# Configuration | Grafana Documentation

http://docs.grafana.org/installation/configuration/

**Configuration**

The Grafana back-end has a number of configuration options that can be specified in a .ini configuration file or specified using environment variables.

**Config file locations**

* Default configuration from $WORKING\_DIR/conf/defaults.ini
* Custom configuration from $WORKING\_DIR/conf/custom.ini
* The custom configuration file path can be overridden using the --config parameter

**Note.** If you have installed Grafana using the deb or rpm packages, then your configuration file is located at /etc/grafana/grafana.ini. This path is specified in the Grafana init.d script using--config file parameter.

**Using environment variables**

All options in the configuration file (listed below) can be overridden using environment variables using the syntax:

GF\_<SectionName>\_<KeyName>

Where the section name is the text within the brackets. Everything should be upper case, . should be replaced by \_. For example, given these configuration settings:

# default section

instance\_name = ${HOSTNAME}

[security]

admin\_user = admin

[auth.google]

client\_secret = 0ldS3cretKey

Then you can override them using:

export GF\_DEFAULT\_INSTANCE\_NAME=my-instance

export GF\_SECURITY\_ADMIN\_USER=true

export GF\_AUTH\_GOOGLE\_CLIENT\_SECRET=newS3cretKey

**instance\_name**

Set the name of the grafana-server instance. Used in logging and internal metrics and in 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.

**logs**

Path to where Grafana will store logs. This path is usually specified via command line in the init.d script or the systemd service file. It can be overridden in the configuration file or in the default environment variable file.

## [server]

**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 put a webserver like Nginx or Apache in front of Grafana and have them proxy requests to Grafana.

**protocol**

http or https

**Note** Grafana versions earlier than 3.0 are vulnerable to [POODLE](https://en.wikipedia.org/wiki/POODLE). So we strongly recommend to upgrade to 3.x or use a reverse proxy for ssl termination.

**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 host header does not match 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.

**static\_root\_path**

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

**cert\_file**

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

**cert\_key**

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

**router\_logging**

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

## [database]

Grafana needs a database to store users and dashboards (and other things). By default it is configured to usesqlite3 which is an embedded database (included in the main Grafana binary).

**url**

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

**type**

Either mysql, postgres or sqlite3, it’s your choice.

**path**

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

**host**

Only applicable to MySQL or Postgres. Includes IP or hostname and port. For example, for MySQL running on the same host as Grafana: host = 127.0.0.1:3306

**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 trippel quotes. Ex """#password;"""

**ssl\_mode**

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

**ca\_cert\_path**

(MySQL only) The path to the CA certificate to use. On many linux systems, certs can be found in/etc/ssl/certs.

**client\_key\_path**

(MySQL only) The path to the client key. Only if server requires client authentication.

**client\_cert\_path**

(MySQL only) The path to the client cert. Only if server requires client authentication.

**server\_cert\_name**

(MySQL only) The common name field of the certificate used by the mysql server. Not necessary if ssl\_mode is set to skip-verify.

## [security]

**admin\_user**

The name of the default Grafana admin user (who has full permissions). Defaults to admin.

**admin\_password**

The password of the default Grafana admin. Set once on first-run. Defaults to admin.

**login\_remember\_days**

The number of days the keep me logged in / remember me cookie lasts.

**secret\_key**

Used for signing keep me logged in / remember me cookies.

**disable\_gravatar**

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

## [users]

**allow\_sign\_up**

Set to false to prohibit users from being able to sign up / create user accounts. Defaults to true. The admin user can still create users from the [Grafana Admin Pages](http://docs.grafana.org/reference/admin)

**allow\_org\_create**

Set to false to prohibit users from creating new organizations. Defaults to true.

**auto\_assign\_org**

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

**auto\_assign\_org\_role**

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

auto\_assign\_org\_role = Read Only Editor

## [auth]

**disable\_login\_form**

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

**disable\_signout\_menu**

Set to true to disable the signout link in the side menu. useful if you use auth.proxy, defaults to false.

## [auth.anonymous]

**enabled**

Set to true to enable anonymous access. Defaults to false

**org\_name**

Set the organization name that should be used for anonymous users. If you change your organization name in the Grafana UI this setting needs to be updated to match the new name.

**org\_role**

Specify role for anonymous users. Defaults to Viewer, other valid options are Editor and Admin.

## [auth.github]

You need to create a GitHub application (you find this under the GitHub profile page). When you create the application you will need to specify a callback URL. Specify this as callback:

http://<my\_grafana\_server\_name\_or\_ip>:<grafana\_server\_port>/login/github

This callback URL must match the full HTTP address that you use in your browser to access Grafana, but with the prefix path of /login/github. When the GitHub application is created you will get a Client ID and a Client Secret. Specify these in the Grafana configuration file. For example:

[auth.github]

enabled = true

allow\_sign\_up = true

client\_id = YOUR\_GITHUB\_APP\_CLIENT\_ID

client\_secret = YOUR\_GITHUB\_APP\_CLIENT\_SECRET

scopes = user:email

auth\_url = https://github.com/login/oauth/authorize

token\_url = https://github.com/login/oauth/access\_token

api\_url = https://api.github.com/user

allow\_sign\_up = false

team\_ids =

allowed\_organizations =

Restart the Grafana back-end. You should now see a GitHub login button on the login page. You can now login or sign up with your GitHub accounts.

You may allow users to sign-up via GitHub authentication by setting the allow\_sign\_up option to true. When this option is set to true, any user successfully authenticating via GitHub authentication will be automatically signed up.

**team\_ids**

Require an active team membership for at least one of the given teams on GitHub. If the authenticated user isn’t a member of at least one of the teams they will not be able to register or authenticate with your Grafana instance. For example:

[auth.github]

enabled = true

client\_id = YOUR\_GITHUB\_APP\_CLIENT\_ID

client\_secret = YOUR\_GITHUB\_APP\_CLIENT\_SECRET

scopes = user:email,read:org

team\_ids = 150,300

auth\_url = https://github.com/login/oauth/authorize

token\_url = https://github.com/login/oauth/access\_token

allow\_sign\_up = true

**allowed\_organizations**

Require an active organization membership for at least one of the given organizations on GitHub. If the authenticated user isn’t a member of at least one of the organizations they will not be able to register or authenticate with your Grafana instance. For example

[auth.github]

enabled = true

client\_id = YOUR\_GITHUB\_APP\_CLIENT\_ID

client\_secret = YOUR\_GITHUB\_APP\_CLIENT\_SECRET

scopes = user:email,read:org

auth\_url = https://github.com/login/oauth/authorize

token\_url = https://github.com/login/oauth/access\_token

allow\_sign\_up = true

# space-delimited organization names

allowed\_organizations = github google

## [auth.google]

You need to create a Google project. You can do this in the [Google Developer Console](https://console.developers.google.com/project). When you create the project you will need to specify a callback URL. Specify this as callback:

http://<my\_grafana\_server\_name\_or\_ip>:<grafana\_server\_port>/login/google

This callback URL must match the full HTTP address that you use in your browser to access Grafana, but with the prefix path of /login/google. When the Google project is created you will get a Client ID and a Client Secret. Specify these in the Grafana configuration file. For example:

[auth.google]

enabled = true

client\_id = YOUR\_GOOGLE\_APP\_CLIENT\_ID

client\_secret = YOUR\_GOOGLE\_APP\_CLIENT\_SECRET

scopes = https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email

auth\_url = https://accounts.google.com/o/oauth2/auth

token\_url = https://accounts.google.com/o/oauth2/token

allowed\_domains = mycompany.com mycompany.org

allow\_sign\_up = true

Restart the Grafana back-end. You should now see a Google login button on the login page. You can now login or sign up with your Google accounts. The allowed\_domains option is optional, and domains were separated by space.

You may allow users to sign-up via Google authentication by setting the allow\_sign\_up option to true. When this option is set to true, any user successfully authenticating via Google authentication will be automatically signed up.

## [auth.generic\_oauth]

This option could be used if have your own oauth service.

This callback URL must match the full HTTP address that you use in your browser to access Grafana, but with the prefix path of /login/generic\_oauth.

[auth.generic\_oauth]

enabled = true

client\_id = YOUR\_APP\_CLIENT\_ID

client\_secret = YOUR\_APP\_CLIENT\_SECRET

scopes =

auth\_url =

token\_url =

api\_url =

allowed\_domains = mycompany.com mycompany.org

allow\_sign\_up = true

Set api\_url to the resource that returns [OpenID UserInfo](https://connect2id.com/products/server/docs/api/userinfo) compatible information.

## [auth.basic]

**enabled**

When enabled is true (default) the http api will accept basic authentication.

## [auth.ldap]

**enabled**

Set to true to enable LDAP integration (default: false)

**config\_file**

Path to the LDAP specific configuration file (default: /etc/grafana/ldap.toml)

**allow\_sign\_up**

Allow sign up should almost always be true (default) to allow new Grafana users to be created (if ldap authentication is ok). If set to false only pre-existing Grafana users will be able to login (if ldap authentication is ok).

For details on LDAP Configuration, go to the [LDAP Integration](http://docs.grafana.org/installation/ldap/) page.

## [auth.proxy]

This feature allows you to handle authentication in a http reverse proxy.

**enabled**

Defaults to false

**header\_name**

Defaults to X-WEBAUTH-USER

**header\_property**

Defaults to username but can also be set to email

**auto\_sign\_up**

Set to true to enable auto sign up of users who do not exist in Grafana DB. Defaults to true.

## [session]

**provider**

Valid values are memory, file, mysql, postgres, memcache or redis. Default is file.

**provider\_config**

This option should be configured differently depending on what type of session provider you have configured.

* **file:** session file path, e.g. data/sessions
* **mysql:** go-sql-driver/mysql dsn config string, e.g. user:password@tcp(127.0.0.1:3306)/database\_name
* **postgres:** ex: user=a password=b host=localhost port=5432 dbname=c sslmode=require
* **memcache:** ex: 127.0.0.1:11211
* **redis:** ex: addr=127.0.0.1:6379,pool\_size=100,prefix=grafana

If you use MySQL or Postgres as the session store you need to create the session table manually.

Mysql Example:

CREATE TABLE `session` (

`key` CHAR(16) NOT NULL,

`data` BLOB,

`expiry` INT(11) UNSIGNED NOT NULL,

PRIMARY KEY (`key`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8;

Postgres Example:

CREATE TABLE session (

key CHAR(16) NOT NULL,

data BYTEA,

expiry INTEGER NOT NULL,

PRIMARY KEY (key)

);

Postgres valid sslmode are disable, require (default), verify-ca, and verify-full.

**cookie\_name**

The name of the Grafana session cookie.

**cookie\_secure**

Set to true if you host Grafana behind HTTPS only. Defaults to false.

**session\_life\_time**

How long sessions lasts in seconds. Defaults to 86400 (24 hours).

## [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 & error counts. It is very helpful to us, so please leave this enabled. Counters are sent every 24 hours. Default value is true.

**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.

## [dashboards.json]

If you have a system that automatically builds dashboards as json files you can enable this feature to have the Grafana backend index those json dashboards which will make them appear in regular dashboard search.

**enabled**

true or false. Is disabled by default.

**path**

The full path to a directory containing your json dashboards.

## [smtp]

Email server settings.

**enabled**

defaults to false

**host**

defaults to localhost:25

**user**

In case of SMTP auth, defaults to empty

**password**

In case of SMTP auth, defaults to empty

**cert\_file**

File path to a cert file, defaults to empty

**key\_file**

File path to a key file, defaults to empty

**skip\_verify**

Verify SSL for smtp server? defaults to false

**from\_address**

Address used when sending out emails, defaults to admin@grafana.localhost

**from\_name**

Name to be used when sending out emails, defaults to Grafana

## [log]

**mode**

Either “console”, “file”, “syslog”. Default is console and file Use space to separate multiple modes, e.g. “console file”

**level**

Either “debug”, “info”, “warn”, “error”, “critical”, default is “info”

**filters**

optional settings to set different levels for specific loggers. Ex filters = sqlstore:debug

## [metrics]

**enabled**

Enable metrics reporting. defaults true. Available via HTTP API /api/metrics.

**interval\_seconds**

Flush/Write interval when sending metrics to external TSDB. Defaults to 10s.

## [metrics.graphite]

Include this section if you want to send internal Grafana metrics to Graphite.

**address**

Format <Hostname or ip>:port

**prefix**

Graphite metric prefix. Defaults to prod.grafana.%(instance\_name)s.

## [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-origin.raintank.io](https://snapshots-origin.raintank.io/))

**external\_snapshot\_name**

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

**remove expired snapshot**

Enabled to automatically remove expired snapshots

**remove snapshots after 90 days**

Time to live for snapshots.

## [external\_image\_storage]

These options control how images should be made public so they can be shared on services like slack.

**provider**

You can choose between (s3, webdav). If left empty Grafana will ignore the upload action.

## [external\_image\_storage.s3]

**bucket\_url**

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/> - [https://grafana.s3-cn-north-1.amazonaws.com.cn](https://grafana.s3-cn-north-1.amazonaws.com.cn/)

**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 to where Grafana will send PUT request with images

**username**

basic auth username

**password**

basic auth password

## [alerting]

**enabled**

Defaults to true. Set to false to disable alerting engine and hide Alerting from UI.

**execute\_alerts**

**execute\_alerts = true**

Makes it possible to turn off alert rule execution.

# 配置位置

curl http://admin:admin@192.168.0.179:3000/api/org

# 数据库配置

;type = sqlite3

;host = 127.0.0.1:3306

;name = grafana

;user = root

