



Deployment Guide

SUSE CaaS Platform 2



Deployment Guide

SUSE CaaS Platform 2

by Jana Halačková and Christoph Wickert

Publication Date: May 06, 2022

SUSE LLC

1800 South Novell Place


Provo, UT 84606

USA

<https://documentation.suse.com> 

Copyright © 2006– 2022 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see <http://www.suse.com/company/legal/> . All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

About This Guide v

- 1 Required Background v
- 2 Available Documentation v
- 3 Feedback vi
- 4 Documentation Conventions vi
- 5 About the Making of This Documentation viii

1 About SUSE CaaS Platform 1

- 1.1 SUSE CaaS Platform Architecture 1
 - The Administration Node 4

2 SUSE CaaS Platform Cluster Deployment 5

- 2.1 Installing SUSE CaaS Platform 5
 - System Requirements 5 • Installing the Administration Node 7 • Initial Cluster Configuration 14 • Installing Cluster Nodes 16 • Bootstrapping the Cluster 19

3 Configuration 21

- 3.1 Configuration 21
- 3.2 `cloud-init` Configuration 21
 - The `LocalDisk` Datasource 22 • The `NoCloud` Datasource 22 • The `cloud.cfg` Configuration File 22 • The meta-data Configuration File 23 • The user-data Configuration File 23 • The vendor-data Configuration File 28

4 Cluster Administration 30

- 4.1 Authentication and Authorization 30
 - Authentication 30 • Managing Users and Groups 31

4.2	Handling Transactional Updates	34
	Manual Installation of Updates	35 • Disabling Transactional Updates 39
4.3	Handling Program Temporary Fixes	39
4.4	Commands for Single Node Management	39
	The issue-generator Command	40 • The transactional-update Command 40 • The create_autoyast_profile Command 42
4.5	Adding Nodes	42
4.6	Cluster Monitoring	45
	Monitoring with Heapster	45 • Monitoring with Grafana 46
5	Software Installation	47
5.1	Deploying Helm and Tiller	47
5.2	Installing Kubernetes Dashboard	47
6	SUSE CaaS Platform Integration with SES	49
6.1	Prerequisites	49
6.2	Mounting a Named Object Storage to a Container	49
6.3	Creating Pods with Persistent Volumes	51
7	Upgrading SUSE CaaS Platform	54
7.1	Upgrading from SUSE CaaS Platform 1	54
	Migrating Users	54 • Upgrading etcd 55 • Adding new settings 56 • Generating the Service Account Key File on the CA 56
A	GNU Licenses	58

About This Guide

SUSE CaaS Platform is a minimalistic operating system designed to be used with containers. SUSE CaaS Platform allows for provisioning, management, and scaling of container-based applications. It is optimized for large, clustered deployments.

To optimize the use of containers, you need an orchestration framework. Kubernetes is typically used for this purpose. Kubernetes is a tool that provides a platform for automating deployment, scaling, and operations of containers.

SUSE CaaS Platform enables the following:


- fast and predictable deployment
- scaling applications on the fly
- roll out new features
- optimization of hardware by using only the required resources.

1 Required Background

To keep the scope of these guidelines manageable, certain technical assumptions have been made:

- You have some computer experience and are familiar with common technical terms.
- You are familiar with the documentation for your system and the network on which it runs.
- You have a basic understanding of Linux systems.

2 Available Documentation

We provide HTML and PDF versions of our books in different languages. Documentation for our products is available at <http://www.suse.com/documentation/> , where you can also find the latest updates and browse or download the documentation in various formats.

The following documentation is available for this product:

Deployment Guide

The SUSE CaaS Platform deployment guide gives you details about installation and configuration of SUSE CaaS Platform along with a description of architecture and minimum system requirements.

3 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests

For services and support options available for your product, refer to <http://www.suse.com/support/>.

To report bugs for a product component, go to <https://scc.suse.com/support/requests>, log in, and click *Create New*.

User Comments

We want to hear your comments about and suggestions for this manual and the other documentation included with this product. Use the User Comments feature at the bottom of each page in the online documentation or go to <http://www.suse.com/documentation/feedback.html> and enter your comments there.







Mail

For feedback on the documentation of this product, you can also send a mail to doc-team@suse.com. Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

4 Documentation Conventions

The following notices and typographical conventions are used in this documentation:

- /etc/passwd: directory names and file names
- PLACEHOLDER: replace PLACEHOLDER with the actual value
- PATH: the environment variable PATH
- ls, --help: commands, options, and parameters

- user : users or groups
- package name : name of a package
- ,  : a key to press or a key combination; keys are shown in uppercase as on a keyboard
- *File*, *File* > *Save As*: menu items, buttons
-  This paragraph is only relevant for the AMD64/Intel 64 architecture. The arrows mark the beginning and the end of the text block. 
-  This paragraph is only relevant for the architectures z Systems and POWER. The arrows mark the beginning and the end of the text block. 
- *Dancing Penguins* (Chapter *Penguins*, ↑*Another Manual*): This is a reference to a chapter in another manual.
- Commands that must be run with root privileges. Often you can also prefix these commands with the sudo command to run them as non-privileged user.

```
root # command
tux > sudo command
```

- Commands that can be run by non-privileged users.

```
tux > command
```

- Notices



Warning: Warning Notice

Vital information you must be aware of before proceeding. Warns you about security issues, potential loss of data, damage to hardware, or physical hazards.



Important: Important Notice

Important information you should be aware of before proceeding.



Note: Note Notice





Additional information, for example about differences in software versions.




Tip: Tip Notice

Helpful information, like a guideline or a piece of practical advice.

5 About the Making of This Documentation

This documentation is written in SUSEDoc, a subset of DocBook 5 (<http://www.docbook.org> ) . The XML source files were validated by jing (see <https://code.google.com/p/jing-trang/> ) , processed by xsltproc , and converted into XSL-FO using a customized version of Norman Walsh's stylesheets. The final PDF is formatted through FOP from Apache Software Foundation (<https://xmlgraphics.apache.org/fop>)  . The open source tools and the environment used to build this documentation are provided by the DocBook Authoring and Publishing Suite (DAPS). The project's home page can be found at <https://github.com/openSUSE/daps>  .

The XML source code of this documentation can be found at <https://github.com/SUSE/doc-caasp>  .

1 About SUSE CaaS Platform

SUSE® CaaS (Container as a Service) Platform is a platform designed for fast deployment of container-based applications. You can deploy SUSE CaaS Platform on your physical machines or use it on virtual machines. After deployment, it is ready to run out of the box and provides high scalability of your cluster.

SUSE CaaS Platform was designed to simplify transformation of applications into container-based applications. Creating clouds with container-based applications is much easier as you get rid of problems with compatibility and dependencies. To enable as fast application deployment as possible, it is recommended to use a container orchestration framework, e.g. Kubernetes. While SUSE CaaS Platform inherits benefits of SUSE Linux Enterprise and uses tools and technologies well-known to system administrators—like `cloud-init`, Kubernetes, or Salt—the main innovation (compared to SUSE Linux Enterprise Server) comes with **transactional updates**. A transactional update is an update that can be installed when the system is running without any down time. The transaction update can be rolled back, so if the upgrade fails or the update is not compatible with your infrastructure, you can restore the previous state.

SUSE CaaS Platform uses the Btrfs file system with the following characteristic:

- The base OS and snapshots are read-only.
- Sub volumes for data sharing are read-write.
- SUSE CaaS Platform introduces overlays of `/etc` directories used by `cloud-init` and Salt

1.1 SUSE CaaS Platform Architecture

A typical SUSE CaaS Platform cluster consists of several types of nodes:

- administration node - is a Salt master that assigns roles to Salt minions. The node runs the GUI dashboard that manages the whole cluster. For details refer to [Section 1.1.1, “The Administration Node”](#).
- cluster node - is a Salt minion that can have one of the following roles:
 - Kubernetes master - that manages nodes running containers.
 - Kubernetes worker - that runs containers.

In large-scale clusters, there are other nodes that can help you to manage and run the cluster:

- a local SMT server that manages subscriptions for workers and so decreases the traffic to SUSE Customer Center
- a log server that stores logs of cluster nodes.

The following figure illustrates interactions of the nodes.

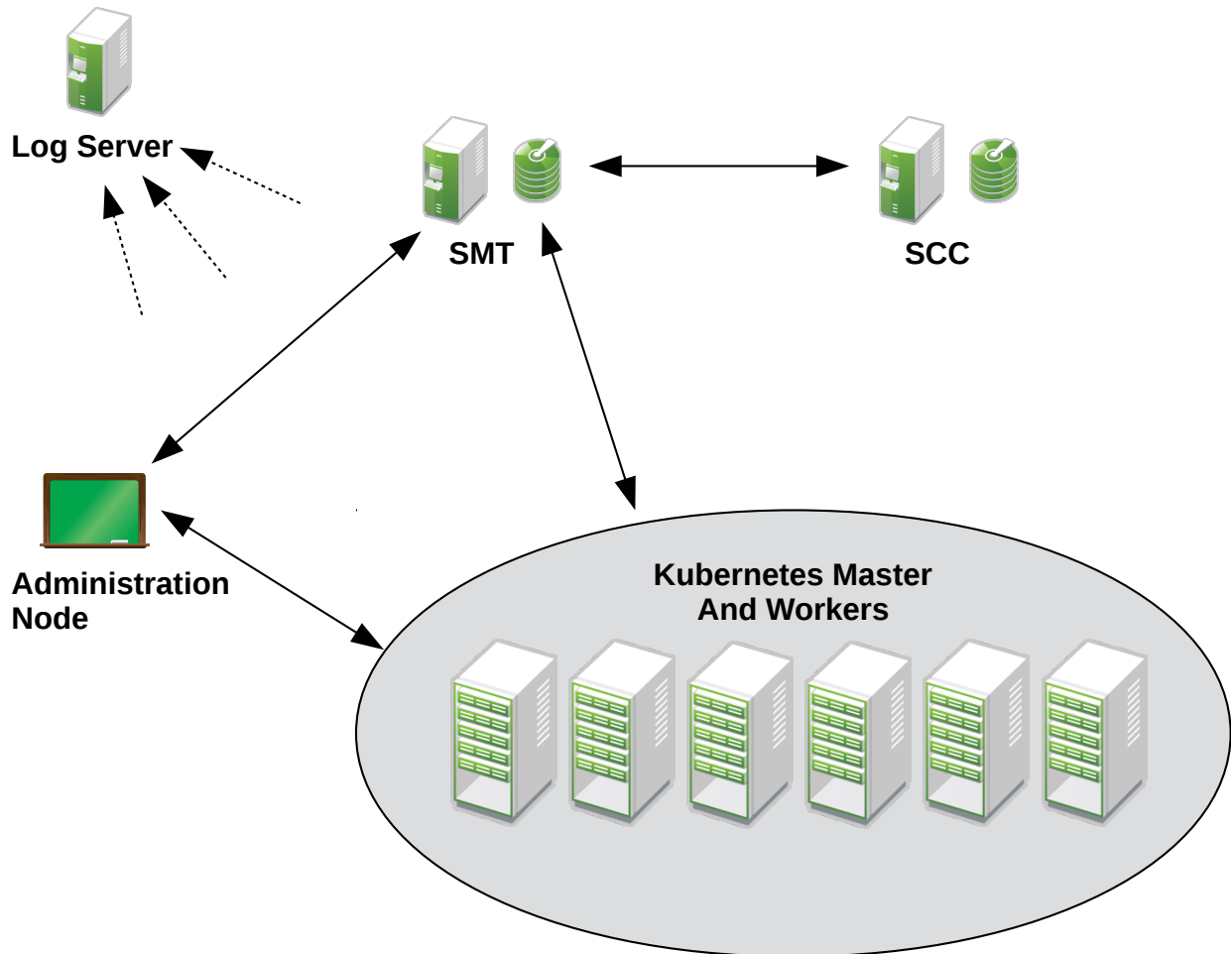


FIGURE 1.1: SUSE CAAS PLATFORM NODES ARCHITECTURE

To run the whole cluster, SUSE CaaS Platform uses various technologies like Salt, flannel network, `etcd` cluster, controller, scheduler, `kubelet`, container engines, and an API Server.

Salt is used to manage deployment and administration of the cluster. The `Salt-api` is used to distribute commands from Velum to the `salt-master` daemon. The `salt-master` daemon stores events in *MariaDB* (the database is also used to store Velum data). The `salt-minion` daemon on the administration node is used to generate required certificates, and on the Salt minions the daemons communicate with the administration node.

As there can be several containers running on each host machine, each container is assigned an IP address that is used for communication with other containers on the same host machine. Containers might need to have a unique IP address exposed for network communication, thus flannel networking is used. Flannel gives each host an IP sub net from which the container engine can allocate IP addresses for containers. The mapping of IP addresses is stored by using etcd. The flanneld is used to manage routing of packets and mapping of IP addresses.

Within the cluster there are several instances of etcd, each with a different purpose. The etcd discovery running on the administration node is used to bootstrap instances of etcd running on other nodes and is not part of the etcd cluster on other nodes. The etcd instance on the master node stores events from the API Server. The etcd instance on worker nodes runs as a proxy that forwards clients to the etcd on the master node.

Kubernetes is used to manage container orchestration. The following services and daemons are used by Kubernetes:

- kubelet - is a node agent that ensures that all containers in a pod are running and healthy.
- kube-apiserver - exposes an REST API used to manage pods. The API server performs authentication and authorization.
- *scheduler*
- a set of *controllers* for handling pod replication, deployment, etc.
- kube-proxy - runs on each node and is used to distribute load and reach services.

Now let's focus on a more detailed view of the cluster that involves also services running on each node type.

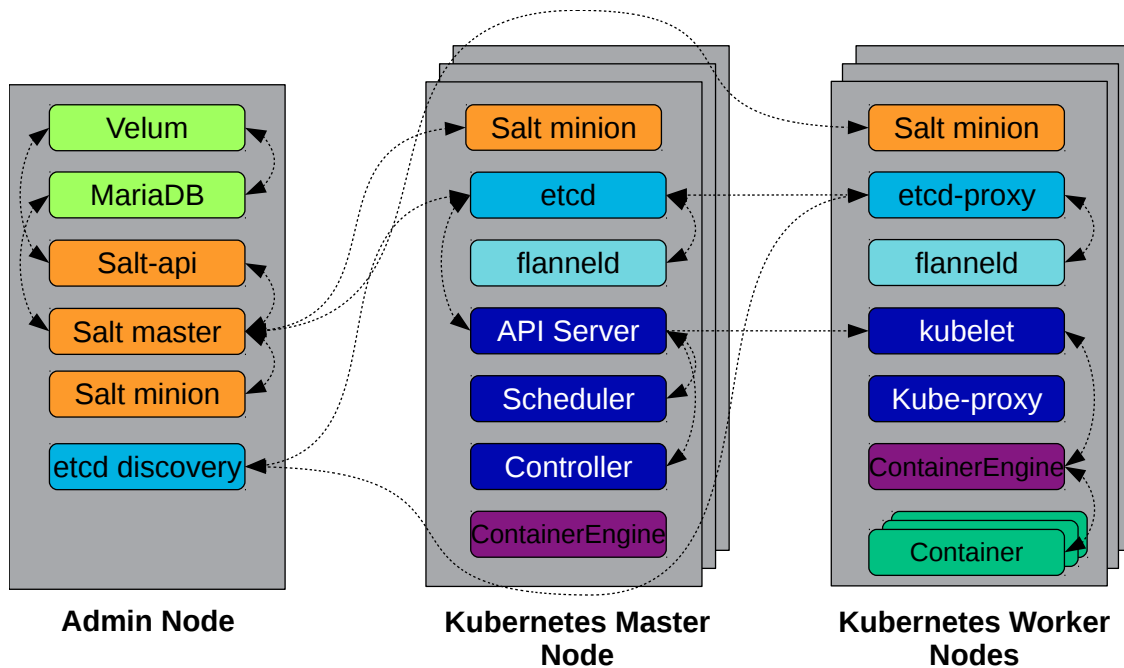


FIGURE 1.2: SERVICES ON NODES

1.1.1 The Administration Node

The administration node is intended to manage deployment of the cluster and run applications required for a proper function of the cluster. The administration node runs the administration dashboard called Velum, the *salt-api*, the MariaDB database, the *etcd discovery* server, the *salt-master* and *salt-minion*. The dashboard, database, and daemons all run in separate containers.

The dashboard is a web application that enables you to manage, monitor, and deploy the cluster. The dashboard manages the cluster by using the *salt-api* to interact with the underlying Salt technology.

The containers on the administration node are managed by *kubelet* as a static pod. Bear in mind that this *kubelet* does not manage the cluster nodes. Each cluster node has its own running instance of *kubelet*.

2 SUSE CaaS Platform Cluster Deployment

This chapter gives an overview on how to run through a default installation of SUSE CaaS Platform for the AMD64/Intel 64 architecture.

2.1 Installing SUSE CaaS Platform

Use the following procedures to set up a new SUSE® CaaS Platform 2 cluster.

2.1.1 System Requirements

Before you begin the installation, please make sure your system meets all requirements listed below.

2.1.1.1 Minimum Hardware Requirements

- Any AMD64/Intel* EM64T processor (quad-core processors are recommended). 32-bit processors are not supported.
- 8 GB physical RAM per node.
- 40 GB available disk space (or more depending on the count and size of container images you intend to run).

2.1.1.2 Cluster Size Requirements

SUSE CaaS Platform only functions in a clustered configuration. It requires a group of physical or virtual machines in order to operate. The minimum supported cluster size is four nodes: one administration node, one master node, and two worker nodes.



Note: Test and Proof-of-Concept Clusters

It is possible to provision a three-node cluster with only a single worker node, but this is not a supported configuration for deployment.

2.1.1.3 Network Requirements

- In order to communicate without interruptions, all nodes should be on the same network.
- A DHCP server to dynamically provide IP addresses and host names for the nodes in your cluster (unless you configure all nodes statically).
- A DNS server to resolve host names. Please make sure you have reliable DNS resolution at all times, especially in combination with DHCP.

Important: Unique Host Names

Host names must be unique. It is recommended to let the DHCP server provide not only IP addresses but also host names of the cluster nodes.

- In a SUSE CaaS Platform cluster, internal TCP/IP ports are managed using iptables controlled by Salt and so need not be manually configured. However, for reference and for environments where there are existing security policies, the following are the standard ports in use.

TABLE 2.1: NODE TYPES AND OPEN PORTS

Node	Port	Internal / External	Open after orchestration	Description	Notes
All nodes	22	Internal	No	ssh	Useful but not essential
Admin	80	External	No	HTTP	
	389	External	No	LDAP	User management
	443	External	No	HTTPS	
	2379	Internal	No	<u>etcd</u> discovery	
	4505 - 4506	Internal	No	Salt	

Node	Port	Internal / External	Open after orchestration	Description	Notes
Masters	2380	Internal	Yes	<u>etcd</u> control	Peer-to-peer <u>etcd</u> traffic
	4789	Internal	No	VXLAN traffic	Used by Flannel
	6443	Both	Yes	Kubernetes API server	
Workers	2380	Internal	Yes	<u>etcd</u> control	Peer-to-peer <u>etcd</u> traffic
	4789	Internal	No	VXLAN traffic	Used by Flannel
	10250	Internal	No	Kubelet	

When some additional ingress mechanism is used, additional ports would also be open.

2.1.1.4 Limitations

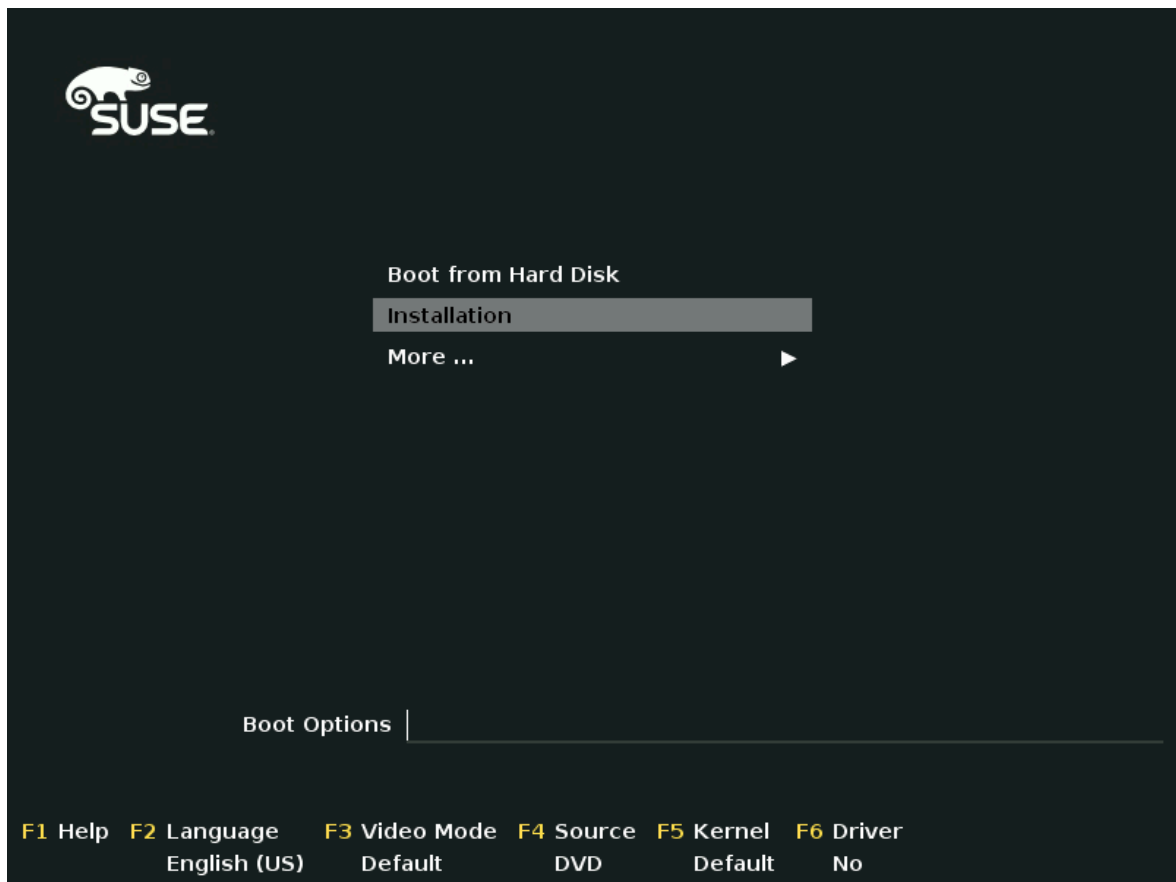
- SUSE CaaS Platform 2 does not support remote installations with Virtual Network Computing (VNC).

2.1.2 Installing the Administration Node

Use these instructions if there is no existing Linux system on your machine, or if you want to replace an existing Linux system.

1. Insert the SUSE CaaS Platform DVD into the drive, then reboot the computer to start the installation program. On machines with a traditional BIOS you will see the graphical boot screen shown below. On machines equipped with UEFI, a slightly different boot screen is used. Secure boot on UEFI machines is supported.

Use **F2** to change the language for the installer. A corresponding keyboard layout is chosen automatically. See https://www.suse.com/documentation/sles-12/book_sle_deployment/data/sec_i_yast2_startup.html for more information about changing boot options.



2. Select *Installation* on the boot screen, then press **Enter**. This boots the system and loads the SUSE CaaS Platform installer.
3. Configure the following mandatory settings on the *Installation Overview* screen.



Note: Help and Release Notes

From this point on, a brief help document and the Release Notes can be viewed from any screen during the installation process by selecting *Help* or *Release Notes* respectively.

Keyboard Layout

The *Keyboard Layout* is initialized with the language settings you have chosen on the boot screen. Change it here, if necessary.

Password for root User

Type a password for the system administrator account (called the root user) and confirm it.



Warning: Do not forget the root Password

You should never forget the root password! After you entered it here, the password cannot be retrieved. See https://www.suse.com/documentation/sles-12/book_sle_deployment/data/sec_i_yast2_user_root.html for more information.

Registration Code or SMT Server URL

Enter the *Registration Code or SMT Server URL*. SMT Server URLs must use the https or http protocol; other schemes are not supported.

System Role

Select the *System Role* for your system.

Administration Node (Dashboard)

The administration node providing the administration dashboard for the cluster.

Cluster Node

A cluster node running containers or the container orchestration framework.


Plain System

A plain node that can be used for testing and debugging purposes in your cluster, but it is not usually needed.

NTP Servers

Enter host names or IP addresses of one or more *NTP Servers* for the node, separated by white spaces or colons. While a single server is sufficient, nodes should use at least three for optimal precision and reliability.

For more information about NTP, refer to https://www.suse.com/documentation/sles-12/book_sle_admin/data/cha_netz_xntp.html



Installation Overview

Language
English (US)

Keyboard Layout
English (US)

Password for root User

Confirm Password

Registration Code or SMT Server URL

System Role
Administration Node (Dashboard)

NTP Servers

System Information

Partitioning

* Standard

Booting

* Boot Loader Type: GRUB2

* Enable Trusted Boot: no

* Status Location: /dev/vda2 ("/")

Network Configuration

* DHCP / eth0

Kdump

* Kdump status: enabled

Help

Release Notes...

Install

Optionally, you can customize the following settings. If you do not make any changes, defaults are used. A brief summary of the settings is displayed below the respective settings option.

Partitioning

Review the partition setup proposed by the system and change it if necessary. You have the following options:

Select a hard disk

Select a disk to which install SUSE CaaS Platform to with the recommended partitioning scheme.

Custom Partitioning (for Experts)

Opens the *Expert Partitioner* described in https://www.suse.com/documentation/sles-12/book_sle_deployment/data/sec_yast2_i_y2_part_expert.html .



Warning: For Experts only

As the name suggests, the *Expert Partitioner* is for experts only. Custom partitioning schemes that do not meet the requirements of SUSE CaaS Platform are not supported.

REQUIREMENTS FOR CUSTOM PARTITIONING SCHEMES

- SUSE CaaS Platform exclusively supports the file system types Btrfs and OverlayFS. A read-only Btrfs file system is used for the root file system which enables transactional updates.
- For snapshots, partitions must have a capacity of at least 11 GB.
- Depending on the number and size of your containers, you will need sufficient space under the `/var` mount point.

To accept the proposed setup without any changes, choose *Next* to proceed.

Bootimg

This section shows the boot loader configuration. Changing the defaults is only recommended if really needed. Refer to https://www.suse.com/documentation/sles-12/book_sle_admin/data/cha_grub2.html for details.

Network Configuration

If the network could not be configured automatically while starting the installation system, you have to manually configure the *Network Settings*. Please make sure at least one network interface is connected to the Internet in order to register your product.


By default, the installer requests a host name from the DHCP server. If you set a custom name in the *Hostname/DNS* tab, make sure the it is unique.

For more information on configuring network connections, refer to https://www.suse.com/documentation/sles-12/book_sle_admin/data/sec_basic-net_yast.html.

Important: Reliable Networking

Please make sure all nodes are on the same network and can communicate without interruptions. If you are using host names to specify nodes, please make sure you have reliable DNS resolution at all times, especially in combination with DHCP.

Kdump


Kdump saves the memory image (“core dump”) to the file system in case the kernel crashes. This enables you to find the cause of the crash by debugging the dump file. See https://www.suse.com/documentation/sles-12/book_sle_tuning/data/cha_tuning_kdump_basic.html  for more information.

Warning: Kdump with large amounts of RAM

If you have a system with large amounts of RAM or a small hard drive, core dumps may not be able to fit on the disk. If the installer warns you about this, there are two options:

1. Enter the *Expert Partitioner* and increase the size of the root partition so that it can accommodate the size of the core dump. In this case, you will need to decrease the size of the data partition accordingly. Remember to keep all other parameters of the partitioning (e.g. the root file system, mount point of data partition) when doing these changes.
2. Disable kdump completely.

System Information

View detailed hardware information by clicking *System Information*. In this screen you can also change *Kernel Settings*. See https://www.suse.com/documentation/sles-12/book_sle_tuning/data/cha_tuning_io.html  for more information.

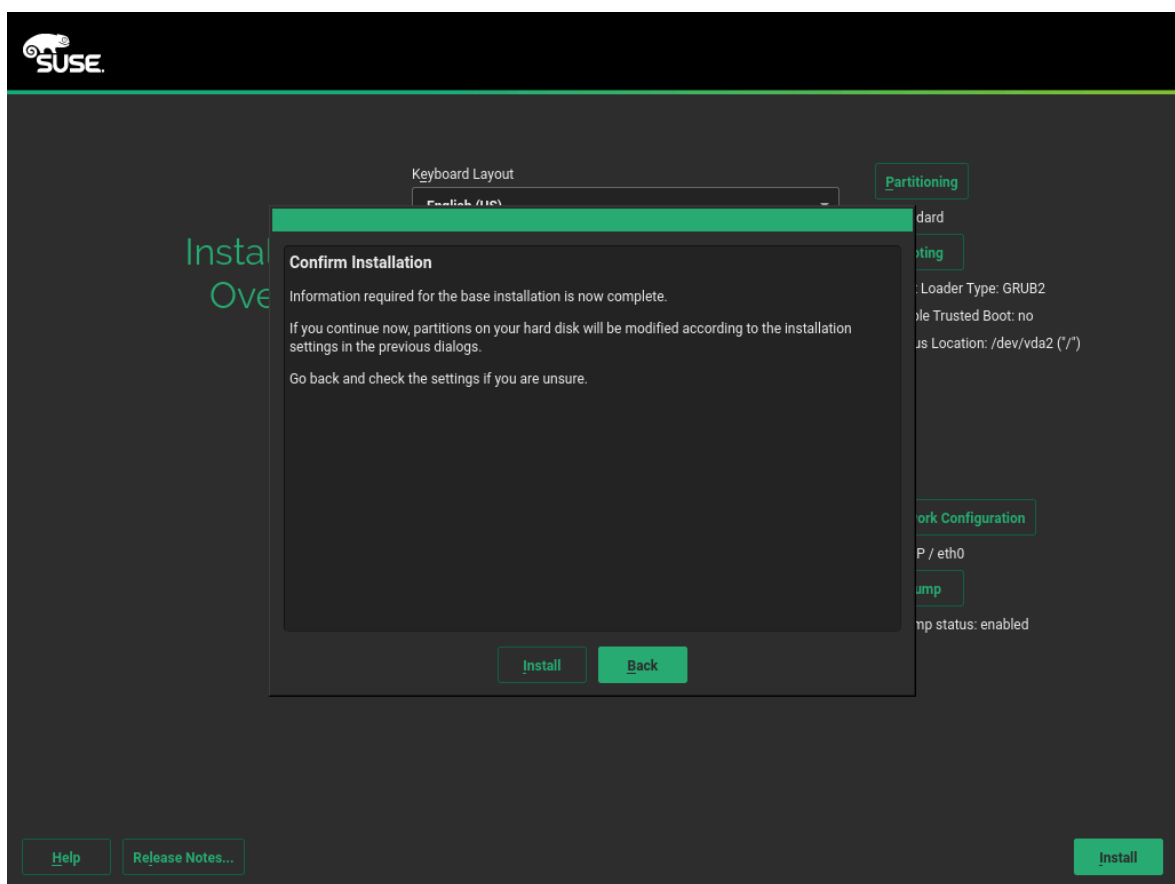
Proceed with *Next*.



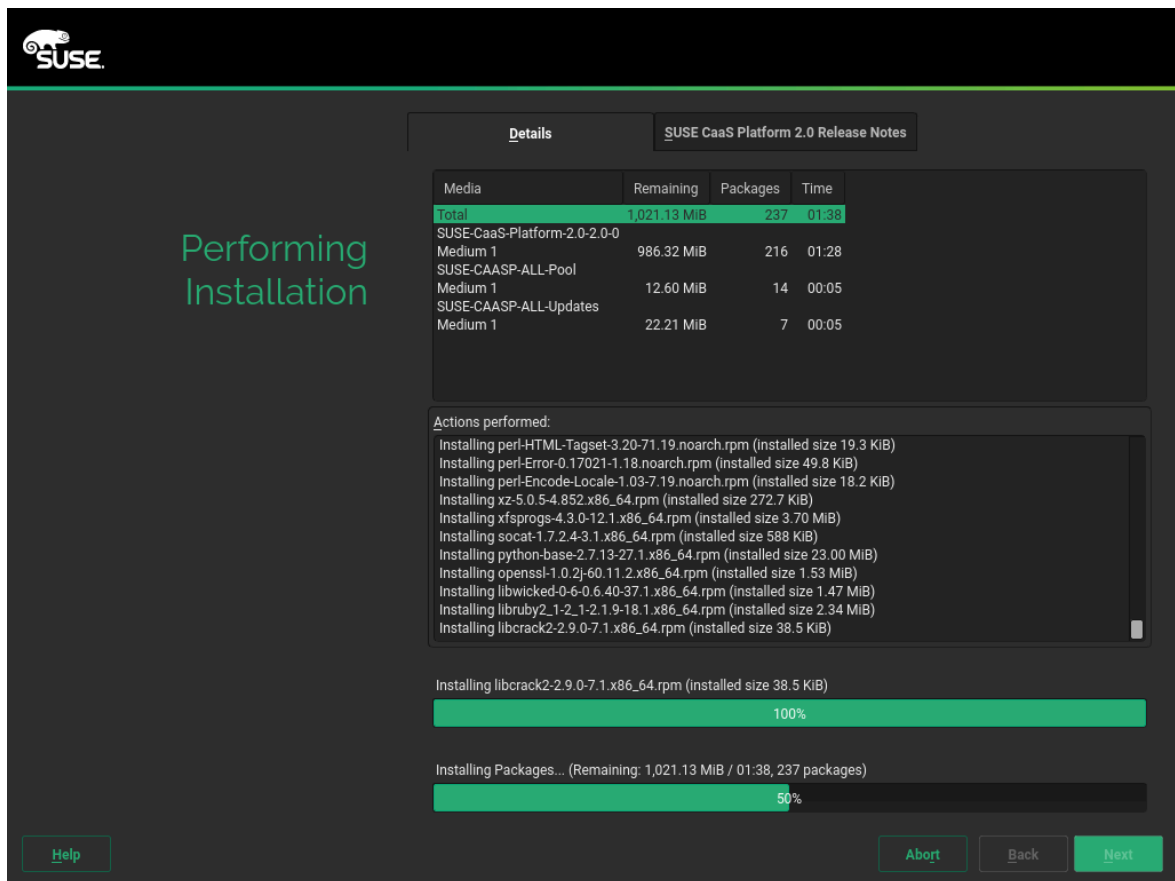
Tip: Installing Product Patches at Installation Time

If SUSE CaaS Platform has been successfully registered at the SUSE Customer Center, you are asked whether to install the latest available online updates during the installation. If choosing *Yes*, the system will be installed with the most current packages without having to apply the updates after installation. Activating this option is recommended.

4. After you have finalized the system configuration on the *Installation Overview* screen, click *Install*. Up to this point no changes have been made to your system. After you click *Install* a second time, the installation process starts.



5. During the installation, the progress is shown in detail on the *Details* tab.



6. After the installation routine has finished, the computer is rebooted into the installed system.

2.1.3 Initial Cluster Configuration

In order to finalize the configuration of the administration node, a few additional steps are required.

1. After the administration node has finished booting and you see the login prompt, point a web browser to <https://caasp-admin.example.com>, where caasp-admin.example.com is the host name or IP address of the administration node. The IP address is shown above the login prompt.
2. To create an Administrator account, click *Create an account* and provide an e-mail address and a password. Confirm the password and click *Create Admin*. You will be logged into the dashboard automatically.

SUSE CaaSP

SUSE CaaS Platform

SUSE CaaS Platform allows you to provision, manage, and scale container-based applications.

It automates your tedious management tasks allowing you to focus on development and writing apps to meet business goals.

I already have an account. [Log in](#)

Sign Up

Enter your email address

Select your password (8 characters min.)

Confirm your password

Create Admin

SUSE® CaaS Platform 1.0.0+dev | © SUSE Linux 2017

3. Fill in the values for *Dashboard Location* and *External Kubernetes API server FQDN*. If necessary, configure the *Proxy Settings*.

Dashboard Location

Host name or IP of the node running this web interface.

External Kubernetes API server FQDN

Fully qualified domain name (FQDN) used to reach the cluster from the outside. In a simple, single-master deployment this will be the FQDN of the node you are about to select as master.

Proxy Settings

If enabled, you can set proxy servers for HTTP and HTTPS. You may also configure exceptions and choose whether to apply the settings only to the container engine or to all processes running on the cluster nodes.

SUSE® CaaS Platform Logout

Initial CaaS Platform Configuration

Generic settings

Dashboard location
caas-admin.example.com i

Overlay network settings Show

Proxy settings Enable Disable

Next

SUSE® CaaS Platform 1.0.0+dev | © SUSE Linux 2017

Click *Next* to proceed and install some cluster nodes as described in [Section 2.1.4, “Installing Cluster Nodes”](#).

2.1.4 Installing Cluster Nodes

Cluster nodes can be installed manually or with AutoYaST. Manual installation is only feasible for a small number of workers. For larger numbers, AutoYaST is recommended.

You can start the setup via PXE. For the full procedure, refer to the SUSE Linux Enterprise 12 Deployment Guide: https://www.suse.com/documentation/sles-12/singlehtml/book_sle_deployment/book_sle_deployment.html#cha.deployment.prep_boot.

You can directly use the `initrd` and `linux` files from your install media, or install the package `tftpboot-installation-CAASP-1.0` on the TFTP server. The package provides the required `initrd` and `linux` files in the `/srv/tftpboot/` directory. You need to modify the paths used in the SUSE Linux Enterprise 12 Deployment Guide to correctly point to the files provided by the package.

Before you can set up a cluster node, you have to install and bootstrap an administration node to run the administration dashboard. Refer to [Section 2.1.2, “Installing the Administration Node”](#) for information on how to install the administration node.

2.1.4.1 Manual Installation

1. Follow the steps as described in [Section 2.1.2, “Installing the Administration Node”](#)
2. In step 3, select Cluster Node as *System Role* and enter the host name or IP address of the *Administration Node*.



Important: Reliable Networking

Please make sure all nodes are on the same network and can communicate without interruptions. If you are using host names to specify nodes, please make sure you have reliable DNS resolution at all times, especially in combination with DHCP.

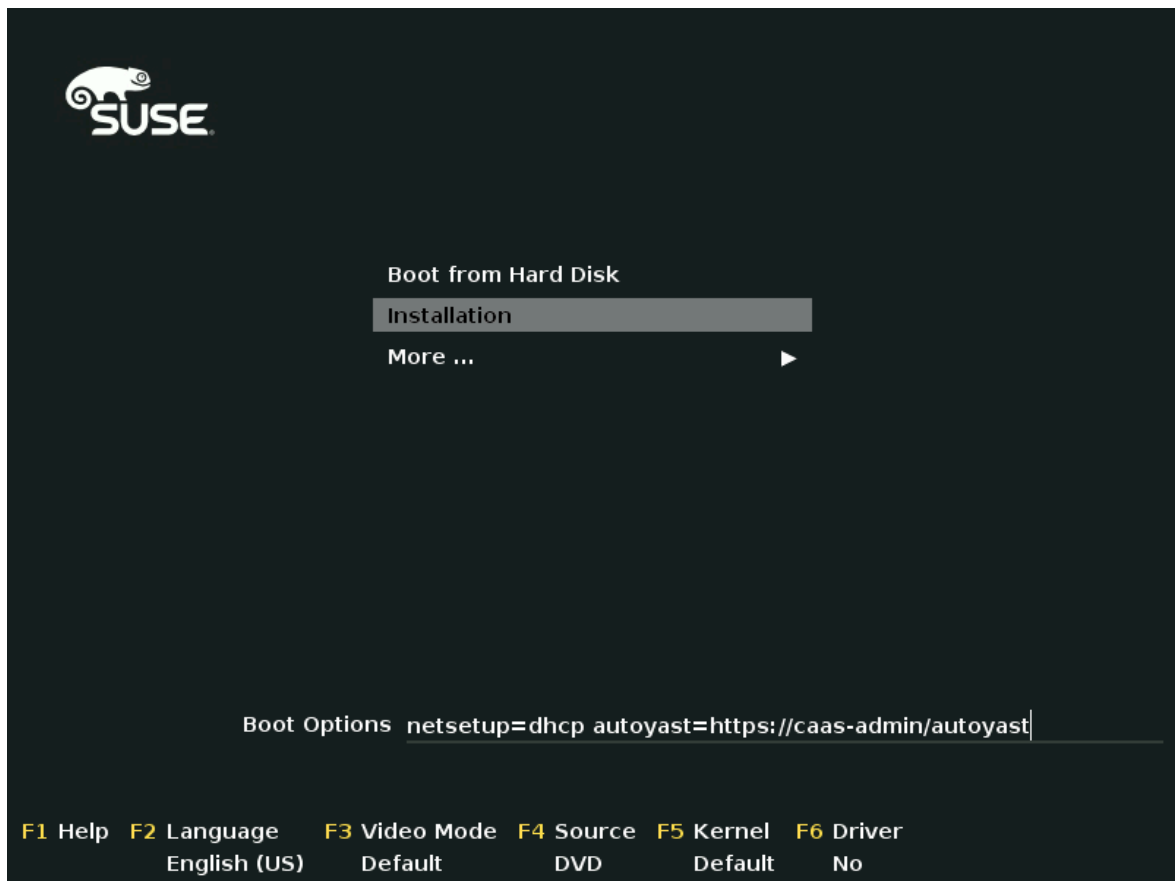
3. After you have finalized the system configuration on the *Installation Overview* screen, click *Install*. Up to this point no changes have been made to your system. After you click *Install* a second time, the installation process starts.
After a reboot, the worker should show up in the dashboard and can be added to your cluster.

2.1.4.2 Installation with AutoYaST

Please make sure you have finished the initial configuration described in [Section 2.1.3, “Initial Cluster Configuration”](#) before installing cluster nodes with AutoYaST.

1. Insert the SUSE CaaS Platform DVD into the drive, then reboot the computer to start the installation program.

2.



Select *Installation* on the boot screen. Enter the necessary *Boot Options* for AutoYaST and networking, namely:

autoyast

Path to the AutoYaST file. For more information, refer to https://www.suse.com/documentation/sles-12/book_autoyast/data/invoking_autoinst.html#commandline_ay ↗

netsetup

Network configuration. If you are using dhcp, you can simply enter dhcp. For manual configurations, refer to https://www.suse.com/documentation/sles-12/book_autoyast/data/ay_adv_network.html ↗.

hostname

The host name for the node, if not provided by DHCP. If you manually specify a host name, make sure it is unique.

Press . This boots the system and loads the SUSE CaaS Platform installer.

3. The rest of the installation will run fully automated. After a reboot, the worker should show up in the dashboard and can be added to your cluster.

2.1.5 Bootstrapping the Cluster

To complete the installation of your SUSE CaaS Platform cluster, it is necessary to bootstrap at least three additional nodes; those will be the Kubernetes master and workers.

1. Point a web browser to <https://caasp-admin.example.com> to open the dashboard, where caasp-admin.example.com is the host name or IP address of the administration node.
2. The dashboard lists all cluster nodes registered at the administration node. Newly installed nodes are be listed as *Pending Nodes*. You can accept individual nodes or all by clicking *Accept All Nodes*.

The screenshot shows the 'Bootstrap Cluster' page in the SUSE CaaS Platform administration interface. The page has a dark blue header with 'SUSE® Container as a Service Platform' and a 'Logout' button. The main content area is light gray and titled 'Bootstrap Cluster'. It displays '3 nodes found' and a message: 'After choosing the master and clicking "Bootstrap cluster" all the other selected nodes will be set to the worker role.' Below this is a table with three columns: 'Id', 'Hostname', and 'Master'. The table lists three nodes, each with a checked checkbox in the 'Id' column and a radio button in the 'Master' column. The second node, 'caasp10node01.kvm', is selected as the master. At the bottom right of the table are 'Back' and 'Bootstrap cluster' buttons. Below the table is a 'Pending Nodes' section with a '✓ Accept All Nodes' button and a message: 'You currently have no nodes to be accepted for bootstrapping.' The footer of the page reads 'SUSE CaaSP® 0.0.0+dev | © SUSE Linux 2017'.

Id	Hostname	Master
<input checked="" type="checkbox"/> 875aa4e096674b419dad54ec3f59fd6f	caasp10node02.kvm	<input type="radio"/>
<input checked="" type="checkbox"/> 2df88d9dc68f43748621317d86dc2500	caasp10node01.kvm	<input checked="" type="radio"/>
<input checked="" type="checkbox"/> aa3bda197bee4b47ba27035d916fb6df	caasp10node03.kvm	<input type="radio"/>

Use the check box in the first column to select the nodes you want to add to your cluster. In the last column, select the one that should become Kubernetes master. All other nodes will be set to the worker role once you click *Bootstrap cluster*.

3. Please wait while the cluster is bootstrapped. Once finished, the status indicator icon of the nodes changes accordingly and you can download the kubect1 configuration file.

SUSE® Container as a Service Platform

Logout

Cluster Status

Summary

Total nodes	3	Updates	Manual
Master nodes	1	# of nodes w/ outdated software	0
New nodes ⓘ	0		

Nodes

⬇️ kubect1 config

Status	ID	Hostname	Role
✔️	875aa4e096674b419dad54ec3f59fd6f	caasp10node02.kvm	<input type="radio"/> worker
✔️	2df88d9dc68f43748621317d86dc2500	caasp10node01.kvm	<input checked="" type="radio"/> master
✔️	aa3bda197bee4b47ba27035d916fb6df	caasp10node03.kvm	<input type="radio"/> worker

Pending Nodes

✔️ Accept All Nodes

You currently have no nodes to be accepted for bootstrapping.

3 Configuration

SUSE CaaS Platform is typically configured in two stages, at first during the installation process and after installation you can configure your cluster by using `cloud-init`. The first stage configuration of SUSE CaaS Platform comes as much preconfigured as possible. The second stage is typically used for large scales clusters, if you have several machines, `cloud-init` is not necessary.

Each configuration stage is described in the following sections.

3.1 Configuration

The defaults for the first stage configuration are the following:

timezone

is set to *UTC* by default, but can be changed by `cloud-init`.

keyboard

is set to *us* by default, but can be changed during the installation process.

locale

is set to *en_US.utf8* by default, but can be changed by `cloud-init`.

3.2 `cloud-init` Configuration

`cloud-init` is used after the installation is complete. You then define a set of configuration files that will be applied during the boot phase.

The `cloud-init` configuration can be loaded from different data sources. Currently there are preconfigured the following datasources and `cloud-init` searches in them for configuration in the following order:

1. the `LocalDisk` datasource
2. the `NoCloud` datasource
3. the `OpenStack` datasource - if there is a running OpenStack service, this datasource will be used. The configuration then depends on a particular setup and thus it is not covered by this manual.

More details about the datasources are provided further.

3.2.1 The LocalDisk Datasource

The `cloud-init` searches for the configuration files: `meta-data`, `user-data` and optionally `vendor-data` in the local directory: `/cloud-init-config`.

3.2.2 The NoCloud Datasource

The `NoCloud` datasource enables you to read the `cloud-init` configuration without running a network service. `cloud-init` searches for the configuration files `meta-data`, `user-data` in the root directory of a local file system formatted as `vfat` or `iso9660` with a label `cidata`. Typically it is an unpartitioned USB stick or disk or a DVD iso.

Alternatively you can specify a remote location of the `cloud.cfg`, but you have to configure network first, e.g. by using local configuration files. The url is specified during boot time and must be in the format: `cloud-init-url=http://hostname.domain/cloud.cfg`. The content of the passed url is copied to `/etc/cloud/cloud.cfg.d/91_kernel_cmdline_url.cfg` and it is not overwritten even though the url changes.

3.2.3 The cloud.cfg Configuration File

The `cloud.cfg` is used to define a datasource and locations of other required configuration files. Use the `#cloud-config` syntax when defining the content.

An example with `NoCloud` datasource follows:

```
#cloud-config
datasource:
  NoCloud:
    # default seedfrom is None
    # if found, then it should contain a url with:
    #   <url>user-data and <url>meta-data
    # seedfrom: http://my.example.com/<path>/
```

3.2.4 The meta-data Configuration File

The file `meta-data` is a YAML formatted file that is intended to configure the system items like network, instance ID, etc. The file typically contains the `instance-id` and `network-interfaces` options, each is described further.

`instance-id`

Defines the instance. If you perform any changes to the configuration (either `user-data` or `meta-data`) you have to update this option with another value. Thus the `cloud-init` recognizes if this the first boot of the particular instantiated host.

```
instance-id: iid-example001
```

`network-interfaces`

Here you can define the following options:

- `auto` to start the network in that configuration automatically during the boot phase.
- `iface` that defines the configured interfaces.

A static network configuration then could look as follows:

```
network-interfaces: |
  auto eth0
  iface eth0 inet static
  address 192.168.1.10
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.1
```

3.2.5 The user-data Configuration File

The configuration file `user-data` is a YAML file used to configure users, SSH keys, time zone, etc. Each part of the file is described in following sections.

3.2.5.1 `user-data` Header

Each `user-data` file must start with `#cloud-config` that indicates the `cloud-config` format. In the snippet below you enable the debugging output and you disable passwordless authentication for `root`. You will have to login with the `root` credentials then.

```
#cloud-config
debug: True
disable_root: False
```

3.2.5.2 `runcmd` Statements

In the `user-data` you can use the `runcmd` statement to run various commands in your system. The `user-data` file can contain only one `runcmd` statement, thus in case you need to run several commands, group them into one statement:

```
runcmd:
- /usr/bin/systemctl enable --now ntpd
```

By using the `runcmd` statement, you can perform the following in your system:

Configure keyboard layout

configure the German keyboard layout with *nodeadkeys*:

```
runcmd:
- /usr/bin/localectl set-keymap de-latin1-nodeadkeys
```

Start services

for example, start the NTP server as described in [Section 3.2.5.9, “NTP Server Configuration”](#).

3.2.5.3 `SSH Keys Management`

You can configure the behaviour of adding SSH keys to the `authorized_keys` and the SSH login pattern.

```
ssh_deletekeys: False
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa XXXKEY mail@example.com
```

The option `ssh_deletekeys` disables/enables automatic deletion of old private and public SSH keys of the host. The default value is `true`—the keys are deleted and new keys are generated. We do not recommend using the default value, as there could be a problem reported by `ssh` that the keys are incorrect or has been changed after the `cloud-init` configuration has been changed.

The option `ssh_pwauth: true` allows you to login by using SSH with a password, if the password is set.

The option `ssh_authorized_keys` defines whether the SSH key will be added to the `authorized_keys` file of the user. If **not** specified otherwise, the default user is `root`.

3.2.5.4 Setting Password

The `user-data` file enables you to set default passwords by using the `chpasswd` option:

```
chpasswd:
  list: |
    root:linux
  expire: True
```

In the example above you set a *linux* password for `root`. The `expire` option defines whether the user will be prompted to change the default password at the first login.

3.2.5.5 Adding Custom Repository

You can add a custom software repository to your system by using the `zypp_repos` option:

```
zypper:
  repos:
    - id: opensuse-oss
      name: os-oss
      baseurl: http://my.example.com/repo/SUSE-CAASP-2-CUSTOM/
      enabled: 1
      autorefresh: 1
    - id: opensuse-oss-update
      name: os-oss-up
      baseurl: http://my.example.com/repo/SUSE-CAASP-2-CUSTOM/update
```

The options available are:

id

The local unique ID of the repository, also known as its alias. (Mandatory.)

name

A more descriptive string describing the repository, used in the UI. (Mandatory.)

baseurl

URL to the directory where the repository's repodata directory lives. (Mandatory.)

type

Zypper is able to work with three types of repository: yast2 and rpm-md (yum) repositories, as well as plaintext - plain directories containing .rpm files.

path

This is relative to the baseurl; the default is /.

gpgcheck

Defines whether the source signatures should be checked using GPG.

gpgkey

Defines the URL for a GPG key.

enabled

Defaults to 1 (on). Set to 0 to disable the repository: it will be known and listed, but not used.

autorefresh

Defaults to 1 (on). When on, the local package cache will be updated to the remote version whenever package management actions are performed.

priority

Defines a source priority, from 1 (lowest) to 200 (highest). The default is 99.

3.2.5.6 Setting Timezone

You can set a default timezone. Bear in mind that the configured value must exist in /usr/share/zoneinfo:

```
timezone: Europe/Berlin
```

3.2.5.7 Setting Host name

You can set either a host name or, preferably, a fully-qualified domain name for the machine:

```
hostname: myhost
```

or

```
fqdn: myhost.example.com
```

The option `preserve_hostname` specifies whether any existing host name should be retained or not. Enter `true` or `false` as required:

```
preserve_hostname: true
```

3.2.5.8 Configuring Name server

You can configure the server to manage the `resolv.conf` file and thus set values of the file:

```
manage_resolv_conf: true
resolv_conf:
  nameservers: ['8.8.4.4', '8.8.8.8']
  searchdomains:
    - foo.example.com
    - bar.example.com
  domain: example.com
  options:
    rotate: true
    timeout: 1
```

3.2.5.9 NTP Server Configuration

You can also configure the NTP server. The following snippet configures three NTP servers during the first boot and the NTP service is enabled and started:

```
ntp:
  servers:
    - ntp1.example.com
    - ntp2.example.com
    - ntp3.example.com
  runcmd:
    - /usr/bin/systemctl enable --now ntpd
```

3.2.5.10 Salt minion Configuration

You can use the file to set the Salt minion and its communication with the Salt master.

```
salt_minion:
```

```
conf:
  master: saltmaster.example.com

public_key: |
  -----BEGIN PUBLIC KEY-----
  XXX
  -----END PUBLIC KEY-----

private_key: |
  -----BEGIN RSA PRIVATE KEY-----
  XXX
  -----END RSA PRIVATE KEY-----
```

3.2.5.11 Assigning Roles to the Cluster Nodes

You need to specify which node of your cluster will be used as the administration node and which nodes will be used as regular cluster nodes.

To assign the administration node role to the cluster node, add the following to the configuration file:

```
suse_caasp:
  role: admin
```

If the cluster node is assigned the administration node, all required containers are imported and started. Bear in mind, that an NTP server must be configured on that machine.

To other cluster nodes you assign the role `cluster`. The machine will register itself as Salt minion on the administration node and configure a timesync service with administration node as a reference. You do not have to install any NTP server, but if you need to use one, you need to disable the `systemd-timesyncd` first. An example of the `cluster` role assignment follows:

```
suse_caasp:
  role: cluster
  admin_node: admin.example.com
```

where the `admin.example.com` is the host name of the administration node.

3.2.6 The vendor-data Configuration File

The `vendor-data` is an optional configuration file that typically stores data related to the cloud you use. The data are provided by the entity that launches the cloud instance.

The format is the same as used for user-data.

4 Cluster Administration

4.1 Authentication and Authorization

Role-based access control (RBAC) adds the ability to perform authentication and authorization of activities performed against a Kubernetes cluster. Authentication is concerned with the “who” and authorization is concerned with the “what”.

4.1.1 Authentication

Starting in SUSE CaaS Platform 2, **kubectl** needs to authenticate against the Kubernetes master node. The necessary authentication information is included in the `kubeconfig` file available from Velum. Click the `kubectl config` button and authenticate with your user name and password. Download the file from Velum and save it as `$HOME/.kube/config`.



Tip: The KUBECONFIG variable

kubectl uses an environment variable named `KUBECONFIG` to locate your `kubeconfig` file. If this variable is not specified, it defaults to `$HOME/.kube/config`. To use a different location, run

```
tux > export KUBECONFIG=/path/to/kube/config/file
```



Note: Obtaining the root CA certificate

You can obtain the root CA certificate from any node in your cluster via SCP:

```
tux > scp NODE:/etc/pki/trust/anchors/SUSE_CaaSP_CA.crt .
```

To trust this root CA certificate on your machine, place it in `/etc/pki/trust/anchors/` and call the `update-ca-certificates` script.

4.1.2 Managing Users and Groups

User information is stored in OpenLDAP running in a container on your SUSE CaaS Platform administration node. You can use standard LDAP administration tools for managing these users remotely. To do so, install the `openldap2` package on a computer in your network and make sure that computer can connect to the administration node on port 389. For further information, refer to [Section 2.1.1.3, “Network Requirements”](#).

4.1.2.1 Obtaining the OpenLDAP Password

Before performing any administrative tasks on the OpenLDAP instance, you will need to retrieve your OpenLDAP administrator account password. To do this, run:

```
tux > ssh root@caasp-admin.example.com \  
cat /var/lib/misc/infra-secrets/openldap-password
```

Make sure to replace `caasp-admin.example.com` with the FQDN or IP of your administration node.

4.1.2.2 Adding New Users

By default, when you create the first user in Velum during bootstrap of your cluster, that user is granted `Cluster Administrator` privileges within Kubernetes. You can add additional users with these rights by adding new entries into the LDAP directory.

To add a new user, create a LDIF file like this:

```
dn: uid=userid ❶,ou=People,dc=infra,dc=caasp,dc=local  
objectClass: person  
objectClass: inetOrgPerson  
objectClass: top  
uid: userid ❶  
userPassword: password hash ❷  
givenname: first name ❸  
sn: surname ❹  
cn: full name ❺  
mail: email address ❻
```

Make sure to replace all the parameters indicated *like this* in the template above as follows:

- ❶ User ID (UID) of the new user. Needs to be unique.
- ❷ The user's hashed password. Use `/usr/sbin/slappasswd` to generate the hash.

- ③ The user's first name
- ④ The user's last name
- ⑤ The user's full name
- ⑥ The user's e-mail address. It is used as the login name to Velum and Kubernetes.

Populate your OpenLDAP server with this LDIF file:

```
root # ldapadd -H ldap://ADMIN NODE IP;:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -
w LDAP ADMIN PASSWORD -f LDIF FILE
```

To add this new user to the existing Administrators group, create a new LDIF file like this:

```
dn: cn=Administrators,ou=Groups,dc=infra,dc=caasp,dc=local
changetype: modify
add: uniqueMember
uniqueMember: uid=userid①,ou=People,dc=infra,dc=caasp,dc=local
```

Make sure to replace all the parameters indicated like this in the template above as follows:

- ① The user ID (UID) of the user

Populate your OpenLDAP server with the LDIF file:

```
root # ldapadd -H ldap://ADMIN NODE IP:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -
w LDAP ADMIN PASSWORD -f LDIF FILE
```

4.1.2.3 Changing a User's Password

To change a user's password, create a LDIF file like this:

```
dn: uid=userid①,ou=People,dc=infra,dc=caasp,dc=local
changetype: modify
modify: userPassword
userPassword: password hash ②
```

Make sure to replace all the parameters indicated like this in the template above as follows:

- ① User ID (UID) of the user.
- ② The user's new hashed password. Use /usr/sbin/slappasswd to generate the hash.

Populate your OpenLDAP server with this LDIF file:

```
root # ldapadd -H ldap://ADMIN NODE IP;:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -
w LDAP ADMIN PASSWORD -f LDIF FILE
```


4.1.2.4 Adding New Groups

Say you have users that you want to grant access to manage a single namespace in Kubernetes. To do this, first create your users as mentioned in [Section 4.1.2.2, “Adding New Users”](#). Then create a new group:

```
dn: cn=group name ❶,ou=Groups,dc=infra,dc=caasp,dc=local
objectclass: top
objectclass: groupOfUniqueNames
cn: group name ❶
uniqueMember: uid=member1, ❷ou=People,dc=infra,dc=caasp,dc=local
uniqueMember: uid=member2, ❷ou=People,dc=infra,dc=caasp,dc=local
uniqueMember: uid=member3, ❷ou=People,dc=infra,dc=caasp,dc=local
```

Make sure to replace all the parameters indicated like this in the template above as follows:

- ❶ The group's name.
- ❷ Members of the group. Repeat the `uniqueMember` attribute for every member of this group.

Populate your OpenLDAP server with the LDIF file:

```
root # ldapadd -H ldap://ADMIN NODE IP:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -
w LDAP ADMIN PASSWORD -f LDIF FILE
```

Next, create a role binding to allow this new LDAP group access in Kubernetes. Create a Kubernetes deployment descriptor like this:

```
# Define a Role and its permissions in Kubernetes
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: role name ❶
  namespace: applicable namespace ❷
# This set of rules amounts to "allow all"
rules:
- apiGroups: [""]
  resources: [""]
  resourceNames: [""]
  verbs: [""]
---
# Map an LDAP group to this Kubernetes role
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: role binding name ❸
  namespace: applicable namespace ❷
```

```
subjects:
- kind: Group
  name: LDAP group name ④
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: role name ①
  apiGroup: rbac.authorization.k8s.io
```

- ① Name of the new role in Kubernetes
- ② Namespace the new group should be allowed to access. Use `default` for Kubernetes' default namespace.
- ③ Name of the role binding in Kubernetes
- ④ Name of the corresponding group in LDAP

Add this role and binding to Kubernetes:

```
root # kubectl apply -f DEPLOYMENT DESCRIPTOR FILE
```

4.1.2.5 Further information

For more details on authorization in Kubernetes, refer to <https://kubernetes.io/docs/admin/authorization/rbac/> ↗

4.2 Handling Transactional Updates

For security and stability reasons, the operating system and application should always be up-to-date. While with a single machine you can keep the system up-to-date quite easily by running several commands, in a large-scaled cluster the update process can become a real burden. Thus transactional automatic updates have been introduced. Transactional updates can be characterized as follows:

- They are atomic.
- They do not influence the running system.
- They can be rolled back.
- The system needs to be rebooted to activate the changes.

Transactional updates are managed by the **transactional-update** script, which is called once a day. The script checks if any update is available. If there is an update to be applied, a new snapshot of the root file system is created and the system is updated by using **zypper dup**. All updates released to this point are applied. The snapshot is then marked as active and will be used after the next reboot of the system. Ensure that the cluster is rebooted as soon as possible after the update installation is complete, otherwise all changes will be lost.



Note: General Notes to the Updates Installation

Only packages that are part of the snapshot of the root file system can be updated. If packages contain files that are not part of the snapshot, the update could fail or break the system.

RPMs that require a license to be accepted cannot be updated.

4.2.1 Manual Installation of Updates

After the **transactional-update** script has run on all nodes, Velum displays any nodes in your cluster running outdated software. Velum then enables you to update your cluster directly. Follow the next procedure to update your cluster.

PROCEDURE 4.1: UPDATING THE CLUSTER WITH VELUM

1. Login to Velum.
2. If required, click *UPDATE ADMIN NODE* to start the update.

Cluster Status

Summary



Total nodes 4

Master nodes 1

New nodes ⓘ 0

! Admin node is running outdated software

Nodes

Status	ID
 Update Available	9c783785923b4d0e9be5
 Update Available	ac9f9b7a09994e2681cca
 Update Available	614c9115c0e848918668
 Update Available	45bd5719e8184bb8b1cfc

Pending Nodes

3. Confirm the update by clicking *Reboot to update*.

The screenshot shows the SUSE Container as a Service Platform interface. A modal dialog box is open, titled "The admin node needs to reboot in order to apply the software update". The dialog contains the following text:

Rebooting the admin node will break ongoing operations like creation of the cluster, the addition or removal of nodes and upgrades.

Please ensure none of the following operations are ongoing before proceeding with the upgrade of the admin node.

Once the admin node is rebooted, the page will be refreshed.

At the bottom of the dialog are two buttons: "Cancel" and "Reboot to update".

In the background, the "Cluster Status" section is visible. It includes a "Summary" table:

Summary	
Total nodes	4
Master nodes	1
New nodes ⓘ	0

Below the summary is a message: "Admin node is running outdated software".

The "Nodes" section is also visible, showing a table with the following data:


Status	ID
Update Available	9c783785923b4d0e9be5880902a79202
Update Available	ac9f9b7a09994e2681ccab753ef0f0fa
Update Available	614c9115c0e84891866888edb3ff5f51
Update Available	45bd5719e8184bb8b1cfc085f4c9b230

The "Pending Nodes" section is also visible, with the text: "You currently have no nodes to be accepted for bootstrapping."

4. Now you have to wait until the administration node reboots and Velum is available again.

5. Click *update all nodes* to update master node and worker nodes.

SUSE® Container as a Service Platform





 Signed in successfully.

Cluster Status

Summary

Total nodes	4	Updates
Master nodes	1	# of nodes w/ outda
New nodes ⓘ	0	

Nodes

Status	ID
 Update Available	9c783785923b4d0e9be5880902a79202
 Update Available	ac9f9b7a09994e2681ccab753ef0f0fa
 Update Available	614c9115c0e84891866888edb3ff5f51
 Update Available	45bd5719e8184bb8b1cfc085f4c9b230

Pending Nodes

You currently have no nodes to be accepted for bootstrapping.

4.2.2 Disabling Transactional Updates

Even though it is not recommended, you can disable transactional updates by issuing the command:

```
systemctl --now disable transactional-update.timer
```

Automatic reboot can also be disabled; use either of the commands:

```
systemctl --now disable rebootmgr
```

or

```
rebootmgrctl set-strategy off
```

4.3 Handling Program Temporary Fixes

Program temporary fixes (PTFs) are available in the SUSE CaaS Platform environment. You install them by using the **transactional-update** script. Typically you invoke the installation of PTFs by running:

```
transactional-update reboot ptf install rpm ... rpm
```

The command installs PTF RPMs. The **reboot** option then schedules a reboot after the installation. PTFs are activate only after rebooting of your system.



Note: Reboot Required

If you install or remove PTFs and you call the **transactional-update** to update the system before reboot, the applied changes by PTFs are lost and need to be done again after reboot.

In case you need to remove the installed PTFs, use the following command:

```
transactional-update reboot ptf remove rpm ... rpm
```

4.4 Commands for Single Node Management

SUSE CaaS Platform comes with several built-in commands that enable you to manage your cluster nodes.

4.4.1 The **issue-generator** Command

The **issue-generator** creates a volatile temporary `/run/issue` file. The file `/etc/issue` should be a symbolic link to the temporary `/run/issue`.

You can use the command to prefix all directories and files with a specified prefix (path in this case):

```
issue-generator --prefix path
```

By using the command you can also create or delete files in the network configuration, for example:

```
issue-generator network remove interface
```

The command removes file `/run/issue.d/70-interface.conf`. The file contains the name of the `interface` and escape codes for **agentty**.

You can use the command to add or delete `/run/issue.d/60-ssh_host_keys.conf` that contains fingerprints of the public SSH keys of the host:

```
issue-generator ssh add|remove
```



Note: The Command without Arguments

If you run the command without any argument, all input files will be applied.

4.4.2 The **transactional-update** Command

The **transactional-update** enables you to install or remove updates of your system in an atomic way. The updates are applied all or none of them if any package cannot be installed. Before the update is applied, a snapshot of the system is created in order to restore the previous state in case of a failure.

If the current root file system is identical to the active root file system (after applying updates and reboot), run cleanup of all old snapshots:

```
transactional-update cleanup
```

Other options of the command are the following:

up

If there are new updates available, a new snapshot is created and **zypper dup** is used to update the snapshot. The snapshot is activated afterwards and is used as the new root file system after reboot.

```
transactional-update up
```

dup

If there are new updates available, a new snapshot is created and **zypper dup --no-all-low-vendor-change** is used to update the snapshot. The snapshot is activated afterwards and is used as the new root file system after reboot.

```
transactional-update dup
```

patch

If there are new updates available, a new snapshot is created and **zypper patch** is used to update the snapshot. The snapshot is activated afterwards and is used as the new root file system after reboot.

```
transactional-update patch
```

ptf install

The command installs the specified RPMs:

```
transactional-update ptf install rpm ... rpm
```

ptf remove

The command removes the specified RPMs from the system:

```
transactional-update ptf remove rpm ... rpm
```

rollback

The command sets the default sub volume. On systems with read-write file system **snapper rollback** is called. On a read-only file system and without any argument, the current system is set to a new default root file system. If you specify a number, that snapshot is used as the default root file system. On a read-only file system, no additional snapshots are created.

```
transactional-update rollback snapshot_number
```

--help

The option outputs possible options and subcommands.

```
transactional-update --help
```

4.4.3 The `create_autoyast_profile` Command

The `create_autoyast_profile` command creates an autoyast profile for fully automatic installation of SUSE CaaS Platform. You can use the following options when invoking the command:

`-o | --output`

Specify to which file the command should save the created profile.

```
create_autoyast_profile -o filename
```

`--salt-master`

Specify the host name of the Salt master.

```
create_autoyast_profile --salt-master saltmaster
```

`--smt-url`

Specify the URL of the SMT server.

```
create_autoyast_profile --smt-url saltmaster
```

`--regcode`

Specify the registration code for SUSE CaaS Platform.

```
create_autoyast_profile --regcode 405XAbs593
```

`--reg-email`

Specify an e-mail address for registration.

```
create_autoyast_profile --reg-email address@example.com
```

4.5 Adding Nodes

After you complete the deployment and you bootstrap the cluster, you may need to perform additional changes to the cluster. By using Velum you can add additional worker nodes to your cluster. The following steps guides you through that procedure:

PROCEDURE 4.2: ADDING NODES TO EXISTING CLUSTER

1. Prepare the node as described in [Section 2.1.4, "Installing Cluster Nodes"](#)




2. Open Velum in your browser and login.
3. You should see the newly added node as a node to be accepted in *Pending Nodes*. Accept the node.

Cluster Status

Summary

Total nodes	3
Master nodes	1
New nodes ⓘ	0

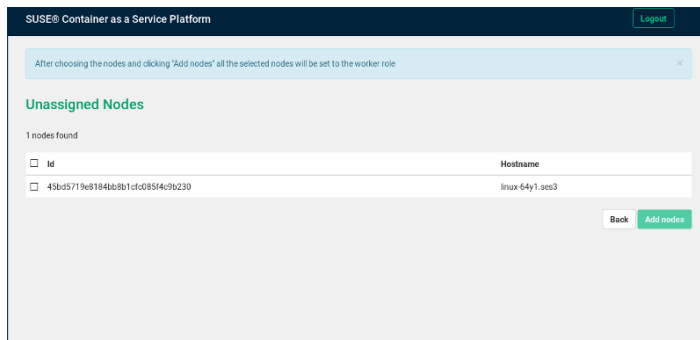
Nodes

Status	ID
	3eb6b007fd3a4248ba96e7597b965e51
	f7d3d0451e0d4f1498f99603a0699222
	509fec04b5ad40f197d9128269779b70

Pending Nodes

ID
13a1a195c0054de98ae344941f2d7de5

4. In the *Summary* you can see the *New* that appears next to *New nodes*. Click the *New* button.



5. Select the node to be added and click *Add nodes*.
6. The node has been added to your cluster.

4.6 Cluster Monitoring

There are three basic ways how you can monitor your cluster:

- by directly accessing the *cAdvisor* on http://WORKER_NODE_ADDRESS:4194/containers/. The *cAdvisor* runs on worker nodes by default.
- By using *Heapster*, for details refer to [Section 4.6.1, "Monitoring with Heapster"](#).
- By using *Grafana*, for details refer to [Section 4.6.2, "Monitoring with Grafana"](#).

4.6.1 Monitoring with Heapster

Heapster is a tool that collects and interprets various signals from your cluster. *Heapster* communicates directly with the *cAdvisor*. The signals from the cluster are then exported using REST endpoints.

To deploy *Heapster*, run the following command:

```
kubectl apply -f https://raw.githubusercontent.com/SUSE/caasp-services/master/contrib/addons/heapster/heapster.yaml
```

Heapster can store data in *InfluxDB*, which can be then used by other tools.

4.6.2 Monitoring with Grafana

Grafana is an analytics platform that processes data stored in *InfluxDB* and displays the data graphically. You can deploy *Grafana* by running the following commands:

```
kubectl apply -f https://raw.githubusercontent.com/SUSE/caasp-services/master/contrib/addons/heapster/heapster.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes/heapster/master/deploy/kube-config/influxdb/influxdb.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes/heapster/release-1.3/deploy/kube-config/influxdb/grafana-deployment.yaml
wget https://raw.githubusercontent.com/kubernetes/heapster/release-1.3/deploy/kube-config/influxdb/grafana-service.yaml
```

Then open the file grafana-service.yaml:

```
vi grafana-service.yaml
```

In the file uncomment the line with the NodePort type.

To finish the *Grafana* installation, apply the configuration by running:

```
kubectl apply -f grafana-service.yaml
```

5 Software Installation

Once your cluster is ready, you may want to deploy additional software that is not installed on SUSE CaaS Platform by default. This chapter provides instructions on how to install and configure various tools and services.

5.1 Deploying Helm and Tiller

Helm charts provide a standard way to deploy Kubernetes applications. While Tiller (the Helm's server-side component) is not installed on SUSE CaaS Platform by default, it can be installed by enabling the appropriate option in the Velum graphical interface. When the *Install Tiller* option is enabled, Tiller is installed as a container on a SUSE CaaS Platform 2 cluster from the installation media.

Once Tiller is installed on SUSE CaaS Platform 2 cluster, the Helm client can be used to interact with Tiller and the cluster. To initialize the Helm client (which automatically discovers Tiller in the cluster), run the `helm init --client-only` command. You can then use the Helm client as described in the official documentation at https://docs.helm.sh/using_helm/#using-helm.



Note: Helm and Tiller are Optional

Both Helm and Tiller are optional, unless you plan to run a SUSE Cloud Application Platform.

Tiller that ships with SUSE CaaS Platform is supported by SUSE. While SUSE does not provide support for third-party Helm charts, you can easily use them if necessary.

5.2 Installing Kubernetes Dashboard



Important: Technology Preview

Even though you can install and use the Kubernetes dashboard, SUSE CaaS Platform currently fully supports only Velum.

To install the Kubernetes dashboard, perform the following steps:

PROCEDURE 5.1: INSTALLATION OF KUBERNETES DASHBOARD

1. Connect to the administration node using SSH.

2. Run the command:

```
kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.6.3/src/
deploy/kubernetes-dashboard.yaml
```

3. After the previous command finishes, run the following to expose the proxy:

```
kubectl proxy
Starting to serve on 127.0.0.1:8001
```

4. In your browser open *http://127.0.0.1:8001/api/v1/proxy/namespaces/kube-system/services/kubernetes-dashboard/#!/workload?namespace=default*. The Kubernetes dashboard is running there.

6 SUSE CaaS Platform Integration with SES

SUSE CaaS Platform can use SUSE's another product as storage for containers—SUSE Enterprise Storage (further called SES). This chapter gives details how to integrate these two products in several ways.

6.1 Prerequisites

Before you start the integration process, you need to ensure the following:

- SUSE CaaS Platform can access the SES cluster.
- SUSE CaaS Platform can communicate with the SES nodes—master, monitoring nodes, OSD nodes and the meta data server in case you need a shared file system. For more details regarding SES refer to [SES documentation \(https://www.suse.com/documentation/ses-4/book_storage_admin/data/book_storage_admin.html\)](https://www.suse.com/documentation/ses-4/book_storage_admin/data/book_storage_admin.html).

6.2 Mounting a Named Object Storage to a Container

The procedure below describes steps to take when you need to mount a fixed name storage volume to a particular container.

PROCEDURE 6.1: MOUNTING STORAGE TO A CONTAINER

1. On the master node apply the configuration that includes Ceph secret by using the **kubectl apply**

```
kubectl apply -f - << *EOF*
apiVersion: v1
kind: Secret
metadata:
  name: ceph-secret
type: "kubernetes.io/rbd"
data:
  key: "the Ceph secret key"
*EOF*
```

The Ceph secret key is stored on the SES master node in the file /etc/ceph/ceph.client-admin.keyring – use the key value.

2. Create an image in the SES cluster. On the master node, run the following command:

```
rbd create -s 1G yourvolume
```

Where 1G is the size of the image and yourvolume is the name of the image.

3. Create a pod that uses the image. On the master node run the following:

```
kubect1 apply -f - << *EOF*
apiVersion: v1
kind: Pod
metadata:
  name: busybox-rbd
  labels:
    app: busybox-rbd
spec:
  containers:
    - name: busybox-rbd-container
      image: busybox
      command: ['sh', '-c', 'echo The app is running! && sleep 3600']
      volumeMounts:
        - mountPath: /mnt/rbdvol
          name: rbdvol
  volumes:
    - name: rbdvol
      rbd:
        monitors:
          - [monitor1 ip:port]
          - [monitor2 ip:port]
          - [monitor3 ip:port]
          - [monitor4 ip:port]
        pool: rbd
        image: yourvolume
        user: admin
        secretRef:
          name: ceph-secret
        fsType: ext4
        readOnly: false
*EOF*
```

4. Verify that the pod exists and its status:

```
kubect1 get po
```

5. Once the pod is running, check the mounted volume:

```
kubectl exec -it busybox-rbd -- df -k

...
/dev/rbd1          999320      1284    929224    0% /mnt/rbdvol
...
```

In case you need to delete the pod, run the following command on master node:

```
kubectl delete pod busybox-rbd
```

6.3 Creating Pods with Persistent Volumes

The following procedure describes how to attach a pod to a persistent SES volume.

PROCEDURE 6.2: CREATING A POD WITH PERSISTENT VOLUME AND PERSISTENT VOLUME CLAIMS

1. Create a volume on the SES cluster:

```
rbd create -s 1G yourvolume
```

Where 1G is the size of the image and yourvolume is the name of the image.

2. Create the persistent volume on the master node:

```
kubectl apply -f - << *EOF*
apiVersion: v1
kind: PersistentVolume
metadata:
  name: yourpv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  rbd:
    monitors:
      - [monitor1 ip:port]
      - [monitor2 ip:port]
      - [monitor3 ip:port]
      - [monitor4 ip:port]
    pool: rbd
    image: yourvolume
```

```
user: admin
secretRef:
  name: ceph-secret
fsType: ext4
readOnly: false
*EOF*
```

3. Create a persistent volume claim on the master node:

```
kubectl apply -f - << *EOF*
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: yourpvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
*EOF*
```



Note: Listing Volumes

This persistent volume claim does not explicitly list the volume. Persistent volume claims work by picking any volume that meets the criteria from a pool. In this case we specified any volume with a size of 1G or larger. When the claim is removed the recycling policy will be followed.

4. Create a pod that uses the persistent volume claim. On the master node run the following:

```
kubectl apply -f - <<*EOF*
apiVersion: v1
kind: Pod
metadata:
  name: busybox-rbd
  labels:
    app: busybox-rbd
spec:
  containers:
    - name: busybox-rbd-container
      image: busybox
      command: ['sh', '-c', 'echo The app is running! && sleep 3600']
      volumeMounts:
        - mountPath: /mnt/rbdvol
```

```
name: rbdvol
volumes:
- name: rbdvol
  persistentVolumeClaim:
    claimName: yourpvc
*EOF*
```

5. Verify that the pod exists and its status. On the master node run:

```
kubectl get po
```

6. Once pod is running, check the volume by running on the master node:

```
kubectl exec -it busybox-rbd -- df -k

...
/dev/rbd3          999320      1284    929224    0% /mnt/rbdvol
...
```

In case you need to delete the pod, run the following command on the master node:

```
kubectl delete pod busybox-rbd
```

And when the command finishes, run

```
kubectl delete persistentvolume yourpv
```



Note: Deleting a Pod

When you delete the pod, the persistent volume claim is deleted as well.

7 Upgrading SUSE CaaS Platform

As SUSE CaaS Platform is constantly developed and improved, new versions get released. You are strongly advised to upgrade to a supported release. These upgrades may involve manual intervention.



Important: Service Window Required

Upgrades may take some time, during which services may be degraded in performance or completely unavailable. Please make sure to plan a service window.

PROCEDURE 7.1: GENERAL UPGRADE PROCEDURE

1. Upgrade the administration node.
2. Manually perform additional upgrade steps of the administration node. These steps are version-specific and described in the following chapters.
3. Upgrade the cluster nodes through Velum.

7.1 Upgrading from SUSE CaaS Platform 1

The following sections contain the necessary steps to upgrade from SUSE CaaS Platform 1 to 2 or later.

7.1.1 Migrating Users

SUSE CaaS Platform 2 comes with Role-Based Access Control (RBAC), which stores user information in OpenLDAP. Therefore, after upgrading from SUSE CaaS Platform 1 to version 2 or higher, you have to migrate existing Velum users to OpenLDAP users.

For more information about RBAC and user management, refer to [Section 4.1, “Authentication and Authorization”](#).

PROCEDURE 7.2: MIGRATE USERS FROM VERSION 1 TO 2

1. Connect to the administration node using SSH.

2. Open a shell in the Velum container:

```
root # docker exec -it $(docker ps | grep dashboard | awk '{print $1}') bash
```

3. Inside the container, execute the following command:

```
bash-4.3 # entrypoint.sh bundle exec rake velum:migrate_users
```

Once the command successfully finishes, existing user accounts will be available for logging into Velum again.

4. Type **exit** or press **Ctrl-D** to exit the Velum container.

7.1.2 Upgrading etcd

SUSE CaaS Platform 2 comes with Kubernetes 1.7, which uses etcd version 3 as default storage backend. Therefore, after upgrading from SUSE CaaS Platform 1 to version 2 or higher, you have to orchestrate the migration between etcd 2 and 3.

Important: Service Window Required

This migration can take a several minutes, during which etcd and kube-api services are unavailable. Please make sure to plan a service window.

PROCEDURE 7.3: MIGRATE FROM ETCD VERSION 2 TO 3

1. Connect to the administration node using SSH.
2. Open a shell in the Salt master container:

```
root # docker exec -it $(docker ps | grep salt-master | awk '{print $1}') bash
```

3. Inside the container, execute the following command:

```
bash-4.3 # salt-run state.orchestrate orch.etcd-migrate
```

The orchestration will shutdown all etcd and kube-apiserver services, perform the etcd migration steps, set the “etcd_version = etcd3” pillar value, and restart etcd and kube-api services.

Once the command successfully finishes, all services will be available again.

4. Type **exit** or press **Ctrl-D** to exit the Salt master container.

7.1.3 Adding new settings

Run the following commands to ensure that default values are set correctly for some new options introduced in SUSE CaaS Platform 2 which were not present in version 1.

PROCEDURE 7.4: ADD NEW SETTINGS INTRODUCED IN SUSE CAAS PLATFORM 2

1. Connect to the administration node using SSH.
2. Open a shell in the Velum container:

```
root # docker exec -it $(docker ps | grep dashboard | awk '{print $1}') bash
```

3. Set `dashboard_external_fqdn` to the Fully Qualified Domain Name (FQDN) of the administration node:

```
bash-4.3 # entrypoint.sh bundle exec rails runner \  
'Pillar.create(pillar: "dashboard_external_fqdn", value: "FQDN")'
```

Replace `FQDN` with the Fully Qualified Domain Name of your administration node.

4. Create the LDAP related pillars:

```
bash-4.3 # entrypoint.sh bundle exec rails runner \  
'Velum::LDAP.ldap_pillar_settings!({}).each \  
{|key, value| Pillar.create(pillar: Pillar.all_pillars[key.to_sym], \  
value: value)}'
```

5. If you intend to use Helm on your CaaS Cluster, you also need to enable Tiller (Helm's server component). Execute the following command in the open shell:

```
bash-4.3 # entrypoint.sh bundle exec rails runner \  
'Pillar.create(pillar: "addons:tiller", value: "true")'
```

6. Type **exit** or press **Ctrl-D** to exit the Velum container.

7.1.4 Generating the Service Account Key File on the CA

Kubernetes distinguishes between user and service accounts. While user accounts are for humans, service accounts are for processes, which run in pods.

In order to use service accounts, you have to generate the service account key file `sa.key` on the Certificate Authority (CA).

PROCEDURE 7.5: GENERATE THE SERVICE ACCOUNT KEY FILE `sa.key` ON THE CA.

1. Connect to the administration node using SSH.
2. Open a shell in the Salt master container with:

```
root # docker exec -it $(docker ps | grep salt-master | awk '{print $1}') bash
```

3. Inside the container, execute the following command:

```
bash-4.3 # salt "ca" state.apply kubernetes-common.generate-serviceaccount-key
```

4. Type `exit` or press `Ctrl-D` to exit the Salt master container.

A GNU Licenses

This appendix contains the GNU Free Documentation License version 1.2.

GNU Free Documentation License

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

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

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

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

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary

formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

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

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

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

2. VERBATIM COPYING

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

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

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

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

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

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

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

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

5. COMBINING DOCUMENTS

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

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

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

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

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

ADDENDUM: How to use this License for your documents

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

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

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

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

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