

Uyuni 2025.10

# 安装和升级指南



U Y U N I

# 章 1. Preface

Installation, Deployment and Upgrade + Uyuni 2025.10

This guide provides comprehensive, step-by-step instructions for deploying, upgrading, and managing Uyuni Server and Proxy.

It is organized into the following sections:

- **Requirements:** Outlines the essential hardware, software, and networking prerequisites to ensure a smooth setup.
  - **Deployment and Installation:** Guides you through deploying Uyuni as a container and completing the initial configuration.
  - **Upgrade and Migration:** Details the process for upgrading and migrating Uyuni while minimizing downtime.
  - **Basic Server Management:** Covers fundamental server operations, helping you get started with Uyuni efficiently.

Publication Date: 2025-10-31

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

# 目录

|   |           |
|---|-----------|
| <b>1. Preface</b>   | <b>1</b>  |
| <b>2. 要求</b>  | <b>3</b>  |
| 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. Reenable router advertisements   | 5         |
| 2.2.4. Deployment behind HTTP or HTTPS OSI level 7 proxy                                | 5         |
| 2.2.5. 物理隔离的部署  | 6         |
| 2.2.6. 所需的网络端口  | 6         |
| 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        |
| <b>3. 部署和安装</b>   | <b>15</b> |
| 3.1. Install Uyuni Server   | 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  | 45        |
| 3.1.5. With tuning files  | 45        |
| 3.1.6. If you have no tuning files, remove the tuning parameters:                       | 45        |
| 3.1.7. mgrpxy install podman  | 46        |
| 3.1.8. mggradm restart  | 52        |
| 3.1.9. mggradm start  | 52        |
| 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:<br>100 GB (this will be verified by the implemented check)<br><br>* 每个 SUSE 产品和软件包中心<br>50 GB<br><br>为每个 Red Hat 产品提供 360 GB<br>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。<br>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 forwarding will be enabled by containerized installation. This means Uyuni Server and Proxies will behave as a router. This behavior is done by podman directly. Podman containers do not run if IP forwarding is disabled.

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. Reenable router advertisements

When the Uyuni is installed using **mgradm install podman** or **mgrpxy install podman**, it sets up Podman which enables IPv4 and IPv6 forwarding. This is needed for communication from the outside of the container.

However, if your system previously had **/proc/sys/net/ipv6/conf/eth0/accept\_ra** set to **1**, it will stop using router advertisements. As a result, the routes are no longer obtained via router advertisements and the default IPv6 route is missing.

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

### Procedure: Reenabling router advertisements

1. Create a file in **/etc/sysctl.d**, for example **99-ipv6-ras.conf**.
2. Add the following parameter and value to the file:

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

3. 重引导。

## 2.2.4. Deployment behind HTTP or HTTPS OSI level 7 proxy

Some environments enforce internet access through a HTTP or HTTPS proxy. This could be a Squid server or similar. To allow the Uyuni Server internet access in such configuration, you need to configure the following.

### Procedure: Configuring HTTP or HTTPS OSI level 7 proxy

1. For operating system internet access, modify **/etc/sysconfig/proxy** according to your needs:

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

2. For **Podman** container internet access, modify **/etc/systemd/system/uyuni-server.service.d/custom.conf** according to your needs. For example, set:

```
[Service]
Environment=TZ=Europe/Berlin
```

```
Environment="PODMAN_EXTRA_ARGS="
Environment="https_proxy=user:password@http://192.168.10.1:3128"
```

3. For Java application internet access, modify **/etc/rhn/rhn.conf** according to your needs. On the container host, execute **mgrctl term** to open a command line inside the server container:

- a. Modify **/etc/rhn/rhn.conf** according to your needs. For example, set:

```
# Use proxy FQDN, or FQDN:port
server.satellite.http_proxy =
server.satellite.http_proxy_username =
server.satellite.http_proxy_password =
# no_proxy is a comma seperated list
server.satellite.no_proxy =
```

4. On the container host, restart the server to enforce the new configuration:

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

## 2.2.5. 物理隔离的部署

If you are on an internal network and do not have access to SUSE Customer Center, you can use an **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](#)。

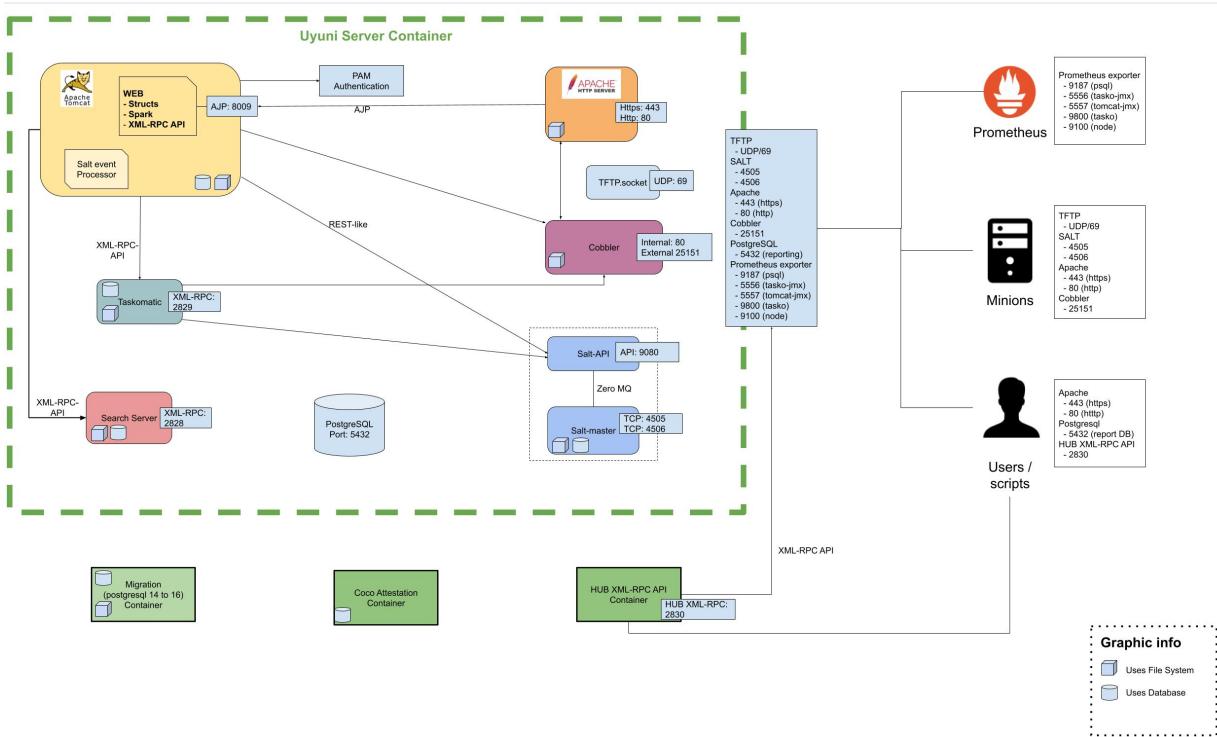
## 2.2.6. 所需的网络端口

本节提供了 Uyuni 中各种通讯使用的端口的综合列表。

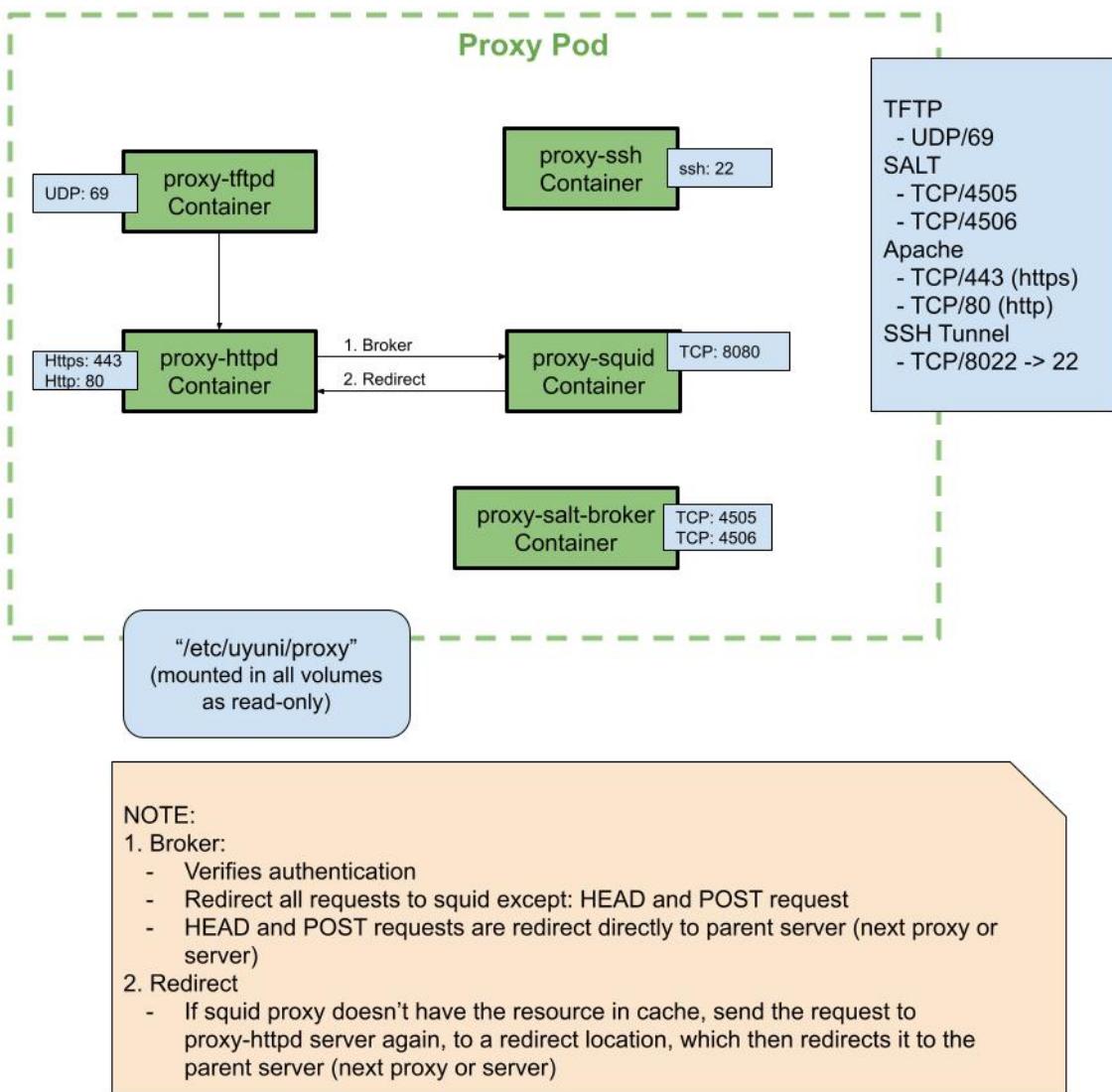
您不需要打开所有这些端口。某些端口只有在您使用需要这些端口的服务时才需打开。

### 2.2.6.1. 概览

#### 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**

| Port number | Protocol | Used By | Notes   |
|-------------|----------|---------|---|
| 67          | TCP/UDP  | DHCP    | Required only if clients are requesting IP addresses from the server.         |
| 69          | TCP/UDP  | TFTP    | Required if server is used as a PXE server for automated client installation. |

| Port number | Protocol | Used By    | Notes   |
|-------------|----------|------------|---|
| 80          | TCP      | HTTP       | Required temporarily for some bootstrap repositories and automated installations.   |
| 443         | TCP      | HTTPS      | Serves the Web UI, client, and server and proxy ( <b>tftpsync</b> ) requests.   |
| 4505        | TCP      | salt       | Required to accept communication requests from clients. The client initiates the connection, and it stays open to receive commands from the Salt master.  |
| 4506        | TCP      | salt       | Required to accept communication requests from clients. The client initiates the connection, and it stays open to report results back to the Salt master. |
| 5432        | TCP      | PostgreSQL | Required to access the reporting database.  |
| 5556        | TCP      | Prometheus | Required for scraping Taskomatic JMX metrics.   |
| 5557        | TCP      | Prometheus | Required for scraping Tomcat JMX metrics.   |
| 9100        | TCP      | Prometheus | Required for scraping Node exporter metrics.  |
| 9187        | TCP      | Prometheus | Required for scraping PostgreSQL metrics.   |
| 9800        | TCP      | Prometheus | Required for scraping Taskomatic metrics.   |
| 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讯。

**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

| Port number | Protocol | Used By | Notes   |
|-------------|----------|---------|---|
| 22          |          |         | Only required if the user wants to manage the proxy host with Salt SSH.           |
| 67          | TCP/UDP  | DHCP    | Required only if clients are requesting IP addresses from the server.             |
| 69          | TCP/UDP  | TFTP    | Required if the server is used as a PXE server for automated client installation. |
| 443         | TCP      | HTTPS   | Web UI, client, and server and proxy ( <b>tftpsync</b> ) requests.                |

| Port number | Protocol | Used By | Notes   |
|-------------|----------|---------|---|
| 4505        | TCP      | salt    | Required to accept communication requests from clients. The client initiates the connection, and it stays open to receive commands from the Salt master.  |
| 4506        | TCP      | salt    | Required to accept communication requests from clients. The client initiates the connection, and it stays open to report results back to the Salt master. |
| 8022        |          |         | Required for ssh-push and ssh-push-tunnel contact methods. Clients connected to the proxy initiate check in on the server and hop through to clients.     |

### 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讯。

**表格 7. Uyuni Proxy 的外部端口要求**

| Port number | Protocol | Used By | Notes  |
|-------------|----------|---------|--|
| 80          |          |         | Used to reach the server.  |
| 443         | TCP      | HTTPS   | Required for SUSE Customer Center.                               |
| 4505        | TCP      | Salt    | Required to connect to Salt master either directly or via proxy. |
| 4506        | TCP      | Salt    | Required to connect to Salt master either directly or via proxy. |

### 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**

| Port number | Direction | Protocol | Notes  |
|-------------|-----------|----------|--|
| 22          | Inbound   | SSH      | Required for ssh-push and ssh-push-tunnel contact methods. |
| 80          | Outbound  |          | Used to reach the server or proxy.                         |

| Port number | Direction | Protocol | Notes  |
|-------------|-----------|----------|--|
| 443         | Outbound  |          | Used to reach the server or proxy.                               |
| 4505        | Outbound  | TCP      | Required to connect to Salt master either directly or via proxy. |
| 4506        | Outbound  | TCP      | Required to connect to Salt master either directly or via proxy. |
| 9090        | Outbound  | TCP      | Required for Prometheus user interface.                          |
| 9093        | Outbound  | TCP      | Required for Prometheus alert manager.                           |
| 9100        | Outbound  | TCP      | Required for Prometheus node exporter.                           |
| 9117        | Outbound  | TCP      | Required for Prometheus Apache exporter.                         |
| 9187        | Outbound  | TCP      | Required for Prometheus PostgreSQL.                              |

## 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](http://scc.suse.com)
- [updates.suse.com](http://updates.suse.com)
- [installer-updates.suse.com](http://installer-updates.suse.com)
- [registry.suse.com](http://registry.suse.com)
- [registry-storage.suse.com](http://registry-storage.suse.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 注册表](#)。

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. Install Uyuni Server

There are various scenarios to deploy a Uyuni Server.

### 3.1.1. Uyuni Server Deployment on openSUSE Tumbleweed

#### 3.1.1.1. Deployment Preparations

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. Container Host General Requirements

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/>

This procedure is described below.

#### 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. Installing Uyuni Tools For Use With Containers

##### 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](#)[]。

====

-- Deploying an Uyuni Container With Podman

==== [command]``mgradm`` Overview

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]

```
=====
{productname} server hosts that are hardened for security may restrict execution of files
from the [path]``/tmp`` folder. In such cases, as a workaround, export the
[literal]``TMPDIR`` environment variable to another existing path before running
[command]``mgradm``.
```

例如：

[source, shell]

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

```
In {productname} updates, tools will be changed to make this workaround unnecessary.
=====
```

: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]

```
=====
```

If you are just testing out {productname} you do not need to specify these volumes.  
[command]``mgradm`` will setup the correct volumes by default.

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。

You can provide custom storage for the volumes by mounting disks on this path or the expected volume path inside it such as: [path]``/var/lib/containers/storage/volumes/var-spacewalk``. This is especially important for the database and package mirrors.

For a list of all persistent volumes in the container, see:

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

:leveloffset!:  
:leveloffset: +3

= {productname} 服务器物理隔离的部署

:revdate: 2025-07-25

:page-revdate: {revdate}

== 什么是物理隔离的部署?

物理隔离部署是指设置和操作与不安全网络（尤其是互联网）物理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.

.过程

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:

+

.Load the server image

podman load -i images.tar

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

{productname} also provides all the needed container images in RPM packages that can be installed on the system.

## [NOTE]

====

User should make the needed RPM available on the internal network. That can be done by using a second {productname} Server or any kind of mirror.

====

- . Procedure: Install {productname} on {opensuse} {tumbleweed} in Air-gapped
- . Install {opensuse} {tumbleweed}.
- . 更新系统。
- . Install tools packages and image packages (replace \$ARCH\$ with the correct architecture):

+

## [source, shell]

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

+

- . Deploy {productname} with [command]``mgradm``. In an Air-gapped environment you may want to use the option [command]``--pullPolicy Never``.

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 配置存档文件
- . 将配置存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

-- Generate Proxy Configuration

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

For the containerized {productname} Proxy, you must build a new proxy configuration file and then redeploy the container for the changes to take effect. This is the process for updating settings, including the SSL certificate.

```
//[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。

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

.Procedure: Generating a Proxy Container Configuration Using {webui}
[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存档 ([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。

-----

**==== Generate Proxy Configuration With [literal]``spacecmd`` and Self-Signed Certificate**

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
```

-----

**==== Generate Proxy Configuration With [literal]``spacecmd`` and Custom Certificate**

You can generate a Proxy configuration using [literal]``spacecmd`` for custom certificates rather than the default self-signed certificates.

- . 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'
```

```
+
. If your setup uses an intermediate CA, copy it as well and include it in the command
with the '-i' option (can be provided multiple times if needed) :
+
```

```
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]]
== Transfer {productname} Proxy Configuration
```

Both [command]``spacecmd`` command and generating via {webui} ways create a configuration archive. This archive needs to be made available on container host. Transfer this generated archive to the container host.

```
[[proxy-setup-containers-transfer-start]]
== Start {productname} Proxy Containers
```

Container can be started with the [literal]``mgrpxy`` command.

```
[[proc-setup-containers-setup-start]]
.Procedure: Start {productname} Proxy Containers
```

```
. Run command:
+
```

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

```
+
. Check if all containers started up as expected by calling:
+
```

```
podman ps
```

Five {productname} Proxy containers should be present and should be part of [literal]``proxy-pod`` container pod.

\* 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。

| 代理硬件要求                               |  |  |
|--------------------------------------|--|--|
| ====                                 |  |  |
| Hardware                             |  |  |
| Details                              |  |  |
| Recommendation                       |  |  |
| CPU                                  |  |  |
| {x86_64}, {arm}                      |  |  |
| Minimum 2 dedicated 64-bit CPU cores |  |  |
| RAM                                  |  |  |
| Minimum                              |  |  |
| 2 GB                                 |  |  |
| Recommended                          |  |  |
| 8 GB                                 |  |  |
| Disk Space                           |  |  |

```

| [path]``/`` (root directory)
| Minimum 40 GB

| [path]``/var/lib/containers/storage/volumes``
| Minimum 100 GB, Storage requirements should be calculated for the number of ISO
distribution images, containers, and bootstrap repositories you will use.

| ===

```

## == Container Host General Requirements

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。

====

## == Installing Uyuni Tools for Use With Containers

.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}.
- . Log in.
- . 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} 代理的配置存档由 {productname} 服务器生成。每个附加代理都需要自身的配置存档。

[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：

. Procedure:

- . 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。

+

在服务器上生成的 CA 证书存储在 [path]``/var/lib/containers/storage/volumes/root/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} 服务器上注册新代理 FQDN，并生成包含容器主机细节的配置存档。
- . 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} 将生成配置存档。需要在代理容器主机上提供此存档。

- . 任务：复制代理配置

- . 将服务器容器中的文件复制到服务器主机操作系统:

+

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

- . 接下来, 将服务器主机操作系统中的文件复制到代理主机:

+

```
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-httpd
* 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注册表路径。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-httpd
/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-https``` 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]]`  
`= Proxy conversion from client`

`-- 概览`

This chapter describes how to convert a client system into a {productname} Proxy using the {webui}.

It assumes that the proxy host system has already been bootstrapped and subscribed to the base operating system channel.

For information about client onboarding, see [xref:client-configuration:registration-overview.adoc\[\]](#).

`-- 要求`

Before starting the conversion, ensure the following requirements are fulfilled.

`==== Client Must Be`

- Already onboarded in {productname}
- Reachable via the network

`-- Preparation`

Before proceeding with the proxy conversion, make sure the following preparations are completed to avoid interruptions during the conversion process.

`==== SSL Certificates`

Valid SSL certificates are required to secure communication between the proxy and other components.

You need:

- \* The public certificate of the Certificate Authority (CA) that signed the certificate on the {productname} server
- \* A certificate for the proxy.
- \* The corresponding private key for the proxy certificate.

`[NOTE]`

`=====`

If your CA uses an intermediate certificate chain, you must include all intermediate

certificates as well.

====

If you are not using third party certificates, you can generate them using the `rhn-ssl-tool` inside the {productname} container.

.Generate a proxy certificate

- . On the {productname} server host, run:

+

[source, shell]

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

+

For more information about other parameters, see [xref:administration:ssl-certs-selfsigned.adoc\[\]](#).

+

- . Transfer the certificates to {productname} server host

+

[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]

====

To confirm the exact folder where the certificates and key files were generated, you can list the directories with:

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

====

- . Transfer the certificates from {productname} server host

+

[source, shell]

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

==== Packages Preparation

==== Install `mgrpxy`

The `mgrpxy` tool must be installed from a repository matching your system. Choose the appropriate repository from:

<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
```

===== Install Container Images

It is recommended to deploy the container images as RPM packages. Please ensure the following packages are installed on the client:

[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
```

For details on air-gapped deployment, see [xref:installation-and-upgrade:container-deployment/mlm/proxy-air-gapped-deployment-mlm.adoc\[\]](#)

== Setup Proxy Client

- . Navigate to the client's [literal]``Overview`` page.
- . Click button `btn:[Convert to Proxy]`.

+  
Confirm you were redirected to the proxy configuration form.

+  
This page can be accessed later from the [literal]``Details`` > [literal]``Proxy`` > [literal]``Configuration`` tab.

- . In the {webui}, navigate to `menu:Proxy[Configuration]` and fill in the required data:

+  
.Procedure: Configuring the Proxy  
.. In the [guimenu]``Parent FQDN`` field, type the fully qualified domain name for the parent server or proxy.  
.. In the [guimenu]``Proxy SSH port`` field, type the SSH port on which the SSH service is listening on the {productname} Proxy. It is recommended to keep the default: 8022.  
.. In the [guimenu]``Max Squid cache size`` field, type the maximum allowed size for the Squid cache, in Gigabytes.  
.. In the [guimenu]``Proxy admin email`` field, type the administrator's email address.  
.. In the [literal]``Certificates`` section, provide the certificates for the {productname} Proxy, obtained in the preparation step.

.. In the [literal]``Source`` section, select one of the two options: [literal]``RPM`` or [literal]``Registry``.

+  
\* The [literal]``RPM`` option is recommended for air-gapped or restricted environments. 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``.

+  
\*\* If [literal]``Simple`` is selected, provide values in the [literal]``Registry URL`` and [literal]``Containers Tag`` fields.  
\*\*\* For [literal]``Registry URL`` use: [literal]``registry.opensuse.org/uyuni``.  
\*\*\* Select the tag from the drop-down list.

\*\* If [literal]``Advanced`` is selected, an additional section of the form is shown:  
\*\*\* For each individual container URL field, use the registry:

[literal]``registry.opensuse.org/uyuni`` followed by the corresponding suffix, for example, `\_proxy-httpd\_` or `\_salt-broker\_`.  
\*\*\* Select the tag from the drop-down list.

- . Once all fields are filled, click btn:[Apply] to apply the configuration and schedule the proxy installation task.

#### == Verify Proxy Activation

Check the client's event history to confirm task success.

(Optional) Access the proxy's HTTP endpoint to validate it shows a welcome page.

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

```
[[installation-proxy-containers-k3s-uyuni]]
= {productname} Proxy Deployment on K3s
: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。

Install Helm by using the installer script:

```
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
```

For more information, see <https://helm.sh/docs/intro/install/#from-script>.

The [literal]``mgrpxy`` package is available in the container utils repository. Pick the one matching the distribution in:  
<https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable:/ContainerUtilities/>.

- . 过程
  - . To install package on Leap Micro run:

```
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`

For more information see:

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

: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: Prepareing 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
```

```
= Legacy {productname} Server Migration to Container
:revdate: 2025-07-30
:page-revdate: {revdate}
```

To migrate a legacy {productname} Server to a container, a new machine is required.

In the context of this migration, the legacy {productname} Server (RPM installation) is sometimes also called \_old server\_.

## == 要求和注意事项

### ==== 主机名

Neither in-place migration is not possible nor allows the migration procedure currently any hostname renaming functionality.

Thus the fully qualified domain name (FQDN) on the new server will remain identical to that on the legacy server.

[IMPORTANT]

=====

After migration, it is necessary to update the DHCP and DNS records to point to the new server.

For more information, see <>migration-finalization>>.

=====

### ==== SSL certificates

:leveloffset: +1

SSL certificates are needed at a later stage. If not using the self-signed generated CA and certificates, ensure you have the following before starting:

- \* A certificate authority (CA) SSL public certificate. If you are using a CA chain, all intermediate CAs must also be available.
- \* An SSL database private key.
- \* An SSL database certificate.

All files must be in PEM format.

The hostname of the SSL server certificate must match the fully qualified hostname of the machine you deploy them on. You can set the hostnames in the [literal]``X509v3 Subject Alternative Name`` section of the certificate. You can also list multiple hostnames if your environment requires it. Supported Key types are [literal]``RSA`` and [literal]``EC`` (Elliptic Curve).

[IMPORTANT]

=====

Database SSL certificate requires [literal]``reportdb`` and [literal]``db`` and the FQDN used to access the report database as [literal]``Subject Alternative Name``.

=====

: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 keys

- \* 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。
- \* The administrator must migrate these keys manually from the legacy {productname} installation to the container host after the actual server migration.

#### .Procedure: Manual Migration of the GPG Keys to New Server

- . Copy the keys from the legacy Uyuni server to the container host of the new server.
- . 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。

### == Initial Preparation on the Legacy Server

The migration can take a very long time depending on the amount of data that needs to be replicated. To reduce downtime it is possible to run the migration multiple times in a process of \_initial replication\_, \_re-replication\_, or \_final replication\_ and switch over while all the services on the legacy server can stay up and running.

Only during the final migration the processes on the legacy server need to be stopped.

For all non-final replications add the parameter [command]``--prepare`` to prevent the automatic stopping the services on the legacy server. For example:

[source, shell]

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

#### .Procedure: Initial Preparation on the Legacy Server

- . 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
- . Ensure that for [systemitem]``root`` an SSH key exists on the new {productnumber} server. If a key does not exist, create it with the command:
  - +

ssh-keygen -t rsa

- . The SSH configuration and agent should be ready on the new server host for a connection to the legacy server that does not prompt for a password.
- +

---

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

```
+
[NOTE]
====
```

To establish a connection without prompting for a password, the migration script relies on an SSH agent running on the new server. If the agent is not active yet, initiate it by running [command]``eval \$(ssh-agent)``. Then add the SSH key to the running agent with [command]``ssh-add`` followed by the path to the private key. You will be prompted to enter the password for the private key during this process.

```
====
```

. Copy the public SSH key to the legacy {productname} Server ([literal]<oldserver.fqdn>) with [command]``ssh-copy-id``. Replace [literal]<oldserver.fqdn> with the FQDN of the legacy server:

```
+
```

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

```
+
```

The SSH key will be copied into the legacy server's [path]``~/.ssh/authorized\_keys`` file. For more information, see the [literal]``ssh-copy-id`` manpage.

. Establish an SSH connection from the new server to the legacy {productname} Server to check that no password is needed. Also there must not be any problem with the host fingerprint. In case of trouble, remove old fingerprints from the [path]``~/.ssh/known\_hosts`` file. Then try again. The fingerprint will be stored in the local [path]``~/.ssh/known\_hosts`` file.

```
==== 执行迁移
```

When planning your migration from a legacy {productname} to a containerized {productname}, ensure that your target instance meets or exceeds the specifications of the legacy setup. This includes, but is not limited to, memory (RAM), CPU Cores, Storage, and Network Bandwidth.

```
:leveloffset: +2
```

```
[NOTE]
```

```
====
```

{productname} server hosts that are hardened for security may restrict execution of files from the [path]``/tmp`` folder. In such cases, as a workaround, export the [literal]``TMPDIR`` environment variable to another existing path before running [command]``mgradm``.

例如:

```
[source, shell]
```

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

```
In {productname} updates, tools will be changed to make this workaround unnecessary.
```

```
====
```

```
: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]

=====

Devices must not have any filesystem. The command aborts if a filesystem exists on the storage device.

=====

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

- . Execute the following command to install a new {productname} server. Replace [literal]``<oldserver.fqdn>`` with the FQDN of the legacy server:

+

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

- . 迁移可信 SSL CA 证书。

===== 证书的迁移

Trusted SSL CA certificates that were installed as part of an RPM and stored on a legacy {productname} in the [path]``/usr/share/pki/trust/anchors`` directory will not be migrated. Because {suse} does not install RPM packages in the container, the administrator must migrate these certificate files manually from the legacy installation after migration:

- . 过程：迁移证书

- . Copy the file from the legacy server to the new server.  
例如，复制为 [path]``/local/ca.file``。
- . Copy the file into the container with the command:  
+

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

[[migration-finalization]]  
==== Finalize migration  
  
[IMPORTANT]  
====  
After successfully running the [command]``mgradm migrate`` command, the {salt} setup on all clients will still point to the legacy server.

To redirect them to the new {productnumber} server, it is required to rename the new server at the infrastructure level (DHCP and DNS) to use the same FQDN and IP address as the legacy server.

====

[IMPORTANT]  
====  
If something goes wrong with the migration it is possible to restart the old system. As root, restart {postgresql} and the spacewalk services with the following commands:

[source, shell]

```
service postgresql start spacewalk-service start
```

====

// FIXME: check the following! Partially probably already covered above.!  
== Kubernetes Preparations

Before executing the migration with [command]``mgradm migrate`` command, it is essential to predefine \*\*Persistent Volumes\*\*, especially considering that the migration job initiates the container from scratch.

For more information, see the installation section on preparing these volumes in [xref:installation-and-upgrade:container-management/persistent-container-volumes.adoc\[\]](#).

== 迁移

Execute the following command to install a new {productname} server, replacing \*\*<oldserver.fqdn>\*\* with the appropriate FQDN of the legacy server:

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

或

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

## [IMPORTANT]

=====

After successfully running the [command]``mgradm migrate`` command, the {salt} setup on all clients will still point to the legacy server. To redirect them to the new server, it is required to rename the new server at the infrastructure level (DHCP and DNS) to use the same FQDN and IP address as the legacy server.

=====

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

= {productname} Server Upgrade
:revdate: 2025-07-29
:page-revdate: {revdate}
```

Before running the upgrade command, it is required to update the host operating system. Updating the host operating system will also result in the update of the {productname} tooling such as the [literal]``mgradm`` tool.

## . 过程：升级服务器

- . Refresh software repositories with [command]``zypper``:

+

[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。

#### .Upgrading with third-party SSL certificate

## [IMPORTANT]

====

If you are using third-party certificates, the database container needs to have an SSL certificate with the following Subject Alternate Names (SANs):

- \* [literal]``db``
- \* [literal]``reportdb``
- \* the externally facing fully qualified domain name

The same certificate can be used for both the main container and the database one, but it needs to have those SANs too.

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]

====

If you do not specify the tag parameter, it will default to upgrading to the most recent version. To upgrade to a specific version, provide the tag parameter with the desired image tag.

====

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
```

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

```
:leveloffset: +1
```

```
= Database Backup Volume
:revdate: 2025-07-29
:page-revdate: {revdate}
```

Server migration or upgrade with [command]``mgradm migration`` or [command]``mgradm upgrade`` can create a volume with the database backup.

When the {postgresql} database version is increased, the old database must be stored in a separate location before running the upgrade. For this purpose [command]``mgradm`` dynamically creates the volume [path]``var-pgsql-backup``. When the migration or upgrade is done and the user has validated that the new system is working as expected, this volume can be removed safely.

```
: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

= Legacy Proxy Migration to Container  
:revdate: 2025-02-06  
:page-revdate: {revdate}

The containerized proxy now is managed by a set of systemd services. For managing the containerized proxy, use the [command]``mgrpxy`` tool.

This section will help you migrate from the legacy [systemitem]``systemd`` proxy using the [command]``mgrpxy`` tool.

[IMPORTANT]

====

An in-place migration from previous releases of {productname} to {productnumber} will remain unsupported due to the HostOS change from {leap} to {leapmicro}.

The traditional contact protocol is no longer supported in {productname} {productnumber} and later. Before migrating from previous {productname} releases to {productnumber}, any existing traditional clients including the traditional proxies must be migrated to {salt}.  
====

**== Migrate From Legacy to Containerized Proxy With Systemd**

**==== Generate Proxy Configuration**

.Procedure: Generate the Proxy Configuration

- . 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。
- . Enter the Proxy port number. \_\_We recommend using the default port of 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**

.Procedure: Transferring the Proxy Configuration

- . 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-httpd proxy-tftpd proxy-squid proxy-ssh

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

.Procedure: Migrate {productname} Containerized Proxy to {productname} {productnumber} New Containerized Proxy

- . 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。
- . Copy your [literal]``tar.gz`` proxy configuration to the host.

== Install Packages Using the {webui}

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。

.Procedure: Installing Packages Using the {webui}

- . After installation, ensure that the {sle-micro} {microversion} parent channel and Proxy child channels are added and synchronized from the menu:Admin[Setup Wizard -> Products] page.
- . In the {webui}，go to menu:Systems[Activation Keys] and create an activation key linked for the synchronized {sle-micro} {microversion} channel.
- . 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。

== Generate Proxy Config With [literal]``spacecmd`` and Self-Signed Certificate

可以使用 spacecmd 生成代理配置。

.Procedure: Generate Proxy Config With [literal]``spacecmd`` and Self-Signed Certificate

- . 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:
- +

```
mgrpctx install podman config.tar.gz
```

```
== Generate Proxy Config With [literal]``spacecmd`` and Custom Certificate
You can generate Proxy configuration using [literal]``spacecmd`` for a custom certificates
rather than default self-signed certificates.
```

[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。

====

.Procedure: Generate Proxy Config With [literal]``spacecmd`` and Custom Certificate

- . 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:
- +

```
mgrpctx install podman config.tar.gz
```

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

= {productname} Proxy Upgrade
:revdate: 2025-05-05
:page-revdate: {revdate}
```

Before running the upgrade command, it is required to update the host operating system.  
Updating the host operating system will also result in the update of the {productname} tooling such as the [literal]``mgrpctx`` tool.

.Procedure: Upgrading Proxy

- . Refresh software repositories with [command]``zypper``:
- +

[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``。

+  
[source, shell]

. The {productname} Proxy containers running on [literal]``podman`` can be updated using the following command:

+  
[source, shell]

mgrpxy upgrade podman

+  
[source, shell]

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

[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]  
====

We highly recommend using the same tag for all proxy containers to ensure consistency under normal circumstances.

====

对于物理隔离的安装, 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]]

```
= Upgrade Clients
: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

```
= 使用 [command]``mgradm`` 进行自定义 YAML 配置和部署
: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。

For security, \*\*using command line parameters to specify passwords should be avoided\*\*.  
Use a configuration file with proper permissions instead.

=====

.Procedure: Deploying the {productname} Container with Podman Using a Custom Configuration File

. 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动、启动和停止 {productname} {productnumber} 服务器容器：

要[command]``重启动`` {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]]  
= Containers used by {productname}  
:revdate: 2025-05-08  
:page-revdate: {revdate}  
  
Below is a list of containers used by {productname} {productnumber}.  
  
.Server Containers  
[cols="name,description"]  
|===  
|Container Name | Description  
  
| uyuni-server  
| Main product container  
  
| uyuni-db  
| Database container for the product  
  
| uyuni-hub-xmlrpc  
| XML-RPC gateway for Hub deployment  
  
| uyuni-server-attestation  
| Server COCO attestation  
  
| uyuni-saline  
| Saline container for Salt observability  
  
| uyuni-server-migration  
| Migration helper container  
|===  
  
.Proxy Containers  
[cols="name,description"]  
|===  
|Container Name | Description  
  
| uyuni-proxy-httpd  
| Main proxy container handling all HTTP communication  
  
| uyuni-proxy-squid  
| Squid cache  
  
| uyuni-proxy-salt-broker  
| Salt forwarder  
  
| uyuni-proxy-ssh  
| SSH forwarder  
  
| uyuni-proxy-tftpd
```

```
| TFTPd to HTTP translator and forwarder
| ===
:leveloffset: 3
:leveloffset: +2

[[persistant-volume-list]]
= Persistent Container Volumes
: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"]  
| ===  
| Volume Name | Volume Directory  
| \*\*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``

| ===

```

```

.Persistent Volumes: **run**
[cols="name,directory"]
| ===
| Volume Name | Volume Directory
|
| **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]]
= Understanding [command]``mgr-storage-server`` and [command]``mgr-storage-proxy``
:revdate: 2025-07-07
:page-revdate: {revdate}
```

[command]``mgr-storage-server`` and [command]``mgr-storage-proxy`` are helper scripts provided with {productname}.

They are designed to configure storage for {productname} Server and Proxy.

The scripts take disk devices as arguments. [command]``mgr-storage-proxy`` requires a single argument for the storage disk device. [command]``mgr-storage-server`` requires a storage disk device and can optionally accept a second argument for a dedicated database disk device. While both normal and database storage can reside on the same disk, it is advisable to place the database on a dedicated, high-performance disk to ensure better performance and easier management.

## == What these tools do

Both [command]``mgr-storage-server`` and [command]``mgr-storage-proxy`` perform standard storage setup operations:

- \* Validate the provided storage devices.
- \* Ensure that devices are empty and suitable for use.
- \* Create XFS filesystems on the specified devices.
- \* Mount the devices temporarily for data migration.
- \* Move the relevant storage directories to the new devices.
- \* Create entries in [path]``/etc/fstab`` so that the storage mounts automatically on boot.
- \* Remount the devices at their final locations.

### .Additional tool-specific behavior

```
[cols="1,3a"]
=====
`mgr-storage-server`
=====
* Optionally supports a separate device for database storage.
* Stops SUSE Manager services during migration, restarts them afterward.
Moves Podman volumes directory [path]``/var/lib/containers/storage/volumes`` to the
prepared storage, and optionally [path]``/var/lib/containers/storage/volumes/var-pgsql``  

to the prepared database storage.
```

### | `mgr-storage-proxy`

```
=====
* Focuses only on proxy storage (no database storage support).
* Stops and restarts the proxy service during migration.
* Moves podman volumes directory [path]``/var/lib/containers/storage/volumes`` to the
prepared storage.
=====
```

## [NOTE]

```
=====
Both tools automate standard Linux storage operations. There is no hidden or custom logic
beyond what a Linux administrator would do manually.
=====
```

## == What these tools do \*not\* do

- \* They do \*not\* create or manage LVM volumes.
- \* They do \*not\* configure RAID or complex storage topologies.
- \* They do \*not\* prevent you from managing storage using normal Linux tools after setup.
- \* They do \*not\* provide dynamic resizing or expansion capabilities — these must be handled using standard Linux storage tools.

## == Post-installation storage management

Once storage has been configured, you can safely manage it using standard Linux commands.

### ==== Examples

#### .Example 1: Extending storage if using LVM

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

#### .Example 2: Migrating to a larger disk

- . Add and format the new disk.
- . Mount it temporarily.
- . Use 'rsync' to copy data.
- . Update '/etc/fstab'.
- . Remount at the correct location.

## == When to use, or not use

### [IMPORTANT]

=====

Always take a backup before making changes to your storage setup.

=====

- \* Use these tools \*only\* during initial storage setup or when migrating to new storage where the tool is expected to handle data migration and update '/etc/fstab'.
- \* Do \*not\* rerun these scripts for resizing or expanding storage. Use standard Linux tools (e.g., [literal]`lvextend`, [literal]`xfs\_growfs`) for such operations.

## == Summary

[command]`mgr-storage-server` and [command]`mgr-storage-proxy` help automate the initial persistent storage setup for {productname} components using standard Linux storage practices. They do not limit or interfere with standard storage management afterward.

After setup, continue managing your storage using familiar Linux tools.

### [IMPORTANT]

=====

A full database volume can cause significant issues with system operation. As disk usage notifications have not yet been adapted for containerized environments, users are encouraged to monitor the disk space used by Podman volumes themselves, either through tools such as Grafana, Prometheus, or any other preferred method. Pay particular attention to the var-pgsql volume, located under [path]`/var/lib/containers/storage/volumes/`.

=====

: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.  
 Secondarily, this License preserves for the author and publisher a way to get credit for  
 their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document  
 must themselves be free in the same sense.

It complements the GNU General Public License, which is a copyleft license designed for  
 free software.

We have designed this License in order to use it for manuals for free software, because  
 free software needs free documentation: a free program should come with manuals providing  
 the same freedoms that the software does.

But this License is not limited to software manuals; it can be used for any textual work,  
 regardless of subject matter or whether it is published as a printed book.

We recommend this License principally for works whose purpose is instruction or reference.

```
[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