Technical Reference Documentation

# Layered Stack Deployment of K3s

## Integrated with Ampere (R) Altra (R)

## Layered Stack Deployment of K3s: Integrated with Ampere (R) Altra (R)

SUSE Linux Enterprise Server 15 SP3, K3s 1.20.14

The purpose of this document is to provide an overview and procedure of implementing SUSE (R) and partner offerings for K3s, an official CNCF sandbox project that delivers a lightweight yet powerful certified Kubernetes distribution designed for production workloads across resource-restrained, remote locations or on Edge IoT devices.



Publication Date: 2022-08-29

https://documentation.suse.com

# Contents

# 1 Introduction

On the digital transformation journey to a full cloud-native landscape, the use 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 de-facto 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

Once on such a digital transformation journey, also relevant to focus on areas like:

**Workload(s)**

Determine how to manage and launch internally developed containerized, microservice workloads

**Kubernetes**

As developers and organizations continue their journey from simple, containerized microservices toward 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 and certified[3] are essential for both development and production workloads.

- For simplified scenarios, like edge, remote or IoT, this is where K3s leads the industry, being simple and secure.

**Compute Platform(s)**

To optimize availability, performance, scalability and integrity, assess current system or hosting platforms

1 https://kubernetes.io/

2 https://www.cncf.io/

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

from Independent Hardware Vendors (IHV), such as Ampere Computing (https://ampere-computing.com/) ® cloud-native processors as the platform for physical, bare metal, hypervisors and virtual machines

## 1.2 Scope

The scope of this document is to provide a layered *reference configuration* for K3s. This can be done in a variety of scenarios to create an edge-oriented, lightweight Kubernetes cluster deployment.

## 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 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

Kubernetes is the leading solution to address edge computing use cases in industry verticals such as manufacturing, transportation, power generation, healthcare, retail and banking. Typical edge systems that leverage Kubernetes to run complex workloads include energy meters, aircraft engines, gas & oil rigs, cruise ships, high-speed trains, retail scanners, wind turbine base stations, internet-connected cars, ATMs and much more.

For such target edge systems, which are often unattended, resource constrained and remote, orchestrating containerized workloads on Kubernetes deployments may seem overbearingly complex.

## 2.2 Business value

After two years of research and development in June 2020, K3s was donated to the CNCF. The donation is a testament of the commitment to the open source community and their mission to run Kubernetes everywhere.

**Perfect for Edge**

K3s is a highly available, certified Kubernetes distribution specifically designed for production workloads in unattended, resource-constrained, remote locations or inside IoT appliances.

**Simplified & Secure**

K3s is packaged as a tiny, single binary that reduces the dependencies and steps needed to install, run and auto-update a production Kubernetes cluster. For workloads, automated Manifest and Helm Chart management deployments can be used. Also, multiple architectures, like x86_64, ARM64, and ARMv7, are supported with binaries and images available.

Given its extensive Kubernetes capabilities, K3s can also be a suitable choice for:

- embedded platforms,
- continuous integration and continuous deployment platforms,
- branch locations or individual developer deployments, and
- even core or cloud production instances

### Tip

When K3s is imported and combined with SUSE Rancher, organizations are equipped with an easy, complete and reliable management solution for Kubernetes at the edge.

With this increased consistency of the deployed and 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, such as:

**Compute Platform**

The above software application and technology solutions are used with the platforms utilizing Ampere Computing (https://amperecomputing.com/) that provides the industry's first cloud native processors. As the term implies, these compute platforms are built to host many containers that must adhere to strict service level agreement (SLA) requirements, uniquely delivering the performance, scalability, security and power efficiency that is focused on today's hyperscale cloud and edge computing workloads and applications. Providing the highest total performance and performance per watt of power, these Ampere

Arm-based processors deliver a sustainable server solution for data center applications and enabling cloud service providers to meet the compute demands of the future with only a fraction of the power and real estate consumed today.

- High Performance
    - Ampere processor products are single-threaded, run at consistently high frequencies, and are built with large low-latency private caches. This results in predictable high performance that elastically scales in a linear fashion for all the cloud native applications built in containers and managed by Kubernetes. The architecture also lends itself to high utilization and delivers consistent performance under maximum load conditions.
- Linear Scalability
    - Ampere cloud native processors add more and often get more of the same performance you got last time. In addition to being the leader in power efficiency, Ampere's high performance cloud native processors contain up to 128 cores, by far the most in the industry. These cloud native processors were architected and designed from the ground up to deliver on the metrics that matter for modern cloud native applications including a level of rack scalability that allows a rack density of over 3000 cores yielding the best container density in the industry for micro-service based services orchestrated by cloud native tools like Kubernetes.
- Predictability
    - Ampere aligns both the hardware and software to provide constant and consistent performance even under heavy load conditions.
- Sustainability
    - Cloud Service Providers' relentless drive to achieve carbon neutrality requires power efficient servers. Ampere's extremely power efficient cores that deliver exceptional performance. Legacy processors fall short of meeting modern cloud compute requirements. In fact data centers built with Ampere Altra Max processors at scale can be up to 60% more efficient from a power consumption perspective for equivalent application performance needs.

# 3 Architectural overview

This section outlines the core elements of the K3s solution, along with the suggested target platforms and components.

## 3.1 Solution architecture

The figure below illustrates the high-level architecture of K3s:

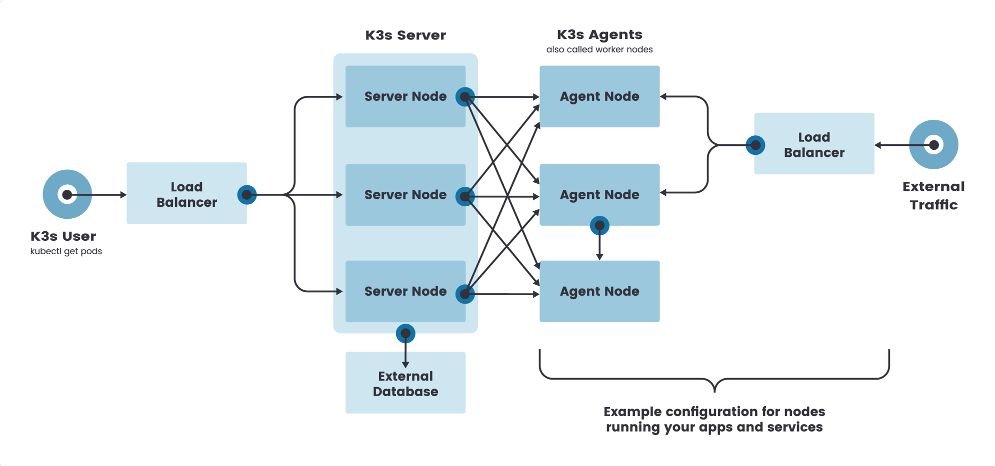


FIGURE 3.1: ARCHITECTURE OVERVIEW - K3S

### Container Runtime

- Containerd & runc
- Kine as a datastore shim that allows `etcd` to be replaced with other databases

### Networking

- Flannel for CNI
- Kube-router for network policy

### Services

- CoreDNS
- Metrics Server

    - Traefik for ingress
    - Klipper-lb as an embedded service load balancer provider
    - Local-path-provisioner for provisioning volumes using local storage

**Workloads**
: Helm-controller to allow for CRD-driven deployment of helm manifests

**Host utilities**
: iptables/nftables, ebtables, ethtool, and socat

When this is set up, users can interact with K3s via

- kubectl
    - directly on the K3s host or
    - remotely, leveraging the KUBECONFIG file of the K3s cluster's deployment (`/etc/rancher/k3s/k3s.yaml`)
- manual or automatic, manifest or Helm Chart based, workload deployments

# 4 Component model

This section describes the various components being used to create a K3s solution deployment, in the perspective of top to bottom ordering. When completed, the K3s instance can be used as the application infrastructure for cloud-native workloads and can be imported into SUSE Rancher for management.

## 4.1 Component overview

By using:

- Software
  - Kubernetes Platform - K3s
  - Linux Operating System - SUSE Linux Enterprise Server
- Compute Platform
  - Ampere Altra Family

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

## 4.2 Software - K3s

K3s is packaged as a single binary, which is about 50 megabytes in size. Bundled in that single binary is everything needed to run Kubernetes anywhere, including low-powered IoT and Edge-based devices. The binary includes:

- the container runtime
- important host utilities such as iptables, socat and du

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). 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, as represented in the following figure:

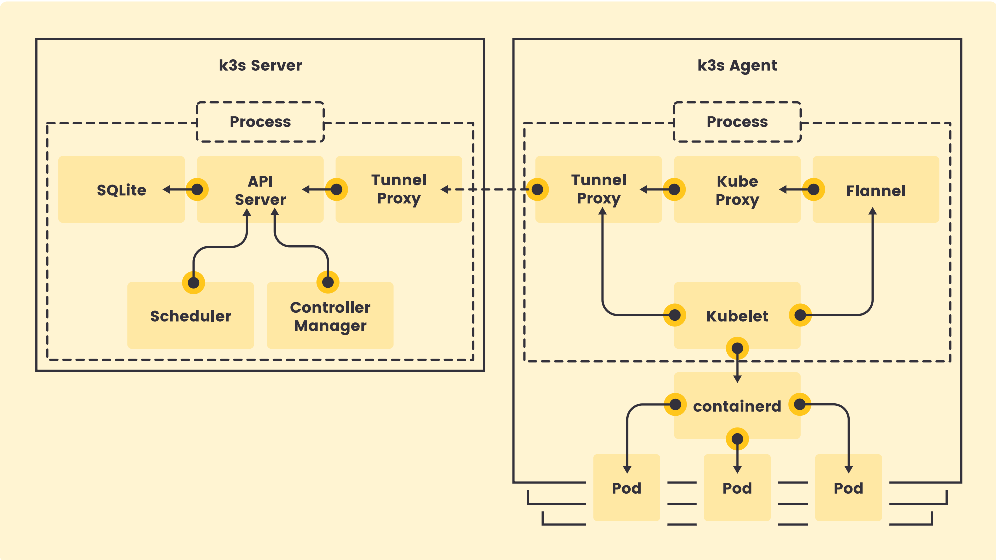


FIGURE 4.1: COMPONENT OVERVIEW - K3S

K3s 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.

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. K3s is now a CNCF Sandbox project, being the first Kubernetes distribution ever to be adopted into sandbox.

### Tip

Learn more information about K3s (https://www.suse.com/products/k3s/)

As K3s can be deployed on a single or multiple nodes, the next sections describe the suggested component layering approach.

## 4.3 Software - SUSE Linux Enterprise Server

SUSE Linux Enterprise Server (SLES (https://www.suse.com/products/server/)) is an adaptable and easy-to-manage platform that allows developers and administrators to deploy business-critical workloads on-premises, in the cloud and at the edge. It is a Linux operating system that is adaptable to any environment – optimized for performance, security and reliability. As a multimodal operating system that paves the way for IT transformation in the software-defined era, this simplifies multimodal IT, makes traditional IT infrastructure efficient and provides an engaging platform for developers. As a result, one can easily deploy and transition business-critical workloads across on-premises and public cloud environments.

Designed for interoperability, SUSE Linux Enterprise Server integrates into classical Unix and Windows environments, supports open standard interfaces for systems management, and has been certified for IPv6 compatibility. This modular, general purpose operating system runs on four processor architectures and is available with optional extensions that provide advanced capabilities for tasks such as real time computing and high availability clustering. SUSE Linux Enterprise Server is optimized to run as a high performing guest on leading hypervisors and supports an unlimited number of virtual machines per physical system with a single subscription. This makes it the perfect guest operating system for virtual computing.

## 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:

- Virtual machines on supported hypervisors or hosted on cloud service providers
- Physical, baremetal or single-board computers, either on-premises or hosted by cloud service providers

### Note

To complete self-testing of hardware with SUSE YES Certified Process (https://www.suse.com/partners/ihv/yes/yes-certified-process) ↗, you can download and install the respective SUSE operating system support-pack version of SUSE Linux Enterprise Server and the YES test suite. Then run the tests per the instructions in the test kit, fixing any problems encountered and when corrected, re-run all tests to obtain clean test results. Submit the test results into the SUSE Bulletin System (SBS) for audit, review and validation.

### Tip

Certified systems and hypervisors can be verified via SUSE YES Certified Bulletins (https://www.suse.com/yessearch/) ↗ and then can be leveraged as supported nodes for this deployment, as long as the certification refers to the respective version of the underlying SUSE operating system required.

### 4.4.1 Ampere Altra Family

The Ampere Altra (https://amperecomputing.com/processors/ampere-altra/) ↗ Arm v8.2 processor portfolio of world's first cloud native processors is widely available with data center ready configurations from our systems partners and for use with many Cloud Service Providers. Explore the Ampere Computing Platforms (https://amperecomputing.com/reference-platforms/ampere-altra-platforms-for-modern-compute/) ↗ offered from our partners. These systems are flexible enough to meet the needs of any cloud deployment and come packed with Ampere 80-core Altra or 128-core Altra Max processors.

The specific processor models that offer relevant choices for Enterprise Kubernetes are designed to meet the requirements of modern data centers, deliver predictable performance, high scalability, and power efficiency for data center deployments from hyperscale cloud to the edge cloud. These processors that drive efficiency in your data center infrastructure workloads, including data analytics, artificial intelligence, database storage, telco stacks, edge computing, and Web hosting, are:

- Ampere Altra 64-Bit Multi-Core Processor
  - Predictable Performance - Ampere Altra offers up to 80 cores at up to 3.30 GHz speed maximum. Each core is single-threaded by design with its own 64 KB L1 I-cache, 64 KB L1 D-cache, and a huge 1 MB L2 cache, delivering predictable performance all along by eliminating the noisy neighbor challenge within each core.
  - Power Efficiency - provides industry-leading power efficiency/core, while packing 80 cores in a single-socket and 160 cores in a dual-socket platform, establishing new levels of power efficiency with scalability to meet the most strenuous application infrastructure needs.
- Ampere Altra Max 64-Bit Multi-Core Processor
  - Predictable Performance - Ampere Altra Max offers up to 128 cores operating at a maximum of 3.0 GHz. Each core is single-threaded by design with its own 64 KB L1 I-cache, 64 KB L1 D-cache, and a huge 1 MB L2 cache, delivering predictable performance 100% of the time by eliminating the noisy neighbor challenge within each core.
  - Power Efficiency - provides industry-leading power efficiency/core, while packing 128 cores in a single-socket and 256 cores in a dual-socket platform, establishing new levels of power efficiency with scalability.

Furthermore, each of these processors features:

- High Scalability - With leading power/core, and multi-socket support, it provides the scalability to maximize the number of servers per rack, unparalleled in the industry.
- Reliability, Availability, and Serviceability (RAS) - provides extensive enterprise-class RAS capabilities. Data in memory is protected with advanced ECC in addition to standard DDR4 RAS features. End-to-end data poisoning ensures corrupted data is tagged and any attempt to use it is flagged as an error. The SLC is also ECC protected, and the processor supports background scrubbing of the SLC cache and DRAM to locate and correct single-bit errors before they accumulate into uncorrectable errors.

## Note

A sample bill of materials, in the *Chapter 9, Appendix*, cites the necessary quantities 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 K3s 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 toward *production*, providing *scaling* guidance that is needed to create the solution.

## 5.1 Deployment overview

The deployment stack is represented in the following figure:

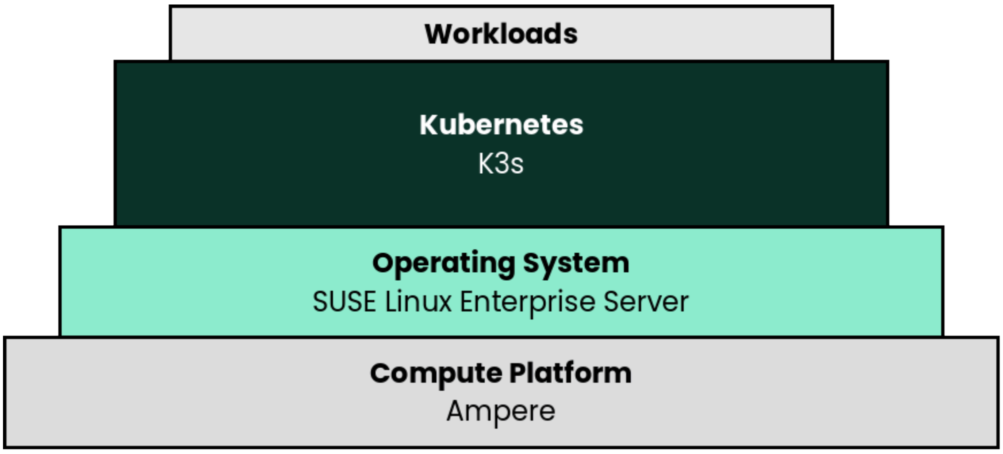


FIGURE 5.1: DEPLOYMENT STACK - K3S

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 Ampere Altra platform. Based upon the relatively small resource requirements for a K3s deployment, one viable approach is to deploy as a virtual machine (VM) on the target nodes, on top of an existing hypervisor, like kernel-based Virtual Machine (KVM). Another option is to use one or more Ampere Altra or Ampere Altra Max baremetal systems for the deployments. For the physical host, there are tools that can be used during the setup of the server (see below).

**Preparation(s)**

If available, the integrated Baseboard Management Controller (BMC) provides remote access to multiple users at different locations for networking. It also allows a system administrator to monitor system health and manage computer events remotely, including media, either virtual or physical) redirection of software image files used for installing operating systems and a console interaction.

**Deployment Process**

On the respective compute module node

1. Given the simplicity of the deployment, the operating system can be
   - installed with the respective SUSE operating system media ISO media,
   - converted to a USB drive, or
   - installed leveraging Preboot Execution Environment (PXE) infrastructure.
2. For deployments targeting virtual machines, determine if a hypervisor is already available or provisioned.
   - 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-SP3/single-html/SLES-virtualization/#book-virt).
   - Then, for the solution VM node, use the hypervisor user interface to allocate the necessary CPU, memory, disk and networking as noted in the SUSE Rancher hardware requirements (https://documentation.suse.com/cloudnative/rancher-manager/latest/en/installation-and-upgrade/requirements/requirements.html).

**Deployment Consideration(s)**

To further optimize deployment factors, leverage the following practices:

- *scaling*

- 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 is beneficial to have the multi-VM deployments spread across multiple hypervisor nodes.
- The consideration of consistent hypervisor and compute module configurations, with the needed resources for the deployed VMs, will yield a robust and reliable production implementation.

## 5.3 SUSE Linux Enterprise Server

As the base software layer, use an enterprise-grade Linux operating system. For example, SUSE Linux Enterprise Server.

**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 used.

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 host names
   - Network Time Protocol (NTP) - an external network-accessible service to obtain and synchronize system times to aid in time stamp 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) ↗ (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-SP3/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:

1. Download the SUSE Linux Enterprise Server (https://www.suse.com/download/sles/) ↗ product (either for the ISO or Virtual Machine image)
   - Identify the appropriate, supported version of SUSE Linux Enterprise Server by reviewing the support matrix for SUSE Rancher (https://www.suse.com/suse-rancher/support-matrix/all-supported-versions/) ↗ versions Web page.
2. The installation process is described and can be performed with default values by following steps from the product documentation, see Installation Quick Start (https://documentation.suse.com/sles/15-SP3/single-html/SLES-installation/#article-installation) ↗

> **Tip**
>
> Adjust both the password and the local network addressing setup to comply with local environment guidelines and requirements.

#### Deployment Consideration(s)

To further optimize deployment factors, leverage the following practices:

- *Automation*
  - To reduce user intervention, unattended deployments of SUSE Linux Enterprise Server can be automated
    - for ISO-based installations, by referring to the AutoYaST Guide (https://documentation.suse.com/sles/15-SP3/single-html/SLES-autoyast/#book-autoyast)

## 5.4 K3s

#### Preparation(s)

1. Identify the appropriate, desired version of the K3s binary (for example vX.YY.ZZ +k3s1) by reviewing
   - the "Supported K3s Versions" associated with the respective SUSE Rancher (https://www.suse.com/suse-rancher/support-matrix/all-supported-versions/) version from "K3s Downstream Clusters" section, or
   - the "Releases" on the Download (https://github.com/k3s-io/k3s/) Web page.
2. For the underlying operating system firewall service, either
   - enable and configure the necessary inbound ports (https://documentation.suse.com/cloudnative/k3s/latest/en/reference/resource-profiling.html) or
   - stop and completely disable the firewall service.

#### Deployment Process

Perform the following steps to install the first K3s server on one of the nodes to be used for the Kubernetes control plane

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 K3s 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*
  - A full high-availability K3s cluster is recommended for production workloads. The etcd key/value store (aka database) requires an odd number of servers (aka master nodes) be allocated to the K3s cluster. In this case, two additional control-plane servers should be added; for a total of three.
    1. Deploy the same operating system on the new compute platform nodes, then log in to 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:
       - Set the following additional variables, as appropriate for this cluster

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

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

# Match the first of the first server
K3s_VERSION=""
```

- Install K3s

```
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 -
```

- Monitor the progress of the installation: `watch -c "kubectl get deployments -A"`
  - The K3s 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

### 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.

- (Optional) In cases where agent nodes are desired, execute the following sets of commands, using the same "*K3s_VERSION*", "*FIRST_SERVER_IP*", and "*NODE_TOKEN*" variable settings as above, on each of the agent nodes to add it to the K3s cluster:

```
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 -
```

After this successful deployment of the K3s solution, review the product documentation (https://documentation.suse.com/cloudnative/k3s/) ↗ for details on how to directly use this Kubernetes cluster. Furthermore, by reviewing the SUSE Rancher product documentation (https://documentation.suse.com/cloudnative/rancher-manager/) ↗ 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 (https://www.suse.com) and the Rancher portfolio plus Ampere Altra (https://amperecomputing.com/) Computing Systems streamline 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

- Using K3s enables you to quickly and simply deploy a Kubernetes cluster in a wide array of locations, across edge, branch, core and cloud.

**Modernize**

Bring applications and data into modern computing

- With K3s, the digital transformation to containerized applications can progress since both developers and production can leverage these deployments for the actual workloads.

**Accelerate**

Accelerate business transformation through the power of open source software

- Given the open source nature of K3s and the minimal underlying software components, you can expand into a very distributed ecosystem, bringing computing to where the data exists or arrives, to answer the necessary business needs.

# 7 References

WHITE PAPERS

- A Buyer's Guide to Enterprise Kubernetes Management Platforms - https://more.suse.com/FY22_Buyers_Guide_to_Enterprise_Container_Management_Buyers-Guide-to-Kubernetes-Management-Platforms.html
- How to Build an Enterprise Kubernetes Strategy - https://more.suse.com/FY22-global-web-How-to-Build-Enterprise-K8s-Strategy.html

BOOKS

- Kubernetes Management - https://more.suse.com/rs/937-DCH-261/images/002022021-DummiesGuide.pdf

TRAINING

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

WEB SITES

- SUSE - https://www.suse.com
    - SUSE Customer Center (SCC) - https://scc.suse.com
    - Products
        - SUSE Rancher - https://www.suse.com/products/rancher/ (documentation (https://documentation.suse.com/cloudnative/rancher-manager/))
        - Rancher Kubernetes Engine (RKE) - https://rancher.com/products/rke/ (documentation (https://rancher.com/docs/rke/latest/en/))
        - K3s - https://www.suse.com/products/k3s/ (documentation (https://documentation.suse.com/cloudnative/k3s/))
        - SUSE Linux Enterprise Micro (SLEMicro) - https://www.suse.com/products/micro/ (documentation (https://documentation.suse.com/sle-micro/5.5/))
        - SUSE Linux Enterprise Server (SLES) - https://www.suse.com/products/server/ (documentation (https://documentation.suse.com/sles/15-SP3/))

    - SUSE Manager - https://www.suse.com/products/suse-manager/ (documentation (https://documentation.suse.com/suma/4.3/))
    - SUSE Repository Mirroring Tool (RMT) - https://www.suse.com/products/server/ (documentation (https://documentation.suse.com/sles/15-SP3/single-html/SLES-rmt/#book-rmt))
  - Projects
    - Rancher Kubernetes Engine Government (RKE2) - https://github.com/rancher/rke2 (documentation (https://docs.rke2.io/))
- Ampere Computing - https://amperecomputing.com/
  - Ampere Altra - https://amperecomputing.com/processors/ampere-altra/

# 8 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 stand-alone 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 toward 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 failure or downtime, 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.

Performance[5]

In the context of a system's expected life cycle, performance is an assessment of transactions, responsiveness and underlying stability of the provider technology while doing tuning and adjustments. Other risk factors and discerning potential impacts to surrounding use cases are also integral parts of the profile to address beyond service levels, capacity and problem management.

Security[6]

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)

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

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

5 link: Performance Engineering (https://en.wikipedia.org/wiki/Performance_engineering)

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

**Proof-of-Concept**[7]

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[8]) that has just enough features to satisfy an initial set of requests. After such insights and feedback are obtained and potentially addressed, redeployments may be used 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 usage 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.

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

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

# 9 Appendix

The following sections provide a bill of materials listing for the respective component layer(s) of the described deployment.

## 9.1 Compute platform bill of materials

Sample set of computing platform models, components and resources.

| Role | Qty | SKU | Component | Notes |
|---|---|---|---|---|
| Example 1 | 1-3 | | 1U Single Socket Rack Server | • items below listed per node<br>• skipped SKUs, see Compute Platform (https://amperecomputing.com/reference-platforms/ampere-altra-platforms-for-modern-compute/) |

| Role | Qty | SKU | Component | Notes |
|---|---|---|---|---|
| | 1 | | • Ampere Altra Max Series Processor, 96 to 128 Cores or<br>• Ampere Altra Series Processor, 32 to 80 Cores | |
| | 2-16 | | • DIMMs (8-channel, 2DPC) Up to DDR4-3200 Up to 4TB | |
| | 1-2 | | • Gen4 x4, M.2 NGFF 22110/2280 | |
| | 1-6 | | • 2.5" U.2 or 2.5" SAS/SATA | |
| | 1 | | • i350 Dual 1GbE | |
| Example 2 | 1-3 | | 2U Single/Dual Socket Rack Server | • items below listed per node<br>• skipped SKUs, see Compute Platform (https://amperecomputing.com/reference-platforms/ampere-al- |

| Role | Qty | SKU | Component | Notes |
|---|---|---|---|---|
| | | | | tra-plat-forms-for-mod-ern-com-pute/) |
| | 1-2 | | • Ampere Altra Max Series Processor, 96 to 128 Cores or<br>• Ampere Altra Series Processor, 32 to 80 Cores | |
| | 2-32 | | • DIMMs (8-chan-nels 2DPC per CPU) Up to DDR4-3200 Up to 8TB | |
| | 1-2 | | • 2.5” SAS/SA-TA/NVMe | |
| | 1-24 | | • 2.5” SAS/SATA | |
| | 1 | | • i350 Dual 1GbE | |

## 9.2 Software bill of materials

Sample set of software, support and services.

| Role | Qty | SKU | Component | Notes |
|---|---|---|---|---|
| Operating System | 1-3 | 874-006875 | SUSE Linux Enterprise Server,<br>• x86_64,<br>• Priority Subscription,<br>• 1 Year | Configuration:<br>• per node (up to 2 sockets, stackable) or 2 VMs |
| Kubernetes Management | 1 | R-0001-PS1 | SUSE Rancher,<br>• x86-64,<br>• Priority Subscription,<br>• 1 Year | Configuration:<br>• per deployed instance |
| Rancher Management | 2 | R-0004-PS1 | Rancher 10 Nodes<br>• x86-64 or aarch64,<br>• Priority Subscription,<br>• 1 Year, | Configuration:<br>• requires priority server subscription |
| Consulting and Training | 1 | R-0001-QSO | Rancher Quick Start,<br>• Go Live Services | |

**Note**

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

## 9.3 Documentation configuration / attributes

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

```
Appendix=1 ArchOv=1 Automation=1 Availability=1 BP=1 BPBV=1 CompMod=1 DepConsiderations=1
 Deployment=1 FCTR=1 FLVR=1 GFDL=1 Glossary=1 HWComp=1 HWDepCfg=1 IHV-Ampere-Altra=1
 IHV-Ampere=1 Integrity=1 LN=1 Performance=1 PoC=1 Production=1 RA=1 RC=1 References=1
 Requirements=1 SWComp=1 SWDepCfg=1 Scaling=1 Security=1 docdate=2022-08-29 env-daps=1
 focusK3s=1 iIHV=1 iK3s=1 iRKE1=1 iRKE2=1 iRMT=1 iRancher=1 iSLEMicro=1 iSLES=1 iSUMa=1
 layerSLES=1
```

# 10 Legal Notice

# 11 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.