

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

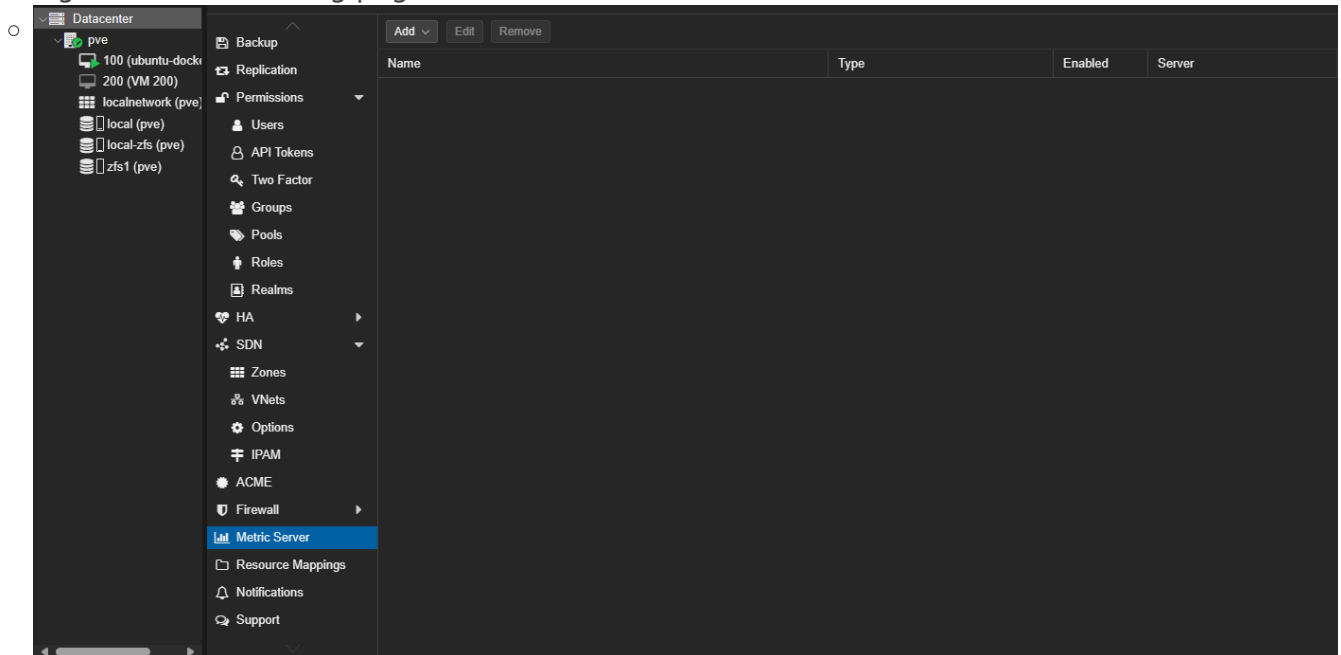
Database that writes and queries data to then be used to populate Grafana Dashboards

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:

A screenshot of the 'Edit: InfluxDB: influxdb2' configuration window in Proxmox. The window contains the following fields:

- Name: influxdb2
- Server: 192.168.2.17
- Port: 8086
- Protocol: HTTP
- Enabled: ☒
- Organization: homeLAB
- Bucket: proxmox
- Token: unchanged
- API Path Prefix: (empty)
- Batch Size (b): 25000000
- Timeout (s): 1
- MTU: 1500
- Verify Certificate: ☒

At the bottom, there are buttons for 'Help', 'Advanced' (checked), 'OK', and 'Reset'.

- 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

Query 1 (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

1

Search nodename tag values

☒ pve

WINDOW PERIOD

CUSTOM

auto (1s)

Fill missing values

AGGREGATE FUNCTION

CUSTOM

mean

median

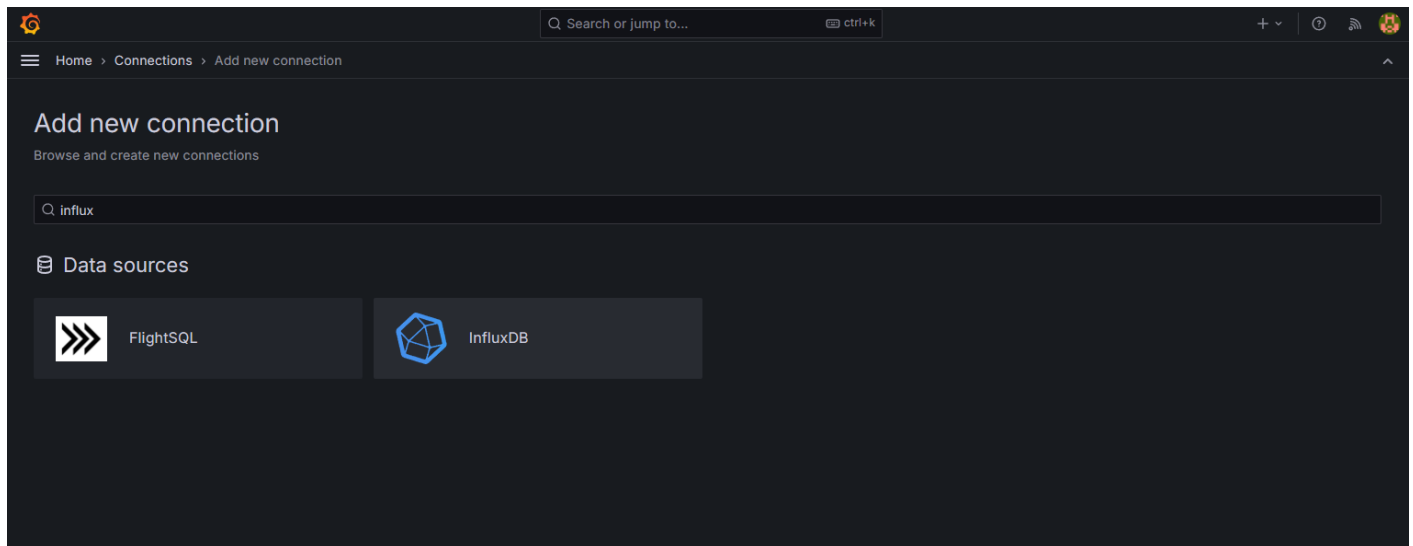
last

○ 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

A screenshot of the 'InfluxDB Details' configuration form in Grafana. The form has a dark theme. It contains five rows of input fields:

- Organization:** A text field containing 'homeLAB'.
- Token:** A text field filled with dots, indicating a masked password or token.
- Default Bucket:** A text field containing 'proxmox'.
- Min time interval:** A text field containing '10s', with a help icon (i) to its left.
- Max series:** A text field containing '1000', with a help icon (i) to its left.

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

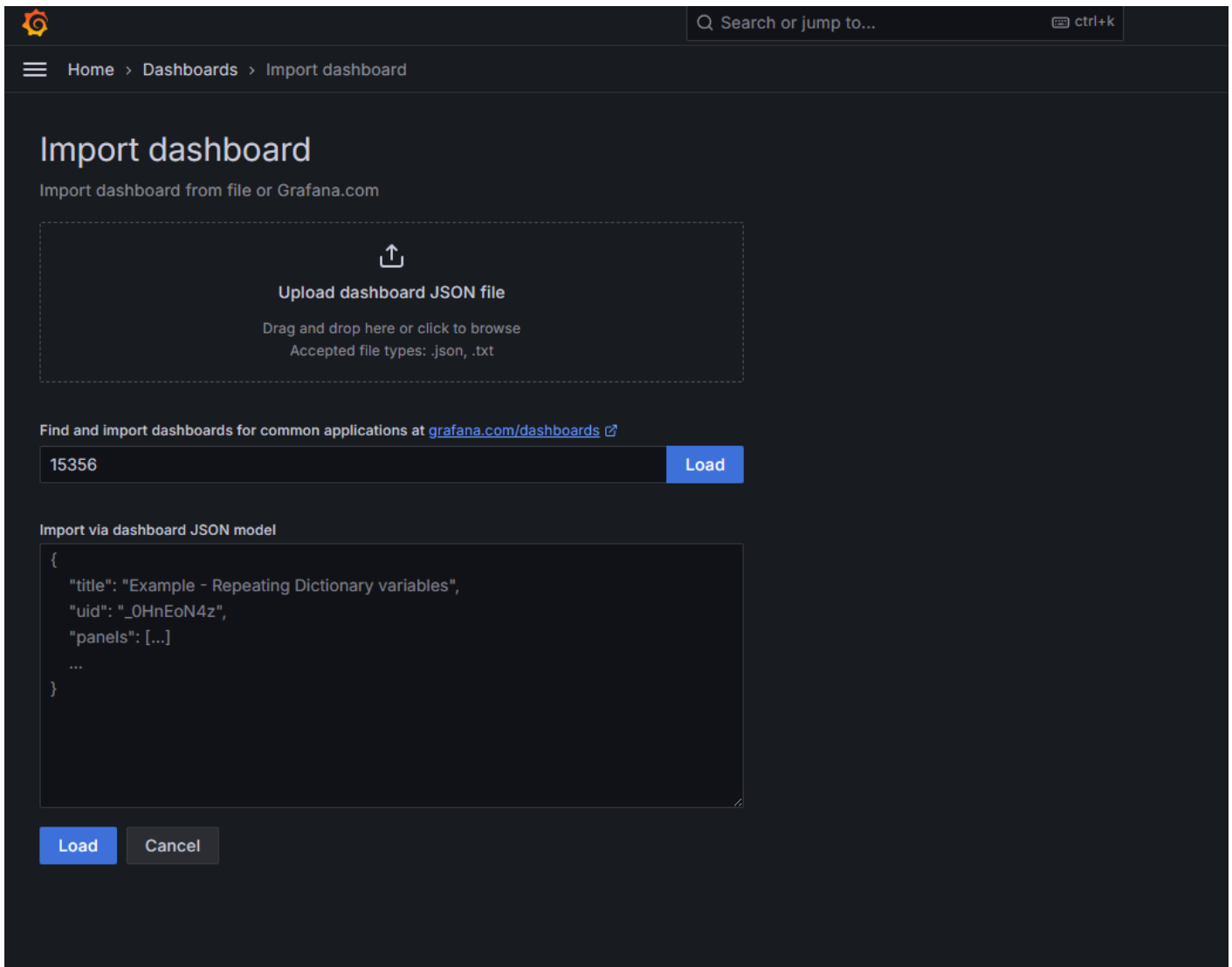
✓ datasource is working. 4 buckets found

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 'Import dashboard' page. 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, there is a large dashed box with 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, a link says 'Find and import dashboards for common applications at grafana.com/dashboards'. Below the link is a text input field containing '15356' and a blue 'Load' button. Further down, there is a section titled 'Import via dashboard JSON model' with a text area containing a JSON snippet: { "title": "Example - Repeating Dictionary variables", "uid": "_OHnEoN4z", "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 for 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.

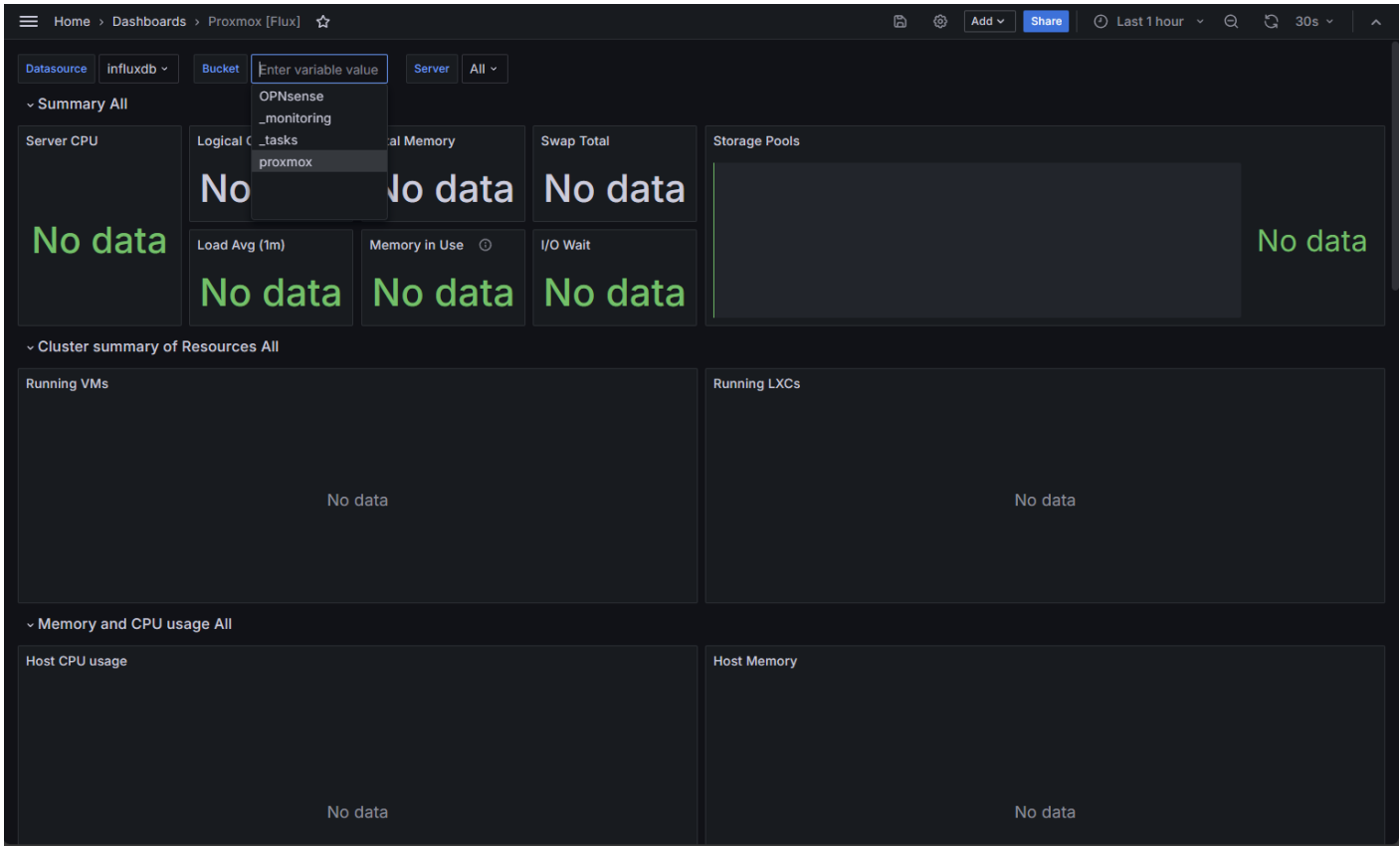
lfgdXjtnk [Change uid](#)

proxmox-flux

 influxdb

[Import](#) [Cancel](#)

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



Home > Dashboards > Proxmox [Flux]

Datasource influxdb Bucket Enter variable value Server All

Summary All

Server CPU No data

Logical C...tasks No data

al Memory No data

Swap Total No data

Storage Pools No data

Load Avg (1m) No data

Memory in Use No data

I/O Wait No data

Cluster summary of Resources All

Running VMs No data

Running LXCs No data

Memory and CPU usage All

Host CPU usage No data

Host Memory No data

Your dashboard is all set!

