



Red Hat Satellite 6.13

管理主机

在 Red Hat Satellite 6 环境中管理主机的指南。

Red Hat Satellite 6.13 管理主机

在 Red Hat Satellite 6 环境中管理主机的指南。

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

本指南论述了如何在 Red Hat Satellite 环境中配置和使用主机。在继续此工作流前，您必须成功安装 Red Hat Satellite 6 服务器和任何所需的 Capsule 服务器。

目录

向红帽文档提供反馈	5
第 1 章 SATELLITE 中的主机概述	6
第 2 章 管理主机	7
2.1. 在 RED HAT SATELLITE 中创建主机	7
2.2. 克隆主机	9
2.3. 从 HYPERVISOR 将虚拟机与 SATELLITE 关联	9
2.4. 编辑主机的系统目的	9
2.5. 编辑多个主机的系统目的	10
2.6. 更改主机的模块流	11
2.7. 创建主机组	12
2.8. 为每个生命周期环境创建主机组	14
2.9. 将主机添加到主机组	15
2.10. 更改主机的主机组	15
2.11. 将主机添加到主机集合	16
2.12. 使用主机详情选项卡	17
2.13. 更改主机的内容源	18
2.14. 更改主机的环境	19
2.15. 更改主机的受管状态	19
2.16. 在主机上启用 TRACER	20
2.17. 在主机上重启应用程序	21
2.18. 将主机分配给特定机构	21
2.19. 将主机分配给特定位置	22
2.20. 在主机间切换	23
2.21. 从内容主机查看主机详情	23
2.22. 选择 HOST COLUMNS	24
2.23. 从 SATELLITE 中删除主机	24
第 3 章 注册主机和设置主机集成	27
3.1. 注册中支持的客户端	27
3.2. 注册方法	28
3.3. 使用全局注册注册主机	29
3.4. 使用启动脚本注册主机	35
3.5. 安装 KATELLO 代理	46
3.6. 安装 TRACER	47
3.7. 在主机注册过程中安装和配置 PUPPET 代理	48
3.8. 手动安装和配置 PUPPET 代理	49
3.9. 在主机注册过程中运行 ANSIBLE 角色	51
第 4 章 将主机从 KATELLO AGENT 迁移到远程执行	52
第 5 章 添加网络接口	55
5.1. 添加物理接口	55
5.2. 添加虚拟接口	57
5.3. 添加绑定接口	58
5.4. SATELLITE 中可用的绑定模式	60
5.5. 添加基板管理控制器(BMC)接口	60
第 6 章 将主机升级到 RED HAT ENTERPRISE LINUX 下一个主版本	63
第 7 章 将主机转换为 RED HAT ENTERPRISE LINUX	66
7.1. 用于转换的 ANSIBLE 变量	68

第 8 章 使用 RHEL WEB 控制台进行主机管理和监控	70
8.1. 在 SATELLITE 中启用 RHEL WEB 控制台	70
8.2. 使用 RHEL WEB 控制台管理和监控主机	70
8.3. 在 SATELLITE 上禁用 RHEL WEB 控制台	71
第 9 章 使用 RED HAT INSIGHTS 监控主机	73
9.1. 在 SATELLITE 中将 RED HAT INSIGHTS 与主机搭配使用	73
9.2. 为主机创建 INSIGHTS 计划	75
第 10 章 使用报告模板来监控主机	77
10.1. 生成主机监控报告	77
10.2. 创建报告模板	79
10.3. 导出报告模板	81
10.4. 使用 SATELLITE API 导出报告模板	82
10.5. 导入报告模板	84
10.6. 使用 SATELLITE API 导入报告模板	85
10.7. 生成已安装软件包列表	86
10.8. 创建报告模板以监控权利	87
10.9. 报告模板安全模式	89
第 11 章 配置主机集合	90
11.1. 创建主机集合	90
11.2. 克隆主机集合	90
11.3. 删除主机集合	91
11.4. 将主机添加到 BULK 中的主机组中	91
11.5. 从主机组中删除主机	92
11.6. 将内容添加到主机集合	93
第 12 章 配置和设置远程作业	101
12.1. 关于在主机上运行作业	101
12.2. 远程执行 workflow	102
12.3. 远程执行的权限	103
12.4. 远程执行的传输模式	104
12.5. 配置主机以使用 PULL 客户端	105
12.6. 创建作业模板	107
12.7. 按名称导入 ANSIBLE PLAYBOOK	108
12.8. 导入所有可用的 ANSIBLE PLAYBOOK	109
12.9. 将 FALLBACK 配置为 SATELLITE 中的任何 CAPSULE REMOTE EXECUTION SETTING	110
12.10. 在 SATELLITE 中配置全局胶囊远程执行设置	111
12.11. 在 PUSH 模式中为远程执行作业设置替代目录	112
12.12. 在 PULL 模式中为远程执行作业设置替代目录	113
12.13. 更改权限提升方法	113
12.14. 为远程执行发布 SSH 密钥	114
12.15. 为远程执行发布 SSH 密钥	115
12.16. 将密码添加到用于远程执行的 SSH 密钥	115
12.17. 使用 SATELLITE API 为远程执行提供 SSH 密钥	116
12.18. 在置备过程中将 KICKSTART 模板配置为分布式 SSH 密钥	116
12.19. 为 KERBEROS TICKET GRANTING TICKET 配置 KEYTAB	117
12.20. 为远程执行配置 KERBEROS 身份验证	118
12.21. 设置作业模板	119
12.22. 执行远程作业	119
12.23. 作业向导中的高级设置	122
12.24. 为主机调度重复的 ANSIBLE 作业	123
12.25. 为主机组调度重复的 ANSIBLE 作业	124

12.26. 监控作业	125
12.27. 在 CAPSULE 上设置作业速率限制	126
第 13 章 SATELLITE 中的主机状态	127
13.1. 主机全局状态概述	127
13.2. 主机子状态概述	127
第 14 章 同步模板存储库	136
14.1. 启用 TEMPLATESYNC 插件	136
14.2. 配置 TEMPLATESYNC 插件	136
14.3. 使用存储库源	137
14.4. 导入和导出模板	140
第 15 章 管理软件包	144
15.1. 在主机上启用和禁用存储库	144
15.2. 在主机上安装软件包	144
15.3. 升级主机上的软件包	147
15.4. 从主机中删除软件包	150
附录 A. 模板编写参考	153
A.1. 在 SATELLITE WEB UI 中访问编写参考的模板	153
A.2. 在模板中使用自动完成	154
A.3. 编写 ERB 模板	154
A.4. 对 ERB 模板进行故障排除	157
A.5. 通用 SATELLITE 特定 MACROS	158
A.6. TEMPLATES MACROS	158
A.7. 特定于主机的变量	162
A.8. 特定于 KICKSTART 的变量	165
A.9. 条件语句	165
A.10. PARSING ARRAYS	166
A.11. 模板片断示例	168
附录 B. 作业模板示例和扩展	171
B.1. 自定义作业模板	171
B.2. 默认任务模板类别	171
B.3. RESTORECON 模板示例	172
B.4. 渲染 RESTORECON 模板	173
B.5. 在多个主机上执行 RESTORECON 模板	173
B.6. 在模板中包括 POWER 操作	174
附录 C. HOST COLUMNS 概述	176

向红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

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

前提条件

- 确保您已注册了 [红帽帐户](#)。

流程

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

第 1 章 SATELLITE 中的主机概述

主机是红帽卫星管理的任何 Linux 客户端。主机可以是物理或虚拟的。虚拟主机可以部署到 Red Hat Satellite 支持的任何平台上，如 Amazon EC2、Google Compute Engine、KVM、libvirt、Microsoft Azure、OpenStack、Red Hat Virtualization、RACKSPACE Cloud Services 或 VMware vSphere。

Red Hat Satellite 可大规模启用主机管理，包括监控、配置、远程执行、配置管理、软件管理和订阅管理。您可以通过 Satellite Web UI 或命令行管理您的主机。

在 Satellite Web UI 中，您可以浏览 Satellite 服务器识别的所有主机，按类型分组：

- **所有主机** - Satellite 服务器可识别的所有主机的列表。
- **发现的主机** - 发现插件在 provisioning 网络中检测到的裸机主机列表。
- **Content Hosts** - 管理与内容和订阅相关的任务的主机列表。
- **Host Collections** - 用于批量操作（如勘误安装）的用户定义的主机集合列表。

要搜索主机，在 **Search** 字段中键入，并使用星号 `principal` 执行部分字符串搜索。例如，如果搜索名为 `dev-node.example.com` 的内容主机，点 **Content Hosts** 页面并在 **Search** 字段中输入 `dev-node*`。另外，`*node*` 也会查找内容主机 `dev-node.example.com`。



警告

Satellite 服务器被列为主机本身，即使它没有自助注册。不要从主机列表中删除卫星服务器。

第 2 章 管理主机

本章论述了创建、注册、管理和删除主机。

2.1. 在 RED HAT SATELLITE 中创建主机

使用这个流程在 Red Hat Satellite 中创建主机。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，点 **Hosts > Create Host**。
2. 在 **Host** 选项卡中输入所需详情。
3. 单击 **Ansible Roles** 选项卡，然后从 **Ansible Roles** 列表中选择您要添加到主机的一个或多个角色。使用 **箭头图标** 管理添加或删除的角色。
4. 在 **Puppet Classes** 选项卡中，选择您要包含的 Puppet 类。
5. 在 **Interfaces** 选项卡中：
 - a. 对于每个接口，点 **Actions** 列中的 **Edit**，并根据需要配置以下设置：
 - **使用 Type** 列表并选择接口类型，键入 `mvapich-literal` 或 `Bond` 或 `BMC` 接口。
 - **MAC 地址** `rhacm-DAEMON` Enter MAC 地址。
 - **DNS 名称** `rhacm-rhacm` Enter 是 DNS 服务器已知的 DNS 名称。这用于 FQDN 的主机部分。
 - **domain** `mvapich-DAEMON` Select 调配网络的域名。这会使用选择合适的子网自动更新 **Subnet** 列表。
 - **IPv4 Subnet** `mvapich-rhacm` select a IPv4 subnet for the host from the list.
 - **IPv6 子网** `mvapich-odbc` 从列表中为主机选择 IPv6 子网。
 - 对于子网启用了 **IPv4 地址** `rhacm-rhacm` If IP 地址管理(IPAM)，则会自动推荐 IP 地址。或者，您可以输入地址。如果启用了置备令牌，则可以省略地址；如果域不管理 DNS，如果子网不管理反向 DNS，或者子网没有管理 DHCP 保留。
 - 对于子网启用了 **IPv6 地址** `rhacm-rhacm` If IP 地址管理(IPAM)，则会自动推荐 IP 地址。或者，您可以输入地址。
 - **受管** the 复选框，以在置备过程中配置接口，以使用由 Capsule 提供的 DHCP 和 DNS 服务。
 - **主 IANA-** the Select 此复选框，将此接口的 DNS 名称用作 FQDN 的主机部分。
 - **置备 IANA-** the Select this 复选框，以使用此接口进行调配。这意味着，TFTP 引导将使用此接口，或者在基于镜像的调配时执行，该脚本将通过此界面执行调配。请注意，许多调配任务（如 `anaconda` 下载 RPM、在 `%post` 脚本中的 Puppet 设置）都将使用主接口。
 - 如果这个接口不是物理设备，则 **虚拟 NIC** `IANA-Demo` Select this 复选框。此设置有两个选项：

- 标签 rhacm-rhacmOptionally 设置了 VLAN 标签。如果未设置，标签将是子网的 VLAN ID。
 - 附加到此虚拟接口的接口的设备名称。
- b. 单击 OK 以保存接口配置。
 - c. (可选) 单击 Add Interface 来包括额外的网络接口。更多信息请参阅 [第 5 章 添加网络接口](#)。
 - d. 单击 Submit 以应用更改并退出。
6. 在 Operating System 选项卡中，输入所需的详情。对于 Red Hat 操作系统，选择 Synced Content for Media Selection。如果要使用非 Red Hat 操作系统，选择 All Media，然后从 Media Selection 列表中选择安装介质。您可以从列表选择一个分区表，或者在 Custom partition table 字段中输入自定义分区表。您不能同时指定。
 7. 在 Parameters 选项卡上，单击 Add Parameter，以添加您要在运行时传递给作业模板的任何参数变量。这包括您要为主机关联的所有 Puppet 类、Ansible playbook 参数和主机参数。要将参数变量与 Ansible 作业模板一起使用，您必须添加一个 Host Parameter。
 当您创建 Red Hat Enterprise Linux 8 主机时，您可以设置系统目的属性。system purpose 属性定义在主机创建时自动附加的订阅。在 Host Parameters 区域中，使用对应的值输入以下参数名称。有关值列表，请参阅 [执行标准 RHEL 8 安装中的系统目的简介](#)。
 - syspurpose_role
 - syspurpose_sla
 - syspurpose_usage
 - syspurpose_addons
 如果要为远程任务执行创建拉取模式的主机，请添加 enable-remote-execution-pull 参数，并将 type boolean 设置为 true。更多信息请参阅 [第 12.4 节 “远程执行的传输模式”](#)。
 8. 在 Additional Information 选项卡中，输入主机的附加信息。
 9. 单击 Submit 以完成您的客户端请求。

CLI 过程

- 要创建与主机组关联的主机，请输入以下命令：

```
# hammer host create \
--name "My_Host_Name" \
--hostgroup "My_Host_Group" \
--interface="primary=true, \
    provision=true, \
    mac=mac_address, \
    ip=ip_address" \
--organization "My_Organization" \
--location "My_Location" \
--ask-root-password yes
```

此命令提示您指定 root 密码。必须指定主机的 IP 和 MAC 地址。主网络接口的其他属性可以从主机组继承，或使用 --subnet 和 --domain 参数设置。您可以使用 --interface 选项设置附加接口，该选项接受键值对列表。如需可用接口设置列表，请输入 `hammer host create --help` 命令。

2.2. 克隆主机

您可以克隆现有主机。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > All Hosts。
2. 在 Actions 菜单中，单击 Clone。
3. 在 Host 选项卡上，确保提供与原始主机不同的名称。
4. 在 Interfaces 选项卡上，确保提供不同的 IP 地址。
5. 点 Submit 以克隆主机。

更多信息请参阅 [第 2.1 节 “在 Red Hat Satellite 中创建主机”](#)。

2.3. 从 HYPERVISOR 将虚拟机与 SATELLITE 关联

流程

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources。
2. 选择计算资源。
3. 在 Virtual Machines 选项卡中，从 Actions 菜单中点 关联虚拟机。

2.4. 编辑主机的系统目的

您可以编辑 Red Hat Enterprise Linux 主机的系统目的属性。系统目的允许您在网络上设置系统的预期使用，并改进 Red Hat Hybrid Cloud Console 订阅服务中的报告准确性。有关系统目的的更多信息，请参阅 [执行标准的 RHEL 8 安装中的系统目的简介](#)。

前提条件

- 您要编辑的主机必须在 subscription-manager 中注册。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > All Hosts。
2. 点您要修改的主机的名称。
3. 在 Overview 选项卡中，点 System purpose 卡上的 Edit。
4. 为主机选择系统目的属性。
5. 点 Save。

CLI 过程

1. 登录主机并编辑所需的系统目的属性。例如，将 usage 类型设置为 **Production**，角色设置为 **Red Hat Enterprise Linux Server**，并添加附加组件添加。有关值列表，请参阅 [执行标准 RHEL](#)

8 安装中的 系统目的简介。

```
# subscription-manager syspurpose set usage 'Production'  
# subscription-manager syspurpose set role 'Red Hat Enterprise Linux Server'  
# subscription-manager syspurpose add addons 'your_addon'
```

2. 验证此主机的系统目的属性：

```
# subscription-manager syspurpose
```

3. 自动将订阅附加到此主机：

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

4. 验证此主机的系统目的状态：

```
# subscription-manager status
```

2.5. 编辑多个主机的系统目的

您可以编辑 Red Hat Enterprise Linux 主机的系统目的属性。系统用途属性定义要自动附加到主机的订阅。有关系统目的的更多信息，请参阅 *执行标准的 RHEL 8 安装* 中的 [系统目的简介](#)。

前提条件

- 您要编辑的主机必须在 `subscription-manager` 中注册。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Content Hosts** 并选择您要编辑的 Red Hat Enterprise Linux 8 主机。
2. 单击 **Select Action** 列表，再选择 **Manage System Purpose**。
3. 选择您要分配给所选主机的系统目的属性。您可以选择以下值之一：

- 设置所有所选主机的特定属性。
 - 没有更改 来保持在所选主机上设置属性。
 - none (Clear) 以清除所选主机上的属性。
4. 单击 **分配**。
 5. 在 **Satellite Web UI** 中，导航到 **Hosts > Content Hosts** 并选择相同的 **Red Hat Enterprise Linux 8** 主机来根据系统目的自动附加订阅。
 6. 单击 **Select Action** 列表，再选择 **Manage Subscriptions**。
 7. 单击 **Auto-Attach**，以根据其系统角色自动将订阅附加到所有选定的主机。

2.6. 更改主机的模块流

如果您的主机正在运行 **Red Hat Enterprise Linux 8**，您可以修改您安装的仓库的模块流。

您可以在 **Satellite Web UI** 中启用、禁用、安装、更新和删除模块流。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 点 **Content** 选项卡，然后点 **模块流** 选项卡。
4. 点模块旁边的垂直 **ellipsis** 并选择您要执行的操作。远程执行作业完成后，您会收到 **REX 作业通知**。

2.7. 创建主机组

如果您创建大量主机，则许多主机都可以具有通用的设置和属性。为每个新主机添加这些设置和属性是耗时的。如果使用主机组，您可以将通用属性应用到您创建的主机。

主机组充当常见主机设置的模板，包含您提供给主机的许多相同详情。当您使用主机组创建主机时，主机将从主机组继承定义的设置。然后，您可以提供额外的详细信息来个性化主机。

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

主机组层次结构

您可以创建主机组的层次结构。目的是提供一个基本级别主机组，它代表您组织中的所有主机并提供常规设置，然后嵌套组来提供特定的设置。例如，您可以有一个基本级别主机组来定义操作系统，以及继承基本级别主机组的两个嵌套主机组：

- **hostgroup: Base (Red Hat Enterprise Linux 8.8)**
 - **hostgroup: Webserver (应用 nginx Puppet 类)**
 - 主机：webserver1.example.com (Web 服务器)
 - 主机：webserver2.example.com (Web 服务器)
 - **hostgroup：存储 (适用于 nfs Puppet 类)**
 - 主机：storage1.example.com (存储服务器)
 - 主机：storage2.example.com (存储服务器)
 - 主机：custom.example.com (自定义主机)

在本例中，所有主机都使用 Red Hat Enterprise Linux 7.6 作为其操作系统，因为它们对基本主机组的继承。两个 Web 服务器主机继承 Webserver 主机组中的设置，其中包括 nginx Puppet 类和 Base 主机组中的设置。这两个存储服务器从 Storage 主机组继承设置，其中包括 nfs Puppet 类以及 Base 主机组中的设置。自定义主机仅继承 Base 主机组中的设置。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Host Groups**，再点 **Create Host Group**。
2. 如果您有一个要从中继承属性的现有主机组，您可以从 **Parent** 列表选择一个主机组。如果没有，请将此字段留空。
3. 输入新主机组的名称。
4. 输入您希望将来的主机继承的任何其他信息。
5. 单击 **Ansible Roles** 选项卡，然后从 **Ansible Roles** 列表中选择您要添加到主机的一个或多个角色。使用 **箭头图标** 管理添加或删除的角色。
6. 单击 **附加选项卡**，并将您要属性的任何详细信息添加到主机组中。

注意

Puppet 在将主机注册到与 **Production** 环境中创建的 Puppet 环境关联的主机组时，无法检索 Puppet CA 证书。

要创建与主机组关联的适当 Puppet 环境，请手动创建目录：

```
# mkdir /etc/puppetlabs/code/environments/example_environment
```

7. 单击 **Submit** 以保存主机组。

CLI 过程

- 使用 `hammer hostgroup create` 命令创建主机组。例如：

```
# hammer hostgroup create --name "Base" \
--architecture "My_Architecture" \
--content-source-id _My_Content_Source_ID_ \
--content-view "_My_Content_View_" \
--domain "_My_Domain_" \
--lifecycle-environment "_My_Lifecycle_Environment_" \
--locations "_My_Location_" \
--medium-id _My_Installation_Medium_ID_ \
--operatingsystem "_My_Operating_System_" \
--organizations "_My_Organization_" \
--partition-table "_My_Partition_Table_" \
--puppet-ca-proxy-id _My_Puppet_CA_Proxy_ID_ \
--puppet-environment "_My_Puppet_Environment_" \
--puppet-proxy-id _My_Puppet_Proxy_ID_ \
--root-pass "My_Password" \
--subnet "_My_Subnet_"
```

2.8. 为每个生命周期环境创建主机组

使用这个流程为 **Library** 生命周期环境创建主机组，并为其他生命周期环境添加嵌套的主机组。

流程

要为每个生命周期环境创建主机组，请运行以下 **Bash** 脚本：

```
MAJOR="My_Major_OS_Version"
ARCH="My_Architecture"
ORG="My_Organization"
LOCATIONS="My_Location"
PTABLE_NAME="My_Partition_Table"
DOMAIN="My_Domain"

hammer --output csv --no-headers lifecycle-environment list --organization "${ORG}" | cut -d
',' -f 2 | while read LC_ENV; do
[[ ${LC_ENV} == "Library" ]] && continue

hammer hostgroup create --name "rhel-${MAJOR}server-${ARCH}-${LC_ENV}" \
--architecture "${ARCH}" \
--partition-table "${PTABLE_NAME}" \
--domain "${DOMAIN}" \
--organizations "${ORG}" \
--query-organization "${ORG}" \
--locations "${LOCATIONS}" \
--lifecycle-environment "${LC_ENV}"
done
```

2.9. 将主机添加到主机组

您可以在 **Satellite Web UI** 中将主机添加到主机组。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 点 **Edit** 按钮。
4. 从 **Host Group** 列表中选择主机组。
5. 点 **Submit**。

验证

- **Overview** 选项卡下的 **Details** 卡现在显示您的主机所属的主机组。

2.10. 更改主机的主机组

使用这个流程更改主机的主机组。

如果在更改主机组后重新置备主机，则主机从主机组继承的新值将被应用。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。

3. 点 **Edit** 按钮。
4. 从 **Host Group** 列表中选择新的主机组。
5. 点 **Submit**。

验证

- **Overview** 选项卡下的 **Details** 卡现在显示您的主机所属的主机组。

2.11. 将主机添加到主机集合

您可以在 **Satellite Web UI** 中将主机添加到主机组中。

前提条件

主机必须注册到 **Red Hat Satellite**，才能将其添加到主机集合。有关注册主机的详情，请参考 [第 3.3 节“使用全局注册注册主机”](#)。

请注意，如果您将主机添加到主机集合，则 **Satellite** 审计系统不会记录更改。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 在 **Host collections** 卡中，点垂直 **ellipsis** 并选择 **Add host to collections**。
4. 选择主机变量。
5. 点 **Add**。

CLI 过程

- 要将主机添加到主机组中，请输入以下命令：

```
# hammer host-collection add-host \  
--host-ids My_Host_ID_1 \  
--id My_Host_Collection_ID
```

2.12. 使用主机详情选项卡

在 **Satellite** 中，您可以在 **Details** 选项卡中查看主机名的详情。您可以展开和折叠单个卡和所有链接。您的浏览器记住卡扩展和折叠状态。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要查看的主机的名称。
3. 选择 **Details** 选项卡。

Details 选项卡中的卡显示 系统属性、BIOS、网络接口、操作系统、置备模板 的详细信息。注册的内容主机显示额外的卡，用于注册详情、安装的产品和 HW 属性，提供有关 模型、CPU 数量、插槽、每个插槽和 RAM 的信息。

在 **操作系统** 卡中，您可以查看 架构、操作系统、引导时间和内核 发行版本 的详细信息。

以下详情卡有交互式功能：

网络接口

1. 单击以折叠并展开每个网络接口。
2. 点链接编辑所有网络接口。

系统属性

1. 点击以将值复制到剪贴板，包括 **Name**、**Subscription UUID** 和 **Domain**。
2. 有关具有虚拟客户机的主机，点芯片来查看客户机列表。
3. 对于作为虚拟客户机的主机，请单击 **Virtual host** 链接来查看其主机。

自定义模板

1. 点击以在弹出窗口中查看模板，而不离开页面。
2. 单击铅笔图标以编辑模板。
3. 单击模态的弹出按钮，以在新标签页中查看模板。
4. 点模态链接编辑模板。
5. 点模态中的 *Copy to clipboard* 按钮，将模板提取到剪贴板中。

2.13. 更改主机的内容源

内容源是主机消耗内容的 **Capsule**。使用这个流程更改主机的内容源。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 单击 **Edit** 按钮旁边的垂直 **ellipsis** 图标，然后选择 **Change content source**。

4. 从列表中选择 **Environment**、**Content View** 和 **Content Source**。
5. 点 **Change content source**。

您可以使用远程执行或手动完成内容源更改。要使用远程执行更新主机上的配置，请单击 **Run job call**。有关运行远程执行作业的更多信息，请参阅 [配置和设置远程作业](#)。要手动更新内容源，请从主机上的 **更改内容源** 执行自动生成的命令。

2.14. 更改主机的环境

使用这个流程更改主机的环境。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 单击 **内容视图详细信息** 卡中的垂直 **ellipsis**，然后选择 **Edit content view assignment**。
4. 选择环境。
5. 选择内容视图。
6. 点 **Save**。

2.15. 更改主机的受管状态

默认情况下，由 **Satellite** 置备的主机是 **Managed**。当主机设置为 **Managed** 时，您可以从 **Satellite** 服务器中配置其他主机参数。这些附加参数在 **Operating System** 标签页中列出。如果您更改了 **Operating System** 选项卡上的任何设置，则在将主机设置为构建并重启后才会生效。

如果您需要使用 **Satellite** 不支持的操作系统获取系统的配置管理报告，请将主机设置为 **Unmanaged**。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 点 **Edit** 按钮。
4. 点 **Manage host** 或 **Unmanage host** 以更改主机的状态。
5. 点 **Submit**。

2.16. 在主机上启用 TRACER

使用这个流程在 **Satellite** 上启用 **Tracer** 并访问 **Traces**。**tracer** 显示需要重启的服务和应用程序列表。**trace** 是 **Satellite Web UI** 中 **tracer** 生成的输出。

前提条件

- **Satellite Client 6** 软件仓库已同步
- **Satellite Client 6** 存储库在内容视图和主机的生命周期阶段中可用
- 为主机启用 **Satellite 客户端 6** 存储库
- 启用远程执行

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 点 **Traces** 选项卡，然后点 **Enable Traces** 按钮。
4. 从列表中选择 **katello-host-tools-tracer** 的供应商。
5. 点 **Enable Tracer** 按钮。远程执行作业完成后，您会收到 **REX** 作业通知。

2.17. 在主机上重启应用程序

使用这个流程从 Satellite Web UI 重新启动应用程序。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 选择 **Traces** 选项卡。
4. 选择您要重启的应用程序。
5. 从 **Restart app** 列表中选择 **Restart via remote execution**。远程执行作业完成后，您将收到 **REX** 作业通知。

2.18. 将主机分配给特定机构

使用这个流程将主机分配给特定的机构。有关机构的一般信息以及如何配置它们，[请参阅管理 Red Hat Satellite 中的管理机构](#)。

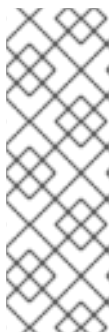


注意

如果您的主机已经注册到不同的机构，您必须首先取消注册该主机，然后才能将其分配给新机构。要取消主机注册，请在主机上运行 `subscription-manager unregister`。将主机分配给新机构后，您可以重新注册主机。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 选中您要更改的主机的复选框。
3. 从 **Select Action** 列表中，选择 **Assign Organization**。此时会打开一个新选项窗口。
4. 在 **Select Organization** 列表中，选择要分配主机的组织。选中 **Fix Organization on Mismatch**。



注意

如果资源（如域或子网）与主机关联的资源（如域或子网）没有与您要分配主机的机构关联，则会发生不匹配。选项 **Fix Organization on Mismatch** 将向机构添加这样的资源，因此是推荐的选择。选项 **Fail on Mismatch** 将始终产生错误消息。例如，将主机从一个机构重新分配给另一个机构将失败，即使设置中没有实际不匹配。

5. 点 **Submit**。

2.19. 将主机分配给特定位置

使用这个流程将主机分配给特定位置。有关位置以及如何配置它们的常规信息，请参阅 [管理内容](#) 中的 [创建位置](#)。

流程

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

2. 选中您要更改的主机的复选框。
3. 从 **Select Action** 列表中，选择 **Assign Location**。此时会打开一个新选项窗口。
4. 导航到 **Select Location** 列表，再选择您要主机的位置。选中 **Fix Location on Mismatch**。



注意

如果资源（如域或子网）与主机关联的资源（如域或子网）没有与您要分配主机的位置关联，则发生不匹配。选项 **修复位置 on Mismatch** 将这样的资源添加到该位置，因此是推荐的选择。选项 **Fail on Mismatch** 将始终产生错误消息。例如，将主机从一个位置重新分配给另一个位置将失败，即使设置中没有实际不匹配。

5. 点 **Submit**。

2.20. 在主机间切换

当您在 **Satellite Web UI** 中的特定主机上时，您可以在主机之间导航，而不必使用主机切换器离开页面。点主机名旁边的 **Demo**。这会按字母顺序显示主机列表，其中带有分页箭头和一个搜索栏来查找您要查找的主机。

2.21. 从内容主机查看主机详情

使用这个流程从内容主机查看主机详情页面。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Content Hosts**
2. 点击您要查看的内容主机。
3. 选择 **Details** 选项卡来查看主机详情页面。

Details 选项卡中的卡显示系统属性、BIOS、网络接口、操作系统、置备模板的详细信息。注册的内容主机显示额外的卡，用于注册详情、安装的产品和 HW 属性，提供有关模型、CPU 数量、插槽、每个插槽和 RAM 的信息。

2.22. 选择 HOST COLUMNS

您可以在 **Hosts > All Hosts** 页面的主机表中选择要查看哪些列。有关主机列的完整列表，请参阅 [附录 C, Host Columns 概述](#)。



注意

无法取消选择 **Name** 列。Name 列充当主机的主要标识方法。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 单击 **Manage** 列。
3. 选择您要显示的列。您可以选择单独的列或列类别。选择或取消选择类别选择或取消选择该类别中的所有列。



注意

有些列包含在多个类别中，但您只能显示特定类型的列。通过选择或取消选择或取消选择或取消选择该列的所有实例，即可选择或取消选择。

验证

- 现在，您可以在 **host** 表中看到所选列。

2.23. 从 SATELLITE 中删除主机

使用这个流程从 Satellite 中删除主机。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 **Hosts > All Hosts** 或 **Hosts > Content Hosts**。请注意，从 **All Hosts** 或 **Content Hosts** 中删除主机的页面没有区别。在这两种情况下，Satellite 都会完全删除主机。
2. 选择您要删除的主机。
3. 从 **Select Action** 列表中，选择 **Delete Hosts**。
4. 单击 **Submit**，以从 Satellite 永久删除主机。



警告

默认情况下，**host delete** 设置中的 **Destroy associated 虚拟机** 被设为 **no**。如果删除了与虚拟机关联的主机记录，则虚拟机将保留在计算资源中。

要删除计算资源上的虚拟机，请导航到 **Administer > Settings** 并选择 **Provisioning** 选项卡。如果在 **host delete** 上将 **Destroy associated 虚拟机** 设置为 **yes**，如果删除了与虚拟机关联的主机记录，则删除虚拟机。为了避免在这种情况下删除虚拟机，请在不将其从计算资源或更改设置的情况下将虚拟机从 Satellite 中解除关联。

CLI 过程

- 从 Satellite 中删除您的主机：

```
$ hammer host delete \
--id My_Host_ID \
--location-id My_Location_ID \
--organization-id My_Organization_ID
```

或者，您可以使用 **--name My_Host_Name** 而不是 **--id My_Host_ID**。

2.23.1. 在不将其从 Hypervisor 中删除的情况下，从 Satellite 中解除虚拟机关联

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**，再选中要解除关联的主机左侧的复选框。
2. 在 **Select Action** 列表中，选择 **Disassociate Hosts** 按钮。
3. 可选：选择复选框来保留主机以备将来操作。
4. 点 **Submit**。

第 3 章 注册主机和设置主机集成

您必须注册尚未通过 **Satellite** 置备的主机，以便使用 **Satellite** 管理它们。您可以通过 **Satellite** 服务器或 **Capsule** 服务器注册主机。

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](#)。

您还必须在主机上安装和配置工具，具体取决于您要使用的集成功能。使用以下步骤安装和配置主机工具：

- [第 3.5 节 “安装 Katello 代理”](#)
- [第 3.6 节 “安装 Tracer”](#)
- [第 3.7 节 “在主机注册过程中安装和配置 Puppet 代理”](#)
- [第 3.8 节 “手动安装和配置 Puppet 代理”](#)

3.1. 注册中支持的客户端

Satellite 支持以下用于注册的操作系统和架构：

支持的主机操作系统

主机可以使用以下操作系统：

- **Red Hat Enterprise Linux 9、8、7**
- 带有 **ELS 附加组件**的 **Red Hat Enterprise Linux 6**
- 您可以注册以下主机以转换为 **RHEL**：

- **CentOS Linux 7**
- **Oracle Linux 7 和 8**

支持的主机架构

主机可以使用以下架构：

- **i386**
- **x86_64**
- **s390x**
- **ppc_64**

3.2. 注册方法

您可以使用以下方法将主机注册到 Satellite：

全局注册

您可以从 Satellite 生成 `curl` 命令，并从无限数量的主机运行这个命令，以通过 Satellite API 使用置备模板注册它们。如需更多信息，请参阅 [第 3.3 节 “使用全局注册注册主机”](#)。

通过使用此方法，您还可以在注册到 Satellite 的过程中将 Satellite SSH 密钥部署到主机，以便为远程执行作业启用主机。更多信息请参阅 [第 12 章 配置和设置远程作业](#)。

通过使用此方法，您还可以在注册到 Satellite 的过程中使用 Red Hat Insights 配置主机。如需更多信息，请参阅 [第 9.1 节 “在 Satellite 中将 Red Hat Insights 与主机搭配使用”](#)。

(已弃用) Katello CA Consumer

您可以从 satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm 下载并安装使用

者 RPM，然后运行 `subscription-manager`。

(已弃用) Bootstrap 脚本

您可以从主机上的 `satellite.example.com/pub/bootstrap.py` 下载 bootstrap 脚本，然后运行该脚本。如需更多信息，请参阅 [第 3.4 节“使用启动脚本注册主机”](#)。

3.3. 使用全局注册注册主机

您可以通过在 Satellite 上生成 `curl` 命令并在主机上运行此命令，将主机注册到 Satellite。此方法使用两个置备模板：`Global Registration template` 和 `Linux host_init_config` 默认模板。这可让您完全控制主机注册过程。

如果需要更大的灵活性，您还可以自定义默认模板。如需更多信息，请参阅 [第 3.3.3 节“自定义注册模板”](#)。

3.3.1. 用于注册的全局参数

您可以通过导航到 `Configure > Global Parameters` 来配置以下全局参数：

- `host_registration_insights` 参数在 `insights` 代码段中使用。如果参数设为 `true`，则注册会在主机上安装并启用 Red Hat Insights 客户端。如果参数设置为 `false`，它会防止 Satellite 和 Red Hat Insights 客户端将清单报告上传到 Red Hat Hybrid Cloud Console。默认值为 `true`。在覆盖参数值时，将参数类型设置为布尔值。

- `host_packages` 参数用于在主机上安装软件包。

- `host_registration_remote_execution` 参数在 `remote_execution_ssh_keys` 代码段中使用。如果设置为 `true`，则注册会在主机上启用远程执行。默认值为 `true`。

- `remote_execution_ssh_keys`, `remote_execution_ssh_user`, `remote_execution_create_user`, 和 `remote_execution_effective_user_method` 参数在 `remote_execution_ssh_keys` 代码片段中使用。如需了解更多详细信息，请参阅代码片段。

您可以通过 **Hosts > Templates > Provisioning Templates** 进入 Satellite web UI 中的代码片段。

3.3.2. 注册主机

您可以使用注册模板注册主机，并在注册过程中设置各种集成功能和主机工具。

前提条件

- 您的用户帐户分配了 `create_hosts` 权限的角色。
- 您必须在要注册的主机上具有 `root` 权限。
- **Satellite 服务器、任何胶囊服务器和所有主机都必须与同一 NTP 服务器同步，并且启用了时间同步工具并运行。**
- 激活密钥必须可供主机使用。如需更多信息，[请参阅管理内容中的管理激活码](#)。
- **可选：如果要注册主机到 Red Hat Insights，您必须同步 `rhel-8-for-x86_64-baseos-rpms` 和 `rhel-8-for-x86_64-appstream-rpms` 存储库，并在您使用的激活码中提供它们。这是在主机上安装 `insights-client` 软件包所必需的。**
- **如果要使用 Capsule 服务器而不是 Satellite 服务器，请确保已相应地配置了 Capsule 服务器。如需更多信息，请参阅 [安装 Capsule 服务器](#) 中的 [为主机注册配置 Capsule](#) 和 [置备](#)。**
- **如果您的 Satellite 服务器或 Capsule 服务器位于 HTTP 代理后面，请将主机上的订阅管理器配置为使用 HTTP 代理进行连接。如需更多信息，请参阅红帽知识库中的 [如何通过防火墙或代理访问 Red Hat Subscription Manager \(RHSM\)](#)。**

流程

1. 在 Satellite Web UI 中，[导航到 Hosts > Register Host](#)。
2. **可选：选择不同的机构。**

3. 可选：选择不同的位置。
4. 可选：在 **Host Group** 列表中，选择要将主机与之关联的主机组。从主机组继承值的字段：操作系统、激活码和生命周期环境。

如果您的主机组分配了任何 **Ansible** 角色，则 **Ansible** 角色将在注册后在主机上运行。

5. 可选：从 **Operating system** 列表中选择您要注册的主机操作系统。
6. 可选：从 **Capsule** 列表中，选择要通过的 **Capsule** 注册主机的 **Capsule**。
7. 可选：如果您要使第一个调用不安全，请选择 **Insecure** 选项。在此第一次调用期间，主机从 **Satellite** 下载 **CA** 文件。主机将使用此 **CA** 文件与将来的所有调用连接 **Satellite**，使其安全。

红帽建议您避免不安全的调用。

如果攻击者位于 **Satellite** 和主机之间的网络中，从第一个不安全的调用中获取 **CA** 文件，攻击者可以访问 **API** 调用的内容，并从注册的主机和 **JSON Web 令牌(JWT)** 获取 **API** 调用的内容。因此，如果您已选择在注册期间部署 **SSH** 密钥，攻击者就可以使用 **SSH** 密钥访问主机。

相反，您可以在注册主机前在每个主机上手动复制并安装 **CA** 文件。

要做到这一点，找到 **Satellite** 通过导航到 **Administer > Settings > Authentication** 并查找 **SSL CA** 文件设置的值来存储 **CA** 文件的位置。

将 **CA** 文件复制到主机上的 `/etc/pki/ca-trust/source/anchors/` 目录中，并输入以下命令：

```
# update-ca-trust enable
# update-ca-trust
```

然后，使用安全 **curl** 命令注册主机，例如：

```
# curl -sS https://satellite.example.com/register ...
```

以下是带 `--insecure` 选项的 `curl` 命令示例：

```
# curl -sS --insecure https://satellite.example.com/register ...
```

8.

选择 **Advanced** 选项卡。

9.

从 **Setup REX** 列表中，选择是否要将 **Satellite SSH 密钥部署到主机**。

如果设置为 **Yes**，则会在注册的主机上安装公共 **SSH 密钥**。继承的值基于 `host_registration_remote_execution` 参数。它可以继承，例如从主机组、操作系统或机构中继承。覆盖后，所选值将存储在 `host` 参数级别上。

10.

从 **Setup Insights** 列表中，选择是否要安装 `insights-client` 并将主机注册到 **Insights**。

Insights 工具仅适用于 **Red Hat Enterprise Linux**。它对其他操作系统没有影响。

您必须在注册的机器中启用以下软件仓库：

- **Red Hat Enterprise Linux 6: rhel-6-server-rpms**
- **Red Hat Enterprise Linux 7: rhel-7-server-rpms**
- **Red Hat Enterprise Linux 8: rhel-8-for-x86_64-appstream-rpms**

`insights-client` 软件包默认安装在 **Red Hat Enterprise Linux 8** 上，除非使用 **"Minimal Install"** 选项部署 **Red Hat Enterprise Linux 8** 的环境中。

11.

可选：在 **Install packages** 字段中，列出您要在注册后在主机上安装的软件包（以空格分开）。这可以通过 `host_packages` 参数设置。

12.

可选：选择 **Update packages** 选项，以在注册后更新主机上的所有软件包。这可以通过 `host_update_packages` 参数设置。

13.

可选：在 **Repository** 字段中，输入在执行注册前要添加的存储库。例如，使 **subscription-manager** 软件包可用于注册目的非常有用。对于 Red Hat 系列发行版本，请输入存储库的 URL，例如 `http://rpm.example.com/`。
14.

可选：在 **Repository GPG 密钥 URL** 字段中，指定公钥来验证 GPG 签名软件包的签名。它需要在 ASCII 表单中使用 GPG 公钥标头指定。
15.

可选：在 **Token 生命周期（小时）** 字段中，更改 Satellite 用于身份验证的 JSON Web Token (JWT) 的有效性持续时间。此令牌的持续时间定义了生成的 curl 命令的工作方式。您可以将持续时间设置为 039) -114999 999 小时或无限。

请注意，Satellite 应用生成 curl 命令的用户的权限到主机授权。如果用户丢失或获得其他权限，则 JWT 的权限也会改变。因此，不要删除、块或更改用户在令牌期间的权限。

JWT 范围仅限于注册端点，无法在其它任何位置使用。
16.

可选：在 **Remote Execution Interface** 字段中输入主机必须用于 SSH 连接的网络接口标识符。如果将此字段留空，Satellite 将使用默认网络接口。
17.

从 REX pull 模式 列表中，选择是否要部署 Satellite 远程执行拉取客户端。

如果设置为 Yes，则远程执行拉取客户端会在注册的主机上安装。继承的值基于 `host_registration_remote_execution_pull` 参数。它可以继承，例如从主机组、操作系统或机构中继承。覆盖后，所选值将存储在 host 参数级别。

注册的主机必须有权访问 Red Hat Satellite Client 6 存储库。

有关拉取模式的详情，请参考 [第 12.4 节“远程执行的传输模式”](#)。
18.

在 **Activation Keys** 字段中输入要分配给主机的一个或多个激活码。
19.

可选：选择 生命周期阶段。

20.

可选：如果要忽略订阅管理器错误，请选择 **Ignore error** 选项。

21.

可选：如果您要在注册前删除任何 `katello-ca-consumer rpms`，并使用 `--force` 参数运行 `subscription-manager`，请选择 **Force** 选项。

22.

点 **Generate** 按钮。

23.

复制生成的 `curl` 命令。

24.

在您要注册的主机上，以 `root` 用户身份运行 `curl` 命令。

3.3.3. 自定义注册模板

您可以通过编辑置备模板来自定义注册过程。请注意，**Satellite** 中的所有默认模板都将被锁定。如果要自定义注册模板，您必须克隆默认模板并编辑克隆。



注意

红帽只提供对原始未编辑的模板的支持。自定义模板不会收到红帽发布的更新。

注册过程使用以下置备模板：

- **Global Registration** 模板包含将主机注册到 **Satellite** 的步骤。此模板会在主机访问 `/register Satellite API` 端点时呈现。
- **Linux host_init_config** 默认模板包含在注册主机后初始配置主机的步骤。

流程

1.

导航到 **Hosts > Templates > Provisioning Templates**。

2. 搜索您要编辑的模板。
3. 在所需模板的行中，单击 **Clone**。
4. 根据需要编辑模板。更多信息请参阅 [附录 A, 模板编写参考](#)。
5. 点 **Submit**。
6. 导航到 **Administer > Settings > Provisioning**。
7. 根据需要更改以下设置：
 - 将 **Default Global registration template** 设置指向您的自定义全局注册模板，
 - 将 **Default 'Host initial configuration'** 模板 设置为您的自定义初始配置模板。

3.4. 使用启动脚本注册主机

弃用了使用 [第 3.3 节 “使用全局注册注册主机”](#) 替代。

使用 **bootstrap** 脚本自动执行内容注册和 **Puppet** 配置。您可以使用 **bootstrap** 脚本注册新主机，或者从 **RHN**、**SAM**、**RHSM** 或其他 **Red Hat Satellite** 实例迁移现有主机。

katello-client-bootstrap 软件包默认安装在卫星服务器的基本操作系统中。**bootstrap.py** 脚本安装在 **/var/www/html/pub/** 目录中，使其可用于 **satellite.example.com/pub/bootstrap.py** 的主机。该脚本包含 **/usr/share/doc/katello-client-bootstrap-版本/README.md** 文件中的文档。

要使用 **bootstrap** 脚本，您必须在主机上安装它。因为脚本只需要一次，且只适用于 **root** 用户，所以您可以将其放在 **/root** 或 **/usr/local/sbin** 中，并在使用后将其删除。这个过程使用 **/root**。

前提条件

- 您有运行 `bootstrap` 脚本所需的权限的 `Satellite` 用户。此流程中的示例指定 `admin` 用户。如果您的安全策略无法接受，请创建一个具有所需最小权限的新角色，并将其添加到将要运行脚本的用户。更多信息请参阅 [第 3.4.1 节“为启动脚本设置权限”](#)。
- 为启用了 `Satellite` 客户端 6 存储库的主机有一个激活码。有关配置激活码的详情，请参考 [管理内容中的管理激活码](#)。
- 您已创建了主机组。有关创建主机组的详情，请参考 [第 2.7 节“创建主机组”](#)。

Puppet 注意事项

如果主机组与 `Production` 环境中创建的 `Puppet` 环境关联，则 `Puppet` 在从该主机组注册主机时无法检索 `Puppet CA` 证书。

要创建与主机组关联的适当 `Puppet` 环境，请按照以下步骤操作：

1. 手动创建目录：

```
# mkdir /etc/puppetlabs/code/environments/example_environment
```

2. 在 `Satellite Web UI` 中，导航到 `Configure > Environments`，再点 `Import environment from`。按钮名称包括内部或外部胶囊的 `FQDN`。
3. 选择创建的目录，再单击 `Update`。

流程

1. 以 `root` 用户身份登录主机。

2. 下载脚本：

```
# curl -O http://satellite.example.com/pub/bootstrap.py
```

3. 使脚本可执行：


```
# chmod +x bootstrap.py
```

4. 通过查看帮助文本确认脚本可以执行：

- 在 Red Hat Enterprise Linux 8 中：

```
# /usr/libexec/platform-python bootstrap.py -h
```

- 在其他 Red Hat Enterprise Linux 版本中：

```
# ./bootstrap.py -h
```

5. 输入 `bootstrap` 命令以及适合您的环境的值。

对于 `--server` 选项，指定 Satellite Server 或 Capsule Server 的 FQDN。对于 `--location`、`--organization` 和 `--hostgroup` 选项，使用 quoted 名称，而非标签，作为选项的参数。有关高级用例，请参阅 [第 3.4.2 节“高级启动脚本配置”](#)。

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \
--login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key"
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# ./bootstrap.py --login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key"
```

6. 输入您通过 `--login` 选项指定的 Satellite 用户的密码。

该脚本向 `stdout` 发送进度通知。

7. 在脚本提示时，批准主机的 Puppet 证书。在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**，并找到您使用 `--server` 选项指定的 Satellite 或 Capsule 服务器。
8. 从 **Actions** 列中的列表，选择 证书。
9. 在 **Actions** 列中，单击 **Sign** 以批准主机的 Puppet 证书。
10. 返回到主机，以查看 `bootstrap` 过程完成的其余部分。
11. 在 Satellite Web UI 中，进入到 **Hosts > All Hosts**，并确保主机连接到正确的主机组。
12. 可选：在主机注册完成后，删除脚本：

```
# rm bootstrap.py
```

3.4.1. 为启动脚本设置权限

使用这个流程配置 Satellite 用户，该用户具有运行 `bootstrap` 脚本所需的权限。要使用 CLI 而不是 Satellite Web UI，请参阅 [CLI 过程](#)。

流程

1. 在 Satellite Web UI 中，进入到 **Administer > Users**。
2. 点所需的 **Username** 来选择现有用户。此时会打开一个新窗格，其中标签页可以修改所选用户的信息。或者，创建一个专门用于运行此脚本的新用户。
3. 单击 **Roles** 选项卡。
4. 从 **Roles** 列表中选择 **Edit hosts and Viewer**。

重要

通过 **Edit hosts** 角色，用户可以编辑和删除主机，并且能够添加主机。如果您的安全策略不接受此，请创建一个具有以下权限的新角色，并将其分配给用户：

- **view_organizations**
- **view_locations**
- **view_domains**
- **view_hostgroups**
- **view_hosts**
- **view_architectures**
- **view_ptables**
- **view_operatingsystems**
- **create_hosts**

5. 点 **Submit**。

CLI 过程

1. 创建具有 **bootstrap** 脚本所需最小权限的角色。这个示例创建一个名为 **Bootstrap** 的角色：

```
# ROLE='Bootstrap'
hammer role create --name "$ROLE"
hammer filter create --role "$ROLE" --permissions view_organizations
```

```
hammer filter create --role "$ROLE" --permissions view_locations
hammer filter create --role "$ROLE" --permissions view_domains
hammer filter create --role "$ROLE" --permissions view_hostgroups
hammer filter create --role "$ROLE" --permissions view_hosts
hammer filter create --role "$ROLE" --permissions view_architectures
hammer filter create --role "$ROLE" --permissions view_ptables
hammer filter create --role "$ROLE" --permissions view_operatingsystems
hammer filter create --role "$ROLE" --permissions create_hosts
```

2.

为现有用户分配新角色：

```
# hammer user add-role --id user_id --role Bootstrap
```

或者，您也可以创建新用户，并为他们分配这个新角色。有关使用 Hammer 创建用户的更多信息，请参阅[管理 Red Hat Satellite 中的管理用户和角色](#)。

3.4.2. 高级启动脚本配置

本节有更多示例来使用 `bootstrap` 脚本注册或迁移主机。



警告

这些示例指定 `admin` Satellite 用户。如果安全策略无法接受，请创建一个具有 `bootstrap` 脚本所需最小权限的新角色。更多信息请参阅 [第 3.4.1 节“为启动脚本设置权限”](#)。

3.4.2.1. 将主机从一个 Satellite 迁移到另一个 Satellite

使用带有 `--force` 的脚本从旧的 Satellite 中删除 `katello-ca-consumer-*` 软件包，并在新 Satellite 上安装 `katello-ca-consumer-*` 软件包。

流程

-

在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \
--login=admin \
```

```
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key" \
--force
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key" \
--force
```

3.4.2.2. 将主机从红帽网络(RHN)或 Satellite 5 迁移到 Satellite

bootstrap 脚本会检测 `/etc/syconfig/rhn/systemid` 的存在以及与 RHN 的有效连接，来指示系统已注册到旧平台。然后，脚本会调用 `rhn-classic-migrate-to-rhsm` 将系统从 RHN 迁移。默认情况下，由于审计原因，脚本不会删除系统的传统配置集。若要移除传统配置文件，可使用 `--legacy-purge`，并使用 `--legacy-login` 提供有权删除配置文件的用户帐户。提示时输入用户帐户密码。

流程

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \
--login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key" \
--legacy-purge \
--legacy-login rhn-user
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
```

```
--activationkey="My_Activation_Key" \  
--legacy-purge \  
--legacy-login rhn-user
```

3.4.2.3. 在没有 Puppet 的情况下将主机注册到 Satellite

默认情况下，`bootstrap` 脚本为内容管理和配置管理配置主机。如果您有一个现有的配置管理系统，且不想在主机上安装 Puppet，请使用 `--skip-puppet`。

流程

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \  
--login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--skip-puppet
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--skip-puppet
```

3.4.2.4. 仅将主机注册到 Satellite 以进行内容管理

要将系统注册为内容主机，并省略置备和配置管理功能，请使用 `--skip-foreman`。

流程

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \  
--server satellite.example.com \  
--organization="My_Organization" \  
--skip-foreman
```

```
--activationkey="My_Activation_Key" \  
--skip-foreman
```

-

在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --server satellite.example.com \  
--organization="My_Organization" \  
--activationkey="My_Activation_Key" \  
--skip-foreman
```

3.4.2.5. 更改 Bootstrap 脚本使用的方法下载 Consumer RPM

默认情况下，bootstrap 脚本使用 HTTP 从 <http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm> 下载消费者 RPM。在某些环境中，您可能希望只允许主机和 Satellite 之间的 HTTPS。使用 `--download-method` 将下载方法从 HTTP 改为 HTTPS。

流程

-

在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \  
--login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--download-method https
```

-

在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--download-method https
```

3.4.2.6. 为 Satellite 提供主机的 IP 地址

在一个接口上有多个接口或多个 IP 地址的主机上，您可能需要覆盖 IP 地址的自动检测，并向 Satellite 提供特定的 IP 地址。使用 `--ip`。

流程

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \  
--login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--ip 192.x.x.x
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--ip 192.x.x.x
```

3.4.2.7. 在主机上启用远程执行

使用 `--rex` 和 `--rex-user` 启用远程执行，并为指定用户添加所需的 SSH 密钥。

流程

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \  
--login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--rex \  
--rex-user root
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--rex \  
--rex-user root
```



```
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--rex \  
--rex-user root
```

3.4.2.8. 在注册期间为主机创建域

若要创建主机记录，运行该脚本之前，卫星中需要存在主机的 DNS 域。如果域不存在，请使用 `--add-domain` 添加该域。

流程

- 在 Red Hat Enterprise Linux 8 中输入以下命令：

```
# /usr/libexec/platform-python bootstrap.py \  
--login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--add-domain
```

- 在 Red Hat Enterprise Linux 6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--server satellite.example.com \  
--location="My_Location" \  
--organization="My_Organization" \  
--hostgroup="My_Host_Group" \  
--activationkey="My_Activation_Key" \  
--add-domain
```

3.4.2.9. 为主机提供替代 FQDN

如果主机的主机名不是 FQDN，或者不是 RFC 兼容（包含下划线）的字符，则脚本将在主机名验证阶段失败。如果无法更新主机以使用 Satellite 接受的 FQDN，您可以使用 `bootstrap` 脚本指定替代 FQDN。

流程

1. 使用 Hammer 将 `create_new_host_when_facts_are_uploaded` 和

`create_new_host_when_report_is_uploaded` 设置为 `false` :

```
# hammer settings set \
--name create_new_host_when_facts_are_uploaded \
--value false
# hammer settings set \
--name create_new_host_when_report_is_uploaded \
--value false
```

2.

使用 `--fqdn` 指定将报告到 **Satellite** 的 **FQDN** :

- 在 **Red Hat Enterprise Linux 8** 中输入以下命令 :

```
# /usr/libexec/platform-python bootstrap.py --login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key" \
--fqdn node100.example.com
```

- 在 **Red Hat Enterprise Linux 6** 或 **7** 中, 输入以下命令 :

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="My_Location" \
--organization="My_Organization" \
--hostgroup="My_Host_Group" \
--activationkey="My_Activation_Key" \
--fqdn node100.example.com
```

3.5. 安装 KATELLO 代理

您可以安装 **Katello** 代理来远程更新 **Satellite** 客户端。



注意

Katello 代理已弃用, 并将在以后的 **Satellite** 版本中删除。迁移进程以使用远程执行功能远程更新客户端。如需更多信息, 请参阅管理主机中的 [从 Katello Agent 迁移到远程执行](#)。

katello-agent 软件包依赖于提供 **goferd** 服务的 **gofer** 软件包。

前提条件

- 您已在 **Satellite** 服务器上启用了 **Satellite** 客户端 6 存储库。如需更多信息，请参阅 [在连接的网络环境中安装 Satellite 服务器中的启用 Satellite 客户端 6 存储库](#)。
- 您已在 **Satellite** 服务器上同步了 **Satellite** 客户端 6 存储库。如需更多信息，请参阅 [在连接的网络环境中安装 Satellite 服务器中的同步 Satellite 客户端 6 存储库](#)。
- 您已在客户端上启用了 **Satellite** 客户端 6 存储库。

流程

1. 安装 **katello-agent** 软件包：

```
# dnf install katello-agent
```

2. 启动 **goferd** 服务：

```
# systemctl start goferd
```

3.6. 安装 TRACER

使用这个流程在 **Red Hat Satellite** 上安装 **Tracer** 并访问 **Traces**。**tracer** 显示已过时且需要重启的服务和应用程序列表。**trace** 是 **Satellite Web UI** 中 **tracer** 生成的输出。

前提条件

- 主机必须注册到 **Red Hat Satellite**。
- 必须在 **Satellite** 客户端 6 存储库上启用并同步，并在主机上启用。

流程

1. 在内容主机上安装 `katello-host-tools-tracer` RPM 软件包：

```
# yum install katello-host-tools-tracer
```
2. 使用以下命令：

```
# katello-tracer-upload
```
3. 在 **Satellite Web UI** 中，进入到 **Hosts > All Hosts**，然后点所需的主机名。
4. 点 **Traces** 选项卡查看 **Traces**。如果没有安装，**Enable Traces** 按钮会启动安装该软件包的远程执行作业。

3.7. 在主机注册过程中安装和配置 PUPPET 代理

您可以在注册过程中在主机上安装和配置 Puppet 代理。主机上需要配置了 Puppet 代理，以便 Puppet 与您的 Satellite 集成。有关 Puppet 的更多信息，请参阅在 [Red Hat Satellite 中使用 Puppet 集成管理配置](#)。

前提条件

- Puppet 必须在您的 Satellite 中启用。如需更多信息，请参阅在 [Red Hat Satellite 中使用 Puppet 集成管理配置中的启用 Puppet 与 Satellite 集成](#)。
- 您已启用并将 Satellite 客户端 6 存储库同步到 Satellite。如需更多信息，请参阅 [管理内容中的导入内容](#)。
- 您创建了为主机启用 Satellite 客户端 6 存储库的激活码。如需更多信息，请参阅 [管理内容中的管理激活码](#)。

流程

1. 在 **Satellite Web UI** 中，进入到 **Configure > Global Parameters** 以全局添加主机参数。或者，您可以导航到 **Configure > Host Groups** 并编辑或创建主机组，来仅将主机参数添加到主机组中。
- 2.

使用全局参数或主机组中的 `host` 参数启用 Puppet 代理。添加名为 `enable-puppet7` 的主机参数，选择布尔值类型，然后将值设为 `true`。

3.

在全局参数或主机组中使用以下主机参数为 Puppet 代理指定配置：

- 添加名为 `puppet_server` 的主机参数，选择字符串类型，并将值设为 Puppet 服务器的主机名，如 `puppet.example.com`。
- 可选：添加名为 `puppet_ca_server` 的主机参数，选择字符串类型，并将值设为 Puppet CA 服务器的主机名，如 `puppet-ca.example.com`。如果没有设置 `puppet_ca_server`，则 Puppet 代理将使用与 `puppet_server` 相同的服务器。
- 可选：添加名为 `puppet_environment` 的主机参数，选择字符串类型，并将值设置为您希望主机使用的 Puppet 环境。

在 [BZ2177730](#) 解析之前，您必须使用 `host` 参数来指定 Puppet 代理配置，即使 Puppet 服务器是 Capsule 服务器。

4.

导航到 `Hosts > Register Host`，并使用适当的激活码注册您的主机。如需更多信息，请参阅 [管理主机中的注册主机](#)。

5.

导航到 `Infrastructure > Capsules`。

6.

从所需胶囊服务器的 `Actions` 列中，选择 `Certificates`。

7.

单击所需主机右侧的 `Sign`，以签署 Puppet 代理的 SSL 证书。

3.8. 手动安装和配置 PUPPET 代理

您可以在主机上手动安装和配置 Puppet 代理。主机上需要配置了 Puppet 代理，以便 Puppet 与您的 Satellite 集成。有关 Puppet 的更多信息，请参阅在 [Red Hat Satellite 中使用 Puppet 集成管理配置](#)。

前提条件

-

Puppet 必须在您的 Satellite 中启用。如需更多信息，请参阅在 Red Hat Satellite 中使用 Puppet 集成管理配置中的启用 Puppet 与 Satellite 集成。

- **主机必须分配有 Puppet 环境。**
- **必须启用 Satellite 客户端 6 存储库并同步到 Satellite 服务器，并在主机上启用。如需更多信息，请参阅 [管理内容](#) 中的 导入内容。**

流程

1. **以 root 用户身份登录主机。**
2. **安装 Puppet 代理软件包。**
 - **在运行 Red Hat Enterprise Linux 8 及更高版本的主机上：**

```
# dnf install puppet-agent
```
 - **在运行 Red Hat Enterprise Linux 7 及以下的主机上：**

```
# yum install puppet-agent
```
3. **使用以下脚本，将 Puppet 代理添加到当前 shell 中的 PATH 中：**

```
./etc/profile.d/puppet-agent.sh
```
4. **配置 Puppet 代理。将 environment 参数设置为主机所属的 Puppet 环境的名称：**

```
# puppet config set server satellite.example.com --section agent
# puppet config set environment My_Puppet_Environment --section agent
```
5. **启动 Puppet 代理服务：**

```
# puppet resource service puppet ensure=running enable=true
```

6. 为主机创建证书：

```
# puppet ssl bootstrap
```

7. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Capsules**。

8. 从所需胶囊服务器的 **Actions** 列中，选择 **Certificates**。

9. 单击所需主机右侧的 **Sign**，以签署 Puppet 代理的 **SSL** 证书。

10. 在主机上再次运行 Puppet 代理：

```
# puppet ssl bootstrap
```

3.9. 在主机注册过程中运行 ANSIBLE 角色

在将主机注册到 **Satellite** 时，您可以运行 **Ansible** 角色。

前提条件

- 所需的 **Ansible** 角色已从 **Capsule** 导入到 **Satellite**。如需更多信息，请参阅在 **Red Hat Satellite** 中使用 **Ansible** 集成管理配置中的 [导入 Ansible 角色和变量](#)。

流程

1. 使用 **Ansible** 角色创建主机组。如需更多信息，请参阅 [第 2.7 节“创建主机组”](#)。
2. 使用主机组和分配的 **Ansible** 角色注册主机。如需更多信息，请参阅 [第 3.3.2 节“注册主机”](#)。

第 4 章 将主机从 KATELLO AGENT 迁移到远程执行

远程执行是管理主机上的软件包内容的首选方法。Katello Agent 已被弃用，并将在以后的 Satellite 版本中删除。按照以下步骤切换到远程执行。

前提条件

- 您已在 Satellite 服务器上启用了 Satellite 客户端 6 存储库。如需更多信息，请参阅 [在连接的网络环境中安装 Satellite 服务器中的启用 Satellite 客户端 6 存储库](#)。
- 您已在 Satellite 服务器上同步了 Satellite 客户端 6 存储库。如需更多信息，请参阅 [在连接的网络环境中安装 Satellite 服务器中的同步 Satellite 客户端 6 存储库](#)。
- 之前已在内容主机上安装 katello-agent 软件包。

流程

1. 如果将远程执行配置为使用 ssh 模式，请将远程执行 SSH 密钥分发到主机。更多信息请参阅 [第 12.14 节“为远程执行发布 SSH 密钥”](#)。
2. 如果将远程执行配置为使用 pull-mqtt 模式，请将远程执行拉取客户端部署到主机。更多信息请参阅 [第 12.5 节“配置主机以使用 Pull 客户端”](#)。
3. 在内容主机上停止 goferd 服务：

```
# systemctl stop goferd
```
4. 在内容主机上禁用 goferd 服务：

```
# systemctl disable goferd
```
5. 删除内容主机上的 Katello 代理：

**警告**

如果您的主机安装在 Red Hat Virtualization 版本 4.4 或较低版本上，请不要删除 `katello-agent` 软件包，因为删除的依赖关系会破坏主机。

```
# dnf remove katello-agent
```

6. 在 **Satellite Web UI** 中，导航到 **Administer > Settings**。
7. 选择 **内容选项卡**。
8. 将默认情况下 **Use remote execution** 设置为 **Yes**。

Satellite 服务器现在通过远程执行使用主机管理，而不是 Katello Agent。

下表显示了用于执行特定软件包操作的远程执行等效命令。请参阅 `hammer job-invocation create --help` 以了解如何指定搜索查询来确定目标主机或主机组。

表 4.1. hammer 命令

操作	Katello 代理	远程执行
安装软件包	hammer 主机软件包安装	hammer job-invocation create --feature katello_package_install
安装软件包（主机集合）	hammer host-collection 软件包安装	hammer job-invocation create --feature katello_package_install
删除软件包	hammer 主机软件包删除	hammer job-invocation create --feature katello_package_remove
删除软件包（主机集合）	hammer host-collection 软件包被删除	hammer job-invocation create --feature katello_package_remove

操作	Katello 代理	远程执行
更新软件包	hammer 主机软件包升级	hammer job-invocation create --feature katello_package_update
更新软件包 (主机集合)	hammer host-collection 软件包更新	hammer job-invocation create --feature katello_package_update
更新所有软件包	hammer 主机软件包更新	hammer job-invocation create --feature katello_package_update
安装勘误	应用 hammer 主机勘误	hammer job-invocation create --feature katello_errata_install
安装勘误 (主机集合)	hammer host-collection 勘误安装	hammer job-invocation create --feature katello_errata_install
安装软件包组	hammer host package-group install	hammer job-invocation create --feature katello_group_install
安装软件包组 (主机集合)	hammer host-collection package-group install	hammer job-invocation create --feature katello_group_install
删除软件包组	hammer host package-group remove	hammer job-invocation create --feature katello_group_remove
删除软件包组 (主机集合)	hammer host-collection package-group remove	hammer job-invocation create --feature katello_group_remove
更新软件包组	hammer 主机软件包更新	hammer job-invocation create --feature katello_group_update
更新软件包组 (主机集合)	hammer host-collection package-group update	hammer job-invocation create --feature katello_group_update

第 5 章 添加网络接口

Satellite 支持为单个主机指定多个网络接口。您可以在创建新主机时配置这些接口，如第 2.1 节“在 **Red Hat Satellite** 中创建主机”所述或编辑现有主机时。

您可以附加到主机的几种网络接口类型。在添加新接口时，请选择以下之一：

- **Interface**：允许您指定额外的物理或虚拟接口。您可以创建两种类型的虚拟接口。当主机需要通过单一接口与多个（虚拟）网络通信时，请使用 VLAN，而这些网络不能相互访问。使用 **alias** 为现有接口添加额外的 IP 地址。

有关添加物理接口的详情请参考第 5.1 节“添加物理接口”。

有关添加虚拟接口的详情请参考第 5.2 节“添加虚拟接口”。

- **绑定**：创建一个绑定接口。NIC 绑定是将多个网络接口绑定到一个单一设备并具有单个 MAC 地址的方法。这可让两个或多个网络接口充当一个，增加带宽并提供冗余。更多信息请参阅第 5.3 节“添加绑定接口”。
- **BMC**：基板管理控制器(BMC)允许您远程监控和管理机器的物理状态。有关 BMC 的更多信息，请参阅在连接的网络环境中安装 **Satellite** 服务器中的在受管主机上启用电源管理。有关配置 BMC 接口的详情请参考第 5.5 节“添加基板管理控制器(BMC)接口”。

注意

其他接口默认启用 **Managed** 标志，这意味着在置备过程中由与所选子网关联的 DNS 和 DHCP 胶囊式服务器自动配置新接口。这需要一个带有正确配置的 DNS 和 DHCP 胶囊式服务器的子网。如果您使用 Kickstart 方法进行主机置备，则会在 `/etc/sysconfig/network-scripts/ifcfg-interface_id` 的安装后阶段为受管接口自动创建配置文件。

注意

虚拟和绑定接口目前需要物理设备的 MAC 地址。因此，这些接口的配置仅适用于裸机主机。

5.1. 添加物理接口

使用此流程向主机添加额外的物理接口。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要编辑的主机旁的 **Edit**。
3. 在 **接口** 选项卡上，单击添加接口。
4. 保持在 **Type** 列表中选择 **Interface** 选项。
5. 指定 **MAC** 地址。此设置是必需的。
6. 指定设备标识符，如 **eth0**。标识符用于在创建绑定接口、**VLAN** 和别名时指定此物理接口。
7. 指定与主机的 **IP** 地址关联的 **DNS** 名称。Satellite 将此名称保存在与所选域(**DNS A**"字段)和与所选子网关联的 **Capsule** 服务器(**DNS PTR**"字段)关联的 **Capsule** 服务器中。因此，单个主机可以有多个 **DNS** 条目。
8. 从 **Domain** 列表选择一个域。要创建和管理域，请导航到 **Infrastructure > Domains**。
9. 从 **Subnet** 列表选择子网。要创建和管理子网，请导航到 **Infrastructure > Subnets**。
10. 指定 **IP** 地址。带有分配的 **DHCP** 胶囊服务器的受管接口需要此设置来创建 **DHCP** 租期。启用 **DHCP** 的受管接口会自动提供推荐的 **IP** 地址。
11. 选择接口是否为 **Managed**。如果接口受管理，则在调配期间从关联的胶囊服务器中提取配置，并且创建 **DNS** 和 **DHCP** 条目。如果使用 **kickstart** 调配，则会自动为接口创建一个配置文件。

12. **选择是否是主机的主接口。来自主接口的 DNS 名称用作 FQDN 的主机部分。**
13. **选择是否是主机的 Provision 接口。使用调配接口进行 TFTP 引导。对于基于镜像的调配, 脚本用于通过置备界面执行。**
14. **选择是否使用接口进行 远程执行。**
15. **保持 虚拟 NIC 复选框清除。**
16. **单击 OK 以保存接口配置。**
17. **点 Submit 将更改应用到主机。**

5.2. 添加虚拟接口

使用这个流程为主机配置虚拟接口。这可以是 VLAN 或别名接口。

别名接口是附加到现有接口的额外 IP 地址。别名接口自动从它所附加的接口继承 MAC 地址, 因此您可以在不指定 MAC 地址的情况下创建别名。接口必须在将引导模式设置为 静态 的子网中指定。

流程

1. **在 Satellite Web UI 中, 导航到 Hosts > All Hosts。**
2. **点您要编辑的主机旁的 Edit。**
3. **在 接口 选项卡上, 单击添加接口。**
4. **保持在 Type 列表中选择 Interface 选项。**

5. 指定常规接口设置。适用的配置选项与第 5.1 节“添加物理接口”中描述的物理接口相同。

为受管虚拟接口指定 MAC 地址，以便正确生成用于调配的配置文件。但是，未管理的虚拟接口不需要 MAC 地址。

如果创建 VLAN，在 Device Identifier 字段中以 eth1.10 的形式指定 ID。如果创建别名，请使用 ID，格式为 eth1:10。

6. 选择 Virtual NIC 复选框。特定于虚拟接口的附加配置选项会附加到表单中：

- 标签：（可选）将 VLAN 标签设置为从物理网络中继网络段到虚拟接口。如果没有指定标签，受管接口会继承关联子网的 VLAN 标签。来自此字段的用户指定的条目不会应用到别名接口。
- 附加到：指定虚拟接口所属的物理接口的标识符，如 eth1。此设置是必需的。

7. 单击 OK 以保存接口配置。

8. 点 Submit 将更改应用到主机。

5.3. 添加绑定接口

使用这个流程为主机配置绑定接口。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 Hosts > All Hosts。
2. 点您要编辑的主机旁的 Edit。
3. 在 接口 选项卡上，单击添加接口。

4. 从 **Type** 列表中选择 **Bond**。特定于类型的配置选项会附加到表单中。
5. 指定常规接口设置。适用的配置选项与第 5.1 节“添加物理接口”中描述的物理接口相同。

绑定接口在 **Device Identifier** 字段中以 **bond0** 的形式使用 ID。

单个 **MAC** 地址 已经足够。

如果要添加二级接口，请选择 **Managed**。否则，**Satellite** 不会应用配置。

6. 指定特定于绑定接口的配置选项：

- **模式**：选择定义容错和负载均衡策略的绑定模式。如需了解每个绑定模式的简短描述，请参阅第 5.4 节“**Satellite** 中可用的绑定模式”。
- **附加的设备**：指定附加设备的、以逗号分隔的标识符列表。这些可以是物理接口或 **VLAN**。
- **绑定选项**：指定一个以空格分隔的配置选项列表，如 **miimon=100**。有关绑定接口的配置选项的更多信息，请参阅 **Red Hat Enterprise Linux 配置和管理网络 指南**中的 **配置网络绑定**。

7. 单击 **OK** 以保存接口配置。

8. 点 **Submit** 将更改应用到主机。

CLI 过程

- 要使用绑定接口创建主机，请输入以下命令：

```
# hammer host create --name bonded_interface \
--hostgroup-id 1 \
--ip=192.168.100.123 \
--mac=52:54:00:14:92:2a \
```

```

--subnet-id=1 \
--managed true \
  --interface="identifier=eth1, \
    mac=52:54:00:62:43:06, \
    managed=true, \
    type=Nic::Managed, \
    domain_id=1, \
    subnet_id=1" \
  --interface="identifier=eth2, \
    mac=52:54:00:d3:87:8f, \
    managed=true, \
    type=Nic::Managed, \
    domain_id=1, \
    subnet_id=1" \
  --interface="identifier=bond0, \
    ip=172.25.18.123, \
    type=Nic::Bond, \
    mode=active-backup, \
    attached_devices=[eth1,eth2], \
    managed=true, \
    domain_id=1, \
    subnet_id=1" \
--organization "My_Organization" \
--location "My_Location" \
--ask-root-password yes

```

5.4. SATELLITE 中可用的绑定模式

绑定模式	Description
balance-rr	在每个绑定接口中按顺序接收和发送传输。
active-backup	传输通过第一个可用的绑定接口接收和发送。只有在活跃的绑定接口失败时，才会使用另一个绑定接口。
balance-xor	传输基于所选哈希策略。在这个模式中，用于特定对等点的流量始终通过同一接口发送。
broadcast	所有传输都在所有绑定接口上发送。
802.a3	创建共享相同设置的聚合组。在活动组中的所有接口上传输和接收。
balance-tlb	传出流量会根据每个绑定接口上的当前负载进行分发。
balance-alb	接收负载平衡通过地址解析协议(ARP)协商来实现。

5.5. 添加基板管理控制器(BMC)接口

使用这个流程为支持此功能的主机配置基板管理控制器(BMC)接口。

前提条件

- 已安装 `ipmitool` 软件包。
- 您知道主机上 BMC 接口的 MAC 地址、IP 地址和其他详情，以及该接口的适当凭证。



注意

只有在管理 BMC 接口时，您只需要 BMC 接口的 MAC 地址，以便它可以创建 DHCP 保留。

流程

1. 如果还没有启用，在 Capsule 服务器上启用 BMC：
 - a. 使用以下选项运行 `satellite-installer` 脚本，在 Capsule 服务器上配置 BMC 电源管理：

```
# satellite-installer --foreman-proxy-bmc=true \  
--foreman-proxy-bmc-default-provider=ipmitool
```
 - b. 在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**。
 - c. 从 **Actions** 列中的列表，单击 **Refresh**。**Features** 列中的列表现在应包含 **BMC**。
2. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
3. 点您要编辑的主机旁的 **Edit**。
4. 在 **接口** 选项卡上，单击添加接口。

5. **从 Type 列表中选择 BMC。特定于类型的配置选项会附加到表单中。**
6. **指定常规接口设置。适用的配置选项与 [第 5.1 节“添加物理接口”](#) 中描述的物理接口相同。**
7. **指定特定于 BMC 接口的配置选项：**
 - **Username 和 Password** : 指定 BMC 所需的任何身份验证凭证。
 - **Provider** : 指定 BMC 供应商。
8. **单击 OK 以保存接口配置。**
9. **点 Submit 将更改应用到主机。**

第 6 章 将主机升级到 RED HAT ENTERPRISE LINUX 下一个主版本

您可以使用作业模板将 Red Hat Enterprise Linux 主机升级到下一个主版本。以下升级路径是可能的：

- **Red Hat Enterprise Linux 7 到 Red Hat Enterprise Linux 8**
- **Red Hat Enterprise Linux 8 到 Red Hat Enterprise Linux 9**

前提条件

- **确保您的 Red Hat Enterprise Linux 主机满足升级要求。**
 - 对于 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8 升级，请参阅从 RHEL 7 升级到 RHEL 8 中的规划升级。https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/8/html/upgrading_from_rhel_7_to_rhel_8/planning-an-upgrade_upgrading-from-rhel-7-to-rhel-8
 - 对于 Red Hat Enterprise Linux 8 升级到 Red Hat Enterprise Linux 9，请参阅从 RHEL 8 升级到 RHEL 9 中的规划升级到 RHEL 9。
- **为升级准备 Red Hat Enterprise Linux 主机。**
 - 对于 Red Hat Enterprise Linux 7 到 Red Hat Enterprise Linux 8 升级，请参阅从 RHEL 7 升级到 RHEL 8 中的为升级准备 RHEL 7 系统。
 - 对于 Red Hat Enterprise Linux 8 到 Red Hat Enterprise Linux 9 升级，请参阅从 RHEL 8 升级到 RHEL 9 中的为升级准备 RHEL 8 系统。
- **在 Satellite 上启用远程执行功能。更多信息请参阅第 12 章配置和设置远程作业。**
- **将 Satellite SSH 密钥分发到您要升级的主机。更多信息请参阅第 12.14 节“为远程执行发布 SSH 密钥”。**

流程

1. 在 **Satellite** 中，启用 **Leapp** 插件：

```
# satellite-installer --enable-foreman-plugin-leapp
```

2. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。

3. 选择您要升级到下一个主 **Red Hat Enterprise Linux** 版本的主机。

4. 在 **Hosts** 窗口的右上角，从 **Select Action** 列表中选择 **Preupgrade check with Leapp**。

5. 点 **Submit** 以开始预升级检查。

6. 检查完成后，点 **Leapp preupgrade report** 选项卡查看主机上的 **Leapp** 是否存在任何问题。具有 **Inhibitor** 标志的问题被视为关键，可能破坏升级过程。具有 **Has Remediation** 标志的问题包含可帮助您解决这个问题的补救。

- a. 点标记为 **Has Remediation** 的问题来扩展它。

- 如果问题包含补救命令，您可以使用远程执行直接从 **Satellite** 修复它。选择问题。
- 如果问题只包含补救 **Hint**，请使用提示来手动修复主机上的问题。

对其他问题重复此步骤。

- b. 在选择了补救命令中的任何问题后，点 **Fix Selected** 并提交作业。

- c. 解决了这个问题后，点 **Rerun** 按钮，然后点 **Submit** 来再次运行预升级检查，以验证您要升级的主机是否有任何问题，并准备好升级。

7. 如果预升级检查验证主机是否有任何问题，点 **Run Upgrade** 按钮，然后点 **Submit** 以启动升级。

8.

可选： 您还可以通过在主机详情页面中选择 **Schedule a Job** 下拉菜单来进行升级。

第 7 章 将主机转换为 RED HAT ENTERPRISE LINUX

您可以将 Red Hat Enterprise Linux 派生版本转换为可支持的 Red Hat Enterprise Linux，同时保持安装的应用程序和配置。Satellite 提供了 Convert2RHEL 工具来简化转换过程。

Satellite 中的 Convert2RHEL 工具包含一个 Ansible 角色和 Ansible playbook。您可以使用 Ansible 角色在 Satellite 服务器上生成转换数据，其中包括启用所需的存储库并创建产品、激活码和主机组。然后，您可以使用 Ansible playbook 在主机上执行实际转换，该 playbook 在主机上安装 Convert2RHEL CLI 工具并运行它。

您可以使用 Ansible 角色为以下转换生成转换数据：

- **CentOS Linux 7 到 Red Hat Enterprise Linux 7**
- **Oracle Linux 7 到 Red Hat Enterprise Linux 7**
- **Oracle Linux 8 到 Red Hat Enterprise Linux 8**

红帽支持这些转换。

转换过程与 Red Hat Enterprise Linux 的次版本升级类似，其中每个 RPM 软件包都被替换。保留在 Red Hat Enterprise Linux 中不提供的第三方软件包和非红帽软件包。

Convert2RHEL 工具会删除不必要的软件包，如在转换过程中造成问题的徽标或软件包。该工具将 CentOS-release 或 Oracle-release 软件包替换为 rhel-release 软件包，以及由 CentOS 或 Oracle 签名的所有软件包及其对应的红帽。该工具还订阅了 Red Hat Subscription Management 的主机。

转换过程的持续时间取决于必须替换的软件包数量、网络速度、存储速度和类似因素。

前提条件

- 请参阅 Red Hat Enterprise Linux 8 中的 [支持的转换路径](#)，使用 Convert2RHEL 工具从 Linux 发行版本转换到 RHEL。

- 您必须已完成使用 `Convert2RHEL` 工具从 Linux 发行版转换到 RHEL 中的准备 RHEL 转换的步骤 `1.wagon -wagon5`。
- 确保您有一个上传到 Satellite 的订阅清单，并且为您想要的转换分配了足够的 Red Hat Enterprise Linux 权利。或者，您可以使用 Ansible 变量告知角色从磁盘导入清单。清单必须导入到要向其注册主机以进行转换的组织。

您可以更新分配并 [从红帽客户门户下载](#) 更新的清单。如需更多信息，请参阅在 Red Hat Subscription Management 中[使用清单](#)。
- 确保您已在 Satellite 中启用并同步了 Red Hat 仓库，用于将主机转换为的次要 Red Hat Enterprise Linux 版本。如需更多信息，请参阅管理内容中的[启用红帽存储库](#)和[同步存储库](#)。

高级转换步骤

1. 导入 `redhat.satellite.convert2rhel` Ansible 角色和变量。如需更多信息，请参阅在 Red Hat Satellite 中使用 Ansible 集成管理配置中的[导入 Ansible 角色和变量](#)。
2. 配置 Ansible 变量以生成转换数据。如需更多信息，请参阅第 7.1 节“[用于转换的 Ansible 变量](#)”。
3. 将 `redhat.satellite.convert2rhel` 角色分配给代表 Satellite 服务器的主机。如需更多信息，请参阅在 Red Hat Satellite 中使用 Ansible 集成管理配置中的[将 Ansible 角色分配给现有主机](#)。
4. 在 Satellite 服务器上运行 Ansible 角色。如需更多信息，请参阅在 Red Hat Satellite 中使用 Ansible 集成管理配置中的[在主机上运行 Ansible 角色](#)。

Ansible 角色生成主机转换所需的数据，即存储库、证书、激活码和主机组。角色启用带有 7Server 发行版本和 x86_64 架构的 `rhel-7-server-rpms` 存储库，或 `rhel-8-for-x86_64-baseos-rpms` 和 `rhel-8-for-x86_64-appstream-rpms`，或两者取决于您在前面的步骤中设置的变量。
5. 注册主机以使用生成的主机组进行转换。

在转换前，使用全局注册模板注册并订阅您的主机。选择您要为转换生成的主机组，如 CentOS 7 转换（如果从 CentOS 7 转换）。如需更多信息，请参阅第 3.3 节“[使用全局注册注册主机](#)”。

6.

在主机组上运行 **Convert2RHEL** **playbook**。使用以下设置执行远程作业：

- 作业类别：转换 2 RHEL
- 作业模板：转换到 RHEL
- 激活码：
 - 如果您转换为 Red Hat Enterprise Linux 7, `convert2rhel_rhel7`
 - 如果您转换为 Red Hat Enterprise Linux 8, `convert2rhel_rhel8`

更多信息请参阅 [第 12.22 节“执行远程作业”](#)。

其他资源

- [如何在红帽知识库中执行从 RHEL 派生的 Linux 发行版本到 RHEL 的不受支持的转换](#)

7.1. 用于转换的 ANSIBLE 变量

在运行 Ansible 角色以生成转换数据之前，请配置以下所需 Ansible 变量的值：

Satellite 从 `redhat.satellite.convert2rhel` 角色导入大多数所需的 Ansible 变量。但是，一些变量不会被导入。这些变量在下表中标有星号。您必须手动创建这些额外变量，并将它们分配到 `redhat.satellite.convert2rhel` 角色。

表 7.1. 转换所需的变量

名称	类型	意图和值
<code>satellite_server_url</code> *	字符串	Satellite 服务器的 URL，如 https://satellite.example.com
<code>satellite_username</code> *	字符串	您的用户名

名称	类型	意图和值
<code>satellite_password *</code>	字符串	您的密码
<code>satellite_organization *</code>	字符串	您的机构名称
<code>satellite_content_rhel_wait_for_syncs *</code>	布尔值	如果您不希望 Satellite 服务器在继续生成数据前等待存储库同步完成，则设置为 false 。（默认为 true ）
<code>satellite_validate_certs *</code>	布尔值	如果要在 Ansible 中启用证书检查，设置为 true 。（默认值： true ）
<code>satellite_convert2rhel_manage_subscription</code>	布尔值	如果您已在 Satellite 服务器上有清单，则设置为 false 。如果您从磁盘上传新清单，则当前清单将被覆盖。（默认值： true ）
<code>satellite_content_rhel_enable_rhel7 *</code>	布尔值	启用 Red Hat Enterprise Linux 7 软件仓库。如果您不打算将主机转换为 Red Hat Enterprise Linux 7，则设置为 false （默认值： true ）
<code>satellite_convert2rhel_enable_oracle7</code>	布尔值	如果要为 Oracle Linux 7 准备转换数据，则设置为 true 。否则，您必须将值设为 false 。
<code>satellite_content_rhel_enable_rhel8 *</code>	布尔值	启用 Red Hat Enterprise Linux 8 软件仓库。如果您不打算将主机转换为 Red Hat Enterprise Linux 8，则设置为 false （默认值： true ）
<code>satellite_convert2rhel_enable_oracle8</code>	布尔值	如果要为 Oracle Linux 8 准备转换数据，则设置为 true 。否则，您必须将值设为 false 。

表 7.2. 用于转换的可选变量

名称	类型	意图和值
<code>satellite_manifest_path *</code>	字符串	要从磁盘上传的清单路径，如 <code>~/manifest.zip</code> 。如果您使用 <code>satellite_convert2rhel_manage_subscription</code> 从磁盘上传新清单，则必须设置此路径。
<code>satellite_content_rhel_rhel8_releasever *</code>	字符串	次发行版本，如 8.5 。如果您的系统的次版本与最新的 Red Hat Enterprise Linux 发行版本不同，则设置此变量以防止转换问题。（默认为 <code>latest</code> ）

第 8 章 使用 RHEL WEB 控制台进行主机管理和监控

RHEL web 控制台是一个交互式 Web 界面，可用于执行操作并监控 Red Hat Enterprise Linux 主机。您可以启用 remote-execution 功能，将 Satellite 与 RHEL web 控制台集成。当您在使用 Satellite 管理的主机上安装 RHEL web 控制台时，您可以从 Satellite web UI 中查看该主机的 RHEL web 控制台仪表盘。您还可以使用与 RHEL web 控制台集成的功能，如 Red Hat Image Builder。

8.1. 在 SATELLITE 中启用 RHEL WEB 控制台

默认情况下，在 Satellite 中禁用了 RHEL web 控制台集成。如果要从 Satellite 中访问主机的 RHEL web 控制台功能，您必须首先在 Satellite 服务器上启用 RHEL web 控制台集成。

流程

- 在 Satellite 服务器上启用 RHEL web 控制台：

```
# satellite-installer --enable-foreman-plugin-remote-execution-cockpit
```

8.2. 使用 RHEL WEB 控制台管理和监控主机

您可以通过 Satellite Web UI 访问 RHEL web 控制台 Web UI，并使用该功能管理和监控 Satellite 中的主机。

前提条件

- RHEL web 控制台在 Satellite 中启用。
- RHEL web 控制台安装在您要查看的主机上：
 - 对于 Red Hat Enterprise Linux 8，请参阅[使用 RHEL 8 web 控制台管理系统指南中的安装 Web 控制台](#)。
 - 对于 Red Hat Enterprise Linux 7，请参阅[使用 RHEL 7 web 控制台管理系统指南中的安装 Web 控制台](#)。
- 卫星或 Capsule 可以使用 SSH 密钥对主机进行身份验证。更多信息请参阅[第 12.14 节“为远程执行发布 SSH 密钥”](#)。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts** 并选择您要使用 **RHEL web 控制台** 管理和监控的主机。
2. 在主机窗口右上角，单击垂直 ellipsis 并选择 **Web Console**。

现在，您可以通过 **RHEL web 控制台** 访问主机监控和管理的完整功能。

有关 **Red Hat web 控制台** 入门的更多信息，请参阅使用 [RHEL 8 web 控制台指南](#) 或 [使用 RHEL 7 web 控制台的管理系统指南](#)。

有关通过 **RHEL web 控制台** 使用 **Red Hat Image Builder** 的更多信息，请参阅 [RHEL 8 web 控制台中的访问镜像构建器 GUI](#) 或 [RHEL 7 web 控制台中的访问镜像构建器 GUI](#)。

8.3. 在 SATELLITE 上禁用 RHEL WEB 控制台

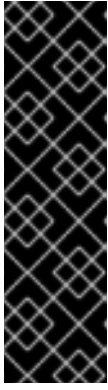
如果要在 **Satellite** 上禁用 **RHEL web 控制台**，请执行以下步骤。

流程

1. 运行这个 `satellite-installer` 命令：

```
satellite-installer --no-enable-foreman-plugin-remote-execution-cockpit
```
2. 在 **Satellite Web UI** 中，进入到 **Administer > Settings** 并点 **Remote execution** 选项卡。
3. 在 **Cockpit URL** 行中，清除 **Value** 下的设置，然后单击 **Submit**。这会从 **Satellite Web UI** 中删除 **Web 控制台** 按钮。
4. 从 **Satellite** 卸载 **RHEL web 控制台** 软件包：

```
dnf remove rubygem-foreman_remote_execution-cockpit
```

**重要**

RHEL web 控制台集成可以在 Capsule 服务器上独立启用或禁用。要防止在 Capsule 服务器上启用 RHEL web 控制台集成，请在完成以下步骤后运行以下命令：

```
satellite-installer --foreman-proxy-plugin-remote-execution-script-cockpit-integration=false
```

第 9 章 使用 RED HAT INSIGHTS 监控主机

在本章中，您可以找到有关使用 Red Hat Insights 创建主机监控报告和监控主机的信息，并创建 Insights 计划。

9.1. 在 SATELLITE 中将 RED HAT INSIGHTS 与主机搭配使用

您可以使用 Red Hat Insights 诊断系统和与安全漏洞相关的系统停机时间、性能下降和稳定性故障。您可以使用控制面板快速识别稳定性、安全性和性能的关键风险。您可以按类别排序，查看影响和解决方案的详细信息，然后确定受影响的系统。

要使用 Red Hat Insights 监控您使用 Satellite 管理的主机，您必须首先在主机上安装 Red Hat Insights，并将主机注册到 Red Hat Insights。

对于新的 Satellite 主机，您可以使用全局注册模板在注册过程中安装和配置 Satellite 主机。如需更多信息，请参阅管理主机中的使用全局注册模板将主机注册到 Red Hat Satellite。

要使用 Puppet 安装和注册您的主机，或手动注册您的主机，请参阅 [Red Hat Insights 入门](#)。

可供主机的 Red Hat Insights 信息

有关主机的更多信息，请参阅 Red Hat Insights。

您可以在两个位置找到此信息：

- 在 Satellite Web UI 中，进入到 **Configure > Insights**，在 **Remediate** 按钮旁的垂直 ellipsis 提供一个指向常规建议页面的 Red Hat Insights 链接。在每个建议行中，垂直的 ellipsis 在 Red Hat Insights 中提供了一个 **View to the recommendations** 规则，如果这个规则可用于该建议，一个 **知识库文章** 链接。
- 如需更多信息，请导航到 **Hosts > All Hosts**。如果主机列出了建议，点建议数。在 **Insights** 选项卡中，**Remediate** 按钮旁边的垂直 ellipsis 提供了一个 **Go To Satellite Insights** 页面链接，指向系统的信息，以及 Red Hat Insights 链接中的 **View to host details**。

从 rh-cloud 和 insights-client 报告中排除主机

您可以将 `host_registration_insights` 参数设置为 `False`，以省略 rh-cloud 和 insights-client 报

告。Satellite 将排除来自 rh-cloud 报告的主机，并阻止 insight-client 将报告上传到云。

您还可以在组织、hostgroup、子网和域级别设置此参数。只要与实体关联，它会自动防止新报告被上传。

如果您已在 Red Hat Hybrid Cloud 上报告的主机上将参数设置为 false，它仍然会被自动从清单中删除。但是，这个过程可能需要一些时间才能完成。

使用 Ansible 角色部署 Red Hat Insights

您可以使用 RedHatInsights.insights-client Ansible 角色自动使用 Red Hat Insights 自动安装和注册主机。有关在 Satellite 中添加此角色的更多信息，请参阅在 Red Hat Satellite 中使用 Ansible 集成管理配置中的开始使用 Satellite 中的 Ansible。

1. 将 RedHatInsights.insights-client 角色添加到主机。

对于新主机，请参阅第 2.1 节“在 Red Hat Satellite 中创建主机”。

对于现有主机，请参阅在 Red Hat Satellite 中使用 Ansible 集成管理配置中的使用 Ansible 角色在客户端上自动化重复的任务。

2. 要在主机上运行 RedHatInsights.insights-client 角色，请导航到 Hosts > All Hosts，再单击要使用的主机的名称。
3. 单击 Run Ansible roles 按钮。

在继续操作前，您必须为 Insights 设置 API 令牌。如需更多信息，请参阅 Red Hat API Tokens。

您可以按照以下流程手动同步建议：

1. 在 Satellite Web UI 中，进入到 Configure > Insights。
2. 点 Start Recommendations Sync 按钮。

如果您还没有设置 API 令牌，则在使用这个页面前提示您创建一个。

其它信息

- 要查看 Red Hat Insights 和所有插件的日志，请转至 `/var/log/foreman/production.log`。
- 如果您在连接到 Red Hat Insights 时出现问题，请确保您的证书是最新的。刷新订阅清单以更新证书。
- 您可以通过在主机上配置 `insights-client.timer` 来更改运行 `insights-client` 的默认调度。如需更多信息，请参阅 Red Hat Insights 的客户端配置指南中的 [更改 insights-client 调度](#)。

9.2. 为主机创建 INSIGHTS 计划

借助 Satellite，您可以创建 Red Hat Insights 修复计划，并在 Satellite 主机上运行计划。

流程

1. 在 Satellite web UI 中，进入 `Configure > Insights`。
2. 在 Red Hat Insights 页面中，选择您要包含在 Insights 计划中的建议数。

您只能选择具有关联 `playbook` 的建议。
3. 点 `Remediate`。
4. 在 Remediation Summary 窗口中，您可以选择要应用的解析。使用 Filter 字段搜索特定的关键字。
5. 点 `Remediate`。
6. 在 Job Invocation 页面中，不要更改预完成字段的内容。

7. 可选。如需远程执行作业的更多高级配置，请单击 **Show Advanced Fields**。
8. 选择您需要的 **查询类型**。
9. 选择您需要的 **Schedule**。
10. 点 **Submit**。

或者：

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 选择主机。
3. 在 **Host** 详情页面上，单击 **Recommendations**。
4. 在 **Red Hat Insights** 页面中，选择您要包含在 **Insights** 计划中的建议数量，如以前一样。

在 **Jobs** 窗口中，您可以查看计划的进度。

第 10 章 使用报告模板来监控主机

您可以使用报告模板来查询 **Satellite** 数据，以获取有关主机状态、注册的主机、适用勘误表、应用的勘误表、订阅详情和用户活动的信息。您可以使用 **Satellite** 附带的报告模板，或编写您自己的自定义报告模板以满足您的要求。报告引擎使用嵌入式 **Ruby (ERB)** 语法。有关编写模板和 **ERB** 语法的更多信息，请参阅 [附录 A, 模板编写参考](#)。

您可以创建一个模板，或克隆模板并编辑克隆。有关模板语法的帮助信息，请单击模板并单击 **帮助** 选项卡。

10.1. 生成主机监控报告

要在 **Satellite Web UI** 中查看报告模板，请导航到 **Monitor > Report Templates**。若要调度报告，请配置 **cron** 任务或使用 **Satellite Web UI**。

流程

1. 在 **Satellite Web UI** 中，导航到 **Monitor > Report Templates**。例如，以下模板可用：

主机 - 安装的产品

将此模板用于 **Simple Content Access (SCA)** 机构中的主机。它生成报告以及安装的产品信息以及 **Subscription - 权利报告** 中包含的其他指标，除了有关订阅的信息。

订阅 - 权利报告

将此模板用于不在 **SCA** 组织中的主机。它生成报告，其中包含有关订阅权利的信息，包括其过期时间。它仅输出不使用 **SCA** 的机构中主机的信息。

2. 在您要使用的报告模板右侧，点 **Generate**。
3. 可选：要在 **Generate at** 字段右侧调度报告，请点击图标来选择您要生成报告的日期和时间。
4. 可选：要将报告发送到电子邮件地址，请选择 **通过电子邮件** 复选框发送报告，并在 **发送到电子邮件地址** 字段中输入所需的电子邮件地址。

5. **可选：应用搜索查询过滤器。** 要查看所有可用结果，请不要使用任何值填充 `filter` 字段。
6. **点 `Submit`。** 下载包含报告的 `CSV` 文件。如果您选择了通过电子邮件发送报告，主机监控报告将发送到您的电子邮件地址。

CLI 过程

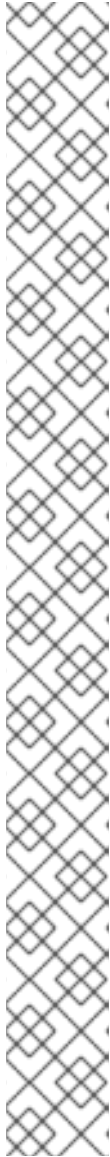
1. **列出所有可用的报告模板：**

```
# hammer report-template list
```

2. **生成一个报告：**

```
# hammer report-template generate --id My_Template_ID
```

此命令会等待报告在完成前完全生成。如果要将报告作为后台任务生成，您可以使用 `hammer report-template schedule` 命令。



注意

如果要生成订阅授权报告，则必须使用 **Now** 选项中的 **Days** 指定权利订阅的最新过期时间。您可以使用 **no limit** 值显示所有权利。

显示所有权利

```
# hammer report-template generate \
--inputs "Days from Now=no limit" \
--name "Subscription - Entitlement Report"
```

显示在 60 天后将过期的所有权利

```
# hammer report-template generate \
--inputs "Days from Now=60" \
--name "Subscription - Entitlement Report"
```

10.2. 创建报告模板

在 **Satellite** 中，您可以创建一个报告模板并自定义模板以符合您的要求。您可以导入现有的报告模板，并使用片断和模板宏进一步自定义它们。

报告模板使用嵌入式 Ruby (ERB) 语法。要查看有关使用 ERB 语法和宏的信息，在 **Satellite Web UI** 中进入 **Monitor > Report Templates**，然后点 **Create Template**，然后点 **Help** 选项卡。

在 **Satellite** 中创建报告模板时，默认启用安全模式。

流程

1. 在 **Satellite Web UI** 中，导航到 **Monitor > Report Templates**，然后点 **Create Template**。

2. 在 **Name** 字段中输入您的报告模板的唯一名称。
3. 如果您希望模板可用于所有位置和组织，请选择 **默认**。
4. 直接在模板编辑器中创建模板，或者通过单击 **Import** 从文本文件导入模板。有关导入模板的更多信息，请参阅 [第 10.5 节“导入报告模板”](#)。
5. 可选：在 **Audit Comment** 字段中，您可以添加有关此模板的任何有用信息。
6. 点 **Input** 选项卡，在 **Name** 字段中输入您可以在模板中引用的输入名称，其格式如下：输入 ('name')。请注意，您必须先保存模板，然后才能在模板正文中引用此输入值。
7. 选择输入值是否是必需的。如果输入值是必需的，请选择 **Required** 复选框。
8. 从 **Value Type** 列表中，选择用户必须输入的输入值类型。
9. 可选：如果要使用事实进行模板输入，请选择 **高级** 复选框。
10. 可选：在 **Options** 字段中，定义用户可以从中的选项。如果此字段未定义，用户会收到一个自由文本字段，他们可以输入想要的值。
11. 可选：在 **Default** 字段中输入一个值，如主机名，您要设置为默认模板输入。
12. 可选：在 **Description** 字段中，您可以输入您要在生成报告时显示为内联帮助的信息。
13. 可选：点 **Type** 选项卡，然后选择此模板是否是一个要包含在其他模板中的代码片段。
14. 点 **Location** 选项卡，并添加您要使用模板的位置。

15. 单击 **Organizations** 选项卡，再添加您要使用该模板的组织。
16. 单击 **Submit** 以保存您的更改。

其他资源

- 有关安全模式的详情，请参考第 10.9 节“报告模板安全模式”。
- 有关编写模板的更多信息，请参阅附录 A, 模板编写参考。
- 有关您可以在报告模板中使用的宏的更多信息，请参阅第 A.6 节“templates Macros”。
- 要通过填充模板的步骤示例来查看步骤，请参阅第 10.8 节“创建报告模板以监控权利”。

10.3. 导出报告模板

您可以导出您在 **Satellite** 中创建的报告模板。

流程

1. 在 **Satellite Web UI** 中，导航到 **Monitor > Report Templates**。
2. 找到您要导出的模板，然后从 **Actions** 列中的列表中，选择 **Export**。
3. 对您要下载的报告模板重复此步骤。

包含模板下载的 **.erb** 文件。

CLI 过程

1. 要查看可用于导出的报告模板，请输入以下命令：

■

```
# hammer report-template list
```

记下要在此命令的输出中导出的模板 ID。

2.

要导出报告模板，请输入以下命令：

```
# hammer report-template dump --id My_Template_ID > example_export.erb
```

10.4. 使用 SATELLITE API 导出报告模板

您可以使用 `Satellite report_templates` API 从 `Satellite` 导出报告模板。有关使用 `Satellite API` 的更多信息，请参阅 [API 指南](#)。

流程

1.

使用以下请求检索可用报告模板列表：

请求示例：

```
$ curl --insecure --user admin:redhat \  
--request GET \  
--config https://satellite.example.com/api/report_templates \  
| json_reformat
```

在本例中，`json_reformat` 工具用于格式化 JSON 输出。

响应示例：

```
{  
  "total": 6,  
  "subtotal": 6,  
  "page": 1,  
  "per_page": 20,  
  "search": null,  
  "sort": {
```

```

    "by": null,
    "order": null
  },
  "results": [
    {
      "created_at": "2019-11-20 17:49:52 UTC",
      "updated_at": "2019-11-20 17:49:52 UTC",
      "name": "Applicable errata",
      "id": 112
    },
    {
      "created_at": "2019-11-20 17:49:52 UTC",
      "updated_at": "2019-11-20 17:49:52 UTC",
      "name": "Applied Errata",
      "id": 113
    },
    {
      "created_at": "2019-11-30 16:15:24 UTC",
      "updated_at": "2019-11-30 16:15:24 UTC",
      "name": "Hosts - complete list",
      "id": 158
    },
    {
      "created_at": "2019-11-20 17:49:52 UTC",
      "updated_at": "2019-11-20 17:49:52 UTC",
      "name": "Host statuses",
      "id": 114
    },
    {
      "created_at": "2019-11-20 17:49:52 UTC",
      "updated_at": "2019-11-20 17:49:52 UTC",
      "name": "Registered hosts",
      "id": 115
    },
    {
      "created_at": "2019-11-20 17:49:52 UTC",
      "updated_at": "2019-11-20 17:49:52 UTC",
      "name": "Subscriptions",
      "id": 116
    }
  ]
}

```

2.

记录下您要导出的模板的 **id**，并使用以下请求导出模板：

请求示例：

```
$ curl --insecure --output /tmp/_Example_Export_Template.erb \  
--user admin:password --request GET --config \  
https://satellite.example.com/api/report_templates/My_Template_ID/export
```

请注意，158 是要导出的模板的示例 ID。

在本例中，导出的模板被重定向到 `host_complete_list.erb`。

10.5. 导入报告模板

您可以将报告模板导入到您要创建的新模板的正文中。请注意，使用 **Satellite Web UI**，您只能单独导入模板。对于批量操作，请使用 **Satellite API**。更多信息请参阅 [第 10.6 节“使用 Satellite API 导入报告模板”](#)。

前提条件

- 您必须已从 **Satellite** 导出模板，才能导入它们以便在新模板中使用。如需更多信息，请参阅 [第 10.3 节“导出报告模板”](#)。

流程

1. 在 **Satellite Web UI** 中，导航到 **Monitor > Report Templates**。
2. 在 **Report Templates** 窗口的右上角，单击 **Create Template**。
3. 在 **Editor** 选项卡的右上角，点文件夹图标，然后选择您要导入的 **.erb** 文件。
4. 编辑模板以符合您的要求。
5. 点 **Submit**。

有关自定义新模板的详情，请参考 [附录 A, 模板编写参考](#)。

10.6. 使用 SATELLITE API 导入报告模板

您可以使用 **Satellite API** 将报告模板导入到 **Satellite**。使用 **Satellite API** 导入报告模板会自动解析报告模板元数据并分配组织和位置。有关使用 **Satellite API** 的更多信息，请参阅 [API 指南](#)。

前提条件

- 使用 **.erb** 语法创建模板，或者从另一个卫星导出模板。

有关编写模板的更多信息，请参阅 [附录 A, 模板编写参考](#)。

有关从 **Satellite** 导出模板的更多信息，请参阅 [第 10.4 节 “使用 Satellite API 导出报告模板”](#)。

流程

1. 使用以下示例格式化您要导入到 **.json** 文件中的模板：

```
# cat Example_Template.json
{
  "name": "Example Template Name",
  "template": "
  Enter ERB Code Here
"
}
```

使用 **ERB** 模板的 **JSON** 文件示例：

```
{
  "name": "Hosts - complete list",
  "template": "
<%#
name: Hosts - complete list
snippet: false
template_inputs:
- name: host
  required: false
  input_type: user
  advanced: false
```

```

value_type: plain
resource_type: Katello::ActivationKey
model: ReportTemplate
-%>
<% load_hosts(search: input('host')).each_record do |host| -%>
<%
  report_row(
    'Server FQDN': host.name
  )
-%>
<% end -%>
<%= report_render %>
"
}

```

2.

使用以下请求导入模板：

```

$ curl --insecure --user admin:redhat \
--data @Example_Template.json --header "Content-Type:application/json" \
--request POST --config https://satellite.example.com/api/report_templates/import

```

3.

使用以下请求来检索报告模板列表，并验证您可以在 **Satellite** 中查看模板：

```

$ curl --insecure --user admin:redhat \
--request GET --config https://satellite.example.com/api/report_templates | json_reformat

```

10.7. 生成已安装软件包列表

使用这个流程在 报告模板中生成已安装的软件包列表。

流程

1.

在 **Satellite Web UI** 中，导航到 **Monitor > Report Templates**。

2.

在主机右侧 - **All Installed Packages**，单击 **Generate**。

3.

可选：使用 **Hosts 过滤器** 搜索字段搜索并应用特定的主机过滤器。

4. **点 Generate。**
5. **如果下载没有自动启动，请单击 Download。**

验证

- **您有电子表格，列出了系统上下载的所选主机的已安装软件包。**

10.8. 创建报告模板以监控权利

您可以使用报告模板返回具有特定订阅的主机列表，并显示这些主机的内核数。有关编写模板的更多信息，请参阅 [附录 A, 模板编写参考](#)。

流程

1. **在 Satellite Web UI 中，导航到 Monitor > Report Templates，然后点 Create Template。**
2. **可选：在 Editor 字段中，使用 <%# > tags 添加注释，其中包含稍后引用可能有用的信息。例如：**

```
<%#
name: Entitlements
snippet: false
model: ReportTemplate
require:
- plugin: katello
  version: 3.14.0
-%>
```

3. **使用 load_hosts () 宏添加一行，并使用以下方法和变量填充宏：**

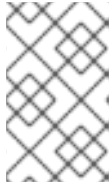
```
<%- load_hosts(includes: [:lifecycle_environment, :operatingsystem, :architecture,
:content_view, :organization, :reported_data, :subscription_facet, :pools =>
[:subscription]]).each_record do |host| -%>
```

要查看您可以使用的变量列表，请点 Help 选项卡和 Safe 模式 方法和变量 表中，找到 Host::Managed 行。

4.

使用每个方法添加带有 `host.pools` 变量的行，例如：

```
<%- host.pools.each do |pool| -%>
```



注意

池只适用于非简单内容访问(SCA)组织。

5.

使用 `report_row ()` 方法添加一行，以创建报告并添加您要目标的变量作为报告的一部分：

```
<%- report_row(
  'Name': host.name,
  'Organization': host.organization,
  'Lifecycle Environment': host.lifecycle_environment,
  'Content View': host.content_view,
  'Host Collections': host.host_collections,
  'Virtual': host.virtual,
  'Guest of Host': host.hypervisor_host,
  'OS': host.operatingsystem,
  'Arch': host.architecture,
  'Sockets': host.sockets,
  'RAM': host.ram,
  'Cores': host.cores,
  'SLA': host_sla(host),
  'Products': host_products(host),
  'Subscription Name': sub_name(pool),
  'Subscription Type': pool.type,
  'Subscription Quantity': pool.quantity,
  'Subscription SKU': sub_sku(pool),
  'Subscription Contract': pool.contract_number,
  'Subscription Account': pool.account_number,
  'Subscription Start': pool.start_date,
  'Subscription End': pool.end_date,
  'Subscription Guest': registered_through(host)
) -%>
```

6.

在模板中添加 `end` 语句：

```
<%- end -%>
<%- end -%>
```

7.

要生成报告，您必须添加 `<%= report_render -%>` 宏：

```
<%= report_render -%>
```

8.

单击 **Submit** 以保存模板。

10.9. 报告模板安全模式

在 **Satellite** 中创建报告模板时，默认启用安全模式。安全模式限制在报告模板中使用的宏和变量。安全模式可防止呈现问题并在报告模板中实施最佳实践。**Satellite Web UI** 中提供了支持的宏和变量列表。

要查看可用的宏和变量，在 **Satellite Web UI** 中，进入 **Monitor > Report Templates** 并点 **Create Template**。在 **Create Template** 窗口中，单击 **Help** 选项卡并展开 **Safe** 模式方法。

虽然启用了安全模式，但如果您尝试使用在 **Safe** 模式方法中没有列出的宏或变量，则模板编辑器会显示错误消息。

要在 **Satellite** 中查看安全模式的状态，请在 **Satellite Web UI** 中进入 **Administer > Settings** 并点 **Provisioning** 选项卡。找到 **Safemode** 呈现行以检查值。

第 11 章 配置主机集合

主机聚合是一组内容主机。此功能允许您一次对多个主机执行相同的操作。这些操作可以包括安装、删除和更新软件包和勘误表，更改分配的生命周期环境，以及内容视图更改。您可以创建主机集合以满足您的要求，以及您公司的要求。例如，根据功能、部门或业务单元对主机集合中的主机进行分组。

11.1. 创建主机集合

以下流程演示了如何创建主机集合。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Host Collections**。
2. 单击 **New Host Collection**。
3. 添加主机聚合的名称。
4. 清除 **Unlimited Content Hosts**，并在 **Limit** 字段中输入所需的最大主机数量。
5. 添加主机聚合的描述。
6. 点 **Save**。

CLI 过程

- 运行以下命令来创建主机集合：

```
# hammer host-collection create \  
--name "My_Host_Collection" \  
--organization "My_Organization"
```

11.2. 克隆主机集合

以下步骤演示了如何克隆主机集合。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Host Collections**。
2. 在左侧面板中，单击您要克隆的主机集合。
3. 单击 **Copy Collection**。
4. 为克隆的集合指定一个名称。
5. 点 **Create**。

11.3. 删除主机集合

使用以下步骤从 **Satellite** 中删除主机集合。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Host Collections**。
2. 选择您要删除的主机集合。
3. 在 **Select Action** 下，单击 **Remove**。
4. 单击 **Delete** 以删除主机集合。

11.4. 将主机添加到 BULK 中的主机组中

您可以将多个主机添加到主机组中。

前提条件

主机必须注册到 Red Hat Satellite 中，才能将其添加到主机组中。有关注册主机的详情，请参考第 3.3 节“使用全局注册注册主机”。

请注意，如果您将主机添加到主机集合，则 Satellite 审计系统不会记录更改。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > Host Collections。
2. 选择应添加主机的主机集合。
3. 在“主机”选项卡上，选择添加子选项卡。
4. 从表中选择要添加的主机，然后单击 Add Selected。

CLI 过程

- 要在主机集合中添加多个主机，请输入以下命令：

```
# hammer host-collection add-host \  
--host-ids My_Host_ID_1,My_Host_ID_2 \  
--id My_Host_Collection_ID
```

11.5. 从主机组中删除主机

以下步骤演示了如何从主机集合中删除主机。

请注意，如果您从主机集合中删除主机，则不会修改数据库中的主机集合记录，以便 Satellite 审计系统不会记录更改。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > Host Collections。

2. 选择所需的主机集合。
3. 在"主机"选项卡上, 选择"列出/删除"子选项卡。
4. 从主机集合中选择您要删除的主机, 然后单击 **Remove Selected**。

11.6. 将内容添加到主机集合

这些步骤演示了如何将内容添加到 Red Hat Satellite 中的主机集合。

11.6.1. 在主机集合中添加软件包

以下流程演示了如何将软件包添加到主机集合。

前提条件

- 要添加的内容应该在其中一个现有软件仓库中可用, 或者在此流程前添加。
- 内容应提升为分配到主机的环境。

流程

1. 在 **Satellite Web UI** 中, 导航到 **Hosts > Host Collections**。
2. 选择应添加软件包的主机集合。
3. 在 **集合操作** 选项卡上, 单击 **Package Installation**、**Removal** 和 **Update**。
4. 要更新所有软件包, 请单击 **Update All Packages** 按钮以使用默认方法。或者, 选择按钮右侧的下拉图标来选择要使用的方法。选择 **via remote execution** - 自定义第一个 菜单条目将使您进入 **作业调用** 页面, 您可以在其中自定义该操作。

5.

根据需要选择 **Package** 或 **Package Group** 单选按钮。

6.

在提供的字段中，指定软件包或软件包组名称。然后点：

•

install - 使用默认方法安装新软件包。或者，选择按钮右侧的下拉图标并选择要使用的方法。选择 **via remote execution** - 自定义第一个 菜单条目将使您进入 作业调用 页面，您可以在其中自定义该操作。

•

update - 使用默认方法更新主机变量中的现有软件包。或者，选择按钮右侧的下拉图标并选择要使用的方法。选择 **via remote execution** - 自定义第一个 菜单条目将使您进入 作业调用 页面，您可以在其中自定义该操作。

11.6.2. 查看安装的软件包

使用以下步骤查看已安装主机的软件包。

流程

1.

在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts** 并选择主机名称。

2.

在 **Content** 选项卡中，**Packages d** 显示已安装软件包列表。

3.

要查看软件包的详情，请选择该软件包。

•

Details 选项卡显示所选软件包的详情。

•

Files 选项卡列出软件包中包含的文件。

•

Dependencies 选项卡列出软件包的依赖项。

•

Repositories 选项卡列出了包含所选软件包的存储库。

4. 您可以通过 **Library** 或 **Default** 机构 来过滤它们。

11.6.3. 升级软件包

使用以下步骤查看已安装主机的软件包。

流程

1. 在 **Satellite Web UI** 中，进入到 **Hosts > All Hosts**，再选择包含您要升级软件包的主机名称。
2. 在 **Content** 选项卡中，选择 **Packages**。

Status 列显示软件包是可升级还是 **Up-to date**。您无法更新最新的软件包。
3. 从软件包列表中选择您要升级的软件包，然后点击行末尾的垂直 **ellipsis** 图标。
4. 如果要自定义远程执行，选择 **Apply via Remote Execution** 来使用 **Remote Execution**，或通过 自定义的远程执行应用 应用它，例如，设置应应用的时间。
5. 点 **Submit** 升级软件包。

11.6.4. 从主机中删除软件包

使用以下步骤从主机中删除已安装的软件包。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**，再选择包含您要删除的软件包的主机。
2. 在 **Content** 选项卡中，选择 **Packages**。

3. 点您要删除的软件包末尾的垂直 ellipsis 图标，然后选择 **Remove** 选项。
4. 点 **Submit**。

11.6.5. 将勘误添加到主机集合

以下步骤演示了如何将勘误表添加到主机集合。

前提条件

- 要添加的勘误应该在其中一个现有软件仓库中提供，或者在此流程前添加。
- 勘误表应提升到要分配主机的环境。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Host Collections**。
2. 选择应添加勘误表的主机集合。
3. 在 **Collection Actions** 选项卡上，点 **Errata Installation**。
4. 选择您要添加到主机集合中的勘误表，然后单击 **Install Selected** 按钮以使用默认方法。或者，选择按钮右侧的下拉图标来选择要使用的方法。选择 **via remote execution - 自定义** 第一个菜单条目会进入 **作业调用** 页面，您可以在其中自定义操作。

11.6.6. 在单一主机中添加勘误

使用以下步骤为主机添加勘误。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts** 并选择您要向其添加勘误的主机。

2. **点 Content 按钮, 再选择 Errata 选项卡。**
3. **选择您要添加到主机的勘误, 或者选择列表顶部的复选框来添加所有可安装勘误。点击您要从完整列表中删除的任何勘误旁边的复选框。**
4. **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, 或者选择 Apply via custom remote execution (如果要自定义远程执行)。如果没有使用 SSH 连接到目标主机, 请选择 Apply via Katello agent。**
5. **点 Submit。**

11.6.7. 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, 则在新的 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。**

11.6.8. 根据类型和严重性过滤勘误

使用以下步骤根据类型或严重性过滤勘误。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**，再点主机名称。
2. 在 **Contents** 选项卡中，勘误 列出了与所选主机关联的勘误。
3. 点 **Type** 按钮根据类型过滤勘误。
4. 您可以过滤以显示类型为 **Security, Bugfix, 或 Enhancement** 的勘误
5. 点 **严重性**按钮根据严重性 进行过滤。
6. 您可以过滤以显示严重性 **N/A、低、中等、重要或严重**的勘误。
7. 要取消选择您的选择，返回到选项列表，然后再次单击所选选项。

您还可以使用主机页面中的 **勘误** 卡在显示前预先过滤类型为的勘误。

11.6.9. 通过 **Applicable** 和 **Installable** 查看勘误

使用以下步骤通过适用或可安装查看勘误。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 选择主机名。

3. 单击 **Overview** 选项卡。在勘误卡下，有两种勘误类型。
4. 点 **Applicable** 查看应用到主机上安装的软件包的勘误。
5. 点 **Installable** 查看主机内容视图和生命周期环境中可用的适用勘误。
6. 点每个类型下带有勘误数量的链接，以查看该类型的所有可用勘误的列表。
7. 点 每种类型的安全公告、程序错误修复 或 增强功能，仅查看相应的勘误类型。

11.6.10. 从主机组中删除内容

以下流程演示了如何从主机集合中删除软件包。

流程

1. 点 **Hosts > Host Collections**。
2. 单击应当删除软件包的主机集合。
3. 在 **集合操作** 选项卡上，单击 **Package Installation、Removal 和 Update**。
4. 根据需要选择 **Package** 或 **Package Group** 单选按钮。
5. 在提供的字段中，指定软件包或软件包组名称。
6. 单击 **Remove** 按钮，以使用默认方法删除软件包或软件包组。或者，选择按钮右侧的下拉图标并选择要使用的方法。选择 **via remote execution - 自定义** 第一个 菜单条目将使您进入 **作业调用** 页面，您可以在其中自定义该操作。

11.6.11. 更改主机集合的生命周期环境或内容视图

以下步骤演示了如何更改所分配的生命周期环境或主机集合的内容视图。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Host Collection**。
2. 选择应更改生命周期环境或内容视图的主机集合。
3. 在 **Collection Actions** 选项卡上，单击 **Change assigned Life Cycle Environment** 或 **Content View**。
4. 选择要分配到主机集合的生命周期环境。
5. 从列表中选择所需的内容视图。
6. 单击 **分配**。



注意

更改在大约 4 小时内生效。要使更改生效，请在主机上输入以下命令：

```
# subscription-manager refresh
```

您可以使用远程执行同时在多个主机上运行此命令。

第 12 章 配置和设置远程作业

使用本节作为将 **Satellite** 配置为在远程主机上执行作业的指南。

要应用到远程主机的任何命令都必须定义为作业模板。定义了作业模板后，您可以多次执行它。

12.1. 关于在主机上运行作业

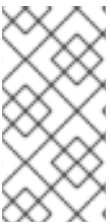
您可以使用 **shell** 脚本或 **Ansible** 任务和 **playbook** 在主机上从胶囊远程运行作业。这称为远程执行。

对于您创建的自定义 **Ansible** 角色或您下载的角色，您必须安装包含胶囊基础操作系统上角色的软件包。在使用 **Ansible** 角色之前，您必须将角色从安装它们的 **Capsule** 中导入到卫星中。

通信通过胶囊服务器进行，这意味着卫星服务器不需要直接访问目标主机，并可扩展为管理许多主机。更多信息请参阅 [第 12.4 节“远程执行的传输模式”](#)。

Satellite 使用 **ERB** 语法作业模板。更多信息请参阅 [附录 A, 模板编写参考](#)。

默认情况下，包括 **shell** 脚本和 **Ansible** 的多个作业模板。如需更多信息，请参阅管理主机中的 [设置作业模板](#)。



注意

任何胶囊服务器基础操作系统是卫星服务器的内部胶囊的客户端，因此本节适用于连接到卫星服务器的任何类型的主机，包括胶囊。

您可以一次在多个主机上运行作业，您可以在命令中使用变量来更精细地控制您运行的作业。您可以使用主机事实和参数来填充变量值。

另外，您可以在运行命令时为模板指定自定义值。

如需更多信息，请参阅管理主机中的 [执行远程作业](#)。

12.2. 远程执行工作流

在主机上运行远程作业时，对于每个主机，卫星将执行以下操作来查找要使用的远程执行胶囊。

卫星仅搜索启用了远程执行功能的胶囊。

1. **Satellite 找到选择了 Remote execution 复选框的主机接口。**
2. **Satellite 查找这些接口的子网。**
3. **卫星查找分配给这些子网的远程执行胶囊。**
4. **从这组胶囊中，卫星选择具有最少运行作业数量的胶囊。通过执行此操作，卫星将确保远程执行胶囊之间作业负载平衡。**

如果您为远程执行启用了 **Prefer 通过 Capsule 注册**，Satellite 会使用主机注册到的 Capsule 运行 REX 作业。

默认情况下，针对远程执行通过 Capsule 注册 **Prefer** 将设置为 **No**。要启用它，在 Satellite web UI 中，进入到 **Administer > Settings**，在 **Content** 标签页中将 **Prefer registered through Capsule for remote execution** 设置为 **Yes**。这样可确保 Satellite 在由注册到的 Capsule 的主机上执行 REX 作业。

如果卫星在此阶段找不到远程执行胶囊，并且启用了 **Fallback 到 Any Capsule** 设置，则卫星会添加另一组胶囊，以从中选择远程执行胶囊。Satellite 从分配给主机的以下类型的胶囊中选择最轻量的载入的胶囊：

- **DHCP、DNS 和 TFTP 胶囊分配给主机的子网**
- **分配给主机的域的 DNS 胶囊**
- **分配给主机的域的 realm Capsule**

- **Puppet 服务器 Capsule**
- **Puppet CA Capsule**
- **OpenSCAP Capsule**

如果卫星在此阶段找不到远程执行胶囊，并且如果启用了 **Enable Global Capsule** 设置，则卫星会从主机组织和位置集合中选择最轻量的远程执行胶囊，以执行远程作业。

12.3. 远程执行的权限

您可以控制哪些角色可以在基础架构中运行哪些作业，包括他们可以目标的主机。远程执行功能提供两个内置角色：

- **远程执行管理器**：可以访问所有远程执行功能和功能。
- **远程执行用户**：只能运行作业。

您可以克隆 **远程执行用户** 角色并自定义其过滤器以增加粒度。如果您使用自定义角色的 **view_job_templates** 权限调整过滤器，则只能根据匹配的作业模板查看并触发作业。您可以使用 **view_hosts** 和 **view_smart_proxies** 权限来限制角色可以看到哪些主机或 Capsule。

execute_template_invocation 权限是一个特殊权限，在执行作业开始前立即检查。此权限定义您可以在特定主机上运行的作业模板。这在指定权限时允许更多粒度。

您可以针对 **Red Hat Satellite** 和 **Capsule** 作为 **Red Hat Satellite** 的主机运行远程执行作业，并具有 **execute_jobs_on_infrastructure_hosts** 权限。默认情况下，标准 **管理器** 和 **站点管理器** 角色具有此权限。如果使用 **Manager** 或 **Site Manager** 角色，或者使用带有 **execute_jobs_on_infrastructure_hosts** 权限的自定义角色，您可以针对注册的 **Red Hat Satellite** 和 **Capsule** 主机执行远程作业。

有关使用角色和权限的更多信息，请参阅管理 **Red Hat Satellite** 中的创建和管理角色。

以下示例显示了 `execute_template_invocation` 权限的过滤器：

```
name = Reboot and host.name = staging.example.com
name = Reboot and host.name ~ *.staging.example.com
name = "Restart service" and host_group.name = webservers
```

使用此示例中的第一行，将 **Reboot** 模板应用到所选主机。使用第二行定义名称以 `.staging.example.com` 结尾的主机池。使用第三行将模板与主机组绑定。



注意

分配给具有这些角色的用户的权限可能会随着时间而改变。如果您已将一些作业调度到将来运行，且权限更改，这可能会导致执行失败，因为权限会在作业执行前立即检查。

12.4. 远程执行的传输模式

您可以将 **Satellite** 配置为使用两种不同传输模式进行远程作业执行。

在 `ssh` 模式的 **Capsule** 上，远程执行使用 **SSH** 服务来传输作业详情。这是默认的传输模式。必须在目标主机上启用并激活 **SSH** 服务。远程执行胶囊必须有权访问目标主机上的 **SSH** 端口。除非具有不同的设置，否则标准 **SSH** 端口为 **22**。



注意

如果您的 **Capsule** 已使用 `pull-mqtt` 模式，而您想要切回到 `ssh` 模式，请运行这个 `satellite-installer` 命令：

```
# satellite-installer --foreman-proxy-plugin-remote-execution-script-mode=ssh
```

在 `pull-mqtt` 模式的 **Capsule** 上，远程执行使用 **Message Queueing Telemetry Transport (mq)** 来发布它从 **Satellite** 服务器接收的作业。主机订阅了 **Capsule** 上的 **MQTT** 代理，用于使用 `yggdrasil pull` 客户端的作业通知。主机收到通知后，它会通过 **HTTPS** 从 **Capsule** 中拉取作业详细信息，运行作业，并将结果报告回 **Capsule**。

要使用 `pull-mqtt` 模式，您必须在 **Capsule** 服务器上启用它，并在目标主机上配置拉取客户端。

其他资源

- 要在 Capsule 服务器上启用拉取模式，请参阅安装 Capsule 服务器中的 [为 Pull Client 配置远程执行](#)。
- 要在现有主机上启用拉取模式，请继续 [第 12.5 节“配置主机以使用 Pull 客户端”](#)。
- 要从 Katello Agent 迁移主机，请参阅 [第 4 章 将主机从 Katello Agent 迁移到远程执行](#)。
- 要在新主机上启用拉取模式，请继续以下流程之一：
 - [第 2.1 节“在 Red Hat Satellite 中创建主机”](#)
 - [第 3.3 节“使用全局注册注册主机”](#)。

12.5. 配置主机以使用 PULL 客户端

对于配置为使用 pull-mqtt 模式的 Capsule，主机可以使用远程执行 pull 客户端订阅远程作业。受管主机不需要 SSH 连接到其 Capsule 服务器。

前提条件

- 您已将主机注册到 Satellite。
- 主机的 Capsule 配置为使用 pull-mqtt 模式。如需更多信息，请参阅安装 [Capsule 服务器](#) 中的 [为 Pull 客户端配置远程执行](#)。
- Red Hat Satellite Client 6 存储库在 Satellite 服务器上启用并同步，并在主机上启用。
- 主机可以使用端口 1883 与 MQTT 进行通信。
- 主机可以通过 HTTPS 与其 Capsule 通信。



注意

创建 `katello-pull-transport-migrate` 软件包，以帮助用户从 Katello Agent 迁移到使用拉取客户端的远程执行。但是，在主机上安装了 Katello 代理并不需要。无论是否安装了 Katello Agent，您可以使用 `katello-pull-transport-migrate`。

流程

1.

在您的主机上安装 `katello-pull-transport-migrate` 软件包：



在 Red Hat Enterprise Linux 8 和 Red Hat Enterprise Linux 9 主机上：

```
# dnf install katello-pull-transport-migrate
```



在 Red Hat Enterprise Linux 7 主机上：

```
# yum install katello-pull-transport-migrate
```

软件包将 `foreman_ygg_worker` 和 `yggdrasil` 安装为依赖项，并在主机上启用拉取模式。主机的 `subscription-manager` 配置和消费者证书用于在主机上配置 `yggdrasil` 客户端，并且启动 `pull` 模式客户端 `worker`。

2.

可选：要验证拉取客户端是否正在运行并正确配置，请检查 `yggdrasild` 服务的状态：

```
# systemctl status yggdrasild
```

3.

可选：安装软件包后，您可以从主机中删除 `katello-agent`。



警告

如果您的主机安装在 Red Hat Virtualization 版本 4.4 或较低版本上，请不要删除 `katello-agent` 软件包，因为删除的依赖关系会破坏主机。

12.6. 创建作业模板

使用这个流程创建作业模板。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 Hosts > Job templates。
2. 单击 New Job Template。
3. 点 Template 选项卡, 在 Name 字段中输入您的作业模板的唯一名称。
4. 选择 Default 以使模板可用于所有组织和位置。
5. 直接在模板编辑器中创建模板, 或者通过单击 Import 从文本文件上传。
6. 可选: 在 Audit Comment 字段中, 添加有关更改的信息。
7. 单击 Job 选项卡, 然后在 Job category 字段中输入您自己的类别, 或者从管理主机的默认作业模板类别中列出的默认类别 中进行选择。
8. 可选: 在 Description Format 字段中输入描述模板。例如, Install package % {package_name}.您还可以在模板中使用 %{template_name} 和 %{job_category}。
9. 从 Provider Type 列表, 为 shell 脚本选择 SSH, 为 Ansible 任务或 playbook 选择 Ansible。
10. 可选: 在 Timeout to kill 字段中, 输入超时值, 以在未完成时终止作业。
11. 可选: 点 Add Input 来定义输入参数。在执行作业时请求参数, 且不必在模板中定义。有关示例, 请参阅帮助 选项卡。

12. **可选**：点 **Foreign input set** 来在此作业中包含其他模板。
13. **可选**：在 **Effective user** 区域中，在命令无法使用默认的 `remote_execution_effective_user` 设置时配置用户。
14. **可选**：如果此模板是要包含在其他模板中的代码片段，点 **Type** 选项卡并选择 **Snippet**。
15. **可选**：如果使用 **Ansible** 供应商，点 **Ansible** 选项卡。选择 **Enable Ansible Callback** 以允许主机发送用于创建配置报告的事实，并在作业完成后返回到 **Satellite**。
16. 点 **Location** 选项卡，并添加您要使用模板的位置。
17. 单击 **Organizations** 选项卡，再添加您要使用该模板的组织。
18. 单击 **Submit** 以保存您的更改。

您可以通过在模板语法中包含其他模板来扩展和自定义模板。如需更多信息，[请参阅管理主机中的模板编写参考](#) 和 [作业模板示例和扩展](#)。

CLI 过程

- 要使用 `template-definition` 文件创建作业模板，请输入以下命令：

```
# hammer job-template create \
--file "Path_to_My_Template_File" \
--job-category "My_Category_Name" \
--name "My_Template_Name" \
--provider-type SSH
```

12.7. 按名称导入 ANSIBLE PLAYBOOK

您可以根据名称从 **Capsule** 上安装的集合将 **Ansible playbook** 导入到 **Satellite**。 **Satellite** 从导入的 **playbook** 创建作业模板，并将模板放在 **Ansible Playbook - 导入** 的作业类别中。

如果您有自定义集合，请将其放在 `/etc/ansible/collections/ansible_collections/My_Namespace/My_Collection` 中。

前提条件

- **Ansible 插件已启用。**
- **您的 Satellite 帐户有一个授予 `import_ansible_playbooks` 权限的角色。**

流程

1. **使用以下 API 请求获取可用的 Ansible playbook :**

```
# curl -X GET -H 'Content-Type: application/json'  
https://satellite.example.com/ansible/api/v2/ansible_playbooks/fetch?  
proxy_id=My_capsule_ID
```

2. **选择您要导入的 Ansible playbook，并记录其名称。**

3. **使用名称导入 Ansible playbook :**

```
# curl -X PUT -H 'Content-Type: application/json' -d '{"playbook_names":  
["My_Playbook_Name"]}'  
https://satellite.example.com/ansible/api/v2/ansible_playbooks/sync?  
proxy_id=My_capsule_ID
```

导入完成后，您将在 **Satellite Web UI** 中获取通知。

后续步骤

- **您可以通过从创建的作业模板执行远程作业来运行 `playbook`。更多信息请参阅 [第 12.22 节“执行远程作业”](#)。**

12.8. 导入所有可用的 ANSIBLE PLAYBOOK

您可以从 **Capsule** 上安装的集合将所有可用的 **Ansible playbook** 导入到 **Satellite**。Satellite 从导入的 **playbook** 创建作业模板，并将模板放在 **Ansible Playbook - 导入** 的作业类别中。

如果您有自定义集合，请将其放在 `/etc/ansible/collections/ansible_collections/My_Namespace/My_Collection` 中。

前提条件

- **Ansible 插件已启用。**
- **您的 Satellite 帐户有一个授予 `import_ansible_playbooks` 权限的角色。**

流程

- **使用以下 API 请求导入 Ansible playbook :**

```
# curl -X PUT -H 'Content-Type: application/json'  
https://satellite.example.com/ansible/api/v2/ansible_playbooks/sync?  
proxy_id=My_capsule_ID
```

导入完成后，您将在 **Satellite Web UI** 中获取通知。

后续步骤

- **您可以通过从创建的作业模板执行远程作业来运行 playbook。更多信息请参阅 [第 12.22 节“执行远程作业”](#)。**

12.9. 将 FALLBACK 配置为 SATELLITE 中的任何 CAPSULE REMOTE EXECUTION SETTING

您可以启用 **Fallback 到 Any Capsule** 设置，以配置卫星，从分配给主机的胶囊列表中搜索远程执行胶囊。如果您需要在没有配置子网的主机上运行远程作业，或者主机子网被分配给没有启用远程执行功能的胶囊，这将非常有用。

如果启用了 **Fallback 到 Any Capsule** 设置，卫星会添加另一组胶囊，以从中选择远程执行胶囊。Satellite 还从分配给主机的所有胶囊集选择最轻量的载入的胶囊，如下所示：

- **DHCP、DNS 和 TFTP 胶囊分配给主机的子网**

- *分配给主机的域的 DNS 胶囊*
- *分配给主机的域的 realm Capsule*
- *Puppet 服务器 Capsule*
- *Puppet CA Capsule*
- *OpenSCAP Capsule*

流程

1. 在 *Satellite Web UI* 中，导航到 **Administer > Settings**。
2. 点 **Remote Execution**。
3. 将 **Fallback** 配置为 **Any Capsule** 设置。

CLI 过程

- 在 *Satellite* 上输入 **hammer settings set** 命令，将 **Fallback** 配置为任何 **Capsule** 设置。要将值设为 **true**，请输入以下命令：

```
# hammer settings set \
--name=remote_execution_fallback_proxy \
--value=true
```

12.10. 在 *SATELLITE* 中配置全局胶囊远程执行设置

默认情况下，无论为主机的子网是否分配了胶囊，*Satellite* 会在主机组织和位置中搜索远程执行胶囊。如果要限制搜索为分配给主机子网的胶囊，可以禁用 **Enable Global Capsule** 设置。

如果启用了 **Enable Global Capsule** 设置，卫星会添加另一组胶囊，以从中选择远程执行胶囊。卫星还从主机组织和位置中的所有胶囊集中选择最轻量的远程执行胶囊，以执行远程作业。

流程

1. 在 **Satellite Web UI** 中，导航到 **Administer > Settings**。
2. 点 **Remote Execution**。
3. 配置 **Enable Global Capsule** 设置。

CLI 过程

- 在 **Satellite** 上输入 **hammer settings set** 命令，以配置 **Enable Global Capsule** 设置。要将值设为 **true**，请输入以下命令：

```
# hammer settings set \  
--name=remote_execution_global_proxy \  
--value=true
```

12.11. 在 **PUSH** 模式中为远程执行作业设置替代目录

默认情况下，**Satellite** 以推送模式将主机上的 **/var/tmp** 目录用于远程执行作业。如果您的主机上的 **/var/tmp** 目录使用 **noexec** 标志挂载，则 **Satellite** 无法在此目录中执行远程执行作业脚本。您可以使用 **satellite-installer** 设置一个替代目录，以便在推送模式下执行远程执行作业。

流程

1. 在主机上，创建一个新目录：

```
# mkdir /My_Remote_Working_Directory
```

2. 从默认的 **/var/tmp** 目录中复制 **SELinux** 上下文：

```
# chcon --reference=/var/tmp/My_Remote_Working_Directory
```

3. 将您的 **Satellite** 服务器或 **Capsule** 服务器配置为使用新目录：

```
# satellite-installer \
--foreman-proxy-plugin-remote-execution-script-remote-working-dir
/My_Remote_Working_Directory
```

12.12. 在 PULL 模式中为远程执行作业设置替代目录

默认情况下，**Satellite** 在拉取模式中使用主机上的 `/run` 目录用于远程执行作业。如果主机上的 `/run` 目录使用 `noexec` 标志挂载，则 **Satellite** 无法在此目录中执行远程执行作业脚本。您可以使用 `yggdrasild` 服务设置在拉取模式中执行远程执行作业的替代目录。

流程

在您的主机上执行以下步骤：

1. 创建新目录：

```
# mkdir /My_Remote_Working_Directory
```

2. 访问 `yggdrasild` 服务配置：

```
# systemctl edit yggdrasild
```

3. 通过在配置中添加以下行来指定替代目录：

```
Environment=FOREMAN_YGG_WORKER_WORKDIR=/My_Remote_Working_Directory
```

4. 重启 `yggdrasild` 服务：

```
# systemctl restart yggdrasild
```

12.13. 更改权限提升方法

默认情况下，基于推送的远程执行使用 `sudo` 从 SSH 用户切换到在主机上执行脚本的有效用户。在某些情况下，您可能需要使用其他方法，如 `su` 或 `dzdo`。您可以在 **Satellite** 设置中全局配置替代方法。

前提条件

- 您的用户帐户分配了一个授予 `view_settings` 和 `edit_settings` 权限的角色。
- 如果要将在 `dzdo` 用于 Ansible 作业，请确保已安装了包含所需 `dzdo become` 插件的 `community.general Ansible` 集合。如需更多信息，请参阅 Ansible [文档中的安装集合](#)。

流程

1. 导航到 **Administer > Settings**。
2. 选择 **Remote Execution** 选项卡。
3. 点 **Effective User Method** 设置的值。
4. 选择新值。
5. 点 **Submit**。

12.14. 为远程执行发布 SSH 密钥

对于 `ssh` 模式中的 `Capsule`，远程执行连接使用 `SSH` 进行身份验证。来自 `Capsule` 的公共 `SSH` 密钥必须分发到您要管理的附加主机。

确保 `SSH` 服务已启用并在主机上运行。配置任何网络或基于主机的防火墙，以启用对端口 `22` 的访问。

使用以下方法之一将公共 `SSH` 密钥从 `Capsule` 分发到目标主机：

1. [第 12.15 节 “为远程执行发布 SSH 密钥”](#)。
2. [第 12.17 节 “使用 Satellite API 为远程执行提供 SSH 密钥”](#)。

3.

[第 12.18 节 “在置备过程中将 Kickstart 模板配置为分布式 SSH 密钥”](#)。

4.

对于新的 Satellite 主机，您可以使用全局注册模板在注册过程中将 SSH 密钥部署到 Satellite 主机。如需更多信息，[请参阅管理主机中的使用全局注册模板将主机注册到 Red Hat Satellite](#)。

默认情况下，Satellite 分发用于远程执行功能的 SSH 密钥到从 Satellite 调配的主机。

如果主机在 Amazon Web Services 上运行，启用密码身份验证。如需更多信息，[请参阅新用户帐户](#)。

12.15. 为远程执行发布 SSH 密钥

要手动分发 SSH 密钥，请完成以下步骤：

流程

- 将 SSH 公钥从 Capsule 复制到您的目标主机：

```
# ssh-copy-id -i ~foreman-proxy/.ssh/id_rsa_foreman_proxy.pub root@client.example.com
```

对您要管理的每个目标主机重复此步骤。

验证

- 要确认密钥已成功复制到目标主机，请在 Capsule 上输入以下命令：

```
# ssh -i ~foreman-proxy/.ssh/id_rsa_foreman_proxy root@client.example.com
```

12.16. 将密码添加到用于远程执行的 SSH 密钥

默认情况下，胶囊使用非密码短语保护的 SSH 密钥来在主机上执行远程作业。您可以按照以下流程使用密码短语保护 SSH 密钥。

流程

-

在 **Satellite 服务器** 或 **Capsule 服务器** 上，使用 `ssh-keygen` 在您的 SSH 密钥中添加密码短语：

```
# ssh-keygen -p -f ~foreman-proxy/.ssh/id_rsa_foreman_proxy
```

后续步骤

- 现在，在主机上运行远程执行作业时，用户必须使用密码短语。

12.17. 使用 SATELLITE API 为远程执行提供 SSH 密钥

要使用 **Satellite API** 从胶囊下载公钥，请在每个目标主机上完成此步骤。

流程

1. 在目标主机上，创建 `~/.ssh` 目录来存储 SSH 密钥：

```
# mkdir ~/.ssh
```

2. 从 **Capsule** 下载 SSH 密钥：

```
# curl https://capsule.example.com:9090/ssh/pubkey >> ~/.ssh/authorized_keys
```

3. 为 `~/.ssh` 目录配置权限：

```
# chmod 700 ~/.ssh
```

4. 为 `authorized_keys` 文件配置权限：

```
# chmod 600 ~/.ssh/authorized_keys
```

12.18. 在置备过程中将 KICKSTART 模板配置为分布式 SSH 密钥

您可以在自定义 **Kickstart** 模板中添加 `remote_execution_ssh_keys` 片断，以便在置备过程中将 SSH 密钥部署到主机。**Satellite** 默认附带的 **Kickstart** 模板包括这个片断。**Satellite** 在调配期间将用于远程执行的 SSH 密钥复制到系统。

流程

- 要在新置备的主机中包含公钥，请在您使用的 Kickstart 模板中添加以下片断：

```
<%= snippet 'remote_execution_ssh_keys' %>
```

12.19. 为 KERBEROS TICKET GRANTING TICKET 配置 KEYTAB

使用这个流程将 Satellite 配置为使用 keytab 获取 Kerberos 票据授予票据。如果没有设置 keytab，则必须手动检索票据。

流程

1. 查找 foreman-proxy 用户的 ID：

```
# id -u foreman-proxy
```

2. 修改 umask 值，以便新文件具有权限 600：

```
# umask 077
```

3. 为 keytab 创建目录：

```
# mkdir -p "/var/kerberos/krb5/user/My_User_ID"
```

4. 创建 keytab，或将现有 keytab 复制到目录中：

```
# cp My_Client.keytab /var/kerberos/krb5/user/My_User_ID/client.keytab
```

5. 将目录所有者改为 foreman-proxy 用户：

```
# chown -R foreman-proxy:foreman-proxy "/var/kerberos/krb5/user/My_User_ID"
```

6. 确保 keytab 文件为只读：

```
# chmod -wx "/var/kerberos/krb5/user/My_User_ID/client.keytab"
```

7.

恢复 SELinux 上下文：

```
# restorecon -RvF /var/kerberos/krb5
```

12.20. 为远程执行配置 KERBEROS 身份验证

您可以使用 Kerberos 身份验证在 Satellite 主机上建立用于远程执行的 SSH 连接。

前提条件

- 在 Kerberos 服务器上注册卫星服务器
- 在 Kerberos 服务器上注册 Satellite 目标主机
- 配置并初始化用于远程执行的 Kerberos 用户帐户
- 确保 Satellite 上的 foreman-proxy 用户具有有效的 Kerberos 票据授予票据

流程

1. 要为远程执行安装并启用 Kerberos 身份验证，请输入以下命令：

```
# satellite-installer --scenario satellite \  
--foreman-proxy-plugin-remote-execution-script-ssh-kerberos-auth true
```

2. 要编辑用于远程执行的默认用户，请在 Satellite Web UI 中进入到 **Administer > Settings** 并点 **Remote Execution** 选项卡。在 **SSH User** 行中，编辑第二列并为 Kerberos 帐户添加用户名。
3. 导航到 **remote_execution_effective_user**，再编辑第二列，为 Kerberos 帐户添加用户名。

验证

- 要确认 Kerberos 身份验证已准备就绪，请在主机上运行远程作业。如需更多信息，请参阅 [管理主机中的执行远程作业](#)。

12.21. 设置作业模板

Satellite 提供了可用于执行作业的默认作业模板。要查看作业模板列表，请导航到 **Hosts > Job templates**。如果要使用没有更改的模板，请执行 [管理主机](#) 中的 **执行远程作业**。

您可以使用默认模板作为自己开发的基础。默认作业模板被锁定进行编辑。克隆模板并编辑克隆。

流程

1. 要克隆模板，请在 **Actions** 列中选择 **Clone**。
2. 输入克隆的唯一名称，然后单击 **Submit** 以保存更改。

作业模板使用嵌入式 Ruby (ERB) 语法。有关编写模板的更多信息，[请参阅管理主机中的模板编写参考](#)。

Ansible 注意事项

要创建 Ansible 作业模板，请使用以下流程，而不使用 ERB 语法，请使用 YAML 语法。使用 `---` 开始模板。您可以将 Ansible playbook YAML 文件嵌入到作业模板正文中。您还可以添加 ERB 语法来自定义 YAML Ansible 模板。您也可以在卫星中导入 Ansible playbook。如需更多信息，[请参阅管理主机中的同步存储库模板](#)。

参数变量

在运行时，作业模板可以接受您为主机定义的参数变量。请注意，只有主机编辑页面中的 **Parameters** 选项卡上显示的参数可以用作作业模板的输入参数。

12.22. 执行远程作业

您可以对一个或多个主机执行基于作业模板的作业。

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 **Monitor > Jobs** 并点 **Run job**。



注意

Satellite 6.13 具有新的作业向导。如果要使用来自以前 **Satellite** 版本的作业向导，请点击作业向导右上角的 **Use legacy form**。请注意，本指南只适用于新的作业向导。

2. 选择 **作业** 类别和您要使用的 **作业模板**，然后单击 **Next**。
3. 选择您要在其上运行作业的主机。如果您没有选择任何主机，该作业将在当前上下文中看到的所有主机上运行。



注意

如果要选择主机组及其所有子组，则不能选择主机组，因为该作业只会直接在该组的主机上运行，而不在子组中的主机上运行。反之，您必须选择主机组及其所有子组，或者使用此搜索查询：

```
hostgroup_fullname ~ "My_Host_Group*"
```

将 **My_Host_Group** 替换为顶级主机组的名称。

4. 如果需要，请为作业模板提供输入。不同的模板具有不同的输入，一些模板没有任何输入。输入所有必要的输入后，点 **Next**。
5. 可选：要为作业配置高级设置，请填写高级字段。要了解更多有关高级设置的信息，请参阅第 12.23 节“作业向导中的高级设置”。
6. 单击 **Next**。
7. 作业调度时间。



要立即执行任务，请保留预先选择的 **Immediate** 执行

交互式执行任务，而不是由计划执行的 immediate 执行。

- 要在以后执行任务，请选择 **future execution**。
 - 要定期执行任务，请选择 **Recurring execution**。
8. 可选：如果您选择将来或重复执行，请选择 **Query** 类型，否则点 **Next**。
- 静态查询 意味着作业在您提供的主机的确切列表上执行。
 - 动态查询 意味着仅在执行任务前评估主机列表。如果您根据某些过滤器输入主机列表，则结果可以与首次使用该过滤器时不同。
- 选择查询类型后，点 **Next**。
9. 可选：如果您选择将来或重复执行，请提供额外的详情：
- 对于未来执行，请输入 **Starts at date and time**。您还可以选择 **Starts before date and time**。如果作业无法在该时间之前启动，它将被取消。
 - 对于 **Recurring execution**，请选择结束重复任务的开始日期和时间、频率以及结束重复任务的条件。您可以选择从不结束、特定时间结束的重复，或者在指定数量的重复后结束。您还可以添加 **Purpose** - 跟踪作业的特殊标签。一次只能有一个具有给定目的的活动作业。
- 输入所需信息后，点 **Next**。
10. 查看作业详情。您可以选择返回到作业向导中的任何部分并编辑信息。
11. 单击 **Submit** 以计划执行的作业。

1.

在 **Satellite** 中输入以下命令：

```
# hammer settings set \  
--name=remote_execution_global_proxy \  
--value=false
```

2.

查找要使用的作业模板的 ID：

```
# hammer job-template list
```

3.

显示模板详情以查看模板所需的参数：

```
# hammer job-template info --id My_Template_ID
```

4.

使用自定义参数执行远程作业：

```
# hammer job-invocation create \  
--inputs My_Key_1="My_Value_1",My_Key_2="My_Value_2",... \  
--job-template "My_Template_Name" \  
--search-query "My_Search_Query"
```

将 `My_Search_Query` 替换为定义主机的过滤器表达式，如 `"name ~ My_Pattern"`。有关使用 `hammer` 执行远程命令的更多信息，请输入 `hammer job-template --help` 和 `hammer job-invocation --help`。

12.23. 作业向导中的高级设置

有些作业模板要求您输入高级设置。某些高级设置仅对某些作业模板可见。以下是常规高级设置列表。

SSH 用户

用于通过 **SSH** 连接到主机的用户。

有效用户

用于执行该作业的用户。默认情况下，它是 **SSH** 用户。如果它与 **SSH** 用户不同，则可使用 `su` 或 `sudo` 来切换帐户。

Description

作业的描述模板。

要终止的超时

作业开始的时间（以秒为单位），如果还没有完成，则应终止作业的时间（以秒为单位）。

抓取时间

客户端没有获取作业，则会取消作业的时间（以秒为单位）。此设置仅适用于使用 `pull-mqtt` 传输的主机。

密码

如果 SSH 身份验证方法是密码而不是 SSH 密钥，则使用。

私钥密码短语

如果 SSH 密钥受密码短语保护，则使用。

有效用户密码

如果有效用户与 `ssh` 用户不同，则使用。

并发级别

定义一次执行的最大作业数。当大量主机上执行作业时，这可以防止过载系统资源。

时间范围

将远程执行分发到所选秒数。作业会定期启动一个间隔，以适应给定的时间窗。与并发级别类似，这也可能会阻止过载系统资源。

执行排序

决定在主机上执行任务的顺序。它可以是字母顺序或随机化。

12.24. 为主机调度重复的 ANSIBLE 作业

您可以调度重复的作业，以便在主机上运行 Ansible 角色。

前提条件

- 确保您有 `view_foreman_tasks`、`view_job_invocations` 和 `view_recurring_logics` 权限。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**，再选择您要在其上执行远程作业的目标主机。
2. 在 **Ansible** 选项卡中，选择 **Jobs**。
3. 点 **Scheduleduplicate job**。
4. 在 **Create New Recurring Ansible Run** 窗口中定义第一次运行的重复频率、开始时间和日期。
5. 点 **Submit**。
6. 可选：在主机概述中查看调度的 **Ansible** 作业，或者导航到 **Ansible > Jobs**。

12.25. 为主机组调度重复的 ANSIBLE 作业

您可以调度重复的作业，以在主机组上运行 **Ansible** 角色。

流程

1. 在 **Satellite Web UI** 中，导航到 **Configure > Host groups**。
2. 在 **Actions** 列中，为您要为其调度 **Ansible** 角色运行的主机组选择 **Configure Ansible Job**。
3. 点 **Scheduleduplicate job**。
4. 在 **Create New Recurring Ansible Run** 窗口中定义第一次运行的重复频率、开始时间和日期。

5. **点 Submit.**

12.26. 监控作业

您可以在作业运行时监控作业的进度。这可帮助执行任何可能需要的故障排除。

Ansible 作业在 100 个主机的批处理上运行，因此您无法取消在特定主机上运行的作业。只有在 Ansible playbook 在批处理中的所有主机上运行后，作业才会完成。

流程

1. **在 Satellite Web UI 中，导航到 Monitor > Jobs。如果您使用 Execute now 设置触发作业，则会自动显示此页面。要监控调度的作业，请导航到 Monitor > Jobs 并选择您要检查的作业运行。**
2. **在 Job 页面上，单击 Hosts 选项卡。这将显示在其上运行该作业的主机列表。**
3. **在 Host 列中，单击您要检查的主机的名称。这将显示 Commands 页面的详细信息，您可以在其中实时监控作业执行。**
4. **随时单击 Back to Job，以返回到 Job Details 页面。**

CLI 过程

1. **查找作业的 ID：**

```
# hammer job-invocation list
```

2. **监控作业输出：**

```
# hammer job-invocation output \
--host My_Host_Name \
--id My_Job_ID
```

3.

可选：要取消作业，请输入以下命令：

```
# hammer job-invocation cancel \  
--id My_Job_ID
```

12.27. 在 CAPSULE 上设置作业速率限制

您可以一次限制 Capsule 上的活动作业的最大数量，以防止性能激增。该作业从时间 Capsule 开始尝试通知主机有关作业，直到作业在主机上完成为止。

作业速率限制只适用于基于 mqtt 的作业。



注意

活动作业的最佳最大数量取决于您的 Capsule 服务器的计算资源。默认情况下，活跃作业的最大数量是无限的。

流程

•

使用 `satellite-installer` 设置活跃作业的最大数量：

```
satellite-installer \  
--foreman-proxy-plugin-remote-execution-script-mqtt-rate-limit MAX_JOBS_NUMBER
```

例如：

```
satellite-installer \  
--foreman-proxy-plugin-remote-execution-script-mqtt-rate-limit 200
```

第 13 章 SATELLITE 中的主机状态

在 *Satellite* 中，每个主机都有一个全局状态，表示哪些主机需要注意。每个主机也具有代表特定功能状态的子状态。随着子状态的任何更改，全局状态都会被重新计算，结果由所有子状态的状态决定。

13.1. 主机全局状态概述

全局状态代表特定主机的整体状态。状态可以有三个可能的值之一：OK、warning 或 Error。您可以在 *Hosts Overview* 页面中找到全局状态。状态显示主机名旁边的小图标，并且具有与状态对应的颜色。将鼠标悬停在图标上可呈现带有子状态信息的工具提示，以快速找到更多详情。要查看主机的全局状态，请在 *Satellite Web UI* 中导航至 *Hosts > All Hosts*。

确定

任何子状态都没有报告错误。此状态以颜色绿色突出显示。

警告

虽然没有检测到错误，但有些子状态会引发一个警告。例如，即使主机被配置为发送报告，主机也不会对主机进行配置管理报告。最好调查任何警告以确保部署健康。这个状态使用颜色黄色突出显示。

Error

有些子状态报告失败。例如，运行包含一些失败的资源。此状态以颜色红色突出显示。

搜索语法

如果要根据其状态搜索主机，请使用管理 Red Hat Satellite 中的 [搜索和书签](#) 中介绍的语法在 *Satellite* 中进行搜索，然后使用以下与状态相关的示例进行搜索：

搜索具有 OK 状态的主机：

```
global_status = ok
```

搜索保留的所有主机：

```
global_status = error or global_status = warning
```

13.2. 主机子状态概述

子状态仅监控主机功能的一部分。

要查看主机的子状态，在 **Satellite web UI** 中进入 **Hosts > All Hosts**，然后单击您要检查的完整状态的主机。您可以查看主机名称和 **Host** 状态卡中的主机子状态旁边的全局主机状态。

每个子状态都有自己的一组可能值，它们映射到三个全局状态值。

下面列出了 **Satellite** 包含的子状态。

配置

只有 **Satellite** 使用 **Ansible**、**Puppet** 或 **Salt** 等配置管理系统时，此子状态才相关。

可能的值：

标签	全局主机状态
禁用警报	确定
Active	确定
待处理	确定
没有更改	确定
没有报告	确定/警告
不同步	警告
Error	Error

有关此子状态值的附加信息：

- **活动**：在上一次配置期间，会应用一些资源。
- **待定**：在最后一次配置期间，将应用一些资源，但您的配置管理集成已配置为以 **noop** 模式运行。

- **没有更改** : 在最后的配置中, 不会更改。
- **没有报告** : 这可以是 **Warning** 或 **OK** 状态。如果没有报告, 但主机使用关联的 **Capsule** 进行配置管理, 或者 **always_show_configuration_status** 设置设为 **true**, 它将映射到 **Warning**。否则, 它会映射到 **OK**。
- **错误** : 这表示配置过程中出现错误。例如, 配置运行无法安装软件包。
- **不同步** : 根据 **outofsync_interval** 设置, 在预期间隔内不会收到配置报告。报告由原始卷标识, 可以根据它有不同间隔。

Build

此子状态仅与通过全局注册注册的 **Satellite** 或主机置备的主机相关。

可能的值 :

标签	全局主机状态	数字值
已安装	确定	0
待处理的安装	确定	1
令牌已过期	Error	2
安装错误	Error	3

Compliance

指明主机是否与 **OpenSCAP** 策略兼容。

可能的值 :

标签	全局主机状态	数字值
Compliant	确定	0

标签	全局主机状态	数字值
inconclusive	警告	1
至少有一个不合规	Error	2

OVAL 扫描

指明主机上是否有发现漏洞

可能的值：

标签	全局主机状态	数字值
没有找到漏洞	确定	0
发现漏洞	警告	1
找到可用补丁的漏洞	Error	2

æ%.§èjCE

最后完成的远程执行作业的状态。

可能的值：

标签	全局主机状态	数字值
最后执行成功 / No execution completed	确定	0
最后执行失败	Error	1
未知执行状态	确定	2 或 3
最后一次执行已取消	确定	4

清单 (Inventory)

指明主机是否同步到 Red Hat Hybrid Cloud Console。Satellite 服务器执行同步本身，但仅将基本信息上传到红帽混合云控制台。

可能的值：

标签	全局主机状态	数字值
主机没有上传到您的 RH 云清单中	警告	0
成功上传到您的 RH 云清单	确定	1

Insights

指明主机是否同步到 Red Hat Hybrid Cloud Console。此同步由主机执行。主机上传比 Satellite 服务器更多的信息。

可能的值：

标签	全局主机状态	数字值
报告	确定	0
Not reporting	Error	1

勘误

指明 Errata 是否在主机上可用。

可能的值：

标签	全局主机状态	数字值
最新的	确定	0
Unknown	警告	1
所需的勘误	Error	2
所需的安全勘误	Error	3

订阅

指明主机是否具有有效的 RHEL 订阅。

可能的值：

标签	全局主机状态	数字值
完全授权	确定	0
部分授权	警告	1
未权利	Error	2
Unknown	警告	3
已取消订阅的 hypervisor	警告	4
启用 SCA	确定	5

服务级别

指明是否可以附加与指定 Service level syspurpose 值匹配的订阅。

可能的值：

标签	全局主机状态	数字值
Unknown	确定	0
不匹配	警告	1
匹配	确定	2
未指定	确定	3

角色

指明是否可以附加与您指定的 Role syspurpose 值匹配的订阅。

可能的值：

标签	全局主机状态	数字值
Unknown	确定	0
不匹配	警告	1
匹配	确定	2
未指定	确定	3

使用方法

指明是否可以附加与您的指定 *Usage syspurpose* 值匹配的订阅。

可能的值：

标签	全局主机状态	数字值
Unknown	确定	0
不匹配	警告	1
匹配	确定	2
未指定	确定	3

附加组件

指明是否可以附加与指定 *Addons syspurpose* 值匹配的订阅。

可能的值：

标签	全局主机状态	数字值
Unknown	确定	0
不匹配	警告	1
匹配	确定	2
未指定	确定	3

系统目的

指明是否可以附加与您指定的 `syspurpose` 值匹配的订阅。

可能的值：

标签	全局主机状态	数字值
Unknown	确定	0
不匹配	警告	1
匹配	确定	2
未指定	确定	3

trace

指明主机是否需要重启或进程重启。

可能的值：

标签	全局主机状态	数字值
Unknown	警告	-1
最新的	确定	0
所需的进程重启	Error	1
所需的重启	Error	2

搜索语法

如果要根据其子状态搜索主机，请使用 [管理 Satellite 指南中的搜索和书签一章](#)中介绍的语法在 `Satellite` 中进行搜索，然后使用以下与状态相关的示例进行搜索：

您可以根据上次报告的状态搜索主机的配置子状态。

例如，查找至少有一个待处理资源的主机：

```
status.pending > 0
```

查找在最后一次运行时重启一些服务的主机：

```
status.restarted > 0
```

要查找最后一次运行的有值得注意的主机，这可能代表发生了以下情况：

```
status.interesting = true
```

第 14 章 同步模板存储库

在 **Satellite** 中，您可以在卫星服务器和本地目录之间同步作业模板、作业模板、报告模板和分区表模板。在本章中，**Git** 存储库用于演示目的。

本节详细介绍了安装和配置 **TemplateSync** 插件和执行导出和导入任务的工作流。

14.1. 启用 **TEMPLATESYNC** 插件

流程

1. 要在 **Satellite** 服务器中启用插件，请输入以下命令：

```
# satellite-installer --enable-foreman-plugin-templates
```

2. 要验证插件是否已正确安装，请确保 **Administer > Settings** 包含 **TemplateSync** 菜单。

14.2. 配置 **TEMPLATESYNC** 插件

在 **Satellite Web UI** 中，导航到 **Administer > Settings > TemplateSync** 来配置插件。下表说明了属性行为。请注意，一些属性仅用于导入或导出任务。

表 14.1. 同步模板插件配置

参数	API 参数名称	导入的含义	导出的含义
associate	associate 接受的值： always、new、never	将模板与基于元数据的操作系统、组织和位置相关联。	N/A
分支	branch	指定 Git 存储库中要从中读取的默认分支。	指定 Git 存储库中要写入的默认分支。
dirname	dirname	指定要从存储库中读取的子目录。	指定要写入的存储库下的子目录。
Filter	filter	仅导入带有与此正则表达式匹配的名称的模板。	仅导出带有与此正则表达式匹配的名称的模板。

参数	API 参数名称	导入的含义	导出的含义
强制导入	force	导入的模板会覆盖具有相同名称的锁定模板。	N/A
锁定模板	锁定	除非启用了 Force import ，否则在导入具有相同名称的新模板时，不要覆盖现有模板。	N/A
元数据导出模式	metadata_export_mode 接受的值： 刷新、保留、删除	N/A	定义导出时如何处理元数据： <ul style="list-style-type: none"> ● refresh - 从模板内容中删除现有元数据，并根据当前的分配和属性生成新的元数据。 ● keep - 保留现有元数据。 ● 删除 - 没有元数据的导出模板。如果要手动添加元数据，可使用。
negate	negate 接受的值： true,false	导入模板忽略 filter 属性。	导出模板忽略 filter 属性。
prefix	prefix	如果模板名称没有以前缀开头，请将指定的字符串添加到模板的开头。	N/A
仓库	repo	定义要从中同步的存储库的路径。	定义要导出到的存储库的路径。
详细程度	详细 接受的值： true,false	启用为此操作向日志中写入详细消息。	N/A

14.3. 使用存储库源

您可以使用现有存储库或本地目录来将模板与 *Satellite* 服务器同步。

14.3.1. 使用现有存储库同步模板

使用这个流程在 **Satellite** 服务器和现有存储库之间同步模板。

流程

1. 如果要使用 **HTTPS** 连接到存储库，并在 **Git** 服务器上使用自签名证书颁发机构(CA)：

- a. 在 `/usr/share/foreman/` 目录下创建一个新目录，以存储证书的 **Git** 配置：

```
# mkdir --parents /usr/share/foreman/.config/git
```

- b. 在新目录中创建一个名为 **config** 的文件：

```
# touch /usr/share/foreman/.config/git/config
```

- c. 允许 **foreman** 用户访问 **.config** 目录：

```
# chown --recursive foreman /usr/share/foreman/.config
```

- d. 使用您自签名 **CA** 证书的路径为 **foreman** 用户更新 **Git** 全局配置：

```
# sudo --user foreman git config --global http.sslCAPath Path_To_CA_Certificate
```

2. 如果要使用 **SSH** 连接到存储库：

- a. 如果您还没有 **SSH** 密钥对，请创建它。不要指定密码短语。

```
# sudo --user foreman ssh-keygen
```

- b. 使用 **Satellite** 中的公钥配置您的版本控制服务器，该密钥位于 `/usr/share/foreman/.ssh/id_rsa.pub` 中。

- c. 以 **foreman** 用户身份接受 **Git SSH** 主机密钥：

```
# sudo --user foreman ssh git.example.com
```

3. 在 **TemplateSync** 选项卡中配置 **TemplateSync** 插件设置。
 - a. 更改 **Branch** 设置，以匹配 **Git** 服务器上的目标分支。
 - b. 更改 **Repo** 设置以匹配 **Git** 存储库。例如，对于位于 `git@git.example.com/templates.git` 的存储库，将设置设置为 `git@git.example.com/templates.git`。

14.3.2. 将模板与本地目录同步

如果您在本地目录中配置了版本控制系统，则与本地目录同步模板非常有用。这样，您可以编辑模板并跟踪目录中编辑历史记录。您也可以在编辑模板后同步对卫星服务器的更改。

前提条件

- 每个模板必须包含模板所属的位置和组织。这适用于所有模板类型。在导入模板前，请确保将以下部分添加到模板中：

```
<%#
kind: provision
name: My_Provisioning_Template
oses:
- My_first_OS
- My_second_OS
locations:
- My_first_Location
- My_second_Location
organizations:
- My_first_Organization
- My_second_Organization
%>
```

流程

1. 在 `/var/lib/foreman` 中，创建一个用于存储模板的目录：

```
# mkdir /var/lib/foreman/My_Templates_Dir
```



注意

您可以将模板放在 `/var/lib/foreman` 以外的自定义目录中，但必须确保 Foreman 服务可以读取其内容。目录必须具有正确的文件权限和 `foreman_lib_t` SELinux 标签。

2.

将新 `templates` 目录的所有者改为 `foreman` 用户：

```
# chown foreman /var/lib/foreman/My_Templates_Dir
```

3.

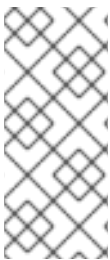
更改 `TemplateSync` 选项卡上的 `Repo` 设置，以匹配 `/var/lib/foreman/My_Templates_Dir/` 目录。

14.4. 导入和导出模板

您可以使用 `Satellite Web UI`、`hammer CLI` 或 `Satellite API` 导入和导出模板。卫星 API 调用使用基于角色的访问控制系统，该工具可让任务以任何用户身份执行。您可以将模板与版本控制系统（如 `Git`）或本地目录同步。

14.4.1. 导入模板

您可以从您选择的存储库导入模板。您可以使用不同的协议指向您的存储库，如 `/tmp/dir`、`git://example.com`、`https://example.com` 和 `ssh://example.com`。



注意

`Satellite` 提供的模板已锁定，您无法默认导入它们。要覆盖此行为，请将 `TemplateSync` 菜单中的 `Force import` 设置改为 `yes`，或将 `force` 参数 `-d '{ "force": "true" }'` 改为 `import` 命令。

前提条件

-

每个模板必须包含模板所属的位置和组织。这适用于所有模板类型。在导入模板前，请确保将以下部分添加到模板中：

```
<%#
kind: provision
name: My_Provisioning_Template
oses:
```



```

- My_first_OS
- My_second_OS
locations:
- My_first_Location
- My_second_Location
organizations:
- My_first_Organization
- My_second_Organization
%>

```

要使用 CLI 而不是 Satellite Web UI，请参阅]。要使用 API，请参阅 [xref:api_Importing_Templates_managing-hosts\[\]](#)。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > Sync Templates。
2. 点 **Import**。
3. 每个字段填充在 **Administer > Settings > TemplateSync** 中配置的值。根据需要更改您要导入的模板的值。有关每个字段的详情请参考第 14.2 节“配置 TemplateSync 插件”。
4. 点 **Submit**。

Satellite Web UI 显示导入的状态。状态不是持久性；如果您离开状态页面，则无法返回它。

CLI 过程

- 要从存储库导入模板，请输入以下命令：

```

$ hammer import-templates \
--branch "My_Branch" \
--filter '.*Template Name$' \
--organization "My_Organization" \
--prefix "[Custom Index]" \
--repo "https://git.example.com/path/to/repository"

```

为了更好地索引和管理模板，请使用 **--prefix** 为您的模板设置类别。要从大型存储库中选择某些模板，请使用 **--filter** 来定义您要导入的模板的标题。例如，**--filter '.*Ansible Default\$'** 导

入各种 Ansible 默认模板。

API 流程

1. 将 POST 请求发送到 `api/v2/templates/import` :

```
# curl -H "Accept:application/json" \  
-H "Content-Type:application/json" \  
-u login:password \  
-k https://satellite.example.com/api/v2/templates/import \  
-X POST
```

如果导入成功，您会收到 `{"message":"Success"}`。

14.4.2. 导出模板

使用这个流程将模板导出到 git 存储库。

要使用 CLI 而不是 Satellite Web UI，请参阅 [\[\]](#)。要使用 API，请参阅 [xref:api_Exporting_Templates_managing-hosts\[\]](#)。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > Sync Templates。
2. 单击 **Export**。
3. 每个字段填充在 Administer > Settings > TemplateSync 中配置的值。根据需要更改您要导出的模板的值。有关每个字段的详情请参考 [第 14.2 节“配置 TemplateSync 插件”](#)。
4. 点 **Submit**。

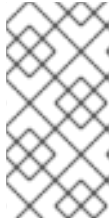
Satellite Web UI 显示导出的状态。状态不是持久性；如果您离开状态页面，则无法返回它。

CLI 过程

1.

要将模板导出到存储库，请输入以下命令：

```
hammer export-templates \
--organization "My_Organization" \
--repo "https://git.example.com/path/to/repository"
```



注意

此命令克隆存储库，在提交中进行更改，并推送回存储库。您可以使用 `--branch "My_Branch"` 选项，将模板导出到特定的分支。

API 流程

1.

将 **POST** 请求发送到 `api/v2/templates/export`：

```
# curl -H "Accept:application/json" \
-H "Content-Type:application/json" \
-u login:password \
-k https://satellite.example.com/api/v2/templates/export \
-X POST
```

如果导出成功，您会收到 `{"message":"Success"}`。



注意

您可以使用 `-d` 参数在请求中指定它们来覆盖默认的 API 设置。以下示例将模板导出到 `git.example.com/templates` 存储库：

```
# curl -H "Accept:application/json" \
-H "Content-Type:application/json" \
-u login:password \
-k https://satellite.example.com/api/v2/templates/export \
-X POST \
-d "{\"repo\":\"git.example.com/templates\"}"
```

第 15 章 管理软件包

您可以使用 **Satellite** 在主机上安装、升级和删除软件包，并在主机上启用或禁用存储库。

15.1. 在主机上启用和禁用存储库

使用这个流程在主机上为 **Satellite** 启用和禁用存储库。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**,
2. 选择主机名。
3. 单击 **Content** 选项卡。
4. 单击 **Repository Sets** 选项卡。
5. 点垂直 ellipsis 选择 **Override to disabled** 或 **Override to enabled to disable** 或 **enable repository on hosts**.

15.2. 在主机上安装软件包

使用这个流程，使用 **Satellite Web UI** 查看并安装软件包。用于安装的软件包列表取决于分配给主机的内容视图和生命周期环境。

流程

1. 在 **Satellite Web UI** 中，进入到 **Hosts > All Hosts** 并选择您要安装的软件包的主机。
2. 在 **Content** 选项卡中，选择 **Packages** 选项卡。

3. 在升级按钮旁边的垂直 ellipsis 图标上，单击 **Install Packages**。
4. 在 **Install packages** 窗口中，选择要在主机上安装的软件包或软件包。
5. 点 **Install**。

默认情况下，软件包使用远程执行安装。有关运行远程执行作业的更多信息，请参阅管理主机中的配置和设置远程作业。

按照以下说明，以 JSON 格式创建 API 请求的正文。

API 流程

1. 创建 "job_invocation" 对象，并将其余正文放在此对象内。
2. 使用字符串类型的 "package" 字段创建 "inputs" 对象，指定您要安装的软件包。如果您要指定多个软件包，请使用空格分开它们。
3. 为字符串类型创建一个 "feature" 字段，其值为 "katello_package_install"。
4. 为字符串类型创建一个 "search_query" 字段，并输入与您要安装软件包的主机匹配的搜索查询。
5. 可选：如果要以特定用户身份安装软件包，请使用字符串类型的以下字段创建一个 ssh 对象：
 - 使用 ssh 用户名称的 "effective_user"
 - 如果需要此密码，使用 ssh 用户的密码 "effective_user_password"
6. 可选：如果要稍后安装软件包，请创建 "scheduling" 对象。该对象可以包含 ISO 8601 格式的字符串类型中的一个或两个字段，包括日期、时间和时区：

- **"start_at"** - 设置安装软件包的时间
- **"start_before"** - 设置安装软件包的最新时间。如果此时无法安装软件包，则会取消这个操作。

如果省略了时间，则默认为 **00:00:00**。如果省略了时区，则默认为 **UTC**。

7. 可选：如果要限制同时运行作业的主机数量，请使用整数类型的 **"concurrency_level"** 字段创建 **"concurrency_control"** 对象。将主机数量分配为字段值。
8. 可选：如果要稍后安装软件包，并且您希望在运行作业时评估主机搜索查询，请使用 **"dynamic_query"** 值创建字符串类型的 **"targeting_type"** 字段。如果您预期搜索查询的结果在运行作业时不同，因为主机状态改变，则这非常有用。如果省略此字段，则默认为 **"static_query"**。
9. 使用创建的正文发送 **POST** 请求到 **Satellite** 服务器的 **/api/job_invocations** 端点，并使用 **python** 等工具来查看格式化的响应。

API 请求示例：

```
curl https://satellite.example.com/api/job_invocations \
-H "content-type: application/json" \
-X POST \
-d @Path_To_My_API_Request_Body \
-u My_Username:My_Password | python3 -m json.tool
```

验证

- 在 **Satellite Web UI** 中，进入到 **Monitor > Jobs** 并查看调度或完成远程执行作业的报告，以便在所选主机上安装软件包。

API 请求正文示例

```
{
  "job_invocation": {
    "concurrency_control": {
```

```
"concurrency_level" : 100
},
"feature" : "katello_package_install",
"inputs" : {
  "package" : "nano vim"
},
"scheduling" : {
  "start_at" : "2023-09-21T19:00:00+00:00",
  "start_before" : "2023-09-23T00:00:00+00:00"
},
"search_query" : "*",
"ssh" : {
  "effective_user" : "My_Username",
  "effective_user_password" : "My_Password"
},
"targeting_type" : "dynamic_query"
}
}
```

15.3. 升级主机上的软件包

您可以在 **Satellite Web UI** 中批量升级主机上的软件包。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 单击 **Content** 选项卡，然后单击 **Packages** 选项卡。
4. 从 **Status** 列表中选择 **Upgradable**。
5. 从 **Upgradable to column** 中的下拉菜单中选择 **upgrade version**。
6. 选择您要升级的软件包。

7. 点 **Upgrade** 按钮。远程执行作业完成后，您会收到 REX 作业通知。

按照以下说明，以 JSON 格式创建 API 请求的正文。

API 流程

1. 创建 "job_invocation" 对象，并将其余正文放在此对象内。
2. 使用字符串类型的 "package" 字段创建 "inputs" 对象，指定您要更新的软件包。如果您要指定多个软件包，请使用空格分开它们。
3. 为字符串类型创建一个 "feature" 字段，其值为 "katello_package_update"。
4. 为字符串类型创建一个 "search_query" 字段，并输入与要更新软件包的主机匹配的搜索查询。
5. 可选：如果要以特定用户身份更新软件包，请使用字符串类型的以下字段创建一个 ssh 对象：
 - 使用 ssh 用户名称的 "effective_user"
 - 如果需要此密码，使用 ssh 用户的密码 "effective_user_password"
6. 可选：如果要稍后更新软件包，请创建 "scheduling" 对象。该对象可以包含 ISO 8601 格式的字符串类型中的一个或两个字段，包括日期、时间和时区：
 - "start_at" - 设置更新软件包的时间
 - "start_before" - 设置更新软件包的最新时间。如果此时无法更新软件包，则会取消这个操作。

如果省略了时间，则默认为 00:00:00。如果省略了时区，则默认为 UTC。

7. 可选：如果要限制同时运行作业的主机数量，请使用整数类型的 "concurrency_level" 字段创建 "concurrency_control" 对象。将主机数量分配为字段值。
8. 可选：如果要稍后更新软件包，并且您希望在运行作业时评估主机搜索查询，请使用 "dynamic_query" 值创建字符串类型的 "targeting_type" 字段。如果您预期搜索查询的结果在运行作业时不同，因为主机状态改变，则这非常有用。如果省略此字段，则默认为 "static_query"。
9. 使用创建的正文发送 POST 请求到 Satellite 服务器的 /api/job_invocations 端点，并使用 python 等工具来查看格式化的响应。

API 请求示例：

```
curl https://satellite.example.com/api/job_invocations \
-H "content-type: application/json" \
-X POST \
-d @Path_To_My_API_Request_Body \
-u My_Username:My_Password | python3 -m json.tool
```

验证

- 在 Satellite Web UI 中，进入到 Monitor > Jobs 并查看调度或完成远程执行作业的报告，以更新所选主机上的软件包。

API 请求正文示例

```
{
  "job_invocation": {
    "concurrency_control": {
      "concurrency_level": 100
    },
    "feature": "katello_package_update",
    "inputs": {
      "package": "nano vim"
    },
    "scheduling": {
      "start_at": "2023-09-21T19:00:00+00:00",
      "start_before": "2023-09-23T00:00:00+00:00"
    },
    "search_query": "*"
  }
}
```

```
"ssh": {  
  "effective_user": "My_Username",  
  "effective_user_password": "My_Password"  
},  
"targeting_type": "dynamic_query"  
}  
}
```

15.4. 从主机中删除软件包

您可以使用 **Satellite Web UI** 从主机中删除软件包。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > All Hosts**。
2. 点您要修改的主机的名称。
3. 单击 **Content** 选项卡，然后单击 **Packages** 选项卡。
4. 单击您要删除的软件包的垂直 **ellipsis**，然后选择 **Remove**。远程执行作业完成后，您会收到 **REX** 作业通知。

按照以下说明，以 **JSON** 格式创建 **API** 请求的正文。

API 流程

1. 创建 **"job_invocation"** 对象，并将其余正文放在此对象内。
2. 使用字符串类型的 **"package"** 字段创建 **"inputs"** 对象，指定您要删除的软件包。如果您要指定多个软件包，请使用空格分开它们。
3. 为字符串类型创建一个 **"feature"** 字段，其值为 **"katello_package_remove"**。

4. 为字符串类型创建一个 "search_query" 字段，并输入与您要删除软件包的主机匹配的搜索查询。
5. 可选：如果要以特定用户身份删除软件包，请使用字符串类型的以下字段创建一个 ssh 对象：
 - 使用 ssh 用户名称的 "effective_user"
 - 如果需要此密码，使用 ssh 用户的密码 "effective_user_password"
6. 可选：如果要稍后删除软件包，请创建 "scheduling" 对象。该对象可以包含 ISO 8601 格式的字符串类型中的一个或两个字段，包括日期、时间和时区：
 - "start_at" - 设置删除软件包的时间
 - "start_before" - 设置删除软件包的最新时间。如果此时无法删除软件包，则会取消这个操作。

如果省略了时间，则默认为 00:00:00。如果省略了时区，则默认为 UTC。
7. 可选：如果要限制同时运行作业的主机数量，请使用整数类型的 "concurrency_level" 字段创建 "concurrency_control" 对象。将主机数量分配为字段值。
8. 可选：如果要稍后删除软件包，并且您希望在运行作业时评估主机搜索查询，请使用 "dynamic_query" 值创建字符串类型的 "targeting_type" 字段。如果您预期搜索查询的结果在运行作业时不同，因为主机状态改变，则这非常有用。如果省略此字段，则默认为 "static_query"。
9. 使用创建的正文发送 POST 请求到 Satellite 服务器的 /api/job_invocations 端点，并使用 python 等工具来查看格式化的响应。

API 请求示例：

```
curl https://satellite.example.com/api/job_invocations \
```

```
-H "content-type: application/json" \  
-X POST \  
-d @Path_To_My_API_Request_Body \  
-u My_Username:My_Password | python3 -m json.tool
```

验证

- 在 **Satellite Web UI** 中，进入到 **Monitor > Jobs** 并查看调度或完成远程执行作业的报告，以删除所选主机上的软件包。

API 请求正文示例

```
{  
  "job_invocation": {  
    "concurrency_control": {  
      "concurrency_level": 100  
    },  
    "feature": "katello_package_remove",  
    "inputs": {  
      "package": "nano vim"  
    },  
    "scheduling": {  
      "start_at": "2023-09-21T19:00:00+00:00",  
      "start_before": "2023-09-23T00:00:00+00:00"  
    },  
    "search_query": "*",  
    "ssh": {  
      "effective_user": "My_Username",  
      "effective_user_password": "My_Password"  
    },  
    "targeting_type": "dynamic_query"  
  }  
}
```

附录 A. 模板编写参考

嵌入式 Ruby (ERB) 是一个基于模板生成文本文件的工具，它将纯文本与 Ruby 代码组合。Red Hat Satellite 在以下情形中使用 ERB 语法：

自定义模板

如需更多信息，请参阅 [置备主机](#) 中的 [创建置备模板](#)。

远程执行任务模板

更多信息请参阅 [第 12 章 配置和设置远程作业](#)。

报告模板

更多信息请参阅 [第 10 章 使用报告模板来监控主机](#)。

分区表的模板

如需更多信息，请参阅 [置备主机](#) 中的 [创建分区表](#)。

智能类参数

如需更多信息，请参阅在 Red Hat Satellite 中使用 [Puppet 集成管理配置](#) 中的 [配置 Puppet 智能类参数](#)。

本节概述了可用于 ERB 模板以及一些用法示例的 Satellite 特定宏和变量。请注意，Red Hat Satellite 提供的默认模板(Hosts > Provisioning templates, Hosts > Job templates, Monitor > Report Templates)还提供很好的 ERB 语法示例源。

在置备主机或运行远程作业时，会执行 ERB 中的代码，变量将被替换为特定于主机的值。这个过程被称为渲染。默认情况下，Satellite 服务器启用了 `safemode rendered` 选项，这样可防止从模板执行任何恶意代码。

A.1. 在 SATELLITE WEB UI 中访问编写参考的模板

您可以在 Satellite Web UI 中访问模板编写参考文档。

流程

1. [登录 Satellite Web UI](#)。

2. 在 **Satellite Web UI** 中, 导航到 **Administer > About**。
3. 单击 **Support** 部分中的 **Templates DSL** 链接。

A.2. 在模板中使用自动完成

您可以使用 **autocompletion** 选项访问模板编辑器中可用宏和使用信息的列表。这适用于 **Satellite** 中的所有模板。

流程

1. 在 **Satellite Web UI** 中, 导航到 **Hosts > Partition table**, **Hosts > Provisioning templates**, 或 **Hosts > Job templates**。
2. 单击模板编辑器右上角的 **设置** 图标, 然后选择 **Autocompletion**。
3. 按模板编辑器中的 **Ctrl + Space** 访问所有可用宏的列表。您可以通过输入有关您要查找的内容的更多信息, 缩小宏列表范围。例如, 如果您查找列出主机内容源 ID 的方法, 您可以键入 **host** 并检查内容源的可用宏列表。
4. 下拉菜单旁边的窗口提供宏、其用法及其返回的值的描述。
5. 当您找到您要查找的方法时, 按 **Enter** 键输入该方法。

您还可以在设置菜单中启用 **实时自动完成** 功能, 以便在您输入内容时查看与模式匹配的宏列表。但是, 这可能会在意外位置输入宏, 如置备模板中的软件包名称。

A.3. 编写 ERB 模板

以下标签是 **ERB** 模板中最重要且常用的标签:

`<% %>`

所有 **Ruby** 代码都包含在 **ERB** 模板中的 `<% %>` 中。代码在渲染模板时执行。它可以包含 **Ruby**

控制流结构，以及特定于 **Satellite** 的宏和变量。例如：

```
<% if @host.operatingsystem.family == "Redhat" && @host.operatingsystem.major.to_i > 6 -%>
systemctl <%= input("action") %> <%= input("service") %>
<% else -%>
service <%= input("service") %> <%= input("action") %>
<% end -%>
```

请注意，此模板会静默地执行带有服务的操作，并在输出中不返回任何操作。

`<%= %>`

这提供的功能与 `<%= %>` 相同，但在执行模板时，代码输出将插入到模板中。这对变量替换很有用，例如：

输入示例：

```
echo <%= @host.name %>
```

渲染示例：

```
host.example.com
```

输入示例：

```
<% server_name = @host.fqdn %>
<%= server_name %>
```

渲染示例：

```
host.example.com
```

请注意，如果您输入了一个不正确的变量，则不会返回任何输出。但是，如果您试图在不正确的变量中调用方法，则会返回以下出错信息：

输入示例：

```
<%= @example_incorrect_variable.fqdn -%>
```

渲染示例：

```
undefined method `fqdn' for nil:NilClass
```

```
<% -%>, <%= -%>
```

默认情况下，如果 Ruby 块在行的末尾关闭，则会在其后插入换行符：

输入示例：

```
<%= "line1" %>  
<%= "line2" %>
```

渲染示例：

```
line1  
line2
```

要更改默认行为，请使用 `-%>` 修改括起的标记：

输入示例：

```
<%= "line1" -%>  
<%= "line2" %>
```

渲染示例：

```
line1line2
```

这用于减少在渲染模板中允许 Ruby 语法允许的行数。ERB 标签中的空格会被忽略。

在报告模板中如何使用此示例来删除 FQDN 和 IP 地址之间的不必要的换行符：

输入示例：

```
<%= @host.fqdn -%>
<%= @host.ip -%>
```

渲染示例：

```
host.example.com10.10.181.216
```

```
<%# %>
```

括起在模板渲染期间被忽略的注释：

输入示例：

```
<%# A comment %>
```

这不会生成输出。

ERB 模板缩进

由于 ERB 标签的不同长度，因此缩进 ERB 语法可能看起来很明显。ERB 语法忽略空格。处理缩进的一种方法是声明每个新行开头的 ERB 标签，然后使用 ERB 标签中的空格来概述语法中的关系，例如：

```
<%- load_hosts.each do |host| -%>
<%- if host.build? %>
<%= host.name %> build is in progress
<%- end %>
<%- end %>
```

A.4. 对 ERB 模板进行故障排除

Satellite Web UI 提供了两种方法来验证特定主机的模板渲染：

- 直接在模板编辑器中 - 编辑模板（在 Hosts > Partition table、Hosts > Provisioning templates, 或 Hosts > Job templates 下）时，在 Template 选项卡中点 Preview 并从列表中选择主机。然后，模板会使用所选主机的参数在文本字段中显示。预览失败有助于识别模板中的问题。



在主机详情页面 - 选择位于 **Hosts > All Hosts** 的主机，点 **Templates** 选项卡列出与主机关联的模板。从所选模板旁边的列表中选择 **Review**，以查看其呈现的版本。

A.5. 通用 SATELLITE 特定 MACROS

本节列出了 ERB 模板的特定于 Satellite 的宏。您可以在所有种类的模板中使用下表中列出的宏。

表 A.1. 通用 Macros

名称	Description
indent(n)	使用未缩进的代码片段模板时，通过 n 缩进代码块很有用。
foreman_url(kind)	将完整 URL 返回到给定类型的主机渲染模板。例如，"provision"类型的模板通常位于 http://HOST/unattended/provision 。
snippet(name)	呈现指定的代码片段模板。对于嵌套自定义模板很有用。
snippets(file)	呈现 Foreman 数据库中找到的指定片段，如果数据库中没有找到，尝试从 无人值守/snippets/ 目录中加载它。
snippet_if_exists(name)	呈现指定的代码片段，如果没有找到指定名称的代码片段，则跳过。

A.6. TEMPLATES MACROS

如果要编写自定义模板，您可以使用以下一些宏。根据模板类型，以下某些宏有不同的要求。

有关报告模板可用宏的更多信息，在 **Satellite Web UI** 中导航至 **Monitor > Report Templates**，然后点 **Create Template**。在 **Create Template** 窗口中，单击 **Help** 选项卡。

如需有关作业模板的可用宏的更多信息，在 **Satellite Web UI** 中，导航到 **Hosts > Job Templates**，然后点 **New Job Template**。在 **New Job Template** 窗口中，单击 **Help** 选项卡。

输入

使用 **输入** 宏，您可以自定义模板可以使用的输入数据。您可以定义用户可用的输入名称、类型和选项。对于报告模板，您只能使用用户输入。当您定义新输入并保存模板时，您可以在模板正文的

ERB 语法中引用输入。

```
<%= input('cpus') %>
```

这会加载用户输入 **cpus** 的值。

load_hosts

使用 **load_hosts** 宏，您可以生成完整的主机列表。

```
<%- load_hosts().each_record do |host| -%>
<%= host.name %>
```

使用 **every_record** 宏的 **load_hosts** 宏来加载 1000 批处理中的记录，以减少内存消耗。

如果要过滤报告的主机列表，您可以添加选项 **搜索**：**input('Example_Host')**：

```
<% load_hosts(search: input('Example_Host')).each_record do |host| -%>
<%= host.name %>
<% end -%>
```

在本例中，您首先会创建一个输入，然后用来优化 **load_hosts** 宏检索的搜索条件。

report_row

使用 **report_row** 宏，您可以创建一个格式化的报告来简化分析。**report_row** 宏需要 **report_render** 宏来生成输出。

输入示例：

```
<%- load_hosts(search: input('Example_Host')).each_record do |host| -%>
<%- report_row(
  'Server FQDN': host.name
) -%>
<%- end -%>
<%= report_render -%>
```

渲染示例：

```
Server FQDN
host1.example.com
host2.example.com
host3.example.com
host4.example.com
host5.example.com
host6.example.com
```

您可以通过添加另一个标头来向报告添加额外的列。以下示例在报告中添加 IP 地址：

输入示例：

```
<%- load_hosts(search: input('host')).each_record do |host| -%>
<%- report_row(
  'Server FQDN': host.name,
  'IP': host.ip
) -%>
<%- end -%>
<%= report_render -%>
```

渲染示例：

```
Server FQDN,IP
host1.example.com,10.8.30.228
host2.example.com,10.8.30.227
host3.example.com,10.8.30.226
host4.example.com,10.8.30.225
host5.example.com,10.8.30.224
host6.example.com,10.8.30.223
```

此宏仅适用于报告模板。

使用 `report_render` 宏，您可以为报告创建输出。在模板渲染过程中，您可以选择您要报告的格式。支持 YAML、JSON、HTML 和 CSV 格式。

```
<%= report_render -%>
```

`render_template()`

此宏仅适用于作业模板。

使用此宏，您可以呈现特定的模板。您还可以启用并定义您要传递给模板的参数。

Very

使用 **事实** 宏，您可以声明是否传递的值为 `true` 或 `false`，无论值是整数还是布尔值还是字符串。

当模板包含多个值类型时，此宏有助于避免混淆。例如，布尔值 `true` 与字符串值 `"true"` 不同。使用这个宏，您可以声明模板如何解释值并避免混淆。

您可以使用 `compare y` 来声明值，如下所示：

```
truthy?("true") => true
truthy?(1) => true
truthy?("false") => false
truthy?(0) => false
```

`falsy`

传真宏的作用与事实宏相同。

使用 `falsy` 宏，您可以声明是否传递的值为 `true` 或 `false`，无论值是整数还是布尔值还是字符串。

您可以使用 `falsy` 来声明值，如下所示：

```
falsy?("true") => false
falsy?(1) => false
falsy?("false") => true
falsy?(0) => true
```

A.7. 特定于主机的变量

以下变量启用在模板中使用主机数据。请注意，作业模板仅接受 `@host` 变量。

表 A.2. 主机特定变量和 Macros

名称	Description
@host.architecture	主机的架构。
@host.bond_interfaces	返回所有绑定接口的数组。请参阅 第 A.10 节 “parsing Arrays” 。
@host.capabilities	系统调配方法，可以是构建（如 kickstart）或镜像。
@host.certname	主机的 SSL 证书名称。
@host.diskLayout	主机的磁盘布局。可以从操作系统继承。
@host.domain	主机的域。
@host.environment 已弃用 使用 host_puppet_environment 变量。	主机的 Puppet 环境。
@host.facts	从 Facter 返回 Ruby 事实哈希。例如，若要访问输出中的 'ipaddress' 事实，请指定 <code>@host.facts['ipaddress']</code> 。
@host.grub_pass	返回主机的引导装载程序密码。
@host.hostgroup	主机的主机组。
host_enc['parameters']	返回包含主机参数信息的 Ruby 哈希。例如，使用 <code>host_enc['parameters']['lifecycle_environment']</code> 获取主机的生命周期环境。
@host.image_build?	如果使用镜像置备主机，则返回 true 。
@host.interfaces	包含包括主接口在内的所有可用主机接口的数组。请参阅 第 A.10 节 “parsing Arrays” 。
@host.interfaces_with_identifier('IDs')	返回带有给定标识符的接口数组。您可以将多个标识符数组作为输入传递，如 <code>@host.interfaces_with_identifier(['eth0', 'eth1'])</code> 。请参阅 第 A.10 节 “parsing Arrays” 。
@host.ip	主机的 IP 地址。

名称	Description
@host.location	主机的位置。
@host.mac	主机的 MAC 地址。
@host.managed_interfaces	返回一组受管接口（不包括 BMC 和绑定接口）。请参阅 第 A.10 节 “parsing Arrays” 。
@host.medium	分配的操作系统安装介质。
@host.name	主机的全名。
@host.operatingsystem.family	操作系统系列。
@host.operatingsystem.major	分配的操作系统的主要版本号。
@host.operatingsystem.minor	分配的操作系统的次要版本号。
@host.operatingsystem.name	分配的操作系统名称。
@host.operatingsystem.boot_files_uri(medium_provider)	内核和 initrd 的完整路径，返回一个数组。
@host.os.medium_uri(@host)	用于置备的 URI（安装介质中配置的路径）。
host_param('parameter_name')	返回指定主机参数的值。
host_param_false?('parameter_name')	如果指定的 host 参数评估为 false ，则返回 false。
host_param_true?('parameter_name')	如果指定的 host 参数评估为 true ，则返回 true。
@host.primary_interface	返回主机的主接口。
@host.provider	计算资源提供程序。
@host.provision_interface	返回主机的调配接口。返回 interface 对象。
@host.ptable	分区表名称。
@host.puppet_ca_server 已弃用 使用 host_puppet_ca_server 变量。	主机必须使用的 Puppet CA 服务器。
@host.puppetmaster 已弃用 使用 host_puppet_server 变量。	主机必须使用的 Puppet 服务器。
@host.pxe_build?	如果使用网络或 PXE 置备主机，则返回 true 。

名称	Description
@host.shortname	主机的短名称。
@host.sp_ip	BMC 接口的 IP 地址。
@host.sp_mac	BMC 接口的 MAC 地址。
@host.sp_name	BMC 接口的名称。
@host.sp_subnet	BMC 网络的子网。
@host.subnet.dhcp	如果为这个主机配置了 DHCP 代理，则返回 true 。
@host.subnet.dns_primary	主机的主 DNS 服务器。
@host.subnet.dns_secondary	主机的辅助 DNS 服务器。
@host.subnet.gateway	主机网关。
@host.subnet.mask	主机的子网掩码。
@host.url_for_boot(:initrd)	与此主机关联的 initrd 镜像的完整路径。不建议，因为它不进行插入变量。
@host.url_for_boot(:kernel)	与此主机关联的内核的完整路径。不建议，因为它不进行插入变量，首选 <code>boot_files_uri</code> 。
@provisioning_type	根据调配类型，等于 'host' 或 'hostgroup'。
@static	如果网络配置是静态的，则返回 true 。
@template_name	被呈现的模板的名称。
grub_pass	返回一个 bootloader 参数，以设置加密的引导装载程序密码，如 <code>--md5pass=114{@host.grub_pass}</code> 。
ks_console	返回使用端口编译的字符串，以及可添加到内核行的主机的 baud 速率。例如 <code>console=ttyS1,9600</code> 。
root_pass	返回为系统配置的根密码。

大多数常见 Ruby 方法都可以应用到特定于主机的变量。例如，要提取主机 IP 地址的最后一部分，您可以使用：

```
<% @host.ip.split('.').last %>
```


A.8. 特定于 KICKSTART 的变量

以下变量设计为在 `kickstart` 调配模板中使用。

表 A.3. Kickstart 特定变量

名称	Description
@arch	主机架构名称，与 @host.architecture.name 相同。
@dynamic	如果使用的分区表是一个 %pre 脚本（如表的第一行），则返回 true 。
@epel	一个命令，它将自动安装 epel-release rpm 的正确版本。在 %post 脚本中使用。
@mediapath	提供 URL 命令的完整 kickstart 行。
@osver	操作系统主要版本号，与 @host. OperatingSystem.major 相同。

A.9. 条件语句

在模板中，您可以根据存在的值来执行不同的操作。要达到此目的，您可以使用 **ERB** 语法中的条件语句。

在以下示例中，**ERB** 语法搜索特定主机名，并根据它找到的值返回输出：

输入示例

```
<% load_hosts().each_record do |host| -%>
<% if @host.name == "host1.example.com" -%>
<%   result="positive" -%>
<% else -%>
<%   result="negative" -%>
<% end -%>
<%= result -%>
```

呈现示例

```
host1.example.com  
positive
```

A.10. PARSING ARRAYS

在编写或修改模板时，您可能会遇到返回数组的变量。例如，与网络接口相关的主机变量，如 `@host.interfaces` 或 `@host.bond_interfaces`，返回在数组中分组的接口数据。要提取特定接口的参数值，请使用 Ruby 方法解析数组。

查找正确的方法以解析数组

以下流程是一个示例，您可以使用它查找相关方法来解析模板中的数组。本例中使用了报告模板，但步骤适用于其他模板。

1. 若要检索内容主机的 NIC，在本例中，使用 `@host.interfaces` 变量返回可用于解析数组的方法。

输入示例：

```
<%= @host.interfaces -%>
```

渲染示例：

```
<Nic::Base::ActiveRecord_Associations_CollectionProxy:0x00007f734036fbe0>
```

2. 在 **Create Template** 窗口中，点 **Help** 选项卡并搜索 `ActiveRecord_Associations_CollectionProxy` 和 `Nic::Base` 类。

3. 对于 `ActiveRecord_Associations_CollectionProxy`，您可以在 `Allowed methods` 或 `members` 列中查看以下方法来解析数组：

```
[] each find_in_batches first map size to_a
```

4. 对于 `Nic::Base`，在 `Allowed method` 或 `members` 列中，您可以查看以下方法来解析数组：

```
alias? attached_devices attached_devices_identifiers attached_to bond_options
children_mac_addresses domain fqdn identifier inheriting_mac ip ip6 link mac managed?
mode mtu nic_delay physical? primary provision shortname subnet subnet6 tag virtual?
vlanid
```

5. 要迭代 `interface` 数组，请将相关方法添加到 `ERB` 语法：

输入示例：

```
<% load_hosts().each_record do |host| -%>
<% host.interfaces.each do |iface| -%>
  iface.alias?: <%= iface.alias? %>
  iface.attached_to: <%= iface.attached_to %>
  iface.bond_options: <%= iface.bond_options %>
  iface.children_mac_addresses: <%= iface.children_mac_addresses %>
  iface.domain: <%= iface.domain %>
  iface.fqdn: <%= iface.fqdn %>
  iface.identifier: <%= iface.identifier %>
  iface.inheriting_mac: <%= iface.inheriting_mac %>
  iface.ip: <%= iface.ip %>
  iface.ip6: <%= iface.ip6 %>
  iface.link: <%= iface.link %>
  iface.mac: <%= iface.mac %>
  iface.managed?: <%= iface.managed? %>
  iface.mode: <%= iface.mode %>
  iface.mtu: <%= iface.mtu %>
  iface.physical?: <%= iface.physical? %>
  iface.primary: <%= iface.primary %>
  iface.provision: <%= iface.provision %>
  iface.shortname: <%= iface.shortname %>
  iface.subnet: <%= iface.subnet %>
  iface.subnet6: <%= iface.subnet6 %>
  iface.tag: <%= iface.tag %>
  iface.virtual?: <%= iface.virtual? %>
  iface.vlanid: <%= iface.vlanid %>
<%- end -%>
```

渲染示例：

```

host1.example.com
iface.alias?: false
iface.attached_to:
iface.bond_options:
iface.children_mac_addresses: []
iface.domain:
iface.fqdn: host1.example.com
iface.identifier: ens192
iface.inheriting_mac: 00:50:56:8d:4c:cf
iface.ip: 10.10.181.13
iface.ip6:
iface.link: true
iface.mac: 00:50:56:8d:4c:cf
iface.managed?: true
iface.mode: balance-rr
iface.mtu:
iface.physical?: true
iface.primary: true
iface.provision: true
iface.shortname: host1.example.com
iface.subnet:
iface.subnet6:
iface.tag:
iface.virtual?: false
iface.vlanid:

```

A.11. 模板片断示例**检查主机是否启用了 Puppet 和 Puppetlabs**

以下示例检查主机是否启用了 **Puppet** 和 **Puppetlabs** 存储库：

```

<%
pm_set = @host.puppetmaster.empty? ? false : true
puppet_enabled = pm_set || host_param_true?('force-puppet')
puppetlabs_enabled = host_param_true?('enable-puppetlabs-repo')
%>

```

捕获主机操作系统的 Major 和 Minor 版本

以下示例演示了如何捕获主机操作系统的次要和主要版本，它们可用于软件包相关的决策：

```
<%
os_major = @host.operatingsystem.major.to_i
os_minor = @host.operatingsystem.minor.to_i
%>

<% if ((os_minor < 2) && (os_major < 14)) -%>
...
<% end -%>
```

将片段导入到模板

以下示例将 `subscription_manager_registration` 片段导入到模板，并将其缩进四个空格：

```
<%= indent 4 do
  snippet 'subscription_manager_registration'
end %>
```

有条件地导入 Kickstart 片段

如果主机子网启用了 DHCP 引导模式，以下示例导入 `kickstart_networking_setup` 片段：

```
<% subnet = @host.subnet %>
<% if subnet.respond_to?(:dhcp_boot_mode?) -%>
<%= snippet 'kickstart_networking_setup' %>
<% end -%>
```

从主机自定义事实解析值

您可以使用 `host.facts` 变量从主机的事实和自定义事实解析值。在本例中，`luks_stat` 是一个自定义事实，您可以使用与 `dmi::system::serial_number` 相同的方式进行解析，这是一个主机事实：

```
'Serial': host.facts['dmi::system::serial_number'],
'Encrypted': host.facts['luks_stat'],
```

在本例中，您可以自定义 **Applicable Errata report** 模板，以解析有关每个主机的内核版本的自定义信息：

```
<%- report_row(
  'Host': host.name,
  'Operating System': host.operatingsystem,
  'Kernel': host.facts['uname::release'],
  'Environment': host.single_lifecycle_environment ? host.single_lifecycle_environment.name :
nil,
```

```
'Erratum': erratum.errata_id,  
'Type': erratum.errata_type,  
'Published': erratum.issued,  
'Applicable since': erratum.created_at,  
'Severity': erratum.severity,  
'Packages': erratum.package_names,  
'CVEs': erratum.cves,  
'Reboot suggested': erratum.reboot_suggested,  
) -%>
```

附录 B. 作业模板示例和扩展

使用本节作为参考，以帮助修改、自定义和扩展您的作业模板以满足您的要求。

B.1. 自定义作业模板

在创建作业模板时，您可以在模板编辑器字段中包含现有模板。这样，您可以组合模板，或者从一般模板创建更具体的模板。

以下模板组合了默认模板，以在客户端上安装和启动 nginx 服务：

```
<%= render_template 'Package Action - SSH Default', :action => 'install', :package => 'nginx' %>
<%= render_template 'Service Action - SSH Default', :action => 'start', :service_name => 'nginx' %>
```

以上模板指定呈现模板的参数值。也可以使用 `input ()` 方法，允许用户在作业执行时为呈现的模板定义输入。例如，您可以使用以下语法：

```
<%= render_template 'Package Action - SSH Default', :action => 'install', :package => input("package") %>
```

使用上述模板，您必须从呈现的模板导入参数定义。为此，请导航到 **Jobs** 选项卡，单击 **Add Foreign Input Set**，然后从 **Target template** 列表中选择呈现的模板。您可以导入所有参数，或者指定以逗号分隔的列表。

B.2. 默认任务模板类别

作业模板类别	Description
软件包	用于执行软件包相关操作的模板。默认情况下包括安装、更新和删除操作。
puppet	在目标主机上运行 Puppet 的模板。
power	用于执行电源相关操作的模板。默认情况下包括重启和关闭操作。
命令	在远程主机上执行自定义命令的模板。
服务	用于执行服务相关操作的模板。默认情况下，包括 start、stop、restart 和 status 操作。

作业模板类别	Description
Katello	用于执行内容相关操作的模板。这些模板主要从 Satellite Web UI 的不同部分使用（例如，用于内容主机的批量操作 UI），但可用于执行勘误表安装等操作。

B.3. RESTORECON 模板示例

本例演示了如何创建名为 **Run Command - restorecon** 的模板，该模板用于恢复目标主机上所选目录中所有文件的默认 SELinux 上下文。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Job templates**。
2. 单击 **New Job Template**。
3. 在 **Name** 字段中输入 **Run Command - restorecon**。选择 **Default** 使模板可供所有组织使用。将以下文本添加到模板编辑器中：

```
restorecon -RvF <%= input("directory") %>
```

在作业调用过程中，`<%= input ("directory")% >` 字符串被替换为用户定义的目录。

4. 在 **Job** 选项卡上，将 **Job category** 设置为 **Commands**。
5. 单击 **Add Input** 以允许自定义作业。在 **Name** 字段中输入 **目录**。输入名称必须与模板编辑器中指定的值匹配。
6. 单击 **Required**，以便在没有用户指定参数的情况下无法执行该命令。
7. 从 **Input type** 列表中选择 **User input**。在作业调用过程中输入要显示的描述，如 **restorecon 的 Target 目录**。

8.

点 **Submit**。如需更多信息，请参阅管理主机中的 [在多个主机上执行 restorecon 模板](#)。

B.4. 渲染 RESTORECON 模板

本例演示了如何创建从 [Example restorecon Template](#) 中创建的 `Run command - restorecon` 模板派生出的模板。此模板不需要用户在作业执行时输入，它将在目标主机上的 `/home/` 目录下的所有文件中恢复 SELinux 上下文。

创建新模板，如 [设置作业模板](#) 中所述，并在模板编辑器中指定以下字符串：

```
<%= render_template("Run Command - restorecon", :directory => "/home") %>
```

B.5. 在多个主机上执行 RESTORECON 模板

本例演示了如何在多个主机上基于 [示例 restorecon](#) 模板中创建的模板运行作业。作业在 `/home/` 目录下的所有文件中恢复 SELinux 上下文。

流程

1. 在 Satellite Web UI 中，导航到 **Monitor > Jobs** 并点 **Run job**。
2. 选择 **Commands as Job category and Run Command - restorecon as Job template**，再单击 **Next**。
3. 选择您要在其上运行作业的主机。如果您没有选择任何主机，该作业将在当前上下文中看到的所有主机上运行。
4. 在 **目录** 字段中，提供一个目录，如 `/home`，然后单击 **Next**。
5. 可选：要为作业配置高级设置，请填写 **高级** 字段。要了解更多有关高级设置的信息，请参阅 [第 12.23 节“作业向导中的高级设置”](#)。当您输入高级设置或不需要时，点 **Next**。
6. 作业调度时间。

要立即执行任务，请保留预先选择的 **Immediate** 执行。

- 要在以后执行任务，请选择 **future execution**。

- 要定期执行任务，请选择 **Recurring execution**。

7. 可选：如果您选择将来或重复执行，请选择 **Query** 类型，否则点 **Next**。

- 静态查询意味着，作业会在您提供的确切主机列表上执行。

- 动态查询意味着仅在执行任务前评估主机列表。如果您根据某些过滤器输入主机列表，则结果可以与首次使用该过滤器时不同。

选择查询类型后，点 **Next**。

8. 可选：如果您选择将来或重复执行，请提供额外的详情：

- 对于未来执行，请输入 **Starts at date and time**。您还可以选择 **Starts before date and time**。如果作业无法在该时间之前启动，它将被取消。

- 对于 **Recurring execution**，请选择结束周期性作业的开始日期和时间、频率和条件。您可以选择从不结束、特定时间结束的重复，或者在指定数量的重复后结束。您还可以添加 **Purpose** - 跟踪作业的特殊标签。一次只能有一个具有给定目的的活动作业。

输入所需信息后，点 **Next**。

9. 查看作业详情。您可以选择返回到作业向导中的任何部分并编辑信息。

10. 单击 **Submit** 以计划执行的作业。

B.6. 在模板中包括 POWER 操作

本例演示了如何设置作业模板来执行电源操作，如 `reboot`。此步骤可防止卫星将重新引导时的断开连接异常解释为错误，因此远程执行作业可以正常工作。

创建新模板，如 [设置作业模板](#) 中所述，并在模板编辑器中指定以下字符串：

```
<%= render_template("Power Action - SSH Default", :action => "restart") %>
```

附录 C. HOST COLUMNS 概述

以下是主机表中可以划分为内容类别的列的完整概述。有些列属于多个类别。有关如何自定义主机表中的列的更多信息，请参阅第 2.22 节“选择 Host Columns”。

General

- **power** - 主机是否开启或关闭（如果可用）
- **name** - 主机的名称
- **操作系统** - 主机的操作系统
- **Model** - 主机硬件模型（如果虚拟主机，或计算资源）
- **所有者** - 拥有主机的用户或组
- **主机组** - 主机的主机组
- **最后报告** - 最后一个主机报告的时间
- **注释** - 提供给主机的注释

内容

- **name** - 主机的名称
- **操作系统** - 主机的操作系统
- **订阅状态** - 主机附加了有效的订阅

- **可安装更新** - 可安装更新的数量分为四个类别：**security, bugfix, enhancement, Total**

- **生命周期阶段** - 主机的生命周期阶段

- **内容视图** - 主机的内容视图

- **registered** - 主机注册到 Satellite 时的时间

- **最后检查** - 主机和 Satellite 服务器之间的通信最后一次时间

Network

- **IPv4** - 主机的 IPv4 地址

- **IPv6** - 主机的 IPv6 地址

- **MAC** - 主机的 MAC 地址

报告的数据

- **socket** - 主机插槽数

- **cores** - 主机处理器内核数

- **RAM** - 内存量

- **引导时间** - 主机最后一次引导时间

- **virtual** - 主机是否识别为虚拟机

- **磁盘空间总量 - 主机存储空间总量**
- **内核版本 - 主机操作系统的内核版本**
- **BIOS 供应商 - 主机 BIOS 的厂商**
- **BIOS 发行日期 - 主机 BIOS 的发行日期**
- **BIOS 版本 - 主机 BIOS 的版本**

puppet (仅在安装了 Puppet 插件时)

- **Environment name - 主机的 Puppet 环境的名称**

RH Cloud

- **建议 - 主机的可用建议数**