



Administration Guide

SUSE Manager 4.0

June 03, 2019



Table of Contents

GNU Free Documentation License	1
Introduction	8
Image Building and Management	9
Image Building Overview	9
Container Images	9
Requirements	9
Creating a Build Host	10
Creating an Image Store	11
Creating an Image Profile	12
Example Dockerfile and add_packages Script	14
Building an Image	16
Importing an Image	16
Troubleshooting	17
OS Images	17
Requirements	18
Creating a Build Host	18
Image Store	21
Creating an Image Profile	22
Example of Kiwi sources	23
Building an Image	24
Image Inspection and Salt Integration	25
Troubleshooting	25
Limitations	26
Listing Image Profiles Available for Building	26
Live Patching with SUSE Manager	27
Live Patching on SLES 15	27
Live Patching on SLES 12	29
Monitoring with Prometheus	33
Prometheus Metrics	33
PromQL	33
Exporters	34
Install and Configure Prometheus	35
Installing Prometheus	35
Configuring Prometheus	35
Monitoring Managed Systems	36
Enable and Configure Monitoring	36
Set up Visualization with Grafana	37
Kubernetes	39
Prerequisites	39
Requirements	39
Register Kubernetes as a Virtual Host Manager	39
View the List of Nodes in a Cluster	39
Obtain Runtime Data about Images	40
Build an image for deployment in Kubernetes	40
Import a Previously Deployed Image in Kubernetes	40
Obtain Additional Runtime Data	41

Rebuild a Previously Deployed Image in Kubernetes	41
Role Based Access Control Permissions and Certificate Data	41
Public Cloud	43
Instance Requirements	43
Network Setup	43
Set the hostname	44
Set up DNS resolution	45

GNU Free Documentation License

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

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

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

then there are none.

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

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

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

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

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

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

2. VERBATIM COPYING

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

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

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

-
- D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retile any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

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

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

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

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

5. COMBINING DOCUMENTS

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

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

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

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

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

ADDENDUM: How to use this License for your documents

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

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

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

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

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

Introduction

This book provides guidance on performing common Administrative tasks on SUSE Manager.

Image Building and Management

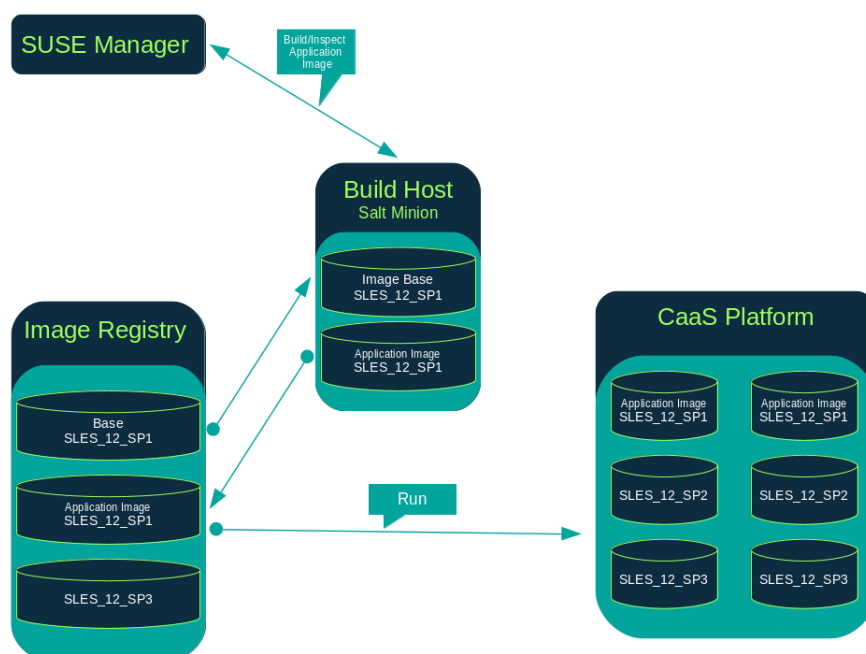
Image Building Overview

SUSE Manager enables system administrators to build containers, systems, and virtual images. SUSE Manager helps with creating Image Stores and managing Image Profiles.

SUSE Manager supports two distinct build types:

- Dockerfile-for more information, see [Container Images](#)
- Kiwi image system-for more information, see [OS Images](#)

Container Images



Requirements

The containers feature is available for Salt minions running SUSE Linux Enterprise Server 12 or later. Before you begin, ensure your environment meets these requirements:

- An existing external GitHub or internal GitLab repository containing a Dockerfile and configuration scripts (example scripts are provided in this chapter).
- A properly configured image registry.



Registry Provider Solutions

If you require a private image registry you can use an open source solution such as [Portus](#). For additional information on setting up Portus as a registry provider, see the [Portus Documentation](#).

For more information on Containers or CaaS Platform, see:

- [SUSE Linux Enterprise Server 12 SP3 Docker Guide](#)
- [SUSE CaaS Platform 2 Documentation](#)

Creating a Build Host

To build images with SUSE Manager, you will need to create and configure a build host. Container build hosts are Salt minions running SUSE Linux Enterprise 12 or later. This section guides you through the initial configuration for a build host.

From the SUSE Manager Web UI perform these steps to configure a build host.

1. Select a minion to be designated as a build host from the **Systems > Overview** page.
2. From the [System Details](#) page for the selected minion assign the containers modules by going to **Software > Software Channels** and enabling [SLE-Module-Containers12-Pool](#) and [SLE-Module-Containers12-Updates](#). Confirm by clicking [**Change Subscriptions**].
3. From the **System Details > Properties** page, enable [Add-on System Type](#) and [Container Build Host](#) and confirm by clicking [**Update Properties**].
4. Install all required packages by applying [Highstate](#). From the system details page select **States > Highstate** and click [Apply Highstate](#). Alternatively, apply Highstate from the SUSE Manager Server command line:

```
salt '$your_minion' state.highstate
```

Define Container Build Channels with an Activation Key

Create an activation key associated with the channel that your images will use.



Relationship Between Activation Keys and Image Profiles

To build containers, you will need an activation key that is associated with a channel other than "SUSE Manager Default".

Create Activation Key ?

Activation Key Details

Systems registered with this activation key will inherit the settings listed below.

Description:
Use this to describe what kind of settings this key will reflect on systems that use it. If left blank, this field will be filled in 'None'.

Key: 1-
Activation key can contains only numbers [0-9], letters [a-z A-Z], ':', '_' and '-'.
Leave blank for automatic key generation. Note that the prefix is an indication of the SUSE Manager organization the key is associated with.

Usage:
Leave blank for unlimited use.

Base Channel:
Choose "SUSE Manager Default" to allow systems to register to the default SUSE Manager provided channel that corresponds to the installed SUSE Linux version. Instead of the default, you may choose a particular SUSE provided channel or a custom base channel, but if a system using this key is not compatible with the selected channel, it will fall back to its SUSE Manager Default channel.

Add-On System Types: ☐ Container Build Host ☐ Virtualization Host

Contact Method:

Universal Default: ☐
Tip: Only one universal default activation key may be set for this organization. By setting this key as universal default, you will remove universal default status from the current universal default key if it exists. If this key is set as universal default, then newly-registered systems to your organization will inherit the properties of this key.

Create Activation Key

1. Select **Main Menu** > **Systems** > **Activation Keys**.
2. Click [**Create Key**].
3. Enter a **Description** and a **Key** name. Use the drop-down menu to select the **Base Channel** to associate with this key.
4. Confirm with [**Create Activation Key**].

For more information, see [\[bp.key.management\]](https://bp.key.management).

Creating an Image Store

Define a location to store all of your images by creating an Image Store.

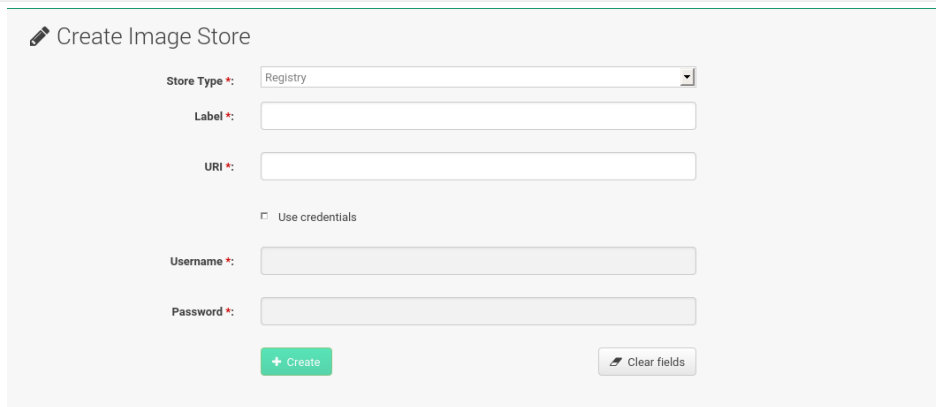
Image Stores ?

Items 0 - 0 of 0 [Select All](#) Items per page

There are no entries to show.

Page 1 of 1

1. Select **Main Menu** > **Images** > **Stores**.
2. Click **Create** to create a new store.



Create Image Store

Store Type *: Registry

Label *:

URI *:

☐ Use credentials

Username *:

Password *:

[+ Create](#) [Clear fields](#)

3. SUSE Manager currently provides support only for the **Registry** store type. Define a name for the image store in the **Label** field.
4. Provide the path to your image registry by filling in the **URI** field, as a fully qualified domain name (FQDN) for the container registry host (whether internal or external).

registry.example.com

The Registry URI can also be used to specify an image store on a used registry.

registry.example.com:5000/myregistry/myproject

5. Click [**Create**] to add the new image store.

Creating an Image Profile

Manage Image Profiles from the **Image Profile** page.

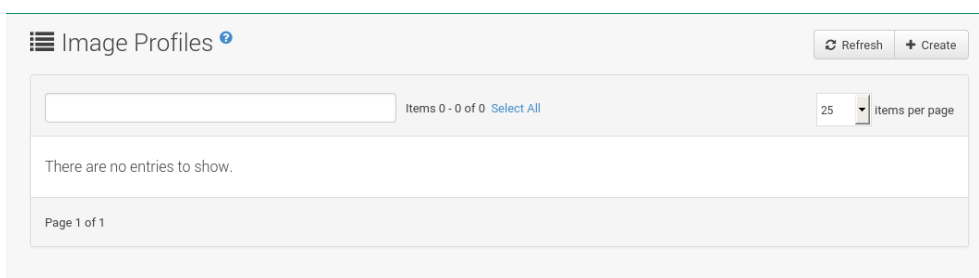


Image Profiles ?

[Refresh](#) [+ Create](#)

Items 0 - 0 of 0 [Select All](#)

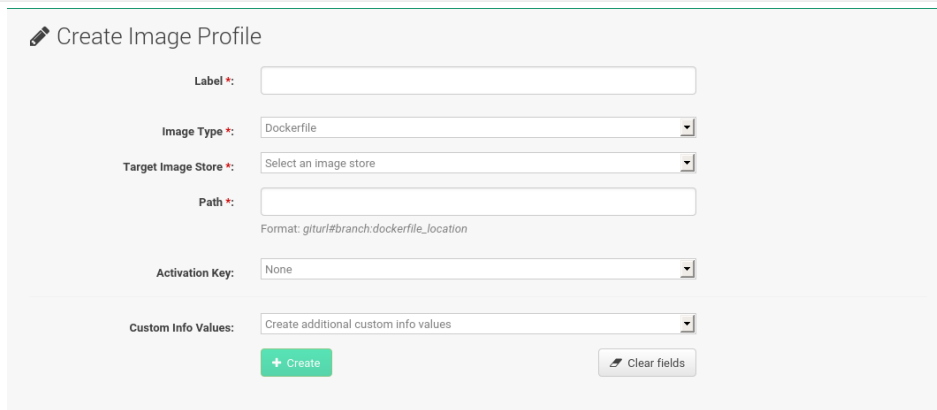
25 items per page

There are no entries to show.

Page 1 of 1

Procedure: Create an Image Profile

1. To create an image profile select **Image > Profiles** and click [**Create**].



Create Image Profile

Label *:

Image Type *:

Target Image Store *:

Path *:
Format: giturl#branch:dockerfile_location

Activation Key:

Custom Info Values:

2. Provide a name for the image profile by filling in the **Label** field.



Only lowercase characters are permitted in container labels. If your container image tag is in a format such as `myproject/myimage`, make sure your image store registry URI contains the `/myproject` suffix.

3. Use a **Dockerfile** as the **Image Type**
4. Use the drop-down menu to select your registry from the **Target Image Store** field.
5. Enter a Github or Gitlab repository URL (http, https, or token authentication) in the **Path** field using one of the following formats:

Github Path Options

- Github single user project repository

```
https://github.com/USER/project.git#branchname:folder
```

- Github organization project repository

```
https://github.com/ORG/project.git#branchname:folder
```

- Github token authentication:

If your git repository is private and not publicly accessible, you need to modify the profile's git URL to include authentication. Use this URL format to authenticate with a Github token:

```
https://USER:<AUTHENTICATION_TOKEN>@github.com/USER/project.git#master:/container/
```

Gitlab Path Options

- Gitlab single user project repository

```
https://gitlab.example.com/USER/project.git#master:/container/
```


- Gitlab groups project repository

```
https://gitlab.example.com/GROUP/project.git#master:/container/
```

- Gitlab token authentication If your git repository is private and not publicly accessible, you need to modify the profile's git URL to include authentication. Use this URL format to authenticate with a Gitlab token:

```
https://gitlab-ci-token:<AUTHENTICATION_TOKEN>@gitlab.example.com/USER/project.git#master:/container/
```



Specifying a Github or Gitlab Branch

If a branch is not specified, the **master** branch will be used by default. If a **folder** is not specified the image sources (**Dockerfile** sources) are expected to be in the root directory of the Github or Gitlab checkout.

1. Select an **Activation Key**. Activation Keys ensure that images using a profile are assigned to the correct channel and packages.



Relationship Between Activation Keys and Image Profiles

When you associate an activation key with an image profile you are ensuring any image using the profile will use the correct software channel and any packages in the channel.

2. Click the [**Create**] button.

Example Dockerfile and add_packages Script

This section contains an example Dockerfile. You specify a Dockerfile that will be used during image building when creating an image profile. A Dockerfile and any associated scripts should be stored within an internal or external Github or Gitlab repository:



Required Dockerfile Lines

The Dockerfile provides access to a specific repository version served by SUSE Manager. This example Dockerfile is used by SUSE Manager to trigger a build job on a build host minion. The **ARG** parameters ensure that the image that is built is associated with the desired repository version served by SUSE Manager. The **ARG** parameters also allow you to build image versions of SUSE Linux Enterprise Server which may differ from the version of SUSE Linux Enterprise Server used by the build host itself.

For example: The **ARG repo** parameter and the **echo** command pointing to the repository file, creates and then injects the correct path into the repository file for the desired channel version.

The repository version is determined by the activation key that you assigned to your image profile.

```
FROM registry.example.com/sles12sp2
MAINTAINER Tux Administrator "tux@example.com"

### Begin: These lines Required for use with {productname}

ARG repo
ARG cert

# Add the correct certificate
RUN echo "$cert" > /etc/pki/trust/anchors/RHN-ORG-TRUSTED-SSL-CERT.pem

# Update certificate trust store
RUN update-ca-certificates

# Add the repository path to the image
RUN echo "$repo" > /etc/zypp/repos.d/susemanager:dockerbuild.repo

### End: These lines required for use with {productname}

# Add the package script
ADD add_packages.sh /root/add_packages.sh

# Run the package script
RUN /root/add_packages.sh

# After building remove the repository path from image
RUN rm -f /etc/zypp/repos.d/susemanager:dockerbuild.repo
```

This is an example **add_packages.sh** script for use with your Dockerfile:

```
#!/bin/bash
set -e

zypper --non-interactive --gpg-auto-import-keys ref

zypper --non-interactive in python python-xml aaa_base aaa_base-extras net-tools timezone vim
less sudo tar
```



Packages Required for Inspecting Your Images

To inspect images and provide the package and product list of a container to the SUSE Manager Web UI you will need to install python and python-xml within the container. If these packages remain uninstalled, your images will still build, but the package and product list will be unavailable from the Web UI.

Building an Image

There are two ways to build an image. You can select **Images > Build** from the left navigation bar, or click the build icon in the **Images > Profiles** list.

Procedure: Build an Image

1. For this example select **Images > Build**.
2. Add a different tag name if you want a version other than the default **latest** (only relevant to containers).
3. Select **Build Profile** and **Build Host**.



Profile Summary

Notice the **Profile Summary** to the right of the build fields. When you have selected a build profile, detailed information about the selected profile will be displayed in this area.

4. To schedule a build click the [**Build**] button.

Importing an Image

You can import and inspect arbitrary images. Select **Images > Images** from the left navigation bar. Complete the text boxes of the **Import** dialog. Once it has processed, the imported image will be listed on the **Images** page.

Procedure: Import an Image

1. From **Images > Images** click [**Import**] to open the **Import Image** dialog.
2. In the **Import Image** dialog complete these fields:

Image store

The registry from where the image will be pulled for inspection.

Image name

The name of the image in the registry.

Image version

The version of the image in the registry.

Build host

The build host that will pull and inspect the image.

Activation key

The activation key that provides the path to the software channel that the image will be inspected with.

For confirmation, click [**Import**].

The entry for the image is created in the database, and an **Inspect Image** action on SUSE Manager is scheduled.

Once it has been processed, you can find the imported image in the **Images** list. It has a different icon in the **Build** column, to indicate that the image is imported (see screenshot). The status icon for the imported image can also be seen on the **Overview** tab for the image.

Troubleshooting

These are some known problems that you might encounter when working with images:

- HTTPS certificates to access the registry or the git repositories should be deployed to the minion by a custom state file.
- SSH git access using Docker is currently unsupported. You may test it, but SUSE will not provide support.
- If the python and python-xml packages are not installed in your images during the build process, Salt cannot run within the container and reporting of installed packages or products will fail. This will result in an **unknown** update status.

OS Images

OS images are built by the Kiwi image system. They can be of various types: PXE, QCOW2, LiveCD images, and others.

For more information about the Kiwi build system, see the [Kiwi documentation](#).

Requirements

The Kiwi image building feature is available for Salt minions running SUSE Linux Enterprise Server 12. It is currently not supported to build SUSE Linux Enterprise 15 images.

Kiwi image configuration files and configuration scripts must be accessible in one of these locations:

- Git repository
- HTTP hosted tarball
- Local build host directory

Example scripts are provided in the following sections.



Hardware Requirements for Hosts Running OS Images

Hosts running OS images built with Kiwi need at least 1 GB of RAM. Disk space depends on the actual size of the image. For more information, see the documentation of the underlying system.

Creating a Build Host

To build all kinds of images with SUSE Manager, create and configure a build host. OS image build hosts are Salt minions running SUSE Linux Enterprise Server 12 (SP3 or later). This procedure will guide you through the initial configuration for a build host.

From the SUSE Manager Web UI perform these steps to configure a build host:

1. Select a minion that will be designated as a build host from the **Main Menu** > **Systems** > **Overview** page.
2. From the **System Details** > **Properties** page, enable the **Add-on System Type: OS Image Build Host** and confirm with [**Update Properties**].

The screenshot shows the 'Edit System Details' page in the SUSE Manager web interface. The system name is 'd186.suse.de'. The base system type is 'Salt'. Under 'Add-On System Types', 'OS Image Build Host' is selected. The description is 'OS Image Build Host (for KIWI Images)'. There are input fields for Facility Address, City, State/Province, Country (set to 'None'), and Building.

3. From the **System Details** > **Software** > **Software Channels** page, enable **SLE-Manager-Tools12-Pool** and **SLE-Manager-Tools12-Updates** (or a later version). Schedule and click [**Confirm**].
4. Install Kiwi and all required packages by applying Highstate. From the system details page select **States** > **Highstate** and click [**Apply Highstate**]. Alternatively, apply Highstate from the SUSE Manager Server command line:

```
salt '$your_minion' state.highstate
```

SUSE Manager Web Server Public Certificate RPM

Build host provisioning copies the SUSE Manager certificate RPM to the build host. This certificate is used for accessing repositories provided by SUSE Manager.

The certificate is packaged in RPM by the **mgr-package-rpm-certificate-osimage** package script. The package script is called automatically during a new SUSE Manager installation.

When you upgrade the **spacewalk-certs-tools** package, the upgrade scenario will call the package script using the default values. However if the certificate path was changed or unavailable, you will need to call the package script manually using **--ca-cert-full-path <path_to_certificate>** after the upgrade procedure has finished.

Listing 1. Package script call example

```
/usr/sbin/mgr-package-rpm-certificate-osimage --ca-cert-full-path /root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT
```

The RPM package with the certificate is stored in a salt-accessible directory such as `/usr/share/susemanager/salt/images/rhn-org-trusted-ssl-cert-osimage-1.0-1.noarch.rpm`.

The RPM package with the certificate is provided in the local build host repository `/var/lib/Kiwi/repo`.

The RPM Package with the SUSE Manager Certificate Must Be Specified in the Build Source

Make sure your build source Kiwi configuration contains `rhn-org-trusted-ssl-cert-osimage` as a required package in the `bootstrap` section.

Listing 2. config.xml




```
...  
  <packages type="bootstrap">  
    ...  
    <package name="rhn-org-trusted-ssl-cert-osimage"  
      bootinclude="true"/>  
  </packages>  
  ...
```



Define Kiwi Build Channels with an Activation Key

Create an activation key associated with the channel that your images will use. Activation keys are mandatory for OS Image building.

Relationship Between Activation Keys and Image Profiles



To build OS Images, you will need an activation key that is associated with a channel other than "SUSE Manager Default".

 **Create Activation Key** 

Activation Key Details

Systems registered with this activation key will inherit the settings listed below.

Description:
 Use this to describe what kind of settings this key will reflect on systems that use it. If left blank, this field will be filled in 'None'.

Key:
 Activation key can contains only numbers [0-9], letters [a-z A-Z], '.', '_' and '-'.
 Leave blank for automatic key generation. Note that the prefix is an indication of the SUSE Manager organization the key is associated with.

Usage:
 Leave blank for unlimited use.

Base Channel:
 Choose "SUSE Manager Default" to allow systems to register to the default SUSE Manager provided channel that corresponds to the installed SUSE Linux version. Instead of the default, you may choose a particular SUSE provided channel or a custom base channel, but if a system using this key is not compatible with the selected channel, it will fall back to its SUSE Manager Default channel.

Add-On System Types: ☐ Container Build Host
☐ Virtualization Host

Contact Method:

Universal Default: ☐
 Tip: Only one universal default activation key may be set for this organization. By setting this key as universal default, you will remove universal default status from the current universal default key if it exists. If this key is set as universal default, then newly-registered systems to your organization will inherit the properties of this key.

[Create Activation Key](#)

1. In the Web UI, select **Main Menu** > **Systems** > **Activation Keys**.
2. Click **Create Key**.
3. Enter a **Description**, a **Key** name, and use the drop-down box to select a **Base Channel** to associate with the key.
4. Confirm with [**Create Activation Key**].

For more information, see [\[bp.key.managment\]](#).

Image Store

OS images can require a significant amount of storage space. Therefore, we recommended that the OS image store is located on a partition of its own or on a btrfs subvolume, separate from the root partition. By default, the image store will be located at `/srv/www/os-images`.



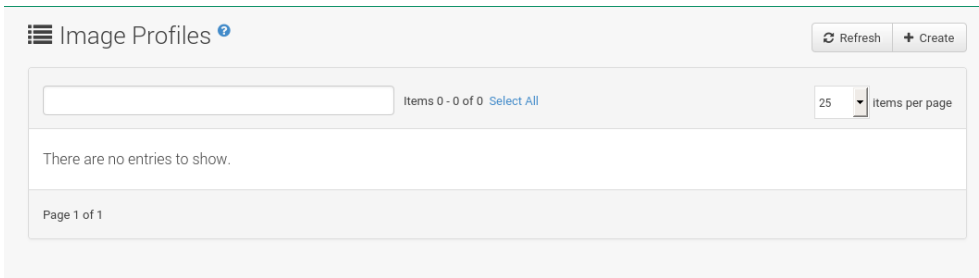
Image stores for Kiwi build type

Image stores for Kiwi build type, used to build system, virtual and other images, are not supported yet.

Images are always stored in `/srv/www/os-images/<organization id>` and are accessible via HTTP/HTTPS `https://<susemanager_host>/os-images/<organization id>`

Creating an Image Profile

Manage Image Profiles using the Web UI.



Procedure: Create an Image Profile

1. To create an image profile select from **Main Menu > Images > Images > Profiles** and click **[Create]**.

2. In the **Label** field, provide a name for the **Image Profile**.
3. Use **Kiwi** as the **Image Type**.
4. Image store is automatically selected.
5. Enter a **Config URL** to the directory containing the Kiwi configuration files:
 - a. Git URI
 - b. HTTPS tarball
 - c. Path to build host local directory
6. Select an **Activation Key**. Activation keys ensure that images using a profile are assigned to the correct channel and packages.



Relationship Between Activation Keys and Image Profiles

When you associate an activation key with an image profile you are ensuring any image using the profile will use the correct software channel and any packages in the channel.

7. Confirm with the **[Create]** button.

Source format options

- Git/HTTP(S) URL to the repository

URL to the Git repository containing the sources of the image to be built. Depending on the layout of the repository the URL can be:

```
https://github.com/SUSE/manager-build-profiles
```

You can specify a branch after the `#` character in the URL. In this example, we use the `master` branch:

```
https://github.com/SUSE/manager-build-profiles#master
```

You can specify a directory that contains the image sources after the `:` character. In this example, we use `OSImage/POS_Image-JeOS6`:

```
https://github.com/SUSE/manager-build-profiles#master:OSImage/POS_Image-JeOS6
```

- HTTP(S) URL to the tarball

URL to the tar archive, compressed or uncompressed, hosted on the webserver.

```
https://myimagesourceserver.example.org/MyKiwiImage.tar.gz
```

- Path to the directory on the build host

Enter the path to the directory with the Kiwi build system sources. This directory must be present on the selected build host.

```
/var/lib/Kiwi/MyKiwiImage
```

Example of Kiwi sources

Kiwi sources consist at least of `config.xml`. Usually `config.sh` and `images.sh` are present as well. Sources can also contain files to be installed in the final image under the `root` subdirectory.

For information about the Kiwi build system, see the [Kiwi documentation](#).

SUSE provides examples of fully functional image sources at the [SUSE/manager-build-profiles](#) public GitHub repository.

Listing 3. Example of JeOS config.xml

```

<?xml version="1.0" encoding="utf-8"?>

<image schemaversion="6.1" name="POS_Image_JeOS6">
  <description type="system">
    <author>Admin User</author>
    <contact>noemail@example.com</contact>
    <specification>SUSE Linux Enterprise 12 SP3 JeOS</specification>
  </description>
  <preferences>
    <version>6.0.0</version>
    <packagemanager>zypper</packagemanager>
    <bootplash-theme>SLE</bootplash-theme>
    <bootloader-theme>SLE</bootloader-theme>

    <locale>en_US</locale>
    <keytable>us.map.gz</keytable>
    <timezone>Europe/Berlin</timezone>
    <hwclock>utc</hwclock>

    <rpm-excludedocs>true</rpm-excludedocs>
    <type boot="saltboot/suse-SLES12" bootloader="grub2" checkprebuilt="true"
compressed="false" filesystem="ext3" fsmountoptions="acl" fsnocheck="true" image="pxe"
kernelcmdline="quiet"></type>
  </preferences>
  <!-- CUSTOM REPOSITORY
  <repository type="rpm-dir">
    <source path="this://repo"/>
  </repository>
  -->
  <packages type="image">
    <package name="patterns-sles-Minimal"/>
    <package name="aaa_base-extras"/> <!-- wouldn't be SUSE without that ;-) -->
    <package name="kernel-default"/>
    <package name="salt-minion"/>
    ...
  </packages>
  <packages type="bootstrap">
    ...
    <package name="sles-release"/>
    <!-- this certificate package is required to access {productname} repositories
    and is provided by {productname} automatically -->
    <package name="rhncert-trusted-ssl-cert-osimage" bootinclude="true"/>

  </packages>
  <packages type="delete">
    <package name="mtools"/>
    <package name="initvbiocons"/>
    ...
  </packages>
</image>

```

Building an Image

There are two ways to build an image using the Web UI. Either select **Main Menu** > **Images** > **Build**, or click the build icon in the **Main Menu** > **Images** > **Profiles** list.

Procedure: Build an Image

1. Select **Main Menu** > **Images** > **Build**.
2. Add a different tag name if you want a version other than the default **latest** (applies only to containers).
3. Select the **Image Profile** and a **Build Host**.



Profile Summary

A **Profile Summary** is displayed to the right of the build fields. When you have selected a build profile detailed information about the selected profile will show up in this area.

4. To schedule a build, click the [**Build**] button.

Image Inspection and Salt Integration

After the image is successfully built, the inspection phase begins. During the inspection phase SUSE Manager collects information about the image:

- List of packages installed in the image
- Checksum of the image
- Image type and other image details



If the built image type is **PXE**, a Salt pillar will also be generated. Image pillars are stored in the `/srv/susemanager/pillar_data/images/` directory and the Salt subsystem can access details about the generated image. Details include where the pillar is located and provided, image checksums, information needed for network boot, and more.

The generated pillar is available to all connected minions.

Troubleshooting

Building an image requires of several dependent steps. When the build fails, investigation of salt states results can help you to identify the source of the failure. Usual checks when the build fails:

- The build host can access the build sources
- There is enough disk space for the image on both the build host and the SUSE Manager server
- The activation key has the correct channels associated with it
- The build sources used are valid
- The RPM package with the SUSE Manager public certificate is up to date and available at [/usr/share/susemanager/salt/images/rhn-org-trusted-ssl-cert-osimage-1.0-1.noarch.rpm](#).

For more on how to refresh a public certificate RPM, see [Creating a Build Host](#).

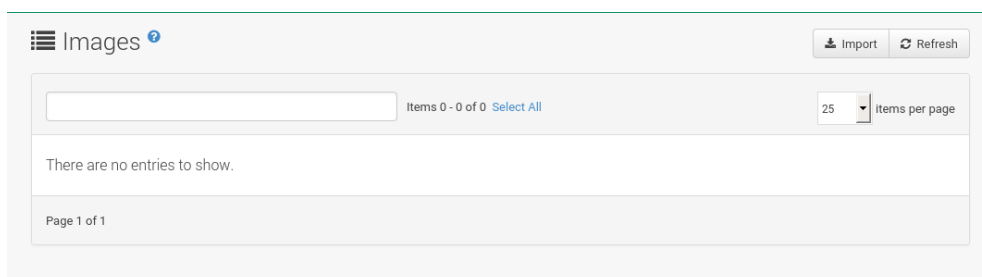
Limitations

The section contains some known issues when working with images.

- HTTPS certificates used to access the HTTP sources or Git repositories should be deployed to the minion by a custom state file, or configured manually.
- Importing Kiwi-based images is not supported.

Listing Image Profiles Available for Building

To list images available for building select **Main Menu** > **Images** > **Images**. A list of all images will be displayed.



Displayed data about images includes an image **Name**, its **Version** and the build **Status**. You will also see the image update status with a listing of possible patch and package updates that are available for the image.

Clicking the [**Details**] button on an image will provide a detailed view including an exact list of relevant patches and a list of all packages installed within the image.



The patch and the package list is only available if the inspect state after a build was successful.

Live Patching with SUSE Manager

Performing a kernel update usually requires a system reboot. Common vulnerability and exposure (CVE) patches should be applied as soon as possible, but if you cannot afford the downtime, you can use Live Patching to inject these important updates and skip the need to reboot.

The procedure for setting up Live Patching is slightly different for SLES 12 and SLES 15. Both procedures are documented in this section.

Live Patching on SLES 15

On SLES 15 systems and newer, live patching is managed by the `klp livepatch` tool.

Before you begin, ensure:

- SUSE Manager is fully updated
- You have one or more Salt clients running SLES 15 (SP1 or later)
- Your SLES 15 Salt clients are registered with SUSE Manager
- You have access to the SLES 15 channels appropriate for your architecture, including the Live Patching child channel (or channels)
- The clients are fully synchronized

Procedure: Setting up for Live Patching

1. Select the client you want to manage with Live Patching from **Systems > Overview**, and navigate to the **Software > Packages > Install** tab. Search for the `kernel-livepatch` package, and install it.
2. Apply the highstate to enable Live Patching, and reboot the client.
3. Repeat for each client that you want to manage with Live Patching.
4. To check that Live Patching has been enabled correctly, select the client from **Systems > Systems List**, and ensure that **Live Patching** appears in the **Kernel** field.

When you have the Live Patching channel installed on the client, you can clone the default vendor channel. This cloned channel will be used to manage Live Patching on your clients.

Cloned vendor channels should be prefixed by `dev` for development, `testing`, or `prod` for production. In this procedure, you will create a `dev` cloned channel, and later, you will need to promote the channel to `testing`.

Procedure: Cloning Live Patching Channels

1. At the command prompt on the client, as root, obtain the current package channel tree:

```
# spacewalk-manage-channel-lifecycle --list-channels
Spacewalk Username: admin
Spacewalk Password:
Channel tree:

1. sles15-sp{sp-ver}-pool-x86_64
   \__ sle-live-patching15-pool-x86_64-sp{sp-ver}
   \__ sle-live-patching15-updates-x86_64-sp{sp-ver}
   \__ sle-manager-tools15-pool-x86_64-sp{sp-ver}
   \__ sle-manager-tools15-updates-x86_64-sp{sp-ver}
   \__ sles15-sp{sp-ver}-updates-x86_64
```

2. Use the `spacewalk-manage-channel` command with the `init` argument to automatically create a new development clone of the original vendor channel:

```
spacewalk-manage-channel-lifecycle --init -c sles15-sp{sp-ver}-pool-x86_64
```

3. Check that `dev-sles15-spSP1-updates-x86_64` is available in your channel list.

Now you can check the `dev` cloned channel you created, and remove any kernel updates that require a reboot.

Procedure: Removing Non-Live Kernel Patches from Cloned Channels

1. Check the current kernel version by selecting the client from **Systems > Systems List**, and taking note of the version displayed in the **Kernel** field.
2. In the SUSE Manager Web UI, select the client from **Systems > Overview**, navigate to the **Software > Manage > Channels** tab, and select `dev-sles15-spSP1-updates-x86_64`. Navigate to the **Patches** tab, and click [**List/Remove Patches**].
3. In the search bar, type `kernel` and identify the kernel version that matches the kernel currently used by your client.
4. Remove all kernel versions that are newer than the currently installed kernel.

Your channel is now set up for Live Patching, and can be promoted to `testing`. In this procedure, you will also add the Live Patching child channels to your client, ready to be applied.

Procedure: Promoting Live Patching Channels

1. At the command prompt on the client, as root, promote and clone the `dev-sles15-spSP1-pool-x86_64` channel to a new testing channel:


```
# spacewalk-manage-channel-lifecycle -promote -c dev-sles15-sp{sp-ver}-pool-x86_64
```

2. In the SUSE Manager Web UI, select the client from **Systems > Overview**, and navigate to the **Software > Software Channels** tab.
3. Check the new `test-sles15-sp3-pool-x86_64` custom channel to change the base channel, and check both corresponding Live Patching child channels.

4. Click [**Next**], confirm that the details are correct, and click [**Confirm**] to save the changes.

You can now select and view available CVE patches, and apply these important kernel updates with Live Patching.

Procedure: Applying Live Patches to a Kernel

1. In the SUSE Manager Web UI, select the client from **Systems > Overview**. You will see a banner at the top of the screen showing the number of critical and non-critical packages available for the client: [scaledwidth=80%]
2. Click [**Critical**] to see a list of the available critical patches.
3. Select any patch with a synopsis reading **Important: Security update for the Linux kernel**. Security bugs will also include their CVE number, where applicable.
4. OPTIONAL: If you know the CVE number of a patch you want to apply, you can search for it in **Audit > CVE Audit**, and apply the patch to any clients that require it.



Not all kernel patches are Live Patches! Non-Live kernel patches are represented by a **Reboot Required** icon located next to the **Security** shield icon. These patches will always require a reboot.



Not all security issues can be fixed by applying a live patch. Some security issues can only be fixed by applying a full kernel update and will require a reboot. The assigned CVE numbers for these issues are not included in live patches. A CVE audit will display this requirement.

Live Patching on SLES 12

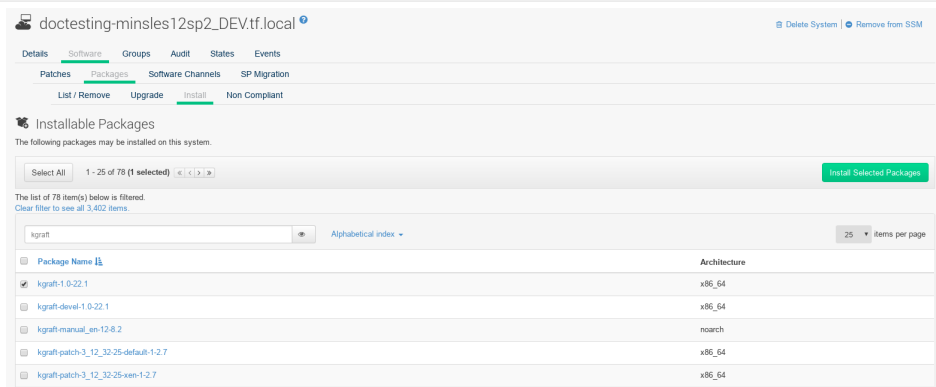
On SLES 12 systems, live patching is managed by kGraft. For in depth information covering kGraft use, see https://www.suse.com/documentation/sles-12/singlehtml/book_sle_admin/book_sle_admin.html#cha.kgraft.

Before you begin, ensure:

- SUSE Manager is fully updated
- You have one or more Salt clients running SLES 12 (SP1 or later)
- Your SLES 12 Salt clients are registered with SUSE Manager
- You have access to the SLES 12 channels appropriate for your architecture, including the Live Patching child channel (or channels)
- The clients are fully synchronized

Procedure: Setting up for Live Patching

1. Select the client you want to manage with Live Patching from **Systems > Overview**, and navigate to the **Software > Packages > Install** tab. Search for the **kgraft** package, and install it.



2. Apply the highstate to enable Live Patching, and reboot the client.
3. Repeat for each client that you want to manage with Live Patching.
4. To check that Live Patching has been enabled correctly, select the client from **Systems > Systems List**, and ensure that **Live Patching** appears in the **Kernel** field.

When you have the Live Patching channel installed on the client, you can clone the default vendor channel. This cloned channel will be used to manage Live Patching on your clients.

Cloned vendor channels should be prefixed by **dev** for development, **testing**, or **prod** for production. In this procedure, you will create a **dev** cloned channel, and later, you will need to promote the channel to **testing**.

Procedure: Cloning Live Patching Channels

1. At the command prompt on the client, as root, obtain the current package channel tree:

```
# spacewalk-manage-channel-lifecycle --list-channels
Spacewalk Username: admin
Spacewalk Password:
Channel tree:

1. sles12-sp{sp-ver}-pool-x86_64
   \__ sle-live-patching12-pool-x86_64-sp{sp-ver}
   \__ sle-live-patching12-updates-x86_64-sp{sp-ver}
   \__ sle-manager-tools12-pool-x86_64-sp{sp-ver}
   \__ sle-manager-tools12-updates-x86_64-sp{sp-ver}
   \__ sles12-sp{sp-ver}-updates-x86_64
```

2. Use the **spacewalk-manage-channel** command with the **init** argument to automatically create a new development clone of the original vendor channel:

```
spacewalk-manage-channel-lifecycle --init -c sles12-sp{sp-ver}-pool-x86_64
```

3. Check that **dev-sles12-spSP1-updates-x86_64** is available in your channel list.

Now you can check the **dev** cloned channel you created, and remove any kernel updates that require a reboot.

Procedure: Removing Non-Live Kernel Patches from Cloned Channels

1. Check the current kernel version by selecting the client from **Systems > Systems List**, and taking note of the version displayed in the **Kernel** field.
2. In the SUSE Manager Web UI, select the client from **Systems > Overview**, navigate to the **Software > Manage > Channels** tab, and select **dev-sles12-spSP1-updates-x86_64**. Navigate to the **Patches** tab, and click [**List/Remove Patches**].
3. In the search bar, type **kernel** and identify the kernel version that matches the kernel currently used by your client.
4. Remove all kernel versions that are newer than the currently installed kernel.

Your channel is now set up for Live Patching, and can be promoted to **testing**. In this procedure, you will also add the Live Patching child channels to your client, ready to be applied.

Procedure: Promoting Live Patching Channels


1. At the command prompt on the client, as root, promote and clone the **dev-sles12-spSP1-pool-x86_64** channel to a new testing channel:

```
# spacewalk-manage-channel-lifecycle -promote -c dev-sles12-sp{sp-ver}-pool-x86_64
```

2. In the SUSE Manager Web UI, select the client from **Systems > Overview**, and navigate to the **Software > Software Channels** tab.
3. Check the new **test-sles12-sp3-pool-x86_64** custom channel to change the base channel, and check both corresponding Live Patching child channels.
4. Click [**Next**], confirm that the details are correct, and click [**Confirm**] to save the changes.

You can now select and view available CVE patches, and apply these important kernel updates with Live Patching.

Procedure: Applying Live Patches to a Kernel

1. In the SUSE Manager Web UI, select the client from **Systems > Overview**. You will see a banner at the top of the screen showing the number of critical and non-critical packages available for the client: [scaledwidth=80%]
2. Click [**Critical**] to see a list of the available critical patches.
3. Select any patch with a synopsis reading **Important: Security update for the Linux kernel**. Security bugs will also include their CVE number, where applicable.
4. OPTIONAL: If you know the CVE number of a patch you want to apply, you can search for it in **Audit > CVE Audit**, and apply the patch to any clients that require it.



Not all kernel patches are Live Patches! Non-Live kernel patches are represented by a **Reboot Required** icon located next to the **Security** shield icon. These patches will always require a reboot.



Not all security issues can be fixed by applying a live patch. Some security issues can only be fixed by applying a full kernel update and will require a reboot. The assigned CVE numbers for these issues are not included in live patches. A CVE audit will display this requirement.

Monitoring with Prometheus

Monitoring can be performed in SUSE Manager using Prometheus and Grafana. The packages for Prometheus and Grafana are shipped with SUSE Manager Client Tools, as well as packages for several Prometheus exporters. SUSE Manager Server and Proxy are now able to provide self-health metrics, or install and manage a limited number of Prometheus exporters on managed client systems.

Prometheus is a monitoring tool, originally built at SoundCloud, that is used to record real-time metrics in a time-series database. Unlike other monitoring systems, Prometheus collects metrics using HTTP pulls, allowing for higher performance and scalability. Prometheus is an open-source software project, mostly written in Go, and its source code is available at <https://github.com/prometheus/>.

Grafana is a tool for data visualization, monitoring and analysis. It is used to create dashboards with panels representing specific metrics over a set period of time. Grafana is commonly used together with Prometheus, but also supports other data sources such as Elasticsearch, MySQL, PostgreSQL, and Influx DB. For more information about Grafana, see: <https://grafana.com/docs/>.

A Grafana package is included in the SUSE Manager Client Tools for SUSE Linux Enterprise 12 and SUSE Linux Enterprise 15.

Prometheus Metrics

Prometheus metrics are time series data, or timestamped values belonging to the same group or dimension. A metric is uniquely identified by its name and set of labels.

metric name	labels	timestamp	value
<pre>http_requests_total{status="200", method="GET"} @1557331801.111 42236</pre>			

Each application or system being monitored must expose metrics in the format above, either through code instrumentation, or Prometheus exporters.

The different metric types are:

- Counter - cumulative values. ex: number of errors
- Gauge - can go up or down. ex: temperature
- Histogram - count observations in buckets
- Summary - similar to histogram, but provides totals (sum and count)

For more information about metric types, see: https://prometheus.io/docs/concepts/metric_types/

PromQL

Prometheus has its own query language called PromQL, which is a functional expression language. PromQL allows you to filter multi-dimensional time series data. It is used in all Prometheus interactions.

In PromQL, an expression can evaluate to one of three types:

- Instant vector: a set of time series containing a single sample for each time series, all sharing the same timestamp
- Range vector: a set of time series containing a range of data points over time for each time series
- Scalar: a numeric floating point value

The core part of any PromQL query is the metric name, for example: **http_requests_total**. Labels can be used as optional selectors. This example returns the total number of HTTP requests that have status **200** and method **GET**:

```
http_requests_total{status="200", method="GET"}
```

For more information about PromQL, see the official Prometheus documentation: <https://prometheus.io/docs/prometheus/latest/querying/basics/>.

Exporters

Exporters are libraries which help in exporting existing metrics from third-party systems as Prometheus metrics. Exporters are useful whenever it is not feasible to instrument a given application or system with Prometheus metrics directly. Multiple exporters can run on a monitored host to export local metrics.

The Prometheus community provides a list of official exporters, and many others can be found as community contributions. For detailed information and an extensive list of exporters, see: <https://prometheus.io/docs/instrumenting/exporters/>.

With SUSE Manager 4, you can set up the Server and Proxy to expose Prometheus metrics to provide insights about SUSE Manager self-health. Metrics are available for these services:

- Hardware and Operating System
- Java Virtual Machines
- Apache
- Squid
- PostgreSQL
- SUSE Manager internals

The self-health metrics are made available by SUSE Manager Java application combined with Prometheus standalone exporters, running as systemd daemons.

SUSE Manager requires these packages to be installed on the Server and the Proxy. The packages are shipped with SUSE Manager Server and Proxy, but their respective systemd daemons are disabled by default.

These exporter packages are shipped with SUSE Manager Server:

- Node exporter: [golang-github-prometheus-node_exporter](https://github.com/prometheus/node_exporter). See https://github.com/prometheus/node_exporter.
- PostgreSQL exporter: [golang-github-wrouesnel-postgres_exporter](https://github.com/wrouesnel/postgres_exporter). See https://github.com/wrouesnel/postgres_exporter.
- JMX exporter: [prometheus-jmx_exporter](https://github.com/prometheus/jmx_exporter). See https://github.com/prometheus/jmx_exporter.
- Apache exporter: [golang-github-lusitaniae-apache_exporter](https://github.com/Lusitaniae/apache_exporter). See https://github.com/Lusitaniae/apache_exporter.

These exporter packages are shipped with SUSE Manager Proxy:

- Node exporter: [golang-github-prometheus-node_exporter](https://github.com/prometheus/node_exporter). See https://github.com/prometheus/node_exporter.
- Squid exporter: [golang-github-boynux-squid_exporter](https://github.com/boynux/squid_exporter). See https://github.com/boynux/squid_exporter.

Install and Configure Prometheus

Prometheus is installed from a package, and needs configuration before you can use it to gather metrics.

Installing Prometheus

Procedure: Installing Prometheus

1. Install the [golang-github-prometheus-prometheus](#) package:

```
zypper in golang-github-prometheus-prometheus
```

2. Enable the Prometheus service:

```
systemctl enable --now prometheus
```

3. Confirm that the Prometheus interface is loading correctly. In your browser, navigate to the URL of the server where Prometheus is installed, on port 9090 (for example, <http://example.com:9090>).

Configuring Prometheus

Prometheus requires some configuration to collect metrics and set up alarms, or to display metrics graphically in Grafana. You can configure Prometheus in the static configuration file at [/etc/prometheus/prometheus.yml](#). It is important to understand how this file is structured. For example:

```

yaml
- job_name: 'suse-manager-server'
  static_configs:
    - targets:
      - 'suse-manager.local:9100' # Node exporter
      - 'suse-manager.local:9187' # PostgreSQL exporter
      - 'suse-manager.local:5556' # JMX exporter (Tomcat)
      - 'suse-manager.local:5557' # JMX exporter (Taskomatic)
      - 'suse-manager.local:9800' # Taskomatic
    - targets:
      - 'suse-manager.local:80' # Message queue
  labels:
    __metrics_path__: /rhn/metrics

```

For more information about configuring Prometheus, see the official Prometheus documentation: <https://prometheus.io/docs/prometheus/latest/configuration/configuration/>

Monitoring Managed Systems

Prometheus metrics exporters can also be used on managed client systems. The packages are available from the SUSE Manager client tools channels, and can be enabled and configured directly on the SUSE Manager Web UI. Currently, two exporters are supported:

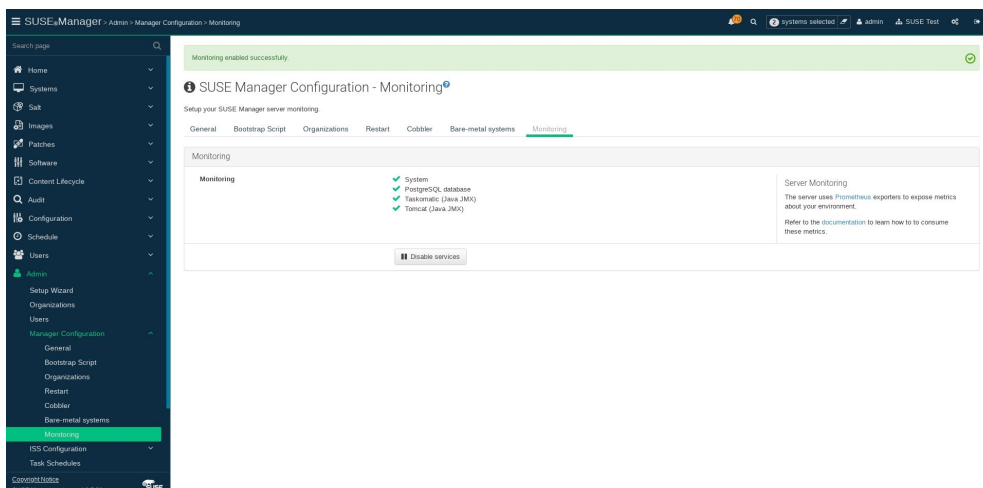
- Node exporter: [golang-github-prometheus-node_exporter](https://github.com/prometheus/node_exporter). See https://github.com/prometheus/node_exporter.
- PostgreSQL exporter: [golang-github-wrouesnel-postgres_exporter](https://github.com/wrouesnel/postgres_exporter). See https://github.com/wrouesnel/postgres_exporter.

Installing and configuring exporters is done using a Salt formula.

Enable and Configure Monitoring

Procedure: Enabling Self Monitoring for SUSE Manager

1. In the SUSE Manager Web UI, navigate to **Admin > Manager Configuration > Monitoring**.
2. Click [**Enable services**].

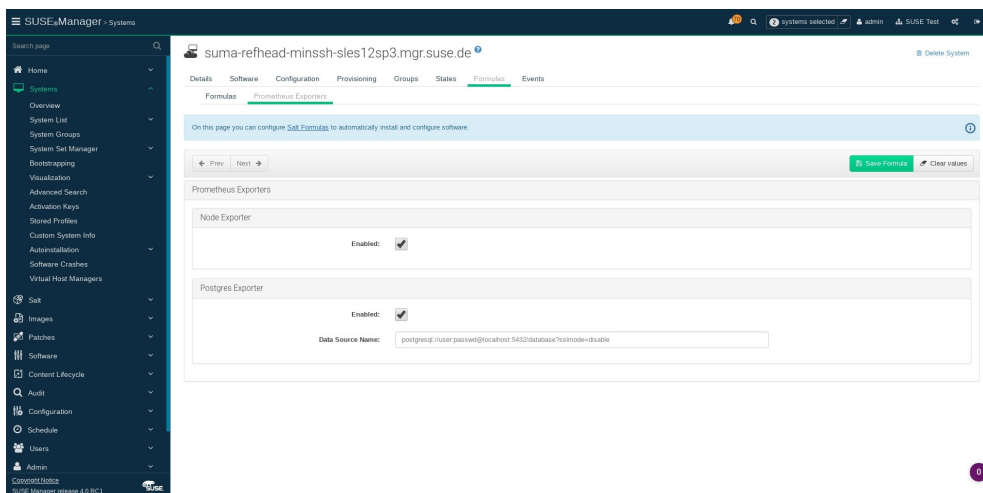


Procedure: Configuring Monitoring Formulas

1. In the SUSE Manager Web UI, open the details page for the server, and navigate to the Formulas tab.
2. Check the **Monitoring** checkbox to select all monitoring formulas, and click [**Save**].
3. Apply the highstate.

Procedure: Configuring the Exporters

1. In the SUSE Manager Web UI, open the details page for the server, and navigate to the **Formulas > Prometheus Exporters** tab.
2. Check the **Enabled** checkbox for both the Node Exporter, and the Postgres Exporter.
3. In the **Postgres Exporter** section, in the **Data Source Name** field, enter the path to your data source (for example `postgresql://user:passwd@localhost:5432/database?sslmode=disable`).
4. Click [**Save Formula**].
5. Apply the highstate.



Set up Visualization with Grafana

The Grafana website contains dozens of dashboards uploaded by the community. For an example SUSE Manager dashboard to help you to get started, see <https://grafana.com/dashboards/10277>. For more information, see: <https://grafana.com/dashboards>

To use Grafana with SUSE Manager, you need to have enabled metrics in the SUSE Manager Web UI, and configured your Prometheus instance to collect those metrics.

Procedure: Setting up Grafana

1. Install the **grafana** package:

```
zypper in grafana
```


2. Enable the Grafana service:

```
systemctl enable --now grafana-server
```

3. Navigate to port 3000 in your browser.



Grafana settings are configured in `/etc/grafana/grafana.ini`.

Kubernetes

Prerequisites

The prerequisites listed below should be met before proceeding.

- At least one *Kubernetes* or *_SUSE CaaS Platform _* cluster available on your network
- SUSE Manager configured for container management



Required channels are present, a registered build host available etc.

- virtual-host-gatherer-Kubernetes package installed on your SUSE Manager server

Requirements

- Kubernetes version 1.5.0 or higher. Alternatively use SUSE CaaS Platform (*SUSE CaaS Platform includes Kubernetes 1.5.0 by default*)
- Docker version 1.12 or higher on the container build host



To enable all Kubernetes related features within the Web UI, the virtual-host-gatherer-Kubernetes package must be installed.

Register Kubernetes as a Virtual Host Manager

Kubernetes clusters are registered with SUSE Manager as **virtual host managers**. Registration and authorization begins with importing a **kubeconfig** file using Kubernetes official command line tool **kubectl**.

Procedure: Registering a Kubernetes Cluster with SUSE Manager

1. Select **Systems** > **Virtual Host Managers** from the navigation menu.
2. Expand the **Create** dropdown in the upper right corner of the page and select **Kubernetes Cluster**.
3. Input a label for the new Virtual Host Manager.
4. Select the **kubeconfig** file which contains the required data for the Kubernetes cluster.
5. Select the correct *context* for the cluster, as specified in the kubeconfig file.
6. Click **Create**.

View the List of Nodes in a Cluster

1. Select **Systems** > **Virtual Host Managers** from the navigation menu.
2. Select the desired Kubernetes cluster to view it.

3. Node data is not refreshed during registration. To refresh node data, click on [Schedule refresh data](#).
4. Refresh the browser. If the node data is not available wait a few moments and try again.

Obtain Runtime Data about Images

See the following steps to find runtime data for images.

1. Select **Images** > **Images** from the navigation menu.
2. In the image list table, take notice of the new runtime columns. These are labeled: [Revision](#), [Runtime](#) and [Instances](#). Initially these columns will not provide useful data.
 - [Revision](#): An artificial sequence number which increments on every rebuild for manager-built images, or on every reimport for externally built images.
 - [Runtime](#): Overall status of the running instances of the image throughout the registered clusters. The status can be one of the following:
 - All instances are consistent with SUSE Manager: All the running instances are running the same build of the image as tracked by SUSE Manager.
 - Outdated instances found: Some of the instances are running an older build of the image. A redeploy of the image into the pod may be required.
 - No information: The checksum of the instance image does not match the image data contained in SUSE Manager. A redeploy of the image into the pod may be required.
 - [Instances](#): Number of instances running this image across all the clusters registered in SUSE Manager. A breakdown of numbers can be seen by clicking on the pop-up icon next to the number.

Build an image for deployment in Kubernetes

The following steps will help you build an image for deployment in Kubernetes.

1. Under **Images** > **Stores**, create an image store.
2. Under **Images** > **Profiles**, create an image profile (with a Dockerfile which is suitable to deploy to Kubernetes).
3. Under **Images** > **Build**, build an image with the new profile and wait for the build to finish.
4. Deploy the image into one of the registered Kubernetes clusters (via [kubect1](#)).
5. Notice the updated data in [Runtime](#) and [Instances](#) columns in the respective image row.

Import a Previously Deployed Image in Kubernetes

The following steps will guide you through importing a previously deployed image in Kubernetes.

1. Select an image that has already been deployed to any of your registered Kubernetes clusters.
2. Add the registry owning the image to SUSE Manager as an image store.
3. Select **Images** > **Images** , click **Import** from the top-right corner, fill in the form fields and click **Import**.
4. Notice the updated data in **Runtime** and **Instances** columns in the respective image row.

Obtain Additional Runtime Data

The following steps will help you find additional runtime data.

1. Select to **Images** > **Images** , click the **Details** button on the right end of a row which has running instances.
2. Under the **Overview** tab, notice the data in **Runtime** and **Instances** fields under **Image Info** section.
3. Select the **Runtime** tab.
4. Here is a breakdown of the Kubernetes pods running this image in all the registered clusters including the following data:
 - Pod name
 - Namespace which the pod resides in
 - The runtime status of the container in the specific pod. Status icons are explained in the preceeding example.

Rebuild a Previously Deployed Image in Kubernetes

The following steps will guide you through rebuilding an image which has been deployed to a Kubernetes cluster.

1. Go to **Images** > **Images** , click the **Details** button on the right end of a row which has running instances. The image must be manager-built.
2. Click the **Rebuild** button located under the **Build Status** section and wait for the build to finish.
3. Notice the change in the **Runtime** icon and title, reflecting the fact that now the instances are running a previous build of the image.

Role Based Access Control Permissions and Certificate Data



Currently, only kubeconfig files containing all embedded certificate data may be used with SUSE Manager

The API calls from SUSE Manager are:

- GET /api/v1/pods
- GET /api/v1/nodes

According to this list, the minimum recommended permissions for SUSE Manager should be as follows:

- A ClusterRole to list all the nodes:

```
resources: ["nodes"]
verbs: ["list"]
```

- A ClusterRole to list pods in all namespaces (role binding must not restrict the namespace):

```
resources: ["pods"]
verbs: ["list"]
```

Due to a 403 response from /pods, the entire cluster will be ignored by SUSE Manager.

For more information on working with RBAC Authorization see: <https://kubernetes.io/docs/admin/authorization/rbac/>

Public Cloud

Some public cloud environments provide images for SUSE Manager Server and Proxy. This section discusses what you will require to run SUSE Manager in a public cloud, and how to set up your installation.



Public clouds provide SUSE Manager under a Bring Your Own Subscription (BYOS) model. This means that you must register them with the SUSE Customer Center. For more information about registering SUSE Manager with SUSE Customer Center, see [installation:general-requirements.pdf](#).

Depending on the public cloud network you are using, you can locate the SUSE Manager installation images by searching for the keywords [suse](#), [manager](#), [proxy](#), or [BYOS](#).

Instance Requirements

Select a public cloud instance that meets the hardware requirements in [installation:hardware-requirements.pdf](#).

In addition, be aware of these important considerations:

- The SUSE Manager setup procedure performs a forward-confirmed reverse DNS lookup. This must succeed in order for the setup procedure to complete successfully and for SUSE Manager to operate as expected. Therefore, it is important that the hostname and IP configuration be performed prior to running the SUSE Manager setup procedure.
- SUSE Manager Server and Proxy instances are expected to run in a network configuration that provides you control over DNS entries, but cannot access the wider internet. Within this network configuration DNS resolution must be provided, such that `hostname -f` returns the FQDN. DNS resolution is also important for connecting clients. DNS is dependent on the cloud framework you choose, refer to the cloud service provider documentation for detailed instructions.
- We recommend that you locate software repositories, the server database, and the Proxy squid cache on an external virtual disk. This prevents data loss if the instance is unexpectedly terminated. Instructions for setting up an external virtual disk are contained in this section.

Network Setup

On a public cloud service, you must run SUSE Manager within a restricted network, such as VPC private subnet with an appropriate firewall setting. The instance must only be able to be accessed by machines in your specified IP ranges.



A world-accessible SUSE Manager instance violates the terms of the SUSE Manager EULA, and it will not be supported by SUSE.

When you are setting up your networking environment, you will need to ensure that you allow https, in order to be able access the SUSE Manager Web UI.

Set the hostname

SUSE Manager requires a stable and reliable hostname. Changing the hostname at a later point can create errors.

In most public cloud environments, the method shown in this section will work correctly. However, you will have to perform the same modification for every client.

You might prefer to manage DNS resolution by creating a DNS entry in your network environment instead.

You can also manage hostname resolution by editing the `/etc/resolv.conf` file. Depending on the order of your setup, if you start the SUSE Manager instance prior to setting up DNS services the file may not contain the appropriate `search` directive. Check that the proper search directive exists in `/etc/resolv.conf` and add it if it is missing.

Procedure: Setting the host name locally

1. Disable hostname setup by editing the DHCP configuration file at `/etc/sysconfig/network/dhcp`, and adding this line:

```
DHCLIENT_SET_HOSTNAME="no"
```

2. Set the hostname locally with the `hostnamectl` command. Ensure you use the system name, not the FQDN. For example, if the FQDN is `system_name.example.com`, the system name is `system_name`, and the domain name is `example.com`.

```
hostnamectl set-hostname system_name
```

3. Create a DNS entry in your network environment for domain name resolution, or force correct resolution by editing the `/etc/hosts` file:

```
$ echo "${local_address} suma.cloud.net suma" >> /etc/hosts
```

You can find the local address by checking your public cloud web console, or from the command line :

- Amazon EC2 instance:

```
$ ec2metadata --local-ipv4
```

- Google Compute Engine:

```
$ gcloud metadata --query instance --network-interfaces --ip
```

- Microsoft Azure:

```
$ azuremetadata --internal-ip
```

Set up DNS resolution

You will need to update the DNS records for the instance within the DNS service of your network environment. Refer to the cloud service provider documentation for detailed instructions: * [DNS setup on Amazon EC2](#) * [DNS setup on Google Compute Engine](#) * [DNS setup on Microsoft Azure](#)

If you run a SUSE Manager Server instance, you can run YaST after the instance is launched to ensure the external storage is attached and prepared correctly, and that DNS resolution is set up as described:

```
$ /sbin/yast2 susemanager_setup
```

PUT THIS COMMENT AT THE TOP OF TROUBLESHOOTING SECTIONS

Troubleshooting format:

One sentence each: Cause: What created the problem? Consequence: What does the user see when this happens? Fix: What can the user do to fix this problem? Result: What happens after the user has completed the fix?

If more detailed instructions are required, put them in a "Resolving" procedure: .Procedure: Resolving Widget Wobbles . First step . Another step . Last step

PUT THIS COMMENT AT THE TOP OF TROUBLESHOOTING SECTIONS

Troubleshooting format:

One sentence each: Cause: What created the problem? Consequence: What does the user see when this happens? Fix: What can the user do to fix this problem? Result: What happens after the user has completed the fix?

If more detailed instructions are required, put them in a "Resolving" procedure: .Procedure: Resolving Widget Wobbles . First step . Another step . Last step

PUT THIS COMMENT AT THE TOP OF TROUBLESHOOTING SECTIONS

Troubleshooting format:

One sentence each: Cause: What created the problem? Consequence: What does the user see when this happens? Fix: What can the user do to fix this problem? Result: What happens after the user has completed the fix?

If more detailed instructions are required, put them in a "Resolving" procedure: .Procedure: Resolving

Widget Wobbles . First step . Another step . Last step

PUT THIS COMMENT AT THE TOP OF TROUBLESHOOTING SECTIONS

Troubleshooting format:

One sentence each: Cause: What created the problem? Consequence: What does the user see when this happens? Fix: What can the user do to fix this problem? Result: What happens after the user has completed the fix?

If more detailed instructions are required, put them in a "Resolving" procedure: .Procedure: Resolving Widget Wobbles . First step . Another step . Last step

PUT THIS COMMENT AT THE TOP OF TROUBLESHOOTING SECTIONS

Troubleshooting format:

One sentence each: Cause: What created the problem? Consequence: What does the user see when this happens? Fix: What can the user do to fix this problem? Result: What happens after the user has completed the fix?

If more detailed instructions are required, put them in a "Resolving" procedure: .Procedure: Resolving Widget Wobbles . First step . Another step . Last step

PUT THIS COMMENT AT THE TOP OF TROUBLESHOOTING SECTIONS

Troubleshooting format:

One sentence each: Cause: What created the problem? Consequence: What does the user see when this happens? Fix: What can the user do to fix this problem? Result: What happens after the user has completed the fix?

If more detailed instructions are required, put them in a "Resolving" procedure: .Procedure: Resolving Widget Wobbles . First step . Another step . Last step