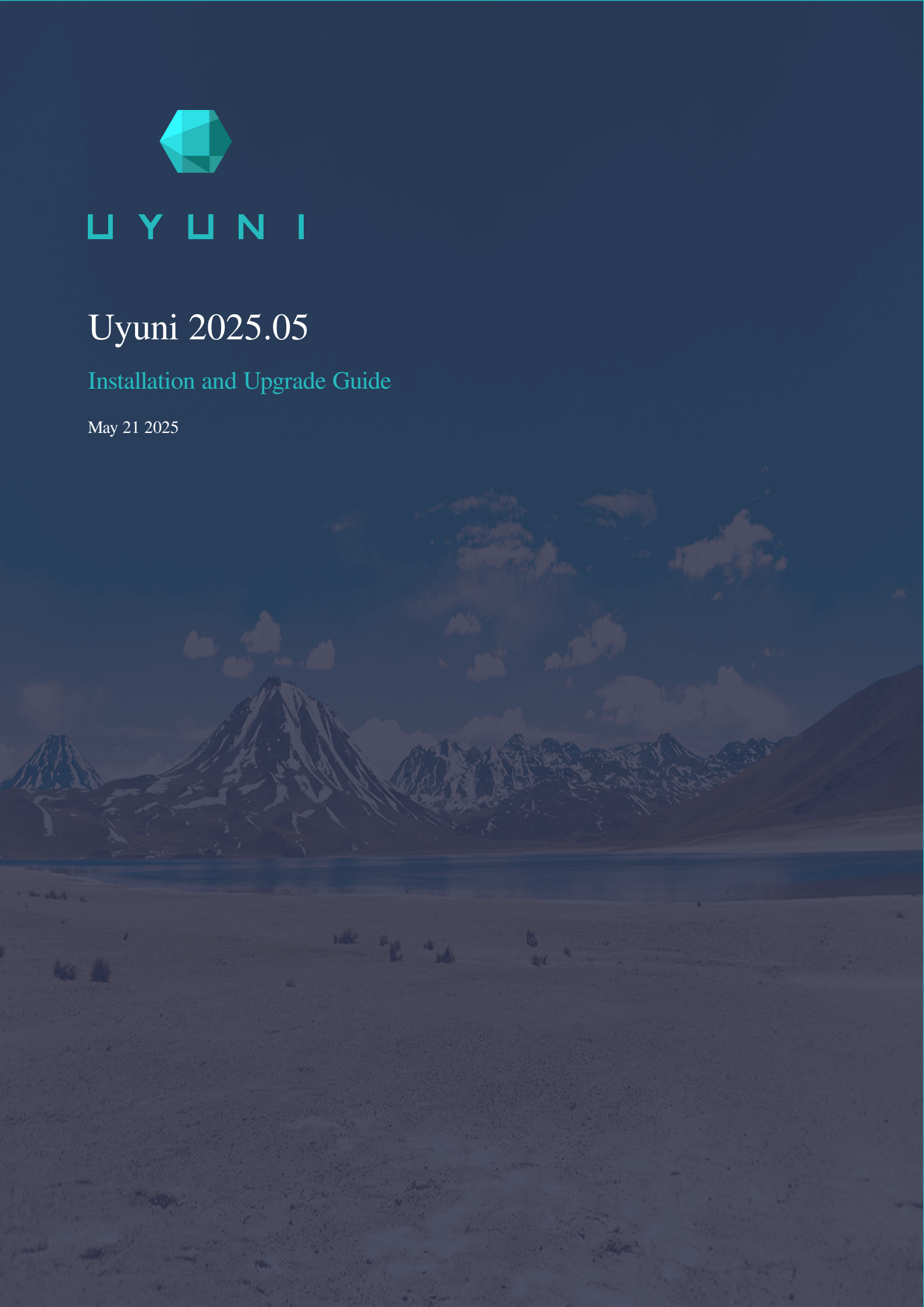


Uyuni 2025.05

Installation and Upgrade Guide

May 21 2025

Chapter 1. Preface

Installation, Deployment and Upgrade
Uyuni 2025.05

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

It is organized into the following sections:

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

Publication Date: 2025-05-21

Contents

Chapter 2. Requirements

2.1. General Requirements

The following tables specify the minimum server and proxy requirements.

Do not use NFS for storage because it does not support SELinux file labeling.

2.1.1. Server Requirements

Table 1. Server Requirements for x86-64 Architecture

Software and Hardware	Details	Recommendation
openSUSE Leap Micro 6.1	Clean installation, up-to-date	openSUSE Leap Micro 6.1
CPU	-	Minimum 4 dedicated 64-bit CPU cores (x86-64)
RAM	Test or Base Installation	Minimum 16 GB
	Production Server	Minimum 32 GB
Disk Space	`/` (root directory)	Minimum 40 GB
	`/var/lib/pgsql`	Minimum 50 GB
	`/var/spacewalk`	Minimum storage required: 100 GB (this will be verified by the implemented check) * 50 GB for each SUSE product and Package Hub * 360 GB for each Red Hat product
	`/var/cache`	Minimum 10 GB. Add 100 MB per SUSE product, 1 GB per Red Hat or other product. Double the space if the server is an ISS Master.
	Swap space	3 GB

2.1.2. Proxy Requirements

Table 2. Proxy Requirements

Software and Hardware	Details	Recommendation
openSUSE Leap Micro 6.1	Clean installation, up-to-date	openSUSE Leap Micro 6.1
CPU		Minimum 2 dedicated 64-bit CPU cores
RAM	Test Server	Minimum 2 GB
	Production Server	Minimum 8 GB
Disk Space	`/` (root directory)	Minimum 40 GB
	`/srv`	Minimum 100 GB
	`/var/cache` (Squid)	Minimum 100 GB

Uyuni Proxy caches packages in the `/var/cache/` directory. If there is not enough space available in `/var/cache/`, the proxy will remove old, unused packages and replace them with newer packages.

As a result of this behavior:

- The larger `/var/cache/` directory is on the proxy, the less traffic there will be between it and the Uyuni Server.
- By making the `/var/cache/` directory on the proxy the same size as `/var/spacewalk/` on the Uyuni Server, you avoid a large amount of traffic after the first synchronization.
- The `/var/cache/` directory can be small on the Uyuni Server compared to the proxy. For a guide to size estimation, see the [server-hardware-requirements] section.

2.2. Network Requirements

This section details the networking and port requirements for Uyuni.

IP forwarding will be enabled by containerized installation. This means Uyuni Server and Proxies will behave as a router. This behavior is done by podman directly. podman containers do not run if IP forwarding is disabled.

Consider achieving network isolation of the Uyuni environment according to your policies.

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

2.2.1. Fully Qualified Domain Name (FQDN)

The Uyuni 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-SP6/html/SLES-all/cha-network.html#sec-network-yast-change-host.

2.2.2. Hostname and IP Address

To ensure that the Uyuni 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-SP6/html/SLES-all/cha-dns.html.

2.2.3. Deployment behind HTTP or HTTPS OSI level 7 proxy

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

- 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"
```

- `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"
```

- Java application Internet access.
  - On the container host, execute `mgrctl term` to open a command line inside the server container: Modify `/etc/rhn/rhn.conf` according to your needs. For example, set:

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

At the end, on the container host, restart the server to enforce the new configuration:

+

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

2.2.4. 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 › Container-deployment**.

In a production environment, the Uyuni Server and clients should always use a firewall. For a comprehensive list of the required ports, see installation-and-upgrade:network-requirements.pdf.

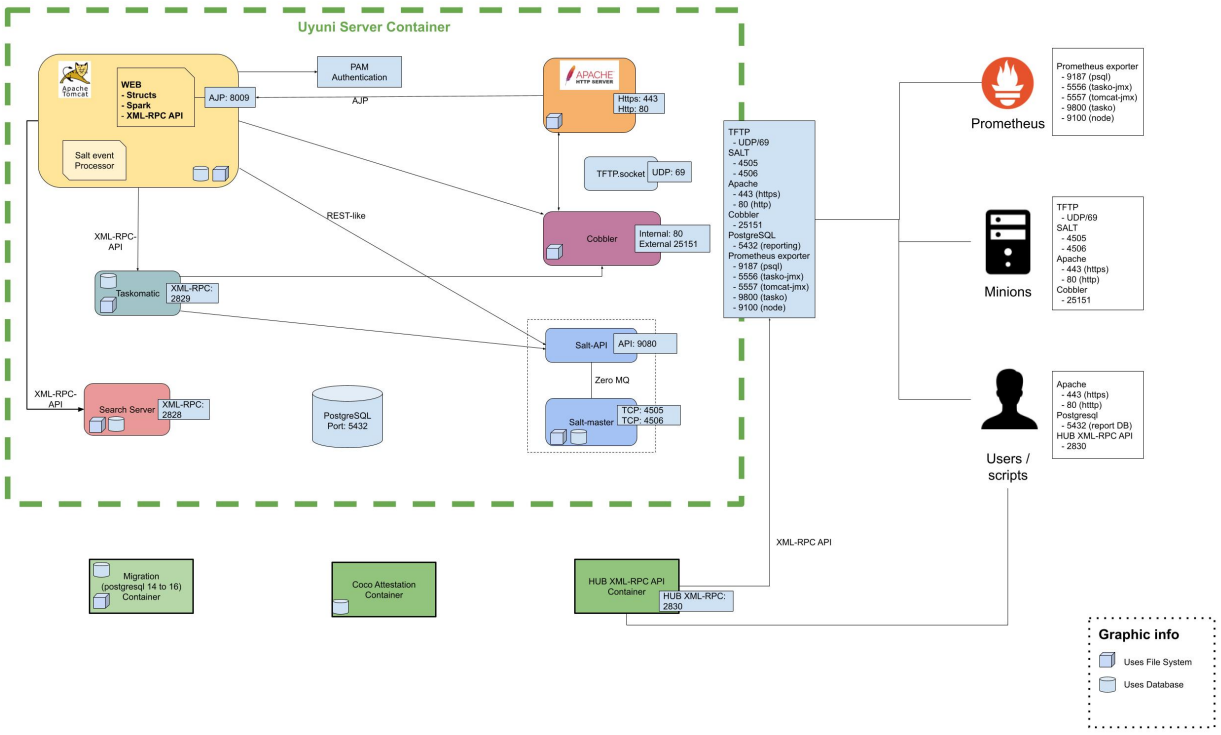
2.2.5. Required Network Ports

This section contains a comprehensive list of ports that are used for various communications within Uyuni.

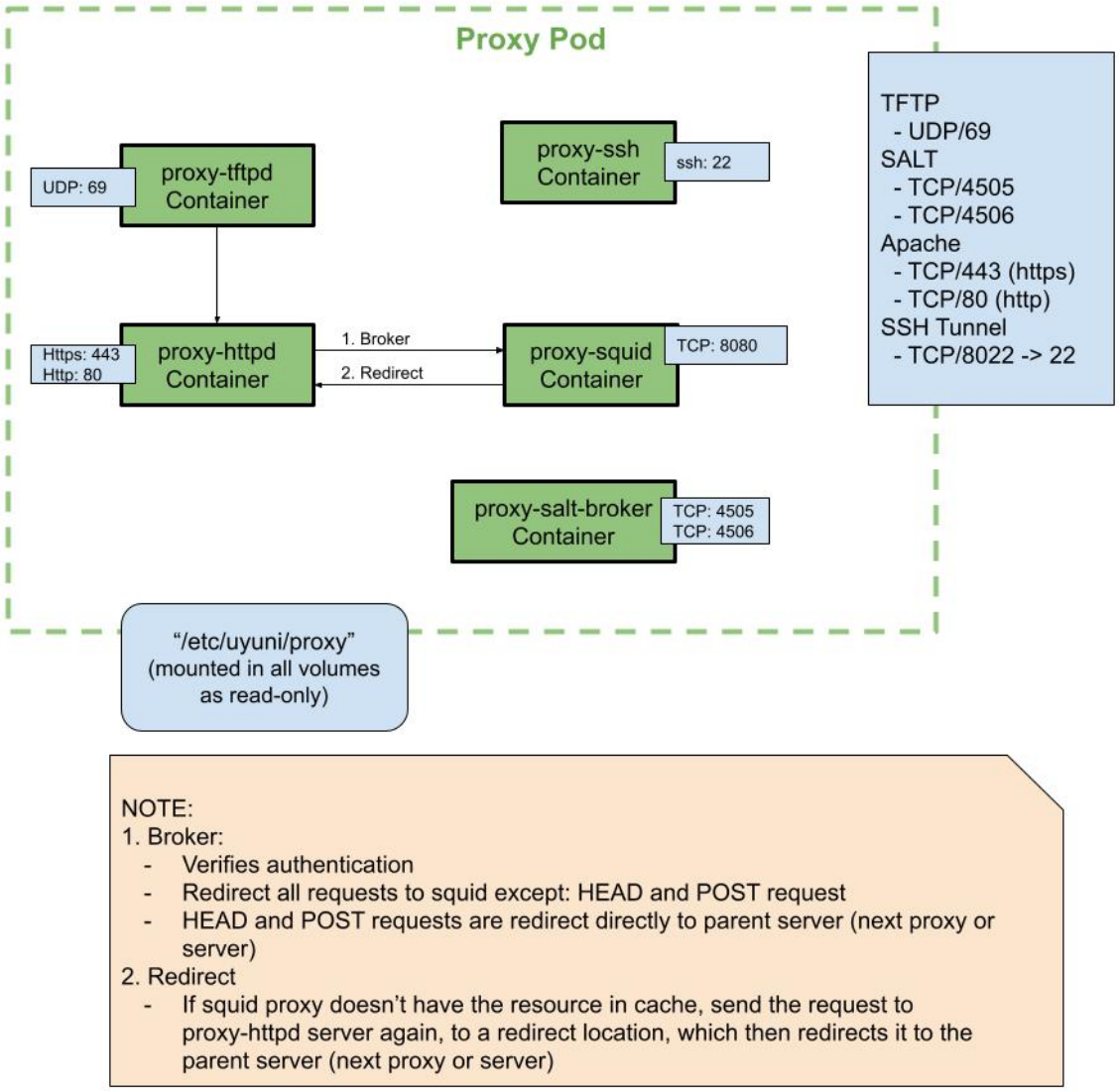
You will not need to open all of these ports. Some ports only need to be opened if you are using the service that requires them.

2.2.5.1. Overview

2.2.5.1.1. Server



2.2.5.1.2. Proxy



2.2.5.2. External Inbound Server Ports

External inbound ports must be opened to configure a firewall on the Uyuni Server to protect the server from unauthorized access.

Opening these ports allows external network traffic to access the Uyuni Server.

Table 3. External Port Requirements for Uyuni Server

Port number	Protocol	Used By	Notes
22			Required for ssh-push and ssh-push-tunnel contact methods.
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.
5556	TCP	Prometheus	Required for scraping Taskomatic JMX metrics.
5557	TCP	Prometheus	Required for scraping Tomcat JMX metrics.
9100	TCP	Prometheus	Required for scraping Node exporter metrics.
9187	TCP	Prometheus	Required for scraping PostgreSQL metrics.
9800	TCP	Prometheus	Required for scraping Taskomatic metrics.
25151	TCP	Cobbler	

2.2.5.3. External Outbound Server Ports

External outbound ports must be opened to configure a firewall on the Uyuni Server to restrict what the server can access.

Opening these ports allows network traffic from the Uyuni Server to communicate with external services.

Table 4. External Port Requirements for Uyuni 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.
25151	TCP	Cobbler	

2.2.5.4. Internal Server Ports

Internal ports are used internally by the Uyuni 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 Uyuni 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.
25151	Cobbler's XMLRPC API
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
```

2.2.5.5. External Inbound Proxy Ports

External inbound ports must be opened to configure a firewall on the Uyuni Proxy to protect the proxy from unauthorized access.

Opening these ports allows external network traffic to access the Uyuni proxy.

Table 6. External Port Requirements for Uyuni Proxy

Port number	Protocol	Used By	Notes
22			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.

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

2.2.5.6. External Outbound Proxy Ports

External outbound ports must be opened to configure a firewall on the Uyuni Proxy to restrict what the proxy can access.

Opening these ports allows network traffic from the Uyuni Proxy to communicate with external services.

Table 7. External Port Requirements for Uyuni Proxy

Port number	Protocol	Used By	Notes
80			Used to reach the server.
443	TCP	HTTPS	Required for SUSE Customer Center.

2.2.5.7. External Client Ports

External client ports must be opened to configure a firewall between the Uyuni Server and its clients.

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

Table 8. External Port Requirements for Uyuni 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.

Port number	Direction	Protocol	Notes
443	Outbound		Used to reach the server or proxy.
9090	Outbound	TCP	Required for Prometheus user interface.
9093	Outbound	TCP	Required for Prometheus alert manager.
9100	Outbound	TCP	Required for Prometheus node exporter.
9117	Outbound	TCP	Required for Prometheus Apache exporter.
9187	Outbound	TCP	Required for Prometheus PostgreSQL.

2.2.5.8. Required URLs

There are some URLs that Uyuni 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`

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 › Troubleshooting**.

2.3. Public Cloud Requirements

This section provides the requirements for installing Uyuni 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 Uyuni setup procedure performs a forward-confirmed reverse DNS lookup. This must succeed in order for the setup procedure to complete and for Uyuni to operate as expected. It is important to perform hostname and IP configuration before you set up Uyuni.

- Uyuni 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.

2.3.1. Network Requirements

When you use Uyuni 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 Uyuni 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 Uyuni instance that lacks adequate perimeter security.

To access the Uyuni Web UI, allow HTTPS when configuring the network access controls. This allows you to access the Uyuni 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.

2.3.2. Prepare Storage Volumes

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

The Uyuni container utilizes default storage locations. These locations should be configured prior to deployment for custom storage. For more information see **Installation-and-upgrade › Container-management**

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 Uyuni. 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 Uyuni 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 Uyuni 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 Uyuni 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>
```

Chapter 3. Deployment and Installation

3.1. Install Uyuni Server

There are various scenarios to deploy a Uyuni Server.

3.1.1. Uyuni Server Deployment on openSUSE Leap Micro 6.1

3.1.1.1. Deployment Preparations

In this section, you will gain expertise in setting up and deploying a Uyuni Server. The process encompasses the installation of `Podman`, `Uyuni container utilities`, deployment, and then initiating interaction with the container through `mgrctl`.

This section assumes you have already configured an openSUSE Leap Micro 6.1 host server, whether it is running on a physical machine or within a virtual environment.

https://download.opensuse.org/distribution/leap-micro/

3.1.1.2. Container Host General Requirements

For general requirements, see **Installation-and-upgrade › General-requirements**.

An openSUSE Leap Micro 6.1 server should be installed from installation media.

https://download.opensuse.org/distribution/leap-micro/

This procedure is described below.

3.1.1.3. Container Host Requirements

For CPU, RAM, and storage requirements, see **Installation-and-upgrade › Hardware-requirements**.

To guarantee that clients can resolve the FQDN domain name, both the containerized server and the host machines must be linked to a functional DNS server. Additionally, it is essential to ensure correct configuration of reverse lookups.

3.1.1.4. Installing Uyuni Tools For Use With Containers

Procedure: Installing Uyuni Tools on openSUSE Leap Micro 6.1

1. On your local host open a terminal window or start up a virtual machine running openSUSE Leap Micro 6.1.
2. Log in.

3. Enter the `transactional-update shell`:

```
transactional-update shell
```

4. Add the following repository to your openSUSE Leap Micro 6.1 server:

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

5. Refresh the repository list and accept the key:

```
zypper ref
```

6. Install the container tools:

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

7. Exit the transactional shell:

```
transactional update # exit
```

8. Reboot the host.

For more information on the Uyuni Container Utilities, see Uyuni Container Utilities.

3.1.1.5. Configure Custom Persistent Storage

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 › Container-management**.

3.1.1.6. Deploying an Uyuni Container With Podman

3.1.1.6.1. `mgradm` Overview

Uyuni is deployed as a container using the `mgradm` tool. There are two methods of deploying a Uyuni server as a container. In this section we will focus on basic container deployment.

For information on using a custom configuration file to deploy, see **Installation-and-upgrade › Container-management**.

For additional information, you can explore further by running `mgradm --help` from the command line.

Procedure: Deploying an Uyuni container with Podman

1. From the terminal run the following command as the sudo user or as root.

```
sudo mgradm install podman
```

You must deploy the container as sudo or root. The following error will be displayed at 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"
```

2. Wait for deployment to complete.
3. Open a browser and proceed to your servers FQDN.

3.1.1.6.2. Persistent Volumes

Many users will want to specify locations for their persistent volumes.

If you are just testing out Uyuni you do not need to specify these volumes. `mgradm` will setup the correct volumes by default.

Specifying volume locations will generally be used for larger production deployments.

By default `podman` stores its volumes in `/var/lib/containers/storage/volumes/`.

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

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

- **Installation-and-upgrade › Container-management**
- **Administration › Troubleshooting**

3.1.2. Uyuni Server Air-gapped Deployment

3.1.2.1. 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.

You can easily deploy container images using `Podman`, `Docker`, or `Skopeo` on a machine with internet access.

Procedure

1. Pull the desired image, then save the image as a `tar` archive. For example:

Listing 1. Podman

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

Listing 2. Docker

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

Listing 3. Skopeo

```
skopeo copy docker://registry.opensuse.org/uyuni/server:latest docker-
archive:server.tar:registry.opensuse.org/uyuni/server:latest
```

2. Transfer the resulting `server-image.tar` to the Server container host and load it using the following command:

Listing 4. Load the server image

```
podman load -i server.tar
```

3.2. Install Uyuni Proxy

There are various scenarios to deploy a Uyuni Proxy. All these scenarios presume you have already successfully deployed a Uyuni 2025.05 Server.

3.2.1. Containerized Uyuni Proxy Setup

Once container host for Uyuni Proxy containers is prepared, setup of containers require few additional steps to finish configuration.

Procedure

1. Generate Uyuni Proxy configuration archive file
2. Transfer configuration archive to the container host prepared in installation step and extract it
3. Start the proxy sevices with `mgrpxy`

3.2.1.1. Generate Proxy Configuration

The configuration archive of the Uyuni Proxy is generated by the Uyuni Server. Each additional Proxy requires its own configuration archive.

For Podman deployment, the container host for the Uyuni Proxy must be registered as a client to the Uyuni 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.

3.2.1.1.1. 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 Uyuni Server or another Uyuni Proxy.
4. In the `Proxy SSH port` field type SSH port on which SSH service is listening on Uyuni 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

Uyuni Proxy or an existing one should be used. You can consider generated certificates as Uyuni builtin (self signed) certificates.

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 › Ssl-certs-imported**.

7. Click Generate to register a new proxy FQDN in the Uyuni 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.

3.2.1.1.2. Generate Proxy Configuration With `spacecmd` and Self-Signed Certificate

You can generate a Proxy configuration using `spacecmd`.

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 .
```

3.2.1.1.3. Generate Proxy Configuration With `spacecmd` and Custom Certificate

You can generate a Proxy configuration using `spacecmd` for a 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 command 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. Copy the generated configuration from the server container:

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

3.2.1.2. Transfer Uyuni Proxy Configuration

Both `spacecmd` command and generating via Web UI ways create a configuration archive. This archive needs to be made available on container host. Transfer this generated archive to the container host.

For installation instructions to use the archive to get the proxy containers, see **Installation-and-upgrade › Container-deployment**.

3.2.1.3. Start Uyuni Proxy Containers

Container can be started with the `mgrpxy` command.

Procedure: Start Uyuni Proxy Containers

1. Run command:

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

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

```
podman ps
```

Five Uyuni Proxy containers should be present and should be part of `proxy-pod` container pod.

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

3.2.2. Uyuni Proxy Deployment on openSUSE Leap Micro 6.1

This guide outlines the deployment process for the Uyuni 2025.05 Proxy. This guide presumes you have already successfully deployed a Uyuni 2025.05 Server. To successfully deploy, you will perform the following actions:

Checklist: Proxy Deployment

1. Review hardware requirements.
2. Install openSUSE Leap Micro 6.1 on a bare-metal machine.
3. Bootstrap the Proxy as a Salt minion.
4. Generate a Proxy configuration.
5. Transfer the Proxy configuration from Server to Proxy
6. Use the Proxy configuration to register the Salt minion as a Proxy with Uyuni.

Supported operating system for the Proxy Container Host

The supported operating system for the container host is openSUSE Leap Micro 6.1.

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.

3.2.2.1. Hardware Requirements for the Proxy

This table shows the hardware requirements for deploying Uyuni Proxy.

Table 9. Proxy Hardware Requirements

Hardware	Details	Recommendation
CPU	x86-64, ARM	Minimum 2 dedicated 64-bit CPU cores
RAM	Minimum	2 GB
	Recommended	8 GB
Disk Space	`/` (root directory)	Minimum 40 GB
	`/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.

3.2.2.2. Container Host General Requirements

For general requirements, see **Installation-and-upgrade › General-requirements**.

An openSUSE Leap Micro 6.1 server should be installed from installation media. This procedure is described below.

3.2.2.3. Container Host Requirements

For CPU, RAM, and storage requirements, see **Installation-and-upgrade › Hardware-requirements**.

To guarantee that clients can resolve the FQDN domain name, both the containerized server and the host machines must be linked to a functional DNS server. Additionally, it is essential to ensure correct configuration of reverse lookups.

3.2.2.4. Installing Uyuni Tools for Use With Containers

Procedure: Installing Uyuni Tools on openSUSE Leap Micro 6.1

1. On your local host open a terminal window or start up a virtual machine running openSUSE Leap Micro 6.1.
2. Log in.
3. Enter the `transactional-update shell`:

```
transactional-update shell
```

4. Add the following repository to your openSUSE Leap Micro 6.1 server:

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

5. Refresh the repository list and accept the key:

```
zypper ref
```

6. Install the container tools:

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

Alternatively you may install `mgrpxy-zsh-completion` or `mgrpxy-fish-completion`.

7. Exit the transactional shell:

```
transactional update # exit
```

8. Reboot the host.

For more information on the Uyuni Container Utilities, see Uyuni Container Utilities.

3.2.2.5. Configure Custom Persistent Storage

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 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 › Container-management**
- **Administration › Troubleshooting**

3.2.2.6. Bootstrap the Proxy Host as a Minion

Task: Bootstrap 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 dropdown.
4. Click + Bootstrap.
5. Wait for the Bootstrap process to complete successfully. Check the **Salt** menu and confirm the Salt minion key is listed and accepted.
6. Reboot the Proxy host.
7. Select the host from the **System** list and trigger a second reboot after all events are finished to conclude the onboarding.

Task: Update the Proxy Host

1. Select the host from the **Systems** list and apply all patches to update it.
2. Reboot the Proxy host.

3.2.2.7. Generate the Proxy Configuration

The configuration archive of the Uyuni Proxy is generated by the Uyuni Server. Each additional Proxy requires its own configuration archive.

The container host for the Uyuni Proxy must be registered as a salt minion to the Uyuni Server prior to generating this Proxy configuration.

You will perform the following tasks:

Procedure:

1. Generate a Proxy configuration file.
2. Transfer the configuration to the Proxy.
3. Start the Proxy with the `mgrpxy` command.

Task: 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 Uyuni Server or another Uyuni Proxy.
4. In the `Proxy SSH port` field type SSH port on which SSH service is listening on Uyuni Proxy. Recommended is to keep default 8022.
5. In the `Max Squid cache size [MB]` field type maximal allowed size for Squid cache. Typically this should be at most 60% of available storage for the containers.
6. In the `SSL certificate` selection list choose if new server certificate should be generated for Uyuni Proxy or an existing one should be used. You can consider generated certificates as Uyuni builtin (self signed) certificates.

   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 on the server are stored in the `/var/lib/containers/storage/volumes/root/ssl-build` directory.

   For more information about existing or custom certificates and the concept of corporate and intermediate certificates, see **Administration › Ssl-certs-imported**.

7. Click Generate to register new proxy FQDN in Uyuni Server and generate configuration archive with details for container host.

8. After a few moments you are presented with file to download. Save this file locally.

3.2.2.8. Transfer the Proxy Configuration

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

Task: Copy the Proxy configuration

1. Copy the files from the Server container to the Server host OS:

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

2. Next copy the files from the Server host OS to the Proxy host:

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

3. Install the Proxy with:

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

3.2.2.9. Start the Uyuni 2025.05 Proxy

Container can now be started with the `mgrpxy` command:

Task: Start and Check Proxy Status

1. Start the Proxy by calling:

```
mgrpxy start
```

2. Check container status by calling:

```
mgrpxy status
```

   Five Uyuni 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

3.2.2.9.1. Using a Custom Container Image for a Service

By default, the Uyuni 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.

3.2.3. Proxy conversion from client

3.2.3.1. Introduction

This chapter describes how Uyuni proxy can be registered with Uyuni server. The main principle consists of using a functionality within Web UI which converts an already onboarded client to a proxy.

The client which is a candidate for conversion to proxy must adhere to the following pre-requisites:

- it must already be onboarded
- it is reachable
- it has access to client tools

3.2.3.2. Convert the client to Uyuni Proxy

The process of conversion is done entirely from the Web UI for already registered clients. For more information about client onboarding, see **Client-configuration › Registration-overview**.

The following two procedures describe the client conversion to a proxy. Either procedure can be used, and will achieve the same outcome.

Procedure: Converting client to Uyuni Proxy using dedicated button

1. For the client chosen to be converted to proxy, go to its `Overview` page.

2. Click button Convert to Proxy.
3. Wait for the conversion to complete.
4. Confirm that the conversion has been successful by locating a new tab `Proxy` on the `Overview` page.

Procedure: Converting client to Uyuni Proxy by changing client's properties

1. For the client chosen to be converted to proxy, go to its `Properties` page.
2. Locate the section `Add-on System Types`.
3. Check the option `Proxy`.
4. Click button Update Properties.
5. Confirm that the conversion has been successful by locating a new tab `Proxy` on the `Overview` page.

3.2.4. Uyuni Proxy Deployment on K3s

3.2.4.1. Installing K3s

On the container host machine, install `K3s` (replace `<K3S_HOST_FQDN>` with the FQDN of your K3s host):

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

3.2.4.2. Installing Tools

The installation requires the `mgrpxy` and `helm` packages.

Install Helm by using the installer script:

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

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

The `mgrpxy` package is available in the container utils repository. Pick the one matching the distribution in: https://download.opensuse.org/repositories/systemsmanagement:/Uyuni:/Stable:/ContainerUtils/.

Procedure

1. To install package on Leap Micro run:

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

2. Reboot.

3.2.4.3. Deploying the Uyuni Proxy Helm Chart

To configure the storage of the volumes to be used by the Uyuni Proxy pod, define persistent volumes for the following claims. If you do not customize the storage configuration, K3s will automatically create the storage volumes for you.

The persistent volume claims are named:

- `squid-cache-pv-claim`
- `/package-cache-pv-claim`
- `/tftp-boot-pv-claim`

Create the configuration for the Uyuni Proxy as documented in **Installation-and-upgrade › Container-deployment**. Copy the configuration `tar.gz` file and then install:

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

For more information see:

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

Chapter 4. Upgrade and Migration

4.1. Server

4.1.1. Legacy Uyuni Server Migration to Container

To migrate a legacy Uyuni Server to a container, a new machine is required.

In the context of this migration, the legacy Uyuni Server (RPM installation) is sometimes also called *old server*.

4.1.1.1. Requirements and Considerations

4.1.1.1.1. General

- An in-place migration is not possible.

4.1.1.1.2. Hostnames

- The migration procedure currently does not include any hostname renaming functionality. The fully qualified domain name (FQDN) on the new server will remain identical to that on the legacy server.

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

4.1.1.2. GPG Keys

- Self trusted GPG keys are not migrated.
- GPG keys that are trusted in the RPM database only are not migrated. Thus synchronizing channels with `spacewalk-repo-sync` can fail.
- The administrator must migrate these keys manually from the legacy Uyuni installation to the container host after the actual server migration.

Procedure: Manual Migration of the GPG Keys to New Server

1. Copy the keys from the legacy Uyuni server to the container host of the new server.
2. Later, add each key to the migrated server with the command `mgradm gpg add <PATH_TO_KEY_FILE>`.

4.1.1.2.1. Initial Preparation on the Legacy Server

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

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

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

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

Procedure: Initial Preparation on the Legacy Server

1. Stop the Uyuni services:

```
spacewalk-service stop
```

2. Stop the PostgreSQL service:

```
systemctl stop postgresql
```

4.1.1.2.2. SSH Connection Preparation

Procedure: Preparing the SSH connection

1. Ensure that for `root` an SSH key exists on the new 2025.05 server. If a key does not exist, create it with the command:

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

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

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

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

3. Copy the public SSH key to the legacy Uyuni Server (`<oldserver.fqdn>`) with `ssh-copy-id`. Replace `<oldserver.fqdn>` with the FQDN of the legacy server:

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

The SSH key will be copied into the legacy server's `~/.ssh/authorized_keys` file. For more information, see the `ssh-copy-id` manpage.

4. Establish an SSH connection from the new server to the legacy Uyuni Server to check that no password is needed. Also there must not by any problem with the host fingerprint. In case of trouble, remove old fingerprints from the `~/.ssh/known_hosts` file. Then try again. The fingerprint will be stored in the local `~/.ssh/known_hosts` file.

4.1.1.2.3. Perform the Migration

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

Procedure: Performing the Migration

1. This step is optional. 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 › Container-management**.

2. Execute the following command to install a new Uyuni server. Replace `<oldserver.fqdn>` with the FQDN of the legacy server:

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

3. Migrate trusted SSL CA certificates.

Migration of the Certificates

Trusted SSL CA certificates that were installed as part of an RPM and stored on a legacy Uyuni in the

`/usr/share/pki/trust/anchors/` directory will not be migrated. Because SUSE does not install RPM packages in the container, the administrator must migrate these certificate files manually from the legacy installation after migration:

Procedure: Migrating the Certificates

1. Copy the file from the legacy server to the new server. For example, as `/local/ca.file`.
2. Copy the file into the container with the command:

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

After successfully running the `mgradm migrate` command, the Salt setup on all clients will still point to the legacy server.

To redirect them to the new 2025.05 server, it is required to rename the new server at the infrastructure level (DHCP and DNS) to use the same FQDN and IP address as legacy server.

4.1.1.3. Kubernetes Preparations

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

For more information, see the installation section on preparing these volumes in **Installation-and-upgrade › Container-management**.

4.1.1.4. Migrating

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

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

or

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

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

4.1.2. Uyuni Server Upgrade

Before running the upgrade command, it is recommended to to update the host operating system.

Updating the host operating system will also result in the update of the Uyuni tooling such as the `mgradm` tool.

Procedure: Upgrading Uyuni Server

1. Refresh software repositories with `zypper`:

```
zypper ref
```

2. Apply available updates with `transactional-update`:

```
transactional-update
```

3. If updates were applied, `reboot`.
4. The Uyuni Server container can be updated using the following command:

```
mgradm upgrade podman
```

This command will bring the status of the container up-to-date and restart the server.

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:

```
mgradm upgrade podman -h
```

For air-gapped installations, first upgrade the container RPM packages, then run the `mgradm` command.

4.2. Proxy

4.2.1. Legacy Proxy Migration to Container

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

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

An in-place migration from previous releases of Uyuni to 2025.05 will remain unsupported due to the HostOS change from openSUSE Leap to openSUSE Leap Micro.

> The traditional contact protocol is no longer supported in Uyuni 2025.05 and later. Before migrating from previous Uyuni releases to 2025.05, any existing traditional clients including the traditional proxies must be migrated to Salt.

4.2.1.1. Migrate From Legacy to Containerized Proxy With Systemd

4.2.1.1.1. Generate Proxy Configuration

Procedure: Generate the Proxy Configuration

1. Log in to the Uyuni Server Web UI.
2. Select **Systems › Proxy Configuration** from the left navigation.
3. Enter your Proxy FQDN. Use the same FQDN as the original proxy host.
4. Enter your Server FQDN.
5. Enter the Proxy port number. *We recommend using the default port of 8022.*
6. Certificate and private key are located on the Server container host in `/var/lib/containers/storage/volumes/root/_data/ssl-build/`.
    - RHN-ORG-TRUSTED-SSL-CERT
    - RHN-ORG-PRIVATE-SSL-KEY
7. Copy the certificate and key to your machine with:

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

8. Select Choose File and browse your local machine for the certificate.
9. Select Choose File and brose your local machine for the private key.
10. Enter the CA password.
11. Click Generate.

4.2.1.1.2. Transfer Proxy Configuration to New Host

Procedure: Transfering the Proxy Configuration

1. From the Server transfer the generated tar.gz file containing the proxy configuration to the new Proxy host:

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

2. Disable the legacy proxy prior to executing the next step:

```
spacewalk-proxy stop
```

3. Deploy the new Proxy with:

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

4. Enable the new Proxy with:

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

5. Run `podman ps` to verify all the containers are present and running:

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

4.2.1.2. Migrate Uyuni Proxy to Uyuni 2025.05 Containerized Proxy

Procedure: Migrate Uyuni Containerized Proxy to Uyuni 2025.05 New Containerized Proxy

1. Boot your new machine and begin installation of openSUSE Leap Micro 6.1.
2. Complete the installation.
3. Update the system:

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

4. Install `mgrpxy` and optionally, `mgrpxy-bash-completion`:

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

5. Reboot.
6. Copy your `tar.gz` proxy configuration to the host.

4.2.1.3. Install Packages Using the Web UI

The `mgrpxy` and `mgrpxy-bash-completion` packages can also be installed via the web UI after the minion has been bootstrapped and registered with the Server.

Procedure: Installing Packages Using the Web UI

1. After installation, ensure that the SLE Micro 6.1 parent channel and Proxy child channels are added and synchronized from the **Admin › Setup Wizard → Products** page.
2. In the Web UI, go to **Systems › Activation Keys** and create an activation key linked for the synchronized SLE Micro 6.1 channel.

3. Bootstrap your system as a minion using the **Systems › Bootstrapping** page.
4. Once the new machine is onboarded and displayed in the systems list, select the system and navigate to the **System Details › Install Package** page.
5. Install the packages `mgrpxy` and `mgrpxy-bash-completion`.
6. Reboot the system.

4.2.1.4. Generate Proxy Config With `spacecmd` and Self-Signed Certificate

You can generate a Proxy configuration using spacecmd.

Procedure: Generate Proxy Config 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 config to the Proxy:

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

4. Deploy the Proxy with:

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

4.2.1.5. Generate Proxy Config With `spacecmd` and Custom Certificate

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

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

Procedure: Generate Proxy Config With `spacecmd` and Custom Certificate

1. SSH into your Server container host.
2. Execute the following command 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. Copy the generated config to the Proxy:

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

4. Deploy the Proxy with:

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

4.2.2. Uyuni Proxy Upgrade

Before running the upgrade command, it is recommended to to update the host operating system. Updating the host operating system will also result in the update of the Uyuni tooling such as the `mgrpxy` tool.

Procedure: Upgrading Uyuni Proxy

1. Refresh software repositories with `zypper`:

```
zypper ref
```

2. Apply available updates with `transactional-update`:

```
transactional-update
```

3. If updates were applied, `reboot`.
4. The Uyuni 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
```

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.

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.

4.3. Clients

4.3.1. Upgrade Clients

Clients use the versioning system of their underlying operating system. For clients using SUSE operating systems, you can perform upgrades within the Uyuni Web UI.

For more information about upgrading clients, see **Client-configuration › Client-upgrades**.

Chapter 5. Basic Server and Proxy Management

5.1. 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.

`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 Uyuni 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.

5.2. Starting and Stopping Containers

The Uyuni 2025.05 Server container can be restarted, started, and stopped using the following commands:

To `restart` the Uyuni 2025.05 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
```

5.3. 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 Uyuni 2025.05.

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.

5.3.1. Server

The following volumes are stored under the **Podman** default storage location on the server.

Table 10. Persistent Volumes: **Podman Default Storage**

Volume Name	Volume Directory
Podman Storage	/var/lib/containers/storage/volumes/

Table 11. Persistent Volumes: **root**

Volume Name	Volume Directory
root	/root

Table 12. Persistent Volumes: **var/**

Volume Name	Volume Directory
var-cobbler	/var/lib/cobbler
var-salt	/var/lib/salt
var-pgsql	/var/lib/pgsql
var-cache	/var/cache
var-spacewalk	/var/spacewalk
var-log	/var/log

Table 13. 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 14. Persistent Volumes: **etc/**

Volume Name	Volume Directory
etc-apache2	/etc/apache2
etc-rhn	/etc/rhn

Volume Name	Volume Directory
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-tls	`/etc/pki/tls`
etc-postfix	`/etc/postfix`
ca-cert	`/etc/pki/trust/anchors`

5.3.2. Proxy

The following volumes are stored under the **Podman** default storage location on the proxy.

Table 15. Persistent Volumes: **Podman Default Storage**

Volume Name	Volume Directory
Podman Storage	`/var/lib/containers/storage/volumes/`

Table 16. Persistent Volumes: **srv/**

Volume Name	Volume Directory
uyuni-proxy-tftpboot	`/srv/tftpboot`

Table 17. Persistent Volumes: **var/**

Volume Name	Volume Directory
uyuni-proxy-rhn-cache	`/var/cache/rhn`
uyuni-proxy-squid-cache	`/var/cache/squid`

Chapter 6. GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections

then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled "GNU
   Free Documentation License".
```