



Red Hat Satellite 6.10

升级和更新 Red Hat Satellite

升级和更新 Red Hat Satellite Server 和 Capsule Server

Red Hat Satellite 6.10 升级和更新 Red Hat Satellite

升级和更新 Red Hat Satellite Server 和 Capsule Server

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

法律通告

Copyright © 2023 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

本指南描述了升级和更新 Red Hat Satellite Server、Capsule Server 和 hosts。

目录

第 1 章 升级概述	3
1.1. 先决条件	3
1.2. 升级路径	4
1.3. 升级的过程	5
1.4. 从 SATELLITE 升级胶囊	5
第 2 章 克隆 SATELLITE 服务器	6
2.1. 克隆过程概述	6
2.2. 先决条件	6
2.3. PULP 数据注意事项	7
2.4. 克隆 SATELLITE 服务器	7
第 3 章 升级 RED HAT SATELLITE	10
3.1. 升级 SATELLITE 服务器	10
3.2. 同步新存储库	21
3.3. 升级胶囊服务器	21
3.4. 通过卫星 WEB UI 同步胶囊服务器	25
3.5. 升级 SATELLITE 客户端	25
第 4 章 升级后的任务	27
4.1. 升级发现	27
4.2. 升级 VIRT-WHO	28
4.3. 删除 SATELLITE 工具存储库的之前版本	29
4.4. 重新声明 POSTGRESQL 空间	29
4.5. 更新模板、参数、查找键和值	30
4.6. 使用预定义的配置集调整卫星服务器	30
4.7. 验证外部 CAPSULE 服务器上的 PUPPET 环境	32
第 5 章 更新卫星服务器、胶囊服务器和内容主机	33
5.1. 更新 SATELLITE 服务器	33
5.2. 更新断开连接的 SATELLITE 服务器	34
5.3. 更新胶囊服务器	38

第 1 章 升级概述

本章详细介绍了 Red Hat Satellite 6.10 的先决条件和可用升级路径。在升级您当前的 Red Hat Satellite 6 安装前，请查看此信息。

在本指南中，术语更新、升级和迁移有以下含义：

升级

将 Satellite 服务器和胶囊服务器安装从 y-stream 发行版本提升到下一个流版本（如 Satellite 6.9 到 Satellite 6.10）的过程。

更新

将 Satellite Server 和 Capsule Server 安装从 z-stream 发行版本提升到下一个版本，如 Satellite 6.10.0 到 Satellite 6.10.1。

迁移

将现有 Satellite 安装移到另一个 Red Hat Enterprise Linux 服务器的过程。

对于交互式升级说明，您还可以使用红帽客户门户网站中的 Red Hat Satellite Upgrade Helper。此应用程序为您提供与当前的版本号完全匹配的确切指南。您可以找到特定于升级路径的说明，以及防止已知问题的步骤。如需更多信息，请参阅客户门户网站中的 [Satellite Upgrade Helper](#)。

请注意，您可以独立于 Satellite 升级胶囊。更多信息请参阅 [第 1.4 节“从 Satellite 升级胶囊”](#)。

1.1. 先决条件

升级到 Satellite 6.10 会影响整个 Satellite 基础架构。在继续前，完成以下操作：

- 阅读 [发行说明](#)。
- 查阅本指南，以便您了解升级过程及其影响。
- 规划升级路径。更多信息请参阅 [第 1.2 节“升级路径”](#)。
- 计划所需的停机时间。Satellite 服务在升级过程中关闭。根据硬件配置、网络速度以及服务器中保存的数据量，升级过程持续时间可能会有所不同。
升级 Satellite 需要大约需要 1 到 2 小时。

升级胶囊大约需要 10 - 30 分钟。

但是，从 6.9 升级到 6.10 也会迁移 Pulp 内容，这一步可能需要一些时间。有关准备 Pulp 迁移和升级过程的更多信息，请参阅 [第 3.1 节“升级 Satellite 服务器”](#)。

- 确定您的服务器上有足够的存储空间。如需更多信息，请参阅从 [连接的网络](#) 中安装 Satellite 服务器 [准备您的环境](#)，并在 [安装胶囊服务器中准备您的环境](#)。
- 备份您的卫星服务器和所有胶囊服务器。有关更多信息，请参阅《[管理红帽卫星 6.9 指南](#)》中的 [备份卫星服务器和胶囊服务器](#)。
- 计划更新包含 Satellite API 命令的任何脚本，因为一些 API 命令因 Satellite 版本之间的不同。



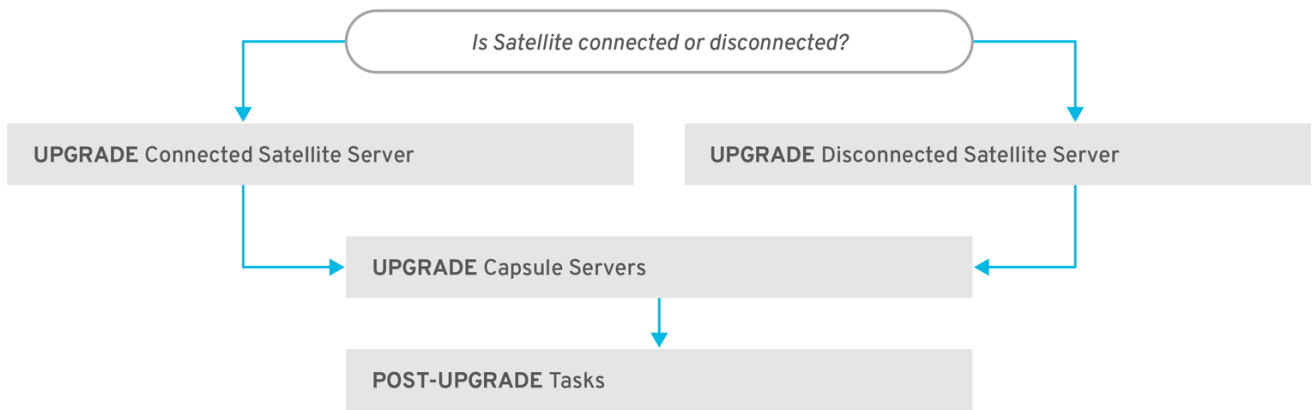
警告

如果您自定义配置文件，手动或使用 Hiera 等工具，当安装脚本升级或更新期间运行时，这些自定义会被覆盖。您可以将 `--noop` 选项与 `satellite-installer` 脚本一起使用以测试是否有更改。如需更多信息，请参阅红帽知识库解决方案 [如何在升级过程中使用 noop 选项检查 Satellite 配置文件的更改](#)。

1.2. 升级路径

您可以从 Red Hat Satellite 6.9 升级到 Red Hat Satellite 6.10。早期版本上的 Satellite 服务器和胶囊服务器必须首先升级到 Satellite 6.9。如需了解更多详细信息，请参阅 [Satellite 6.9 升级和更新 Red Hat Satellite 指南](#)。

图 1.1. Satellite 6.10 升级路径概述



SATELLITE_461763_0718



警告

不支持从 Beta 升级到 GA 版本。

升级至 Satellite 6.10 的高级别步骤如下。

1. 克隆现有的 Satellite 服务器。更多信息请参阅 [第 2 章 克隆 Satellite 服务器](#)。
2. 将 Satellite 服务器和所有胶囊服务器升级到 Satellite 6.10。更多信息请参阅 [第 3.1 节 “升级 Satellite 服务器”](#)。
3. 在所有 Satellite 客户端中升级到 Satellite 工具 6.10。更多信息请参阅 [第 3.5 节 “升级 Satellite 客户端”](#)。

将 Satellite 升级到将来的版本的注意事项

在开始升级到 Satellite 6.10 前，包括升级到 Satellite 服务器上的 Pulp 3，强烈建议您完成 Pulp 内容预迁移。如果您在 6.9 上不执行迁移，则在升级到 6.10 的过程中都会执行它们。对于较小的内容设置，或者您可以花费更长的时间停机时间，您可以继续执行预迁移。

对于 Capsules，您可以选择部署新的 6.10 胶囊，而不是升级。

对于将来在 Satellite 6.11 后升级，您将需要在 Satellite 服务器和 Capsules 上将操作系统从 RHEL 7 升级到 RHEL 8。您可以原地升级操作系统，或者通过克隆过程升级。后者包括迁移所有数据、配置和同步内容。



注意

如果您计划避免从 Pulp 2 升级到 Pulp 3，并因为 Pulp 3 更改而部署新的 Satellite 6.10 基础架构，您可能需要等待 Satellite 6.11 直接部署 RHEL 8。

1.3. 升级的过程

由于升级时间较长，请使用 **屏幕** 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 shell。有关使用 `screen` 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令](#)？您还可以查看 **屏幕** 手册页面以了解更多信息。

如果您丢失了运行 `upgrade` 命令的命令 shell，您可以在 `/var/log/foreman-installer/satellite.log` 中看到日志，以检查进程是否成功完成。

1.4. 从 SATELLITE 升级胶囊

您可以将 Satellite 升级到 6.10，并在版本 6.9 上保留 Capsules，直到您有能力也升级它们。

以前在 6.9 Capsules 上工作的所有功能。但是，在将 Capsules 升级到 6.10 之前，6.10 发行版本中添加的功能将无法正常工作。

在升级 Satellite 后升级胶囊可在以下示例情况下很有用：

1. 如果您想有几个较小的中断窗口，而不是一个较大的窗口。
2. 如果组织中的胶囊由几个团队管理，且位于不同的位置。
3. 如果使用负载均衡的配置，您可以升级一个负载均衡的 Capsule，并以较低版本保留其他负载均衡的 Capsules。这样，您可以在不中断任何中断的情况下升级所有胶囊。

第 2 章 克隆 SATELLITE 服务器

升级卫星服务器时，您可以选择创建并升级 Satellite 克隆，以确保在升级过程中不会丢失任何数据。升级完成后，您可以弃用早期版本的卫星服务器。

使用以下步骤克隆 Satellite 实例，以保留环境以准备升级。

Satellite 克隆工具不支持将 Capsule Server 迁移到 Red Hat Enterprise Linux 7。相反，您必须备份现有的 Capsule 服务器，在 Red Hat Enterprise Linux 7 上恢复它，然后重新配置胶囊服务器。

术语

请确定您理解以下术语：

源服务器：您克隆的服务器。

目标服务器：您将文件复制到新的服务器，并将源服务器克隆到其中。

2.1. 克隆过程概述

1. 备份源服务器。
2. 将源服务器克隆到目标服务器。
3. 关闭源服务器。
4. 更新目标服务器上的网络配置，使其与目标服务器的 IP 地址与其新主机名匹配。
5. 在 Content hosts 和 Capsules 中重启 `goferd` 以刷新连接。
6. 测试新的目标服务器。

2.2. 先决条件

要克隆 Satellite 服务器，请确保您具有以下资源可用：

- Red Hat Enterprise Linux 7 服务器的最小安装，成为目标服务器。不要安装 Red Hat Enterprise Linux 7 软件组或第三方应用程序。请确保您的服务器符合 [安装卫星服务器](#) 中的 [准备环境的所有规格](#)。
- 使用 `satellite-maintain` 备份您进行的 **Satellite 6.9** 备份。您可以通过或不使用 Pulp 数据使用备份。
- 目标服务器的 Satellite 订阅。

在开始克隆前，请确定以下条件：

- 目标服务器位于隔离的网络中。这可避免与胶囊服务器和主机进行不必要的通信。
- 目标服务器具有存储源服务器中的所有备份文件的容量。

自定义配置文件

如果您在没有由 `satellite-installer` 工具或 Satellite 备份过程管理的源服务器上有任何自定义配置，您必须手动备份这些文件。

2.3. PULP 数据注意事项

您可以克隆 Satellite 服务器，但不包括 Pulp 数据。但是，要使克隆的环境正常工作，您需要 Pulp 数据。如果目标服务器没有 Pulp 数据。它不是完全正常工作的 Satellite。

要将 Pulp 数据传送到目标服务器，有两个选项：

- 使用 Pulp 数据的备份克隆
- 在不使用 Pulp 数据的情况下克隆使用备份，并从源服务器中手动复制 `/var/lib/pulp`。

如果您的 `pulp_data.tar` 文件大于 500 GB，或者使用较慢的存储系统，如 NFS，您的 `pulp_data.tar` 文件大于 100 GB，请不要在备份中包含 `pulp_data.tar`，因为这可以在提取过程中造成内存错误。将源服务器的 `pulp_data.tar` 文件复制到目标服务器。

在没有 Pulp 数据的情况下备份

按照 [第 2.4 节 “克隆 Satellite 服务器”](#) 中的步骤操作，并使用以下步骤替换涉及使用 Pulp 数据克隆的步骤：

1. 使用活跃的 PostgreSQL 数据库执行备份，不包括 Pulp 数据：

```
# satellite-maintain backup offline --skip-pulp-content \
--assumeyes /var/backup
```

2. 停止并禁用 `satellite-maintain` 服务

```
# satellite-maintain service stop
# satellite-maintain service disable
```

3. 将 Pulp 数据复制到目标服务器：

```
# rsync --archive --partial --progress --compress \
/var/lib/pulp target_server.example.com:/var/lib/pulp
```

继续 [第 2.4.2 节 “克隆到目标服务器”](#)。

2.4. 克隆 SATELLITE 服务器

使用以下步骤克隆卫星服务器。请注意，由于您必须复制和转让的大量数据作为这些程序的一部分，这可能要花费大量时间才能完成。

2.4.1. 准备源服务器进行克隆

在源服务器中，完成以下步骤：

1. 验证 Satellite 订阅的池 ID：

```
# subscription-manager list --consumed \
--matches 'Red Hat Satellite'|grep "Pool ID:"|awk '{print $3}'
```

记下 `池 ID` 以供以后使用。

2. 删除 Red Hat Satellite 订阅：

—

```
# subscription-manager remove --serial=$(subscription-manager list \
--consumed \
--matches 'Red Hat Satellite'|grep "Serial:"|awk '{print $2}')
```

3. 确定 Pulp 数据的大小：

```
# du -sh /var/lib/pulp/
```

4. 如果您的 Pulp 数据小于 500 GB，则使用活跃的 PostgreSQL 数据库（包括 Pulp 数据）执行备份。如果您有超过 500 GB 的 Pulp 数据，请跳过以下步骤，并在继续前完成 [第 2.3 节“Pulp 数据注意事项”](#) 中的步骤。

```
# satellite-maintain backup offline --assumeeyes /var/backup
```

5. 停止并禁用 **satellite-maintain** 服务：

```
# satellite-maintain service stop
# satellite-maintain service disable
```

继续 [第 2.4.2 节“克隆到目标服务器”](#)。

2.4.2. 克隆到目标服务器

要克隆您的服务器，请在目标服务器上完成以下步骤：

1. **satellite-clone** 工具默认使用 **/backup/** 作为备份文件夹。如果您将复制到不同的文件夹，更新 **/etc/satellite-clone/satellite-clone-vars.yml** 文件中的 **backup_dir** 变量。
2. 将源 Satellite 中的备份文件放在目标服务器上的 **/backup/** 文件夹中。您可以挂载共享存储，或将备份文件复制到目标服务器上的 **/backup/** 文件夹。
3. 关闭源服务器。
4. 输入以下命令在客户门户网站中注册，附加订阅，并只启用所需的订阅：

```
# subscription-manager register your_customer_portal_credentials
# subscription-manager attach --pool=pool_ID
# subscription-manager repos --disable=*
# subscription-manager repos \
--enable=rhel-7-server-rpms \
--enable=rhel-server-rhsc1-7-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-satellite-6.9-rpms
```

5. 安装 **satellite-clone** 软件包：

```
# yum install satellite-clone
```

安装 **satellite-clone** 工具后，您可以调整任何配置，使其适用于 **/etc/satellite-clone/satellite-clone-vars.yml** 文件中的您自己的部署。

6. 运行 **satellite-clone** 工具：

```
# satellite-clone
```

- 7. 重新配置 DHCP、DNS、TFTP 和远程执行服务。克隆过程禁用目标卫星服务器上的这些服务，以避免与源卫星服务器冲突。
- 8. 在卫星 Web UI 中重新配置并启用 DHCP、DNS 和 TFTP。如需更多信息，请参阅[安装卫星服务器中的在卫星服务器上配置外部服务](#)。
- 9. 启用远程执行：

```
# satellite-installer --scenario satellite \
--enable-foreman-plugin-remote-execution \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

- 10. 使用用户名 **admin** 和密码 **changeme** 登录卫星 Web UI。立即更新 admin 密码以保护凭据。
- 11. 确定选择了正确的机构。
- 12. 导航到 **Content > Subscriptions**，然后单击 **Manage Manifest**。
- 13. 单击 **Refresh** 按钮，然后单击 **Close** 以返回到订阅列表。
- 14. 验证可用的订阅是否正确。
- 15. 按照 `/usr/share/satellite-clone/logs/reassociate_capsules.txt` 文件中的说明恢复胶囊与其生命周期环境之间的关联。
- 16. 更新网络配置，如 DNS，使其与目标服务器的 IP 地址与其新主机名匹配。**satellite-clone** 工具会将主机名更改为源服务器的主机名。如果要将其更改为不同的内容，您可以使用 **satellite-change-hostname** 工具。如需更多信息，请参阅在[管理 Red Hat Satellite 中调整 Satellite 或 Capsule Server](#)。
- 17. 如果源服务器使用 **virt-who** 守护进程，请在目标服务器上安装和配置它。将 `/etc/virt-who.d/` 目录中的所有 `virt-who.d/` 目录复制到目标服务器上的相同目录中。如需更多信息，请参阅在[Red Hat Satellite 中配置虚拟机订阅](#)。在使用下列章节执行升级后，您可以安全地弃用源服务器。

第 3 章 升级 RED HAT SATELLITE



警告

如果您在高可用性配置中安装了 Satellite 6，请在升级到 Satellite 6.10 前联系红帽支持。

使用以下步骤将现有 Red Hat Satellite 升级到 Red Hat Satellite 6.10：

1. [第 3.1 节 “升级 Satellite 服务器”](#)
2. [第 3.2 节 “同步新存储库”](#)
3. [第 3.3 节 “升级胶囊服务器”](#)
4. [第 3.5 节 “升级 Satellite 客户端”](#)
5. [第 4 章 升级后的任务](#)

在升级前，请参阅 [第 1.1 节 “先决条件”](#)。

3.1. 升级 SATELLITE 服务器

本节论述了如何将 Satellite 服务器从 6.9 升级到 6.10。

在继续操作前，将您的卫星服务器 6.9.z 更新至最新的 6.9.z 版本。您可以使用以下命令更新到最新的 z-stream。

```
# satellite-maintain upgrade run --target-version 6.9.z
```

在升级到 6.10 前，请确保已更新到最新的 z-stream。有关更新 Satellite 服务器的详情请参考 [第 5.1 节 “更新 Satellite 服务器”](#)。

虽然从 6.9 升级到 6.10 的过程也会将 Pulp 内容迁移，但这可能需要一些时间。您可以按以下流程减少此时间 [第 3.1.1 节 “准备将内容迁移到 Pulp 3”](#)

如果您尚未预从 Pulp 2 迁移到 Pulp 3，必须先升级到最新版本的卫星服务器，然后再开始迁移 Pulp 内容。

开始前

- 请注意，您可以独立于 Satellite 升级胶囊。更多信息请参阅 [第 1.4 节 “从 Satellite 升级胶囊”](#)。
- 在升级卫星服务器之前，检查和更新您的防火墙配置。如需更多信息，请参阅 [在安装 Satellite Server 中为安装准备您的环境](#)。
- 确保您没有从客户门户网站或 Satellite Web UI 中删除清单，因为这会移除您的内容主机的所有权利。

- 在升级前备份并删除所有 Foreman hook。仅在 Satellite 已知后恢复所有 hook，才能在升级完成后工作。请注意，Foreman Hooks 功能已弃用，并将在下一个 Satellite 版本中删除。
- 如果您编辑了任何默认模板，请通过克隆或导出文件来备份文件。推荐的方法克隆是推荐的方法，因为这会在以后更新或升级过程中被覆盖。要确认模板是否已编辑，您可以在升级前查看其历史，或者在升级后查看审计日志中的更改。在 Satellite Web UI 中，进入 **Monitor > Audits** 并搜索模板以查看所做的更改记录。如果您使用导出方法，请通过比较导出的模板和默认模板来恢复您的更改，请手动应用您的更改。
- Pulp 内容迁移是在升级到 6.10 前在 6.9 中进行的。
- 确保 `/var/lib/pulp/published` 有足够的存储空间，以便在开始迁移过程前以加倍。检查 `/var/lib/pulp/published` 的大小：

```
# du -sh /var/lib/pulp/published/
```

- 内容迁移可以在线运行，但使用处理器、磁盘和内存资源。同步和内容视图发布操作可能较长。
- 如果您没有预迁移 Pulp 内容，`PULP_CONTENT_PREMIGRATION_BATCH_SIZE` 设置定义同时处理的内容单元数。它会影响使用的 RAM 量和 I/O 负载。越低，`satellite-maintain` 内容准备的时间也越长。如果您保留了相关的内容，则升级停机时间也越长。
 - 默认值为 1000。
 - 如果系统有硬盘，或者存在有关 I/O 负载的问题，则推荐的值为 50。
 - 不建议使用较低值。
- 使用以下方法设置 `PULP_CONTENT_PREMIGRATION_BATCH_SIZE`：

1. 创建目录和文件：

```
$ mkdir /etc/systemd/system/pulpcore-worker@.service.d/
$ vi /etc/systemd/system/pulpcore-worker@.service.d/settings.conf
```

包含内容：

```
[Service]
User=pulp
Environment=PULP_CONTENT_PREMIGRATION_BATCH_SIZE=1000
```

2. 使用以下命令重启 `satellite-maintain` 服务：

```
# systemctl daemon-reload
# satellite-maintain service restart
```

如需更多信息，[请参阅准备将内容迁移到 Pulp 3。](#)

胶囊注意事项

- 如果使用 Content Views 来控制对胶囊服务器的基本操作系统的更新，或为胶囊服务器存储库控制更新，您必须发布这些内容视图的更新版本。
- 请注意，卫星服务器从 6.9 升级到 6.10，仍可使用位于 6.9 的胶囊服务器。



警告

如果实施自定义证书，您必须保留 `/root/ssl-build` 目录的内容，以及您在其中创建与自定义证书关联的所有源文件的内容。

在升级过程中无法保留这些文件会导致升级失败。如果删除了这些文件，则必须从备份中恢复这些文件，以便进行升级。

升级场景

- 要升级连接到红帽 Content Delivery Network 的卫星服务器，请执行 [第 3.1.2 节“升级连接的 Satellite 服务器”](#)。
- 要升级没有连接到 Red Hat Content Delivery Network 的 Satellite 服务器，请执行 [第 3.1.3 节“升级断开连接的 Satellite 服务器”](#)。

您无法升级自注册 Satellite。您必须将自注册的 Satellite 迁移到 Red Hat Content Delivery Network(CDN)，然后执行升级。要将自注册 Satellite 迁移到 CDN，请参阅在 [Satellite 6.10 升级和更新 Red Hat Satellite 指南中的迁移自助注册 Satellite](#)。

FIPS 模式

您不能将 Satellite 服务器从没有以 FIPS 模式运行的 RHEL 基础操作系统升级到在 FIPS 模式下运行的 RHEL 基础操作系统。

要在使用 FIPS 模式的 Red Hat Enterprise Linux 基础操作系统中运行 Satellite，您必须使用 FIPS 模式在全新置备的 RHEL 基础操作系统上安装 Satellite。如需更多信息，请参阅 [在安装 Satellite Server 中为安装准备您的环境](#)。

3.1.1. 准备将内容迁移到 Pulp 3

为 Pulp 3 准备内容的时间取决于内容的数量和内容视图的数量。对于大型系统，这可能意味着停机几天。为防止这种情况，请在运行最新版本的卫星服务器 6.9 时预迁移 Pulp 内容。这可减少总体升级停机时间。

在迁移到 Pulp 3 时，存储在 `/var/lib/pulp/published/` 目录中的数据量可以加倍。确保 `/var/lib/pulp` 卷上有足够的空间。如果用户运行 `satellite-maintain prep-6.10-upgrade`，则 `/var/lib/pulp/content` 中的内容不需要重复。删除 Pulp 2 后，可以声明额外的空间。但是，无法缩小 XFS 卷。可能需要将数据复制到不同的分区，以便将卷大小缩减到以前的值。

在迁移到 Pulp 3 时，存储在 PostgreSQL 数据目录中的数据量可能会显著增加。确保有足够的空间用于迁移。预估测 MongoDB Pulp 2 数据库的大小是 2-3 倍。

预迁移检查

如果您有任何 Composite Content Views，首先对它们执行操作，然后操作内容视图，则应始终是该序列。

- 确保所有启用的存储库完成同步过程。
- 如果有任何存储库与 **Warning** 或 **Error** 状态同步，请修复底层问题并重新同步存储库。

- 如果 Pulp 2 到 Pulp 3 迁移失败并显示 **NoMethodError: undefined method 'link?' for nil:NilClass**，请参阅红帽知识库解决方案 [Pulp 2 到 Pulp 3 迁移失败，并显示错误信息为 "NoMethodError: undefined method 'link?' for nil:NilClass"](#)。
- 确保系统中未暂停的任务与任何存储库或内容视图无关。
- 如果不再需要某些内容视图或 Composite Content View 版本，请考虑对它们进行清理。
- 有关如何使用 **hammer_cli** 执行旧的 [内容视图或 Composite Content View 版本](#) 的更多信息，请参阅 [如何使用 CLI/hammer 删除 Red Hat Satellite 6 中的旧内容视图版本](#)？
- 如果您不再需要或使用某些软件仓库，请禁用这些特定的软件仓库。
- 确保所有迁移计划都已禁用 till 迁移已完成。
- 当清理过程完成后，请按照 [如何删除 /var/lib/pulp on Capsule?](#) 中的步骤清理孤立数据？
- 确保您可以在 Satellite 中输入以下命令，且无错误：

```
# foreman-rake katello:correct_repositories COMMIT=true --trace
```

使用此流程开始将内容从 Pulp 2 迁移到 Pulp 3。

流程

1. 使用以下命令升级 Satellite 服务器前更新文件权限：

```
# satellite-maintain prep-6.10-upgrade
```

这在高延迟系统中可能需要一些时间。

2. 使用以下命令查看预备内容的详情：

```
# satellite-maintain content migration-stats
```

如迁移过程中经常使用此命令，以确定流程所需的时间。它还标识可能导致迁移失败的损坏或缺失的内容。输出类似以下示例：

```
Running Retrieve Pulp 2 to Pulp 3 migration statistics
=====
Retrieve Pulp 2 to Pulp 3 migration statistics:
=====Migration Summary=====
Migrated/Total RPMs: 0/367633
Migrated/Total errata: 0/20780140
Migrated/Total repositories: 0/33924
Estimated migration time based on yum content: 47 hours, 23 minutes
Note: ensure there is sufficient storage space for /var/lib/pulp/published to
double in size before starting the migration process.
Check the size of /var/lib/pulp/published with 'du -sh /var/lib/pulp/published/'
Note: ensure there is sufficient storage space for postgresql.
You will need additional space for your postgresql database.
The partition holding '/var/opt/rh/rh-postgresql12/lib/pgsql/data/' will need
additional free space equivalent to the size of your Mongo db database (/var/lib/mongodb/).
[OK]
```

3. 使用以下命令为 Pulp 迁移准备内容：

```
# satellite-maintain content prepare
```

作为最后一个步骤的一部分，**satellite-maintain 内容准备** 检查任何内容单元是否未迁移。如果它标识了损坏或缺失的内容，您可能会看到类似如下的内容：

```
=====Missing/Corrupted Content Summary=====
WARNING: MISSING OR CORRUPTED CONTENT DETECTED
Corrupted or Missing Rpm: 1000/104583
Corrupted or missing content has been detected, you can examine the list of content in
/tmp/unmigratable_content-20211025-74422-16cxfae and take action by either:
  1. Performing a 'Verify Checksum' sync under Advanced Sync Options, let it complete, and
re-running the migration
  2. Deleting/disabling the affected repositories and running orphan cleanup (foreman-rake
katello:delete_orphaned_content) and re-running the migration.
  3. Manually correcting files on the filesystem in /var/lib/pulp/content/ and re-running the
migration
  4. Mark currently corrupted or missing content as skipped (foreman-rake
katello:approve_corrupted_migration_content). This will skip migration of missing or
corrupted content.
```

有一些损坏或缺少内容的原因：

- 迁移期间发生存储库同步或内容视图发布或提升。
- 由于磁盘 rot 导致 `/var/lib/pulp/` 中磁盘上的文件被破坏。
- 由于过去的事件（如有部分恢复的磁盘丢失）导致 `/var/lib/pulp/content/` 中的文件缺少。
- Satellite 中的子系统（如 Katello 和 Pulp）之间有一些不匹配。如果存储库或内容视图在暂停的 Red Hat Satellite 任务里的步骤被不当删除，则会出现这种情况。运行 **foreman-rake katello:correct_repositories COMMIT=true** 可以更正此问题。

作为 **satellite-maintain content migration-stats** 命令、`/tmp/unmigratable_content-20211025-74422-16cxfae20211120-2149-1j4pmfae` 的一部分，查看已损坏或缺失的 RPM 列表。

有关确定哪些存储库不可数据库的内容所属的更多信息，请参阅 [如何确定要运行 Verify Checksum on reported corrupted RPM 的存储库](#)。

在大多数情况下，当使用按需下载策略将带有 **按需** 下载内容的 Satellite 配置软件仓库以及缺失或损坏的 RPM 的数量较低，您可以使用以下命令跳过这些内容：

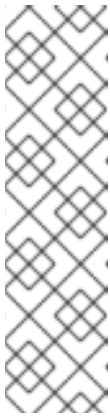
```
# foreman-rake katello:approve_corrupted_migration_content
```

如果您批准损坏或缺失的内容并进行升级，则升级后的内容不会出现在仓库或内容视图版本中。

如果上游存储库中存在这些软件包，则在升级后重新同步存储库时会再次添加它们。除非重新发布这些内容视图，否则不会恢复到内容视图。因为这些软件包可能不可用，所以升级过程仅检测到这一事实。

如果您怀疑迁移过程中发生了存储库同步或内容视图发布或提升，则再次运行 **satellite-maintain 内容准备** 迁移剩余内容。如果这是您第一次运行 **satellite-maintain 内容准备** 了，红帽建议根据需要运行它，以便尽量减少破坏或缺少 RPM 的数量。迁移剩余的内容会花费

较少的时间。



注意

您不能使用 **Ctrl + C** 终止 **satellite-maintain 内容准备过程**。如果您尝试使用 **Ctrl + C** 停止该进程，或者断开连接 SSH 会话，该进程不会终止，但在后台继续。您可以根据需要，使用以下命令来安全地终止进程，以便以后继续。

```
# satellite-maintain content prepare-abort
```

请注意，**satellite-maintain content prepare-abort** 可能需要几分钟的时间来终止该进程。在方便时，您可以继续使用 **satellite-maintain 内容准备** 迁移过程。

4. 此过程不会确认迁移已完成。您可以使用以下命令来决定该进程的接近程度：

```
# satellite-maintain content migration-stats
```

在指定迁移时间达到或接近零前的间隔。

5. 当将卫星服务器从 6.9 升级到 6.10 时，完成 Pulp 内容迁移的最后步骤。



注意

如果出现问题，您必须使用以下命令从开始重启迁移过程：

```
# satellite-maintain content migration-reset
```

6. 当将卫星服务器从 6.9 升级到 6.10 时，完成 Pulp 内容迁移的最后步骤。

3.1.2. 升级连接的 Satellite 服务器

对可访问公共互联网的 Satellite 服务器使用这个步骤



警告

如果您通过手动自定义配置文件，或者使用 Hiera 等工具，当安装脚本升级或更新期间运行时，这些更改会被覆盖。您可以将 **--noop** 选项与 **satellite-installer** 脚本一起使用以测试是否有更改。如需更多信息，请参阅红帽知识库解决方案 [如何在升级过程中使用 noop 选项检查 Satellite 配置文件的更改](#)。

升级 Satellite 服务器

1. 创建备份。
 - 在虚拟机上执行快照。
 - 在物理机器上创建备份。

有关备份的更多信息，请参阅《[管理红帽卫星 6.9 指南](#)》中的[备份卫星服务器和胶囊服务器](#)。

2. 可选：如果您在 `/etc/zones.conf` 或 `/etc/dhcp/dhcpd.conf` 文件中手动编辑 DNS 或 DHCP 配置，备份配置文件，因为安装程序只支持一个域或子网，因此可能需要从这些备份中恢复更改。
3. 可选：如果您手动编辑 DNS 或 DHCP 配置文件且不想覆盖更改，请输入以下命令：

```
# satellite-installer --foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

4. 在 Satellite Web UI 中，导航到 **Hosts > Discovered hosts**。在 Discovered Hosts 页面上，关闭电源，然后删除发现的主机。在 **Select an Organization** 菜单中，选择每个机构，再重复该过程以关闭并删除发现的主机。当升级完成后，请注意重新引导这些主机。
5. 确保 `apache::purge_configs: false` 条目在 Satellite 6.9/Capsule 6.9 服务器中的 `/etc/foreman-installer/custom-hiera.yaml` 文件中不存在或被注释掉。
6. 确保启用了 Satellite Maintenance 存储库：

```
# subscription-manager repos --enable \
rhel-7-server-satellite-maintenance-6-rpms
```

7. 检查可用版本以确认您要列出的版本：

```
# satellite-maintain upgrade list-versions
```

8. 使用健康检查选项来确定系统是否已准备好升级。在提示时，输入 hammer admin 用户凭证以使用 hammer 凭证配置 `satellite-maintain`。这些更改应用到 `/etc/foreman-maintain/foreman-maintain-hammer.yml` 文件。

```
# satellite-maintain upgrade check --target-version 6.10
```

在执行升级前，查看结果并解决所有突出显示的错误条件。

9. 由于升级时间较长，请使用 `屏幕` 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 shell。有关使用 `screen` 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令](#)？

如果您丢失了运行 `upgrade` 命令的命令 shell，您可以看到 `/var/log/foreman-installer/satellite.log` 文件中的日志信息，以检查进程是否已成功完成。

10. 执行升级：

```
# satellite-maintain upgrade run --target-version 6.10
```

11. 检查内核软件包是否最后一次更新：

```
# rpm -qa --last | grep kernel
```

12. 可选：如果自上次重启以来发生内核更新，请重启系统：

```
# reboot
```

13. 如果使用 BASH shell，在成功或失败升级后，请输入：

```
# hash -d satellite-maintain service 2> /dev/null
```

14. 如果您已经将内容从 Pulp 2 迁移到 Pulp 3，请删除所有 Pulp 2 内容。

```
# satellite-maintain content remove-pulp2
```

这将删除 Pulp 2 RPM、`/var/lib/pulp/content/` 中的内容、mongo 数据库和 Pulp 3 数据库中迁移内容。

3.1.3. 升级断开连接的 Satellite 服务器

如果您的 Satellite 服务器没有连接到 Red Hat Content Delivery Network，请使用这个步骤。



警告

- 如果您自定义了配置文件，可以手动或使用 Hiera 等工具，在升级或更新期间输入 **satellite-maintain** 命令时，这些更改会被覆盖。您可以将 **--noop** 选项与 **satellite-installer** 命令一起使用，以检查在升级或更新期间应用的更改。如需更多信息，请参阅红帽知识库解决方案 [如何在升级过程中使用 noop 选项检查 Satellite 配置文件中的更改](#)。
- **hammer** 导入和导出命令已被 **hammer content-import** 和 **hammer content-export** 工具替代。
如果您有使用 **hammer content-view** 版本导出、**hammer content-view** 版本 **export-legacy**、**hammer** 仓库导出 或其相应的导入命令的脚本，则必须调整它们以使用 **hammer content-export** 命令及其相应的导入命令。
- 如果实施自定义证书，您必须保留 `/root/ssl-build` 目录的内容，以及您在其中创建与自定义证书关联的所有源文件的内容。
在升级过程中无法保留这些文件会导致升级失败。如果删除了这些文件，则必须从备份中恢复这些文件，以便进行升级。

开始前

- 在升级 Satellite 服务器前，请检查和更新防火墙配置。如需更多信息，请参阅 [从断开连接的网络安装卫星服务器中的端口和防火墙要求](#)。
- 确保您没有从客户门户网站或 Satellite Web UI 中删除清单，因为这会移除您的内容主机的所有权利。
- 在升级前备份并删除所有 Foreman hook。只有在升级完成后，才会在 Satellite 已知后重新恢复 hook。



警告

如果实施自定义证书，您必须保留 `/root/ssl-build` 目录的内容，以及您在其中创建与自定义证书关联的所有源文件的内容。

在升级过程中无法保留这些文件会导致升级失败。如果删除了这些文件，则必须从备份中恢复这些文件，以便进行升级。

升级断开连接的 Satellite 服务器

1. 创建备份。

- 在虚拟机上执行快照。
- 在物理机器上创建备份。

2. 可以使用预升级脚本来检测冲突并列在 Satellite 服务器中具有重复条目的主机，这些条目可在升级后取消注册和删除。此外，它还将检测未分配给组织的主机。如果在 **Hosts > All hosts** 下列出主机，且没有机构关联，并且如果具有相同名称的内容主机已关联，则内容主机将自动取消注册。可通过在升级前将此类主机与机构关联来避免这种情况。

运行预升级检查脚本，获取升级后可以删除的主机列表。如果找到任何未关联的主机，请在升级前将它们与机构关联。

```
# foreman-rake katello:upgrade_check
```

3. 可选：如果您在 `/etc/zones.conf` 或 `/etc/dhcp/dhcpd.conf` 文件中手动编辑 DNS 或 DHCP 配置，备份配置文件，因为安装程序只支持一个域或子网，因此可能需要从这些备份中恢复更改。

4. 可选：如果您手动编辑 DNS 或 DHCP 配置文件且不想覆盖更改，请输入以下命令：

```
# satellite-installer --foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

5. 可选：如果将 PostgreSQL 用作外部数据库，请在 PostgreSQL 服务器上安装 **rh-postgresql12-postgresql-evr** 软件包，该软件包可从 **rhel-7-server-satellite-6.10-rpms** 存储库中获取：

```
# yum install rh-postgresql12-postgresql-evr
```

6. 在 Satellite Web UI 中，导航到 **Hosts > Discovered hosts**。如果发现主机可用，请关闭它们，然后删除 **Discovered hosts** 页面中的所有条目。依次选择所有其他组织，使用机构设置菜单，并根据需要重复此操作。在升级完成后重启这些主机。

7. 确保 **apache::purge_configs: false** 条目在 Satellite 6.9/Capsule 6.9 服务器中的 `/etc/foreman-installer/custom-hiera.yaml` 文件中不存在或被注释掉。

8. 确保所有外部胶囊服务器都分配给一个组织，否则可能会因为主机取消通知更改而取消注册。

9. 删除旧的软件仓库：

```
# rm /etc/yum.repos.d/*
```

10. 停止 **satellite-maintain** 服务。

```
# satellite-maintain service stop
```

有关额外步骤的更多信息，请参阅 [第 5.2 更新断开连接的 Satellite 服务器](#)。

1. 可选：如果您应用了自定义 Apache 服务器配置，请注意，当执行升级时自定义配置会恢复到安装默认值。

要预览升级过程中应用更改，请使用 **--noop**（无操作）选项输入 **satellite-installer** 命令。在以下步骤中输入 **satellite-maintain upgrade** 命令时，这些更改会被应用。

- a. 将以下行添加到 **/etc/httpd/conf/httpd.conf** 配置文件：

```
Include /etc/httpd/conf.modules.d/*.conf
```

- b. 重新启动 **httpd** 服务。

```
# systemctl restart httpd
```

- c. 启动 **postgresql** 和 **rh-mongodb34-mongod** 数据库服务。

```
# systemctl start postgresql
# systemctl start rh-mongodb34-mongod
```

- d. 使用 **--noop** 选项输入 **satellite-installer** 命令：

```
# satellite-installer --scenario satellite --verbose --noop
```

查看 **/var/log/foreman-installer/satellite.log**，以预览升级过程中应用的更改。找到 **+++** 和 **---** 符号来指示配置文件的更改。虽然使用 **--noop** 选项进入 **satellite-installer** 命令时不会应用任何更改，但模块中的一些 Puppet 资源会应用，并可能会显示失败消息。

- e. 停止 **satellite-maintain** 服务：

```
# satellite-maintain service stop
```

2. 由于升级时间较长，请使用 **屏幕** 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 shell。有关使用 **screen** 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令](#)？

如果您丢失了运行 **upgrade** 命令的命令 shell，您可以在 **/var/log/foreman-installer/satellite.log** 中看到日志，以检查进程是否成功完成。

3. 检查可用版本以确认您要列出的版本：

```
# satellite-maintain upgrade list-versions
```

4. 使用健康检查选项来确定系统是否已准备好升级。在提示时，输入 **hammer admin** 用户凭证以使用 **hammer** 凭证配置 **satellite-maintain**。这些更改应用到 **/etc/foreman-maintain/foreman-maintain-hammer.yml** 文件。

```
# satellite-maintain upgrade check --target-version 6.10 \
--whitelist="repositories-validate,repositories-setup"
```

在执行升级前，查看结果并解决所有突出显示的错误条件。

5. 执行升级：

```
# satellite-maintain upgrade run --target-version 6.10 \
--whitelist="repositories-validate,repositories-setup"
```



警告

如果您在包含配置子目录 *的目录中* 运行命令，您将遇到以下错误：

```
ERROR: Scenario (config/satellite.yaml) was not found, can not
continue.
```

在这种情况下，请更改目录，例如更改为 *root* 用户的主目录，然后再次运行此命令。

如果因为丢失或过时的软件包而导致脚本失败，则必须单独下载并安装。如需更多信息，请参阅从断开连接的 *网络安装 Satellite* 服务器中的 [解决 Package Dependency Errors](#) 部分。

6. 如果使用 BASH shell，在成功或失败升级后，请输入：

```
# hash -d satellite-maintain service 2> /dev/null
```

7. 如果您已经将内容从 Pulp 2 迁移到 Pulp 3，请删除所有 Pulp 2 内容：

```
# satellite-maintain content remove-pulp2
```

这将删除 Pulp 2 软件包上传并同步的 Pulp 2 软件包、*/var/lib/pulp/content/*、MongoDB 和迁移内容（Pulp 3 数据库中）。

8. 检查内核软件包是否最后一次更新：

```
# rpm -qa --last | grep kernel
```

9. 可选：如果自上次重启以来发生内核更新，停止 **satellite-maintain** 服务并重启系统：

```
# satellite-maintain service stop
# reboot
```

10. 可选：如果您手动编辑 DNS 或 DHCP 配置文件，请使用您所做的备份来检查和恢复 DNS 和 DHCP 配置文件所需的任何更改。

11. 如果您在上一步中进行更改，请重新启动 **satellite-maintain** 服务。

```
# satellite-maintain service restart
```

12. 如果您已安装了 OpenSCAP 插件，但没有可用的默认 OpenSCAP 内容，请输入以下命令。


```
# foreman-rake foreman_openscap:bulk_upload:default
```

13. 在 Satellite Web UI 中，进入 **Configure > Discovery Rules**，并将所选机构和位置与发现规则关联。

3.2. 同步新存储库

在升级胶囊服务器和 Satellite 客户端前，您必须启用并同步新的 6.10 软件仓库。

流程

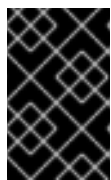
1. 在 Satellite Web UI 中，进入 **Content > Red Hat Repositories**。
2. 将 **Recommended Repositories** 开关切换到 **On** 位置。
3. 在结果列表中，展开以下软件仓库并点击 **Enable** 图标启用存储库：
 - 要升级 Satellite 客户端，请为客户端使用的所有 Red Hat Enterprise Linux 版本启用 Red Hat Satellite 工具 6.10 软件仓库。
 - 如果您有胶囊服务器，要升级它们，请同时启用以下软件仓库：
 - Red Hat Satellite Capsule 6.10（用于 RHEL 7 服务器）(RPMs)
 - Red Hat Satellite Maintenance 6（用于 RHEL 7 服务器）(RPMs)
 - Red Hat Ansible Engine 2.9 RPMs for Red Hat Enterprise Linux 7 Server
 - Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server



注意

如果 6.10 软件仓库不可用，请刷新订阅清单。导航到 **Content > Subscriptions**，单击 **Manage Manifest**，然后单击 **Refresh**。

4. 进入 **Content > Sync Status**。
5. 单击产品旁边的箭头，以查看可用的存储库。
6. 为 6.10 选择软件仓库。
7. 单击 **立即同步**。



重要

如果在尝试同步存储库时出现错误，请刷新清单。如果问题仍然存在，则引发支持请求。不要从客户门户网站或 Satellite Web UI 中删除清单，这将删除您的内容主机的所有权利。

8. 如果您使用 Content Views 来控制对胶囊服务器的基础操作系统的更新，请使用新的存储库更新这些内容视图，发布并提升其更新版本。如需更多信息，请参阅 [内容管理指南中的管理内容视图](#)。

3.3. 升级胶囊服务器

本节论述了如何将胶囊服务器从 6.9 升级到 6.10。

开始前

- 您必须升级 Satellite 服务器，然后才能升级任何胶囊服务器。请注意，您可以独立于 Satellite 升级胶囊。更多信息请参阅 [第 1.4 节“从 Satellite 升级胶囊”](#)。
- 确定在 Satellite 服务器中启用 Red Hat Satellite Capsule 6.10 软件仓库并同步。
- 确保您在卫星服务器上同步所需的存储库。更多信息请参阅 [第 3.2 节“同步新存储库”](#)。
- 如果使用 Content Views 来控制对胶囊服务器的基础操作系统的更新，请使用新的存储库更新这些内容视图，并发布其更新的版本。如需更多信息，请参阅 [内容管理指南中的管理内容视图](#)。
- 确保 Capsule 的基本系统已注册到新升级的卫星服务器。
- 确保胶囊在新升级的卫星服务器中具有正确的组织和位置设置。
- 在升级您的胶囊服务器前，检查和更新您的防火墙配置。如需更多信息，请参阅安装 [胶囊服务器中的为胶囊安装准备您的环境](#)。
- 确保创建 `/etc/puppetlabs/code/environments` 文件的备份。



警告

如果实施自定义证书，您必须保留 `/root/ssl-build` 目录的内容，以及您在其中创建与自定义证书关联的所有源文件的内容。

在升级过程中无法保留这些文件会导致升级失败。如果删除了这些文件，则必须从备份中恢复这些文件，以便进行升级。

升级胶囊服务器

1. 创建备份。
 - 在虚拟机上执行快照。
 - 在物理机器上创建备份。
有关备份的详情，请参考 [管理 Red Hat Satellite 6.9 指南中的备份 Satellite 服务器和胶囊服务器](#)。

2. 清理 yum 缓存：

```
# yum clean metadata
```

3. 确定安装了提供 `satellite-maintain` 的 `rubygem-foreman_maintain` 软件包，并处于最新状态：

```
# yum install rubygem-foreman_maintain
```

4. 在 Capsule Server 上，验证 `foreman_url` 设置指向 Satellite FQDN：

```
# grep foreman_url /etc/foreman-proxy/settings.yml
```

5. 确保 **apache::purge_configs: false** 条目在 Satellite 6.9/Capsule 6.9 服务器中的 **/etc/foreman-installer/custom-hiera.yaml** 文件中不存在或被注释掉。
6. 检查可用版本以确认您要列出的版本：

```
# satellite-maintain upgrade list-versions
```

7. 由于升级时间较长，请使用 **屏幕** 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 shell。有关使用 screen 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令？](#)

如果您丢失了运行 upgrade 命令的命令 shell，您可以看到 **/var/log/foreman-installer/satellite.log** 文件中的日志信息，以检查进程是否已成功完成。

8. 使用健康检查选项来确定系统是否已准备好升级：

```
# satellite-maintain upgrade check --target-version 6.10
```

在执行升级前，查看结果并解决所有突出显示的错误条件。

9. 执行升级：

```
# satellite-maintain upgrade run --target-version 6.10
```



警告

如果您在包含配置子目录 **的目录中** 运行命令，您将遇到以下错误：

```
ERROR: Scenario (config/capsule.yaml) was not found, can not continue.
```

在这种情况下，请更改目录，例如更改为 **root** 用户的主目录，然后再次运行此命令。

10. 如果您已经将内容从 Pulp 2 迁移到 Pulp 3，请删除所有 Pulp 2 内容。

```
# satellite-maintain content remove-pulp2
```

这将删除 Pulp 2 RPM、**/var/lib/pulp/content/** 中的内容、mongo 数据库和 Pulp 3 数据库中迁移内容。

通过此胶囊访问内容将失败，直到执行升级的胶囊的完整同步为止。

11. 检查内核软件包是否最后一次更新：

```
# rpm -qa --last | grep kernel
```

12. 可选：如果自上次重启以来发生内核更新，请重启系统：

```
# reboot
```

13. 可选：如果您手动编辑 DNS 或 DHCP 配置文件，请使用之前进行的备份检查并恢复对 DNS 和 DHCP 配置文件所需的任何更改。
14. 可选：如果您使用自定义软件仓库，请确保在升级完成后启用这些自定义软件仓库。
15. 在卫星服务器上，执行升级的胶囊的完整同步，因为 MongoDB 和 RPM 存储库不会自动使用 Satellite 迁移。

```
# hammer capsule content synchronize --name ${Capsule} --skip-metadata-check true --
async
```



注意

如果您在升级 Satellite 服务器前没有同步您的存储库，则运行这个命令会失败，将您的内容与胶囊服务器同步。在这种情况下，请按照 [第 3.4 节“通过卫星 Web UI 同步胶囊服务器”](#) 来同步您的胶囊服务器。

在 Satellite Web UI 中使用远程执行升级胶囊服务器

- 创建备份。
 - 在虚拟机上执行快照。
 - 在物理机器上创建备份。
有关备份的详情，请参考 [管理 Red Hat Satellite 6.9 指南中的备份 Satellite 服务器和胶囊服务器](#)。
- 可选：如果您在 `/etc/zones.conf` 或 `/etc/dhcp/dhcpd.conf` 文件中手动编辑 DNS 或 DHCP 配置，备份配置文件，因为安装程序只支持一个域或子网，因此可能需要从这些备份中恢复更改。
- 可选：如果您手动编辑 DNS 或 DHCP 配置文件且不想覆盖更改，请输入以下命令：

```
# {foreman-installer} --foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

- 在 Satellite Web UI 中，进入 **Monitor > Jobs**。
- 单击 **Run Job**。
- 从 **Job category** 列表中，选择 **Maintenance Operations**。
- 从 **Job template** 列表中，选择 **Capsule Upgrade Playbook**。
- 在 **Search Query** 字段中输入 Capsule 的主机名。
- 确保 **解决** 以显示 1 个主机。
- 在 **target_version** 字段中，输入 Capsule 的目标版本。
- 在 **whitelist_options** 字段中，输入白名单选项。

12. 对于 **查询的类型**，请点击 **Static Query** 或 **Dynamic Query**，具体取决于查询的类型。
13. 在 **Schedule** 中，选择作业执行的调度。

3.4. 通过卫星 WEB UI 同步胶囊服务器

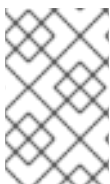
如果您在升级 Satellite 服务器前没有同步库，请使用这个步骤。

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**。
2. 单击您要同步的胶囊服务器的名称。
3. 单击 **Synchronize**。
4. 选择 **Complete Sync**。

3.5. 升级 SATELLITE 客户端

Satellite Tools 6.10 存储库提供 **katello-agent** 和 **katello-host-tools**，它提供用于管理勘误的通信服务。



注意

Katello 代理已弃用，并将在以后的 Satellite 版本中删除。迁移您的工作负载，以使用远程执行功能远程更新客户端。如需更多信息，请参阅 [管理主机指南中的从 Katello Agent 迁移到远程执行](#)。

对于使用 **katello-agent** 和 **goferd** 的部署，请将所有客户端更新至 **katello-agent** 的新版本。对于不使用 **katello-agent** 和 **goferd** 的部署，请将所有客户端更新至 **katello-host-tools** 的新版本。尽快完成这个操作，以便您的客户端与卫星服务器完全兼容。

先决条件

- 您必须已升级 Satellite 服务器。
- 您必须在 Satellite 中启用了新的 Satellite 工具 6.10 软件仓库。
- 您必须在 Satellite 中同步新存储库。
- 如果您之前没有在客户端中安装 **katello-agent**，而您想要安装它，请使用 manual 方法。更多信息请参阅 [手动升级 Satellite 客户端](#)。



警告

如果实施自定义证书，您必须保留 **/root/ssl-build** 目录的内容，以及您在其中创建与自定义证书关联的所有源文件的内容。

在升级过程中无法保留这些文件会导致升级失败。如果删除了这些文件，则必须从备份中恢复这些文件，以便进行升级。

使用 Bulk Repository Set UI 升级 Satellite 客户端：

1. 在 Satellite Web UI 中，导航到 **Hosts > Content Hosts** 并选择您要升级的 Content Hosts。
2. 在 **Select Action** 列表中，选择 **Manage Repository Sets**。
3. 在 **Repository Sets Management** 列表中，选择 **Red Hat Satellite Tools 6.9** 复选框。
4. 在 **Select Action** 列表中，选择 **Override to Disabled**，然后单击 **Done**。
5. 当进程完成后，从 **Select Action** 列表中，选择 **Manage Repository Sets** 的相同主机集合。
6. 在 **Repository Sets Management** 列表中，选择 **Red Hat Satellite Tools 6.10** 复选框。
7. 在 **Select Action** 列表中，选择 **Override to Enabled**，然后单击 **Done**。
8. 进程完成后，从 **Select Action** 列表中，在前面几组主机中，选择 **Manage Packages**。
9. 在 **Package search** 字段中，根据您的配置输入以下选项之一：
 - 如果您的部署使用 **katello-agent** 和 **goferd**，请输入 **katello-agent**。
 - 如果您的部署没有使用 **katello-agent** 和 **goferd**，请输入 **katello-host-tools**。
10. 在 **Update** 列表中，您必须选择 **via remote execution** 选项。这是必要的，因为如果您使用 Katello 代理更新软件包，软件包更新会破坏客户端和 Satellite 或 Capsule Server 之间的通信，从而导致更新失败。如需更多信息，请参阅 [管理主机指南中的配置和设置远程作业](#)。

手动升级 Satellite 客户端

1. 登录到客户端系统。
2. 为 Satellite 的早期版本禁用存储库。

```
# subscription-manager repos \
--disable=rhel-7-server-satellite-tools-6.9-rpms
```

3. 为 Satellite 的这个版本启用 Satellite 工具 6.10。

```
# subscription-manager repos \
--enable=rhel-7-server-satellite-tools-6.10-rpms
```

4. 根据您的配置，请完成以下步骤之一：

- 如果您的部署使用 **katello-agent** 和 **goferd**，请输入以下命令安装或升级 **katello-agent**：

```
# yum install katello-agent
```

- 如果您的部署没有使用 **katello-agent** 和 **goferd**，请输入以下命令安装或升级 **katello-host-tools**：

```
# yum install katello-host-tools
```

第 4 章 升级后的任务

本节中的一些流程是可选的。您可以选择只执行与安装相关的程序。

如果您使用基于 PXE 的发现过程，则必须在 Satellite 上完成发现升级过程，并在所有您要列在 **Hosts > Discovered hosts** 页面中的主机的胶囊服务器上。

- 在新的 Satellite 6.10 安装时，会禁用 **katello-agent**，使 **qpidd** 和 **qdrouted** 服务不可用。
- 如果 Satellite 6.9 升级到 Satellite 6.10，则 **katello-agent** 以及 **qpidd** 和 **qdrouted** 服务仍保持启用。
- 如果您不使用 **katello-agent** 并转换为远程执行，则可选择性地将 **katello-agent** 做为 Satellite 6.10 和 Capsule 6.10 的升级后任务禁用：

```
# satellite-installer --foreman-proxy-content-enable-katello-agent false
```

4.1. 升级发现

本节介绍了更新 PXELinux 模板以及传递给使用 PXE 引导的主机的引导镜像，将自身注册到卫星服务器。

从 Satellite 6.10，自定义模板与子网有一个单独的关联，且默认为该子网使用 TFTP Capsule。如果在升级后创建子网，则必须特别启用 Satellite 或 Capsule，以便为发现模板提供代理服务，然后使用发现的主机配置所有子网以使用特定的 *模板胶囊*。

在升级过程中，对于启用了 TFTP 代理的每个子网，模板胶囊都设置为与 TFTP Capsule 相同。升级后，请检查所有子网以验证是否已正确设置。

如果您不使用 PXE 引导主机来启用 Satellite 发现新主机，则不需要这些程序。

4.1.1. 在卫星服务器上升级 Discovery

1. 在 Satellite Web UI 中更新 Discovery 模板：
 - a. 进入 **Hosts > Provisioning templates**。
 - b. 在 **PXELinux 全局默认** 行上，单击 **Clone**。
 - c. 在 **Name** 字段中输入模板的新名称，如 **ACME PXE global default**。
 - d. 在模板编辑器字段中，将行 **ONTIMEOUT local** 更改为 **ONTIMEOUT discovery**，然后单击 **Submit**。
 - e. 进入 **Administer > Settings**。
 - f. 找到 **Global default PXELinux template**，再单击 **其值**。
 - g. 从菜单中选择新创建的模板的名称，然后单击“勾号”按钮。
 - h. 进入 **Hosts > Provisioning templates**。
 - i. 单击 **Build PXE Default**，然后单击 **OK**。
2. 在 Satellite Web UI 中，进入 **Configure > Discovery Rules**，并将所选机构和位置与发现规则关联。

4.1.2. 在胶囊服务器上升级发现

1. 验证 Foreman Discovery 软件包是否在卫星服务器上是最新的。

```
# satellite-maintain packages install tfm-rubygem-foreman_discovery
```

2. 如果上一步中的更新，请重启 **satellite-maintain** 服务。

```
# satellite-maintain service restart
```

3. 在卫星胶囊上升级发现镜像，它们通过发现的主机连接到 provisioning 网络，或为发现的主机提供 TFTP 服务。

```
# satellite-maintain packages install foreman-discovery-image
```

4. 在同一个实例上，安装提供 Proxy 服务的软件包，然后重新启动 **foreman-proxy** 服务。

```
# satellite-maintain packages install tfm-rubygem-smart_proxy_discovery
# service foreman-proxy restart
```

5. 在 Satellite Web UI 中，进入 **Infrastructure > Capsules**，并验证相关的 Capsule 在 features 列中列出了 **Discovery**。如果需要，从 **Actions** 下拉菜单中选择 **Refresh**。
6. 进入 **Infrastructure > Subnets** 以及您要使用发现的每个子网：
 - a. 点子网名称。
 - b. 在 **Capsules** 选项卡上，确保 **Discovery Capsule** 设置为您在上述配置的胶囊。

4.1.3. 验证子网具有模板胶囊

确定所有带有已发现主机的子网具有模板胶囊：

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Subnets**。
2. 选择您要检查的子网。
3. 在 **Capsules** 选项卡上，确保已经为此子网设置了 **Template Capsule**。

有关使用模板胶囊配置子网的更多信息，请参阅 [置备指南中的配置发现服务](https://access.redhat.com/documentation/zh-cn/red_hat_satellite/6.10/html-single/provisioning_guide/index#configuring-the-discovery-service)
https://access.redhat.com/documentation/zh-cn/red_hat_satellite/6.10/html-single/provisioning_guide/index#configuring-the-discovery-service

4.2. 升级 VIRT-WHO

如果卫星服务器或胶囊服务器上安装了 virt-who，则会在升级时升级它们。不需要进一步操作。如果在其他位置安装了 virt-who，则必须手动升级。

开始前

如果在注册到卫星服务器或胶囊服务器的主机上安装了 virt-who，则首先将主机升级到 Satellite Tools 6.10 存储库中提供的最新软件包。有关升级主机的详情，请参考 [第 3.5 节“升级 Satellite 客户端”](#)。

手动升级 virt-who

1. 升级 virt-who.

```
# yum upgrade virt-who
```

2. 重启 virt-who 服务，以便激活新版本。

```
# systemctl restart virt-who.service
```

4.3. 删除 SATELLITE 工具存储库的之前版本

完成 Satellite 6.10 后，可以从 Content Views 中删除 Red Hat Satellite Tools 6.9 存储库，然后禁用。

禁用 Satellite 工具程序库的版本 6.9:

1. 在 Satellite Web UI 中，进入 **Content > Red Hat Repositories**。
2. 在 **Enabled Repositories** 区域，找到 **Red Hat Satellite Tools 6.9 for RHEL 7 Server RPMs x86_64**。
3. 点击右侧的 **Disable** 图标。

如果存储库仍然在内容视图中，则无法禁用它。禁用的软件仓库中的软件包由调度的任务自动删除。

4.4. 重新声明 POSTGRESQL 空间

PostgreSQL 数据库可以使用大量磁盘空间，特别是载入的部署中。使用此流程在 Satellite 上重新声明一些磁盘空间。

流程

1. 停止除 **postgresql** 服务外的所有服务：

```
# satellite-maintain service stop --exclude postgresql
```

2. 切换到 **postgres** 用户并重新声明数据库中的空间：

```
# su - postgres -c 'vacuumdb --full --dbname=foreman'
```

3. vacuum 完成后启动其他服务：

```
# satellite-maintain service start
```

4. 确认 **/var/lib/pgsql/** 目录中存在这些文件：

```
# ls -l /var/lib/pgsql/
```

```
# du -sh /var/lib/pgsql/
```

5. 从 **/var/lib/pgsql/** 目录中删除数据：

```
# rm -rf /var/lib/pgsql/*
```

4.5. 更新模板、参数、查找键和值

在升级过程中，Satellite 会尝试定位为 Satellite 6.10 中弃用的宏，并将旧语法转换为默认的 Satellite 模板、参数和查找键和值的新语法。但是，Satellite 不会将旧语法转换为您已在克隆的模板中创建和创建的自定义模板中。

这个过程使用简单文本替换，例如：

```
@host.params['parameter1'] -> host_param('parameter1')
@host.param_true?('parameter1') -> host_param_true?('parameter1')
@host.param_false?('parameter1') -> host_param_false?('parameter1')
@host.info['parameters'] -> host_enc['parameters']
```



警告

如果您在 Satellite 中使用克隆的模板，请验证克隆的模板是否已从 Satellite 中原始模板的最新版本分离。同一模板的语法可能因 Satellite 版本而异。如果您的克隆的模板包含过时的语法，请更新语法以匹配模板的最新版本。

为确保此文本替换在升级过程中不会中断或省略您文件中的任何变量，请检查所有模板、参数以及查找旧语法的键和值，并手动替换。

升级后会发生以下错误，因为旧的语法保留在文件中：

```
undefined method '#params' for Host::Managed::Jail
```

修复过时的 `subscription_manager_registration` 片断

Satellite 6.4 使用 `redhat_register` 片断而不是 `subscription_manager_registration` 片断。

如果从 Satellite 6.3 及更早的版本升级，请确保替换自定义模板中的 `subscription_manager_registration` 片断，如下所示：

```
<%= snippet "subscription_manager_registration" %>
  ↓
<%= snippet 'redhat_register' %>
```

4.6. 使用预定义的配置集调整卫星服务器

如果您的 Satellite 部署包含 5000 的主机，您可以使用预定义的调优配置文件来改进 Satellite 的性能。

请注意，您不能在 Capsules 上使用调优配置集。

您可以根据 Satellite 管理及可用的硬件资源的主机数量，选择其中一个配置集。

调优配置文件位于 `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes` 目录中。

当使用 `--tuning` 选项运行 `satellite-installer` 命令时，请按照以下顺序将部署配置设置应用到 Satellite：

1. `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` 文件中定义的默认调优配置文件
2. 要应用到部署的调优配置文件，并在 `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/` 目录中定义
3. 可选：如果您配置了 `/etc/foreman-installer/custom-hiera.yaml` 文件，Satellite 会应用这些配置设置。

请注意，在 `/etc/foreman-installer/custom-hiera.yaml` 文件中定义的配置设置会覆盖调优配置文件中定义的配置设置。

因此，在应用调优配置文件前，您必须比较 `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` 文件中默认调节配置集中定义的配置设置，以及从 `/etc/foreman-installer/custom-hiera.yaml` 文件中删除任何重复的配置。yaml 文件从 `/etc/foreman-installer/custom-hiera.yaml` 文件中删除任何重复的配置。

default

受管主机数量：0-5000

RAM：20G

CPU 内核数：4

中

受管主机数量：5001-10000

RAM: 32G

CPU 内核数：8

大

受管主机数量：10001-20000

RAM：64G

CPU 内核数：16

extra-large

受管主机数量：20001-60000

RAM：128G

CPU 内核数：32

extra-extra-large

受管主机数量：60000+

RAM: 256G

CPU 内核数：48+

流程

1. 可选：如果您已经在 Satellite 服务器中配置了 `custom-hiera.yaml` 文件，请将 `/etc/foreman-installer/custom-hiera.yaml` 文件备份到 `custom-hiera.original`。如果文件被破坏，您可以使用备份文件将 `/etc/foreman-installer/custom-hiera.yaml` 文件恢复到其原始状态：

■

```
# cp /etc/foreman-installer/custom-hiera.yaml \  
/etc/foreman-installer/custom-hiera.original
```

2. 可选：如果您已经在 Satellite 服务器中配置了 **custom-hiera.yaml** 文件，请查看 **/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** 和您要应用的 **/usr/share/foreman-installer/config/foreman-installer.hiera/tuning/size.yaml** 中应用的默认调节配置文件的定义。将配置条目与 **/etc/foreman-installer/custom-hiera.yaml** 文件中的条目进行比较，并删除 **/etc/foreman-installer/custom-hiera.yaml** 文件中的任何重复配置设置。
3. 使用您要应用的配置集的 **--tuning** 选项输入 **satellite-installer** 命令。例如，要应用 medium 调节配置集设置，请输入以下命令：

```
# satellite-installer --tuning medium
```

4.7. 验证外部 CAPSULE 服务器上的 PUPPET 环境

将 Capsule 服务器升级到 6.10 后，请确保 Capsule 服务器上有 puppet 环境。

流程

1. 要查看 Puppet 环境，可打开 **/etc/puppetlabs/code/environments** 文件。

```
# vim /etc/puppetlabs/code/environments
```

2. 如果缺少任何 puppet 环境，请将其从 Satellite 服务器复制回来，或在升级前进行备份。
3. 在恢复后，确保具有 **/etc/puppetlabs/code/environments/** 文件中缺少的内容的适当权限和所有权。
4. 保存并关闭该文件。

第 5 章 更新卫星服务器、胶囊服务器和内容主机

使用本章将您现有的 Satellite Server、Capsule Server 和 Content Hosts 更新为一个新的次版本，例如从 6.10.0 升级到 6.10.1。

更新补丁安全漏洞和代码发布后发现的次要问题，通常对您的操作环境进行快速和破坏性。

在更新之前，备份您的卫星服务器和所有胶囊服务器。有关更多信息，请参阅《[管理红帽卫星](#)》指南中的[备份卫星服务器和胶囊服务器](#)。

5.1. 更新 SATELLITE 服务器

先决条件

- 确保您已同步了 Satellite、Capsule 和 Satellite 工具 6.10 的 Satellite 服务器存储库。
- 通过将更新的存储库提升到所有相关内容视图，确保可以更新每个外部胶囊和内容主机。



警告

如果您自定义配置文件，手动或使用 Hiera 等工具，当安装脚本升级或更新期间运行时，这些自定义会被覆盖。您可以将 `--noop` 选项与 `satellite-installer` 脚本一起使用以测试是否有更改。如需更多信息，请参阅红帽知识库解决方案 [如何在升级过程中使用 noop 选项检查 Satellite 配置文件中的更改](#)。

将 Satellite 服务器更新至下一个次版本

更新 Satellite 服务器：

1. 确定启用了 Satellite Maintenance 存储库：

```
# subscription-manager repos --enable \
  rhel-7-server-satellite-maintenance-6-rpms
```

2. 检查可用的版本以确认列出了下一个次版本：

```
# satellite-maintain upgrade list-versions
```

3. 使用健康检查选项来确定系统是否已准备好升级。首次使用这个命令时，`satellite-maintain` 会提示您输入 `hammer admin` 用户凭证，并将其保存在 `/etc/foreman-maintain/foreman-maintain-hammer.yml` 文件中。

```
# satellite-maintain upgrade check --target-version 6.10.z
```

在执行升级前，查看结果并解决所有突出显示的错误条件。

4. 由于更新时间较长，请使用 **屏幕** 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 shell。有关使用 `screen` 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令](#)？

如果您丢失了运行 `upgrade` 命令的命令 shell，您可以看到 `/var/log/foreman-installer/satellite.log` 文件中的日志信息，以检查进程是否已成功完成。

5. 执行升级：

```
# satellite-maintain upgrade run --target-version 6.10.z
```

6. 检查内核软件包是否最后一次更新：

```
# rpm -qa --last | grep kernel
```

7. 可选：如果自上次重启以来发生内核更新，停止 **satellite-maintain** 服务并重启系统：

```
# satellite-maintain service stop
# reboot
```

5.2. 更新断开连接的 SATELLITE 服务器

本节介绍了在 Air-gapped Disconnected setup 中更新所需的步骤，其中连接的卫星服务器（从 CDN 同步内容）与断开连接的 Satellite 服务器没有差距。

在连接的卫星服务器上完成以下步骤。

1. 确保您已在连接的卫星服务器中同步以下存储库。

```
rhel-7-server-ansible-2.9-rpms
rhel-7-server-rpms
rhel-7-server-satellite-6.10-rpms
rhel-7-server-satellite-maintenance-6-rpms
rhel-server-rhsc1-7-rpms
```

2. 下载机构的调试证书并将其保存到本地，例如 `/etc/pki/katello/certs/org-debug-cert.pem` 或您选择的位置。
3. 使用以下存储库信息在 `/etc/yum.repos.d` 中创建 Yum 配置文件：

```
[rhel-7-server-ansible-2.9-rpms]
name=Ansible 2.9 RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/server/7/$releasever/$basearch/ansible/2.9/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1
```

```
[rhel-7-server-rpms]
name=Red Hat Enterprise Linux 7 Server RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/server/7/7Server/x86_64/os/
enabled=1
```

```

sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

```

```

[rhel-7-server-satellite-6.10-rpms]
name=Red Hat Satellite 6 for RHEL 7 Server RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/server/7/7Server/x86_64/satellite/6.10/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt

```

```

[rhel-7-server-satellite-maintenance-6-rpms]
name=Red Hat Satellite Maintenance 6 for RHEL 7 Server RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/server/7/7Server/x86_64/sat-maintenance/6/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

```

```

[rhel-server-rhsc1-7-rpms]
name=Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/server/7/7Server/x86_64/rhsc1/1/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

```

4. 在配置文件中，将 `sslclientcert` 中的 `/etc/pki/katello/certs/org-debug-cert.pem` 替换为下载的机构调试证书的位置。

5. 为您的部署更新 `satellite.example.com` 正确的 FQDN。

6. 将 `My_Organization` 替换为 `baseurl` 中的正确组织标签。要获取机构标签，请输入以下命令：

```
# hammer organization list
```

7. 输入 `reposync` 命令：

```
# reposync --delete --download-metadata -p ~/Satellite-repos -n \
-r rhel-7-server-ansible-2.9-rpms \
```

```
-r rhel-7-server-rpms \
-r rhel-7-server-satellite-6.10-rpms \
-r rhel-7-server-satellite-maintenance-6-rpms \
-r rhel-server-rhsc-7-rpms
```

这会将存储库的内容从连接的卫星服务器下载，并将它们存储在 `~/Satellite-repos` 目录中。Red Hat Enterprise Linux 7 中的 `reposync` 命令下载 RPM，而不是 Yum 元数据。

因此，您必须在 `Satellite-repos` 的每个子目录中手动运行 `createrepo`。请确定您安装了 `createrepo rpm`。如果没有使用以下命令进行安装：

```
# satellite-maintain packages install createrepo
```

运行以下命令，以在 `~/Satellite-repos` 的每个子目录中创建 `repopdata`。

```
# cd ~/Satellite-repos
# for directory in */
do
  echo "Processing $directory"
  cd $directory
  createrepo .
  cd ..
done
```

8. 验证 RPM 已经下载，并且位于 `~/Satellite-repos` 的各个子目录中是否生成了存储库数据目录。

9. 归档目录的内容

```
# cd ~
# tar czf Satellite-repos.tgz Satellite-repos
```

10. 使用生成的 `Satellite-repos.tgz` 文件在非联网卫星服务器中升级。

在非联网卫星服务器上执行以下步骤

1. 将生成的 `Satellite-repos.tgz` 文件复制到断开连接的 Satellite 服务器中

2.

将存档 提取到可由 root 用户访问的任意位置。在以下示例中，/root 是提取位置。

```
# cd /root
# tar zxf Satellite-repos.tgz
```

3.

使用以下存储库信息在 /etc/yum.repos.d 中创建 Yum 配置文件：

```
[rhel-7-server-ansible-2.9-rpms]
name=Ansible 2.9 RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-ansible-2.9-rpms
enabled=1

[rhel-7-server-rpms]
name=Red Hat Enterprise Linux 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-rpms
enabled=1

[rhel-7-server-satellite-6.10-rpms]
name=Red Hat Satellite 6 for RHEL 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-satellite-6.10-rpms
enabled=1

[rhel-7-server-satellite-maintenance-6-rpms]
name=Red Hat Satellite Maintenance 6 for RHEL 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-satellite-maintenance-6-rpms
enabled=1

[rhel-server-rhsc1-7-rpms]
name=Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=file:///root/Satellite-repos/rhel-server-rhsc1-7-rpms
enabled=1
```

4.

在配置文件中，将 /root/Satellite-repos 替换为提取的位置。

5.

检查可用的版本以确认列出了下一个次版本：

```
# satellite-maintain upgrade list-versions
```

6.

使用健康检查选项来确定系统是否已准备好升级。首次使用这个命令时，satellite-maintain 会提示您输入 hammer admin 用户凭证，并将其保存在 /etc/foreman-maintain/foreman-maintain-hammer.yml 文件中。

```
# satellite-maintain upgrade check --whitelist="check-upstream-repository,repositories-validate" --target-version 6.10.z
```

7.

在执行升级前，查看结果并解决所有突出显示的错误条件。

8.

由于更新时间较长，请使用 `屏幕` 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 `shell`。有关使用 `screen` 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令](#)？

如果您丢失了运行 `upgrade` 命令的命令 `shell`，您可以看到 `/var/log/foreman-installer/satellite.log` 文件中的日志信息，以检查进程是否已成功完成。

9.

执行升级：

```
# satellite-maintain upgrade run --whitelist="check-upstream-repository,repositories-validate" --target-version 6.10.z
```

10.

检查内核软件包是否最后一次更新：

```
# rpm -qa --last | grep kernel
```

11.

可选：如果自上次重启以来发生内核更新，停止 `satellite-maintain` 服务并重启系统：

```
# satellite-maintain service stop
# reboot
```

5.3. 更新胶囊服务器

使用这个步骤将 `Capsule Servers` 更新至下一个次版本。

流程

1.

确保启用了 `Satellite Maintenance` 存储库：

```
# subscription-manager repos --enable \
rhel-7-server-satellite-maintenance-6-rpms
```

2.

检查可用的版本以确认列出了下一个次版本：

```
# satellite-maintain upgrade list-versions
```

3. 使用健康检查选项来确定系统是否已准备好升级：

```
# satellite-maintain upgrade check --target-version 6.10.z
```

在执行升级前，查看结果并解决所有突出显示的错误条件。

4. 由于更新时间较长，请使用 `屏幕` 等实用程序暂停并重新附加通信会话。然后，您可以检查升级进度，而无需持续连接到命令 shell。有关使用 `screen` 命令的更多信息，请参阅 [Red Hat 知识库](#) 中的 [如何使用 screen 命令](#)？

如果您丢失了运行 `upgrade` 命令的命令 shell，您可以看到 `/var/log/foreman-installer/satellite.log` 文件中的日志信息，以检查进程是否已成功完成。

5. 执行升级：

```
# satellite-maintain upgrade run --target-version 6.10.z
```

6. 检查内核软件包是否最后一次更新：

```
# rpm -qa --last | grep kernel
```

7. 可选：如果自上次重启以来发生内核更新，停止 `satellite-maintain` 服务并重启系统：

```
# satellite-maintain service stop  
# reboot
```