



U Y U N I

Uyuni 2023.02

客户端配置指南

2023年02月23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 SUSE 和 openSUSE 客户端功能	4
1.4. Supported SLE Micro Client Features	6
1.5. openSUSE Leap Micro Client Features	8
1.6. 支持的 Alibaba Cloud Linux 功能	9
1.7. 支持的 AlmaLinux 功能	11
1.8. 支持的 Amazon Linux 功能	13
1.9. 支持的 CentOS 功能	15
1.10. 支持的 Debian 功能	17
1.11. 支持的 Oracle 功能	19
1.12. 支持的 Red Hat Enterprise Linux 功能	21
1.13. 支持的 Rocky Linux 功能	23
1.14. 支持的 Ubuntu 功能	25
2. 配置基本知识	28
2.1. 软件通道	28
2.1.1. 通过 SUSE Package Hub 提供的软件包	28
2.1.2. 通过 AppStream 提供的软件包	29
2.1.3. 通过 EPEL 提供的软件包	29
2.1.4. SUSE Linux Enterprise 客户端的 Unified Installer 更新通道	29
2.1.5. 软件软件源	29
2.1.6. 软件产品	31
2.2. 引导软件源	31
2.2.1. 准备创建引导软件源	31
2.2.2. 自动模式的选项	31
2.2.3. 手动生成引导软件源	32
2.2.4. 引导和自定义通道	33
2.3. 激活密钥	34
2.3.1. 组合多个激活密钥	36
2.3.2. 重新激活密钥	36
2.3.3. 激活密钥最佳实践	37
2.4. GPG 密钥	38
2.4.1. 在客户端上信任 GPG 密钥	38

客户端配置指南概述

更新日期：2023-02-23

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

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，您可以安装传统客户端或 Salt 客户端，运行 SUSE Linux Enterprise 或其他 Linux 操作系统。

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

下表列出了传统客户端和 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

Operating System	Architecture	Traditional Clients	Salt Clients
SUSE Linux Enterprise 15	x86-64, ppc64le, IBM Z, ARM	✓	✓
SUSE Linux Enterprise 12	x86-64, ppc64le, IBM Z, ARM	✓	✓
SUSE Linux Enterprise Server for SAP 15	x86-64, ppc64le	✓	✓
SUSE Linux Enterprise Server for SAP 12	x86-64, ppc64le	✓	✓
SLE Micro	x86-64, ppc64le, aarch64	✗	✓
openSUSE Leap 15	x86-64, aarch64	✓	✓
Alibaba Cloud Linux 2	x86-64, aarch64	✗	✓
AlmaLinux 9	x86-64, ppc64le, IBM Z, aarch64	✗	✓
AlmaLinux 8	x86-64, aarch64	✗	✓
Amazon Linux 2	x86-64, aarch64	✗	✓
CentOS 7	x86-64, ppc64le, aarch64	✓	✓

Operating System	Architecture	Traditional Clients	Salt Clients
Debian 11	x86-64	✗	✓
Debian 10	x86-64	✗	✓
Oracle Linux 9	x86-64, aarch64	✗	✓
Oracle Linux 8	x86-64, aarch64	✗	✓
Oracle Linux 7	x86-64, aarch64	✓	✓
Red Hat Enterprise Linux 9	x86-64	✗	✓
Red Hat Enterprise Linux 8	x86-64	✗	✓
Red Hat Enterprise Linux 7	x86-64	✓	✓
Rocky Linux 9	x86-64, aarch64, ppc64le, s390x	✗	✓
Rocky Linux 8	x86-64, aarch64	✗	✓
Ubuntu 22.04	amd64	✗	✓
Ubuntu 20.04	amd64	✗	✓
Ubuntu 18.04	amd64	✗	✓



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

1.3. 支持的 SUSE 和 openSUSE 客户端功能

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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

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	✓	✓	Salt
安装软件包	✓	✓	Salt
应用补丁	✓	✓	Salt
远程命令	✓	✓	Salt
系统软件包状态	Salt	Salt	Salt
系统自定义状态	Salt	Salt	Salt
组自定义状态	Salt	Salt	Salt
组织自定义状态	Salt	Salt	Salt
系统集管理器 (SSM)	✓	✓	Salt
产品迁移	✓	✓	Salt
基本虚拟 Guest 管理 *	✓	✓	Salt
高级虚拟 Guest 管理 *	Salt	Salt	Salt
虚拟 Guest 安装 (AutoYaST)，作为主机操作系统	Traditional	Traditional	✗
虚拟 Guest 安装（映像模板），作为主机操作系统	Salt	Salt	Salt

功能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
虚拟 Guest 管理	Salt	Salt	Salt
系统部署 (PXE/AutoYaST)	✓	✓	✓
系统重新部署 (AutoYaST)	✓	✓	Salt
联系方法	Traditional ：OSAD、RHNSD、SSH-push。Salt ：ZeroMQ、Salt-SSH	Traditional ：OSAD、RHNSD、SSH-push。Salt ：ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	✓	✓	Salt
操作链	✓	✓	Salt
暂存（预先下载软件包）	✓	✓	Salt
重复软件包报告	✓	✓	Salt
CVE 审计	✓	✓	Salt
SCAP 审计	✓	✓	Salt
软件包校验	Traditional	Traditional	✗
软件包锁定	Salt	Salt	Salt
系统锁定	Traditional	Traditional	✗
维护时段	✓	✓	✓
系统快照	Traditional	Traditional	✗
配置文件管理	✓	✓	Salt
软件包配置文件	Traditional。Salt：支持配置文件，不支持同步	Traditional。Salt：支持配置文件，不支持同步	Salt：支持配置文件，不支持同步
电源管理	✓	✓	✓
监视服务器	Salt	Salt	Salt
受监视客户端	Salt	Salt	Salt
Docker buildhost	Salt	Salt	?
构建含操作系统的 Docker 映像	Salt	Salt	Salt
Kiwi buildhost	Salt	?	?
构建含操作系统的 Kiwi 映像	Salt	?	✗
重复性操作	Salt	Salt	Salt
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. Supported SLE Micro Client Features



- The operating system you run on a client is supported by the organization that supplies the operating system. SLE Micro is supported by SUSE.

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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

表格 4. Supported Features on SLE Micro Operating Systems

Feature	SLE Micro
Client	Salt
Operating system packages	Salt
Registration	Salt
Install packages	Salt
Apply patches (requires CVE ID)	Salt
Remote commands	Salt
System package states	Salt
System custom states	Salt
Group custom states	Salt
Organization custom states	Salt
System set manager (SSM)	Salt
Product migration	Salt
Basic Virtual Guest Management *	Salt
Advanced Virtual Guest Management *	Salt
Virtual Guest Installation (Kickstart), as Host OS	Salt

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

1.5. openSUSE Leap Micro Client Features



openSUSE Leap Micro is supported by the SUSE community.

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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

表格 5. Supported Features on openSUSE Leap Micro Operating Systems

Feature	openSUSE Leap Micro
Client	Salt
Operating system packages	Salt
Registration	Salt
Install packages	Salt
Apply patches (requires CVE ID)	Salt
Remote commands	Salt
System package states	Salt
System custom states	Salt
Group custom states	Salt
Organization custom states	Salt
System set manager (SSM)	Salt
Product migration	Salt
Basic Virtual Guest Management *	Salt
Advanced Virtual Guest Management *	Salt
Virtual Guest Installation (Kickstart), as Host OS	Salt
Virtual Guest Installation (image template), as Host OS	Salt
System deployment (PXE/Kickstart)	Salt
System redeployment (Kickstart)	Salt
Contact methods	Salt: ZeroMQ
Works with Uyuni Proxy	Salt

Feature	openSUSE Leap Micro
Action chains	Salt
Staging (pre-download of packages)	?
Duplicate package reporting	Salt
CVE auditing (requires CVE ID)	Salt
SCAP auditing	?
Package verification	?
Package locking	Salt
System locking	?
Maintenance Windows	?
System snapshot	✗
Configuration file management	Salt
Snapshots and profiles	Salt: Profiles supported, Sync not supported
Power management	Salt
Monitoring server	✗
Monitored clients	Salt
Docker buildhost	✗
Build Docker image with OS	✗
Kiwi buildhost	✗
Build Kiwi image with OS	Salt
Recurring Actions	Salt
AppStreams	N/A
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。

1.6. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持

表格 6. Alibaba Cloud Linux 操作系统上支持的功能

功能	Alibaba Cloud Linux 2
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	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	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	Salt
操作链	Salt
暂存（预先下载软件包）	Salt
重复软件包报告	Salt

功能	Alibaba Cloud Linux 2
CVE 审计（需要 CVE ID）	Salt
SCAP 审计	Salt
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	Salt
快照和配置文件	Salt: 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	Salt
Docker buildhost	Salt
构建含操作系统的 Docker 映像	Salt
Kiwi buildhost	Salt
构建含操作系统的 Kiwi 映像	Salt
重复性操作	Salt
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。

*传统堆栈在 Alibaba Cloud Linux 上可用，但其不受支持。

1.7. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

表格 7. AlmaLinux 操作系统上支持的功能

功能	AlmaLinux 9	AlmaLinux 8
客户端	Salt (单纯的 AlmaLinux)	Salt (单纯的 AlmaLinux)
系统软件包	AlmaLinux 社区	AlmaLinux 社区
注册	Salt	Salt
安装软件包	Salt	Salt
应用补丁	Salt	Salt
远程命令	Salt	Salt
系统软件包状态	Salt	Salt
系统自定义状态	Salt	Salt
组自定义状态	Salt	Salt
组织自定义状态	Salt	Salt
系统集管理器 (SSM)	Salt	Salt
产品迁移	不适用	不适用
基本虚拟 Guest 管理 *	Salt	Salt
高级虚拟 Guest 管理 *	Salt	Salt
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	Salt	Salt
系统部署 (PXE/Kickstart)	Salt	Salt
系统重新部署 (Kickstart)	Salt	Salt
联系方法	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	Salt	Salt
操作链	Salt	Salt
暂存（预先下载软件包）	Salt	Salt
重复软件包报告	Salt	Salt
CVE 审计	Salt	Salt

功能	AlmaLinux 9	AlmaLinux 8
SCAP 审计	Salt	Salt
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	Salt	Salt
快照和配置文件	Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步
电源管理	Salt	Salt
监视服务器	✗	✗
受监视客户端	Salt	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	✗	✗
重复性操作	Salt	Salt
AppStream	✓	✓
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。

1.8. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持

表格 8. Amazon Linux 操作系统上支持的功能

功能	Amazon Linux 2
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	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	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	Salt
操作链	Salt
暂存（预先下载软件包）	Salt
重复软件包报告	Salt
CVE 审计（需要 CVE ID）	Salt
SCAP 审计	Salt
软件包校验	✗
软件包锁定	✗

功能	Amazon Linux 2
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	?
监视服务器	✗
受监视客户端	Salt
Docker buildhost	Salt
构建含操作系统的 Docker 映像	Salt
Kiwi buildhost	Salt
构建含操作系统的 Kiwi 映像	Salt
重复性操作	Salt
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。

*传统堆栈在 Amazon Linux 上可用，但其不受支持。

1.9. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

表格 9. CentOS 操作系统上支持的功能

功能	CentOS 7
客户端	✓ (单纯的 CentOS)
系统软件包	CentOS 社区
注册	✓
安装软件包	✓
应用补丁（需要 CVE ID）	✓ (需要使用第三方服务进行勘误)
远程命令	✓
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)	✓
产品迁移	不适用
基本虚拟 Guest 管理 *	✓
高级虚拟 Guest 管理 *	Salt
虚拟 Guest 安装 (Kickstart)，作为主机操作系统	Traditional
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	Traditional: OSAD、RHNSD、SSH-push。Salt : ZeroMQ、Salt-SSH
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	Traditional
软件包锁定	✓
系统锁定	Traditional

功能	CentOS 7
维护时段	✓
系统快照	Traditional
配置文件管理	✓
快照和配置文件	Traditional。Salt: 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	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	✗
重复性操作	Salt
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。

1.10. 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
- Traditional 该功能仅在传统客户端上受支持

- Salt 该功能仅在 Salt 客户端上受支持。

表格 10. Debian 操作系统上支持的功能

功能	Debian 10	Debian 11
客户端	✓	✓
系统软件包	Debian 社区	Debian 社区
注册	Salt	Salt
安装软件包	Salt	Salt
应用补丁	?	?
远程命令	Salt	Salt
系统软件包状态	Salt	Salt
系统自定义状态	Salt	Salt
组自定义状态	Salt	Salt
组织自定义状态	Salt	Salt
系统集管理器 (SSM)	Salt	Salt
产品迁移	不适用	不适用
基本虚拟 Guest 管理 *	Salt	Salt
高级虚拟 Guest 管理 *	Salt	Salt
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	Salt	Salt
系统部署 (PXE/Kickstart)	✗	✗
系统重新部署 (Kickstart)	✗	✗
联系方法	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	Salt	Salt
操作链	Salt	Salt
暂存（预先下载软件包）	Salt	Salt
重复软件包报告	Salt	Salt
CVE 审计	?	?
SCAP 审计	?	?
软件包校验	✗	✗
软件包锁定	✓	✓
系统锁定	✗	✗
维护时段	✓	✓

功能	Debian 10	Debian 11
系统快照	✗	✗
配置文件管理	Salt	Salt
软件包配置文件	Salt: 支持配置文件, 不支持同步	Salt: 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	Salt	Salt
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	Salt	Salt
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。

1.11. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持

表格 11. Oracle Linux 操作系统上支持的功能

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

功能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
维护时段	✓	✓	✓
系统快照	Traditional	✗	✗
配置文件管理	✓	Salt	Salt
快照和配置文件	Traditional。Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步
电源管理	✓	Salt	Salt
监视服务器	✗	✗	✗
受监视客户端	Salt	Salt	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	✗	✗	✗
重复性操作	Salt	Salt	Salt
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。

1.12. 支持的 Red Hat Enterprise Linux 功能

This table lists the availability of various features on native Red Hat Enterprise Linux clients.



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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

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

功能	RHEL 7	RHEL 8	RHEL 9
软件包校验	Traditional	✗	✗
软件包锁定	✓	?	?
系统锁定	Traditional	✗	✗
维护时段	✓	✓	✓
系统快照	Traditional	✗	✗
配置文件管理	✓	Salt	Salt
快照和配置文件	Traditional。Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步
电源管理	✓	Salt	Salt
监视服务器	✗	✗	✗
受监视客户端	Salt	Salt	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	✗	✗	✗
重复性操作	Salt	Salt	Salt
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。

1.13. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

表格 13. Rocky Linux 操作系统上支持的功能

功能	Rocky Linux 8	Rocky Linux 9
客户端	Salt (单纯的 Rocky Linux)	Salt (单纯的 Rocky Linux)
系统软件包	Rocky Linux 社区	Rocky Linux 社区
注册	Salt	Salt
安装软件包	Salt	Salt
应用补丁	Salt	Salt
远程命令	Salt	Salt
系统软件包状态	Salt	Salt
系统自定义状态	Salt	Salt
组自定义状态	Salt	Salt
组织自定义状态	Salt	Salt
系统集管理器 (SSM)	Salt	Salt
产品迁移	不适用	不适用
基本虚拟 Guest 管理 *	Salt	Salt
高级虚拟 Guest 管理 *	Salt	Salt
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	Salt	Salt
系统部署 (PXE/Kickstart)	Salt	Salt
系统重新部署 (Kickstart)	Salt	Salt
联系方法	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	Salt	Salt
操作链	Salt	Salt
暂存（预先下载软件包）	Salt	Salt
重复软件包报告	Salt	Salt
CVE 审计	Salt	Salt
SCAP 审计	Salt	Salt

功能	Rocky Linux 8	Rocky Linux 9
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	Salt	Salt
快照和配置文件	Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步
电源管理	Salt	Salt
监视服务器	✗	✗
受监视客户端	Salt	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	✗	✗
重复性操作	Salt	Salt
AppStream	✓	✓
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。

1.14. 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
- Traditional 该功能仅在传统客户端上受支持
- Salt 该功能仅在 Salt 客户端上受支持。

表格 14. Ubuntu 操作系统上支持的功能

功能	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
客户端	✓	✓	✓
系统软件包	Canonical	Canonical	Canonical
注册	Salt	Salt	Salt
安装软件包	Salt	Salt	Salt
应用补丁	✓	✓	✓
远程命令	Salt	Salt	Salt
系统软件包状态	Salt	Salt	Salt
系统自定义状态	Salt	Salt	Salt
组自定义状态	Salt	Salt	Salt
组织自定义状态	Salt	Salt	Salt
系统集管理器 (SSM)	Salt	Salt	Salt
产品迁移	不适用	不适用	不适用
基本虚拟 Guest 管理 *	Salt	Salt	Salt
高级虚拟 Guest 管理 *	Salt	Salt	Salt
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	Salt	Salt	Salt
系统部署 (PXE/Kickstart)	✗	✗	✗
系统重新部署 (Kickstart)	✗	✗	✗
联系方法	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
使用 Uyuni Proxy	Salt	Salt	Salt
操作链	Salt	Salt	Salt
暂存（预先下载软件包）	Salt	Salt	Salt
重复软件包报告	Salt	Salt	Salt
CVE 审计	?	?	?
SCAP 审计	?	?	?
软件包校验	✗	✗	✗
软件包锁定	✓	✓	✓

功能	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
系统锁定	✗	✗	✗
系统快照	✗	✗	✗
配置文件管理	Salt	Salt	Salt
软件包配置文件	Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步	Salt: 支持配置文件，不支持同步
电源管理	✓	✓	✓
监视	Salt	Salt	Salt
Docker buildhost	?	?	?
构建含操作系统的 Docker 映像	Salt	Salt	Salt
Kiwi buildhost	✗	✗	✗
构建含操作系统的 Kiwi 映像	✗	✗	✗
重复性操作	Salt	Salt	Salt
AppStream	不适用	不适用	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。

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。

- 有关安装 Uyuni 的详细信息，请参见 [Installation-and-upgrade > Install-uyuni](#)。
- 有关设置 Uyuni 的详细信息，请参见 [Installation-and-upgrade > Uyuni-server-setup](#)。

2.1. 软件通道

通道是一种用于对软件包分组的方法。软件包由软件源提供，而软件源与通道关联。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。

当您在 Uyuni Web UI 中管理 AppStream 软件包时，您可能会注意到系统会显示相互矛盾的软件包更新建议。这是由于 Uyuni 无法正确解释模块化元数据。您可以使用内容生命周期管理 (CLM) AppStream 过滤器将 AppStream 软件源转换为非模块化软件源，以在执行某些升级操作时使用。有关 CLMR AppStream 过滤器的详细信息，请参见 Administration > Content-lifecycle-examples。

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 是可选软件包软件源，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 软件源。

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. 软件软件源

软件源用于收集软件包。如果您有权访问软件软件源，便可以安装软件源提供的任何软件。在 Uyuni 中，必须至少有一个软件源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软件源关联。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软件源。

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

2.1.5.1. 本地软件源位置

您可以在 Salt 客户端上配置本地软件源，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软件源。本地软件源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软件源。

对于 Salt 客户端，每次执行通道状态时都将禁用本地软件源。例如，当应用 highstate 或执行软件包操作时。

如果初始配置后本地软件源应保持启用状态，则必须为受影响的 Salt 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软件源：

表格 15. 本地软件源位置

Client Operating System	Local Repository Directory
SUSE Linux Enterprise Server	<code>/etc/zypp/repos.d</code>
openSUSE	<code>/etc/zypp/repos.d</code>
Red Hat Enterprise Linux	<code>/etc/yum.repos.d/</code>
CentOS	<code>/etc/yum.repos.d/</code>
Ubuntu	<code>/etc/apt/sources.list.d/</code>
Debian	<code>/etc/apt/sources.list.d/</code>

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。**显示产品** 的 **通道** 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 **包括建议** 项开关会打开。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。

有关详细信息，请参见 **Installation-and-upgrade > Setup-wizard**。

2.2. 引导软件源

引导软件源包含在引导期间注册 Salt 或传统客户端所需的软件包，以及在客户端上安装 Salt 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软件源。

2.2.1. 准备创建引导软件源

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软件源。

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，然后单击与软件源关联的通道。
2. 导航到 **软件源** 选项卡，然后单击 **同步** 并选中 **同步状态**。

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软件源的创建方式。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软件源，而您可以将其配置为始终从空软件源开始。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

默认会启用引导软件源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。该数据文件储存在 `/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. 手动生成引导软件源

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

过程：生成 SUSE Linux Enterprise 的引导软件源

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软件源的可用发行套件：

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

2. 创建引导软件源，并使用适当的软件源名称作为产品标签：

```
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。

客户端软件源位于 `/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，在创建引导软件源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：指定引导软件源的父通道

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软件源

如果您要创建包含多个不同体系结构的引导软件源，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软件源 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软件源时，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，则自动创建引导软件源可能会失败。在这种情况下，您需要手动创建软件源。

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

2.3. 激活密钥

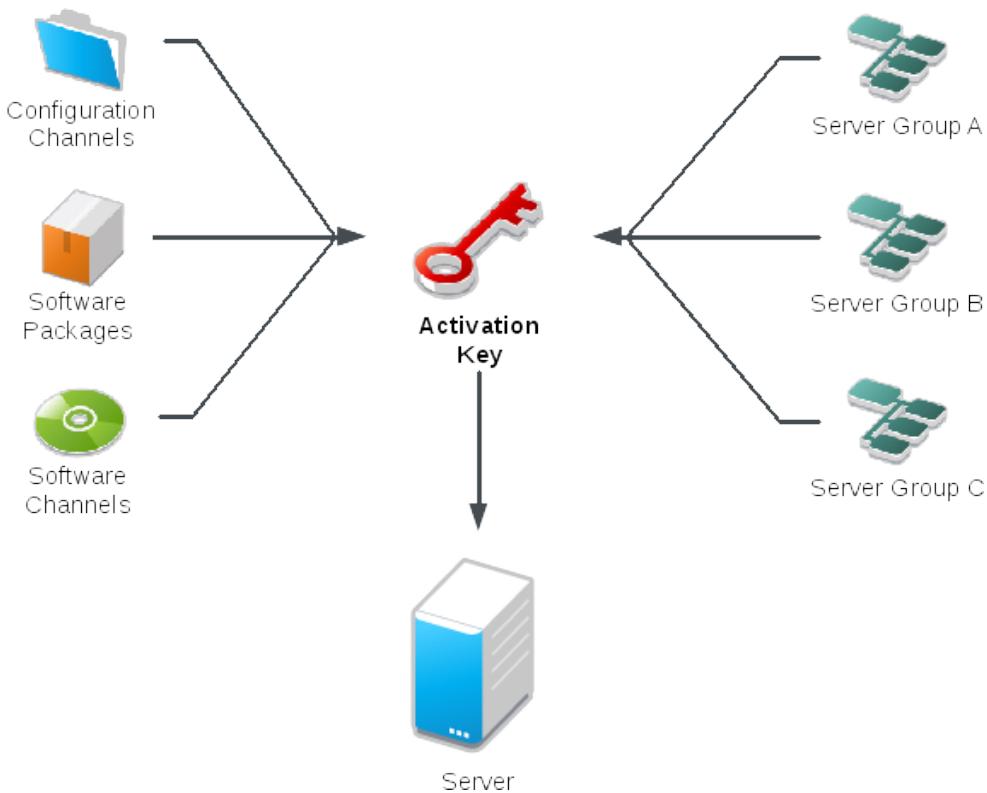
将激活密钥与传统客户端和 Salt 客户端搭配使用可确保您的客户端拥有正确的软件权利、可连接到适当的通道以及订阅相关的组。每个激活密钥都绑定到一个组织，您可以在创建密钥时设置此项。

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
- 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 **激活密钥细节** 页面的 **说明** 字段中，输入激活密钥的说明。
4. 在 **密钥** 字段中，输入激活密钥的名称。例如，**SLES15-SP4** 表示 SUSE Linux Enterprise Server 15 SP4。



Do not use commas in the **Key** field for any SUSE products. However, you **must** use commas for Red Hat Products.

5. 在 **基础通道** 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 **附加系统类型** 复选框。
8. 建议您保留 **默认** 设置的 **联系方法**。
9. 建议您将 **统一默认** 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 **配置文件部署** 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，**配置文件部署** 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。

2.3.1. 组合多个激活密钥

在传统客户端上执行引导脚本时，可以组合多个激活密钥。组合密钥可让您更好地控制系统上可以安装哪些产品，并可减少大型或复杂环境密钥的重复几率。



组合激活密钥仅可用于传统客户端。Salt 客户端不支持组合激活密钥。如果您对 Salt 客户端使用组合密钥，系统只会使用第一个密钥。

您可以在命令提示符处或在单个自动安装配置文件中指定多个激活密钥。

At the command prompt on the Uyuni Server, use the `rhnreg_ks` command, and separate the key names with a comma. To specify multiple keys in a Kickstart profile, navigate to **Systems > Autoinstallation** and edit the profile you want to use.

在组合激活密钥时需小心，因为如果某些值相互冲突，则可能会导致客户端注册失败。开始前，请检查以下值，确保它们未包含冲突信息：

- 软件包
- 软件子通道
- 配置通道。

如果检测到冲突，系统的处理方式如下：

- 基础软件通道中的冲突：注册失败。
- 系统类型中的冲突：注册失败。
- 启用配置标志中的冲突：启用配置管理。
- 如果其中一个密钥为系统特定密钥：注册失败。

2.3.2. 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 **系统**，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**生成新密钥**以创建重新激活密钥。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。

对于 Salt 客户端，当您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 `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](#)。

对于传统客户端，当您创建重新激活密钥后，可结合 `rhnreg_ks` 命令行实用程序使用该密钥。此命令会重新注册客户端并恢复其 Uyuni 设置。在传统客户端上，您可以将重新激活密钥与激活密钥进行合并，以便为单个系统配置文件聚合多个密钥的设置。例如：

```
rhnreg_ks --server=<server-url>/XMLRPC \
--activationkey=<reactivation-key>,<activationkey> \
--force
```



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。

2.3.3. 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。

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 **软 件 包** 部分中的所有软件包。
- **配 置** 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
```



对于任何SUSE产品，请不要在 **密钥** 字段中使用逗号。不过，对于RedHat产品，则必须 使用逗号。有关详细信息，请参见 [Reference > Systems](#)。

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，还会使用已签名元数据。

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。

Salt clients use now GPG key information entered for a software channel to manage the trusted keys. When a software channel with GPG key information is assigned to a client, the key gets trusted as soon as the channel is refreshed or the first package gets installed from this channel.

作为软件通道集的 GPG 密钥 URL 必须存在。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。

The GPG keys for the Client Tools Channels of Red Hat based clients are deployed on the client into [**/etc/pki/rpm-gpg/**](#) and can be referenced with file URLs.