

Changelog

Please note that this is not a direct replacement of Grafana’s “add to changelog” label. It is *mostly* for internal consumption of the Alerting team that owns this part of Grafana.

Scope Glossary

[ADMIN]

The ADMIN scope denotes a change that affect the structure and layout of this repository. This includes updates to the following:

- CODEOWNERS
- README
- DotFiles (.gitignore, .git-attributes, etc)

Anything that a developer working on this repo should be aware of from a standards and practice perspective.

[BUGFIX]

The BUGFIX scope denotes a change that fixes an issue with the project in question. A BUGFIX should align the behaviour of the service with the current expected behaviour of the service. If a BUGFIX introduces new unexpected behaviour to ameliorate the issue, a corresponding FEATURE or ENHANCEMENT scope should also be added to the changelog.

[CHANGE]

The CHANGE scope denotes a change that changes the expected behavior of the project while not adding new functionality or fixing an underling issue. This commonly occurs when renaming things to make them more consistent or to accommodate updated versions of vendored dependencies.

[FEATURE]

The FEATURE scope denotes a change that adds new functionality to the project/service.

[ENHANCEMENT]

The ENHANCEMENT scope denotes a change that improves upon the current functionality of the project/service. Generally, an enhancement is something that improves upon something that is already present. Either by making it simpler, more powerful, or more performant. For Example:

An optimization on a particular process in a service that makes it more performant
Simpler syntax for setting a configuration value, like allowing 1m instead of 60 for a duration setting.

Order

Scopes must have an order to ensure consistency and ease of search, this helps us identify which section do we need to look for what. The order must be:

1. [CHANGE]
2. [FEATURE]
3. [BUGFIX]
4. [ENHANCEMENT]
5. [ADMIN]

Grafana Alerting - main / unreleased

- [CHANGE] Prometheus Compatible API: Use float-like values for `api/prometheus/grafana/api/v1/alerts` and `api/prometheus/grafana/api/v1/rules` instead of the evaluation string #47216
- [ENHANCEMENT] Templates: Enable the use of classic condition values in templates #46971