



SUSE Multi-Linux Manager 5.2 Beta 2

Installation and Upgrade Guide

Preface

Installation, Deployment and Upgrade | SUSE Multi-Linux Manager 5.2 Beta 2

This guide provides comprehensive, step-by-step instructions for deploying, upgrading, and managing SUSE Multi-Linux Manager 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 SUSE Multi-Linux Manager as a container and completing the initial configuration.
- **Upgrade and Migration:** Details the process for upgrading and migrating SUSE Multi-Linux Manager while minimizing downtime.
- **Basic Server Management:** Covers fundamental server operations, helping you get started with SUSE Multi-Linux Manager efficiently.

Publication Date: 2026-05-22

Copyright © 2011–2026 SUSE LLC and contributors. All rights reserved. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled **Legal Guide › GNU Free Documentation License**.

For SUSE trademarks, see <https://www.suse.com/company/legal/> . All third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks. All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors nor the translators shall be held liable for possible errors or the consequences thereof.

Table of Contents

Preface	1
1. Requirements	4
1.1. General Requirements	4
1.1.1. SUSE Customer Center Account and Credentials	4
1.1.2. Supported Browsers for SUSE Multi-Linux Manager Web UI	4
1.1.3. SSL Certificates	5
1.2. Hardware Requirements	5
1.2.1. Server Requirements	5
1.2.2. Proxy Requirements	7
1.2.3. Swap space	8
1.2.4. Database Requirement	8
1.2.5. Persistent Storage and Permissions	9
1.2.6. Logical Volume Management (LVM)	10
1.3. Network Requirements	10
1.3.1. Fully Qualified Domain Name (FQDN)	11
1.3.2. Hostname and IP Address	11
1.3.3. Reenable router advertisements	11
1.3.4. Deployment behind HTTP or HTTPS OSI level 7 proxy	12
1.3.5. Air-gapped Deployment	13
1.3.6. Required Network Ports	13
1.4. Public Cloud Requirements	20
1.4.1. Network Requirements	21
1.4.2. Prepare Storage Volumes	21
2. Installation and Deployment	23
2.1. Install SUSE Multi-Linux Manager Server	23
2.1.1. SUSE Multi-Linux Manager 5.2 Beta 2 Server Deployment	23
2.1.2. SUSE Multi-Linux Manager 5.2 Beta 2 Server Deployment as a Virtual Machine - KVM	36
2.1.3. SUSE Multi-Linux Manager 5.2 Beta 2 Server Deployment as a Virtual Machine - VMware	43
2.1.4. Server Deployment on Kubernetes	46
2.1.5. SUSE Multi-Linux Manager Server Air-gapped Deployment	53
2.1.6. Public Cloud Deployment	56
2.1.7. Connect PAYG instance	56
2.2. Install SUSE Multi-Linux Manager Proxy	60
2.2.1. SUSE Multi-Linux Manager Proxy Deployment	60
2.2.2. Convert a Client to MLM Proxy	73
2.2.3. SUSE Multi-Linux Manager Proxy Deployment as a Virtual Machine - KVM	76
2.2.4. SUSE Multi-Linux Manager Proxy Deployment as a Virtual Machine - VMware	87
2.2.5. SUSE Multi-Linux Manager 5.2 Beta 2 Proxy Deployment on Kubernetes	96
2.2.6. SUSE Multi-Linux Manager Proxy Air-gapped Deployment	104
3. Upgrade and Migration	106
3.1. Server	106
3.1.1. Distribution Upgrade and Server Migration from 5.1 to 5.2	106
3.1.2. Distribution Upgrade and Server Migration from 5.0 to 5.2	111
3.1.3. SUSE Multi-Linux Manager Server Upgrade	118
3.2. Proxy	120

3.2.1. Proxy Migration from 5.1 to 5.2	120
3.2.2. Proxy Migration from 5.0 to 5.2	127
3.2.3. SUSE Multi-Linux Manager Proxy Upgrade	134
3.3. Clients	136
3.3.1. Upgrade Clients	136
4. Basic Server and Proxy Management	137
4.1. Custom YAML Configuration and Deployment with mgradm	137
4.2. Starting and Stopping Containers	138
4.3. Containers used by SUSE Multi-Linux Manager	138
4.4. Persistent Container Volumes	139
4.4.1. Server	139
4.4.2. Proxy	141
4.5. Understanding mgr-storage-server and mgr-storage-proxy	142
4.5.1. What these tools do	142
4.5.2. What these tools do not do	143
4.5.3. Post-installation storage management	143
4.5.4. When to use, or not use	144
4.5.5. Summary	144
5. GNU Free Documentation License	145

Chapter 1. Requirements

1.1. General Requirements

Before you begin installation, ensure that you have:

1. A SUSE Customer Center account. This account gives you access to organization credentials and registration keys for SUSE Multi-Linux Manager Server, Proxy and Retail Branch Server.
2. Supported Browsers for SUSE Multi-Linux Manager Web UI.
3. SSL certificates for your environment. By default SUSE Multi-Linux Manager 5.2 Beta 2 uses a self-signed certificate.



The SL Micro 6.2 entitlement is included within the SUSE Multi-Linux Manager entitlement, so it does not require a separate registration key.

The following section contains more information on these requirements.

1.1.1. SUSE Customer Center Account and Credentials

Create an account with SUSE Customer Center prior to deployment of SUSE Multi-Linux Manager 5.2 Beta 2.

Procedure: Obtain Your Organization Credentials

1. Navigate to <https://scc.suse.com/login> in your web browser.
2. Log in to your SCC account, or follow the prompts to create a new account.
3. If you have not yet done so, click **[Connect to an Organization]** and type or search for your organization.
4. Click **[Manage my Organizations]** and select your organization from the list by clicking the organization name.
5. Click the **[Users]** tab, and then select the **[Organization Credentials]** sub-tab.
6. Record your login information for use during SUSE Multi-Linux Manager setup.

Depending on your organization's setup, you might also need to activate your subscription, using the **[Activate Subscriptions]** menu from the left navigation bar.

For more information about using SCC, see <https://scc.suse.com/docs/help>.

1.1.2. Supported Browsers for SUSE Multi-Linux Manager Web UI

To use the Web UI to manage your SUSE Multi-Linux Manager environment, you must run an up to date web browser.

SUSE Multi-Linux Manager is supported on:

- Latest Firefox browser shipped with SUSE Linux Enterprise Server
- Latest Chrome browser on all operating systems
- Latest Edge browser shipped with Windows

Windows Internet Explorer is not supported. The SUSE Multi-Linux Manager Web UI will not render correctly under Windows Internet Explorer.

1.1.3. SSL Certificates

SUSE Multi-Linux Manager uses SSL certificates to ensure that clients are registered to the correct server. By default, SUSE Multi-Linux Manager uses a self-signed certificate. If you have certificates signed by a third-party CA, you can import them to your SUSE Multi-Linux Manager installation.

- For more on self-signed certificates, see **Administration Guide › Ssl Certs Selfsigned**.
- For more on imported certificates, see **Administration Guide › Ssl Certs Imported**.

1.2. Hardware Requirements

This table outlines hardware and software requirements for the SUSE Multi-Linux Manager Server and Proxy, on x86-64, ARM, ppc64le and s390x architecture.



SUSE Multi-Linux Manager installations based on ppc64le or s390x architecture cannot use secure boot for network booting clients. This limitation exists because the shim bootloader is not available for both these architectures.

For SUSE Multi-Linux Manager for Retail hardware requirements, see **Retail Guide › Retail Requirements**.

1.2.1. Server Requirements

One of SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 is the operating system of the container host.

In the following, SUSE Linux Enterprise Server as the installed host operating system is explicitly mentioned only if it matters. Otherwise we either write SL Micro or just host operating system.

The container host with SL Micro as operating system requires as free disk space:

- Minimum for base installation 100 GB
- Plus a minimum of 130 GB for repository data

Depending on the amount of selected software, SUSE Linux Enterprise Server as operating system can require considerably more disk space.

By default the SUSE Multi-Linux Manager Server container stores mirrored repository (packages or products), database, and other data in subdirectories of the `/var/lib/containers/storage/volumes/` directory. Repository synchronization fails if this directory runs out of disk space. Estimate how much space the `/var/lib/containers/storage/volumes/` directory requires based on the number and kind of clients and repositories you plan to mirror.

For more information about filesystem and partitioning details, see **Installation and Upgrade Guide › Hardware Requirements › Install Hardware Requirements Storage** and the detailed installation instructions in the Installation and Deployment sections of this guide.

Table 1. Server Hardware Requirements

Hardware	Details	Recommendation
CPU	x86-64, ARM, ppc64le or s390x	Minimum 4 dedicated 64-bit CPU cores
RAM	Minimum	16 GB
	Recommended	32 GB
Disk Space	/ (root directory)	40 GB
	<code>/var/lib/containers/storage/volumes</code>	Minimum 150 GB (depending on the number of products)
	<code>/var/lib/containers/storage/volumes/var-pgsql</code>	Minimum 50 GB
Swap space	Systems can benefit from additional swap space. SUSE recommends using a swap file instead of a swap partition. For more information about swap space, see Installation and Upgrade Guide › Hardware Requirements › Installation Swap Space .	8 to 12 GB

The images by default have a 40 GB / partition. The cloud image of SL Micro 6.2 has just a 5 GB / partition. Both work flawlessly with SUSE Multi-Linux Manager. As long as external storage is mounted to `/var/lib/containers/storage/volumes`, SUSE Multi-Linux Manager does not need or use storage on the / partition, but leaves that to the management of the container host itself.



SUSE Multi-Linux Manager performance depends on hardware resources, network bandwidth, latency between clients and server, etc.

Based on the experience and different deployments that are in use, the advice for optimal

performance of SUSE Multi-Linux Manager Server with an adequate number of proxies is to not exceed 10,000 clients per single server. It is highly recommended to move to the Hub setup and involve consultancy when you have more than 10,000 clients. Even with fine-tuning and an adequate number of proxies, such a large number of clients can lead to performance issues.

For more information about managing a large number of clients, see **Specialized Guides › Large Deployments › Hub Multi Server**.

1.2.2. Proxy Requirements

One of SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 is the operating system of the container host.



Minimum requirements are suitable for a quick test installation, such as a Proxy with one client. If you want to use a production environment start with recommended values.

Table 2. Proxy Hardware Requirements

Hardware	Details	Recommendation
CPU	x86-64, ARM	Minimum 2 dedicated 64-bit CPU cores
	Recommended	The same as minimum values
RAM	Minimum	2 GB
	Recommended	8 GB
Disk Space	/ (root directory)	Minimum 40 GB
	/var/lib/containers/storage/volumes	Minimum 100 GB
Swap space	Systems can benefit from additional swap space. SUSE recommends using a swap file instead of a swap partition. For more information about swap space, see Installation and Upgrade Guide › Hardware Requirements › Installation Swap Space .	4 to 8 GB

By default the SUSE Multi-Linux Manager Proxy container caches packages in the /var/lib/containers/storage/volumes/uyuni-proxy-squid-cache/ directory. If there is not enough space available, the proxy will remove old, unused packages and replace them with newer packages.

As a result of this behavior:

- The larger `/var/lib/containers/storage/volumes/uyuni-proxy-squid-cache/` directory is on the proxy, the less traffic will be between the proxy and the SUSE Multi-Linux Manager Server.
- By making the `/var/lib/containers/storage/volumes/uyuni-proxy-squid-cache/` directory on the proxy the same size as `/var/lib/containers/storage/volumes/var-spacewalk/` on the SUSE Multi-Linux Manager Server, you avoid a large amount of traffic after the first synchronization.
- The `/var/lib/containers/storage/volumes/uyuni-proxy-squid-cache/` directory can be small on the SUSE Multi-Linux Manager Server compared to the proxy. For a guide to size estimation, see the [Server Requirements](#) section.



In general, SUSE recommends to adjust the value for the cache directory to about 80 % of available free space. The `cache_dir` value is set when generating proxy configuration on the server. You cannot set the option directly in `squid.conf`.

1.2.3. Swap space

Workloads differ from system to system. Systems with heavy or unpredictable workloads can benefit from additional swap space, regardless of total RAM. It is recommended to place swap on the fastest available storage (for example, SSD). SUSE recommends using a swap file instead of a swap partition.

For size recommendations, see the tables above.

The following shell command snippet creates a 8GiB swap file.



A swap file on Btrfs filesystem prevent creating of a snapshot of that volume. `/var` in the following example is already excluded from the snapper snapshots, so it is safe to use `/var/swap`.

```
## setup swapfile at /var/swap
# run following as a root user

# allocate 8GiB for swap file
fallocate -l 8G /var/swap
# ensure CoW is disabled for the swap file
chattr +C /var/swap
# allow only root access
chmod 600 /var/swap
# make swap file based on allocated file
mkswap /var/swap
# activate swap use for the running system
swapon /var/swap
# activate swap during the next boots
echo "/var/swap swap swap defaults 0 0" >> /etc/fstab
```

1.2.4. Database Requirement

PostgreSQL is the only supported database. Using a remote PostgreSQL database or remote file systems (such

as NFS) with the PostgreSQL database is not supported. In other words, PostgreSQL should be on the fastest available storage device for SUSE Multi-Linux Manager.



Because of potential performance issues, running a PostgreSQL database remotely from SUSE Multi-Linux Manager is discouraged. While such an environment is possible and even stable in many cases, there is always a risk of data loss if something goes wrong.

SUSE might not be able to provide assistance in such cases.

1.2.5. Persistent Storage and Permissions

Persistent volumes are created by default when deploying the container.

However, it is recommended that the volumes are stored on one or more separate storage devices. Such a setup helps avoid data loss in production environments. This can be done after container deployment.

Storage devices best should be set up after first deploying the container. For more details, see **Installation and Upgrade Guide › Container Management › Persistent Container Volumes**.

We recommend you use XFS as the filesystem type for all volumes. The size of the disk for repositories storage is dependent on the number of distributions and channels you intend to manage with SUSE Multi-Linux Manager. See the tables in this section for guides to estimate the size required.



Do not use NFS for Cobbler or PostgreSQL storage, neither NFS on SELinux environments. These scenarios are not supported.

On the SUSE Multi-Linux Manager Server, use this command to find all available storage devices:

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

Use the `lsblk` command to see the name and size of each device.

Use the `mgr-storage-server` command with the device names to set up the external disks as the locations for the storage and, optionally on a disk of its own, for the database:

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

For example:

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

The external storage volumes are set up as XFS partitions mounted at `/manager_storage` and `/pgsql_storage`.



This command will create the persistent storage volumes at `/var/lib/containers/storage/volumes`.

For more information, see **Installation and Upgrade Guide › Container Management › Persistent Container Volumes**.

It is possible to use the same storage device for both channel data and the database. This is not recommended, as growing channel repositories might fill up the storage, which poses a risk to database integrity. Using separate storage devices may also increase performance. If you want to use a single storage device, run `mgr-storage-server` with a single device name parameter.

If you are installing a proxy, the `mgr-storage-proxy` command takes only one device name parameter and will set up the external storage location as the Squid cache.

1.2.6. Logical Volume Management (LVM)

For all kind of virtual machines (VM), LVM is generally not needed and not recommended. The disk setup is virtual and separate disks for volumes are possible and recommended.

For other deployments, separate disks for volumes are also recommended.

On the container host of the SUSE Multi-Linux Manager Server, the `mgr-storage-server` command moves the complete content of the `/var/lib/containers/storage/volumes` directory to a separate disk and remounts it to `/var/lib/containers/storage/volumes`.

Optionally, if a second device name is specified, `mgr-storage-server` moves the content of the `/var/lib/containers/storage/volumes/var-pgsql` database directory to a second separate disk and remounts it to `/var/lib/containers/storage/volumes/var-pgsql`.

Similarly, on the container host of the SUSE Multi-Linux Manager Proxy, the `mgr-storage-proxy` command moves the complete content of the `/var/lib/containers/storage/volumes` directory to a separate disk and remounts it to `/var/lib/containers/storage/volumes`.

1.3. Network Requirements

This section details the networking and port requirements for SUSE Multi-Linux Manager.



IP forwarding will be enabled by containerized installation. This means SUSE Multi-Linux Manager 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.

Consider achieving network isolation of the SUSE Multi-Linux Manager environment according to your policies.

For more information, see <https://www.suse.com/support/kb/doc/?id=000020166>.

1.3.1. Fully Qualified Domain Name (FQDN)

The SUSE Multi-Linux Manager server must resolve its FQDN correctly. If the FQDN cannot be resolved, it can cause serious problems in a number of different components.

For more information about configuring the hostname and DNS, see <https://documentation.suse.com/sles/15-SP7/html/SLES-all/cha-network.html#sec-network-yast-change-host>.

1.3.2. Hostname and IP Address

To ensure that the SUSE Multi-Linux Manager domain name can be resolved by its clients, both server and client machines must be connected to a working DNS server. You also need to ensure that reverse lookups are correctly configured.

For more information about setting up a DNS server, see <https://documentation.suse.com/sles/15-SP7/html/SLES-all/cha-dns.html>.

1.3.3. Reenable router advertisements

When the SUSE Multi-Linux Manager is installed using `mgradm install podman` or `mgrpky 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 depending on whether:

- server and proxy are based on 15 SP7 (without Network manager)
- server and proxy are based on SL Micro 6.2 (with Network manager)

Procedure: Reenabling router advertisements without Network Manager

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. Reboot.



Network management is not working when you have no wicked.

Procedure: Reenabling router advertisements with Network Manager

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

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

3. Reboot.
4. The file should look similar to this:

```
[connection]
id=Wired connection 1
type=ethernet
interface-name=eth0

[ethernet]

[ipv4]
dns-priority=20
method=auto

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

1.3.4. 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 SUSE Multi-Linux Manager 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 separated list
server.satellite.no_proxy =
```

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

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

1.3.5. Air-gapped Deployment

If you are on an internal network and do not have access to SUSE Customer Center, you can use an **Installation and Upgrade Guide › Container Deployment › Air-gapped Deployment**.

In a production environment, the SUSE Multi-Linux Manager Server and clients should always use a firewall. For a comprehensive list of the required ports, see **Installation and Upgrade Guide › Network Requirements › Ports**.

1.3.6. Required Network Ports

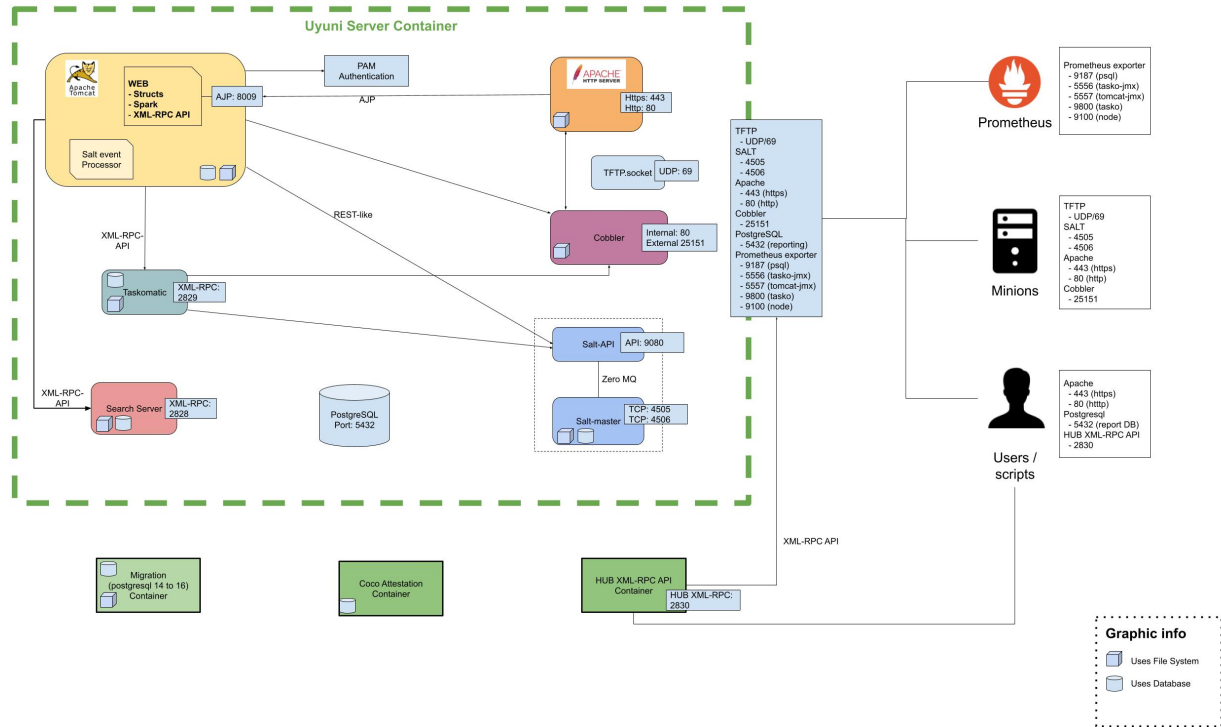
This section contains a comprehensive list of ports that are used for various communications within SUSE Multi-Linux Manager.

You will not need to open all of these ports. Some ports only need to be opened if you are using the service that

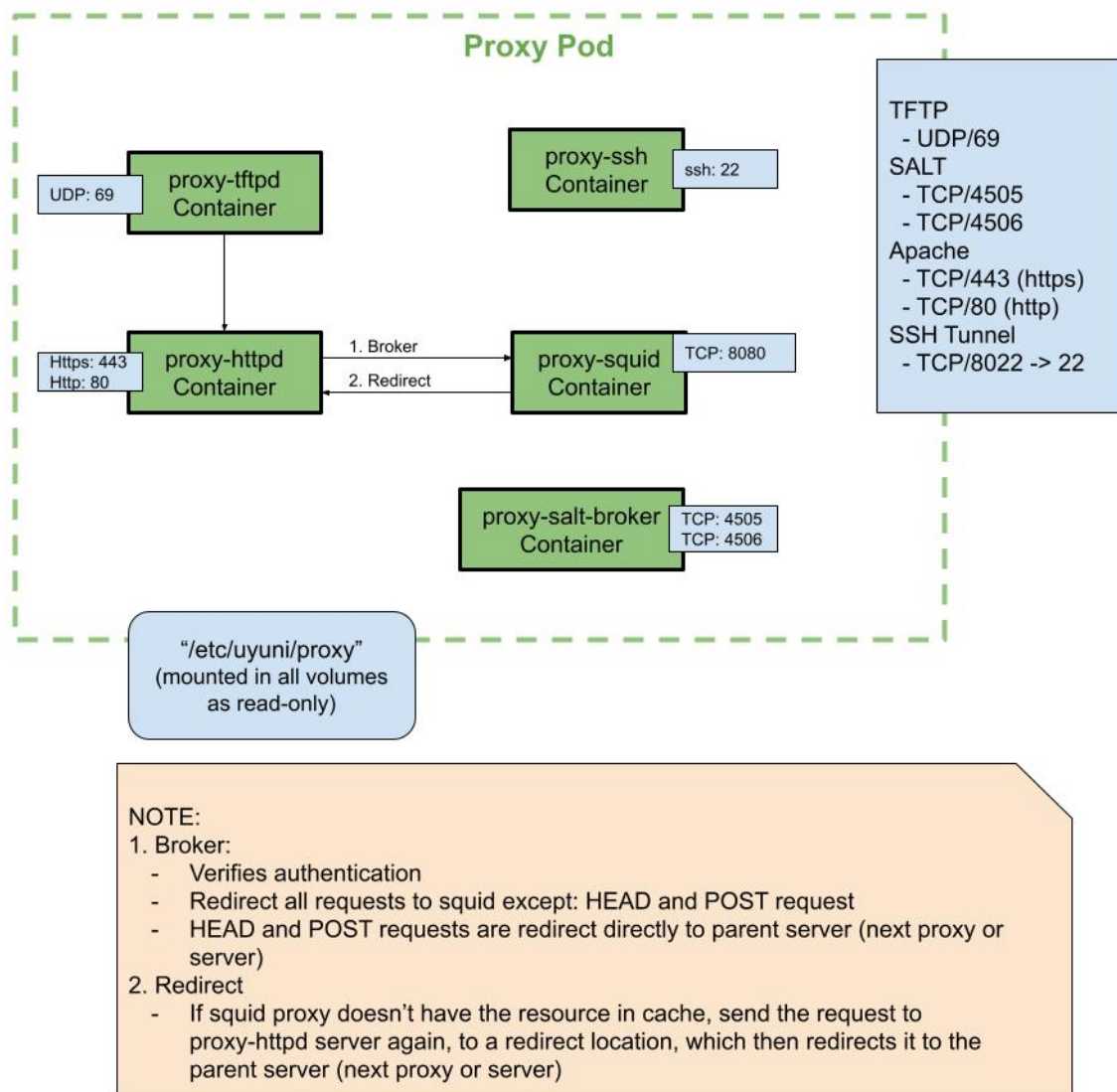
requires them.

1.3.6.1. Overview

1.3.6.1.1. Server



1.3.6.1.2. Proxy



1.3.6.2. External Inbound Server Ports

External inbound ports must be opened to configure a firewall on the SUSE Multi-Linux Manager Server to protect the server from unauthorized access.

Opening these ports allows external network traffic to access the SUSE Multi-Linux Manager Server.

Table 3. External Port Requirements for SUSE Multi-Linux Manager Server

Port number	Protocol	Used By	Notes
67	TCP/UDP	DHCP	Required only if clients are requesting IP addresses from the server.

Port number	Protocol	Used By	Notes
69	TCP/UDP	TFTP	Required if server is used as a PXE server for automated client installation.
80	TCP	HTTP	Required temporarily for some bootstrap repositories and automated installations.
443	TCP	HTTPS	Serves the Web UI, client, and server and proxy (tftpsync) 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.

1.3.6.3. External Outbound Server Ports

External outbound ports must be opened to configure a firewall on the SUSE Multi-Linux Manager Server to restrict what the server can access.

Opening these ports allows network traffic from the SUSE Multi-Linux Manager Server to communicate with external services.

Table 4. External Port Requirements for SUSE Multi-Linux Manager Server

Port number	Protocol	Used By	Notes
80	TCP	HTTP	Required for SUSE Customer Center. Port 80 is not used to serve the Web UI.
443	TCP	HTTPS	Required for SUSE Customer Center.

1.3.6.4. Internal Server Ports

Internal ports are used internally by the SUSE Multi-Linux Manager Server. Internal ports are only accessible from localhost.

In most cases, you will not need to adjust these ports.

Table 5. Internal Port Requirements for SUSE Multi-Linux Manager Server

Port number	Notes
2828	Satellite-search API, used by the RHN application in Tomcat and Taskomatic.
2829	Taskomatic API, used by the RHN application in Tomcat.
8005	Tomcat shutdown port.
8009	Tomcat to Apache HTTPD (AJP).
8080	Tomcat to Apache HTTPD (HTTP).
9080	Salt-API, used by the RHN application in Tomcat and Taskomatic.
32000	Port for a TCP connection to the Java Virtual Machine (JVM) that runs Taskomatic and satellite-search.

Port 32768 and higher are used as ephemeral ports. These are most often used to receive TCP connections. When a TCP connection request is received, the sender will choose one of these ephemeral port numbers to match the destination port.

You can use this command to find out which ports are ephemeral ports:

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

1.3.6.5. External Inbound Proxy Ports

External inbound ports must be opened to configure a firewall on the SUSE Multi-Linux Manager Proxy to protect the proxy from unauthorized access.

Opening these ports allows external network traffic to access the SUSE Multi-Linux Manager proxy.

Table 6. External Port Requirements for SUSE Multi-Linux Manager Proxy

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 (tftpsync) 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.
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.

1.3.6.6. External Outbound Proxy Ports

External outbound ports must be opened to configure a firewall on the SUSE Multi-Linux Manager Proxy to restrict what the proxy can access.

Opening these ports allows network traffic from the SUSE Multi-Linux Manager Proxy to communicate with external services.

Table 7. External Port Requirements for SUSE Multi-Linux Manager Proxy

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.

1.3.6.7. External Client Ports

External client ports must be opened to configure a firewall between the SUSE Multi-Linux Manager Server and its clients.

In most cases, you will not need to adjust these ports.

Table 8. External Port Requirements for SUSE Multi-Linux Manager Clients

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.
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.

1.3.6.8. Required URLs

There are some URLs that SUSE Multi-Linux Manager must be able to access to register clients and perform updates. In most cases, allowing access to these URLs is sufficient:

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

Additionally, you may need access to these URLs for non-SUSE products:

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

You can find additional details on whitelisting the specified URLs and their associated IP addresses in this article: [Accessing SUSE Customer Center and SUSE registry behind a firewall and/or through a proxy](#).

If you are using non-SUSE clients you might also need to allow access to other servers that provide specific packages for those operating systems. For example, if you have Ubuntu clients, you will need to be able to access the Ubuntu server.

For more information about troubleshooting firewall access for non-SUSE clients, see **Administration Guide › Troubleshooting › Tshoot Firewalls**.

1.4. Public Cloud Requirements

This section provides the requirements for installing SUSE Multi-Linux Manager on public cloud infrastructure. We have tested these instructions on Amazon EC2, Google Compute Engine, and Microsoft Azure, but they should work on other providers as well, with some variation.

Before you begin, here are some considerations:

- The SUSE Multi-Linux Manager setup procedure performs a forward-confirmed reverse DNS lookup. This must succeed in order for the setup procedure to complete and for SUSE Multi-Linux Manager to operate as expected. It is important to perform hostname and IP configuration before you set up SUSE Multi-Linux Manager.
- SUSE Multi-Linux Manager Server and Proxy instances need to run in a network configuration that provides you control over DNS entries, but cannot be accessed from the internet at large.
- Within this network configuration DNS resolution must be provided: `hostname -f` must return the fully qualified domain name (FQDN).
- DNS resolution is also important for connecting clients.
- DNS is dependent on the cloud framework you choose. Refer to the cloud provider documentation for detailed instructions.
- We recommend that you locate software repositories, the server database, and the proxy squid cache on an external virtual disk. This prevents data loss if the instance is unexpectedly terminated. This section includes instructions for setting up an external virtual disk.

1.4.1. Network Requirements

When you use SUSE Multi-Linux Manager on a public cloud, you must use a restricted network. We recommend using a VPC private subnet with an appropriate firewall setting. Only machines in your specified IP ranges must be able to access the instance.



Running SUSE Multi-Linux Manager on the public cloud means implementing robust security measures. It is essential to limit, filter, monitor, and audit access to the instance. SUSE strongly advises against a globally accessible SUSE Multi-Linux Manager instance that lacks adequate perimeter security.

To access the SUSE Multi-Linux Manager Web UI, allow HTTPS when configuring the network access controls. This allows you to access the SUSE Multi-Linux Manager Web UI.

In EC2 and Azure, create a new security group, and add inbound and outbound rules for HTTPS. In GCE, check the Allow HTTPS traffic box under the Firewall section.

1.4.2. Prepare Storage Volumes

We recommend that the repositories and the database for SUSE Multi-Linux Manager are stored on separate storage devices from the root volume. This will help to avoid data loss and possibly increase performance.

The SUSE Multi-Linux Manager container utilizes default storage locations. These locations should be configured prior to deployment for custom storage. For more information see **Installation and Upgrade Guide**

› **Container Management › Persistent container volumes**



Do not use logical volume management (LVM) for public cloud installations.

The size of the disk for repositories storage is dependent on the number of distributions and channels you intend to manage with SUSE Multi-Linux Manager. When you attach the virtual disks, they will appear in your instance as Unix device nodes. The names of the device nodes will vary depending on your provider, and the instance type selected.

Ensure the root volume of the SUSE Multi-Linux Manager Server is 100 GB or larger. Add an additional storage disk of 500 GB or more, and choose SSD storage if you can. The cloud images for SUSE Multi-Linux Manager Server use a script to assign this separate volume when your instance is launched.

When you launch your instance, you can log in to the SUSE Multi-Linux Manager Server and use this command to find all available storage devices:

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

If you are not sure which device to choose, use the `lsblk` command to see the name and size of each device.

Choose the name that matches with the size of the virtual disk you are looking for.

You can set up the external disk with the `mgr-storage-server` command. This creates an XFS partition mounted at `/manager_storage` and uses it as the location for the database and repositories:

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

For more information about setting up storage volumes and partitions, including recommended minimum sizes, see **Installation and Upgrade Guide › Hardware Requirements**.

Chapter 2. Installation and Deployment

2.1. Install SUSE Multi-Linux Manager Server

There are various scenarios to deploy a SUSE Multi-Linux Manager Server.

2.1.1. SUSE Multi-Linux Manager 5.2 Beta 2 Server Deployment

This guide shows you how to install and configure a SUSE Multi-Linux Manager 5.2 Beta 2 container on SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7.

2.1.1.1. Hardware Requirements for SUSE Multi-Linux Manager

This table shows the software and hardware requirements for deploying SUSE Multi-Linux Manager Server on your bare metal machine. For the purposes of this guide your machine should have 16 GB of RAM, and at least 200 GB of disk space. For background information about disk space, see **Installation and Upgrade Guide › Hardware Requirements**.

Table 9. Software and Hardware Requirements

Software and Hardware	Recommended
Operating System	SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7
Architecture	x86-64, ARM, s390x, ppc64le
Processor (CPU)	Minimum of four (4) 64-bit CPU cores
RAM	16 GB
Disk Space	200 GB
Channel Requirements	50 GB per SUSE or openSUSE product 360 GB per Red Hat product
Swap space:	8 to 12 GB



Supported operating system for the Server Container Host

The supported operating system for the container host is SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7.

Container host

A container host is a server equipped with a container engine like Podman, which lets it manage and deploy containers. These containers hold applications and their essential parts, such as libraries, but not a full operating system, making them lightweight. This setup ensures applications run the same way in different environments. The container host supplies the necessary resources such as CPU, memory, and storage for these containers.

Server deployment mandates the use of a fully qualified domain name (FQDN). In the absence of automatic DNS provision of an FQDN by your router or network, the deployment process will not proceed successfully. An FQDN typically follows the format <host>.<domain>.com.

For instance:

- mlm.example.com
- mlm.container.lab

For more information, see the section on network requirements in **Installation and Upgrade Guide › Network Requirements**.

2.1.1.2. Persistent Volumes

SUSE Multi-Linux Manager 5.2 Beta 2 defines the required persistent storage volumes by default. These are created during installation by the mgradm tool if they do not already exist.

These volumes are created in /var/lib/containers/storage/volumes/, where Podman stores its volumes by default.

Recommendations

You can leverage the simplicity of storage by mounting an external storage device to this directory. Because it will store the PostgreSQL database, binary packages for repositories, caches, operating system images, autoinstallation distributions, and configuration files, we have three recommendations:

Fast Storage

This mount point should ideally be NVMe or SSD-class devices. Slower storage will adversely affect SUSE Multi-Linux Manager performance.

Large Capacity

Recommended minimum size for this is at least 300 GB, and larger if there will be multiple Linux distributions or architectures to manage.

Recommended Filesystem

XFS (though any supported filesystem for SL Micro 6.2 could work).

Optional

You can provide custom storage for the volumes by mounting disks on the expected volume path inside it such as `/var/lib/containers/storage/volumes/var-spacewalk`. This adds to the complexity of a SUSE Multi-Linux Manager deployment, and may affect the resilience the default storage recommendation provides.

For a list of all persistent volumes in the container, see **Installation and Upgrade Guide › Container Management › Persistent Container Volumes**.

2.1.1.3. Prepare SUSE Multi-Linux Manager Server Host

You can deploy SUSE Multi-Linux Manager on SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7. SL Micro is a transactional system, while SUSE Linux Enterprise Server is a full server operating system.

Depending on your decision, either continue with **Installation and Upgrade Guide › Container Deployment › Server Deployment Mlm › Deploy Mlm Server Micro** or with **Installation and Upgrade Guide › Container Deployment › Server Deployment Mlm › Deploy Mlm Server Sles** and skip the not selected section.

2.1.1.3.1. Prepare SL Micro 6.2 Host

Download the installation media**Procedure: Downloading the installation media**

1. Locate the SL Micro 6.2 installation media at <https://www.suse.com/download/sle-micro/>, and download the appropriate media file.
2. Prepare a DVD or USB flash drive with the downloaded .iso image for installation.

Install SL Micro 6.2**Procedure: Installing SL Micro 6.2**

1. Insert the DVD or USB flash drive (USB disk or key) containing the installation image for SLE Micro 6.2.
2. Boot or reboot your system.

3. Use the arrow keys to select **Installation**.
4. Adjust Keyboard and language.
5. Click the checkbox to accept the license agreement.
6. Click **Next** to continue.
7. Select the registration method. For this example, we will register the server with SUSE Customer Center.



The SUSE Multi-Linux Manager 5.2 Beta 2 containers are installed as extensions. Depending on the specific extension needed from the list below, additional SUSE Customer Center registration codes will be required for each.

- SUSE Multi-Linux Manager 5.2 Beta 2 Server
- SUSE Multi-Linux Manager 5.2 Beta 2 Proxy
- SUSE Multi-Linux Manager 5.2 Beta 2 Retail Branch Server



The SL Micro 6.2 entitlement is included within the SUSE Multi-Linux Manager entitlement, so it does not require a separate registration code.

8. Enter your SUSE Customer Center email address.
9. Enter your registration code for SL Micro 6.2.
10. Click **Next** to continue.
11. To install a proxy, select the SUSE Multi-Linux Manager 5.2 Beta 2 Proxy extension; to install a server, select the SUSE Multi-Linux Manager 5.2 Beta 2 Server extension **Checkbox**.
12. Click **Next** to continue.

13. Enter your SUSE Multi-Linux Manager 5.2 Beta 2 extension registration code.
14. Click **[Next]** to continue.
15. On the NTP Configuration page click **[Next]**.
16. On the Authentication for the System page enter a password for the root user. Click **[Next]**.
17. On the Installation Settings page click **[Install]**.

This concludes installation of SL Micro 6.2 and SUSE Multi-Linux Manager 5.2 Beta 2 as an extension. For more information about preparing your machines (virtual or physical), see the [SL Micro Deployment Guide](#).

OPTIONAL: Registration from the command line

If you added SUSE Multi-Linux Manager 5.2 Beta 2 as an extension during SL Micro 6.2 installation then you can skip this procedure. However, optionally you may skip registration during SL Micro 6.2 installation by selecting the **[Skip Registration]** button. This section provides steps on registering your products after SL Micro 6.2 installation.



The following steps register a SUSE Multi-Linux Manager 5.2 Beta 2 extension with the x86-64 architecture and thus require a registration code for the x86-64 architecture. To register ARM or s390x architectures use the correct registration code.

Procedure: Registering from the command line

1. List available extensions with the following command:

```
transactional-update --quiet register --list-extensions
```

2. From the list of available extensions, select the one you wish to install:
 - a. If installing the Server, use your SUSE Multi-Linux Manager Server Extension 5.2 Beta 2 x86_64 registration code with following command:

```
transactional-update register -p Multi-Linux-Manager-Server/5.2/x86_64 -r <reg_code>
```

- b. If installing the Proxy, use your SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2 x86_64 registration code with following command:

```
transactional-update register -p Multi-Linux-Manager-Proxy/5.2/x86_64 -r
<reg_code>
```

3. Reboot.

Update the system

Procedure: Updating the system

1. Log in as **root**.
2. Run **transactional-update**:

```
transactional-update
```

3. Reboot.



SL Micro is designed to update itself automatically by default and will reboot after applying updates. However, this behavior is not desirable for the SUSE Multi-Linux Manager environment. To prevent automatic updates on your server, SUSE Multi-Linux Manager disables the transactional-update timer during the bootstrap process.

If you prefer the SL Micro default behavior, enable the timer by running the following command:

```
systemctl enable --now transactional-update.timer
```

To continue with deployment, see **Installation and Upgrade Guide › Container Deployment › Server Deployment Mlm › Deploy Mlm Server Persistent Storage**.

2.1.1.3.2. Prepare SUSE Linux Enterprise Server 15 SP7 host

Alternatively, you can deploy SUSE Multi-Linux Manager on SUSE Linux Enterprise Server 15 SP7.

The following procedures describe the main steps of the installation process.

Install SUSE Multi-Linux Manager extensions on SUSE Linux Enterprise Server**Procedure: Installing SUSE Multi-Linux Manager Extensions on SUSE Linux Enterprise Server**

1. Locate and download SUSE Linux Enterprise Server 15 SP7 .iso at <https://www.suse.com/download/sles/>.
2. Make sure that you have registration codes both for the host operating system (SUSE Linux Enterprise Server 15 SP7) and extensions
3. Start the installation of SUSE Linux Enterprise Server 15 SP7.
 - a. On the Language, keyboard and product selection select the product to install.
 - b. On the License agreement read the agreement and check I Agree to the License Terms.
4. Select the registration method. For this example, we will register the server with SUSE Customer Center.
5. Enter your SUSE Customer Center email address.
6. Enter your registration code for SUSE Linux Enterprise Server 15 SP7.
7. Click Next to continue.



Please note that for SUSE Linux Enterprise Server 15 SP7, you are required to have a valid SUSE Linux Enterprise Server subscription and corresponding registration code, which you must provide on this screen. You will be required to enter the SUSE Multi-Linux Manager Extension registration code below.

8. In the screen Extensions and Modules Selection check the following:
 - Select the SUSE Multi-Linux Manager Server Extension to install the Server, or the SUSE Multi-Linux Manager Proxy Extension to install the Proxy.

- Basesystem Module
- Containers Module

9. Click Next to continue.
10. Enter your SUSE Multi-Linux Manager 5.2 Beta 2 extension registration code.
11. Click **[Next]** to continue.
12. Complete the installation.
13. When the installation completes, log in to the newly installed server as root.
14. Update the System (optional, if the system was not set to download updates during install):

```
zypper up
```

15. Reboot.

OPTIONAL: Registration from the command line

If you added SUSE Multi-Linux Manager 5.2 Beta 2 as an extension during SUSE Linux Enterprise Server installation then you can skip this procedure.

However, optionally you may skip registration during SUSE Linux Enterprise Server installation by selecting the **[Skip Registration]** button. This section provides steps on registering your products after SUSE Linux Enterprise Server installation.



The following steps register a SUSE Multi-Linux Manager 5.2 Beta 2 extension with the x86-64 architecture and thus require a registration code for the x86-64 architecture.

To register ARM or s390x architectures use the correct registration code.

Procedure: Registering from the command line

1. List available extensions with the following command:

```
SUSEConnect --list-extensions
```

2. From the list of available extensions, select the one you wish to install.

- If installing the Server, use your SUSE Multi-Linux Manager Server Extension 5.2 Beta 2 x86_64 registration code.

For example for SUSE Linux Enterprise 15 SP7, use the following commands:

```
SUSEConnect -r <regcode>
SUSEConnect -p sle-module-containers/15.7/x86_64
SUSEConnect -p Multi-Linux-Manager-Server-SLE/5.2/x86_64 -r <regcode>
```

- If installing the Proxy, use your SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2 x86_64 registration code with the following command:

```
SUSEConnect -p Multi-Linux-Manager-Proxy-SLE/5.2/x86_64 -r <regcode>
```

Install and enable podman

Procedure: Installing podman

1. Log in as root and install the product package.

- On the server:

```
zypper in podman
zypper in -t product Multi-Linux-Manager-Server-SLE
```

- On the proxies:

```
zypper in podman
zypper in -t product Multi-Linux-Manager-Proxy-SLE
```



Before continuing, make sure that these packages are installed on the target system: * podman **server:** **mgradm, mgradm-bash-completion** proxy: mgrpxy, mgrpxy-bash-completion

2. Start the Podman service by rebooting the system, or running a command:


```
systemctl enable --now podman.service
```

To continue with deployment, see **Installation and Upgrade Guide › Container Deployment › Server Deployment Mlm › Deploy Mlm Server Persistent Storage**.

2.1.1.4. Configure Custom Persistent Storage

Configuring persistent storage is optional, but it is the only way to avoid serious trouble with container full disk conditions. It is highly recommended to configure custom persistent storage with the `mgr-storage-server` tool.

For more information, see `mgr-storage-server --help`. This tool simplifies creating the container storage and database volumes.

Use the command in the following manner:

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

For example:

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



This command will create the persistent storage volumes at `/var/lib/containers/storage/volumes`.

For more information, see

- **Installation and Upgrade Guide › Container Management › Persistent Container Volumes**
- **Administration Guide › Troubleshooting › Tshoot Container Full Disk**

2.1.1.5. Deploy SUSE Multi-Linux Manager with mgradm



If you want to use third-party SSL certificates instead of the self-signed certificates, import them in the run of the following deployment procedure.

For more information about the requirements of third-party SSL certificates, see **Administration Guide › Ssl Certs Imported**.



SUSE Multi-Linux Manager server hosts that are hardened for security may restrict execution of files from the `/tmp` folder. In such cases, as a workaround, export the `TMPDIR` environment variable to another existing path before running `mgradm`.

For example:

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

In SUSE Multi-Linux Manager updates, tools will be changed to make this workaround unnecessary.

Procedure: Deploying SUSE Multi-Linux Manager 5.2 Beta 2 Using mgradm

1. Log in as root.
2. Deploy SUSE Multi-Linux Manager.



If you use VM images as a migration target, here as the last step, execute the command `mgradm migrate` instead of `mgradm install`.

Execute one of the following commands, depending on the SSL certificate variant (self-signed or third-party). Replace `<FQDN>` with your fully qualified domain name of the SUSE Multi-Linux Manager Server:

- Using self-signed certificates provided by SUSE Multi-Linux Manager:

```
mgradm install podman <FQDN>
```

- With importing SSL certificates using third-party SSL certificate flags (the example can adjusted if not all these certificates are needed):

```
mgradm install podman <FQDN> \
--ssl-ca-intermediate <strings> \
--ssl-ca-root <string> \
--ssl-server-cert <string> \
--ssl-server-key <string> \
--ssl-db-ca-intermediate <strings> \
--ssl-db-ca-root <string> \
--ssl-db-cert <string> \
--ssl-db-key <string>
```

For more information, see `mgradm install podman --help`.



If the executed command fails ensure that you have registered SUSE Multi-Linux Manager 5.2 Beta 2. If you skipped registration during installation and now need to register from the command line, follow the steps below to log in to the registry:

```
set +o history
echo SCC_MIRRORING_PASSWORD | podman login -u "SCC_MIRRORING_USER"
--password-stdin registry.suse.com
set -o history
```

Use the SUSE Multi-Linux Manager 5.2 Beta 2 registration key when prompted.

3. Enter CA key (certificate authority) and administrator account password when prompted.



The administrator account password must be at least 5 characters and less than 48 characters in length.

4. Press **[Enter]**.
5. Enter the email address of the administration account. Press **[Enter]**.
6. Wait for deployment to complete.
7. Open a browser and proceed to your servers FQDN.
8. Enter your username (default is admin) and the password you set during the deployment process.

In this guide you deployed SUSE Multi-Linux Manager 5.2 Beta 2 Server as a container. Proceed to the next section to add your organization credentials for syncing with SUSE Customer Center.

2.1.1.6. Connect SUSE Multi-Linux Manager 5.2 Beta 2 to SUSE Customer Center

This section covers synchronizing with SCC from the Web UI and adding your first client channel.

Procedure: Entering Organization Credentials

1. Open a browser and proceed to your servers FQDN.
2. Enter your username (default is admin) and the password you set during the deployment process.
3. In the SUSE Multi-Linux Manager Web UI, select **Admin › Setup Wizard**.
4. From the Setup Wizard page select the **[Organization Credentials]** tab.
5. Click **[Add a new credential]**.
6. Point your browser to the SUSE Customer Center.
7. Select your organization from the left navigation.
8. Select the users tab from the top of the page then **[Organization Credentials]**.
9. Make a note of your **Mirroring credentials**.
10. Back in the SUSE Multi-Linux Manager Web UI enter your Username and Password, and confirm with **[Save]**.

When the credentials are confirmed with a green check-mark icon, proceed with [Procedure: Synchronizing with SUSE Customer Center](#).

Procedure: Synchronizing with SUSE Customer Center

1. In the Web UI, navigate to **Admin › Setup Wizard**.
2. From the Setup Wizard page select the SUSE Products tab. If you recently registered with SUSE Customer Center a list of products will begin populating the table. This operation could take up to a few minutes. You can monitor the progress of the operation in section on the right Refresh the product catalog from SUSE Customer Center. The table of products lists architecture, channels, and status information. For more information, see **Reference Guide › Admin › Wizard**.

Setup Wizard

HTTP Proxy Organization Credentials **SUSE Products**

Clear + Add products

Filter by product Description Filter by architecture 25 items per page

Items 1 - 25 of 94

	Product Description	Arch	Channels
<input type="checkbox"/>	Open Enterprise Server 2018	x86_64	
<input type="checkbox"/>	RHEL Expanded Support 5	i386	
<input type="checkbox"/>	RHEL Expanded Support 5	x86_64	
<input type="checkbox"/>	> RHEL Expanded Support 6	i386	
<input type="checkbox"/>	> RHEL Expanded Support 6	x86_64	
<input type="checkbox"/>	> RHEL Expanded Support 7	x86_64	
<input type="checkbox"/>	SUSE Container as a Service Platform 1.0	x86_64	
<input type="checkbox"/>	SUSE Container as a Service Platform 2.0	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 11 SP2	i586	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 11 SP2	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 11 SP3	i586	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 11 SP3	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 11 SP4	i586	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 11 SP4	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 12	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 12 SP1	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 12 SP2	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Desktop 12 SP3	x86_64	
<input checked="" type="checkbox"/>	> SUSE Linux Enterprise Desktop 15	x86_64	100%
<input type="checkbox"/>	> SUSE Linux Enterprise High Performance Computing 15	aarch64	
<input type="checkbox"/>	> SUSE Linux Enterprise High Performance Computing 15	x86_64	
<input type="checkbox"/>	> SUSE Linux Enterprise Server 10 SP3	i586	
<input type="checkbox"/>	> SUSE Linux Enterprise Server 10 SP3	ia64	
<input type="checkbox"/>	> SUSE Linux Enterprise Server 10 SP3	ppc	
<input type="checkbox"/>	> SUSE Linux Enterprise Server 10 SP3	s390x	

Page 1 of 4 First Prev Next Last

← Prev 3 of 3

Refresh the product catalog from SUSE Customer Center

- ☐ Channels
- ☐ Channel Families
- ☐ Products
- ☐ Product Channels
- ☐ Subscriptions

Refresh

Why aren't all SUSE products displayed in the list?

The products displayed on this list are directly linked to your Organization credentials (Mirror credentials) as well as your SUSE subscriptions.

If you believe there are products missing, make sure you have added the correct Organization credentials in the previous wizard step.

3. Use the **Filter by product description** and **Filter by architecture** to filter the list of displayed products. The channels listed on the **[Products]** page provide repositories for clients.
 - Add channels to SUSE Multi-Linux Manager by selecting the check box to the left of each channel. Click the arrow symbol to the left of the description to unfold a product and list available modules.
 - Click **[Add Products]** at the top of the page to start product synchronization.

After adding the channel, SUSE Multi-Linux Manager will schedule the channel to be synchronized. This can take a long time as SUSE Multi-Linux Manager will copy channel software sources from the SUSE repositories located at SUSE Customer Center to the local `/var/lib/containers/storage/volumes/var-spacewalk/` directory of your server.

When the channel is fully synchronized, a bootstrap repository for it will be automatically generated. This step is crucial for successfully bootstrapping clients, ensuring that the channel synchronization and distribution are operational on the client side. This completes the installation and configuration of SUSE Multi-Linux Manager, along with preparing the channels necessary for bootstrapping clients.

When the channel synchronization process is complete, you can proceed with registering the SUSE Multi-Linux Manager 5.2 Beta 2 Proxy or additional clients.

For more instructions, see **Client Configuration Guide › Registration Overview**.

2.1.1.7. Entering the Container for Management

To get to a shell inside the container, run on the container host:

```
mgrctl term
```

2.1.2. SUSE Multi-Linux Manager 5.2 Beta 2 Server Deployment as a Virtual Machine - KVM

This chapter provides the required Virtual Machine settings for deployment of SUSE Multi-Linux Manager 5.2 Beta 2 as an image. KVM will be combined with Virtual Machine Manager (virt-manager) as a sandbox for this installation.

2.1.2.1. Available Images



The preferred method for deploying SUSE Multi-Linux Manager 5.2 Beta 2 Server is to use one of the following available images. All tools are included in these images greatly simplifying deployment.

Images for SUSE Multi-Linux Manager 5.2 Beta 2 are available at [SUSE Multi-Linux Manager 5.2 Beta 2 VM](#)

images.



Customized SUSE Multi-Linux Manager 5.2 Beta 2 VM images are provided only for SL Micro 6.2. To run the product on SUSE Linux Enterprise Server 15 SP7, use the standard SUSE Linux Enterprise Server 15 SP7 installation media available at <https://www.suse.com/download/sles/> and enable the SUSE Multi-Linux Manager 5.2 Beta 2 extensions on top of it.

Table 10. Available Server Images

Architecture	Image Format
aarch64	qcow2, vmdk
x86_64	qcow2, vmdk, raw, Self Installer
ppc64le	raw, Self Installer
s390x *	qcow2, raw

* Two storage options are available for s390x: CDL DASD and FBA.

2.1.2.2. Virtual Machine Manager (virt-manager) Settings

Enter the following settings when creating a new virtual machine using **virt-manager**.



This table specifies the minimum requirements. These are suitable for a quick test installation, such as a server with one client. If you want to use a production environment and need background information about disk space, see **Installation and Upgrade Guide › Hardware Requirements**.

KVM Settings	
Installation Method	Import Existing Disk Image
OS:	Linux
Version:	SUSE Multi-Linux Manager-Server.x86_64-5.2 Beta 2.*.qcow2
Memory:	Minimum *)
CPU's:	Minimum *)
Storage Format:	.qcow2 40 GB (Default) Root Partition
Name:	test-setup

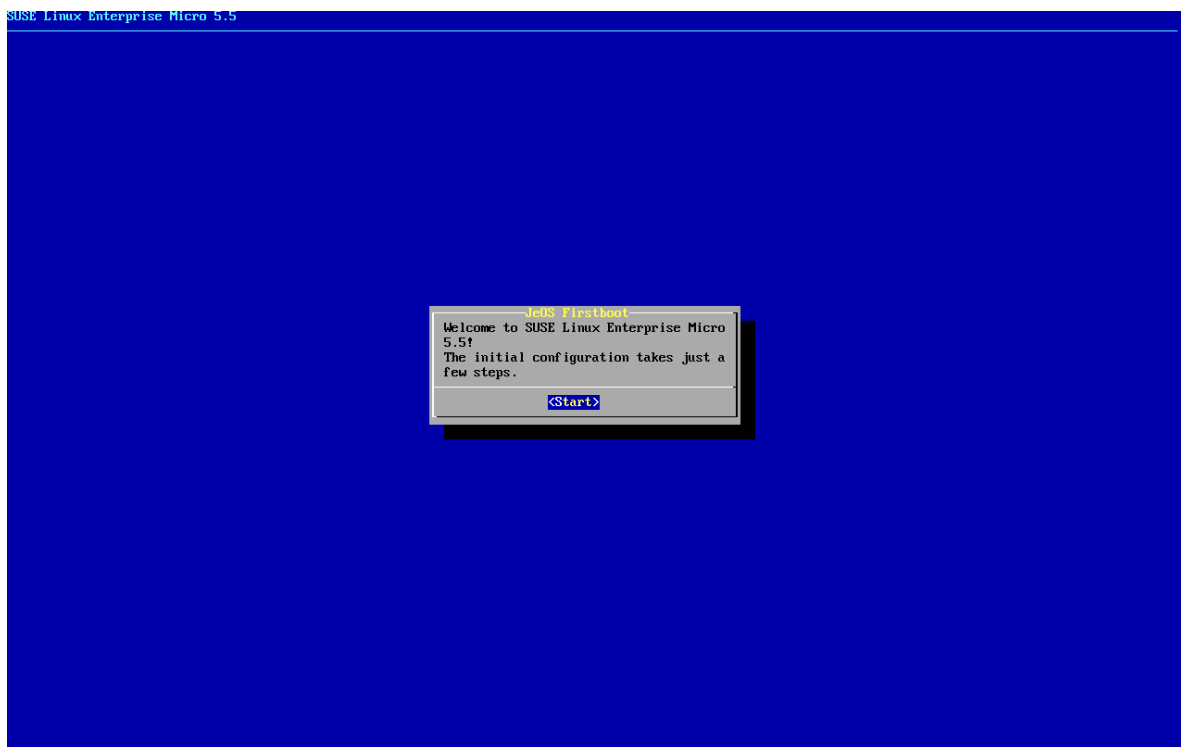
KVM Settings	
Network	Bridge br0

*) For minimum values, see **Installation and Upgrade Guide › Hardware Requirements › Server Hardware Requirements**.

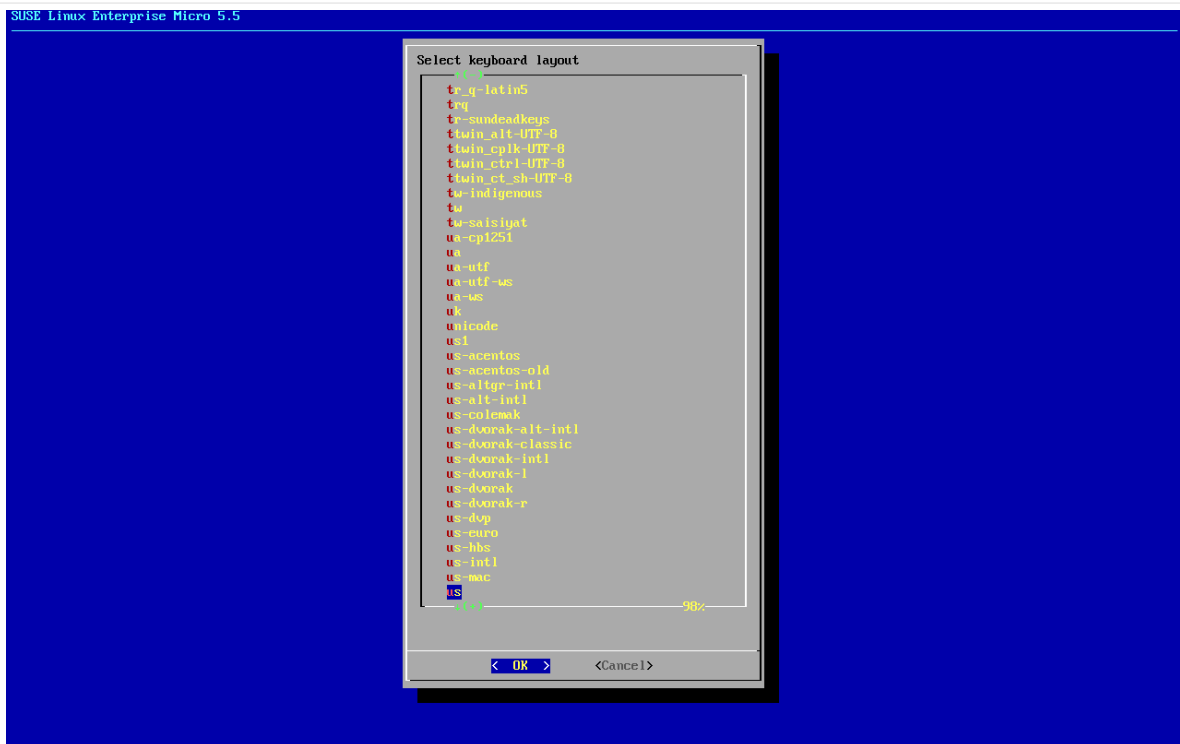
2.1.2.3. Initial KVM Setup

Procedure: Creating Initial Setup

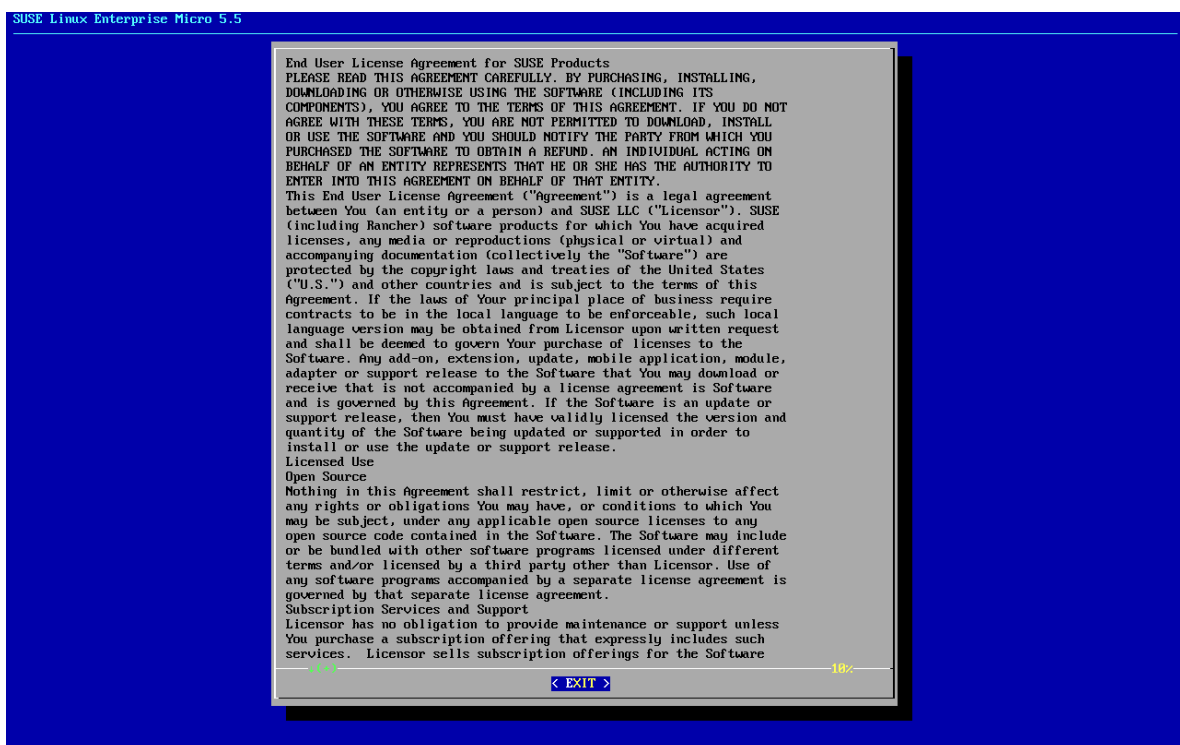
1. Create a new virtual machine using the downloaded Minimal KVM image and select Import existing disk image.
2. Configure RAM and number of CPUs.
3. Name your KVM machine.
4. Click **[Begin Installation]** to boot from the image.
5. At the JeOS Firstboot screen select start to continue.



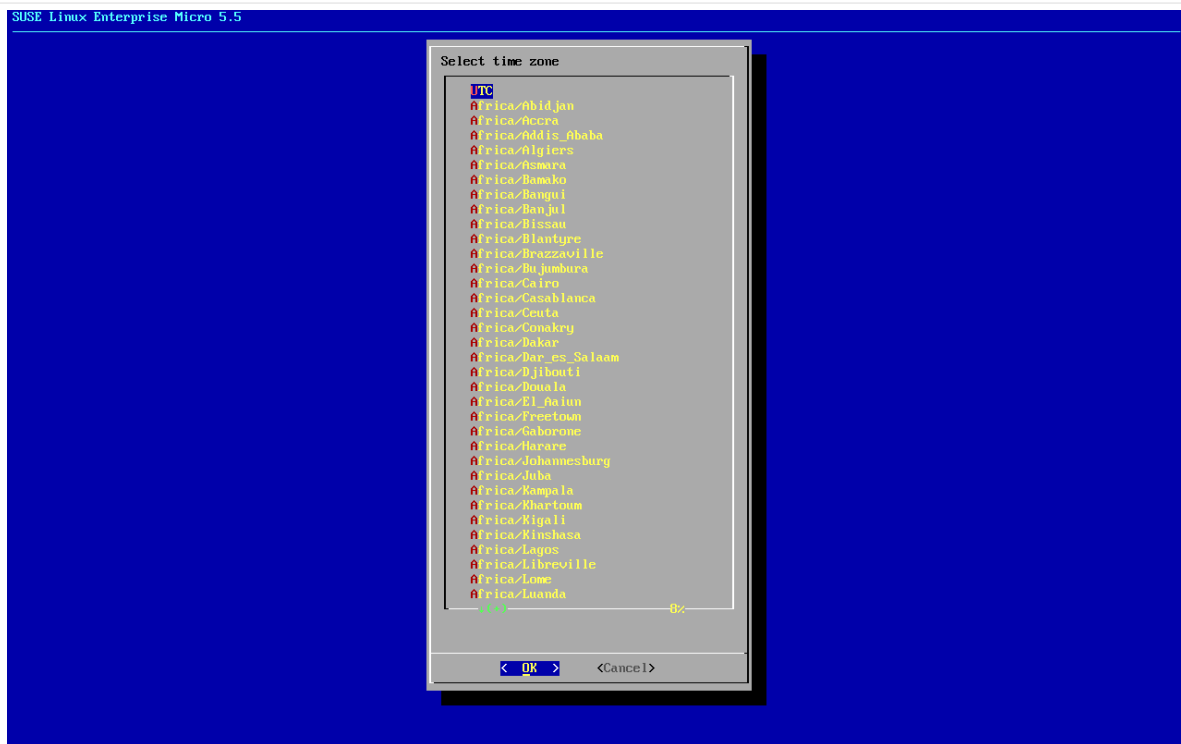
6. Select keyboard layout.



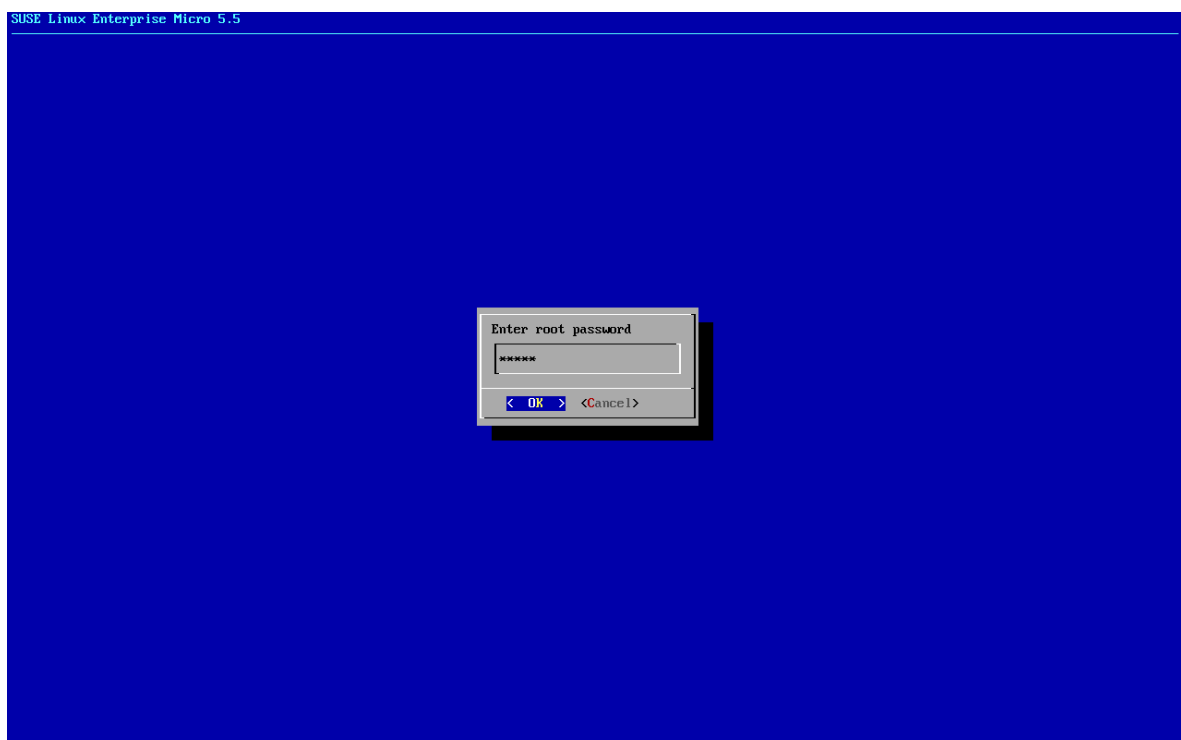
7. Accept the license agreement.



8. Select your time zone.



9. Enter a password for root.



10. When installation completes log in as root.
11. Proceed to the next section.

2.1.2.4. Register SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2 Server



The SL Micro 6.2 entitlement is included within the SUSE Multi-Linux Manager entitlement, so it does not require a separate registration code.



SUSE Multi-Linux Manager server hosts that are hardened for security may restrict execution of files from the /tmp folder. In such cases, as a workaround, export the TMPDIR environment variable to another existing path before running mgradm.

For example:

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

In SUSE Multi-Linux Manager updates, tools will be changed to make this workaround unnecessary.

Procedure: Registering SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2

1. Boot the virtual machine.
2. Log in as root.
3. Register SL Micro with SCC.

```
transactional-update register -r <REGCODE> -e <your_email>
```

4. Reboot.
5. Register SUSE Multi-Linux Manager 5.2 Beta 2 with SUSE Customer Center.

```
transactional-update register -p Multi-Linux-Manager-Server/5.2/x86_64 -r <REGCODE>
```

6. Reboot.
7. Update the system:

```
transactional-update
```

8. If updates were applied reboot.
9. This step is optional. However, if custom persistent storage is required for your infrastructure, use the mgr-storage-server tool.

For more information, see `mgr-storage-server --help`. This tool simplifies creating the container storage and database volumes. Use the command in the following manner:

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

For example:

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



This command will move the persistent storage volumes at /var/lib/containers/storage/volumes to specified storage devices.

For more information, see

- **Installation and Upgrade Guide › Container Management › Persistent Container Volumes**
- **Administration Guide › Troubleshooting › Tshoot Container Full Disk**

10. Deploy SUSE Multi-Linux Manager.



If you use VM images as a migration target, here as the last step, execute the command `mgradm migrate` instead of `mgradm install`.

Execute one of the following commands, depending on the SSL certificate variant (self-signed or third-party). Replace <FQDN> with your fully qualified domain name of the SUSE Multi-Linux Manager Server:

- Using self-signed certificates provided by SUSE Multi-Linux Manager:

```
mgradm install podman <FQDN>
```

- With importing SSL certificates using third-party SSL certificate flags (the example can adjusted if not all these certificates are needed):

```
mgradm install podman <FQDN> \
  --ssl-ca-intermediate <strings> \
  --ssl-ca-root <string> \
  --ssl-server-cert <string> \
  --ssl-server-key <string> \
  --ssl-db-ca-intermediate <strings> \
  --ssl-db-ca-root <string> \
  --ssl-db-cert <string> \
  --ssl-db-key <string>
```

For more information, see `mgradm install podman --help`.

2.1.3. SUSE Multi-Linux Manager 5.2 Beta 2 Server Deployment as a Virtual Machine - VMware

This chapter provides the required Virtual Machine settings for deployment of SUSE Multi-Linux Manager 5.2 Beta 2 as an Image. VMware will be used as a sandbox for this installation.

2.1.3.1. Available Images



The preferred method for deploying SUSE Multi-Linux Manager 5.2 Beta 2 Server is to use one of the following available images. All tools are included in these images greatly simplifying deployment.

Images for SUSE Multi-Linux Manager 5.2 Beta 2 are available at [SUSE Multi-Linux Manager 5.2 Beta 2 VM images](#).



Customized SUSE Multi-Linux Manager 5.2 Beta 2 VM images are provided only for SL Micro 6.2. To run the product on SUSE Linux Enterprise Server 15 SP7, use the standard SUSE Linux Enterprise Server 15 SP7 installation media available at <https://www.suse.com/download/sles/> and enable the SUSE Multi-Linux Manager 5.2 Beta 2 extensions on top of it.



For more information on preparing raw images, see <https://documentation.suse.com/sle-micro/6.1/html/Micro-deployment-raw-images-virtual-machines/index.html#deployment-preparing-configuration-device>.

For additional information on the self install images, see <https://documentation.suse.com/sle-micro/6.1/html/Micro-deployment-selfinstall-images/index.html>

Table 11. Available Server Images

Architecture	Image Format
aarch64	qcow2, vmdk
x86_64	qcow2, vmdk, raw, Self Installer
ppc64le	raw, Self Installer
s390x *	qcow2, raw

* Two storage options are available for s390x: CDL DASD and FBA.

2.1.3.2. SUSE Multi-Linux Manager Virtual Machine Settings - VMware

This section describes VMware configurations, focusing on the creation of an extra virtual disk essential for the SUSE Multi-Linux Manager storage partition within VMware environments.

Procedure: Creating the VMware Virtual Machine

1. Download SUSE Multi-Linux Manager Server .vmdk file then transfer a copy to your VMware storage.
2. Make a copy of uploaded .vmdk file using VMware web interface. This will convert provided .vmdk file to the format suitable for vSphere hypervisor.
3. Create and name a new virtual machine based on the Guest OS Family Linux and Guest OS Version SUSE Linux Enterprise 15 (64-bit).
4. Add an additional Hard Disk 2 of 500 GB (or more).
5. Configure RAM and number of CPUs with minimum values. *)
6. Set the network adapter as required.
7. Power on the VM, and follow firstboot dialogs (keyboard layout, license agreement, time zone, password for root).
8. When installation completes log in as root.
9. Proceed to the next section.

*) For minimum values, see **Installation and Upgrade Guide › Hardware Requirements › Proxy Hardware Requirements**.

2.1.3.3. Register SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2 Server

Before starting obtain your SUSE Multi-Linux Manager Registration Code from SUSE Customer Center - <https://scc.suse.com>.



The SL Micro 6.2 entitlement is included within the SUSE Multi-Linux Manager entitlement, so it does not require a separate registration code.



SUSE Multi-Linux Manager server hosts that are hardened for security may restrict execution of files from the /tmp folder. In such cases, as a workaround, export the TMPDIR environment variable to another existing path before running mgradm.

For example:

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

In SUSE Multi-Linux Manager updates, tools will be changed to make this workaround unnecessary.

Procedure: Registering SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2

1. Boot the virtual machine.
2. Log in as root.
3. Register SL Micro with SCC.

```
transactional-update register -r <REGCODE> -e <your_email>
```

4. Reboot.
5. Register SUSE Multi-Linux Manager 5.2 Beta 2 with SUSE Customer Center.

```
transactional-update register -p Multi-Linux-Manager-Server/5.2/x86_64 -r <REGCODE>
```

6. Reboot
7. Update the system:

```
transactional-update
```

8. If updates were applied reboot.
9. This step is optional. However, if custom persistent storage is required for your infrastructure, use the `mgr-storage-server` tool.

For more information, see `mgr-storage-server --help`. This tool simplifies creating the container storage and database volumes. Use the command in the following manner:

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

For example:

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



This command will create the persistent storage volumes at `/var/lib/containers/storage/volumes`.

For more information, see

- [Installation and Upgrade Guide › Container Management › Persistent Container Volumes](#)
- [Administration Guide › Troubleshooting › Tshoot Container Full Disk](#)

10. Deploy SUSE Multi-Linux Manager.



If you use VM images as a migration target, here as the last step, execute the command `mgradm migrate` instead of `mgradm install`.

Execute one of the following commands, depending on the SSL certificate variant (self-signed or third-party). Replace `<FQDN>` with your fully qualified domain name of the SUSE Multi-Linux Manager Server:

- Using self-signed certificates provided by SUSE Multi-Linux Manager:

```
mgradm install podman <FQDN>
```

- With importing SSL certificates using third-party SSL certificate flags (the example can adjusted if not all these certificates are needed):

```
mgradm install podman <FQDN> \
--ssl-ca-intermediate <strings> \
--ssl-ca-root <string> \
--ssl-server-cert <string> \
--ssl-server-key <string> \
--ssl-db-ca-intermediate <strings> \
--ssl-db-ca-root <string> \
--ssl-db-cert <string> \
--ssl-db-key <string>
```

For more information, see `mgradm install podman --help`.

2.1.4. Server Deployment on Kubernetes

SUSE Multi-Linux Manager can also be deployed on Kubernetes. This guide shows you how to install and configure a SUSE Multi-Linux Manager 5.2 Beta 2 on Kubernetes on RKE2.

Several personas are involved in the installation of SUSE Multi-Linux Manager on Kubernetes:

- the Kubernetes administrator manages the cluster, its users and accesses,
- the infrastructure administrator takes care of wiring the network access to the cluster,
- the PKI administrator is responsible for the TLS certificates generation and deployment infrastructure,
- the SUSE Multi-Linux Manager administrator controls the application itself and its deployment.

In some cases, some of those personas can be merged into a single person or team, but keeping then in mind

will explain why the installation is not a one-shot script doing everything. Kubernetes clusters can also vary a lot between organizations, so the SUSE Multi-Linux Manager core installation is designed to be as agnostic as possible of those specific cases.

2.1.4.1. Prerequisites

Installing the Kubernetes cluster and configuring it is out of the scope of this document.

The cluster is assumed to be ready to be used with a user having rights on a namespace dedicated to SUSE Multi-Linux Manager.

If there isn't one defined yet, a Role and RoleBinding with minimum rights required to deploy server-helm would look like the following:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: example-resource-manager
  namespace: $NAMESPACE
rules:
- apiGroups: [""]
  resources: ["pods", "pods/log", "services", "secrets", "configmaps",
"persistentvolumeclaims"]
  verbs: ["*"]
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["*"]
- apiGroups: ["networking.k8s.io"]
  resources: ["ingresses"]
  verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: example-resource-manager-binding
  namespace: $NAMESPACE
subjects:
- kind: User
  name: $USERNAME
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: example-resource-manager
  apiGroup: rbac.authorization.k8s.io
```



This guide assumes the reader knows how to work with Kubernetes: the concepts will not be explained here as they are extensively documented in the official Kubernetes documentation.

The SUSE Multi-Linux Manager administrator needs to deploy the server-helm Helm chart. However, this chart requires to prepare:

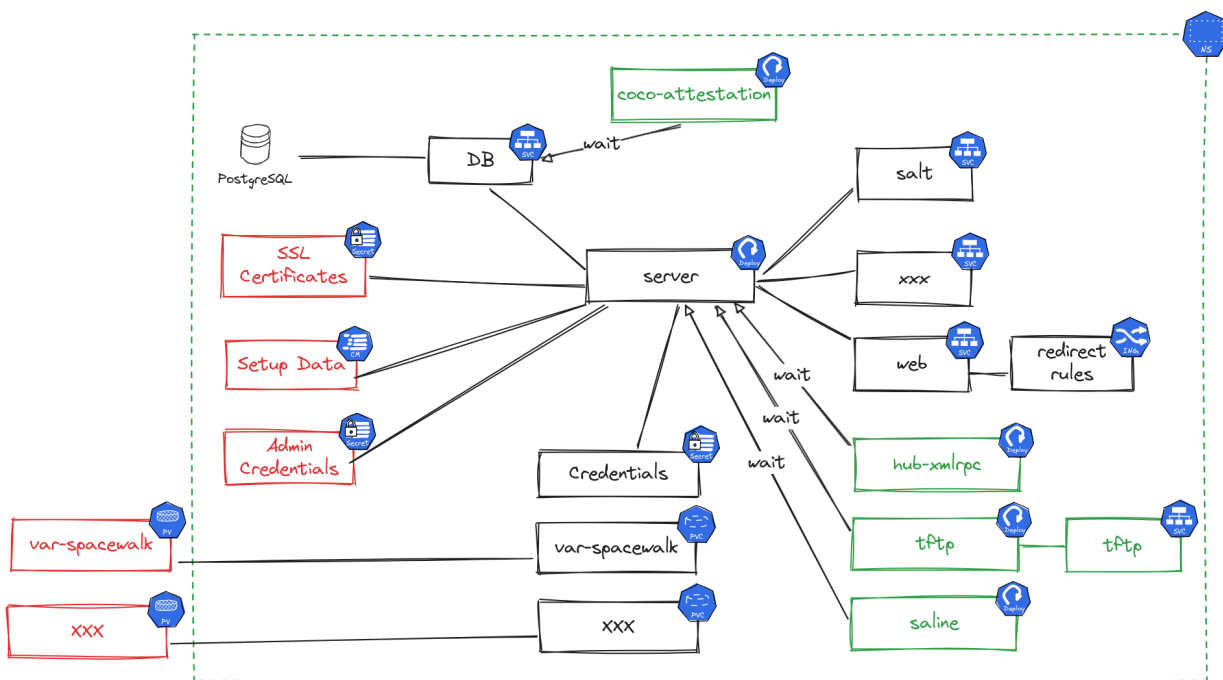
- TLS certificates chain for the server and database,
- ConfigMaps for the server and database root CA certificates,
- persistent volumes for the claims the chart will create or a storage class automatically creating them,
- credentials secrets for the database users and the administrator,
- Load balancers or other mechanisms to expose the Salt, report database and optionally TFTP ports.

Run the following command to read the full details on how to use the server Helm chart:

```
helm show readme --version 5.2.0-beta2 \
oci://registry.suse.com/suse/multi-linux-manager/5.2/server-helm
```

2.1.4.1.1. Global architecture

This diagram shows the components deployed by the server-helm chart and the ones expected to be created before hand.



- Red items are required,
- Green items are optional and can be enabled using the chart values,
- Black components are the core components.



Even if there are values to disable the internal database, using an external one is not supported yet. Those properties are only present for testing purpose.

The next sections will explain the resources that are expected.

2.1.4.1.2. Credentials

The deployment requires four specific secrets of `kubernetes.io/basic-auth` type, each containing a username and a password key. Here are example commands to create them, set the `NAMESPACE` variable to the namespace where SUSE Multi-Linux Manager will be installed. Adjust these commands to set actual passwords.



Using `--from-literal` for the passwords is not a secure practice, read the command help to use `--from-file` or `--from-env-file`. If using declarative YAML definitions instead of these commands, make sure these files are encrypted before pushing them to a remote version control.

```
kubectl create secret generic -n $NAMESPACE --type 'kubernetes.io/basic-auth' \
  --from-literal=username=dbadmin \
  --from-literal=password=supersecret \
  db-admin-credentials
kubectl create secret generic -n $NAMESPACE --type 'kubernetes.io/basic-auth' \
  --from-literal=username=dbuser \
  --from-literal=password=supersecret \
  db-credentials
kubectl create secret generic -n $NAMESPACE --type 'kubernetes.io/basic-auth' \
  --from-literal=username=reportdb \
  --from-literal=password=supersecret \
  reportdb-credentials
kubectl create secret generic -n $NAMESPACE --type 'kubernetes.io/basic-auth' \
  --from-literal=username=admin \
  --from-literal=password=supersecret \
  admin-credentials
```

A secret with the SCC credentials needs to be defined in order to pull the images from `registry.suse.com`. Refer to <https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/> for the instructions to prepare the secret. Set the `registrySecret` server-helm chart value to the name of the secret containing those credentials to use it.

2.1.4.1.3. TLS setup

The following TLS secrets are expected:

- `db-cert`: TLS certificate for the report database and needs to have the `db` and `reportdb` Subject Alternate Names, and the FQDN exposed to the outside world.
- `uyuni-cert`: TLS certificate for the Ingress rule and needs to have the public FQDN as Subject Alternate Name.

These secrets can be created using the `kubectl create secret tls -n $NAMESPACE` command. The certificate file passed to this command needs to start with the server certificate followed by the chain of intermediary CA certificates if any. The root CAs are not needed in these secrets as they are expected in ConfigMaps.

The Root CA certificate of `db-cert` and `uyuni-cert` are expected in ConfigMaps named `db-ca` and `uyuni-ca`

stored in the ca.crt key. Those can be created with a command like `kubectl create cm -n $NAMESPACE db-ca --from-file=ca.crt=/path/to/db-ca.crt`.

2.1.4.1.4. Storage

The server chart defines volumes as Persistent Volume Claims (PVCs).



- The creation of the underlying PVs is the responsibility of the cluster administrators.
- The PVCs use the ReadWriteOnce access mode.

The created PVCs can be tuned using Helm chart values. Each of the PVCs can have the following values:

- `size`: to set the requested size of the PVC.
- `storageClass`: can be used to select the storage class to use for the PVC. This can be useful to select faster storage for the database or the packages storage.
- `extraLabels`: can be used to add custom labels to the PVC.
- `annotations`: can be used to set custom annotations on the PVC.
- `volumeName`: can be used to hard code which volume the PVC should be bound to.
- `selector`: is the YAML fragment of the PVC selector to use to find the PV to bind to.

Refer to <https://kubernetes.io/docs/concepts/storage/persistent-volumes/> for more information on persistent volumes and their claims.

Refer to the server-helm README for the list of persistent volume claims which will be created and will need to be bound to persistent volumes.



While the default sizes are provided, it is highly recommended to change them based on the distributions you plan to synchronize.

For more information on storage requirements see **Installation and Upgrade Guide › General Requirements**.

2.1.4.1.5. Exposing ports

SUSE Multi-Linux Manager requires some TCP and UDP ports to be routed to its services. Refer to the server-helm README for the list of ports to be exposed.



RKE2 ships with nginx as the default ingress controller. However, as this is deprecated and soon to be unsupported, the server-helm chart defaults to use Traefik as ingress controller. Using the nginx ingress controller might work and will not be documented, use at your own risk.



The server-helm chart supports Gateway API version 1.4. Since this requires experimental CRDs which are not shipped with RKE2 1.35, it is not recommended to be used in production.

There are multiple ways to expose the ports, but this documentation will only mention how to configure RKE2's Traefik for this. This is not a task for the SUSE Multi-Linux Manager administrator, but the Kubernetes cluster administrator as it requires configuration to be set on the cluster nodes.

To set Traefik to expose and route the needed ports, create a `/var/lib/rancher/rke2/server/manifests/uyuni-traefik.yaml` on each node with the following content. Note that Traefik takes a few seconds to be reinstalled after saving the file.

```
apiVersion: helm.cattle.io/v1
kind: HelmChartConfig
metadata:
  name: rke2-traefik
  namespace: kube-system
spec:
  valuesContent: |-
    ports:
      reportdb-pgsql:
        port: 5432
        expose:
          default: true
          exposedPort: 5432
          protocol: TCP
          hostPort: 5432
          containerPort: 5432
      salt-publish:
        port: 4505
        expose:
          default: true
          exposedPort: 4505
          protocol: TCP
          hostPort: 4505
          containerPort: 4505
      salt-request:
        port: 4506
        expose:
          default: true
          exposedPort: 4506
          protocol: TCP
          hostPort: 4506
          containerPort: 4506
```

When using the server as a TFTP server there are a few issues to consider. TFTP is complex to expose from a Kubernetes pod due to the nature of the protocol: the TFTP server receives requests on port 69, but negotiates another random port to continue. This port also needs to stay the same through the whole session for the server to recognize the client as being the same. This means that there are only two possible ways to use the TFTP server:

- using a load balancer compatible with TFTP,

- using the host network for the TFTP pod. This can be achieved by setting the `tftp.hostNetwork` helm chart value to `true`.

If Traefik is used as the ingress controller, the user needs access to additional resources. Add the following to the rules of the previously defined role:

```
- apiGroups: ["traefik.io", "traefik.containo.us"]
  resources: ["ingressroutetcps", "middlewares"]
  verbs: ["*"]
```

If Gateway API is used instead, add the following to the rules of the previously defined role:

```
- apiGroups: ["gateway.networking.k8s.io"]
  resources: ["gateways", "httproutes", "tcproutes"]
  verbs: ["*"]
```

2.1.4.1.6. Security framework setup

The main container of the server runs `systemd` even though this is not a cloud native practice. This means that the default Kubernetes security profiles for this container are not sufficient. There are two possibilities to solve this issue:

- run the main container as super privileged by setting the `server.superPrivileged` helm value to `true`,
- add custom policies or profile and use it for the container.

Configuring SELinux or AppArmor is documented in the `server-helm` chart README, report to it for more details.



Keep in mind, that running the container as super privileged is not the safest thing to do.

2.1.4.2. Installation

The `server-helm` chart requires one value to be set: `global.fqdn`. The other values have sensible defaults, report to the `server-helm` chart README for more details on those.

The server can be installed with a command like the following:

```
helm install smlm-server \
  oci://registry.suse.com/suse/multi-linux-manager/5.2/server-helm \
  -n $NAMESPACE \
  --description "Server installation" \
  --set "global.fqdn=the.server.fqdn" \
  --set "registrySecret=the-scc-secret" \
  --version 5.2.0-beta2
```

When setting multiple values, using a YAML values file is recommended instead of passing several `--set`

parameters. Refer to the `helm` command help for more details.

As the main container is not fully cloud-native yet, it is sometimes useful to get a terminal inside it. This can be achieved using the following commands. Variables have been used here for the sake of readability, but it could fit in a single line for convenience. This could be used in place of `mgrctl` term through the other pages of the documentation.

```

NAMESPACE=`kubectl get pod -A -lapp.kubernetes.io/part
-of=uyuni,app.kubernetes.io/component=server -o "jsonpath={.items[0].metadata.namespace}"`
POD=`kubectl get pod -A -lapp.kubernetes.io/part
-of=uyuni,app.kubernetes.io/component=server -o "jsonpath={.items[0].metadata.name}"`
kubectl exec -ti -n $NAMESPACE $POD -- bash

```

2.1.4.3. Example helm charts

Some helm charts using the `server-helm` chart can be found in the `Manager-5.2` branch of the [uyuni-charts](#) git repository. They show case how the TLS certificates can be generated using `cert-manager` and `trust-manager`. Those examples may assume to have Kubernetes cluster administrator permissions.



These examples are not supported, only provided for documentation purpose.

2.1.4.4. Diagnostics / Troubleshooting

- Troubleshooting:
 - Check the pods status and errors using `kubectl get pod -n $NAMESPACE` and `kubectl describe pod -n $NAMESPACE <POD NAME>`.
 - If the pods are ready, but Salt minions cannot access the master, check the network configuration and how the ports are exposed.

2.1.5. SUSE Multi-Linux Manager Server Air-gapped Deployment

2.1.5.1. What is Air-gapped deployment?

Air-gapped deployment refers to the setup and operation of any networked system that is physically isolated from insecure networks, especially the internet. This type of deployment is commonly used in high-security environments such as military installations, financial systems, critical infrastructure, and anywhere sensitive data is handled and must be protected from external threats.

2.1.5.2. Deployments

SUSE Multi-Linux Manager supports two deployment variants.

2.1.5.2.1. Deploy with Virtual Machine

The recommended installation method is using the provided SUSE Multi-Linux Manager Virtual Machine Image option, since all the needed tools and container images are pre-loaded and will work out of the box.

For more information about installing SUSE Multi-Linux Manager Server Virtual Machine, see [Deploy Server as a Virtual Machine](#).

To upgrade SUSE Multi-Linux Manager Server, users should upgrade all packages in the system and follow the procedures defined in [Server Upgrade](#).

2.1.5.2.2. Deploy SUSE Multi-Linux Manager on SL Micro

SUSE Multi-Linux Manager also provides all the needed container images in RPM's that can be installed on the system.



User should make the needed RPM available on the internal network. That can be done by using a second SUSE Multi-Linux Manager Server or an RMT server.

Procedure: Install SUSE Multi-Linux Manager on SL Micro in air-gapped deployment

1. Install SL Micro.
2. Update the system.
3. Install tools packages and image packages (replace \$ARCH\$ with the correct architecture).

```
transactional-update pkg install mgradm* mgrecl* suse-multi-linux-manager-5.2-  
$ARCH$-server-*
```

4. Reboot.
5. Deploy SUSE Multi-Linux Manager with mgradm.

For more detailed information about installing SUSE Multi-Linux Manager Server on SL Micro, see [Deploy Server as a Virtual Machine](#).

To upgrade SUSE Multi-Linux Manager Server, users should upgrade all packages in the system and follow the procedures defined in [Server Upgrade](#).

2.1.5.3. PTFs

The PTF images are not available as packages. This means that they should be pulled using podman on a machine with internet access, then saved in an archive, transferred to the air-gapped machine and loaded there.

Procedure: Pulling the image on a machine with internet access

1. Install podman.
2. Authenticate against the SUSE Registry using the SCC credentials:

```
set +o history
echo SCC_MIRRORING_PASSWORD | podman login -u "SCC_MIRRORING_USER" --password
-stdin registry.suse.com
set -o history
```

3. Run the following commands in a shell.



This example assumes that the PTF is for the server image, adjust the example if the PTF concerns another image.

```
SCC_USERID=aXXXX
PTFID=12345
echo "registry.suse.com/a/$SCC_USERID/$PTFID/suse/multi-linux-
manager/5.2/x86_64/server:latest-ptf-$PTFID" >>/tmp/ptf-images
```

4. Verify that the file contains the full path to the PTF image. If needed, other images full paths can be added to the file, one per line.
5. Pull each of the container images of the PTF and save them in a tar archive by running the following commands.

```
for image in `cat /tmp/ptf-images`; do
    podman pull $image
done
podman save -o /tmp/ptf-images.tar $(cat /tmp/ptf-images)
```

6. Transfer the /tmp/ptf-images.tar images archive on the server to patch.

Procedure: Loading the images on the server to patch

1. Ensure the ptf-images.tar file is available on the server.

2. Load the images from the archive:

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

3. Install the PTF as it would be done on a connected machine, using command:

```
mgradm support ptf podman [--test|--ptf] NUMBER --user SCC_ACCOUNT
```

Because the images are already loaded they will not be pulled.

2.1.6. Public Cloud Deployment

Public clouds provide SUSE Multi-Linux Manager under a Bring-your-own-subscription (BYOS) or Pay-as-you-go (PAYG) models.

For more information about using SUSE Multi-Linux Manager in the public cloud, see **Specialized Guides › Public Cloud Guide › Public Cloud Guide**.

2.1.7. Connect PAYG instance

In the major public cloud providers (AWS, Azure), SUSE:

- provides customized PAYG product images for SLES, SLES for SAP, etc.
- operates per-region RMT Servers mirroring repositories for products available as PAYG

This document describes how to connect existing PAYG instance to SUSE Multi-Linux Manager server, and gives basic information about credentials collection from the instance. The goal of this connection is to extract authentication data so the SUSE Multi-Linux Manager Server can connect to a cloud RMT host. Then the SUSE Multi-Linux Manager Server has access to products on the RMT host that are not already available with the SCC organization credentials.

Before using PAYG feature make sure that:

- The PAYG instance is launched from the correct SUSE product image (for example, SLES, SLES for SAP, or SLE HPC) to allow access to the desired repositories
- SUSE Multi-Linux Manager Server has connectivity to the PAYG instance (ideally in the same region) either directly or via a bastion
- A basic SCC account is required. Enter your valid SCC credentials in **Admin › Setup Wizard › Organization Credentials**. This account is required for accessing the SUSE Multi-Linux Manager client tools for

bootstrapping regardless of PAYG instances.

- If you bootstrap the PAYG instance to SUSE Multi-Linux Manager, it will disable its PAYG repositories then add repositories from where it mirrored the data from the RMT server. The final result will be PAYG instances acquiring the same repositories from the RMT servers but through the SUSE Multi-Linux Manager server itself. Of course repositories can still be setup primarily from SCC.

2.1.7.1. Connecting PAYG instance

Procedure: Connecting new PAYG instance

1. In the SUSE Multi-Linux Manager Web UI, navigate to **Admin › Setup Wizard › PAYG**, and click **[Add PAYG]**.
2. Start with the page section PAYG connection Description.
3. In the Description field, add the description.
4. Move to the page section Instance SSH connection data.
5. In the Host field, enter the instance DNS or IP address to connect from SUSE Multi-Linux Manager.
6. In the SSH Port field, enter the port number or use default value 22.
7. In the User field, enter the username as specified in the cloud.
8. In the Password field, enter the password.
9. In the SSH Private Key field, enter the instance key.
10. In the SSH Private Key Passphrase field, enter the key passphrase.



⋮ Authentication keys must always be in PEM format.

If you are not connecting directly to the instance, but via SSH bastion, proceed with [Procedure: Adding SSH bastion connection data](#).

Otherwise, continue with [Procedure: Finishing PAYG connecting](#).

Procedure: Adding SSH bastion connection data

1. Navigate to the page section Bastion SSH connection data.
2. In the Host field, enter the bastion hostname.
3. In the SSH Port field, enter the bastion port number.
4. In the User field, enter the bastion username.
5. In the Password field, enter the bastion password.
6. In the SSH Private Key field, enter the bastion key.

7. In the SSH Private Key Passphrase field, enter the bastion key passphrase.

Complete the setup process with [Procedure: Finishing PAYG connecting](#).

Procedure: Finishing PAYG connecting

1. To complete adding new PAYG connection data, click **[Create]**.
2. Return to PAYG connection data Details page. The updated connection status is displayed on the top section named Information.
3. Connection status is shown in Admin > Setup Wizard > Pay-as-you-go screen too.
4. If the authentication data for the instance are correct, the column Status shows "Credentials successfully updated."



If the invalid data are entered at any point, the newly created instance is shown in Admin > Setup Wizard > PAYG, with column Status displaying error message.

As soon as the authentication data is available on the server, the list of available products is updated.

Available products are all versions of the same product family and architecture as the one installed in the PAYG instance. For example, if the instance has the SUSE Linux Enterprise Server 15 SP1 product installed, SUSE Linux Enterprise Server 15 SP2, SUSE Linux Enterprise Server 15 SP3, SUSE Linux Enterprise Server 15 SP4 and SUSE Linux Enterprise Server 15 SP5 are automatically shown in Admin > Setup Wizard > Products.

Once the products are shown as available, the user can add a product to SUSE Multi-Linux Manager by selecting the checkbox next to the product name and clicking **[Add product]**.

After the success message you can verify the newly added channels in the Web UI, by navigating to Software > Channel List > All.

To monitor the syncing progress of each channel, check the log files in the `/var/log/rhn/reposync` directory on the SUSE Multi-Linux Manager Server.



If a product is provided by both the PAYG instance and one of the SCC subscriptions, it will appear only once in the products list.

When the channels belonging to that product are synced, the data might still come from the SCC subscription, and not from the Pay-As-You-Go instance.

2.1.7.2. Instance credential collect status

SUSE Multi-Linux Manager server uses credentails collected from the instance to connect to the RMT server and to download the packages using reposync. These credentials are refreshed every 10 minutes by taskomatic using the defined SSH connection data. Connection to RMT server always uses the last known authentication

credentials collected from the PAYG instance.

The status of the PAYG instance credentials collect is shown in the column **Status** or on the instance details page. When the instance is not reachable, the credential update process will fail.

When the instance is unreachable, the credential update process will fail and the credentials will become invalid after the second failed refresh. Synchronization of channels will fail when the credentials are invalid. To avoid this keep the connected instances running.

PAYG instance remains connected to SUSE Multi-Linux Manager server unless SSH connection data is explicitly deleted. To delete the SSH connection data to the instance, use [\[proc-deleting-connection-data-to-instance\]](#).

PAYG instance may not be accessible from the SUSE Multi-Linux Manager server at all times.

- If the instance exists, but is stopped, the last known credentials will be used to try to connect to the instance. How long the credentials remain valid depends on the cloud provider.
- If the instance no longer exists, but is still registered with SUMA, its credentials are no longer valid and the authentication will fail. The error message is shown in the column **Status**.



The error message only indicates that the instance is not available. Further diagnostics about the status of the instance needs to be done on the cloud provider.



Any of the following actions or changes in the PAYG instance will lead to credentials failing: * removing zypper credentials files * removing the imported certificates * removing cloud-specific entries from `/etc/hosts`

2.1.7.3. Registering PAYG system as a client

You can register a PAYG instance from where you harvest the credentials as a Salt client. The instance needs to have a valid cloud connection registered, otherwise it will not have access to channels. If the user removes the cloud packages, the credentials harvesting may stop working.

First set up the PAYG instance to collect authentication data, so it can synchronize the channels.

The rest of the process is the same as for any non-public-cloud client and consists of synchronizing channels, automatic bootstrap script creation, activation key creation and starting the registration.

For more about registering clients, see **Client Configuration Guide › Registration Overview**.

2.1.7.4. Troubleshooting

Checking the credentials

- If the script fails to collect the credentials, it should provide a proper error message in the logs and in the Web UI.
- If the credentials are not working, `reposync` should show the proper error.

Using `registercloudguest`

- Refreshing or changing the `registercloudguest` connection to the public cloud update infrastructure should not interfere with the credentials usage.
- Running `registercloudguest --clean` will cause problems if no new cloud connection is registered with the cloud guest command.

2.2. Install SUSE Multi-Linux Manager Proxy

There are various scenarios to deploy a SUSE Multi-Linux Manager Proxy. All these scenarios presume you have already successfully deployed a SUSE Multi-Linux Manager 5.2 Beta 2 Server.

2.2.1. SUSE Multi-Linux Manager Proxy Deployment

This guide outlines the deployment process for the SUSE Multi-Linux Manager 5.2 Beta 2 Proxy container on SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7. This guide presumes you have already successfully deployed a SUSE Multi-Linux Manager 5.2 Beta 2 Server.



SL Micro is only supported as regular minion (default contact method) for the time being. We are working on managing it as Salt SSH client (salt-ssh contact method), too.



It is possible to convert existing client to proxy. For more information, see **Installation and Upgrade Guide › Container Deployment › Proxy Conversion from Client Mlm**.

To successfully deploy a new proxy, follow the procedure:

Procedure: Deploying proxy

1. Review hardware requirements.
2. Synchronize the SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 parent channel and the proxy extension child channel on the server.
3. Install SL Micro or SUSE Linux Enterprise Server on a bare-metal machine.
4. Create a Salt activation key with Proxy Extension.
5. Bootstrap the proxy as a client with the default connection method.

6. Generate a proxy configuration.
7. Transfer the proxy configuration from server to proxy.
8. Install packages on the proxy.
9. Use the proxy configuration to register the client as a proxy with SUSE Multi-Linux Manager.

Supported operating system for the Proxy Container Host

The supported operating system for the container host are SL Micro 6.2 and SUSE Linux Enterprise Server 15 SP7.



Container host

A container host is a server equipped with a container engine like Podman, which lets it manage and deploy containers. These containers hold applications and their essential parts, such as libraries, but not a full operating system, making them lightweight. This setup ensures applications run the same way in different environments. The container host supplies the necessary resources such as CPU, memory, and storage for these containers.

2.2.1.1. Hardware requirements for the proxy

For more information about hardware requirements for deploying SUSE Multi-Linux Manager Proxy, see **Installation and Upgrade Guide › Hardware Requirements › Proxy Hardware Requirements**.

2.2.1.2. Synchronize the parent and proxy extension child channels

This section presumes that you have already entered your organization credentials under the **Admin › Setup Wizard › Organization Credentials** in the server's Web UI. Products are listed on the **Admin › Setup Wizard › Products** page. This channel must be fully synchronized on the server, with the child channel Proxy as an extension option selected.

Procedure: Synchronizing the parent channel and proxy extension

1. In the SUSE Multi-Linux Manager Web UI select **Admin › Products**.
2. From the products page enter SL Micro or SUSE Linux Enterprise Server in the filter field.

3. Next use the drop-down to select the required architecture. For this example x86-64.
4. In the Product Description field select the SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 checkbox then use the drop-down to select the SUSE Multi-Linux Manager Proxy Extension 5.2 x86_64 extension.
5. Click the **[Add products]** button.
6. Wait for the synchronization to complete.

2.2.1.3. Prepare the SUSE Multi-Linux Manager proxy host

In the following subsections, you either prepare the proxy host with SLE Micro or SUSE Linux Enterprise Server.

2.2.1.3.1. Prepare SL Micro 6.2 Host

Download the installation media

Procedure: Downloading the installation media

1. Locate the SL Micro 6.2 installation media at <https://www.suse.com/download/sle-micro/>, and download the appropriate media file.
2. Prepare a DVD or USB flash drive with the downloaded .iso image for installation.

Install SL Micro 6.2

Procedure: Installing SL Micro 6.2

1. Insert the DVD or USB flash drive (USB disk or key) containing the installation image for SLE Micro 6.2.
2. Boot or reboot your system.
3. Use the arrow keys to select Installation.
4. Adjust Keyboard and language.
5. Click the checkbox to accept the license agreement.

6. Click Next to continue.
7. Skip the registration. The SL Micro 6.2 entitlement is included within the SUSE Multi-Linux Manager entitlement.
8. Click **[Next]** to continue.
9. On the NTP Configuration page click **[Next]**.
10. On the Authentication for the System page enter a password for the root user. Click **[Next]**.
11. On the Installation Settings page click **[Install]**.

This concludes installation of SL Micro 6.2 and SUSE Multi-Linux Manager 5.2 Beta 2 as an extension. For more information about preparing your machines (virtual or physical), see the [SL Micro Deployment Guide](#).

Update the system

Procedure: Updating the system

1. Log in as **root**.
2. Run **transactional-update**:

```
transactional-update
```

3. Reboot.



SL Micro is designed to update itself automatically by default and will reboot after applying updates. However, this behavior is not desirable for the SUSE Multi-Linux Manager environment. To prevent automatic updates on your server, SUSE Multi-Linux Manager disables the transactional-update timer during the bootstrap process.

If you prefer the SL Micro default behavior, enable the timer by running the following command:

```
systemctl enable --now transactional-update.timer
```

To continue with deployment, see **Installation and Upgrade Guide › Container Deployment › Proxy Deployment Mlm › Deploy Mlm Proxy Persistent Storage**.

2.2.1.3.2. Prepare SUSE Linux Enterprise Server 15 SP7 host

Alternatively, you can deploy SUSE Multi-Linux Manager on SUSE Linux Enterprise Server 15 SP7.

The following procedures describe the main steps of the installation process.

Install SUSE Multi-Linux Manager extensions on SUSE Linux Enterprise Server

Procedure: Installing SUSE Multi-Linux Manager Extensions on SUSE Linux Enterprise Server

1. Locate and download SUSE Linux Enterprise Server 15 SP7 .iso at <https://www.suse.com/download/sles/>.
2. Start the installation of SUSE Linux Enterprise Server 15 SP7.
 - a. On the Language, keyboard and product selection select the product to install.
 - b. On the License agreement read the agreement and check I Agree to the License Terms.
3. Skip the registration.
4. Click Next to continue.



Please note that for SUSE Linux Enterprise Server 15 SP7, you are required to have a valid SUSE Linux Enterprise Server subscription configured on the server.

5. In the screen Extensions and Modules Selection check the following:
 - Basesystem Module
 - Containers Module
6. Click **[Next]** to continue.
7. Complete the installation.
8. When the installation completes, log in to the newly installed server as root.

9. Update the System (optional, if the system was not set to download updates during install):

```
zypper up
```

10. Reboot.

To continue with deployment, see **Installation and Upgrade Guide › Container Deployment › Proxy Deployment Mlm › Deploy Mlm Proxy Persistent Storage**.

2.2.1.4. Configure custom persistent storage

Configuring persistent storage is optional, but it is the only way to avoid serious trouble with container full disk conditions. If custom persistent storage is required for your infrastructure, use the `mgr-storage-proxy` tool.

- For more information, see `mgr-storage-proxy --help`. This tool simplifies creating the container storage and Squid cache volumes.

Use the command in the following manner:

```
mgr-storage-proxy <storage-disk-device>
```

For example:

```
mgr-storage-proxy /dev/nvme1n1
```



This command will create the persistent storage volumes at `/var/lib/containers/storage/volumes`.

For more information, see

- **Installation and Upgrade Guide › Container Management › Persistent Container Volumes**
- **Administration Guide › Troubleshooting › Tshoot Container Full Disk**

2.2.1.5. Create an Activation Key for the Proxy

Procedure: Creating an Activation Key

1. Navigate to **Systems › Activation Keys**, and click **[Create key]**.

2. Create an activation key for the proxy host with SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 as the parent channel. This key should include all recommended channels and the proxy as an extension child channel.
3. Proceed to bootstrapping the proxy host as a default client.



Ensure the Proxy is assigned only to original vendor channels.

Assigning cloned channels is not supported at this moment.

2.2.1.6. Bootstrap the Proxy Host as a Client

Procedure: Bootstrapping the Proxy Host

1. Select **Systems › Bootstrapping**.
2. Fill in the fields for your proxy host.
3. Select the activation key created in the previous step from the drop-down.
4. Click **[Bootstrap]**.
5. Wait for the bootstrap process to complete successfully. Check the **Salt** menu and confirm the Salt key is listed and accepted.
6. Reboot the proxy host if the operating system is SL Micro.
7. Select the host from the **System** list and trigger a second reboot in case of SL Micro after all events are finished to conclude the onboarding.

Procedure: Updating the Proxy Host

1. Select the host from the **Systems** list and apply all patches to update it.
2. Reboot the proxy host if the operating system is SL Micro.

2.2.1.7. Generate Proxy Configuration

The configuration archive of the SUSE Multi-Linux Manager Proxy is generated by the SUSE Multi-Linux Manager Server. Each additional Proxy requires its own configuration archive.

For the containerized SUSE Multi-Linux Manager 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.



For Podman deployment, the container host for the SUSE Multi-Linux Manager Proxy must be registered as a client to the SUSE Multi-Linux Manager Server prior to generating this proxy configuration.

If a proxy FQDN is used to generate a proxy container configuration that is not a registered client (as in the Kubernetes use case), a new system entry will appear in system list. This new entry will be shown under previously entered Proxy FQDN value and will be of Foreign system type.



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

On the hub server, run the following command.

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

The files to use will be

1. /root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT as the root CA,
2. /root/ssl-build/<hostname>/server.crt as the proxy certificate and
3. /root/ssl-build/<hostname>/server.key as the proxy certificate's key.

2.2.1.7.1. Generate the Proxy Configuration with Web UI

Procedure: Generating a Proxy Container Configuration Using Web UI

1. In the Web UI, navigate to **Systems › Proxy Configuration** and fill the required data:
2. In the Proxy FQDN field type fully qualified domain name for the proxy.
3. In the Parent FQDN field type fully qualified domain name for the SUSE Multi-Linux Manager Server or another SUSE Multi-Linux Manager Proxy.
4. In the Proxy SSH port field type SSH port on which SSH service is listening on SUSE Multi-Linux Manager Proxy. Recommended is to keep default 8022.

5. In the 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.



2 GB represents the default proxy squid cache size. This will need to be adjusted for your environment.

6. In the SSL certificate selection list choose if new server certificate should be generated for SUSE Multi-Linux Manager Proxy or an existing one should be used. You can consider generated certificates as SUSE Multi-Linux Manager builtin (self signed) certificates. If SUSE Multi-Linux Manager server runs on Kubernetes, the generated certificate option is not possible and replaced with no SSL certificate as they are managed outside the containers.

Depending on the choice then provide either path to signing CA certificate to generate a new certificate or path to an existing certificate and its key to be used as proxy certificate.

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

For more information about existing or custom certificates and the concept of corporate and intermediate certificates, see **Administration Guide › Ssl Certs Imported**.

7. Click **[Generate]** to register a new proxy FQDN in the SUSE Multi-Linux Manager Server and generate a configuration archive (`config.tar.gz`) containing details for the container host.
8. After a few moments you are presented with file to download. Save this file locally.

2.2.1.7.2. Generate Proxy Configuration With spacecmd and Self-Signed Certificate

You can generate a Proxy configuration using spacecmd. This is only possible if SUSE Multi-Linux Manager server runs on podman and has a self-signed root CA certificate.

Procedure: Generating Proxy Configuration with spacecmd and Self-Signed Certificate

1. SSH into your container host.
2. Execute the following command replacing the Server and Proxy 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'
```

3. Copy the generated configuration from the server container:

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

2.2.1.7.3. Generate Proxy Configuration With spacecmd and Custom Certificate

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

Procedure: Generating Proxy Configuration with spacecmd and Custom Certificate

1. SSH into your Server container host.
2. 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'
```

3. 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'
```

4. Copy the generated configuration from the server container:

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

2.2.1.7.4. Generate Proxy Configuration With spacecmd and no Certificate

You can generate a Proxy configuration using spacecmd with no TLS certificates. This is needed for SUSE Multi-Linux Manager running on Kubernetes as the certificates are handled outside of the containers.

Procedure: Generating Proxy Configuration with spacecmd and no Certificate

1. SSH into your Server container host.
2. 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_nossl -- -p 8022
pxy.example.com srv.example.com 2048 email@example.com -o /tmp/config.tar.gz'
```

3. Copy the generated configuration from the server container:

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

2.2.1.8. Transfer the Proxy Configuration

The Web UI generates a configuration archive. This archive needs to be made available on the proxy container host.

Procedure: Copying the Proxy Configuration

1. If not already done, copy the configuration archive (config.tar.gz) generated in the previous step from the server container to the server host:

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

2. If not already done, copy the files from the server host to the proxy host:

```
scp config.tar.gz <proxy-FQDN>:/root
```

2.2.1.9. Install packages and enable podman

Before using proxy, some packages need to be present on host and podman needs to be running.

Procedure: Preparing the prerequisites

1. On the proxy host, ensure that the following packages are installed:
 - podman
 - mgrpxy-bash-completion
 - suse-multi-linux-manager-<version>-<arch>-proxy-httpd-image
 - suse-multi-linux-manager-<version>-<arch>-proxy-salt-broker-image
 - suse-multi-linux-manager-<version>-<arch>-proxy-squid-image
 - suse-multi-linux-manager-<version>-<arch>-proxy-ssh-image
 - suse-multi-linux-manager-<version>-<arch>-proxy-tftpd-image

For example, for version 5.2 and architecture of x86_64, the package name would be suse-multi-linux-manager-5.1-x86_64-proxy-httpd-image.

2. Start the Podman service on the proxy host by rebooting the system, or running a command:

```
systemctl enable --now podman.service
```

3. On the proxy host, install the Proxy with:

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


2.2.1.10. Start the SUSE Multi-Linux Manager proxy

Container can now be started with the mgrpxy command:

Procedure: Starting and checking proxy status

1. Start the proxy by calling:

```
mgrpxy start
```

2. Check container status by calling:

```
mgrpxy status
```

Five SUSE Multi-Linux Manager Proxy containers should be present and should be part of the proxy-pod container pod:

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

2.2.1.11. Use a custom container image for a service

By default, the SUSE Multi-Linux Manager Proxy suite is configured to use the same image version and registry path for each of its services. However, it is possible to override the default values for a specific service using the install parameters ending with `-tag` and `-image`.

For example:

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

It adjusts the configuration file for the httpd service, where `registry.opensuse.org/uyuni/proxy-httpd` is the image to use and `0.1.0` is the version tag, before restarting it.

To reset the values to defaults, run the install command again without those parameters:

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

This command first resets the configuration of all services to the global defaults and then reloads it.

2.2.2. Convert a Client to MLM Proxy

2.2.2.1. Overview

This chapter describes how to convert a client system into a SUSE Multi-Linux Manager Proxy using the Web UI.

It assumes that the proxy host system has already been bootstrapped, is subscribed to the base operating system channel (such as SUSE Linux Enterprise Server 15 SP7 or SL Micro 6.2) and to the Proxy Extension channel.

For information about client onboarding, see **Client Configuration Guide › Registration Overview**.

2.2.2.2. Requirements

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

2.2.2.2.1. Supported Systems

Only the following operating systems are currently supported for proxy conversion:

- SUSE Linux Enterprise Server 15 SP7
- SL Micro 6.1

2.2.2.2.2. Client Must Be

- Already onboarded in SUSE Multi-Linux Manager
- Reachable via the network
- Subscribed to the appropriate proxy extension channel:
 - SUSE Multi-Linux Manager Proxy Extension 5.2 (matching architecture)

2.2.2.3. Preparation

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

2.2.2.3.1. 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 SUSE Multi-Linux Manager server
- A certificate for the proxy.
- The corresponding private key for the proxy certificate.



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 SUSE Multi-Linux Manager container.

Generate a proxy certificate

1. On the SUSE Multi-Linux Manager server host, run:

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

For more information about other parameters, see **Administration Guide › Ssl Certs Selfsigned**.

2. Transfer the certificates to SUSE Multi-Linux Manager server host

```
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
```



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/*/
```

3. Transfer the certificates from the SUSE Multi-Linux Manager server host to your local machine or other target system:

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

2.2.2.3.2. Packages Preparation

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

- suse-multi-linux-manager-5.2-<ARCH>-proxy-httpd-image
- suse-multi-linux-manager-5.2-<ARCH>-proxy-salt-broker-image
- suse-multi-linux-manager-5.2-<ARCH>-proxy-squid-image
- suse-multi-linux-manager-5.2-<ARCH>-proxy-ssh-image
- suse-multi-linux-manager-5.2-<ARCH>-proxy-tftpd-image

You can install these packages from the Web UI by navigating to the Software > Packages > Install tab, then searching for the packages above, and installing them.

For details on air-gapped deployment, see **Installation and Upgrade Guide › Container Deployment › Proxy Air Gapped Deployment Mlm**

2.2.2.4. Setup Proxy Client

1. Navigate to the client's Overview page.
2. Click button **[Convert to Proxy]**.

Confirm you were redirected to the proxy configuration form.

This page can be accessed later from the Details > Proxy > Configuration tab.

3. In the Web UI, navigate to **Proxy › Configuration** and fill in the required data:

Procedure: Configuring the Proxy

- a. In the Parent FQDN field, type the fully qualified domain name for the parent server or proxy.
- b. In the Proxy SSH port field, type the SSH port on which the SSH service is listening on the SUSE Multi-Linux Manager Proxy. It is recommended to keep the default: 8022.
- c. In the Max Squid cache size field, type the maximum allowed size for the Squid cache, in Gigabytes.
- d. In the Proxy admin email field, type the administrator's email address.
- e. In the Certificates section, provide the certificates for the SUSE Multi-Linux Manager Proxy, obtained in the preparation step.
- f. In the Source section, select one of the two options: RPM or Registry.

- The RPM option is recommended for air-gapped or restricted environments.
- The 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: Simple or Advanced.

- If Simple is selected, provide values in the Registry URL and Containers Tag fields.
 - For Registry URL use: `registry.suse.com/suse/multi-linux-manager/5.2/x86_64`.
 - Select the tag from the drop-down list.
- If Advanced is selected, an additional section of the form is shown:
 - For each individual container URL field, use the registry: `registry.suse.com/suse/multi-linux-manager/5.2/x86_64` followed by the corresponding suffix, for example, `proxy-httpd` or `salt-broker`.
 - Select the tag from the drop-down list.

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

2.2.2.5. 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.

2.2.3. SUSE Multi-Linux Manager Proxy Deployment as a Virtual Machine - KVM

This chapter provides the Virtual Machine settings for deployment of SUSE Multi-Linux Manager 5.2 Beta 2 Proxy as an image. KVM will be combined with Virtual Machine Manager (virt-manager) as a sandbox for this installation.

2.2.3.1. Available Images



The preferred method for deploying SUSE Multi-Linux Manager Proxy is to use one of the following available images. All tools are included in these images simplifying deployment.

Images for SUSE Multi-Linux Manager 5.2 Beta 2 Proxy are available at [SUSE Multi-Linux Manager 5.2 Beta 2 VM images](#).



Customized SUSE Multi-Linux Manager 5.2 Beta 2 VM images are provided only for SL Micro 6.2. To run the product on SUSE Linux Enterprise Server 15 SP7, use the standard SUSE Linux Enterprise Server 15 SP7 installation media available at <https://www.suse.com/download/sles/> and enable the SUSE Multi-Linux Manager 5.2

Beta 2 extensions on top of it.

For more information on preparing raw images, see <https://documentation.suse.com/sle-micro/6.1/html/Micro-deployment-raw-images-virtual-machines/index.html#deployment-preparing-configuration-device>.



For additional information on the self install images, see <https://documentation.suse.com/sle-micro/6.1/html/Micro-deployment-selfinstall-images/index.html>

Table 12. Available Proxy Images

Architecture	Image Format
aarch64	qcow2, vmdk
x86_64	qcow2, vmdk, raw, Self Installer

2.2.3.2. Virtual Machine Manager (virt-manager) Settings

Enter the following settings when creating a new virtual machine using **virt-manager**.



This table specifies the minimum requirements. These are suitable for a quick test installation, such as a proxy with one client.

If you want to use a production environment and need background information about disk space, see **Installation and Upgrade Guide › Hardware Requirements**.

KVM Settings	
Installation Method	Import Existing Disk Image
OS:	Linux
Version:	SUSE Multi-Linux Manager-Proxy.x86_64-5.2.*.qcow2
Memory:	Minimum *)
CPU's:	Minimum *)
Storage Format:	.qcow2 40 GB (Default) Root Partition
Name:	test-setup
Network	Bridge br0

*) For minimum values, see **Installation and Upgrade Guide › Hardware Requirements › Proxy Hardware Requirements**.



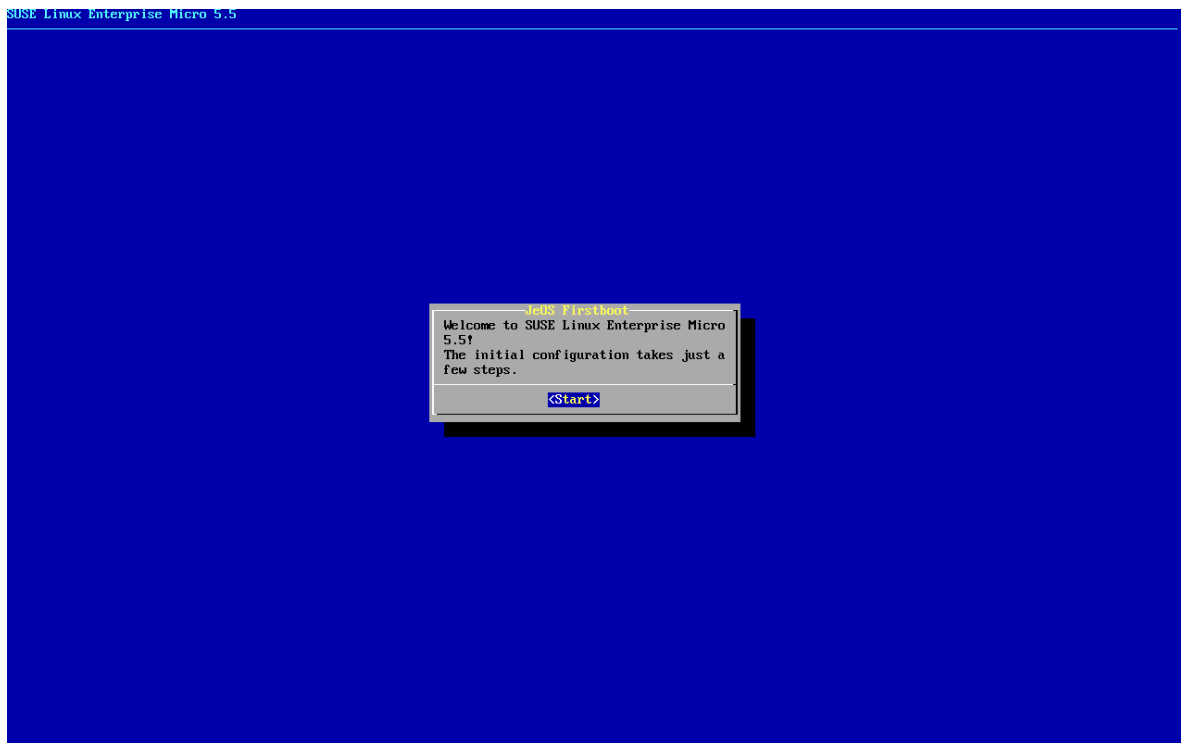
/var/lib/containers/storage/volumes Minimum 100 GB. Storage requirements should be calculated for the number of ISO distribution images, containers, and bootstrap repositories you will use.

2.2.3.3. Initial KVM Setup

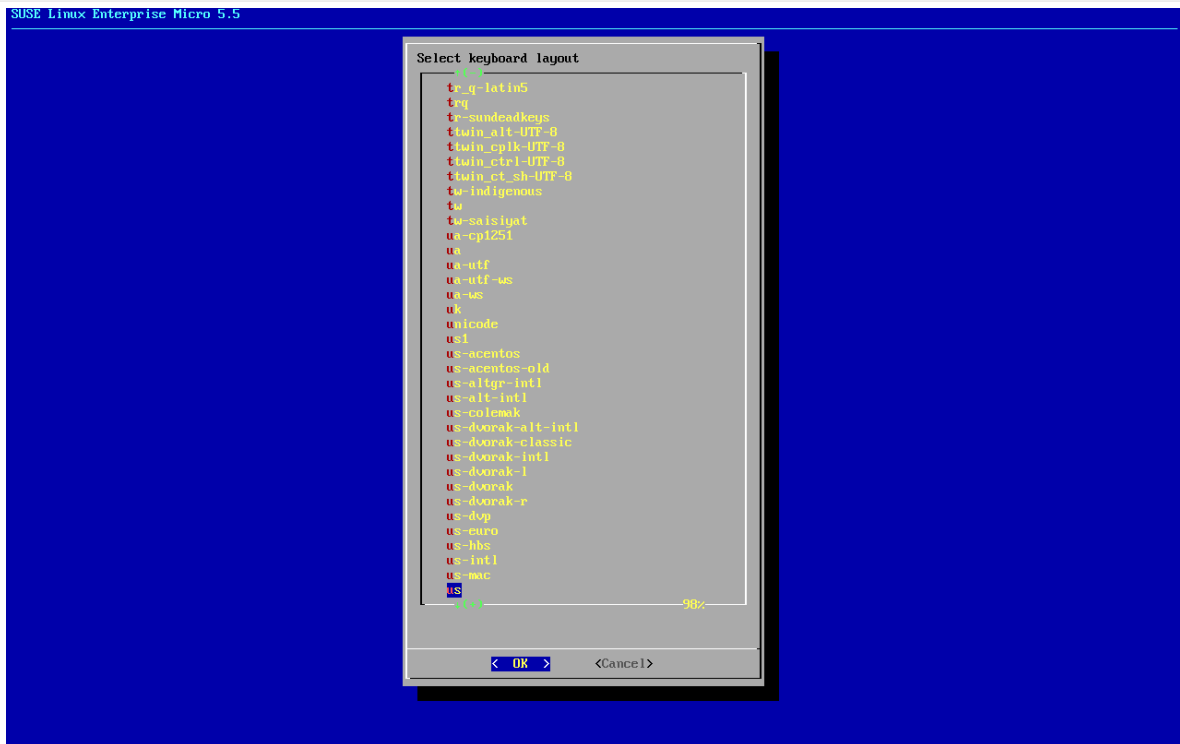
For settings, see **Installation and Upgrade Guide › Container Deployment › Proxy Deployment Vm Mlm › Proxy Quickstart.sect.kvm.settings**.

Procedure: Creating Initial Setup

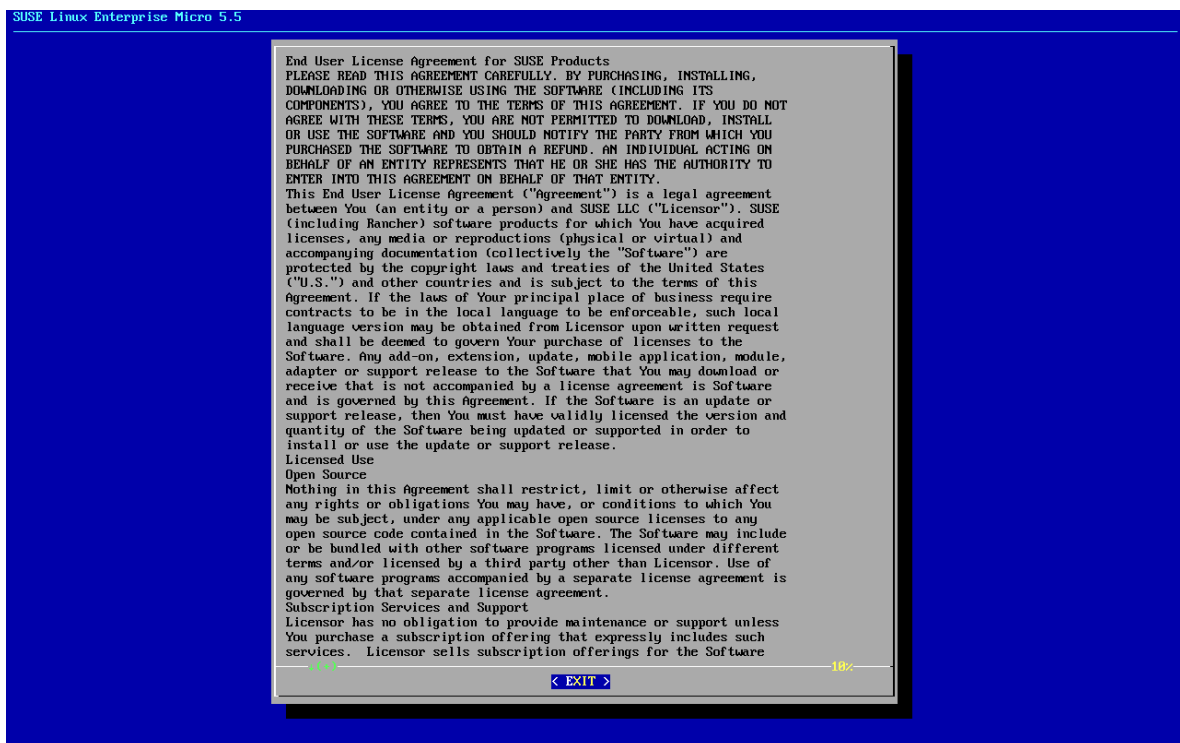
1. Create a new virtual machine using the downloaded Minimal KVM image and select Import existing disk image.
2. Configure RAM and number of CPUs with minimum values. *)
3. Name your KVM machine and select the Customize configuration before install check box.
4. Click **[Begin Installation]** to boot from the image.
5. At the JeOS Firstboot screen select start to continue.



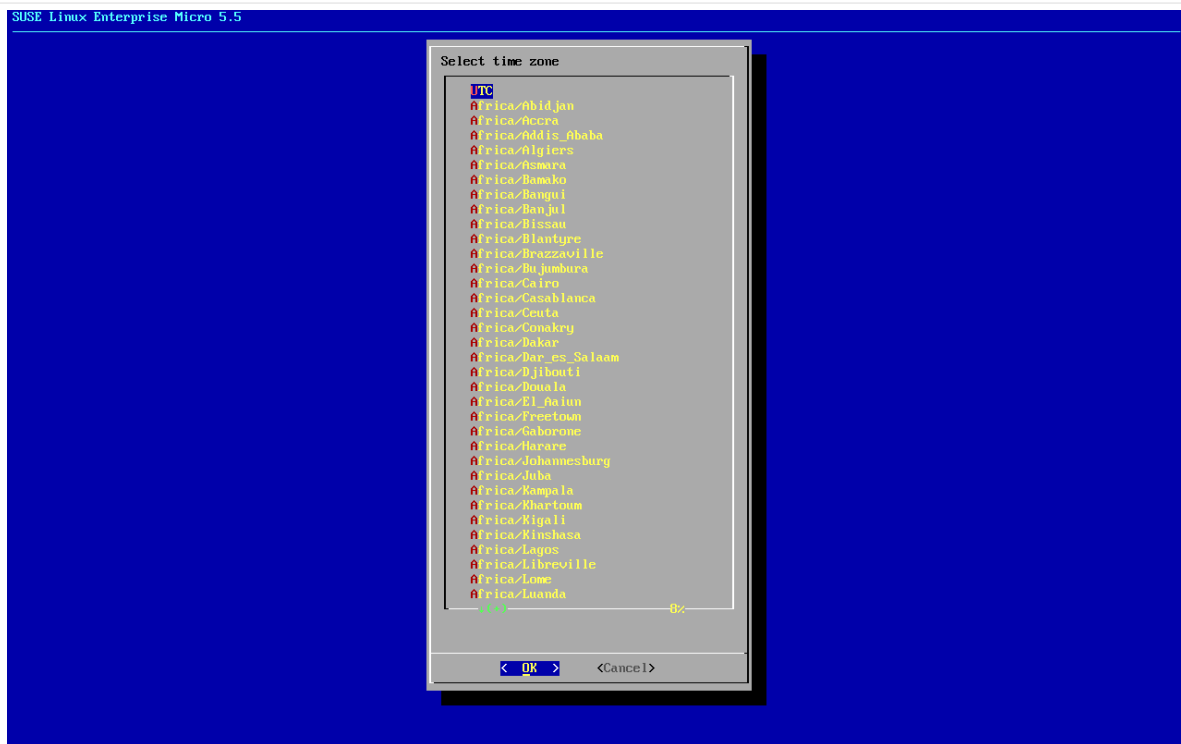
6. Select keyboard layout.



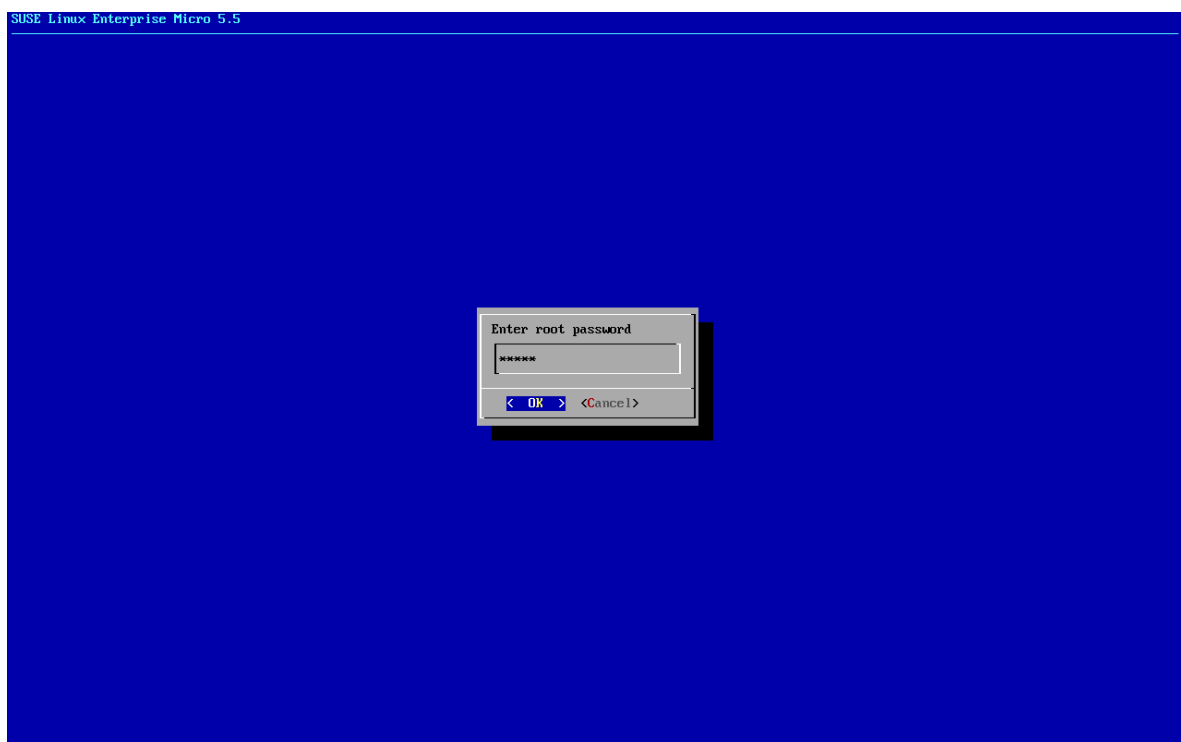
7. Accept the license agreement.



8. Select your time zone.



9. Enter a password for root.



10. When installation completes log in as root.

11. Proceed to the next section.

*) For minimum values, see **Installation and Upgrade Guide › Hardware Requirements › Proxy Hardware Requirements**.

2.2.3.4. Register SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

Procedure: Registering SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

1. Boot the virtual machine.
2. Log in as root.
3. Register SL Micro with SCC.

```
transactional-update register -r <REGCODE> -e <your_email>
```

4. Reboot.
5. Register SUSE Multi-Linux Manager 5.2 Beta 2 Proxy with SUSE Customer Center.

```
transactional-update register -p Multi-Linux-Manager-Proxy/5.2/x86_64 -r <REGCODE>
```

6. Reboot.
7. Update the system:

```
transactional-update
```

8. If updates were applied reboot.
9. This step is optional. However, if custom persistent storage is required for your infrastructure, use the `mgr-storage-proxy` tool.
 - For more information, see `mgr-storage-proxy --help`. This tool simplifies creating the container volumes.
 - Use the command in the following manner:

```
mgr-storage-proxy <storage-disk-device>
```

For example:

```
mgr-storage-proxy /dev/nvme1n1
```



This command will move the persistent storage volumes at `/var/lib/containers/storage/volumes` to the specified storage device.

For more information, see

- [Installation and Upgrade Guide › Container Management › Persistent Container Volumes](#)
- [Administration Guide › Troubleshooting › Tshoot Container Full Disk](#)

2.2.3.5. Create an Activation Key for the Proxy

Procedure: Creating an Activation Key

1. Navigate to **Systems › Activation Keys** , and click **[Create key]**.
2. Create an activation key for the proxy host with SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 as the parent channel. This key should include all recommended channels and the proxy as an extension child channel.
3. Proceed to bootstrapping the proxy host as a default client.



Ensure the Proxy is assigned only to original vendor channels.

Assigning cloned channels is not supported at this moment.

2.2.3.6. Generate Proxy Configuration

The configuration archive of the SUSE Multi-Linux Manager Proxy is generated by the SUSE Multi-Linux Manager Server. Each additional Proxy requires its own configuration archive.

For the containerized SUSE Multi-Linux Manager 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.



For Podman deployment, the container host for the SUSE Multi-Linux Manager Proxy must be registered as a client to the SUSE Multi-Linux Manager Server prior to generating this proxy configuration.

If a proxy FQDN is used to generate a proxy container configuration that is not a registered client (as in the Kubernetes use case), a new system entry will appear in system list. This new entry will be shown under previously entered Proxy FQDN value and will be of Foreign system type.



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

On the hub server, run the following command.

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

The files to use will be

1. /root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT as the root CA,
2. /root/ssl-build/<hostname>/server.crt as the proxy certificate and
3. /root/ssl-build/<hostname>/server.key as the proxy certificate's key.

2.2.3.6.1. Generate the Proxy Configuration with Web UI

Procedure: Generating a Proxy Container Configuration Using Web UI

1. In the Web UI, navigate to **Systems › Proxy Configuration** and fill the required data:
2. In the Proxy FQDN field type fully qualified domain name for the proxy.
3. In the Parent FQDN field type fully qualified domain name for the SUSE Multi-Linux Manager Server or another SUSE Multi-Linux Manager Proxy.
4. In the Proxy SSH port field type SSH port on which SSH service is listening on SUSE Multi-Linux Manager Proxy. Recommended is to keep default 8022.
5. In the 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.



2 GB represents the default proxy squid cache size. This will need to be adjusted for your environment.

6. In the SSL certificate selection list choose if new server certificate should be generated for SUSE Multi-Linux Manager Proxy or an existing one should be used. You can consider generated certificates as SUSE Multi-Linux Manager builtin (self signed) certificates. If SUSE Multi-Linux Manager server runs on Kubernetes, the generated certificate option is not possible

and replaced with no SSL certificate as they are managed outside the containers.

Depending on the choice then provide either path to signing CA certificate to generate a new certificate or path to an existing certificate and its key to be used as proxy certificate.

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

For more information about existing or custom certificates and the concept of corporate and intermediate certificates, see **Administration Guide › Ssl Certs Imported**.

7. Click **[Generate]** to register a new proxy FQDN in the SUSE Multi-Linux Manager Server and generate a configuration archive (config.tar.gz) containing details for the container host.
8. After a few moments you are presented with file to download. Save this file locally.

2.2.3.6.2. Generate Proxy Configuration With spacecmd and Self-Signed Certificate

You can generate a Proxy configuration using spacecmd. This is only possible if SUSE Multi-Linux Manager server runs on podman and has a self-signed root CA certificate.

Procedure: Generating Proxy Configuration with spacecmd and Self-Signed Certificate

1. SSH into your container host.
2. Execute the following command replacing the Server and Proxy 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'
```

3. Copy the generated configuration from the server container:

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

2.2.3.6.3. Generate Proxy Configuration With spacecmd and Custom Certificate

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

Procedure: Generating Proxy Configuration with spacecmd and Custom Certificate

1. SSH into your Server container host.
2. 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'
```

3. 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'
```

4. Copy the generated configuration from the server container:

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

2.2.3.6.4. Generate Proxy Configuration With spacecmd and no Certificate

You can generate a Proxy configuration using spacecmd with no TLS certificates. This is needed for SUSE Multi-Linux Manager running on Kubernetes as the certificates are handled outside of the containers.

Procedure: Generating Proxy Configuration with spacecmd and no Certificate

1. SSH into your Server container host.
2. 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_nossl -- -p 8022
pxy.example.com srv.example.com 2048 email@example.com -o /tmp/config.tar.gz'
```

3. Copy the generated configuration from the server container:

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

2.2.3.7. Transfer the Proxy Configuration

The Web UI generates a configuration archive. This archive needs to be made available on the proxy container host.

Procedure: Copying the Proxy Configuration

1. If not already done, copy the configuration archive (config.tar.gz) generated in the previous step from the server container to the server host:

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

2. If not already done, copy the files from the server host to the proxy host:

```
scp config.tar.gz <proxy-FQDN>:/root
```

2.2.3.8. Start the SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

Container can now be started with the mgrpxy command:

Procedure: Start and Check Proxy Status

1. Start the Proxy by calling:

```
mgrpxy start
```

2. Check container status by calling:

```
mgrpxy status
```

Five SUSE Multi-Linux Manager Proxy containers should be present and should be part of the proxy-pod container pod:

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

2.2.3.8.1. Using a Custom Container Image for a Service

By default, the SUSE Multi-Linux Manager Proxy suite is set to use the same image version and registry path for each of its services. However, it is possible to override the default values for a specific service using the install parameters ending with `-tag` and `-image`.

For example, use it like this:

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

It adjusts the configuration file for the httpd service, where `registry.opensuse.org/uyuni/proxy-httpds` is the image to use and `0.1.0` is the version tag, before restarting it.

To reset the values to defaults, run the install command again without those parameters:

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

This command first resets the configuration of all services to the global defaults and then reloads it.

2.2.4. SUSE Multi-Linux Manager Proxy Deployment as a Virtual Machine - VMware

This chapter provides the Virtual Machine settings for deployment of SUSE Multi-Linux Manager 5.2 Beta 2 Proxy as an image. VMware will be used as a sandbox for this installation.

2.2.4.1. Available Images



The preferred method for deploying SUSE Multi-Linux Manager Proxy is to use one of the

following available images. All tools are included in these images simplifying deployment.

Images for SUSE Multi-Linux Manager 5.2 Beta 2 Proxy are available at [SUSE Multi-Linux Manager 5.2 Beta 2 VM images](#).



Customized SUSE Multi-Linux Manager 5.2 Beta 2 VM images are provided only for SL Micro 6.2. To run the product on SUSE Linux Enterprise Server 15 SP7, use the standard SUSE Linux Enterprise Server 15 SP7 installation media available at <https://www.suse.com/download/sles/> and enable the SUSE Multi-Linux Manager 5.2 Beta 2 extensions on top of it.



For more information on preparing raw images, see <https://documentation.suse.com/sle-micro/6.1/html/Micro-deployment-raw-images-virtual-machines/index.html#deployment-preparing-configuration-device>.

For additional information on the self install images, see <https://documentation.suse.com/sle-micro/6.1/html/Micro-deployment-selfinstall-images/index.html>

Table 13. Available Proxy Images

Architecture	Image Format
aarch64	qcow2, vmdk
x86_64	qcow2, vmdk, raw, Self Installer

2.2.4.2. Virtual Machine Settings - VMware

This section describes VMware configurations, focusing on the creation of an extra virtual disk essential for the SUSE Multi-Linux Manager Proxy storage partition within VMware environments.



This section specifies the minimum requirements. These are suitable for a quick test installation, such as a proxy with one client.

If you want to use a production environment and need background information about disk space, see **Installation and Upgrade Guide › Hardware Requirements**.

Procedure: Creating the VMware Virtual Machine

1. Download SUSE Multi-Linux Manager Proxy .vmdk file then transfer a copy to your VMware storage.
2. Make a copy of uploaded .vmdk file using VMware web interface. This will convert provided .vmdk file to the format suitable for vSphere hypervisor.

3. Create and name a new virtual machine based on the Guest OS Family Linux and Guest OS Version SUSE Linux Enterprise 15 (64-bit).
4. Add an additional Hard Disk 2 of 100 GB (or more).
5. Configure RAM and number of CPUs with minimum values. *)
6. Set the network adapter as required.
7. Power on the VM, and follow firstboot dialogs (keyboard layout, license agreement, time zone, password for root).
8. When installation completes log in as root.
9. Proceed to the next section.

*) For minimum values, see **Installation and Upgrade Guide › Hardware Requirements › Proxy Hardware Requirements**.

2.2.4.3. Register SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

Procedure: Registering SL Micro and SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

1. Boot the virtual machine.
2. Log in as root.
3. Register SL Micro with SCC.

```
transactional-update register -r <REGCODE> -e <your_email>
```

4. Reboot.
5. Register SUSE Multi-Linux Manager 5.2 Beta 2 Proxy with SUSE Customer Center.

```
transactional-update register -p Multi-Linux-Manager-Proxy/5.2/x86_64 -r <REGCODE>
```

6. Reboot.
7. Update the system:

```
transactional-update
```

8. If updates were applied reboot.
9. This step is optional. However, if custom persistent storage is required for your infrastructure, use the mgr-storage-proxy tool.

- For more information, see `mgr-storage-proxy --help`. This tool simplifies creating the container volumes.
- Use the command in the following manner:

```
mgr-storage-proxy <storage-disk-device>
```

For example:

```
mgr-storage-proxy /dev/nvme1n1
```



This command will move the persistent storage volumes at `/var/lib/containers/storage/volumes` to the specified storage device.

For more information, see

- [Installation and Upgrade Guide › Container Management › Persistent Container Volumes](#)
- [Administration Guide › Troubleshooting › Tshoot Container Full Disk](#)

2.2.4.4. Create an Activation Key for the Proxy

Procedure: Creating an Activation Key

1. Navigate to **Systems › Activation Keys**, and click **[Create key]**.
2. Create an activation key for the proxy host with SL Micro 6.2 or SUSE Linux Enterprise Server 15 SP7 as the parent channel. This key should include all recommended channels and the proxy as an extension child channel.
3. Proceed to bootstrapping the proxy host as a default client.



Ensure the Proxy is assigned only to original vendor channels.

Assigning cloned channels is not supported at this moment.

2.2.4.5. Generate Proxy Configuration

The configuration archive of the SUSE Multi-Linux Manager Proxy is generated by the SUSE Multi-Linux Manager Server. Each additional Proxy requires its own configuration archive.

For the containerized SUSE Multi-Linux Manager 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.



For Podman deployment, the container host for the SUSE Multi-Linux Manager Proxy must be registered as a client to the SUSE Multi-Linux Manager Server prior to generating this proxy configuration.

If a proxy FQDN is used to generate a proxy container configuration that is not a registered client (as in the Kubernetes use case), a new system entry will appear in system list. This new entry will be shown under previously entered Proxy FQDN value and will be of Foreign system type.



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

On the hub server, run the following command.

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

The files to use will be

1. /root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT as the root CA,
2. /root/ssl-build/<hostname>/server.crt as the proxy certificate and
3. /root/ssl-build/<hostname>/server.key as the proxy certificate's key.

2.2.4.5.1. Generate the Proxy Configuration with Web UI

Procedure: Generating a Proxy Container Configuration Using Web UI

1. In the Web UI, navigate to **Systems › Proxy Configuration** and fill the required data:
2. In the Proxy FQDN field type fully qualified domain name for the proxy.
3. In the Parent FQDN field type fully qualified domain name for the SUSE Multi-Linux Manager Server or another SUSE Multi-Linux Manager Proxy.
4. In the Proxy SSH port field type SSH port on which SSH service is listening on SUSE Multi-Linux Manager Proxy. Recommended is to keep default

8022.

5. In the 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.



2 GB represents the default proxy squid cache size. This will need to be adjusted for your environment.

6. In the SSL certificate selection list choose if new server certificate should be generated for SUSE Multi-Linux Manager Proxy or an existing one should be used. You can consider generated certificates as SUSE Multi-Linux Manager builtin (self signed) certificates. If SUSE Multi-Linux Manager server runs on Kubernetes, the generated certificate option is not possible and replaced with no SSL certificate as they are managed outside the containers.

Depending on the choice then provide either path to signing CA certificate to generate a new certificate or path to an existing certificate and its key to be used as proxy certificate.

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

For more information about existing or custom certificates and the concept of corporate and intermediate certificates, see **Administration Guide › Ssl Certs Imported**.

7. Click **[Generate]** to register a new proxy FQDN in the SUSE Multi-Linux Manager Server and generate a configuration archive (config.tar.gz) containing details for the container host.
8. After a few moments you are presented with file to download. Save this file locally.

2.2.4.5.2. Generate Proxy Configuration With spacecmd and Self-Signed Certificate

You can generate a Proxy configuration using spacecmd. This is only possible if SUSE Multi-Linux Manager server runs on podman and has a self-signed root CA certificate.

Procedure: Generating Proxy Configuration with spacecmd and Self-Signed Certificate

1. SSH into your container host.
2. Execute the following command replacing the Server and Proxy 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'
```

3. Copy the generated configuration from the server container:

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

2.2.4.5.3. Generate Proxy Configuration With spacecmd and Custom Certificate

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

Procedure: Generating Proxy Configuration with spacecmd and Custom Certificate

1. SSH into your Server container host.
2. 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'
```

3. 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'
```

4. Copy the generated configuration from the server container:

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

2.2.4.5.4. Generate Proxy Configuration With spacecmd and no Certificate

You can generate a Proxy configuration using spacecmd with no TLS certificates. This is needed for SUSE Multi-Linux Manager running on Kubernetes as the certificates are handled outside of the containers.

Procedure: Generating Proxy Configuration with spacecmd and no Certificate

1. SSH into your Server container host.
2. 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_nossl -- -p 8022
pxy.example.com srv.example.com 2048 email@example.com -o /tmp/config.tar.gz'
```

3. Copy the generated configuration from the server container:

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

2.2.4.6. Transfer the Proxy Configuration

The Web UI generates a configuration archive. This archive needs to be made available on the proxy container host.

Procedure: Copying the Proxy Configuration

1. If not already done, copy the configuration archive (config.tar.gz) generated in the previous step from the server container to the server host:

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

2. If not already done, copy the files from the server host to the proxy host:

```
scp config.tar.gz <proxy-FQDN>:/root
```

2.2.4.7. Start the SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

Container can now be started with the mgrpxy command:

Procedure: Start and Check Proxy Status

1. Start the Proxy by calling:

```
mgrpxy start
```

2. Check container status by calling:

```
mgrpxy status
```

Five SUSE Multi-Linux Manager Proxy containers should be present and should be part of the proxy-pod container pod:

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

2.2.4.7.1. Using a Custom Container Image for a Service

By default, the SUSE Multi-Linux Manager Proxy suite is set to use the same image version and registry path for each of its services. However, it is possible to override the default values for a specific service using the install parameters ending with `-tag` and `-image`.

For example, use it like this:

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

It adjusts the configuration file for the httpd service, where `registry.opensuse.org/uyuni/proxy-httpds` is the image to use and `0.1.0` is the version tag, before restarting it.

To reset the values to defaults, run the install command again without those parameters:

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

This command first resets the configuration of all services to the global defaults and then reloads it.

2.2.5. SUSE Multi-Linux Manager 5.2 Beta 2 Proxy Deployment on Kubernetes

2.2.5.1. Proxy on Kubernetes changes

There were multiple changes in how to install SUSE Multi-Linux Manager proxies running on Kubernetes:

- mgrpxy is no longer handling proxies on Kubernetes, helm and the proxy-helm chart need to be used instead.
- The TLS certificates have to be in secrets, rather than in the configuration tarball. This aims at allowing cloud-native TLS certificates management for the proxies.
- The proxy queries the server at the start of the container to verify that the versions are compatible.
- The needed persistent volume claims has been reduced to the squid cache only.
- The SUSE Multi-Linux Manager proxy is supported when running on an RKE2 cluster, K3S is longer supported.

2.2.5.2. Prerequisites

Installing the Kubernetes cluster and configuring it is out of the scope of this document.

The cluster is assumed to be ready to be used with a user having rights on a namespace dedicated to SUSE Multi-Linux Manager.

Create Role and RoleBinding if they do not exist already. The minimum rights required to deploy proxy-helm are defined as:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: example-resource-manager
  namespace: $NAMESPACE
rules:
- apiGroups: [""]
  resources: ["pods", "pods/log", "services", "secrets", "configmaps",
"persistentvolumeclaims"]
  verbs: ["*"]
- apiGroups: ["apps"]
  resources: ["deployments"]
```

```

verbs: ["*"]
- apiGroups: ["networking.k8s.io"]
  resources: ["ingresses"]
  verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: example-resource-manager-binding
  namespace: $NAMESPACE
subjects:
- kind: User
  name: $USERNAME
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: example-resource-manager
  apiGroup: rbac.authorization.k8s.io

```



This guide assumes the reader knows how to work with Kubernetes: the concepts will not be explained here as they are extensively documented in the official Kubernetes documentation.

The SUSE Multi-Linux Manager administrator needs to deploy the proxy-helm Helm chart. However, this chart requires to prepare:

- TLS certificates chain for the proxy,
- a ConfigMap for the proxy root CA certificate,
- a persistent volumes for the claim the chart will create or a storage class automatically creating it,
- Load balancers or other mechanisms to expose the Salt, SSH and TFTP ports.

Run the following command to read the full details on how to use the proxy Helm chart:

```

helm show readme --version 5.2.0-beta2 \
  oci://registry.suse.com/suse/multi-linux-manager/5.2/proxy-helm

```

2.2.5.2.1. Credentials

A secret with the SCC credentials needs to be defined in order to pull the images from registry.suse.com. Refer to <https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/> for the instructions to prepare the secret. Set the registrySecret proxy-helm chart value to the name of the secret containing those credentials to use it.

2.2.5.2.2. TLS setup

The proxy-cert TLS secret is expected. It contains the TLS certificate and key for the Ingress rule and needs to have the public FQDN as Subject Alternate Name.

These secrets can be created using the `kubectl create secret tls -n $NAMESPACE` command. The certificate file passed to this command needs to start with the server certificate followed by the chain of intermediary CA certificates if any. The root CA is not needed in these secrets as it is expected in a ConfigMap.

The Root CA certificate of proxy-cert is expected in a ConfigMap named `uyuni-ca` stored in the `ca.crt` key. It can be created with a command like `kubectl create cm -n $NAMESPACE uyuni-ca --from-file=ca.crt=/path/to/uyuni-ca.crt`.

2.2.5.2.3. Storage

The proxy chart defines a volume as a Persistent Volume Claim (PVC).



- The creation of the underlying PV is the responsibility of the cluster administrators.
- The PVC use the `ReadWriteOnce` access mode.

The created PVC can be tuned Helm chart values, it can have the following values:

- `size`: to set the requested size of the PVC.
- `storageClass`: can be used to select the storage class to use for the PVC.
- `extraLabels`: can be used to add custom labels to the PVC.
- `annotations`: can be used to set custom annotations on the PVC.
- `volumeName`: can be used to hard code which volume the PVC should be bound to.
- `selector`: is the YAML fragment of the PVC selector to use to find the PV to bind to.

Refer to <https://kubernetes.io/docs/concepts/storage/persistent-volumes/> for more information on persistent volumes and their claims.

Refer to the proxy-helm README for the list of persistent volume claims which will be created and will need to be bound to persistent volumes.



While the default sizes are provided, it is highly recommended to change them based on the distributions you plan to synchronize.

For more information on storage requirements see **Installation and Upgrade Guide › General Requirements**.

2.2.5.2.4. Exposing ports

SUSE Multi-Linux Manager proxy requires some TCP and UDP ports to be routed to its services. Refer to the proxy-helm README for the list of ports to be exposed.



RKE2 ships with nginx as the default ingress controller. However, as this is deprecated and soon to be unsupported, the proxy-helm chart defaults to use Traefik as ingress controller. Using the nginx ingress controller might work and will not be documented, use at your own risk.



The proxy-helm chart supports Gateway API version 1.4. Since this requires experimental CRDs which are not shipped with RKE2 1.35, it is not recommended to be used in production.

There are multiple ways to expose the ports, but this documentation will only mention how to configure RKE2's Traefik for this. This is not a task for the SUSE Multi-Linux Manager administrator, but the Kubernetes cluster administrator as it requires configuration to be set on the cluster nodes.

To set Traefik to expose and route the needed ports, create a `/var/lib/rancher/rke2/server/manifests/uyuni-traefik.yaml` on each node with the following content. Note that Traefik takes a few seconds to be reinstalled after saving the file.

```
apiVersion: helm.cattle.io/v1
kind: HelmChartConfig
metadata:
  name: rke2-traefik
  namespace: kube-system
spec:
  valuesContent: |-
    ports:
      ssh:
        port: 8022
        expose:
          default: true
          exposedPort: 8022
          protocol: TCP
          hostPort: 8022
      salt-publish:
        port: 4505
        expose:
          default: true
          exposedPort: 4505
          protocol: TCP
          hostPort: 4505
          containerPort: 4505
      salt-request:
        port: 4506
        expose:
          default: true
          exposedPort: 4506
          protocol: TCP
          hostPort: 4506
          containerPort: 4506
```

If Traefik is used as the Ingress controller, the user needs access to additional resources. Add the following to the rules of the previously defined role:

```
- apiGroups: ["traefik.io", "traefik.containo.us"]
```

```
resources: ["ingressroutetcps"]
verbs: ["*"]
```

If Gateway API is used instead, add the following to the rules of the previously defined role:

```
- apiGroups: ["gateway.networking.k8s.io"]
  resources: ["gateways", "httproutes", "tcproutes"]
  verbs: ["*"]
```

TFTP is complex to expose from a Kubernetes pod due to the nature of the protocol: the TFTP server receives requests on port 69, but negotiates another random port to continue. This port also needs to stay the same through the whole session for the server to recognize the client as being the same. This means that there are only two possible ways to use the TFTP server:

- using a load balancer compatible with TFTP,
- using the host network for the TFTP pod. This can be achieved by setting the `tftp.hostNetwork` helm chart value to true.

2.2.5.3. Configuration generation

Before deploying the SUSE Multi-Linux Manager proxy, a configuration archive needs to be generated.

2.2.5.3.1. Generate the Proxy Configuration with Web UI

Procedure: Generating a Proxy Container Configuration Using Web UI

1. In the Web UI, navigate to **Systems › Proxy Configuration** and fill the required data:
2. In the Proxy FQDN field type fully qualified domain name for the proxy.
3. In the Parent FQDN field type fully qualified domain name for the SUSE Multi-Linux Manager Server or another SUSE Multi-Linux Manager Proxy.
4. In the Proxy SSH port field type SSH port on which SSH service is listening on SUSE Multi-Linux Manager Proxy. Recommended is to keep default 8022.
5. In the 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.



2 GB represents the default proxy squid cache size. This will need to be adjusted for your environment.

6. In the SSL certificate selection list choose if new server certificate should be generated for SUSE Multi-Linux Manager Proxy or an existing one should be used. You can consider generated certificates as SUSE Multi-Linux Manager builtin (self signed) certificates. If SUSE Multi-Linux Manager server runs on Kubernetes, the generated certificate option is not possible and replaced with no SSL certificate as they are managed outside the containers.

Depending on the choice then provide either path to signing CA certificate to generate a new certificate or path to an existing certificate and its key to be used as proxy certificate.

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

For more information about existing or custom certificates and the concept of corporate and intermediate certificates, see **Administration Guide › Ssl Certs Imported**.

7. Click **[Generate]** to register a new proxy FQDN in the SUSE Multi-Linux Manager Server and generate a configuration archive (config.tar.gz) containing details for the container host.
8. After a few moments you are presented with file to download. Save this file locally.

2.2.5.3.2. Generate Proxy Configuration With spacecmd and Self-Signed Certificate

You can generate a Proxy configuration using spacecmd. This is only possible if SUSE Multi-Linux Manager server runs on podman and has a self-signed root CA certificate.

Procedure: Generating Proxy Configuration with spacecmd and Self-Signed Certificate

1. SSH into your container host.
2. Execute the following command replacing the Server and Proxy 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'
```

3. Copy the generated configuration from the server container:

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

2.2.5.3.3. Generate Proxy Configuration With spacecmd and Custom Certificate

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

Procedure: Generating Proxy Configuration with spacecmd and Custom Certificate

1. SSH into your Server container host.
2. 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'
```

3. 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'
```

4. Copy the generated configuration from the server container:

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

2.2.5.3.4. Generate Proxy Configuration With spacecmd and no Certificate

You can generate a Proxy configuration using spacecmd with no TLS certificates. This is needed for SUSE Multi-Linux Manager running on Kubernetes as the certificates are handled outside of the containers.

Procedure: Generating Proxy Configuration with spacecmd and no Certificate

1. SSH into your Server container host.
2. 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_nossl -- -p 8022
pxy.example.com srv.example.com 2048 email@example.com -o /tmp/config.tar.gz'
```

3. Copy the generated configuration from the server container:

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

2.2.5.4. Deploying the SUSE Multi-Linux Manager Proxy Helm Chart

Copy and extract the generated configuration tar.gz file and then install using helm:

```
helm install smlm-proxy \
oci://registry.suse.com/suse/multi-linux-manager/5.2/proxy-helm \
-n $NAMESPACE \
--description "Proxy installation" \
--set "registrySecret=the-scc-secret" \
--set-file global.config=path/to/config.yaml \
--set-file global.ssh=path/to/ssh.yaml \
--set-file global.httpd=path/to/httpd.yaml \
```

When setting multiple values, using a YAML values file is recommended instead of passing several `--set` parameters. Refer to the helm command help for more details.

2.2.5.5. Example helm charts

Some helm charts using the proxy-helm chart can be found in the Manager-5.2 branch of the [uyuni-charts](#) git repository. They show case how the TLS certificate can be generated using cert-manager and trust-manager. Those examples may assume to have Kubernetes cluster administrator permissions.



These examples are not supported, only provided for documentation purpose.

2.2.6. SUSE Multi-Linux Manager Proxy Air-gapped Deployment

2.2.6.1. What is air-gapped deployment?

Air-gapped deployment refers to the setup and operation of any networked system that is physically isolated from insecure networks, especially the internet. This type of deployment is commonly used in high-security environments such as military installations, financial systems, critical infrastructure, and anywhere sensitive data is handled and must be protected from external threats.



At the moment, air-gapped deployment is available only on SL Micro.

2.2.6.2. Deploy with Virtual Machine

The recommended installation method is using the provided SUSE Multi-Linux Manager Virtual Machine Image option, since all the needed tools and container images are pre-loaded and will work out of the box.

For more information about installing SUSE Multi-Linux Manager Proxy Virtual Machine, see [Deploy Proxy as a Virtual Machine](#).

To upgrade SUSE Multi-Linux Manager Proxy, users should follow the procedures defined in [Proxy Upgrade](#).

2.2.6.3. Deploy SUSE Multi-Linux Manager on SL Micro

SUSE Multi-Linux Manager also provides all the needed container images in RPM's that can be installed on the system.

Procedure: Install SUSE Multi-Linux Manager on SL Micro in Air-gapped

1. Install SL Micro.
2. Bootstrap the Proxy Host OS as a Client on SUSE Multi-Linux Manager Server.
3. Update the system.

4. Install tools packages and image packages (replace \$ARCH\$ with the correct architecture)

```
transactional-update pkg install mgrpxy* mgrctl* suse-multi-linux-manager-5.2-  
$ARCH$-proxy-*
```

5. Reboot.
6. Deploy SUSE Multi-Linux Manager with mgrpxy.

For more detailed information about installing SUSE Multi-Linux Manager Proxy on SL Micro, see [Deploy Proxy as a Virtual Machine](#).

To upgrade SUSE Multi-Linux Manager Proxy, users should follow the procedures defined in [Proxy Upgrade](#).

2.2.6.4. Formula images

Some formulas, like Bind and DHCP (Kea), also use containers. If you plan to use them in an air-gapped environment, you need to pull their images, save them to an archive, and load them on your SUSE Multi-Linux Manager Proxy or another managed system.

The images are available from registry.suse.com.

Procedure: Getting formula images for air-gapped environments

1. On a system with Internet access, pull the required images.

```
podman pull registry.suse.com/suse/bind:latest  
podman pull registry.suse.com/suse/kea:2.6
```

2. Save the images to a TAR archive:

```
podman save -m -o formula-images.tar registry.suse.com/suse/bind:latest  
registry.suse.com/suse/kea:2.6
```

3. Transfer the formula-images.tar file to your air-gapped system.
4. Load the images on the air-gapped system:

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

Chapter 3. Upgrade and Migration

3.1. Server

3.1.1. Distribution Upgrade and Server Migration from 5.1 to 5.2



SUSE Multi-Linux Manager 5.1 must be stopped before the upgrade.

SUSE Multi-Linux Manager server hosts that are hardened for security may restrict execution of files from the `/tmp` folder. In such cases, as a workaround, export the `TMPDIR` environment variable to another existing path before running `mgradm`.



For example:

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

In SUSE Multi-Linux Manager updates, tools will be changed to make this workaround unnecessary.

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 X509v3 Subject Alternative Name section of the certificate. You can also list multiple hostnames if your environment requires it. Supported Key types are RSA and EC (Elliptic Curve).



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

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

3.1.1.1. SL Micro 6.1 to SL Micro 6.2

This document provides the tested procedure to upgrade a SL Micro 6.1 host deployed with SUSE Multi-Linux Manager 5.1 Server to SL Micro 6.2 and migrate to SUSE Multi-Linux Manager 5.2 Beta 2.

3.1.1.1.1. Prerequisites

- SUSE Multi-Linux Manager 5.1 is installed and running on SL Micro 6.1.
- System is registered and has active subscriptions with SCC.

3.1.1.1.2. Distribution upgrade and server migration

Procedure: Migration from SUSE Multi-Linux Manager 5.1 to SUSE Multi-Linux Manager 5.2 Beta 2

1. Verify current product status.

```
SUSEConnect --status-text
```

Confirm:

- Base OS: SL Micro 6.1
- Extension: SUSE Multi-Linux Manager Server 5.1 Extension

2. Ensure the system is updated.

```
transactional-update patch
```

- If patches were applied, stop the server and then reboot the system before proceeding to migration:

```
mgradm stop  
reboot
```

- If no updates were found, you can proceed directly to the migration step.

3. Start the migration.

```
transactional-update migration --auto-agree-with-licenses --gpg-auto-import
-keys
```

Follow the prompts and select the available migration to **SUSE Linux Micro 6.2** and **SUSE Multi-Linux Manager Server Extension 5.2 Beta 2**.

4. Stop the server and then reboot to apply changes.

```
mgradm stop
reboot
```

5. Perform post-reboot checks.

Verify SUSE Multi-Linux Manager extension and SUSE Multi-Linux Manager version:

```
SUSEConnect --status-text
mgradm --version
```

Expected output:

- Extension: SUSE Multi-Linux Manager Server 5.2 Beta 2 Extension
- Version: mgradm version 5.2.0 or higher

6. Verify PostgreSQL database version.

```
podman ps
```

Expected output:

- server:5.2.0 or higher
- server-postgresql:5.2.0 or higher

7. Verify running containers:

```
podman ps
```

You should see all the expected server containers are up and running.

3.1.1.1.3. Migration complete

The server host system is now running SL Micro 6.2 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Server packages.

If you have a SUSE Multi-Linux Manager 5.1 proxy connected to this server, proceed to the **Installation and Upgrade Guide › Container Deployment › Proxy Migration 5.1 > 5.2** guide to upgrade the proxy host.

Validate your setup before resuming production operations.

3.1.1.1.4. Database Backup Volume

Server migration or upgrade with `mgradm migration` or `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 `mgradm` dynamically creates the volume `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.

3.1.1.2. SUSE Linux Enterprise Server 15 SP7

This document provides the procedure to upgrade a SUSE Linux Enterprise Server 15 SP7 host deployed with SUSE Multi-Linux Manager 5.1 Server to SUSE Multi-Linux Manager 5.2 Beta 2.

3.1.1.2.1. Prerequisites

- SUSE Multi-Linux Manager 5.1 is installed and running on SUSE Linux Enterprise Server 15 SP7.
- System is registered and has active subscriptions with SCC.

3.1.1.2.2. Server package update and migration

Procedure: Update SUSE Multi-Linux Manager components on SUSE Linux Enterprise Server 15 SP7

1. Verify current product status.

```
SUSEConnect --status-text
```

Confirm:

- Base OS: SUSE Linux Enterprise Server 15 SP7
- Extension: SUSE Multi-Linux Manager Server 5.1 Extension

2. Ensure the system is updated.

```
zypper patch
```

If patches were applied, stop the server and then reboot before proceeding:

```
mgradm stop  
reboot
```

3. Perform the migration to SUSE Multi-Linux Manager 5.2 Beta 2.

```
zypper migration
```

Select to migrate to:

- SUSE Multi-Linux Manager Server Extension 5.2 Beta 2

4. Stop the server and reboot.

```
mgradm stop  
reboot
```

5. Perform post-reboot checks.

Verify SUSE Multi-Linux Manager extension:

```
SUSEConnect --status-text
```

Expected output:

- Extension: SUSE Multi-Linux Manager Server 5.2 Beta 2 Extension

6. Verify SUSE Multi-Linux Manager version.

```
mgradm --version
```

Expected output:

- Version: mgradm version 5.2.0 or higher

7. Verify containers:

```
podman ps
```

Expected output:

- server:5.2.0 or higher
- server-postgresql:5.2.0 or higher

3.1.1.2.3. Migration complete

The server host system is now running SUSE Linux Enterprise Server 15 SP7 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Server packages.

If you have a SUSE Multi-Linux Manager 5.1 proxy connected to this server, proceed to the **Installation and Upgrade Guide › Container Deployment › Proxy Migration 5.1 > 5.2** guide to upgrade the proxy host.

Validate your setup before resuming production operations.

3.1.1.2.4. Database Backup Volume

Server migration or upgrade with mgradm migration or 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 mgradm dynamically creates the volume 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.

3.1.2. Distribution Upgrade and Server Migration from 5.0 to 5.2



SUSE Multi-Linux Manager 5.0 must be stopped before the upgrade.



SUSE Multi-Linux Manager server hosts that are hardened for security may restrict execution of files from the /tmp folder. In such cases, as a workaround, export the TMPDIR environment variable to another existing path before running mgradm.

For example:

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

In SUSE Multi-Linux Manager updates, tools will be changed to make this workaround unnecessary.

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 X509v3 Subject Alternative Name section of the certificate. You can also list multiple hostnames if your environment requires it. Supported Key types are RSA and EC (Elliptic Curve).



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

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

SUSE Multi-Linux Manager 5.0 peripheral servers are always using third-party SSL certificates. If the hub server has generated the certificates for the peripheral server, it needs to generate the certificate for the peripheral database too.

On the hub server, run the following command for each of the peripheral server to migrate.



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

The files to use will be inside the server container and need to be copied to the new peripheral server host:

1. /root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT as the root CA,
2. /root/ssl-build/<hostname>-reportdb/server.crt as the db certificate and
3. /root/ssl-build/<hostname>-reportdb/server.key as the db certificate's key.

3.1.2.1. SLE Micro 5.5 to SL Micro 6.2

This document provides the tested procedure to upgrade a SLE Micro 5.5 host deployed with SUSE Multi-Linux Manager 5.0 Server to SL Micro 6.2 and migrate to SUSE Multi-Linux Manager 5.2 Beta 2.

3.1.2.1.1. Prerequisites

- SUSE Multi-Linux Manager 5.0 is installed and running on SLE Micro 5.5.
- System is registered and has active subscriptions with SCC.

3.1.2.1.2. Distribution upgrade and server migration

Procedure: Migration from SUSE Multi-Linux Manager 5.0 to SUSE Multi-Linux Manager 5.2 Beta 2

1. Verify current product status.

```
SUSEConnect --status-text
```

Confirm:

- Base OS: SUSE Linux Enterprise Micro 5.5
- Extension: SUSE Manager Server 5.0 Extension

2. Ensure the system is updated.

```
transactional-update patch
```

- If patches were applied, stop the server and then reboot the system before proceeding to migration:

```
mgradm stop
reboot
```

- If no updates were found, you can proceed directly to the migration step.

3. Start the migration.

```
transactional-update migration --auto-agree-with-licenses --gpg-auto-import
-keys
```

Follow the prompts and select the available migration to **SUSE Linux Micro 6.2** and **SUSE Multi-Linux Manager Server Extension 5.2 Beta 2**.

4. Stop the server and then reboot to apply changes.

```
mgradm stop
reboot
```

5. Perform post-reboot checks.

Verify SUSE Multi-Linux Manager extension and SUSE Multi-Linux Manager version:

```
SUSEConnect --status-text
mgradm --version
```

Expected output:

- Extension: SUSE Multi-Linux Manager Server 5.2 Beta 2 Extension
- Version: referencing 5.2.0 or higher

6. Verify PostgreSQL database version.

```
podman ps
```

Expected output:

- server:5.2.0 or higher
- server-postgresql:5.2.0 or higher

7. Verify running containers:

```
podman ps
```

You should see all the expected server containers are up and running.

3.1.2.1.3. Migration complete

The server host system is now running SL Micro 6.2 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Server packages.

If you have a SUSE Multi-Linux Manager 5.0 proxy connected to this server, proceed to the **Installation and Upgrade Guide › Container Deployment › Proxy Migration 5.0 > 5.2** guide to upgrade the proxy host.

Validate your setup before resuming production operations.

3.1.2.1.4. Database Backup Volume

Server migration or upgrade with `mgradm migration` or `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 `mgradm` dynamically creates the volume `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.

3.1.2.2. SUSE Linux Enterprise Server 15 SP6 to 15 SP7

This document provides the procedure to upgrade a SUSE Linux Enterprise Server 15 SP6 host deployed with SUSE Multi-Linux Manager 5.0 Server to SUSE Linux Enterprise Server 15 SP7 and migrate to SUSE Multi-Linux Manager 5.2 Beta 2.

3.1.2.2.1. Prerequisites

- SUSE Multi-Linux Manager 5.0 is installed and running on SUSE Linux Enterprise Server 15 SP6.
- System is registered and has active subscriptions with SCC.

3.1.2.2.2. Distribution upgrade and server migration

Procedure: Upgrade and Migrate SUSE Multi-Linux Manager on SUSE Linux Enterprise Server 15 SP6

1. Verify current product status.

```
SUSEConnect --status-text
```

Confirm:

- Base OS: SUSE Linux Enterprise Server 15 SP6
- Extension: SUSE Manager Server 5.0 Extension

2. Ensure the system is updated.

```
zypper patch
```

If patches were applied, stop the server and then reboot before proceeding:

```
mgradm stop  
reboot
```

3. Perform the migration to SP7 and SUSE Multi-Linux Manager 5.2 Beta 2.

```
zypper migration
```

Select to migrate to:

- SUSE Linux Enterprise Server 15 SP7
- SUSE Multi-Linux Manager Server Extension 5.2 Beta 2

4. Stop the server and reboot.

```
mgradm stop  
reboot
```

5. Perform post-reboot checks.

Verify SUSE Multi-Linux Manager extension:

```
SUSEConnect --status-text
```

Expected output:

- Extension: SUSE Multi-Linux Manager Server 5.2 Beta 2 Extension

6. Verify SUSE Multi-Linux Manager version.

```
mgradm --version
```

Expected output:

- Version: referencing 5.2.0 or higher

7. Verify containers:

```
podman ps
```

Expected output:

- server:5.2.0 or higher
- server-postgresql:5.2.0 or higher

3.1.2.2.3. Migration complete

The server host system is now running SUSE Linux Enterprise Server 15 SP7 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Server packages.

If you have a SUSE Multi-Linux Manager 5.0 proxy connected to this server, proceed to the **Installation and Upgrade Guide › Container Deployment › Proxy Migration 5.0 > 5.2** guide to upgrade the proxy host.

Validate your setup before resuming production operations.

3.1.2.2.4. Database Backup Volume

Server migration or upgrade with `mgradm migration` or `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 `mgradm` dynamically creates the volume `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.

3.1.3. SUSE Multi-Linux Manager Server Upgrade

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 SUSE Multi-Linux Manager tooling such as the `mgradm` tool.

Procedure: Upgrading Server

1. Refresh software repositories with `zypper`:

```
zypper ref
```

2. Depending on the host operating system, proceed with these steps:

For a transactional system such as SL Micro:

1. Apply available updates with `transactional-update`:

```
transactional-update
```

2. If updates were applied, reboot.

For SUSE Linux Enterprise Server:

Update installed software with `zypper`:

```
zypper up
```

3. The SUSE Multi-Linux Manager Server container can be updated using the following command:



Risk of Automated Version Downgrade and PTF Loss

Running the `mgradm upgrade podman` command when no newer upgrade is available will cause the system to automatically revert to the base version. This process removes all currently applied Program Temporary Fixes (PTFs) without a

confirmation prompt.

To avoid unintended data or fix loss, verify upgrade availability before execution. Future releases will include a confirmation prompt to prevent this behavior.

+

```
mgradm upgrade podman
```

+

This command will bring the status of the container up-to-date and restart the server.

+

1. Clean up the unused container images to free disk space:

```
podman image prune -a
```

Upgrading with third-party SSL certificate

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

- db
- 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 `--ssl-db-ca-root`, `--ssl-db-cert` and `--ssl-db-key` parameters.

Upgrading to specific version



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.

For more information on the upgrade command and its parameters, use the following command:



Risk of Automated Version Downgrade and PTF Loss

Running the `mgradm upgrade podman` command when no newer upgrade is available will cause the system to automatically revert to the base version. This process removes all currently applied Program Temporary Fixes (PTFs) without a confirmation prompt.

To avoid unintended data or fix loss, verify upgrade availability before execution. Future releases will include a confirmation prompt to prevent this behavior.

```
mgradm upgrade podman -h
```

For air-gapped installations, first upgrade the container RPM packages, then run the `mgradm` command.

3.1.3.1. Database Backup Volume

Server migration or upgrade with `mgradm migration` or `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 `mgradm` dynamically creates the volume `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.

3.2. Proxy

3.2.1. Proxy Migration from 5.1 to 5.2

3.2.1.1. Introduction

This document provides the tested and validated procedures for migrating the **host operating system** and the **proxy extension** in environments managed by **SUSE Multi-Linux Manager**, specifically targeting systems deployed with **SUSE Multi-Linux Manager Proxy 5.1**.

The upgrade scenarios covered include:

- Migrating from **SL Micro 6.1** to **SL Micro 6.2**
- Updating **SLES 15 SP7** with the new **SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2**

- Upgrading the **SUSE Multi-Linux Manager Proxy Extension** from **version 5.1** to **version 5.2 Beta 2**



Before migrating the proxy, it is required to first migrate the SUSE Multi-Linux Manager 5.1 Server to SUSE Multi-Linux Manager 5.2 Beta 2.

3.2.1.2. SL Micro 6.1 to SL Micro 6.2

This section provides the tested procedure to upgrade a SL Micro 6.1 host deployed with SUSE Multi-Linux Manager 5.1 Proxy to SL Micro 6.2 with SUSE Multi-Linux Manager 5.2 Beta 2 Proxy.

3.2.1.2.1. Prerequisites

- SUSE Multi-Linux Manager 5.1 Proxy is installed and running on SL Micro 6.1.
- Proxy system is registered with the SUSE Multi-Linux Manager Server.

3.2.1.2.2. Distribution upgrade and proxy migration

Procedure: Migrate SUSE Multi-Linux Manager 5.1 Proxy to SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

1. Verify System and SUSE Multi-Linux Manager Tools version.

```
cat /etc/os-release  
mgrpxy --version
```

Confirm:

- Operating System: SL Micro 6.1
- Tools version: mgrpxy version 5.1.x or higher

2. Check running containers.

```
podman ps
```

Ensure the following containers are running:

- proxy-squid
- proxy-ssh

- proxy-httpd
- proxy-tftpd
- proxy-salt-broker

3. Synchronize the new Proxy Products in SUSE Multi-Linux Manager Server.
For more information, see **Client Configuration Guide › Products**.

Use the Web UI to synchronize:

- SL Micro 6.2
- SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2

4. Perform proxy product migration. For more information, see **Client Configuration Guide › Client Upgrades Product Migration**.

Navigate to the proxy system and select **Systems › Overview › Software › Product Migration**.

Migrate from

- SL Micro 6.1 + SUSE Multi-Linux Manager Proxy 5.1 Extension

to

- SL Micro 6.2 + SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2



Do not select optional channels when prompted, unless you have confirmed they are required.



It is recommended to do a dry-run first before performing the actual migration.

5. Monitor the migration action.

You can follow the process under **Systems › Details › Events** tab in the Web UI.

6. After the upgrade completes, stop the proxy container and then reboot the system.

```
mgrpky stop
reboot
```

7. Perform post-reboot checks.

Verify upgraded OS and SUSE Multi-Linux Manager extension:

```
cat /etc/os-release
SUSEConnect --status-text
```

8. Verify SUSE Multi-Linux Manager tools version.

```
mgrpky --version
```

Expected output:

- mgrpky version 5.2.0 or higher

9. Install the new proxy container images as RPM packages if they are not pulled from the registry.

```
transactional-update pkg install suse-multi-linux-manager-5.2-<arch>-proxy*
```

10. Reboot the Proxy.
11. Upgrade proxy containers and restart them.

```
mgrpky upgrade podman
mgrpky stop
mgrpky start
```

12. Confirm proxy containers are operational.

```
podman ps
```

All expected proxy containers should be up and running:

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

3.2.1.2.3. Migration complete

The proxy host system is now running SL Micro 6.2 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Proxy packages and synchronized product channels.

3.2.1.3. SUSE Linux Enterprise Server 15 SP7

This section provides the procedure to update a SUSE Linux Enterprise Server 15 SP7 host deployed with SUSE Multi-Linux Manager 5.1 Proxy to SUSE Multi-Linux Manager 5.2 Beta 2 Proxy.

3.2.1.3.1. Prerequisites

- SUSE Multi-Linux Manager Proxy 5.1 is installed and running on SUSE Linux Enterprise Server 15 SP7.
- Proxy system is registered with the SUSE Multi-Linux Manager Server.

3.2.1.3.2. Proxy component update

Procedure: Update SUSE Multi-Linux Manager Proxy components on SUSE Linux Enterprise Server 15 SP7

1. Verify operating system and SUSE Multi-Linux Manager tools version.

```
cat /etc/os-release  
mgrpxy --version
```

Confirm:

- Operating System: SUSE Linux Enterprise Server 15 SP7
- Tools version: mgrpxy version 5.1.x or higher

2. List running proxy containers.

```
podman ps
```

Ensure the following containers are running:

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

3. Synchronize the new Products in SUSE Multi-Linux Manager Server using the Web UI.

For more information, see **Client Configuration Guide › Products**.

Synchronize:

- SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2

4. Perform proxy product migration.

For more information, see **Client Configuration Guide › Client Upgrades Product Migration**.

Navigate to the proxy system and select **Systems › Overview › Software › Product Migration**.

Migrate from:

- SUSE Linux Enterprise Server 15 SP7 + SUSE Multi-Linux Manager Proxy 5.1 Extension

to:

- SUSE Linux Enterprise Server 15 SP7 + SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2



Do not select optional channels when prompted, unless you have confirmed they are required.



It is recommended to do a dry-run first before performing the actual migration.

5. Monitor the migration action.

You can follow the process under **Systems › Details › Events** in the Web UI.

6. After the upgrade completes, stop the proxy container and then reboot the system.

```
mgrpky stop
reboot
```

7. Perform post-reboot checks.

Verify the OS and SUSE Multi-Linux Manager extension version:

```
cat /etc/os-release
SUSEConnect --status-text
```

8. Verify SUSE Multi-Linux Manager tools version.

```
mgrpky --version
```

Expected output:

- mgrpky version 5.2.0 or higher

9. Install the new proxy container images as RPM packages if they are not pulled from the registry.

```
zypper install suse-multi-linux-manager-5.2-<arch>-proxy*
```

10. Reboot the Proxy.

11. Upgrade proxy containers and restart them.

```
mgrpky upgrade podman
mgrpky stop
mgrpky start
```

12. Confirm proxy containers are operational.

```
podman ps
```

All expected proxy containers should be up and running:

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

3.2.1.3.3. Migration complete

The proxy host system is now running SUSE Linux Enterprise Server 15 SP7 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Proxy packages and synchronized product channels.

3.2.2. Proxy Migration from 5.0 to 5.2

3.2.2.1. Introduction

This document provides the tested and validated procedures for migrating both the **host operating system** and the **proxy extension** in environments managed by **SUSE Multi-Linux Manager**, specifically targeting systems deployed with **SUSE Multi-Linux Manager Proxy 5.0**.

The upgrade scenarios covered include:

- Migrating from **SUSE Linux Enterprise Micro (SLE Micro) 5.5** to **SL Micro 6.2**
- Migrating from **SUSE Linux Enterprise Server (SLES) 15 SP6** to **SLES 15 SP7**
- Upgrading the **SUSE Multi-Linux Manager Proxy Extension** from **version 5.0** to **version 5.2 Beta 2**



Before migrating the proxy, it is required to first migrate the SUSE Manager 5.0 Server to

3.2.2.2. SLE Micro 5.5 to SL Micro 6.2

This section provides the tested procedure to upgrade a SLE Micro 5.5 host deployed with SUSE Multi-Linux Manager 5.0 Proxy to SL Micro 6.2 with SUSE Multi-Linux Manager 5.2 Beta 2 Proxy.

3.2.2.2.1. Prerequisites

- SUSE Multi-Linux Manager 5.0 Proxy is installed and running on SLE Micro 5.5.
- Proxy system is registered with the SUSE Multi-Linux Manager Server.

3.2.2.2.2. Distribution upgrade and proxy migration

Procedure: Migrate SUSE Multi-Linux Manager 5.0 Proxy to SUSE Multi-Linux Manager 5.2 Beta 2 Proxy

1. Verify System and SUSE Multi-Linux Manager Tools version.

```
cat /etc/os-release  
mgrpxy --version
```

Confirm:

- Operating System: SLE Micro 5.5
- Tools version: mgrpxy version 0.1.29 or higher

2. Check running containers.

```
podman ps
```

Ensure the following containers are running:

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

- proxy-salt-broker

3. Synchronize the new Proxy Products in SUSE Multi-Linux Manager Server.
For more information, see **Client Configuration Guide › Products**.

Use the Web UI to synchronize:

- SL Micro 6.2
- SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2

4. Perform proxy product migration. For more information, see **Client Configuration Guide › Client Upgrades Product Migration**.

Navigate to the proxy system and select **Systems › Overview › Software › Product Migration**.

Migrate from

- SLE Micro 5.5 + SUSE Manager Proxy 5.0 Extension

to

- SL Micro 6.2 + SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2



Do not select optional channels when prompted, unless you have confirmed they are required.



It is recommended to do a dry-run first before performing the actual migration.

5. Monitor the migration action.

You can follow the process under **Systems › Details › Events** tab in the Web UI.

6. After the upgrade completes, stop the proxy container and then reboot the system.

```
reboot
mgrpxy stop
```

7. Perform post-reboot checks.

Verify upgraded OS and SUSE Multi-Linux Manager extension:

```
cat /etc/os-release
SUSEConnect --status-text
```

8. Verify SUSE Multi-Linux Manager tools version.

```
mgrpxy --version
```

Expected output:

- mgrpxy version 5.2 or higher

9. Enable Root SSH Access (if required). SL Micro 6.2 disables root login via SSH by default. Edit `/etc/ssh/sshd_config.d/sshd.conf`:

```
PermitRootLogin yes
```

Restart the service:

```
systemctl restart sshd
```

For more information, see **Administration Guide › Troubleshooting › Tshoot Remote Root On Micro**.

10. Install the new proxy container images as RPM packages.

```
transactional-update pkg install suse-multi-linux-manager-5.2-<arch>-proxy*
```

11. Reboot the Proxy.

12. Upgrade proxy containers and restart them.

```
mgrpxy upgrade podman
mgrpxy stop
```

```
mgrpky start
```

13. Confirm proxy containers are operational.

```
podman ps
```

All expected proxy containers should be up and running:

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

3.2.2.2.3. Migration complete

The proxy host system is now running SL Micro 6.2 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Proxy packages and synchronized product channels.

3.2.2.3. SUSE Linux Enterprise Server 15 SP6 to 15 SP7

This section provides the procedure to upgrade a SUSE Linux Enterprise Server SP6 host deployed with SUSE Multi-Linux Manager 5.0 Proxy to SUSE Linux Enterprise Server SP7 with SUSE Multi-Linux Manager 5.2 Beta 2 Proxy.

3.2.2.3.1. Prerequisites

- SUSE Multi-Linux Manager Proxy 5.0 is installed and running on SUSE Linux Enterprise Server 15 SP6.
- Proxy system is registered with the SUSE Multi-Linux Manager Server.

3.2.2.3.2. Distribution upgrade and proxy migration

Procedure: Update SUSE Multi-Linux Manager Proxy components on SUSE Linux Enterprise Server 15 SP6

1. Verify operating system and SUSE Multi-Linux Manager tools version.

```
cat /etc/os-release
mgrpxy --version
```

Confirm:

- Operating System: SUSE Linux Enterprise Server 15 SP6
- Tools version: mgrpxy version 0.1.29 or higher

2. List running proxy containers.

```
podman ps
```

Ensure the following containers are running:

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

3. Synchronize the new Products in SUSE Multi-Linux Manager Server using the Web UI.

For more information, see **Client Configuration Guide › Products**.

Synchronize:

- SUSE Linux Enterprise Server SP7
- SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2

4. Perform proxy product migration.

For more information, see **Client Configuration Guide › Client Upgrades Product Migration**.

Navigate to the proxy system and select **Systems › Overview > Software ›**

Product Migration.

Migrate from:

- SUSE Linux Enterprise Server 15 SP6 + SUSE Manager Proxy 5.0 Extension

to:

- SUSE Linux Enterprise Server SP7 + SUSE Multi-Linux Manager Proxy Extension 5.2 Beta 2



Do not select optional channels when prompted, unless you have confirmed they are required.



It is recommended to do a dry-run first before performing the actual migration.

5. Monitor the migration action.

You can follow the process under **Systems › Details › Events** in the Web UI.

6. After the upgrade completes, stop the proxy container and then reboot the system.

```
mgrpxy stop
reboot
```

7. Perform post-reboot checks.

Verify the OS and SUSE Multi-Linux Manager extension version:

```
cat /etc/os-release
SUSEConnect --status-text
```

8. Verify SUSE Multi-Linux Manager tools version.

```
mgrpky --version
```

Expected output:

- mgrpky version 5.2 or higher

9. Install the new proxy container images as RPM packages.

```
zypper install suse-multi-linux-manager-5.2-<arch>-proxy*
```

10. Reboot the Proxy.

11. Upgrade proxy containers and restart them.

```
mgrpky upgrade podman
mgrpky stop
mgrpky start
```

12. Confirm proxy containers are operational.

```
podman ps
```

All expected proxy containers should be up and running:

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

3.2.2.3.3. Migration complete

The proxy host system is now running SUSE Linux Enterprise Server SP7 with updated SUSE Multi-Linux Manager 5.2 Beta 2 Proxy packages and synchronized product channels.

3.2.3. SUSE Multi-Linux Manager Proxy Upgrade

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 SUSE Multi-Linux Manager tooling such as the mgrpxy tool.

Procedure: Upgrading Proxy

1. Refresh software repositories with zypper:

```
zypper ref
```

2. Depending on the host operating system, proceed with these steps:

For a transactional system such as SL Micro:

1. Apply available updates with transactional-update:

```
transactional-update
```

2. If updates were applied, reboot.

For SUSE Linux Enterprise Server:

Update installed software with zypper:

```
zypper up
```

3. The SUSE Multi-Linux Manager Proxy containers running on podman can be updated using the following command:

```
mgrpxy upgrade podman
```

Or, those running on a Kubernetes cluster can update using:

```
mgrpxy upgrade kubernetes
```

4. On podman, clean up the unused container images to free disk space:

```
podman image prune -a
```

On Kubernetes the image cleanup is handled automatically, or it depends on the Kubernetes distribution.



If you do not specify the tag parameter when upgrading to specific version, 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.

While there is an option to upgrade a specific container using its specific tag, this feature is intended for applying PTFs only.

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

For air-gapped installations, first upgrade the container RPM packages, then run the `mgrpxy upgrade podman` command.

3.3. Clients

3.3.1. Upgrade Clients

Clients use the versioning system of their underlying operating system. For clients using SUSE operating systems, you can perform upgrades within the SUSE Multi-Linux Manager Web UI.

For more information about upgrading clients, see **Client Configuration Guide › Client Upgrades**.

Chapter 4. Basic Server and Proxy Management

4.1. Custom YAML Configuration and Deployment with mgradm

You have the option to create a custom `mgradm.yaml` file, which the `mgradm` tool can utilize during deployment. All `mgradm` arguments have their YAML counterparts. For example, the `mgradm` argument `--ssl-db-ca-intermediate` can be specified in the `mgradm.yaml` file as follows:

```
ssl:
  db:
    ca:
      intermediate: /path/to/ca-intermediate.crt
```



mgradm will prompt for basic variables if they are not provided using command line parameters or the `mgradm.yaml` configuration file.

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

Procedure: Deploying the SUSE Multi-Linux Manager Container with Podman Using a Custom Configuration File

1. Prepare a configuration file named `mgradm.yaml` similar to the following example:

```
# Database password. Randomly generated by default
db:
  password: MySuperSecretDBPass

# Password for the CA certificate
ssl:
  password: MySuperSecretSSLPassword

# Your SUSE Customer Center credentials
scc:
  user: ccUsername
  password: ccPassword

# Organization name
organization: YourOrganization

# Email address sending the notifications
emailFrom: notifications@example.com

# Administrators account details
admin:
  password: MySuperSecretAdminPass
  login: LoginName
  firstName: Admin
  lastName: Admin
  email: email@example.com
```

2. From the terminal, as root, run the following command. Entering your server's FQDN is optional.

```
mgradm -c mgradm.yaml install podman <FQDN>
```



You must deploy the container as sudo or root. The following error will be displayed on the terminal if you miss this step.

```
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"
```

3. Wait for deployment to complete.
4. Open a browser and proceed to your server's FQDN or IP address.

4.2. Starting and Stopping Containers

The SUSE Multi-Linux Manager 5.2 Beta 2 Server container can be restarted, started, and stopped using the following commands:

To restart the SUSE Multi-Linux Manager 5.2 Beta 2 Server execute the following command:

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

To start the server execute the following command:

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

To stop the server execute the following command:

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

4.3. Containers used by SUSE Multi-Linux Manager

Below is a list of containers used by SUSE Multi-Linux Manager 5.2 Beta 2.

Table 14. Server Containers

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

Table 15. Proxy Containers

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

4.4. Persistent Container Volumes

Modifications performed within containers are not retained. Any alterations made outside of persistent volumes will be discarded. Below is a list of persistent volumes for SUSE Multi-Linux Manager 5.2 Beta 2.

To customize the default volume locations, ensure you create the necessary volumes before launching the pod for the first time, utilizing the podman volume create command.



Ensure that this table aligns precisely with the volumes mapping outlined in both the Helm chart and the systemctl services definitions.

4.4.1. Server

The following volumes are stored under the **Podman** default storage location on the server.

Table 16. Persistent Volumes: Podman Default Storage

Volume Name	Volume Directory
Podman Storage	/var/lib/containers/storage/volumes/

Table 17. Persistent Volumes: root

Volume Name	Volume Directory
root	/root

Table 18. Persistent Volumes: var/

Volume Name	Volume Directory
var-cobbler	/var/lib/cobbler
var-salt	/var/lib/salt
var-pgsql	/var/lib/pgsql/data
var-pgsql-backup	/var/lib/pgsql-backup
var-cache	/var/cache
var-spacewalk	/var/spacewalk
var-log	/var/log

Table 19. Persistent Volumes: srv/

Volume Name	Volume Directory
srv-salt	/srv/salt
srv-www	/srv/www/
srv-tftpboot	/srv/tftpboot
srv-formulametadata	/srv/formula_metadata
srv-pillar	/srv/pillar
srv-susemanager	/srv/susemanager
srv-spacewalk	/srv/spacewalk

Table 20. Persistent Volumes: etc/

Volume Name	Volume Directory
etc-apache2	/etc/apache2
etc-rhn	/etc/rhn
etc-systemd-multi	/etc/systemd/system/multi-user.target.wants
etc-systemd-sockets	/etc/systemd/system/sockets.target.wants
etc-salt	/etc/salt
etc-sssd	/etc/sssd
etc-tomcat	/etc/tomcat
etc-cobbler	/etc/cobbler
etc-sysconfig	/etc/sysconfig
etc-postfix	/etc/postfix
ca-cert	/etc/pki/trust/anchors

Table 21. Persistent Volumes: run/

Volume Name	Volume Directory
run-salt-master	/run/salt/master

4.4.2. Proxy

The following volumes are stored under the **Podman** default storage location on the proxy.

Table 22. Persistent Volumes: Podman Default Storage

Volume Name	Volume Directory
Podman Storage	/var/lib/containers/storage/volumes/

Table 23. Persistent Volumes: srv/

Volume Name	Volume Directory
uyuni-proxy-tftpboot	/srv/tftpboot

Table 24. Persistent Volumes: var/

Volume Name	Volume Directory
uyuni-proxy-rhn-cache	/var/cache/rhn
uyuni-proxy-squid-cache	/var/cache/squid

4.5. Understanding mgr-storage-server and mgr-storage-proxy

mgr-storage-server and mgr-storage-proxy are helper scripts provided with SUSE Multi-Linux Manager 5.0 and later.

They are designed to configure storage for SUSE Multi-Linux Manager Server and Proxy.

The scripts take disk devices as arguments. mgr-storage-proxy requires a single argument for the storage disk device. 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.

4.5.1. What these tools do

Both mgr-storage-server and 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 /etc/fstab so that the storage mounts automatically on boot.
- Remount the devices at their final locations.

Table 25. Additional tool-specific behavior

mgr-storage-server	<p>Optionally supports a separate device for database storage.</p> <p>Stops SUSE Multi-Linux Manager services during migration, restarts them afterward.</p> <p>Moves Podman volumes directory /var/lib/containers/storage/volumes to the prepared storage, and optionally /var/lib/containers/storage/volumes/var-pgsql to the prepared database storage.</p>
--------------------	--

mgr-storage-proxy	<p>Focuses only on proxy storage (no database storage support).</p> <p>Stops and restarts the proxy service during migration.</p> <p>Moves podman volumes directory /var/lib/containers/storage/volumes to the prepared storage.</p>
-------------------	--



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

4.5.2. 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.

4.5.3. Post-installation storage management

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

4.5.3.1. Examples

Listing 1. 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

1. Add and format the new disk.
2. Mount it temporarily.
3. Use rsync to copy data.
4. Update /etc/fstab.
5. Remount at the correct location.

4.5.4. When to use, or not use



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., `lvextend`, `xfs_growfs`) for such operations.

4.5.5. Summary

`mgr-storage-server` and `mgr-storage-proxy` help automate the initial persistent storage setup for SUSE Multi-Linux Manager 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.



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 `/var/lib/containers/storage/volumes/`.

Chapter 5. GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

-
- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
 - B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
 - C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these

sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other

respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".