

SUSE Linux Enterprise Update Infrastructure

Setup Guide for Managed Service Providers

SUSE Linux Enterprise Server
Repository Mirroring Tool

Mike Friesenegger, Solution Architect (SUSE)
Gopala Krishnan A, Solution Architect (SUSE)
Terry Smith, Partner Solution Director (SUSE)

SUSE Linux Enterprise Update Infrastructure

Setup Guide for Managed Service Providers

This guide describes the setup of the recommended infrastructure for offering SUSE Linux Enterprise Server and SUSE Linux Enterprise Server for SAP Applications as on-demand offerings in a cloud environment. The update infrastructure scales from small to large cloud installations. The guide is an optional, but highly recommended element of a Managed Service Provider's (MSP's) infrastructure. It allows for central license and repository management.

Disclaimer: Documents published as part of the SUSE Best Practices series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Contents

- 1 Introduction 4
- 2 High level overview 5
- 3 Detailed setup guide 7
- 4 Migrating from Subscription Management Tool to Repository Mirroring Tool 17
- 5 Setting up RMT registration sharing for Managed Service Providers 19
- 6 Configuring and testing the Region Server 23
- 7 Configuring and testing the client instance 27
- 8 References 30
- 9 Legal notice 31
- 10 GNU Free Documentation License 32

1 Introduction

The use of cloud resources is one of the fastest growing areas of the IT industry. Often Infrastructure as a Service (IaaS) is a leading use case of public clouds. The public cloud brings with it a “start and use” expectation. This poses a challenge for products such as SUSE Linux Enterprise Server that require a formal registration process to access update repositories.

In a traditional data center, a SUSE customer will set up a new machine (physical or virtual) and configure the system to be managed by SUSE Multi-Linux Manager, or to connect to a local Repository Mirroring Tool (RMT) or to the SUSE Customer Center (SCC).

Generating registration entitlements for every instance in a cloud environment for use with SCC and providing these to the user does not meet the “start and use” expectation.

RMT establishes a local cache of the SCC content for SUSE Linux Enterprise Server based products. Registration can be fully automated against RMT to meet the expectations in a cloud environment.

The setup described in this guide can be used for a private cloud or can be used in a public cloud type offering. For a public cloud offering access controls must be implemented that verify access privileges. For a private cloud setup RMT servers may not be exposed to ingress traffic from the Internet.

The components described below comprise an implementation provided by SUSE which requires all components of the system be implemented as described. The system is cloud framework agnostic and as such can be implemented for any framework implementation. This does not imply that the system must be set up in the described way. However, if components are modified or alternative architectures are used, then the components provided by SUSE will no longer function in a "plug and play" fashion. It is up to the MSP to implement the necessary components to fit the architectural changes being made.

2 High level overview

MSP Multi-Region RMT Overview

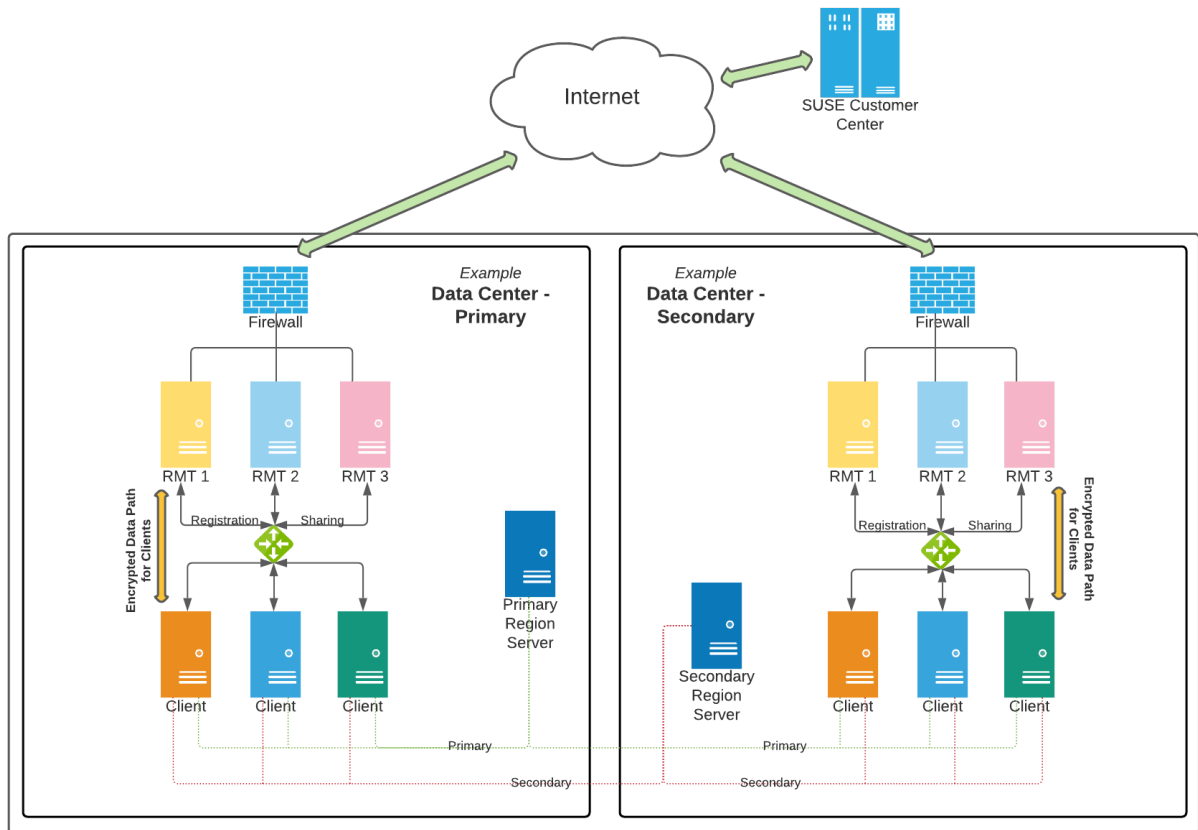


FIGURE 1: UPDATE INFRASTRUCTURE OVERVIEW

The update infrastructure consists of three major components:

- Region Servers
- Repository Mirroring Tool (RMT). This is a specific implementation/tool. The functionality the system provides is referred to as *Update Server*.
- Registration client in the image

Both components run as virtual machines (VMs). All services run on SUSE Linux Enterprise Server 15 SP6 or later. These systems may be registered directly to SCC or may be managed using SUSE Multi-Linux Manager. All systems must have the Public Cloud Module repository enabled.

2.1 Region Server(s)



The function of the Region Server is to provide information about the RMT servers in a given region to the connecting guest.

For MSP's operating in various geographical locations, the Region Servers provide instances information on the RMT server to use for that location. If the MSP has only one region, it is still appropriate to use the Region Server to allow for an easier future expansion.



Note: Optional component

The Region Server is an *optional* component. We advise the implementation of an automatic RMT server selection mechanism if you decided **NOT** to deploy a Region Server. This enables automatic registration, giving end customers a “launch and use” experience, without the need to manually enter or select a registration server.

If a cloud provider chooses not to deploy the Region Server it is not possible to use the SUSE provided client tools, `cloud-regionssrv-client` (<https://github.com/SUSE-Enceladus/cloud-regionssrv-client>) . The SUSE client tool includes a URL Resolver (<https://doc.opensuse.org/projects/libzypp/HEAD/zypp-plugins.html#plugin-url-resolver>)  for `libzypp` that translates the `plugin:susecloud` repository access directive into a URL that can be used by Zypper to access the update repositories provided by the RMT servers. A custom URL resolver must be implemented in the file `/usr/lib/zypp/plugins/urlresolver/susecloud` and must be part of the image from which customers launch instances.

2.1.1 Behind the scenes

A SUSE Linux Enterprise Server guest instance, with the appropriate configuration and the `cloud-regionssrv-client` package installed and the proper configuration set, will connect to a Region Server to receive a list of RMT servers available in the region in which the guest instance was launched. The information is provided to the client in XML format and is sufficient for the client to automatically register with one of the region-local update servers.

Generally, multiple Region Server instances should be operated to ensure availability of the Region Service if any Region Server is too distant (high latency), down, or otherwise unavailable. Information about the Region Servers in the cloud framework is encoded in the guest images. The registration code is configured via the `/etc/regionserverclnt.cfg` configuration file.

If Region Servers have self-signed certificates, the location of the signing certificates can be configured. The verification happens via IP address or DNS name. The client code randomizes the list of configured Region Servers to distribute the access load.

2.2 RMT server(s)

The RMT server serves as cache for the package repositories obtained from SCC. The RMT server itself is registered with SCC, or managed via SUSE Multi-Linux Manager, as it would be in a traditional data center.

Given the data provided by a Region Server, the client proceeds through a “regular” automated registration process. This registration process is identical to the process an administrator would complete when registering a new system against an RMT server operated in a traditional data center.

Registration sharing between multiple RMT servers within a datacenter or region should be configured for redundancy. Guest instances can continue to receive updates if an RMT server is unavailable.

3 Detailed setup guide

Although the Region Server is the first service used by a client, its setup and configuration is dependent on the setup of the RMT servers. Therefore, the setup guide will describe the setup in reverse order as compared to the previous section.

3.1 Prerequisites

- Retrieve the SCC mirroring credentials.
- 1. Visit the SUSE Customer Center at <http://scc.suse.com> and log in.
 2. If you are member of multiple organizations, choose the organization you want to work with from the sidebar on the left.
 3. Select **Proxies** in the top menu.

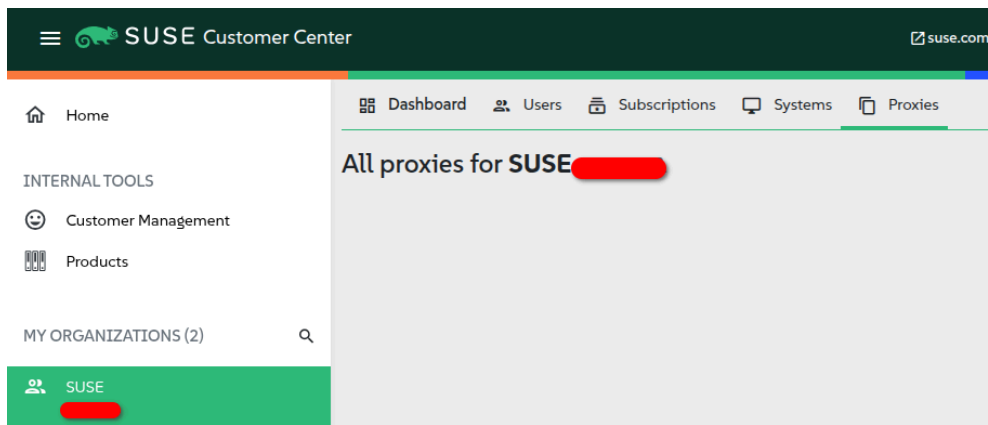


FIGURE 2: SUSE CUSTOMER CENTER ORGANIZATIONS

4. The credentials are displayed in the top right corner.

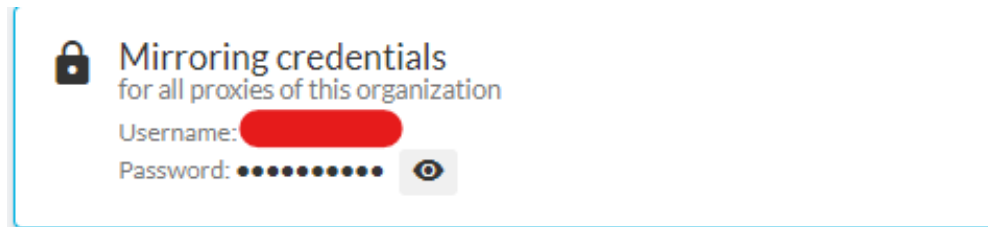


FIGURE 3: SUSE CUSTOMER CENTER MIRRORING CREDENTIALS

5. To see the password, select the eye symbol.

3.2 General setup

Before any of the servers are set up and configured, some general preparations should be completed. Access restriction to the servers is a multi-level process as described in more detail later. This is usually achieved with firewalls.

For each region there should be at least two RMT servers. Additionally, there should be at least two Region Servers. Depending on the footprint of the cloud environment, more Region Servers, with instances running in different regions, may be desired.

3.3 Firewall rules

3.3.1 RMT servers

The firewall rules for the RMT server need to allow incoming traffic on ports 22 and 443. For this configuration the client side also needs to be configured to use HTTPS (port 443) only. RMT can be configured to serve content over port 80 (HTTP) as well in which case no specific client side configuration is necessary. Access rules based on traffic origination are dependent on your cloud network configuration. The examples provide general considerations.

The update servers may run in a network segment that is only accessible from instances within the cloud. Instances automatically get this network segment setup via the network configuration handed out by the cloud DHCP servers or other network configuration mechanism. In this scenario ingress to the RMT servers can simply be wild carded to the IP-ranges of this private network. In this case, no additional access verification in the RMT authentication plugin infrastructure is necessary and installation of the `rmt-server-pubcloud` package is sufficient.

The cloud does not support VPN, direct connections, or a bring-your-own IP range feature. The effect of such a configuration is that traffic destined for the update servers can only originate from the cloud itself. Therefore the firewall rules should be configured such that only traffic from the cloud itself is allowed as ingress. In this configuration it is also sufficient to fall back to the generic authentication process provided by the `rmt-server-pubcloud` package.

The cloud supports routing through a customers data center and/or a bring-your-own IP range feature. In this case, the firewall must allow ingress from all addresses (`0.0.0.0/0; ::/0`). It is necessary to develop an authentication plugin for RMT that works with the default access controls provided by the `rmt-server-pubcloud` package.

All other ports should be blocked.

3.3.2 Region Servers

The firewall rules for the Region Servers need to allow incoming traffic on ports 22 and 443. As with the configuration for the update servers originating IP access rules may be configured and of interest. Unlike for RMT server however, no additional access verification is required if access is granted from all addresses (`0.0.0.0/0; ::/0`).

3.3.3 Setting up SSH access and port 22

Setting up SSH access to the RMT and Region Servers provides for system administrative access. Port 22 is well known as the standard port for SSH traffic and as such is often probed by network scanners. It is possible to move the SSH port following standard SSH configuration practices to a port of your choice on both RMT and Region Servers.

3.3.4 Preparing SSH key pairs for the server

The final preparatory step is to generate SSH key pairs for the servers. It is recommended to use different keys for the RMT server and the Region Server.

```
ssh-keygen -t rsa -f RMT
ssh-keygen -t rsa -f regionsrv
```

3.4 Setting up the RMT server

It is recommended to have three RMT servers per physical location (region) available. A setup of three servers supports a configuration where a minimum HA setup of two is still achieved if one server goes down or is otherwise unavailable. The number of RMT servers depends on the bandwidth within the data center and the number of expected simultaneous users. As a reference, SUSE operates three RMT servers per region on Amazon Web Services (AWS) and can easily satisfy the throughput needs for registered clients and new registrations.

Thus, it is unlikely that more than three RMT servers are needed in your setup. The setup of an RMT server inside a virtual machine is no different from the setup of an RMT Server on a physical machine. A standard RMT server installation is described in the next chapter. For detailed information on the RMT server, refer to the [Repository Mirroring Tool Guide \(https://documentation.suse.com/sles/15-SP6/html/SLES-all/book-rmt.html\)](https://documentation.suse.com/sles/15-SP6/html/SLES-all/book-rmt.html)⁷. A step-by-step instruction for a standard installation is provided in that document.

3.4.1 Performing a default RMT server installation

3.4.1.1 Installing the system

During the system installation, make sure you select the rmt-server package.

1. When getting to the **Installation Summary** step of the installation, select **Software**.

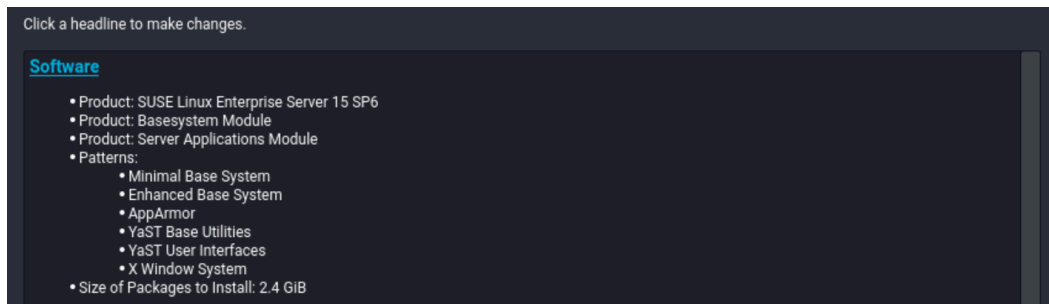


FIGURE 4: INSTALLATION SUMMARY

2. On the software selection page click **Details**. Go to the **Search Tab**. Type *rmt* and select **rmt-server**. The dependencies are automatically selected.

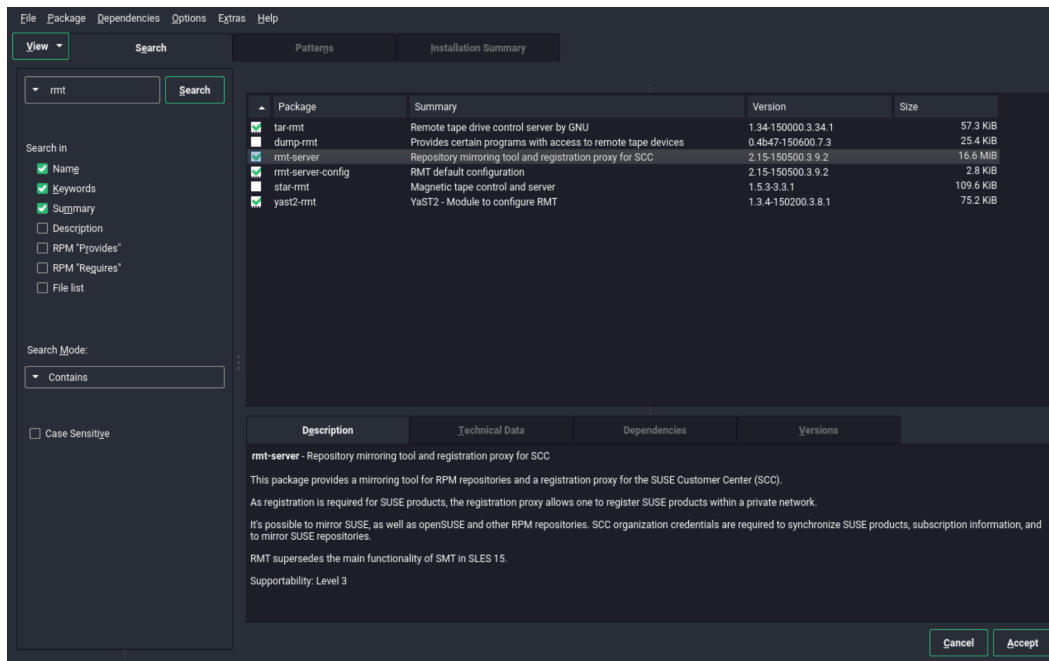


FIGURE 5: SOFTWARE SELECTION

3. Click **Accept**. Click **Continue** to accept the automatic dependencies added.
4. Continue with the installation.

3.4.1.2 Installing an RMT server using the command line

To install RMT on a running SUSE Linux Enterprise Server installation, use `zypper`. Type the following command:

```
sudo zypper in rmt-server
```

RMT is now installed.

3.5 Configuring RMT

3.5.1 Performing the initial configuration

1. Start YaST with the `rmt` module

```
sudo yast2 rmt
```

2. Enter your organization's credentials from SCC and disable forwarding of registrations.
3. Enter the credentials for a new MariaDB user and database name. This user will then be created. Then select **Next**. If a password for the MariaDB root user is already set, you are required to enter it. If no password is set for root, you are asked to enter a new one.
4. Enter a common name for the SSL certificates. The common name should usually be the fully qualified domain name (FQDN) of the server. Enter all domain names and IP addresses with which you want to reach the RMT server as alternative common names. When all common names are entered, select **Next**.
5. To view the summary, click **Next**. Close YaST by clicking **Finish**. YaST then enables and starts all `systemd` services and timers.

3.5.2 Configuring data disks to hold repositories and DB

After the installation of the RMT server package, the directory structure required by RMT is set up in `/var/lib/rmt/public/repo`. The repositories should **not** be stored on the root volume. Storing the repository cache on the root volume will significantly increase the recovery time should the RMT system experience issues and need to be re-created. The following process outlines the steps necessary to set up the external device to hold the repositories.

For the RMT repository disk, follow the steps outlined below:

1. Add the disk to the SUSE Linux Enterprise Server instance that holds the RMT server.
2. Use `sudo lsblk` to list the available block devices.
3. Create a partition table (using YaST, `gparted`, or `fdisk`) and create one partition on the device.
4. Optional but recommended if your cloud framework does not have the capabilities to grow attached devices dynamically: Create a volume group, to easily add more storage for repository growth.
5. Create a file system on the newly created partition. XFS is the recommended file system for this storage device.

6. Copy the content of the RMT directory to a “safe” place:

```
sudo mkdir /tmp/RMTData; rsync -av /var/lib/rmt/public/repo /tmp/RMTData
```

7. Mount the storage partition. Note that `sdX` needs to be replaced with a device identifier that is valid for the server where RMT is running.

```
sudo mount /dev/sdX /var/lib/rmt/public/repo
```

8. Make sure to have the `rmt` user and the `nginx` group owners of the newly mounted directory:

```
sudo chown _rmt /var/lib/rmt/public/repo -R  
sudo chgrp nginx /var/lib/rmt/public/repo -R
```

9. Restore the content of the backed-up repository directory:

```
rsync -av /tmp/RMTData/ /var/lib/rmt/public/repo; rm -rf /tmp/RMTData
```

10. Make sure the disks are mounted on start-up by having the entries in `/etc/fstab`. It is recommended to include *no-fail* which will allow the server to boot even if there are issues with the disk attachment.

The procedure for placing the DB data onto a separate device, this is recommended, is the same:

1. Attach a disk of at least 40 GB to the RMT server.
2. Use `sudo lsblk` to list the available block devices.

3. Create a partition table (using YaST, `gparted`, or `fdisk`) and create one partition on the device.
4. Create a file system on the newly created partition. XFS is the recommended file system for this storage device.
5. Copy the content of the DB directory to a “safe” place:

```
mkdir /tmp/dbData; rsync -av /var/lib/mysql/ /tmp/dbData
```

6. Mount the storage partition. NOTE: `sdX` needs to be replaced with a device identifier that is valid for the server where RMT is running.:

```
mount /dev/sdX /var/lib/mysql
```

7. Restore the content of the repository directory:

```
rsync -av /tmp/dbData/ /var/lib/mysql; rm -rf /tmp/dbData
```

8. Change ownership and group membership to the default:

```
chgrp root mysql -R  
chown mysql mysql -R
```

9. Make sure the disks are mounted on start-up by having the entries in `/etc/fstab`. It is recommended to include *no-fail* which will allow the server to boot even if there are issues with the disk attachment.

3.5.3 Repository management

3.5.3.1 Synchronizing repository metadata

The local RMT database needs to be updated periodically with the information downloaded from SUSE Customer Center. This includes information about available products and repositories. This synchronization is done automatically using the `systemd` timer `rmt-server-sync.timer`. You can manually run the synchronization command. When first installing RMT, this is a good option to get the initial synchronization quicker. To do so, run the following command:

```
sudo rmt-cli sync
```

3.5.3.2 Mirroring products

Packages for enabled repositories are mirrored on your RMT server. Enabling products will enable multiple repositories for a specific SUSE product and version. Packages are downloaded into repositories periodically once a day. The download can also be triggered manually at any time.

The periodic mirroring is done by the `systemd` timer `rmt-server-mirror.timer`. It is recommended that you modify the timer settings such that not all RMT servers run the synchronization at the same time. That means every system should have a different time value for the **OnCalendar** entry in the `/usr/lib/systemd/system/rmt-server-sync.timer`. In addition you want to have a generous value for the **RandomizedDelaySec**. SUSE operations uses a value of **6h** to accommodate latency differences in different parts of the world with regard to access of the SCC caches provided by the SUSE CDN provider.

1. Enable products/repositories to be mirrored:

- Select **Using Products**.

- i. Enter the following command to display the products available:

```
rmt-cli products list --all
```

- ii. Note the ID or product name of the products you want to enable.

- iii. Enter the following command for each product you want to enable:

```
rmt-cli products enable ID/name
```

Example:

```
tux > rmt-cli products list --all
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID   | Product                                     | Version |
| Arch | Mirror?      | Last mirrored |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1946 | Basesystem Module                         | 15 SP2  |
| x86_64 | Don't Mirror |          |
| 2660 | Transactional Server Module               | 15 SP6  |
| x86_64 | Don't Mirror |          |
|      | sle-module-transactional-server/15.6/x86_64 |          |
|      |          |          |
```

```
| 2646 | Web and Scripting Module | 15 SP6 |  
x86_64 | Don't Mirror |  
[...]
```

```
tux > sudo rmt-cli products enable 2660  
Found product by target 2660: Transactional Server Module 15 SP6 x86_64.  
Enabling Transactional Server Module 15 SP6 x86_64:  
Transactional Server Module 15 SP6 x86_64:  
  Enabled repository SLE-Module-Transactional-Server15-SP6-Pool.  
  Enabled repository SLE-Module-Transactional-Server15-SP6-Updates.
```

To enable or disable multiple products at once, specify a space delimited list of their IDs or product strings. Example:

```
tux > sudo rmt-cli repos disable 2526 3263
```

2. Select Using Repositories.

This is similar to the above which uses specific products instead of the complete repository that holds them.

- a. Enter the following command to get a list of repositories to enable:

```
rmt-cli products list --all
```

- b. Note the ID or product name of the repositories you want to enable.
- c. Enter the following command to enable the repository:

```
rmt-cli repos enable ID
```

3. Prior to initiating the mirroring consider using the following command:

```
sudo rmt-cli mirror
```

3.5.3.3 Automating repository management

The manual configuration described in the previous section is not conducive to maintaining more than a few RMT systems. If you operate more than 3 servers, which means an update infrastructure in more than one region, it is recommended that you use the tooling provided in the [update-infra-utils \(https://github.com/SUSE-Enceladus/update-infra-utils\)](https://github.com/SUSE-Enceladus/update-infra-utils) project on GitHub. With this setup, the repositories to be mirrored can be described in `json` format. This helps to keep all systems in synchronization regarding mirroring the same repositories, and supports

keeping the mirroring configuration in a source code control system of your choice. Lastly, the `json` description can be used to monitor the servers and ensure that all configured repositories are indeed mirrored.

4 Migrating from Subscription Management Tool to Repository Mirroring Tool

4.1 Exporting Subscription Management Tool data

1. Update your Subscription Management Tool (SMT) server installation by running the following command:

```
sudo zypper up
```

2. If you want to export your SSL certificates along with the rest of the data, run the following command:

```
sudo smt data-export
```

Remember to keep your certificates in a “safe” place.

If you do not want to export the SSL certificates from SMT, run the command:

```
sudo smt-data-export --no-ssl-export
```

3. The exported configuration is now saved to `smt-export.XXXXXX.tar.gz`. Copy the file to a location which can be accessed by the new RMT server.

4.2 Importing SMT data to RMT

1. Update your RMT server installation by running the following command:

```
sudo zypper up
```

2. Copy the exported `.tar.gz` file to an empty directory. Then unpack it:

```
sudo mkdir EMPTY_DIR
```

```
sudo cd EMPTY_DIR
sudo cp /PATH/T0/smt-export.XXXXXX.tar.gz ./
sudo tar xf smt-export.XXXXXX.tar.gz
```

3. If you chose to export the SSL certificates from SMT, copy the CA private key and certificate to /etc/rmt/ssl/:

```
sudo cp ssl/cacert.key /etc/rmt/ssl/rmt-ca.key
sudo cp ssl/cacert.pem /etc/rmt/ssl/rmt-ca.crt
```

4. Run the YaST RMT configuration module. If you imported the SMT CA certificate, add the domain of the SMT server to the common names of the new SSL certificate.
5. Run the RMT synchronization to get the products and repositories data from SUSE Customer Center:

```
sudo rmt-cli sync
```

6. Import the data from the SMT server:

```
sudo rmt-data-import -d ./
```

7. Optional: Move the mirrored repository data from SMT to RMT and adjust the ownership of the copied data:

```
sudo cp -r /var/www/htdocs/repo/* /var/lib/rmt/public/repo
sudo chown -R _rmt:nginx /var/lib/rmt/public/repo
```

8. Update the packages in the repositories by starting the mirroring process:

```
sudo rmt-cli mirror
```

4.3 Optional: Moving mirrored data from the SMT to the RMT server

1. Option 1: If a separate disk has been used for the mirrored data on the SMT server, perform the following actions:

- a. Unmount the disk from the SMT server with:

```
sudo umount /dev/sdX
```

- b. Mount it to the new RMT server (this can be the same in case of an in-place upgrade):

```
mount /dev/sdX/ /var/lib/rmt/public/repo
```

2. Option 2: In case of an in-place upgrade, and when no separate disk has been used, perform the following action:

- a. Copy the mirrored data from the location SMT uses to the location RMT uses:

- b.

```
sudo cp -r /var/www/htdocs/repo/* /var/lib/rmt/public/repo
```

3. Finally, set the permissions on the repository directory so the RMT server can access it:

```
sudo chown -R _rmt:nginx /var/lib/rmt/public/repo
```

5 Setting up RMT registration sharing for Managed Service Providers

5.1 Introduction

Registration sharing replicates client registrations across independent servers. This enables Managed Service Providers (MSPs) to provide redundant RMT servers per region so registered instances will continue to have access to valid repositories in case of an RMT failure in the region.

For registration sharing you will need to ensure the following:

- Two or more RMT servers need to be available in a region that have been deployed using SUSE Linux Enterprise Server 15 SP6 or later.
- NTP is enabled for all RMT servers.
- Confirm that peer RMT servers within a region are DNS resolvable within the cloud framework or added to /etc/hosts on all RMT server peers within the region.
- Each RMT server is registered with SCC.
- Each RMT server is synchronizing repositories from SUSE Customer Center and the repositories mirrored must be identical.

- Instances can successfully register and deregister with each RMT server in a region.
- Instance registrations forwarding to SUSE Customer Center is disabled in each RMT setup.

5.2 Deployment

1. Stop the RMT servers in the region and install the `rmt-server-pubcloud` package. The following is an example when installing on SUSE Linux Enterprise Server 15 SP6:

```
systemctl stop rmt-server
SUSEConnect -p sle-module-public-cloud/15.6/x86_64
zypper in rmt-server-pubcloud
```

The following message is an example of what may be displayed:

```
Problem: rmt-server-config-2.5.7-3.15.1.x86_64 conflicts with rmt-server-
configuration provided by rmt-server-pubcloud-2.5.7-3.15.1.x86_64
Solution 1: uninstallation of rmt-server-config-2.5.7-3.15.1.x86_64
Solution 2: do not install rmt-server-pubcloud-2.5.7-3.15.1.x86_64
Choose from above solutions by number or cancel [1/2/c/d/?] (c):
```

If so, choose **Solution 1** to uninstall `rmt-server-config`.

2. Instance verification is implemented by placing the implementation into `/usr/share/rmt/engines/instance_verification/lib/instance_verification/providers/`. The implementation is in Ruby and needs to verify, based on instance information sent by the client, whether the client is authorized to access the repositories. The implementation must support the **instance_valid** method. The following example provides guidance for the implementation of the verification plugin.

```
class InstanceVerification::Providers::Example < InstanceVerification::ProviderBase
# Unique identifier for SLES sent by client instance
SLES_PRODUCT_IDENTIFIER = '1234_SUSE_SLES'.freeze
# Unique identifier for SLES For SAP sent by client instance
SLES4SAP_PRODUCT_IDENTIFIER = '6789_SUSE_SAP'.freeze

def instance_valid?
# Extract the instance identifier from the instance data sent by the client
instance_product_id = validate_instance_data(@instance_data)
return true if (@product_hash[:identifier].casecmp('sles').zero? &&
instance_product_id == SLES_PRODUCT_IDENTIFIER)
return true if (@product_hash[:identifier].casecmp('sles_sap').zero? &&
instance_product_id == SLES4SAP_PRODUCT_IDENTIFIER)
```

```

raise InstanceVerification::Exception, 'Product/instance type mismatch'
end

def validate_instance_data(_instance_data)
  # The instance data format is determined by the client implementation and needs to
  # be processed here accordingly. Ideally the data is signed in a way such that it
  # can
  # be independently verified here to not have been tampered with in flight or
  # injected
  # into the stream. The AWS implementation of the Instance Identity Document is one
  # possible
  # implementation route. In this example it is assumed the instance data is json and
  # contains instance_product_id

  return '1234_SUSE_SLES' if @product_hash[:identifier].casecmp('sles').zero?

  return '6789_SUSE_SAP' if @product_hash[:identifier].casecmp('sles_sap').zero?

end

```

In addition to the custom verification module described above, the `rmt-server-pubcloud` package implements generic instance verification applicable to MSPs deployments. Product upgrade from SUSE Linux Enterprise Server to SUSE Linux Enterprise Server for SAP applications, for example, is not supported in a MSP framework. This is enforced as part of the default verification. For additional details, you may reference the implementation maintained in https://github.com/SUSE/rmt/tree/master/engines/instance_verification [GitHub](#).

3. Add a registration sharing section to the bottom of `/etc/rmt.conf`.

For more details, review `/usr/share/rmt/engines/registration_sharing/README.md` on any of the RMT servers with the `rmt-server-pubcloud` package installed.

```

regsharing:
  api_secret: s3cr3t_t0k3n
  data_dir: /var/lib/rmt/regsharing/data-dir
  peers:
    - <remote rmt 1>.domain.com
    - <remote rmt 2>.domain.com
  smt_allowed_ips:
    - 1.2.3.4
    - 5.6.7.8

```

In the above screen:

- *api_secret* is a secret token only shared between the RMT servers in the region. The same token must be configured on all servers that are expected to communicate with each other.
- *data_dir* a directory where the registration sharing replay log is written. The log is processed by the process controlled by the `rmt-server-regsharing.timer` which is provided by `rmt-server-pubcloud` and must be enabled.
- *peers* is a list of the other RMT servers in the region.
- *smt_allowed_ips* are the IP addresses of the peers and used to verify data originates from those systems.

4. Obtain the server certificate from each remote RMT server:

```
curl --tlsv1.2 --silent --insecure --connect-timeout 10 https://<remote rmt  
l>.domain.com/rmt.crt --output /etc/pki/trust/anchors/<remote rmt l>.domain.com.pem
```

Repeat the command for each RMT server listed in `/etc/rmt.conf`.

5. Create hashes for the new server certificates:

```
update-ca-certificates
```

6. Start the `rmt-server` service:

```
systemctl start rmt-server
```

5.3 Starting and enabling the registration sharing timer

The systemd timer `rmt-server-regsharing.timer` defaults to starting the `rmt-server-regsharing` every thirty seconds to synchronize registration information.

```
systemctl start rmt-server-regsharing.timer  
systemctl enable rmt-server-regsharing.timer
```

6 Configuring and testing the Region Server

6.1 Introduction

The function of the Region Server is to provide information about the RMT servers in a given region to a connecting client. The Region Server runs as a Python script using the Flask framework in Apache. The Region Server is provided with the `cloud-regionsrv` package.

Region servers are located in a MSP network. An example of the Region Server placement could be based on geographic boundaries like Americas, Europe/Middle East and Asia. It is recommended that a minimum of three Region Servers are deployed for redundancy. Install, register and fully patch SUSE Linux Enterprise Server 15 SP6 or later for each of the region servers.

A client will attempt to contact a Region Server from a preconfigured list of available Region Servers. The client, if configured and with an appropriate implementation may provide a so called **regionHint** to the Region Server. In the absence of a region hint the Region server will use the originating IP of the client instance to return a list of the closest RMT servers to the client.

6.2 Deployment

1. Use `SUSEConnect --list-extension` to determine the command to enable the Public Cloud module.
2. Install the `cloud-regionsrv` package.

```
zypper in cloud-regionsrv
```

The service itself uses two configuration files:

- `/etc/regionService/regionInfo.cfg` is used to configure the service and contains the location of the log file and the location of the `regionData.cfg`.
- `/etc/regionService/regionData.cfg` contains the data the Region Server will provide to the connecting client

Both files use the **ini** format. If you have frequent changes to the IP addresses used by the cloud framework or IP addresses move between regions, it is recommended to implement a generator for the `/etc/regionService/regionData.cfg` file. In general it is recommended to implement sending the region hint on the client and use the IP address-based look up as a fall back.

3. Make a copy of `/etc/regionService/regionData.cfg`.

```
cp /etc/regionService/regionData.cfg /etc/regionService/regionData.cfg.orig
```

4. Add region information in `/etc/regionService/regionData.cfg`.

The default `regionData.cfg` file provides a template for the configuration file. This file can be maintained manually or be auto-generated, depending on your setup for IP address allocation within your cloud framework.

The `regionData.cfg` file needs to contain one configuration section per region. The hint is processed with string matching. Thus having section names match the configured region names is important. The server implementation has no option of name mapping. For each region all options in the section must be configured.

The section options are as follows:

public-ips

The value for this option is a comma-separated list of IP ranges in CIDR format, for example:

```
public-ips = 62.135.16.0/18,56.56.130.0/16
```

These are the ranges the DHCP server in the given region is configured to use.

smt-server-ip

The value for this option is a comma-separated list of the RMT server IPv4 addresses in the region being configured.

smt-server-ipv6

The value for this option is a comma-separated list of the RMT server IPv6 addresses in the region being configured. Remove this line if IPv6 is not used.

smt-server-name

The value for this option sets the host name of the RMT server that was encoded into the certificate during the setup of the RMT server. If only one value is supplied, it will be used for all IP addresses provided by the `smt-server-ip` setting. If more than one value is supplied, the number of names must match the number of IP addresses given with the `smt-server-ip` option. The order of the names and IP addresses must match as well.

smt-fingerprint

The value for this option is the fingerprint of the root CA created during the RMT Server setup. On the RMT server, the root CA is located in `/etc/rmt/ssl/rmt-ca.crt`. This file is aliased within the nginx configuration to `rmt.crt`. Obtain the fingerprint with the command:

```
curl --tlsv1.2 --silent --insecure --connect-timeout 10 https://<remote rmt
1>.domain.com/rmt.crt | /usr/bin/openssl x509 -noout -fingerprint | cut -d'='
-f2
```

Use this fingerprint for the `smt-fingerprint` value. As with the `smt-server-name`, supplying one value is sufficient if all RMT servers have the same root CA. If each server has its own CA, supply a comma-separated list. The order must match the order of the IP addresses, or certificate acceptance will fail and the guest cannot register with the RMT server.

Example: Completed Section for a Region in a Cloud

The following shows an example of a completed section for a region in a cloud setup.

```
[nor-north]
        public-ips = 62.135.16.0/18,56.56.130.0/16
        smt-server-ip = 62.153.16.20,56.56.130.253
        smt-server-name = rmt-nor.supertuxcloud.com
        smt-fingerprint =
9D:B9:88:DB:87:52:00:55:F0:FF:5D:5C:66:60:D3:E0:5C:D4:FB:79
```

In the example above, both RMT servers share the same certificate. If this were not the case, another value for the `smt-server-name` and for the `smt-fingerprint` options would need to be configured.

```
[mid-north]
        public-ips = 62.135.16.0/18,56.56.130.0/16
```

```
smt-server-ip = 62.153.16.20,56.56.130.253
smt-server-name = rmt-mid-a.supertuxcloud.com,rmt-mid-
b.supertuxcloud.com
smt-fingerprint =
9D:B9:88:DB:87:52:00:55:F0:FF:5D:5C:66:60:D3:E0:5C:D4:FB:79
```

In this example, the servers share the same certificate, but have different names. The certificate in this case would contain a wild card.

5. Generate a Region Server certificate.

As with the RMT servers, the implementation of the `cloud-regionsrv-client` is such that self-signed certificates and non-DNS resolvable names are assumed to be in use. However, it is possible to use publicly signed certificates and DNS resolvable names. The Region Server package (`cloud-regionsrv`) provides a convenient executable to generate the server certificate.

```
genRegionServerCert -c COUNTRY -d DEPARTMENT --host IP_ADDRESS_OR_HOSTNAME -l
LOCATION -o ORGANIZATION -s STATE
```

Example: Using `genRegionServerCert`

The following shows an example using `genRegionServerCert` country and host are the only options with data while the others are blank.

```
genRegionServerCert -c US -d ' ' -l ' ' -o ' ' -s ' ' --host 192.168.100.8
```

This will generate the server certificate in `/etc/apache2/ssl.crt` and the public certificate in `/root/regionServCert/`. The certificate generation script will restart the Apache Web server.

6. Modify `/etc/apache2/vhosts.d/regionsrv_vhost.conf`.

Replace the text `SUBSTITUTE_WITH_CLOUD_SPECIFIC_NAME` in the `ServerName` option by a fully-qualified host name used in the **`genRegionServerCert`**.

Modify the code accordingly, so that you see something similar to the below:

```
Listen 443

<VirtualHost *:443>
ServerName SUBSTITUTE_WITH_CLOUD_SPECIFIC_NAME
SSLEngine on
```

7. Enable and restart the Region Server service.

```
systemctl enable apache2.service
```

```
systemctl restart apache2.service
```

The Region Server reads the `regionData.cfg` file as configured in the `regionInfo.cfg` file at start-up. When a client requests information, the provided region hint from the client data is used as the section name in `regionData.cfg` and the update server data is returned. If no region hint is provided then a longest prefix match is used to attempt a client IP look up to match the update server data to the client.

With the configuration in place the Region Server setup is complete.

7 Configuring and testing the client instance

7.1 Client image configuration recommendations

The Region Server provides the `regionInfo` REST API option that is used by the client to obtain RMT information. The client image accesses the Region Server via `https://IP_ADDRESS_OF_REGION_SERVER/regionInfo`.

As mentioned previously, using the **regionHint** argument with the REST API is recommended. This requires the implementation of a plugin, described below, and configuration of the plugin in the Region Server client configuration file. When a region hint is used the client adds the information provided by the plugin to for the URL `https://IP_ADDRESS_OF_REGION_SERVER/regionInfo?regionHint=REGION_NAME`.

In this case, `REGION_NAME` must match a name of one of the sections in the `regionData.cfg` file as indicated previously. The name is generated by a plugin for Region Server client. The plugin must be placed in `cloudregister` sub-directory for the Python interpreter site-packages directory tree. It is recommended to create an RPM package for the plugin. Details may be obtained from the Region Server client GitHub repository <https://github.com/SUSE-Enceladus/cloud-regionsrv-client> which contains plugin implementations for Amazon, Azure, and Google in the `lib/cloudregister` sub-directory. The spec file in the repository can be used as a packaging example. The implemented plugin must provide the `generateRegionSrvArgs()` function which is expected to return the `"regionHint=REGION_NAME"` string.

The knowledge of the IP addresses of the Region Servers and the certificates for the Region Servers are built into the guest image. The `cloud-regionsrv-client` must be installed in the client image.

It is recommended to create a package with the Region Service client configuration that contains the following.


- The public key files for all Region Servers generated with the **genRegionServerCert** command. Files must be in pem format and end with the .pem extension. Further the names must match the names set in the configuration file. The files need to be placed in the /usr/lib/regionService/certs/ directory.
- The configuration file for the client is /etc/regionserverclnt.cfg. The configuration file is init format.

The client configuration file /etc/regionserverclnt.cfg. It may have 3 sections with predefined options as follows:

- server

This section is processed by the client to obtain the necessary data to request update server information. The **server** supports the following options:

- api

Specifies the API to use on the Region Server. This should be set to **regionInfo** unless you are not using the *regionserver* code from the [cloud-regionsrv](https://github.com/SUSE-Enceladus/cloud-regionsrv) (<https://github.com/SUSE-Enceladus/cloud-regionsrv>)  project or you have a customized version of the code running server side that supports a different API. This option is mandatory if the regionsrv is set.

- regionsrv

A comma-separated list of the Region Servers to query for update server information. The list can be resolvable host names or IP addresses. For each name, a .pem file must exist in the location specified with the certLocation. The option is mutually exclusive with the metadata_server option. If both are specified, only the metadata_server is considered.

- certLocation

The fully qualified directory path where the Region Server certificates are located. This must match with the placement of the certificates in the file system.

- metadata_server

The URL for a metadata server that provides the update server information in the expected format. If the metadata server uses the HTTPS protocol the certificate of the server must be publicly signed. It is expected that the specified metadata server is part of the framework infrastructure and as such is highly available. Therefore only one URL can be specified. The option is mutually exclusive with the `regionsrv` setting.

- `instance`

This section has options to configure information to be retrieved from the instance to send to the Region Server.

- `dataProvider`

Set the command to execute to collect instance data to be sent to the update server. This data is generally used to verify whether the instance is eligible to access the repositories on the update server. As such the data format created by the command and written to *STDOUT* by the executed command must match the format expected server side.

- `instanceArgs`

Specifies the name of the plugin that generates the **regionHint** as discussed previously. The name specifies the Python module name without the `.py` extension.

- `service`

Reserved section name not used.

- `verifyAccess`

Reserved option name not used.

The registration with the update infrastructure is set up by enabling the `guestregister` service in the image. This will ensure an instance gets register upon first boot.

7.2 Manually testing and troubleshooting a client registration

Use the following command to manually test registering a cloud guest:

```
registercloudguest
```

Review the file `/var/log/cloudregister` if troubleshooting is needed.

Use the command `registercloudguest --clean` to clean up the files written after a previous `registercloudguest` command has been run.

7.3 Verifying registration sharing

The steps for verifying registration sharing after successfully registering an instance are below:

1. Register an instance to a single RMT server and verify that the registration exists:

```
rmt-cli systems list
```

2. Initiate a single registration synchronization on the RMT server where the instance was registered:

```
systemctl start rmt-server-regsharing.service
```

3. Verify the registration was shared by viewing the log:

```
journalctl -u rmt-server-regsharing.service
```

4. On the RMT peers, verify the registration exists:

```
rmt-cli systems list
```

Use **`rmt-cli systems remove`** to delete test instances from each RMT server database.

8 References


For more detailed information and references, have a look at the following resources:

- Repository Mirroring Tool Guide (<https://documentation.suse.com/sles/15-SP6/single-html/SLES-rmt/index.html>) ↗
- Registration Sharing - [/usr/share/rmt/engines/registration_sharing/README.md](#)
- Region Server - [/usr/share/doc/packages/cloud-regionsrv/README.md](#)

9 Legal notice

Copyright ©2006-2025 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 “GNU Free Documentation License”.

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see <http://www.suse.com/company/legal/> . Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the **SUSE Best Practices** series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Below we draw your attention to the license under which the articles are published.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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 non-commercially. 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 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".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts". line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.