SUSE

使用 Performance Co-Pilot 分析工具包分析性能指标

解释

出于性能监控目的，SUSE Linux Enterprise Micro 提供了一个容器映像，用于在容器中运行 Performance Co-Pilot (PCP) 分析工具包。

原因

您想要监控系统性能。本文提供了有关如何配置和使用该工具包的信息。

工作量

读完本文大约需要 40 分钟。

目标

您可以使用符合需求的配置启动 PCP 容器。

要求

- 一个正在运行的 SLE Micro 实例。

出版日期：2025 年 12 月 11 日

目录

1 Performance Co-Pilot 分析工具包

该工具包中的工具可以实时收集或从 PCP 存档日志收集性能信息并对其进行处理。

性能数据由**性能指标域代理**收集，并传递给 `pmcd` 守护程序。该守护程序协调性能统计信息的收集和导出，以响应 PCP 监控工具的请求。然后使用 **`pmlogger`** 来记录指标。有关细节，请参见 PCP 文档 (https://pcp.readthedocs.io/en/latest/UAG/IntroductionToPcp.html#)。

1.1 获取 PCP 容器映像

PCP 容器映像基于 **BCI-Init** 容器，该容器利用 `systemd` 来管理 PCP 服务。

可以使用 Podman 或者从 Cockpit Web 管理控制台提取容器映像。要使用 Podman 提取映像，请运行以下命令：

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

要使用 Cockpit 获取容器映像，请转到 Podman 容器，单击获取新映像，并搜索 **pcp**。然后在 `registry.suse.com` 中选择并下载适用于 SLE 15 SP4 的映像。

2 运行 PCP 容器

以下命令显示了为运行 PCP 容器而至少需要使用的选项：

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

其中各选项的含义如下：

`-d`

容器在没有 tty 的情况下以分离模式运行。

`--systemd always`

在 `systemd` 模式下运行容器。需要在 PCP 容器中运行的所有服务由容器中的 `systemd` 自动启动。

`--privileged`

容器以扩展的特权运行。如果您的系统启用了 SELinux，请使用此选项，否则收集的指标是不完整的。

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

创建绑定挂载，以便将 **pmlogger** 存档写入主机上的 `HOST_DIR`。默认情况下，**pmlogger** 将收集的指标存储在 `/var/log/pcp/pmlogger` 中。

`PCP_CONTAINER_IMAGE`

下载的 PCP 容器映像。

podman run 命令的其他有用选项如下：

其他选项

`-p HOST_IP:HOST_PORT:CONTAINER_PORT`

通过将容器端口映射到主机端口来发布容器端口。如果未指定 `HOST_IP`，则端口将映射到本地主机。如果省略 `HOST_PORT` 值，则使用随机端口号。默认情况下，**pmcd** 守护程序会侦听并公开 PMAPI 以接收端口 **44321** 上的指标，因此我们建议将此端口映射到主机上的相同端口号。默认情况下，**pmproxy** 守护程序侦听并公开 REST PMWEBAPI 以访问端口 **44322** 端口上的指标，因此建议将此端口映射到相同的主机端口号。

`--net host`

容器使用主机的网络。使用此选项可从主机的网络接口收集指标。

`-e`

使用该选项可设置以下环境变量：

PCP_SERVICES

容器中 `systemd` 启动的服务的逗号分隔列表。

默认服务为：`pmcd`、`pmie`、`pmlogger`、`pmproxy`。

可以通过此变量使用非默认的服务列表（例如，仅使用 `pmlogger`）来运行容器：

```
# 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

容器内部指向主机根文件系统的绑定挂载点的路径。未设置默认值。

REDIS_SERVERS

指定与 Redis 服务器的连接。在非群集设置中，请提供逗号分隔的主机规格列表。在群集设置中，请提供任一群集主机，系统会自动发现群集中的其他主机。默认值为：`localhost:6379`。

如果需要使用其他配置，而不是环境变量提供的配置，请按照第 3 节 “配置 PCP 服务”中所述继续操作。

2.1 引导时自动启动 PCP 容器

运行 PCP 容器后，可将 `systemd` 配置为在引导时启动容器。为此，请按以下过程操作:

1. 使用 **`podman generate systemd`** 命令为容器创建单元文件：

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

   其中，`CONTAINER_NAME` 是从容器映像运行容器时使用的 PCP 容器名称。

2. 在 `systemd` 中启用服务：

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

3 配置 PCP 服务

在 PCP 容器内部运行的所有服务的默认配置可能不符合您的需要。如果需要创建无法被环境变量覆盖的自定义配置，请为 PCP 服务创建配置文件，并使用绑定挂载点将其传递给 PCP，如下所示：

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

其中：

`CONTAINER_NAME`

可选的容器名称。

`HOST_CONFIG`

在主机上创建的配置的绝对路径。您可以选择所需的任何文件名。

`CONTAINER_CONFIG_PATH`

容器内部特定配置文件的绝对路径。后续相应章节将介绍每个可用的配置文件。

`HOST_LOGS_PATH`

一个目录，应是容器日志的绑定挂载点。

例如，以下命令运行名为 `pcp` 的容器，其中包含主机上的 `pmcd` 配置文件以及主机上的 `pcp-archives` 日志目录：

```
# 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
```

3.1 自定义 pmcd 守护程序配置

pmcd 守护程序配置存储在 /etc/sysconfig/pmcd 文件中。该文件存储了用于修改 **pmcd** 守护程序行为的环境变量。

可将以下变量添加到 /etc/sysconfig/pmcd 文件以配置 **pmcd** 守护程序：

PMCD_LOCAL

定义远程主机是否可以连接到 **pmcd** 守护程序。如果设置为 **0**，则允许与守护程序建立远程连接。如果设置为 **1**，则守护程序仅侦听本地主机。默认值为 **0**。

PMCD_MAXPENDING

定义与代理建立的最大等待中连接计数。默认值为 **5**。

PMCD_ROOT_AGENT

如果启用了 pmdaroot（值设置为 **1**），则添加新的 PDMA 不会触发其他 PMDA 的重启动。如果未启用 pmdaroot，则 **pmcd** 要求在添加新的 PMDA 后重启动所有 PMDA。默认值为 **1**。

PMCD_RESTART_AGENTS

如果设置为 **1**，**pmcd** 守护程序将尝试重启动任何已退出的 PMDA。请仅在启用了 pmdaroot 时才启用此选项，因为 **pmcd** 本身无权重启动 PMDA。

PMCD_WAIT_TIMEOUT

定义 **pmcd** 可以等待接受连接的最长时间（以秒为单位）。在此时间过后，会将连接报告为失败。默认值为 **60**。

PCP_NSS_INIT_MODE

定义在使用安全连接时，**pmcd** 以哪种模式初始化 NSS 证书数据库。默认值为 readonly. 可将模式设置为 readwrite，但如果初始化失败，将使用默认值作为回退模式。

示例如下：

```
PMCD_LOCAL=0
      PMCD_MAXPENDING=5
      PMCD_ROOT_AGENT=1
      PMCD_RESTART_AGENTS=1
      PMCD_WAIT_TIMEOUT=70
```

```
    PCP_NSS_INIT_MODE=readwrite
```

3.2 自定义 pmlogger 配置

pmlogger 的自定义配置存储在以下配置文件中：

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

3.2.1 /etc/sysconfig/pmlogger 文件

可使用以下属性来配置 **pmlogger**：

PMLOGGER_LOCAL

定义 **pmlogger** 是否允许来自远程主机的连接。如果设置为 **1**，则 **pmlogger** 只允许来自本地主机的连接。

PMLOGGER_MAXPENDING

定义最大等待中连接计数。默认值为 5。

PMLOGGER_INTERVAL

定义 **pmlogger** 使用的默认采样间隔。默认值为 **60 s**。请记住，可以通过 **pmlogger** 命令行覆盖此值。

PMLOGGER_CHECK_SKIP_LOGCONF

如果将此选项设置为 **yes**，在 **pmlogger** 配置来自 **pmlogconf** 的情况下，会禁用 **pmlogger** 配置的重新生成和检查。默认行为是重新生成配置文件，并在每次启动 **pmlogger** 时检查更改。

示例如下：

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

```
PMLOGGER_CHECK_SKIP_LOGCONF=yes
```

3.2.2 /etc/pcp/pmlogger/control.d/local 文件

文件 `/etc/pcp/pmlogger/control.d/local` 存储主机的规格，即应记录的指标、日志记录频率（默认值为 24 小时）和 **pmlogger** 选项。例如：

```
# === 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
```

注意：默认值指向本地主机

如果在其他计算机而不是运行 **pmcd** 的计算机（客户端）上的容器中运行 **pmlogger**，请更改以下行以指向客户端：

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

例如，对于 `slemicro_1` 主机名，该行应如下所示：

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

4 管理 PCP 指标

4.1 列出 PCP 指标

在容器中，可以使用 **`pminfo`** 命令列出指标。例如，要列出所有可用的性能指标，请运行：

```
# pminfo
```

可以通过指定指标前缀来列出一组相关指标：

```
# pminfoMETRIC_PREFIX
```

例如，要列出与内核相关的所有指标，请使用：

```
# pminfo disk

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

还可以指定其他字符串来缩小指标列表的范围，例如：

```
# piminfo disk.dev

disk.dev.read
disk.dev.write
```

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

要获取特定指标的联机帮助文本，请使用 `-t` 选项加上指标，例如：

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

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

要显示特定指标的说明文本，请使用 `-T` 选项加上指标，例如：

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

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

4.2 检查本地指标

启动 PCP 容器后，可以通过在容器内部运行以下命令来校验是否正确记录了指标：

```
# 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
```

现在检查日志是否已写入正确的目标：

```
# lsPATH_TO_PMLOGGER_LOGS
```

在本例中，PATH_TO_PMLOGGER_LOGS 应是 /var/log/pcp/pmlogger/localhost/。

4.3 从远程系统记录指标

您可以部署收集器容器，用于从其他远程系统（而不是运行 **pmlogger** 容器的系统）收集指标。每个远程收集器系统都需要 **pmcd** 守护程序和一组 **pmda**。要使用集中式监控系统部署多个收集器，请执行以下步骤。

1. 在要从中收集指标的每个系统（客户端）上，使用 **pmcd** 守护程序运行容器：

```
# 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. 在监控系统上，为每个客户端 control.CLIENT 创建包含以下内容的 **pmlogger** 配置文件：

```
$version=1.1

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

请记住，CLIENT_HOSTNAME 必须在 DNS 中可解析。可以改用 IP 地址或完全限定的域名 (FQDN)。

3. 在监控系统上，为每个客户端创建一个目录用于存储记录的日志：

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

例如，对 `slemicro_1` 运行以下命令：

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

4. 在监控系统上，使用每个客户端的 **`pmlogger`** 运行一个容器：

```
# 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
```

例如，对名为 `slemicro_1` 的客户端运行以下命令：

```
# 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
```

注意

第二个绑定挂载点指向 步骤 2 中创建的配置文件，并替换默认的 **`pmlogger`** 配置。如果您未创建此绑定挂载点，**`pmlogger`** 将使用默认的 `/etc/pcp/pmlogger/control.d/local` 文件，并且从客户端记录日志将会失败，因为默认配置指向本地主机。有关配置文件的细节，请参见 第 3.2.2 节 “/etc/pcp/pmlogger/control.d/local 文件”。

5. 要检查日志收集是否正常进行，请运行：

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

例如：

```
# 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
```

5 法律声明

A GNU 自由文档许可证

0. 导言

此许可证的目的是赋予手册、教科书或其他功能性的和有用的文档以“自由”：即保证每个人都有复制和再分发此类文档的有效自由，无论是否进行修改，也无论将其用于商业或非商业用途。其次，此许可证为作者和出版者保留了因工作获得声誉但不视为对他人所做修改负责的方式。

本许可证是一种“非盈利版权”，这意味着从该文档衍生的作品也必须是以同一方式自由的。它补充了 GNU 通用公共许可证（为自由软件设计的非盈利版权许可证）。

我们设计此许可证旨在将其用于免费软件的手册，因为免费软件需要自由文档：免费程序所附手册应具有与软件本身同样的自由。但是此许可证不限于软件手册；它可以用于任何文本作品，无论主题如何或它是否作为印刷书籍出版。建议本许可证主要用于目的是指导或参考的作品。

1. 适用性和定义

此许可证适用的对象：由版权所有者在其中明确声明可按照此许可证条款以任何媒体分发的任何手册和其他作品。此类声明授予在此处所述的条款和条件下使用该作品的全球无限期无版权许可证。下述“文档”指任何此类手册或作品。任何公众成员都是一个被许可人，以下称为“您”。如果您以需要版权法许可的任何方式复制、修改或分发该作品，则表示您接受该许可证。

该文档的“修改版本”表示包含该文档或其一部分（或者逐字复制或者有修改和/或翻译为另一语言）的任何作品。

“次要章节”是该文档的命名附录或扉页章节，专门讲述该文档的出版者或作者与该文档整个主题（或相关问题）的关系，不包含与整个主题相关的内容。（因此，如果该文档是数学课本的一部分，则辅助部分可能不说明任何数学问题。）这种关系可以是与主题或相关问题的历史联系，或与它们相关的法律、商业、哲学、伦理或政治地位。

在该文档基于此许可证项发布的声明中，“固定章节”是将其标题指定为固定章节标题的一些辅助章节。如果一个章节不适用上述辅助章节的定义，则不允许将其指定为固定章节。该文档可能不包含固定章节。如果该文档不标识任何固定章节，则表示没有固定章节。

在该文档基于此许可证项发布的声明中，“封页文本”是作为封面文本或封底文本列出的简短文本段落。封面文本最多 5 个单词，封底文本最多 25 个单词。

文档的“透明”副本是一个机器可读的副本，使用公众可以得到其规范的格式表达，这样的副本适合于使用通用文本编辑器、（对于像素构成的图像）通用绘图程序、（对于绘制的图形）广泛使用的绘画编辑器直接修改文档，也适用于输入到文本格式处理程序或自动翻译成各种适于适用于输入到文本格式处理程序的格式。一个用其他透明文件格式表示的副本，如果该格式的标记（或缺少标记）已经构成了对读者的后续修改的障碍，那么就是不透明的。表示实质性数量的文本的图像格式都是不透明的。不“透明”的副本称为“不透明”。

适于作为透明副本的格式的示例有：没有标记的纯 ASCII 文本、Texinfo 输入格式、LaTeX 输入格式、使用公共可用 DTD 的 SGML 或 XML，符合标准的简单 HTML、可以人为修改的 PostScript 或 PDF。透明图像格式的示例有 PNG、XCF 和 JPG。不透明的格式包括：仅可以被私有版权的字处理软件使用的私有版权格式、所用的 DTD 和/或处理工具不是广泛可用的的 SGML 或 XML、机器生成的 HTML、一些字处理器生成的只用于输出目的的 PostScript 或 PDF。

对于印刷书籍，“扉页”就是扉页本身以及随后的一些用于补充的页，显然本许可资料需要出现在扉页上。对于那些没有扉页的作品形式，“扉页”代表接近作品最突出标题的、在文本正文之前的文本。

“命名为 XYZ”的章节表示文档的一个特定的子单元，其标题就是 XYZ 或在括号中包含 XYZ 且后跟 XYZ 的其他语言翻译文本。（这里 XYZ 代表下面提及的特定章节名称，比如“致谢”、“题献”、“签名”或“历史”。）要在修改文档时对这类章节“保留标题”就是依据此定义保持这样一个“命名为 XYZ”的章节。

文档可能在文档遵照此许可证的声明后面包含免责声明。这些免责声明应作为参考信息包含在此许可证中，但是只能将其视作免责声明：这些免责声明暗指的任何其他含义均无效，且对此许可证的含义不产生任何影响。

2. 逐字复制

您可以用任何媒体复制并分发文档，无论是出于商业还是非商业目的，只要保证此许可证、版权声明和声称此许可证应用于文档的声明都完整地、无任何附加条件地存在于所有副本中。不能使用任何技术手段阻碍或控制您制作或发布的副本的阅读或再次复制。不过您可以在副本交易中得到报酬。如果发布足够多的副本，则您必须遵循下面第三节中的条件。

您也可以在如上的条件下出租副本和向公众放映副本。

3. 大量复制

如果您出版的文档印刷版副本（或是有印制封页的其他媒体副本）多于 100 份，而文档的许可证声明中要求有封页文本，则您必须将它清晰地置于封页之上，封面文本在封面上，封底文本在封底上。封面和封底上还必须标明您是这些副本的出版者。封面必须同等显著地完整展现标题的所有文字。您可以在封页上加入其他资料。改动仅限于封页的复制，只要保持文档的标题不变并满足这些条件，可以在其他方面被视为逐字复制。

如果需要加上的文本对于封面或封底过多，无法明显地表示，您应该在封页上列出前面的（在合理的前提下尽量多），把其他的放在邻近的页面上。

如果您出版或分发了超过 100 份文档的不透明副本，则必须在每个不透明副本中包含一份计算机可读的透明副本，或是在每个不透明副本中给出一个计算机网络地址，通过这个地址，网络公共用户可以使用标准网络协议下载文档的无任何附加资料的完整透明副本。如果您选择后者，则必须在开始大量分发非透明副本的时候采用相当谨慎的步骤，保证透明副本在其所给出的位置在（直接或通过代理和零售商）分发最后一次该版本的非透明副本的时间之后一年之内始终是有效的。

在重新大量发布副本之前，请您（但不是必须）与文档的作者联系，以便他们可以有机会向您提供文档的更新版本。

4. 修改

在上述第 2、3 节的条件下，您可以复制和分发文档的修改版本，前提是严格按照此许可证发布修改后的文档，将修改版本用作文档，从而允许任何拥有此修改版副本的人执行分发或修改。另外，在修改版中，您需要做到如下几点：

A. 用于与文档以及以前各个版本（如果有，应该列在文档的“历史”章节中）显著不同的扉页（和封页，如果有）。如果那个版本的原始发行者允许的话，您可以使用和以前版本相同的标题。

B. 与作者一样，在扉页上列出承担修改版本中的修改的作者责任的一个或多个人或实体和至少五个文档的原作者（如果原作者不足五个就全部列出），除非他们免除了您的这个责任。

C. 与原来的发行者一样，在扉页上列出修改版的发行者的姓名。

D. 保持该文档的全部版权声明不变。

E. 在与其他版权声明邻近的位置加入恰当的针对您的修改的版权声明。

F. 在紧接着版权声明的位置加入许可声明，按照下面附录中给出的形式，以本许可证给公众授于是用修订版本的权利。

G. 保持原文档的许可声明中的全部不可变章节、封面文字和封底文字的声明不变。

H. 包含一份未作任何修改的本协议的副本。

I. 保持命名为“历史”的章节不变，保持它的标题不变，并在其中加入一项，至少声明扉页上的已修改版本的标题、年份、新作者和出版者。如果文档中没有命名为“历史”章节，则请新建它，并加入一项以声明原文档扉页上所列的标题、年份、作者与出版者，再在其后加入如上所说的描述修改版本的项。

J. 如果文档中有用于公共用户访问的文档透明副本的网址，则保持网址不变，并同样提供它所基于的以前文档版本的网址。这些网址可以放在“历史”章节。您可以不给出那些在原文档发行之前已经发行至少四年的版本给出的网址，或者该版本的发行者授权不列出网址。

K. 对于任何命名为“致谢”或“题献”的章节，保持其标题不变，并保持其全部内容以及对每位贡献者致谢和/或题献的语气不变。

L. 保持文档的所有固定章节不变，不改变它们的标题和内容。章节的编号或相当的内容不被认为是章节标题的一部分。

M. 删除命名为“签名”的章节。这样的章节不可以被包含在修改后的版本中。

N. 不要把任何现有章节重命名为“签名”或与任何不可变章节相冲突的标题。

O. 保持任何免责声明不变。

如果修改版本加入了新的符合次要章节定义的引言或附录章节，并且不含有从原文档中复制的内容，您可以选择将其标记为固定。如果需要这样做，则将它们的标题加入修改版本许可声明的不可变章节列表之中。这些标题必须和其他章节的标题相区分。

您可以加入一个命名为“签名”的章节，只要它只包含对您的修改版本由不同的各方给出的签名，例如书评或是声明文本已经被一个组织认定为一个标准的权威定义。

您可以加入一个最多 5 个字的段落作为封面文本和一个最多 25 个字的段落作为封底文本，将它们加入修改版本的封页文本列表末端。一个实体只可以添加（或编排）一段封面和一段封底文本。如果原文档已经为该封页（封面或封底）包含了封页文本，由您或您所代表的实体先前加入或排列的文本，不能再新加入一个，但您可以在原来的发行者的明确许可下替换掉原来的那个。

作者和发行者不能通过本许可证授权公众使用他们的名字推荐或暗示认可任何一个修改版本。

5. 组合文档

遵照第 4 节所说的修改版本的规定，您以将文档和其他文档合并并以本许可证发布，只要您在合并结果中包含原文档的所有不可变章节，对它们不加以任何改动，并在合并结果的许可声明中将它们全部列为不可变章节，而且维持原作者的免责声明不变。

合并作品仅需要包含一份此许可证，多个相同的固定章节可以由一个副本取代。如果有多个名称相同但内容不同的固定章节，通过在章节名称后面的括号中加上原作者或出版者的姓名（如果已知）来加以区别，或者使用唯一编号加以区别。并对合并作品许可声明中的固定章节列表中的章节标题做相同的调整。

在合并过程中，必须合并不同原始文档中任何命名为“历史”的章节，从而形成新的命名为“历史”的章节；类似地，还要合并命名为“致谢”和“题献”的章节。必须删除所有命名为“签名”的章节。

6. 文档的合集

您可以制作一个文档和其他文档的合集，在本许可证下发布，并在合集中将不同文档中的多个本许可证的副本以一个单独的副本来代替，只要您在文档的其他方面遵循本许可证的逐字复制的条款即可。

您可以从一个这样的合集中提取一个单独的文档，并将它在本许可证下单独发布，只要您想这个提取出的文档中加入一份本许可证的副本，并在文档的其他方面遵循本许可证的逐字复制的原则。

7. 独立作品的聚合体

将文档或其派生品以及其他独立和无关文档或作品编撰在一个储存卷中或分发媒体上，这称为文档的“聚合体”，前提是编撰成品的著作权对其使用者的法律权限的限制未超出各个独立作品的许可范围。当基于此许可证发布的文档包含在一个聚合体中时，此许可证不适用于聚合体中的本非该文档派生作品的其他作品。

如果第 3 节中的封页文本要求适用于这些文档的副本，则若文档在聚合体中所占的比重小于全作品的一半，文档的封页文本可以放置在聚合体内包含文档部分的封页上，或是电子文档中的等效部分。否则，它必须位于整个聚合体的印刷的封页上。

8. 翻译

翻译被视为一种修改，因此您可以根据第 4 节的条款分发文档的翻译。将固定章节替换为翻译内容需要经得其版权所有者的特别许可，但除了这些固定章节的原始版本之外，您还可以包含一部分或所有固定章节的翻译。您可以包含一个此许可证以及所有许可证声明和免责声明的翻译版本，前提是同时包含它们的原始英文版本。当翻译版本和英文版发生冲突的时候，原始版本有效。

如果在文档中有命名为“致谢”、“题献”或“历史”的章节，保持标题（第 1 节）的要求（第 4 节）恰恰需要更换实际的标题。

9. 终止

除非此许可证中有明确规定，否则您不能对该文档进行复制、修改、分授许可或分发。在此许可证规定外对该文档所进行的任何复制、修改、分授许可或分发都是无效的，并且将自动终止您在此许可证下所拥有的权利。但是，对于在此许可证的规定下从您这里获得副本或权利的各方，只要其完全遵守此许可证的规定，其许可证将不会被终止。

10. 本许可的未来修订版本

自由软件基金会有时会发布 GNU 自由文档许可证的新的修订版版本。这些新版本的主旨和精神与当前版本是一致的，但在解决新问题的具体细节方面可能有所不同。请参见 https://www.gnu.org/copyleft/。

许可证的每个版本都有一个不同的版本号。如果文档指定了适用于它的此许可证“或任何后续版本”的特定带编号版本，则您可以选择遵从指定版本或自由软件基金会发布的任何随后版本（非草稿）的条款和条件。如果文档没有指定此许可证的版本号，您可以选择自由软件基金会发布的任何许可证版本（非草稿）。

附录：如何针对您的文档使用此许可证

```
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”.
```

如果您有固定章节、封面文本和封底文本，请将“with...Texts”部分替换为：

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

如果有不可变章节而没有封页文本，或这三种内容（不可变章节、封面文本、封底文本）的任何其他组合，请合并这两个备选项以适应您的情况。

如果您的文档包含不一般的程序代码示例，建议同时选择自由软件许可证（如 GNU 通用公共许可证）发布这些示例，以允许它们可以用于自由软件。