SLE BCI

NVIDIA GPU driver With SUSE Linux Enterprise Base Container Image on Rancher Kubernetes Engine 2

Creating and deploying an NVIDIA GPU driver on SUSE Linux Enterprise Base Container Image on an RKE2 Kubernetes cluster with Helm

SUSE Linux Enterprise Base Container Image

Alex Arnoldy, Embedded Solutions Architect (SUSE Alliance Architects)

NVIDIA GPU driver With SUSE Linux Enterprise Base Container Image on Rancher Kubernetes Engine 2

Creating and deploying an NVIDIA GPU driver on SUSE Linux Enterprise Base Container Image on an RKE2 Kubernetes cluster with Helm

Date: 2023-09-12

Summary

This document describes the process of creating an NVIDIA GPU Driver based on SLE BCI and deploying it an NVIDIA GPU Operator on the Rancher Kubernetes Engine 2 distribution of Kubernetes.

Disclaimer

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Contents

1 Introduction

This Getting Started guide provides comprehensive instructions for two imporant tasks. The first is the creation of an OCI compliant container image based on the SUSE Linux Enterprise Base Container Image, that runs the NVIDIA GPU driver. The second is deploying the container image on Rancher Kubernetes Engine 2. The primary objective is to enable you to seamlessly integrate the NVIDIA GPU Operator, simplifying GPU management and support within Kubernetes clusters for GPU-intensive workloads.

The choice of SUSE Linux Enterprise Base Container Image is motivated by the unparalleled security certifications and enhanced supportability it offers, particularly when operating heterogeneous software stacks. SUSE's SUSE Linux Enterprise Base Container Image is distinguished by a wide range of security certifications, including Common Criteria, FIPS, and EAL, making it a trusted choice for organizations with stringent security requirements.

2 Scope

This guide will help you:

- Build an OCI compliant container image that incorporate an appropriate NVIDIA GPU driver installed into a SUSE Linux Enterprise Base Container Image.
- Validate the functionality of that container image.
- Push the image to a central container image registry so that it can be accessed by a RKE2 Kubernetes cluster.
- Deploy the NVIDIA GPU Operator Helm chart to an RKE2 Kubernetes cluster in a way that leverages the NVIDIA GPU driver container image.
- Verify the Helm installation process completed correctly.

Tip

SUSE always recommends you use the most current Service Pack of SUSE Linux Enterprise Server that is available.

Important

It is assumed that you are using Data Center class NVIDIA GPU(s). Integrating consumer grade GPUs is beyond the scope of this document.

2.1 Audience

This guide is intened for an audience comprising Kubernetes administrators, proficient DevOps practitioners, and application developers. It assumes a foundational understanding of Podman and/or Docker, Kubernetes, and NVIDIA GPU technologies. This guide should be suitable for most high technology professionals seeking to unlock the full potential of their GPU accellerated containerized applications.

3 Prerequisites

BEFORE EMBARKING ON THE PROCEDURES OUTLINED IN THIS GUIDE, YOU SHOULD ENSURE THE FOLLOWING PREREQUISITES HAVE BEEN MET

- You have access to a SLES15 SP5 build host. You should ensure the build host is the same Service Pack (e.g. 15 SP5) version as the SLE BCI image to be used.

Tip

The build host does not require access to an NVIDIA GPU.

- The SLES Containers Module has been enabled on the build host and the SLES Containers Module plus the NVIDIA Compute Module have been enabled on all Kubernetes worker nodes. You can use the following commands to enable the appropriate software modules. on each host/node.

  Note

  Change the variable SLE15_SP_VERSION in the following command to match the service pack release of your SUSE Linux Enterprise Server Kubernetes worker nodes.

```
export SLE15_SP_VERSION=5
```

```
sudo SUSEConnect -p sle-module-containers/15.${SLE15_SP_VERSION}/x86_64
sudo SUSEConnect -p sle-module-desktop-applications/15.${SLE15_SP_VERSION}/x86_64
sudo SUSEConnect -p sle-module-development-tools/15.${SLE15_SP_VERSION}/x86_64
sudo SUSEConnect -p sle-module-NVIDIA-compute/15/x86_64
```

- You have installed the required SUSE Linux Enterprise Server NVIDIA software packages on each Kubernetes worker node that is configured with an NVIDIA GPU. You can use the following commands to install the appropriate software.

```
sudo zypper install \
  kernel-firmware-nvidia \
  libnvidia-container-tools \
  libnvidia-container1 \
  nvidia-container-runtime \
  sle-module-NVIDIA-compute-release
```

- Podman is installed on the build host and at least one of the NVIDIA GPU equiped Kubernetes worker nodes. You can use the following command to install Podman.

```
sudo zypper install podman
```

- The git utility is installed on the build host. You can use the following command to install git.

```
sudo zypper install git-core
```

- You have access to a container image registry that is available to the build host AND the target Kubernetes cluster. This will allow the NVIDIA GPU driver to be deployed across all Kubernetes worker nodes that have NVIDIA GPUs, when installing the NVIDIA GPU Operator Helm chart.

> **Note**
>
> The container registry does not need to support authenication, but should be configured with a valid TLS certificate.

- You have access to a Kubernetes cluster that is equipped and correctly configured with Data Center class NVIDIA GPUs. These instructions leverage SUSE's security focused Kubernetes distribution, RKE2.

- You possess a basic familiarity with Podman and/or Docker, Kubernetes, and NVIDIA GPU concepts.

4 Technical overview

The process of building the OCI compliant NVIDIA GPU driver container image using SLE BCI and deploying it to an RKE2 cluster will leverage the following primary components:

+

- A build host. This can be an RKE2 worker node or a separate node or VM with, or without access to an NVIDIA GPU.
  - If the build host has access to an NVIDIA GPU, the specified validations can be done on that node.
- A container image registry to which the build host and the RKE2 cluster can access.
- An RKE2 Kubernetes cluster with at least one worker node that is confgiured with an NVIDIA GPU.
- Access to the SUSE Linux Enterprise Server software modules through the SUSE Customer Center or an RMT server.

The process of building the OCI compliant container image and deploying it to an RKE2 cluster will entail the following steps:

+

1. Build the container image
2. Validate the state of the image
3. Save it to a container image registry so it can be accessed by the RKE2 cluster
4. Deploy the entire NVIDIA GPU Operator Helm chart, including the NVIDIA GPU driver image to the RKE2 cluster
5. Validate the state of the NVIDIA GPU Operator installation

5 Set the command variables

Tip

These variables will be consumed throughout this procedure.

> **Important**
>
> The following commands must be run on the SUSE Linux Enterprise Server build host.

1. Create the /tmp/.build-variables.sh file:

> **Tip**
>
> For the REGISTRY variable, provide the URL of the registry where the new image will be saved.
>
> For the SLE15_SP_VERSION, enter the number for the SLES 15 SP5 service pack to be used for the container image (i.e. 5 for SP5).
>
> For the DRIVER_VERSION variable, provide the NVIDIA GPU driver version (e.g. 535.104.05). Find the latest "Data Center Driver for Linux x64" version for your GPU at https://www.nvidia.com/download/index.aspx.
>
> For the OPERATOR_VERSION variable, provide the NVIDIA GPU Operator version (e.g.v23.6.1). Find the associated NVIDIA GPU Operator version at https://docs.nvidia.com/datacenter/cloud-native/gpu-operator/latest/platform-support.html.
>
> For the CUDA_VERSION variable, provide the required CUDA version (e.g. 12.2.2) for NVIDIA GPU driver by selecting your driver version at https://docs.nvidia.com/datacenter/cloud-native/gpu-operator/latest/platform-support.html. The CUDA version will be listed under "Software Versions".

```
cat <<EOF> /tmp/.build-variables.sh
export REGISTRY=""
export SLE15_SP_VERSION=""
export DRIVER_VERSION=""
export OPERATOR_VERSION=""
export CUDA_VERSION=""
EOF
```

2. Edit the /tmp/.build-variables.sh file to provide the appropriate values.

> **Tip**
>
> Disconnecting from your current terminal environment will cause the variables to be lost. Repeat the following step to set the variables again.

> **Important**
>
> After setting and sourcing the variables, you will be able to copy and paste the entire code blocks from this document to the command line and execute them without editing. For the best results, be sure and execute each code block in its entirety as a single command set.

3. Source the variables into your current terminal environment:

```
source /tmp/.build-variables.sh
```

6 Build the container image

This section offers a detailed explanation of the steps required to create OCI compliant container images with the NVIDIA GPU Operator inside the SUSE Linux Enterprise Base Container Image.

1. Clone the NVIDIA GitLab repository and change to the `driver/sle15` directory:

```
git clone {nvidia-git-lab-url} && cd {nvidia-git-lab-directory}
```

2. Update the Dockerfile in this directory:

3. Make a backup copy of the Dockerfile before modifying:

```
cp Dockerfile /tmp/Dockerfile.orig
```

4. Update the golang build container image to version `1.18`:

```
sed -i '/^FROM/ s/golang\:1\.../golang\:1.18/' Dockerfile
```

5. Update the Dockerfile's base container image to the SLES 15 SP5 BCI:

```
sed -i '/^FROM/ s/suse\/sle15/bci\/bci-base/' Dockerfile
```

6. Verify the changes that have been made to the Dockerfile:

```
diff /tmp/Dockerfile.orig Dockerfile
```

7. Build the SLES container image with the NVIDIA GPU driver installed:

   a. Build the container:

      > **Important**
      >
      > When building the container image, you may be prompted for the registry that contains the `nvidia/cuda` image. If so, select the image located in docker.io.

      > **Note**
      >
      > If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
## Validate the variables before using them in the subsequent command
echo &&
echo "
REGISTRY=${REGISTRY}
SLE15_SP_VERSION=${SLE15_SP_VERSION}
DRIVER_VERSION=${DRIVER_VERSION}
CUDA_VERSION=${CUDA_VERSION}" && echo && read -n1 -p "Press CTRL+C now if these
 variables are NOT correct, otherwise press Enter" BAILOUT &&

## Build the container
sudo podman build -t \
${REGISTRY}/nvidia-sle15sp${SLE15_SP_VERSION}-${DRIVER_VERSION}:
${DRIVER_VERSION} \
  --build-arg SLES_VERSION="15.${SLE15_SP_VERSION}" \
  --build-arg DRIVER_ARCH="x86_64" \
  --build-arg DRIVER_VERSION="${DRIVER_VERSION}" \
  --build-arg CUDA_VERSION="${CUDA_VERSION}" \
  --build-arg PRIVATE_KEY=empty  \
.
```

   b. Watch the build output for errors, warnings, and failures. You can safely ignore errors and warnings that don't stop the build process.

   c. The build process should finish with a message saying that the final image was committed and tagged. For example:

```
 COMMIT registry.susealliances.com/nvidia-sle15sp5-535.104.05
--> cf976870489
```

```
Successfully tagged registry.susealliances.com/nvidia-
sle15sp5-535.104.05:latest
cf9768704892c4b8b9e37a4ef591472e121b81949519204811dcc37d2be9d16c
```

8. Remove the intermediate build container image that was created as part of the build process (and any other leftover artifacts):

```
for EACH in $(sudo podman images | awk '/none/ {print$3}'); do sudo podman rmi
 ${EACH}; done
```

9. Push the newly build image to the container registry:

Important

If the target container registry requires authentication, use the `Podman login` command to successfully authenticate before continuing. See https://docs.podman.io/en/latest/markdown/podman-login.1.html for more information.

Note

If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
## Validate the variables before using them in the subsequent command
echo &&
echo "
REGISTRY=${REGISTRY}
SLE15_SP_VERSION=${SLE15_SP_VERSION}
DRIVER_VERSION=${DRIVER_VERSION}" && echo && read -n1 -p "Press CTRL+C now if these
 variables are NOT correct, otherwise press Enter" BAILOUT &&

## Tag the image with the format that Helm will need when deploying on Kubernetes
sudo podman tag ${REGISTRY}/nvidia-sle15sp${SLE15_SP_VERSION}-${DRIVER_VERSION}:
${DRIVER_VERSION} ${REGISTRY}/driver:${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION} &&
## Push the image (with both tags) to the container registry
sudo podman push ${REGISTRY}/nvidia-sle15sp${SLE15_SP_VERSION}-${DRIVER_VERSION}:
${DRIVER_VERSION} &&
sudo podman push ${REGISTRY}/driver:${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION}
```

a. Verify the image is saved in the registry, and remotely available:

> **Note**
>
> If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
sudo podman search --list-tags ${REGISTRY}/driver:${DRIVER_VERSION}-sles15.
${SLE15_SP_VERSION}
```

10. Validate the container image

> **Note**
>
> This step is optional and requires running the newly created NVIDIA GPU driver container locally with `Podman`, outside of the Kubernetes context. This can be done on a 15 SP5 host configured with the same kind of NVIDIA GPU the container was created for, or on a Kuberentes worker node that is configured with an NVIDIA GPU.

a. Open a command line session to the host or Kubernetes worker node on which you will test the container image.

b. Create the `/run/nvidia` directory, if it does not yet exist:

```
sudo mkdir -p /run/nvidia
```

c. Run the NVIDIA GPU driver container locally:

> **Note**
>
> If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
## Validate the variables before using them in the subsequent command
echo &&
echo "
REGISTRY=${REGISTRY}
SLE15_SP_VERSION=${SLE15_SP_VERSION}
```

```
DRIVER_VERSION=${DRIVER_VERSION}" && echo && read -n1 -p "Press CTRL+C now if
 these variables are NOT correct, otherwise press Enter" BAILOUT && \

## Run the container image
sudo podman run -d \
  --name driver.${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION} \
  --privileged \
  --pid=host \
  -v /run/nvidia:/run/nvidia:shared \
  -v /var/log:/var/log \
  --restart=unless-stopped \
${REGISTRY}/driver:${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION}
```

d. Verify the container is running:

```
sudo podman ps -a
```

i. The container's `STATUS` field should show that it is "Up" and the amount of time it has been up should increment with repeated runs of the command.

e. Monitor the deployment of the NVIDIA GPU driver:

Note

If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
## Validate the variables before using them in the subsequent command
echo &&
echo "
REGISTRY=${REGISTRY}
SLE15_SP_VERSION=${SLE15_SP_VERSION}
DRIVER_VERSION=${DRIVER_VERSION}" && echo && read -n1 -p "Press CTRL+C now if
 these variables are NOT correct, otherwise press Enter" BAILOUT && \

## Review the standard output of the running container
sudo podman logs -f driver.${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION}
```

f. The deployment process is complete when the following message is shown:

Mounting NVIDIA driver rootfs… Done, now waiting for signal

g. Press `CTRL+C` to close the log viewing session

h. Ensure the NVIDIA kernel modules have been loaded:

```
sudo lsmod | grep nvidia
```

i. You should see modules such as `nvidia`, `nvidia_modeset`, and `nvidia_uvm`

i. Verify the `nvidia-smi` utility can communicate withe the GPU:

Note

If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
## Validate the variables before using them in the subsequent command
echo &&
echo "
REGISTRY=${REGISTRY}
SLE15_SP_VERSION=${SLE15_SP_VERSION}
DRIVER_VERSION=${DRIVER_VERSION}" && echo && read -n1 -p "Press CTRL+C now if
 these variables are NOT correct, otherwise press Enter" BAILOUT && \

## Verify the nvidia-smi utitlity can communicate with the GPU
sudo podman exec -it driver.${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION}
 nvidia-smi
```

j. When ready to move forward, stop and remove the Podman container:

Note

If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in this process before continuing.

```
## Validate the variables before using them in the subsequent command
echo &&
echo "
SLE15_SP_VERSION=${SLE15_SP_VERSION}
DRIVER_VERSION=${DRIVER_VERSION}" && echo && read -n1 -p "Press CTRL+C now if
 these variables are NOT correct, otherwise press Enter" BAILOUT &&

## Stop and remove the container instance
sudo podman stop driver.${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION} &&
```

```
sudo podman rm driver.${DRIVER_VERSION}-sles15.${SLE15_SP_VERSION}
```

k. Before continuing to the Kubernetes deployment procedure, ensure the NVIDIA kernel modules are not loaded on any of the NVIDIA GPU equipped Kuberenetes worker nodes:

```
sudo lsmod | grep nvidia
```

i. You should receive no output.

If you see any modules containing the name `nvidia`, use the command `sudo modprobe -r <module name>` to unload them. If any modules fail to unload, reboot the node.

7 Deploy to a Kubernetes cluster

Note

The preferred method for installing the NVIDIA GPU Operator is with the Helm Kuberenetes package manager.

Important

The following steps must be run from a Linux system that has the kubectl and Helm (version 3) utilities, as well as the KUBECONFIG file for the target Kubernetes cluster available to it. See these documents for more information: https://www.suse.com/c/rancher_blog/how-to-manage-kubernetes-with-kubectl/ and https://docs.rke2.io/cluster_access.

In addition, if the container build host is a different system than the one being used to perform the Helm install, the /tmp/.build-variables.sh file will need to be created on the second system. Return to *Section 5, "Set the command variables"* in the proceeding procedure before continuing.

1. Add the NVIDIA helm software repository:

```
helm repo add https://helm.ngc.nvidia.com/nvidia
```

```
helm repo update
```

2. Deploy the NVIDIA GPU Operator with Helm:

Note

If any of the following variables are not set correctly, press `CTRL+C` and return to *Section 5, "Set the command variables"* in the proceeding procedure before continuing.

```
## Verify the selected cluster before deploying
echo &&
echo "Cluster name: $(kubectl config current-context)" &&
echo "" &&
kubectl get nodes -o wide &&
echo "" &&
read -n1 -p "Is this the target Kubernetes cluster for the Helm chart? (y/n) " YESNO
 &&
echo "" &&
[ ${YESNO} != y ] && { echo "Exiting."; echo ""; exit; } || echo "" &&

## Validate the variables before using them in the subsequent command
echo &&
echo "
REGISTRY=${REGISTRY}
SLE15_SP_VERSION=${SLE15_SP_VERSION}
OPERATOR_VERSION=${OPERATOR_VERSION}
DRIVER_VERSION=${DRIVER_VERSION}" && echo && read -n1 -p "Press CTRL+C now if these
 variables are NOT correct, otherwise press Enter" BAILOUT &&

## Deploy the Helm chart
helm install -n gpu-operator \
  --generate-name \
  --wait \
  --create-namespace \
  --version=${OPERATOR_VERSION} \
    nvidia/gpu-operator \
  --set driver.repository=${REGISTRY} \
  --set driver.version=${DRIVER_VERSION} \
  --set operator.defaultRuntime=containerd \
  --set toolkit.env[0].name=CONTAINERD_CONFIG \
  --set toolkit.env[0].value=/var/lib/rancher/rke2/agent/etc/containerd/config.toml
 \
  --set toolkit.env[1].name=CONTAINERD_SOCKET \
  --set toolkit.env[1].value=/run/k3s/containerd/containerd.sock \
  --set toolkit.env[2].name=CONTAINERD_RUNTIME_CLASS \
```

```
  --set toolkit.env[2].value=nvidia \
  --set toolkit.env[3].name=CONTAINERD_SET_AS_DEFAULT \
  --set-string toolkit.env[3].value=true
```

3. Verify the NVIDIA GPU Operator, NVIDIA GPU driver and associated elements have been deployed correctly with the command:

```
kubectl get pods -n gpu-operator
```

   a. The output should be similar to the following:

```
NAME                                                          READY   STATUS
   RESTARTS   AGE

gpu-feature-discovery-crrsq                                   1/1     Running
   0          60s

gpu-operator-7fb75556c7-x8spj                                 1/1     Running
   0          5m13s

gpu-operator-node-feature-discovery-master-58d884d5cc-w7q7b   1/1     Running
   0          5m13s

gpu-operator-node-feature-discovery-worker-6rht2              1/1     Running
   0          5m13s

gpu-operator-node-feature-discovery-worker-9r8js              1/1     Running
   0          5m13s

nvidia-container-toolkit-daemonset-lhgqf                      1/1     Running
   0          4m53s

nvidia-cuda-validator-rhvbb                                   0/1     Completed
   0          54s

nvidia-dcgm-5jqzg                                             1/1     Running
   0          60s

nvidia-dcgm-exporter-h964h                                    1/1     Running
   0          60s

nvidia-device-plugin-daemonset-d9ntc                          1/1     Running
   0          60s

nvidia-device-plugin-validator-cm2fd                          0/1     Completed
   0          48s
```

```
nvidia-driver-daemonset-5xj6g                                   1/1     Running
   0          4m53s

nvidia-mig-manager-89z9b                                        1/1     Running
   0          4m53s

nvidia-operator-validator-bwx99                                 1/1     Running
   0          58s
```

8 Validation

> **Note**
>
> The NVIDIA GPU Operator Helm chart provides two pods that validate the state of the installed software.

1. Validate the state of the NVIDIA GPU Operator software:

```
kubectl logs -n gpu-operator -l app=nvidia-operator-validator
```

   a. The output should be similar to:

      Defaulted container "nvidia-operator-validator" out of: nvidia-operator-validator, driver-validation (init), toolkit-validation (init), cuda-validation (init), plugin-validation (init)

      all validations are successful

2. Validate the state of the NVIDIA CUDA driver software:

```
kubectl logs -n gpu-operator -l app=nvidia-cuda-validator
```

   a. The output should be similar to the following:

      Defaulted container "nvidia-cuda-validator" out of: nvidia-cuda-validator, cuda-validation (init)

cuda workload validation is successful

3. To validate that the NVIDIA GPU driver is communicating with the GPU, you can run this command to view the statics of the Kubernetes workers that are configured with GPUs:

```
kubectl exec -it \
"$(for EACH in \
$(kubectl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done)" \
-n gpu-operator \
nvidia-smi
```

> **Note**
>
> This command can also be used to verify which application processes are running on the NVIDIA GPUs, and how many resources are being consumed.

9 Summary

This guide has effectively steered the creation of OCI compliant container images leveraging the SUSE Linux Enterprise Base Container Image and incorporating the NVIDIA GPU driver. Furthermore, it has provided coherent instructions for validating the functionality of the container image and the seamless deployment of the image within a Kubernetes cluster, specifically RKE2.

This integrated solution is aimed at affording containerized applications GPU-acceleration, while avoiding the need to manage additional software on each GPU equiped node. The strategic choice of SUSE's SUSE Linux Enterprise Base Container Image as the foundation for this integration underscores an organization's commitment to security, certifications, and supportability.

Organizations with exacting security requirements depend on SUSE Linux Enterprise Server's numerous certifications such as Common Criteria, FIPS, and EAL. Additionally, SUSE's commitment to providing robust support for heterogeneous software stacks guarantees customer's the freedom to design their IT landscape to suit their unique business challenges.

A pivotal point to underscore is the indispensability of a Kubernetes cluster, preferably RKE2, that provides full NVIDIA GPU support and the requisite NVIDIA GPU Operator to fully harness GPU resources when deploying GPU intensive workloads.

10 Legal notice

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

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.