

# Proxmox Dash: InfluxDB + Grafana

Dashboard for Proxmox using Proxmox bucket from InfluxDB & Proxmox [Flux] Dashboard from Grafana

- [InfluxDB](#)
  - [Proxmox Connection](#)
- [Grafana Container - Docker](#)
  - [Set Up & Data Collection](#)
- [The Dashboard](#)

# InfluxDB

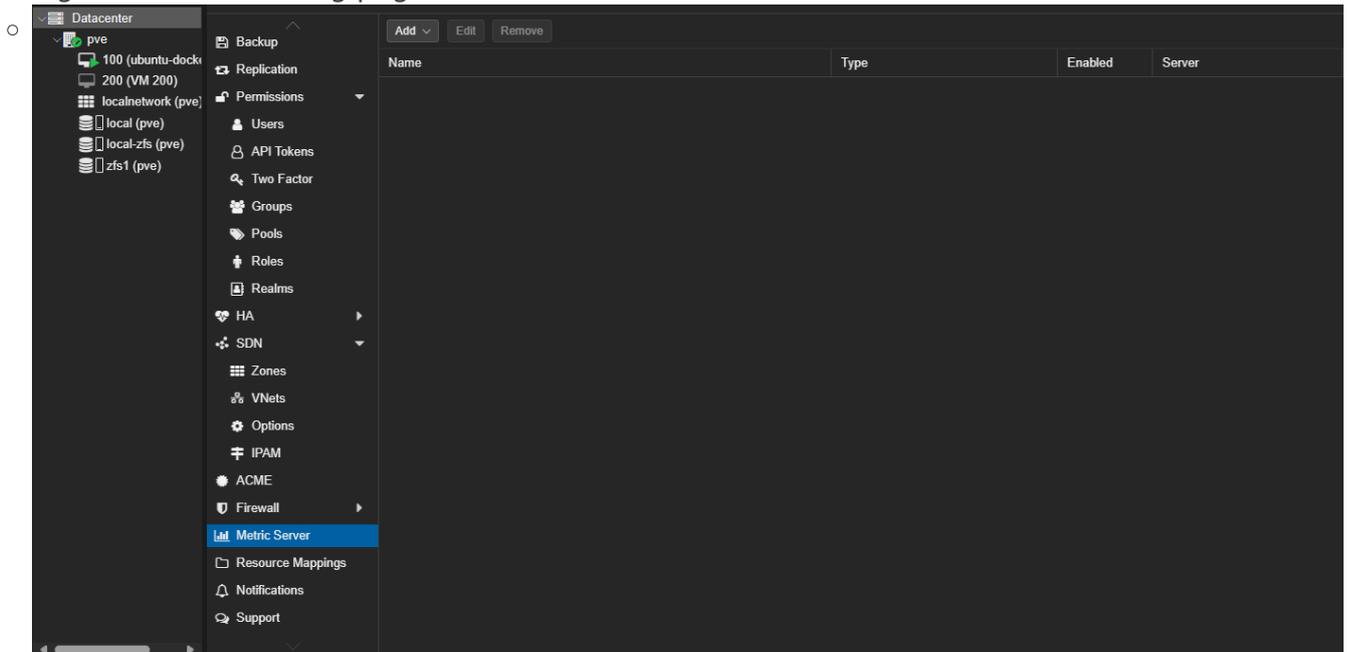
Database that writes and queries data to then be used to populate Grafana Dashboards

Check out the Docker shelf to see how to set up the [InfluxDB container here](#)

# Proxmox Connection

With your InfluxDB container running, you can now configure Proxmox to send data to it via the IP address and Port Number you assigned it.

- Navigate to the following page: Datacenter ---> Metrics Server



- Select Add and enter your InfluxDB information:
  - Create an API Token via the InfluxDB GUI and save it to be entered here:

Name	Server	Port	Protocol	Enabled	Organization	Bucket	Token	API Path Prefix	Batch Size (b)	Timeout (s)	MTU	Verify Certificate
influxdb2	192.168.2.17	8086	HTTP	<input checked="" type="checkbox"/>	homeLAB	proxmox	unchanged		25000000	1	1500	<input checked="" type="checkbox"/>

- Click ok, and navigate back to your InfluxDB GUI. To confirm it's worked, navigate to the Data Explorer tab. Select your bucket, and you should start to see categories of Proxmox data:

# Data Explorer

Simple Table CUSTOMIZE UTC

table	_measurement	_field	_value	_time	host	nodename	object	vm1
last	group string	group string	no group double	no group dateTime:RFC3339	group string	group string	group string	group string
0	system	cpu	0	2024-03-23T23:29:24.000Z	VM 200	pve	qemu	200
0	system	cpu	0	2024-03-23T23:29:34.000Z	VM 200	pve	qemu	200
0	system	cpu	0	2024-03-23T23:29:44.000Z	VM 200	pve	qemu	200

Query1 (0.07s) View Raw Data Past 1m SCRIPT EDITOR

FROM: Search buckets

homeLAB  
OPNsense  
proxmox  
\_monitoring  
\_tasks  
+ Create Bucket

Filter: \_measurement 3  
Search \_measurement tag values

- ballooninfo
- blockstat
- cpustat
- memory
- nics
- proxmox-support
- system

Filter: \_field 2  
Search \_field tag values

- cpu
- cpus
- ctime
- disk
- diskread
- diskwrite
- enabled
- freemem
- guest

Filter: host 3  
Search host tag values

- VM 200
- pve
- ubuntu-docker-engine

Filter: nodename  
Search nodename tag values

- pve

WINDOW PERIOD: CUSTOM auto (fs) Fill missing values

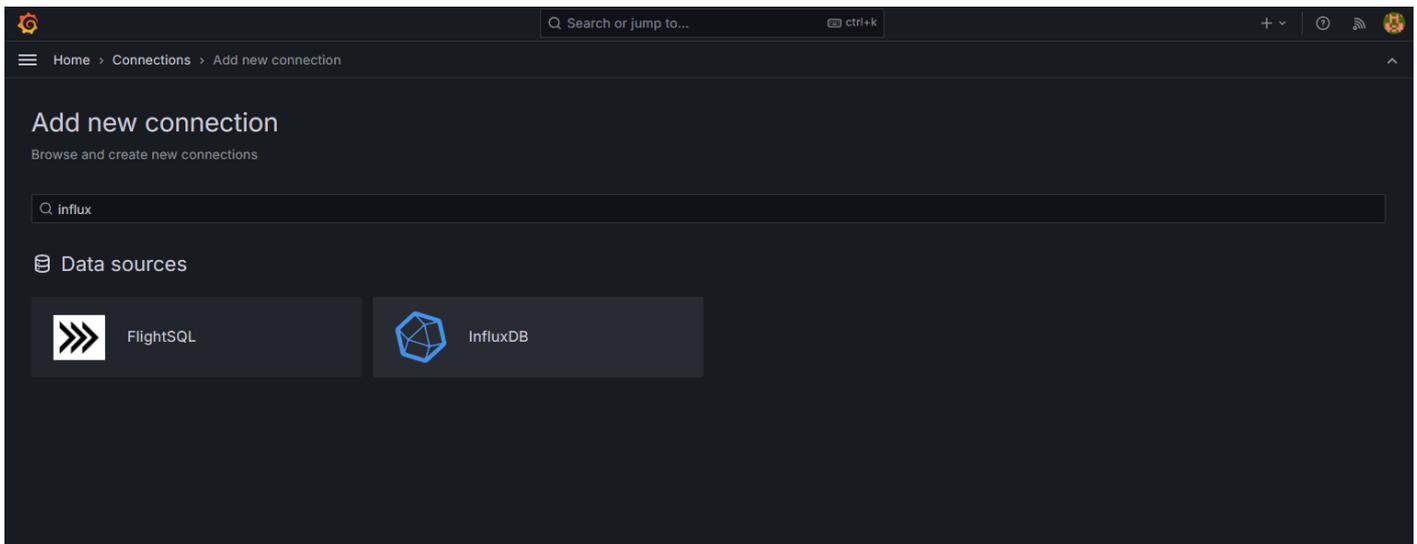
AGGREGATE FUNCTION: CUSTOM mean median **last**

o Your InfluxDB is now actively collecting data!

# Grafana Container - Docker

# Set Up & Data Collection

Deploy your container, the log into the Web GUI. The navigate to Home ---> Connections ---> InfluxDB:



Select "Add new Data Source" then follow the below steps to establish connection to your InfluxDB:

- Change Query Language from Influxdb to Flux
- Enter <http://localhost:8086> as the URL
  - Make sure your containers are running on the same network. You can verify this via the Portainer Web GUI
- Enter your Influxdb details, and then save&test



If your connection is successful, you'll see the following message display:

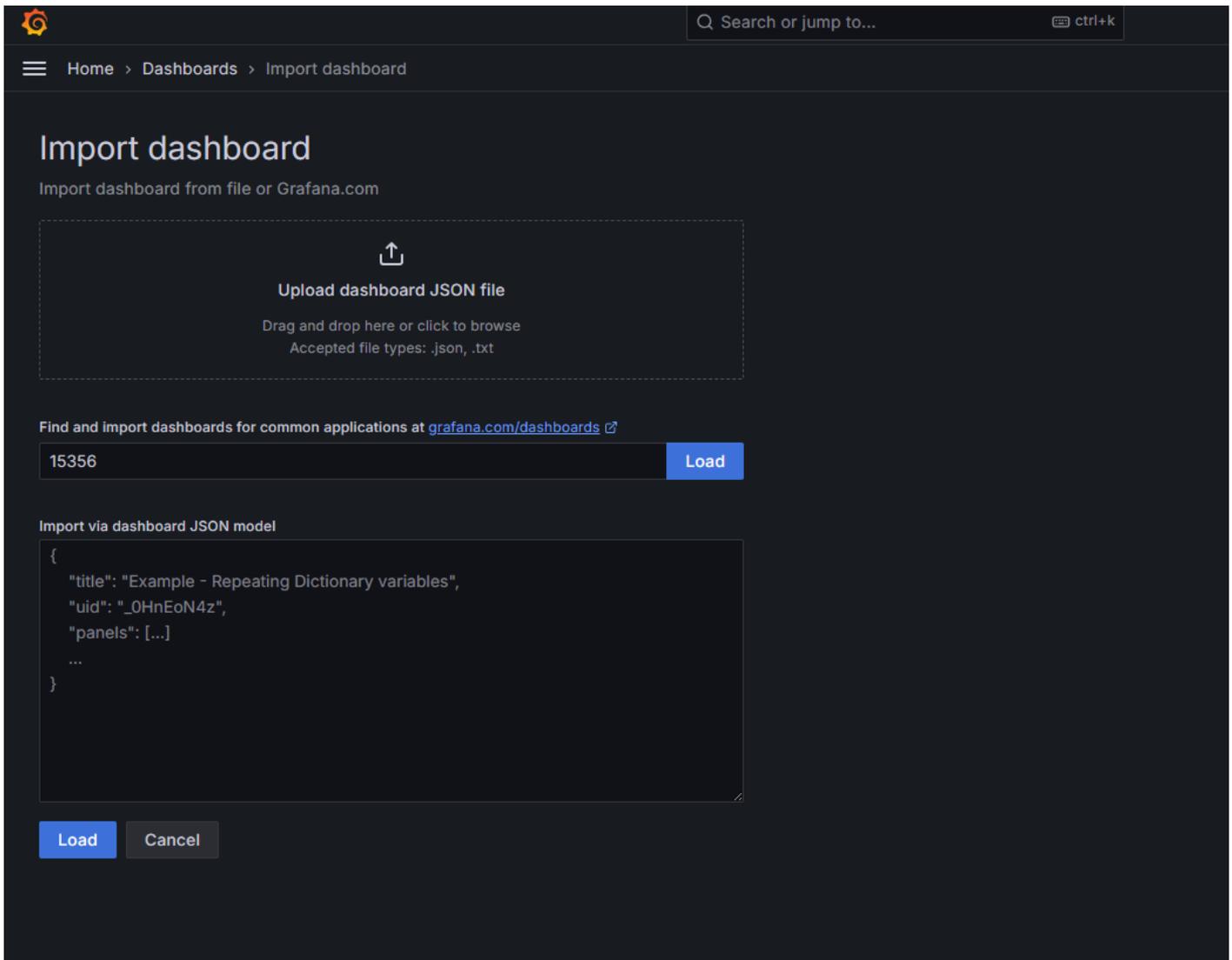
✓ datasource is working, 4 buckets found

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

# The Dashboard

With your connections set up, you can build your dashboard. There are many templates on Grafana to choose from, or you can create your own. I chose to use the Proxmox Cluster [Flux] dashboard. See below:

Navigate to Home ---> Dashboards ---> Import Dashboards in the Grafana Web GUI:



The screenshot shows the Grafana web interface for importing a dashboard. At the top, there is a search bar and a navigation breadcrumb: Home > Dashboards > Import dashboard. The main heading is "Import dashboard" with a subtitle "Import dashboard from file or Grafana.com". Below this is a dashed box containing an upload icon and the text "Upload dashboard JSON file", "Drag and drop here or click to browse", and "Accepted file types: .json, .txt". Underneath, there is a link to "grafana.com/dashboards" and a search input field containing "15356" with a "Load" button. The bottom section is titled "Import via dashboard JSON model" and features a text area with a JSON snippet: 

```
{
  "title": "Example - Repeating Dictionary variables",
  "uid": "_0HnEoN4z",
  "panels": [...]
  ...
}
```

 At the bottom of this section are "Load" and "Cancel" buttons.

Here, import the Grafana template ID, then select load, and adjust name your Dashboard, then click Import:

# Options

Name

Proxmox [Flux]

Folder

Dashboards

Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used to uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

lfgdXjtnk

Change uid

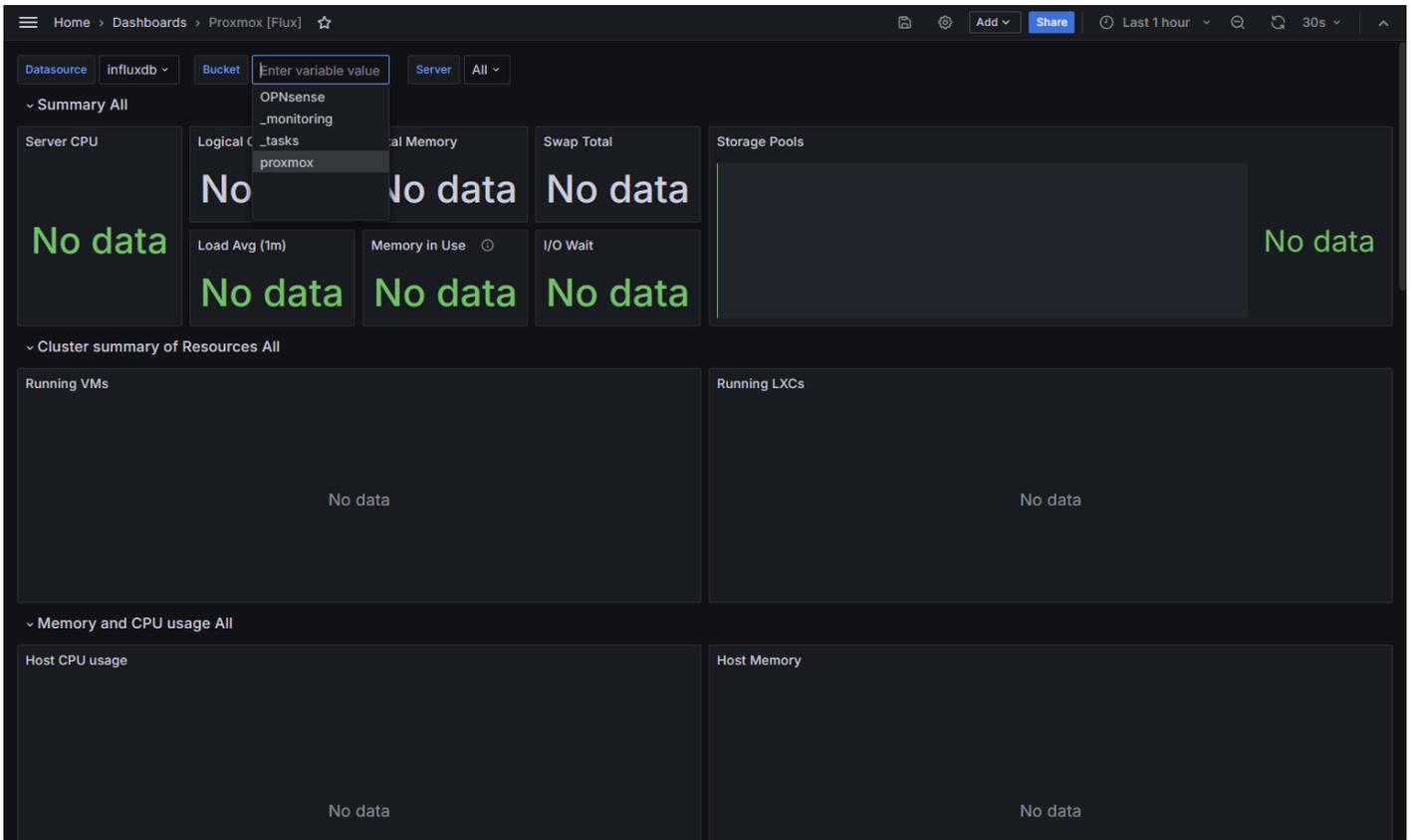
proxmox-flux

 influxdb

Import

Cancel

Once loaded, you'll need to select your InfluxDB bucket:



The screenshot shows the Grafana interface for the 'Proxmox [Flux]' dashboard. The top navigation bar includes 'Home > Dashboards > Proxmox [Flux]'. The main content area is divided into several sections, all of which display 'No data' in green text. A dropdown menu is open over the 'Bucket' field, showing options: 'OPNsense', '\_monitoring', and 'proxmox'. The sections visible are:

- Summary All:** Server CPU, Logical C...tasks, al Memory, Swap Total, Storage Pools, Load Avg (1m), Memory in Use, I/O Wait.
- Cluster summary of Resources All:** Running VMs, Running LXCs.
- Memory and CPU usage All:** Host CPU usage, Host Memory.

Your dashboard is all set!

