Client Configuration Guide

SUSE Manager 4.0

September 17, 2019

Table of Contents

Introduction

Registering clients is the first step after installing SUSE Manager, and most of the time you spend with SUSE Manager will be spent on maintaining those clients.

SUSE Manager is compatible with a range of client technologies: you can install traditional or Salt clients, running SUSE Linux Enterprise or another Linux operating system, with a range of hardware options.

For a complete list of supported clients, see **[Installation › Client-requirements ›]**.

This guide discusses how to register and configure different clients, both manually and automatically.

Compare Traditional and Salt Clients

This table lists the availability of various features for traditional and Salt clients. The icons in this table indicate:

- ✔ the feature is available
- ✖ the feature is not available.
- ? the feature is under consideration, and may or may not be made available at a later date

Table 1. Features of Traditional and Salt Clients

Feature	Notes	Traditional	Salt
Registration		✔	✔
Install Packages		✔	✔
Apply Patches		✔	✔
Remote Commands		✔	✔
System Package States		✖	✔
System Custom States		✖	✔
Group Custom States		✖	✔
Organization Custom States		✖	✔
System Set Manager		✔	✔
Service Pack Migration		✔	✔
Virtual Host Management		✔	✔
Virtual Guest Installation		✔	?
System Redeployment	With Auto-installation	✔	?

Feature	Notes	Traditional	Salt
Contact Methods	Between server and client	osad, rhnsd, ssh push	zeromq (Salt default), salt-ssh
SUSE Manager Proxy		✔	✔
Action Chains		✔	✔
Software Crash Reporting		✔	✖
Duplicate Package Reporting		✔	✔
SCAP Auditing		✔	✔
Support for Multiple Organizations		✔	✔
Package Verification		✔	?
System Locking		✔	?
Configuration File Management		✔	✔
Snapshots and Profiles		✔	? (Profiles are supported, sync is not)
Power Management		✔	✔

Client Registration Overview

There are two ways to register clients to your SUSE Manager Server.

For Salt clients, the simplest method is to register your clients using the SUSE Manager Web UI.

If you need to have more control over the process, want to register many clients, or are registering traditional clients, we recommend creating a bootstrap script.

Both methods are described in this manual.

Registering Clients with the Web UI

Registering clients with the SUSE Manager Web UI works for Salt clients only.

Procedure: Registering clients in the Web UI

1. In the SUSE Manager Web UI, navigate to **Systems › Bootstrapping**.
2. In the `Host` field, type the fully qualified domain name (FQDN) of the client to be bootstrapped.
3. In the `SSH Port` field, type the SSH port number to use to connect and bootstrap the client. By default, the SSH port is `22`.
4. In the `User` field, type the username to log in to the client. By default, the username is `root`.
5. In the `Password` field, type password to log in to the client.
6. In the `Activation Key` field, select the activation key that is associated with the software channel you want to use to bootstrap the client.
7. By default, the `Disable SSH Strict Key Host Checking` checkbox is selected. This allows the bootstrap process to automatically accept SSH host keys without requiring you to manually authenticate.
8. OPTIONAL: Check the `Manage System Completely via SSH` checkbox. If you check this option, the client will be configured to use SSH for its connection to the Server, and no other connection method will be configured.
9. Click **`[Bootstrap]`** to begin registration. When the bootstrap process has completed, your client will be listed at **Systems › System List**.

When new packages or updates are installed on the client using SUSE Manager, any end user license agreements (EULAs) are automatically accepted. To review a package EULA, open the package detail page in the Web UI.

Registering Clients with a Bootstrap Script

Registering your clients with a bootstrap script gives you more control over parameters, and can help if you have to register a large number of clients at once. This method works for both Salt and traditional

clients.

To register clients using a bootstrap script, we recommend you create a template bootstrap script to begin, which can then be copied and modified. The bootstrap script you create is executed on the client when it is registered, and ensures all the necessary packages are deployed to the client. There are some parameters contained in the bootstrap script which ensure the client system can be assigned to its base channel, including activation keys, and GPG keys.

It is important that you check the repository information carefully, to ensure it matches the base channel repository. If the repository information does not match exactly, the bootstrap script will not be able to download the correct packages.

If you are bootstrapping Salt clients using the Web UI, you will need to ensure that the client system has Python installed before you begin. For Salt clients running SUSE Linux Enterprise Server 12 or older, you will also require the `python-xml` package.

GPG Keys and Uyuni Client Tools

The GPG key used by Uyuni Client Tools is not trusted by default. When you create your bootstrap script, add a path to the file containing the public key fingerprint with the `ORG_GPG_KEY` parameter.

SLES 15 and Python 3

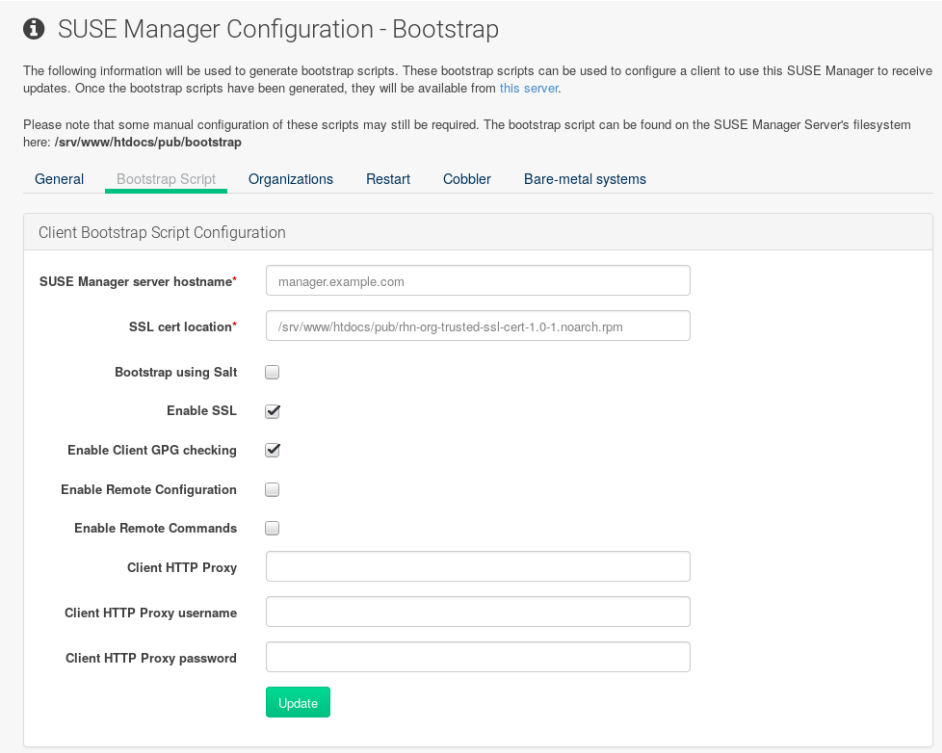
SLE 15 uses Python 3 by default. Bootstrap scripts based on Python 2 must be re-created for SLE 15 systems. Attempting to register SLE 15 systems using Python 2 bootstrap scripts will fail.

Create a Bootstrap Script

This procedure describes how to generate a bootstrap script.

Procedure: Creating a Bootstrap Script

1. In the SUSE Manager Web UI, navigate to **Admin › Manager Configuration › Bootstrap Script**.
2. In the `SUSE Manager Configuration - Bootstrap` dialog, uncheck the `Bootstrap using Salt` checkbox if you are installing a traditional client. For Salt clients, leave it checked. Use default settings and click the **`[Update]`** button.



Using SSL

Unchecking **Enable SSL** in the Web UI or setting `USING_SSL=0` in the bootstrap script is not recommended. If you disable SSL nevertheless you will need to manage custom CA certificates to be able to run the registration process successfully.

3. A template bootstrap script is generated and stored on the server's file system in the `/srv/www/htdocs/pub/bootstrap` directory.

```
cd /srv/www/htdocs/pub/bootstrap
```

The bootstrap script is also available at `https://example.com/pub/bootstrap/bootstrap.sh`.

Edit a Bootstrap Script

You can copy and modify the template bootstrap script you created to customize it.

A minimal requirement when modifying a bootstrap script for use with SUSE Manager is the inclusion of an activation key.

Most packages are signed with GPG, so you will also need to have trusted GPG keys on your system to install them.

In this procedure, you will need to know the exact name of your activation keys. Navigate to **Home** ›

Overview and click on `Manage Activation keys`. All keys created for channels are listed on this page. You must enter the full name of the key you wish to use in the bootstrap script exactly as presented in the key field.

Procedure: Modifying a Bootstrap Script

1. Login as root from the command line on your SUSE Manager server.
2. Navigate to the bootstrap directory with:

```
cd /srv/www/htdocs/pub/bootstrap/
```

3. Create and rename two copies of the template bootstrap script for use with each of your clients.

```
cp bootstrap.sh bootstrap-sles11.sh
cp bootstrap.sh bootstrap-sles12.sh
```

4. Open `sles12.sh` for modification. Scroll down and modify both lines marked in green. You must comment out `exit 1` with a hash mark (`#`) to activate the script and then enter the name of the key for this script in the `ACTIVATION_KEYS=` field as follows:

```
echo "Enable this script: comment (with #'s) this block (or, at least just"
echo "the exit below)"
echo
#exit 1

# can be edited, but probably correct (unless created during initial install):
# NOTE: ACTIVATION_KEYS *must* be used to bootstrap a client machine.
ACTIVATION_KEYS=1-sles12
ORG_GPG_KEY=
```

5. When you have finished, save the file, and repeat this procedure for the second bootstrap script.

Connect Clients

When you have finished creating your script, you can use it to register clients.

Procedure: Running the Bootstrap Script

1. On the SUSE Manager Server, log in as root at the command prompt, and navigate to this directory:

```
cd /srv/www/htdocs/pub/bootstrap/
```

2. Run this command to execute the bootstrap script on the client:

```
cat MODIFIED-SCRIPT.SH | ssh root@example.com /bin/bash
```

The script will execute and proceed to download the required dependencies located in the

repositories directory you created earlier.

3. When the script has finished running, you can check that your client is registered correctly by opening the SUSE Manager Web UI and navigating to **Systems › Overview** to ensure the new client is listed.

When new packages or updates are installed on the client using SUSE Manager, any end user license agreements (EULAs) are automatically accepted. To review a package EULA, open the package detail page in the Web UI.

Package Locks

Package locks can only be used on traditional clients that use the Zypper package manager. The feature is not currently supported on Red Hat Enterprise Linux or Salt clients.

Package locks are used to prevent unauthorized installation or upgrades to software packages on traditional clients. When a package has been locked, it will show a padlock icon, indicating that it can not be installed. Any attempt to install a locked package will be reported as an error in the event log.

Locked packages can not be installed, upgraded, or removed, either through the SUSE Manager Web UI, or directly on the client machine using a package manager. Locked packages will also indirectly lock any dependent packages.

Procedure: Using Package Locks

1. On the client machine, install the `zypp-plugin-spacewalk` package:

```
# zypper in zypp-plugin-spacewalk
```

2. Navigate to the **Software › Packages › Lock** tab on the managed system to see a list of all available packages.
3. Select the packages to lock, and click **`[Request Lock]`**. You can also choose to enter a date and time for the lock to activate. Leave the date and time blank if you want the lock to activate as soon as possible. Note that the lock might not activate immediately.
4. To remove a package lock, select the packages to unlock and click **`[Request Unlock]`**. Leave the date and time blank if you want the lock to deactivate as soon as possible. Note that the lock might not deactivate immediately.

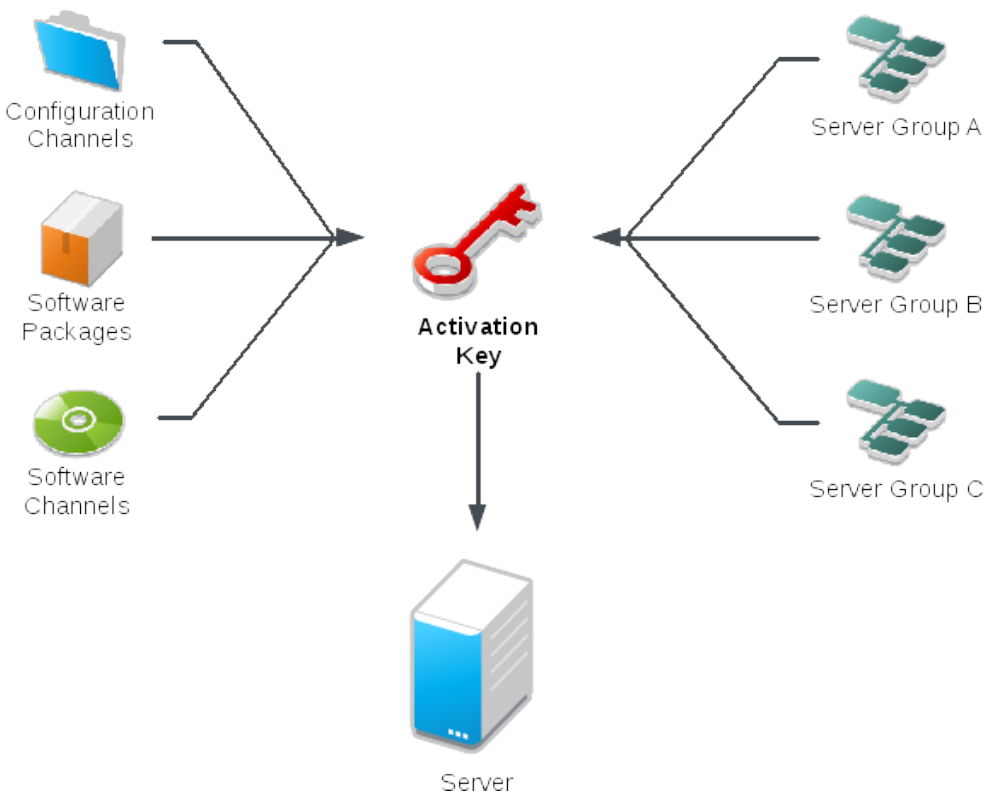
Activation Keys

Activation keys are used with traditional and Salt clients to ensure that your clients have the correct software entitlements, are connecting to the appropriate channels, and are subscribed to the relevant groups. Each activation key is bound to an organization, which you can set when you create the key.

In SUSE Manager, an activation key is a group of configuration settings with a label. You can apply all configuration settings associated with an activation key by adding its label as a parameter to a bootstrap script. We recommend you use an activation key label in combination with a bootstrap script. When the bootstrap script is executed all configuration settings associated with the label are applied to the system the script is run on.

An activation key can specify:

- Channel Assignment
- System Types (Traditionally called Add-on Entitlements)
- Contact Method
- Configuration Files
- Packages to be Installed
- System Group Assignment



Procedure: Creating an Activation Key

1. In the SUSE Manager Web UI, as an administrator, navigate to **Systems › Activation Keys**.
2. Click the **[Create Key]** button.
3. On the `Activation Key Details` page, in the `Description` field, enter a name for the activation key.
4. In the `Key` field, enter the distribution and service pack associated with the key. For example, `SLES12-SP4` for SUSE Linux Enterprise Server 12 SP4.

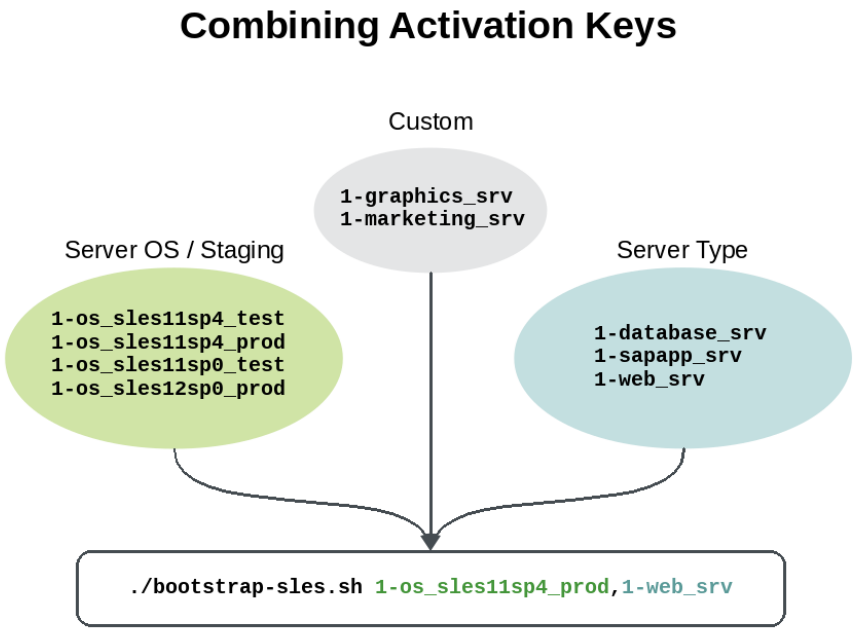
Do not use commas in the `Key` field for any SUSE products. However, you **must** use commas for Red Hat Products. For more information, see [**Reference › System-details ›**].

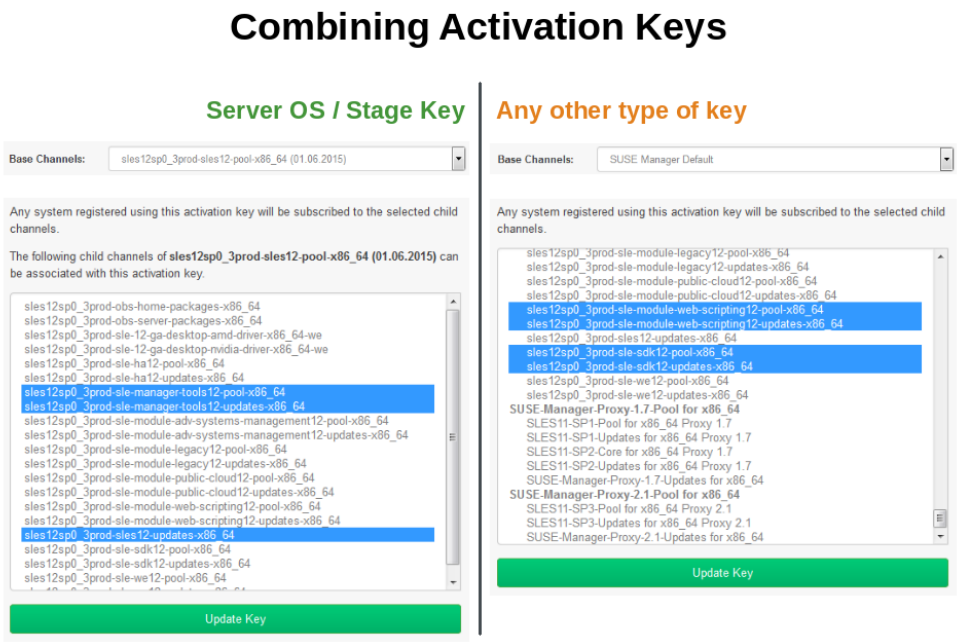
5. In the `Base Channels` drop-down box, select the appropriate base software channel, and allow the relevant child channels to populate.
6. Select the child channels you need (for example, the mandatory SUSE Manager tools and updates channels).
7. We recommend you leave the `Contact Method` set to `Default`.
8. We recommend you leave the `Universal Default` setting unchecked.
9. Click **`[Update Activation Key]`** to create the activation key.
10. Check the `Configuration File Deployment` check box to enable configuration management for this key, and click **`[Update Activation Key]`** to save this change.

The `Configuration File Deployment` check box does not appear until after you have created the activation key. Ensure you go back and check the box if you need to enable configuration management.

Combining Activation Keys

You can combine activation keys when executing the bootstrap script on your clients. Combining keys allows for more control on what is installed on your systems and reduces duplication of keys for large or complex environments.





Activation Key Best Practices

Default Parent Channel

Avoid using the `SUSE Manager Default` parent channel. This setting forces SUSE Manager to choose a parent channel that best corresponds to the installed operating system, which can sometimes lead to unexpected behavior. Instead, we recommend you create activation keys specific to each distribution and architecture.

Bootstrapping with Activation Keys

If you are using bootstrap scripts, consider creating an activation key for each script. This will help you align channel assignments, package installation, system group memberships, and configuration channel assignments. You will also need less manual interaction with your system after registration.

Bandwidth Requirements

Using activation keys might result in automatic downloading of software at registration time, which might not be desirable in environments where bandwidth is constrained.

These options create bandwidth usage:

- Assigning a SUSE Product Pool channel will result in the automatic installation of the corresponding product descriptor package.
- Any package in the `Packages` section will be installed.
- Any Salt state from the `Configuration` section might trigger downloads depending on its contents.

Key Label Naming

If you do not enter a human-readable name for your activation keys, the system will automatically generate a number string, which can make it difficult to manage your keys.

Consider a naming scheme for your activation keys to help you keep track of them. Creating names which are associated with your organization's infrastructure will make it easier for you when performing more complex operations.

When creating key labels, consider these tips:

- OS naming (mandatory): Keys should always refer to the OS they provide settings for
- Architecture naming (recommended): Unless your company is running on one architecture only, for example x86_64, then providing labels with an architecture type is a good idea.
- Server type naming: What is, or what will this server be used for?
- Location naming: Where is the server located? Room, building, or department?
- Date naming: Maintenance windows, quarter, etc.
- Custom naming: What naming scheme suits your organizations needs?

Example activation key label names:

```
sles12-sp2-web_server-room_129-x86_64
```

```
sles12-sp2-test_packages-blg_502-room_21-ppc64le
```

Do not use commas in the `Key` field for any SUSE products. However, you **must** use commas for Red Hat Products. For more information, see **[Reference › System-details ›]**.

Included Channels

When creating activation keys you also need to keep in mind which software channels will be associated with it.

Keys should have a specific base channel assigned to them, for example: `SLES12-SP2-Pool-x86_64`. If this is not the case, SUSE Manager cannot use specific stages. Using the default base channel is not recommended and may cause problems.

- Channels to be included:
  - suse-manager-tools
- Typical packages to be included:
  - mgr-osad (pushing tasks)

        - Installs `python-jabberpy` and `pyxml` as dependencies
    - `mgr-cfg-actions` (Remote Command, Configuration Management)
        - Installs `mgr-cfg` and `mgr-cfg-client` as dependencies

The `suse-manager-tools` channel is mandatory.

Typical packages to be included:

- osad (pushing tasks): Installs `python-jabberpy` and `pyxml` as dependencies
- `rhncfg-actions` (Remote Command, Configuration Managment): Installs `rhncfg` and `rhncfg-client` as dependencies

Registering Clients on a Proxy

Proxy servers can act as a broker and package cache for both tradional and Salt clients. Registering clients on a SUSE Manager Proxy is very similar to registering them directly on SUSE Manager, with a few key differences.

This section contains information on registering Salt clients on a proxy using the Web UI, or with a bootstrap script.

Within the Web UI, proxy pages will show information about both Salt and traditional clients.

You can see a list of clients that are connected to a proxy by clicking on the name of the proxy in **Main Navigation › Systems › Systems › Proxy**, selecting the `Details` tab, and then selecting the `Proxy` tab.

A list of chained proxies for a Salt client can be seen by clicking on the name of the client in **Main Navigation › Systems › All**, selecting the `Details` tab, and then selecting the `Connection` tab.

If you decide to move any of your clients between proxies or the server you will need to repeat the registration process from the beginning.

Registering with the Web UI on a Proxy

Registering Salt clients to a SUSE Manager Proxy using the Web UI is similar to registering clients directly with the SUSE Manager Server.

Procedure: Registering clients to a proxy in the Web UI

1. In the SUSE Manager Web UI, navigate to **Systems › Bootstrapping**.
2. In the `Host` field, type the fully-qualified domain name (FQDN) of the client to be bootstrapped.
3. In the `SSH Port` field, type the SSH port number that will be used to connect and bootstrap the client. By default, the SSH port is `22`.
4. In the `User` field, type the username to log in to the client. By default, the username is `root`.
5. In the `Password` field, type password to log in to the client.
6. In the `Activation Key` field, select the activation key that is associated with the software channel you want to use to bootstrap the client.
7. In the `Proxy` field, select the proxy server you want to register to.
8. By default, the `Disable SSH Strict Key Host Checking` checkbox is selected. This allows the bootstrap process to automatically accept SSH host keys without requiring you to manually authenticate.
9. OPTIONAL: Check the `Manage System Completely via SSH` checkbox. If you check this option, the client will be configured to use SSH for its connection to the server, and no other connection method will be configured.

10. Click **[Bootstrap]** to begin registration. When the bootstrap process has completed, your client will be listed at **Systems › System List**.

Instead of the Web UI, you can use the command line to register a Salt client through a proxy. This procedure requires that you have installed the Salt package on the Salt client before registration, and have activated the `Advanced` systems module.

Procedure: Registering clients to a proxy using the command line

1. Add the proxy FQDN as the master in the clients configuration file located at:

```
/etc/salt/minion
```

   or:

```
/etc/salt/minion.d/NAME.conf
```

2. Add the FQDN to the minion file:

```
master: proxy123.example.com
```

   Save and restart the salt-minion service:

```
systemctl restart salt-minion
```

3. On the Server, accept the new client key with:

```
salt-key -a 'client'
```

   The client connects to the proxy exclusively for Salt operations and normal HTTP package downloads.

Registering with Bootstrap on a Proxy

Registering clients (either traditional or Salt) via SUSE Manager Proxy with a script is done almost the same way as registering clients directly with the SUSE Manager server. The difference is that you create the bootstrap script on the SUSE Manager Proxy with a command-line tool. The bootstrap script then deploys all necessary information to the clients. The bootstrap script requires some parameters (such as activation keys or GPG keys) that depend on your specific setup.

Procedure: Registering clients to a proxy with a bootstrap script

1. Create a client activation key on the SUSE Manager server using the Web UI. See **[Client-configuration › Clients-and-activation-keys ›]**.

2. On the proxy, execute the `mgr-bootstrap` command line tool as root. If needed, use the additional command line switches to tune your bootstrap script. To install a traditional client instead of a Salt client, ensure you use the `--traditional` switch.

   To view available options type `mgr-bootstrap --help` from the command line:

   ```
   # mgr-bootstrap --activation-keys=key-string
   ```

3. Optional: edit the resulting bootstrap script.
4. Execute the bootstrap script on the clients.

Introduction

Kickstart and AutoYaST configuration files allow you to automate client system installations. This is especially useful if you need to install a large number of clients.

For SUSE Linux Enterprise clients, use AutoYaST. When you have created an AutoYaST file, you can upload it and manage it with SUSE Manager.

For Red Hat Enterprise Linux clients, use Kickstart. Kickstart files are created, modified, and managed within the SUSE Manager Web UI.

The Cobbler installation server allows you to automate bare-metal installations. It uses DHCP to access a PXE boot server, and can be used in virtualized environments.

It is also possible to perform a disconnected setup, in an environment where you cannot access the internet.

AutoYaSt

When you install a SUSE Linux Enterprise client, there are a number of questions you need to answer. To automate installation, you can create an AutoYaST file with all the answers to those questions, so that no user intervention is required.

AutoYaST files can be kept on a server and read by individual clients during installation. The same AutoYaST file is used to install multiple clients.

AutoYaST can be used to schedule a registered system to be installed with a new operating system and package profile, or you can use it to install a new system that was not previously registered, or does not yet have an operating system installed.

For more information about AutoYaST, see https://doc.opensuse.org/projects/autoyast/.

Before you Begin

Some preparation is required for your infrastructure to handle AutoYaST installations. Before you create an AutoYaST profile, consider:

- A DHCP server is not required for AutoYaST, but it can make things easier. If you are using static IP addresses, you should select static IP while developing your AutoYaST profile.
- Host the AutoYaST distribution trees via HTTP, provided by SUSE Manager.
- If you are performing a bare metal AutoYaST installation, use these settings:
    - Configure DHCP to assign the required networking parameters and the bootloader program location.
    - In the bootloader configuration file, specify the kernel and appropriate kernel options to be used.

Build a Bootable ISO

You will need to create a bootable ISO image to be used by the target system for installation. When the system is rebooted or switched on, it boots from the image, loads the AutoYaST configuration from your SUSE Manager, and installs SUSE Linux Enterprise Server according to the AutoYaST profile.

To use the ISO image, boot the system and type `autoyast` at the prompt (assuming you left the label for the AutoYaST boot as `autoyast`). Press *`Enter`* to begin the AutoYaST installation.

This is managed by the KIWI image system. For more information about KIWI, see http://doc.opensuse.org/projects/kiwi/doc/.

Integrate with PXE

Instead of using a bootable ISO image, you can use a PXE image instead. This is less error-prone, allows AutoYaST installation from bare metal, and integrates with existing PXE/DHCP environments.

To use this method, make sure your systems have network interface cards (NICs) that support PXE. You will need to install and configure a PXE server, ensure DHCP is running, and place the installation repository on an HTTP server that is reachable by the SUSE Manager Server.

Upload the AutoYaST profile to the SUSE Manager Server using the SUSE Manager Web UI.

When the AutoYaST profile has been created, use the URL from the **Autoinstallation Overview** page as the image location.

For more information about PXE boot, see https://www.suse.com/documentation/sles-15/singlehtml/book_sle_deployment/book_sle_deployment.html#cha.deployment.prep_pxe.

For more information about autoinstallation profiles, see **[Reference › Systems ›]**.

Kickstart

When you install a Red Hat Enterprise Linux client, there are a number of questions you need to answer. To automate installation, you can create a Kickstart file with all the answers to those questions, so that no user intervention is required.

Kickstart files can be kept on a server and read by individual clients during installation. The same Kickstart file is used to install multiple clients.

Kickstart can be used to schedule a registered system to be installed with a new operating system and package profile, or you can use it to install a new system that was not previously registered, or does not yet have an operating system installed.

For more information about Kickstart, see the Red Hat documentation.

Before you Begin

Some preparation is required for your infrastructure to handle Kickstart installations. Before you create a Kickstart profile, consider:

- A DHCP server is not required for kickstarting, but it can make things easier. If you are using static IP addresses, select static IP while developing your Kickstart profile.
- An FTP server can be used instead of hosting the Kickstart distribution tree using HTTP.
- If you are performing a bare metal Kickstart installation, use these settings:
  - Configure DHCP to assign the required networking parameters and the bootloader program location.
  - In the bootloader configuration file, specify the kernel and appropriate kernel options to be used.

Build a Bootable ISO

You will need to create a bootable ISO image to be used by the target system for installation. When the system is rebooted or switched on, it boots from the image, loads the Kickstart configuration from your SUSE Manager, and installs Red Hat Enterprise Linux according to the Kickstart profile.

Building a Bootable ISO

1. Copy the contents of `/isolinux` from the first CD-ROM of the target distribution.
2. Edit the `isolinux.cfg` file to default to 'ks'. Change the 'ks' section to read:

   ```
   label ks
   kernel vmlinuz
     append text ks=`url`initrd=initrd.img lang= devfs=nomount \
       ramdisk_size=16438`ksdevice`
   ```

   IP address-based Kickstart URLs will look like this:

   ```
   http://`my.manager.server`/kickstart/ks/mode/ip_range
   ```

   The Kickstart distribution defined via the IP range should match the distribution from which you are building, or errors will occur.

3. OPTIONAL: If you want to use the `ksdevice`, it looks like:

   ```
   ksdevice=eth0
   ```

   It is possible to change the distribution for a Kickstart profile within a family, such as Red Hat Enterprise Linux AS 4 to Red Hat Enterprise Linux ES 4, by specifying the new distribution label. Note that you cannot move between versions (4 to 5) or between updates (U1 to U2).

4. Customize `isolinux.cfg` further as required. For example, you can add multiple options, different boot messages, or shorter timeout periods.
5. Create the ISO with this command:

```
mkisofs -o file.iso -b isolinux.bin -c boot.cat -no-emul-boot \
  -boot-load-size 4 -boot-info-table -R -J -v -T isolinux/
```

   Note that `isolinux/` is the relative path to the directory containing the modified isolinux files copied from the distribution CD, while `file.iso` is the output ISO file, which is placed into the current directory.

6. Burn the ISO to CD-ROM and insert the disk.
7. Boot the system and type `ks` at the prompt (if you left the label for the Kickstart boot as 'ks').
8. Press *`Enter`* to start Kickstart.

Integrating with PXE

Instead of using a bootable ISO image, you can use a PXE image instead. This is less error-prone, allows Kickstart installation from bare metal, and integrates with existing PXE/DHCP environments.

To use this method, make sure your systems have network interface cards (NICs) that support PXE. You will need to install and configure a PXE server, ensure DHCP is running, and place the installation repository on an HTTP server that is reachable by the SUSE Manager Server.

Upload the Kickstart profile to the SUSE Manager Server using the SUSE Manager Web UI.

When the AutoYaST profile has been created, use the URL from the **Autoinstallation Overview** page as the image location.

For more information about PXE boot, see https://www.suse.com/documentation/sles-15/singlehtml/book_sle_deployment/book_sle_deployment.html#cha.deployment.prep_pxe.

For more information about autoinstallation profiles, see **[Reference › Systems ›]**.

Cobbler

Cobbler is an installation server that allows you to perform unattended system installations. It can be used on server, client, or guest systems, and in virtual environments.

This section explains the Cobbler features most commonly used with SUSE Manager:

- Installation environment analysis using the `cobbler check` command
- Multi-site installation server configuration using the `cobbler replicate` command
- Virtual machine guest installation automation with the `koan` client-side tool

- Building installation ISOs with PXE-like menus using the `cobbler buildiso` command (for SUSE Manager systems with x86_64 architecture)

For more detailed Cobbler documentation, see http://cobbler.github.io/manuals/.

SUSE only supports Cobbler functions that are available in the SUSE Manager Web UI, or through the SUSE Manager API. All features documented here are supported.

Cobbler is not currently supported within SUSE Manager for Retail environments. If you intend to use your installation with SUSE Manager for Retail formulas, do not configure Cobbler.

Cobbler Requirements

To use Cobbler for system installation with PXE, you will require a TFTP server. SUSE Manager installs a TFTP server by default. To PXE boot systems, you will require a DHCP server, or have access to a network DHCP server. Edit the `/etc/dhcp.conf` configuration file to change `next-server` to the hostname or IP address of your Cobbler server.

Cobbler requires an open HTTP port to synchronize data between the Server and the Proxy. By default, Cobbler uses port 80, but you can configure it to use port 443 instead if that suits your environment.

Cobbler uses host names as a unique key for each system. If you are using the `pxe-default-image` to onboard bare metal systems, make sure every system has a unique host name. Non-unique host names will cause all systems with the same host name to have the configuration files overwritten when a provisioning profile is assigned.

Configure Cobbler

Cobbler configuration is primarily managed using the `/etc/cobbler/settings` file. Cobbler will run with the default settings unchanged. All configurable settings are explained in detail in the `/etc/cobbler/settings` file, including information on each setting, and recommendations.

Cobbler uses DHCP to automate bare metal installations from a PXE boot server. You must have administrative access to the network's DHCP server, or be able to configure DHCP directly on the the Cobbler server.

If you have an existing DHCP server, you will need to edit the DHCP configuration file so that it points to the Cobbler server and PXE boot image.

Procedure: Configuring DHCP for Cobbler

1. On the DHCP server, as root, open the `/etc/dhcpd.conf` file.

2. Append a new class with options for performing PXE boot installation. For example:

```
allow booting;
allow bootp;
class "PXE"
{match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
next-server 192.168.2.1;
filename "pxelinux.0";}
```

   This example:

   - Enables the `bootp` protocol for network booting.
   - Creates a class called `PXE`.
   - Identifies systems as `PXEClient` if they are configured with PXE as the first boot priority.
   - Directs PXEClients to the Cobbler server at `192.168.2.1`.
   - Retrieves the `pxelinux.0`bootloader file.

3. Save the file.

Procedure: Configuring PXE Boot in KVM

While it is possible to use KVM with PXE booting, it can be unreliable. We do not recommend you use this on production systems.

1. Use the `virsh` command to produce a dump of the current network XML description:

```
virsh net-dumpxml --inactive network > network.xml
```

2. Open the XML dump file at `network.xml` and add a `bootp` parameter within the `<dhcp>` element:

```
<bootp file='/pxelinux.0' server='192.168.100.153'/>
```

3. Use the `virsh` command to install the updated description:

```
virsh net-define network.xml
```

   Alternatively, you can use the `net-edit` subcommand, which will also perform some error checking.

Listing 1. Example: Minimal Network XML Description for KVM

```
<network>
  <name>default</name>
  <uuid>1da84185-31b5-4c8b-9ee2-a7f5ba39a7ee</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535'/>
    </nat>
  </forward>
  <bridge name='virbr0' stp='on' delay='0'/>
  <mac address='52:54:00:29:59:18'/>
  <domain name='default'/>
  <ip address='192.168.100.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.100.128' end='192.168.100.254'/>
      <bootp file='/pxelinux.0' server='192.168.100.153'/>
</dhcp>
  </ip>
</network>
```

TFTP

SUSE Manager uses the `atftpd` daemon, but it can also use TFTP. The `atftpd` daemon is the recommended method for PXE services, and is installed by default. The default configuration works in most cases. However, if you need to change the configuration, do so using YaST Services Manager.

The TFTP service must be running so it can serve the `pxelinux.0` boot image. Start YaST and use **System › Services Manager** to configure the `tftpd` daemon.

You can also synchronize Cobbler-generated TFTP contents to a SUSE Manager Proxy.

Procedure: Installing TFTP

1. On the SUSE Manager Server, as root, install the `susemanager-tftpsync` package:

   ```
   zypper install susemanager-tftpsync
   ```

2. On the SUSE Manager Proxy, as root user, install the `susemanager-tftpsync-recv` package:

   ```
   zypper install susemanager-tftpsync-recv
   ```

Procedure: Configuring TFTP on a Proxy

1. On the SUSE Manager Proxy, as root, run the `configure-tftpsync.sh` script.
2. The script will interactively ask you for details on the host names and IP addresses of the SUSE Manager Server and Proxy, as well for the location of the `tftpboot` directory on the Proxy.

For more information, use the `configure-tftpsync.sh --help` command.

Procedure: Configuring TFTP on a Server

1. On the SUSE Manager Server, as root, run the `configure-tftpsync.sh` script.

```
configure-tftpsync.sh proxy1.example.com proxy2.example.com
```

2. Run the `cobbler sync` command to push the files to the proxy. This will fail if you have not configured the proxies correctly.
3. If you want to change the list of proxies later on, you can use the `configure-tftpsync.sh` script to edit them.

If you reinstall an already configured proxy and want to push all the files again, you must remove the cache file at `/var/lib/cobbler/pxe_cache.json` before you call `cobbler sync`.

Synchronize and Start the Cobbler Service

The SUSE Manager Server must be able to access the SUSE Manager Proxy systems directly. You cannot push using a proxy.

Before you start the Cobbler service, check that all the prerequisites are configured according to your requirements. You can do this by running the `cobbler check` command.

When your configuration is correct, start the SUSE Manager service:

```
/usr/sbin/spacewalk-service start
```

Do not start or stop the `cobblerd` service independent of the SUSE Manager service. Doing so can cause errors. Always use `/usr/sbin/spacewalk-service` to start or stop SUSE Manager.

Kickstart Templates

Kickstart files are used to automate Red Hat Enterprise Linux client installations. Kickstart templates are used to describe how to create Kickstart files. To help with creating Kickstart templates, you can create Kickstart variables within the SUSE Manager Web UI. This allows you to create and manage large numbers of profiles and systems, without having to manually create Kickstart files for each.

Kickstart templates are shared by various profiles and systems that can each have their own variables and values. These variables modify the templates, and a template engine parses the template and variables into a usable Kickstart file.

Cobbler uses a template engine called Cheetah that provides support for templates, variables, and snippets.

For more information on creating Kickstart profile variables, see **[Reference › Systems ›]**.

Kickstart Template Syntax

Kickstart templates can have static values for certain common items such as PXE image file names, subnet addresses, and common paths such as `/etc/sysconfig/network-scripts/`. However, templates differ from standard Kickstart files in their use of variables.

For example, a standard Kickstart file might have a networking section like this:

```
network --device=eth0 --bootproto=static --ip=192.168.100.24 \
  --netmask=255.255.255.0 --gateway=192.168.100.1 --nameserver=192.168.100.2
```

In a Kickstart template file, the networking section would look like this instead:

```
network --device=$net_dev --bootproto=static --ip=$ip_addr \
  --netmask=255.255.255.0 --gateway=$my_gateway --nameserver=$my_nameserver
```

These variables are substituted with the values set in your Kickstart profile variables or in your system detail variables. If the same variable is defined in both the profile and the system detail, then the system detail variable takes precedence.

Kickstart templates use syntax rules that rely on punctuation symbols. To avoid clashes, they need to be properly treated.

If the template contains shell script variables like `$(example)`, the content needs to be escaped with a backslash: `\$(example)`. If the variable is defined in the template, the templating engine will evaluate it correctly. If there is no such variable, the content will be left unchanged. Escaping the *$* symbol will prevent the templating engine from evaluating the symbol as an internal variable.

Long scripts or strings can be escaped by wrapping them with the `\#raw` and `\#end raw` directives. For example:

```
#raw
#!/bin/bash
for i in {0..2}; do
 echo "$i - Hello World!"
done
#end raw
```

Any line with a # symbol followed by a whitespace is treated as a comment and is therefore not evaluated. For example:

```
#start some section (this is a comment)
echo "Hello, world"
#end some section (this is a comment)
```

For more information about Kickstart templates and Cobbler, see https://fedorahosted.org/cobbler/wiki/KickstartTemplating

Kickstart Snippets

Kickstart snippets are sections of Kickstart code that can be called by a `$SNIPPET()` function. The snippet is parsed by Cobbler and substituted with the contents of the snippet.

This example sets up a snippet for a common hard drive partition configuration:

```
clearpart --all
part /boot --fstype ext3 --size=150 --asprimary
part / --fstype ext3 --size=40000 --asprimary
part swap --recommended

part pv.00 --size=1 --grow

volgroup vg00 pv.00
logvol /var --name=var vgname=vg00 --fstype ext3 --size=5000
```

Save this snippet of the configuration to a file in `/var/lib/cobbler/snippets/`, where Cobbler can access it.

Use the snippet by calling the `$SNIPPET()` function in your Kickstart templates. For example:

```
$SNIPPET('my_partition')
```

Cobbler will parse the function with the snippet of code contained in the `my_partition` file.

Build ISOs with Cobbler

Cobbler can create ISO boot images that contain a set of distributions, kernels, and a menu, that work similar to a PXE installation.

Building ISOs with Cobbler is not supported on IBM Z.

The Cobbler `buildiso` command takes parameters to define the name and output location of the boot ISO. For example:

```
cobbler buildiso --iso=/path/to/boot.iso
```

The boot ISO includes all profiles and systems by default. You can limit which profiles and systems are used, with the `--profiles` and `--systems` options. For example:

```
cobbler buildiso --systems="system1,system2,system3" \
  --profiles="profile1,profile2,profile3"
```

If you cannot write an ISO image to a public `tmp` directory, check your systemd settings in `/usr/lib/systemd/system/cobblerd.service`.

Bare Metal Provisioning

Systems that have not yet been provisioned are called bare metal systems. You can provision bare metal systems using Cobbler. Once a bare metal system has been provisioned in this way, it will appear in the `Systems` list, where you can perform regular provisioning with autoinstallation, for a completely unattended installation.

To successfully provision a bare metal system, you will require a fully patched SUSE Manager server.

The system to be provisioned must have x86_64 architecture, with at least 2 GB RAM, and be capable of PXE booting.

The server uses TFTP to provision the bare metal client, so the appropriate port and networks must be configured correctly in order for provisioning to be successful. In particular, ensure that you have a DHCP server, and have set the `next-server` parameter to the SUSE Manager server IP address or hostname.

Enable Bare Metal Systems Management

Bare metal systems management can be enabled or disabled in the SUSE Manager Web UI by navigating to **Admin › SUSE Manager Configuration › Bare-metal systems**.

New systems are added to the organization of the administrator who enabled the bare metal systems management feature. To change the organization, log in as an Administrator of the required organization, and re-enable the feature.

When the feature has been enabled, any bare metal system connected to the server network will be automatically added to the organization when it is powered on. The process can take a few minutes, and the system will automatically shut down once it is complete. After the reboot, the system will appear in the `Systems` list. Click on the name of the system to see basic information, or go to the `Properties`, `Notes`, and `Hardware` tabs for more details. You can migrate bare metal systems to other organizations if required, using the `Migrate` tab.

Provision Bare Metal Systems

Provisioning bare metal systems is similar to provisioning other systems, and can be done using the `Provisioning` tab. However, you will not be able to schedule provisioning, it will happen automatically as soon as the system is configured and powered on.

System Set Manager can be used with bare metal systems. However, not all SSM features are available, because bare metal systems do not have an operating system installed. This also applies to mixed sets that contain bare metal systems. All features will be re-enabled if the bare metal systems are removed from the set.

Disconnected Setup

When it is not possible to connect SUSE Manager to the internet, you can use it within a disconnected environment.

The repository mirroring tool (RMT) is available on SUSE Linux Enterprise 15 and later. RMT replaces the subscription management tool (SMT), which can be used on older SUSE Linux Enterprise installations.

In a disconnected SUSE Manager setup, RMT or SMT uses an external network to connect to SUSE Customer Center. All software channels and repositories are synchronized to a removable storage device. The storage device can then be used to update the disconnected SUSE Manager installation.

This setup allows your SUSE Manager installation to remain in an offline, disconnected environment.

Your RMT or SMT instance must be used to managed a SUSE Manager Server directly. It cannot be used to manage a second RMT or SMT instance, in a cascade.

For more information on RMT, see https://www.suse.com/documentation/sles-15/book_rmt/data/book_rmt.html.

Synchronize RMT

You can use RMT on SUSE Linux Enterprise 15 installations to manage clients running SUSE Linux Enterprise 12 or later.

We recommend you set up a dedicated RMT instance for each SUSE Manager installation.

Procedure: Setting up RMT

1. On the RMT instance, install the RMT package:

```
zypper in rmt-server
```

2. Configure RMT using YaST:

```
yast2 rmt
```

3. Follow the prompts to complete installation. For more information on setting up RMT, see https://www.suse.com/documentation/sles-15/book_rmt/data/book_rmt.html.

Procedure: Synchronizing RMT with SCC

1. On the RMT instance, list all available products and repositories for your organization:

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

2. Synchronize all available updates for your organization:

```
rmt-cli sync
```

   You can also configure RMT to synchronize regularly using systemd.

3. Enable the products you require. For example, to enable SLES 15:

```
rmt-cli product enable sles/15/x86_64
```

4. Export the synchronized data to your removable storage. In this example, the storage medium is mounted at `/mnt/usb`:

```
rmt-cli export data /mnt/usb
```

5. Export the enabled repositories to your removable storage:

```
rmt-cli export settings /mnt/usb
```

Ensure that the external storage is mounted to a directory that is writeable by the RMT user. You can change RMT user settings in the `cli` section of `/etc/rmt.conf`.

Synchronize SMT

SMT is included with SUSE Linux Enterprise 12, and can be used to manage clients running SUSE Linux Enterprise 10 or later.

SMT requires you to create a local mirror directory on the SMT instance in order to synchronize repositories and packages.

For more details on installing and configuring SMT, see https://www.suse.com/documentation/sles-12/book_smt/data/book_smt.html.

Procedure: Synchronizing SMT with SCC

1. On the SMT instance, create a database replacement file:

```
smt-sync --createdbreplacementfile /tmp/dbrepl.xml
```

2. Export the synchronized data to your removable storage. In this example, the storage medium is mounted at `/mnt/usb`:

```
smt-sync --todir /mnt/usb
smt-mirror --dbreplfile /tmp/dbrepl.xml --directory /mnt/usb \
           --fromlocalsmt -L /var/log/smt/smt-mirror-export.log
```

3. Export the enabled repositories to your removable storage:

```
rmt-cli export settings /mnt/usb
```

Ensure that the external storage is mounted to a directory that is writeable by the RMT user. You can change SMT user settings in `/etc/smt.conf`.

Synchronize a Disconnected Server

When you have removable media loaded with your SUSE Customer Center data, you can use it to synchronize your disconnected server.

Procedure: Synchronizing a Disconnected Server

1. Mount your removable media device to the SUSE Manager server. In this example, the mount point is `/media/disk`.
2. Open `/etc/rhn/rhn.conf` and define the mount point by adding or editing this line:

```
server.susemanager.fromdir = /media/disk
```

3. Restart the Tomcat service:

```
systemctl restart tomcat
```

4. Refresh the local data:

```
mgr-sync refresh
```

5. Perform a synchronization:

```
mgr-sync list channels
mgr-sync add channel channel-label
```

The removable disk that you use for synchronization must always be available at the same mount point. Do not trigger a synchronization, if the storage medium is not mounted. This will result in data corruption.

Other Clients

It is possible to register clients using operating systems from Red Hat, CentOS, or Ubuntu.

This section contains information specific to clients running operating systems other than those provided by SUSE.

Registering Red Hat Enterprise Linux Clients

This section contains information about registering traditional and Salt clients running Red Hat Enterprise Linux operating systems.

Set up a Red Hat Enterprise Linux Client

Ensure that your client meets these requirements before you start:

- 8 GB RAM or more
- Two or more physical or virtual CPU cores
- Access to Red Hat Enterprise Linux installation and subscription media
- An LVM or an NFS mount is recommended

You will need to ensure you provision enough disk space. The `/var/spacewalk` directory contains all mirrored RPMs, and can take a significant amount of disk space. For example, the Red Hat Enterprise Linux 6 x86_64 channels require over 90 GB.

Taskomatic will use one CPU core, and requires at least 3072 MB RAM. To ensure that Taskomatic has access to enough memory, open the `/etc/rhn/rhn.conf` configuration file, and add this line:

```
taskomatic.java.maxmemory=3072
```

You are responsible for arranging access to Red Hat base media repositories and RHEL installation media. You must obtain support from either Red Hat or SUSE for all your RHEL systems. If you do not do this, you might be violating your terms with Red Hat.

Red Hat Enterprise Linux Channel Management

The base Red Hat Enterprise Linux software channel does not contain any packages. This is because SUSE does not provide Red Hat Enterprise Linux base media. You will need to obtain base media from Red Hat, which you can then add as a child channel to the Red Hat Enterprise Linux parent channel.

The Red Hat Enterprise Linux and tools channels are provided by SUSE Customer Center. You can synchronize your client with the `mgr-sync` command to get them.

Because the Red Hat Enterprise Linux channels are particularly large, it can take up to 24 hours for an initial channel synchronization to complete. When you have completed the initial synchronization, we recommended you clone the channel before working with it. This provides you with a backup of the original synchronization data.

The following procedure guides you through setup of the Red Hat Enterprise Linux media as a SUSE Manager channel. All packages on the media will be mirrored into a child channel located under the distribution name and architecture.

Procedure: Setting up a Red Hat Enterprise Linux Channel

1. In the SUSE Manager Web UI, navigate to **Channels › Manage Software Channels**, and click **`[Create Channel]`**.
2. Fill in the channel details, and add the channel as a child to the corresponding Red Hat Enterprise Linux distribution channel for your architecture. The base parent channel will not contain any packages.
3. Modify the associated activation key to include your new child channel.
4. At the command prompt, as root, copy your installation disk image to the `/tmp` directory.
5. Create a directory to contain the media content:

```
mkdir -p /srv/www/htdocs/pub/rhel
```

6. Mount the ISO:

```
mount -o loop /tmp/name_of_iso /srv/www/htdocs/pub/rhel
```

7. Synchronize the packages with `spacewalk-repo-sync`:

   For Red Hat Enterprise Linux 7:

```
spacewalk-repo-sync -c channel_name -u https://127.0.0.1/pub/rhel/
Repo URL: https://127.0.0.1/pub/rhel/
Packages in repo:              [...]
Packages already synced:       [...]
Packages to sync:              [...]
[...]
```

   For Red Hat Enterprise Linux 6:

```
spacewalk-repo-sync -c channel_name -u https://127.0.0.1/pub/rhel/Server/
Repo URL: https://127.0.0.1/pub/rhel/Server/
Packages in repo:              [...]
Packages already synced:       [...]
Packages to sync:              [...]
[...]
```

Sometimes, the `spacewalk-repo-sync` will stop running during a synchronization, which will give this error:

```
[Errno 256] No more mirrors to try.
```

If this occurs, you can run `spacewalk-repo-sync` in debugging mode to determine the error.

Start debugging mode:

```
export URLGRABBER_DEBUG=DEBUG
```

Check the output:

```
/usr/bin/spacewalk-repo-sync --channel _<channel-label>_ --type yum
```

Disable debug mode:

```
unset URLGRABBER_DEBUG``
```

Register Red Hat Enterprise Linux Clients

Before you register Red Hat Enterprise Linux clients to your SUSE Manager Server, check that you have the corresponding Red Hat Enterprise Linux product enabled, and the required channels are fully synchronized.

You will also need an activation key associated with the Red Hat Enterprise Linux channel. For more information on activation keys, see **[Client-configuration › Clients-and-activation-keys ›]**.

Missing packages will cause your registration to fail. For Red Hat Enterprise Linux clients, packages are contained on the Red Hat Enterprise Linux installation media. Ensure you have loop-mounted the installation media, and added it as a child channel to the base Red Hat Enterprise Linux channel.

Red Hat Enterprise Linux 7

- Product: Red Hat Enterprise Linux 7
- Mandatory channels: `rhel-x86_64-server-7` , `res7-suse-manager-tools-x86_64` , `res7-x86_64`

Red Hat Enterprise Linux 6

- Product: Red Hat Enterprise Linux 6
- Mandatory channels: `rhel-x86_64-server-6` , `res6-suse-manager-tools-x86_64` , `res6-x86_64`

There are two ways to check if a channel has finished synchronizing:

- In the SUSE ManagerWeb UI, navigate to **Admin › Setup Wizard** and select the **SUSE Products** tab. This dialog displays a completion bar for each product when they are being synchronized.
- You can also check the synchronization log file at the command prompt. Use the `cat` or `tail -f` command to view the `/var/log/rhn/reposync/channel-label.log` file. If you use this method, remember that base channels can contain multiple child channels. Each of the child channels will generate its own log during the synchronization progress. You will need to check all the base and child channel log files to be sure that the synchronization is complete.

When you are ready to register your Red Hat Enterprise Linux client, follow the instructions in **[Client-configuration › Registration-overview ›]**.

Registering {centos} Clients

This section contains information about registering traditional and Salt clients running {centos} operating systems.

Set up a {centos} Client

The `spacewalk-utils` package contains a number of upstream command line tools required for client administration. You will also require the `spacewalk-common-channels` tool. Keep in mind SUSE only provides support for `spacewalk-clone-by-date` and `spacewalk-manage-channel-lifecycle` tools.

The `/etc/rhn/spacewalk-common-channels.ini` configuration file must contain the channel references you want to add. If a channel is not listed, check the latest version for updates: https://github.com/spacewalkproject/spacewalk/tree/master/utils

You will also need an activation key associated with the {centos} channel. For more information on activation keys, see **[Client-configuration › Clients-and-activation-keys ›]**.

Procedure: Adding Channels and Repositories

1. At the command prompt on the SUSE Manager Server, as root, install the `spacewalk-utils` package:

```
zypper in spacewalk-utils
```

2. Add the {centos} base, updates, and client channels using the `spacewalk-common-channels` script:

```
spacewalk-common-channels -u admin -p`secret`-a x86_64 'centos7'
spacewalk-common-channels -u admin -p`secret`-a x86_64 'centos7-updates'
spacewalk-common-channels -u admin -p`secret`-a x86_64 'centos7-uyuni-client-x86_64'
```

Procedure: Synchronizing {centos} Clients

1. In the SUSE Manager Web UI, navigate to **Main Menu › Software › Manage**, and select the base channel you want to synchronize.
2. In the `Repositories` tab, navigate to the `Sync` subtab, and click **`[Sync Now]`**. You can also create a regular synchronization schedule on this page.

When you have prepared you SUSE Manager Server, you can install your {centos} client using your preferred installation media and method.

Procedure: Setting up a {centos} Client

1. At the command prompt, copy all relevant GPG keys to `/srv/www/htdocs/pub`. If you intend to use a bootstrap script to register your client, you can add the GPG keys to your bootstrap script.
2. Check that the client machine can resolve itself and your SUSE Manager Server using DNS.
3. Check that there is an entry in `/etc/hosts` for the real IP address of the client.
4. Create an activation key called `centos7` on the SUSE Manager Server that points to the correct parent and child channels, including the {centos} base repository, updates, and client channels.

When you are ready to register your {centos} client, follow the instructions in **[Client-configuration › Registration-overview ›]**.

Registering Ubuntu Clients

This section contains information about registering Salt clients running Ubuntu operating systems.

SUSE Manager supports Ubuntu 16.04 LTS and 18.04 LTS Clients using Salt. Traditional clients are not supported.

Supported features:

- Bootstrapping
- Synchronizing `.deb` channels
- Assigning `.deb` channels to clients
- GPG signing `.deb` repositories
- Information displayed in System details pages
- Package install, update, and remove
- Package install using `Package States`
- Configuration and state channels

Bootstrapping is supported for starting Ubuntu clients and performing initial state runs such as setting repositories and performing profile updates. However, the root user on Ubuntu is disabled by default, so in order to use bootstrapping, you will require an existing user with `sudo` privileges for Python.

Some actions are not yet supported:

- Patch and errata support
- Bare metal installations, PXE booting, and virtual host provisioning
- Live patching
- CVE Audit
- If you use are using a repository from storage media (`server.susemanager.fromdir = …` option in rhn.conf), Ubuntu Client Tools will not work.

Canonical does not endorse or support SUSE Manager.

Prepare to Register Ubuntu Clients

Some preparation is required before you can register Ubuntu clients to the SUSE Manager Server.

Before you begin, ensure you have the Ubuntu product enabled, and have synchronized the Ubuntu channels:

For Ubuntu 18.04:

- Product: Ubuntu Client 18.04
- Mandatory channels: `ubuntu-18.04-pool-amd64`

For Ubuntu 16.04:

- Product: Ubuntu Client 16.04
- Mandatory channels: `ubuntu-16.04-pool-amd64`

The mandatory channels do not contain Ubuntu upstream packages. The repositories and channels for synchronizing upstream content must be configured manually.

Procedure: Preparing to Register Ubuntu Clients with Spacewalk

Before you begin, ensure you have installed the `spacewalk-common-channels` utility from the `spacewalk-utils` package.

1. Ensure that you have the appropriate software channels available on your system. In the SUSE Manager Web UI, navigate to **Software › Channel List › All**. You should see a base channel and a child channel for your architecture, for example:

```
ubuntu-18.04-pool for amd64
|
+- Ubuntu-18.04-SUSE-Manager-Tools for amd64
```

2. Open the `/etc/rhn/spacewalk-common-channels.ini` file, and locate the sections that begin with `ubuntu` and end with `main` or `updates`. Change the `yumrepo_url` to an existing repository URL. Do not change the `ubuntu-$VERSION-pool-$ARCH` section.

```
[ubuntu-1804-pool-amd64]
; do not change
label    = ubuntu-18.04-pool-amd64
checksum = sha256
archs    = amd64-deb
repo_type = deb
name     = ubuntu-18.04-pool for amd64
gpgkey_url =
gpgkey_id =
gpgkey_fingerprint =
yumrepo_url = http://localhost/pub/repositories/empty-deb/

[ubuntu-1804-amd64-main]
label    = ubuntu-1804-amd64-main
checksum = sha256
archs    = amd64-deb
repo_type = deb
name     = Ubuntu 18.04 LTS AMD64 Main
base_channels = ubuntu-18.04-pool-amd64
; change URL
yumrepo_url = http://mirror.example.com/ubuntu/dists/bionic/main/binary-amd64/

[ubuntu-1804-amd64-updates]
label    = ubuntu-1804-amd64-main-updates
name     = Ubuntu 18.04 LTS AMD64 Updates
archs    = amd64-deb
repo_type = deb
checksum = sha256
base_channels = ubuntu-18.04-pool-amd64
; change URL
yumrepo_url = http://mirror.example.com/ubuntu/dists/bionic-updates/main/binary-amd64/
```

3. Use the `spacewalk-common-channels` command to create the required channels and repositories. Ensure you use the appropriate version number in this command, either [systemitem]ubuntu-1604 or [systeitem]ubuntu-1804``:

```
spacewalk-common-channels -u <admin_user> -p <admin_pass> -a amd64-deb -v 'ubuntu-1804*'
```

Enable the Ubuntu Root User

The root user on Ubuntu is disabled by default. You can enable it by editing the `sudoers` file.

Procedure: Granting Root User Access

1. On the client, edit the `sudoers` file:

```
sudo visudo
```

Grant `sudo` access to the user by adding this line to the `sudoers` file. Replace `<user>` with the

name of the user that will be used to bootstrap the client in the Web UI:

```
<user>   ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2, /usr/bin/python3
```

Virtualization

You can use SUSE Manager to manage virtualized clients in addition to regular traditional or Salt clients. In this type of installation, a virtual host is installed on the SUSE Manager Server to manage any number of virtual guests. If you choose to, you can install several virtual hosts to manage groups of guests.

The range of capabilities that virtualized clients have depends on the third-party virtualization provider you choose.

Xen and KVM hosts and guests can be managed directly in SUSE Manager. This enables you to autoinstall hosts and guests using AutoYaST or Kickstart, and manage guests in the Web UI.

For VMWare, including VMWare vSphere, SUSE Manager requires you to set up a virtual host manager (VHM) to control the VMs. This gives you control over the hosts and guests, but in a more limited way than available with Xen and KVM.

Other third-party virtualization providers are not directly supported by SUSE Manager. However, if your provider allows you to export a JSON configuration file for the VM, you can upload that configuration file to SUSE Manager and manage it with a VHM.

Virtualization with Xen and KVM

Xen and KVM virtualized clients can be managed directly in SUSE Manager.

To begin, you will need to set up a virtual host on your SUSE Manager Server. You can then set up autoinstallation using AutoYaST or Kickstart for future virtual hosts, and for virtual guests.

This section also includes information about administering your virtual guests after they have been installed.

Host Setup

The way that you set up Xen or KVM on a VM host depends on what operating system you want to use on its associated guests.

For SUSE operating systems, see the SLES Virtualization Guide available from https://www.suse.com/documentation/sles-15/book_virt/data/book_virt.html.

For Red Hat Enterprise Linux operating systems, refer to the Red Hat documentation for your version.

SUSE Manager uses `libvirt` to install and manage guests. You must have the `libvirtd` package installed on your host. In most cases, the default settings are usually sufficient, and you should not need to adjust them. However, if you want to access the VNC console on your guests as a non-root user, you will need to perform some configuration changes. For more information about how to set this up, consult the relevant documentation for your operating system.

You will require a bootstrap script on the SUSE Manager Server. Your bootstrap script must include the

activation key for your host. We also recommend that you include your GPG key for additional security. For more on creating a bootstrap script, see **[Client-configuration › Registration-bootstrap ›]**.

When your bootstrap script is ready, execute it on the host to register it with the SUSE Manager Server. For more on client registration, see **[Client-configuration › Manual-registration-overview ›]**.

For Salt clients, you will need to enable the `Virtualization Host` entitlement. This allows you to see VM changes instantly. To do this, in the SUSE Manager Web UI, navigate to the `System Details` page for the host, and click on the `Properties` tab. In the `Add-On System Types` section, check `Virtualization Host`, and click **`[Update Properties]`** to save the changes. You will need to schedule a hardware refresh to activate the change. Navigate to **System Details › Hardware**, and click **`[Schedule Hardware Refresh]`**.

By default, VM hosts use the `rhnsd` service to check for scheduled actions every four hours, in order to load balance in environments where there are a lot of clients. This can create delays of up to four hours before an action is carried out. When you are managing VM guests, this long delay is not always ideal, especially for actions like rebooting a guest. To address this, you can disable the `rhnsd` service, and enable the `osad` service. The `osad` service receives commands using a jabber protocol, and will execute commands instantly.

To disable the `rhnsd` service, and enable the `osad` daemon, run these commands as the root user:

```
service rhnsd stop
service rhnsd disable
```

```
service osad enable
service osad start
```

Autoinstallation

You can use AutoYaST or Kickstart to automatically install and register Xen and KVM guests.

You will require an activation key for the VM host you want to register the guests to, and for each guest. Your activation key must have the `provisioning` and `Virtualization Platform` entitlements. Your activation key must also have access to the `mgr-virtualization-host` and `mgr-osad` packages. For more on creating activation keys, see **[Client-configuration › Clients-and-activation-keys ›]**.

If you want to automatically register the guests with SUSE Manager after installation, you will need to create a bootstrap script. For more on creating a bootstrap script, see **[Client-configuration › Registration-bootstrap ›]**.

Autoinstallation of VM guests works only if they are configured as Traditional clients. Salt clients can be created using a template disk image, but not by using AutoYaST or Kickstart.

Create an Autoinstallable Distribution

You will need to create an autoinstallable distribution on the VM host to be able to autoinstall clients from SUSE Manager. The distribution can be made available from a mounted local or remote directory, or on a loop-mounted ISO image.

The configuration of the autoinstallable distribution will differ depending on whether you are using a SLES or Red Hat Enterprise Linux operating system on your guests. The packages for a Red Hat Enterprise Linux installation are fetched from the associated base channel. Packages for installing SUSE systems are fetched from the autoinstallable distribution. Therefore, for SLES systems, the autoinstallable distribution must be a complete installation source.

Table 2. Paths for autoinstallable distributions

Operating System Type	Kernel Location	initrd Location
Red Hat Enterprise Linux	`images/pxeboot/vmlinuz`	`images/pxeboot/initrd.img`
SLES	`boot/<arch>/loader/initrd`	`boot/<arch>/loader/linux`

In all cases, ensure that the base channel matches the autoinstallable distribution.

Before you begin, ensure you have a installation media available to your VM Host. It can be on a network resource, a local directory, or an loop-mounted ISO image. Additionally, ensure that all files and directories are world-readable.

Procedure: Creating an Autoinstallable Distribution

1. In the SUSE Manager Web UI, navigate to **Systems › Autoinstallation › Distributions** and click **`[Create Distribution]`**.
2. In the `Create Autoinstallable Distribution` section, use these parameters:
    - In the `Distribution Label` section, type a unique name for the distribution. Use only letters, numbers, hyphens (`-`), periods (`.`), and underscores (`_`), and ensure the name is longer than four characters.
    - In the `Tree Path` field, type an absolute path to the installation source.
    - In the `Base Channel` field, select the channel that matches the installation source. This channel is used as the package source for non-SUSE installations.
    - In the `Installer Generation` field, select the operating system version that matches the installation source.
    - In the `Kernel Options` field, type any options to be passed to the kernel when booting for the installation. The `install=` parameter and the `self_update=0` `pt.options=self_update` parameter are added by default.
    - In the `Post Kernel Options` section, type any options to be passed to the kernel when booting the installed system for the first time.

3. Click **[Create Autoinstallable Distribution]** to save.

When you have created an autoinstallable distribution, you can edit it by navigating to **Systems › Autoinstallation › Distributions** and selecting the distribution you want to edit.

Create and Upload an Autoinstallation Profile

Autoinstallation profiles contain all the installation and configuration data needed to install a system. They can also contain scripts to be executed after the installation is complete.

Kickstart profiles can be created using the SUSE Manager Web UI, by navigating to **Systems › Autoinstallation › Profiles**, clicking **[Create New Kickstart File]**, and following the prompts. You can also create AutoYaST or Kickstart autoinstallation profiles by hand.

An example AutoYaST profile that includes a script for registering the client with SUSE Manager is available in **[Administration › Autoyast-example ›]**. If you are using AutoYaST to install SLES, you will also need to include this snippet:

```
<products config:type="list">
  <listentry>SLES</listentry>
</products>
```

For more on AutoYaST, see **[Client-configuration › Autoyast ›]**.

For more on Kickstart, see **[Client-configurastion › Kickstart ›]**, or refer to the Red Hat documentation for your installation.

Procedure: Uploading an Autoinstallation Profile

1. In the SUSE Manager Web UI, navigate to **Systems › Autoinstallation › Profiles** and click **[Upload Kickstart/AutoYaST File]**.
2. In the `Create Autoinstallation Profile` section, use these parameters:
    - In the `Label` field, type a unique name for the profile. Use only letters, numbers, hyphens (`-`), periods (`.`), and underscores (_), and ensure the name is longer than six characters.
    - In the `Autoinstall Tree` field, select the autoinstallable distribution you created earlier.
    - In the `Virtualization Type` field, select the relevant Guest type (for example, `KVM Virtualized Guest`. Do not choose `Xen Virtualized Host` here.
    - OPTIONAL: If you want to manually create your autoinstallation profile, you can type it directly into the `File Contents` field. If you have a file already created, leave the `File Contents` field blank.
    - In the `File to Upload` field, click **[Choose File]**, and use the system dialog to select the file to upload. If the file is successfully uploaded, the filename will be shown in the `File to Upload` field.
    - The contents of the uploaded file will be shown in the `File Contents` field. If you need to

make edits, you can do so directly.

3. Click **[Create]** to save your changes and store the profile.

When you have created an autoinstallation profile, you can edit it by navigating to **Systems** › **Autoinstallation** › **Profiles** and selecting the profile you want to edit. Make the desired changes and save your settings by clicking **[Create]**.

If you change the `Virtualization Type` of an existing Kickstart profile, it might also modify the bootloader and partition options, potentially overwriting any custom settings. Carefully review the `Partitioning` tab to verify these settings before making changes.

Automatically Register Guests

When you install VM guests automatically, they are not registered to SUSE Manager. If you want your guests to be automatically registered as soon as they are installed, you can add a section to the autoinstallation profile that invokes a bootstrap script, and registers the guests.

This section gives instructions for adding a bootstrap script to an existing AutoYaST profile.

For more on creating a bootstrap script, see **[Client-configuration › Registration-bootstrap ›]**. For instructions on how to do this for {kickstart], refer to the Red Hat documentation for your installation.

Procedure: Adding a Bootstrap Script to an AutoYaST Profile

1. Ensure your bootstrap script contains the activation key for the VM guests you want to register with it, and that is located on the host at `/srv/www/htdocs/pub/bootstrap_vm_guests.sh`.
2. In the SUSE Manager Web UI, navigate to **Systems** › **Autoinstallation** › **Profiles**, and select the AutoYaST profile to associate this script with.
3. In the `File Contents` field, add this snippet at the end of the file, immediately before the closing `</profile>` tag. Ensure you replace the example IP address in the snippet with the correct IP address for your SUSE Manager Server:

```
<scripts>
  <init-scripts config:type="list">
    <script>
      <interpreter>shell </interpreter>
      <location>
        http://`192.168.1.1`/pub/bootstrap/bootstrap_vm_guests.sh
      </location>
    </script>
  </init-scripts>
</scripts>
```

4. Click **Update** to save your changes.

If your AutoYaST profile already contains a `<scripts>` section, do not add a second one. Place the bootstrap snippet inside the existing `<scripts>` section.

Autoinstall VM Guests

Once you have everything set up, you can start to autoinstall your VM guests.

Each VM host can only install one guest at a time. If you are scheduling more than one autoinstallation, make sure you time them so that the next installation does not begin before the previous one has completed. If a guest installation starts while another one is still running, the running installation will be canceled.

1. In the SUSE Manager Web UI, navigate to **Systems › Overview**, and select the VM host you want to install guests on.
2. Navigate to the `Virtualization` tab, and the `Provisioning` subtab.
3. Select the autoinstallation profile you want to use, and specify a unique name for the guest.
4. Choose a proxy if applicable and enter a schedule.
5. To change the guest's hardware profile and configuration options, click **`[Advanced Options]`**.
6. Click **`[Schedule Autoinstallation and Finish]`** to complete.

Manage VM Guests

You can use the SUSE Manager Web UI to manage your VM Guests, including actions like shutting downa nd restarting, and adjusting CPU and memory allocations.

To do this, you will need your Xen or KVM VM host registered to the SUSE Manager Server, and have the `libvirtd` service running on the host. You will also need the `mgr-cfg-actions` package installed on your SUSE Manager Server.

In the SUSE Manager Web UI, navigate to **Systems › System List**, and click on the VM host for the guests you want to manage. Navigate to the `Virtualization` tab to see all guests registered to this host, and access the management functions.

For more information on managing VM guests using the Web UI, see **[Reference › Systems ›]**.

Virtualization with VMWare

You can use VMWare vSphere virtual machines, including ESXi and vCenter, with SUSE Manager by setting up a virtual host manager (VHM).

To begin, you will need to set up a VHM on your SUSE Manager Server, and inventory the available VM hosts. Taskomatic can then begin data collection using the VMs API.

VHM Setup

The Virtal Host Manager (VHM) runs on the SUSE Manager Server.

To run a VHM, your SUSE Manager Server will need to have port 443 open, in order to access the VMWare API.

VMWare hosts use access roles and permissions to control access to hosts and guests. Ensure that any VMWare objects or resources that you want to be inventoried by the VHM have at least `read-only` permissions. If you want to exclude any objects or resources, mark them with `no-access`.

When you are adding new hosts to SUSE Manager, you will need to consider if the roles and permissions that have been assigned to users and objects need to be inventoried by SUSE Manager.

For more on users, roles, and permissions, see the VMWare vSphere documentation: https://docs.vmware.com/en/VMware-vSphere/index.html

Procedure: Creating a VMWare VHM

1. In the SUSE Manager Web UI, navigate to **Systems › Virtual Host Managers**.
2. Click **`[Create]`** and select `VMWare-based`.
3. In the `Add a VMWare-based Virtual Host Manager` section, use these parameters:
    - In the `Label` field, type a custom name for your VHM.
    - In the `Hostname` field, type the fully-qualified domain name (FQDN) or host IP address.
    - In the `Port` field, type the ESXi API port to use (for example, `443`).
    - In the `Username` field, type the username associated with the VM host.
    - In the `Password` field, type the password associated with the VM host user.
4. Click **`[Create]`** to save your changes and create the VHM.
5. On the `Virtual Host Managers` page select the new VHM.
6. On the `Properties` page, click **`[Refresh Data]`** to inventory the new VHM.

To see which objects and resources have been inventoried, navigate to **Systems › System List › Virtual Systems**.

Connecting to the ESXi server from a browser using HTTPS can sometimes log an `invalid certificate` error. If this occurs, refreshing the data from the virtual hosts server will fail. To correct the problem, extract the certificate from the ESXi server, and copy it to `/etc/pki/trust/anchors`. Re-trust the certificate by running the `update-ca-certificates` command on the command line, and restart the spacewalk services.

After your VHM has been created and configured, Taskomatic will run data collection automatically. If

you want to manually perform data collection, navigate to **Systems › Virtual Host Managers**, select the appropriate VHM, and click **[Refresh Data]**.

SUSE Manager ships with a tool called `virtual-host-gatherer` that can connect to VHMs using their API, and request information about virtual hosts. `virtual-host-gatherer` maintains the concept of optional modules, where each module enables a specific VHM. This tool is automatically invoked nightly by Taskomatic. Log files for the `virtual-host-gatherer` tool are located at `/var/log/rhn/gather.log`.

Virtualization with Other Third Party Providers

If you want to use a third-party virtualization provider other than Xen, KVM, or VMware, you can import a JSON configuration file to SUSE Manager.

Similarly, if you have a VMWare installation that does not provide direct access to the API, a file-based VHM will provide you with some basic management features.

This option is for importing files that have been created with the `virtual-host-gatherer` tool. It is not designed for manually created files.

Procedure: Exporting and Importing a JSON File

1. Export the JSON configuration file by running `virtual-host-gatherer` on the VM network.
2. Save the produced file to a location accessible by your SUSE Manager Server.
3. In the SUSE Manager Web UI, navigate to **Systems › Virtual Host Managers**.
4. Click **[Create]** and select `File-based`.
5. In the `Add a file-based Virtual Host Manager` section, use these parameters:
    - In the `Label` field, type a custom name for your VHM.
    - In the `Url` field, type the path to your exported JSON configuration file.
6. Click **[Create]** to save your changes and create the VHM.
7. On the `Virtual Host Managers` page, select the new VHM.
8. On the `Properties` page, click **[Refresh Data]** to inventory the new VHM.

Listing 2. Example: Exported JSON configuration file:

```
{
    "examplevhost": {
        "10.11.12.13": {
            "cpuArch": "x86_64",
            "cpuDescription": "AMD Opteron(tm) Processor 4386",
            "cpuMhz": 3092.212727,
            "cpuVendor": "amd",
            "hostIdentifier": "'vim.HostSystem:host-182'",
            "name": "11.11.12.13",
            "os": "VMware ESXi",
            "osVersion": "5.5.0",
            "ramMb": 65512,
            "totalCpuCores": 16,
            "totalCpuSockets": 2,
            "totalCpuThreads": 16,
            "type": "vmware",
            "vms": {
                "vCenter": "564d6d90-459c-2256-8f39-3cb2bd24b7b0"
            }
        },
        "10.11.12.14": {
            "cpuArch": "x86_64",
            "cpuDescription": "AMD Opteron(tm) Processor 4386",
            "cpuMhz": 3092.212639,
            "cpuVendor": "amd",
            "hostIdentifier": "'vim.HostSystem:host-183'",
            "name": "10.11.12.14",
            "os": "VMware ESXi",
            "osVersion": "5.5.0",
            "ramMb": 65512,
            "totalCpuCores": 16,
            "totalCpuSockets": 2,
            "totalCpuThreads": 16,
            "type": "vmware",
            "vms": {
                "49737e0a-c9e6-4ceb-aef8-6a9452f67cb5": "4230c60f-3f98-2a65-f7c3-
600b26b79c22",
                "5a2e4e63-a957-426b-bfa8-4169302e4fdb": "42307b15-1618-0595-01f2-
427ffcddd88e",
                "NSX-gateway": "4230d43e-aafe-38ba-5a9e-3cb67c03a16a",
                "NSX-l3gateway": "4230b00f-0b21-0e9d-dfde-6c7b06909d5f",
                "NSX-service": "4230e924-b714-198b-348b-25de01482fd9"
            }
        }
    }
}
```

For more information, see the man page on your SUSE Manager server for `virtual-host-gatherer`:

```
man virtual-host-gatherer
```

The `README` file of that package provides background information about the `type` of a hypervisor, etc.:

```
/usr/share/doc/packages/virtual-host-gatherer/README.md
```

The man page and the `README` file also contain example configuration files.

Software Channels

Channels are a method of grouping software packages. In SUSE Manager, channels are divided into base channels and child channels. Organizing channels in this way ensures that only compatible packages are installed on each system.

A base channel consists of packages built for a specific operating system type, version, and architecture. For example, all of the packages in SUSE Linux Enterprise Server 12 for the `x86_64` architecture make up a base channel. The list of packages in SUSE Linux Enterprise Server 12 for the `s390x` architecture make up a different base channel. A system must be subscribed to only one base channel, which is assigned automatically during registration based on the SUSE Linux Enterprise release and system architecture. For paid channels provided by a vendor, you must have an associated subscription.

A child channel is associated with a specific base channel and provides only packages that are compatible with that base channel. A system can be subscribed to multiple child channels of its base channel. When a system has been assigned to a base channel, it is only possible for that system to install the related child channels. For example, if a system has been assigned to the SUSE Linux Enterprise Server 12 `x86_64` base channel, they will only be able to install or update packages compatible with SUSE Linux Enterprise Server 12 `x86_64`.

In the SUSE Manager Web UI you can browse your available channels by navigating to **Software › Channels**. You can modify or create new channels by navigating to **Software › Manage Software Channels**.

Custom Channels

If you require packages that are not provided by the standard SUSE Manager base channels, you can create custom channels. SUSE Manager Administrators and Channel Administrators have channel management authority, which gives them the ability to create and manage their own custom channels.

For more on creating custom channels, see **[Administration › Channel-management ›]**.

Creating the SUSE Manager Tools Repository

A tools repository contains packages for installing Salt on clients, as well as the required packages for registering traditional clients during bootstrapping. You can create a tools repository on the SUSE Manager Server.

When you have created the tools repository, the packages in the repository will be installed during client registration.

Preparing to Create a Tools Repository

Before you create the tools repository, ensure client is fully synchronized with your vendor channel.

There are two ways to check if a channel has finished synchronizing:

- In the SUSE ManagerWeb UI, navigate to **Admin › Setup Wizard** and select the **SUSE Products** tab. This dialog displays a completion bar for each product when they are being synchronized.
- You can also check the synchronization log file at the command prompt. Use the `cat` or `tail -f` command to view the `/var/log/rhn/reposync/channel-label.log` file. If you use this method, remember that base channels can contain multiple child channels. Each of the child channels will generate its own log during the synchronization progress. You will need to check all the base and child channel log files to be sure that the synchronization is complete.

Generate a Tools Repository

Procedure: Generating the Tools Repository for SUSE Linux Enterprise

1. At the command prompt on the SUSE Manager Server, as root, list the available bootstrap repositories:

```
mgr-create-bootstrap-repo -l
```

2. Create the bootstrap repository, using the appropriate repository name as the product label:

```
mgr-create-bootstrap-repo -c SLE-version-x86_64
```

The client tools repository is located in `/srv/www/htdocs/pub/repositories/`.

Procedure: Specify a Bootstrap Repository

If you have mirrored more than one SUSE Linux Enterprise 15 Product (for example, SLES and SLES for SAP), you can specify the one you are actually interested in.

1. Check what bootstrap repositories you have available:

```
mgr-create-bootstrap-repo -c SLE-15-x86_64 --with-custom-channel
Multiple options for parent channel found. Please use option
--with-parent-channel <label> and choose one of:
- sle-product-sles15-pool-x86_64
- sle-product-sles_sap15-pool-x86_64
- sle-product-sled15-pool-x86_64
```

2. Specify the appropriate repository:

```
mgr-create-bootstrap-repo -c SLE-15-x86_64 --with-parent-channel sle-product-sled15-
pool-x86_64
```

Contact Methods

There are a number of ways that the SUSE Manager Server can communicate with traditional and Salt clients. Which one you use depends on your network architecture.

SUSE Manager Daemon (rhnsd)

The SUSE Manager daemon (`rhnsd`) runs on traditional client systems and periodically connects with SUSE Manager to check for new updates and notifications. It does not apply to Salt clients.

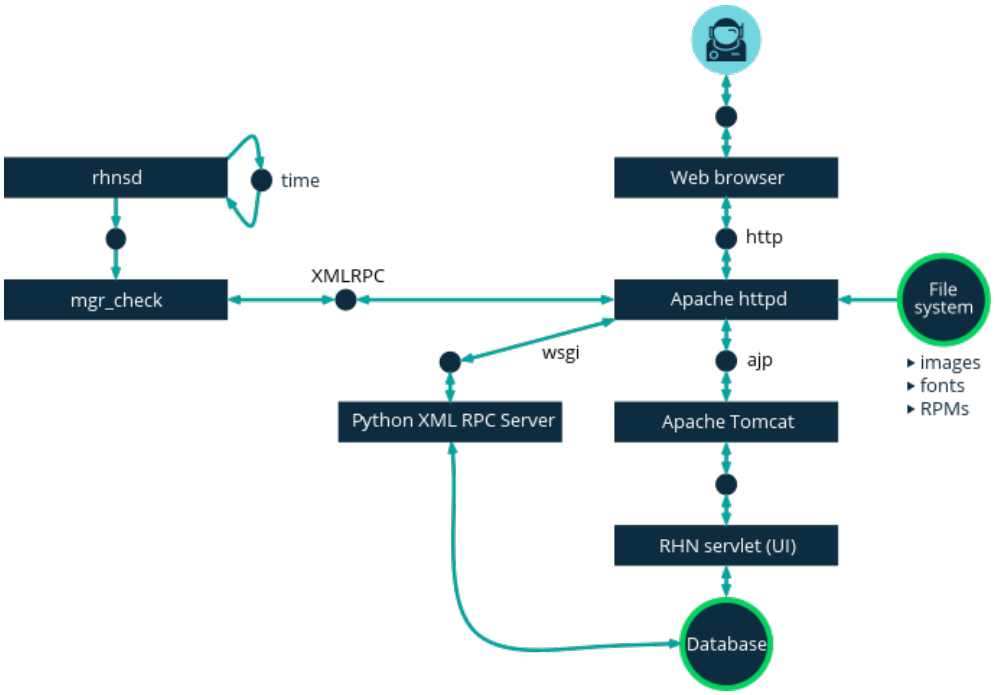
It is only used on SUSE Linux Enterprise 11 and Red Hat Enterprise Linux Server 6, as these systems do not use systemd. On later operating systems, a systemd timer (`rhnsd.timer`) is used and controlled by `rhnsd.service`.

Start the daemon with `/etc/init.d/rhnsd`.

By default, it will check every four hours for new actions. This means it can take some time for clients to execute scheduled actions.

To check for updates, `rhnsd` runs the external `mgr_check` program located in `/usr/sbin/`. This is a small application that establishes the network connection to SUSE Manager. The SUSE Manager daemon does not listen on any network ports or talk to the network directly. All network activity is performed by the `mgr_check` utility.

This figure provides an overview of the default `rhnsd` process path. All items left of the `Python XMLRPC server` block represent processes running on a SUSE Manager client.



Configure rhnsd

The `rhnsd` initialization script has a configuration file on the client system at `/etc/sysconfig/rhn/rhnsd`.

An important parameter for the daemon is its check-in frequency. The default interval time is four hours (240 minutes). The minimum allowed time interval is one hour (60 minutes). If you set the interval below one hour, it will change back to the default of 4 hours (240 minutes).

On SUSE Linux Enterprise 12 and later, the default time interval is set in `/etc/systemd/system/timers.target.wants/rhnsd.timer`, in this section:

```
[Timer]
OnCalendar=00/4:00
RandomizedDelaySec=30min
```

You can create an overriding drop-in file for `rhnsd.timer` using `systemctl`:

```
systemctl edit rhnsd.timer
```

For example, if you want configure a two hour time interval:

```
[Timer]
OnCalendar=00/2:00
```

The file will be saved as `/etc/systemd/system/rhnsd.timer.d/override.conf`.

For more information about system timers, see the `systemd.timer` and `systemctl` manpages.

If you modify the `rhnsd` configuration file, execute this command as root to restart the daemon and pick up your changes:

```
/etc/init.d/rhnsd restart
```

To see the status of `rhnsd`, use this command as root:

```
/etc/init.d/rhnsd status
```

Push via SSH

Push via SSH is used in environments where traditional clients cannot reach the SUSE Manager Server directly. In this environment, clients are located in a firewall-protected zone called a DMZ. No system within the DMZ is authorized to open a connection to the internal network, including the SUSE Manager

Server.

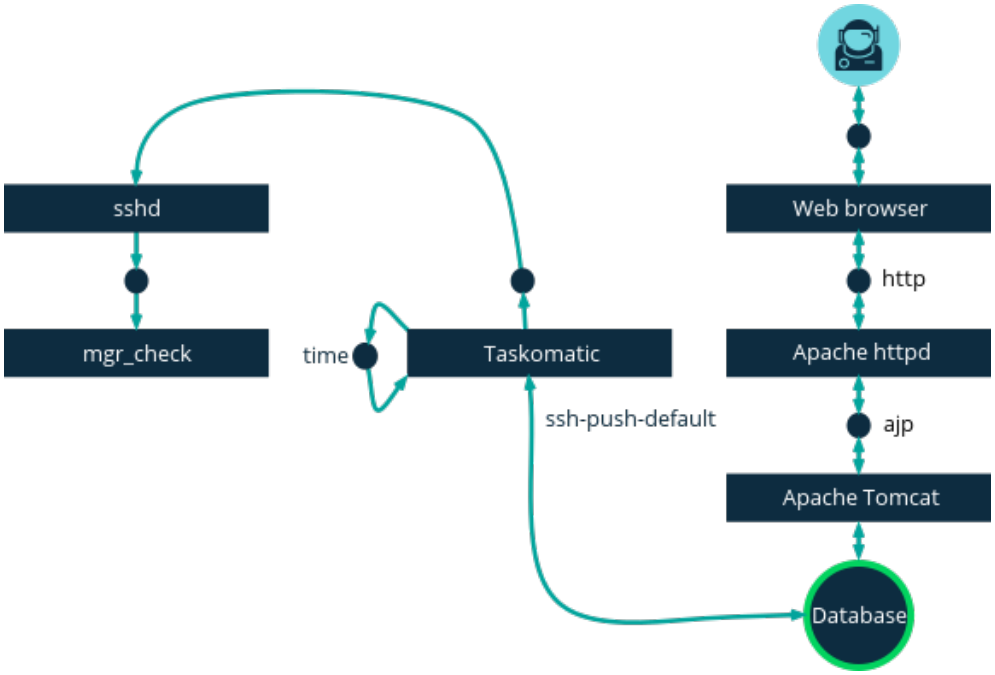
The Push via SSH method creates an encrypted tunnel from the SUSE Manager Server on the internal network to the clients located on the DMZ. After all actions and events are executed, the tunnel is closed.

The server uses SSH to contact the clients at regular intervals, checking in and performing scheduled actions and events.

This contact method works for traditional clients only. For Salt clients, use Push via Salt SSH.

Re-installing systems using the provisioning model is not currently supported on clients managed with push via SSH.

This image demonstrates the push via SSH process path. All items left of the `Taskomatic` block represent processes running on a SUSE Manager client.



For tunneling connections via SSH, two available port numbers are required, one for tunneling HTTP and the second for tunneling via HTTPS (HTTP is only necessary during the registration process). The port numbers used by default are `1232` and `1233`. To overwrite these, you can add two custom port numbers greater than 1024 to `/etc/rhn/rhn.conf`:

```
ssh_push_port_http = high_port_1
ssh_push_port_https = high_port_2
```

If you would like your clients to be contacted using their hostnames instead of an IP address, set this option:

```
ssh_push_use_hostname = true
```

It is also possible to adjust the number of threads to use for opening client connections in parallel. By default two parallel threads are used. Set `taskomatic.ssh_push_workers` in `/etc/rhn/rhn.conf`:

```
taskomatic.ssh_push_workers = number
```

For security reasons, you might want to use sudo with SSH, to access the system as an unprivileged user instead of as root.

Procedure: Configuring Unprivileged SSH Access

1. Ensure you have the latest `spacewalk-taskomatic` and `spacewalk-certs-tools` packages installed on the SUSE Manager Server.
2. On each client system, create an appropriate unprivileged user on each client system.
3. On each client system, open the `/etc/sudoers` file and comment out these lines:

   ```
   #Defaults targetpw   # ask for the password of the target user i.e. root
   #ALL    ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults targetpw'!
   ```

4. On each client system, in the `User privilege specification` section, add these lines:

   ```
   <user> ALL=(ALL) NOPASSWD:/usr/sbin/mgr_check
   <user> ALL=(ALL) NOPASSWD:/home/<user>/enable.sh
   <user> ALL=(ALL) NOPASSWD:/home/<user>/bootstrap.sh
   ```

5. On each client system, in the `/home/user/.bashrc` file, add these lines:

   ```
   PATH=$PATH:/usr/sbin
   export PATH
   ```

6. On the SUSE Manager Server, in the `/etc/rhn/rhn.conf` configuration file, add or amend this line to include the unprivileged username:

   ```
   ssh_push_sudo_user = <user>
   ```

Because clients are in the DMZ and cannot reach the server, you need to use the `mgr-ssh-push-init` tool to register them with the SUSE Manager Server.

To use the tool, you will need the client hostname or IP address, and the path to a valid bootstrap script on the SUSE Manager Server. For more information about bootstrapping, see **[Client-configuration › Registration-bootstrap ›]**.

The bootstrap script will need to have an activation key associated with it that is configured for Push via

SSH. For more information on activation keys, see **[Client-configuration › Clients-and-activation-keys ›]**.

Before you begin, you need to ensure that you have specified which ports to use for SSH tunneling. If you have registered clients before changing the port numbers, they will need to be registered again.

Clients that are managed with Push via SSH cannot reach the server directly. When you use the `mgr-ssh-push-init` tool, the `rhnsd` daemon is disabled.

Procedure: Registering Clients with Push via SSH

1. At the command prompt on the SUSE Manager Server, as root, execute this command:

```
# mgr-ssh-push-init --client <client> --register \
/srv/www/htdocs/pub/bootstrap/bootstrap_script --tunnel
```

   OPTIONAL: You can remove the `--tunnel` option, if you do not want to use tunneling.

2. Verify that the SSH connection is active:

```
# ssh -i /root/.ssh/id_susemanager -R <high_port>:<susemanager>:443 \
<client> zypper ref
```

Example: API Access to Push via SSH

You can use the API to manage which contact method to use. This example Python code sets the contact method to `ssh-push`.

Valid values are:

- `default` (pull)
- `ssh-push`
- `ssh-push-tunnel`

```
client = xmlrpclib.Server(SUMA_HOST + "/rpc/api", verbose=0)
key = client.auth.login(SUMA_LOGIN, SUMA_PASSWORD)
client.system.setDetails(key, 1000012345, {'contact_method' : 'ssh-push'})
```

If you have a client that has already been registered, and you want to migrate it to use Push via SSH, some extra steps are required. You can use the `mgr-ssh-push-init` tool to set up your client.

Procedure: Migrating Registered Systems to Push via SSH

1. At the command prompt on the SUSE Manager Server, as root, set up the client:

```
# mgr-ssh-push-init --client <client> \
/srv/www/htdocs/pub/bootstrap/bootstrap_script --tunnel
```

2. Using the SUSE Manager Web UI, change the client's contact method to `ssh-push` or `ssh-push-tunnel`.
3. OPTIONAL: If you need to edit an existing activation key, you can do so with this command:

```
client.activationkey.setDetails(key, '1-mykey', {'contact_method' : 'ssh-push'})
```

You can also use Push via SSH for clients that connect using a SUSE Manager Proxy. Ensure your proxy is updated before you begin.

Procedure: Registering Clients with Push via SSH to a Proxy

1. At the command prompt on the SUSE Manager Proxy, as root, set up the client:

```
# mgr-ssh-push-init --client <client> \
/srv/www/htdocs/pub/bootstrap/bootstrap_script --tunnel
```

2. At the command prompt on the SUSE Manager Server, copy the SSH key to the proxy:

```
mgr-ssh-push-init --client <proxy>
```

Push via Salt SSH

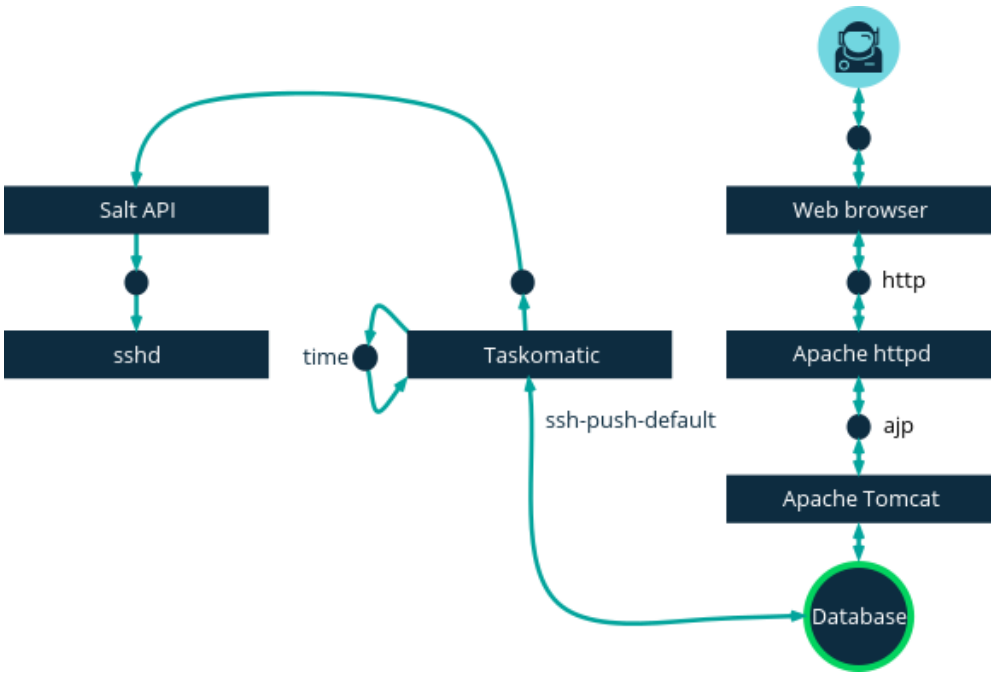
Push via Salt SSH is used in environments where Salt clients cannot reach the SUSE Manager Server directly. In this environment, clients are located in a firewall-protected zone called a DMZ. No system within the DMZ is authorized to open a connection to the internal network, including the SUSE Manager Server.

The Push via Salt SSH method creates an encrypted tunnel from the SUSE Manager Server on the internal network to the clients located on the DMZ. After all actions and events are executed, the tunnel is closed.

The server uses the `salt-ssh` tool to contact the clients at regular intervals, checking in and performing scheduled actions and events. For more information about Salt SSH, see **[Salt › Salt-ssh ›]**.

This contact method works for Salt clients only. For traditional clients, use Push via SSH.

This image demonstrates the Push via Salt SSH process path. All items left of the `Taskomatic` block represent processes running on a SUSE Manager client.



To use Push via Salt SSH, you must have the SSH daemon running on the client, and reachable by the `salt-api` daemon running on the SUSE Manager Server. Additionally, Python must be available on the remote system, and be a version supported by Salt.

Red Hat Enterprise Linux 5, {centos} 5, and earlier are not supported, as they use unsupported versions of Python.

Procedure: Registering Clients with Push via Salt SSH

1. In the SUSE Manager Web UI, navigate to **Systems › Bootstrapping** and complete the appropriate fields.
2. Select an activation key with the Push via SSH contact method configured. For more information about activation keys, see **[Client-configuration › Clients-and-activation-keys ›]**.
3. Check the `Manage system completely via SSH` checkbox.
4. Click **`[Bootstrap]`** to begin registration.
5. Confirm that the system has been registered correctly by navigating to **Systems › Overview**.

When you are configuring Push via Salt SSH, you can modify parameters that are used when a system is registered, including the host, activation key, and password. The password is used only for bootstrapping, it is not saved anywhere. All future SSH sessions are authorized via a key/certificate pair. These parameters are configured in **Systems › Bootstrapping**.

You can also configure persistent parameters that are are used system-wide, including the sudo user. For more information on configuring the sudo user, see the Push via SSH section in this chapter.

The Push via Salt SSH feature uses taskomatic to execute scheduled actions using `salt-ssh`. The taskomatic job periodically checks for scheduled actions and executes them. Unlike Push via SSH on traditional clients, the Push via Salt SSH feature executes a complete `salt-ssh` call based on the

scheduled action.

There are some features that are not yet supported on Push via Salt SSH. These features will not work on Salt SSH clients:

- OpenSCAP auditing
- Beacons, resulting in:
  - Installing a package on a system using `zypper` will not invoke the package refresh.
  - Virtual Host functions (for example, a host to guests) will not work if the virtual host system is Salt SSH-based.

For more information about Salt SSH, see https://docs.saltstack.com/en/latest/topics/ssh/.

OSAD

OSAD is an alternative contact method between SUSE Manager and its clients. By default, SUSE Manager uses `rhnsd`, which contacts the server every four hours to execute scheduled actions. OSAD allows registered client systems to execute scheduled actions immediately.

Use OSAD in addition to `rhnsd`. If you disable `rhnsd` your client will be shown as not checking in after 24 hours.

OSAD has several distinct components:

- The `osa-dispatcher` service runs on the server, and uses database checks to determine if clients need to be pinged, or if actions need to be executed.
- The `osad` service runs on the client. It responds to pings from `osa-dispatcher` and runs `mgr_check` to execute actions when directed to do so.
- The `jabberd` service is a daemon that uses the `XMPP` protocol for communication between the client and the server. The `jabberd` service also handles authentication.
- The `mgr_check` tool runs on the client to execute actions. It is triggered by communication from the `osa-dispatcher` service.

The `osa-dispatcher` periodically runs a query to check when clients last showed network activity. If it finds a client that has not shown activity recently, it will use `jabberd` to ping all `osad` instances running on all clients registered with your SUSE Manager server. The `osad` instances respond to the ping using `jabberd`, which is running in the background on the server. When the `osa-dispatcher` receives the response, it marks the client as online. If the `osa-dispatcher` fails to receive a response within a certain period of time, it marks the client as offline.

When you schedule actions on an OSAD-enabled system, the task will be carried out immediately. The `osa-dispatcher` periodically checks clients for actions that need to be executed. If an outstanding action is found, it uses `jabberd` to execute `mgr_check` on the client, which will then execute the action.

OSAD clients use the fully qualified domain name (FQDN) of the server to communicate with the `osa-dispatcher` service.

SSL is required for `osad` communication. If SSL certificates are not available, the daemon on your client systems will fail to connect. Make sure your firewall rules are set to allow the required ports. For more information, see [tab.install.ports.server].

Procedure: Enabling OSAD

1. At the command prompt on the SUSE Manager Server, as root, start the `osa-dispatcher` service:

```
systemctl start osa-dispatcher
```

2. On each client, install the `mgr-osad` package from the `Tools` child channel. The `mgr-osad` package should be installed on clients only. If you install the `mgr-osad` package on your SUSE Manager Server, it will conflict with the `osa-dispatcher` package.
3. On each client, as root, start the `osad` service:

```
systemctl start osad
```

   Because `osad` and `osa-dispatcher` are run as services, you can use standard commands to manage them, including `stop`, `restart`, and `status`.

Each OSAD component is configured using local configuration files. We recommend you keep the default configuration parameters for all OSAD components.

Component	Location	Path to Configuration File
`osa-dispatcher`	Server	`/etc/rhn/rhn.conf` Section: `OSA configuration`
`osad`	Client	`/etc/sysconfig/rhn/osad.conf`
`osad` log file	Client	`/var/log/osad`
`jabberd` log file	Both	`/var/log/messages`

Troubleshooting OSAD

If your OSAD clients cannot connect to the server, or if the `jabberd` service takes a lot of time responding to port 5552, it could be because you have exceeded the open file count.

Every client needs one always-open TCP connection to the server, which consumes a single file handler. If the number of file handlers currently open exceeds the maximum number of files that `jabberd` is allowed to use, `jabberd` will queue the requests, and refuse connections.

To resolve this issue, you can increase the file limits for `jabberd` by editing the `/etc/security/limits.conf` configuration file and adding these lines:

```
jabbersoftnofile5100
jabberhardnofile6000
```

Calculate the limits required for your environment by adding 100 to the number of clients for the soft limit, and 1000 to the current number of clients for the soft limit. In the example above, we have assumed 500 current clients, so the soft limit is 5100, and the hard limit is 6000.

You will also need to update the `max_fds` parameter in the `/etc/jabberd/c2s.xml` file with your chosen hard limit:

```
<max_fds>6000</max_fds>
```

Using the System Set Manager

System Set Manager is used to administrate groups of systems, rather than performing actions on one system at a time. It works for both Salt and traditional clients.

For a complete list of the tasks that you can perform with the System Set Manager, see **[Reference › Systems ›]**.

Setting up System Set Manager

You need to select which systems you want to work with before you can use System Set Manager to perform operations.

Navigate to **Main Menu › Systems › System List › All** and check the boxes to the left of the systems you want to work with. This will automatically add your chosen systems to System Set Manager.

You can access System Set Manager in three different ways:

- Navigating to **Systems › System Set Manager**.
- Navigating to **Systems › System Groups** and clicking **`[Use in SSM]`** for the system group you want to work with.
- Navigating to **Systems › System Groups**, selecting the group you want to work with, and clicking **`[Work with Group]`**.

Using System Set Manager

The details you see in System Set Manager might differ slightly from the details available in other parts of the SUSE Manager Web UI. If you are looking at the details of a single system in the Web UI, then you will only be able to see the latest available versions of package updates. When you look at the same system in System Set Manager, all available versions will be shown. This is intended to make it easier for system administrators to manage package versions, and choose to upgrade to packages that might not be the latest version.

Troubleshooting Clients

Bare Metal Systems

If a bare metal system on the network is not automatically added to the `Systems` list, check these things first:

- You must have the `pxe-default-image` package installed.
- File paths and parameters must be configured correctly. Check that the `vmlinuz0` and `initrd0.img` files, which are provided by `pxe-default-image`, are in the locations specified in the `rhn.conf` configuration file.
- Ensure the networking equipment connecting the bare metal system to the SUSE Manager server is working correctly, and that you can reach the SUSE Manager server IP address from the server.
- The bare metal system to be provisioned must have PXE booting enabled in the boot sequence, and must not be attempting to boot an operating system.
- The DHCP server must be responding to DHCP requests during boot. Check the PXE boot messages to ensure that:
  - the DHCP server is assigning the expected IP address
  - the DHCP server is assigning the the SUSE Manager server IP address as `next-server` for booting.
- Ensure Cobbler is running, and that the Discovery feature is enabled.

If you see a blue Cobbler menu shortly after booting, discovery has started. If it does not complete successfully, temporarily disable automatic shutdown in order to help diagnose the problem. To disable automatic shutdown:

1. Select `pxe-default-profile` in the Cobbler menu with the arrow keys, and press the Tab key before the timer expires.
2. Add the kernel boot parameter `spacewalk-finally=running` using the integrated editor, and press Enter to continue booting.
3. Enter a shell with the username `root` and password `linux` to continue debugging.

Duplicate profiles

Due to a technical limitation, it is not possible to reliably distinguish a new bare metal system from a system that has previously been discovered. Therefore, we recommended that you do not power on bare metal systems multiple times, as this will result in duplicate profiles.

Cloned Salt Clients

If you have used your hypervisor clone utility, and attempted to register the cloned Salt client, you might get this error:

```
We're sorry, but the system could not be found.
```

This is caused by the new, cloned, system having the same machine ID as an existing, registered, system. You can adjust this manually to correct the error and register the cloned system successfully.

For more information and instructions, see **[Administration › Tshoot-registerclones ›]**.

Mounting /tmp with noexec

Salt runs remote commands from `/tmp` on the client's filesystem. Therefore you must not mount `/tmp` with the `noexec` option.

SSL errors

On SLES 11 systems, clients can sometimes have SSL errors which make some operations unusable, including package management and bootstrapping. In this case, you will see an error like this:

```
Repository 'SLES11-SP4-SUSE-Manager-Tools x86_64' is invalid.
[|] Valid metadata not found at specified URL(s)
Please check if the URIs defined for this repository are pointing to a valid repository.
Skipping repository 'SLES11-SP4-SUSE-Manager-Tools x86_64' because of the above error.
Download (curl) error for 'www.example.com':
Error code: Unrecognized error
Error message: error:1409442E:SSL routines:SSL3_READ_BYTES:tlsv1 alert protocol version
```

This occurs because Apache requires TLS v1.2, but older versions of SLES do not support this version of the TLS protocol. To fix this error, you need to force Apache to accept a greater range of protocol versions. Open the `/etc/apache2/ssl-global.conf` configuration file, locate the `SSLProtocol` line, and update it to read:

```
SSLProtocol all -SSLv2 -SSLv3
```

This will need to be done manually on the server, and with a Salt state on the Proxy. Restart the `apache` service on each system after making the changes.

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{ldquo}GNU
   Free Documentation License{rdquo}.
```

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.