



Red Hat Satellite 6.11

管理内容

管理红帽和自定义源内容的指南

Red Hat Satellite 6.11 管理内容

管理红帽和自定义源内容的指南

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 使用一组在应用程序生命周期提升的内容视图来管理此内容。本指南介绍了如何创建适用于您的机构和内容视图的应用程序生命周期，该视图在生命周期阶段中造成主机状态。这些内容视图最终形成在 Red Hat Satellite 6 环境中置备和更新主机的基础。

目录

向红帽文档提供反馈	5
第 1 章 内容管理简介	6
第 2 章 内容管理类型概述	7
第 3 章 管理机构	8
3.1. 创建机构	8
3.2. 设置机构上下文	9
3.3. 创建机构调试证书	9
3.4. 使用机构调试证书浏览存储库内容	9
3.5. 删除机构	10
第 4 章 管理位置	12
4.1. 创建位置	12
4.2. 创建多个位置	12
4.3. 设置位置上下文	13
4.4. 删除位置	13
第 5 章 管理红帽订阅	14
5.1. 将 RED HAT SUBSCRIPTION 清单导入到 SATELLITE 服务器中	14
5.2. 查找红帽订阅	15
5.3. 将红帽订阅添加到订阅分配中	15
5.4. 从订阅分配中删除红帽订阅	16
5.5. 更新和刷新红帽订阅	16
5.6. 将红帽订阅附加到内容主机	16
5.7. 更新多个主机上的红帽订阅	17
第 6 章 导入内容	19
6.1. 在 SATELLITE 中使用产品和软件仓库	19
6.2. 导入自定义 SSL 证书	19
6.3. 创建自定义产品	20
6.4. 添加自定义 RPM 存储库	20
6.5. 启用红帽软件仓库	22
6.6. 同步软件仓库	23
6.7. 下载策略概述	24
6.8. 更改默认下载策略	25
6.9. 更改存储库的下载策略	25
6.10. 镜像策略概述	26
6.11. 更改存储库的镜像策略	26
6.12. 将内容上传到自定义 RPM 存储库	27
6.13. 配置 SELINUX 以在自定义端口中允许内容同步	27
6.14. 恢复修正的存储库	28
6.15. 添加 HTTP 代理	29
6.16. 更改一个产品的 HTTP 代理策略	30
6.17. 更改一个存储库的 HTTP 代理策略	30
6.18. 创建同步计划	31
6.19. 将同步计划分配给产品	32
6.20. 为多个产品分配同步计划	32
6.21. 限制同步 CONCURRENCY	33
6.22. 导入自定义 GPG 密钥	33
6.23. 在 SATELLITE 中将自定义软件仓库限制为 RHEL 9	34

第 7 章 管理应用程序生命周期	35
7.1. 应用程序生命周期介绍	35
7.2. 在应用程序生命周期期间提升内容	36
7.3. 创建生命周期阶段路径	37
7.4. 从 SATELLITE 服务器中删除执行环境	38
7.5. 从 CAPSULE 服务器中删除执行环境	38
7.6. 将虚拟环境添加到 CAPSULE 服务器	39
第 8 章 管理内容视图	41
8.1. 创建内容视图	42
8.2. 查看模块流	43
8.3. 提升内容视图	43
8.4. 在机构中推广跨所有生命周期环境的内容视图	44
8.5. 复合内容视图概述	45
8.6. 创建 COMPOSITE 内容视图	46
8.7. 内容过滤器概述	48
8.8. 解决软件包依赖项	49
8.9. 为内容视图启用依赖解析	50
8.10. 内容过滤器示例	50
8.11. 为 YUM 内容创建内容过滤器	51
8.12. 删除内容视图版本	53
第 9 章 在 SATELLITE 服务器之间同步内容	54
9.1. 如何使用导出和导入同步内容	54
9.2. 同步自定义存储库	56
9.3. 导出库环境	57
9.4. 以递增方式导出库环境	58
9.5. 导出内容视图版本	58
9.6. 逐渐导出内容视图版本	60
9.7. 导出存储库	60
9.8. 永久导出存储库	62
9.9. 跟踪您的导出	62
9.10. 导入到库环境	63
9.11. 导入内容视图版本	64
9.12. 导入仓库	65
9.13. 使用HAMMER CLI CHEAT SHEET 导出和导入内容	65
第 10 章 管理激活码	67
10.1. 创建激活码	67
10.2. 更新与激活码关联的订阅	70
10.3. 使用激活码进行主机注册	71
10.4. 启用自动附加	72
10.5. 设置服务级别	73
第 11 章 管理勘误	74
11.1. 检查可用勘误	74
11.2. 可用于勘误搜索的参数	76
11.3. 应用可安装勘误	76
11.4. 订阅勘误通知	77
11.5. 存储库依赖解析的限制	77
11.6. 为勘误创建内容视图过滤器	77
11.7. 将勘误添加到事件内容视图	79
11.8. 将勘误应用到主机	80
11.9. 将勘误应用到多个主机	81

11.10. 将勘误应用到主机组	83
第 12 章 管理容器镜像	84
12.1. 导入容器镜像	84
12.2. 管理容器名称模式	85
12.3. 管理容器 REGISTRY 身份验证	86
12.4. 配置 PODMAN 和 DOCKER 以信任证书颁发机构	86
12.5. 使用容器 REGISTRY	87
第 13 章 管理 ISO 镜像	88
13.1. 从红帽导入 ISO 镜像	88
13.2. 导入单个 ISO 镜像和文件	89
第 14 章 管理自定义文件类型内容	91
14.1. 创建自定义文件类型存储库	91
14.2. 在本地目录中创建自定义文件类型存储库	93
14.3. 创建远程文件类型存储库	95
14.4. 将文件上传到自定义文件类型存储库	97
14.5. 从自定义文件类型存储库下载文件到主机	98
附录 A. 将 NFS 共享用于内容存储	100
附录 B. 导入 KICKSTART 存储库	102
B.1. 为 RED HAT ENTERPRISE LINUX 7 导入 KICKSTART 存储库	102
B.2. 为 RED HAT ENTERPRISE LINUX 8 导入 KICKSTART 软件仓库	103

向红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

您可以通过在 Bugzilla 中记录一个 ticket 来提交反馈：

1. 导航到 [Bugzilla](#) 网站。
2. 在 **Component** 字段中，使用 **Documentation**。
3. 在 **Description** 字段中，输入您要改进的建议。包括文档相关部分的链接。
4. 点 **Submit Bug**。

第 1 章 内容管理简介

在 Satellite 的上下文中，*内容* 定义为系统上安装的软件。这包括，但不仅限于基础操作系统、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

这为机构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方法来存储红帽内容并以不同的方式对其进行组织。

第 2 章 内容管理类型概述

使用 Red Hat Satellite，您可以管理以下内容类型：

RPM 软件包

从与您的红帽订阅相关的仓库中导入 RPM 软件包。Satellite 服务器从红帽的 Content Delivery Network 下载 RPM 文件，并将它们存储在本地。您可以在内容视图中使用这些存储库及其 RPM 文件。

Kickstart Trees

导入用于创建系统的 kickstart 树。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 还包含一些预定义的 kickstart 模板，以及创建自己的 kickstart 模板，可用于置备系统和自定义安装。

您还可以在 Satellite 中管理其他类型的自定义内容。例如：

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 镜像和客户机镜像。

您可以使用流程为您需要的任何类型内容添加自定义内容，例如 SSL 证书和 OVAL 文件。

第 3 章 管理机构

组织根据所有权、目的、内容、安全级别或其他部门将 Red Hat Satellite 资源分成逻辑组。您可以通过 Red Hat Satellite 创建和管理多个机构，然后将红帽订阅划分并分配到每个机构。这提供了一种方式来管理一个管理系统下多个组织的内容。以下是机构管理的一些示例：

单个机构

带有简单的系统管理链的小型业务。在这种情况下，您可以为业务创建一个组织，并为它分配内容。

多个机构

拥有多个较小的商业单位的大型公司。例如，具有单独的系统管理和软件开发组的公司。在这种情况下，您可以为公司及其拥有的每个业务部门创建组织。这会为每个独立保留系统基础架构。然后，您可以根据每个组织的需求将内容分配给每个组织。

外部机构

为其他组织管理外部系统的公司。例如，为客户提供云计算和 Web 托管资源的公司。在这种情况下，您可以为公司自己的系统基础架构创建一个组织，然后为每个外部业务创建一个机构。然后，您可以根据需要为每个机构分配内容。

Red Hat Satellite 的默认安装有一个名为 **Default_Organization** 的默认机构。

新用户

如果没有为新用户分配默认机构，则其访问权限会被限制。要为用户授予系统权限，请将它们分配到默认机构。当用户下次登录 Satellite 时，用户帐户具有正确的系统权限。

3.1. 创建机构

使用这个流程创建机构。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Administer > Organizations**。
2. 单击 **New Organization**。
3. 在 **Name** 字段中输入机构的名称。
4. 在 **Label** 字段中输入机构的唯一标识符。这用于创建和映射某些资产，如内容存储的目录。使用字母、数字、下划线和短划线，但没有空格。
5. 可选：在 **Description** 字段中输入机构的描述。
6. 点 **Submit**。
7. 如果您的主机没有分配机构，请选择您要添加到机构中的主机，然后单击 **Proceed to Edit**。
8. 在 **Edit** 页面中，分配您要添加到机构中的基础架构资源。这包括网络资源、安装介质、Kickstart 模板和其他参数。您可以随时通过导航到 **Administer > Organizations**，然后选择要编辑的机构来返回到此页面。
9. 点 **Submit**。

CLI 过程

1. 运行以下命令来创建机构：

```
# hammer organization create \
--name "My_Organization" \
--label "My_Organization_Label" \
--description "My_Organization_Description"
```

2. 可选：要编辑机构，请输入 **hammer organization update** 命令。例如，以下命令为机构分配一个计算资源：

```
# hammer organization update \
--name "My_Organization" \
--compute-resource-ids 1
```

3.2. 设置机构上下文

机构上下文定义了用于主机及其相关资源的组织。

流程

机构菜单是菜单栏中的第一个菜单项，位于 Satellite Web UI 的左上角。如果您还没有选择当前的机构，则菜单显示 **Any Organization**。点 **Any Organization** 按钮，再选择要使用的机构。

CLI 过程

在使用 CLI 时，将 **--organization "My_Organization"** 或 **--organization-label "My_Organization_Label"** 作为一个选项。例如：

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

此命令输出为 *My_Organization* 分配的订阅。

3.3. 创建机构调试证书

如果需要您的机构的 debug 证书，请使用以下步骤。

流程

1. 在 Satellite Web UI 中，导航到 **Administer > Organizations**。
2. 选择您要为其生成调试证书的机构。
3. 点 **Generate and Download**。
4. 将证书文件保存到安全位置。

置备模板的调试证书

如果置备模板下载在要下载的机构中不存在，则会自动生成调试证书。

3.4. 使用机构调试证书浏览存储库内容

如果您使用该机构的调试证书，您可以使用 Web 浏览器或通过 API 查看机构的存储库内容。

先决条件

1. 创建并下载机构证书，如第 3.3 节“创建机构调试证书”所述。
2. 为默认机构打开 X.509 证书：

```
$ vi 'Default Organization-key-cert.pem'
```

3. 将文件的内容从 `-----BEGIN RSA PRIVATE KEY-----` 复制到 `-----END RSA PRIVATE KEY-----` 中，并将其复制到 `key.pem` 文件中。
4. 将文件的内容从 `-----BEGIN CERTIFICATE-----` 复制到 `-----END CERTIFICATE-----`，并将其复制到 `cert.pem` 文件中。

流程

要使用浏览器，您必须首先将 X.509 证书转换为浏览器支持的格式，然后导入证书。

对于 Firefox 用户

要在 Firefox 中使用机构调试证书，请完成以下步骤：

1. 要创建 PKCS12 格式证书，请输入以下命令：

```
$ openssl pkcs12 -keypbe PBE-SHA1-3DES -certpbe PBE-SHA1-3DES -export -in cert.pem
-inkey key.pem -out My_Organization_Label.pfx -name My_Organization
```

2. 在 Firefox 浏览器中，导航到 **Edit > Preferences > Advanced Tab**。
3. 选择 **View Certificates**，然后单击 **Your Certificates** 选项卡。
4. 点 **Import** 并选择要加载的 **.pfx** 文件。
5. 在地址栏中输入以下 URL，以浏览所有仓库的可访问路径并检查其内容：

```
https://satellite.example.com/pulp/content/
```

因此，pulp 使用机构标签，因此您必须在 URL 中输入机构标签。

对于 CURL 用户

要将机构调试证书与 CURL 搭配使用，请输入以下命令：

```
$ curl -k --cert cert.pem --key key.pem \
https://satellite.example.com/pulp/content/My_Organization_Label/Library/content/dist/rhel/server/7/7S
erver/x86_64/os/
```

确保 `cert.pem` 和 `key.pem` 的路径是正确的绝对路径，否则命令会静默地失败。

3.5. 删除机构

如果机构没有与任何生命周期阶段或主机组关联，您可以删除机构。如果有与您要删除的机构关联的执行环境或主机组，请进入到 **Administer > Organization** 并点相关机构来删除它们。不要删除安装期间创建的默认机构，因为默认机构是 Satellite 环境中任何未关联主机的占位符。在任意给定时间，环境中必须至少有一个机构。

流程

1. 在 Satellite Web UI 中，导航到 **Administer > Organizations**。
2. 从您要删除的机构名称右侧的列表中，选择 **Delete**。
3. 单击 **OK** 以删除组织。

CLI 过程

1. 输入以下命令来检索您要删除的机构 ID：

```
# hammer organization list
```

在输出中，记下您要删除的机构的 ID。

2. 输入以下命令删除机构：

```
# hammer organization delete --id Organization_ID
```

第 4 章 管理位置

位置功能类似于组织：它们提供了一种分组资源和分配主机的方法。机构和位置有以下概念区别：

- 位置基于物理或地理设置。
- 位置具有分层结构。

4.1. 创建位置

使用这个流程创建位置，以便您可以按位置管理主机和资源。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Administer > Locations**。
2. 单击 **New Location**。
3. 可选：从 **Parent** 列表中选择父位置。这会创建一个位置层次结构。
4. 在 **Name** 字段中输入位置的名称。
5. 可选：在 **Description** 字段中输入位置的描述。
6. 点 **Submit**。
7. 如果您的主机没有分配位置，请添加您要分配给新位置的任何主机，然后单击 **Proceed to Edit**。
8. 分配您要添加到位置的任何基础架构资源。这包括网络资源、安装介质、Kickstart 模板和其他参数。您可以随时返回此页面，方法是导航到 **Administer > Locations**，然后选择要编辑的位置。
9. 点 **Submit** 保存您的更改。

CLI 过程

- 输入以下命令来创建位置：

```
# hammer location create \  
--description "My_Location_Description" \  
--name "My_Location" \  
--parent-id "My_Location_Parent_ID"
```

4.2. 创建多个位置

以下示例 Bash 脚本创建三个位置 - London、Munich、Chhi - 并将其分配到 Example 机构。

```
ORG="Example Organization"  
LOCATIONS="London Munich Boston"  
  
for LOC in ${LOCATIONS}  
do
```

```
hammer location create --name "${LOC}"
hammer location add-organization --name "${LOC}" --organization "${ORG}"
done
```

4.3. 设置位置上下文

位置上下文定义用于主机及其关联的资源的位置。

流程

位置菜单是菜单栏中的第二个菜单项，位于 Satellite Web UI 的左上角。如果您还没有选择当前位置，则菜单会显示 **任何位置**。点 **Any location** 并选择要使用的位置。

CLI 过程

在使用 CLI 时，将 `--location "My_Location"` 或 `--location-id "My_Location_ID"` 作为选项。例如：

```
# hammer host list --location "My_Location"
```

此命令列出与 `My_Location` 位置关联的主机。

4.4. 删除位置

如果位置没有与任何生命周期阶段或主机组关联，您可以删除位置。如果有与您要删除的位置关联的执行环境或主机组，请进入到 **Administer > Locations** 并点相关位置来删除它们。不要删除安装期间创建的默认位置，因为默认位置是 Satellite 环境中任何未关联主机的占位符。在任意给定时间环境中必须至少有一个位置。

流程

1. 在 Satellite Web UI 中，导航到 **Administer > Locations**。
2. 从您要删除的位置名称右侧的列表中选择 **Delete**。
3. 单击 **OK** 以删除位置。

CLI 过程

1. 输入以下命令来检索您要删除的位置的 ID：

```
# hammer location list
```

在输出中，记下您要删除的位置的 ID。

2. 输入以下命令删除位置：

```
# hammer location delete --id Location ID
```

第 5 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[使用红帽订阅管理指南中的使用清单](#)。

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 需要为 Satellite 服务器中配置的每个组织有一个分配。这样做的优点是，每个组织都维护单独的订阅，以便可以支持多个机构，每个机构都有自己的红帽帐户。

将来的日期订阅

您可以在订阅分配中使用将来的订阅。当您在现有订阅的过期日期前向内容主机添加将来的订阅时，您可以获得对仓库的不间断访问。

在当前订阅过期前，手动将将来的订阅附加到您的内容主机。不要依赖 auto-attach 方法，因为此方法是不同的目的设计的，且可能无法正常工作。如需更多信息，请参阅[第 5.6 节“将红帽订阅附加到内容主机”](#)。

5.1. 将 RED HAT SUBSCRIPTION 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。

先决条件

- 您必须有从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，导航到 **Content > Subscriptions** 并点 **Manage Manifest**。
3. 在 Manage Manifest 窗口中，单击 **Browse**。
4. 导航到包含红帽订阅清单文件的位置，然后单击 **Open**。如果 Manage Manifest 窗口没有自动关闭，点 **Close** 返回到 Subscriptions 窗口。

CLI 过程

1. 将红帽订阅清单文件从您的客户端复制到 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，[请参阅内容管理指南中的导入内容](#)。

5.2. 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的订阅。

前提条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。更多信息请参阅 [第 5.1 节“将 Red Hat Subscription 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，导航到 **Content > Subscriptions**。
3. 在 Subscriptions 窗口中，点击 **Search** 字段查看用于构建搜索查询的搜索条件列表。
4. 选择搜索条件来显示其他选项。
5. 构建搜索查询后，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。如果您选择 **& gt;** 并按空格栏，则会出现另一个自动选项列表。您还可以输入自己的标准。

5.3. 将红帽订阅添加到订阅分配中

使用以下步骤在 Satellite Web UI 中的订阅分配中添加红帽订阅。

前提条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。更多信息请参阅 [第 5.1 节“将 Red Hat Subscription 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，导航到 **Content > Subscriptions**。
3. 在 Subscriptions 窗口中，点 **Add Subscriptions**。
4. 在您要添加的每个订阅行中，在 **Quantity to assign** 列中输入数量。
5. 点 **Submit**

5.4. 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

清单不能被删除。如果您从红帽客户门户网站或 Satellite Web UI 中删除清单，则所有内容主机的权利都会被删除。

前提条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。更多信息请参阅 [第 5.1 节“将 Red Hat Subscription 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，导航到 **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。

5.5. 更新和刷新红帽订阅

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
- 购买附加订阅

您可以直接在 Satellite Web UI 中刷新清单。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，导航到 **Content > Subscriptions**。
3. 在 Subscriptions 窗口中，点 **Manage Manifest**。
4. 在 Manage Manifest 窗口中，点 **Refresh**。

5.6. 将红帽订阅附加到内容主机

使用激活码是在置备过程中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。如果您需要将新的或附加订阅（如将来的订阅）附加到一个主机，请使用以下步骤。

有关更新多个主机的详情，请参考 [第 5.7 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 10 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，因为它们不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，以便您自己的记录或跟踪目的。

前提条件

- 您必须有一个红帽订阅清单文件导入到 Satellite 服务器。

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，导航到 **Hosts > Content Hosts**。
3. 在您要更改的每个内容主机行中，选择对应的复选框。
4. 从 **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. 选中您要添加或删除的订阅左侧的复选框，并根据需要点击 **Add Selected** 或 **Remove Selected**。
7. 点 **Done** 保存更改。

CLI 过程

1. 以 root 用户身份连接到 Satellite 服务器，然后列出可用的订阅：

```
# hammer subscription list \
--organization-id 1
```

2. 将订阅附加到主机：

```
# hammer host subscription attach \
--host host_name \
--subscription-id subscription_id
```

5.7. 更新多个主机上的红帽订阅

使用这个流程同时更改多个内容主机。

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，导航到 **Hosts > Content Hosts**。
3. 在您要更改的每个内容主机行中，选择对应的复选框。

4. 从 **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. 选中要添加或删除的订阅左侧的复选框, 并根据需要点 **Add Selected** 或 **Remove Selected**。
7. 点 **Done** 保存更改。

第 6 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。如果要导入 RPM、文件或不同的内容类型到 Satellite，则会使用本章中的相同步骤进行。

例如，您可以使用以下章节来了解有关特定类型自定义内容的信息，但底层流程相同：

- [第 13 章 管理 ISO 镜像](#)
- [第 14 章 管理自定义文件类型内容](#)

6.1. 在 SATELLITE 中使用产品和软件仓库

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才能访问，类似于红帽产品的订阅。Satellite 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。该产品的软件仓库由不同的版本、架构和附加组件组成。对于红帽软件仓库，产品会在启用存储库后自动创建。如需更多信息，请参阅 [第 6.5 节“启用红帽软件仓库”](#)。

其他自定义内容可以组织到产品中。例如，您可以创建一个 EPEL (Extra Packages for Enterprise Linux) 产品并为其添加“EPEL 7 x86_64”存储库。

有关创建和打包 RPM 的详情，请参考 Red Hat Enterprise Linux 文档中的 [RPM 打包指南](#)。

6.2. 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。这包括您要同步的上游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 RPM，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，导航到 **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 证书**。
4. 在 **Content Credentials Content** 字段中，粘贴您的 SSL 证书，或者单击 **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 证书下载到您的 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
```

6.3. 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，导航到 **Content > Products**，点 **Create Product**。
2. 在 **Name** 字段中输入产品的名称。Satellite 根据您为 **Name** 输入的内容自动完成 **Label** 字段。
3. 可选：从 **GPG Key** 列表中，选择产品的 GPG 密钥。
4. 可选：从 **SSL CA Cert** 列表中，为产品选择 SSL CA 证书。
5. 可选：从 **SSL Client Cert** 列表中，选择产品的 SSL 客户端证书。
6. 可选：从 **SSL Client Key** 列表中，选择产品的 SSL 客户端密钥。
7. 可选：从 **Sync Plan** 列表中，选择现有同步计划，或者点 **Create Sync Plan** 并根据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. 点击 **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"
```

6.4. 添加自定义 RPM 存储库

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 窗口也提供了一个 **Repo Discovery** 功能，从 URL 中找到所有存储库，您可以选择要添加到自定义产品中的存储库。例如，您可以使用 **Repo Discovery** 搜索 <http://yum.postgresql.org/9.5/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

红帽不支持来自第三方站点的上游 RPM。这些 RPM 用于演示同步过程。有关这些 RPM 中的任何问题，请联系第三方开发人员。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**，再选择要使用的产品，然后单击 **New Repository**。
2. 在 **Name** 字段中输入存储库的名称。Satellite 根据您为 **Name** 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：从 **Restrict to Architecture** 列表选择一个架构。如果您要使存储库可供所有主机使用，无论架构是什么，请确保选择 **不受限制**。
6. 可选：从 **Restrict to OS Version** 列表中，选择 OS 版本。如果您要使存储库可供所有主机使用，无论操作系统版本是什么，请确保选择 **不受限制**。
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://**。如果使用 **file://** 存储库，则必须将其放在 **/var/lib/pulp/sync_imports/** 目录下。
如果没有输入上游 URL，您可以手动上传软件包。
8. 可选：选中 **Ignore SRPMs** 复选框，以将源 RPM 软件包从同步到 Satellite 中排除。
9. 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。
10. 可选：在 **Upstream Username** 字段中，输入上游存储库的用户名（如果需要）。如果存储库不需要身份验证，请清除此字段。
11. 可选：在 **Upstream Password** 字段中输入上游存储库的对应密码。如果存储库不需要身份验证，请清除此字段。
12. 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。
13. 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 6.7 节“下载策略概述”](#)。
14. 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 6.10 节“镜像策略概述”](#)。
15. 可选：在 **Retain package versions** 字段中输入您要保留的版本数量。
16. 可选：在 **HTTP Proxy Policy** 字段中，选择一个 HTTP 代理。
17. 从 **Checksum** 列表中，为存储库选择 checksum 类型。

18. 可选：您可以清除 **Unprotected** 复选框，以便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。
19. 可选：从 **GPG Key** 列表中，选择产品的 GPG 密钥。
20. 可选：在 **SSL CA Cert** 字段中，为存储库选择 SSL CA 证书。
21. 可选：在 **SSL Client cert** 字段中，为存储库选择 SSL 客户端证书。
22. 可选：在 **SSL Client Key** 字段中，为存储库选择 SSL 客户端密钥。
23. 单击 **Save** 以创建存储库。

CLI 过程

1. 运行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](#)。

6.5. 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 HTTP 代理。请参见[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存储库的产品，然后根据相关的发行版本和基础架构启用该存储库。对于 Red Hat Enterprise Linux 8，您必须同时启用 AppStream 和 BaseOS 软件仓库。

仓库版本控制

将 Red Hat Enterprise Linux 操作系统的发行版本与 7 Server 仓库或 7.X 仓库相关联的不同在于：7 Server 仓库包括所有最新的更新，Red Hat Enterprise Linux 7.X 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。

对于 Red Hat Enterprise Linux 8 客户端

要置备 Red Hat Enterprise Linux 8 客户端，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 客户端

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

流程

1. 在 Satellite Web UI 中，导航到 **Content > Red Hat Repositories**。

2. 要查找存储库，请输入存储库名称，或者将 **推荐的存储库按钮** 切换到 on 位置，以查看您需要的存储库列表。
3. 在 Available Repositories 窗格中，点存储库以展开存储库集。
4. 点基本架构和发行版本旁的 **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. 列出为产品设置的存储库：

```
# 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。包括发行版本，如 **7Server** 和基础架构，例如 **x86_64**。例如：

```
# 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"
```

6.6. 同步软件仓库

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**，再选择包含您要同步的存储库的产品。
2. 选择您要同步的存储库，然后单击 **Sync Now**。

要在 Satellite Web UI 中查看同步的进度，请导航到 **Content > Sync Status** 并展开对应的产品或存储库树。

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"
```

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。下表提供根据可用的互联网带宽来同步内容所需的时间：

	单一软件包(10Mb)	次发行版本(750Mb)	主发行版本(6Gb)
256 Kbps	5 分钟 27 Sec	6 hrs 49 Mins 36 Secs	2 天 7 Hrs 55 Mins
512 Kbps	2 分钟 43.84 Sec	3 hrs 24 Mins 48 Secs	1 天 3 天 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 Sec	1 HR 25 Mins 53.96 Secs
100 Mbps	0.84 secs	1 min 2.91 Secs	8 最小 35.4 Secs
1000 Mbps	0.08 secs	6.29 secs	51.54 secs

创建同步计划以确保定期进行更新。请参阅 [第 6.18 节“创建同步计划”](#)。

6.6.1. 同步机构中的所有存储库

使用这个流程同步机构中的所有软件仓库。

流程

要同步机构中的所有存储库，请在 Satellite 服务器上运行以下 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
```

6.7. 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，您可能只想下载内容元数据，同时延迟以后为实际内容下载。

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。

on Demand

Satellite 服务器仅在同步期间下载元数据。Satellite 服务器仅在 Capsule 或直接连接的客户端请求它们时获取并存储文件系统上的软件包。如果您将 Capsule 上的相应存储库设置为 **Immediate**，则此设置无效，因为 Satellite 服务器被强制下载所有软件包。

On Demand 策略充当 Lazy 同步功能，因为它们可节省时间内容。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阶段。

Capsule 服务器有以下策略：

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。

on Demand

Capsule 服务器仅在同步期间下载元数据。胶囊式服务器仅在直接连接的客户端请求它们时，仅在文件系统中获取并存储软件包。当您使用 **On Demand** 下载策略时，如果 Capsule 服务器上不可用，内容将从 Satellite 服务器下载。

继承

Capsule 服务器从 Satellite 服务器上的对应存储库继承存储库的下载策略。

6.8. 更改默认下载策略

您可以设置 Satellite 适用于您在所有机构中创建的存储库的默认下载策略。

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的设置。

流程

1. 在 Satellite Web UI 中，导航到 **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

- 要将红帽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_redhat_download_policy \
--value immediate
```

- 要将非红帽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
```

6.9. 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，导航到 **Content > Products**。
2. 选择所需的产品名称。
3. 在 **Repositories** 选项卡中，点所需的存储库名称，找到 **Download Policy** 字段，然后点编辑图标。
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更改为 **immediate** 或 **on_demand**：

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

6.10. 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移除任何内容，则也会从本地存储库中移除。

您可以在同步存储库时，使用镜像策略对 `repopdata` 和内容的镜像进行精细控制。例如，如果无法镜像仓库的 `repopdata`，您可以将镜像策略设置为仅镜像此存储库的内容。

Satellite 服务器有以下镜像策略：

additive

内容和 `repopdata` 都不会被镜像(mirror)。因此，只有添加的新内容才会添加到本地存储库中，并且不会删除任何同步。

仅限内容

仅镜像内容而不是 `repopdata`。有些存储库不支持元数据镜像，在这种情况下，您只能将镜像策略设置为只镜像内容。

完成镜像

镜像内容以及 `repopdata`。这是最快的方法。此镜像策略仅适用于 Yum 内容。

6.11. 更改存储库的镜像策略

您可以为存储库设置镜像策略。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**。
2. 选择产品名称。

3. 在 **Repositories** 选项卡中，点存储库名称，找到 **Mirroring Policy** 字段，然后点 **edit** 图标。
4. 从列表中选择镜像策略并点击 **Save**。

6.12. 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，点 **Content > Products**。
2. 点自定义产品的名称。
3. 在 **Repositories** 选项卡中，点自定义 RPM 存储库的名称。
4. 在 **Upload Package** 下，点 **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，请单击 **Content Counts** 下的 **Packages** 旁边的数字。

CLI 过程

- 输入以下命令上传 RPM：

```
# hammer repository upload-content \
--id 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 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** 命令查看源 RPM 的信息。

6.13. 配置 SELINUX 以在自定义端口中允许内容同步

SELinux 仅允许对 Satellite 的访问，仅在特定端口上进行内容同步。默认情况下，允许连接到运行在以下端口上的 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，请按如下所示输入命令：

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

2. 要将 SELinux 配置为允许端口进行内容同步，如 10011，请输入以下命令：

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

6.14. 恢复修正的存储库

如果存储库崩溃，您可以使用高级同步来恢复它，其有三个选项：

优化同步

同步存储库会绕过没有检测到上游 RPM 的不同的 RPM。

完成同步

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

验证内容检查

同步所有 RPM，然后在本地验证所有 RPM 的校验和。如果 RPM 的校验和与上游不同，它会重新下载 RPM。这个选项只适用于 **yum** 软件仓库。如果您有以下错误之一，请使用这个选项：

- 在与 **yum** 同步时，特定的 RPM 会导致一个 **404** 错误。
- **package** 与预期的下载错误不匹配，这意味着特定的 RPM 已损坏。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**。
2. 选择包含已损坏存储库的产品。
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. 使用必要的选项同步损坏的存储库：

- 对于优化的同步：

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

- 进行完整的同步：

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

- 对于验证内容同步：

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

6.15. 添加 HTTP 代理

使用这个流程将 HTTP 代理添加到 Satellite。然后，您可以指定用于产品、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，那么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，导航到 **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. 单击 **Locations** 选项卡，再添加一个位置。

7. 点 **Organization** 选项卡并添加机构。
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 name` 和 `--password password` 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 \(RHSM\)](#)。

6.16. 更改一个产品的 HTTP 代理策略

要精细控制网络流量，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，除非您为单个软件仓库设置了不同的策略。

要为单个软件仓库设置 HTTP 代理策略，请参阅 [第 6.17 节“更改一个存储库的 HTTP 代理策略”](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**，再选中您要更改的每个产品旁边的复选框。
2. 从 **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**：请勿使用 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 6.15 节“添加 HTTP 代理”](#)。
4. 点 **Update**。

6.17. 更改一个存储库的 HTTP 代理策略

要精细控制网络流量，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 6.16 节“更改一个产品的 HTTP 代理策略”](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**，然后点击包含该存储库的产品名称。
2. 在 **Repositories** 选项卡中，点存储库的名称。

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** : 请勿使用 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 6.15 节“添加 HTTP 代理”](#)。
5. 点击 **Save**。

CLI 过程

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

```
# hammer repository update --id repository-ID \
--http-proxy-policy policy
```

为 **--http-proxy-policy** 指定以下选项之一 :

- **none** : 请勿使用 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**: 指定一个 HTTP 代理, 使用 **--http-proxy proxy-name** 或 **--http-proxy-id proxy-ID**。要在 Satellite 中添加新 HTTP 代理, 请参阅 [第 6.15 节“添加 HTTP 代理”](#)。

6.18. 创建同步计划

同步计划在调度的日期和时间检查并更新内容。在 Satellite 中, 您可以创建一个同步计划, 并为计划分配产品。

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

流程

1. 在 Satellite Web UI 中, 导航到 **Content > Sync Plans**, 再点 **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, 选择您要运行计划的时间间隔。
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 \
```

```
--name "Red Hat Products 2" \  
--description "Example Plan for Red Hat Products" \  
--interval daily \  
--sync-date "2016-02-01 01:00:00" \  
--enabled true \  
--organization "My_Organization"
```

2. 查看某个机构的可用同步计划，以验证是否已创建了同步计划：

```
# hammer sync-plan list --organization "Default Organization"
```

6.19. 将同步计划分配给产品

同步计划在调度的日期和时间检查并更新内容。在 Satellite 中，您可以为产品分配一个同步计划，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，导航到 **Content > Products**。
2. 选择一个产品。
3. 在 **Details** 选项卡中，从下拉菜单中选择 **Sync Plan**。

CLI 过程

1. 为产品分配同步计划：

```
# hammer product set-sync-plan \  
--name "Product_Name" \  
--sync-plan "Sync_Plan_Name" \  
--organization "My_Organization"
```

6.20. 为多个产品分配同步计划

使用这个流程为机构中的产品分配至少同步一次并至少包含一个存储库的同步计划。

流程

1. 运行以下 Bash 脚本：

```
ORG="Your_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，查看分配给同步计划的产品：

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

6.21. 限制同步 CONCURRENCY

默认情况下，每个仓库同步作业每次都可以获取最多十个文件。这可基于每个存储库进行调整。

增加限制可能会提高性能，但可能导致上游服务器过载或开始拒绝请求。如果您因为上游服务器拒绝请求而看到 Repository 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"
```

6.22. 导入自定义 GPG 密钥

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

红帽内容已使用适当的 GPG 密钥进行配置，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 密钥副本，用于为要在 Satellite 中使用和管理的 RPM 内容签名。大多数 RPM 发行版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客户端系统中：

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

2. 在不安装它的情况下提取 RPM 文件：

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

GPG 密钥相对于位于 `etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE-95` 的提取。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Content Credentials**，并在窗口右上角点击 **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，或者点击 **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多个 GPG 密钥签名的内容，您必须在 **Content Credential Contents** 字段中输入所有所需的 GPG 密钥，以及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
WI5GnzcLGAAnUSRamfqGUWcyMMinHHIKlc1X1P4I=
=WPpl
-----END PGP PUBLIC KEY BLOCK-----

```

4. 点击 **Save**。

CLI 过程

1. 将 GPG 密钥复制到您的 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
```

6.23. 在 SATELLITE 中将自定义软件仓库限制为 RHEL 9

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**。
2. 选择包含您要限制的存储库集合的产品。
3. 在 **Repositories** 选项卡中，点击您要限制的存储库。
4. 从 **Restrict to OS** 列表中，选择包含 Red Hat Enterprise Linux 操作系统及其对应的 Red Hat Enterprise Linux 软件仓库的操作系统。
5. 点 **Save** 提交您的更改。

第 7 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。

7.1. 应用程序生命周期介绍

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比较简单；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。这为应用程序生命周期添加了额外的阶段：

- 开发
- 测试
- Beta 版本
- Production

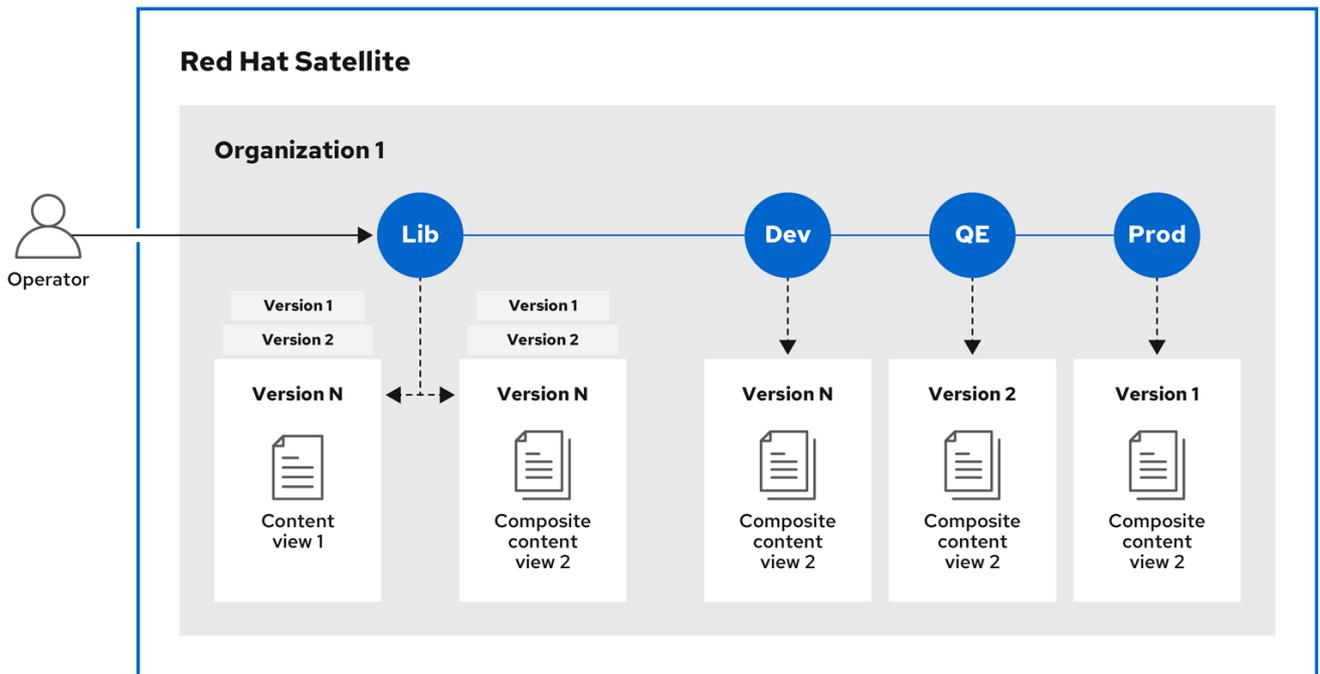
Satellite 提供了自定义每个应用程序生命周期阶段的方法，使其适合您的规格。

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。这为您提供了定义为每个环境指定的特定内容集合的方法。

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，其中有用于真实用途的生产级服务器以及测试最新的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的软件，请在质量保证环境中对其进行测试，预发布为 beta，然后将软件作为生产级应用程序发布。

图 7.1. Satellite 应用程序生命周期



207_Satellite_0222

7.2. 在应用程序生命周期间提升内容

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 promote。

Satellite 生命周期阶段的内容提升示例

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 will 将软件包从一个环境提升到下一个环境时，目标环境的软件仓库会收到新的软件包版本。因此，目标环境中的每个系统都可以更新到新的软件包版本。

开发	测试	Production
example_software-1.1-0.noarch.rpm	example_software-1.0-0.noarch.rpm	example_software-1.0-0.noarch.rpm

在补丁上完成开发后，您可以将软件包提升到测试环境，以便质量工程团队可以查看补丁。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
example_software-1.1-0.noarch.rpm	example_software-1.1-0.noarch.rpm	example_software-1.0-0.noarch.rpm

虽然服务质量团队审查补丁，但开发团队在 example_software 2.0 上开始工作。这会生成以下应用程序生命周期：

开发	测试	Production
exampleware-2.0-0.noarch.rpm	exampleware-1.1-0.noarch.rpm	exampleware-1.0-0.noarch.rpm

质量工程团队完成他们对补丁的审查。现在，example_software 1.1 已准备好发布。您可以将 1.1 提升到生产环境：

开发	测试	Production
example_software-2.0-0.noarch.rpm	example_software-1.1-0.noarch.rpm	example_software-1.1-0.noarch.rpm

开发团队在 example_software 2.0 上完成自己工作，并将其提升到测试环境：

开发	测试	Production
example_software-2.0-0.noarch.rpm	example_software-2.0-0.noarch.rpm	example_software-1.1-0.noarch.rpm

最后，质量工程团队将检查软件包。成功检查后，将软件包提升到生产环境：

开发	测试	Production
exampleware-2.0-0.noarch.rpm	exampleware-2.0-0.noarch.rpm	exampleware-2.0-0.noarch.rpm

如需更多信息，请参阅 [第 8.3 节“提升内容视图”](#)。

7.3. 创建生命周期阶段路径

要为开发和发布软件创建应用程序生命周期，请使用库环境作为初始环境来创建环境路径。然后，选择性地环境路径中添加额外的环境。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Lifecycle Environments**。
2. 单击 **New Environment Path** 以启动新应用程序生命周期。
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，请点击 **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. 要查看生命周期阶段的链，请输入以下命令：

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

7.4. 从 SATELLITE 服务器中删除执行环境

使用这个流程删除生命周期阶段。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Life Cycle Environments**。
2. 单击您要删除的生命周期阶段的名称，然后单击 **Remove Environment**。
3. 单击 **Remove** 以删除环境。

CLI 过程

1. 列出您的机构生命周期阶段，并记录您要删除的生命周期阶段的名称：

```
# 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"
```

7.5. 从 CAPSULE 服务器中删除执行环境

当生命周期阶段不再与主机系统或环境错误添加到 Capsule 服务器时，您可以从 Capsule 服务器中删除执行环境。

您可以使用 Satellite Web UI 和 hammer CLI 从 Capsule 中删除生命周期阶段。

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**，然后选择您要从其中删除生命周期的 Capsule。
2. 点 **Edit** 并点 **Life Cycle Environments** 选项卡。
3. 从右侧菜单中选择您要从 Capsule 中删除的生命周期阶段，然后单击 **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. 从列表中选择您想要的 Capsule 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 Capsule_ID
```

3. 验证当前附加到 Capsule 服务器的生命周期阶段列表，并记下 **环境 ID**：

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

4. 从 Capsule 服务器中删除生命周期阶段：

```
# hammer capsule content remove-lifecycle-environment \
--id Capsule_ID
--lifecycle-environment-id Lifecycle_Environment_ID
```

对您要从 Capsule 服务器中删除的每个生命周期阶段重复此步骤。

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

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

7.6. 将虚拟环境添加到 CAPSULE 服务器

如果您的 Capsule 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。

不要将库虚拟化环境分配给您的 Capsule 服务器，因为它在每次 CDN 更新存储库时触发自动 Capsule 同步。这可能会在 Capsule 上消耗多个系统资源、Satellite 和 Capsule 之间的网络带宽，以及胶囊上的可用磁盘空间。

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，导航到 **Infrastructure > Capsules**，然后选择您要向其添加生命周期的 Capsule。
2. 点 **Edit** 并点 **Life Cycle Environments** 选项卡。
3. 在左侧菜单中选择您要添加到 Capsule 的生命周期阶段，然后单击 **Submit**。
4. 若要同步 Capsule 上的内容，请单击 **Overview** 选项卡，然后单击 **同步**。
5. 选择 **Optimized Sync** 或 **Complete Sync**。
有关每种同步类型的定义，请参阅[恢复仓库](#)。

CLI 过程

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

```
# hammer capsule list
```

请注意您要向其添加生命周期的 Capsule 的 Capsule ID。

2. 使用 ID，验证您的 Capsule 的详情：

```
# hammer capsule info --id capsule_id
```

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

```
# hammer capsule content available-lifecycle-environments --id capsule_id
```

4. 将生命周期阶段添加到您的 Capsule 服务器中：

```
# hammer capsule content add-lifecycle-environment \  
--id capsule_id --organization "My_Organization" \  
--lifecycle-environment-id lifecycle-environment_id
```

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. 将内容从 Satellite 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 capsule_id
```

- 要将特定生命周期环境从 Satellite 服务器同步到 Capsule 服务器，请输入以下命令：

```
# hammer capsule content synchronize --id external_capsule_id \  
--lifecycle-environment-id lifecycle-environment_id
```

第 8 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。这些过滤器包括软件包过滤器、软件包组过滤器、勘误过滤器、模块流过滤器和容器镜像标签过滤器。您可以使用内容视图来定义特定环境使用的软件版本。例如，生产环境可能会使用包含较老的软件包版本的内容视图，而开发环境可能会使用包含较新的软件包版本的内容视图。

或者，默认机构视图是与 Satellite 同步的所有内容的应用控制内容视图。如果要将主机注册到 Satellite 并使用订阅访问内容而无需操作内容视图和生命周期，则此类型很有用。

每个内容视图在每个环境中创建一组软件仓库，Satellite 服务器存储和管理它们。当您将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 服务器上的相应存储库会更新并发布软件包。

	开发	测试	Production
内容视图版本和内容	版本 2 - example_software-1.1-0.noarch.rpm	版本 1 - example_software-1.0-0.noarch.rpm	版本 1 - example_software-1.0-0.noarch.rpm

测试和生产的存储库包含 **example_software-1.0-0.noarch.rpm** 软件包。如果您将内容视图的版本 2 从 Development 提升到测试，则测试将重新生成的存储库，然后包含 **example_software-1.1-0.noarch.rpm** 软件包：

	开发	测试	Production
内容视图版本和内容	版本 2 - example_software-1.1-0.noarch.rpm	版本 2 - example_software-1.1-0.noarch.rpm	版本 1 - example_software-1.0-0.noarch.rpm

这样可确保系统指定为特定环境，但在该环境使用新版本的内容视图时接收更新。

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. 为内容视图添加一个或多个您需要的仓库。
3. 可选：创建一个或多个过滤器来重新定义内容视图的内容。更多信息请参阅 [第 8.10 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。更多信息请参阅 [第 8.8 节“解决软件包依赖项”](#)。
5. 发布内容视图。
6. 可选：将内容视图复制到另一个环境。更多信息请参阅 [第 8.3 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` 仍为空，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。更多信息请参阅 [第 8.6 节“创建 Composite 内容视图”](#)。

8.1. 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

虽然您可以停止是否想通过内容视图解析任何软件包依赖项，但您可能希望更改默认 Satellite 设置以为所有内容视图启用或禁用软件包解析。更多信息请参阅 [第 8.8 节“解决软件包依赖项”](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Content Views**，再点 **Create content view**。
2. 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。
3. 在 **Description** 字段中，输入视图的描述。
4. 在 **Type** 字段中，选择 **Content view** 或 **Composite 内容视图**。
5. 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。依赖项会减慢发布时间，并可能会忽略您使用的任何内容视图过滤器。这也可以在解析勘误依赖项时导致错误。
6. 可选：如果要指定从上游服务器导入此内容视图，请选择 **Import only** 复选框。无法直接发布仅导入的内容视图。
7. 点 **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，从您要添加到内容视图的 **Type** 列表中选择存储库，选择您要添加的可用存储库旁边的复选框，然后单击 **Add repositories**。
3. 单击 **Publish new version**，然后在 **Description** 字段中输入有关日志更改的版本的版本信息。
4. 可选：您可以通过点 **Promote** 从可用的提升路径中选择执行环境来提升新版本。
5. 单击 **Next**。
6. 在 **Review** 页面中，您可以查看您要发布的环境。
7. 点 **Finish**。



注意

删除 和 **删除** 类似，但 **Delete** 选项会删除整个内容视图以及与该生命周期阶段关联的版本。**Remove** 选项允许您选择要从生命周期阶段中删除哪个版本。

您可以在内容视图窗口中查看内容视图。要查看有关内容视图的更多信息，请单击内容视图名称。要将一个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存储库：

```
# 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：要将存储库添加到现有的内容视图中，请输入以下命令：

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

Satellite 服务器会创建新版本的视图，并将其发布到库环境。

8.2. 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。

流程

1. 在 Satellite Web UI 中，导航到公布的内容视图 **> Module Streams** 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**。

8.3. 提升内容视图

在不同生命周期阶段间提升内容视图。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哪些内容视图。

`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，您可以限制用户，以便允许他们提升到测试环境，但不能提升到生产环境。

您必须为用户分配这两个权限，以允许他们提升内容视图。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Content Views**，再选择您要提升的内容视图。
2. 选择您要提升的版本，点垂直 ellipsis 图标，然后点 **Promote**。
3. 选择您要提升内容视图的环境，然后点 **Promote**。

现在，内容视图的存储库出现在所有环境中。

CLI 过程

- 每次使用 `hammer` 内容查看版本提升内容视图：

```
# 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，所有环境中都提供了数据库内容。

要将一个主机注册到您的内容视图，请参阅管理主机指南中的[注册主机](#)。

8.4. 在机构中推广跨所有生命周期环境的内容视图

使用这个流程在组织中的所有生命周期环境中提升内容视图。

流程

1. 要在机构中所有生命周期环境中从库提升所选内容视图版本，请运行以下 Bash 脚本：

```

ORG="My_Organization"
CVV_ID=3

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

```

2. 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中：

```
# hammer content-view version info --id 3
```

8.5. 复合内容视图概述

Composite Content View 组合了多个内容视图中的内容。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 Composite Content View 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
操作系统

四个独立内容视图示例：

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

在前面的内容视图中，您可以创建两个 Composite Content Views。

PostgreSQL 数据库的 Composite Content View 示例：

复合内容视图 1 - PostgreSQL 上的 example_software
example_software (Application)
PostgreSQL (Database)

复合内容视图 1 - PostgreSQL 上的 example_software

Red Hat Enterprise Linux (操作系统)

*MariaDB 的 Composite Content View 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，每个内容视图都会单独管理和发布。当您创建应用版本时，您将发布 Composite 内容视图的新版本。您也可以在创建 Composite Content View 时选择 **Auto Publish** 选项，然后在包含内容视图包含的内容视图时自动重新发布 Composite Content View。

仓库限制

Docker 存储库不能包含在 Composite Content View 中多次。例如，如果您试图在 Composite Content View 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。

8.6. 创建 COMPOSITE 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，导航到 **Content > Content Views**，再点 **Create content view**。
2. 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。
3. 可选：在 **Description** 字段中输入视图描述。
4. 在 **Type** 选项卡上，选择 **Composite 内容视图**。
5. 可选：如果您要在重新发布内容视图时自动发布 Composite Content View 的新版本，请选择 **Auto publish** 复选框。
6. 点 **Create content view**。
7. 在 **内容视图** 选项卡上，选择要添加到 Composite Content View 的内容视图，然后单击 **Add content view**。
8. 在 **Add content view** 窗口中，选择每个内容视图的版本。
9. 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。
10. 点 **Add**，然后点 **Publish new version**。

11. 可选：在 **Description** 字段中，输入内容视图的描述。
12. 在 **Publish** 窗口中，设置 **Promote** 开关，然后选择 生命周期阶段。
13. 单击 **Next**，然后单击 **Finish**。

CLI 过程

1. 在创建 Composite 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 Composite 内容视图。当 **--auto-publish** 选项设置为 **yes** 时，当内容视图包含它时，composite Content View 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. 向复合内容视图添加内容视图。您可以通过其 ID 或名称来识别命令中的内容视图、内容视图版本和机构。要在 Composite Content View 中添加多个内容视图，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 Composite 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 Composite 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"

```

8.7. 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 RPM 内容。如果没有这些过滤器，内容视图包含所选存储库的所有内容。

内容过滤器有两种类型：

表 8.1. 过滤类型

过滤类型	描述
Include	您以任何内容开始，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。
exclude	您从所选存储库中的所有内容开始，然后选择要删除的内容。当您想使用大多数特定的内容存储库，但排除某些软件包（如列入黑名单的软件包）时，请使用这个过滤器。过滤器使用存储库中的所有内容，但您选择的内容除外。

包括和排除过滤器组合

如果使用 Include 和 Exclude 过滤器的组合，则发布内容视图会首先触发 include 过滤器，然后排除 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哪些内容。

内容类型

另外，还有五类内容来过滤：

表 8.2. 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。
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。

内容类型	描述
勘误 (按 ID)	选择要添加到过滤器中的特定勘误。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的日期范围和勘误类型 (漏洞修复、增强或安全性)。
模块流	选择是否包含或排除特定模块流。 Module Streams 选项过滤模块 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容器镜像标签。

8.8. 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的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, 依赖项解决可能会导致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。

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 会在内容视图中针对依赖项检查每个存储库。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。

8.9. 为内容视图启用依赖解析

使用这个流程启用对内容视图的依赖项。

前提条件

- 依赖项需要仅在有限的上下文中有用。在启用它前，请确定您读和了解 [第 8.8 节“解决软件包依赖项”](#)

流程

1. 在 Satellite Web UI 中，导航到 **Content > 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**。

8.10. 内容过滤器示例

使用以下任何一个示例，以及构建自定义内容过滤器的步骤。



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 存储库。

filter:

- **包含类型**：包括
- **内容类型**：软件包组
- **filter**：只选择 **Base** 软件包组

示例 2

在特定日期后，创建不包括所有勘误的软件仓库（除安全更新除外）。如果您要定期执行系统更新，但必须立即应用的关键安全更新外，这非常有用。这个过滤器需要添加到内容视图中的 Red Hat Enterprise Linux 存储库。

filter:

- **包含类型**：Exclude
- **内容类型**：Erratum（日期和类型）
- **过滤器**：仅选择 **bugfix** 和 **Enhancement** 勘误类型，并清除 **安全勘误** 类型。将 **Date 类型** 设置为 **Updated On**。将 **Start Date** 设置为您要限制勘误的日期。将 **End Date** 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，其中您只需要操作系统软件包，并希望排除最新的bug 修复和增强勘误。这需要两个过滤器附加到同一个内容视图。内容视图首先处理 Include 过滤器，然后是 Exclude 过滤器。

过滤1：

- **包含类型：** 包括
- **内容类型：** 软件包组
- **filter：** 只选择 Base 软件包组

过滤2：

- **包含类型：** Exclude
- **内容类型：** Erratum（日期和类型）
- **过滤器：** 仅选择 bugfix 和 Enhancement 勘误类型，并清除 安全勘误 类型。将 Date 类型设置为 Updated On。将 Start Date 设置为您要限制勘误的日期。将 End Date 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的模块流。

过滤1：

- **包含类型：** 包括
- **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，然后单击 Add Module Stream。

过滤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。如果您不过滤软件包，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，请参阅以下文章：["内容过滤器在 Satellite 6 中的工作方式"](#)。

8.11. 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软件包、软件包组、勘误或模块流。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 8.10 节 "内容过滤器示例"](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Content Views** 并选择 Content View。

2. 在 **Filters** 选项卡中，点 **Create filter**。
3. 输入名称。
4. 从 **Content type** 列表中，选择内容类型。
5. 从 **Inclusion Type** 列表中选择 **Include filter** 或 **Exclude filter**。
6. 可选：在 **Description** 字段中输入过滤器的描述。
7. 点 **Create filter** 创建内容过滤器。
8. 根据您为 **Content Type** 输入的内容，添加规则来创建您想要的过滤器。
9. 如果您希望过滤器应用到 **存储库的子集**，或者 **应用到所有存储库**。
10. 单击 **Publish New Version** 以发布过滤的存储库。
11. 可选：在 **Description** 字段中，输入更改的描述。
12. 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. 在内容视图中添加过滤器。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"
```

8.12. 删除内容视图版本

使用这个流程删除内容视图版本。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Content Views**。
2. 选择内容视图。
3. 在 **Versions** 选项卡中，选择您要删除的版本，并点击 version 行右侧的垂直 ellipsis。
4. 点 **Delete** 打开显示任何受影响的环境的删除向导。
5. 可选：如果存在受影响的环境，请在删除前重新分配任何主机或激活码。
6. 点 **Next** 并查看操作的详情。
7. 点 **Delete**。

第 9 章 在 SATELLITE 服务器之间同步内容

在具有多个 Satellite 服务器的 Satellite 设置中，您可以使用 Satellite 同步(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。

根据您如何部署基础架构，可以使用两个 Satellite 的 ISS 配置。根据您的用例，为您的 ISS 配置 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 同步](#)。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"](#)。

9.1. 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：

- 您可以将上游 Satellite 服务器用作内容存储，这意味着您同步整个库，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。更多信息请参阅[第 9.1.1 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。更多信息请参阅[第 9.1.2 节“使用上游 Satellite 服务器同步内容视图版本”](#)。
- 您同步单个存储库。如果您使用 Content-View 同步方法，但这非常有用，但您想要同步额外的存储库，而无需将其添加到现有的内容视图中。更多信息请参阅[第 9.1.3 节“同步单个存储库”](#)。

9.1.1. 使用上游 Satellite 服务器作为内容存储

在这种情况下，您可以使用上游 Satellite 服务器作为更新的内容存储，而不是管理内容。您可以使用下游 Satellite 服务器管理隔离网络后面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，并将其导入到下游 Satellite 服务器中。

在上游 Satellite 服务器上

1. 确存储库使用以下方法之一使用 **Immediate** 下载策略：
 - a. 对于使用 **On Demand** 的现有软件仓库，请将存储库详细信息页面上的下载策略更改为 **Immediate**。
 - b. 对于新的软件仓库，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**，并且为自定义存储库将 **Default** 下载策略设置为 **Immediate**。

更多信息请参阅[第 6.7 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 6.5 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. 在第一次导出时，**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，稍后可以导入到一个或多个下游 Satellite 服务器中。有关执行完整的库导出的详情，请参考[第 9.3 节“导出库环境”](#)。
 - b. 以递增方式导出上游 Satellite 服务器上的所有更新。这会生成仅包含最新更新集的普通内容存档。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新启用的存储库中的内容。有关执行增量库导出的详情，请参考[第 9.4 节“以递增方式导出库环境”](#)。

在下游 Satellite 服务器上

1. 将从上游 Satellite 服务器导出的内容移动到硬盘中。
2. 将它放在 `/var/lib/pulp/imports` 下的目录中。
3. 使用 [第 9.10 节“导入到库环境”](#) 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执行环境来管理内容。

9.1.2. 使用上游 Satellite 服务器同步内容视图版本

在这种情况下，您将上游 Satellite 服务器用作内容存储，还同步隔离网络后面的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阶段后，您可以从上游 Satellite 服务器导出内容并将其导入到下游 Satellite 服务器中。

在上游 Satellite 服务器上

1. 确保存储库使用以下方法之一使用 **Immediate** 下载策略：
 - a. 对于使用 **On Demand** 的现有软件仓库，请将存储库详细信息页面上的下载策略更改为 **Immediate**。
 - b. 对于新的软件仓库，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**，并且为自定义存储库将 **Default** 下载策略设置为 **Immediate**。

更多信息请参阅 [第 6.7 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 6.5 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 9.5 节“导出内容视图版本”](#)。这会生成可导入到一个或多个下游 Satellite 服务器的内容存档。
 - b. 以连接 Satellite 服务器逐步导出所有将来的更新。这会生成简洁的内容存档，该存档仅包含最新更新集合的更改。例如，如果您的内容视图具有新的存储库，则此导出的内容存档仅包含最新的更改。更多信息请参阅 [第 9.6 节“逐渐导出内容视图版本”](#)。
 - c. 当您有新内容时，在导出递增前重新发布包含此内容的内容视图。更多信息请参阅 [第 8 章管理内容视图](#)。这会创建一个新的内容视图版本，其中包含要导出的适当内容。

在下游 Satellite 服务器上

1. 将从上游 Satellite 服务器导出的内容移动到硬盘中。
2. 将它放在 `/var/lib/pulp/imports` 下的目录中。
3. 将内容导入到您想要的组织。更多信息请参阅 [第 9.11 节“导入内容视图版本”](#)。这将从导出的内容存档创建内容视图版本，然后正确导入内容。

9.1.3. 同步单个存储库

在这种情况下，您将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** 的现有软件仓库，请将存储库详细信息页面上的下载策略更改为 **Immediate**。
 - b. 对于新的软件仓库，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**，并且为自定义存储库将 **Default** 下载策略设置为 **Immediate**。

更多信息请参阅 [第 6.7 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 6.5 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. 在第一次导出时，**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，稍后可以导入到一个或多个下游 Satellite 服务器中。有关执行完整存储库导出的详情，请参考 [第 9.7 节“导出存储库”](#)。
 - b. 以递增方式导出上游 Satellite 服务器上的所有更新。这会生成仅包含最新更新集的普通内容存档。有关执行增量存储库导出的详情，请参考 [第 9.8 节“永久导出存储库”](#)。

在下游 Satellite 服务器上

1. 将从上游 Satellite 服务器导出的内容移动到硬盘中。
2. 将它放在 `/var/lib/pulp/imports` 下的目录中。
3. 将内容导入到机构。请参阅 [第 9.12 节“导入仓库”](#)。然后，您可以根据需要使用内容视图或执行环境来管理内容。

9.2. 同步自定义存储库

在使用 Satellite 同步网络同步时，会自动配置红帽存储库，但不会自定义存储库。使用这个流程，通过 Satellite 同步(ISS)网络同步，将连接的 Satellite 服务器上的自定义存储库的内容同步到断开连接的 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，导航到 **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 6.2 节“导入自定义 SSL 证书”。
3. 在 Satellite Web UI 中，导航到 **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 认证**：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 6.3 节“创建自定义产品”。

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

9.3. 导出库环境

您可以将组织库环境中所有 Yum 存储库的内容从 Satellite 服务器导出到存档文件，并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。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
- 包含来自组织库环境中所有存储库的存档文件。

Satellite 服务器仅导出内容视图版本中包含的 RPM 和 kickstart 文件。Satellite 不导出以下内容：

- Docker 内容

先决条件

要导出组织的库生命周期阶段的内容，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的存储空间来容纳导出。
- 确保 **/var/lib/pulp/exports** 目录具有与为导出过程中创建的临时文件导出的存储库大小相同的空闲空间。
- 确保为您导出的库生命周期阶段中的所有存储库将下载策略设置为 **Immediate**。更多信息请参阅第 6.7 节“下载策略概述”。
- 确保您同步了导出至所需日期的产品。

导出机构的库内容

1. 使用要导出的机构名称或 ID。

```
# 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是否位于导出目录中：

```
# 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** 和 **metadata.json** 文件才能导入。

3. 在机构中创建新的内容视图 **Export-Library**。此内容视图包含属于此机构的所有存储库。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，导出的存档内容可能会以几个 GB 为单位。如果要将其分成较小的大小或块。您可以在 `export` 命令中使用 **--chunk-size-gb** 标志来处理此功能。在以下示例中，您可以看到如何指定 **--chunk-size-gb=2** 将存档分成 **2 GB** 块。

```
# 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/
```

9.4. 以递增方式导出库环境

在系统资源方面，导出库内容可能非常昂贵的操作。具有多个 Red Hat Enterprise Linux 树的组织可以占据 Satellite 服务器上几 GB 空间。

在这种情况下，您可以使用 **Incremental Export** 只导出自之前导出以来更改的部分内容。增量导出通常会导致存档文件比完整导出要小。

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"

Generated /var/lib/pulp/exports/My_Organization/Export-Library/3.0/2021-03-02T04-22-14-00-00/metadata.json
```

2. 可选：查看导出的数据：

```
# ls -lh /var/lib/pulp/exports/My_Organization/Export-Library/3.0/2021-03-02T04-22-14-00-00/
total 172K
-rw-r--r--. 1 pulp pulp 161K 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 pulp pulp 492 Mar  2 04:22 metadata.json
```

9.5. 导出内容视图版本

您可以将内容视图的版本导出到 Satellite 服务器的存档文件中，并使用此存档文件在另一个 Satellite 服务器上或另一个 Satellite 服务器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

Satellite 服务器只导出添加到内容视图版本的 RPM 和 kickstart 文件。Satellite 不导出以下内容：

- Docker 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的存储空间来容纳导出。
- 确保 `/var/lib/pulp/exports` 目录具有与为导出过程中创建的临时文件导出的存储库大小相同的空闲空间。
- 确保为您导出的内容视图中的所有存储库将下载策略设置为 **Immediate**。更多信息请参阅 [第 6.7 节“下载策略概述”](#)。
- 确保您同步了导出至所需日期的产品。
- 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的版本：

```
# 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是否位于导出目录中：

```
# 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，即 **toc.json** 和 **metadata.json**，才能成功导入内容。

使用块导出

在很多情况下，导出的存档内容的大小可以是几个 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** 选项来处理此功能。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/
```

9.6. 逐渐导出内容视图版本

在系统资源方面，导出完整版本可能非常昂贵。具有多个 Red Hat Enterprise Linux 树的内容视图版本可以占据 Satellite 服务器上几 GB 空间。

在这种情况下，您可以使用 **Incremental Export** 只导出自之前导出以来更改的部分内容。增量导出通常会导致存档文件比完整导出要小。

在以下示例中，版本 **2.0** 是导出的目标，因为版本 **1.0** 在以前已被导出。

流程

1. 创建增量导出：

```
# hammer content-export incremental version \
--content-view="Content_View_Name" \
--organization="My_Organization" \
--version=2.0
```

2. 可选：查看导出的内容视图：

```
# ls -lh /var/lib/pulp/exports/My_Organization/view/2.0/2021-02-25T21-45-34-00-00/
```

9.7. 导出存储库

您可以从 Satellite 服务器在机构的库环境中导出存储库的内容。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：

- RPM 软件仓库
- Kickstart 软件仓库

- Ansible 存储库
- 文件仓库

您无法从 Satellite 服务器导出 Docker 内容。

导出包含以下数据：

- 包含存储库元数据的两个 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

要导出存储库的内容，请确保要从中导出的 Satellite 服务器，满足以下条件：

- 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，相当于为导出过程中创建的临时文件导出库大小。
- 确保您导出的库生命周期阶段中将下载策略设置为 **Immediate**。如需更多信息，请参阅 [第 6.7 节“下载策略概述”](#)。
- 确保您同步了导出至所需日期的产品。

导出存储库

1. 使用要导出的存储库名称或 ID。

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

2. 可选：验证导出的归档是否位于导出目录中：

```
# ls -lh /var/lib/pulp/exports/My_Organization/Export-My_Repository/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 root root 443 Mar  2 03:35 metadata.json
```

使用 Chunking 导出存储库

在很多情况下，导出的内容存档大小可能有多个 GB。如果要将其分成较小的大小的块，您可以在 `export` 命令中使用 `--chunk-size-gb` 参数，并使用整数值以 GB 为单位限制大小。

1. 将内容导出到有限大小的归档块，如 2 GB：

```
# hammer content-export complete repository \
```

```
--chunk-size-gb=2 \
--organization="My_Organization" \
--product="My_Product" \
--name="My_Repository"
```

```
Generated /var/lib/pulp/exports/My_Organization/Export-My_Repository/1.0/2021-03-02T03-35-24-00-00/metadata.json
```

2. 可选：查看导出的数据：

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

9.8. 永久导出存储库

在系统资源方面，导出存储库可能非常昂贵的操作。典型的 Red Hat Enterprise Linux 树可能会占据 Satellite 服务器上几 GB 空间。

在这种情况下，您可以使用 **Incremental Export** 只导出自之前导出以来更改的部分内容。增量导出通常会导致存档文件比完整导出要小。

以下示例显示了库生命周期环境中存储库的增量导出。

流程

1. 创建增量导出：

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

```
Generated /var/lib/pulp/exports/My_Organization/Export-My_Repository/3.0/2021-03-02T03-35-24-00-00/metadata.json
```

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
```

9.9. 跟踪您的导出

Satellite 保留所有导出的记录。每次您在上游 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，特别是在以递增方式导出时非常有用。

在为几个下游 Satellite 服务器从上游 Satellite 服务器导出内容时，您还可以跟踪为特定服务器导出的内容。这有助于您跟踪哪些内容被导出和在哪里。

在导出过程中使用 `--destination-server` 参数来指示目标服务器。这个选项可用于 **所有内容导出** 操作。

跟踪库导出的目的地

- 指定导出库时的目标服务器：

```
# 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"
```

9.10. 导入到库环境

您可以将导出的库内容导入到另一个 Satellite 服务器上组织的库生命周期阶段。有关从库环境中导出内容的更多信息，请参阅 [第 9.3 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。
- 如果导出的内容中存在任何红帽软件仓库，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 Role。

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` 的子目录。
- 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
# 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
```

3. 识别您要导入到的机构。
4. 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`。相对路径无法正常工作。

5. 要验证您导入了库内容，请检查 Product 和 Repositories 的内容。目标机构中会创建一个名为 **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** 并不表示直接分配给主机。相反，将主机分配到 **Default Organization View** 或其他内容视图，就像您通常一样。

9.11. 导入内容视图版本

您可以导入导出的内容视图版本，以在另一个 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 9.5 节“导出内容视图版本”](#)。

当您导入内容视图版本时，它具有相同的主号和次要版本号，并包含与软件包和勘误相同的存储库。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。
- 如果导出的内容中存在任何红帽软件仓库，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. 将导出的文件复制到您要导入的 Satellite 服务器上的 `/var/lib/pulp/imports` 的子目录。
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. 验证所有权是否已正确设置：

```
# 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-id=My_Organization_ID \
--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`。相对路径无法正常工作。

5. 要验证您成功导入了内容视图版本，请列出您的机构的内容视图版本：

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

9.12. 导入仓库

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。有关导出存储库内容的更多信息，请参阅 [第9.7节“导出存储库”](#)。

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软件仓库，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 Role。

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` 的子目录。
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
# 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
```

3. 识别您要导入到的机构。
4. 要将库内容导入到 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`。相对路径无法正常工作。

5. 要验证您导入了存储库，请检查 Product 和 Repository 的内容。

9.13. 使用HAMMER CLI CHEAT SHEET 导出和导入内容

表9.1. Export

作用	命令
完全导出机构的库	hammer content-export complete library --organization="My_Organization"
逐步导出组织的库（假设您已经导出了一些内容）	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"
以较小的块导出内容视图 (2GB slab)	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（假设您已经导出了一些内容）	hammer content-export incremental version --content-view="My_Content_View" --version=2.0 --organization="My_Organization"
完全导出仓库	hammer content-export complete repository --product="My_Product" --name="My_Repository" --organization="My_Organization"
逐步导出仓库（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 9.2. Import

作用	命令
导入到机构的库	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"
导入仓库	hammer content-import repository --organization="My_Organization" --path="/var/lib/pulp/imports/My_Exported_Repo_Dir"

第 10 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关联。例如，您可以将基本激活码与 Red Hat Enterprise Linux 工作站的订阅创建，并将其与特定环境的内容视图关联。

您可以在内容主机注册过程中使用激活码来提高进程的速度、简单性和一致性。请注意，只有在主机注册时使用激活码。如果对激活码进行了更改，它仅适用于将来使用被修改的激活码注册的主机。这些更改不会对现有主机进行任何更改。

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
- 生命周期阶段和内容视图
- 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

当您调配主机时，Satellite 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。注册主机后，激活码中的内容视图会覆盖主机组或主机设置的原始内容视图。然后，Satellite 将激活码中的内容视图用于以后的任务，例如重建主机。

当您重建主机时，请确保设置了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使用。

将同样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，激活码仅为内容主机设置初始配置。当内容主机注册到某个机构时，可以手动将组织的内容附加到内容主机。

将多个激活码与内容主机搭配使用

内容主机可以与组合的多个激活码关联，以定义主机设置。如果设置冲突，则最后指定的激活码将具有优先权。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", ...
```

10.1. 创建激活码

您可以使用激活码定义在注册过程中附加到主机的特定订阅集合。您添加到激活码的订阅必须在关联的内容视图中可用。

Subscription Manager 根据以下因素的不同附加订阅：

- 是否有任何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：在激活码中是否设置了系统目的？

请注意，Satellite 仅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在 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 auto-attach。

根据前面的因素，使用激活码订阅三个可能的情况：

1. 自动附加订阅的激活码。
如果没有指定订阅并自动附加，使用激活码搜索主机会从与激活码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，以增强自动订阅附加。
2. 为自动附加提供一组自定义订阅的激活码。
如果指定了订阅并启用 auto-attach，则使用激活码的主机会从激活码中指定的列表中选择最适合的订阅。在激活码中设置系统目的不会影响这种情况。
3. 带有完全一组订阅的激活码。
如果指定了订阅并禁用自动附加，则使用激活码的主机将与激活码中指定的所有订阅关联。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红帽提供的内容，分配给激活码，无论自动附加设置是什么，都会为注册的内容启用此产品。

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

流程

1. 在 Satellite Web UI 中，导航到 **Content > Activation keys**，再点 **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，请在 **Limit** 字段中输入您可以使用激活码注册的最大系统数量。如果您希望无限主机使用激活码注册，请确保选择了 **thick 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 **Content View** 列表中，选择要使用的内容视图。如果要使用已弃用的 **Katello 代理** 而不是 **远程执行**，内容视图必须包含 Satellite Client 6 存储库，因为它包含 **katello-agent** 软件包。
7. 如果启用了简单内容访问(SCA)：
 - a. 在 **Repository Sets** 选项卡中，确保只启用命名的存储库。
8. 如果没有启用 SCA：
 - a. 单击 **Subscriptions** 选项卡，然后单击 **Add** 子菜单。
 - b. 单击之前创建的订阅旁边的复选框。
 - c. 单击 **Add Selected**。
9. 单击 **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，以增强订阅自动附加。

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，输入以下命令使用注册过程中在主机上设置的系统目的配置激活码，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. 获取您的订阅 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 客户端 6 存储库的默认自动启用状态。默认状态被设置为 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"
```

10.2. 更新与激活码关联的订阅

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 5.7 节“更新多个主机上的红帽订阅”](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Activation keys**，然后点激活码的名称。
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 **Add**，然后选择要添加的订阅左侧的复选框，然后单击 **Add Selected**。
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，选择此激活码的内容视图，然后单击 **Save**。

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。

10.3. 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 置备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。

您可以使用主机注册功能、Satellite API 或 hammer CLI 将主机注册到 Satellite。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Register Host**。
2. 点 **Generate** 来创建注册命令。
3. 点 **文件** 图标将命令复制到您的剪贴板中。
4. 登录到您要注册并运行之前生成的命令的主机。
5. 检查 `/etc/yum.repos.d/redhat.repo` 文件，并确保已启用适当的存储库。

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. 登录到您要注册并运行之前生成的命令的主机。
3. 检查 `/etc/yum.repos.d/redhat.repo` 文件，并确保已启用适当的存储库。

API 流程

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}}'
```

使用激活码简化指定环境。如需更多信息，请参阅内容管理指南中的[管理激活码](#)。

要以命令行参数形式输入密码，请使用 `username:password` 语法。请记住，这会在 shell 历史记录中保存密码。

有关注册的更多信息，请参阅 [管理主机](#) 中的 [将主机注册到 Red Hat Satellite](#)。

2. 登录到您要注册并运行之前生成的命令的主机。
3. 检查 `/etc/yum.repos.d/redhat.repo` 文件，并确保已启用适当的存储库。

多个激活码

您可以在注册内容主机时使用多个激活码。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将它们合并。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中：

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

设置冲突

如果激活码中存在冲突的设置，则右侧最键将具有优先权。

- 有冲突的设置：**Service Level, Release Version, Environment, Content View, 和 Product Content**。
- 不冲突且主机获取它们的 union 设置：**Subscription 和 Host Collections**。
- 影响密钥本身而非主机配置行为的设置：**Content Host Limit 和 Auto-Attach**。

10.4. 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、架构和首选项等一组标准选择和附加与最佳相关订阅。

您可以启用自动附加，且没有与密钥关联的订阅。这种类型的密钥通常用于在您不希望虚拟机使用物理订阅时注册虚拟机，但要从 hypervisor 中继承基于主机的订阅。如需更多信息，请参阅 [在 Red Hat Satellite 中配置虚拟机订阅](#)。

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，导航到 **Content > Activation Keys**。
2. 点击您要编辑的激活码名称。
3. 点击 **Subscriptions** 选项卡。
4. 点 **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
```

10.5. 设置服务级别

您可以配置激活码，为使用激活码创建的新主机定义默认服务级别。设置默认服务级别仅选择要附加到主机的匹配订阅。例如，如果激活码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，导航到 **Content > Activation Keys**。
2. 点击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。该列表仅包含可用于激活码的服务级别。
5. 点击 **Save**。

CLI 过程

- 输入以下命令在激活码上将默认服务级别设置为 Premium：

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

第 11 章 管理勘误

作为红帽质量控制和发布过程的一部分，我们为客户提供每个官方红帽 RPM 版本的更新。红帽将相关软件包组编译成勘误以及提供更新描述的公告。公告有三种类型（重要）：

安全公告

描述软件包中找到的固定安全问题。此问题的安全性影响可能是 Low, Moderate, Important, 或 Critical。

程序错误修复公告

描述软件包的程序错误修复。

产品增强公告

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 Content Delivery Network (CDN) 同步时，Red Hat Satellite 会导入此勘误信息。Red Hat Satellite 还提供了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，并通过内容视图将它们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，标记为产品增强公告的勘误只能包含增强更新，同时程序错误修复公告勘误可以同时包含 bug 修复和增强，且安全公告可以包含所有三种类型。

在 Red Hat Satellite 中，有两个关键字来描述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的勘误，并可用于在内容主机上安装。可以从生命周期阶段和相关内容视图（但尚未安装）内容主机提供可安装的勘误。

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。

11.1. 检查可用勘误

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，导航到 **Content > 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：
 - 从列表中选择要检查的存储库。**所有** 软件仓库都被默认选择。
 - 默认选择 **Applicable** 复选框，以仅查看所选存储库中适用的勘误。选择 **Installable** 复选框，仅查看标记为可安装的勘误。
 - 要搜索勘误表，以以下方式在 **Search** 字段中输入查询：

parameter operator value

有关可用于搜索的参数列表，请参阅 [第 11.2 节“可用于勘误搜索的参数”](#)。在管理 Red Hat Satellite 中，查找 [支持的用于 Granular Search 的 Operator](#) 中的适用 Operator 列表。自动建

议在您输入时可以正常工作。您还可以将查询与使用 **and** 和 **or** 或运算符结合使用。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勘误 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 11.9 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 页面查看来检查可用的勘误表并选择要安装的勘误。

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts** 并选择您需要的主机。
2. 如果与主机关联有勘误，则新主机页面中的可安装勘误卡会显示一个交互式饼图，显示安全公告、错误修复和增强的分类。
3. 在新主机页面上，选择 **Content** 选项卡。
4. 在 Content 页面中，选择 **Errata** 选项卡。
5. 该页面显示所选主机的可安装勘误表。
6. 点击您要安装的任何勘误的复选框。
7. 使用您要添加到主机的勘误旁的垂直 ellipsis 图标，如果您没有使用 SSH 连接到目标主机，请选择 **Apply via Katello agent**。
8. 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**。
9. 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，要查看包含指定程序漏洞的所选产品的适用勘误，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"
```

11.2. 可用于勘误搜索的参数

参数	描述	示例
错误	根据 Bugzilla 号搜索。	<code>bug = 1172165</code>
CVE	按 CVE 号搜索。	<code>cve = CVE-2015-0235</code>
id	根据勘误 ID 搜索。在输入时，自动暂停系统会显示可用 ID 列表。	<code>id = RHBA-2014:2004</code>
发布日期	根据问题日期搜索。您可以指定准确日期，如 "Feb16, 2015" 或使用关键字，如 "是terday" 或 "1小时前"。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	根据完整的软件包构建名称进行搜索。在输入时，自动暂停系统会显示可用软件包列表。	<code>package = glib2-2.22.5-6.el6.i686</code>
package_name	按软件包名称搜索。在输入时，自动暂停系统会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的严重性搜索。指定 <i>Critical</i> 、 <i>Important</i> 或 <i>Moderate</i> 。	<code>severity = Critical</code>
title	按公告标题搜索。	<code>title ~ openssl</code>
type	根据公告类型搜索。指定 <i>security</i> 、 <i>bugfix</i> 或 <i>enhancement</i> 。	<code>type = bugfix</code>
已更新	根据上次更新的日期搜索。您可以使用与 发布 的参数相同的格式。	<code>updated = "6 days ago"</code>

11.3. 应用可安装勘误

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，导航到 **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，则新的 Host 页面的 **Installable Errata** 卡中会显示它们。

3. 在 **Content** 选项卡中，**勘误** 显示所选主机的可安装勘误表。
4. 点击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的垂直 ellipsis 图标，选择 **Apply via Remote Execution** 来使用 Remote Execution。如果要自定义 **远程执行**，请选择 **Apply via custom remote execution**，或者如果您没有使用 SSH 连接到目标主机，请选择 **Apply via Katello agent**。
6. 点 **Submit**。

11.4. 订阅勘误通知

您可以为 Satellite 用户配置电子邮件通知。用户会收到适用、可安装的勘误摘要、内容视图提升的通知或同步存储库后的通知。有关更多信息，请参阅管理 Red Hat Satellite 中的[配置电子邮件通知](#)。

11.5. 存储库依赖解析的限制

通过 Satellite，使用内容视图的增量更新可以解决一些存储库依赖项问题。但是，存储库级别的依赖项解析仍然在 occasion 中出现问题。

当存储库更新通过新依赖项可用时，Satellite 会检索软件包的最新版本以解决依赖项，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进一步的依赖项解决问题。

示例情境

客户端上的存储库具有软件包 **example_repository-1.0**，其依赖项为 **example_repository-libs-1.0**。该程序库也具有另一个软件包 **example_tools-1.0**。

软件包 **example_tools-1.1** 提供了安全勘误。**example_tools-1.1** 软件包需要 **example_repository-libs-1.1** 软件包作为依赖项。

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。存储库也具有软件包 **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**。由于您可以使用 yum 安装所有这些软件包，因此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，会发生依赖项解析问题，因为无法安装 **example_repository-libs-1.0** 和 **example_repository-libs-1.1**。

目前，这个问题还没有临时解决方案。时间帧和应用基本 RPM 和勘误之间的主要 Y 版本越大，依赖项解析问题的可能性越高。

11.6. 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。这些过滤器包括：

- **id** - 选择特定勘误以允许您的生成的存储库。
- **日期范围** - 定义日期范围，并包括在该日期范围内发布的一组勘误。
- **类型** - 选择包括程序错误修复、功能增强和安全更新等勘误类型。

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

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的存储库的内容视图。更多信息请参阅 [第 8.1 节“创建内容视图”](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Content Views** 并选择您要用来应用勘误表的内容视图。
2. 选择 **Yum Content > Filters** 并点 **New Filter**。
3. 在 **Name** 字段中输入 **Errata Filter**。
4. 从 **Content Type** 列表中，选择 **Erratum - Date** 和 **Type**。
5. 从 **Inclusion Type** 列表中，选择 **Exclude**。
6. 在 **Description** 字段中，从 **YYYY-MM-DD** 输入 **Exclude Errata items**。
7. 点击 **Save**。
8. 对于 **Errata Type**，请选择您要排除的勘误类型的复选框。例如，选中 **Enhancement and Bug fixes** 并清除 **Security** 复选框，以便在某些日期后排除增强和程序错误修复勘误，但包含所有安全勘误。
9. 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。
10. 选择 **Start Date** 来排除在所选日期或之后的所有勘误。
11. 将 **End Date** 字段留空。
12. 点击 **Save**。
13. 单击 **Publish New Version** 以发布生成的存储库。
14. 在 **Description** 字段中输入 **Adding errata filter**。
15. 点击 **Save**。
当内容视图完成发布时，**Content** 列会报告初始存储库中较少的软件包和勘误数量。这意味着该过滤器已成功排除上一年中的所有非安全勘误。
16. 点 **Versions** 选项卡。
17. 点公布的版本右侧的 **Promote**。
18. 选择您要将在内容视图版本提升到的环境。
19. 在 **Description** 字段中，输入提升的描述。
20. 点 **Promote Version** 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 --name "Filter Name" \
--description "Exclude errata items from the YYYY-MM-DD" \
--content-view "CV Name" --organization "Default 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 --start-date "YYYY-MM-DD" \
--content-view "CV Name" --content-view-filter="Filter Name" \
--organization "Default Organization" --types=security,enhancement,bugfix
```

3. 发布内容视图：

```
# hammer content-view publish --name "CV Name" \
--organization "Default Organization"
```

4. 将内容视图提升到生命周期环境中，以便包含的勘误可用于该生命周期阶段：

```
# hammer content-view version promote \
--content-view "CV Name" \
--organization "Default Organization" \
--to-lifecycle-environment "Lifecycle Environment Name"
```

11.7. 将勘误添加到事件内容视图

如果勘误可用，但无法安装，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中。例如，如果内容视图是版本 1.0，它将变为 Content View 版本 1.1，当您发布时，它将变为 Content View 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，导航到 **Content > Errata**。
2. 在 **Errata** 列表中，点击您要应用的勘误的名称。
3. 选择您要应用勘误表的内容主机，然后单击 **Apply to Hosts**。这会创建内容视图的增量更新。
4. 如果要应用勘误表到内容主机，请在 **发布复选框**后立即选择对 **Content Hosts** 的 **Apply Errata to Content Hosts**。
5. 点 **Confirm** 以应用勘误。

CLI 过程

1. 列出勘误及其对应的 ID :

```
# hammer erratum list
```

2. 列出不同的 content-view 版本和对应的 ID :

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

3. 将单个勘误应用到 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
```

11.8. 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 存储库与红帽提供的最新勘误同步。更多信息请参阅 [第 6.6 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 [管理主机指南中的注册主机](#)。
- 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 [管理主机指南](#) 中的配置和设置 [远程作业](#)。



注意

如果主机已经配置为接收已弃用 Katello Agent 的内容更新，请迁移到远程执行。如需更多信息，请参阅 [管理主机中的将主机从 Katello Agent 迁移到远程执行](#)。

将勘误应用到受管主机的过程取决于其操作系统。

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

使用这些步骤，将勘误表应用到运行 Red Hat Enterprise Linux 7 的主机。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Content Hosts**，再选择您要将勘误表应用到的主机。
2. 进入 **Errata** 选项卡，以查看勘误列表。
3. 选择要应用的勘误，然后单击 **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。使用勘误 ID 识别要应用的勘误。
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"
```

使用 **Katello 代理** (已弃用)

```
# hammer host errata apply --host "client.example.com" \
--errata-ids ERRATUM_ID1,ERRATUM_ID2...
```

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

CLI 过程

1. 在 Satellite 上, 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 :

```
# yum update Module_Stream_Name
```

11.9. 将勘误应用到多个主机

使用这些步骤检查并将勘误表应用到多个 RHEL 7 主机。

先决条件

- 将 Red Hat Satellite 存储库与红帽提供的最新勘误同步。更多信息请参阅 [第 6.6 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 [管理主机指南](#)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 [管理主机指南](#) 中的配置和设置 [远程作业](#)。



注意

如果主机已经配置为接收已弃用 Katello Agent 的内容更新，请迁移到远程执行。如需更多信息，请参阅[管理主机中的将主机从 Katello Agent 迁移到远程执行](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Errata**。
2. 点击您要应用的勘误的名称。
3. 单击 **Content Hosts** 选项卡。
4. 选择您要应用勘误表的主机，然后单击 **Apply to Hosts**。
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"
```

使用 **Katello 代理**（已弃用）

确定您要使用的勘误，并列此勘误适用的主机：

```
# hammer host list \
--search "applicable_errata = ERRATUM_ID" \
--organization "Default Organization"
```

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
```

此命令将带有 Errata_ID 的所有主机标识为适用的勘误，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中查找对应的任务：

```
# hammer task list
```

4. 查看所选任务的状态：

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

11.10. 将勘误应用到主机组

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"
```

使用 Katello 代理（已弃用）

```
# hammer host-collection erratum install \  
--errata "erratum_ID1,erratum_ID2,..." \  
--name "host_collection_name" \  
--organization "My_Organization"
```

第12章 管理容器镜像

借助 Satellite，您可以从各种来源导入容器镜像，并使用内容视图将它们分发到外部容器。

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

12.1. 导入容器镜像

您可以从 Red Hat Registry 或其他镜像 registry 导入容器镜像存储库。

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，导航到 **Content > Products**，再点 **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：要在 **Filter** 字段中输入您要使用的附加搜索条件来进一步重新定义 **Discovered Repository** 列表。
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 6.9 节“更改存储库的下载策略”](#)。
10. 可选：如果要从 **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. 仓库创建完成后，您可以点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，请点存储库，然后导航到 **Limit Sync Tags**。单击以编辑，并添加您要限制同步到 Satellite 的内容的任何标签。
16. 在 Satellite Web UI 中，导航到 **Content > Products** 并选择您的产品名称。
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，导航到 **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，您可以单击 **Container Image Manifests** 来列出可用的清单。在列表中，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 "Red Hat Container Catalog content" \
--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 \
--name "RHEL7" \
--content-type "docker" \
--url "http://registry.access.redhat.com/" \
--docker-upstream-name "rhel7" \
--product "Red Hat Container Catalog" \
--organization "My_Organization"
```

3. 同步存储库：

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

其他资源

- 有关手动创建产品和存储库的详情请参考 [第 6 章 导入内容](#)。

12.2. 管理容器名称模式

当您使用 Satellite 创建和管理容器时，容器会通过 Content View 版本和 Satellite 生命周期阶段的不同阶段进行更改。例如，如果您将容器镜像与上游存储库的名称 **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 name:

my_organization_production-custom_prompt-my_product-custom_ssh。当您要拉取容器镜像时，可能会造成问题，因为容器 registry 只能包含容器名称的一个实例。为避免 Satellite 命名惯例出现问题，您可以设置 registry 名称模式来覆盖默认名称，以确保您的容器名称明确以供以后使用。

限制

如果您使用 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，您可能会遇到命名冲突问题。例如：

- 如果设置 **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产品生命周期的内容视图。
- 如果设置 **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 命名模式时要小心。

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，导航到 **Content > Lifecycle Environments**，并创建执行环境，或选择要编辑的生命周期阶段。
2. 在 **Container Image Registry** 区域，点 **Registry Name Pattern** 区域右侧的编辑图标。
3. 使用变量和示例列表来确定您需要的 registry 名称模式。
4. 在 **Registry Name Pattern** 字段中，输入您要使用的 registry 名称模式。例如，使用 **repository.docker_upstream_name**：

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

5. 点击 **Save**。

12.3. 管理容器 REGISTRY 身份验证

您可以管理用于从 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进行身份验证以访问生命周期阶段中的 Satellite 中的容器镜像。例如，您可能希望允许用户从产品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，导航到 **Content > Lifecycle Environments**。
2. 选择您要管理身份验证的生命周期阶段。
3. 要允许未经身份验证的访问此生命周期环境中的容器，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**。

12.4. 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/**。

使用由服务器主机名决定的准确路径，将 root CA 文件复制到这些位置之一，并将文件命名为 **ca.crt**

在以下示例中，根据您的用例，将 `hostname.example.com` 替换为 `satellite-server.example.com` 或 `evince-server.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，将 root CA 文件复制到等同的 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!
```

12.5. 使用容器 REGISTRY

Podman 和 Docker 可用于从容器 registry 获取内容。

Capsules 中的容器 registry

在包含内容的 Capsule 上，容器 [网关](#) Capsule 插件充当容器 registry。它缓存 Katello 的身份验证信息，并将传入的请求代理到 Pulp。默认情况下，容器网关在 Capsule 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 13 章 管理 ISO 镜像

您可以使用 Satellite 从红帽的 Content Delivery Network 或其他源存储 ISO 镜像。您还可以上传其他文件，如虚拟机镜像，并在存储库中发布这些文件。

13.1. 从红帽导入 ISO 镜像

Red Hat Content Delivery Network 为某些产品提供 ISO 镜像。导入此内容的流程与为 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，导航到 **Content > Red Hat Repositories**。
2. 在 **Search** 字段中输入镜像名称，如 **Red Hat Enterprise Linux 7 Server (ISO)**。
3. 在 Available Repositories 窗口中，展开 **Red Hat Enterprise Linux 7 Server (ISO)**。
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，导航到 **Content > Products**，再点 **Red Hat Enterprise Linux Server**。
6. 单击 Red Hat Enterprise Linux Server 窗口的 **Repositories** 选项卡，然后点 **Red Hat Enterprise Linux 7 Server ISOs x86_64 7.2**。
7. 在 Red Hat Enterprise Linux 7 Server ISOs x86_64 7.2 窗口的右上角，单击 **Select Action** 并选择 **Sync Now**。

查看同步状态

- 在 Satellite Web UI 中，导航到 **Content > Sync Status** 并展开 **Red Hat Enterprise Linux Server**。

CLI 过程

1. 找到 Red Hat Enterprise Linux Server for **file** 存储库：

```
# 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"
```

13.2. 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 中完成以下步骤，或使用 hammer CLI。但是，如果要上传的文件大小大于 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，导航到 **Content > Products**，然后在 Products 窗口中点 **Create Product**。
2. 在 **Name** 字段中输入名称来识别产品。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：从 **Sync Plan** 列表中，为产品选择一个迁移计划。
5. 可选：在 **Description** 字段中输入产品的描述。
6. 点击 **Save**。
7. 在 Products 窗口中，单击新的产品，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**，再点 **Browse**。
14. 选择 **.iso** 文件，再点 **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 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更大文件，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。这提供了一种通用方法，用于在产品中纳入任意文件。

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，您可以使用正常的 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。客户端必须使用 **curl -O** 通过 HTTP 或 HTTPS 下载文件。

您只能在 Satellite 服务器中创建文件类型存储库，但创建文件类型存储库的方式具有灵活性。您可以在安装了 Satellite 的目录中创建独立文件类型存储库，或者在远程 HTTP 服务器上创建该目录的内容，然后将该目录的内容同步到 Satellite 中。当您有多个文件添加到 Satellite 存储库时，此方法很有用。

14.1. 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过程与创建任何自定义内容的步骤相同，但您选择 **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，导航到 **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。
8. 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。
9. 可选：在 **Upstream Username** 字段中，输入上游存储库的用户名（如果需要）。如果存储库不需要身份验证，请清除此字段。
10. 可选：在 **Upstream Password** 字段中输入上游存储库的对应密码。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。更多信息请参阅 [第 6.10 节“镜像策略概述”](#)。
13. 可选：在 **HTTP Proxy Policy** 字段中，选择或取消选择使用 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，以便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 客户端证书。
17. 可选：在 **SSL Client Key** 字段中，为存储库选择 SSL 客户端密钥。
18. 单击 **Save** 以创建存储库。

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 <i>sync_plan_name</i>	要搜索的同步计划名称
--sync-plan-id <i>sync_plan_id</i>	同步计划数字标识符

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>	支持存储库校验和，当前 'sha1' 和 'sha256'
--download-policy <i>policy_name</i>	下载 yum 存储库策略（即 'immediate' 或 'on_demand'）。
--gpg-key <i>gpg_key_name</i>	要搜索的键名称
--gpg-key-id <i>gpg_key_id</i>	GPG 密钥数字标识符

选项	描述
--mirror-on-sync <i>boolean</i>	同步时，该仓库必须从源以及过时的 RPM 进行镜像(mirror)？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-username <i>repository_username</i>	上游存储库用户（如果需要）
--upstream-password <i>repository_password</i>	上游存储库用户的密码
--url <i>source_repo_url</i>	源存储库的 URL
--verify-ssl-on-sync <i>boolean</i>	must Katello 验证上游 URL 的 SSL 证书是否由可信 CA 签名？设置为 true 或 false 、 yes 或 no 、 1 或 0 。

14.2. 在本地目录中创建自定义文件类型存储库

您可以使用 **pulp-manifest** 命令在安装 Satellite 的基本系统上创建自定义文件类型存储库。然后，您可以将文件同步到 Satellite 服务器中。将文件添加到文件类型存储库时，您可以像任何其他存储库一样使用这些文件。

使用这个流程在安装 Satellite 的基本系统上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 14.3 节“创建远程文件类型存储库”。

流程

要在本地目录中创建文件类型存储库，请完成以下步骤：

1. 确保启用了 Utils 存储库。

```
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms \
--enable=satellite-utils-6.11-for-rhel-8-x86_64-rpms
```

2. 安装 Pulp 清单软件包：

```
# satellite-maintain packages install python38-pulp_manifest
```

请注意，这个命令会停止 Satellite 服务并重新运行 `satellite-installer`。另外，为了避免停止该服务导致的停机时间，您可以使用以下内容：

```
# satellite-maintain packages unlock
# yum install python38-pulp_manifest
# satellite-maintain packages lock
```

3. 在 HTTP 服务器的公共文件夹中创建一个您要用作文件类型存储库的目录，例如：
`/var/lib/pulp/sync_imports/new_repo`

-

```
# mkdir /var/lib/pulp/sync_imports/new_repo
```

4. 在目录或创建测试文件中添加文件：

```
# touch my_file_repo/test.txt
```

5. 输入 Pulp Manifest 命令来创建清单：

```
# pulp-manifest my_file_repo
```

6. 验证清单是否已创建：

```
# ls my_file_repo
PULP_MANIFEST test.txt
```

7. 配置额外的导入路径，以同步 Satellite 中存在的文件存储库。例如，如果文件存储库位于 `/path/to/myrepo`，请输入：

```
# satellite-installer --foreman-proxy-content-pulpcore-additional-import-paths /path/to/myrepo
```

从文件类型存储库导入文件

要从本地目录中的文件类型存储库导入文件，请完成以下步骤：

1. 确保 Satellite 服务器中存在自定义产品。
2. 在 Satellite Web UI 中，导航到 **Content > Products**。
3. 选择产品的名称。
4. 单击 **Repositories** 选项卡，再选择 **New Repository**。
5. 在 **Name** 字段中输入存储库的名称。Satellite 根据您为 **Name** 输入的内容自动完成此字段。
6. 从 **Type** 列表中，选择存储库的内容类型。
7. 在 **Upstream URL** 字段中，输入带有要用作源的存储库的本地目录，格式为 **file:///my_file_repo**。
8. 选中 **Verify SSL** 复选框，以选中存储库的 SSL 证书或清除 **验证 SSL** 复选框。
9. 可选：在 **Upstream Username** 字段中输入您需要的上游用户名。
10. 可选：在 **Upstream Password** 字段中，为您的上游用户名输入对应的密码。
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 6.10 节“镜像策略概述”](#)。
13. 可选：在 **HTTP Proxy Policy** 字段中，选择所需的 HTTP 代理。默认值为 **Global Default**。
14. 可选：您可以清除 **Unprotected** 复选框，以便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 **GPG Key** 字段中，为存储库选择 GPG 密钥。
16. 可选：在 **SSL CA Cert** 字段中，为存储库选择 SSL CA 证书。
17. 可选：在 **SSL Client cert** 字段中，为存储库选择 SSL 客户端证书。
18. 可选：在 **SSL Client Key** 字段中，为存储库选择 SSL 客户端密钥。
19. 选择 **Save** 保存此存储库条目。

更新文件类型存储库

要更新文件类型存储库，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Content > Products**。
2. 选择产品的名称。
3. 选择您要更新的存储库的名称。
4. 在 **Select Action** 菜单中，选择 **Sync Now**。
5. 访问发布存储库的 URL 以查看文件。

14.3. 创建远程文件类型存储库

您可以使用 **pulp-manifest** 命令从 Satellite 服务器外部的文件目录创建自定义文件类型存储库。然后，您可以通过 HTTP 或 HTTPS 将文件同步到 Satellite 服务器中。将文件添加到文件类型存储库时，您可以像任何其他存储库一样使用这些文件。

使用这个流程在远程服务器上的目录中配置存储库。要在安装了 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 14.2 节“在本地目录中创建自定义文件类型存储库”](#)。

先决条件

在创建远程文件类型存储库前，请确保存在以下条件：

- 您已在 Satellite 或 Red Hat CDN 中注册了 Red Hat Enterprise Linux 7 服务器。
- 您的服务器具有 Red Hat Enterprise Linux Server 和 Satellite Client 6 软件仓库的权利。
- 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 7 系统管理员指南中的 [Apache HTTP 服务器](#)。

流程

要在远程目录中创建文件类型存储库，请完成以下步骤：

1. 在您的机器上，确保启用了正确的软件仓库。

- Red Hat Enterprise Linux 8 :

```
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms \
--enable=satellite-utils-6.11-for-rhel-8-x86_64-rpms
```

- Red Hat Enterprise Linux 7 :

■

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-7-server-satellite-utils-6.11-rpms
```

2. 安装 Pulp 清单软件包：

- Red Hat Enterprise Linux 8：

```
# yum install python38-pulp_manifest
```

- Red Hat Enterprise Linux 7：

```
# yum install tfm-pulpcore-python3-pulp_manifest
```

3. 在 HTTP 服务器的公共文件夹中创建一个您要用作文件类型存储库的目录：

```
# mkdir /var/www/html/pub/my_file_repo
```

4. 在目录或创建测试文件中添加文件：

```
# touch /var/www/html/pub/my_file_repo/test.txt
```

5. 输入 Pulp Manifest 命令来创建清单：

```
# pulp-manifest /var/www/html/pub/my_file_repo
```

6. 验证清单是否已创建：

```
# ls /var/www/html/pub/my_file_repo
PULP_MANIFEST test.txt
```

从远程文件类型存储库导入文件

要从远程文件类型存储库导入文件，请完成以下步骤：

1. 确保 Satellite 服务器中存在自定义产品，或创建自定义产品。如需更多信息，请参阅 [第 14.1 节 “创建自定义文件类型存储库”](#)
2. 在 Satellite Web UI 中，导航到 **Content > Products**。
3. 选择产品的名称。
4. 单击 **Repositories** 选项卡，再选择 **New Repository**。
5. 在 **Name** 字段中输入存储库的名称。Red Hat Satellite 根据您为 **Name** 输入的内容自动完成此字段。
6. 从 **Type** 列表中，选择 **文件**。
7. 在 **Upstream URL** 字段中，输入要用作源的上游存储库的 URL。
8. 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。
9. 可选：在 **Upstream Username** 字段中，输入上游存储库的用户名（如果需要）。如果存储库不需要身份验证，请清除此字段。

10. 可选：在 **Upstream Password** 字段中输入上游存储库的对应密码。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 6.10 节“镜像策略概述”。
13. 可选：在 **HTTP Proxy Policy** 字段中，选择所需的 HTTP 代理。默认值为 **Global Default**。
14. 可选：您可以清除 **Unprotected** 复选框，以便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 **GPG Key** 字段中，为存储库选择 GPG 密钥。
16. 可选：在 **SSL CA Cert** 字段中，为存储库选择 SSL CA 证书。
17. 可选：在 **SSL Client cert** 字段中，为存储库选择 SSL 客户端证书。
18. 可选：在 **SSL Client Key** 字段中，为存储库选择 SSL 客户端密钥。
19. 点 **Save**。
20. 要更新文件类型存储库，请导航到 **Content > Products**。选择包含您要更新的存储库的产品名称。
21. 在产品的窗口中，选择您要更新的仓库的名称。
22. 在 **Select Action** 菜单中，选择 **Sync Now**。

访问发布存储库的 URL 以查看文件。

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，导航到 **Content > Products**。
2. 根据名称选择自定义产品。
3. 按名称选择文件类型存储库。
4. 点击 **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 选项可指示文件、文件的目录或文件的 glob 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 将文件下载到客户端，如果选择了 **Unprotected** 选项，则可以选择通过 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 14.1 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，您需要客户端中的以下证书：
 1. **katello-server-ca.crt**。如需更多信息，请参阅 [管理 Red Hat Satellite 指南中的安装 Katello Root CA 证书](#)。
 2. 机构调试证书。如需更多信息，请参阅 [第 3.3 节“创建机构调试证书”](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Content > Products**。
2. 根据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. 复制发布存储库的 URL。

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，以适当的 HTTP 或 HTTPS 格式输入命令：

对于 HTTP：

```
# curl -O
```

```
http://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_File_Product_Label/My_Files_Label/my_file
```

对于 HTTPS：

```
# curl -O --cert ./_My-Organization-key-cert.pem_ --cacert katello-server-ca.crt
```

```
https://satellite.example.com/pulp/content/My_Organization_Label/Library/custom/My_File_Product_Label/My_Files_Label/my_file
```

附录 A. 将 NFS 共享用于内容存储

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。本附录演示了如何将 NFS 共享挂载到 Satellite 服务器的内容管理组件上。



重要

对 `/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。本例使用 `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. 在 Satellite 服务器上停止 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保持不变。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, 确认现有内容存在于 **var/lib/pulp** 上的挂载中 :

```
# 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。更多信息请参阅 [第 6.6 节“同步软件仓库”](#)。

附录 B. 导入 KICKSTART 存储库

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 中。

要为 Red Hat Enterprise Linux 7 导入 Kickstart 软件仓库，请完成 [第 B.1 节“为 Red Hat Enterprise Linux 7 导入 Kickstart 存储库”](#)。

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”](#)。

B.1. 为 RED HAT ENTERPRISE LINUX 7 导入 KICKSTART 存储库

要为 Red Hat Enterprise Linux 7 导入 Kickstart 软件仓库，请在 Satellite 中完成以下步骤。

流程

1. 进入位于 access.redhat.com 的红帽客户门户网站并登录。
2. 在窗口的左上角，单击 **Downloads**。
3. 在 **Red Hat Enterprise Linux 7** 右侧，单击 **Versions 7 及下面的版本**。
4. 从 **Version** 列表中，选择 Red Hat Enterprise Linux 7 所需的版本，如 7.7。
5. 在 **Download Red Hat Enterprise Linux** 窗口中，找到 ISO 镜像的二进制 DVD 版本，如 **Red Hat Enterprise Linux 7.7 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，创建一个挂载点并在该位置临时挂载 ISO 镜像：

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

8. 创建 Kickstart 目录：

```
# mkdir --parents /var/www/html/pub/sat-
import/content/dist/rhel/server/7/7.7/x86_64/kickstart/
```

9. 从 ISO 镜像复制 **kickstart** 文件：

```
# cp -a /mnt/iso/* /var/www/html/pub/sat-
import/content/dist/rhel/server/7/7.7/x86_64/kickstart/
```

10. 在列表文件中添加以下条目：

要到 **/var/www/html/pub/sat-import/content/dist/rhel/server/7/listing** 文件，请使用新行附加版本号。例如，对于 RHEL 7.7 ISO，请附加 **7.7**。

要到 **/var/www/html/pub/sat-import/content/dist/rhel/server/7/7.7/listing** 文件，请使用新行附加构架。例如，**x86_64**。

要到 `/var/www/html/pub/sat-import/content/dist/rhel/server/7/7.7/x86_64/listing` 文件，请使用新行附加 `kickstart`。

11. 从 ISO 镜像复制 `.treeinfo` 文件：

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/sat-
import/content/dist/rhel/server/7/7.7/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。

B.2. 为 RED HAT ENTERPRISE LINUX 8 导入 KICKSTART 软件仓库

要为 Red Hat Enterprise Linux 8 导入 Kickstart 软件仓库，请在 Satellite 中完成以下步骤。

流程

1. 进入位于 access.redhat.com 的红帽客户门户网站并登录。
2. 在窗口的左上角，单击 **Downloads**。
3. 点 **Red Hat Enterprise Linux 8**。
4. 在 **Download Red Hat Enterprise Linux** 窗口中，找到 ISO 镜像的二进制 DVD 版本，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，创建一个挂载点并在该位置临时挂载 ISO 镜像：

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

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/sat-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
# mkdir --parents /var/www/html/pub/sat-
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/sat-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart

# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/sat-
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：

要到 `/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/appstream/listing` 文件，请使用新行附加 **kickstart**。

要 `/var/www/html/pub/sat-import/content/dist/rhel8/8.1/x86_64/baseos/listing` 文件，请使用新行附加 **kickstart**：

要到 `/var/www/html/pub/sat-import/content/dist/rhel8/listing` 文件，请使用新行附加版本号。例如，对于 RHEL 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/sat-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo

# cp /mnt/iso/.treeinfo /var/www/html/pub/sat-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo
```

11. 打开 `/var/www/html/pub/sat-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 **variant = AppStream,BaseOS** 改为 **variants = BaseOS**

13. 在 **[tree]** 部分中，将 **variant = AppStream,BaseOS** 更改为 **variants = BaseOS**。

14. 在 **[variant-BaseOS]** 部分中，进行以下更改：

- 将 **package = BaseOS/Packages** 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/sat-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo` 文件的格式如下：

```
[checksums]
images/efiboot.img =
sha256:9ad9beee4c906cd05d227a1be7a499c8d2f20b3891c79831325844c845262bb6
images/install.img =
sha256:e246bf4aedfff3bb54ae9012f959597cdab7387aadb3a504f841bdc2c35fe75e
images/pxeboot/initrd.img =
sha256:a66e3c158f02840b19c372136a522177a2ab4bd91cb7269fb5bfdaaf7452efef
images/pxeboot/vmlinuz =
sha256:789028335b64ddad343f61f2abfdc9819ed8e9dfad4df43a2694c0a0ba780d16

[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/sat-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 = .`

- 将 **variant = AppStream,BaseOS** 改为 **variants = AppStream**
20. 在 **[tree]** 部分中, 将 **variant = AppStream,BaseOS** 改为 **variants = AppStream**
 21. 在 **[variant-AppStream]** 部分中, 进行以下更改 :
 - 将 **package = 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/sat-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。