GitLab OAuth2 Authentication

 › [Administration](https://grafana.com/docs/grafana/latest/administration/) › [Authentication](https://grafana.com/docs/grafana/latest/auth/) › GitLab

To enable GitLab OAuth2 you must register the application in GitLab. GitLab will generate a client ID and secret key for you to use.

为了启用GitLab OAuth2，你必须使用GitLab注册你的应用。GitLab会生成客户端ID和密钥供你使用。

Create GitLab OAuth keys

You need to [create a GitLab OAuth application](https://docs.gitlab.com/ce/integration/oauth_provider.html). Choose a descriptive *Name*, and use the following *Redirect URI*:

你需要去创建一个GitLab OAuth应用。选择一个描述名字，并使用下面的重定向URI：

https://grafana.example.com/login/gitlab

where https://grafana.example.com is the URL you use to connect to Grafana. Adjust it as needed if you don’t use HTTPS or if you use a different port; for instance, if you access Grafana at http://203.0.113.31:3000, you should use

<http://203.0.113.31:3000/login/gitlab>

https://grafana.example.com 是你使用去连接Grafana的URL。如果你不使用HTTPS或使用不同的端口的话，就调整这个URL；例如，如果你使用http://203.0.113.31:3000来访问Grafana，你应该使用http://203.0.113.31:3000/login/gitlab

Finally, select *api* as the *Scope* and submit the form. Note that if you’re not going to use GitLab groups for authorization (i.e. not setting allowed\_groups, see below), you can select *read\_user* instead of *api* as the *Scope*, thus giving a more restricted access to your GitLab API.

最后，选项api作为Scope并提交表单。注意，如果你不准备去使用GitLab组用于认证（如：不设置allowed\_groups，看下面），你可以选择read\_user代替api作为Scope，这样就使得访问GitLab API更受限制。

You’ll get an *Application Id* and a *Secret* in return; we’ll call them GITLAB\_APPLICATION\_ID and GITLAB\_SECRET respectively for the rest of this section.

你在返回结果中可以获取应用ID和一个机密；我们把它们命名为GITLAB\_APPLICATION\_ID和GITLAB\_SECRET分别用于本节的其余部分。

Enable GitLab in Grafana

在Grafana中启用GitLab

Add the following to your Grafana configuration file to enable GitLab authentication:

在你的Grafana配置文件中添加以下内容来启用GitLab认证：

[auth.gitlab]

enabled = true

allow\_sign\_up = false

client\_id = GITLAB\_APPLICATION\_ID

client\_secret = GITLAB\_SECRET

scopes = api

auth\_url = https://gitlab.com/oauth/authorize

token\_url = https://gitlab.com/oauth/token

api\_url = https://gitlab.com/api/v4

allowed\_groups =

Bash

You may have to set the root\_url option of [server] for the callback URL to be correct. For example in case you are serving Grafana behind a proxy.

Restart the Grafana backend for your changes to take effect.

If you use your own instance of GitLab instead of gitlab.com, adjust auth\_url, token\_url and api\_url accordingly by replacing the gitlab.com hostname with your own.

With allow\_sign\_up set to false, only existing users will be able to login using their GitLab account, but with allow\_sign\_up set to true, *any* user who can authenticate on GitLab will be able to login on your Grafana instance; if you use the public gitlab.com, it means anyone in the world would be able to login on your Grafana instance.

You can limit access to only members of a given group or list of groups by setting the allowed\_groups option.

allowed\_groups

To limit access to authenticated users that are members of one or more [GitLab groups](https://docs.gitlab.com/ce/user/group/index.html), set allowed\_groups to a comma- or space-separated list of groups. For instance, if you want to only give access to members of the example group, set

allowed\_groups = example

Ini

If you want to also give access to members of the subgroup bar, which is in the group foo, set

allowed\_groups = example, foo/bar

Ini

Note that in GitLab, the group or subgroup name doesn’t always match its display name, especially if the display name contains spaces or special characters. Make sure you always use the group or subgroup name as it appears in the URL of the group or subgroup.

Here’s a complete example with allow\_sign\_up enabled, and access limited to the example and foo/bar groups:

[auth.gitlab]

enabled = true

allow\_sign\_up = true

client\_id = GITLAB\_APPLICATION\_ID

client\_secret = GITLAB\_SECRET

scopes = api

auth\_url = https://gitlab.com/oauth/authorize

token\_url = https://gitlab.com/oauth/token

api\_url = https://gitlab.com/api/v4

allowed\_groups = example, foo/bar

Ini

Team Sync (Enterprise only)

团队同步（仅限企业版）

Only available in Grafana Enterprise v6.4+

只在Grafana企业版6.4版本后可用

With Team Sync you can map your GitLab groups to teams in Grafana so that your users will automatically be added to the correct teams.

通过团队同步，你可以映射你的GitLab组到Grafana的团队中，这样你的用户就可以自动被添加到正确的团队中。

Your GitLab groups can be referenced in the same way as allowed\_groups, like example or foo/bar.

你的GitLab组能够像allowed\_groups一样的方法被引用，像example或foo/bar。

[Learn more about Team Sync](https://grafana.com/docs/grafana/latest/auth/team-sync/)