**Install Grafana on ubuntu 16.04**

curl https://packagecloud.io/gpg.key | sudo apt-key add -

Next, add the packagecloud repository to your APT sources.

sudo add-apt-repository "deb https://packagecloud.io/grafana/stable/debian/ stretch main"

**Note:** Although this tutorial is written for Ubuntu 16.04, packagecloud only provides Debian, Python, RPM, and RubyGem packages. You can use the Debian-based repository in the previous command, though, because the Grafana package it contains is the same as the one for Ubuntu. Just be sure to use the stretch repository to get the latest version of Grafana.

Refresh your APT cache to update your package lists.

sudo apt-get update

And, make sure Grafana will be installed from the packagecloud repository.

apt-cache policy grafana

The output tells you the version of Grafana that will be installed and where the package will be retrieved from. Verify that the installation candidate will come from the official Grafana repository at https://packagecloud.io/grafana/stable/debian.

Output of apt-cache policy grafana

grafana:

Installed: (none)

Candidate: 4.6.2

Version table:

4.6.2 500

500 https://packagecloud.io/grafana/stable/debian stretch/main amd64 Packages

...

You can now proceed with the installation.

sudo apt-get install grafana

Once Grafana’s installed, you’re ready to start it.

sudo systemctl start grafana-server

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

sudo systemctl status grafana-server

The output contains information about Grafana’s process, including its status, Main Process Identifier (PID), memory use, and more.

If the service status isn’t active (running), review the output and re-trace the preceding steps to resolve the problem.

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 Thu 2017-12-07 12:10:33 UTC; 19s ago

Docs: http://docs.grafana.org

Main PID: 14796 (grafana-server)

Tasks: 6

Memory: 32.0M

CPU: 472ms

CGroup: /system.slice/grafana-server.service

└─14796 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/var/run/grafana/grafana-server.pid cfg:default.paths.logs=/var/log/grafana cfg:default.paths.data=/var/lib/grafana cfg:default.paths.plugins=/var/lib/grafana/plugins

...

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

sudo systemctl enable grafana-server

The output confirms that systemd has created the necessary symbolic links to autostart Grafana. If you receive an error message, follow the instructions in the terminal to fix the problem before continuing.

Output of systemctl enable grafana-server

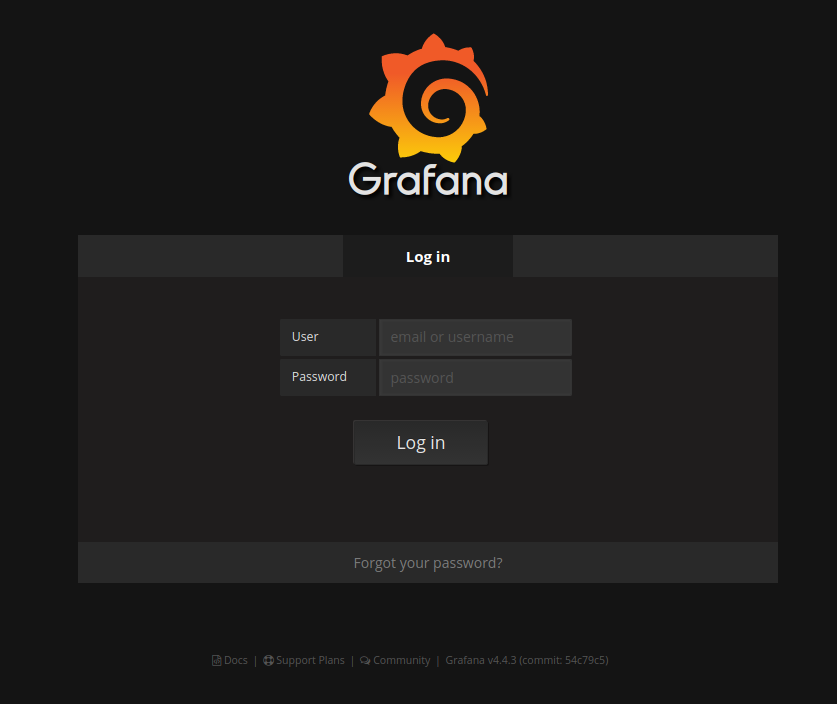
Synchronizing state of grafana-server.service with SysV init with /lib/systemd/systemd-sysv-install...

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

## **Step 3 — Updating Credentials**

Because every Grafana installation uses the same administrative login credentials by default, in this step, you’ll update the credentials to improve security.

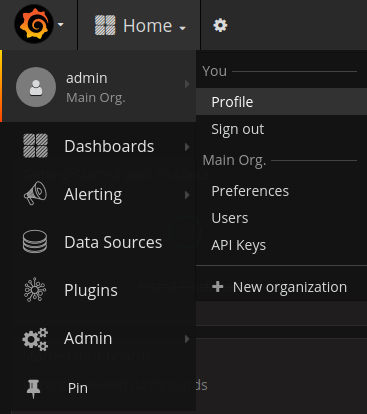
Start by navigating to https://example.com from your web browser. This will bring up the default login screen where you’ll see the Grafana logo, a form asking you to enter a **User** and **Password**, a **Log in** button, and a **Forgot your password?** link.



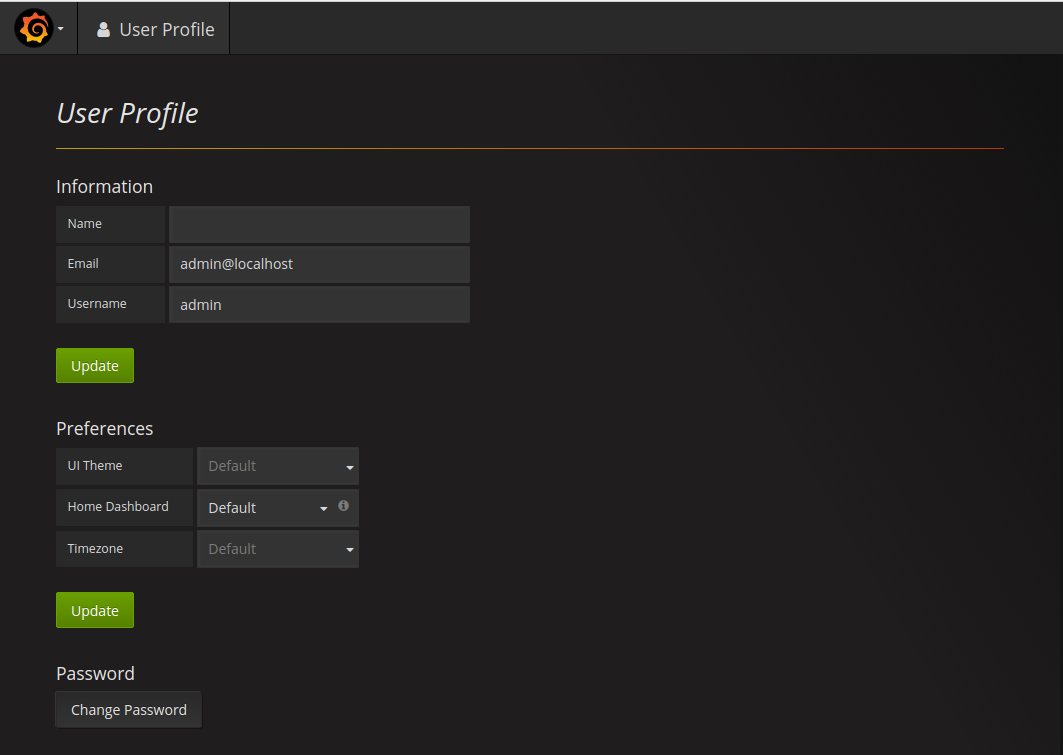
Enter **admin** into both the **User** and **Password** fields and then click on the **Log in** button.

On the next screen, you’ll be welcomed to the **Home Dashboard**. Here you can add data sources and create, preview, and modify dashboards.

Click on the small Grafana logo in the upper, left-hand corner of the screen to bring up the application’s main menu. Then, hover over the **admin** button with your mouse to open up a secondary set of menu options. Finally, click on the **Profile** button.



You’re now on the **User Profile** page, where you can change the **Name**, **Email**, and **Username** associated with your account. You can also update your **Preferences** for settings like the **UI Theme**, and you can change your password.



Enter your name, email address, and the username you want to use in the **Name**, **Email**, and **Username** fields and then click the **Update** button in the **Information** section to save your settings.

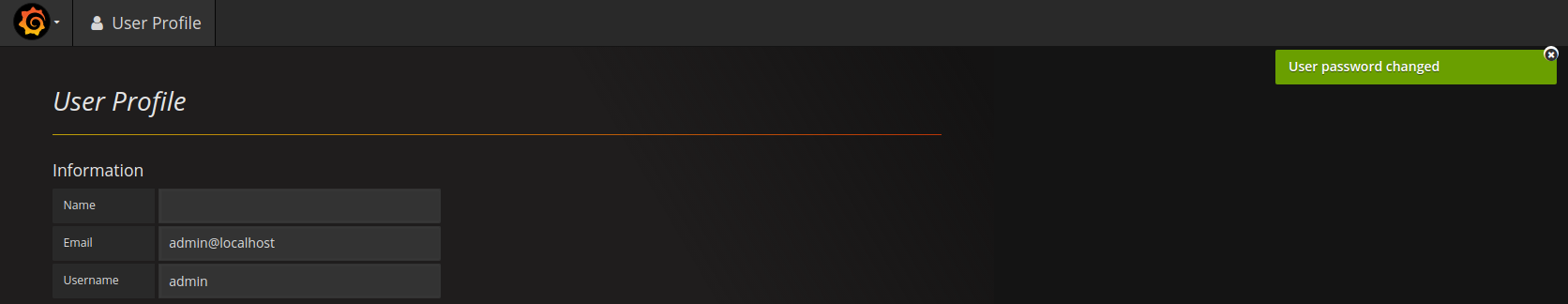
If you want, you can also change the **UI Theme** and **Timezone** to fit your needs and then press the **Update** button in the **Preferences** area to save your changes. Grafana offers **Dark** and **Light** UI themes, as well as a **Default** theme, which is set to **Dark** by default.

Finally, change the password associated with your account by clicking on the **Change Password** button at the bottom of the page. This will take you to the **Change password** screen.

Enter your current password, **admin**, into the **Old Password** field and then enter the password you’d like to start using into the **New Password** and **Confirm Password** fields.

Click **Change Password** to save the new information or press **Cancel** to abandon your changes.

From there, you’ll be returned to the **User Profile** page where you’ll see a green box in the upper, right-hand corner of the screen telling you that the **User password changed**.



You’ve now secured your account by changing the default credentials, so let’s also make sure 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. As you’re exposing Grafana on the internet, this could be a security problem. However, when Grafana isn’t accessible via the internet or when working with publicly-available data, like service statuses, you may want to allow these features. So, it’s important that you know how to configure Grafana to meet your needs.

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

sudo nano /etc/grafana/grafana.ini

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

/etc/grafana/grafana.ini

...

[users]

# disable user signup / registration

;allow\_sign\_up = true

...

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.

Unless you need to allow anonymous visitors to register themselves, uncomment this directive by removing the ; at the beginning of the line and then set the option to false.

/etc/grafana/grafana.ini

...

[users]

# disable user signup / registration

allow\_sign\_up = false

...

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

/etc/grafana/grafana.ini

...

[auth.anonymous]

# enable anonymous access

;enabled = false

...

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

Unless you need to allow anonymous access to your dashboards, uncomment this directive by removing the ; at the beginning of the line and then set the option to false.

/etc/grafana/grafana.ini

...

[auth.anonymous]

enabled = false

...

Save the file and exit your text editor.

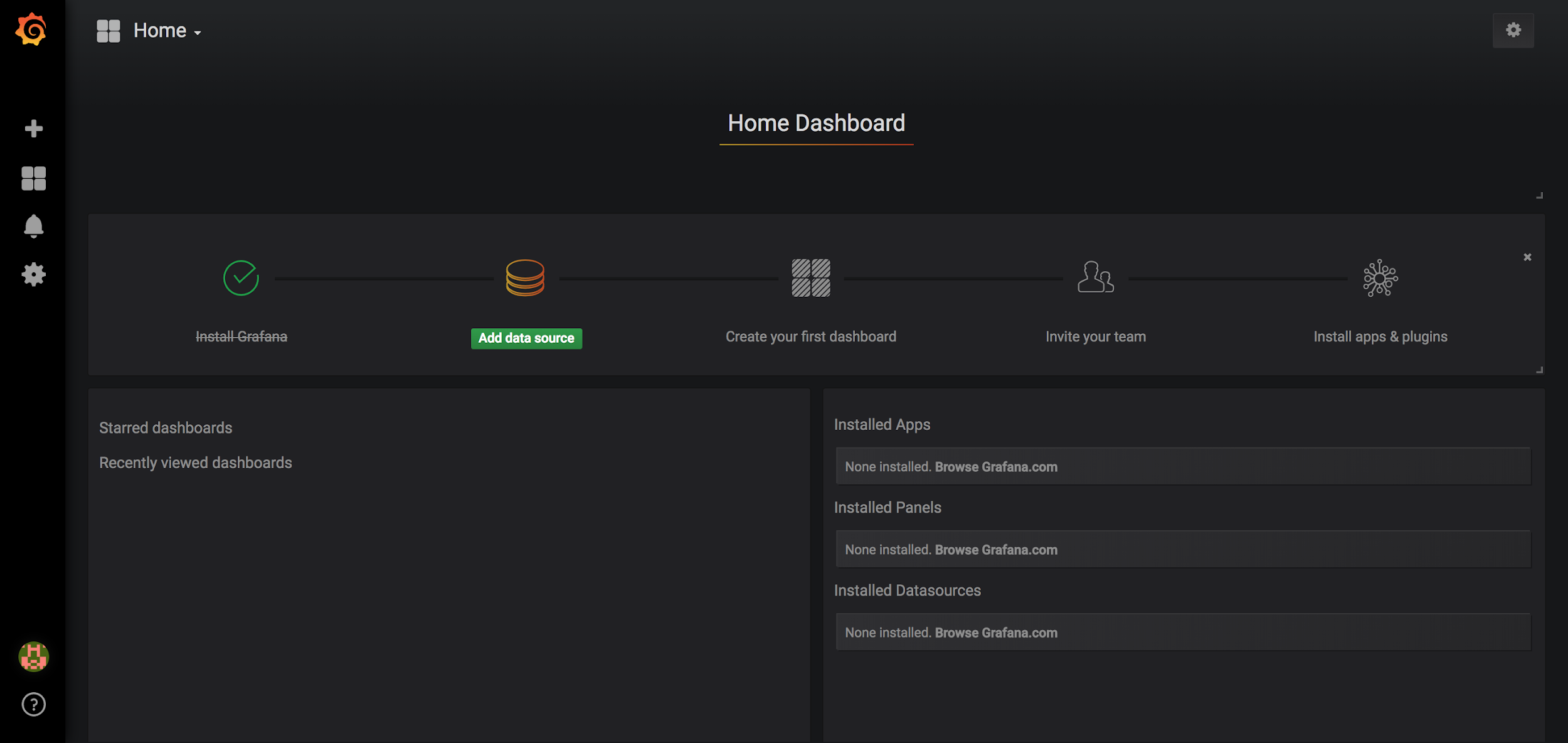
To activate the changes, restart Grafana.

sudo systemctl restart grafana-server

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

sudo systemctl status grafana-server

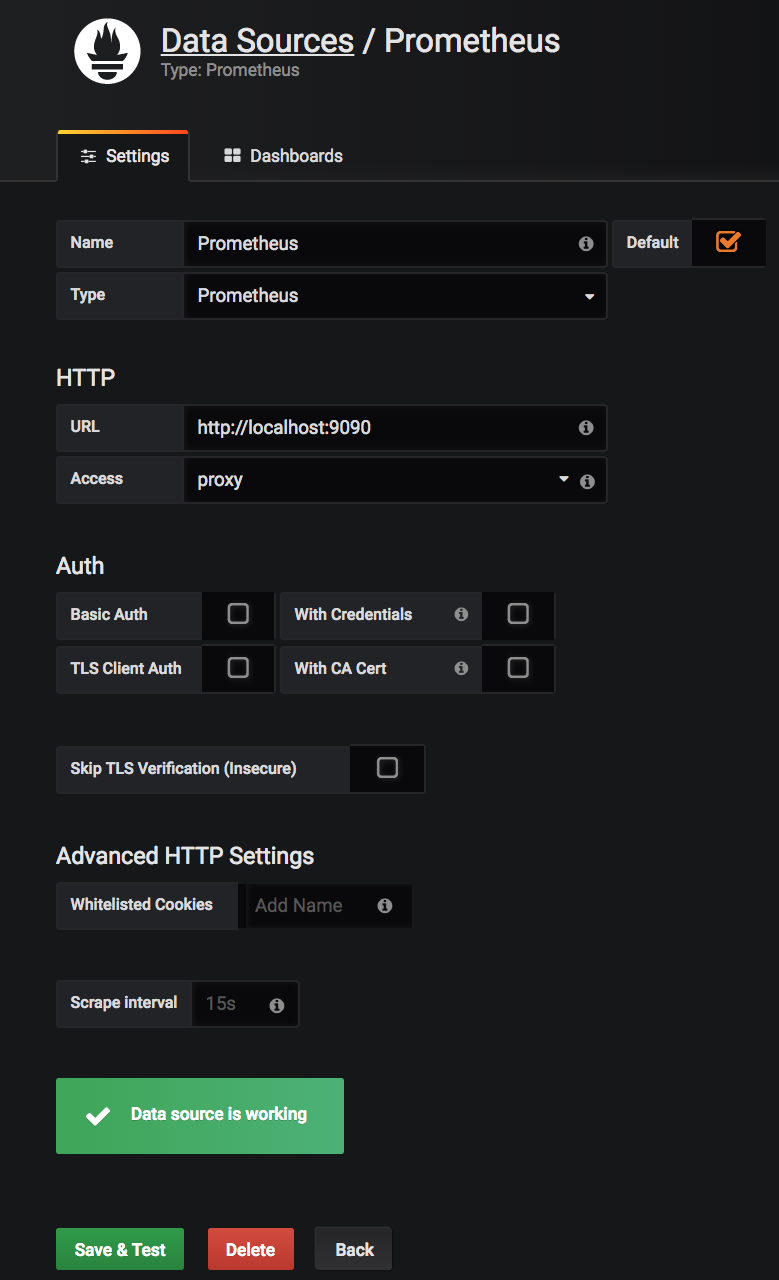
Grafana is running now, and we can connect to it at http://your.server.ip:3000. The default user and password is admin / admin.



Now you have to create a Prometheus data source:

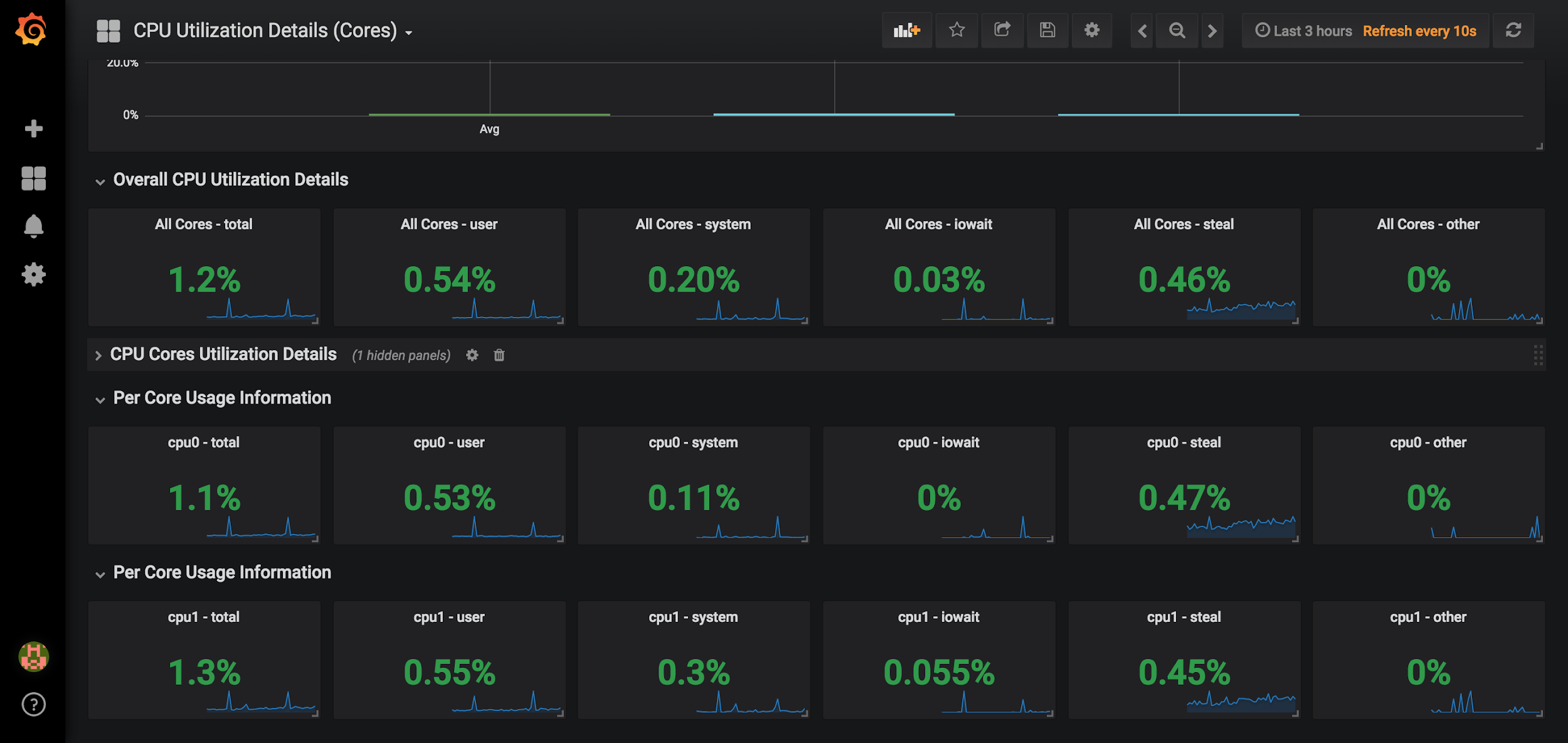
* Click on the Grafana logo to open the sidebar.
* Click on “Data Sources” in the sidebar.
* Choose “Add New”.
* Select “Prometheus” as the data source.
* Set the Prometheus server URL (in our case: http://localhost:9090/)
* Click “Add” to test the connection and to save the new data source.

Your settings should look like this:



You are now ready to create your first dashboard from the information collected by Prometheus. You can also import some dashboards from a collection of [shared dashboards](https://grafana.com/dashboards?dataSource=prometheus)

Here is an example of a Dashboard that uses the CPU usage of our node and presents it in Grafana:



In this tutorial, we were able to configure a Prometheus server with two data collectors that are scraped by our Prometheus server which provides the data to build Dashboards with Grafana. Don’t hesitate to consult the official documentation of [Prometheus](https://prometheus.io/docs/introduction/overview/) and [Grafana](http://docs.grafana.org/).