



## SETUP GRAFANA / INFLUXDB / TELEGRAF

SHALA / HEILING / PUNTIGAM  
4AHIT

# Setup ***Grafana / InfluxDB / Telegraf***

*This documentation provides instructions on how to set up a TIG (Telegraf, InfluxDB, Grafana) stack for monitoring system metrics. It includes steps such as cloning a GitHub repository, running docker containers, accessing InfluxDB and Grafana, setting up data sources and creating a dashboard to display RAM, CPU, and disk usage. The guide also includes instructions on how to customize the graph settings and save the dashboard.*

## Contents:

- Prerequisites for Setting up Grafana, InfluxDB, and Telegraf
- Step-by-Step Guide to Setting up the TIG Stack:
  - Cloning the repository
  - Start Docker containers
  - Access InfluxDB webpage
  - Access Grafana webpage
  - Creating a dashboard

## Prerequisites:

Installations:

- Docker
- VS-Code (or any code editor)

## Step-by-Step Guide to Setting up the TIG Stack:

### Docker setup:

- 1) Clone the repository in VS-code: <https://github.com/huntabyte/tig-stack>
- 2) Open the .env file.
  - Look for DOCKER\_INFLUXDB\_INIT\_ADMIN\_TOKEN=changeme.
    - Replace "changeme" with a 32-byte HEX token.
      - For example:  
ef70b0af60a65a0cb9bb66bd0bbc12a42825be9f5a28a  
9813b8e2bb6650c1cb6
- 3) Save the file.
- 4) Make sure you are in the tig-stack repository in the terminal.
  - Execute the command: "docker-compose up -d".
  - Run "docker container ps" to check if the containers are running.

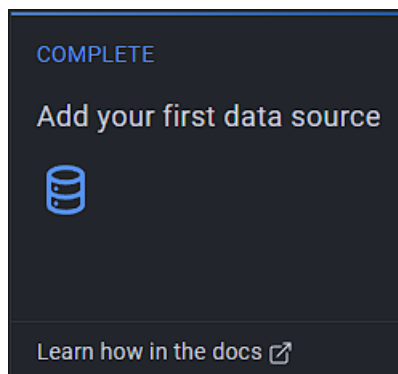
## Webpage setup:

### InfluxDB

1. Access the InfluxDB page with "localhost:8086" as the URL.
  - Log in with the following credentials:
    - Username: changeme
    - Password: changeme

### Grafana:

1. Access the Grafana page with "localhost:3000" as the URL.
  - Log in with the following credentials:
    - Username: admin
    - Password: admin
    - New password: Kennwort1
2. On the landing page, click "Add your first data source".



3. Select InfluxDB.
  - Change the Query Language to Flux.
  - Change the URL to "http://influxdb:8086".
  - Disable Basic auth if enabled.

- Fill in the following fields:
  - Organization: changeme
  - Token: The same token used in the .env file.:  
ef70b0af60a65a0cb9bb66bd0bbc12a42825be9f5a28a  
9813b8e2bb6650c1cb6
  - Default Bucket: changeme

*Should look like this:*

The screenshot shows the Grafana configuration page for InfluxDB. It is divided into three main sections: HTTP, Auth, and InfluxDB Details. The HTTP section includes fields for URL (http://influxdb:8086), Access (Server (default)), Allowed cookies (New tag (enter key to add)), and Timeout (Timeout in seconds). The Auth section includes toggle switches for Basic auth, TLS Client Auth, Skip TLS Verify, and Forward OAuth Identity, along with checkboxes for With Credentials and With CA Cert. The InfluxDB Details section includes fields for Organization (changeme), Token (a masked token), Default Bucket (changeme), Min time interval (10s), and Max series (1000).

HTTP	
URL	http://influxdb:8086
Access	Server (default) <a href="#">Help &gt;</a>
Allowed cookies	New tag (enter key to add)
Timeout	Timeout in seconds

Auth	
Basic auth	<input type="checkbox"/> With Credentials <input type="checkbox"/>
TLS Client Auth	<input type="checkbox"/> With CA Cert <input type="checkbox"/>
Skip TLS Verify	<input type="checkbox"/>
Forward OAuth Identity	<input type="checkbox"/>

InfluxDB Details	
Organization	changeme
Token	.....
Default Bucket	changeme
Min time interval	10s
Max series	1000

- Click "Save and Test".
- If it shows "3 buckets found", everything is correct.

---

***DONE WITH THE SETUP OF A TIG STACK  
(TELEGRAF, INFLUXDB, GRAFANA)***

---

## Step-by-Step Guide to *Monitoring Docker containers with TIG stack*

### Modify Docker:

1. Add the configuration from this repository to the telegraf.conf file:
  - <https://github.com/influxdata/telegraf/blob/master/plugins/inputs/docker/README.md>
2. Add the following line to the volumes section of the docker-compose.yml file:
  - - /var/run/docker.sock:/var/run/docker.sock
3. Grant permissions:
  - Log in with the following credentials:
    - Cd var/run/ • chmod 666 docker.sock

### Dashboard setup:

1. Go to influxDB
  - In the menu bar on the left, click "Explore".
  - At the bottom left of the page, you can see the query builder.
    - If not, on the middle right, you can switch to it
    - Example on Next page →

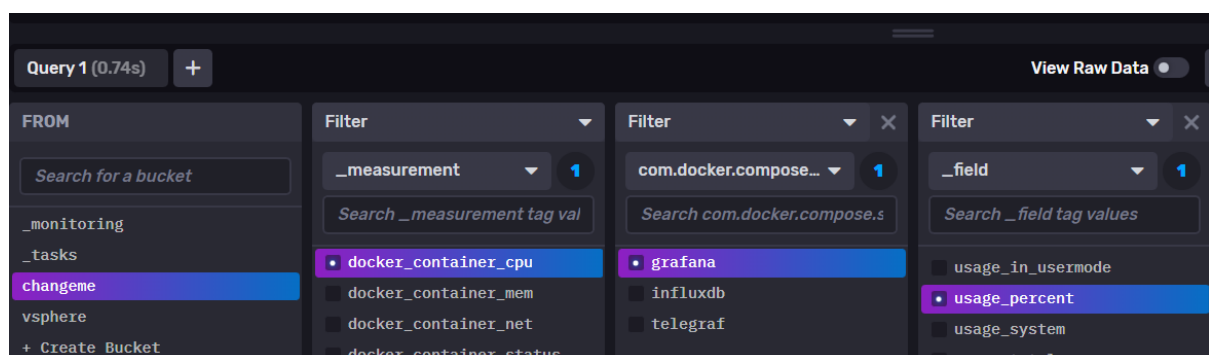
### Example:

get Grafana cpu usage in percent:

1. Go to query builder
2. First box "FROM"
  - Check changeme
3. On the second box "filter"
  - Check docker\_container\_cpu
4. On the third box "filter"
  - click on the dropdown where "\_field"
  - select com.docker.compose.service
- check grafana
5. On the fourth filter
  - Check usage\_percent
6. Now click script editor
  - Copy the script

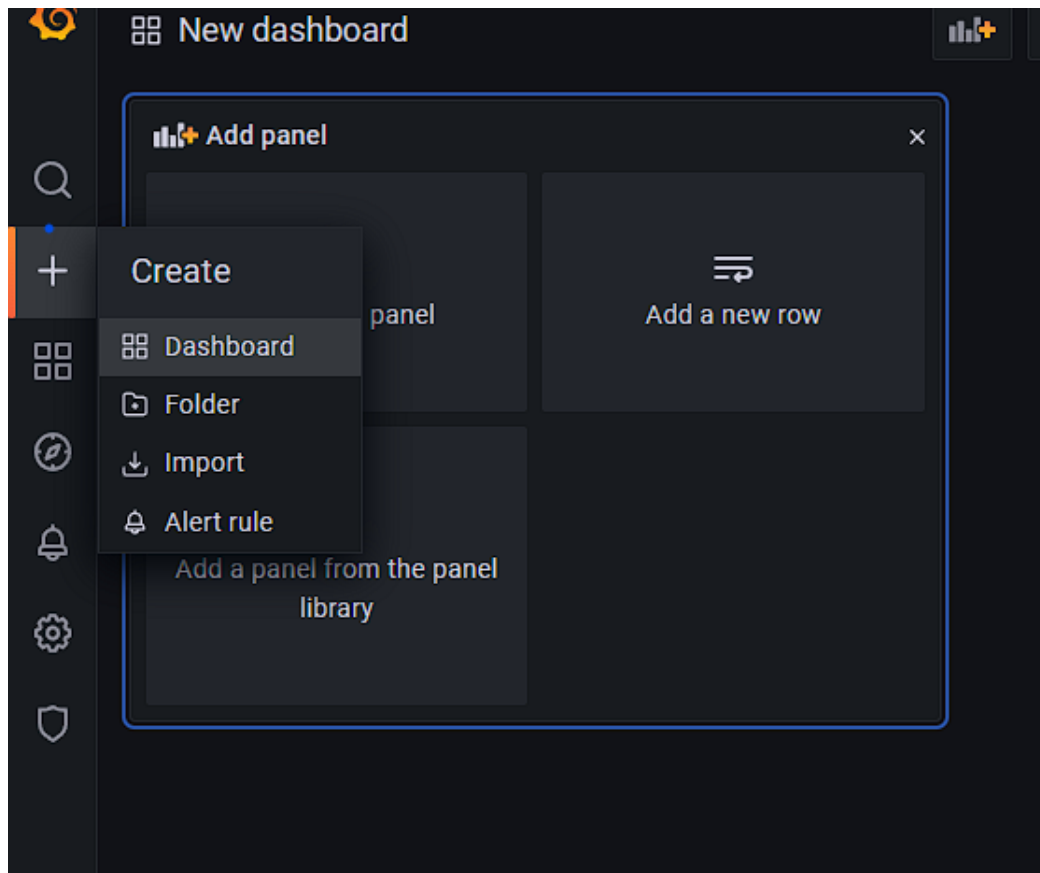
```
Query 1 (0.07s) +
1 from(bucket: "changeme")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "docker_container_cpu")
4   |> filter(fn: (r) => r["com.docker.compose.service"] == "grafana")
5   |> filter(fn: (r) => r["_field"] == "usage_percent")
6   |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
7   |> yield(name: "mean")
```

Query builder:



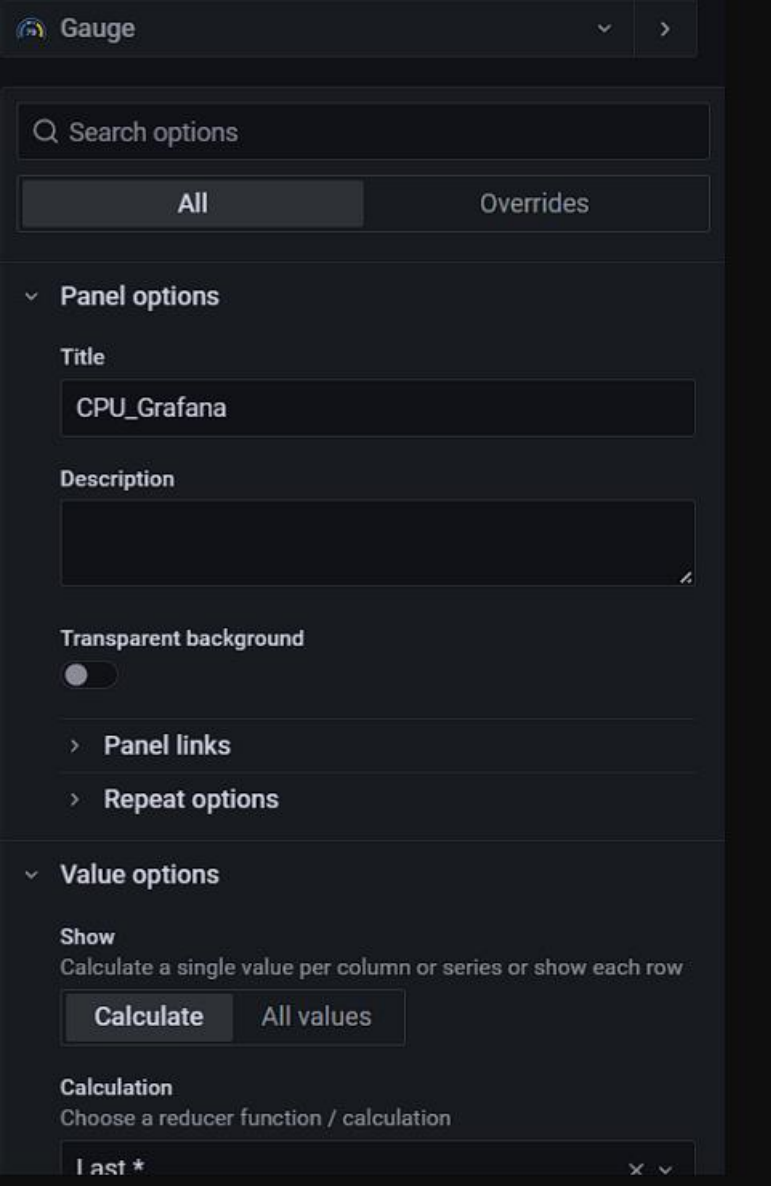
7. Go to Grafana
  - To create a dashboard:
  - Click on the "+" icon on the left.

- Select "New dashboard".
- Add a new panel.



- Paste in the query you got from InfluxDB
- If you want to edit the graph, on the right you can change:
  - title, width, length, unit (e.g. Percent for CPU usage), etc.





The image shows the configuration interface for a Gauge panel in Grafana. At the top, there's a header with the 'Gauge' icon and a dropdown arrow. Below this is a search bar labeled 'Search options'. Two tabs, 'All' and 'Overrides', are visible. The main configuration area is divided into two sections: 'Panel options' and 'Value options'. Under 'Panel options', there's a 'Title' field with the value 'CPU\_Grafana', a 'Description' text area, and a 'Transparent background' toggle switch. Below these are expandable sections for 'Panel links' and 'Repeat options'. The 'Value options' section includes a 'Show' dropdown set to 'Calculate a single value per column or series or show each row', with 'Calculate' and 'All values' buttons. Below that is a 'Calculation' dropdown set to 'Last \*'.

- Once the graph looks good, click "Save" in the upper right corner to save the dashboard and apply

---

***DONE WITH THE SETUP OF MONITORING DOCKER CONTAINERS  
WITH TIG STACK***

---

## Step-by-Step Guide to *Alerting with slack on TIG stack*

### Setup Slack

*Create Slack Webhook:*

<https://www.youtube.com/watch?v=sxtC40gUS2A>

- Here you create a slack Webhook

*Connect Slack-Webhook with Grafana:*

<https://www.youtube.com/watch?v=xgOqLCkQioK>

- Here you connect the slack Webhook to your Grafana

*Add a stress container:*

1. Copy the stressLinux\_CPU-RAM folder from repository into your repository
  - <https://github.com/lukasheiling/TIG.git>
2. In CMD go to the folder location run
  - `docker-compose up -d`
  - and reload your TIG-Stack

This creates a Stress container that helps visualize more CPU & MEM usage.

*Setup alerts in Grafana:*

1. Hover over “+”
  - Folder
  - Enter Folder name
2. Create Dashboard
3. Add new panel
4. On the bottom where the query editor is
  - add query for Stress container CPU usage
    - Example on the next page

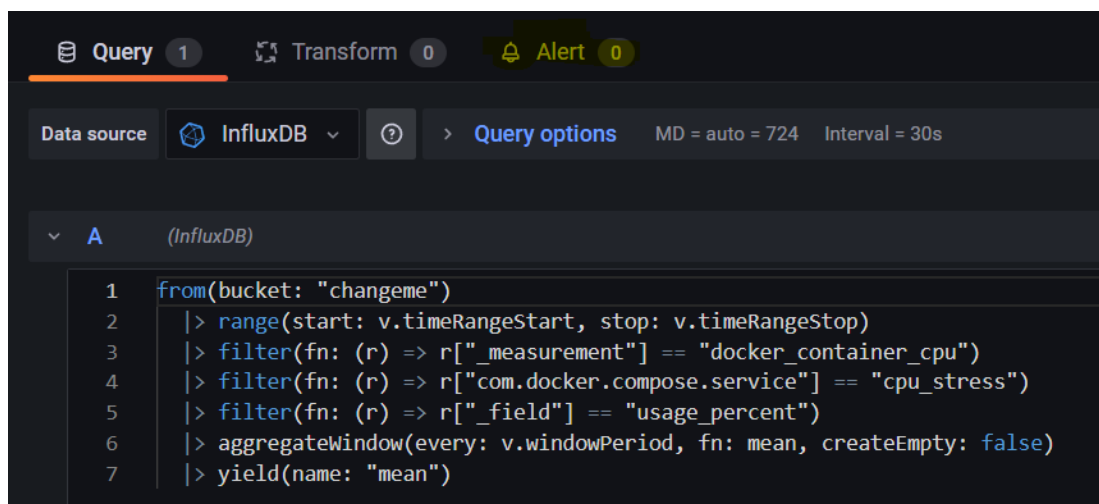
*Example query from InfluxDB for CPU usage of Stress container:*

```
from(bucket: "changeme")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "docker_container_cpu")
  |> filter(fn: (r) => r["com.docker.compose.service"] == "cpu_stress")
  |> filter(fn: (r) => r["_field"] == "usage_percent")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

5. Save & Apply

6. Edit panel

7. Press on Alert



8. Press on Create alert rule from this panel

9. Add two expressions

The screenshot shows the Grafana expression builder interface. It features two expression panels, B and C, each with a dropdown for 'Operation' and a text input for 'Function'. Panel B has 'Reduce' as the operation and a long alphanumeric string as the function. Panel C has 'Math' as the operation and '\$B > 95' as the function. Below the function input is an 'Expression' label and a small icon.

- Define the  $\$B > "95"$  to your desired value when you want to be alerted

10. Define alert conditions:

The screenshot shows the 'Define alert conditions' step in Grafana. It includes a 'Condition' section with a dropdown menu set to 'C'. Below this is an 'Evaluate' section with 'Evaluate every' set to '10s' and 'for' set to '20s'. There is a section for 'Configure no data and error handling' with two dropdown menus, both set to 'Alerting'. A 'Preview alerts' button is at the bottom.

11. When the alert fires you should now receive a slack notification that the alert fired

---

***12. DONE WITH THE GUIDE TO ALERTING WITH SLACK ON TIG  
STACK***

---