

1. Step 1: Install and Secure MySQL on Ubuntu

First, you need to get the MySQL server running on your Ubuntu machine.

1. **Update Package Lists:** Open your terminal and refresh your package manager's lists.

```
sudo apt update
```

```
[sudo] password for madhav:  
Get:1 https://packages.grafana.com/oss/deb stable InRelease [7,660 B]  
Get:2 https://packages.grafana.com/oss/deb stable/main amd64 Packages [442 kB]  
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]  
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]  
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [731 kB]  
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]  
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [915 kB]  
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2,008 kB]  
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [497 kB]  
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [54.6 kB]  
Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [4,695 kB]  
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,074 kB]  
Get:14 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [875 kB]  
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]  
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [680 kB]  
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,008 kB]  
Get:18 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [220 kB]  
Get:19 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [125 kB]  
Get:20 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Metadata [268 B]  
Get:21 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [474 kB]  
Get:22 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [112 kB]  
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [4,848 kB]  
Get:24 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [902 kB]  
Get:25 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]  
Get:26 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]  
Get:27 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]  
Get:28 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,152 B]  
Get:29 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]  
Get:30 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [9,660 B]  
Get:31 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]  
Fetched 23.1 MB in 34s (691 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
W: https://packages.grafana.com/oss/deb/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
madhav@madhav-virtual-machine: ~ $
```

2. **Install MySQL Server:** This command installs the server software.

```
sudo apt install mysql-server -y
```

```
madhav@madhav-virtual-machine:~$ sudo apt install mysql-server -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
mysql-server is already the newest version (8.0.43-0ubuntu0.22.04.2).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

3. **Run the Secure Installation Script:** This is a critical step. The script will guide you through setting a root password, removing anonymous users, and securing your server.

```
sudo mysql_secure_installation
```

```

madhav@madhav-virtual-machine:~$ sudo mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

```

```

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
madhav@madhav-virtual-machine:~$ █

```

Note: For the "Validate Password Component," it's safe to say 'no' (n) if you want to use a simple password for this test. For a production server, you would want this enabled.

4. Check Service Status: Verify that the MySQL server is active and running.

sudo systemctl status mysql

```

madhav@madhav-virtual-machine:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2025-11-07 10:19:08 IST; 18min ago
    Main PID: 1165 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 19036)
       Memory: 425.6M
          CPU: 13.219s
        CGroup: /system.slice/mysql.service
                  └─1165 /usr/sbin/mysqld

Nov 07 10:19:01 madhav-virtual-machine systemd[1]: Starting MySQL Community Server...
Nov 07 10:19:08 madhav-virtual-machine systemd[1]: Started MySQL Community Server.

```

2. 📡 Step 2: Create Dummy Database, User, and Data

Now, let's log in to MySQL and create the database Grafana will connect to.

1. Log in to MySQL as Root:

sudo mysql

```
madhav@madhav-virtual-machine: ~ $ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.43-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2. Create a New Database:

Let's call it grafana_dummy_db.

```
CREATE DATABASE grafana_dummy_db;
```

```
mysql> CREATE DATABASE grafana_dummy_db;
Query OK, 1 row affected (0.00 sec)

mysql> █
```

3. Create a New User for Grafana:

It's bad practice to use the root user. Let's create a dedicated, read-only user for Grafana.

- Replace YourStrongPassword with a password of your choice.
- '%' allows the user to connect from any IP address (which is useful if Grafana is on another server or in a Docker container). If Grafana is on the *same machine*, you can use 'localhost'.

```
CREATE USER 'grafana_user'@'%' IDENTIFIED BY 'aDYPu@12345';
```

```
mysql> CREATE USER 'grafana_user'@'%' IDENTIFIED BY 'aDYPu@12345';
Query OK, 0 rows affected (0.01 sec)

mysql> █
```

4. Grant Privileges:

Give your new user SELECT (read-only) permission on the new database.

```
GRANT SELECT ON grafana_dummy_db.* TO 'grafana_user'@'%';
```

```
mysql> GRANT SELECT ON grafana_dummy_db.* TO 'grafana_user'@'%';
Query OK, 0 rows affected (0.00 sec)
```

5. Apply Changes:

```
FLUSH PRIVILEGES;
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

6. Create a Table and Add Data:

Now, let's create a sample table. Time-series data is best for Grafana.

```
-- Switch to your new database
```

```
USE grafana_dummy_db;
```

```
mysql> USE grafana_dummy_db;
Database changed
mysql> █
```

```
-- Create a table to store dummy server metrics
```

```
CREATE TABLE server_metrics (
    time_stamp DATETIME NOT NULL,
    server_name VARCHAR(50) NOT NULL,
    cpu_usage DECIMAL(5, 2),
    memory_usage_mb INT
);
```

```
mysql> CREATE TABLE server_metrics (
    ->     time_stamp DATETIME NOT NULL,
    ->     server_name VARCHAR(50) NOT NULL,
    ->     cpu_usage DECIMAL(5, 2),
    ->     memory_usage_mb INT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
-- Insert some sample data
```

```
INSERT INTO server_metrics (time_stamp, server_name, cpu_usage, memory_usage_mb) VALUES
(NOW() - INTERVAL 4 HOUR, 'web-server-01', 15.5, 1024),
(NOW() - INTERVAL 4 HOUR, 'db-server-01', 25.2, 4096),
(NOW() - INTERVAL 3 HOUR, 'web-server-01', 18.1, 1150),
(NOW() - INTERVAL 3 HOUR, 'db-server-01', 30.0, 4120),
(NOW() - INTERVAL 2 HOUR, 'web-server-01', 12.0, 1000),
(NOW() - INTERVAL 2 HOUR, 'db-server-01', 22.7, 4100),
(NOW() - INTERVAL 1 HOUR, 'web-server-01', 14.2, 1080),
(NOW() - INTERVAL 1 HOUR, 'db-server-01', 28.5, 4150),
(NOW(), 'web-server-01', 20.0, 1200),
(NOW(), 'db-server-01', 35.1, 4200);
```

```
mysql> INSERT INTO server_metrics (time_stamp, server_name, cpu_usage, memory_usage_mb) VALUES
-> (NOW() - INTERVAL 4 HOUR, 'web-server-01', 15.5, 1024),
-> (NOW() - INTERVAL 4 HOUR, 'db-server-01', 25.2, 4096),
-> (NOW() - INTERVAL 3 HOUR, 'web-server-01', 18.1, 1150),
-> (NOW() - INTERVAL 3 HOUR, 'db-server-01', 30.0, 4120),
-> (NOW() - INTERVAL 2 HOUR, 'web-server-01', 12.0, 1000),
-> (NOW() - INTERVAL 2 HOUR, 'db-server-01', 22.7, 4100),
-> (NOW() - INTERVAL 1 HOUR, 'web-server-01', 14.2, 1080),
-> (NOW() - INTERVAL 1 HOUR, 'db-server-01', 28.5, 4150),
-> (NOW(), 'web-server-01', 20.0, 1200),
-> (NOW(), 'db-server-01', 35.1, 4200);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

7. Exit MySQL:

SQL

EXIT;

```
mysql> EXIT;  
Bye  
madhav@madhav-virtual-machine:~$ S
```

3. Step 3: Configure MySQL and Grafana Connection

By default, MySQL only listens for connections from localhost. If your Grafana is on a different machine, you must allow remote connections.

1. Edit MySQL Config File:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
GNU nano 6.2                                     /etc/mysql/mysql.conf.d/mysqld.cnf

The MySQL database server configuration file.

One can use all long options that the program supports.
Run program with --help to get a list of available options and with
--print-defaults to see which it would actually understand and use.

For explanations see
http://dev.mysql.com/doc/mysql/en/server-system-variables.html

Here is entries for some specific programs
The following values assume you have at least 32M ram

[mysqld]

# * Basic Settings

user          = mysql
pid-file     = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
port          = 3306
datadir       = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
tmpdir        = /tmp

Instead of skip-networking the default is now to listen only on
localhost which is more compatible and is not less secure.
bind-address    = 127.0.0.1
mysqlx-bind-address = 127.0.0.1

# * Fine Tuning

key_buffer_size      = 16M
max_allowed_packet   = 64M
thread_stack         = 256K
thread_cache_size    = -1

# This replaces the startup script and checks MyISAM tables if needed

# Help           ^G Write Out   ^Y Where Is   ^K Cut            ^X Execute   ^C Location   M-^U Undo   M-^A Set Mark   M-^T To Bracket   M-^Q Previous   M-^B Back
# Exit          ^D Read File   ^R Replace   ^W Copy          ^E Paste     ^G Go To Line M-^S Redo   M-^P Set Conv   M-^H Where Was   M-^N Next     M-^F Forward
# Prev Word     M-^B Backward M-^F Forward  M-^P Previous M-^N Next     M-^B Backward M-^F Forward  M-^P Previous M-^N Next     M-^B Backward M-^F Forward
```

2. **Change Bind Address:** Find the line that says bind-address = 127.0.0.1. Change it to allow all IPs:

Ini, TOML

bind-address = 0.0.0.0

Press Ctrl+O to save and Ctrl+X to exit.

3. Restart MySQL:

```
sudo systemctl restart mysql
```

4. **(If using a firewall) Allow Port 3306:** If you have ufw (Uncomplicated Firewall) enabled:

```
sudo ufw allow 3306/tcp
```

```
madhav@madhav-virtual-machine:~$ sudo ufw allow 3306/tcp
Rules updated
Rules updated (v6)
```

5. Add Data Source in Grafana:

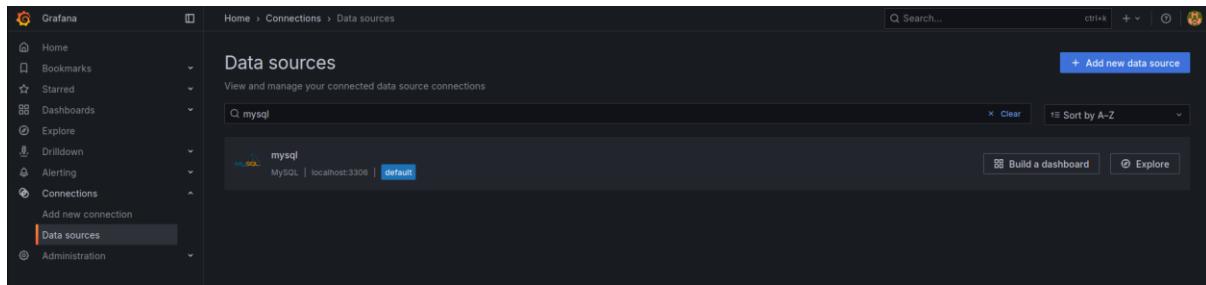
- Log in to your Grafana web interface.

The screenshot shows the Grafana home page at <http://127.0.0.1:3000/>. The left sidebar has 'Data sources' selected. The main area displays a 'Welcome to Grafana' card with sections for 'Basic', 'TUTORIAL DATA SOURCE AND DASHBOARDS', 'Grafana fundamentals', and 'Add your first data source'. Below this, there's a 'Dashboards' section and a 'Latest from the blog' sidebar featuring an article about Grafana Mimir 3.0.

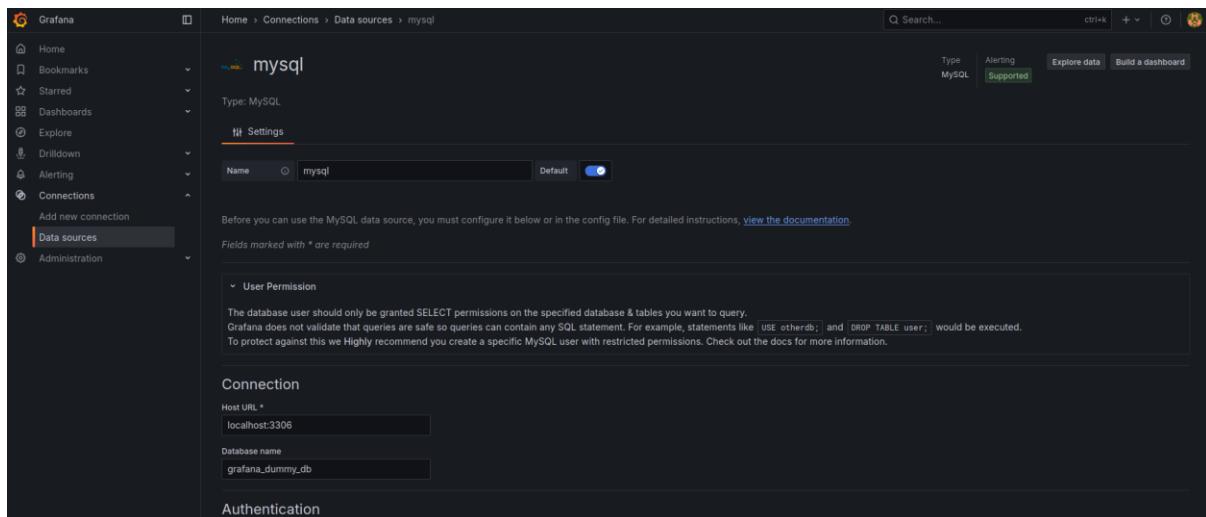
- On the left menu, go to **Connections** (plug icon) -> **Data sources**.

The screenshot shows the Grafana left sidebar. The 'Data sources' option under the 'Connections' section is highlighted with a red border. Other options like Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, and Administration are also listed.

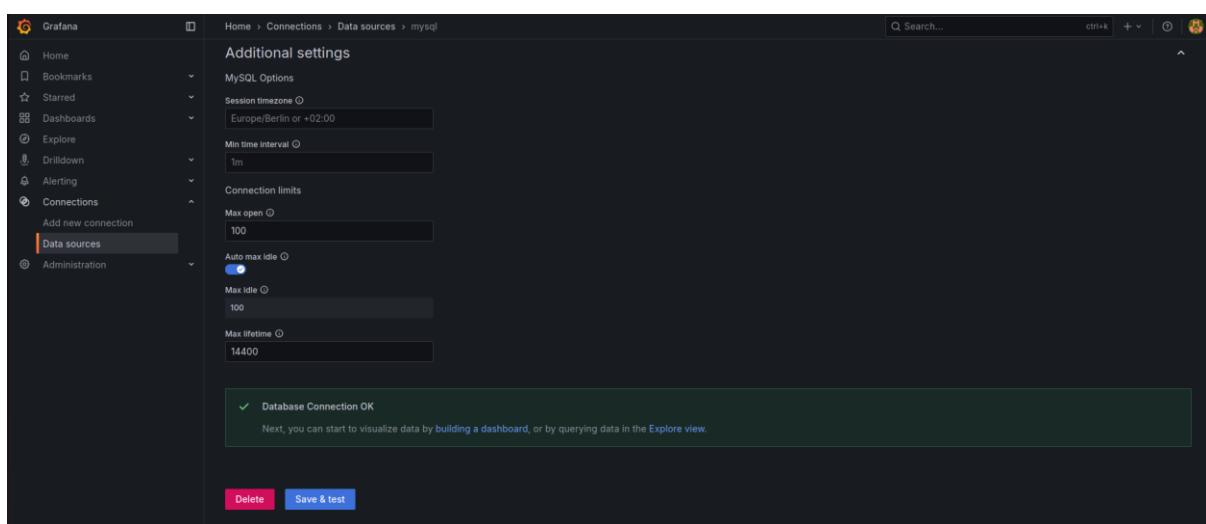
- Click **Add data source**.
- Search for and select **MySQL**.



- Fill in the connection details:
 - **Host:** Your-Ubuntu-Server-IP:3306
 - **Database:** grafana_dummy_db
 - **User:** grafana_user
 - **Password:** YourStrongPassword
 - **TLS/SSL Mode:** disable (for this test)



- Click **Save & test**. You should see a green "Database connection OK" message.

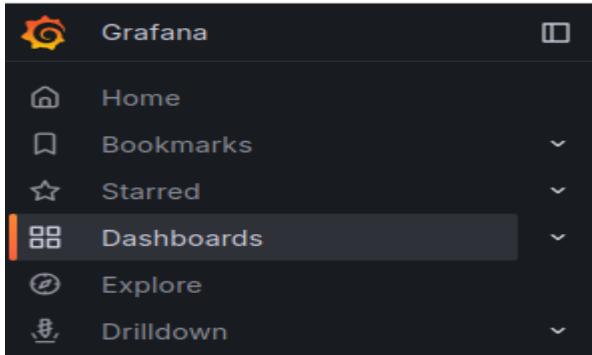


4. Step 4: Create a Grafana Dashboard

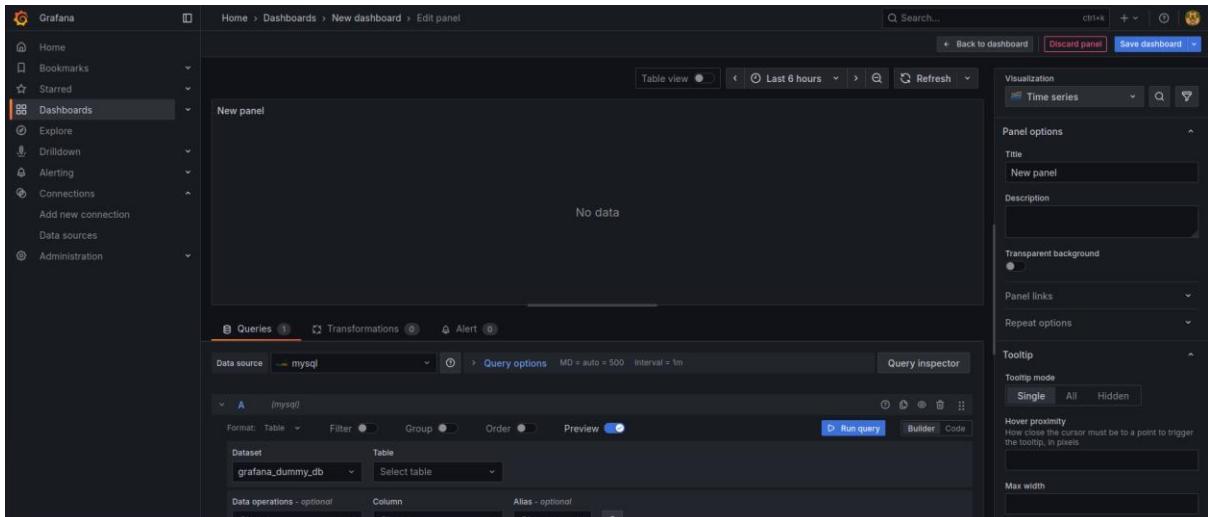
Let's use the dummy data to build a graph.

1. Create a New Dashboard:

- On the left menu, click the **Dashboards** (four squares) icon.

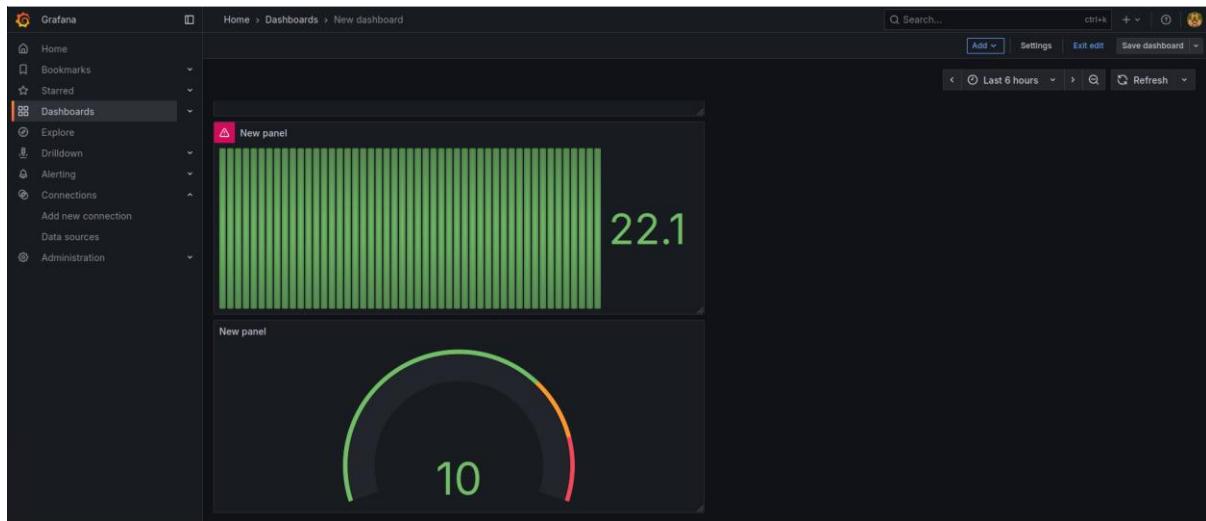


- Click **New -> New Dashboard**.



- Click **Add visualization**.

2. Select Data Source: At the top, select your MySQL data source.



3. Enter the Query:

- At the bottom, click the </> **Code** button to open the query editor.
- Delete any placeholder text and paste in the following SQL query:

SQL

SELECT

```
time_stamp AS "time", -- Grafana needs a column named 'time'  
server_name AS "metric", -- This will name the lines (e.g., 'web-server-01')  
cpu_usage          -- This is the value to plot
```

FROM

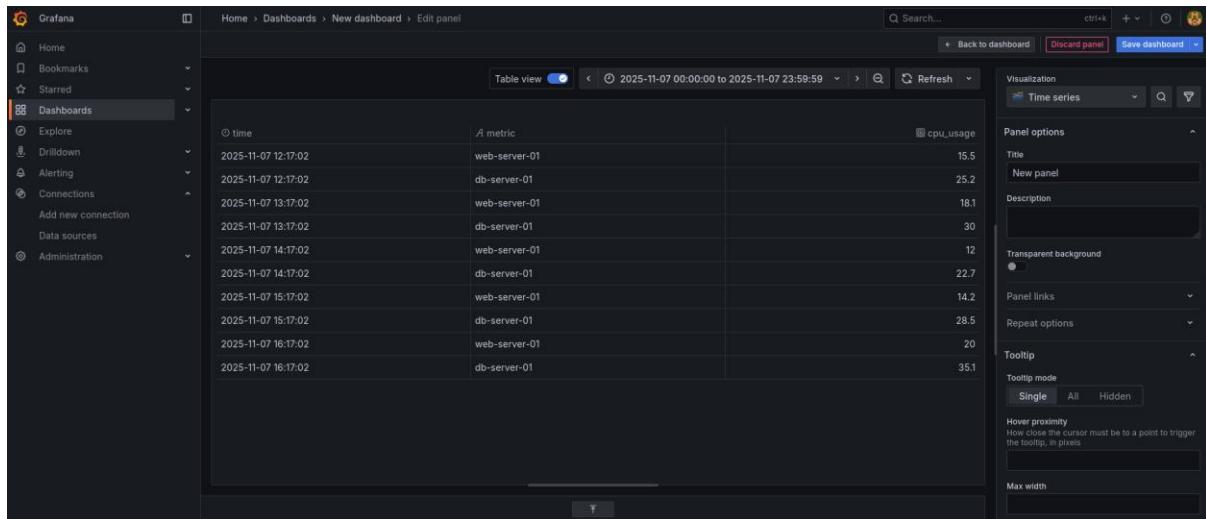
```
server_metrics
```

WHERE

```
$__timeFilter(time_stamp) -- This macro lets Grafana's time picker control the query
```

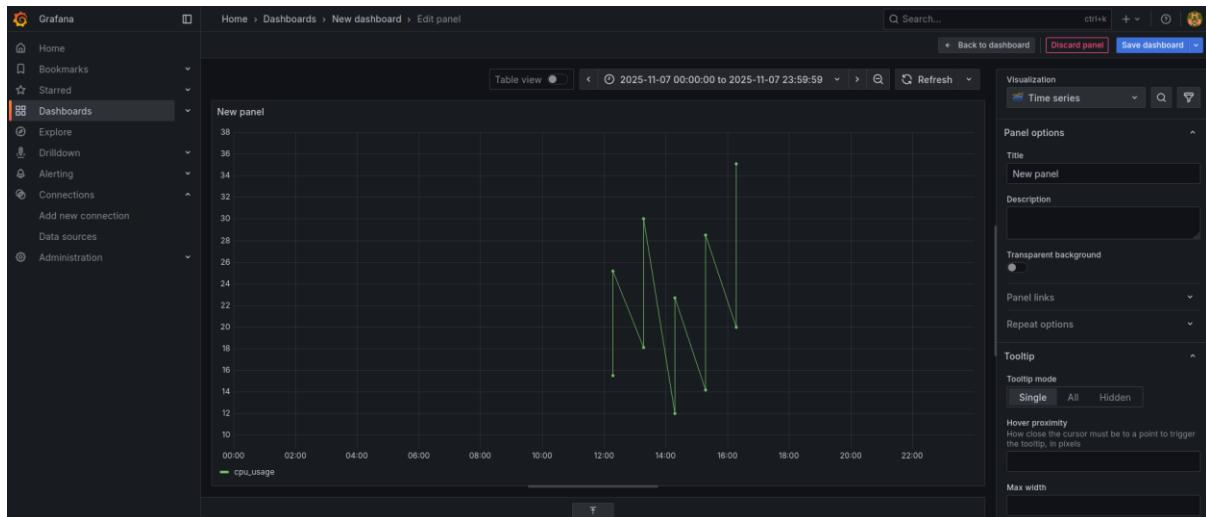
ORDER BY

```
time_stamp ASC;
```



4. Visualize!

- If you see "No data," **change the time range** at the top right of the dashboard from "Last 6 hours" to "**Last 6 hours**" or "**Last 2 days**". This will ensure your sample data is visible.
- On the right-hand panel, you can change the visualization type. It should default to **Time series**, which is perfect.
- You should now see two lines on your graph: one for web-server-01 and one for db-server-01, plotting their CPU usage over time.



5. Save Your Dashboard:

- Click **Apply** in the top right of the panel.
- Click the **Save** (disk) icon at the top of the dashboard.
- Give your dashboard a name (e.g., "MySQL Server Metrics") and click **Save**.

You now have a complete, working Grafana dashboard visualizing data from your MySQL server!