



Red Hat Satellite 6.12

管理主机

管理红帽卫星 6 环境中主机的指南。

Red Hat Satellite 6.12 管理主机

管理红帽卫星 6 环境中主机的指南。

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 workflow 前，您必须已成功安装 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 卫星中的主机概述	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. 从虚拟机监控程序将虚拟机与 SATELLITE 相关联	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	11
2.8. 为每个生命周期环境创建主机组	12
2.9. 将主机添加到主机组	13
2.10. 更改主机的主机组	13
2.11. 在主机集合中添加主机	14
2.12. 更改主机的内容源	14
2.13. 更改主机的环境	15
2.14. 更改主机的受管状态	15
2.15. 在主机上启用 TRACER	15
2.16. 在主机上重启应用程序	16
2.17. 将主机分配给特定机构	16
2.18. 将主机分配给特定位置	17
2.19. 在主机间切换	17
2.20. 从 SATELLITE 中删除主机	17
第 3 章 注册主机和设置主机集成	19
3.1. 注册中支持的客户端	19
3.2. 注册方法	19
3.3. 使用全局注册注册主机	20
3.4. 使用启动脚本注册主机	23
3.5. 安装 KATELLO AGENT	32
3.6. 安装 TRACER	32
3.7. 在主机注册过程中安装和配置 PUPPET 代理	33
3.8. 手动安装和配置 PUPPET 代理	34
第 4 章 将主机从 KATELLO AGENT 迁移到远程执行	36
第 5 章 添加网络接口	39
5.1. 添加物理接口	39
5.2. 添加虚拟接口	40
5.3. 添加绑定接口	41
5.4. SATELLITE 中可用的绑定模式	42
5.5. 添加基板管理控制器(BMC)接口	42
第 6 章 将主机升级到 NEXT MAJOR RED HAT ENTERPRISE LINUX 发行版本	44
第 7 章 将主机转换为 RED HAT ENTERPRISE LINUX	46
7.1. 用于转换的 ANSIBLE 变量	47
第 8 章 使用 RHEL WEB 控制台主机管理和监控	49
8.1. 在 SATELLITE 中启用 RHEL WEB 控制台	49
8.2. 使用 RHEL WEB 控制台管理和监控主机	49

8.3. 在 SATELLITE 上禁用 RHEL WEB 控制台	50
第 9 章 使用 RED HAT INSIGHTS 监控主机	51
9.1. 在 SATELLITE 中将 RED HAT INSIGHTS 与主机搭配使用	51
9.2. 为主机创建 INSIGHTS 计划	52
第 10 章 使用报告模板监控主机	54
10.1. 生成主机监控报告	54
10.2. 创建报告模板	55
10.3. 导出报告模板	56
10.4. 使用 SATELLITE API 导出报告模板	56
10.5. 导入报告模板	58
10.6. 使用 SATELLITE API 导入报告模板	58
10.7. 创建报告模板以监控权利	59
10.8. 报告 TEMPLATE SAFE MODE	61
第 11 章 配置主机集合	62
11.1. 创建主机集合	62
11.2. 克隆主机集合	62
11.3. 删除主机集合	62
11.4. 批量为主机集合添加主机	63
11.5. 从主机集合中删除主机	63
11.6. 在主机集合中添加内容	64
第 12 章 配置和设置远程作业	69
12.1. 关于在主机上运行作业	69
12.2. 远程执行工作流	69
12.3. 远程执行的权限	70
12.4. 远程执行的传输模式	71
12.5. 配置主机以使用 PULL 客户端	71
12.6. 创建作业模板	72
12.7. 根据名称导入 ANSIBLE PLAYBOOK	73
12.8. 导入所有可用 ANSIBLE PLAYBOOK	74
12.9. 在 SATELLITE 中将 FALLBACK 配置为 ANY CAPSULE REMOTE EXECUTION SETTING	74
12.10. 在 SATELLITE 中配置全局胶囊远程执行设置	75
12.11. 在 PUSH 模式中为远程执行作业设置替代目录	76
12.12. 在 PULL 模式中为远程执行作业设置替代目录	76
12.13. 更改权限提升方法	77
12.14. 为远程执行分发 SSH 密钥	77
12.15. 手动将 SSH 密钥分发到远程执行	77
12.16. 将密码添加到用于远程执行的 SSH 密钥	78
12.17. 使用 SATELLITE API OBTAIN SSH KEYS FOR REMOTE EXECUTION	78
12.18. 在置备过程中将 KICKSTART 模板配置为分布式 SSH 密钥	78
12.19. 为 KERBEROS TICKET GRANTING TICKETS 配置 KEYTAB	79
12.20. 为远程执行配置 KERBEROS 身份验证	79
12.21. 设置作业模板	80
12.22. 执行远程作业	80
12.23. 为主机调度周期性 ANSIBLE 作业	82
12.24. 为主机组调度周期性 ANSIBLE 作业	82
12.25. 监控作业	82
第 13 章 SATELLITE 中的主机状态	84
13.1. 主机全局状态概述	84
13.2. 主机子状态概述	84

第 14 章 同步模板存储库	91
14.1. 启用 TEMPLATESYNC 插件	91
14.2. 配置 TEMPLATESYNC 插件	91
14.3. 使用存储库源	92
14.4. 导入和导出模板	94
14.5. 卸载 FOREMAN 模板插件	96
第 15 章 管理软件包	98
15.1. 在主机上安装软件包	98
15.2. 升级主机上的软件包	99
15.3. 从主机中删除软件包	101
附录 A. 模板编写参考	104
A.1. 在卫星 WEB UI 中访问模板编写参考	104
A.2. 在模板中使用自动完成功能	104
A.3. 编写 ERB 模板	105
A.4. ERB 模板故障排除	107
A.5. 通用的针对与 SATELLITE 的宏	107
A.6. TEMPLATES MACROS	107
A.7. 特定于主机的变量	110
A.8. 特定于 KICKSTART 的变量	112
A.9. 条件语句	113
A.10. PARSING ARRAYS	113
A.11. TEMPLATE SNIPPETS 示例	115
附录 B. 作业模板示例和扩展	117
B.1. 自定义模板	117
B.2. 默认任务模板类别	117
B.3. RESTORECON 模板示例	117
B.4. 渲染 RESTORECON 模板	118
B.5. 在多个主机上执行 RESTORECON 模板	118
B.6. 模板中包括 POWER ACTIONS	118

向红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

您可以通过在 Bugzilla 中记录一个 ticket 来提交反馈：

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

第 1 章 卫星中的主机概述

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。虚拟主机可以部署在 Red Hat Satellite 支持的任意平台上，如 Amazon EC2、Google Compute Engine、KVM、libvirt、Microsoft Azure、OpenStack、Red Hat Virtualization、Rackspace Cloud Services 或 VMware vSphere。

红帽卫星可大规模实现主机管理，包括监控、配置、远程执行、配置管理、软件管理和订阅管理。您可以从 Satellite Web UI 或命令行管理主机。

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

- **所有主机** - 卫星服务器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。
- **内容主机** - 管理与内容和订阅相关的任务的主机列表。
- **主机集合** - 用于批量操作（如勘误表安装）的用户定义的主机列表。

要搜索主机，请在 **搜索** 字段中键入，并使用星号(*)来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，请单击 **Content Hosts** 页面并在 **Search** 字段中输入 **dev-node***。或者，***node*** 还会找到内容主机 **dev-node.example.com**。



警告

即使没有自助注册，卫星服务器也将列为主机本身。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

使用这个步骤在 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. 点 **Ansible Roles** 选项卡，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 **接口** 标签页中：
 - a. 对于每个接口，点 **Actions** 列中的 **Edit**，并根据需要配置以下设置：
 - **输入** InventoryService-jaxbFor a Bond 或 BMC 接口，使用 **Type** 列表并选择接口类型。
 - **MAC 地址** 时间为 the MAC 地址。
 - **DNS 名称** the DNS name the DNS name is known to the DNS server.这用于 FQDN 的主机部分。
 - **Domain** – 选择置备网络的域名。这会使用特定的适当的子网自动更新 **Subnet** 列表。
 - **IPv4 Subnet** abrt-openmpi 从列表中为主机选择 IPv4 子网。
 - **IPv6 Subnet** abrt-abrt 从列表中为主机选择 IPv6 子网。
 - **IPv4 address** – 如果 IP address management (IPAM) 已为子网启用，IP 地址会被自动推荐。或者，您可以输入一个地址。如果启用了置备令牌，如果子网没有管理反向 DNS，或者子网没有管理 DHCP 保留，则地址可以被省略。
 - **IPv6 address** – 如果 IP address management (IPAM) 已为子网启用，IP 地址会被自动推荐。或者，您可以输入一个地址。
 - **受管** 时间是选择此复选框，以在置备期间配置接口以使用提供的 DHCP 和 DNS 服务。
 - **主** InventoryService-latexSelect this 复选框，以使用此界面中的 DNS 名称作为 FQDN 的主机部分。
 - **provision** abrt-abrtSelect this 复选框，以将此接口用于置备。这意味着，TFTP 引导将使用此接口进行，或者在基于映像的调配时进行，整个置备将通过此界面执行。请注意，很多置备任务（如通过 **anaconda** 下载 RPM（在 **%post** 脚本中的 Puppet 设置）将使用主接口。
 - 如果这个接口不是物理设备，则 **虚拟 NIC** 会显示这个复选框。此设置有两个选项：
 - **标签** InventoryService-latexly 设置 VLAN 标签。如果未设置，则标签将是子网的 VLAN ID。

- o 将这个虚拟接口 **附加到** 的接口的设备名称已附加。
- b. 点 **OK** 保存接口配置。
 - c. (可选) 点击 **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. 点 **Submit** 以应用更改并退出。
6. 在 **Operating System** 选项卡中，输入所需详情。对于 Red Hat 操作系统，选择 **Synced Content for Media Selection**。如果要使用非红帽操作系统，请选择 **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。您无法同时指定两者。
 7. 在 **Parameters** 选项卡中，点 **Add Parameter** 添加您要在运行时传递给作业模板的任何参数变量。这包括您要与主机关联的所有 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 **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 属性定义要在主机创建时自动附加哪些订阅。在 **Host Parameters** 区域，输入以下参数名以及对应的值。如需值列表，请参阅 [执行标准 RHEL 8 安装中的系统目的介绍](#)。

 - **syspurpose_role**
 - **syspurpose_sla**
 - **syspurpose_usage**
 - **syspurpose_addons**

如果要为远程作业执行创建带有 pull 模式的主机，请添加 **enable-remote-execution-pull** 参数，并将 type 布尔值 设为 **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，输入有关主机的附加信息。
 9. 单击 **Submit** 以完成您的调配请求。

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
```

这个命令会提示您输入 root 密码。需要指定主机的 IP 和 MAC 地址。主网络接口的其他属性可以从主机组继承，或者使用 **--subnet** 和 **--domain** 参数进行设置。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

您可以克隆现有的主机。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 在 **Actions** 菜单中，点 **Clone**。
3. 在 **Host** 选项卡中，确保提供与原始主机不同的**名称**。
4. 在 **Interfaces** 标签页上，确保 提供不同的 IP 地址。
5. 单击 **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. 从虚拟机监控程序将虚拟机与 SATELLITE 相关联

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Compute Resources**。
2. 选择一个计算资源。
3. 在 **Virtual Machines** 选项卡中，从 **Actions** 菜单中选择 **Associate VM**。

2.4. 编辑主机的系统目的

您可以为 Red Hat Enterprise Linux 主机编辑系统目的的属性。系统目的允许您设置网络上的预期使用系统，并改进 Red Hat Hybrid Cloud Console Subscriptions 服务中的报告准确性。如需与系统目的相关的信息，请参阅[执行标准 RHEL 8 安装中的系统目的介绍](#)。

先决条件

- 要编辑的主机必须使用 subscription-manager 注册。

流程

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

CLI 过程

1. 登录主机，再编辑所需的系统目的属性。例如，将使用类型设置为 **Production**，将角色设置为 **Red Hat Enterprise Linux Server**，并添加 **addon** 附加组件。如需值列表，请参阅 [执行标准 RHEL 8 安装中的系统目的介绍](#)。

```
# 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。System purpose 属性定义要自动附加到主机的订阅。如需与系统目的相关的信息，请参阅 [执行标准 RHEL 8 安装中的系统目的介绍](#)。

先决条件

- 要编辑的主机必须使用 subscription-manager 注册。

流程

1. 在 Satellite web UI 中，进入 **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。
 - **不要修改**，以保留在所选主机上设置的属性。
 - **none(Clear)**，清除所选主机上的属性。
4. 点 **Assign**。
5. 在 Satellite web UI 中，进入 **Hosts > Content Hosts** 并选择相同的 Red Hat Enterprise Linux 8 主机，以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. 点 **Auto-Attach** 以根据其系统角色自动将订阅附加到所有选择的主机。

2.6. 更改主机的模块流

如果您有一个主机正在运行 Red Hat Enterprise Linux 8，您可以修改您安装的软件仓库的模块流。

您可以在卫星 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. 点击要修改的主机的名称。
3. 单击 **内容选项卡**，然后单击 **模块流** 选项卡。
4. 单击模块旁边的垂直省略，然后选择您要执行的操作。远程执行作业完成后，您将获得一个 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共同的设置和属性。为每个新主机添加这些设置和属性非常耗时。如果使用主机组，您可以将常见属性应用到您创建的主机。

主机组作为常见主机设置的模板，包含您提供给主机的许多相同详细信息。使用主机组创建主机时，主机将从主机组中继承定义的设置。然后您可以为主机提供额外的详情。

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。目的是具有一个代表机构中所有主机的基本级别主机组，并提供常规设置，然后嵌套组以提供特定的设置。例如，您可以有一个用来定义操作系统的基本级别主机组，以及继承基本级别主机组的两个嵌套式主机组：

- **Hostgroup: Base** (Red Hat Enterprise Linux 7.6)
 - **Hostgroup: Webserver** (应用 **nginx** Puppet 类)
 - **Host: webserver1.example.com** (web server)
 - **主机：webserver2.example.com** (Web 服务器)
 - **hostgroup：存储** (应用 **nfs** Puppet 类)
 - **Host: storage1.example.com** (storage server)
 - **Host: storage2.example.com** (storage server)
 - **host: 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，因为它们都具有 **Base** 主机组的继承。两个 Web 服务器主机从 **Webserver** 主机组继承设置，其中包括 **nginx** Puppet 类以及来自 **Base** 主机组中的设置。两个存储服务器从 **Storage** 主机组继承设置，其中包含 **nfs** Puppet 类和 **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，进入 **Configure > Host Groups**，然后单击 **Create Host Group**。

2. 如果要从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 **Name**。
4. 输入您希望将来主机继承的任何更多信息。
5. 点 **Ansible Roles** 选项卡，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. 点附加标签页，并将您要属性的任何详情添加到主机组。



注意

在注册与生产环境中创建的 Puppet 环境关联的主机组时，Puppet 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** 以保存该主机组。

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式主机组。

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

您可以在卫星 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. 点击要修改的主机的名称。
3. 点 **编辑按钮**。
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

使用此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，则会应用主机从主机组继承的全新的值。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 点击要修改的主机的名称。
3. 点 **编辑按钮**。
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. 在主机集合中添加主机

您可以在卫星 Web UI 中将主机添加到主机集合。

先决条件

主机必须注册到 Red Hat Satellite，才能将其添加到主机集合中。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，卫星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. 点击要修改的主机的名称。
3. 在 **主机集合** 卡中，点垂直的 ellipsis 并选择 **Add host to collections**。
4. 选择主机集合。
5. 点 **Add**。

CLI 过程

- 要在主机集合中添加主机，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. 更改主机的内容源

内容源是一个主机消耗内容的胶囊。使用此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. 点击要修改的主机的名称。
3. 点 **Edit** 按钮旁边的垂直省略图标，然后选择 **Change content source**。
4. 从列表中选择 Environment、Content View 和 Content Source。
5. 点 **更改内容源**。

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 invocation**。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。

2.13. 更改主机的环境

使用此流程更改主机的环境。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
2. 点击要修改的主机的名称。
3. 点 **Content view details** 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. 点击 **Save**。

2.14. 更改主机的受管状态

Satellite 调配的主机默认为 Managed。当主机设置为 Managed 时，您可以从 Satellite 服务器配置额外的主机参数。这些附加参数在 **Operating System** 标签页中列出。如果您更改了 **操作系统** 标签上的任何设置，在将主机设置为构建并重启前，它们才会生效。

如果您需要使用 Satellite 不支持的操作系统获取系统管理的报告，请将主机设置为 Unmanaged。

流程

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

2.15. 在主机上启用 TRACER

使用这个步骤在 Satellite 上启用 Tracer，并访问 Trace。tracer 显示需要重启的服务和应用程序列表。trace 是卫星 Web UI 中 Tracer 生成的输出。

先决条件

- Satellite Client 6 存储库被同步
- 卫星客户端 6 存储库在内容视图和主机的生命周期环境中提供
- 为主机启用了 Satellite Client 6 软件仓库

- 启用远程执行

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 点击要修改的主机的名称。
3. 点 **Traces** 选项卡，然后点 **Enable Traces** 按钮。
4. 从列表中选择要安装 **katello-host-tools-tracer** 的供应商。
5. 点 **Enable Tracer** 按钮。远程执行作业完成后，您将获得一个 REX 作业通知。

2.16. 在主机上重启应用程序

使用这个步骤从 Satellite Web UI 重启应用程序。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 点击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，您将得到 REX 作业通知。

2.17. 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。有关机构以及如何配置它们的常规信息，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. 选择您要更改的主机的复选框。
3. 在 **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，选择您要为其分配主机的机构。选中 **Fix Organization on Mismatch** 的复选框。



注意

如果有一个与主机关联的资源（如域或子网），且未与您要为其分配主机的组织相关联，则不匹配。选项 **Fix Organization on Mismatch** 会将这样的资源添加到机构中，因此建议选择。选项 **Fail on Mismatch** 将始终出现错误消息。例如，将主机从一个机构重新分配给另一个机构会失败，即使设置中没有实际不匹配。

5. 点 **Submit**。

2.18. 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 [管理内容中的创建一个位置](#)。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
2. 选择您要更改的主机的复选框。
3. 在 **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，再选择您要用于您的主机的位置。选中 **修复位置上的"修复位置"**。



注意

如果有一个与主机关联的资源（如域或子网），且与您要为其分配主机的位置没有关联，则不匹配。选项 **Fix Location on Mismatch** 会将这样的资源添加到位置，因此建议选择。选项 **Fail on Mismatch** 将始终出现错误消息。例如，将主机从一个位置重新分配给另一个位置将会失败，即使设置中没有实际不匹配。

5. 点 **Submit**。

2.19. 在主机间切换

当您在卫星 Web UI 中的特定主机上时，您可以使用主机切换器在主机间导航，而无需离开页面。点 **主机名** 旁的 **WWN**。这将以字母顺序显示主机列表，包含分页箭头和搜索栏来查找您要查找的主机。

2.20. 从 SATELLITE 中删除主机

使用这个流程从 Red Hat Satellite 中删除主机。

流程

1. 在 Satellite web UI 中，导航到 **Hosts > All hosts** or **Hosts > Content Hosts**。请注意，您从 **所有主机** 或 **内容主机** 中删除主机的页面没有区别。在这两种情况下，Satellite 都彻底删除主机。
2. 选择您要删除的主机。
3. 在 **Select Action** 列表中，选择 **Delete Hosts**。
4. 单击 **Submit** 以从 Red Hat Satellite 永久删除主机。



警告

默认情况下，主机删除设置上的 **Destroy associated VM** 被设置为 **no**。如果删除了与虚拟机关联的主机记录，虚拟机将保留在计算资源中。

要删除计算资源上的虚拟机，请进入到 **Administer > Settings** 并选择 **Provisioning** 选项卡。如果删除了与虚拟机关联的主机记录，在 **host delete** 上将 **Destroy associated VM** 设置为 **yes** 会删除该虚拟机。为避免删除此情形中的虚拟机，在不将其从计算资源中移除或更改设置的情况下，将虚拟机与 Satellite 解除关联。

2.20.1. 从 Satellite 取消关联虚拟机，而不将其从 Hypervisor 中删除

流程

1. 在卫星 Web UI 中，导航到 **Hosts > All Hosts**，再选择要解除主机左侧的复选框。
2. 在 **Select Action** 列表中，选择 **Disassociate Hosts** 按钮。
3. 可选：选择保存主机作为将来的操作的复选框。
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 Agent”](#)
- [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

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。

先决条件

- 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。
- 卫星服务器、任何胶囊服务器和所有主机都必须与同一 NTP 服务器同步，并且使时间同步工具已启用并在运行。
- 主机必须有可用的激活码。如需更多信息，请参阅 [管理内容](#) 中的 [管理激活密钥](#)。
- 可选：如果要将主机注册到 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 [为主机注册和置备配置 Capsule](#)。
- 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** 列表中选择要关联主机的主机。从 **主机组** 继承值的字段：**操作系统**、**激活密钥** 和 **生命周期环境**。
5. 可选：**在操作系统** 列表中，选择您要注册的主机的操作系统。
6. 可选：在 **Capsule** 列表中，选择用于注册主机的胶囊。
7. 可选：如果要使第一个调用不安全，请选择 **Insecure** 选项。在第一个调用过程中，主机从 Satellite 下载 CA 文件。主机将使用此 CA 文件连接到卫星，以备将来调用它们的安全。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，则攻击者可以从第一个不安全的调用中获取 CA 文件，攻击者可以从注册的主机和 JSON Web Tokens(JWT)访问 API 调用的内容。因此，如果您选择在注册过程中部署 SSH 密钥，攻击者将能够使用 SSH 密钥访问主机。

反之，您可以在注册主机前手动在每台主机上复制并安装 CA 文件。

要做到这一点，找到 Satellite 存储 CA 文件的位置。导航到 **Administer > Settings > Authentication** 并找到 **SSL CA 文件** 设置的值。

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 ...
```

以下是使用 **--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. 在 **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**，则在注册的主机上将安装公共 SSH 密钥。继承的值基于 **host_registration_remote_execution** 参数。它可以被继承，例如，来自主机组、操作系统或组织。覆盖后，所选值将存储在主机参数级别上。

10. 在 **Setup Insights** 列表中，选择是否安装 **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，列出您要在注册时要在主机上安装的软件包（使用空格分隔）。这可以通过 **host_packages** 参数设置。
 12. 可选：选择 **Update packages** 选项在注册时更新主机中的所有软件包。这可以通过 **host_update_packages** 参数设置。
 13. 可选：在 **Repository** 字段中输入执行注册前要添加的存储库。例如，使 **subscription-manager** 软件包在注册时很有用。对于 Red Hat 系列发行版本，请输入该程序库的 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。需要使用 GPG 公钥标头以 ASCII 形式指定它。
 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生成的 **curl** 命令的工作方式。您可以将持续时间设置为 0-999 999 小时，或没有限制。
请注意，Satellite 应用生成 **curl** 命令对主机的授权的用户权限。如果用户丢失或获得额外的权限，则 JWT 的权限也会发生。因此，在令牌期间，不要删除、块或更改用户的权限。

JWT 的范围仅限于注册端点，在其他任何位置都无法使用。
 16. 可选：在 **Remote Execution Interface** 字段中输入主机必须用于 SSH 连接的网络接口的标识符。如果将此字段留空，Satellite 将使用默认网络接口。
 17. 从 **REX 拉取模式** 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**，则将在注册的主机上安装远程执行 pull 客户端。继承的值基于 **host_registration_remote_execution_pull** 参数。它可以被继承，例如，来自主机组、操作系统或组织。当被覆盖时，所选值存储在主机参数级别上。

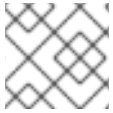
注册的主机必须有权访问红帽卫星客户端 6 存储库。

有关 pull 模式的更多信息，请参阅 [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 **Lifecycle** 环境。
 20. 可选：如果要忽略订阅管理器错误，请选择 **Ignore** 错误选项。
 21. 可选：如果要在注册前删除任何 **katello-ca-consumer** rpms，选择 **Force** 选项，并运行带有 **--force** 参数的 **subscription-manager**。
 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，卫星中的所有默认模板都锁定。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** 包含主机注册后初始配置的步骤。

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 脚本来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，或者将现有主机从 RHN、SAM、rhm 或其他 Red Hat Satellite 实例迁移。

在卫星服务器的基本操作系统中默认安装 **katello-client-bootstrap** 软件包。**bootstrap.py** 脚本安装在 **/var/www/html/pub/** 目录中，使它可用于 **satellite.example.com/pub/bootstrap.py** 上的主机。该脚本包含 **/usr/share/doc/katello-client-bootstrap-version/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 once，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**。

先决条件

- 您有 Satellite 用户，具有运行 bootstrap 脚本所需的权限。此流程中的示例指定 **admin** 用户。如果您的安全策略不可接受，请创建一个具有最少权限的新角色，并将其添加到运行脚本的用户中。更多信息请参阅 [第 3.4.1 节“为 Bootstrap 脚本设置权限”](#)。
- 您的主机有一个激活码，并启用了 Satellite Client 6 存储库。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。有关创建主机组的更多信息，请参阅 [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**，再点击 **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，然后单击 **更新**。

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 help 文本确认脚本可以执行：

- On Red Hat Enterprise Linux 8:

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

- 在其他 Red Hat Enterprise Linux 版本中：

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

5. 输入适合您环境的 bootstrap 命令。

对于 **--server** 选项，指定卫星服务器的 FQDN 或胶囊服务器。对于 **--location**、**--organization** 和 **--hostgroup** 选项，请使用带引号名称（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. 输入您使用 **--login** 选项指定的 Satellite 用户的密码。
该脚本会向 **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 Server。
8. 从 **Actions** 列的列表中，选择 **Certificates**。
9. 在 **Actions** 列中，单击 **Sign** 以批准主机的 Puppet 证书。
10. 返回主机以查看 bootstrap 过程的剩余部分。
11. 在 Satellite Web UI 中，导航到 **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 Bootstrap 脚本设置权限

使用这个流程为 Satellite 用户配置运行 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，导航到 **Administer > Users**。
2. 点所需的**用户名**来选择现有用户。此时会打开一个新窗格，其中包含修改所选用户的信息的选项卡。或者，创建一个新用户，专门用于运行此脚本。
3. 点 **Roles** 选项卡。
4. 从 **Roles** 列表中选择 **Edit hosts and Viewer**。



重要

Edit hosts 角色允许用户编辑和删除主机，并可添加主机。如果您的安全策略不接受它，请创建带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 脚本所需的权限的角色。这个示例创建了一个名为 *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

本节还有更多示例，它使用 bootstrap 脚本来注册或迁移主机。



警告

这些示例指定 **admin** Satellite 用户。如果您的安全策略无法接受此项，则创建一个具有 bootstrap 脚本所需最少权限的新角色。更多信息请参阅 [第 3.4.1 节“为 Bootstrap 脚本设置权限”](#)。

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

使用脚本及 **--force** 中移除旧 Satellite 中的 **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 Red Hat Network(RHN)或 Satellite 5 迁移到 Satellite

bootstrap 脚本检测到存在 **/etc/syconfig/rhn/systemid**，以及到 RHN 的有效连接作为系统注册到旧平台的指示。然后该脚本会调用 **rhn-classic-migrate-to-rhsm** 将系统从 RHN 迁移。默认情况下，因为审计原因，该脚本不会删除系统的旧配置集。要删除旧的配置集，请使用 **--legacy-purge**，并使用 **--legacy-login** 提供具有删除配置集的适当权限的用户帐户。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. 在不使用 Puppet 的情况下将主机注册到 Satellite

默认情况下，bootstrap 脚本配置用于内容管理和配置管理的主机。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 中以进行内容管理

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 \  
--skip-foreman
```



```
--server satellite.example.com \  
--organization="My_Organization" \  
--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. 更改引导脚本使用的方法下载使用者 RPM

默认情况下，bootstrap 脚本使用 HTTP 从 <http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm> 下载使用者 RPM。在某些环境中，您可能只想在主机和 Satellite 间进行 HTTPS。使用 **--download-method** 将下载方法从 HTTP 更改为 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

在一个接口上使用多个接口或多个 IP 地址的主机上，您可能需要覆盖 IP 地址的自动探测，并为 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 \  
--ip
```

```
--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 \  
--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. 在注册过程中为主机创建域

要创建主机记录，在运行脚本之前，主机的 DNS 域需要存在于 Satellite 中。如果域不存在，则使用 `--add-domain` 添加它。

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. 为主机提供备用 FQDN

如果主机名不是 FQDN，或者与 RFC 不兼容（包含下划线等字符），脚本会在主机名验证阶段失败。如果无法将主机更新为使用 Satellite 接受的 FQDN，您可以使用 bootstrap 脚本指定备选 FQDN。

流程

1. 使用 Hammer 将 **create_new_host_when_facts_are_uploaded** 和 **create_new_host_when_report_is_uploaded** 设为 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** 指定将报告给 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 AGENT

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 **gofer** 软件包，它提供了 **goferd** 服务。

先决条件

- 您已在 Satellite 服务器上启用了 Satellite Client 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 Client 6 仓库](#)。
- 您已在卫星服务器上同步了卫星客户端 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 Client 6 仓库](#)。
- 您已在客户端上启用了 Satellite Client 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

使用这个步骤在 Red Hat Satellite 上安装 Tracer 并访问 Trace。tracer 显示已过时且需要重启的服务和应用程序列表。trace 是卫星 Web UI 中 Tracer 生成的输出。

先决条件

- 主机必须注册到红帽卫星。
- 必须在卫星服务器上启用和同步红帽卫星客户端 6 存储库，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，导航到 **Hosts > All hosts**，然后单击所需的主机名。
4. 点 **Traces** 选项卡来查看 Traces。如果没有安装，启用 **Traces** 按钮会启动安装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](#)。

先决条件

- 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 [管理激活密钥](#)。

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，您可以进入到 **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 **boolean** 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](#) 被解决前，您需要使用 **host** 参数来指定 Puppet 代理配置，即使 Puppet 服务器是 Capsule 服务器。

4. 进入到 **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. 进入到 **Infrastructure > Capsules**。
6. 从所需胶囊服务器的 **操作** 列中的列表中，选择 **证书**。
7. 单击所需主机右侧的 **Sign to**，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](#)。

先决条件

- 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 Client 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 及更早版本的主机上：


```
# 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 **操作** 列中的列表中，选择 **证书**。
9. 单击所需主机右侧的 **Sign to**，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*

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软件包内容的首选方法。Katello Agent 已弃用，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。

先决条件

- 您已在 Satellite 服务器上启用了 Satellite Client 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 Client 6 仓库](#)。
- 您已在卫星服务器上同步了卫星客户端 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 Client 6 仓库](#)。
- 您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“为远程执行分发 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 或较低版本中，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 **Use remote execution** 设置为 **Yes**。

卫星服务器现在通过远程执行而非 Katello Agent 使用主机管理。

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

表 4.1. Hammer 命令

操作	Katello Agent	远程执行
安装软件包	Hammer 主机软件包安装	<code>hammer job-invocation create --feature katello_package_install</code>
安装软件包 (主机集合)	Hammer host-collection 软件包安装	<code>hammer job-invocation create --feature katello_package_install</code>
删除软件包	Hammer 主机软件包删除	<code>hammer job-invocation create --feature katello_package_remove</code>
删除软件包 (主机集合)	<code>hammer host-collection package remove</code>	<code>hammer job-invocation create --feature katello_package_remove</code>
更新软件包	Hammer 主机软件包升级	<code>hammer job-invocation create --feature katello_package_update</code>
更新软件包 (主机集合)	Hammer host-collection 软件包更新	<code>hammer job-invocation create --feature katello_package_update</code>
更新所有软件包	Hammer 主机软件包更新	<code>hammer job-invocation create --feature katello_package_update</code>
安装勘误	应用 Hammer 主机勘误表	<code>hammer job-invocation create --feature katello_errata_install</code>
安装勘误 (主机集合)	Hammer host-collection 勘误表安装	<code>hammer job-invocation create --feature katello_errata_install</code>
安装软件包组	Hammer host package-group 安装	<code>hammer job-invocation create --feature katello_group_install</code>
安装软件包组 (主机集合)	<code>hammer host-collection package-group install</code>	<code>hammer job-invocation create --feature katello_group_install</code>

操作	Katello Agent	远程执行
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 host package-group update	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

卫星支持为单一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”](#) 或编辑现有主机中所述。

您可以将不同类型的网络接口附加到主机。在添加新接口时，请选择以下之一：

- **接口**：允许您指定其他物理或者虚拟接口。您可以创建两种类型的虚拟接口。当主机需要使用单一接口与多个（虚拟）网络通信时，使用 VLAN，而这些网络不可相互访问。使用 **alias** 向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**：创建绑定接口。NIC 绑定是将多个网络接口绑定到一个接口，它显示为单一设备，并具有单个 MAC 地址。这可让两个或者多个网络接口作为一个网络接口，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 接口的更多信息，请参阅 [第 5.5 节“添加基板管理控制器\(BMC\)接口”](#)。



注意

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



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

使用这个步骤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** 选项卡中，点 **添加接口**。
4. 保留在 **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）关联的胶囊服务器中，以及与所选子网（"DNS PTR" 字段）关联的胶囊服务器。因此单个主机可以有多个 DNS 条目。

8. 从 **Domain** 列表选择一个域。要创建并管理域，请导航到 **Infrastructure > Domains**。
9. 从 **Subnet** 列表选择一个子网。要创建和管理子网，请导航到 **Infrastructure > Subnets**。
10. 指定 **IP 地址**。具有分配的 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 的托管接口会自动提供推荐的 IP 地址。
11. 选择接口是否被 **管理**。如果接口被管理，则在置备过程中从关联的胶囊服务器中拉取配置，并且创建 DNS 和 DHCP 条目。如果使用 kickstart 调配，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。主接口中的 DNS 名称用作 FQDN 的主机部分。
13. 选择这是否是主机的 **Provision** 接口。TFTP 引导通过置备接口进行。对于基于镜像的调配，通过置备接口执行要完成调配的脚本。
14. 选择是否将接口用于远程执行。
15. 清除 **虚拟 NIC** 复选框。
16. 点 **OK** 保存接口配置。
17. 单击 **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。别名接口自动从其附加到的接口继承 MAC 地址，因此您可以在不指定 MAC 地址的情况下创建一个别名。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** 选项卡中，点 **添加接口**。
4. 保留在 **Type** 列表选择的 **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。指定受管虚拟接口的 **MAC 地址**，以便正确生成用于调配的配置文件。但是，不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，请以 **eth1:10** 的形式使用 ID。
6. 选择 **Virtual NIC** 复选框。特定于虚拟接口的其他配置选项会附加到表单中：
 - **标签**：（可选）将 VLAN 标签设置为通过向虚拟接口将物理网络中的网络段中继到虚拟接口。如果没有指定标签，受管接口会继承关联子网的 VLAN 标签。此字段中的用户指定条目不适用于别名接口。
 - **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. 点 **OK** 保存接口配置。

- 单击 **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

- 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
- 点您要编辑的主机旁的 **Edit**。
- 在 **接口** 选项卡中，点 **添加接口**。
- 从 **Type** 列表选择 **Bond**。其他特定于类型的配置选项会附加到表单中。
- 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** 字段中使用 **bond0** 的形式。

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。

- 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的策略的绑定模式。有关每个绑定模式的详情，请参考 [第 5.4 节“Satellite 中可用的绑定模式”](#)。
 - 附加设备**：指定一个以逗号分隔的附加设备的标识符列表。这些可以是物理接口或 VLAN。
 - bond 选项**：指定一个以空格分隔的配置选项列表，例如 **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](#)。
- 点 **OK** 保存接口配置。
- 单击 **Submit**，将更改应用到主机。

CLI 过程

- 要创建带有绑定接口的主机，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

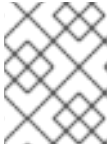
绑定模式	描述
balance-rr	在每个绑定接口上接收并按顺序发送传输。
active-backup	通过第一个可用绑定接口接收和发送传输。只有活跃绑定接口失败时，才会使用另一个绑定接口。
balance-xor	传输基于所选哈希策略。在这个模式中，目的地特定同级服务器的流量始终通过同一个接口发送。
broadcast	所有传输都会在所有绑定接口上发送。
802.a3	创建共享同一设置的聚合组。在活动组中的所有接口上传输和接收。
balance-tlb	传出流量根据每个绑定接口的当前负载分布。
balance-alb	通过地址解析协议(ARP)协商实现负载均衡。

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。

先决条件

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



注意

如果管理 BMC 接口，您只需要 BMC 接口的 MAC 地址，以便它能够创建 DHCP 保留。

流程

1. 如果还没有启用，在胶囊服务器上启用 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** 选项卡中，点 **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. 点 **OK** 保存接口配置。
 9. 单击 **Submit**，将更改应用到主机。

第6章 将主机升级到NEXT MAJOR RED HAT ENTERPRISE LINUX 发行版本

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。可能会在升级路径下：

- Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8
- Red Hat Enterprise Linux 8 to Red Hat Enterprise Linux 9

先决条件

- 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，请参阅从 RHEL 7 升级到 RHEL 8 中的[规划升级](#)。
 - 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](#)。
- 准备您的 Red Hat Enterprise Linux 主机进行升级。
 - 对于 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8，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 升级到 Red Hat Enterprise Linux 9，请参阅从 RHEL 8 升级到 RHEL 9 中的[为升级准备 RHEL 8 系统](#)。
- 在卫星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 “为远程执行分发 SSH 密钥”](#)。

流程

1. 在 Satellite 上，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** 启动预升级检查。
6. 检查完成后，点 **Leapp preupgrade report** 选项卡查看 Leapp 是否在您的主机上发现任何问题。带有 **Inhibitor** 标记的问题被视为关键，可能会破坏升级过程。具有 **Has Remediation** 标志的问题包含了可帮助您解决这个问题的补救。
 - a. 点带有 **Has Remediation** 标记的问题扩展它。
 - 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**，使用 **hint** 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. 修复问题后，点 **Rerun** 按钮，然后点 **Submit** 以再次运行预升级检查，以验证您要升级的主机没有任何问题并准备好升级。
7. 如果预升级检查验证主机没有问题，请点击 **Run Upgrade** 按钮并点 **Submit** 来启动升级。

第7章 将主机转换为 RED HAT ENTERPRISE LINUX

您可以将 Red Hat Enterprise Linux 衍生发行版本转换为主机上的可支持性 Red Hat Enterprise Linux，同时保留已安装的应用程序和配置。Satellite 提供了 **Convert2RHEL** 工具来简化转换过程。

卫星中的 **Convert2RHEL** 工具由 Ansible 角色和 Ansible playbook 组成。您可以使用 Ansible 角色在卫星服务器上生成转换数据，其中包括启用所需的存储库和创建产品、激活密钥和主机组。然后，您可以使用 Ansible playbook 在主机上执行实际的转换，它会在主机上安装 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 to Red Hat Enterprise Linux 7
- Oracle Linux 7 到 Red Hat Enterprise Linux 7
- CentOS Linux 8 to Red Hat Enterprise Linux 8
- Oracle Linux 8 到 Red Hat Enterprise Linux 8

红帽支持这些转换。

转换过程类似于 Red Hat Enterprise Linux 的次版本升级，其中每个 RPM 软件包都会被替换。Red Hat Enterprise Linux 中不可用的第三方软件包和非红帽软件包会被保留。

Convert2RHEL 工具删除不必要的软件包，如已知在转换过程中导致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 CentOS 或 Oracle 所签署的所有软件包及其同等红帽产品。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。

先决条件

- 查看从基于 RPM 的 Linux 发行版转换到 RHEL 的 [支持的转换路径](#)。
- 您必须已完成从基于 RPM 的 Linux 发行版转换到 RHEL 中的 [准备 RHEL 转换](#) 的步骤 1 到 5。
- 确保有上传到 Satellite 的订阅清单，并且有足够的 Red Hat Enterprise Linux 权利用于您所需的转换。或者，您可以使用 Ansible 变量告诉角色从磁盘导入清单。清单必须导入到要转换的机构，用于注册主机以转换。
您可以更新您的分配并 [从红帽客户门户网站下载](#) 更新的清单。如需更多信息，请参阅红帽订阅管理中的 [使用清单](#)。
- 确保已在 Satellite 中为转换主机的次 Red Hat Enterprise Linux 版本启用了红帽软件仓库。

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 变量来生成转换数据。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** 角色分配给代表卫星服务器的主机。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 [为一个现存的主机分配 Ansible 角色](#)。
4. 在卫星服务器上运行 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、激活密钥和主机组。该角色启用 **rhel-7-server-rpms** 存储库（带有 7Server 发行版本和 x86_64 架构），或 **rhel-8-for-x86_64-baseos-rpms** 和 **rhel-8-for-x86_64-appstream-rpms**，或两者均取决于您在前面的步骤中设置的变量。

5. 使用生成的主机组注册主机进行转换。

使用全局注册模板在转换前注册并订阅您的主机。选择为转换生成的主机组，例如，如果您希望从 CentOS 8 转换主机，选择 **CentOS 8 converting**。如需更多信息，请参阅 [第 3.3 节“使用全局注册注册主机”](#)。

6. 在主机上运行 Convert2RHEL playbook。使用以下设置执行远程作业：

- **作业类别** : **Convert 2 RHEL**
- **作业模板** : **转换到 RHEL**
- **激活码**: **convert2rhel_rhel7** 或 **convert2rhel_rhel8**

更多信息请参阅 [第 12.22 节“执行远程作业”](#)。

其他资源

- [如何执行一个不支持的、从 RHEL 提供的 Linux 发行版本到 RHEL 的转换](#)

7.1. 用于转换的 ANSIBLE 变量

在运行 Ansible 角色来生成转换数据前，请配置以下所需 Ansible 变量的值：

卫星从 **redhat.satellite.convert2rhel** 角色导入大多数需要的 Ansible 变量。但是，有些变量不会被导入。这些变量在下表中带有一个星号 *。您必须手动创建这些额外变量，并将其分配给 **redhat.satellite.convert2rhel** 角色。

表 7.1. 转换所需的变量

Name	类型	意图和值
satellite_server_url *	字符串	Satellite 服务器的 URL，如 https://satellite.example.com
satellite_username *	字符串	您的用户名
satellite_password *	字符串	您的密码
satellite_organization *	字符串	您的机构名称
satellite_content_rhel_wait_for_syncs *	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 ）
satellite_validate_certs *	布尔值	如果要在 Ansible 中启用证书检查，则设置为 true 。（默认值： true ）

Name	类型	意图和值
satellite_convert2rhel_manage_subscription	布尔值	如果卫星服务器上已有清单，则设置为 false 。如果您从磁盘上传新清单，则当前的清单将被覆盖。（默认值： true ）
satellite_content_rhel_enable_rhel7 *	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 ）
satellite_convert2rhel_enable_oracle7	布尔值	如果您想要为 Oracle Linux 7 准备转换数据，设置为 true 。否则，您必须将值设为 false 。
satellite_content_rhel_enable_rhel8 *	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 ）
satellite_convert2rhel_enable_oracle8	布尔值	如果您想要为 Oracle Linux 8 准备转换数据，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

Name	类型	意图和值
satellite_manifest_path *	字符串	从磁盘（如 ~/manifest.zip）上传的清单的路径。如果使用 satellite_convert2rhel_manage_subscription 从磁盘上传新清单，则必须设置此路径。
satellite_content_rhel_rhel8_releasever *	字符串	次发行版本，如 8.5 。如果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：latest）

第 8 章 使用 RHEL WEB 控制台主机管理和监控

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。您可以启用远程执行功能，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，您可以在 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，例如 Red Hat Image Builder。

8.1. 在 SATELLITE 中启用 RHEL WEB 控制台

默认情况下，RHEL web 控制台集成在 Satellite 中被禁用。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

- 在卫星服务器上，使用 `--enable-foreman-plugin-remote-execution-cockpit` 选项运行 `satellite-installer`：

```
# 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，并使用功能管理和监控 Satellite 中的主机。

先决条件

- 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 控制台指南中的[安装 web 控制台](#)。
- 卫星或胶囊可以使用 SSH 密钥向主机进行身份验证。更多信息请参阅 [第 12.14 节“为远程执行分发 SSH 密钥”](#)。

流程

1. 在 Satellite Web UI 中，进入到 **Hosts > All Hosts**，再选择您要使用 RHEL web 控制台管理和监控的主机。
2. 在主机窗口右上角，单击 **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，如 Red Hat Image Builder。

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

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

第 9 章 使用 RED HAT INSIGHTS 监控主机

在本章中，您可以查找有关使用 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 诊断与安全漏洞相关的系统和停机时间、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 监控您通过 Satellite 管理的主机，您必须首先在主机上安装 Red Hat Insights，并使用 Red Hat Insights 注册您的主机。

对于新的 Satellite 主机，您可以使用全局注册模板通过 Insights 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 或手动安装和注册您的主机，请参阅[Red Hat Insights 开始使用](#)。

Red Hat Insights 信息适用于主机

有关通过 Red Hat Insights 的主机提供了其他信息。

您可以在两个位置找到这些信息：

- 在 Satellite Web UI 中，进入 **Configure > Insights**，其中 **Remediate** 按钮旁的垂直 ellipsis 在 **Red Hat Insights** 中提供了到常规建议页面的 View。在每个推荐行中，垂直 ellipsis 在 **Red Hat Insights** 中提供了一个到推荐规则的 View，如果推荐一个文档链接，则为 **知识库文章** 链接。
- 如需其他信息，请导航到 **Hosts > All hosts**。如果主机已列出建议，请点击建议数量。在 **Insights** 标签页中，**Remediate** 按钮旁边的垂直 ellipsis 提供 **Go To Satellite Insights** 页面链接到系统的信息，以及 **Red Hat Insights** 中的 View 链接，指向控制台上主机详情。

从 rh-cloud 和 insights-client 报告排除主机

您可以将 **host_registration_insights** 参数设置为 **False**，以省略 rh-cloud 和 insight-client 报告。卫星将从 rh-cloud 报告中排除主机，并且阻止 insight-client 将报告上传到云。

您也可以在此机构、主机组、子网和域级别上设置此参数。它会阻止新报告上传，只要它们与实体关联。

如果您在 [红帽混合云](#) 上已报告的主机上将该参数设为 false，则仍然会从清单中自动删除。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 角色通过 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”](#)。

对于已存在的主机，请参阅在 Red Hat Satellite 中使用 Ansible 集成管理配置中的[使用 Ansible 角色自动化客户端中重复的任务](#)。

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。

您必须为 Insights 设置 API 令牌，然后才能继续。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 **Configure > Insights**。
2. 单击 **Start Recommendations Sync** 按钮。

如果您没有设置 API 令牌，则会使用在此页面前创建一个令牌。

其它信息

- 要查看 Red Hat Insights 和所有插件的日志，请访问 `/var/log/foreman/production.log`。
- 如果您在连接到 Red Hat Insights 时遇到问题，请确保您的证书是最新的。刷新订阅清单以更新您的证书。
- 您可以通过在一个主机上配置 `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

通过 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 计划中的建议数量。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决方案。使用 **Filter** 字段搜索特定关键字。
5. 点 **Remediate**。
6. 在 **Job Invocation** 页面中，不要更改 `precompleted` 字段的内容。
7. 可选。有关 Remote Execution Job 的更多高级配置，请单击 **Show Advanced Fields**。
8. 选择您需要的查询的类型。
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，点 **Recommendations**。
4. 在 Red Hat Insights 页面中，选择您要包括在 Insights 计划中的推荐数量，然后继续操作。

在 Jobs 窗口中，您可以查看计划的进度。

第 10 章 使用报告模板监控主机

您可以使用报告模板来查询 Satellite 数据，如主机状态、注册主机、适用勘误表、应用勘误表、订阅详情和用户活动等信息。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来满足您的要求。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](#)。

您可以创建模板，或克隆模板并编辑克隆。要获得模板语法的帮助，请单击模板并点击 **Help** 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**。要调度报告，请配置 cron 作业或使用 Satellite Web UI。

流程

1. 在 Satellite Web UI 中，导航到 **Monitor > Report Templates**。
2. 在要使用的报告模板右侧点击 **Generate**。
3. 可选：要将报告调度到 **Generate at** 字段的右侧，请点击图标来选择您要在其中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，请选中 **通过电子邮件发送报告** 复选框，并在 **Deliver to email-mail address** 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过滤器字段。
6. 点 **Submit**。包含报告的 CSV 文件会被下载。如果您选择了 **通过电子邮件发送报告** 复选框，则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。如果您想将报告生成为后台任务，您可以使用 **hammer report-template schedule** 命令。



注意

如果要生成订阅权利报告，您必须使用 **现在选项中的 Days** 指定授权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"
```

显示所有将在 60 天后过期的权利

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

10.2. 创建报告模板

在 Satellite 中，您可以创建报告模板并自定义模板以符合您的要求。您可以导入现有报告模板，并使用代码片段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。要在 Satellite Web UI 中查看有关使用 ERB 语法和宏的信息，请导航到 **Monitor > Report Templates**，再单击 **Create Template**，然后单击 **Help** 选项卡。

当您在 Satellite 中创建报告模板时，安全模式会被默认启用。

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 **Default**。
4. 直接在模板编辑器中创建模板，或通过单击 **Import** 来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. 单击 **输入** 选项卡，在 **Name** 字段中输入一个可以以下格式在模板中引用的输入名称：**输入 ('name')**。请注意，您必须保存模板，然后才能在模板正文中引用此输入值。
7. 选择是否需要输入值。如果输入值是必需的，请选择 **所需** 复选框。
8. 在 **Value Type** 列表中，选择用户必须输入的输入值类型。
9. 可选：如果要将事实用于模板输入，请选择 **Advanced** 复选框。
10. 可选：在 **Options** 字段中，定义用户可从中选择的选项。如果此字段没有定义，用户会收到可以在其中输入所需值的自由文本字段。
11. 可选：在 **Default** 字段中输入值，如主机名，您想要设置为默认模板输入。
12. 可选：在 **Description** 字段中输入您要在生成报告时作为内联帮助显示的信息。

13. 可选：单击 **Type** 选项卡，然后选择此模板是否包含在其他模板中。
14. 单击 **Location** 选项卡，再添加您要使用模板的位置。
15. 单击 **Organizations** 选项卡，再添加您要使用模板的组织。
16. 点 **Submit** 保存您的更改。

其他资源

- 有关安全模式的详情，请参考 [第10.8节“报告 Template Safe Mode”](#)。
- 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”](#)。
- 要查看填充模板的步骤示例，请参阅 [第10.7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，并从 **Actions** 列中的列表中选择 **Export**。
3. 对您要下载每个报告模板重复此操作。

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
```

请注意在此命令输出中要导出的模板模板 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. 请注意您要导出的模板的 **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。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。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 窗口右上角的 **Create Template**。
3. 在 **Editor** 选项卡的右上角，单击文件夹图标，然后选择要导入的 **.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更多信息，请参阅 [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，并分配机构和位置。有关使用 Satellite API 的更多信息，请参阅 [API 指南](#)。

先决条件

- 使用 **.erb** 语法或从另一个 Satellite 导出模板来创建模板。
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 导出模板的详情，请参考 [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. 创建报告模板以监控权利

您可以使用报告模板返回带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 `<%#>` 标签添加带有稍后引用信息的注释。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，点 **Help** 选项卡并在 **Safe 模式方法和变量** 表中找到 **Host::Managed** 行。

4. 使用 **每个** 方法添加 `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作为报告一部分的变量：

```
<%- 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. 在模板中添加最终语句：

```
<%- 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。

10.8. 报告 TEMPLATE SAFE MODE

当您在 Satellite 中创建报告模板时，安全模式会被默认启用。安全模式限制在报告模板中使用的宏和变量。安全模式可防止在报告模板中呈现问题并强制实施最佳实践。卫星 Web UI 中提供了支持的宏和变量列表。

要在 Satellite Web UI 中查看可用的宏和变量，请导航到 **Monitor > Report Templates**，然后单击 **Create Template**。在 Create Template 窗口中，单击 **Help** 选项卡并展开 **Safe 模式方法**。

虽然启用安全模式时，如果您尝试使用未在 **Safe 模式方法** 中列出的宏或变量，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 中导航到 **Administer > Settings** 并点 **Provisioning** 选项卡。找到 **Safemode rendering** 行以检查值。

第 11 章 配置主机集合

主机集合是一组内容主机。此功能允许您在同时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，以及更改分配的生命周期环境，以及更改内容视图。您可以创建主机集合来满足您的要求，以及您的公司中的那些要求。例如，按功能、部门或业务部门在主机集合中分组主机。

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 **主机集合的名称**。
4. 清除 **Unlimited Content Hosts**，在 **Limit** 自动中输入期望的最多的主机数量。
5. 添加主机集合的 **Description**。
6. 单击 **Save**。

CLI 过程

- 要创建主机集合，请输入以下命令：

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

11.2. 克隆主机集合

以下流程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. 点 **Copy Collection**。
4. 指定克隆的集合的名称。
5. 点 **Create**。

11.3. 删除主机集合

以下流程演示了如何删除主机集合。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Host Collections**。
2. 选择要删除的主机集合。
3. 单击 **Remove**。此时会出现警报框：

Are you sure you want to remove host collection Host Collection Name?

4. 单击 **Remove**。

11.4. 批量为主机集合添加主机

您可以将多个主机添加到主机集合。

先决条件

主机必须注册到红帽卫星，才能将其添加到主机集合中。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，卫星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**添加**所选项。

CLI 过程

- 要将多个主机添加到主机集合，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. 从主机集合中删除主机

以下流程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，则数据库中的主机集合记录不会被修改，因此 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. 在主机集合中添加内容

这些步骤演示了如何在 Red Hat Satellite 中的主机集合中添加内容。

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。

先决条件

- 要添加的内容应该可以在现有软件仓库之一中可用，或者在此过程前添加。
- 内容应当提升到分配了主机的环境。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Host Collections**。
2. 选择应当添加软件包的主机集合。
3. 在 **Collection Actions** 选项卡中，点 **Package Installation、Removal 和更新**。
4. 要更新所有软件包，请点击 **Update All Packages** 按钮使用默认方法。或者，选择按钮右侧的下拉图标，以选择要使用的方法。选择 **通过远程执行 - 自定义第一个** 菜单条目将进入 **作业调用** 页面，您可以在其中自定义操作。
5. 根据需要选择 **Package** 或 **Package Group** 按钮。
6. 在提供的字段中，指定软件包或者软件包组名称。然后点击：
 - **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，然后选择要使用的方法。选择 **通过远程执行 - 自定义第一个** 菜单条目将进入 **作业调用** 页面，您可以在其中自定义操作。
 - **更新** - 使用默认方法更新主机集合中的现有软件包。或者，选择按钮右侧的下拉图标，然后选择要使用的方法。选择 **通过远程执行 - 自定义第一个** 菜单条目将进入 **作业调用** 页面，您可以在其中自定义操作。

11.6.2. 查看安装的软件包

使用以下步骤查看主机的已安装软件包。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts** 并选择主机的名称。
2. 在 **Content** 选项卡上，**软件包** 会显示安装的软件包列表。
3. 要查看软件包的详情，请选择该软件包。
 - **Details** 标签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** 选项卡列出包含所选软件包的存储库。

4. 您可以根据 **Library** 或 **Default organization** 进行过滤。

11.6.3. 升级软件包

使用以下步骤查看主机的已安装软件包。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts** 并选择包含您要升级的软件包的主机名称。
2. 在 **内容** 选项卡上，选择 **软件包**。
3. **Status** 列显示软件包是否为 **upgradable** 还是 **Up to date**。您不能更新最新的软件包。
4. 在软件包列表中，选择您要升级的软件包，并点击行末尾的垂直 ellipsis 图标。
5. 选择 **Apply via Remote Execution** 以使用 Remote Execution，或者在要自定义 **远程执行时通过自定义远程执行应用**，以便在应用时设置时间。
6. 单击 **Submit** 以升级软件包。

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** 并选择包含您要删除的软件包的主机。
2. 在 **内容** 选项卡上，选择 **软件包**。
3. 点击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. 在主机集合中添加勘误

以下流程演示了如何在主机集合中添加勘误。

先决条件

- 要添加的勘误应该可以在现有软件仓库之一中可用，或者在此过程前添加。
- 勘误应提升到分配了主机的环境。

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，单击 **Errata Installation**。

4. 选择您要添加到主机集中的勘误表，然后单击 **安装所选项按钮**，以使用默认方法。或者，选择按钮右侧的下拉图标，以选择要使用的方法。选择 **通过远程执行 - 自定义** 第一个菜单条目，您可以进入 **作业调用** 页面，从中可以自定义操作。

11.6.6. 将勘误添加到单一主机

使用以下步骤将勘误表添加到主机。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts** 并选择您要添加勘误的主机。
2. 单击 **内容** 按钮，再选择 **勘误表** 选项卡。
3. 选择您要添加到主机的勘误，或选择列表顶部的复选框来添加所有可安装勘误表。点您要从完整列表中删除的任何勘误旁的复选框。
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** 以使用 Remote Execution，如果您想要 **自定义远程执行**，请选择 **Apply**。如果您没有使用 SSH 连接到目标主机，请选择 **Apply via Katello 代理**。
5. 点 **Submit**。

11.6.7. 应用安装勘误表

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 **内容** 选项卡中，**勘误** 显示所选主机的可安装勘误表。
4. 单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，选择 **Apply via Katello 代理**。
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表。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**，并点击主机的名称。
2. 在 **Contents** 选项卡上，**Errata** 列出与所选主机关联的勘误。
3. 点 **Type** 按钮根据类型过滤勘误。

4. 您可以过滤来显示安全、错误修复或功能增强的勘误
5. 点 **Severity** 按钮按严重性进行过滤。
6. 您可以过滤来显示严重性 N/A、Low、Moderate、Important 或 Critical 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。

在显示前，您还可以使用主机页面中的 **Errata** 卡来预过滤类型。

11.6.9. 从主机集合中删除内容

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. 单击应删除软件包的主机集合。
3. 在 **Collection Actions** 选项卡中，点 **Package Installation、Removal 和更新**。
4. 根据需要选择 **Package** 或 **Package Group** 按钮。
5. 在提供的字段中，指定软件包或者软件包组名称。
6. 点 **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，然后选择要使用的方法。选择 **通过远程执行 - 自定义** 第一个菜单条目将进入 **作业调用** 页面，您可以在其中自定义操作。

11.6.10. 更改主机集合的生命周期环境或内容视图

以下流程演示了如何更改已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，进入 **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** 选项卡中，点 **Change assigned Life Cycle Environment 或 Content View**。
4. 选择要分配给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. 点 **Assign**。



注意

更改将在约 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

使用本节作为配置 Satellite 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 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，或者您下载的角色，您必须安装包含 Capsule 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胶囊导入到卫星中。

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，并且可以扩展来管理多个主机。更多信息请参阅 [第 12.4 节“远程执行的传输模式”](#)。

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

默认情况下包含了几个用于 shell 脚本和 Ansible 的作业模板。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都是卫星服务器的内部胶囊的客户端，因此本节适用于与卫星服务器连接的任何类型的主机，包括胶囊。

您可以一次在多个主机上运行作业，您可以使用命令中的变量对运行的作业进行更精细的控制。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 workflow

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

Satellite 只搜索启用了远程执行功能的 Capsules。

1. 卫星找到了选中了 **远程执行** 复选框的主机接口。
2. 卫星找到这些接口的子网。
3. 卫星找到分配给这些子网的远程执行胶囊。
4. 从这一组胶囊中，Satellite 选择具有最少数量的运行作业的胶囊。这样，卫星可确保作业负载在远程执行胶囊之间平衡。

如果您已启用了 **Prefer registered through Capsule for remote execution**，Satellite 会使用主机注册到的 Capsule 运行 REX 作业。

默认情况下，**Prefer registered through Capsule for remote execution** 被设置为 **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**。这可确保卫星在主机上由他们注册到的胶囊执行 REX 作业。

如果卫星没有在这个阶段找到远程执行胶囊，如果启用了 **Fallback 到 Any Capsule** 设置，则卫星会添加另一组 Capsules 来选择远程执行胶囊。Satellite 从分配给主机的以下类型的胶囊中选择最轻便加载的胶囊：

- 分配给主机子网的 DHCP、DNS 和 TFTP 胶囊
- 分配给主机的域的 DNS 胶囊
- 分配给主机的 realm 的 realm Capsule
- Puppet 服务器胶囊
- Puppet CA Capsule
- OpenSCAP Capsule

如果 Satellite 没有在这个阶段找到远程执行胶囊，如果启用了 **Enable Global Capsule** 设置，则卫星从主机的组织和位置中所有胶囊中的一系列上选加载的远程执行胶囊，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在您的基础架构中运行哪些作业，包括它们可以目标的主机。远程执行功能提供了两个内置角色：

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

您可以克隆 **Remote Execution User** 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使用 **view_job_templates** 权限调整过滤器，您只能根据匹配的作业模板查看和触发作业。您可以使用 **view_hosts** 和 **view_smart_proxies** 权限来限制哪些主机或胶囊对角色可见。

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。这在指定权限时具有更大的粒度。

您可以针对 Red Hat Satellite 运行远程执行作业，并在使用 **execute_jobs_on_infrastructure_hosts** 权限将主机注册为 Red Hat Satellite。默认情况下，标准的 **Manager** 和 **Site Manager** 角色已具有这个权限。如果使用 **Manager** 或 **Site Manager** 角色，或者使用 **execute_jobs_on_infrastructure_hosts** 权限的自定义角色，您可以根据注册的 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
```

使用本例中的第一行将 **Reboot** 模板应用到所选主机。使用第二行定义一个主机池，其名称以 **.staging.example.com** 结尾。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权限可能会随时间而变化。如果您已将一些作业调度到将来运行，并且权限更改，则可能会导致执行失败，因为在作业执行前会立即检查权限。

12.4. 远程执行的传输模式

您可以将卫星配置为使用两种不同的传输模式进行远程作业执行。

在 **ssh** 模式下，远程执行使用 SSH 服务传输作业详情。这是默认的传输模式。SSH 服务必须在目标主机上启用并激活。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，否则标准的 SSH 端口是 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** 模式的胶囊上，远程执行使用 Message Queueing Telemetry Transport (MQTT) 来发布它从卫星服务器接收的作业。主机使用 **yggdrasil** 拉取客户端，在胶囊上订阅 MQTT 代理以进行作业通知。主机收到通知后，它会通过 HTTPS 从胶囊中提取作业详情，运行作业，并将结果报告回胶囊。

要使用 **pull-mqtt** 模式，您必须在胶囊服务器上启用它，并在目标主机上配置 pull 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** 模式的胶囊，主机可以使用远程执行 pull 客户端订阅远程作业。受管主机不需要与其胶囊服务器建立 SSH 连接。

先决条件

- 您已将主机注册到卫星。
- 主机的胶囊配置为使用 **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](#)。
- 红帽卫星客户端 6 存储库在卫星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 与其胶囊通信。
- 主机可以通过 HTTPS 与胶囊通信。



注意

创建 **katello-pull-transport-migrate** 软件包，帮助用户从 Katello Agent 迁移到使用 pull 客户端的远程执行。但是，主机上安装了 Katello Agent 并不是必需的。无论安装了 Katello Agent，都可以使用 **katello-pull-transport-migrate**。

流程

1. 在主机上安装 **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
```

- On Red Hat Enterprise Linux 7 hosts:

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

该软件包将 **foreman_ygg_worker** 和 **yggdrasil** 作为依赖项安装，并在主机上启用拉取模式。主机的 **subscription-manager** 配置和使用者证书用于配置主机上的 **yggdrasil** 客户端，并启动了 pull 模式客户端 worker。

2. 可选：要验证 pull 客户端是否正在运行并正确配置，请检查 **yggdrasil** 服务的状态：

```
# systemctl status yggdrasil
```

3. 可选：在安装软件包后，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 或较低版本中，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

使用此流程创建作业模板。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. 单击 **Template** 选项卡，并在 **Name** 字段中输入作业模板的唯一名称。
4. 选择 **Default** 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. 点 **Job** 选项卡，并在 **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](#)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，安装软件包 `%{package_name}`。您还可以在模板中使用 `%{template_name}` 和 `%{job_category}`。
9. 在 **Provider Type** 列表中，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，输入超时值来终止作业（如果未完成）。
11. 可选：点 **Add Input** 来定义输入参数。执行作业时，不需要在模板中定义参数。例如，请参阅 [帮助](#) 选项卡。
12. 可选：点击 **Foreign 输入集** 来在此作业中包含其他模板。
13. 可选：在 **Effective 用户** 区域中，如果命令无法使用默认的 `remote_execution_effective_user` 设置，则配置用户。
14. 可选：如果此模板是要包含在其他模板的代码片段，点 **Type** 选项卡并选择 **Snippet**。
15. 单击 **Location** 选项卡，再添加您要使用模板的位置。
16. 单击 **Organizations** 选项卡，再添加您要使用模板的组织。
17. 点 **Submit** 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作业模板。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. 根据名称导入 ANSIBLE PLAYBOOK

您可以从胶囊上安装的集合，将名称导入到卫星，以导入 Ansible playbook。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。

先决条件

- 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 并记下其名称。
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，您可以在卫星 Web UI 中获得通知。

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 ANSIBLE PLAYBOOK

您可以从胶囊上安装的集合将所有可用 Ansible playbook 导入到卫星。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。

先决条件

- 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，您可以在卫星 Web UI 中获得通知。

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. 在 SATELLITE 中将 FALLBACK 配置为 ANY 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 Capsules 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，或者如果主机的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中选择最轻便载入的胶囊，如下所示：

- 分配给主机子网的 DHCP、DNS 和 TFTP 胶囊
- 分配给主机的域的 DNS 胶囊
- 分配给主机的 realm 的 realm Capsule
- Puppet 服务器胶囊
- 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

- 在卫星上输入 **hammer settings set** 命令，将 **Fallback** 配置为 **Any Capsule** 设置。要将值设为 **true**，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，无论胶囊是否分配给主机子网，卫星将搜索主机上的远程执行胶囊。如果要将搜索限制到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，卫星添加另一组胶囊，以便从中选择远程执行胶囊。卫星还从主机的组织和位置集合中选择最轻量加载的远程执行胶囊，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

- 在卫星上输入 **hammer settings set** 命令来配置 **Enable Global Capsule** 设置。要将值设为 **true**，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 yggdrasil
```

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。

先决条件

- 您的用户帐户分配了一个授予 **view_settings** 和 **edit_settings** 权限的角色。
- 如果要使用 **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. 为远程执行分发 SSH 密钥

对于 **ssh** 模式中的胶囊，远程执行连接是使用 SSH 进行身份验证。胶囊中的公共 SSH 密钥必须分发到您要管理的附加主机上。

确保启用了 SSH 服务并在主机上运行。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“手动将 SSH 密钥分发到远程执行”](#)。
2. [第 12.17 节“使用 Satellite API Obtain SSH Keys for Remote Execution”](#)。
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 密钥部署到卫星主机。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](#)。

卫星将用于远程执行功能的 SSH 密钥分发到从 Satellite 调配的主机。

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

12.15. 手动将 SSH 密钥分发到远程执行

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 OBTAIN SSH KEYS FOR REMOTE EXECUTION

要使用 Satellite API 从 Capsule 下载公钥，请在每个目标主机上完成以下步骤。

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。卫星提供的 Kickstart 模板默认包括这个代码片段。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 TICKETS 配置 KEYTAB

使用这个流程将 Satellite 配置为使用 keytab 获取 Kerberos ticket 授权票据。如果您没有设置 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，使新文件的权限为 `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. 将目录所有者更改为 `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。

先决条件

- 在 Kerberos 服务器中注册 Satellite 服务器
- 在 Kerberos 服务器中注册 Satellite 目标主机
- 配置并初始化 Kerberos 用户帐户以进行远程执行
- 确保 Satellite 上的 foreman-proxy 用户具有有效的 Kerberos ticket 授权票据

流程

1. 要为远程执行安装并启用 Kerberos 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，编辑第二列并添加 Kerberos 帐户的用户名。
3. 导航到 **remote_execution_effective_user**，再编辑第二列来添加 Kerberos 帐户的用户名。

验证

- 要确认 Kerberos 身份验证已准备好使用，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**。如果要使用没有经过改变的模板，执行管理主机中的 [执行一个远程作业](#)。

您可以使用默认模板作为自行开发的基础。默认任务模板已锁定以进行编辑。克隆模板并编辑克隆。

流程

1. 要克隆模板（在 **Actions** 列中），请选择 **Clone**。
2. 输入克隆的唯一名称，然后点 **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，请使用以下步骤而不是 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，以自定义 YAML Ansible 模板。您还可以在 Satellite 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** 标签页中可见的参数才能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 **Hosts > All Hosts** 并选择您要在其上执行远程作业的目标主机。您可以使用搜索字段来过滤主机列表。
2. 从 **Select Action** 列表中，选择 **Schedule a Job**。
3. 在 **Job invocation** 页面中，定义主作业设置：
4. 选择要使用的 **Job category** 和 **Job 模板**。
5. 可选：在 **Bookmark** 列表选择一个存储的搜索字符串来指定目标主机。
6. 可选：通过输入 **搜索查询** 来限制目标主机。连续到行显示受您的查询影响的主机数。使用刷新按钮重新计算更改查询后的数字。preview 图标列出了目标主机。
7. 剩余的设置取决于所选的作业模板。如需有关在模板中添加自定义参数的信息，请参阅 [创建作业模板](#)。
8. 可选：要为作业配置高级设置，请点击 **Display advanced fields**。一些高级设置取决于作业模板，以下设置是常规的：
 - **有效的用户** 定义了执行作业的用户，默认为 SSH 用户。
 - **并发级别** 定义了一次执行的最大作业数，这可在大量主机上执行作业时防止超载系统资源。
 - **超时（以秒为单位）** 定义作业应该被终止的时间间隔（如果作业尚未完成）。在定义的时间间隔期间无法启动的任务（例如，如果上一任务需要很长时间才能完成），则会取消。
 - **查询类型** 定义了搜索查询何时被评估。这有助于使查询保持最新状态以进行调度任务。
 - **执行顺序** 决定了在主机上执行该作业的顺序：字母顺序或随机。
通过 **并发级别** 和 **超时** 以中断设置，您可以定制作业执行，以适应基础架构硬件和需求。
9. 要立即运行作业，请确保 **Schedule** 设置为 **Execute now**。您还可以定义一个一次性作业，或者设置周期性作业。对于周期性任务，您可以定义开始和结束日期、运行次数和频率。您还可以使用 cron 语法来定义重复性。有关 cron 的更多信息，请参阅使用 [cron 自动执行 Linux 系统任务](#)。
10. 点 **Submit**。您可以在同一页面上的 **Recent Jobs** 部分中查看作业的状态。

CLI 过程

1. 在 Satellite 中输入以下命令：

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

2. 查找您要使用的作业模板 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. 为主机调度周期性 ANSIBLE 作业

您可以调度周期性作业，以在主机上运行 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** 并选择您要在其上执行远程作业的目标主机。
2. 在 **Ansible** 标签页中，选择 **Jobs**。
3. 单击 **Schedule 周期性作业**。
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，或导航到 **Ansible > Jobs**。

12.24. 为主机组调度周期性 ANSIBLE 作业

您可以调度周期性作业，以在主机组上运行 Ansible 角色。

流程

1. 在 Satellite Web UI 中，进入 **Configure > Host groups**。
2. 在 **Actions** 列中，为您要调度运行 Ansible 角色的主机组选择 **Configure Ansible Job**。
3. 单击 **Schedule 周期性作业**。
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**。

12.25. 监控作业

您可以在作业运行时监控作业的进度。这对可能需要的任何故障排除都有帮助。

Ansible 作业在 100 个主机上运行，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中运行后，作业才会完成。

流程

1. 在 Satellite Web UI 中，进入 **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。这将显示运行作业的主机列表。
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
```

第13章 SATELLITE 中的主机状态

在卫星中，每个主机都有一个全局状态，指示需要注意哪些主机。每个主机也具有代表特定功能的状态的子状态。当子状态的任何更改时，全局状态会被重新计算，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** 页面上找到全局状态。状态在主机名旁边显示一个小图标，并且具有与状态对应的颜色。鼠标悬停于图标上会显示一个提示提示和子状态信息，以便快速查找更多详细信息。要查看主机全局状态，请在 Satellite Web UI 中导航到 **Hosts > All Hosts**。

确定

任何子状态都没有报告错误。这个状态以颜色绿色突出显示。

警告

当没有检测到错误时，一些子状态会引发警告。例如，即使主机被配置为发送报告，主机也没有配置管理报告。最好调查任何警告以确保您的部署处于健康状态。此状态以颜色黄色突出显示。

Error

有些子状态报告失败。例如，一个运行包含一些失败的资源。这个状态以红色表示突出显示。

搜索语法

如果要根据其状态搜索主机，请使用管理 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：

搜索具有 **OK** 状态的主机：

```
global_status = ok
```

搜索拒绝的所有主机：

```
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**，然后点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

标签	全局主机状态
Active	确定
待处理	确定
没有更改	确定
没有报告	确定/警告
没有同步	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。报告由原始卷识别，并根据报告有不同的间隔。

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
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%oSèiQ

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

标签	全局主机状态	数字值
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

标签	全局主机状态	数字值
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

标签	全局主机状态	数字值
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：

您可以根据主机的最后报告状态搜索主机的配置子状态。

例如，要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期间重启一些服务的主机：

```
status.restarted > 0
```

要查找有有趣的最后一次运行的主机，可能已经发生：

```
status.interesting = true
```

第 14 章 同步模板存储库

在 Satellite 中，您可以同步作业模板、置备模板、报告模板和 Satellite 服务器之间的分区表模板，以及版本控制系统或本地目录的分区表模板。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 插件并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，进入 **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	导出方法的含义
"关联"	associate 接受的值： always、new、never	基于元数据将模板与操作系统、组织和位置相关联。	N/A
分支	branch	指定要从 Git 存储库中读取的默认分支。	指定要写入的 Git 存储库中的默认分支。
dirname	dirname	指定要从存储库下读取的子目录。	指定要写入的存储库下的子目录。
Filter	filter	仅导入与此正则表达式匹配的名称的模板。	仅导出与此正则表达式匹配的名称的模板。
强制导入	force	导入的模板将覆盖具有相同名称的已锁定模板。	N/A
锁定模板	lock	在导入具有相同名称的新模板时，请不要覆盖现有模板，除非启用了 Force import 。	N/A

参数	API 参数名称	导入的含义	导出方法的含义
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"> ● 刷新 - 从模板内容删除现有元数据，并根据当前分配和属性生成新元数据。 ● keep - 保留现有元数据。 ● 删除 - 在没有元数据的情况下导出模板。如果要手动添加元数据，则非常有用。
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，则向模板的开头添加指定字符串。	N/A
仓库	repo	定义要从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	verbose 接受的值： true、false	支持将详细消息写入这个操作的日志。	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身份验证，请验证证书：

```
# sudo -u foreman git config --global http.sslCAPath Path_To_My_Certificate
```

2. 如果要使用 SSH 连接到存储库，请执行以下步骤：

- a. 如果还没有 SSH 密钥对，请创建一个。不要指定密码短语。

```
# sudo -u foreman ssh-keygen
```

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

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

```
# sudo -u 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，使其与 Git 服务器上的目标分支匹配。
- b. 将 Repo 设置更改为与 Git 存储库匹配。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，将模板与本地目录同步非常有用。这样，您可以编辑模板并跟踪目录中编辑的历史记录。您也可以在编辑模板后将更改同步到 Satellite 服务器。

先决条件

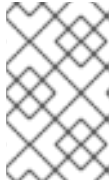
- 每个模板都必须包含模板所属的位置和组织。这适用于所有模板类型。在导入模板前，请确保将以下部分添加到模板中：

```
<%#
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
```

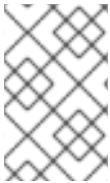
3. 更改 **TemplateSync** 选项卡上的 **Repo** 设置，以匹配 `/var/lib/foreman/My_Templates_Dir/` 目录。

14.4. 导入和导出模板

您可以使用 Satellite Web UI、Hammer CLI 或 Satellite API 来导入和导出模板。卫星 API 调用使用基于角色的访问控制系统，使任务能够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

您可以从所选存储库导入模板。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。

先决条件

- 每个模板都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[\[1\]](#)。要使用 API，请参阅[xref:api_Importing_Templates_managing-hosts\[1\]](#)。

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。有关每个字段的更多信息，请参阅 [第 14.2 节“配置 TemplateSync 插件”](#)。
4. 点 **Submit**。

卫星 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，则您无法返回该状态。

CLI 过程

- 要从存储库中导入模板，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。要从大型仓库中选择特定的模板，请使用 **--filter** 来定义要导入的模板的标题。例如，**-- filter '*Ansible Default\$'** 导入各种 Ansible Default templates。

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

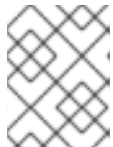
1. 在 Satellite Web UI 中，导航到 **Hosts > Sync Templates**。
2. 单击 **Export**。
3. 每个字段填充在 **Administer > Settings > TemplateSync** 中配置的值。根据需要更改您要导出的模板的值。有关每个字段的更多信息，请参阅 [第 14.2 节“配置 TemplateSync 插件”](#)。
4. 点 **Submit**。

卫星 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，则您无法返回该状态。

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** 参数在请求中指定默认 API 设置来覆盖它们。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"}'
```

14.5. 卸载 FOREMAN 模板插件

在删除 `foreman_templates` 插件后避免错误：

流程

1. 使用 Satellite 安装程序禁用插件：

```
# satellite-installer --no-enable-foreman-plugin-templates
```

2. 清理插件的自定义数据。该命令不会影响您创建的任何模板。

```
# foreman-rake templates:cleanup
```

3. 卸载插件：

```
█ # satellite-maintain packages remove foreman-plugin-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。

15.1. 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，点 **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"
  }
}
```

15.2. 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. 点击要修改的主机的名称。

3. 点 **Content** 选项卡，然后点 **Packages** 选项卡。
4. 从 **Status** 列表中选择 **Upgradable**。
5. 选择您要升级的软件包。
6. 点 **Upgrade** 按钮。远程执行作业完成后，您将获得一个 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"
  }
}
```

15.3. 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. 点击要修改的主机的名称。
3. 点 **Content** 选项卡，然后点 **Packages** 选项卡。
4. 点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，您将获得一个 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，它可将纯文本与 Ruby 代码组合。Red Hat Satellite 在以下情况下使用 ERB 语法：

虚拟机模板

如需更多信息，请参阅置备主机中的[创建置备模板](#)。

远程执行作业模板

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 使用报告模板监控主机](#)。

分区表模板

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 集成配置配置 Puppet 智能类参数](#)。

本节概述特定于 Satellite 的宏和变量，它们可以在 ERB 模板中以及一些用法示例。请注意，由 Red Hat Satellite 提供的默认模板([Hosts > Provisioning templates](#), [Hosts > Job templates](#), [Monitor > Report Templates](#))也提供了一个很好的 source of ERB 语法示例。

在置备主机或运行远程作业时，将执行 ERB 中的代码，并将变量替换为特定于主机的值。这个过程被称为 **渲染**。默认情况下，卫星服务器启用了安全模式渲染选项，这可防止从模板执行任何有害的代码。

A.1. 在卫星 WEB UI 中访问模板编写参考

您可以在 Satellite Web UI 中访问模板编写参考文档。

流程

1. 登录卫星 Web UI。
2. 在 Satellite Web UI 中，导航到 **Administer > About**。
3. 点 Support 部分中的 **Templates DSL** 链接。

A.2. 在模板中使用自动完成功能

您可以使用自动完成选项，在模板编辑器中访问可用宏和使用信息列表。这适用于卫星内的所有模板。

流程

1. 在 Satellite Web UI 中，进入 **Hosts > Partition table**, **Hosts > Provisioning templates**, or **Hosts > Job templates**。
2. 点击模板编辑器右上角的 **设置** 图标，然后选择 **Autocompletion**。
3. 在模板编辑器中按 **Ctrl + Space** 访问所有可用宏的列表。您可以通过键入更多有关您要查找的内容的信息来缩小宏列表。例如，如果您查找一种方法来列出主机的内容源的 ID，您可以键入 **主机** 并检查内容源的可用宏列表。
4. 下拉菜单旁边的窗口会提供宏描述、使用及其返回值。

5. 当您找到查找的方法时，按 **Enter** 键输入方法。

您还可以在 **设置** 菜单中启用 **Live Autocompletion**，以查看每个输入时与模式匹配的宏列表。但是，这可能会在意外的地方输入宏，如调配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内容。

`<%= %>`

这提供了与 `<% %>` 相同的功能，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，则会返回以下错误消息：

输入示例：

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

渲染示例：

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

-

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

默认情况下，如果在一行的末尾关闭，在 Ruby 块后插入换行符：

输入示例：

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

渲染示例：

```
line1
line2
```

要更改默认行为，请使用 `-%>` 修改包含的标记：

输入示例：

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

渲染示例：

```
line1line2
```

这用于减少呈现的模板中的 Ruby 语法允许的行数量。ERB 标签中的空格将被忽略。

了解如何在报告模板中使用该示例来在 FQDN 和 IP 地址之间删除不必要的换行符：

输入示例：

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

渲染示例：

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

`<%# %>`

包含在模板渲染过程中被忽略的注释：

输入示例：

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

这会生成任何输出。

ERB 模板中的缩进

由于 ERB 标签的长度不同，所以缩进 ERB 语法可能看上去。ERB 语法忽略空格。处理缩进的一个方法是，在每个新行开头声明 ERB 标签，然后使用 ERB 标签中的空格来概述语法内的关系，例如：

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

A.4. ERB 模板故障排除

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 tables**、**HostHosts > Provisioning templates** 或 **Hosts > Job templates**）时，在 **Template** 选项卡中点击 **Preview** 并选择主机。然后，模板使用所选主机的参数在文本字段中呈现。预览故障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，并点击 **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**，以查看其渲染的版本。

A.5. 通用的针对与 SATELLITE 的宏

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

表 A.1. 通用 Macros

Name	描述
indent(n)	将代码块缩进为 n 个空格，在使用未缩进的代码片段模板时很有用。
foreman_url(kind)	返回给定类型的托管模板的完整 URL。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。可用于嵌套自定义模板。
snippets(file)	呈现 Foreman 数据库中指定的代码片段，如果数据库中没有从 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，以下一些宏有不同的要求。

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

如需有关作业模板可用宏的更多信息，在 Satellite Web UI 中，导航到 **Hosts > Job Templates**，然后点击 **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。您可以定义输入名称、类型和用户可用的选项。如需报告模板，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') %>
```

这会从用户输入 **cpus** 加载值。

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 %>
```

使用带有 **each_record** 宏的 **load_hosts** 宏以 1000 批量加载记录，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 **search: input('Example_Host')**：

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

在这个示例中，您首先创建一个输入，用于优化 **load_hosts** 宏检索的搜索标准。

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
```

report_render

这个宏只可用于报告模板。

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()

这个宏仅适用于作业模板。

使用这个宏，您可以呈现特定的模板。您还可以启用并定义要传递给模板的参数。

真相

使用 **相**依宏，您可以声明传递的值为 true 或 false，无论该值是整数还是布尔值还是字符串。

这个宏有助于避免在模板包含多个值类型时混淆。例如，布尔值 **true** 与字符串值 "true" 不同。使用这个宏，您可以声明您希望模板解释值并避免混淆。

您可以使用数据源 **声明** 值，如下所示：

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

falsy

宏充当了与真实宏相同的用途。

使用 **falsy** 宏，您可以声明中传递的值是否为 true 或 false，无论该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

以下变量允许在模板中使用主机数据。请注意，作业模板只接受 `@host` 变量。

表 A.2. 特定于主机变量和 Macros

Name	描述
@host.architecture	主机的构架。
@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	系统调配的方法可以是构建（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	返回一个来自 Facter 的 Ruby 事实哈希。例如，若要从输出中访问 "ipaddress" 事实，指定 @host.facts['ipaddress']。
@host.grub_pass	返回主机的引导加载程序密码。
@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 host_enc['parameters']['lifecycle_environment'] 来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	包含包括了主接口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的数组作为输入，例如 @host.interfaces_with_identifier(['eth0', 'eth1'])。请参阅 第 A.10 节 “parsing Arrays” 。
@host.ip	主机的 IP 地址。
@host.location	主机的位置。

Name	描述
@host.mac	主机的 MAC 地址。
@host.managed_interfaces	返回一系列受管接口（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	主机的完整名称。
@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)	kernel 和 initrd 的完整路径，返回数组。
@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')	如果指定的主机参数评估为 false ，则返回 false。
host_param_true?('parameter_name')	如果指定的主机参数评估为 true ，则返回 true。
@host.primary_interface	返回主机的主接口。
@host.provider	计算资源提供商。
@host.provision_interface	返回主机的调配接口。返回接口对象。
@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 。
@host.shortname	主机的短名称。

Name	描述
@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	主机的网关。
@host.subnet.mask	主机的子网掩码。
@host.url_for_boot(:initrd)	与这个主机关联的 initrd 镜像的完整路径。不建议这样做，因为它不会插入变量。
@host.url_for_boot(:kernel)	与此主机关联的内核的完整路径。不建议使用此变量，首选 boot_files_uri。
@provisioning_type	等于 'host' 或 'hostgroup'，具体取决于置备类型。
@static	如果网络配置为静态，则返回 true 。
@template_name	正在渲染的模板的名称。
grub_pass	返回 bootloader 参数以设置加密的引导装载程序密码，如 <code>--md5pass=#{@host.grub_pass}</code> 。
ks_console	使用端口和主机的 baud 编译返回字符串，可以被添加到内核行中。例如 <code>console=ttyS1,9600</code> 。
root_pass	返回为系统配置的 root 密码。

大多数常见 Ruby 方法都可用于特定于主机的变量。例如，要提取主机的 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 自定义模板中使用。

表 A.3. Kickstart 特定变量

Name	描述
@arch	主机架构名称，与 @host.architecture.name 相同。
@dynamic	如果使用的分区表是 %pre 脚本，则返回 true （行为将 #Dynamic 选项作为表的第一行）。
@epel	用来自动安装 epel-release rpm 的正确版本的命令。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

模板中，您可以根据值存在的不同操作。要实现此目的，您可以在 ERB 语法中使用条件语句。

在以下示例中，provisioning 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 -%>
```

渲染示例

```
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 方法解析阵列。

找到 Correct Method 以解析数组

以下流程是一个示例，您可以使用它来查找用于解析模板中的数组的相关方法。本例中使用了报告模板，但步骤适用于其他模板。

1. 要检索内容主机的NIC，在本示例中，使用 `@host.interfaces` 变量返回类值，然后使用它来查找用于解析阵列的方法。

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`，在 Allowed method 或 members 列中，您可以查看以下方法来解析阵列：

```
[] 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阵列：

```
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. 要迭代接口数组，请将相关的方法添加到 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. TEMPLATE SNIPPETS 示例

检查主机是否已启用 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主版本和次版本

以下示例显示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

如果主机的子网启用了 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` 变量解析主机事实和自定义事实的值。在这个示例中 `luks_stat` 是一个自定义事实，您可以以与 `dmi::system::serial_number` 相同的方式解析，它是一个主机事实：

```
'Serial': host.facts['dmi::system::serial_number'],
'Encrypted': host.facts['luks_stat'],
```

在本例中，您可以自定义 `Applicable Errata` 报告模板来解析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.lifecycle_environment,
  '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

使用此部分作为帮助修改、自定义和扩展作业模板以满足您的要求的参考。

B.1. 自定义模板

创建作业模板时，您可以在模板编辑器字段中包含现有的模板。这样，您可以组合模板，或者从一般模板中创建更具体的模板。

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' %>
```

以上模板直接指定 `rendered` 模板的参数值。也可以使用 `input ()` 方法，允许用户在作业执行上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") %>
```

使用以上模板，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，点 **Add Foreign Input Set**，然后从 **Target 模板列表** 中选择呈现的模板。您可以导入所有参数或指定一个用逗号分开的列表。

B.2. 默认任务模板类别

作业模板类别	描述
软件包	用于执行与软件包相关的操作的模板。安装、更新和删除操作默认包含。
Puppet	用于在目标主机上运行 Puppet 的模板。
Power	用于执行与电源相关的操作的模板。默认包括重启和关闭操作。
命令	用于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。默认包括 <code>start</code> 、 <code>stop</code> 、 <code>restart</code> 和 <code>status</code> 操作。
Katello	用于执行内容相关操作的模板。这些模板主要用于卫星 Web UI 的不同部分（例如，用于内容主机的批量操作 UI），但可以单独用来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一个名为 `Run Command - restorecon` 的模板，它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**。单击 **New Job Template**。
2. 在 **Name** 字段中输入 **Run Command - restorecon**。选择 **Default** 使模板可供所有机构使用。在模板编辑器中添加以下文本：

```
restorecon -RvF <%= input("directory") %>
```

在作业调用过程中，`<%= input("directory") %>` 字符串被用户定义的目录替代。

3. 在 **Job** 选项卡上，将 **Job category** 设置为 **Commands**。
4. 单击 **Add Input** 以允许作业自定义。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。
5. 点 **Required**，以便在没有用户指定的参数的情况下无法执行该命令。
6. 从 **Input type** 列表中选择 **User 输入**。输入在作业调用期间要显示的描述，如 **Target directory for restorecon**。
7. 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。此模板在在作业执行时不需要用户输入，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。

按照 [设置作业模板](#) 所述创建新模板，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中在基于[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 **Hosts > All hosts** 并选择目标主机。从 **Select Action** 列表中，选择 **Schedule Remote Job**。
2. 在 **Job invocation** 页面中，选择 **Commands** 作业类别和 **Run Command - restorecon** 作业模板。
3. 在 **目录** 字段中键入 **/home**。
4. 将 **Schedule** 设置为 **Execute now**。
5. 点 **Submit**。您会进入 **Job invocation** 页面，从中可以监控作业执行的状态。

B.6. 模板中包括 POWER ACTIONS

本例演示了如何为执行电源操作（如重启）设置作业模板。此流程可防止 Satellite 在重新引导时解释断开连接异常，因此，远程执行作业可以正常工作。

按照 [设置作业模板](#) 所述创建新模板，并在模板编辑器中指定以下字符串：

```
<%= render_template("Power Action - SSH Default", :action => "restart") %>
```