Reference Configuration - Rancher Kubernetes Engine

SUSE Linux Enterprise Server 15-SP2, Rancher Kubernetes Engine 1.2.7

Reference Configuration - Rancher Kubernetes Engine: Including integration content from Cisco

SUSE Linux Enterprise Server 15-SP2, Rancher Kubernetes Engine 1.2.7

The purpose of this document is to provide an overview and procedure of implementing SUSE (R) and partner offerings for Rancher Kubernetes Engine (RKE), a Kubernetes distribution that runs entirely within containers on bare-metal and virtualized nodes. RKE solves the problem of installation complexity and the operation is both simplified and easily automated, while entirely accommodating the operating system and platform it is running on.

Publication Date: 2021-06-15

SUSE LLC
1800 South Novell Place
Provo, UT 84606
USA
https://documentation.suse.com

Contents

1 Introduction 1

1.1 Motivation 1

1.2 Scope 2

1.3 Audience 2

2 Business aspect 3

2.1 Business problem 3

2.2 Business value 4

3 Architectural overview 7

3.1 Solution architecture 7

4 Component model 9

4.1 Component overview 9

4.2 Software - Rancher Kubernetes Engine 9

4.3 Software - SUSE Linux Enterprise Micro 11

4.4 Compute Platform 12

5 Deployment 14

5.1 Deployment overview 14

5.2 Compute Platform 14

5.3 SUSE Linux Enterprise Server 16

5.4 K3s 17

5.5 Rancher Kubernetes Engine 20

6 Summary 25

7 References 26

Glossary 28

A Appendix 31

A.1 Compute Platform Bill of Materials 31

A.2 Software Bill of Materials 32

A.3 Documentation Configuration / Attributes 32

8 Legal Notice 34

9 GNU Free Documentation License 35

1 Introduction

On the digital transformation journey to a full cloud native landscape, utilization of microservices becomes the main approach with the dominant technology for such container orchestration being Kubernetes [1] With its large community of developers and abundant features and capabilities, Kubernetes has become the defacto standard and is included across most container-as-a-service platforms. With all of these technologies in place, both developer and operation teams can effectively deploy, manage and deliver functionality to their end users in a resilient and agile manner.

1.1 Motivation

As developers and organizations continue their journey from simple, containerized microservices towards having these workloads orchestrated and deployed where ever they need, being able to install, monitor and use such Kubernetes infrastructures is a core need. Such deployments, being Cloud Native Computing Foundation (CNCF [2]) conformant [3] and certified [4] are essential for both development and production workloads.

Solving common frustrations around installation complexity, Rancher Kubernetes Engine reduces many host dependencies and provides a stable path for deployment, upgrades, and rollbacks for core use cases.

Once on such a digital transformation journey, some of the next focus areas are:

- Compute Platform

1 https://kubernetes.io/

2 https://www.cncf.io/

3 https://www.cncf.io/certification/software-conformance

4 https://www.cncf.io/certification/cka/

- To optimize availability, performance, scalability and integrity, assess current system platforms or acquire and utilize new variations from:
  - Independent Hardware Vendors (IHV), like Cisco (https://www.cisco.com/) ® , as the platform for physical, baremetal, hypervisors and virtual machines

1.2 Scope

The scope of this document is to provide a layered *reference configuration* for Rancher Kubernetes Engine. This can be done in a variety of scenarios to create an enterprise Kubernetes cluster deployment anywhere.

1.3 Audience

This document is intended for IT decision makers, architects, system administrators and technicians who are implementing a flexible, software-defined Kubernetes platform. One should still be familiar with the traditional IT infrastructure pillars — networking, computing and storage — along with the local use cases for sizing, scaling and limitations within each pillars' environments.

2 Business aspect

Agility is the name of the game in modern application development. This is driving developers toward more cloud native methodologies that focus on microservices architectures and streamlined workflows. Container technologies, like Kubernetes, embody this agile approach and help enable cloud native transformation.

By unifying IT operations with Kubernetes, organizations realize key benefits like increased reliability, improved security and greater efficiencies with standardized automation. Therefore, Kubernetes infrastructure platforms are adopted by enterprises to deliver:

Cluster Operations

Improved Production and DevOps efficiencies with simplified cluster usage and robust operations

Security Policy & User Management

Consistent security policy enforcement plus advanced user management on any Kubernetes infrastructure

Access to Shared Tools & Services

A high level of reliability with easy, consistent access to a broad set of tools and services

2.1 Business problem

Many organizations are deploying Kubernetes clusters everywhere – in the cloud, on-premises, and at the edge - to unify IT operations. Such organizations can realize dramatic benefits, including:

- Consistently deliver a high level of reliability on any infrastructure
- Improve DevOps efficiency with standardized automation
- Ensure enforcement of security policies on any infrastructure

However, simply relying on upstream Kubernetes alone can introduce extra overhead and risk because Kubernetes clusters are typically deployed:

- Without central visibility
- Without consistent security policies
- And must be managed independently

Deploying a scalable kubernetes infrastructure requires consideration of a larger ecosystem, encompassing many software and infrastructure components and providers. Further, the ability to continually address the needs and concerns of:

Developers

For those who just focus on writing code to build their apps securely using a preferred workflow, providing a simple, push-button deployment mechanism of their containerized workloads where needed.

IT Operators

General infrastructure requirements still rely upon traditional IT pillars are for the stacked, underlying infrastructure. Ease of deployment, availability, scalability, resiliency, performance, security and integrity are still core concerns to be addressed for administrative control and observability.

Beyond just the core infrastructure software layers of managed Kubernetes clusters, organizations may be also be impacted by:

Compute Platform

Potential inconsistencies and impacts of multiple target system platforms for the distributed deployments of the cluster elements, across:

- physical, baremetal, hypervisors and virtual machines

2.2 Business value

With Rancher Kubernetes Engine, the operation of Kubernetes is easily automated and entirely independent of the operating system and platform running. Using a supported version of the container runtime engine, one can deploy and run Kubernetes with Rancher Kubernetes Engine. It builds a cluster from a single command in just a few minutes, and its declarative configuration makes Kubernetes upgrades atomic and safe.

By allowing operation teams to focus on infrastructure and developers to deploy code the way they want too, SUSE and the Rancher offerings helps bring products to market faster and accelerate an organization's digital transformation.

SUSE Rancher is a fundamental part of the complete software stack for teams adopting containers. It provides DevOps teams with integrated tools for running containerized workloads while also addressing the operational and security challenges of managing multiple Kubernetes clusters across any targetedd infrastructure.

Developers

SUSE Rancher makes it easy to securely deploy containerized applications no matter where the Kubernetes infrastructure runs -– in the cloud, on-premises, or at the edge. Using Helm or the App Catalog to deploy and manage applications across any or all these environments, ensuring multi-cluster consistency with a single deployment process.

IT Operators

SUSE Rancher not only deploys and manages production-grade Kubernetes clusters from datacenter to cloud to the edge, it also unites them with centralized authentication, access control and observability. Further, it streamlines cluster deployment on bare metal or virtual machines and maintains them using defined security policies.

With this increased consistency of the managed Kubernetes infrastructure clusters, organizations benefit from an even higher level of the Cloud Native Computing model where each layer only relies upon the API and version of the adjacent layer. For example:

Compute Platform

Utilizing the above software application and technology solutions with the server platforms offered by Cisco (https://www.cisco.com/) Unified Computing System (UCS (https://www.cisco.com/c/en/us/products/servers-unified-computing/index.html)) brings increased productivity, reduced total cost of ownership, and scalability into your computing realm. Cisco UCS is based upon industry-standard, x86-architecture servers with Cisco innovations and delivers a better balance of CPU, memory, and I/O resources. This balance brings processor power to life with more than 150 world-record-setting benchmark results that demonstrate leadership in application areas including virtualization, cloud computing, enterprise applications, database management systems, enterprise middleware, high-performance computing, and basic CPU integer and floating-point performance metrics.

- Match servers to workloads - The breadth of the server product line makes the process of matching servers to workloads straightforward, enabling you to achieve the best balance of CPU, memory, I/O, internal disk, and external storage-access resources using the blade, rack, multinode, or storage server form factor that best meets your organization's data center requirements and preferred purchasing model.
- Powered by AMD EPYC processors or Intel Xeon Scalable processors
- Industry-leading bandwidth - Cisco UCS virtual interface cards have dramatically simplified the deployment of servers for specific applications. By making the number and type of I/O devices programmable on demand, enables organizations to deploy and repurpose server I/O configurations without ever touching the hardware.
- Lower infrastructure cost - Designed for lower infrastructure cost per server, is a choice that makes scaling fast, easy, and inexpensive in comparison to manually configured approaches.
- Rack server deployment flexibility - Cisco UCS C-Series Rack Servers unique in the industry because they can be integrated with Cisco UCS connectivity and management or used as standalone servers
    - Integrated Management Controller (IMC) - Running in the system's Baseboard Management Controller (BMC), when a Cisco UCS C-Series Rack Servers is integrated into a Cisco UCS domain, the fabric interconnects interface with the IMC to make the server part of a single unified management domain. When a server is used as a standalone server, direct access to the IMC through the servers's management port allows a range of software tools (including Cisco Intersight) to configure the server through its API.

3 Architectural overview

This section outlines the core elements of the Rancher Kubernetes Engine solution, along with the suggested target platforms and components.

3.1 Solution architecture

The figure below illustrates the high-level architecture of Rancher Kubernetes Engine:

FIGURE 3.1: ARCHITECTURE OVERVIEW - RANCHER KUBERNETES ENGINE

FixMe-Authentication Proxy

A user is authenticated via Rancher Kubernetes Engine and then, if authorized, can access both the Rancher Kubernetes Engine environment and the downstream clusters and workloads.

FixMe-API Server

This provides the programmatic interface backend for user, command-line interactions with Rancher Kubernetes Engine and the managed clusters.

FixMe-Data Store

The purpose of this service is to capture the configuration and state of Rancher Kubernetes Engine and the managed clusters to aid in backup and recovery processes.

FixMe-Cluster Controller

Interacting with a cluster agent on the downstream cluster, the cluster controller allows the communication path for users and services to leverage for workloads and cluster management.

FixMe-Once setup, users can potentially interact with Rancher Kubernetes Engine through the web-based user interface (UI), the command-line interface (CLI), and programatically through the application programming interface (API). Depending upon the assigned roles, group membership and privileges, a user could:

- manage all clusters, users, roles, projects
- deploy new clusters, import other clusters, or remove existing ones
- manage workloads across respective or labeled clusters
- simply view clusters or workloads, or just benefit from what is running

FixMe-For the best performance and security, the recommended deployment is a dedicated Kubernetes cluster for the Rancher management server. Running user workloads on this cluster is not advised. After deploying Rancher, you can create or import clusters for running workloads.

FixMe-NOTE: Regardless of the deployment target, Rancher Kubernetes Engine should always run on a node or cluster that is separate from the downstream clusters that it manages. Running user workloads on this cluster is not advised.

4 Component model

FixMe-This section describes the various components being used to create a Rancher Kubernetes Engine deployment, in the perspective of top to bottom ordering. Once completed, the Rancher Kubernetes Engine instance enables the management of multiple Kubernetes clusters, as shown in the following figure:

4.1 Component overview

By utilizing:

- Kubernetes Platform - Rancher Kubernetes Engine
- Operating System - ifdef::iSLES[SUSE Linux Enterprise Server] ifdef::iSLEMicro[SUSE Linux Enterprise Micro}]
- Compute Platform

one can create the necessary infrastructure and services. Further details for these components are described in the following sections.

4.2 Software - Rancher Kubernetes Engine

FixMe-Rancher Kubernetes Engine is packaged as a single binary, which is about 50 megabytes in size. Bu ndled in that single binary is everything needed to run Kubernetes anywhere, inc luding low-powered IoT and Edge-based devices. The binary includes:

- the container runtime
- any important host utilities like
  - iptables, socat and du.

FixMe-The only OS dependencies are the Linux kernel itself and a proper dev, proc and sysfs mounts (this is done automatically on all modern Linux distributions). {pn _K3s} bundles the Kubernetes components:

- kube-apiserver,
- kube-controller-manager,

- kube-scheduler,
- kubelet and
- kube-proxy

into combined processes that are presented as a simple server and agent model, a s represented in the following figure:

FIGURE 4.1: FIXME-OVERVIEW OF RANCHER KUBERNETES ENGINE

Rancher Kubernetes Engine can run as a complete cluster on a single node or can be expanded into a multi-node cluster. Besides the core Kubernetes components, these are also included:

- containerd,
- Flannel,
- CoreDNS,
- ingress controller and
- a simple host port-based service load balancer.

FixMe-All of these components are optional and can be swapped out for your implementation of choice. With these included components, you get a fully functional and CNCF-conformant cluster so you can start running apps right away. Rancher Kubernetes Engine is now a CNCF Sandbox project, being the first Kubernetes distribution ever to be adopt ed into sandbox.

Learn more information about Rancher Kubernetes Engine at https://rancher.com/docs/rke/latest/en/ .

As Rancher Kubernetes Engine relies upon being deployed on a Kubernetes platform, the next section describes such a suggested component layer.

4.3 Software - SUSE Linux Enterprise Micro

SUSE Linux Enterprise Micro combines the assurance of enterprise-grade security and compliance with the immutability and portability of a modern, lightweight operating system. The top 4 features are:

Immutable OS

Immutable design ensures the OS is not altered during runtime and runs reliably every single time. Security signed and verified transactional updates are easy to rollback if things go wrong.

Security and Compliance

Fully open source and built using open standards, SUSE Linux Enterprise Micro leverages SUSE Linux Enterprise common code base, to provide FIPS 140-2, DISA SRG/STIG, integration with CIS and Common Criteria certified configurations. Includes fully supported security framework (SELinux) with policies.

Architectural Flexibility

Both Arm and x86-64 architectures are supported so you can deploy edge applications with confidence across multiple architectures.

Kubernetes-Ready

You can easily combine SUSE Linux Enterprise Micro with the latest cloud-native technologies including SUSE Rancher, Rancher Kubernetes Engine, Longhorn persistent block storage, and K3s, the world's most popular Kubernetes distribution for use in low resource, distributed edge locations.

As a result, you get an ultra-reliable infrastructure platform that is also simple to use and comes out-of-the-box with best-in-class compliance. Furthermore, SUSE's flexible subscription model ensures enterprise assurance for any edge, embedded or IoT deployment without vendor lock-in. A free, evaluation copy can be downloaded (https://www.suse.com/download/sle-micro/) or if the organization already has subscriptions, both install media and updates can be obtained from SUSE Customer Center (https://scc.suse.com/login).

With the flexibility of SUSE Linux Enterprise Micro, multiple compute platform variants can be considered, as outlined in the next section.

4.4 Compute Platform

Leveraging the enterprise grade functionality of the operating system mentioned in the previous section, many compute platforms can be the foundation of the deployment:

Cisco UCS C-Series Rack Servers (https://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/index.html)

UCS C-Series Rack Servers delivers unified computing in an industry-standard form factor to reduce TCO and increase agility. Each server addresses varying workload challenges through a balance of processing, memory, I/O, and internal storage resources. These servers can be deployed as standalone servers or as part of a Cisco Unified Computing System (Cisco UCS) managed environment to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' Total Cost of Ownership (TCO) and increase their business agility. ~

Server product-line and model options abound in the Cisco UCS C-Series Rack Servers (https://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-c-series-rack-servers/index.html), including:

- Cisco UCS C240 SD M5 is a high-performance compute solution in a dense 2-socket, 2-Rack-Unit, 22" form-factor to handle the most critical real-time compute applications. This front-access server can be deployed as standalone servers or as part of a Cisco Unified Computing System (Cisco UCS) to deliver an exceptional management experience for a variety of applications by:
  - incorporating the 2nd generation of Intel® Xeon® Scalable processors, Intel® Optane™ Memory, and various drive options including All-NVMe, SAS and SATA drives.
  - being density optimized to accommodate space constrained environments while still offering industry-leading performance and expandability. It supports a wide range of workloads from enterprise to edge applications such as Multi-access Edge Compute (MEC).

> **Note**
>
> Cisco UCS Hardware Compatibilty List (https://ucshcltool.cloudapps.cisco.com/public/) provides a lookup tool for Servers & OS Support, for versions of SUSE offerings.

Tip

Any SUSE YES (https://www.suse.com/yessearch/) certified platform can be used for the nodes of this deployment, as long as the certification refers to the major version of the underlying SUSE operating system required by its release.

Note

A sample bill of materials, in the *Appendix A, Appendix*, cites the necessary quantites of all components, along with a reference to the minimum resource requirements needed by the software components.

5 Deployment

This section describes the process steps for the deployment of the Rancher Kubernetes Engine solution. It describes the process steps to deploy each of the component layers starting as a base functional *proof-of-concept*, having considerations on migration towards *production*, providing *scaling* guidance that is needed to create the solution.

5.1 Deployment overview

The deployment stack is represented in the following figure:

FIGURE 5.1: SUSE RANCHER DEPLOYMENT STACK

and details are covered for each layer in the following sections.

Note

The following section's content is ordered and described from the bottom layer up to the top.

5.2 Compute Platform

The base, starting configuration can reside all within a single Cisco UCS. Based upon the relatively small resource requirements for a deployment, a viable approach is to deploy as a virtual machine (VM) on the target nodes, on top of an existing hypervisor, like KVM.

Preparation(s)

For a physical host, like C240 SD M5 (https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/datasheet-c78-743260.html) used in the deployment:

1. If using Cisco UCS Manager

- Log into the Cisco UCS Manager
  - Select the Equipment tab
  - In the navigation pane expand Rack-Mounts and then Servers
  - Right-click the server and select KVM console
  - In the right pane, click the KVM Console
    - Click the link to launch the KVM console
    - Select the Virtual Media tab and activate Virtual Devices found in Virtual Media tab
    - Click the Virtual Media tab to select CD/DVD
    - Select Map Drive in the Virtual Disk Management window and browse to respective operating system media, open and use the image for a system boot.

Deployment Process

On the respective compute module node, determine if a hypervisor is already available for the solution's virtual machines.

1. If this will be the first use of this node, an option is to deploy a KVM hypervisor, based upon SUSE Linux Enterprise Server by following the Virtualization Guide (https://documentation.suse.com/sles/15-SP2/single-html/SLES-virtualization/#book-virt).
   - Given the simplicity of the deployment, the operating system and hypervisor can be installed with the SUSE Linux Enterprise Server ISO media and the Cisco IMC virtual media and virtual console methodology.
2. Then for the solution VM, utilize the hypervisor user interface to allocate the necessary CPU, memory, disk and networking as noted in the SUSE Rancher hardware requirements (https://rancher.com/docs/rancher/v2.x/en/installation/requirements/#hardware-requirements).

Deployment Consideration(s)

To further optimize deployment factors, leverage the following practices:

- *Availability*

- While the initial deployment only requires a single VM, as noted in later deployment sections, having multiple VMs provides resiliency to accomplish high availability. To reduce single points of failure, it would be beneficial to have the multi-VM deployments spread across multiple hypervisor nodes. So consideration of consistent hypervisor and compute module configurations, with the needed resources for the SUSE Rancher VMs will yield a robust, reliable production implementation.

5.3 SUSE Linux Enterprise Server

Utilize an enterprise-grade Linux operating system , like SUSE Linux Enterprise Server, as the base software layer.

Preparation(s)

To meet the solution stack prerequisites and requirements, SUSE operating system offerings, like SUSE Linux Enterprise Server (https://www.suse.com/products/server/) can be utilized.

1. Ensure these services are in place and configured for this node to use:
   - Domain Name Service (DNS) - an external network-accessible service to map IP Addresses to hostnames
   - Network Time Protocol (NTP) - an external network-accessible service to obtain and synchronize system times to aid in timestamp consistency
   - Software Update Service - access to a network-based repository for software update packages. This can be accessed directly from each node via registration to
     - the general, internet-based SUSE Customer Center (https://scc.suse.com/login) (SCC) or
     - an organization's SUSE Manager (https://www.suse.com/products/suse-manager/) infrastructure or
     - a local server running an instance of Repository Mirroring Tool (https://documentation.suse.com/sles/15-SP2/single-html/SLES-rmt/#book-rmt) (RMT)

> **Note**
>
> During the node's installation, it can be pointed to the respective update service. This can also be accomplished post-installation with the command-line tool named SUSEConnect (https://www.suse.com/support/kb/doc/?id=000018564) ↗.

Deployment Process

On the compute platform node, install the noted SUSE operating system, by following these steps:

Deployment Consideration(s)

To further optimize deployment factors, leverage the following practices:

- *Automation*
  - To reduce user intervention, unattended deployments of SUSE Linux Enterprise Micro can be automated
    - for ISO-based installations, by referring to the AutoYaST Guide (https://documentation.suse.com/sle-micro/5.0/single-html/SLE-Micro-autoyast/#book-autoyast) ↗
    - for raw-image based installation, by configuring the Ignition and Combustion tooling as described in the Installation Quick Start (https://documentation.suse.com/sle-micro/5.0/single-html/SLE-Micro-installation/#article-installation) ↗

5.4 K3s

Utilize an enterprise-grade Linux operating system , like SUSE Linux Enterprise Server, as the base software layer.

Preparation(s)

To meet the solution stack prerequisites and requirements, SUSE operating system offerings, like SUSE Linux Enterprise Server (https://www.suse.com/products/server/) ↗ can be utilized.

1. Ensure these services are in place and configured for this node to use:

   - Domain Name Service (DNS) - an external network-accessible service to map IP Addresses to hostnames
   - Network Time Protocol (NTP) - an external network-accessible service to obtain and synchronize system times to aid in timestamp consistency
   - Software Update Service - access to a network-based repository for software update packages. This can be accessed directly from each node via registration to
     - the general, internet-based SUSE Customer Center (https://scc.suse.com/login) (SCC) or
     - an organization's SUSE Manager (https://www.suse.com/products/suse-manager/) infrastructure or
     - a local server running an instance of Repository Mirroring Tool (https://documentation.suse.com/sles/15-SP2/single-html/SLES-rmt/#book-rmt) (RMT)

   Note

   During the node's installation, it can be pointed to the respective update service. This can also be accomplished post-installation with the command-line tool named SUSEConnect (https://www.suse.com/support/kb/doc/?id=000018564).

2. Identify the appropriate, desired version of the K3s binary (e.g. vX.YY.ZZ+k3s1), by reviewing the "Releases" on the Download (https://github.com/k3s-io/k3s/) web page.

Deployment Process

The primary steps for deploying this K3s Kubernetes are:

1. Set the following variable with the noted version of K3s, as found during the preparation steps.

```
K3s_VERSION=""
```

2. Install the version of K3s with embedded Etcd enabled:

```
curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION=${K3s_VERSION}
 INSTALL_K3S_EXEC='server --cluster-init --write-kubeconfig-mode=644' sh -s -
```

> **Tip**
>
> To address *Availability* and possible *scaling* to a multiple node cluster, Etcd is enabled instead of using the default SQLite datastore.

- Monitor the progress of the installation: `watch -c "kubectl get deployments -A"`
  - The deployment is complete when elements of all the deployments (coredns, local-path-provisioner, metrics-server, and traefik) show at least "1" as "AVAILABLE"
  - Use Ctrl+c to exit the watch loop after all deployment pods are running

Deployment Consideration(s)

To further optimize deployment factors, leverage the following practices:

- *Availability*
  - While a single K3s node works perfectly fine, a full high-availability K3s cluster is recommended for production workloads. The Etcd key/value store (aka database) requires an odd number of nodes be allocated to the K3s plane (aka master nodes). In this case, two additional control-plane nodes should be added; for a total of three.

    1. Deploy the same operating system on the new compute platform nodes, then log into the new nodes as root or as a user with sudo privileges.
    2. Execute the following sets of commands on each of the remaining control-plane nodes:

    ```
    FIRST_SERVER_IP=""      # Private IP preferred, if available
    ```

```
NODE_TOKEN=""            # From the /var/lib/rancher/k3s/server/node-
token file on the first server
K3s_VERSION=""           # Match the first of the first server
```

```
curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION=${K3s_VERSION}
 K3S_URL=https://${FIRST_SERVER_IP}:6443 K3S_TOKEN=${NODE_TOKEN}
 K3S_KUBECONFIG_MODE="644" INSTALL_K3S_EXEC='server' sh -
```

> **Note**
>
> This can be changed to the normal Kubernetes default by adding a taint to each server node. See the official Kubernetes documentation for more information on how to do that.

3. (Optional) In cases where agent nodes are desired, execute the following sets of commands on each of the agent nodes to add it to the K3s cluster:

```
FIRST_SERVER_IP=""       # Private IP preferred, if available
NODE_TOKEN=""            # From the /var/lib/rancher/k3s/server/node-
token file on the first server
K3s_VERSION=""           # Match the first of the first server
```

```
curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION=${K3s_VERSION}
 K3S_URL=https://${FIRST_SERVER_IP}:6443 K3S_TOKEN=${NODE_TOKEN}
 K3S_KUBECONFIG_MODE="644" sh -
```

include::../SUSE/RKE1/SA_vars.adoc

5.5 Rancher Kubernetes Engine

Utilize an enterprise-grade Linux operating system , like SUSE Linux Enterprise Server, as the base software layer.

Preparation(s)

To meet the solution stack prerequisites and requirements, SUSE operating system offerings, like SUSE Linux Enterprise Server (https://www.suse.com/products/server/) can be utilized.

1. Ensure these services are in place and configured for this node to use:

   - Domain Name Service (DNS) - an external network-accessible service to map IP Addresses to hostnames

   - Network Time Protocol (NTP) - an external network-accessible service to obtain and synchronize system times to aid in timestamp consistency

   - Software Update Service - access to a network-based repository for software update packages. This can be accessed directly from each node via registration to

     - the general, internet-based SUSE Customer Center (https://scc.suse.com/login) (SCC) or

     - an organization's SUSE Manager (https://www.suse.com/products/suse-manager/) infrastructure or

     - a local server running an instance of Repository Mirroring Tool (https://documentation.suse.com/sles/15-SP2/single-html/SLES-rmt/#book-rmt) (RMT)

     Note

     During the node's installation, it can be pointed to the respective update service. This can also be accomplished post-installation with the command-line tool named SUSEConnect (https://www.suse.com/support/kb/doc/?id=000018564).

2. Identify the appropriate, desired version of the Rancher Kubernetes Engine binary (e.g. vX.Y.Z), by reviewing the "Releases" on the Download (https://github.com/k3s-io/k3s/) web page.

Deployment Process

NOTE: Installing Rancher Kubernetes Engine requires a client system (i.e. admin workstation) that has been configured with kubectl.

The primary steps for deploying this Rancher Kubernetes Engine Kubernetes are:

+ . Download the Rancher Kubernetes Engine binary according to the instructions on this webpage: https://rancher.com/docs/rke/latest/en/installation/ ↗. Follow the directions on that page, but with the following exceptions:

- Create the cluster.yml file with the command `rke config`

> **Note**
>
> See https://rancher.com/docs/rke/latest/en/example-yamls/ ↗ and https://rancher.com/docs/rke/latest/en/config-options/ ↗ for full descriptions of the cluster.yml parameters

- It is recommended to create a unique SSH key for this Rancher Kubernetes Engine cluster with the command `ssh-keygen`
  - Provide the path to that key for the option "Cluster Level SSH Private Key Path"
- The option "Number of Hosts" refers to the number of hosts to configure at this time
  - Additional hosts can be added very easily after Rancher Kubernetes Engine cluster creation
  - For this implementation, it is recommended to configure one, three, or five hosts
- Give all hosts the roles of "Control Plane", "Worker", and "etcd"
- Answer "n" for the option "Enable PodSecurityPolicy"
  - Update the cluster.yml file before continuing with the step "Deploying Kubernetes with RKE"
    1. If a load balancer has been deployed for the Rancher Kubernetes Engine control-plane nodes, update the cluster.yml file (created with the command `rke config`) before deploying RKE to include the IP address or FQDN of the load balancer. The appropriate location is under authentication.sans. For example:

```
authentication:
  strategy: x509
  sans: ["rancher.susealliances.com"]
```

- Verify password-less SSH is available from the system with the `rke` utility to each of the cluster hosts, as the user specified in the cluster.yml file
- When ready, run `rke up` to create the RKE cluster
- After the `rke up` command completes, the RKE cluster will continue the Kubernetes installation process
- Monitor the progress of the installation:
  - Export the variable KUBECONFIG to the absolute pathname of the kube_config_cluster.yml file. I.e. `export KUBECONFIG=~/rke-cluster/kube_config_cluster.yml`
  - Run the command: `watch -c "kubectl get deployments -A"`
    - The cluster deployment is complete when elements of all the deployments show at least "1" as "AVAILABLE"
    - Use Ctrl+c to exit the watch loop after all deployment pods are running

> **Tip**
>
> To address *Availability* and possible *scaling* to a multiple node cluster, Etcd is enabled instead of using the default SQLite datastore.

Deployment Consideration(s)

To further optimize deployment factors, leverage the following practices:

- *Availability*

- While a single Rancher Kubernetes Engine node works perfectly fine, a full high-availability Rancher Kubernetes Engine cluster is recommended for production workloads. The Etcd key/value store (aka database) requires an odd number of nodes be allocated to the Rancher Kubernetes Engine etcd nodes. In this case, two additional etcd nodes should be added; for a total of three.

After this successful deployment of the Rancher Kubernetes Engine solution, review the product documentation (https://rancher.com/docs/rke/latest/en/) for details on how to directly utilize this Kubernetes cluster. Furthermore, by reviewing the SUSE Rancher product documentation (https://rancher.com/docs/rancher/v2.x/en/) this solution can also be:

- imported (refer to sub-section "Importing Existing Clusters"), then
- managed (refer to sub-section "Cluster Administration") and
- accessed (refer to sub-section "Cluster Access") to address orchestration of workloads, maintaining security and many more functions are readily available.

6 Summary

Using components and offerings from SUSE and the Rancher portfolio streamlines the ability to quickly and effectively engage in a digital transformation, taking advantage of cloud native resources and disciplines. Using such technology approaches lets you deploy and leverage transformations of infrastructure into a durable, reliable enterprise-grade environment.

Simplify

Simplify and optimize your existing IT environments

- FixMe-Using Rancher Kubernetes Engine enables you to simplify Kubernetes cluster deployment and management of the the infrastructure components.

Modernize

Bring applications and data into modern computing

- FixMe-With Rancher Kubernetes Engine, the digital transformation to containerized applications can benefit from the ability both to manage many target clusters, for each of the respective user bases and to facilitate the actual workload deployments.

Accelerate

Accelerate business transformation through the power of open source software

- FixMe-Given the open source nature of Rancher Kubernetes Engine and the underlying software components, you can simplify management and make significant IT savings as you scale orchestrated, microservice deployments anywhere you need to and for whatever use cases are needed in an agile and innovative way.

7 References

WHITE PAPERS

- A Buyer's Guide to Enterprise Kubernetes Management Platforms - https://info.rancher.com/enterprise-kubernetes-management-buyers-guide
- How to Build an Enterprise Kubernetes Strategy - https://info.rancher.com/how-to-build-enterprise-kubernetes-strategy

BOOKS

- Kubernetes Management - https://info.rancher.com/kubernetes-management-for-dummies-rancher-and-suse-0-0

TRAINING

- SUSE - https://training.suse.com/
  - Rancher - https://rancher.com/training/

WEBSITES

- SUSE - https://www.suse.com
  - SUSE Customer Center (SCC) - https://scc.suse.com/login
  - Products
    - SUSE Rancher - https://rancher.com/products/rancher/ (documentation (https://rancher.com/docs/rancher/v2.x/en/))
    - Rancher Kubernetes Engine (RKE) - https://rancher.com/products/rke/ (documentation (https://rancher.com/docs/rke/latest/en/))
    - K3s - https://rancher.com/products/k3s/ (documentation (https://rancher.com/docs/k3s/latest/en/))
    - SUSE Linux Enterprise Micro (SLEMicro) - https://www.suse.com/products/micro/ (documentation (https://documentation.suse.com/sle-micro/5.0/))
    - SUSE Linux Enterprise Server (SLES) - https://www.suse.com/products/server/ (documentation (https://documentation.suse.com/sles/15-SP2/))

- SUSE Manager - https://www.suse.com/products/suse-manager/ (documentation (https://documentation.suse.com/suma/4.1/))
- SUSE Repository Mirroring Tool (RMT) - https://www.suse.com/products/server/ (documentation (https://documentation.suse.com/sles/15-SP2/single-html/SLES-rmt/#book-rmt))

- Projects
  - Rancher Kubernetes Engine Government (RKE2) - https://github.com/rancher/rke2 (documentation (https://docs.rke2.io/))

Glossary

- Document Scope

Reference Configuration

A guide with the basic steps to deploy the layered stack of components from both the SUSE and partner portfolios. This is considered a fundamental basis to demonstrate a specific, tested configuration of components.

Reference Architectures [1]

A guide with the general steps to deploy and validate the structured solution components from both the SUSE and partner portfolios. This provides a shareable template of consistency for consumers to leverage for similar production ready solutions, including design considerations, implementation suggestions and best practices.

Best Practice

Information that can overlap both the SUSE and partner space. It can either be provided as a standalone guide that provides reliable technical information not covered in other product documentation, based on real-life installation and implementation experiences from subject matter experts or complementary, embedded sections within any of the above documentation types describing considerations and possible steps forward.

- Factor(s)

Automation [2]

Infrastructure automation enables speed through faster execution when configuring the infrastructure and aims at providing visibility to help other teams across the enterprise work quickly and more efficiently. Automation removes the risk associated with human error, like manual misconfiguration; removing this can decrease downtime and increase reliability. These outcomes and attributes help the enterprise move towards implementing a culture of DevOps, the combined working of development and operations.

---

1 link: Reference Architecture (https://en.wikipedia.org/wiki/Reference_architecture) ↗

2 link: Infrastructure-as-Code (https://en.wikipedia.org/wiki/Infrastructure_as_code) ↗

Availability [3]

The probability that an item operates satisfactorily, without failures or downtimes, under stated conditions as a function of its reliability, redundancy and maintainability attributes. Some major objectives to achieve a desired service level objectives are:

- Preventing or reducing the likelihood and frequency of failures via design decisions within the allowed cost of ownership
- Correcting or coping with possible component failures via resiliency, automated failover and disaster-recovery processes
- Estimating and analyzing current conditions to prevent unexpected failures via predictive maintenance

Integrity [4]

Integrity is the maintenance of, and the insurance of the accuracy and consistency of a specific element over its entire lifecycle. Both physical and logical aspects must be managed to ensure stability, performance, re-usability and maintainability.

Security [5]

Security is about ensuring freedom from or resilience against potential harm, including protection from destructive or hostile forces. To minimize risks, one mus manage governance to avoid tampering, maintain access controls to prevent unauthorized usage and integrate layers of defense, reporting and recovery tactics.

- Deployment Flavor(s)

Proof-of-Concept [6]

A partial or nearly complete prototype constructed to demonstrate functionality and feasibility for verifying specific aspects or concepts under consideration. This is often a starting point when evaluating a new, transitional technology. Sometimes it starts as a Minimum Viable Product (MVP [7]) that has just enough features to satisfy an

3 link: Availability (https://en.wikipedia.org/wiki/Minimum_viable_product)

4 link: Data Integrity (https://en.wikipedia.org/wiki/Data_integrity)

5 link: Security (https://en.wikipedia.org/wiki/Security)

6 link: Proof of Concept (https://en.wikipedia.org/wiki/Proof_of_concept)

7 link: Minimum Viable Product (https://en.wikipedia.org/wiki/Minimum_viable_product)

initial set of requests. After such insights and feedback are obtained and potentially addressed, redeployments may be utilized to iteratively branch into other realms or to incorporate other known working functionality.

Production

A deployed environment that target customers or users can interact with and rely upon to meet their needs, plus be operationally sustainable in terms of resource utilization and economic constraints.

Scaling

The flexibility of a system environment to either vertically scale-up, horizontally scale-out or conversely scale-down by adding or subtracting resources as needed. Attributes like capacity and performance are often the primary requirements to address, while still maintaining functional consistency and reliability.

A Appendix

The following sections provide a bill of materials listing for each component layer.

A.1 Compute Platform Bill of Materials

Role	Qty	SKU	Component	Notes
Compute Platform	1-3	UCSC-C240-M5SD	Cisco UCS C240 SD M5	Configuration • 2x Intel 5218 (16-core, 2.3GHz) • 256GB RAM • 2x 600GB SAS 12G 10k HDD (OS) • 2x 1.2-2.4TB SAS 12G 10k HDD (local storage)

A.2 Software Bill of Materials

Role	Qty	SKU	Component	Notes
Operating System		1-3	874-006875	SUSE Linux Enterprise Server, • x86_64, • Priority Subscription, • 1 Year
Configuration: * per node (up to 2 sockets, stackable) or 2 VMs	Kubernetes	1	R-0003-PS1	Rancher Kubernetes Engine, • x86-64, • Priority Subscription, • 1 Year

Note

For the software components, other durations of support terms are also available.

A.3 Documentation Configuration / Attributes

This document was built using the following AsciiDoc (https://github.com/asciidoc/asciidoc) and DocBook Authorting and Publishing Suite (DAPS (https://github.com/openSUSE/daps)) attributes:

```
Automation=1@ Availability=1@ FCTR=1@ FLVR=1@ IHV-Cisco-C240-SD=1@ IHV-Cisco=1@
 Integrity=1@ PoC=1@ Production=1@ RC=1@ Scaling=1@ Security=1@ focusRKE1=1@ iHV=1@
```

```
layerK3s=1@ layerSLES=1@ Appendix=1 ArchOv=1 BP=1 BPBV=1 CompMod=1 DepConsiderations=1
Deployment=1 GFDL=1 Glossary=1 HWComp=1 HWDepCfg=1 LN=1 RA=1 References=1 Requirements=1
SWComp=1 SWDepCfg=1 env-daps=1 iK3s=1 iRKE1=1 iRKE2=1 iRMT=1 iRancher=1 iSLEMicro=1
iSLES=1 iSUMa=1
```

8 Legal Notice

This article is part of a series of documents called "SUSE Best Practices". The individual documents in the series were contributed voluntarily by SUSE's employees and by third parties. The articles are intended only to be one example of how a particular action could be taken.

Also, SUSE cannot verify either that the actions described in the articles do what they claim to do or that they don't have unintended consequences.

All information found in this article has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Therefore, we need to specifically state that neither SUSE LLC, its affiliates, the authors, nor the translators may be held liable for possible errors or the consequences thereof. Below we draw your attention to the license under which the articles are published.

9 GNU Free Documentation License

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

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

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

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

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

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

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

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

2. VERBATIM COPYING

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

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

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

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

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

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

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

5. COMBINING DOCUMENTS

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

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

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
```

```
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled “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.