



U Y U N I

Uyuni '2024.10'

客户端配置指南

2024年10月17日



# 目录

客户端配置指南概述	1
1. 支持的客户端和功能	2
1.1. 支持的客户端系统	2
1.2. 支持的工具软件包	3
1.3. 支持的 SLES 和 openSUSE 功能	3
1.4. 支持的 SLE Micro 功能	5
1.5. 支持的 SL Micro 客户端功能	7
1.6. openSUSE Leap Micro 客户端功能	9
1.7. 支持的 Alibaba Cloud Linux 功能	11
1.8. 支持的 AlmaLinux 功能	13
1.9. 支持的 Amazon Linux 功能	15
1.10. 支持的 CentOS 功能	17
1.11. 支持的 Debian 功能	19
1.12. 支持的 openEuler 功能	20
1.13. 支持的 Oracle 功能	22
1.14. 支持的 Raspberry Pi OS 功能	24
1.15. 支持的 Red Hat Enterprise Linux 功能	26
1.16. 支持的 Rocky Linux 功能	28
1.17. 支持的 Ubuntu 功能	30
2. 配置基本知识	33
2.1. 软件通道	33
2.1.1. 通过 SUSE Package Hub 提供的软件包	33
2.1.2. 通过 AppStream 提供的软件包	34
2.1.3. 通过 EPEL 提供的软件包	34
2.1.4. SUSE Linux Enterprise 客户端的 Unified Installer 更新通道	34
2.1.5. 软件储存库	34
2.1.6. 软件产品	35
2.1.7. AppStream 模块	36
2.2. 引导储存库	36
2.2.1. 准备创建引导储存库	36
2.2.2. 自动模式的选项	37
2.2.3. 手动生成引导储存库	37
2.2.4. 引导和自定义通道	39
2.3. 激活密钥	39
2.3.1. 重新激活密钥	41
2.3.2. 激活密钥最佳实践	41
2.4. GPG 密钥	43
2.4.1. 在客户端上信任 GPG 密钥	43
3. 客户端联系方法	45
3.1. 默认联系方法 (Salt)	45
3.1.1. 初始配置细节	46
3.2. SSH 推送联系方法	46
3.2.1. 可用参数	48
3.2.2. 操作的执行	49
3.2.3. 未来的能力	49
3.3. SSH 推送（使用隧道）联系方法	49
3.4. Salt 捆绑包	51
3.4.1. 什么是 Salt 捆绑包？	51
3.4.2. 使用 Salt 捆绑包将客户端注册为受控端	51
3.4.3. 使用 Salt 捆绑包执行 SSH 推送	53
3.4.4. 使用 pip 通过 Python 软件包扩展 Salt 捆绑包	53
4. 客户端注册	54

4.1. 注册方法	54
4.1.1. 使用 Web UI 注册客户端	54
4.1.2. 使用引导脚本注册客户端	56
4.1.3. 在命令行上注册客户端	59
4.2. SUSE 客户端注册	61
4.2.1. 注册 SUSE Linux Enterprise 客户端	62
4.2.2. 注册 SLE Micro 客户端	65
4.2.3. 注册 SL Micro 客户端	69
4.3. openSUSE 客户端注册	72
4.3.1. 注册 openSUSE Leap 客户端	72
4.3.2. 注册 openSUSE Leap Micro 客户端	74
4.4. Alibaba Cloud Linux 客户端注册	76
4.4.1. 注册 Alibaba Cloud Linux 客户端	76
4.5. AlmaLinux 客户端注册	78
4.5.1. 注册 AlmaLinux 客户端	78
4.6. Amazon Linux 客户端注册	81
4.6.1. 注册 Amazon Linux 客户端	81
4.7. CentOS 客户端注册	83
4.7.1. 注册 CentOS 客户端	83
4.8. Debian 客户端注册	87
4.8.1. 注册 Debian 客户端	88
4.9. OpenEuler 客户端注册	90
4.9.1. 注册 openEuler 客户端	91
4.10. Oracle 客户端注册	94
4.10.1. 注册 Oracle Linux 客户端	94
4.11. Raspberry Pi OS 客户端注册	97
4.11.1. 注册 Raspberry Pi OS 客户端	97
4.12. Red Hat 客户端注册	100
4.12.1. 使用 CDN 注册 Red Hat Enterprise Linux 客户端	101
4.12.2. 使用 RHUI 注册 Red Hat Enterprise Linux 客户端	108
4.13. Rocky Linux 客户端注册	113
4.13.1. 注册 Rocky Linux 客户端	113
4.14. Ubuntu 客户端注册	116
4.14.1. Registering Ubuntu 20.04, 22.04 and 24.04 Clients	116
4.15. 将客户端注册到代理	120
4.15.1. 在代理之间移动客户端	120
4.15.2. 将客户端从代理移到服务器	121
4.15.3. 使用 Web UI 将客户端注册到代理	121
4.15.4. 在命令行上注册客户端	122
4.16. 在公有云上注册客户端	123
4.16.1. 添加产品并同步储存库	123
4.16.2. 准备按需映像	123
4.16.3. 注册客户端	124
4.16.4. 激活密钥	124
4.16.5. 自动注册 Terraform 创建的客户端	124
5. 客户端升级	126
5.1. 主要版本升级	126
5.1.1. 准备迁移	126
5.1.2. 迁移	128
5.2. 使用内容生命周期管理器升级	129
5.2.1. 准备升级	129
5.2.2. 升级	130
5.3. 产品迁移	131
5.3.1. 单系统迁移	131
5.3.2. 产品大量迁移	132
5.4. 升级 Uyuni 客户端	135

5.4.1. 准备升级 .....	135
5.4.2. 升级 .....	136
6. 客户端删除 .....	137
6.1. 使用 Web UI 删除客户端 .....	137
6.2. 在命令行上删除客户端（通过 API 调用） .....	137
6.3. 从命令行删除客户端 .....	138
7. 客户端操作 .....	139
7.1. 软件包管理 .....	139
7.1.1. 使用配置文件比较软件包 .....	139
7.1.2. Orphaned Packages .....	139
7.2. 补丁管理 .....	140
7.2.1. 创建补丁 .....	140
7.2.2. 将补丁应用到客户端 .....	142
7.3. 系统锁定 .....	143
7.3.1. 客户端上的系统锁 .....	143
7.3.2. 软件包锁 .....	143
7.4. 配置管理 .....	144
7.4.1. 创建配置通道 .....	145
7.4.2. 添加配置文件、目录或符号链接 .....	146
7.4.3. 为客户端订阅配置通道 .....	146
7.4.4. 比较配置文件 .....	147
7.4.5. 在客户端上建立 Jinja 模板 .....	147
7.5. 电源管理 .....	147
7.5.1. 电源管理和 Cobbler .....	148
7.6. 自定义系统信息 .....	148
7.7. 系统集管理器 .....	149
7.7.1. 在 SSM 中更改基础通道 .....	150
7.8. 系统组 .....	151
7.8.1. 创建组 .....	151
7.8.2. 将客户端添加到组 .....	151
7.8.3. 使用组 .....	152
7.9. 系统类型 .....	152
8. 操作系统安装 .....	153
8.1. 重新安装已注册系统 .....	153
8.2. 通过 CD-ROM 或 USB 密钥安装 .....	154
8.2.1. 使用 Cobbler 构建 ISO 映像 .....	154
8.2.2. 使用 KIWI 构建 SUSE ISO 映像 .....	155
8.2.3. 使用 Cobbler 构建 Red Hat ISO 映像 .....	155
8.3. 可自动安装的发行套件 .....	155
8.3.1. 基于 ISO 映像的发行套件 .....	156
8.3.2. 基于 RPM 软件包的发行套件 .....	156
8.3.3. 声明可自动安装的发行套件 .....	157
8.3.4. Handling Kernel Options for Distributions and Profiles .....	157
8.4. 自动安装配置文件 .....	159
8.4.1. 声明配置文件 .....	160
8.4.2. AutoYaST 配置文件 .....	161
8.4.3. Kickstart 配置文件 .....	161
8.4.4. 模板语法 .....	162
8.5. 无人照管的置备 .....	164
8.6. 使用您自己的 GPG 密钥 .....	164
9. 虚拟化 .....	166
9.1. 管理虚拟化主机 .....	166
9.2. 创建虚拟 Guest .....	166
9.3. 使用 Xen 和 KVM 虚拟化 .....	167
9.3.1. 主机设置 .....	167

9.3.2. 自动安装 VM Guest .....	168
9.3.3. 管理 VM Guest .....	171
10. 虚拟主机管理器 .....	173
10.1. 虚拟主机管理器和 Amazon Web Services .....	173
10.1.1. 创建 Amazon EC2 VHM .....	173
10.1.2. 虚拟主机管理器的 AWS 权限 .....	174
10.2. 虚拟主机管理器和 Azure .....	175
10.2.1. 先决条件 .....	175
10.2.2. 创建 Azure VHM .....	175
10.2.3. 指派权限 .....	175
10.2.4. Azure UUID .....	176
10.3. 虚拟主机管理器和 Google Compute Engine .....	176
10.3.1. 先决条件 .....	176
10.3.2. 创建 GCE VHM .....	177
10.3.3. 指派权限 .....	177
10.3.4. GCE UUID .....	178
10.4. 使用 Nutanix 虚拟化 .....	178
10.4.1. VHM 设置 .....	178
10.5. Virtualization with VMware .....	179
10.5.1. VHM 设置 .....	179
10.5.2. Troubleshooting SSL Errors on VMware .....	180
10.6. 使用其他第三方提供者虚拟化 .....	181
11. GNU Free Documentation License .....	183

# 客户端配置指南概述

更新日期：2024-10-17

安装 Uyuni 后首先要做的就是注册客户端，您花在 Uyuni 上的大部分时间都是用来维护这些客户端。

Uyuni 与 SUSE Linux Enterprise 或其他 Linux 操作系统兼容，并提供多种硬件选项。

有关支持的客户端和功能的完整列表，请参见 [Client-configuration > Supported-features](#)。

本指南介绍了如何注册以及配置不同的客户端，包括手动和自动两种方式。

# Chapter 1. 支持的客户端和功能

Uyuni 与很多客户端技术兼容。有了多种硬件选项，您可以安装运行 SUSE Linux Enterprise 或其他 Linux 操作系统的 Salt 客户端。

本章提供了支持的客户端系统摘要。有关每个客户端上可用功能的详细列表，请参见后面的页面。

## 1.1. 支持的客户端系统

下表列出了 Salt 客户端支持的操作系统。

此表中图标的含意如下：

- ✓ 运行此操作系统的客户端受 SUSE 支持
- ✗ 运行此操作系统的客户端不受 SUSE 支持
- ? 客户端正在考虑之中，日后可能受支持，也可能不受支持。



客户端操作系统的版本和 SP 级别必须享受标准支持（常规或 LTSS）才受 Uyuni 的支持。有关受支持产品版本的细节，请参见 <https://www.suse.com/lifecycle>。



客户端上运行的操作系统由提供操作系统的组织支持。

表格 1. 支持的客户端系统和体系结构

Operating System	x86-64	ppc64le	IBM Z	aarch64	arm64 / armhf
SUSE Linux Enterprise 15, 12	✓	✓	✓	✓	✗
SUSE Linux Enterprise Server for SAP 15, 12	✓	✓	✗	✗	✗
SLE Micro	✓	✗	✗	✓	✗
SL Micro	✓	✗	✗	✓	✗
openSUSE Leap Micro	✓	✗	✗	✓	✗
openSUSE Leap 15	✓	✗	✗	✓	✗
Alibaba Cloud Linux 2	✓	✗	✗	✓	✗
AlmaLinux 9, 8	✓	✗	✗	✓	✗
Amazon Linux 2003, 2	✓	✗	✗	✓	✗
CentOS 7	✓	✓	✗	✓	✗
Debian 12, 11	✓	✗	✗	✗	✗
openEuler 22.03	✓	✗	✗	✗	✗
Oracle Linux 9, 8, 7	✓	✗	✗	✓	✗
Raspberry Pi OS 12	✗	✗	✗	✗	✓

Operating System	x86-64	ppc64le	IBM Z	aarch64	arm64 / armhf
Red Hat Enterprise Linux 9, 8, 7	✓	✗	✗	✗	✗
Rocky Linux 9, 8	✓	✗	✗	✓	✗
Ubuntu 24.04, 22.04, 20.04 (*)	✓	✗	✗	✗	✗



(\*) Debian 和 Ubuntu 将 x86-64 体系结构列为 amd64。

发行套件到达生命周期结束日期时，将进入 3 个月的宽限期，届时支持将视为处于弃用状态。宽限期结束之后，产品即视为不受支持。此后我们将只能尽最大努力提供支持。

有关生命周期结束日期的详细信息，请参见 <https://endoflife.software/operating-systems>。

## 1.2. 支持的工具软件包

spacewalk-utils 和 spacewalk-utils-extras 软件包可提供额外的服务和功能。

表格 2. Spacewalk 实用程序

工具名称	说明	受支持?
spacewalk-common-channels	添加 SUSE Customer Center 不提供的通道	✓
spacewalk-hostname-rename	更改 Uyuni 服务器的主机名	✓
spacewalk-clone-by-date	按特定日期克隆通道	✓
spacewalk-sync-setup	设置 ISS 主组织和从属组织映射	✓
spacewalk-manage-channel-lifecycle	管理通道生命周期	✓

## 1.3. 支持的 SLES 和 openSUSE 功能

下表列出了 SUSE 和 openSUSE 客户端上各种功能的可用性。该表涵盖了 SUSE Linux Enterprise 操作系统的所有变体，包括 SLES、SLED、SUSE Linux Enterprise Server for SAP 和 SUSE Linux Enterprise Server for HPC。



客户端上运行的操作系统由提供操作系统的组织支持。SUSE Linux Enterprise 由 SUSE 支持。openSUSE 由 SUSE 社区支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 3. SUSE 和 openSUSE 操作系统上支持的功能

功能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
客户端	✓	✓	✓
系统软件包	SUSE	SUSE	openSUSE 社区
注册	✓	✓	✓
安装软件包	✓	✓	✓
应用补丁	✓	✓	✓
远程命令	✓	✓	✓
系统软件包状态	✓	✓	✓
系统自定义状态	✓	✓	✓
组自定义状态	✓	✓	✓
组织自定义状态	✓	✓	✓
系统集管理器 (SSM)	✓	✓	✓
产品迁移	✓	✓	✓
基本虚拟 Guest 管理 *	✓	✓	✓
高级虚拟 Guest 管理 *	✓	✓	✓
虚拟 Guest 安装 (AutoYaST)，作为主机操作系统	✗	✗	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	✓	✓	✓
虚拟 Guest 管理	✓	✓	✓
系统部署 (PXE/AutoYaST)	✓	✓	✓
系统重新部署 (AutoYaST)	✓	✓	✓
联系方法	✗	✗	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓	✓	✓
操作链	✓	✓	✓
暂存（预先下载软件包）	✓	✓	✓
重复软件包报告	✓	✓	✓
CVE 审计	✓	✓	✓
SCAP 审计	✓	✓	✓
软件包校验	✗	✗	✗
软件包锁定	✓	✓	✓
系统锁定	✗	✗	✗

功能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
维护时段	✓	✓	✓
系统快照	✗	✗	✗
配置文件管理	✓	✓	✓
软件包配置文件	✓ 支持配置文件，不支持同步	✓ 支持配置文件，不支持同步	✓ 支持配置文件，不支持同步
电源管理	✓	✓	✓
监视服务器	✓	✓	✓
受监视客户端	✓	✓	✓
Docker buildhost	✓	✓	?
构建含操作系统的 Docker 映像	✓	✓	✓
Kiwi buildhost	✓	?	?
构建含操作系统的 Kiwi 映像	✓	?	✗
重复性操作	✓	✓	✓
AppStream	不适用	不适用	不适用
Yomi	✗	✓	✓

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.4. 支持的 SLE Micro 功能



客户端上运行的操作系统由提供操作系统的组织支持。SLE Micro 由 SUSE 支持。



目前，SLE Micro 仅支持作为常规的受控端（----联系方法）。我们正力求让其也能作为 Salt SSH 客户端（salt-ssh 联系方法）进行管理。

此表中图标的含意如下：

- ✓ 该功能可用

- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 4. SLE Micro 操作系统上支持的功能

功能	SLE Micro
客户端	✓
操作系统软件包	✓
注册	✓
安装软件包	✓
应用补丁（需要 CVE ID）	✓
远程命令	✓
系统软件包状态	✓
系统自定义状态	✓
组自定义状态	✓
组织自定义状态	✓
系统集管理器 (SSM)	✓
产品迁移	✓
基本虚拟 Guest 管理 *	✓
高级虚拟 Guest 管理 *	✓
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✓
虚拟 Guest 安装（映像模板），作为主机操作系统	✓
系统部署 (PXE/Kickstart)	✓
系统重新部署 (Kickstart)	✓
联系方法	✓ ZeroMQ
使用 Uyuni Proxy	✓
操作链	✓
暂存（预先下载软件包）	?
重复软件包报告	✓
CVE 审计（需要 CVE ID）	✓
SCAP 审计	?
软件包校验	?
软件包锁定	✓
系统锁定	?
维护时段	?
系统快照	✗

功能	SLE Micro
配置文件管理	✓
快照和配置文件	✓ 支持配置文件，不支持同步
电源管理	✓
监控服务器	✗
受监视客户端 <b>**</b>	✓
Docker buildhost	✗
构建含操作系统的 Docker 映像	✗
Kiwi buildhost	✗
构建含操作系统的 Kiwi 映像	✓
重复性操作	✓
AppStreams	不适用
Yomi	?

**\*\*** 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

**\*\*** 在 SLE Micro 上，只能使用 Node Exporter 和 Blackbox Exporter。

## 1.5. 支持的 SL Micro 客户端功能



客户端上运行的操作系统由提供操作系统的组织支持。SL Micro 由 SUSE 支持。



目前 SL Micro 仅支持作为使用 ---- 联系方法的 Salt 客户端。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 5. SL Micro 操作系统上支持的功能

功能	SL Micro
客户端	Salt
操作系统软件包	Salt
注册	Salt
安装软件包	Salt
应用补丁（需要 CVE ID）	Salt
远程命令	Salt
系统软件包状态	Salt
系统自定义状态	Salt
组自定义状态	Salt
组织自定义状态	Salt
系统集管理器 (SSM)	Salt
产品迁移	Salt
基本虚拟 Guest 管理 *	?
高级虚拟 Guest 管理 *	?
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	?
系统部署 (PXE/Kickstart)	?
系统重新部署 (Kickstart)	✗
联系方法	Salt: ZeroMQ
使用 Uyuni Proxy	Salt
操作链	Salt
暂存（预先下载软件包）	?
重复软件包报告	Salt
CVE 审计（需要 CVE ID）	Salt
SCAP 审计	?
软件包校验	?
软件包锁定	Salt
系统锁定	?
维护时段	?
系统快照	✗
配置文件管理	Salt
快照和配置文件	Salt: 支持配置文件，不支持同步
电源管理	Salt

功能	SL Micro
监控服务器	✗
受监视客户端 <b>**</b>	Salt
Docker buildhost	✗
构建含操作系统的 Docker 映像	✗
Kiwi buildhost	✗
构建含操作系统的 Kiwi 映像	Salt
重复性操作	Salt
AppStream	不适用
Yomi	?

**\*** 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

**\*\*** 在 SL Micro 上，只能使用 Node Exporter 和 Blackbox Exporter。

## 1.6. openSUSE Leap Micro 客户端功能



openSUSE Leap Micro 由 SUSE 社区提供支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 6. openSUSE Leap Micro 操作系统上支持的功能

功能	openSUSE Leap Micro
客户端	✓
操作系统软件包	✓
注册	✓
安装软件包	✓
应用补丁（需要 CVE ID）	✓

功能	openSUSE Leap Micro
远程命令	✓
系统软件包状态	✓
系统自定义状态	✓
组自定义状态	✓
组织自定义状态	✓
系统集管理器 (SSM)	✓
产品迁移	✓
基本虚拟 Guest 管理 *	✓
高级虚拟 Guest 管理 *	✓
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✓
虚拟 Guest 安装（映像模板），作为主机操作系统	✓
系统部署 (PXE/Kickstart)	✓
系统重新部署 (Kickstart)	✓
联系方法	✓ ZeroMQ
使用 Uyuni 代理	✓
操作链	✓
暂存（预先下载软件包）	?
重复软件包报告	✓
CVE 审计（需要 CVE ID）	✓
SCAP 审计	?
软件包校验	?
软件包锁定	✓
系统锁定	?
维护时段	?
系统快照	✗
配置文件管理	✓
快照和配置文件	✓ 支持配置文件，不支持同步
电源管理	✓
监视服务器	✗
受监视客户端	✓
Docker buildhost	✗
构建含操作系统的 Docker 映像	✗
Kiwi buildhost	✗

功能	openSUSE Leap Micro
构建含操作系统的 Kiwi 映像	✓
重复性操作	✓
AppStream	不适用
Yomi	?

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.7. 支持的 Alibaba Cloud Linux 功能

下表列出了 Alibaba Cloud Linux 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。Alibaba Cloud Linux 由 Alibaba Cloud 支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 7. Alibaba Cloud Linux 操作系统上支持的功能

功能	Alibaba Cloud Linux 2
客户端	✓
操作系统软件包	✓
注册	✓
安装软件包	✓
应用补丁（需要 CVE ID）	✓
远程命令	✓
系统软件包状态	✓
系统自定义状态	✓
组自定义状态	✓

功能	Alibaba Cloud Linux 2
组织自定义状态	✓
系统集管理器 (SSM)	✓
产品迁移	不适用
基本虚拟 Guest 管理 *	?
高级虚拟 Guest 管理 *	?
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	?
系统部署 (PXE/Kickstart)	?
系统重新部署 (Kickstart)	?
联系方法	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓
操作链	✓
暂存（预先下载软件包）	✓
重复软件包报告	✓
CVE 审计（需要 CVE ID）	✓
SCAP 审计	✓
软件包校验	✗
软件包锁定	✗
系统锁定	✗
维护时段	✓
系统快照	✗
配置文件管理	✓
快照和配置文件	✓ 支持配置文件，不支持同步
电源管理	?
监视服务器	✗
受监视客户端	✓
Docker buildhost	✓
构建含操作系统的 Docker 映像	✓
Kiwi buildhost	✓
构建含操作系统的 Kiwi 映像	✓
重复性操作	✓
AppStream	不适用
Yomi	不适用

## \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.8. 支持的 AlmaLinux 功能

下表列出了 AlmaLinux 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。AlmaLinux 由 AlmaLinux 社区支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✘ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 8. AlmaLinux 操作系统上支持的功能

Feature	AlmaLinux 9	AlmaLinux 8
Client	✓ (plain AlmaLinux)	✓ (plain AlmaLinux)
System packages	AlmaLinux Community	AlmaLinux Community
Registration	✓	✓
Install packages	✓	✓
Apply patches	✓	✓
Remote commands	✓	✓
System package states	✓	✓
System custom states	✓	✓
Group custom states	✓	✓
Organization custom states	✓	✓
System set manager (SSM)	✓	✓
Product migration *	✓	✓
Basic Virtual Guest Management **	✓	✓

Feature	AlmaLinux 9	AlmaLinux 8
Advanced Virtual Guest Management **	✓	✓
Virtual Guest Installation (Kickstart), as Host OS	✗	✗
Virtual Guest Installation (image template), as Host OS	✓	✓
System deployment (PXE/Kickstart)	✓	✓
System redeployment (Kickstart)	✓	✓
Contact methods	✓ ZeroMQ, Salt-SSH	✓ ZeroMQ, Salt-SSH
Works with Uyuni Proxy	✓	✓
Action chains	✓	✓
Staging (pre-download of packages)	✓	✓
Duplicate package reporting	✓	✓
CVE auditing	✓	✓
SCAP auditing	✓	✓
Package verification	✗	✗
Package locking	✗	✗
System locking	✗	✗
Maintenance Windows	✓	✓
System snapshot	✗	✗
Configuration file management	✓	✓
Snapshots and profiles	✓ Profiles supported, Sync not supported	✓ Profiles supported, Sync not supported
Power management	✓	✓
Monitoring server	✗	✗
Monitored clients	✓	✓
Docker buildhost	✗	✗
Build Docker image with OS	✗	✗
Kiwi buildhost	✗	✗
Build Kiwi image with OS	✗	✗
Recurring Actions	✓	✓
AppStreams	✓	✓
Yomi	N/A	N/A

\* to matching SUSE Liberty Linux

## \*\* Virtual Guest Management:

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.9. 支持的 Amazon Linux 功能

下表列出了 Amazon Linux 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。Amazon Linux 由 Amazon 支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 9. Amazon Linux 操作系统上支持的功能

功能	Amazon Linux 2	Amazon Linux 2023
客户端	✓	✓
操作系统软件包	✓	✓
注册	✓	✓
安装软件包	✓	✓
应用补丁（需要 CVE ID）	✓	✓
远程命令	✓	✓
系统软件包状态	✓	✓
系统自定义状态	✓	✓
组自定义状态	✓	✓
组织自定义状态	✓	✓
系统集管理器 (SSM)	✓	✓
产品迁移	不适用	不适用
基本虚拟 Guest 管理 *	?	?

功能	Amazon Linux 2	Amazon Linux 2023
高级虚拟 Guest 管理 *	?	?
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✗	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	?	?
系统部署 (PXE/Kickstart)	?	?
系统重新部署 (Kickstart)	?	?
联系方法	✓ ZeroMQ、Salt-SSH	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓	✓
操作链	✓	✓
暂存（预先下载软件包）	✓	✓
重复软件包报告	✓	✓
CVE 审计（需要 CVE ID）	✓	✓
SCAP 审计	✓	✓
软件包校验	✗	✗
软件包锁定	✗	✗
系统锁定	✗	✗
维护时段	✓	✓
系统快照	✗	✗
配置文件管理	✓	✓
快照和配置文件	✓ 支持配置文件，不支持同步	✓ 支持配置文件，不支持同步
电源管理	?	?
监视服务器	✗	✗
受监视客户端	✓	✓
Docker buildhost	✓	✓
构建含操作系统的 Docker 映像	✓	✓
Kiwi buildhost	✓	✓
构建含操作系统的 Kiwi 映像	✓	✓
重复性操作	✓	✓
AppStream	不适用	不适用
Yomi	不适用	不适用

\* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.10. 支持的 CentOS 功能

下表列出了 CentOS 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。CentOS 由 CentOS 社区支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 10. CentOS 操作系统上支持的功能

Feature	CentOS 7
Client	✓ (plain CentOS)
System packages	CentOS Community
Registration	✓
Install packages	✓
Apply patches (requires CVE ID)	✓ (third-party service required for errata)
Remote commands	✓
System package states	✓
System custom states	✓
Group custom states	✓
Organization custom states	✓
System set manager (SSM)	✓
Product migration *	✓
Basic Virtual Guest Management **	✓
Advanced Virtual Guest Management **	✓
Virtual Guest Installation (Kickstart), as Host OS	✗
Virtual Guest Installation (image template), as Host OS	✓
System deployment (PXE/Kickstart)	✓

Feature	CentOS 7
System redeployment (Kickstart)	✓
Contact methods	✓ ZeroMQ, Salt-SSH
Works with Uyuni Proxy	✓
Action chains	✓
Staging (pre-download of packages)	✓
Duplicate package reporting	✓
CVE auditing (requires CVE ID)	✓
SCAP auditing	✓
Package verification	✗
Package locking	✓
System locking	✗
Maintenance Windows	✓
System snapshot	✗
Configuration file management	✓
Snapshots and profiles	✓ Profiles supported, Sync not supported
Power management	✓
Monitoring server	✗
Monitored clients	✓
Docker buildhost	✗
Build Docker image with OS	✗
Kiwi buildhost	✗
Build Kiwi image with OS	✗
Recurring Actions	✓
AppStreams	N/A
Yomi	N/A

\* to matching SUSE Liberty Linux

## \*\* Virtual Guest Management:

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.11. 支持的 Debian 功能

下表列出了 Debian 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。Debian 由 Debian 社区支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 11. Debian 操作系统上支持的功能

功能	Debian 11	Debian 12
客户端	✓	✓
系统软件包	Debian 社区	Debian 社区
注册	✓	✓
安装软件包	✓	✓
应用补丁	?	?
远程命令	✓	✓
系统软件包状态	✓	✓
系统自定义状态	✓	✓
组自定义状态	✓	✓
组织自定义状态	✓	✓
系统集管理器 (SSM)	✓	✓
产品迁移	不适用	不适用
基本虚拟 Guest 管理 *	✓	✓
高级虚拟 Guest 管理 *	✓	✓
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✓	✓
虚拟 Guest 安装（映像模板），作为主机操作系统	✓	✓
系统部署 (PXE/Kickstart)	✓	✓
系统重新部署 (Kickstart)	✗	✗
联系方法	✓ ZeroMQ、Salt-SSH	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓	✓

功能	Debian 11	Debian 12
操作链	✓	✓
暂存（预先下载软件包）	✓	✓
重复软件包报告	✓	✓
CVE 审计	?	?
SCAP 审计	?	?
软件包校验	✗	✗
软件包锁定	✓	✓
系统锁定	✗	✗
维护时段	✓	✓
系统快照	✗	✗
配置文件管理	✓	✓
软件包配置文件	✓ 支持配置文件，不支持同步	✓ 支持配置文件，不支持同步
电源管理	✓	✓
监视服务器	✗	✗
监视客户端	✓	✓
Docker buildhost	?	?
构建含操作系统的 Docker 映像	Salt	Salt
Kiwi buildhost	✗	✗
构建含操作系统的 Kiwi 映像	✗	✗
重复性操作	✓	✓
AppStream	不适用	不适用
Yomi	不适用	不适用

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.12. 支持的 openEuler 功能

下表列出了 openEuler 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织提供支持。openEuler 由 openEuler 社区提供支持。

此表中图标的含意如下：

- ✓ 该功能在 Salt 客户端和传统客户端上均可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 12. openEuler 操作系统上支持的功能

功能	openEuler 22.03
客户端	✓ (单纯的 openEuler)
系统软件包	openEuler 社区
注册	✓
安装软件包	✓
应用补丁	✓
远程命令	✓
系统软件包状态	✓
系统自定义状态	✓
组自定义状态	✓
组织自定义状态	✓
系统集管理器 (SSM)	✓
产品迁移	不适用
基本虚拟 Guest 管理 *	✓
高级虚拟 Guest 管理 *	✓
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	✓
系统部署 (PXE/Kickstart)	✓
系统重新部署 (Kickstart)	✓
联系方法	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓
操作链	✓
暂存（预先下载软件包）	✓
重复软件包报告	✓
CVE 审计	✓
SCAP 审计	✓

功能	openEuler 22.03
软件包校验	✗
软件包锁定	✗
系统锁定	✗
维护时段	✓
系统快照	✗
配置文件管理	✓
快照和配置文件	✓ 支持配置文件，不支持同步
电源管理	✓
监视	✓
Docker buildhost	✗
构建含操作系统的 Docker 映像	✗
Kiwi buildhost	✗
构建含操作系统的 Kiwi 映像	✗
重复性操作	✓
Yomi	不适用

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.13. 支持的 Oracle 功能

下表列出了 Oracle Linux 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。Oracle Linux 由 Oracle 支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 13. Oracle Linux 操作系统上支持的功能

Feature	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
Client	✓	✓	✓
Operating system packages	✓	✓	✓
Registration	✓	✓	✓
Install packages	✓	✓	✓
Apply patches (requires CVE ID)	✓	✓	✓
Remote commands	✓	✓	✓
System package states	✓	✓	✓
System custom states	✓	✓	✓
Group custom states	✓	✓	✓
Organization custom states	✓	✓	✓
System set manager (SSM)	✓	✓	✓
Product migration *	✓	✓	✓
Basic Virtual Guest Management **	✓	✓	✓
Advanced Virtual Guest Management ***	✓	✓	✓
Virtual Guest Installation (Kickstart), as Host OS	✗	✓	✓
Virtual Guest Installation (image template), as Host OS	✓	✓	✓
System deployment (PXE/Kickstart)	✓	✓	✓
System redeployment (Kickstart)	✓	✓	✓
Contact methods	✓ ZeroMQ, Salt-SSH	✓ ZeroMQ, Salt-SSH	✓ ZeroMQ, Salt-SSH
Works with Uyuni Proxy	✓	✓	✓
Action chains	✓	✓	✓
Staging (pre-download of packages)	✓	✓	✓
Duplicate package reporting	✓	✓	✓

Feature	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
CVE auditing (requires CVE ID)	✓	✓	✓
SCAP auditing	✓	✓	✓
Package verification	✗	✗	✗
Package locking	✓	✓	✓
System locking	✗	✗	✗
Maintenance Windows	✓	✓	✓
System snapshot	✗	✗	✗
Configuration file management	✓	✓	✓
Snapshots and profiles	✓ Profiles supported, Sync not supported	✓ Profiles supported, Sync not supported	✓ Profiles supported, Sync not supported
Power management	✓	✓	✓
Monitoring server	✗	✗	✗
Monitored clients	✓	✓	✓
Docker buildhost	✗	✗	✗
Build Docker image with OS	✗	✗	✗
Kiwi buildhost	✗	✗	✗
Build Kiwi image with OS	✗	✗	✗
Recurring Actions	✓	✓	✓
AppStreams	N/A	✓	✓
Yomi	N/A	N/A	N/A

\* to matching SUSE Liberty Linux

\*\* Virtual Guest Management:

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.14. 支持的 Raspberry Pi OS 功能

下表列出了 Raspberry Pi OS 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织提供支持。Raspberry Pi OS 由 Raspberry Pi OS 社区提供支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 14. Raspberry Pi OS 操作系统上支持的功能

功能	Raspberry Pi OS 12
客户端	✓
系统软件包	Raspberry Pi OS Community
注册	✓
安装软件包	✓
应用补丁	?
远程命令	✓
系统软件包状态	✓
系统自定义状态	✓
组自定义状态	✓
组织自定义状态	✓
系统集管理器 (SSM)	✓
产品迁移	不适用
基本虚拟 Guest 管理 *	✓
高级虚拟 Guest 管理 *	✓
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	✓
系统部署 (PXE/Kickstart)	✗
系统重新部署 (Kickstart)	✗
联系方法	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓
操作链	✓
暂存（预先下载软件包）	✓
重复软件包报告	✓
CVE 审计	?
SCAP 审计	?

功能	Raspberry Pi OS 12
软件包校验	✗
软件包锁定	✓
系统锁定	✗
维护时段	✓
系统快照	✗
配置文件管理	✓
软件包配置文件	✓ 支持配置文件，不支持同步
电源管理	✓
监视服务器	✗
监视客户端	✓
Docker buildhost	?
构建含操作系统的 Docker 映像	✓
Kiwi buildhost	✗
构建含操作系统的 Kiwi 映像	✗
重复性操作	✓
AppStream	不适用
Yomi	不适用

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.15. 支持的 Red Hat Enterprise Linux 功能

下表列出了本机 Red Hat Enterprise Linux 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。Red Hat Enterprise Linux 由 Red Hat 支持。

此表中图标的含意如下：

- ✓ 该功能可用

- ✗ 该功能不可用
- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 15. Red Hat Enterprise Linux 操作系统上支持的功能

功能	RHEL 7	RHEL 8	RHEL 9
客户端	✓	✓	✓
系统软件包	Red Hat	Red Hat	Red Hat
注册	✓	✓	✓
安装软件包	✓	✓	✓
应用补丁	✓	✓	✓
远程命令	✓	✓	✓
系统软件包状态	✓	✓	✓
系统自定义状态	✓	✓	✓
组自定义状态	✓	✓	✓
组织自定义状态	✓	✓	✓
系统集管理器 (SSM)	✓	✓	✓
产品迁移	不适用	不适用	不适用
基本虚拟 Guest 管理 *	✓	✓	✓
高级虚拟 Guest 管理 *	✓	✓	✓
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	✗	✗	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	✓	✓	✓
系统部署 (PXE/Kickstart)	✓	✓	✓
系统重新部署 (Kickstart)	✓	✓	✓
联系方法	✓ ZeroMQ、Salt-SSH	✓ ZeroMQ、Salt-SSH	✓ ZeroMQ、Salt-SSH
使用 Uyuni 代理	✓	✓	✓
操作链	✓	✓	✓
暂存（预先下载软件包）	✓	✓	✓
重复软件包报告	✓	✓	✓
CVE 审计	✓	✓	✓
SCAP 审计	✓	✓	✓
软件包校验	✗	✗	✗
软件包锁定	✓	✓	✓
系统锁定	✗	✗	✗

功能	RHEL 7	RHEL 8	RHEL 9
维护时段	✓	✓	✓
系统快照	✗	✗	✗
配置文件管理	✓	✓	✓
快照和配置文件	✓ 支持配置文件，不支持同步	✓ 支持配置文件，不支持同步	✓ 支持配置文件，不支持同步
电源管理	✓	✓	✓
监视服务器	✗	✗	✗
受监视客户端	✓	✓	✓
Docker buildhost	✓	✓	✓
构建含操作系统的 Docker 映像	?	?	?
Kiwi buildhost	✗	✗	✗
构建含操作系统的 Kiwi 映像	✗	✗	✗
重复性操作	✓	✓	✓
AppStream	不适用	✓	✓
Yomi	不适用	不适用	不适用

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.16. 支持的 Rocky Linux 功能

下表列出了 Rocky Linux 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织提供支持。Rocky Linux 由 Debian 社区提供支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用

- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 16. Rocky Linux 操作系统上支持的功能

Feature	Rocky Linux 8	Rocky Linux 9
Client	✓ (plain Rocky Linux)	✓ (plain Rocky Linux)
System packages	Rocky Linux Community	Rocky Linux Community
Registration	✓	✓
Install packages	✓	✓
Apply patches	✓	✓
Remote commands	✓	✓
System package states	✓	✓
System custom states	✓	✓
Group custom states	✓	✓
Organization custom states	✓	✓
System set manager (SSM)	✓	✓
Product migration *	✓	✓
Basic Virtual Guest Management **	✓	✓
Advanced Virtual Guest Management ***	✓	✓
Virtual Guest Installation (Kickstart), as Host OS	✗	✗
Virtual Guest Installation (image template), as Host OS	✓	✓
System deployment (PXE/Kickstart)	✓	✓
System redeployment (Kickstart)	✓	✓
Contact methods	✓ ZeroMQ, Salt-SSH	✓ ZeroMQ, Salt-SSH
Works with Uyuni Proxy	✓	✓
Action chains	✓	✓
Staging (pre-download of packages)	✓	✓
Duplicate package reporting	✓	✓
CVE auditing	✓	✓
SCAP auditing	✓	✓
Package verification	✗	✗
Package locking	✓	✓

Feature	Rocky Linux 8	Rocky Linux 9
System locking	✗	✗
Maintenance Windows	✓	✓
System snapshot	✗	✗
Configuration file management	✓	✓
Snapshots and profiles	✓ Profiles supported, Sync not supported	✓ Profiles supported, Sync not supported
Power management	✓	✓
Monitoring server	✗	✗
Monitored clients	✓	✓
Docker buildhost	✗	✗
Build Docker image with OS	✗	✗
Kiwi buildhost	✗	✗
Build Kiwi image with OS	✗	✗
Recurring Actions	✓	✓
AppStreams	✓	✓
Yomi	N/A	N/A

\* to matching SUSE Liberty Linux

\*\* Virtual Guest Management:

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

## 1.17. 支持的 Ubuntu 功能

下表列出了 Ubuntu 客户端上各种功能的可用性。



客户端上运行的操作系统由提供操作系统的组织支持。Ubuntu 由 Canonical 支持。

此表中图标的含意如下：

- ✓ 该功能可用
- ✗ 该功能不可用

- ? 该功能正在考虑之中，日后可能提供，也可能不提供

表格 17. Ubuntu 操作系统上支持的功能

Feature	Ubuntu 20.04	Ubuntu 22.04	Ubuntu 24.04
Client	✓	✓	✓
System packages	Canonical	Canonical	Canonical
Registration	✓	✓	✓
Install packages	✓	✓	✓
Apply patches	✓	✓	✓
Remote commands	✓	✓	✓
System package states	✓	✓	✓
System custom states	✓	✓	✓
Group custom states	✓	✓	✓
Organization custom states	✓	✓	✓
System set manager (SSM)	✓	✓	✓
Product migration	N/A	N/A	N/A
Basic Virtual Guest Management *	✓	✓	✓
Advanced Virtual Guest Management *	✓	✓	✓
Virtual Guest Installation (Kickstart), as Host OS	✗	✗	✗
Virtual Guest Installation (image template), as Host OS	✓	✓	✓
System deployment (PXE/Kickstart)	✗	✗	✗
System redeployment (Kickstart)	✗	✗	✗
Contact methods	✓ ZeroMQ, Salt-SSH	✓ ZeroMQ, Salt-SSH	✓ ZeroMQ, Salt-SSH
Works with Uyuni Proxy	✓	✓	✓
Action chains	✓	✓	✓
Staging (pre-download of packages)	✓	✓	✓
Duplicate package reporting	✓	✓	✓
CVE auditing	?	?	?

Feature	Ubuntu 20.04	Ubuntu 22.04	Ubuntu 24.04
SCAP auditing	?	?	?
Package verification	✗	✗	✗
Package locking	✓	✓	✓
System locking	✗	✗	✗
System snapshot	✗	✗	✗
Configuration file management	✓	✓	✓
Package profiles	✓ Profiles supported, Sync not supported	✓ Profiles supported, Sync not supported	✓ Profiles supported, Sync not supported
Power management	✓	✓	✓
Monitoring	✓	✓	✓
Docker buildhost	?	?	?
Build Docker image with OS	✓	✓	✓
Kiwi buildhost	✗	✗	✗
Build Kiwi image with OS	✗	✗	✗
Recurring Actions	✓	✓	✓
AppStreams	N/A	N/A	N/A
Yomi	N/A	N/A	N/A

#### \* 虚拟 Guest 管理：

在此表格中，虚拟 Guest 管理分为基本管理和高级管理。

基本虚拟 Guest 管理包括列出 VM、慢速刷新、VM 生命周期操作（开始、停止、继续、暂停）以及修改 VM vCPU 和内存。

高级虚拟 Guest 管理包括基本虚拟 Guest 管理的所有功能以及快速刷新、VM 生命周期操作（删除、重置、关机）、修改 VM 磁盘、网络、图形显示和图形显示配置。

# Chapter 2. 配置基本知识

Uyuni 需要您执行一些步骤来准备客户端注册环境，然后才能使用该环境的各项操作。

本章总结了在成功安装和设置 Uyuni 后为支持环境操作而必须执行的初始配置步骤。

- For more information about installing and setting up Uyuni, see [Installation-and-upgrade](#) > [Installation-and-upgrade-overview](#).

## 2.1. 软件通道

通道是一种用于对软件包分组的方法。软件包由储存库提供，而储存库与通道关联。客户端订阅软件通道后，便可安装并更新与其关联的任何软件。

在 Uyuni 中，通道分为基础通道和子通道。以此方式组织通道可确保每个系统上只安装兼容的软件包。客户端只能订阅一个基础通道，该通道是在注册期间根据客户端操作系统和体系结构指派的。对于供应商提供的付费通道，您必须具有关联的订阅。

基础通道中包含为特定操作系统类型、版本和体系结构构建的软件包。例如，SUSE Linux Enterprise Server 15 x86-64 基础通道中仅包含与该操作系统和体系结构兼容的软件。

子通道与基础通道关联，仅提供与基础通道兼容的软件包。一个系统可订阅其基础通道的多个子通道。将系统指派到基础通道后，该系统便只能安装相关的子通道。例如，如果将系统指派到 SUSE Linux Enterprise Server 15 x86\_64 基础通道，那么便只能安装或更新通过兼容基础通道或它的任何关联子通道提供的软件包。

在 Uyuni Web UI 中，您可以导航到软件，通道列表，所有来浏览可用通道。您可以通过导航到软件，管理，通道修改或创建新通道。

有关使用通道（包括自定义通道）的详细信息，请参见 [Administration](#) > [Channel-management](#)。

### 2.1.1. 通过 SUSE Package Hub 提供的软件包

SUSE Package Hub 是 SUSE Linux Enterprise 产品的扩展，用于提供额外的开源软件（由 openSUSE 社区提供）。



SUSE Package Hub 中的软件包由 openSUSE 社区提供。SUSE 不会为这些软件包提供支持。

如果您在客户端上使用的是 SUSE Linux Enterprise 操作系统，则可启用 SUSE Package Hub 扩展来访问这些额外的软件包。这样会提供 SUSE Package Hub 通道，您可以为客户端订阅这些通道。

SUSE Package Hub 提供了大量软件包，可能需要花很长时间进行同步，并会占用大量磁盘空间。除非您确实需要 SUSE Package Hub 提供的软件包，否则请不要启用它。

为了避免无意间安装或更新不支持的软件包，建议您实施一开始会拒绝所有 SUSE Package Hub 软件包的内容生命周期管理策略。然后，您可以显式启用所需的特定软件包。有关内容生命周期管理的详细信息，请参见 [Administration](#) > [Content-lifecycle](#)。

## 2.1.2. 通过 AppStream 提供的软件包

对于基于 Red Hat 的客户端，通过 AppStream 来提供额外的软件包。大多数情况下都需要 AppStream 软件包来确保您已拥有所有必需的软件。

## 2.1.3. 通过 EPEL 提供的软件包

对于基于 Red Hat 的客户端，通过 EPEL 提供额外的软件包（适用于企业版 Linux 的额外软件包）。EPEL 是可选软件包储存库，用于提供额外的软件。



- EPEL 中的软件包由 Fedora 社区提供。SUSE 不会为这些软件包提供支持。

如果您在客户端上使用的是 Red Hat 操作系统，则可启用 EPEL 扩展来访问这些额外的软件包。这样会提供 EPEL 通道，您可以为客户端订阅这些通道。

EPEL 提供了大量软件包，可能需要花很长时间进行同步，并会占用大量磁盘空间。除非您确实需要 EPEL 提供的软件包，否则请不要启用 EPEL 储存库。

为了避免无意间安装或更新不支持的软件包，建议您实施一开始会拒绝所有 EPEL 软件包的内容生命周期管理 (CLM) 策略。然后，您可以显式启用所需的特定软件包。有关内容生命周期管理的详细信息，请参见 [Administration > Content-lifecycle](#)。

## 2.1.4. SUSE Linux Enterprise 客户端的 Unified Installer 更新通道

Unified Installer 使用此通道来确保其在安装操作系统前是最新的。所有 SUSE Linux Enterprise 产品在安装期间都应能够访问安装程序更新通道。

对于 SUSE Linux Enterprise Server 客户端，默认会在您添加包含安装程序更新通道的产品时对这些通道进行同步，并会在您创建包含这些产品通道的可自动安装发行套件时予以启用。

对于所有其他 SUSE Linux Enterprise 变体，包括 SUSE Linux Enterprise for SAP，您必须手动添加安装程序更新通道。要完成此操作，请克隆这些 SUSE Linux Enterprise 变体的基础通道下的相应 SUSE Linux Enterprise Server 安装程序更新通道。克隆通道后，为这些 SUSE Linux Enterprise 变体创建可自动安装的发行套件时，会自动使用相应通道。

## 2.1.5. 软件储存库

储存库用于收集软件包。如果您有权访问软件储存库，便可以安装储存库提供的任何软件。在 Uyuni 中，必须至少有一个储存库与您的软件通道相关联，才能向该通道指派客户端并在客户端上安装和更新软件包。

Uyuni 中的大多数默认通道都已与正确的储存库关联。如果您要创建自定义通道，则需要关联您有权访问的或您自己创建的储存库。

有关自定义储存库和通道的详细信息，请参见 [Administration > Custom-channels](#)。

### 2.1.5.1. 本地储存库位置

可以在客户端上配置本地储存库，以提供 Uyuni 通道所不提供的软件包。



大多数情况下，客户端系统不需要本地储存库。本地储存库可能会导致无法确定客户端上哪些软件包可用，而这可能会导致安装非预期的软件包。

初始配置期间会禁用本地储存库。

每次执行通道状态时都会禁用本地储存库。例如，当应用 Highstate 或执行软件包操作时。

如果初始配置后本地储存库应保持启用状态，则必须为受影响的客户端设置以下 pillar：

编辑 /srv/pillar/top.sls 文件：

```
base:
  'minionid':
    - localrepos
```

编辑 /srv/pillar/localrepos.sls 文件：

```
mgr_disable_local_repos: False
```

客户端完成初始配置后，您可以在以下位置添加本地储存库：

表格 18. 本地储存库位置

客户端操作系统	本地软件源目录
SUSE Linux Enterprise Server	/etc/zypp/repos.d
openSUSE	/etc/zypp/repos.d
Red Hat Enterprise Linux 和类似衍生产品	/etc/yum.repos.d/
Ubuntu	/etc/apt/sources.list.d/
Debian	/etc/apt/sources.list.d/

### 2.1.6. 软件产品

在 Uyuni 中，软件通过不同的产品提供。利用 SUSE 订阅，您可以访问各种不同的产品，通过在 Uyuni Web UI 中导航到 **管理 > 安装向导 > 产品** 可以浏览和选择这些产品。

产品包含任意数量的软件通道。单击 图标可查看产品中包含的通道。成功添加并同步产品后，您便可以访问产品提供的通道，并可以在 Uyuni 服务器和客户端上使用产品中的软件包。

过程：添加软件通道

1. 在 Uyuni Web UI 中，导航到**管理**、**安装向导**、**产品**。
2. 使用搜索栏找到适用于您的客户端操作系统和体系结构的产品，然后选中相应产品。这样会自动选中所有必需的通道。此外，建议的所有通道也将选中，并且`-----`开关会打开。单击箭头以查看相关产品的完整列表，确保您需要的所有额外产品都已选中。
3. 单击`[添加产品]`并等待产品完成同步。

## 2.1.7. AppStream 模块

如果您正在管理已订阅 AppStream 通道的客户端，Uyuni 将根据当前启用的 AppStream 模块集来过滤客户端可用的软件包。

过程：管理已启用的 AppStream 模块

1. 在 Uyuni Web UI 中，导航到**系统**、**软件**、**AppStream**。
2. 在模块流表中，选择要在客户端上启用的流。
3. 单击`[应用更改]`。
4. 在下一页中，检查要做出的更改列表，并选择执行操作的时间。
5. 单击`[确认]`安排更改的应用。

## 2.2. 引导储存库

引导储存库包含引导期间注册客户端所需的软件包。同步产品时，会自动在 Uyuni 服务器上创建及重新生成引导储存库。

### 2.2.1. 准备创建引导储存库

如果您选择某个产品以进行同步，当所有必需的通道都完全镜像后，会立即自动创建引导储存库。

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到**软件**、**管理**、**通道**，然后单击与储存库关联的通道。
2. 导航到`-----`选项卡，然后单击`-----`并选中`-----`。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

## 2.2.2. 自动模式的选项

您可以更改自动引导储存库的创建方式。本节介绍了各种相应设置。

刷新模式：

### 刷新模式

默认只会使用最新软件包更新现有储存库，而您可以将其配置为始终从空储存库开始。要启用此行为，请在 `/etc/rhn/rhn.conf` 中添加或编辑此值：

```
server.susemanager.bootstrap_repo_flush = 1
```

自动模式：

### 自动模式

默认会启用引导储存库的自动重新生成功能。要禁用此功能，请在 `/etc/rhn/rhn.conf` 中添加或编辑此值：

```
server.susemanager.auto_generate_bootstrap_repo = 0
```

## 2.2.2.1. 配置引导数据文件

该工具使用包含有关每个发行套件所需软件包的信息的数据文件。该数据文件存储在 `/usr/share/susemanager/mgr_bootstrap_data.py`。SUSE 会定期更新此文件。如果您要更改此文件，请不要直接编辑，而是应在同一目录中创建一份副本，然后编辑该副本：

```
cd /usr/share/susemanager/
cp mgr_bootstrap_data.py my_data.py
```

更改后，将 Uyuni 配置为使用这个新文件。在 `/etc/rhn/rhn.conf` 中添加或编辑此值：

```
server.susemanager.bootstrap_repo_datamodule = my_data
```



下次更新时，SUSE 提供的新数据将重写原来的数据文件，而不是这个新文件。您需要在新文件中使用 SUSE 提供的更改覆盖相应内容，以使其保持最新。

## 2.2.3. 手动生成引导储存库

默认情况下，每天都会重新生成引导储存库。您可以在命令提示符处手动创建引导储存库。

过程：生成 SUSE Linux Enterprise 的引导储存库

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份列出要为其创建引导储存库的可用发行套件：

```
mgr-create-bootstrap-repo -l
```

2. 创建引导储存库，并使用适当的储存库名称作为产品标签：

```
mgr-create-bootstrap-repo -c SLE-version-x86_64
```

3. 或者，使用可用发行套件列表中发行套件名称旁边显示的编号。

客户端储存库位于 `/srv/www/htdocs/pub/repositories/` 中。

如果您镜像了多个产品（例如 SLES 和 SLES for SAP），或者您使用的是自定义通道，在创建引导储存库时将需要指定要使用的父通道。并不是在所有情况下都需要如此。例如，部分 SLES 15 版本使用共同的代码库，因此无需指定父通道。只有您的环境需要时，才需使用此过程。

可选过程：指定引导储存库的父通道

1. 检查您有哪些父通道可用：

```
mgr-create-bootstrap-repo -c SLE-15-x86_64
```

找到多个父通道选项。请使用

`--with-parent-channel <label>` 选项并选择以下其中一个父通道：

- `sle-product-sles15-pool-x86_64`
- `sle-product-sles_sap15-pool-x86_64`
- `sle-product-sled15-pool-x86_64`

2. 指定适当的父通道：

```
mgr-create-bootstrap-repo -c SLE-15-x86_64 --with-parent-channel
sle-product-sled15-pool-x86_64
```

### 2.2.3.1. 包含多个体系结构的储存库

如果您要创建包含多个不同体系结构的引导储存库，则需要确保所有体系结构是否都已正确更新。例如，SLE 的 x86-64 和 IBM Z 体系结构使用相同的引导储存库 URL：`/srv/www/htdocs/pub/repositories/sle/15/2/bootstrap/`。

如果启用 `flush` 选项，当您尝试生成多个体系结构的引导储存库时，将仅生成一个体系结构。要避免此问题，请在创建其他体系结构时于命令提示符处使用 `--no-flush` 选项。例如：

```
mgr-create-bootstrap-repo -c SLE-15-SP2-x86_64
mgr-create-bootstrap-repo --no-flush -c SLE-15-SP2-s390x
```

## 2.2.4. 引导和自定义通道

如果您要使用自定义通道，则可以在 `mgr-create-bootstrap-repo` 命令中使用 `--with-custom-channels` 选项。在此情况下，您还需要指定要使用的父通道。

如果您使用自定义通道，则自动创建引导储存库可能会失败。在这种情况下，您需要手动创建储存库。

有关自定义通道的详细信息，请参见 [Administration > Custom-channels](#)。

## 2.3. 激活密钥

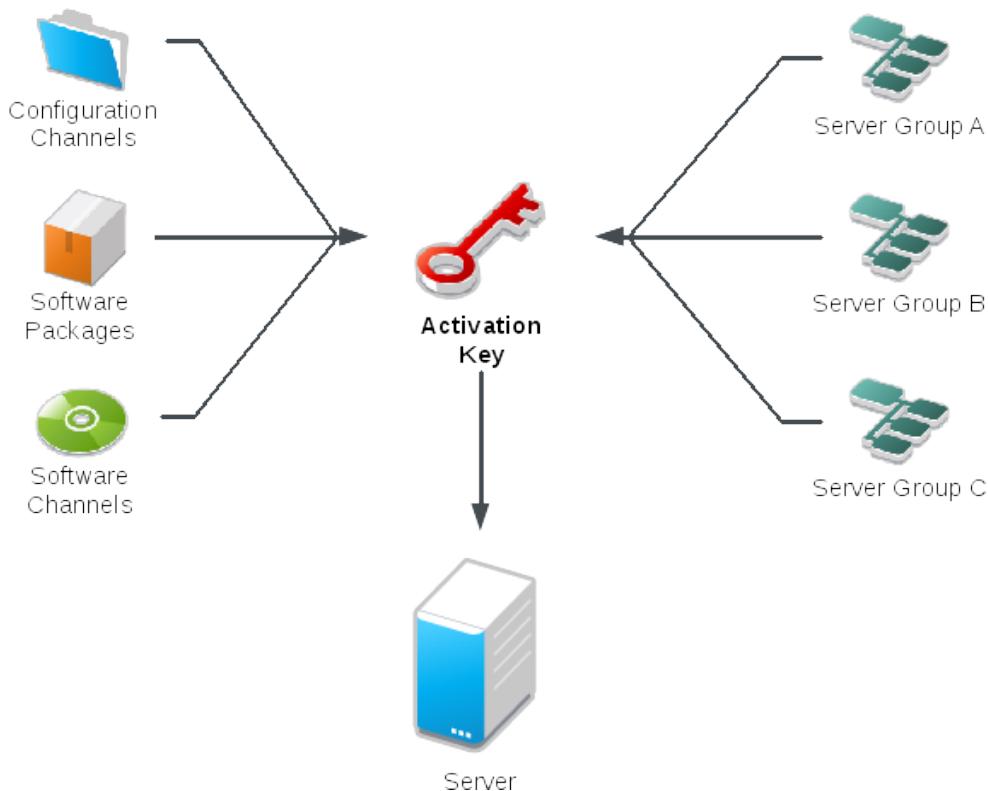
激活密钥用于确保客户端拥有正确的软件权利、可连接到适当的通道以及订阅相关的组。每个激活密钥都绑定到一个组织，您可以在创建密钥时设置此项。

在 Uyuni 中，激活密钥是一组带有标签的配置设置。您可以通过将某个激活密钥的标签作为参数添加到引导脚本，来应用与该激活密钥关联的所有配置设置。我们建议您结合引导脚本来使用激活密钥标签。当执行引导脚本时，与该标签关联的所有配置设置都将应用到运行脚本的系统上。

激活密钥可以指定以下各项：

- 通道指派
- 系统类型或附加权利
- 联系方法
- 配置文件
- 要安装的软件包
- AppStreams to be enabled
- 系统组

在注册客户端时需使用激活密钥，之后将不再使用。无论激活密钥指定了什么内容，注册好客户端后，就能以任何方式对其进行更改。记录激活密钥与客户端之间的关联仅为了历史记载。



过程：创建激活密钥

1. 在 Uyuni Web UI 中，以管理员身份导航到系统，激活密钥。
2. 单击 **[ 创建激活密钥 ]** 按钮。
3. 在 **说明** 页面的 **说明** 字段中，输入激活密钥的说明。
4. 在 **名称** 字段中输入激活密钥的名称。例如，SLES15-SP5 表示 SUSE Linux Enterprise Server 15 SP5。



对于任何 SUSE 产品，请不要在 **名称** 字段中使用逗号或双引号。不过，对于 Red Hat 产品，则 **必须** 使用逗号。

- 所有其他字符均可使用，不过系统会自动去除 <> (){}（包括空格）。
- 如果将字段留空，系统会生成一个随机字符串。

5. 在 **基础软件通道** 下拉框中，选择适当的基础软件通道，并允许填充相关子通道。有关详细信息，请参见 [reference:admin/setup-wizard.pdf](#) 和 **Administration > Custom-channels**。
6. 选择所需的子通道（例如，必需的 SUSE Manager 工具和更新通道）。
7. 如果需要启用任何选项，请选中 **自动激活** 复选框。
8. 建议您保留 **自动激活** 设置的 **自动激活**。
9. 建议您将 **自动激活** 设置保留未选中状态。
10. 单击 **[ 创建激活密钥 ]** 以创建激活密钥。

11. 选中复选框以对此密钥启用配置管理，然后单击 **[更新激活密钥]** 保存此更改。



在您创建激活密钥前，复选框将不会显示。如果您需要启用配置管理，请务必返回并选中该复选框。

When created, tabs such as

- Packages
- Configuration
- Groups
- Activated Systems
- AppStreams (available whenever a modular channel is associated with the activation key)

allow you to check and set additional features.

### 2.3.1. 重新激活密钥

重新激活密钥可用来重新注册客户端并重新获得所有 Uyuni 设置，但只能使用一次。重新激活密钥为客户端特定密钥，包含系统 ID、历史记录、组和通道。

要创建重新激活密钥，请导航到，单击要为其创建重新激活密钥的客户端，然后导航到**细节**，**重新激活**选项卡。单击**[生成新密钥]**创建重新激活密钥。记录密钥细节以供日后使用。与未与特定系统关联的典型激活密钥不同，此处创建的密钥不会显示在**系统**，**激活密钥**页面中。

创建重新激活密钥后，可在 `/etc/salt/minion.d/susemanager.conf` 中将其作为 `management_key` grain 使用。例如：

```
grains:
  susemanager:
    management_key: "re-1-daf44db90c0853edbb5db03f2b37986e"
```

重启动 `venv-salt-minion` 或 `salt-minion` 进程以应用重新激活密钥。

您可以结合引导脚本来使用重新激活密钥。有关引导脚本的详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。



如果您使用客户端现有的 Uyuni 配置文件自动安装客户端，该配置文件会使用重新激活密钥来重新注册系统并恢复其设置。请勿在正在进行基于配置文件的自动安装时重新生成、删除或使用此密钥，否则会导致自动安装失败。

### 2.3.2. 激活密钥最佳实践

默认父通道

避免使用 SUSE Manager Default 父通道。此设置会强制 Uyuni 选择与所安装操作系统最匹配的父通道，而这有时可能会导致非预期的行为。请改为创建特定于每个发行套件和体系结构的激活密钥。

#### 使用激活密钥引导

如果您要使用引导脚本，请考虑为每个脚本创建一个激活密钥。这有助于您调整通道指派、软件包安装、系统组成员资格和配置通道指派。注册后，您需要与系统手动交互的情形也会减少。

#### 引导 LTSS 客户端

如果您要引导订阅了 LTSS 的客户端，请在创建激活密钥期间包含 LTSS 通道。

#### 带宽要求

使用激活密钥可能导致在注册时自动下载软件，这对于存在带宽限制的环境来说可能并不适宜。

下列选择会产生带宽用量：

- 指派 SUSE Product Pool 通道会导致自动安装相应的产品描述符软件包。
- 安装部分中的所有软件包。
- 部分中的任何 Salt 状态都可能会触发下载，具体视其内容而定。

#### 密钥标签命名

如果您没有为激活密钥输入直观易懂的名称，系统会自动生成一串数字字符串，这会使得您的密钥难以管理。

请考虑为您的激活密钥使用一种命名模式，以方便您进行跟踪。设计与您组织的基础结构相关的名称有助于您更轻松地执行较复杂的操作。

在创建密钥标签时，请考虑以下提示：

- 操作系统命名（必需）：密钥应始终代表提供的设置所适用的操作系统
- 体系结构命名（建议）：除非您的公司仅在一个体系结构上运行（例如 x86\_64），否则最好提供含体系结构类型的标签。
- 服务器类型命名：此服务器的用途是什么？
- 位置命名：服务器位于何处？机房、建筑物或部门？
- 日期命名：维护时段、季度等。
- 自定义命名：哪种命名模式符合您组织的需求？

激活密钥标签名称示例：

```
sles15-sp4-web_server-room_129-x86_64
```

```
sles15-sp4-test_packages-blg_502-room_21-ppc64le
```

包含的通道

在创建激活密钥时，您还需要注意哪些软件通道与其关联。应为密钥指派特定的基础通道，不建议使用默认基础通道。有关详细信息，请在 **Client-configuration > Registration-overview** 中查看您要安装的客户端操作系统。

## 2.4. GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。

在大多数情况下，您无需调整 GPG 设置即可在您的客户端上安装软件。

可以直接对 RPM 软件包签名，而基于 Debian 的系统仅会对元数据签名并使用校验和链来保障软件包的安全。大多数基于 RPM 的系统除了使用已签名软件包外，还会使用已签名人数据。

### 2.4.1. 在客户端上信任 GPG 密钥

操作系统要么直接信任自己的 GPG 密钥，要么至少将它们与最小系统一起安装，但通过其他 GPG 密钥签名的第三方软件包需要手动处理。客户端可以在不信任 GPG 密钥的情况下成功引导，但除非密钥受信任，否则您将无法安装新的客户端工具软件包或更新软件包。

客户端现在使用为软件通道输入的 GPG 密钥信息来管理受信任密钥。如果将具有 GPG 密钥信息的软件通道指派给客户端，当通道刷新后或从此通道安装第一个软件包后，系统就会信任该密钥。

软件通道页面中的 GPG 密钥 URL 参数可以包含以空格分隔的多个密钥 URL。如果其为文件 URL，则必须在使用软件通道之前在客户端上部署 GPG 密钥文件。

基于 Red Hat 的客户端的客户端工具通道的 GPG 密钥会部署到客户端的 `/etc/pki/rpm-gpg/` 中，并可通过文件 URL 引用。

仅当为客户端指派了软件通道时，系统才会导入并信任这些密钥。



由于基于 Debian 的系统仅会对元数据签名，因此不需要为各个通道指定确切的密钥。如果用户按 **Administration > Repo-metadata** 中的“使用自己的 GPG 密钥”所述配置了自己的 GPG 密钥来对元数据签名，系统会自动部署并信任该密钥。

#### 2.4.1.1. 用户定义的 GPG 密钥

用户可以定义要部署到客户端的自定义 GPG 密钥。

通过提供某些 pillar 数据并在 Salt 文件系统中提供 GPG 密钥文件，系统会自动将它们部署到客户端。

这些密钥会分别部署到 `/etc/pki/rpm-gpg/`（在基于 RPM 的操作系统上）和 `/usr/share/keyrings/`（在 Debian 系统上）中：

为您要将密钥部署到的客户端定义 pillar 键 `custom_gpgkeys` 并列出密钥文件名。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

此外，在 Salt 文件系统中创建名为 gpg 的目录，并将 custom\_gpgkeys pillar 数据中所指定的 GPG 密钥文件存储在该目录下。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

密钥将部署到客户端的 `/etc/pki/rpm-gpg/my_first_gpg.key` 和 `/etc/pki/rpm-gpg/my_second_gpgkey.gpg` 中。

最后一步是将 URL 添加到软件通道的“GPG 密钥 URL”字段中。请导航到软件，管理，通道，然后选择要修改的通道。将值 `file:///etc/pki/rpm-gpg/my_first_gpg.key` 添加到 GPG URL 中。

### 2.4.1.2. 引导脚本中的 GPG 密钥

过程：在客户端上使用引导脚本信任 GPG 密钥

1. 在 Uyuni 服务器上的命令提示符处，检查 `/srv/www/htdocs/pub/` 目录的内容。此目录包含所有可用的公共密钥。记下为您正在注册的客户端指派的通道适用的密钥。
2. 打开相关的引导脚本，找到 `ORG_GPG_KEY=` 参数并添加所需的密钥。例如：

```
uyuni-gpg-pubkey-0d20833e.key
```

您无需删除任何以前存储的密钥。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。如果不信任 GPG 密钥，便无法为客户端指派软件通道。

# Chapter 3. 客户端联系方法

Uyuni 服务器可以使用三种联系方法来与客户端通讯。这些方法基于 Salt 协议。

使用的 Salt 联系方法取决于客户端类型以及网络体系结构：

## 默认 (Salt)

默认选项，除非有特殊需求，否则建议使用此方法。有关详细信息，请参见 [Client-configuration > Contact-methods-default](#)。

## SSH 推送

仅当由于网络限制而导致客户端无法与服务器建立联系时才有用。只能通过 Salt 捆绑包支持此方法。此联系方法有严重的限制。有关详细信息，请参见 [Client-configuration > Contact-methods-saltssh](#)。

## SSH 推送（使用隧道）

与 SSH 推送相同，但使用安全通讯隧道。有关详细信息，请参见 [Client-configuration > Contact-methods-saltssh-tunnel](#)。

Salt 通讯软件实现选项：

## Salt 捆绑包

是包含 Salt 受控端、Python 3、所需 Python 模块和库的单个二进制软件包。因此，Salt 联系方法独立于客户端上安装的软件。默认会使用 Salt 捆绑包，它是唯一支持用于 SSH 推送或 SSH 推送（使用隧道）联系方法的实现方案。有关详细信息，请参见 [Client-configuration > Contact-methods-saltbundle](#)。

## Salt 受控端

安装在客户端系统上的 Salt 软件。



Uyuni 5.0 及更高版本不再支持所谓的传统联系协议。在从 Uyuni 4 升级到 5 之前，必须将所有现有的传统客户端（包括传统代理）迁移到 Salt，或将其替换为 Salt 代理。

有关将传统 Uyuni 4 客户端迁移到 Salt 客户端的详细信息，请参见 <https://documentation.suse.com/suma/4.3/en/suse-manager/client-configuration/contact-methods-migrate-traditional.html>

## 3.1. 默认联系方法 (Salt)

除非有特殊需求，否则建议使用协议为 Salt 的默认联系方式。有关 Salt 的其他一般信息，请参见 [Specialized-guides > Salt](#)。

软件更新通常从服务器推送到客户端。连接将从客户端发起。这意味着，您必须在服务器而不是客户端上打开端口。Salt 客户端也称为 Salt 受控端。Uyuni 服务器将在每个客户端上安装守护程序。

如果您需要在断开连接的设置中使用 Salt 客户端，可以配置 SSH 推送作为联系方法。如果使用此联系方法，可将客户端定位在防火墙保护的区域（称为 DMZ）中。有关 SSH 推送的详细信息，请参见 [Client-configuration > Contact-methods-saltssh](#)。

`configuration > Contact-methods-saltssh.`

### 3.1.1. 初始配置细节

Salt 有自身的数据库用于保存受控端的密钥。此数据库需与 Uyuni 数据库保持同步。一旦 Salt 中接受了密钥，Uyuni 中的初始配置过程即会开始。

初始配置过程通过搜索 `minion_id` 和 `machine_id` 在 Uyuni 数据库中查找现有系统。根据搜索结果，可能会出现以下情况：

- 如果未找到任何系统，则会创建新系统。
- 如果找到包含 `minion_id` 或 `machine_id` 的项，则会迁移该系统以与新系统匹配。
- 如果找到匹配这两项的系统，并且它们不是同一个系统，则会中止初始配置并显示错误。
- 在这种情况下，管理员需要至少去除一个系统来解决冲突。

## 3.2. SSH 推送联系方法

SSH 推送 (`ssh-push`) 用于 Salt 客户端无法直接访问 Uyuni 服务器的环境。在此环境中，客户端位于受防火墙保护的区域（称为 DMZ）。DMZ 内的所有系统均无权打开连至 Uyuni 服务器所在内部网络的连接。

如果无法在客户端上安装守护程序代理，也可以使用 SSH 推送。



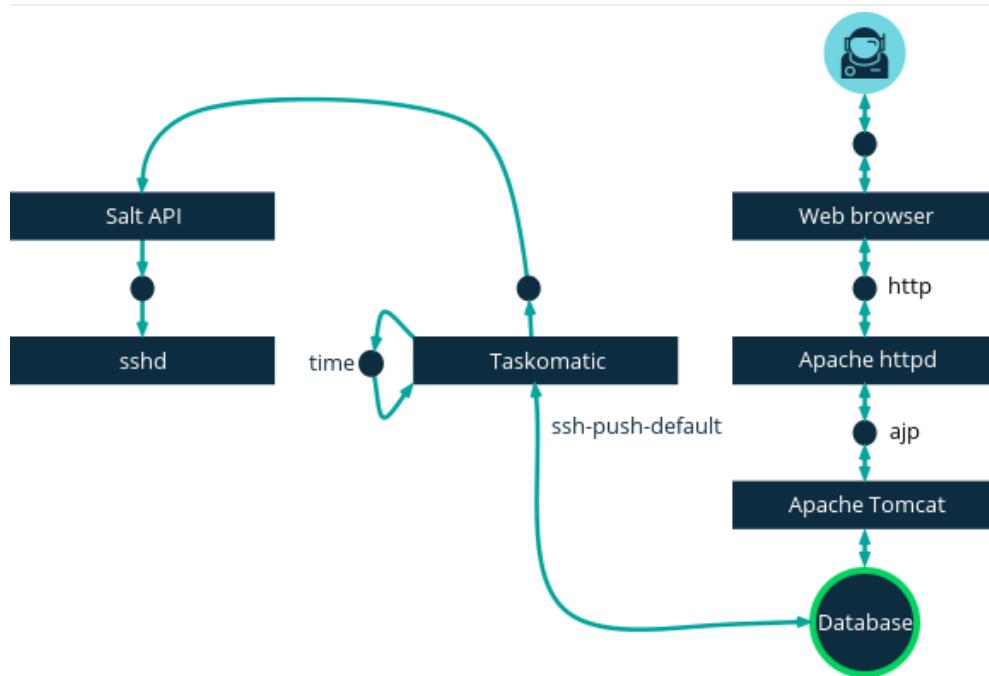
SSH 推送方法存在严重的局限性。它的缩放能力不是很好，并且比普通 Salt 方法 (→→→) 消耗的服务器资源和网络带宽更多。大型设置（1000 或更多个客户端）完全不支持 SSH 推送方法。

服务器使用 SSH 推送来定期联系客户端，以签入和执行安排的操作和事件。



在使用 SSH 推送方法管理的客户端上，目前不支持使用置备模式重新安装系统。

下图说明了 SSH 推送的进程路径。Taskomatic 块左侧的各项表示在 Uyuni 客户端上运行的进程。



要使用 SSH 推送，必须在客户端上运行 SSH 守护程序，并且 Uyuni 服务器上运行的 salt-api 守护程序必须能够访问 SSH 守护程序。此外，将使用 Salt 编包在客户端系统上部署所需的 Python 版本。

在开始执行以下注册过程之前，请先在配置了 SSH 推送联系方法的情况下定义一个激活密钥。此方法要求与服务器建立 HTTPS 直接连接。

需要使用 Web UI 或 API 将这些客户端注册到 Uyuni 服务器。请参见以下过程或示例。

过程：使用 SSH 推送注册客户端

1. 在 Uyuni Web UI 中，导航到 **系统 > 引导**，然后填写相应的字段。
2. 选择一个配置了 **SSH** 推送联系方法的激活密钥。有关激活密钥的详细信息，请参见 **Client-configuration > Activation-keys**。
3. 选中 **SSH** 复选框。
4. 单击 **[Bootstrap]** 开始注册。
5. 导航到 **系统 > 概览**，确认该系统已正确注册。

示例：通过 API 使用 SSH 推送方法

您可以使用 API 来管理要使用的联系方法。下面的示例 Python 代码将联系方法设为 `ssh-push`。

有效值为：

- `default (pull)`
- `ssh-push`
- `ssh-push-tunnel`

```
client = xmlrpclib.Server(SUMA_HOST + "/rpc/api", verbose=0)
key = client.auth.login(SUMA_LOGIN, SUMA_PASSWORD)
client.system.setDetails(key, 1000012345, {'contact_method' : 'ssh-push'})
```

### 3.2.1. 可用参数

如果您要配置通过 SSH 推送，可以修改注册系统时使用的参数，包括主机、激活密钥和口令。口令只能用于引导，不会保存在任何位置。所有将来的 SSH 会话均通过密钥/证书对获得授权。这些参数在系统>引导中配置。

您也可以配置在系统范围使用的持久性参数，包括配置 sudo 用户来以非特权用户而不是 root 身份访问系统。

过程：配置非特权 SSH 访问

1. 确保您已在 Uyuni 服务器上安装最新的 spacewalk-taskomatic 和 spacewalk-certs-tools 软件包。
2. 在每个客户端系统上，创建相应的非特权用户。
3. 在每个客户端系统上，编辑 sudoers 文件：

```
sudo visudo
```

4. 在 sudoers 文件末尾添加下面一行，以向用户授予 sudo 访问权限。以在 Web UI 中引导客户端的用户的名称替换 <user>：

```
<user>  ALL=NOPASSWD: /usr/bin/python3, /var/tmp/venv-salt-minion/bin/python
```



此过程无需口令便可授予 root 访问权限，而注册客户端需要提供口令。客户端成功安装后会以 root 特权运行，因此将不再需要该访问权限。客户端成功安装之后，建议您从 sudoers 文件中去除该行。

5. 在 Uyuni 服务器上，于 /etc/rhn/rhn.conf 配置文件中添加或修改下面一行以包含非特权用户名：

```
ssh_push_sudo_user = <user>
```

更改此配置参数后，必须重启 salt-secrets-config.service、tomcat.service 和 taskomatic.service 等服务。为了涵盖全部所需服务，最好以 root 身份重启动 Uyuni 服务器：

```
mgradm restart
```

## 3.2.2. 操作的执行

SSH 推送功能使用 taskomatic 来通过 salt-ssh 执行安排的操作。taskomatic 作业会定期检查并执行安排的操作。SSH 推送功能根据安排的操作执行完整的 salt-ssh 调用。

默认可以同时执行 20 项 Salt SSH 操作。您可以在配置文件中添加下面几行，并上调 parallel\_threads 的值，以增加可同时执行的操作数。建议您将并行操作数设置为较低的值，以免出现问题：

```
taskomatic.sshminion_action_executor.parallel_threads = <number>
org.quartz.threadPool.threadCount = <value of parallel_threads + 20>
```

这样可调整任何客户端上同时运行的操作数，以及 taskomatic 使用的工作器线程总数。如果需要在多个客户端上运行操作，则每个客户端上的操作始终按顺序执行。

如果客户端是通过代理连接的，则需要调整代理上的 MaxSessions 设置。在此情况下，请将并行连接的数量设置为客户端总数的三倍。

## 3.2.3. 未来的功能

SSH 推送目前尚不受支持某些功能。这些功能在 Salt SSH 客户端上不可用：

- OpenSCAP 审计
- 导致以下事件的信标：
  - 使用 zypper 在系统上安装软件包不会调用软件包刷新。
  - 如果虚拟主机系统基于 Salt SSH，则虚拟主机功能（例如 Guest 主机）将无法正常工作。

### 3.2.3.1. 更多信息

- 有关 Salt SSH 的一般信息，请参见 **Specialized-guides** › **Salt** 和 <https://docs.saltstack.com/en/latest/topics/ssh/>。
- 有关 SSH 密钥轮替的信息，请参见 [specialized-guides:salt/salt-ssh.pdf](#)。

## 3.3. SSH 推送（使用隧道）联系方法

SSH 推送（使用隧道）(ssh-push-tunnel) 用于客户端无法直接访问 Uyuni 服务器的环境。在此环境中，客户端位于受防火墙保护的区域（称为 DMZ）。DMZ 内的所有系统均无权打开连至内部网络（包括 Uyuni 服务器）的连接。

此 SSH 方法会创建一个加密隧道，该隧道从内部网络上的 Uyuni 服务器连到位于 DMZ 中的客户端。执行完所有操作和事件之后，该隧道即会关闭。

服务器使用 SSH 定期联系客户端，以签入和执行安排的操作和事件。



在使用 SSH 推送方法管理的客户端上，目前不支持使用置备模式重新安装系统。



隧道用于通过加密隧道提供对服务器的访问。指派给 SSH 推送客户端（使用隧道）的储存库只会通过此隧道提供，因此无法直接从客户端系统使用软件包管理器工具，因为储存库仅在隧道已启动时才可用。换而言之，仅当会话由服务器发起时才可访问储存库。客户端上的所有软件包管理操作只能从服务器端执行。

要通过 SSH 实现隧道连接，需要提供一个端口号用于通过 HTTPS 建立隧道。默认使用的端口号为 1233。要重写此端口号，可以在 `/etc/rhn/rhn.conf` 中添加大于 1024 的自定义端口号：

```
ssh_push_port_https = high_port
```

更改此配置参数后，必须重启动 `salt-secrets-config.service`、`tomcat.service` 和 `taskomatic.service` 等服务。为了涵盖全部所需服务，最好以 root 身份重启动 `spacewalk-service`：

```
spacewalk-service restart
```

出于安全原因，您可能需要结合使用 `sudo` 和 SSH，以非特权用户身份而不是 root 身份访问系统。

过程：配置非特权 SSH 访问

1. 在每个客户端系统上，创建相应的非特权用户。
2. 在每个客户端系统上，编辑 `sudoers` 文件：

```
sudo visudo
```

3. 在 `sudoers` 文件末尾添加下面一行，以向用户授予 `sudo` 访问权限。以在 Web UI 中引导客户端的用户的名称替换 `<user>`：

```
<user>  ALL=NOPASSWD: /usr/bin/python3, /var/tmp/venv-salt-minion/bin/python
```



此过程无需口令便可授予 root 访问权限，而注册客户端需要提供口令。客户端成功安装后会以 root 特权运行，因此将不再需要该访问权限。客户端成功安装之后，建议您从 `sudoers` 文件中去除该行。

4. 在 Uyuni 服务器上，于 `/etc/rhn/rhn.conf` 配置文件中添加或修改下面一行以包含非特权用户名：

```
ssh_push_sudo_user = <user>
```

更改此配置参数后，必须重启动 `salt-secrets-config.service`、`tomcat.service` 和 `taskomatic.service` 等服务。为了涵盖全部所需服务，最好以 root 身份重启动 `spacewalk-service`：

```
spacewalk-service restart
```

需要使用 Web UI 或 API 将这些客户端注册到 Uyuni 服务器。

开始前，需要确保已指定要用于 SSH 隧道的端口。如果在更改端口号之前已注册客户端，则需要使用重新激活密钥再次注册客户端。

- 有关引导的详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。
- 有关引导的详细信息，请参见 [client-configuration:activation-keys.pdf](#)。

示例：通过 API 使用 SSH 推送方法（利用隧道）

可以使用 API 来管理要使用的联系方法。以下示例 Python 代码将联系方法设为 ssh-push-tunnel。

有效值为：

- default (pull)
- ssh-push
- ssh-push-tunnel

```
client = xmlrpclib.Server(SUMA_HOST + "/rpc/api", verbose=0)
key = client.auth.login(SUMA_LOGIN, SUMA_PASSWORD)
client.system.setDetails(key, 1000012345, {'contact_method' : 'ssh-push-tunnel'})
```

## 3.4. Salt 捆绑包

### 3.4.1. 什么是 Salt 捆绑包？

Salt 捆绑包是单个二进制软件包，包含 Salt 受控端、Python 3、所需的 Python 模块和库。

Salt 捆绑包随附 Python 3 以及运行 Salt 所需满足的所有条件。因此 Salt 捆绑包不会将客户端上安装的 Python 版本用作系统软件。Salt 捆绑包可以安装在不满足指定 Salt 版本的要求的客户端上。

此外，还可以在所运行 Salt 受控端连接到 Uyuni Salt 主控端以外的 Salt 主控端的系统上使用 Salt 捆绑包。

### 3.4.2. 使用 Salt 捆绑包将客户端注册为受控端

建议的注册方法是使用 Salt 捆绑包进行注册。本节介绍当前实现的优点和局限性。Salt 捆绑包以 venv-salt-minion 的形式提供，其中包含 Salt、Python 3 以及 Salt 依赖的 Python 模块。通过 Web UI 引导时使用的也是 Salt 捆绑包，因此通过 Web UI 引导不依赖 Python 来进行。借助 Salt 捆绑包，客户端不再需要提供任何 Python 解释器或模块。

如果您引导新客户端，默认的注册方法是使用 Salt 捆绑包注册。您可以将现有客户端切换为使用 Salt 捆绑包方

法。如果切换，将会安装 salt-minion 软件包及其依赖项。

### 3.4.2.1. 将 Salt 捆绑包与 Salt 受控端结合使用

可以同时使用 Salt 捆绑包与由 Uyuni 服务器以外的 Salt 主控端管理的 Salt 受控端。如果将 Salt 捆绑包安装在客户端上，Uyuni 服务器将会管理 Salt 捆绑包的配置文件，在此情况下，salt-minion 的配置文件将不会受到管理。有关详细信息，请参见 [Salt 捆绑包配置](#)。



- 如果某个客户端的 Salt 受控端由 Uyuni 服务器以外的 Salt 主控端进行管理，要引导该客户端，建议在生成引导脚本时使用 `mgr-bootstrap --force -bundle`，或在引导脚本中将 `FORCE_VENV_SALT_MINION` 设为 1。
- 要使用 Web UI 进行引导，可以全局指定 `mgr_force_venv_salt_minion` 设为 `true` 的 pillar。有关详细信息，请参见 [Specialized-guides > Salt](#)。

### 3.4.2.2. 从 Salt 受控端切换到 Salt 捆绑包

可以使用 Salt 状态 `util.mgr_switch_to_venv_minion` 从 salt-minion 切换到 venv-salt-minion。建议分两步来切换到 venv-salt-minion，以免在切换过程中出现任何问题：

过程：使用 `util.mgr_switch_to_venv_minion` 状态切换到 venv-salt-minion

1. 首先，在不指定任何 pillar 的情况下应用 `util.mgr_switch_to_venv_minion`。这样会通过复制配置文件等数据切换到 venv-salt-minion。此过程不会清理原始的 salt-minion 配置及其软件包。

```
salt <minion_id> state.apply util.mgr_switch_to_venv_minion
```

在 `mgr_purge_non_venv_salt` 设为 `True`（用于去除 salt-minion）以及 `mgr_purge_non_venv_salt_files` 设为 `True`（用于去除与 salt-minion 相关的所有文件）的情况下应用 `util.mgr_switch_to_venv_minion`。此第二步用于确保第一步已经处理，然后去除旧配置文件以及现已过时的 salt-minion 软件包。

```
salt <minion_id> state.apply util.mgr_switch_to_venv_minion
pillar='{"mgr_purge_non_venv_salt_files": True,
"mgr_purge_non_venv_salt": True}'
```



如果切换过程中跳过第一步来运行第二步，状态应用过程可能会失败，因为此过程需要停止用于在客户端执行命令的 salt-minion。

反过来，您也可以不安装 Salt 捆绑包，而是继续使用 salt-minion。在此情况下，请指定以下其中一个选项：

- 指定 `--no-bundle` 选项来执行 `mgr-bootstrap`。
- 在生成的引导脚本中，将 `AVOID_VENV_SALT_MINION` 设为 1。

- 对于引导状态，将 `mgr_avoid_venv_salt_minion` pillar 设为 `True`。

### 3.4.3. 使用 Salt 捆绑包执行 SSH 推送

在对客户端执行 SSH 推送操作时，也可使用 Salt 捆绑包。

在执行任何 Salt 命令前，外壳脚本会将 Salt 捆绑包部署到目标系统上，而不会安装 `venv-salt-minion`。由于 Salt 捆绑包包含整个 Salt 代码库，因此不会部署 `salt-thin`。SSH 推送（包括使用 Web UI 进行引导）使用捆绑包中的 Python 3 解释器。目标系统上不需要安装任何其他 Python 解释器。

随该捆绑包部署的 Python 3 用于处理客户端上的 SSH 推送会话，因此 SSH 推送（包括使用 Web UI 进行引导）不依赖于在系统上安装 Python。

可以将 `salt-thin` 用作一种后备方法，但这种方法需要在客户端上安装 Python 3。不建议也不支持使用此方法，此方法仅用于开发目的。请在 `/etc/rhn/rhn.conf` 配置文件中将 `web.ssh_use_salt_thin` 设为 `true`。

- 
- 使用 Web UI 引导客户端之前必须创建引导软件源。SSH 推送使用根据检测到的目标操作系统从引导软件源获取的 Salt 捆绑包。有关详细信息，请参见[client-configuration:bootstrap-repository.pdf](#)。
  - SSH 推送使用 `/var/tmp` 在安装了捆绑 Python 的客户端上部署 Salt 捆绑包和执行 Salt 命令。因此，切勿使用 `noexec` 选项挂载 `/var/tmp`。无法通过 Web UI 引导使用 `noexec` 选项挂载 `/var/tmp` 的客户端，因为引导过程使用 SSH 推送来访问客户端。

### 3.4.4. 使用 pip 通过 Python 软件包扩展 Salt 捆绑包

Salt 捆绑包提供了 `pip`，使用该命令可以通过额外的 Python 软件包扩展捆绑 Salt 受控端的功能。

默认情况下，`salt <minion_id> pip.install <package-name>` 会将 `<package_name>` 指定的 Python 软件包安装到 `/var/lib/venv-salt-minion/local` 中。



如果需要，可以通过为 `venv-salt-minion.service` 设置 `VENV_PIP_TARGET` 环境变量来覆盖路径 `/var/lib/venv-salt-minion/local`。建议为该服务使用 `systemd` 普适性配置文件。可以使用配置文件 `/etc/systemd/system/venv-salt-minion.service.d/10-pip-destination.conf` 来实现该目的：

```
[Service]
Environment=VENV_PIP_TARGET=/new/path/local/venv-salt-
minion/pip
```



更新 Salt 捆绑包时，通过 `pip` 安装的 Python 软件包不会发生变化。为了确保更新后这些软件包可用且可正常运行，建议在 Salt 捆绑包更新后在应用了 Salt 状态的情况下安装这些软件包。

# Chapter 4. 客户端注册

有多种方法可将客户端注册到 Uyuni 服务器。本章介绍了各种可用的方法。此外，还提供了特定于您要在客户端上运行的操作系统的信息。

开始前，请进行以下检查：

- 注册前客户端的日期和时间已与 Uyuni 服务器正确同步。
- 您已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。



请不要将 Uyuni 服务器基础操作系统注册到 Uyuni 自身。必须单独管理 Uyuni 服务器基础操作系统，或使用另一个独立的 Uyuni 服务器进行管理。

要管理 Uyuni 服务器容器，请使用 `mgradm` 工具。

## 4.1. 注册方法

有多种方法可将客户端注册到 Uyuni 服务器。

- 对于少数客户端，我们建议使用 Uyuni Web UI 对其进行注册。有关详细信息，请参见 [Client-configuration > Registration-webui](#)。
- 如果您想更好地控制注册过程或必须注册许多客户端，我们建议您创建引导脚本。有关详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。
- 如果您想进一步提高注册过程的控制度，在命令行上执行单个命令可能更好。有关详细信息，请参见 [Client-configuration > Registration-cli](#)。

注册前，客户端的日期和时间必须已与 Uyuni 服务器正确同步。

您必须先创建激活密钥，然后才能使用引导脚本或命令行方法。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。



请不要将 Uyuni 服务器注册到其自身。必须单独管理 Uyuni 服务器，或者使用另一个独立的 Uyuni 服务器来管理它。有关使用多个服务器的详细信息，请参见 [Specialized-guides > Large-deployments](#)。

### 4.1.1. 使用 Web UI 注册客户端

使用 Web UI 引导客户端的过程使用 [Specialized-guides > Salt](#) 在客户端上执行引导过程。Salt SSH 使用 Salt 捆绑包及其包含的 Python 解释器。因此，不需要在客户端上安装其他 Python 解释器。



由于 Salt 捆绑包通过引导储存库提供，因此在启动客户端的引导过程前，必须创建该储存库。外壳脚本会使用与引导脚本相同的逻辑检测客户端上的操作系统，并部署来自适当引导储存库的 Salt 捆绑包。有关详细信息，请参见 [准备创建引导储存库](#)。



请不要将 Uyuni 服务器注册到其自身。必须单独管理 Uyuni 服务器，或者使用另一个独立的 Uyuni 服务器来管理它。有关使用多个服务器的详细信息，请参见 [Specialized-guides > Large-deployments](#)。

过程：使用 Web UI 注册客户端

1. 在 Uyuni Web UI 中，导航到 **系统 > 引导**。
2. 在 **引导客户端** 字段中，键入要引导的客户端的完全限定域名 (FQDN)。
3. 在 **SSH 端口号** 字段中，键入用于连接和引导客户端的 SSH 端口号。SSH 端口默认为 22。
4. 在 **用户名** 字段中，键入用于登录客户端的用户名。用户名默认为 `root`。
5. 要使用 SSH 引导客户端，请在 **激活密钥** 字段中选中 **SSH**，并上载用于登录客户端的 SSH 私用密钥。如果您的 SSH 私用密钥需要通行口令，请在 **SSH 口令** 字段中键入通行口令。如果不需，则将该字段留空。
6. 要使用口令引导客户端，请在 **口令** 字段中选中 **口令**，并键入用于登录客户端的口令。
7. 在 **激活密钥** 字段中，选择与您要用于引导客户端的软件通道关联的激活密钥。有关详细信息，请参见 [Client-configuration > Activation-keys](#)。
8. 可选：在 **代理** 字段中，选择要将客户端注册到的代理。
9. 默认会选中 **自动接受 SSH 主机密钥** 复选框。如此可让引导过程自动接受 SSH 主机密钥，而无需您手动进行身份验证。
10. 可选：选中 **配置为使用 SSH 连接** 复选框。如果您选中此选项，客户端将会配置为使用 SSH 来连接服务器，且不再配置其他连接方法。
11. 单击 **[Bootstrap]** 开始注册。

引导过程完成时，您的客户端即会列在 **系统 > 系统列表** 中。



SSH 私用密钥仅在引导过程期间存储，引导完成后，将会立即从 Uyuni 服务器删除。



使用 Uyuni 在客户端上安装新的软件包或更新时，会自动接受所有最终用户许可协议 (EULA)。要查看软件包 EULA，请打开 Web UI 中的软件包细节页面。

### 4.1.1.1. 处理本地指派的储存库

直接将储存库指派给客户端而不通过 Uyuni 提供这种使用情形不常见。这样可能会导致出现问题。因此通过 Salt 引导会在引导过程开始时禁用所有本地储存库。

以后在每次使用通道状态时（例如执行 Highstate 或软件包安装时），所有本地指派的储存库会再次禁用。

客户端上使用的所有软件包均应来自 Uyuni 提供的通道。有关如何创建自定义通道的详细信息，请参见 [Administration > Custom-channels](#) 中的 **自定义通道**。

## 4.1.2. 使用引导脚本注册客户端

使用引导脚本注册客户端可让您控制参数，并且便于您在需要时一次性注册大量客户端。

要使用引导脚本注册客户端，建议您先创建一个模板引导脚本，之后可以复制和修改该脚本。注册客户端时，您创建的引导脚本会在客户端上执行，并会确保所有必要的软件包都部署到该客户端。引导脚本中的某些参数确保能够使用激活密钥和 GPG 密钥将客户端系统指派到它的基础通道。

请务必仔细检查储存库信息，以确保其与基础通道储存库匹配。如果储存库信息未完全匹配，引导脚本将无法下载正确的软件包。



所有客户端都需要引导储存库。同步产品时，系统会自动在 Uyuni 服务器上创建并注册该储存库。引导储存库包含用于在客户端上安装 Salt 的软件包，以及用于注册客户端的软件包。

有关创建引导储存库的详细信息，请参见 [Client-configuration > Bootstrap-repository](#)。



GPG 密钥和 Uyuni 客户端工具

Uyuni 客户端工具使用的 GPG 密钥默认不受信任。创建引导脚本时，请使用 `ORG_GPG_KEY` 参数添加包含公共密钥指纹的文件的路径。



openSUSE Leap 15 和 SLE 15 默认使用 Python 3。必须为 openSUSE Leap 15 和 SLE 15 系统创建基于 Python 2 的引导脚本。如果您使用 Python 2 注册 Leap 15 或 SLE 15 系统，引导脚本将会失败。

### 4.1.2.1. 使用 `mgr-bootstrap` 创建引导脚本

`mgr-bootstrap` 命令会生成自定义引导脚本。Uyuni 客户端系统可使用引导脚本简化其初始注册和配置流程。

`--activation-keys` 和 `--script` 参数是仅有的必需参数。请在 Uyuni 服务器上的命令行中以 root 身份执行以下命令，并提供必需的参数。请将 `<ACTIVATION_KEY>` 和 `<EDITED_NAME>` 替换为相应值：

```
mgr-bootstrap --activation-keys=<ACTIVATION_KEY> --script=bootstrap
-<EDITED_NAME>.sh
```

`mgr-bootstrap` 命令提供了几个其他选项设置特定主机名、特定 GPG 密钥和注册方法（salt-minion 或 salt-bundle）的功能。

有关详细信息，请参见 `mgr-bootstrap` 手册页或运行 `mgr-bootstrap --help`。

### 4.1.2.2. 从 Web UI 创建引导脚本

您可以使用 Uyuni Web UI 创建可编辑的引导脚本。

过程：创建引导脚本

1. 在 Uyuni Web UI 中，导航到 **管理 > 管理器配置 > 引导脚本**。
2. 必填字段中会预填充从之前的安装步骤获得的值。有关每个设置的细节，请参见 **Reference > Admin**。
3. 单击 **[ 新 ]** 创建脚本。
4. 服务器上的 `/srv/www/htdocs/pub/bootstrap` 目录中即会生成并存储引导脚本。或者，您也可以通过 HTTPS 访问引导脚本。请将 `<example.com>` 替换为您的 Uyuni 服务器的主机名：

```
https://<example.com>/pub/bootstrap/bootstrap.sh
```



请勿在引导脚本中禁用 SSL。请在 Web UI 中确保 `--> SSL` 处于选中状态，或者引导脚本中包含 `USING_SSL=1` 设置。如果您禁用 SSL，将需要在注册过程中提供自定义 SSL 证书。

有关自定义证书的详细信息，请参见 **Administration > Ssl-certs**。

### 4.1.2.3. 编辑引导脚本

您可以复制和修改所创建的模板引导脚本，以对其进行自定义。要将引导脚本用于 Uyuni，对其进行修改时至少需包含激活密钥。大多数软件包都是使用 GPG 签名的，因此系统上还需要有可信的 GPG 密钥才能安装这些软件包。

在执行此过程时，您需要知道激活密钥的确切名称。导航到 **首页 > 概览**，然后在 **密钥** 框中，单击 **密钥**。此页面上会列出为通道创建的所有密钥。您在引导脚本中输入的要使用的密钥完整名称必须与密钥字段中显示的名称完全相同。有关激活密钥的详细信息，请参见 **Client-configuration > Activation-keys**。

过程：修改引导脚本

1. 在 Uyuni 服务器上的命令行中，以 root 身份运行以下命令切换到引导目录：

```
cd /srv/www/htdocs/pub/bootstrap/
```

2. 创建并重命名用于每个客户端的两个模板引导脚本副本。

```
cp bootstrap.sh bootstrap-sles12.sh
cp bootstrap.sh bootstrap-sles15.sh
```

- 打开 `bootstrap-sles15.sh` 进行修改。向下滚动，直到看到如下所示的文本。如果文件中包含 `exit 1`，请在该行的开头键入井号 (#) 将其注释掉。这样会激活脚本。在 `ACTIVATION_KEYS=` 字段中，输入此脚本的密钥的名称：

```
echo "Enable this script: comment (with #'s) this block (or, at
least just"
echo "the exit below)"
echo
#exit 1

# can be edited, but probably correct (unless created during initial
install):
# NOTE: ACTIVATION_KEYS *must* be used to bootstrap a client
machine.
ACTIVATION_KEYS=1-sles15
ORG_GPG_KEY=
```

- 完成后，保存该文件，然后对第二个引导脚本重复此过程。



默认情况下，如果引导储存库中提供了 `venv-salt-minion`，引导脚本会尝试安装该软件包，如果引导储存库中没有 Salt 捆绑包，则会安装 `salt-minion`。如果您出于某种原因需要使用 `salt-minion`，则可以避免安装 Salt 捆绑包，并继续使用该软件包。

有关详细信息，请参见 [Client-configuration > Contact-methods-saltbundle](#)。

#### 4.1.2.4. 运行引导脚本来注册客户端

创建好脚本后，您便可以使用它来注册客户端。

过程：运行引导脚本

- 在 Uyuni 服务器上以 root 身份登录。在命令提示符处，切换到引导目录：

```
cd /srv/www/htdocs/pub/bootstrap/
```

- 运行以下命令在客户端上执行引导脚本（将 `EXAMPLE.COM` 替换为客户端的主机名）：

```
cat bootstrap-sles15.sh | ssh root@EXAMPLE.COM /bin/bash
```

- 或者，在客户端上运行以下命令：

```
curl -Sks https://server_hostname/pub/bootstrap/bootstrap-sles15.sh
| /bin/bash
```



为避免出现问题，请确保引导脚本是使用 `bash` 执行的。

此脚本会下载位于您先前创建的储存库目录下的所需依赖项。

4. 脚本运行完成后，可以检查客户端是否已正确注册。打开 Uyuni Web UI 并导航到 **系统** > **概览**，确保列出了新客户端。如果未列出该客户端，请在 Uyuni Web UI 中导航到 **Salt** > **密钥**，检查是否接受了客户端密钥。



使用 Uyuni 在客户端上安装新的软件包或更新时，会自动接受所有最终用户许可协议 (EULA)。要查看软件包 EULA，请打开 Web UI 中的软件包细节页面。

## 4.1.3. 在命令行上注册客户端

### 4.1.3.1. 手动客户端注册

在大多数情况下，使用默认的引导方法即可正确注册 Salt 客户端。不过，您也可以在客户端上编辑 Salt 受控端文件，并提供服务器的完全限定域名 (FQDN)，使用 Salt 手动将客户端注册到 Uyuni 服务器。此方法使用传入服务器的端口 4505 和 4506 来进行。除了需确保这些端口处于打开状态之外，此方法无需在 Uyuni 服务器上进行任何配置。

要执行此过程，您需要在注册前于 Salt 客户端上安装 `venv-salt-minion` (Salt 捆绑包) 或 `salt-minion` 软件包。两种安装会使用位于不同位置但文件名相同的配置文件。`systemd` 服务文件名不同。



只有在您使用属于客户端工具通道或正式 SUSE 发行套件一部分的 `salt-minion` 时，才能以这种方式引导。

### 4.1.3.2. Salt 捆绑包配置

#### Salt 捆绑包 (`venv-salt-minion`)

- `/etc/venv-salt-minion/`
- `/etc/venv-salt-minion/minion`
- `/etc/venv-salt-minion/minion.d/NAME.conf`
- `systemd` 服务文件：`venv-salt-minion.service`

有关 Salt 捆绑包的详细信息，请参见 [Client-configuration > Contact-methods-saltbundle](#)。

过程：使用 Salt 捆绑包配置文件注册客户端

1. 在 Salt 客户端上，打开 `minion` 配置文件。配置文件位于以下位置：

```
/etc/venv-salt-minion/minion
```

或：

```
/etc/venv-salt-minion/minion.d/NAME.conf
```

- 在文件中添加或编辑 Uyuni 服务器或代理的 FQDN 以及激活密钥（如果有）。另外请添加以下其他配置参数。

```
master: SERVER.EXAMPLE.COM

grains:
  susemanager:
    activation_key: "<Activation_Key_Name>"

  server_id_use_crc: adler32
  enable_legacy_startup_events: False
  enable_fqdns_grains: False
```

- 重启动 venv-salt-minion 服务：

```
systemctl restart venv-salt-minion
```

- 在 Uyuni 服务器上，接受新客户端密钥（将 <client> 替换为您客户端的名称）：

```
salt-key -a '<客户端>'
```

### 4.1.3.3. 客户端配置

#### 客户端 (salt-minion)

- /etc/salt/
- /etc/salt/minion
- /etc/salt/minion.d/NAME.conf
- systemd 服务文件：salt-minion.service

过程：使用 Salt 受控端配置文件注册客户端

- 在 Salt 客户端上，打开 minion 配置文件。配置文件位于以下位置：

```
/etc/salt/minion
```

或：

```
/etc/salt/minion.d/NAME.conf
```

- 在文件中添加或编辑 Uyuni 服务器或代理的 FQDN 以及激活密钥（如果有）。另请添加下列其他配置参数。

```
master: SERVER.EXAMPLE.COM

grains:
  susemanager:
    activation_key: "<Activation_Key_Name>"

  server_id_use_crc: adler32
  enable_legacy_startup_events: False
  enable_fqdns_grains: False
```

- 重启动 salt-minion 服务：

```
systemctl restart salt-minion
```

- 在 Uyuni 服务器上，接受新客户端密钥（将 <client> 替换为您客户端的名称）：

```
salt-key -a '<客户端>'
```

有关 Salt 受控端配置文件的详细信息，请参见 <https://docs.saltstack.com/en/latest/ref/configuration/minion.html>。

## 4.2. SUSE 客户端注册

可以将 SUSE Linux Enterprise 客户端注册到 Uyuni 服务器。

方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 **Client-configuration** > **Activation-keys**。



请不要将 Uyuni 服务器注册到其自身。必须单独管理 Uyuni 服务器，或者使用另一个独立的 Uyuni 服务器来管理它。有关使用多个服务器的详细信息，请参见 [Specialized-guides > Large-deployments](#)。

## 4.2.1. 注册 SUSE Linux Enterprise 客户端

本节包含有关注册运行 SUSE Linux Enterprise 操作系统的客户端的信息。

请按照本章中的说明准备所有 SUSE Linux Enterprise 产品，包括：

- SUSE Linux Enterprise Server for SAP
- SUSE Linux Enterprise Desktop
- SUSE Linux Enterprise
- SUSE Linux Enterprise Real Time

您也可以按照这些说明准备较旧的 SUSE Linux Enterprise 版本和服务包。

### 4.2.1.1. 添加软件通道



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

将 SUSE Linux Enterprise 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

此过程所需的产品包括：

表格 19. SLE 产品 - WebUI

操作系统版本	产品名称
SUSE Linux Enterprise Server 15 SP6	SUSE Linux Enterprise Server 15 SP6 x86_64
SUSE Linux Enterprise Server 15 SP5	SUSE Linux Enterprise Server 15 SP5 x86_64
SUSE Linux Enterprise Server 15 SP4	SUSE Linux Enterprise Server 15 SP4 x86_64
SUSE Linux Enterprise Server 15 SP3	SUSE Linux Enterprise Server 15 SP3 x86_64
SUSE Linux Enterprise Server 15 SP2	SUSE Linux Enterprise Server 15 SP2 x86_64
SUSE Linux Enterprise Server 15 SP1	SUSE Linux Enterprise Server 15 SP1 x86_64
SUSE Linux Enterprise Server 12 SP5	SUSE Linux Enterprise Server 12 SP5 x86_64

过程：添加软件通道

1. 在 Uyuni Web UI 中，导航到 **管理 > 安装向导 > 产品**。
2. 使用搜索栏找到适用于您的客户端操作系统和体系结构的产品，然后选中相应产品。这样会自动选中所有必需的通道。此外，建议的所有通道也将选中，并且  开关会打开。单击箭头以查看相关产品的完整列表，确保您需要的所有额外产品都已选中。

3. 单击  并等待产品完成同步。

或者，您也可以在命令提示符处添加通道。此过程所需的通道包括：

表格 20. SLE 产品 - CLI

操作系统版本	基础通道
SUSE Linux Enterprise Server 15 SP6	sle-product-sles15-sp6-pool-x86_64
SUSE Linux Enterprise Server 15 SP5	sle-product-sles15-sp5-pool-x86_64
SUSE Linux Enterprise Server 15 SP4	sle-product-sles15-sp4-pool-x86_64
SUSE Linux Enterprise Server 15 SP3	sle-product-sles15-sp3-pool-x86_64
SUSE Linux Enterprise Server 15 SP2	sle-product-sles15-sp2-pool-x86_64
SUSE Linux Enterprise Server 15 SP1	sle-product-sles15-sp1-pool-x86_64
SUSE Linux Enterprise Server 12 SP5	sle-product-sles15-sp5-pool-x86_64

要查找较旧产品的通道名称，请在 Uyuni 服务器上的命令提示符处以 root 身份使用 `mgr-sync` 命令：

```
mgr-sync list --help
```

然后指定您感兴趣的参数，例如 `channels`：

```
mgr-sync list channels [-c]
```

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `mgr-sync` 命令添加相应的通道：

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同步会自动启动。如果您要手动同步通道，请使用以下命令：

```
mgr-sync sync --with-children <channel_name>
```

3. 确保同步已完成，然后再继续操作。

要添加客户端工具，请在命令提示符处添加以下通道：

表格 21. SUSE Linux Enterprise 通道 - CLI

操作系统版本	客户端通道
SUSE Linux Enterprise Server 15 SP6	sles15-sp6-uyuni-client
SUSE Linux Enterprise Server 15 SP5	sles15-sp5-uyuni-client
SUSE Linux Enterprise Server 15 SP4	sles15-sp4-uyuni-client
SUSE Linux Enterprise Server 15 SP3	sles15-sp3-uyuni-client
SUSE Linux Enterprise Server 15 SP2	sles15-sp2-uyuni-client
SUSE Linux Enterprise Server 15 SP1	sles15-sp1-uyuni-client
SUSE Linux Enterprise Server 12 SP5	sles12-sp5-uyuni-client

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。

### 4.2.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到**软件** > **管理** > **通道**，然后单击与储存库关联的通道。
2. 导航到**同步**选项卡，然后单击**同步**并选中**开始同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



SUSE Linux Enterprise 通道可能会非常大。同步所需的时间可能会长达数小时。

### 4.2.1.3. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。



为 SUSE Linux Enterprise Server 15 和 SUSE Linux Enterprise Server 12 客户端使用相同的 GPG 密钥。正确的密钥名为 `sle12-gpg-pubkey-39db7c82.key`。

### 4.2.1.4. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.2.2. 注册 SLE Micro 客户端

本节包含有关注册运行 SLE Micro 操作系统 5.1、5.2、5.3、5.4 和 5.5（x86-64、arm64 和 IBM Z (s390x) 体系结构）的客户端的信息。

SLE Micro 是一个超级可靠的轻量级操作系统，专为边缘计算而构建。它利用了 SUSE Linux Enterprise 的企业强化安全性和合规组件，并将它们整合到一个对开发人员友好的现代化、不可变操作系统平台。

SLE Micro 使用事务更新。

事务更新是原子更新（仅当所有更新都成功时，才应用所有更新）且支持回滚。它们不会影响正在运行的系统，因为在系统重引导之前，任何更改都不会生效。此信息显示在[系统 > 细节 > 概览](#)子选项卡中。

有关事务更新和重引导的详细信息，请访问 <https://documentation.suse.com/sles/html/SLES-all/cha-transactional-updates.html>。

### 4.2.2.1. 添加软件通道

将 SLE Micro 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的产品包括：

表格 22. SLE Micro 产品 - WebUI

操作系统版本	产品名称
SLE Micro 5.5 x86-64	SUSE Linux Enterprise Micro 5.5 x86_64
SLE Micro 5.5 arm64	SUSE Linux Enterprise Micro 5.5 aarch64
SLE Micro 5.5 s390x	SUSE Linux Enterprise Micro 5.5 s390x
SLE Micro 5.4 x86-64	SUSE Linux Enterprise Micro 5.4 x86_64
SLE Micro 5.4 arm64	SUSE Linux Enterprise Micro 5.4 aarch64
SLE Micro 5.4 s390x	SUSE Linux Enterprise Micro 5.4 s390x
SLE Micro 5.3 x86-64	SUSE Linux Enterprise Micro 5.3 x86_64
SLE Micro 5.3 arm64	SUSE Linux Enterprise Micro 5.3 aarch64
SLE Micro 5.3 s390x	SUSE Linux Enterprise Micro 5.3 s390x
SLE Micro 5.2 x86-64	SUSE Linux Enterprise Micro 5.2 x86_64
SLE Micro 5.2 arm64	SUSE Linux Enterprise Micro 5.2 aarch64
SLE Micro 5.2 s390x	SUSE Linux Enterprise Micro 5.2 s390x
SLE Micro 5.1 x86-64	SUSE Linux Enterprise Micro 5.1 x86_64
SLE Micro 5.1 arm64	SUSE Linux Enterprise Micro 5.1 aarch64
SLE Micro 5.1 s390x	SUSE Linux Enterprise Micro 5.1 s390x

过程：添加软件通道

- 在 Uyuni Web UI 中，导航到管理，安装向导，产品。
- 使用搜索栏找到适用于您的客户端操作系统和体系结构的产品，然后选中相应产品。这样会自动选中所有必需的通道。此外，建议的所有通道也将选中，并且  开关会打开。单击箭头以查看相关产品的完整列表，确保您需要的所有额外产品都已选中。
- 单击 并等待产品完成同步。

或者，您也可以在命令提示符处添加通道。此过程所需的通道包括：

表格 23. SLE Micro 产品 - CLI

操作系统版本	基础通道	更新通道
SLE Micro 5.5 x86-64	sle-micro-5.5-pool-x86_64	sle-micro-5.5-updates-x86_64
SLE Micro 5.5 arm64	sle-micro-5.5-pool-arm64	sle-micro-5.5-updates-arm64
SLE Micro 5.5 IBM Z (s390x)	sle-micro-5.5-pool-s390x	sle-micro-5.5-updates-s390x
SLE Micro 5.4 x86-64	sle-micro-5.4-pool-x86_64	sle-micro-5.4-updates-x86_64

操作系统版本	基础通道	更新通道
SLE Micro 5.4 arm64	sle-micro-5.4-pool-arm64	sle-micro-5.4-updates-arm64
SLE Micro 5.4 IBM Z (s390x)	sle-micro-5.4-pool-s390x	sle-micro-5.4-updates-s390x
SLE Micro 5.3 x86-64	sle-micro-5.3-pool-x86_64	sle-micro-5.3-updates-x86_64
SLE Micro 5.3 arm64	sle-micro-5.3-pool-arm64	sle-micro-5.3-updates-arm64
SLE Micro 5.3 IBM Z (s390x)	sle-micro-5.3-pool-s390x	sle-micro-5.3-updates-s390x
SLE Micro 5.2 x86-64	suse-microos-5.2-pool-x86_64	suse-microos-5.2-updates-x86_64
SLE Micro 5.2 arm64	suse-microos-5.2-pool-aarch64	suse-microos-5.2-updates-aarch64
SLE Micro 5.2 IBM Z (s390x)	suse-microos-5.2-pool-s390x	suse-microos-5.2-updates-s390x
SLE Micro 5.1 x86-64	suse-microos-5.1-pool-x86_64	suse-microos-5.1-updates-x86_64
SLE Micro 5.1 arm64	suse-microos-5.1-pool-aarch64	suse-microos-5.1-updates-aarch64
SLE Micro 5.1 IBM Z (s390x)	suse-microos-5.1-pool-s390x	suse-microos-5.1-updates-s390x

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `mgr-sync` 命令添加相应的通道：

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同步会自动启动。如果您要手动同步通道，请使用以下命令：

```
mgr-sync sync --with-children <channel_name>
```

3. 确保同步已完成，然后再继续操作。

要添加客户端工具，请在命令提示符处添加以下通道：

表格 24. SLE Micro 通道 - CLI

操作系统版本	客户端通道
SLE Micro 5.4	sle-micro-5.4-uyuni-client
SLE Micro 5.3	suse-micro-5.3-uyuni-client
SLE Micro 5.2	suse-microos-5.2-uyuni-client
SLE Micro 5.1	sle-microos-5.1-uyuni-client

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。

### 4.2.2.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到**软件**，**管理**，**通道**，然后单击与储存库关联的通道。
2. 导航到**同步**选项卡，然后单击**同步**并选中**立即同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.2.2.3. 注册客户端

SLE Micro 客户端在注册后需要重引导。



完成引导过程后，将在 SLE Micro 客户端上禁用自动引导。实施此项更改的原因在于间歇性自动重引导，在 Salt 中继引导 Salt 状态的应用结果之前可能会发生这种重引导情况。

尽管系统在完成注册后会自动安排重引导，但会采用默认的重引导管理器维护时段。此时段在客户端注册后可能会持续几个小时。建议在注册脚本完成后手动重引导客户端，以加速注册过程并使系统显示在系统列表中。

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

对 SLE Micro 系统使用引导脚本时，请确保脚本的 certificate 部分包含以下内容：

```
ORG_CA_CERT=RHN-ORG-TRUSTED-SSL-CERT
ORG_CA_CERT_IS_RPM_YN=0
```

直接编辑引导脚本并添加该设置，或使用以下参数创建引导脚本：

```
mgr-bootstrap --script=bootstrap-sle-micro.sh \
--ssl-cert=/srv/www/htdocs/pub/RHN-ORG-TRUSTED-SSL-CERT
```

#### 4.2.2.4. 重引导 SLE Micro

SLE Micro 是一种事务系统。事务更新一般支持数种重引导方法。建议在 Uyuni 所管理的系统中使用 `systemd` 来进行重引导。使用其他方法可能会导致系统出现意外行为。

如果 Uyuni 上的事务系统使用的是默认配置，引导该系统时，`systemd` 将会配置为引导方法 (`REBOOT_METHOD`)。该默认配置允许 Uyuni 控制重引导操作，重引导可以立即执行或使用 Uyuni 根据需要安排重引导时间。

##### 4.2.2.4.1. 背景信息

默认情况下，在安装客户端的过程中会将重引导方法设置为 `auto`。采用 `auto` 引导方法时，如果 `rebootmgrd` 正在运行，将使用该服务根据配置的策略来重引导系统。策略可能设置为立即重引导或在维护时段重引导。有关详细信息，请参见 `rebootmgrd(8)` 手册页。如果 `rebootmgrd` 没有运行，Uyuni 将会调用 `systemctl reboot`。



使用 `systemd` 之外的任何其他方法都可能会导致系统出现意外行为。

#### 4.2.3. 注册 SL Micro 客户端

本节包含有关注册运行 SL Micro 操作系统 6.0 x86-64、arm64 和 IBM Z (s390x) 的客户端的信息。

SL Micro 是一个超级可靠的轻量级操作系统，专为边缘计算而构建。它利用了 SUSE Linux Enterprise 的企业强化安全性和合规组件，并将它们整合到一个对开发人员友好的现代化、不可变操作系统平台。

SL Micro 使用事务更新。事务更新是原子更新（仅当所有更新都成功时，才应用所有更新）且支持回滚。它们

不会影响正在运行的系统，因为在系统重引导之前，任何更改都不会生效。此信息显示在系统 > 细节 > 概览子选项卡中。

有关事务更新和重引导的详细信息，请访问 <https://documentation.suse.com/sles/html/SLES-all/cha-transactional-updates.html>。

### 4.2.3.1. 添加软件通道

将 SL Micro 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的产品包括：

表格 25. SL Micro 产品 - WebUI

操作系统版本	产品名称
SL Micro 6.0 x86-64	SUSE Linux Micro 6.0 x86_64
SL Micro 6.0 arm64	SUSE Linux Micro 6.0 arch64
SL Micro 6.0 s390x	SUSE Linux Micro 6.0 s390x

过程：添加软件通道

1. 在 Uyuni Web UI 中，导航到管理，安装向导，产品。
2. 使用搜索栏找到适用于您的客户端操作系统和体系结构的产品，然后选中相应产品。这样会自动选中所有必需的通道。此外，建议的所有通道也将选中，并且  开关会打开。单击箭头以查看相关产品的完整列表，确保您需要的所有额外产品都已选中。
3. 单击 **[添加产品]** 并等待产品完成同步。

或者，您也可以在命令提示符处添加通道。此过程所需的通道包括：

表格 26. SL Micro 产品 - CLI

操作系统版本	基础通道
SL Micro 6.0 x86-64	sl-micro-6.0-pool-x86_64

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `mgr-sync` 命令添加相应的通道：

```

mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>

```

2. 同步会自动启动。如果您要手动同步通道，请使用以下命令：

```
mgr-sync sync --with-children <channel_name>
```

3. 确保同步已完成，然后再继续操作。

### 4.2.3.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到软件，管理，通道，然后单击与储存库关联的通道。
2. 导航到同步选项卡，然后单击同步并选中同步。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.2.3.3. 注册客户端



SL Micro 客户端在注册后需要进行重引导。尽管系统在完成注册后会自动安排重引导，但会采用默认的重引导管理器维护时段。此时段在客户端注册后可能会持续几个小时。建议在注册脚本完成后手动重引导客户端，以加速注册过程并使系统显示在系统列表中。

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

对 SL Micro 系统使用引导脚本时，请确保脚本的 certificate 部分包含以下内容：

```
ORG_CA_CERT=RHN-ORG-TRUSTED-SSL-CERT
ORG_CA_CERT_IS_RPM_YN=0
```

直接编辑引导脚本并添加该设置，或使用以下参数创建引导脚本：

```
mgr-bootstrap --script=bootstrap-sl-micro.sh \
--ssl-cert=/srv/www/htdocs/pub/RHN-ORG-TRUSTED-SSL-CERT
```

#### 4.2.3.4. 重引导 SL Micro

SL Micro 是一种事务系统。事务更新一般支持数种重引导方法。建议在 Uyuni 所管理的系统中使用 `systemd` 来进行重引导。使用其他方法可能会导致系统出现意外行为。

如果 Uyuni 上的事务系统使用的是默认配置，引导该系统时，`systemd` 将会配置为引导方法 (`REBOOT_METHOD`)。该默认配置允许 Uyuni 控制重引导操作，重引导可以立即执行或使用 Uyuni 根据需要安排重引导时间。

##### 4.2.3.4.1. 背景信息

默认情况下，在安装客户端的过程中会将重引导方法设置为 `auto`。采用 `auto` 引导方法时，如果 `rebootmgrd` 正在运行，将使用该服务根据配置的策略来重引导系统。策略可能设置为立即重引导或在维护时段重引导。有关详细信息，请参见 `rebootmgrd(8)` 手册页。如果 `rebootmgrd` 没有运行，Uyuni 将会调用 `systemctl reboot`。



使用 `systemd` 之外的任何其他方法都可能会导致系统出现意外行为。

### 4.3. openSUSE 客户端注册

可以将 openSUSE 和 openSUSE Leap Micro 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。



请不要将 Uyuni 服务器注册到其自身。必须单独管理 Uyuni 服务器，或者使用另一个独立的 Uyuni 服务器来管理它。有关使用多个服务器的详细信息，请参见 [Specialized-guides > Large-deployments](#)。

#### 4.3.1. 注册 openSUSE Leap 客户端

本节包含有关注册运行 openSUSE 操作系统的客户端的信息。Uyuni 支持使用 Salt 的 openSUSE Leap 15 客户端。

支持使用引导功能启动 openSUSE 客户端并执行初始状态运行，例如设置储存库和执行配置文件更新。

##### 4.3.1.1. 添加软件通道

将 openSUSE 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

当前支持的系统结构为 `x86_64` 和 `aarch64`。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



- 下面的小节中的说明通常默认使用 `x86_64` 体系结构。请根据情况将其替换为其他体系结构。

例如，使用 `x86_64` 体系结构时，您需要如下产品：

表格 27. openSUSE 通道 - CLI

操作系统版本	openSUSE Leap 15.6	openSUSE Leap 15.5	openSUSE Leap 15.4
基础通道	<code>opensuse_leap15_6</code>	<code>opensuse_leap15_5</code>	<code>opensuse_leap15_4</code>
客户端通道	<code>opensuse_leap15_6-uyuni-client</code>	<code>opensuse_leap15_5-uyuni-client</code>	<code>opensuse_leap15_4-uyuni-client</code>
更新通道	<code>opensuse_leap15_6-updates</code>	<code>opensuse_leap15_5-updates</code>	<code>opensuse_leap15_4-updates</code>
非 OSS 通道	<code>opensuse_leap15_6-non-oss</code>	<code>opensuse_leap15_5-non-oss</code>	<code>opensuse_leap15_4-non-oss</code>
非 OSS 更新通道	<code>opensuse_leap15_6-non-oss-updates</code>	<code>opensuse_leap15_5-non-oss-updates</code>	<code>opensuse_leap15_4-non-oss-updates</code>
向后移植更新通道	<code>opensuse_leap15_6-backports-updates</code>	<code>opensuse_leap15_5-backports-updates</code>	<code>opensuse_leap15_4-backports-updates</code>
SLE 更新通道	<code>opensuse_leap15_6-sle-updates</code>	<code>opensuse_leap15_5-sle-updates</code>	<code>opensuse_leap15_4-sle-updates</code>

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。

### 4.3.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到软件，管理，通道，然后单击与储存库关联的通道。
2. 导航到——选项卡，然后单击——并选中——。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



openSUSE 通道可能会非常大。同步所需的时间可能会长达数小时。

### 4.3.1.3. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

### 4.3.1.4. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.3.2. 注册 openSUSE Leap Micro 客户端

本节包含有关注册运行 x86-64 和 aarch64 体系结构 openSUSE Leap Micro 操作系统的客户端的信息。

openSUSE Leap Micro 是一个超级可靠的轻量级操作系统，专为边缘计算而构建。它利用了 SUSE Linux Enterprise 的强化安全性和合规组件，并将它们整合到一个对开发人员友好的现代化、不可变操作系统平台。

openSUSE Leap Micro 使用事务更新。事务更新是原子更新（仅当所有更新都成功时，才应用所有更新）且支

持回滚。它们不会影响正在运行的系统，因为在系统重引导之前，它们不会激活任何更改。此信息显示在**系统**，**细节**，**概览**子选项卡中。

有关事务更新和重引导的详细信息，请访问 <https://documentation.suse.com/sles/html/SLES-all/cha-transactional-updates.html>。

表格 28. openSUSE 通道 - CLI

操作系统版本	openSUSE Leap Micro 5.5	openSUSE Leap Micro 5.4	openSUSE Leap Micro 5.3
基础通道	opensuse_micro5_5	opensuse_micro5_4	opensuse_micro5_3
客户端通道	opensuse_micro5_5-uyuni-client	opensuse_micro5_4-uyuni-client	opensuse_micro5_3-uyuni-client
SLE 更新通道	opensuse_micro5_5-sle-updates	opensuse_micro5_4-sle-updates	opensuse_micro5_3-sle-updates

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。

### 4.3.2.1. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到**软件**，**管理**，**通道**，然后单击与储存库关联的通道。
2. 导航到**同步**选项卡，然后单击**同步**并选中**同步进度**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



openSUSE Leap Micro 通道可能非常大。同步所需的时间可能长达数小时。

### 4.3.2.2. 注册客户端



openSUSE Leap Micro 客户端在注册后需要重引导。

完成引导过程后，将在 openSUSE Leap Micro 上禁用自动引导。实施此项更改的原因在于间歇性自动重引导，在 Salt 中继引导 Salt 状态的应用结果之前可能会发生这种重引导情况。

系统在完成注册后会自动安排重引导，但会遵循默认的重引导管理器维护时段。此时段在客户端注册后可能会持续几个小时。要加速 openSUSE Leap Micro 注册过程，请在注册脚本完成后手动重引导客户端。

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.4. Alibaba Cloud Linux 客户端注册

您可以将 Alibaba Cloud Linux 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.4.1. 注册 Alibaba Cloud Linux 客户端

本节包含有关注册运行 Alibaba Cloud Linux 操作系统的客户端的信息。



一些 Alibaba Cloud Linux 2 实例需要尝试两次才能成功注册。

#### 4.4.1.1. 添加软件通道

将 Alibaba Cloud Linux 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 29. Alibaba Cloud Linux 通道 - CLI

操作系统版本	核心通道	更新通道	客户端通道
Alibaba Cloud Linux 2	alibaba-2	alibaba-2-updates	alibaba-2-uyuni-client

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。



`spacewalk-common-channels` 提供的客户端工具通道来自 Uyuni 而非 SUSE。

#### 4.4.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件**、**管理**、**通道**，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **所有通道**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.4.1.3. 创建激活密钥

您需要创建与您的 Alibaba Cloud Linux 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.4.1.4. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

某些 Alibaba Cloud Linux 2 实例在首次尝试注册时会失败。这由 Alibaba Cloud Linux 2 映像中的一个已知 Bug 所导致。

`python-urlgrabber3` 软件包以 Python pip 软件包和 RPM 软件包两种形式提供，这可能会导致在首次尝试注册时发生冲突。

如果您的实例基于其中一个受影响的映像版本，客户端在第二次尝试注册时应该会正确注册。

## 4.5. AlmaLinux 客户端注册

您可以将 AlmaLinux 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您必须已创建激活密钥。

- 有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。
- 有关将 AlmaLinux 迁移到 SUSE Liberty Linux 的详细信息，请参见 [client-configuration:clients-sleses.pdf](#)。

### 4.5.1. 注册 AlmaLinux 客户端

本节包含有关注册运行 AlmaLinux 操作系统的客户端的信息。



如果 AlmaLinux 实例是在 AWS 中创建的，则 `/etc/machine-id` 中的 `machine-id` ID 一律相同。请务必在创建实例后重新生成 `machine-id`。有关详细信息，请参见 [Administration > Troubleshooting](#)。

### 4.5.1.1. 添加软件通道



我们已使用----策略并采用默认的 enforcing SELinux 配置对将 AlmaLinux 客户端注册到 Uyuni 的过程进行了测试。将 AlmaLinux 客户端注册到 Uyuni 时，您无需禁用 SELinux。

将 AlmaLinux 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

目前支持的系统结构有 x86\_64 和 aarch64，在版本 9 上，另外还支持 ppc64le 和 s390x。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 30. AlmaLinux 通道 - CLI

操作系统版本	基础通道	客户端通道	AppStream 通道
AlmaLinux 9	almalinux9	almalinux9-uyuni-client	almalinux9-appstream
AlmaLinux 8	almalinux8	almalinux8-uyuni-client	almalinux8-appstream

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道。请确保指定正确的体系结构：

```
spacewalk-common-channels \
-a <体系结构> \
<基础通道名称> \
<子通道名称 1> \
<子通道名称 2> \
... <子通道名称 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <base_channel_label>-<architecture>
```

3. 确保同步已完成，然后再继续操作。



`spacewalk-common-channels` 提供的客户端工具通道来自 Uyuni 而非 SUSE。



对于 AlmaLinux 9 和 AlmaLinux 8 客户端，请添加基础通道和 AppStream 通道。您需要来自这两个通道的软件包。如果未添加这两个通道，将会因缺少软件包而无法创建引导储存库。

如果您使用的是模块化通道，则必须在 AlmaLinux 8 客户端上启用 Python 3.6 模块流。如果不提供 Python 3.6，spacecmd 软件包安装将会失败。



您可能会发现 AppStream 通道中提供的软件包数量在上游通道和 Uyuni 通道之间存在一定的差异。如果您对在不同时间添加的同一通道进行比较，会发现其数量也不相同。这是由 AlmaLinux 管理其储存库的方式所致。当有新版本发布时，AlmaLinux 会去除软件包的较旧版本，而 Uyuni 则会保留所有版本，无论新旧与否。

### 4.5.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件** > **管理** > **通道**，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.5.1.3. 创建激活密钥

您需要创建与您的 AlmaLinux 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.5.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

### 4.5.1.5. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 4.5.1.6. 管理勘误

当您更新 AlmaLinux 客户端时，软件包会包含有关更新的元数据。

## 4.6. Amazon Linux 客户端注册

您可以将 Amazon Linux 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.6.1. 注册 Amazon Linux 客户端

本节包含有关注册运行 Amazon Linux 操作系统的客户端的信息。



如果 Amazon Linux 2 实例是在 AWS 中创建的，则其在 `/etc/machine-id` 中的 `machine-id` ID 始终相同。如果您要创建 Amazon Linux 2 实例，请务必在创建实例后重新生成 `machine-id`。有关详细信息，请参见 [Administration > Troubleshooting](#)。

Amazon Linux 2023 不受影响。

#### 4.6.1.1. 添加软件通道

将 Amazon Linux 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

当前支持的系统结构为 `x86_64` 和 `aarch64`。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



下面的小节中的说明通常默认使用 `x86_64` 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 31. Amazon Linux 通道 - CLI

操作系统版本	基础通道	客户端通道
Amazon Linux 2023	amazonlinux2023	amazonlinux2023-uyuni-client
Amazon Linux 2	amazonlinux2-core	amazonlinux2-uyuni-client



对于 Amazon Linux 2，如果您打算在 Amazon Linux 实例中使用 Docker，另请务必添加并同步 amazonlinux2-extra-docker 通道。



对于 Amazon Linux 2023，如果您打算在 Amazon Linux 实例中使用 Kernel Live 补丁，另请务必添加并同步 amazonlinux2023-kernel-livepatch 通道。

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 spacewalk-common-channels 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。



spacewalk-common-channels 提供的客户端工具通道来自 Uyuni 而非 SUSE。

#### 4.6.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件** > **管理** > **通道**，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **开始同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

- 每个子通道在同步过程中都会生成自己的日志。 您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.6.1.3. 创建激活密钥

您需要创建与您的 Amazon Linux 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.6.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

### 4.6.1.5. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.7. CentOS 客户端注册

您可以将 CentOS 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您必须已创建激活密钥。

- 有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。
- 有关将 CentOS 迁移到 SUSE Liberty Linux 的详细信息，请参见 [client-configuration:clients-sleses.pdf](#)。

### 4.7.1. 注册 CentOS 客户端

本节包含有关注册运行 CentOS 操作系统的客户端的信息。



您需负责安排对 CentOS 基础媒体储存库和 CentOS 安装媒体的访问权限，以及将 Uyuni 服务器连接到 CentOS 内容分发网络。



我们已使用 `--enforcing` 策略并采用默认的 SELinux 配置对将 CentOS 客户端注册到 Uyuni 的过程进行了测试。将 CentOS 客户端注册到 Uyuni 时，您无需禁用 SELinux。

### 4.7.1.1. 添加软件通道

将 CentOS 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

当前支持的系统结构为 `x86_64` 和 `aarch64`。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



下面的小节中的说明通常默认使用 `x86_64` 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 32. CentOS 通道 - CLI

操作系统版本	基础通道	客户端通道	更新/Appstream 通道
CentOS 7	centos7	centos7-uyuni-client	centos7-updates

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道。请确保指定正确的体系结构：

```
spacewalk-common-channels \
-a <体系结构> \
<基础通道名称> \
<子通道名称 1> \
<子通道名称 2> \
... <子通道名称 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <base_channel_label>-<architecture>
```

3. 确保同步已完成，然后再继续操作。



`spacewalk-common-channels` 提供的客户端工具通道来自 Uyuni 而非 SUSE。

如果您使用的是模块化通道，则必须在客户端上启用 Python 3.6 模块流。如果不提供 Python 3.6，spacecmd 软件包安装将会失败。



您可能会发现 AppStream 通道中提供的软件包数量在上游通道和 Uyuni 通道之间存在一定的差异。如果您对在不同时间添加的同一通道进行比较，会发现其数量也不相同。这是由 CentOS 管理其储存库的方式所致。当有新版本发布时，CentOS 会去除软件包的较旧版本，而 Uyuni 则会保留所有版本，无论新旧与否。

### 4.7.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件 > 管理 > 通道**，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **显示所有通道**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.7.1.3. 创建激活密钥

您需要创建与您的 CentOS 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.7.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

### 4.7.1.5. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

#### 4.7.1.6. 管理勘误



本节仅适用于非容器化 Uyuni，在 CentOS 7 生命周期结束后将被去除。

当您更新 CentOS 客户端时，软件包不包含有关更新的元数据。您可以使用第三方勘误服务来获取此信息。



CEFS 的作者会按照尽力而为的原则提供补丁或勘误，希望它们有用，但无法保证其正确性或及时性。这意味着补丁日期可能不正确，且至少在一种情况下所显示的发布数据滞后一个多月。有关这些情况的详细信息，请参见 <https://github.com/stevemeier/cefs/issues/28#issuecomment-656579382> 和 <https://github.com/stevemeier/cefs/issues/28#issuecomment-656573607>。

任何有关补丁数据的问题或滞后都可能导致将不可靠的补丁信息导入到您的 Uyuni 服务器中，进而导致报告、审计、CVE 更新或其他与补丁相关的信息也不正确。请考虑使用其他方案来替代此服务，例如独立校验补丁数据或选择其他操作系统，具体取决于您的安全相关要求和认证准则。

过程：安装勘误服务

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份添加 sle-module-development-tools 模块：

```
SUSEConnect --product sle-module-development-tools/15.2/x86_64
```

2. 安装勘误服务依赖项：

```
zypper in perl-Text-Unidecode
```

3. 在 /etc/rhn/rhn.conf 中添加或编辑下面一行：

```
java.allow_adding_patches_via_api = centos7-x86_64-updates,centos7-x86_64,centos7-x86_64-extras
```

4. 重启 Tomcat：

```
systemctl restart tomcat
```

5. 为勘误脚本创建一个文件：

```
touch /usr/local/bin/cent-errata.sh
```

6. 编辑新文件以包含此脚本，并视需要编辑储存库细节。此脚本会从外部勘误服务提取勘误细节，将其解压缩，然后发布这些细节：

```
#!/bin/bash
mkdir -p /usr/local/centos
cd /usr/local/centos
rm *.xml
wget -c http://cefs.steve-meier.de/errata.latest.xml
wget -c https://www.redhat.com/security/data/oval/v2/RHEL7/rhel-
7.oval.xml.bz2
bzip2 -d rhel-7.oval.xml.bz2
wget -c http://cefs.steve-meier.de/errata-import.tar
tar xvf errata-import.tar
chmod +x /usr/local/centos/errata-import.pl
export SPACEWALK_USER='<adminname>' ; export
SPACEWALK_PASS='<password>'
/usr/local/centos/errata-import.pl --server '<servername>' \
--errata /usr/local/centos/errata.latest.xml \
--include-channels=centos7-x86_64-updates,centos7-x86_64,centos7
-x86_64-extras \
--publish --rhsa-oval /usr/local/centos/rhel-7.oval.xml
```

7. 设置 cron 作业以每日运行该脚本：

```
ln -s /usr/local/bin/cent-errata.sh /etc/cron.daily
```

有关此工具的详细信息，请参见 <https://cefs.steve-meier.de/>。

## 4.8. Debian 客户端注册

您可以将 Debian 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

## 4.8.1. 注册 Debian 客户端

本节包含有关注册运行 Debian 操作系统的客户端的信息。

可以通过引导 Debian 客户端来执行初始状态的运行以及配置文件的更新。

### 4.8.1.1. 准备注册

您需要完成一些准备工作，然后才能将 Debian 客户端注册到 Uyuni 服务器：

- 确保 DNS 配置正确并提供客户端对应的项。或者，您也可以配置 Uyuni 服务器和客户端上的 /etc/hosts 文件，在其中添加相应的项。
- 注册前，客户端的日期和时间必须已与 Uyuni 服务器同步。

### 4.8.1.2. 添加软件通道

将 Debian 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 33. Debian 通道 - CLI

操作系统版本	基础通道	客户端通道	更新通道	安全通道
Debian 12	debian-12-pool-amd64-uyuni	debian-12-amd64-uyuni-client	debian-12-amd64-main-updates-uyuni	debian-12-amd64-main-security-uyuni
Debian 11	debian-11-pool-amd64-uyuni	debian-11-amd64-uyuni-client	debian-11-amd64-main-updates-uyuni	debian-11-amd64-main-security-uyuni

过程：在命令提示符处添加软件通道

- 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 spacewalk-common-channels 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

- 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。

### 4.8.1.3. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到软件，管理，通道，然后单击与储存库关联的通道。
2. 导航到同步选项卡，然后单击同步并选中同步。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



Debian 通道可能会非常大。同步所需的时间可能会长达数小时。

### 4.8.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。



Debian 客户端可能需要安装多个 GPG 密钥。

同步第三方 Debian 储存库时，您需要在服务器上导入相应的 GPG 密钥。如果缺少 GPG 密钥，同步将会失败。

使用 Debian 储存库时，系统只会对元数据签名。因此不需要导入软件通道的 GPG 密钥。Uyuni 将不会对软件包重新签名。

要查看已导入 Uyuni 服务器的 GPG 密钥，请运行以下命令：

```
mgrctl exec -- gpg --homedir /var/lib/spacewalk/gpgdir --list-keys
```

要导入新的 GPG 密钥，请运行以下命令：

```
mgradm gpg add <文件名>.gpg
```

#### 4.8.1.5. Root 访问权限

默认会为 Debian 上的 root 用户禁用 SSH 访问权限。

要想能够使用普通用户进行初始配置，您需要编辑 sudoers 文件。

过程：向 Root 用户授予访问权限

1. 在客户端上，编辑 sudoers 文件：

```
sudo visudo
```

在 sudoers 文件末尾添加下面一行，以向用户授予 sudo 访问权限。以在 Web UI 中引导客户端的用户的名称替换 <user>：

```
<user> ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2,  
/usr/bin/python3, /var/tmp/venv-salt-minion/bin/python
```



此过程无需口令便可授予 root 访问权限，而注册客户端需要提供口令。客户端成功安装后会以 root 特权运行，因此将不再需要该访问权限。客户端成功安装之后，建议您从 sudoers 文件中去除该行。

#### 4.8.1.6. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 4.9. OpenEuler 客户端注册

可以将 openEuler 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

## 4.9.1. 注册 openEuler 客户端

本节包含有关注册运行 openEuler 操作系统的客户端的信息。



如果 openEuler 实例是在 AWS 中创建的，则其在 `/etc/machine-id` 中的 machine-id ID 始终相同。请务必在创建实例后重新生成 machine-id。有关详细信息，请参见 [Administration > Troubleshooting](#)。

### 4.9.1.1. 添加软件通道

将 openEuler 客户端注册到 Uyuni 服务器之前，需要添加并同步所需的软件通道。

当前支持的系统结构为 `x86_64` 和 `aarch64`。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



下面的小节中的说明通常默认使用 `x86_64` 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 34. openEuler 通道 - CLI

操作系统版本	核心通道	客户端通道	更新通道	EPOL 通道	统括性通道
openEuler 22.03	openeuler2203	openeuler2203-uyuni-client	openeuler2203-update	openeuler2203-epol	openeuler2203-everything

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。



spacewalk-common-channels 提供的客户端工具通道来自 Uyuni 而非 SUSE。

### 4.9.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到软件，管理，通道，然后单击与储存库关联的通道。
2. 导航到同步选项卡，然后单击同步并选中同步。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.9.1.3. 创建激活密钥

需要创建与您的 openEuler 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.9.1.4. 在客户端上信任 GPG 密钥

### 4.9.1.5. 在客户端上信任 GPG 密钥

操作系统要么直接信任自己的 GPG 密钥，要么至少将它们与最小系统一起安装，但通过其他 GPG 密钥签名的第三方软件包需要手动处理。客户端可以在不信任 GPG 密钥的情况下成功引导，但除非密钥受信任，否则您将无法安装新的客户端工具软件包或更新软件包。

客户端现在使用为软件通道输入的 GPG 密钥信息来管理受信任密钥。如果将具有 GPG 密钥信息的软件通道指派给客户端，当通道刷新后或从此通道安装第一个软件包后，系统就会信任该密钥。

软件通道页面中的 GPG 密钥 URL 参数可以包含以空格分隔的多个密钥 URL。如果其为文件 URL，则必须在使用软件通道之前在客户端上部署 GPG 密钥文件。

基于 Red Hat 的客户端的客户端工具通道的 GPG 密钥会部署到客户端的 /etc/pki/rpm-gpg/ 中，并可通过文件 URL 引用。

仅当为客户端指派了软件通道时，系统才会导入并信任这些密钥。



由于基于 Debian 的系统仅会对元数据签名，因此不需要为各个通道指定确切的密钥。如果用户按 **Administration > Repo-metadata** 中的“使用自己的 GPG 密钥”所述配置了自己的 GPG 密钥来对元数据签名，系统会自动部署并信任该密钥。

#### 4.9.1.5.1. 用户定义的 GPG 密钥

用户可以定义要部署到客户端的自定义 GPG 密钥。

通过提供某些 pillar 数据并在 Salt 文件系统中提供 GPG 密钥文件，系统会自动将它们部署到客户端。

这些密钥会分别部署到 `/etc/pki/rpm-gpg/`（在基于 RPM 的操作系统上）和 `/usr/share/keyrings/`（在 Debian 系统上）中：

为您的要将密钥部署到的客户端定义 pillar 键 `custom_gpgkeys` 并列出密钥文件名。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

此外，在 Salt 文件系统中创建名为 `gpg` 的目录，并将 `custom_gpgkeys` pillar 数据中所指定的 GPG 密钥文件存储在该目录下。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

密钥将部署到客户端的 `/etc/pki/rpm-gpg/my_first_gpg.key` 和 `/etc/pki/rpm-gpg/my_second_gpgkey.gpg` 中。

最后一步是将 URL 添加到软件通道的“GPG 密钥 URL”字段中。请导航到 **软件 > 管理 > 通道**，然后选择要修改的通道。将值 `file:///etc/pki/rpm-gpg/my_first_gpg.key` 添加到 GPG URL 中。

#### 4.9.1.5.2. 引导脚本中的 GPG 密钥

过程：在客户端上使用引导脚本信任 GPG 密钥

1. 在 Uyuni 服务器上的命令提示符处，检查 `/srv/www/htdocs/pub/` 目录的内容。此目录包含所有可用的公共密钥。记下为您正在注册的客户端指派的通道适用的密钥。
2. 打开相关的引导脚本，找到 `ORG_GPG_KEY=` 参数并添加所需的密钥。例如：

```
uyuni-gpg-pubkey-0d20833e.key
```

您无需删除任何以前存储的密钥。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。如果不信任 GPG 密钥，便无法为客户端指派软件通道。

#### 4.9.1.6. 注册客户端

openEuler 客户端的注册方式与所有其他客户端的注册方式相同。有关详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 4.10. Oracle 客户端注册

您可以将 Oracle Linux 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您必须已创建激活密钥。

- 有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。
- 有关将 Oracle Linux 迁移到 SUSE Liberty Linux 的详细信息，请参见 [client-configuration:clients-sleses.pdf](#)。

#### 4.10.1. 注册 Oracle Linux 客户端

本节包含有关注册运行 Oracle Linux 操作系统的客户端的信息。



目前不支持直接将 Unbreakable Linux Network (ULN) 储存库与 Uyuni 同步。必须使用 ULN 的 Oracle 本地发行版。有关设置本地 ULN 镜像的详细信息，请参见 <https://docs.oracle.com/en/operating-systems/oracle-linux/software-management/sfw-mgmt-UseSoftwareDistributionMirrors.html#local-uln-mirror> 上提供的相关 Oracle 文档。

##### 4.10.1.1. 添加软件通道

将 Oracle Linux 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

当前支持的系统结构为 x86\_64 和 aarch64。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 35. Oracle 通道 - CLI

操作系统版本	基础通道	客户端通道	更新通道
Oracle Linux 9	oraclelinux9	oraclelinux9-uyuni-client	oraclelinux9-appstream
Oracle Linux 8	oraclelinux8	oraclelinux8-uyuni-client	oraclelinux8-appstream
Oracle Linux 7	oraclelinux7	oraclelinux7-uyuni-client	-

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。



spacewalk-common-channels 提供的客户端工具通道来自 Uyuni 而非 SUSE。



对于 Oracle Linux 9 和 Oracle Linux 8 客户端，请添加基础通道和 AppStream 通道。您需要来自这两个通道的软件包。如果未添加这两个通道，将会因缺少软件包而无法创建引导储存库。

如果您使用的是模块化通道，则必须在客户端上启用 Python 3.6 模块流。如果不提供 Python 3.6，`spacecmd` 软件包安装将会失败。

### 4.10.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件** > **管理** > **通道**，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **开始同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

- 每个子通道在同步过程中都会生成自己的日志。 您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

#### 4.10.1.3. 创建激活密钥

您需要创建与您的 Oracle Linux 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

#### 4.10.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。



对于 Oracle Linux 9 或 Oracle Linux 8 客户端，请使用

```
ol8-gpg-pubkey-82562EA9AD986DA3.key
```

对于 Oracle Linux 7 客户端，请使用

```
ol67-gpg-pubkey-72F97B74EC551F0A3.key
```

#### 4.10.1.5. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.11. Raspberry Pi OS 客户端注册

可以将 Raspberry Pi OS 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.11.1. 注册 Raspberry Pi OS 客户端

本节包含有关注册运行 Raspberry Pi OS 操作系统的客户端的信息。

可以通过引导 Raspberry Pi OS 客户端来执行初始状态的运行以及配置文件的更新。

#### 4.11.1.1. 准备注册

需要完成一些准备工作，然后才能将 Raspberry Pi OS 客户端注册到 Uyuni 服务器：

- 确保 DNS 配置正确并提供客户端对应的项。或者，您也可以配置 Uyuni 服务器和客户端上的 `/etc/hosts` 文件，在其中添加相应的项。
- 注册前，客户端的日期和时间必须已与 Uyuni 服务器同步。

#### 4.11.1.2. 添加软件通道

将 Raspberry Pi OS 客户端注册到 Uyuni 服务器之前，需要添加并同步所需的软件通道。

当前支持的系统结构为 arm64 和 armhf。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。

此过程所需的通道包括：

表格 36. Raspberry Pi OS 通道 - CLI

通道说明	arm64	armhf
基础通道	raspberrypios-12-pool-arm64-uyuni	raspberrypios-12-pool-armhf-uyuni
客户端通道	raspberrypios-12-arm64-uyuni-client	raspberrypios-12-armhf-uyuni-client
更新通道	raspberrypios-12-arm64-main-updates-uyuni	-
贡献通道	raspberrypios-12-arm64-contrib-uyuni	raspberrypios-12-armhf-contrib-uyuni
非免费通道	raspberrypios-12-arm64-non-free-uyuni	raspberrypios-12-armhf-non-free-uyuni

通道说明	arm64	armhf
非免费固件通道	raspberrypios-12-arm64-non-free-firmware-uyuni	-
Raspberry 通道	raspberrypios-12-arm64-raspberry-uyuni	raspberrypios-12-armhf-raspberry-uyuni
贡献更新	raspberrypios-12-arm64-contrib-updates-uyuni	-
非免费更新	raspberrypios-12-arm64-non-free-updates-uyuni	-
非免费固件更新	raspberrypios-12-arm64-non-free-firmware-updates-uyuni	-
安全主通道	raspberrypios-12-arm64-main-security-uyuni	-
安全贡献通道	raspberrypios-12-arm64-contrib-security-uyuni	-
安全非免费通道	raspberrypios-12-arm64-non-free-security-uyuni	-
安全非免费固件通道	raspberrypios-12-arm64-non-free-firmware-security-uyuni	-
RPI 通道	-	raspberrypios-12-armhf-rpi-uyuni

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。



spacewalk-common-channels 提供的客户端工具通道来自 Uyuni 而非 SUSE。



在引导任何 Raspberry Pi OS 客户端之前，需要完全同步所有新通道。

### 4.11.1.3. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件** > **管理** > **通道**，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **所有通道**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



Raspberry Pi OS 通道可能非常大。同步所需的时间可能长达数小时。

### 4.11.1.4. 创建激活密钥

需要创建与您的 Raspberry Pi OS 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.11.1.5. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。



Raspberry Pi OS 客户端可能需要安装多个 GPG 密钥。

同步第三方 Raspberry Pi OS 储存库时，需要在服务器上导入相应的 GPG 密钥。如果缺少 GPG 密钥，同步将会失败。

使用 Raspberry Pi OS 储存库时，系统只会对元数据签名。因此不需要导入软件通道的 GPG 密钥。Uyuni 将不会对软件包重新签名。

要查看已导入 Uyuni 服务器的 GPG 密钥，请运行以下命令：

```
mgrctl exec -- gpg --homedir /var/lib/spacewalk/gpgdir --list-keys
```

要导入新的 GPG 密钥，请运行以下命令：

```
mgradm gpg add <文件名>.gpg
```

#### 4.11.1.6. Root 访问权限

默认会为 Raspberry Pi OS 上的 root 用户禁用 SSH 访问权限。

要想能够使用普通用户进行初始配置，您需要编辑 `sudoers` 文件。

过程：向 Root 用户授予访问权限

- 在客户端上，编辑 `sudoers` 文件：

```
sudo visudo
```

在 `sudoers` 文件末尾添加下面一行，以向用户授予 `sudo` 访问权限。以在 Web UI 中引导客户端的用户的名称替换 `<user>`：

```
<user> ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2,  
/usr/bin/python3, /var/tmp/venv-salt-minion/bin/python
```



此过程无需口令便可授予 root 访问权限，而注册客户端需要提供口令。客户端成功安装后会以 root 特权运行，因此将不再需要该访问权限。客户端成功安装之后，建议您从 `sudoers` 文件中去除该行。

#### 4.11.1.7. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.12. Red Hat 客户端注册

您可以使用 Red Hat 内容分发网络 (CDN) 或 Red Hat 更新基础结构 (RHUI) 将 Red Hat Enterprise Linux 客户

端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您必须已创建激活密钥。

- 有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。
- 有关将 Red Hat Enterprise Linux 迁移到 SUSE Liberty Linux 的详细信息，请参见 [client-configuration:clients-sleses.pdf](#)。

## 4.12.1. 使用 CDN 注册 Red Hat Enterprise Linux 客户端

本节包含有关使用 Red Hat 内容分发网络 (CDN) 注册运行 Red Hat Enterprise Linux 操作系统的客户端的信息。

而有关使用 Red Hat 更新基础结构 (RHUI) 的信息，请参见 [Client-configuration > Clients-rh-rhui](#)。



您需负责安排对 Red Hat 基础媒体储存库和 RHEL 安装媒体的访问权限，以及将 Uyuni 服务器连接到 Red Hat 内容分发网络。对于您的所有 RHEL 系统，均须从 Red Hat 获取支持。如果不这么做，可能会违反 Red Hat 的条款。

### 4.12.1.1. 导入权利和证书

Red Hat 客户端需要 Red Hat 证书颁发机构 (CA) 和权利证书以及权利密钥。

权利证书内嵌失效日期，与支持订阅的期限匹配。为避免服务中断，您需要在每个支持订阅期结束时重复此过程。

Red Hat 提供了一个订阅管理器工具，可用于管理订阅指派。此工具在本地运行，可跟踪安装的产品和订阅。客户端必须在订阅管理器中注册才能获得证书。

Red Hat 客户端使用 URL 来复制储存库。URL 更改取决于 Red Hat 客户端是在何处注册的。

Red Hat 客户端可通过三种方式注册：

- redhat.com 上的 Red Hat 内容分发网络 (CDN)
- Red Hat 从属服务器
- 云中的 Red Hat 更新基础结构 (RHUI)

本指南将介绍注册到 Red Hat CDN 的客户端。您至少须有一个注册到 CDN 的系统并拥有授权订阅，以获得储存库内容。

而有关使用 Red Hat 更新基础结构 (RHUI) 的信息，请参见 [Client-configuration > Clients-rh-rhui](#)。



要为客户端系统使用从属证书，必须有从属服务器和订阅。Uyuni 服务器不支持使用从属证书的客户端。



- 权利证书内嵌失效日期，与支持订阅的期限匹配。为避免服务中断，您需要在每个支持订阅期结束时重复此过程。

Red Hat 提供了订阅管理器工具，可用于管理订阅指派。此工具在客户端系统本地运行，可跟踪安装的产品和订阅。使用订阅管理器注册到 redhat.com，然后执行下列过程获得证书。

过程：将客户端注册到订阅管理器

1. 在客户端系统上的命令提示符处注册订阅管理器工具：

```
subscription-manager register
```

出现提示时，输入您的 Red Hat 门户用户名和口令。

2. 运行命令：

```
subscription-manager activate
```

3. 将客户端系统中的权利证书和密钥复制到 Uyuni 服务器可访问的位置：

```
cp /etc/pki/entitlement/ <example>/entitlement/
```



■ 您的权利证书和密钥的文件扩展名均为 .pem。密钥的文件名中还包含 key。

4. 将客户端系统中的 Red Hat CA 证书文件复制到权利证书和密钥所在的相同网络位置：

```
cp /etc/rhsm/ca/redhat-uep.pem <example>/entitlement
```

要管理 Red Hat 客户端上的储存库，您需要将 CA 和权利证书导入 Uyuni 服务器。这需要您执行三次导入过程以创建相应的三项：三项分别对应权利证书、权利密钥和 Red Hat 证书。

过程：将证书导入服务器

1. 在 Uyuni 服务器 Web UI 上，导航到 **系统**，**自动安装**，**GPG 和 SSL 密钥**。

2. 单击 **[创建] 存储的密钥/证书**，并为权利证书设置下列参数：

- 在 **Entitlement-Cert-date** 字段中键入 Entitlement-Cert-date。
- 在 **SSL** 字段中，选择 SSL。
- 在 **证书文件** 字段中，浏览到您保存权利证书的位置，然后选择 .pem 证书文件。

3. 单击 **[创建] 密钥**。

4. 单击 **[创建存储的权利密钥/证书]**，并为权利密钥设置下列参数：
  - 在 **密钥 ID** 字段中键入 Entitlement-key-date。
  - 在 **协议** 字段中，选择 SSL。
  - 在 **密钥文件** 字段中，浏览到您保存权利密钥的位置，然后选择 .pem 密钥文件。
5. 单击 **[创建密钥]**。
6. 单击 **[创建存储的权利密钥/证书]**，并为 Red Hat 证书设置下列参数：
  - 在 **密钥 ID** 字段中键入 redhat-uep。
  - 在 **协议** 字段中，选择 SSL。
  - 在 **证书文件** 字段中，浏览到您保存 Red Hat 证书的位置，然后选择证书文件。
7. 单击 **[创建密钥]**。

### 4.12.1.2. 添加软件通道

将 Red Hat 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 37. Red Hat 通道 - CLI

操作系统版本	基础通道	客户端通道	工具通道
Red Hat 9	rhel9-pool-uyuni	-	rhel9-uyuni-client
Red Hat 8	rhel8-pool-uyuni	-	rhel8-uyuni-client
Red Hat 7	rhel7-pool-uyuni	-	rhel7-uyuni-client

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 spacewalk-common-channels 命令添加相应的通道。请确保指定正确的体系结构：

```
spacewalk-common-channels \
-a <体系结构> \
<基础通道名称> \
<子通道名称 1> \
<子通道名称 2> \
... <子通道名称 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <base_channel_label>-<architecture>
```

3. 确保同步已完成，然后再继续操作。



spacewalk-common-channels 提供的客户端工具通道来自 Uyuni 而非 SUSE。

### 4.12.1.3. 准备自定义储存库和通道

要从 Red Hat CDN 镜像软件，需要在 Uyuni 中创建自定义通道和储存库，它们都将通过 URL 链接到 CDN。必须在 Red Hat 门户中拥有这些产品的权利，此功能才能正常工作。可以使用订阅管理器工具获得要镜像的储存库的 URL：

```
subscription-manager repos
```

可以使用这些储存库 URL 来创建自定义储存库。这样您便可只镜像管理客户端所需的内容。



如果您在 Red Hat 门户中拥有正确的权利，就可以只创建 Red Hat 储存库的自定义版本。

此过程所需的细节包括：

表格 38. Red Hat 自定义储存库设置

选项	设置
储存库 URL	Red Hat CDN 提供的内容 URL
包含已签名的元数据？	取消选中所有 Red Hat Enterprise 储存库
SSL CA 证书	redhat-uep
SSL 客户端证书	Entitlement-Cert-date
SSL 客户端密钥	Entitlement-Key-date

过程：创建自定义储存库

1. 在 Uyuni 服务器 Web UI 上，导航到 **软件 > 管理 > 储存库**。
2. 单击 **[ 创建储存库 ]**，然后为储存库设置适当的参数。
3. 单击 **[ 创建储存库 ]**。
4. 对需要创建的所有储存库重复以上步骤。

此过程所需的通道包括：

表格 39. Red Hat 自定义通道

操作系统版本	基础通道
Red Hat 9	rhel9-pool-uyuni
Red Hat 8	rhel8-pool-uyuni
Red Hat 7	rhel7-pool-uyuni

过程：创建自定义通道

1. 在 Uyuni 服务器 Web UI 上，导航到软件，管理，通道。
2. 单击 **[ 创建通道 ]**，然后为通道设置相应的参数。
3. 在 **基础通道** 字段中，选择相应的基础通道。
4. 单击 **[ 创建通道 ]**。
5. 对需要创建的所有通道重复以上步骤。每个自定义储存库都应该有一个自定义通道。

您可以导航到软件，通道列表，所有，以检查是否已创建所有相应的通道和储存库。



对于 Red Hat 9 和 Red Hat 8 客户端，请添加基础通道和 AppStream 通道。您需要来自这两个通道的软件包。如果未添加这两个通道，将会因缺少软件包而无法创建引导储存库。

如果您使用的是模块化通道，则必须在客户端上启用 Python 3.6 模块流。如果不提供 Python 3.6，spacecmd 软件包安装将会失败。

创建所有通道之后，可以将其与您创建的储存库关联：

过程：将通道与储存库关联

1. 在 Uyuni 服务器 Web UI 上，导航到软件，管理，通道，然后单击要关联的通道。
2. 导航到 **关联** 选项卡，然后选中要与此通道关联的储存库。
3. 单击 **[ 更新储存库 ]** 以将通道与储存库相关联。
4. 对需要关联的所有通道和储存库重复以上步骤。
5. 可选：导航到 **同步** 选项卡，为此储存库设置定期同步日程安排。
6. 单击 **[ 立即同步 ]** 以立即开始同步。

#### 4.12.1.4. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到软件，管理，通道，然后单击与储存库关联的通道。
2. 导航到 **同步** 选项卡，然后单击 **同步** 并选中 **立即同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



Red Hat Enterprise Linux 通道可能非常大。同步所需的时间可能长达数小时。

过程：可选：创建 Salt 状态以部署配置文件

1. 在 Uyuni 服务器 Web UI 上，导航到 **配置 > 通道**。
2. 单击 **[ 创建 | 状态 | 通道 ]**。
  - 在 **名称** 字段中，键入 `subscription-manager: disable yum plugins`。
  - 在 **命令** 字段中，键入 `subscription-manager-disable-yum-plugins`。
  - 在 **SLS** 字段中，键入 `subscription-manager: disable yum plugins`。
  - 将 SLS **文件** 字段留空。
3. 单击 **[ 创建 | 配置 | 通道 ]**
4. 单击 **[ 创建 | 配置 | 文件 ]**
  - 在 **文件** 字段中，键入 `/etc/yum/pluginconf.d/subscription-manager.conf`。
  - 在 **内容** 字段中，键入：

```
[main]
enabled=0
```

5. 单击 **[ 创建 | 配置 | 文件 ]**
6. 记下 Salt **文件** 字段的值。
7. 单击配置通道的名称。
8. 单击 **编辑 / 目录 'init.sls'**
  - 在 **文件** 字段中，键入：

```
configure_subscription-manager-disable-yum-plugins:
cmd.run:
  - name: subscription-manager config
--rhsm.auto_enable_yum_plugins=0
  - watch:
    - file: /etc/yum/pluginconf.d/subscription-manager.conf
file.managed:
  - name: /etc/yum/pluginconf.d/subscription-manager.conf
  - source: salt:///etc/yum/pluginconf.d/subscription-
manager.conf
```

9. 单击 。



 Salt 是可选过程。

过程：为 Red Hat Enterprise Linux 客户端创建系统组

1. 在 Uyuni 服务器 Web UI 上，导航到 **系统** > **系统组**。
2. 单击 。
  - 在 **名称** 字段中，键入 `rhel-systems`。
  - 在 **别名** 字段中，键入 `RHEL`。
3. 单击 .
4. 单击 **常规** 选项卡。
5. 单击 **订阅** 选项卡。
6. 在搜索框中键入 `subscription-manager: disable yum plugins`。
7. 单击  以查看状态。
8. 单击 **状态** 列中的状态对应的复选框。
9. 单击 .
10. 单击 .

如果您已将 RHEL 系统添加到 Uyuni，请将其指派给新的系统组，然后应用 Highstate。

过程：将系统组添加到激活密钥

您需要修改用于 RHEL 系统的激活密钥，以包含之前创建的系统组。

1. 在 Uyuni 服务器 Web UI 上，导航到 **系统** > **激活密钥**。
2. 单击用于 RHEL 系统的每个激活密钥并：
3. 依次导航到 **常规** 选项卡和 **子系统** 子选项卡。
4. 选中 **rhel-systems**。

5. 单击 **[加入所选的组]**。

### 4.12.1.5. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。



对于 Red Hat 自定义通道，如果您要选中 **GPG URL** 字段，则需填充 **GPG URL** 字段。您可以使用 Red Hat 受控端 /etc/pki/rpm-gpg 目录下存储的 GPG 密钥的文件 URL。

示例：`file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release`

有关 Red Hat 产品签名密钥的完整列表，请参见 <https://access.redhat.com/security/team/key>。

### 4.12.1.6. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 4.12.2. 使用 RHUI 注册 Red Hat Enterprise Linux 客户端

本节包含有关使用 Red Hat 更新基础架构 (RHUI) 注册运行 Red Hat Enterprise Linux 操作系统的客户端的信息。

如果您在公有云（例如 Amazon EC2）中运行客户端，请使用此方法。

可以将 RHUI 与 Red Hat 内容分发网络 (CDN) 搭配使用来管理您的 Red Hat Enterprise Linux 订阅。有关使用 Red Hat CDN 的信息，请参见 [Client-configuration > Clients-rh-cdn](#)。



您需负责将 Uyuni 服务器连接到 Red Hat 更新基础结构。所有使用此 RHUI 证书进行更新的客户端都需要获得正确许可，有关详细信息，请咨询您的云服务提供商，并查看服务的 Red Hat 条款。



当使用 RHUI 注册的 Red Hat Enterprise Linux 客户端关闭时，Red Hat 可能会声称证书无效。在此情况下，您需要再次打开客户端，或获取新的 RHUI 证书。

### 4.12.2.1. 导入权利和证书

过去，必须手动将证书和权利数据导入到 Uyuni 服务器。可以使用用于 SUSE PAYG 实例的相同机制来自动完成此任务。有关详细信息，请参见 [Installation-and-upgrade > Connect-payg](#)。

本指南将介绍注册到 Red Hat 更新基础结构 (RHUI) 的客户端。您至少须有一个注册到 RHUI 的系统并拥有授权订阅，以获得储存库内容。

而有关使用 Red Hat 内容分发网络 (CDN) 的信息，请参见 [Client-configuration > Clients-rh-cdn](#)。



要为客户端系统使用从属证书，必须有从属服务器和订阅。Uyuni 服务器不支持使用从属证书的客户端。



PAYG 连接会定期与客户端通讯以获取最新的身份验证数据。请务必保持客户端运行状态并定期更新。如果不这样，储存库同步有时将会失败，并显示身份验证错误。



请在连接任何 Red Hat 7 实例前先将其更新。



要连接到 Red Hat 9 实例，必须已为其配置加密策略 LEGACY。请执行 `sudo update-crypto-policies --set LEGACY` 进行相应配置。

### 4.12.2.2. 连接到 Red Hat 更新基础结构

过程：连接新的 Red Hat 实例

1. 在 Uyuni Web UI 中，导航到 **管理 > 安装向导 > PAYG**，然后单击 **[+] [添加 PAYG]**。
2. 从页面的 **PAYG** 部分开始。
3. 在 **说明** 字段中添加说明。
4. 转移到页面的 **SSH** 部分。
5. 在 **DNS 或 IP 地址** 字段中，键入要从 Uyuni 连接的实例 DNS 或 IP 地址。
6. 在 **端口号** 字段中输入端口号或使用默认值 22。
7. 在 **用户名** 字段中输入云中指定的用户名。
8. 在 **口令** 字段中输入口令。
9. 在 **实例密钥** 字段中输入实例密钥。
10. 在 **密钥通行口令** 字段中输入密钥通行口令。



身份验证密钥必须采用 PEM 格式。

如果您要通过 SSH 堡垒连接实例而不是直接连接，请执行[过程：添加 SSH 堡垒连接数据](#)。

否则，请执行[过程：完成 Red Hat 连接](#)。

过程：添加 SSH 堡垒连接数据

1. 导航到页面的 **SSH 堡垒连接** 部分。
2. 在 **堡垒主机名** 字段中输入堡垒主机名。
3. 在 **堡垒端口号** 字段中输入堡垒端口号。
4. 在 **堡垒用户名** 字段中输入堡垒用户名。
5. 在 **堡垒口令** 字段中输入堡垒口令。
6. 在 **堡垒密钥** 字段中输入堡垒密钥。
7. 在 **堡垒密钥通行口令** 字段中输入堡垒密钥通行口令。

执行[过程：完成 Red Hat 连接](#)完成设置。

过程：完成 Red Hat 连接

1. 要完成添加新 Red Hat 连接数据的过程，请单击 **[创建]**。
2. 返回 PAYG 连接数据 **连接** 页面。顶部的 **连接** 部分会显示更新的连接状态。
3. **连接** > **PAYG** > Pay-as-you-go 屏幕中也会显示连接状态。
4. 如果实例的身份验证数据正确，**连接** 列会显示 **已连接**。



如果在任何时间输入了无效数据，**连接** > **PAYG** > PAYG 中会显示新创建的实例，同时 **连接** 列会显示错误消息。

当服务器上有可用的身份验证数据后，系统会立即为已连接实例上所有可用的储存库添加储存库。您可以在 **连接** > **储存库** 中查看储存库。



Red Hat 连接会创建默认由组织 1 拥有的自定义储存库。如果应由另一个组织拥有自动生成的储存库，请在 `/etc/rhn/rhn.conf` 中配置 `java.rhui_default_org_id`。

此设置仅可定义和更新储存库。如果您想对受管客户端使用某个储存库，则需要指定软件通道并将储存库关联到该软件通道。

### 4.12.2.3. 添加软件通道

将 Red Hat 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 40. Red Hat 通道 - CLI

操作系统版本	基础通道	客户端通道	工具通道
Red Hat 7	rhel7-pool-uyuni	-	rhel7-uyuni-client
Red Hat 8	rhel8-pool-uyuni	-	rhel8-uyuni-client
Red Hat 9	rhel9-pool-uyuni	-	rhel9-uyuni-client

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道。请确保指定正确的体系结构：

```
spacewalk-common-channels \
-a <体系结构> \
<基础通道名称> \
<子通道名称 1> \
<子通道名称 2> \
... <子通道名称 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <base_channel_label>-<architecture>
```

3. 确保同步已完成，然后再继续操作。



spacewalk-common-channels 提供的客户端工具通道来自 Uyuni 而非 SUSE。

#### 4.12.2.4. 准备自定义通道

要从 RHUI 镜像软件，您需要在 Uyuni 中创建关联到自动生成的储存库的自定义通道。

此过程所需的通道包括：

表格 41. Red Hat 自定义通道

操作系统版本	基础通道
Red Hat 7	rhel7-pool-uyuni
Red Hat 8	rhel8-pool-uyuni
Red Hat 9	rhel9-pool-uyuni

过程：创建自定义通道

1. 在 Uyuni 服务器 Web UI 上，导航到[软件](#)，[管理](#)，[通道](#)。

2. 单击 **[创建通道]**，然后为通道设置相应的参数。
3. 在 **基础通道** 字段中，选择相应的基础通道。
4. 单击 **[创建通道]**。
5. 对需要创建的所有通道重复以上步骤。每个自定义储存库都应该有一个自定义通道。

您可以导航到 **软件** > **通道列表** > **所有**，以检查是否已创建所有相应的通道和储存库。



对于 Red Hat 9 和 Red Hat 8 客户端，请添加基础通道和 AppStream 通道。您需要来自这两个通道的软件包。如果未添加这两个通道，将会因缺少软件包而无法创建引导储存库。

创建所有通道之后，可以将其与您创建的储存库关联：

过程：将通道与储存库关联

1. 在 Uyuni 服务器 Web UI 上，导航到 **软件** > **管理** > **通道**，然后单击要关联的通道。
2. 导航到 **储存库** 选项卡，然后选中要与此通道关联的储存库。
3. 单击 **[更新储存库]** 以将通道与储存库相关联。
4. 对需要关联的所有通道和储存库重复以上步骤。
5. 可选：导航到 **日程安排** 选项卡，为此储存库设置定期同步日程安排。
6. 单击 **[立即同步]** 以立即开始同步。

#### 4.12.2.5. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到 **软件** > **管理** > **通道**，然后单击与储存库关联的通道。
2. 导航到 **储存库** 选项卡，然后单击 **同步** 并选中 **立即同步**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



Red Hat Enterprise Linux 通道可能非常大。同步所需的时间可能长达数小时。

#### 4.12.2.6. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

#### 4.12.2.7. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 4.13. Rocky Linux 客户端注册

您可以将 Rocky Linux} 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您必须已创建激活密钥。

- 有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。
- 有关将 Rocky Linux 迁移到 SUSE Liberty Linux 的详细信息，请参见 [client-configuration:clients-sleses.pdf](#)。

#### 4.13.1. 注册 Rocky Linux 客户端

本节包含有关注册运行 Rocky Linux 操作系统的客户端的信息。



我们已使用----策略并采用默认的 SELinux 配置对将 Rocky Linux 客户端注册到 Uyuni 的过程进行了测试。将 Rocky Linux 客户端注册到 Uyuni 时，您无需禁用 SELinux。

##### 4.13.1.1. 添加软件通道

将 Rocky Linux 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。

目前支持的系统结构有 x86\_64 和 aarch64，在版本 9 上，另外还支持 ppc64le 和 s390x。有关支持的产品和体系结构的完整列表，请参见 [Client-configuration > Supported-features](#)。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 42. Rocky Linux 通道 - CLI

操作系统版本	基础通道	客户端通道	AppStream 通道
Rocky Linux 9	rockylinux9	rockylinux9-uyuni-client	rockylinux9-appstream
Rocky Linux 8	rockylinux8	rockylinux8-uyuni-client	rockylinux8-appstream

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道。请确保指定正确的体系结构：

```
spacewalk-common-channels \
-a <体系结构> \
<基础通道名称> \
<子通道名称 1> \
<子通道名称 2> \
... <子通道名称 n>
```

2. 如果 [自动同步](#) 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <base_channel_label>-<architecture>
```

3. 确保同步已完成，然后再继续操作。



`spacewalk-common-channels` 提供的客户端工具通道来自 Uyuni 而非 SUSE。



对于 Rocky Linux 9 和 Rocky Linux 8 客户端，请添加基础通道和 AppStream 通道。您需要来自这两个通道的软件包。如果未添加这两个通道，将会因缺少软件包而无法创建引导储存库。



您可能会发现 AppStream 通道中提供的软件包数量在上游通道和 Uyuni 通道之间存在一定的差异。如果您对在不同时间添加的同一通道进行比较，会发现其数量也不相同。这是由 Rocky Linux 管理其储存库的方式所致。当有新版本发布时，Rocky Linux 会去除软件包的较旧版本，而 Uyuni 则会保留所有版本，无论新旧与否。

如果您在 Rocky Linux 8 中使用模块化通道，则必须在客户端上启用 Python 3.6 模块流。如果不提供 Python 3.6，`spacecmd` 软件包安装将会失败。

### 4.13.1.2. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到软件，管理，通道，然后单击与储存库关联的通道。
2. 导航到同步选项卡，然后单击同步并选中同步复选框。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 tail 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。

### 4.13.1.3. 创建激活密钥

您需要创建与您的 Rocky Linux 通道关联的激活密钥。

有关激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.13.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。如果不信任 GPG 密钥，便无法为客户端指派软件通道。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

### 4.13.1.5. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 4.13.1.6. 管理勘误

当您更新 Rocky Linux 客户端时，软件包会包含有关更新的元数据。

## 4.14. Ubuntu 客户端注册

您可以将 Ubuntu 客户端注册到 Uyuni 服务器。方法和细节视客户端的操作系统而异。

开始前，请确保客户端的日期和时间已与 Uyuni 服务器正确同步。

您还必须已创建好激活密钥。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

### 4.14.1. Registering Ubuntu 20.04, 22.04 and 24.04 Clients

This section contains information about registering clients running Ubuntu 20.04 LTS, 22.04 LTS and 24.04 LTS operating systems.

支持使用引导功能启动 Ubuntu 客户端，并执行初始状态运行，例如设置储存库和执行配置文件更新。不过，默认会禁用 Ubuntu 上的 root 用户，因此要使用引导，需要有一个具有 Python sudo 特权的现有用户。



Canonical 不为 Uyuni 背书，也不为其提供支持。

#### 4.14.1.1. 添加软件通道

将 Ubuntu 客户端注册到您的 Uyuni 服务器之前，您需要添加所需的软件通道，并同步这些通道。



下面的小节中的说明通常默认使用 x86\_64 体系结构。请根据情况将其替换为其他体系结构。

此过程所需的通道包括：

表格 43. Ubuntu 通道 - CLI

OS Version	Base Channel	Main Channel	Updates Channel	Security Channel	Client Channel
Ubuntu 24.04	ubuntu-2404-pool-amd64-uyuni	ubuntu-2404-amd64-main-uyuni	ubuntu-2404-amd64-main-updates-uyuni	ubuntu-2404-amd64-main-security-uyuni	ubuntu-2404-amd64-uyuni-client
Ubuntu 22.04	ubuntu-2204-pool-amd64-uyuni	ubuntu-2204-amd64-main-uyuni	ubuntu-2204-amd64-main-updates-uyuni	ubuntu-2204-amd64-main-security-uyuni	ubuntu-2204-amd64-uyuni-client
Ubuntu 20.04	ubuntu-2004-pool-amd64-uyuni	ubuntu-2004-amd64-main-uyuni	ubuntu-2004-amd64-main-updates-uyuni	ubuntu-2004-amd64-main-security-uyuni	ubuntu-2004-amd64-uyuni-client

版本 20.04 还需要通用通道：

表格 44. Ubuntu 20.04 通用通道 - CLI

Ubuntu 20.04	
通用通道	ubuntu-2004-amd64-universe-uyuni
通用更新通道	ubuntu-2004-amd64-universe-updates-uyuni
通用安全更新通道	ubuntu-2004-amd64-universe-security-uyuni
通用向后移植通道	ubuntu-2004-amd64-universe-backports-uyuni

过程：在命令提示符处添加软件通道

1. 在 Uyuni 服务器上的命令提示符下，以 root 身份使用 `spacewalk-common-channels` 命令添加相应的通道：

```
spacewalk-common-channels \
<基础通道标签> \
<子通道标签 1> \
<子通道标签 2> \
... <子通道标签 n>
```

2. 如果 **自动同步** 处于关闭状态，请同步通道：

```
spacewalk-repo-sync -p <基础通道标签>
```

3. 确保同步已完成，然后再继续操作。



在引导任何 Ubuntu 客户端之前，您需要完全同步所有新通道。

### 4.14.1.2. 镜像 Ubuntu ESM 软件包

Canonical 为 Ubuntu Pro 用户和客户提供 **扩展安全维护 (ESM)** 软件包。这些软件包为多个操作系统组件和选定的应用程序提供更长的维护期（10 至 12 年）。

如果您从已注册到 Ubuntu Pro 的系统中提取了所需的 **GPG 密钥** 和您的个人**持有者令牌**，则这些储存库还可以在 Uyuni 中同步。

#### 4.14.1.2.1. 提取 GPG 密钥和持有者令牌

将 Ubuntu 主机注册到 Ubuntu Pro。您可以在 **Ubuntu Pro 仪表板** 中找到自己的个人注册令牌。为此您需要创建一个 **Ubuntu One 帐户**。

```
sudo apt-get install ubuntu-adantage-tools
sudo pro attach <个人令牌>
```

注册后，可以在 `/etc/apt/auth.conf.d/90ubuntu-adantage` 文件中找到持有者令牌：

```
machine esm.ubuntu.com/apps/ubuntu/ login bearer password <令牌> #
ubuntu-pro-client
machine esm.ubuntu.com/infra/ubuntu/ login bearer password <令牌> #
ubuntu-pro-client
```



将为每个储存库使用一个专用的持有者令牌。

在 Uyuni 中配置以下储存库：

#### 4.14.1.2.2. 配置 Ubuntu ESM 储存库

使用以下 URL 创建储存库：

表格 45. Ubuntu ESM 储存库

URL	说明
<a href="https://bearer:&lt;token&gt;@esm.ubuntu.com/infra/ubuntu/dists/&lt;release&gt;-infra-updates/main/binary-&lt;arch&gt;/">https://bearer:&lt;token&gt;@esm.ubuntu.com/infra/ubuntu/dists/&lt;release&gt;-infra-updates/main/binary-&lt;arch&gt;/</a>	操作系统功能更新
<a href="https://bearer:&lt;token&gt;@esm.ubuntu.com/infra/ubuntu/dists/&lt;release&gt;-infra-security/main/binary-&lt;arch&gt;/">https://bearer:&lt;token&gt;@esm.ubuntu.com/infra/ubuntu/dists/&lt;release&gt;-infra-security/main/binary-&lt;arch&gt;/</a>	操作系统安全更新
<a href="https://bearer:&lt;token&gt;@esm.ubuntu.com/apps/ubuntu/dists/&lt;release&gt;-apps-updates/main/binary-&lt;arch&gt;/">https://bearer:&lt;token&gt;@esm.ubuntu.com/apps/ubuntu/dists/&lt;release&gt;-apps-updates/main/binary-&lt;arch&gt;/</a>	应用程序功能更新
<a href="https://bearer:&lt;token&gt;@esm.ubuntu.com/apps/ubuntu/dists/&lt;release&gt;-apps-security/main/binary-&lt;arch&gt;/">https://bearer:&lt;token&gt;@esm.ubuntu.com/apps/ubuntu/dists/&lt;release&gt;-apps-security/main/binary-&lt;arch&gt;/</a>	应用程序安全更新

请将 `<token>` 替换为您的个人持有者令牌。此外，必须将 `arch` 和 `release` 替换为以下值之一：

表格 46. Ubuntu ESM 体系结构和版本

体系结构	版本
amd64、arm64、armel、armhf、i386、powerpc 、ppc64el、s390x	bionic、focal、jammy、noble、trusty、xenial

要让 Uyuni 同步储存库，必须导入相应的 GPG 密钥 (`ubuntu-advantage-esm-infra-trusty.gpg`、`ubuntu-advantage-esm-apps.gpg`)。在已注册到 Ubuntu Pro 的系统上的 `/etc/apt/trusted.gpg.d` 中可以找到这些密钥。将这些文件复制到 Uyuni 系统，并如下所示将其导入：

```
mgradm gpg add /path/to/gpg.key
```

在已同步的 Ubuntu 父通道下创建相应的子通道。之后，便可以同步储存库了。



此处所述的过程可用于规避订阅限制 - 但是，这样会违反服务条款并可能导致**法律后果**。始终必须根据所用的系统数量提供足够的订阅。

### 4.14.1.3. 检查同步状态

过程：在 Web UI 中检查同步进度

1. 在 Uyuni Web UI 中，导航到**软件 > 管理 > 通道**，然后单击与储存库关联的通道。
2. 导航到**同步**选项卡，然后单击**同步**并选中**所有通道**。

过程：在命令提示符处检查同步进度

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 `tail` 命令检查同步日志文件：

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. 每个子通道在同步过程中都会生成自己的日志。您需要检查所有基础通道和子通道日志文件，以确保同步已完成。



Ubuntu 通道可能会非常大。同步所需的时间可能会长达数小时。

### 4.14.1.4. 管理 GPG 密钥

安装软件包之前，客户端会使用 GPG 密钥检查这些软件包的真实性。只有可信软件才能安装在客户端上。



信任 GPG 密钥对于客户端的安全非常重要。由管理员来决定需要哪些密钥，可以信任哪些密钥。这是因为如果不信任 GPG 密钥，便无法使用软件通道，是否将通道指派给客户端取决于是否信任密钥。

有关 GPG 密钥的详细信息，请参见 [Client-configuration > Gpg-keys](#)。

### 4.14.1.5. Root 访问权限

默认会为 Ubuntu 上的 root 用户禁用 SSH 访问权限。

要想能够使用普通用户进行初始配置，您需要编辑 `sudoers` 文件。



自行安装版 Ubuntu 会发生此问题。如果在安装期间为默认用户授予了管理权限，则需要使用 `sudo` 提供口令来进行特权提升。云实例不会出现此问题，因为 `cloud-init` 会自动在 `/etc/sudoers.d` 下创建一个文件，并且无需口令即可通过 `sudo` 进行特权提升。

过程：向 Root 用户授予访问权限

1. 在客户端上，编辑 `sudoers` 文件：

```
sudo visudo
```

在 sudoers 文件末尾添加下面一行，以向用户授予 sudo 访问权限。以在 Web UI 中引导客户端的用户的名称替换 <user>：

```
<user> ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2,
/usr/bin/python3, /var/tmp/venv-salt-minion/bin/python
```



此过程无需口令便可授予 root 访问权限，而注册客户端需要提供口令。客户端成功安装后会以 root 特权运行，因此将不再需要该访问权限。客户端成功安装之后，建议您从 sudoers 文件中去除该行。

#### 4.14.1.6. 注册客户端

要注册您的客户端，需要有引导储存库。默认会自动创建引导储存库，并且每天会为所有同步的产品重新生成该储存库。您可以在命令提示符处使用以下命令手动创建引导储存库：

```
mgr-create-bootstrap-repo
```

有关注册客户端的详细信息，请参见 [Client-configuration > Registration-overview](#)。

## 4.15. 将客户端注册到代理

代理服务器可充当客户端的中介和软件包缓存。将客户端注册到代理与将其直接注册到 Uyuni 服务器的过程类似，主要差异只有几处。

以下章节包含有关使用 Web UI、命令行命令或引导脚本将客户端注册到代理的信息，以及有关将客户端从一个 Uyuni 代理移至另一个代理或 Uyuni 服务器的过程。

在 Web UI 中，代理页面会显示有关客户端的信息。可以在 **系统**，**系统列表**，**代理**中单击代理的名称，然后选择 **-----** 选项卡的 **-----** 子选项卡，以查看连接到代理的客户端列表。

在 **系统**，**所有** 中单击某个客户端的名称，然后选择 **-----** 选项卡的 **-----** 子选项卡，可以查看该客户端关联的代理列表。

### 4.15.1. 在代理之间移动客户端

无需重复注册过程，即可在代理之间移动 Salt 客户端和 Salt SSH 推送客户端。



链式代理无法移动。您应创建新代理，移动客户端，然后删除旧代理，而不是移动链式代理。

过程：在代理之间移动客户端或 Salt SSH 推送客户端

1. 在 Uyuni Web UI 中，导航到要在代理之间移动的客户端的 **-----** 页面。
2. 导航到 **-----** 选项卡，然后单击 **-----** 链接，您会看到下拉菜单。
3. 在 **-----** 下拉菜单中，选择要移动到的目标代理，然后单击 **[更改代理]**。

过程：使用 SSM 在代理之间移动多个客户端或 Salt SSH 推送客户端

1. 在 Uyuni Web UI 中，导航到 **系统**，**系统列表**，然后选中要移动的每个客户端，如此会将这些客户端添加到系统集管理器中。
2. 导航到 **系统**，**系统集管理器**，然后转到 **其他**，**代理** 选项卡。
3. 在 **-----** 下拉菜单中，选择要将客户端移动到的目标代理，然后单击 **[更改代理]**。

通过调用 `system.changeProxy` 也能使用该功能。

### 4.15.1.1. 背景信息

此功能对常规 Salt 客户端与 Salt SSH Push 客户端的作用有所不同。

#### 4.15.1.1.1. 默认客户端

该功能会安排一项 Salt 状态操作，该操作会修改 `susemanager.conf` Salt 配置文件中的 `master:` 设置，使其指向新代理。然后该功能会重启 Salt 客户端。



通过手动编辑 `susemanager.conf` 文件来更改 `master:` 也有同样的效果，并且系统也支持这种方式。

当客户端重启并通过新代理重新连接时，服务器会在数据库中更新代理路径，并安排另一项操作来刷新通道 URL。

#### 4.15.1.1.2. Salt SSH 推送客户端

该功能会立即在数据库中更新代理路径并安排新操作来刷新通道 URL。

### 4.15.2. 将客户端从代理移到服务器

如果您要将客户端从代理移到服务器，请从代理列表中选择 **--**。

### 4.15.3. 使用 Web UI 将客户端注册到代理

可以使用 Web UI 将客户端注册到 Uyuni 代理。

一般的非 SLE 客户端和早于版本 15 的 SLE 客户端都需要引导储存库。

有关创建引导储存库的信息，请参见 **Client-configuration > Bootstrap-repository**。

过程：使用 Web UI 将客户端注册到代理

1. 在 Uyuni Web UI 中，导航到系统 > 引导。
2. 在 FQDN 字段中，键入要引导的客户端的完全限定域名 (FQDN)。
3. 在 SSH 端口号字段中，键入用于连接和引导客户端的 SSH 端口号。SSH 端口默认为 22。
4. 在用户名字段中，键入用于登录客户端的用户名。用户名默认为 root。
5. 在身份验证方法字段中，选择用于引导客户端的身份验证方法。
  - 如果选择的是口令身份验证，请在口令字段中，键入用于登录客户端的口令。
  - 如果选择的是 SSH 私用密钥身份验证，请输入私用密钥和关联的通行口令。该密钥的存储期截止到引导过程完成。
6. 在激活密钥字段中，选择与您要用于引导客户端的软件通道关联的激活密钥。
7. 在代理服务器字段中，选择要注册到的代理服务器。
8. 默认会选中 SSH 主机密钥复选框。如此可让引导过程自动接受 SSH 主机密钥，而无需您手动进行身份验证。
9. 可选：选中 SSH 通行口令复选框。如果您选中此选项，客户端将会配置为使用 SSH 来连接服务器，且不再配置其他连接方法。
10. 单击 **[Bootstrap]** 开始注册。

引导过程完成时，您的客户端即会列在系统 > 系统列表中。

#### 4.15.4. 在命令行上注册客户端

除了 Web UI 以外，您也可以使用命令行将客户端注册到代理。要执行此过程，您需要在注册前于客户端上安装 Salt 软件包。对于基于 SLE 12 的客户端，您还需要激活 `minion` 模块。

过程：使用命令行将客户端注册到代理

1. 选择位于以下位置的客户端配置文件：

```
/etc/salt/minion
```

或：

```
/etc/salt/minion.d/NAME.conf
```

该文件有时也称为受控端文件。

2. 将代理 FQDN 作为 master 添加到客户端配置文件：

```
master: PROXY123.EXAMPLE.COM
```

3. 重启 salt-minion 服务：

```
systemctl restart salt-minion
```

4. 在服务器上，接受新客户端密钥（将 <client> 替换为您客户端的名称）：

```
salt-key -a '<客户端>'
```

## 4.16. 在公有云上注册客户端

设置好 Uyuni 服务器后，您便可开始注册客户端。

### 4.16.1. 添加产品并同步储存库

确保您已添加客户端的对应产品并已将储存库同步到 Uyuni。要创建用于注册客户端的引导储存库，必须执行此操作。

有关详细信息，请参见 [Specialized-guides > Public-cloud-guide](#)。

### 4.16.2. 准备按需映像

使用 SUSE 提供的按需映像开始的实例会自动进行注册，并且更新基础结构和 SUSE Linux Enterprise 模块均已激活。要使用按需映像作为 Uyuni 客户端，您需要在开始前禁用此自动功能。

过程：准备按需映像

1. 登录到按需实例。
2. 在命令提示符处，以 root 身份去除注册数据和储存库：

```
registercloudguest --clean
```

3. 去除自动注册的触发服务：

```
systemctl disable guestregister.service
```

4. 在 Microsoft Azure 中，需要禁用另一项服务：

```
systemctl disable regionsrv-enabler-azure.timer
```

### 4.16.3. 注册客户端

在 Uyuni Web UI 中，导航到 **系统 > 引导**，然后填写 **名称**、**SSH** **用户名**、**密码** 和 **FQDN** 字段。请确保为 **FQDN** 字段使用稳定的 FQDN，否则，如果公有云为您分配一个不同的临时 FQDN，Uyuni 将找不到您的主机。

公有云映像通常不允许使用用户名和口令进行 SSH 登录，仅允许通过证书进行 SSH 登录。如果您要从 Web UI 中使用引导，则需启用通过用户名和 SSH 密钥进行 SSH 登录的功能。您可以通过导航到 **系统 > 引导** 并更改身份验证方法来启用此功能。

如果您的云提供商是 Microsoft Azure，您可以通过用户名和口令登录。要实现此目的，您需要允许 AzureUser 以 root 身份无口令运行命令。为此，请打开 `/etc/sudoers.d/waagent` 文件，并添加或编辑下面一行：

```
AzureUser ALL=(ALL) NOPASSWD: ALL
```



允许 AzureUser 以 root 身份无口令运行命令会带来安全风险。请仅将此方法用于测试目的，不要用于生产系统。

引导过程成功完成时，您的客户端即会列在 **系统 > 系统列表** 中。

- 如果您想更好地控制注册过程或必须注册许多客户端，请创建引导脚本。有关详细信息，请参见 **Client-configuration > Registration-bootstrap**。
- 如果要注册 Salt 客户端而且想更好地控制注册过程，在命令行上执行单个命令较为合适。有关详细信息，请参见 **Client-configuration > Registration-cli**。
- 注册从公有云映像（例如 AWS AMI）启动的客户端时，需要进行额外的配置以防它们彼此重写。有关注册克隆客户端的详细信息，请参见 **Administration > Troubleshooting**。

### 4.16.4. 激活密钥

激活密钥用于确保客户端拥有正确的软件权利、可连接到适当的通道以及订阅相关的组。每个激活密钥都绑定到一个组织，您可以在创建密钥时设置此项。

有关激活密钥的详细信息，请参见 **Client-configuration > Activation-keys**。

### 4.16.5. 自动注册 Terraform 创建的客户端

Terraform 创建的新客户端可以自动注册到 Uyuni 中。可以通过以下两种方式来完成注册：

- 基于 `cloud-init` 的注册
- 基于 `remote-exec` 置备器的注册

#### 4.16.5.1. 基于 `cloud-init` 的客户端注册

自动注册新创建的虚拟机的首选方式是利用 `cloud-init` 进行注册。采用这种方法无需配置与主机的 SSH 连接。此外，无论使用哪种工具创建客户端，都可以采用这种方法。

用户可以在使用 Terraform 部署映像时传递用户数据集，以便将虚拟机自动注册到 Uyuni 中。`user_data` 只会在虚拟机首次启动时的引导期间运行一次。

使用 `cloud-init` 注册客户端之前，用户必须配置：

- 引导脚本。有关详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。
- 激活密钥。有关详细信息，请参见 [Client-configuration > Activation-keys](#)。

下面的命令会下载引导脚本并在新虚拟机创建后注册该虚拟机。应将此命令添加到 `cloud-init` 配置中：

```
curl -s http://hub-server.tf.local/pub/bootstrap/bootstrap-default.sh |  
bash -s
```



无论何时更新 `user_data` 来更改置备，Terraform 都会销毁虚拟机，然后使用新 IP 等信息重新创建虚拟机。

有关 AWS 上的 `cloud-init` 的详细信息，请参见 [https://registry.terraform.io/providers/hashicorp/template/latest/docs/data-sources/cloudinit\\_config](https://registry.terraform.io/providers/hashicorp/template/latest/docs/data-sources/cloudinit_config)。

有关 `cloud-init` 示例，请参见 [https://registry.terraform.io/providers/hashicorp/cloudinit/latest/docs/data-sources/cloudinit\\_config#example-usage](https://registry.terraform.io/providers/hashicorp/cloudinit/latest/docs/data-sources/cloudinit_config#example-usage)。

#### 4.16.5.2. 基于 `remote-exec` 置备器的注册

第二种自动注册新创建的虚拟机的方法是使用 Terraform 的 `remote-exec` 置备器。

`remote-exec` 置备器会与新创建的虚拟机进行交互。它会建立 SSH 连接，然后便可在该虚拟机上运行命令。



使用 `remote-exec` 置备器注册客户端时，用户必须确保运行 Terraform 的虚拟机在新虚拟机创建后可以访问该虚拟机。

其他要求与使用 [基于 `cloud-init` 的客户端注册](#) 时相同：

- 引导脚本。有关详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。
- 激活密钥。有关详细信息，请参见 [Client-configuration > Activation-keys](#)。

下面的命令会下载引导脚本并在新虚拟机创建后注册该虚拟机。应将此命令定义为要运行的远程命令：

```
curl -s http://hub-server.tf.local/pub/bootstrap/bootstrap-default.sh |  
bash -s
```

有关 `remote-exec` 置备器的详细信息，请参见 <https://www.terraform.io/docs/provisioners/remote-exec.html>。

# Chapter 5. 客户端升级

客户端采用底层操作系统的版本控制系统，需要定期升级。

对于运行 SUSE 操作系统且已在 SCC 中注册的客户端，可在 Uyuni Web UI 中进行升级。有关支持的 SUSE Linux Enterprise 15 升级路径，请参见 <https://documentation.suse.com/sles/15-SP4/html/SLES-all/cha-upgrade-paths.html>

将运行 SLE 12 的客户端升级到 SLE 15 的过程可自动完成，不过在开始前您需要完成一些准备步骤。有关详细信息，请参见 **Client-configuration** > **Client-upgrades-major**。

您也可以使用内容生命周期管理器自动执行客户端升级。有关详细信息，请参见 **Client-configuration** > **Client-upgrades-lifecycle**。

有关产品迁移（例如服务包升级、openSUSE Leap 次要版本升级或将 openSUSE Leap 迁移到 SUSE Linux Enterprise）的详细信息，请参见 **Client-configuration** > **Client-upgrades-product-migration**。

有关激活密钥的详细信息，请参见 **Client-configuration** > **Activation-keys**。

## 5.1. 主要版本升级

客户端必须有所安装操作系统的最新可用服务包 (SP) 且已应用所有最新更新。开始前，请确保系统是最新的，并且已成功安装所有更新。

升级由 YaST 和 AutoYaST 控制，该过程不使用 Zypper。

### 5.1.1. 准备迁移

需要先完成以下步骤，才能将客户端从 SLE 12 迁移到 SLE 15：

1. 准备安装媒体
2. 声明自动安装发行套件
3. 创建激活密钥
4. 创建自动安装配置文件

#### 5.1.1.1. 准备安装媒体

过程：准备安装媒体

1. 在容器主机上，下载包含安装源的 ISO 映像。
2. 使用 `mgradm` 从 ISO 映像导入安装数据：

```
mgradm distribution copy <映像名称>.iso <映像名称>
```

3. 记下 `mgradm` 报告的发行套件路径。向 Uyuni 声明发行套件时需要提供该路径。



此映像可用于多个自动安装发行套件。

有关详细信息，请参见 **Client-configuration > Autoinst-distributions**。

### 5.1.1.2. 声明自动安装发行套件

过程：声明自动安装发行套件

1. 在 Uyuni Web UI 中，导航到 **系统 > 自动安装 > 发行套件**。
2. 单击 **创建可自动安装的发行套件**，并填写以下字段：

在 **名称** 字段中，输入用于标识自动安装发行套件的名称。例如 `sles15sp6-x86_64`。  
 在 **安装媒体** 字段中，输入已保存的安装媒体的路径。  
 \* 选择匹配的 **安装媒体**。所选值必须与安装媒体相匹配。  
 \* 选择 **引导内核选项**。所选值必须与安装媒体相匹配。  
 \* 可选：指定引导此发行套件时要使用的内核选项。可以通过多种方式来提供内核选项。此处仅添加了发行套件通用的选项。

3. 单击 **创建可自动安装的发行套件**。

### 5.1.1.3. 创建激活密钥

例如，要从旧 SLE 12 基础通道切换到新的 SLE 15 通道，需要有激活密钥。

过程：创建激活密钥

1. 在 Uyuni 服务器 Web UI 中，导航到 **系统 > 激活密钥**，然后单击 **创建激活密钥**。
2. 输入密钥说明。
3. 输入密钥或将其留空以生成自动密钥。
4. 可选：如果您要限制使用次数，请在 **限制次数** 文本字段中输入值。
5. 选择 `SLE-Product-SLES15-SP6-Pool for x86_64` 基础通道。
6. 可选：选择任何 **通道**。有关详细信息，请参见 <https://documentation.suse.com/sles/15-SP4/html/SLES-all/article-modules.html>。
7. 单击 **创建激活密钥**。
8. 单击 **通道** 选项卡，然后选择所需通道。
9. 单击 **更新密钥**。

### 5.1.1.4. 创建自动安装配置文件

自动安装配置文件包含安装系统所需的所有安装和配置数据，还包含在完成安装后需要执行的脚本。例如，可用作着手点的脚本，请参见 <https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaST>。

有关有效的 AutoYaST 升级设置，请参见 <https://doc.opensuse.org/projects/autoyast/#CreateProfile-upgrade>。

过程：创建自动安装配置文件

- 在 Uyuni Web UI 中，导航到系统，自动安装，配置文件，然后上载自动安装配置文件脚本。

有关可用作着手点的示例脚本，请参见：

<https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaST>。

- 在 `autoupgrade` 字段中，键入 `autoupgrade=1`。

(可选) 您也可以包含 `Y2DEBUG=1` 选项。调试设置不是必需的设置，不过此设置有助于对您未来可能遇到的任何问题进行查错。



在 Azure 云中运行的客户端必须将 `textmode=1 console=ttyS0` 添加到 `autoupgrade` 中。

- 粘贴自动安装配置文件或使用文件上载字段。
- 单击 **[ 创建 ]** 保存设置。
- 如果需要为上载的配置文件设置变量，请导航到系统，自动安装，配置文件，选择要编辑的配置文件，然后导航到 `Variables` 选项卡。

使用以下格式指定所需的变量：

```
<key>=<value>
```

## 5.1.2. 迁移

在开始前，需检查自动安装配置文件中引用的所有通道是否可用并已完全同步。

您可以在 `/var/log/rhn/reposync/<channel-label>.log` 中监控镜像进度。

过程：迁移

- 在 Uyuni 服务器 Web UI 中，导航到 `Agents`，然后选择要升级的客户端。
- 导航到 `Agents` 选项卡，然后选择上载的自动安装配置文件。
- 单击 **[ 安排 | 自动 | 安装 | 并 | 完成 ]**。系统会下载所需的文件，更改引导加载程序项，重引导并开始升级。

客户端下次与 Uyuni 服务器同步时，会收到重新安装作业。重新安装作业会提取新内核和 `initrd` 软件包，还会写入新的 `/boot/grub/menu.lst` (GRUB Legacy) 或 `/boot/grub2/grub.cfg` (GRUB 2)，包含指向新内核和 `initrd` 软件包的指针。

客户端下次引导时，会使用 `grub` 来引导新内核及其 `initrd`。此过程中不会使用 PXE 引导。

提取作业约 3 分钟后，客户端会关闭以重引导。



对于客户端，请在迁移完成后使用 `spacewalk/minion_script` 代码段再次注册客户端。

## 5.2. 使用内容生命周期管理器升级

如果您需要管理的 SUSE Linux Enterprise Server 客户端有很多，可以使用内容生命周期管理器自动执行就地升级。

### 5.2.1. 准备升级

升级客户端之前，需要完成以下准备工作：

- 创建内容生命周期项目
- 创建激活密钥
- 创建可自动安装的发行套件
- 创建自动安装配置文件

过程：创建内容生命周期项目

1. 为您的发行套件创建内容生命周期项目。

有关详细信息，请参见 **Administration > Content-lifecycle**。

2. 务必为您的项目选择简短的非描述性名称。
3. 添加您的发行套件所需的所有源通道模块。
4. 视需要添加过滤器，并至少设置一个环境。

过程：创建激活密钥

1. 为您的发行套件创建激活密钥。

有关详细信息，请参见 **Client-configuration > Activation-keys**。

2. 确保您的激活密钥包含所有过滤出的项目通道。

过程：创建可自动安装的发行套件

1. 为要迁移的每个基础通道创建可自动安装的发行套件。

有关详细信息，请参见 **Client-configuration > Autoinst-distributions**。

2. 为您的发行套件指定一个标签来表示内容生命周期项目的名称。
3. 在 `-----` 字段中，选择要使用的 SLES 版本。

过程：创建自动安装配置文件

1. 为要升级到的每个目标发行套件和服务包创建自动安装配置文件。

有关详细信息，请参见 [Client-configuration > Autoinst-profiles](#)。

2. 您可以在配置文件中使用变量来区分不同的生命周期环境。

有关自动安装配置文件的示例，请参见 <https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaST>。

在自动安装配置文件中使用以下变量来实现自动就地升级：

列表 1. 例如：在自动安装配置文件中使用的变量

```
registration_key=1-15sp1-demo-test
org=1
channel_prefix=15sp1-demo-test
distro_label=15sp1-demo-test
```

列表 2. 例如：在自动安装配置文件中使用的项

```
<listentry>
    <ask_on_error config:type="boolean">true</ask_on_error>

    <media_url>https://$redhat_management_server/ks/dist/child/$channel_prefix-sle-module-web-scripting15-sp1-pool-x86_64/$distro_label</media_url>
        <name>$channel_prefix SLE-Module-Web-Scripting15-SP1 Pool for x86_64
    </name>
        <product>Web Scripting Module 15 SP1 x86_64 Pool</product>
    </listentry>
```

## 5.2.2. 升级

准备好要升级的服务器后，您便可置备客户端。

过程：置备客户端

1. 在 Uyuni Web UI 中，导航到 **系统**，**系统列表**，选择要置备的客户端，以将其添加到系统集管理器中。
2. 导航到 **系统**，**系统集管理器**，**概览**，然后单击 选项卡。
3. 选择要使用的自动安装配置文件。

对于可以使用 PXE 的客户端，当您完成置备后，客户端便会立即自动迁移。对于所有其他客户端，您可以使用 Cobbler 来执行升级。

过程：使用 Cobbler 升级客户端

1. 在命令提示符处，以 root 身份运行以下命令，以访问容器内的外壳：

```
mgrctl term
```

1. 查看可用的 Cobbler 配置文件：

```
cobbler profile list
```

2. 使用您选择的配置文件和发行套件构建 ISO 文件：

```
cobbler buildiso --iso=/tmp/SLE_15-sp1.iso --profiles=SLE_15
-sp1:1:Example --distro=SLE_15-sp1
```

有关使用 CD-ROM 置备客户端的详细信息，请参见 [Client-configuration > Autoinst-cdrom](#)。

## 5.3. 产品迁移

Product migration allows you to upgrade SLE-based client systems from an Service Pack (SP) level to a later one. You can also upgrade clients such as AlmaLinux, CentOS, Oracle Linux, or Rocky Linux to SUSE Liberty Linux. For example, you can migrate SUSE Linux Enterprise Server 15 SP1 to SUSE Linux Enterprise Server 15 SP2.

产品迁移用于在同一主要版本内升级。您无法通过产品迁移从 SUSE Linux Enterprise Server 12 迁移到 SUSE Linux Enterprise Server 15。有关主要升级的详细信息，请参见 [Client-configuration > Client-upgrades-major](#)。

您还可以将 openSUSE Leap 迁移到更高的次要版本或对应的 SUSE Linux Enterprise Server SP 级别，例如：

- 从 openSUSE Leap 15.1 迁移到 15.2
- 从 openSUSE Leap 15.1 迁移到 SUSE Linux Enterprise Server 15 SP1
- 将 openSUSE Leap 15.5 迁移到 SUSE Linux Enterprise Server 15 SP5

在 SUSE Linux Enterprise Server 12 及更高版本中，SUSE 支持跳过服务包（如果 SUSE Customer Center 提供该服务包）。例如，您可以从 SUSE Linux Enterprise Server 15 升级到 SP2，而无需安装 SP1。

有关支持的 SUSE Linux Enterprise Server 升级路径，请参见 <https://documentation.suse.com/en-us/sles/15-SP4/html/SLES-all/cha-upgrade-paths.html#sec-upgrade-paths-supported>。



迁移期间，Uyuni 会在安装前自动接受任何必要的许可协议 (EULA)。

### 5.3.1. 单系统迁移

开始迁移产品之前：

- 确保没有待处理更新或补丁。请检查客户端系统**细节** > **概览**页面上的，然后安装提供的所有更新或补丁。如果您的客户端系统不是最新的，产品迁移可能会失败。
- 确保目标产品的所有通道都已完全同步。要在 Web UI 中检查同步状态，请导航到**管理** > **安装向导** > **产品**页面。
- 请确保您具有有效的系统备份，以防万一。产品迁移没有回滚功能。迁移过程一旦开始，就无法回滚。

过程：执行单系统迁移

- 从**系统** > **概览**页面中，选择一个客户端。
- 从客户端的系统细节页面，导航到**软件** > **产品迁移**选项卡。
- 选择目标迁移路径，然后单击 **[选择通道]**。
- 从**基础通道** - **迁移**页面选择正确的基础通道，包括**迁移通道**和任何其他**通道**。
- 可选：选中**允许安装供应商发生更改的软件包**，以允许安装供应商发生更改的软件包。如果发生此情况，开始迁移前会显示一条包含相关细节的通知。



要将 openSUSE Leap 迁移到 SUSE Linux Enterprise Server，必须选中**迁移通道**选项。

- 正确配置您的通道后，单击 **[安排迁移]**。

## 5.3.2. 产品大量迁移

如果您要将大量客户端迁移到下一 SP 版本，可以使用 Uyuni API 调用。

`spacecmd` 命令行工具提供了 `system_scheduleproductmigration` 子命令，可用于安排将大量客户端迁移到下一次要版本。

### 5.3.2.1. 执行产品大量迁移



产品大量迁移操作非常危险，该过程应经过全面测试。至少应先进行试运行。

请务必小心，不要无意中升级系统。

过程：执行产品大量迁移

- 列出可用的迁移目标，并记下要迁移的系统 ID：

```
spacecmd api -- system.listMigrationTargets -A 1000010001
```

- 针对每个系统 ID 调用 `listMigrationTarget`，并查看所需的目标产品是否可用。

- 如果系统 ID 有可用目标，则调用 `system.scheduleProductMigration`。
- 如果所需目标不可用，则跳过该系统。

3. 根据您的环境调整以下模板：

```

target = '[....]'
basechannel = 'channel-label'
system_ids = [1, 2, 3]

session = auth.login(user, pass)
for system in system_ids
    if system.listMigrationTargets(session, system).ident == target
        system.scheduleProductMigration(session, system, target,
basechannel, [] , False, <now>)
    else
        print "无法迁移到请求的目标 -- 正在跳过系统"
    endif
endfor

```

### 5.3.2.2. 示例：从 SLES 15 SP2 迁移到 SLES 15 SP3

对于此示例，将会创建一个组，以便于进行大量迁移。

过程：创建大量产品迁移组

1. 在 Uyuni Web UI 中，导航到 **系统** > **系统组**，然后单击 **[ 创建组 ]**。
2. 将该组命名为 `mpm-target-sles15sp3`。
  - 您应该仅将订阅了相同基础通道的系统添加到创建的组中。本示例只会将订阅了 `SLE-Product-SLES15-SP2-Pool for x86_64` 的系统添加到该组中。

有关将客户端添加到组的详细信息，请参见 [Client-configuration > System-groups](#)。

过程：将系统添加到组

1. 运行以下命令，以获取该组中所有系统的 ID：

```
spacecmd -- system_listmigrationtargets group:mpm-target-sles15sp3
```

2. 命令的输出为由多个 ID 组成的字符串。
  - 请只选择一个目标，这样为\*所有\*系统报告的就是该目标。
  - 该字符串是另一个命令的 MIGRATIONTARGET 的标识符。



spacecmd 的子命令 system\_scheduleproductmigration 和 system\_listmigrationtargets 会遍历该组中的所有系统。

如果该组中有 100 个系统，您会发现安排了 100 个操作。

该组中的所有系统都必须支持相同的迁移目标。

过程：运行大量迁移命令

1. system\_scheduleproductmigration 命令的语法如下：

```
spacecmd -- system_scheduleproductmigration <SYSTEM>
<BASE_CHANNEL_LABEL> \
<MIGRATION_TARGET> [选项]
```

2. 对于此示例，要将 mpm-target-sles15sp3 组中的所有系统从 SLES 12 SP2 升级到 SLES 15 SP3，请在命令行上输入以下命令：

```
spacecmd -- system_scheduleproductmigration group:mpm-target-
sles15sp3 \
sle-product-sles15-sp3-pool-x86_64 "[190,203,195,1242]" -d
```

### 5.3.2.2.1. 命令语法说明

要查看 system\_scheduleproductmigration 的语法和选项，请运行以下命令：

```
spacecmd system_scheduleproductmigration help
```

#### <SYSTEM>

对于此示例，我们将使用我们创建的组选择该组中的所有系统：

```
group:mpm-target-sles15sp3
```

#### <BASE\_CHANNEL\_LABEL>

这是目标基础通道的标签。在本例中，系统将升级到 SLES 15 SP3，且标签为 sle-product-sles15-sp3-pool-x86\_64。

要查看当前镜像的所有基础通道的列表，请运行以下命令：

```
spacecmd softwarechannel_listbasechannels
```

请注意，除非通道是当前基础通道的可用目标，否则您无法升级到该通道。

#### <MIGRATION\_TARGET>

要为 group:mpm-target-sles15sp3 组中的系统确定此值，请运行以下命令：

```
spacecmd -- system_listmigrationtargets group:mpm-target-sles15sp3
```

必须采用以下格式传递 MIGRATION\_TARGET 参数；请注意需要使用引号将其括住以防中括号会产生问题：  
：

```
"[190,203,195,1242]"
```

#### 选项

- -s START\_TIME
- -d 如果您要执行试运行，请传递此标志（建议在实际执行迁移前执行试运行）
- -c CHILD\_CHANNELS（逗号分隔的子通道标签（不带空格））

在本例中，我们包含了 -d 选项，在成功执行试运行后可以将其去除。

如果成功，安排的每个系统的命令输出将如下所示：

```
正在为以下系统安排产品迁移 mpm-sles152-1  
已安排操作 ID: 66
```

可以在 Web UI 中跟踪该组中某个指定系统的操作，在本例中即为试运行。从客户端的系统细节页面导航到事件，历史。如果试运行期间出现任何失败，应对系统进行调查。

如果一切正常，则可从命令中去除 -d 选项，以运行实际的迁移。完成迁移后，可以从 Uyuni Web UI 中重引导系统。

## 5.4. 升级 Uyuni 客户端

在本节中，我们以 openSUSE Leap 为例。

### 5.4.1. 准备升级

过程：准备客户端升级

1. 在 Uyuni 服务器上的命令提示符处，以 root 身份使用 spacewalk-common-channels 命令添加相应的通道。

```
spacewalk-common-channels \
opensuse_leap15_4 \
opensuse_leap15_4-non-oss \
opensuse_leap15_4-non-oss-updates \
opensuse_leap15_4-updates \
opensuse_leap15_4-uyuni-client
```

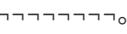
2. Fully synchronize all channels with spacewalk-repo-sync.
3. 在 Uyuni 服务器 Web UI 中，导航到软件 > 管理 > 通道，然后单击 Uyuni Client Tools for openSUSE Leap 15.4 (x86\_64) 通道名称。
4. 在右上角单击 。
5. 单击  选项卡，然后选择  - Uyuni Client Tools for openSUSE Leap 15.3 (x86\_64)。
6. 单击 。
7. 导航到 储存库 > 同步子选项卡，然后单击 。
8. 对 openSUSE Leap 15.4 (x86\_64) 和  执行相同的操作。

展开 openSUSE Leap 15.4 (x86\_64) 以查看填充了软件包的所有子通道。

## 5.4.2. 升级

要升级客户端，您需要替换软件储存库，然后更新软件，最后重引导客户端。

过程：升级客户端

1. 在 Uyuni 服务器 Web UI 中，导航到系统，然后单击客户端的名称。
2. 单击 软件 > 软件通道，选择  列表中所列的 openSUSE Leap 15.5 通道作为基础通道。
3. 在  窗格中，选择 15.5 子通道。
4. 单击 ，然后单击  以 .
5. 单击 ，选择客户端上要更新的所有软件包，然后应用选择。单击 ，查看细节，然后单击  以完成更新。
6. 重引导客户端。

如果需要更新许多客户端，可以在 Uyuni 服务器上创建由此命令序列组成的操作链。您可以使用操作链同时对多个客户端执行更新。

# Chapter 6. 客户端删除

如果需要从 Uyuni 服务器中去除客户端，您可以：

- 使用 Web UI 将其删除
- 从命令行去除客户端。

## 6.1. 使用 Web UI 删除客户端

过程：删除客户端

1. 在 Uyuni Web UI 中，导航到 **系统 > 系统列表**，然后选择要删除的客户端。
2. 单击 **[删除]**。
3. 检查细节并单击 **[删除]** 配置文件确认。
4. 对于 Salt 客户端，Uyuni 会尝试清理其他配置。如果无法联系客户端，您可以选择取消删除，或者仅删除客户端而不清理配置文件。

还可以使用系统集管理器删除多个客户端。有关系统集管理器的详细信息，请参见 [Client-configuration > System-set-manager](#)。



清理客户端只会禁用 Salt 并停止服务（如果可能）。此操作不会卸装软件包。

## 6.2. 在命令行上删除客户端（通过 API 调用）

过程：从服务器上删除客户端

1. 使用 FQDN（完全限定的域名）删除客户端：

```
spacecmd system_delete FQDN
```

spacecmd system\_delete 还可删除 Salt 密钥。

system\_delete 提供以下选项：

```
usage: system_delete [选项] <系统>
```

选项：

- c TYPE - 可能的值：
  - \* 'FAIL\_ON\_CLEANUP\_ERR' - 如果发生清理错误则失败，
  - \* 'NO\_CLEANUP' - 不清理，仅删除，
  - \* 'FORCE\_DELETE' - 先尝试清理，但在发生错误时仍会删除服务器

## 6.3. 从命令行删除客户端



此过程仅适用于 Uyuni 客户端。请勿在 Uyuni 服务器上运行。



请勿在运行 Red Hat Enterprise Linux、Debian 或克隆的客户端上执行以下过程，而是应使用与 zypper 效果相同的打包器命令，例如 yum、dnf 或 apt。

过程：删除 SLES 12 和 15 客户端

1. 停止 salt-minion 服务：

```
systemctl stop salt-minion
```

2. 去除储存库和配置文件：

```
rm /etc/zypp/repos.d/susemanager\channels.repo  
rm -r /etc/sysconfig/rhn/  
rm -r /etc/salt/
```

3. 去除客户端软件包：

```
zypper rm salt salt-minion python*-salt sle-manager-tools-release
```

过程：Salt 捆绑包客户端 - 手动进行注册清理

1. 要取消注册，请运行以下命令：

```
systemctl stop venv-salt-minion  
zypper rm -y venv-salt-minion  
rm /etc/zypp/repos.d/susemanager\channels.repo /etc/venv-salt-  
minion/*  
rm -r /etc/venv-salt-minion/*
```

有关 Salt 捆绑包的信息，请参见 [Client-configuration > Contact-methods-saltbundle](#)。

# Chapter 7. 客户端操作

除了注册、升级、安装软件或删除客户端外，还可以执行其他操作。可以逐个管理 Uyuni 客户端，也可以对一系列客户端进行统一管理。可以使用系统集管理器将一系列客户端分组在一起以便一次性地对其执行操作，或者使用系统组将其永久分组在一起。

对于高级客户端配置选项，可以使用常规配置管理。可以使用 Uyuni Web UI 获取自定义系统信息、开机或关机，以及重引导客户端。

以下章节包含其中每项操作的详细说明。

## 7.1. 软件包管理

客户端使用软件包来安装、卸装和升级软件。



- 安装或升级软件包时，将自动接受许可证或 EULA。

要管理客户端上的软件包，请导航到 ，单击要管理的客户端，然后导航到 **系统** > **软件** > **软件包** 子选项卡。此部分提供的选项取决于您选择的客户端类型及其当前的通道订阅。

大多数软件包管理操作都可以添加到操作链中。有关操作链的详细信息，请参见 **Reference** > **Schedule**。

### 7.1.1. 使用配置文件比较软件包

您可以将某个客户端上安装的软件包与存储的配置文件进行比较，或者与其他客户端上安装的软件包进行比较。比较时，您可以选择修改选定要匹配的客户端。

将软件包与某个配置文件进行比较之前，您需要先存储一个配置文件。配置文件基于当前安装的客户端上的软件包创建。如果已创建配置文件，您可以使用它安装更多客户端，这些客户端中均会安装相同的软件包。

过程：创建存储的配置文件

1. 在 Uyuni Web UI 中，导航到 ，单击配置文件要基于的客户端，然后导航到 **系统** > **软件** > **软件包** > **配置文件** 子选项卡。
2. 单击 **[创建]** **[系]** **[统]** **[配]** **[置]** **[文]** **[件]**。
3. 键入您的配置文件的名称和说明，然后单击 **[创]** **[建]** **[配]** **[置]** **[文]** **[件]**。

过程：比较客户端软件包

1. 在 Uyuni Web UI 中，导航到 ，单击要比较的客户端，然后导航到 **系统** > **软件** > **安装包** > **配置文件** 子选项卡。要与存储的配置文件进行比较，请选择该配置文件，然后单击 **[比]** **[较]**。
2. 要与其他客户端进行比较，请选择客户端名称，然后单击 **[比]** **[较]** 查看两个客户端之间的差别的列表。

### 7.1.2. Orphaned Packages

Orphaned packages are packages synchronized by Uyuni that are not associated with a software channel. Therefore, orphaned packages are usually not available to Uyuni clients and users cannot

install such packages without additional effort.

A package can become orphaned as a result of one of the following events:

- A synchronized repository removes a package. By default, Uyuni 5.0 unlinks such package from the channel, but does not remove the package. In Uyuni 4.3, you can enable this behavior by using the `strict mode` channel configuration.
- A repository supersedes a package with a newer version and thus removes the previous version.
- The reposync process got interrupted (for example, because of an out of space exception) and thus downloaded packages were not associated with a channel.
- Users manually uploaded packages, but did not associate them with any channel.

Orphaned packages cost space in user environment and cannot be easily distributed to clients because they are not associated with a channel. Orphaned package can make sense with specific workflows like bootstrapping or customer-specific procedures.

You can view orphaned packages in the Web UI by clicking **Software > Manage > Packages > view Packages in no channel**

Uyuni periodically runs a Taskomatic job that searches for orphaned packages and modifies the package organization ID to 1. This means that you can delete orphaned packages only after the Taskomatic job execution. If you encounter an orphaned package that you cannot delete, wait 24 hours and try the deletion again.

You can deleted orphaned packages:

- with commandline tools targeting individual packages. For example:

```
spacecmd package_remove zypper-1.14.52-150400.1.9.x86_64
```

- with commandline tools targeting all orphaned packages at once:

```
spacecmd package_removeorphans
```

## 7.2. 补丁管理

您可以在组织中使用自定义补丁来管理客户端。通过这种方式，您可以在自定义通道中发布软件包的补丁警报、安排补丁安装，并在组织中管理补丁。

### 7.2.1. 创建补丁

要使用自定义补丁，您需要创建一个补丁，向其添加软件包，并将其添加到一个或多个通道。

过程：创建自定义补丁

1. 在 Uyuni Web UI 中，导航到**补丁**，**管理补丁**，单击  **[创|建|补|丁]**。
2. 在部分，使用以下细节：
  - 在字段中，键入补丁的简要说明。
  - 在字段中，键入补丁的标签。建议您为组织设计一个命名约定，以便于进行补丁管理。
  - 在字段中，输入补丁的版本号。例如，如果这是该补丁的第一个版本，请使用 1。
  - 在字段中，选择要使用的补丁类型。例如，针对修复错误的补丁选择 **Bug**。
  - 如果您选择建议类型，请在字段中选择要使用的严重性级别。
  - 在字段中，键入该补丁所代表的产品的名称。
  - 可选：在字段中，键入补丁作者的名称。
  - 在、和字段中填写补丁的其他相关信息。
3. 可选：在 Bug 部分，使用以下细节指定任何相关 Bug 的信息：
  - 在 ID 字段中，输入 Bug 编号。
  - 在字段中，键入 Bug 的简要说明。
  - 在 Bugzilla URL 字段中，键入 Bug 的地址。
  - 在字段中，键入与该 Bug 有关的任何关键字。请在各关键字之间使用逗号。
  - 在和字段中填写 Bug 的其他相关信息。
  - 选择要向其添加新补丁的一个或多个通道。
4. 单击  **[创|建|补|丁]**。

您也可以通过克隆现有补丁来创建补丁。克隆方法会保留软件包关联并简化补丁发布过程。

过程：克隆补丁

1. 在 Uyuni Web UI 中，导航到**补丁**，**克隆补丁**。
2. 在字段中，选择要克隆的补丁对应的软件通道。
3. 选择要克隆的一个或多个补丁，然后单击  **[克|隆|补|丁]**。
4. 选择要向其添加克隆的补丁的一个或多个通道。
5. 确认细节以开始克隆。

创建补丁后，您便可以向其指派软件包。

过程：向补丁指派软件包

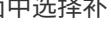
1. 在 Uyuni Web UI 中，导航到**补丁**，**管理补丁**，然后单击补丁的建议名称以查看补丁细节。
2. 导航到**软件包**，**添加**选项卡。
3. 在字段中，选择包含要指派给补丁的软件包的软件通道，然后单击  **[查|看|软|件|包]**。您可以选择以查看所有通道中的可用软件包。

4. 选择要包含的软件包，然后单击 。
5. 确认软件包的细节，然后单击  将其指派给补丁。
6. 导航到 **软件包** > **列出/去除** 选项卡，检查是否已正确指派软件包。

将软件包指派给补丁后，补丁缓存即会更新以反映更改。更新缓存可能需要几分钟时间。

如果您需要更改现有补丁的细节，可以通过  页面来更改。

过程：编辑和删除现有补丁警报

1. 在 Uyuni Web UI 中，导航到 **补丁** > **管理补丁**。
2. 单击补丁的建议名称以查看补丁细节。
3. 根据需要进行更改，然后单击 。
4. 要删除补丁，请在  页面中选择补丁，然后单击 。删除补丁可能需要几分钟时间。

## 7.2.2. 将补丁应用到客户端

某个补丁就绪后，您可以将其单独应用到客户端，也可以与其他补丁一起应用到客户端。

补丁内的每个软件包都来自一个或多个通道。如果客户端未订阅相应通道，将不会安装更新。

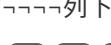
如果客户端已安装更新版本的软件包，将不会安装更新。如果客户端安装的是较旧版本的软件包，将会升级该软件包。

过程：应用所有适用的补丁

1. 在 Uyuni Web UI 中，导航到 **系统** > **概览**，然后选择要更新的客户端。
2. 导航到 **软件** > **补丁** 选项卡。
3. 单击  选择所有适用的补丁。
4. 单击  更新客户端。

如果您是以管理员特权登录的，还可以批量进行更大规模的客户端升级。

过程：将单个补丁应用到多个客户端

1. 在 Uyuni Web UI 中，导航到 **补丁** > **补丁列表**。
2. 找到要应用的补丁，然后单击该补丁对应的  列下面的数字。
3. 选择要应用补丁的客户端，然后单击 。
4. 确认要执行更新的客户端列表。

过程：将多个补丁应用到多个客户端

1. 在 Uyuni Web UI 中，导航到 **系统** > **概览**，然后选中要更新的客户端，以将其添加到系统集管理器。
2. 导航到 **系统** > **系统集管理器**，然后导航到  选项卡。

3. 选择要应用到客户端的补丁，然后单击 **[应用补丁]**。

4. 安排更新的日期和时间，然后单击 **[确认]**。

5. 要查看更新进度，请导航到**日程安排**，待执行的操作。



使用为每个客户端配置的联系方法安装安排的软件包更新。有关详细信息，请参见 **Client-configuration > Contact-methods-intro**。

## 7.3. 系统锁定

系统锁用于防止对客户端执行某些操作。例如，系统锁可防止更新或重启客户端。此功能对于运行生产软件的客户端或想要阻止意外更改时很有用。您可以在准备好执行操作时禁用系统锁。

### 7.3.1. 客户端上的系统锁

当客户端被锁定或置于中断模式后，便无法安排操作，并会禁止执行命令，而且-----页面上会显示一个黄色标题。在此模式下，虽然可以使用 Web UI 或 API 为锁定的客户端安排操作，但操作会失败。



锁定机制不适用于 Salt SSH 客户端。

过程：对客户端执行系统锁定

1. 在 Uyuni Web UI 中，导航到要锁定的客户端的-----页面。

2. 导航到-----选项卡，选中系统锁公式，然后单击 **[保存]**。

3. 导航到**公式系统锁**选项卡，选中-----，然后单击 **[保存]**。在此页面上，您也可以在客户端处于锁定状态时启用特定的 Salt 模块。

4. 进行更改后，您可能需要应用 highstate。在此情况下，Web UI 中会显示一个标题通知您。客户端会保持锁定状态，直到您去除系统锁公式。

有关 Salt 中的中断模式的详细信息，请参见 <https://docs.saltstack.com/en/latest/topics/blackout/index.html>。

### 7.3.2. 软件包锁

可以在多个客户端上使用软件包锁定，但可用的功能集不同。必须在 SUSE Linux Enterprise 和 openSUSE（基于 zypp）客户端与 Red Hat Enterprise Linux 或 Debian 客户端之间进行区分。

#### 7.3.2.1. 基于 Zypp 的系统上的软件包锁

软件包锁用于防止在未经授权的情况下安装或升级软件包。当软件包被锁定后，会显示一个挂锁图标，表示无法安装该软件包。任何尝试安装锁定软件包的操作均会在事件日志中报告为错误。

您无法安装、升级或去除锁定的软件包，无论是通过 Uyuni Web UI 还是使用软件包管理器在客户端计算机上直接执行均是如此。锁定的软件包还会间接锁定任何依赖的软件包。

过程：使用软件包锁

1. 在受管系统上导航到软件，软件包，锁定选项卡查看所有可用软件包的列表。
2. 选择要锁定的软件包，然后 **[请求] [锁定]**。选择要激活软件包锁的日期和时间。默认情况下，软件包锁会尽快激活。请注意，软件包锁可能不会立即激活。
3. 要去除软件包锁，请选择要解锁的软件包，然后 **[请求] [解锁]**。就像激活软件包锁一样选择日期和时间。

### 7.3.2.2. 类似 Red Hat Enterprise Linux 和 Debian 的系统上的软件包锁



一些类似 Red Hat Enterprise Linux 和 Debian 的系统可对客户端进行软件包锁定。

在类似 Red Hat Enterprise Linux 和 Debian 的系统上，软件包锁仅用于防止在未经授权的情况下升级或去除软件包。当软件包被锁定后，会显示一个挂锁图标，表示无法更改该软件包。任何尝试更改锁定软件包的操作均会在事件日志中报告为错误。

您无法升级或去除锁定的软件包，无论是通过 Uyuni Web UI 还是使用软件包管理器在客户端计算机上直接执行均是如此。锁定的软件包还会间接锁定任何依赖的软件包。

过程：使用软件包锁

1. 在 Red Hat Enterprise Linux 7 系统和 Red Hat Enterprise Linux 8 或 9 系统上，分别以 `root` 身份安装 `yum-plugin-versionlock` 和 `python3-dnf-plugin-versionlock`。在 Debian 系统上，`apt` 工具已包含锁定功能。
2. 在受管系统上导航到软件，软件包，锁定选项卡查看所有可用软件包的列表。
3. 选择要锁定的软件包，然后 **[请求] [锁定]**。选择要激活软件包锁的日期和时间。默认情况下，软件包锁会尽快激活。请注意，软件包锁可能不会立即激活。
4. 要去除软件包锁，请选择要解锁的软件包，然后 **[请求] [解锁]**。就像激活软件包锁一样选择日期和时间。

## 7.4. 配置管理

您可以使用配置文件和通道管理客户端的配置，而无需手动配置每个客户端。

配置参数编写在脚本中并存储在配置文件中。您可以使用 Uyuni Web UI 直接写入配置文件，也可以上载或关联到其他位置存在的文件。

可以集中管理配置文件。集中管理的配置文件由全局配置通道提供，并可应用于任何订阅了 Uyuni 服务器的客户端。

配置通道用于组织配置文件。您可以为客户端订阅配置通道，并根据需要部署配置文件。

配置文件有版本控制，因此您可以添加配置设置，在您的客户端上对其进行测试，并根据需要将其回滚到之前的修订版。如果您已创建配置通道，还可以在不同的配置文件以及同一配置文件的不同修订版之间进行比较。

集中管理的配置文件由全局配置通道提供。

下表显示了支持的功能，其中：

- ✓ 功能受 SUSE 支持
- ✗ 功能不受 SUSE 支持
- ? 功能正在考虑之中，日后可能受支持，也可能不受支持。

表格 47. 配置管理支持的功能

功能	状态
全局配置通道	✓
部署文件	✓
比较文件	?
本地管理的文件	✓ (结合 Salt 功能)
沙箱文件	✗
应用 Highstate	✓
从客户端导入文件	✗
建立 Jinja 模板	✓
配置宏	✓ (结合 Salt 功能：grain、pillar 数据等)

## 7.4.1. 创建配置通道

### 7.4.1.1. 中心配置通道

要创建新的中心配置通道，请执行以下操作：

过程：创建中心配置通道

- 在 Uyuni Web UI 中，导航到配置>通道，然后单击 **[创建配置通道]**。
- 键入通道的名称。
- 键入通道的标签。此字段只能包含字母、数字、连字符 (-) 和下划线 (\_)。
- 键入用于与其他通道进行区分的通道说明。
- 单击 **[创建配置通道]** 以创建新通道。

### 7.4.1.2. Salt 状态通道

还可以使用配置通道来管理客户端上的 Salt 状态：

过程：创建 Salt 状态通道

- 在 Uyuni Web UI 中，导航到配置>通道，然后单击 **[创建状态通道]**。
- 键入通道的名称。
- 键入通道的标签。此字段只能包含字母、数字、连字符 (-) 和下划线 (\_)。
- 键入用于与其他通道进行区分的通道说明。

5. 为 init.sls 文件键入 SLS 代码。
6. 单击  以创建新通道。

## 7.4.2. 添加配置文件、目录或符号链接

如果您已创建配置通道，则可以添加配置文件、目录或符号链接：

过程：添加配置文件、目录或符号链接

1. 在 Uyuni Web UI 中，导航到 **配置 > 通道**，单击要向其添加配置文件的配置通道的名称，然后导航到 **添加文件 > 创建文件** 子选项卡。
2. 在 **类型** 字段中，选择是要创建文本文件、目录还是符号链接。
3. 在 **路径** / **部署** 字段中，键入应部署文件的位置的绝对路径。
4. 如果您要创建符号链接，请在 **目标** / **路径** 字段中键入目标文件和路径。
5. 在 **权限** 字段中键入该文件的 **所有者** 和 **组**，以及 **权限**。
6. 如果客户端启用了 SELinux，您可以配置 SELinux **策略** 以启用所需的文件属性（例如用户、角色和文件类型）。
7. 如果配置文件包含宏，请输入标记宏开始位置和结束位置的符号。
8. 在 **脚本语言** 文本框中，使用脚本下拉框选择相应的脚本语言输入配置文件内容。
9. 单击 。

## 7.4.3. 为客户端订阅配置通道

您可以为单个客户端订阅配置通道，方法是导航到 **系统 > 系统列表**，选择要使用的客户端，然后导航到 **订阅** 选项卡。要为多个客户端订阅配置通道，可以使用系统集管理器 (SSM)。

过程：为多个客户端订阅配置通道

1. 在 Uyuni Web UI 中，导航到 **系统 > 系统列表**，然后选择要使用的客户端。
2. 导航到 **系统 > 系统集管理器**，然后转到 **配置 > 订阅通道** 子选项卡，以查看可用配置通道的列表。
3. 可选：单击 **订阅** 列中的数字以查看目前订阅了配置通道的客户端。
4. 选中要订阅的配置通道，然后单击 。
5. 使用向上和向下箭头对配置通道排名。两个配置通道之间的设置发生冲突时，系统会优先使用更靠近列表顶部的通道。
6. 确定通道应用到所选客户端的方式：  
 - **订阅为最低优先**：将新通道添加为比目前订阅的通道都低的优先级。  
 - **订阅为最高优先**：将新通道添加为比目前订阅的通道都高的优先级。  
 - **替换现有订阅**：可去除现有通道并将其替换为新通道。
7. 单击 。



如果新配置通道优先级与现有通道冲突，系统会去除重复的通道并根据新优先级进行替换。如果某个操作要对客户端的配置优先级重新排序，Web UI 会要求您在继续之前先确认更改。

#### 7.4.4. 比较配置文件

您也可以使用系统集管理器 (SSM) 将客户端上部署的配置文件与存储在 Uyuni 服务器上的配置文件进行比较。

过程：比较配置文件

1. 在 Uyuni Web UI 中，导航到 **系统**，**系统列表**，然后选择订阅了要比较的配置文件的客户端。
2. 导航到 **系统**，**系统集管理器**，然后转到 **配置**，**比较文件**子选项卡，以查看可用配置文件的列表。
3. 可选：单击列中的数字以查看目前订阅了配置文件的客户端。
4. 选中要比较的配置文件，然后单击 **[安排] [文件] [比较]**。

#### 7.4.5. 在客户端上建立 Jinja 模板

在 Salt 客户端上可以建立 Jinja 模板。Jinja 通过 pillar 或 grain 提供变量。它们可用于配置文件或 Salt 状态中。

有关详细信息，请参见 <https://docs.saltproject.io/salt/user-guide/en/latest/topics/jinja.html>，该页面包含以下示例：

```
{% if grains.os_family == 'RedHat' %}
  {% set dns_cfg = '/etc/named.conf' %}
{% elif grains.os_family == 'Debian' %}
  {% set dns_cfg = '/etc/bind/named.conf' %}
{% else %}
  {% set dns_cfg = '/etc/named.conf' %}
{% endif %}
dns_conf:
  file.managed:
    - name: {{ dns_cfg }}
    - source: salt://dns/files/named.conf
```

### 7.5. 电源管理

您可以使用 Uyuni Web UI 打开、关闭和重引导客户端。

此功能使用 IPMI 或 Redfish 协议，通过 Cobbler 配置文件管理。所选客户端的电源管理控制器必须支持以上其中一个协议。

对于 Redfish，请确保您可以在客户端和 Uyuni 服务器之间建立有效的 SSL 连接。您必须信任用于对 Redfish 管理控制器的 SSL 服务器证书签名的证书颁发机构。CA 证书必须为 .pem 格式，并存储在 Uyuni 服务器上的

/etc/pki/trust/anchors/ 中。保存证书后，请运行 update-ca-certificate。

过程：启用电源管理

1. 在 Uyuni Web UI 中，导航到 **系统**，**系统列表**，选择要管理的客户端，然后导航到 **置备**，**电源管理** 选项卡。
2. 在 **协议** 字段中，选择要使用的电源管理协议。
3. 填写电源管理服务器的细节，然后单击要执行的操作的相应按钮，或者单击 **仅保存** 保存细节而不执行任何操作。

您可以将电源管理操作添加到系统集管理器，以将这些操作同时应用到多个客户端。有关使用系统集管理器的详细信息，请参见 **Client-configuration > System-set-manager**。

### 7.5.1. 电源管理和 Cobbler

首次使用电源管理功能时，如果客户端没有相应的 Cobbler 系统记录，将会自动创建该记录。这些自动创建的系统记录无法从网络上引导，其内包含虚设系统映像的引用。之所以需要该引用，是因为 Cobbler 目前不支持没有配置文件或映像的系统记录。

Cobbler 电源管理使用 fence-agent 工具来支持 IPMI 以外的协议。Uyuni 只支持 IPMI 和 Redfish 协议。您可以将客户端配置为使用其他协议，方法是以逗号分隔列表形式将 fence- agent 名称添加到 rhn.conf 配置文件中的 java.power\_management.types 配置参数。

## 7.6. 自定义系统信息

您可以包含有关您的客户端的自定义信息。系统信息定义为 **--:--** 对，您可将其指派给客户端。例如，您可以为某个特定的处理器定义一个 **--:--**，然后将该键指派给安装了该处理器的所有客户端。系统会对自定义系统信息分类，您可以使用 Uyuni Web UI 搜索该信息。

开始前，需要创建可让您存储自定义信息的键。

过程：创建自定义系统信息键

1. 在 Uyuni Web UI 中，导航到 **系统**，**自定义系统信息**，然后单击 **[创建键]**。
2. 在 **键** 字段中，添加键的名称。不要使用空格。例如，intel-x86\_64-quadcore。
3. 在 **值** 字段中，提供任何其他所需信息。
4. 对需要创建的每个键重复以上步骤。

可以通过 Salt pillar 获取此信息。可以使用如下命令检索此信息：

```
salt $minionid pillar.get custom_info:key1
```

此命令将产生类似如下的输出：

```
$minionid:  
val1
```

创建了一些自定义系统信息键之后，您便可以将其应用到客户端。

过程：将自定义信息键应用到客户端

1. 在 Uyuni Web UI 中，导航到 **客户端**，单击要应用自定义信息的客户端，然后导航到 **细节**，**自定义信息** 选项卡。
2. 单击 。
3. 找到要应用的值，然后单击键标签。
4. 在 **值** 字段中，提供任何其他信息。
5. 单击  以向客户端应用自定义信息。

有关配置管理的详细信息，请参见 **Client-configuration > Configuration-management**。

## 7.7. 系统集管理器

系统集管理器 (SSM) 可用于同时对多个客户端执行操作。SSM 会创建多组临时客户端，将它们变为可用状态，以执行您需要应用到多个客户端的一次性操作。如果您需要存留时间更长的客户端组，请考虑使用系统组。有关系统组的详细信息，请参见 **Client-configuration > System-groups**。

下表列出了可在 SSM 中使用的操作。此表中图标的含意如下：

-  在 SSM 中，此操作可用于此客户端类型
-  在 SSM 中，此操作不可用于此客户端类型
-  正在考虑将此操作用于此客户端类型，日后可能支持此操作，也可能不支持此操作。

表格 48. 可用的 SSM 操作

列出系统	
安装补丁	
安排补丁更新	
升级软件包	
安装软件包	
去除软件包	
校验软件包	
创建组	
管理组	
通道成员资格	

列出系统	✓
通道订阅	✗
部署/比较通道	✗
自动安装客户端	✗
快照标记	✗
远程命令	✗
电源管理	✗
更新系统首选项	✓
更新硬件配置文件	✓
更新软件包配置文件	✓
设置/去除自定义值	✓
重引导客户端	✓
将客户端迁移到另一个组织中	✓
删除客户端	✓

您可以通过多种方式选择 SSM 的客户端：

- 导航到**系统**，**系统列表**，然后选中要使用的客户端。
- 导航到**系统**，**系统组**，然后针对要使用的系统组单击**[在 SSM 中使用]**。
- 导航到**系统**，**系统组**，选中要使用的组，然后单击**[使用组]**。

选择要使用的客户端后，导航到**系统**，**系统集管理器**，或单击顶部菜单栏中的-----图标。



SSM 中的细节与 Uyuni Web UI 其他部分中的细节略有不同。在 SSM 中，所有可用的更新都会显示。这样您可以升级到可能不是最新版本的软件包。

## 7.7.1. 在 SSM 中更改基础通道

您可以使用 SSM 同时更改多个客户端的基础通道。



更改基础通道会大量更改可用于受影响客户端的软件包和补丁。须谨慎使用该功能。

过程：使用 SSM 更改多个客户端的基础通道

1. 在 Uyuni Web UI 中，导航到**系统**，**系统列表**，选中要使用的客户端，然后导航到**系统**，**系统集管理器**。
2. 导航到-----子选项卡。
3. 在列表中找到当前的基础通道，并校验-----列中显示的数字是否正确。 您可以单击此列中的数字查看要更改的客户端的更多细节。
4. 在-----字段中选择新基础通道，然后单击**[下一页]**。

5. 对于每个子通道，选择“`-----`”、“`---`”或“`-----`”，然后单击 。
6. 选中要进行的更改，然后选择希望操作执行的时间。
7. 单击  安排更改。

## 7.8. 系统组

通过系统组，您可以更轻松地管理大量客户端。组可用于对客户端执行批量操作，例如应用更新、配置通道、Salt 状态或公式。

您可以采用任何适合环境的方式来对客户端分组。例如，可以按客户端上安装的操作系统、所在的实际位置或所处理的工作负载类型来组织客户端。客户端可以划分到任意数量的组中，因此您可以采用不同的方式来定义组。

对客户端分组后，您可以在一个或多个组中的所有客户端上或不同组之间的交集客户端上执行更新。例如，可为所有包含 Web 服务器软件的客户端定义一个组，为所有 SLES 客户端定义另一个组。然后，可对所有包含 Web 服务器软件的客户端执行更新，或者使用组之间的交集来更新所有包含 Web 服务器软件的 SLES 客户端。

### 7.8.1. 创建组

在使用组对客户端分组之前，需要先创建一些组。

过程：创建新系统组

1. 在 Uyuni Web UI 中，导航到 **系统 > 系统组**。
2. 单击 。
3. 指定新组的名称和说明。
4. 单击  以保存组。
5. 对需要创建的每个组重复以上步骤。

### 7.8.2. 将客户端添加到组

您可以向组中添加单个客户端，也可以同时添加多个客户端。

过程：将单个客户端添加到组

1. 在 Uyuni Web UI 中，导航到 **系统 > 系统列表**，然后单击要添加的客户端的名称。
2. 导航到 **组 > 加入** 选项卡。
3. 选中要加入的组，然后单击 。

过程：将多个客户端添加到组

1. 在 Uyuni Web UI 中，导航到 **系统 > 系统列表**，然后单击要向其添加客户端的组的名称。
2. 导航到 **-----** 选项卡。
3. 选中要添加的客户端，然后单击 。

过程：使用 SSM 将多个客户端添加到组

1. 在 Uyuni Web UI 中，导航到**系统** > **系统列表**，然后选中要添加的每个客户端，这会将这些客户端添加到系统集管理器中。
2. 导航到**系统** > **系统集管理器**，然后转到--选项卡。
3. 找到要加入的组并选中----。
4. 单击 **更改成员资格**。
5. 单击 **确认** 将客户端加入所选的组。

有关系统集管理器的详细信息，请参见 **Client-configuration** > **System-set-manager**。

您可以导航到**系统** > **系统组**并单击某个组的名称，然后导航到----选项卡，以查看该组中有哪些客户端。

### 7.8.3. 使用组

将客户端分组后，可以使用组来管理更新。

在 Uyuni Web UI 中，导航到**系统** > **系统组**。如果组中的任何客户端有可用更新，列表会显示相应图标。如果组中有任何客户端的检查更新状态功能被禁用，列表会显示问号。单击图标可查看有关可用更新的详细信息，并向客户端应用更新。

您也可以同时使用多个组。选择要使用的多个组，然后单击 **使用并集** 以选择每个选定组中的每个客户端。

或者，可以使用组的交集。选择两个或更多组，然后单击 **使用交集** 仅选择所有选定组中都存在的那些客户端。例如，可为所有包含 Web 服务器软件的客户端定义一个组，为所有 SLES 客户端定义另一个组。这些组的交集是所有包含 Web 服务器软件的 SLES 客户端。

## 7.9. 系统类型

客户端按系统类型分类。每个客户端都可以被指派一个基础系统类型和一个附加系统类型。

所有客户端的基础系统类型为 Salt。

附加系统类型包括----（作为虚拟主机运行的客户端）和----（作为构建主机运行的客户端）。

您可以导航到**系统** > **系统列表** > **系统类型**调整附加系统类型。选中要更改其附加系统类型的客户端，选择----，然后单击 **添加系统类型** 或 **去除系统类型**。

# Chapter 8. 操作系统安装

通常，您注册的是已经在运行的客户端。您可能是在将这些计算机注册到 Uyuni 之前手动安装了它们，也可能它们是在您将 Uyuni 添加到您的环境之前就已安装的现有系统。

您也可以使用 Uyuni 一次性完成安装操作系统和将其注册到 Uyuni 的操作。此方法部分或完全自动化，因此可为您节省回答安装程序问题的时间，特别适合您有许多客户端需要安装和注册的情况。

可通过多种方法从 Uyuni 安装操作系统：

- 在已注册的客户端上就地安装；
- 使用 PXE 引导通过网络安装；
- 准备安装 CD-ROM 或 USB 密钥，然后转到计算机在该媒体上进行引导；
- 作为 Uyuni for Retail 解决方案的一部分安装。

就地安装方法假设客户端上已经安装了以前的操作系统，并且客户端已注册到 Uyuni。

有关就地安装方法的信息，请参见 **Client-configuration > Autoinst-reinstall**。

网络引导安装方法适用于未格式化的计算机。不过，该方法只能在特定网络配置中执行：

- Uyuni 服务器（或其代理之一）与您要安装的计算机位于同一本地网络中，或者您具有可用于跨越两者之间的所有路由器的 DHCP 中继；
- 您能够设置新的 DHCP 服务器或配置现有的 DHCP 服务器；
- 要安装的客户端能够使用 PXE 引导，并且您可以如此配置它们。

可移动媒体方法可让您绕过这些网络约束。但是，该方法假设计算机能够读取 CD-ROM 或 USB 密钥，并从它们引导。它还需要对客户端计算机进行物理访问。

有关可移动媒体方法的信息，请参见 **Client-configuration > Autoinst-cdrom**。

有关 Uyuni for Retail 方法的信息，请参见 **Retail > Retail-overview**。



不支持自动安装 Ubuntu 和 Debian 客户端。必须手动安装这些操作系统。

Uyuni 的自动安装功能基于名为 Cobbler 的软件运行。有关 Cobbler 的详细信息，请访问 <https://cobbler.readthedocs.io>。



SUSE 仅支持 Uyuni Web UI 中或通过 Uyuni API 提供的 Cobbler 功能。Cobbler 支持的唯一一个命令行命令为 buildiso。本文中所述的仅为支持的功能。

## 8.1. 重新安装已注册系统

就地重新安装从本地客户端系统开始。因此，客户端不需要具有使用 PXE 通过网络引导的能力。

要就地重新安装已注册客户端，必须定义可自动安装的发行套件和自动安装配置文件。有关信息，请参见 [Client-configuration > Autoinst-distributions](#) 和 [Client-configuration > Autoinst-profiles](#)。

定义自动安装配置文件和发行套件后，便可执行重新安装。

过程：重新安装已注册客户端

1. 在 Uyuni Web UI 中，导航到 **系统**，**系统列表**，选择要重新安装的客户端，然后转到 **置备**，**自动重新安装**，**日程安排**子选项卡。
2. 选择您准备的自动安装配置文件，根据需要选择代理，然后单击 **[安排自动安装并完成]**。
3. 您可以导航到 **置备**，**自动安装**，**会话状态**来监视安装进度，也可以直接在客户端上监视。客户端将重引导，并在引导菜单中选择名为 `reinstall-system` 的新选项。

安装随即会通过 HTTP 协议进行。

## 8.2. 通过 CD-ROM 或 USB 密钥安装

对于尚未注册到 Uyuni 的客户端，如果无法通过 PXE 进行网络引导，可以使用可引导 CD-ROM 或 USB 密钥来安装系统。

准备此类可移动媒体的其中一个方法是使用 Cobbler。有关使用 Cobbler 准备 ISO 映像的信息，请参见[使用 Cobbler 构建 ISO 映像](#)。

对于 SUSE 系统，通常建议使用 KIWI 来准备 ISO 映像。有关详细信息，请参见[使用 KIWI 构建 SUSE ISO 映像](#)。

在所有情况下，您都需要使用生成的映像刻录 CD-ROM 或准备 USB 密钥。

### 8.2.1. 使用 Cobbler 构建 ISO 映像

Cobbler 可创建包含一组发行套件、内核和菜单的 ISO 引导映像。该映像的工作方式与 PXE 安装类似。



IBM Z 上不支持使用 Cobbler 构建 ISO。

与通过 PXE 进行网络引导类似，为了使用 Cobbler 准备 ISO 映像，您需要准备发行套件和配置文件。有关创建发行套件的信息，请参见 [Client-configuration > Autoinst-distributions](#)。有关创建配置文件的信息，请参见 [Client-configuration > Autoinst-profiles](#)。

Cobbler `buildiso` 命令接受用于定义引导 ISO 的名称和输出位置的参数。运行 `buildiso` 命令时，必须使用 `--distro` 指定发行套件。

```
cobbler buildiso --iso=/path/to/boot.iso --distro=<发行套件标签>
```



必须按 Cobbler 中所列的名称（而不是简单地按 UI 中所示）使用发行套件和配置文件标签。

要列出 Cobbler 存储的发行套件和配置文件的名称，请运行以下命令：

```
# cobbler distro list
# cobbler profile list
```

引导 ISO 默认包含所有配置文件和系统。您可以通过 `--profiles` 和 `--systems` 选项来限制使用的配置文件和系统。例如：

```
cobbler buildiso --systems="系统 1 系统 2 系统 3" \
--profiles=<配置文件 1 标签> <配置文件 2 标签> <配置文件 3 标签> --
distro=<发行套件标签>
```



如果您无法将 ISO 映像写入到公共 tmp 目录,请检查 /usr/lib/systemd/system/cobblerd.service 中的 systemd 设置。

## 8.2.2. 使用 KIWI 构建 SUSE ISO 映像

KIWI 是一个映像创建系统。您可以使用 KIWI 创建供目标系统用于安装 SUSE 系统的可引导 ISO 映像。重引导或打开系统时，它会从映像引导，从 Uyuni 装载 AutoYaST 配置，并根据 AutoYaST 配置文件安装 SUSE Linux Enterprise Server。

要使用 ISO 映像，请引导系统并在提示符处键入 `autoyast`（假设您将 AutoYaST 引导的标签保留为 `autoyast`）。按 `kbd: [Enter]` 开始 AutoYaST 安装。

有关 KIWI 的详细信息，请参见 <http://doc.opensuse.org/projects/kiwi/doc/>。

## 8.2.3. 使用 Cobbler 构建 Red Hat ISO 映像

有关详细信息，请参见 [client-configuration:autoinst-cdrom.pdf](#)。

## 8.3. 可自动安装的发行套件

自动安装过程依赖于几个文件来启动安装。这些文件包括 Linux 内核、初始 RAM 磁盘和在安装模式下引导操作系统所需的其他文件。

Uyuni 使用 `mgradm` 工具将安装文件从源复制到服务器容器。

您可以从 DVD 映像中提取所需的文件。有关信息，请参见 [基于 ISO 映像的发行套件](#)。

或者，您也可以安装 `tftpboot-installation` 软件包。有关信息，请参见 [基于 RPM 软件包的发行套件](#)。

对于与这些文件相同的操作系统版本，您还必须在 Uyuni 服务器上同步基础通道。

当您准备好文件并已同步基础通道时，您需要声明发行套件。此操作会将安装文件关联到基础通道。发行套件

可能会由一个或多个安装配置文件引用。有关信息，请参见[声明可自动安装的发行套件](#)。

### 8.3.1. 基于 ISO 映像的发行套件

此方法假设您有要在客户端上安装的操作系统的安装媒体。这通常是 DVD .iso 映像，其中包含 Linux 内核、initrd 文件和在安装模式下引导操作系统所需的其他文件。

过程：从安装媒体导入文件

1. 使用 `mgradm` 从 ISO 映像导入安装数据：

```
# mgradm distribution copy <映像名称>.iso <映像名称>
```

2. 记下 `mgradm` 报告的发行套件路径。向 Uyuni 声明发行套件时需要提供该路径。

#### 8.3.1.1. 发行套件自动检测和注册

`mgradm` 能够自动检测发行套件名称并将其注册到服务器。提供的 ISO 映像必须包含 `.treeinfo` 文件。

过程：导入发行套件文件并执行自动检测和注册

1. 使用 `mgradm`：

```
# mgradm distribution copy --api-user <用户名> --api-password <口令>
<映像名称>.iso
```

### 8.3.2. 基于 RPM 软件包的发行套件

此方法适用于 SUSE 系统。它比从安装媒体导入内容更简单，因为它使用的是安装系统的预打包文件。

过程：从安装软件包提取文件

1. 在 Uyuni 服务器上，安装名称以 `tftpboot-installation` 开头的软件包。您可以通过 `zypper se tftpboot-installation` 命令确定它的确切名称
2. 可使用以下命令将软件包安装到其他根目录，以避免重启动：

```
# mkdir /opt/tftpinstall
# zypper --installroot /opt/tftpinstall install tftpboot-
installation-SLE-Micro-5.5-x86_64
```

3. 使用命令 `ls -d /opt/tftpinstall/usr/share/tftpboot-installation/*` 查找安装文件。

4. 使用 `mgradm` 复制安装文件：

```
# mgradm distribution copy /opt/tftpinstall/usr/share/tftpboot-
installation/SLE-Micro-5.5-x86_64 SLE-Micro-5.5-x86_64
```

5. 记下 `mgradm` 工具报告的发行套件路径。向 Uyuni 声明发行套件时需要提供该路径。

6. `mgradm` 工具运行完成后，您可以去除 `/opt/tftpinstall` 目录。

此过程将准备安装与驱动 Uyuni 服务器的操作系统版本相同的操作系统版本。如果您想在客户端上安装不同的操作系统或版本，需要从其所属的发行套件手动获取 `tftpboot-installation-*` 软件包。在 Uyuni 的输入框中，搜索名称以 `tftpboot-installation` 开头的软件包，然后查看软件包的细节。`/var/spacewalk/` 下会显示本地路径。

### 8.3.3. 声明可自动安装的发行套件

提取自动安装文件后，接下来要声明可自动安装的发行套件。

过程：声明可自动安装的发行套件

1. 在 Uyuni Web UI 中，导航到 **系统** > **自动安装** > **发行套件**。
2. 单击，并填写以下字段：
  - 在 `名称` 字段中，输入用于识别可自动安装的发行套件的名称。
  - 在 `安装媒体` 字段中，输入保存在 Uyuni 服务器上的安装媒体的路径。
  - 选择匹配的 `安装媒体`。所选值必须与安装媒体相匹配。
  - 选择 `内核选项`。所选值必须与安装媒体相匹配。
  - 可选：指定引导此发行套件时要使用的内核选项。您可以通过多种方式来提供内核选项。此处仅添加了发行套件通用的选项。
3. 单击 **[创建可自动安装的发行套件]**。

您准备的安装文件可能不包含需要安装的软件包。如果这些软件包未包含在内，请将 `useonlinerepo=1` 添加到 `内核选项` 字段。

软件包储存库包含的元数据有可能未签名。如果元数据未签名，请将 `insecure=1` 添加到 `内核选项` 字段，或者使用您自己的 GPG 密钥（如 **Client-configuration** > **Autoinst-ownpgpkey** 中所述）。

有些情况下需要这些内核选项，例如，当您使用“联机安装程序”ISO 映像而非完整的 DVD 时，或者当您使用 `tftpboot-installation` 软件包时。

导航到 **系统** > **自动安装** > **发行套件** 可管理您的可自动安装的发行套件。

### 8.3.4. Handling Kernel Options for Distributions and Profiles

Uyuni is able to combine the kernel options that you assign. This is done using a special inheritance logic. There are three object types that are relevant for this:

1. Distributions (or short "Distros")
2. Profiles
3. Systems

As a fourth special point of influence on the final kernel options is the Cobbler settings file `/etc/cobbler/settings.yaml`. The Cobbler settings file defines default kernel options for all distributions. This is not supported in the context of Uyuni.

Understanding raw and resolved values is crucial for managing kernel options effectively.

1. **Raw values:** These refer to values attached directly to a specific Cobbler item and stored as-is in Cobbler's internal database.
2. **Resolved values:** These values are dynamically generated at runtime, respecting the inheritance hierarchy of Cobbler items.

If you prefix an option with `!` then the option will be removed from the final kernel command line.

Uyuni will manage the kernel options for both Profiles and Systems for you. As such you may only edit the kernel options for Distros.

### 8.3.4.1. Examples

#### 8.3.4.1.1. Basic Inheritance Example

Distribution raw value

```
install=http://uyuni.server/ks/dist/SLES15SP4 self_update=0
```

Profile raw value

```
console=tty1
```

System raw value

```
console=ttyS0
```

**Resolved** value for a system inheriting this profile

```
install=http://uyuni.server/ks/dist/SLES15SP4 self_update=0 console=ttyS0
```

### 8.3.4.1.2. Option Removal Example

Distribution raw value

```
install=http://uyuni.server/ks/dist/SLES15SP4 self_update=0
```

Profile raw value

```
console=tty1
```

System raw value

```
!self_update
```

**Resolved** value for a system inheriting this profile

```
install=http://uyuni.server/ks/dist/SLES15SP4 console=ttyS0
```

## 8.4. 自动安装配置文件

自动安装配置文件决定了如何安装操作系统。例如，您可以指定更多要传递给安装程序的内核参数。

配置文件最重要的部分是“自动安装文件”。手动执行安装时，必须向安装程序提供相关信息，例如分区和网络信息以及用户细节。您可以使用自动安装文件以脚本形式提供此类信息。这种类型的文件有时也称为“答案文件”。

在 Uyuni 中，您可以使用两种不同的配置文件，具体取决于您要安装的客户端的操作系统：

- 对于 SUSE Linux Enterprise 或 openSUSE 客户端，请使用 AutoYaST。
- 对于 Red Hat Enterprise Linux 客户端，请使用 Kickstart。

如果您要安装具有不同操作系统的客户端，可以使用 AutoYaST 和 Kickstart 配置文件。

- 有关如何声明配置文件的信息，请参见[声明配置文件](#)
- 有关 AutoYaST 配置文件的信息，请参见[AutoYaST 配置文件](#)。
- 有关 Kickstart 配置文件的信息，请参见[Kickstart 配置文件](#)。

配置文件中包含的自动安装文件可以包含变量和代码段。有关变量和代码段的信息，请参见[模板语法](#)。

## 8.4.1. 声明配置文件

准备好自动安装文件和发行套件后，您可以创建配置文件来管理 Uyuni 服务器上的自动安装。配置文件会决定如何安装您选择的此发行套件。创建配置文件的其中一个方式是上载 AutoYaST 或 Kickstart 文件。或者，您也可以使用 Web UI 向导（仅针对 Kickstart）。

过程：通过上载创建自动安装配置文件

1. 在 Uyuni Web UI 中，导航到系统，自动安装，配置文件。
2. 单击 上 载 Kickstart/AutoYaST 文件。
3. 在 字段中，键入配置文件的名称。不要使用空格。
4. 在 字段中，选择要用于此配置文件的可自动安装的发行套件。
5. 在 字段中，选择要用于此配置文件的虚拟化类型，或选择 （如果您不想使用此配置文件创建新的虚拟机）。
6. 将自动安装文件的内容复制到 字段，或使用 字段直接上载文件。

有关此处要包含的细节的详细信息，请参见 [AutoYaST 配置文件](#) 或 [Kickstart 配置文件](#)。

7. 单击 以创建配置文件。

过程：通过向导创建 Kickstart 配置文件

1. 在 Uyuni Web UI 中，导航到系统，自动安装，配置文件。
2. 单击 配置文件。
3. 在 字段中，键入配置文件的名称。不要使用空格。
4. 在 字段中，选择要用于此配置文件的基础通道。系统会根据可用的发行套件填充此字段。如果您需要的基础通道不可用，请检查您是否正确创建了发行套件。
5. 在 字段中，选择要用于此配置文件的虚拟化类型，或选择 （表示不进行虚拟化）。
6. 单击 。
7. 在 中，键入 Uyuni 服务器上安装的安装媒体的路径。
8. 单击 。
9. 提供客户端上 root 用户的口令。
10. 单击 。
11. 查看新配置文件的细节，并视需要进行自定义。

创建自动安装配置文件时，您可以选中 。此设置可让 Uyuni 自动提取与指定基础通道关联的最新发行套件。如果您以后添加新发行套件，Uyuni 会使用最近创建或修改的发行套件。

更改 通常需要更改配置文件加载程序和分区选项。这会重写您的自定义设置。请在保存前导航到 选项卡校验新的或更改后的设置。

来自发行套件和配置文件的内核选项会进行合并。

您可以更改自动安装配置文件的细节和设置，方法是导航到 **系统** > **自动安装** > **配置文件**，然后单击要编辑的配置文件的名称。或者，可导航到 **系统** > **系统列表**，选择要置备的客户端，然后导航到 **置备** > **自动安装子选项卡**。

## 8.4.2. AutoYaST 配置文件

AutoYaST 配置文件由标识该配置文件的——、指向可自动安装的发行套件的————、各种选项以及最重要的 AutoYaST 安装文件组成。

AutoYaST 安装文件是一个 XML 文件，用于向 AutoYaST 安装程序提供指示。AutoYaST 将其称为“控制文件”。有关 AutoYaST 安装文件的完整语法，请参见 <https://doc.opensuse.org/projects/autoyast/#cha-configuration-installation-options>。

SUSE 提供了 AutoYaST 安装文件模板，您可以基于它们创建自己的自定义文件。您可在 <https://github.com/SUSE/manager-build-profiles> 上的 AutoYast 目录中找到这些模板。使用前，需要为其中的每个配置文件设置一些变量。请参见脚本随附的 README 文件确定您需要的变量。有关在 AutoYaST 脚本中使用变量的详细信息，请参见 [变量](#)。

在 AutoYaST 安装文件中，用于通过 Uyuni 安装的最重要的部分如下：

- <add-on> 用于向安装添加子通道。有关示例，请参见 <https://doc.opensuse.org/projects/autoyast/#Software-Selections-additional>。
- <general>\$SNIPPET('spacewalk/sles\_no\_signature\_checks')</general> 禁用签名检查
- <software> 用于为 Unified Installer 指定产品
  - 请参见 <https://doc.opensuse.org/projects/autoyast/#CreateProfile-Software>，其中包括“<software>”示例
- <init-scripts config:type="list">\$SNIPPET('spacewalk/minion\_script')</init-scripts> 用于将客户端作为 Salt 客户端注册到 Uyuni。

有关 AutoYaST 的详细信息，请参见 <https://doc.opensuse.org/projects/autoyast/>。

近期一个基于 Salt 的方案 Yomi 可替代 AutoYaST。有关 Yomi 的信息，请参见 [Specialized-guides > Salt](#)。

## 8.4.3. Kickstart 配置文件

Kickstart 配置文件提供了大量配置选项。要创建这些配置文件，您可以上载它们，也可以使用专用向导。

Kickstart 配置文件允许您使用文件保留列表。如果您有许多自定义配置文件位于要使用 Kickstart 重新安装的客户端上，则可以将它们保存为列表，并将该列表与 Kickstart 配置文件相关联。

过程：创建文件保留列表

1. 在 Uyuni Web UI 中，导航到 **系统** > **自动安装** > **文件保留**，然后单击 **[创|建|文|件|保|留|列|表]**。
2. 输入适当的标签，并列出要保存的所有文件和目录的绝对路径。
3. 单击 **[创|建|列|表]**。

4. 将该文件保留列表包含到 Kickstart 配置文件中。
5. 导航到系统，自动安装，配置文件，选择要编辑的配置文件，转到系统细节，文件保留子选项卡，然后选择要包含的文件保留列表。



文件保留列表总大小上限为 1 MB。/dev/hda1 和 /dev/sda1 等特殊设备无法保留。只能使用文件名和目录名，不能使用正则表达式通配符。

有关 Kickstart 的详细信息，请参见 Red Hat 文档。

#### 8.4.4. 模板语法

安装文件的部分内容在安装过程中会被替换。变量会被替换为相应的单个值，代码段会被替换为整段文本。转义符或转义部分不会被替换。

名为 Cheetah 的模板引擎允许 Cobbler 进行这些替换。利用此机制，您在重新安装大量系统时就无需为每个系统手动创建配置文件。

您可以在 Uyuni Web UI 中创建自动安装变量和代码段。在配置文件中，Variables 选项卡可让您查看替换的结果。

- 有关变量的信息，请参见[变量](#)。
- 有关代码段的信息，请参见[代码段](#)。
- 有关转义符或文本块的信息，请参见[转义](#)。

##### 8.4.4.1. 变量

可以使用自动安装变量来替换 Kickstart 和 AutoYaST 配置文件中的值。要定义变量，请从配置文件中导航到 Variables 子选项卡，然后在文本框中创建 name=value 对。

例如，您可以创建一个变量来存储客户端的 IP 地址，创建另一个变量来存储其网关的地址。然后可以为通过同一配置文件安装的所有客户端定义这些变量。要执行此操作，请将下面几行添加到 Variables 文本框中：

```
ipaddr=192.168.0.28
gateway=192.168.0.1
```

要使用变量，请在配置文件中附加一个 \$ 符号来替换相应值。例如，Kickstart 文件的 network 部分可能如下所示：

```
network --bootproto=static --device=eth0 --onboot=on --ip=$ipaddr \
--gateway=$gateway
```

\$ipaddr 会解析为 192.168.0.28，\$gateway 会解析为 192.168.0.1。

在安装文件中，变量使用层次结构。系统变量优先于配置文件变量，而配置文件变量优先于发行套件变量。

### 8.4.4.2. 代码段

Uyuni 附带了大量预定义的代码段。导航到 **系统** > **自动安装** > **自动安装代码段**，以查看现有代码段列表。

通过在自动安装文件中插入 `$SNIPPET()` 宏来使用代码段。例如，在 Kickstart 中，插入该宏：

```
$SNIPPET('spacewalk/rhel_register_script')
```

或者，在 AutoYaST 中，插入该宏：

```
<init-scripts config:type="list">
  $SNIPPET('spacewalk/sles_register_script')
</init-scripts>
```

宏会经过 Cobbler 的分析，并会替换为代码段的内容。您还可以存储自己的代码段，以供日后在自动安装文件中使用。单击  **创建代码段** 以创建新代码段。

下面的示例设置了通用硬盘分区配置的 Kickstart 代码段：

```
clearpart --all
part /boot --fstype ext3 --size=150 --asprimary
part / --fstype ext3 --size=40000 --asprimary
part swap --recommended

part pv.00 --size=1 --grow

volgroup vg00 pv.00
logvol /var --name=var vgname=vg00 --fstype ext3 --size=5000
```

下面是使用该代码段的一个示例：

```
$SNIPPET('my_partition')
```

### 8.4.4.3. 转义

如果自动安装文件包含 `$example` 这样的外壳脚本变量，则需要使用反斜杠对内容进行转义：`\$example`。对 `$` 符号进行转义可防止模板引擎将该符号评估为内部变量。

可以使用 `\#raw` 和 `\#end raw` 指令封装代码段或脚本等文本块，从而对其转义。例如：

```
#raw
#!/bin/bash
for i in {0..2}; do
    echo "$i - Hello World!"
done
#end raw
```

所有包含 # 符号后接空格的行均视为注释行，因此不会对其进行评估。例如：

```
# start some section (此为注释)
echo "Hello, world"
# end some section (此为注释)
```

## 8.5. 无人照管的置备

您可以使用 API 调用来声明通过 MAC 地址标识的客户端与自动安装配置文件之间的关联。系统下次重引导时，会根据指定的配置文件开始安装。

过程：通过手动声明的配置文件重新安装

1. 在 Uyuni 服务器上的命令提示符处，使用 `system.createSystemRecord` API 调用。在此示例中，请将 `name` 替换为您的客户端名称，将 `<profile>` 替换为配置文件标签，将 `<iface>` 替换为客户端上的接口名称（例如 `eth0`），将 `<hw_addr>` 替换为其硬件地址（例如 `00:25:22:71:e7:c6`）：

```
$ spacecmd api -- --args '[ "<name>", "<profile>", "", "", \
[ {"name": "<iface>", "mac": "<hw_addr>"} ]' \
system.createSystemRecord
```

2. 打开客户端。它会从网络进行引导，并且系统会选择正确的配置文件以执行安装。

此命令会在 Cobbler 中创建一个系统记录。您也可以指定其他参数，例如内核选项、客户端的 IP 地址及其域名。有关详细信息，请参见 `createSystemRecord` 的 API 文档。

## 8.6. 使用您自己的 GPG 密钥

如果用于自动安装的储存库包含未签名的元数据，则您通常必须使用 `insecure=1` 内核参数作为可自动安装的发行套件的选项，并在 AutoYaST 安装文件中使用 `spacewalk/sles_no_signature_checks` 代码段。

更为安全的替代方案是提供您自己的 GPG 密钥。



此方法仅适用于 SUSE 客户端。

过程：包含您自己的 GPG 密钥

1. 创建 GPG 密钥。
2. 使用它对软件包的元数据签名。
3. 将其添加到安装媒体的初始 RAM 磁盘中。
4. 要将 GPG 密钥复制到容器，请使用 `mgradm gpg add`。此命令会复制 GPG 密钥并将其添加到客户密钥环中。
5. 运行命令 `mgradm restart`。此命令会使用 SUSE 和客户密钥环来创建一个唯一的密钥环。



在您使用新 GPG 密钥对元数据签名时，任何已进行初始配置的客户端都不会知道这个新密钥。理想情况下，您应当在注册任何客户端之前对元数据签名。

如果一些已初始配置的客户端使用的是这些储存库，可以暂时对它们禁用 GPG 密钥检查。

# Chapter 9. 虚拟化

可以使用 Uyuni 管理虚拟化客户端。在这种安装中，会在 Uyuni 服务器上安装一个虚拟主机来管理任意数量的虚拟 Guest。可以安装多个虚拟主机来管理多组 Guest（如果您选择这么做）。

虚拟化客户端具有的功能范围取决于您选择的第三方虚拟化提供者。

您可以直接在 Uyuni 中管理 Xen 和 KVM 主机及 Guest。采用这种方式，您可以使用 AutoYaST 或 Kickstart 自动安装主机和 Guest，并在 Web UI 中管理 Guest。

For VMware, including VMware vSphere, and Nutanix AHV, Uyuni requires you to set up a virtual host manager (VHM) to control the VMs. This gives you control over the hosts and guests, but in a more limited way than available with Xen and KVM; Uyuni cannot create or edit VMs on VMware vSphere or Nutanix AHV.

Uyuni 不直接支持其他第三方虚拟化提供者。不过，如果您的提供者允许您导出 VM 的 JSON 配置文件，您可以将该配置文件上载到 Uyuni 并使用 VHM 进行管理。

有关使用 VHM 来管理虚拟化的详细信息，请参见 [Client-configuration > Vhm](#)。

## 9.1. 管理虚拟化主机

开始前，请确保已为要用作虚拟化主机的客户端指派系统类型。请导航到 **系统 > 系统列表**，然后单击要用作虚拟化主机的客户端的名称。如果系统类型未列出，请初始化公式。有关详细信息，请参见 see [client-configuration:virt-xenkvm.pdf](#)。

客户端具有系统类型后，该客户端的“系统细节”页面中便会显示选项卡。选项卡可用于创建和管理虚拟 Guest，以及管理存储池和虚拟网络。

## 9.2. 创建虚拟 Guest

您可以在 Uyuni Web UI 中向虚拟化主机添加虚拟 Guest。

过程：创建虚拟 Guest

1. 在 Uyuni Web UI 中，导航到 **系统 > 系统列表**，单击虚拟化主机的名称，然后导航到选项卡。

2. 在部分，填写以下细节：

- 在 Guest 子选项卡中，单击  [建 Guest]。
- 在字段中，键入 Guest 的名称。
- 在字段中，选择要使用的超级管理程序。
- 在字段中，选择全虚拟化或半虚拟化。
- 在字段中，以 MiB 为单位键入 Guest 磁盘的大小上限。
- 在 CPU 中，键入 Guest 的 vCPU 数量。
- 在字段中，选择要在 Guest 上使用的模拟 CPU 体系结构。默认会选择与虚拟主机匹

配的体系结构。

- 在“**自动安装工具**”字段中，选择要用于安装 Guest 的自动安装工具。如果不使用自动安装，请将此字段留空。
- 3. 在“**URL**”部分，填写要用于客户端的虚拟磁盘的细节。在“**URL**” URL 字段中，务必键入操作系统映像的路径。如果不这么做，创建的 Guest 的磁盘将会是空磁盘。
- 4. 在“**MAC**”部分，填写要用于客户端的虚拟网络接口的细节。将 MAC “**MAC**”字段留空以生成 MAC 地址。
- 5. 在“**显卡驱动程序**”部分，填写要用于客户端的显卡驱动程序的细节。
- 6. 安排创建 Guest 的时间，然后单击  [创] [建] 以创建 Guest。
- 7. 新虚拟 Guest 在成功创建后会立即启动。

也可以在 Uyuni Web UI 中的 Pacemaker 群集上添加虚拟 Guest。

过程：创建由群集管理的虚拟 Guest

1. 按照在群集某个节点上“**添加 Guest**”的过程操作，并遵循以下额外要求：
  - 确保“**共享文件夹**”字段处于选中状态。
  - 在“**VM**”“**共享文件夹**”字段中，键入由所有群集节点共享的文件夹的路径，Guest 配置将存储在该位置。
  - 确保每个磁盘都位于由所有群集节点共享的存储池上。

可实时迁移由群集管理的虚拟 Guest。

## 9.3. 使用 Xen 和 KVM 虚拟化

您可以直接在 Uyuni 中管理 Xen 和 KVM 虚拟化客户端。

开始前，需要在 Uyuni 服务器上设置虚拟主机。然后，可为其他虚拟主机和虚拟 Guest 设置使用 AutoYaST 或 Kickstart 进行的自动安装。

本节还包含有关在安装虚拟 Guest 后对其进行管理的信息。

### 9.3.1. 主机设置

在 VM 主机上设置 Xen 或 KVM 的方式取决于您要在主机关联的 Guest 上使用的操作系统：

- 对于 SUSE 操作系统，请参见 <https://documentation.suse.com/sles/15-SP4/html/SLES-all/book-virtualization.html> 上的《SLES Virtualization Guide》（SLES 虚拟化指南）。
- 对于 Red Hat Enterprise Linux 操作系统，请参见适用于您所用版本的 Red Hat 文档。

借助“**启动**”公式，您可以初始化主机。有关详细信息，请参见 [client-configuration:virt-xenkvm.pdf](#)。

#### 9.3.1.1. 背景信息

Uyuni 使用 libvirt 安装和管理 Guest。您的主机上必须已安装 libvirt-daemon 软件包。大多数情况下，

默认设置足以满足要求，您无需进行调整。不过，如果您要在 Guest 上以非 root 用户身份访问 VNC 控制台，则需要对配置进行一些更改。有关如何设置 VNC 控制台的详细信息，请参见操作系统的文档。

Uyuni 服务器上需要有引导脚本。引导脚本必须包含主机的激活密钥。我们建议包含您的 GPG 密钥以增强安全性。有关创建引导脚本的详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。

准备好引导脚本后，使用该脚本在 Uyuni 服务器中注册主机。有关客户端注册的详细信息，请参见 [Client-configuration > Registration-overview](#)。

### 9.3.1.2. 初始化

使用公式初始化主机：

过程：初始化

1. 在 Uyuni Web UI 中，导航到主机的页面，然后单击选项卡。
2. 选择公式，然后单击 **[保存]**。
3. 单击子选项卡。
4. 选中相应设置，然后单击 **[保存公式]**。
5. 应用 Highstate，以使更改生效。
6. 重启动 salt-minion 服务以激活新配置：

```
systemctl restart salt-minion
```

### 9.3.2. 自动安装 VM Guest

您可以使用 AutoYaST 或 Kickstart 自动安装并注册 Xen 和 KVM Guest。

需准备好您要将 Guest 注册到的 VM 主机以及每个 Guest 的激活密钥。激活密钥必须具有和权利。有关创建激活密钥的详细信息，请参见 [Client-configuration > Activation-keys](#)。

如果您希望在安装后将 Guest 自动注册到 Uyuni，则需要创建引导脚本。有关创建引导脚本的详细信息，请参见 [Client-configuration > Registration-bootstrap](#)。

#### 9.3.2.1. 创建可自动安装的发行套件

您需要在 VM 主机上创建可自动安装的发行套件，才能通过 Uyuni 自动安装客户端。可以在挂载的本地或远程目录提供发行套件，也可以在以循环方式挂载的 ISO 映像中提供。

根据您在 Guest 上使用的是 Red Hat Enterprise Linux 还是 SUSE 操作系统，可自动安装发行套件的配置有所不同。Red Hat Enterprise Linux 安装的软件包从关联的基础通道提取。用于安装 SUSE 系统的软件包从可自动安装的发行套件中提取。因此，对于 SUSE 系统，可自动安装的发行套件必须是完整的安装源。

表格 49. 可自动安装的发行套件的路径

操作系统类型	内核位置	initrd 位置
Red Hat Enterprise Linux	images/pxeboot/vmlinuz	images/pxeboot/initrd.img
SUSE	boot/<arch>/loader/initrd	boot/<arch>/loader/linux

在所有情况下，均需确保基础通道与可自动安装的发行套件匹配。

开始前，请确保 VM 主机可以使用您的安装媒体。该媒体可以位于远程资源、本地目录或以循环方式挂载的 ISO 映像中。此外，还需确保所有文件和目录都是全局可读的。

过程：创建可自动安装的发行套件

1. 在 Uyuni Web UI 中，导航到系统，自动安装，发行套件，然后单击 **[创建发行套件]**。
2. 在 **部分**，使用以下参数：
  - 在 **部分**，键入发行套件的唯一名称。请仅使用字母、数字、连字符 (-)、点 (.) 和下划线 (\_)，并确保名称包含四个以上字符。
  - 在 **字段中**，键入安装源的绝对路径。
  - 在 **字段中**，选择与安装源匹配的通道。此通道用作非 SUSE 安装的软件包源。
  - 在 **字段中**，选择与安装源匹配的操作系统版本。
  - 在 **字段中**，键入在安装期间引导时要传递给内核的任何选项。默认会添加 `install-` 参数和 `self_update=0` 参数。
  - 在 **部分**，键入在首次引导安装的系统时要传递给内核的任何选项。
3. 单击 **[创建可自动安装的发行套件]** 保存设置。

创建可自动安装的发行套件后，您可以导航到系统，自动安装，发行套件，然后选择要编辑的发行套件进行编辑。

### 9.3.2.2. 创建并上载自动安装配置文件

自动安装配置文件包含安装系统所需的所有安装和配置数据，还包含安装完成后需要执行的脚本。

在 UyunWeb U 中，导航到系统自动安装配置文件，单击 **[创建新 Kickstart]**，然后按照提示操作即可创建 Kickstart 配置文件。

您也可以手动创建 AutoYaST 或 Kickstart 自动安装配置文件。SUSE 提供了 AutoYaST 安装文件模板，您可以基于它们创建自己的自定义文件。您可以在 <https://github.com/SUSE/manager-build-profiles> 中找到这些模板。

如果要使用 AutoYaST 安装 SLES，则还需要包含以下代码段：

```
<products config:type="list">
  <listentry>SLES</listentry>
</products>
```

- 有关 AutoYaST 的详细信息，请参见 [client-configuration:autoinst-profiles.pdf](#)。
- 有关 Kickstart 的详细信息，请参见 [client-configuration:autoinst-profiles.pdf](#) 或适用于您的安装的 Red Hat 文档。

过程：上载自动安装配置文件

1. 在 Uyuni Web UI 中，导航到系统，自动安装，配置文件，然后单击 **[上] 载 Kickstart/Autoyast [文] 件**。
2. 在 **名称** 部分，使用以下参数：
  - 在 **名称** 字段中，为配置文件键入一个唯一的名称。请仅使用字母、数字、连字符 (-)、点 (.) 和下划线 (\_)，并确保名称包含六个以上字符。
  - 在 **发行套件** 字段中，选择您之前创建的可自动安装的发行套件。
  - 在 **Guest 类型** 字段中，选择相关的 Guest 类型（例如 KVM **Xen Guest**）。请勿在此处选择 Xen。
  - 可选：如果您要手动创建自动安装配置文件，可以直接在 **内容** 字段中键入相应内容。如果您已创建文件，请将 **内容** 字段留空。
  - 在 **文件** 字段中，单击 **[选] 择 [文] 件**，然后使用系统对话框选择要上载的文件。如果文件成功上载，**文件** 字段中会显示相应文件名。
  - 内容** 字段中会显示上载的文件的内容。如果您需要编辑其内容，可以直接编辑。
3. 单击 **[创] 建** 以保存更改并存储配置文件。

创建自动安装配置文件后，您可以导航到系统，自动安装，配置文件，然后选择要编辑的配置文件进行编辑。进行所需更改，然后单击 **[创] 建** 保存您的设置。



如果您更改了现有 Kickstart 配置文件的 **引导加载程序和分区选项**，则可能也会修改引导加载程序和分区选项，并可能重写任何自定义设置。请在更改前仔细查看 **引导加载程序和分区选项** 选项卡以校验这些设置。

### 9.3.2.3. 自动注册 Guest

自动安装 VM Guest 后，它们并不会注册到 Uyuni 中。如果您希望 Guest 在安装后立即自动注册，您可以在自动安装配置文件中添加一段用于调用引导脚本并注册 Guest 的内容。

此部分提供向现有 AutoYaST 配置文件添加引导脚本的指令。

有关创建引导脚本的详细信息，请参见 **Client-configuration > Registration-bootstrap**。有关如何针对 Kickstart 执行此操作的说明，请参见适用于您的安装的 Red Hat 文档。

过程：在 AutoYaST 配置文件中添加引导脚本

1. 确保引导脚本包含要注册的 **VM Guest** 的激活密钥，并且脚本位于主机上的 `/srv/www/htdocs/pub/bootstrap_vm_guests.sh` 中。
2. 在 Uyuni Web UI 中，导航到系统，自动安装，配置文件，然后选择要与此脚本关联的 AutoYaST 配置文件。
3. 在 **内容** 字段中，于文件末尾的 `</profile>` 结束标记前面添加以下代码段。务必在代码段中的示

例 IP 地址 192.168.1.1 替换为 Uyuni 服务器的正确 IP 地址：

```
<scripts>
<init-scripts config:type="list">
<script>
<interpreter>shell</interpreter>
<location>
    http://192.168.1.1/pub/bootstrap/bootstrap_vm_guests.sh
</location>
</script>
</init-scripts>
</scripts>
```

4. 单击 **更新** 保存您的更改。



如果 AutoYaST 配置文件已包含 `<scripts>` 部分，请勿再添加，而是将引导代码段放在现有 `<scripts>` 部分内。

#### 9.3.2.4. 自动安装 VM Guest

一切都设置好后，就可以开始自动安装 VM Guest 了。



每个 VM 主机一次只能安装一个 Guest。如果您要安排多个自动安装，请务必安排合理的时间，确保下一个安装不会在现有安装完成前开始。如果某个 Guest 安装在另一个安装仍在进行时开始，则正在进行的安装可能会被取消。

1. 在 Uyuni Web UI 中，导航到 **系统** > **概览**，然后选择要在其中安装 Guest 的 VM 主机。
2. 依次导航到 **自动安装** 选项卡和 **安排** 子选项卡。
3. 选择要使用的自动安装配置文件，并为 Guest 指定唯一的名称。
4. 选择代理（如果适用）并输入日程安排。
5. 要更改 Guest 的硬件配置文件和配置选项，请单击 **[高级选项]**。
6. 单击 **[安排自动安装并完成]** 以完成设置。

#### 9.3.3. 管理 VM Guest

您可以使用 Uyuni Web UI 来管理 VM Guest，包括执行关机、重启动以及调整 CPU 和内存分配的操作。

为此，需要将 Xen 或 KVM VM 主机注册到 Uyuni 服务器，并在主机上运行 libvirtd 服务。

在 Uyuni Web UI 中，导航到 **系统** > **系统列表**，然后单击要管理的 Guest 的 VM 主机。导航到 **自动安装** 选项卡以查看所有注册到此主机中的 Guest，并访问管理功能。

---

有关使用 Web UI 管理 VM Guest 的详细信息, 请参见 **Reference > Systems**。

# Chapter 10. 虚拟主机管理器

虚拟主机管理器 (VHM) 用于收集各种客户端类型的信息。

VHM 可用于收集私有云或公有云实例，并将其分为不同的虚拟化组。以这种方式组织虚拟化客户端后，Taskomatic 便可收集客户端的相关数据以显示在 Uyuni Web UI 中。借助 VHM，您还可以在虚拟化客户端上使用订阅匹配。

您可以在 Uyuni 服务器上创建 VHM，并使用它来清点可用的公有云实例。您还可以使用 VHM 管理通过 Kubernetes 创建的群集。

- 有关将 VHM 与 Amazon Web Services 搭配使用的详细信息，请参见 [Client-configuration > Vhm-aws](#)。
- 有关将 VHM 与 Microsoft Azure 搭配使用的详细信息，请参见 [Client-configuration > Vhm-azure](#)。
- 有关将 VHM 与 Google Compute Engine 搭配使用的详细信息，请参见 [Client-configuration > Vhm-gce](#)。
- 有关将 VHM 与 Nutanix 搭配使用的详细信息，请参见 [Client-configuration > Vhm-nutanix](#)。
- 有关将 VHM 与 VMWare vSphere 搭配使用的详细信息，请参见 [Client-configuration > Vhm-vmware](#)。
- 有关将 VHM 与其他主机搭配使用的详细信息，请参见 [Client-configuration > Vhm-file](#)。

## 10.1. 虚拟主机管理器和 Amazon Web Services

您可以使用虚拟主机管理器 (VHM) 收集 Amazon Web Services (AWS) 中的实例。

VHM 允许 Uyuni 获取并报告有关您的群集的信息。有关 VHM 的详细信息，请参见 [Client-configuration > Vhm](#)。

### 10.1.1. 创建 Amazon EC2 VHM

虚拟主机管理器 (VHM) 在 Uyuni 服务器上运行。

确保您已在 Uyuni 服务器上安装 `virtual-host-gatherer-libcloud` 软件包。

过程：创建 Amazon EC2 VHM

1. 在 Uyuni Web UI 中，导航到 **系统 > 虚拟主机管理器**。
2. 单击 **[创]** 并从下拉菜单中选择 **Amazon EC2**。
3. 在 **Amazon EC2** 部分，使用以下参数：
  - 在 **名称** 字段中，为 VHM 键入自定义名称。
  - 在 **访问密钥 ID** 字段中，键入 Amazon 提供的访问密钥 ID。
  - 在 **机密访问密钥** 字段中，键入与 Amazon 实例关联的机密访问密钥。

- 在“地区”字段中，键入要使用的地区。
- 在“区域”字段中，键入您的 VM 所在的区域。要使订阅匹配功能正常工作，就必须提供此信息。有关设置地区和区域的详细信息，请参见 [client-configuration:virtualization.pdf](#)。

4. 单击  [创建] 保存更改并创建 VHM。

5. 在“VM”页面中，选择新 VHM。

6. 在“VM”页面中，单击  [刷新] [新数据] 以清点新 VHM。

要查看已清点的对象和资源，请导航到 **系统** > **系统列表** > **虚拟系统**。

在 Amazon 公有云上运行的实例会向 Uyuni 服务器报告 UUID，其格式为一个 `i` 后跟 17 个十六进制数字：

```
I1234567890abcdef0
```

## 10.1.2. 虚拟主机管理器的 AWS 权限

出于安全原因，请始终为要执行的任务授予尽可能最小的权限。不建议对连接到 AWS 的用户使用具有过高权限的访问密钥。

为了让 SUSE Manager 从 AWS 收集所需的信息，VHM 需要相权限来说明 EC2 实例和地址。一种收集此信息的方法是专门为此任务创建一个新的 IAM 用户 (Identity and Access Management)，创建一个如下策略并关联到该用户：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAddresses",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

您可以通过限制对特定地区的访问来对权限施加更多限制。有关详细信息，请参见  
[https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ExamplePolicies\\_EC2.html#iam-example-read-only](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ExamplePolicies_EC2.html#iam-example-read-only)。

## 10.2. 虚拟主机管理器和 Azure

您可以使用虚拟主机管理器 (VHM) 收集 Microsoft Azure 中的实例。

VHM 允许 Uyuni 获取并报告有关您的虚拟机的信息。有关 VHM 的详细信息，请参见 [Client-configuration > Vhm](#)。

### 10.2.1. 先决条件

要让您创建的 VHM 访问 Azure VM，需要为其指派正确的权限。

请以订阅管理员身份登录您的 Azure 帐户，并确保该 Azure 用户帐户和应用程序在正确的组中。应用程序所在的组决定了应用程序所拥有的角色，因而决定了其拥有的权限。

### 10.2.2. 创建 Azure VHM

虚拟主机管理器 (VHM) 在 Uyuni 服务器上运行。

确保您已在 Uyuni 服务器上安装 virtual-host-gatherer-libcloud 软件包。

过程：创建 Azure VHM

1. 在 Uyuni Web UI 中，导航到系统 > 虚拟主机管理器。
2. 单击  并从下拉菜单中选择 Azure。
3. 在 Azure 部分，使用以下参数：
  - 在名称字段中，为 VHM 键入自定义名称。
  - 在订阅 ID 字段中，键入在 Azure 订阅 > 订阅 > 订阅页面中找到的订阅 ID。
  - 在应用程序 ID 字段中，键入您在注册应用程序时获得的应用程序 ID。
  - 在租户 ID 字段中，键入您在注册应用程序时获得的、由 Azure 提供的租户 ID。
  - 在机密密钥字段中，键入与 Azure 实例关联的机密密钥。
  - 在区域字段中，键入您的 VM 所在的区域。例如，如果位于西欧，则输入欧洲。要使订阅匹配功能正常工作，就必须提供此信息。
4. 单击  保存更改并创建 VHM。
5. 在 VHM 页面中，选择新 VHM。
6. 在 VHM 页面中，单击  以清点新 VHM。

要查看已清点的对象和资源，请导航到系统 > 系统列表 > 虚拟系统。

### 10.2.3. 指派权限

如果未正确设置权限，您在运行 virtual-host-gatherer 时可能会收到如下所示的错误：

一般错误：[AuthorizationFailed] 对象 ID 为 'object\_ID' 的客户端 'client\_name' 无权在 '/subscriptions/not-very-secret-subscription-id' 范围执行操作 'Microsoft.Compute/virtualMachines/read'，或该范围无效。如果访问权限是最近授予的，请刷新您的身份凭证。

要确定正确的身份凭证，请在 Uyuni 服务器上的提示符处运行以下命令：

```
virtual-host-gatherer -i input_azure.json -o out_azure.json -vvv
```

input\_azure.json 文件应包含以下信息：

```
[  
  {  
    "id": "azure_vhm",  
    "module": "Azure",  
    "subscription_id": "subscription-id",  
    "application_id": "application-id",  
    "tenant_id": "tenant-id",  
    "secret_key": "secret-key",  
    "zone": "zone"  
  }  
]
```

## 10.2.4. Azure UUID

在 Azure 公有云上运行的实例会向 Uyuni 服务器报告如下 UUID：

```
13f56399-bd52-4150-9748-7190aae1ff21
```

## 10.3. 虚拟主机管理器和 Google Compute Engine

您可以使用虚拟主机管理器 (VHM) 收集 Google Compute Engine (GCE) 中的实例。

VHM 允许 Uyuni 获取并报告有关您的虚拟机的信息。有关 VHM 的详细信息，请参见 [Client-configuration > Vhm](#)。

### 10.3.1. 先决条件

要让您创建的 VHM 可以访问 GCE VM，需要为其指派正确的权限。

以管理员身份登录您的 Google Cloud Platform 帐户，并使用 Cloud Identity and Access Management (IAM)

工具确保服务帐户拥有适当的角色。

### 10.3.2. 创建 GCE VHM

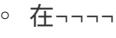
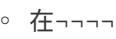
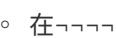
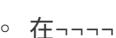
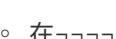
虚拟主机管理器 (VHM) 在 Uyuni 服务器上运行。

要运行 VHM，需要打开端口 443 以使您的 Uyuni 服务器可访问客户端。

确保您已在 Uyuni 服务器上安装 virtual-host-gatherer-libcloud 软件包。

开始前，请登录 GCE 面板并下载证书文件。将此文件存储在 Uyuni 服务器本地，并记下路径。

过程：创建 GCE VHM

1. 在 Uyuni Web UI 中，导航到系统，虚拟主机管理器。
2. 单击  并从下拉菜单中选择 Google Compute Engine。
3. 在  部分，使用以下参数：
  - 在  字段中，为 VHM 键入自定义名称。
  - 在  字段中，键入与您的服务帐户关联的电子邮件地址。
  - 在  字段中，键入 Uyuni 服务器上用于存储您从 GCE 面板下载的密钥的本地路径。
  - 在  字段中，键入 GCE 实例使用的项目 ID。
  - 在  字段中，键入您的 VM 所在的区域。要使订阅匹配功能正常工作，就必须提供此信息。
4. 单击  保存更改并创建 VHM。
5. 在  页面中，选择新 VHM。
6. 在  页面中，单击    以清点新 VHM。

要查看已清点的对象和资源，请导航到系统，系统列表，虚拟系统。

### 10.3.3. 指派权限

如果未正确设置权限，您在运行 virtual-host-gatherer 时可能会收到如下所示的错误：

```
错误: {'domain': 'global', 'reason': 'forbidden', 'message': "需要
'compute.zones.list' 权限, 'projects/project-id'"}
错误: 无法使用指定的身份凭证连接 Google Compute Engine 公有云。
```

要确定正确的身份凭证，请在 Uyuni 服务器上的提示符处运行以下命令：

```
virtual-host-gatherer -i input_google.json -o out_google.json -vvv
```

input\_google.json 文件应包含以下信息：

```
[  
  {  
    "id": "google_vhm",  
    "module": "GoogleCE",  
    "service_account_email": "mail@example.com",  
    "cert_path": "secret-key",  
    "project_id": "project-id",  
    "zone": "zone"  
  }  
]
```

### 10.3.4. GCE UUID

在 Google 公有云上运行的实例会向 Uyuni 服务器报告如下 UUID：

152986662232938449

## 10.4. 使用 Nutanix 虚拟化

您可以通过在 Uyuni 中设置虚拟主机管理器 (VHM) 来使用 Nutanix AHV 虚拟机。开始前，您需要在 Uyuni 服务器上设置 VHM，然后清点可用的 VM 主机。

### 10.4.1. VHM 设置

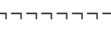
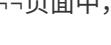
虚拟主机管理器 (VHM) 在 Uyuni 服务器上运行。

确保您已在 Uyuni 服务器上安装 `virtual-host-gatherer-Nutanix` 软件包。

要运行 VHM，必须打开端口 9440 以使您的 Uyuni 服务器可访问 Nutanix Prism Element API。

过程：创建 Nutanix VHM

1. 在 Uyuni Web UI 中，导航到系统 > 虚拟主机管理器。
2. 单击 **[ 创建 ]**，然后选择 Nutanix AHV。
3. 在 **Nutanix AHV** 部分，使用以下参数：
  - 在 **名称** 字段中，为 VHM 键入自定义名称。
  - 在 **FQDN** 字段中，键入完全限定的域名 (FQDN) 或主机的 IP 地址。
  - 在 **端口** 字段中，键入要使用的 Prism Element API 端口（例如，9440）。
  - 在 **用户名** 字段中，键入与 VM 主机关联的用户名。
  - 在 **口令** 字段中，键入与 VM 主机用户关联的口令。

4. 单击  保存更改并创建 VHM。
5. 在  页面中选择新 VHM。
6. 在  页面中，单击  以清点新 VHM。

要查看已清点的对象和资源，请导航到 **系统** > **系统列表** > **虚拟系统**。



有时，在浏览器中使用 HTTPS 连接 Nutanix Prism API 服务器可能会发生  错误。如果发生此情况，刷新来自虚拟主机管理器的数据将会失败。Nutanix API 服务器上必须存在有效的 SSL 证书（不是自我签名证书）。如果您为 Nutanix SSL 证书使用了自定义 CA 机构，请将自定义 CA 证书复制到 Uyuni 服务器上的 /etc/pki/trust/anchors 中。

在命令行上运行 `update-ca-certificates` 命令以重新信任证书，然后重启 spacewalk 服务。

创建并配置好 VHM 后，Taskomatic 即会自动运行数据收集过程。如果您要手动进行数据收集，请导航到 **系统** > **虚拟主机管理器**，选择适当的 VHM，然后单击 。

Uyuni 随附了一个名为 `virtual-host-gatherer` 的工具，可以使用相应 API 连接到 VHM 并请求虚拟主机的相关信息。`virtual-host-gatherer` 计算机会维护可选模块的概念，每个模块可启用一个特定的 VHM。Taskomatic 会在夜间自动调用此工具。`virtual-host-gatherer` 工具的日志文件位于 `/var/log/rhn/gatherer.log`。

## 10.5. Virtualization with VMware

You can use VMware vSphere virtual machines, including ESXi and vCenter, with Uyuni by setting up a virtual host manager (VHM).

开始前，您需要在 Uyuni 服务器上设置 VHM，然后清点可用的 VM 主机。之后，Taskomatic 便可以开始使用 VM API 收集数据。

### 10.5.1. VHM 设置

虚拟主机管理器 (VHM) 在 Uyuni 服务器上运行。

To run a VHM, your Uyuni Server needs to have port 443 open, to access the VMware API.

VMware hosts use access roles and permissions to control access to hosts and guests. Ensure that any VMware objects or resources that you want to be inventoried by the VHM have at least `read-only` permissions. If you want to exclude any objects or resources, mark them with `no-access`.

当您向 Uyuni 添加新主机时，需考虑是否需要让 Uyuni 清点已指派给用户和对象的角色和权限。

For more on users, roles, and permissions, see the VMware vSphere documentation:  
<https://docs.vmware.com/en/VMware-vSphere/index.html>

Procedure: Creating a VMware VHM

1. 在 Uyuni Web UI 中，导航到系统 > 虚拟主机管理器。
2. Click [Create] and select VMware-based.
3. In the Add a VMware-based Virtual Host Manager section, use these parameters:
  - 在“名称”字段中，为 VHM 键入自定义名称。
  - 在“完全限定域名 (FQDN) 或 IP 地址”字段中，键入完全限定的域名 (FQDN) 或主机的 IP 地址。
  - 在“端口”字段中，键入要使用的 ESXi API 端口（例如，443）。
  - 在“用户名”字段中，键入与 VM 主机关联的用户名。
  - 在“口令”字段中，键入与 VM 主机用户关联的口令。
4. 单击 [Create] 保存更改并创建 VHM。
5. 在“虚拟主机管理器”页面中选择新 VHM。
6. 在“对象”页面中，单击 [刷新] 以清点新 VHM。

要查看已清点的对象和资源，请导航到系统 > 系统列表 > 虚拟系统。



有时，在浏览器中使用 HTTPS 连接 ESXi 服务器可能会发生“SSL 错误”。如果发生此情况，刷新来自虚拟主机服务器的数据将会失败。要纠正该问题，请解压缩来自 ESXi 服务器的证书，然后将其复制到 /etc/pki/trust/anchors。在命令行上运行 update-ca-certificates 命令以重新信任证书，然后重启 spacewalk 服务。

创建并配置好 VHM 后，Taskomatic 即会自动运行数据收集过程。如果您要手动进行数据收集，请导航到系统 > 虚拟主机管理器，选择适当的 VHM，然后单击 [刷新]。

Uyuni 随附了一个名为 virtual-host-gatherer 的工具，可以使用相应 API 连接到 VHM 并请求虚拟主机的相关信息。virtual-host-gatherer 计算机会维护可选模块的概念，每个模块可启用一个特定的 VHM。Taskomatic 会在夜间自动调用此工具。virtual-host-gatherer 工具的日志文件位于 /var/log/rhn/gatherer.log。

## 10.5.2. Troubleshooting SSL Errors on VMware

If you see SSL errors while configuring your VMware installation, you need to download the CA certificate file from VMware, and trust it on Uyuni.

Procedure: Trusting VMware CA Certificates

1. Download the CA Certificate from your VMware installation. 您可以通过登录 vCenter Web UI 并单击 [下载] 来实现此目的。
2. 如果下载的 CA 证书文件为 .zip 格式，请解压缩该存档文件。证书文件的扩展名有多种。例如，certificate.0。
3. 将证书文件复制到 Uyuni 服务器上并保存到 /etc/pki/trust/anchors/ 目录中。
4. 将复制的证书的文件名后缀改为 .crt 或 .pem。
5. 在 Uyuni 服务器上的命令提示符处更新 CA 证书记录：

```
update-ca-certificates
```

## 10.6. 使用其他第三方提供者虚拟化

如果您要使用 Xen、KVM 或 VMware 以外的第三方虚拟化提供者，可以将 JSON 配置文件导入到 Uyuni 中。

Similarly, if you have a VMware installation that does not provide direct access to the API, a file-based VHM provides you with some basic management features.



此选项用于导入使用 `virtual-host-gatherer` 工具创建的文件，不可用于导入手动创建的文件。

过程：导出和导入 JSON 文件

1. 在 VM 网络上运行 `virtual-host-gatherer` 以导出 JSON 配置文件。
2. 将产生的文件保存到 Uyuni 服务器可访问的位置。
3. 在 Uyuni Web UI 中，导航到系统，虚拟主机管理器。
4. 单击 **[创建]**，然后选择 **VHM**。
5. 在 **VHM** 部分，使用以下参数：
  - 在 **名称** 字段中，为 VHM 键入自定义名称。
  - 在 **URL** 字段中，键入导出的 JSON 配置文件的路径。
6. 单击 **[创建]** 保存更改并创建 VHM。
7. 在 **VHM** 页面中，选择新 VHM。
8. 在 **VHM** 页面中，单击 **[刷新]** 以清点新 VHM。

列表 3. 例如：导出的 JSON 配置文件：

```
{
  "examplevhost": {
    "10.11.12.13": {
      "cpuArch": "x86_64",
      "cpuDescription": "AMD Opteron(tm) 处理器 4386",
      "cpuMhz": 3092.212727,
      "cpuVendor": "amd",
      "hostIdentifier": "'vim.HostSystem:host-182'",
      "name": "11.11.12.13",
      "os": "VMware ESXi",
      "osVersion": "5.5.0",
      "ramMb": 65512,
      "totalCpuCores": 16,
      "totalCpuSockets": 2,
    }
  }
}
```

```

    "totalCpuThreads": 16,
    "type": "vmware",
    "vms": {
        "vCenter": "564d6d90-459c-2256-8f39-3cb2bd24b7b0"
    }
},
"10.11.12.14": {
    "cpuArch": "x86_64",
    "cpuDescription": "AMD Opteron(tm) 处理器 4386",
    "cpuMhz": 3092.212639,
    "cpuVendor": "amd",
    "hostIdentifier": "'vim.HostSystem:host-183'",
    "name": "10.11.12.14",
    "os": "VMware ESXi",
    "osVersion": "5.5.0",
    "ramMb": 65512,
    "totalCpuCores": 16,
    "totalCpuSockets": 2,
    "totalCpuThreads": 16,
    "type": "vmware",
    "vms": {
        "49737e0a-c9e6-4ceb-aef8-6a9452f67cb5": "4230c60f-3f98-2a65-f7c3-600b26b79c22",
        "5a2e4e63-a957-426b-bfa8-4169302e4fdb": "42307b15-1618-0595-01f2-427ffcd88e",
        "NSX-gateway": "4230d43e-aafe-38ba-5a9e-3cb67c03a16a",
        "NSX-l3gateway": "4230b00f-0b21-0e9d-dfde-6c7b06909d5f",
        "NSX-service": "4230e924-b714-198b-348b-25de01482fd9"
    }
}
}
}

```

有关详细信息，请参见 Uyuni 服务器上 `virtual-host-gatherer` 的手册页：

```
man virtual-host-gatherer
```

该软件包的 README 文件提供有关超级管理程序----的背景信息及其他信息：

```
/usr/share/doc/packages/virtual-host-gatherer/README.md
```

手册页和 README 文件还包含示例配置文件。

# Chapter 11. GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

---

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- 
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
  - H. Include an unaltered copy of this License.
    - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
    - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
  - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
  - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
  - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
  - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
  - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the

Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".