SUSE Best Practices

SAP

Infrastructure Monitoring for SAP Systems

SUSE Linux Enterprise Server 15 SP7 and later

Thomas Schlosser, SAP Solution Architect (SUSE)

Infrastructure Monitoring for SAP Systems

Date: 2026-05-21

This guide provides detailed instructions on installing and customizing SUSE Linux Enterprise Server and Grafana Alloy to monitor hardware metrics. These insights help maximize the uptime of critical SAP applications. While based on SUSE Linux Enterprise Server 15 SP7, the concepts are also applicable to SUSE Linux Enterprise Server 16.0.

Contents

1 Introduction

Many customers deploy SAP systems, such as SAP S/4HANA, to support mission-critical business functions for their global operations. Consequently, maximizing system availability becomes crucial. IT departments face very demanding SLAs, with many companies now requiring 24x7 reliability for their SAP environments.

The foundation of every SAP system is a solid infrastructure supporting it.

Operating System

SUSE Linux Enterprise Server for SAP applications is the leading Linux platform for SAP HANA, SAP NetWeaver and SAP S/4HANA solutions. It helps reduce downtime by providing the flexibility to configure and deploy various HA/DR scenarios for SAP HANA and NetWeaver-based applications. Additionally, system data monitoring enables proactive problem avoidance.

Hardware

Most modern hardware platforms running SAP systems rely on Intel's system architecture. The combination of SUSE Linux Enterprise Server, the latest generation of Intel Xeon Scalable processors, and Intel Optane DC persistent memory helps deliver fast, innovative, and secure IT services while providing resilient enterprise S/4HANA platforms. The Intel platform allows for deep hardware monitoring, gaining insights into system performance at the physical level. This level of monitoring helps reduce SAP systems downtime in several ways:

Failure prediction

Identifying hardware failures in advance allows customers to react early and in a scheduled manner. This reduces the risk of errors that often occur during emergency operations under the pressure of an unplanned outage.

Failure remediation

Having hardware metrics readily available when diagnosing a root cause of an issue speeds up analysis and reduces the time required to restore operations on that system. It also improves reaction time by providing more precise data about an issue. This is especially valuable for enterprise customers who usually outsource operations to multiple service providers and do not control the environment directly.

This paper describes a monitoring solution for SAP systems that enables metrics to be analyzed within a specific SAP context.

2 Monitoring for SAP systems overview

The solution presented in this document consists mainly of the open source tool Grafana Alloy. The tool acts as a single, local agent that replaces most single exporters by gathering metrics and logs directly via built-in collector functions (Data Collection). It transforms metrics and logs via relabeling and filtering (Data Processing), and forwards the unified telemetry to central backends via a push-based model (Data Push).

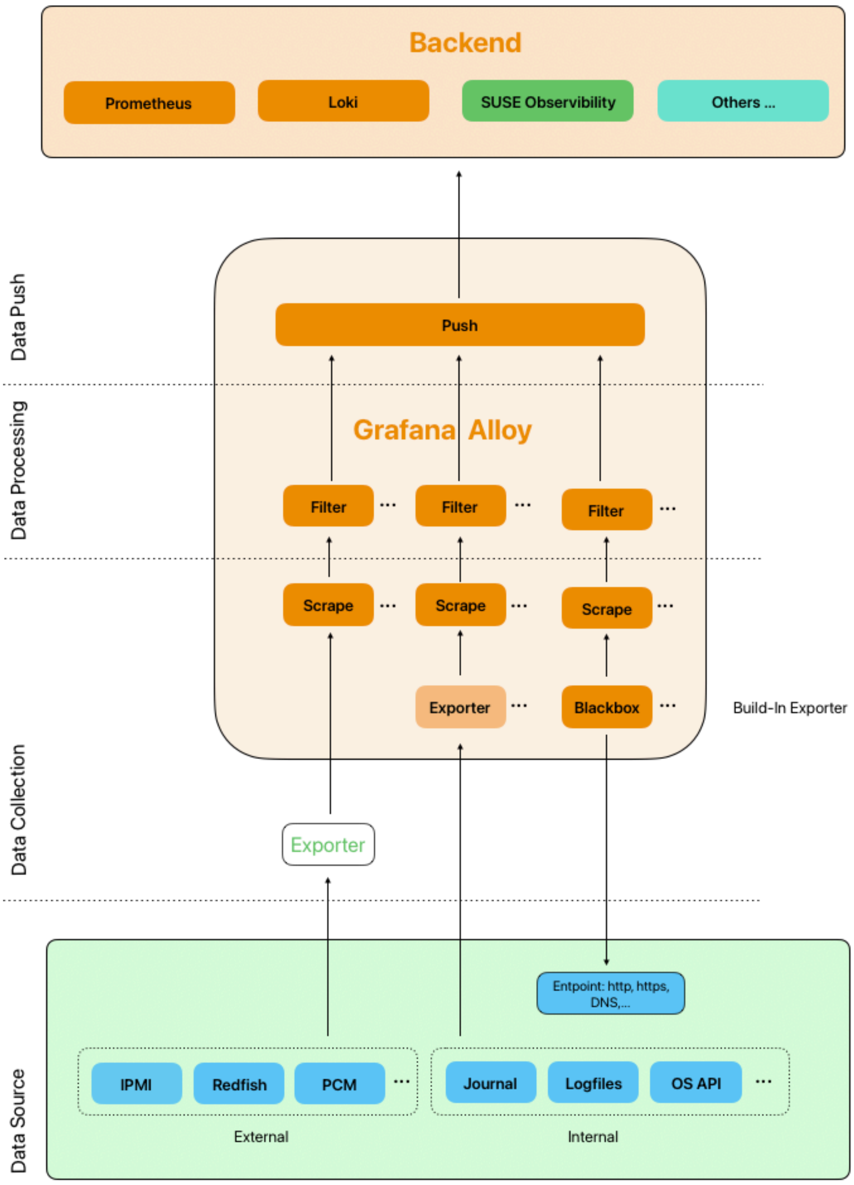


FIGURE 1: MONITORING COMPONENTS

2.1 Data Source

As illustrated in the diagram, **Data Sources** can be categorized by the operating system (Internal Sources) and the underlying hardware (External Sources):

- **Internal Sources:** Information from the operating system itself, such as log files, the systemd journal, and system APIs.
- **External Sources:** Information from outside the main OS, such as hardware data via SNMP and Redfish, or network checks via Blackbox.

Note

In the previous SUSE Best Practices document, the **Data Source** was defined by the exporters and collectors themselves. We now define the **Data Source** strictly as the origin of the information, independent of the collection tool.

2.2 Data Collection

Data Collection can be performed in two ways: Grafana Alloy either gathers information directly using its built-in exporter function, or it acts as a local receiver for external exporters.

- **Built-in function:** Grafana Alloy features many native exporter functions (such as the legacy Node Exporter and Promtail).
- **Local aggregator:** For specialized datasets, Grafana Alloy acts as a local aggregator for third-party collectors and exporters (for example: PCM, Collectd),

Note

In the previous SUSE Best Practices document, the **Data Collection** was defined by the storage component (Prometheus and Loki), which had to actively "pull" data from every individual exporter across the network. We have now shifted this function to a dedicated layer on the host, centralizing the gathering process locally.

2.3 Data Processing

Data Processing can clean and organize metrics and logs, before the data leaves the system.

- **Relabeling:** Labels can be added or changed on individual data points. This ensures data is easy to find, group and query across all subsequent stages.
- **Filtering:** Unnecessary data can be dropped or reduced to only the essential parts.

2.4 Data Push

Finally, the data is sent to the central databases (for example, Prometheus and Loki). The key benefits of the Alloy-driven approach are:

- **Enhanced Security (Push Approach):** Using a push-based model eliminates the need to open multiple incoming ports (for example, port 9100 for Prometheus Node Exporter) on every host. Grafana Alloy pushes data out, allowing your systems to remain closed and secure.
- **Unified Stream:** Metrics, logs, and traces are pushed together, making network traffic much more efficient.

3 Implementing the monitoring solution

The following sections show how to set up a monitoring solution based on Grafana Alloy that was introduced in the solution overview.

3.1 Installing Grafana Alloy

Grafana Alloy can be installed directly from the SUSE repository. With SUSE Linux Enterprise Server 15 SP7 and SUSE Linux Enterprise Server 16.0., it is part of SUSE Linux Enterprise Server.

```
# zypper in alloy
```

3.2 Base Configuration

This section describes the base configuration of Grafana Alloy. It covers the Alloy service configuration, data collection, data processing, and data push.

3.2.1 Environment File

The service settings are defined in a central environment file. The `/etc/sysconfig/alloy` file controls how the service is launched. The main configuration options are explained below:

/etc/sysconfig/alloy

```
## Path:
## Description: Grafana Alloy settings
## Type:        string
## Default:     ""
## ServiceRestart: alloy
#
# Command line options for Alloy.
#
# The configuration file holding the Alloy config.
CONFIG_FILE="/etc/alloy/config.alloy"

# User-defined arguments to pass to the run command.
CUSTOM_ARGS="--server.http.listen-addr=0.0.0.0:12345"

# Restart on system upgrade. Defaults to true.
RESTART_ON_UPGRADE=true

# Monitoring Group
monitoring_group=infra
```

CONFIG_FILE

Defines the path to the main Alloy configuration. By default, it points to a single file, but is also supports directory-based loading where all files within that folder ending in `.alloy` are used at once.

CUSTOM_ARGS

Allows you to pass additional command-line flags to the service during startup. It is used to enable extra features, adjust performance limits, or fine-tune global settings.

monitoring_group

The `monitoring_group` key-value pair is an example of creating an environment variable. This example sets a custom environment label, similar to the traditional Prometheus `job_name`. It can be used later on in the Alloy configuration via `sys.env("monitoring_group")`.

3.3 Metric configuration

Alloy operates as a programmable pipeline where modular components are linked to manage the metric flow. This architecture allows you to connect specialized exporters, relabeling rules, and scraping engines to transform raw system data into structured time-series metrics. The default config file is stored under:

```
/etc/alloy/config.alloy
```

The following sections describe the individual components used in the configuration. A complete, functional example combining all these elements is provided at the end of this chapter.

3.3.1 Data Collection

The Exporter is the data entry point for the pipeline. Exporters do not send data themselves; instead, they "expose" a list of values that are collected by a scraper later in the process. The following sections discuss the most essential exporters for a standard infrastructure deployment.

prometheus.exporter.unix

The `prometheus.exporter.unix` component is the integrated version of the classic Prometheus Node Exporter. Its primary role is to retrieve hardware and operating system metrics directly from the Linux kernel.

```
prometheus.exporter.unix "unix" {
}
```

prometheus.exporter.process

The `prometheus.exporter.process` component provides granular visibility into individual processes running on the host. While the Unix exporter (`prometheus.exporter.unix`) looks at the "big picture" of the entire server, the Process exporter allows you to zoom in on specific applications.

As shown in the example, this component is configured to monitor `nginx` as the specific process.

```
prometheus.exporter.process "process" {
  track_children = false

  matcher {
    comm = ["nginx"]
  }
}
```

prometheus.exporter.blackbox

The `prometheus.exporter.blackbox` component monitors the availability of services by sending network probes to specific endpoints. It tests the availability and the response quality of network services, such as Websites, APIs, or DNS servers, to ensure they are responding within acceptable limits.

```
prometheus.exporter.blackbox "blackbox" {
  config = "{ modules: { http_2xx: { prober: http, timeout: 5s } } }"
  target {
      name    = "Webserver"
      address = "https://webserver01.domain"
      module  = "http_2xx"
    }
}
```

prometheus.exporter.snmp

The `prometheus.exporter.snmp` component enables Grafana Alloy to collect telemetry from network hardware by using the Simple Network Management Protocol (SNMP) to query devices like routers, switches, and similar. The example below collects SNMP data from a Cisco switch. (Note that this component requires an accompanying configuration file.)

```
prometheus.exporter.snmp "cisco_switches" {
  config_file = "/etc/alloy/snmp.yml"

  target "cbs250_main" {
    address = "192.168.178.2"
    module  = "if_mib"
    auth    = "alloy_v2c"
  }
}
```

prometheus.exporter.self

The `prometheus.exporter.self` component is a specialized internal monitor designed for the health of Alloy itself. It allows the Alloy agent to collect and expose its own operational metrics.

```
prometheus.exporter.self "self" {
}
```

3.3.2 Data Collection - Scraper

The scrape component is the active "worker" of the metrics pipeline. While exporters are responsible for making data available, the scraper is responsible for collecting that data. It functions by "pulling" metrics from one or more exporters at a defined time interval.

prometheus.scrape

In the example below, the scraper pulls the data from the `prometheus.exporter.unix` and forwards it to the Data Push component `prometheus.remote_write`:

```
prometheus.scrape "os" {
    scrape_interval = "15s"
    targets     = prometheus.exporter.unix.os.targets
    forward_to = [ prometheus.remote_write.to_prometheus.receiver]
}
```

3.3.3 Data Processing

When telemetry has been identified and collected, it often requires modification before being sent to the backend. Alloy uses relabeling components to standardize data, remove sensitive information, or add organizational context.

prometheus.relabel

The `prometheus.relabel` component is the standard component for modifying metrics after they have been scraped. It acts as a filter and transformation engine that typically sits between the scraper and the Data Push components. The example below creates or modifies the following labels:

```
prometheus.relabel "to_be_compatible" {
    forward_to = [ prometheus.remote_write.to_prometheus.receiver, ]

    // rule01
    rule {
        source_labels  = ["exporter"]
        target_label   = "exporter"
        replacement    = "alloy"
    }


    // rule02
    rule {
        source_labels  = ["instance"]
        target_label   = "instance"
        replacement    = constants.hostname
    }

    // rule03
    rule {
        source_labels  = ["job"]
        target_label   = "job"
        replacement    = sys.env("monitoring_group")
    }
}
```

- **Identifying the Agent (rule01):** This rule ensures the data is tagged with the name `alloy` so you can easily identify which software collected the metrics.
- **Standardizing the Hostname (rule02):** This rule replaces the technical address (such as an IP) with the actual system host name, making it easier to find the server.
- **Using Environment Variables (rule03):** This rule grabs the `monitoring_group` variable from your system settings (see: `/etc/sysconfig/alloy`) and works like the legacy Prometheus `job_name`. By setting this to values like `infra`, `test`, or `services`, you can easily categorize and filter your targets in Grafana.

3.3.4 Data Push

prometheus.remote_write

The `prometheus.remote_write` component is the final destination handler for metrics within the Alloy pipeline. When data has been collected by a scraper and processed by any relabeling stages, this component handles the secure transmission of that data to a backend, such as Prometheus.

```
prometheus.remote_write "to_prometheus" {
    endpoint { url = "http://prometheus-server:9090/api/v1/write" }
}
```

3.3.5 Putting it all together

To provide a clear picture of the entire configuration, the following example shows a complete Alloy setup. The `prometheus.exporter.unix` component is the best example of a native replacement for the traditional Prometheus Node Exporter. In this scenario, Alloy scrapes the internal exporter data, applies custom labels to identify the source, and forwards the processed data to a central Prometheus backend.

```
prometheus.exporter.unix "localhost" {
}

prometheus.scrape "unix" {
    scrape_interval = "10s"
    targets     = prometheus.exporter.unix.localhost.targets
    forward_to = [ prometheus.relabel.to_be_compatible.receiver, ]
}

prometheus.relabel "to_be_compatible" {
    forward_to = [ prometheus.remote_write.to_prometheus.receiver, ]

    rule {
        source_labels  = ["exporter"]
        target_label   = "exporter"
        replacement    = "alloy"
    }
    rule {
        source_labels  = ["instance"]
        target_label   = "instance"
```

```
        replacement    = constants.hostname
    }
    rule {
        source_labels  = ["job"]
        target_label   = "job"
        replacement    = sys.env("monitoring_group")
    }
}

prometheus.remote_write "to_prometheus" {
    endpoint { url = "http://prometheus-server:9090/api/v1/write" }
}
```

3.4 Log configuration

Following the same logic as metrics, the log components allow you to collect, parse and filter system events before they reach the backend.

3.4.1 Data Collection

Unlike metric exporters, which take a snapshot at specific time intervals, log components stay "attached" to the files. As soon as a new line is written to a log file, the collector detects it and sends it immediately (similar to the `tail -f` command).

The following sections outline the essential log components.

loki.source.journal

The `loki.source.journal` component reads logs directly from the systemd journal. It retreives not only the log message itself but also a complete set of metadata, such as the unit name, PID, and priority.

```
loki.source.journal "journal" {
  max_age       = "24h0m0s"
  relabel_rules = discovery.relabel.journal.rules
  forward_to    = [loki.relabel.journald.receiver]
}
```

loki.source.file

The `loki.source.file` component is the standard tool for collecting logs from traditional text files stored on the local file system.

```
loki.source.file "logfile" {
  targets    = [
    {__path__ = "/var/log/logfile.log"},
  ]
  forward_to    = [loki.relabel.logfiles.receiver]
}
```

3.4.2 Data Processing

Similar to metric data, logs often require modifications before being sent to the backend.

loki.relabel

Like `prometheus.relabel`, the `loki.relabel` component is the standard component for modifying logs after they have been captured.

```
loki.relabel "logfiles" {
  forward_to    = [loki.write.to_loki.receiver]

  rule {
    source_labels  = ["job"]
    target_label   = "job"
    replacement    = sys.env("monitoring_group")
   }

  rule {
    source_labels  = ["log_type"]
    target_label   = "log_type"
    replacement    = "journald02"
   }
}
```

3.4.3 Data Push

loki.write

The `loki.write` cocmponent serves as the final destination handler for logs within the Grafana Alloy pipeline. When log entries have been collected and processed through various stages, this component handles the transmission of the resulting streams to a log backend, such as Grafana Loki.

```
loki.write "to_loki" {
    endpoint { url = "http://loki-server:3100/loki/api/v1/push" }
}
```

3.4.4 Putting it all together

To provide a clear picture of how Grafana Alloy handles logging, the example below shows a complete configuration for `journald`. In this setup, Alloy replaces the traditional Promtail agent by reading system logs directly, adding custom labels for better searchability, and forwarding them to a central Loki backend.

```
loki.source.journal "journal" {
  max_age       = "24h0m0s"
  forward_to    = [loki.relabel.journald.receiver]
  labels        = {log_type = "journald"}
}


loki.relabel "journald" {
  forward_to    = [loki.write.to_loki.receiver]
  rule {
    source_labels = ["__journal__systemd_unit"]
    target_label  = "unit"
  }
  rule {
    source_labels = ["__journal__hostname"]
    target_label  = "hostname"
  }
  rule {
    source_labels = ["__journal__kernel_device"]
    target_label  = "kernel_device"
  }
```

```
  rule {
    source_labels = ["__journal__priority"]
    target_label  = "prio"
  }
  rule {
    source_labels  = ["instance"]
    target_label   = "instance"
    replacement    = constants.hostname
  }
  rule {
    source_labels  = ["job"]
    target_label   = "job"
    replacement    = sys.env("monitoring_group")
   }
  rule {
    source_labels  = ["log_type"]
    target_label   = "log_type"
    replacement    = "journald"
   }
}

loki.write "to_loki" {
    endpoint { url = "http://loki-server:3100/loki/api/v1/push" }
}
```

4 Migrating to Grafana Alloy

Migrating to Grafana Alloy is straightforward and does not require starting from scratch. It also supports a mixed environment, providing the flexibility to integrate existing exporters alongside the new system. This ensures a safe transition with no data loss.

4.1 Migrating from metrics exporter

The metrics migration process is straightforward because Grafana Alloy is natively compatible with the Prometheus ecosystem. Most standard collectors are now built-in, allowing you to replace external binaries with the internal components described above. No formal migration tool is required for this process; instead, you can simply follow the configuration steps outlined above.

4.1.1 Keeping existing exporters

For cases where a specialized exporter is not yet built into, you can use Grafana Alloy as a local scraper to collect that data from that existing (external) exporter. This still gives you the power to filter and relabel your metrics before they are sent to the backend.

Integrating Intel Processor Counter Monitor (PCM)

One example of integrating special exporter is the Intel Processor Counter Monitor (PCM).

Intel Processor Counter Monitor (PCM) is a specialized tool used to monitor real-time performance and energy metrics of Intel processors, such as memory bandwidth and CPU resource utilization.

Unlike the previous SUSE Best Practices guide, this integration runs the PCM sensor within a Docker container, ensuring access to the latest version in an isolated environment.

```
# modprobe msr

# docker run -d --name pcm --privileged -p 9738:9738 ghcr.io/intel/pcm
```

Grafana Alloy can then scrape these hardware metrics locally, apply custom labels, and forward the data to a central Prometheus backend.

```
prometheus.scrape "intel_pcm" {
  targets = [
    {"__address__" = "localhost:9738"},
  ]
  forward_to = [ prometheus.relabel.to_be_compatible.receiver, ]
}


prometheus.relabel "to_be_compatible" {
    forward_to = [ prometheus.remote_write.to_prometheus.receiver, ]

    rule {
        source_labels  = ["exporter"]
        target_label   = "exporter"
        replacement    = "alloy"
    }
    rule {
        source_labels  = ["instance"]
        target_label   = "instance"
        replacement    = constants.hostname
```

```
    }
    rule {
        source_labels  = ["job"]
        target_label   = "job"
        replacement    = sys.env("monitoring_group")
    }
}


prometheus.remote_write "to_prometheus" {
    endpoint { url = "http://prometheus-server:9090/api/v1/write" }
}
```

4.2 Migrating from Promtail log exporter

As Promtail has officially reached EOL on March 2, 2026, Grafana Alloy has taken over its role as the primary logging agent. To simplify the transition, the `alloy convert` utility automatically translates your existing Promtail configurations into the new Alloy syntax.

The command is straightforward and requires two components: the source Promtail configuration file and a target output path.

```
# alloy convert --source-format=promtail --output=<OUTPUT_CONFIG_PATH> /etc/loki/
promtail.yaml
```

For demonstration, the following `promtail.yaml` file is used:

```
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /var/lib/promtail/positions.yaml

clients:
  - url: http://loki-host:3100/loki/api/v1/push

scrape_configs:
- job_name: journal
  journal:
    max_age: 24h
    labels:
      job: loki_messages

  relabel_configs:
```

```
    - source_labels: ['__journal__systemd_unit']
      target_label: 'unit'

    - source_labels: ['__journal__hostname']
      target_label: 'host'
```

The conversion tool reads the Promtail file and automatically changes the settings into the correct Alloy format. Execute the following command to start the process:

```
alloy convert --source-format=promtail --output=/tmp/config.alloy /etc/loki/promtail.yml
```

This generates the following Grafana Alloy configuration:

```
discovery.relabel "journal" {
  targets = []

  rule {
    source_labels = ["__journal__systemd_unit"]
    target_label  = "unit"
  }

  rule {
    source_labels = ["__journal__hostname"]
    target_label  = "host"
  }
}

loki.source.journal "journal" {
  max_age       = "24h0m0s"
  relabel_rules = discovery.relabel.journal.rules
  forward_to    = [loki.write.default.receiver]
  labels        = {
    job = "loki_messages",
  }
}

loki.write "default" {
  endpoint {
    url = "http://loki-host:3100/loki/api/v1/push"
  }
  external_labels = {}
}
```

5 Troubleshooting

5.1 Alloy Web UI

The Grafana Alloy Web GUI provides a wealth of information. By default, it can usually only be accessed locally via:

```
http://localhost:12345
```

External Access

By setting the option `--server.http.listen-addr=0.0.0.0:12345`, the Grafana Alloy UI can also be accessed externally. When enabling external access, ensure the port is protected by a firewall or a reverse proxy, as Grafana Alloy does not provide native authentication for its Web interface.

This option can be configured in `/etc/sysconfig/alloy` under `CUSTOM_ARGS` (see section 3.1.1 Environment File).

```
[...]

# User-defined arguments to pass to the run command.
CUSTOM_ARGS="--server.http.listen-addr=0.0.0.0:12345"

[...]
```

5.1.1 Component health

The first step is to verify the health of each individual component. This status overview is available directly on the main page of the Alloy Web UI. The dashboard provides a real-time status overview of every active component in the pipeline.

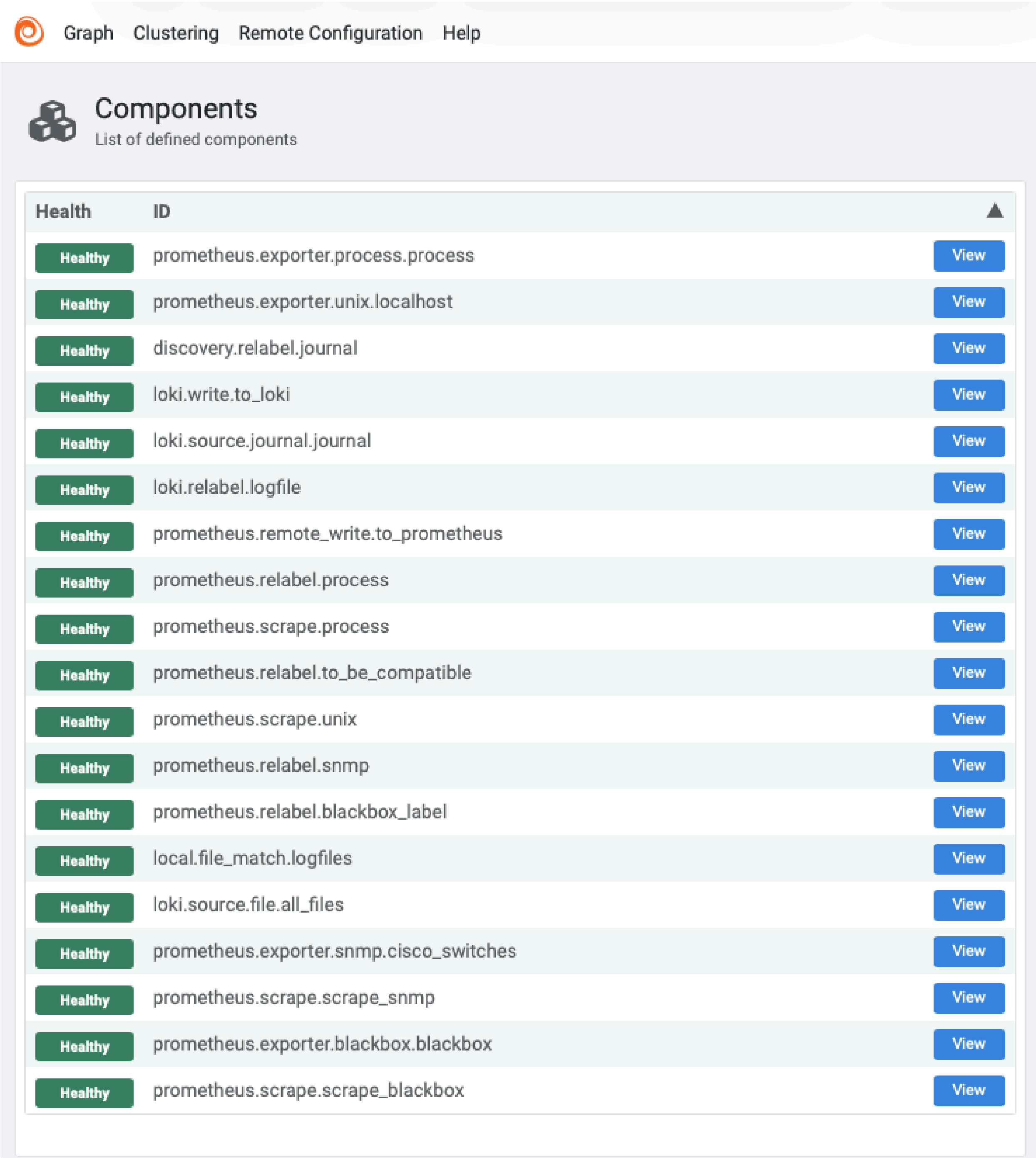


FIGURE 2: COMPONENT HEALTH STATUS

5.1.2 Verifying Data Flow

The Graph tab shows a visual map of all active pipelines. It displays how components connect and how data flows between them.

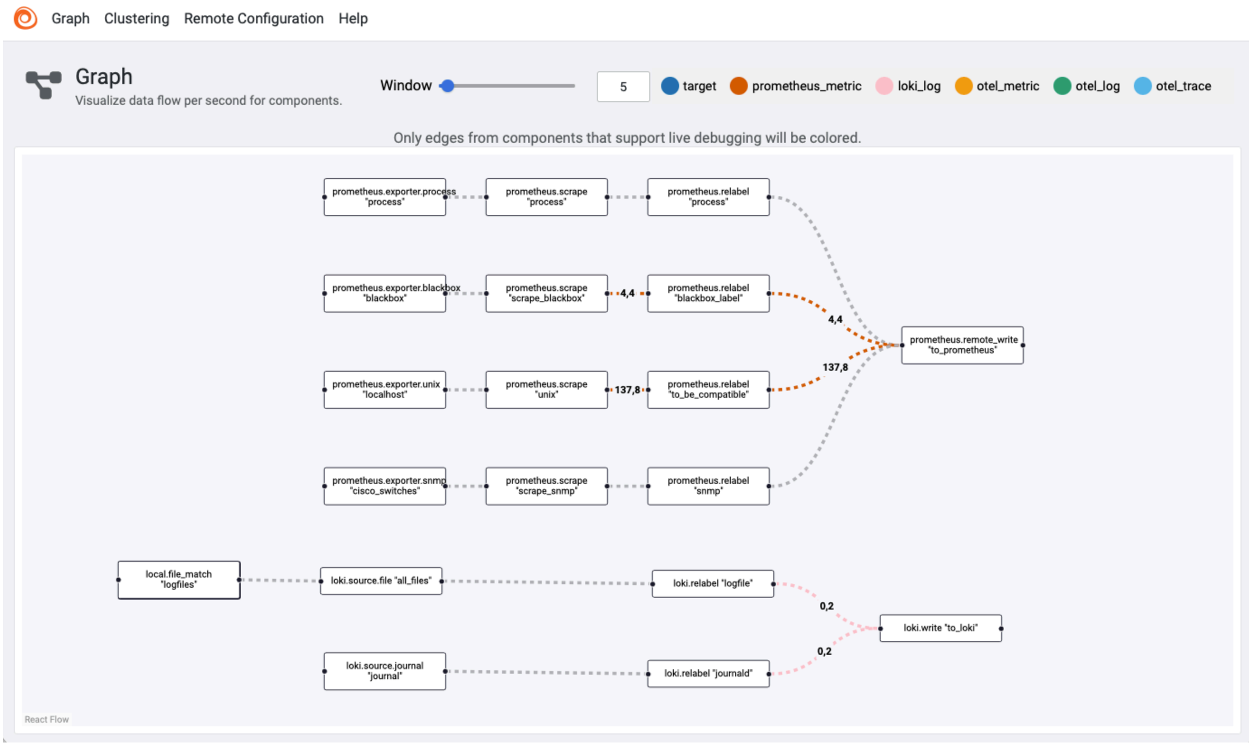
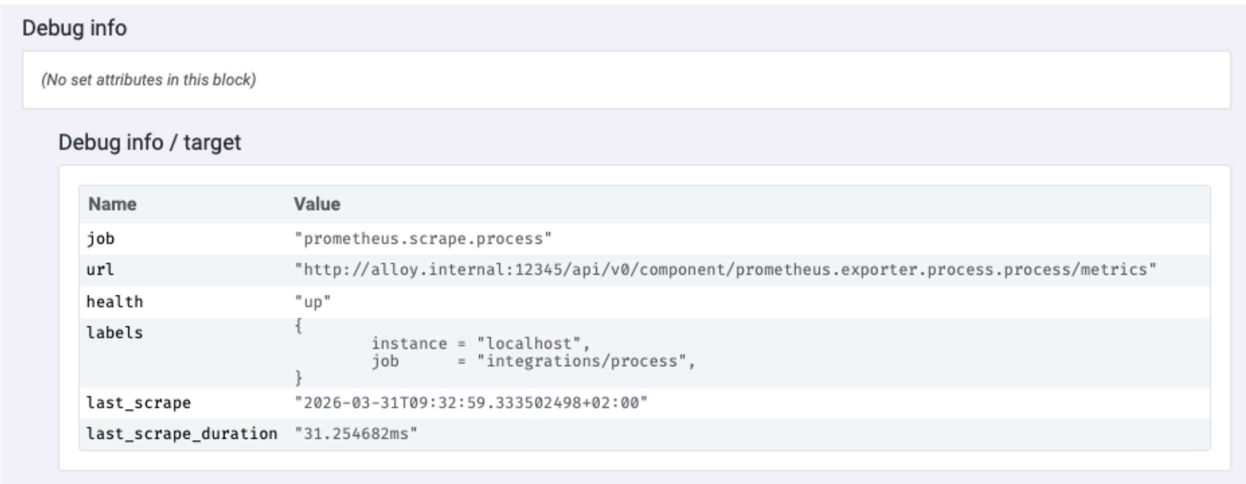


FIGURE 3: VISUALIZING DATA FLOW

5.1.3 Checking the scrape debug information

Because Grafana Alloy is scraping the data from build-in and external exporters, it is important to check the scrape data. The following illustration highlights essential debug information.



Debug info

(No set attributes in this block)

Debug info / target

Name	Value
job	"prometheus.scrape.process"
url	"http://alloy.internal:12345/api/v0/component/prometheus.exporter.process.process/metrics"
health	"up"
labels	{ instance = "localhost", job = "integrations/process", }
last_scrape	"2026-03-31T09:32:59.333502498+02:00"
last_scrape_duration	"31.254682ms"

FIGURE 4: VISUALIZING THE SCRAPE COMPONENT

job

The **Job** field simply identifies the component type.

url

By accessing this URL endpoint in a browser, you can verify the raw metric data. If the output contains only a few lines of metrics and these lines point only to the scrape values themselves (for example to `scrape_errors`), Grafana Alloy is failing to retrieve data from the exporter.

health

It is important to note that a status of "Up" only indicates that the component itself is running and functional. This status does not guarantee that data is being successfully scraped or delivered.

labels

Displays the labels and their current values at this stage of the pipeline. Keep in mind that a subsequent relabeling component could change or even remove these labels.

last_scrape

This field indicates the exact timestamp of the last successful scrape performed by the component.

last_scrape_duration

This value represents the total time required to scrape all data from the target. An unexpectedly short duration may indicate that the component is failing to retrieve the full dataset. Conversely, an excessively long scrape duration could suggest network latency, resource bottlenecks, or a target that is struggling to respond.

5.2 Live Debugging

This feature allows you to inspect live data as it moves through the pipeline in real time. It is used for verifying the output of specific components, ensuring that labels and values are correctly assigned after the data has been processed.

To uilize this functionality, specific debugging options must be enabled within the Alloy configuration file.

```
livedebugging {
  enabled = true
}
```

Keep in mind that this feature is not available on all components. Live debugging is currently available only for a specific subset of components:

- loki.relabel
- prometheus.relabel
- prometheus.scrape
- prometheus.remote_write

To view live data, select one of the supported components mentioned above and click the blue **Live Debugging** button at the top.

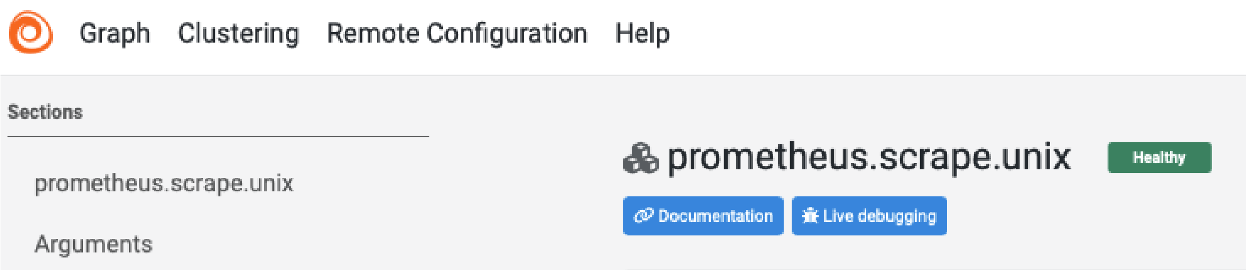


FIGURE 5: VISUALIZING THE SCRAPE COMPONENT STEP 1

The **Live Debugging** view opens, displaying a real-time stream of data.

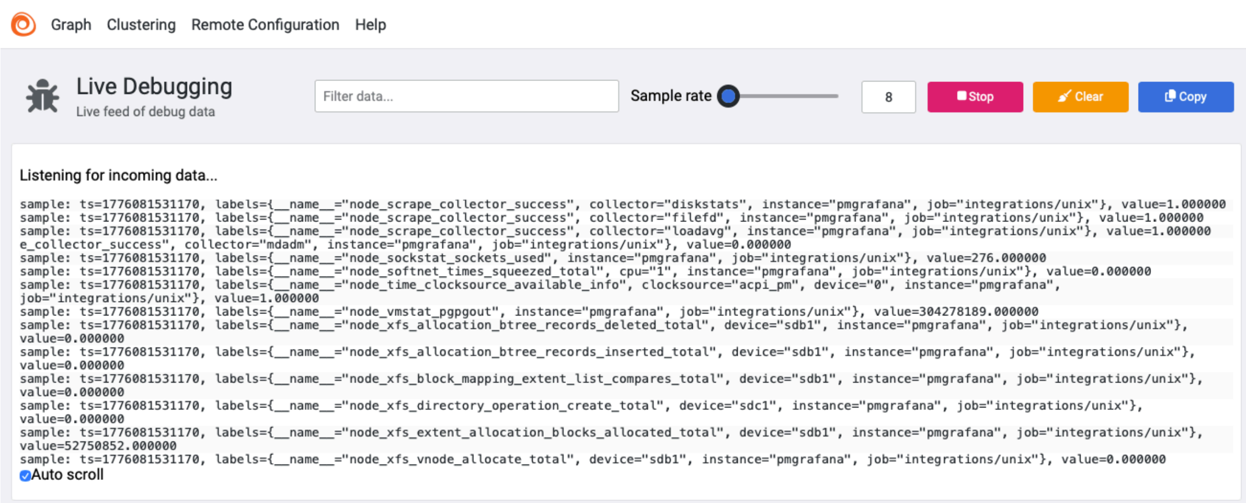


FIGURE 6: VISUALIZING THE SCRAPE COMPONENT STEP 2

> **Note**
>
> While live debugging is highly effective for resolving configuration issues, it should not be left enabled in high-traffic production environments. This feature consumes additional CPU and memory to buffer data for real-time visualization, which can impact overall system performance.

5.3 Support Bundle

The support bundle provides a collection of diagnostic data from the running Grafana Alloy instance. It creates a compressed file containing the current configuration and logs.

The bundle can be generated locally by running the following command:

```
# curl -o support-bundle.zip  http://localhost:12345/-/support
```

5.4 Verifying permissions

Alloy runs under the `alloy` user account. By default, this user has the following permissions:

```
# id alloy
uid=461(alloy) gid=471(alloy) Gruppen=476(systemd-journal),471(alloy)
```

The `alloy` user may encounter issues reading log files if the correct permissions are not configured. You can verify whether the user has the necessary access to a specific file by executing the following command:

```
# sudo -u alloy head -n 3 /var/log/zypper.log
head: cannot open '/var/log/zypper.log' for reading: Permission denied
```

6 Examples

This section provides configuration examples for different use cases.

6.1 Blackbox

The configuration example below shows how to use the blackbox exporter to monitor the availability of API-endpoints or Websites.

```
prometheus.exporter.blackbox "blackbox" {
  config = "{ modules: { http_2xx: { prober: http, timeout: 5s } } }"
  target {
      name    = "SUSE"
      address = "https://suse.com"
      module  = "http_2xx"
    }
  target {
      name    = ""
      address = "https://rancher.com"
      module  = "http_2xx"
    }
}

prometheus.scrape "scrape_blackbox" {
    scrape_interval = "15s"
    targets    = prometheus.exporter.blackbox.blackbox.targets
    forward_to = [ prometheus.relabel.blackbox_label.receiver]
}

prometheus.relabel "blackbox_label" {
    rule {
        source_labels = ["job"]
        regex         = ".*/(.+)"
        target_label  = "instance"
        replacement   = "$1"
    }

    rule {
        target_label = "job"
        replacement  = "blackbox"
    }

    forward_to = [prometheus.remote_write.to_prometheus.receiver]
}

prometheus.remote_write "to_prometheus" {
    endpoint { url = "http://prometheus-server:9090/api/v1/write" }
}
```

6.2 SNMP

The configuration example below shows how to use the SNMP exporter to collect SNMP traps.

```
// wget https://raw.githubusercontent.com/prometheus/snmp_exporter/main/snmp.yml -O /etc/
alloy/snmp.yml

prometheus.exporter.snmp "cisco_switches" {
  config_file = "/etc/alloy/snmp.yml"
  target "cbs250_main" {
    address = "192.168.178.2"
    module  = "if_mib"
    auth    = "alloy_v2c"
  }
}

prometheus.scrape "scrape_snmp" {
  targets         = prometheus.exporter.snmp.cisco_switches.targets
  scrape_interval = "60s"
  scrape_timeout  = "55s"
  forward_to      = [prometheus.relabel.snmp.receiver]
}

prometheus.relabel "snmp" {
    forward_to = [ prometheus.remote_write.to_prometheus.receiver, ]
    rule {
        target_label   = "instance"
        replacement    = "cisco"
    }
    rule {
        target_label   = "job"
        replacement    = "snmp"
    }
}

prometheus.remote_write "to_prometheus" {
    endpoint { url = "http://prometheus-server:9090/api/v1/write" }
}
```

6.2.1 Process

The configuration example below shows how to use the process exporter to monitor the status of a process in the operating system.

```
prometheus.exporter.process "process" {
  track_children = false
  matcher {
    comm = ["nginx"]
  }
}

prometheus.scrape "process" {
  targets     = prometheus.exporter.process.process.targets
  forward_to = [prometheus.relabel.process.receiver]
}

prometheus.relabel "process" {
    forward_to = [ prometheus.remote_write.to_prometheus.receiver, ]
    rule {
        target_label    = "exporter"
        replacement     = "alloy"
    }
    rule {
        target_label    = "job"
        replacement     = "process"
    }
}

prometheus.remote_write "to_prometheus" {
    endpoint { url = "http://prometheus-server:9090/api/v1/write" }
}
```

6.2.2 Logfiles

The configuration example below shows how to use the `file_match` functionality to collect log files.

```
local.file_match "logfiles" {
    path_targets = [
        { "__path__" = "/var/log/messages" },
        { "__path__" = "/var/log/warn" },
        { "__path__" = "/var/log/*.log" },
    ]
}

loki.source.file "all_files" {
    targets     = local.file_match.logfiles.targets
    forward_to     = [loki.relabel.logfile.receiver]
}

loki.relabel "logfile" {
  forward_to = [loki.write.to_loki.receiver]
  rule {
    source_labels  = ["instance"]
    target_label   = "instance"
    replacement    = constants.hostname
  }
  rule {
    source_labels  = ["job"]
    target_label   = "job"
    replacement    = sys.env("monitoring_group")
   }
  rule {
    target_label = "log_type"
    replacement  = "files"
  }
}

loki.write "to_loki" {
    endpoint { url = "http://loki-server:3100/loki/api/v1/push" }
}
```

7 Miscellaneous

Reloading the configuration

After modifying the configuration file, use the following command to apply the changes without restarting the service:

```
curl -X POST 127.0.0.1:12345/-/reload
```

Splitting configuration files

In complex environments, splitting the `config.alloy` file into separate files can improve organization and simplify management. For example, individual pipelines can be stored in dedicated files such as `unix.alloy` or `journald.alloy`.

Common components, such as `prometheus.remote_write` or `loki.write`, can also be placed in a single shared file. This allows multiple pipelines to push data to the same destination without duplicating the configuration.

To enable this, modify the `CONFIG_FILE` variable in `/etc/sysconfig/alloy` as follows:

```
# The configuration file or directory holding the Alloy config.
CONFIG_FILE="/etc/alloy/"
```

Alloy automatically loads and merges all files within that folder ending in `.alloy`, allowing them to function as a single configuration.

Directory Structure Example

When `CONFIG_FILE` is set to `/etc/alloy/`, the directory can be organized as follows:

- **unix.alloy:** Contains components for collecting Linux system metrics (for example, `prometheus.exporter.unix`, `prometheus.relabels`).
- **journald.alloy:** Contains components for collecting system logs (for example, `loki.source.journal`, `loki.relabel`).
- **outputs.alloy:** Contains the central `prometheus.remote_write` and `loki.write` components used by all pipelines.

8 Summary

Maintaining the availability of SAP S/4HANA is critical for ensuring uninterrupted business operations. This document has demonstrated how Grafana Alloy provides a modern, unified approach to monitor these systems, enabling administrators to identify and resolve issues before they cause costly downtime.

The primary benefits include:

- **All-in-One Tool:** One single agent handles metrics, logs, and traces, significantly simplifying the overall monitoring infrastructure.
- **Streamlined Troubleshooting:** Tools like the Graph view and Live Debugging allow you to quickly isolate and fix pipeline or configuration errors.
- **Backend Flexibility:** The data destination (backend) is easily interchangeable without requiring changes to the core collection logic.

By switching to Grafana Alloy, you gain a more efficient, scalable, and flexible foundation for enterprise-grade observability.

9 Legal notice

Documents published as part of the SUSE Best Practices series 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.

Below we draw your attention to the license under which the articles are published.

10 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.