



Red Hat Satellite 6.15

管理内容

从 Red Hat 和自定义源导入内容，管理跨生命周期环境的应用程序生命周期，使用内容视图过滤内容，在 Satellite 服务器间同步内容等

Red Hat Satellite 6.15 管理内容

从 Red Hat 和自定义源导入内容，管理跨生命周期环境的应用程序生命周期，使用内容视图过滤内容，在 Satellite 服务器间同步内容等

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

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

使用本指南了解和管理 Satellite 6 中的内容。此类内容的示例包括 RPM 文件和 ISO 镜像。Red Hat Satellite 6 使用在应用程序生命周期中提升的一组内容视图来管理此内容。本指南演示如何创建符合您机构和内容视图的应用程序生命周期，该视图在生命周期环境中对主机状态进行模糊处理。这些内容视图最终构成了在 Red Hat Satellite 6 环境中置备和更新主机的基础。

目录

使开源包含更多	5
对红帽文档提供反馈	6
第 1 章 内容管理简介	7
1.1. RED HAT SATELLITE 中的内容类型	7
第 2 章 管理红帽订阅	8
2.1. 将红帽订阅清单导入到 SATELLITE 服务器中	8
2.2. 查找红帽订阅	9
2.3. 在订阅分配中添加红帽订阅	9
2.4. 从订阅分配中删除红帽订阅	10
2.5. 更新并刷新红帽订阅清单	10
2.6. 将红帽订阅附加到内容主机	11
2.7. 更新多个主机上的红帽订阅	12
2.8. 内容交付网络结构	12
第 3 章 管理备用内容源	14
3.1. 配置自定义备用内容源	14
3.2. 配置简化的备用内容源	16
3.3. 配置 RHUI 备用内容源	17
第 4 章 导入内容	19
4.1. SATELLITE 中的产品和存储库	19
4.2. 产品和存储库的最佳实践	19
4.3. 导入自定义 SSL 证书	19
4.4. 创建自定义产品	20
4.5. 添加自定义 RPM 软件仓库	21
4.6. 启用红帽软件仓库	22
4.7. 同步软件仓库	23
4.8. 同步机构中的所有软件仓库	24
4.9. 下载策略概述	25
4.10. 更改默认下载策略	25
4.11. 更改存储库的下载策略	26
4.12. 镜像策略概述	27
4.13. 更改存储库的镜像策略	27
4.14. 将内容上传到自定义 RPM 存储库	28
4.15. 刷新 CAPSULE 中的内容计数	28
4.16. 配置 SELINUX 以允许自定义端口上的内容同步	29
4.17. 恢复损坏的软件仓库	29
4.18. 重新发布存储库元数据	30
4.19. 重新发布内容视图元数据	31
4.20. 添加 HTTP 代理	31
4.21. 更改产品的 HTTP 代理策略	32
4.22. 更改存储库的 HTTP 代理策略	33
4.23. 创建迁移计划	33
4.24. 为产品分配迁移计划	34
4.25. 为多个产品分配迁移计划	34
4.26. 迁移计划的最佳实践	35
4.27. 限制同步并发	35
4.28. 导入自定义 GPG 密钥	36
4.29. 将自定义存储库限制为 SATELLITE 中的特定操作系统或架构	37

第 5 章 限制主机对内容的访问	38
第 6 章 管理应用程序生命周期	40
6.1. 应用程序生命周期简介	40
6.2. 应用程序生命周期的内容提升	41
6.3. 生命周期环境的最佳实践	42
6.4. 创建生命周期环境路径	43
6.5. 在 CAPSULE 服务器中添加生命周期环境	44
6.6. 从 SATELLITE 服务器中删除生命周期环境	45
6.7. 从 CAPSULE 服务器中删除生命周期环境	45
第 7 章 管理内容视图	47
7.1. RED HAT SATELLITE 中的内容视图	47
7.2. 内容视图的最佳实践	48
7.3. 修补内容主机的最佳实践	48
7.4. 创建内容视图	49
7.5. 查看模块流	50
7.6. 提升内容视图	51
7.7. 复合内容视图概述	52
7.8. 创建复合内容视图	53
7.9. 内容过滤器概述	55
7.10. 解决软件包依赖项	56
7.11. 为内容视图启用依赖项	57
7.12. 内容过滤器示例	58
7.13. 为 YUM 内容创建内容过滤器	59
7.14. 删除多个内容视图版本	61
7.15. 清除搜索过滤器	61
7.16. 标准化内容视图空状态	61
7.17. 比较内容视图版本	62
7.18. 分发归档的内容视图版本	62
第 8 章 在 SATELLITE 服务器间同步内容	63
8.1. 如何使用导出和导入同步内容	63
8.2. 同步自定义软件仓库	65
8.3. 导出库环境	66
8.4. 以可同步格式导出库环境	67
8.5. 导入可同步导出	68
8.6. 递增导出库环境	68
8.7. 导出内容视图版本	69
8.8. 以可同步格式导出内容视图版本	70
8.9. 以递增方式导出内容视图版本	71
8.10. 导出软件仓库	72
8.11. 以可同步格式导出存储库	73
8.12. 以递增方式导出存储库	74
8.13. 以可同步格式逐步导出存储库	75
8.14. 记录您的导出	75
8.15. 导入到库环境中	76
8.16. 从 WEB 服务器导入到库环境中	77
8.17. 导入内容视图版本	77
8.18. 从 WEB 服务器导入内容视图版本	78
8.19. 导入软件仓库	78
8.20. 从 WEB 服务器导入软件仓库	79
8.21. 使用 HAMMER CLI CHEAT 导出和导入内容	80

第 9 章 管理激活码	82
9.1. 激活码的最佳实践	82
9.2. 创建激活码	83
9.3. 更新与激活码关联的订阅	85
9.4. 使用激活码进行主机注册	86
9.5. 启用自动附加	88
9.6. 设置服务级别	89
9.7. 在激活码中启用和禁用软件仓库	89
第 10 章 管理勘误	91
10.1. 勘误的最佳实践	91
10.2. 检查可用的勘误	91
10.3. 可用于勘误搜索的参数	93
10.4. 应用可安装的勘误	94
10.5. 订阅勘误通知	94
10.6. 存储库依赖项解析的限制	94
10.7. 为勘误创建内容视图过滤器	95
10.8. 将勘误添加到增量内容视图	96
10.9. 将勘误应用到主机	97
10.10. 将勘误应用到多个主机	99
10.11. 将勘误应用到主机集合	100
第 11 章 管理容器镜像	102
11.1. 导入容器镜像	102
11.2. 管理容器名称模式	103
11.3. 管理容器 REGISTRY 身份验证	104
11.4. 配置 PODMAN 和 DOCKER 以信任证书颁发机构	105
11.5. 使用容器 REGISTRY	105
第 12 章 管理 ISO 镜像	107
12.1. 从红帽导入 ISO 镜像	107
12.2. 导入单个 ISO 镜像和文件	108
第 13 章 管理 ANSIBLE 内容	110
13.1. 同步 ANSIBLE 集合	110
第 14 章 管理自定义文件类型内容	112
14.1. 为自定义文件类型存储库创建本地源	112
14.2. 为自定义文件类型存储库创建远程源	113
14.3. 创建自定义文件类型存储库	114
14.4. 将文件上传到自定义文件类型存储库	116
14.5. 从自定义文件类型存储库下载文件到主机	117
附录 A. 使用 NFS 共享进行内容存储	120
附录 B. 导入 KICKSTART 软件仓库	122
B.1. 为 RED HAT ENTERPRISE LINUX 9 导入 KICKSTART 软件仓库	122
B.2. 为 RED HAT ENTERPRISE LINUX 8 导入 KICKSTART 软件仓库	126
B.3. 为 RED HAT ENTERPRISE LINUX 7 导入 KICKSTART 软件仓库	129

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。由于这项工作的艰巨性，这些变化正在尽可能地逐步更新。详情请查看 [CTO Chris Wright 的信息](#)。

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

使用 Red Hat JIRA 中的 **Create Issue** 表单提供您的反馈。JIRA 问题在 Red Hat Satellite Jira 项目中创建，您可以在其中跟踪其进度。

先决条件

- 确保您已注册了 [红帽帐户](#)。

流程

1. 单击以下链接：[创建问题](#)。如果 Jira 显示登录错误，则登录并在您重定向到表单后继续。
2. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节号以及问题的详细描述。不要修改表单中的任何其他字段。
3. 点 **Create**。

第 1 章 内容管理简介

在 Satellite 的上下文中，*内容* 被定义为系统上安装的软件。这包括但不限于基础操作系统、中间件服务和最终用户应用程序。使用 Red Hat Satellite，您可以在软件生命周期的每个阶段管理 Red Hat Enterprise Linux 系统的各种内容。

Red Hat Satellite 管理以下内容：

订阅管理

这为组织提供了一种管理其红帽订阅信息的方法。

内容管理

这为组织提供了一种存储红帽内容的方法，并以各种方式组织它。

1.1. RED HAT SATELLITE 中的内容类型

使用 Red Hat Satellite，您可以导入和管理许多内容类型。

例如，Satellite 支持以下内容类型：

RPM 软件包

从与您的红帽订阅相关的软件仓库导入 RPM 软件包。Satellite 服务器从 Red Hat Content Delivery Network 下载 RPM 文件，并将其存储在本地。您可以在内容视图中使用这些存储库及其 RPM 文件。

Kickstart 树

导入 Kickstart 树以调配主机。新系统通过网络访问这些 Kickstart 树，以用作其安装的基本内容。Red Hat Satellite 包含预定义的 Kickstart 模板。您还可以创建自己的 Kickstart 模板。

ISO 和 KVM 镜像

下载并管理用于安装和配置的介质。例如，Satellite 下载、存储和管理特定 Red Hat Enterprise Linux 和非红帽操作系统的 ISO 镜像以及客户机镜像。

自定义文件类型

为您需要的任何类型文件（如 SSL 证书、ISO 镜像和 OVAL 文件）管理自定义内容。

第 2 章 管理红帽订阅

Red Hat Satellite 可以从 Red Hat Content Delivery Network (CDN) 导入内容。Satellite 需要红帽订阅清单，以便从相应的存储库查找、访问和下载内容。您必须有一个红帽订阅清单，其中包含 Satellite 服务器上每个机构的订阅分配。所有订阅信息可在您的红帽客户门户网站帐户中找到。

在完成本章中的任务前，您必须在客户门户网站中创建红帽订阅清单。

请注意，基于权利的订阅模型已弃用，并将在以后的发行版本中删除。红帽建议您改用[简单内容访问](#)的基于访问的订阅服务。

要在客户门户网站中创建、管理和导出红帽订阅清单，请参阅在 *Subscription Central* 中[为连接的 Satellite 服务器创建和管理清单](#)。

使用本章导入红帽订阅清单，并在 Satellite Web UI 中管理清单。

订阅分配和机构

如果您有多个订阅分配，您可以管理多个机构。Satellite 要求在 Satellite 服务器中配置的每个组织有一个分配。这样做的好处是，每个机构都维护单独的订阅，以便您可以支持多个机构，每个机构都有其自身的红帽帐户。

将来的订阅

您可以在订阅分配中使用未来日期的订阅。当您在现有订阅的到期日期前向内容主机添加未来日期的订阅时，您可以对存储库进行不间断访问。

在当前订阅到期前，手动将未来订阅附加到您的内容主机。不要依赖于自动附加方法，因为此方法是不同的目的而设计，且可能无法正常工作。如需更多信息，请参阅 [第 2.6 节“将红帽订阅附加到内容主机”](#)。

当将来的订阅处于活跃状态时，您必须刷新清单以同步存储库内容。如需更多信息，请参阅 [第 2.5 节“更新并刷新红帽订阅清单”](#)。

其他资源

- [在断开连接的网络环境中安装 Satellite 服务器中的将 Satellite 服务器配置为使用来自自定义 CDN 的内容](#)

2.1. 将红帽订阅清单导入到 SATELLITE 服务器中

使用以下步骤将红帽订阅清单导入到 Satellite 服务器中。



注意

在机构上设置简单内容访问(SCA)，而不是清单。导入清单不会更改您机构的简单内容访问状态。

先决条件

- 您必须有一个 [从红帽客户门户网站导出的红帽订阅](#) 清单文件。如需更多信息，请参阅 [使用红帽订阅管理](#) 中的 [创建和管理](#) 清单。

流程

1. 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。

2. 在 Satellite Web UI 中，进入到 **Content > Subscriptions** 并点 **Manage Manifest**。
3. 在 **Manage Manifest** 窗口中，单击 **Choose File**。
4. 导航到包含红帽订阅清单文件的位置，然后单击 **Open**。

CLI 过程

1. 将红帽订阅清单文件从本地机器复制到 Satellite 服务器：

```
$ scp ~/manifest_file.zip root@satellite.example.com:~/
```

2. 以 **root** 用户身份登录 Satellite 服务器，再导入 Red Hat 订阅清单文件：

```
# hammer subscription upload \  
--file ~/manifest_file.zip \  
--organization "My_Organization"
```

现在，您可以启用软件仓库并导入红帽内容。如需更多信息，[请参阅管理内容中的导入内容](#)。

2.2. 查找红帽订阅

当您将在红帽订阅清单导入到 Satellite 服务器时，清单中的订阅会在 Subscriptions 窗口中列出。如果您有大量订阅，您可以过滤结果以查找特定订阅。

先决条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。如需更多信息，请参阅 [第 2.1 节“将红帽订阅清单导入到 Satellite 服务器中”](#)。

流程

1. 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。
2. 在 Satellite Web UI 中，进入到 **Content > Subscriptions**。
3. 在 Subscriptions 窗口中，点 **Search** 字段查看构建搜索查询的搜索条件列表。
4. 选择搜索条件来显示更多选项。
5. 构建了搜索查询后，点搜索图标。

例如，如果您将光标移到 **Search** 字段中并选择 **expires**，然后按空格，则会出现一个另一个列表，包括 **>**, **<**, 或 **=** 字符。如果选择 **& gt;** 并按空格栏，会显示另一个自动选项列表。您还可以输入您自己的标准。

2.3. 在订阅分配中添加红帽订阅

使用以下步骤将红帽订阅添加到 Satellite Web UI 中的订阅分配中。

先决条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。如需更多信息，请参阅 [第 2.1 节“将红帽订阅清单导入到 Satellite 服务器中”](#)。

流程

1. 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。
2. 在 Satellite Web UI 中，进入到 **Content > Subscriptions**。
3. 在 Subscriptions 窗口中，点 **Add Subscriptions**。
4. 在要添加的每个订阅行中，输入 **Quantity to Allocate** 列中的数量。
5. 点 **Submit**

2.4. 从订阅分配中删除红帽订阅

使用以下步骤从 Satellite Web UI 中的订阅分配中删除红帽订阅。



注意

不得删除清单。如果您从红帽客户门户网站或 Satellite Web UI 中删除清单，则会删除所有内容主机的所有权利。

先决条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。如需更多信息，请参阅 [第 2.1 节“将红帽订阅清单导入到 Satellite 服务器中”](#)。

流程

1. 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。
2. 在 Satellite Web UI 中，进入到 **Content > Subscriptions**。
3. 在您要删除的每个订阅行中，选择对应的复选框。
4. 单击 **Delete**，然后确认删除。

2.5. 更新并刷新红帽订阅清单

每次更改订阅分配时，您必须刷新清单以反映这些更改。例如，如果您执行以下操作，您必须刷新清单：

- 续订订阅
- 调整订阅数量
- 购买额外订阅

您可以在 Satellite Web UI 中直接刷新清单。或者，您可以导入包含更改的更新清单。

流程

1. 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。
2. 在 Satellite Web UI 中，进入到 **Content > Subscriptions**。
3. 在 Subscriptions 窗口中，点 **Manage Manifest**。

- 在 Manage Manifest 窗口中，点 **Refresh**。

2.6. 将红帽订阅附加到内容主机

在置备过程中，使用激活码是将订阅附加到内容主机的主要方法。但是，激活码无法更新现有主机。如果您需要将新的订阅（如将来的订阅）附加到一个主机，请使用以下步骤。



重要

只有在 Satellite 上禁用了 Simple Content Access (SCA) 时，这个过程才有效。启用 SCA 后，您不需要将订阅附加到您的主机。请注意，新创建的机构默认启用 SCA。要了解有关 SCA 的更多信息，请参阅 [简单内容访问](#)。

有关更新多个主机的详情，请参考 [第 2.7 节“更新多个主机上的红帽订阅”](#)。

有关激活码的详情请参考 [第 9 章 管理激活码](#)。

Satellite 订阅

在 Satellite 中，您必须为您要管理的每个 Red Hat Enterprise Linux 主机维护一个 Red Hat Enterprise Linux Satellite 订阅（以前称为 Red Hat Enterprise Linux 智能管理）。

但是，您不需要将 Satellite 订阅附加到每个内容主机。Satellite 订阅无法自动附加到 Satellite 中的内容主机，因为它们没有与任何产品证书关联。在内容主机中添加 Satellite 订阅不提供任何内容或存储库访问。如果需要，您可以在清单中添加 Satellite 订阅，以满足您自己的记录或跟踪目的。

先决条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。如需更多信息，请参阅 [第 2.1 节“将红帽订阅清单导入到 Satellite 服务器中”](#)。

流程

- 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。
- 在 Satellite Web UI 中，进入到 **Hosts > Content Hosts**。
- 在您要更改其订阅的每个内容主机的行中，选择对应的复选框。
- 在 **Select Action** 列表中，选择 **Manage Subscriptions**。
- （可选）在 **Search** 字段中输入键和值来过滤显示的订阅。
- 选中您要添加或删除的订阅的复选框，然后单击 **Add Selected** 或 **Remove Selected**。
- 点 **Done** 保存更改。

CLI 过程

- 在 Satellite 服务器上列出所有可用的订阅：

```
# hammer subscription list \
--organization-id My_Organization_ID
```

- 在您的主机中附加订阅：

```
# hammer host subscription attach \
--host My_Host_Name \
--subscription-id My_Subscription_ID
```

2.7. 更新多个主机上的红帽订阅

使用此流程同时更改多个内容主机。



重要

只有在 Satellite 上禁用了 Simple Content Access (SCA) 时，这个过程才有效。启用 SCA 后，您不需要将订阅附加到您的主机。请注意，新创建的机构默认启用 SCA。要了解有关 SCA 的更多信息，请参阅 [简单内容访问](#)。

流程

1. 在 Satellite Web UI 中，确保将上下文设置为您要使用的组织。
2. 在 Satellite Web UI 中，进入到 **Hosts > Content Hosts**。
3. 在您要更改其订阅的每个内容主机的行中，选择对应的复选框。
4. 在 **Select Action** 列表中，选择 **Manage Subscriptions**。
5. （可选）在 **Search** 字段中输入键和值来过滤显示的订阅。
6. 选中要添加的订阅左侧的复选框，然后单击添加所选项或根据需要 **删除** 所选项。
7. 点 **Done** 保存更改。

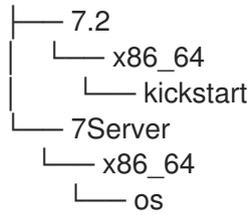
2.8. 内容交付网络结构

位于 cdn.redhat.com 的 Red Hat Content Delivery Network (CDN) 是地理分布的静态 webserver 系列，其中包括供系统使用的内容和勘误。此内容可以通过使用 Subscription Manager 或通过 Satellite Web UI 注册的系统访问。CDN 的可访问子集是通过使用 [Red Hat Subscription Management](#) 或使用 Satellite 服务器附加到系统的订阅进行配置。

Red Hat Content Delivery 网络受 X.509 证书身份验证保护，以确保只有有效用户可以访问它。

CDN 的目录结构

```
$ tree -d -L 11
├── content 1
│   ├── beta 2
│   │   ├── rhel 3
│   │   │   ├── server 4
│   │   │   │   ├── 7 5
│   │   │   │   ├── x86_64 6
│   │   │   │   └── sat-tools 7
│   └── dist
│       ├── rhel
│       │   ├── server
│       │   └── 7
```



- 1 content 目录。
- 2 负责内容生命周期的目录。通用目录包括 **beta**（用于 Beta 代码）、**dist**（用于生产环境）和 **eus**（用于延长更新支持）目录。
- 3 负责产品名称的目录。通常，适用于 Red Hat Enterprise Linux 的 **rhel**。
- 4 负责产品类型的目录。对于 Red Hat Enterprise Linux，这可能包括 **服务器**、**workstation** 和 **compuenode** 目录。
- 5 负责发行版本的目录，如 **7**、**7.2** 或 **7Server**。
- 6 负责基本架构的目录，如 **i386** 或 **x86_64**。
- 7 负责存储库名称的目录，如 **sat-tools**、**kickstart**、**rhscl**。有些组件还有额外的子目录，它们可能有所不同。

此目录结构也用于 Red Hat Subscription Manifest。

第 3 章 管理备用内容源

备用内容源定义了在不同步期间下载内容的替代路径。内容本身从备用内容源下载，而元数据则从 Satellite 服务器或上游 URL 下载，具体取决于配置。如果内容位于本地文件系统或最接近的网络上，您可以使用备用内容源来加快同步。您可以为 Satellite 服务器和 Capsule 设置备用内容源。您必须在创建后或进行任何更改后刷新备用内容源。每周 cron 作业会刷新所有备用内容源。您还可以使用 Satellite Web UI 或 Hammer CLI 手动刷新备用内容源。与 Satellite 服务器关联的备用内容源或附加到多个组织的胶囊服务器会影响所有组织。

备用内容源有三种类型：

Custom

自定义备用内容源从网络或文件系统上的任何上游存储库下载内容。

简化

简化备用内容源，从您的 Satellite 服务器中为所选产品复制上游存储库信息。从胶囊连接到上游存储库比从 Satellite 服务器的连接更快的情况，简化的备用内容源是理想的选择。在创建简化的备用内容源时，选择红帽产品将从红帽 CDN 下载内容到胶囊。

RHUI

RHUI 备用内容源从 Red Hat Update Infrastructure 服务器下载内容。Satellite Web UI 提供了示例，可帮助您查找网络路径和导入身份验证凭据。RHUI 备用内容源必须是 RHUI 版本 4 或更高版本，并使用默认安装配置。例如，不支持 AWS RHUI，因为它使用具有唯一身份验证要求的安装场景。

备用内容源的权限要求

非管理员用户必须具有以下权限来管理备用内容源：

1. `view_smart_proxies`
2. `view_content_credentials`
3. `view_organizations`
4. `view_products`

除了以上权限外，根据用户可以执行的操作，分配特定于备用内容源的权限：

1. `view_alternate_content_sources`
2. `create_alternate_content_sources`
3. `edit_alternate_content_sources`
4. `destroy_alternate_content_sources`

3.1. 配置自定义备用内容源

先决条件

- 如果存储库需要 SSL 身份验证，请将 SSL 证书和密钥导入到 Satellite。如需更多信息，请参阅 [管理内容](#) 中的 [导入自定义 SSL 证书](#)
- 请注意，备用内容源路由附加您提供的子路径的基本 URL 组成。例如：如果您的基本 URL 是 <https://server.example.com>，且您的子路径为 `rhel7/` 和 `rhel8/`，则将搜索 <https://server.example.com/rhel7/> 和 <https://server.example.com/rhel8/>。

流程

1. 在 Satellite Web UI 中，进入到 Content > Alternate Content Sources。
2. 单击 Add Source，将 Source 类型设置为 Custom。
3. 选择内容类型。
4. 在 Name 字段中输入备用内容源的名称。
5. 可选：在 Description 字段中，为 ACS 提供描述。
6. 选择要同步备用内容源的 Capsule。
7. 输入备用内容源的基本 URL。
8. 输入以逗号分隔的子路径列表。
9. 如果需要，请提供手动身份验证或内容身份验证凭证。
10. 如果需要 SSL 验证，请启用 验证 SSL 并选择 SSL 证书。
11. 检查详情并点 Add。
12. 导航到 Content > Alternate Content Sources > 点新创建的备用内容源 > SelectRefresh 旁的垂直 ellipsis。

CLI 过程

1. 在 Satellite 服务器上输入以下命令：

```
# hammer alternate-content-source create \
--alternate-content-source-type custom \
--base-url "https://local-repo.example.com:port" \
--name "My_ACS_Name" \
--smart-proxy-ids My_Capsule_ID
```

2. 检查新创建的备用内容源是否已列出：

```
# hammer alternate-content-source list
```

3. 刷新备用内容源：

```
# hammer alternate-content-source refresh --id My_Alternate_Content_Source_ID
```

4. 添加要同步备用内容源的 Capsule：

```
# hammer alternate-content-source update \
--id My_Alternate_Content_Source_ID \
--smart-proxy-ids My_Capsule_ID
```

5. 刷新备用内容源：

```
# hammer alternate-content-source refresh --id My_Alternate_Content_Source_ID
```

3.2. 配置简化的备用内容源

流程

1. 在 Satellite Web UI 中，进入到 Content >Alternate Content Sources。
2. 单击 Add Source，将 Source 类型设置为 Simplified。
3. 选择内容类型。
4. 在 Name 字段中输入备用内容源的名称。
5. 可选：在 Description 字段中，为 ACS 提供描述。
6. 选择要同步备用内容源的 Capsule。
7. 可选：如果您希望 ACS 使用 Capsule 服务器的 HTTP 代理，请选择 Use HTTP 代理。
8. 选择应使用备用内容源的产品。
9. 检查详情并点 Add。
10. 导航到 Content >Alternate Content Sources，点新创建的备用内容源旁边的垂直 ellipsis，然后选择 Refresh。

CLI 过程

1. 创建简化的 ACS：

```
# hammer alternate-content-source create \  
--alternate-content-source-type simplified \  
--name My_ACS_Name \  
--product-ids My_Product_ID \  
--smart-proxy-ids My_Capsule_ID
```

2. 检查新创建的 ACS 是否已列出：

```
# hammer alternate-content-source list
```

3. 刷新 ACS：

```
# hammer alternate-content-source refresh --id My_ACS_ID
```

3.2.1. 直接从红帽 CDN 同步 Capsule

您可以使用简化的备用内容源将 Capsule 配置为直接从 Red Hat CDN 同步内容。

流程

1. 在 Satellite Web UI 中，进入到 Content >Alternate Content Sources。
2. 单击 Add Source，将 Source 类型设置为 Simplified。
3. 将内容类型设置为 Yum。

4. 在 Name 字段中输入备用内容源的名称。
5. 可选：在 Description 字段中，为备用内容源提供描述。
6. 选择您要直接从 Red Hat CDN 同步的 Capsules。
7. 可选：如果您希望 ACS 使用 Capsule 服务器的 HTTP 代理，请选择 Use HTTP 代理。
8. 从 Red Hat CDN 选择应同步到 Capsule 的红帽产品。
9. 检查详情并点 Add。
10. 导航到 Content > Alternate Content Sources，点新创建的备用内容源旁边的垂直 ellipsis，然后选择 Refresh。

现在，胶囊将从红帽 CDN 下载内容，而不是从 Satellite 下载内容。

3.3. 配置 RHUI 备用内容源

先决条件

- 在 RHUA 节点上为所需的仓库生成 [客户端授权证书](#)，如 [配置和管理 Red Hat Update Infrastructure 中的使用红帽更新基础架构管理工具创建客户端授权证书](#) 中所述。
- 将客户端授权证书导入到 Satellite。如需更多信息，请参阅 [管理内容](#) 中的 [导入自定义 SSL 证书](#)。
- 获取所需存储库的子路径列表。在您的 RHUA 服务器上执行以下命令：

```
# rhui-manager repo info --repo_id My_Repo_ID
```

- 请注意，备用内容源路由附加您提供的子路径的基本 URL 组成。例如：如果您的基本 URL 是 <https://server.example.com>，且您的子路径为 `rhel7/` 和 `rhel8/`，则将搜索 <https://server.example.com/rhel7/> 和 <https://server.example.com/rhel8/>。

流程

1. 在 Satellite Web UI 中，进入到 Content > Alternate Content Sources。
2. 点 Add Source，将 Source 类型设置为 RHUI。
3. 使用 Satellite Web UI 中提供的命令生成 RHUI 证书。确保传递所需存储库的 repo 标签。
4. 在 Name 字段中输入备用内容源的名称。
5. 可选：在 Description 字段中，为 ACS 提供描述。
6. 选择要同步备用内容源的 Capsule。
7. 可选：如果您希望 ACS 使用胶囊的 HTTP 代理，请选择 Use HTTP 代理。
8. 输入 Red Hat Update Infrastructure CDS 节点的基本 URL。
9. 输入以逗号分隔的子路径列表。
10. 如果需要，提供内容凭据。

11. 如果需要 SSL 验证，请启用 验证 SSL 并选择 SSL 证书。
12. 检查详情并点 Add。
13. 导航到 Content >Alternate Content Sources，点新创建的备用内容源旁边的垂直 ellipsis，然后选择 Refresh。

CLI 过程

1. 在 Satellite 服务器上输入以下命令：

```
# hammer alternate-content-source create \  
--alternate-content-source-type rhui \  
--base-url "https://rhui-cds-node/pulp/content" \  
--name "My_ACS_Name" \  
--smart-proxy-ids My_Capsule_ID \  
--ssl-client-cert-id My_SSL_Client_Certificate_ID \  
--ssl-client-key-id My_SSL_Client_Key_ID \  
--subpaths path/to/repo/1/,path/to/repo/2/ \  
--verify-ssl 1
```

2. 检查新创建的备用内容源是否已列出：

```
# hammer alternate-content-source list
```

3. 刷新备用内容源：

```
# hammer alternate-content-source refresh --id My_Alternate_Content_Source_ID
```

4. 添加要同步备用内容源的 Capsule：

```
# hammer alternate-content-source update \  
--id My_Alternate_Content_Source_ID \  
--smart-proxy-ids My_Capsule_ID
```

5. 刷新备用内容源：

```
# hammer alternate-content-source refresh --id My_Alternate_Content_Source_ID
```

第 4 章 导入内容

本章概述了如何将不同类型的自定义内容导入到 Satellite。例如，您可以使用以下章节获取特定类型的自定义内容的信息，但底层步骤是相同的：

- [第 12 章 管理 ISO 镜像](#)
- [第 14 章 管理自定义文件类型内容](#)

4.1. SATELLITE 中的产品和存储库

Satellite 中的红帽内容和自定义内容都有相似性：

- 产品及其存储库之间的关系相同，存储库仍然需要同步。
- 自定义产品需要订阅主机才能访问，类似于红帽产品的订阅。Satellite 会为您创建的每个自定义产品创建一个订阅。

红帽内容已组织到产品中。例如，Red Hat Enterprise Linux 服务器是 Satellite 中的 *产品*。该产品的存储库由不同的版本、架构和附加组件组成。对于红帽软件仓库，在启用存储库后会自动创建产品。如需更多信息，请参阅 [第 4.6 节 “启用红帽软件仓库”](#)。

其他内容可以组织到自定义产品中。例如，您可以创建一个 EPEL (Extra Packages for Enterprise Linux) 产品，并向其添加一个 "EPEL 7 x86_64" 存储库。

有关创建和打包 RPM 的更多信息，请参阅 [Red Hat Enterprise Linux RPM 打包指南](#)

4.2. 产品和存储库的最佳实践

- 每个产品和内容视图使用一个内容视图，例如：`yum content only`。
- 通过 HTTP 提供文件存储库。如果将 `Protected` 设为 `true`，则只能使用全局调试证书下载内容。
- 使用 Hammer 脚本或 [Ansible playbook](#) 自动创建多个产品和存储库。
- 对于红帽内容，请将您的红帽清单导入到 Satellite 中。如需更多信息，请参阅 [第 2 章 管理红帽订阅](#)。
- 避免使用 Upstream URL 将内容上传到存储库。相反，创建一个存储库，以在不设置 Upstream URL 的情况下将内容同步并上传到。
如果您将内容上传到已同步另一个存储库的存储库，则内容可能会被覆盖，具体取决于镜像策略和内容类型。

4.3. 导入自定义 SSL 证书

在从外部源同步自定义内容前，您可能需要将 SSL 证书导入到自定义产品中。这可能包括您要同步的 upstream 存储库的客户端证书和密钥或 CA 证书。

如果您需要 SSL 证书和密钥下载软件包，您可以将其添加到 Satellite 中。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content >Content Credentials。在 Content Credentials 窗口中，单击 Create Content Credential。
2. 在 Name 字段中输入您的 SSL 证书的名称。
3. 从 Type 列表中，选择 SSL Certificate。
4. 在 Content Credentials Content 字段中，粘贴您的 SSL 证书，或者点击 Browse 上传您的 SSL 证书。
5. 单击 Save。

CLI 过程

1. 将 SSL 证书复制到 Satellite 服务器中：

```
$ scp My_SSL_Certificate root@satellite.example.com:~/.
```

或者从在线源将 SSL 证书下载到 Satellite 服务器中：

```
$ wget -P ~ http://upstream-satellite.example.com/pub/katello-server-ca.crt
```

2. 将 SSL 证书上传到 Satellite：

```
# hammer content-credential create \
  --content-type cert \
  --name "My_SSL_Certificate" \
  --organization "My_Organization" \
  --path ~/My_SSL_Certificate
```

4.4. 创建自定义产品

创建自定义产品，以便您可以将存储库添加到自定义产品中。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content >Products，点 Create Product。
2. 在 Name 字段中输入产品的名称。Satellite 根据您为名称输入的内容自动完成 Label 字段。
3. 可选：在 GPG Key 列表中，为产品选择 GPG 密钥。
4. 可选：在 SSL CA Cert 列表中，为产品选择 SSL CA 证书。
5. 可选：在 SSL Client Cert 列表中，为产品选择 SSL 客户端证书。
6. 可选：在 SSL Client Key 列表中，为产品选择 SSL 客户端密钥。
7. 可选：在 Sync Plan 列表中，选择现有的迁移计划，或者点 Create Sync Plan 并为您的产品要求创建一个迁移计划。
8. 在 Description 字段中，输入产品的描述。
9. 单击 Save。

CLI 过程

运行以下命令来创建产品：

```
# hammer product create \
--name "My_Product" \
--sync-plan "Example Plan" \
--description "Content from My Repositories" \
--organization "My_Organization"
```

4.5. 添加自定义 RPM 软件仓库

使用这个流程在 Satellite 中添加自定义 RPM 存储库。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

Satellite Web UI 中的 Products 窗口还提供了一个 Repo Discovery 功能，它从 URL 中查找所有存储库，您可以选择要添加到自定义产品的所有存储库。例如，您可以使用 Repo Discovery 搜索 <https://download.postgresql.org/pub/repos/yum/16/redhat/> 并列不同 Red Hat Enterprise Linux 版本和架构的所有软件仓库。这有助于帮助用户节省从单一源导入多个存储库的时间。

支持自定义 RPM

红帽不支持直接来自第三方站点的上游 RPM。这些 RPM 用于演示同步过程。对于这些 RPM 的问题，请联系第三方开发人员。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products 并选择您要使用的产品，然后点 New Repository。
2. 在 Name 字段中输入存储库的名称。Satellite 根据您为名称输入的内容自动完成 Label 字段。
3. 可选：在 Description 字段中输入存储库的描述。
4. 从 Type 列表中，选择 yum 作为存储库类型。
5. 可选：在 Restrict to Architecture 列表选择一个架构。如果您要将存储库提供给所有主机，无论架构是什么，请确保选择 No limitations。
6. 可选：在 Restrict to OS Version 列表中，选择 OS 版本。如果您要将存储库提供给所有主机，无论操作系统版本是什么，请确保选择 No limitations。
7. 可选：在 Upstream URL 字段中输入要用作源的外部存储库的 URL。Satellite 支持三种协议：<http://>、<https://> 和 <file://>。如果您使用 <file://> 存储库，则必须将其放在 `/var/lib/pulp/sync_imports/` 目录中。
如果没有输入上游 URL，您可以手动上传软件包。
8. 可选：检查 Ignore SRPMs 复选框，以排除源 RPM 软件包与 Satellite 同步。
9. 可选：如果您收到错误 Treeinfo 文件，则检查 Ignore treeinfo 复选框，其格式为 INI。如果跳过了 treeinfo 文件，则与 Kickstart 相关的所有文件都会在存储库中丢失。
10. 如果要验证上游存储库的 SSL 证书是否由可信 CA 签名，请选择 Verify SSL 复选框。
11. 可选：在 Upstream Username 字段中，如果需要身份验证，请输入上游存储库的用户名。如果存储库不需要身份验证，请清除此字段。

12. 可选：在 Upstream Password 字段中，为 upstream 仓库输入对应的密码。如果存储库不需要身份验证，请清除此字段。
13. 可选：在 Upstream Authentication Token 字段中，提供上游存储库用户的令牌以进行身份验证。如果存储库不需要身份验证，请将此字段留空。
14. 从 Download Policy 列表中，选择同步 Satellite 服务器执行的类型。更多信息请参阅第 4.9 节“下载策略概述”。
15. 从 Mirroring Policy 列表中，选择 Satellite 服务器执行的内容类型。更多信息请参阅第 4.12 节“镜像策略概述”。
16. 可选：在 Retain 软件包版本字段中，输入您要为每个软件包保留的版本数量。
17. 可选：在 HTTP Proxy Policy 字段中，选择一个 HTTP 代理。
18. 从 Checksum 列表中，选择存储库的校验和类型。
19. 可选：您可以清除 Unprotected 复选框，以要求订阅权利证书才能访问此存储库。默认情况下，存储库通过 HTTP 发布。
20. 可选：在 GPG Key 列表中，为产品选择 GPG 密钥。
21. 可选：在 SSL CA Cert 字段中，为存储库选择 SSL CA 证书。
22. 可选：在 SSL Client cert 字段中，为存储库选择 SSL Client Certificate。
23. 可选：在 SSL Client Key 字段中，为存储库选择 SSL Client Key。
24. 单击 Save 以创建该存储库。

CLI 过程

1. 输入以下命令来创建存储库：

```
# hammer repository create \
--arch "My_Architecture" \
--content-type "yum" \
--gpg-key-id My_GPG_Key_ID \
--name "My_Repository" \
--organization "My_Organization" \
--os-version "My_OS_Version" \
--product "My_Product" \
--publish-via-http true \
--url My_Upstream_URL
```

继续 [同步存储库](#)。

4.6. 启用红帽软件仓库

如果外部网络访问需要使用 HTTP 代理，请为您的服务器配置默认的 HTTP 代理。如需更多信息，请参阅 [向 Satellite 添加默认 HTTP 代理](#)。

要选择要同步的存储库，您必须首先识别包含该存储库的产品，然后根据相关发行版本和基本架构启用该存储库。

对于 Red Hat Enterprise Linux 8 主机

要置备 Red Hat Enterprise Linux 8 主机，您需要 Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs) 和 Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) 软件仓库。

对于 Red Hat Enterprise Linux 7 主机

要置备 Red Hat Enterprise Linux 7 主机，您需要 Red Hat Enterprise Linux 7 Server (RPMs) 存储库。

将 Red Hat Enterprise Linux 操作系统的发行版本与 7Server 仓库或 7.X 仓库相关联的不同在于：7Server 仓库包括所有最新的更新，Red Hat Enterprise Linux 7X 仓库会在下一个次版本发行后停止获取更新。请注意，Kickstart 软件仓库只有次版本。

流程

1. 在 Satellite Web UI 中，进入到 Content > Red Hat Repositories。
2. 要查找存储库，请输入存储库名称，或者将 Recommended Repositories 按钮切换到位置，以查看您需要的存储库列表。
3. 在 Available Repositories 窗格中，单击存储库以展开存储库集。
4. 单击您想要的基本架构和发行版本旁边的 Enable 图标。

CLI 过程

1. 要搜索您的产品，请输入以下命令：

```
# hammer product list --organization "My_Organization"
```

2. 列出产品设置的存储库：

```
# hammer repository-set list \
--product "Red Hat Enterprise Linux Server" \
--organization "My_Organization"
```

3. 使用名称或 ID 号启用存储库。包含发行版本，如 7Server，以及基本架构，如 x86_64。

```
# hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (RPMs)" \
--releasever "7Server" \
--basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" \
--organization "My_Organization"
```

4.7. 同步软件仓库

您必须同步存储库，才能将内容下载到 Satellite。您可以使用此流程对软件仓库进行初始同步，或者根据需要手动同步存储库。

您还可以同步机构中的所有存储库。更多信息请参阅 [第 4.8 节“同步机构中的所有软件仓库”](#)。

创建一个迁移计划，以确保定期更新。更多信息请参阅 [第 4.23 节“创建迁移计划”](#)。

同步持续时间取决于每个存储库的大小以及网络连接的速度。下表提供了根据可用的互联网带宽，对同步内容所需的时间进行估算：

	单软件包(10Mb)	次发行版本(750Mb)	主发行版本(6Gb)
256 Kbps	5 分钟 27 个 Secs	6 hrs 49 Mins 36 Secs	2 天 7 Hrs 55 Mins
512 Kbps	2 分钟 43.84 Secs	3 hrs 24 Mins 48 Secs	1 天 3 Hrs 57 Mins
T1 (1.5 Mbps)	54.33 secs	1 HR 7 Mins 54.78 Secs	9 hrs 16 Mins 20.57 Secs
10 Mbps	8.39 secs	10 分钟 29.15 Secs	1 HR 25 Mins 53.96 Secs
100 Mbps	0.84 secs	1 分钟 2.91 Secs	8 分钟 35.4 Secs
1000 Mbps	0.08 secs	6.29 秒	51.54 secs

流程

1. 在 Satellite Web UI 中，进入到 Content > Products 并选择 Product，其中包含您要同步的存储库。
2. 选择您要同步的存储库，然后单击 Sync Now。
3. 可选：要在 Satellite Web UI 中查看同步的进度，请导航到 Content > Sync Status 并展开对应的 Product 或 repository 树。

CLI 过程

- 同步整个产品：

```
# hammer product synchronize \
--name "My_Product" \
--organization "My_Organization"
```

- 同步单个存储库：

```
# hammer repository synchronize \
--name "My_Repository" \
--organization "My_Organization" \
--product "My_Product"
```

4.8. 同步机构中的所有软件仓库

使用这个流程同步机构中的所有存储库。

流程

1. 使用 SSH 登录您的 Satellite 服务器。
2. 运行以下 Bash 脚本：

```
ORG="My_Organization"
```

```
for i in $(hammer --no-headers --csv repository list --organization $ORG --fields Id)
do
  hammer repository synchronize --id ${i} --organization $ORG --async
done
```

4.9. 下载策略概述

Red Hat Satellite 提供多个下载策略来同步 RPM 内容。例如，您可能希望只下载内容元数据，同时延迟实际内容下载。

Satellite 服务器有以下策略：

immediate

Satellite 服务器在同步期间下载所有元数据和软件包。

按需

Satellite 服务器仅在同步期间下载元数据。只有当胶囊或直接连接的客户端请求时，Satellite 服务器才会获取并存储在文件系统中。如果将 Capsule 上的对应存储库设置为 Immediate，因为 Satellite 服务器被强制下载所有软件包，则此设置无效。

On Demand 策略充当 *Lazy Synchronization* 功能，因为它们节省时间同步内容。lazy 同步功能必须仅用于 Yum 存储库。您可以将软件包添加到内容视图中，并正常提升到生命周期环境。

胶囊服务器具有以下策略：

immediate

Capsule 服务器在同步期间下载所有元数据和软件包。如果 Satellite 服务器上对应的存储库被设置为 On Demand，则不要使用此设置，因为 Satellite 服务器被强制下载所有软件包。

按需

Capsule 服务器仅在同步期间下载元数据。胶囊服务器仅在直接连接的客户端请求时获取并存储文件系统上的软件包。使用 On Demand 下载策略时，如果胶囊服务器上不可用内容，则会从 Satellite 服务器下载其内容。

inherit

胶囊服务器将从卫星服务器上对应的存储库继承存储库的下载策略。

流的下载策略

胶囊的流下载策略允许胶囊避免缓存任何内容。从 Capsule 请求内容时，它将充当代理，并直接从 Satellite 请求内容。

4.10. 更改默认下载策略

您可以设置 Satellite 应用到您在所有机构中创建的软件仓库的默认下载策略。

根据它是红帽还是非红帽自定义存储库，Satellite 使用单独的设置。更改默认值不会更改现有设置。

流程

1. 在 Satellite Web UI 中，进入到 Administer > Settings。
2. 单击 Content 选项卡。

3. 根据您的要求更改默认下载策略：

- 要更改红帽存储库的默认下载策略，请更改 Default Red Hat Repository 下载策略设置的值。
- 要更改自定义存储库的默认下载策略，请更改 Default Custom Repository 下载策略设置的值。

CLI 过程

- 要将 Red Hat 仓库的默认下载策略改为 **immediate** 或 **on_demand** 之一，请输入以下命令：

```
# hammer settings set \
--name default_redhat_download_policy \
--value immediate
```

- 要将非 Red Hat 自定义存储库的默认下载策略改为 **immediate** 或 **on_demand** 之一，请输入以下命令：

```
# hammer settings set \
--name default_download_policy \
--value immediate
```

4.11. 更改存储库的下载策略

您可以为存储库设置下载策略。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 选择所需的产品名称。
3. 在 Repositories 选项卡上，单击所需的存储库名称，找到 Download Policy 字段，然后单击编辑图标。
4. 从列表中选择所需的下载策略，然后单击 Save。

CLI 过程

1. 列出机构的软件仓库：

```
# hammer repository list \
--organization-label My_Organization_Label
```

2. 将存储库的下载策略改为 **immediate** 或 **on_demand**：

```
# hammer repository update \
--download-policy immediate \
--name "My_Repository" \
--organization-label My_Organization_Label \
--product "My_Product"
```

4.12. 镜像策略概述

镜像使本地存储库完全与上游存储库同步。如果自上次同步以来从上游存储库中删除了任何内容，则也会将其从本地存储库中移除。

您可以使用镜像策略在同步存储库时对 repodata 和内容的镜像进行精细控制。例如，如果无法为存储库镜像 repodata，您可以将镜像策略设置为仅镜像此存储库的内容。

Satellite 服务器具有以下镜像策略：

additive

内容和 repodata 都不会被镜像。因此，仅添加新内容，因为最后一个同步被添加到本地存储库中，且不会删除任何内容。

仅限内容

仅镜像内容，而不镜像 repodata。有些软件仓库不支持元数据镜像，在这种情况下，您可以将镜像策略设置为只镜像内容。

完整镜像

镜像内容和 repodata。这是最快的方法。此镜像策略仅适用于 Yum 内容。



警告

避免使用 Complete Mirror 镜像策略为存储库重新发布元数据。这也适用于包含带有 Complete Mirror 镜像策略的存储库的内容视图。

4.13. 更改存储库的镜像策略

您可以为存储库设置镜像策略。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 选择产品名称。
3. 在 Repositories 选项卡上，单击存储库名称，找到 Mirroring Policy 字段，再单击编辑图标。
4. 从列表中选择一个镜像策略，再点 Save。

CLI 过程

1. 列出机构的软件仓库：

```
# hammer repository list \
--organization-label My_Organization_Label
```

2. 将一个仓库的镜像（mirroring）策略改为 `additive, mirror_complete`, 或 `mirror_content_only` :

```
# hammer repository update \
--id 1 \
--mirroring-policy mirror_complete
```

4.14. 将内容上传到自定义 RPM 存储库

您可以将单个 RPM 和源 RPM 上传到自定义 RPM 存储库。您可以使用 Satellite Web UI 或 Hammer CLI 上传 RPM。您必须使用 Hammer CLI 上传源 RPM。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 点自定义产品的名称。
3. 在 Repositories 选项卡中，单击自定义 RPM 存储库的名称。
4. 在 Upload Package 下，单击 Browse... 并选择您要上传的 RPM。
5. 点 Upload。

要查看此存储库中的所有 RPM，请点 Content Counts 下 Packages 旁边的数字。

CLI 过程

- 输入以下命令上传 RPM :

```
# hammer repository upload-content \
--id My_Repository_ID \
--path /path/to/example-package.rpm
```

- 输入以下命令上传源 RPM :

```
# hammer repository upload-content \
--content-type srpm \
--id My_Repository_ID \
--path /path/to/example-package.src.rpm
```

上传完成后，您可以使用命令 `hammer srpm list` 和 `hammer srpm info --id srpm_ID` 来查看源 RPM 的信息。

4.15. 刷新 CAPSULE 中的内容计数

如果 Capsules 启用了同步内容，您可以刷新与 Capsule 关联的环境可用的内容数量。这将显示这些环境中可供胶囊使用的内容视图。然后，您可以扩展内容视图来查看与该内容视图版本关联的存储库。

流程

1. 在 Satellite Web UI 中，进入到 Infrastructure > Capsules，再选择您要查看同步内容的 Capsule。

2. 选择 Overview 选项卡。
3. 在 Content Sync 下，切换 Synchronize 按钮，以执行 Optimized Sync 或 Complete Sync 来同步刷新内容计数的 Capsule。
4. 选择内容选项卡。
5. 点 > 来查看这些胶囊可用的内容视图的环境。
6. 点 > 查看可用于内容视图和环境的特定版本的仓库，以展开内容视图和环境的特定版本。
7. 查看 Packages 中特定于 yum 仓库的内容数量。
8. 在附加内容下查看勘误表、软件包组、文件、容器标签、容器清单和 Ansible 集合的数量。
9. 点环境旁边的列中的垂直 ellipsis，然后点 Refresh counts 刷新 Packages 下同步的内容计数。

4.16. 配置 SELINUX 以允许自定义端口上的内容同步

SELinux 仅允许 Satellite 访问特定端口上的内容同步。默认情况下，允许连接到以下端口上运行的 Web 服务器：80, 81, 443, 488, 8008, 8009, 8443, 和 9000。

流程

1. 在 Satellite 上，要验证 SELinux 允许进行内容同步的端口，请输入以下命令：

```
# semanage port -l | grep ^http_port_t
http_port_t  tcp    80, 81, 443, 488, 8008, 8009, 8443, 9000
```

2. 要将 SELinux 配置为允许内容同步的端口，如 10011，请输入如下命令：

```
# semanage port -a -t http_port_t -p tcp 10011
```

4.17. 恢复损坏的软件仓库

如果存储库崩溃，您可以使用高级同步来恢复它，该同步有三个选项：

优化的同步

同步存储库，绕过没有检测到的与上游软件包不同的软件包。

完成同步

同步所有软件包，无论检测到的更改是什么。如果特定软件包无法下载到本地存储库，请使用此选项，即使它们存在于上游存储库中。

验证内容检查

同步所有软件包，然后在本地验证所有软件包的校验和。如果 RPM 的校验和与上游不同，它会重新下载 RPM。此选项仅适用于 Yum 内容。如果您有以下错误之一，请使用这个选项：

- 在与 yum 同步时，特定软件包会导致 404 错误。
- 软件包与预期的下载错误不匹配，这意味着特定软件包已损坏。

流程

1. 在 Satellite Web UI 中，进入到 Content >Products。
2. 选择包含已损坏的存储库的产品。
3. 选择您要同步的存储库的名称。
4. 要执行优化的同步或完成同步，请从 Select Action 菜单中选择 Advanced Sync。
5. 选择所需选项，然后单击 Sync。
6. 可选：要验证校验和，请点击 Select Action 菜单中的 Verify Content Checksum。

CLI 过程

1. 获取存储库 ID 列表：

```
# hammer repository list \
--organization "My_Organization"
```

2. 使用所需选项同步损坏的存储库：

- 对于优化的同步：

```
# hammer repository synchronize \
--id My_ID
```

- 要完成同步：

```
# hammer repository synchronize \
--id My_ID \
--skip-metadata-check true
```

- 验证内容同步：

```
# hammer repository synchronize \
--id My_ID \
--validate-contents true
```

4.18. 重新发布存储库元数据

当存储库分发没有应根据存储库内容分发的内容时，您可以重新发布存储库元数据。

请谨慎使用此流程。红帽建议完整的存储库同步或发布新的内容视图版本来修复损坏的元数据。

流程

1. 在 Satellite Web UI 中，进入到 Content >Products。
2. 选择包含您要重新发布元数据的存储库的产品。
3. 在 Repositories 选项卡上，选择一个存储库。
4. 要重新发布存储库的元数据，请点击 Select Action 菜单中的 Republish Repository Metadata。



注意

此操作不适用于使用 Complete Mirroring 策略的存储库，因为元数据是从存储库的上游来源复制的。

4.19. 重新发布内容视图元数据

使用这个流程重新发布内容视图元数据。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 选择内容视图。
3. 在 Versions 选项卡中，选择一个内容视图版本。
4. 要重新发布内容视图版本的元数据，请点击垂直 ellipsis 图标中的 Republish repository metadata。

重新发布存储库元数据将会为内容视图版本中的所有存储库重新生成元数据，这些存储库不遵循完全镜像策略。

4.20. 添加 HTTP 代理

使用此流程将 HTTP 代理添加到 Satellite。然后，您可以指定要用于产品、存储库和支持的计算资源的 HTTP 代理。

先决条件

您的 HTTP 代理必须允许访问以下主机：

主机名	端口	协议
subscription.rhsm.redhat.com	443	HTTPS
cdn.redhat.com	443	HTTPS
*.akamaiedge.net	443	HTTPS
cert.console.redhat.com (如果使用 Red Hat Insights)	443	HTTPS
api.access.redhat.com (如果使用 Red Hat Insights)	443	HTTPS
cert-api.access.redhat.com (如果使用 Red Hat Insights)	443	HTTPS

如果 Satellite 服务器使用代理与 subscription.rhsm.redhat.com 和 cdn.redhat.com 通信，则代理不得在这些通信上执行 SSL 检查。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Infrastructure > HTTP Proxies 并选择 New HTTP Proxy。
2. 在 Name 字段中输入 HTTP 代理的名称。
3. 在 URL 字段中，输入 HTTP 代理的 URL，包括端口号。
4. 如果您的 HTTP 代理需要身份验证，输入用户名和密码。
5. 可选：在 Test URL 字段中，输入 HTTP 代理 URL，然后点 Test Connection 以确保您可以从 Satellite 连接到 HTTP 代理。
6. 点 Locations 标签页并添加一个位置。
7. 点 Organization 标签页并添加一个机构。
8. 点 Submit。

CLI 过程

- 在 Satellite 服务器上，输入以下命令添加 HTTP 代理：

```
# hammer http-proxy create \
  --name proxy-name \
  --url proxy-URL:port-number
```

如果您的 HTTP 代理需要身份验证，请添加 `--username 名称` 和 `--password 密码` 选项。

如需更多信息，请参阅知识库文章如何 [通过红帽客户门户网站上的防火墙或代理访问 Red Hat Subscription Manager \(RHSM\)](#)。

4.21. 更改产品的 HTTP 代理策略

对于对网络流量的精细控制，您可以为每个产品设置 HTTP 代理策略。产品 HTTP 代理策略适用于产品中的所有存储库，除非您为各个存储库设置了不同的策略。

要为独立存储库设置 HTTP 代理策略，请参阅 [第 4.22 节“更改存储库的 HTTP 代理策略”](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products 并选择您要更改的每个 Products 旁边的复选框。
2. 在 Select Action 列表中，选择 Manage HTTP Proxy。
3. 从列表中选择 HTTP Proxy Policy：
 - 全局默认值：使用全局默认代理设置。
 - 无 HTTP 代理：不使用 HTTP 代理，即使配置了全局默认代理。
 - 使用特定的 HTTP 代理：从列表中选择 HTTP 代理。您必须将 HTTP 代理添加到 Satellite，然后才能从此列表中选择代理。更多信息请参阅 [第 4.20 节“添加 HTTP 代理”](#)。
4. 点 Update。

4.22. 更改存储库的 HTTP 代理策略

对于对网络流量的精细控制，您可以为每个存储库设置 HTTP 代理策略。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

要为产品中的所有软件仓库设置相同的 HTTP 代理策略，请参阅 [第 4.21 节“更改产品的 HTTP 代理策略”](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products，再点包含该仓库的 Product 的名称。
2. 在 Repositories 选项卡中，单击存储库的名称。
3. 找到 HTTP Proxy 字段并点编辑图标。
4. 从列表中选择 HTTP Proxy Policy :
 - 全局默认值 : 使用全局默认代理设置。
 - 无 HTTP 代理 : 不使用 HTTP 代理，即使配置了全局默认代理。
 - 使用特定的 HTTP 代理 : 从列表中选择 HTTP 代理。您必须将 HTTP 代理添加到 Satellite，然后才能从此列表中选择代理。更多信息请参阅 [第 4.20 节“添加 HTTP 代理”](#)。
5. 点击 Save。

CLI 过程

- 在 Satellite 服务器上，输入以下命令指定您要使用的 HTTP 代理策略：

```
# hammer repository update \
--http-proxy-policy HTTP_Proxy_Policy \
--id Repository_ID
```

为 `--http-proxy-policy` 指定以下选项之一：

- `none` : 不使用 HTTP 代理，即使配置了全局默认代理。
- `global_default_http_proxy` : 使用全局默认代理设置。
- `use_selected_http_proxy`: 使用 `--http-proxy My_HTTP_Proxy_Name` 或 `--http-proxy-id My_HTTP_Proxy_ID` 指定 HTTP 代理。要向 Satellite 添加新 HTTP 代理，请参阅 [第 4.20 节“添加 HTTP 代理”](#)。

4.23. 创建迁移计划

迁移计划检查并更新内容在计划的日期和时间。在 Satellite 中，您可以创建一个迁移计划并将产品分配给计划。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Sync Plans 并点 New Sync Plan。

2. 在 Name 字段中输入计划的名称。
3. 可选：在 Description 字段中输入计划的描述。
4. 从 Interval 列表中，选择您希望计划运行的时间间隔。
5. 从 Start Date 和 Start Time 列表中，选择何时开始运行同步计划。
6. 点击 Save。

CLI 过程

1. 运行以下命令来创建同步计划：

```
# hammer sync-plan create \
--description "My_Description" \
--enabled true \
--interval daily \
--name "My_Products" \
--organization "My_Organization" \
--sync-date "2023-01-01 01:00:00"
```

2. 查看机构的可用迁移计划，以验证已创建了迁移计划：

```
# hammer sync-plan list --organization "My_Organization"
```

4.24. 为产品分配迁移计划

迁移计划检查并更新内容在计划的日期和时间。在 Satellite 中，您可以为产品分配迁移计划以定期更新内容。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 选择一个产品。
3. 在 Details 标签页中，从下拉菜单中选择 Sync Plan。

CLI 过程

1. 为产品分配迁移计划：

```
# hammer product set-sync-plan \
--name "My_Product_Name" \
--organization "My_Organization" \
--sync-plan "My_Sync_Plan_Name"
```

4.25. 为多个产品分配迁移计划

使用这个流程，为机构中的产品分配迁移计划，至少一次同步并包含一个存储库。

流程

1. 运行以下 Bash 脚本：

```
ORG="My_Organization"
SYNC_PLAN="daily_sync_at_3_a.m"

hammer sync-plan create --name $SYNC_PLAN --interval daily --sync-date "2023-04-5
03:00:00" --enabled true --organization $ORG
for i in $(hammer --no-headers --csv --csv-separator="|" product list --organization
$ORG --per-page 999 | grep -vi not_synced | awk -F'|' '$5 != "0" { print $1}')
do
  hammer product set-sync-plan --sync-plan $SYNC_PLAN --organization $ORG --id $i
done
```

2. 执行脚本后，查看分配给迁移计划的产品：

```
# hammer product list --organization $ORG --sync-plan $SYNC_PLAN
```

4.26. 迁移计划的最佳实践

- 向产品添加同步计划，并定期同步内容，以便在同步期间保持对 Satellite 的负载。同步内容，而不是更频繁地进行同步。例如，设置一个同步计划，以便每天同步内容，而不是只在一个月一次同步一次。
- 使用 Hammer 脚本或 [Ansible playbook](#) 自动创建或更新同步计划。
- 使用 Custom Cron 工具创建多个同步计划，在几小时内分发同步任务，以减少任务负载。

表 4.1. Cron 表达式示例

Cron 表达式	解释
0 22 * * 1-5	每天 22:00（从星期一到周五）
30 3 * * 6,0	每天为 03:30 个
30 2 8-14 * *	每天 02:30 到月第 14 天之间每天

4.27. 限制同步并发

默认情况下，每个存储库同步作业一次可最多获取 10 个文件。这可以基于每个存储库进行调整。

增加限制可能会提高性能，但可能会导致上游服务器过载或开始拒绝请求。如果您遇到因为上游服务器拒绝请求而看到存储库同步失败，您可能需要尝试降低限制。

CLI 过程

```
# hammer repository update \
--download-concurrency 5 \
--id Repository_ID \
--organization "My_Organization"
```

4.28. 导入自定义 GPG 密钥

当客户端消耗签名的自定义内容时，请确保客户端被配置为使用适当的 GPG 密钥验证软件包安装。这有助于确保只能安装来自授权源的软件包。

红帽内容已配置了适当的 GPG 密钥，因此不支持红帽存储库的 GPG 密钥管理。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

先决条件

确保您有一个 GPG 密钥的副本，用于为要在 Satellite 中使用和管理的 RPM 内容签名。大多数 RPM 分发提供程序在其网站上提供其 GPG 密钥。您还可以从 RPM 手动提取它：

1. 将版本特定存储库软件包的副本下载到您的本地机器中：

```
$ wget http://www.example.com/9.5/example-9.5-2.noarch.rpm
```

2. 提取 RPM 文件而不安装该文件：

```
$ rpm2cpio example-9.5-2.noarch.rpm | cpio -idmv
```

GPG 密钥位于 `etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE-95`。

流程

1. 在 Satellite Web UI 中，进入到 Content > Content Credentials，并在窗口右上角点 Create Content Credential。
2. 输入存储库的名称，然后从 Type 列表中选择 GPG Key。
3. 将 GPG 密钥粘贴到 Content Credential Contents 字段中，或者点 Browse 并选择您要导入的 GPG 密钥文件。

如果您的自定义存储库包含由多个 GPG 密钥签名的内容，您必须在 Content Credential Contents 字段中输入每个密钥之间的新行，例如：

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFy/HE4BEADttv2TCPzVrre+aJ9f5QsR6oWZMm7N5Lwxjm5x5zA9BLiPPGFN
4aTUR/g+K1S0aqCU+ZS3Rnxb+6fnBxD+COH9kMqXHi3M5UNzbp5WhCdUpISXjipU
XIFFWBPuBfyr/FKRknFH15P+9kLZLxCpVZZLsweLWCuw+JKCMmnA
=F6VG
-----END PGP PUBLIC KEY BLOCK-----

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFw467UBEACmREzDeK/kuScCmfJfHJa0Wgh/2fbJLLt3KSvsgDhORlptf+PP
OTFDIKuLkXj99ZYG5xMnBG47C7ByoMec1j94YeXczuBbynOyyPlvduma/zf8oB9e
Wl5GnzcLGAAnUSRamfqGUWcyMMinHHIKIc1X1P4I=
=WPpl
-----END PGP PUBLIC KEY BLOCK-----
```

4. 点击 Save。

CLI 过程

1. 将 GPG 密钥复制到 Satellite 服务器中：

```
$ scp ~/etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE-95 root@satellite.example.com:~/.
```

2. 将 GPG 密钥上传到 Satellite：

```
# hammer content-credentials create \  
--content-type gpg_key \  
--name "My_GPG_Key" \  
--organization "My_Organization" \  
--path ~/RPM-GPG-KEY-EXAMPLE-95
```

4.29. 将自定义存储库限制为 SATELLITE 中的特定操作系统或架构

您可以配置 Satellite，使其仅在具有特定操作系统版本或架构的主机上提供自定义存储库。例如，您只能将自定义软件仓库限制为 Red Hat Enterprise Linux 9 主机。



注意

仅限制自定义产品的架构和操作系统版本。Satellite 对红帽软件仓库自动应用这些限制。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 点包含您要限制的存储库集的产品。
3. 在 Repositories 选项卡中，点您要限制的存储库。
4. 在 Publishing Settings 部分中，设置以下选项：
 - 将 Restrict to OS 版本 设置为限制操作系统版本。
 - 将 Restrict to architecture 设置为限制架构。

第 5 章 限制主机对内容的访问

Satellite 提供多个选项来限制主机对内容的访问。要为主机授予对 Satellite 管理的内容的特定子集的访问权限，您可以使用以下策略：红帽建议考虑按此处列出的顺序实施策略：

内容视图和生命周期环境

根据需要使用内容视图和生命周期环境，并整合内容视图过滤器。
有关内容视图的更多信息，请参阅 [第 7 章 管理内容视图](#)。

有关生命周期环境的更多信息，请参阅 [第 6 章 管理应用程序生命周期](#)。

内容覆盖

默认情况下，Satellite 托管的内容可以启用或禁用。在自定义产品中，默认情况下，存储库总是被禁用，而红帽产品可以根据特定的软件仓库默认启用或禁用。启用存储库可让主机访问存储库软件包或其他内容，允许主机下载并安装可用内容。

如果禁用了存储库，则主机无法访问存储库内容。内容覆盖为您提供了选项来覆盖任何存储库的 Enabled 或 Disabled 的默认启用值。您可以向主机或激活码添加内容覆盖。

有关向主机添加内容覆盖的更多信息，[请参阅管理主机中的在主机上启用和禁用存储库](#)。

有关在激活码中添加内容覆盖的详情，请参考 [第 9.7 节 “在激活码中启用和禁用软件仓库”](#)。

复合内容视图

您可以使用复合内容视图来组合主机并授予主机对来自多个内容视图的内容的访问权限。有关复合内容视图的更多信息，请参阅 [第 7.8 节 “创建复合内容视图”](#)。

架构和操作系统版本限制

在自定义产品中，您可以为产品可用的 yum 存储库设置架构和操作系统版本限制。例如，如果您将自定义软件仓库限制为 Red Hat Enterprise Linux 8，则仅在运行 Red Hat Enterprise Linux 8 的主机上可用。架构和操作系统版本限制在所有其他策略中均具有最高优先级。它们无法被内容覆盖或无效，对内容视图的更改或更改生命周期环境。因此，建议您考虑在使用构架或操作系统版本限制前提到的其他策略。红帽软件仓库自动设置架构和操作系统版本限制。

发行版本

某些红帽软件仓库（如 Red Hat Enterprise Linux dot 发行版本仓库）在其仓库元数据中包含发行版本。然后，发行版本会与主机系统目的属性中指定的发行版本进行比较。根据这个比较，对内容的访问可能会受限制或限制。有关设置系统目的属性的更多信息，请参阅管理主机中的 [在 Red Hat Satellite 中创建主机](#)。

整合所有策略

只有以下所有项都为 true 时，主机才有特定的软件包或存储库：

- 存储库包含在主机内容视图和生命周期环境中。
- 在将存储库添加到后，主机内容视图已被发布。
- 存储库未被内容视图过滤器过滤掉。
- 存储库默认是启用的，或使用内容覆盖覆盖为 Enabled。
- 存储库没有架构或操作系统版本限制，或者其具有与主机匹配的构架或操作系统版本限制。
- 对于某些红帽软件仓库，没有设置发行版本，或者与主机的发行版本匹配。

使用激活码

使用激活码可以简化其中一些策略的工作流。您可以使用激活码执行以下操作：

- 将主机分配到内容视图和生命周期环境。
- 向主机添加内容覆盖。
- 在主机上设置系统目的属性，包括发行版本。

激活码只在注册期间影响主机。如果已经注册了主机，可以为每个主机单独更改上述属性，或者通过内容主机批量操作单独更改。如需更多信息，[请参阅管理内容中的管理 激活码](#)。

第 6 章 管理应用程序生命周期

本章概述了 Satellite 中的应用程序生命周期以及如何为 Satellite 和 Capsule 创建和删除应用程序生命周期。

6.1. 应用程序生命周期简介

*应用程序生命周期*是 Satellite 的内容管理功能的核心概念。应用程序生命周期定义了特定系统及其软件如何查看特定阶段。例如，应用程序生命周期可能很简单；您可能只有一个开发阶段和产品阶段。在这种情况下，应用程序生命周期可能类似如下：

- 开发
- Production

但是，更复杂的应用程序生命周期可能会有进一步的阶段，如用于测试或 beta 版本的阶段。这会在应用程序生命周期中添加额外的阶段：

- 开发
- 测试
- Beta 版本
- Production

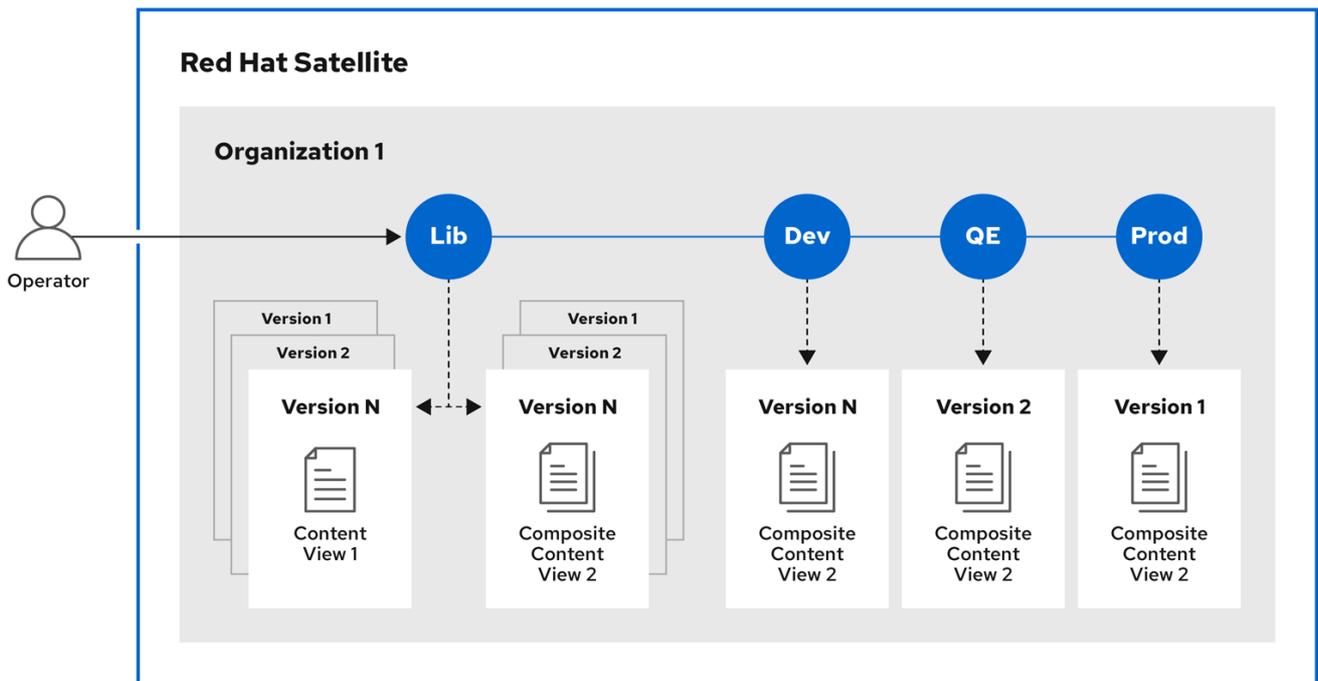
Satellite 提供了自定义每个应用程序生命周期阶段的方法，使其适合您的规格。

应用程序生命周期中的每个阶段都在 Satellite 中称为 *环境*。每个环境都使用特定的内容集合。Satellite 将这些内容集合定义为内容视图。每个内容视图都充当一个过滤器，您可以在其中定义特定环境中要包含哪些存储库和软件包。这为您提供了定义特定内容集到每个环境的方法。

例如，电子邮件服务器可能只需要一个简单的应用程序生命周期，其中您有用于真实用途的生产级服务器以及用于尝试最新邮件服务器软件包的测试服务器。当测试服务器通过初始阶段时，您可以将生产级服务器设置为使用新软件包。

另一个例子是软件产品的开发生命周期。要在开发环境中开发新软件，请在质量保证环境中进行测试，预发布为测试版，然后将软件作为生产级应用发布。

图 6.1. Satellite 应用程序生命周期



278_Satellite_0922

6.2. 应用程序生命周期的内容提升

在应用程序生命周期链中，当内容从一个环境移到下一个环境时，这称为 **提升**。

示例：Satellite 生命周期环境的内容提升

每个环境都包含一组注册到 Red Hat Satellite 的系统。这些系统只能访问与其环境相关的存储库。当您 将软件包从一个环境提升到下一个环境时，目标环境的存储库会接收新的软件包版本。因此，目标环境中的 每个系统都可以更新至新的软件包版本。

开发	测试	Production
<code>example_software-1.1-0.noarch.rpm</code>	<code>example_software-1.0-0.noarch.rpm</code>	<code>example_software-1.0-0.noarch.rpm</code>

完成补丁的开发后，您可以将软件包提升到测试环境，以便 Quality Engineering 团队可以查看补丁。然后，应用程序生命周期在每个环境中包含以下软件包版本：

开发	测试	Production
<code>example_software-1.1-0.noarch.rpm</code>	<code>example_software-1.1-0.noarch.rpm</code>	<code>example_software-1.0-0.noarch.rpm</code>

虽然 Quality Engineering 团队审查补丁，但开发团队开始处理 `example_software 2.0`。这会生成以下应用程序生命周期：

开发	测试	Production
<code>example_software-2.0-0.noarch.rpm</code>	<code>example_software-1.1-0.noarch.rpm</code>	<code>example_software-1.0-0.noarch.rpm</code>

Quality Engineering 团队完成对补丁的审查。现在 `example_software` 1.1 已准备好发布。您可以将 1.1 提升到 *Production* 环境：

开发	测试	Production
<code>example_software-2.0-0.noarch.rpm</code>	<code>example_software-1.1-0.noarch.rpm</code>	<code>example_software-1.1-0.noarch.rpm</code>

Development 团队完成其对 `example_software` 2.0 的工作，并将其提升到测试环境：

开发	测试	Production
<code>example_software-2.0-0.noarch.rpm</code>	<code>example_software-2.0-0.noarch.rpm</code>	<code>example_software-1.1-0.noarch.rpm</code>

最后，Qe Quality Engineering 团队审查软件包。成功检查后，将软件包提升到 *Production* 环境：

开发	测试	Production
<code>example_software-2.0-0.noarch.rpm</code>	<code>example_software-2.0-0.noarch.rpm</code>	<code>example_software-2.0-0.noarch.rpm</code>

如需更多信息，请参阅 [第 7.6 节“提升内容视图”](#)。

6.3. 生命周期环境的最佳实践

- 使用多个生命周期环境路径实施内容消耗的多个后续阶段。每个阶段都包含一组定义的内容，例如在 *Production* 生命周期环境中。
- 使用 Hammer 脚本或 [Ansible playbook](#) 自动创建生命周期环境。
- 默认用例：修复每个生命周期环境路径中的阶段，如 *Development*、*Test* 和 *Production*。
 - 将内容视图提升到生命周期环境，例如从 *Test* 提升到 *Production*。消耗此内容视图或复合内容视图的所有内容主机都能够从 *Production* 生命周期环境安装软件包。请注意，这些软件包不会被自动安装或更新。
 - 如果您在修补内容主机过程中遇到错误，请将主机附加到内容视图的以前的版本。这只会影响软件包的可用性，但不会降级安装的软件包。
- 备选用例：在生命周期阶段为固定内容使用阶段，例如每季度更新，且仅使用勘误表的增量更新发布新的次版本。
 - 当修补内容主机时，使用 Satellite Web UI、Satellite API、Hammer CLI 或激活码将生命周

期环境从 2023-Q4 更改为 2024-Q1。

- 优点：您可以通过查看其生命周期环境直接查看主机接收的软件包。
- 缺点：在没有明确定义的阶段（如开发、测试和生产）时，提升内容的动态。
- 使用多个生命周期环境路径为不同的环境定义多个阶段，例如分离 Web 服务器和数据库主机。
- 胶囊服务器使用生命周期环境同步内容。如果您将内容分成多个生命周期环境路径，它们可以更有效地同步内容。如果特定的胶囊服务器只在单一生命周期环境路径中为一个操作系统提供内容，则它只同步所需的内容。

6.4. 创建生命周期环境路径

要为开发和释放软件创建应用程序生命周期，请使用 Library 环境作为初始环境来创建环境路径。然后（可选）在环境路径中添加额外的环境。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Lifecycle Environments。
2. 点 New Environment Path 启动新的应用程序生命周期。
3. 在 Name 字段中输入您的环境的名称。
4. 在 Description 字段中，输入您的环境的描述。
5. 点击 Save。
6. 可选：要在环境路径中添加环境，请点 Add New Environment，完成 Name 和 Description 字段，然后从 Prior Environment 列表中选择之前的环境。

CLI 过程

1. 要创建环境路径，请输入 `hammer lifecycle-environment create` 命令，并使用 `--prior` 选项指定库环境：

```
# hammer lifecycle-environment create \
--name "Environment Path Name" \
--description "Environment Path Description" \
--prior "Library" \
--organization "My_Organization"
```

2. 可选：要在环境路径中添加环境，请输入 `hammer lifecycle-environment create` 命令，并使用 `--prior` 选项指定父环境：

```
# hammer lifecycle-environment create \
--name "Environment Name" \
--description "Environment Description" \
--prior "Prior Environment Name" \
--organization "My_Organization"
```

3. 要查看生命周期环境的链，请输入以下命令：

```
# hammer lifecycle-environment paths --organization "My_Organization"
```

6.5. 在 CAPSULE 服务器中添加生命周期环境

如果您的胶囊服务器启用了内容功能，您必须添加一个环境，以便 Capsule 可以从 Satellite 服务器同步内容，并将内容提供给主机系统。

不要将库生命周期环境分配给您的胶囊服务器，因为它在每次 CDN 更新存储库时触发自动胶囊同步。这可能会消耗 Capsules 上的多个系统资源、Satellite 和 Capsule 之间的网络带宽，以及 Capsules 上的可用磁盘空间。

您可以在 Satellite 服务器或 Satellite Web UI 上使用 Hammer CLI。

流程

1. 在 Satellite Web UI 中，进入到 Infrastructure > Capsules，然后选择您要向其添加生命周期的 Capsule。
2. 单击 Edit，再单击 Lifecycle Environments 选项卡。
3. 在左侧菜单中选择您要添加到胶囊的生命周期环境，然后点 Submit。
4. 要同步胶囊中的内容，请单击 Overview 选项卡，再单击 Synchronize。
5. 选择 Optimized Sync 或 Complete Sync。
有关每种同步类型的定义，请参阅恢复存储库。

CLI 过程

1. 要显示所有 Capsule 服务器的列表，请在 Satellite 服务器上输入以下命令：

```
# hammer capsule list
```

记录下您要将生命周期添加到的胶囊 ID。

2. 使用 ID 验证 Capsule 的详情：

```
# hammer capsule info \  
--id My_capsule_ID
```

3. 要查看 Capsule 服务器可用的生命周期环境，请输入以下命令并记录 ID 和机构名称：

```
# hammer capsule content available-lifecycle-environments \  
--id My_capsule_ID
```

4. 将生命周期环境添加到 Capsule 服务器中：

```
# hammer capsule content add-lifecycle-environment \  
--id My_capsule_ID \  
--lifecycle-environment-id My_Lifecycle_Environment_ID \  
--organization "My_Organization"
```

对您要添加到 Capsule 服务器的每个生命周期环境重复此操作。

5. 将 Satellite 的内容同步到 Capsule。
 - 要将 Satellite 服务器环境中的所有内容同步到 Capsule 服务器，请输入以下命令：

```
# hammer capsule content synchronize \  
--id My_capsule_ID
```

- 要将 Satellite 服务器的特定生命周期环境同步到 Capsule 服务器，请输入以下命令：

```
# hammer capsule content synchronize \  
--id My_capsule_ID \  
--lifecycle-environment-id My_Lifecycle_Environment_ID
```

- 在不检查元数据的情况下，将 Satellite 服务器中的所有内容同步到您的 Capsule 服务器：

```
# hammer capsule content synchronize \  
--id My_capsule_ID \  
--skip-metadata-check true
```

这等同于在 Satellite Web UI 中选择 Complete Sync。

6.6. 从 SATELLITE 服务器中删除生命周期环境

使用这个流程删除生命周期环境。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle Environments。
2. 单击您要删除的生命周期环境的名称，然后单击 Remove Environment。
3. 单击 Remove 以删除环境。

CLI 过程

1. 列出机构的生命周期环境，并记录您要删除的生命周期环境的名称：

```
# hammer lifecycle-environment list \  
--organization "My_Organization"
```

2. 使用 **hammer lifecycle-environment delete** 命令删除环境：

```
# hammer lifecycle-environment delete \  
--name "My_Environment" \  
--organization "My_Organization"
```

6.7. 从 CAPSULE 服务器中删除生命周期环境

当生命周期环境不再与主机系统或环境相关的错误添加到 Capsule 服务器时，您可以从 Capsule 服务器中删除生命周期环境。

您可以使用 Satellite Web UI 和 Hammer CLI 从 Capsule 中删除生命周期环境。

流程

1. 在 Satellite Web UI 中，进入到 Infrastructure > Capsules，然后选择您要从中删除生命周期的 Capsule。
2. 单击 Edit，再单击 Lifecycle Environments 选项卡。
3. 在右侧菜单中选择您要删除的生命周期环境，然后点 Submit。
4. 要同步 Capsule 的内容，点 Overview 选项卡，然后点 Synchronize。
5. 选择 Optimized Sync 或 Complete Sync。

CLI 过程

1. 从列表中选择您想的胶囊服务器，并记录其 id：

```
# hammer capsule list
```

2. 要验证 Capsule 服务器的详情，请输入以下命令：

```
# hammer capsule info \  
--id My_Capsule_ID
```

3. 验证当前附加到 Capsule 服务器的生命周期环境列表，并记录环境 ID：

```
# hammer capsule content lifecycle-environments \  
--id My_Capsule_ID
```

4. 从 Capsule 服务器中删除生命周期环境：

```
# hammer capsule content remove-lifecycle-environment \  
--id My_Capsule_ID \  
--lifecycle-environment-id My_Lifecycle_Environment_ID
```

对您要从 Capsule 服务器中删除的每个生命周期环境重复此步骤。

5. 将 Satellite 服务器环境的内容同步到 Capsule 服务器：

```
# hammer capsule content synchronize \  
--id My_Capsule_ID
```

第 7 章 管理内容视图

Red Hat Satellite 使用内容视图来允许主机访问独立策展的内容子集。要做到这一点，您必须定义要使用的存储库，然后将某些过滤器应用到内容。

创建用于过滤和创建快照的内容视图的一般工作流程如下：

1. 创建内容视图。
2. 将一个或多个您要添加到内容视图的存储库。
3. 可选：创建一个或多个过滤器来优化内容视图的内容。如需更多信息，请参阅 [第 7.12 节“内容过滤器示例”](#)。
4. 可选：解决内容视图的任何软件包依赖项。如需更多信息，请参阅 [第 7.10 节“解决软件包依赖项”](#)。
5. 发布内容视图。
6. 可选：将内容视图提升到另一个环境。如需更多信息，请参阅 [第 7.6 节“提升内容视图”](#)。
7. 将内容主机附加到内容视图。

如果存储库没有与内容视图关联，则文件 `/etc/yum.repos.d/redhat.repo` 会保持空，并且注册到它的系统无法接收更新。

主机只能与单个内容视图关联。要将主机与多个内容视图关联，请创建一个复合内容视图。如需更多信息，请参阅 [第 7.8 节“创建复合内容视图”](#)。

7.1. RED HAT SATELLITE 中的内容视图

内容视图是主机可访问的内容的一个独立策展子集。通过创建内容视图，您可以定义特定环境或胶囊服务器使用的软件版本。

每个内容视图在每个环境中创建一组存储库。您的 Satellite 服务器存储并管理这些存储库。例如，您可以使用以下方法创建内容视图：

- 生产环境的旧软件包版本的内容视图，以及开发环境的较新软件包版本的另一个内容视图。
- 具有操作系统所需软件包存储库的内容视图，以及包含应用所需软件包存储库的另一个内容视图。
- 用于模块化方法管理内容视图的复合内容视图。例如，您可以使用一个内容视图来管理操作系统，以及使用另一个内容视图来管理应用。通过创建一个组合了这两个内容视图的复合内容视图，您可以创建一个从每个内容视图合并存储库的新存储库。但是，内容视图的存储库仍然存在，您也可以单独管理它们。

默认机构视图

Default Organization View 是与 Satellite 同步的所有内容的应用程序控制的内容视图。使用 Default Organization View，您可以使用订阅将主机注册到 Satellite 并访问内容，而无需操作内容视图和生命周期环境。

在环境间提升内容视图

当您内容视图从一个环境提升到应用程序生命周期中的下一个环境时，Satellite 会更新存储库并发布软件包。

例 7.1. 将软件包从 Development 提升到测试

测试和生产的存储库包含 `<my_software>-1.0-0.noarch.rpm` 软件包：

	开发	测试	Production
内容视图的版本	版本 2	版本 1	版本 1
内容视图的内容	<code><my_software>-1.1-0.noarch.rpm</code>	<code><my_software>-1.0-0.noarch.rpm</code>	<code><my_software>.0-0.noarch.rpm</code>

如果您将版本 2 的内容视图从 Development 提升到 Testing，则测试的存储库包含 `<my_software>-1.1-0.noarch.rpm` 软件包：

	开发	测试	Production
内容视图的版本	版本 2	版本 2	版本 1
内容视图的内容	<code><my_software>-1.1-0.noarch.rpm</code>	<code><my_software>-1.1-0.noarch.rpm</code>	<code><my_software>-1.0-0.noarch.rpm</code>

这样可确保主机被指定为特定的环境，但当环境使用新版本的内容视图时接收更新。

7.2. 内容视图的最佳实践

- 捆绑内容（如 Red Hat Enterprise Linux）以及其他软件（如 Apache-2.4 或 PostgreSQL-16.2）的内容视图更易于维护。太小的内容视图需要更多维护”。
- 如果您需要每日更新内容，请使用内容视图 **Default Organization View**，其中包含来自所有存储库的最新同步内容，并在 Library 生命周期中提供。
- 将复合内容视图限制为需要更大的灵活性的情况，例如，如果您每周更新一个内容视图以及每月的另一内容视图。
- 如果您使用复合内容视图，首先发布内容视图，然后发布复合内容视图。您捆绑到复合内容视图中的更多内容视图，需要更努力更改或更新内容。
- 如果内容视图仅捆绑到复合内容视图，则不需要为内容视图设置生命周期环境。
- 使用 Hammer 脚本或 Ansible playbook 自动创建和发布复合内容视图和生命周期环境。使用 cron 作业、systemd 计时器或重复逻辑来获得更可见性。
- 将更改和日期添加到每个发布的内容视图或复合内容视图版本的描述中。无论内容本身的最新更改，都会在 Satellite Web UI 中按日期显示最新的活动，如将内容移到新的生命周期环境。
- 发布新的内容视图或复合内容视图将创建新的主要版本。增量勘误更新会递增次版本。请注意，您无法更改或重置这个计数器。

7.3. 修补内容主机的最佳实践

- 将主机注册到 Satellite 需要 Satellite 客户端 6，其中包含 `subscription-manager` 软件包、`katello-host-tools` 软件包及其依赖项。如需更多信息，请参阅[管理主机中的注册主机](#)。
- 使用 Satellite Web UI 从主机安装、升级和删除软件包。您可以使用 SSH 和 Ansible 使用作业模板更新内容主机。
- 使用 Satellite Web UI 在内容主机上应用勘误表。当使用默认软件包管理器在主机上修补软件包时，Satellite 会收到软件包和存储库列表，以重新计算适用的勘误表和可用更新。
- 修改或替换作业模板以添加自定义步骤。这可让您在主机上运行命令或执行命令。
- 在主机上运行批量操作时，使用主要操作系统版本捆绑它们，特别是在升级软件包时。
- 选择 `via remote execution - 自定义`，以定义在执行批量操作时将补丁应用到主机的时间。
- 您不能将勘误表应用到不是 Satellite 上存储库一部分的软件包以及附加内容视图的软件包。
- 使用 `rpm` 或 `dpkg` 对安装的软件包进行修改会通过下一次运行 `apt`、`yum`、`zypper` 发送到 Satellite。

7.4. 创建内容视图

使用这个流程创建简单的内容视图。要使用 CLI 而不是 Satellite Web UI，请参阅[CLI 过程](#)。

先决条件

虽然您可以基于内容视图，确定是否要根据内容视图解析任何软件包依赖项，但您可能希望更改默认的 Satellite 设置以为所有内容视图启用或禁用软件包解析。如需更多信息，请参阅[第 7.10 节“解决软件包依赖项”](#)。

流程

1. 在 Satellite Web UI 中，进入到 `Content > Lifecycle > Content Views`。
2. 单击 `Create content view`。
3. 在 `Name` 字段中输入视图的名称。Satellite 会根据您输入的名称自动完成 `Label` 字段。
4. 在 `Description` 字段中，输入视图的描述。
5. 在 `Type` 字段中，选择 `Content view` 或 `Composite` 内容视图。
6. 可选：如果要在每次发布此内容视图时自动解决依赖项，请选择 `Solve dependencies` 复选框。依赖项会妨碍发布时间缓慢，并可能会忽略您所使用的任何内容视图过滤器。这也可能导致在解析勘误依赖项时出现错误。
7. 单击 `Create content view`。

内容视图步骤

1. 单击 `Create content view` 以创建内容视图。
2. 在 `Repositories` 选项卡中，从您要添加到您的内容视图的 `Type` 列表中选择存储库，选中您要添加的可用存储库旁边的复选框，然后单击 `Add repositories`。
3. 点 `Publish new version`，在 `Description` 字段中输入有关日志更改的信息。

4. **可选**：您可以通过点 **Promote** 到 **Select a lifecycle environment from the available promotion paths to promote new version** 来启用一个提升。
5. 点击 **Next**。
6. 在 **Review** 页面中，您可以查看您要发布的环境。
7. 点 **Finish**。

您可以查看 **Content Views** 页面上的内容视图。要查看有关内容视图的更多信息，请点内容视图名称。要将主机注册到您的内容视图，请参阅管理 [主机](#) 中的 **注册主机**。

CLI 过程

1. 获取存储库 ID 列表：

```
# hammer repository list --organization "My_Organization"
```

2. 创建内容视图并添加软件仓库：

```
# hammer content-view create \
--description "My_Content_View" \
--name "My_Content_View" \
--organization "My_Organization" \
--repository-ids 1,2
```

对于 **--repository-ids** 选项，您可以在 **hammer repository list** 命令的输出中找到 ID。

3. 发布视图：

```
# hammer content-view publish \
--description "My_Content_View" \
--name "My_Content_View" \
--organization "My_Organization"
```

4. **可选**：要在现有内容视图中添加存储库，请输入以下命令：

```
# hammer content-view add-repository \
--name "My_Content_View" \
--organization "My_Organization" \
--repository-id repository_ID
```

Satellite 服务器创建视图的新版本，并将其发布到 Library 环境。

7.5. 查看模块流

在 Satellite 中，您可以在内容视图中查看存储库的模块流。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 **Content View > Module Streams** 的 **published** 版本，以查看可用于 **Content Types** 的模块流。

2. 使用 Search 字段搜索特定模块。
3. 要查看模块的相关信息，点模块及其对应选项卡，使其包括 Details, Repositories, Profiles, 和 Artifacts。

CLI 过程

1. 列出所有机构：

```
# hammer organization list
```

2. 查看您的机构的所有模块流：

```
# hammer module-stream list \
--organization-id My_Organization_ID
```

7.6. 提升内容视图

使用这个流程在不同生命周期环境中提升内容视图。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

内容视图提升的权限要求

非管理员用户需要两个权限才能将内容视图提升到环境：

1. `promote_or_remove_content_views`
2. `promote_or_remove_content_views_to_environment`.

`promote_or_remove_content_views` 权限限制用户可以提升的内容视图。

`promote_or_remove_content_views_to_environment` 权限限制用户可以提升内容视图的环境。

使用这些权限，您可以分配用户权限来将某些内容视图提升到某些环境，但不能分配给其他环境。例如，您可以限制一个用户，以便他们能够提升测试环境，但不能提升到生产环境。

您必须将这两个权限分配给用户，以便他们能够提升内容视图。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 选择您要提升的内容视图。
3. 选择您要提升的版本，单击垂直 ellipsis 图标，然后单击 Promote。
4. 选择您要提升内容视图的环境，然后单击 Promote。

现在，内容视图的存储库出现在所有环境中。

CLI 过程

- 为每个生命周期环境使用 Hammer 提升内容视图：

```
# hammer content-view version promote \
```

```
--content-view "Database" \
--version 1 \
--to-lifecycle-environment "Development" \
--organization "My_Organization"
# hammer content-view version promote \
--content-view "Database" \
--version 1 \
--to-lifecycle-environment "Testing" \
--organization "My_Organization"
# hammer content-view version promote \
--content-view "Database" \
--version 1 \
--to-lifecycle-environment "Production" \
--organization "My_Organization"
```

现在，数据库内容在所有环境中都可用。

- 或者，您可以使用以下 Bash 脚本在机构中的所有生命周期环境中提升内容视图：

```
ORG="My_Organization"
CVV_ID=My_Content_View_Version_ID

for i in $(hammer --no-headers --csv lifecycle-environment list --organization $ORG |
awk -F, {'print $1'} | sort -n)
do
  hammer content-view version promote --organization $ORG --to-lifecycle-
environment-id $i --id $CVV_ID
done
```

验证

- 显示内容视图版本的信息，以验证它是否已提升到所需的生命周期环境：

```
# hammer content-view version info --id My_Content_View_Version_ID
```

后续步骤

- 要将主机注册到您的内容视图，请参阅管理 [主机](#) 中的注册主机。

7.7. 复合内容视图概述

复合内容视图组合了多个内容视图的内容。例如，您可能有单独的内容视图来单独管理操作系统和应用程序。您可以使用复合内容视图将两个内容视图的内容合并到新存储库中。原始内容视图的存储库仍然存在，但组合内容也会存在新的存储库。

如果要开发支持不同数据库服务器的应用程序。example_application 显示为：

example_software
Application (应用程序)
数据库

`example_software`

操作系统

四个单独的内容视图示例：

- Red Hat Enterprise Linux (操作系统)
- PostgreSQL (Database)
- MariaDB (Database)
- `example_software` (Application)

从前面的内容视图，您可以创建两个复合内容视图。

PostgreSQL 数据库的复合内容视图示例：

复合内容视图 1 - `example_software` on PostgreSQL

`example_software` (Application)

PostgreSQL (Database)

Red Hat Enterprise Linux (操作系统)

MariaDB 的复合内容视图示例：

复合内容视图 2 - MariaDB 上的 `example_software`

`example_software` (Application)

MariaDB (Database)

Red Hat Enterprise Linux (操作系统)

然后，每个内容视图都会被单独管理和发布。当您创建应用的版本时，您要发布复合内容视图的新版本。您还可以在创建复合内容视图时选择 `Auto Publish` 选项，然后在包含的内容视图重新发布时自动重新发布复合内容视图。

仓库限制

Docker 存储库不能超过复合内容视图中的一次。例如，如果您试图在复合内容视图中使用相同的 docker 存储库包含两个内容视图，则 Satellite 服务器会报告错误。

7.8. 创建复合内容视图

使用这个流程创建复合内容视图。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 单击 Create content view。
3. 在 Create content view 窗口中，在 Name 字段中输入视图的名称。Red Hat Satellite 从您输入的名称自动完成 Label 字段。
4. 可选：在 Description 字段中输入视图的描述。
5. 在 Type 选项卡中，选择 Composite 内容视图。
6. 可选：如果您要在内容视图重新发布时自动发布复合内容视图的新版本，请选择 Auto publish 复选框。
7. 单击 Create content view。
8. 在 Content views 选项卡上，选择要添加到复合内容视图的内容视图，然后单击 Add content view。
9. 在 Add content view 窗口中，选择每个内容视图的版本。
10. 可选：如果要自动将内容视图更新至最新版本，请选择 Always update to latest version 复选框。
11. 点 Add，然后点 Publish new version。
12. 可选：在 Description 字段中输入内容视图的描述。
13. 在 Publish 窗口中，设置 Promote 开关，然后选择 生命周期环境。
14. 单击 Next，然后单击 Finish。

CLI 过程

1. 在创建复合内容视图前，列出现有内容视图的版本 ID：

```
# hammer content-view version list \
--organization "My_Organization"
```

2. 创建新的复合内容视图。当 `--auto-publish` 选项设为 `yes` 时，当包含的内容视图重新发布时，复合内容视图会自动重新发布：

```
# hammer content-view create \
--composite \
--auto-publish yes \
--name "Example_Composite_Content_View" \
--description "Example composite content view" \
--organization "My_Organization"
```

3. 将内容视图添加到复合内容视图。您可以通过其 ID 或名称来识别命令中的内容视图、内容视图版本和组织。要将多个内容视图添加到复合内容视图，请对您要包含的每个内容视图重复此步骤。

- 如果您为内容视图启用了 Always update to latest version 选项：

```
# hammer content-view component add \
```

```
--component-content-view-id Content_View_ID \
--composite-content-view "Example_Composite_Content_View" \
--latest \
--organization "My_Organization"
```

- 如果您为内容视图禁用了 *Always update to latest version* 选项：

```
# hammer content-view component add \
--component-content-view-id Content_View_ID \
--composite-content-view "Example_Composite_Content_View" \
--component-content-view-version-id Content_View_Version_ID \
--organization "My_Organization"
```

4. 发布复合内容视图：

```
# hammer content-view publish \
--name "Example_Composite_Content_View" \
--description "Initial version of composite content view" \
--organization "My_Organization"
```

5. 在所有环境中提升复合内容视图：

```
# hammer content-view version promote \
--content-view "Example_Composite_Content_View" \
--version 1 \
--to-lifecycle-environment "Development" \
--organization "My_Organization"
# hammer content-view version promote \
--content-view "Example_Composite_Content_View" \
--version 1 \
--to-lifecycle-environment "Testing" \
--organization "My_Organization"
# hammer content-view version promote \
--content-view "Example_Composite_Content_View" \
--version 1 \
--to-lifecycle-environment "Production" \
--organization "My_Organization"
```

7.9. 内容过滤器概述

内容视图也使用过滤器来包含或限制某些 Yum 内容。如果没有这些过滤器，内容视图会包括所有来自所选存储库的所有内容。

内容过滤器有两种类型：

表 7.1. 过滤类型

过滤类型	描述
Include	您从没有内容开始，然后从所选存储库中选择要添加的内容。使用此过滤器组合多个内容项目。

过滤类型	描述
exclude	从所选存储库中的所有内容开始，然后选择要删除的内容。当您想在排除某些软件包的同时使用大多数特定内容存储库时，请使用此过滤器。过滤器使用存储库中的所有内容，但您选择的内容除外。

include 和 Exclude 过滤器组合

如果使用 Include 和 Exclude 过滤器的组合，首先发布内容视图会触发 include 过滤器，然后是 exclude 过滤器。在这种情况下，选择要包含的内容，然后从包含的子集中排除的内容。

内容类型

您可以根据以下内容类型过滤内容：

表 7.2. 内容类型

内容类型	描述
RPM	根据软件包的名称和版本号过滤软件包。RPM 选项过滤非模块化 RPM 软件包和勘误表。源 RPM 不受此过滤器的影响，在内容视图中仍可用。
软件包组	根据软件包组过滤软件包。软件包组列表基于添加到内容视图的存储库。
勘误（按 ID）	选择要添加到过滤器的特定勘误。勘误列表基于添加到内容视图的存储库。
勘误（按日期和类型）	选择要添加到过滤器中的发布或更新日期范围和勘误类型 (Bugfix、enhancement 或 Security)。
模块流	选择是否包含或排除特定的模块流。Module Streams 选项过滤模块化 RPM 和勘误，但不过滤与所选模块流关联的非模块化内容。
容器镜像标签	选择是否包含或排除特定的容器镜像标签。

7.10. 解决软件包依赖项

在发布内容视图时，Satellite 可以在内容视图中添加软件包依赖项到依赖的存储库。要配置此功能，您可以启用依赖项解析。

例如，当您逐步将单个软件包添加到内容视图版本时，依赖项会很有用。您可能需要启用依赖项，以安装该软件包。

然而，在大多数情况下，依赖项无法解决。例如：

- 当逐步添加安全勘误到内容视图时，依赖项寻求显著延迟，而对内容视图发布造成大量延迟，而不会造成主要好处。
- 较新的勘误中的软件包可能会具有与旧内容视图版本中的软件包不兼容的依赖项。使用依赖项解析以递增方式添加勘误可能包括不需要的软件包。作为替代方案，请考虑更新内容视图。



注意

依赖项有助于仅考虑内容视图的存储库中的软件包。它不考虑客户端上安装的软件包。例如，如果内容视图仅包含 AppStream，依赖项需要包括在发布时不包含依赖 BaseOS 内容。

如需更多信息，请参阅管理内容中的[对存储库依赖解析的限制](#)。

依赖项解决可能会导致以下问题：

内容视图发布中的显著延迟

Satellite 会在内容视图中检查依赖项的每个存储库。因此，发布时间会增加存储库。要缓解这个问题，请使用带有较少存储库的多个内容视图，并将它们合并到复合内容视图中。

忽略依赖软件包的内容视图过滤器

Satellite 优先选择根据过滤器中的规则解析软件包依赖项。例如，如果您为安全目的创建过滤器，但启用依赖项解析，Satellite 您可以添加可能认为不安全的软件包。

要缓解这个问题，请仔细测试过滤规则以确定所需的依赖项。如果依赖项解决包括不需要的软件包，请手动识别额外的软件包和勘误所需的核心基本依赖项。

例 7.2. 将排除过滤器与依赖项解决合并

您需要使用内容视图过滤器重新创建 Red Hat Enterprise Linux 8.3，并包括后续 Red Hat Enterprise Linux 8 次版本中的所选勘误。要做到这一点，您可以在 Red Hat Enterprise Linux 8.3 发行日期后创建过滤器来排除大多数勘误，除了几个您需要的。然后，您可以实现依赖项解决。

在这种情况下，依赖项解决可能会包括比预期更多的软件包。因此，主机被认为是 Red Hat Enterprise Linux 8.3 机器。

如果您不需要额外的勘误表和软件包，请不要配置内容视图过滤。反之，在 Satellite Web UI 中的 Content > Red Hat Repositories 页面中启用并使用 Red Hat Enterprise Linux 8.3 软件仓库。

例 7.3. 排除软件包有时会使 DNF 无法进行依赖项解决

如果您制作带有一些排除的软件包的 Red Hat Enterprise Linux 8.3 软件仓库，dnf update 有时可能会失败。

不要启用依赖项来解决问题。相反，应调查 dnf 中的错误，并调整过滤器以停止排除缺少的依赖项。

否则，依赖项解决可能会导致存储库从 Red Hat Enterprise Linux 8.3 分离。

7.11. 为内容视图启用依赖项

使用这个流程启用对内容视图的依赖项。

先决条件

- 依赖项需要仅在有限的上下文中有用。在启用它前，请确定您读和理解 [第 7.10 节“解决软件包依赖项”](#)

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 从内容视图列表中，选择所需的内容视图。
3. 在 Details 标签页中，切换 Solve 依赖项。

7.12. 内容过滤器示例

使用以下任何示例来构建自定义内容过滤器。



注意

过滤器可能会显著增加发布内容视图的时间。例如，如果内容视图发布任务在短短几分钟内完成，则添加 exclude 或 include 勘误过滤器后可能需要 30 分钟。

示例 1

使用基本 Red Hat Enterprise Linux 软件包创建软件仓库。这个过滤器需要添加到内容视图中的 Red Hat Enterprise Linux 软件仓库。

Filter:

- 包含类型：Include
- 内容类型：软件包组
- filter：只选择基本软件包组

示例 2

创建一个软件仓库，在特定日期后排除所有勘误（但安全更新除外）。如果您要定期执行系统更新，但关键安全更新除外，这非常有用，必须立即应用。这个过滤器需要添加到内容视图中的 Red Hat Enterprise Linux 软件仓库。

Filter:

- 包含类型：Exclude
- 内容类型：Erratum（根据日期和类型）
- filter：只选择 Bugfix 和 Enhancement 勘误类型，并清除安全勘误类型。将 Date Type 设置为 Updated On。将 Start Date 设置为您要限制勘误的日期。将结束日期留空，以确保过滤任何新的非安全勘误。

示例 3

示例 1 和示例 2 的组合，其中您需要操作系统软件包，并希望排除最新的程序错误修复和增强勘误。这需要两个过滤器附加到同一内容视图。内容视图首先处理 Include 过滤器，然后处理 Exclude 过滤器。

过滤器 1：

- 包含类型：Include
- 内容类型：软件包组
- filter：只选择基本软件包组

过滤器 2：

- 包含类型：Exclude
- 内容类型：Erratum（根据日期和类型）
- filter：只选择 Bugfix 和 Enhancement 勘误类型，并清除安全勘误类型。将 Date Type 设置为 Updated On。将 Start Date 设置为您要限制勘误的日期。将结束日期留空，以确保过滤任何新的非安全勘误。

示例 4

在内容视图中过滤特定模块流。

过滤器 1：

- 包含类型：Include
- 内容类型：模块流
- filter：只选择您要用于内容视图的特定模块流，如 ant，然后点 Add Module Stream。

过滤器 2：

- 包含类型：Exclude
- 内容类型：Package
- filter：添加一个规则来过滤您要从内容视图中排除的任何非模块化软件包。如果没有过滤软件包，则内容视图过滤器包括与模块流 ant 关联的所有非模块化软件包。添加一条规则来排除所有 * 软件包，或指定一个您要排除的软件包名称。

有关内容过滤器的工作方式的另一个示例，请参见以下文章：[“如何在 Satellite 6 中工作内容过滤器”](#)。

7.13. 为 YUM 内容创建内容过滤器

您可以过滤包含 Yum 内容的内容视图，使其包含或排除特定的软件包、软件包组、勘误表或模块流。基于 name, version, 和 architecture 组合进行过滤。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

有关如何构建过滤器的示例，请参阅 [第 7.12 节“内容过滤器示例”](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。

2. 选择内容视图。
3. 在 Filters 选项卡中，点 Create filter。
4. 输入名称。
5. 从 Content type 列表中，选择一个内容类型。
6. 从 Inclusion Type 列表中，选择 Include filter 或 Exclude filter。
7. 可选：在 Description 字段中输入过滤器的描述。
8. 点 Create filter 创建您的内容过滤器。
9. 根据您为 Content Type 输入的内容，添加规则以创建您想要的过滤器。
10. 如果您希望过滤器应用存储库的子集或应用到所有存储库，请选择。
11. 单击 Publish New Version 以发布过滤的存储库。
12. 可选：在 Description 字段中输入更改的描述。
13. 单击 Create filter，以发布内容视图的新版本。
您可以在所有环境中提升此内容视图。

CLI 过程

1. 向内容视图添加过滤器。使用 `--inclusion false` 选项将过滤器设置为 Exclude 过滤器：

```
# hammer content-view filter create \
--name "Errata Filter" \
--type erratum --content-view "Example_Content_View" \
--description "My latest filter" \
--inclusion false \
--organization "My_Organization"
```

2. 在过滤器中添加规则：

```
# hammer content-view filter rule create \
--content-view "Example_Content_View" \
--content-view-filter "Errata Filter" \
--start-date "YYYY-MM-DD" \
--types enhancement,bugfix \
--date-type updated \
--organization "My_Organization"
```

3. 发布内容视图：

```
# hammer content-view publish \
--name "Example_Content_View" \
--description "Adding errata filter" \
--organization "My_Organization"
```

4. 在所有环境中提升视图：

```
# hammer content-view version promote \
--content-view "Example_Content_View" \
--version 1 \
--to-lifecycle-environment "Development" \
--organization "My_Organization"
# hammer content-view version promote \
--content-view "Example_Content_View" \
--version 1 \
--to-lifecycle-environment "Testing" \
--organization "My_Organization"
# hammer content-view version promote \
--content-view "Example_Content_View" \
--version 1 \
--to-lifecycle-environment "Production" \
--organization "My_Organization"
```

7.14. 删除多个内容视图版本

您可以同时删除多个内容视图版本。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 选择您要删除版本的内容视图。
3. 在 Versions 选项卡中，选择您要删除的版本或版本的复选框。
4. 点内容视图列表顶部的垂直 ellipsis 图标。
5. 点 Delete 以打开显示任何受影响环境的删除向导。
6. 如果没有受影响的环境，请查看详情并点 Delete。
7. 如果有受影响的环境，请在删除前重新分配任何主机或激活码。
8. 检查操作的详情。
9. 点击 Delete。

7.15. 清除搜索过滤器

如果您在 Search 文本框中使用关键字搜索特定内容类型，且搜索返回没有结果，请单击 Clear search 以清除所有搜索查询并重置搜索文本框。

如果您使用过滤器在 Type 文本框中搜索特定存储库，且搜索返回没有结果，请单击 Clear filters 以清除所有活动过滤器并重置 Type 文本框。

7.16. 标准化内容视图空状态

如果没有为内容视图列出过滤器，请单击 Create filter。此时会打开一个模态，显示创建过滤器的后续步骤。按照以下步骤添加新过滤器来创建新内容类型。

7.17. 比较内容视图版本

使用这个流程比较 Satellite 的内容视图版本功能。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 选择您要比较的版本的内容视图。
3. 在 Versions 选项卡中，选中您要比较的两个版本旁边的复选框。
4. 单击 Compare。

Compare 屏幕在版本下拉菜单中有预先选择的版本，以及所有版本中存在的所有内容类型选项卡。您可以过滤结果，只显示相同的、不同的或所有内容类型。您可以从下拉菜单中选择不同的内容视图版本来比较不同的内容视图版本。

7.18. 分发归档的内容视图版本

通过设置 Distribute 归档的内容视图版本，可以在 Satellite 内容 Web 应用和其他存储库中托管非提升的内容视图版本存储库。这在调试时很有用，以查看您的内容视图版本中存在什么内容。

流程

1. 在 Satellite Web UI 中，进入到 Administer > Settings。
2. 单击 Content 选项卡。
3. 将 Distribute 归档的内容视图版本 参数设置为 Yes。
4. 点 Submit。

这可让没有生命周期环境的内容视图版本存储库在 `satellite.example.com/pulp/My_Organization/content_views/My_Content_View/My_Content_View_Version/` 中分发。



注意

启用设置后，不会分发旧的非提升内容视图版本。只有新的内容视图版本才会被分发。

第 8 章 在 SATELLITE 服务器间同步内容

在带有多个 Satellite 服务器的 Satellite 设置中，您可以使用 Inter-Satellite Synchronization (ISS) 将内容从一个上游服务器同步到一个或多个下游服务器。

Satellite 有两种可能的 ISS 配置，具体取决于您部署基础架构的方式。根据您的用例场景为 ISS 配置您的 Satellite。如需更多信息，请参阅 [在断开连接的网络环境中安装 Satellite 服务器中的如何配置 Inter-Satellite Synchronization](#)。要更改 pulp 导出路径，请参阅 [知识库文章 Hammer 内容导出失败，并带有 "Path '/the/path' is not an allowed export path "](#)。

8.1. 如何使用导出和导入同步内容

使用导出和导入工作流同步内容的方法有多种：

- 您可以使用上游 Satellite 服务器作为内容存储，这意味着您可以同步整个库而不是内容视图版本。此方法提供了最简单的导出/导入工作流。在这种情况下，您可以管理下游版本。如需更多信息，请参阅 [第 8.1.1 节“使用上游 Satellite 服务器作为内容存储”](#)。
- 您可以使用上游 Satellite 服务器同步内容视图版本。这种方法可以更好地控制 Satellite 服务器之间同步的内容。如需更多信息，请参阅 [第 8.1.2 节“使用上游 Satellite 服务器同步内容视图版本”](#)。
- 您可以同步单个存储库。如果您使用 Content-View 同步方法，但您想要同步附加存储库而无需将其添加到现有内容视图中，这非常有用。如需更多信息，请参阅 [第 8.1.3 节“同步单个软件仓库”](#)。



注意

使用导出和导入同步内容需要在下游和上游 Satellite 服务器上具有相同的主版本、次版本和补丁版本。

当您无法匹配上游和下游 Satellite 版本时，您可以使用：

- 可同步的导出和导入。
- 通过上游 Satellite 与上游 Satellite 进行间同步(ISS)，以及连接到上游 Satellite 的下游 Satellite。

8.1.1. 使用上游 Satellite 服务器作为内容存储

在这种情况下，您将使用上游 Satellite 服务器作为内容存储进行更新，而不是管理内容。您可以使用下游卫星服务器管理隔离网络之后的所有基础架构的内容。您可以从上游 Satellite 服务器导出库内容，并将它导入到下游卫星服务器。

在上游 Satellite 服务器上

1. 确保软件仓库以以下方法之一使用 Immediate 下载策略：
 - a. 对于使用 On Demand 的现有存储库，请将存储库详情页面上的下载策略更改为 Immediate。
 - b. 对于新存储库，请确保在启用红帽软件仓库前将 Default Red Hat Repository 下载策略设置为 Immediate，并且 Default 下载策略被设置为自定义存储库的 Immediate。

更多信息请参阅 [第 4.9 节“下载策略概述”](#)。

2. 启用您要同步的内容。更多信息请参阅 [第 4.6 节“启用红帽软件仓库”](#)。如果要同步自定义内容，首先 [创建一个自定义产品并同步产品存储库](#)。
3. 同步启用的内容：
 - a. 在第一个导出上，执行完整的库导出，以便导出所有同步的内容。这会生成内容存档，稍后您可以导入到一个或多个下游 Satellite 服务器。有关执行完整的库导出的详情，请参考 [第 8.3 节“导出库环境”](#)。
 - b. 逐步导出上游 Satellite 服务器上的所有将来的更新。这会生成一个更精简的内容存档，该存档仅包含最新的更新集合。例如，如果您启用并同步新存储库，下一个导出的内容存档仅包含来自新启用的存储库的内容。有关执行增量库导出的详情，请参考 [第 8.6 节“递增导出库环境”](#)。

在下游 Satellite 服务器上

1. 将从上游 Satellite 服务器导出的内容传递给硬盘。
2. 将它放到 `/var/lib/pulp/imports` 下的目录中。
3. 按照 [第 8.15 节“导入到库环境中”](#) 中介绍的步骤将内容导入到机构。然后，您可以根据需要使用内容视图或生命周期环境来管理内容。

8.1.2. 使用上游 Satellite 服务器同步内容视图版本

在这种情况下，您不仅将上游 Satellite 服务器用作内容存储，还用于同步隔离网络后面所有基础架构的内容。您将来自 CDN 的更新策展到内容视图和生命周期环境中。将内容提升到指定的生命周期环境后，您可以从上游 Satellite 服务器导出内容并将其导入到下游卫星服务器。

在上游 Satellite 服务器上

1. 确保软件仓库以以下方法之一使用 Immediate 下载策略：
 - a. 对于使用 On Demand 的现有存储库，请将存储库详情页面上的下载策略更改为 Immediate。
 - b. 对于新存储库，请确保在启用红帽软件仓库前将 Default Red Hat Repository 下载策略设置为 Immediate，并且 Default 下载策略被设置为自定义存储库的 Immediate。

更多信息请参阅 [第 4.9 节“下载策略概述”](#)。

2. 启用您要同步的内容。更多信息请参阅 [第 4.6 节“启用红帽软件仓库”](#)。如果要同步自定义内容，首先 [创建一个自定义产品并同步产品存储库](#)。
3. 同步启用的内容：
 - a. 对于第一个导出，在您要导出的内容视图版本上执行完整的版本导出。如需更多信息，请参阅 [第 8.7 节“导出内容视图版本”](#)。这会生成内容存档，您可以导入到一个或多个下游 Satellite 服务器。
 - b. 逐步导出连接的 Satellite 服务器中的所有更新。这会生成更精简的内容存档，该存档仅包含最近一组更新的更改。例如，如果您的内容视图具有新存储库，则此导出的内容存档仅包含最新的更改。如需更多信息，请参阅 [第 8.9 节“以递增方式导出内容视图版本”](#)。
 - c. 有新内容时，请在导出递增前重新发布包含此内容的内容视图。如需更多信息，请参阅 [第 7 章管理内容视图](#)。这会创建一个新的内容视图版本，其中包含要导出的相应内容。

在下游 Satellite 服务器上

1. 将从上游 Satellite 服务器导出的内容传递给硬盘。
2. 将它放到 `/var/lib/pulp/imports` 下的目录中。
3. 将内容导入到您想要的组织。如需更多信息，请参阅 [第 8.17 节“导入内容视图版本”](#)。这将从导出的内容存档中创建内容视图版本，然后相应地导入内容。

8.1.3. 同步单个软件仓库

在这种情况下，您可以导出和导入单个存储库。

在上游 Satellite 服务器上

1. 确保软件仓库以以下方式之一使用 Immediate 下载策略：
 - a. 对于使用 On Demand 的现有存储库，请将存储库详情页面上的下载策略更改为 Immediate。
 - b. 对于新存储库，请确保在启用红帽软件仓库前将 Default Red Hat Repository 下载策略设置为 Immediate，并且 Default 下载策略被设置为自定义存储库的 Immediate。

更多信息请参阅 [第 4.9 节“下载策略概述”](#)。

2. 启用您要同步的内容。更多信息请参阅 [第 4.6 节“启用红帽软件仓库”](#)。如果要同步自定义内容，首先[创建一个自定义产品并同步产品存储库](#)。
3. 同步启用的内容：
 - a. 在第一个导出上，执行完整的存储库导出，以便导出所有同步的内容。这会生成内容存档，稍后您可以导入到一个或多个下游 Satellite 服务器。有关执行完整存储库导出的详情，请参考 [第 8.10 节“导出软件仓库”](#)。
 - b. 逐步导出上游 Satellite 服务器上的所有将来的更新。这会生成一个更精简的内容存档，该存档仅包含最新的更新集合。有关执行增量存储库导出的详情，请参考 [第 8.12 节“以递增方式导出存储库”](#)。

在下游 Satellite 服务器上

1. 将从上游 Satellite 服务器导出的内容传递给硬盘。
2. 将它放到 `/var/lib/pulp/imports` 下的目录中。
3. 将内容导入到机构。请参阅 [第 8.19 节“导入软件仓库”](#)。然后，您可以根据需要使用内容视图或生命周期环境来管理内容。

8.2. 同步自定义软件仓库

在使用 Inter-Satellite Synchronization Network Sync 时，会自动配置红帽存储库，但不会自动配置自定义存储库。使用这个流程，通过 Inter-Satellite Synchronization (ISS) Network Sync 将内容从一个位于已连接的 Satellite 服务器同步到一个断开连接的 Satellite 服务器。

在完成断开连接的 Satellite 服务器的步骤前，请遵循连接的 Satellite 服务器的步骤。

连接的 Satellite 服务器

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 点自定义产品。
3. 点自定义存储库。
4. 复制 Published At: URL。
5. 在断开连接的 Satellite 服务器上继续操作。

断开连接的 Satellite 服务器

1. 从连接的 Satellite 服务器下载 `katello-server-ca.crt` 文件：

```
# curl http://satellite.example.com/pub/katello-server-ca.crt
```

2. 创建一个 SSL 内容凭据，其内容为 `katello-server-ca.crt`。有关创建 SSL 内容凭证的详情，请参考第 4.3 节“导入自定义 SSL 证书”。
3. 在 Satellite Web UI 中，进入到 Content > Products。
4. 使用以下方法创建自定义产品：
 - Upstream URL : 粘贴您之前复制的链接。
 - SSL CA Cert : 选择从您连接的 Satellite 服务器传输的 SSL 证书。有关创建自定义产品的更多信息，请参阅第 4.4 节“创建自定义产品”。

完成这些步骤后，断开连接的 Satellite 服务器就正确配置了自定义存储库。

8.3. 导出库环境

您可以将组织的 Library 环境中所有 Yum 存储库的内容导出到 Satellite 服务器的存档文件，并使用此存档文件在另一个卫星服务器或其他 Satellite Server 组织中创建相同的存储库。导出的存档文件包含以下数据：

- 包含内容视图版本元数据的 JSON 文件。
- 包含机构库环境中所有存储库的存档文件。

Satellite 服务器仅导出库环境中包含的 RPM、Kickstart 文件和 Docker 内容。

先决条件

- 确保导出目录有可用存储空间来容纳导出。
- 确保 `/var/lib/pulp/exports` 目录有空闲的存储空间，相当于导出在导出过程中为临时文件导出的库大小。
- 确保将您导出的库生命周期环境中所有存储库的下载策略设置为 Immediate。更多信息请参阅第 4.9 节“下载策略概述”。
- 确保您导出的产品与所需日期同步。

流程

1. 使用组织名称或 ID 导出。

```
# hammer content-export complete library --organization="My_Organization"
```

2. 验证包含内容视图导出的版本的存档位于导出目录中：

```
# ls -lh /var/lib/pulp/exports/My_Organization/Export-Library/1.0/2021-03-02T03-35-24-00-00
total 68M
-rw-r--r--. 1 pulp pulp 68M Mar  2 03:35 export-1e25417c-6d09-49d4-b9a5-23df4db3d52a-
20210302_0335.tar.gz
-rw-r--r--. 1 pulp pulp 333 Mar  2 03:35 export-1e25417c-6d09-49d4-b9a5-23df4db3d52a-
20210302_0335-toc.json
-rw-r--r--. 1 pulp pulp 443 Mar  2 03:35 metadata.json
```

您需要所有三个文件、**tar.gz**、**toc.json** 以及 **metadata.json** 文件才能导入。

3. 组织中创建了一个新的内容视图 Export-Library。此内容视图包含属于此组织的所有存储库。此内容视图的新版本会自动发布并导出。

使用块导出

在很多情况下，导出的存档内容的大小可能为几 GB。如果要将其分成较小的大小或块。您可以在 `export` 命令中直接使用 `--chunk-size-gb` 标志进行处理。在以下示例中，您可以看到如何指定 `--chunk-size-gb=2` 来分割 2 GB 块中的存档。

```
# hammer content-export complete library \
--chunk-size-gb=2 \
--organization="My_Organization"

Generated /var/lib/pulp/exports/My_Organization/Export-Library/2.0/2021-03-02T04-01-25-00-
00/metadata.json

# ls -lh /var/lib/pulp/exports/My_Organization/Export-Library/2.0/2021-03-02T04-01-25-00-00/
```

8.4. 以可同步格式导出库环境

您可以将机构的 Library 环境中的所有 yum 存储库、Kickstart 存储库和文件存储库导出到可同步格式，您可以使用它来创建自定义 CDN 并通过 HTTP/HTTPS 同步自定义 CDN 的内容。

然后，您可以在本地 Web 服务器上提供生成的内容，并在导入 Satellite 服务器或另一个 Satellite Server 组织中同步它。

您可以使用生成的内容在另一个 Satellite 服务器或使用内容导入功能在另一个 Satellite Server 组织中创建相同的存储库。在导入导出的存档时，会在导入 Satellite 服务器上创建或更新常规内容视图。如需更多信息，请参阅 [第 8.17 节“导入内容视图版本”](#)。

您可以以同步格式从 Satellite 服务器导出以下内容：

- yum 软件仓库
- Kickstart 软件仓库
- 文件软件仓库

您无法导出 Ansible、Deb 或 Docker 内容。

导出包含包含软件包、列出文件以及 Yum 格式的存储库元数据的目录，这些元数据可用于在导入卫星服务器中同步。

先决条件

- 确保将您导出的库生命周期环境中所有存储库的下载策略设置为 Immediate。更多信息请参阅第 4.9 节“下载策略概述”。
- 确保您导出的产品与所需日期同步。
- 确保导出内容的用户具有 Content Exporter 角色。

流程

1. 使用机构名称或 ID 导出：

```
# hammer content-export complete library \
--organization="My_Organization" \
--format=syncable
```

2. 可选：验证导出内容是否在导出目录中：

```
# du -sh /var/lib/pulp/exports/My_Organization/Export-My_Repository/1.0/2021-03-02T03-35-24-00-00
```

8.5. 导入可同步导出

流程

- 使用机构名称或 ID 导入同步导出：

```
# hammer content-import library
--organization="My_Organization"
--path="My_Path_To_Syncable_Export"
```



注意

可同步导出必须位于您的 `ALLOWED_IMPORT_PATHS` 中，如 `/etc/pulp/settings.py` 中指定的。默认情况下，这包括 `/var/lib/pulp/imports`。

8.6. 递增导出库环境

在系统资源方面，导出库内容可能是一个非常昂贵的操作。具有多个 Red Hat Enterprise Linux 树的组织可以在 Satellite 服务器上占用几 GB 的空间。

在这种情况下，您可以创建一个增量导出，其中包含自上导出后更改的内容片段。增量导出通常会导致存档文件小于完整导出。

以下示例显示了组织库中所有存储库的增量导出。

流程

1. 创建增量导出：

```
# hammer content-export incremental library \
--organization="My_Organization"
```

如果要创建可同步导出，请添加 `--format=syncable`。默认情况下，Satellite 创建一个可导入的导出。

后续步骤

- 可选：查看导出的数据：

```
# find /var/lib/pulp/exports/My_Organization/Export-Library/
```

8.7. 导出内容视图版本

您可以将一个内容视图版本导出到 Satellite 服务器中的存档文件，并使用此存档文件在其他 Satellite 服务器或其他 Satellite Server 组织上创建相同的内容视图版本。Satellite 将复合内容视图导出为普通内容视图。复合性质不会被保留。在导入导出的存档时，会在下游 Satellite 服务器上创建或更新常规内容视图。导出的存档文件包含以下数据：

- 包含内容视图版本元数据的 JSON 文件
- 包含内容视图版本中包含的所有存储库的存档文件

您只能导出添加到内容视图版本的 Yum 存储库、Kickstart 文件和 Docker 内容。Satellite 不会导出以下内容：

- 内容视图定义和元数据，如软件包过滤器。

先决条件

要导出内容视图，请确保要导出的 Satellite 服务器满足以下条件：

- 确保导出目录有可用存储空间来容纳导出。
- 确保 `/var/lib/pulp/exports` 目录有空闲的存储空间，相当于导出在导出过程中为临时文件导出的库大小。
- 确保将您导出的内容视图中所有存储库的下载策略设置为 Immediate。如需更多信息，请参阅第 4.9 节“下载策略概述”。
- 确保您导出的产品与所需日期同步。
- 确保导出内容的用户具有 Content Exporter 角色。

导出内容视图版本

1. 列出可用于导出的内容视图版本：

```
# hammer content-view version list \
--content-view="My_Content_View" \
--organization="My_Organization"

---|-----|-----|-----|-----
```

```

ID | NAME | VERSION | DESCRIPTION | LIFECYCLE ENVIRONMENTS
---|-----|-----|-----|-----
5 | view 3.0 | 3.0 | | Library
4 | view 2.0 | 2.0 | |
3 | view 1.0 | 1.0 | |
---|-----|-----|-----|-----

```

导出内容视图版本

1. 获取所需版本的版本号。以下示例目标版本 1.0 用于导出。

```

# hammer content-export complete version \
--content-view="Content_View_Name" \
--version=1.0 \
--organization="My_Organization"

```

2. 验证包含内容视图导出的版本的存档位于导出目录中：

```

# ls -lh /var/lib/pulp/exports/My_Organization/Content_View_Name/1.0/2021-02-25T18-59-26-00-00/

```

您需要所有三个文件，如 `tar.gz` 归档文件、`toc.json` 和 `metadata.json` 来成功导入内容。

使用块导出

在很多情况下，导出的存档内容的大小可能会超过几 GB。您可能想要分割较小的大小或块。您可以在 `hammer content-export` 命令中使用 `--chunk-size-gb` 选项来处理此操作。以下示例使用 `--chunk-size-gb=2` 将存档分成 2 GB 块。

```

# hammer content-export complete version \
--chunk-size-gb=2 \
--content-view="Content_View_Name" \
--organization="My_Organization" \
--version=1.0
# ls -lh /var/lib/pulp/exports/My_Organization/view/1.0/2021-02-25T21-15-22-00-00/

```

8.8. 以可同步格式导出内容视图版本

您可以将一个内容视图版本导出到可用于创建自定义 CDN 的可同步格式。导出内容视图后，您可以执行以下操作之一：

- 通过 HTTP/HTTPS 同步自定义 CDN 的内容。
- 使用 `hammer content-import` 导入内容。请注意，这需要导出和导入服务器来运行 Satellite 6.15。

然后，您可以在导入 Satellite 服务器或另一个 Satellite Server 组织中使用本地 Web 服务器提供生成的内容。

您无法直接导入 Syncable Format 导出。相反，在导入 Satellite 服务器上，您必须：

- 将生成的内容复制到可以导入 Satellite 服务器的 HTTP/HTTPS Web 服务器。
- 将 CDN 配置更新为自定义 CDN。

- 将 CDN URL 设置为指向 Web 服务器。
- 可选：如果 web 服务器需要，请设置 SSL/TLS CA 凭证。
- 启用存储库。
- 同步存储库。

您可以以同步格式从 Satellite 服务器导出以下内容：

- yum 软件仓库
- Kickstart 软件仓库
- 文件软件仓库

您无法导出 Ansible、Deb 或 Docker 内容。

导出包含包含软件包、列出文件以及 Yum 格式的存储库元数据的目录，这些元数据可用于在导入卫星服务器中同步。

先决条件

- 确保将您导出的内容视图中所有存储库的下载策略设置为 Immediate。如需更多信息，请参阅 [第 4.9 节“下载策略概述”](#)。
- 确保您导出的产品与所需日期同步。
- 确保导出内容的用户具有 Content Exporter 角色。

导出内容视图版本

- 列出可用于导出的内容视图版本：

```
# hammer content-view version list \
--content-view="My_Content_View" \
--organization="My_Organization"
```

流程

1. 获取所需版本的版本号。以下示例为导出目标版本 1.0：

```
# hammer content-export complete version \
--content-view="Content_View_Name" \
--version=1.0 \
--organization="My_Organization" \
--format=syncable
```

2. 可选：验证导出内容是否在导出目录中：

```
# ls -lh /var/lib/pulp/exports/My_Organization/My_Content_View_Name/1.0/2021-02-25T18-59-26-00-00/
```

8.9. 以递增方式导出内容视图版本

在系统资源方面，导出完整的内容视图版本可能非常昂贵。内容视图有多个 Red Hat Enterprise Linux 树的版本可以占用 Satellite 服务器上的几 GB 空间。

在这种情况下，您可以创建一个增量导出，其中包含自上导出后更改的内容片段。增量导出通常会导致存档文件小于完整导出。

流程

1. 创建增量导出：

```
# hammer content-export incremental version \
--content-view="My_Content_View" \
--organization="My_Organization" \
--version="My_Content_View_Version"
```

如果要创建可同步导出，请添加 `--format=syncable`。默认情况下，Satellite 创建一个可导入的导出。

后续步骤

- 可选：查看导出的内容视图：

```
# find
/var/lib/pulp/exports/My_Organization/My_Exported_Content_View/My_Content_View_Ve
rsion/
```

- 您可以将导出的内容视图版本导入到 Satellite 服务器中。如需更多信息，请参阅 [第 8.17 节“导入内容视图版本”](#)。

8.10. 导出软件仓库

您可以从 Satellite 服务器导出组织的 Library 环境中的存储库内容。您可以使用此存档文件在另一个 Satellite 服务器或其他 Satellite 服务器机构中创建相同的存储库。

您可以从 Satellite 服务器导出以下内容：

- Ansible 存储库
- Kickstart 软件仓库
- yum 软件仓库
- 文件软件仓库
- Docker 内容

导出包含以下数据：

- 两个包含存储库元数据的 JSON 文件。
- 一个或多个存档文件，其中包含来自机构的库环境的存储库的内容。

您需要所有文件 `tar.gz`，`toc.json` 和 `metadata.json` 才能导入。

先决条件

- 确保导出目录有足够的可用空间来容纳导出。
- 确保 `/var/lib/pulp/exports` 目录有足够的可用空间，等同于您要导出的所有存储库的大小。
- 确保将您导出的库生命周期环境中的下载策略设置为 `Immediate`。更多信息请参阅 [第 4.9 节“下载策略概述”](#)。
- 确保您导出的产品与所需日期同步。

流程

1. 使用产品名称和存储库名称导出存储库：

```
# hammer content-export complete repository \
  --name="My_Repository" \
  --organization="My_Organization" \ --product="_My_Product"
```



注意

导出存档的大小取决于存储库中软件包的数量和大小。如果要导出存档分成块，请使用 `--chunk-size-gb` 参数导出存储库，以 GB 为单位限制其大小，如 `--chunk-size-gb=2`。

2. 可选：验证导出的存档是否位于导出目录中：

```
# ls -lh /var/lib/pulp/exports/My_Organization/Export-My_Repository/1.0/2022-09-02T03-35-24-00-00/
```

8.11. 以可同步格式导出存储库

您可以将组织的 Library 环境中的存储库内容导出到可用于创建自定义 CDN 的可同步格式，并通过 HTTP/HTTPS 从自定义 CDN 同步内容。

然后，您可以在导入 Satellite 服务器或另一个 Satellite Server 组织中使用本地 Web 服务器提供生成的内容。

您无法直接导入 Syncable Format 导出。相反，在导入 Satellite 服务器上，您必须：

- 将生成的内容复制到可以导入 Satellite 服务器的 HTTP/HTTPS Web 服务器。
- 将 CDN 配置更新为自定义 CDN。
- 将 CDN URL 设置为指向 Web 服务器。
- 可选：如果 web 服务器需要，请设置 SSL/TLS CA 凭证。
- 启用存储库。
- 同步存储库。

您可以以同步格式从 Satellite 服务器导出以下内容：

- yum 软件仓库
- Kickstart 软件仓库

- 文件软件仓库

您无法导出 Ansible、Deb 或 Docker 内容。

导出包含包含软件包、列出文件以及 Yum 格式的存储库元数据的目录，这些元数据可用于在导入卫星服务器中同步。

先决条件

- 确保将下载策略设置为您导出的库在库生命周期环境中的 Immediate。更多信息请参阅 [第 4.9 节“下载策略概述”](#)。

流程

1. 使用存储库名称或 ID 导出存储库：

```
# hammer content-export complete repository \
--organization="My_Organization" \
--product="My_Product" \
--name="My_Repository" \
--format=syncable
```

2. 可选：验证导出内容是否在导出目录中：

```
# du -sh /var/lib/pulp/exports/My_Organization/Export-My_Repository/1.0/2021-03-02T03-35-24-00-00
```

8.12. 以递增方式导出存储库

在系统资源方面，导出存储库可能是非常昂贵的操作。典型的 Red Hat Enterprise Linux 树可以在 Satellite 服务器上占用几 GB 的空间。

在这种情况下，您可以使用 Incremental Export 来仅导出从上一个导出后更改的内容片段。增量导出通常会导致存档文件小于完整导出。

以下示例显示了库生命周期环境中的存储库的增量导出。

流程

1. 创建增量导出：

```
# hammer content-export incremental repository \
--name="My_Repository" \
--organization="My_Organization" \
--product="My_Product"
```

2. 可选：查看导出的数据：

```
# ls -lh /var/lib/pulp/exports/My_Organization/Export-My_Repository/3.0/2021-03-02T03-35-24-00-00/
total 172K
-rw-r--r--. 1 pulp pulp 20M Mar 2 04:22 export-436882d8-de5a-48e9-a30a-17169318f908-20210302_0422.tar.gz
```

```
-rw-r--r--. 1 pulp pulp 333 Mar 2 04:22 export-436882d8-de5a-48e9-a30a-17169318f908-20210302_0422-toc.json
-rw-r--r--. 1 root root 492 Mar 2 04:22 metadata.json
```

8.13. 以可同步格式逐步导出存储库

在系统资源方面，导出存储库可能是非常昂贵的操作。典型的 Red Hat Enterprise Linux 树可以在 Satellite 服务器上占用几 GB 的空间。

在这种情况下，您可以使用 Incremental Export 来仅导出从上一个导出后更改的内容片段。增量导出通常会导致存档文件小于完整导出。

以下流程显示了库生命周期环境中的存储库的增量导出。

流程

1. 创建增量导出：

```
# hammer content-export incremental repository \
--format=syncable \
--name="My_Repository" \
--organization="My_Organization" \
--product="My_Product"
```

2. 可选：查看导出的数据：

```
# find /var/lib/pulp/exports/Default_Organization/My_Product/2.0/2023-03-09T10-55-48-05-00/ -name "*.rpm"
```

8.14. 记录您的导出

Satellite 保留所有导出的记录。每次在上游 Satellite 服务器上导出内容时，都会记录并维护导出以备将来查询。您可以使用记录来组织和管理导出，这在增量导出时特别有用。

当为多个下游 Satellite 服务器导出上游 Satellite 服务器的内容时，您还可以跟踪为特定服务器导出的内容。这有助于您跟踪导出和到哪里的内容。

在导出期间使用 `--destination-server` 参数来指示目标服务器。这个选项适用于所有 `content-export` 操作。

跟踪库导出的目的地

- 在导出库时指定目标服务器：

```
# hammer content-export complete library \
--destination-server=My_Downstream_Server_1 \
--organization="My_Organization" \
--version=1.0
```

跟踪内容视图导出的目的地

- 在导出内容视图版本时指定目标服务器：

```
# hammer content-export complete version \
--content-view="Content_View_Name" \
--destination-server=My_Downstream_Server_1 \
--organization="My_Organization" \
--version=1.0
```

查询导出记录

- 使用以下命令列出内容导出：

```
# hammer content-export list --organization="My_Organization"
```

8.15. 导入到库环境中

您可以将导出的库内容导入到另一个 Satellite 服务器上的机构的库生命周期环境中。有关从库环境中导出内容的更多信息，请参阅第 8.3 节“导出库环境”。

先决条件

- 导出的文件必须位于 `/var/lib/pulp/imports` 下的目录中。
- 如果导出内容中存在任何红帽存储库，导入机构的清单必须包含导出中包含的产品的订阅。
- 导入内容的用户必须具有 Content Importer 角色。

流程

1. 将导出的文件复制到您要导入的 Satellite 服务器上的 `/var/lib/pulp/imports` 子目录中。
2. 将导入目录及其内容的所有权设置为 `pulp:pulp`。

```
# chown -R pulp:pulp /var/lib/pulp/imports/2021-03-02T03-35-24-00-00
```

3. 验证是否已正确设置所有权：

```
# ls -lh /var/lib/pulp/imports/2021-03-02T03-35-24-00-00
total 68M
-rw-r--r--. 1 pulp pulp 68M Mar  2 04:29 export-1e25417c-6d09-49d4-b9a5-23df4db3d52a-20210302_0335.tar.gz
-rw-r--r--. 1 pulp pulp 333 Mar  2 04:29 export-1e25417c-6d09-49d4-b9a5-23df4db3d52a-20210302_0335-toc.json
-rw-r--r--. 1 pulp pulp 443 Mar  2 04:29 metadata.json
```

1. 确定您要导入到的组织。
2. 要将库内容导入到 Satellite 服务器，请输入以下命令：

```
# hammer content-import library \
--organization="My_Organization" \
--path=/var/lib/pulp/imports/2021-03-02T03-35-24-00-00
```

请注意，您必须输入完整路径 `/var/lib/pulp/imports/My_Exported_Library_Dir`。相对路径不起作用。

- 若要验证您是否导入了 Library 内容，请检查 Product 和 Repositories 的内容。目标组织中会创建一个名为 Import-Library 的新内容视图。此内容视图用于促进库内容导入。
默认情况下，Satellite Web UI 中不会显示此内容视图。import-Library 并不意味着直接分配到主机。相反，将主机分配到 Default Organization View 或另一个内容视图，就像您正常一样。

8.16. 从 WEB 服务器导入到库环境中

您可以将导出的库内容直接从 Web 服务器导入到另一个 Satellite 服务器上的机构的 Library 生命周期环境。有关从库环境中导出内容的更多信息，请参阅第 8.3 节“导出库环境”。

先决条件

- 导出的文件必须采用可同步格式。
- 导出的文件必须通过 HTTP/HTTPS 访问。
- 如果导出内容中存在任何红帽存储库，导入机构的清单必须包含导出中包含的产品的订阅。
- 导入内容视图版本的用户必须具有 Content Importer 角色。

流程

- 确定您要导入到的组织。
- 要将库内容导入到 Satellite 服务器，请输入以下命令：

```
# hammer content-import library \
--organization="My_Organization" \
--path=http://server.example.com/pub/exports/2021-02-25T21-15-22-00-00/
```

目标组织中会创建一个名为 Import-Library 的新内容视图。此内容视图用于促进库内容导入。

默认情况下，Satellite Web UI 中不会显示此内容视图。import-Library 并不意味着直接分配到主机。相反，请将您的主机分配到 Default Organization View 或其他内容视图。

8.17. 导入内容视图版本

您可以导入导出的内容视图版本，以便在另一个 Satellite 服务器上的机构中创建具有相同内容的版本。有关导出内容视图版本的更多信息，请参阅第 8.7 节“导出内容视图版本”。

当您导入内容视图版本时，它具有相同的主版本和次版本号，并包含与相同软件包和勘误表相同的存储库。如果导入机构中不存在自定义存储库、产品和内容视图，则会自动创建它们。

先决条件

- 导出的文件必须位于 /var/lib/pulp/imports 下的目录中。
- 如果导出内容中存在任何红帽存储库，导入机构的清单必须包含导出中包含的产品的订阅。
- 导入内容视图版本的用户必须具有 Content Importer 角色。

流程

- 将导出的文件复制到您要导入的 Satellite 服务器上的 /var/lib/pulp/imports 子目录中。

2. 将导入目录及其内容的所有权设置为 `pulp:pulp`。

```
# chown -R pulp:pulp /var/lib/pulp/imports/2021-02-25T21-15-22-00-00/
```

3. 验证是否已正确设置所有权：

```
# ls -lh /var/lib/pulp/imports/2021-02-25T21-15-22-00-00/
```

4. 要将内容视图版本导入到 Satellite 服务器，请输入以下命令：

```
# hammer content-import version \
  --organization=My_Organization \
  --path=/var/lib/pulp/imports/2021-02-25T21-15-22-00-00/
```

请注意，您必须输入完整路径 `/var/lib/pulp/imports/My_Exported_Version_Dir`。相对路径不起作用。

5. 要验证您是否成功导入了内容视图版本，请列出机构的内容视图版本：

```
# hammer content-view version list \
  --organization-id=My_Organization_ID
```

8.18. 从 WEB 服务器导入内容视图版本

您可以直接从 Web 服务器导入导出的内容视图版本，以便在另一个 Satellite 服务器上的机构中创建具有相同内容的版本。有关导出内容视图版本的更多信息，请参阅 [第 8.7 节“导出内容视图版本”](#)。

当您导入内容视图版本时，它具有相同的主版本和次版本号，并包含与相同软件包和勘误表相同的存储库。如果导入机构中不存在自定义存储库、产品和内容视图，则会自动创建它们。

先决条件

- 导出的文件必须采用可同步格式。
- 导出的文件必须通过 HTTP/HTTPS 访问。
- 如果导出内容中存在任何红帽存储库，导入机构的清单必须包含导出中包含的产品的订阅。
- 导入内容视图版本的用户必须具有 Content Importer 角色。

流程

- 将内容视图版本导入到 Satellite 服务器中：

```
# hammer content-import version \
  --organization=My_Organization \
  --path=http://server.example.com/pub/exports/2021-02-25T21-15-22-00-00/
```

8.19. 导入软件仓库

您可以将导出的存储库导入到另一个 Satellite 服务器上的机构中。有关导出软件仓库内容的更多信息，请参阅 [第 8.10 节“导出软件仓库”](#)。

先决条件

- 导出文件必须位于 `/var/lib/pulp/imports` 下的目录中。
- 如果导出包含任何红帽存储库，导入机构的清单必须包含导出中包含的产品的订阅。
- 导入内容的用户必须具有 Content Importer 角色。

流程

1. 将导出的文件复制到您要导入的 Satellite 服务器上的 `/var/lib/pulp/imports` 子目录中。
2. 将导入目录及其内容的所有权设置为 `pulp:pulp`。

```
# chown -R pulp:pulp /var/lib/pulp/imports/2021-03-02T03-35-24-00-00
```

3. 验证是否已正确设置所有权：

```
# ls -lh /var/lib/pulp/imports/2021-03-02T03-35-24-00-00
total 68M
-rw-r--r--. 1 pulp pulp 68M Mar  2 04:29 export-1e25417c-6d09-49d4-b9a5-23df4db3d52a-20210302_0335.tar.gz
-rw-r--r--. 1 pulp pulp 333 Mar  2 04:29 export-1e25417c-6d09-49d4-b9a5-23df4db3d52a-20210302_0335-toc.json
-rw-r--r--. 1 pulp pulp 443 Mar  2 04:29 metadata.json
```

1. 确定您要导入到的组织。
2. 要将存储库内容导入到 Satellite 服务器，请输入以下命令：

```
# hammer content-import repository \
  --organization="My_Organization" \
  --path=/var/lib/pulp/imports/2021-03-02T03-35-24-00-00
```

请注意，您必须输入完整路径 `/var/lib/pulp/imports/My_Exported_Repo_Dir`。相对路径不起作用。

3. 若要验证您是否导入了存储库，请检查 Product 和 Repository 的内容。

8.20. 从 WEB 服务器导入软件仓库

您可以将导出的存储库直接从 Web 服务器导入到另一个 Satellite 服务器上的组织。有关导出软件仓库内容的详情，请参考第 8.10 节“导出软件仓库”。

先决条件

- 导出的文件必须采用可同步格式。
- 导出的文件必须通过 HTTP/HTTPS 访问。
- 如果导出包含任何红帽存储库，导入机构的清单必须包含导出中包含的产品的订阅。
- 导入内容视图版本的用户必须具有 Content Importer 角色。

流程

1. 选择您要导入的组织。
2. 要将存储库导入到 Satellite 服务器，请输入以下命令：

```
# hammer content-import repository \
--organization="My_Organization" \
--path=http://server.example.com/pub/exports/2021-02-25T21-15-22-00-00/
```

8.21. 使用 HAMMER CLI CHEAT 导出和导入内容

表 8.1. Export

作用	命令
完全导出机构的库	hammer content-export complete library --organization="My_Organization"
以递增方式导出组织的库 (假设您之前已导出了一些内容)	hammer content-export incremental library --organization="My_Organization"
完全导出内容视图版本	hammer content-export complete version --content-view="My_Content_View" --version=1.0 --organization="My_Organization"
导出提升到 Dev 环境的内容视图版本	hammer content-export complete version --content-view="My_Content_View" --organization="My_Organization" --lifecycle-environment="Dev"
以较小的块(2-GB slab)导出内容视图	hammer content-export complete version --content-view="My_Content_View" --version=1.0 --organization="My_Organization" --chunk-size-gb=2
逐步导出内容视图版本 (假设您之前已导出了一些信息)	hammer content-export incremental version --content-view="My_Content_View" --version=2.0 --organization="My_Organization"
完全导出存储库	hammer content-export complete repository --product="My_Product" --name="My_Repository" --organization="My_Organization"
逐步导出存储库 (假设您之前已导出了一些内容)	hammer content-export incremental repository --product="My_Product" --name="My_Repository" --organization="My_Organization"
列出导出	hammer content-export list --content-view="My_Content_View" --organization="My_Organization"

表 8.2. Import

作用	命令
导入到机构的库中	hammer content-import library --organization="My_Organization" --path="/var/lib/pulp/imports/My_Exported_Library_Dir"
导入到内容视图版本	hammer content-import version --organization="My_Organization" --path="/var/lib/pulp/imports/My_Exported_Version_Dir"
导入存储库	hammer content-import repository --organization="My_Organization" --path="/var/lib/pulp/imports/My_Exported_Repo_Dir"

第 9 章 管理激活码

激活码提供了一种自动注册系统并订阅附加的方法。您可以创建多个密钥，并将它们与不同的环境和内容视图相关联。例如，您可以使用 Red Hat Enterprise Linux 工作站的订阅创建基本激活码，并将其与特定环境中的内容视图相关联。



重要

如果您在 Satellite 上启用了简单内容访问(SCA)，则无法将订阅附加到您的激活码中。启用 SCA 后，您不需要将订阅附加到您的主机。请注意，新创建的机构默认启用 SCA。要了解有关 SCA 的更多信息，请参阅 [简单内容访问](#)。

您可以在内容主机注册过程中使用激活码，以提高进程的速度、简单和一致性。请注意，只有在主机注册时，才会使用激活码。如果对激活码进行了更改，它只适用于在将来使用修改的激活码注册的主机。不会对现有主机进行更改。

激活码可以为内容主机定义以下属性：

- 关联的订阅和订阅附加行为
- 可用的产品和软件仓库
- 生命周期环境和内容视图
- 主机集合成员资格
- 系统目的

主机创建和注册之间的内容视图冲突

当您置备主机时，Satellite 将使用从主机组或主机设置中设置的内容视图中的置备模板和其他内容。主机注册后，来自激活密钥的内容视图将覆盖主机组或主机设置的原始内容视图。然后，Satellite 将激活密钥中的内容视图用于将来的每个任务，例如重建主机。

在重建主机时，请确保设置要在激活码中使用的内容视图，而不是在主机组或主机设置中。

将相同的激活码与多个内容主机一起使用

如果包含足够的订阅，您可以将相同的激活码应用到多个内容主机。但是，激活密钥仅设置内容主机的初始配置。当内容主机注册到组织时，可以手动将组织的内容附加到内容主机。

使用带有内容主机的多个激活码

内容主机可以和多个激活密钥关联，它们结合在一起以定义主机设置。如果设置冲突，则最后指定的激活码具有优先权。您可以通过设置主机组参数来指定优先级顺序，如下所示：

```
$ hammer hostgroup set-parameter \
--hostgroup "My_Host_Group" \
--name "My_Activation_Key" \
--value "name_of_first_key", "name_of_second_key", ...
```

9.1. 激活码的最佳实践

- 为每个用例创建一个激活码。这种结构、模块化并简化主机上的内容管理。

- 使用激活密钥的命名规则来指示内容和生命周期环境，例如 `red-hat-enterprise-linux-webserver`。
- 使用 Hammer 脚本或 [Ansible playbook](#) 自动执行激活密钥管理。

9.2. 创建激活码

您可以使用激活码来定义在注册期间要附加到主机的一组特定的订阅。您添加到激活码的订阅必须在关联的内容视图中可用。



重要

如果您在 Satellite 上启用了简单内容访问(SCA)，则无法将订阅附加到您的激活码中。启用 SCA 后，您不需要将订阅附加到您的主机。请注意，新创建的机构默认启用 SCA。要了解有关 SCA 的更多信息，请参阅 [简单内容访问](#)。

订阅管理器根据以下因素以不同的方式附加订阅：

- 是否存在与激活码关联的订阅？
- `auto-attach` 选项是否已启用？
- 对于 Red Hat Enterprise Linux 8 主机：激活码中是否设定了系统用途？

请注意，Satellite 只会为主机上安装的产品自动附加订阅。对于默认情况下不列出 Red Hat Enterprise Linux 上安装的产品订阅，如延长更新支持(EUS)订阅，请使用指定所需订阅的激活码，并禁用自动附加。

根据前面的因素，可以使用激活码订阅 3 个场景：

1. 自动附加订阅的激活码。
在没有指定和自动附加订阅的情况下，使用激活码搜索主机，搜索与激活密钥关联的内容视图提供的最适合订阅。这与输入 `subscription-manager --auto-attach` 命令类似。对于 Red Hat Enterprise Linux 8 主机，您可以在注册过程中将激活码配置为在主机上设置系统目的，以增强自动订阅附加。
2. 为自动附加一组自定义订阅的激活码。
如果指定了订阅，且启用了自动附加，使用激活码的主机从激活码中指定的列表中选择最适合订阅。在激活码中设置系统目的不会影响这种情况。
3. 带有准确订阅集的激活码。
如果指定了订阅并禁用自动附加，使用激活码的主机会与激活码中指定的所有订阅相关联。在激活码中设置系统目的不会影响这种情况。

自定义产品

如果自定义产品（通常包含不是由红帽提供的内容）被分配给激活码，无论自动附加设置是什么，这个产品总是为注册的内容主机启用。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Activation Keys 并点 Create Activation Key。

2. 在 **Name** 字段中输入激活码的名称。
3. 如果要设置限制，清除 **Unlimited hosts** 复选框，然后在 **Limit** 字段中输入您可以使用激活码注册的最大系统数量。如果您希望没有限制的主机使用激活码进行注册，确保选中了 **Unlimited Hosts** 复选框。
4. 可选：在 **Description** 字段中输入激活码的描述。
5. 从 **Environment** 列表中，选择要使用的环境。
6. 从 **内容视图** 列表中，选择要使用的内容视图。
7. 如果启用了简单内容访问(SCA)：
 - a. 在 **Repository Sets** 选项卡中，确保只启用您的命名存储库。
8. 如果没有启用 SCA：
 - a. 点 **Subscriptions** 选项卡，然后点 **Add** 子菜单。
 - b. 点之前创建的订阅旁边的复选框。
 - c. 单击 **Add Selected**。
9. 点击 **Save**。
10. 可选：对于 Red Hat Enterprise Linux 8 主机，在 **System Purpose** 部分中，您可以使用系统目的配置激活码，以便在注册过程中在主机上设置来增强订阅自动附加。

CLI 过程

1. 创建激活码：

```
# hammer activation-key create \
--name "My_Activation_Key" \
--unlimited-hosts \
--description "Example Stack in the Development Environment" \
--lifecycle-environment "Development" \
--content-view "Stack" \
--organization "My_Organization"
```

2. 可选：对于 Red Hat Enterprise Linux 8 主机，输入以下命令配置激活码，以便在注册过程中在主机上设置系统目的，以增强订阅自动附加。

```
# hammer activation-key update \
--organization "My_Organization" \
--name "My_Activation_Key" \
--service-level "Standard" \
--purpose-usage "Development/Test" \
--purpose-role "Red Hat Enterprise Linux Server" \
--purpose-addons "addons"
```

3. 获取订阅 ID 列表：

```
# hammer subscription list --organization "My_Organization"
```

4. 将 Red Hat Enterprise Linux 订阅 UUID 附加到激活码中：

```
# hammer activation-key add-subscription \
--name "My_Activation_Key" \
--subscription-id My_Subscription_ID \
--organization "My_Organization"
```

5. 列出与激活码关联的产品内容：

a. 如果启用了简单内容访问(SCA)：

```
# hammer activation-key product-content \
--content-access-mode-all true \
--name "My_Activation_Key" \
--organization "My_Organization"
```

b. 如果没有启用 SCA：

```
# hammer activation-key product-content \
--name "My_Activation_Key" \
--organization "My_Organization"
```

6. 覆盖 Satellite Client 6 存储库的默认自动启用状态。默认状态设置为 disabled。要启用，请输入以下命令：

```
# hammer activation-key content-override \
--name "My_Activation_Key" \
--content-label rhel-7-server-satellite-client-6-rpms \
--value 1 \
--organization "My_Organization"
```

9.3. 更新与激活码关联的订阅



重要

只有在 Satellite 上禁用了 Simple Content Access (SCA) 时，这个过程才有效。启用 SCA 后，您不需要将订阅附加到您的主机。请注意，新创建的机构默认启用 SCA。要了解有关 SCA 的更多信息，请参阅 [简单内容访问](#)。

使用这个步骤更改与激活码关联的订阅。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

请注意，对激活码的更改仅适用于更改后置备的机器。要更新现有内容主机上的订阅，请参阅 [第 2.7 节“更新多个主机上的红帽订阅”](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Activation Keys 并点激活码的名称。
2. 点击 Subscriptions 选项卡。
3. 要删除订阅，请选择 List/Remove，然后选择要删除的订阅左侧的复选框，然后单击 Remove Selected。

4. 要添加订阅，请选择添加，然后选中要添加的订阅左侧的复选框，然后单击添加所选项。
5. 点 Repository Sets 选项卡，并检查存储库的状态设置。
6. 要启用或禁用存储库，请选中存储库的复选框，然后使用 Select Action 列表更改状态。
7. 单击 Details 选项卡，选择此激活密钥的内容视图，然后单击保存。

CLI 过程

1. 列出激活码当前包含的订阅：

```
# hammer activation-key subscriptions \
--name My_Activation_Key \
--organization "My_Organization"
```

2. 从激活码中删除所需的订阅：

```
# hammer activation-key remove-subscription \
--name "My_Activation_Key" \
--subscription-id ff808181533518d50152354246e901aa \
--organization "My_Organization"
```

对于 `--subscription-id` 选项，您可以使用 UUID 或订阅的 ID。

3. 将新订阅附加到激活码：

```
# hammer activation-key add-subscription \
--name "My_Activation_Key" \
--subscription-id ff808181533518d50152354246e901aa \
--organization "My_Organization"
```

对于 `--subscription-id` 选项，您可以使用 UUID 或订阅的 ID。

4. 列出与激活码关联的产品内容：

```
# hammer activation-key product-content \
--name "My_Activation_Key" \
--organization "My_Organization"
```

5. 覆盖所需存储库的默认自动启用状态：

```
# hammer activation-key content-override \
--name "My_Activation_Key" \
--content-label content_label \
--value 1 \
--organization "My_Organization"
```

对于 `--value` 选项，输入 1 代表启用，输入 0 代表禁用。

9.4. 使用激活码进行主机注册

您可以使用激活码完成以下任务：

- 在通过 Red Hat Satellite 置备过程中注册新主机。Red Hat Satellite 中的 kickstart 自定义模板包含使用创建主机时定义的激活码注册主机的命令。
- 注册现有的 Red Hat Enterprise Linux 主机。将 Subscription Manager 配置为使用 Satellite 服务器注册，并在运行 `subscription-manager register` 命令时指定激活码。

您可以使用 Satellite Web UI、Hammer CLI 或 Satellite API 中的主机注册功能在 Satellite 中注册主机。如需更多信息，请参阅管理 [主机](#) 中的 [注册主机](#)。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > Register Host。
2. 从 Activation Keys 列表中，选择要分配给您的主机的激活密钥。
3. 点 Generate 来创建注册命令。
4. 点 files 图标将命令复制到您的剪贴板。
5. 使用 SSH 连接到您的主机，并运行注册命令。
6. 检查 `/etc/yum.repos.d/redhat.repo` 文件，并确保启用了适当的存储库。

CLI 过程

1. 使用 Hammer CLI 生成主机注册命令：

```
# hammer host-registration generate-command \
--activation-keys "My_Activation_Key"
```

如果您的主机不信任 Satellite 服务器的 SSL 证书，您可以通过在注册命令中添加 `--insecure` 标志来禁用 SSL 验证。

```
# hammer host-registration generate-command \
--activation-keys "My_Activation_Key" \
--insecure true
```

2. 使用 SSH 连接到您的主机，并运行注册命令。
3. 检查 `/etc/yum.repos.d/redhat.repo` 文件，并确保启用了适当的存储库。

API 过程

1. 使用 Satellite API 生成主机注册命令：

```
# curl -X POST https://satellite.example.com/api/registration_commands \
--user "My_User_Name" \
-H 'Content-Type: application/json' \
-d '{"registration_command": {"activation_keys": ["My_Activation_Key_1, My_Activation_Key_2"]}}'
```

如果您的主机不信任 Satellite 服务器的 SSL 证书，您可以通过在注册命令中添加 `--insecure` 标志来禁用 SSL 验证。

```
# curl -X POST https://satellite.example.com/api/registration_commands \
```

```
--user "My_User_Name" \  
-H 'Content-Type: application/json' \  
-d '{"registration_command": {"activation_keys": ["My_Activation_Key_1,  
My_Activation_Key_2"], "insecure": true }}'
```

使用激活码来简化指定环境。如需更多信息，[请参阅管理内容中的管理激活码](#)。

要将密码作为命令行参数输入密码，请使用 `username:password` 语法。请记住，这会在 shell 历史记录中保存密码。或者，您可以使用临时个人访问令牌而不是密码。要在 Satellite Web UI 中生成令牌，请导航到 `My Account > Personal Access Tokens`。

2. 使用 SSH 连接到您的主机，并运行注册命令。
3. 检查 `/etc/yum.repos.d/redhat.repo` 文件，并确保启用了适当的存储库。

多个激活码

在注册内容主机时，您可以使用多个激活码。然后，您可以为特定订阅集创建激活码，并根据内容主机要求合并它们。例如，以下命令使用 VDC 和 OpenShift 订阅将内容主机注册到您的机构：

```
# subscription-manager register \  
--activationkey="ak-VDC,ak-OpenShift" \  
--org="My_Organization"
```

设置冲突

如果激活码中存在冲突的设置，则最右侧的键具有优先权。

- 有冲突的设置：Service Level, Release Version, Environment, Content View, 和 Product Content。
- 没有冲突和主机的设置会取消它们：订阅和主机集合。
- 影响密钥本身而不是主机配置行为的设置：Content Host Limit 和 Auto-Attach。

9.5. 启用自动附加

当在激活码上启用自动附加并且有与密钥关联的订阅时，订阅管理服务会根据当前安装的产品、架构和首选选项等一组条件选择和附加最适合的关联订阅。



重要

只有在 Satellite 上禁用了 Simple Content Access (SCA) 时，这个过程才有效。启用 SCA 后，您不需要将订阅附加到您的主机。请注意，新创建的机构默认启用 SCA。要了解有关 SCA 的更多信息，[请参阅简单内容访问](#)。

您可以启用自动附加，且没有与密钥关联的订阅。当您不希望虚拟机消耗物理订阅，但要从 hypervisor 中继承基于主机的订阅时，这种密钥通常用于注册虚拟机。如需更多信息，[请参阅配置虚拟机订阅](#)。

`auto-attach` 默认启用。如果要强制附加与激活码关联的所有订阅，请禁用该选项。

流程

1. 在 Satellite Web UI 中，进入到 `Content > Lifecycle > Activation Keys`。

2. 点您要编辑的激活码名称。
3. 点击 Subscriptions 选项卡。
4. 单击 Auto-Attach 旁边的编辑图标。
5. 选择或清除复选框以启用或禁用自动附加。
6. 点击 Save。

CLI 过程

- 输入以下命令在激活码中启用自动附加：

```
# hammer activation-key update --name "My_Activation_Key" \
--organization "My_Organization" --auto-attach true
```

9.6. 设置服务级别

您可以配置激活码来为使用激活码创建的新主机定义默认服务级别。设置默认服务级别只选择要附加到主机的匹配订阅。例如，如果激活码上的默认服务级别被设置为 Premium，则注册后只有具有 Premium 服务级别的订阅才会附加到主机。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Activation Keys。
2. 点您要编辑的激活码名称。
3. 点 Service Level 旁边的编辑图标。
4. 从列表中选择所需的服务级别。该列表仅包含可供激活密钥使用的服务级别。
5. 点击 Save。

CLI 过程

- 在您的激活码中将服务级别设置为 Premium：

```
# hammer activation-key update \
--name "My_Activation_Key" \
--organization "My_Organization" \
--service-level premium
```

9.7. 在激活码中启用和禁用软件仓库

作为简单内容访问(SCA)用户，您可以在 Satellite Web UI 中的激活码中启用或禁用存储库。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Activation Keys。
2. 选择激活码。

3. 选择 *Repository Sets* 选项卡。
4. 从下拉菜单中，您可以过滤 *Repository type* 栏到 *Custom* 或 *Red Hat*。
5. 选择所需的存储库，或者点击 *Select All* 复选框来选择所有存储库。
6. 从 *Select Action* 列表中，选择 *Override to Enabled*, *Override to Disabled*, 或 *Reset to Default*。

第 10 章 管理勘误

作为红帽质量控制和发布过程的一部分，我们为客户提供了每个官方红帽 RPM 版本的更新。红帽将相关软件包组编译为一个勘误 (erratum) 以及一个提供更新描述的公告。公告类型有三种 (按重要顺序)：

安全公告

描述软件包中发现的固定安全问题。问题的安全影响可以是 Low、Moderate、Important 或 Critical。

错误修正公告

描述软件包的 bug 修复。

产品功能增强公告

描述添加到软件包中的增强功能和新功能。

当将存储库与红帽的内容交付网络(CDN)同步时，Red Hat Satellite 会导入这个勘误信息。Red Hat Satellite 还提供用于检查和过滤勘误表的工具，允许精确更新管理。这样，您可以选择相关的更新，并通过内容视图将其传播到所选内容主机。

勘误根据它们所包含的最重要的公告类型进行标记。因此，标记为产品功能增强公告的勘误只能包含功能增强更新，而程序错误修复公告可能会包含所有三种类型。

在 Red Hat Satellite 中，有两个关键字来描述勘误与可用内容主机的关系：

适用

适用于一个或多个内容主机的勘误，这意味着它会更新内容主机上存在的软件包。虽然这些勘误会应用到内容主机，在其状态变为 Installable 之前，勘误还不能安装。可安装的勘误会自动适用。

可安装

适用于一个或多个内容主机并可在内容主机上安装的勘误。可以从生命周期环境和关联的内容视图 (但尚未安装) 的内容主机提供可安装勘误表。

本章介绍了如何管理勘误表并将其应用到单个主机或多个主机。

10.1. 勘误的最佳实践

- 使用勘误表将安全问题的补丁添加到一组内容，而无需不必要的更新其他不受影响的软件包。
- 使用 Hammer 脚本或 Ansible playbook 自动执行勘误表管理。
- 查看内容主机页面上的勘误表，并将当前内容视图和生命周期环境的勘误表与 Library 生命周期环境 (包含最新同步的软件包) 进行比较。
您只能应用主机生命周期的内容视图版本中包含的勘误。您可以查看适用的勘误，作为创建增量内容视图为主机提供勘误的建议。如需更多信息，请参阅 [第 10.8 节“将勘误添加到增量内容视图”](#)。

10.2. 检查可用的勘误

以下流程描述了如何查看和过滤可用的勘误以及如何显示所选公告的元数据。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Content Types > Errata 来查看可用勘误列表。

2. 使用页面顶部的过滤工具来限制显示的勘误数量：

- 从列表中选择要检查的存储库。所有存储库会被默认选择。
- 默认选择适用复选框以仅查看所选存储库中适用的勘误。选择Installable复选框可以只查看被标记为installable的勘误。
- 要搜索勘误表，在Search字段中输入查询，格式为：

```
parameter operator value
```

有关搜索的参数列表，请参阅第10.3节“可用于勘误搜索的参数”。在支持的Operator中找到管理Red Hat Satellite中的Granular Search的适用Operator列表。自动建议与您输入一样。您还可以将查询与and和or运算符的使用相结合。例如，只显示与内核软件包相关的安全公告，请输入：

```
type = security and package_name = kernel
```

按Enter键开始搜索。

3. 点击您要检查的勘误的Errata ID：

- Details选项卡包含更新的软件包的描述，以及更新提供的重要修复和增强文档。
- 在Content Hosts选项卡中，您可以将勘误应用到所选内容主机，如第10.10节“将勘误应用到多个主机”所述。
- Repositories选项卡列出了已包含勘误的存储库。您可以根据环境和内容视图过滤存储库，并根据存储库名称搜索它们。

您还可以使用新的Host页面来查看检查可用的勘误表并选择要安装的勘误。

1. 在Satellite Web UI中，进入到Hosts > All Hosts并选择您需要的主机。
2. 如果存在与主机关联的勘误，则新Host页面中的Installable Errata卡会显示一个互动的pie chart，其中显示了安全公告、程序错误修正和增强的分类。
3. 在新主机页面中，选择Content选项卡。
4. 在Content页面上，选择Errata选项卡。
5. 页面中显示所选主机的可安装勘误表。
6. 点您要安装的任何勘误的复选框。
7. 选择Apply via Remote Execution来使用远程执行；如果希望自定义远程执行，选择Apply via customized remote execution。
8. 点Submit。

CLI 过程

- 要查看所有机构可用的勘误，请输入以下命令：

```
# hammer erratum list
```

- 要查看特定勘误的详情，请输入以下命令：

```
# hammer erratum info --id erratum_ID
```

- 您可以使用 `--search` 选项输入查询来搜索勘误。例如，要查看所选产品的适用勘误，其中包含排序的指定错误，以便在顶部显示安全勘误，请输入以下命令：

```
# hammer erratum list \
--product-id 7 \
--search "bug = 1213000 or bug = 1207972" \
--errata-restrict-applicable 1 \
--order "type desc"
```

10.3. 可用于勘误搜索的参数

参数	描述	示例
bug	按照 Bugzilla 号搜索。	<code>bug = 1172165</code>
cve	按照 CVE 号搜索。	<code>cve = CVE-2015-0235</code>
id	按照勘误表 ID 搜索。auto-suggest 系统显示可用 ID 列表，作为您的类型。	<code>id = RHBA-2014:2004</code>
发布日期	按照问题日期搜索。您可以指定确切的日期，如 "Feb16,2015"，或使用关键字，如 "Yesterday" 或 "1 hour ago"。可以使用 "<" 和 ">" 运算符来指定时间范围。	<code>发布日期"Jan 12,2015"</code>
package	按照完整的软件包构建名称搜索。auto-suggest 系统在您键入时显示可用软件包列表。	<code>package = glib2-2.22.5-6.el6.i686</code>
package_name	按照软件包名称搜索。auto-suggest 系统在您键入时显示可用软件包列表。	<code>package_name = glib2</code>
严重性	根据安全更新修复的问题严重性搜索。指定 <i>Critical</i> 、 <i>Important</i> 或 <i>Moderate</i> 。	<code>severity = Critical</code>
title	按照公告标题搜索。	<code>title ~ openssl</code>
type	按照公告类型搜索。指定 <i>security</i> 、 <i>bugfix</i> ，或 <i>enhancement</i> 。	<code>type = bugfix</code>

参数	描述	示例
已更新	按照最后一次更新的日期搜索。您可以使用与 issued 参数相同的格式。	<code>updated = "6 days ago"</code>

10.4. 应用可安装的勘误

使用以下步骤查看可安装的勘误列表，并选择要安装的勘误。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > All Hosts 并选择您需要的主机。
2. 如果存在与主机关联的勘误，则在新的 Host 页面的 Installable Errata 卡中会显示它们。
3. 在 Content 选项卡上，勘误显示所选主机的可安装勘误表。
4. 点您要安装的任何勘误的复选框。
5. 使用您要添加到主机的勘误旁边的垂直 ellipsis 图标，选择 Apply via Remote Execution 来使用远程执行。如果要自定义远程执行，请选择 Apply via custom remote execution。
6. 点 Submit。

10.5. 订阅勘误通知

您可以为 Satellite 用户配置电子邮件通知。用户收到适用和可安装的勘误表摘要、内容视图提升或同步存储库后的通知。如需更多信息，请参阅管理 [Red Hat Satellite](#) 中的配置电子邮件通知首选项。

10.6. 存储库依赖项解析的限制

借助 Satellite，对您的内容视图使用增量更新可以解决一些存储库依赖项问题。但是，在存储库级别进行依赖关系解析仍然会在占用时出现问题。

当存储库更新以新依赖项可用时，Satellite 会检索软件包的最新版本来解决依赖项，即使现有存储库软件包中存在旧版本。这可在安装软件包时创建其他依赖项解析问题。

示例情境

客户端上的存储库带有依赖项 `example_repository-1.0` 和依赖项 `example_repository-libs-1.0`。存储库也具有另一个软件包 `example_tools-1.0`。

软件包 `example_tools-1.1` 会提供安全勘误。`example_tools-1.1` 软件包需要 `example_repository-libs-1.1` 软件包作为依赖项。

在增量内容视图更新后，`example_tools-1.1`、`example_tools-1.0` 和 `example_repository-libs-1.1` 现在位于存储库中。存储库也包含软件包 `example_repository-1.0` 和 `example_repository-libs-1.0`。请注意，内容视图的增量更新没有添加软件包 `example_repository-1.1`。因为您可以使用 `dnf` 安装所有这些软件包，所以不会检测到潜在的问题。但是，当客户端安装 `example_tools-1.1` 软件包时，会出现依赖项解析问题，因为无法安装 `example_repository-libs-1.0` 和 `example_repository-libs-1.1`。

这个问题目前还没有临时解决方案。基本软件包和所应用的勘误集合之间的次要 Y 版本越大，依赖项解决的可能性越大。

10.7. 为勘误创建内容视图过滤器

您可以使用内容过滤器来限制勘误表。此类过滤器包括：

- **id** - 选择特定的勘误以允许生成的软件仓库。
- **date Range** - 定义日期范围，并包括该日期范围内发布的一系列勘误。
- **Type** - 选择要包括程序错误修复、功能增强和安全更新的勘误类型。

创建一个内容过滤器，在特定日期后排除勘误。这样可确保您的应用程序生命周期中的生产系统保持最新状态。然后，您可以修改过滤器的开始日期，在测试环境中引入新勘误，以测试新软件包在应用程序生命周期中的兼容性。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

先决条件

- 将创建包含所需勘误表的存储库的内容视图。如需更多信息，请参阅 [第 7.4 节“创建内容视图”](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Content Views。
2. 选择您要用于应用勘误表的内容视图。
3. 选择 Yum Content > Filters 并点 New Filter。
4. 在 Name 字段中输入 **Errata Filter**。
5. 从 Content Type 列表中，选择 **Erratum - Date** 和 **Type**。
6. 从 Inclusion Type 列表中，选择 **Exclude**。
7. 在 Description 字段中，输入 YYYY-MM-DD 中的 **Exclude errata** 项。
8. 点击 **Save**。
9. 对于勘误类型，请选择您要排除的勘误表类型的复选框。例如，选择 **Enhancement** 和 **Bugfix** 复选框并清除 **Security** 复选框来排除特定日期之后的增强和程序错误修复勘误，但包括所有安全勘误。
10. 对于 Date Type，请选择两个复选框之一：
 - **Issued On** 是勘误的发布日期。
 - **Updated On** 是勘误最新更新的日期。
11. 选择 **Start Date** 以排除所选日期或之后的所有勘误。
12. 将 **End Date** 字段留空。
13. 点击 **Save**。
14. 单击 **Publish New Version** 以发布生成的存储库。
15. 在 Description 字段中输入 **Adding errata filter**。

16. 点击 Save。
内容视图完成发布后，请注意，Content 列从初始存储库中报告较少的软件包和勘误表。这意味着过滤器可以成功排除上一年内的所有非安全勘误。
17. 单击 Versions 选项卡。
18. 单击 published version 右侧的 Promote。
19. 选择您要提升内容视图版本的环境。
20. 在 Description 字段中，输入提升的描述。
21. 单击 Promote Version，以在所需环境中提升此内容视图版本。

CLI 过程

1. 为勘误创建过滤器：

```
# hammer content-view filter create \
--content-view "My_Content_View" \
--description "Exclude errata items from the YYYY-MM-DD" \
--name "My_Filter_Name" \
--organization "My_Organization" \
--type "erratum"
```

2. 创建一个过滤规则，以排除您要设置的开始日期 或之后的所有勘误：

```
# hammer content-view filter rule create \
--content-view "My_Content_View" \
--content-view-filter="My_Content_View_Filter" \
--organization "My_Organization" \
--start-date "YYYY-MM-DD" \
--types=security,enhancement,bugfix
```

3. 发布内容视图：

```
# hammer content-view publish \
--name "My_Content_View" \
--organization "My_Organization"
```

4. 将内容视图提升到生命周期环境，以便包含的勘误可供该生命周期环境使用：

```
# hammer content-view version promote \
--content-view "My_Content_View" \
--organization "My_Organization" \
--to-lifecycle-environment "My_Lifecycle_Environment"
```

10.8. 将勘误添加到增量内容视图

如果勘误表可用但不可安装，您可以创建一个增量内容视图版本，以将勘误表添加到您的内容主机。例如，如果内容视图是 1.0 版本，它将成为内容视图版本 1.1，并且在发布时，它将变为内容视图版本 2.0。



重要

如果您的内容视图版本旧，则增量添加增强勘误时可能会遇到不兼容的问题。这是因为增强功能通常为存储库中的最新软件设计。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Content Types > Errata。
2. 在 Errata 列表中，点您要应用的勘误的名称。
3. 选择您要对其应用勘误表的内容主机，然后单击 Apply to Hosts。这会向内容视图创建增量更新。
4. 如果要勘误表应用到内容主机，请在发布复选框后立即选择 Apply Errata to Content Hosts。
5. 点 Confirm 以应用勘误。

CLI 过程

1. 列出勘误及其相应的 ID：

```
# hammer erratum list
```

2. 列出不同的内容视图版本以及对应的 ID：

```
# hammer content-view version list
```

3. 将一个勘误应用到 content-view 版本。您可以在以逗号分隔的列表中添加更多 ID。

```
# hammer content-view version incremental-update \  
--content-view-version-id 319 --errata-ids 34068b
```

10.9. 将勘误应用到主机

使用这些步骤检查并将勘误表应用到主机。

先决条件

- 将 Red Hat Satellite 软件仓库与红帽提供的最新勘误同步。如需更多信息，请参阅 [第 4.7 节“同步软件仓库”](#)。
- 将主机注册到 Satellite 服务器上的环境和内容视图。如需更多信息，请参阅 [管理主机](#) 中的 [注册主机](#)。
- 配置主机以进行远程执行。有关运行远程执行作业的更多信息，请参阅 [管理主机](#) 中的 [配置和设置远程作业](#)。

将勘误应用到主机的过程取决于其操作系统。

10.9.1. 将勘误应用到运行 Red Hat Enterprise Linux 9 的主机

使用这个流程查看并把勘误应用到运行 Red Hat Enterprise Linux 9 的主机。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > Content Hosts 并选择您要应用勘误表的主机。
2. 导航到 Errata 选项卡，以查看勘误表的列表。
3. 选择要应用的勘误，然后点 Apply Selected。在确认窗口中，单击 Apply。
4. 在任务更新与所选勘误关联的所有软件包完成后，点 Details 选项卡查看更新的软件包。

CLI 过程

1. 列出主机的所有勘误：

```
# hammer host errata list \  
--host client.example.com
```

2. 查找勘误所属的模块流：

```
# hammer erratum info --id ERRATUM_ID
```

3. 在主机上更新模块流：

```
# dnf update Module_Stream_Name
```

10.9.2. 将勘误应用到运行 Red Hat Enterprise Linux 8 的主机

使用这个流程查看并把勘误应用到运行 Red Hat Enterprise Linux 8 的主机。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > Content Hosts 并选择您要应用勘误表的主机。
2. 导航到 Errata 选项卡，以查看勘误表的列表。
3. 选择要应用的勘误，然后点 Apply Selected。在确认窗口中，单击 Apply。
4. 在任务更新与所选勘误关联的所有软件包完成后，点 Details 选项卡查看更新的软件包。

CLI 过程

1. 列出主机的所有勘误：

```
# hammer host errata list \  
--host client.example.com
```

2. 查找勘误所属的模块流：

```
# hammer erratum info --id ERRATUM_ID
```

3. 在主机上更新模块流：

```
# dnf update Module_Stream_Name
```

10.9.3. 将勘误应用到运行 Red Hat Enterprise Linux 7 的主机

使用这个流程审核并应用到运行 Red Hat Enterprise Linux 7 的主机。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > Content Hosts 并选择您要应用勘误表的主机。
2. 导航到 Errata 选项卡，以查看勘误表的列表。
3. 选择要应用的勘误，然后点 Apply Selected。在确认窗口中，单击 Apply。
4. 在任务更新与所选勘误关联的所有软件包完成后，点 Details 选项卡查看更新的软件包。

CLI 过程

1. 列出主机的所有勘误：

```
# hammer host errata list \
--host client.example.com
```

2. 将最新的勘误应用到主机。识别使用勘误 ID 应用的勘误。
使用远程执行

```
# hammer job-invocation create \
--feature katello_errata_install \
--inputs errata=ERRATUM_ID1,ERRATUM_ID2 \
--search-query "name = client.example.com"
```

10.10. 将勘误应用到多个主机

使用这些步骤审核并应用到多个 RHEL 主机。

前提条件

- 将 Red Hat Satellite 软件仓库与红帽提供的最新勘误同步。如需更多信息，请参阅 [第 4.7 节“同步软件仓库”](#)。
- 将主机注册到 Satellite 服务器上的环境和内容视图。如需更多信息，请参阅 [管理主机中的注册主机](#)。
- 配置主机以进行远程执行。有关运行远程执行作业的更多信息，请参阅 [管理主机中的配置和设置远程作业](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Content Types > Errata。
2. 点击您要应用的勘误的名称。
3. 单击 Content Hosts 选项卡。
4. 选择您要对其应用勘误表的主机，然后单击 应用到主机。
5. 单击 Confirm。

CLI 过程

1. 列出所有可安装的勘误：

```
# hammer erratum list \
--errata-restrict-installable true \
--organization "Default Organization"
```

2. 将其中一个勘误应用到多个主机：
使用远程执行

```
# hammer job-invocation create \
--feature katello_errata_install \
--inputs errata=ERRATUM_ID \
--search-query "applicable_errata = ERRATUM_ID"
```

以下 Bash 脚本对这个勘误可用的每个主机应用勘误：

```
for HOST in hammer --csv --csv-separator "|" host list --search "applicable_errata =
ERRATUM_ID" --organization "Default Organization" | tail -n+2 | awk -F "|" '{ print $2 }'
;
do
  echo "=== Applying to $HOST ==="; hammer host errata apply --host $HOST --errata-
ids ERRATUM_ID1,ERRATUM_ID2 ;
done
```

这个命令识别带有 errata_ID 作为适用勘误的所有主机，然后将勘误应用到每个主机。

3. 要查看勘误是否已成功应用，请在以下命令的输出结果中找到对应的任务：

```
# hammer task list
```

4. 查看所选任务的状态：

```
# hammer task progress --id task_ID
```

10.11. 将勘误应用到主机集合

使用远程执行

```
# hammer job-invocation create \
```

```
--feature katello_errata_install \  
--inputs errata=ERRATUM_ID1,ERRATUM_ID2,... \  
--search-query "host_collection = HOST_COLLECTION_NAME"
```

第 11 章 管理容器镜像

使用 Satellite，您可以从各种源导入容器镜像，并使用内容视图将其分发到外部容器。

有关 Red Hat Enterprise Linux Atomic Host 7 的容器的详情，请参考 Red Hat Enterprise Linux Atomic Host 7 中的[容器入门](#)。

有关 Red Hat Enterprise Linux 8 容器的详情，请参考在[Red Hat Enterprise Linux 8 中构建、运行和管理容器](#)。

有关 Red Hat Enterprise Linux 9 的容器的详情，请参考在[Red Hat Enterprise Linux 9 中构建、运行和管理容器](#)。

11.1. 导入容器镜像

您可以从 Red Hat Registry 或其他镜像 registry 中导入容器镜像仓库。

要使用 CLI 而不是 Satellite Web UI，请参阅[CLI 过程](#)。

存储库发现过程

1. 在 Satellite Web UI 中，进入到 Content > Products 并点 Repo Discovery。
2. 从 Repository Type 列表中，选择 Container Images。
3. 在 Registry to Discover 字段中，输入要从中导入镜像的 registry 的 URL。
4. 在 Registry Username 字段中，输入与容器镜像 registry 的用户名对应的名称。
5. 在 Registry Password 字段中，输入与您输入的用户名对应的密码。
6. 在 Registry Search Parameter 字段中输入您要用来过滤搜索的任何搜索条件，然后点 Discover。
7. 可选：要进一步优化 Discovered Repository 列表，在 Filter 字段中输入您要使用的任何其他搜索条件。
8. 从 Discovered Repository 列表中，选择您要导入的任何存储库，然后单击 Create Selected。
9. 可选：要将此容器存储库的下载策略更改为 按需，请参阅[第 4.11 节“更改存储库的下载策略”](#)。
10. 可选：如果希望创建一个产品，在 Product 列表中选择 New Product。
11. 在 Name 字段中输入产品名称。
12. 可选：在 Repository Name 和 Repository Label 列中，您可以编辑存储库名称和标签。
13. 单击 Run Repository Creation。
14. 完成存储库创建后，您可以单击每个新存储库来查看更多信息。
15. 可选：要过滤您导入到存储库的内容，点存储库，然后导航到 Limit Sync Tags。单击以编辑，并添加您要限制 Satellite 同步内容的任何标签。
16. 在 Satellite Web UI 中，进入到 Content > Products 并选择产品的名称。

17. 选择新存储库，然后点 Sync Now 以启动同步过程。

手动创建存储库的步骤

1. 在 Satellite Web UI 中，进入到 Content > Products。点所需产品的名称。
2. 单击 New repository。
3. 从 Type 列表中，选择 docker。输入存储库的详细信息，然后单击保存。
4. 选择新存储库，然后单击 Sync Now。

后续步骤

- 要查看同步的进度，请导航到 Content > Sync Status 并展开存储库树。
- 同步完成后，您可以点 Container Image Manifests 来列出可用的清单。从列表中，您还可以删除您不需要的任何清单。

CLI 过程

1. 创建自定义 Red Hat Container Catalog 产品：

```
# hammer product create \
--description "My_Description" \
--name "Red Hat Container Catalog" \
--organization "My_Organization" \
--sync-plan "My_Sync_Plan"
```

2. 为容器镜像创建存储库：

```
# hammer repository create \
--content-type "docker" \
--docker-upstream-name "rhel7" \
--name "RHEL7" \
--organization "My_Organization" \
--product "Red Hat Container Catalog" \
--url "http://registry.access.redhat.com/"
```

3. 同步存储库：

```
# hammer repository synchronize \
--name "RHEL7" \
--organization "My_Organization" \
--product "Red Hat Container Catalog"
```

其他资源

- 有关手动创建产品和存储库的详情，请参考 [第 4 章 导入内容](#)。

11.2. 管理容器名称模式

当您使用 Satellite 创建和管理容器时，当容器进入内容视图版本和 Satellite 生命周期环境的不同阶段时，容器名称会在每个阶段进行更改。例如，如果您将容器镜像与来自上游存储库的名称 ssh 同步，当您

将其添加到 Satellite 产品和机构中，然后作为内容视图的一部分发布，则容器镜像可以具有以下名称：**my_organization_production-custom_spin-my_product-custom_ssh**。这可能会在要拉取容器镜像时造成问题，因为容器 registry 只能包含容器名称的一个实例。为了避免 Satellite 命名约定出现问题，您可以设置 registry 名称模式来覆盖默认名称，以确保容器名称明确用于将来的使用。

限制

如果使用 registry 名称模式来管理容器命名约定，因为 registry 命名模式必须生成全局唯一的名称，您可能遇到命名冲突问题。例如：

- 如果您设置了 `repository.docker_upstream_name` registry 名称模式，则无法发布或提升具有相同存储库名称的容器内容到 Production 生命周期的内容视图。
- 如果设置了 `lifecycle_environment.name` registry 名称模式，这可以防止创建具有相同名称的第二个容器存储库。

为容器定义 registry 命名模式时，您必须小心。

流程

要使用 registry 名称模式管理容器命名，请完成以下步骤：

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Lifecycle Environments。
2. 创建生命周期环境或选择编辑现有的生命周期环境。
3. 在 Container Image Registry 区域中，点 Registry Name Pattern 区域右侧的编辑图标。
4. 使用变量列表和示例来确定您需要的 registry 名称模式。
5. 在 Registry Name Pattern 字段中输入您要使用的 registry 名称模式。例如，要使用 `repository.docker_upstream_name`：

```
<%= repository.docker_upstream_name %>
```

6. 点击 Save。

11.3. 管理容器 REGISTRY 身份验证

您可以管理从 Satellite 访问容器镜像的身份验证设置。默认情况下，用户必须进行身份验证才能访问 Satellite 中的容器镜像。

您可以指定是否希望用户对在生命周期环境中访问 Satellite 中的容器镜像进行身份验证。例如，您可能希望允许用户从 Production 生命周期访问容器镜像，而无需任何身份验证要求，并将访问 Development 和 QA 环境限制为经过身份验证的用户。

流程

1. 在 Satellite Web UI 中，进入到 Content > Lifecycle > Lifecycle Environments。
2. 选择您要管理身份验证的生命周期环境。
3. 要允许在这个生命周期环境中对容器进行未经身份验证的访问，请选择 Unauthenticated Pull 复选框。要限制未经身份验证的访问，请清除 Unauthenticated Pull 复选框。
4. 点击 Save。

11.4. 配置 PODMAN 和 DOCKER 以信任证书颁发机构

Podman 使用两个路径来查找 CA 文件，即 `/etc/containers/certs.d/` 和 `/etc/docker/certs.d/`。

将 root CA 文件复制到这些位置之一，其准确路径由服务器主机名决定，并命名文件 `ca.crt`

在以下示例中，根据您的用例，将 `hostname.example.com` 替换为 `satellite.example.com` 或 `Capsule.example.com`。

- 您可能需要使用以下方法创建相关位置：

```
# mkdir -p /etc/containers/certs.d/hostname.example.com
```

或者

```
# mkdir -p /etc/docker/certs.d/hostname.example.com
```

- 对于 podman，请使用：

```
# cp rootCA.pem /etc/containers/certs.d/hostname.example.com/ca.crt
```

- 或者，如果您使用 Docker，请将 root CA 文件复制到对等的 Docker 目录中：

```
# cp rootCA.pem /etc/docker/certs.d/hostname.example.com/ca.crt
```

登录到 registry 时不再需要使用 `--tls-verify=false` 选项：

```
$ podman login hostname.example.com
```

```
Username: admin
```

```
Password:
```

```
Login Succeeded!
```

11.5. 使用容器 REGISTRY

Podman 和 Docker 可用于从容器注册表获取内容。

Capsules 上的容器 registry

在包含内容的 Capsule 上，[Container Gateway](#) Capsule 插件充当容器 registry。它缓存来自 Katello 的身份验证信息，并将传入请求代理到 Pulp。默认情况下，容器网关在 Capsules 上提供内容。

流程

登录到容器 registry：

```
# podman login satellite.example.com
```

列出容器镜像：

```
# podman search satellite.example.com/
```

拉取容器镜像：

`# podman pull satellite.example.com/my-image:<optional_tag>`

第 12 章 管理 ISO 镜像

您可以使用 Satellite 存储 ISO 镜像，可以是来自红帽的 Content Delivery Network 或其他源。您还可以上传其他文件，如虚拟机镜像，并将它们发布到存储库中。

12.1. 从红帽导入 ISO 镜像

Red Hat Content Delivery Network 为某些产品提供 ISO 镜像。导入此内容的过程与为 RPM 内容启用存储库的步骤类似。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Red Hat Repositories。
2. 在 Search 字段中输入镜像名称，例如：**Red Hat Enterprise Linux 7 Server (ISO)**。
3. 在 Available Repositories 窗口中，展开 Red Hat Enterprise Linux 7 Server (ISOs)。
4. 对于 x86_64 7.2 条目，点 Enable 图标为镜像启用存储库。
5. 在 Satellite Web UI 中，进入到 Content > Products 并点 Red Hat Enterprise Linux Server。
6. 单击 Red Hat Enterprise Linux Server 窗口的 Repositories 选项卡，再单击 Red Hat Enterprise Linux 7 Server ISOs x86_64 7.2。
7. 在 Red Hat Enterprise Linux 7 Server ISOs x86_64 7.2 窗口右上角，单击 Select Action 并选择 Sync Now。

查看同步状态

- 在 Satellite Web UI 中，进入到 Content > Sync Status 并展开 Red Hat Enterprise Linux Server。

CLI 过程

1. 找到用于文件存储库的 Red Hat Enterprise Linux Server 产品：

```
# hammer repository-set list \
--product "Red Hat Enterprise Linux Server" \
--organization "My_Organization" | grep "file"
```

2. 为 Red Hat Enterprise Linux 7.2 Server ISO 启用文件存储库：

```
# hammer repository-set enable \
--product "Red Hat Enterprise Linux Server" \
--name "Red Hat Enterprise Linux 7 Server (ISOs)" \
--releasever 7.2 \
--basearch x86_64 \
--organization "My_Organization"
```

3. 在产品中找到存储库：

```
# hammer repository list \
--product "Red Hat Enterprise Linux Server" \
--organization "My_Organization"
```

4. 同步产品中的存储库：

```
# hammer repository synchronize \
--name "Red Hat Enterprise Linux 7 Server ISOs x86_64 7.2" \
--product "Red Hat Enterprise Linux Server" \
--organization "My_Organization"
```

12.2. 导入单个ISO 镜像和文件

使用这个流程手动将ISO 内容和其他文件导入到Satellite 服务器。要导入文件，您可以在Satellite Web UI 或使用Hammer CLI 中完成以下步骤。但是，如果要上传的文件的大小大于15 MB，则必须使用Hammer CLI 将其上传到存储库。

要使用CLI 而不是Satellite Web UI，请参阅[CLI 过程](#)。

流程

1. 在Satellite Web UI 中，进入到Content >Products，然后在Products 窗口中点Create Product。
2. 在Name 字段中输入用于标识产品的名称。此名称填充Label 字段。
3. 可选：在GPG Key 字段中，为产品输入GPG 密钥。
4. 可选：在Sync Plan 列表中，为产品选择一个同步计划。
5. 可选：在Description 字段中输入产品描述。
6. 点击Save。
7. 在Products 窗口中，单击新产品，然后单击Create Repository。
8. 在Name 字段中输入存储库的名称。这会自动填充Label 字段。
9. 从Type 列表中，选择文件。
10. 在Upstream URL 字段中，输入要用作源的registry 的URL。在Upstream Username 和 Upstream Password 字段中添加对应的用户名和密码。
11. 点击Save。
12. 选择新存储库。
13. 导航到Upload File，再单击Browse。
14. 选择.iso 文件，然后单击Upload。

CLI 过程

1. 创建自定义产品：

```
# hammer product create \  
--name "My_ISOs" \  
--sync-plan "Example Plan" \  
--description "My_Product" \  
--organization "My_Organization"
```

2. 创建存储库：

```
# hammer repository create \  
--name "My_ISOs" \  
--content-type "file" \  
--product "My_Product" \  
--organization "My_Organization"
```

3. 将 ISO 文件上传到存储库：

```
# hammer repository upload-content \  
--path ~/bootdisk.iso \  
--id repo_ID \  
--organization "My_Organization"
```

第13章 管理 ANSIBLE 内容

您可以将 Ansible 集合从多个源导入到 Satellite 服务器。

如需有关 Satellite 中的 Ansible 集成的更多信息，[请参阅使用 Ansible 集成管理配置](#)。

13.1. 同步 ANSIBLE 集合

在 Satellite 上，您可以从 Private Automation Hub、console.redhat.com 和其他 Satellite 实例同步 Ansible Collections。在同步后，Ansible Collections 将会出现在 Satellite web UI 菜单的 Content 下作为一个新的仓库类型。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 选择所需的产品名称。
3. 在 Products 窗口中，选择要为其创建存储库的产品名称。
4. 单击 Repositories 选项卡，然后单击 New Repository。
5. 在 Name 字段中输入存储库的名称。
Label 字段会根据名称自动填充。
6. 从 Type 列表中，选择 ansible collection。
7. 在 Upstream URL 字段中，输入上游集合存储库的 URL。
URL 可以是任何 Ansible Galaxy 端点。例如：<https://console.redhat.com/api/automation-hub/>。
8. 可选：在 requirements.yml 字段中，您可以指定您要从端点同步的集合列表及其版本。
如果没有指定集合列表，则会同步来自端点的所有内容。

```
---
collections:
- name: my_namespace.my_collection
  version: 1.2.3
```

如需更多信息，请参阅 Galaxy 用户指南中的[从同一 requirements.yml 文件安装角色和集合](#)。

9. 身份验证。
 - a. 要从 Private Automation Hub 同步 Satellite，请在 Auth Token 字段中输入您的令牌。
如需更多信息，请参阅[连接到 Hub 中的连接私有 Automation Hub](#)。
 - b. 要从 console.redhat.com 同步 Satellite，请在 Auth Token 字段中输入您的令牌，然后在 Auth URL 字段中输入您的 SSO URL。
如需更多信息，请参阅[开始使用自动化中心](#)。
 - c. 要从 Satellite 同步 Satellite，请将这两个身份验证字段留空。
10. 单击 Save。
11. 导航到 Ansible Collections 存储库。

12. 在 *Select Action* 菜单中, 选择 *Sync Now*。

第 14 章 管理自定义文件类型内容

在 Satellite 中，您可能需要管理和分发 SSH 密钥和源代码文件或大型文件，如虚拟机镜像和 ISO 文件。为实现此目的，Red Hat Satellite 中的自定义产品包括用于自定义文件类型的软件仓库。这提供了一种将任意文件合并到产品中的通用方法。

您可以将文件上传到存储库，并从上游 Satellite 服务器同步文件。将文件添加到自定义文件类型存储库时，您可以使用普通的 Satellite 管理功能，如将特定版本添加到内容视图，以提供版本控制，并在各种胶囊服务器上提供文件存储库。您必须使用 `curl -O` 通过 HTTP 或 HTTPS 下载文件。

您只能在自定义产品中创建文件类型存储库，但当您创建存储库源时具有灵活性。您可以在 Satellite 服务器上的目录或远程 HTTP 服务器中创建独立存储库源，然后将该目录的内容同步到 Satellite。当您有多个要添加到 Satellite 存储库的文件时，此方法很有用。

14.1. 为自定义文件类型存储库创建本地源

您可以在使用 Pulp Manifest 安装 Satellite 的基本系统上，从文件的目录创建自定义文件类型存储库源。然后，您可以将文件同步到存储库中，并像任何其他内容类型一样管理自定义文件类型内容。

使用这个流程在安装 Satellite 的基本系统的目录中配置存储库。要在远程服务器的目录中创建文件类型存储库，请参考第 14.2 节“为自定义文件类型存储库创建远程源”。

流程

1. 确保 Utils 存储库已启用。

```
# subscription-manager repos \
--enable=rhel-8-for-x86_64-appstream-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=satellite-utils-6.15-for-rhel-8-x86_64-rpms
```

2. 启用 satellite-utils 模块：

```
# dnf module enable satellite-utils
```

3. 安装 Pulp Manifest 软件包：

```
# satellite-maintain packages install python39-pulp_manifest
```

请注意，此命令将停止 Satellite 服务并重新运行 `satellite-installer`。另外，为了避免停止服务导致的停机，您可以使用以下内容：

```
# satellite-maintain packages unlock
# satellite-maintain packages install python39-pulp_manifest
# satellite-maintain packages lock
```

4. 创建您要用作文件类型存储库的目录，例如：

```
# mkdir -p /var/lib/pulp/local_repos/my_file_repo
```

5. 将父文件夹添加到允许的导入路径中：

```
# satellite-installer --foreman-proxy-content-pulpcore-additional-import-paths
/var/lib/pulp/local_repos
```

6. 在目录中添加文件或创建测试文件：

```
# touch /var/lib/pulp/local_repos/my_file_repo/test.txt
```

7. 运行 Pulp Manifest 命令来创建清单：

```
# pulp-manifest /var/lib/pulp/local_repos/my_file_repo
```

8. 验证清单是否已创建：

```
# ls /var/lib/pulp/local_repos/my_file_repo
PULP_MANIFEST test.txt
```

现在，您可以将本地源导入为自定义文件类型存储库。使用 `file://` URL 方案和目录名称来指定 Upstream URL，如 `file:///var/lib/pulp/local_repos/my_file_repo`。如需更多信息，请参阅 [第 14.3 节“创建自定义文件类型存储库”](#)。

如果更新目录的内容，请重新运行 Pulp 清单并在 Satellite 中同步存储库。如需更多信息，请参阅 [第 4.7 节“同步软件仓库”](#)。

14.2. 为自定义文件类型存储库创建远程源

您可以使用 Pulp Manifest 从位于 Satellite 服务器外部的文件目录创建自定义文件类型存储库源。然后，您可以通过 HTTP 或 HTTPS 将文件同步到 Satellite 服务器上的存储库，并像任何其他内容类型一样管理自定义文件类型内容。

使用这个流程在远程服务器的目录中配置存储库。要在安装 Satellite 服务器的基本系统的目录中创建文件类型存储库，请参阅 [第 14.1 节“为自定义文件类型存储库创建本地源”](#)。

前提条件

- 您有一个运行 Red Hat Enterprise Linux 7 或 8 的服务器，已注册到 Satellite 或 Red Hat CDN。
- 您的服务器具有 Red Hat Enterprise Linux Server 和 Red Hat Satellite Utils 存储库的权利。
- 已安装 HTTP 服务器。有关配置 Web 服务器的更多信息，请参阅 Red Hat Enterprise Linux 8 中的 [设置 Apache HTTP web 服务器](#)，在 Red Hat Enterprise Linux 7 系统管理员指南中的 [部署不同类型的服务器](#) 或 [Apache HTTP 服务器](#)。

流程

1. 在您的服务器上，确保启用了正确的存储库。

- 在 Red Hat Enterprise Linux 8 中：

```
# subscription-manager repos \
--enable=rhel-8-for-x86_64-appstream-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=satellite-utils-6.15-for-rhel-8-x86_64-rpms
```

- 在 Red Hat Enterprise Linux 7 中：

```
# subscription-manager repos \
--enable=rhel-7-server-satellite-utils-6.15-rpms \
--enable=rhel-7-server-rpms
```

2. 启用 `satellite-utils` 模块：

```
# dnf module enable satellite-utils
```

3. 安装 Pulp Manifest 软件包：

- 在 Red Hat Enterprise Linux 8 中：

```
# dnf install python39-pulp_manifest
```

- 在 Red Hat Enterprise Linux 7 中：

```
# yum install tfm-pulpcore-python3-pulp_manifest
```

4. 在 HTTP 服务器的公共文件夹中创建要用作文件类型存储库的目录：

```
# mkdir /var/www/html/pub/my_file_repo
```

5. 在目录中添加文件或创建测试文件：

```
# touch /var/www/html/pub/my_file_repo/test.txt
```

6. 运行 Pulp Manifest 命令来创建清单：

```
# pulp-manifest /var/www/html/pub/my_file_repo
```

7. 验证清单是否已创建：

```
# ls /var/www/html/pub/my_file_repo
PULP_MANIFEST test.txt
```

现在，您可以将远程源导入为自定义文件类型存储库。使用目录的路径指定 Upstream URL，如 http://server.example.com/my_file_repo。如需更多信息，请参阅 [第 14.3 节“创建自定义文件类型存储库”](#)。

如果更新目录的内容，请重新运行 Pulp 清单并在 Satellite 中同步存储库。如需更多信息，请参阅 [第 4.7 节“同步软件仓库”](#)。

14.3. 创建自定义文件类型存储库

创建自定义文件类型存储库的步骤与创建任何自定义内容的步骤相同，除非您创建存储库时，您可以选择文件类型。您必须创建一个产品，然后添加自定义存储库。

要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 选择您要为其创建存储库的产品。
3. 在 Repositories 选项卡上，单击 New Repository。
4. 在 Name 字段中输入存储库的名称。Satellite 根据名称自动完成 Label 字段。
5. 可选：在 Description 字段中输入存储库的描述。
6. 从 Type 列表中，选择 file 作为存储库类型。
7. 可选：在 Upstream URL 字段中输入要用作源的上游存储库的 URL。如果没有输入上游 URL，您可以手动上传软件包。如需更多信息，请参阅第 14.4 节“将文件上传到自定义文件类型存储库”。
8. 选择 Verify SSL 来验证存储库的 SSL 证书是否由可信 CA 签名。
9. 可选：在 Upstream Username 字段中，如果需要身份验证，请输入上游存储库的用户名。如果存储库不需要身份验证，请清除此字段。
10. 可选：在 Upstream Password 字段中，为 upstream 仓库输入对应的密码。如果存储库不需要身份验证，请清除此字段。
11. 可选：在 Upstream Authentication Token 字段中，提供上游存储库用户的令牌以进行身份验证。如果存储库不需要身份验证，请将此字段留空。
12. 从 Mirroring Policy 列表中，选择 Satellite 服务器执行的内容类型。如需更多信息，请参阅第 4.12 节“镜像策略概述”。
13. 可选：在 HTTP Proxy Policy 字段中，选择一个 HTTP 代理。默认情况下，它使用 Global Default HTTP 代理。
14. 可选：您可以清除 Unprotected 复选框，以要求订阅权利证书才能访问此存储库。默认情况下，存储库通过 HTTP 发布。
15. 可选：在 SSL CA Cert 字段中，为存储库选择 SSL CA 证书。
16. 可选：在 SSL Client Cert 字段中，为存储库选择 SSL Client Certificate。
17. 可选：在 SSL Client Key 字段中，为存储库选择 SSL Client Key。
18. 单击 Save 以创建该存储库。

CLI 过程

1. 创建自定义产品：

```
# hammer product create \
--description "My_Files" \
--name "My_File_Product" \
--organization "My_Organization" \
--sync-plan "My_Sync_Plan"
```

表 14.1. hammer product create 命令的可选参数

选项	描述
--gpg-key-id <i>gpg_key_id</i>	GPG 密钥数字标识符
--sync-plan-id <i>sync_plan_id</i>	迁移计划数字标识符
--sync-plan <i>sync_plan_name</i>	要搜索的迁移计划名称

2. 创建文件类型存储库：

```
# hammer repository create \
--content-type "file" \
--name "My_Files" \
--organization "My_Organization" \
--product "My_File_Product"
```

表 14.2. `hammer repository create` 命令的可选参数

选项	描述
--checksum-type <i>sha_version</i>	存储库校验和(sha1 或 sha256)
--download-policy <i>policy_name</i>	下载存储库的策略(immediate 或 on_demand)
--gpg-key-id <i>gpg_key_id</i>	GPG 密钥数字标识符
--gpg-key <i>gpg_key_name</i>	要搜索的键名称
--mirror-on-sync <i>boolean</i>	必须从源中镜像此存储库，并同步时删除过时的软件包？设置为 true 或 false , yes 或 no ,1 或 0。
--publish-via-http <i>boolean</i>	这还必须使用 HTTP 发布？设置为 true 或 false , yes 或 no ,1 或 0。
--upstream-password <i>repository_password</i>	上游存储库用户的密码
--upstream-username <i>repository_username</i>	上游存储库用户（如果需要）
--url <i>My_Repository_URL</i>	远程存储库的 URL
--verify-ssl-on-sync <i>boolean</i>	验证远程存储库的上游 SSL 证书是否由可信 CA 签名？设置为 true 或 false , yes 或 no ,1 或 0。

14.4. 将文件上传到自定义文件类型存储库

使用这个流程将文件上传到自定义文件类型存储库。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 按名称选择自定义产品。
3. 按名称选择文件类型存储库。
4. 点 Browse 搜索并选择您要上传的文件。
5. 点 Upload，将所选文件上传到 Satellite 服务器。
6. 访问发布存储库的 URL 以查看该文件。

CLI 过程

```
# hammer repository upload-content \
--id repo_ID \
--organization "My_Organization" \
--path example_file
```

`--path` 选项可以指示文件、文件目录或文件通配表达式。globs 必须使用单引号或双引号转义。

14.5. 从自定义文件类型存储库下载文件到主机

您可以使用 `curl -O` 通过 HTTPS 下载文件，如果选择了 Unprotected 选项，则可以通过 HTTP 将文件下载到客户端。

先决条件

- 您有一个自定义文件类型存储库。更多信息请参阅 [第 14.3 节“创建自定义文件类型存储库”](#)。
- 您知道要从文件类型存储库下载至客户端的文件名。
- 要使用 HTTPS，您需要客户端上的以下证书：
 - `katello-server-ca.crt` 如需更多信息，请参阅管理 Red Hat Satellite 中的 [导入 Katello Root CA 证书](#)。
 - 机构调试证书。如需更多信息，请参阅管理 Red Hat Satellite 中的 [创建机构调试证书](#)。

流程

1. 在 Satellite Web UI 中，进入到 Content > Products。
2. 按名称选择自定义产品。
3. 按名称选择文件类型存储库。
4. 确保选中 Unprotected 复选框，以访问通过 HTTP 发布的存储库。
5. 复制 Published At URL。
6. 在客户端上，从 Satellite 服务器下载该文件：
 - 对于 HTTPS：

```
# curl \
--cacert ./_katello-server-ca.crt \
--cert ./_My_Organization_key-cert.pem \
--remote-name \
https://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_Product_Label/My_Repository_Label/My_File
```

- 对于 HTTP :

```
# curl \
--remote-name \
http://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_Product_Label/My_Repository_Label/My_File
```

CLI 过程

1. 列出文件类型存储库。

```
# hammer repository list --content-type file
---|-----|-----|-----|-----|
ID | NAME      | PRODUCT          | CONTENT TYPE | URL
---|-----|-----|-----|-----|
7  | My_Files  | My_File_Product  | file         |
---|-----|-----|-----|-----|
```

2. 显示存储库信息。

```
# hammer repository info \
--name "My_Files" \
--organization-id My_Organization_ID \
--product "My_File_Product"
```

如果启用了 Unprotected，输出类似如下：

```
Publish Via HTTP: yes
Published At:
https://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_File_Product_Label/My_Files_Label/
```

如果没有启用 Unprotected，输出类似如下：

```
Publish Via HTTP: no
Published At:
https://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_File_Product_Label/My_Files_Label/
```

3. 在客户端上，从 Satellite 服务器下载该文件：

- 对于 HTTPS :

```
# curl \
--cacert ./_katello-server-ca.crt \
--cert ./_My_Organization_key-cert.pem \
```

```
--remote-name \  
https://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_Product_Label/My_Repository_Label/My_File
```

- 对于 HTTP :

```
# curl \  
--remote-name \  
http://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_Product_Label/My_Repository_Label/My_File
```

附录 A. 使用 NFS 共享进行内容存储

您的环境需要足够的硬盘空间才能履行内容存储。在某些情况下，使用 NFS 共享来存储此内容非常有用。本附录演示了如何在 Satellite 服务器的内容管理组件上挂载 NFS 共享。



重要

为 `/var/lib/pulp` 文件系统使用高带宽、低延迟存储。Red Hat Satellite 有很多 I/O 密集型操作，因此高延迟、低带宽存储可能会遇到性能下降的问题。

流程

1. 创建 NFS 共享。这个示例使用 `nfs.example.com:/Satellite/pulp` 的共享。确保此共享为 Satellite 服务器及其 `apache` 用户提供适当的权限。

2. 停止 Satellite 服务器中的 Satellite 服务：

```
# satellite-maintain service stop
```

3. 确保 Satellite 服务器安装了 `nfs-utils` 软件包：

```
# satellite-maintain packages install nfs-utils
```

4. 您需要将 `/var/lib/pulp` 的现有内容复制到 NFS 共享。首先，将 NFS 共享挂载到临时位置：

```
# mkdir /mnt/temp
# mount -o rw nfs.example.com:/Satellite/pulp /mnt/temp
```

将 `/var/lib/pulp` 的现有内容复制到临时位置：

```
# cp -r /var/lib/pulp/* /mnt/temp/.
```

5. 设置共享上所有文件的权限，以使用 `pulp` 用户。

6. 卸载临时存储位置：

```
# umount /mnt/temp
```

7. 删除 `/var/lib/pulp` 的现有内容：

```
# rm -rf /var/lib/pulp/*
```

8. 编辑 `/etc/fstab` 文件并添加以下行：

```
nfs.example.com:/Satellite/pulp /var/lib/pulp nfs
rw,hard,intr,context="system_u:object_r:pulpcore_var_lib_t:s0"
```

这使得挂载在系统重启后持久保留。确保包含 SELinux 上下文。

9. 启用挂载：

```
# mount -a
```

10. 确认 NFS 共享挂载到 `var/lib/pulp` :

```
# df
Filesystem                1K-blocks  Used Available Use% Mounted on
...
nfs.example.com:/Satellite/pulp 309506048 58632800 235128224 20% /var/lib/pulp
...
```

另外，确认 `var/lib/pulp` 上的挂载中存在现有内容：

```
# ls /var/lib/pulp
```

11. 在 Satellite 服务器上启动 Satellite 服务：

```
# satellite-maintain service start
```

Satellite 服务器现在使用 NFS 共享来存储内容。运行内容同步以确保 NFS 共享按预期工作。如需更多信息，请参阅 [第 4.7 节“同步软件仓库”](#)。

附录 B. 导入 KICKSTART 软件仓库

Kickstart 软件仓库不是由内容 ISO 镜像提供。要在断开连接的 Satellite 中使用 Kickstart 软件仓库，您必须为您要使用的 Red Hat Enterprise Linux 版本下载二进制 DVD ISO 文件，并将 Kickstart 文件复制到 Satellite。

要为 Red Hat Enterprise Linux 9 导入 Kickstart 软件仓库，请完成 [第 B.1 节“为 Red Hat Enterprise Linux 9 导入 Kickstart 软件仓库”](#)。

要为 Red Hat Enterprise Linux 8 导入 Kickstart 软件仓库，请完成 [第 B.2 节“为 Red Hat Enterprise Linux 8 导入 Kickstart 软件仓库”](#)。

要为 Red Hat Enterprise Linux 7 导入 Kickstart 软件仓库，请完成 [第 B.3 节“为 Red Hat Enterprise Linux 7 导入 Kickstart 软件仓库”](#)。

B.1. 为 RED HAT ENTERPRISE LINUX 9 导入 KICKSTART 软件仓库

使用这个流程为 Red Hat Enterprise Linux 9 导入 Kickstart 软件仓库。

流程

1. 访问红帽客户门户，请访问 access.redhat.com/downloads 并登录。
2. 点 Red Hat Enterprise Linux。
3. 从列表中选择产品变体和产品版本。例如，产品变体 Red Hat Enterprise Linux for x86_64 和产品版本 9.0。
4. 找到完整的安装镜像，例如，Red Hat Enterprise Linux 9.0 Binary DVD，然后点 Download Now。请注意，您无法使用最小 ISO 置备主机。
5. 下载完成后，将 ISO 镜像复制到 Satellite 服务器。
6. 在 Satellite 服务器上，创建一个挂载点，并将 ISO 镜像临时挂载到该位置：

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

将 `rhel-binary-dvd.iso` 替换为 ISO 镜像的名称。

7. 为 Red Hat Enterprise Linux 9 AppStream 和 BaseOS Kickstart 软件仓库创建目录：

```
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/appstream/kickstart
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/baseos/kickstart
```

8. 从 ISO 镜像复制 kickstart 文件：

```
# cp -a /mnt/iso/AppStream/* /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/appstream/kickstart
# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/baseos/kickstart
```

请注意，对于 BaseOS，还必须复制 `/mnt/iso/images/` 目录的内容。

9. 在列表文件中添加以下条目：

在 `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/listing` 文件中，附加带有新行的 kickstart。

在 `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/listing` 文件中，附加带有新行的 kickstart。

在 `/var/www/html/pub/satellite-import/content/dist/rhel8/listing` 文件中，附加带有新行的版本号。例如，对于 Red Hat Enterprise Linux 9.0 二进制 ISO，请附加 9.0。

10. 从 ISO 镜像复制 `.treeinfo` 文件：

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo
```

11. 打开 `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo` 文件进行编辑。

12. 在 `[general]` 部分中，进行以下更改：

- 将 `packagedir = AppStream/Packages` 改为 `packagedir = Packages`
- 将 `repository = AppStream` 更改为 `repository = .`
- 将 `variant = AppStream` 改为 `variant = BaseOS`
- 将 `变体 = AppStream,BaseOS` 改为 `variants = BaseOS`

13. 在 `[tree]` 部分中，将 `variants = AppStream,BaseOS` 更改为 `variants = BaseOS`。

14. 在 `[variant-BaseOS]` 部分进行以下更改：

- 将 `软件包 = BaseOS/Packages` 更改为 `packages= Packages`
- 将 `repository = BaseOS` 更改为 `repository = .`

15. 删除 `[media]` 和 `[variant-AppStream]` 部分。

16. 保存并关闭该文件。

17. 验证 `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo` 文件是否具有以下格式：

```
[checksums]
images/efiboot.img =
sha256:c01c18acc6778d6e66c8d0872bac59bfd7219ccf3cfa70a5c605c0fb37f33a83
images/install.img =
sha256:ddd08e5a5d92edee150f91ff4f12f39253eae72ff496465cf1b2766fe4a4df49
images/pxeboot/initrd.img =
sha256:a09a8ec89d485d71ed1bdad83584d6d816e67448221172d9aad97886cd70adca
images/pxeboot/vmlinuz =
sha256:6e523d7c3266e26c695923ab12b2873b16b0c61fb2e48ade608ad8998821584b

[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
```

```
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 9.0.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = BaseOS
variants = BaseOS
version = 9.0.0

[header]
type = productmd.treeinfo
version = 1.2

[images-x86_64]
efiboot.img = images/efiboot.img
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[images-xen]
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 9.0.0

[stage2]
mainimage = images/install.img

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = BaseOS

[variant-BaseOS]
id = BaseOS
name = BaseOS
packages = Packages
repository = .
type = variant
uid = BaseOS
```

18. 打开 `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo` 文件进行编辑。
19. 在 `[general]` 部分中，进行以下更改：
 - 将 `packagedir = AppStream/Packages` 改为 `packagedir = Packages`
 - 将 `repository = AppStream` 更改为 `repository = .`

- 更改变体 = `AppStream,BaseOS` to `variants = AppStream`
20. 在 `[tree]` 部分中, 将 `variants = AppStream,BaseOS` 改为 `variants = AppStream`
 21. 在 `[variant-AppStream]` 部分中进行以下更改 :
 - 将软件包 = `AppStream/Packages` 改为 `packages = Packages`
 - 将 `repository = AppStream` 更改为 `repository = .`
 22. 删除文件中的以下部分 : `[checksums]`, `[images-x86_64]`, `[images-xen]`, `[media]`, `[stage2]`, `[variant-BaseOS]`。
 23. 保存并关闭该文件。
 24. 验证 `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo` 文件是否具有以下格式 :

```
[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 9.0.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = AppStream
variants = AppStream
version = 9.0.0

[header]
type = productmd.treeinfo
version = 1.2

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 9.0.0

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = AppStream

[variant-AppStream]
id = AppStream
name = AppStream
packages = Packages
repository = .
type = variant
uid = AppStream
```

25. 如果您不打算使用挂载的二进制 DVD ISO 镜像, 卸载并删除该目录 :

```
# umount /mnt/iso
# rmdir /mnt/iso
```

26. 在 Satellite Web UI 中，启用 Kickstart 存储库。

B.2. 为 RED HAT ENTERPRISE LINUX 8 导入 KICKSTART 软件仓库

使用这个流程为 Red Hat Enterprise Linux 8 导入 Kickstart 软件仓库。

流程

1. 访问红帽客户门户，请访问 access.redhat.com/downloads 并登录。
2. 点 Red Hat Enterprise Linux。
3. 从列表中选择产品变体和产品版本。例如，产品变体 Red Hat Enterprise Linux for x86_64 和产品版本 8.1。
4. 找到完整的安装镜像，例如：Red Hat Enterprise Linux 8.1 Binary DVD，然后点 Download Now。
5. 下载完成后，将 ISO 镜像复制到 Satellite 服务器。
6. 在 Satellite 服务器上，创建一个挂载点，并将 ISO 镜像临时挂载到该位置：

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

将 `rhel-binary-dvd.iso` 替换为 ISO 镜像的名称。

7. 为 Red Hat Enterprise Linux 8 AppStream 和 BaseOS Kickstart 软件仓库创建目录：

```
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart
```

8. 从 ISO 镜像复制 kickstart 文件：

```
# cp -a /mnt/iso/AppStream/* /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart
```

请注意，对于 BaseOS，还必须复制 `/mnt/iso/images/` 目录的内容。

9. 在列表文件中添加以下条目：
在 `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/listing` 文件中，附加带有新行的 kickstart。

在 `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/listing` 文件中，附加带有新行的 kickstart。

在 `/var/www/html/pub/satellite-import/content/dist/rhel8/listing` 文件中，附加带有新行的版本号。例如，对于 Red Hat Enterprise Linux 8.1 二进制 ISO，请附加 8.1。

10. 从 ISO 镜像复制 `.treeinfo` 文件：

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo
```

11. 打开 `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo` 文件进行编辑。
12. 在 `[general]` 部分中，进行以下更改：
 - 将 `packagedir = AppStream/Packages` 改为 `packagedir = Packages`
 - 将 `repository = AppStream` 更改为 `repository = .`
 - 将 `variant = AppStream` 改为 `variant = BaseOS`
 - 将 `变体 = AppStream,BaseOS` 改为 `variants = BaseOS`
13. 在 `[tree]` 部分中，将 `variants = AppStream,BaseOS` 更改为 `variants = BaseOS`。
14. 在 `[variant-BaseOS]` 部分进行以下更改：
 - 将 `软件包 = BaseOS/Packages` 更改为 `packages = Packages`
 - 将 `repository = BaseOS` 更改为 `repository = .`
15. 删除 `[media]` 和 `[variant-AppStream]` 部分。
16. 保存并关闭该文件。
17. 验证 `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo` 文件是否具有以下格式：

```
[checksums]
images/efiboot.img =
sha256:c01c18acc6778d6e66c8d0872bac59bfd7219ccf3cfa70a5c605c0fb37f33a83
images/install.img =
sha256:ddd08e5a5d92edee150f91ff4f12f39253eae72ff496465cf1b2766fe4a4df49
images/pxeboot/initrd.img =
sha256:a09a8ec89d485d71ed1bdad83584d6d816e67448221172d9aad97886cd70adca
images/pxeboot/vmlinuz =
sha256:6e523d7c3266e26c695923ab12b2873b16b0c61fb2e48ade608ad8998821584b

[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 8.1.0
packagedir = Packages
platforms = x86_64,xen
repository = .
```

```
timestamp = 1571146127
variant = BaseOS
variants = BaseOS
version = 8.1.0

[header]
type = productmd.treeinfo
version = 1.2

[images-x86_64]
efiboot.img = images/efiboot.img
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[images-xen]
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 8.1.0

[stage2]
mainimage = images/install.img

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = BaseOS

[variant-BaseOS]
id = BaseOS
name = BaseOS
packages = Packages
repository = .
type = variant
uid = BaseOS
```

18. 打开 `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo` 文件进行编辑。
19. 在 `[general]` 部分中, 进行以下更改:
 - 将 `packagedir = AppStream/Packages` 改为 `packagedir = Packages`
 - 将 `repository = AppStream` 更改为 `repository = .`
 - 更改变体 = `AppStream,BaseOS` to `variants = AppStream`
20. 在 `[tree]` 部分中, 将 `variants = AppStream,BaseOS` 改为 `variants = AppStream`
21. 在 `[variant-AppStream]` 部分中进行以下更改:
 - 将软件包 = `AppStream/Packages` 改为 `packages = Packages`

- 将 `repository = AppStream` 更改为 `repository = .`
22. 删除文件中的以下部分：`[checksums]`, `[images-x86_64]`, `[images-xen]`, `[media]`, `[stage2]`, `[variant-BaseOS]`。
 23. 保存并关闭该文件。
 24. 验证 `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo` 文件是否具有以下格式：

```
[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 8.1.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = AppStream
variants = AppStream
version = 8.1.0

[header]
type = productmd.treeinfo
version = 1.2

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 8.1.0

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = AppStream

[variant-AppStream]
id = AppStream
name = AppStream
packages = Packages
repository = .
type = variant
uid = AppStream
```

25. 如果您不打算使用挂载的二进制 DVD ISO 镜像，卸载并删除该目录：

```
# umount /mnt/iso
# rmdir /mnt/iso
```

26. 在 Satellite Web UI 中，启用 Kickstart 存储库。

B.3. 为 RED HAT ENTERPRISE LINUX 7 导入 KICKSTART 软件仓库

使用这个流程为 Red Hat Enterprise Linux 7 导入 Kickstart 软件仓库。

流程

1. 访问红帽客户门户，请访问access.redhat.com/downloads 并登录。
2. 点 Red Hat Enterprise Linux。
3. 点 Product Variant 列表上面的 Switch to version 7 and below。
4. 从列表中选择产品变体和产品版本。例如，产品变体 Red Hat Enterprise Linux for x86_64 和产品版本 7.9。
5. 找到完整的安装镜像，例如：Red Hat Enterprise Linux 7.9 Binary DVD，然后点 Download Now。
6. 下载完成后，将 ISO 镜像复制到 Satellite 服务器。
7. 在 Satellite 服务器上，创建一个挂载点，并将 ISO 镜像临时挂载到该位置：

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

将 `rhel-binary-dvd.iso` 替换为 ISO 镜像的名称。

8. 创建 Kickstart 目录：

```
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel/server/7/7.9/x86_64/kickstart/
```

9. 从 ISO 镜像复制 kickstart 文件：

```
# cp -a /mnt/iso/* /var/www/html/pub/satellite-
import/content/dist/rhel/server/7/7.9/x86_64/kickstart/
```

10. 在列表文件中添加以下条目：

在 `/var/www/html/pub/satellite-import/content/dist/rhel/server/7/listing` 文件中，附加带有新行的版本号。例如，对于 Red Hat Enterprise Linux 7.9 ISO，请附加 7.9。

在 `/var/www/html/pub/satellite-import/content/dist/rhel/server/7/7.9/listing` 文件中，附加带有新行的架构。例如，x86_64。

在 `/var/www/html/pub/satellite-import/content/dist/rhel/server/7/7.9/x86_64/listing` 文件中，附加带有新行的 kickstart。

11. 从 ISO 镜像复制 .treeinfo 文件：

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel/server/7/7.9/x86_64/kickstart/treeinfo
```

12. 如果您不打算使用挂载的二进制 DVD ISO 镜像，卸载并删除该目录：

```
# umount /mnt/iso
# rmdir /mnt/iso
```

13. 在 Satellite Web UI 中, 启用 Kickstart 存储库。