备测试服务器时才需要。

```
# rhcert-provision --role test-server <path_to_test_plan_document>
```


将 <path_to_test_plan_document> 替换为系统中保存的测试计划文件。

7.3. 使用 CLI 运行认证测试

流程

1. 运行以下命令:

```
# rhcert-run
```

2. 出现提示时, 选择是否通过键入 **yes** 或 **no** 运行每个测试。
您还可以通过键入 **select**, 从列表中选择特定的测试。



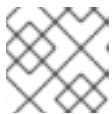
注意

测试重启后, **rhcert** 会在后台运行, 以验证镜像。使用 **tail -f /var/log/rhcert/RedHatCertDaemon.log** 查看当前验证的进度和状态。

7.4. 提交测试结果文件

流程

1. 登录以验证您的设备。



注意

要提交测试结果文件, 需要登录。

```
# rhcert-cli login
```

- a. 在新浏览器窗口或标签页中打开生成的 URL。
 - b. 输入登录名和密码, 并单击 **Log in**。
 - c. 点 **Grant access**。
显示设备日志成功信息。
 - d. 返回到终端, 然后在 **您授予访问权限后**输入 **yes** 给 **Please confirm**。
2. 提交结果文件。

```
# rhcert-submit
```

出现提示时, 输入您的认证 ID。

第 8 章 认证 workflow

8.1. 在之前认证的硬件中添加认证

使用此流程为已经完成了早期 RHEL 版本的硬件认证流程的系统或组件创建新的认证请求，或针对目前认证的系统或组件创建新的认证请求。

流程

1. 登录到 [红帽认证门户](#)。
2. 点 **New Certification**。
3. 选择红帽产品、版本和平台进行认证。然后，单击 **Next**。
4. 从下拉列表中选择供应商、生成和已认证产品的名称。然后，单击 **Next**。

创建请求后，监控审查团队中问题的请求，然后再创建官方测试计划。

8.2. 在现有认证中更改功能或硬件

应用补充认证，将硬件或功能添加到现有认证中。

您可以为之前未经过认证的功能请求补充认证，例如，因为它们没有测试，或者因为它们的测试失败。在其他功能经过认证后，红帽会将功能添加到认证目录中。

流程

1. 登录到 [红帽认证门户](#)。
2. 点现有硬件认证。
3. 点 **Product**。
4. 单击 **产品详细信息**。
5. 在 **Attachment** 字段中，点 **Browse** 为新的 supplemental 组件附加规格文件。选择 **就是一个规格**。
红帽认证团队会将补充组件添加到硬件认证中。
6. 创建补充认证。您不需要等待补充组件显示在现有硬件认证中。
 - a. 进入硬件认证页面。
 - b. 点 **Certification** 部分。
 - c. 点相关 **认证** 选项卡，再前往 **Supplemental Certification** 部分。
 - d. 单击 **New Certification** 以创建新的补充认证。
红帽认证团队将把测试计划添加到补充认证中。
7. 运行认证测试。您不需要等待新的测试计划。

8.3. 使用现有规格文件创建系统直通证书

系统直通认证创建了认证系统的副本，并在不同的供应商名称、不同的制作或不同的模型下将其列出。

当供应商将其系统销售给合作伙伴，或者供应商销售两个或更多个系统（其中一个系统是一个超集）时，会使用直通。

流程

1. 登录到 [红帽认证门户](#)。
2. 点现有硬件认证。
3. 点相关认证。
4. 点 **Add Related Certification**，然后选择 **Pass-through**。
5. 选择适当的产品：
 - 如果产品已创建好，请选择它。
 - 如果产品不在列表中，请将其创建为新产品。
6. 单击 **New Certification** 以创建新的直通认证。

红帽认证团队将审查硬件规格并发布新系统 .certification。发布新认证后，合作伙伴可以将其称为直通认证。

8.3.1. 将现有系统认证复制到新条目

流程

1. 要创建通过认证，请访问 [红帽认证 Web 用户界面](#)，请单击已认证的现有硬件系统认证。点 **Certification** 部分。在 **相关认证** 选项卡中，转至 **Pass through Certification** 部分，然后单击 **New Certification** 按钮。
2. 在 **Vendor** 字段中，选择您需要传递的产品的供应商。在 **Make** 字段中，选择您需要传递的 make。
3. 点 **Create** 按钮。这将生成一个请求，以创建通过系统规格进行传递，并通过对生成的规格的认证。

如果原始系统规格和直通系统规格相同，或者没有区别，则不需要额外的测试。如果发现不同，红帽认证团队将与您讨论，以便考虑它们。

8.3.2. 使用现有规格文件创建系统直通证书

流程

1. 进入 [红帽认证 Web 用户界面](#)，单击认证的现有硬件系统认证。点 **Certification** 部分。
2. 在 **相关认证** 选项卡中，进入 **Pass through Certification** 部分，再选择通过规范文件创建的传递。

这将使用相同的规格条目创建第二个直通证书。

8.4. 创建并发布组件直通认证

组件直通认证实质上创建了认证组件的副本，将其列出在不同的供应商名称、不同的 make 或不同的模型下。当系统供应商希望包含已经由组件厂商认证的组件时，使用这类直通。当组件厂商将其组件销售给他们撤销的第三方时，或者供应商销售了两个或更多个组件，则使用此类型。

流程

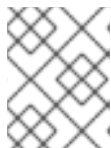
1. 创建系统认证。请参阅[使用红帽客户门户网站打开新的认证问题单](#)。
2. 选择 **Vendor, Make** 和 **Name**。单击**新建产品**按钮。这将使您选择**认证计划**网页。
3. 选择 **Vendor** 和 **Program** 作为 Hardware。点 **Next** 按钮。这将使您定义**红帽硬件认证供应商产品**网页。
4. 填写所有相关详情。从 **Category** 的下拉列表中，将类别选择为 **Component/Peripheral**。

这将创建组件认证。红帽认证团队认证并发布新创建的组件认证。在认证并发布证书后，其他合作伙伴就变得公开，以将其称为通过组件。

8.4.1. 将现有组件认证复制到新条目

流程

1. 要复制组件认证，请访问 **红帽认证** Web 用户界面，单击认证的现有硬件系统认证。点 **Certification** 部分。在 **相关认证** 选项卡中，进入 **Pass through Certification** 部分，然后单击 **New Certification** 按钮。
2. 在 **Vendor** 字段中，选择需要传递的产品的组件供应商。在 **Make** 字段中，选择需要传递的组件 Make。



注意

在这里，组件供应商和组件 Make 是创建并发布组件认证的步骤 1 到 4 时生成的字段。

如果原始组件规格和直通组件规格相同，则不需要额外的测试。如果发现不同，红帽认证团队将与您讨论，以便考虑它们。

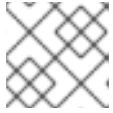
8.5. 在产品认证中添加缺少的数据

为确保准确和完整的认证信息，请按照此简化流程在发布认证前添加缺少的属性。

流程

1. 登录到 [红帽认证门户](#)。
2. 点现有硬件认证。
3. 在 **Certification Status** 部分中，点问号图标。 **Completion Requirements** notification banner 显示有关缺失的属性的信息。
4. 点一个缺少的属性，您将重定向到该认证的 **Properties** 选项卡。
5. 可选：点合作伙伴产品部分下的产品，并导航到 **属性** 选项卡。

6. 在 **Properties** 选项卡中，输入缺少的详细信息，如 **详细信息说明**、**短文描述**、**合作伙伴产品徽标** 或 **产品徽标**，以及 **系统类型**。



注意

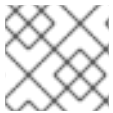
只有在选择了产品类别为 **System** 时，系统类型选项才适用。

7. 从 **System Types** 列表中选择一个或多个系统类型。
8. 在 **Enter url** 字段中为不同的系统类型添加营销 URL。
9. 点 **Update**。

验证

如果所有所需的数据都存在或更新，则问号图标将不再可见。

认证完成并发布后，更新的产品数据以及系统类型将在 [红帽生态系统目录](#) 上提供。



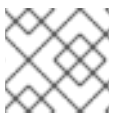
注意

所有标有星号 * 的字段是必需的，必须在发布认证前完成。

8.6. 认证 64K 内核

64k 页大小内核是 ARM 平台上大型数据集的一个有用的选项。它适用于内存密集型工作负载，因为它在整体系统性能方面显著提高，特别是在大型数据库、HPC 和高网络性能方面。

从 RHEL 9.2 开始，ARM 架构使用 64k 页大小内核作为可选，4k 内核作为默认值。要认证 64k 页大小内核，您需要首先使用默认的 4k 内核完成 RHEL 9 认证，然后您可以使用 64k 内核执行第二个认证。在完成第二个认证后，[知识库文章](#) 将附加到 4k 大小内核认证中，指示对 64k 页大小的支持，以及有关如何使用 64k 内核的说明。



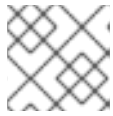
注意

您必须创建一个补充认证来认证 64k 内核。

8.7. 在测试执行过程中下载客户机镜像

流程

1. 验证 guest 镜像是否在系统上本地可用。如果是，则测试执行将开始。
2. 如果 guest 镜像在本地不可用，测试将尝试从预先配置的测试服务器下载它们。
3. 如果本地可用性和测试服务器下载失败，测试将与 CWE API 建立连接，以获取 AWS 的预签名 S3 URL。然后，使用提供的 URL 从 AWS 下载客户机镜像。
4. 如果从 AWS 下载遇到问题，测试将使用 CWE API 直接流并下载客户机镜像。
5. 如果以前尝试获取客户机镜像失败，则整个测试都会标记为 FAIL。



注意

以上流程适用于 rhcert 版本 8.66 及更高版本。

如果测试运行期间 FV 镜像下载失败，请按照以下步骤执行：

- a. 从红帽客户门户下载文件。
- b. 下载文件后，将它们移到 test 下的主机上的 `/var/lib/libvirt/images` 目录中。
- c. 要手动提取文件，请使用 `tar xmvfj <tarred file name>`。
- d. 提取文件后，使用命令 `mv <extracted file> <image file name>` 对其进行重命名。例如 - `mv hwcertData-20211116.img hwcertData.img`。
对于文件名，请参考下表：

打包文件名	镜像文件名
hwcertData.img.tar.bz2	hwcertData.img
hwcert-x86_64.img.tar.bz2	hwcert-x86_64.img
rhel-kvm-rt-image.qcow2.tar.bz2	rhel-kvm-rt-image.qcow2

第 9 章 层次产品认证

9.1. 认证层次产品

层次产品认证是指已针对 RHEL 认证的系统提供的附加认证。

您可以通过两种方式创建分层产品认证：

1. [第 9.1.1 节 “自动生成分层认证”](#)
2. [第 9.1.2 节 “手动创建分层认证”](#)

9.1.1. 自动生成分层认证

只有 **认证硬件** 认证时，才能自动生成分层认证。红帽认证团队使硬件认证 **公开** 在红帽客户门户上列出。

流程

执行以下步骤自动生成分层认证：

1. 登录到 [红帽认证门户](#)。
2. 单击 **必须经过认证并公开的硬件认证**。
3. 单击 **Dialog** 选项卡。
4. 在 **New Comment** 文本框中，输入红帽认证团队的评论以认证并公开认证。
5. 点 **添加注释** 按钮。

在为请求的硬件认证添加评论后，红帽认证团队会认证并公开证书。认证经过认证后，您将收到一封电子邮件。

您可以在红帽客户门户网站中看到新的自动创建的层次 **认证**。如果没有，请单击 **Refresh** 按钮，并且应下载新的认证。

9.1.2. 手动创建分层认证

如果具有分类为常规的分类，则可以为硬件认证手动创建分层认证。



注意

- 所有常规硬件认证都不支持分层认证。
- 在发布分层认证前，应发布基本认证。

流程

执行以下步骤手动创建分层认证：

1. 在 [红帽认证门户](#) 上，单击红帽认证团队认证的常规硬件认证。
2. 点相关 **认证** 选项卡。

3. 单击 **Add Related Certification**，然后从 **认证类型** 中选择 **Layered**。
4. 从 **Red Hat Certification** 和 **Product Version** 下拉菜单中选择 **认证类型**，然后点 **Next**。
5. 查看认证信息并点 **Open**。

在红帽认证门户中会创建一个新的 [认证](#)，您会被重定向到新创建的认证页面。

9.2. RED HAT ENTERPRISE LINUX FOR REAL-TIME

支持的 RHEL 版本	支持的 RHEL for Real Time 版本	预备认证
7、8 和 9	7、8 和 9	RHEL

在您的系统获得上表中详述的前提条件认证后，请申请 Red Hat Enterprise Linux for Real Time 认证。

红帽硬件认证测试套件包括获取 Real Time 认证所需的测试。

9.2.1. 其他资源

- [Red Hat Enterprise Linux for Real Time 生命周期](#)

9.3. RED HAT VIRTUALIZATION

支持的 RHEL 版本	支持的 RHEL for Real Time 版本	预备认证
7 和 8	7 和 8	RHEL

Red Hat Virtualization 依赖于 Red Hat Virtualization，并与 RHEL 联合设计。因此，如果您的系统在 Red Hat Enterprise Linux 认证过程中通过基本的虚拟化测试，则不需要运行额外的测试来接收 Red Hat Virtualization 认证。如果您的系统也通过高级虚拟化测试，您的 Red Hat Virtualization 认证将自动包含高级功能。

红帽将代表您对所有 64 位 Intel 和 AMD 自动打开 Red Hat Virtualization 认证，以及为 RHEL 提交的所有 IBM Power little endian 服务器认证。

如果您希望但没有自动接收认证，请为 Red Hat Virtualization 产品创建一个新的层次认证请求。

9.3.1. 其他资源

- [Red Hat Virtualization](#)
- [Red Hat Virtualization 生命周期](#)

9.4. RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM COMPUTE

支持的 RHEL 版本	支持的 RHEL for Real Time 版本	预备认证
-------------	---------------------------	------

支持的 RHEL 版本	支持的 RHEL for Real Time 版本	预备认证
8 和 9	请参阅支持性列表	RHEL（基本虚拟化测试必须通过）

Red Hat OpenStack Platform 依赖于 RHEL，并与 RHEL 联合设计。因此，如果您的系统满足上表中详述的前提条件认证，则不需要运行额外的测试来接收 Red Hat OpenStack Platform Compute 认证。您不需要打开新的认证。

但是，对于 Red Hat OpenStack Platform RHEL 8，分层产品认证专门适用于基于 LPAR（逻辑分区）的 POWER 系统。要在 RHEL 8 上获得分层产品认证，该系统必须在测试过程中作为裸机管理程序运行。

对于 RHEL 9，红帽不支持 POWER 系统上的 KVM。因此，没有创建基于虚拟化的层次产品认证。

对于其他架构或类别，或者在您想要但不自动收到认证的情况下，为 Red Hat OpenStack Platform Compute 产品创建一个新的分层认证请求。

红帽建议合作伙伴使用基板管理控制器(BMC)认证系统，以适用于 Red Hat OpenStack Platform Bare Metal 认证。

其他资源

- [Red Hat OpenStack Platform](#)
- [Red Hat OpenStack Platform 生命周期](#)
- [Red Hat OpenStack Platform 硬件裸机认证策略指南](#)
- [红帽裸机硬件认证 workflow 指南](#)

9.5. RED HAT OPENSTACK PLATFORM FOR REAL-TIME APPLICATIONS

支持的 RHEL 版本	支持的 RHEL for Real Time 版本	预备认证
8	请参阅支持性列表	<ul style="list-style-type: none"> • RHEL（基本虚拟化测试必须通过） • Red Hat Enterprise Linux for Real Time • Red Hat OpenStack Platform Compute

在对 Red Hat OpenStack Platform for Real-Time Applications 认证进行应用后，已获得上表中详述的前提条件认证。红帽硬件认证测试套件包括获取 Real Time 认证所需的测试。

9.6. RED HAT OPENSIFT CONTAINER PLATFORM

支持的 RHEL 版本	支持的 RHEL for Real Time 版本	预备认证
9.2、9.3 和 9.4 (RHOCP 4.13、4.14 或 4.15)	请参阅支持性列表	RHEL (基本虚拟化测试必须通过)
8 (RHOCP 4.12)	请参阅支持性列表	RHEL (基本虚拟化测试必须通过)

Red Hat OpenShift Container Platform 依赖于 RHEL，并与 RHEL 联合设计。因此，如果您的系统满足上表中详述的前提条件认证，则不需要运行额外的测试来接收 Red Hat OpenShift Container Platform 认证。

但是，对于 Red Hat OpenShift Container Platform，基于 RHEL 8 支持的版本（当前 v4.11 和 v4.12），分层产品认证特别适用于基于 LPAR（逻辑分区）的 POWER 系统。要在 RHEL 8 上获得分层产品认证，该系统必须在测试过程中作为裸机管理程序运行。

对于 RHEL 9，红帽不支持 POWER 系统上的 KVM。因此，没有创建基于虚拟化的层次产品认证。

对于其他架构或类别，或者在您想要但不自动收到认证的情况下，为 Red Hat OpenShift Container Platform 产品创建一个新的分层认证请求。

应用 Red Hat OpenShift Platform Bare Metal 认证，将 IPI 和安装程序功能添加到目录中的 RHOCP 条目。

其他资源

- [Red Hat OpenShift Container Platform](#)
- [Red Hat OpenShift Life Cycle](#)
- [Red Hat OpenStack Platform 硬件裸机认证策略指南](#)
- [红帽裸机硬件认证 workflow 指南](#)

第 10 章 利用

利用是重复使用认证的系统中的硬件的测试结果，以便在新的认证请求中涵盖对相同硬件的测试。它只能用于某些可选项目，这些项目必须相同。您无法将测试结果用于旧模型中的测试结果，无论它们的相似程度，项目都必须完全匹配。另外，利用只能用于测试 **您的组织或其代理** 是否已执行。

10.1. 同一供应商的系统认证规则

以下是在同一供应商处利用系统或组件认证时应小心的准则：

1. 组件必须相同。
2. 生成的结果必须来自相同架构的硬件。
3. 系统利用组件结果必须认证相同的主要版本。
4. 无法执行跨供应商利用系统认证。

例如，

- ACME Computers 可以利用从任何组件传递测试结果，以涵盖另一个 Acme 系统的同一项

但是，

- ACME Computers 无法引用 Cloverleaf Industries 执行的认证的测试结果

10.2. 为不同供应商利用系统认证的规则

1. 在组件制造商使用厂商、制作和模型信息（由经销商销售）创建原始认证时，应考虑以下准则：
 - i. 在 **Advanced** 选项卡中，选择 **Create using Pass-Through of the original Certification**，就像系统通过认证一样
 - ii. 如果很多经销商都使用相同的组件，则组件制造商应为每个经销商创建一个直通
 - iii. 如果经销商为同一卡使用多个名称，则组件制造商应为每个名称创建一个直通
2. 在红帽认证团队确认所使用的硬件相同，并使用规格文件文档完成直通认证后，认证将发布或未发布。
3. 经销商应在包含此硬件的系统认证请求中的相应 **Leverage** 字段使用认证 ID。

10.3. 生成用于利用系统认证的测试结果 ID

以下是生成利用系统认证的测试结果 ID 的步骤。

流程

1. 从红帽认证的 Web 用户界面创建源硬件产品和认证。
2. 要将组件添加到新创建的认证中，从红帽认证 Web 用户界面点认证的硬件证书。点 **Product** 部分，然后点 **Product Details** 选项卡。
3. 在 **Attachments** 部分中，单击 **Choose File** 按钮，以上传规范文件。规范文件由需要添加的组件组成。

4. 选择 **是一个规格** 复选框，并在 **Attachment Description** 文本框中添加简短的备注。例如："这是 spec.file"。
5. 在 **Red Hat Certification** Web 用户界面中点认证的硬件证书。点 **Certification** 部分。在 **Progress** 选项卡中，您会看到根据组件生成测试计划。
红帽认证团队审查并添加规范文件中提到的组件，稍后为添加的组件创建测试计划。
6. 单击 **Run** 按钮，以运行表格中显示的组件的测试。这将使您进入 **测试** 选项卡。
7. 单击 **Add Test System**。这将带您 **选择主机** 网页。
8. 选择您要运行测试的主机，然后单击 **Test** 按钮。
9. 在测试选项卡中，单击 **Continue Testing** 按钮，这将生成组件列表。
10. 选择您要运行测试的组件，再单击 **Run Selected** 按钮。
11. 测试运行完成后，您将收到消息 **Finished test run**。
12. 单击测试，运行测试的组件将结果为 **PASS**。要将测试结果提交到红帽认证团队，请从 **Actions** 字段中选择 **Submit**。这将把您带到 **提交的文件** 网页。
13. 单击 **Submit** 按钮。
红帽认证团队批准测试结果。批准的 **测试结果会生成与组件关联的测试结果 id**。
14. 在 **Red Hat Certification** Web 用户界面中，点认证的硬件证书。点 **Certification** 部分。在 **Progress** 选项卡中，您将看到 **Test Plan Credit** 为 **Confirmed**。**Test Result** 列将显示生成的 **Test Result ID**。

10.4. 利用现有组件

如果要使用同一组件创建新认证，您可以以两种方式使用该组件：

流程

1. 通过复制结果 ID 来利用

请参阅 [生成测试结果 ID，以便从系统认证中利用](#)。

Red Hat 认证团队批准了规格文件中提到的组件。

- a. 在 **Red Hat Certification** Web 用户界面中，点认证的硬件证书，并利用其测试结果 ID。点 **Certification** 部分。在 **Progress** 选项卡中，进入 **Test Result** 列，该列显示您将利用的组件生成的 **Test Result ID**。选择 ID 来复制测试结果 id。
- b. 如果 ID 成功复制，您将收到一个 "Copied Leverage Information from System Certification<the_component_name>" 信息。
- c. 在 **Red Hat Certification** Web 用户界面中，点您要添加 leverage 组件的硬件证书。
- d. 点 **Certification** 部分。在 **Progress** 选项卡中，进入 **Test Result** 列，点 **Test Result ID** 以应用复制的 Test Result ID。
如果成功应用了利用 ID，您将收到消息 "利用 System Test 已成功应用"。

2. 使用结果 ID 或认证 ID 进行利用

- a. 点 **Certification** 部分。在 **Progress** 选项卡中，进入 **Test Result ID** 列，再点 **Leverage Result**。
- b. 在 **Leverage Result** 窗口中，从 **Leverage Using** 下拉菜单中选择 **Result ID** 或 **认证 ID**。

**注意**

使用 **Result ID** 使用测试结果 ID，然后选择 **认证 ID** 以使用直通证书利用。

- c. 输入 **Leverage ID** 并点 **Submit**。
如果 Leveraging 成功，您将收到成功消息，如果提交 ID 出错，则会显示其他失败消息。

第 11 章 查看测试结果并完成认证

11.1. 红帽检查测试结果

提交结果后，审查团队将分析他们的内容和奖分，并为每个属于测试计划的测试而获得奖分。

当他们验证每个通过测试时，团队会在认证的站点测试计划中将每个测试计划项目设置为 **Confirmed**，您可以在目录的 **Results** 选项卡下看到。这样，您可以看到哪些测试是未完成的，并已在通过过程中验证了哪些测试。

如果发现任何问题，审阅团队将把认证请求更新为一个问题，该请求将自动通过电子邮件发送给提交证书的人员。

您可以在认证的 **Dialog** 选项卡上查看所有讨论，并响应或询问任何问题。

11.2. 完成认证

在红帽确认测试计划中的所有项目都已通过后，认证已完成。此时，您可以选择是否关闭并发布认证或关闭认证，并使认证未发布。

补充认证始终保持未发布。如果您不想公告认证状态或系统或组件是否存在，可以取消发布系统和组件认证。

在发布的认证中，您和红帽评论团队之间的系统信息和讨论将对公众不可见。

如果这些发布选项不符合您的要求，请在认证打开时提交一个例外请求，如果已经关闭了问题单。

附录 A. 测试

在本节中，我们提供有关硬件认证的每个测试的更多详细信息。每个 test 部分都使用以下格式：

测试涵盖了什么

本节列出了此特定测试运行的硬件类型。

支持的 RHEL 版本

本节列出了测试支持的 RHEL 版本。

测试的作用

本节介绍测试脚本的作用。请记住，所有测试都是 python 脚本，您可以在目录 `/usr/lib/python2.7/site-packages/rhcert/suites/hwcert/tests` 中查看，如果您想准确了解在测试中执行哪些命令。

准备测试

本节讨论为测试准备所需的步骤。例如，它讨论了将 USB 设备手动用于 USB 测试，使用空白磁盘进行重写光盘。

执行测试

本节标识测试是交互式还是非互动，并解释了运行测试所需的命令。

您可以选择运行测试的方法：

- 按照 [使用 CLI 运行认证测试进行操作，以运行测试](#)。使用以下命令，从显示的列表中选择合适的测试名称：

```
rhcert-run
```

- 如果在规划过程中出现硬件检测问题或其他与硬件相关的问题，[请继续手动添加并运行测试](#)。通过指定所需的测试名称来运行 `rhcert-cli` 命令。

```
rhcert-cli run --test=<test name>
```

运行时

本节介绍运行此测试所需的时间。每个部分都提到了 **可支持** 测试的时间信息，因为它是每次运行测试套件的必需测试。

A.1. ACPI 密钥

测试涵盖了什么

ACPI 键 测试从系统集成键盘捕获各种输入事件。

支持的 RHEL 版本

- RHEL 8.6 及更新版本
- RHEL 9

测试的作用

测试捕获以下内容：

- 与 ACPI 相关的信号，如 power、suspend 和 sleep。
- 按键发送与全局键盘快捷键（如 `<Meta+E>`）关联的信号，这将打开文件浏览器。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 **ACPI 密钥** 测试名称。

```
rhcert-run
```

此测试需要捕获所有输入事件。在测试过程中，按设备上的所有非标准和多媒体键。

随时按 **Escape** 键以结束测试并看到一个键列表。如果测试的所有键都出现在列表中，则测试可以成功。

运行时

完成测试需要不到 5 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.2. AUDIO

测试涵盖了什么

可移动介质和集成的声音设备通过 **音频** 测试进行测试。当硬件检测例程在 udev 数据库中找到以下字符串时，会调度测试：

```
E: SUBSYSTEM=sound  
E: SOUND_INITIALIZED=1
```

您可以在命令 `udevadm info --export-db` 中看到这些字符串，以及触发本指南中其他测试调度的字符串。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试在同时将其记录到文件中时扮演了预先录制的声音(guitar chords 或记录的语音)，那么它将回放记录，并询问您是否听到听起来。

准备测试

在开始测试运行前，您应该确保调度音频测试，并且系统可以播放并记录声音。如果测试没有出现在安装音频设备的系统中，请联络您的支持联系人。如果正确调度测试，请继续了解如何手动测试声音设备的回放和记录功能。

使用内置发言人或 speakers/headphones 插入到 headphone/line-out jack 中，可以通过以下方法进行测试前确认回放：

1. 在 **Settings** 应用程序中点 **Sound** 选项。

2. 点 **Output** 选项卡，选择您要测试的声音卡，并将 **Output 卷** 调整为适当的级别。
3. 单击 **Test jpegs**。
4. 在 **showing Testing** 弹出窗口中，单击 **Test** 按钮来生成声音。

如果无法听取声音，请确保将发言人插入到正确的端口。您可以使用任何行外或头手机 jack（我们无需使用哪个端口）。验证声音没有静音，并尝试在 speakers 和操作系统本身中调整卷。

如果音频设备具有记录功能，则还应在尝试运行测试前测试这些功能。将微手机插入系统上的行或 Mic jacks 之一，如果您测试笔记本，您可以使用内置的微手机。同样，我们不要求您使用特定的输入 jack；前提是，如果测试可以正常工作，测试将会通过。

1. 在 **Settings** 应用程序中点 **Sound** 选项。
2. 单击 **Input** 选项卡，选择适当的输入设备，并将 **输入卷** 调整为 100%。
3. 对话、tap 或者激活输入设备，并观察 **输入级别** 图形。如果您看到它移动，则正确设置输入设备。如果没有移动，请尝试另一个输入选择或微手机端口来插入输入设备。

如果您无法听到听起来或看到输入级别显示移动，请联络您的支持人员，因为这会导致音频测试失败。如果您能够成功播放声音并看到在微手机附近的声音时显示输入级别的移动，请继续下一小节以了解如何运行测试。

执行测试

音频测试是交互式的。在执行包含音频测试的测试运行前，请连接用于手动测试的微手机，并将其放置在发言人的前面，或者确保内置的微手机是没有模糊的。或者，如果您在 noisy 环境中测试，您可以将行外 jack 直接连接到 mic/line-in jack。

运行以下命令，然后从显示的列表中选择 **适当的** Audio 测试名称。

```
rhcert-run
```

互动步骤如下：

1. 系统将做声音并询问您是否听到了它们。根据情况回答 *y* 或 *n*。如果您决定在输出和输入之间使用直接连接，而不是 speakers 和一个微手机，则需要为回答选择 *y*，因为您的发言人将被跳线绕过。
2. 系统将下一个 play 返回它所记录的文件。如果您听到声音，则在提示时回答 *y*。否则，回答 *n*。

运行时

对于同时回放和记录，音频测试需要不到 1 分钟，然后是记录的声音的回放。所需的 **可支持** 测试将大约一分钟时间添加到整个运行时。

A.3. BACKLIGHT

测试涵盖了什么

当检测到附加显示的系统中并支持软件后端控制时，后端测试会运行。

支持的 RHEL 版本

- RHEL 8

- RHEL 9

测试的作用

测试通过将附加显示的显示亮度调整为最小值，确保后端控制可按预期工作。

准备测试

- 确保 test 下的主机正在运行 RHEL 8.0 或更高版本。
- 确保系统有后端支持，并且显示已连接至系统。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 **backlight** 测试名称。

```
rhcert-run
```

显示亮度将从最大值改为 minimum 和 back。测试将在确认显示亮度改变后通过。

运行时

运行此测试需要不到一分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.4. 电池

测试涵盖了什么

电池测试有效，只能在具有内置电池的系统中运行。在不向系统提供主要、内部电源的外部电池上不支持测试，如 UPS 或 BIOS 电池。当硬件检测例程在 **udev** 数据库中找到以下字符串时，会调度测试：

```
POWER_SUPPLY_TYPE=Battery
```

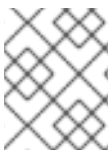
支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试会检测电池是否已连接，AC 适配器将插入到系统中，以及电池的状态。

准备测试



注意

当电池达到 100% 费用时，请勿执行测试。在执行测试前，将电池解放到较低级别，以避免潜在的测试失败。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 电池 测试名称。

```
rhcert-run
```

测试检测到 10 mWh 电池收费和取消收费，显示当前容量，以及电池的状态。根据屏幕的说明，在提示时拔出并插入 AC 适配器。

运行时

测试的执行时间取决于电池和计费速度。由于此测试在笔记本电脑上运行，因此所需的可支持测试将其挂起测试运行。总体而言，它需要 7-10 分钟。

A.5. 蓝牙

测试涵盖了什么

带有 bluetooth v3、v4 或 v5 控制器的系统支持 bluetooth 测试。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

测试使用 `rftkill` 命令检查系统中 bluetooth 控制器的可用性。然后，它运行 `hciconfig` 或 `btmgmt` 命令以获取 bluetooth 控制器版本。之后，测试会使用 `bluetoothctl` 命令工具验证控制器是否可以扫描、发现、通过选择并信任另一个选择并校准 bluetooth v3、v4 或 v5 设备。如果 HUT 有多个 bluetooth 控制器，则每个 bluetooth 控制器会自动计划 bluetooth 测试。

准备测试

在开始测试前，请确保：

- 具有支持与控制器相同的或更新版本的设备。
- 在 HUT 和对设备上启用蓝牙。
- 使用设置应用程序手动对设备进行配对，以确认连接。
- 对设备解除对。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 bluetooth 测试名称。

```
rhcert-run
```

然后，选择要测试 bluetooth 功能的设备。

运行时

完成测试需要大约 5 分钟。但是，时间可能会因 bluetooth 网络连接而异。

A.6. BLURAY

测试涵盖了什么

Bluray 测试在以下介质和相关驱动器类型中运行：

- 只读介质和驱动器(BD-ROM)
- 写出介质和驱动器(BD-R)
- 重写介质和驱动器(BD-RE)

根据 `udev` 命令的信息，测试套件决定要调度哪些光驱测试(Blu-ray、DVD 或者 CD-ROM)以及要测试的介质类型（只读、可写和重写）。例如，测试套件将针对带有可读取 DVD 和 CD-ROM 磁盘功能的 Blu-ray 驱动器规划以下测试：

- Blu-ray 介质的重写（删除、写入和读取）测试
- DVD 介质的读测试
- CD-ROM 介质的读测试

您只需要为给定驱动器运行 Bluray 测试。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试会根据驱动器的功能执行以下任务：

- 只读驱动器 - 首先，它会从磁盘读取数据并将其复制到硬盘。然后，它会将磁盘中的数据与硬盘上的副本进行比较。如果所有文件校验和都匹配，则测试会通过。
- 具有写入功能的驱动器 - 首先，它会从硬盘读取数据并将其写入可写空白磁盘。然后，它会将硬盘中的数据与磁盘中的副本进行比较。如果所有文件校验和都匹配，则测试会通过。
- 具有重写功能的驱动器 - 首先，它会从重写磁盘中删除所有信息。然后，它会从硬盘读取数据并将其写入重写磁盘。最后，它会将硬盘上的数据与磁盘的副本进行比较。如果删除操作成功，且所有文件校验和都匹配，则测试会通过。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 Bluray 测试名称。

```
rhcet-run
```

按照屏幕上的说明插入相应的介质，并在适当时关闭驱动器的跟踪。

运行时

Bluray 测试的运行时间取决于介质和驱动器的速度。对于 2x 25G BD-RE 磁盘，测试将在大约 14 分钟内完成。

A.7. CD ROM

测试涵盖了什么

CD ROM 测试在以下介质和相关驱动器类型中运行：

- 只读介质和驱动器(CD-ROM)
- 写出介质和驱动器(CD-R)
- 重写介质和驱动器(CD-RW)

根据 `udev` 命令的信息，测试套件决定要调度哪些光驱测试(Blu-ray、DVD 或者 CD-ROM)以及要测试的介质类型（只读、可写和重写）。例如，测试套件将针对带有可读取 DVD 和 CD-ROM 磁盘功能的 Blu-ray 驱动器规划以下测试：

- Blu-ray 介质的重写（删除、写入和读取）测试
- DVD 介质的读测试
- CD-ROM 介质的读测试

您只需要为给定驱动器运行 CD ROM 测试一次。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试会根据驱动器的功能执行以下任务：

- 只读驱动器 - 首先，它会从磁盘读取数据并将其复制到硬盘。然后，它会将磁盘中的数据与硬盘上的副本进行比较。如果所有文件校验和都匹配，则测试会通过。
- 具有写入功能的驱动器 - 首先，它会从硬盘读取数据并将其写入可写空白磁盘。然后，它会将硬盘中的数据与磁盘中的副本进行比较。如果所有文件校验和都匹配，则测试会通过。
- 具有重写功能的驱动器 - 首先，它会从重写磁盘中删除所有信息。然后，它会从硬盘读取数据并将其写入重写磁盘。最后，它会将硬盘上的数据与磁盘的副本进行比较。如果删除操作成功，且所有文件校验和都匹配，则测试会通过。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择合适的 CD ROM 测试名称。

```
rhcert-run
```

按照屏幕上的说明插入相应的介质，并在适当时关闭驱动器的跟踪。

运行时

CD ROM 测试的运行时间取决于介质和驱动器的速度。对于 12x 714MB CD-RW 磁盘，测试将在大约 7 分钟内完成。

A.8. CORE

测试涵盖了什么

核心测试检查系统的 CPU，并确保它们能够在负载下正常工作。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

核心测试实际上由两个独立的例程组成。第一个测试旨在检测时钟 jitter。jitter 是一个当系统时钟不相互同步时发生的条件。系统时钟与 CPU 时钟速度不同，这只是引用 CPU 操作速度的另一个方法。jitter 测试使用 `gettimeofday ()` 函数来获取每个逻辑 CPU 观察到的时间，然后分析返回的值。如果所有 CPU 时钟都是彼此的 .2 纳秒内，则测试会通过。jitter 测试的容错非常紧密。为了获得良好的结果，务必要确保 rhcert 测试在执行测试时仅加载在系统上运行。存在的任何其他计算负载都可能会干扰时间，并导致测试失败。jitter 测试还会检查内核正在使用的时钟源。如果 Intel 处理器没有使用 TSC，它将在日志中打印警告，但这不会影响测试的 PASS/FAIL 状态。

第二个例程在核心测试中运行，是 CPU 负载测试。它是所需的 stress 软件包提供的测试。如果您寻找一种压力测试系统，可以在 rhcert 套件外使用 stress 程序，请在系统上启动多个同时活动，然后监控任何故障。具体来说，它指示每个逻辑 CPU 计算方括号根，它会通过使用 `malloc ()` 和 `free ()` 例程来分别保留和释放内存，并通过调用 `sync ()` 来强制写入磁盘。这些活动持续 10 分钟，如果该时间段内没有发生故障，则测试通过。如果您有兴趣在硬件认证测试之外使用它，请参阅 stress man page。

准备测试

核心测试的唯一准备是安装符合策略指南中规定要求的 CPU。

执行测试

核心测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 Core 测试名称。

```
rhcert-run
```

运行时间，裸机

核心测试本身需要大约 12 分钟才能在裸机系统上运行。测试的 jitter 部分需要一两分钟，并且 stress 部分会完全运行 10 分钟。所需的可支持测试将大约一分钟时间添加到整个运行时。

运行时间，full-virt 客户机

fv_core 测试需要比裸机版本（大约 14 分钟）运行的时间稍长。添加的时间是因为客户机启动/关闭活动以及客户机中运行的所需可支持的测试。在裸机系统上所需的可支持测试将大约一分钟到整个运行时间。

A.9. CPU 扩展

测试涵盖了什么

cpuscaling 测试检查 CPU 的能力，根据其上的计算需求增加和降低时钟速度。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试练习使用不同的扩展监管器（告知 CPU 何时更改为更高或降低时钟速度以及其完成标准化工作负载时的差异）的不同情况。当硬件检测例程在 `/sys` 中找到以下目录时，会调度测试，其中包含多个 cpu 频率：

```
/sys/devices/system/cpu/cpuX/cpufreq
```

每个软件包计划每软件包进行一次 cpuscaling 测试，而不是在每个逻辑 CPU 中列出一行。测试运行时，它将通过 `/sys/devices/system/cpu/cpuX/topology/physical_package_id` 来确定拓扑，并对特定软件包中的所有逻辑 CPU 并行运行测试。

测试首先运行 `turbostat` 命令来收集处理器统计信息。在支持的构架上，`turbostat` 检查 `turbostat` 输出文件中是否看到了提前统计列，但如果文件不包含列，则返回警告。然后，测试会尝试执行 `cstate` 子测试，如果失败，则执行 `pstate` 子测试。

每个 CPU 软件包的测试过程如下：

测试使用 `sysfs` 文件系统中找到的值来确定最大和最小 CPU 频率。您可以使用以下命令查看任何系统的这些值：

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

这里至少会显示两个 `frequencies`（最大值和最小值），但有些处理器能够更精细的 CPU 速度控制，并在文件中显示超过两个值。测试过程中不会特别使用 `max` 和 `min` 间的任何其他 CPU 速度，但它们可用作 `max` 和 `min frequencies` 之间的 CPU 转换。测试过程如下：

1. 测试记录了文件 `/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies` 中的最大和最小处理器速度。
2. 选择了 `userspace governor`，选择最大频率。
3. 通过读取所有处理器的 `/sys/devices/system/cpu/cpuX/cpufreq/scaling_cur_freq` 值来确认最大速度。如果这个值与所选频率不匹配，测试会报告失败。
4. 软件包中的每个处理器都会被赋予将 `pi` 计算到 2×10^{12} 位的同步任务。选择了 `pi` 计算的值，因为它需要有意义的时间才能完成（大约 30 秒）。
5. 计算 `pi` 所需的时间为每个 CPU 进行记录，并为软件包计算平均值。
6. 选择了 `userspace governor`，并设置了最小速度。
7. `sysfs` 数据确认最小速度，如果任何 CPU 不在请求的速度上发生失败。
8. 相同的 `pi` 计算由软件包中的每个处理器和记录的结果执行。

9. 选择 `ondemand governor`，它会根据工作负载在最小和最大速度之间节流 CPU。
10. `sysfs` 数据确认最小速度，如果任何 CPU 不在请求的速度上发生失败。
11. 相同的 `pi` 计算由软件包中的每个处理器和记录的结果执行。
12. 选择 `performance governor`，它会强制 CPU 始终最大速度。
13. `sysfs` 数据确认最大速度，如果任何 CPU 不在请求的速度上发生失败。
14. 相同的 `pi` 计算由每个处理器处理器和记录的结果执行。

现在，分析会在三个子部分上执行。在第 8 步到 8 步，我们获得最大和最小 CPU 速度的 `pi` 计算时间。在两个速度上计算 `pi` 所需的时间的区别应该与 CPU 速度差异成比例。例如，如果假设测试系统的最大频率为 2GHz，最小为 1GHz，则需要系统 30 秒才能以最大速度运行 `pi` 计算，我们希望系统在短短时间内花费 60 秒来计算 `pi`。我们知道，出于各种原因不会获得完美的结果，因此我们可以在结果上获得 10% 利润的错误（比预期快或慢）。在我们的假设示例中，这意味着运行的最小速度在 54 到 66 秒之间可能要花费，并且仍然被视为通过测试(90% 为 $60 = 54$ 和 110% 为 $60 = 66$)。

在 9 步到第 9 步，我们使用 `ondemand` 调控器测试 `pi` 计算时间。这确认系统可以在工作时快速将 CPU 速度提高到最大。我们在步骤 eleven 中获得计算时间，并将其与在第五步中获取的最大速度计算时间进行比较。传递测试具有这两个值，值不超过 10%。

在四十四步中，我们使用 `performance` 调控器测试 `pi` 计算。这确认系统始终可以保持 CPU 处于最大频率。我们采取第 14 步获取的 `pi` 计算时间，并将其与在第 5 步中获取的最大速度计算时间进行比较。同样，通过测试的两个值都不同，超过 10%。

当带有 TurboBoost 功能的 Intel 处理器被 `/proc/cpuinfo` 中存在 `ida` CPU 标记时，会检测到 `cpuscaling` 测试的额外部分。此测试选择每个软件包中的一个 CPU，省略了 CPU0 进行内务处理，并使用 `ondemand governor` 以最大速度来测量性能。当软件包中的所有内核被并行测试时，它预期性能至少比之前的测试快 5%。

准备测试

要准备测试，请确保在 BIOS 中启用了 CPU 频率扩展，并确保安装了满足策略指南中解释要求的 CPU。

执行测试

`cpuscaling` 测试不是交互式的。运行以下命令，然后从显示的列表中选择适当的 CPU 扩展 测试名称。

```
rhcert-run
```

运行时

对于 2013-era、单一 CPU、6 核/12-thread 3.3GHz Intel 的工作站运行 Red Hat Enterprise Linux 6.4、AMD64 和 Intel 64，`cpuscaling` 测试大约需要 42 分钟。有大量内核数量和更多填充套接字的系统将需要更长的时间。所需的 可支持 测试将大约一分钟时间添加到整个运行时。

A.10. DVD

测试涵盖了什么

DVD 测试在以下介质和相关驱动器类型中运行：

- 只读介质和驱动器(DVD-ROM)
- 写出介质和驱动器(DVD+R 和 DVD-R)

- 重写介质和驱动器(DVD+RW 和 DVD-RW)

根据 `udev` 命令的信息，测试套件决定要调度哪些光驱测试(Blu-ray、DVD 或者 CD-ROM)以及要测试的介质类型（只读、可写和重写）。例如，测试套件将针对带有可读取 DVD 和 CD-ROM 磁盘功能的 Blu-ray 驱动器规划以下测试：

- Blu-ray 介质的重写（删除、写入和读取）测试
- DVD 介质的读测试
- CD-ROM 介质的读测试

如果您的驱动器同时支持 DVD-RW 和 DVD+RW 格式，您可以在测试过程中使用任一磁盘类型。您不需要测试这两种格式。另外，您只需要为给定驱动器运行 DVD 测试一次。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试会根据驱动器的功能执行以下任务：

- 只读驱动器 - 首先，它会从磁盘读取数据并将其复制到硬盘。然后，它会将磁盘中的数据与硬盘上的副本进行比较。如果所有文件校验和都匹配，则测试会通过。
- 具有写入功能的驱动器 - 首先，它会从硬盘读取数据并将其写入可写空白磁盘。然后，它会将硬盘中的数据与磁盘中的副本进行比较。如果所有文件校验和都匹配，则测试会通过。
- 具有重写功能的驱动器 - 首先，它会从重写磁盘中删除所有信息。然后，它会从硬盘读取数据并将其写入重写磁盘。最后，它会将硬盘上的数据与磁盘的副本进行比较。如果删除操作成功，且所有文件校验和都匹配，则测试会通过。

执行测试

运行以下命令，然后从显示的列表中选择合适的 DVD 测试名称。

```
rhcert-run
```

按照屏幕上的说明插入相应的介质，并在适当时关闭驱动器的跟踪。

运行时

DVD 测试的运行取决于介质和驱动器的速度。对于 4x 4.7GB DVD-RW 磁盘，测试在大约 13 分钟内完成。

A.11. ETHERNET

测试涵盖了什么

只有在测试套件无法识别网络设备的速度时，才会显示以太网测试。这可能是由于未插入电缆或某些其他故障导致意外检测连接速度。请退出测试套件，检查您的连接，并在设备正确连接时再次运行测试套件。如果问题仍然存在，请联系您的红帽支持代表以获得帮助。



注意

如果您在测试中使用带有 RHEL 7 主机的 RHEL 8 测试服务器，则必须手动启动 **httpd** 服务，因为测试套件不会自动启动此服务。

以下示例显示有两个千兆以太网设备的系统，即 eth0 和 eth1。设备 eth0 正确连接，但 eth1 没有插入。

ethtool 命令的输出显示 eth0 的预期千兆以太网速度为 1000Mb/s：

```
# ethtool eth0
Settings for eth0:
Supported ports: [ TP ]
Supported link modes:  10baseT/Half 10baseT/Full
                      100baseT/Half 100baseT/Full
                      1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
                      100baseT/Half 100baseT/Full
                      1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 2
Transceiver: internal
Auto-negotiation: on
MDI-X: on
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000007 (7)
                      drv probe link
Link detected: yes
```

但是，在 eth1 上，**ethtool** 命令显示未知的速度，这会导致计划以太网测试。

```
# ethtool eth1
Settings for eth1:
Supported ports: [ TP ]
Supported link modes:  10baseT/Half 10baseT/Full
                      100baseT/Half 100baseT/Full
                      1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
                      100baseT/Half 100baseT/Full
                      1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: Unknown!
```

```
Duplex: Unknown! (255)
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: Unknown
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000007 (7)
      drv probe link
Link detected: no
```

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

A.12. EXPRESSCARD

测试涵盖了什么

expresscard 测试查找带有两个 ExpressCard 接口、USB 和 PCI Express (PCIe) 的设备，并确认系统可以同时通信。ExpressCard 插槽检测不像检测系统中的其他设备那样简单。ExpressCard 专门设计为不需要任何专用的网桥设备。它只是组合了 PCIe 和 USB 的 novel 表单因素接口。因此，我们没有可在 udev 输出中看到的特定 "ExpressCard 插槽" 条目。我们决定在包含电池、USB 和 PCIe 接口的系统上调度测试，因为我们看到没有使用此硬件组合的 ExpressCard- Containing 笔记本电脑以外的设备。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试首先使用 `lsusb` 和 `lspci` 命令获取 USB 和 PCIe 总线中所有设备的快照。然后，它会询问测试者系统中存在多少个 ExpressCard 插槽。测试程序被要求在其中一个插槽中插入一个卡。系统扫描 USB 和 PCIe 总线，并将结果与原始 `lsusb` 和 `lspci` 输出进行比较，以检测任何新设备。如果检测到 USB 设备，系统会要求您删除卡，并将带有 PCIe 接口的卡插入到同一插槽中。如果检测到基于 PCIe 的卡，系统会要求您删除它，并将基于 USB 的卡插入到同一插槽中。如果使用两个接口（例如，使用站卡）插入卡，它会一次性满足两个插槽测试要求。对于系统中的所有插槽，这个过程会重复。

准备测试

您需要带有 USB 和 PCIe 总线的 ExpressCard 卡。这可以是两个独立的卡，也可以是带有两个接口的卡。在运行测试前删除所有 ExpressCard 卡。

执行测试

expresscard 测试是交互式的。运行以下命令，然后从显示的列表中选择合适的 Expresscard 测试名称。

```
rhcert-run
```

它将提示您删除所有 ExpressCards，然后询问是否加载 PCI Express 热插拔模块(pciehp)的权限。在系统运行时，需要 PCIe 热插拔功能来添加或删除基于 PCIe 的 ExpressCard 卡。接下来，测试将要求您输入系统中的 ExpressCard 插槽数量，然后提示以任何顺序插入和移除具有两种类型的接口(USB 和 PCIe)的卡。

A.13. FINGERPRINTREADER

测试涵盖了什么

如果系统有内置或插件 指纹读取器，则计划指纹reader 测试。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

此测试会验证指纹读取器是否可以扫描、注册和验证指纹管理器中注册的指纹。

准备测试

确保指纹读取器已连接到系统。

执行测试

此测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 指纹reader 测试名称。

```
rhcert-run
```

测试将开始检测指纹读取器，然后提示您放置和扫描指纹读取器的右索引，直到注册完成为止。为进行验证，系统将提示您再次扫描与注册指纹匹配的 finger。

运行时

完成测试需要几分钟时间，直到读取器完成扫描并显示 *注册完成状态*。

A.14. FIRMWARE

测试涵盖了什么

对于使用统一可扩展固件接口(UEFI)和 EFI 系统资源表(ESRT)的 x86_64 架构系统支持在 RHEL 版本 8 上运行固件测试。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

测试运行以下子测试：

- **安全检查子测试**：子测试通过验证系统和设备固件是否满足 **HSI-1 级别标准**，检查测试中的主机是否遵循安全最佳实践。测试使用 `fwupdagent security --force` 命令检查 HSI-1 安全属性并捕获输出。
- **更新服务子测试**：子测试会验证测试中的主机是否可以通过 Linux Vendor Firmware Service (LVFS) 下载并安装固件更新。

成功标准

- 只有在所有 HSI-1 属性都通过时，测试才会通过。
- 如果系统安装 LVFS 更新，则测试通过。

准备测试

- 确保 `test` 下的主机正在运行 RHEL 8.0 或更高版本。
- 确定系统以 UEFI 模式引导，而不是传统的 BIOS 模式。

执行测试

此测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的固件测试名称。

```
rhcert-run
```

运行时

此测试需要一分钟才能运行。任何其它强制或所选的测试都会添加到整个运行时。

A.15. FV_CORE

`fv_core` 测试是启动 FV 客户机并在其上运行核心测试的打包程序。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9



注意

第一次运行任何完全虚拟化测试时，测试工具将需要获取 FV 客户机文件。测试工具的执行时间取决于 FV 客户机文件的传输速度。例如，

- 如果 FV 客户机文件位于测试服务器上，并且您使用 1GbE 或更快的网络，则大约需要一分钟或两个来传输大约 300MB 的客户机文件。
- 如果从 CWE API 检索文件，当客户机文件没有在测试服务器上安装或找到时，第一次运行时将取决于 CWE API 的传输速度。

当客户机文件在 Host Under Test (HUT)上可用时，它们将用于稍后运行 `fvuildDefaults` 测试。

其他资源

- 有关测试方法和运行时间的更多信息，请参阅 [内核](#)。
- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.16. FV_CPU_PINNING

CPU 固定是一种将系统资源专用于特定进程的方法。例如，应用程序可能会锁定特定逻辑内核，以减少任务切换。

虚拟化(fv) CPU 固定方法类似，但固定是从基于 KVM 虚拟机中的虚拟 CPU (vCPU)到主机上的物理内核完成。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试涵盖了什么

`fv_cpu_pinning` 测试验证客户虚拟机(VM)的 vCPU 可以配置为主机机器的专用 CPU。此测试在主机机器中运行，并在 RHEL 8 上支持基于 RHEL 8 的功能资格。

测试的作用

`fv_cpu_pinning` 测试运行三个子测试：设置客户机虚拟机 VCPU、执行 FV CPU Pinning，并验证 FV CPU Pinning。Setup guest VM VCPU 子测试计算主机的逻辑内核数，并将最后一个编号的核心与专用于虚拟机中的隔离。Perform FV CPU Pinning 子测试会将虚拟机的 vCPU 进一步固定到主机中的 CPU。然后，测试会使用命令 `virt vcpupin` 和 `vcpuinfo` 验证固定，并检查 `/proc` 目录信息。最后，验证 FV CPU Pinning 使用负载测试来验证客户虚拟机 vCPU 工作负载是否仅由固定 CPU 处理。

准备测试

运行此测试没有特殊要求。

执行测试

`fv_cpu_pinning` 测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `fv_cpu_pinning` 测试名称。

-

```
rhcert-run
```

运行时

`fv_cpu_pinning` 测试需要大约 5 分钟才能完成。任何其它强制或所选的测试都会添加到整个运行 ti

其他资源

- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.17. FV_LIVE_MIGRATION

测试涵盖了什么

`fv_live_migration` 测试检查 Host Under Test (HUT)将正在运行的虚拟机迁移到测试服务器的功能。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

测试执行多个子测试，以完成正在运行的虚拟机从 HUT 迁移到测试服务器。成功完成测试需要所有子测试通过。该测试会检查 HUT 是否满足迁移要求、配置虚拟机并在 HUT 上启动。然后，它会将正在运行的虚拟机从 HUT 迁移到测试服务器。迁移后，验证虚拟机不再运行在 HUT 上，并在测试服务器上运行。最后，测试将正在运行的虚拟机从测试服务器迁移到 HUT，并再次检查虚拟机是否在 HUT 上运行，且不再在测试服务器上运行。

准备测试

确保测试服务器和 HUT 正在运行 Red Hat Enterprise Linux 8，并且 `redhat-certification-hardware` 软件包已安装在测试服务器和 HUT 上。

在测试服务器和 HUT 的相应 `/etc/hosts` 文件中添加主机名，并制作完全限定名称的主机名，如下所示：

- `<IP 地址 HUT> <hostname of HUT>`
- `<IP address of test server> <hostname of test server>`

执行测试

测试是非交互式的。目前，此测试只能通过 CLI 手动计划并执行。

对于 RHEL 8:

```
# rhcert-cli plan --add -t fv_live_migration
```

```
# rhcert-cli run -t fv_live_migration --server=<server name>
```

在 RHEL 9 中：

```
# rhcert-cli plan --add -t fv_live_migration
```

```
# rhcert-cli run --test fv_live_migration --server=<server name>
```

运行时

完成测试需要大约 5 分钟。但是，如果测试服务器和 HUT 分别属于同一或不同的实验或网络，则时间可能会减少或增加。

其他资源

- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

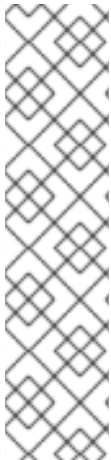
A.18. FV_MEMORY

fv_memory 测试是启动 FV 客户机并在其上运行内存测试的打包程序。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9



注意

第一次运行任何完全虚拟化测试时，测试工具将需要获取 FV 客户机文件。测试工具的执行时间取决于 FV 客户机文件的传输速度。例如，

- 如果 FV 客户机文件位于测试服务器上，并且您使用 1GbE 或更快的网络，则大约需要一分钟或两个来传输大约 300MB 的客户机文件。
- 如果从 CWE API 检索文件，当客户机文件没有在测试服务器上安装或找到时，第一次运行时将取决于 CWE API 的传输速度。

当客户机文件在 Host Under Test (HUT) 上可用时，它们将用于稍后运行 fvuildDefaults 测试。

其他资源

- 有关测试方法和运行时的更多信息，请参阅 [内存](#)。
- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.19. FV_PCIE_STORAGE_PASSTHROUGH

测试涵盖了什么

fv_pcie_storage_passthrough 测试用于验证主机机器上的基于 PCIe 的存储设备（如 SAS 和 SATA）的控制是否可以传送到虚拟机。测试在 Red Hat Enterprise Linux 8 上被支持，且必须在主机中运行。如果主机支持设备透传并启用了 IOMMU，则会自动计划此测试。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

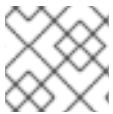
- RHEL 8
- RHEL 9

测试的作用

测试执行多个子测试，将主机的 HBA 设备附加到虚拟机，然后在虚拟机中运行存储测试。成功完成测试需要所有子测试通过。测试会验证连接到主机虚拟机的 PCIe 设备是否可以分配在客户虚拟机中原生显示，将客户虚拟机配置为使用 passthrough PCIe 设备，并启动虚拟机并确保该设备按预期工作。

准备测试

确保主机机器支持设备透传，并启用了 IOMMU。要配置，请参阅 [PCI Passthrough 配置主机](#)。



注意

不要在带有主机 root 分区的存储设备上运行测试。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `fv_pcie_storage_passthrough` 测试名称。

```
rhcert-run
```

运行时

运行测试需要大约 30 分钟。

其他资源

- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.20. FV_USB_NETWORK_PASSTHROUGH

测试涵盖了什么

`fv_usb_network_passthrough` 测试用于验证主机机器中 USB 附加网络设备的控制是否可以传送到虚拟机。测试在 Red Hat Enterprise Linux 8 及更高版本中被支持，且必须在主机上运行。如果主机机器支持设备透传并启用了 IOMMU，则会自动计划此测试。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

测试执行多个子测试，将主机的 USB 设备附加到虚拟机，然后在虚拟机中运行网络测试。成功完成测试需要所有子测试通过。测试会验证连接到主机的 USB 设备是否可以被分配为在客户虚拟机中原生显示，将客户虚拟机配置为使用 passthrough USB 设备，并启动虚拟机并确保该设备按预期工作。

准备测试

- 确保 USB 设备插入到支持设备透传的 HUT 中，并启用了 IOMMU。要配置，请参阅 [PCI Passthrough 配置主机](#)。
- 确保 HUT 至少有两个 NIC，并且两个网络都可以路由到测试服务器。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `fv_usb_network_passthrough` 测试名称。

```
rhcert-run
```



注意

如果因为网络带宽问题导致测试失败，则可能需要增加分配给虚拟机的 CPU 和 RAM，才能达到更高的带宽。

运行时

测试需要大约 90 分钟才能运行，但根据 USB 设备和连接的大小和速度，长度会有所不同。

其他资源

- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.21. FV_USB_STORAGE_PASSTHROUGH

测试涵盖了什么

`fv_usb_storage_passthrough` 测试用于验证主机机器上的对 USB 附加存储设备的控制是否可以传送到虚拟机。测试在 Red Hat Enterprise Linux 8 上被支持，且必须在主机中运行。如果主机支持设备透传并启用了 IOMMU，则会自动计划此测试。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

测试执行多个子测试，将主机的 USB 设备附加到虚拟机，然后在虚拟机中运行存储测试。成功完成测试需要所有子测试通过。测试会验证连接到主机的 USB 设备是否可以被分配为在客户虚拟机中原生显示，将客户虚拟机配置为使用 passthrough USB 设备，并启动虚拟机并确保该设备按预期工作。

准备测试

确保 USB 设备插入到支持设备透传的主机机器中，并启用了 IOMMU。要配置，请参阅 [Red Hat Virtualization 管理指南中的为 PCI Passthrough 配置主机部分](#)。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `fv_usb_storage_passthrough` 测试名称。

```
rhcert-run
```

运行时

测试需要大约 90 分钟才能运行，但根据 USB 设备和连接的大小和速度，长度会有所不同。

其他资源

- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.22. FV_PCIE_NETWORK_PASSTHROUGH

测试涵盖了什么

`fv_pcie_network_passthrough` 测试用于验证对基于 PCIe 的网络设备（如 NIC、LOMs、ALOMs）的控制是否可以传送到虚拟机。测试在 Red Hat Enterprise Linux 8 及更高版本上被支持，且必须在主机中运行。如果主机机器支持设备透传并启用了 IOMMU，则会自动计划此测试。

从 RHEL 9.4 开始，这个测试支持在 ARM 系统上运行。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

测试执行多个子测试，将主机的网络设备附加到虚拟机，然后在虚拟机中运行网络测试。成功完成测试需要所有子测试通过。测试会验证连接到主机虚拟机的 PCIe 设备是否可以分配在客户虚拟机中原生显示，将客户机虚拟机配置为使用 passthrough PCIe 设备，并启动虚拟机并确保该设备按预期工作。

准备测试

- 确保 Host Under Test (HUT) 支持设备透传，并启用了 IOMMU。要配置，请参阅 *Red Hat Virtualization 管理指南* 中的 [为 PCI Passthrough 配置主机部分](#)。
- 确保 HUT 至少有两个 NIC，并且两个网络都可以路由到测试服务器。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `fv_pcie_network_passthrough` 测试名称。

```
rhcert-run
```



注意

如果因为网络带宽问题导致测试失败，则可能需要增加分配给虚拟机的 CPU 和 RAM，才能达到更高的带宽。

运行时

运行测试需要大约 30 分钟。

其他资源

- 有关客户机镜像的更多信息，请参阅 [测试执行期间下载客户机镜像](#)。

A.23. INFINIBAND 连接

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

Infiniband Connection 测试运行以下子tests，以确保使用基准功能（如果适用），从测试的 onset 下拉菜单中选择的 IP 地址：

1. ping 测试
从 HUT 上测试的设备起始 IP 地址运行 ping 到测试服务器的所选 IP 地址。
2. Rping 测试
使用所选测试服务器 IP 地址对测试服务器和 HUT 运行 rping，然后比较结果以验证其运行完成。
3. Rcopy test
在测试服务器和 HUT 上运行 rcopy，发送随机生成的文件并在测试服务器和 HUT 上比较 md5sums 以验证是否成功传输。
4. RDMA-ndd 服务测试
验证 stop、start 和 restart service 命令可以正常工作。
5. opensm 服务测试
验证 stop、start 和 restart service 命令可以正常工作。
6. LID 验证测试
验证该设备的 LID 是否已设置，而不是默认值。
7. Smpquery test
使用设备和端口在测试服务器上运行 smpquery，以便通过光纤注册了设备/端口的另一个验证。
8. ib_write_bw test
从 HUT 运行 ib_write_bw 到所选测试服务器的 IP 地址，以测试 InfiniBand 写入带宽并验证它是否可以访问所需的带宽。在带宽测试过程中调整 queue pair 参数，以达到接近行率的吞吐量。
9. ib_read_bw test
从 HUT 运行 ib_read_bw 到所选测试服务器的 IP 地址，以测试 InfiniBand 读取带宽，并确认它是否可以访问所需的带宽。在带宽测试过程中调整 queue pair 参数，以达到接近行率的吞吐量。
10. ib_send_bw test
从 HUT 运行 ib_send_bw 到所选测试服务器的 IP 地址，以测试 InfiniBand 发送带宽并验证它是否可以访问所需的带宽。在带宽测试过程中调整 queue pair 参数，以达到接近行率的吞吐量。

准备测试

- 确保测试服务器和 HUT 是同一光纤上的独立计算机。

执行测试

这是一个交互式测试。运行以下命令，然后从显示的列表中选择适当的 **infiniband** 连接测试名称。

```
rhcert-run
```

系统将提示您选择一个执行测试的 IP 地址（测试服务器的 IP 地址）。在您运行测试的 HUT 设备的同一光纤中，选择一个与设备对应的 IP 地址。

表 A.1. 手动添加并运行测试

速率类型	手动添加 infiniband 连接测试的命令	手动运行 infiniband 连接测试的命令
Infiniband_QDR	<pre>rhcert-cli plan --add --test Infiniband_QDR --device <devicename>_devicePort_ <port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test Infiniband_QDR --server <test server IP addr></pre>
Infiniband_FDR	<pre>rhcert-cli plan --add --test Infiniband_FDR --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test Infiniband_FDR --server <test server IP addr></pre>
Infiniband_EDR	<pre>rhcert-cli plan --add --test Infiniband_EDR --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test Infiniband_EDR --server <test server IP addr></pre>
Infiniband_HDR	<pre>rhcert-cli plan --add --test Infiniband_HDR --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test Infiniband_HDR --server <test server IP addr></pre>

速率类型	手动添加 infiniband 连接测试的命令	手动运行 infiniband 连接测试的命令
Infiniband_NDR	<pre>rhcert-cli plan --add --test Infiniband_NDR --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test Infiniband_NDR --server <test server IP addr></pre>
Infiniband_Socket_Direct	<pre>rhcert-cli plan --add --test Infiniband_Socket_Direct</pre>	<pre>rhcert-cli run --test Infiniband_Socket_Direct -- server <test server IP addr></pre>

将 `<device name>` , `<port number >` , `<net device >` , 和 `<test server IP addr >` 替换为适当的值。

运行时

运行此测试需要少于 10 分钟。

其他资源

- 有关 InfiniBand 和 RDMA 的更多信息，[请参阅了解 InfiniBand 和 RDMA 技术。](#)

A.24. INTEL_SST

测试涵盖了什么

intel_sst 测试是一个 CPU 频率扩展测试，它对 Intel 的 Speed Select Technology (SST)功能进行练习。使用此功能自定义每个内核的性能，以匹配 CPU 到工作负载，并为每个性能分配。这可让您在运行时提高目标应用程序的性能。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

intel_sst 测试只在启用了 SST 的系统上运行，并支持以下功能：

- speed Select Base Freq (SST-BF)- 允许特定内核通过减少其他内核的基本频率(P1)来运行更高的基本频率(P1)。
- 频率优先级(SST-CP)- 通过减少运行较低优先级软件的内核的频率，允许特定内核进行时钟。

测试会检查系统上是否支持并配置了上述功能。根据结果，它将只执行相应功能的一个子测试。

准备测试

您必须只在 Intel 芯片组构架上运行此测试。

- 在基于 Red Hat Enterprise Linux (RHEL)的平台上使用 Intel® SST-BF 功能。
先决条件：
 1. 在 BIOS 中启用 Intel® SST-BF 功能
 2. 配置内核参数 - intel_idle.max_cstate=1
- 在基于 Red Hat Enterprise Linux (RHEL)的平台上使用 Intel® SST-CP 功能。
先决条件：
 1. 在 BIOS 中启用 Intel® SST-CP 功能
 2. 配置内核参数 - intel_idle.max_cstate=1 intel_pstate=disable

执行测试

此测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 intel_sst 测试名称。

```
rhcert-run
```

运行时

完成此测试需要 5 分钟时间。任何其它强制或所选的测试都会添加到整个运行时。

A.25. IPXE

iPXE 测试覆盖的内容

iPXE 测试是在 x86 Red Hat Enterprise Linux (RHEL)系统上运行的交互式测试。系统应该以 UEFI 引导模式引导。

如果 efi 目录存在，则机器在 UEFI 引导模式下运行。运行以下命令，以确定机器是否在 UEFI 模式下运行：

```
ls /sys/firmware/efi/
```

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

iPXE 是领先的开源网络引导固件。它提供了一个完整的 PXE 实现，它带有额外的功能，如从 HTTP、SAN 和 Wireless Network 启动等。此测试会检查底层 NIC 是否使用 HTTP 引导支持 iPXE。

在执行 iPXE 时，测试服务器不会返回任何可引导镜像。引导屏幕将显示错误无法引导，这是预期的错误消息。测试服务器将使用 RHEL OS 的下一个引导装载程序引导。

准备测试

- 确保 test 下的主机处于 UEFI 引导模式。iPXE 首先测试它在测试下找到的接口，因此请确保在其中插入需要测试的接口。
- 在运行此测试时，请确保 httpd 服务未在测试服务器上运行，因为此测试使用端口 80 与测试服务器通信。

执行测试

1. 运行以下命令，然后从显示的列表中选择适当的 iPXE 测试名称。

```
rhcert-run
```

2. ipxe 测试没有出现在测试计划中，因此您必须分别使用下列命令来手动规划和执行它。

```
# rhcert-cli plan --add --test iPXE
```

```
# rhcert-cli run --test iPXE
```

3. 测试将首先为 iPXE 测试配置 Host Under Test (HUT)。它将保存 HUT 的 MAC 详情，然后创建一个带有 ipxe 二进制文件的新引导装载程序，并将引导装载程序标记为下一次启动。之后，它将提示重新启动，按 Yes 继续。测试服务器将在发送 reboot 命令后显示等待响应。
4. HUT 将重启到新的引导装载程序，后者又加载 iPXE 提示符并执行 GET 请求，以查看它是否能够访问测试服务器。因为它只是一个 GET 重新排队引导会失败，系统会回退到下一个引导装载程序，如 RHEL OS。
5. 测试服务器将持续监控测试下的主机，以查看它是否已重启。重新引导后，测试将继续。测试将首先恢复为 iPXE 进行的引导更改，然后验证 iPXE 引导是否成功。
6. 它将从 iPXE 引导的 GET 请求接收的 MAC 地址与已保存的 MAC 进行比较。如果 MAC 与 iPXE 测试匹配成功。

运行时

运行测试需要少于 5 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.26. IWARP 连接

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

IWarp Connection 测试运行以下子测试，以确保使用基准功能（如果适用），从测试的 onset 的下拉列表中选择 IP 地址：

1. ping test - 从 HUT 上测试的设备的起始 IP 地址运行 ping 到测试服务器的所选 IP 地址。

2. Rping test - 在测试服务器上运行 rping 和使用所选测试服务器 IP 地址的 HUT，然后比较结果以验证其运行完成。
3. Rcopy test - 在测试服务器和 HUT 上运行 rcopy，发送随机生成的文件并在测试服务器上比较 md5sums，验证是否成功传输。
4. ethtool test - 运行在检测到的 roce 设备的 net 设备传递的 ethtool 命令。
5. ib_write_bw test
从 HUT 运行 ib_write_bw 到测试服务器的所选 IP 地址，以测试 IWrap 写入带宽，并验证它是否可以访问所需的带宽。
6. ib_read_bw test
从 HUT 运行 ib_read_bw 到测试服务器的所选 IP 地址，以测试 IWrap 读取带宽，并验证它是否可以访问所需的带宽。
7. ib_send_bw test
从 HUT 运行 ib_send_bw 到测试服务器的所选 IP 地址，以测试 IWrap 发送带宽，并验证它是否可以访问所需的带宽。

准备测试

- 确保测试服务器和 HUT 是同一光纤上的独立计算机。

执行测试

这是一个交互式测试。运行以下命令，然后从显示的列表中选择合适的 iwrap 连接 测试名称。

```
rhcert-run
```

系统将提示您选择一个执行测试的 IP 地址（测试服务器的 IP 地址）。在您运行测试的 HUT 设备的同一光纤中，选择一个与设备对应的 IP 地址。

表 A.2. 手动添加并运行测试

速度类型	手动添加 IWarpConnection Test 的命令	手动运行 IWarpConnection 测试的命令
10GigiWarp	<pre>rhcert-cli plan --add --test 10GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 10Gigiwarp --server <test server IP addr></pre>
20GigiWarp	<pre>rhcert-cli plan --add --test 20GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 20GigiWarp --server <test server IP addr></pre>

速度类型	手动添加 IWarpConnection Test 的命令	手动运行 IWarpConnection 测试的命令
25GigiWarp	<pre>rhcert-cli plan --add --test 25GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 25GigiWarp --server <test server IP addr></pre>
40GigiWarp	<pre>rhcert-cli plan --add --test 40GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 40GigiWarp --server <test server IP addr></pre>
50GigiWarp	<pre>rhcert-cli plan --add --test 50GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 50GigiWarp --server <test server IP addr></pre>
100GigiWarp	<pre>rhcert-cli plan --add --test 100GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 100GigiWarp --server <test server IP addr></pre>
200GigiWarp	<pre>rhcert-cli plan --add --test 200GigiWarp --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 200GigiWarp --server <test server IP addr></pre>

将 `<device name>` , `<port number >` , `<net device >` , 和 `<test server IP addr >` 替换为适当的值。

运行时

运行此测试需要少于 10 分钟。

其他资源

- 有关 InfiniBand 和 RDMA 的更多信息，[请参阅了解 InfiniBand 和 RDMA 技术](#)。

A.27. KDUMP

测试涵盖了什么

kdump 测试使用 kdump 服务来检查系统可在崩溃后捕获 vmcore 文件，并且捕获的文件是否有效。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

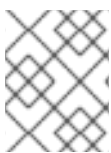
测试的作用

测试包括以下子测试：

- kdump 带有本地：使用 kdump 服务，这个子测试执行以下任务：
 - 在 test (HUT)下崩溃主机。
 - 将 vmcore 文件写入本地 /var/crash 目录。
 - 验证 vmcore 文件。
- kdump with NFS: 使用 kdump 服务，这个子测试执行以下任务：
 - 在 HUT 的 /var/rhcert/export 目录中挂载 /var/rhcert/ export 文件系统。此文件系统通过 NFS 从测试服务器共享。
 - 崩溃 HUT。
 - 将 vmcore 文件写入 /var/crash 目录。
 - 验证 vmcore 文件。

准备测试

- 在运行测试前，请确保 HUT 连接到测试服务器。
- 确保 rhcertd 进程在测试服务器上运行。认证测试套件自动准备 NFS 文件系统。如果套件无法设置环境，则测试会失败。



注意

如果您在测试中使用带有 RHEL 7 主机的 RHEL 8 测试服务器，则必须手动启动 httpd 服务，因为测试套件不会自动启动此服务。

执行测试

1. 登录到 HUT。
2. 运行 kdump 测试：

- 要使用 `rhcert-run` 命令，请执行以下步骤：

- i. 运行 `rhcert-run` 命令：

```
# rhcert-run
```

- ii. 选择 `kdump` 测试。
测试会按顺序运行两个子测试。

- 要使用 `rhcert-cli` 命令，请选择是否按顺序运行两个子测试，或者指定子测试：

- 要按顺序运行这两个子测试，请使用以下命令：

```
# rhcert-cli run -test=kdump --server=<test server's IP>
```

- 要只使用本地子测试运行 `kdump`，请使用以下命令：

```
# rhcert-cli run -test=kdump -device=local
```

- 要只使用 NFS 子测试运行 `kdump`，请使用以下命令：

```
# rhcert-cli run -test=kdump -device=nfs --server=<test server's IP>
```

另外，对于带有 NFS 测试的 `kdump`，在 Test 服务器上执行以下命令：

```
# rhcertd start
```

3. 等待 HUT 在崩溃后重新启动。

`kdump` 服务在将 `vmcore` 文件保存到 `/var/crash` 目录中时显示一些信息。保存 `vmcore` 文件后，HUT 重启。

4. 重启后登录到 HUT，`rhcert` 套件将验证 `vmcore` 文件是否存在，如果有效。如果文件不存在或无效，则测试会失败。

如果您按顺序运行子测试，则带有 NFS 子测试的 `kdump` 将在以前的 `vmcore` 文件验证完成后启动。

运行时

`kdump` 测试的运行时间因 HUT 中的 RAM 量、测试服务器和 HUT 的磁盘速度、测试服务器的网络连接速度以及重启 HUT 所需的时间而有所不同。

对于具有 8GB RAM 的 2013-era 工作站，7200 RPM 6Gb/s SATA 驱动器、到测试服务器的千位以太网连接以及 1.5 分钟重启时间，本地 `kdump` 测试可在大约四分钟内完成，包括重启。同样的 2013-era 工作站可以在约五分钟内完成 NFS `kdump` 测试，以便类似装置网络测试服务器。可支持的测试将大约添加一分钟到整个运行时间。

A.28. LID

测试涵盖了什么

`lid` 测试仅对已集成显示的系统有效，因此有一个可打开和关闭的 `lid`。通过搜索具有“`lid`”的设备的 `udev` 数据库来检测到 `lid`：

```
E: NAME="Lid Switch"
```

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试可确保系统可以确定其 lid 何时通过 udev 中的参数打开，并在 lid 关闭时关闭显示的后端。

准备测试

要准备测试，请确保电源管理设置没有将系统置于休眠状态，或者在 lid 关闭时休眠系统。在 Red Hat Enterprise Linux 7 中，使用 Tweak 工具在 lid close 中禁用 suspend 或 hibernate。在开始测试运行前，请确保 lid 为 open。

执行测试

lid 测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 lid 测试名称。

```
rhcert-run
```

系统将询问您是否已准备好开始测试，因此 *回答是继续*。提示时关闭 lid，监视是否关闭后端。当笔记本电脑关闭时，您可能必须查看键盘和 lid 之间的小空间，以验证后端是否已关闭。如果后端关闭，则回答是，如果后端未关闭。

运行时

lid 测试需要大约 30 秒才能执行，基本上是关闭 lid 所需的时间，以便关闭后端。由于此测试在笔记本电脑上运行，因此每个运行所需的可支持的测试必须附带挂起测试。挂起测试会在每个测试运行时大约添加 6 分钟，并且支持可以再加一分钟。

A.29. MEMORY

内存测试涵盖了什么

内存测试用于测试系统 RAM。它不测试 USB 闪存内存、SSD 存储设备或任何其他基于 RAM 的硬件。它只测试主内存。

在规划过程中添加了一个内存每个 CPU 内核检查，以验证 HUT 满足 RHEL 最低要求内存标准。它是多个硬件认证测试的规划条件，包括内存、核心、实时以及所有完整虚拟化测试。

如果每个 CPU 内核检查的内存没有通过，则不会自动规划上述测试。但是，可以通过 CLI 手动规划这些测试。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用：测试使用文件 `/proc/meminfo` 来确定系统中安装了多少内存。一旦知道安装了多少，它会检查系统架构是否为 32 位或 64 位。然后，它会确定交换空间是否可用，或者是否没有交换分区。根据系统是否有交换文件，测试会运行一次或使用稍有不同的设置：

1. 如果交换可用，请为内存测试分配比系统中实际安装更多的 RAM。这会强制在运行时使用交换空间。
2. 无论交换存在，为内存测试分配尽可能多的 RAM，同时保持在强制内存不足(OOM)终止的限制下。此测试版本始终运行。

在内存测试的迭代中，`malloc ()` 用于分配 RAM，RAM 被认为是任意十六进制字符串(0xDEADBEEF)的写入，并执行测试以确保 0xDEADBEEF 在预期地址的 RAM 中实际存储在 RAM 中。测试在测试完成后调用 `free ()` 以释放 RAM。将使用多个线程或多个进程来分配 RAM，具体取决于进程大小大于要测试的内存量。

准备测试

根据策略指南中的规则，在系统中安装正确 RAM 量。

执行测试

内存测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的内存测试名称。

```
rhcert-run
```

运行时间，裸机

内存测试需要大约 16 分钟才能在 2013 年 3 月运行，单 CPU、6 核/12-thread 3.3GHz 基于 Intel 的工作站，运行 Red Hat Enterprise Linux、AMD64 和 Intel 64 的 RAM。在 RAM 更多系统中测试将花费更长的时间。所需的可支持测试将大约一分钟时间添加到整个运行时。

运行时间，full-virt 客户机

`fv_memory` 测试的时间比裸机版本大约 18 分钟要长，才能在客户机中运行。添加的时间是因为客户机启动/关闭活动以及客户机中运行的所需可支持的测试。在裸机系统上所需的可支持测试将大约一分钟到整个运行时间。`fv_memory` 测试运行时间与裸机内存测试不同，因为分配给我们预构建的客户机的 RAM 量始终相同。底层实际系统速度将存在变化，但测试过程中使用的 RAM 量不会从计算机更改为计算机。

为 EC2 创建并激活交换空间：合作伙伴可执行以下步骤来创建和激活 EC2 的交换

```
sudo dd if=/dev/zero of=/swapfile bs=1M count=8000
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
swapon -s
edit file /etc/fstab and add the following line:
/swapfile swap swap defaults 0 0
write file and quit/exit
```

A.29.1. memory_HBM

memory_HBM 测试覆盖的内容

`memory_HBM` 测试用于在存在它的系统中测试系统高带宽内存(HBM)。根据 HBM 操作模式计划三个可能的测试。如果系统 HBM 不支持常规内存测试，则计划进行常规内存测试。

支持的 RHEL 版本

- RHEL 8
- RHEL 9

测试的作用

`memory_HBM` 测试是存在 HBM 的系统的内存测试。

准备测试

确保系统 HBM 满足策略指南中指定的要求。

执行测试

1. 运行 `rhcert-cli` 计划。如果您的 HBM 配置满足要求以及以下条件之一，则会规划 `memory_HBM` 测试之一：
 - `memory_HBM_only`: 系统中没有安装 DIMM
 - `memory_HBM_cache`: HBM 充当 DIMM 的缓存
 - `memory_HBM_flat`: DIMM 和 HBM 作为内存总量提供
2. 要运行测试，请使用命令 `rhcert-cli run --test`。例如，`rhcert-cli run --test hwcert/memory_HBM_cache` 运行 `memory_HBM_cache` 测试。
3. 按照测试的说明进行操作。
4. 使用 `rhcert-print` 检查结果。
5. 要保存结果，请使用 `rhcert-save`。

A.30. NETWORK

测试涵盖了什么

网络测试检查通过 TCP/IP 网络传输数据的设备。测试可以根据针对它设计的对应测试来检查有线和无线设备的多个连接速度和带宽，如下表所示：

Network test 下的不同的测试

以太网测试	描述
1GigEthernet	网络测试增加了 1 千位以太网连接的速度检测。
10GigEthernet	网络测试增加了 10 GB 位以太网连接的速度检测。
20GigEthernet	网络测试增加了 20 千位以太网连接的速度检测。
25GigEthernet	网络测试增加了 25 千位以太网连接的速度检测。
40GigEthernet	网络测试增加了 40 千位以太网连接的速度检测。

以太网测试	描述
50GigEthernet	网络测试增加了 50 千位以太网连接的速度检测。
100GigEthernet	网络测试增加了 100 千位以太网连接的速度检测。
200GigEthernet	网络测试增加了 200 千兆位以太网连接的速度检测。
Ethernet	如果您的本地测试计划中列出了以太网测试，这表示测试套件无法识别该设备的速度。在尝试测试特定设备前检查连接。

无线测试	描述
WirelessG	为 802.11g 无线以太网连接添加了速度检测的网络测试。
WirelessN	为 802.11n 无线以太网连接添加了速度检测的网络测试。
WirelessAC	为 802.11ac 无线以太网连接添加了速度检测的网络测试。
WirelessAX (Superseded by WiFi6)	为 802.11ax 无线以太网连接添加了速度检测的网络测试。
WiFi6	为 802.11ac 无线以太网连接添加了速度检测的网络测试。
WiFi6E	为 802.11ac 无线以太网连接添加了速度检测的网络测试。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试运行以下子tests以收集有关所有网络设备的信息：

1. 界面上的 bounce 测试使用 `nmcli conn up` 和 `nmcli conn down` 命令进行（如果为 RHEL 7 中的 `ifdown`）。

2.

如果 root 分区不是 NFS 或 iSCSI，则在接口上执行 bounce 测试。另外，所有其他没有测试

的接口都会被关闭，以确保流量通过被测试的接口路由。

3. 如果 root 分区是 NFS 或 iSCSI 挂载，则跳过对负责 iSCSI 或 NFS 连接的接口的 bounce 测试，并且所有其他接口（除处理 iSCSI 或 NFS 连接除外）将被关闭。
4. 测试文件在位置 /dev/urandom 处创建，并且使用 NIC 的速度调整其大小。
5. **TCP 和 UDP 测试 - 测试使用 iperf 工具：**
 - a. 测试测试服务器与测试下的主机之间的 TCP 延迟。测试检查系统是否运行到任何操作系统超时，如果这样做，则会失败。
 - b. 测试测试服务器与测试下的主机之间的带宽。对于有线设备，建议速度接近理论上最大。
 - c. 测试服务器与测试下主机之间的 UDP 延迟。测试检查系统是否运行到任何操作系统超时，如果这样做，则会失败。
6. **文件传输测试 - 测试使用 SCP 将文件从测试下的主机传输到远程系统或测试服务器，然后在测试下将其传输回主机，以检查传输是否正常工作。**
7. **ICMP (ping)测试 - 脚本会导致 ping 填充默认数据包大小，以确保系统出现什么操作失败（系统不应该重启或重置，或者指示无法与 ping 填充相关的任何内容）。发送 5000 数据包并预期是 100% 的成功率。测试重试 5 次，以获得可接受的成功率。**
8. 最后，测试会将所有接口返回到执行测试时的原始状态（活跃或不活跃）。

准备测试有线设备

您可以在每个测试中运行中测试您想要的网络设备。

开始前：

- 确保以原生（最大）速度连接每个设备，否则测试会失败。
- 确保测试服务器已启动并在运行。
- 确保每个网络设备具有静态或动态分配的 IP 地址。
- 确保打开了多个防火墙端口，供 iperf 工具运行 TCP 和 UDP 子测试。



注意

默认情况下，端口 52001-52101 被打开。如果要更改默认端口，更新 `/etc/rhcert.xml` 配置文件中的 `iperf-port` 和 `total-iperf-ports` 值。

Example:

```
<server listener-port="8009" iperf-port="52001" total-iperf-ports="100">
```

如果没有打开防火墙端口，测试会提示在测试运行时打开防火墙端口。

可分区网络

测试通过检查全速度和分区功能的数据传输来检查任何网络设备支持分区。

根据 NIC 的性能运行测试：

- 如果 NIC 在分区时以完整速度运行，则配置 NIC 以原生速度运行的分区，并在该配置中执行网络测试。
- 如果 NIC 在分区时没有以完整速度运行，则先运行测试两次 - 第一次运行测试，无需分区查看全速操作，第二次运行它，使其启用了分区来查看分区功能。



注意

红帽建议为您的分区配置选择 **1Gb/s** 或 **10Gb/s**，以便它符合现有网络速度测试。

准备测试无线以太网设备

根据被测试的无线卡，您连接到的无线接入点必须具有执行 **WirelessG**、**WireN**、**WireAC**、**WirelessAX**、**WiFi6** 和 **WiFi6E** 网络测试的能力。

执行测试

网络测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的网络测试名称。

```
rhcert-run
```

表 A.3. 手动添加并运行测试

速度类型	手动添加以太网测试的命令	手动运行以太网测试的命令
1GigEthernet	<pre>rhcert-cli plan --add --test 1GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 1GigEthernet --server <test server IP addr></pre>
10GigEthernet	<pre>rhcert-cli plan --add --test 10GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 10GigEthernet --server <test server IP addr></pre>
20GigEthernet	<pre>rhcert-cli plan --add --test 20GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 20GigEthernet --server <test server IP addr></pre>
25GigEthernet	<pre>rhcert-cli plan --add --test 25GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 25GigEthernet --server <test server IP addr></pre>
40GigEthernet	<pre>rhcert-cli plan --add --test 40GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 40GigEthernet --server <test server IP addr></pre>

速度类型	手动添加以太网测试的命令	手动运行以太网测试的命令
50GigEthernet	<pre>rhcert-cli plan --add --test 50GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 50GigEthernet --server <test server IP addr></pre>
100GigEthernet	<pre>rhcert-cli plan --add --test 100GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 100GigEthernet --server <test server IP addr></pre>
200GigEthernet	<pre>rhcert-cli plan --add --test 200GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 200GigEthernet --server <test server IP addr></pre>
400GigEthernet	<pre>rhcert-cli plan --add --test 400GigEthernet --device <device name></pre>	<pre>rhcert-cli run --test 400GigEthernet --server <test server IP addr></pre>

将 `<device name>` 和 `<test server IP addr>` 替换为适当的值。

运行时

网络测试需要大约 2 分钟时间来测试每个基于 PCIe、gigabit、有线以太网卡，以及所需的可支持测试，整个运行时大约会一分钟。???

其他资源

- 有关剩余的测试功能的更多信息，请参阅 [以太网测试](#)。

A.31. NETWORKMANAGEABLECHECK

测试涵盖了什么

NetworkManageableCheck 测试针对系统中所有可用的网络接口运行。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试包含两个执行以下任务的子测试：

1. 检查 BIOS 设备名称，以确认接口是否遵循固件设置的术语。



注意

BIOS 设备名称验证仅在 x86 系统上运行。

2. 检查网络管理器是否管理该接口，以评估当前的网络管理状态。

执行测试

NetworkManageableCheck 测试是必须的。它通过自我检查和可支持的测试来计划和执行，以确保对网络接口进行全面的检查和验证。

运行时

完成测试需要大约 1 分钟时间。但是，测试的持续时间因系统的细节和接口数量而异。

A.32. 使用 FABRIC 测试的 NVME

Fabric 上的 NVMe（也称为 NVMe-oF 和非易失性内存表达）是一个协议规格，旨在使用 NVMe 协议将主机连接到网络光纤中的存储。

该协议旨在通过基于 NVMe 消息的命令实现主机计算机和目标固态存储设备或系统之间的数据传输。数据传输可以通过以太网或 InfiniBand 等方法传输。

A.32.1. nvme_infiniband

测试涵盖了什么

nvme_infiniband 测试通过 RDMA 网络验证 NVMe SSD 驱动器的访问和使用。Host Under Test 被配置为 NVMe 客户端，lab 代理系统被配置为 NVMe 目标。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试运行多个子测试：

1. 验证是否载入了必要的内核模块，并且 NVMe 客户端是否已连接到 NVMe 目标。
2. 通过运行发现、断开连接和连接命令，在 NVMe 目标和客户端之间建立并确认连接。
3. 检测用于连接到目标系统中存储设备的网络接口，并相应地从每种 **STORAGE** 和 **infiniband 连接** 测试中运行一种测试类型。

对于通过 Fabric 存储测试的 NVMe，测试在 NVMe 客户端系统上执行，但 NVMe 设备实际位于 NVMe 目标主机上。NVMe 客户端和服务器使用 RDMA 协议进行通信。

准备测试

在开始测试前，请确保：

- NVMe 目标和 NVMe 客户端系统会被正确配置，是 RDMA 网络的一部分。
- NVMe 客户端和 NVMe 目标运行相同的 RHEL 版本。否则，RHEL 9.0 上的 NVMe 客户端和 RHEL 8.5 上的 NVMe 目标之间的通信将导致与 **Invalid MNAN value 1024** 尝试 nvme 连接的错误。

执行测试

测试是非交互式的。目前，此测试只能通过 CLI 规划和执行。

运行时

运行此测试需要大约 15 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.32.2. nvme_iwarp

测试涵盖了什么

`nvme_iwarp` 测试通过 RDMA 网络验证 NVMe SSD 驱动器的访问和使用。在 RHEL 8 上运行测试支持。Host Under Test 被配置为 NVMe 客户端，lab 代理系统被配置为 NVMe 目标。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试运行多个子测试：

1. 验证是否载入了必要的内核模块，并且 NVMe 客户端是否已连接到 NVMe 目标。
2. 通过运行发现、断开连接和连接命令，在 NVMe 目标和客户端之间建立并确认连接。
3. 检测用于连接到目标系统中存储设备的网络接口，并从每种 **STORAGE** 和 **iwarp 连接** 测试运行一种测试类型。

对于通过 Fabric 存储测试的 NVMe，测试在 NVMe 客户端系统上执行，但 NVMe 设备实际位于 NVMe 目标主机上。NVMe 客户端和服务器使用 RDMA 协议进行通信。

准备测试

在开始测试前，请确保：

- **NVMe 客户端正在运行 RHEL 8.x 或 9.x。**
- **NVMe 目标和 NVMe 客户端系统会被正确配置，是 RDMA 网络的一部分。**
- **NVMe 客户端和 NVMe 目标正在运行相同的 RHEL 版本，否则，RHEL 9.0 上的 NVMe 客户端与 RHEL 8.5 上的 NVMe 目标之间的通信将导致错误，Invalid MNAN value 1024 attempting nvme connect。**

执行测试

测试是非交互式的。目前，此测试只能通过 CLI 规划和执行。

运行时

运行此测试需要大约 15 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.32.3. nvme_omnipath

测试涵盖了什么

nvme_omnipath 测试通过 RDMA 网络验证 NVMe SSD 驱动器的访问和使用。在 RHEL 8 上运行测试支持。Host Under Test 被配置为 NVMe 客户端，lab 代理系统被配置为 NVMe 目标。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试运行多个子测试：

1. 验证是否载入了必要的内核模块，并且 NVMe 客户端是否已连接到 NVMe 目标。
2. 通过运行发现、断开连接和连接命令，在 NVMe 目标和客户端之间建立并确认连接。
3. 检测用于连接到目标系统中存储设备的网络接口，并相应地从每种 **STORAGE** 和 **omnipath 连接** 测试运行一种测试类型。

对于通过 Fabric 存储测试的 NVMe，测试在 NVMe 客户端系统上执行，但 NVMe 设备实际位于 NVMe 目标主机上。NVMe 客户端和服务器使用 RDMA 协议进行通信。

准备测试

在开始测试前，请确保：

- NVMe 客户端正在运行 RHEL 8.x 或 9.x。
- NVMe 目标和 NVMe 客户端系统会被正确配置，是 RDMA 网络的一部分。
- NVMe 客户端和 NVMe 目标正在运行相同的 RHEL 版本，否则，RHEL 9.0 上的 NVMe 客户端与 RHEL 8.5 上的 NVMe 目标之间的通信将导致错误，`Invalid MNAN value 1024 attempting nvme connect.`

执行测试

测试是非交互式的。目前，此测试只能通过 CLI 规划和执行。

运行时

运行此测试需要大约 15 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.32.4. nvme_roce

测试涵盖了什么

`nvme_roce` 测试验证通过 RDMA 网络使用 NVMe SSD 驱动器的访问和使用。在 RHEL 8 上运行测试支持。Host Under Test 被配置为 NVMe 客户端，lab 代理系统被配置为 NVMe 目标。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试运行多个子测试：

1. 验证是否载入了必要的内核模块，并且 NVMe 客户端是否已连接到 NVMe 目标。
2. 通过运行发现、断开连接和连接命令，在 NVMe 目标和客户端之间建立并确认连接。
3. 检测用于连接到目标系统中存储设备的网络接口，并相应地从每种 **STORAGE** 和 **RoCE** 连接测试中运行一种测试类型。

对于通过 **Fabric** 存储测试的 NVMe，测试在 NVMe 客户端系统上执行，但 NVMe 设备实际位于 NVMe 目标主机上。NVMe 客户端和服务器使用 RDMA 协议进行通信。

准备测试

在开始测试前，请确保：

- NVMe 客户端正在运行 RHEL 8.x 或 9.x。
- NVMe 目标和 NVMe 客户端系统会被正确配置，是 RDMA 网络的一部分。
- NVMe 客户端和 NVMe 目标正在运行相同的 RHEL 版本，否则，RHEL 9.0 上的 NVMe 客户端与 RHEL 8.5 上的 NVMe 目标之间的通信将导致错误，`Invalid MNAN value 1024 attempting nvme connect.`

执行测试

测试是非交互式的。目前，此测试只能通过 CLI 规划和执行。

运行时

运行此测试需要大约 15 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.32.5. nvme_tcp

测试涵盖了什么

nvme_tcp 测试通过 TCP 网络验证 NVMe SSD 驱动器的访问和使用。测试目前作为技术预览提供，并支持在 RHEL 8 中运行。Host Under Test 被配置为 NVMe 客户端，lab 代理系统被配置为 NVMe 目标。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试运行多个子测试：

1. 验证是否载入了必要的内核模块，并且 NVMe 客户端是否已连接到 NVMe 目标。
2. 通过运行发现、断开连接和连接命令，在 NVMe 目标和客户端之间建立并确认连接。
3. 检测用于连接到目标系统中存储设备的网络接口，并从每种 **STORAGE** 和 **NETWORK** 测试运行一种测试类型。

对于通过 Fabric 存储测试的 NVMe，测试在 NVMe 客户端系统上执行，但 NVMe 设备实际位于 NVMe 目标主机上。NVMe 客户端和服务器使用 TCP 协议进行通信。

准备测试

在开始测试前，请确保：

- **NVMe 客户端正在运行 RHEL 8.x。**
- **NVMe 目标和 NVMe 客户端系统会被正确配置，是 RDMA 网络的一部分。**



注意

通过 TCP 的 NVMe 的默认 TCP 端口号为 8009。RDMA 上的 NVMe 的默认 TCP 端口号为 4420。您可以使用任何与其他当前应用程序冲突的 TCP 端口号。如果存在端口冲突，请使用不同的 TCP 端口号重新配置 NVMe 端口号 8009。

执行测试

测试是非交互式的。目前，此测试只能通过 CLI 规划和执行。

运行时

运行此测试需要大约 10 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.33. OMNIPATH 连接

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

Omnipath Connection 测试运行以下子tests，以确保使用基准功能（如果适用），从测试的 onset 下拉菜单中选择的 IP 地址：

1. **ping test** - 从 HUT 上测试的设备的起始 IP 地址运行 ping 到测试服务器的所选 IP 地址。
2. **Rping test** - 在测试服务器上运行 rping 和使用所选测试服务器 IP 地址的 HUT，然后比较结果以验证其运行完成。
3. **Rcopy test** - 在测试服务器和 HUT 上运行 rcopy，发送随机生成的文件并在测试服务器上比较 md5sums，验证是否成功传输。
4. **RDMA-ndd 服务测试** - 验证停止、启动和重启服务命令可以正常工作。
5. **opensm 服务测试** - 按预期停止、启动和重启服务命令功能。
6. **LID 验证测试** - 验证是否设置了设备的 LID，而不是默认值。
7. **链接速度测试** - 证明检测到的链接速度为 100Gb。
8. **Smpquery test** - 使用设备和端口在测试服务器上运行 smpquery，以通过光纤注册了另一个验证设备/端口。
9. **ib_write_bw test**

从 HUT 运行 **ib_write_bw** 到测试服务器的所选 IP 地址，以测试 Omnipath 写入带宽，并验证它是否可以访问所需的带宽。在带宽测试过程中调整 queue pair 参数，以达到接近行率的吞吐量。
10. **ib_read_bw test**

从 HUT 运行 **ib_read_bw** 到测试服务器的所选 IP 地址，以测试 Omnipath 读取带宽，并验证它是否可以访问所需的带宽。在带宽测试过程中调整 queue pair 参数，以达到接近行率的吞吐量。
11. **ib_send_bw test**

从 HUT 运行 `ib_send_bw` 到测试服务器的所选 IP 地址，以测试 Omnipath 发送带宽，并验证它是否可以访问所需的带宽。在带宽测试过程中调整 `queue pair` 参数，以达到接近行率的吞吐量。

准备测试

- 确保测试服务器和 HUT 是同一光纤上的独立计算机。您需要从红帽客户门户网页的 [Downloads](#) 部分在测试服务器上安装 `opa-basic-tools`。



注意

如果您在测试中使用带有 RHEL 7 主机的 RHEL 8 测试服务器，则必须手动启动 `httpd` 服务，因为测试套件不会自动启动此服务。

执行测试

这是一个交互式测试。运行以下命令，然后从显示的列表中选择合适的 `omnipath` 连接 测试名称。

```
rhcert-run
```

系统将提示您选择一个执行测试的 IP 地址（测试服务器的 IP 地址）。在您运行测试的 HUT 设备的同一光纤中，选择一个与设备对应的 IP 地址。

手动添加并运行测试

使用以下命令手动添加 `OmnipathConnectionTest` :

```
rhcert-cli plan --add --test Omnipath --device <device name>_devicePort_<port number>
```

使用以下命令手动运行 `OmnipathConnectionTest` :

```
rhcert-cli run --test Omnipath --server <test server IP addr>
```

运行时

运行此测试需要少于 10 分钟。

其他资源

- 有关 InfiniBand 和 RDMA 的更多信息，[请参阅了解 InfiniBand 和 RDMA 技术](#)。

A.34. POWER_STOP

测试涵盖了什么

启用时，**Suspend-to-Idle** 状态允许处理器在系统挂起时处于最深空闲状态。它冻结用户空间，并将所有 I/O 设备置于低电源状态，从而节省系统上的功耗。

power_stop 测试旨在验证是否在 ppc64le CPU 架构机器上（特别是基于 Power9 的系统上启用这些 Stop（或 idle）状态可以正常工作。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试使用 **lsprop** 命令收集特定系统支持的所有 idle-stop 状态的信息，以及 **cpupower** 命令启用和禁用这些状态。如果启用或未启用，测试会观察每个 cpu idle 状态的用量和持续时间计数器递增。

成功标准：

在启用前后，更改 stop 状态的 usage 和 duration 参数值。

- **PASS**：如果每个状态都增加其计数器参数值
- **警告**：如果有任何一个状态无法增加其计数器参数值
- **FAIL**：如果任何状态没有增加其计数器

- **预览：任何其他未知问题**

准备测试

如果 Host Under Test (HUT) 满足以下要求，则会自动计划此测试：

- **HUT 运行其中一个受支持的 RHEL 版本。**
- **底层架构是 ppc64le**
- **CPU 型号是 POWER9**



注意

此测试不受支持，并在任何其他 RHEL 版本和架构上执行时失败。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `power_stop` 测试名称。

```
rhcert-run
```

运行时

测试需要超过五分钟的时间才能完成，但可能会因 CPU Idle Stop 状态的数量而异。

A.35. PROFILER

`profiler` 测试从 Host Under Test 收集性能指标，并确定从 RHEL 内核支持的软件或硬件性能监控单元 (PMU) 收集指标。如果指标基于硬件，测试会进一步确定 PMU 是否只包括每个内核计数器，或者包括每个软件包计数器。`profiler` 测试被分为三个测试，`profiler` `_hardware_core`、`profiler` `_hardware_uncore` 和 `profiler` `_software`。

A.35.1. profiler_hardware_core

测试涵盖了什么

`profiler_hardware_core` 测试通过检查循环事件来使用每个核心计数器基于硬件来收集性能指标。核心事件测量处理器内核的功能，如 L2 缓存。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

如果找到核心硬件事件计数器，并在 `/sys/devices` 目录中的 `cpu*cycles` 文件找到了测试，方法是运行 `find /sys/devicesAttr -type f -name 'cpu*cycles'` 命令。

测试执行多个命令来积累 'cycle' 事件示例，检查是否检测到了 'cpu cycle' 事件，并检查是否收集了样本。



注意

此测试并不适用于详细，它不会测试给定处理器可能或没有的每个可能的核心计数器事件。

准备测试

运行此测试没有特殊要求。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `profiler_hardware_core` 测试名称。

```
rhcert-run
```

运行时

测试需要大约 30 秒。任何其它强制或所选的测试都会添加到整个运行时。

A.35.2. profiler hardware uncore

测试涵盖了什么

`profiler hardware uncore` 测试使用基于硬件的软件包范围计数器收集性能指标。非核心事件测量位于核心之外的处理器的功能，但位于软件包中，例如内存控制器。

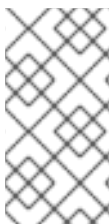
支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

如果找到了未核心硬件事件计数器，则计划测试。如果测试找到任何非核心事件并收集任何事件的统计信息，则测试通过。如果发现未核心事件但没有收集统计信息，则测试会失败，因为不支持这些事件。

测试执行多个命令，以收集未核心事件列表和未核心事件统计信息。



注意

此测试并不适用于详细，它不会测试给定处理器可能或未具有的所有可能的非内核计数器事件。

准备测试

运行此测试没有特殊要求。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 `profiler hardware uncore` 测试名称。

```
rhcert-run
```

运行时

测试需要大约 30 秒。任何其它强制或所选的测试都会添加到整个运行时。

A.35.3. profiler_software

测试涵盖了什么

profiler_software 测试通过检查 **cpu_clock** 事件来使用基于软件的计数器来收集性能指标。

软件计数器可以使用此测试进行认证。但是，对于具有高性能要求的客户，此测试可能会限制。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

如果没有找到核心硬件事件计数器，则计划测试。

测试执行多个命令来积累 **cpu-clock** 事件示例，检查是否检测到 **cpu-clock** 事件，并检查样本是否已收集。

准备测试

运行此测试没有特殊要求。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择合适的 **profiler_software** 测试名称。

```
rhcert-run
```

运行时

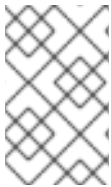
测试需要大约 30 秒。任何其它强制或所选的测试都会添加到整个运行时。

A.36. REALTIME

测试涵盖了什么

实时测试涵盖了通过两组测试运行 Red Hat Enterprise Linux for Real Time 的系统测试：一个用于查找基于系统管理的执行延迟，另一个用于确定服务计时器事件的延迟。

另外，对于 RHEL 8 和 RHEL 9，测试可确保为内务保留内核，而不是完全利用它们。



注意

该测试只在运行红帽内核的系统上进行计划。

支持的 RHEL 版本

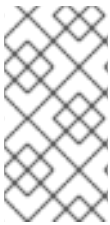
- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试的第一个部分加载一个名为 `hwlat_detector.ko` 的特殊内核模块。此模块会创建一个内核线程，它轮询 **Timestamp Counter Register (TSC)**，在连续读取超过指定阈值之间查找间隔。连续 TSC 读取中的差距意味着系统在读取和执行其他代码之间中断，通常是系统 BIOS 定义的系统管理模式(SMM)代码。

测试的第二部分会启动名为 `cyclictst` 的程序，它为每个 CPU 启动测量线程（以高实时优先级运行）。这些线程具有执行以下计算的句点(100 微秒)：

1. 获取时间戳(t1)
2. 周期休眠
3. 获取第二个时间戳(t2)
4. $latency = t2 - (t1 + period)$
5. goto 1



注意

延迟是理论唤醒时间($t1+period$)和实际唤醒时间($t2$)之间的时间差异。每个测量线程跟踪最小、最大和平均延迟，并报告每个数据点。

虽然 `cyclictest` 运行，`rteval` 会启动一对系统负载，但一个是并行的 `linux` 内核编译，另一个是名为 `hackbench` 的调度程序基准。

运行完成后，`rteval` 对数据点执行统计分析，计算含义、模式、媒体、差异以及标准偏差。

另外，对于 RHEL 8 和 RHEL 9，测试会检查 `/sys/devices/system/cpu/isolated` 中是否配置了隔离 CPU，并且 `tuned` 版本包括对 `isolated_cores` 的初始自动设置（大于或等于 2.19.0）的支持。它还检查 `realtime tuned` 配置集是否活跃。如果有任何检查失败，测试会在继续执行前发出警告。

准备测试

- 在将系统添加到认证之前，安装并引导 `realtime kernel-rt` 内核。命令将检测正在运行的内核是实时的，并将调度实时测试以运行。
- 对于 RHEL 8 和 RHEL 9，`tuned` 版本大于或等于 2.19.0，请选择 `tuned` 配置集作为 `realtime` 并重启系统。



注意

如果您需要 **realtime** 调优帮助，则必须提供对您的系统的访问权限，以允许进行必要的更改，包括修改 BIOS。



注意

新安装的内核继承之前配置的内核中的内核命令行参数。如需更多信息，请参阅 [更改所有引导条目的内核命令行参数](#)。

执行测试

实时测试是非交互式的。运行以下命令，然后从显示的列表中选择合适的实时测试名称。

```
rhcert-run
```

只有在系统运行 **rt-kernel** 时，测试才会显示。

运行时

系统管理模式在两个小时内运行，计时器事件分析会在所有计算机上运行 12 小时。所需的可支持测试将大约一分钟时间添加到整个运行时。

A.37. REBOOT

测试涵盖了什么

重启测试确认系统在系统提示时能够重新引导。目前不需要认证。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试发出 `shutdown -r 0` 命令，以便立即重启系统，且无延迟。

准备测试

通过关闭任何正在运行的应用程序，确保在运行此测试前重新引导系统。

执行测试

重启测试是交互式的。运行以下命令，然后从显示的列表中选择适当的重启测试名称。

```
rhcert-run
```

当您到达测试程序的重启部分时，会要求您重新启动？如果您准备好执行测试，请回答 `y`。测试服务器将重新启动，并在恢复后，测试服务器将验证重新引导是否成功完成。

A.38. ROCE 连接

测试的作用

RoCE Connection 测试运行以下子测试，以确保在适当的时候使用基准功能，这是测试之下的下拉菜单中选择的 IP 地址：

1. **ping test** - 从 HUT 上测试的设备的起始 IP 地址运行 ping 到测试服务器的所选 IP 地址。
2. **Rping test** - 在测试服务器上运行 rping 和使用所选测试服务器 IP 地址的 HUT，然后比较结果以验证其运行完成。
3. **Rcopy test** - 在测试服务器和 HUT 上运行 rcopy，发送随机生成的文件并在测试服务器上比较 md5sums，验证是否成功传输。
4. **ethtool test** - 运行在检测到的 roce 设备的 net 设备传递的 ethtool 命令。
5. **ib_write_bw test**

从 HUT 运行 `ib_write_bw` 到测试服务器的所选 IP 地址，以测试 RoCE 写入带宽，并验证它是否可以访问所需的带宽。

6.

`ib_read_bw` test

从 HUT 运行 `ib_read_bw` 到所选测试服务器的 IP 地址，以测试 RoCE 读取带宽，并验证它是否可以访问所需的带宽。

7.

`ib_send_bw` test

从 HUT 运行 `ib_send_bw` 到所选测试服务器的 IP 地址，以测试 RoCE 发送带宽，并验证它是否可以访问所需的带宽。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

准备测试

- 确保测试服务器和 HUT 是同一光纤上的独立计算机。

执行测试

这是一个交互式测试。运行以下命令，然后从显示的列表中选择适当的 RoCE 连接 测试名称。

```
rhcrt-run
```

系统将提示您选择一个执行测试的 IP 地址（测试服务器的 IP 地址）。在您运行测试的 HUT 设备的同一光纤中，选择一个与设备对应的 IP 地址。

表 A.4. 手动添加并运行测试

速度类型	手动添加 RoCEConnection Test 的命令	手动运行 RoCEConnection Test 的命令
10GigRoCE	<pre>rhcert-cli plan --add --test 10GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 10GigRoCE --server <test server IP addr></pre>
20GigRoCE	<pre>rhcert-cli plan --add --test 20GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 20GigRoCE --server <test server IP addr></pre>
25GigRoCE	<pre>rhcert-cli plan --add --test 25GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 25GigRoCE --server <test server IP addr></pre>
40GigRoCE	<pre>rhcert-cli plan --add --test 40GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 40GigRoCE--server <test server IP addr></pre>
50GigRoCE	<pre>rhcert-cli plan --add --test 50GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 50GigRoCE --server <test server IP addr></pre>
100GigRoCE	<pre>rhcert-cli plan --add --test 100GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 100GigRoCE --server <test server IP addr></pre>

速度类型	手动添加 RoCEConnection Test 的命令	手动运行 RoCEConnection Test 的命令
200GigRoCE	<pre>rhcert-cli plan --add --test 200GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 200GigRoCE --server <test server IP addr></pre>
400GigRoCE	<pre>rhcert-cli plan --add --test 400GigRoCE --device <device name>_devicePort_<port number>_netDevice_<net device></pre>	<pre>rhcert-cli run --test 400GigRoCE --server <test server IP addr></pre>

将 `<device name>` , `<port number >` , `<net device >` , 和 `<test server IP addr >` 替换为适当的值。

其他资源

- 有关 InfiniBand 和 RDMA 的更多信息, [请参阅了解 InfiniBand 和 RDMA 技术。](#)

A.39. SATA

SATA 测试涵盖的内容

当今的系统中有很多不同类型的持久的 on-line 存储设备。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

SATA 测试旨在测试在 udev 数据库中报告 ID_TYPE "disk" 的任何内容。此测试适用于 SATA 驱动器。如果出现以下情况，则计划 hwcert/storage/SATA 测试：

- **任何磁盘的控制器名称都提到 SATA 或**
- **磁盘连接的主机的 lsscsi 传输以提及 SATA**

如果上述两个条件不符合，则计划针对检测到的设备进行存储测试。

其他资源

- 有关 **测试的作用和 准备测试** 的更多信息，请参阅 [STORAGE](#)。

A.40. SATA_SSD

SATA_SSD 测试涵盖的内容

如果测试确定感兴趣的存储单元是 SSD，其接口是 SATA，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

SATA_SSD 测试的作用

测试找到 SCSI 存储类型，并确定位置上连接的存储接口 `/sys/block/sdap/queue/rotational`。如果 SSD 的 `rotational` 位设为零，则计划测试。

以下是要打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关 **测试的作用和准备测试** 的更多信息，请参阅 [STORAGE](#)。

A.41. M2_SATA

M2_SATA 测试涵盖了什么

如果测试确定接口是 **SATA** 并通过 **M2** 连接连接，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

手动添加并运行测试

使用以下命令手动添加 **M2_SATA** 测试：

```
rhcert-cli plan --add --test M2_SATA --device host0
```

以下是要打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关测试的作用和准备测试的更多信息，请参阅 [STORAGE](#)。

A.42. U2_SATA

U2_SATA 测试涵盖的内容

如果测试确定接口是 **SATA** 并通过 **U2** 连接连接，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**

- **RHEL 9**

手动添加并运行测试

使用以下命令手动添加 **U2_SATA** 测试：

```
rhcert-cli plan --add --test U2_SATA --device host0
```

以下是要打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关测试的作用和准备测试的更多信息，请参阅 [STORAGE](#)。

A.43. SAS

SAS 测试覆盖的内容

当今的系统中有许多不同类型的持久的 on-line 存储设备。

支持的 RHEL 版本

- **RHEL 7**

- **RHEL 8**
- **RHEL 9**

测试的作用

SAS 测试旨在测试在 udev 数据库中报告 ID_TYPE 的 ID_TYPE "disk"。此测试适用于 SAS 驱动器。如果出现以下情况，则计划 hwcert/storage/SAS 测试：

- **任何磁盘的控制器名称都应提到 SAS，或者**
- **连接到的磁盘的主机的 lsscsi 传输应该提到 SAS**

如果上述两个条件不符合，则计划针对检测到的设备进行存储测试。

其他资源

- **有关 测试的作用和 准备测试 的更多信息，请参阅 [STORAGE](#)。**

A.44. SAS_SSD

SAS_SSD 测试覆盖的内容

如果测试确定感兴趣的存储单元是 SSD，其接口是 SAS，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

SAS_SSD 测试的作用

测试找到 SCSI 存储类型，并确定位置上连接的存储接口 `/sys/block/sdap/queue/rotational`。如果 SSD 的 `rotational` 位设为零，则计划测试。

以下是作为测试的一部分打印的设备参数值：

- `logical_block_size` - 用于处理该设备中的位置
- `physical_block_size` - 设备可以操作的小型单元
- `minimum_io_size` - 设备随机输入/输出的首选最小单位
- `optimal_io_size` - 它是设备用于流传输输入/输出的首选单位
- `alignment_offset` - 它是底层物理对齐的偏移值

其他资源

- 有关测试的作用和准备测试的更多信息，请参阅 [STORAGE](#)。

A.45. PCIE_NVME

PCle_NVMe 测试涵盖的内容

如果测试确定接口是 NVMe，并通过 PCIE 连接附加，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**

- **RHEL 9**

PCIe_NVMe 测试的作用

如果逻辑设备主机名字符串包含 "nvme[0-9]", 则计划此测试。

以下是要打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关 **测试的作用和准备测试** 的更多信息，请参阅 [STORAGE](#)。

A.46. M2_NVME

M2_NVMe 测试涵盖什么

如果测试确定接口是 NVMe，并通过 M2 连接附加，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**

- **RHEL 8**
- **RHEL 9**

手动添加并运行测试

使用以下命令手动添加 **M2_NVMe** 测试：

```
rhcert-cli plan --add --test M2_NVMe --device nvme0
```

以下是打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关 **测试的作用和准备测试** 的更多信息，请参阅 [STORAGE](#)。

A.47. U2_NVME

U2_NVMe 测试涵盖的内容

如果测试确定接口是 **NVMe**，并通过 **U2** 连接附加，则此测试将运行。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

手动添加并运行测试

使用以下命令手动添加 **U2_NVMe** 测试：

```
rhcert-cli plan --add --test U2_NVMe --device nvme0
```

以下是要打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关 **测试的作用和准备测试** 的更多信息，请参阅 [STORAGE](#)。

A.48. NVDIMM

NVDIMM 测试涵盖了什么

此测试与任何其他 **SSD 非轮转存储测试** 运行，并标识 **NVDIMM 存储设备**

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

在以下情况下为存储设备计划测试：

- 存在该磁盘设备的命名空间（非易失性内存设备）由 "ndctl list" 报告
- 它报告了 **sda** 的 "DEVTYPE" 等于 'disk'

以下是要打印为测试的一部分的设备参数值：

- **logical_block_size** - 用于处理该设备中的位置
- **physical_block_size** - 设备可以操作的小型单元
- **minimum_io_size** - 设备随机输入/输出的首选最小单位
- **optimal_io_size** - 它是设备用于流传输输入/输出的首选单位
- **alignment_offset** - 它是底层物理对齐的偏移值

其他资源

- 有关 **测试的作用** 和 **准备测试** 的更多信息，请参阅 [STORAGE](#)。

A.49. SR-IOV

测试的作用

SR-IOV 测试通过检查卡上是否支持 **SR-IOV** 功能，在**测试(HUT)**和**测试服务器上安装的主机上的 NIC 卡**。

测试基于单根 I/O 虚拟化(SR-IOV)技术，该技术使单个物理硬件设备能够在多个虚拟机或容器之间共享，从而提高 I/O 操作的**网络性能和效率**。

支持的 RHEL 版本

- **RHEL 9**

测试涵盖了什么

测试会检查 **x86_64** 系统上安装的 **NIC 卡**是否支持 **SR-IOV** 技术。

准备测试

在运行测试前：

- 在 **HUT** 和**测试服务器上安装 NIC 卡**。
- 确保 **NIC 卡**进入测试各自有两个端口。
- 确保**测试服务器和 HUT** 通过两个直接以太网电缆连接回测试，以便**测试成功通过**。
- 确保在系统的 **BIOS** 中启用 **Intel VT-d** 或 **AMD IOMMU** 和 **SR-IOV Global Enable** 参数。如需了解更多详细信息，请参阅系统的 **BIOS 配置手册**或**制造商提供的任何其他方法**。
- 确保在 **Device Settings** 下为 **NIC 卡**启用 **SRIOV**。如需了解更多详细信息，请参阅 **NIC 卡配置手册**或**制造商提供的其它方法**。

- 按照上述顺序，在 HUT 和测试服务器上安装以下 RPM。

1. 启用 epel 存储库，然后安装 beakerlib ：

```
# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

```
# yum install beakerlib
```

2. **SR-IOV**

- **配置巨页**

```
# grubby --args='intel_iommu=on iommu=pt default_hugepagesz=1G hugepagesz=1G hugepages=32' --update-kernel=$(grubby --default-kernel)
```

确保在配置完成后重启系统。

- 确保调配 HUT 并测试服务器。请参阅[使用 Cockpit 配置系统并运行测试](#)，或使用[CLI 配置系统并运行测试](#)。

执行测试

在 HUT:

1. 根据您的系统配置 `/etc/redhat-certification/sriov/nic_cert.conf` 编辑在这个路径中生成的配置文件



注意

您必须在运行 `provision` 命令后每次更新并保留配置文件备份。

2. 运行测试

```
# rhcert-run
```

在 HUT 上执行测试时，也会在测试服务器上以 **auto** 模式执行对应的测试运行。测试是非交互式的，并在后台运行。

运行时

测试需要大约 2 小时才能运行。任何其它强制或所选的测试都会添加到整个运行时。

A.50. STORAGE

存储测试涵盖了什么

当今的系统中有许多不同类型的持久的 on-line 存储设备。STORAGE 测试旨在测试在 udev 数据库中报告 ID_TYPE "disk" 的任何内容。这包括 IDE、SCSI、SATA、SAS 和 SSD 驱动器、PCIe SSD 块存储设备以及 SD 介质、xD 介质、MemoryStick 和 MMC 卡。test 计划脚本通过 udev 数据库读取，并查找满足上述条件的存储设备。找到后，它将记录该设备及其父级，并将其与任何其他记录设备的父级进行比较。它的目的是确保只测试具有唯一父级的设备。如果之前未看到父级，则会将设备添加到测试计划中。这可加快测试，因为每个控制器只有一个设备将根据策略指南进行测试。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

STORAGE 测试对具有唯一父级的所有存储设备执行以下操作：

1. 该脚本通过分区表来查找不在 LVM 或者软件 RAID 设备上的交换分区。如果找到，它将在 **swapoff** 中停用，并将该空间用于测试。如果没有交换，系统仍然可以测试驱动器（如果其完全为空（没有分区）。请注意，交换设备必须处于活动状态，以便测试可以正常工作（测试可读取 `/proc/swaps` 来查找交换分区），并且 swap 分区不能在任何基于软件的容器内（没有 LVM 或软件 RAID，但硬件 RAID 可以正常工作，因为它对系统不可见）。

2. 该工具会在设备上创建一个文件系统，无论是在空白驱动器上的交换分区中。
3. 文件系统已挂载，并且 `fio` 或 `dt` 命令用于测试该设备。`fio` 或 `dt` 命令是一个 I/O 测试程序，它是一个可以测试、读取和写入设备的通用测试工具。多组测试模式验证存储设备的功能。
4. 在挂载的文件系统测试后，文件系统会被卸载，并根据块设备执行 `dt` 测试，忽略文件系统。`dt` 测试使用 `"direct"` 参数来处理此操作。



注意

Storage 测试在 RHEL 7.4 和旧版本中使用 `dt` 软件包，在 RHEL 7.5 及更新的版本中使用 `fio` 软件包。

准备测试

您应该安装官方测试计划中列出的所有驱动器和存储控制器。对于多个存储选项，一次可以测试多个存储选项，也可以在单个运行时测试系统，或者每个存储设备可以单独安装，并自行运行存储测试。您可以决定每个测试的测试顺序和控制器数量。附加到系统的每个逻辑驱动器都必须在其它分区之外包含一个交换分区，或者完全为空。这是为了为测试提供创建文件系统并运行测试的位置。使用交换分区将导致测试更快，因为设备完全测试。它们几乎总是大于放置在驱动器上的交换分区。



注意

如果测试 SD 介质卡，请使用您可以获取的最快卡。虽然类 4 SD 卡可能需要 8 小时或更长时间来运行测试，但类 10 或 UHS 1/2 卡可以在 30 分钟或更短时间内完成测试运行。

在为官方测试计划选择存储设备时，审查团队运行的规则是“每个代码路径一个测试”。这意味着，我们想要查看使用控制器可以使用的每个驱动程序运行存储测试。同一控制器的多个驱动程序的场景通常涉及某些类型的 RAID 存储。当处于常规磁盘模式时，存储控制器通常使用一个驱动程序，在使用 RAID 模式时另一个驱动程序。有些情况下，根据它们所在的 RAID 模式，也会使用多个驱动程序。审查团队将分析所有存储硬件，以确定需要使用的驱动程序来满足所有测试要求。这就是为什么您可能看到在官方测试计划中列出的一次相同的存储设备。有关存储设备测试的完整信息，请参阅策略指南。

执行测试

存储测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 **STORAGE** 测试名称。

```
rhcert-run
```


运行时间，裸机

存储测试在 2013-era workstation 系统中安装的 6Gb/s SATA 硬盘驱动器大约需要 22 分钟。在 2013-era workstation 系统中安装的 6Gb/s SATA 固态驱动器上，同样的测试需要大约 3 分钟。所需的可支持测试将大约一分钟时间添加到整个运行时。

其他资源

- 有关适当交换文件大小的更多信息，请参阅红帽平台推荐的 [swap 大小是什么？](#)

A.51. 特殊键

测试涵盖了什么

特殊键 测试从系统集成键盘捕获各种输入事件。此测试只在有电池的系统中运行。

支持的 RHEL 版本

- **RHEL 8.6 及更新版本**
- **RHEL 9**

测试的作用

测试捕获以下内容：

- 与非 CPI 相关的信号，如卷上和下移、音量静默、显示后向和下移等。
- 按键发送与全局键盘快捷键关联的信号，如 `< Meta+E >`，这将打开文件浏览器。

执行测试

测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 **Special key** 测试名称。

```
rhcert-run
```

此测试需要捕获所有输入事件。在测试过程中，按设备上的所有非标准和多媒体键。

随时按 **Escape** 键以结束测试并看到一个键列表。如果测试的所有键都出现在列表中，则测试可以成功。

运行时

完成测试需要不到 5 分钟。任何其它强制或所选的测试都会添加到整个运行时。

A.52. SUPPORTABLE

测试涵盖了什么

可支持的测试 可收集测试下主机的基本信息(HUT)。红帽使用此信息来验证系统是否符合认证要求。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试有几个执行以下任务的子测试：

1. **确认 `/proc/sys/kernel/tainted` 文件包含零(0)，这表示内核没有污点。**
2. **使用 `rpm -V` 命令确认软件包验证显示没有修改任何文件。**
3. **确认 `rpm -qa 内核` 命令显示内核软件包的 `buildhost` 是红帽服务器。**

4. 记录 `/proc/cmdline` 文件中的引导参数。
5. 确认 `'rpm -V redhat-certification'` 命令显示没有对任何认证测试套件文件进行任何修改。
6. 确认 `lsmod` 命令显示的所有模块都显示在带有 `rpm -ql kernel` 命令的 `kernel` 文件列表中。
7. 确认所有模块都位于 **Kernel Application Binary Interface (kABI) stablelist** 中。
8. 确认模块 `vendor` 和 `buildhost` 是适当的红帽条目。
9. 确认内核是 Red Hat 次版本的 GA 内核。

子测试尝试使用 `redhat-certification` 软件包中的数据验证内核。如果没有内核，`subtest` 会尝试使用互联网连接来验证内核。

要使用互联网连接来验证内核，您必须配置 `HUT` 的路由和 `DNS` 解析来访问互联网或设置 `ftp_proxy=http://proxy.domain:80` 环境变量。

10. 检查内核报告的已知硬件漏洞。`subtest` 读取 `/sys/devices/system/cpu/vulnerabilities/` 目录中的文件，如果文件包含单词“Vulnerable”，则退出并显示警告。
11. 通过检查 `lscpu` 命令的输出确认系统是否有离线 CPU。
12. 确认系统中是否有 **Simultaneous Multithreading (SMT)** 可用、启用并激活。
13. 检查运行 **RHEL 8** 或更高版本的系统中是否有未维护的硬件或驱动程序。

未维护的硬件和驱动程序不再被定期测试或更新。红帽可能会修复严重的问题，包括安全问题，但您不能以任何计划节奏进行更新。

尽快替换或删除未维护的硬件或驱动程序。

14.

检查运行 RHEL 8 或更高版本的系统中是否有已弃用的硬件或驱动程序。

弃用的硬件和驱动程序仍然经过测试和维护，但计划在以后的发行版本中变为不维护并最终禁用。

尽快替换或删除已弃用的设备或硬件。

15.

检查运行 RHEL 8 或更高版本的系统中是否禁用了硬件。

RHEL 无法使用禁用的硬件。在再次运行测试前，替换或从您的系统中删除禁用的硬件。

16.

对软件 RPM 软件包运行以下检查：

●

检查 RPM 构建主机信息，以隔离非红帽软件包。

测试将要求您解释包含非红帽软件包的原因。红帽将检查原因，并单独批准或拒绝每个软件包。

●

检查已安装的 RPM 软件包是否来自产品中可用的红帽产品，且尚未修改。

红帽在 `rpm_verification_report.log` 文件中检查验证失败。您需要重新安装失败的软件包并重新运行测试。

17.

检查系统中是否有红帽和非红帽固件文件。它列出了非红帽文件（如果存在），并退出 REVIEW 状态。

18.

通过 `getconf PAGESIZE` 命令检查系统的页大小。

执行这些任务后，测试会收集 `sosreport` 和 `dmidecode` 命令的输出。

执行测试

`rhcert` 工具会在测试套件的每个运行过程中自动运行可支持测试。可支持的测试在任何其他测试之前运行。

作为测试套件日志的一部分，需要可支持测试的输出。红帽将拒绝不包含可支持测试的输出的测试日志。

如果需要，使用以下命令手动运行测试：

```
$ rhcert-cli run -test supportable
```

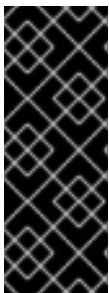
运行时

可支持的测试在 2013 年 3 月 1 分钟，单 CPU、3.3GHz、6 核或 12-thread Intel 工作站，运行有 8 GB 的 RAM，运行 Red Hat Enterprise Linux 6.4、AMD64 和 Intel 64，该指南中使用 Kickstart 文件安装。根据计算机速度和安装的 RPM 文件数量，时间将有所不同。

A.53. SUSPEND

测试涵盖了什么

(laptops only) 暂停测试涵盖了从 S3 睡眠状态（挂起到 RAM）的 suspend/resume，以及 S4 休眠状态（挂起到磁盘）的挂起/恢复。该测试还包括可保存更多能源的冻结（暂停 - s2idle）状态。此测试仅调度到具有内置电池的系统，如笔记本电脑。



重要

挂起 RAM 和挂起磁盘功能是笔记本电脑的基本特性。因此，我们计划在所有认证测试开始时进行自动暂停测试，并在笔记本电脑上运行。这样可确保所有硬件功能通常被恢复。测试将始终在笔记本电脑上运行，就像可支持的测试一样，无论计划什么测试是什么。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试查询 `/sys/power/state` 文件，并确定硬件支持哪些状态。如果在文件中看到“mem”，它会调度 S3 睡眠测试。如果在文件中看到“磁盘”，它会调度 S4 休眠测试。如果同时看到这两者，它会调度两者。以下是支持 S3 和 S4 状态的系统的步骤。如果您的系统不支持这两个类型，它将只运行与支持类型相关的测试。

RHEL 8 和 RHEL 9 上的挂起状态在 `/sys/power/state` 文件中写入。RHEL 7 改为使用 `pm-utils` 命令。

- 如果支持 S3 睡眠状态，脚本将使用 `pm-suspend` 命令挂起到 RAM。测试器在系统睡眠后唤醒了系统，脚本会检查 `pm-suspend` 的退出代码，以验证系统是否正确中断。然后，测试将继续测试。
- 如果支持 S4 休眠功能，脚本将使用 `pm-suspend` 命令挂起到磁盘。测试器会在系统休眠后唤醒系统，脚本会检查 `pm-suspend` 的退出代码，以验证系统是否正确中断。然后，测试将继续测试。
- 如果支持 S3 sleep，则测试程序会被提示按手动调用的键(`kbd:[Fn]+kbd:[F-key]` 组合或专用 `kbd:[Sleep]` 键) (如果存在这个键)。测试器在系统睡眠后唤醒了系统，脚本会检查 `pm-suspend` 的退出代码，以验证系统是否正确中断。然后，测试将继续测试。如果系统没有 `suspend` 密钥，可以跳过本节。
- 如果支持 S4 休眠，则测试程序会被提示按手动调用的键(`kbd:[Fn]+kbd:[F-key]` 组合或专用 `kbd:[Hibernate]` 键)。测试器会在系统休眠后唤醒系统，脚本会检查 `pm-suspend` 的退出代码，以验证系统是否正确中断。然后，测试将继续测试。如果系统没有 `suspend` 密钥，可以跳过本节。

准备测试

确保在安装系统时创建了足够大的交换文件来保存 RAM 的内容。在 `Host Under Test` 中必须存在某人，才能唤醒挂起和休眠状态。

执行测试

`suspend` 测试是交互式的。运行以下命令，然后从显示的列表中选择合适的 暂停 测试名称。

```
rhcert-run
```

测试将提示 挂起？回答 Yes 以暂停笔记本电脑。测试服务器将在发送 `suspend` 命令后显示等待响

应。检查笔记本电脑，并确认它已完成暂停，然后按 **power** 按钮或任何可唤醒它的任何键。测试服务器将持续监控测试下的主机，以查看它是否已被忽略。完成测试服务器 GUI 后，测试服务器 GUI 将显示问题 **Has resume?**。按 **Yes** 或 **No** 按钮告知测试服务器发生的情况。

然后，服务器将继续休眠测试。同样，单击 **suspend?** 问题的 **Yes** 按钮，将笔记本电脑置于休眠模式。

测试服务器将在发送 **hibernate** 命令后显示等待响应。检查笔记本电脑，并确认它已完成休眠，然后按 **power** 按钮或任何可以从休眠中唤醒的任何其他键。测试服务器将持续监控 **Host Under Test**，以查看它是否已强制使用。完成测试服务器 GUI 后，测试服务器 GUI 将显示问题 **已完成?**。按 **Yes** 或 **No** 按钮告知测试服务器发生的情况。

接下来，测试服务器将询问您是否有键盘键，该密钥将导致 **Host Under Test** 挂起。如果存在，请单击问题中的 **Yes** 按钮，此系统有一个功能密钥(Fn)来挂起系统到 **mem?**。按照上述步骤，验证挂起和唤醒系统以继续测试。

最后，测试服务器将询问您是否有键盘键，该密钥将导致 **Host Under Test** 休眠。如果存在，请单击问题中的 **Yes** 按钮，此系统有一个功能密钥(Fn)来挂起系统到磁盘？按照上述步骤，验证休眠和唤醒系统，以继续使用您计划的任何其他测试。

运行时

在带有 **4GB RAM** 和非 **SSD** 硬盘驱动器的 **2012-era** 笔记本电脑上，暂停测试需要大约 **6** 分钟。这是完整的测试，包括基于 **pm-suspend** 和基于功能键的挂起和休眠运行的时间。具体时间取决于笔记本电脑可以写入磁盘的速度、安装的 **RAM** 的数量和速度，以及笔记本电脑通过功能键进入挂起和休眠状态的能力。所需的 **可支持** 测试将大约一分钟时间添加到整个运行时。

其他资源

- 有关适当交换文件大小的更多信息，请参阅红帽平台推荐的 [swap 大小是什么？](#)

A.54. 磁带

测试涵盖了什么

磁带测试涵盖了所有类型的磁带驱动器。此测试不会测试与驱动器关联的任何机器人。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试使用 `mt` 命令重新卷磁带，然后执行 `/usr` 目录的 `tar`，并将其存储在磁带上。`tar` 比较用于确定磁带上的数据是否与磁盘上的数据匹配。如果数据匹配，则测试会通过。

准备测试

将适当大小的磁带插入到驱动器中。

执行测试

磁带测试不是交互式的。运行以下命令，然后从显示的列表中选择合适的磁带测试名称。

```
rhcert-run
```

A.55. THUNDERBOLT3

测试涵盖了什么

Thunderbolt3 测试涵盖了来自热插拔和基本功能角度的 Thunderbolt 3 端口，确保连接到端口的 OS 和设备都可以正确添加和删除所有端口。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试的目的是确保系统中的所有 Thunderbolt 3 端口按预期工作。它要求提供可用 Thunderbolt3 端

口的数量，然后要求测试程序将 Thunderbolt 3 设备插入到每个端口上。测试会监视 Thunderbolt 3 设备附加和分离事件并记录它们。如果它同时检测到测试者输入的唯一端口数量的插件和拔出事件，则测试将传递。请注意，虽然 Thunderbolt 3 设备使用与 USB C 设备相同的物理连接器，但 USB C 设备不是 Thunderbolt 3 设备。如果使用 USB C 设备，则测试不会通过，包括声明与 Thunderbolt 3 端口兼容的 USB C 设备。此测试只能使用 Thunderbolt 3 设备。

准备测试

计算可用的 Thunderbolt3 端口，并有一个在测试过程中要使用的可用 Thunderbolt3 设备。

执行测试

Thunderbolt3 测试是交互式的。运行以下命令，然后从显示的列表中选择合适的 Thunderbolt3 测试名称。

```
rhcert-run
```

当系统提示时，输入系统中存在的可用 Thunderbolt3 端口的数量。系统将要求将 Thunderbolt3 设备插入到端口中，然后暂停直到测试程序按 **y** 继续为止。然后，系统会要求设备被取消插入，然后再次暂停，直到测试者按 **y** 继续。这些步骤对输入的端口数量重复。请注意，测试端口没有正确的或错误的顺序，但每个端口必须只测试一次。

运行时

Thunderbolt3 测试每 Thunderbolt3 端口大约需要 15 秒。这包括手动插入设备、扫描端口、拔出设备并再次扫描端口的时间。任何其它强制或所选的测试都会添加到整个运行时。

A.56. THUNDERBOLT4

测试涵盖了什么

Thunderbolt4 测试涵盖了来自热插拔和基本功能角度的 Thunderbolt 4 端口，确保附加到端口的 OS 和设备都可以正确添加和删除所有端口。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试的目的是确保系统中的所有 Thunderbolt 4 端口按预期工作。它要求提供可用 Thunderbolt4 端口的数量，然后要求测试程序将 Thunderbolt 4 设备插入并拔出每个端口。测试会监视 Thunderbolt 4 设备附加和分离事件并记录它们。如果它同时检测到测试者输入的唯一端口数量的插件和拔出事件，则测试将传递。请注意，虽然 Thunderbolt 4 设备使用与 USB C 设备相同的物理连接器，但 USB C 设备不是 Thunderbolt 4 设备。如果使用 USB C 设备，则测试不会通过，包括声明与 Thunderbolt 4 端口兼容的 USB C 设备。此测试只能使用 Thunderbolt 4 设备。此测试还验证主机和连接的设备之间的连接生成是否为 Thunderbolt 4。

准备测试

计算可用的 Thunderbolt4 端口，并有一个在测试过程中要使用的可用 Thunderbolt4 设备。

执行测试

Thunderbolt4 测试是交互式的。运行以下命令，然后从显示的列表中选择合适的 Thunderbolt4 测试名称。

```
rhcet-run
```

当系统提示时，输入系统中存在的可用 Thunderbolt4 端口的数量。系统将要求将 Thunderbolt4 设备插入到端口中，然后暂停直到测试者按 **y** 继续为止。然后，系统会要求设备被取消插入，然后再次暂停，直到测试者按 **y** 继续。这些步骤对输入的端口数量重复。请注意，测试端口没有正确的或错误的顺序，但每个端口必须只测试一次。

运行时

Thunderbolt4 测试每 Thunderbolt4 端口大约需要 15 秒。这包括手动插入设备、扫描端口、拔出设备并再次扫描端口的时间。任何其它强制或所选的测试都会添加到整个运行时。

A.57. USB_STORAGE

测试涵盖了什么

`usb_storage` 测试为现有存储测试添加了速度检测功能。`usb_storage` 测试组成：

- 用于检测连接的 USB 设备的版本的 `USB2_storage` 测试
- `USB3_storage` 测试检测连接的 USB 设备的版本和接口速度，并支持多种速度(5Gbps、10Gbps、20Gbps、40Gbps)

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试检测到连接到系统的 USB 设备的接口速度和版本，并相应地规划对应的测试。例如，如果检测到支持接口速度为 10Gbps 的 USB 3.0 设备，则将计划并执行 `USB3_10Gbps_Storage subtest`。

准备测试

确定 USB 存储设备已连接到系统。

执行测试

您可以选择运行测试的方法：

- 运行以下命令，然后从显示的列表中选择适当的 USB 测试名称。

```
rhcert-run
```

- 通过指定所需的测试名称来运行 `rhcert-cli` 命令。例如，

```
rhcert-cli run --test=USB3_10Gbps_Storage
```

其他资源

- 有关其余测试功能的更多信息，请参阅 [STORAGE](#)。

A.58. USB2

测试涵盖了什么

USB2 测试涵盖了从基本功能角度的 USB2 端口，确保操作系统可以访问所有端口。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试的目的是确保系统中的所有 USB2 端口按预期工作。它要求提供可用 USB2 端口的数量（减去用于键盘/移动使用的任何端口），然后要求测试程序将 USB2 设备插入每个端口。测试会监视附加和分离事件，并记录它们。如果它同时检测到测试者输入的唯一端口数量的插件和拔出事件，则测试将传递。

准备测试

计算可用的 USB2 端口，并具有可在测试过程中使用备用 USB2 设备。您可能需要从主板标头跟踪 USB 端口，以区分 USB2 和 USB3 端口。

执行测试

USB2 测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 USB2 测试名称。

```
rhcert-run
```

当系统提示时，输入系统中存在的可用 USB2 端口的数量。不要计算任何当前由键盘或鼠标使用的任何信息。系统将要求将测试 USB2 设备插入到端口中，然后暂停直到测试器按下 **y** 继续为止。然后，系统会要求设备被取消插入，然后再次暂停，直到测试者按 **y** 继续。这些步骤对输入的端口数量重复。请注意，测试端口没有正确的或错误的顺序，但每个端口必须只测试一次。

运行时

USB2 测试每个 USB2 端口大约需要 15 秒。这包括手动插入设备、扫描端口、拔出设备并再次扫描端口的时间。所需的可支持测试将大约一分钟时间添加到整个运行时。

A.59. USB3

测试涵盖了什么

USB3 测试涵盖了从基本功能角度的 USB3 端口，确保所有端口都可以被操作系统枚举、访问和热插。USB3 测试支持三个不同的基于速度的测试，每个测试都有 5Gbps、10Gbps 和 20Gbps。如果系统支持 USB3，则计划所有三个测试。每次测试成功时，将导致红帽生态系统目录中包含的相应功能进行认证。测试及其成功标准如下：

成功标准：

- **USB3_5Gbps** - 当设备传输速度为 5Gbps 时，测试将传递。
- **USB3_10Gbps** - 当设备传输速度为 10Gbps 时，测试将传递。
- **USB3_20Gbps** - 当设备传输速度为 20Gbps 时，测试将传递。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试的目的是确保系统中的所有 USB3 端口按预期工作。它要求提供可用 USB3 端口的数量（减去任何用于键盘/鼠标等），然后要求测试程序将 USB3 设备插入并拔出每个端口。测试会监视附加和分离事件，并记录它们。如果它同时检测到测试者输入的唯一端口数量的插件和拔出事件，则测试将传递。

准备测试

计算可用的 USB3 端口，并具有在测试过程中要使用的可用 USB3 设备。您可能需要从主板标头跟踪 USB 端口，以区分 USB2 和 USB3 端口。确保设备的行速度与测试的预期速度匹配，即 5Gbps、10 Gbps 或 20Gbps。

执行测试

USB3 测试是交互式的。 运行以下命令，然后从显示的列表中选择适当的 **USB3 测试名称**。

```
rhcert-run
```

当系统提示时，输入系统中存在的可用 **USB3 端口** 的数量。不要计算任何当前由键盘或鼠标使用的任何信息。系统将要求将测试 **USB3 设备** 插入到端口中，然后暂停直到测试器按下 **y** 继续为止。然后，系统会要求设备被取消插入，然后再次暂停，直到测试者按 **y** 继续。这些步骤对输入的端口数量重复。请注意，测试端口没有正确的或错误的顺序，但每个端口必须只测试一次。

运行时

USB3 测试 每个 **USB3 端口** 大约需要 **15 秒**。这包括手动插入设备、扫描端口、拔出设备并再次扫描端口的时间。所需的 **可支持 测试** 将大约一分钟时间添加到整个运行时。

A.60. USB4

测试涵盖了什么

USB4 测试 涵盖了从基本功能角度的 **USB4 端口**，确保所有端口都可以被操作系统枚举、访问和热插。USB4 测试支持两种不同的基于速度的测试，一个用于 **20Gbps**，一个用于 **40Gbps**。如果系统支持 **USB4**，则计划这两个测试。每次测试成功时，将导致红帽生态系统目录中包含的相应功能进行认证。测试及其成功标准如下：

成功标准

- **USB4_20Gbps** - 当设备传输速度为 **20Gbps** 时，测试将传递。
- **USB4_40Gbps** - 当设备传输速度为 **40Gbps** 时，测试将传递。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试的目的是确保系统中的所有 USB4 端口按预期工作。它要求提供可用 USB4 端口的数量（减去任何用于键盘/鼠标等），然后要求测试程序将 USB4 设备插入并拔出每个端口。测试会监视附加和分离事件，并记录它们。如果它同时检测到测试者输入的唯一端口数量的插件和拔出事件，则测试将传递。

准备测试：

计算可用的 USB4 端口，并具有在测试过程中要使用的可用 USB4 设备。您可能需要从主板标头跟踪 USB 端口，以区分 USB2、USB3 和 USB4 端口。确保设备的行速度与测试的预期速度匹配，即 20Gbps 或 40Gbps。

执行测试

USB4 测试是交互式的。运行以下命令，然后从显示的列表中选择适当的 USB4 测试名称。

```
rhcert-run
```

当系统提示时，输入系统中存在的可用 USB4 端口的数量。不要计算任何当前由键盘或鼠标使用的任何信息。系统将要求将测试 USB4 设备插入到端口中，然后暂停直到测试器按下 **y** 继续为止。然后，系统会要求设备被取消插入，然后再次暂停，直到测试者按 **y** 继续。这些步骤对输入的端口数量重复。请注意，测试端口没有正确的或错误的顺序，但每个端口必须只测试一次。

运行时

USB4 测试每个 USB4 端口大约需要 15 秒。这包括手动插入设备、扫描端口、拔出设备并再次扫描端口的时间。所需的 可支持 测试将大约一分钟时间添加到整个运行时。

A.61. VIDEO

测试涵盖了什么

对于 RHEL 7 和 RHEL 8，VIDEO 测试会检查主板上所有可移动或集成的视频硬件。选择设备以通过 PCI 类 ID 进行测试。具体来说，测试会在 `udev` 命令输出中检查具有 PCI 类的设备作为 **Display Controller**。

对于 RHEL 9，VIDEO 测试保持不变。但是，对于帧缓冲图形解决方案，会在它标识显示内核驱动程序是否用作帧缓冲，并使用 `glxinfo` 命令不支持直接渲染。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试运行多个子测试：

1. **check Connections** - 记录 `xrandr` 命令输出。此子测试是可选的，其失败不会影响整个测试结果。
2. **Set Configuration** - 检查必要的配置前提条件，如为下一个子测试设置显示深度、标志和配置。
3. **X Server Test** - 使用新配置文件启动另一个显示服务器，并运行 `glxgears`、轻量级 **MESA OpenGL** 演示程序来检查性能。
4. **log Module 和 Drivers** - 运行 `xdpyinfo` 以确定屏幕分辨率和颜色深度。此外，在测试开始时创建的配置文件也应该允许系统在最大分辨率能力下运行。

最后，测试使用 `grep` 在 `/var/log/Xorg.0.log` 日志文件中搜索，以确定正在使用的模块和驱动程序。

准备测试

- 确保系统中的监控器和视频卡可在每个 `pixel (bpp)` 为 24 位的解析时运行，颜色深度为 24 位。更高分辨率或颜色深度也可以接受。在 24 `bpp` 或更高版本上检查 `xrandr` 命令输出 `1024x768` 以确认。
- 如果您没有看到卡或监控组合可以生成的所有解析，请确保删除监视器和显卡之间的任何 **KVM** 交换机。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 VIDEO 测试名称。

```
rhcert-run
```

首先，测试系统屏幕将留空，然后显示 x11perf 测试程序中的一系列测试模式。测试完成后，它将返回到桌面或虚拟终端屏幕。

运行时

完成测试需要大约 1 分钟时间。任何其它强制或所选的测试都会添加到整个运行时。

A.62. VIDEO_PORTS

测试涵盖了什么

VIDEO_PORTS 测试检查系统中每个图形处理器的所有图形输出端口是否正常运行。

测试在具有一个或多个图形输出端口的机器上运行。还支持具有一个或多个嵌入式或附加图形处理器的机器，包括带有端口 wired 到tegral 面板的笔记本电脑。

如果没有检测连接到该端口的显示，则测试不会在端口上运行。

支持的 RHEL 版本

- **RHEL 9**

测试的作用

测试执行以下操作：

1. 测试通过它检测到的每个端口运行，这些端口连接了 monitor。
2. 然后测试会启动一个 glmark2 窗口，并提示您将测试窗口拖到每个连接的显示中。
- 3.

如果测试检测到未测试的其他端口，它会进入交互模式。它会提示您为每个未测试的端口附加一个显示，并重复测试。

4. 测试将继续在此循环中运行，直到它测试了所有检测到的端口，或者您指出未测试的端口可供客户使用。如果没有端口，测试会提示您输入说明。
5. 当循环退出时，如果所有端口都已测试，则测试会显示 **PASS** 结果，如果某些端口已被识别为不可用，则测试会显示 **REVIEW** 结果。

准备测试

- 为您的系统准备一组具有适当连接器的监控器。这包括内置监控器和至少一个外部监控器。
- 如果监控器小于端口，测试将运行循环，并允许您将显示连接到批处理中的端口。除了附加的每个外部监视器外，内置监控器还必须继续工作。
- 取决于物理端口，可能比物理端口更为电子，这意味着硬件支持的显示比用户可用的系统数量更高。测试开始与测试无关时，屏幕上显示的端口列表。
- 系统中可能存在比一次全部使用的物理端口。也可能存在 ghost 端口，如服务端口或 USB。您必须能够区分由于与其他端口不兼容或者由于其为 ghost 端口不兼容的端口，以及在根本不正运行的端口之间区分无法正常工作的端口。

执行测试

VIDEO_PORTS 测试是交互式的。在执行测试前，请将 **monitor** 连接到至少一个图形输出端口。

1. 置备系统：
 - a. 运行这个命令：

```
# rhcert-provision
```
 - b. 出现提示时，输入系统上保存的测试计划的路径。
 - c.

如有提示，请提供测试服务器的主机名或 IP 地址，以设置免密码 SSH。只有在您第一次添加新系统时，才会提示您。

2.

启动测试：

```
# rhcert-cli run --test VIDEO_PORTS
```



注意

测试首先列出一组内部显示，包括连接和断开连接。它们并不代表正在测试的物理端口。

3.

对于每个连接的图形输出端口，请按照以下步骤操作：

a.

等待测试识别端口。出现提示时，按任意键继续。

b.

此时会打开 `glmark2` 窗口。如果与活跃监控器不同，请将此窗口移到连接到端口的监控器。

`glmark2` 基准测量所识别显示上 OpenGL (ES) 2.0 性能的各个方面。该基准调用一系列镜像，它们测试不同的面面组合、单色和简易。

c.

等待 `glmark2` 窗口关闭。您将看到每次成功测试的 `glmark2` 分数和 `Test` 传递消息。

4.

对于每个未连接的图形输出端口，请按照以下步骤操作：

a.

出现提示时，将 `monitor` 连接到图形端口，并输入 `yes` 以继续。

在第一个提示中，您可以输入 `no` 结束 `GRAPHICS_PORTS` 测试。对于每个额外的提示，会显示计时器，为您提供连接监控器 20 秒。计时器在超时前重复三次。

b.

等待测试识别端口。出现提示时，按任意键继续。

c.

将 `glmark2` 窗口移到连接到端口的监控器。等待窗口关闭。

5.

如果没有要连接的额外端口，让计时器运行 60 秒，直到超时为止。如果所有端口都成功测试，则测试会以 **PASS** 结果退出。

6.

(可选) 将测试结果保存到日志文件中：

```
# rhcert-save
```

7.

导航到日志文件的位置，从浏览器访问日志文件。

运行时

测试时间因被测试的端口数量而异。每个端口需要大约 2-3 分钟才能进行测试。

影响测试时间的其他因素包括系统性能，如内存频率和 CPU。任何其它强制或所选的测试都会添加到整个运行时。

A.63. VIDEO_DRM

测试涵盖了什么

VIDEO_DRM 测试验证图形控制器，该控制器使用具有基本图形支持的原生 DRM 内核驱动程序。

测试将计划 if:

- 正在使用的显示驱动程序被识别为内核 `mode-setting` 驱动程序。
- 显示驱动程序不是帧缓冲。
- `glxinfo` 命令不支持直接渲染，OpenGL renderer 字符串是 LLVM pipe。

支持的 RHEL 版本

- **RHEL 9**

测试的作用

测试会验证图形控制器的功能，类似于 **VIDEO** 测试。

准备测试

- 确保系统中的监控器和视频卡可在每个 pixel (bpp) 为 24 位的解析时运行，颜色深度为 24 位。更高分辨率或颜色深度也可以接受。在 24 bpp 或更高版本上检查 `xrandr` 命令输出 `1024x768` 以确认。
- 如果您没有看到卡或监控组合可以生成的所有解析，请确保删除监视器和显卡之间的任何 KVM 交换机。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 **VIDEO_DRM** 测试名称。

```
rhcert-run
```

首先，测试系统屏幕将留空，然后显示 `x11perf` 测试程序中的一系列测试模式。测试完成后，它将返回到桌面或虚拟终端屏幕。

运行时

完成测试需要大约 1 分钟时间。任何其它强制或所选的测试都会添加到整个运行时。

A.64. VIDEO_DRM_3D

测试涵盖了什么

VIDEO_DRM_3D 测试验证图形控制器，该控制器使用具有加速图形支持的原生 **DRM** 内核驱动程序。

测试将计划 `if`:

- 正在使用的显示驱动程序被识别为内核 `mode-setting` 驱动程序。
- 显示驱动程序不是帧缓冲。
- 直接渲染受到 `glxinfo` 命令标识的，`OpenGL renderer` 字符串不是 `LLVM pipe`。

测试使用 Prime GPU 卸载技术来执行所有视频测试子测试。

支持的 RHEL 版本

- **RHEL 9**

测试的作用

测试会验证图形控制器的功能，类似于 **VIDEO** 测试。另外，测试运行以下子测试：

1. **Vulkaninfo 测试** - 记录 `vulkaninfo` 命令输出以收集 Vulkan 信息，如标识 GPU 的设备属性、每个 GPU 支持的 Vulkan 扩展、可识别的层、支持的镜像格式和格式属性。
2. **Glmak2 基准测试测试** - 运行 `glmark2` 命令，以根据 OpenGL 2.0 & ES 2.0 基准测试集生成分数，并确认 3D 功能。子测试使用不同的参数集执行实用程序两次，首先使用硬件渲染程序，之后使用软件呈现器执行实用程序。如果 `Hardware renderer` 命令运行的结果比软件更强，测试会成功通过，确认显示控制器具有更好的 3D 功能，否则会失败。

准备测试

- 确保系统中的监控器和视频卡可在每个 pixel (bpp) 为 24 位的解析时运行，颜色深度为 24 位。更高分辨率或颜色深度也可以接受。在 24 bpp 或更高版本上检查 `xrandr` 命令输出 `1024x768` 以确认。
- 如果您没有看到卡或监控组合可以生成的所有解析，请确保删除监视器和显卡之间的任何 KVM 交换机。

执行测试

测试是非交互式的。运行以下命令，然后从显示的列表中选择适当的 VIDEO_DRM_3D 测试名称。

```
rhcert-run
```

首先，测试系统屏幕将留空，然后显示 x11perf 测试程序中的一系列测试模式。测试完成后，它将返回到桌面或虚拟终端屏幕。

运行时

完成测试需要大约 1 分钟时间。任何其它强制或所选的测试都会添加到整个运行时。

A.65. WIRELESSG

测试涵盖了什么

WirelessG 测试在所有无线以太网连接中运行，最大连接速度为 802.11g。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

这是一个新的测试，它组合了现有的 wlan 和 network 测试。除了传递所有现有网络测试项目外，此测试还必须检测到 iw 报告的 "g" 链接类型，并演示要传递的最小吞吐量 22Mb/s。

其他资源

- 有关测试功能的其余部分的更多信息，请参阅 [网络](#)。

A.66. WIRELESSN

测试涵盖了什么

测试涵盖了什么

WirelessN 测试在所有无线以太网连接中运行，最大连接速度为 802.11n。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

这是一个新的测试，它组合了现有的 wlan 和 network 测试。除了传递所有现有网络测试项目外，此测试还必须检测到 iw 报告的"n"链接类型，并演示要传递的最小吞吐量 100Mb/s。

其他资源

- 有关测试功能的其余部分的更多信息，请参阅 [网络](#)。

A.67. WIRELESSAC

测试涵盖了什么

WirelessAC 测试在所有无线以太网连接中运行，最大连接速度为 802.11ac。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

这是一个新的测试，它组合了现有的 wlan 和 network 测试。除了传递所有现有网络测试项目外，此测试还必须检测到 iw 报告的"ac"链接类型，并演示要传递的最小吞吐量 300Mb/s。

其他资源

- 有关测试功能的其余部分的更多信息，请参阅 [网络](#)。

A.68. WIRELESSAX (SUPERSED BY WIFI6)

测试涵盖了什么

WirelessAX 测试在所有无线以太网连接上运行，最大连接速度为 802.11ax。

支持的 RHEL 版本

- RHEL 7
- RHEL 8
- RHEL 9

测试的作用

测试检测到 iw 报告的"ax"链接类型，并与带有 "wifi 6" 或 "AX" 的产品名称匹配，以确定该设备有 AX 支持。如果设备通过 Wireless AC 测试，并且展示了 1200 Mb/s 以便通过的最小吞吐量，则还计划对 Wireless AX 测试进行测试。此测试不会被自动规划，但可以通过 CLI 手动计划。相反，会自动规划 WiFi6 测试。

A.69. WIFI6

测试涵盖了什么

WiFi6 测试在所有无线以太网连接上运行，最大连接速度为 802.11ax。

支持的 RHEL 版本

- **RHEL 7**
- **RHEL 8**
- **RHEL 9**

测试的作用

测试检测到 `iw` 报告的"ax"链接类型，并与带有 "wifi 6" 或 "AX" 的产品名称匹配，以确定该设备是否具有 AX 支持。如果设备通过 Wireless AC 测试，并演示 1200 Mb/s 时要传递的最小吞吐量，则还计划使用 WiFi6 的测试。

A.70. WIFI6E

测试涵盖了什么

WiFi6E 测试在所有无线以太网连接上运行，最大连接速度为 802.11ax，利用 6GHz 频率带。

支持的 RHEL 版本

- **RHEL 8**
- **RHEL 9**

测试的作用

测试检测到 `iw` 报告的"ax"链接类型，并与包含"wifi 6E"或"AX"的产品名称匹配，以确定该设备是否具有 AX 支持。如果设备通过 Wireless AC 测试，并展示了要传递的 6000 Mb/s 的最小吞吐量，则还计划使用 WiFi6E 测试。

A.71. 手动添加并运行测试

在罕见的情况下，测试可能会因为硬件检测或其他问题解决硬件、操作系统或测试脚本的问题而无法规划。如果出现这种情况，您应该联系您的红帽支持联系人以获得进一步帮助。它们可能会要求您为问题创建一个支持问题单，然后解释了如何在 HUT 上使用 `rhcert-cli` 命令手动将测试添加到本地测试计划。您对本地图测试计划所做的任何修改都将发送到测试服务器，以便您可以继续在测试服务器上使用 Web 界面来运行测试。该命令运行如下：

```
# rhcert-cli plan --add --test=<testname> --device=<devicename> --udi-<udi>
```

此处使用的 `rhcert-cli` 命令的选项有：

- **`plan`** - 修改测试计划
- **`--add`** - 在测试计划中添加项
- **`--test=<testname>`** - 要添加的测试。测试名称如下：
 - **`hwcert/suspend`**
 - **`hwcert/audio`**
 - **`hwcert/battery`**
 - **`hwcert/lid`**
 - **`hwcert/usbbase/expresscard`**
 - **`hwcert/usbbase/usbbase/usb2`**
 - **`hwcert/usbbase/usbbase/usb3`**
 - **`hwcert/kdump`**
 - **`hwcert/network/Ethernet/100MegEthernet`**
 - **`hwcert/network/Ethernet/1GigEthernet`**

- ***hwcert/network/Ethernet/10GigEthernet***
- ***hwcert/network/Ethernet/40GigEthernet***
- ***hwcert/network/wlan/WirelessG***
- ***hwcert/network/wlan/WirelessN***
- ***hwcert/network/wlan/WirelessAC*** (仅适用于 Red Hat Enterprise Linux 7)
- ***hwcert/memory***
- ***hwcert/core***
- ***hwcert/cpuscaling***
- ***hwcert/fvtest/fv_core***
- ***hwcert/fvtest/fv_live_migration***
- ***hwcert/fvtest/fv_memory***
- ***hwcert/fvtest/fv_network***
- ***hwcert/fvtest/fv_storage***
- ***hwcert/fvtest/fv_pcie_storage_passthrough***

- *hwcert/fvtest/fv_pcie_network_passthrough*
- *hwcert/fvtest/fv_usb_storage_passthrough*
- *hwcert/fvtest/fv_usb_network_passthrough*
- *hwcert/fvtest/fv_cpu_pinning*
- *hwcert/profiler*
- *hwcert/storage*
- *hwcert/video*
- *hwcert/supportable*
- *hwcert/optical/bluray*
- *hwcert/optical/dvd*
- *hwcert/optical/cdrom*
- *hwcert/fencing*
- *hwcert/realtime*
- *hwcert/reboot*

- ***hwcert/tape***

- ***hwcert/rdma/Infiniband_QDR***

- ***hwcert/rdma/Infiniband_FDR***

- ***hwcert/rdma/Infiniband_EDR***

- ***hwcert/rdma/Infiniband_HDR***

- ***hwcert/rdma/Infiniband_Socket_Direct***

- ***hwcert/rdma/10GigRoCE***

- ***hwcert/rdma/20GigRoCE***

- ***hwcert/rdma/25GigRoCE***

- ***hwcert/rdma/40GigRoCE***

- ***hwcert/rdma/50GigRoCE***

- ***hwcert/rdma/100GigRoCE***

- ***hwcert/rdma/200GigRoCE***

- ***hwcert/rdma/10GigiWarp***

- ***hwcert/rdma/20GigiWarp***
- ***hwcert/rdma/25GigiWarp***
- ***hwcert/rdma/40GigiWarp***
- ***hwcert/rdma/50GigiWarp***
- ***hwcert/rdma/100GigiWarp***
- ***hwcert/rdma/200GigiWarp***
- ***hwcert/rdma/Omnipath***
- ***hwcert/network/Ethernet/2_5GigEthernet***
- ***hwcert/network/Ethernet/5GigEthernet***
- ***hwcert/network/Ethernet/20GigEthernet***
- ***hwcert/network/Ethernet/25GigEthernet-***
- ***hwcert/network/Ethernet/50GigEthernet***
- ***hwcert/network/Ethernet/100GigEthernet***
- ***hwcert/network/Ethernet/200GigEthernet***

- ***rhcert/self-check***

- ***hwcert/sosreport***

- ***hwcert/storage/U2 SATA***

- ***hwcert/storage/M2 SATA***

- ***hwcert/storage/SATA_SSD***

- ***hwcert/storage/SATA***

- ***hwcert/storage/SAS_SSD***

- ***hwcert/storage/SAS***

- ***hwcert/storage/U2_NVME***

- ***hwcert/storage/M2_NVME***

- ***hwcert/storage/PCIE_NVME***

- ***hwcert/storage/NVDIMM***

- ***hwcert/storage/STORAGE***

- 只有在必须指定设备时，才需要其他选项，如在网络和存储测试中，需要告知需要运行哪个设备。您需要查看各种位置来确定此处要使用的设备名称或 UDI。支持可以帮助确定正确的名称或 UDI。找到后，您将使用以下两个选项之一来指定该设备：

- `--device=<devicename >` - 应该测试的设备，用设备名称标识，如 "enp0s25" 或 "host0"。
- `--UDI =<UDI>` - 要测试的设备的唯一设备 ID，由 UDI 字符串标识。
- 通过指定测试名称运行 `rhcert-cli` 命令：

```
rhcert-cli run --test=<test_name>
```

例如：

```
rhcert-cli run --test=audio
```

- 您可以指定 `--device` 来运行特定设备：

```
rhcert-cli run --test=<test name> --device=<device name>
```

例如：

```
rhcert-cli run --test=kdump --device=nfs
```



注意

建议单独使用 `rhcert-cli` 或 `rhcert-run`，并保存结果。混合使用 `rhcert-cli` 和 `rhcert-run` 并保存结果，可能会导致结果无法正确处理结果。

更新于 2024-06-25