



Red Hat Satellite 6.9

置备指南

在 Red Hat Satellite 服务器上配置物理和虚拟主机的指南。

Red Hat Satellite 6.9 置备指南

在 Red Hat Satellite 服务器上配置物理和虚拟主机的指南。

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Provisioning_Guide.ent file |.

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

红帽卫星调配指南包含调配物理和虚拟主机上的说明。这包括设置所需的网络拓扑、配置必要的服务，并提供所有其他配置信息以在网络上置备主机。

目录

第1章 简介	5
1.1. 置备概述	5
1.2. 网络引导置备 workflow	5
第2章 配置置备资源	7
2.1. 置备上下文	7
2.2. 设置置备上下文	7
2.3. 创建操作系统	7
2.4. 更新多个操作系统的详情	8
2.5. 创建架构	9
2.6. 创建硬件模型	9
2.7. 为主机的操作系统使用同步的 KICKSTART 存储库	10
2.8. 在 SATELLITE 中添加安装介质	10
2.9. 创建分区表	11
2.10. 动态分区示例	12
2.11. 调配模板	13
2.12. 调配模板的类型	13
2.13. 创建调配模板	14
2.14. 克隆调配模板	15
2.15. 创建计算配置文件	15
2.16. 为主机设置默认加密根密码	16
2.17. 使用 NOVNC 访问虚拟机	16
第3章 配置网络	18
3.1. 网络资源	18
3.2. SATELLITE 和 DHCP 选项	19
3.3. 卫星中的 DHCP 问题故障排除	20
3.4. 基于镜像的置备的先决条件	21
3.5. 配置网络服务	22
3.5.1. 通过安装程序进行多个子网或域	23
3.5.2. 用于网络配置的 DHCP、DNS 和 TFTP 选项	24
3.5.3. 通过 NAT 使用 TFTP 服务	26
3.6. 将域添加到卫星服务器	27
3.7. 将子网添加到卫星服务器	28
第4章 使用 INFOBLOX 作为 DHCP 和 DNS 供应商	31
4.1. 限制	31
4.2. 前提条件	31
4.3. 在胶囊服务器上安装 INFOBLOX CA 证书	31
4.4. 安装 DHCP INFOBLOX 模块	32
4.5. 安装 DNS INFOBLOX 模块	33
第5章 将 IPXE 配置为 REDUCE PROVISIONING TIMES	35
5.1. 为 IPXE 设置发现服务	37
5.2. 链引导虚拟机	37
5.3. CHAINBOOTING SATELLITE SERVER 以直接使用 IPXE	40
5.4. CHAINBOOTING RED HAT SATELLITE CAPSULE 以直接使用 IPXE	44
第6章 使用 PXE 调配主机	45
6.1. 裸机置备先决条件	46
6.2. 配置安全令牌有效期持续时间	46
6.3. 使用未置备置备创建主机	47

6.4. 使用无 PXE 的置备创建主机	49
6.5. 使用 UEFI HTTP 引导置备创建主机	52
6.6. 在置备过程中部署 SSH 密钥	56
第 7 章 配置发现服务	58
7.1. 安装 DISCOVERY 服务	59
7.2. PROVISIONING TEMPLATE PXELINUX DISCOVERY SNIPPET	60
7.3. DISCOVERED HOSTS 的自动上下文	61
7.4. 发现模板和 SNIPPETS 设置	61
7.5. 从发现的主机创建主机	63
7.6. 创建发现规则	65
7.7. 实施 PXE 无发现	67
7.8. 构建发现镜像	69
7.9. 无人值守的 USE、CUSTOMIZATION 和 IMAGE REMASTERING	72
7.10. 扩展发现镜像	74
7.11. 发现故障排除	75
第 8 章 使用 RED HAT IMAGE BUILDER IMAGE 进行置备	78
第 9 章 在 KVM (LIBVIRT)上置备虚拟机	80
9.1. 为 KVM 连接配置 SATELLITE 服务器	82
9.2. 为 SATELLITE 服务器添加 KVM 连接	83
9.3. 在 SATELLITE 服务器中添加 KVM 镜像	84
9.4. 在计算配置集中添加 KVM 详情	85
9.5. 在 KVM 上创建主机	86
第 10 章 在 RED HAT VIRTUALIZATION 上置备虚拟机	90
10.1. 将 RED HAT VIRTUALIZATION 连接添加到卫星服务器	92
10.2. 在 RED HAT VIRTUALIZATION 中准备 CLOUD-INIT 镜像	93
10.3. 在 SATELLITE 服务器中添加 RED HAT VIRTUALIZATION 镜像	94
10.4. 准备 CLOUD-INIT 模板	95
10.5. 在计算配置集中添加 RED HAT VIRTUALIZATION 详情	97
10.6. 在 RED HAT VIRTUALIZATION 上创建主机	99
第 11 章 在 VMWARE VSPHERE 中置备虚拟机	103
11.1. VMWARE VSPHERE PROVISIONING 的先决条件	103
11.2. 创建 VMWARE VSPHERE 用户	103
11.3. 为 SATELLITE 服务器添加 VMWARE VSPHERE 连接	104
11.4. 在 SATELLITE 服务器中添加 VMWARE VSPHERE 镜像	106
11.5. 在 COMPUTE PROFILE 中添加 VMWARE VSPHERE DETAILS	107
11.6. 在 VMWARE VSPHERE 服务器上创建主机	109
11.7. 使用 VMWARE VSPHERE CLOUD-INIT 和 USERDATA 模板进行置备	112
11.8. 计算资源缓存	117
11.8.1. 启用计算资源的缓存	117
11.8.2. 刷新计算资源缓存	117
第 12 章 在容器原生虚拟化中置备虚拟机	119
12.1. 将容器原生虚拟化连接添加到卫星服务器	120
第 13 章 PROVISIONING CLOUD INSTANCES ON RED HAT OPENSTACK PLATFORM	123
13.1. 将 RED HAT OPENSTACK PLATFORM 连接添加到卫星服务器	123
13.2. 将 RED HAT OPENSTACK PLATFORM 镜像添加到卫星服务器	125
13.3. 在计算配置集中添加 RED HAT OPENSTACK PLATFORM 详情	126
13.4. CREATING IMAGE-BASED HOSTS ON RED HAT OPENSTACK PLATFORM	127

第 14 章 在 AMAZON EC2 中置备云实例	130
14.1. AMAZON EC2 置备先决条件	130
14.2. 将 AMAZON EC2 连接添加到卫星服务器	130
14.3. 将 HTTP 代理用于计算资源	132
14.4. 将 AMAZON EC2 镜像添加到卫星服务器	133
14.5. 在计算配置文件中添加 AMAZON EC2 详情	134
14.6. 在 AMAZON EC2 上创建基于镜像的主机	135
14.7. 使用 SSH 连接到 AMAZON EC2 实例	136
14.8. 为 AMAZON WEB SERVICE EC2 环境配置完成模板	137
14.9. 有关 AMAZON WEB SERVICES 和 SATELLITE 的更多信息	139
第 15 章 在 GOOGLE COMPUTE ENGINE 上置备云实例	140
15.1. 在 SATELLITE 服务器中添加 GOOGLE COMPUTE ENGINE 连接	141
15.2. 在 SATELLITE 服务器中添加 GOOGLE COMPUTE ENGINE 镜像	142
15.3. 在 COMPUTE PROFILE 中添加 GOOGLE COMPUTE ENGINE DETAILS	144
15.4. 在 GOOGLE COMPUTE ENGINE 上创建基于镜像的主机	145
第 16 章 在 MICROSOFT AZURE RESOURCE MANAGER 上置备云实例	148
16.1. 向 SATELLITE 服务器添加 MICROSOFT AZURE RESOURCE MANAGER 连接	149
16.2. 将 MICROSOFT AZURE 资源管理器镜像添加到 SATELLITE 服务器	150
16.3. 在 COMPUTE PROFILE 中添加 MICROSOFT AZURE RESOURCE MANAGER DETAILS	152
16.4. 在 MICROSOFT AZURE RESOURCE MANAGER 上创建基于镜像的主机	154
附录 A. 调配示例的初始化脚本	157
附录 B. 置备 FIPS-COMPLIANT HOSTS	159
B.1. 更改 PROVISIONING PASSWORD HASHING ALGORITHM	159
B.2. 设置 FIPS-ENABLED PARAMETER	160
B.3. 验证 FIPS 模式是否已启用	160
附录 C. 为 RED HAT SATELLITE 构建云镜像	161
C.1. 创建自定义 RED HAT ENTERPRISE LINUX 镜像	161
C.2. 创建 RED HAT ENTERPRISE LINUX 7 镜像	162
C.3. 创建 RED HAT ENTERPRISE LINUX 6 镜像	165
C.4. 配置主机以进行注册	166
C.5. 注册主机	168
C.6. 安装 KATELLO AGENT	169
C.7. 安装 PUPPET 代理	170
C.8. 完成 RED HAT ENTERPRISE LINUX 7 镜像	170
C.9. 完成 RED HAT ENTERPRISE LINUX 6 镜像	172
C.10. 后续步骤	173

第 1 章 简介

1.1. 置备概述

调配是一个以裸机或虚拟机开始的进程，结尾是完全配置的、可直接使用的操作系统。使用红帽卫星，您可以为大量主机定义和自动化精细配置。

有很多调配方法。例如，您可以使用卫星服务器的集成胶囊或外部胶囊服务器来利用基于 PXE 和非 PXE 的方法调配裸机主机。您还可以通过其 API 从特定供应商置备云实例。这些配置方法是 Red Hat Satellite 应用程序生命周期的一部分，用于创建、管理和更新主机。

Red Hat Satellite 有不同的置备主机方法：

裸机置备

卫星主要通过 PXE 引导和 MAC 地址识别来调配裸机主机。您可以创建主机条目，并指定要置备的物理主机的 MAC 地址。您还可以引导空白主机以使用卫星的发现服务，它会创建随时可用的主机池。您还可以通过无 PXE 的方法引导和置备主机。

云供应商

卫星连接到私有和公共云提供商，从存储在云环境中的镜像调配主机实例。这也包括选择要使用的硬件配置集或类别。

虚拟化基础架构

卫星连接到虚拟化基础架构服务，如红帽虚拟化和 VMware，以便从虚拟机模板置备虚拟机，或使用与裸机供应商相同的基于 PXE 的引导方法。

1.2. 网络引导置备 workflow

PXE 引导假定主机（物理或虚拟）配置为从网络引导作为第一个引导设备，从硬盘驱动器作为第二个引导设备启动。

置备过程遵循基本的 PXE workflow：

1. 您可以创建一个主机并选择一个域和子网。卫星从 DHCP 胶囊服务器请求与子网或卫星中的 PostgreSQL 数据库关联的可用 IP 地址。卫星将此 IP 地址加载到 Create Host 窗口中的 **IP 地址** 字段。完成新主机的所有选项后，提交新主机请求。
2. 根据主机及其域和子网的配置规格，卫星会创建以下设置：
 - 胶囊服务器上与子网关联的 DHCP 记录。
 - 胶囊服务器上与域关联的转发 DNS 记录。
 - 与子网关联的 DNS 胶囊服务器上的反向 DNS 记录。
 - 与子网关联的 TFTP 胶囊服务器中主机的 PXELinux、Grub、Grub2 和 iPXE 配置文件。
 - 关联的 Puppet 服务器上的 Puppet 证书。
 - 关联的身份服务器上的一个域。
3. 新主机从 DHCP 服务器请求 DHCP 保留。
4. DHCP 服务器响应保留请求，并返回 TFTP **next-server** 和 **filename** 选项。
5. 主机根据 PXELoader 设置从 TFTP 服务器请求引导装载程序和菜单。

6. 引导装载程序通过 TFTP 返回。
7. 引导装载程序通过其置备接口 MAC 地址获取主机的配置。
8. 引导装载程序会获取操作系统安装程序内核、init RAM 磁盘和引导参数。
9. 安装程序从 Satellite 请求调配模板。
10. 卫星呈现调配模板并将结果返回到主机。
11. 安装程序执行操作系统的安装。
 - 安装程序使用 Red Hat Subscription Manager 将主机注册到 Satellite。
 - 安装程序安装管理工具，如 **katello-agent** 和 **puppet**。
 - 安装程序 会在安装后脚本中通知卫星成功构建。
12. PXE 配置文件恢复到本地引导模板。
13. 主机重新启动。
14. 新主机从 DHCP 服务器请求 DHCP 保留。
15. DHCP 服务器响应保留请求，并返回 TFTP **next-server** 和 **filename** 选项。
16. 主机根据 PXELoader 设置从 TFTP 服务器请求引导加载程序和菜单。
17. 引导装载程序通过 TFTP 返回。
18. 引导装载程序通过其置备接口 MAC 地址获取主机的配置。
19. 引导装载程序从本地驱动器启动。
20. 如果将主机配置为使用任何 Puppet 类，则主机使用模块来配置自身。

此工作流程根据自定义选项的不同而有所不同。例如：

Discovery (发现)

如果使用发现服务，卫星将自动检测新主机的 MAC 地址并在提交请求后重新启动主机。请注意，TCP 端口 8443 必须可以被附加到该主机的胶囊访问，以重新启动主机。

无 PXE 的置备

提交新的主机请求后，您必须使用从 Satellite 下载的引导磁盘引导特定的主机，并使用主机的 USB 端口进行传输。

计算资源

卫星创建虚拟机并检索 MAC 地址，并将 MAC 地址存储在卫星中。如果使用基于镜像的置备，则主机不会遵循标准 PXE 引导和操作系统安装。计算资源创建镜像的副本，供主机使用。根据 Satellite 中的镜像设置，查看的数据可以在 中进行传递，例如使用 **cloud-init**。卫星可以使用 SSH 连接到主机，并执行模板来完成自定义。

第 2 章 配置置备资源

2.1. 置备上下文

调配上下文是您为 Satellite 组件指定的组织和位置的组合。组件所属的组织和位置设定该组件的所有权和权限。

组织根据所有权、目的、内容、安全级别和其他部门将红帽卫星 6 组件划分为逻辑组。您可以通过 Red Hat Satellite 6 创建和管理多个组织，并为每个单独的组织分配组件。这可确保卫星服务器在特定组织内调配主机，并且仅使用分配给该组织的组件。有关机构的更多信息，[请参阅 内容管理指南中的管理机构](#)。

位置功能与组织类似。区别在于，位置基于物理或地理设置。用户可以将位置嵌套在层次结构中。有关位置的更多信息，[请参阅 内容管理指南中的管理位置](#)。

2.2. 设置置备上下文

设置调配上下文时，您可以定义要用于置备主机的组织和位置。

组织和位置菜单位于卫星 Web UI 左上角的菜单栏中）。如果您尚未选择要使用的组织和位置，则菜单将显示：**Any Organization** 和 **Any Location**。

流程

1. 单击 **Any Organization**，再选择该组织。
2. 单击 **Any Location**，再选择要使用的位置。

每个用户都可在其帐户设置中设置默认的调配上下文。单击卫星 Web UI 右上角的用户名，再选择 **My account** 以编辑用户帐户设置。

对于 CLI 用户

- 使用 CLI 时，将 **--organization** 或 **--organization-label** 和 **--location** 或 **--location-id** 作为一个选项。例如：

```
# hammer host list --organization "Default_Organization" --location "Default_Location"
```

此命令输出为 Default_Organization 和 Default_Location 分配的主机。

2.3. 创建操作系统

操作系统是资源的集合，用于定义卫星服务器如何在主机上安装基础操作系统。操作系统条目将之前定义的资源（如安装介质、分区表、调配模板和其他）组合在一起。

从红帽 CDN 导入操作系统在 **Hosts > Operating Systems** 页面中创建新的条目。

您还可以使用以下步骤添加自定义操作系统：

流程

1. 在 Satellite Web UI 中，进入 **Hosts > Operating systems** 并点 **New Operating system**。
2. 在 **Name** 字段中输入代表操作系统条目的名称。

3. 在 **Major** 字段中，输入与操作系统主要版本对应的数字。
4. 在 **Minor** 字段中，输入与操作系统的次版本对应的数字。
5. 在 **Description** 字段中，输入操作系统的描述。
6. 从 **Family** 列表中，选择操作系统系列。
7. 从 **Root Password Hash** 列表中，选择 root 密码的编码方法。
8. 从 **架构** 列表中，选择操作系统所使用的架构。
9. 点 **Partition table** 选项卡，然后选择适用于此操作系统的可能的分区表。
10. 可选：如果您使用非红帽内容，点安装介质选项卡并选择应用于此操作系统的安装介质。更多信息请参阅 [第 2.8 节“在 Satellite 中添加安装介质”](#)。
11. 单击 **Templates** 选项卡，再选择 **PXELinux 模板**、**Provisioning 模板**，以及您的操作系统要使用的 **Finish 模板**。如果计划使用 **iPXE** 进行配置，可以选择其他模板（如 **iPXE 模板**）。
12. 单击 **Submit** 以保存您的调配模板。

对于 CLI 用户

- 使用 **hammer os create** 命令创建操作系统：

```
# hammer os create --name "MyOS" \
--description "My_custom_operating_system" \
--major 7 --minor 3 --family "Redhat" --architectures "x86_64" \
--partition-tables "My_Partition" --media "Red_Hat" \
--provisioning-templates "My_Provisioning_Template"
```

2.4. 更新多个操作系统的详情

使用这个流程更新多个操作系统的详情。本例演示了如何为每个操作系统分配一个名为 **Kickstart default** 的分区表，以及名为 **Kickstart default PXELinux** 的配置模板，以及名为 **Kickstart Default** 的调配模板。

流程

1. 在卫星服务器上，运行以下 Bash 脚本：

```
PARTID=$(hammer --csv partition-table list | grep "Kickstart default," | cut -d, -f1)
PXEID=$(hammer --csv template list --per-page=1000 | grep "Kickstart default PXELinux" |
cut -d, -f1)
SATID=$(hammer --csv template list --per-page=1000 | grep "provision" | grep ",Kickstart
default" | cut -d, -f1)

for i in $(hammer --no-headers --csv os list | awk -F, {'print $1'})
do
  hammer partition-table add-operatingsystem --id="{PARTID}" --operatingsystem-id="{i}"
  hammer template add-operatingsystem --id="{PXEID}" --operatingsystem-id="{i}"
  hammer os set-default-template --id="{i}" --config-template-id="{PXEID}"
```

```
hammer os add-config-template --id="{i}" --config-template-id=${SATID}
hammer os set-default-template --id="{i}" --config-template-id=${SATID}
done
```

2. 显示更新的操作系统信息，以验证操作系统是否已正确更新：

```
# hammer os info --id 1
```

2.5. 创建架构

Satellite 中的架构代表主机和操作系统的逻辑分组。使用 Puppet 中的主机签入时，卫星会自动创建架构。基本的 i386 和 x86_64 架构已在卫星中预先设置。

使用此流程在 Satellite 中创建架构。

支持的构架

仅支持使用 PXE、发现和引导磁盘调配的 Intel x86_64 架构。如需更多信息，请参阅 [Satellite 6 中的红帽知识库解决方案支持的架构和置备场景](#)。

流程

1. 在 Satellite Web UI 中，进入 **Hosts > Architectures** 并点 **Create Architecture**。
2. 在 **Name** 字段中输入架构的名称。
3. 从 **Operating Systems** 列表中，选择一个操作系统。如果没有可用，您可以在 **Hosts > Operating Systems** 中创建并分配它们。
4. 点 **Submit**。

对于 CLI 用户

- 输入 **hammer 架构 create** 命令来创建架构。指定包含此架构的名称和操作系统：

```
# hammer architecture create --name "Architecture_Name" \
--operatingsystems "os"
```

2.6. 创建硬件模型

使用这个步骤在卫星中创建硬件模型，以便您可以指定主机使用的硬件模型。

流程

1. 在 Satellite Web UI 中，进入 **Hosts > Hardware Models** 并点 **Create Model**。
2. 在 **Name** 字段中输入硬件模型的名称。
3. 另外，在 **Hardware Model** 和 **Vendor Class** 字段中，您可以为您的系统输入相应的信息。
4. 在 **Info** 字段中输入硬件模型的描述。
5. 点 **Submit** 保存您的硬件模型。

对于 CLI 用户

- 使用 **hammer 模型 create** 命令创建硬件模型。唯一必需参数是 **--name**。另外，还可使用 **--hardware-model** 选项输入硬件模型、带有 **--vendor-class** 选项的供应商类，以及 **--info** 选项的描述：

```
# hammer model create --name "model_name" --info "description" \
--hardware-model "hardware_model" --vendor-class "vendor_class"
```

2.7. 为主机的操作系统使用同步的 KICKSTART 存储库

Satellite 包含一组用来安装调配的主机操作系统的同步 kickstart 存储库。有关添加存储库的更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。

使用这个流程设置 kickstart 软件仓库。

流程

1. 添加您要用于现有内容视图的同步 kickstart 存储库，或创建新的内容视图并添加 kickstart 存储库。
对于 Red Hat Enterprise Linux 8，请确定您同时添加了 **Red Hat Enterprise Linux 8 for x86_64 - AppStream Kickstart x86_64 8** 和 **Red Hat Enterprise Linux 8 for x86_64 - BaseOS Kickstart x86_64 8** 软件仓库。

如果您使用断开连接的环境，则必须从 Red Hat Enterprise Linux 二进制 DVD 中导入 Kickstart 软件仓库。有关更多信息，请参阅 [内容管理指南中的导入 Kickstart 存储库](#)。

2. 发布新版本的 Content View，在其中添加 Kickstart 存储库并将其提升到所需的生命周期环境。如需更多信息，请参阅 [内容管理指南中的管理内容视图](#)。
3. 在创建主机时，在“操作系统”选项卡中的“操作系统”选项卡中，选择“同步内容”复选框。

要查看 kickstart 树，请输入以下命令：

```
# hammer medium list --organization "your_organization"
```

2.8. 在 SATELLITE 中添加安装介质

安装介质是卫星服务器用于从外部存储库在机器上安装基础操作系统的来源。您可以使用此参数来安装第三方内容。红帽内容通过存储库同步提供。

安装介质必须采用操作系统安装树的格式，且必须可以通过 HTTP URL 访问托管安装程序的机器。您可以通过导航到 [Hosts > Installation Media](#) 菜单来查看安装介质。

默认情况下，卫星包含一些官方 Linux 发行版本的安装介质。请注意，其中有些安装介质是针对操作系统的特定版本的目标。例如，对于 CentOS 7 或更早的版本必须使用 [CentOS 镜像\(7.x\)](#) 示例，并且 [CentOS 镜像\(8.x\)](#) 必须用于 CentOS 8 或更高版本。

如果要使用安装介质在多个主机上安装操作系统时，如果要提高下载性能，您必须修改安装介质的路径以指向最接近的镜像或本地副本。

流程

1. 在 Satellite Web UI 中，进入 [Hosts > Installation Media](#) 并点 [Create Medium](#)。

2. 在 Name 字段中输入代表安装介质条目的名称。
3. 在 Path 中输入包含安装树的 URL 或 NFS 共享。您可以在路径中使用以下变量来代表多个不同的系统架构和版本：
 - `$arch` - 系统架构。
 - `$version` - 操作系统版本。
 - `$major` - 操作系统主要版本。
 - `$minor` - 操作系统次版本。

HTTP 路径示例：

```
http://download.example.com/centos/$version/Server/$arch/os/
```

NFS 路径示例：

```
nfs://download.example.com:/centos/$version/Server/$arch/os/
```

胶囊服务器上的同步内容始终使用 HTTP 路径。胶囊服务器管理的内容不支持 NFS 路径。

4. 从 Operating system family 列表中，选择安装介质的分发或系列。例如，CentOS 和 Fedora 位于红帽系列中。
5. 单击 Organizations 和 Locations 选项卡，以更改调配上下文。卫星服务器将安装介质添加到设置调配上下文中。
6. 点 Submit 保存您的安装介质。

对于 CLI 用户

- 使用 `hammer medium create` 命令创建安装介质：

```
# hammer medium create --name "CustomOS" --os-family "Redhat" \
--path 'http://download.example.com/centos/$version/Server/$arch/os/' \
--organizations "My_Organization" --locations "My_Location"
```

2.9. 创建分区表

分区表是一个模板类型，用于定义卫星服务器在新主机上可用磁盘的方式。分区表使用与调配模板相同的 ERB 语法。Red Hat Satellite 包含一组要使用的默认分区表，包括 Kickstart 默认。您还可以编辑分区表条目来配置首选分区方案，或者创建分区表条目并将其添加到操作系统条目中。

流程

1. 在 Satellite Web UI 中，进入 Hosts > Partition Tables 并点 Create Partition Table。
2. 在 Name 字段中输入分区表的名称。
3. 如果要模板设置为自动与新机构或位置关联，请选择 Default 复选框。
4. 如果要模板标识为其他分区表的可重用片断，请选择 Snippet 复选框。

5. 从 Operating System Family 列表中，选择分区布局的分发或系列。例如，Red Hat Enterprise Linux、CentOS 和 Fedora 是红帽系列的。
6. 在 Template 编辑器 字段中，输入磁盘分区的布局。例如：

```
zerombr
clearpart --all --initlabel
autopart
```

您还可以使用模板文件 浏览器 上传模板文件。

布局的格式必须与预期操作系统匹配。例如，Red Hat Enterprise Linux 7.2 需要与 kickstart 文件匹配的布局。

7. 在 Audit Comment 字段中，在分区布局中添加更改概述。
8. 单击 组织和位置 选项卡，以添加您要与分区表关联的任何其他调配上下文。卫星将分区表添加到当前的调配上下文中。
9. 点 Submit 保存您的分区表。

对于 CLI 用户

1. 在使用 CLI 创建分区表之前，先创建一个包含分区布局的纯文本文件。这个示例使用 `~/my-partition` 文件。
2. 使用 `hammer 分区table create` 命令创建 安装介质：

```
# hammer partition-table create --name "My Partition" --snippet false \
--os-family Redhat --file ~/my-partition --organizations "My_Organization" \
--locations "My_Location"
```

2.10. 动态分区示例

使用 Anaconda kickstart 模板，以下部分指示 Anaconda 擦除整个磁盘，自动分区，将分区增大到最大大小，然后继续进行置备过程中的下一个事件序列：

```
zerombr
clearpart --all --initlabel
autopart <%= host_param('autopart_options') %>
```

动态分区由安装程序执行。因此，您可以编写自己的规则来根据节点的运行时信息指定如何对磁盘进行分区，例如：磁盘大小、驱动器、供应商或制造商的数量。

如果要置备服务器并使用动态分区，请作为模板添加以下示例。当包含 `#Dynamic` 条目时，模板的内容加载到 `%pre shell scriptlet` 并创建一个 `/tmp/diskpart.cfg`，然后包含在 Kickstart 分区部分。

```
#Dynamic (do not remove this line)

MEMORY=$(('grep MemTotal: /proc/meminfo | sed 's/^MemTotal: */'|sed 's/ .*//' / 1024))
if [ "$MEMORY" -lt 2048 ]; then
    SWAP_MEMORY=$(( $MEMORY * 2 ))
elif [ "$MEMORY" -lt 8192 ]; then
    SWAP_MEMORY=$MEMORY
elif [ "$MEMORY" -lt 65536 ]; then
```

```

    SWAP_MEMORY=$((MEMORY / 2))
else
    SWAP_MEMORY=32768
fi

cat <<EOF > /tmp/diskpart.cfg
zerombr yes
clearpart --all --initlabel
part /boot --fstype ext4 --size 200 --asprimary
part swap --size "$SWAP_MEMORY"
part / --fstype ext4 --size 1024 --grow
EOF

```

2.11. 调配模板

调配模板定义了卫星服务器在主机上安装操作系统的方式。

红帽卫星包含许多模板示例。在 Satellite Web UI 中，导航到 Hosts > Provisioning templates 来查看它们。您可以创建模板或克隆模板并编辑克隆。对于模板的帮助，请导航至 Hosts > Provisioning templates > Create Template > Help。

模板接受嵌入式 Ruby (ERB) 语法。有关更多信息，请参阅管理 [主机中的模板编写参考](#)。

您可以下载调配模板。在下载模板前，您必须创建 debug 证书。有关更多信息，请参阅 [内容管理指南中的创建机构调试证书](#)。

您可以在卫星服务器和 Git 存储库或本地目录之间同步模板。有关更多信息，请参阅管理主机指南中的 [同步模板存储库指南中的内容](#)。

要查看应用到模板的更改历史记录，请导航到 Hosts > Provisioning templates，选择其中一个模板，然后点 History。点 Revert 使用之前的版本覆盖内容。您还可以恢复到之前的更改。点 Show Diff 查看特定更改的信息：

- Template Diff 选项卡显示调配模板的正文中的更改。
- Details 选项卡显示模板描述中的更改。
- History 选项卡显示更改更改的用户，以及更改日期。

2.12. 调配模板的类型

有很多类型的调配模板：

provision

置备过程的主要模板。例如，一个 kickstart 模板。有关 kickstart 模板语法的更多信息，请参阅 [Red Hat Enterprise Linux 7 安装指南中的 Kickstart 语法参考](#)。

PXELinux, PXEGrub, PXEGrub2

基于 PXE 的模板部署到与子网关联的模板胶囊，以确保主机使用正确的内核选项使用安装程序。对于 BIOS 调配，请选择 PXELinux template。对于 UEFI 置备，请选择 PXEGrub2。

finish

在主置备过程完成后，使用 SSH 连接执行后配置脚本。您只能将 Finishing 模板用于不支持 user_data 的虚拟或云环境中的基于镜像的配置。不要将镜像与 foreman 发现 ISO 混淆，这有时被称为 Foreman 发现镜像。此上下文中的镜像是在虚拟化环境中安装镜像，以方便部署。

当完成脚本随返回代码 0 成功退出时，红帽卫星将代码视为成功，主机将退出构建模式。请注意，有一些完成带有构建模式的脚本，它使用 *调用后端* HTTP 调用。这些脚本不用于基于镜像的调配，而是用于配置 Debian、Ubuntu 和 BSD 等操作系统安装。

user_data

接受自定义数据的供应商的后配置脚本（也称为 seed 数据）。您可以使用 user_data 模板仅在云或虚拟环境中置备虚拟机。此模板不要求卫星能够到达主机；云或虚拟化平台负责向镜像传输数据。

确保要置备的镜像具有读取安装数据的软件，并设置为在启动过程中启动。例如，cloud-init 需要 YAML 输入或 ignition，它预期为 JSON 输入。

cloud_init

有些环境（如 VMWare）不支持自定义数据，或者有自己的数据格式来限制在自定义过程中可以执行的操作。在这种情况下，您可以使用 foreman 插件配置 cloud-init 客户端，该插件会尝试通过 HTTP 或 HTTPS 直接从 Satellite 下载模板。这种技术可以在任何环境中使用，最好是虚拟化。

确保您满足使用 cloud_init 模板的以下要求：

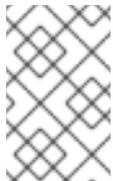
- 确保要置备的镜像具有读取安装数据的软件，并设置为在启动过程中启动。
- 调配的主机能够从与主机调配接口 IP 匹配的 IP 地址访问卫星。
请注意，cloud-init 无法在 NAT 后面工作。

Bootdisk

无 PXE 引导方法的模板。

内核执行(kexec)

用于无 PXE 引导方法的内核执行模板。



注意

内核执行是技术预览功能。在 Red Hat Subscription 服务等级协议(SLA)下不支持技术预览功能，可能无法完成，且不适用于生产环境。但是，这些功能可提前访问最新的产品创新，使客户能够在开发过程中测试功能并提供反馈。

脚本

默认不使用任意脚本，但对自定义任务很有用。

ZTP

零接触调配模板。

POAP

PowerOn Auto Provisioning 模板。

iPXE

要使用的 iPXE 或 gPXE 环境的模板，而不使用 PXELinux。

2.13. 创建调配模板

调配模板定义了卫星服务器在主机上安装操作系统的方式。使用此流程创建新调配模板。

流程

1. 在 Satellite Web UI 中，进入 Hosts > Provisioning Templates 并点 Create Template。

2. 在 Name 字段中输入调配模板的名称。
3. 根据需要填写其余字段。Help 选项卡提供有关模板语法的信息，并详细说明可以在模板中不同类型的对象上调用的可用函数、变量和方法。

对于 CLI 用户

1. 在使用 CLI 创建模板之前，先创建一个包含模板的纯文本文件。这个示例使用 ~/my-template 文件。
2. 使用 `hammer template create` 命令创建模板，并使用 `--type` 选项指定类型：

```
# hammer template create --name "My Provisioning Template" \
--file ~/my-template --type provision --organizations "My_Organization" \
--locations "My_Location"
```

2.14. 克隆调配模板

调配模板定义了卫星服务器在主机上安装操作系统的方式。使用此流程克隆模板，并将您的更新添加到克隆中。

流程

1. 在 Satellite Web UI 中，进入 Hosts > Provisioning Templates，再搜索您要使用的模板。
2. 点 Clone 以复制模板。
3. 在 Name 字段中输入调配模板的名称。
4. 选择 Default 复选框，以设置模板以自动与新组织或位置关联。
5. 在 Template 编辑器 字段中，输入调配模板的正文。您还可以使用模板文件浏览器 上传模板文件。
6. 在 Audit Comment 字段中，为审计目的输入对调配模板的更改摘要。
7. 点 Type 选项卡，如果模板是代码片段，请选择 Snippet 复选框。代码片段不是单机调配模板，而是可插入到其他调配模板中的调配模板的一部分。
8. 从 Type 列表中，选择模板的类型。例如，Provisioning 模板。
9. 点 Association 选项卡和 Applicable Operating Systems 列表中，选择您要与调配模板关联的操作系统名称。
10. （可选）点击 Add combination，然后从 Host Group 列表选择一个主机组，或者从 Environment 列表中将调配模板与主机组和环境关联。
11. 单击 组织和位置 选项卡，为模板添加任何额外上下文。
12. 单击 Submit 以保存您的调配模板。

2.15. 创建计算配置文件

您可以使用计算配置集预定义虚拟机硬件详情，如 CPU、内存和存储。Red Hat Satellite 的默认安装包含三个预定义的配置集：

- 1-Small
- 2-medium
- 3-Large

流程

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Profiles，再点击 Create Compute Profile。
2. 在 Name 字段中输入配置集的名称。
3. 点 Submit。这时将打开一个新窗口，其中包含 compute 配置集的名称。
4. 在新窗口中，单击每个计算资源的名称，再编辑为此计算配置文件设置的属性。

对于 CLI 用户

计算配置文件 CLI 命令尚未在红帽卫星 6.9 中实施。

2.16. 为主机设置默认加密根密码

如果您不想为您置备的主机设置纯文本默认 root 密码，您可以使用默认的加密密码。

流程

1. 生成加密的密码：

```
# python -c 'import crypt,getpass;pw=getpass.getpass(); print(crypt.crypt(pw)) if (pw==getpass.getpass("Confirm: ")) else exit()'
```

2. 复制密码以供以后使用。
3. 在 Satellite Web UI 中，导航到 Administer > Settings。
4. 在 Settings 页面中，选择 Provisioning 选项卡。
5. 在 Name 列中，导航到 Root 密码，然后单击 单击以编辑。
6. 粘贴加密的密码，然后单击保存。

2.17. 使用 NOVNC 访问虚拟机

您可以使用浏览器访问由 Satellite 创建的虚拟机的 VNC 控制台。

Satellite 支持在以下虚拟化平台上使用 noVNC：

- VMware
- libvirt
- Red Hat Virtualization

前提条件

- 您必须具有由 Satellite 创建的虚拟机。
- 对于现有虚拟机，请确保 Compute 资源 设置中的 Display 类型是 VNC。
- 您必须将 Katello root CA 证书导入到卫星服务器中。在浏览器中添加安全例外并不足以用于使用 noVNC。如需更多信息，请参阅管理 *Red Hat Satellite* 指南中的 [安装 Katello Root CA 证书](#) 部分。

流程

1. 在 VM 主机系统中，将防火墙配置为允许端口 5900 到 5930 的 VNC 服务：

- On Red Hat Enterprise Linux 6:

```
# iptables -A INPUT -p tcp --dport 5900:5930 -j ACCEPT  
# service iptables save
```

- On Red Hat Enterprise Linux 7:

```
# firewall-cmd --add-port=5900-5930/tcp  
# firewall-cmd --add-port=5900-5930/tcp --permanent
```

2. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources 并选择计算资源的名称。
3. 在 Virtual Machines 选项卡中，选择 VM 主机的名称。确保机器已打开，然后选择 Console。

第 3 章 配置网络

每个置备类型都需要一些网络配置。使用本章在卫星服务器上配置集成胶囊中的网络服务。

新主机必须有权访问您的胶囊服务器。胶囊服务器可以是您在卫星服务器或外部胶囊服务器上集成胶囊。当主机位于隔离的网络上时，您可能要调配来自外部胶囊服务器的主机，并且无法直接连接到卫星服务器，或者内容与胶囊服务器同步时。使用外部胶囊服务器进行调配可以保存在网络带宽上。

配置胶囊服务器有两个基本要求：

1. 配置网络服务,这包括：
 - 内容交付服务
 - 网络服务(DHCP、DNS 和 TFTP)
 - Puppet 配置
2. 在卫星服务器中定义网络资源数据，以帮助在新主机上配置网络接口。

以下说明与配置独立胶囊管理特定网络类似的应用。要将卫星配置为使用外部 DHCP、DNS 和 TFTP 服务，请参阅 [从连接的网络安装卫星服务器](https://access.redhat.com/documentation/en-us/red_hat_satellite/6.9/html-single/installing_satellite_server_from_a_connected_network/index#configuring-external-services)。 https://access.redhat.com/documentation/en-us/red_hat_satellite/6.9/html-single/installing_satellite_server_from_a_connected_network/index#configuring-external-services

3.1. 网络资源

Satellite 包含您必须设置和配置才能创建主机的网络资源。Satellite 包括以下网络资源：

域

您必须将由卫星管理的每个主机分配到某个域。利用该域，卫星可以管理 A、AAAA 和 PTR 记录。即使您不希望 Satellite 管理 DNS 服务器，您仍必须创建并关联至少一个域。域包含在卫星主机的命名惯例中，例如，example.com 域中名称为 test123 的主机，其具有完全限定域名 test123.example.com。

子网

您必须将由 Satellite 管理的每个主机分配给子网。然后，卫星可以使用子网管理 IPv4 保留。如果没有保留集成，您仍必须创建并关联至少一个子网。当您在卫星中管理子网时，您无法为 Satellite 外部的子网创建 DHCP 记录。在 Satellite 中，您可以使用 IP 地址管理(IPAM)使用以下选项之一管理 IP 地址：

- **DHCP**：DHCP Capsule 通过从范围的第一个地址开始查找下一个可用 IP 地址来管理 IP 地址分配，并跳过保留的所有地址。在分配 IP 地址之前，胶囊发送 ICMP 和 TCP ping 来检查 IP 地址是否在使用。请注意，如果主机已关闭，或者配置了要禁用连接的防火墙，卫星将假定 IP 地址可用。此检查不适用于关闭的主机，因此 DHCP 选项只能与卫星控制的子网一起使用，并且没有外部创建的主机。
胶囊 DHCP 模块会在短时间内保留提供的 IP 地址，以防止并发访问期间发生冲突，因此 IP 范围内的一些 IP 地址可能会临时使用。
- **内部 DB**：卫星通过按顺序从卫星数据库中排除所有 IP 地址，从子网范围内查找下一个可用 IP 地址。数据的主要来源是数据库，而不是 DHCP 保留。当多个主机被并行创建时，这个 IPAM 无法安全，在这种情况下，使用 DHCP 或 Random DB IPAM。

- **随机 DB**：卫星通过随机从卫星数据库中排除所有 IP 地址，从子网范围内查找下一个可用 IP 地址。数据的主要来源是数据库，而不是 DHCP 保留。这个 IPAM 可以安全地用于并发主机创建，因为 IP 地址以随机顺序返回，从而最大程度地减少冲突的可能性。
- **EUI-64**：扩展唯一识别符(EUI) 64 位 IPv6 地址生成，根据 RFC2373，通过 48 位 MAC 地址获取。
- **外部 IPAM**：通过胶囊功能将 IPAM 划分为外部系统。Satellite 目前没有附带任何外部 IPAM 实施，但一些插件正在开发中。
- **无**：必须手动输入每个主机的 IP 地址。
选项 DHCP、内部 DB 和 Random DB 可能会导致 DHCP 与外部创建的记录冲突。这些子网必须受专属的卫星控制。

有关添加子网的更多信息，请参阅 [第 3.7 节“将子网添加到卫星服务器”](#)。

DHCP 范围

您可以在卫星服务器中为发现和置备的系统定义相同的 DHCP 范围，但在同一子网中的每个服务都使用单独的范围。

3.2. SATELLITE 和 DHCP 选项

卫星通过 DHCP 胶囊管理 DHCP 保留。卫星还设置 `next-server` 和 `filename` DHCP 选项。

next-server 选项

`next-server` 选项提供从其引导的 TFTP 服务器的 IP 地址。默认情况下不设置这个选项，必须为每个 TFTP 胶囊设置。您可以使用 `satellite-installer` 命令及 `--foreman-proxy-tftp-servername` 选项在 `/etc/foreman-proxy/settings.d/tftp.yml` 文件中设置 TFTP 服务器：

```
# satellite-installer --foreman-proxy-tftp-servername 1.2.3.4
```

然后，每个 TFTP 胶囊通过 API 报告此设置，卫星可在创建 DHCP 记录时检索配置信息。

当 PXE loader 设置为 `none` 时，卫星不会将 `next-server` 选项填充到 DHCP 记录中。

如果 `next-server` 选项处于未定义，卫星使用反向 DNS 搜索来查找 TFTP 服务器地址来分配，但您可能会遇到以下问题：

- 置备过程中 DNS 超时
- 查询不正确的 DNS 服务器。例如，权威而不是缓存
- 有关 TFTP 服务器 IP 地址错误的问题。例如，PTR 记录无效

如果您遇到这些问题，请检查卫星和胶囊上的 DNS 设置，特别是 PTR 记录解析。

文件名选项

`filename` 选项包含文件在调配期间下载并执行的完整路径。您为主机或主机组选择的 PXE 装载程序定义了要使用的文件名选项。当 PXE loader 设置为 `none` 时，卫星不会将 `filename` 选项填充到 DHCP 记录中。根据 PXE loader 选项，文件名更改如下：

PXE loader 选项	文件名条目	备注
PXELinux BIOS	pxelinux.0	
PXELinux UEFI	pxelinux.efi	
iPXE 链 BIOS	undionly.kpxe	
PXEGrub2 UEFI	grub2/grubx64.efi	x64 会根据构架的不同
iPXE UEFI HTTP	http://capsule.example.com:8000/httpboot/ipxe-x64.efi	需要 httpboot 功能并将 文件名 呈现为完整的 URL，其中 <i>capsule.example.com</i> 是卫星中的已知主机名胶囊。
Grub2 UEFI HTTP	http://capsule.example.com:8000/httpboot/grub2/grubx64.efi	需要 httpboot 功能并将 文件名 呈现为完整的 URL，其中 <i>capsule.example.com</i> 是卫星中的已知主机名胶囊。

3.3. 卫星中的 DHCP 问题故障排除

卫星可以在内部或外部 DHCP 胶囊上管理 ISC DHCP 服务器。Satellite 可以列出、创建和删除 DHCP 保留和租期。但是，您可能遇到了一些问题。

同步 DHCP 记录

当在 DHCP 编配过程中发生错误时，卫星数据库中 DHCP 记录和 DHCP 服务器可能不匹配。要解决这个问题，您必须在 DHCP 服务器中添加 Satellite 数据库中缺少的 DHCP 记录，然后执行以下步骤从 DHCP 服务器中删除不需要的记录：

1. 要预览将添加到 DHCP 服务器的 DHCP 记录，请输入以下命令：

```
# foreman-rake orchestration:dhcp:add_missing subnet_name=NAME
```

2. 如果您对上一步中的预览更改满意，请使用 **perform=1** 参数输入以上命令来应用它们：

```
# foreman-rake orchestration:dhcp:add_missing subnet_name=NAME perform=1
```

3. 要在卫星中和 DHCP 服务器中保持 DHCP 记录同步，您可以从 DHCP 服务器中删除不需要的 DHCP 记录。请注意，卫星假设所有受管 DHCP 服务器不包含第三方记录，因此此步骤可能会删除这些意外的记录。要预览将哪些记录从 DHCP 服务器中删除，请输入以下命令：

```
# foreman-rake orchestration:dhcp:remove_offending subnet_name=NAME
```

4. 如果您对上一步中的预览更改满意，请使用 **perform=1** 参数输入以上命令来应用它们：

```
# foreman-rake orchestration:dhcp:remove_offending subnet_name=NAME perform=1
```

PXE loader 选项更改

当为现有主机更改 PXE loader 选项时，这会导致 DHCP 冲突。唯一的临时解决方案是覆盖 DHCP 条目。

DHCP 文件的权限不正确

操作系统更新可以更新 `dhcpd` 软件包。这会导致重要目录和文件的权限重置，以便 DHCP 胶囊无法读取所需的信息。

如需更多信息，请参阅从 [Satellite 服务器错误 ERF12-6899 ProxyAPI::ProxyException 置备主机时的 DHCP 错误：无法设置 DHCP 条目 RestClient::ResourceNotFound 404 Resource Not Found on Red Hat Knowledgebase](#)。

更改 DHCP Capsule 条目

卫星只为分配给具有 DHCP 胶囊设置的子网的主机管理 DHCP 记录。如果您创建主机，在尝试删除主机时清除或更改 DHCP Capsule，则操作会失败。

如果您在不设置 DHCP Capsule 的情况下创建主机，并尝试设置 DHCP 胶囊，这会导致 DHCP 冲突。

已删除 `dhcpd.leases` 文件中的主机条目

对 DHCP 租期的任何更改都会附加到 `dhcpd.leases` 文件的末尾。由于条目附加到文件中，所以同一租期的两个或更多条目可以同时存在于 `dhcpd.leases` 文件中。当同一租期有两个或多个条目时，文件中的最后一个条目将具有优先权。租期文件中的组、子组和主机声明以相同的方式处理。如果删除了租期，则 { 删除; } 会附加到声明中。

3.4. 基于镜像的置备的先决条件

后引导配置方法

使用完成的启动后配置脚本的镜像需要受管 DHCP 服务器，如卫星集成胶囊或外部胶囊。主机必须与与 DHCP 胶囊关联的子网创建，主机的 IP 地址必须是来自 DHCP 范围内的有效 IP 地址。

可以使用外部 DHCP 服务，但必须手动输入 IP 地址。必须在卫星中配置与镜像中配置对应的 SSH 凭据，以启用启动后配置。

在对从依赖于安装后脚本的镜像引导虚拟机时，检查以下项目：

- 主机在卫星服务器中分配了子网。
- 子网在卫星服务器中分配有 DHCP 胶囊。
- 主机在卫星服务器中分配有有效的 IP 地址。
- 使用 DHCP 的虚拟机获取的 IP 地址与卫星服务器中配置的地址相匹配。

- 从镜像创建的虚拟机响应 SSH 请求。
- 从镜像创建的虚拟机通过 SSH 授权用户和密码，与要部署的镜像相关联。
- 卫星服务器可以通过 SSH 密钥访问虚拟机。虚拟机需要从卫星服务器接收后配置脚本。

预初始化配置方法

使用 cloud-init 脚本的镜像需要 DHCP 服务器，以避免在镜像中包含 IP 地址。首选使用托管的 DHCP 胶囊。镜像必须将 cloud-init 服务配置为在系统启动时启动，并获取要完成配置中使用的脚本或配置数据。

在对虚拟机从依赖于镜像中包含的初始化脚本进行引导时，请检查以下项目：

- 子网中有一个 DHCP 服务器。
- 虚拟机已安装并启用了 cloud-init 服务。

有关在虚拟机镜像中完成和 cloud-init 脚本的不同支持级别的信息，请参阅红帽客户门户网站上 [支持的计算资源是什么](#)，以及红帽客户门户网站中的 [cloud-init 脚本](#) 和 [cloud-init 脚本](#)。

3.5. 配置网络服务

某些调配方法使用胶囊服务器服务。例如，网络可能需要胶囊服务器来充当 DHCP 服务器。网络也可以使用 PXE 引导服务在新主机上安装操作系统。这要求将胶囊服务器配置为使用主 PXE 引导服务：DHCP、DNS 和 TFTP。

使用 `satellite-installer` 命令及选项来在卫星服务器上配置这些服务。

要在外部胶囊服务器上配置这些服务，请运行 `satellite-installer --scenario Capsule`。

卫星服务器使用 `eth0` 进行外部通信，如连接到红帽的 CDN。

流程

要在 **Satellite** 集成胶囊上配置网络服务，请完成以下步骤：

1.

输入 **satellite-installer** 命令来配置所需的网络服务：

```
# satellite-installer --foreman-proxy-dhcp true \  
--foreman-proxy-dhcp-managed true \  
--foreman-proxy-dhcp-gateway "192.168.140.1" \  
--foreman-proxy-dhcp-interface "eth1" \  
--foreman-proxy-dhcp-nameservers "192.168.140.2" \  
--foreman-proxy-dhcp-range "192.168.140.10 192.168.140.110" \  
--foreman-proxy-dhcp-server "192.168.140.2" \  
--foreman-proxy-dns true \  
--foreman-proxy-dns-managed true \  
--foreman-proxy-dns-forwarders "8.8.8.8; 8.8.4.4" \  
--foreman-proxy-dns-interface "eth1" \  
--foreman-proxy-dns-reverse "140.168.192.in-addr.arpa" \  
--foreman-proxy-dns-server "127.0.0.1" \  
--foreman-proxy-dns-zone "example.com" \  
--foreman-proxy-tftp true \  
--foreman-proxy-tftp-managed true
```

2.

找到您配置的胶囊服务器：

```
# hammer proxy list
```

3.

刷新胶囊服务器的功能以查看更改：

```
# hammer proxy refresh-features --name "satellite.example.com"
```

4.

验证胶囊服务器上配置的服务：

```
# hammer proxy info --name "satellite.example.com"
```

3.5.1. 通过安装程序进行多个子网或域

satellite-installer 选项只允许单个 **DHCP** 子网或 **DNS** 域。定义多个子网的一种方法是使用自定义配置文件。

对于每个额外子网或域，在 `/etc/foreman-installer/custom-hiera.yaml` 文件中创建一个条目：

```
dhcp::pools:
  isolated.lan:
    network: 192.168.99.0
    mask: 255.255.255.0
    gateway: 192.168.99.1
    range: 192.168.99.5 192.168.99.49

dns::zones:
  # creates @ SOA $::fqdn root.example.com.
  # creates $::fqdn A $::ipaddress
  example.com: {}

  # creates @ SOA test.example.net. hostmaster.example.com.
  # creates test.example.net A 192.0.2.100
  example.net:
    soa: test.example.net
    soaip: 192.0.2.100
    contact: hostmaster.example.com.

  # creates @ SOA $::fqdn root.example.org.
  # does NOT create an A record
  example.org:
    reverse: true

  # creates @ SOA $::fqdn hostmaster.example.com.
  2.0.192.in-addr.arpa:
    reverse: true
    contact: hostmaster.example.com.
```

执行 `satellite-installer` 来执行更改，并验证 `/etc/dhcp/dhcpd.conf` 是否包含相应的条目。然后，必须在卫星数据库中定义子网。

3.5.2. 用于网络配置的 DHCP、DNS 和 TFTP 选项

DHCP 选项

`--foreman-proxy-dhcp`

启用 DHCP 服务。您可以将这个选项设置为 `true` 或 `false`。

`--foreman-proxy-dhcp-managed`

启用 Foreman 管理 DHCP 服务。您可以将这个选项设置为 `true` 或 `false`。

`--foreman-proxy-dhcp-gateway`

DHCP 池网关。将其设置为您专用网络上主机的外部网关地址。

--foreman-proxy-dhcp-interface

设置要侦听请求的 DHCP 服务的接口。将其设置为 eth1。

--foreman-proxy-dhcp-nameservers

设置通过 DHCP 向客户端提供的名称服务器的地址。把它设置为 eth1 上卫星服务器的地址。

--foreman-proxy-dhcp-range

空格分隔的 DHCP 池范围用于 Discovered 和 Unmanaged 服务。

--foreman-proxy-dhcp-server

设置要管理的 DHCP 服务器的地址。

DNS 选项

--foreman-proxy-dns

启用 DNS 服务。您可以将这个选项设置为 true 或 false。

--foreman-proxy-dns-managed

启用 Foreman 管理 DNS 服务。您可以将这个选项设置为 true 或 false。

--foreman-proxy-dns-forwarders

设置 DNS 转发器。将其设置为 DNS 服务器。

--foreman-proxy-dns-interface

设置侦听 DNS 请求的接口。将其设置为 eth1。

--foreman-proxy-dns-reverse

DNS 反向区域名称。

--foreman-proxy-dns-server

设置要管理的 DNS 服务器的地址。

--foreman-proxy-dns-zone

设置 DNS 区域名称。

TFTP 选项

`--foreman-proxy-tftp`

启用 TFTP 服务。您可以将这个选项设置为 `true` 或 `false`。

`--foreman-proxy-tftp-managed`

启用 Foreman 管理 TFTP 服务。您可以将这个选项设置为 `true` 或 `false`。

`--foreman-proxy-tftp-servername`

设置要使用的 TFTP 服务器。确保您使用胶囊的 IP 地址。

运行 `satellite-installer --help` 查看更多与 DHCP、DNS、TFTP 和其他 Satellite Capsule 服务相关的选项

3.5.3. 通过 NAT 使用 TFTP 服务

您可以通过 NAT 使用 Satellite TFTP 服务。要做到这一点，在所有 NAT 路由器或防火墙中，您必须在 UDP 端口 69 上启用 TFTP 服务，并启用 TFTP 状态跟踪功能。如需更多信息，请参阅您的 NAT 设备文档。

在 Red Hat Enterprise Linux 7 中使用 NAT:

使用以下命令，允许在 UDP 端口 69 上允许 TFTP 服务，加载内核 TFTP 状态跟踪模块，并永久保留更改：

```
# firewall-cmd --add-service=tftp && firewall-cmd --runtime-to-permanent
```

对于在 Red Hat Enterprise Linux 6 上运行的 NAT:

1. 配置防火墙以允许端口 69 上的 TFTP 服务 UDP。

```
# iptables -A OUTPUT -i eth0 -p udp --sport 69 -m state \
--state ESTABLISHED -j ACCEPT
# service iptables save
```

2. 加载 `ip_conntrack_tftp` 内核 TFTP 状态模块。在 `/etc/sysconfig/iptables-config` 文件中，

找到 `IPTABLES_MODULES` 并添加 `ip_conntrack_ftp`，如下所示：

```
IPTABLES_MODULES="ip_conntrack_ftp"
```

3.6. 将域添加到卫星服务器

卫星服务器为网络上的每个主机定义域名。卫星服务器必须具有关于域和负责域名分配的胶囊服务器的信息。

检查现有域

卫星服务器可能已经有作为卫星服务器安装一部分创建的相关域。将上下文切换到 **Any Organization** 和 **Any Location**，然后检查域列表以查看其是否存在。

DNS 服务器配置注意事项

在创建 DNS 记录期间，卫星执行冲突 DNS 查找来验证主机名是否不在活跃使用中。这个检查针对以下 DNS 服务器之一运行：

- 如果 **Administer > Settings > Query local nameserver** 设置为 `true`，则系统范围解析器为 `true`。
- 与主机关联的子网中定义的名称服务器。
- 从 SOA 查询的权威 NS-Records，来自与主机关联的域名。

如果您在 DNS 冲突解析过程中遇到超时，请检查以下设置：

- 子网名称服务器必须可从卫星服务器访问。
- 域名必须具有卫星服务器的 **Start of Authority (SOA)** 记录。
- `/etc/resolv.conf` 文件中的系统解析器必须具有有效且正常工作的配置。

流程

要在 Satellite 中添加域，请完成以下步骤：

1. 在 Satellite Web UI 中，进入 Infrastructure > Domains 并点 Create Domain。
2. 在 DNS Domain 字段中，输入完整 DNS 域名。
3. 在 Fullname 字段中，输入域的纯文本名称。
4. 单击 Parameters 选项卡，再配置任何域级别参数，以应用到附加到此域的主机。例如，要在模板中使用的用户定义布尔值或字符串参数。
5. 单击 Add Parameter，并填写 Name 和 Value 字段。
6. 单击位置选项卡，再添加域所在的位置。
7. 单击组织选项卡，再添加域所属的组织。
8. 单击 Submit 以保存更改。

对于 CLI 用户

使用 hammer 域 create 命令创建域：

```
# hammer domain create --name "domain_name.com" \  
--description "My example domain" --dns-id 1 \  
--locations "My_Location" --organizations "My_Organization"
```

在本例中，--dns-id 选项使用 1，这是您在卫星服务器上集成胶囊的 ID。

3.7. 将子网添加到卫星服务器

您必须将每个子网的信息添加到卫星服务器，因为卫星为新主机配置接口。要配置接口，卫星服务器必须具有连接这些接口的网络的所有信息。

流程

要在 Satellite 服务器中添加子网，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Subnets**，并在 **Subnets** 窗口中点击 **Create Subnet**。
2. 在 **Name** 字段中输入子网的名称。
3. 在 **Description** 字段中，输入子网的描述。
4. 在 **Network address** 字段中，输入子网的网络地址。
5. 在 **Network prefix** 字段中，输入子网的网络前缀。
6. 在 **Network mask** 字段中，输入子网的网络掩码。
7. 在 **Gateway 地址** 字段中，输入子网的外部网关。
8. 在 **主 DNS 服务器** 字段中，为子网输入主 DNS。
9. 在 **Secondary DNS 服务器** 中，为子网输入二级 DNS。
10. 从 **IPAM** 列表中，选择您要用于 IP 地址管理(IPAM)的方法。有关 IPAM 的详情，请参考 [第 3.1 节“网络资源”](#)。
11. 输入您选择的 IPAM 方法的信息。单击 **Remote Execution** 选项卡，再选择控制远程执行的胶囊。

12. 点 **Domains** 选项卡，再选择应用到此子网的域。
13. 单击 **胶囊** 选项卡，再选择应用于子网中每个服务的胶囊，包括 **DHCP**、**TFTP** 和反向 **DNS** 服务。
14. 单击 **Parameters** 选项卡，再配置任何子网级别参数，以应用到附加到此子网的主机。例如，要在模板中使用的用户定义布尔值或字符串参数。
15. 单击 **位置** 选项卡，再选择使用此胶囊的位置。
16. 单击 **Organizations** 选项卡，再选择使用此胶囊的组织。
17. 单击 **Submit** 以保存子网信息。

对于 **CLI** 用户

使用以下命令创建子网：

```
# hammer subnet create --name "My_Network" \  
--description "your_description" \  
--network "192.168.140.0" --mask "255.255.255.0" \  
--gateway "192.168.140.1" --dns-primary "192.168.140.2" \  
--dns-secondary "8.8.8.8" --ipam "DHCP" \  
--from "192.168.140.111" --to "192.168.140.250" --boot-mode "DHCP" \  
--domains "example.com" --dhcp-id 1 --dns-id 1 --tftp-id 1 \  
--locations "My_Location" --organizations "My_Organization"
```



注意

在本例中，**--dhcp-id**、**--dns-id** 和 **--tftp-id** 选项使用 **1**，这是卫星服务器中集成胶囊的 ID。

第 4 章 使用 INFOBLOX 作为 DHCP 和 DNS 供应商

您可以使用胶囊服务器连接到您的 Infoblox 应用来创建和管理 DHCP 和 DNS 记录，并保留 IP 地址。

支持的 Infoblox 版本是 NIOS 8.0 或更高版本，以及 Satellite 6.9 或更高版本。

4.1. 限制

所有 DHCP 和 DNS 记录都只能在单个网络或 DNS 视图中进行管理。在 Capsule 上安装 Infoblox 模块并使用 `satellite-installer` 命令设置视图后，您无法编辑视图。

胶囊服务器使用标准 HTTPS Web API 与单个 Infoblox 节点通信。如果要配置集群和高可用性，请在 Infoblox 中进行配置。

不支持使用 Infoblox 的 TFTP 功能托管与 PXE 相关的文件。您必须将胶囊用作 TFTP 服务器进行 PXE 调配。更多信息请参阅 [第 3 章 配置网络](#)。

Satellite IPAM 功能不能与 Infoblox 集成。

4.2. 前提条件

您必须具有 Infoblox 帐户凭证来管理 Satellite 中的 DHCP 和 DNS 条目。

确定您有 Infoblox 管理角色，其名称为：`DHCP Admin` 和 `DNS Admin`。

管理角色必须具有权限或属于 `admin` 组，以允许帐户通过 Infoblox API 执行任务。

4.3. 在胶囊服务器上安装 INFOBLOX CA 证书

您必须在基础系统上安装 Infoblox HTTPS CA 证书，用于您要与 Infoblox 应用程序集成的所有胶囊。

您可以从 Infoblox web UI 下载证书，也可以使用以下 OpenSSL 命令下载证书：

```
# update-ca-trust enable
# openssl s_client -showcerts -connect infoblox.example.com:443 </dev/null | \
openssl x509 -text >/etc/pki/ca-trust/source/anchors/infoblox.crt
# update-ca-trust extract
```

- ***infoblox.example.com* 条目必须与 X509 证书中的 Infoblox 应用程序的主机名匹配。**

要测试 CA 证书，请使用 CURL 查询：

```
# curl -u admin:password https://infoblox.example.com/wapi/v2.0/network
```

正响应示例：

```
[
  {
    "_ref":
    "network/ZG5zLm5ldHdvcmskMTkyLjE2OC4yMDluMC8yNC8w:infoblox.example.com/24/default",
    "network": "192.168.202.0/24",
    "network_view": "default"
  }
]
```

使用以下红帽知识库文章 来安装证书：[如何在 Red Hat Enterprise Linux 6 / 7 中安装 CA 证书。](#)

4.4. 安装 DHCP INFOBLOX 模块

使用此流程在胶囊上安装 DHCP Infoblox 模块。请注意，您不能在单独的视图中管理记录。

您还可以通过组合使用这个步骤和 DNS Infoblox 模块来同时安装 DHCP 和 DNS Infoblox 模块。
第 4.5 节“[安装 DNS Infoblox 模块](#)”

DHCP Infoblox Record Type 注意事项

仅使用 `--foreman-proxy-plugin-dhcp-infoblox-record-type fixedaddress` 选项来配置 DHCP 和 DNS 模块。

使用主机记录类型设置配置 DHCP 和 DNS Infoblox 模块会导致 DNS 冲突且不被支持。如果您在胶囊服务器上安装了 Infoblox 模块，且 `--foreman-proxy-plugin-dhcp-infoblox-record-type` 选项设为

host, 则必须设置 **DNS Capsule** 和 **Reverse DNS Capsule** 选项, 因为 **Infoblox** 执行 **DNS** 管理本身。您不能在不创建冲突的情况下使用 **host** 选项, 例如, 无法在卫星中重命名主机。

流程

要为 **DHCP** 安装 **Infoblox** 模块, 请完成以下步骤 :

1.

在 **Capsule** 上输入以下命令 :

```
# satellite-installer --enable-foreman-proxy-plugin-dhcp-infoblox \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed false \
--foreman-proxy-dhcp-provider infoblox \
--foreman-proxy-plugin-dhcp-infoblox-record-type fixedaddress \
--foreman-proxy-dhcp-server infoblox.example.com \
--foreman-proxy-plugin-dhcp-infoblox-username admin \
--foreman-proxy-plugin-dhcp-infoblox-password infoblox \
--foreman-proxy-plugin-dhcp-infoblox-network-view default \
--foreman-proxy-plugin-dhcp-infoblox-dns-view default
```

2.

在卫星 **Web UI** 中, 导航到 **Infrastructure > Capsules**, 然后使用 **Infoblox DHCP** 模块选择胶囊, 然后点击 **Refresh**。

3.

确保列出了 **dhcp** 功能。

4.

对于通过 **Infoblox** 管理的所有域, 请确保为该域设置了 **DNS** 胶囊。要验证, 在 **Satellite Web UI** 中, 导航到 **Infrastructure > Domains**, 并检查每个域的设置。

5.

对于通过 **Infoblox** 管理的所有子网, 请确保设置了 **DHCP Capsule** 和 **Reverse DNS Capsule**。要验证, 在 **Satellite Web UI** 中, 导航到 **Infrastructure > Subnets**, 并检查每个子网的设置。

4.5. 安装 DNS INFOBLOX 模块

使用此流程在胶囊上安装 **DNS Infoblox** 模块。您还可以通过组合使用此流程和 [第 4.4 节 “安装 DHCP Infoblox 模块”](#) 来同时安装 **DHCP** 和 **DNS Infoblox** 模块。

DNS 记录只在默认 **DNS** 视图中管理, 无法指定要使用的 **DNS** 视图。

要安装 DNS Infoblox 模块，请完成以下步骤：

1. 在胶囊上，输入以下命令来配置 Infoblox 模块：

```
# satellite-installer --enable-foreman-proxy-plugin-dns-infoblox \  
--foreman-proxy-dns true \  
--foreman-proxy-dns-managed false \  
--foreman-proxy-dns-provider infoblox \  
--foreman-proxy-plugin-dns-infoblox-dns-server infoblox.example.com \  
--foreman-proxy-plugin-dns-infoblox-username admin \  
--foreman-proxy-plugin-dns-infoblox-password infoblox \  
--foreman-proxy-plugin-dns-infoblox-dns-view default
```

另外，您可以更改 `--foreman-proxy-plugin-dns-infoblox-dns-view` 选项的值，以指定除默认视图以外的 DNS Infoblox 视图。

2. 在卫星 Web UI 中，导航到 **Infrastructure > Capsules**，然后使用 **Infoblox DNS** 模块选择胶囊，然后单击 **Refresh**。
3. 确保列出了 **dns** 功能。

第 5 章 将 IPXE 配置为 REDUCE PROVISIONING TIMES

如果您有以下限制，您可以使用 **Satellite** 配置 **PXELinux** 来串联引导 **iPXE**，并使用 **HTTP** 协议引导：

- 带有非受管 **DHCP** 服务器的网络。
- 将网络列入黑名单或受防火墙限制的 **PXE** 服务。
- 由于网络低带宽网络，因此基于 **UDP** 的不可靠 **TFTP** 协议。

iPXE 工作流程概述

使用 **iPXE** 的置备过程遵循此工作流：

- 发现的主机通过 **PXE** 引导。
- 主机加载 **ipxe.efi** 或 **undionly.0**。
- 主机使用 **DHCP** 在网络上再次初始化。
- **DHCP** 服务器检测到 **iPXE** 固件，并使用 **bootstrap** 标记返回 **iPXE** 模板 **URL**。
- 主机请求 **iPXE** 模板。卫星无法识别主机，因为已设置了 **bootstrap** 标志，主机接收与 **Satellite** 附带的 **iPXE** 中间脚本模板。
- 主机运行中间 **iPXE** 脚本并下载发现镜像。
- 主机启动发现操作系统，并执行发现请求。
- 为置备和重启主机调度了主机。

- 主机通过 PXE 引导。
- 上述 workflow 重复，但卫星会识别主机的远程 IP 地址，而不是中间模板，主机会收到常规 iPXE 模板。
- 主机读取 iPXE 配置并引导安装程序。
- 此时，安装遵循常规 PXE 安装 workflow。

请注意，workflow 使用发现过程，它是可选的。要设置发现服务，请参阅 [第 5.1 节“为 iPXE 设置发现服务”](#)。

使用 Satellite 时，您可以将主机设置为通过 TFTP 下载 ipxe.efi 或 undionly.kpxe。文件下载时，所有通信都会继续使用 HTTP。卫星使用 iPXE 调配脚本，以按顺序加载操作系统安装程序或下一个条目。

在 Red Hat Satellite 中使用 iPXE 的方法有三种：

1. 使用 iPXE 作为主固件的虚拟机监控程序进行链引导虚拟机。
2. 使用 PXELinux 通过 TFTP 直接在裸机主机上链式加载 iPXE。
3. 通过 UNDI 使用 PXELinux，它将使用 HTTP 来传输裸机主机上的内核和初始 RAM 磁盘。

安全信息

Red Hat Enterprise Linux 中的 iPXE 二进制文件构建设没有一些安全功能。因此，您只能使用 HTTP，且无法使用 HTTPS。

+ 不支持 Red Hat Enterprise Linux 中 iPXE 的所有与安全相关的功能。如需更多信息，请参阅 [iPXE 中的 Red Hat Enterprise Linux HTTPS 支持](#)。

前提条件

开始前，请确保满足以下条件：

- 要使用 Red Hat Satellite 上有一个主机。
- 置备接口的 MAC 地址与主机配置匹配。
- 主机的置备接口具有有效的 DHCP 保留。
- NIC 能够进行 PXE 启动。如需更多信息，请参阅 http://ipxe.org/appnote/hardware_drivers。
- NIC 与 iPXE 兼容。

5.1. 为 IPXE 设置发现服务

1. 在胶囊服务器上安装 Foreman 发现服务：

```
# yum install foreman-discovery-image
```

2. 在胶囊服务器上，启用 httpboot 服务：

```
# satellite-installer --foreman-proxy-httpboot true
```

3. 在 Satellite Web UI 中，导航到 Administer > Settings，然后点 Provisioning 选项卡。
4. 找到 Default PXE 全局模板条目行，在 Value 列中，将值更改为 discovery。

5.2. 链引导虚拟机

有些虚拟化虚拟机监控程序使用 iPXE 作为 PXE 启动的主固件。因此，您可以在没有 TFTP 和 PXELinux 的情况下串联引导。

链引导虚拟机 workflow

使用虚拟化管理程序可免除对 TFTP 和 PXELinux 的需要。它具有以下 workflow：

1. **虚拟机启动**
2. **iPXE 使用 DHCP 检索网络凭证**
3. **iPXE 使用 DHCP 检索 HTTP 地址**
4. **iPXE 链从模板胶囊加载 iPXE 模板**
5. **iPXE 加载安装程序的内核和初始 RAM 磁盘**

如果要使用发现服务与 iPXE 一起使用，请参阅 [第 5.1 节“为 iPXE 设置发现服务”](#)。

确保您要使用的虚拟机监控程序支持 iPXE。以下虚拟化虚拟机监控程序支持 iPXE：

- **libvirt**
- **Red Hat Virtualization**
- **RHEV**

配置卫星服务器以使用 iPXE

您可以使用默认模板为主机配置 iPXE 引导。如果要更改模板中的默认值，请克隆模板并编辑克隆。

流程

1. 将引导文件复制到 Satellite 服务器上的 TFTP 目录中：

- 对于 EFI 系统，复制 ipxe.efi 文件：

```
# cp /usr/share/ipxe/ipxe.efi /var/lib/tftpboot/
```

- 对于 BIOS 系统，复制 undi.kpxe 文件：

```
# cp /usr/share/ipxe/undionly.kpxe /var/lib/tftpboot/undionly.0
```

2. 在 Satellite Web UI 中，导航到 Hosts > Provisioning Templates，输入 Kickstart default iPXE 并点 Search。
3. 可选：如果要更改模板，点 Clone，输入唯一名称，点 Submit。
4. 点击您要使用的模板的名称。
5. 如果克隆模板，您可以在 Template 选项卡中进行更改。
6. 点 关联 选项卡，然后选择主机使用的操作系统。
7. 单击位置选项卡，再添加主机所在的位置。
8. 单击组织选项卡，再添加主机所属的组织。
9. 单击 Submit 以保存更改。
10. 进入 Hosts > Operating systems 并选择您的主机操作系统。
11. 点 Templates 选项卡。
12. 从 iPXE Template 列表中，选择您要使用的模板。

13. 单击 **Submit** 以保存更改。
14. 进入 **Hosts > All Hosts**。
15. 在 **Hosts** 页面中，选择您要使用的主机。
16. 选择 **Templates** 选项卡。
17. 从 **iPXE 模板** 列表中，选择 **Review** 以验证 **Kickstart** 默认 **iPXE** 模板是否为正确的模板。
18. 要将 **iPXE bootstrapping** 功能用于 **Satellite**，请按如下所示配置 **dhcpd.conf** 文件：

```
if exists user-class and option user-class = "iPXE" {
    filename "http://satellite.example.com/unattended/iPXE?bootstrap=1";
} elsif option architecture = 00:06 {
    filename "ipxe.efi";
} elsif option architecture = 00:07 {
    filename "ipxe.efi";
} elsif option architecture = 00:09 {
    filename "ipxe.efi";
} else {
    filename "undionly.0";
}
```

如果使用隔离网络，请使用带 **TCP 端口 8000** 的胶囊服务器 **URL**，而不是卫星服务器的 **URL**。



注意

当禁用胶囊 **HTTP** 端点时，请使用 <http://satellite.example.com/unattended/iPXE?bootstrap=1> (**installer** 选项 **--foreman-proxy-http false**)。模板胶囊插件在启用后的默认值为 **8000**，可以通过 **--foreman-proxy-http-port** 安装程序选项更改。在这种情况下，使用 <http://capsule.example.com:8000>。每次升级后，您必须更新 **/etc/dhcp/dhcpd.conf** 文件。

5.3. CHAINBOOTING SATELLITE SERVER 以直接使用 IPXE

使用此流程将 iPXE 设置为使用内置驱动程序进行网络通信或 UNDI 接口。有单独的程序将卫星服务器和胶囊配置为使用 iPXE。

您只能在裸机主机中使用这个步骤。

直接或使用 UNDI 工作流程进行链启动 iPXE

1. **主机电源**
2. **PXE 驱动程序使用 DHCP 检索网络凭证**
3. **PXE 驱动程序使用 TFTP 检索 PXELinux 固件 pxelinux.0**
4. **PXELinux 在 TFTP 服务器上搜索配置文件**
5. **PXELinux chainloads iPXE ipxe.lkrn 或 undionly-ipxe.0**
6. **iPXE 再次使用 DHCP 检索网络凭证**
7. **iPXE 使用 DHCP 检索 HTTP 地址**
8. **iPXE 链从模板胶囊加载 iPXE 模板**
9. **iPXE 加载安装程序的内核和初始 RAM 磁盘**

如果要发现服务与 iPXE 一起使用，请参阅 [第 5.1 节“为 iPXE 设置发现服务”](#)。

配置红帽卫星服务器以使用 iPXE

您可以使用默认模板为主机配置 iPXE 引导。如果要更改模板中的默认值，请克隆模板并编辑克隆。

流程

1. 在卫星 Web UI 中，导航到 **Hosts > Provisioning Templates**，输入 **PXELinux chain iPXE** 或 **BIOS 系统**，输入 **PXELinux chain iPXE UNDI**，然后单击 **Search**。
2. 可选：如果要更改模板，点 **Clone**，输入唯一名称，点 **Submit**。
3. 单击您要使用的模板的名称。
4. 如果克隆模板，您可以在 **Template** 选项卡中进行更改。
5. 点 **关联** 选项卡，然后选择主机使用的操作系统。
6. 单击 **位置** 选项卡，再添加主机所在的位置。
7. 单击 **组织** 选项卡，再添加主机所属的组织。
8. 单击 **Submit** 以保存更改。
9. 在 **Provisioning Templates** 页面中，在搜索字段中输入 **Kickstart default iPXE**，然后单击 **Search**。
10. 可选：如果要更改模板，点 **Clone**，输入唯一名称，点 **Submit**。
11. 单击您要使用的模板的名称。
12. 如果克隆模板，您可以在 **Template** 选项卡中进行更改。
13. 点 **关联** 选项卡，并将模板与主机使用的操作系统关联。
14. 单击 **位置** 选项卡，再添加主机所在的位置。

15. 单击组织选项卡，再添加主机所属的组织。
16. 单击 **Submit** 以保存更改。
17. 进入 **Hosts > Operating systems** 并选择您的主机操作系统。
18. 点 **Templates** 选项卡。
19. 从 **PXELinux template** 列表中，选择您要使用的模板。
20. 从 **iPXE 模板** 列表中，选择您要使用的模板。
21. 单击 **Submit** 以保存更改。
22. 导航到 **Hosts > All Hosts**，然后选择要使用的主机。
23. 选择 **Templates** 选项卡，从 **PXELinux template** 列表中选择 **Review**，以验证模板是否正确。
24. 从 **iPXE 模板** 列表中，选择 **Review** 来验证模板是否正确。如果没有 **PXELinux** 条目，或者您无法找到新模板，请导航到 **Hosts > All Hosts**，然后在您的主机上点击 **Edit**。单击 **Operating system** 选项卡，再单击 **Provisioning Template Resolve** 按钮来刷新模板列表。
25. 要将 **iPXE bootstrapping** 功能用于 **Satellite**，请按如下所示配置 **dhcpd.conf** 文件：

```
if exists user-class and option user-class = "iPXE" {
    filename "http://satellite.example.com/unattended/iPXE?bootstrap=1";
} elsif option architecture = 00:06 {
    filename "ipxe.efi";
} elsif option architecture = 00:07 {
    filename "ipxe.efi";
} elsif option architecture = 00:09 {
    filename "ipxe.efi";
```

```

} else {
  filename "undionly.0";
}

```

如果使用隔离网络，请使用带 TCP 端口 8000 的胶囊服务器 URL，而不是卫星服务器的 URL。



注意

对于 <http://satellite.example.com/unattended/iPXE>，您还可以使用 Red Hat Satellite Capsule <http://capsule.example.com:8000/unattended/iPXE>。每次升级后，您必须更新 `/etc/dhcp/dhcpd.conf` 文件。

5.4. CHAINBOOTING RED HAT SATELLITE CAPSULE 以直接使用 IPXE

您必须在所有胶囊上执行此步骤。

流程

1. 安装 `ipxe-bootimgs` RPM 软件包：

```
# yum install ipxe-bootimgs
```

2. 将 iPXE 固件复制到 TFTP 服务器的根目录。不要使用符号链接，因为 TFTP 在 `chroot` 环境中运行。

- 对于 EFI 系统，复制 `ipxe.efi` 文件：

```
# cp /usr/share/ipxe/ipxe.lkrn /var/lib/tftpboot/
```

- 对于 BIOS 系统，复制 `undi.kpxe` 文件：

```
# cp /usr/share/ipxe/undionly.kpxe /var/lib/tftpboot/undionly-ipxe.0
```

3. 修正文件上下文：

```
# restorecon -RvF /var/lib/tftpboot/
```

第 6 章 使用 PXE 调配主机

使用 Red Hat Satellite 6.9 置备裸机实例有四种主要方法：

无人值守置备

新主机由 MAC 地址标识，卫星服务器使用 PXE 引导过程调配主机。

通过 Discovery 进行无人值守置备

新主机使用 PXE 引导来加载卫星发现服务。此服务识别主机的硬件信息，并将它列为要置备的可用主机。更多信息请参阅 [第 7 章 配置发现服务](#)。

无 PXE 的置备

使用卫星服务器生成的引导磁盘或无 PXE 的发现镜像置备新主机。

使用 Discovery 进行无 PXE 的置备

新主机使用 ISO 引导磁盘来加载卫星发现服务。此服务识别主机的硬件信息，并将它列为要置备的可用主机。更多信息请参阅 [第 7.7 节 “实施 PXE 无发现”](#)。

BIOS 和 UEFI 支持

使用 Red Hat Satellite，可以执行 BIOS 和基于 UEFI 的 PXE 调配。

BIOS 和 UEFI 接口作为计算机操作系统和固件之间的解释器工作，并在启动时初始化硬件组件并启动操作系统。

有关支持的工作流的信息，[请参阅支持的架构和置备场景](#)。

在卫星调配中，PXE loader 选项定义调配期间要使用的 DHCP 文件名选项。对于 BIOS 系统，使用 PXELinux BIOS 选项启用调配的节点，以通过 TFTP 下载 pxelinux.0 文件。对于 UEFI 系统，使用 PXEGrub2 UEFI 选项使 TFTP 客户端下载 grub2/grubx64.efi 文件。

对于 BIOS 调配，您必须将 PXELinux 模板与操作系统关联。

对于 UEFI 置备，您必须将 PXEGrub2 模板与操作系统关联。

如果您同时关联 PXELinux 和 PXEGrub2 模板，卫星 6 可以在 TFTP 服务器上为两个配置文件部署配置文件，以便您轻松地在 PXE 加载程序间切换。

6.1. 裸机置备先决条件

裸机置备的要求包括：

- 管理裸机主机的胶囊服务器。为了进行无人值守调配和基于发现的配置，卫星服务器需要 PXE 服务器设置。

有关网络要求的详情请参考 [第 3 章 配置网络](#)。

有关发现服务的更多信息，[第 7 章 配置发现服务](#)。
- 裸机主机或空白虚拟机。
- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。

有关无人值守和无 PXE 的置备安全令牌的详情，请参考 [第 6.2 节“配置安全令牌有效期持续时间”](#)。

6.2. 配置安全令牌有效期持续时间

当执行无人值守和无 PXE 的调配时，卫星会自动生成唯一令牌，并将此令牌添加到 PXE 配置文件 (PXELinux, Grub2) 中的 kickstart URL 中。

默认情况下，令牌有效的 360 分钟。当您置备主机时，请确保在这个时间范围内重新引导主机。如果令牌过期，它不再有效，并且您收到 404 错误，并且操作系统安装程序下载会失败。

要调整令牌有效期，在卫星 Web UI 中，导航到 **Administer > Settings**，然后单击 **Provisioning** 选项卡。找到 **Token duration** 选项，点编辑图标并编辑持续时间，或者输入 0 来禁用令牌生成。

如果禁用令牌生成，攻击者可能会欺骗客户端 IP 地址并从卫星服务器下载 kickstart，包括加密的 root 密码。

6.3. 使用未置备置备创建主机

无人值守调配是最简单的主机调配形式。您在卫星服务器上输入主机详情并启动主机。卫星服务器自动管理 PXE 配置，组织网络服务，并为主机提供操作系统和配置。

这种置备主机的方法在安装过程中使用最少的交互。

流程

要创建带有无人值守置备的主机，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Hosts > Create Host**。
2. 在 **Name** 字段中输入主机的名称。
3. 点击 **Organization** 和 **Location** 选项卡，并更改上下文以符合您的要求。
4. 从 **Host Group** 列表中，选择您要用来填充表单的主机组。
5. 点 **Interface** 选项卡，在主机的接口上点 **Edit**。
6. 验证字段是否填充了值。特别是：
 - **Host** 选项卡中的 **Name** 将成为 DNS 名称。
 - 卫星服务器会自动为新主机分配 IP 地址。

7.
在 **MAC 地址** 字段中，为主机输入 **MAC 地址**。这样可确保在 **PXE 引导** 过程中识别主机。
8.
确保卫星服务器自动为主机上的第一个接口选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。
9.
在 **MAC 地址** 字段中，输入主机的置备接口的 **MAC 地址**。这样可确保在 **PXE 引导** 过程中识别主机。
10.
点 **OK** 保存。要添加其他接口，请单击 **Add Interface**。您只能将一个接口用于 **Provision** 和 **Primary**。
11.
单击 **Operating System** 选项卡，然后验证所有字段是否包含值。确认操作系统的各个方面。
12.
可选：点 **Provisioning** 模板中的 **Resolve** 来检查新主机可识别要使用的正确调配模板。

有关关联调配模板的更多信息，请参阅 [第 2.11 节“调配模板”](#)。
13.
单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。
14.
单击 **Submit** 以保存主机详情。

有关网络接口的更多信息，请参阅添加 [网络接口](#)。

这会创建主机条目和相关置备设置。这也包括为 **PXE 引导** 裸机主机创建所需的目录和文件。如果启动物理主机并将其引导模式设置为 **PXE**，则主机检测到卫星服务器集成胶囊的 **DHCP** 服务，接收 **Kickstart** 树的 **HTTP** 端点并安装操作系统。

安装完成后，主机还使用激活密钥注册到卫星服务器，并从 **Satellite Tools 6.9** 存储库安装必要的配置和管理工具。

对于 **CLI** 用户

使用 `hammer host create` 命令创建主机。

```
# hammer host create --name "My_Unattended_Host" --organization "My_Organization" \
--location "My_Location" --hostgroup "My_Host_Group" --mac "aa:aa:aa:aa:aa:aa" \
--build true --enabled true --managed true
```

使用 `hammer 主机接口更新` 命令，确保设置了网络接口选项。

```
# hammer host interface update --host "test1" --managed true \
--primary true --provision true
```

6.4. 使用无 PXE 的置备创建主机

有些硬件不提供 PXE 引导接口。在卫星中，您可以在不 PXE 引导的情况下调配主机。这也被称为无 PXE 的置备，它涉及生成主机可以使用的引导 ISO。使用此 ISO，主机可以连接到卫星服务器，引导安装介质并安装操作系统。

卫星还提供无 PXE 的发现服务，它在没有基于 PXE 的服务的情况下运行，如 DHCP 和 TFTP。更多信息请参阅第 7.7 节“实施 PXE 无发现”。

启动 ISO 类型

有四种类型的引导 ISO：

主机镜像

特定主机的引导 ISO。此镜像仅包含访问卫星服务器中的安装介质所需的引导文件。用户在卫星中定义子网数据，并且镜像是使用静态网络创建的。

完整主机镜像

包含特定主机的内核和初始 RAM 磁盘镜像的引导 ISO。如果主机无法正确链加载，则此镜像很有用。调配模板仍从卫星服务器下载。

通用镜像

没有与特定主机关联的引导 ISO。ISO 将主机的 MAC 地址发送到卫星服务器，该服务器与主机条目匹配。该镜像不存储 IP 地址详情，并需要访问网络上的 DHCP 服务器到 bootstrap。此镜像也可以从卫星服务器上的 `/bootdisk/disks/generic` URL 获取，例如 <https://satellite.example.com/bootdisk/disks/generic>。

子网镜像

类似于通用镜像的引导 ISO，但配置了胶囊服务器的地址。此镜像对于在同一子网上具有调配 NIC 的所有主机都是通用的。

主机镜像 和完整主机 镜像包含调配令牌，因此生成的镜像具有有限的期限。有关配置安全令牌的更多信息，请阅读 [第 6.2 节“配置安全令牌有效期持续时间”](#)。



注意

完整主机镜像 基于 SYSLINUX，并与所有红帽企业 Linux 认证硬件一起工作。使用主机映像、通用映像或子网映像时，请参见 http://ipxe.org/appnote/hardware_drivers 了解预期处理基于 IPXE 的引导磁盘的硬件驱动程序列表。

流程

要创建带有 PXE 无 PXE 置备的主机，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Hosts > Create Host**。
2. 在 **Name** 字段中输入您要成为置备的系统主机名的名称。
3. 单击 **Organization** 和 **Location** 选项卡，并更改上下文以符合您的要求。
4. 从 **Host Group** 列表中，选择您要用来填充表单的主机组。
5. 点 **Interface** 选项卡，在主机的接口上点 **Edit**。
6. 验证字段是否填充了值。特别是：
 - **Host** 选项卡中的 **Name** 将成为 DNS 名称。
 - 卫星服务器会自动为新主机分配 IP 地址。

7. 在 **MAC 地址** 字段中，为主机输入 **MAC 地址**。
8. 确保卫星服务器自动为主机上的第一个接口选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。
9. 单击 **Operating System** 选项卡，然后验证所有字段是否包含值。确认操作系统的各个方面。
10. 单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。

有关关联调配模板的更多信息，请参阅 [第 2.11 节“调配模板”](#)。
11. 单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。
12. 单击 **Submit** 以保存主机详情。

这会创建一个主机条目，并显示主机详情页面。

窗口右上角的选项是引导磁盘菜单。在这个菜单中，以下镜像之一可用于下载：主机映像、完整主机映像、通用映像和子网镜像。

对于 **CLI 用户**

使用 **hammer host create** 命令创建主机。

```
# hammer host create --name "My_Bare_Metal" --organization "My_Organization" \
--location "My_Location" --hostgroup "My_Host_Group" --mac "aa:aa:aa:aa:aa:aa" \
--build true --enabled true --managed true
```

确定使用 **hammer 主机接口更新** 命令设置您的网络接口选项。

```
# hammer host interface update --host "test3" --managed true \
--primary true --provision true
```

使用 `hammer bootdisk host` 命令从 Satellite 服务器下载引导磁盘：

- 对于主机镜像：

```
# hammer bootdisk host --host test3.example.com
```

- 对于完整主机映像：

```
# hammer bootdisk host --host test3.example.com --full true
```

- 对于通用镜像：

```
# hammer bootdisk generic
```

- 对于子网镜像：

```
# hammer bootdisk subnet --subnet subnetName
```

这会为您的主机创建一个引导 ISO。

如果需要，使用 `dd` 实用程序或 `livecd-tools` 将 ISO 写入 USB 存储设备。

当您启动物理主机并从 ISO 或 USB 存储设备引导时，主机将连接到卫星服务器并从其 `kickstart` 树开始安装操作系统。

安装完成后，主机还使用激活密钥注册到卫星服务器，并从 Satellite Tools 6.9 存储库安装必要的配置和管理工具。

6.5. 使用 UEFI HTTP 引导置备创建主机

您可以使用 UEFI HTTP 引导从 Satellite 置备主机。这是您可以在 IPv6 网络中置备主机的唯一方法。

前提条件

- 确保您满足 HTTP 启动的要求。有关更多信息，请参阅规划卫星中的 [HTTP 引导要求](#)。

流程

1. 在您置备的胶囊中，将 `grub2-efi` 软件包更新至最新版本：

```
# satellite-maintain packages install grub2-efi
```

2. 启用 `foreman-proxy-http`、`foreman-proxy-httpboot` 和 `foreman-proxy-tftp` 功能。

```
# satellite-installer --scenario satellite \  
--foreman-proxy-httpboot true \  
--foreman-proxy-http true \  
--foreman-proxy-tftp true
```

3. 在 **Satellite Web UI** 中，导航到 **Hosts > Create Host**。

4. 在 **Name** 字段中输入主机的名称。

5. 点击 **Organization** 和 **Location** 选项卡，并更改上下文以符合您的要求。

6. 从 **Host Group** 列表中，选择您要用来填充表单的主机组。

7. 点 **Interface** 选项卡，在主机的接口上点 **Edit**。

8. 验证字段是否填充了值。特别是：

- **Host** 选项卡中的 **Name** 将成为 **DNS** 名称。

- 卫星服务器会自动为新主机分配 **IP** 地址。

9. 在 **MAC** 地址字段中，输入主机的置备接口的 **MAC** 地址。这样可确保在 **PXE** 引导过程中识别主机。

10. 确保卫星服务器自动为主机上的第一个接口选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。
11. 点 **OK** 保存。要添加其他接口，请单击 **Add Interface**。您只能将一个接口用于 **Provision** 和 **Primary**。
12. 单击 **Operating System** 选项卡，然后验证所有字段是否包含值。确认操作系统的各个方面。
13. 从 **PXE Loader** 列表中，选择 **Grub2 UEFI HTTP**。
14. 可选：点 **Provisioning** 模板中的 **Resolve** 来检查新主机可识别要使用的正确调配模板。

有关关联调配模板的更多信息，请参阅 [第 2.13 节“创建调配模板”](#)。
15. 单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。
16. 单击 **Submit** 以保存主机详情。

有关网络接口的更多信息，请参阅添加 [网络接口](#)。
17. 将主机设置为从网络在 **UEFI** 模式中引导。
18. 启动主机。
19. 在引导菜单中，选择 **Kickstart** 默认 **PXEGrub2**。

这会创建主机条目和相关置备设置。这也包括为 **UEFI** 引导裸机主机创建所需的目录和文件。当您启动物理主机并将其引导模式设置为 **UEFI HTTP** 时，主机会检测定义的 **DHCP** 服务，接收带有 **Kickstart** 树胶囊的 **HTTP** 端点，并安装操作系统。

安装完成后，主机还使用激活密钥注册到卫星服务器，并从 **Satellite Tools 6.9** 存储库安装必要的配置和管理工具。

对于 **CLI** 用户

1. 在您置备的胶囊中，将 **grub2-efi** 软件包更新至最新版本：

```
# satellite-maintain packages install grub2-efi
```

2. 启用 **foreman-proxy-http**、**foreman-proxy-httpboot** 和 **foreman-proxy-tftp true** 功能。

```
# satellite-installer --scenario satellite \  
--foreman-proxy-httpboot true \  
--foreman-proxy-http true \  
--foreman-proxy-tftp true
```

3. 使用 **hammer host create** 命令创建主机。

```
# hammer host create --name "My_Host" \  
--organization "My_Organization" \  
--location "My_Location" \  
--hostgroup "My_Host_Group" \  
--mac "aa:aa:aa:aa:aa:aa" \  
--build true \  
--enabled true \  
--managed true \  
--pxe-loader "Grub2 UEFI HTTP"
```

4. 使用 **hammer 主机接口更新** 命令，确保设置了网络接口选项。

```
# hammer host interface update --host "My_Host" \  
--managed true \  
--primary true \  
--provision true
```

5. 将主机设置为从网络在 **UEFI** 模式中引导。

6. 启动主机。

7.

在引导菜单中，选择 **Kickstart 默认 PXEGrub2**。

这会创建主机条目和相关置备设置。这也包括为 UEFI 引导裸机主机创建所需的目录和文件。当您启动物理主机并将其引导模式设置为 UEFI HTTP 时，主机会检测定义的 DHCP 服务，接收带有 Kickstart 树胶囊的 HTTP 端点，并安装操作系统。

安装完成后，主机还使用激活密钥注册到卫星服务器，并从 Satellite Tools 6.9 存储库安装必要的配置和管理工具。

6.6. 在置备过程中部署 SSH 密钥

在置备过程中，使用此流程部署添加到用户的 SSH 密钥。有关向用户添加 SSH 密钥的详情，请参阅在 [管理 Red Hat Satellite 中的用户管理 SSH 密钥](#)。

流程

要在置备过程中部署 SSH 密钥，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 Hosts > Provisioning Templates。
2. 创建调配模板或克隆并编辑现有模板。更多信息请参阅 [第 2.13 节“创建调配模板”](#)。
3. 在模板中，单击 **Template** 选项卡。
4. 在 **Template 编辑器** 字段中，将 `create_users` 片断添加到 `%post` 部分：

```
<%= snippet('create_users') %>
```
5. 选择 **Default** 复选框。
6. 点 **关联** 选项卡。

7. 从 **Application Operating Systems** 列表中，选择一个操作系统。
8. 单击 **Submit** 以保存调配模板。
9. 创建与调配模板关联的主机，或者使用与修改后的模板关联的操作系统重建主机。有关更多信息，请参阅管理 [主机指南中的创建 主机](#)。

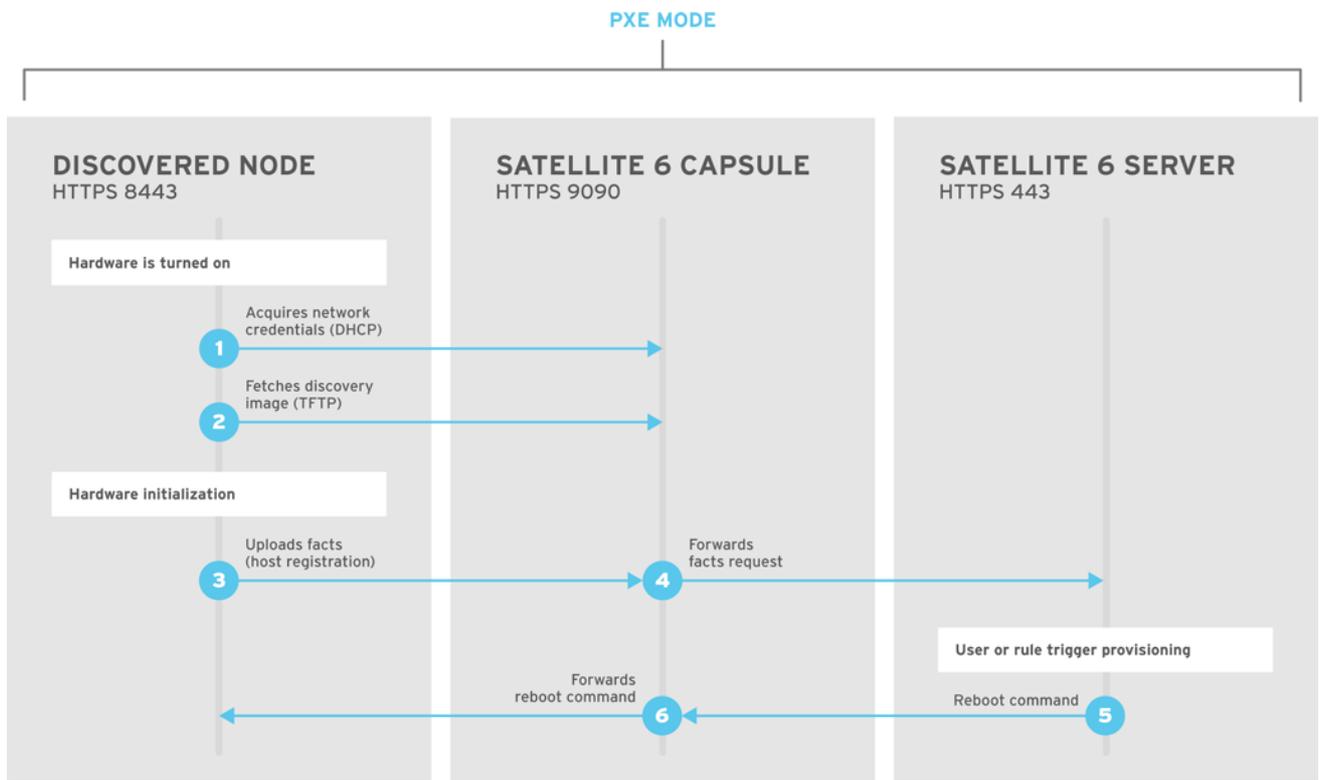
在置备过程中执行 `create_users` 代码片段时，会自动添加用户的 SSH 密钥。您可以将 **Owned** 设置为单独的用户或用户组。如果您将 **Owned** 设置为用户组，则用户组中所有用户的 SSH 密钥将自动添加。

第 7 章 配置发现服务

Red Hat Satellite 可以检测网络上不在 Satellite 清单中的主机。这些主机引导执行硬件检测的发现镜像，并将此信息转发回卫星服务器。此方法在卫星服务器中创建一个可供调配的主机列表，而无需输入每个主机的 MAC 地址。

当您引导空白裸机主机时，引导菜单有两个选项：**local** 和 **discovery**。如果您选择了 **Discovery** 来引导发现镜像，几分钟后，发现镜像会完成引导并显示状态屏幕。

默认情况下，卫星服务器上启用了 **Discovery** 服务。但是，全局模板的默认设置是从本地硬盘引导的。要更改默认设置，在 **Satellite Web UI** 中，导航到 **Administer > Settings**，然后点 **Provisioning** 选项卡。找到 **Default PXE** 全局模板条目行，然后在 **Value** 列中输入 **discovery**。



SATELLITE6_376339_1115

要使用 **Satellite** 服务器提供发现镜像，请安装以下 **RPM** 软件包：

- **tfm-rubygem-foreman_discovery**
- **foreman-discovery-image**

- **tfm-rubygem-smart_proxy_discovery**

tfm-rubygem-foreman_discovery 软件包包含用于处理发现的节点、连接和必要的数据库结构以及 API 的卫星插件。

foreman-discovery-image 软件包将发现 ISO 安装到 `/usr/share/foreman-discovery-image/` 目录中。您可以使用 `livecd-iso-to-pxeboot` 工具从这个 ISO 构建 PXE 引导镜像。该工具将这个 PXE 引导镜像保存到 `/var/lib/tftpboot/boot` 目录中。更多信息请参阅 [第 7.8 节“构建发现镜像”](#)。

tfm-rubygem-smart_proxy_discovery 软件包配置胶囊服务器，如集成卫星服务器的胶囊，以充当发现服务的代理。

安装完成后，您可以通过导航到 **Hosts > Discovered Hosts** 来查看新菜单选项。

7.1. 安装 DISCOVERY 服务

完成以下步骤，在胶囊服务器上启用发现服务。

流程

1. 在胶囊服务器上输入以下命令：

```
# satellite-maintain packages install foreman-discovery-image tfm-rubygem-smart_proxy_discovery
```

2. 重启 **satellite-maintain** 服务：

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

3. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Capsule**。

4. 单击胶囊服务器，再从 **Actions** 列表中选择 **Refresh**。在功能列表中找到 **发现**，以确认发现服务现在正在运行。

子网

带有可发现主机的所有子网都需要选择适当的胶囊服务器来提供发现服务。

要检查这一点，请导航到 **Infrastructure > Capsules**，并验证您要使用的胶囊服务器是否列出了 **Discovery** 功能。如果没有，请单击 **Refresh features**。

在 **Satellite Web UI** 中，导航到 **Infrastructure > Subnets**，选择子网，点 **Capsules** 选项卡，然后选择要使用的 **Discovery Proxy**。针对您要使用的每个子网执行此操作。

7.2. PROVISIONING TEMPLATE PXELINUX DISCOVERY SNIPPET

对于 **BIOS** 调配，**Hosts > Provisioning Templates** 窗口中的 **PXELinux** 全局默认模板包含代码片段 `pxelinux_discovery`。该片段包括以下几行：

```

LABEL discovery
MENU LABEL Foreman Discovery Image
KERNEL boot/fdi-image/vmlinuz0
APPEND initrd=boot/fdi-image/initrd0.img rootflags=loop root=live:/fdi.iso rootfstype=auto ro
rd.live.image acpi=force rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset
proxy.url=<%= foreman_server_url %> proxy.type=foreman
IPAPPEND 2

```

KERNEL 和 **APPEND** 选项引导发现镜像和 **ramdisk**。**APPEND** 选项包含一个 `proxy.url` 参数，其 `foreman_server_url` 宏作为其参数。此宏解析到卫星服务器的完整 URL。

对于 **UEFI** 调配，**Hosts > Provisioning Templates** 窗口中的 **PXEgrub2** 全局默认模板包含代码片段 `pxegrub2_discovery`：

```

menuentry 'Foreman Discovery Image' --id discovery {
  linuxefi boot/fdi-image/vmlinuz0 rootflags=loop root=live:/fdi.iso rootfstype=auto ro rd.live.image
acpi=force rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset proxy.url=<%=
foreman_server_url %> proxy.type=foreman BOOTIF=01-$mac
  initrdefi boot/fdi-image/initrd0.img
}

```

要使用胶囊来代理发现步骤，请编辑 `/var/lib/tftpboot/pxelinux.cfg/default` 或 `/var/lib/tftpboot/grub2/grub.cfg`，并将 URL 更改为您要使用的胶囊服务器 FQDN。

全局模板可用于卫星服务器，以及启用了 TFTP 功能的所有胶囊。

7.3. DISCOVERED HOSTS 的自动上下文

卫星服务器会根据以下规则序列为发现的主机分配机构和位置：

1. 如果发现的主机使用卫星中定义的子网，则主机将使用与子网关联的第一个组织和位置。
2. 如果设置了 `discovery_organization` 或 `discovery_location` 事实值，则发现的主机将使用这些事实值作为组织和位置。要设置这些事实值，请导航到 **Administer > Settings > Discovered**，并将这些事实添加到 **Default organization** 和 **Default location** 字段中。确保发现的主机的子网也属于事实设置的组织和位置，否则卫星出于安全原因而拒绝设置。
3. 如果没有上述条件，卫星会按名称分配第一个组织和位置。

您可以使用 **Discovered hosts** 页面的批量操作菜单来更改机构或位置。选择要修改的发现主机，然后从 **Select Action** 菜单中选择 **Assign Organization or Assign Location**。

请注意，`foreman_organization` 和 `foreman_location` 事实不再是为发现的主机分配上下文的有效值。您仍然可以使用这些事实来配置 Puppet 运行的主机。为此，请导航到 **Administer > Settings > Puppet** 部分，并将 `foreman_organization` 和 `foreman_location` 事实添加到 **Default organization** 和 **Default 位置** 字段。

7.4. 发现模板和 SNIPPETS 设置

要使用 **Discovery** 服务，您必须配置置备设置，将 **Discovery** 设为默认的服务，并设置要使用的模板。

将发现服务设置为默认

对于 **BIOS** 和 **UEFI**，将发现服务设置为当前 **Satellite** 清单中不存在的主机的默认服务，请完成以下步骤：

1. 在 **Satellite Web UI** 中，导航到 **Administer > Settings** 并点 **Provisioning** 选项卡。
2. 对于 **Default PXE** 全局模板条目，在 **Value** 列中输入，输入 **discovery**。

要使用模板，在 **Satellite Web UI** 中导航到 **Administer > Settings** 并点 **Provisioning** 选项卡，并设置要使用的模板。

自定义模板和片段

模板和代码片段被锁定，以防止更改。如果要编辑模板或代码段，克隆它，使用唯一名称保存，然后编辑克隆。

当您更改其包含的模板或代码段时，必须将更改传播到卫星服务器的默认 **PXE** 模板。

在 **Satellite Web UI** 中，导航到 **Hosts > Provisioning Templates**，点 **Build PXE Default**。

这会在卫星服务器上刷新默认 **PXE** 模板。

附加设置

proxy.url 参数

在卫星安装过程中，如果您使用默认选项 `--enable-foreman-plugin-discovery`，您可以编辑模板中的 `proxy.url` 参数来设置提供发现服务的胶囊服务器的 URL。您可以将 `proxy.url` 参数更改为要使用的另一个置备胶囊的 IP 地址或 FQDN，但请确保附加端口号，例如 `9090`。如果您在卫星安装期间使用带有 `--foreman-proxy-ssl-port` 选项的替代端口号，您必须添加该端口号。您还可以编辑 `proxy.url` 参数，以使用卫星 IP 地址或 FQDN，以便发现的主机直接与卫星服务器通信。

proxy.type 参数

如果您将胶囊服务器 FQDN 用于 `proxy.url` 参数，请确保将 `proxy.type` 参数设置为 `proxy`。如果使用 **Satellite FQDN**，请将 `proxy.type` 参数更新为 `foreman`。

```
proxy.url=https://capsule.example.com:9090 proxy.type=proxy
```

渲染胶囊的主机名称

卫星 6 将相同的模板部署到所有 TFTP 胶囊，并且没有可用于渲染胶囊的主机名的变量或宏。硬编码的 `proxy.url` 无法用于两个或多个 TFTP 胶囊。作为临时解决方案，每次点击 **Build PXE Defaults** 时，使用 **SSH** 编辑 TFTP 目录中的配置文件，或者为适当的子网使用通用 DNS 别名。

标记的 VLAN 置备

如果要使用标记的 VLAN 置备，并且希望发现服务发送发现请求，请在发现模板中的 **KERNEL** 选项中添加以下信息：

```
fdi.vlan.primary=example_VLAN_ID
```

7.5. 从发现的主机创建主机

调配发现的主机遵循与 PXE 调配类似的调配过程。主要区别在于，您可以从发现的主机列表中选择要调配的主机，而不是手动输入主机的 MAC 地址。

前提条件

- 在卫星上配置域和子网。有关网络要求的详情请参考 [第 3 章 配置网络](#)。
- 在卫星上配置发现服务。更多信息请参阅 [第 7.1 节 “安装 Discovery 服务”](#)。
- 裸机主机或空白虚拟机。
- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。

有关无人值守和无 PXE 的置备安全令牌的详情，请参考 [第 6.2 节 “配置安全令牌有效期持续时间”](#)。

流程

要从发现的主机创建主机，请完成以下步骤：

1. 在 Satellite Web UI 中，进入 Hosts > Discovered host。选择要使用的主机，然后单击列表右侧的 Provision。
2. 从以下两个选项之一中选择：
 - 要从主机组中调配主机，请选择主机组、组织和位置，然后单击 Create Host。
 -

要置备带有进一步自定义的主机，请单击 **Customize Host** 并输入您要为新主机指定的附加详情。

3.

验证字段是否填充了值。特别是：

- **Host** 选项卡中的 **Name** 将成为 **DNS** 名称。
- 卫星服务器会自动为新主机分配 **IP** 地址。
- 卫星服务器从发现结果自动填充 **MAC** 地址。

4.

确保卫星服务器自动为主机上的第一个接口选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。

5.

单击 **Operating System** 选项卡，然后验证所有字段是否包含值。确认操作系统的各个方面。

6.

单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。主机必须解析为以下调配模板：

- **kexec Template: Discovery Red Hat kexec**
- **Provisioning Template: Satellite Kickstart Default**

有关关联调配模板的更多信息，请参阅 [第 2.11 节“调配模板”](#)。

7.

单击 **Submit** 以保存主机详情。

主机调配完成后，发现的主机会变成内容主机。要查看主机，请导航到 **Hosts > Content Hosts**。

对于 **CLI** 用户

1. 识别用于置备的发现主机：

```
# hammer discovery list
```

2. 选择主机并使用主机组进行调配。使用 `--new-name` 选项设置新主机名：

```
# hammer discovery provision --name "host_name" \
--new-name "new_host_name" --organization "My_Organization" \
--location "My_Location" --hostgroup "My_Host_Group" --build true \
--enabled true --managed true
```

这将从发现的主机列表中删除主机，并创建具有调配设置的主机条目。发现镜像会自动重置主机，以便它能够引导到 PXE。主机在卫星服务器的集成胶囊上检测到 DHCP 服务，并开始安装操作系统。其余的进程与第 6.3 节“使用未置备置备创建主机”中描述的正常 PXE 工作流相同。

7.6. 创建发现规则

作为为发现的主机自动化调配流程的方法，卫星提供了相应的功能来创建发现规则。这些规则定义了发现的主机根据分配的主机组自动置备自身的方式。例如，您可以自动置备具有高 CPU 数量为虚拟机监控程序的主机。同样，您可以将使用大硬盘作为存储服务器的主机调配。

NIC 注意事项

自动置备目前不允许配置 NIC；所有系统都被置备使用在发现过程中检测到的 NIC 配置。但是，您可以使用脚本或在 kickstart scriptlet 中设置 NIC，或使用后续部分中的配置管理。

流程

要创建规则，请完成以下步骤：

1. 在 Satellite Web UI 中，进入 **Configure > Discovery rules**，然后选择 **Create Rule**。
2. 在 **Name** 字段中输入规则的名称。
3. 在 **Search** 字段中输入用于决定是否置备主机的规则。此字段提供您输入的值的建议，并允许操作员输入多个规则。例如：`cpu_count > 8`。

4. 从 **Host Group** 列表中，选择要用作此主机的主机组。
5. 在 **Hostname** 字段中，输入用于确定多个主机的主机名称的模式。这将使用与调配模板相同的 **ERB** 语法。主机名可以使用特定于主机的值的 **@host** 属性，而 **rand** 函数可用于随机数。有关调配模板的更多信息，请参阅 [第 2.11 节“调配模板”](#)。
 - `myhost-<%= rand(99999) %>`
 - `abc-<%= @host.facts['bios_vendor'] %>-<%= rand(99999) %>`
 - `xyz-<%= @host.hostgroup.name %>`
 - `srv-<%= @host.discovery_rule.name %>`
 - `server-<%= @host.ip.gsub('.', '-') + '-' + @host.hostgroup.subnet.name %>`

在创建主机名模式时，请确保生成的主机名是唯一的，不要以数字开头，且不会包含下划线或点。最好使用 **Facter** 提供的唯一信息，如 **MAC** 地址、**BIOS** 或串行 **ID**。
6. 在 **Hosts limit** 字段中，输入您可以使用该规则置备的最大主机数量。为无限输入 **0**。
7. 在 **Priority** 字段中，输入要设置优先级的数字，规则优先于其他规则。较低值的规则具有更高的优先级。
8. 从 **Enabled** 列表中，选择是否要启用该规则。
9. 要为规则设置不同的调配上下文，请单击 **Organizations** 和 **Locations** 选项卡，再选择您要使用的上下文。
10. 单击 **Submit** 以保存您的规则。

11.

进入 **Hosts > Discovered Host** 并选择以下两个选项之一：

- 从右侧的发现主机列表中，选择 **Auto-Provision** 以自动调配单个主机。
- 在窗口的右上角，单击 **Auto-Provision All** 以自动调配所有主机。

对于 **CLI 用户**

使用 **hammer discovery-rule create** 命令创建规则：

```
# hammer discovery-rule create --name "Hypervisor" \
--search "cpu_count > 8" --hostgroup "My_Host_Group" \
--hostname "hypervisor-<%= rand(99999) %>" \
--hosts-limit 5 --priority 5 --enabled true
```

使用 **hammer discovery auto-provision** 命令自动置备主机：

```
# hammer discovery auto-provision --name "macabcdef123456"
```

7.7. 实施 PXE 无发现

卫星提供了无 PXE 的发现服务，无需基于 PXE 的服务(DHCP 和 TFTP)。您可以使用卫星服务器的发现镜像完成此操作。为安装调度发现的节点后，它使用 **kexec** 命令在不重新引导节点的情况下使用 OS 安装程序重新载入 Linux 内核。



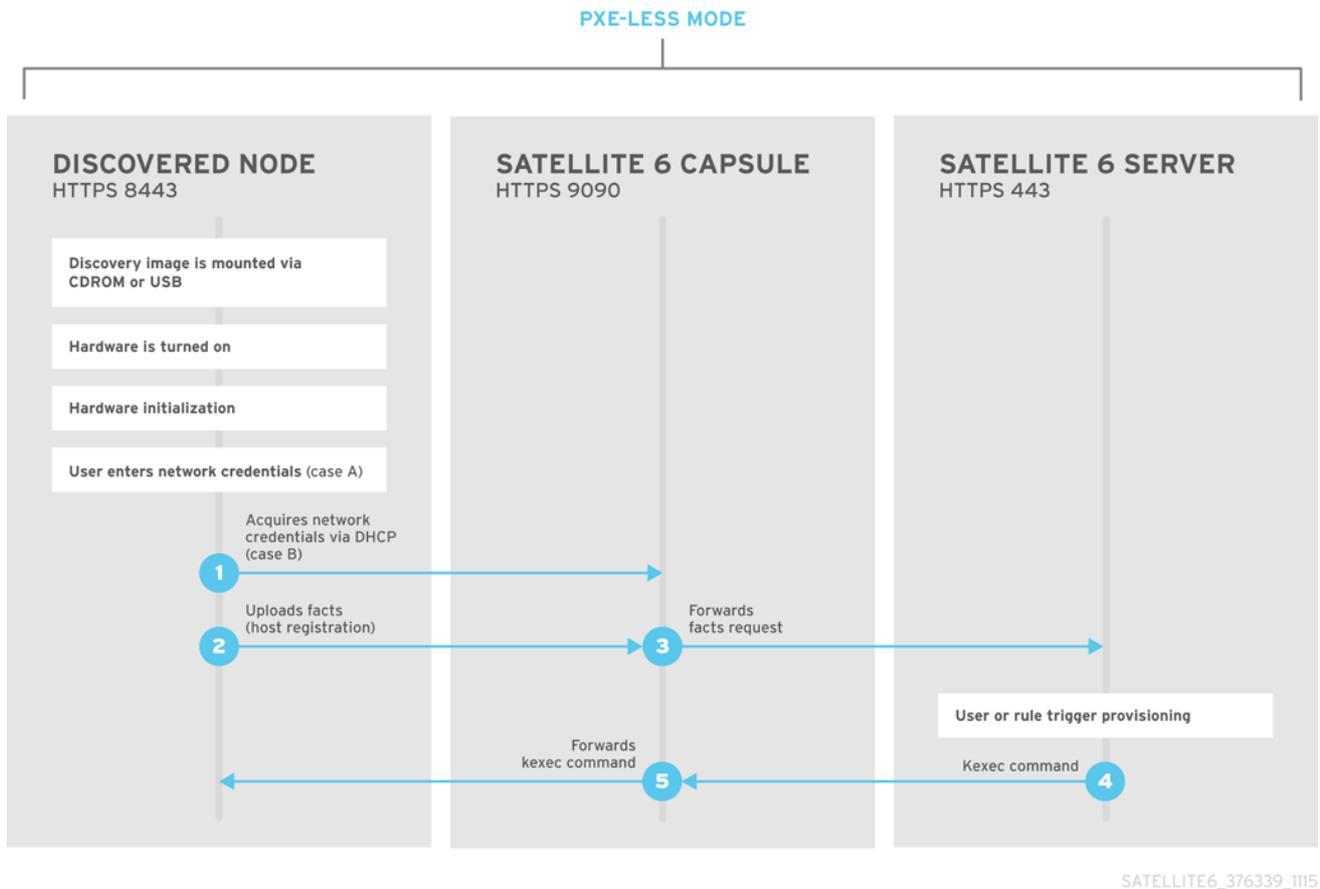
重要

Discovery kexec 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的详情，请参考

<https://access.redhat.com/support/offerings/techpreview/>。

已知问题

控制台在升级过程中可能会停滞。在某些硬件中，您可能会遇到图形硬件问题。



如果您还没有安装 **Discovery 服务或镜像**，请参阅 [第 7.8 节“构建发现镜像”](#)。

Discovery 服务的 ISO 驻留在 `/usr/share/foreman-discovery-image/`，并使用 `foreman-discovery-image` 软件包安装。

已参加的使用

1. 将此介质复制到 **CD、DVD 或者 USB 记忆棒**。例如，要复制到位于 `/dev/sdb` 的 **USB 记忆棒**：

```
# dd bs=4M \
if=/usr/share/foreman-discovery-image/foreman-discovery-image-3.4.4-5.iso \
of=/dev/sdb
```

2. 将发现引导介质插入到裸机主机，启动主机并从介质引导。**Discovery Image** 显示 **手动网络设置** 或使用 **DHCP 发现** 的选项：

如果您选择 **Manual network setup**，发现镜像会请求一组网络选项。这包括连接到卫星服务器的主要网络接口。此发现镜像还会要求提供网络接口配置选项，如 **IPv4 地址、IPv4 网关和 IPv4 DNS 服务器**。

3. 输入这些详细信息后，选择 下一步。
4. 如果选择 **Discovery with DHCP**，则发现镜像仅请求连接到卫星服务器的主要网络接口。它尝试使用 DHCP 服务器自动配置网络接口，例如胶囊服务器提供的一个网络接口。
5. 在主接口配置后，发现镜像会请求 服务器 URL，这是提供发现服务的卫星服务器或胶囊服务器的 URL。例如，要在 Satellite 服务器上使用集成胶囊，请使用以下 URL：

```
https://satellite.example.com:9090
```
6. 将 连接类型 设置为 **Proxy**，然后选择 **Next**。
7. 可选：**Discovery** 镜像还提供一组字段，用于为 **Facter** 工具输入 自定义事实，以转发回卫星服务器。它们以 **name-value** 格式输入。提供您需要的任何自定义事实，然后选择 **Confirm** 以继续。

卫星报告了与卫星服务器的发现服务成功通信。进入 **Hosts > Discovered Hosts** 并查看新发现的主机。

有关置备发现的主机的更多信息，请参阅 [第 7.5 节“从发现的主机创建主机”](#)。

7.8. 构建发现镜像

发现镜像是一个最小的操作系统，在主机上进行 PXE 引导以获取初始硬件信息，并使用 **Satellite** 签入。发现的主机会继续运行 **Discovery** 镜像，直到被重启到 **Anaconda** 中，然后启动置备过程。

操作系统镜像基于红帽企业 Linux 7。

foreman-discovery-image 软件包包含此镜像。您必须在提供 TFTP 服务的胶囊上安装该软件包。

如果您更改配置文件，请使用此流程构建 **Satellite** 发现镜像或重新构建镜像。

请勿在您的生产卫星或胶囊中使用此步骤。使用专用环境，或者将同步的存储库和 **kickstart** 文件复制

到单独的服务器。

前提条件

- **安装 livecd-tools 软件包：**

```
# satellite-maintain packages install livecd-tools
```

- 对于构建发现镜像所需的以下 Red Hat Enterprise Linux 7 软件仓库，请将下载策略更改为 **Immediate**。这是必要的，因为卫星仅在与即时下载策略同步存储库期间下载所有软件包。
 - **最新的 Red Hat Enterprise Linux 7 服务器(Kickstart) 软件仓库。例如， Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.7。**
 - **红帽卫星胶囊 6.9（适用于 RHEL 7 服务器） RPM x 86_64.**

有关更改下载策略的更多信息，请参阅 [内容管理指南中的更改存储库下载策略](#)。

- **同步构建发现镜像所需的以下 Red Hat Enterprise Linux 7 软件仓库：**
 - **最新的 Red Hat Enterprise Linux 7 服务器(Kickstart) 软件仓库。例如， Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.7。**
 - **红帽卫星胶囊 6.9（适用于 RHEL 7 服务器） RPM x 86_64.**

有关同步存储库的更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。

流程

要构建 Satellite 发现镜像，请完成以下步骤：

1. **打开 `/usr/share/foreman-discovery-image/foreman-discovery-image.ks` 文件进行编辑：**

```
# vim /usr/share/foreman-discovery-image/foreman-discovery-image.ks
```

2.

将 `kickstart` 文件中的存储库行替换为存储库 URL :

```
repo --name=rhel --
baseurl=file:///var/lib/pulp/published/yum/https/repos/Default_Organization/Library/content/dist/r
hel/server/7/7.7/x86_64
repo --name=sat --
baseurl=file:///var/lib/pulp/published/yum/https/repos/Default_Organization/Library/content/dist/r
hel/server/7/7Server/x86_64/sat-capsule/6.9/os
```

3.

运行 `livecd-creator` 工具 :

```
# livecd-creator --title="Discovery-Image" \
--compression-type=xz \
--cache=var/cache/build-fdi \
--config /usr/share/foreman-discovery-image/foreman-discovery-image.ks \
--fslabel fdi \
--tmpdir /var/tmp
```

如果您在 `--fslabel` 选项中更改 `fdi`，在载入镜像时，还必须更改内核命令行中的 `root` 标签。 `fdi` 或备用名称附加到作为此流程一部分创建的 `.iso` 文件中。当从 `.iso` 转换为 PXE 时，PXE 发现工具使用此名称。

使用 `/var /tmp`，因为如果系统在 `swap` 空间较低，这个过程需要接近 3GB 空间，因此可能出现的问题。

4.

验证您的 `fdi.iso` 文件是否已创建 :

```
# ls -h *.iso
```

在创建 `.iso` 文件时，您可以通过网络或本地引导 `.iso` 文件。完成以下步骤之一。

要通过网络引导 `iso` 文件 :

1.

要通过网络从 `.iso` 文件中提取初始 `ramdisk` 和内核文件，请输入以下命令 :

```
# discovery-iso-to-pxe fdi.iso
```

2. **创建用于存储您的引导文件的目录：**

```
# mkdir /var/lib/tftpboot/boot/myimage
```
3. **将 `initrd0.img` 和 `vmlinuz0` 文件复制到新目录。**
4. **编辑 `/var/lib/tftpboot/pxelinux.cfg` 文件中的 `KERNEL` 和 `APPEND` 条目，以添加您自己初始 `ramdisk` 和内核文件的信息。**

在本地引导 iso 文件：

如果要为本地引导创建混合 .iso 文件，请完成以下步骤：

1. **要将 .iso 文件转换为 .iso 混合文件进行 PXE 调配，请输入以下命令：**

```
# isohybrid --partok fdi.iso
```

如果您安装了 `grub2` 软件包，您可以使用以下命令安装 `grub2` 引导装载程序：

```
# isohybrid --partok --uefi fdi.iso
```

2. **要在 .iso 文件中添加 `md5 checksum`，以便它在 `Satellite` 中传递安装介质验证测试，请输入以下命令：**

```
# implantisomd5 fdi.iso
```

7.9. 无人值守的 `USE`、`CUSTOMIZATION` 和 `IMAGE REMASTERING`

您可以创建自定义发现 ISO，以在引导后自动执行镜像配置过程。发现镜像对操作系统使用 Linux 内核，它会传递内核参数来配置发现服务。这些内核参数包括以下条目：

`proxy.url`

提供 `Discovery` 服务的胶囊服务器或卫星服务器的 URL。

`proxy.type`

代理类型。这通常设置为用于连接到胶囊服务器的代理。此参数还支持传统的 `foreman` 选项，其中通信直接发送到卫星服务器，而非胶囊服务器。

`fdi.pxmac`

主接口的 MAC 地址，格式为 `AA:BB:CC:DD:EE:FF`。这是您要用于与胶囊服务器通信的接口。在自动模式中，使用第一个 NIC（按字母顺序使用网络标识符）使用链接。在半自动化模式中，会出现一个屏幕并要求您选择正确接口。

`fdi.pxip, fdi.pxgw, fdi.pxdns`

为主网络接口手动配置 IP 地址(`fdi.pxip`)、**网关**(`fdi.pxgw`)和 **DNS** (`fdi.pxdns`)。如果省略这些参数，镜像使用 DHCP 来配置网络接口。

`fdi.pxfactname1, fdi.pxfactname2 ... fdi.pxfactnameN`

使用 指定自定义事实名称。

`fdi.pxfactvalue1, fdi.pxfactvalue2 ... fdi.pxfactvalueN`

每个自定义事实的值。每个值对应一个事实名称。例如，`Fdi.pxfactvalue1` 使用 `fdi.pxfactname1` 设置事实的值。

`fdi.pxauto`

设置自动或半自动模式。如果设置为 0，则镜像使用半自动模式，该模式允许您通过一组对话框选项确认您的选择。如果设置为 1，则镜像使用自动模式，并在没有任何确认的情况下继续操作。

`fdi.initnet`

默认情况下，镜像会初始化所有网络接口（值为 `all`）。当此设置设置为 `bootif` 时，只有从中启动它的网络接口才会被初始化。

`fdi.rootpw`

默认情况下，root 帐户被锁定。使用这个选项设置 root 密码。您可以同时输入明文的密码。

`fdi.ssh`

默认情况下禁用 SSH 服务。将其设置为 1 或 `true` 以启用 SSH 访问。

`fdi.ipv4.method`

默认情况下，NetworkManager IPv4 方法设置为 `auto`。这个选项覆盖它，将其设置为 `ignore` 以禁用 IPv4 堆栈。这个选项只适用于 DHCP 模式。

`fdi.ipv6.method`

默认情况下，NetworkManager IPv6 方法设置为 `auto`。这个选项覆盖它，将其设置为 `ignore` 以禁

用 IPv6 堆栈。这个选项只适用于 DHCP 模式。

使用 `discovery-remaster` 工具恢复操作系统镜像

卫星服务器在 `foreman-discovery-image` 软件包中提供 `discovery-remaster` 工具。此工具调整镜像以包括这些内核参数。要重新管理镜像，请运行 `discovery-remaster` 工具。例如：

```
# discovery-remaster ~/iso/foreman-discovery-image-3.4.4-5.iso \
"fdi.pxip=192.168.140.20/24 fdi.pxgw=192.168.140.1 \
fdi.pxdns=192.168.140.2 proxy.url=https://satellite.example.com:9090 \
proxy.type=proxy fdi.pxfactname1=customhostname fdi.pxfactvalue1=myhost \
fdi.pxmac=52:54:00:be:8e:8c fdi.pxauto=1"
```

将此介质复制到 CD、DVD 或者 USB 记忆棒。例如，要复制到位于 `/dev/sdb` 的 USB 记忆棒：

```
# dd bs=4M \
if=/usr/share/foreman-discovery-image/foreman-discovery-image-3.4.4-5.iso \
of=/dev/sdb
```

将发现引导介质插入到裸机主机，启动主机并从介质引导。

有关置备发现的主机的更多信息，请参阅 [第 7.5 节“从发现的主机创建主机”](#)。

7.10. 扩展发现镜像

您可以使用自定义事实、软件或设备驱动程序来扩展卫星发现镜像。您还可以提供一个压缩的存档文件，其中包含供镜像使用的额外代码。

流程

1. 创建以下目录结构：

```
.
├── autostart.d
│   └── 01_zip.sh
├── bin
│   └── ntpdate
├── facts
│   └── test.rb
└── lib
```

```

├── libcrypto.so.1.0.0
├── ruby
└── test.rb

```

- **autostart.d** 目录包含在系统启动时由 POSIX 顺序执行的脚本，但在主机注册到卫星之前。

- **bin** 目录添加到 `$PATH` 变量中，您可以将二进制文件放在这个目录中，并在自动启动脚本中使用它们。

- **事实** 目录添加到 `FACTERLIB` 变量中，以便可以配置自定义事实并发送到卫星。

- **lib** 目录添加到 `LD_LIBRARY_PATH` 变量，`lib/ruby` 被添加到 `RUBYLIB` 变量中，以便可以正确执行 `/bin` 中的二进制文件。

2.

创建目录结构后，使用以下命令创建一个 `.zip` 文件存档：

```
zip -r my_extension.zip .
```

3.

要告知发现它必须使用的扩展镜像，请将您的 `zip` 文件放在您的 TFTP 服务器上，使用发现镜像将 `APPEND` 行放在 `PXELinux` 模板的 `APPEND` 行，然后使用 `fdi.zips` 选项更新路径相对于 TFTP root 的路径。例如，如果您有两个归档，地址为 `$TFTP/zip1.zip` 和 `$TFTP/boot/zip2.zip`，则使用以下语法：

```
fdi.zips=zip1.zip,boot/zip2.zip
```

您可以将新指令和选项附加到现有环境变量中 (`PATH`、`LD_LIBRARY_PATH`、`RUBYLIB` 和 `FACTERLIB`)。如果要在脚本中明确指定路径，则将 `.zip` 文件内容提取到镜像的 `/opt/extension` 目录中。

您可以创建多个 `.zip` 文件，但请注意它们会提取到 Discovery 镜像上的同一位置。如果在以后的 `.zip` 文件中提取的文件具有相同的文件名，则会覆盖更早的版本。

7.11. 发现故障排除

如果 `Hosts > Discovered Hosts` 中的 Satellite web UI 中没有列出机器，请检查以下配置区域来帮助隔离错误：

- 导航到 **Hosts > Provisioning Templates**，再使用 **Build PXE Default** 按钮重新部署 default PXELinux 模板。
- 验证 TFTP 胶囊上的 `pxelinux.cfg/default` 配置文件。
- 确保主机、胶囊服务器和卫星服务器之间的适当的网络连接。
- 检查使用中的 PXELinux 模板，并确定它所包含的 PXE 发现代码片段。代码片段被命名如下：`pxelinux_discovery`、`pxegrub_discovery` 或 `pxegrub2_discovery`。验证 PXE 发现片断中的 `proxy.url` 和 `proxy.type` 选项。
- 确保已发现节点的 DNS 正常工作，或使用您正在使用的 PXE 发现代码段中的 `proxy.url` 选项中的 IP 地址。
- 确保 DHCP 服务器正确向引导的镜像提供 IP 地址。
- 确保发现的主机或虚拟机至少有 1200 MB 内存。较少的内存可能会导致各种随机内核 panic 错误，因为镜像被提取到内存中。

要收集重要系统事实，请使用 `discovery-debug` 命令。它将打印出系统日志、网络配置、事实列表以及其他有关标准输出的信息。典型的用例是重定向此输出结果，并使用 `scp` 命令复制该输出结果以进行进一步调查。

发现的主机上的第一个虚拟控制台预留给 `systemd` 日志。特别有用的系统日志被标记为如下：

- `discover-host` - 初始事实上传
- `Foreman-discovery` - 事实刷新，重启远程命令
- `nm-prepare` - 预配置 NetworkManager 的引导脚本



NetworkManager - 网络信息

使用 TTY2 或更高版本登录发现的主机。默认情况下，root 帐户和 SSH 访问被禁用，但您可以使用 APPEND 行中的 Default PXELinux 模板中的以下内核命令行选项启用 SSH 和设置 root 密码：

```
fdi.ssh=1 fdi.rootpw=My_Password
```

第 8 章 使用 RED HAT IMAGE BUILDER IMAGE 进行置备

在卫星中，您可以与红帽 Web 控制台集成，以执行操作并监控您的主机。使用 Red Hat Web 控制台，您可以访问 Red Hat Image Builder 并构建镜像，然后您可以上传到 HTTP 服务器并使用此镜像来置备主机。当您为 Satellite 配置为镜像置备时，Anaconda 安装程序分区磁盘，将镜像下载并安装到主机中。首选镜像类型是 TAR。

有关将 Red Hat Web 控制台与 Satellite 集成的更多信息，请参阅[管理主机指南中的使用 Red Hat Web 控制台进行主机管理和监控](#)。

前提条件

1. 通过 Red Hat Image Builder 创建的现有 TAR 镜像。

流程

要将 Red Hat Image Builder 镜像与 Satellite 中的 Anaconda Kickstart 搭配使用，请完成以下步骤：

1. 在卫星上，创建自定义产品，添加自定义文件存储库到该产品，并将镜像上传到存储库。有关更多信息，请参阅[内容管理指南中的导入单个 ISO 镜像和文件](#)。
2. 在 Satellite Web UI 中，进入 **Configure > Host Groups**，然后选择要使用的主机组。
3. 单击 **Parameters** 选项卡，然后单击 **Add Parameter**。
4. 在 **Name** 字段中输入 `kickstart_liveimg`。
5. 从 **Type** 列表中，选择 **字符串**。
6. 在 **Value** 字段中输入绝对路径或相对路径，格式为 `custom/product/repository/image_name`，指向存储镜像的确切位置。
7. 单击 **Submit** 保存您的更改。

您可以使用此镜像进行裸机置备，并使用计算资源进行置备。

有关裸机置备的更多信息，请参阅 [第 6 章 使用 PXE 调配主机](#)。有关使用不同计算资源进行调配的更多信息，请参阅您要使用的计算资源的相关章节。

第 9 章 在 KVM (LIBVIRT)上置备虚拟机

基于内核的虚拟机(KVM)使用开源虚拟化守护进程和称为 `libvirt` 的 API，在红帽企业 Linux 上运行。Red Hat Satellite 可以连接到 KVM 服务器上的 `libvirt` API，在虚拟机监控程序上配置主机并控制某些虚拟化功能。

您可以使用 KVM 置备通过网络连接或从现有镜像创建主机。

前提条件

- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。
- 在 KVM 服务器上管理网络的胶囊服务器。确保此网络中没有运行其他 DHCP 服务以避免与胶囊服务器冲突。有关胶囊服务器的网络服务配置的更多信息，请参阅 [第 3 章配置网络](#)。
- 运行 KVM 虚拟化工具(`libvirt` 守护进程)的 Red Hat Enterprise Linux 服务器。如需更多信息，请参阅 [Red Hat Enterprise Linux 7 虚拟化入门指南](#)。
- 如果要使用基于镜像的置备，则现有虚拟机镜像。确保此镜像存在于 KVM 主机上的存储池中。默认存储池通常位于 `/var/lib/libvirt/images` 中。只有目录池存储类型可以通过卫星进行管理。
- 可选：这些步骤中的示例将 `root` 用户用于 KVM。如果要在 KVM 服务器上使用非 `root` 用户，您必须将用户添加到 KVM 服务器上的 `libvirt` 组中：

```
useradd -a -G libvirt non_root_user
```
- 具有以下角色的 Satellite 用户帐户：
 - 编辑主机

- **查看主机**

有关更多信息，请参阅管理 Red Hat Satellite 指南中的 [将角色分配到用户](#)。

- **Satellite 中的自定义角色，具有以下权限：**

- **`view_compute_resources`**
- **`destroy_compute_resources_vms`**
- **`power_compute_resources_vms`**
- **`create_compute_resources_vms`**
- **`view_compute_resources_vms`**
- **`view_locations`**
- **`view_subnets`**

有关创建角色的更多信息，请参阅管理 Red Hat Satellite 指南中的 [创建角色](#)。https://access.redhat.com/documentation/en-us/red_hat_satellite/6.9/html-single/administering_red_hat_satellite/index#creating-a-role_admin 有关添加角色权限的更多信息，请参阅管理 Red Hat Satellite 指南中的 [角色添加权限](#)。

流程概述

1. [第 9.1 节 “为 KVM 连接配置 Satellite 服务器”](#)。
2. [第 9.2 节 “为 Satellite 服务器添加 KVM 连接”](#)。
3. 可选：[第 9.3 节 “在 Satellite 服务器中添加 KVM 镜像”](#)。如果要使用基于镜像的置备，请使

用这个步骤。

4. [第 9.4 节“在计算配置集中添加 KVM 详情”](#)。
5. [第 9.5 节“在 KVM 上创建主机”](#)。

9.1. 为 KVM 连接配置 SATELLITE 服务器

在添加 KVM 连接前，为 **foreman** 用户创建 SSH 密钥对，以确保卫星服务器和 KVM 之间的安全连接。

流程

1. 在卫星服务器上，切换到 **foreman** 用户：

```
# su foreman -s /bin/bash
```

2. 生成密钥对：

```
$ ssh-keygen
```

3. 将公钥复制到 KVM 服务器：

```
$ ssh-copy-id root@kvm.example.com
```

4. 退出 **foreman** 用户的 **bash shell**：

```
$ exit
```

5. 安装 **libvirt-client** 软件包：

```
# satellite-maintain packages install libvirt-client
```

6. 使用以下命令测试与 KVM 服务器的连接：

```
# su foreman -s /bin/bash -c 'virsh -c qemu+ssh://root@kvm.example.com/system list'
```

9.2. 为 SATELLITE 服务器添加 KVM 连接

使用这个步骤将 KVM 添加为 Satellite 中的计算资源。

流程

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources，点 Create Compute Resource。
2. 在 Name 字段中输入新计算资源的名称。
3. 从 Provider 列表中，选择 Libvirt。
4. 在 Description 字段中，输入计算资源的描述。
5. 在 URL 字段中，输入 KVM 服务器的连接 URL。例如：

```
qemu+ssh://root@kvm.example.com/system
```

6. 从 Display 类型 列表中，选择 VNC 或 Spice。
7. 可选：要使用随机生成密码保护新主机的控制台访问，请在 显示连接复选框上选择设置随机生成的密码。您可以检索 VNC 控制台的密码，从 KVM 服务器上执行的以下命令的输出结果访问客户机虚拟机控制台：

```
# virsh edit your_VM_name
<graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'
passwd='your_randomly_generated_password'>
```

每次打开虚拟机的控制台时，密码都会随机生成，例如使用 virt-manager。

8. 单击 Test Connection，以确保卫星服务器在没有故障的情况下连接到 KVM 服务器。

9. 验证 **Locations** 和 **Organizations** 选项卡已自动设置为您当前的上下文。如果需要，在这些标签页中添加额外的上下文。
10. 点 **Submit** 保存 KVM 连接。

对于 CLI 用户

- 要创建计算资源，请输入 **hammer compute-resource create** 命令：

```
# hammer compute-resource create --name "My_KVM_Server" \  
--provider "Libvirt" --description "KVM server at kvm.example.com" \  
--url "qemu+ssh://root@kvm.example.com/system" --locations "New York" \  
--organizations "My_Organization"
```

9.3. 在 SATELLITE 服务器中添加 KVM 镜像

要使用基于镜像的置备来创建主机，您必须将镜像的相关信息（如访问详细信息和镜像位置）添加到您的卫星服务器。

请注意，您只能通过 **Satellite 6** 管理目录池存储类型。

流程

1. 在 **Satellite Web UI** 中，进入 **Infrastructure > Compute Resources** 并点 **KVM 连接** 的名称。
2. 点 **Create Image**。
3. 在 **Name** 字段中输入镜像的名称。
4. 从 **Operating System** 列表中，选择镜像的基本操作系统。
5. 从 **Architecture** 列表中，选择操作系统架构。

6. 在 **Username** 字段中输入镜像访问的 **SSH 用户名**。这通常是 **root 用户**。
7. 在 **Password** 字段中，为镜像访问输入 **SSH 密码**。
8. 在 **Image path** 字段中，输入指向 **KVM 服务器上镜像的完整路径**。例如：


```
/var/lib/libvirt/images/TestImage.qcow2
```
9. 可选：如果镜像支持用户数据输入，如 **cloud-init 数据**，请选择 **User Data** 复选框。
10. 单击 **Submit** 以保存镜像详情。

对于 CLI 用户

- 使用 **hammer compute-resource image create** 命令创建镜像。使用 **--uuid** 字段将镜像位置的完整路径存储在 **KVM 服务器**上。

```
# hammer compute-resource image create \
--name "KVM Image" \
--compute-resource "My_KVM_Server" \
--operatingsystem "RedHat version" \
--architecture "x86_64" \
--username root \
--user-data false \
--uuid "/var/lib/libvirt/images/KVMimage.qcow2" \
```

9.4. 在计算配置集中添加 KVM 详情

使用这个步骤将 **KVM 硬件设置**添加到计算配置集中。当您使用此计算配置文件在 **KVM** 上创建主机时，这些设置会被自动填充。

流程

1. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Profiles**。
2. 在 **Compute Profiles** 窗口中，单击现有计算配置文件的名称，或者单击 **Create Compute Profile**，输入 **Name**，再单击 **Submit**。

3. 单击 **KVM 计算资源** 的名称。
4. 在 **CPU** 字段中，输入分配给新主机的 **CPU** 数量。
5. 在 **Memory** 字段中输入要分配给新主机的内存量。
6. 从 **Image** 列表中，选择在执行基于镜像置备时要使用的镜像。
7. 从 **Network Interfaces** 列表中，为主机网络接口选择网络参数。您可以创建多个网络接口。但是，至少有一个接口必须指向胶囊管理网络。
8. 在 **Storage** 区域，为主机输入存储参数。您可以为主机创建多个卷。
9. 单击 **Submit**，将设置保存到 **compute** 配置文件。

对于 CLI 用户

1. 运行以下命令来创建计算配置集：

```
# hammer compute-profile create --name "Libvirt CP"
```

2. 要为 **compute** 配置集添加值，请输入以下命令：

```
# hammer compute-profile values create --compute-profile "Libvirt CP" \
--compute-resource "My_KVM_Server" \
--interface
"compute_type=network,compute_model=virtio,compute_network=examplenetwork" \
--volume "pool_name=default,capacity=20G,format_type=qcow2" \
--compute-attributes "cpus=1,memory=1073741824"
```

9.5. 在 KVM 上创建主机

在 **Satellite** 中，您可以使用 **KVM** 置备通过网络连接或从现有镜像创建主机：

-

如果要通过网络连接创建主机，则新主机必须能够访问卫星服务器的集成胶囊或 KVM 虚拟网络上的外部胶囊服务器，以便主机可以访问 PXE 调配服务。这个新主机条目会触发 KVM 服务器创建和启动虚拟机。如果虚拟机通过虚拟网络检测到定义的胶囊服务器，虚拟机将启动到 PXE，并开始安装所选操作系统。

- 如果要使用现有镜像创建主机，则新主机条目会触发 KVM 服务器使用预先存在的镜像作为新卷的基础来创建虚拟机。

DHCP Conflicts

对于基于网络的调配，如果您使用 KVM 服务器上的虚拟网络进行置备，请选择不提供 DHCP 分配的网络。这会导致在引导新主机时 DHCP 与卫星服务器冲突。

流程

1. 在 Satellite Web UI 中，导航到 Hosts > Create Host。
2. 在 Name 字段中输入主机的名称。
3. 点 Organization 和 Location 选项卡，以确保置备上下文自动设置为当前上下文。
4. 从 Host Group 列表中，选择您要用于填充表单的主机组。
5. 从 Deploy on 列表中，选择 KVM 连接。
6. 从 Compute Profile 列表中，选择一个配置集来自动填充虚拟机设置。
7. 点 Interface 选项卡，再单击主机的界面上的 Edit。
8. 验证字段是否已自动填充，特别是以下项目：
 - Host 选项卡中的 Name 将成为 DNS 名称。

- *卫星服务器会自动为新主机分配 IP 地址。*
 - *MAC 地址 字段为空。KVM 服务器为主机分配 MAC 地址。*
 - *对于主机上的第一个接口，会自动选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。*
 - *特定于 KVM 的字段填充 **compute** 配置集中的设置。如果需要，修改这些设置。*
9. *单击 **Operating System** 选项卡，并确认所有字段都自动包含值。*
 10. *选择您要使用的置备方法：*
 - *对于基于网络的调配，请单击 **基于网络的**。*
 - *对于基于镜像的调配，可单击 **基于镜像的**。*
 11. *单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。*
 12. *单击 **Virtual Machine** 选项卡，并确认这些设置已填充了主机组和计算配置集的详细信息。修改这些设置以符合您的需要。*
 13. *单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。*
 14. *单击 **Submit** 以保存主机条目。*

对于 CLI 用户

- *要使用基于网络的调配，请使用 `hammer host create` 命令创建主机，并包含 `--provision-method` 构建。将以下示例中的值替换为您环境的相应值。*

```
# hammer host create \
--name "kvm-host1" \
--organization "My_Organization" \
--location "New York" \
--hostgroup "Base" \
--compute-resource "My_KVM_Server" \
--provision-method build \
--build true \
--enabled true \
--managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=network,compute_network=exam
plenetwork" \
--compute-attributes="cpus=1,memory=1073741824" \
--volume="pool_name=default,capacity=20G,format_type=qcow2" \
--root-password "password"
```

•

要使用基于镜像的置备，请使用 `hammer host create` 命令创建主机，并包含 `--provision-method` 镜像。将以下示例中的值替换为您环境的相应值。

```
# hammer host create \
--name "kvm-host2" \
--organization "My_Organization" \
--location "New York" \
--hostgroup "Base" \
--compute-resource "My_KVM_Server" \
--provision-method image \
--image "KVM Image" \
--enabled true \
--managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=network,compute_network=exampl
enetwork" \
--compute-attributes="cpus=1,memory=1073741824" \
--volume="pool_name=default,capacity=20G,format_type=qcow2"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 `hammer 主机 create --help` 命令。

第 10 章 在 RED HAT VIRTUALIZATION 上置备虚拟机

Red Hat Virtualization 是一个企业级服务器和桌面虚拟化平台。在 **Red Hat Satellite** 中，您可以通过 **Red Hat Virtualization** 的 REST API 管理虚拟化功能。这包括创建虚拟机和控制其电源状态。

您可以使用 **Red Hat Virtualization** 置备来通过网络连接或从现有镜像创建虚拟机。

您可以使用 **cloud-init** 配置您置备的虚拟机。使用 **cloud-init** 可避免网络上的任何特殊配置，如托管的 **DHCP** 和 **TFTP**，以完成虚拟机的安装。此方法不需要卫星通过 **SSH** 连接调配的虚拟机，以运行完成脚本。

前提条件

- 为 **Red Hat Enterprise Linux** 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。
- 在 **Red Hat Virtualization** 环境中管理逻辑网络的胶囊服务器。确保此网络中没有运行其他 **DHCP** 服务以避免与胶囊服务器冲突。更多信息请参阅 [第 3 章配置网络](#)。
- 如果要使用基于镜像的置备，则现有模板（空白模板除外）。有关为虚拟机创建模板的更多信息，请参阅 [Red Hat Virtualization 虚拟机管理指南中的模板](#)。
- **Red Hat Virtualization** 上的管理与卫星服务器通信的用户。不要将 **admin@internal** 用户用于此通信。反之，创建一个具有以下权限的新 **Red Hat Virtualization** 用户：
 - **System > 配置系统 > Login Permissions**
 - **network > Configure vNIC Profile > Create**
 - **network > Configure vNIC Profile > Edit Properties**

- *network > Configure vNIC Profile > Delete*
- *network > Configure vNIC Profile > Assign vNIC Profile to VM*
- *network > Configure vNIC Profile > Assign vNIC Profile to Template*
- *Template > Provisioning Operations > Import/Export*
- *VM > Provisioning Operations > Create*
- *VM > Provisioning Operations > Delete*
- *VM > Provisioning Operations > Import/Export*
- *VM > Provisioning Operations > Edit Storage*
- *Disk > Provisioning Operations > Create*
- *Disk > Disk Profile > Attach Disk Profile*

有关如何在 Red Hat Virtualization 中创建用户并添加权限的更多信息，请参阅 [《红帽虚拟化管理指南》](#) 中的“从管理门户管理”管理用户任务。

流程概述

1. [第 10.1 节“将 Red Hat Virtualization 连接添加到卫星服务器”](#)。
2. 可选：[第 10.2 节“在 Red Hat Virtualization 中准备 Cloud-init 镜像”](#)。如果要使用 cloud-init 配置基于镜像的虚拟机，请使用这个步骤。
3. 可选：[第 10.3 节“在 Satellite 服务器中添加 Red Hat Virtualization 镜像”](#)。如果要使用基于

镜像的置备，请使用这个步骤。

4. 可选：第 10.4 节“准备 Cloud-init 模板”。如果要使用 cloud-init 配置基于镜像的虚拟机，请使用这个步骤。
5. 第 10.5 节“在计算配置集中添加 Red Hat Virtualization 详情”。
6. 第 10.6 节“在 Red Hat Virtualization 上创建主机”。

10.1. 将 RED HAT VIRTUALIZATION 连接添加到卫星服务器

使用此流程将 Red Hat Virtualization 添加为 Satellite 中的计算资源。

流程

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources，点 Create Compute Resource。
2. 在 Name 字段中输入新计算资源的名称。
3. 从 Provider 列表中，选择 RHV。
4. 在 Description 字段中，输入计算资源的描述。
5. 在 URL 字段中输入 Red Hat Virtualization Manager API 的连接 URL，格式为：
6. (可选) 选择 Use APIv4 复选框来评估新的 Red Hat Virtualization Manager API。
7. 在 User 字段中输入具有访问权限的用户名称，以访问 Red Hat Virtualization Manager 的资源。

8. 在 **Password** 字段中，输入用户的密码。
9. 点击 **Load Datacenters**，从 **Red Hat Virtualization** 环境中填充 **数据中心** 列表。
10. 从 **数据中心** 列表中选择数据中心。
11. 从 **Quota ID** 列表中，选择配额来限制可供卫星使用的资源。
12. 在 **X509 认证颁发机构** 字段中，为 **SSL/TLS** 访问输入证书颁发机构。或者，如果将字段留空，则服务器的第一个 **API** 请求上会生成自签名证书。
13. 单击 **位置** 选项卡，再选择要使用的位置。
14. 单击 **组织** 选项卡，再选择要使用的组织。
15. 单击 **Submit** 以保存计算资源。

对于 CLI 用户

- 使用 `--provider` 的 **Ovirt** 输入 `hammer compute-resource create` 命令，以及要用于 `--datacenter` 的数据中心的名称。

```
# hammer compute-resource create \
--name "My_RHV" --provider "Ovirt" \
--description "RHV server at rhv.example.com" \
--url "https://rhv.example.com/ovirt-engine/api" \
--use-v4 "false" --user "Satellite_User" \
--password "My_Password" \
--locations "New York" --organizations "My_Organization" \
--datacenter "My_Datacenter"
```

另外，要评估新的 **Red Hat Virtualization Manager API**，对于 `--use-v4` 选项，将 `false` 更改为 `true`。

10.2. 在 RED HAT VIRTUALIZATION 中准备 CLOUD-INIT 镜像

要在置备过程中使用 `cloud-init`，您必须准备一个在 Red Hat Virtualization 中安装 `cloud-init` 的镜像，然后将镜像导入到 Satellite 以供调配使用。

流程

1. 在 Red Hat Virtualization 中，创建一个虚拟机，以便在 Satellite 中进行基于镜像的调配。

2. 在虚拟机上，安装 `cloud-init`:

```
yum install cloud-init
```

3. 在 `/etc/cloud/cloud.cfg` 文件中添加以下信息：

```
datasource_list: ["NoCloud", "ConfigDrive"]
```

4. 在 Red Hat Virtualization 中，从此虚拟机创建镜像。

将此镜像添加到卫星时，请确保您选择了 **User Data** 复选框。

10.3. 在 SATELLITE 服务器中添加 RED HAT VIRTUALIZATION 镜像

要使用基于镜像的置备来创建主机，您必须将镜像的相关信息（如访问详细信息和镜像位置）添加到您的卫星服务器。

流程

1. 在 Satellite Web UI 中，进入 **Infrastructure > Compute Resources** 并点 **Red Hat Virtualization** 连接的名称。

2. 点 **Create Image**。

3. 在 **Name** 字段中输入镜像的名称。

4. 从 **Operating System** 列表中，选择镜像的基本操作系统。
5. 从 **Architecture** 列表中，选择操作系统架构。
6. 在 **Username** 字段中输入镜像访问的 **SSH** 用户名。这通常是 **root** 用户。
7. 在 **Password** 字段中，为镜像访问输入 **SSH** 密码。
8. 从 **Image** 列表中，从 **Red Hat Virtualization** 计算资源中选择镜像。
9. 可选：如果镜像支持用户数据输入，如 **cloud-init** 数据，请选择 **User Data** 复选框。
10. 单击 **Submit** 以保存镜像详情。

对于 CLI 用户

- 使用 **hammer compute-resource image create** 命令创建镜像。使用 **--uuid** 选项在 **Red Hat Virtualization** 服务器上存储模板 **UUID**。

```
# hammer compute-resource image create \
--name "RHV_Image" \
--compute-resource "My_RHV" \
--operatingsystem "RedHat version" \
--architecture "x86_64" \
--username root \
--uuid "9788910c-4030-4ae0-bad7-603375dd72b1" \
```

10.4. 准备 CLOUD-INIT 模板

流程

1. 在 **Satellite Web UI** 中，进入 **Hosts > Provisioning Templates**，然后点 **Create Template**。
2. 在 **Name** 字段中输入模板的名称。

3.

在 **Editor** 字段中输入以下模板详情：

```

<%#
kind: user_data
name: Cloud-init
-%>
#cloud-config
hostname: <%= @host.shortname %>

<%# Allow user to specify additional SSH key as host paramter -%>
<% if @host.params['sshkey'].present? ||
@host.params['remote_execution_ssh_keys'].present? -%>
ssh_authorized_keys:
<% if @host.params['sshkey'].present? -%>
- <%= @host.params['sshkey'] %>
<% end -%>
<% if @host.params['remote_execution_ssh_keys'].present? -%>
<% @host.params['remote_execution_ssh_keys'].each do |key| -%>
- <%= key %>
<% end -%>
<% end -%>
<% end -%>
runcmd:
- |
  #!/bin/bash
  <%= indent 4 do
    snippet 'subscription_manager_registration'
  end %>
  <% if @host.info['parameters']['realm'] && @host.realm && @host.realm.realm_type == 'Red
Hat Identity Management' -%>
  <%= indent 4 do
    snippet 'idm_register'
  end %>
<% end -%>
<% unless @host.operatingsystem.atomic? -%>
  # update all the base packages from the updates repository
  yum -t -y -e 0 update
<% end -%>
<%
  # safemode renderer does not support unary negation
  non_atomic = @host.operatingsystem.atomic? ? false : true
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = non_atomic && (pm_set || @host.params['force-puppet'])
%>
<% if puppet_enabled %>
  yum install -y puppet
  cat > /etc/puppet/puppet.conf << EOF
  <%= indent 4 do
    snippet 'puppet.conf'
  end %>
  EOF
  # Setup puppet to run on system reboot
  /sbin/chkconfig --level 345 puppet on

```

```

/usr/bin/puppet agent --config /etc/puppet/puppet.conf --onetime --tags no_such_tag <%=
@host.puppetmaster.blank? ? " : "--server #{@host.puppetmaster}" %> --no-daemonize
/sbin/service puppet start
<% end -%>
phone_home:
url: <%= foreman_url('built') %>
post: []
tries: 10pp

```

4. 单击 **Type** 选项卡，再从 **Type** 列表中选择 **User data template**。
5. 点 **Association** 选项卡，从 **Applicable Operating Systems** 列表中选择您要与模板关联的操作系统。
6. 单击 **位置** 选项卡，然后从 **位置** 列表中选择要与模板关联的位置。
7. 单击 **Organizations** 选项卡，从 **Organization** 列表中选择您要与模板关联的机构。
8. 点 **Submit**。
9. 进入 **Hosts > Operating Systems**，然后选择您要与模板关联的操作系统。
10. 点 **Templates** 选项卡，从 **User data template** 列表中选择新模板的名称。
11. 点 **Submit**。

10.5. 在计算配置集中添加 RED HAT VIRTUALIZATION 详情

使用此流程将 Red Hat Virtualization 硬件设置添加到计算配置集中。当您使用此计算配置文件在 KVM 上创建主机时，这些设置会被自动填充。

流程

1. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Profiles**。

- 2.

在 **Compute Profiles** 窗口中，单击现有计算配置文件的名称，或者单击 **Create Compute Profile**，输入 **Name**，再单击 **Submit**。

3.

点 **Red Hat Virtualization** 计算资源的名称。

4.

在 **Cluster** 列表中，选择 **Red Hat Virtualization** 环境中的目标主机集群。

5.

在 **Template** 列表中，选择要用于 **Cores** 和 **Memory** 设置的 **RHV** 模板。

6.

在 **Cores** 字段中，输入要分配给新主机的 **CPU** 内核数。

7.

在 **Memory** 字段中输入要分配给新主机的内存量。

8.

从 **Image** 列表中，选择要用于基于镜像的调配的镜像。

9.

在 **Network Interfaces** 区域，为主机的网络接口输入网络参数。您可以创建多个网络接口。但是，至少有一个接口必须指向胶囊管理网络。对于每个网络接口，请输入以下详情：

a.

在 **Name** 字段中输入网络接口的名称。

b.

在 **Network** 列表中，选择您要使用的逻辑网络。

10.

在 **Storage** 区域，为主机输入存储参数。您可以为主机创建多个卷。对于每个卷，请输入以下详情：

a.

在 **Size (GB)** 中，为新卷输入大小（以 **GB** 为单位）。

b.

从 **Storage domain** 列表中，为卷选择存储域。

c.

从 **预分配磁盘** 中，选择精简配置或预分配完整磁盘。

- d. 在 **Bootable** 列表中, 选择您是否希望可引导或者不可启动的卷。

11. 单击 **Submit** 以保存计算配置文件。

对于 CLI 用户

1. 运行以下命令来创建计算配置集：

```
# hammer compute-profile create --name "Red Hat Virtualization CP"
```

2. 要为计算配置集设置值, 请输入以下命令：

```
# hammer compute-profile values create --compute-profile "Red Hat Virtualization CP" \
--compute-resource "My_RHV" \
--interface
"compute_interface=Interface_Type,compute_name=eth0,compute_network=satnetwork" \
--volume "size_gb=20G,storage_domain=Data,bootable=true" \
--compute-attributes "cluster=Default,cores=1,memory=1073741824,start=true"
```

10.6. 在 RED HAT VIRTUALIZATION 上创建主机

在 **Satellite** 中, 您可以使用 **Red Hat Virtualization** 置备通过网络连接或从现有镜像创建主机：

- 如果要通过网络连接创建主机, 则新主机必须能够访问卫星服务器的集成胶囊或 **Red Hat Virtualization** 虚拟网络上的外部胶囊服务器, 以便主机可以访问 **PXE** 调配服务。这个新主机条目会触发 **Red Hat Virtualization** 服务器创建并启动虚拟机。如果虚拟机通过虚拟网络检测到定义的胶囊服务器, 虚拟机将启动到 **PXE**, 并开始安装所选操作系统。
- 如果要使用现有镜像创建主机, 则新的主机条目会触发 **Red Hat Virtualization** 服务器使用预先存在的镜像作为新卷的基础来创建虚拟机。

DHCP Conflicts

对于基于网络的配置, 如果您使用 **Red Hat Virtualization** 服务器上的虚拟网络进行置备, 请选择不提供 **DHCP** 分配的网络。这会导致在引导新主机时 **DHCP** 与卫星服务器冲突。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Create Host**。
2. 在 **Name** 字段中输入主机的名称。
3. 点 **Organization** 和 **Location** 选项卡，以确保置备上下文自动设置为当前上下文。
4. 从 **Host Group** 列表中，选择您要用于填充表单的主机组。
5. 从 **Deploy on** 列表中，选择 **Red Hat Virtualization** 连接。
6. 从 **Compute Profile** 列表中，选择一个配置集来自动填充虚拟机设置。
7. 点 **Interface** 选项卡，再单击主机的界面上的 **Edit**。
8. 验证字段是否已自动填充，特别是以下项目：
 - **Host** 选项卡中的 **Name** 将成为 **DNS** 名称。
 - 卫星服务器会自动为新主机分配 **IP** 地址。
 - **MAC 地址** 字段为空。**Red Hat Virtualization** 服务器为主机分配 **MAC** 地址。
 - 对于主机上的第一个接口，会自动选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。
 - **Red Hat Virtualization** 特定的字段使用计算配置集中的设置填充。如果需要，修改这些设置。
9. 单击 **Operating System** 选项卡，并确认所有字段都自动包含值。

10.

选择您要使用的置备方法：

- 对于基于网络的调配，请单击 **基于网络的**。
- 对于基于镜像的调配，可单击 **基于 镜像的**。

11.

单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。

12.

单击 **Virtual Machine** 选项卡，并确认这些设置已填充了主机组和计算配置集的详细信息。修改这些设置以符合您的需要。

13.

单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。

14.

单击 **Submit** 以保存主机条目。

对于 CLI 用户

- 要使用基于网络的调配，请使用 `hammer host create` 命令创建主机，并包含 `--provision-method` 构建。将以下示例中的值替换为您环境的相应值。

```
# hammer host create \
--name "RHV-vm1" \
--organization "My_Organization" \
--location "New York" \
--hostgroup "Base" \
--compute-resource "My_RHV" \
--provision-method build \
--build true \
--enabled true \
--managed true \
--interface
"managed=true,primary=true,provision=true,compute_name=eth0,compute_network=satnetwor
k" \
--compute-attributes="cluster=Default,cores=1,memory=1073741824,start=true" \
--volume="size_gb=20G,storage_domain=Data,bootable=true"
```

- 要使用基于镜像的置备，请使用 `hammer host create` 命令创建主机，并包含 `--provision-method` 镜像。将以下示例中的值替换为您环境的相应值。

```
# hammer host create \  
--name "RHV-vm2" \  
--organization "My_Organization" \  
--location "New York" \  
--hostgroup "Base" \  
--compute-resource "My_RHV" \  
--provision-method image \  
--image "RHV_Image" \  
--enabled true \  
--managed true \  
--interface  
"managed=true,primary=true,provision=true,compute_name=eth0,compute_network=satnetwor  
k" \  
--compute-attributes="cluster=Default,cores=1,memory=1073741824,start=true" \  
--volume="size_gb=20G,storage_domain=Data,bootable=true"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 **hammer 主机 create --help** 命令。

第 11 章 在 VMWARE VSPHERE 中置备虚拟机

VMware vSphere 是 VMware 企业级虚拟化平台。红帽卫星 6 可以与 vSphere 平台交互，包括创建新虚拟机和控制其电源管理状态。

11.1. VMWARE VSPHERE PROVISIONING 的先决条件

VMware vSphere 置备的要求包括：

- 在 vSphere 环境中管理网络的胶囊服务器。确保此网络中没有运行其他 DHCP 服务以避免与胶囊服务器冲突。更多信息请参阅 [第 3 章 配置网络](#)。
- 如果要使用基于镜像的置备，则现有的 VMware 模板。
- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。

11.2. 创建 VMWARE VSPHERE 用户

VMware vSphere 服务器需要类似管理的用户进行卫星服务器通信。为安全起见，请不要使用管理员用户进行此类通信。反之，创建具有以下权限的用户：

对于 VMware vCenter Server 版本 6.9，设置以下权限：

- All Privileges → Datastore → Allocate Space, Browse data, Update Virtual Machine files, Low level file operations
- All Privileges → Network → Assign Network
- All Privileges → Resource → Assign machine to resource pool

- *All Privileges → Virtual Machine → Change Config (All)*
- *All Privileges → Virtual Machine → Interaction (All)*
- *All Privileges → Virtual Machine → Edit Inventory (All)*
- *All Privileges → Virtual Machine → Provisioning (All)*

请注意，相同的步骤也适用于 VMware vCenter Server 版本 7.0。

对于 VMware vCenter Server 版本 6.5，设置以下权限：

- *All Privileges → Datastore → Allocate Space, Browse data, Update Virtual Machine files, Low level file operations*
- *All Privileges → Network → Assign Network*
- *All Privileges → Resource → Assign machine to resource pool*
- *All Privileges → Virtual Machine → Configuration (All)*
- *All Privileges → Virtual Machine → Interaction (All)*
- *All Privileges → Virtual Machine → Inventory (All)*
- *All Privileges → Virtual Machine → Provisioning (All)*

11.3. 为 SATELLITE 服务器添加 VMWARE VSPHERE 连接

使用这个步骤在卫星服务器的计算资源中添加 VMware vSphere 连接。

确保将主机和基于网络的防火墙配置为允许 Satellite 到 TCP 端口 443 上的 vCenter 通信。验证 Satellite 能否解析 vCenter 和 vCenter 的主机名，以便能解析 Satellite 服务器的主机名。

流程

要添加连接，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources，并在 Compute Resources 窗口中点击 Create Compute Resource。
2. 在 Name 字段中输入资源的名称。
3. 从 Provider 列表中，选择 VMware。
4. 在 Description 字段中输入资源描述。
5. 在 VCenter/Server 字段中输入 vCenter 服务器的 IP 地址或主机名。
6. 在 User 字段中，输入具有访问 vCenter 资源的权限的用户名。
7. 在 Password 字段中，输入用户的密码。
8. 点 Load Datacenters 从 VMware vSphere 环境中填充数据中心的列表。
9. 从 Datacenter 列表中，从此列表中选择要管理的特定数据中心。
10. 在 Fingerprint 字段中，确保此字段填充了数据中心的指纹。
11. 在 Display Type 列表中，选择一个控制台类型，如 VNC 或 VMRC。请注意，VMware ESXi

6.5 及之后的版本不支持 VNC 控制台。

12.

可选：在 **VNC Console Passwords** 字段中，选择 **Set a random generated password on the display connection** 复选框，以使用随机生成的密码保护新主机的控制台访问。您可以从以下命令的输出结果获取 VNC 控制台的密码，从 `libvirtd` 主机访问客户机虚拟机控制台：

```
# virsh edit your_VM_name
<graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'
passwd='your_randomly_generated_password'>
```

每次打开虚拟机的控制台时，密码会随机生成，例如使用 `virt-manager`。

13.

从 **Enable caching** 列表中，您可以选择是否启用计算资源缓存。更多信息请参阅 [第 11.8 节“计算资源缓存”](#)。

14.

单击 **Locations** 和 **Organizations** 选项卡，然后验证这些值是否已自动设置为当前上下文。您还可以添加额外的上下文。

15.

点 **Submit** 保存连接。

对于 CLI 用户

使用 `hammer compute-resource create` 命令创建连接。选择 `Vmware` 作为 `--provider`，并将数据中心的实例 `UUID` 设置为 `--uuid`：

```
# hammer compute-resource create --name "My_vSphere" \
--provider "Vmware" \
--description "vSphere server at vsphere.example.com" \
--server "vsphere.example.com" --user "My_User" \
--password "My_Password" --locations "My_Location" --organizations "My_Organization" \
--datacenter "My_Datacenter"
```

11.4. 在 SATELLITE 服务器中添加 VMWARE VSPHERE 镜像

VMware vSphere 使用模板作为创建新虚拟机的镜像。如果使用基于镜像的置备来创建新主机，则需要将 **VMware** 模板详情添加到卫星服务器中。这包括访问详情和模板名称。

流程

要添加镜像，请完成以下步骤：

1. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Resources**，并在 **Compute Resources** 窗口中点击 **VMware vSphere** 连接。
2. 在 **Name** 字段中输入镜像的名称。
3. 在 **Operatingssystem** 列表中，选择镜像的基本操作系统。
4. 从 **Architecture** 列表中，选择操作系统架构。
5. 在 **User** 字段中，输入镜像访问的 **SSH** 用户名。这通常是 **root** 用户。
6. 在 **Password** 字段中，为镜像访问输入 **SSH** 密码。
7. 从 **User data** 列表中，选择您是否希望镜像支持用户数据输入，如 **cloud-init** 数据。
8. 在 **Image** 字段中，在 **vSphere** 环境中输入模板的相对路径和名称。不要在相对路径中包含数据中心。
9. 单击 **Submit** 以保存镜像详情。

对于 **CLI** 用户

使用 **hammer compute-resource image create** 命令创建镜像。使用 **--uuid** 字段将相对模板路径存储在 **vSphere** 环境中。

```
# hammer compute-resource image create --name "Test_vSphere_Image" \
--operatingsystem "RedHat 7.2" --architecture "x86_64" \
--username root --uuid "Templates/RHEL72" \
--compute-resource "My_vSphere"
```

11.5. 在 COMPUTE PROFILE 中添加 VMWARE VSPHERE DETAILS

您可以预定义 VMware vSphere 上虚拟机的某些硬件设置。您可以通过在计算配置文件中添加这些硬件设置来达到此目的。

流程

要将 VMware vSphere 详情添加到计算配置集，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 Infrastructure & Compute Profiles，然后在 Compute Profiles 窗口中点击 compute 配置集的名称，然后点 vSphere 连接。
2. 在 CPU 字段中，输入分配给新主机的 CPU 数量。
3. 在 Cores per socket 字段中输入要分配给每个 CPU 的内核数。
4. 在 Memory 字段中输入要分配给新主机的内存量。
5. 在 Cluster 字段中输入 VMware 环境中的目标主机集群名称。
6. 从 Resource pool 列表中，为主机选择可用资源分配。
7. 在 Folder 字段中，输入文件夹以组织主机。
8. 在 Guest OS 列表中选择您要在 VMware vSphere 中使用的操作系统。
9. 从 SCSI 控制器列表中，为主机选择磁盘访问方法。
10. 在 Virtual H/W 版本列表中选择用于虚拟机的底层 VMware 硬件抽象。
11. 如果要在虚拟机开机时添加更多资源，选择 Memory hot add 或 CPU 热添加复选框。

12. 从 **Image** 列表中，选择在执行基于镜像置备时要使用的镜像。
13. 从 **Network Interfaces** 列表中，为主机网络接口选择网络参数。您可以创建多个网络接口。但是，至少有一个接口必须指向胶囊管理网络。
14. 如果要使用零个配置，选择 **Eager zero** 复选框。如果取消选中，磁盘使用 **lazy zero thick provisioning**。
15. 单击 **Submit** 以保存计算配置文件。

对于 CLI 用户

1. 运行以下命令来创建计算配置集：

```
# hammer compute-profile create --name "VMWare CP"
```

2. 要为 **compute** 配置集添加值，请输入以下命令：

```
# hammer compute-profile values create --compute-profile "VMWare CP" \
--compute-resource "My_vSphere" \
--interface "compute_type=VirtualE1000,compute_network=mynetwork \
--volume "size_gb=20G,datastore=Data,name=myharddisk,thin=true" \
--compute-attributes
"cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=true"
```

11.6. 在 VMWARE VSPHERE 服务器上创建主机

VMware vSphere 置备过程提供了通过网络连接或使用现有镜像创建主机的选项。

对于基于网络的调配，您必须创建一个主机来访问卫星服务器的集成胶囊或 **VMware vSphere** 虚拟网络上的外部胶囊服务器，以便主机能够访问 **PXE** 调配服务。新主机条目会触发 **VMware vSphere** 服务器来创建虚拟机。如果虚拟机通过虚拟网络检测到定义的胶囊服务器，虚拟机将启动到 **PXE**，并开始安装所选操作系统。

DHCP Conflicts

如果您使用 **VMware vSphere** 服务器上的虚拟网络置备，请确保选择一个没有提供 **DHCP** 分配的虚拟网络。这会导致在引导新主机时 **DHCP** 与卫星服务器冲突。

对于基于镜像的调配，请使用预先存在的镜像作为新卷的基础。

流程

要为 VMware vSphere 服务器创建主机，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 Hosts > Create Host。
2. 在 Name 字段中输入您要成为置备的系统主机名的名称。
3. 点 Organization 和 Location 选项卡，以确保置备上下文自动设置为当前上下文。
4. 从 Host Group 列表中，选择您要用于填充表单的主机组。
5. 从 Deploy on 列表中，选择 VMware vSphere 连接。
6. 从 Compute Profile 列表中，选择一个配置集来自动填充基于虚拟机的设置。
7. 点 Interface 选项卡，再单击主机的界面上的 Edit。
8. 验证字段是否自动填充了值。特别是：
 - Host 选项卡中的 Name 将成为 DNS 名称。
 - 卫星服务器会自动为新主机分配 IP 地址。
9. 确保 MAC 地址 字段为空。VMware vSphere 服务器为主机分配一个。
10. 验证主机上第一个接口已自动选择 Managed、Primary 和 Provision 选项。如果没有，请选

择它们。

11. 在接口窗口中，查看使用我们计算配置集中的设置填充的 VMware vSphere 特定字段。修改这些设置以符合您的需要。
12. 单击 **Operating System** 选项卡，并确认所有字段都自动包含值。
13. 选择您想要的置备方法：
 - 对于基于网络的调配，请单击 **基于网络的**。
 - 对于基于镜像的调配，可单击 **基于 镜像的**。
 - 对于引导磁盘调配，请点 **基于 的引导磁盘**。
14. 单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。
15. 单击 **Virtual Machine** 选项卡，并确认这些设置已填充了主机组和计算配置集的详细信息。修改这些设置以符合您的要求。
16. 单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。
17. 单击 **Submit** 以保存主机条目。

对于 CLI 用户

使用 `hammer host create` 命令通过网络创建主机，并包含 `--provision-method` 构建以使用基于网络的置备。

```
# hammer host create --name "vmware-test1" --organization "My_Organization" \
--location "New York" --hostgroup "Base" \
--compute-resource "My_vSphere" --provision-method build \
--build true --enabled true --managed true \
--interface
```

```
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=mynetwork" \
--compute-
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=true" \
--volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 **hammer 主机 create --help** 命令。

对于 CLI 用户

使用 **hammer host create** 命令从镜像创建主机，并包含 **--provision-method** 镜像以使用基于镜像的置备。

```
# hammer host create --name "vmware-test2" --organization "My_Organization" \
--location "New York" --hostgroup "Base" \
--compute-resource "My_VMware" --provision-method image \
--image "Test VMware Image" --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=mynetwork" \
--compute-
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=true" \
--volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"
```

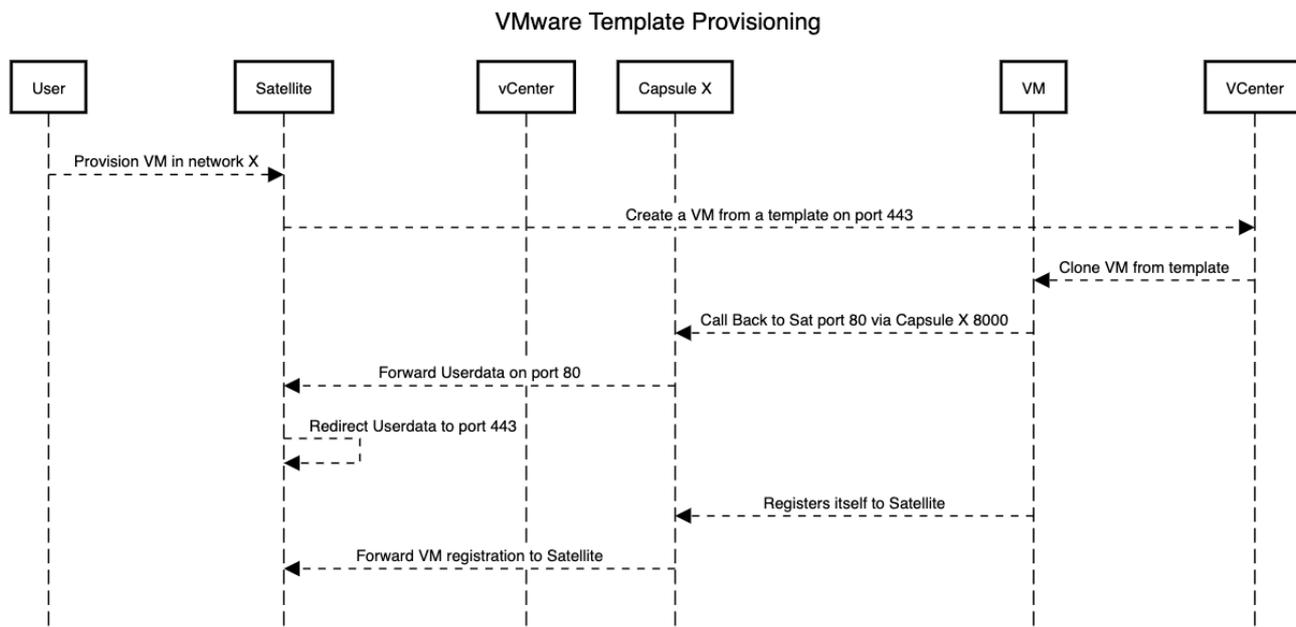
有关此计算资源的额外主机创建参数的更多信息，请输入 **hammer 主机 create --help** 命令。

11.7. 使用 VMWARE VSPHERE CLOUD-INIT 和 USERDATA 模板进行置备

您可以将 VMware 与 Cloud-init 和 Userdata 模板搭配使用，将用户数据插入到新的虚拟机中，进行进一步的 VMware 自定义，并让 VMware 托管的虚拟机重新调用 Satellite。

您可以使用相同的步骤在卫星中设置 VMware 计算资源，对 workflow 进行了一些修改。

VMware cloud-init 调配概述



当您在 **Satellite** 中为 **VMware** 置备设置计算资源和镜像时，会发生以下一系列置备事件：

- 用户使用 **Satellite web UI**、**API** 或 **hammer** 置备一个或多个虚拟机
- **Satellite** 调用 **VMware vCenter** 来克隆虚拟机模板
- **Satellite userdata** 调配模板添加自定义身份信息
- 置备完成后，**Cloud-init** 调配模板指示虚拟机在 **cloud-init** 运行时重新调用胶囊
- **VMware vCenter** 将模板克隆到虚拟机
- **VMware vCenter** 对虚拟机的身份应用自定义，包括主机名、IP 和 DNS
- 虚拟机构建 **cloud-init** 被调用，并在端口 80 上调用回 **Satellite**，然后重定向到 443

端口和防火墙要求

由于 **cloud-init** 服务，虚拟机总是调用卫星，即使您将虚拟机注册到胶囊。确保配置端口和防火墙设

置以打开任何必要的连接。

有关端口和防火墙要求的更多信息，请参阅安装胶囊服务器中的 [安装卫星](#) 和 [端口和防火墙要求中的端口和防火墙要求](#)。

将 `userdata` 和 `Cloud-init` 模板与操作系统关联

1. 在 `Satellite Web UI` 中，进入 `Hosts > Operating Systems`，然后选择您要用于置备的操作系统。
2. 点 `Template` 选项卡。
3. 从 `Cloud-init` 模板列表中，选择 `Cloudinit default`。
4. 从 `User data template` 列表中，选择 `UserData open-vm-tools`。
5. 单击 `Submit` 以保存更改。

准备镜像以使用 `cloud-init` 模板

要准备镜像，您必须首先在虚拟机上配置您需要的设置，然后保存为要在 `Satellite` 中使用的镜像。

要使用 `cloud-init` 模板进行置备，您必须配置虚拟机，以便安装、启用并配置为返回卫星服务器。

为了安全起见，您必须安装 `CA` 证书以使用 `HTTP` 进行所有通信。此流程包括清理虚拟机的步骤，以便不需要发送到您置备的镜像的信息。

如果您有带有 `cloud-init` 的镜像，您仍需要按照此流程启用 `cloud-init` 与 `Satellite` 通信，因为 `cloud-init` 会被默认禁用。

1. 在用于创建镜像的虚拟机上，安装 `cloud-init`、`open-vm-tools` 和 `perl`：

```
# yum -y install cloud-init open-vm-tools perl
```

2. 为 **cloud-init** 创建配置文件 :

```
# vi /etc/cloud/cloud.cfg.d/example_cloud-init_config.cfg
```

3. 在 **example_cloud_init_config.cfg** 文件中添加以下信息 :

```
datasource_list: [NoCloud]
datasource:
  NoCloud:
    seedfrom: https://satellite.example.com/userdata/
EOF
```

4. 为镜像启用 **CA** 证书 :

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

5. 从 **Satellite** 服务器下载 **katello-server-ca.crt** 文件 :

```
# wget -O /etc/pki/ca-trust/source/anchors/cloud-init-ca.crt
http://satellite.example.com/pub/katello-server-ca.crt
```

6. 要更新证书记录, 请输入以下命令 :

```
# update-ca-trust extract
```

7. 使用以下命令清理镜像 :

```
# systemctl stop rsyslog
# systemctl stop auditd
# package-cleanup --oldkernels --count=1
# yum clean all
```

8. 使用以下命令减少日志空间、删除旧日志和截断日志 :

```
# logrotate -f /etc/logrotate.conf
# rm -f /var/log/*-???????? /var/log/*.gz
# rm -f /var/log/dmesg.old
# rm -rf /var/log/anaconda
# cat /dev/null > /var/log/audit/audit.log
```

```
# cat /dev/null > /var/log/wtmp
# cat /dev/null > /var/log/lastlog
# cat /dev/null > /var/log/grubby
```

9.

删除 udev 硬件规则：

```
# rm -f /etc/udev/rules.d/70*
```

10.

从 ifcfg 脚本中删除 uuid：

```
# cat > /etc/sysconfig/network-scripts/ifcfg-ens192 <<EOM
DEVICE=ens192
ONBOOT=yes
EOM
```

11.

删除 SSH 主机密钥：

```
# rm -f /etc/ssh/SSH_keys
```

12.

删除 root 用户的 shell 历史记录：

```
# rm -f ~root/.bash_history
# unset HISTFILE
```

13.

删除 root 用户的 SSH 历史记录：

```
# rm -rf ~root/.ssh/known_hosts
```

现在，您可以从此虚拟机创建镜像。

您可以使用 [第 11.4 节“在 Satellite 服务器中添加 VMware vSphere 镜像”](#) 部分将镜像添加到 Satellite 中。

配置胶囊以转发用户数据模板

如果您使用胶囊模板功能部署 Satellite，则必须将卫星配置为识别通过 X-Forwarded-For HTTP 标头转发的主机 IP 地址，以提供正确的模板有效负载。

为安全起见，卫星只识别来自本地主机的这个 HTTP 标头。对于每个单独的胶囊，您必须配置正则表达式来识别主机的 IP 地址。在 Web UI 中，您可以通过导航到 **Administer > Settings > Provisioning** 并更改 **Remote address** 设置来执行此操作。通过 CLI，您可以输入以下命令来实现：

```
# hammer settings set --name remote_addr --value '(localhost(4|6|4to6)?|192.168.122.(1|2|3))'
```

11.8. 计算资源缓存

计算资源的缓存可加快 VMware 信息渲染速度。

11.8.1. 启用计算资源的缓存

启用或禁用计算资源缓存：

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Compute Resources**。
2. 点击您要更新的 VMware 服务器右侧的 **Edit** 按钮。
3. 选择 **Enable caching** 复选框。

11.8.2. 刷新计算资源缓存

刷新计算资源缓存以更新计算资源信息：

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Compute Resources**。
2. 选择您要刷新计算资源缓存的 VMware 服务器，然后单击 **Refresh Cache** 按钮。

对于 CLI 用户

使用此 API 调用来刷新计算资源缓存：

```
# curl -H "Accept:application/json,version=2" \  
-H "Content-Type:application/json" -X PUT \  
-u username:password -k \  
https://satellite.example.com/api/compute_resources/compute_resource_id/refresh_cache
```

使用 `hammer compute-resource list` 命令确定您要刷新计算资源缓存的 VMware 服务器的 ID。

第 12 章 在容器原生虚拟化中置备虚拟机

容器原生虚拟化需要采用或希望采用 Kubernetes 但拥有现有虚拟机(VM)的工作负载（无法轻松容器化）的开发团队。该技术提供了一个统一的开发平台，开发人员可以在共享环境中构建、修改和部署驻留于应用程序容器和虚拟机中的应用。这些功能支持跨开放混合云的快速应用程序现代化。

使用 Red Hat Satellite，您可以为容器原生虚拟化创建计算资源，以便可以使用 Satellite 来置备和管理 Kubernetes 虚拟机。

请注意，此版本不支持模板置备。



重要

容器原生虚拟化计算资源只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽技术预览功能支持范围的详情，请参考

<https://access.redhat.com/support/offerings/techpreview/>。

前提条件

- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。
- 具有 OpenShift Container Platform 虚拟集群的 cluster-admin 权限的 Container-native Virtualization 用户。如需更多信息，请参阅 OpenShift Container Platform 文档中的 [身份验证指南中的使用 RBAC 定义和应用权限](#)。
- 管理容器原生虚拟化服务器上的网络的胶囊服务器。确保此网络中没有运行其他 DHCP 服务以避免与胶囊服务器冲突。有关胶囊服务器的网络服务配置的更多信息，请参阅 [第 3 章配置网络](#)。
- 具有以下角色的 Satellite 用户帐户：

- **编辑主机**
- **查看主机**

有关更多信息，请参阅管理 Red Hat Satellite 指南中的 [将角色分配到用户](#)。

- **Satellite 中的自定义角色，具有以下权限：**

- **view_compute_resources**
- **destroy_compute_resources_vms**
- **power_compute_resources_vms**
- **create_compute_resources_vms**
- **view_compute_resources_vms**
- **view_locations**
- **view_subnets**

有关创建角色的更多信息，请参阅管理 Red Hat Satellite 指南中的 [创建角色](#)。 https://access.redhat.com/documentation/en-us/red_hat_satellite/6.9/html-single/administering_red_hat_satellite/index#creating-a-role_admin 有关添加角色权限的更多信息，请参阅管理 Red Hat Satellite 指南中的 [角色添加权限](#)。

12.1. 将容器原生虚拟化连接添加到卫星服务器

使用此流程将 Container-native Virtualization 添加为 Satellite 中的计算资源。

流程

1. 输入以下 `satellite-installer` 命令，为 **Satellite** 启用 **Container-native Virtualization** 插件：

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

2. 生成用于 HTTP 和 HTTP 验证的 bearer 令牌。在 **Container-native Virtualization** 服务器上列出包含令牌的 secret：

```
# kubectl get secrets
```

3. 列出 secret 的令牌：

```
# kubectl get secrets YOUR_SECRET -o jsonpath='{.data.token}' | base64 -d | xargs
```

记下此令牌以便在此流程的后续部分中使用。

4. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Resources**，然后点 **Create Compute Resource**。
5. 在 **Name** 字段中输入新计算资源的名称。
6. 从 **Provider** 列表中，选择 **Container-native Virtualization**。
7. 在 **Description** 字段中，输入计算资源的描述。
8. 在 **Hostname** 字段中，输入您要使用的 **Container-native Virtualization** 服务器地址。
9. 在 **API Port** 字段中，输入您要用于从 **Satellite** 到 **Container-native Virtualization** 置备请求的端口号。
10. 在 **Namespace** 字段中，输入您要使用的 **Container-native Virtualization** 裸机集群的用户名。

11. 在 **Token** 字段中, 为 **HTTP** 和 **HTTP** 输入 **bearer** 令牌。

12. 可选 : 在 **X509 Certificate Authorities** 字段中输入证书, 为 **API 服务器调用启用客户端证书验证**。

第 13 章 PROVISIONING CLOUD INSTANCES ON RED HAT OPENSTACK PLATFORM

Red Hat OpenStack Platform 为构建私有或公共基础架构即服务(IaaS)云提供了基础。它提供了一个高度可扩展的容错平台，用于开发支持云的工作负载。在红帽卫星 6 中，您可以与红帽 OpenStack 平台 REST API 交互，以创建云实例并控制其电源管理状态。

前提条件

- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的 同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的 创建激活码](#)。
- 在您的 OpenStack 环境中管理网络的胶囊服务器。更多信息请参阅 [第 3 章 配置网络](#)。
- 添加至 OpenStack Image Storage (glance)服务的镜像，用于基于镜像的置备。如需更多信息，请参阅 [Red Hat OpenStack Platform 实例和镜像指南](#)。

流程概述

1. [第 13.1 节 “将 Red Hat OpenStack Platform 连接添加到卫星服务器”](#)。
2. [第 13.2 节 “将 Red Hat OpenStack Platform 镜像添加到卫星服务器”](#)。
3. [第 13.3 节 “在计算配置集中添加 Red Hat OpenStack Platform 详情”](#)。
4. [第 13.4 节 “Creating Image-based Hosts on Red Hat OpenStack Platform”](#)。

13.1. 将 RED HAT OPENSTACK PLATFORM 连接添加到卫星服务器

使用这个步骤将 Red Hat OpenStack Platform 添加为 Satellite 中的计算资源。

流程

- 1.

在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Resources**，点 **Create Compute Resource**。

2. 在 **Name** 字段中输入新计算资源的名称。
3. 从 **Provider** 列表中，选择 **RHEL OpenStack Platform**。
4. 在 **Description** 字段中，输入计算资源的描述。
5. 在 **URL** 字段中，在令牌资源中输入 **OpenStack Authentication keystone** 服务的 API 的 URL。使用以下命令：<http://openstack.example.com:5000/v3.0/tokens>。
6. 在 **User and Password** 字段中，输入卫星的身份验证用户和密码来访问环境。
7. 在 **Domain** 字段中，为 **V3** 身份验证输入域。
8. 从 **Tenant** 列表中，选择要管理的租户或项目。
9. 要将外部网络用作主机的主网络，请选择 **Allow external network as main network** 复选框。
10. 单击 **Locations** 和 **Organizations** 选项卡，然后验证您要使用的位置和组织是否设置为当前上下文。在这些标签页中添加您想要的额外上下文。
11. 点 **Submit** 保存 **Red Hat OpenStack Platform** 连接。

对于 CLI 用户

- 要创建计算资源，请输入 **hammer compute-resource create** 命令：

```
# hammer compute-resource create --name "My_OpenStack" \  
--provider "OpenStack" \  
--description "My OpenStack environment at openstack.example.com" \  
--url "http://openstack.example.com:5000/v3.0/tokens" \  
--user "admin" --password "admin" --tenant "admin" --domain "V3" \  
--allow-external-network-as-main-network
```

```
--url "http://openstack.example.com:5000/v3.0/tokens" --user "My_Username" \  
--password "My_Password" --tenant "openstack" --locations "New York" \  
--organizations "My_Organization"
```

13.2. 将 RED HAT OPENSTACK PLATFORM 镜像添加到卫星服务器

要使用基于镜像的置备来创建主机，您必须将镜像的相关信息（如访问详细信息和镜像位置）添加到您的卫星服务器。

流程

1. 在 **Satellite Web UI** 中，进入 **Infrastructure > Compute Resources** 并点 **Red Hat OpenStack Platform** 连接的名称。
2. 点 **Create Image**。
3. 在 **Name** 字段中输入镜像的名称。
4. 从 **Operating System** 列表中，选择镜像的基本操作系统。
5. 从 **Architecture** 列表中，选择操作系统架构。
6. 在 **Username** 字段中输入镜像访问的 **SSH** 用户名。这通常是 **root** 用户。
7. 在 **Password** 字段中，为镜像访问输入 **SSH** 密码。
8. 从 **Image** 列表中，从 **Red Hat OpenStack Platform** 计算资源中选择镜像。
9. 可选：如果镜像支持用户数据输入，如 **cloud-init** 数据，请选择 **User Data** 复选框。
10. 单击 **Submit** 以保存镜像详情。

对于 **CLI** 用户

●

使用 `hammer compute-resource image create` 命令创建镜像。使用 `--uuid` 字段将镜像位置的完整路径保存在 Red Hat OpenStack Platform 服务器上。

```
# hammer compute-resource image create \
--name "OpenStack Image" \
--compute-resource "My_OpenStack_Platform" \
--operatingsystem "RedHat version" \
--architecture "x86_64" \
--username root \
--user-data true \
--uuid "/path/to/OpenstackImage.qcow2"
```

13.3. 在计算配置集中添加 RED HAT OPENSTACK PLATFORM 详情

使用这个步骤将 Red Hat OpenStack Platform 硬件设置添加到计算配置集。当您使用此计算配置文件在 Red Hat OpenStack Platform 上创建主机时，这些设置会自动填充。

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Compute Profiles**。
2. 在 **Compute Profiles** 窗口中，单击现有计算配置文件的名称，或者单击 **Create Compute Profile**，输入 **Name**，再单击 **Submit**。
3. 点 **Red Hat OpenStack Platform** 计算资源的名称。
4. 从 **Flavor** 列表中，选择用于主机的 **Red Hat OpenStack Platform** 上的硬件配置集。
5. 从 **可用区** 列表中选择要在 **Red Hat OpenStack Platform** 环境中使用的目标集群。
6. 从 **Image** 列表中，选择要用于基于镜像调配的镜像。
7. 从 **Tenant** 列表中，为 **Red Hat OpenStack Platform** 实例选择租户或项目。
8. 从 **Security Group** 列表中，为端口和 IP 地址选择基于云的访问规则。

9. 从 **内部网络**，选择要加入主机的专用网络。
10. 从 **浮动 IP 网络**，选择要加入的主机的外部网络并分配浮动 IP 地址。
11. 从 **Boot from volume** 中选择是否从镜像创建卷。如果没有选择，实例将直接引导镜像。
12. 在 **New boot 卷大小(GB)** 字段中，输入新引导卷的大小（以 GB 为单位）。
13. 单击 **Submit** 以保存计算配置文件。

对于 CLI 用户

计算配置文件 CLI 命令尚未在红帽卫星 6.9 中实施。另外，您可以在主机创建过程中直接包含相同的设置。

13.4. CREATING IMAGE-BASED HOSTS ON RED HAT OPENSTACK PLATFORM

在卫星中，您可以使用 Red Hat OpenStack Platform 置备从现有镜像创建主机。新主机条目触发 Red Hat OpenStack Platform 服务器，以使用预先存在的镜像作为新卷的基础创建实例。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Create Host**。
2. 在 **Name** 字段中输入主机的名称。
3. 点 **Organization** 和 **Location** 选项卡，以确保置备上下文自动设置为当前上下文。
4. 从 **Host Group** 列表中，选择您要用于填充表单的主机组。
5. 从 **Deploy on** 列表中，选择 **Red Hat OpenStack Platform** 连接。

6. 从 **Compute Profile** 列表中，选择一个配置集来自动填充虚拟机设置。
7. 从 **Lifecycle Environment** 列表中，选择环境。
8. 点 **Interfaces** 选项卡，再单击主机接口上的 **Edit**。
9. 验证字段是否已自动填充，特别是以下项目：
 - **Host** 选项卡中的 **Name** 将成为 **DNS** 名称。
 - **MAC 地址** 字段为空。**Red Hat OpenStack Platform** 在置备过程中为主机分配 **MAC 地址**。
 - 卫星服务器会自动为新主机分配 **IP 地址**。
 - 对于主机上的第一个接口，会自动选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。
10. 单击 **Operating System** 选项卡，并确认所有字段都自动包含值。
11. 如果要更改从计算配置文件自动填充的镜像，请从 **Images** 列表中选择另一个镜像到新主机的根卷。
12. 单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。
13. 单击 **Virtual Machine** 选项卡，并确认这些设置已填充了主机组和计算配置集的详细信息。修改这些设置以符合您的需要。
14. 单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。

15.

单击 **Submit** 以保存主机条目。

对于 CLI 用户

- 使用 `hammer host create` 命令创建主机，并包含 `--provision-method` 镜像。将以下示例中的值替换为您环境的相应值。

```
# hammer host create \  
--name "openstack-host1" \  
--organization "My_Organization" \  
--location "New York" \  
--hostgroup "Base" \  
--compute-resource "My_OpenStack_Platform" \  
--provision-method image \  
--image "OpenStack Image" \  
--enabled true \  
--managed true \  
--interface "managed=true,primary=true,provision=true" \  
--compute-attributes="flavor_ref=m1.small,tenant_id=openstack,security_groups=default,network=mynetw  
ork"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 `hammer 主机 create --help` 命令。

第 14 章 在 AMAZON EC2 中置备云实例

Amazon Elastic Compute Cloud (Amazon EC2) 是一个提供公共云计算资源的 Web 服务。使用卫星，您可以和 Amazon EC2 的公共 API 交互，以创建云实例并控制其电源管理状态。使用本章中的步骤添加连接到 Amazon EC2 帐户并调配云实例。

14.1. AMAZON EC2 置备先决条件

Amazon EC2 置备的要求包括：

- 在 EC2 环境中管理网络的胶囊服务器。使用虚拟私有云(VPC)确保主机和胶囊服务器之间的安全网络。
- 用于基于镜像的置备的 Amazon Machine Image (AMI)。
- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。

14.2. 将 AMAZON EC2 连接添加到卫星服务器

使用这个步骤将 Amazon EC2 连接添加到卫星服务器的计算资源中。

时间设置和 Amazon Web Services

Amazon Web Services 使用时间设置作为身份验证过程的一部分。确保正确同步卫星服务器的时间。确保 NTP 服务（如 ntpd 或 chronyd）在卫星服务器上正常运行。如果没有为 Amazon Web Services 提供正确的时间，可能会导致身份验证失败。

有关在 Satellite 中同步时间的更多信息，请参阅 [从连接的网络安装卫星服务器中的同步时间](#)。

流程

要添加 Amazon EC2 连接，请完成以下步骤：

1. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Resources**，并在 **Compute Resources** 窗口中点击 **Create Compute Resource**。
2. 在 **Name** 字段中输入一个名称来标识 **Amazon EC2** 计算资源。
3. 从 **Provider** 列表中，选择 **EC2**。
4. 在 **Description** 字段中，输入有助于区分资源供以后使用的信息。
5. 可选：从 **HTTP 代理** 列表，选择要连接到外部 **API** 服务的 **HTTP** 代理。您必须将 **HTTP** 代理添加到卫星，然后才能从此列表中选择代理。更多信息请参阅 [第 14.3 节“将 HTTP 代理用于计算资源”](#)。
6. 在 **Access Key** 和 **Secret Key** 字段中，输入 **Amazon EC2** 帐户的访问密钥。如需更多信息，请参阅 [Amazon 文档网站上为您的 AWS 帐户管理访问密钥](#)。
7. 可选：点击 **Load Regions** 按钮来填充 **Regions** 列表。
8. 从 **Region** 列表中，选择要使用的 **Amazon EC2** 区域或数据中心。
9. 单击位置选项卡，并确保选择要使用的位置，或添加其他位置。
10. 点 **Organizations** 选项卡，并确保选择要使用的组织，或添加其他组织。
11. 点 **Submit** 保存 **Amazon EC2** 连接。
12. 选择新计算资源，然后单击 **SSH 密钥** 选项卡，然后单击 **Download** 以保存要用于 **SSH** 身份验证的 **SSH** 密钥副本。在 [BZ1793138](#) 解析之前，您只能在创建 **Amazon EC2** 计算资源后立即下载 **SSH** 密钥的副本。如果需要 **SSH** 密钥在以后的阶段，请按照 [第 14.7 节“使用 SSH 连接到 Amazon EC2 实例”](#) 中的步骤操作。

对于 **CLI** 用户

使用 `hammer compute-resource create` 命令创建连接。使用 `--user` 和 `--password` 选项分别添加 `access key` 和 `secret key`。

```
# hammer compute-resource create --name "My_EC2" --provider "EC2" \  
--description "Amazon EC2 Public Cloud" --user "user_name" \  
--password "secret_key" --region "us-east-1" --locations "New York" \  
--organizations "My_Organization"
```

14.3. 将 HTTP 代理用于计算资源

在某些情况下，您使用的 EC2 计算资源可能需要特定的 HTTP 代理与 Satellite 通信。在卫星中，您可以创建 HTTP 代理，然后将 HTTP 代理分配给 EC2 计算资源。

但是，如果您在 **Administer > Settings** 中配置 Satellite 的 HTTP 代理，并为计算资源添加另一个 HTTP 代理，则您在 **Administer > Settings Settings** 中定义的 HTTP 代理会具有优先权。

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > HTTP Proxies**，然后选择 **New HTTP Proxy**。
2. 在 **Name** 字段中输入 HTTP 代理的名称。
3. 在 **URL** 字段中输入 HTTP 代理的 URL，包括端口号。
4. 可选：如果您的 HTTP 代理需要身份验证，请输入用户名和密码来向 HTTP 代理进行身份验证。
5. 单击 **Test Connection**，以确保您可以从 Satellite 连接到 HTTP 代理。
6. 单击位置选项卡并添加位置。
7. 点 **Organization** 选项卡并添加机构。

8. 点 **Submit**。

14.4. 将 AMAZON EC2 镜像添加到卫星服务器

Amazon EC2 使用基于镜像的置备来创建主机。您必须将镜像详情添加到卫星服务器。这包括访问详情和镜像位置。

流程

1. 在 **Satellite Web UI** 中，导航到 **Infrastructure > Compute Resources** 并选择 **Amazon EC2** 连接。
2. 单击 **Images** 选项卡，然后单击 **New Image**。
3. 在 **Name** 字段中输入一个名称来标识镜像以备将来使用。
4. 在 **Operating System** 列表中，选择与您要添加的镜像对应的操作系统。
5. 从 **Architecture** 列表中，选择操作系统的架构。
6. 在 **Username** 字段中输入镜像访问的 **SSH** 用户名。这通常是 **root** 用户。
7. 在 **Password** 字段中，为镜像访问输入 **SSH** 密码。
8. 在 **Image ID** 字段中，输入镜像的 **Amazon Machine Image (AMI) ID**。这通常采用以下格式：**ami-xxxxxxx**。
9. 可选：如果镜像支持用户数据输入，如 **cloud-init** 数据，请选择 **User Data** 复选框。如果您启用用户数据，则“完成脚本”将自动禁用。这也适用于反向：如果启用 **Finish** 脚本，这将禁用用户数据。
10. 可选：在 **IAM 角色** 字段中，输入用于创建镜像的 **Amazon** 安全角色。

11. 单击 **Submit** 以保存镜像详情。

对于 CLI 用户

使用 `hammer compute-resource image create` 命令创建镜像。使用 `--uuid` 字段将镜像位置的完整路径存储在 Amazon EC2 服务器上。

```
# hammer compute-resource image create --name "Test Amazon EC2 Image" \  
--operatingsystem "RedHat 7.2" --architecture "x86_64" --username root \  
--user-data true --uuid "ami-my_ami_id" --compute-resource "My_EC2"
```

14.5. 在计算配置文件中添加 AMAZON EC2 详情

您可以将 Amazon EC2 上实例的硬件设置添加到计算配置文件中。

流程

要添加硬件设置，请完成以下步骤：

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Compute Profiles** 并点您的配置集的名称，然后点 **EC2** 连接。
2. 从 **Flavor** 列表中，选择 EC2 上的硬件配置集以用于主机。
3. 从 **Image** 列表中，选择要用于基于镜像调配的镜像。
4. 从 **Availability zone** 列表中，选择要在所选 EC2 区域中使用的目标集群。
5. 从 **Subnet** 列表，为 EC2 实例添加子网。如果您的 VPC 用于置备新主机，请使用其子网。
6. 从 **Security Groups** 列表中，为要应用到主机的端口和 IP 地址选择基于云的访问规则。
7. 从 **Managed IP** 列表中，选择公共 IP 或一个 Private IP。

8. 单击 **Submit** 以保存计算配置文件。

对于 CLI 用户

计算配置文件 CLI 命令尚未在红帽卫星中实施。另外，您可以在主机创建过程中直接包含相同的设置。

14.6. 在 AMAZON EC2 上创建基于镜像的主机

Amazon EC2 置备过程从 Amazon EC2 服务器上的现有镜像创建主机。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Create Host**。
2. 在 **Name** 字段中输入主机的名称。
3. 从 **Host Group** 列表中，您可以选择一个主机组来填充新主机的大部分字段。
4. 从 **Deploy on** 列表中，选择 **EC2** 连接。
5. 从 **Compute Profile** 列表中，选择一个配置集来自动填充基于虚拟机的设置。
6. 单击 **Interface** 选项卡，然后单击主机的界面上的 **Edit**，然后验证字段是否填充了值。将 **Mac Address** 字段留空。卫星服务器会自动为主机上的第一个接口选择和 **IP 地址** 和 **Managed**、**Primary** 和 **Provision** 选项。
7. 单击 **Operating System** 选项卡，并确认所有字段都填充了值。
8. 点 **Virtual Machine** 选项卡，并确认所有字段都填充了值。
9. 单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。

10.

点 **Submit** 保存您的更改。

这个新主机条目会触发 **Amazon EC2 服务器创建实例**，并将预先存在的镜像用作新卷的基础。

对于 **CLI 用户**

使用 **hammer host create** 命令创建主机，并包含 **--provision-method** 镜像 以使用基于镜像的置备。

```
# hammer host create --name "ec2-test1" --organization "My_Organization" \
--location "New York" --hostgroup "Base" \
--compute-resource "My_EC2" --provision-method image \
--image "Test Amazon EC2 Image" --enabled true --managed true \
--interface "managed=true,primary=true,provision=true,subnet_id=EC2" \
--compute-attributes="flavor_id=m1.small,image_id=TestImage,availability_zones=us-east-1a,security_group_ids=Default,managed_ip=Public"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 **hammer 主机 create --help** 命令。

14.7. 使用 SSH 连接到 AMAZON EC2 实例

您可以使用 **SSH** 从卫星服务器远程连接到 **Amazon EC2** 实例。但是，要连接到您通过 **Red Hat Satellite** 置备的任何 **Amazon Web Services EC2** 实例，您必须首先访问与 **Foreman** 数据库中计算资源关联的私钥，并使用该密钥进行身份验证。

要找到私钥并使用 **SSH** 连接到 **Amazon EC2** 服务器，请完成以下步骤：

1.

要定位计算资源列表，请在 **Satellite** 服务器基础系统中输入以下命令，并记录您要使用的计算资源的 **ID**：

```
# hammer compute-resource list
```

2.

将用户切换到 **postgres** 用户：

```
# su - postgres
```

3.

启动 **postgres shell**：

```
$ psql
```

4. 以用户 **postgres** 的身份连接到 **Foreman** 数据库：

```
# postgres=# \c foreman
```

5. 从 **compute_resource_id = 3** 的 **key_pairs** 选择 **secret**：

```
# select secret from key_pairs where compute_resource_id = 3; secret
```

6. 从 **-----BEGIN RSA PRIVATE KEY-----** 后复制密钥，直到 **-----END RSA PRIVATE KEY-----**。

7. 创建 **.pem** 文件并将您的密钥粘贴到文件中：

```
# vim Keyname.pem
```

8. 确保限制对 **.pem** 文件的访问：

```
# chmod 600 Keyname.pem
```

9. 要连接到 **Amazon EC2** 实例，请输入以下命令：

```
ssh -i Keyname.pem ec2-user@example.aws.com
```

14.8. 为 AMAZON WEB SERVICE EC2 环境配置完成模板

您可以在 **Amazon EC2** 环境中置备 **Red Hat Enterprise Linux** 实例期间使用 **Red Hat Satellite** 完成模板。

如果您要使用带有 **SSH** 的 **Finish** 模板，卫星必须位于 **EC2** 环境中，且位于正确安全组中。卫星目前直接执行 **SSH** 完成调配，而不是使用胶囊服务器。如果卫星服务器不在 **EC2** 中，则 **EC2** 虚拟机报告内部 IP，而不是访问它所需的外部 IP。

要为 **Amazon EC2** 配置结束模板，请完成以下步骤：

1. 在 Red Hat Satellite Web UI 中，进入 Hosts > Provisioning Templates。
2. 在 Provisioning Templates 页面中，在搜索字段中输入 Kickstart default finish，然后单击 Search。
3. 在 Kickstart 默认完成 模板上，选择 Clone。
4. 在 Name 字段中输入模板的唯一名称。
5. 在模板中，为每个命令添加需要 root 特权的前缀 sudo，但 subscription-manager register 和 yum 命令除外，或者添加以下行，以 sudo 用户身份运行整个模板：

```
sudo -s << EOS
__Template__Body__
EOS
```

6. 点击关联选项卡，并将模板与您要使用的红帽企业 Linux 操作系统相关联。
7. 单击位置选项卡，再添加主机所在的位置。
8. 单击组织选项卡，再添加主机所属的组织。
9. 进行所需的任何其他自定义或更改，然后点 Submit 以保存模板。
10. 进入 Hosts > Operating systems 并选择您要用于您的主机的操作系统。
11. 点 Templates 选项卡，从 Finish Template 列表中选择您的 finish 模板。
12. 进入 Hosts > Create Host 并输入您要创建的主机的信息。

13. 单击 **Parameters** 选项卡，再前往 **Host** 参数。
14. 在 **Host** 参数中，单击 **Add Parameter** 按钮三次，以添加三个新参数字段。添加以下三个参数：
 - a. 在 **Name** 字段中输入 `remote_execution_ssh_keys`。在对应的 **Value** 字段中，输入 `cat /usr/share/foreman-proxy/.ssh/id_rsa_foreman_proxy.pub` 的输出。
 - b. 在 **Name** 字段中输入 `remote_execution_ssh_user`。在对应的 **Value** 字段中，输入 `ec2-user`。
 - c. 在 **Name** 字段中输入 `activation_keys`。在对应的 **Value** 字段中，输入您的激活码。
15. 单击 **Submit** 以保存更改。

14.9. 有关 AMAZON WEB SERVICES 和 SATELLITE 的更多信息

有关如何在 Amazon Web Services EC2 上找到 Red Hat Gold Images 的详情，请参考 [如何在 AWS EC2 上提取 Red Hat Cloud Access Gold 镜像](#)。

有关如何在 Linux 中安装和使用 Amazon Web Service 客户端的详情，请参考 Amazon Web Services 文档中的 [在 Linux 上安装 AWS 命令行界面](#)。

有关在 Amazon Web Services 中导入和导出虚拟机的详情，请参考 Amazon Web Services 文档中的 [VM Import/Export](#)。

第 15 章 在 GOOGLE COMPUTE ENGINE 上置备云实例

红帽卫星可以与 Google Compute Engine (GCE)交互，包括创建新虚拟机和控制其电源管理状态。您只能将 Red Hat 支持的金级镜像用于 Satellite 创建 GCE 主机。

前提条件

- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。
- 在 GCE 项目中，使用所需的 IAM Compute 角色配置服务帐户。如需更多信息，请参阅 [GCE 文档中的 Compute Engine IAM 角色](#)。
- 在 GCE 项目范围元数据中，将 `enable-oslogin` 设置为 `FALSE`。如需更多信息，请参阅 [GCE 文档中的启用或禁用 OS Login](#) 部分。
- 可选：如果要将在 Puppet 与 GCE 主机搭配使用，请导航到 `Administer > Settings > Puppet` 并启用 `Use UUID for certificate` 设置来配置 Puppet 来使用一致的 Puppet 证书 ID。
- 根据您的需要，将 `finish` 或 `user_data` 调配模板与您要使用的操作系统关联。有关调配模板的更多信息，请参阅 [第 2.11 节“调配模板”](#)。

流程概述

1. [第 15.1 节“在 Satellite 服务器中添加 Google Compute Engine 连接”](#)。
2. [第 15.2 节“在 Satellite 服务器中添加 Google Compute Engine 镜像”](#)。
3. [第 15.3 节“在 Compute Profile 中添加 Google Compute Engine Details”](#)。
4. [第 15.4 节“在 Google Compute Engine 上创建基于镜像的主机”](#)。

15.1. 在 SATELLITE 服务器中添加 GOOGLE COMPUTE ENGINE 连接

使用这个步骤将 Google Compute Engine (GCE) 添加为 Satellite 中的计算资源。

流程

1. 在 GCE 中，以 JSON 格式生成服务帐户密钥，并将此文件上传到卫星服务器上的 `/usr/share/foreman/` 目录中。

2. 在 Satellite 服务器中，将服务帐户密钥的所有者改为 foreman 用户：

```
# chown foreman /usr/share/foreman/gce_key.json
```

3. 为服务帐户密钥配置权限，以确保该文件可读：

```
# chmod 0600 /usr/share/foreman/gce_key.json
```

4. 为服务帐户密钥恢复 SELinux 上下文：

```
# restorecon -vv /usr/share/foreman/gce_key.json
```

5. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources，点 Create Compute Resource。

6. 在 Name 字段中输入计算资源的名称。

7. 从 Provider 列表中，选择 Google。

8. 可选：在 Description 字段中输入资源描述。

9. 在 Google Project ID 字段中，输入项目 ID。

10. 在 Client Email 字段中，输入客户端电子邮件。

11. 在 **Certificate Path** 字段中，输入服务帐户密钥的路径。例如：
`/usr/share/foreman/gce_key.json`。
12. 点 **Load Zones** 从 GCE 环境中填充区列表。
13. 从 **Zone** 列表中，选择要使用的 GCE 区域。
14. 点 **Submit**。

对于 CLI 用户

1. 在 GCE 中，以 JSON 格式生成服务帐户密钥，并将此文件上传到卫星服务器上的 `/usr/share/foreman/` 目录中。

2. 在 Satellite 服务器中，将服务帐户密钥的所有者改为 **foreman** 用户：

```
# chown foreman /usr/share/foreman/gce_key.json
```

3. 为服务帐户密钥配置权限，以确保该文件可读：

```
# chmod 0600 /usr/share/foreman/gce_key.json
```

4. 为服务帐户密钥恢复 SELinux 上下文：

```
# restorecon -vv /usr/share/foreman/gce_key.json
```

5. 使用 **hammer compute-resource create** 命令将 GCE 计算资源添加到 Satellite 中：

```
# hammer compute-resource create --name 'gce_cr' \  
--provider 'gce' \  
--project 'gce_project_id' \  
--key-path 'gce_key.json' \  
--zone 'us-west1-b' \  
--email 'gce_email'
```

15.2. 在 SATELLITE 服务器中添加 GOOGLE COMPUTE ENGINE 镜像

要使用基于镜像的置备来创建主机，您必须将镜像的相关信息（如访问详细信息和镜像位置）添加到您的卫星服务器。

流程

1. 在 **Satellite Web UI** 中，进入 **Infrastructure > Compute Resources** 并点 **Google Compute Engine** 连接的名称。
2. 点 **Create Image**。
3. 在 **Name** 字段中输入镜像的名称。
4. 从 **Operating System** 列表中，选择镜像的基本操作系统。
5. 从 **Architecture** 列表中，选择操作系统架构。
6. 在 **Username** 字段中输入镜像访问的 **SSH** 用户名。指定 **root** 以外的用户，因为 **root** 用户无法使用 **SSH** 密钥连接到 **GCE** 实例。用户名必须以字母开头，并且由小写字母和数字组成。
7. 从 **Image** 列表中，从 **Google Compute Engine** 计算资源中选择镜像。
8. 可选：如果镜像支持用户数据输入，如 **cloud-init** 数据，请选择 **User Data** 复选框。
9. 单击 **Submit** 以保存镜像详情。

对于 CLI 用户

- 使用 **hammer compute-resource image create** 命令创建镜像。使用 **--username** 选项指定 **root** 以外的用户，因为 **root** 用户无法使用 **SSH** 密钥连接到 **GCE** 实例。用户名必须以字母开头，并且由小写字母和数字组成。

```
# hammer compute-resource image create \  
--name 'gce_image_name' \  
--compute-resource 'gce_cr' \  

```

```
--operatingsystem-id 1 \  
--architecture-id 1 \  
--uuid '3780108136525169178' \  
--username 'admin'
```

15.3. 在 COMPUTE PROFILE 中添加 GOOGLE COMPUTE ENGINE DETAILS

使用这个流程将 GCE 硬件设置添加到计算配置集中。当使用此 compute 配置集在 GCE 上创建主机时，这些设置会被自动填充。

流程

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Profiles。
2. 在 Compute Profiles 窗口中，单击现有计算配置文件的名称，或者单击 Create Compute Profile，输入 Name，再单击 Submit。
3. 点 GCE 计算资源的名称。
4. 从 Machine Type 列表中，选择要用于置备的机器类型。
5. 从 Image 列表中，选择要用于调配的镜像。
6. 从 Network 列表中，选择 GCE 网络以用于置备。
7. 可选：选择关联 Ephemeral External IP 复选框，以分配卫星用于与主机通信的动态临时 IP 地址。重启主机时，此公共 IP 地址会改变。如果需要永久 IP 地址，请在 GCE 上保留静态公共 IP 地址并将其附加到主机。
8. 在 Size (GB) 字段中，输入要在主机上创建的存储大小。
9. 单击 Submit 以保存计算配置文件。

对于 CLI 用户

1. **创建用于 GCE 计算资源的计算配置集：**

```
# hammer compute-profile create --name gce_profile
```

2. **在计算配置集中添加 GCE 详情。**

```
# hammer compute-profile values create --compute-profile gce_profile \  
--compute-resource 'gce_cr' \  
--volume "size_gb=20" \  
--compute-attributes "machine_type=f1-micro,associate_external_ip=true,network=default"
```

15.4. 在 GOOGLE COMPUTE ENGINE 上创建基于镜像的主机

在 Satellite 中，您可以使用 Google Compute Engine 置备从现有镜像创建主机。新主机条目会触发 Google Compute Engine 服务器，以使用预先存在的镜像作为新卷的基础创建实例。

流程

1. **在 Satellite Web UI 中，导航到 Hosts > Create Host。**
2. **在 Name 字段中输入主机的名称。**
3. **点 Organization 和 Location 选项卡，以确保置备上下文自动设置为当前上下文。**
4. **从 Host Group 列表中，选择您要用于填充表单的主机组。**
5. **在 Deploy on 列表中，选择 Google Compute Engine 连接。**
6. **从 Compute Profile 列表中，选择一个配置集来自动填充虚拟机设置。**
7. **从 Lifecycle Environment 列表中，选择环境。**
8. **点 Interfaces 选项卡，再单击主机接口上的 Edit。**

9.

验证字段是否已自动填充，特别是以下项目：

- **Host** 选项卡中的 **Name** 将成为 **DNS** 名称。
- **MAC 地址** 字段为空。Google Compute Engine 在置备过程中为主机分配 **MAC 地址**。
- **卫星服务器**会自动为新主机分配 **IP 地址**。
- **Domain** 字段填充了所需域。
- 对于主机上的第一个接口，会自动选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有，请选择它们。

10.

单击 **Operating System** 选项卡，并确认所有字段都自动包含值。

11.

单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。

12.

单击 **Virtual Machine** 选项卡，并确认这些设置已填充了主机组和计算配置集的详细信息。修改这些设置以符合您的需要。

13.

单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。

14.

单击 **Submit** 以保存主机条目。

对于 CLI 用户

- 使用 `hammer host create` 命令创建主机，并包含 `--provision-method` 镜像。将以下示例中的值替换为您环境的相应值。

```
# hammer host create \  
--name "GCE_VM" \  
--organization "Your_Organization" \  
--location "Your_Location" \  

```

```
--compute-resource gce_cr_name  
--compute-profile "gce_profile_name" \  
--provision-method 'image' \  
--image gce_image_name \  
--root-password "your_root_password" \  
--interface "type=interface,domain_id=1,managed=true,primary=true,provision=true" \  
--puppet-environment-id 1 \  
--puppet-ca-proxy-id 1 \  
--puppet-proxy-id 1 \  
--architecture x86_64 \  
--operatingsystem "operating_system_name"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 **hammer 主机 create --help** 命令。

第 16 章 在 MICROSOFT AZURE RESOURCE MANAGER 上置备云实例

红帽卫星可以与 Microsoft Azure 资源管理器交互，包括创建新虚拟机和控制其电源管理状态。仅支持创建 Azure 主机的基于镜像的置备。这包括使用 Marketplace 镜像、自定义镜像和共享镜像 gallery 置备。

如需有关 Azure Resource Manager 概念的更多信息，请参阅 [Azure Resource Manager 文档](#)。

请注意，Satellite 不支持 Azure Government。

前提条件

- 为 Red Hat Enterprise Linux 同步内容仓库。有关更多信息，请参阅 [内容管理指南中的同步红帽存储库](#)。
- 主机注册的激活码。有关更多信息，请参阅 [内容管理指南中的创建激活码](#)。
- 确保具有创建 Azure Active Directory 应用程序的正确权限。如需更多信息，请参阅 Microsoft 身份平台 ([Azure Active Directory for developers](#) 的 Azure Active Directory) 中的 Check Azure AD 权限。
- 您必须创建和配置 Azure Active Directory 应用程序以及服务原则，以获取应用程序或客户端 ID、目录或租户 ID 和客户端 Secret。如需更多信息，请参阅使用 [门户创建 Azure AD 应用程序和服务主体](#)，该主体可以访问 Microsoft 身份平台 (Azure Active Directory for developers) 文档中的资源。
- 可选：如果要与 Puppet 与 Azure 主机搭配使用，请导航到 Administer > Settings > Puppet，并为证书设置启用 Use UUID 来配置 Puppet 来使用一致的 Puppet 证书 ID。
- 根据您的需要，将 finish 或 user_data 调配模板与您要使用的操作系统关联。有关调配模板的更多信息，请参阅 [第 2.11 节“调配模板”](#)。
- 可选：如果您希望虚拟机使用静态私有 IP 地址，请在 Satellite 中创建一个与 Azure 子网地址匹配的 Network Address 字段的子网。
-

在创建 RHEL BYOS 镜像前，您必须接受 Azure CLI 或门户中的镜像术语，以便镜像可用于创建和管理您的订阅的虚拟机。

流程概述

1. [第 16.1 节“向 Satellite 服务器添加 Microsoft Azure Resource Manager 连接”](#)。
2. [第 16.2 节“将 Microsoft Azure 资源管理器镜像添加到 Satellite 服务器”](#)。
3. [第 16.3 节“在 Compute Profile 中添加 Microsoft Azure Resource Manager Details”](#)。
4. [第 16.4 节“在 Microsoft Azure Resource Manager 上创建基于镜像的主机”](#)。

16.1. 向 SATELLITE 服务器添加 MICROSOFT AZURE RESOURCE MANAGER 连接

使用此流程将 Microsoft Azure 资源管理器添加为 Satellite 中的计算资源。请注意，您必须为要使用的每个 Azure Resource Manager 区域添加单独的计算资源。

流程

1. 在 Satellite Web UI 中，导航到 Infrastructure > Compute Resources，点 Create Compute Resource。
2. 在 Name 字段中输入计算资源的名称。
3. 从 Provider 列表中，选择 Azure Resource Manager。
4. 可选：在 Description 字段中输入资源描述。
5. 在 Client ID 字段中，输入应用程序或客户端 ID。
6. 在 Client Secret 字段中，输入客户端 secret。

7. 在 **Subscription ID** 字段中，输入订阅 ID。
8. 在 **Tenant ID** 字段中，输入 **Directory** 或 **租户 ID**。
9. 单击 **Load Regions**。这会测试您的与 **Azure** 资源管理器的连接是否成功，并加载您的订阅中的区域。
10. 从 **Azure Region** 列表中，选择要使用的 **Azure** 区域。
11. 点 **Submit**。

对于 CLI 用户

- 使用 `hammer compute-resource create` 命令将 **Azure** 计算资源添加到 **Satellite**。请注意，`--region` 选项的值必须为小写，且不得包含特殊字符。

```
# hammer compute-resource create --name azure_cr_name \  
--provider azurearm \  
--tenant tenant-id \  
--app-ident client-id \  
--secret-key client-secret \  
--sub-id subscription-id \  
--region 'region'
```

16.2. 将 MICROSOFT AZURE 资源管理器镜像添加到 SATELLITE 服务器

要使用基于镜像的置备来创建主机，您必须将镜像的相关信息（如访问详细信息和镜像位置）添加到您的卫星服务器。

流程

1. 在 **Satellite Web UI** 中，进入 **Infrastructure > Compute Resources** 并点 **Microsoft Azure Resource Manager** 连接的名称。
2. 点 **Create Image**。

3. 在 **Name** 字段中输入镜像的名称。
4. 从 **Operating System** 列表中，选择镜像的基本操作系统。
5. 从 **Architecture** 列表中，选择操作系统架构。
6. 在 **Username** 字段中输入镜像访问的 SSH 用户名。您不能使用 **root** 用户。
7. 可选：在 **Password** 字段中，输入要进行身份验证的密码。
8. 在 **Azure Image Name** 字段中输入镜像名称，格式为 **prefix://UUID**。
 - 对于自定义镜像，请使用前缀 **custom**。例如，**custom://image-name**。
 - 对于共享的 **gallery** 镜像，使用前缀 **gallery**。例如：**gallery://image-name**。
 - 对于公共和 **RHEL** 提供您的 **Own** 订阅(BYOS)镜像，请使用前缀 **marketplace**。例如：**marketplace://OpenLogicCentOS:7.5:latest**。

如需更多信息，请参阅使用 [Azure CLI 在 Azure Marketplace 中查找 Linux 虚拟机镜像](#)。
9. 可选：如果镜像支持用户数据输入，如 **cloud-init** 数据，请选择 **User Data** 复选框。
10. 单击 **Submit** 以保存镜像详情。

对于 CLI 用户

- 使用 **hammer compute-resource image create** 命令创建镜像。请注意，您为镜像输入的用户名必须与创建使用此镜像时使用的用户名相同。创建镜像时 **--password** 选项是可选的。您不能使用 **root** 用户。

```
# hammer compute-resource image create \  
--name Azure_image_name \  
--compute-resource azure_cr_name \  
--uuid 'marketplace://RedHat:RHEL:7-RAW:latest' \  
--username 'azure_username' \  
--user-data no
```

16.3. 在 COMPUTE PROFILE 中添加 MICROSOFT AZURE RESOURCE MANAGER DETAILS

使用这个流程将 Azure 硬件设置添加到计算配置集中。当使用此 compute 配置集在 Azure 上创建主机时，这些设置会被自动填充。

流程

1. 在 Satellite Web UI 中，导航到 **Infrastructure > Compute Profiles**。
2. 在 **Compute Profiles** 窗口中，单击现有计算配置文件的名称，或者单击 **Create Compute Profile**，输入 **Name**，再单击 **Submit**。
3. 点 **Azure** 计算资源的名称。
4. 从 **Resource group** 列表中，选择要置备的资源组。
5. 在 **VM Size** 列表中，选择要置备的虚拟机大小。
6. 从 **Platform** 列表中，选择 **Linux**。Satellite 不支持配置运行 Microsoft 操作系统的虚拟机。
7. 在 **Username** 字段中输入要进行身份验证的用户名。请注意，您输入用于 compute 配置集的用户名必须与创建镜像时使用的用户名相同。
8. 要验证用户，请使用以下选项之一：
 - 要使用密码进行身份验证，请在 **Password** 字段中输入密码。

- 要使用 SSH 密钥进行身份验证，请在 SSH Key 字段中输入 SSH 密钥。
9. 可选：如果您希望虚拟机使用高级虚拟机磁盘，请选择 Premium OS Disk 复选框。
 10. 从 OS 磁盘缓存 列表中，选择磁盘缓存设置。
 11. 可选：在 Custom Script Command 字段中输入在虚拟机置备时要在虚拟机上执行的命令。
 12. 可选：如果要在置备完成后运行自定义脚本，在 Comma separated file URIs 字段中，输入要使用的脚本的逗号分隔列表。脚本必须首先包含 sudo，因为红帽卫星将文件下载到主机和主机和脚本上的 /var/lib/oaagent/custom-script/download/0/ 目录，因此需要执行 sudo 权限。
 13. 可选：如果您要在虚拟机上创建额外卷，点 Add Volume 按钮，在 GB 中输入 Size 并选择 Data Disk Cache 方法。
 14. 单击添加接口。
 15. 从 Azure Subnet 列表中，选择要置备的 Azure 子网。
 16. 从 公共 IP 列表中，选择公共 IP 设置。
 17. 可选：如果您希望虚拟机使用静态私有 IP，请选择 Static Private IP 复选框。
 18. 点 Submit。

对于 CLI 用户

1. 创建用于 Azure Resource Manager 计算资源的计算配置集：

```
# hammer compute-profile create --name compute_profile_name
```

- 2.

将 **Azure** 详情添加到 **compute 配置集**。使用 **用户名 设置**，为镜像访问输入 **SSH 用户名**。请注意，您输入用于 **compute 配置集** 的用户名必须与创建镜像时使用的用户名相同。

```
# hammer compute-profile values create \
--compute-profile "compute_profile_name" \
--compute-resource azure_cr_name \
--compute-
attributes="resource_group=resource_group,vm_size=Standard_B1s,username=azure_user,
password=azure_password,platform=Linux,script_command=touch /var/tmp/text.txt" \
--
interface="compute_public_ip=Dynamic,compute_network=mysubnetID,compute_private_ip=fa
se"
--volume="disk_size_gb=5,data_disk_caching=None"
```

可选：如果要在置备后在虚拟机上运行脚本，请指定以下设置：

- 要直接输入脚本，使用 **script_command** 设置，在置备的虚拟机上输入一个要执行的命令。
- 要从 **URI**（使用 **script_uris** 设置）运行脚本，请输入要使用的脚本 **URI**。脚本必须首先包含 **sudo**，因为红帽卫星将文件下载到主机上的 **/var/lib/waagent/custom-script/download/0/** 目录，因此脚本需要执行 **sudo** 权限。

16.4. 在 MICROSOFT AZURE RESOURCE MANAGER 上创建基于镜像的主机

在 **Satellite** 中，您可以使用 **Microsoft Azure 资源管理器** 置备从现有镜像创建主机。新主机条目触发 **Microsoft Azure 资源管理器** 服务器，以使用预先存在的镜像作为新卷的基础来创建实例。

流程

1. 在 **Satellite Web UI** 中，导航到 **Hosts > Create Host**。
2. 在 **Name** 字段中输入主机的名称。
3. 点 **Organization** 和 **Location** 选项卡，以确保置备上下文自动设置为当前上下文。
4. 从 **Host Group** 列表中，选择您要用于填充表单的主机组。

5. 从 **Deploy on** 列表中, 选择 **Microsoft Azure Resource Manager** 连接。
6. 从 **Compute Profile** 列表中, 选择一个配置集来自动填充虚拟机设置。
7. 从 **Lifecycle Environment** 列表中, 选择 环境。
8. 点 **Interfaces** 选项卡, 再单击主机接口上的 **Edit**。
9. 验证字段是否已自动填充, 特别是以下项目 :
 - **Host** 选项卡中的 **Name** 将成为 **DNS** 名称。
 - **MAC 地址** 字段为空。 **Microsoft Azure Resource Manager** 在置备过程中为主机分配 **MAC 地址**。
 - **Azure Subnet** 字段会填充所需的 **Azure** 子网。
 - 对于主机上的第一个接口, 会自动选择 **Managed**、**Primary** 和 **Provision** 选项。如果没有, 请选择它们。
10. 可选 : 如果要使用静态私有 IP 地址, 请从 **IPv4 Subnet** 列表中选择 **Satellite** 子网与 **Azure** 子网地址匹配。在 **IPv4 Address** 字段中, 在 **Azure** 子网范围内输入一个 **IPv4** 地址。
11. 单击 **Operating System** 选项卡, 并确认所有字段都自动包含值。
12. 对于 **Provisioning** 方法, 请确保选择了基于 镜像。
13. 在 **Image** 列表中, 选择您要用于置备的 **Azure Resource Manager** 镜像。

14. 在 **Root Password** 字段中，输入 **root** 密码进行验证。
15. 单击 **Provisioning** 模板中的 **Resolve**，以检查新主机可识别要使用的正确调配模板。
16. 单击 **Virtual Machine** 选项卡，并确认这些设置已填充了主机组和计算配置集的详细信息。修改这些设置以符合您的需要。
17. 单击 **Parameters** 选项卡，并确保存在一个提供激活密钥的参数。如果没有，请添加激活码。
18. 单击 **Submit** 以保存主机条目。

对于 CLI 用户

- 使用 **hammer host create** 命令创建主机，并包含 **--provision-method** 镜像。将以下示例中的值替换为您环境的相应值。

```
# hammer host create \  
--name="Azure_VM" \  
--organization "Your_Organization" \  
--location "Your_Location" \  
--compute-resource azure_cr_name \  
--compute-profile "compute_profile_name" \  
--provision-method 'image' \  
--image Azure_image_name \  
--domain domain_name \  
--architecture x86_64 \  
--operatingsystem "operating_system_name"
```

有关此计算资源的额外主机创建参数的更多信息，请输入 **hammer 主机 create --help** 命令。

附录 A. 调配示例的初始化脚本

如果您尚未遵循 Red Hat Satellite 6 内容管理指南中的示例，您可以使用以下初始化脚本创建用于置备示例的环境。

创建一个脚本文件(`content-init.sh`)并包含以下内容：

```
#!/bin/bash

MANIFEST=$1

# Import the content from Red Hat CDN
hammer organization create --name "ACME" --label "ACME" \
--description "Our example organization for managing content."
hammer subscription upload --file ~/$MANIFEST --organization "ACME"
hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (RPMs)" \
--releasever "7Server" --basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" --organization "ACME"
hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (Kickstart)" \
--releasever "7Server" --basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" --organization "ACME"
hammer repository-set enable \
--name "Red Hat Satellite Tools 6.9 (for RHEL 7 Server) (RPMs)" \
--basearch "x86_64" --product "Red Hat Enterprise Linux Server" \
--organization "ACME"
hammer product synchronize --name "Red Hat Enterprise Linux Server" \
--organization "ACME"

# Create our application life cycle
hammer lifecycle-environment create --name "Development" \
--description "Environment for ACME's Development Team" \
--prior "Library" --organization "ACME"
hammer lifecycle-environment create --name "Testing" \
--description "Environment for ACME's Quality Engineering Team" \
--prior "Development" --organization "ACME"
hammer lifecycle-environment create --name "Production" \
--description "Environment for ACME's Product Releases" \
--prior "Testing" --organization "ACME"

# Create and publish our Content View
hammer content-view create --name "Base" \
--description "Base operating system" \
--repositories "Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server,Red Hat Satellite Tools 6.9
for RHEL 7 Server RPMs x86_64" \
--organization "ACME"
hammer content-view publish --name "Base" \
--description "Initial content view for our operating system" \
--organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 \
```

```
--to-lifecycle-environment "Development" --organization "ACME"  
hammer content-view version promote --content-view "Base" --version 1 \  
--to-lifecycle-environment "Testing" --organization "ACME"  
hammer content-view version promote --content-view "Base" --version 1 \  
--to-lifecycle-environment "Production" --organization "ACME"
```

在脚本上设置可执行权限：

```
# chmod +x content-init.sh
```

从红帽客户门户网站下载订阅清单的副本，并在清单上运行脚本：

```
# ./content-init.sh manifest_98f4290e-6c0b-4f37-ba79-3a3ec6e405ba.zip
```

这会导入本指南中置备示例所需的红帽内容。

附录 B. 置备 FIPS-COMPLIANT HOSTS

卫星支持与国家标准与技术安全要求相符的调配主机，用于 [Cryptographic Modules](#) 标准，引用号 FIPS 140-2，这里称为 FIPS。

要启用与 FIPS 兼容的主机置备，请完成以下任务：

- 更改操作系统的置备密码哈希算法
- 创建主机组并设置主机组以启用 FIPS

有关创建主机组的更多信息，请参阅管理 [主机指南中的创建主机组](#)。

置备的主机应用 FIPS 兼容设置。要确认启用了这些设置，请完成以下步骤 [第 B.3 节“验证 FIPS 模式是否已启用”](#)。

B.1. 更改 PROVISIONING PASSWORD HASHING ALGORITHM

要置备 FIPS 兼容主机，您必须首先将您在置备中使用的密码哈希算法设置为 SHA256。对于您要部署为 FIPS 的每个操作系统，必须应用此配置设置。

1. 识别操作系统 ID。

```
$ hammer os list
```

2. 更新每个操作系统的密码哈希值。

```
$ hammer os update --title Operating_System \  
--password-hash SHA256
```

3. 使用 TITLE 列中的匹配值，为每个操作系统重复此命令：

```
$ hammer os update --title "RedHat version_number" \  
--password-hash SHA256
```

请注意，您不能使用以逗号分隔的值列表。

B.2. 设置 FIPS-ENABLED PARAMETER

要置备 FIPS 兼容主机，您必须创建一个主机组，并将主机组 `fips_enabled` 设为 `true`。如果没有将其设置为 `true`，或者不存在 FIPS 特定的更改，则特定于 FIPS 的更改不会应用到系统。在置备主机或主机组时，您可以设置此参数。

要在置备主机时设置此参数，请将 `--parameters fips_enabled=true` 附加到 Hammer 命令。

```
$ hammer hostgroup set-parameter --name fips_enabled \  
--value 'true' \  
--hostgroup prod_servers
```

如需更多信息，请参阅 `hammer hostgroup set-parameter --help` 命令的输出。

B.3. 验证 FIPS 模式是否已启用

要验证这些 FIPS 合规更改是否成功，您必须置备主机并检查其配置。

1. 以 `root` 或具有管理员级帐户登录主机。
2. 输入以下命令：

```
$ cat /proc/sys/crypto/fips_enabled
```

值 `1` 确认启用了 FIPS 模式。

附录 C. 为 RED HAT SATELLITE 构建云镜像

使用本节来构建和注册镜像到 Red Hat Satellite。

您可以使用预配置的 Red Hat Enterprise Linux KVM 客户机 QCOW2 镜像：

- [Latest RHEL 7 KVM Guest Image](#)
- [Latest RHEL 6 KVM Guest Image](#)

这些镜像包含 cloud-init。要正常工作，必须使用 ec2 兼容元数据服务来置备 SSH 密钥。



注意

对于 KVM 客户机镜像：

- 镜像中的 root 帐户被禁用，但 sudo 访问权限被授予一个名为 cloud-user 的特殊用户。
- 此镜像没有设置 root 密码。

root 密码在 /etc/shadow 中锁定，只需将 !! 放在第二个字段中。

如果要创建自定义 Red Hat Enterprise Linux 镜像，请参阅 [创建 Red Hat Enterprise Linux 7 镜像并创建 Red Hat Enterprise Linux 6 镜像](#)。

C.1. 创建自定义 RED HAT ENTERPRISE LINUX 镜像

先决条件：

- 使用 Linux 主机机器创建镜像。在这个示例中，使用 Red Hat Enterprise Linux 7 Workstation。

- 在工作站上使用 `virt-manager` 完成此步骤。如果您在远程服务器中创建镜像，使用 `virt-manager` 从工作站连接服务器。
- Red Hat Enterprise Linux 7 或 6 ISO 文件（请参阅 [Red Hat Enterprise Linux 7.4 Binary DVD](#) 或者 [Red Hat Enterprise Linux 6.9 Binary DVD](#)）。

有关安装 Red Hat Enterprise Linux Workstation 的详情，请参考 [Red Hat Enterprise Linux 7 安装指南](#)。

在创建自定义镜像前，先安装以下软件包：

- 安装 `libvirt`、`qemu-kvm` 和图形工具：

```
[root@host]# yum install virt-manager virt-viewer libvirt qemu-kvm
```

- 安装以下命令行工具：

```
[root@host]# yum install virt-install libguestfs-tools-c
```



注意

在以下步骤中，在托管 `libvirt` 环境的工作站输入 `[root@host]#` 提示符。

C.2. 创建 RED HAT ENTERPRISE LINUX 7 镜像

使用本节，使用 Red Hat Enterprise Linux 7 ISO 文件以 QCOW2 格式创建镜像。

1. 使用您的 Web 浏览器，将 Red Hat Enterprise Linux 二进制 ISO 文件下载到临时位置，例如 `Downloads` 目录。
2. 将 Red Hat Enterprise Linux 二进制文件 ISO 文件复制到 `/var/lib/libvirt/images/` 目录中。

```
[root@host]# cp ~/home/user/Downloads/rhel-server-7.4-x86_64-dvd.iso
/var/lib/libvirt/images/
```

3. 验证 **virtbr0** 是否是虚拟网桥：

```
[root@host]# ip a
```

4. 启动 **libvirtd**：

```
[root@host]# systemctl start libvirtd
```

5. 进入 **/var/lib/libvirt/images/** 目录：

```
[root@host]# cd /var/lib/libvirt/images/
```

6. 准备 **QEMU** 镜像：

```
[root@host]# qemu-img create -f qcow2 rhel7.qcow2 8G
```

7. 使用 **virt-install** 开始安装。使用以下示例作为指南：

```
[root@host]# virt-install --virt-type qemu --name rhel7 --ram 2048 \
--cdrom rhel-server-7.4-x86_64-dvd.iso \
--disk rhel7.qcow2,format=qcow2 \
--network=bridge:virbr0 --graphics vnc,listen=0.0.0.0 \
--noautoconsole --os-type=linux --os-variant=rhel7
```



注意

对于 **GUI** 用户，如果实例没有自动启动，请输入 **virt-manager** 命令以查看控制台：

```
[root@host]# virt-manager
```

8. 按照 **Red Hat Enterprise Linux 安装向导** 的步骤进行操作。

- a. 对于安装源，在 **Red Hat Satellite** 中添加 **HTTP** 链接到您的存储库，如

satellite.example.com/pub/export/RHEL7/content/dist/rhel/server/7/7Server/x86_64/os/

- b. 对于安装使用的设备类型，请选择 **Auto-detected 安装介质**。
 - c. 对于安装目的地的类型，请选择 **Local Standard Disks**。
 - d. 对于其他存储选项，请选择 **Automatically configure partitioning**。
 - e. 对于软件选择，请选择 **Minimal Install**。
 - f. 将网络接口设置为 **ON**，以确保接口在系统启动时激活。
 - g. 输入主机名，然后单击“应用”。
 - h. 输入 **root** 密码。
9. 安装完成后，重新引导实例并以 **root** 用户身份登录。
10. 确认网络接口已启动并且分配了 IP 地址：
- ```
ip a
```
11. 确认主机名正确：
- ```
# hostname
```
12. 创建 `/etc/NetworkManager/conf.d/XX-cloud-image.conf` 文件，其中 **XX** 是指示优先顺序的两位数。在文件中添加以下内容：

```
[main]  
dns=none
```

13.

继续 [配置主机以进行注册](#)。

C.3. 创建 RED HAT ENTERPRISE LINUX 6 镜像

使用本节，使用 Red Hat Enterprise Linux 6 ISO 文件以 QCOW2 格式创建镜像。

1.

使用 `virt-install` 开始安装：

```
[root@host]# qemu-img create -f qcow2 rhel6.qcow2 4G
[root@host]# virt-install --connect=qemu:///system --network=bridge:virbr0 \
--name=rhel6 --os-type linux --os-variant rhel6 \
--disk path=rhel6.qcow2,format=qcow2,size=10,cache=none \
--ram 4096 --vcpus=2 --check-cpu --accelerate \
--hvm --cdrom=rhel-server-6.8-x86_64-dvd.iso
```

这会启动实例并启动安装过程。



注意

如果实例没有自动启动，输入 `virt-viewer` 命令来查看控制台：

```
[root@host]# virt-viewer rhel6
```

2.

按照如下所示设置虚拟机：

a.

在初始安装程序引导菜单中，选择 **安装或升级现有系统** 选项。

b.

选择适当的 **语言和键盘** 选项。

c.

当提示输入安装所使用的设备类型时，选择 **Basic Storage Devices**。

d.

为您的设备选择一个主机名。默认主机名为 `localhost.localdomain`。

- e. **设置 root 密码。**
 - f. **根据磁盘上的空间，选择安装类型。**
 - g. **选择 *Basic Server install*，其中包括 SSH 服务器。**
3. **重新引导实例，并以 root 用户身份登录。**
 4. **更新 `/etc/sysconfig/network-scripts/ifcfg-eth0` 文件，使其只包含以下值：**

```
TYPE=Ethernet
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
NM_CONTROLLED=no
```

5. **重启服务网络：**

```
# service network restart
```

6. **继续 [配置主机以进行注册](#)。**

C.4. 配置主机以进行注册

Red Hat Enterprise Linux 虚拟机默认在客户门户网站订阅管理中注册。您必须更新每个虚拟机配置，以便它们从正确的卫星服务器或胶囊服务器接收更新。

前提条件

- **主机必须使用以下 Red Hat Enterprise Linux 版本：**
 - **6.4 或更高版本**

- **7.0 或更高版本**
- **支持 Red Hat Enterprise Linux 的所有架构(i386、x86_64、s390x、ppc_64)。**
- **确保已启用时间同步工具，并在卫星服务器、任何胶囊服务器和主机上运行。**
- **For Red Hat Enterprise Linux 6:**

```
# chkconfig ntpd on; service ntpd start
```
- **For Red Hat Enterprise Linux 7:**

```
# systemctl enable chronyd; systemctl start chronyd
```
- **确保后台程序 rhsmcertd 已启用并在主机上运行。**
- **For Red Hat Enterprise Linux 6:**

```
# chkconfig rhsmcertd on; service rhsmcertd start
```
- **For Red Hat Enterprise Linux 7:**

```
# systemctl start rhsmcertd
```

配置主机以进行注册：

1. **记录卫星服务器或胶囊服务器的完全限定域名(FQDN)，如 server.example.com。**
2. **在主机上，以 root 用户身份连接到主机上的终端**
3. **从要注册到主机的卫星服务器或胶囊服务器安装使用者 RPM。使用者 RPM 更新主机的内容源位置，并允许主机从 Red Hat Satellite 中指定的内容源下载内容。**

```
# rpm -Uvh http://server.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

C.5. 注册主机

前提条件

- 确保主机存在与适当的内容视图和环境关联的激活密钥。如需更多信息，请参阅 [内容管理指南](#) 中的 [管理激活码](#)。默认情况下，激活密钥启用了 `auto-attach` 功能。该功能通常用于用作虚拟机监控程序的主机。
- 确保 `subscription-manager` 工具的版本为 1.10 或更高版本。软件包位于标准 Red Hat Enterprise Linux 软件仓库中。
 1. 在 Red Hat Enterprise Linux Workstation 上，以 root 用户身份连接到终端。
 2. 使用 Red Hat Subscription Manager 注册主机：

```
# subscription-manager register --org="My_Organization" --activationkey="MyKey"
```

注意

您可以使用 `--environment` 选项覆盖激活密钥定义的内容视图和生命周期环境。例如，在 "Development" 生命周期环境中将主机注册到内容视图 "MyView"：

```
# subscription-manager register --org="My_Organization" \
--environment=Development/MyView \
--activationkey="MyKey"
```

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，且必须指向 6.3 存储库。

1. 在 Red Hat Satellite 中，选择 Hosts > Content Hosts。
2. 选择需要更改的主机的名称。
3. 在内容主机内容 部分中，单击 Release Version 右侧的编辑图标。
4. 从 Release Version 下拉菜单中选择 "6.3"。
5. 点 Save。

C.6. 安装 KATELLO AGENT

使用以下步骤在注册到 Satellite 6 的主机上安装 Katello 代理。katello-agent 软件包依赖于提供 gofer d 服务的 gofer 软件包。此服务必须启用，以便红帽卫星服务器或胶囊服务器可以提供适用于内容主机的勘误表信息。

注意

Katello 代理已弃用，并将在以后的 Satellite 版本中删除。迁移您的工作负载以使用远程执行功能来远程更新客户端。有关更多信息，请参阅管理主机指南中的 [安装 Katello 代理](#)。

前提条件

必须启用 Satellite Tools 6.9 存储库，并同步到卫星服务器，并在主机提供所需软件包时可用。有关启用 Satellite 工具 6.9 的更多信息，请参阅管理主机中的 [安装 Katello 代理](#)。

要安装 Katello Agent

1. 使用以下命令安装 katello-agent RPM 软件包：

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

2.

确保 goferd 正在运行：

```
# systemctl start goferd
```

C.7. 安装 PUPPET 代理

使用本节来在主机上安装和配置 Puppet 代理。当您正确安装和配置 Puppet 代理时，您可以导航到 **Hosts > All hosts**，以列出对 Red Hat Satellite Server 可见的所有主机。

1.

使用以下命令安装 Puppet 代理 RPM 软件包：

```
# yum install puppet
```

2.

将 puppet 代理配置为在引导时启动：

On Red Hat Enterprise Linux 6:

```
# chkconfig puppet on
```

在 Red Hat Enterprise Linux 7 上：

```
# systemctl enable puppet
```

C.8. 完成 RED HAT ENTERPRISE LINUX 7 镜像

1.

更新系统：

```
# {package-update}
```

2.

安装 cloud-init 软件包：

```
# yum install cloud-utils-growpart cloud-init
```

3. 打开 `/etc/cloud/cloud.cfg` 配置文件：

```
# vi /etc/cloud/cloud.cfg
```

4. 在标题 `cloud_init_modules` 下，添加：

```
- resolv-conf
```

`resolv-conf` 选项会在实例首次引导时自动配置 `resolv.conf`。此文件包含与实例相关的信息，如名称服务器、`domain` 和其他选项。

5. 打开 `/etc/sysconfig/network` 文件：

```
# vi /etc/sysconfig/network
```

6. 添加以下行以避免访问 EC2 元数据服务时出现问题：

```
NOZEROCONF=yes
```

7. 取消注册虚拟机，以便生成的镜像不包含基于它所克隆的每个实例的相同的订阅详情：

```
# subscription-manager repos --disable=*  
# subscription-manager unregister
```

8. 关闭实例：

```
# poweroff
```

9. 在 Red Hat Enterprise Linux Workstation 中，以 `root` 用户身份连接到终端，并进入 `/var/lib/libvirt/images/` 目录：

```
[root@host]# cd /var/lib/libvirt/images/
```

10. 使用 `virt-sysprep` 命令重置并清理镜像，以便它能够用于创建没有问题的实例：

```
[root@host]# virt-sysprep -d rhel7
```

11.

使用 **virt-sparsify** 命令减少镜像大小。这个命令将磁盘镜像中的任何可用空间重新转换为主机的可用空间：

```
[root@host]# virt-sparsify --compress rhel7.qcow2 rhel7-cloud.qcow2
```

这会在您输入命令的位置创建一个新的 **rhel7-cloud.qcow2** 文件。

C.9. 完成 RED HAT ENTERPRISE LINUX 6 镜像

1.

更新系统：

```
# {package-update}
```

2.

安装 **cloud-init** 软件包：

```
# yum install cloud-utils-growpart cloud-init
```

3.

编辑 **/etc/cloud/cloud.cfg** 配置文件和 **cloud_init_modules add:**

```
- resolv-conf
```

resolv-conf 选项会在实例首次引导时自动配置 **resolv.conf** 配置文件。此文件包含与实例相关的信息，如名称服务器、域及其他选项。

4.

要防止网络问题，请按如下所示创建 **/etc/udev/rules.d/75-persistent-net-generator.rules** 文件：

```
# echo "#" > /etc/udev/rules.d/75-persistent-net-generator.rules
```

这可防止 **/etc/udev/rules.d/70-persistent-net.rules** 文件被创建。如果创建了 **/etc/udev/rules.d/70-persistent-net.rules**，在从快照引导时，网络可能无法正常工作（网络接口创建为 **"eth1"** 而不是 **"eth0"**，且没有分配 IP 地址）。

5. 在 `/etc/sysconfig/network` 中添加以下行以避免访问 EC2 元数据服务时出现问题：

```
NOZEROCONF=yes
```

6. 取消注册虚拟机，以便生成的镜像不包含基于它所克隆的每个实例的相同的订阅详情：

```
# subscription-manager repos --disable=*
# subscription-manager unregister
# yum clean all
```

7. 关闭实例：

```
# poweroff
```

8. 在您的 Red Hat Enterprise Linux Workstation 中，以 root 身份登录并使用 `virt-sysprep` 命令重置并清理镜像，以便它可用于在不出现问题的情况下创建实例：

```
[root@host]# virt-sysprep -d rhel6
```

9. 使用 `virt-sparsify` 命令减少镜像大小。这个命令将磁盘镜像中的任何可用空间重新转换为主机的可用空间：

```
[root@host]# virt-sparsify --compress rhel6.qcow2 rhel6-cloud.qcow2
```

这会在您输入命令的位置创建一个新的 `rhel6-cloud.qcow2` 文件。



注意

您必须根据镜像手动调整实例分区，根据该类别中的磁盘空间应用到实例。

C.10. 后续步骤

- 对您要通过 Satellite 置备的每个镜像重复上述步骤。
- 将镜像移动到要存储以供将来使用的位置。

