SUSE Linux Enterprise Micro 5.2

Administration Guide

This guide describes the administration of SUSE Linux Enterprise Micro.

Publication Date: September 28, 2023

Contents

1 Snapshots

Warning: Snapshots are mandatory

As snapshots are crucial for the correct functioning of SLE Micro, do not disable the feature, and ensure that the root partition is big enough to store the snapshots.

When a snapshot is created, both the snapshot and the original point to the same blocks in the file system. So, initially a snapshot does not occupy additional disk space. If data in the original file system is modified, changed data blocks are copied while the old data blocks are kept for the snapshot.

Snapshots always reside on the same partition or subvolume on which the snapshot has been taken. It is not possible to store snapshots on a different partition or subvolume. As a result, partitions containing snapshots need to be larger than partitions which do not contain snapshots. The exact amount depends strongly on the number of snapshots you keep and the amount of data modifications. As a rule of thumb, give partitions twice as much space as you normally would. To prevent disks from running out of space, old snapshots are automatically cleaned up.

Snapshots that are known to be working properly are marked as *important*.

1.1 Directories excluded from snapshots

As some directories store user-specific or volatile data, these directories are excluded from snapshots:

`/home`
: Contains users' data. Excluded so that the data will not be included in snapshots and thus potentially overwritten by a rollback operation.

`/root`
: Contains root's data. Excluded so that the data will not be included in snapshots and thus potentially overwritten by a rollback operation.

`/opt`
: Third-party products usually get installed to `/opt`. Excluded so that these applications are not uninstalled during rollbacks.

`/srv`

Contains data for Web and FTP servers. Excluded in order to avoid data loss on rollbacks.

`/usr/local`

This directory is used when manually installing software. It is excluded to avoid uninstalling these installations on rollbacks.

`/var`

This directory contains many variable files, including logs, temporary caches, third-party products in `/var/opt`, and is the default location for virtual machine images and databases. Therefore, a separate subvolume is created with Copy-On-Write disabled, so as to exclude all of this variable data from snapshots.

`/tmp`

The directory contains temporary data.

the architecture-specific `/boot/grub2` directory

Rollback of the boot loader binaries is not supported.

1.2 Showing exclusive disk space used by snapshots

Snapshots share data, for efficient use of storage space, so using ordinary commands like **`du`** and **`df`** won't measure used disk space accurately. When you want to free up disk space on Btrfs with quotas enabled, you need to know how much exclusive disk space is used by each snapshot, rather than shared space. The **`btrfs`** command provides a view of space used by snapshots:

```
# btrfs qgroup show -p /
qgroupid         rfer         excl parent
--------         ----         ---- ------
0/5          16.00KiB     16.00KiB ---
[...]
0/272         3.09GiB     14.23MiB 1/0
0/273         3.11GiB    144.00KiB 1/0
0/274         3.11GiB    112.00KiB 1/0
0/275         3.11GiB    128.00KiB 1/0
0/276         3.11GiB     80.00KiB 1/0
0/277         3.11GiB    256.00KiB 1/0
0/278         3.11GiB    112.00KiB 1/0
0/279         3.12GiB     64.00KiB 1/0
0/280         3.12GiB     16.00KiB 1/0
1/0           3.33GiB    222.95MiB ---
```

The `qgroupid` column displays the identification number for each subvolume, assigning a qgroup level/ID combination.

The `rfer` column displays the total amount of data referred to in the subvolume.

The `excl` column displays the exclusive data in each subvolume.

The `parent` column shows the parent qgroup of the subvolumes.

The final item, `1/0`, shows the totals for the parent qgroup. In the above example, 222.95 MiB will be freed if all subvolumes are removed. Run the following command to see which snapshots are associated with each subvolume:

```
# btrfs subvolume list -st /
```

2 Administration using transactional updates

SLE Micro was designed to use a read-only root file system. This means that after the deployment is complete, you are not able to perform direct modifications to the root file system, e.g. by using **`zypper`**. Instead, SUSE Linux Enterprise Micro introduces the concept of transactional updates which enables you to modify your system and keep it up to date.

The key features of transactional updates are the following:

- They are atomic - the update is applied only if it completes successfully.
- Changes are applied in a separate snapshot and so do not influence the running system.
- Changes can easily be rolled back.

Each time you call the **`transactional-update`** command to change your system—either to install a package, perform an update or apply a patch—the following actions take place:

PROCEDURE 1: MODIFYING THE ROOT FILE SYSTEM

1. A new read-write snapshot is created from your current root file system, or from a snapshot that you specified.
2. All changes are applied (updates, patches or package installation).
3. The snapshot is switched back to read-only mode.
4. The new root file system snapshot is prepared, so that it will be active after you reboot.
5. After rebooting, the new root file system is set as the default snapshot.

> **Note**
>
> Bear in mind that without rebooting your system, the changes will not be applied.

> **Warning**
>
> In case you do not reboot your machine before performing further changes, the **transactional-update** command will create a new snapshot from the current root file system. This means that you will end up with several parallel snapshots, each including that particular change but not changes from the other invocations of the command. After reboot, the most recently created snapshot will be used as your new root file system, and it will not include changes done in the previous snapshots.

2.1 `transactional-update` usage

The **transactional-update** command enables atomic installation or removal of updates; updates are applied only if all of them can be successfully installed. **transactional-update** creates a snapshot of your system and use it to update the system. Later you can restore this snapshot. All changes become active only after reboot.

The **transactional-update** command syntax is as follows:

```
transactional-update [option] [general_command] [package_command] standalone_command
```

> **Note: Running `transactional-update` without arguments.**
>
> If you do not specify any command or option while running the **transactional-update** command, the system updates itself.

Possible command parameters are described further.

`transactional-update` OPTIONS

`--interactive, -i`
: Can be used along with a package command to turn on interactive mode.

`--non-interactive, -n`
: Can be used along with a package command to turn on non-interactive mode.

`--continue [number], -c`

The **`--continue`** option is for making multiple changes to an existing snapshot without rebooting.

The default **`transactional-update`** behavior is to create a new snapshot from the current root file system. If you forget something, such as installing a new package, you have to reboot to apply your previous changes, run **`transactional-update`** again to install the forgotten package, and reboot again. You cannot run the **`transactional-update`** command multiple times without rebooting to add more changes to the snapshot, because this will create separate independent snapshots that do not include changes from the previous snapshots.

Use the **`--continue`** option to make as many changes as you want without rebooting. A separate snapshot is made each time, and each snapshot contains all the changes you made in the previous snapshots, plus your new changes. Repeat this process as many times as you want, and when the final snapshot includes everything you want, reboot the system, and your final snapshot becomes the new root file system.

Another useful feature of the **`--continue`** option is you may select any existing snapshot as the base for your new snapshot. The following example demonstrates running **`transactional-update`** to install a new package in a snapshot based on snapshot 13, and then running it again to install another package:

```
# transactional-update pkg install package_1
```

```
# transactional-update --continue 13 pkg install package_2
```

`--no-selfupdate`

Disables self updating of **`transactional-update`**.

`--drop-if-no-change, -d`

Discards the snapshot created by **`transactional-update`** if there were no changes to the root file system. If there are some changes to the `/etc` directory, those changes merged back to the current file system.

`--quiet`

The **`transactional-update`** command will not output to `stdout`.

`--help, -h`

Prints help for the **`transactional-update`** command.

`--version`

Displays the version of the **`transactional-update`** command.

The general commands are the following:

GENERAL COMMANDS

`cleanup-snapshots`
: The command marks all unused snapshots that are intended to be removed.

`cleanup-overlays`
: The command removes all unused overlay layers of `/etc`.

`cleanup`
: The command combines the `cleanup-snapshots` and `cleanup-overlays` commands. For more details refer to *Section 2.2, “Snapshots cleanup”*.

`grub.cfg`
: Use this command to rebuild the GRUB boot loader configuration file.

`bootloader`
: The command reinstall the boot loader.

`initrd`
: Use the command to rebuild `initrd`.

`kdump`
: In case you perform changes to your hardware or storage, you may need to rebuild the kdump initrd.

`shell`
: Opens a read-write shell in the new snapshot before exiting. The command is typically used for debugging purposes.

`reboot`
: The system reboots after the transactional-update is complete.

`run <command>`
: Runs the provided command in a new snapshot.

`setup-selinux`
: Installs and enables targeted SELinux policy.

The package commands are the following:

Important: Installing packages outside of the official SLE Micro repositories

The installation of packages from repositories other than the official ones (for example, the SUSE Linux Enterprise Server repositories) is **not** supported and not recommended. To use the tools available for SUSE Linux Enterprise Server, run the `toolbox` container and install the tools inside the container. For details about the `toolbox` container, refer to *Section 5, "`toolbox` for SLE Micro debugging"*.

PACKAGE COMMANDS

`dup`
: Performs upgrade of your system. The default option for this command is `--non-interactive`.

`migration`
: The command migrates your system to a selected target. Typically it is used to upgrade your system if it has been registered via SUSE Customer Center.

`patch`
: Checks for available patches and installs them. The default option for this command is `--non-interactive`.

`pkg install`
: Installs individual packages from the available channels using the **`zypper install`** command. This command can also be used to install Program Temporary Fix (PTF) RPM files. The default option for this command is `--interactive`.

```
# transactional-update pkg install package_name
```

or

```
# transactional-update pkg install rpm1 rpm2
```

`pkg remove`
: Removes individual packages from the active snapshot using the **`zypper remove`** command. This command can also be used to remove PTF RPM files. The default option for this command is `--interactive`.

```
# transactional-update pkg remove package_name
```

`pkg update`

Updates individual packages from the active snapshot using the **`zypper update`** command. Only packages that are part of the snapshot of the base file system can be updated. The default option for this command is `--interactive`.

```
# transactional-update pkg update package_name
```

`register`

The register command enables you to register/deregister your system. For a complete usage description, refer to *Section 2.1.1, "The `register` command"*.

`up`

Updates installed packages to newer versions. The default option for this command is `--non-interactive`.

The standalone commands are the following:

STANDALONE COMMANDS

`rollback` *<snapshot number>*

This sets the default subvolume. The current system is set as the new default root file system. If you specify a number, that snapshot is used as the default root file system. On a read-only file system, it does not create any additional snapshots.

```
# transactional-update rollback snapshot_number
```

`rollback last`

This command sets the last known to be working snapshot as the default.

`status`

This prints a list of available snapshots. The currently booted one is marked with an asterisk, the default snapshot is marked with a plus sign.

2.1.1 The `register` command

The `register` command enables you to handle all tasks regarding registration and subscription management. You can supply the following options:

`--list-extensions`

With this option, the command will list available extensions for your system. You can use the output to find a product identifier for product activation.

`-p, --product`
: Use this option to specify a product for activation. The product identifier has the following format: <*name*>/<*version*>/<*architecture*>, for example `sle-module-live-patching/15.3/x86_64`. The appropriate command will then be the following:

```
# transactional-update register -p sle-module-live-patching/15.3/x86_64
```

`-r, --regcode`
: Register your system with the provided registration code. The command will register the subscription and enable software repositories.

`-d, --de-register`
: The option deregisters the system, or when used along with the `-p` option, deregisters an extension.

`-e, --email`
: Specify an email address that will be used in SUSE Customer Center for registration.

`--url`
: Specify the URL of your registration server. The URL is stored in the configuration and will be used in subsequent command invocations. For example:

```
# transactional-update register --url https://scc.suse.com
```

`-s, --status`
: Displays the current registration status in JSON format.

`--write-config`
: Writes the provided options value to the `/etc/SUSEConnect` configuration file.

`--cleanup`
: Removes old system credentials.

`--version`
: Prints the version.

`--help`
: Displays usage of the command.

2.2 Snapshots cleanup

If you run the command **`transactional-update cleanup`**, all old snapshots without a cleanup algorithm will have one set. All important snapshots are also marked. The command also removes all unreferenced (and thus unused) `/etc` overlay directories in `/var/lib/overlay`.

The snapshots with the set `number` cleanup algorithm will be deleted according to the rules configured in `/etc/snapper/configs/root` by the following parameters:

NUMBER_MIN_AGE
: Defines the minimum age of a snapshot (in seconds) that can be automatically removed.

NUMBER_LIMIT/NUMBER_LIMIT_IMPORTANT
: Defines the maximum count of stored snapshots. The cleaning algorithms delete snapshots above the specified maximum value, without taking the snapshot and file system space into account. The algorithms also delete snapshots above the minimum value until the limits for the snapshot and file system are reached.

The snapshot cleanup is also preformed regularly by `systemd`.

2.3 System rollback

GRUB 2 enables booting from btrfs snapshots and thus allows you to use any older functional snapshot in case that the new snapshot does not work correctly.

When booting a snapshot, the parts of the file system included in the snapshot are mounted read-only; all other file systems and parts that are excluded from snapshots are mounted read-write and can be modified.

Tip: Rolling back to a specific installation state

An initial bootable snapshot is created at the end of the initial system installation. You can go back to that state at any time by booting this snapshot. The snapshot can be identified by the description `after installation`.

There are two methods how you can perform a system rollback.

- From a running system you can set the default snapshot, see more in *Procedure 2, "Rollback from a running system"*.
- Especially in cases where the current snapshot is broken, you can boot to the new snapshot and set it then default, for details refer to *Procedure 3, "Rollback to a working snapshot"*.

In case your current snapshot is functional, you can use the following procedure for system rollback.

PROCEDURE 2: ROLLBACK FROM A RUNNING SYSTEM

1. Choose the snapshot that should be set as default, run:

   ```
   # transactional-update status
   ```

   to get a list of available snapshots. Note the number of the snapshot to be set as default.

2. Set the snapshot as the default by running:

   ```
   # transactional-update rollback snapshot_number
   ```

   If you omit the `snapshot number`, the current snapshot will be set as default.

3. Reboot your system to boot in to the new default snapshot.

The following procedure is used in case the current snapshot is broken and you are not able to boot into it.

PROCEDURE 3: ROLLBACK TO A WORKING SNAPSHOT

1. Reboot your system and select `Start bootloader from a read-only snapshot`
2. Choose a snapshot to boot. The snapshots are sorted according to the date of creation, with the latest one at the top.
3. Log in to your system and check whether everything works as expected. Data written to directories excluded from the snapshots will stay untouched.
4. If the snapshot you booted into is not suitable for rollback, reboot your system and choose another one.
   If the snapshot works as expected, you can perform rollback by running the following command:

   ```
   # transactional-update rollback
   ```

And reboot afterwards.

2.4 Managing automatic transactional updates

Automatic updates are controlled by a **systemd.timer** that runs once per day. This applies all updates, and informs **rebootmgrd** that the machine should be rebooted. You may adjust the time when the update runs, see systemd.timer(5) documentation.

You can disable automatic transactional updates with this command:

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

3 Health checker

Health checker is a program delivered with SLE Micro that checks whether services are running properly during booting of your system.

During the boot process, `systemd` calls Health checker, which in turn calls its plugins. Each plugin checks a particular service or condition. If each check passes, a status file (`/var/lib/misc/health-checker.state`) is created. The status file marks the current root file system as correct.

If any of the health checker plugins reports an error, the action taken depends on a particular condition, as described below:

The snapshot is booted for the first time.

If the current snapshot is different from the last one that worked properly, an automatic rollback to the last working snapshot is performed. This means that the last change performed to the file system broke the snapshot.

The snapshot has already booted correctly in the past.

There could be just a temporary problem, and the system is rebooted automatically.

The reboot of a previously correctly booted snapshot has failed.

If there was already a problem during boot and automatic reboot has been triggered, but the problem still persists, then the system is kept running to enable to the administrator to fix the problem. The services that are tested by the health checker plugins are stopped if possible.

3.1 Adding custom plugins

Health checker supports the addition of your own plugins to check services during the boot process. Each plugin is a bash script that must fulfill the following requirements:

- Plugins are located within a specific directory— `/usr/libexec/health-checker`
- The service that will be checked by the particular plugin must be defined in the `Unit` section of the `/usr/lib/systemd/system/health-checker.service` file. For example, the `etcd` service is defined as follows:

  ```
  [Unit]
  ...
  After=etcd.service
  ...
  ```

- Each plugin must have functions called `run.checks` and `stop_services` defined. The `run.checks` function checks whether a particular service has started properly. Bear in mind that service that has not been enabled by systemd, should be ignored. The function `stop_services` is called to stop the particular service in case the service has not been started properly. You can use the plugin template for your reference.

4 SLE Micro administration using Cockpit

Cockpit is a web-based graphical interface that enables you to manage your SLE Micro deployments from one place. Cockpit is included in the delivered pre-built images, or can be installed if you are installing your own instances manually. For details regarding the manual installation, refer to *Book “Deployment Guide”, Chapter 12 “Installation steps”, Section 12.9.2 “Software”*.

Though Cockpit is present in the pre-built images by default, the plugin for administration of virtual machines needs to be installed manually. You can do so by installing the `microos-cockpit` pattern as described bellow. Use the command below as well in case Cockpit is not installed on your system.

```
# transactional-update pkg install -t pattern microos-cockpit
```

Reboot your machine to switch to the latest snapshot.

Note: Cockpit's plugins installed from the `microos-cockpit` pattern may differ according to technologies installed on your system

The plugin `Podman containers` is installed only if the *Containers Runtime for non-clustered systems* patterns are installed on your system. Similarly, the `Virtual Machines` plugin is installed only if the *KVM Virtualization Host* pattern is installed on your system.

Before running Cockpit on you machine, you need to enable the cockpit socket in systemd by running:

```
# systemctl enable --now cockpit.socket
```

In case you have enabled the firewall, you also must open the firewall for Cockpit as follows:

```
# firewall-cmd --permanent --zone=public --add-service=cockpit
```

And then reload the firewall configuration by running:

```
# firewall-cmd --reload
```

Now you can access the Cockpit web interface by opening the following address in your web browser:

```
https://IP_ADDRESS_OF_MACHINE:9090
```

A login screen opens. To login, use the same credentials as you use to login to your machine via console or SSH.

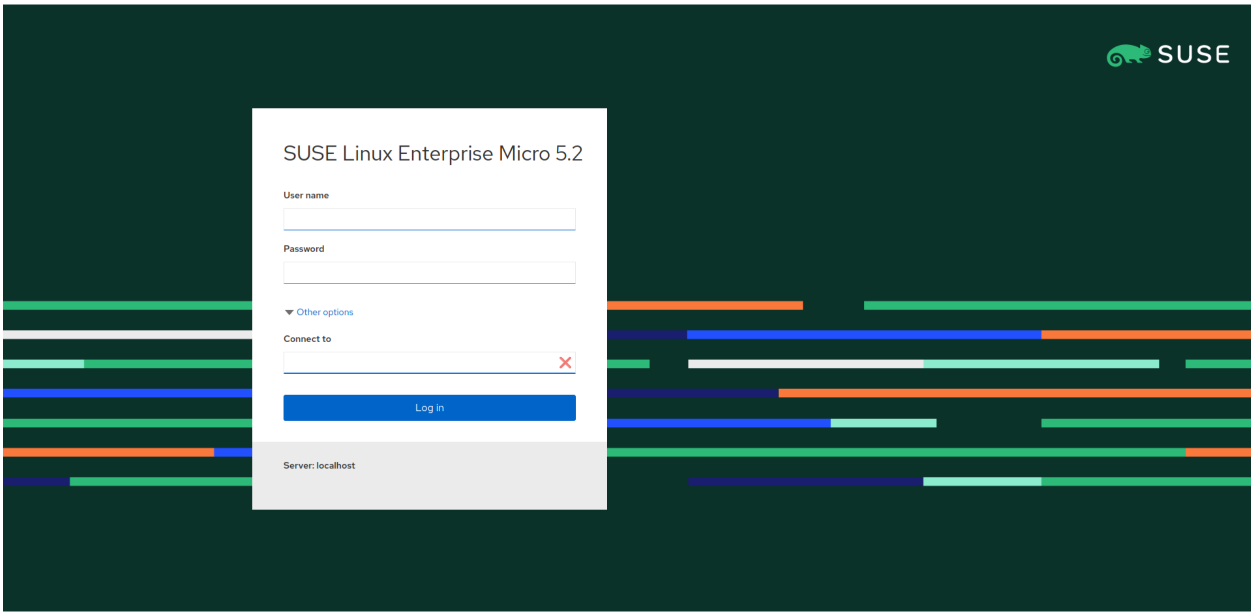

FIGURE 1: COCKPIT LOGIN SCREEN

After successful login, the Cockpit web console opens. Here you can view and administer your system's performance, network interfaces, Podman containers, your virtual machines, services, accounts and logs. You can also access your machine using shell in a terminal emulator.

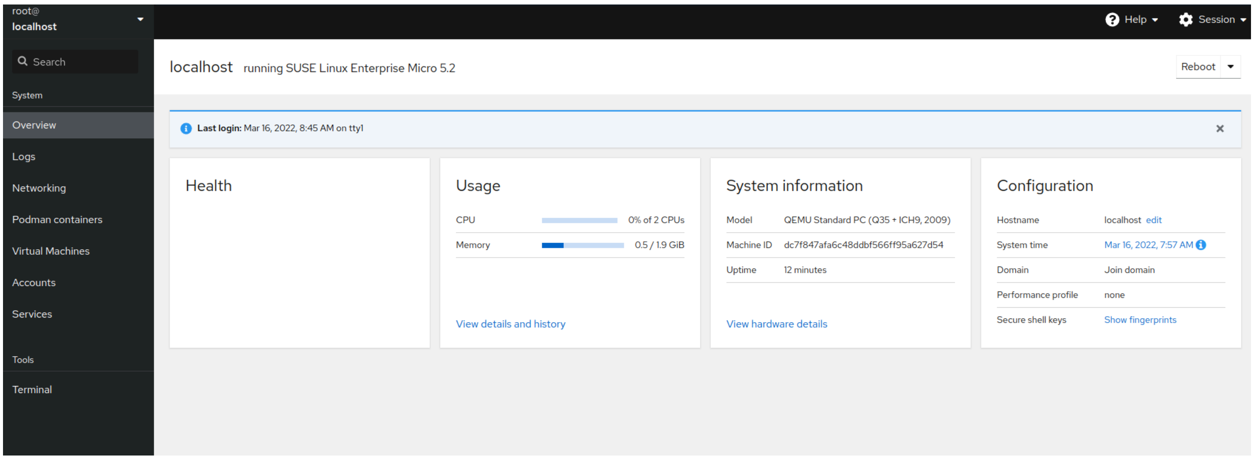

FIGURE 2: COCKPIT DASHBOARD

4.1 Users administration

Note: Users administration only for server administrators

Only users with the *Server administrator* role can edit other users.

Cockpit enables you to manage users of your system. Click *Accounts* to open the user administration page. Here you can create a new account by clicking *Create new account* or manage already existing accounts by clicking on the particular account.

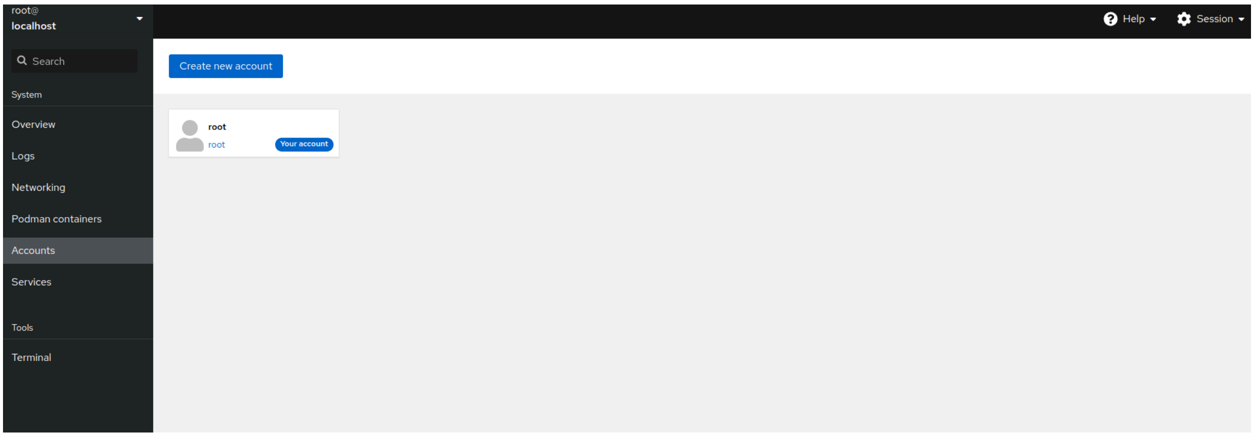

FIGURE 3: THE *ACCOUNTS* SCREEN

4.1.1 Creating new accounts

Click *Create new account* to open the window that enables you to add a new user. Fill in the user's login and/or full name and password, then confirm the form by clicking *Create*.

To add authorized SSH keys for the new user or set the *Server administrator* role, edit the already created account by clicking on it. For details, refer to *Section 4.1.2, "Modifying accounts"*.

4.1.2 Modifying accounts

After clicking the user icon in the *Accounts* page, the user details view opens and you can edit the user.

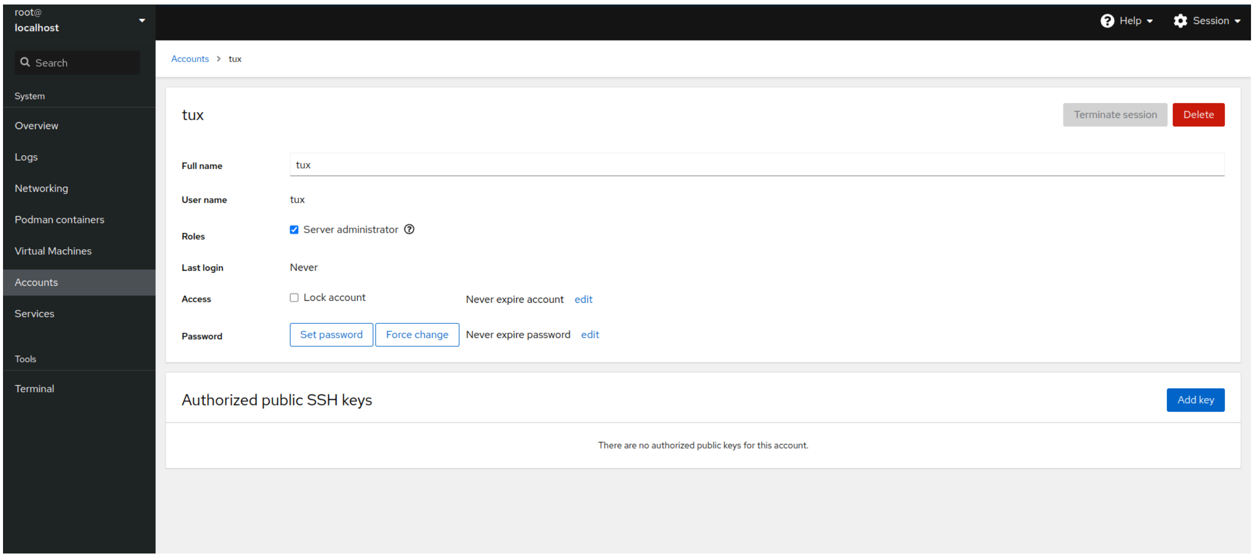

FIGURE 4: USER DETAILS

In the user's details view, you perform the following actions.

Delete the user

Click *Delete* to remove the user from the system.

Terminate user's session

By clicking *Terminate session*, you can log out the particular user from the system.

Change user's role

By checking/unchecking the *Server administrator* check box, you can assign or remove the administrator role from the user.

Manage access of the account

You can lock the account or you can set a date when the account will expire.

Manage the user's password

Click *Set password* to set a new password for the account.

By clicking *Force change*, the user will have to change the password on the next login.

Click *edit* to set whether or when the password expires.

Add SSH key

You can add a SSH key for passwordless authentication via SSH. Click *Add key*, paste the contents of the public SSH key and confirm it by clicking *Add*.

5 toolbox for SLE Micro debugging

SLE Micro uses the **`transactional-update`** command to apply changes to the system, but the changes are applied only after reboot. That solution has several benefits, but it also has some disadvantages. If you need to debug your system and install a new tool, the tool will be available only after reboot. Therefore you are not able to debug the currently running system. For this reason a utility called `toolbox` has been developed.

`toolbox` is a small script that pulls a container image and runs a privileged container based on that image. In the toolbox container you can install any tool you want with **`zypper`** and then use the tool without rebooting your system.

To start the `toolbox` container, run the following:

```
# /usr/bin/toolbox
```

If the script completes successfully, you will see the `toolbox` container prompt.

Note: Obtaining the `toolbox` image

You can also use Podman or Cockpit to pull the `toolbox` image and start a container based on that image.

6 Monitoring performance

For performance monitoring purposes, SLE Micro provides a container image that enables you to run the Performance Co-Pilot (PCP) analysis toolkit in a container. The toolkit comprises tools for gathering and processing performance information collected either in real time or from PCP archive logs.

The performance data are collected by *performance metrics domain agents* and passed to the *pmcd* daemon. The daemon coordinates the gathering and exporting of performance statistics in response to requests from the PCP monitoring tools. **`pmlogger`** is then used to log the metrics. For details, refer to the PCP documentation (https://pcp.readthedocs.io/en/latest/UAG/IntroductionToPcp.html#).

6.1 Getting the PCP container image

The PCP container image is based on the *BCI-Init* container that utilizes `systemd` used to manage the PCP services.

You can pull the container image using podman or from the Cockpit web management console. To pull the image by using podman, run the following command:

```
# podman pull registry.suse.com/suse/pcp:latest
```

To get the container image using Cockpit, go to *Podman containers*, click *Get new image*, and search for *pcp*. Then select the image from the `registry.suse.com` for SLE 15 SP3 and download it.

6.2 Running the PCP container

The following command shows minimal options that you need to use to run a PCP container:

```
# podman run -d  \
  --systemd always \
  -p HOST_IP:HOST_PORT:CONTAINER_PORT \
  -v HOST_DIR:/var/log/pcp/pmlogger \
  PCP_CONTAINER_IMAGE
```

where the options have the following meaning:

`-d`

Runs the container in the `systemd` mode. All services needed to run in the PCP container will be started automatically by `systemd` in the container.

`--systemd always`

Runs the container in the `systemd` mode. All services needed to run in the PCP container will be started automatically by `systemd` in the container.

`--privileged`

The container runs with extended privileges. Use this option if your system has SELinux enabled, otherwise the collected metrics will be incomplete.

`-v HOST_DIR:/var/log/pcp/pmlogger`

Creates a bind mount so that **`pmlogger`** archives are written to the *`HOST_DIR`* on the host. By default, **`pmlogger`** stores the collected metrics in `/var/log/pcp/pmlogger`.

`PCP_CONTAINER_IMAGE`

Is the downloaded PCP container image.

Other useful options of the **`podman run`** command follow:

OTHER OPTIONS

`-p` *`HOST_IP:HOST_PORT:CONTAINER_PORT`*
: Publishes ports of the container by mapping a container port onto a host port. If you do not specify *`HOST_IP`*, the ports will be mapped on the local host. If you omit the *`HOST_PORT`* value, a random port number will be used. By default, the **`pmcd`** daemon listens and exposes the PMAPI to receive metrics on the port *44321*, so it is recommended to map this port on the same port number on the host. The **`pmproxy`** daemon listens on and exposes the REST PMWEBAPI to access metrics on the *44322* port by default, so it is recommended to map this port on the same host port number.

`--net host`
: The container uses the host's network. Use this option if you want to collect metrics from the host's network interfaces.

`-e`
: The option enables you to set the following environment variables:

PCP_SERVICES
: Is a comma-separated list of services to start by `systemd` in the container.
Default services are: `pmcd`, `pmie`, `pmlogger`, `pmproxy`.
You can use this variable, if you want to run a container with a list of services that is different from the default one, for example, only with `pmlogger`:

```
# podman run -d \
  --name pmlogger \
  --systemd always \
  -e PCP_SERVICES=pmlogger  \
  -v pcp-archives:/var/log/pcp/pmlogger  \
  registry.suse.com/suse/pcp:latest
```

HOST_MOUNT
: Is a path inside the container to the bind mount of the host's root file system. The default value is not set.

REDIS_SERVERS
: Specifies a connection to a Redis server. In a non-clustered setup, provide a comma-separated list of hosts specs. In a clustered setup, provide any individual cluster host, other hosts in the cluster are discovered automatically. The default value is: `localhost:6379`.

If you need to use different configuration then provided by the environment variables, proceed as described in *Section 6.3, "Configuring PCP services"*.

6.3 Configuring PCP services

All services that run inside the PCP container have a default configuration that might not suit your needs. If you need a custom configuration that cannot be covered by the environment variables described above, create configuration files for the PCP services and pass them to the PCP using a bind mount as follows:

```
# podman run -d \
  --name CONTAINER_NAME \
  --systemd always \
  -v $HOST_CONFIG:CONTAINER_CONFIG_PATH:z \
  -v HOST_LOGS_PATH:/var/log/pcp/pmlogger  \
  registry.suse.com/suse/pcp:latest
```

Where:

CONTAINER_NAME
: Is an optional container name.

HOST_CONFIG
: Is an absolute path to the config you created on the host machine. You can choose any file name you want.

CONTAINER_CONFIG_PATH
: Is an absolute path to a particular configuration file inside the container. Each available configuration file is described in the corresponding sections further.

HOST_LOGS_PATH
: Is a directory that should be bind mount to the container logs.

For example, a container called `pcp`, with the configuration file `pmcd` on the host machine and the `pcp-archives` directory for logs on the host machine, is run by the following command:

```
# podman run -d \
  --name pcp  \
  --systemd always \
  -v $(pwd)/pcp-archives:/var/log/pcp/pmlogger \
  -v $(pwd)/pmcd:/etc/sysconfig/pmcd \
```

```
registry.suse.com/suse/pcp:latest
```

6.3.1 Custom **pmcd** daemon configuration

The **`pmcd`** daemon configuration is stored in the `/etc/sysconfig/pmcd` file. The file stores environment variables that modify the behavior of the **`pmcd`** daemon.

6.3.1.1 The `/etc/sysconfig/pmcd` file

You can add the following variables to the file to configure the **`pmcd`** daemon:

PMCD_LOCAL

Defines whether the remote host can connect to the **`pmcd`** daemon. If set to *0*, remote connections to the daemon are allowed. If set to *1*, the daemon listens only on the local host. The default value is *0*.

PMCD_MAXPENDING

Defines the maximum count of pending connections to the agent. The default value is 5.

PMCD_ROOT_AGENT

If the `pmdaroot` is enabled (the value is set to *1*), adding a new PDMA does not trigger restarting of other PMDAs. If `pmdaroot` is not enabled, **`pmcd`** will require to restart all PMDAs when a new PMDA is added. The default value is *1*.

PMCD_RESTART_AGENTS

If set to *1*, the **`pmcd`** daemon tries to restart any exited PMDA. Enable this option only if you have enabled `pmdaroot`, as **`pmcd`** itself does not have privileges to restart PMDA.

PMCD_WAIT_TIMEOUT

Defines the maximum time in seconds, **`pmcd`** can wait to accept a connection. After this time, the connection is reported as failed. The default value is *60*.

PCP_NSS_INIT_MODE

Defines the mode in which **`pmcd`** initializes the NSS certificate database when secured connections are used. The default value is `readonly`. You can set the mode to `readwrite`, but if the initialization fails, the default value is used as a fallback.

An example follows:

```
PMCD_LOCAL=0
```

```
PMCD_MAXPENDING=5
PMCD_ROOT_AGENT=1
PMCD_RESTART_AGENTS=1
PMCD_WAIT_TIMEOUT=70
PCP_NSS_INIT_MODE=readwrite
```

6.3.2 Custom `pmlogger` configuration

The custom configuration for the **`pmlogger`** is stored in the following configuration files:

- `/etc/sysconfig/pmlogger`
- `/etc/pcp/pmlogger/control.d/local`

6.3.2.1 The `/etc/sysconfig/pmlogger` file

You can use the following attributes to configure the **`pmlogger`**:

PMLOGGER_LOCAL
: Defines whether **`pmlogger`** allows connections from remote hosts. If set to *1*, **`pmlogger`** allows connections from local host only.

PMLOGGER_MAXPENDING
: Defines the maximum count of pending connections. The default value is 5.

PMLOGGER_INTERVAL
: Defines the default sampling interval **`pmlogger`** uses. The default value is *60 s*. Keep in mind that this value can be overridden by the **`pmlogger`** command line.

PMLOGGER_CHECK_SKIP_LOGCONF
: Setting this option to *yes* disables the regeneration and checking of the **`pmlogger`** configuration if the configuration **`pmlogger`** comes from **`pmlogconf`**. The default behavior is to regenerate configuration files and check for changes every time **`pmlogger`** is started.

An example follows:

```
PMLOGGER_LOCAL=1
PMLOGGER_MAXPENDING=5
PMLOGGER_INTERVAL=10
PMLOGGER_CHECK_SKIP_LOGCONF=yes
```

6.3.2.2 The /etc/pcp/pmlogger/control.d/local file

The file `/etc/pcp/pmlogger/control.d/local` stores specifications of the host, which metrics should be logged, the logging frequency (default is 24 hours), and **`pmlogger`** options. For example:

```
# === VARIABLE ASSIGNMENTS ===
#
# DO NOT REMOVE OR EDIT THE FOLLOWING LINE
$version=1.1

# Uncomment one of the lines below to enable/disable compression behaviour
# that is different to the pmlogger_daily default.
# Value is days before compressing archives, 0 is immediate compression,
# "never" or "forever" suppresses compression.
#
#$PCP_COMPRESSAFTER=0
#$PCP_COMPRESSAFTER=3
#$PCP_COMPRESSAFTER=never

# === LOGGER CONTROL SPECIFICATIONS ===
#
#Host           P?  S?  directory                      args

# local primary logger
LOCALHOSTNAME    y   n   PCP_ARCHIVE_DIR/LOCALHOSTNAME   -r -T24h10m -c config.default -v
 100Mb
```

Note: Defaults point to local host

If you run the **`pmlogger`** in a container on a different machine than the one that runs the **`pmcd`** (a client), change the following line to point to the client:

```
# local primary logger
CLIENT_HOSTNAME    y   n   PCP_ARCHIVE_DIR/CLIENT_HOSTNAME   -r -T24h10m -c
 config.default -v 100Mb
```

For example, for the `slemicro_1` host name, the line should look as follows:

```
# local primary logger
slemicro_1    y   n   PCP_ARCHIVE_DIR/slemicro_1   -r -T24h10m -c config.default -v
 100Mb
```

6.4 Starting the PCP container automatically on boot

After you run the PCP container, you can configure `systemd` to start the container on boot. To do so, follow the procedure below:

1. Create a unit file for the container by using the **`podman generate systemd`** command:

   ```
   # podman generate systemd --name CONTAINER_NAME > /etc/systemd/system/
   container-CONTAINER_NAME.service
   ```

   where *`CONTAINER_NAME`* is the name of the PCP container you used when running the container from the container image.

2. Enable the service in `systemd`:

   ```
   # systemctl enable container-CONTAINER_NAME
   ```

6.5 Metrics management

6.5.1 Listing available performance metrics

From within the container, you can use the command **`pminfo`** to list metrics. For example, to list all available performance metrics, run:

```
# pminfo
```

You can list a group of related metrics by specifying the metrics prefix:

```
# pminfo METRIC_PREFIX
```

For example, to list all metrics related to kernel, use:

```
# pminfo disk

disk.dev.r_await
disk.dm.await
disk.dm.r_await
disk.md.await
disk.md.r_await
...
```

You can also specify additional strings to narrow down the list of metrics, for example:

```
# piminfo disk.dev

disk.dev.read
disk.dev.write
disk.dev.total
disk.dev.blkread
disk.dev.blkwrite
disk.dev.blktotal
...
```

To get online help text of a particular metric, use the `-t` option followed by the metric, for example:

```
# pminfo -t kernel.cpu.util.user

kernel.cpu.util.user [percentage of user time across all CPUs, including guest CPU time]
```

To display a description text of a particular metric, use the `-T` option followed by the metric, for example:

```
# pminfo -T kernel.cpu.util.user

Help:
percentage of user time across all CPUs, including guest CPU time
```

6.5.2 Checking local metrics

After you start the PCP container, you can verify that metrics are being recorded properly by running the following command inside the container:

```
# pcp

Performance Co-Pilot configuration on localhost:

 platform: Linux localhost 5.3.18-150300.59.68-default #1 SMP Wed May 4 11:29:09 UTC 2022
 (ea30951) x86_64
 hardware: 1 cpu, 1 disk, 1 node, 1726MB RAM
 timezone: UTC
 services: pmcd pmproxy
     pmcd: Version 5.2.2-1, 9 agents, 4 clients
     pmda: root pmcd proc pmproxy xfs linux mmv kvm jbd2
 pmlogger: primary logger: /var/log/pcp/pmlogger/localhost/20220607.09.24
```

```
pmie: primary engine: /var/log/pcp/pmie/localhost/pmie.log
```

Now check if the logs are written to a proper destination:

```
# ls PATH_TO_PMLOGGER_LOGS
```

where `PATH_TO_PMLOGGER_LOGS` should be `/var/log/pcp/pmlogger/localhost/` in this case.

6.5.3 Recording metrics from remote systems

You can deploy collector containers that collect metrics from different remote systems than the ones where the **`pmlogger`** containers are running. Each remote collector system needs the **`pmcd`** daemon and a set of *pmda*. To deploy several collectors with a centralized monitoring system, proceed as follows.

1. On each system you want to collect metrics from (clients), run a container with the **`pmcd`** daemon:

   ```
   # podman run -d \
       --name pcp-pmcd \
       --privileged \
       --net host \
       --systemd always \
       -e PCP_SERVICES=pmcd \
       -e HOST_MOUNT=/host \
       -v /:/host:ro,rslave \
       registry.suse.com/suse/pcp:latest
   ```

2. On the monitoring system, create a **`pmlogger`** configuration file for each client `control.CLIENT` with the following content:

   ```
   $version=1.1

   CLIENT_HOSTNAME n n PCP_ARCHIVE_DIR/CLIENT -N -r -T24h10m -c config.default -v 100Mb
   ```

   Keep in mind that the `CLIENT_HOSTNAME` must be resolvable in DNS. You can use IP addresses or fully qualified domain names (FQDN) instead.

3. On the monitoring system, create a directory for each client to store the recorded logs:

   ```
   # mkdir /root/pcp-archives/CLIENT
   ```

For example, for `slemicro_1`:

```
# mkdir /root/pcp-archives/slemicro_1
```

4. On the monitoring system, run a container with **`pmlogger`** for each client:

```
# podman run -d \
    --name pcp-pmlogger-CLIENT \
    --systemd always \
    -e PCP_SERVICES=pmlogger \
    -v /root/pcp-archives/CLIENT:/var/log/pcp/pmlogger:z \
    -v $(pwd)/control.CLIENT:/etc/pcp/pmlogger/control.d/local:z \
    registry.suse.com/suse/pcp:latest
```

For example, for a client called `slemicro_1`:

```
# podman run -d \
    --name pcp-pmlogger-slemicro_1 \
    --systemd always \
    -e PCP_SERVICES=pmlogger \
    -v /root/pcp-archives:/var/log/pcp/pmlogger:z \
    -v $(pwd)/control.slemicro_1:/etc/pcp/pmlogger/control.d/local:z \
    registry.suse.com/suse/pcp:latest
```

Note

The second bind mount points to the configuration file created in *Step 2* and replaces the default **`pmlogger`** configuration. If you do not create this bind mount, **`pmlogger`** uses the default `/etc/pcp/pmlogger/control.d/local` file and logging from clients fails as the default configuration points to a local host. For details about the configuration file, refer to *Section 6.3.2.2, "The `/etc/pcp/pmlogger/control.d/local` file"*.

5. To check if the logs collection is working properly, run:

```
# ls -l pcp-archives/CLIENT/CLIENT
```

For example:

```
# ls -l pcp-archives/slemicro_1/slemicro_1

total 1076
-rw-r--r--. 1 systemd-network systemd-network 876372 Jun  8 11:24 20220608.10.58.0
```

```
-rw-r--r--. 1 systemd-network systemd-network    312 Jun  8 11:22
 20220608.10.58.index
-rw-r--r--. 1 systemd-network systemd-network 184486 Jun  8 10:58
 20220608.10.58.meta
-rw-r--r--. 1 systemd-network systemd-network    246 Jun  8 10:58 Latest
-rw-r--r--. 1 systemd-network systemd-network  24595 Jun  8 10:58 pmlogger.log
```

7 User management

You can define users during the deployment process of SLE Micro. Although you can define users as you want when deploying pre-built images, during the manual installation, you define only the `root` user in the installation flow. Therefore, you might want to use other users than those provided during the installation process. There are two possibilities for adding users to an already installed system:

- using CLI - the command **`useradd`**. Run the following command for usage:

  ```
  # useradd --help
  ```

  Bear in mind that a user that should have the server administrator role, must be included in the group `wheel`.

- using Cockpit; for details refer to *Section 4.1, “Users administration”*.

A GNU licenses

This appendix contains the GNU Free Documentation License version 1.2.

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

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

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

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

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

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

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

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

2. VERBATIM COPYING

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

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

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

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

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

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

D. Preserve all the copyright notices of the Document.

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

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

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

H. Include an unaltered copy of this License.

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

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

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

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

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

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

O. Preserve any Warranty Disclaimers.

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

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

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

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

5. COMBINING DOCUMENTS

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

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

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

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

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

ADDENDUM: How to use this License for your documents

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

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

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

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

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