LOGZILLA DOCUMENTATION

Receiving Syslog Events

Configure LogZilla to receive standard syslog events over UDP and TCP, including listener ports, config.yaml locations, and conf.d customization

Receiving Data in LogZilla · Generated June 11, 2026 · logzilla.ai/docs/receiving-data/receiving-syslog-events

Receiving Syslog Events

LogZilla receives standard syslog events by default and typically requires no changes. To view or change listener ports and runtime options, use the UI: `Settings` → `System Settings` → `Syslog Daemon`. For detailed field descriptions, see Syslog Settings (https://www.logzilla.ai/docs/administration/syslog-settings).

When advanced customization is required (custom sources, `conf.d/`, or pipeline rules), see Syslog pipeline customization (https://www.logzilla.ai/docs/administration/syslog-pipeline-customization). Apply changes cautiously because they can affect event ingest and performance.

Configuration is managed primarily via the UI. For background material and defaults, see Network Communications (https://www.logzilla.ai/docs/administration/network-communications) and Syslog Basics (https://www.logzilla.ai/docs/administration/syslog-basics).

Configuration locations

- `/etc/logzilla/syslog-ng/config.yaml`: main YAML used to render the syslog-ng configuration inside the container.
- `/etc/logzilla/syslog-ng/conf.d/`: directory for additional `*.conf` files included by the main template. The path is controlled by the `custom_conf_dir` key in `config.yaml`.

Avoid creating custom top-level `log` statements. Use `extra_log_rules` to insert filters and rewrites into the main pipeline.

When to customize

- Add a dedicated listener for a specific source or transport.
- Tag a source using `source_type` for dedicated rule processing.
- Forward or archive events using the Forwarder module (do not configure forwarding in syslog-ng). See Downstream Syslog Receivers (https://www.logzilla.ai/docs/forwarding-module/downstream-syslog-receivers).

Most values in `config.yaml` are generated from LogZilla settings. To inject lightweight filters or rewrites into the main pipeline, use the `extra_log_rules` string.

Important

Do not configure destinations in syslog-ng for forwarding or archival. Use the Forwarder module so downstream systems receive parsed and enriched data. See Downstream Syslog Receivers (https://www.logzilla.ai/docs/forwarding-module/downstream-syslog-receivers).

Forwarding and destinations

Forwarding or archival should be configured in the Forwarder module, not in syslog-ng. For raw troubleshooting captures, use the procedures in Syslog Troubleshooting (https://www.logzilla.ai/docs/administration/syslog-troubleshooting). For forwarding options (including file outputs), see the Forwarder module: Downstream Syslog Receivers (https://www.logzilla.ai/docs/forwarding-module/downstream-syslog-receivers).

Sources

Custom sources can be defined with dedicated ports. A `source_type` can be used to tag events from a source for specialized parsing.

Standard sources provided by default configuration (do not change unless necessary):

- `bsd` - TCP on port 514 (or the value set by SYSLOG_BSD_TCP_PORT), for BSD-style syslog messages
- `bsd_udp` - UDP on port 514 (or the value set by SYSLOG_BSD_UDP_PORT), for BSD-style syslog messages using UDP
- `rfc5424` - TCP on port 601 (or the value set by SYSLOG_RFC5424_PORT), for RFC 5424 style syslog messages
- `json` - TCP on port 515 (or the value set by SYSLOG_JSON_PORT), for sending raw JSON messages (newline separated) over a TCP connection
- `tls` - TCP on port 6514 (or the value set by SYSLOG_TLS_PORT). TLS-encrypted RFC 5424 reception
- `raw` - TCP on port 516 (or the value set by SYSLOG_RAW_PORT), for sources not complying with the syslog standard; no parsing is performed and the raw message is sent to LogZilla as is
- `raw_udp` - UDP on port 516 (or the value set by SYSLOG_RAW_UDP_PORT), same as raw-without parsing, the message is sent to LogZilla as is

To add a custom source, define an entry in the `sources` array with:

- `name`: unique source name.
- `enabled`: boolean toggle.
- `type`: `network` or `syslog`.
- `port`: listener port.
- `transport`: `tcp`, `udp`, or `tls` (for TLS-encrypted TCP).
- `tls_cert_file` / `tls_key_file`: paths to TLS certificate and key when `transport` is `tls`.
- `flags`: list of syslog-ng flags.
- `program_override`: override the `program` field value.
- `extra_fields`: key-value map added to the event `extra_fields`.
- `source_type`: tag string added to events from this source (in `extra_fields._source_type`) for dedicated parsing workflows.

Dedicated sources (`source_type`)

Dedicated parsing can be enabled by tagging events from a specific source and loading rules that target that tag:

Set `source_type` on the syslog-ng source in `config.yaml`.
In the relevant Lua rule, set `SOURCE_FILTER = "<tag>"`.
Only events with the matching `source_type` are processed by rules that declare the corresponding `SOURCE_FILTER`.

Examples: config.yaml customization

The following examples show minimal, safe changes to `config.yaml`. Add new entries to the existing `sources` list.

Example 1: Add a TLS source with a dedicated tag
Adds a TLS listener on port 6514 with certificate files and a `source_type` for dedicated rule routing. Optionally sets flags and a program name.

```
sources:
  - name: tls_west
    enabled: true
    type: network
    port: 6514
    transport: tls
    tls_cert_file: /etc/logzilla/server.crt
    tls_key_file: /etc/logzilla/server.key
    flags: ["syslog-protocol"]
    program_override: "tls-wf"
    extra_fields:
      site: "west-dc"
    source_type: "west"
```

Example 2: Add a raw UDP source for unparsed logs
Adds a UDP listener on port 1516 that bypasses syslog parsing and tags events for dedicated handling.

```
sources:
  - name: raw_udp_1516
    enabled: true
    type: network
    port: 1516
    transport: udp
    flags: ["no-parse"]
    program_override: "raw-udp"
    extra_fields:
      log_type: "raw"
    source_type: "devices"
```

After editing `config.yaml`, restart the module as shown below.

Adding extra files in `/etc/logzilla/syslog-ng/conf.d` directory

For more complex cases, additional `*.conf` files can be added in this directory, and they will be included in the main config. This can be used to add syslog-ng sources, filters, or rewrite rules.

Important

Do not create destinations here for forwarding or archival. Use the Forwarder module instead. See Downstream Syslog Receivers (https://www.logzilla.ai/docs/forwarding-module/downstream-syslog-receivers). Create sources as in `config.yaml` as described above.

To accomplish this:

Create a `xxx.conf` file (where `xxx` is the desired name) in the `/etc/logzilla/syslog-ng/conf.d` directory. (More than one of these files can be created, as desired, and they can all take effect.)
Add configuration directives appropriate for a source, filter, or rewrite rule to the new `xxx.conf` file. These should follow standard syslog-ng syntax (see the syslog-ng Open Source Edition Administration Guide (https://syslog-ng.github.io/admin-guide/README)).
Important: Custom `log` entries should **not** be created or configured. It is required that the `log` section be modified only by LogZilla, or LogZilla may cease receiving events.
If `log` customization is desired, such as adding new *filters* or *rewrites*, then see below for detailed instructions.

For many cases, adding a file in `conf.d` is enough. Sources and destinations defined in these files are implicitly added to the main config. Restart the module after changes.

For some advanced cases, like when you want to add some extra filters, then `/etc/logzilla/syslog-ng/config.yaml` should be modified. In particular, if extra *syslog-ng* configuration directives are needed, they should be added to the `extra_log_rules` entry in this file.

Example: apply a filter via `conf.d` and `extra_log_rules`

This example filters events from a specific host using a small `conf.d` file and the `extra_log_rules` hook. This avoids custom top-level `log` statements and keeps the main pipeline intact.

Create `/etc/logzilla/syslog-ng/conf.d/select_host.conf` with:

```
filter f_only_host {
    host("1.2.3.4");
};
```

Edit `/etc/logzilla/syslog-ng/config.yaml` and set:

```
extra_log_rules: "filter(f_only_host);"
```

The filter is injected into the main pipeline. Any file destinations defined in `config.yaml` or `conf.d` are included automatically. For built-in troubleshooting options, see the Debugging page below.

Restarting syslog-ng after changes

After changes to the syslog-ng configuration, restart the module:

```
logzilla restart -c syslog
```

Verification

For verification and troubleshooting steps, see Syslog Troubleshooting (https://www.logzilla.ai/docs/administration/syslog-troubleshooting).