**How To Install and Secure Grafana on Ubuntu 18.04**

Introduction

[Grafana](https://grafana.com/grafana) is an open-source data visualization and monitoring tool that integrates with complex data from sources like [Prometheus](https://prometheus.io/), [InfluxDB](https://www.influxdata.com/), [Graphite](https://graphiteapp.org/), and [ElasticSearch](https://www.elastic.co/). Grafana lets you create alerts, notifications, and ad-hoc filters for your data while also making collaboration with your teammates easier through built-in sharing features.

In this tutorial, you will install Grafana and secure it with an [SSL certificate](https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs) and an [Nginx reverse proxy](https://www.digitalocean.com/community/tutorials/understanding-nginx-http-proxying-load-balancing-buffering-and-caching). Once you have set up Grafana, you’ll have the option to configure user authentication through GitHub, allowing you to better organize your team permissions.

Prerequisites

To follow this tutorial, you will need:

* One Ubuntu 18.04 server set up by following the [Initial Server Setup Guide for Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04), including a non-root user with sudo privileges and a firewall configured with ufw.
* A fully registered domain name. This tutorial uses your\_domain throughout. You can purchase a domain name on [Namecheap](https://namecheap.com/), get one for free on [Freenom](http://www.freenom.com/en/index.html), or use the domain registrar of your choice.
* The following DNS records set up for your server. You can follow [How To Set Up a Host Name with DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-set-up-a-host-name-with-digitalocean) for details on how to add them.
  + An **A** record with your\_domain pointing to your server’s public IP address.
  + An **A** record with www.your\_domain pointing to your server’s public IP address.
* Nginx set up by following the [How To Install Nginx on Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04) tutorial, including a [server block](https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04#step-5-%E2%80%93-setting-up-server-blocks-(recommended)) for your domain.
* An Nginx server block with Let’s Encrypt configured, which you can set up by following [How To Secure Nginx with Let’s Encrypt on Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04).
* Optionally, to set up [GitHub](https://github.com/) authentication, you’ll need a [GitHub account associated with an organization](https://github.com/business).

Step 1 — Installing Grafana

In this first step, you will install Grafana onto your Ubuntu 18.04 server. You can install Grafana either by [downloading it directly from its official website](https://grafana.com/grafana/download) or by going through an [APT repository](https://www.digitalocean.com/community/tutorials/ubuntu-and-debian-package-management-essentials#debian-package-management-tools-overview). Because an APT repository makes it easier to install and manage Grafana’s updates, you’ll use that method in this tutorial.

Although Grafana is available in [the official Ubuntu 18.04 packages repository](https://packages.ubuntu.com/bionic/), the version of Grafana there may not be the latest, so use Grafana’s official repository.

Download the Grafana [GPG key](https://www.digitalocean.com/community/tutorials/how-to-use-gpg-to-encrypt-and-sign-messages) with [wget](https://www.gnu.org/software/wget/), then [pipe the output](https://www.digitalocean.com/community/tutorials/an-introduction-to-linux-i-o-redirection#pipes) to apt-key. This will add the key to your APT installation’s list of trusted keys, which will allow you to download and verify the GPG-signed Grafana package.

* wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

Copy

In this command, the option -q turns off the status update message for wget, and -O outputs the file that you downloaded to the terminal. These two options ensure that only the contents of the downloaded file are pipelined to apt-key.

Next, add the Grafana repository to your APT sources:

* sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"

Copy

Refresh your APT cache to update your package lists:

* sudo apt update

Copy

Next, make sure Grafana will be installed from the Grafana repository:

* apt-cache policy grafana

Copy

The output of the previous command tells you the version of Grafana that you are about to install, and where you will retrieve the package from. Verify that the installation candidate at the top of the list will come from the official Grafana repository at https://packages.grafana.com/oss/deb.

Output of apt-cache policy grafana

grafana:

Installed: (none)

Candidate: 6.3.3

Version table:

6.3.3 500

500 https://packages.grafana.com/oss/deb stable/main amd64 Packages

...

You can now proceed with the installation:

* sudo apt install grafana

Copy

Once Grafana is installed, use systemctl to start the Grafana server:

* sudo systemctl start grafana-server

Copy

Next, verify that Grafana is running by checking the service’s status:

* sudo systemctl status grafana-server

Copy

You will receive output similar to this:

Output of grafana-server status

● grafana-server.service - Grafana instance

Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)

Active: active (running) since Tue 2019-08-13 08:22:30 UTC; 11s ago

Docs: http://docs.grafana.org

Main PID: 13630 (grafana-server)

Tasks: 7 (limit: 1152)

...

This output contains information about Grafana’s process, including its status, Main Process Identifier (PID), and more. active (running) shows that the process is running correctly.

Lastly, enable the service to automatically start Grafana on boot:

* sudo systemctl enable grafana-server

Copy

You will receive the following output:

Output of systemctl enable grafana-server

Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemd-sysv-install.

Executing: /lib/systemd/systemd-sysv-install enable grafana-server

Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /usr/lib/systemd/system/grafana-server.service.

This confirms that systemd has created the necessary symbolic links to autostart Grafana.

Grafana is now installed and ready for use. Next, you wil secure your connection to Grafana with a reverse proxy and SSL certificate.

Step 2 — Setting Up the Reverse Proxy

Using an SSL certificate will ensure that your data is secure by encrypting the connection to and from Grafana. But, to make use of this connection, you’ll first need to reconfigure Nginx as a reverse proxy for Grafana.

Open the Nginx configuration file you created when you set up the Nginx server block with Let’s Encrypt in the [Prerequisites](https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-grafana-on-ubuntu-18-04#prerequisites). You can use any text editor, but for this tutorial we’ll use nano:

* sudo nano /etc/nginx/sites-available/your\_domain

Copy

Locate the following block:

/etc/nginx/sites-available/your\_domain

...

location / {

try\_files $uri $uri/ =404;

}

...

Copy

Because you already configured Nginx to communicate over SSL and because all web traffic to your server already passes through Nginx, you just need to tell Nginx to forward all requests to Grafana, which runs on port 3000 by default.

Delete the existing try\_files line in this location block and replace it with the following proxy\_pass option.

/etc/nginx/sites-available/your\_domain

...

location / {

proxy\_pass http://localhost:3000;

}

...

Copy

This will map the proxy to the appropriate port. Once you’re done, save and close the file by pressing CTRL+X, followed by Y and then ENTER if you’re using nano.

Now, test the new settings to make sure everything is configured correctly:

* sudo nginx -t

Copy

You will receive the following output:

Output

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

Finally, activate the changes by reloading Nginx:

* sudo systemctl reload nginx

Copy

You can now access the default Grafana login screen by pointing your web browser to https://your\_domain. If you’re unable to reach Grafana, verify that your firewall is set to allow traffic on port 443 and then re-trace the previous instructions.

With the connection to Grafana encrypted, you can now implement additional security measures, starting with changing Grafana’s default administrative credentials.

Step 3 — Updating Credentials

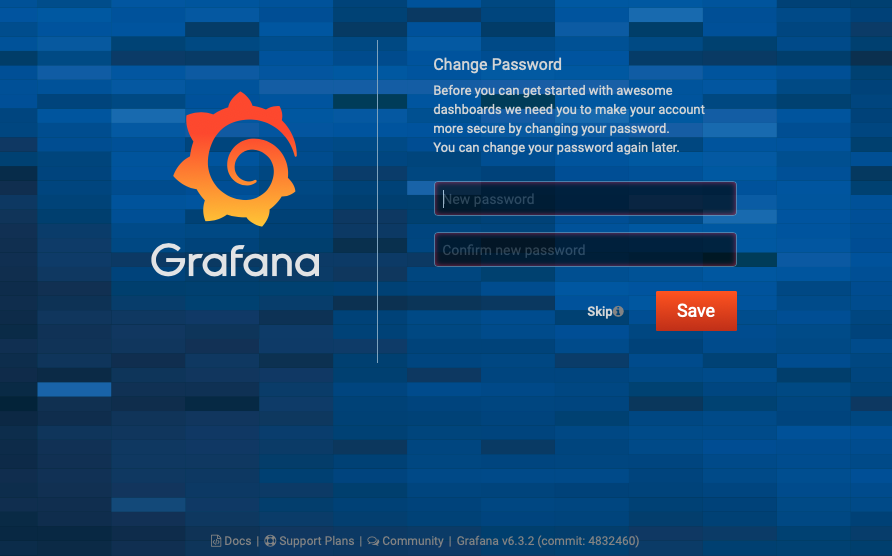
Because every Grafana installation uses the same administrative credentials by default, it is best practice to change your login information as soon as possible. In this step, you’ll update the credentials to improve security.

Start by navigating to https://your\_domain from your web browser. This will bring up the default login screen where you’ll see the Grafana logo, a form asking you to enter an **email or username** and **password**, a **Log in** button, and a **Forgot your password?** link.



Enter admin into both the **User** and **Password** fields and then click on the **Log in** button.

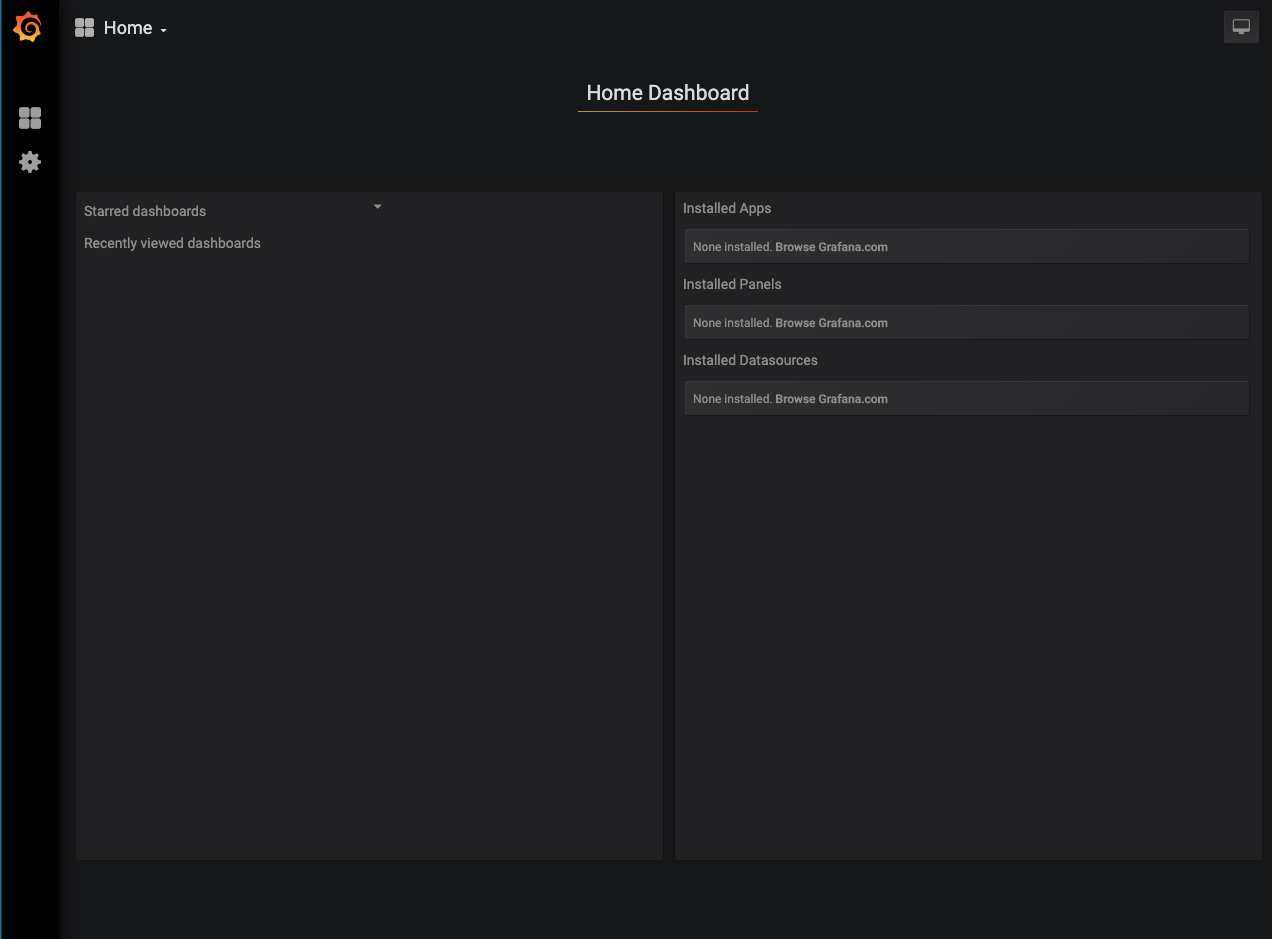
On the next screen, you’ll be asked to make your account more secure by changing the default password:



Enter the password you’d like to start using into the **New password** and **Confirm new password** fields.

From here, you can click **Save** to save the new information or press **Skip** to skip this step. If you skip, you will be prompted to change the password next time you login.

In order to increase the security of your Grafana setup, click **Save**. You’ll return to the **Home Dashboard** page:



You’ve now secured your account by changing the default credentials. Next, you will make changes to your Grafana configuration so that nobody can create a new Grafana account without your permission.

Step 4 — Disabling Grafana Registrations and Anonymous Access

Grafana provides options that allow visitors to create user accounts for themselves and preview dashboards without registering. When Grafana isn’t accessible via the internet or when it’s working with publicly available data like service statuses, you may want to allow these features. However, when using Grafana online to work with sensitive data, anonymous access could be a security problem. To fix this problem, make some changes to your Grafana configuration.

Start by opening Grafana’s main configuration file for editing:

* sudo nano /etc/grafana/grafana.ini

Copy

Locate the following allow\_sign\_up directive under the [users] heading:

/etc/grafana/grafana.ini

...

[users]

# disable user signup / registration

;allow\_sign\_up = true

...

Copy

Enabling this directive with true adds a **Sign Up** button to the login screen, allowing users to register themselves and access Grafana.

Disabling this directive with false removes the **Sign Up** button and strengthens Grafana’s security and privacy.

Uncomment this directive by removing the ; at the beginning of the line and then setting the option to false:

/etc/grafana/grafana.ini

...

[users]

# disable user signup / registration

allow\_sign\_up = false

...

Copy

Next, locate the following enabled directive under the [auth.anonymous] heading:

/etc/grafana/grafana.ini

...

[auth.anonymous]

# enable anonymous access

;enabled = false

...

Copy

Setting enabled to true gives non-registered users access to your dashboards; setting this option to false limits dashboard access to registered users only.

Uncomment this directive by removing the ; at the beginning of the line and then setting the option to false.

/etc/grafana/grafana.ini

...

[auth.anonymous]

enabled = false

...

Copy

Save the file and exit your text editor.

To activate the changes, restart Grafana:

* sudo systemctl restart grafana-server

Copy

Verify that everything is working by checking Grafana’s service status:

* sudo systemctl status grafana-server

Copy

Like before, the output will report that Grafana is active (running).

Now, point your web browser to https://your\_domain. To return to the **Sign Up** screen, bring your cursor to your avatar in the lower left of the screen and click on the **Sign out** option that appears.

Once you have signed out, verify that there is no **Sign Up** button and that you can’t sign in without entering login credentials.

At this point, Grafana is fully configured and ready for use. Next, you can simplify the login process for your organization by authenticating through GitHub.

(Optional) Step 5 — Setting Up a GitHub OAuth App

For an alternative approach to signing in, you can configure Grafana to authenticate through GitHub, which provides login access to all members of authorized GitHub organizations. This can be particularly useful when you want to allow multiple developers to collaborate and access metrics without having to create Grafana-specific credentials.

Start by logging into a GitHub account associated with your organization and then navigate to your GitHub profile page at <https://github.com/settings/profile>.

Click on your organization’s name under **Organization settings** in the navigation menu on the left-hand side of the screen.

GitHub Organization settings

On the next screen, you’ll see your **Organization profile** where you can change settings like your **Organization display name**, organization **Email**, and organization **URL**.

Because Grafana uses [OAuth](https://oauth.net/) — an open standard for granting remote third parties access to local resources — to authenticate users through GitHub, you’ll need to create a new [OAuth application within GitHub](https://developer.github.com/apps/building-oauth-apps/creating-an-oauth-app/).

Click the **OAuth Apps** link under **Developer settings** on the lower left-hand side of the screen.

If you don’t already have any OAuth applications associated with your organization on GitHub, you’ll be told there are **No Organization Owned Applications**. Otherwise, you’ll see a list of the OAuth applications already connected to your account.

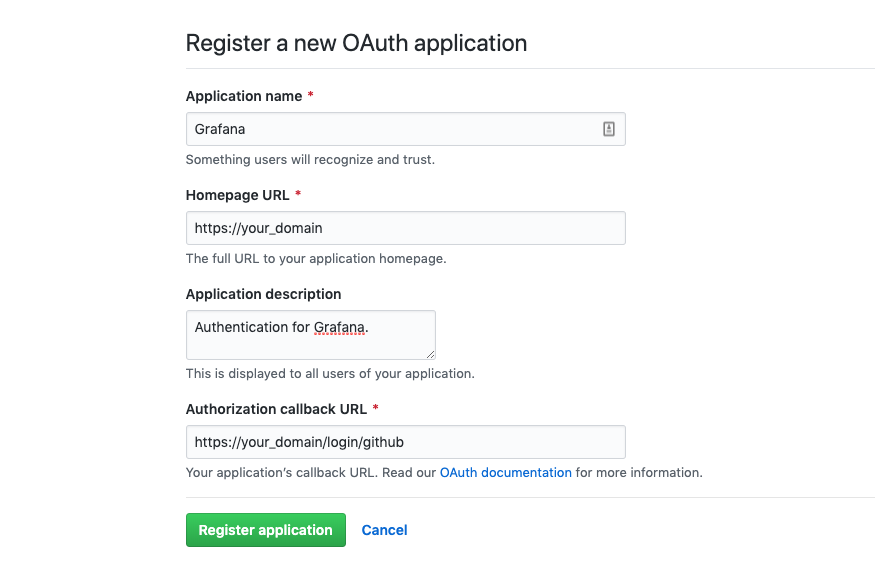
Click the **Register an application** button to continue.

On the next screen, fill in the following details about your Grafana installation:

* **Application name** - This helps you distinguish your different OAuth applications from one another.
* **Homepage URL** - This tells GitHub where to find Grafana. Type https://your\_domain into this field, replacing your\_domain with your domain.
* **Application Description** - This provides a description of your OAuth application’s purpose.
* **Application callback URL** - This is the address where users will be sent once successfully authenticated. For Grafana, this field must be set to https://your\_domain/login/github.

Keep in mind that Grafana users logging in through GitHub will see the values you entered in the first three preceding fields, so be sure to enter something meaningful and appropriate.

When completed, the form will look something like:



Click the green, **Register application** button.

You will now be redirected to a page containing the **Client ID** and **Client Secret** associated with your new OAuth application. Make note of both values, because you will need to add them to Grafana’s main configuration file to complete the setup.

**Warning:** Make sure to keep your **Client ID** and **Client Secret** in a secure and non-public location, because they could be used as the basis of an attack.

With your GitHub OAuth application created, you’re now ready to reconfigure Grafana to use GitHub for authentication.

(Optional) Step 6 — Configuring Grafana as a GitHub OAuth App

To complete GitHub authentication for your Grafana setup, you will now make some changes to your Grafana configuration files.

To begin, open the main Grafana configuration file.

* sudo nano /etc/grafana/grafana.ini

Copy

Locate the [auth.github] heading, and uncomment this section by removing the ; at the beginning of every line except ;team\_ids=, which will not be changed in this tutorial.

Next, configure Grafana to use GitHub with your OAuth application’s client\_id and client\_secret values.

* Set enabled and allow\_sign\_up to true. This will enable GitHub Authentication and permit members of the allowed organization to create accounts themselves. Note that this setting is different than the allow\_sign\_up property under [users] that you changed in [Step 4](https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-grafana-on-ubuntu-18-04#step-4-%E2%80%94-disabling-grafana-registrations-and-anonymous-access).
* Set client\_id and client\_secret to the values you got while creating your GitHub OAuth application.
* Set allowed\_organizations to the name of your organization to ensure that only members of your organization can sign up and log into Grafana.

The complete configuration will look like:

/etc/grafana/grafana.ini

...

[auth.github]

enabled = true

allow\_sign\_up = true

client\_id = your\_client\_id\_from\_github

client\_secret = your\_client\_secret\_from\_github

scopes = user:email,read:org

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

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

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

;team\_ids =

allowed\_organizations = your\_organization\_name

...

Copy

You’ve now told Grafana everything it needs to know about GitHub. To complete the setup, you’ll need to enable redirects behind a reverse proxy. This is done by setting a root\_url value under the [server] heading.

/etc/grafana/grafana.ini

...

[server]

root\_url = https://your\_domain

...

Copy

Save your configuration and close the file.

Then, restart Grafana to activate the changes:

* sudo systemctl restart grafana-server

Copy

Lastly, verify that the service is up and running.

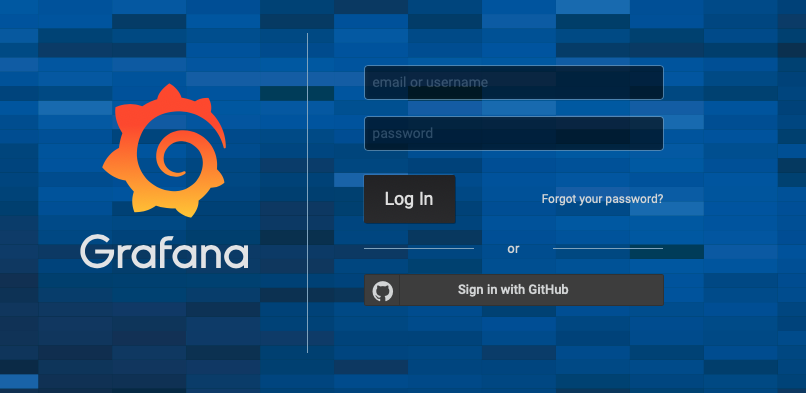
* sudo systemctl status grafana-server

Copy

The output will indicate that the service is active (running).

Now, test your new authentication system by navigating to https://your\_domain. If you are already logged into Grafana, hover your mouse over the avatar log in the lower left-hand corner of the screen, and click on **Sign out** in the secondary menu that appears next to your name.

On the login page, you’ll see a new section under the original **Log in** button that includes a **Sign in with GitHub** button with the GitHub logo.



Click on the **Sign in with GitHub** button to be redirected to GitHub, where you’ll sign into your GitHub account and confirm your intention to **Authorize Grafana**.

Click the green, **Authorize your\_github\_organization** button.

**Note:** Make sure your GitHub account is a member of your approved organization and your Grafana email address matches your GitHub email address. If you try to authenticate with a GitHub account that isn’t a member of your approved organization, you’ll get a **Login Failed** message telling you, **User not a member of one of the required organizations**.

You will now be logged in with your existing Grafana account. If a Grafana account doesn’t already exist for the user you logged in as, Grafana will create a new user account with **Viewer** permissions, ensuring that new users can only use existing dashboards.

To change the default permissions for new users, open the main Grafana configuration file for editing.

* sudo nano /etc/grafana/grafana.ini

Copy

Locate the auto\_assign\_org\_role directive under the [users] heading, and uncomment the setting by removing the ; at the beginning of the line.

Set the directive to one of the following values:

* Viewer — can only use existing dashboards
* Editor — can change use, modify, and add dashboards
* Admin — has permission to do everything

This tutorial will set the auto-assign to Viewer:

/etc/grafana/grafana.ini

...

[users]

...

auto\_assign\_org\_role = Viewer

...

Copy

Once you’ve saved your changes, close the file and restart Grafana:

* sudo systemctl restart grafana-server

Copy

Check the service’s status:

* sudo systemctl status grafana-server

Copy

Like before, the status will read active (running).

At this point, you have fully configured Grafana to allow members of your GitHub organization to register and use your Grafana installation.

Conclusion

Grafana is a well organized and easy to use for Dashboard creation with real time data suitable for control room monitoring all measurements in one company or for top management reviewing their achievement.

You can easily install Grafana on DigitalOcean by one click. Try DigitalOcean Free and earn $100 credit for 60 days trial period by click below:

<https://m.do.co/c/fbc7a836fe39>