



Red Hat Satellite 6.10

管理主机

在 Red Hat Satellite 6 环境中管理主机的指南。

Red Hat Satellite 6.10 管理主机

在 Red Hat Satellite 6 环境中管理主机的指南。

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

法律通告

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

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

目录

第 1 章 RED HAT SATELLITE 6 中的主机概述	4
第 2 章 管理主机	5
2.1. 在 RED HAT SATELLITE 中创建主机	5
2.2. 编辑主机的系统目的	6
2.3. 编辑多个主机的系统目的	7
2.4. 更改主机的模块流	8
2.5. 创建主机组	8
2.6. 为每个 LIFECYCLE 环境创建一个主机组	10
2.7. 更改主机的组	10
2.8. 更改主机的环境	10
2.9. 更改主机的受管状态	11
2.10. 将主机分配给特定机构	11
2.11. 将主机分配给特定位置	12
2.12. 从 RED HAT SATELLITE 中删除主机	12
第 3 章 注册主机	14
3.1. 将主机注册到 RED HAT SATELLITE	15
3.2. 手动将主机注册到 RED HAT SATELLITE	19
3.3. 将 ATOMIC HOST 注册到 RED HAT SATELLITE	20
3.4. 使用 BOOTSTRAP 脚本将主机注册到 RED HAT SATELLITE	21
3.5. 安装 KATELLO 代理	29
3.6. 安装 TRACER	30
3.7. 安装和配置 PUPPET 代理	30
3.8. 从 KATELLO 代理迁移到远程执行	31
第 4 章 添加网络接口	34
4.1. 添加物理接口	34
4.2. 添加虚拟接口	35
4.3. 添加绑定接口	36
4.4. 添加基板管理控制器(BMC)接口	37
第 5 章 将主机从 RHEL 7 升级到 RHEL 8	39
第 6 章 使用 RED HAT WEB 控制台主机管理和监控	40
6.1. 将 SATELLITE 与 RED HAT WEB 控制台集成	40
6.2. 使用 RED HAT WEB 控制台管理和监控主机	40
第 7 章 使用 RED HAT INSIGHTS 监控主机	41
7.1. 在 SATELLITE 中使用 RED HAT INSIGHTS 和主机	41
7.2. 为主机创建 INSIGHTS 计划	41
第 8 章 使用报告模板监控主机	43
8.1. 生成主机监控报告	43
8.2. 创建报告模板	43
8.3. 导出报告模板	44
8.4. 使用 SATELLITE API 导出报告模板	45
8.5. 导入报告模板	46
8.6. 使用 SATELLITE API 导入报告模板	47
8.7. 创建报告模板来监控权利	48
8.8. 报告模板保护模式	49
第 9 章 配置主机集合	50

9.1. 创建主机组	50
9.2. 克隆主机组	50
9.3. 删除主机组	50
9.4. 将主机添加到主机组	51
9.5. 从主机组中删除主机	51
9.6. 在主机组中添加内容	52
9.7. 从主机组中删除内容	53
9.8. 更改主机集合的生命周期环境或内容视图	53
第 10 章 配置和设置远程作业	54
10.1. 关于在主机上运行作业	54
10.2. 远程执行工作流	54
10.3. 远程执行的权限	55
10.4. 创建作业模板	55
10.5. 将 FALLBACK 配置为 SATELLITE 中的任何 CAPSULE 远程执行设置	56
10.6. 在 SATELLITE 中配置全局胶囊远程执行设置	57
10.7. 将 SATELLITE 配置为使用备用目录在主机上执行远程作业	57
10.8. 为远程执行分发 SSH 密钥	58
10.9. 手动分发用于远程执行的 SSH 密钥	58
10.10. 使用 SATELLITE API 获取远程执行的 SSH 密钥	58
10.11. 在置备过程中将 KICKSTART 模板配置为分布式 SSH 密钥	59
10.12. 为 KERBEROS TICKET GRANTING TICKETS 配置 KEYTAB	59
10.13. 为远程执行配置 KERBEROS 身份验证	60
10.14. 设置作业模板	60
10.15. 执行远程作业	61
10.16. 监控作业	62
10.17. 同步模板存储库	63
附录 A. 编写参考的模板	70
A.1. 在 SATELLITE WEB UI 中访问编写参考的模板	70
A.2. 编写 ERB 模板	70
A.3. ERB 模板故障排除	72
A.4. 通用的针对与 SATELLITE 的宏	73
A.5. 模板 MACROS	73
A.6. 特定于主机的变量	75
A.7. 特定于 KICKSTART 的变量	78
A.8. 条件声明	78
A.9. 解析阵列	79
A.10. 模板片断示例	81
附录 B. 作业模板示例和扩展	83
B.1. 自定义作业模板	83
B.2. 默认任务模板类别	83
B.3. RESTORECON 模板示例	83
B.4. 呈现 RESTORECON 模板	84
B.5. 在多个主机上执行 RESTORECON 模板	84
B.6. 在模板中包括电源操作	84

第 1 章 RED HAT SATELLITE 6 中的主机概述

主机是 Red Hat Satellite 管理的任何 Linux 客户端。主机可以是物理或虚拟主机。虚拟主机可以部署到 Red Hat Satellite 支持的任何平台上，如 KVM、VMware vSphere、OpenStack、Amazon EC2、Rackspace Cloud Services 或 Google Compute Engine。

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

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

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

要搜索主机，请在 **Search** 字段中输入一个星号，并使用星号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，**Choffnode** 也会找到内容主机 **dev-node.example.com**。



警告

Satellite 服务器被列为主机本身，即使它没有自我注册。不要从主机列表中删除 Satellite 服务器。

第 2 章 管理主机

本章论述了创建、注册、管理和删除主机。

2.1. 在 RED HAT SATELLITE 中创建主机

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

步骤

1. 在 Satellite Web UI 中，点 **Hosts > Create Host**。
2. 在 **Host** 选项卡中，输入所需的详情。
3. 单击 **Ansible Roles** 选项卡，然后从 **Ansible Roles** 列表中选择您要添加到主机的一个或多个角色。使用 **箭头图标** 管理您添加或删除的角色。
4. 在 **Puppet Classes** 选项卡上，选择您要包含的 Puppet 类。
5. 在 **Interfaces** 选项卡中：
 - a. 对于每个接口，点 **Actions** 列中的 **Edit**，并根据需要配置以下设置：
 - **Type** – 对于一个 Bond 或 BMC 接口，使用 **Type** 列表并选择接口类型。
 - **MAC 地址** IANA- Enter the MAC 地址。
 - **DNS 名称** IANA- Enter DNS 服务器已知的 DNS 名称。这用于 FQDN 的主机部分。
 - **Domain** – 选择置备网络的域名。这通过选择合适的 **子网** 自动更新子网列表。
 - 来自列表的 **IPv4 子网**，为主机选择 IPv4 子网。
 - **IPv6 子网** IANA- the 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，您可以输入地址。如果启用了置备令牌，则可以省略该地址，如果域没有强制 DNS，如果子网不管理反向 DNS，或者子网不管理 DHCP 保留。
 - **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，您可以输入地址。
 - **受管** the the 复选框，以在置备过程中配置接口，以使用 Capsule 提供的 DHCP 和 DNS 服务。
 - **主** IANA- theSelect this 复选框，将此接口的 DNS 名称用作 FQDN 的主机部分。
 - **置备** IANA- theSelect this 复选框，以使用此接口进行调配。这意味着 TFTP 引导将使用这个接口，或者在基于镜像的置备时，将通过这个接口执行置备。请注意，许多置备任务（如由 **anaconda** 在 **%post** 脚本中的 Puppet 设置下载 RPM）将使用主接口。
 - 如果这个接口不是物理设备，则 **虚拟 NIC** eXecut-DemoSelect this 复选框。此设置有两个选项：
 - **标签** rhcs- 可选地设置 VLAN 标签。如果未设置，标签将是子网的 VLAN ID。
 - **Attached to** – 输入这个虚拟机要附加到的接口名称。

- b. 单击 **OK** 以保存接口配置。
 - c. （可选）单击 **Add Interface** 使其包含额外的网络接口。详情请查看 [第 4 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，输入所需的详情。对于红帽操作系统，请选择 **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。您可以从列表中选择分区表，或者在 **Custom 分区表** 字段中输入自定义分区表。您不能同时指定。
 7. 在 **Parameters** 选项卡中，单击 **Add Parameter** 以添加您要在运行时传递给作业模板的任何参数变量。这包括您要与主机关联的所有 Puppet 类、Ansible playbook 参数和主机参数。要将参数变量与 Ansible 作业模板搭配使用，您必须添加 **Host 参数**。
 当您创建 Red Hat Enterprise Linux 8 主机时，您可以设置系统目的属性。系统目的属性定义了主机创建时自动附加哪些订阅。在 **Host Parameters** 区域中，使用对应的值输入以下参数名称。有关值列表，请参阅[执行标准 RHEL 安装指南中的配置系统目的](#)。
 - **syspurpose_role**
 - **syspurpose_sla**
 - **syspurpose_usage**
 - **syspurpose_addons**
 8. 在 **Additional Information** 选项卡中，输入有关主机的附加信息。
 9. 点 **Submit** 以完成您的拉取请求。

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 "host_name" \
--hostgroup "hostgroup_name" \
--interface="primary=true, \
    provision=true, \
    mac=mac_address, \
    ip=ip_address" \
--organization "Your_Organization" \
--location "Your_Location" \
--ask-root-password yes
```

这个命令会提示您输入 root 密码。必须指定主机的 IP 和 MAC 地址。主网络接口的其他属性可以从主机组继承，也可以使用 **--subnet**、和 **--domain** 参数进行设置。您可以使用 **--interface** 选项设置其他接口，该选项接受键值对列表。有关可用接口设置列表，请输入 **hammer host create -help** 命令。

2.2. 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。系统目的属性定义要自动附加到此主机的订阅。有关系统目的的更多信息，请参阅[执行标准 RHEL 安装指南中的配置系统目的](#)。

前提条件

- 您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 > Content Hosts**，然后单击您要编辑的 Red Hat Enterprise Linux 8 主机的名称。
2. 在 **System Purpose** 区域中，点您要编辑、添加或删除的系统目的属性的 **Edit** 或 **Remove** 图标。
3. 点 **Save**。
4. 点 **Subscriptions** 选项卡，然后选择 **Subscriptions**。
5. 点 **Run Auto-Attach** 自动将订阅附加到您的主机。
6. 刷新页面以验证订阅列表是否包含正确的订阅。

CLI 过程

1. 登录主机并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。有关值列表，请参阅 [执行标准 RHEL 安装指南中的配置系统目的](#)。

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

2. 验证此主机的系统目的属性：

```
# syspurpose show
```

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
```

2.3. 编辑多个主机的系统目的

您可以编辑 Red Hat Enterprise Linux 主机的系统目的属性。系统目的属性定义要自动附加到主机的订阅。有关系统目的的更多信息，请参阅 [执行标准 RHEL 安装指南中的配置系统目的](#)。

先决条件

- 您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 > Content Hosts** 并选择您要编辑的 Red Hat Enterprise Linux 8 主机。
2. 单击 **Select Action** 列表，再选择 **Manage System Purpose**。
3. 选择您要分配给所选主机的系统目的属性。您可以选择以下值之一：
 - 设置所有所选主机的特定属性。
 - **没有更改** 来保留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. 进入 **Hosts > Content Hosts** 并选择相同的 Red Hat Enterprise Linux 8 主机来根据系统目的自动附加订阅。
6. 单击 **Select Action** 列表，再选择 **Manage Subscriptions**。
7. 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。

2.4. 更改主机的模块流

如果您有 Red Hat Enterprise Linux 8 主机，您可以修改您安装的仓库的模块流。

创建主机后，您可以在 Satellite Web UI 中启用、禁用、安装、更新和删除模块流。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Content Hosts**，再点击包含您要更改的模块的主机名称。
2. 点 **Module Streams** 选项卡。
3. 在 **Available Module Streams** 列表中，找到您要更改的模块。您可以使用 **Filter** 字段来优化列表条目。您还可以使用 **Filter Status** 列表搜索具有特定状态的模块。
4. 从 **Actions** 列表中，选择您要对该模块所做的更改。
5. 在 **作业调用** 窗口中，确保作业信息准确。更改您需要的任何详细信息，然后单击 **Submit**。

2.5. 创建主机组

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

主机组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，主机将从主机组继承定义的设置。然后，您可以提供额外的信息来分隔主机。

要使用 CLI 而不是 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。旨在有一个基本级别的主机组，它代表您的机构中的所有主机，并提供常规设置，然后嵌套组以提供特定的设置。例如，您可以有一个基本级别的主机组，用于定义操作系统，以及继承基本级别主机组的两个嵌套主机组：

- **hostgroup: Base** (Red Hat Enterprise Linux 7.6)
 - **hostgroup: Webserver** (应用 **httpd** Puppet 类)
 - **Host: webserver1.example.com** (web server)
 - **主机 : webserver2.example.com** (网页服务器)
 - **hostgroup : 存储** (应用到 **nfs** Puppet 类)
 - **Host: storage1.example.com** (存储服务器)
 - **Host: storage2.example.com** (存储服务器)
 - **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，因为它们对 **基本** 主机组的继承。两个 Web 服务器主机继承 **Webserver** 主机组中的设置，其中包括 **httpd** Puppet 类和 **Base** 主机组中的设置。两个存储服务器继承 **存储** 主机组的设置，其中包括 **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，导航到 **Configure > Host Groups**，再点 **Create Host Group**。
2. 如果您有一个要从中继承属性的现有主机组，您可以从 **Parent** 列表中选择主机组。如果没有，请将此字段留空。
3. 输入新 **主机组的名称**。
4. 输入您希望将来主机继承的任何其他信息。
5. 单击 **Ansible Roles** 选项卡，然后从 **Ansible Roles** 列表中选择您要添加到主机的一个或多个角色。使用 **箭头图标** 管理您添加或删除的角色。
6. 单击附加选项卡，并将您要属性的任何详细信息添加到主机组。



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。

要创建要与主机组关联的适当 Puppet 环境，请手动创建目录并更改所有者：

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

7. 点 **Submit** 保存主机组。

CLI 过程

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

```
# hammer hostgroup create --name "Base" \
--lifecycle-environment "Production" --content-view "Base" \
--puppet-environment "production" --content-source-id 1 \
```

```
--puppet-ca-proxy-id 1 --puppet-proxy-id 1 --domain "example.com" \
--subnet `ACME's Internal Network` --architecture "x86_64" \
--operatingsystem "RedHat 7.2" --medium-id 9 \
--partition-table "Kickstart default" --root-pass "p@55w0rd!" \
--locations "New York" --organizations "ACME"
```

2.6. 为每个 LIFECYCLE 环境创建一个主机组

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

流程

要为每个生命周期阶段创建主机组，请运行以下 Bash 脚本：

```
MAJOR="7"
ARCH="x86_64"
ORG="Your Organization"
LOCATIONS="Your Location"
PTABLE_NAME="Kickstart default"
DOMAIN="example.com"

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
```

2.7. 更改主机的组

使用这个流程更改主机组。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
2. 选中您要更改的主机的复选框。
3. 从 **Select Action** 列表中，选择 **Change Group**。此时会打开一个新选项窗口。
4. 从 **Host Group** 列表中，选择您要主机的组。
5. 点 **Submit**。

2.8. 更改主机的环境

使用这个流程更改主机的环境。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
2. 选中您要更改的主机的复选框。
3. 从 **Select Action** 列表中，选择 **Change Environment**。此时会打开一个新选项窗口。
4. 从 **Environment** 列表中，为您的主机选择新环境。
5. 点 **Submit**。

2.9. 更改主机的受管状态

Satellite 置备的主机默认为 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** 选项卡中列出。如果您在 **Operating System** 选项卡中更改任何设置，则在设置主机来构建并重启它之前，它们才会生效。

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

使用这个流程在受管和非受管状态间切换主机。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
2. 选择主机。
3. 点 **Edit**。
4. 单击 **Manage host** 或 **Unmanage host** 以更改主机的状态。
5. 点 **Submit**。

2.10. 将主机分配给特定机构

使用这个流程将主机分配给特定机构。有关机构的一般信息以及如何配置它们，请参阅 [内容管理指南](#) 中的 [管理机构](#)。

流程

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



注意

如果存在与主机关联的资源，如域或子网，以及与您要为其分配主机的机构不关联，则会发生不匹配的情况。选项 **Fix Organization on Mismatch** 将向机构添加此类资源，因此推荐的选择。**Mismatch 选项 Fail 将始终产生错误消息**。例如，将主机从一个机构重新分配给另一个机构将失败，即使设置中没有实际不匹配。

5. 点 **Submit**。

2.11. 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。有关位置的一般信息以及如何配置它们，请参阅 [内容管理指南中的创建位置](#)。

流程

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



注意

如果存在与主机关联的资源，如域或子网，以及与您要为其分配主机的位置不关联，则会发生不匹配的情况。选项 **Fix Location on Mismatch** 会将此类资源添加到位置，因此推荐的选择。**Mismatch 选项 Fail 将始终产生错误消息**。例如，将主机从一个位置重新分配给另一个位置将失败，即使设置中没有实际不匹配。

5. 点 **Submit**。

2.12. 从 RED HAT 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** 或 **Hosts > Content Hosts**。请注意，从 **All hosts** 页或从 **Content Hosts** 页删除主机没有区别。在这两种情况下，Satellite 会完全删除主机。
2. 选择您要删除的主机。
3. 从 **Select Action** 列表中，选择 **Delete Hosts**。
4. 单击 **Submit**，以从 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 关联的虚拟机** 被设置为 **no**。如果删除了与虚拟机关联的主机记录，则虚拟机将保留在计算资源中。

要删除计算资源上的虚拟机，请进入到 **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，将 **Destroy associated VM on host delete** 设置为 **yes** 会删除虚拟机。为避免在这种情况下删除虚拟机，请从 Satellite 中解除关联，而不将其从计算资源中移除或更改设置。

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

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

第 3 章 注册主机

将主机注册到 Satellite 服务器或 Capsule 服务器有三个主要方法：

1. **(默认方法)** 从 Satellite 生成 `curl` 命令，并从无限数量的主机运行这个命令，以使用全局注册模板注册它们。此方法适用于之前注册的全新安装的主机和主机，例如 Satellite 5 或其他 Satellite 6。
通过使用此方法，您还可以在注册 Satellite 期间将 Satellite SSH 密钥部署到主机，以启用远程执行任务的主机。有关远程执行作业的更多信息，[请参阅配置和设置远程作业](#)。

通过使用此方法，您还可以在 Satellite 注册过程中使用 Red Hat Insights 配置主机。有关在 Satellite 主机中使用 Insights 的更多信息，[请参阅使用 Red Hat Insights 监控主机](#)。
2. 下载并安装消费者 RPM `server.example.com/pub/katello-ca-consumer-latest.noarch.rpm`，然后运行 Subscription Manager。此方法适用于新安装的主机。
3. **(已弃用)** 下载并运行 bootstrap 脚本(`server.example.com/pub/bootstrap.py`)。此方法适用于之前注册的全新安装的主机和主机，例如 Satellite 5 或其他 Satellite 6。

您还可以将 Atomic Hosts 注册到 Satellite 服务器或 Capsule 服务器。

使用以下步骤之一注册主机：

- [第 3.1 节 “将主机注册到 Red Hat Satellite”](#)
- [第 3.2 节 “手动将主机注册到 Red Hat Satellite”](#)
- [第 3.3 节 “将 Atomic Host 注册到 Red Hat Satellite”](#)
- [第 3.4 节 “使用 Bootstrap 脚本将主机注册到 Red Hat Satellite”](#)

使用以下步骤安装和配置主机工具：

- [第 3.5 节 “安装 Katello 代理”](#)
- [第 3.6 节 “安装 Tracer”](#)
- [第 3.7 节 “安装和配置 Puppet 代理”](#)

支持的主机操作系统

主机必须使用以下 Red Hat Enterprise Linux 版本之一：

- 6.1 或更高版本*
- 7.0 或更高版本
- 8.0 或更高版本



注意

Red Hat Enterprise Linux 6.1、6.2 和 6.3 需要手动更新 **subscription-manager** 和相关软件包。如需更多信息，[请参阅 https://access.redhat.com/solutions/1256763](https://access.redhat.com/solutions/1256763)。

支持的构架

支持 Red Hat Enterprise Linux 的所有构架：

- i386
- x86_64
- s390x
- ppc_64

3.1. 将主机注册到 RED HAT SATELLITE

通过在 Satellite 上生成 `curl` 命令并在主机上运行此命令，可以将主机注册到 Satellite。此方法使用两种模板：**全局注册** 模板和 **主机初始配置** 模板。这可让您完全控制主机注册过程。您可以通过导航到 `Administer > Settings`，然后点击 `Provisioning` 选项卡来设置默认模板。

先决条件

- 生成 `curl` 命令的 Satellite 用户必须具有 `create_hosts` 权限。
- 您必须在要注册的主机上具有 `root` 权限。
- 您必须已创建了激活码。
- 可选：如果要注册到 Red Hat Insights，您必须同步 `rhel-7-server-rpms` 存储库，并使其可用在您使用的激活码中。这是在主机上安装 `insights-client` 软件包所必需的。
- 可选：如果要通过 Capsule 注册主机，请确保在此 Capsule 上启用了 **Registration** 功能。导航到 `Infrastructure > Capsules`，点您要使用的 Capsule，然后在 `Active features` 列表中找到 **Registration** 功能。

可选：如果您的 Capsule 上没有启用 **Registration** 功能，请在 Capsule 中输入以下命令启用它：

```
# satellite-installer \
--foreman-proxy-registration true \
--foreman-proxy-templates true \
--foreman-proxy-template-url 'http://capsule.example.com'
```

步骤

1. 进入 `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：从 `Host Group` 列表中，选择要将主机关联的主机组。从 **主机组继承值的字段**：**操作系统**、**激活码** 和 **生命周期环境**。
5. 可选：从 `Capsule` 列表中，选择要通过的 Capsule 注册主机的 Capsule。如果您不想使用外部 Capsule，则必须选择内部 Capsule。
6. 可选：从 `Operating system` 列表中选择您要注册的主机操作系统。
7. 可选：如果您要使第一个调用不安全，请选择 **Insecure** 选项。在此第一次调用期间，主机从 Satellite 下载 CA 文件。主机将使用此 CA 文件与以后的所有调用连接 Satellite，使其安全。建议您避免不安全的调用。

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

相反，您可以在注册主机前在每个主机上手动复制并安装 CA 文件。

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

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

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

然后，使用安全 **curl** 命令注册主机，例如：

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

以下是带有 **--insecure** 选项的 **curl** 命令示例：

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

8. 选择 **Advanced** 选项卡。
9. 从 **Setup REX** 列表中，选择是否要将 Satellite SSH 密钥部署到主机。
如果设置为 **Yes**，则会在注册的主机上安装公共 SSH 密钥。继承的值基于 **host_registration_remote_execution** 参数。它可以继承，例如从主机组、操作系统或机构中继承。覆盖后，所选值将存储在 `host` 参数级别上。
10. 从 **Setup Insights** 列表中，选择是否要安装 **insights-client** 并将主机注册到 Insights。
Insights 工具仅适用于 Red Hat Enterprise Linux。它对其他操作系统没有影响。

您必须在注册的机器中启用以下软件仓库：

- RHEL 6: **rhel-6-server-rpms**
- RHEL 7: **rhel-7-server-rpms**
- RHEL 8: **rhel-8-for-x86_64-appstream-rpms**
在 RHEL 8 上默认安装 **insights-client** 软件包，但使用 "Minimal Install" 选项部署 RHEL 8 的环境中。

11. 可选：在 **Token 生命周期（小时）** 字段中，更改 Satellite 用于身份验证的 JSON Web Token (JWT) 的有效性持续时间。此令牌的持续时间定义了生成的 **curl** 命令的工作方式。您可以将持续时间设置为 0 - 999 999 小时或无限。
请注意，Satellite 应用生成 **curl** 命令的用户的权限到主机授权。如果用户丢失或获得额外权限，则 JWT 更改的权限也会有变化。因此，不要在令牌期间删除、阻止或更改用户的权限。

JWT 范围仅限于注册端点，无法在其它任何位置使用。

12. 可选：在 **Remote Execution Interface** 字段中输入主机必须用于 SSH 连接的网络接口标识符。
如果将此字段留空，Satellite 将使用默认网络接口。
13. 可选：在 **Install packages** 字段中，列出您要在注册后在主机上安装的软件包（以空格分开）。这可以通过 **host_packages** 参数设置。

14. 可选：选择 **Update packages** 选项，以在注册后更新主机上的所有软件包。这可以通过 **host_update_packages** 参数设置。
15. 可选：在 **Repository** 字段中，输入在执行注册前要添加的存储库。例如，使 **subscription-manager** 软件包可用于注册目的非常有用。对于 Red Hat 系列发行版本，请输入存储库的 URL，例如 **http://rpm.example.com/**。
16. 可选：在 **Repository GPG 密钥 URL** 字段中，指定公钥来验证 GPG 签名软件包的签名。它需要在 ASCII 表单中使用 GPG 公钥标头指定。
17. 在 **Activation Keys** 字段中输入要分配给主机的一个或多个激活码。
18. 可选：选择 **生命周期阶段**。
19. 可选：如果要忽略订阅管理器 错误，请选择 **Ignore error** 选项。
20. 可选：如果您要在注册前删除任何 **katello-ca-consumer** rpms，并使用 **--force** 参数运行 **subscription-manager**，请选择 **Force** 选项。
21. 点 **Generate** 按钮。
22. 复制生成的 **curl** 命令。
23. 在您要注册的主机上，以 **root** 用户身份运行 **curl** 命令。

3.1.1. 自定义注册模板

如果要自定义注册过程，请使用本节中的信息。

请注意，Satellite 中的所有默认模板都被锁定。如果要自定义注册过程，您需要克隆默认模板并编辑克隆。然后，在 **Administer > Settings > Provisioning** 中修改 **Default Global registration template** 和 **Default 'Host initial configuration' template** 以指向您的自定义模板。

模板

注册过程使用以下注册模板：

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

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_remote_execution** 参数用于 **remote_execution_ssh_keys** 片断，默认值为 **true**。
- 在 **insights** 片断中使用 **host_registration_insights** 参数，默认值为 **true**。
- **host_packages** 参数用于在主机上安装软件包。
- **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** 中使用。如需更多详细信息，请参阅代码片段的详细信息。
- **encrypt_grub** 参数为主机启用加密引导装载程序密码的设置，默认值为 **false**。

要实际设置密码，请在模板中使用 `grub_pass` 宏。

片断

Linux `host_init_config` 默认模板中使用代码片段：

- 只有在 `host_registration_remote_execution` 参数为 `true` 时，`remote_execution_ssh_keys` 片断才会将 SSH 密钥部署到主机。
- 当全局参数 `host_registration_insights` 被设置为 `true` 时，`insights` 片断下载并安装 Red Hat Insights 客户端。
- `puppetlabs_repo` 和 `puppet_setup` 片断会在主机上安装 puppet 代理（仅在存在 puppet master 时）
- `host_init_config_post` 是主机初始配置期间用户自定义操作的空片段。

变量

下表描述了 全局注册 模板中使用什么变量。

表 3.1. 全局注册模板变量

变量	命令参数	描述
<code>@user</code>	<code>none</code>	当前经过身份验证的用户对象。
<code>@organization</code>	<code>organization_id</code>	如果没有设置 <code>organization_id</code> ，则设置了用户的默认机构，或者从用户机构列表中的第一个机构。
<code>@location</code>	<code>location_id</code>	如果没有设置 <code>location_id</code> ，则设置用户的默认位置，或者从用户的位置列表中的第一个位置。
<code>@hostgroup</code>	<code>hostgroup_id</code>	主机的主机组。
<code>@operatingsystem</code>	<code>operatingsystem_id</code>	主机操作系统。
<code>@setup_insights</code>	<code>setup_insights</code>	覆盖注册主机的 <code>host_registration_insights</code> 全局参数的值，并安装 insights 客户端。
<code>@setup_remote_execution</code>	<code>setup_remote_execution</code>	覆盖注册主机的 <code>host_registration_remote_execution</code> 全局参数的值，并为远程执行部署 SSH 密钥。
<code>@setup_remote_execution</code>	<code>setup_remote_execution</code>	为远程执行设置默认主机接口。
<code>@packages</code>	软件包	要安装的软件包

变量	命令参数	描述
@repo	repo	在主机上添加存储库
@repo_gpg_key_url	repo_gpg_key_url	设置存储库 GPG 密钥表单 URL
@activation_keys	activation_keys	主机激活码。
@force	force	删除任何 katello-ca-consumer rpms rpms，并使用 --force 参数运行 subscription-manager register 命令。
@ignore_subman_errors	ignore_subman_errors	忽略 subscription-manager 错误
@lifecycle_environment_id	lifecycle_environment_id	生命周期阶段 ID
@registration_url	none	/register 端点的 URL。

3.2. 手动将主机注册到 RED HAT SATELLITE

使用以下步骤手动将主机注册到 Red Hat Satellite 6。

前提条件

- Satellite 服务器、任何 Capsule 服务器和所有主机都必须与同一 NTP 服务器同步，并且启用了和运行时间同步工具。
- 守护进程 **rhsmcertd** 必须运行在主机上运行。
- 主机必须有一个激活码。如需更多信息，请参阅 [内容管理指南](#) 中的 [管理激活码](#)。
- Subscription Manager 需要是 1.10 或更高版本。软件包位于标准 Red Hat Enterprise Linux 软件仓库中。

流程

Red Hat Enterprise Linux 主机默认注册到 Red Hat Content Delivery Network。

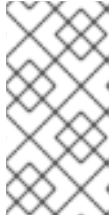
更新每个主机配置，以便它们从正确的 Satellite 服务器或 Capsule 服务器接收更新：

1. 请注意 Satellite 服务器或 Capsule 服务器的完全限定域名(FQDN)，如 *server.example.com*。
2. 以 **root** 用户身份登录主机，并从要注册到主机的 Satellite 服务器或 Capsule 服务器下载 **katello-ca-consumer-latest.noarch.rpm** 软件包。消费者 RPM 配置主机以从 Red Hat Satellite 中指定的内容源下载内容。

```
# curl --insecure --output katello-ca-consumer-latest.noarch.rpm
https://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

3. 安装 **katello-ca-consumer-latest.noarch.rpm** 软件包：

```
# yum localinstall katello-ca-consumer-latest.noarch.rpm
```



注意

RPM 软件包没有签名。如果需要，您可以添加 `--nosignature` 选项来安装软件包。`katello-ca-consumer-hostname-1.0-1.noarch.rpm` 软件包是包含服务器主机名的额外 `katello-ca-consumer` RPM。`katello-ca-consumer-latest.noarch.rpm` 软件包始终反映最新版本。这两者都满足同样目的。

- 清除与 Red Hat Subscription Manager (RHSM) 相关的任何旧主机数据：

```
# subscription-manager clean
```

- 使用 RHSM 注册主机：

```
# subscription-manager register --org=your_org_name \
--activationkey=your_activation_key
```

例 3.1. 注册后命令输出：

```
# subscription-manager register --org=MyOrg --activationkey=TestKey-1
The system has been registered with id: 62edc0f8-855b-4184-b1b8-72a9dc793b96
```



注意

您可以使用 `--environment` 选项覆盖激活码定义的内容视图和执行环境。例如，要在“开发”生命周期阶段的内容视图“MyView”中注册主机：

```
# subscription-manager register --org=your_org_name \
--environment=Development/MyView \
--activationkey=your_activation_key
```

注意

对于 Red Hat Enterprise Linux 6.3 主机，发行版本默认为 Red Hat Enterprise Linux 6 服务器，需要指向 6.3 存储库：

- 在 Satellite Web UI 中，导航到 **Hosts > Content Hosts**。
- 选中需要更改的主机旁边的复选框。
- 从 **Select Action** 列表中，选择 **Set Release Version**。
- 从 **Release Version** 列表中，选择 **6.3**。
- 点 **Done**。

3.3. 将 ATOMIC HOST 注册到 RED HAT SATELLITE

使用以下步骤将 Atomic Host 注册到 Red Hat Satellite 6。

流程

1. 以 **root** 用户身份登录 Atomic Host。
2. 从 Satellite 服务器检索 **katello-rhsm-consumer** :

```
# wget http://satellite.example.com/pub/katello-rhsm-consumer
```

3. 更改 **katello-rhsm-consumer** 的模式使其可执行 :

```
# chmod +x katello-rhsm-consumer
```

4. 运行 **katello-rhsm-consumer** :

```
#!/katello-rhsm-consumer
```

使用 Red Hat Subscription Manager 注册 :

```
# subscription-manager register
```



注意

Atomic 主机上不支持 Katello 代理。

3.4. 使用 BOOTSTRAP 脚本将主机注册到 RED HAT SATELLITE

弃用了 通过全局注册功能进行注册。

使用 bootstrap 脚本自动执行内容注册和 Puppet 配置。您可以使用 bootstrap 脚本注册新主机，或者从 Satellite 5、RHN、SAM 或 RHSM 中将现有主机迁移到 Red Hat Satellite 6。

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

要使用 bootstrap 脚本，您必须在主机上安装它。由于脚本只需要一次，并且只对 **root** 用户来说，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

- 您有一个具有运行 bootstrap 脚本所需的权限的 Satellite 用户。此流程中的示例指定 **admin** 用户。如果安全策略不接受，创建一个带有所需最小权限的新角色，并将其添加到将运行脚本的用户中。如需更多信息，请参阅 [第 3.4.1 节“为引导脚本设置权限”](#)。
- 您已启用了 Satellite 工具 6.10 存储库的主机有一个激活码。有关配置激活码的详情，请参考 [内容管理指南中的管理激活码](#)。
- 您已创建了主机组。有关创建主机组的详情，请参考 [第 2.5 节“创建主机组”](#)。

Puppet 注意事项

如果主机组与一个生产环境中创建的 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。

要创建要与主机组关联的适当 Puppet 环境，请按照以下步骤操作：

1. 手动创建目录并更改所有者：

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

2. 导航到 **Configure > Environments**，再点 **Import environment from**。按钮名称包括内部或外部 Capsule 的 FQDN。
3. 选择创建的目录，再单击 **Update**。

流程

1. 以 **root** 用户身份登录主机。
2. 下载脚本：

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

3. 使脚本可执行：

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

4. 通过查看帮助文本确认脚本可以执行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 在其他 Red Hat Enterprise Linux 版本中：

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

5. 输入适合您环境的 bootstrap 命令。

对于 **--server** 选项，指定 Satellite 服务器的 FQDN 或 Capsule 服务器。对于 **--location**、**--organization** 和 **--hostgroup** 选项，请使用带引号的名称，而不是标签，作为选项的参数。有关高级用例，请参阅 [第 3.4.2 节“高级引导脚本配置”](#)。

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# ./bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
```

```
--organization="Example Organization"\  
--hostgroup="Example Host Group"\  
--activationkey=activation_key
```

6. 输入您使用 **--login** 选项指定的 Satellite 用户的密码。
脚本将进度通知发送到 **stdout**。
7. 脚本提示时，批准主机的 Puppet 证书。在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**，并找到您使用 **--server** 选项指定的 Satellite 或 Capsule 服务器。
8. 从 **Actions** 列中的列表中，选择 **Certificates**。
9. 在 **Actions** 列中，单击 **Sign** 以批准主机的 Puppet 证书。
10. 返回到主机，以查看 bootstrap 过程完成的其余部分。
11. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**，并确保主机连接到正确的主机组。
12. 可选：主机注册完成后，删除脚本：

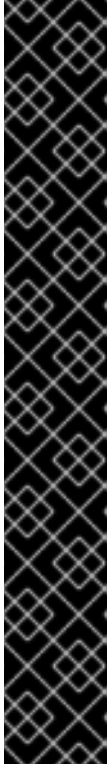
```
# rm bootstrap.py
```

3.4.1. 为引导脚本设置权限

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

步骤

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，其中包含用于修改所选用户信息的选项卡。或者，为运行此脚本的目的创建新用户。
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，并能够添加主机。如果您的安全策略不接受此功能，请使用以下权限创建新角色，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. 使用 `bootstrap` 脚本所需的最小权限创建角色。这个示例创建一个名为 *Bootstrap* 的角色：

```
# ROLE='Bootstrap'
hammer role create --name "$ROLE"
hammer filter create --role "$ROLE" --permissions view_organizations
hammer filter create --role "$ROLE" --permissions view_locations
hammer filter create --role "$ROLE" --permissions view_domains
hammer filter create --role "$ROLE" --permissions view_hostgroups
hammer filter create --role "$ROLE" --permissions view_hosts
hammer filter create --role "$ROLE" --permissions view_architectures
hammer filter create --role "$ROLE" --permissions view_ptables
hammer filter create --role "$ROLE" --permissions view_operatingsystems
hammer filter create --role "$ROLE" --permissions create_hosts
```

2. 将新角色分配给现有用户：

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

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

3.4.2. 高级引导脚本配置

本节包含更多使用 `bootstrap` 脚本注册或迁移主机的示例。



警告

这些示例指定了 **admin** Satellite 用户。如果安全策略不接受此功能，请使用 `bootstrap` 脚本所需的最小权限创建新角色。更多信息请参阅 [第 3.4.1 节“为引导脚本设置权限”](#)。

将主机从一个 Satellite 6 迁移到另一个 Satellite 6

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

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--force
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--force
```

将主机从 Red Hat Network (RHN)或 Satellite 5 迁移到 Satellite 6

`bootstrap` 脚本检测到 `/etc/syconfig/rhn/systemid` 存在以及与 RHN 的有效连接，作为系统注册到旧平台的指标。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，该脚本不会因为审计原因删除系统的传统配置集。若要删除旧配置文件，可使用 `--legacy-purge`，并使用 `--legacy-login` 提供一个具有适当权限的用户帐户来删除配置集。提示时输入用户帐户密码。

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--legacy-purge \
--legacy-login rhn-user
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--server satellite.example.com \  
--location="Example Location" \  
--organization="Example Organization" \  
--hostgroup="Example Host Group" \  
--activationkey=activation_key \  
--legacy-purge \  
--legacy-login rhn-user
```

将主机注册到 Satellite 6，省略 Puppet 设置

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

- 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="Example Location" \  
--organization="Example Organization" \  
--hostgroup="Example Host Group" \  
--activationkey=activation_key \  
--skip-puppet
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \  
--server satellite.example.com \  
--location="Example Location" \  
--organization="Example Organization" \  
--hostgroup="Example Host Group" \  
--activationkey=activation_key \  
--skip-puppet
```

仅将主机注册到 Satellite 6 以进行内容管理

要将系统注册为内容主机，并省略调配和配置管理功能，请使用 **--skip-foreman**。

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

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

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --server satellite.example.com \  
--organization="Example Organization" \  
--activationkey=activation_key \  
--skip-foreman
```

更改 bootstrap 脚本用来下载消费者 RPM 的方法

默认情况下，bootstrap 脚本使用 HTTP 下载消费者 RPM (`server.example.com/pub/katello-ca-consumer-latest.noarch.rpm`)。在某些环境中，您可能只想在主机和 Satellite 之间允许 HTTPS。使用 `--download-method` 将下载方法从 HTTP 更改为 HTTPS。

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--download-method https
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--download-method https
```

为 Satellite 提供主机的 IP 地址

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

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--ip 192.x.x.x
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--ip 192.x.x.x
```

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。

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--rex \
--rex-user root
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--rex \
--rex-user root
```

在注册过程中为主机创建域

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

- 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="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--add-domain
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中，输入以下命令：

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--add-domain
```

为主机提供替代 FQDN

如果主机的主机名不是 FQDN，或者不是 RFC 兼容的（包含下划线等字符），该脚本将在主机名验证阶段失败。如果您无法将主机更新为使用 Satellite 可接受的 FQDN，您可以使用 bootstrap 脚本指定替代 FQDN。

1. 使用hammer 将 `create_new_host_when_facts_are_uploaded` 和 `create_new_host_when_report_is_uploaded` 设置为 false :

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

2. 使用 `--fqdn` 指定将报告给 Satellite 的 FQDN :

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

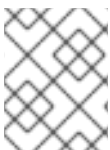
```
# /usr/libexec/platform-python bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--fqdn node100.example.com
```

- 在 Red Hat Enterprise Linux 5、6 或 7 中, 输入以下命令 :

```
# bootstrap.py --login=admin \
--server satellite.example.com \
--location="Example Location" \
--organization="Example Organization" \
--hostgroup="Example Host Group" \
--activationkey=activation_key \
--fqdn node100.example.com
```

3.5. 安装 KATELLO 代理

您可以安装 Katello 代理来远程更新 Satellite 客户端。



注意

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

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

先决条件

- 您已在 Satellite 服务器上启用了 Satellite Tools 6.10 存储库。如需更多信息, 请参阅 [从连接的网络安装 Satellite 服务器](#) 中的 [启用 Satellite 工具 6.10 存储库](#)。
- 您已在 Satellite 服务器上同步了 Satellite Tools 6.10 存储库。如需更多信息, 请参阅 [从连接的网络安装 Satellite 服务器](#) 中的 [同步 Satellite 工具 6.10 存储库](#)。
- 您已在客户端上启用了 Satellite Tools 6.10 存储库。

步骤

1. 安装 **katello-agent** 软件包：

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

2. 启动 **goferd** 服务：

```
# systemctl start goferd
```

3.6. 安装 TRACER

使用这个流程在 Red Hat Satellite 6.10 上安装 Tracer，并访问 Traces。tracer 显示过时的服务和应用程序列表，需要重启。trace 是 Satellite Web UI 中 tracer 生成的输出。

先决条件

- 主机必须注册到 Red Hat Satellite。
- 必须在 Satellite Tools 6.10 软件仓库中启用并同步 Red Hat Satellite Tools 6.10 软件仓库，并在主机上启用。

步骤

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**，然后点所需的主机名。
4. 在 **Properties** 选项卡中，检查 **Properties** 表并找到 Traces 项。如果您无法在 **Properties** 表中找到 Traces 项，则不会安装 Tracer。
5. 导航到 **Hosts > Content Hosts**，然后点所需的主机名。
6. 点 **Traces** 选项卡查看 Traces。

3.7. 安装和配置 PUPPET 代理

使用这个流程在主机上安装和配置 Puppet 代理。有关 Puppet 的更多信息，请参阅 [Puppet 指南](#)。

前提条件

- 主机必须注册到 Red Hat Satellite。
- 主机必须分配有 Puppet 环境。
- 必须在 Satellite Tools 6.10 软件仓库中启用并同步 Red Hat Satellite Tools 6.10 软件仓库，并在主机上启用。

步骤

1. 以 **root** 用户身份登录主机。
2. 安装 Puppet 代理软件包：

```
# yum install puppet-agent
```

3. 将 Puppet 代理配置为在引导时启动：

- 在 Red Hat Enterprise Linux 6 中：

```
# chkconfig puppet on
```

- 在 Red Hat Enterprise Linux 7 上：

```
# systemctl enable puppet
```

4. 将以下服务器和环境设置附加到 `/etc/puppetlabs/puppet/puppet.conf` 文件中：将 **environment** 参数设置为主机所属的 Puppet 环境的名称：

```
environment = My_Example_Org_Library
server = satellite.example.com
ca_server = satellite.example.com
```

5. 在主机上运行 Puppet 代理：

```
# puppet agent -t
```

6. 在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**。
7. 从所需 Capsule 服务器的 **Actions** 列中的列表中，选择 **Certificates**。
8. 单击所需主机右侧的 **Sign**，以签署 Puppet 客户端的 SSL 证书。
9. 再次输入 **puppet agent** 命令：

```
# puppet agent -t
```

3.8. 从 KATELLO 代理迁移到远程执行

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

先决条件

- 您已在 Satellite 服务器上启用了 Satellite Tools 6.10 存储库。如需更多信息，请参阅[从连接的网络安装 Satellite 服务器中的启用 Satellite 工具 6.10 存储库](#)。
- 您已在 Satellite 服务器上同步了 Satellite Tools 6.10 存储库。如需更多信息，请参阅[从连接的网络安装 Satellite 服务器中的同步 Satellite 工具 6.10 存储库](#)。
- 之前已在内容主机上安装 **katello-agent** 软件包。

步骤

1. 在内容主机上停止 goferd 服务：

```
# systemctl stop goferd.service
```

2. 在内容主机上禁用 goferd 服务：

```
# systemctl disable goferd.service
```

3. 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 或更高版本上，请不要删除 **katello-agent** 软件包，因为删除的依赖项会破坏主机。

```
# yum remove katello-agent
```

4. 将远程执行 SSH 密钥分发到内容主机。如需更多信息，请参阅 [第 10.8 节“为远程执行分发 SSH 密钥”](#)。
5. 在 Satellite Web UI 中，导航到 **Administer > Settings**。
6. 选择 **Content** 选项卡。
7. 将 **Use remote execution 默认** 参数设为 **Yes**。

Satellite 服务器现在使用远程执行的主机管理，而不是 katello-agent。

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

表 3.2. hammer 命令

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

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

第 4 章 添加网络接口

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

您可以将多种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**：允许您指定额外的物理或虚拟接口。您可以创建两种虚拟接口：当主机需要使用单一接口与多个（虚拟）网络通信时，请使用 **VLAN**，而这些网络不能相互访问。使用 **别名** 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 4.1 节“添加物理接口”](#)。
有关添加虚拟接口的详情请参考 [第 4.2 节“添加虚拟接口”](#)。
- **Bond**：创建一个绑定接口。NIC 绑定是将多个网络接口绑定到一个接口，它显示为单个设备并只有一个 MAC 地址。这可让两个或多个网络接口充当一个，增加带宽并提供冗余。详情请查看 [第 4.3 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 [从连接的网络安装 Satellite 服务器中的在受管主机上启用电源管理](#)。有关配置 BMC 接口的详情请参考 [第 4.4 节“添加基板管理控制器\(BMC\)接口”](#)。



注意

其他接口默认启用 **Managed** 标志，这意味着新的接口由 DNS 和与所选子网关联的 DHCP 胶囊服务器置备过程中自动配置。这需要一个正确配置了 DNS 和 DHCP 胶囊式服务器的子网。如果您使用 Kickstart 方法进行主机置备，则在安装后阶段（位于 `/etc/sysconfig/network-scripts/ifcfg-interface_id`）中为受管接口自动创建配置文件。



注意

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

4.1. 添加物理接口

使用这个流程为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 **Interfaces** 选项卡中，点 **Add Interface**。
4. 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`。标识符用于创建绑定接口、VLAN 和别名时指定此物理接口。
7. 指定与主机 IP 地址关联的 **DNS 名称**。Satellite 将此名称保存在与所选域(DNS A"字段)和与所选子网关联的 Capsule 服务器(DNS PTR"字段)关联的 Capsule 服务器中。因此，单个主机可以有多个 DNS 条目。

8. 从 **Domain** 列表选择一个域。要创建和管理域，请导航到 **Infrastructure > Domains**。
9. 从 **Subnet** 列表选择子网。要创建和管理子网，请导航到 **Infrastructure > Subnets**。
10. 指定 **IP 地址**。带有分配的 DHCP Capsule 服务器的受管接口需要此设置来创建 DHCP 租期。启用 DHCP 的受管接口由建议的 IP 地址自动提供。
11. 选择接口是否为 **Managed**。如果接口被管理，则在置备过程中从关联的 Capsule 服务器拉取配置，并创建 DNS 和 DHCP 条目。如果使用 kickstart 置备，则会为接口自动创建配置文件。
12. 选择这是否为 **主机** 的主接口。主接口的 DNS 名称用作 FQDN 的主机部分。
13. 选择这是否为主机的 **Provision** 接口。TFTP 引导使用置备接口进行。对于基于镜像的调配，用于完成调配的脚本通过调配接口执行。
14. 选择是否将接口用于**远程执行**。
15. 保持 **Virtual NIC** 复选框。
16. 单击 **OK** 以保存接口配置。
17. 点 **Submit** 将更改应用到主机。

4.2. 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。别名接口会自动从它附加到的接口继承 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 **static** 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 **Interfaces** 选项卡中，点 **Add Interface**。
4. 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 4.1 节“添加物理接口”中描述的物理接口相同。为受管虚拟接口指定 **MAC 地址**，以便正确生成用于置备的配置文件。但是，不管理的虚拟接口不需要 **MAC 地址**。

如果创建 VLAN，在 **Device Identifier** 字段中指定 **eth1.10** 格式的 ID。如果创建别名，请使用 **eth1:10** 格式的 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**：可选地设置 VLAN 标签，将来自物理网络的网络段中继到虚拟接口。如果没有指定标签，则受管接口会继承相关子网的 VLAN 标签。此字段中的用户指定的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. 单击 **OK** 以保存接口配置。

- 点 **Submit** 将更改应用到主机。

4.3. 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 Web UI，请参阅 [CLI 过程](#)。

步骤

- 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
- 点您要编辑的主机旁的 **Edit**。
- 在 **Interfaces** 选项卡中，点 **Add Interface**。
- 从 **Type** 列表中选择 **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 4.1 节“添加物理接口”](#) 中描述的物理接口相同。绑定接口在 **Device Identifier** 字段中以 **bond0** 的形式使用 ID。
单个 **MAC 地址** 就足够了。
- 指定特定于绑定接口的配置选项：
 - 模式**：选择绑定模式来定义容错和负载均衡策略。如需了解每个绑定模式的简短描述，请参阅 [表 4.1“Red Hat Satellite 中提供的绑定模式”](#)。
 - 附加设备**：指定以逗号分隔的附加设备的标识符列表。可以是物理接口或 VLAN。
 - 绑定选项**：指定以空格分隔的配置选项列表，如 **miimon=100**。如需了解您可以为绑定接口指定的配置选项的详细信息，请参阅 [Red Hat Enterprise Linux 7 网络指南](#)。
- 单击 **OK** 以保存接口配置。
- 点 **Submit** 将更改应用到主机。

CLI 过程

- 要使用绑定接口创建主机，请输入以下命令：

```
# hammer host create --name bonded_interface \
--hostgroup-id 1 \
--ip=192.168.100.123 \
--mac=52:54:00:14:92:2a \
--subnet-id=1 \
--managed true \
--interface="identifier=eth1, \
mac=52:54:00:62:43:06, \
managed=true, \
type=Nic::Managed, \
domain_id=1, \
subnet_id=1" \
--interface="identifier=eth2, \
mac=52:54:00:d3:87:8f, \
managed=true, \
type=Nic::Managed, \
domain_id=1, \
```



```

        subnet_id= 1" \
--interface="identifier=bond0, \
        ip= 172.25.18.123, \
        type=Nic::Bond, \
        mode=active-backup, \
        attached_devices=[eth1,eth2], \
        managed=true, \
        domain_id= 1, \
        subnet_id= 1" \
--organization "Your_Organization" \
--location "Your_Location" \
--ask-root-password yes

```

表 4.1. Red Hat Satellite 中提供的绑定模式

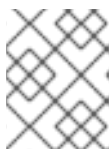
绑定模式	描述
balance-rr	每个绑定接口上都会按顺序接收并发送传输。
active-backup	传输将通过第一个可用的绑定接口接收和发送。只有在活跃的绑定接口失败时，才会使用另一个绑定接口。
balance-xor	传输基于所选哈希策略。在这个模式中，目标特定对等点的流量始终在同一接口上发送。
broadcast	所有传输都会在所有绑定接口上发送。
802.a3	创建共享相同设置的聚合组。在活动组中的所有接口上传输并接收。
balance-tlb	传出流量会根据每个绑定接口的当前负载进行分发。
balance-alb	接收负载平衡是通过地址解析协议(ARP)协商来实现的。

4.4. 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 接口详情，以及该接口的适当凭证。



注意

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

流程

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

```
# satellite-installer --foreman-proxy-bmc=true \  
--foreman-proxy-bmc-default-provider=ipmitool
```

- b. 在 Satellite Web UI 中，导航到 **Infrastructure > Capsules**。
 - c. 从 **Actions** 列中的列表中，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 **Interfaces** 选项卡中，点 **Add Interface**。
 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 4.1 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：
 - **用户名和密码**：指定 BMC 所需的任何身份验证凭证。
 - **Provider**：指定 BMC 供应商。
 8. 单击 **OK** 以保存接口配置。
 9. 点 **Submit** 将更改应用到主机。

第 5 章 将主机从 RHEL 7 升级到 RHEL 8

您可以使用作业模板将 Red Hat Enterprise Linux 7 主机升级到 Red Hat Enterprise Linux 8。

前提条件

- 确保您的 RHEL 7 主机满足升级到 RHEL 8 的要求。如需更多信息，请参阅[从 RHEL 7 升级到 RHEL 8 指南中的规划升级](#)。
- 为升级准备主机。如需更多信息，请参阅[从 RHEL 7 升级到 RHEL 8 指南中的为升级准备 RHEL 7 系统](#)。
- 在 Satellite 上启用远程执行功能。更多信息请参阅[第 10 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 10.8 节 “为远程执行分发 SSH 密钥”](#)。

流程

1. 在 Satellite 中，启用 **foreman_plugin_leapp** puppet 模块：

```
# 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 Red Hat Enterprise Linux 8 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 是否发现 RHEL 7 主机上的任何问题。具有 **Inhibitor** 标志的问题被视为至关重要，并可能会破坏升级过程。具有 **Has Remediation** 标志的问题包含可帮助您解决这个问题的补救。
 - a. 点标记为 **Has Remediation** 的问题来扩展它。
 - 如果问题包含补救 **命令**，您可以使用远程执行直接从 Satellite 修复它。选择问题。
 - 如果问题只包含补救 **Hint**，请使用提示来手动修复主机上的问题。对其他问题重复此步骤。
 - b. 在选择了补救命令中的任何问题后，点 **Fix Selected** 并提交作业。
 - c. 解决了这个问题后，点 **Rerun** 按钮，然后点 **Submit** 来再次运行预升级检查，以验证您要升级的主机是否有任何问题，并准备好升级。
7. 当您的系统准备好升级时，点 **Run Upgrade** 按钮，然后点 **Submit** 启动升级。

第 6 章 使用 RED HAT WEB 控制台主机管理和监控

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

6.1. 将 SATELLITE 与 RED HAT WEB 控制台集成

默认情况下，在 Satellite 中禁用了红帽 Web 控制台集成。如果要从 Satellite 中访问主机的 Red Hat Web 控制台功能，您必须首先在 Satellite 上启用 Red Hat Web 控制台集成。

流程

- 在 Satellite 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
```

6.2. 使用 RED HAT WEB 控制台管理和监控主机

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

先决条件

- Red Hat Web 控制台在 Satellite 中启用。
- Red Hat 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](#)。
- Satellite 或 Capsule 可以使用 SSH 密钥向主机进行身份验证。如需更多信息，请参阅 [第 10.8 节 “为远程执行分发 SSH 密钥”](#)。

流程

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

现在，您可以通过 Red Hat Web 控制台访问主机监控和管理的完整功能。

有关开始使用 Red Hat web 控制台的更多信息，[请参阅使用 RHEL 8 web 控制台指南](#) 或使用 [RHEL 7 web 控制台管理系统指南](#)。

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

第 7 章 使用 RED HAT INSIGHTS 监控主机

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

7.1. 在 SATELLITE 中使用 RED HAT INSIGHTS 和主机

您可以使用 Red Hat Insights 来诊断与安全漏洞、性能下降和稳定性故障相关的系统和停机时间。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，查看影响和分辨率的详细信息，然后确定哪些系统会受到影响。

要使用 Red Hat Insights 监控您使用 Satellite 管理的主机，您必须首先在主机上安装 Red Hat Insights，并使用 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在注册过程中使用 Insights 安装和配置 Satellite 主机。如需更多信息，请参阅 [管理主机](#) 指南中的使用 [全局注册模板将主机注册到 Red Hat Satellite](#)。

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

使用 Ansible 角色部署 Red Hat Insights

您可以使用 `RedHatsights.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 [Satellite 中的 Ansible 入门](#) 以使用 Ansible。

1. 将 `RedHatsights.insights-client` 角色添加到主机。
对于新主机，请参阅 [第 2.1 节“在 Red Hat Satellite 中创建主机”](#)。

对于现有主机，请参阅 [配置 Satellite 来使用 Ansible](#) 中的 [在 Satellite 主机上使用 Ansible 角色自动重复任务](#)。
2. 要在主机上运行 `RedHatsights.insights-client` 角色，请导航到 **Hosts > All Hosts**，然后点击您要使用的主机名称。
3. 单击 **Run Ansible roles** 按钮。

角色完成后，您可以查看并使用您在 Satellite Web UI 的 **Insights > Overview** 页面中添加的主机。

其它信息

- 要将任何系统更新应用到 Red Hat Insights 插件，请在更新后输入 `httpd` 重启。
- 要查看 Red Hat Insights 的日志和所有插件，请转至 `/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](#)。

7.2. 为主机创建 INSIGHTS 计划

使用 Satellite 6，您可以创建一个 Red Hat Insights 修复计划，并在 Satellite 主机上运行计划。

流程

要创建计划，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Configure > Insights**。
2. 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** 字段搜索特定的关键字。
5. 点 **Remediate**。
6. 在 **Job Invocation** 页面中，不要更改预完成字段的内容。
7. 可选。如需远程执行作业的更多高级配置，请单击 **Show Advanced Fields**。
8. 选择您需要的 **查询类型**。
9. 选择您需要的 **Schedule**。
10. 点 **Submit**。

在 Jobs 窗口中，您可以查看计划的进度。

第 8 章 使用报告模板监控主机

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

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，请点击模板并点 **Help** 选项卡。

8.1. 生成主机监控报告

要查看报告模板，在 Satellite Web UI 中导航至 **Monitor > Report Templates**。

要调度报告，您可以配置 cron 任务或使用 Satellite Web UI。

流程

1. 在 Satellite Web UI 中，导航到 **Monitor > Report Templates**。
2. 在您要使用的报告模板右侧，点 **Generate**。
3. 可选：要在 **Generate at** 字段右侧调度报告，请点击图标来选择您要生成报告的日期和时间。
4. 可选：要将报告发送到电子邮件地址，请选择 **通过电子邮件** 复选框发送报告，并在 **发送到电子邮件地址** 字段中输入所需的电子邮件地址。
5. 可选：应用搜索查询过滤器。要查看所有可用的结果，请不要使用任何值填充 filter 字段。
6. 点 **Submit**。包含报告的 CSV 文件会被下载。如果您已 **通过电子邮件复选框** 选择了 **Send report**，则主机监控报告将发送到您的电子邮件地址。

CLI 过程

要生成报告，请完成以下步骤：

1. 列出所有可用报告模板：

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

2. 生成一个报告：

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

此命令会等待报告在完成前完全生成。如果要报告生成为后台任务，您可以使用 **hammer report-template schedule** 命令。

8.2. 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。您可以导入现有的报告模板，并使用片断和模板宏进一步自定义它们。

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

当您在 Satellite 中创建报告模板时，默认启用安全模式。有关安全模式的详情，请参考 [第 8.8 节“报告模板保护模式”](#)。

有关编写模板的更多信息，请参阅 [附录 A, 编写参考的模板](#)。

要通过填充模板的步骤示例来查看步骤，请参阅 [第 8.7 节“创建报告模板来监控权利”](#)。

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.5 节“模板 Macros”](#)。

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. 在模板编辑器中直接创建模板，或通过单击 **Import** 从文本文件导入模板。有关导入模板的更多信息，请参阅 [第 8.5 节“导入报告模板”](#)。
5. 可选：在 **Audit 注释** 字段中，您可以添加有关此模板的任何有用信息。
6. 单击 **Input** 选项卡，在 **Name** 字段中输入您可以在模板中以以下格式引用输入的名称：**input ('name')**。请注意，您必须在模板正文中引用此输入值前保存该模板。
7. 选择输入值是否是必需的。如果输入值是必需的，请选择 **Required** 复选框。
8. 从 **Value Type** 列表中，选择用户必须输入的输入值类型。
9. 可选：如果要使用 facts 进行模板输入，请选择 **高级** 复选框。
10. 可选：在 **Options** 字段中，定义用户可以从中的选择的选项。如果此字段保持未定义，用户会收到一个自由文本字段，其中可以输入他们所需的值。
11. 可选：在 **Default** 字段中输入值，如主机名，您要设置为默认模板输入。
12. 可选：在 **Description** 字段中，您可以在生成报告时输入您要显示的信息作为内联帮助。
13. 可选：点 **Type** 选项卡，然后选择此模板是否为要包含在其他模板中的代码片段。
14. 点 **Location** 选项卡，再添加您要使用模板的位置。
15. 单击 **Organizations** 选项卡，再添加您要使用该模板的组织。
16. 点 **Submit** 保存您的更改。

8.3. 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，从 **Actions** 列中的列表中，选择 **Export**。
3. 对您要下载的报告模板重复此操作。

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
```

请注意您要在此命令的输出中导出的模板的模板 ID。

2. 要导出报告模板，请输入以下命令：

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

8.4. 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. 使用以下请求来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/158/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**。

8.5. 导入报告模板

您可以将报告模板导入到您要创建的新模板的正文中。请注意，使用 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 8.6 节“使用 Satellite API 导入报告模板”](#)。

前提条件

- 您必须从 Satellite 导出的模板才能导入它们，才能在新模板中使用。如需更多信息，请参阅 [第 8.3 节“导出报告模板”](#)。

流程

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

有关自定义新模板的详情，请参考 [附录 A, 编写参考的模板](#)。

8.6. 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，请参考 [API 指南](#)。

前提条件

- 使用 **.erb** 语法创建模板，或者从另一个 Satellite 导出模板。
有关编写模板的更多信息，请参考 [附录 A, 编写参考的模板](#)。

有关从 Satellite 导出模板的更多信息，请参考 [第 8.4 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
"
}
```

带有 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 FQND': host.name
  )
-%>
-%>
"
```

```

-%>
<% end -%>
<%= report_render %>
"
}

```

2. 使用以下请求导入模板：

```

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

```

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

```

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

```

8.7. 创建报告模板来监控权利

您可以使用报告模板返回具有特定订阅的主机列表，并显示这些主机的内核数。

有关编写模板的更多信息，请参阅 [附录 A, 编写参考的模板](#)。

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 `<%# >` 标签添加注释，其中的信息可能有助于以后参考。例如：

```

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

```

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

```

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

```

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

4. 添加一个带有 `each` 方法的 `host.pools` 变量的一行，例如：

```

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

```

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的一部分目标的变量：

```

<%- 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 end 语句：

```

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

```

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

```

<%= report_render -%>

```

8. 点 **Submit** 保存模板。

8.8. 报告模板保护模式

当您在 Satellite 中创建报告模板时，默认启用安全模式。安全模式限制可以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。Satellite Web UI 中提供了支持的宏和变量列表。

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

启用安全模式时，如果您尝试使用没有在 **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 中进入 **Administer > Settings** 并点 **Provisioning** 选项卡。找到 **Safemode** 呈现行来检查值。

第 9 章 配置主机集合

主机聚合是一组多个内容主机。此功能允许您在多个主机上执行相同的操作。这些操作可包括安装、删除和更新软件包和勘误，以及所分配的生命周期阶段的变化，以及内容视图的变化。您可以创建主机聚合以满足您的要求以及公司的人员。例如，按功能、部门或单位在主机集合中对主机进行分组。

9.1. 创建主机组

以下流程演示了如何创建主机聚合。

步骤

1. 点 **Hosts > Host Collections**。
2. 单击 **New Host Collection**。
3. 添加主机聚合的名称。
4. 清除 **kiosk Content Hosts**，然后在 **Limit** 字段中输入所需的最大主机数量。
5. 添加主机聚合的描述。
6. 点 **Save**。

CLI 过程

- 运行以下命令来创建主机聚合：

```
# hammer host-collection create \  
--organization "Your_Organization" \  
--name hc_name
```

9.2. 克隆主机组

以下流程演示了如何克隆主机聚合。

克隆主机集合：

1. 点 **Hosts > Host Collections**。
2. 在左侧面板中，点您要克隆的主机控制器。
3. 点 **Copy Collection**。
4. 为克隆的集合指定一个名称。
5. 点 **Create**。

9.3. 删除主机组

以下流程演示了如何删除主机集合。

删除主机集合：

1. 点 **Hosts > Host Collections**。
2. 选择要删除的主机控制器。
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**。

9.4. 将主机添加到主机组

以下流程演示了如何将主机添加到主机组中。

先决条件

主机必须注册到 Red Hat Satellite 中，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 章 注册主机](#)。

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

步骤

1. 点 **Hosts > Host Collections**。
2. 单击应添加主机的主机控制器。
3. 在 **Hosts** 选项卡上，选择 **Add** 子选项卡。
4. 从表中选择要添加的主机，然后单击 **Add Selected**。

CLI 过程

要将主机添加到主机组中，请输入以下命令：

```
# hammer host-collection add-host \  
--id hc_ID \  
--host-ids host_ID1,host_ID2...
```

9.5. 从主机组中删除主机

以下流程演示了如何从主机组中删除主机。

请注意，如果您从主机组中删除主机，则不会修改数据库中的主机变量记录，因此 Satellite 审计系统不会记录更改。

从主机组中删除主机：

1. 点 **Hosts > Host Collections**。
2. 选择所需的主机聚合。
3. 在 **Hosts** 选项卡上，选择 **List/Remove** 子选项卡。
4. 选择您要从主机集合中删除的主机，然后单击 **Remove Selected**。

9.6. 在主机组中添加内容

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

9.6.1. 在主机组中添加软件包

以下流程演示了如何将软件包添加到主机组中。

先决条件

- 要添加的内容应该在其中一个现有存储库中可用，或者在此流程之前添加。
- 内容应提升到分配主机的环境。

将软件包添加到主机组中：

1. 点 **Hosts > Host Collections**。
2. 单击应添加该软件包的主机集合。
3. 在 **Collection Actions** 选项卡中，单击 **Package Installation、Removal 和 Update**。
4. 要更新所有软件包，请单击 **Update All Packages** 按钮以使用默认方法。或者，选择按钮右侧的下拉菜单图标，以选择要使用的方法。选择 **via remote execution - 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** 单选按钮。
6. 在提供的字段中，指定软件包或软件包组名称。然后单击：
 - **Install** - 使用默认方法安装新软件包。或者，选择按钮右侧的下拉菜单图标，然后选择要使用的方法。选择 **via remote execution - 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。
 - **update** - 使用默认方法更新主机变量中的现有软件包。或者，选择按钮右侧的下拉菜单图标，然后选择要使用的方法。选择 **via remote execution - 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。

9.6.2. 在主机组中添加勘误

以下流程演示了如何在主机组中添加勘误。

先决条件

- 要添加的勘误应该在其中一个现有存储库中可用，或者在此流程前添加。
- 勘误应提升到分配了主机的环境。

将勘误添加到主机组：

1. 点 **Hosts > Host Collections**。
2. 选择要在其中添加勘误表的主机集合。
3. 在 **Collection Actions** 选项卡中，点 **Errata Installation**。

- 选择您要添加到主机组的勘误，然后单击 **Install Selected** 按钮以使用默认方法。或者，选择按钮右侧的下拉菜单图标，以选择要使用的方法。选择 **via remote execution - 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。

9.7. 从主机组中删除内容

以下流程演示了如何从主机组中删除软件包。

从主机组中删除内容：

- 点 **Hosts > Host Collections**。
- 单击应删除该软件包的主机集合。
- 在 **Collection Actions** 选项卡中，单击 **Package Installation、Removal 和 Update**。
- 根据需要选择 **Package** 或 **Package Group** 单选按钮。
- 在提供的字段中，指定软件包或软件包组名称。
- 单击 **Remove** 按钮，以使用默认方法删除软件包或软件包组。或者，选择按钮右侧的下拉菜单图标，然后选择要使用的方法。选择 **via remote execution - 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。

9.8. 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分配的生命周期阶段或主机组的内容视图。

更改主机集合的生命周期环境或内容视图：

- 点 **Hosts > Host Collection**。
- 选择应该更改生命周期阶段或内容视图的主机集合。
- 在 **集合操作** 选项卡上，单击 **Change assigned Life Cycle Environment 或 Content View**
- 选择要分配给主机聚合的生命周期阶段。
- 从列表中选择所需的内容视图。
- 点 **Assign**。



注意

更改在大约 4 小时内生效。要使更改立即生效，请在主机上输入以下命令：

```
# subscription-manager refresh
```

您可以使用远程执行在多个主机上运行此命令。

第 10 章 配置和设置远程作业

使用本节作为配置 Satellite 在远程主机上执行任务的指南。

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。

10.1. 关于在主机上运行作业

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

对于您创建的自定义 Ansible 角色，或您下载的角色，您必须在 Capsule 基础操作系统上安装包含角色的软件包。在使用 Ansible 角色之前，您必须将角色从安装它们的 Capsule 中导入到 Satellite。

通信通过 Capsule 服务器进行，这意味着 Satellite 服务器不需要直接访问目标主机，并可扩展以管理许多主机。远程执行使用必须启用并在目标主机上运行的 SSH 服务。确保远程执行 Capsule 能够访问目标主机上的端口 22。

Satellite 使用 ERB 语法作业模板。有关更多信息，请参阅 [管理主机指南](#) 中的 [模板编写参考](#)。

默认包含多个 shell 脚本和 Ansible 的作业模板。如需更多信息，[请参阅设置作业模板](#)。



注意

任何 Capsule 服务器基础操作系统都是 Satellite 服务器内部胶囊的客户端，因此本节适用于连接到 Satellite 服务器的任何类型的主机，包括 Capsule。

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更精细的控制您运行的作业。您可以使用主机事实和参数来填充变量值。

另外，您可以在运行命令时为模板指定自定义值。

如需更多信息，[请参阅执行远程作业](#)。

10.2. 远程执行 workflow

当您在主机上运行远程作业时，Satellite 都会执行以下操作来查找要使用的远程执行胶囊：

Satellite 仅搜索启用了远程执行功能的 Capsule。

1. Satellite 找到选择了 **Remote execution** 复选框的主机接口。
2. Satellite 找到这些接口的子网。
3. Satellite 找到分配给这些子网的远程执行胶囊。
4. 从这一组 Capsules 中，Satellite 选择具有最少运行作业的 Capsule。通过这样做，Satellite 可确保在远程执行胶囊之间平衡作业负载。
5. 如果 Satellite 在此阶段找不到远程执行胶囊，如果启用了 **Fallback to any Capsule** 设置，Satellite 会添加另一组 Capsules 来从中选择远程执行胶囊。Satellite 从分配给主机的以下类型的 Capsule 中选择最轻量级的 Capsule：
 - 分配给主机子网的 DHCP、DNS 和 TFTP 胶囊
 - 分配给主机的域的 DNS Capsule

- 分配给主机的域的 realm Capsule
 - puppet Master Capsule
 - Puppet CA Capsule
 - OpenSCAP Capsule
6. 如果 Satellite 在此阶段找不到远程执行胶囊，如果启用了 **Enable Global Capsule** 设置，Satellite 会从主机的机构和位置集合中选择最轻量级的远程执行胶囊。

10.3. 远程执行的权限

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

- **远程执行管理器**：此角色允许访问所有远程执行功能和功能。
- **远程执行用户**：此角色仅允许运行作业；它不提供修改作业模板的权限。

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

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

本例中的第一行允许用户将 **Reboot** 模板应用到所选主机。第二行定义了名称以 **.staging.example.com** 结尾的主机池。第三行将模板与主机组绑定。



注意

分配给用户的权限可能会随着时间而改变。如果用户已在将来调度了一些作业，且权限已更改，这可能会导致执行失败，因为权限会在作业执行前立即检查。

10.4. 创建作业模板

使用这个流程创建作业模板。要使用 CLI 而不是 Web UI，请参阅 [CLI 过程](#)。

步骤

1. 进入 **Hosts > Job templates**。
2. 单击 **New Job Template**。
3. 点 **Template** 选项卡，在 **Name** 字段中输入作业模板的唯一名称。
4. 选择 **Default**，使模板可用于所有机构和位置。

5. 在模板编辑器中直接创建模板，或通过单击 **Import** 从文本文件上传该模板。
6. 可选：在 **Audit 注释** 字段中，添加有关更改的信息。
7. 单击 **Job** 选项卡，并在 **Job category** 字段中输入您自己的类别，或者从 [Default Job Template Categories](#) 中列出的默认类别中选择。
8. 可选：在 **Description Format** 字段中输入描述模板。例如，安装 **package % {package_name}**。您还可以在模板中使用 **%{template_name}** 和 **%{job_category}**。
9. 从 **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（如果它未完成）。
11. 可选：点 **Add Input** 来定义输入参数。执行作业时请求参数，且不必在模板中定义参数。例如，请参阅 **Help** 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：如果此模板是包含在其他模板中的代码片段，点 **Type** 选项卡并选择 **Snippet**。
15. 点 **Location** 选项卡，再添加您要使用模板的位置。
16. 单击 **Organizations** 选项卡，再添加您要使用该模板的组织。
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 [管理主机指南中的附加信息](#)。

CLI 过程

1. 要使用模板确定的文件创建作业模板，请输入以下命令：

```
# hammer job-template create \
--file "path_to_template_file" \
--name "template_name" \
--provider-type SSH \
--job-category "category_name"
```

10.5. 将 FALLBACK 配置为 SATELLITE 中的任何 CAPSULE 远程执行设置

您可以启用 **Fallback to any Capsule** 设置，将 Satellite 配置为从分配给主机的 Capsule 列表中搜索远程执行 Capsule。如果您需要在没有配置子网的主机上运行远程作业，或者主机子网分配给没有启用远程执行功能的 Capsule，这将非常有用。

如果启用了 **Fallback to any Capsule** 设置，Satellite 会添加另一组 Capsules 来从中选择远程执行胶囊。Satellite 还从分配给主机的所有胶囊集合中选择最轻便的 Capsule，例如：

- 分配给主机子网的 DHCP、DNS 和 TFTP 胶囊
- 分配给主机的域的 DNS Capsule

- 分配给主机的域的 realm Capsule
- puppet Master Capsule
- Puppet CA Capsule
- OpenSCAP Capsule

流程

1. 在 Satellite Web UI 中，导航到 **Administer > Settings**。
2. 单击 **RemoteExecution**。
3. 将 **Fallback** 配置为任何 Capsule 设置。

CLI 过程

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

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

10.6. 在 SATELLITE 中配置全局胶囊远程执行设置

默认情况下，Satellite 在主机的机构和位置中搜索远程执行 Capsule，无论 Capsule 是否分配给主机的子网。如果要将搜索限制为分配给主机的子网的 Capsule，您可以禁用 **Enable Global Capsule** 设置。

如果启用了 **Enable Global Capsule** 设置，Satellite 会添加另一组 Capsules 来从中选择远程执行胶囊。Satellite 还从主机的机构和位置，选择最轻便的远程执行胶囊，以执行远程作业。

流程

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

CLI 过程

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

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

10.7. 将 SATELLITE 配置为使用备用目录在主机上执行远程作业

默认情况下，Satellite 使用客户端系统上的 **/var/tmp** 目录来执行远程执行作业。如果客户端系统为 **/var/** 卷或文件系统设置 **noexec**，您需要将 Satellite 配置为使用替代目录，否则远程执行任务会失败，因为无法运行脚本。

流程

1. 创建新目录，如 `new_place`：

```
# mkdir /remote_working_dir
```

2. 从默认 `var` 目录中复制 SELinux 上下文：

```
# chcon --reference=/var /remote_working_dir
```

3. 配置系统：

```
# satellite-installer --foreman-proxy-plugin-remote-execution-ssh-remote-working-dir /remote_working_dir
```

10.8. 为远程执行分发 SSH 密钥

要使用 SSH 密钥来验证远程执行连接，您必须将公共 SSH 密钥从 Capsule 分发到您要管理的附加主机。确保 SSH 服务已启用并在主机上运行。配置任何基于网络的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 10.9 节 “手动分发用于远程执行的 SSH 密钥”。
2. 第 10.10 节 “使用 Satellite API 获取远程执行的 SSH 密钥”。
3. 第 10.11 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，您可以在使用全局注册模板注册过程中将 SSH 密钥部署到 Satellite 主机。如需更多信息，请参阅 [使用全局注册模板将主机注册到 Red Hat Satellite](#)。

默认情况下，Satellite 将远程执行功能的 SSH 密钥分发到从 Satellite 置备的主机。

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 <https://aws.amazon.com/premiumsupport/knowledge-center/new-user-accounts-linux-instance>。

10.9. 手动分发用于远程执行的 SSH 密钥

要手动分发 SSH 密钥，请完成以下步骤：

流程

1. 在 Capsule 上输入以下命令。对您要管理的每个目标主机重复：

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

2. 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@target.example.com
```

10.10. 使用 SATELLITE API 获取远程执行的 SSH 密钥

要使用 Satellite API 从 Capsule 下载公钥，请在每个目标主机上完成此流程。

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
```

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

您可以将 `remote_execution_ssh_keys` 片断添加到自定义 kickstart 模板，以在置备过程中将 SSH 密钥部署到主机。Satellite 附带的 Kickstart 模板默认包括此代码片段。因此，Satellite 在调配期间将用于远程执行的 SSH 密钥复制到系统。

流程

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

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

10.12. 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 票据授予票据。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，以便新文件具有权限 `600`：

```
# umask 077
```

3. 为 keytab 创建目录：

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

4. 创建 keytab 或将现有 keytab 复制到目录中：

```
# cp your_client.keytab /var/kerberos/krb5/user/USER_ID/client.keytab
```

- 将目录所有者改为 **foreman-proxy** 用户：

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

- 确保 keytab 文件是只读的：

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

- 恢复 SELinux 上下文：

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

10.13. 为远程执行配置 KERBEROS 身份验证

您可以使用 Kerberos 身份验证建立 SSH 连接，以便在 Satellite 主机上远程执行。

前提条件

- 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 票据授予票据

流程

- 要为远程执行安装并启用 Kerberos 身份验证，请输入以下命令：

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

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

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。

10.14. 设置作业模板

Satellite 提供可用于执行任务的默认作业模板。要查看作业模板列表，请导航到 **Hosts > Job templates**。如果要在不进行任何更改的情况下使用模板，[请继续执行远程作业](#)。

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

流程

- 要克隆模板，请在 **Actions** 列中选择 **Clone**。

2. 输入克隆的唯一名称，然后点 **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 [管理主机指南](#) 中的 [模板编写参考](#)。

Ansible 注意事项

要创建 Ansible 作业模板，请使用以下流程而不是 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 语法来自定义 YAML Ansible 模板。您还可以在 Satellite 中导入 Ansible playbook。如需更多信息，请参阅 [管理主机指南](#) 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参数变量。请注意，只有主机编辑页面中的 **参数** 才会用作作业模板的输入参数。如果您不希望 Ansible 作业模板在运行时接受参数变量，在 Satellite Web UI 中导航至 **Administer > Settings** 并点 **Ansible** 选项卡。在 **顶级 Ansible 变量** 行中，将 **Value** 参数更改为 **No**。

10.15. 执行远程作业

您可以针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 Web UI，请参阅 [CLI 过程](#)。

步骤

1. 导航到 **Hosts > All Hosts**，再选择您要在其上执行远程作业的目标主机。您可以使用 **search** 字段过滤主机列表。
2. 从 **Select Action** 列表中，选择 **Schedule Remote Job**。
3. 在 **Job 调用** 页面中，定义主作业设置：
4. 选择要使用的 **Job category** 和 **Job template**。
5. 可选：在 **书签** 列表中选择存储的搜索字符串，以指定目标主机。
6. 可选：通过输入 **搜索查询** 来限制目标主机。**Resolves to** 行显示受查询影响的主机数量。更改查询后，使用刷新按钮重新计算数字。preview 图标列出目标主机。
7. 剩余的设置取决于所选作业模板。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，以下设置通常是常规设置：
 - **有效的用户** 定义了用于执行作业的用户，默认为 SSH 用户。
 - **并发级别** 定义一次执行的最大作业数量，这可以防止在大量主机上执行作业时过载系统资源。
 - **终止的超时时间**（以秒为单位）定义作业应该被终止的时间间隔（如果还没有完成）。在定义的间隔期间无法启动的任务（例如，如果上一任务用时过长才能完成），则取消了。
 - **查询** 类型定义何时评估搜索查询。这有助于保留已调度任务的查询最新。
 - **执行顺序** 决定在主机上执行作业的顺序：字母或随机化。

通过 **并发级别** 和 **超时** 来 **终止** 设置，您可以定制作业执行以适应您的基础架构硬件和需求。

- 要立即运行作业，请确保将 **Schedule** 设置为 **Execute now**。您还可以定义一次性作业，或设置重复的作业。对于重复任务，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 Red Hat Enterprise Linux 7 [系统管理员指南中的自动化系统任务](#) 部分。
- 点 **Submit**。这将显示 **Job Overview** 页面，作业完成后也会显示作业的状态。

CLI 过程

- 在 Satellite 中输入以下命令：

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

要使用自定义参数执行远程作业，请完成以下步骤：

- 查找您要使用的作业模板的 ID：

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

- 显示模板详情以查看模板所需的参数：

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

- 使用自定义参数执行远程作业：

```
# hammer job-invocation create \
--job-template "template_name" \
--inputs key1="value",key2="value",... \
--search-query "query"
```

使用定义主机的过滤器表达式替换 *query*，如 "**name ~ rex01**"。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**。

10.16. 监控作业

您可以在作业运行时监控作业的进度。这有助于进行任何可能需要的故障排除。

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

流程

- 导航到 **Job** 页面。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。
- 在 **Job** 页面中，点 **Hosts** 选项卡。这将显示在其上运行作业的主机列表。
- 在 **Host** 列中，点您要检查的主机的名称。这将显示 **Commands** 页面的详情，您可以在其中实时监控作业执行。
- 随时点 **Back to Job** 以返回到 **Job Details** 页面。

CLI 过程

要在作业运行时监控作业的进度，请完成以下步骤：

1. 查找作业的 ID：

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

2. 监控作业输出：

```
# hammer job-invocation output \
--id job_ID \
--host host_name
```

3. 可选：要取消作业，请输入以下命令：

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

10.17. 同步模板存储库

在 Satellite 中，您可以同步 Satellite 服务器和版本控制系统或本地目录之间的作业模板、置备模板、报告模板和分区表模板的存储库。在本章中，Git 存储库用于演示目的。

本节详细介绍了以下工作流：

- 安装和配置 TemplateSync 插件
- 执行导出和导入任务

10.17.1. 启用 TemplateSync 插件

1. 要在 Satellite 服务器上启用插件，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。

10.17.2. 配置 TemplateSync 插件

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

表 10.1. 同步模板插件配置

参数	API 参数名称	导入的含义	导出的含义
关联	关联 接受的值： always、new、never	根据元数据将模板与操作系统、机构和位置相关联。	N/A

参数	API 参数名称	导入的含义	导出的含义
分支	分支	指定要从中读取的 Git 存储库中的默认分支。	指定要写入的 Git 存储库中的默认分支。
dirname	dirname	指定要从中读取的存储库下的子目录。	指定要写入的存储库下的子目录。
Filter	filter	仅导入名称与这个正则表达式匹配的模板。	仅导出名称与这个正则表达式匹配的模板。
强制导入	force	导入的模板会覆盖具有相同名称的锁定模板。	N/A
锁定模板	lock	当您导入具有相同名称的新模板时，不要覆盖现有模板，除非启用了 Force 导入。	N/A
元数据导出模式	metadata_export_mode 接受的值： 刷新、保留、删除	N/A	定义导出时如何处理元数据： <ul style="list-style-type: none"> ● refresh - 从模板内容中删除现有元数据，并根据当前的分配和属性生成新的元数据。 ● keep - 保留现有的元数据。 ● remove - 在没有元数据的情况下导出模板。如果要手动添加元数据，则很有用。
negate	negate 接受的值： true,false	导入忽略过滤器属性的模板。	导出忽略过滤器属性的模板。
prefix	prefix	如果模板名称没有以前缀开头，请在模板的开头添加指定字符串。	N/A
仓库	repo	定义要从中同步的存储库的路径。	定义要导出到的存储库的路径。
详细程度	详细 接受的值： true,false	启用将详细消息写入此操作的日志。	N/A

10.17.3. 导入和导出模板

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

10.17.3.1. 导入模板

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

先决条件

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

```
<%#
kind: provision
name: My Kickstart File
oses:
- RedHat 7
- RedHat 6
locations:
- First Location
- Second Location
organizations:
- Default Organization
- Extra Organization
%>
```

流程

- 在 Satellite Web UI 中，导航到 **Hosts > Sync Templates**。
- 点 **Import**。
- 每个字段都会填充在 **Administer > Settings > TemplateSync** 中配置的值。根据需要更改您要导入的模板的值。有关每个字段的详情请参考 [第 10.17.2 节“配置 TemplateSync 插件”](#)。
- 点 **Submit**。

Satellite Web UI 显示导入的状态。状态不是持久性的；如果您保留状态页面，则无法返回它。

CLI 过程

- 要从存储库导入模板，请输入以下命令：

```
$ hammer import-templates \
  --prefix '[Custom Index]' \
  --filter '*. Template Name' \
  --repo https://github.com/examplerepo/exampledirectory \
  --branch my_branch \
  --organization 'Default Organization'
```

要更好地索引和管理模板，请使用 `--prefix` 为您的模板设置类别。要从大型存储库中选择某些模板，请使用 `--filter` 来定义您要导入的模板的标题。例如，`-- filter 'DemoAnsible Default$'` 导入各种 Ansible 默认模板。

10.17.3.2. 导出模板

您可以将模板导出到版本控制系统，如 Git 存储库。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > Sync Templates**。
2. 单击 **Export**。
3. 每个字段都会填充在 **Administer > Settings > TemplateSync** 中配置的值。根据需要更改您要导出的模板的值。有关每个字段的详情请参考 [第 10.17.2 节“配置 TemplateSync 插件”](#)。
4. 点 **Submit**。

Satellite Web UI 显示导出的状态。状态不是持久性的；如果您保留状态页面，则无法返回它。

CLI 过程

1. 克隆 Git 存储库的本地副本：

```
$ git clone https://github.com/foreman/community-templates /custom/templates
```

2. 将本地目录的所有者改为 **foreman** 用户，并使用以下命令更改 SELinux 上下文：

```
# chown -R foreman:foreman /custom/templates
# chcon -R -t httpd_sys_rw_content_t /custom/templates
```

3. 要将模板导出到您的本地存储库，请输入以下命令：

```
hammer export-templates --organization 'Default Organization' --repo /custom/templates
```

在导出模板时，请避免类似 `/tmp` 或 `/var/tmp` 的临时目录，因为后端服务使用 `systemd` 私有临时目录运行。

10.17.3.3. 使用 Satellite API 同步模板

先决条件

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

```
<%#
kind: provision
name: My Kickstart File
oses:
- RedHat 7
- RedHat 6
locations:
- First Location
```

```
- Second Location
organizations:
- Default Organization
- Extra Organization
%>
```

流程

1. 配置使用 SSH 授权的版本控制系统，如 gitosis、gitolite 或 git 守护进程。
2. 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. 更改 Repo 设置以匹配 Git 存储库。例如，对于位于 `git@git.example.com/templates.git` 中的存储库，将设置设置为 `ssh://git@git.example.com/templates.git`。
3. 以 **foreman** 用户身份接受 Git SSH 主机密钥：

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

您可以看到 **Permission denied**，重试输出中的消息（这是预期的），因为 SSH 连接还无法成功。

4. 如果还没有 SSH 密钥对，请创建一个 SSH 密钥对。不要指定密码短语。

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

5. 使用 Satellite 中的公钥配置您的版本控制服务器，该密钥位于 `/usr/share/foreman/.ssh/id_rsa.pub` 中。
6. 将 Satellite Server 中的模板导出到 TemplateSync 菜单中指定的版本控制存储库：

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-u login:password \
-k https://_satellite.example.com/api/v2/templates/export \
-X POST

{"message":"Success"}
```

7. 在内容更改后，将模板导入到 Satellite 服务器：

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-u login:password \
-k https://_satellite.example.com/api/v2/templates/import \
-X POST

{"message":"Success"}
```

请注意，Satellite 提供的模板被锁定，默认情况下不导入它们。要覆盖此行为，请将 TemplateSync 菜单中的 **Force import** 设置改为 **yes**，或者在 import 命令中添加 **force** 参数 `-d '{"force": "true"}'`。

10.17.3.4. 使用 Satellite API 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存储库，请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 Kickstart File
oses:
- RedHat 7
- RedHat 6
locations:
- First Location
- Second Location
organizations:
- Default Organization
- Extra Organization
%>
```

流程

1. 创建存储模板的目录并应用适当的权限和 SELinux 上下文：

```
# mkdir -p /usr/share/templates_dir/
# chown foreman /usr/share/templates_dir/
# chcon -t httpd_sys_rw_content_t /usr/share/templates_dir/ -R
```

2. 更改 **TemplateSync** 选项卡中的 **Repo** 设置，使其与导出目录 `/usr/share/templates_dir/` 匹配。
3. 将 Satellite 服务器的模板导出到本地目录：

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-u login:password \
-k https://_satellite.example.com/api/v2/templates/export \
-X POST \

{"message":"Success"}
```

4. 在内容更改后，将模板导入到 Satellite 服务器：

```
$ curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" \
-u login:password \
-k https://_satellite.example.com/api/v2/templates/import \
-X POST

{"message":"Success"}
```


请注意，Satellite 提供的模板被锁定，默认情况下不导入它们。要覆盖此行为，请将 **TemplateSync** 菜单中的 **Force import** 设置改为 **yes**，或者在 `import` 命令中添加 **force** 参数 `-d '{"force": "true"}'`。



注意

您可以通过在请求中使用 `-d` 参数指定默认的 API 设置来覆盖默认的 API 设置。以下示例将模板导出到 `git.example.com/templates` 存储库：

```
$ curl -H "Accept:application/json,version=2" \
-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"}'
```

10.17.4. 高级 Git 配置

您可以使用命令行或编辑 `.gitconfig` 文件为 `TemplateSync` 插件执行额外的 Git 配置。

接受自签名 Git 证书

如果您在 Git 服务器上使用自签名证书身份验证，请使用 `git config http.sslCAPath` 命令验证证书。

例如，以下命令验证存储在 `/cert/cert.pem` 中的自签名证书：

```
# sudo -u foreman git config --global http.sslCAPath cert/cert.pem
```

有关高级选项的完整列表，请参阅 `git-config` 手册页。

10.17.5. 卸载插件

在删除 `foreman_templates` 插件后避免错误：

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
```

附录 A. 编写参考的模板

嵌入式 Ruby (ERB) 是一个根据模板生成文本文件的工具，用于将纯文本与 Ruby 代码合并。Red Hat Satellite 在以下情况下使用 ERB 语法：

置备模板

如需更多信息，请参阅 [置备指南中的创建置备模板](#)。

远程执行作业模板

更多信息请参阅 [第 10 章 配置和设置远程作业](#)。

报告模板

更多信息请参阅 [第 8 章 使用报告模板监控主机](#)。

分区表的模板

如需更多信息，请参阅 [置备指南中的创建分区表](#)。

智能变量

有关更多信息，请参阅 [Puppet 指南中的配置智能变量](#)。

智能类参数

有关更多信息，请参阅 [Puppet 指南中的配置智能类参数](#)。

本节概述了 ERB 模板中使用的特定于 Satellite 的宏和变量，以及一些用法示例。请注意，Red Hat Satellite 提供的默认模板(Host > Provisioning templates, Hosts > Job templates, Monitor > Report Templates)也提供了很好的 ERB 语法示例源。

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

A.1. 在 SATELLITE WEB UI 中访问编写参考的模板

您可以在 Satellite Web UI 中访问模板编写参考文档。

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

A.2. 编写 ERB 模板

以下标签是 ERB 模板中最重要的、常用的标签：

`<% %>`

所有 Ruby 代码都用 ERB 模板中的 `<it;% %>` 括起来。当模板呈现时，代码会被执行。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，此模板静默与服务执行操作，并在输出中返回任何内容。

```
<%= %>
```

这提供了与 `<% %>` 相同的功能，但在执行模板时，代码输出会插入到模板中。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，则不会返回输出。但是，如果您试图在不正确的变量上调用方法，则会返回以下出错信息：

输入示例：

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

渲染示例：

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

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

默认情况下，如果在行末尾关闭，则会在 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 `-%>` 修改保护标记：

输入示例：

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

渲染示例：

```
line1line2
```

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

有关如何在报告模板中使用该方法删除 FQDN 和 IP 地址之间的不必要的新行的示例：

输入示例：

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

渲染示例：

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

```
<%# %>
```

包含模板渲染过程中忽略的注释：

输入示例：

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

这不会生成任何输出。

在 ERB 模板中缩进

由于 ERB 标签的不同长度，处理 ERB 语法似乎可能看似意义。ERB 语法忽略空格。处理缩进的一种方法是在每个新行的开头声明 ERB 标签，然后使用 ERB 标签中的空格来概述语法中的关系，例如：

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

A.3. 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：

- **Directly in the template editor**– 当编辑一个模板 (在 **Hosts > Partition tables, Hosts > Provisioning templates** 下, 或 **Hosts > Job templates**), 在 **Template** 标签页中点 **Preview** 并从列表选择一个主机。然后，模板使用所选主机的参数在文本字段中显示。预览失败有助于识别模板中的问题。
- **在主机详情页面中**，在 **Hosts > All hosts** 中选择一个主机，然后点击 **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**，以查看它的呈现版本。

A.4. 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)	使用未缩进的代码片段模板时，将代码块用 n 空格来缩进。
foreman_url(kind)	将完整 URL 返回给给定 kind 的主机渲染模板。例如，"provision"类型的模板通常位于 http://HOST/unattended/provision 。
snippet(name)	呈现指定的代码片段模板。对于嵌套置备模板非常有用。
snippets(file)	呈现 Foreman 数据库中找到的指定片段，如果数据库中没有找到，尝试从 <code>无人值守/snippets/</code> 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.5. 模板 MACROS

如果要编写自定义模板，您可以使用以下几个宏。

根据模板类型，以下部分宏有不同的要求。

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

如需有关作业模板的可用宏的更多信息，在 Satellite Web UI 中，导航到 **Hosts > Job Templates**，然后点 **New Job Template**。在 New Job Template 窗口中，点 **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、类型和可供用户使用的选项。对于报告模板，您只能使用用户输入。当您定义新输入并保存模板时，您可以在模板正文的 ERB 语法中引用输入。

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

这会从用户输入 **cpus** 加载值。

load_hosts

使用 **load_hosts** 宏，您可以生成主机的完整列表。

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

使用带有 **each_record** 宏的 **load_hosts** 宏来批量加载 1000 个记录，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 -%>
```

在本例中，您首先创建一个输入，然后用来重新定义 **load_hosts** 宏检索的搜索条件。

report_row

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

输入示例：

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

渲染示例：

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

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

输入示例：

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

渲染示例：

```
Server FQDN,IP
host1.example.com,10.8.30.228
host2.example.com,10.8.30.227
```

```
host3.example.com,10.8.30.226
host4.example.com,10.8.30.225
host5.example.com,10.8.30.224
host6.example.com,10.8.30.223
```

report_render

此宏仅适用于报告模板。

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

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

render_template()

此宏仅适用于作业模板。

使用这个宏，您可以呈现特定模板。您还可以启用和定义要传递给模板的参数。

A.6. 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.9 节“解析阵列” 。
@host.capabilities	系统置备的方法可以是构建（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	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	返回主机的引导装载程序密码。
@host.hostgroup	主机的主机组。

Name	描述
host_enc['parameters']	返回包含主机参数信息的 Ruby 哈希。例如，使用 <code>host_enc['parameters']['lifecycle_environment']</code> 获取主机的生命周期阶段。
@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.9 节“解析阵列” 。
@host.interfaces_with_identifier('IDs')	返回具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传递，如 <code>@host.interfaces_with_identifier(['eth0', 'eth1'])</code> 。请参阅 第 A.9 节“解析阵列” 。
@host.ip	主机的 IP 地址。
@host.location	主机的位置。
@host.mac	主机的 MAC 地址。
@host.managed_interfaces	返回一组受管接口（不包括 BMC 和绑定接口）。请参阅 第 A.9 节“解析阵列” 。
@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)	到内核和 initrd 的完整路径，返回数组。
@host.os.medium_uri(@host)	用于置备的 URI（在安装介质中配置的路径）。
host_param('parameter_name')	返回指定 host 参数的值。
host_param_false?('parameter_name')	如果指定的 host 参数评估为 false ，则返回 false。

Name	描述
host_param_true?('parameter_name')	如果指定的 host 参数评估为 true ，则返回 true。
@host.primary_interface	返回主机的主接口。
@host.provider	计算资源提供程序。
@host.provision_interface	返回主机的调配接口。返回接口对象。
@host.phtable	分区表名称。
@host.puppet_ca_server	主机必须使用的 Puppet CA 服务器。
@host.puppetmaster	主机必须使用的 Puppet master。
@host.pxe_build?	如果主机使用网络或 PXE 置备，则返回 true 。
@host.shortname	主机的短名称。
@host.sp_ip	BMC 接口的 IP 地址。
@host.sp_mac	BMC 接口的 MAC 地址。
@host.sp_name	BMC 接口的名称。
@host.sp_subnet	BMC 网络的子网。
@host.subnet.dhcp	如果为这个主机配置了 DHCP 代理，则返回 true 。
@host.subnet.dns_primary	主机的主 DNS 服务器。
@host.subnet.dns_secondary	主机的辅助 DNS 服务器。
@host.subnet.gateway	主机的网关。
@host.subnet.mask	主机的子网掩码。
@host.url_for_boot(:initrd)	与此主机关联的 initrd 镜像的完整路径。不建议，因为它不会反过变量。
@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	等同于 'host' 或 'hostgroup'，具体取决于调配类型。
@static	如果网络配置是静态的，则返回 true 。

Name	描述
@template_name	正在呈现的模板的名称。
grub_pass	返回一个 bootloader 参数，以设置加密的引导装载程序密码，如 <code>--md5pass=114{@host.grub_pass}</code> 。
ks_console	使用端口和可添加到内核行的主机的 baud 速率返回字符串。例如 <code>console=ttyS1, 9600</code> 。
root_pass	返回为系统配置的根密码。

大多数常见 Ruby 方法都可以应用到特定于主机的变量。例如，要提取主机 IP 地址的最后片段，您可以使用：

```
<% @host.ip.split('.').last %>
```

A.7. 特定于 KICKSTART 的变量

以下变量设计为在 kickstart 置备模板中使用。

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（在表的第一行有一个 #Dynamic 选择），则返回 true 。
@epel	一个命令，该命令将自动安装正确的 epel-release rpm 版本。在 %post 脚本中使用。
@mediapath	提供 URL 命令的完整 kickstart 行。
@osver	操作系统主版本号，与 @host.operatingsystem.major 相同。

A.8. 条件声明

在模板中，您可以根据存在的值来执行不同的操作。要达到此目的，您可以使用 ERB 语法中的条件语句。

在以下示例中，ERB 语法搜索特定主机名并根据它找到的值返回输出：

输入示例：

```
<% load_hosts().each_record do |host| -%>
<% if @host.name == "host1.example.com" -%>
```

```
<% result="positive" -%>
<% else -%>
<% result="negative" -%>
<% end -%>
<%= result -%>
```

渲染示例：

```
host1.example.com
positive
```

A.9. 解析阵列

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**，返回分组中的接口数据。要提取特定接口的参数值，请使用 Ruby 方法解析阵列。

查找修复方法解析阵列

以下流程是一个示例，您可以用来查找模板中解析阵列的相关方法。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，在本例中，使用 **@host.interfaces** 变量返回类值，然后您可以使用查找方法来解析阵列。

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 **member** 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 **member** 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，请将相关方法添加到 ERB 语法中：

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.10. 模板片断示例

检查主机是否启用了 Puppet 和 Puppet

以下示例检查主机是否启用了 Puppet 和 Puppet the 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

以下示例演示了如何捕获主机操作系统的次版本和主版本，它们可用于打包相关的决策：

```
<%
os_major = @host.operatingsystem.major.to_i
os_minor = @host.operatingsystem.minor.to_i
%>

<% if ((os_minor < 2) && (os_major < 14)) -%>
...
<% end -%>
```

将片断导入到模板

以下示例将 `subscription_manager_registration` 代码片段导入到模板，并以四个空格缩进：

```
<%= indent 4 do
snippet 'subscription_manager_registration'
end %>
```

条件地导入 Kickstart 片断

如果主机的子网启用了 DHCP 引导模式，以下示例导入 `kickstart_networking_setup` 片断：

```
<% 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` 变量来解析主机事实和自定义事实中的值。

在本例中，`luks_stat` 是一个自定义事实，您可以使用与 `dmi::system::serial_number` 相同的方式进行解析，这是一个主机事实：

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

在本例中，您可以自定义 Applicable Errata 报告模板，以解析有关每个主机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

使用本节作为帮助修改、自定义和扩展作业模板以满足您的要求的引用。

B.1. 自定义作业模板

在创建作业模板时，您可以在模板编辑器字段中包含现有模板。这样，您可以组合模板，或者从常规模板中创建更具体的模板。

以下模板组合了默认模板，以便在 Red Hat Enterprise Linux 系统上安装和启动 **httpd** 服务：

```
<%= render_template 'Package Action - SSH Default', :action => 'install', :package => 'httpd' %>
<%= render_template 'Service Action - SSH Default', :action => 'start', :service_name => 'httpd' %>
```

以上模板直接为呈现的模板指定参数值。也可以使用 **input** () 方法在作业执行上定义呈现模板的输入。例如，您可以使用以下语法：

```
<%= render_template 'Package Action - SSH Default', :action => 'install', :package =>
input("package") %>
```

使用上述模板，您必须从呈现的模板导入参数定义。为此，请导航到 **Jobs** 选项卡，单击 **Add Foreign Input Set**，然后从 **Target template** 列表中选择呈现的模板。您可以导入所有参数，或指定以逗号分隔的列表。

B.2. 默认任务模板类别

作业模板类别	Description
软件包	用于执行软件包相关操作的模板。默认包括安装、更新和删除操作。
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。默认情况下，会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	用于执行与服务相关的操作的模板。start, stop, restart, 和 status 操作会被默认包括。
Katello	用于执行与内容相关的操作的模板。这些模板主要从 Satellite Web UI 的不同部分使用（例如，用于内容主机的批量操作 UI），但可用于执行勘误表安装等操作。

B.3. RESTORECON 模板示例

本例演示了如何创建名为 **Run Command - restorecon** 的模板，该模板为目标主机上所选目录中的所有文件恢复默认 SELinux 上下文。

1. 进入 **Hosts > Job templates**。单击 **New Job Template**。

- 在 **Name** 字段中输入 **Run Command - restorecon**。选择 **Default** 使模板可供所有组织使用。在模板编辑器中添加以下文本：

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

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

- 在 **Job** 选项卡中，将 **Job category** 设置为 **Commands**。
- 单击 **Add Input** 以允许自定义作业。在 **Name** 字段中输入 **directory**。输入名称必须与模板编辑器中指定的值匹配。
- 单击 **Required**，以便在没有指定用户参数的情况下无法执行命令。
- 从 **Input type** 列表中选择 **User input**。输入在作业调用期间要显示的描述，如 **Target directory for restorecon**。
- 点 **Submit**。

有关如何 [基于此模板执行任务的信息](#)，请参阅在多个主机上执行 **restorecon** 模板。

B.4. 呈现 RESTORECON 模板

本例演示了如何创建从 **Run 命令派生的模板 - restorecon** 模板在 [Example restorecon Template](#) 中创建。此模板不需要用户在作业执行时输入，它将在目标主机上的 **/home/** 目录下的所有文件中恢复 SELinux 上下文。

按照 [设置作业模板](#) 中所述创建新模板，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

本例演示了如何根据在多个主机上 [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。

- 导航到 **Hosts > All hosts** 并选择目标主机。从 **Select Action** 列表中选择 **Schedule Remote Job**。
- 在 **作业调用** 页面中，选择 **Commands** job category 和 **Run Command - restorecon** job template。
- 在 **directory** 字段中输入 **/home**。
- 将 **Schedule** 设置为 **Execute now**。
- 点 **Submit**。您进入 **Job 调用** 页面，您可以在其中监控作业执行的状态。

B.6. 在模板中包括电源操作

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 在重启时解释断开连接的异常，因此任务的远程执行可以正常工作。

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


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