+++ title = “Configuration” description = “Configuration documentation” keywords = [“grafana”, “configuration”, “documentation”] aliases = [“/docs/grafana/latest/installation/configuration/”] weight = 150 +++

# Configuration

Grafana has default and custom configuration files. You can customize your Grafana instance by modifying the custom configuration file or by using environment variables. To see the list of settings for a Grafana instance, refer to [View server settings]({{< relref “view-server/view-server-settings.md” >}}).

> **Note:** After you add custom options, uncomment the relevant sections of the configuration file. Restart Grafana for your changes to take effect.

## Configuration file location

The default settings for a Grafana instance are stored in the `$WORKING_DIR/conf/defaults.ini` file. *Do not* change this file.

Depending on your OS, your custom configuration file is either the `$WORKING_DIR/conf/defaults.ini` file or the `/usr/local/etc/grafana/grafana.ini` file. The custom configuration file path can be overridden using the `--config` parameter.

### Linux

If you installed Grafana using the `deb` or `rpm` packages, then your configuration file is located at `/etc/grafana/grafana.ini` and a separate `custom.ini` is not used. This path is specified in the Grafana init.d script using `--config` file parameter.

### Docker

Refer to [Configure a Grafana Docker image]({{< relref “configure-docker.md” >}}) for information about environmental variables, persistent storage, and building custom Docker images.

### Windows

On Windows, the `sample.ini` file is located in the same directory as `defaults.ini` file. It contains all the settings commented out. Copy `sample.ini` and name it `custom.ini`.

### macOS

By default, the configuration file is located at `/usr/local/etc/grafana/grafana.ini`. For a Grafana instance installed using Homebrew, edit the `grafana.ini` file directly. Otherwise, add a configuration file named `custom.ini` to the `conf` folder to override the settings defined in `conf/defaults.ini`.

## Remove comments in the .ini files

Grafana uses semicolons (the `;` char) to comment out lines in a `.ini` file. You must uncomment each line in the `custom.ini` or the `grafana.ini` file that you are modify by removing `;` from the beginning of that line. Otherwise your changes will be ignored.

For example:

```
# The HTTP port  to use
;http_port = 3000
```

## Override configuration with environment variables

Do not use environment variables to *add* new configuration settings. Instead, use environmental variables to *override* existing options.

To override an option:

```
GF_<SectionName>_<KeyName>
```

Where the section name is the text within the brackets. Everything should be uppercase, `.` and `-` should be replaced by `_`. For example, if you have these configuration settings:

```
# default section
instance_name = ${HOSTNAME}

[security]
admin_user = admin

[auth.google]
client_secret = 0ldS3cretKey

[plugin.grafana-image-renderer]
rendering_ignore_https_errors = true
```

You can override them on Linux machines with:

```
export GF_DEFAULT_INSTANCE_NAME=my-instance
export GF_SECURITY_ADMIN_USER=owner
export GF_AUTH_GOOGLE_CLIENT_SECRET=newS3cretKey
export GF_PLUGIN_GRAFANA_IMAGE_RENDERER_RENDERING_IGNORE_HTTPS_ERRORS=true
```

## Variable expansion

> **Note:** Only available in Grafana 7.1+.

If any of your options contains the expression `$__<provider>{<argument>}` or `${<environment variable>}`, then they will be processed by Grafana's variable expander. The expander runs the provider with the provided argument to get the final value of the option.

There are three providers: `env`, `file`, and `vault`.

### Env provider

The `env` provider can be used to expand an environment variable. If you set an option to `$__env{PORT}` the `PORT` environment variable will be used in its place. For environment variables you can also use the short-hand syntax `${PORT}`. Grafana's log directory would be set to the `grafana` directory in the directory behind the `LOGDIR` environment variable in the following example.

```
[paths]
logs = $__env{LOGDIR}/grafana
```

### File provider

`file` reads a file from the filesystem. It trims whitespace from the beginning and the end of files. The database password in the following example would be replaced by the content of the `/etc/secrets/gf_sql_password` file:

```
[database]
password = $__file{/etc/secrets/gf_sql_password}
```

### Vault provider

The `vault` provider allows you to manage your secrets with Hashicorp Vault.

> Vault provider is only available in Grafana Enterprise v7.1+. For more information, refer to [Vault integration]({{< relref "../enterprise/vault.md" >}}) in [Grafana Enterprise]({{< relref "../enterprise" >}}).

## app_mode

Options are `production` and `development`. Default is `production`. *Do not* change this option unless you are working on Grafana development.

## instance_name

Set the name of the grafana-server instance. Used in logging, internal metrics, and clustering info. Defaults to: `${HOSTNAME}`, which will be replaced with environment variable `HOSTNAME`, if that is empty or does not exist Grafana will try to use system calls to get the machine name.

## [paths]

### data

Path to where Grafana stores the sqlite3 database (if used), file-based sessions (if used), and other data. This path is usually specified via command line in the init.d script or the systemd service file.

**macOS:** The default SQLite database is located at `/usr/local/var/lib/grafana`

### temp_data_lifetime

How long temporary images in `data` directory should be kept. Defaults to: `24h`. Supported modifiers: `h` (hours), `m` (minutes), for example: `168h`, `30m`, `10h30m`. Use `0` to never clean up temporary files.

### logs

Path to where Grafana stores logs. This path is usually specified via command line in the init.d script or the systemd service file. You can override it in the configuration file or in the default environment variable file. However, please note that by overriding this the default log path will be used temporarily until Grafana has fully initialized/started.

Override log path using the command line argument `cfg:default.paths.logs`:

```
./grafana-server --config /custom/config.ini --homepath /custom/homepath cfg:default.paths.l
```

**macOS:** By default, the log file should be located at `/usr/local/var/log/grafana/grafana.log`.

### plugins

Directory where Grafana automatically scans and looks for plugins. For information about manually or automatically installing plugins, refer to [Install Grafana plugins]({{< relref "../plugins/installation.md" >}}).

**macOS:** By default, the Mac plugin location is: `/usr/local/var/lib/grafana/plugins`.

### provisioning

Folder that contains provisioning({{< relref "provisioning.md" >}}) config files that Grafana will apply on startup. Dashboards will be reloaded when the json files changes.

## [server]

### protocol

`http`,`https`,`h2` or `socket`

### http_addr

The IP address to bind to. If empty will bind to all interfaces

### http_port

The port to bind to, defaults to `3000`. To use port 80 you need to either give the Grafana binary permission for example:

```
$ sudo setcap 'cap_net_bind_service=+ep' /usr/sbin/grafana-server
```

Or redirect port 80 to the Grafana port using:

```
$ sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3000
```

Another way is to put a web server like Nginx or Apache in front of Grafana and have them proxy requests to Grafana.

### domain

This setting is only used in as a part of the `root_url` setting (see below). Important if you use GitHub or Google OAuth.

### enforce_domain

Redirect to correct domain if the host header does not match the domain. Prevents DNS rebinding attacks. Default is `false`.

### root_url

This is the full URL used to access Grafana from a web browser. This is important if you use Google or GitHub OAuth authentication (for the callback URL to be correct).

> **Note:** This setting is also important if you have a reverse proxy in front of Grafana that exposes it through a subpath. In that case add the subpath to the end of this URL setting.

### serve_from_sub_path

Serve Grafana from subpath specified in `root_url` setting. By default it is set to `false` for compatibility reasons.

By enabling this setting and using a subpath in `root_url` above, e.g. `root_url = http://localhost:3000/grafana`, Grafana is accessible on `http://localhost:3000/grafana`.

### router_logging

Set to `true` for Grafana to log all HTTP requests (not just errors). These are logged as Info level events to the Grafana log.

### static_root_path

The path to the directory where the front end files (HTML, JS, and CSS files). Defaults to `public` which is why the Grafana binary needs to be executed with working directory set to the installation path.

### enable_gzip

Set this option to `true` to enable HTTP compression, this can improve transfer speed and bandwidth utilization. It is recommended that most users set it to `true`. By default it is set to `false` for compatibility reasons.

### cert_file

Path to the certificate file (if `protocol` is set to `https` or `h2`).

### cert_key

Path to the certificate key file (if `protocol` is set to `https` or `h2`).

### socket

Path where the socket should be created when `protocol=socket`. Make sure that Grafana has appropriate permissions before you change this setting.

### cdn_url

> **Note**: Available in Grafana v7.4 and later versions.

Specify a full HTTP URL address to the root of your Grafana CDN assets. Grafana will add edition and version paths.

For example, given a cdn url like `https://cdn.myserver.com` grafana will try to load a javascript file from `http://cdn.myserver.com/grafana-oss/7.4.0/public/build/app.<hash>.js`.

### read_timeout

Sets the maximum time using a duration format (5s/5m/5ms) before timing out read of an incoming request and closing idle connections. `0` means there is no timeout for reading the request.

## database

Grafana needs a database to store users and dashboards (and other things). By default it is configured to use `sqlite3` which is an embedded database (included in the main Grafana binary).

### type

Either `mysql`, `postgres` or `sqlite3`, it's your choice.

### host

Only applicable to MySQL or Postgres. Includes IP or hostname and port or in case of Unix sockets the path to it. For example, for MySQL running on the same host as Grafana: `host = 127.0.0.1:3306` or with Unix sockets: `host = /var/run/mysqld/mysqld.sock`

### name

The name of the Grafana database. Leave it set to `grafana` or some other name.

### user

The database user (not applicable for `sqlite3`).

### password

The database user's password (not applicable for `sqlite3`). If the password contains `#` or `;` you have to wrap it with triple quotes. For example `"""#password;"""`

### url

Use either URL or the other fields below to configure the database Example: `mysql://user:secret@host:port/database`

### max_idle_conn

The maximum number of connections in the idle connection pool.

### max_open_conn

The maximum number of open connections to the database.

### conn_max_lifetime

Sets the maximum amount of time a connection may be reused. The default is 14400 (which means 14400 seconds or 4 hours). For MySQL, this setting should be shorter than the `wait_timeout` variable.

### locking_attempt_timeout_sec

For "mysql", if `lockingMigration` feature toggle is set, specify the time (in seconds) to wait before failing to lock the database for the migrations. Default is 0.

### log_queries

Set to `true` to log the sql calls and execution times.

### ssl_mode

For Postgres, use either `disable`, `require` or `verify-full`. For MySQL, use either `true`, `false`, or `skip-verify`.

### isolation_level

Only the MySQL driver supports isolation levels in Grafana. In case the value is empty, the driver's default isolation level is applied. Available options are "READ-UNCOMMITTED", "READ-COMMITTED", "REPEATABLE-READ" or "SERIALIZABLE".

### ca_cert_path

The path to the CA certificate to use. On many Linux systems, certs can be found in `/etc/ssl/certs`.

### client_key_path

The path to the client key. Only if server requires client authentication.

### client_cert_path

The path to the client cert. Only if server requires client authentication.

### server_cert_name

The common name field of the certificate used by the `mysql` or `postgres` server. Not necessary if `ssl_mode` is set to `skip-verify`.

### path

Only applicable for `sqlite3` database. The file path where the database will be stored.

### cache_mode

For "sqlite3" only. Shared cache setting used for connecting to the database. (private, shared) Defaults to `private`.

## [remote_cache]

Caches authentication details and session information in the configured database, Redis or Memcached. This setting does not configure [Query Caching in Grafana Enterprise]({{< relref "../enterprise/query-caching.md" >}}).

### type

Either `redis`, `memcached`, or `database`. Defaults to `database`

#### connstr

The remote cache connection string. The format depends on the `type` of the remote cache. Options are `database`, `redis`, and `memcache`.

**database** Leave empty when using `database` since it will use the primary database.

**redis** Example connstr: `addr=127.0.0.1:6379,pool_size=100,db=0,ssl=false`

- `addr` is the host `:` port of the redis server.
- `pool_size` (optional) is the number of underlying connections that can be made to redis.
- `db` (optional) is the number identifier of the redis database you want to use.
- `ssl` (optional) is if SSL should be used to connect to redis server. The value may be `true`, `false`, or `insecure`. Setting the value to `insecure` skips verification of the certificate chain and hostname when making the connection.

**memcache** Example connstr: `127.0.0.1:11211`

### [dataproxy]

#### logging

This enables data proxy logging, default is `false`.

#### timeout

How long the data proxy should wait before timing out. Default is 30 seconds.

This setting also applies to core backend HTTP data sources where query requests use an HTTP client with timeout set.

#### keep_alive_seconds

Interval between keep-alive probes. Default is `30` seconds. For more details check the Dialer.KeepAlive documentation.

#### tls_handshake_timeout_seconds

The length of time that Grafana will wait for a successful TLS handshake with the datasource. Default is `10` seconds. For more details check the Transport.TLSHandshakeTimeout documentation.

#### expect_continue_timeout_seconds

The length of time that Grafana will wait for a datasource's first response headers after fully writing the request headers, if the request has an "Expect: 100-continue" header. A value of `0` will result in the body being sent immediately. Default is `1` second. For more details check the Transport.ExpectContinueTimeout documentation.

#### max_conns_per_host

Optionally limits the total number of connections per host, including connections in the dialing, active, and idle states. On limit violation, dials are blocked. A value of `0` means that there are no limits. Default is `0`. For more details check the Transport.MaxConnsPerHost documentation.

#### max_idle_connections

The maximum number of idle connections that Grafana will maintain. Default is `100`. For more details check the Transport.MaxIdleConns documentation.

#### max_idle_connections_per_host

[Deprecated - use max_idle_connections instead]

The maximum number of idle connections per host that Grafana will maintain. Default is `2`. For more details check the Transport.MaxIdleConnsPerHost documentation.

#### idle_conn_timeout_seconds

The length of time that Grafana maintains idle connections before closing them. Default is `90` seconds. For more details check the Transport.IdleConnTimeout documentation.

#### send_user_header

If enabled and user is not anonymous, data proxy will add X-Grafana-User header with username into the request. Default is `false`.

#### response_limit

Limits the amount of bytes that will be read/accepted from responses of outgoing HTTP requests. Default is `0` which means disabled.

#### row_limit

Limits the number of rows that Grafana will process from SQL (relational) data sources. Default is `1000000`.

## [analytics]

### reporting_enabled

When enabled Grafana will send anonymous usage statistics to `stats.grafana.org`. No IP addresses are being tracked, only simple counters to track running instances, versions, dashboard and error counts. It is very helpful to us, so please leave this enabled. Counters are sent every 24 hours. Default value is `true`.

### check_for_updates

Set to false, disables checking for new versions of Grafana from Grafana's GitHub repository. When enabled, the check for a new version runs every 10 minutes. It will notify, via the UI, when a new version is available. The check itself will not prompt any auto-updates of the Grafana software, nor will it send any sensitive information.

### check_for_plugin_updates

> **Note**: Available in Grafana v8.5.0 and later versions.

Set to false disables checking for new versions of installed plugins from https://grafana.com. When enabled, the check for a new plugin runs every 10 minutes. It will notify, via the UI, when a new plugin update exists. The check itself will not prompt any auto-updates of the plugin, nor will it send any sensitive information.

### google_analytics_ua_id

If you want to track Grafana usage via Google analytics specify *your* Universal Analytics ID here. By default this feature is disabled.

### google_tag_manager_id

Google Tag Manager ID, only enabled if you enter an ID here.

### rudderstack_write_key

If you want to track Grafana usage via Rudderstack specify *your* Rudderstack Write Key here. The `rudderstack_data_plane_url` must also be provided for this feature to be enabled. By default this feature is disabled.

### rudderstack_data_plane_url

Rudderstack data plane url that will receive Rudderstack events. The `rudderstack_write_key` must also be provided for this feature to be enabled.

### rudderstack_sdk_url

Optional. If tracking with Rudderstack is enabled, you can provide a custom URL to load the Rudderstack SDK.

### rudderstack_config_url

Optional. If tracking with Rudderstack is enabled, you can provide a custom URL to load the Rudderstack config.

### application_insights_connection_string

If you want to track Grafana usage via Azure Application Insights, then specify *your* Application Insights connection string. Since the connection string contains semicolons, you need to wrap it in backticks (‘). By default, tracking usage is disabled.

### application_insights_endpoint_url

```
    Optionally, use this option to override the default endpoint address for Application Ins
```

## [security]

### disable_initial_admin_creation

> Only available in Grafana v6.5+.

Disable creation of admin user on first start of Grafana. Default is `false`.

### admin_user

The name of the default Grafana Admin user, who has full permissions. Default is `admin`.

### admin_password

The password of the default Grafana Admin. Set once on first-run. Default is `admin`.

### secret_key

Used for signing some data source settings like secrets and passwords, the encryption format used is AES-256 in CFB mode. Cannot be changed without requiring an update to data source settings to re-encode them.

### disable_gravatar

Set to `true` to disable the use of Gravatar for user profile images. Default is `false`.

### data_source_proxy_whitelist

Define a whitelist of allowed IP addresses or domains, with ports, to be used in data source URLs with the Grafana data source proxy. Format: `ip_or_domain:port` separated by spaces. PostgreSQL, MySQL, and MSSQL data sources do not use the proxy and are therefore unaffected by this setting.

### disable_brute_force_login_protection

Set to `true` to disable brute force login protection. Default is `false`.

### cookie_secure

Set to `true` if you host Grafana behind HTTPS. Default is `false`.

### cookie_samesite

Sets the `SameSite` cookie attribute and prevents the browser from sending this cookie along with cross-site requests. The main goal is to mitigate the risk of cross-origin information leakage. This setting also provides some protection against cross-site request forgery attacks (CSRF), read more about SameSite here. Valid values are `lax`, `strict`, `none`, and `disabled`. Default is `lax`. Using value `disabled` does not add any `SameSite` attribute to cookies.

### allow_embedding

When `false`, the HTTP header `X-Frame-Options: deny` will be set in Grafana HTTP responses which will instruct browsers to not allow rendering Grafana in a `<frame>`, `<iframe>`, `<embed>` or `<object>`. The main goal is to mitigate the risk of Clickjacking. Default is `false`.

### strict_transport_security

Set to `true` if you want to enable HTTP `Strict-Transport-Security` (HSTS) response header. Only use this when HTTPS is enabled in your configuration, or when there is another upstream system that ensures your application does HTTPS (like a frontend load balancer). HSTS tells browsers that the site should only be accessed using HTTPS.

### strict_transport_security_max_age_seconds

Sets how long a browser should cache HSTS in seconds. Only applied if strict_transport_security is enabled. The default value is `86400`.

### strict_transport_security_preload

Set to `true` to enable HSTS `preloading` option. Only applied if strict_transport_security is enabled. The default value is `false`.

### strict_transport_security_subdomains

Set to `true` if to enable the HSTS includeSubDomains option. Only applied if strict_transport_security is enabled. The default value is `false`.

### x_content_type_options

Set to `true` to enable the X-Content-Type-Options response header. The X-Content-Type-Options response HTTP header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should not be changed and be followed. The default value is `false`.

### x_xss_protection

Set to `false` to disable the X-XSS-Protection header, which tells browsers to stop pages from loading when they detect reflected cross-site scripting (XSS) attacks. The default value is `false` until the next minor release, `6.3`.

### content_security_policy

Set to `true` to add the Content-Security-Policy header to your requests. CSP allows to control resources that the user agent can load and helps prevent XSS attacks.

### content_security_policy_template

Set Content Security Policy template used when adding the Content-Security-Policy header to your requests. `$NONCE` in the template includes a random nonce.

### angular_support_enabled

This currently defaults to `true` but will in Grafana v9 default to `false`. When set to false the angular framework and support components will not be loaded. This means that all plugins and core features that depend on angular support will stop working.

Current core features that will stop working:

- Heatmap panel
- Old graph panel
- Old table panel
- Postgres, MySQL and MSSQL data source query editors
- Legacy alerting edit rule UI

Before we disable angular support by default we plan to migrate these remaining areas to React.

## [snapshots]

### external_enabled

Set to `false` to disable external snapshot publish endpoint (default `true`).

### external_snapshot_url

Set root URL to a Grafana instance where you want to publish external snapshots (defaults to https://snapshots.raintank.io).

### external_snapshot_name

Set name for external snapshot button. Defaults to `Publish to snapshots.raintank.io`.

### public_mode

Set to true to enable this Grafana instance to act as an external snapshot server and allow unauthenticated requests for creating and deleting snapshots. Default is `false`.

### snapshot_remove_expired

Enable this to automatically remove expired snapshots. Default is `true`.

## [dashboards]

### versions_to_keep

Number dashboard versions to keep (per dashboard). Default: `20`, Minimum: `1`.

### min_refresh_interval

> Only available in Grafana v6.7+.

This feature prevents users from setting the dashboard refresh interval to a lower value than a given interval value. The default interval value is 5 seconds. The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. `30s` or `1m`.

As of Grafana v7.3, this also limits the refresh interval options in Explore.

### default_home_dashboard_path

Path to the default home dashboard. If this value is empty, then Grafana uses StaticRootPath + "dashboards/home.json".

> **Note:** On Linux, Grafana uses `/usr/share/grafana/public/dashboards/home.json` as the default home dashboard location.

## [users]

### allow_sign_up

Set to `false` to prohibit users from being able to sign up / create user accounts. Default is `false`. The admin user can still create users. For more information about creating a user, refer to [Add a user]({{< relref "../administration/manage-users-and-permissions/manage-server-users/add-user.md" >}}).

### allow_org_create

Set to `false` to prohibit users from creating new organizations. Default is `false`.

### auto_assign_org

Set to `true` to automatically add new users to the main organization (id 1). When set to `false`, new users automatically cause a new organization to be created for that new user. Default is `true`.

### auto_assign_org_id

Set this value to automatically add new users to the provided org. This requires `auto_assign_org` to be set to `true`. Please make sure that this organization already exists. Default is 1.

### auto_assign_org_role

The role new users will be assigned for the main organization (if the above setting is set to true). Defaults to `Viewer`, other valid options are `Admin` and `Editor`. e.g.:

```
auto_assign_org_role = Viewer
```

### verify_email_enabled

Require email validation before sign up completes. Default is `false`.

### login_hint

Text used as placeholder text on login page for login/username input.

### password_hint

Text used as placeholder text on login page for password input.

### default_theme

Set the default UI theme: `dark` or `light`. Default is `dark`.

#### home_page

Path to a custom home page. Users are only redirected to this if the default home dashboard is used. It should match a frontend route and contain a leading slash.

#### External user management

If you manage users externally you can replace the user invite button for organizations with a link to an external site together with a description.

#### viewers_can_edit

Viewers can access and use [Explore]({{< relref "../explore/_index.md" >}}) and perform temporary edits on panels in dashboards they have access to. They cannot save their changes. Default is `false`.

#### editors_can_admin

Editors can administrate dashboards, folders and teams they create. Default is `false`.

#### user_invite_max_lifetime_duration

The duration in time a user invitation remains valid before expiring. This setting should be expressed as a duration. Examples: 6h (hours), 2d (days), 1w (week). Default is `24h` (24 hours). The minimum supported duration is `15m` (15 minutes).

#### hidden_users

This is a comma-separated list of usernames. Users specified here are hidden in the Grafana UI. They are still visible to Grafana administrators and to themselves.

### [auth]

Grafana provides many ways to authenticate users. Refer to the Grafana [Authentication overview]({{< relref "../auth/overview.md" >}}) and other authentication documentation for detailed instructions on how to set up and configure authentication.

#### login_cookie_name

The cookie name for storing the auth token. Default is `grafana_session`.

### login_maximum_inactive_lifetime_duration

The maximum lifetime (duration) an authenticated user can be inactive before being required to login at next visit. Default is 7 days (7d). This setting should be expressed as a duration, e.g. 5m (minutes), 6h (hours), 10d (days), 2w (weeks), 1M (month). The lifetime resets at each successful token rotation (token_rotation_interval_minutes).

### login_maximum_lifetime_duration

The maximum lifetime (duration) an authenticated user can be logged in since login time before being required to login. Default is 30 days (30d). This setting should be expressed as a duration, e.g. 5m (minutes), 6h (hours), 10d (days), 2w (weeks), 1M (month).

### token_rotation_interval_minutes

How often auth tokens are rotated for authenticated users when the user is active. The default is each 10 minutes.

### disable_login_form

Set to true to disable (hide) the login form, useful if you use OAuth. Default is false.

### disable_signout_menu

Set to `true` to disable the signout link in the side menu. This is useful if you use auth.proxy. Default is `false`.

### signout_redirect_url

URL to redirect the user to after they sign out.

### oauth_auto_login

Set to `true` to attempt login with OAuth automatically, skipping the login screen. This setting is ignored if multiple OAuth providers are configured. Default is `false`.

### oauth_state_cookie_max_age

How many seconds the OAuth state cookie lives before being deleted. Default is `600` (seconds) Administrators can increase this if they experience OAuth login state mismatch errors.

### oauth_skip_org_role_update_sync

Skip forced assignment of OrgID `1` or `auto_assign_org_id` for external logins. Default is `false`. Use this setting to distribute users with external login to multiple organizations. Otherwise, the users' organization would get reset on every new login, for example, via AzureAD.

### api_key_max_seconds_to_live

Limit of API key seconds to live before expiration. Default is -1 (unlimited).

### sigv4_auth_enabled

> Only available in Grafana 7.3+.

Set to `true` to enable the AWS Signature Version 4 Authentication option for HTTP-based datasources. Default is `false`.

### sigv4_verbose_logging

> Only available in Grafana 8.4+.

Set to `true` to enable verbose request signature logging when AWS Signature Version 4 Authentication is enabled. Default is `false`.

## [auth.anonymous]

Refer to [Anonymous authentication]({{< relref "../auth/grafana.md/#anonymous-authentication" >}}) for detailed instructions.

## [auth.github]

Refer to [GitHub OAuth2 authentication]({{< relref "../auth/github.md" >}}) for detailed instructions.

## [auth.gitlab]

Refer to [Gitlab OAuth2 authentication]({{< relref "../auth/gitlab.md" >}}) for detailed instructions.

## [auth.google]

Refer to [Google OAuth2 authentication]({{< relref "../auth/google.md" >}}) for detailed instructions.

## [auth.grafananet]

Legacy key names, still in the config file so they work in env variables.

## [auth.grafana_com]

Legacy key names, still in the config file so they work in env variables.

## [auth.azuread]

Refer to [Azure AD OAuth2 authentication]({{< relref "../auth/azuread.md" >}}) for detailed instructions.

## [auth.okta]

Refer to [Okta OAuth2 authentication]({{< relref "../auth/okta.md" >}}) for detailed instructions.

## [auth.generic_oauth]

Refer to [Generic OAuth authentication]({{< relref "../auth/generic-oauth.md" >}}) for detailed instructions.

## [auth.basic]

Refer to [Basic authentication]({{< relref "../auth/overview.md#basic-authentication" >}}) for detailed instructions.

## [auth.proxy]

Refer to [Auth proxy authentication]({{< relref "../auth/auth-proxy.md" >}}) for detailed instructions.

## [auth.ldap]

Refer to [LDAP authentication]({{< relref "../auth/ldap.md" >}}) for detailed instructions.

## [aws]

You can configure core and external AWS plugins.

### allowed_auth_providers

Specify what authentication providers the AWS plugins allow. For a list of allowed providers, refer to the data-source configuration page for a given plugin. If you configure a plugin by provisioning, only providers that are specified in `allowed_auth_providers` are allowed.

Options: `default` (AWS SDK default), `keys` (Access and secret key), `credentials` (Credentials file), `ec2_iam_role` (EC2 IAM role)

### assume_role_enabled

Set to `false` to disable AWS authentication from using an assumed role with temporary security credentials. For details about assume roles, refer to the AWS API reference documentation about the AssumeRole operation.

If this option is disabled, the **Assume Role** and the **External Id** field are removed from the AWS data source configuration page. If the plugin is configured using provisioning, it is possible to use an assumed role as long as `assume_role_enabled` is set to `true`.

### list_metrics_page_limit

Use the List Metrics API option to load metrics for custom namespaces in the CloudWatch data source. By default, the page limit is 500.

## [azure]

Grafana supports additional integration with Azure services when hosted in the Azure Cloud.

### cloud

Azure cloud environment where Grafana is hosted:

| Azure Cloud | Value |
|---|---|
| Microsoft Azure public cloud | AzureCloud (*default*) |
| Microsoft Chinese national cloud | AzureChinaCloud |
| US Government cloud | AzureUSGovernment |
| Microsoft German national cloud ("Black Forest") | AzureGermanCloud |

### managed_identity_enabled

Specifies whether Grafana hosted in Azure service with Managed Identity configured (e.g. Azure Virtual Machines instance). Disabled by default, needs to be explicitly enabled.

### managed_identity_client_id

The client ID to use for user-assigned managed identity.

Should be set for user-assigned identity and should be empty for system-assigned identity.

## [auth.jwt]

Refer to [JWT authentication]({{< relref "../auth/jwt.md" >}}) for more information.

## [smtp]

Email server settings.

### enabled

Enable this to allow Grafana to send email. Default is `false`.

### host

Default is `localhost:25`.

### user

In case of SMTP auth, default is `empty`.

### password

In case of SMTP auth, default is `empty`. If the password contains `#` or `;`, then you have to wrap it with triple quotes. Example: “““#password;”“”

### cert_file

File path to a cert file, default is `empty`.

### key_file

File path to a key file, default is `empty`.

### skip_verify

Verify SSL for SMTP server, default is `false`.

### from_address

Address used when sending out emails, default is `admin@grafana.localhost`.

### from_name

Name to be used when sending out emails, default is `Grafana`.

### ehlo_identity

Name to be used as client identity for EHLO in SMTP dialog, default is `<instance_name>`.

### startTLS_policy

Either “OpportunisticStartTLS”, “MandatoryStartTLS”, “NoStartTLS”. Default is `empty`.

## [emails]

### welcome_email_on_sign_up

Default is `false`.

### templates_pattern

Enter a comma separated list of template patterns. Default is `emails/*.html, emails/*.txt`.

### content_types

Enter a comma-separated list of content types that should be included in the emails that are sent. List the content types according descending preference, e.g. `text/html, text/plain` for HTML as the most preferred. The order of the parts is significant as the mail clients will use the content type that is supported and most preferred by the sender. Supported content types are `text/html` and `text/plain`. Default is `text/html`.

## [log]

Grafana logging options.

### mode

Options are “console”, “file”, and “syslog”. Default is “console” and “file”. Use spaces to separate multiple modes, e.g. `console file`.

### level

Options are “debug”, “info”, “warn”, “error”, and “critical”. Default is `info`.

### filters

Optional settings to set different levels for specific loggers. For example: `filters = sqlstore:debug`

## [log.console]

Only applicable when “console” is used in `[log]` mode.

### level

Options are “debug”, “info”, “warn”, “error”, and “critical”. Default is inherited from `[log]` level.

### format

Log line format, valid options are text, console and json. Default is `console`.

## [log.file]

Only applicable when "file" used in `[log]` mode.

### level

Options are "debug", "info", "warn", "error", and "critical". Default is inherited from `[log]` level.

### format

Log line format, valid options are text, console and json. Default is `text`.

### log_rotate

Enable automated log rotation, valid options are `false` or `true`. Default is `true`. When enabled use the `max_lines`, `max_size_shift`, `daily_rotate` and `max_days` to configure the behavior of the log rotation.

### max_lines

Maximum lines per file before rotating it. Default is `1000000`.

### max_size_shift

Maximum size of file before rotating it. Default is `28`, which means `1 << 28`, `256MB`.

### daily_rotate

Enable daily rotation of files, valid options are `false` or `true`. Default is `true`.

### max_days

Maximum number of days to keep log files. Default is `7`.

## [log.syslog]

Only applicable when "syslog" used in `[log]` mode.

### level

Options are "debug", "info", "warn", "error", and "critical". Default is inherited from `[log]` level.

#### format

Log line format, valid options are text, console, and json. Default is `text`.

#### network and address

Syslog network type and address. This can be UDP, TCP, or UNIX. If left blank, then the default UNIX endpoints are used.

#### facility

Syslog facility. Valid options are user, daemon or local0 through local7. Default is empty.

#### tag

Syslog tag. By default, the process's `argv[0]` is used.

### [log.frontend]

**Note:** This feature is available in Grafana 7.4+.

#### enabled

Sentry javascript agent is initialized. Default is `false`.

#### sentry_dsn

Sentry DSN if you want to send events to Sentry

#### custom_endpoint

Custom HTTP endpoint to send events captured by the Sentry agent to. Default, `/log`, will log the events to stdout.

#### sample_rate

Rate of events to be reported between `0` (none) and `1` (all, default), float.

#### log_endpoint_requests_per_second_limit

Requests per second limit enforced per an extended period, for Grafana backend log ingestion endpoint, `/log`. Default is `3`.

#### log_endpoint_burst_limit

Maximum requests accepted per short interval of time for Grafana backend log ingestion endpoint, `/log`. Default is `15`.

## [quota]

Set quotas to `-1` to make unlimited.

### enabled

Enable usage quotas. Default is `false`.

### org_user

Limit the number of users allowed per organization. Default is 10.

### org_dashboard

Limit the number of dashboards allowed per organization. Default is 100.

### org_data_source

Limit the number of data sources allowed per organization. Default is 10.

### org_api_key

Limit the number of API keys that can be entered per organization. Default is 10.

### org_alert_rule

Limit the number of alert rules that can be entered per organization. Default is 100.

### user_org

Limit the number of organizations a user can create. Default is 10.

### global_user

Sets a global limit of users. Default is -1 (unlimited).

### global_org

Sets a global limit on the number of organizations that can be created. Default is -1 (unlimited).

### global_dashboard

Sets a global limit on the number of dashboards that can be created. Default is -1 (unlimited).

### global_api_key

Sets global limit of API keys that can be entered. Default is -1 (unlimited).

### global_session

Sets a global limit on number of users that can be logged in at one time. Default is -1 (unlimited).

### global_alert_rule

Sets a global limit on number of alert rules that can be created. Default is -1 (unlimited).

## [unified_alerting]

For more information about the Grafana alerts, refer to [Unified Alerting]({{< relref "../alerting/unified-alerting/_index.md" >}}).

### enabled

Enable the Unified Alerting sub-system and interface. When enabled we'll migrate all of your alert rules and notification channels to the new system. New alert rules will be created and your notification channels will be converted into an Alertmanager configuration. Previous data is preserved to enable backwards compatibility but new data is removed. The default value is `false`.

Alerting Rules migrated from dashboards and panels will include a link back via the `annotations`.

### disabled_orgs

Comma-separated list of organization IDs for which to disable Grafana 8 Unified Alerting.

### admin_config_poll_interval

Specify the frequency of polling for admin config changes. The default value is `60s`.

The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

### alertmanager_config_poll_interval

Specify the frequency of polling for Alertmanager config changes. The default value is `60s`.

The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

### ha_listen_address

Listen address/hostname and port to receive unified alerting messages for other Grafana instances. The port is used for both TCP and UDP. It is assumed other Grafana instances are also running on the same port. The default value is `0.0.0.0:9094`.

### ha_advertise_address

Explicit address/hostname and port to advertise other Grafana instances. The port is used for both TCP and UDP.

### ha_peers

Comma-separated list of initial instances (in a format of host:port) that will form the HA cluster. Configuring this setting will enable High Availability mode for alerting.

### ha_peer_timeout

Time to wait for an instance to send a notification via the Alertmanager. In HA, each Grafana instance will be assigned a position (e.g. 0, 1). We then multiply this position with the timeout to indicate how long should each instance wait before sending the notification to take into account replication lag. The default value is `15s`.

The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

### ha_gossip_interval

The interval between sending gossip messages. By lowering this value (more frequent) gossip messages are propagated across cluster more quickly at the expense of increased bandwidth usage. The default value is `200ms`.

The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

### ha_push_pull_interval

The interval between gossip full state syncs. Setting this interval lower (more frequent) will increase convergence speeds across larger clusters at the expense of increased bandwidth usage. The default value is `60s`.

The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

### execute_alerts

Enable or disable alerting rule execution. The default value is `true`. The alerting UI remains visible. This option has a [legacy version in the alerting section]({{< relref "#execute_alerts-1">}}) that takes precedence.

### evaluation_timeout

Sets the alert evaluation timeout when fetching data from the datasource. The default value is `30s`. This option has a [legacy version in the alerting section]({{< relref "#evaluation_timeout_seconds">}}) that takes precedence.

The timeout string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

### max_attempts

Sets a maximum number of times we'll attempt to evaluate an alert rule before giving up on that evaluation. The default value is `3`. This option has a [legacy version in the alerting section]({{< relref "#max_attempts-1">}}) that takes precedence.

### min_interval

Sets the minimum interval to enforce between rule evaluations. The default value is `10s` which equals the scheduler interval. Rules will be adjusted if they are less than this value or if they are not multiple of the scheduler interval (10s). Higher values can help with resource management as we'll schedule fewer evaluations over time. This option has [a legacy version in the alerting section]({{< relref "#min_interval_seconds">}}) that takes precedence.

The interval string is a possibly signed sequence of decimal numbers, followed by a unit suffix (ms, s, m, h, d), e.g. 30s or 1m.

> **Note.** This setting has precedence over each individual rule frequency. If a rule frequency is lower than this value, then this value is enforced.

## [alerting]

For more information about the legacy dashboard alerting feature in Grafana, refer to [Alerts overview]({{< relref "../alerting/_index.md" >}}).

### enabled

Set to `false` to [enable Grafana alerting]({{<relref "#unified_alerting">}}) and to disable legacy alerting engine. to disable Grafana alerting, set to `true`.

### execute_alerts

Turns off alert rule execution, but alerting is still visible in the Grafana UI.

### error_or_timeout

Default setting for new alert rules. Defaults to categorize error and timeouts as alerting. (alerting, keep_state)

### nodata_or_nullvalues

Defines how Grafana handles nodata or null values in alerting. Options are `alerting`, `no_data`, `keep_state`, and `ok`. Default is `no_data`.

### concurrent_render_limit

Alert notifications can include images, but rendering many images at the same time can overload the server. This limit protects the server from render overloading and ensures notifications are sent out quickly. Default value is `5`.

### evaluation_timeout_seconds

Sets the alert calculation timeout. Default value is `30`.

### notification_timeout_seconds

Sets the alert notification timeout. Default value is `30`.

### max_attempts

Sets a maximum limit on attempts to sending alert notifications. Default value is `3`.

### min_interval_seconds

Sets the minimum interval between rule evaluations. Default value is `1`.

> **Note.** This setting has precedence over each individual rule frequency. If a rule frequency is lower than this value, then this value is enforced.

### max_annotation_age =

Configures for how long alert annotations are stored. Default is 0, which keeps them forever. This setting should be expressed as a duration. Examples: 6h (hours), 10d (days), 2w (weeks), 1M (month).

### max_annotations_to_keep =

Configures max number of alert annotations that Grafana stores. Default value is 0, which keeps all alert annotations.

## [annotations]

### cleanupjob_batchsize

Configures the batch size for the annotation clean-up job. This setting is used for dashboard, API, and alert annotations.

## [annotations.dashboard]

Dashboard annotations means that annotations are associated with the dashboard they are created on.

### max_age

Configures how long dashboard annotations are stored. Default is 0, which keeps them forever. This setting should be expressed as a duration. Examples: 6h (hours), 10d (days), 2w (weeks), 1M (month).

### max_annotations_to_keep

Configures max number of dashboard annotations that Grafana stores. Default value is 0, which keeps all dashboard annotations.

## [annotations.api]

API annotations means that the annotations have been created using the API without any association with a dashboard.

### max_age

Configures how long Grafana stores API annotations. Default is 0, which keeps them forever. This setting should be expressed as a duration. Examples: 6h (hours), 10d (days), 2w (weeks), 1M (month).

### max_annotations_to_keep

Configures max number of API annotations that Grafana keeps. Default value is 0, which keeps all API annotations.

## [explore]

For more information about this feature, refer to [Explore]({{< relref "../explore/_index.md" >}}).

### enabled

Enable or disable the Explore section. Default is `enabled`.

## [help]

Configures the help section.

### enabled

Enable or disable the Help section. Default is `enabled`.

## [profile]

Configures the Profile section.

### enabled

Enable or disable the Profile section. Default is `enabled`.

## [metrics]

For detailed instructions, refer to [Internal Grafana metrics]({{< relref "view-server/internal-metrics.md" >}}).

### enabled

Enable metrics reporting. defaults true. Available via HTTP API `<URL>/metrics`.

### interval_seconds

Flush/write interval when sending metrics to external TSDB. Defaults to `10`.

### disable_total_stats

If set to `true`, then total stats generation (`stat_totals_*` metrics) is disabled. Default is `false`.

### basic_auth_username and basic_auth_password

If both are set, then basic authentication is required to access the metrics endpoint.

## [metrics.environment_info]

Adds dimensions to the `grafana_environment_info` metric, which can expose more information about the Grafana instance.

```
; exampleLabel1 = exampleValue1
; exampleLabel2 = exampleValue2
```

## [metrics.graphite]

Use these options if you want to send internal Grafana metrics to Graphite.

### address

Enable by setting the address. Format is `<Hostname or ip>`:port.

### prefix

Graphite metric prefix. Defaults to `prod.grafana.%(instance_name)s.`

## [grafana_net]

### url

Default is https://grafana.com.

## [grafana_com]

### url

Default is https://grafana.com.

## [tracing.jaeger]

Configure Grafana's Jaeger client for distributed tracing.

You can also use the standard `JAEGER_*` environment variables to configure Jaeger. See the table at the end of https://www.jaegertracing.io/docs/1.16/client-features/ for the full list. Environment variables will override any settings provided here.

### address

The host:port destination for reporting spans. (ex: `localhost:6831`)

Can be set with the environment variables `JAEGER_AGENT_HOST` and `JAEGER_AGENT_PORT`.

### always_included_tag

Comma-separated list of tags to include in all new spans, such as `tag1:value1,tag2:value2`.

Can be set with the environment variable `JAEGER_TAGS` (use `=` instead of `:` with the environment variable).

### sampler_type

Default value is `const`.

Specifies the type of sampler: `const`, `probabilistic`, `ratelimiting`, or `remote`.

Refer to https://www.jaegertracing.io/docs/1.16/sampling/#client-sampling-configuration for details on the different tracing types.

Can be set with the environment variable `JAEGER_SAMPLER_TYPE`.

### sampler_param

Default value is `1`.

This is the sampler configuration parameter. Depending on the value of `sampler_type`, it can be `0`, `1`, or a decimal value in between.

- For `const` sampler, `0` or `1` for always `false`/`true` respectively
- For `probabilistic` sampler, a probability between `0` and `1.0`
- For `rateLimiting` sampler, the number of spans per second
- For `remote` sampler, param is the same as for `probabilistic` and indicates the initial sampling rate before the actual one is received from the mothership

May be set with the environment variable `JAEGER_SAMPLER_PARAM`.

### sampling_server_url

sampling_server_url is the URL of a sampling manager providing a sampling strategy.

### zipkin_propagation

Default value is `false`.

Controls whether or not to use Zipkin's span propagation format (with `x-b3-` HTTP headers). By default, Jaeger's format is used.

Can be set with the environment variable and value `JAEGER_PROPAGATION=b3`.

### disable_shared_zipkin_spans

Default value is `false`.

Setting this to `true` turns off shared RPC spans. Leaving this available is the most common setting when using Zipkin elsewhere in your infrastructure.

## [external_image_storage]

These options control how images should be made public so they can be shared on services like Slack or email message.

### provider

Options are s3, webdav, gcs, azure_blob, local). If left empty, then Grafana ignores the upload action.

## [external_image_storage.s3]

### endpoint

Optional endpoint URL (hostname or fully qualified URI) to override the default generated S3 endpoint. If you want to keep the default, just leave this empty. You must still provide a `region` value if you specify an endpoint.

### path_style_access

Set this to true to force path-style addressing in S3 requests, i.e., `http://s3.amazonaws.com/BUCKET/KEY`, instead of the default, which is virtual hosted bucket addressing when possible (`http://BUCKET.s3.amazonaws.com/KEY`).

> **Note:** This option is specific to the Amazon S3 service.

### bucket_url

(for backward compatibility, only works when no bucket or region are configured) Bucket URL for S3. AWS region can be specified within URL or defaults to 'us-east-1', e.g.

- http://grafana.s3.amazonaws.com/
- https://grafana.s3-ap-southeast-2.amazonaws.com/

### bucket

Bucket name for S3. e.g. grafana.snapshot.

### region

Region name for S3. e.g. 'us-east-1', 'cn-north-1', etc.

### path

Optional extra path inside bucket, useful to apply expiration policies.

### access_key

Access key, e.g. AAAAAAAAAAAAAAAAAAAA.

Access key requires permissions to the S3 bucket for the 's3:PutObject' and 's3:PutObjectAcl' actions.

### secret_key

Secret key, e.g. AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.

## [external_image_storage.webdav]

### url

URL where Grafana sends PUT request with images.

### username

Basic auth username.

### password

Basic auth password.

### public_url

Optional URL to send to users in notifications. If the string contains the sequence `${file}`, it is replaced with the uploaded filename. Otherwise, the file name is appended to the path part of the URL, leaving any query string unchanged.

## [external_image_storage.gcs]

### key_file

Optional path to JSON key file associated with a Google service account to authenticate and authorize. If no value is provided it tries to use the application default credentials. Service Account keys can be created and downloaded from https://console.developers.google.com/permissions/serviceaccounts.

Service Account should have "Storage Object Writer" role. The access control model of the bucket needs to be "Set object-level and bucket-level permissions". Grafana itself will make the images public readable when signed urls are not enabled.

### bucket

Bucket Name on Google Cloud Storage.

### path

Optional extra path inside bucket.

### enable_signed_urls

If set to true, Grafana creates a signed URL for the image uploaded to Google Cloud Storage.

### signed_url_expiration

Sets the signed URL expiration, which defaults to seven days.

## [external_image_storage.azure_blob]

### account_name

Storage account name.

### account_key

Storage account key

### container_name

Container name where to store "Blob" images with random names. Creating the blob container beforehand is required. Only public containers are supported.

## [external_image_storage.local]

This option does not require any configuration.

## [rendering]

Options to configure a remote HTTP image rendering service, e.g. using https://github.com/grafana/grafana-image-renderer.

### server_url

URL to a remote HTTP image renderer service, e.g. http://localhost:8081/render, will enable Grafana to render panels and dashboards to PNG-images using HTTP requests to an external service.

### callback_url

If the remote HTTP image renderer service runs on a different server than the Grafana server you may have to configure this to a URL where Grafana is reachable, e.g. http://grafana.domain/.

### concurrent_render_request_limit

Concurrent render request limit affects when the /render HTTP endpoint is used. Rendering many images at the same time can overload the server, which this setting can help protect against by only allowing a certain number of concurrent requests. Default is `30`.

## [panels]

### enable_alpha

Set to `true` if you want to test alpha panels that are not yet ready for general usage. Default is `false`.

### disable_sanitize_html

If set to true Grafana will allow script tags in text panels. Not recommended as it enables XSS vulnerabilities. Default is false. This setting was introduced in Grafana v6.0.

## plugins

### enable_alpha

Set to `true` if you want to test alpha plugins that are not yet ready for general usage. Default is `false`.

### allow_loading_unsigned_plugins

Enter a comma-separated list of plugin identifiers to identify plugins to load even if they are unsigned. Plugins with modified signatures are never loaded.

We do *not* recommend using this option. For more information, refer to [Plugin signatures]({{< relref "../plugins/plugin-signatures.md" >}}).

### plugin_admin_enabled

Available to Grafana administrators only, enables installing / uninstalling / updating plugins directly from the Grafana UI. Set to `true` by default. Setting it to `false` will hide the install / uninstall / update controls.

For more information, refer to [Plugin catalog]({{< relref "../plugins/catalog.md" >}}).

### plugin_admin_external_manage_enabled

Set to `true` if you want to enable external management of plugins. Default is `false`. This is only applicable to Grafana Cloud users.

### plugin_catalog_url

Custom install/learn more URL for enterprise plugins. Defaults to https://grafana.com/grafana/plugins/.

### plugin_catalog_hidden_plugins

Enter a comma-separated list of plugin identifiers to hide in the plugin catalog.

## [live]

### max_connections

> **Note**: Available in Grafana v8.0 and later versions.

The `max_connections` option specifies the maximum number of connections to the Grafana Live WebSocket endpoint per Grafana server instance. Default is `100`.

Refer to [Grafana Live configuration documentation]({{< relref "../live/configure-grafana-live.md" >}}) if you specify a number higher than default since this can require some operating system and infrastructure tuning.

0 disables Grafana Live, -1 means unlimited connections.

### allowed_origins

> **Note**: Available in Grafana v8.0.4 and later versions.

The `allowed_origins` option is a comma-separated list of additional origins (`Origin` header of HTTP Upgrade request during WebSocket connection establishment) that will be accepted by Grafana Live.

If not set (default), then the origin is matched over root_url({{< relref "#root_url" >}}) which should be sufficient for most scenarios.

Origin patterns support wildcard symbol "*".

For example:

```
[live]
allowed_origins = "https://*.example.com"
```

### ha_engine

> **Note**: Available in Grafana v8.1 and later versions.

**Experimental**

The high availability (HA) engine name for Grafana Live. By default, it's not set. The only possible value is "redis".

For more information, refer to [Configure Grafana Live HA setup]({{< relref "../live/live-ha-setup.md" >}}).

### ha_engine_address

> **Note**: Available in Grafana v8.1 and later versions.

**Experimental**

Address string of selected the high availability (HA) Live engine. For Redis, it's a `host:port` string. Example:

```
[live]
ha_engine = redis
ha_engine_address = 127.0.0.1:6379
```

## [plugin.grafana-image-renderer]

For more information, refer to [Image rendering]({{< relref "../image-rendering/" >}}).

### rendering_timezone

Instruct headless browser instance to use a default timezone when not provided by Grafana, e.g. when rendering panel image of alert. See ICUs metaZones.txt for a list of supported timezone IDs. Fallbacks to TZ environment variable if not set.

### rendering_language

Instruct headless browser instance to use a default language when not provided by Grafana, e.g. when rendering panel image of alert. Refer to the HTTP header Accept-Language to understand how to format this value, e.g. 'fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5'.

### rendering_viewport_device_scale_factor

Instruct headless browser instance to use a default device scale factor when not provided by Grafana, e.g. when rendering panel image of alert. Default is `1`. Using a higher value will produce more detailed images (higher DPI), but requires more disk space to store an image.

### rendering_ignore_https_errors

Instruct headless browser instance whether to ignore HTTPS errors during navigation. Per default HTTPS errors are not ignored. Due to the security risk, we do not recommend that you ignore HTTPS errors.

### rendering_verbose_logging

Instruct headless browser instance whether to capture and log verbose information when rendering an image. Default is `false` and will only capture and log error messages.

When enabled, debug messages are captured and logged as well.

For the verbose information to be included in the Grafana server log you have to adjust the rendering log level to debug, configure [log].filter = rendering:debug.

### rendering_dumpio

Instruct headless browser instance whether to output its debug and error messages into running process of remote rendering service. Default is `false`.

It can be useful to set this to `true` when troubleshooting.

### rendering_args

Additional arguments to pass to the headless browser instance. Defaults are `--no-sandbox,--disable-gpu`. The list of Chromium flags can be found at (https://peter.sh/experiments/chromium-command-line-switches/). Separate multiple arguments with commas.

### rendering_chrome_bin

You can configure the plugin to use a different browser binary instead of the pre-packaged version of Chromium.

Please note that this is *not* recommended. You might encounter problems if the installed version of Chrome/Chromium is not compatible with the plugin.

### rendering_mode

Instruct how headless browser instances are created. Default is `default` and will create a new browser instance on each request.

Mode `clustered` will make sure that only a maximum of browsers/incognito pages can execute concurrently.

Mode `reusable` will have one browser instance and will create a new incognito page on each request.

### rendering_clustering_mode

When rendering_mode = clustered, you can instruct how many browsers or incognito pages can execute concurrently. Default is `browser` and will cluster using browser instances.

Mode `context` will cluster using incognito pages.

### rendering_clustering_max_concurrency

When rendering_mode = clustered, you can define the maximum number of browser instances/incognito pages that can execute concurrently. Default is `5`.

### rendering_clustering_timeout

> **Note**: Available in grafana-image-renderer v3.3.0 and later versions.

When rendering_mode = clustered, you can specify the duration a rendering request can take before it will time out. Default is `30` seconds.

### rendering_viewport_max_width

Limit the maximum viewport width that can be requested.

### rendering_viewport_max_height

Limit the maximum viewport height that can be requested.

### rendering_viewport_max_device_scale_factor

Limit the maximum viewport device scale factor that can be requested.

### grpc_host

Change the listening host of the gRPC server. Default host is `127.0.0.1`.

### grpc_port

Change the listening port of the gRPC server. Default port is `0` and will automatically assign a port not in use.

## [enterprise]

For more information about Grafana Enterprise, refer to [Grafana Enterprise]({{< relref "../enterprise/_index.md" >}}).

## [feature_toggles]

### enable

Keys of alpha features to enable, separated by space.

## [date_formats]

> **Note:** The date format options below are only available in Grafana v7.2+.

This section controls system-wide defaults for date formats used in time ranges, graphs, and date input boxes.

The format patterns use Moment.js formatting tokens.

### full_date

Full date format used by time range picker and in other places where a full date is rendered.

### intervals

These intervals formats are used in the graph to show only a partial date or time. For example, if there are only minutes between Y-axis tick labels then the `interval_minute` format is used.

Defaults

```
interval_second = HH:mm:ss
interval_minute = HH:mm
interval_hour = MM/DD HH:mm
interval_day = MM/DD
interval_month = YYYY-MM
interval_year = YYYY
```

### use_browser_locale

Set this to `true` to have date formats automatically derived from your browser location. Defaults to `false`. This is an experimental feature.

### default_timezone

Used as the default time zone for user preferences. Can be either `browser` for the browser local time zone or a time zone name from the IANA Time Zone database, such as `UTC` or `Europe/Amsterdam`.

### default_week_start

Set the default start of the week, valid values are: `saturday`, `sunday`, `monday` or `browser` to use the browser locale to define the first day of the week. Default is `browser`.

## [expressions]

> **Note:** This feature is available in Grafana v7.4 and later versions.

### enabled

Set this to `false` to disable expressions and hide them in the Grafana UI. Default is `true`.

## [geomap]

This section controls the defaults settings for Geomap Plugin.

### default_baselayer_config

The json config used to define the default base map. Four base map options to choose from are `carto`, `esriXYZTiles`, `xyzTiles`, `standard`. For example, to set cartoDB light as the default base layer:

```
default_baselayer_config = `{
  "type": "xyz",
  "config": {
    "attribution": "Open street map",
    "url": "https://tile.openstreetmap.org/{z}/{x}/{y}.png"
  }
}`
```

### enable_custom_baselayers

Set this to `true` to disable loading other custom base maps and hide them in the Grafana UI. Default is `false`.