

Uyuni 2026.01

安装和升级指南



U Y U N I

章 1. 前言

Installation, Deployment and Upgrade + Uyuni 2026.01

本指南提供了部署、升级和管理 Uyuni 服务器及代理的详尽分步说明。

指南分为以下几个章节：

- **要求**: 概述确保安装流程顺畅进行的必要硬件、软件和网络前提条件。
 - **部署与安装**: 指导您将 Uyuni 部署为容器并完成初始配置。
 - **升级与迁移**: 详细说明升级和迁移 Uyuni 的过程，同时最大限度减少停机时间。
 - **基本的服务器管理**: 涵盖基本的服务器操作，帮助您高效上手使用 Uyuni。

发布日期: 2026-01-28

+ + + + + + + + + + + + + + + + + + + +

目录

| | |
|---|----|
| 1. 前言 | 1 |
| 2. 要求 | 3 |
| 2.1. 一般要求 | 3 |
| 2.1.1. 服务器要求 | 3 |
| 2.1.2. 代理要求 | 3 |
| 2.2. 网络要求 | 4 |
| 2.2.1. 完全限定的域名 (FQDN) | 4 |
| 2.2.2. 主机名和 IP 地址 | 5 |
| 2.2.3. 重新启用路由器通告 | 5 |
| 2.2.4. 在 HTTP 或 HTTPS OSI 七层代理后部署 | 6 |
| 2.2.5. 隔离的部署 | 7 |
| 2.2.6. 所需的网络端口 | 7 |
| 2.3. 公有云要求 | 12 |
| 2.3.1. 网络要求 | 13 |
| 2.3.2. 准备存储卷 | 13 |
| 3. 部署和安装 | 15 |
| 3.1. 安装 Uyuni 服务器 | 15 |
| 3.1.1. Uyuni Server Deployment on openSUSE Tumbleweed | 15 |
| 3.1.2. Default paths expected by mgrpxy parameters (adjust/move your files accordingly) | 45 |
| 3.1.3. Apache tuning: /etc/uyuni/proxy/apache.conf | 45 |
| 3.1.4. Squid tuning: /etc/uyuni/proxy/squid.conf | 46 |
| 3.1.5. With tuning files | 46 |
| 3.1.6. If you have no tuning files, remove the tuning parameters: | 46 |
| 3.1.7. mgrpxy install podman | 46 |
| 3.1.8. mggradm restart | 53 |
| 3.1.9. mggradm start | 53 |
| 3.1.10. mggradm stop | 53 |

章 2. 要求

2.1. 一般要求

下表指定了服务器和代理的最低要求。



请勿使用 NFS 存储数据，因为它不支持 SELinux 文件标记。

2.1.1. 服务器要求

表格 1. x86-64 体系结构的服务器要求

| Software and Hardware | Details | Recommendation |
|-----------------------|--------------------------------|--|
| Tumbleweed | Clean installation, up-to-date | Tumbleweed |
| CPU | - | Minimum 4 dedicated 64-bit CPU cores (x86-64) |
| RAM | Test or Base Installation | Minimum 16 GB |
| | Production Server | Minimum 32 GB |
| Disk Space | / (root directory) | Minimum 40 GB |
| | /var/lib/pgsql | Minimum 50 GB |
| | /var/spacewalk | Minimum storage required:
100 GB (this will be verified by the implemented check)

* 每个 SUSE 产品和软件包中心
50 GB

为每个 Red Hat 产品提供 360 GB
空间 |
| | /var/cache | 至少 10 GB 空间。为每个 SUSE 产品增加 100 MB 空间，为每个 Red Hat 或其他产品增加 1 GB 空间。
如果服务器为 ISS 主服务器，则空间需要翻倍。 |
| | 交换空间 | 3 GB |

2.1.2. 代理要求

表格 2. 代理要求

| Software and Hardware | Details | Recommendation |
|-----------------------|--------------------------------|--------------------------------------|
| Tumbleweed | Clean installation, up-to-date | Tumbleweed |
| CPU | | Minimum 2 dedicated 64-bit CPU cores |
| RAM | Test Server | Minimum 2 GB |
| | Production Server | Minimum 8 GB |
| Disk Space | / (root directory) | Minimum 40 GB |
| | /srv | Minimum 100 GB |
| | /var/cache (Squid) | Minimum 100 GB |

Uyuni Proxy 将软件包缓存在 `/var/cache/` 目录中。如果 `/var/cache/` 中的可用空间不足，代理将去除旧的未使用软件包，并将其替换为较新的软件包。

鉴于这种行为：

- 代理上的 `/var/cache/` 目录越大，代理与 Uyuni 服务器之间的流量就越少。
- 使代理上的 `/var/cache/` 目录与 Uyuni 服务器上的 `/var/spacewalk/` 保持相同的大小，可以避免在首次同步后出现大量的流量。
- Uyuni 服务器上的 `/var/cache/` 目录相比代理上的目录可能较小。有关大小估算的指导，请参见 [server-hardware-requirements] 一节。

2.2. 网络要求

本节详细说明 Uyuni 的网络和端口要求。

IP 转发将通过容器化安装来实现。这意味着 Uyuni 服务器和代理将充当路由器。此行为由 podman 直接完成。如果禁用 IP 转发，podman 容器将不会运行。



您可以考虑根据您的策略实现 Uyuni 环境的网络隔离。

有关详细信息，请参见 <https://www.suse.com/support/kb/doc/?id=000020166>。

2.2.1. 完全限定的域名 (FQDN)

Uyuni 服务器必须正确解析其 FQDN。如果无法解析 FQDN，可能会导致许多不同的组件出现严重问题。

有关配置主机名和 DNS 的详细信息，请参见 <https://documentation.suse.com/sles/15-SP6/html/SLES-all/cha-network.html#sec-network-yast-change-host>。

2.2.2. 主机名和 IP 地址

为确保 Uyuni 域名可由其客户端解析，服务器和客户端计算机都必须连接到一台正常工作的 DNS 服务器。还需要确保正确配置反向查找。

有关设置 DNS 服务器的详细信息，请参见 <https://documentation.suse.com/sles/15-SP6/html/SLES-all/cha-dns.html>。

2.2.3. 重新启用路由器通告

当使用 **mgradm install podman** 或 **mgrpxy install podman** 安装 Uyuni 时，系统会设置 Podman，后者会启用 IPv4 和 IPv6 转发。这是容器与外部通信的必要条件。

然而，如果系统此前将 **/proc/sys/net/ipv6/conf/eth0/accept_ra** 设置为 **1**，则会停止使用路由器通告 (RA)。这将导致路由无法再通过 RA 获取，且默认 IPv6 路由将会缺失。

To recover correct functioning of the IPv6 routing, follow the procedure depending on whether:

- server and proxy are based on 15 SP7 (without Network manager)
- server and proxy are based on SL Micro 6.1} (with Network manager)

Procedure: Reenabling router advertisements without Network Manager

1. 在 **/etc/sysctl.d** 中创建文件，例如 **99-ipv6-ras.conf**。
2. 向文件中添加以下参数和值：

```
net.ipv6.conf.eth0.accept_ra = 2
```

3. 重引导。

Procedure: Reenabling router advertisements with Network Manager

1. List your connections with **nmcli connection show**.
2. Create or modify the file **/etc/NetworkManager/system-connections/<name of connection>.nmconnection** to add this setting:

```
[ipv6]
addr-gen-mode=eui64
```

3. 重引导。
4. The file should look similar to this:

```
[connection]
id=Wired connection 1
```

```

type=ether
interface-name=eth0

[ether]
[dhcpv4]
dns-priority=20
method=auto

[dhcpv6]
addr-gen-mode=eui64
method=auto

```

2.2.4. 在 HTTP 或 HTTPS OSI 七层代理后部署

有些环境强制要求通过 HTTP 或 HTTPS 代理（如 Squid 服务器或类似服务器）访问互联网。要在此类配置下允许 Uyuni 服务器访问互联网，需进行以下配置。

过程：配置 HTTP 或 HTTPS OSI 七层代理

- 要进行操作系统互联网访问配置，请根据需求修改 `/etc/sysconfig/proxy`：

```

PROXY_ENABLED="no"
HTTP_PROXY=""
HTTPS_PROXY=""
NO_PROXY="localhost, 127.0.0.1"

```

- 要进行 `podman` 容器互联网访问配置，请根据需求修改 `/etc/systemd/system/uyuni-server.service.d/custom.conf`。例如，设置以下内容：

```

[Service]
Environment=TZ=Europe/Berlin
Environment="PODMAN_EXTRA_ARGS="
Environment="https_proxy=user:password@http://192.168.10.1:3128"

```

- 要进行 Java 应用程序互联网访问配置，请根据需求修改 `/etc/rhn/rhn.conf`。在容器主机上，执行 `mgrctl term` 以打开服务器容器内的命令行：

- 修改 `/etc/rhn/rhn.conf`。例如，设置以下内容：

```

# 使用代理 FQDN 或 FQDN:端口
server.satellite.http_proxy =
server.satellite.http_proxy_username =
server.satellite.http_proxy_password =
# no_proxy 为逗号分隔的列表
server.satellite.no_proxy =

```

- 在容器主机上，重启服务器以强制应用新配置：

```
systemctl restart uyuni-server.service
```

2.2.5. 隔离的部署

如果您在内部网络中操作，无法访问 SUSE Customer Center，可以使用 **Installation-and-upgrade > Container-deployment**。

在生产环境中，Uyuni 服务器和客户端始终应使用防火墙。有关所需端口的完整列表，请参见 [installation-and-upgrade:network-requirements.pdf](#)。

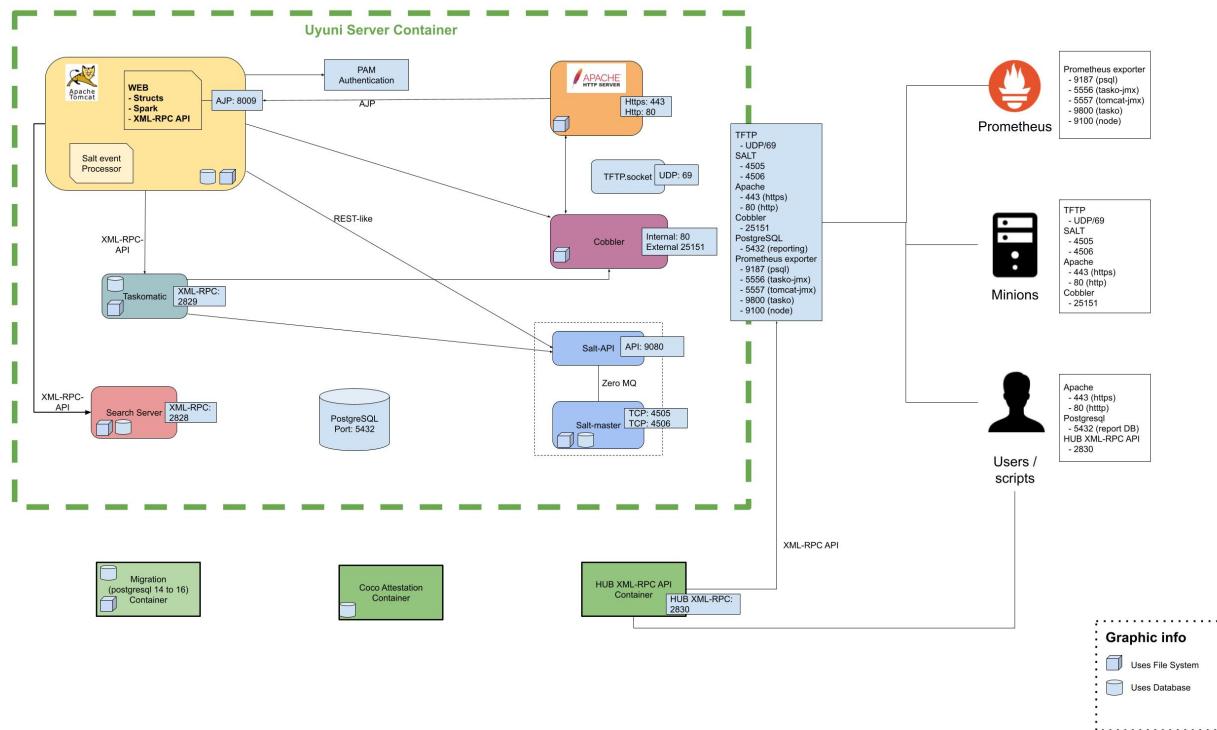
2.2.6. 所需的网络端口

本节提供了 Uyuni 中各种通信使用的端口的综合列表。

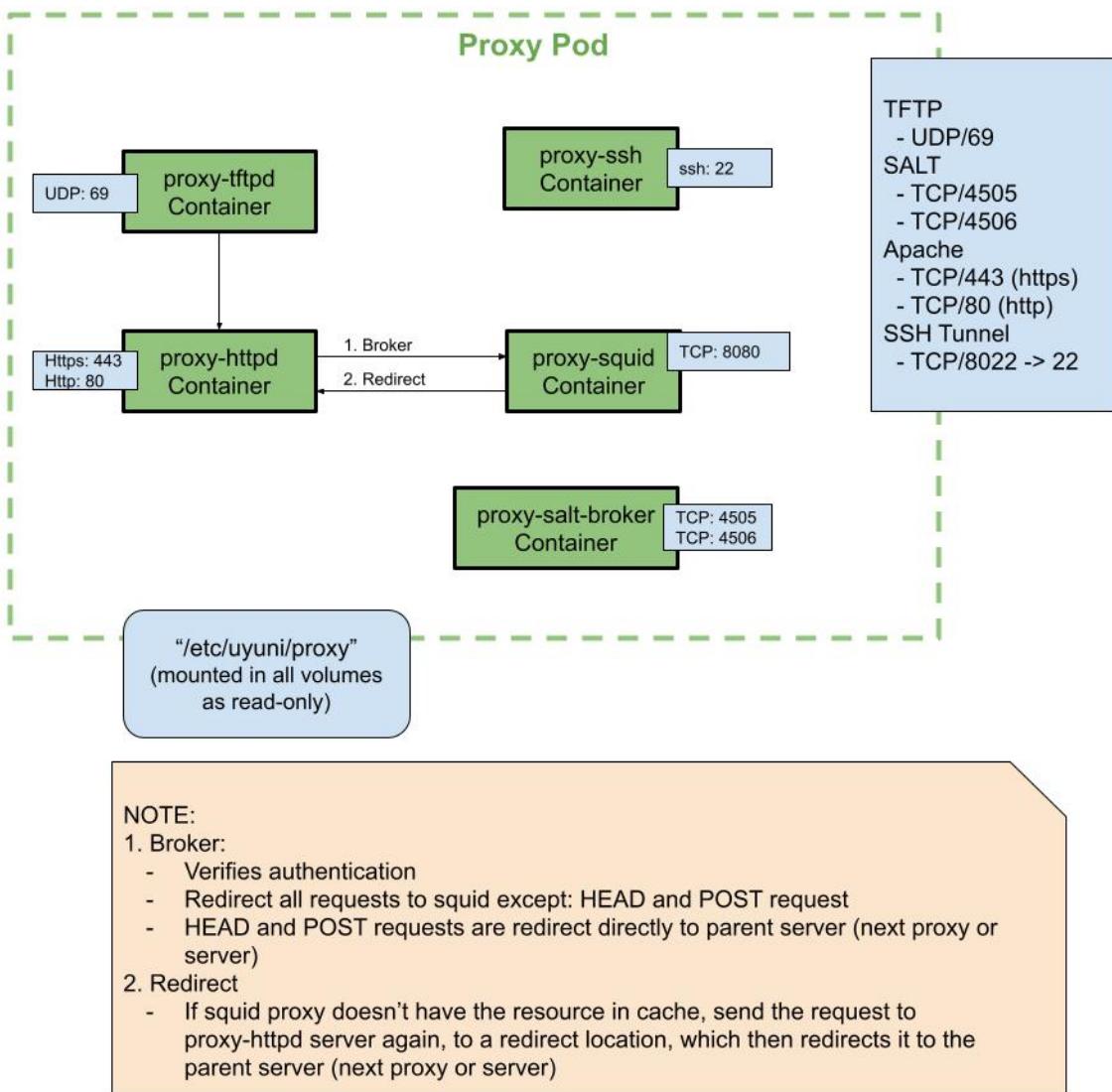
您不需要打开所有这些端口。某些端口只有在您使用需要这些端口的服务时才需打开。

2.2.6.1. 概述

2.2.6.1.1. 服务器



2.2.6.1.2. 代理



2.2.6.2. 外部入站服务器端口

必须打开外部入站端口，以在 Uyuni 服务器上配置防火墙用于防范未经授权访问服务器。

打开这些端口将允许外部网络流量访问 Uyuni 服务器。

表格 3. Uyuni Server 的外部端口要求

| 端口号 | 协议 | 使用方 | 备注 |
|-----|---------|------|---------------------------------|
| 67 | TCP/UDP | DHCP | 仅当客户端向服务器请求 IP 地址时才需要此端口。 |
| 69 | TCP/UDP | TFTP | 将服务器用作 PXE 服务器进行自动化客户端安装时需要此端口。 |

| 端口号 | 协议 | 使用方 | 备注 |
|-------|-----|------------|---|
| 80 | TCP | HTTP | 某些引导储存库以及进行自动化安装时需要临时使用此端口。 |
| 443 | TCP | HTTPS | 处理 Web UI、客户端、服务器和代理 (tftpsync) 请求。 |
| 4505 | TCP | salt | 接受来自客户端的通信请求时需要此端口。客户端会发起连接，且该连接将保持开启状态以接收 Salt 主控端发出的命令。 |
| 4506 | TCP | salt | 接受来自客户端的通信请求时需要此端口。客户端会发起连接，且该连接将保持开启状态以向 Salt 主控端报告结果。 |
| 5432 | TCP | PostgreSQL | 访问报告数据库时需要此端口。 |
| 5556 | TCP | Prometheus | 抓取 Taskomatic JMX 指标时需要此端口。 |
| 5557 | TCP | Prometheus | 抓取 Tomcat JMX 指标时需要此端口。 |
| 9100 | TCP | Prometheus | 抓取 节点导出器指标时需要此端口。 |
| 9187 | TCP | Prometheus | 抓取 PostgreSQL 指标时需要此端口。 |
| 9800 | TCP | Prometheus | 抓取 Taskomatic 指标时需要此端口。 |
| 25151 | TCP | Cobbler | |

2.2.6.3. 外部出站服务器端口

必须打开外部出站端口，以在 Uyuni 服务器上配置防火墙用于限制服务器可访问的内容。

打开这些端口将允许来自 Uyuni 服务器的网络流量与外部服务通信。

表格 4. Uyuni Server 的外部端口要求

| 端口号 | 协议 | 使用方 | 备注 |
|-------|-----|---------|--|
| 80 | TCP | HTTP | SUSE Customer Center 需要此端口。端口 80 不用于为 Web UI 传递数据。 |
| 443 | TCP | HTTPS | SUSE Customer Center 需要此端口。 |
| 25151 | TCP | Cobbler | |

2.2.6.4. 内部服务器端口

内部端口由 Uyuni 服务器在内部使用。只能从 **localhost** 访问内部端口。

大多数情况下无需调整这些端口。

表格 5. Uyuni Server 的内部端口要求

| 端口号 | 备注 |
|-------|---|
| 2828 | Satellite-search API，由 Tomcat 和 Taskomatic 中的 RHN 应用程序使用。 |
| 2829 | Taskomatic API，由 Tomcat 中的 RHN 应用程序使用。 |
| 8005 | Tomcat 关机端口。 |
| 8009 | Tomcat 到 Apache HTTPD (AJP)。 |
| 8080 | Tomcat 到 Apache HTTPD (HTTP)。 |
| 9080 | Salt-API，由 Tomcat 和 Taskomatic 中的 RHN 应用程序使用。 |
| 25151 | Cobbler 的 XMLRPC API |
| 32000 | 与运行 Taskomatic 和 satellite-search 的 Java 虚拟机 (JVM) 建立 TCP 连接时使用此端口。 |

32768 和更高的端口用作临时端口。这些端口往往用于接收 TCP 连接。收到 TCP 连接请求后，发送方将选择其中一个临时端口号来与目标端口进行匹配。

可使用以下命令来确定哪些端口是临时端口：

```
cat /proc/sys/net/ipv4/ip_local_port_range
```

2.2.6.5. 外部入站代理端口

必须打开外部入站端口，以在 Uyuni Proxy 上配置防火墙用于防范未经授权访问代理。

打开这些端口将允许外部网络流量访问 Uyuni Proxy。

表格 6. Uyuni Proxy 的外部端口要求

| 端口号 | 协议 | 使用方 | 备注 |
|------|---------|-------|---|
| 22 | | | 仅当用户需要通过 Salt SSH 管理代理主机时才需要此端口。 |
| 67 | TCP/UDP | DHCP | 仅当客户端向服务器请求 IP 地址时才需要此端口。 |
| 69 | TCP/UDP | TFTP | 将服务器用作 PXE 服务器进行自动化客户端安装时需要此端口。 |
| 443 | TCP | HTTPS | Web UI、客户端、服务器和代理 (tftpsync) 请求。 |
| 4505 | TCP | salt | 接受来自客户端的通信请求时需要此端口。客户端会发起连接，且该连接将保持开启状态以接收 Salt 主控端发出的命令。 |

| 端口号 | 协议 | 使用方 | 备注 |
|------|-----|------|---|
| 4506 | TCP | salt | 接受来自客户端的通信请求时需要此端口。客户端会发起连接，且该连接将保持开启状态以向 Salt 主控端报告结果。 |
| 8022 | | | 使用 ssh-push 和 ssh-push-tunnel 联系方法时需要此端口。连接至代理的客户端会主动向服务器发起签入请求，并通过该端口跳转至目标客户端。 |

2.2.6.6. 外部出站代理端口

必须打开外部出站端口，以在 Uyuni Proxy 上配置防火墙用于限制代理可访问的内容。

打开这些端口将允许来自 Uyuni Proxy 的网络流量与外部服务通信。

表格 7. Uyuni Proxy 的外部端口要求

| 端口号 | 协议 | 使用方 | 备注 |
|------|-----|-------|-----------------------------|
| 80 | | | 用于访问服务器。 |
| 443 | TCP | HTTPS | SUSE Customer Center 需要此端口。 |
| 4505 | TCP | Salt | 直接或通过代理连接到 Salt 主控端时需要此端口。 |
| 4506 | TCP | Salt | 直接或通过代理连接到 Salt 主控端时需要此端口。 |

2.2.6.7. 外部客户端端口

必须打开外部客户端端口，以在 Uyuni 服务器及其客户端之间配置防火墙。

大多数情况下无需调整这些端口。

表格 8. Uyuni 客户端的外部端口要求

| 端口号 | 方向 | 协议 | 备注 |
|------|----|-----|---|
| 22 | 入站 | SSH | 使用 ssh-push 和 ssh-push-tunnel 联系方法时需要此端口。 |
| 80 | 出站 | | 用于访问服务器或代理。 |
| 443 | 出站 | | 用于访问服务器或代理。 |
| 4505 | 出站 | TCP | 直接或通过代理连接到 Salt 主控端时需要此端口。 |
| 4506 | 出站 | TCP | 直接或通过代理连接到 Salt 主控端时需要此端口。 |
| 9090 | 出站 | TCP | Prometheus 用户界面需要此端口。 |
| 9093 | 出站 | TCP | Prometheus 警报管理器需要此端口。 |

| 端口号 | 方向 | 协议 | 备注 |
|------|----|-----|------------------------------|
| 9100 | 出站 | TCP | Prometheus 节点导出器需要此端口。 |
| 9117 | 出站 | TCP | Prometheus Apache 导出器需要此端口。 |
| 9187 | 出站 | TCP | Prometheus PostgreSQL 需要此端口。 |

2.2.6.8. 所需的 URL

Uyuni 必须能够访问某些 URL 才能注册客户端和执行更新。大多数情况下，允许访问以下 URL 便已足够：

- **scc.suse.com**
- **updates.suse.com**
- **installer-updates.suse.com**
- **registry.suse.com**
- **registry-storage.suse.com**
- **opensuse.org**

此外，您可能需要访问以下非 SUSE 产品的 URL：

- **download.nvidia.com**
- **public.dhe.ibm.com**
- **nu.novell.com**

您可以在以下文章中找到有关将指定 URL 及其关联 IP 地址列入白名单的更多详细信息：[访问受防火墙和/或代理保护的 SUSE Customer Center 和 SUSE 仓库](#)。

如果您正在使用非 SUSE 客户端，则还可能需要允许访问为这些操作系统提供特定软件包的其他服务器。例如，如果您使用的是 Ubuntu 客户端，则需要能够访问 Ubuntu 服务器。

有关为非 SUSE 客户端排查防火墙访问权限问题的详细信息，请参见 [Administration > Troubleshooting](#)。

2.3. 公有云要求

本节介绍在公有云基础结构上安装 Uyuni 所要满足的要求。我们已在 Amazon EC2、Google Compute Engine 和 Microsoft Azure 上对这些指令进行过测试，不过它们进行一定修改后在其他提供商的云服务上也应能正常工作。

在开始之前，请注意以下一些事项：

- Uyuni 设置过程执行正向确认的反向 DNS 查找。此操作必须成功，设置过程才能完成，并且 Uyuni 才能按预期方式运行。请务必在设置 Uyuni 之前执行主机名和 IP 配置。
- Uyuni Server 和 Proxy 实例需在适当的网络配置中运行，该网络配置可让您控制 DNS 项，但无法通过因特网自由访问。

- 在此网络配置中必须提供 DNS 解析：**hostname -f** 必须返回完全限定的域名 (FQDN)。
- DNS 解析对于连接客户端也很重要。
- DNS 取决于所选的云框架。有关详细说明，请参见云提供商文档。
- 我们建议将软件储存库、服务器数据库和代理 squid 缓存存储在外部虚拟磁盘上。这可以防止在实例意外终止时丢失数据。本节包含有关设置外部虚拟磁盘的说明。

2.3.1. 网络要求

在公有云上使用 Uyuni 时，必须使用受限制的网络。我们建议使用带有适当防火墙设置的 VPN 专用子网。只能允许指定 IP 范围内的计算机访问该实例。



在公有云上运行 Uyuni 意味着需要实施强大的安全措施。限制、过滤、监控并审计对实例的访问至关重要。SUSE 强烈建议不要配置全球均可访问但缺少充足边界安全保护的 Uyuni 实例。

要访问 Uyuni Web UI，请在配置网络访问控制时允许 HTTPS。这将允许您访问 Uyuni Web UI。

在 EC2 和 Azure 中，创建一个新安全组，并添加 HTTPS 入站和出站规则。在 GCE 中，选中**防火墙**部分下的**允许 HTTPS 流量**复选框。

2.3.2. 准备存储卷

我们建议将 Uyuni 的储存库和数据库存储在不同于根卷的存储设备上。这有助于避免丢失数据，有时还可以提高性能。

Uyuni 容器利用默认存储位置。应在部署之前为自定义存储配置这些位置。有关详细信息，请参见[Installation-and-upgrade > Container-management](#)



不要使用逻辑卷管理 (LVM) 进行公有云安装。

用于存储储存库的磁盘大小取决于您要使用 Uyuni 管理的发行套件和通道数目。挂接虚拟磁盘时，它们将作为 Unix 设备节点显示在实例中。设备节点的名称因提供商及所选实例类型而异。

确保 Uyuni 服务器的根卷大小不少于 100 GB。如果可能，请另外添加一个 500 GB 或以上大小的存储磁盘，并选择 SSD 存储类型。当您的实例启动时，Uyuni 服务器的云映像会使用脚本来指派这个单独的卷。

启动实例后，您便可登录 Uyuni 服务器，并使用以下命令查找所有可用的存储设备：

```
hwinfo --disk | grep -E "Device File:"
```

如果您不确定应选择哪个设备，可使用 **lsblk** 命令查看每个设备的名称和大小。请选择与要寻找的虚拟磁盘大小匹配的名称。

可以使用 **mgr-storage-server** 命令设置外部磁盘。这会创建一个挂载到 **/manager_storage** 的 XFS 分区，并使用它作为存储数据库和储存库的位置：

```
/usr/bin/mgr-storage-server <devicename>
```

章 3. 部署和安装

3.1. 安装 Uyuni 服务器

部署 Uyuni 服务器的场景多种多样。

3.1.1. Uyuni Server Deployment on openSUSE Tumbleweed

3.1.1.1. 部署准备工作

本节介绍有关设置和部署 Uyuni 服务器的专业知识。过程包括安装 **Podman** 和 **Uyuni 容器实用程序**、进行部署，然后通过 **mgrctl** 开始与容器交互。



This section assumes you have already configured an openSUSE Tumbleweed host server, whether it is running on a physical machine or within a virtual environment.

<https://download.opensuse.org/tumbleweed/>

3.1.1.2. 容器主机一般要求

有关一般要求，请参见[Installation-and-upgrade > General-requirements](#)。

An openSUSE Tumbleweed server should be installed from installation media.

<https://download.opensuse.org/tumbleweed/>

下面介绍此过程。

3.1.1.3. 容器主机要求

有关 CPU、RAM 和存储要求，请参见[Installation-and-upgrade > Hardware-requirements](#)。



为了保证客户端能够解析 FQDN 域名，容器化服务器和主机都必须连接到正常运行的 DNS 服务器。此外，必须确保反向查找的配置正确。

3.1.1.4. 安装用于容器的 Uyuni 工具

Procedure: Installing Uyuni Tools on openSUSE Tumbleweed

1. On your local host, open a terminal window and log in.
2. Add the following repository to your openSUSE Tumbleweed server.
You might need to use **sudo** for the following commands.

```
zypper ar
https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable/images/repo/Uyuni-Server-POOL-$(arch)-Media1/ uyuni-server-stable
```

3. Refresh the repository list and import the key:

```
zypper ref
```

When prompted, trust and import the new repository GPG key.

4. 安装容器工具：

```
zypper in mgradm mgrctl mgradm-bash-completion mgrctl-bash-completion uyuni-storage-setup-server
```

有关 Uyuni 容器实用程序的详细信息，请参见 [Uyuni 容器实用程序](#)。

3.1.1.5. 配置自定义永久性存储

此步骤是可选的。但是，如果您的基础架构需要自定义的永久性存储，请使用 **mgr-storage-server** 工具。

有关详细信息，请参见 **mgr-storage-server --help**。此工具可以简化容器存储和数据库卷的创建。

如下所示使用命令：

```
mgr-storage-server <storage-disk-device> [<database-disk-device>]
```

3.1.1.6. 例如：

```
mgr-storage-server /dev/nvme1n1 /dev/nvme2n1
```

[NOTE]

====

此命令将在 [path]``/var/lib/containers/storage/volumes`` 中创建永久性存储卷。

有关详细信息，请参见 [xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc](#)[]。

====

-- 使用 Podman 部署 Uyuni 容器

==== [command]``mgradm`` 概述

使用 [command]``mgradm`` 工具将 {productname} 部署为容器。可使用两种方法将 {productname} 服务器部署为容器。本节重点介绍基本容器部署。

有关使用自定义配置文件进行部署的信息，请参见 [xref:installation-and-upgrade:container-management/mgradm-yaml-custom-configuration.adoc](#)[]。

可以在命令行中运行 [command]``mgradm --help`` 来了解更多信息。

:leveloffset: +2

[NOTE]

=====
经过安全加固的 {productname} 服务器主机可能会限制从 [path]``/tmp`` 文件夹执行文件。在这种情况下，可通过以下临时解决方案规避：在运行 [command]``mgradm`` 前，将 [literal]``TMPDIR`` 环境变量导出至其他现有路径。

例如：

[source, shell]

```
export TMPDIR=/path/to/other/tmp
```

在 {productname} 的后续更新中，相关工具将进行优化，届时无需再使用此临时解决方案。

:leveloffset: 3

- . 过程：使用 Podman 部署 Uyuni 容器
- . 在终端中以 sudo 或 root 用户身份运行以下命令。

+

[source, shell]

```
sudo mgradm install podman
```

+
[IMPORTANT]
=====

必须以 sudo 或 root 用户身份部署容器。如果您遗漏此步骤，终端中将显示以下错误。

[source, shell]

```
INF Setting up uyuni network 9:58AM INF Enabling system service 9:58AM FTL Failed to open /etc/systemd/system/uyuni-server.service for writing error="open /etc/systemd/system/uyuni-server.service: permission denied"
```

=====

- . 等待部署完成。
- . 打开浏览器并访问您的服务器 FQDN。

//In this section you learned how to deploy an {productname} Server container.

==== 永久性卷

许多用户希望指定其永久性卷的位置。

[NOTE]

=====

如果您只是在试用 {productname}，则无需指定这些卷。默认情况下，[command]``mgradm`` 会设置正确的卷。

通常只需为较大规模的生产部署指定卷位置。

=====

默认情况下，[command]``podman`` 将其卷存储在 [path]``/var/lib/containers/storage/volumes`` 中。

可以通过将磁盘挂载到此路径或文件系统中的预期卷路径（例如 [path]``/var/lib/containers/storage/volumes/var-spacewalk``）来为卷提供自定义存储配置。这一点对于数据库和软件包镜像尤为重要。

有关容器中所有永久性卷的列表，请参见：

- * [xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc\[\]](#)
- * [xref:administration:troubleshooting/tshoot-container-full-disk.adoc\[\]](#)

:leveloffset!:
:leveloffset: +3

= {productname} 服务器隔离的部署

:revdate: 2025-07-25

:page-revdate: {revdate}

== 什么是隔离的部署？

隔离部署是指设置和操作与不安全网络（尤其是互联网）隔离的任何联网系统。这种部署通常用于军事设施、金融系统、关键基础架构等高安全性环境，以及处理敏感数据，因而必须防范其受到外部威胁的任何位置。

You can easily pull container images using [systemitem]``Podman`` or [systemitem]``Docker`` on a machine with internet access.

.Procedure: Pulling the images

. Pull the desired images, then save the images as a [literal]``tar`` archive. For example:

+
.Podman

```
podman pull registry.opensuse.org/uyuni/server:latest registry.opensuse.org/uyuni/server-postgresql:latest podman save --output images.tar registry.opensuse.org/uyuni/server:latest registry.opensuse.org/uyuni/server-postgresql:latest
```

+
.Docker

```
docker pull registry.opensuse.org/uyuni/server:latest registry.opensuse.org/uyuni/server-postgresql:latest docker save --output images.tar registry.opensuse.org/uyuni/server:latest registry.opensuse.org/uyuni/server-postgresql:latest
```

+
. Transfer the resulting [filename]``images.tar`` to the Server container host and load it using the following command:
+
.加载服务器映像

```
podman load -i images.tar
```

== Obtaining container images for {salt} formulas in air-gapped environments

Some formulas, like Bind and DHCP (Kea), also use containers. If you plan to use them in an air-gapped environment, you need to pull their images, save them to an archive, and

load them on your {productname} Proxy or another managed system.

The images are available from `registry.opensuse.org`.

.Procedure: Obtaining formula images for air-gapped environments

. On a system with Internet access, pull the required images. For example:

+

[source, shell]

```
podman pull registry.opensuse.org/opensuse/bind:latest podman pull
registry.opensuse.org/opensuse/kea:latest
```

. Save the images to a TAR archive:

+

[source, shell]

```
podman save -o formula-images.tar registry.opensuse.org/opensuse/bind:latest
registry.opensuse.org/opensuse/kea:latest
```

. Transfer the `formula-images.tar` file to your air-gapped system.

. Load the images on the air-gapped system:

+

[source, shell]

```
podman load -i formula-images.tar
```

==== Deploy {productname} on {opensuse} {tumbleweed}

{productname} 还在 RPM 软件包中提供了可在系统上安装的全部所需容器映像。

[NOTE]

=====

用户应在内部网络上提供所需的 RPM。这可以通过使用第二个 {productname} 服务器或任意类型的镜像来完成。

=====

.Procedure: Install {productname} on {opensuse} {tumbleweed} in air-gapped environment

. Install {opensuse} {tumbleweed}.

. 更新系统。

. 安装工具软件包和映像软件包（将 \$ARCH\$ 替换为适当的体系结构）：

+

[source, shell]

```
zypper install mgradm* mgrctl* uyuni-server*-image*
```

+

. 使用 [command]``mgradm`` 部署 {productname}。在隔离环境中，可能需要使用 [command]``--pullPolicy Never`` 选项。

For more detailed information about installing {productname} Server on {opensuse} {tumbleweed}, see [xref:container-deployment/uyuni/server-deployment-uyuni.adoc\[Server Deployment\]](#).

要升级 {productname} 服务器，用户应升级系统中的所有软件包，并按照xref:container-management/updating-server-containers.adoc[服务器升级]中定义的过程操作。

```
:leveloffset: 3
:leveloffset: +2

[[installation-proxy]]
= 安装 {productname} 代理
:revdate: 2025-02-19
:page-revdate: {revdate}

// **This file is needed to link generically to proxy installation**
```

部署 {productname} 代理的场景多种多样。所有这些场景都假定您已成功部署 {productname} {productnumber} 服务器。

```
:leveloffset: 3
:leveloffset: +3

[[proxy-setup-containers-uyuni]]
= 容器化 {productname} Proxy 设置
:revdate: 2025-06-30
:page-revdate: {revdate}
```

为 {productname} Proxy 容器准备好容器主机后，需要额外执行几步容器设置才能完成配置。

.过程

- 生成 {productname} Proxy 配置归档文件
- 将配置归档传输到在安装步骤中准备的容器主机并解压缩
- 使用 [literal]``mgrpxy`` 启动代理服务

== 生成代理配置

{productname} 代理的配置归档由 {productname} 服务器生成。每个附加代理都需要自身的配置归档。

对于容器化 {productname} 代理，您必须构建新的代理配置文件，然后重新部署容器以使更改生效。此流程适用于更新设置（包括 SSL 证书）。

```
//[NOTE]
//=====
//2 GB represents the default proxy squid cache size.
//This will need to be adjusted for your environment.
//=====
```

[IMPORTANT]

=====

对于 Podman 部署，在生成此代理配置之前，必须将 {productname} 代理的容器主机作为客户端注册到 {productname} 服务器。

=====

如果使用代理 FQDN 生成非注册客户端的代理容器配置（如 Kubernetes 用例中那样），系统列表中将出现一个新的系统项。此新项将显示在之前输入的“代理 FQDN”值下方并属于[literal]``外部``系统类型。

[NOTE]

=====

Peripheral servers are always using third-party SSL certificates. If the hub server has generated the certificates for the peripheral server, it needs to generate the certificate of each proxy too.

On the hub server, run the following command.

[source, shell]

```
mgrctl exec -ti --rhn-ssl-tool --gen-server --dir="/root/ssl-build" --set-country="COUNTRY" \ --set-state="STATE" --set-city="CITY" --set-org="ORGANIZATION" \ --set-org-unit="ORGANIZATION UNIT" --set-email="name@example.com" \ --set-hostname=PROXY --set-cname="proxy.example.com"
```

The files to use will be

- . [litteral]``/root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT`` as the root CA,
- . [litteral]``/root/ssl-build/<hostname>/server.crt`` as the proxy certificate and
- . [litteral]``/root/ssl-build/<hostname>/server.key`` as the proxy certificate's key.

====

// tag::generate-proxy-config-section[]
==== 使用 {webui} 生成代理配置

. 过程: 使用 {webui} 生成代理容器配置
[role=procedure]

- . 在 {webui} 中, 导航到menu:系统[代理配置], 然后填写所需数据:
- . 在[guimenu]``代理 FQDN``字段中, 键入代理的完全限定域名。
- . 在[guimenu]``父 FQDN``字段中, 键入 {productname} 服务器或另一个 {productname} 代理的完全限定域名。
- . 在[guimenu]``代理 SSH 端口``字段中, 键入 SSH 服务在 {productname} 代理上监听的 SSH 端口。建议保留默认值 8022。
- . In the [guimenu]``Max Squid cache size [MB]`` field type maximal allowed size for Squid cache. Recommended is to use at most 80% of available storage for the containers.

+

[NOTE]

====

2 GB 表示默认的代理 squid 缓存大小。需要根据您的环境调整此大小。

====

+

在 [guimenu]``SSH 证书``选择列表中, 选择应为 {productname} 代理生成新服务器证书还是使用现有证书。您可以考虑作为 {productname} 内置 (自我签名) 证书生成的证书。

+

然后根据所做的选择提供用于生成新证书的签名 CA 证书的路径, 或者要用作代理证书的现有证书及其密钥的路径。

+

服务器生成的 CA 证书存储在 [path]``/var/lib/containers/storage/volumes/root/_data/ssl-build`` 目录中。

+

有关现有或自定义证书的详细信息以及企业和中间证书的概念, 请参见 [xref:administration:ssl-](#)

`certs-imported.adoc[]。`

- . 单击 `btn:[生成]` 以在 `{productname}` 服务器中注册新代理 FQDN，并生成包含容器主机细节的配置归档 (`[filename]``config.tar.gz```)。
 - . 片刻之后，系统会显示文件可供下载。请将此文件保存在本地。
-

`==== 使用 [literal]``spacecmd`` 和自我签名证书生成代理配置`

可以使用 `[literal]``spacecmd``` 生成代理配置。

. 过程：使用 `spacecmd` 和自我签名证书生成代理配置
[role=procedure]

- . 通过 SSH 连接到您的容器主机。
- . 执行以下命令（替换其中的服务器和代理 FQDN）：

+

```
mgrctl exec -ti 'spacecmd proxy_container_config_generate_cert --dev-pxy.example.com dev-srv.example.com 2048 email@example.com -o /tmp/config.tar.gz'
```

- . 从服务器容器复制生成的配置：

+

```
mgrctl cp server:/tmp/config.tar.gz
```

`==== 使用 [literal]``spacecmd`` 和自定义证书生成代理配置`

可以使用 `[literal]``spacecmd``` 为自定义证书（而不是默认的自我签名证书）生成代理配置。

. 过程：使用 `spacecmd` 和自定义证书生成代理配置
[role=procedure]

- . 通过 SSH 连接到您的服务器容器主机。
- . Execute the following commands, replacing the Server and Proxy FQDN:

+

```
for f in ca.crt proxy.crt proxy.key; do mgrctl cp $f server:/tmp/$f done
mgrctl exec -ti 'spacecmd proxy_container_config --p 8022 pxy.example.com srv.example.com 2048 email@example.com /tmp/ca.crt /tmp/proxy.crt /tmp/proxy.key -o /tmp/config.tar.gz'
```

+

- . 如果您的设置使用中间 CA，请同时复制该证书，并在命令中通过 `'-i'` 选项（可根据需要多次提供）包含该证书：

+

```
mgrctl cp intermediateCA.pem server:/tmp/intermediateCA.pem mgrctl exec -ti 'spacecmd proxy_container_config --p 8022 -i /tmp/intermediateCA.pem pxy.example.com srv.example.com 2048 email@example.com /tmp/ca.crt /tmp/proxy.crt /tmp/proxy.key -o /tmp/config.tar.gz'
```

- 从服务器容器复制生成的配置：

+

```
mgrctl cp server:/tmp/config.tar.gz
```

```
// end::generate-proxy-config-section[]

[[proxy-setup-containers-transfer-config]]
== 传输 {productname} 代理配置
```

执行 [command]``spacecmd`` 命令和使用 {webui}
 这两种方式均会创建配置归档。该归档需在容器主机上可用，请将生成的此归档传输至容器主机。

```
[[proxy-setup-containers-transfer-start]]
== 启动 {productname} 代理容器
```

可以执行 [literal]`mgrpxy` 命令来启动容器。

```
[[proc-setup-containers-setup-start]]
.过程：启动 {productname} 代理容器
```

- 运行以下命令：

+

```
mgrpxy start uyuni-proxy-pod
```

+

- 调用以下命令检查所有容器是否都已按预期启动：

+

```
podman ps
```

应该会显示以下五个 {productname} 代理容器，并且它们应该是 [literal]``proxy-pod`` 容器 Pod 的一部分。

```
* proxy-salt-broker
* proxy-httpd
* proxy-tftpd
* proxy-squid
* proxy-ssh
```

```
:leveloffset: 3
:leveloffset: +3
```

```
[[installation-proxy-containers]]
= {productname} Proxy Deployment on {opensuse} {tumbleweed}
:revdate: 2025-10-08
:page-revdate: {revdate}
```

本指南概述了 {productname} {productnumber} 代理的部署过程。本指南假定您已成功部署 {productname} {productnumber} 服务器。要成功完成部署，请执行以下操作：

- . 核对清单：代理部署
- . 查看硬件要求。
- . Install {opensuse} {tumbleweed} on a bare-metal machine.
- . 将代理作为 {salt} 受控端进行引导。
- . 生成代理配置。
- . 将服务器中的代理配置传输到代理
- . 使用代理配置将 {salt} 受控端作为代理注册到 {productname}。

. 代理容器主机支持的操作系统

[NOTE]

====

The supported operating system for the container host is {opensuse} {tumbleweed}.

容器主机:: 容器主机是配备了容器引擎（例如 Podman）的服务器，可用于管理和部署容器。这些容器包含应用程序及其必备组件（例如库），但不包含完整的操作系统，因此体量很小。此设置可确保应用程序能够在不同环境中以一致的方式运行。容器主机为这些容器提供必要的资源，例如 CPU、内存和存储。

====

== 代理的硬件要求

下表列出了部署 {productname} 代理所要满足的硬件要求。

[cols="1,3,2", options="header"]

. 代理硬件要求

|====

| | | |
|------|--|--|
| 硬件 | | |
| 详细信息 | | |
| 建议 | | |

| | | |
|-----------------------|--|--|
| CPU | | |
| {x86_64}、{arm} | | |
| 至少 2 个专用的 64 位 CPU 核心 | | |

| | | |
|------|--|--|
| RAM | | |
| 至少 | | |
| 2 GB | | |

| | | |
|------|--|--|
| 建议 | | |
| 8 GB | | |

| | | |
|-------------------|--|--|
| 磁盘空间 | | |
| [path]``/`` (根目录) | | |
| 至少 40 GB | | |

| | | |
|---|--|--|
| [path]``/var/lib/containers/storage/volumes`` | | |
| 至少 100 GB，应根据您要使用的 ISO 发行套件映像、容器和引导储存库的数量来计算存储要求。 | | |

```
|==
```

== 容器主机一般要求

有关一般要求, 请参见xref:installation-and-upgrade:general-requirements.adoc[一般要求]。

An {opensuse} {tumbleweed} server should be installed from installation media. This procedure is described below.

[[installation-proxy-containers-requirements]]

== 容器主机要求

有关 CPU、RAM 和存储要求, 请参见xref:installation-and-upgrade:hardware-requirements.adoc[硬件要求]。

[IMPORTANT]

```
====
```

为了保证客户端能够解析 FQDN 域名, 容器化服务器和主机都必须连接到正常运行的 DNS 服务器。此外, 必须确保反向查找的配置正确。

```
====
```

== 安装用于容器的 Uyuni 工具

.Procedure: Installing Uyuni Tools on {opensuse} {tumbleweed}
[role=procedure]

. On your local host open a terminal window or start up a virtual machine running {opensuse} {tumbleweed}.

- . 登录系统。
- . Add the following repository to your {opensuse} {tumbleweed} server:

+

```
zypper ar https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable/images/repo/Uyuni-Proxy-POOL-$(arch)-Media1/ uyuni-proxy-stable
```

- . 刷新储存库列表并接受密钥:

+

```
zypper ref
```

- . 安装容器工具:

+

[source, shell]

```
zypper in mgrpxy mgrpxy-bash-completion uyuni-storage-setup-proxy
```

+

[NOTE]

====

或者，可以安装 [systemitem]``mgrpxy-zsh-completion`` 或 [systemitem]``mgrpxy-fish-completion``。

====

====

有关 Uyuni 容器实用程序的详细信息，请参见

[link:https://build.opensuse.org/repositories/systemsmanagement:Uyuni:Stable:ContainerUtils](https://build.opensuse.org/repositories/systemsmanagement:Uyuni:Stable:ContainerUtils) [Uyuni 容器实用程序]。

== 配置自定义永久性存储

此步骤是可选的。但是，如果您的基础架构需要自定义的永久性存储，请使用 [command]``mgr-storage-proxy`` 工具。

有关详细信息，请参见 [command]``mgr-storage-proxy --help``。此工具可以简化容器存储和 Squid 缓存卷的创建。

如下所示使用命令：

`mgr-storage-proxy <存储磁盘设备>`

例如：

`mgr-storage-proxy /dev/nvme1n1`

[NOTE]

====

此命令将在 [path]``/var/lib/containers/storage/volumes`` 中创建永久性存储卷。

有关详细信息，请参见

* [xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc\[\]](#)
 * [xref:administration:troubleshooting/tshoot-container-full-disk.adoc\[\]](#)

====

== 将代理主机作为受控端进行引导

. 任务：引导代理主机

- . 选择 `menu:系统[引导]`。
- . 填写代理主机的相关字段。
- . 从下拉列表中选择上一步骤中创建的激活密钥。
- . 单击 `btn:[+ 引导]`。
- . 等待引导过程成功完成。检查 `menu:Salt[]` 菜单，确认 {salt} 受控端密钥已列出并已接受。
- . 重引导代理主机。
- . 从 `menu:系统[]` 列表中选择主机，并在所有事件完成后再次触发重引导以完成初始配置。

. 任务：更新代理主机

- . 从 `menu:系统[]` 列表中选择主机，并应用所有补丁以将其更新。
- . 重引导代理主机。

```
// FIXME 2024-12-10, ke: use the snippet also here (see MLM equiv)
[[proxy-setup-containers-generate-config]]
== 生成代理配置
```

{productname} 代理的配置归档由 {productname} 服务器生成。每个附加代理都需要自身的配置归档。

[IMPORTANT]

=====
在生成此代理配置之前，必须将 {productname} 代理的容器主机作为 Salt 受控端注册到 {productname} 服务器。
=====

您将执行以下任务：

- . 过程：
- . 生成代理配置文件。
- . 将配置传输到代理。
- . 使用 [literal]``mgrpxy`` 命令启动代理。

[[proc-proxy-containers-setup-webui]]

. 任务：使用 Web UI 生成代理容器配置

- . 在 {webui} 中，导航到menu:系统[代理配置]，然后填写所需数据：
- . 在[guimenu]``代理 FQDN``字段中，键入代理的完全限定域名。
- . 在[guimenu]``父 FQDN``字段中，键入 {productname} 服务器或另一个 {productname} 代理的完全限定域名。
- . 在[guimenu]``代理 SSH 端口``字段中，键入 SSH 服务在 {productname} 代理上监听的 SSH 端口。建议保留默认值 8022。
- . 在[guimenu]``最大 Squid 缓存大小 [MB]``字段中，键入允许的最大 Squid 缓存大小。一般该值最多应为容器可用存储空间的 60 %。
- 在 [guimenu]``SSH 证书``选择列表中，选择应为 {productname} 代理生成新服务器证书还是使用现有证书。您可以考虑作为 {productname} 内置（自我签名）证书生成的证书。

+

然后根据所做的选择提供用于生成新证书的签名 CA 证书的路径，或者要用作代理证书的现有证书及其密钥的路径。

+

The CA certificates generated on the server are stored in the [path]``/var/lib/containers/storage/volumes/root/_data/ssl-build`` directory.

+

有关现有或自定义证书的详细信息以及企业和中间证书的概念，请参见 [xref:administration:ssl-certs-imported.adoc](#)。

- . 单击 btn:[生成] 以在 {productname} 服务器中注册新代理 FQDN，并生成包含容器主机细节的配置归档。
- . 片刻之后，系统会显示文件可供下载。请将此文件保存在本地。

[[proxy-deploy-containers-transfer-config]]

== 传输代理配置

{webui} 将生成配置归档。需要在代理容器主机上提供此归档。

. 任务：复制代理配置

- . Copy the files to the Proxy host:

```
scp config.tar.gz <代理 FQDN>/root
```

- . 使用以下命令安装代理:

+

`mgrpxy install podman config.tar.gz`

`[[proxy-deploy-containers-transfer-start]]`
== 启动 {productname} {productnumber} 代理

现在可以使用 [literal]`mgrpxy` 命令启动容器:

`[[proc-install-containers-setup-start]]`
.任务: 启动代理并检查状态

- . 调用以下命令启动代理:

+

`mgrpxy start`

- . 调用以下命令检查容器状态:

+

`mgrpxy status`

+

应该会显示以下五个 {productname} 代理容器，并且它们应该是 [literal]``proxy-pod`` 容器 Pod 的一部分:

* proxy-salt-broker
* proxy-httppd
* proxy-tftpd
* proxy-squid
* proxy-ssh

==== 为服务使用自定义容器映像

默认情况下, {productname}

代理套件设置为针对其每个服务使用相同的映像版本和仓库路径。但是，可以使用以 [literal]``-tag`` 和 [literal]``-image`` 结尾的 `install` 参数覆盖特定服务的默认值。

例如，可以按如下方式使用此命令：

`mgrpxy install podman --httpd-tag 0.1.0 --httpd-image registry.opensuse.org/uyuni/proxy-httppd /path/to/config.tar.gz`

该命令会在重启 `httpd` 服务之前调整其配置文件。其中
`[path]``registry.opensuse.org/uyuni/proxy-httppds``` 是要使用的映像, `[literal]``0.1.0``` 是版本标记。

要重置为默认值，请再次运行 `install` 命令但不要指定这些参数:

`mgrpxy install podman /path/to/config.tar.gz`

此命令首先将所有服务的配置重置为全局默认值，然后重新装载配置。

```
:leveloffset: 3
:leveloffset: +3

[[proxy-conversion-from-client-uyuni]]
= 从客户端转换为代理
```

== 概述

本章介绍如何通过 {webui} 将客户端系统转换为 {productname} 代理。

本文假定代理主机系统已完成引导，且已订阅基础操作系统通道。

有关客户端初始配置的详细信息，请参见 [xref:client-configuration:registration-overview.adoc\[\]](#)。

== 要求

开始转换前，请确保满足以下要求。

==== 客户端需满足的条件

- 已在 {productname} 中完成初始配置
- 可通过网络访问

== 准备工作

进行代理转换前，请完成以下准备工作，以免转换过程中出现中断。

==== SSL 证书

为确保代理与其他组件之间的通信安全，需要具备有效的 SSL 证书。

您需要准备：

- * 为 {productname} 服务器证书签名的证书颁发机构 (CA) 的公共证书。
- * 代理专用证书。
- * 代理证书对应的私用密钥。

[NOTE]

=====

如果您的 CA 使用中间证书链，还必须包含所有中间证书。

=====

如果不使用第三方证书，可在 {productname} 容器内通过 `rhn-ssl-tool` 生成证书。

. 生成代理证书

- . 在 {productname} 服务器主机上运行以下命令：
- + [source, shell]

```
mgrctl exec -ti -- rhn-ssl-tool --gen-server \ --set-hostname=<PROXY-FQDN> \ --dir=/root/ssl-build"
```

+
有关其他参数的详细信息，请参见 [xref:administration:ssl-certs-selfsigned.adoc\[\]](#)。
+

- . 将证书传输至 {productname} 服务器主机

+
[source, shell]

```
mgrctl cp server:/root/ssl-build/<PROXY-FQDN>/server.crt /root/proxycert.pem mgrctl cp server:/root/ssl-build/<PROXY-FQDN>/server.key /root/proxykey.pem mgrctl cp server:/root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT /root/rootca.pem
```

+
[NOTE]

===== 如需确认证书和密钥文件的具体生成目录，可以执行以下命令列出相关目录：

```
mgrctl exec -ti -ls -ltd /root/ssl-build/*
```

=====

- . 从 {productname} 服务器主机传输证书
- +
[source, shell]

```
scp <UYUNI-FQDN>:/root/proxycert.pem ./ scp <UYUNI-FQDN>:/root/proxykey.pem ./ scp <UYUNI-FQDN>:/root/rootca.pem ./
```

==== 软件包准备

==== 安装 ‘mgrpxy’ 工具

必须从与您的系统匹配的储存库安装 ‘mgrpxy’ 工具。请从以下地址选择适当的储存库：

<https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable:/ContainerUtils/>

- .Example {opensuse} {tumbleweed} installation:
[source, shell]

```
zypper ar https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable:/ContainerUtils/openSUSE\_Tumbleweed/ uyuni-containerutils zypper ref zypper in mgrpxy
```

==== 安装容器映像

建议以 RPM 软件包形式部署容器映像。请确保客户端上已安装以下软件包：

[source, shell]

```
zypper ar https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable/containerfile/ uyuni-proxy-images zypper ref zypper in uyuni-proxy-httpd-image \ uyuni-proxy-salt-broker-image \ uyuni-proxy-squid-image \ uyuni-proxy-ssh-image \ uyuni-proxy-tftpd-image
```

有关隔离部署的详细信息，请参见 [xref:installation-and-upgrade:container-deployment/mlm/proxy-air-gapped-deployment-mlm.adoc](#)。

== 设置代理客户端

- . 导航到客户端的[literal] ``概览`` 页面。
 - . 单击 `btn:[转换为代理]` 按钮。
- +
确认您已重定向至代理配置表单。
+
以后可通过[literal] ``细节`` > [literal] ``代理`` > [literal] ``配置`` 选项卡访问该页面。

- . 在 {webui} 中，导航到menu:代理[配置]，并填写所需数据：

+
. 过程：配置代理
.. 在[guimenu] ``父 FQDN `` 字段中，键入父服务器或代理的完全限定域名。
.. 在[guimenu] ``代理 SSH 端口 `` 字段中，键入 {productname} 代理上 SSH 服务的监听端口。建议保留默认值 8022。
.. 在[guimenu] ``最大 Squid 缓存大小 `` 字段中，键入允许的最大 Squid 缓存大小（以 GB 为单位）。
.. 在[guimenu] ``代理管理员电子邮件 `` 字段中，键入管理员的电子邮件地址。
.. 在[literal] ``证书 `` 部分，提供在进行准备时获取的 {productname} 代理证书。

.. 在[literal] ``源 `` 部分，选择以下两个选项之一：[literal] ``RPM `` 或 [literal] ``仓库 `` 。
+
* 对于隔离或受限环境，建议使用 [literal] ``RPM `` 选项。
The [literal] ``Registry `` option can be used if connectivity to the container image registry is available. + If selected, you will be prompted to choose between two sub-options: [literal] ``Simple `` or [literal] ``Advanced `` .
+
** 如果选择[literal] ``简单 ``，需要在[literal] ``仓库 URL `` 和 [literal] ``容器标记 `` 字段中输入对应值。
*** 在[literal] ``仓库 URL `` 中填写 [literal] ``registry.opensuse.org/uyuni `` 。
*** 从下拉列表中选择标记。
** 如果选择[literal] ``高级 ``，表单会显示一个额外的部分：
*** 对于每个独立容器的 URL 字段，均需填写仓库地址
[literal] ``registry.opensuse.org/uyuni ``，并添加对应的后缀（例如 ` _proxy-httpd_ ` 或 ` _salt-broker_ ` ）。
*** 从下拉列表中选择标记。

- . 填写完所有字段后，单击 `btn:[应用]` 按钮保存配置并调度代理安装任务。

== 校验代理激活状态

查看客户端的事件历史记录，确认任务执行成功。

(可选) 访问代理的 HTTP 端点，验证其是否显示欢迎页面。

```
:leveloffset: 3
:leveloffset: +3

[[installation-proxy-containers-k3s-uyuni]]
= 在 K3s 上部署 {productname} 代理
:revdate: 2025-02-06
:page-revdate: {revdate}
```

```
[[installation-proxy-containers-k3s-k3s]]
== 安装 K3s
```

在容器主机计算机上，安装 [literal] ``K3s ``（请将 [literal] ``<K3S_HOST_FQDN> `` 替换为 k3s 主机的 FQDN）：

```
curl -sfL https://get.k3s.io | INSTALL_K3S_EXEC="--tls-san=<K3S_HOST_FQDN>" sh -
```

[[installation-proxy-containers-k3s-helm]]
== 安装工具

在安装时需要提供 [literal]``mgrpxy`` 和 [literal]``helm`` 软件包。

使用安装程序脚本安装 Helm:

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3  
chmod 700 get_helm.sh ./get_helm.sh
```

有关详细信息，请参见 [https://helm.sh/docs/intro/install/#from-script\[\]](https://helm.sh/docs/intro/install/#from-script[])。

[literal]``mgrpxy`` 软件包可从容器实用程序储存库中获取。请在 <https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable:/ContainerUtilities/> 中选择与发行套件匹配的软件包。

. 过程

. 要在 Leap Micro 上安装软件包，请运行以下命令：

+

```
transactional-update pkg install mgrpxy
```

. 重引导。

[[installation-proxy-containers-k3s-deploy]]
== 部署 {productname} 代理 helm 图表

要配置 {productname} 代理 Pod

使用的卷存储空间，请为以下声明定义永久性卷。如果您未自定义存储配置，K3s 将自动为您创建存储卷。

永久性卷声明已命名为：

- * [literal]``squid-cache-pv-claim``
- * [literal]``/package-cache-pv-claim``
- * [literal]``/tftp-boot-pv-claim``

按照 [xref:installation-and-upgrade:container-deployment/uyuni/proxy-container-setup-uyuni.adoc\[\]](#) 中所述创建 {productname} 代理的配置。复制 [literal]``tar.gz`` 配置文件，然后安装：

```
mgrpxy install kubernetes /path/to/config.tar.gz
```

有关详细信息，请参见：

- * link:[https://kubernetes.io/docs/concepts/storage/persistent-volumes/\[\]](https://kubernetes.io/docs/concepts/storage/persistent-volumes/) (Kubernetes)
- * link:[https://rancher.com/docs/k3s/latest/en/storage/\[\]](https://rancher.com/docs/k3s/latest/en/storage/) (K3s) 文档

:leveloffset: 3

== 升级和迁移

==== 服务器

```
:leveloffset: +3

= Migrating the {productname} Server to {opensuse} Tumbleweed
:revdate: 2025-10-27
:page-revdate: {revdate}
:description: This page describes how to migrate a {productname} Server running on openSUSE Leap Micro 5.5 to a fresh host running openSUSE Tumbleweed as the base OS.

This page describes a simple, backup-and-restore migration of a {productname} Server running on openSUSE Leap Micro 5.5 to a fresh host running openSUSE Tumbleweed as the base OS.
```

== Overview of the Migration Process

You will:

- * Create a full server backup with [command]``mgradm backup`` on the openSUSE Leap Micro 5.5 host.
- * Reinstall the host with openSUSE Tumbleweed (server profile).
- * Install {productname} tools and prerequisites on Tumbleweed.
- * Restore the backup with [command]``mgradm backup restore``.
- * Start services and verify the server.

== 要求和注意事项

- * Source server: openSUSE Leap Micro 5.5 running {productname} (for example: {productnumber}).
- * Target server: openSUSE Tumbleweed with the same hostname/FQDN and IP (recommended) to avoid client-side changes.
- * SSH/scp access between machines for transferring the backup tarball.
- * Sufficient free disk space on both source and target for the backup and restore.

[IMPORTANT]

====

Restore to the same {productname} version you backed up, or a version explicitly documented as compatible for restore. If you use development or preview repositories (for example, Uyuni Master), expect changes and re-validate.

====

== Migration Procedure

==== Step 1: Create a Backup on the openSUSE Leap Micro 5.5 Server

.Procedure: Create a Backup

[role="procedure"]

. As root on the old server, create a backup directory and run the backup:

+

[source, shell]

mgradm backup /tmp/uyuni-backup

. Package the backup for transfer:

+

[source, shell]

tar -C /tmp -cvf /tmp/uyuni-backup.tar uyuni-backup

- . Copy the backup to a safe location you can reach from the new host:

+

[source, shell]

```
scp /tmp/uyuni-backup.tar <USER>@<HOST>:/path/to/store/
```

[TIP]

====

You can store the backup to external storage or an object store as long as you can fetch it on the new host.

====

====

== Step 2: Reinstall the Host with openSUSE Tumbleweed

.Procedure: Reinstalling the Host

[role="procedure"]

====

- . Reprovision the VM or bare-metal host with openSUSE Tumbleweed.

- . Choose a basic “server profile” installation.

- . Set the same hostname/FQDN and IP address as the original server if you want clients to reconnect seamlessly.

====

== Step 3: Install {productname} Tools and Prerequisites on Tumbleweed

.Procedure: Installing Tools and Prerequisites

[role="procedure"]

====

- . Add the Uyuni Stable repository and install tools:

+

[source, shell]

```
zypper ar https://download.opensuse.org/repositories/systemsmanagement/Uyuni/Stable/images/repo/Uyuni-Server-POOL-x86_64-Media1 uyuni-server-stable zypper ref zypper in mgradm mgrctl mgradm-bash-completion mgrctl-bash-completion uyuni-storage-setup-server
```

- . Install Podman if it was not automatically pulled in:

+

[source, shell]

```
zypper in podman
```

[NOTE]

====

The package [package]``uyuni-storage-setup-server`` provides the [command]``mgr-storage-server`` tool for preparing persistent volumes. Installing [package]``podman`` explicitly may be necessary on some installations.

====

====

== Step 4: Optional - Prepare Persistent Storage

.Procedure: Preparing Persistent Storage
[role="procedure"]

====

It is recommended to configure persistent storage with [command]``mgr-storage-server`` to avoid container full-disk issues.

[source, shell]

mgr-storage-server <storage-disk-device> [<database-disk-device>]

[NOTE]

====

Devices must be raw (no existing filesystem). The tool creates volumes at [path]``/var/lib/containers/storage/volumes``.

For details, see:

- * [xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc\[\]](#)
- * [xref:administration:troubleshooting/tshoot-container-full-disk.adoc\[\]](#)

====

====

== Step 5 Fetch and Restore the Backup on Tumbleweed

.Procedure: Fetching and Restoring the Backup
[role="procedure"]

====

- . Copy the backup to the new server and unpack it:

+

[source, shell]

scp <USER>@<HOST>:/path/to/store/uyuni-backup.tar /tmp/ tar -C /tmp -xvf /tmp/uyuni-backup.tar

- . Restore using [command]``mgradm`` (point to the extracted backup directory):

+

[source, shell]

mgradm backup restore /tmp/uyuni-backup

====

== Step 6: Start Services and Verify

.Procedure: Starting Services and Verifying

- . Start the server services:

+

[source, shell]

mgradm start

```
. Verify:
** Check that all containers are up: [command]``mgrctl ps`` or [command]``podman ps``.
** Access the Web UI (HTTPS) and log in.
** Review logs for errors: [command]``mgrctl logs server`` and other components as needed.

** ---

== Notes and Troubleshooting

* If Podman wasn't installed automatically, install it with [command]``zypper in podman`` and rerun the restore/start steps.
* Ensure the target host has the same time, hostname, and IP configuration expected by your setup (especially if clients exist).
* For large environments, ensure adequate disk throughput and space. The backup and restore can take a long time.
```

[IMPORTANT]

====

If the restore fails or the new system cannot start, you can still boot the original openSUSE Leap Micro 5.5 system and continue service. Keep the original VM/snapshots until you fully validate the new Tumbleweed-based server.

====

:leveloffset: 3
:leveloffset: +3

= 将旧版 {productname} 服务器迁移到容器
:revdate: 2025-07-30
:page-revdate: {revdate}

要将旧版 {productname} 服务器迁移到容器，需要准备一台新计算机。

在该迁移场景中，旧版 {productname} 服务器（RPM 安装方式）有时也被称为_旧服务器_。

== 要求和注意事项

==== 主机名

目前，该迁移过程既不支持就地迁移，也不提供主机名重命名功能。

因此，新服务器的完全限定域名（FQDN）将与旧版服务器保持一致。

[IMPORTANT]

====

迁移之后，需要更新 DHCP 和 DNS 记录以指向新的服务器。

有关详细信息，请参见 <>migration-finalization>>。

====

==== SSL certificates

:leveloffset: +1

后续流程需要 SSL 证书。如果不使用自我签名的 CA 和证书，请在开始前准备好以下内容：

- * 证书颁发机构 (CA) SSL 公共证书。如果您使用 CA 链，则所有中间 CA 也必须可用。
- * SSL 数据库私用密钥。
- * SSL 数据库证书。

所有文件必须采用 PEM 格式。

SSL 服务器证书的主机名必须与部署证书的计算机的完全限定主机名匹配。您可以在证书的 [literal]``X509v3 Subject Alternative Name`` 部分中设置主机名，也可以根据环境的需要列出多个主机名。支持的密钥类型为 [literal]``RSA`` 和 [literal]``EC`` (椭圆曲线)。

[IMPORTANT]

====

数据库 SSL 证书的 [literal]``Subject Alternative Name`` 必须是 [literal]``reportdb``、[literal]``db`` 及用于访问报告数据库的 FQDN。

====

:leveloffset: 3

During a migration, the server SSL certificate and CA chain are copied from the source server, meaning that only the database certificates are required

== GPG 密钥

- * 自信任 GPG 密钥不会被迁移。
- * 仅在 RPM 数据库中可信的 GPG 密钥不会迁移。因此，使用 [command]``spacewalk-repo-sync`` 同步通道可能会失败。
- * 在完成服务器的实际迁移后，管理员必须手动将这些密钥从旧版 {productname} 系统迁移到容器主机。

.过程：将 GPG 密钥手动迁移到新服务器

- . 将旧版 Uyuni 服务器中的密钥复制到新服务器的容器主机。
- . 稍后，使用命令 [command]``mgradm gpg add <PATH_TO_KEY_FILE>`` 将每个密钥添加到迁移的服务器。

==== 旧版服务器上的初始准备工作

迁移可能需要很长时间，具体时长取决于需要复制的数据量。为了减少停机时间，可以在初始复制、重新复制或最终复制和切换过程中多次执行迁移，同时旧版服务器上的所有服务都可以保持正常运行。

只有在最终迁移期间才需要停止旧版服务器上的进程。

对于最终复制以外的所有复制，请添加参数 [command]``--prepare``，以防止自动停止旧版服务器上的服务。例如：

[source, shell]

```
mgradm migrate podman <oldserver.fqdn> --prepare
```

.过程：旧版服务器上的初始准备工作

- . 停止 {productname} 服务：
- +

```
spacewalk-service stop
```

- . 停止 PostgreSQL 服务:

+

```
systemctl stop postgresql
```

==== SSH 连接准备

- . 过程: 准备 SSH 连接

. 确保新 {productnumber} 服务器上的 [systemitem]``root`` 已存在 SSH 密钥。如果密钥不存在, 请使用以下命令创建一个:

+

```
ssh-keygen -t rsa
```

- . 新服务器上的 SSH

配置和代理应准备就绪, 这样在连接旧版服务器时, 就不会出现输入口令的提示。

+

```
eval $(ssh-agent); ssh-add
```

+
[NOTE]

=====
迁移脚本依赖新服务器上运行的 SSH

代理来建立不提示输入口令的连接。如果该代理尚未激活, 请运行 [command]``eval \$(ssh-agent)`` 将其启动。然后, 使用 [command]``ssh-add`` (后跟私用密钥的路径) 将 SSH 密钥添加到正在运行的代理。在此过程中, 系统将提示您输入私用密钥的口令。

=====

- . 使用 [command]``ssh-copy-id`` 将公共 SSH 密钥复制到旧版 {productname} 服务器 ([literal]``<oldserver.fqdn>``)。请将 [literal]``<oldserver.fqdn>`` 替换为旧版服务器的 FQDN:

+

```
ssh-copy-id <oldserver.fqdn>
```

+
SSH 密钥会被复制到旧版服务器的 [path]``~/.ssh/authorized_keys`` 文件中。有关详细信息, 请参见 [literal]``ssh-copy-id`` 手册页。

. 在新服务器上与旧版 {productname} 服务器建立 SSH
连接, 检查是否不需要口令。此外, 主机指纹不得有任何问题。如果遇到问题, 请从 [path]``~/.ssh/known_hosts`` 文件中去除旧指纹。然后重试。指纹将存储在本地 [path]``~/.ssh/known_hosts`` 文件中。

==== 执行迁移

在规划从旧版 {productname} 到容器化 {productname} 的迁移时, 请确保您的目标实例达到或超过旧版设置的规格。这包括但不限于内存 (RAM)、CPU 核心、存储空间和网络带宽。

```
:leveloffset: +2
```

[NOTE]

=====
经过安全加固的 {productname} 服务器主机可能会限制从 [path]``/tmp`` 文件夹执行文件。在这种情况下，可通过以下临时解决方案规避：在运行 [command]``mgradm`` 前，将 [literal]``TMPDIR`` 环境变量导出至其他现有路径。

例如：

[source, shell]

```
export TMPDIR=/path/to/other/tmp
```

在 {productname} 的后续更新中，相关工具将进行优化，届时无需再使用此临时解决方案。

:leveloffset: 3

// Unify the following with matching MLM section and => snippet

===== 配置自定义永久性存储

配置永久性存储空间并非强制性要求，但这是唯一可避免在容器全盘空间用尽的情况下出现严重问题的方法。强烈建议您使用 [command]``mgr-storage-server`` 工具来配置自定义永久性存储空间。

有关详细信息，请参见 [command]``mgr-storage-server --help``。此工具可以简化容器存储和数据库卷的创建。

如下所示使用命令：

[source, shell]

```
mgr-storage-server <storage-disk-device> [<database-disk-device>]
```

[NOTE]

=====
设备上不得存在任何文件系统。如果存储设备上存在文件系统，命令会中止。

例如：

[source, shell]

```
mgr-storage-server /dev/nvme1n1 /dev/nvme2n1
```

[NOTE]

=====
此命令将在 [path]``/var/lib/containers/storage/volumes`` 中创建永久性存储卷。

有关详细信息，请参见

- * [xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc\[\]](#)
- * [xref:administration:troubleshooting/tshoot-container-full-disk.adoc\[\]](#)

===== 执行迁移

. 执行以下命令来安装新的 {productname} 服务器。请将 [literal]``<oldserver fqdn>``

替换为旧版服务器的 FQDN:

+

`mgradm migrate podman <oldserver.fqdn>`

- 迁移可信 SSL CA 证书。

===== 证书的迁移

作为 RPM 的一部分安装并存储在旧版 `{productname}` 上 `[path]``/usr/share/pki/trust/anchors``` 目录中的可信 SSL CA 证书将不会迁移。由于 `{suse}` 不会在容器中安装 RPM 软件包，因此迁移完成后，管理员必须手动从以前安装的旧版系统中迁移这些证书文件：

- . 过程：迁移证书
- . 将旧版服务器中的该文件复制到新服务器。
例如，复制为 `[path]``/local/ca.file```。
- . 使用以下命令将文件复制到容器中：

+

`mgrctl cp /local/ca.file server:/etc/pki/trust/anchors/`

[[migration-finalization]]
===== 完成迁移

[IMPORTANT]

=====

成功运行 `[command]``mgradm migrate``` 命令后，所有客户端上的 `{salt}` 设置仍会指向旧版服务器。

要将其重定向到新的 `{productnumber}` 服务器，需要在基础架构级别（DHCP 和 DNS）重命名新服务器，以使用与旧版服务器相同的 FQDN 和 IP 地址。

=====

[IMPORTANT]

=====

如果迁移过程中出现问题，可重启旧系统。请以 `root` 身份执行以下命令，重启 `{postgresql}` 和 `spacewalk` 服务：

[source, shell]

`service postgresql start spacewalk-service start`

=====

// FIXME: check the following! Partially probably already covered above.!
== Kubernetes 准备工作

在使用 `[command]``mgradm migrate``` 命令执行迁移之前，必须预先定义**永久性卷**，尤其要注意的是，迁移作业会从头开始初始化容器。

有关详细信息，请参见安装章节中关于准备这些卷的内容，具体可查看 `xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc[]`。

== 迁移

执行以下命令安装新的 {productname} 服务器（请将 **<oldserver.fqdn>** 替换为旧版服务器的相应 FQDN）：

```
mgradm migrate podman <oldserver.fqdn>
```

或

```
mgradm migrate kubernetes <oldserver.fqdn>
```

[IMPORTANT]

=====

成功运行 [command]``mgradm migration`` 命令后，所有客户端上的 {salt} 设置仍会指向旧版服务器。要将其重定向到新服务器，需要在基础架构级别（DHCP 和 DNS）重命名新服务器，以使用与旧版服务器相同的 FQDN 和 IP 地址。

=====

:leveloffset: 3
:leveloffset: +3

= {productname} 服务器升级
:revdate: 2025-07-29
:page-revdate: {revdate}

运行升级命令前，需要更新主机操作系统。更新主机操作系统时，{productname} 相关工具（如 [literal]``mgradm``）也会同步更新。

.过程：升级服务器
[role=procedure]

-- 使用 [command]``zypper`` 刷新软件储存库：

+

[source, shell]

```
zypper ref
```

. 使用 [command]``transactional-update`` 应用可用的更新：

+

[source, shell]

```
transactional-update
```

. 如果已应用更新，请[literal]``重引导``。

+

. 可使用以下命令更新 {productname} 服务器容器：

+

[source, shell]

mgradm upgrade podman

+

此命令可使容器保持最新状态并重启服务器。

+

- . 清理未使用的容器映像以释放磁盘空间：

+

[source, shell]

podman image prune -a

====

. 使用第三方 SSL 证书升级

[IMPORTANT]

====

如果您使用第三方证书，数据库容器需配备包含以下主体备用名称 (SAN) 的 SSL 证书：

- * [literal]``db``
- * [literal]``reportdb``
- * 对外提供服务的完全限定域名

同一证书可同时用于主容器与数据库容器，但需包含上述 SAN。

In order to pass the new certificate to the upgrade command, use the [command]``--ssl-db-ca-root``，[command]``--ssl-db-cert`` and [command]``--ssl-db-key`` parameters.

====

. 升级到特定版本

[NOTE]

====

如果未指定标记参数，则默认会升级到最新版本。要升级到特定版本，请为标记参数提供所需的映像标记。

====

要查看 upgrade 命令及其参数的详细信息，请使用以下命令：

[source, shell]

mgradm upgrade podman -h

对于隔离的安装，请先升级容器 RPM 软件包，然后运行 [literal]``mgradm`` 命令。

:leveloffset: +1

= 数据库备份卷

:revdate: 2025-07-29

:page-revdate: {revdate}

通过 `[command]``mgradm migration``` 或 `[command]``mgradm upgrade``` 执行服务器迁移或升级时，会创建一个包含数据库备份的卷。

当 `{postgresql}`

数据库有更新的版本时，必须在执行升级流程前将旧数据库存储到单独的位置。为此，`[command]``mgradm``` 会动态创建卷 `[path]``var-pgsql-backup```。当迁移或升级完成且用户已验证新系统可按预期正常运行后，可安全去除该卷。

`:leveloffset: 3`

`:leveloffset: 3`

`==== 代理`

`:leveloffset: +3`

`= Migrating the {productname} Proxy to {opensuse} Tumbleweed`

`:revdate: 2025-10-27`

`:page-revdate: {revdate}`

`:description: This page describes how to migrate a {productname} Proxy host from openSUSE Leap Micro 5.5 to a fresh openSUSE Tumbleweed installation using the proxy administration tool [command]``mgrpxy``.`

This page describes how to migrate a {productname} Proxy host from openSUSE Leap Micro 5.5 to a fresh openSUSE Tumbleweed installation using the proxy administration tool `[command]``mgrpxy```.

`[IMPORTANT]`

`=====`

This guide was tested on Tumbleweed only. There is no known reason it wouldn't work on other supported bases, but always validate in a test environment before production.

`=====`

== Overview of the Proxy Migration Process

You will:

- * Save proxy configuration from the old system (including Apache/Squid tuning).
- * Reinstall the host with openSUSE Tumbleweed.
- * Re-register the host using the system reactivation key.
- * Install `[command]``mgrpxy``` (and Podman if needed).
- * Restore configuration and run `[command]``mgrpxy install podman``` with optional tuning files.

== 要求和注意事项

- * Keep the same hostname/FQDN and IP when possible so the server and clients interact with the proxy as before.
- * Ensure you have the “system reactivation key” for the existing proxy system (UI: Systems > select the proxy > Details > Reactivation).
- * Ensure SSH/scp access to move configuration archives off and onto the machine.

== Migration Procedure

==== Step 1: Save Proxy Configuration and Tuning Files

.Procedure: Save Proxy Configuration and Tuning Files
`[role="procedure"]`

`=====`

. Copy the Uyuni proxy configuration directory to a safe location:

`+`

[source, shell]

```
scp -r /etc/uyuni <USER>@<HOST>:/some/where/safe/
```

- . Identify Apache and Squid tuning files currently in use by the legacy proxy services:

+

[source, shell]

```
systemctl cat uyuni-proxy-httpd.service | grep EXTRA_CONF= | sed 's/.=-v\([^\:]+\)\.:./1/' systemctl cat uyuni-proxy-squid.service | grep EXTRA_CONF= | sed 's/.=-v\([^\:]+\)\.:./1/'
```

- . Copy those tuning files to the same safe location as well.

[TIP]

====

Typical default paths after you copy them back will be:

- * Apache tuning: [path]``/etc/uyuni/proxy/apache.conf``
- * Squid tuning: [path]``/etc/uyuni/proxy/squid.conf``

====

==== Step 2: Reinstall the Host with openSUSE Tumbleweed

.Procedure: Reinstalling the Host with openSUSE Tumbleweed
[role="procedure"]

====

- . Reinstall the machine with openSUSE Tumbleweed (server profile recommended).
- . Set the same hostname/FQDN and IP as before when possible.

====

==== Step 3: Re-register the Host with the Reactivation Key

.Procedure: Re-registering the Host with the Reactivation Key
[role="procedure"]

====

- . From the {productname} Web UI, obtain the system reactivation key for the existing proxy system record (Systems > Details > Reactivation).

- . Bootstrap/re-register the Tumbleweed host using that reactivation key so it claims the existing system entry.

[NOTE]

====

Use your standard bootstrapping process for Tumbleweed hosts in your environment (for example, the bootstrap script or your configuration management), ensuring the reactivation key is applied.

====

====

```
==== Step 4: Install {productname} Proxy Tools and Podman
.Procedure: Installing Proxy Tools and Podman
[role="procedure"]
-----
. Add the Uyuni Stable repository and install tools:
+
[source, shell]
```

```
zypper ar https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable/
images/repo/Uyuni-Proxy-POOL-x86_64-Media1 uyuni-proxy-stable zypper ref zypper in mgrpxy
mgrctl mgrpxy-bash-completion mgrctl-bash-completion
```

```
. Ensure Podman is installed (required to run containers):
+
[source, shell]
```

```
zypper in podman
```

```
-----
==== Step 5: Restore Configuration and Install the Proxy
.Procedure: Restoring Configuration and Install the Proxy
[role="procedure"]
-----
. Copy back the saved configuration directory to the new host:
+
[source, shell]
```

```
scp -r <USER>@<HOST>:/some/where/safe/uyuni /etc/
```

```
. If you saved Apache/Squid tuning files, place them at the expected default paths or note
their locations for parameters in the next command:
+
[source, shell]
```

3.1.2. Default paths expected by mgrpxy parameters (adjust/move your files accordingly)

3.1.3. Apache tuning: /etc/uyuni/proxy/apache.conf

3.1.4. Squid tuning: /etc/uyuni/proxy/squid.conf

. Run the proxy installation with Podman. If you do not use tuning files, omit the corresponding parameters:

+

[source, shell]

3.1.5. With tuning files

```
mgrpxy install podman \ --tuning-httpd /etc/uyuni/proxy/apache.conf \ --tuning-squid
/etc/uyuni/proxy/squid.conf
```

3.1.6. If you have no tuning files, remove the tuning parameters:

3.1.7. mgrpxy install podman

[NOTE]

====

In an upcoming release, if tuning files are placed at the default paths noted above, the explicit parameters will not be required.

====

====

== Step 6: Verify the Proxy

.Procedure: Verifying the Proxy
[role="procedure"]

====

. Check containers are running:

+

[source, shell]

mgrctl ps # or podman ps

. Confirm the proxy appears healthy in the {productname} Web UI and that clients using this proxy operate normally.

====

== 查错

- * If Podman was missing, install it and rerun the [command]``mgrpxy install`` step.
- * Verify the host's time, hostname, and IP match expectations.
- * If the host did not reattach to the existing system record, confirm you used the correct reactivation key and repeat the bootstrap.

:leveloffset: 3
:leveloffset: +3

= 将旧版代理迁移到容器

```
:revdate: 2025-02-06
:page-revdate: {revdate}
```

容器化代理现在通过一组 `systemd` 服务进行管理。要管理容器化代理，请使用 `[command]``mgrpxy``` 工具。

本节内容将介绍如何使用 `[command]``mgrpxy``` 工具从旧版 `[systemitem]``systemd``` 代理迁移。

[IMPORTANT]

====

由于主机操作系统已从 `{leap}` 更改为 `{leapmicro}`，因此不支持从旧版 `{productname}` 就地迁移到 `{productnumber}`。

`{productname} {productnumber}` 及更高版本不再支持传统联系协议。从旧版 `{productname}` 迁移到 `{productnumber}` 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}`。

====

== 使用 `Systemd` 从旧版迁移到容器化代理

==== 生成代理配置

. 过程：生成代理配置

- . 登录到 `{productname}` 服务器 `{webui}`。
- . 在左侧导航栏中，选择 `menu:系统[代理配置]`。
- . 输入您的代理 FQDN。使用与原始代理主机相同的 FQDN。
- . 输入您的服务器 FQDN。
- . 输入代理端口号。建议使用默认端口 8022。--
- . 证书和私用密钥位于服务器容器主机上的 `/var/lib/containers/storage/volumes/root/_data/ssl-build/`` 中。
 - * `RHN-ORG-TRUSTED-SSL-CERT`
 - * `RHN-ORG-PRIVATE-SSL-KEY`
- . 使用以下命令将证书和密钥复制到您的计算机：

+

```
scp root@uyuni-server-example.com:/root/ssl-build/RHN-ORG-PRIVATE-SSL-KEY scp root@uyuni-
server-example.com:/root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT
```

- . 选择 `btn:[选择文件]` 并在本地计算机上通过浏览找到证书。
- . 选择 `btn:[选择文件]` 并在本地计算机上通过浏览找到私用密钥。
- . 输入 CA 口令。
- . 单击 `btn:[生成]`。

==== 将代理配置传输到新主机

. 过程：传输代理配置

- . 在服务器中，将生成的包含代理配置的 `tar.gz` 文件传输到新代理主机：

+

```
scp config.tar.gz <uyuni 代理 FQDN>/root/
```

- . 在执行下一步之前，请先禁用旧版代理：

+

```
spacewalk-proxy stop
```

- . 使用以下命令部署新代理:

+

```
systemctl start uyuni-proxy-pod
```

- . 使用以下命令启用新代理:

+

```
systemctl enable --now uyuni-proxy-pod
```

- . 运行 ``podman ps`` 来校验所有容器是否存在并正在运行:

+

```
proxy-salt-broker proxy-httdp proxy-tftpd proxy-squid proxy-ssh
```

== 将 {productname} 代理迁移到 {productname} {productnumber} 容器化代理

. 过程: 将 {productname} 容器化代理迁移到 {productname} {productnumber} 新容器化代理

. 引导新计算机, 然后开始安装 {leapmicro} {microversion}。

. 完成安装。

. 更新系统:

+

```
transactional-update --continue
```

- . 安装 [command]``mgrpxy``, 并根据需要安装 [command]``mgrpxy-bash-completion``:

+

```
transactional-update pkg install mgrpxy mgrpxy-bash-completion
```

+

. 重引导。

. 将 [literal]``tar.gz`` 代理配置复制到主机。

== 使用 {webui} 安装软件包

在受控端已引导并已注册到服务器后, 还可以通过 Web UI 安装 [package]``mgrpxy`` 和 [package]``mgrpxy-bash-completion`` 软件包。

. 过程: 使用 {webui} 安装软件包

. 安装后, 确保在menu:管理[安装向导 -> 产品]页面中添加并同步 {sle-micro} {microversion} 父通道和代理子通道。

. 在 {webui} 中, 转到menu:系统[激活密钥], 并创建一个与已同步 {sle-micro} {microversion} 通道关联的激活密钥。

. 使用menu:系统[引导]页面将系统作为受控端进行引导。

. 在初始配置新计算机并且其显示在系统列表中后, 选择系统并导航到menu:系统细节[安装软件包]页面。

. 安装软件包 [package]``mgrpxy`` 和 [package]``mgrpxy-bash-completion``。

. 重引导系统。

== 使用 [literal]``spacecmd`` 和自我签名证书生成代理配置

可以使用 spacecmd 生成代理配置。

. 过程：使用 [literal]``spacecmd`` 和自我签名证书生成代理配置

. 通过 SSH 连接到您的容器主机。

. 执行以下命令（替换其中的服务器和代理 FQDN）：

+

```
mgrctl exec -ti 'spacecmd proxy_container_config_generate_cert --dev-pxy.example.com dev-srv.example.com 2048 email@example.com -o /tmp/config.tar.gz'
```

. 将生成的配置复制到代理：

+

```
mgrctl cp server:/tmp/config.tar.gz
```

. 使用以下命令部署代理：

+

```
mgrpxy install podman config.tar.gz
```

== 使用 [literal]``spacecmd`` 和自定义证书生成代理配置

可以使用 [literal]``spacecmd`` 为自定义证书（而不是默认的自我签名证书）生成代理配置。

[NOTE]

====

2 GB 表示默认的代理 squid 缓存大小。需要根据您的环境调整此大小。

====

. 过程：使用 [literal]``spacecmd`` 和自定义证书生成代理配置

. 通过 SSH 连接到您的服务器容器主机。

. 执行以下命令（替换其中的服务器和代理 FQDN）：

+

```
for f in ca.crt proxy.crt proxy.key; do mgrctl cp $f server:/tmp/$f done mgrctl exec -ti 'spacecmd proxy_container_config --p 8022 pxy.example.com srv.example.com 2048 email@example.com /tmp/ca.crt /tmp/proxy.crt /tmp/proxy.key -o /tmp/config.tar.gz'
```

. 将生成的配置复制到代理：

+

```
mgrctl cp server:/tmp/config.tar.gz
```

. 使用以下命令部署代理：

+

```
mgrpxy install podman config.tar.gz
```

```
:leveloffset: 3  
:leveloffset: +3
```

```
= {productname} 代理升级
:revdate: 2025-05-05
:page-revdate: {revdate}
```

运行升级命令前，需要更新主机操作系统。更新主机操作系统时，{productname} 相关工具（如 [literal]``mgrpxy``）也会同步更新。

- . 过程：升级代理
[role=procedure]

- . 使用 [command]``zypper`` 刷新软件储存库：

+

[source, shell]

`zypper ref`

- . 使用 [command]``transactional-update`` 应用可用的更新：

+

[source, shell]

`transactional-update`

- . 如果已应用更新，请[literal]``重引导``。

+

- . 可使用以下命令来更新 [literal]``podman`` 上运行的 {productname} 代理容器：

+

[source, shell]

`mgrpxy upgrade podman`

+

或者，可使用以下命令更新 Kubernetes 群集上运行的容器：

+

[source, shell]

`mgrpxy upgrade kubernetes`

+

- . 在 Podman 上，清理未使用的容器映像以释放磁盘空间：

+

[source, shell]

```
podman image prune -a
```

=====
在 Kubernetes 中，映像清理工作要么自动完成，要么由所用的 Kubernetes 发行版决定具体执行方式。

[NOTE]

=====如果升级到特定版本时未指定标记参数，则默认会升级到最新版本。要升级到特定版本，请为标记参数提供所需的映像标记。
=====

[IMPORTANT]

=====

在正常情况下，我们强烈建议为所有代理容器使用相同的标记，以确保配置一致性。
=====

对于隔离的安装，请先升级容器 RPM 软件包，然后运行 [command]``mgrpxy upgrade podman`` 命令。

:leveloffset: 3

==== 客户端

:leveloffset: +3

[[client-upgrade]]

= 升级客户端

:revdate: 2024-11-19

:page-revdate: {revdate}

客户端采用底层操作系统的版本控制系统。对于运行 {suse} 操作系统的客户端，可在 {productname} {webui} 中进行升级。

有关升级客户端的详细信息，请参见 [xref:client-configuration:client-upgrades.adoc\[\]](#)。

:leveloffset: 3

== 基本的服务器和代理管理

:leveloffset: +2

= Custom YAML Configuration and Deployment with mgradm

:revdate: 2024-11-19

:page-revdate: {revdate}

您可以选择创建自定义的 [filename]``mgradm.yaml`` 文件，供 [command]``mgradm`` 工具在部署期间使用。

[IMPORTANT]

=====

如果未提供基本变量，[command]``mgradm`` 将提示您使用命令行参数或 [filename]``mgradm.yaml`` 配置文件来提供这些变量。

为了安全起见，**应避免使用命令行参数指定口令**。请改用具有适当权限的配置文件。
=====

.过程：使用自定义配置文件通过 Podman 部署 {productname} 容器

. 准备一个名为 [filename]``mgradm.yaml`` 的配置文件，以以下示例所示：

```
+  
[source, yaml]  
....  
# 数据库口令。默认会随机生成  
db:  
    password: MySuperSecretDBPass  
  
# CA 证书的口令  
ssl:  
    password: MySuperSecretSSLPASSWORD  
  
# 您的 SUSE Customer Center 身份凭证  
scc:  
    user: ccUsername  
    password: ccPassword  
  
# 组织名称  
organization: YourOrganization  
  
# 用于发送通知的电子邮件地址  
emailFrom: notifications@example.com  
  
# 管理员帐户细节  
admin:  
    password: MySuperSecretAdminPass  
    login: LoginName  
    firstName: Admin  
    lastName: Admin  
    email: email@example.com  
....  
. 在终端中，以 root 身份运行以下命令。服务器 FQDN 是选填的。  
+[source, shell]
```

mgradm -c mgradm.yaml install podman <FQDN>

```
+  
[IMPORTANT]  
=====  
必须以 sudo 或 root 用户身份部署容器。如果您遗漏此步骤，终端中将显示以下错误。  
[source, shell]
```

INF Setting up uyuni network 9:58AM INF Enabling system service 9:58AM FTL Failed to open /etc/systemd/system/uyuni-server.service for writing error="open /etc/systemd/system/uyuni-server.service: permission denied"

```
=====  
. 等待部署完成。  
. 打开浏览器并访问您的服务器 FQDN 或 IP 地址。  
//In this section you learned how to deploy an {productname} {productnumber} Server container using a custom YAML configuration.  
:leveloffset: 3  
:leveloffset: +2
```

```
= 启动和停止容器
:revdate: 2024-02-13
:page-revdate: {revdate}
```

可使用以下命令重启、启动和停止 {productname} {productnumber} 服务器容器：

要[command]``重启`` {productname} {productnumber} 服务器，请执行以下命令：

3.1.8. mgradm restart

5:23PM INF Welcome to mgradm 5:23PM INF Executing command: restart

要[command]``启动``服务器，请执行以下命令：

3.1.9. mgradm start

5:21PM INF Welcome to mgradm 5:21PM INF Executing command: start

要[command]``停止``服务器，请执行以下命令：

3.1.10. mgradm stop

5:21PM INF Welcome to mgradm 5:21PM INF Executing command: stop

```
// Coming soon:
// You can also check on the status of services running in the container with:
//----
//mgradm status
//----

:leveloffset: 3
:leveloffset: +2

[[container-list]]
= {productname} 使用的容器
:revdate: 2025-05-08
:page-revdate: {revdate}
```

下面列出了 {productname} {productnumber} 使用的容器。

| .服务器容器 | |
|--------------------------|--|
| cols="name,description"] | |
| == | |
| 容器名称 说明 | |
| uyuni-server | |
| 主产品容器 | |
| uyuni-db | |
| 产品的数据库容器 | |
| uyuni-hub-xmlrpc | |
| 用于 Hub 部署的 XML-RPC 网关 | |

```

| uyuni-server-attestation
| 服务器 COCO 证明容器

| uyuni-saline
| 用于实现 Salt 可观测性的 Saline 容器

| uyuni-server-migration
| 迁移辅助容器
| ===

.代理容器
[cols="name,description"]
| ===
| 容器名称 | 说明

| uyuni-proxy-httpd
| 处理所有 HTTP 通信的主代理容器

| uyuni-proxy-squid
| Squid 缓存容器

| uyuni-proxy-salt-broker
| Salt 转发器容器

| uyuni-proxy-ssh
| SSH 转发器容器

| uyuni-proxy-tftpd
| TFTP 到 HTTP 的转换器与转发器容器
| ===

:leveloffset: 3
:leveloffset: +2

[[persistant-volume-list]]
= 永久性容器卷
:revdate: 2025-07-29
:page-revdate: {revdate}

在容器中执行的修改不会保留。在永久性卷外部所做的任何更改都将被丢弃。下面列出了
{productname} {productnumber} 的永久性卷。

要自定义默认卷位置，请确保在首次启动 Pod 之前使用 [command]``podman volume create``命令创建必要的卷。

```

[NOTE]

====

请确保此表格与 Helm 图表和 systemctl 服务定义中所述的卷映射完全一致。

====

== 服务器

以下卷存储在服务器上的 **Podman** 默认存储位置。

.永久性卷: **Podman 默认存储**

```

[cols="name,directory"]
| ===
| 卷名称 | 卷目录

| **Podman 存储**
| [path]``/var/lib/containers/storage/volumes/``
| ===

```

```

.永久性卷: **root**
[cols="name,directory"]
===
| 卷名称 | 卷目录

| **root**
| [path]``/root``
| ===

.永久性卷: **var/***
[cols="name,directory"]
===
| 卷名称 | 卷目录

| **var-cobbler**
| [path]``/var/lib/cobbler``

| **var-salt**
| [path]``/var/lib/salt``

| **var-pgsql**
| [path]``/var/lib/pgsql/data``

| **var-pgsql-backup**
| [path]``/var/lib/pgsql-backup``

| **var-cache**
| [path]``/var/cache``

| **var-spacewalk**
| [path]``/var/spacewalk``

| **var-log**
| [path]``/var/log``
| ===

.永久性卷: **srv/***
[cols="name,directory"]
===
| 卷名称 | 卷目录

| **srv-salt**
| [path]``/srv/salt``

| **srv-www**
| [path]``/srv/www/``

| **srv-tftpboot**
| [path]``/srv/tftpboot``

| **srv-formulametadata**
| [path]``/srv/formula_metadata``

| **srv-pillar**
| [path]``/srv/pillar``

| **srv-susemanager**
| [path]``/srv/susemanager``

| **srv-spacewalk**
| [path]``/srv/spacewalk``
```

```
| ===
```

```
.永久性卷: **etc/**
[cols="name,directory"]
| ===
| Volume Name | Volume Directory

| **etc-apache2**
| [path]``/etc/apache2``

| **etc-rhn**
| [path]``/etc/rhn``

| **etc-systemd-multi**
| [path]``/etc/systemd/system/multi-user.target.wants``

| **etc-systemd-sockets**
| [path]``/etc/systemd/system/sockets.target.wants``

| **etc-salt**
| [path]``/etc/salt``

| **etc-sssd**
| [path]``/etc/sssd``

| **etc-tomcat**
| [path]``/etc/tomcat``

| **etc-cobbler**
| [path]``/etc/cobbler``

| **etc-sysconfig**
| [path]``/etc/sysconfig``

| **etc-postfix**
| [path]``/etc/postfix``

| **ca-cert**
| [path]``/etc/pki/trust/anchors``

| ===
```

```
.永久性卷: **run/**
[cols="name,directory"]
| ===
| 卷名称 | 卷目录

| **run-salt-master**
| [path]``/run/salt/master``

| ===
```

== 代理

以下卷存储在代理上的 **Podman** 默认存储位置。

```
.永久性卷: **Podman 默认存储**
[cols="name,directory"]
| ===
| 卷名称 | 卷目录
```

```

| **Podman 存储**
| [path]``/var/lib/containers/storage/volumes``
| ===

.永久性卷: **srv/**
[cols="name,directory"]
| ===
| 卷名称 | 卷目录

| **uyuni-proxy-tftpboot**
| [path]``/srv/tftpboot``
| ===

.永久性卷: **var/**
[cols="name,directory"]
| ===
| 卷名称 | 卷目录

| **uyuni-proxy-rhn-cache**
| [path]``/var/cache/rhn``
| ===

| **uyuni-proxy-squid-cache**
| [path]``/var/cache/squid``
| ===

:leveloffset: 3
:leveloffset: +2

[[understanding-storage-scripts]]
= 了解 [command]``mgr-storage-server`` 和 [command]``mgr-storage-proxy``
:revdate: 2025-07-07
:page-revdate: {revdate}

```

[command]``mgr-storage-server`` 和 [command]``mgr-storage-proxy`` 是 {productname} 提供的辅助脚本。

它们用于为 {productname} 服务器和代理指定存储配置。

这些脚本以磁盘设备作为参数。[command]``mgr-storage-proxy`` 仅需要一个参数，即存储磁盘设备。[command]``mgr-storage-server`` 需要用户指定存储磁盘设备，此外还可选择传入第二个参数，即专用数据库磁盘设备。虽然普通存储设备和数据库存储设备可位于同一磁盘，但建议将数据库部署在专用的高性能磁盘上，以确保实现更优的性能和更便捷的管理。

-- 这些工具的功能

[command]``mgr-storage-server`` 和 [command]``mgr-storage-proxy`` 均会执行以下标准存储设置操作：

- * 验证提供的存储设备。
- * 确保设备为空且适合使用。
- * 在指定设备上创建 XFS 文件系统。
- * 临时挂载设备以进行数据迁移。
- * 将相关存储目录迁移到新设备。
- * 在 [path]``/etc/fstab`` 中创建条目，确保存储设备在系统引导时自动挂载。
- * 将设备重新挂载到最终位置。

.工具的专属特性

```
[cols="1,3a"]
```

```
====
```

```
 `mgr-storage-server`
```

* 选择性支持数据库存储专用设备。

* 迁移期间停止 SUSE Manager 服务，迁移完成后会重启服务。

将 Podman 卷目录 [path]``/var/lib/containers/storage/volumes``

迁移到准备好的存储设备；以及选择性将 [path]``/var/lib/containers/storage/volumes/var-
pgsql`` 迁移到准备好的数据库存储设备。

```
 | `mgr-storage-proxy`
```

* 仅专注于设置代理存储设备（不支持设置数据库存储设备）。

* 迁移期间停止并重启代理服务。

* 将 Podman 卷目录 [path]``/var/lib/containers/storage/volumes`` 迁移到准备好的存储设备。

```
|====
```

[NOTE]

```
=====
```

这两个工具均会自动执行标准的 Linux 存储操作，不存在 Linux 管理员手动操作之外的隐藏逻辑或自定义逻辑。

```
=====
```

== 这些工具**不**具备的功能

* **不会**创建或管理 LVM 卷。

* **不会**配置 RAID 或复杂存储拓扑。

* **不会**阻止您在完成设置后使用常规 Linux 工具管理存储设备。

* **不**提供动态调整大小或扩展功能 — 这些操作必须通过标准 Linux 存储工具完成。

== 安装后的存储管理

完成存储配置后，您可通过标准 Linux 命令安全地管理存储设备。

==== 示例

.示例 1：扩展存储容量（如果使用 LVM）

```
lvextend -L +10G /dev/your_vg/your_lv xfs_growfs /var/lib/containers/storage/volumes
```

.示例 2：迁移到更大容量的磁盘

. 添加并格式化新磁盘。

. 临时挂载新磁盘。

. 使用 `rsync` 复制数据。

. 更新 `/etc/fstab`。

. 将新磁盘重新挂载到适当位置。

== 适用与不适用场景

[IMPORTANT]

```
=====
```

更改存储设置前，务必创建备份。

```
=====
```

* **仅**在初始存储设置或迁移到新存储设备（需要使用工具处理数据迁移和更新 `/etc/fstab`）时使用这些工具。

* **不要**重新运行这些脚本来扩容存储空间或调整其大小。要执行此类操作，请使用标准 Linux 工具（如 [literal]``lvextend``、[literal]``xfs_growfs``）。

== 总结

[command]``mgr-storage-server`` 和 [command]``mgr-storage-proxy`` 会根据标准 Linux 存储实践，自动完成 {productname} 组件的初始持久化存储设置，且不会限制或干扰后续的标准存储管理操作。

完成设置后，您可继续使用熟悉的 Linux 工具管理存储设备。

[IMPORTANT]

====

数据库存储卷空间耗尽会对系统运行造成严重影响。由于容器化环境尚未适配磁盘使用率通知功能，建议用户通过 Grafana、Prometheus 或其他偏好的工具，自行监控 Podman 卷的磁盘空间使用情况，尤其需要关注位于 [path]``/var/lib/containers/storage/volumes/`` 下的 var-pgsql 卷。

====

```
:leveloffset: 3
:leveloffset: +1
= GNU Free Documentation License
:revdate: 2024-01-22
:page-revdate: {revdate}
```

Copyright (C) 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.

[float]
== 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. Secondary, 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.

[float]
== 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.

[float]
== 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.

[float]
== 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.

[float]
== 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:

[upperalpha]

- . 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.
- . 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.
- . State on the Title page the name of the publisher of the Modified Version, as the publisher.
- . Preserve all the copyright notices of the Document.
- . Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- . 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.
- . Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- . Include an unaltered copy of this License.
- . 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.
- . 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.
- . 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.
- . 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.
- . Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- . Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- . 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.

[float]
== 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".

[float]
== 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.

[float]
== 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.

[float]
== 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.

[float]
== 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.

[float]
== 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.

[float]
== 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".

:leveloffset: 3