Getting Started

Getting Started

Publication Date: 2018-10-25

SUSE LLC
10 Canal Park Drive
Suite 200
Cambridge MA 02141
USA
https://www.suse.com/documentation

Contents

1 Introduction 1

1.1 Introduction to Uyuni 1

1.2 Prerequisites for Installation 2

Obtaining your SUSE Customer Center Credentials 2 • Obtaining Installation Media 2 • Hardware Requirements 3 • Network Requirements 4 • Supported Client Systems 6

2 JeOS Installation 8

2.1 Virtual Machine Manager (virt-manager) Settings 8

2.2 JeOS KVM Settings 9

2.3 Preparing JeOS for Uyuni 10

3 SUSE Linux Enterprise Server Installation 12

3.1 SLES KVM Requirements 12

SLES KVM Settings 13

3.2 Selecting the Uyuni Extension 14

4 SUSE Manager Setup 15

4.1 Topics 15

4.2 Uyuni Setup 15

Creating the Main Administration Account 18 • Syncing Products from SUSE Customer Center 20

5 Registering Clients 23

5.1 Introduction 23

5.2 Creating Activation Keys 23

5.3 Creating the Uyuni Tools Repository 26

5.4 Registering Traditional Clients 28
Generating a Bootstrap Script 28 • Editing the Bootstrap Script 30 • Connecting Clients 31 • Package Locks 31
5.5 Registering Salt Clients 32
6 Getting Started with Salt 35
6.1 Introduction 35
6.2 Understanding Salt Calls 35
6.3 Common Salt Terminology 37
6.4 Useful Salt Commands 39
6.5 Salt File Locations and Structure 40
file_roots 41 • pillar_roots 41
6.6 Install the SUSE Manager Locale Formula 42

1 Introduction

This Getting Started Guide provides guidance on setting up Uyuni server with KVM. You will learn the fundamentals of managing both traditional and Salt clients.

This guide is intended for system administrators.

Note: Uyuni Version Information

In this manual if not other specified, Uyuni version 3.2 is assumed and this version is required if a feature is discussed. Uyuni 3.2 and Uyuni 3.2 Proxy were originally released as a SLES 12 SP3 extension. Whenever features of the Uyuni3.2 host operating system are documented and not other specified version 12 SP3 is assumed.

1.1 Introduction to Uyuni

Uyuni is a solution for organizations that require absolute control over maintenance and package deployment on their servers. It lets you manage large sets of Linux systems and keep them up to date, with automated software management, asset management, and system provisioning. Uyuni allows you to maintain a high level of security while effectively managing system life-cycle requirements.

Uyuni uses *Salt* to provide event-driven configuration and management control. The Salt-master orchestrates thousands of Salt-minions (Uyuni Clients) using remote execution.

Uyuni is fully compatible with Red Hat Satellite Server and offers seamless management of both SUSE Linux Enterprise and Red Hat Enterprise Linux client systems.

Uyuni can be integrated with your network infrastructure in multiple ways. This book will guide you through an initial proof-of-concept setup, using these steps:

1. Install an operating system (either JeOS or SLES) for use with Uyuni
2. Install Uyuni Server
3. Register Uyuni with SUSE Customer Center
4. Perform initial setup of your Uyuni Server
5. Register a traditional client
6. Register a Salt minion

1.2 Prerequisites for Installation

Before you begin your installation, ensure you have fulfilled these prerequisites:

- Current SUSE Customer Center organization credentials
- Access to installation media for your chosen operating system
- Your environment meets the hardware and networking requirements
- You understand the supported client operating systems

This section contains more information on each of these prerequisites.

Note

Uyuni 3.2 is based on SLES 12 SP3 as the host operating system.

1.2.1 Obtaining your SUSE Customer Center Credentials

You will need to create an account with SUSE Customer Center before you install SUSE Linux Enterprise Server and Uyuni. To obtain your SUSE Customer Center credentials:

PROCEDURE: OBTAINING YOUR SCC ORGANIZATION CREDENTIALS

1. Open a browser and direct it to https://scc.suse.com/login.
2. If you have not done so, create an account now.
3. Log in to your new SCC account.
4. Under the *Management tools* widget select *Manage Users*.
5. Click the *Organization Credentials* tab.
6. Record your login information for use during Uyuni setup.

1.2.2 Obtaining Installation Media

This book describes installation methods for both JeOS and SUSE Linux Enterprise Server. The JeOS image provides the quickest installation and setup, and is suitable for a test or proof of concept installation. Alternatively, SUSE Linux Enterprise Server provides a more robust instal-

lation, which requires a larger initial download. Choose your preferred operating system based on the type of environment you want to install, and the amount of bandwidth and time you have available.

You can find installation images for JeOS and SLES in your SUSE Customer Center account. Log in, then navigate to the URL for your chosen operating system:

- JeOS - Media Download (https://www.suse.com/products/server/jeos/)
- SUSE Linux Enterprise Server - Media Download (https://www.suse.com/products/server/download/)

1.2.3 Hardware Requirements

This table outlines hardware and software requirements on x86_64 and IBM Power PC architecture. For installation on z Systems, see:

- *Book "Advanced Topics", Chapter 1 "SUSE Manager on IBM z Systems"*

TABLE 1.1: HARDWARE REQUIREMENTS FOR X86_64 ARCHITECTURE

Hardware	Recommended
CPU	Multi-core 64-bit CPU
RAM:	*Test Server* Minimum 8 GB
	Base Installation Minimum 16 GB
	Production Server Minimum 32 GB
Disk Space:	`/` *(root) The default JeOS root partition size of 24 GB is sufficient for this guide*
	`/var/lib/pgsql` Minimum 50 GB
	`/var/spacewalk` Minimum 50 GB per SUSE product and 250 GB per Red Hat product

TABLE 1.2: HARDWARE REQUIREMENTS FOR IBM POWER8 OR POWER9 ARCHITECTURE

Hardware	Recommended
CPU	Minimum 4 dedicated cores
RAM:	*Test Server* Minimum 8 GB
	Base Installation Minimum 16 GB
	Production Server Minimum 32 GB
Disk Space:	`/` Minimum 100 GB
	`/var/lib/pgsql` Minimum 50 GB
	`/var/spacewalk` Minimum 50 GB per SUSE product and 250 GB per Red Hat product

1.2.4 Network Requirements

This section details the networking and port requirements for Uyuni.

Fully Qualified Domain Name (FQDN)

The Uyuni server must resolve its FQDN correctly or cookies will not work properly on the WebUI. For more information about configuring the hostname and DNS, see SUSE Linux Enterprise Server Documentation - Configuring Host Name and DNS (https://www.suse.com/documentation/sles-12/book_sle_admin/data/sec_basicnet_yast.html#sec_basicnet_yast_change_host)

Hostname and IP Address

To ensure that the Uyuni domain name can be resolved by its clients, both server and client machines must be connected to a working DNS server. For more information about setting up a DNS server, see SUSE Linux Enterprise Server Documentation - The Domain Name System (https://www.suse.com/documentation/sles-12/book_sle_admin/data/cha_dns.html)

Using a Proxy When Installing from SUSE Linux Enterprise Media

If you are on an internal network and do not have access to SUSE Customer Center, you can set up and use a proxy during installation. For more information about configuring a proxy for access to SUSE Customer Center during a SUSE Linux Enter-

prise installation, see SUSE Linux Enterprise Server Documentation - Using a Proxy During Installation (https://www.suse.com/documentation/sled-12/singlehtml/book_sle_deployment/book_sle_deployment.html#sec.i.yast2.start.parameters.proxy)

Important: Naming Your Server

The hostname of Uyuni must not contain uppercase letters as this may cause *jabberd* to fail. Choose the hostname of your Uyuni server carefully. Although changing the server name is possible, it is a complex process and unsupported.

In a production environment, Uyuni server and its clients should always use a firewall. This table gives an overview of required ports, to be used when you are setting up your firewall rules.

TABLE 1.3: REQUIRED SERVER PORTS

Port	Protocol	Description
22	TCP	SSH
67	UDP	DHCP
69	UDP	TFTP, used to support PXE services
80	TCP	HTTP, used in some bootstrap cases
123	UDP	NTP time service
443	TCP	HTTPS, used for Web UI, client, Proxy server, and API traffic
4505	TCP	Salt, used by the Salt-master to accept communication requests from minions
4506	TCP	Salt, used by the Salt-master to accept communication requests from minions

Port	Protocol	Description
5222	TCP	XMPP client, used for communications with the `osad` daemon on traditional client systems
5269	TCP	XMPP server, used for pushing actions to SUSE Manager Proxy

For more information on disconnected setup and port configuration, see:

-
-

1.2.5 Supported Client Systems

Supported operating systems for traditional and Salt clients are listed in this table.

TABLE 1.4: SUPPORTED CLIENT SYSTEMS

Operating Systems	Architecture	Traditional Clients	Salt Clients
SUSE Linux Enterprise 11 SP4	x86, x86_64, Itanium, IBM POWER, z Systems	Supported	Supported
SUSE Linux Enterprise 12 SP3	x86_64, IBM POWER (IBM Power PC), z Systems, ARM	Supported	Supported
SUSE Linux Enterprise 15	x86_64, IBM POWER (IBM Power PC), z Systems, ARM	Supported	Supported
Latest minor release Red Hat Enterprise Linux Server 6	x86, x86_64	Supported	Supported

Operating Systems	Architecture	Traditional Clients	Salt Clients
Latest minor release Red Hat Enterprise Linux Server 7	x86_64	Supported	Supported
Open Enterprise Server 2015, 2015 SP1, 2018	x86_64	Supported	Supported

Note: Supported Versions and SP Levels

Client operating system versions and SP levels must be under general support (normal or LTSS) to be supported with Uyuni. For details on supported product versions, see https://www.suse.com/lifecycle.

2 JeOS Installation

2.1 Virtual Machine Manager (virt-manager) Settings

This chapter provides the required (KVM) settings for installation of SUSE Linux Enterprise Just Enough Operating System (JeOS) 12 as the base for Uyuni. A kernel virtual machine (KVM) combined with Virtual Machine Manager (*virt-manager*) will be used as a sandbox for your first installation.

Tip: SUSEVirtualization Guide

For more information on virtualization, see: SUSE Linux Enterprise Virtualization Guide (https://www.suse.com/documentation/sles-12/singlehtml/book_virt/book_virt.html)

Enter the following settings when creating a new virtual machine using **virt-manager**. In the following table replace *version* with the actual product version string.

KVM Settings	
Installation Method	Import Existing Disk Image
OS:	Linux
Version:	SLES<VERSION>-JeOS-for-kvm-and-xen.x86_64-GM.qcow2
Memory:	4096 MB
CPU's:	2
Storage Format:	*.qcow2* 24 GB (Default) JeOS Root Partition
Virtual Disks:	
VirtIO Disk 2	101 GB for `/var/spacewalk`
VirtIO Disk 3	50 GB for `/var/lib/pgsql`

KVM Settings	
VirtIO Disk 4	4 GB for swap
Name:	test-setup
Network	Bridge *br0*

Tip: SUSE Virtualization Guide

For more information on virtualization, see: SUSE Linux Enterprise Virtualization Guide (https://www.suse.com/documentation/sles-12/singlehtml/book_virt/book_virt.html)

2.2 JeOS KVM Settings

Create three additional virtual disks required for the Uyuni storage partitions.

PROCEDURE: CREATING THE REQUIRED PARTITIONS WITH KVM

1. Create a new virtual machine using the downloaded JeOS KVM image and select *Import existing disk image* .
2. Configure RAM and number of CPUs (At least 4 GB RAM and 2 CPUs).
3. Name your KVM machine and select the *Customize configuration before install* check box.
4. Select the *Add Hardware* button and create three new virtual disks with the following specifications. These disks will be partitioned and mounted in *Procedure: Preparing JeOS for Uyuni Installation*.

VirtIO Storage Disks	Name	Sizing
VirtIO Disk 2	spacewalk	101 GB
VirtIO Disk 3	pgsql	50 GB
VirtIO Disk 4	swap	4 GB

5. Click *Begin Installation* and your new VM will boot from the JeOS image.

Proceed through the basic JeOS installation prompts until you reach the command line.

Tip: Root Password

During the basic installation prompts you are asked to enter the root password. Select a strong password and then in the next message box *Confirm root Password*.

2.3 Preparing JeOS for Uyuni

PROCEDURE: PREPARING JEOS FOR UYUNI INSTALLATION

1. Register with SCC:

```
SUSEConnect -e`EMAIL_ADDRESS`-r`SUSE_MANAGER_CODE`
```

2. Add Uyuni repositories:

```
SUSEConnect -p SUSE-Manager-Server/<productnumber>/x86_64 -r`SUSE_MANAGER_CODE`
```

3. Install yast2-storage with all required dependencies (approx. 40 packages, 30 MB when installed). This basic administration package is required for preparing storage partitions:

```
zypper in -t package yast2-storage
```

4. Partition and mount the virtual disks at the following locations using YaST Partitioner (`yast2 disk`).

VirtIO Storage Disks	Name	Storage Size	File System Type
VirtIO Disk 2	`/var/spacewalk`	101 GB	XFS
VirtIO Disk 3	`/var/lib/pgsql`	50 GB	XFS
VirtIO Disk 4	`swap`	4 GB	swap

5. SLES by default uses the BTRFS file system. A mount point is created automatically for `/var/lib/pgsql/` (even when not installed). This must be removed or commented out from the `/etc/fstab` entries. As root, edit `/etc/fstab` and comment out or remove the line:

```
/var/lib/pgsql btrfs subvol=@/var/lib/pgsql 0 0
```

Warning: Remove `pgsql` from the fstab Configuration File

If you do not remove this line from fstab the first time you shut down the server you will lose your database. This occurs because you will have duplicate entries in the fstab file. *Updated tools shipped with recent SPs will no longer require human intervention.*

6. Exit the partitioner and install the Uyuni pattern:

```
zypper in -t pattern suma_server
```

For proceeding with Uyuni setup, see SUSE Manager Setup (quickstart3_chap_sumaset-up_with_yast.xml#suma.setup.with.yast).

3 SUSE Linux Enterprise Server Installation

This chapter provides the required KVM settings for installation of SUSE Linux Enterprise Server media as the base for Uyuni. A kernel virtual machine KVM combined with Virtual Machine Manager (`virt-manager`) will be used as a sandbox for this installation.

3.1 SLES KVM Requirements

Enter the following settings when creating a new virtual machine using `virt-manager` (replace `version` with the actual version string):

KVM Settings for SLES	**Installation Method:**
Local install media (ISO image or CDROM)	OS:
Linux	Version:
`SLE-[replaceable]` version `-Server-x86_64-GM-DVD1.iso`	Memory:
4096 MB	CPUs:
2	Storage Format:
ISO 3 GB	Disk Space:
234 GB split between 4 GB swap and 130 GB mounted at `/var/spacewalk/`	
(Virtual Disk 1) and 50 GB mounted at `/var/lib/pgsql`	
(Virtual Disk 2). The rest for the root partition (100 GB+).	Name:
example-server	Network

3.1.1 SLES KVM Settings

This section provides guidance on installation of Uyuni utilizing the full installation media with KVM and `virt-manager`. This section assumes you have previously setup an account with SCC and downloaded the SLES full installation media.

PROCEDURE: PREPARING FOR SLES INSTALLATION

1. In `virt-manager` select *File* › *New Virtual Machine*.
2. Select *Local install media (ISO image or CDROM)*.
3. Ensure *Use ISO Image* is selected then click *Browse* and locate the full SLES image you downloaded from your SCC account.
4. Configure your machine with at least 4096 MB RAM and a minimum of 2 CPUs.
5. Create a storage device with a minimum of 234 GB storage space for the installation. During the partitioning setup of the SLES installation this disk should be partitioned into the following disks:

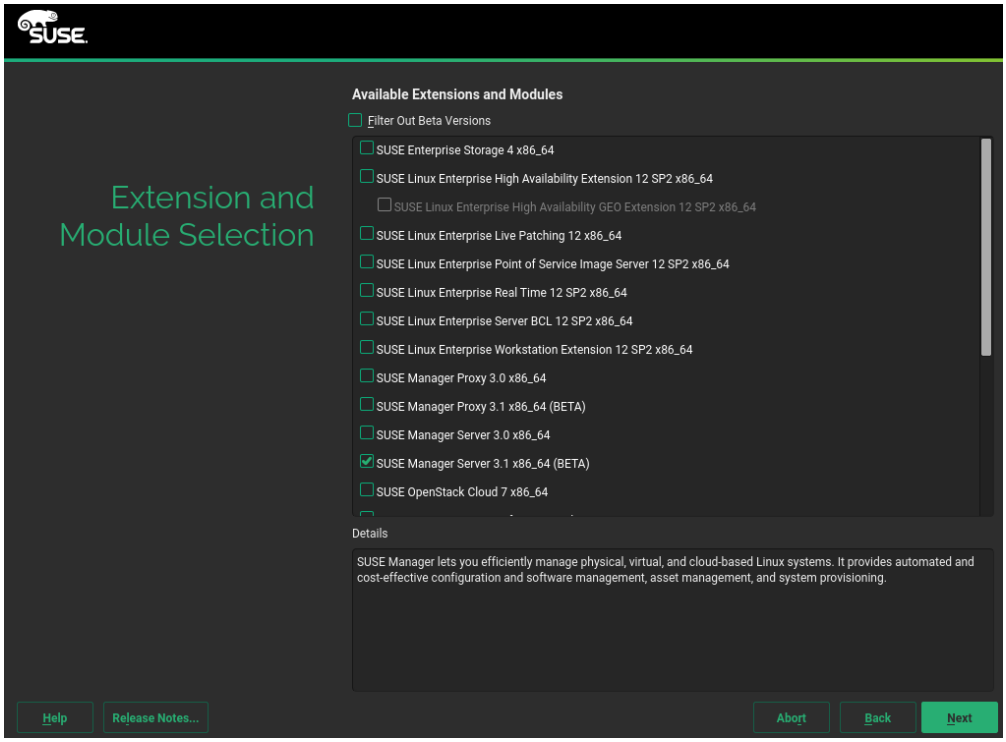
Disk Space Requirements
4 GB Swap space
130 GB XFS partition (or dedicated virtual disk) for `/var/spacewalk/`
50 GB XFS partition (or dedicated virtual disk) for `/var/lib/pgsql/`

6. The remaining storage space will be used by the operating system for the root partition. Select *Finish* to begin the installation.

Installation of SUSE Linux Enterprise Server will begin. For more information on completing an installation of SUSE Linux Enterprise Server, see: SUSE Linux Enterprise Installation Quickstart (https://www.suse.com/documentation/sles-12/book_quickstarts/data/sec_sle_installquick.html).

3.2 Selecting the Uyuni Extension

1. During the SUSE Linux Enterprise Server installation you will be presented with the *Extension and Module Selection*.
2. Select the Uyuni Extension and then click the *Next* button.
3. Complete the SUSE Linux Enterprise Server installation.



4 SUSE Manager Setup

4.1 Topics

This section covers Uyuni setup. You will perform the following procedures:

- Start Uyuni setup via YaST or command line
- Create the main administration account with the Uyuni Web UI
- Name your base organization and add login credentials
- Sync the SUSE Linux Enterprise product channel from SUSE Customer Center

4.2 Uyuni Setup

Warning: Third Party Software

Uyuni is an extension of SUSE Linux Enterprise Server and compatible with the software shipped with SUSE Linux Enterprise Server.

Uyuni is a complex system, and therefore installing third party is not allowed. Installing monitoring software provided by a third party vendor is allowed only if you do not exchange basic libraries such as SSL, cryptographic software, and similar tools. In case of emergency, SUSE reserves the right to ask to remove any third party software (and associated configuration changes) and then to reproduce the problem on a clean system.

This section will guide you through Uyuni setup procedures.

PROCEDURE: UYUNI SETUP

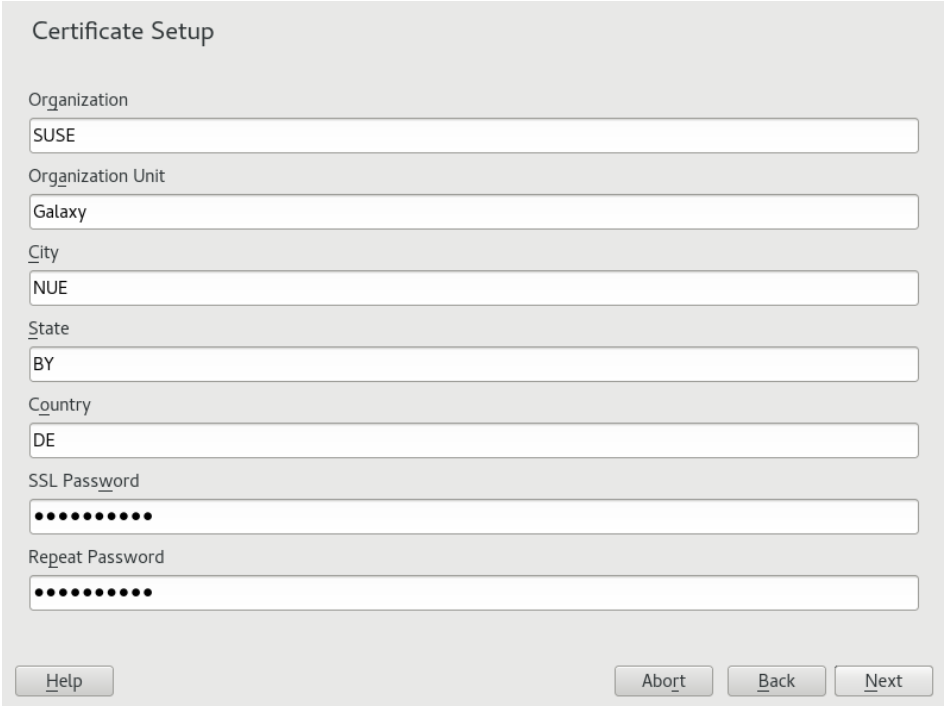
1. Login to the Uyuni server desktop and perform one of the following actions to begin setup:
   - Select *Applications* › *System Tools* › *YaST* › *SUSE Manager Setup*.
   - Open a terminal as root and type `yast2 susemanager_setup` to begin setup.
2. From the introduction screen select *SUSE Manager Setup* › *Setup SUSE Manager from scratch*. Then click *Next* to continue.

3. Enter the email address that should receive status notifications about Uyuni. The number of emails sent from Uyuni can be extensive, therefore notifications via email may be disabled from the Web UI after setup. Then click *Next* to continue.

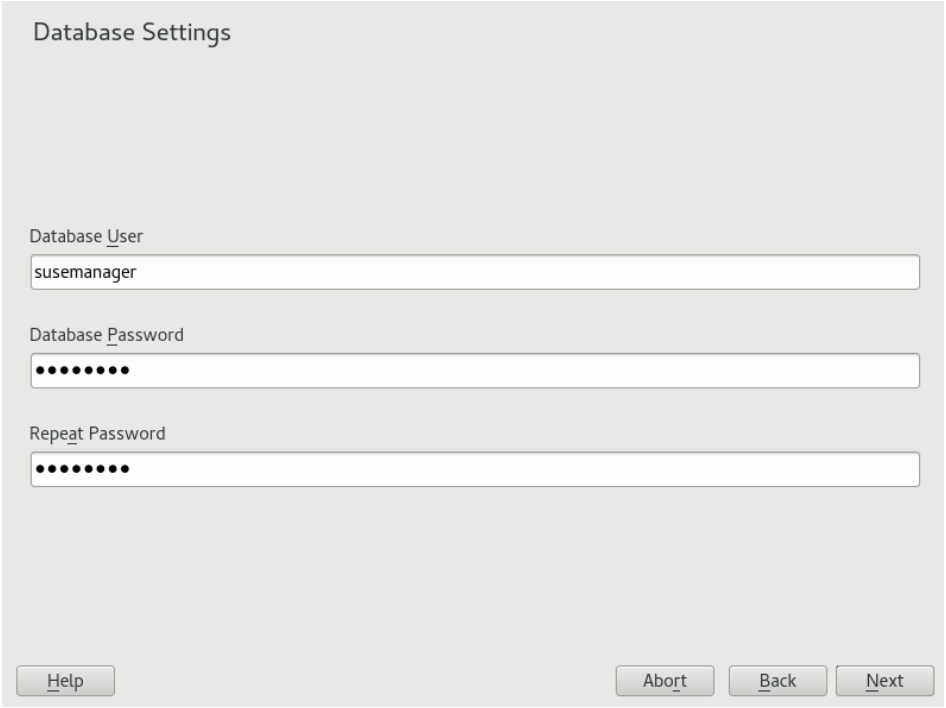
4. Enter your certificate information and a password. The password should be stored in a secure location.

> **Important: Certificate Password**
>
> *Without this password it will not be possible to set up a Uyuni Proxy Server.*

5. Click *Next* to continue.



6. From the *SUSE Manager Setup* › *Database Settings* screen, enter a database user and password. This password should be stored in a secure location. Then click *Next* to continue.

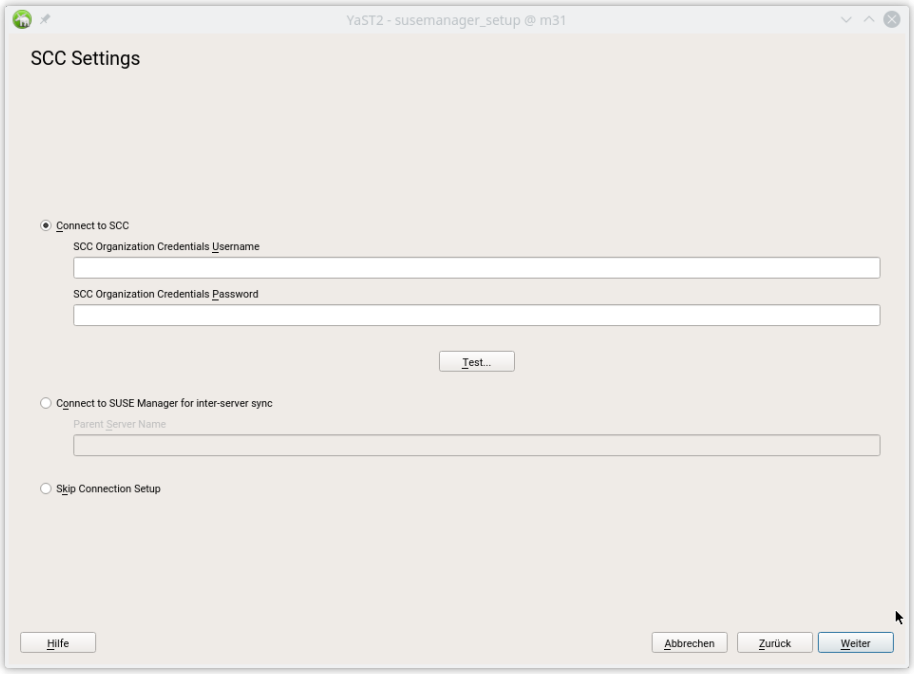


7. Enter your SUSE Customer Center `Organization Credentials`. Open https://scc.suse.com/login to register or access to your organization credentials.

Note: *Skip*

If you are using SUSE Enterprise products, Uyuni requires that you connect to SUSE Customer Center for software, updates and patches. You will not be able to synchronize or provide Enterprise channels to your clients without this information.

However if you would like to work with open source software channels and repositories then click the *Skip* button to continue. You can setup your SUSE Customer Center credentials or configure inter-server sync at a later time.



8. Click *Next* to continue.
9. Click *Yes* to run setup when prompted.
10. Once setup has completed, click *Next* to continue. You will see the address of the Uyuni Web UI.
11. Click *Finish* to complete Uyuni setup.

In the next section you will create the administrator's account and synchronize with SUSE Customer Center.

4.2.1 Creating the Main Administration Account

This section will walk you through creating your organizations main administration account for Uyuni.

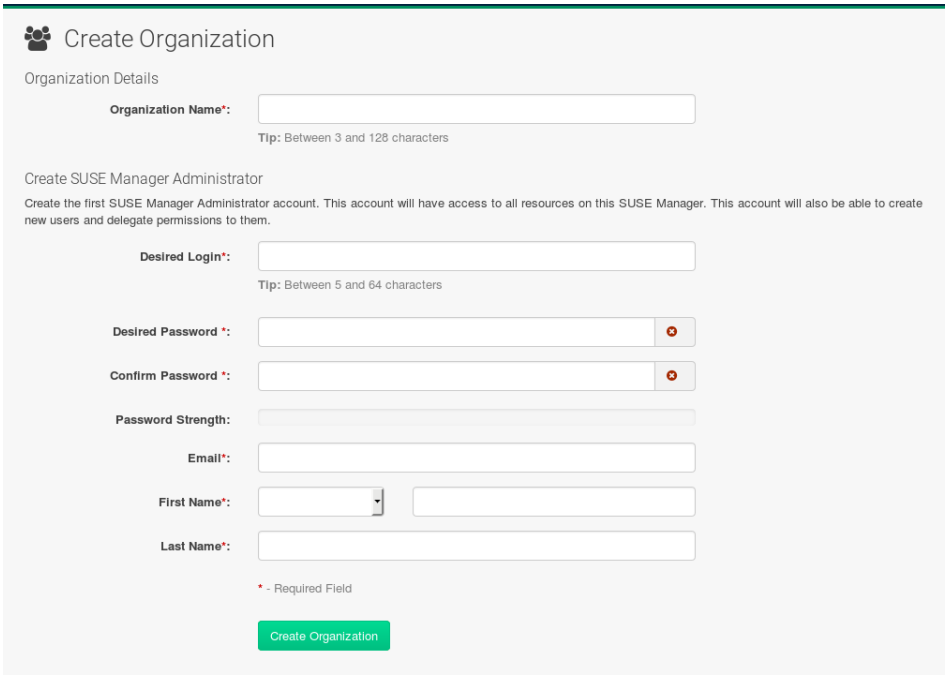
Warning: Admin and User Accounts

The main administration account is the *highest authority account* within Uyuni and therefore account access information should be stored in a secure location.

For security it is recommended that the main administrator creates *low level admin accounts* designated for administration of organizations and individual groups.

PROCEDURE: SETUP THE MAIN ADMINISTRATION ACCOUNT

1. In the browser, enter the address provided after completing setup and open the Uyuni Web UI.
2. Add your organization name to the *Create Organization* › *Organization Name* field.
3. Add your username and password to the *Create Organization* › *Desired Login* and *Create Organization* › *Desired Password* fields.
4. Fill in the Account Information fields including an email for system notifications.
5. Select *Create Organization* to finish creating your administration account.



You should now be presented with the Uyuni Front Page. In the next section you will prepare the server for connecting the first client.

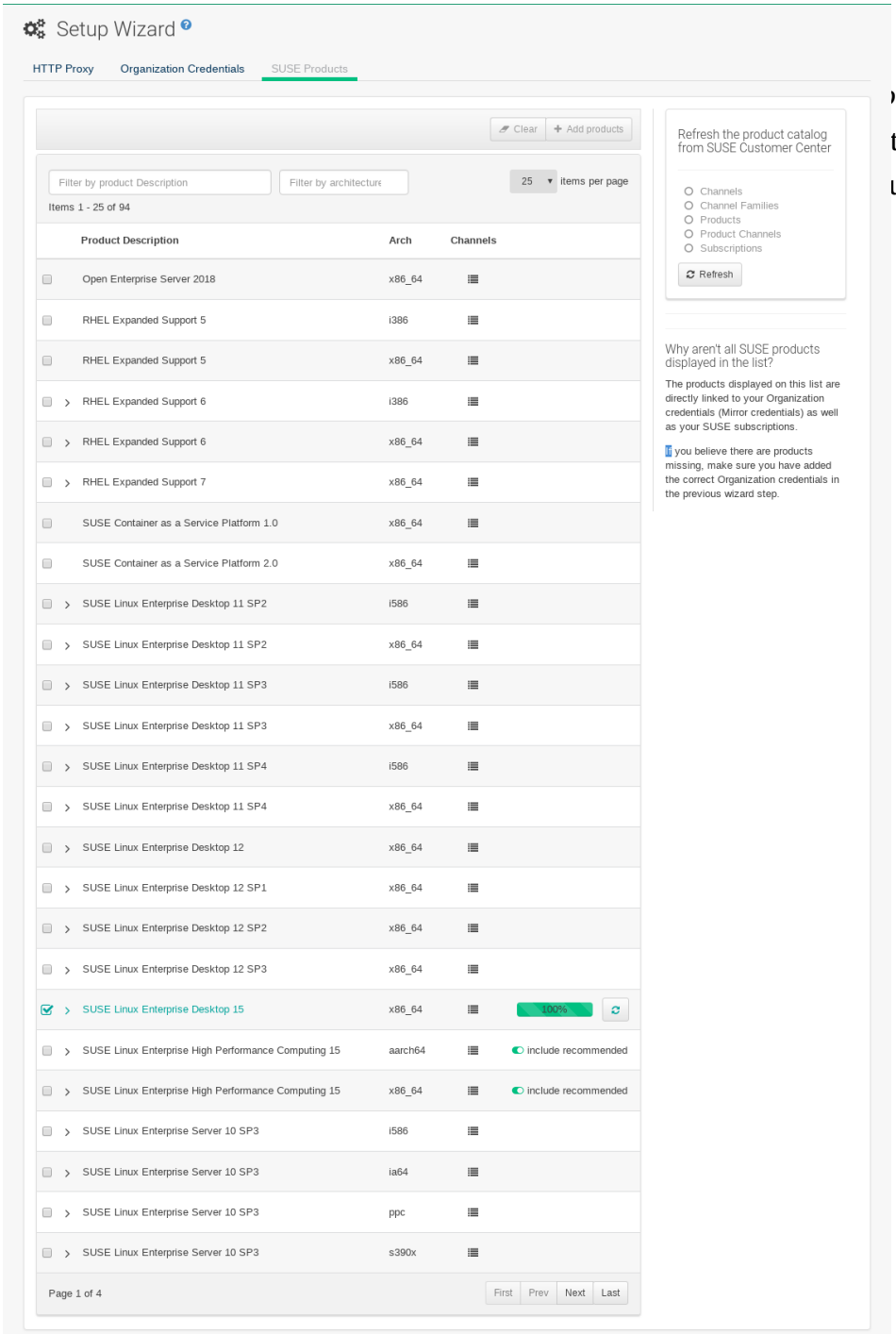
4.2.2 Syncing Products from SUSE Customer Center

SUSE Customer Center (SCC) maintains a collection of repositories which contain packages, software and updates for all supported enterprise client systems. These repositories are organized into channels each of which provide software specific to a distribution, release and architecture. After synchronizing with SCC clients may receive updates, and be organized into groups and assigned to specific product software channels.

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

PROCEDURE: SYNCHRONIZING WITH SUSE CUSTOMER CENTER

1. From the Uyuni Web UI start page select *Admin* › *Setup Wizard*.

2. 

ducts tab. Wait a
th SUSE Customer
ure, channels, and

3. Since Your SUSE Linux Enterprise client is based on `x86_64` architecture scroll down the page and select the check box for this channel now.

- Add channels to Uyuni by selecting the check box to the left of each channel. Click the arrow symbol to the left of the description to unfold a product and list available modules.
- Start product synchronization by clicking the *Add Products* button.

After adding the channel Uyuni will schedule the channel to be copied. This can take a long time as Uyuni will copy channel software sources from the SUSE repositories located at SUSE Customer Center to local `/var/spacewalk/` directory of your server.

Tip: PostgreSQL and Transparant Huge Pages

In some environments, *Transparent Huge Pages* provided by the kernel may slow down PostgreSQL workloads significantly.

To disable *Transparant Huge Pages* set the `transparent_hugepage` kernel parameter to `never`. This has to be changed in `/etc/default/grub` and added to the line `GRUB_CMDLINE_LINUX_DEFAULT`, for example:

```
GRUB_CMDLINE_LINUX_DEFAULT="resume=/dev/sda1 splash=silent quiet showopts
 elevator=noop transparent_hugepage=never"
```

To write the new configuration run `grub2-mkconfig -o /boot/grub2/grub.cfg`. To update the grub2 during boot run `grub2-install /dev/sda`.

Monitor channel synchronization process in real-time by viewing channel log files located in the directory `/var/log/rhn/reposync`:

```
tailf /var/log/rhn/reposync/<CHANNEL_NAME>.log
```

After the channel sync process has completed proceed to: *Chapter 5, Registering Clients*

5 Registering Clients

5.1 Introduction

For Uyuni 3 and later, you can choose to use either traditional or Salt client management framework, or a mixture of both, depending on your environment and requirements.

Salt

Is an end-to-end data-center automation tool which may also be used outside the scope of Uyuni to introduce reactive, real-time orchestration, and configuration management.

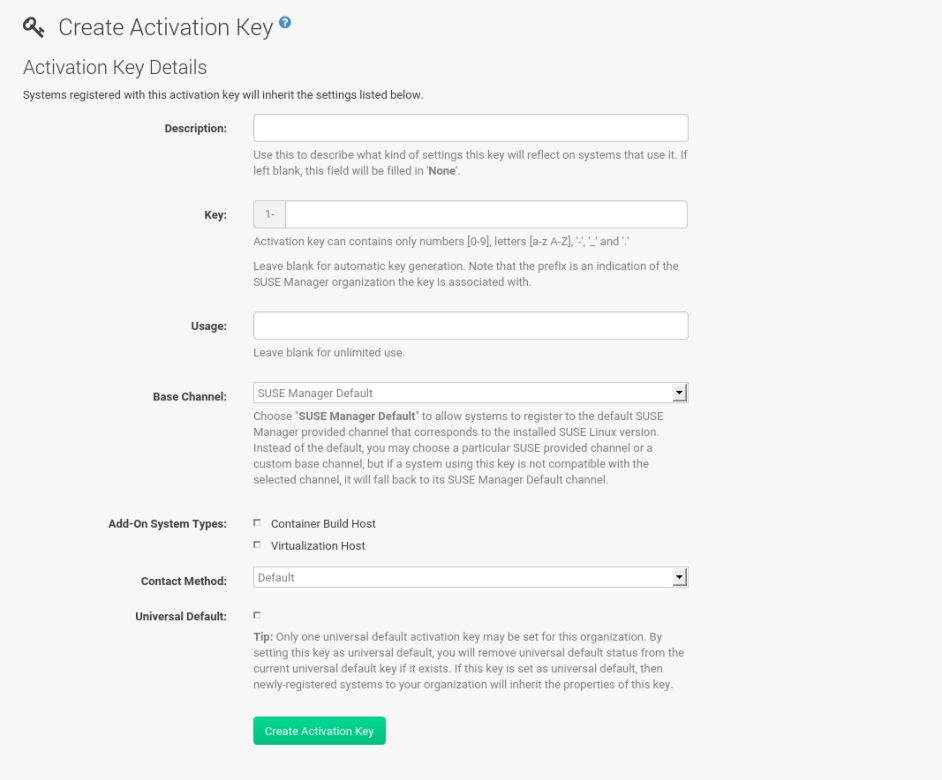
5.2 Creating Activation Keys

Activation keys are used with both 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.

This section contains information on how to create activation keys for both traditional and Salt clients, and provides some best practices for working with activation keys.

PROCEDURE: CREATING ACTIVATION KEYS

1. As the administrator login to the Uyuni Web UI.
2. Navigate to *Systems* › *Activation Keys*.
3. To open the `Activation Key Details` page click the *Create Key* button in the upper right corner.



4. On the `Activation Key Details` page in the `Description` field, enter a name for the activation key.

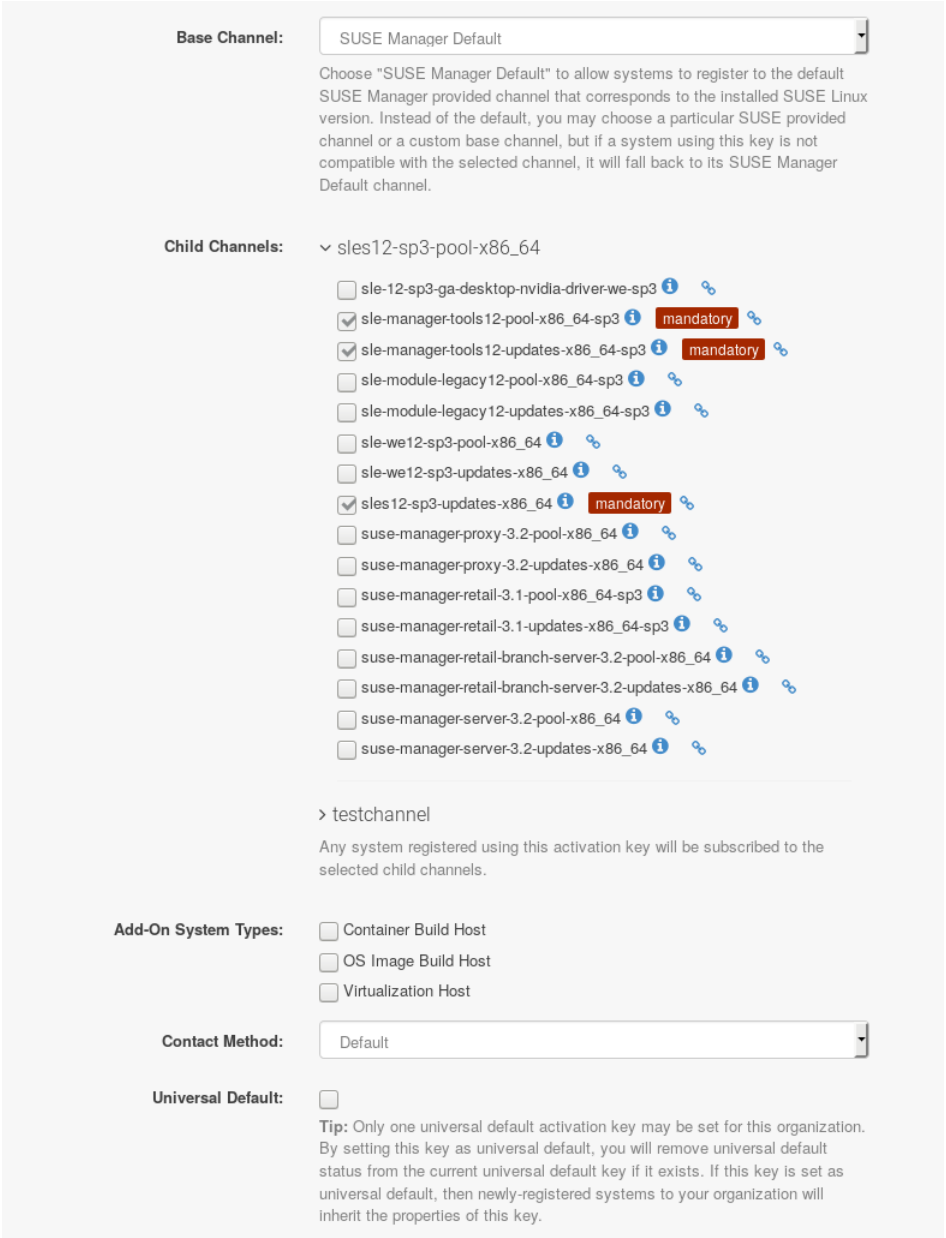
5. In the `Key` field, enter the distribution and service pack associated with the key. For example, `SLES12-SP3` for SUSE Linux Enterprise Server 12 SP3.

Warning: Allowed Characters

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 .

6. In the `Base Channels` drop-down box, select the SUSE Linux Enterprise channel that you added during First Channel Sync (proc-quickstart-first-channel-sync.xml#proc-quickstart-first-channel-sync).

7. When the base channel is selected the list of available child channels will get fetched and displayed in real time below the base channel. Select the child channels you need (for example, the SUSE Manager tools and the updates channels that are actually mandatory).



8. We recommend you leave the `Contact Method` set to `Default`.
9. We recommend you leave the `Universal Default` setting unchecked.
10. Click *Update Activation Key* to create the activation key.
11. Check the `Configuration File Deployment` check box to enable configuration management for this key, and click *Update Activation Key* to save this change.

When you create activation keys, keep these best practices in mind:

- Avoid using the `SUSE Manager Default` parent channel. This setting forces Uyuni 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.
- 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.
- 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.
- Note that 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.

5.3 Creating the Uyuni Tools Repository

In this section you will create a tools repository on the Uyuni Server for providing client tools. The client tools repository contains packages for installing Salt on minions as well as required packages for registering traditional clients during the bootstrapping procedure. These packages will be installed from the newly generated repository during the registration process. In the following procedure you will create the SUSE Linux Enterprise tools repository.

Important: Creating a Tools Repository when an SCC Channel has not been Synced

Before following the procedure to create the tools repository make sure the SUSE vendor channel you will be using with your client has been completely synced. You can check this by running `tail -f /var/log/rhn/reposync/` <CHANNEL_NAME> `.log` as *root*. In the following example replace `version` with the actual version string:

```
# tail -f /var/log/rhn/reposync/sles`version`-pool-x86_64.log
```

Once completed you should see the following output in your terminal:

```
2017/12/12 15:20:32 +02:00 Importing packages started.
2017/12/12 15:22:02 +02:00 1.07 %
...
2017/12/12 15:34:25 +02:00 86.01 %
2017/12/12 15:35:49 +02:00 Importing packages finished.
2017/12/12 15:35:49 +02:00 Linking packages to channel.
...
2017/12/12 15:35:59 +02:00 Sync completed.
```

PROCEDURE: GENERATING THE TOOLS REPOSITORY FOR SUSE LINUX ENTERPRISE

1. Open a terminal on the server as root and enter the following command to list available bootstrap repositories:

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

2. Then invoke the same command using the listed repository as the product label to actually create the bootstrap repository:

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

3. Uyuni will create and add the client tools to the newly created `repositories` directory located at `/srv/www/htdocs/pub/repositories/` .

This repository is suitable for both Server and Desktop of SUSE Linux Enterprise.

5.4 Registering Traditional Clients

5.4.1 Generating a Bootstrap Script

This section goes over generating a template bootstrap script which will be copied and modified for use with “traditional” clients. Traditional clients register with Uyuni via a bootstrap script executed on the client which deploys all necessary packages to it. The bootstrap script contains parameters which assigns a client system to its base channel. Two of these important parameters are:

- Activation Keys
- GNU Privacy Guard (GPG) Keys

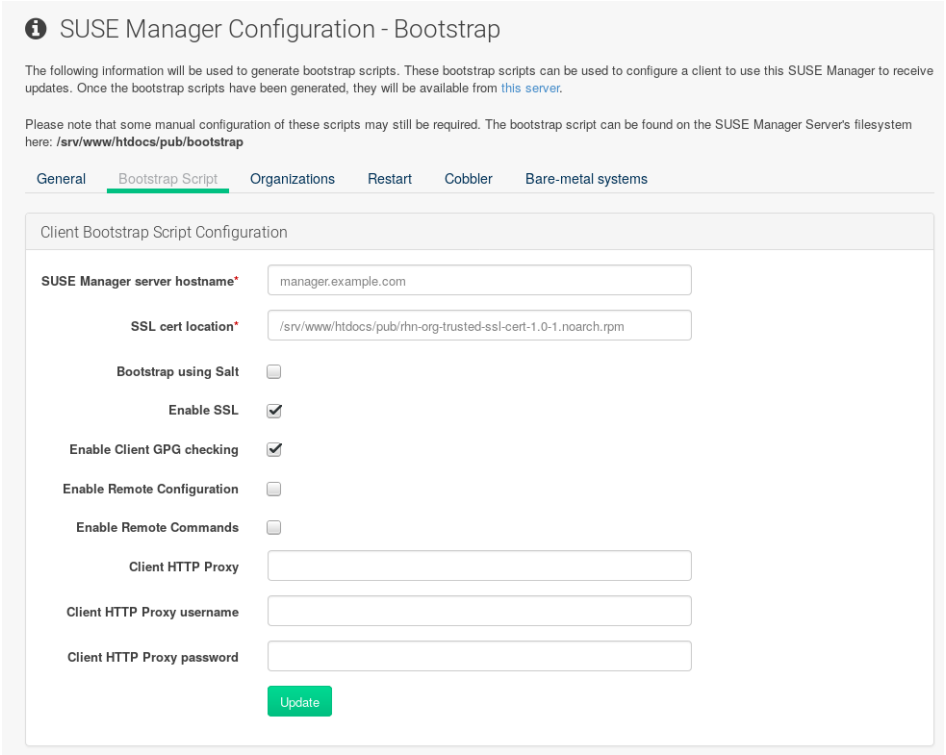
Note: SLES 15 and Python 3

SLES 15 utilizes Python 3 by default. Because of this change any older bootstrap scripts (based on python 2) must be re-created for SLES 15 systems. Attempting to register SLES 15 systems with SUSE Manager using Python 2 versions of the bootstrap script will fail.

The following procedure will guide you through generating a bootstrap script.

PROCEDURE: CREATING A BOOTSTRAP SCRIPT

1. From the Uyuni Web UI, browse to *Main Menu* › *Admin* › *Manager Configuration* › *Bootstrap Script*. For more information, see .
2. In the `SUSE Manager Configuration - Bootstrap` dialog disable `Bootstrap using Salt`. Use default settings and click the *Update* button.



Warning: 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` .

Section 5.4.2, "Editing the Bootstrap Script" will cover copying and modifying your bootstrap template for use with each client.

5.4.2 Editing the Bootstrap Script

In this section you will copy and modify the template bootstrap script you created from *Section 5.4.1, "Generating a Bootstrap Script"*.

A minimal requirement when modifying a bootstrap script for use with Uyuni is the inclusion of an activation key. Depending on your organizations security requirements it is strongly recommended to include one or more (GPG) keys (for example, your organization key, and package signing keys). For this tutorial you will be registering with the activation keys created in the previous section.

PROCEDURE: MODIFYING THE BOOTSTRAP SCRIPT

1. Login as root from the command line on your Uyuni 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. Once you have completed your modifications save the file and repeat this procedure for the second bootstrap script. Proceed to *Section 5.4.3, "Connecting Clients"*.

Note: Finding Your Keys

To find key names you have created: In the Web UI, click *Home* › *Overview* › *Manage Activation keys* › *Key Field*. 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.

5.4.3 Connecting Clients

This section covers connecting your clients to Uyuni with the modified bootstrap script.

PROCEDURE: RUNNING THE BOOTSTRAP SCRIPT

1. From your Uyuni Server command line as root navigate to the following directory:

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

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

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

3. The script will execute and proceed to download the required dependencies located in the repositories directory you created earlier. Once the script has finished running, log in to the Web UI and click *Systems* › *Overview* to see the new client listed.

This concludes the bootstrap section of this guide. *Section 5.5, "Registering Salt Clients"* will go over registering Salt minions for use with Uyuni.

5.4.4 Package Locks

Package locks are used to prevent unauthorized installation or upgrades to software packages on traditional clients. When a package has been locked, it will display to users with 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 Uyuni Web UI, or directly on the client machine using a package manager. Locked packages will also indirectly lock any dependent packages.

Note

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.

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. Note that even if you do not select a date and time, the lock might not activate immediately.

4. To remove a package lock, select the packages to unlock and click *Request Unlock*. You can also choose to enter a date and time for the lock to deactivate. Note that even if you do not select a date and time, the lock might not deactivate immediately.

5.5 Registering Salt Clients

There are currently three methods for registering Salt minions. This section describes the first method and uses a bootstrap repository. The second method uses bootstrap script, and is described in *Section 5.4, "Registering Traditional Clients"*. The third method uses the Web UI, and is described in .

You can also use these methods to change existing traditional clients into Salt minions.

The following section assumes you have created a SUSE Manager tools repository. You can review creating a tools repository in *Section 5.3, "Creating the Uyuni Tools Repository"*.

Warning: Ensure the Salt Master is Reachable During Bootstrap

The Salt master and its proxy should always be reachable via both IP address and the FQDN. In the following rare scenario:

- The Salt master (SUSE Manager) is in some DNS.
- Your Minions are in a different subnet bound to an alternate DNS and the Salt master record is absent.
- The Salt master cannot know that the minion is not utilizing the same DNS record. The the Salt master nevertheless sends the FQDN of itself to the minion expecting it to join.
- The minion looks for a different DNS, one where the master record does not exist therefore bootstrap fails.

Once you have fully synced a base channel from the Web UI for clients to obtain software sources from, for example: `SLES12-SP3-Pool_for_x86_64` perform the following procedure to register a Salt minion.

PROCEDURE: REGISTERING SALT MINIONS

1. On your minion as root enter the following command:

```
zypper ar http://`FQDN.SUSE.Manager.com`/pub/repositories/sle/12/3/bootstrap/ \
   sles12-sp3
```

Note

Do not use `HTTPS`. Use `HTTP` instead to avoid errors.

2. After adding the repository containing the necessary Salt packages execute:

```
zypper in salt-minion
```

3. Modify the minion configuration file to point to the fully qualified domain name (`FQDN`) of the Uyuni server (master):

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

Find and change the line:

```
master: salt
```

to:

```
master:`FQDN.SUSE.Manager.com`
```

4. Restart the Salt minion with:

```
systemctl restart salt-minion
```

Your newly registered minion should now show up within the Web UI under *Salt › Keys*. Accept the `pending` key to begin management.

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.

6 Getting Started with Salt

6.1 Introduction

This section introduces you to the new Salt features added in Uyuni 3. This chapter assumes you have completed all previous Getting Started sections. At a minimum have the following setup:

- A freshly installed Uyuni server with a main admin account and a synced product channel
- Preferably two registered Salt minions to experiment with.

If you find yourself stuck at any point refer to the *SaltStack Get Started* tutorial located at https://docs.saltstack.com/en/getstarted/fundamentals/index.html.

Note

This guide does not attempt to cover all that Salt has to offer. This guide is a primer for using Salt with Uyuni. For comprehensive Salt documentation, see https://docs.saltstack.com/en/latest/contents.html.

The current version of Salt in Uyuni is 2018.3.0.

6.2 Understanding Salt Calls

Salt Calls

Salt calls are defined by three main properties:

```
salt 'target' <function> [arguments]
```

Target

Use the second property in a Salt call to target a single machine or group of machines. Specify the minion or group of minions you would like to run a function on.

General Targeting

List available grains on all minions:

```
salt '*' grains.ls
```

Ping a specific minion:

```
salt 'web1.example.com' test.ping
```

Glob Targeting

Ping all minions using a domain:

```
salt '*example.com' test.ping
```

Display the OS name of all minions with the `webserver` label:

```
salt 'webserver*' grains.item oscodename
```

List Targeting

```
salt -L 'webserver.example.com,db.example.com' test.ping
```

Regular Expression Targeting

You may use PCRE-compliant regular expressions:

```
salt -E '(?!web)' test.ping
```

IP Address Targeting

List minion IP addresses:

```
salt '*' network.ip_addrs
```

Ping a specific minion IP address:

```
salt -S '172.31.60.74' test.ping
```

Ping all minions on a subnet:

```
salt -S 172.31.0.0/16 test.ping
```

Tip: Lookup a Subnet Using the `ip` Command

You can use the `ip` command to find the subnet mask in the format of `192.168.1.1/24`:

```
ip -o -f inet addr show | awk '/scope global/ {print $4}'
```

Function

Once you have specified a target, provide the function you would like to call. Functions also accept arguments. Arguments are space-delimited, for example:

```
salt '*' cmd.run 'echo "Hello: $FIRST_NAME"' env='{FIRST_NAME: "John"}'
```

Locating Additional Minion Functions

Find more functions which can be called on minions by running:

```
salt '*' sys.doc
```

A full list of callable functions are located here: https://docs.saltstack.com/en/2015.8/ref/modules/all/index.html

Arguments

Provides the extra data needed by a function you are calling. The command `pkg.install` requires an argument specifying a package to install. YaST has been selected for installation, for example:

```
salt '*' pkg.install yast2
```

6.3 Common Salt Terminology

Grains

Grains provide information about the hardware of a minion. For example, the operating system, IP addresses, network interfaces, memory, etc. When running a Salt command from keep in mind any modules and functions called are run locally from the system being called. Salt modules are stored on minions and master within the following directory:

```
/usr/lib/python2.7/site-packages/salt/
```

List all available grains with the `grains.ls` function:

```
salt '*' grains.ls
```

List collected grain system data by using the `grains.ls` function:

```
salt '*' grains.items
```

For more information on grains, see https://docs.saltstack.com/en/latest/topics/grains/.

States

States are templates which place systems into a known configuration, for example which applications and services are installed and running on those systems. States are a way for you to describe what each of your systems should look like. Once written, states are applied to target systems automating the process of managing and maintaining a large numbers of systems into a known state. For more information on states, see https://docs.saltstack.com/en/latest/topics/tutorials/starting_states.html.

Warning: Updating Salt

Do not update salt itself using Salt states. First update all other system packages using Salt states then update salt as a separate stand-alone step from the Uyuni Web UI.

Pillar

Pillars unlike grains are created on the master. Pillar files contain information about a minion or group of minions. Pillars allow you to send confidential information to a targeted minion or group of minions. Pillars are useful for sensitive data, configuration of minions, variables, and any arbitrary data which should be defined. For more information on pillars, see https://docs.saltstack.com/en/latest/topics/tutorials/pillar.html.

Beacons

Beacons allow an admin to use the event system in Salt to monitor non-Salt processes. Minions may use beacons to hook into many types of system proceses for constant monitoring. Once a targeted monitored activity occurs an event is sent on the Salt event bus that may be used to trigger a reactor.

Important: Enabling Beacons

To work with beacons on Salt minions the package python-pyinotify must be installed for SUSE systems. For RES systems install python-inotify. This package is not installed automatically during the salt minion package installation.

Note: Peer Communication with salt-broker

The salt-broker acts like a switch and not like a hub, therefore Peer communication will only work for minions behind the same broker/Proxy. For more information on Salt and peer communication see: https://docs.saltstack.com/en/latest/ref/peer.html

6.4 Useful Salt Commands

The following list provides several useful Salt commands.

salt-run

Print a list of all minions that are up:

```
salt-run manage.up
```

Print a list of all minions that are down:

```
salt-run manage.down
```

Print a list with the current status of all Salt minions:

```
salt-run manage.status
```

Check the version of Salt running on the master and active minions:

```
salt-run manage.versions
```

salt-cp

Copy a file to a minion or set of minions.

```
salt-cp '*' foo.conf /root
```

For more information, see https://docs.saltstack.com/en/latest/ref/cli/salt-cp.html.

salt-key -l

List public keys:

```
salt-key -l
```

salt-key -A

Accept all pending keys:

```
salt-key -A
```

6.5 Salt File Locations and Structure

The following screen describes Salt file structures and their locations used by the Uyuni Server. These files are listed in `/etc/salt/master.d/susemanager.conf`:

```
# Configure different file roots

file_roots:
  base:
    - /usr/share/susemanager/salt    #Should not be touched by a user
    - /srv/susemanager/salt          #Should not be touched by a user
    - /srv/salt                      #Your custom states go here

# Configure different pillar roots

pillar_roots:
  base:
    - /usr/share/susemanager/pillar  #Should not be touched by a user
    - /srv/pillar                    #Custom pillars go here

# Extension modules path

extension_modules: /usr/share/susemanager/modules

# Master top configuration

master_tops:
  mgr_master_tops: True
```

The following tips should be kept in mind when working with `/etc/salt/master.d/susemanager.conf`.

- Files listed are searched in the order they appear.
- The first file found is called.

6.5.1 file_roots

Uyuni as the Salt master reads its state data from three specific file root directories.

/usr/share/susemanager/salt

This directory is created by Uyuni and its content generated by the `/usr/share/susemanager/modules/tops/mgr_master_tops.py` python module. It is shipped and updated together with Uyuni and includes certificate setup and common state logic that will be applied to packages and channels.

Warning: Do Not Edit

You should not edit or add custom Salt data to this directory.

/srv/susemanager/salt

This directory is created by Uyuni and contains assigned channels and packages for minions, groups, and organizations. These files will be overwritten and regenerated. A good analogy for this directory would be the SUSE Manager database translated into Salt directives.

Warning: Do Not Edit

You should not edit or add custom Salt data to this directory.

/srv/salt

The directory `/srv/salt` is for your custom state data, salt modules etc. SUSE Manager does not perform any actions on this directory. However the state data placed here affects the Highstate of minions and is merged with the result generated by Uyuni.

Note: Editable

Place custom Salt data here.

6.5.2 pillar_roots

Uyuni as the Salt master reads its pillar data from two specific pillar root directories.

/usr/share/susemanager/pillar

This directory is generated by Uyuni. It is shipped and updated together with Uyuni.

Warning: Do Not Edit

You should not edit or add custom Salt data to this directory.

/srv/pillar

Uyuni by default does not touch or do anything with this directory. However the custom pillar data placed here is merged with the pillar result created by Uyuni.

Tip: Editable Directory

Place your custom Salt pillar data here.

6.6 Install the SUSE Manager Locale Formula

The following section provides guidance on installing and using SUSE provided Salt formulas.

PROCEDURE: INSTALLING THE LOCALE FORMULA

1. Install the locale formula with:

```
zypper install locale-formula
```

Note

This installs the package contents to `/usr/share/susemanager/formulas/{metadata,states}`

2. After installing the RPM, log in to the Uyuni Web UI.
3. Browse to the *Main Menu* › *System Details* page of any minion you would like to apply the formula to.
4. On the *Main Menu* › *System Details* page you will see a new *Formulas* tab. Select it to view a list of installed formulas.
5. From the *Formulas* list select *Formulas* › *Locale* and click *Save*.

6. A new tab will appear next to the *Formula* › *Locale* subtab. Select the new *Formulas* › *Locale* tab.

7. The *Formalas* › *Locale* tab contains options for setting the language, keyboard layout, timezone, and whether hardware clock is set to UTC. Select the desired options and click *Save*.

8. Run the following command to verify pillar settings. The output has been truncated.

```
salt '$your_minion' pillar.items
```

```
...
   keyboard_and_language:
       ----------
       keyboard_layout:
           English (US)
       language:
           English (US)
   machine_password:
       foobar
   mgr_server:
       manager_server
   org_id:alt '$your_minion_here'
       1
   timezone:
       ----------
       hardware_clock_set_to_utc:
           True
       name:
           CET
    ...
```

9. Apply this state to your minion by applying the highstate from the command line with:

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

Note

You can also apply the highstate from the previous formula tab from the Uyuni Web UI by selecting *System Details* › *States* and clicking *Apply Highstate*.