

TAUBER

Grafana Integration for BMC BHOM

User Guide

Version 3.x.xx
November 2025



Contacting TAUBER Business Solutions Software

You can access TAUBER Business Solutions Software website at <http://www.tabuso.com> where you can also obtain information about the company, its products and services.

Address

TAUBER Business Solutions Software

Rheinstrasse 35

53332 Bornheim

Germany

Phone: +49 2222 991869

Mail: info@tabuso.com

Web: www.tabuso.com

Support: support@tabuso.com

© Copyright 2008 till 2025 TAUBER Business Solutions UG as an unpublished work. All rights reserved

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc, all other trademarks belong to their respective companies

TAUBER Business Solutions UG considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User Agreement for product and the proprietary and restriction rights included in this document.

Index

Index **3**

Version History..... **5**

Introduction **6**

 How does the Integration work?..... 7

 Supported BMC Helix Operations Manager and Grafana versions 8

Install **9**

 Installing the TAUBER PlugIn Proxy on Windows 9

 Prerequisites 9

 Installation 10

 Installing the TAUBER PlugIn Proxy on Linux..... 10

 Prerequisites 10

 Installation 10

 Make Proxy autostart on boot..... 11

 Installing the TAUBER Grafana Datastore PlugIn..... 12

 Prerequisites 12

 Installation 12

 Installing the license..... 13

 Files and Operating Systems 14

Upgrade **15**

 Upgrading the TAUBER PlugIn Proxy on Windows 15

 Upgrading the TAUBER PlugIn Proxy on Linux..... 16

 Upgrading the TAUBER Grafana PlugIn 17

Uninstall..... **18**

 Uninstalling the TAUBER PlugIn Proxy on Windows..... 18

 Uninstalling the TAUBER PlugIn Proxy on Linux 18

 Uninstalling the TAUBER Grafana Datastore PlugIn 18

Configuration **19**

 Configuring the TAUBER PlugIn Proxy 19

 Configuring the TAUBER Grafana Data Source PlugIn 23

 Using https for the PlugIn Proxy and the PlugIn..... 26

How to use the TAUBER Grafana PlugIn..... **30**

 Introduction..... 30

 Static Dashboards and Panels 31

 Dynamic Metric Query 32

Attr. Alias	33
Last point?	33
Metric Query	34
Last point?	35
Event Query.....	36
Advanced Event Query	38
Device List Panel Query	41
Dynamic Dashboards and Panels	42
Variables and Queries	44
The deviceName variable	47
The agentHostName variable	53
The monitorTypeName variable	59
The monitorInstanceName variable	65
The attributeName and attributeName2 variable	72
The templateVariable1 to 100 variables.....	79
Chained variables.....	82
The Asterix(*) for Monitor instance names	83
Creating dynamic Dashboards with the “Dynamic Metric Query”	84
Troubleshooting	94
The PlugIn cannot connect to Proxy	94
The Proxy cannot connect to BHOM	96
No Metric or Event data in the Panel	97
Known problems	99

Version History

V1.0.00

- Initial version

V1.0.11

- Smaller changes and bugfixes

V1.0.20

- Fixing a bug preventing to create an Advanced Event Dashboard with Cross Launch L
- Link using the Default Slotlist

V2.0.00

- New “Dynamic Metric Query” for easy creating static and dynamic Metric Dashboards.

V2.0.10

- Bug fixed in Advanced Event Query

V3.0.20

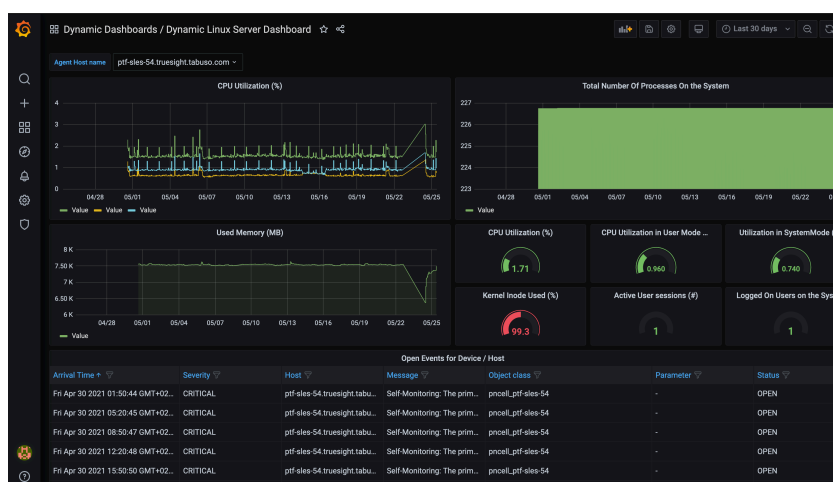
- Backend Code now in Golang
- Support for Grafana 12.x

V3.0.24

- Fixed a bug in Advanced Event Crosslaunch function
- Enabled Template Variables in Advanced Event Query
- Bugfixes

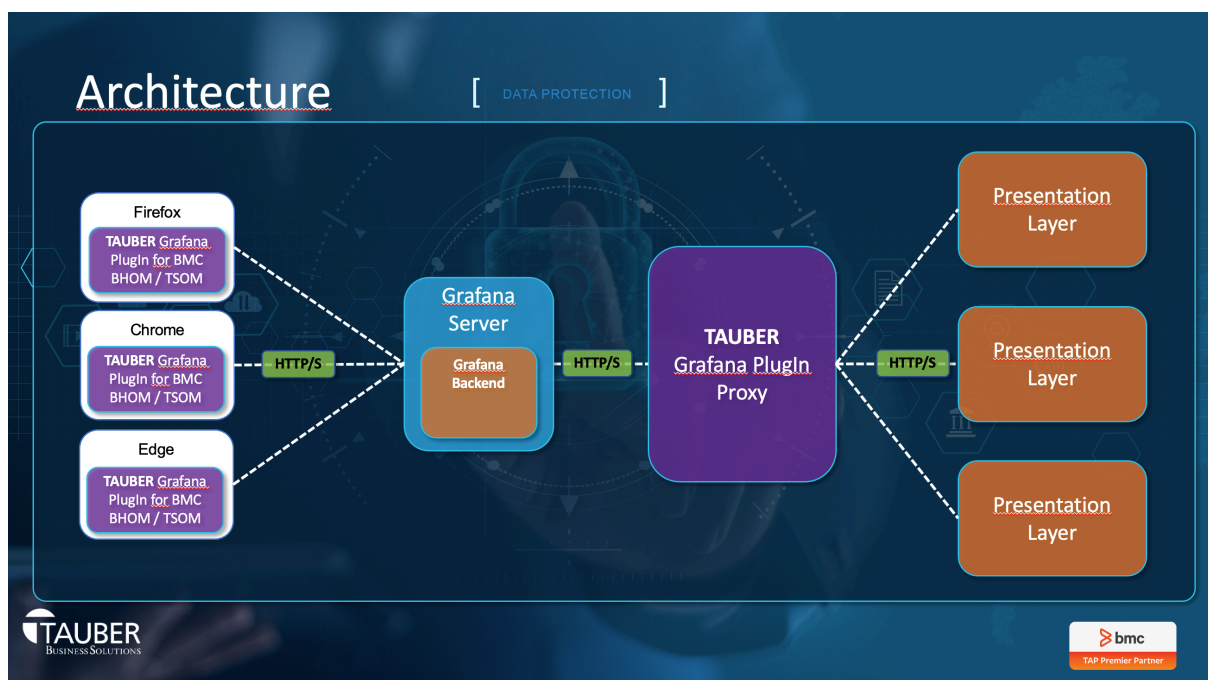
Introduction

Showing Availability, Performance and Business data from different Data Sources in Dashboards is a common Use case at most customers. With TAUBER Grafana Integration for BMC Helix Operations Manager, you can quickly connect your Grafana to BMC Helix Operations Manager SAS and OnPrem installations and create Dashboards the way you want with Metric and Event data from your BMC Helix Operations Manager.



Dynamic Grafana Dashboard with Metric and Event data

The TAUBER Grafana Integration for BMC Helix Operations Manager connects Grafana to the Presentation Layer REST-API and enables you to create static and / or dynamic Dashboards based on BHOM Metric and Event Data. The connection is established using a lightweight proxy that can be deployed in a few minutes. The Proxy is used to gather and prepare the data from the Presentation Layer and forward it to the Grafana Plugin.



TAUBER Grafana Integration Architecture

How does the Integration work?

The integration is based on two components. A lightweight Node based Windows or Linux Program called PlugIn Proxy and the Grafana Data Source PlugIn itself.

The Data Source PlugIn will be configured to connect to the REST-API of the BMC Helix Operations Manager through the PlugIn Proxy when Data or Events are required. The PlugIn Proxy will connect to the Presentation Layer configured in the PlugIn, store and if required check the Authorization Token and finally send a request to the REST-API of the Presentation Layer of BHOM to collect the Data and / or Events.

Based on the required Data the Proxy will process the result of the Presentation Layer, prepare it for the PlugIn and finally return the result to the Data Source PlugIn and use Grafana to display it.

The PlugIn Proxy has several functions like connecting to the Presentation Layer, storing the Authorization Tokens to reduce Logon / Logoff requests, and validate the tokens. Next to this the PlugIn Proxy creates a Cache that translates Device, Monitor Type, Monitor Instance and Attribute names to BHOM internal Ids.

Supported BMC Helix Operations Manager and Grafana versions

The following table shows the supported Helix Operations Manager and Grafana versions:

BMC Helix Operations Manager

Helix Operations Manager Versions	Supported?
22.x	Yes
23.x	Yes
24.x	Yes
25.x	Yes

TAUBER Grafana PlugIn

Grafana Versions	Supported?
7.4.x	Yes
8.x.x	Yes
9.x.x	Yes
10.x.x	Yes
11.x.x	Yes
12.x.x	Yes

TAUBER Grafana PlugIn Proxy

Operating System	Supported?
Windows 2016 Server	Yes
Windows 2019 Server	Yes
Windows 2022 Server	Yes
Windows 2025 Server	Yes
Suse Enterprise Server 15.x	Yes
Oracle Linux 7.x, 8.x	Yes
RHEL 8.x	Yes

Install

The following chapter describes how to deploy, install, and run the TAUBER Grafana PlugIn Proxy and the TAUBER Grafana PlugIn. We recommend creating a separate Windows or Linux VM for the PlugIn Proxy. Use the following sizing.

Test / Development Systems

2 CPUs / 8 GB RAM / 100 GB Disk space

Production Systems

4 CPUs / 16 GB RAM / 100 GB Disk space

The PlugIn Proxy is the component "between" the TAUBER Grafana PlugIn and the BMC Helix Operations Manager Presentation Layer. The Proxy receives request from the PlugIn to talk to the BHOM Presentation Layer to gather and process the data and sends it back to the PlugIn.

To install the PlugIn Proxy, download the actual product file TAUBER_Grafana_Integration_for_BMC_BHOM_VX.X.XX.zip from the TAUBER Business Solutions Website.

The file contains the documentation, a ZIP file called TAUBERGrafanaPlugInProxy_for_BMC_BHOM_VX.X.XX.zip for the PlugIn Proxy and a ZIP file called TAUBERGrafanaPlugIn_for_BMC_BHOM_VX.X.XX.zip for the Grafana Data Source PlugIn.

Installing the TAUBER PlugIn Proxy on Windows

Prerequisites

- Make sure the Windows user you want to use to deploy the Proxy has full access to the installation directory of the Proxy.
- Make sure the Windows user has Administrator privileges.
- Add the user to the local Administrators group.

Installation

- Login to the Server you want to deploy the PlugIn Proxy on .
- Create the target parent directory e.g., D:\BMC
- Copy the TAUBERGrafanaPlugInProxy_for_BMC_BHOM_VX.X.XX.zip file to the target directory and unzip it.
- Run a Command Shell as Administrator (elevated!).
- cd to the unzipped directory bhPlugInProxy.
- Run the command “bhPlugInProxy.bat -add “ to add the Windows Service for the PlugIn Proxy.
- Start the Windows Service “TAUBER PlugIn Proxy for BMC BHOM”.

Note:

To see communication is working you can also run the “bhPlugInProxy.exe” from an elevated Command Window.

Installing the TAUBER PlugIn Proxy on Linux

Prerequisites

- Make sure the Linux user you want to use to deploy the Proxy has full access to the installation directory of the Proxy. In this installation we use a user called “bhom” who’s part of the “users” group.

Installation

- Login to the Server you want to deploy the PlugIn Proxy on as root.
- Create the target parent directory e.g., D:/opt/bmc
- Execute “chown bhom:users /opt/bmc”.
- Logout and back in as the user owning the Proxy, e.g “bhom”.
- Copy the TAUBERGrafanaPlugInProxy_for_BMC_BHOM_VX.X.XX.zip file to the target directory and unzip it.
- cd into the installation directory “bhPlugInProxy” and execute
 - chmod 755 bhPlugInProxy
 - chmod 755 modules/rest-client/bh-rest-client

- chmod 755 modules/rest-client/curl/curl
- Test the Proxy by running “./bhPlugInProxy”.

IMPORTANT

When installing the PlugIn Proxy on RedHat Enterprise Linux please check if the REST Client is working properly.

- Go to the subdirectory modules/rest-client.
- Execute the REST Client using “./bh-rest-client”
If you get an error like “.. missing libnsl.so.1 ...”.
- Manually install the libraries using
“dnf install libnsl*”.
- Execute the REST client again and make sure the error is gone.

Make Proxy autostart on boot

The following process will describe how to add the PlugIn Proxy to the systemctl daemon to be able to start the Proxy automatically on boot.

- Login to the Server you installed the PlugIn Proxy on as root.
- Create a service unit file using
“vi /etc/systemd/system/bhPlugInProxy.service”.
- Put the following content in the file and adjust the paths.

```
[Unit]
Description=TAUBER Grafana PlugInProxy for BMC BHOM
```

```
Wants=network.target
After=syslog.target network-online.target
```

```
[Service]
Type=simple
WorkingDirectory=/opt/tabuso/bhPlugInProxy
ExecStart=/opt/tabuso/bhPlugInProxy/bhPlugInProxy
Restart=on-failure
RestartSec=10
KillMode=process
```

```
[Install]
WantedBy=multi-user.target
```

- Run “systemctl daemon-reload” to reload the config.
- Enable autostart using “systemctl enable bhPlugInProxy.service”.
- Start the Proxy using “systemctl start bhPlugInProxy.service”.

You can start, stop and check the service using

```
systemctl [stop | start | status] bhPluginProxy.service
```

Installing the TAUBER Grafana Datastore Plugin

The TAUBER Grafana Datastore Plugin is the component installed and running in Grafana and connecting “through” the Plugin Proxy to the BMC Helix Operations Manager Presentation Layer.

Prerequisites

- Make sure you run a supported Grafana version.

Installation

- Login to your Grafana Server.
- Unzip the file TAUBERGrafanaPlugin_for_BMC_BHOM_VX.X.XX.zip to the grafana\data\plugins folder.
- Restart the Grafana Server by restarting the Windows Service or the Linux Grafana Process.

IMPORTANT

If you do not find the `bmc-datasource-plugin` in the list of Plugins please add the plugin to your Grafana “.ini” file as follows:

`allow_loading_unsigned_plugins` property:

```
allow_loading_unsigned_plugins = tabuso-bmcbhom-datasource-plugin
```



Installing the license

If you have a license provided as license.txt file, please copy the file to the following directory on the PlugIn Proxy:

Windows: [Installation directory]\modules\rest-client

Linux: [Installation directory]/modules/rest-client

The license is immediately active without restarting the Proxy!

Files and Operating Systems

The following table shows all files on the PlugIn Proxy ZIP file, which Operating Systems they are used for and a description.

Folder + Filename	OS	Description
bhPlugInProxy	Linux	PlugInProxy for Linux.
bhPlugInProxy.bat	Windows	Batch to add / remove Windows Service.
bhPlugInProxy.exe	Windows	PlugInProxy for Windows.
conf/setup.properties	Linux & Windows	Configuration file for the Proxy.
conf/secure	Linux & Windows	Folder for https certificates.
data/cache/[Systemname]/devices	Linux & Windows	Cache directory for devices.
data/cache/[Systemname]/groups	Linux & Windows	Cache directory for groups.
log/	Linux & Windows	Folder for logfiles.
modules/nssm/nssm.conf	Windows	Config file for Windows Service Management tool.
modules/nssm/nssm.exe	Windows	Executable of Windows Service Management tool.
modules/rest-client/config	Linux & Windows	Folder for Connection files of the REST Client.
modules/rest-client/curl/curl	Linux	Curl executable for Linux.
modules/rest-client/curl/curl.exe	Windows	Curl executable for Windows.
modules/rest-client/curl/ca-bundle.crt	Linux & Windows	Curl Root Certificates.
modules/rest-client/data	Linux & Windows	Folder for temporary files for the REST-Client.
modules/rest-client/status	Linux & Windows	Folder for Auth Tokens.
modules/rest-client/bh-rest-client	Linux	REST Client for Linux.
modules/rest-client/bh-rest-client.exe	Windows	REST Client for Windows.

Upgrade

The following chapter describes how to upgrade an existing installation of the TAUBER Grafana PlugIn Proxy and the TAUBER Grafana PlugIn.

Upgrading the TAUBER PlugIn Proxy on Windows

To upgrade an existing installation of the TAUBER Grafana PlugIn Proxy.

- Login to the Server you want to upgrade the PlugIn Proxy on.
- Stop the Windows Service “TAUBER Grafana PlugIn Proxy for BMC BHOM”.
- Rename your setup.properties configuration file in the \conf subfolder to setup.properties.old.
- Backup your license file modules\rest-client\license.txt.
- Copy the new TAUBERGrafanaPlugInProxy_for_BMC_BHOM_VX.X.XX.zip file to the target directory and unzip it.
- Copy the properties you have customized from the setup.properties.old files to the new setup.properties.
- Override the Trial license with your backed up production license.
- Delete the setup.properties.old file.
- Start the Windows Service “TAUBER Grafana PlugIn Proxy for BHOM”

Upgrading the TAUBER PlugIn Proxy on Linux

To upgrade an existing installation of the TAUBER Grafana PlugIn Proxy on Linux

- Login to the Server want to upgrade the PlugIn Proxy on as root.
- Stop the Service using “systemctl stop bhPlugInProxy.service”.
- Logout and log back into the Server as the user owning the installation, e.g. “bhom”.
- Rename your setup.properties configuration file in the /conf subfolder to setup.properties.old.
- Backup your license file modules/rest-client/license.txt.
- Copy the new TAUBERGrafanaPlugInProxy_for_BMC_BHOM_VX.X.XX.zip file to the target directory and unzip it.
- Copy the properties you have customized from the setup.properties.old files to the new setup.properties.
- Override the Trial license with your backed up production license.
- Delete the setup.properties.old file.
- Start the Service using “systemctl start bhPlugInProxy”

Upgrading the TAUBER Grafana PlugIn

To upgrade the TAUBER Grafana PlugIn, unzip the new product file to the same location in your Grafana directory as you did with the actual version. Finally restart the Grafana Service and check the connectivity and the Version of the Data Source PlugIn.

Uninstall

This chapter describes how to uninstall the PlugIn Proxy and the PlugIn.

Uninstalling the TAUBER PlugIn Proxy on Windows

- Login to the Server running the PlugIn Proxy.
- Run a Command Shell as Administrator.
- Run “bhPlugInProxy.bat -remove” to remove the Windows Service.
- Delete the installation folder.

Uninstalling the TAUBER PlugIn Proxy on Linux

- Login to the Server running the PlugIn Proxy as root.
- Run the command “systemctl disable bhPlugInProxy.service”.
- Delete the file /etc/systemd/system/bhPlugInProxy.service”.
- Reload the systemctl daemon using “systemctl daemon-reload”.
- Delete the installation folder.

Uninstalling the TAUBER Grafana Datastore PlugIn

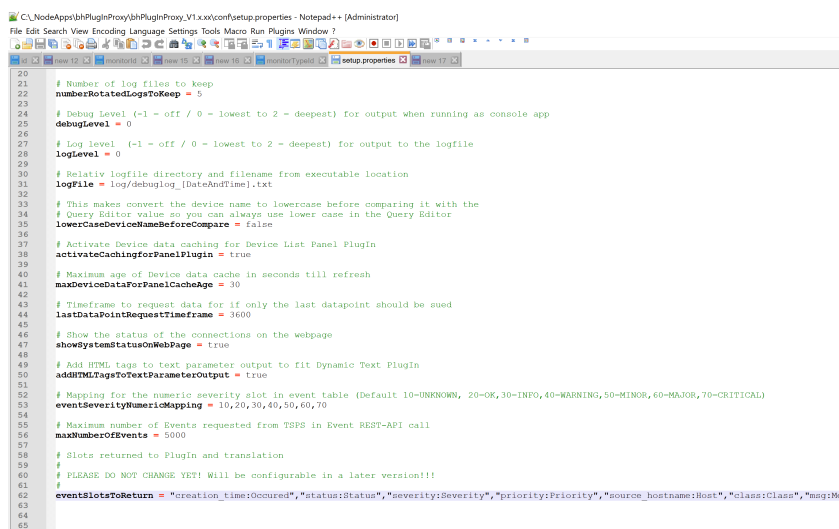
- Login to your Grafana Server
- Delete your TAUBER Grafana Data Source configurations in the Data Source folder
- Delete the plugin folder “bmc_bhom-datasource-plugin” from your external plugins folder in Grafana\data\plugins-bundled.
- Remove the plugin from your Grafana “.ini” file allow_loading_unsigned_plugins property.
- Restart the Grafana Server by restarting the Windows Service.

Configuration

This chapter describes how to configure the PlugIn Proxy and the Grafana PlugIn.

Configuring the TAUBER PlugIn Proxy

The PlugIn Proxy has one single file called “setup.properties” located in the “conf” subfolder of the PlugIn Proxy installation directory. Open the file in Notepad or Notepad++.



```

20
21 # Number of log files to keep
22 numberRotatedLogsToKeep = 5
23
24 # Debug Level (-1 = off / 0 = lowest to 2 = deepest) for output when running as console app
25 debugLevel = 0
26
27 # Log level (-1 = off / 0 = lowest to 2 = deepest) for output to the logfile
28 logLevel = 0
29
30 # Relativ logfile directory and filename from executable location
31 logfile = log/debuglog_[DateAndTime].txt
32
33 # This makes convert the device name to lowercase before comparing it with the
34 # Query Editor value so you can always use lower case in the Query Editor
35 lowercaseDeviceNameBeforeCompare = false
36
37 # Activate Device data caching for Device List Panel Plugin
38 activateCachingforPanelPlugin = true
39
40 # Maximum age of Device data cache in seconds till refresh
41 maxDeviceDataForPanelCacheAge = 30
42
43 # Timeframe to request data for if only the last datapoint should be used
44 lastDataPointRequestTimeframe = 3600
45
46 # Show the status of the connections on the webpage
47 showSystemStatusOnWebPage = true
48
49 # Add HTML tags to text parameter output to fit Dynamic Text Plugin
50 addHTMLTagsToTextParameterOutput = true
51
52 # Mapping for the numeric severity slot in event table (Default 10=UNKNOWN, 20=OK, 30=INFO, 40=WARNING, 50=MINOR, 60=MAJOR, 70=CRITICAL)
53 eventSeverityNumericMapping = 10,20,30,40,50,60,70
54
55 # Maximum Number of Events requested from TSPS in Event REST-API call
56 maxNumberOfEvents = 5000
57
58 # Slots returned to Plugin and translation
59 # PLEASE DO NOT CHANGE YET! Will be configurable in a later version!!!
60 #
61 #
62 eventSlotsToReturn = "creation_time:Occurred", "status:Status", "severity:Severity", "priority:Priority", "source_hostname:Host", "class:Class", "msg:Msg"
63
64
65
  
```

PlugIn Proxy setup.properties in Notepad++

By default, the Proxy will start on Port 3000 using the http Protocol. Below you find the properties you can configure.

httpProtocol = http / https

The protocol used by the Webserver.

certificateFile = ...

keyFile = ...

The path + filename of the certificate and key file used by the Webserver when using https protocol.

Example

certificateFile = conf/secure/selfsigned-cert.pem

keyFile = conf/secure/selfsigned-key.pem

Note: Please store the files in the “secure” subfolder of the “conf” directory! Use the forward slash as separator although running on Windows Operating System!

httpPort = 3000

The port the Webserver will run on.

ignoreErrorOnRestProblemOrNoData = true

This makes the Proxy send an empty response instead of an error for problem requests.

numberRotatedLogsToKeep = 5

This is the number of log files that the Proxy will keep in the log directory.

debugLevel = 0

This is the level of debug when running the proxy as a console application send to standard output.

-1 = off (ONLY ERRORS) / 0 = lowest / 1 = middle / 2 = highest

logLevel = 0

This is the level of debug written to the log files. The logfiles are stored in the “log” subdirectory of the installation folder.

-1 = off (ONLY ERRORS) / 0 = lowest / 1 = middle / 2 = highest

logFile = log/debuglog_[DateAndTime].txt

This is the relative location of the logfile + the logfile name. [DateAndTime] is replaced by the actual daily date stamp. The logfile is rotated every day.

lowerCaseDeviceNameBeforeCompare = false

This property will enable that all Device names and Agent Hostnames will be converted to lower case before using them internally.

activateFilesystemBasedCaching = true

This property will enable the Id caching mechanism for the Devices and Groups.

activateCachingforPanelPlugin = true

This property will enable the data caching mechanism for the Device List Panel PlugIn.

maxDeviceDataForPanelCacheAge = 30

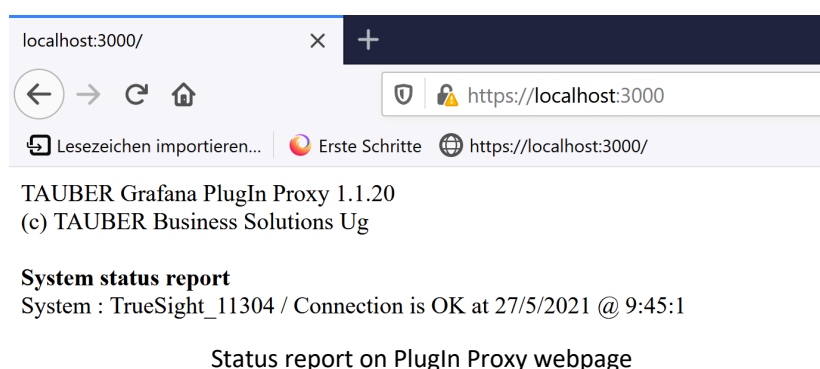
This property will define the time in seconds other Queries will use the data from the Cache for the Device List Panel PlugIn.

lastDataPointRequestTimeframe = 3600

This is the time frame the data is requested for when using the “Last point?” or “Text?” option.

showSystemStatusOnWebPage = true / false

Setting this property to true will show the last connection status on the Webpage of the PlugIn Proxy.



eventSeverityNumericMapping = 10,20,30,40,50,60,70

These are the values the PlugIn Proxy will put into the numeric severity field in the event response based on the event severity (Default 10=UNKNOWN,20=OK,30=INFO,40=WARNING,50=MINOR,60=MAJOR,70=CRITICAL)

maxNumberOfEvents = 1000

This is the maximum number of events returned when requesting the events from Presentation Server. The property can be used to reduce the impact on the Presentation Layer.

eventSlotsToReturn =

"creation_time:Occured", "status:Status", "severity:Severity", "priority:Priority", "source_hostname:Host", "class:Class", "msg:Message"

Note: This property should NOT be changed yet!

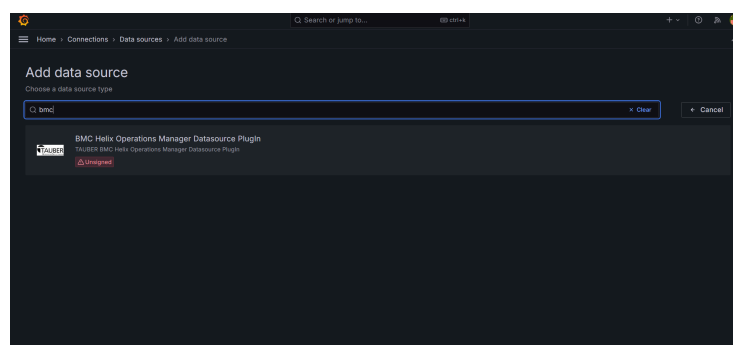
This is the list of Event slots and column names returned in Event list queries. This list is fixed and should NOT be changed.

NOTE: If you want to create Event list with own slots, please use the “Advanced Event Query”.

Configuring the TAUBER Grafana Data Source Plugin

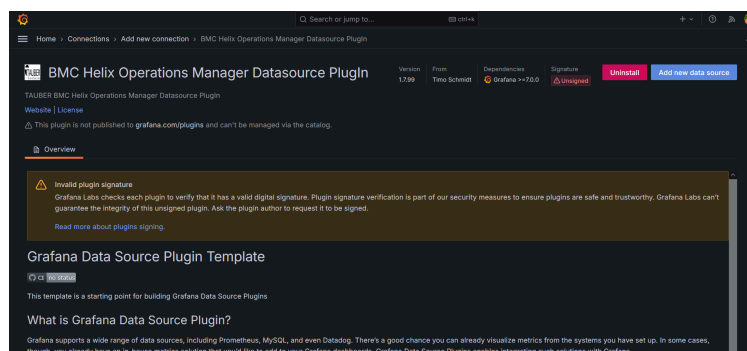
This chapter will describe how to load and configure the TAUBER Grafana Data Source Plugin for BMC Helix Operations Manager. You must install the Plugin as described in “Installing the TAUBER Grafana Datastore Plugin” chapter.

- Login to Grafana and change to “Connections / Add new Connection”
- Type “BMC” in the Search Box



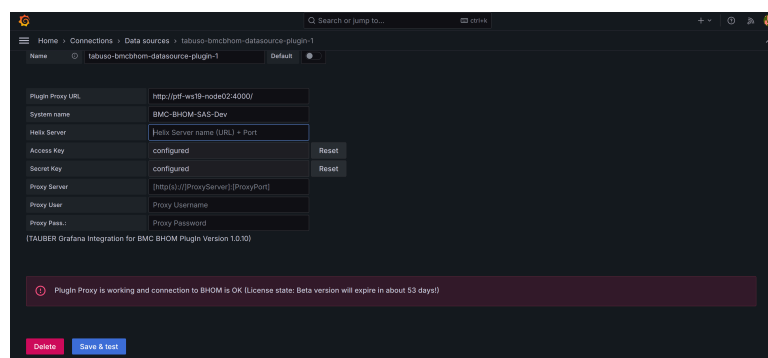
Search Box

- Select the “BMC Helix Operations Manager Datasource Plugin”



Datasource Dialog

- Grafana will switch to the Data source configuration Dialog



Grafana Data Source configuration dialog

Enter the following values in the configuration dialog.

Name

The unique internal name of this PlugIn instance. Select this name in the Query Editor to collect Data and Events from this BHOM system. If you want to connect to multiple systems, you must use different unique names.

Example: tabuso-bmcbhom-datastore-plugin-prod

Note: Select the “Default” switch to use this PlugIn as Default in the Query Editor.

PlugIn Proxy URL (TAUBER PlugIn Proxy URL)

The URL the PlugIn uses to connect to the TAUBER PlugIn Proxy including the Protocol and the Port the PlugIn Proxy is running on.

Example: <https://testproxyserver:3000/>

Note:

By Default, the PlugIn Proxy is running the Webserver using http. To run the PlugIn Proxy and to connect to the Proxy using https read the following chapter.

System name

The System name is a logical name for the BHOM System you connect with this PlugIn to.

Example: BHOM Production

Helix Server

Enter the name of the Helix Presentation Server and the Port HTTPS is running on.

Example: production.helix.tabuso.com:443

Access Key

Enter the Access Key you have generated in your BMC Helix User Profile.

Secret Key

Enter the Secret Key you have generated in your BMC Helix User Profile.

Proxy Server (HTTP / HTTPS Proxy)

Enter the Protocol + Proxy Server + Port if you need a Proxy to access BHOM.

Proxy Username

Enter the name of the User to authenticate at the Proxy.

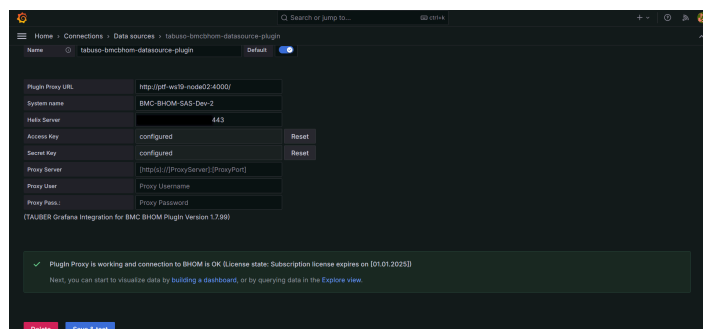
Proxy User Password

Enter the Password to use to authenticate at the Proxy.

IMPORTANT:

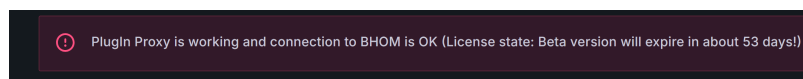
Click the “Save & Test” Button after entering the Access Key. Grafana will show “configured” in the Access Key Input Field. Click the “Save & Test” Button again after entering the Access Key Secret and Grafana will show “configured” also in the Access Key Secret Box.

Finally you must check and save the connection by selecting the “Save & Test” button. Give the PlugIn and the Proxy some time to connect to the Presentation Layer and check the result. After this you can leave the Dialog using the “Back” Button.



Successfully configured and tested PlugIn

If you fail to connect, please check resulting message and the configuration. Read the Troubleshooting chapter to see how to find the cause for problems.



The message above means everything is fine but you have a Beta version with an expiring license!

Using https for the PlugIn Proxy and the PlugIn

By default, the PlugIn Proxy is running using the http Protocol and though the PlugIn must connect to the PlugIn Proxy using the http Protocol. To run the PlugIn Proxy using https and to connect from the PlugIn to the PlugIn Proxy using https follow the description below.

Make sure you have a valid certificate and private key for the PlugIn Proxy Web Server!

Configure the PlugIn Proxy to use https

- Stop the “TAUBER PlugIn Proxy for BMC BHOM” Service.
- Copy the certificate and private key file to the “conf\secure” subfolder of the PlugIn Proxy installation folder.
- Open and edit the “setup.properties” file from the “conf” subfolder of the PlugIn Proxy installation folder.
- Change the “httpProtocol” property to “https”.
- Change the “certificateFile” and “keyFile” properties to relatively point to certificate and key file, e.g.

certificateFile = conf/secure/selfsigned-cert.pem

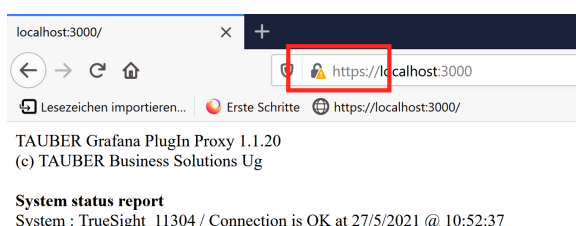
keyFile = conf/secure/selfsigned-key.pem

- Open an elevated (Administrator) Command Shell and start the PlugIn Proxy using “bhPlugInProxy.exe”. Watch out for the following debug lines:

```
DEBUG (Thu May 27 2021 10:05:33 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / StorePath:localcache\neon\disk = True
DEBUG (Thu May 27 2021 10:05:33 GMT+0200 (Mitteleuropäische Sommerzeit)) : Initializing HTTPS Server.
DEBUG (Thu May 27 2021 10:05:33 GMT+0200 (Mitteleuropäische Sommerzeit)) : Checking if files exist.
DEBUG (Thu May 27 2021 10:05:33 GMT+0200 (Mitteleuropäische Sommerzeit)) : Reading certificate and key file.
DEBUG (Thu May 27 2021 10:05:33 GMT+0200 (Mitteleuropäische Sommerzeit)) : HTTPS Server running on port 3000
```

Debug saying that PlugIn Proxy is using https

- Connect to the PlugIn Proxy using your preferred Browser and https.



Proxy running https

- Press CTRL+C to stop the Proxy in the Command Shell.

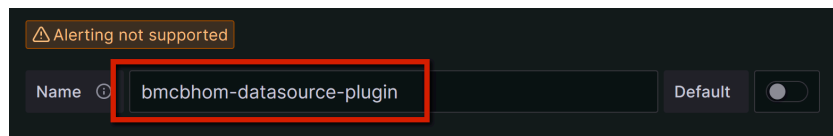
- Start the “TAUBER PlugIn Proxy for BMC BHOM” Service.

The PlugIn Proxy is using https. Now we must configure the Grafana PlugIn to use the certificates!

Configure the PlugIn to use https

In this version of the Grafana PlugIn configuring https is only possible using a so-called provisioning file for the BMC BHOM Data Source PlugIn. Provisioning files are a common way in Grafana to automatically configure PlugIns based on file data.

- Login to your Grafana Server running the BMC BHOM Data Source PlugIn.
- Go to the following folder in your Grafana installation
“...\GrafanaLabs\grafana\conf\provisioning\datasources”
- Create a file called [Data Source Name].yaml. The Data Source Name is the Name field you find in the Data Source Configuration Dialog:



Data Source Name

- Copy the content from the following page to the file and change it with your configuration data. The properties will be explained on the next page.

```

## config file version
apiVersion: 1

## list of datasources to insert/update depending
## # on what's available in the database
datasources:

## <string, required> name of the datasource. Required
- name: tabuso-bmcbhom-datasource-plugin

## <string, required> datasource type. Required
type: tabuso-bmcbhom-datasource-plugin

## <string, required> access mode. direct or proxy. Require
access: proxy

## <map> fields that will be converted to json and stored in json_data
secureJsonData:
accessKey: "[Place your Access Secret Key here]"
accessSecretKey: "[Place your Access Secret Key here]"

tlsClientCert: |

    -----BEGIN CERTIFICATE-----
    Two Example lines: MIID4jCCAsocCCQckFmwnDEer7TANBgkqhkiG9w0BAQsFADCBsjELMAkGA1UEBhMC
    ROUxDDAKBgNVBAGMA05SVzERMA8GA1UEBwwlQm9ybmhlaW0xJTAjBgNVBAoMHFRB
    ... Your Certificate Data Lines!!!
    -----END CERTIFICATE-----

tlsClientKey: |

    -----BEGIN PRIVATE KEY-----
    Two Example lines: MIIEpAIBAAKCAQEAAurxJS6y5/4x/fXpoazvC04mrr+SODkiyBxyVimBeoK7R9cg4
    OT0Ga4MOReyukV13WCMmBT+GVdZQtYDv9yG17NmoODf0DsZtU45Z2IKLfflobP9U
    ... Your Private Key Data Lines!!!
    -----END PRIVATE KEY-----

jsonData:

    proxyUrl: [Your Plugin Proxy URL]
    systemName: [Your System name]
    helixServer: [Your Helix Server + Port]
    proxyServer: [Your HTTP or HTTPS Proxy Server + Port]
    proxyUsername: [Your Proxy Username]
    proxyPassword: [Your Proxy User Password]

    tlsAuth: true
    tlsSkipVerify: true # <string> json object of data that will be encrypted.

# certs can be yaml formatted with "| actual string" OR "Base64PEM"
## <bool> allow users to edit datasources from the UI.

editable: true

```

Just change the **red** marked values in square brackets with your connection / configuration data. If you have changed the name of the Data Source, you also must change the name in this Provisioning file to match your Data Source name. Finally, you must paste your certificate and private key into the file between the following lines and remove the example:

```
-----BEGIN CERTIFICATE-----
```

```
-----END CERTIFICATE-----
```

and

```
-----BEGIN PRIVATE KEY-----
```

```
-----END PRIVATE KEY-----
```

Restart the Grafana Service, login and change to the Data Source. If you have problems to connect change to the “Troubleshooting” chapter.

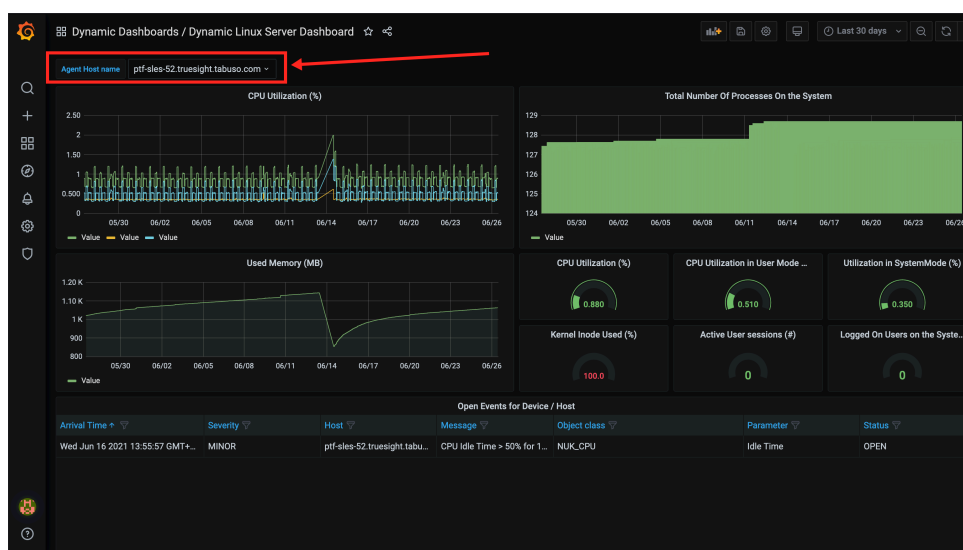
How to use the TAUBER Grafana Plugin

Introduction

After you installed and configured the PlugIn Proxy and configured the PlugIn to connect to your BHOM System you can start adding Panels to your Dashboards that show Metric and Event data from the BHOM.

There are basically two different types of Dashboards or Panels you can create. Static Panels have a static connection to a Device or Device, Monitor Type, Monitor Instance and Attribute and always show the Data or Events from the same Metric.

A dynamic Panel has a Device name created based on a so called Grafana Variable. The actual value of the variable and so of the Device name is generated based on a query that is executed for example when the Dashboard is loaded, and a selection is made by the User afterwards.

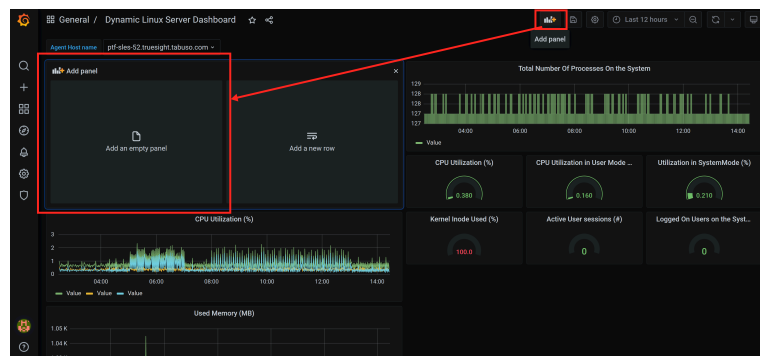


Dynamic Dashboard with Drop Down Selection at the top left corner

Static Dashboards and Panels

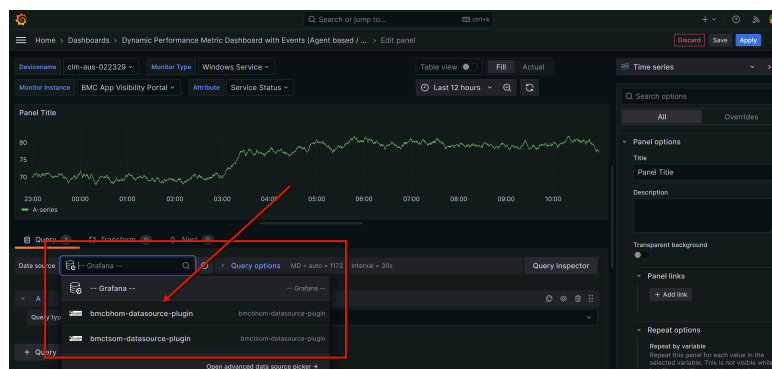
To create a Panel with a static connection to a metric and display a Graph or Event list follow the process below:

- Go to the Dashboard you want to add the Panel to
- Select the “Add panel” button at the top right and “Add an empty panel” afterwards



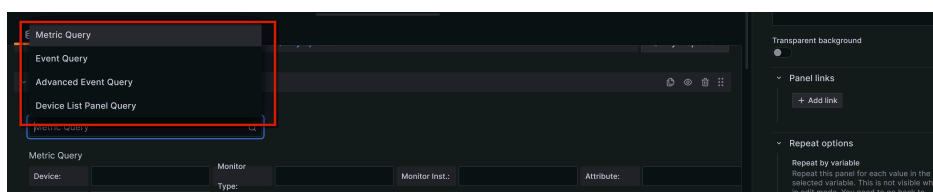
Adding a new static Panel to a Dashboard

- Now you are in the so-called Query Editor



Query Editor for new Panel

- Select the “tabuso-bmcchom-datasource-plugin” connected to your BHOM in the Drop-Down list. By using the “Query Selection” dropdown list you can choose one of the Query Types.
- Select the Query Type you want to use from the Dropdown list. Based on your selection the required input group will be enabled.



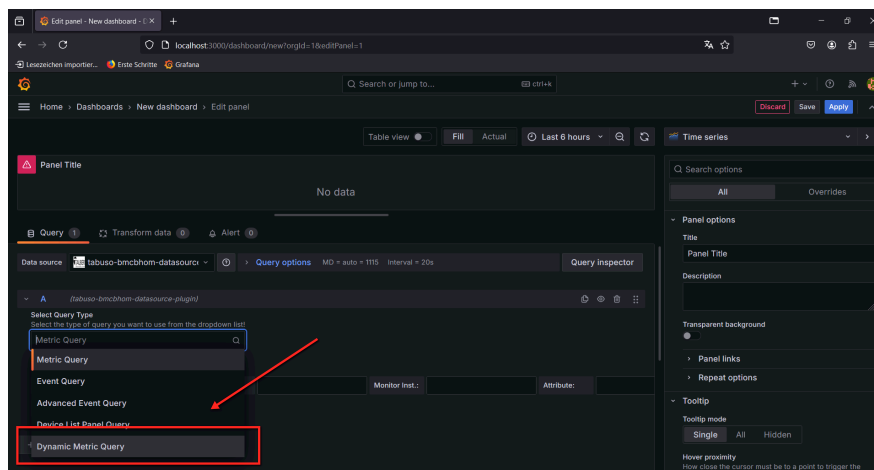
Query Selection

Note: The “Device List Panel Query” only works with the TAUBER Device List Panel which is a separate TAUBER Product!

Dynamic Metric Query

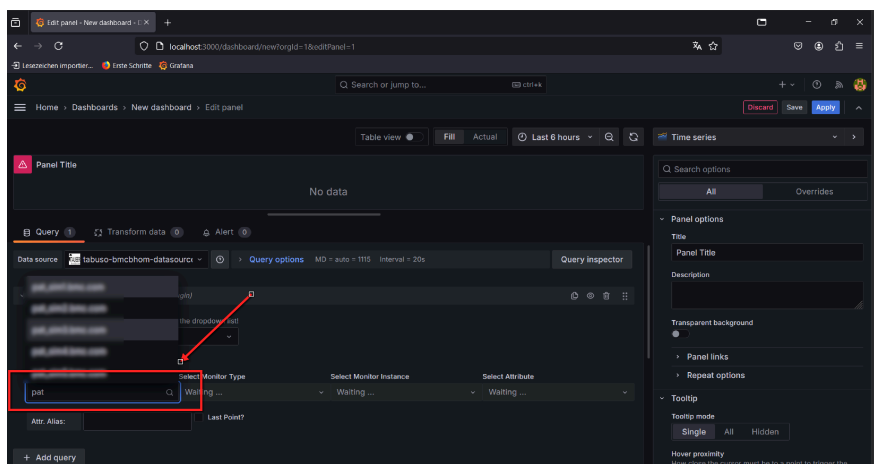
The Dynamic Metric Query is an enhanced version of the Metric Query to create Panels based on Metric without finding out the Device, Monitor Type, Monitor Instance and Attribute name on your own. The Query connects to BHOM and shows the Available Devices etc. in a Dropdown Box that can be searched, and you can select the Input value from.

Select the “Dynamic Metric Query” from the List of available Queris.



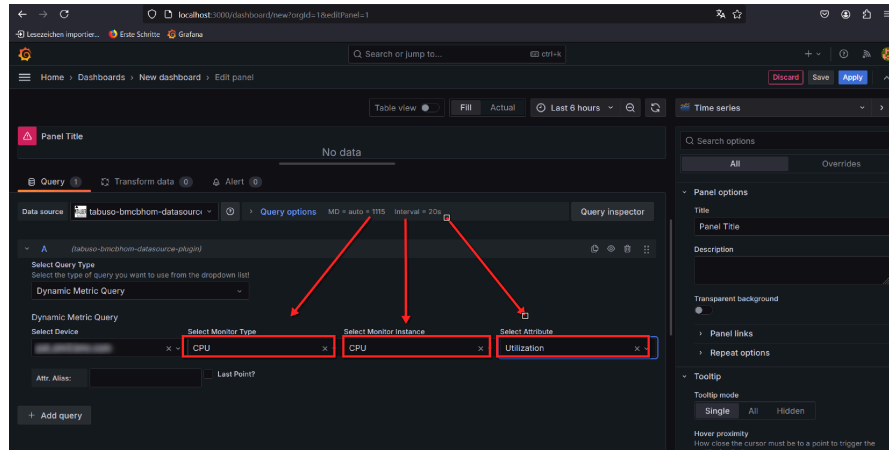
Query Editor with Metric definition

Wait till the Device Data is loaded and select a Device from the Dropdown list. You can search for Devices by typing in the Dropdown Selection Box.

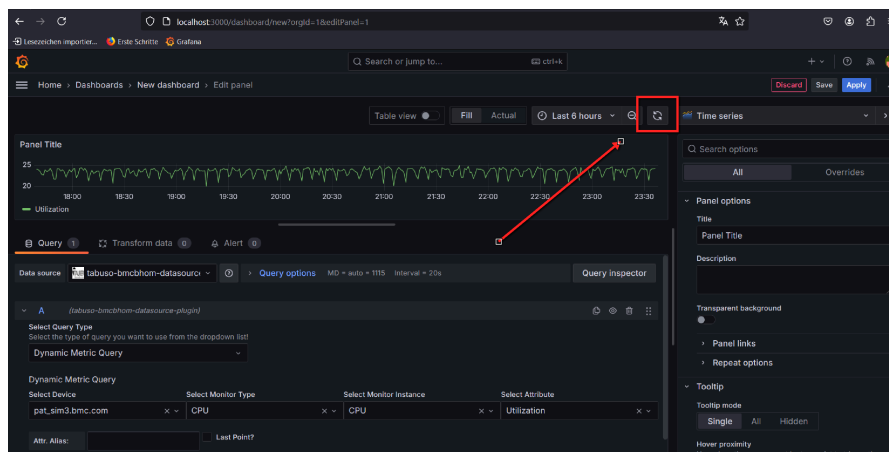


Serching Device in the Dynamic Metric Query

After selecting a Device, the Data for the Monitor Type will be loaded. Select the “Monitor Type”, then the Monitor Instance and finally the Attribute.



Query with selected values



Query with data

Attr. Alias

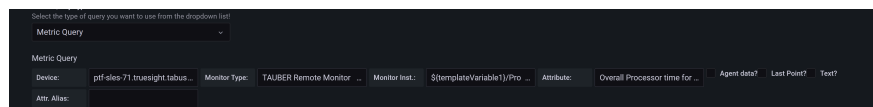
Use the “Attribute Alias” to override the predefined Attribute name received from BHOM.

Last point?

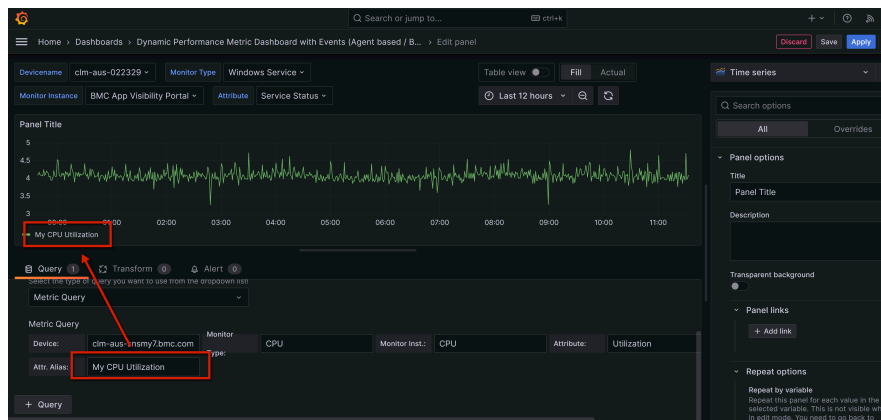
Use the “Last Point” Option if you use a Panel like the Traffic Light PlugIn that needs and displays only the last status of the Metric. This will reduce the Impact because the Datasource will request only the last current datapoint from BHOM.

Metric Query

To collect Metric Data type in a Device name that is present in BMC Helix Operations Manager and use the same name as used in BHOM. Type the Monitor Type name you want to collect the data from. The Monitor must be available on this Device e.g., “CPU”. Type the Monitor Instance name, e.g., “CPU” and finally type the Attribute name you want to collect the data from e.g., “Utilization”.

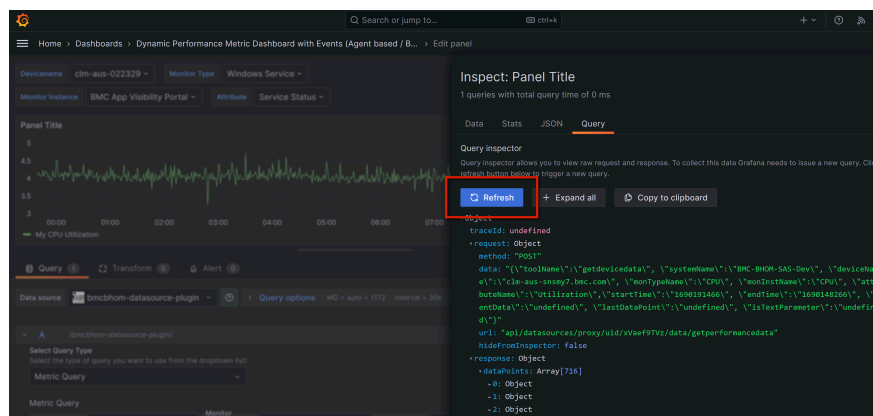


Query Editor with Metric definition



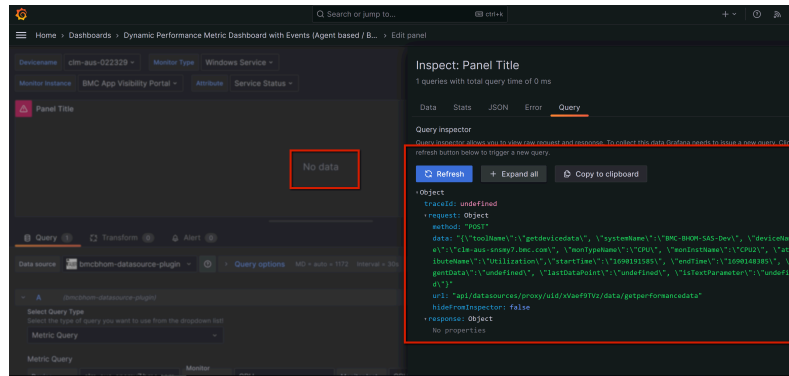
Legend override with Attribute Alias

To check if your definition works select the “Query inspector” button.



Query Inspector after executing query

- Click the “Refresh” button to execute the Query against your Data Source. Below the button you see the result of your Query and in the Background the Graph.



Query Inspector with Error message

If you try to collect data from a not existing Device or Monitor, you will get a “Bad request” or “No data” error in the Query Inspector. First, you should check in BHOM if the Device and Monitor exists. Try to graph the data in the BHOM Device view.

Last point?

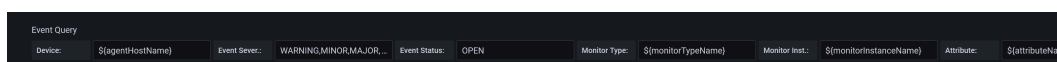
A lot of Panels in Grafana just need the actual value of a metric so collecting the whole timeframe for these attributes creates overhead on almost all components. To reduce this overhead, select this option. As an example, you can have a look at the Stop light Panel.

Congratulations! You have created your first Panel with Data from BHOM!!!

Event Query

To create an Event list Panel, select the “Event Query” from the Dropdown list. You can show a list of Events for a Device. Enter the Device name in the appropriate Input field.

In the “Event Severity” Input field you can enter a comma separated list of severities you want to get the Events for, e.g., “CRITICAL, MAJOR, MINOR”. In the “Event Status” field you can enter a comma separated list of status e.g., “OPEN”.



Pic 20: Query Editor with Event data configuration

If you have a dynamic Dashboard with Monitor Type, Instance and Attribute Variables you can use them in the filter. Adding only the Monitor Type name and leaving instance and attribute empty will show all Events from the Device and this Monitor Type.

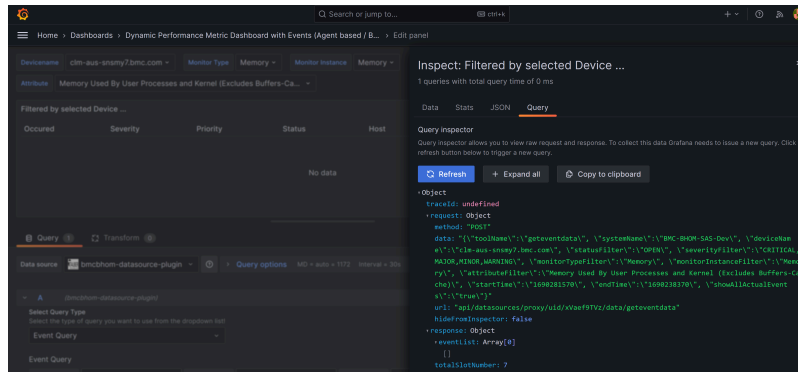
IMPORTANT

The filters need the following slots in the Event set with the corresponding IDs. “object_class”, “object_id” and “a1_parameter_name”. For example, PATROL Events do not have these slots so will not appear.

Note:

By default, the selected Events are only displayed if they match the time frame selected for the Panel or Dashboard. If you choose to display Events with the Status “Open” they must have been **created** in the selected time frame e.g., the last 6 hours. To change this, select the Checkbox “Show all events” and all matching events will be displayed independent from the selected Dashboard timeframe.

- Go to the Query Inspector and run the Query by selecting the “Refresh” Button. You might see NO Event Data ... this might have two reasons. First, there are no “Open” Events that have been created in the selected time frame.



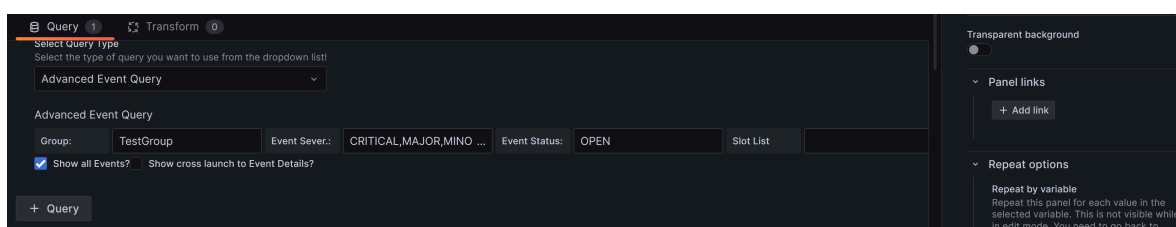
Query Inspector with no Event data

Second, you choose a visualization for Metric Data not a Table view for Event data.

Advanced Event Query

To create an Event list Panel independent from a BMC Helix Operations Manager Device you can use the “Advanced Event Query”. Select the “Advanced Event Query” from the Dropdown List.

Use the following Input Fields and Options to configure your Query:



Advanced Event Query

Group: There are no Eventgroups in BHOM anymore!

Event Sever.: Enter a comma separated list of Severities you want to show in the Table, e.g.

CRITICAL, MAJOR, MINOR

Event Status: Enter a comma separated list of Event status you want to show in the Table, e.g.

OPEN, CLOSED

Slot List: Use the slot list to define which Event slots you want to show and how the slot should be named in the Table view, e.g.

creation_time:Occured,status:Status,severity:Severity,priority:Priority,source_hostname:Host,msg:Message

IMPORTANT: If you leave the field empty the definition from the property eventSlotsToReturn in the setup.properties file on the PlugInProxy will be used.

Slot Filter: Use this field to create a Slot based filter. You can use RegEx on the Expression side of the filter, e.g.

mc_object_class=NT_CPU. || mc_object_class=NT_LOGICAL_DISKS.**

Use the “&” and “||” operator to create a slot filter with multiple expressions.

IMPORTANT: Please keep in mind that the filter is processed from the left to the right without any brackets!!!

mc_object_class=NT_CPU. || mc_object_class=NT_LOGICAL_DISKS.**

With version 1.9.10 we introduce the “&” and “|” operator to create a slot filter with multiple expression.

IMPORTANT: Please keep in mind that the filter is processed from the left to the right without any brackets!!!

onBHOM Option

By default, the PlugIn Proxy requests all Events from BHOM based on the Status and Severity Filter and afterwards applies the Slot filter on his own. Using the “onBHOM:” String at the beginning of the Slot filter (no spaces) tells the PlugIn Proxy to include the Filter in the REST-Call to BHOM which reduces the number of returned Events and though the speed of the request.

IMPORTANT:

In this version just a simple [Slot name]=[Value] filter is supported by the onBHOM Option, e.g.:

onBHOM:mc service=TESTSERVICE & class=ALARM & msg=Linux.*

Show all events? To show all matching Events independent from the timeframe of the Dashboard select this checkbox.

Show cross launch to Event Details? To add a slot with a Link to the TSPS Console Event Details please select this checkbox.

IMPORTANT:

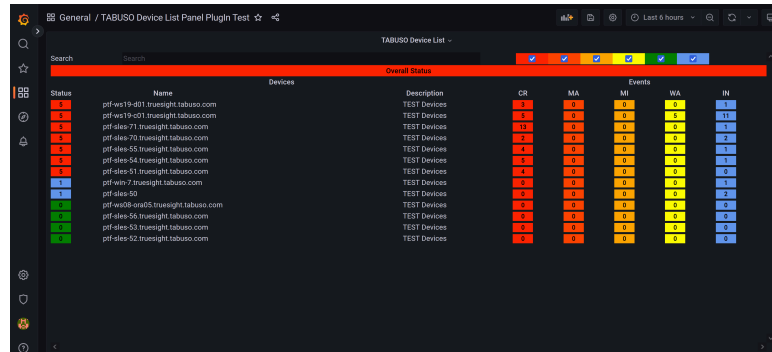
To finally add a clickable link to the Table view please follow the steps below:

- Edit the Table Panel properties and add a Data Link.
- Add “BHOM Event Details” in the Title field
- Add the following Text to the URL field:
`${__data.fields["Link"]}`
- *Select “Open in new tab” if necessary*

NOTE: Actually, we found some problems clicking the link. If you can't login to BMC Helix Operations Manager by clicking the Link please copy and paste it into a new Browser tab.

Device List Panel Query

The Device List Panel Query is designed to provide the data for the TAUBER Grafana Device List Panel PlugIn.



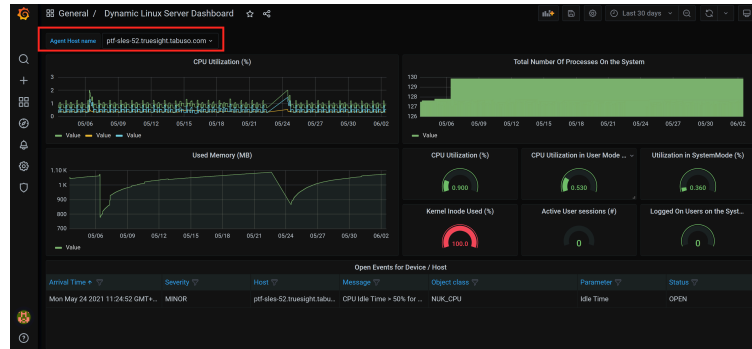
Status	Name	Description	Overall Status				Events			
			CR	MA	MI	WA	IN	WA	IN	IN
OK	pf-ws19-001.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-ws19-001.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-71.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-70.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-53.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-54.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-51.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-ws07.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-50	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-ws08-ora05.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-56.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-53.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0
OK	pf-slep-52.trueights.tabuso.com	TEST Devices	0	0	0	0	0	0	0	0

Device List Panel PlugIn

If you are interested in the Device List Panel PlugIn please contact TAUBER Business Solutions.

Dynamic Dashboards and Panels

Dynamic Dashboards in Grafana are Dashboards that have a dynamic meaning selectable component that finally can be used in the Query Editor to replace input values.



Dynamic Dashboard with Drop Down List for Server names

To create such a Drop-Down list and use the selected value in the Panels of your Dashboard you must create a so-called Variable in the Dashboard Definition. The TAUBER Grafana PlugIn for BMC TSOM supports the following variables:

- `deviceName`: Selectable list of Device names
- `agentHostName`: Selectable list of Agent Hostnames
- `monitorTypeName`: Selectable list of Monitor Types for a Device
- `monitorInstanceName`: Selectable list of Monitor Instances for a Device and Monitor Type
- `attributeName` & `attributeName2`: Selectable list of Attribute names for a Device, Monitor Type and Monitor Instance
- `templateVariable1` to 99: User based static content

IMPORTANT

The “Dynamic Metric Query” only supports `templateVariable1` to 5.

The following chapters will show you how to add the variables in your Dashboard and use them in the Query Editor.

IMPORTANT

In the current version we support different operators for the different variables in the query. Please see next page to see the supported operators and syntax.

Variables and Queries

In the current version we support different operators and attributes for the different Variables in the query. The following table will show the supported operators, attributes, and some sample queries:

#	Variable	Attributes	Operator	Operand	Examples
1	deviceName (dynamic)	id: Numeric internal ID of the Device e.g., 248 name String DNS name of the Device e.g. or dispName: "demosever.demodom.de" type: Type of the Device, e.g. "Server" totalAgents: Number of PATROL Agents	& (AND) (OR)	contains equals	Example #1 <i>name contains "www."</i> Add all Devices that have a "www." in the name Variable.
2	agentHostName (dynamic)	hostname: Name of the host running the Agent fqdnname: FQDN of the host running the Agent agentID: Numeric PATROL Agent Id e.g., 184. agentPort: Numeric PATROL Agent Port e.g., 3181. tagDetails: Comma separated list of tags applied agentOS: String containing the Operating System e.g., "Windows Server 2019 Standard". agentVersion: String containing the Agent version e.g., "V11.3.02.01i". policyStatus: Numeric value of Policy application status e.g., 3. connectionStatus: Numeric value of the PATROL Agent Status e.g., 1 = PATROL connected. deployStatus: Numeric value of latest deployment status e.g.,5. markForDeleteStatus: Numeric value of MfD status e.g., 0 policiesApplied: Comma separated String value containing the applied policies e.g., "NT_WINDOWS_KM". policiesFailedToApply: Comma separated String value .containing the policies that failed to apply, e.g. "NT_WINDOWS_KM". agentip: Sstring value containing the IP Address of the Agent host "192.168.1.107".	& (AND) (OR)	contains equals	Example #1 <i>tagDetails contains "MonitoringServer" tagDetails contains "BHOM" & connectionStatus equals 1</i> Add all Agents that have a tag containing "MonitoringServer" or "BHOM" and are connected.

		policyManagedAgent: String value containing true or false if Agent is Policy managed.			
3	monitorTypeName (static)	name: Name of the Device to get the Monitors from. dispName: Name of the Device to get the Monitors from.	& (AND)	equals	Example #1 <i>name equals "\$agentHostName"</i>
4	monitorInstanceName (static)	name Name of the Device to get the Monitor or dispName: instances from. monitorType: Name of the Monitor Type to get the instances for. cleanUp: [true false]: Clean up the Monitor instance path?	& (AND)	equals	Example #1 <i>dispName equals "\$agentHostName" & monitorType equals "\$monitorTypeName" & cleanUp equals "true"</i>
5	attributeName & attributeName2 (static)	name Name of the Device to get the Attributes or dispName: from. monitorType: Name of the Monitor Type to get the Attributes for. monitorInstance: Name if the Monitor instance to get the Attributes for. cleanUp: [true false]: Clean up the Monitor instance path? excludeTextParameters: [true false] to exclude Textparameters from the list	& (AND)	equals	Example #1 <i>dispName equals "\$agentHostName" & monitorType equals "\$monitorTypeName" & monitorInstance equals "\$monitorInstanceName" & cleanUp equals "true" & excludeTextParameters equals "true"</i>
6	templateVariable (fixed)	-	-	-	Example #1 <i>ptf-sles-70,ptf-sles-71</i> Comma separated list of hostnames.

What does the Variable type mean?

The Variable type (dynamic, static, fixed) tells you how flexible you can create the Query in this Variable. The Query of Variables set to “dynamic” can have multiple Query parts like “[Attribute] [contains | equals] “[Value]”. The parts of the query can be combined with the “&” (AND) or “|” (OR) operator. Here is an example:

tagDetails contains "MonitoringServer" || tagDetails contains "BHOM" & connectionStatus equals 1

This query will add all Agents to the list that have a tag set containing "MonitoringServer" or "BHOM". Next to this these Agents must be currently connected to BHOM (connectionStatus equals 1). Variables set to "static" have a static Query syntax with some values that can be modified.

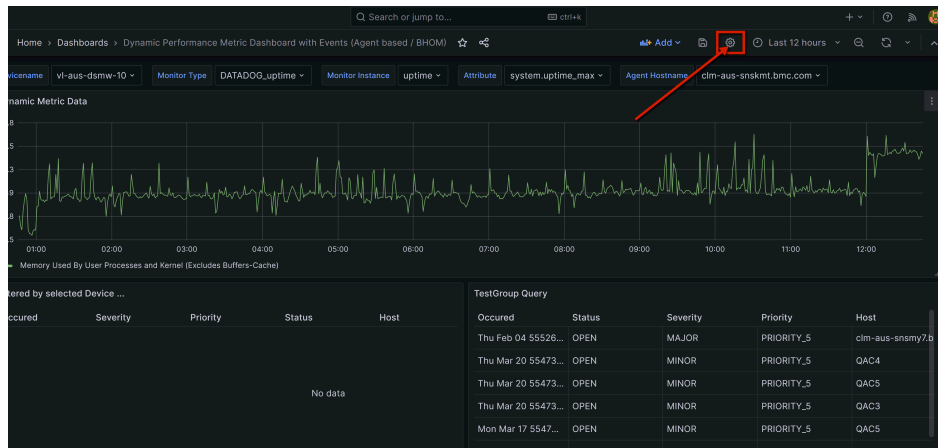
dispName equals "\$agentHostName" & monitorType equals "\$monitorTypeName" & cleanUp equals "true"

The values \$agentHostName and \$monitorTypeName could be replaced by values like "CPU", but the overall syntax must be kept. Finally, "fixed" Variables have a comma separated list of values that can be changed but the values are fixed.

The deviceName variable

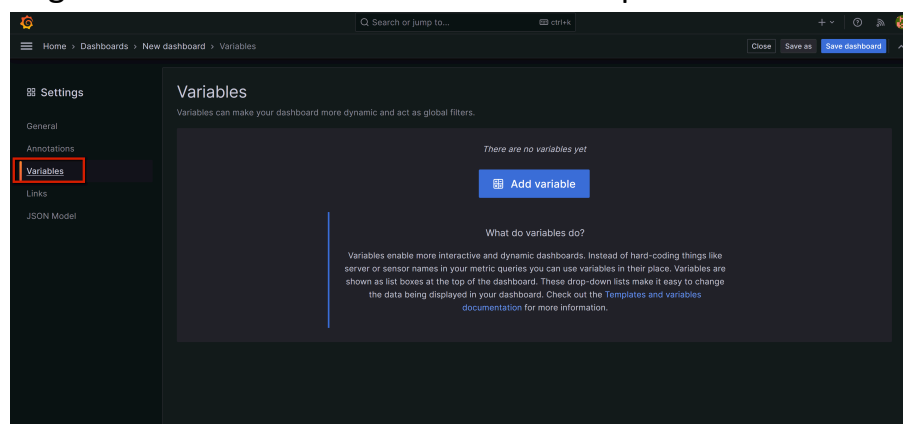
The deviceName Variable is used to create a Drop-Down List with Device names from BHOM. To create this Variable, follow the steps below:

- Open the Dashboard you want to add the Grafana Variable to and enter the Settings of the Dashboard.



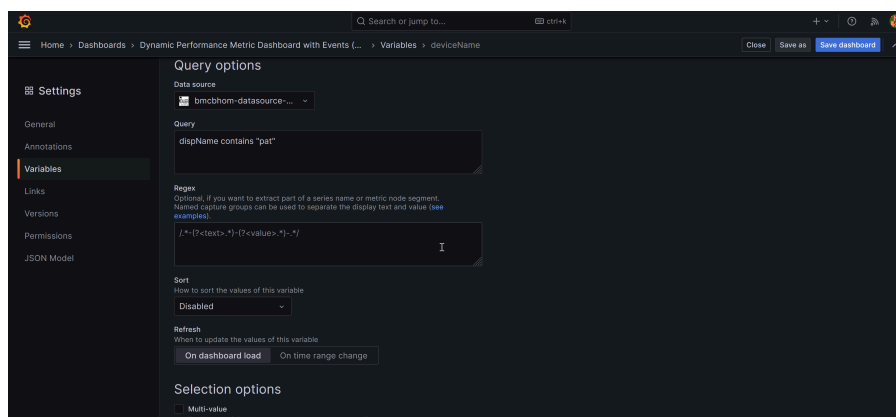
Dynamic Dashboard with Settings Button

- Change to the “Variables” Section in the top left corner of the menu.



Dashboard Settings Variable Section

- Click the “Add variable” button.



Dashboard Settings Variable Section with deviceName Variable

- In the Name field you MUST enter “deviceName”. The Type must be “Query”
- The Label and Description can be configured the way you want. The Label is the text next to the Drop-Down list in the Dashboard. The Description will appear when you hover the Drop-Down list.
- In the “Query Options” you must select the TAUBER Grafana Data Source PlugIn as “Data source”. I suggest selecting “On Dashboard Load” for the Refresh. This will make the Dashboard update the Device list every time you enter the Dashboard.
- You can filter Devices using a simple Query Language in the Query Field. The Device has the following properties you can use in the Query field:

id:	Numeric internal ID of the Device e.g., 248
name	String Display name of the Device, e.g.
or dispName:	“demosever.demodom.de”
type:	Type of the Device, e.g. “Server”
totalAgents:	Number of PATROL Agents

NOTE: To show all Devices just enter “ALL” in the Query input box!

You can use the “contains” and the “equals” operator. As an example:

dispName contains “www”

will only show the Devices that have a “www” in the Device Display name. To make sure only URLs are displayed use the following expression:

dispName contains “[www.](#)”

NOTE: The deviceName Variable supports the & (and) and || (or) Operator. Please see the Demo Dashboards for examples!

- Next to the Query field you can use RegEx in the RegEx field to extract Devices applying the RegEx to the resulting Device names.
- Select a Sort option from the Drop-Down list to apply sorting.

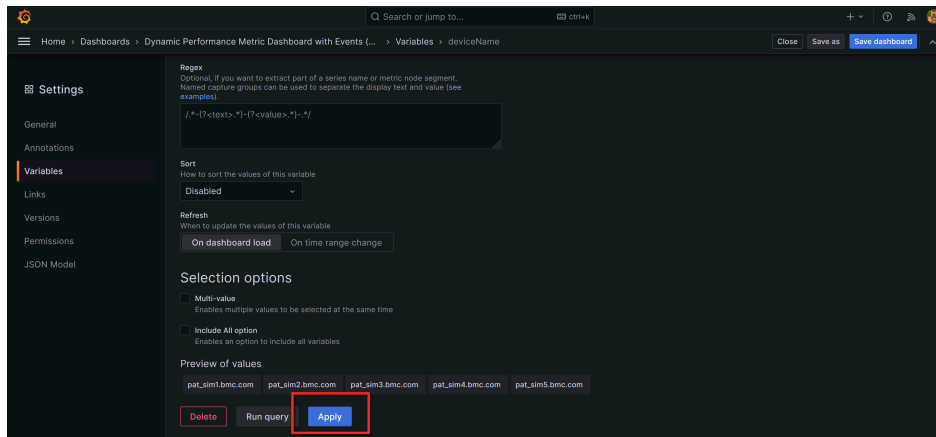
NOTE: “Multi-value” and “Include All option” is **not supported** yet and should be switched off! Also, value tags are NOT supported yet!

- Select the “Run query” Button to preview the result including Query and RegEx.



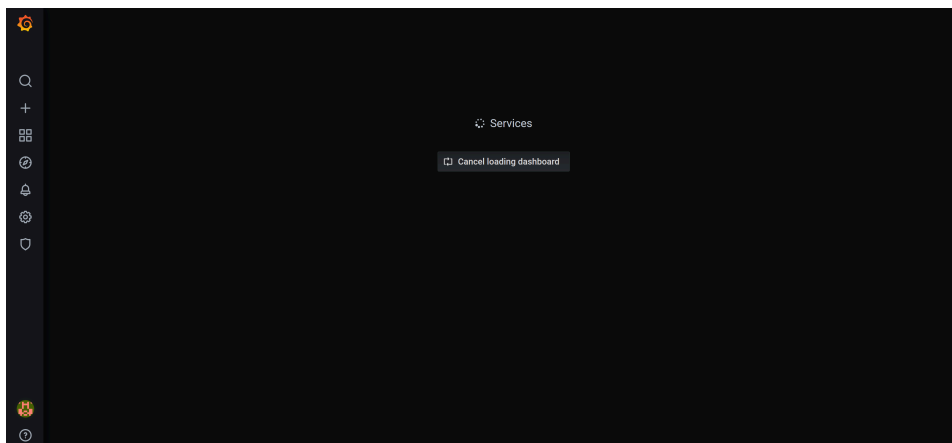
deviceName Variable with Preview

- Select the “Apply” button.



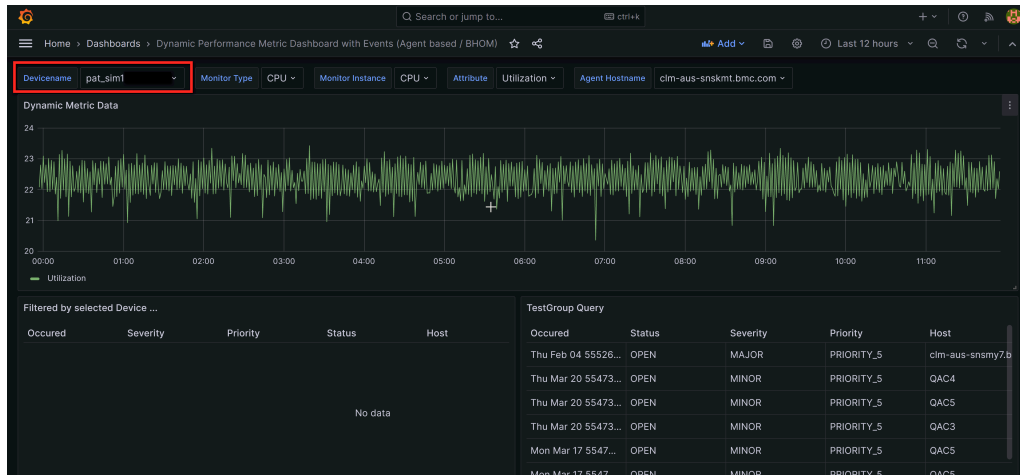
Save Dashboard after adding a Variable

- Open the Dashboard.



Dashboard loading data for Variable = Drop-Down list

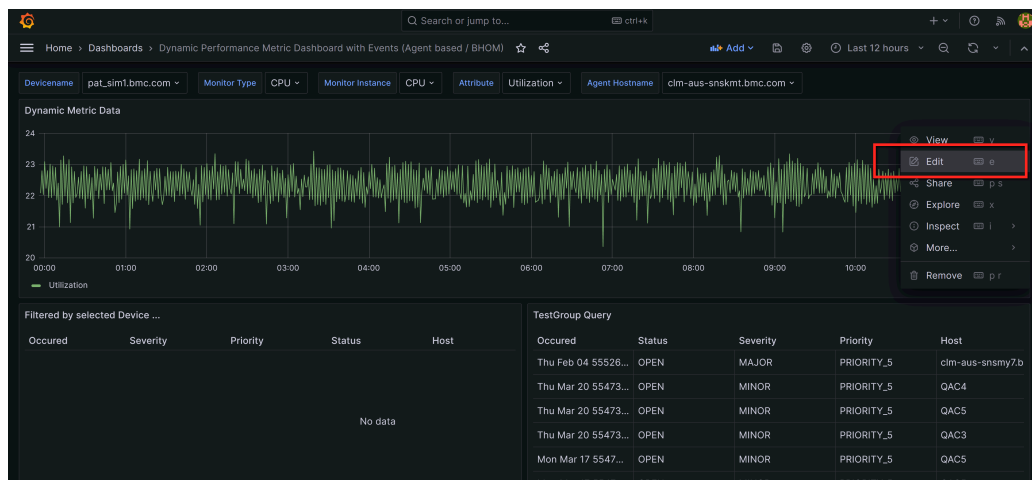
- You will see a message that the “Services” are loading which means the REST call to fill the Drop-Down list with values is currently executing. After this you find the Drop-Down list in the top left.



Dynamic Dashboard with Drop-Down list

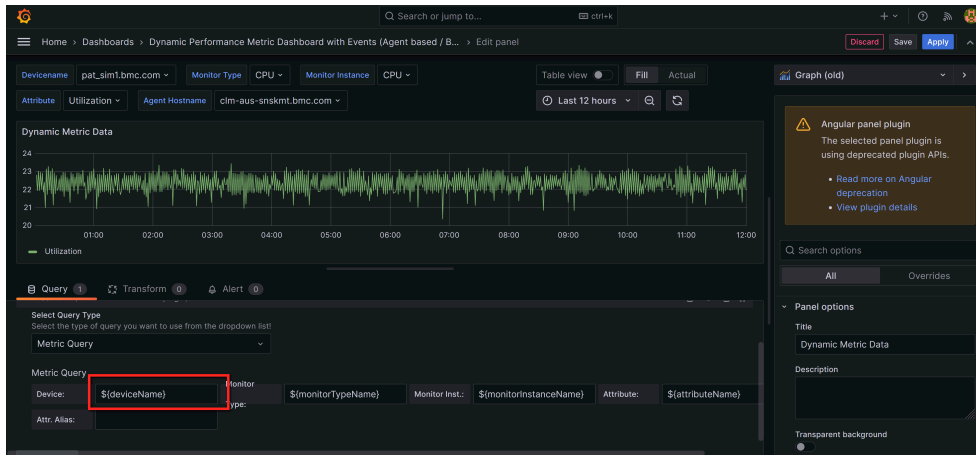
NOTE: The Panels in the Dashboard are still static because they have a fixed Device name in the Device name input field. To finally make the Dashboard dynamic we must modify all Panels to use the value from the Drop-Down list.

- Open one Panel after the other by selecting the three dots in the title of the Panel and select “Edit” from the menu.



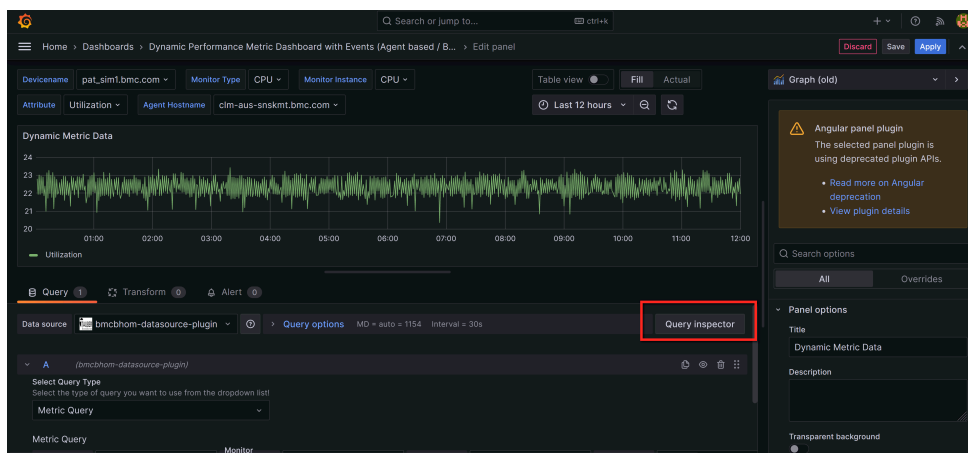
Editing a Panel in the Dashboard

- At the bottom of the screen scroll down to the Metric Definition Input fields and replace the static Device name with the Variable definition “\${deviceName}” which tells Grafana to use the value from the Drop-Down list.



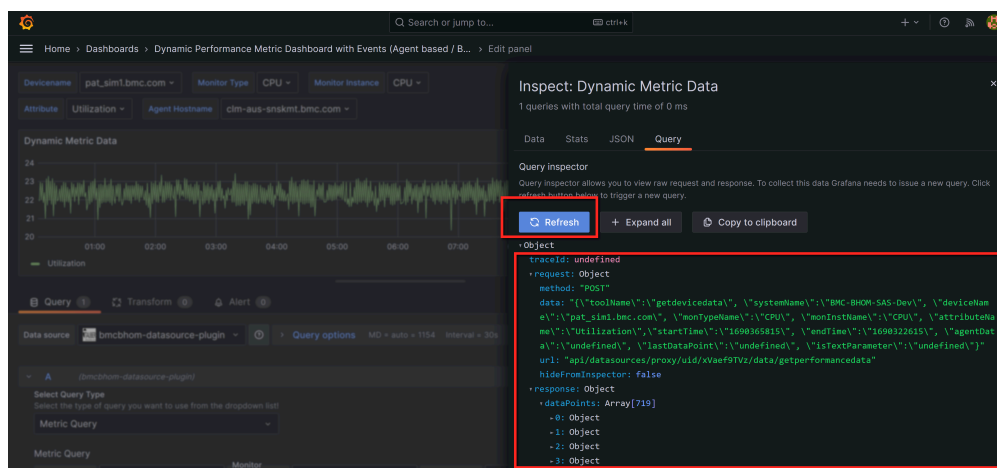
Query Editor with deviceName Variable

- Scroll from the Query Editor (At the bottom of the screen) to the top to see the “Query inspector” button.



Query Inspector button

- Press the “Query inspector” button to check if the Query with the variable works.



Query Inspector with Query result

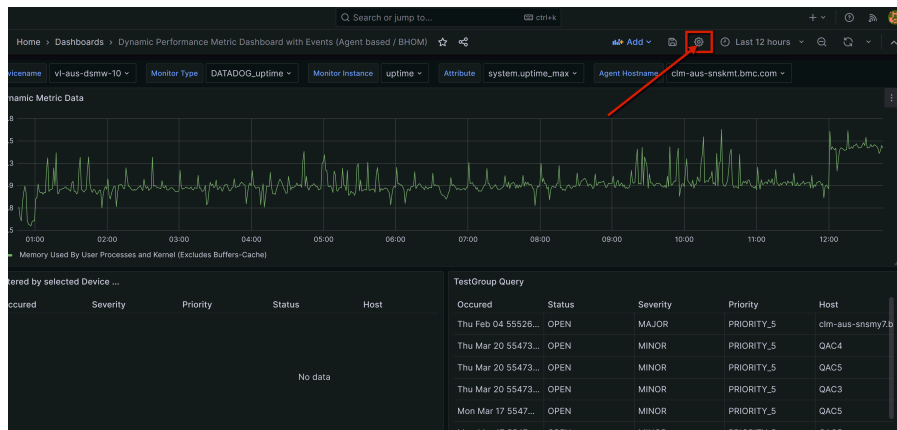
Check the result of the Query. For Metric data you should see a “dataPoints” Array with several entries containing a timestamp and a value. If you get an error, check the “deviceName” Variable and if the Monitor Type, Instance and Attribute name exist on the selected Device(s).

For more change to the “Troubleshooting” section of this manual!

The agentHostName variable

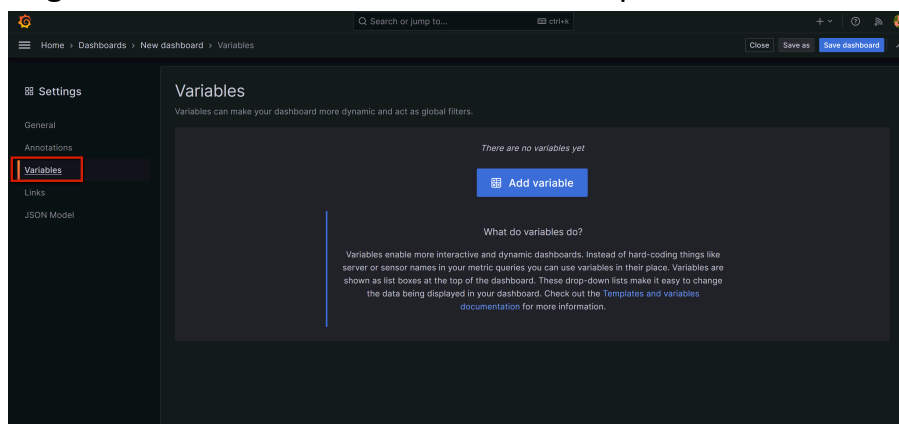
The agentHostName Variable is used to create a Drop-Down List with Hostnames from BHOM running a PATROL Agent. To create this Variable, follow the steps below:

- Open the Dashboard you want to add the Grafana Variable to and enter the Settings of the Dashboard.



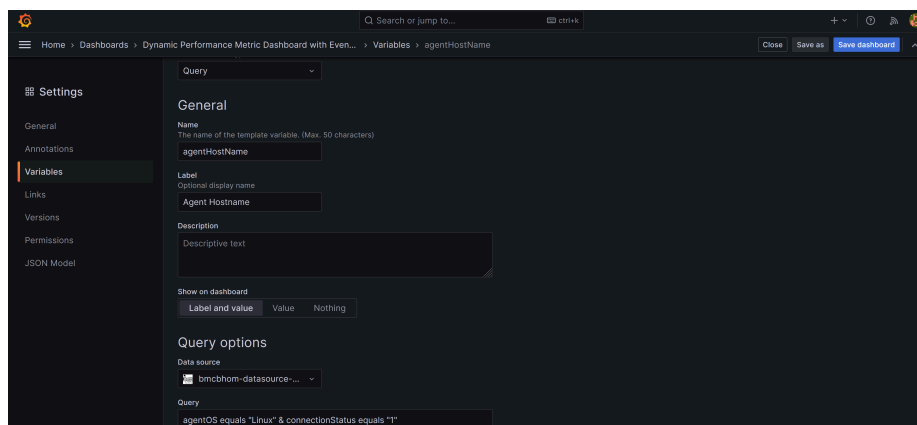
Dynamic Dashboard with Settings Button

- Change to the “Variables” Section in the top left corner of the menu.



Dashboard Settings Variable Section

- Click the “Add variable” button.



Dashboard Settings Variable Section with agentHostName Variable

- In the Name field you MUST enter agentHostName. Type must be “Query”
- The Label and Description can be configured the way you want. The Label is the text next to the Drop-Down list in the Dashboard. The Description will appear when you hover the Drop-Down list.
- In the “Query Options” you must select the TAUBER Grafana Data Source PlugIn as “Data source”. I suggest selecting “On Dashboard Load” for Refresh. This will make the Dashboard update the Agent list every time you enter the Dashboard.
- You can filter Hosts using a simple Query Language in the Query Field. The Hostname has the following properties you can use in the Query field:

hostname:	Name of the host running the Agent
fqdnname:	FQDN of the host running the Agent
agentID:	Numeric PATROL Agent Id e.g., 184
agentPort:	Numeric PATROL Agent Port e.g., 3181
tagDetails:	String containing associated tags e.g., “Linux”
agentOS:	String containing the Operating System e.g., “Windows Server 2019 Standard”
agentVersion:	String containing the Agent version e.g., “V11.3.02.01i”
policyStatus:	Numeric value of Policy application status e.g., 3
connectionStatus:	Numeric value of the PATROL Agent Status e.g., 1 = PATROL connected
blackoutStatus:	Numeric value of PATROL Agent blackout status e.g., 0
deployStatus:	Numeric value of latest deployment status e.g., 5
markForDeleteStatus:	Numeric value of MfD status e.g., 0
policiesApplied:	Comma separated String value containing the applied policies e.g., “NT_WINDOWS_KM”
policiesFailedToApply:	Comma separated String value containing the policies that failed to apply, e.g. “NT_WINDOWS_KM”
agentIp:	String value containing the IP Address of the Agent host “192.168.1.107”

policyManagedAgent: String value containing true or false if Agent is Policy managed

You can use the “contains” and “equals” for the values. As an example: agentOS contains “Windows”

will only show the Hosts running the Windows Operating System.

NOTE: The agentHostName Variable supports the & (and) and || (or) Operator. Please see the Demo Dashboards for examples!

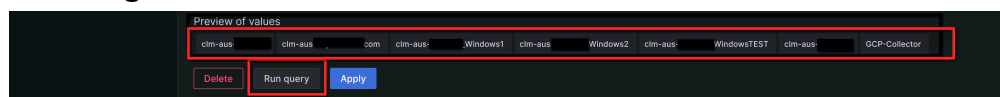
Example:

tagDetails contains "MonitoringServer1" || tagDetails contains "MonitoringServer2" & connectionStatus equals "1"

- Next to the Query field you can use RegEx in the RegEx field to extract Hosts applying the RegEx to the resulting Hostnames!
- Select a Sort option from the Drop-Down list to apply sorting.

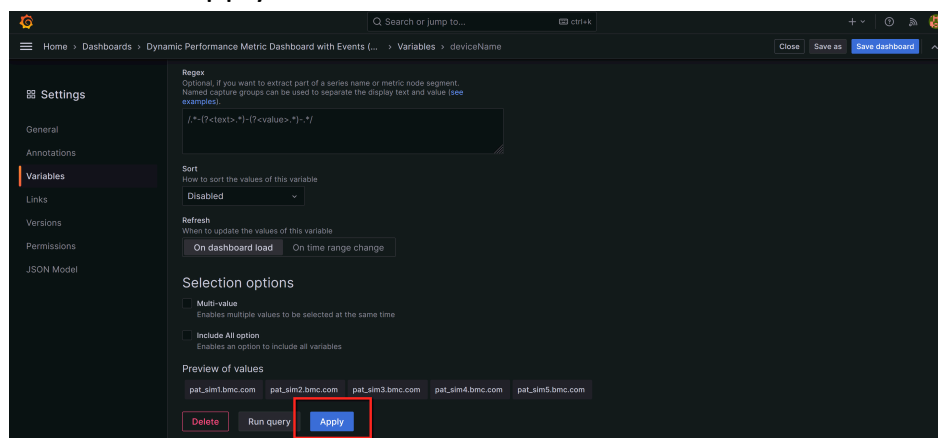
NOTE: “Multi-value” and “Include All option” is not supported yet and should be switched off! Also, value tags are NOT supported yet!

- Select the “Run query” Button to preview the result including Query and RegEx.



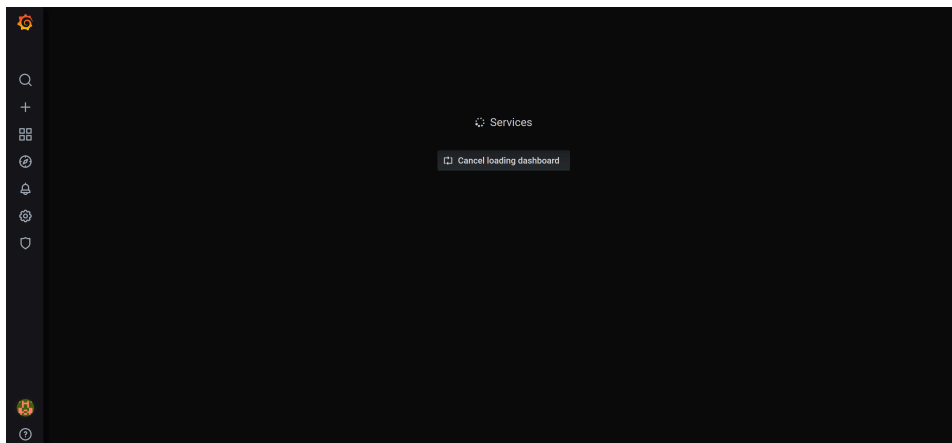
agentHostName Variable with Preview

- Select the “Apply” button.



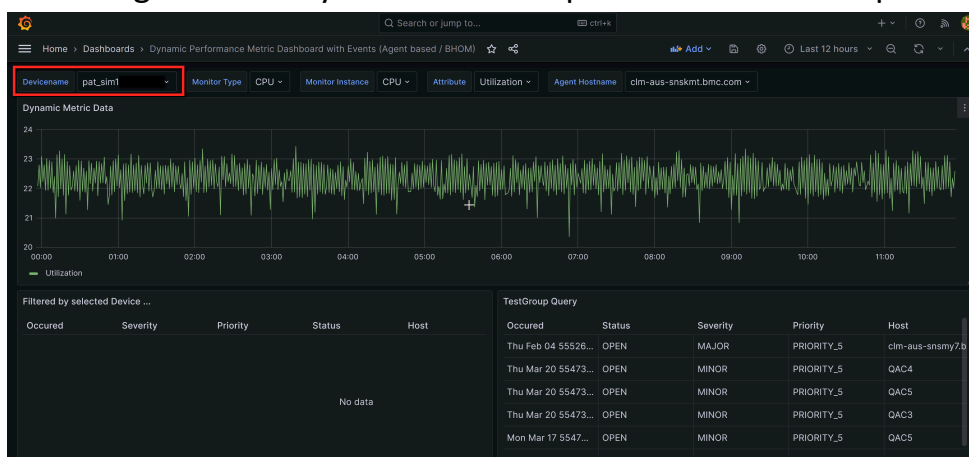
Save Dashboard after adding a Variable

- Open the Dashboard.



Dashboard loading data for Variable = Drop-Down list

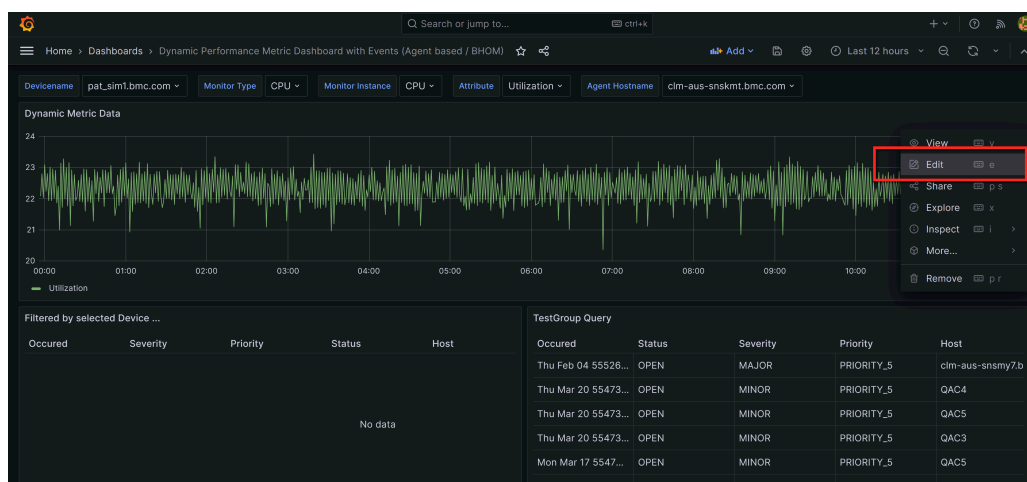
- You will see a message that the “Services” are loading which means the REST call to fill the Drop-Down list with values is currently executing. After this you find the Drop-Down list in the top left.



Dynamic Dashboard with Drop-Down list

NOTE: The Panels in the Dashboard are still static because they have a fixed Agent Hostname in the Device name input field. To finally make the Dashboard dynamic we must modify all Panels to use the value from the Drop-Down list.

- Open one Panel after the other by selecting the three dots in the title of the Panel and select “Edit” from the menu.



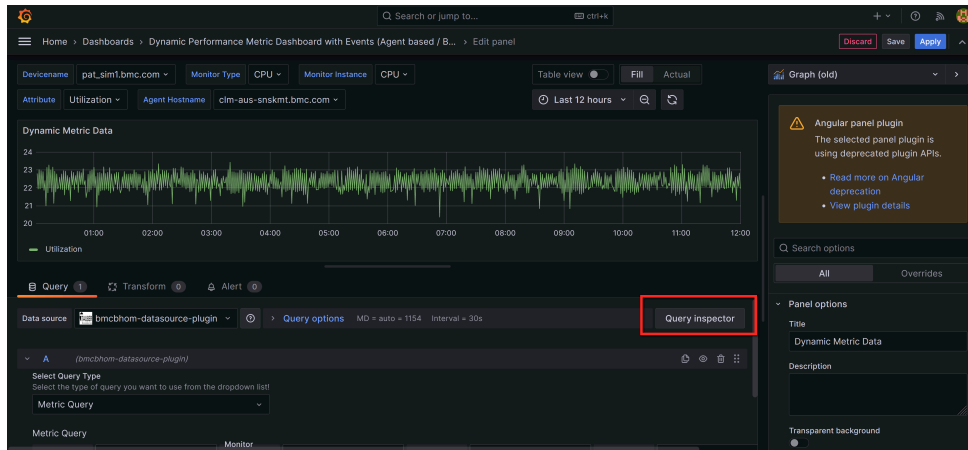
Editing a Panel in the Dashboard

- At the bottom of the screen scroll down to the Metric Definition Input fields and replace the static Device name with the Variable definition “`{agentHostName}`” which tells Grafana to use the value from the Drop-Down list.



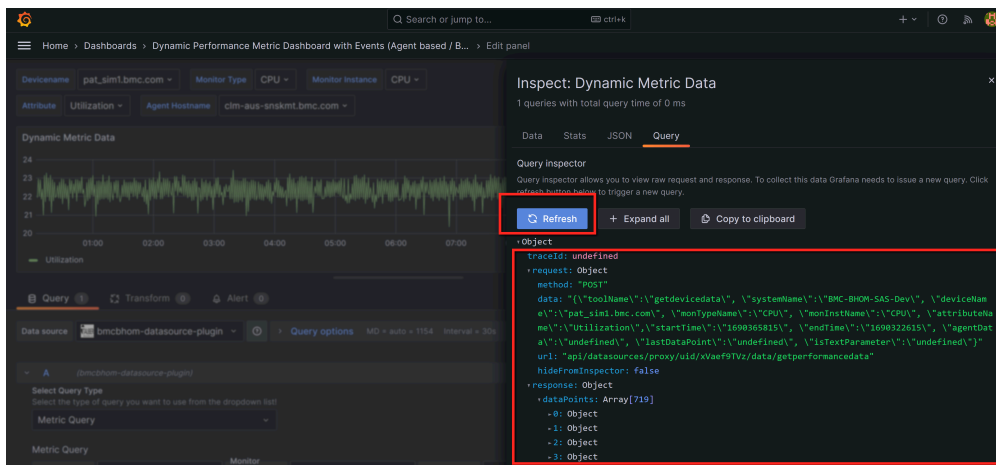
Query Editor with agentHostName Variable

- Scroll from the Query Editor (At the bottom of the screen) to the top to see the “Query inspector” button.



Query Inspector button

- Press the “Query inspector” button to check if the Query with the variable works.



Query Inspector with Query result

Check the result of the Query. For Metric data you should see a “dataPoints” Array with several entries containing a timestamp and a value. If you get an error, check the “agentHostName” Variable and if the Monitor Type, Instance and Attribute name exist on the selected Device(s).

For more change to the “Troubleshooting” section of this manual!

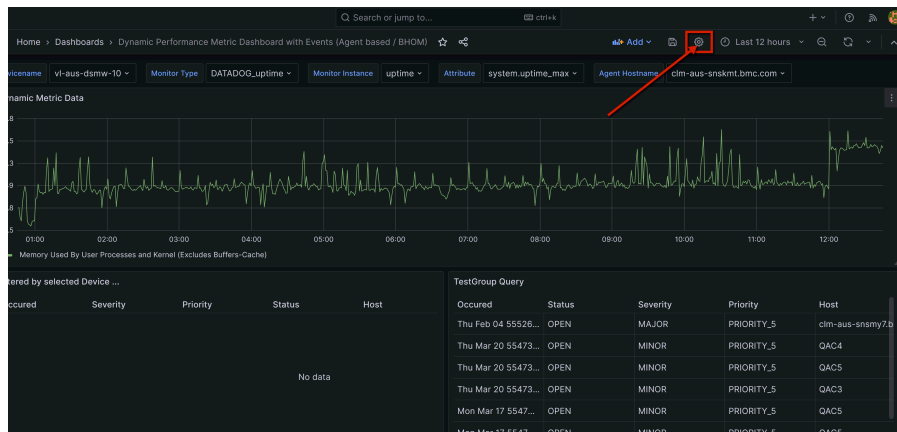
The monitorTypeName variable

The monitorTypeName Variable is used to create a Drop-Down List with Monitor Types of a specific Device in BHOM.

Note: We will show how to use this Variable as Chained Variable in a Performance Metric Dashboard available for Download in the Dashboard Package on the TAUBER Download pages.

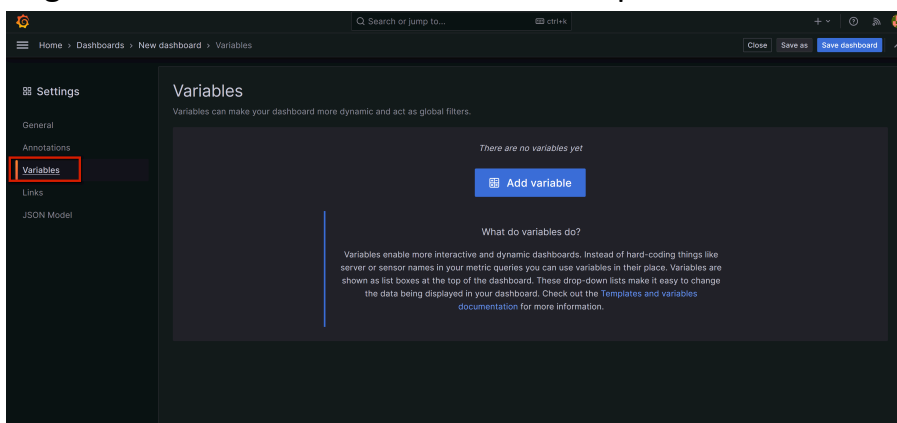
To create this Variable, follow the steps below:

- Open the Dashboard you want to add the Grafana Variable to and enter the Settings of the Dashboard.



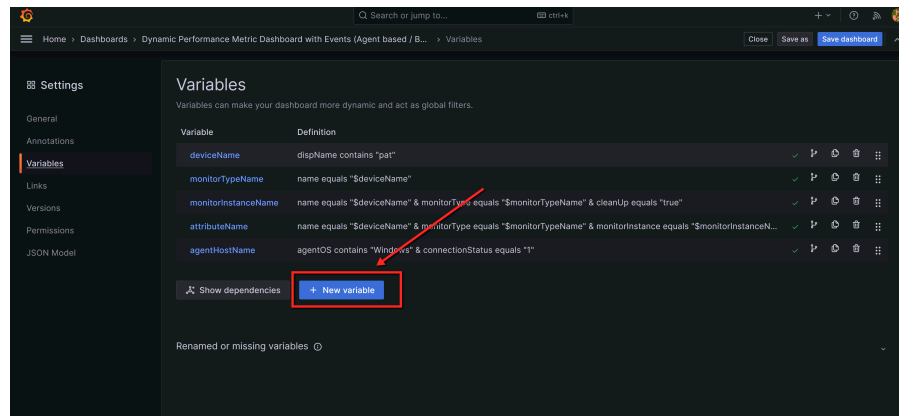
Dynamic Dashboard with Settings Button

- Change to the “Variables” Section in the top left corner of the menu.



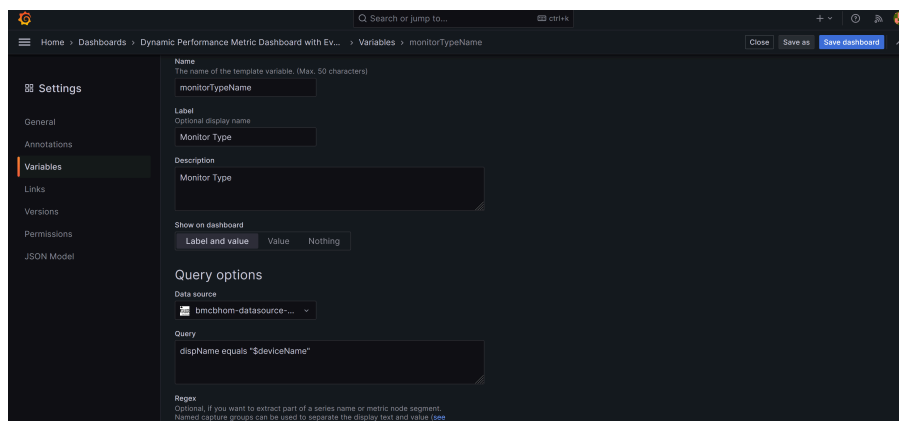
Dashboard Settings Variable Section

- Click the “Add variable” button.



Dashboard with Variable overview

- In the Name field you **MUST** enter monitorTypeName. Type must be “Query”.



Dashboard with monitorTypeName variable

- The Label and Description can be configured the way you want. The Label is the text next to the Drop-Down list in the Dashboard. The Description will appear when you hover the Drop-Down list.
- In the “Query Options” you must select the TAUBER Grafana Data Source PlugIn as “Data source”. I suggest selecting “On Dashboard Load” for Refresh. This will make the Dashboard update the Device list every time you enter the Dashboard.

- In the Query field you can use a simple Query to pass a Device to the system. Use
dispName equals “[Device name]”

Replace [Device name] with the device you want to use. You can also pass the value of another Variable into this Query e.g., use

\$agentHostName or \$deviceName e.g,
dispName equals “\$deviceName”

IMPORTANT:

When you use Chained Variables in Grafana you reference another Variable using the notation \$[Variable name] NOT \${[Variable name]} like in the Query Editor. **The Syntax is different!**

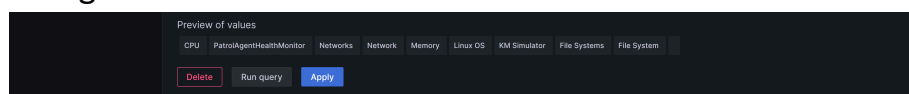
Note: The mechanism in Grafana to use an existing Variable in the Query of another Variable is called “Chained Variables”. Make sure that the Variables have the right order because Grafana will process the Variables from the Top to the Bottom.

Read more about that in the chapter “Chained variables”.

- Next to the Query field you can use RegEx in the RegEx field to extract Monitor Types applying the RegEx to the resulting Monitor Types!
- Select a Sort option from the Drop-Down list to apply sorting.

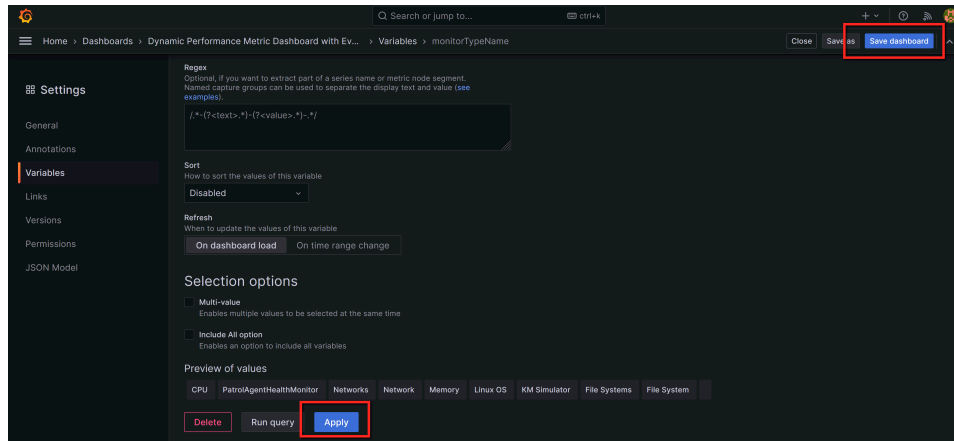
NOTE: “Multi-value” and “Include All option” is not supported yet and should be switched off! Also, value tags are NOT supported yet!

- Select the “Run query” Button to preview the result including Query and RegEx.



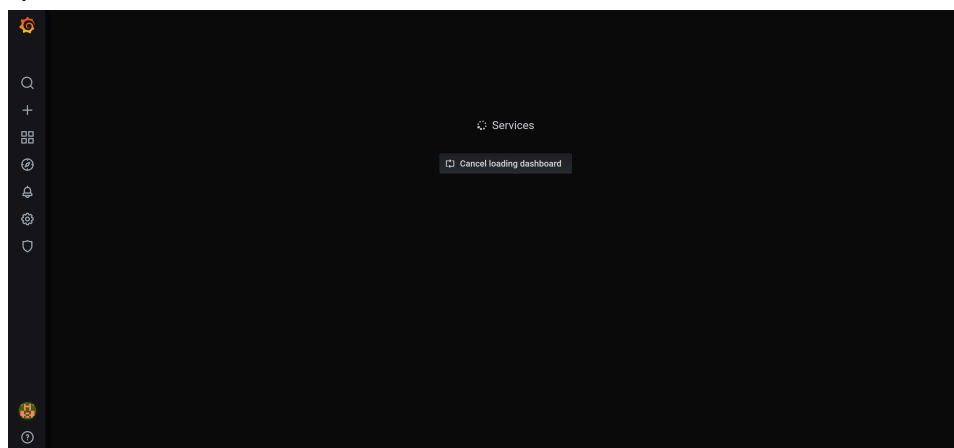
monitorTypeName Variable with Preview

- Select the “Apply” button and “Save dashboard”.



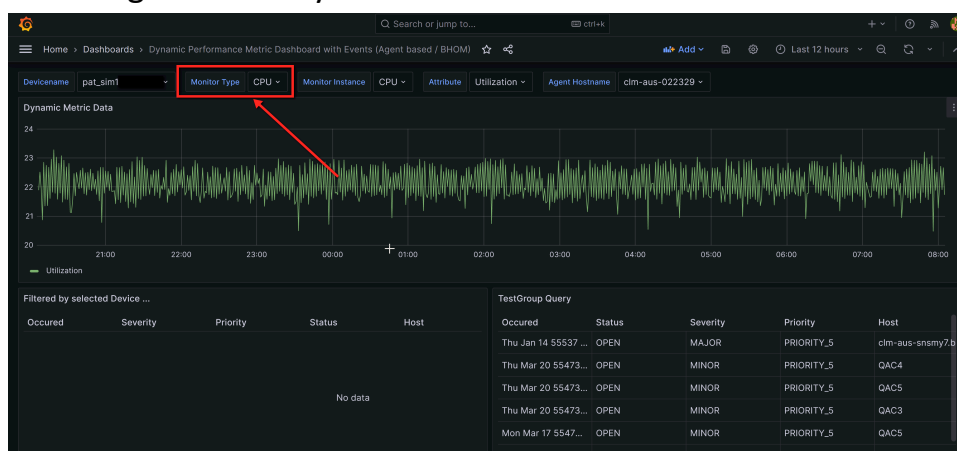
Save Dashboard after adding a Variable

- Open the Dashboard.



Dashboard loading data for Variable = Drop-Down list

- You will see a message that the “Services” are loading which means the REST call to fill the Drop-Down list with values is currently executing. After this you find the

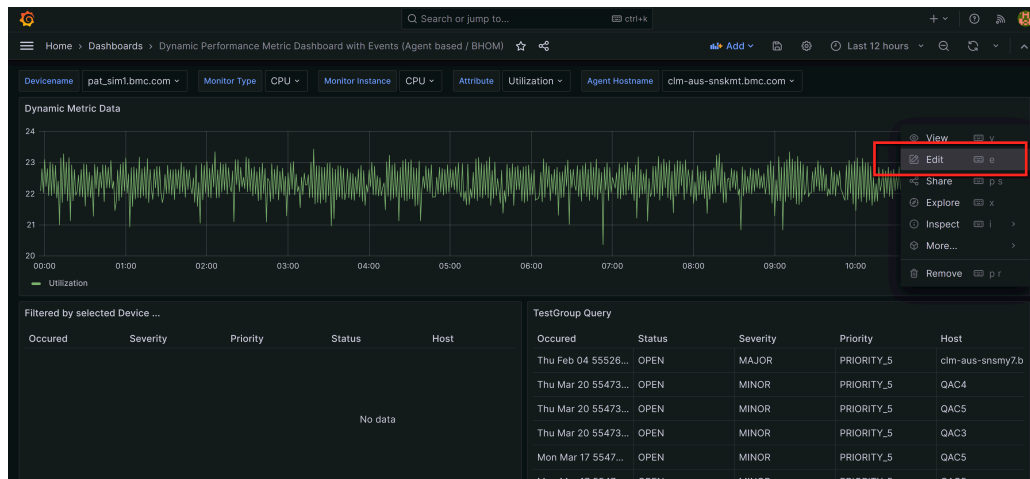


Dynamic Dashboard with Drop-Down list

NOTE: The Panels in the Dashboard still have a static Monitor Type because they have a fixed Monitor Type name in the Monitor Type

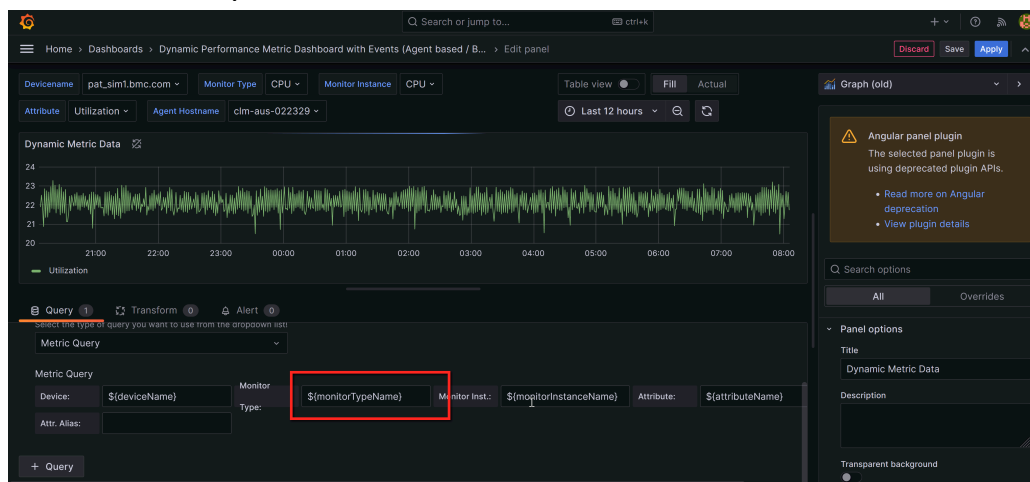
input field. To finally make the Dashboard dynamic we must modify all Panels to use the value from the Drop-Down list.

- Open one Panel after the other by selecting the three dots in the title of the Panel and select “Edit” from the menu.



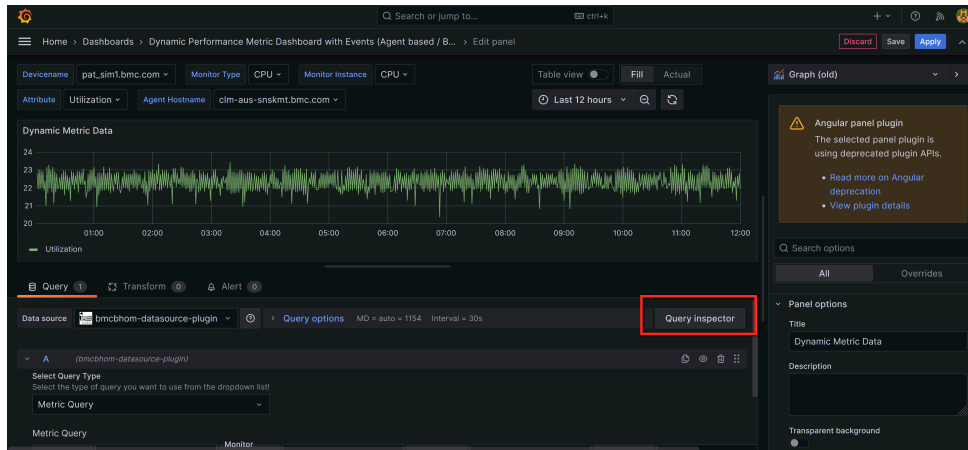
Editing a Panel in the Dashboard

- At the bottom of the screen scroll down to the Metric Definition Input fields and replace the static Monitor Type name with the Variable definition “\${monitorTypeName}” which tells Grafana to use the value from the Drop-Down list.



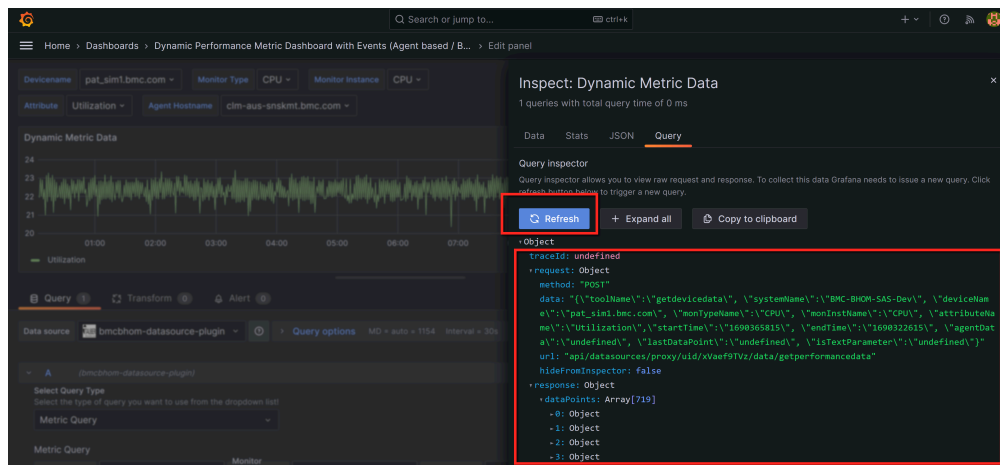
Query Editor with monitorTypeName Variable

- Scroll from the Query Editor (At the bottom of the screen) to the top to see the “Query inspector” button.



Query Inspector button

- Press the “Query inspector” button to check if the Query with the variable works.



Query Inspector with Query result

Check the result of the Query. For Metric data you should see a dataPoints array with several entries containing a timestamp and a value. If you get an error, check all the Variables and if the Monitor Type, Instance and Attribute name exist on the selected Device.

For more change to the “Troubleshooting” section of this manual!

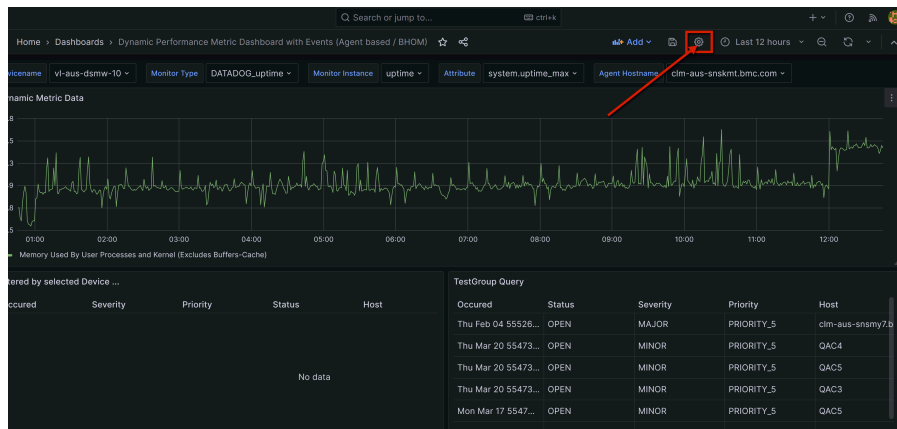
The monitorInstanceName variable

The monitorInstanceName Variable is used to create a Drop-Down List with Monitor Instances of a specific Device and Monitor Type in BHOM.

Note: We will show how to use this Variable as Chained Variable in a Performance Metric Dashboard available for Download in the Dashboard Package on the TAUBER Download pages.

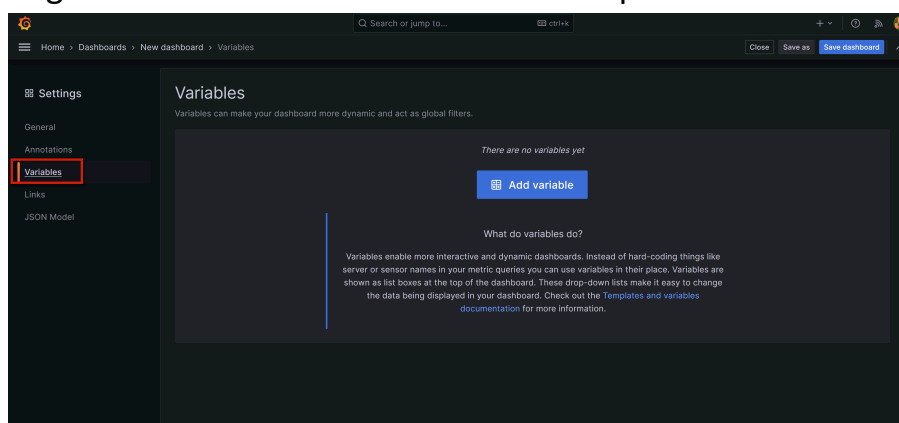
To create this Variable, follow the steps below:

- Open the Dashboard you want to add the Grafana Variable to and enter the Settings of the Dashboard.



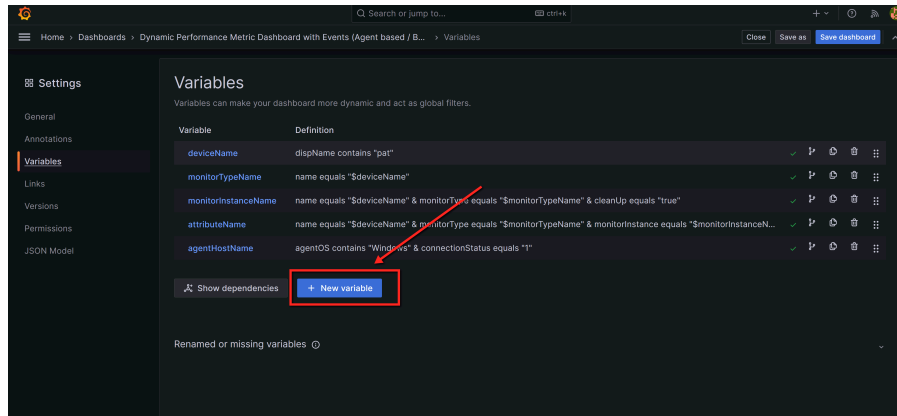
Dynamic Dashboard with Settings Button

- Change to the “Variables” Section in the top left corner of the menu.



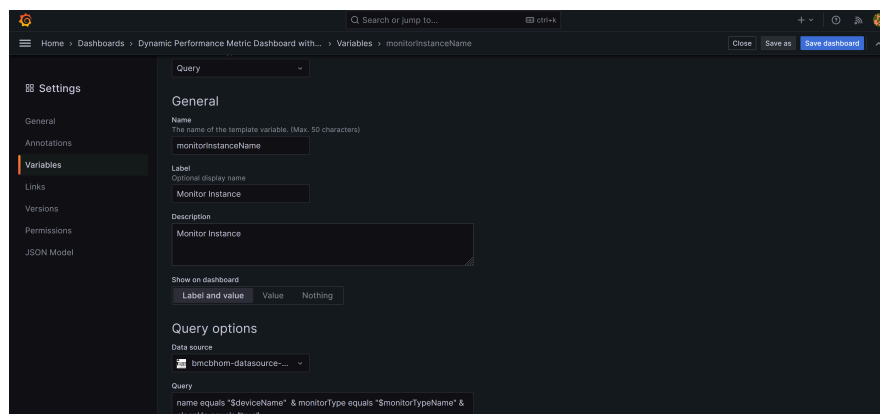
Dashboard Settings Variable Section

- Click the “Add variable” button.



Dashboard with Variable overview

- In the Name field you **MUST** enter monitorInstanceName. The type must be “Query”



Dashboard with monitorInstanceName variable

- The Label and Description can be configured the way you want. The Label is the text next to the Drop-Down list in the Dashboard. The Description will appear when you hover the Drop-Down list.
- In the “Query Options” you must select the TAUBER Grafana Data Source PlugIn as “Data source”. I suggest selecting “On Dashboard Load” for Refresh. This will make the Dashboard update the Device list every time you enter the Dashboard.

- In the Query field you can use a simple Query to pass a Device and Monitor Type to the system and select whether you want to clean up the Monitor Instance name.

dispName equals “[Device name]” & monitorType equals “[Monitor Type name]” & cleanUp equals “[false|true]”

Replace [Device name] with the device you want to use and “Monitor Type Name” with the Monitor Type name you want to have the instances for. You can also pass the value of another Variable into this Query e.g., use

*\$agentHostName, \$deviceName or \$monitorTypeName e.g,
dispName equals “\$deviceName” & monitorType equals
“\$monitorTypeName” & cleanUp equals “true”*

What does cleanUp mean?

In BHOM internally the Monitor Instance name is an Object path containing the parent and child folders the instance is nested in. See the Filesystem instance below:

Linux OS/Storage/FileSystem//sys/kernel/debug

If you select “false” for the cleanUp Option, this will be the result for the Filesystem. Setting cleanUp to “true” will remove the Object path and just show the “/sys/kernel/debug” as Instance name.

IMPORTANT:

- You must use the same setting for “cleanUp” in the monitorInstanceName and attributeName Variable if you chain them. Otherwise, the attributeName Variable cannot find the instance and will throw an Error.
- When you use Chained Variables in Grafana you reference another Variable using the annotation `$(Variable name)` NOT `${[Variable name]}` like in the Query Editor. **The Syntax is different!**

NOTE: The mechanism in Grafana to use an existing Variable in the Query of another Variable is called “Chained Variables”. Make sure

that the Variables have the right order because Grafana will process the Variables from the Top to the Bottom.

Read more about that in the chapter “Chained variables”.

- Next to the Query field you can use RegEx in the RegEx field to extract Monitor Types applying the RegEx to the resulting Monitor Types!
- Select a Sort option from the Drop-Down list to apply sorting.

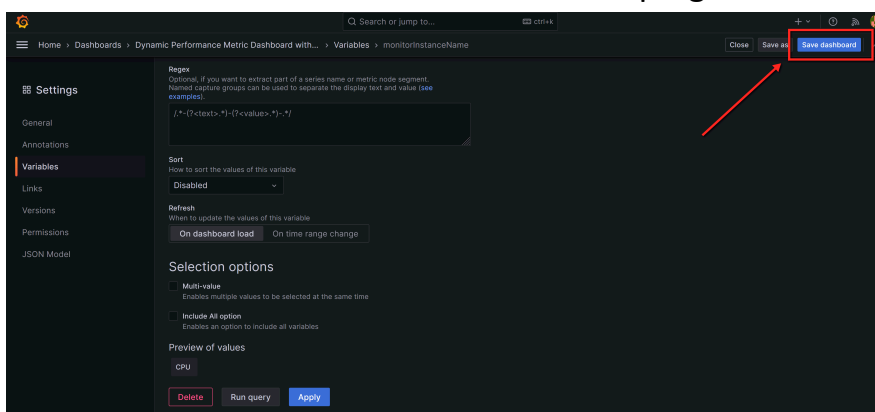
Note: “Multi-value” and “Include All option” is not supported yet and should be switched off! Also, value tags are NOT supported yet!

- Select the “Run query” Button to preview the result including Query and RegEx.



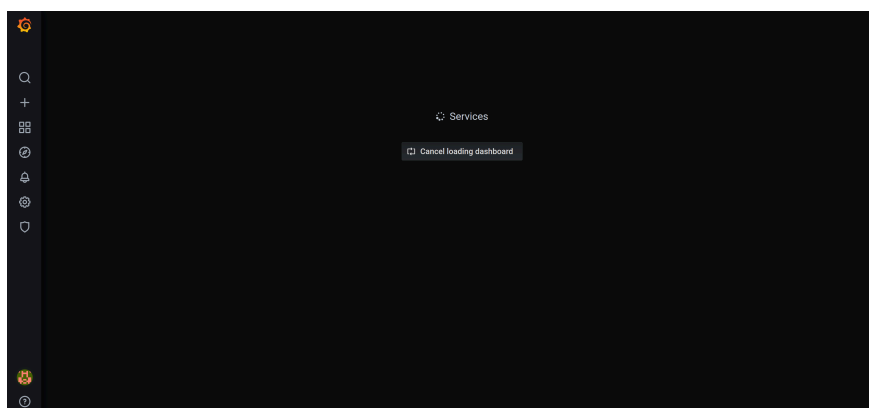
Preview of the monitorInstanceName Variable with cleanUp equals true

- Select the “Save Dashboard” button in the top right corner.



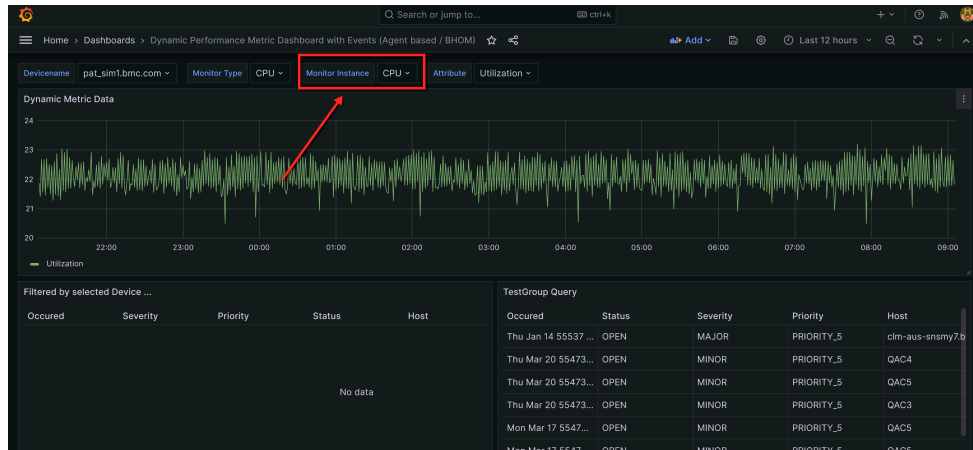
Save Dashboard after adding a Variable

- Open the Dashboard.



Dashboard loading data for Variable = Drop-Down list

- You will see a message that the “Services” are loading which means the REST call to fill the Drop-Down list with values is currently executing. After this you find the instances drop-down list at the top of the Dashboard.

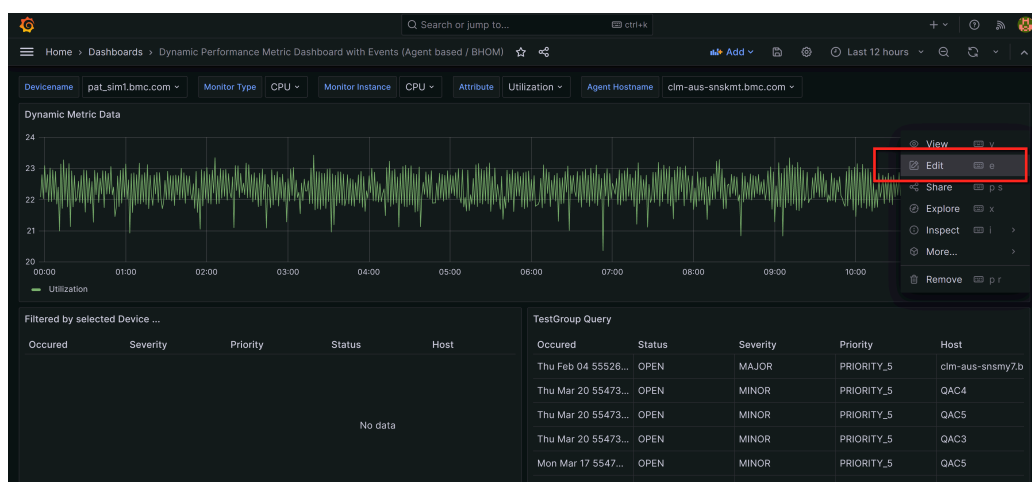


Dynamic Dashboard with Drop-Down lists

Note: The Panels in the Dashboard still have a static Monitor Instance because they have a fixed Monitor Instance name in the Monitor Type Instance field.

To finally make the Dashboard dynamic we must modify all Panels to use the value from the Monitor Instance Drop-Down list.

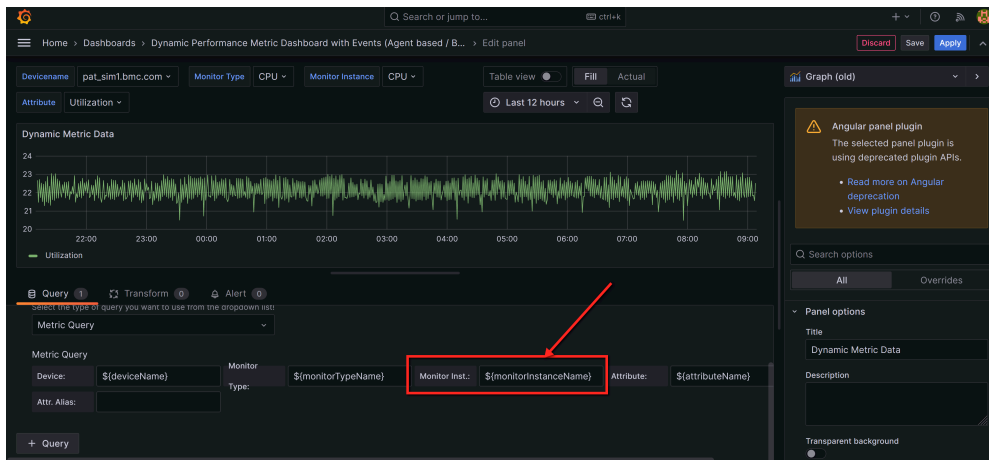
- Open one Panel after the other by selecting the three dots in the title of the Panel and select “Edit” from the menu.



Editing a Panel in the Dashboard

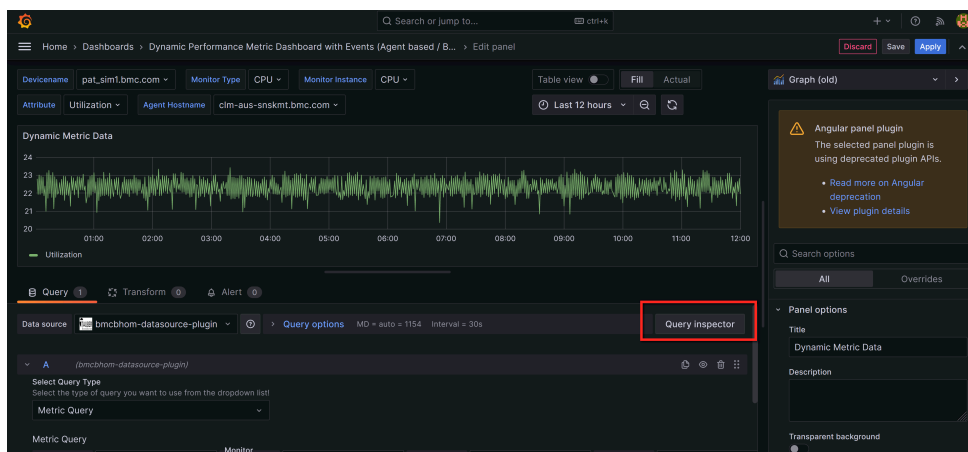
- At the bottom of the screen scroll down to the Metric Definition Input fields and replace the static Monitor Instance name with the Variable

definition “`${monitorInstanceName}`” which tells Grafana to use the value from the Drop-Down list.



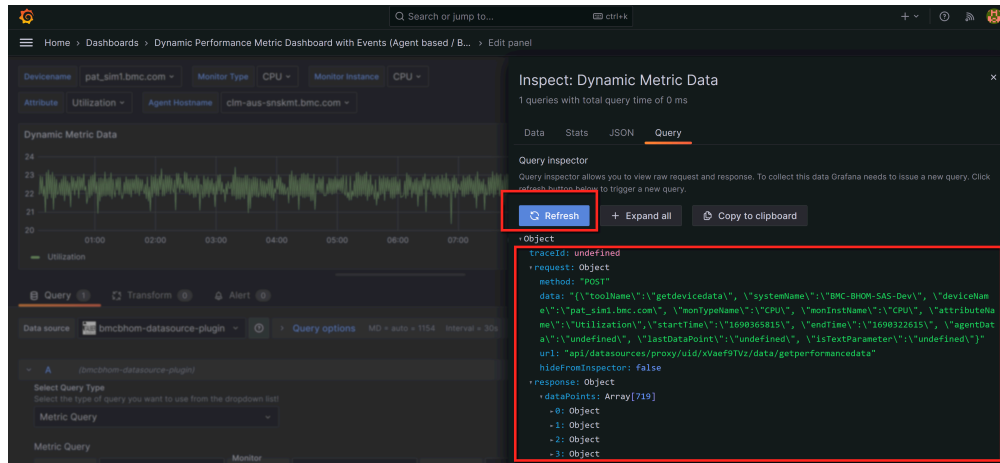
Query Editor with monitorInstanceName Variable

- Scroll from the Query Editor (At the bottom of the screen) to the top to see the “Query inspector” button.



Query Inspector button

- Press the “Query inspector” button to check if the Query with the variable works.



Query Inspector with Query result

- Check the result of the Query. For Metric data you should see a dataPoints array with several entries containing a timestamp and a value. If you get an error, check all the Variables and if the Monitor Type, Instance and Attribute name exist on the selected Device.

For more change to the “Troubleshooting” section of this manual!

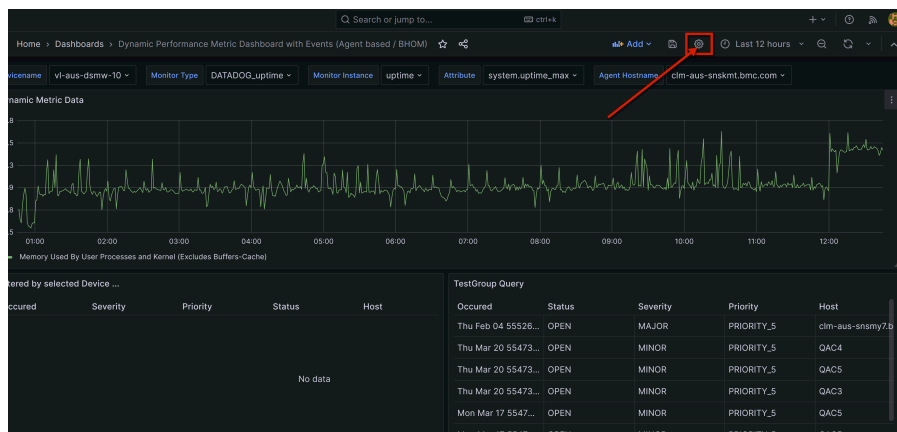
The attributeName and attributeName2 variable

The attributeName Variable is used to create a Drop-Down List with Attribute Names of a specific Device, Monitor Type and Monitor Instance in BHOM.

Note: We will show how to use this Variable as Chained Variable in a Performance Metric Dashboard available for Download in the Dashboard Package on the TAUBER Download pages.

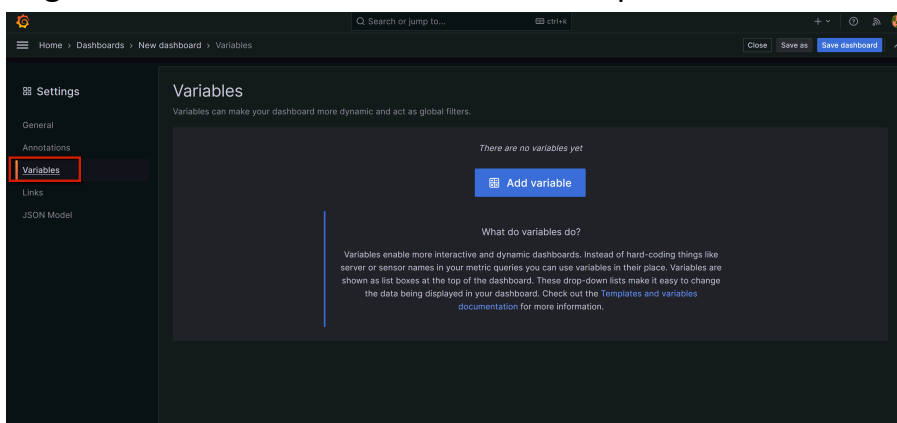
To create this Variable, follow the steps below:

- Open the Dashboard you want to add the Grafana Variable to and enter the Settings of the Dashboard.



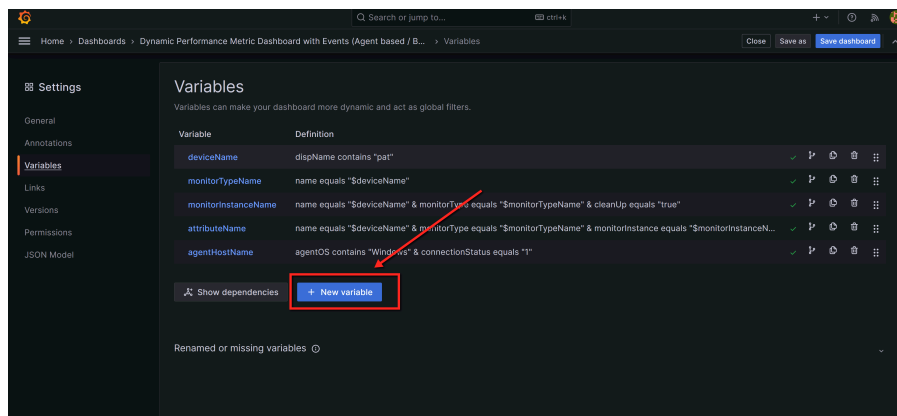
Dynamic Dashboard with Settings Button

- Change to the “Variables” Section in the top left corner of the menu.



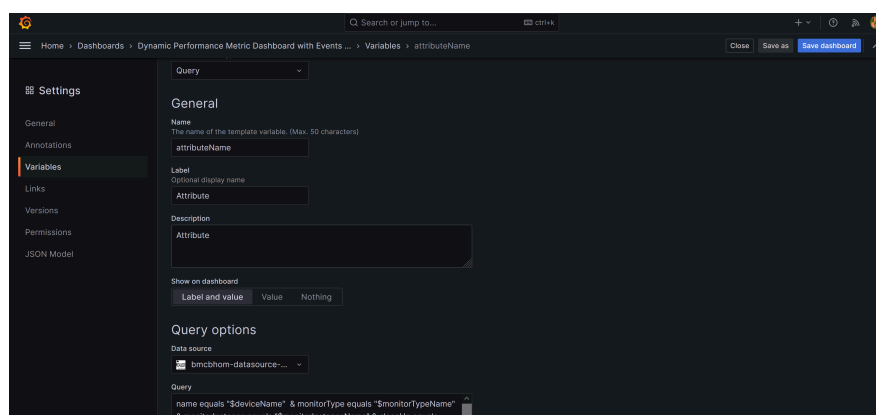
Dashboard Settings Variable Section

- Click the “Add variable” button.



Dashboard with Variable overview

- In the Name field you MUST enter “attributeName”. The type must be “Query”



Dashboard with attributeName variable

- The Label and Description can be configured the way you want. The Label is the text next to the Drop-Down list in the Dashboard. The Description will appear when you hover the Drop-Down list.
- In the “Query Options” you must select the TAUBER Grafana Data Source PlugIn as “Data source”. I suggest selecting “On Dashboard Load” for Refresh. This will make the Dashboard update the Device list every time you enter the Dashboard.

- In the Query field you can use a simple Query to pass a Device, Monitor Type and Monitor Instance to the system and select whether you use the cleanUp option on the Monitor Instance name.

dispName equals "[Device name]" & monitorType equals "[Monitor Type name]" & monitorInstanceName equals "[Monitor Instance Name]" & cleanUp equals "[false|true]" & excludeTextParameters equals "[true| false]"

Replace [Device name] with the device, [Monitor Type Name] with the Monitor Type name and [Monitor Instance Name] with the Monitor Instance you want to have the Attributes for. You can also pass the value of another Variable into this Query e.g., use

\$agentHostName, \$deviceName or \$monitorTypeName,
\$monitorInstanceName e.g,
dispName equals "\$deviceName" & monitorType equals "\$monitorTypeName" & monitorInstanceName equals "\$monitorInstanceName" & cleanUp equals "true"& excludeTextParameters equals "true"

What does cleanUp mean?

In BHOM internally the Monitor Instance name is an Object path containing the parent and child folders the instance is nested in. See the Filesystem instance below:

Linux OS/Storage/FileSystem//sys/kernel/debug

If you select "false" for the cleanUp Option, this will be the result for the Filesystem. Setting cleanUp to "true" will remove the Object path and just show the "/sys/kernel/debug" as Instance name.

IMPORTANT:

- You must use the same setting for "cleanUp" in the monitorInstanceName and attributeName Variable if you chain them. Otherwise, the attributeName Variable cannot find the instance and will throw an Error.

- When you use Chained Variables in Grafana you reference another Variable using the annotation `$(Variable name)` NOT `{{[Variable name]}}` like in the Query Editor. **The Syntax is different!**

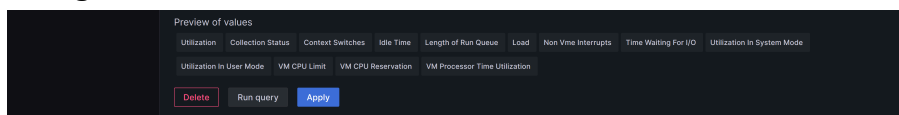
Note: The mechanism in Grafana to use an existing Variable in the Query of another Variable is called “Chained Variables”. Make sure that the Variables have the right order because Grafana will process the Variables from the Top to the Bottom.

Read more about that in the chapter “Chained variables”.

- Next to the Query field you can use RegEx in the RegEx field to extract Monitor Types applying the RegEx to the resulting Monitor Types!
- Select a Sort option from the Drop-Down list to apply sorting.

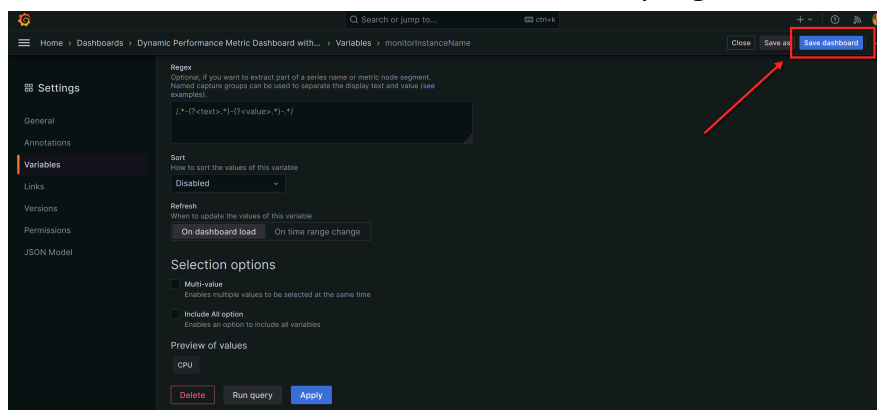
Note: “Multi-value” and “Include All option” is not supported yet and should be switched off! Also, value tags are NOT supported yet!

- Select the “Run query” Button to preview the result including Query and RegEx.



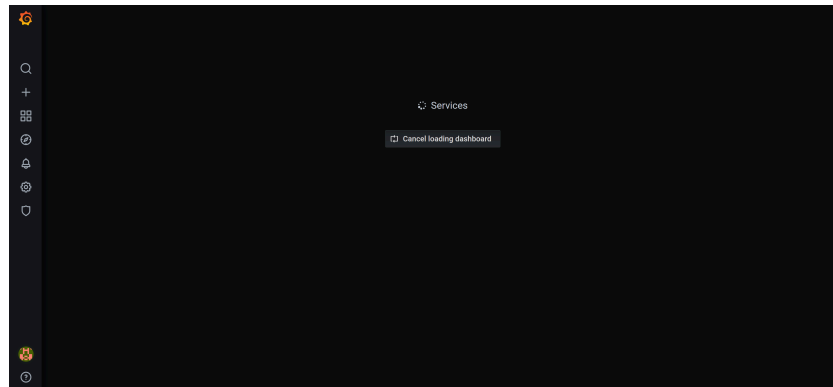
Preview of the attributeName Variable

- Select the “Save Dashboard” button in the top right corner.



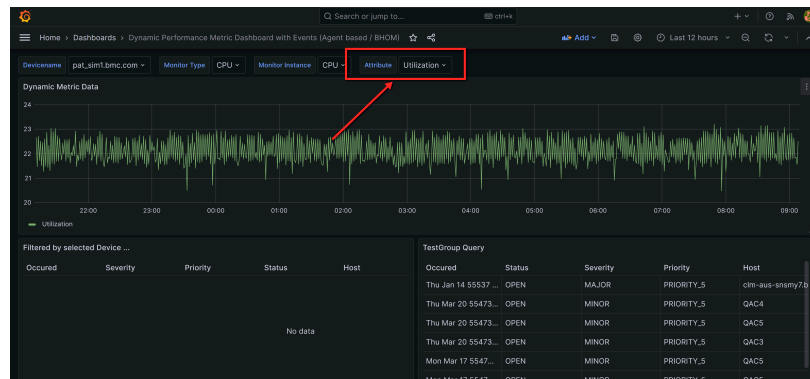
Save Dashboard after adding a Variable

- Open the Dashboard



Dashboard loading data for Variable = Drop-Down list

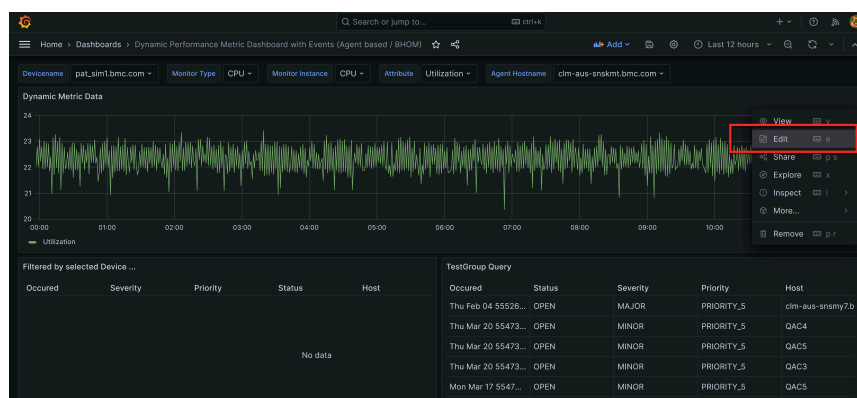
- You will see a message that the “Services” are loading which means the REST call to fill the Drop-Down list with values is currently executing. After this you find the Attributes drop-down list at the top of the Dashboard.



Dynamic Dashboard with Drop-Down lists

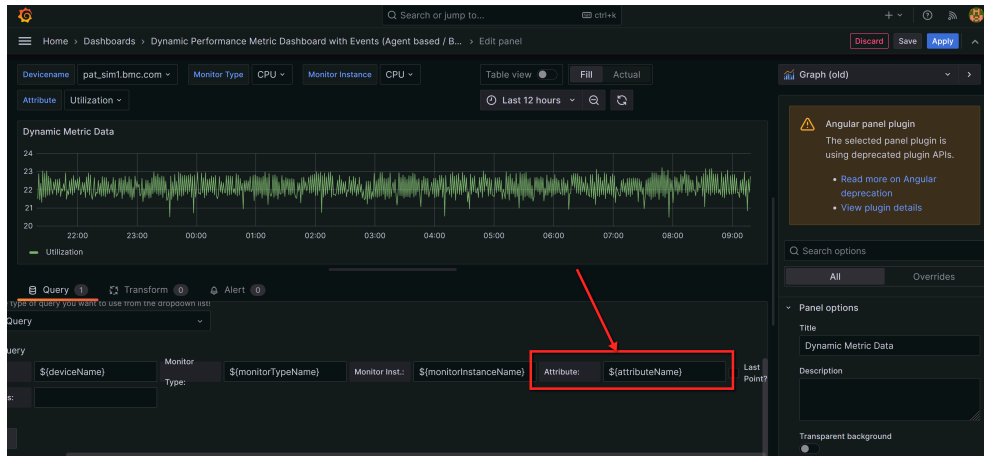
Note: The Panels in the Dashboard still have a static Attribute name because they have a fixed Attribute name in the Attribute field. To finally make the Dashboard dynamic we must modify all Panels to use the value from the Attribute Drop-Down list.

- Open one Panel after the other by selecting the three dots in the title of the Panel and select “Edit” from the menu.



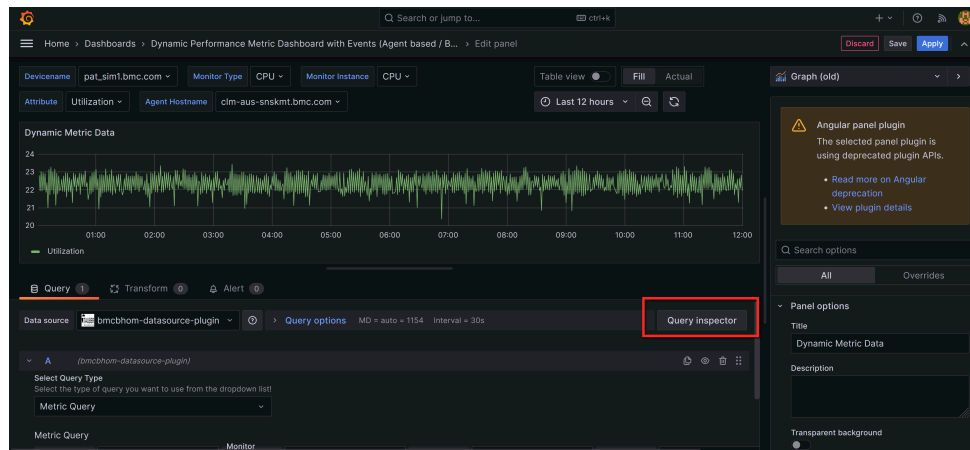
Editing a Panel in the Dashboard

- At the bottom of the screen scroll down to the Metric Definition Input fields and replace the static Attribute name with the Variable definition “`${attributeName}`” which tells Grafana to use the value from the Drop-Down list.



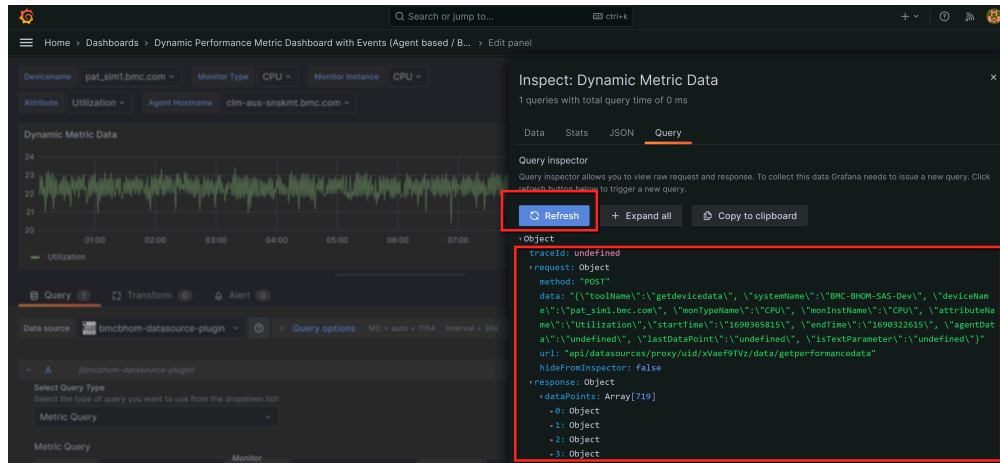
Query Editor with attributeName Variable

- Scroll from the Query Editor (At the bottom of the screen) to the top to see the “Query inspector” button.



Query Inspector button

- Press the “Query inspector” button to check if the Query with the variable works.



Query Inspector with Query result

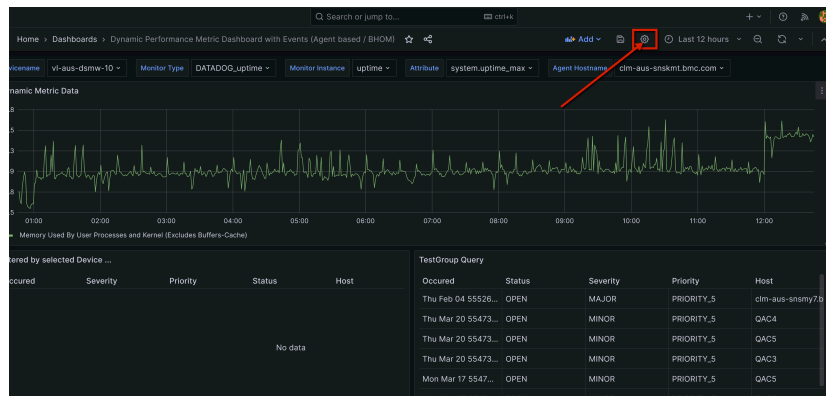
Check the result of the Query. For Metric data you should see a dataPoints array with several entries containing a timestamp and a value. If you get an error, check all the Variables and if the Monitor Type, Instance and Attribute name exist on the selected Device.

For more change to the “Troubleshooting” section of this manual!

The templateVariable1 to 100 variables

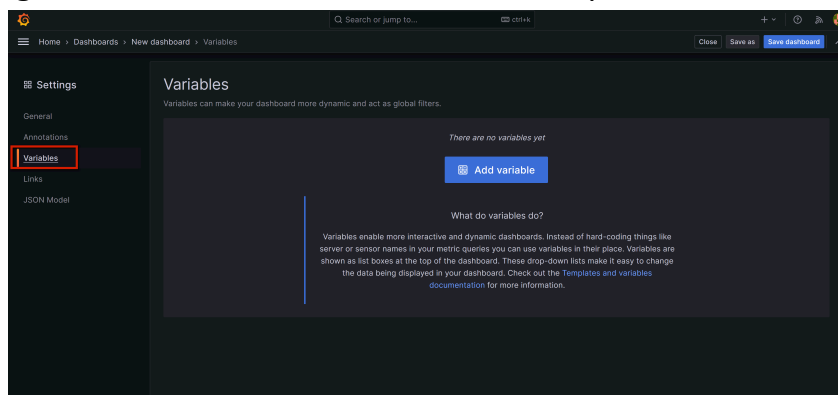
Currently the TAUBER Grafana PlugIn supports up to one hundred Template Variables (templateVariable1 to 100) for custom usage like static or list value. Below you find an example how to use the Variables for a Custom Remote Monitor.

- Open the Dashboard you want to add the Grafana Variable to and enter the Settings of the Dashboard.



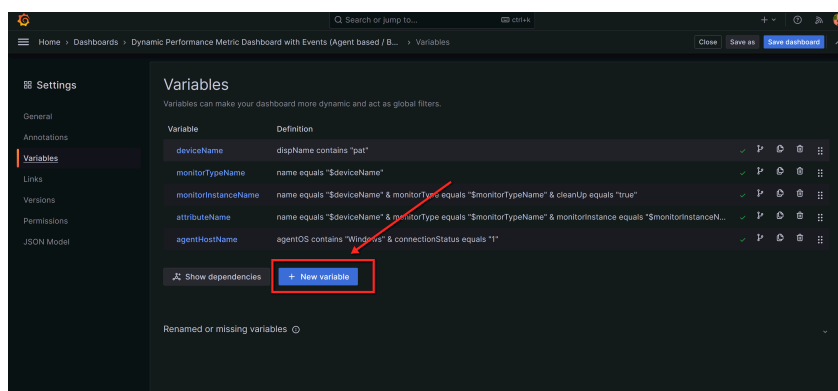
Dynamic Dashboard with Settings Button

- Change to the “Variables” Section in the top left corner of the menu.

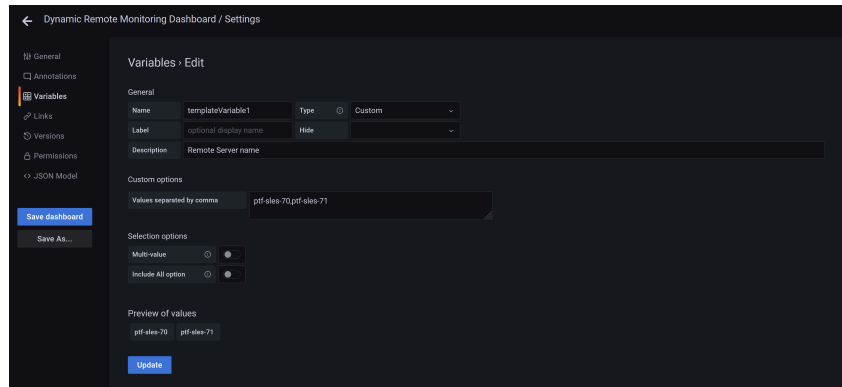


Dashboard Settings Variable Section

- Click the “Add variable” button.

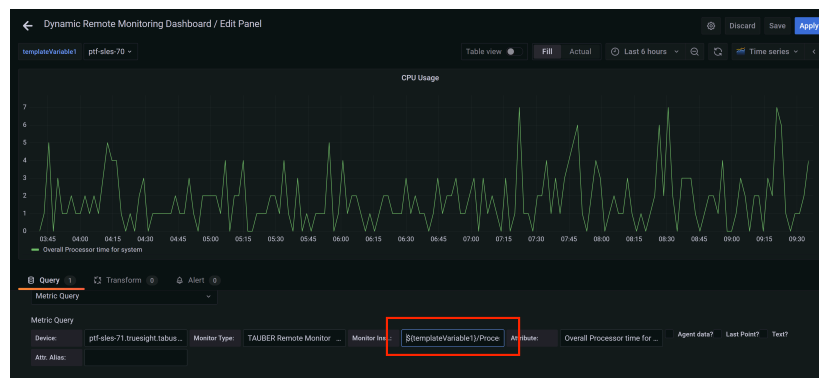


Dashboard with Variable overview



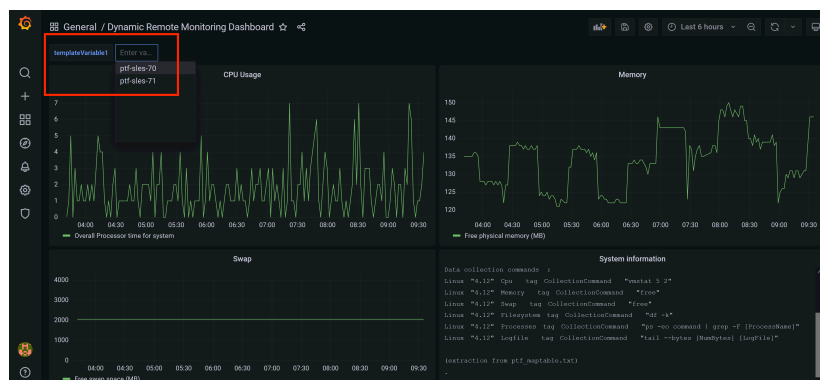
Dashboard Settings Variable Section

- In the name field you **MUST** enter templateVariable1 to templateVariable100. Currently a max of one hundred variables is set!
- As an example, you can switch the type to “Custom” and enter comma separated values like hostnames into the appropriate field.



Template Variable in Monitor Instance name

You can use the value of the Template Variable in every query by entering `$(templateVariable1)`. In the above example we use it in a Monitor Instance name of a remote monitor. The values can be selected as always in the top left corner of the Dashboard.



Template Variable selection

You can use the value of the Template Variable also as chained Variable. See the next chapter.

IMPORTANT

Please make sure that you use the values one after the other (1, 2, 3 ...) and not randomly spread because the PlugIn will stop replacing the template Variable with the selected value as soon as it finds an empty variable, e.g.

```
templateVariable1 = "host1,host2"  
templateVariable2 = "host3,host4"  
templateVariable4 = "host5,host6"
```

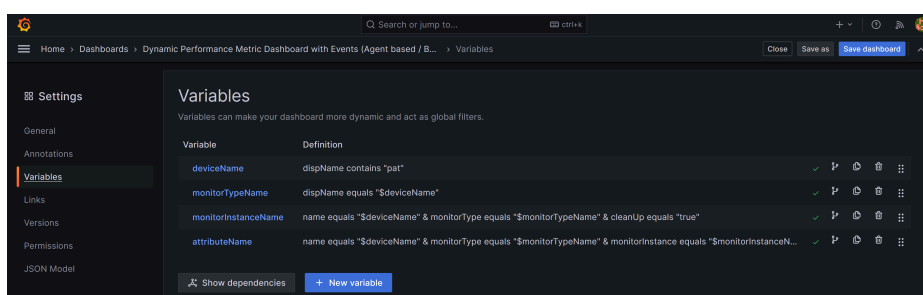
The PlugIn will process the first two variables but NOT variable templateVariable4. To make that work use it the following way:

```
templateVariable1 = "host1,host2"  
templateVariable2 = "host3,host4"  
templateVariable3 = "host5,host6"
```

Chained variables

Chained variables in a Grafana Dashboard are Variables that use other Variables in their own Query. Open your Dashboard and change to the Variables section.

In the example below we have four different Variables. The deviceName Variable is the only native not chained Variable because it does not have another Variable in its Query.



Chained Variables

The Variables are processed from Top to the Bottom so make sure all Variables you want to use are available correctly. If you want to reference a Variable in the Query of another Variable you have to use the “\${[Variable name]}” e.g., “\$deviceName” annotation which differs from the Query Editor annotation which is “\${[Variable name]}” e.g., “\${deviceName}”.

If you have trouble creating chained Variables look at the Example Dashboards available for Download on the TAUBER Grafana Integration Download page on the TAUBER Website.

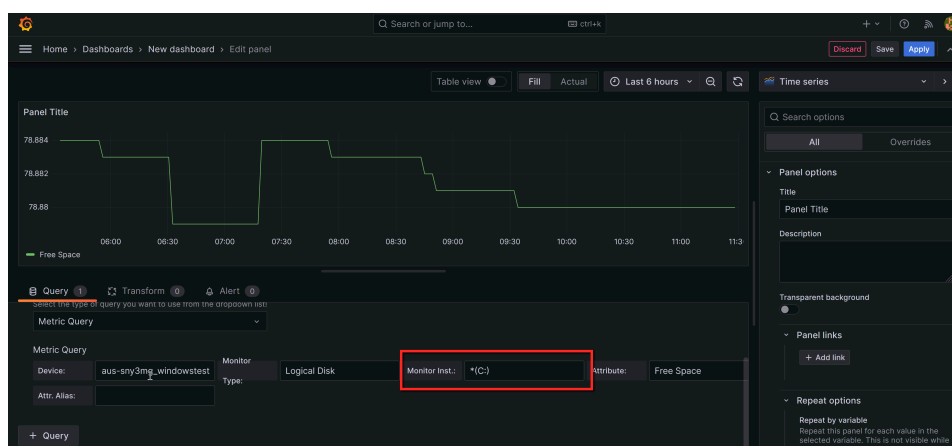
The Asterix(*) for Monitor instance names

Sometimes Monitor instance names have Device or Agent specific names so creating a Dynamic Dashboard might be difficult because the name is different on all instances.

To be able to create these kinds of Dashboards you can use the Asterix in the Query Editor for the Monitor Instance (only!). Below you find an example for Windows Logical Disk Monitor instances.

Let's say you have Windows Servers with Logical Disk D: drives with different names like "Data (D:)" and "Backup (D:)". You want to create a Dashboard for all Servers showing the Free space of the D drive.

- Go to the Query Editor of your Panel
- Enter the following expression for the Monitor instance name: "* (D:)"



Query Editor with Asterix (*) expression for instance

When the PlugIn Proxy receives the request from the PlugIn, he will check all instances of this Monitor Type for a match ignoring the String before the "(D:)". When he has a match, he gathers and returns the data for this Monitor instance to the PlugIn.

- Check the Query in the Query Inspector

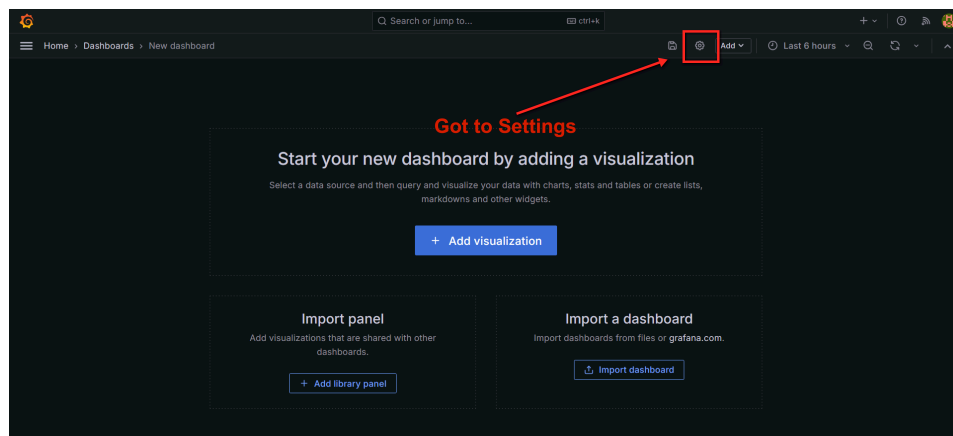
Creating dynamic Dashboards with the “Dynamic Metric Query”

To create a dynamic Dashboard is also possible with the Dynamic Metric Query. Although the Dynamic Metric Query does not support all possible kinds of dynamic Dashboards the Metric Query supports a lot of Use cases can be covered. We will show to different Use cases and how you can quickly create a dynamic Dashboard.

Use case 1: Windows Agent CPU Performance Dashboard

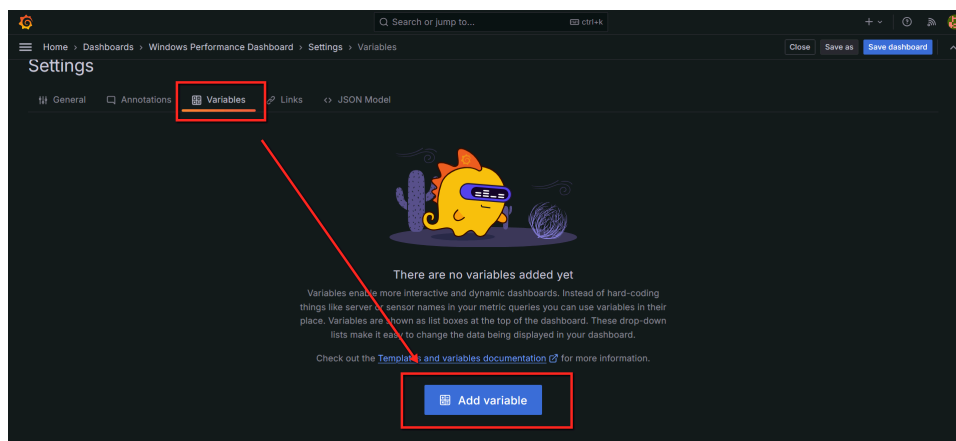
We want to create a Dashboard where we can select a Windows Agent from a Dropdown list updating the major CPU Performance metrics in the Dashboard.

1. Create a new Dashboard and. Add a Visualization



New Dashboard

2. Go to the settings and then to the Variables Section of the new Dashboard. IMPORTANT: Before adding Panels, you should create all Variables you need in the Panel!

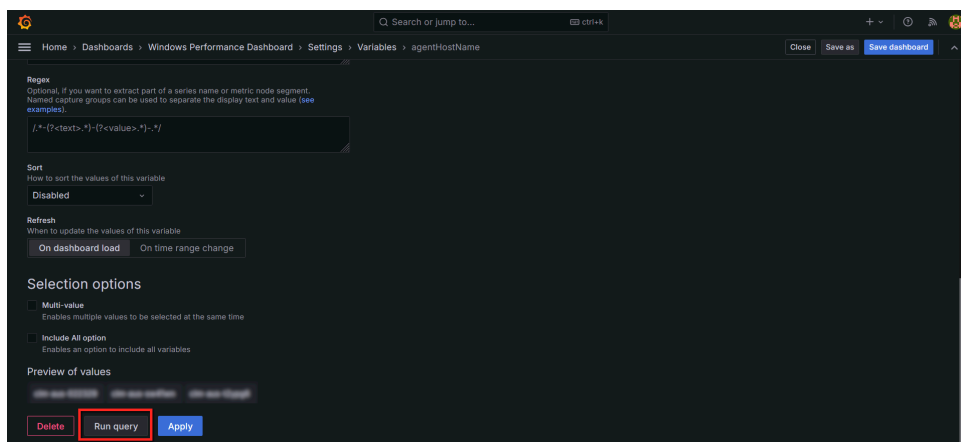


Add Variable

3. To add the agentHostName Variable enter or select:

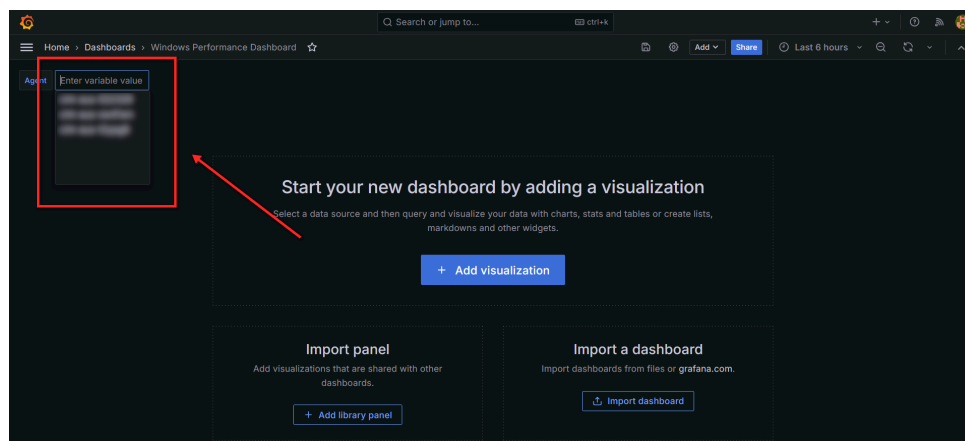
Variable type: Query
 Name: agentHostName
 Label name: Agent
 Data source: Select the BHOM Data source
 Query: agentOS contains "Windows"

4. Run the Query and check the result.



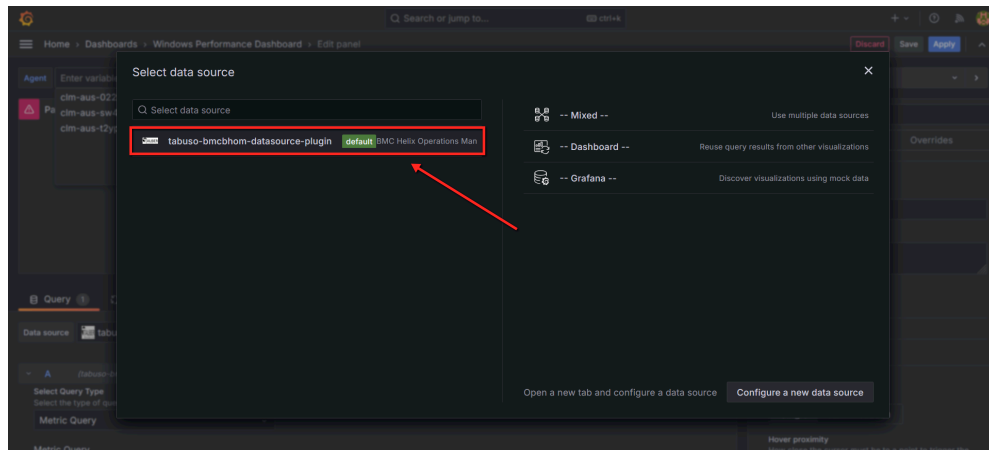
agentHostName Variable with Result

5. Apply the Variable and go back to the Dashboard.



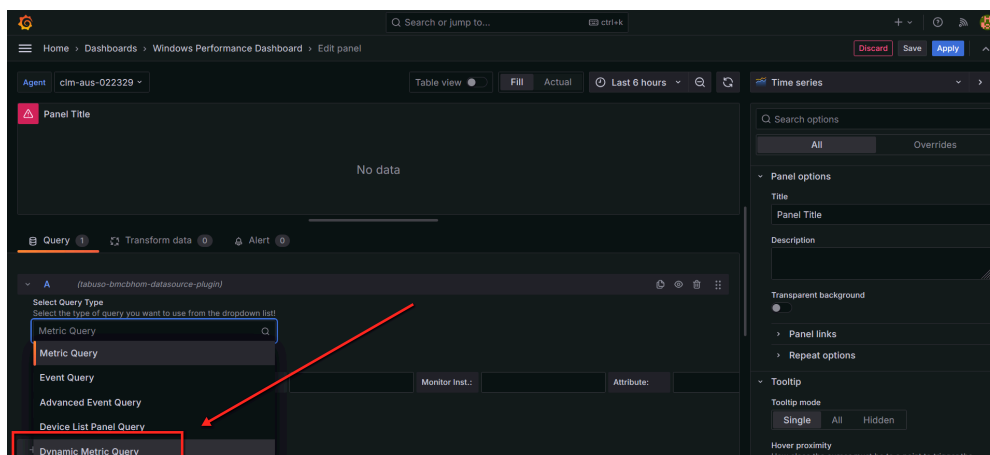
Check the Dropdown Box in the top left corner

6. Add a Visualization and select the TAUBER BHOM Data source.



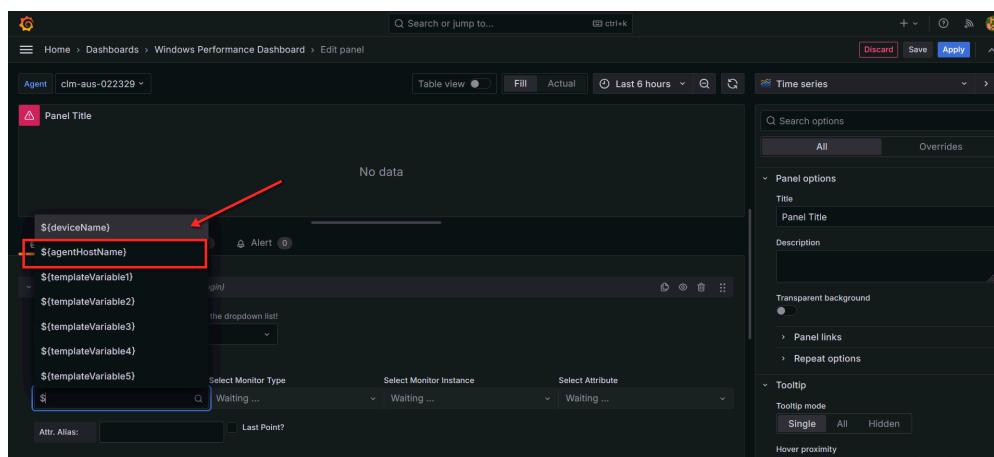
Select the TAUBER BHOM Data source

7. Select the Dynamic Metric Query from the Metric Type Dropdown Box.



Dynamic Metric Query

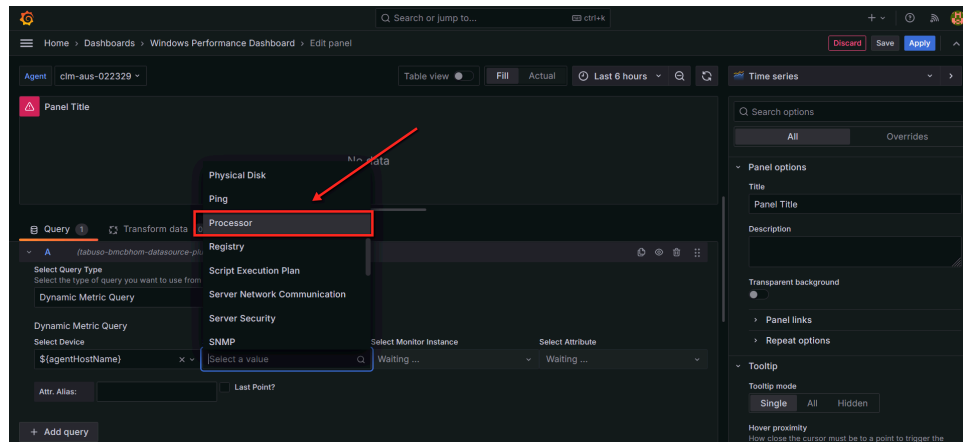
8. Select the $\${agentHostName}$ Variable from the Device Dropdown list.



$\${agentHostName}$ Variable as Device

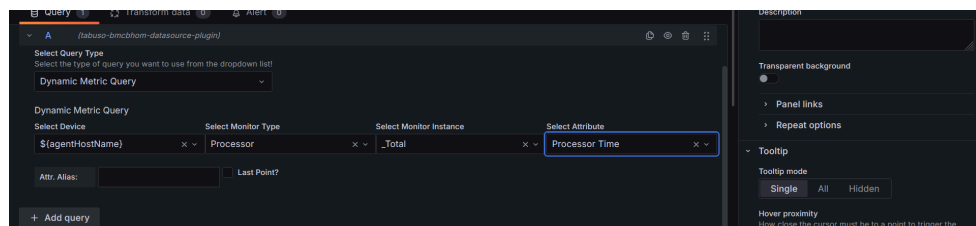
The Query will use the Device from the Agent in the next steps of the configuration process and later in the Dashboard.

9. Configure the rest of the Query by selecting the values from the Dropdown list.



``${agentHostName}`` Variable as Device

10. Use the following configuration for the Query.



Query configuration

11. Change the Panel title to “Processor Time” and save the Query and place the Panel on the Dashboard.

12. Add additional Panels with the Metrics “Privileged”, “User Time” and “Idle Time” and “Interrupts” and place the Panels on the Dashboard.

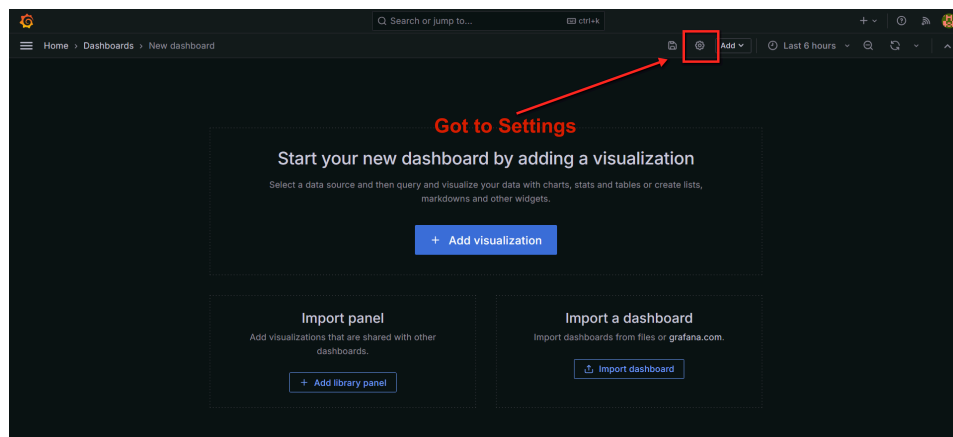


Final Dashboard

Use case 2: Windows Agent Logical Disk Used Space with Events for the Device.

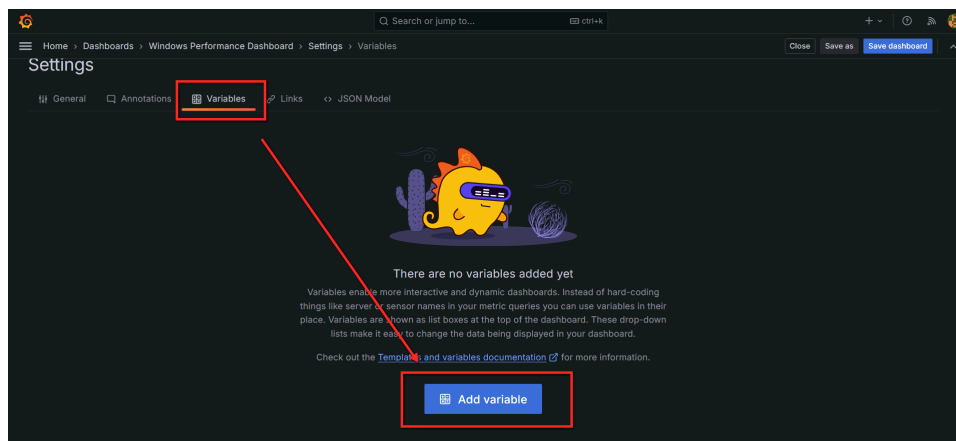
We want to create a Dashboard where we can select from the List of Windows Agents and second a Logical Disk. The Dashboard should show a Metric Panel for the current Used Space in Percent in the upper area of the Dashboard and a Table with the current open Events for the Device.

1. Create a new Dashboard and. Add a Visualization



New Dashboard

2. Go to the settings and then to the Variables Section of the new Dashboard. IMPORTANT: Before adding Panels, you should create all Variables you need in the Panel!



Add Variable

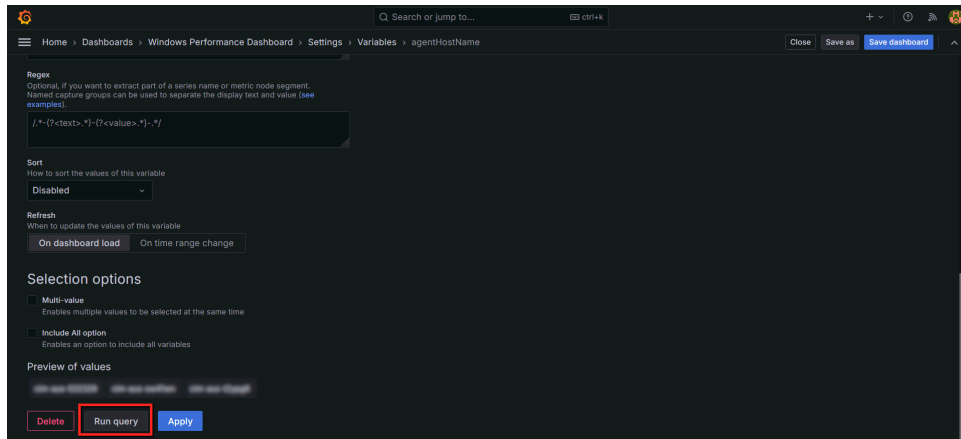
3. To add the agentHostName Variable enter or select:

Variable type: Query

Name: agentHostName

Label name: Agent
 Data source: Select the BHOM Data source
 Query: agentOS contains "Windows"

4. Run the Query and check the result.

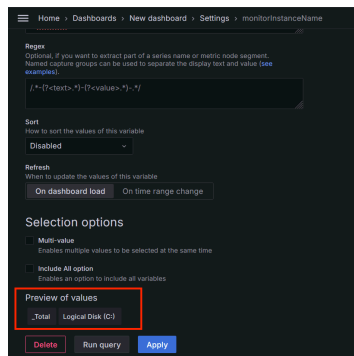


agentHostName Variable with Result

5. Apply the Variable and go back to the Variables section.

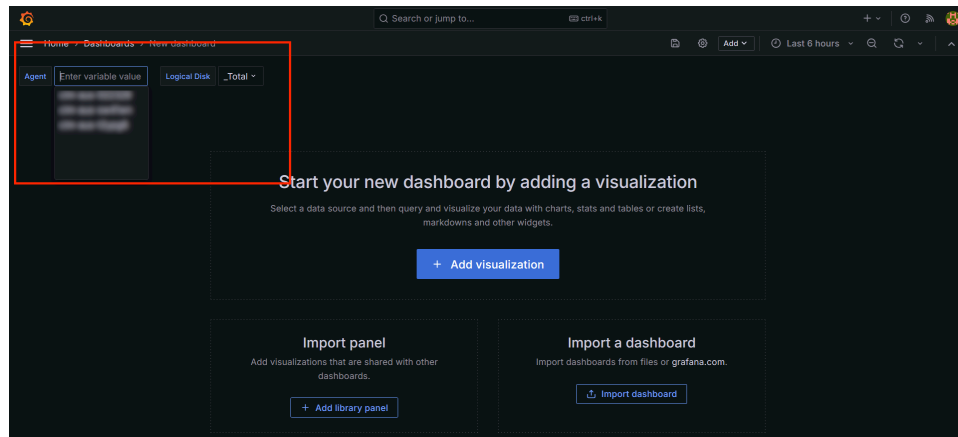
6. Select "New variable" again and add a Variable with the following values:

Variable type: Query
 Name: monitorInstanceName
 Label name: Logical Disk
 Data source: Select your BHOM Data source
 Query; dispName equals "\$agentHostName" & monitorType equals "Logical Disk" & cleanUp equals "true"



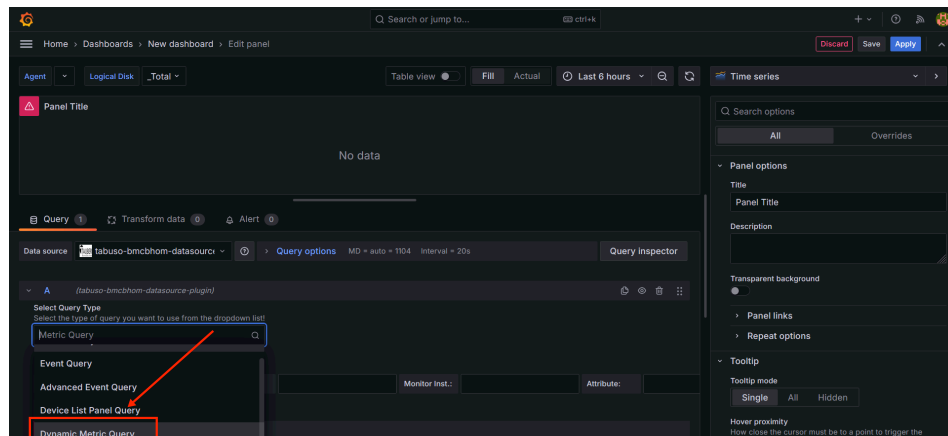
Run Query and check results

7. Return to the Dashboard screen.

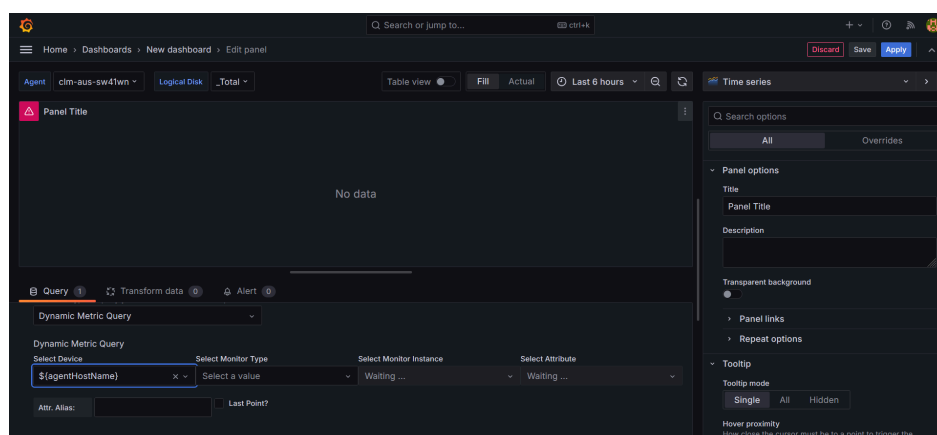


See the Variable data in the top left

8. Select an Agent that has the Metric you want to use and a Logical Disk.



Select the “Dynamic Metric Query”



Select “ $\${agentHostName}$ ” as Device

This will make the Visualization use the value from the Dropdown list as Device name.

9. Select “Logical Disk” as Monitor Type
10. Select “\${monitorInstanceName}” as Monitor Instance. This will make the Visualization use the value from the monitorInstanceName variable.
11. Select “Free Space” as Attribute.
12. Change the “Title” to “Free Space left”



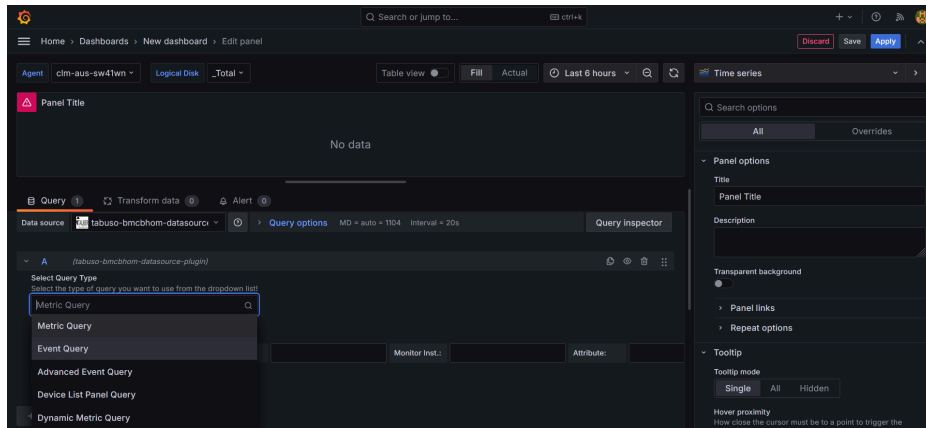
Run the Query and check the data

13. Apply the Visualization



Place it in the top area of your Dashboard

14. Add another “Visualization”
15. Select the “Event Query” from the Dropdown list



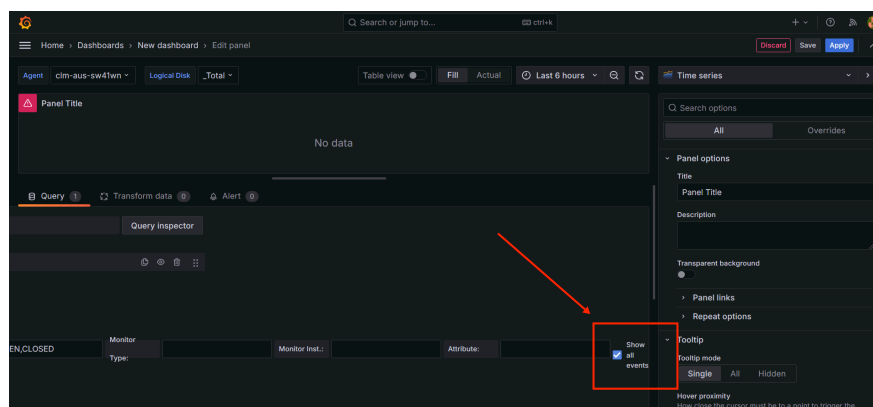
Select the “Event Query” from the Dropdown list

16. Add the following values

Device: \${agentHostName}

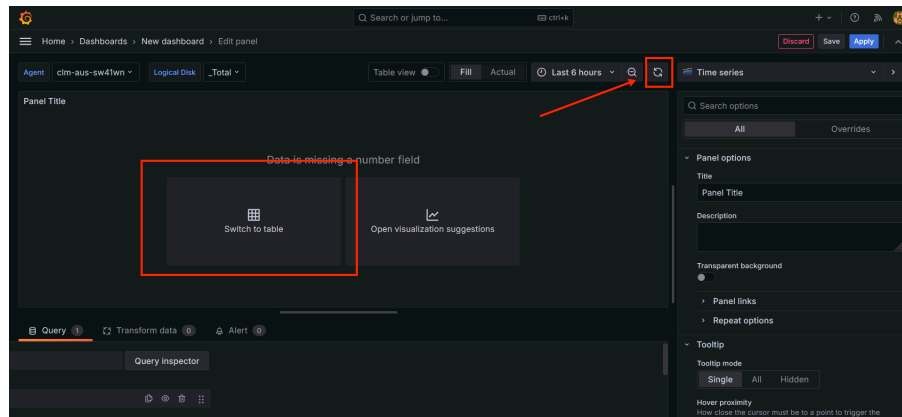
Event Sever.: CRITICAL,MAJOR,MINOR,WARNING,INFO

Event Status: OPEN,CLOSED



Make sure you select “Show all events”

This will make the Events show up independent from the timeframe selected in the Dashboard. Run the Query. Grafana will suggest to switch to a table view. Select this option!



Grafana suggesting table view!

17. Change the Panel Title to “Events for Device” and select “Apply”.



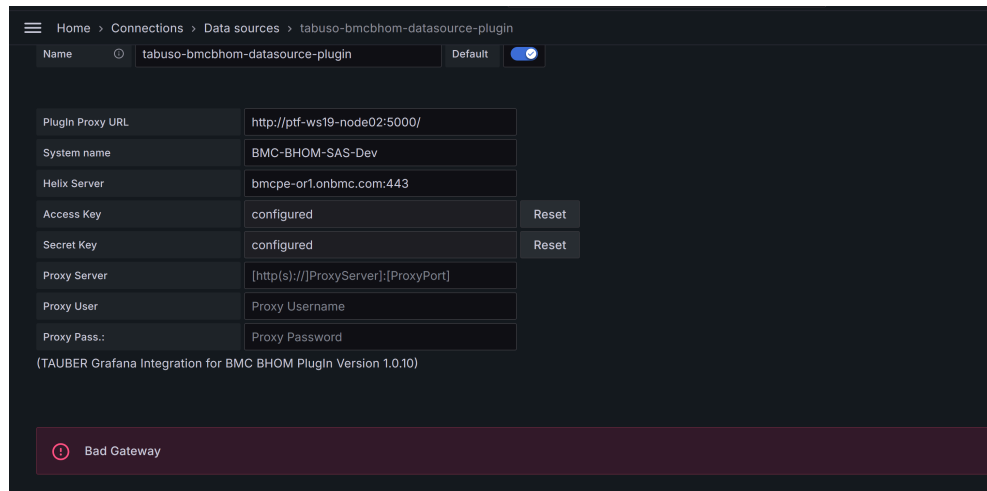
18. Move the Panel to the bottom

19. DONE!

Troubleshooting

The PlugIn cannot connect to Proxy

When you have problems connecting the PlugIn to the Proxy you should have a “Bad Gateway” error like the one below.



“Bad Gateway” error when configuring the Data Store PlugIn

1.) First, check if the Proxy is reachable from the Grafana Server.

- Open a Browser on the Grafana Server
- Enter the URL of the PlugIn Proxy e.g.,
[http\(s\)://\[PlugInProxyServerName\]:\[PlugInProxyPort\]/](http(s)://[PlugInProxyServerName]:[PlugInProxyPort]/)

You should see a page like the following:

```
TAUBER Grafana PlugIn Proxy 1.0.00 for BMC BHOM
(c) TAUBER Business Solutions UG
System status report
System : BMC-BHOM-SAS-Dev / Connection is OK at 27/7/2023 @ 11:30:11
```

Browser with PlugIn Status page

- If you cannot see the page, make sure you can access the URL and Port. Check the DNS for the URL and the Firewall for the Port.

2.) Start the PlugIn Proxy as Console App to check whether the connection to the Proxy is fine.

- Login to Server running the PlugIn Proxy
- Stop the Windows Service “TAUBER Grafana PlugIn Proxy”
- Make sure to set the debugLevel in conf/setup.properties to 0 to receive Debug output
- Open a Command Shell with Administrator privileges
- Change to the installation directory of the PlugIn Proxy
- Start the PlugIn Proxy using “bhPlugInProxy.exe”

```

DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : TAUBER Grafana PlugIn Proxy for BMC 1.0.0
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : © TAUBER Business Solutions US
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Start processing event slot definitions.
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Addes 7 event slots to event response definition.
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Finished processing event slot definitions.
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / httpPort = 4000
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / httpPort = 4000
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / certificateFile =
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / keyFile =
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / debugLevel = 0
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / logFile =
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / logFile =
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / log/debugLog_[dateAndTime].txt
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / maxInferSize = 10000
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / activateInMemoryBaseCaching = true
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / activateCacheForwarding = true
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / maxObjectCacheForwardingCacheSize = 30
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / showSystemOutMessages = true
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / maxNumberOfWorkers = 5000
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / eventSlotDefinition =
"grafana_tlb_resource", "status", "severity", "priority", "source_hostname", "class"
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / lastDataInRequestHeader = 3000
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / makeRotateLogFilePath = 5
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / ignoreProxyHostHeaderData = true
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / eventSeverityNameMapping = 03,20,30,40,50,60,70
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / showSystemOutMessages = true
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Configuration / addHttpTagsToHttpRequestOutput = true
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : Initialize HTTP Server
DEBUG (Thu Jul 27 2023 11:15:38 GMT+0200 (Mitteleuropäische Sommerzeit)) : HTTP Server running on port 4000
  
```

PlugIn Proxy as Console App with debugLevel set to 0

- Go to your TAUBER Grafana Data Source PlugIn, configure the connection to the Proxy and BHOM and click “Save & Test”

```

DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisSystemName = [MC-9436-545-509]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisProxyServer = [https://grafana.com:443]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisProxyServer = [no value]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisProxyName = [no value]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisProxyPassword = [REDACTED]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisProxyUser = [SET]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data / thisProxySecretKey = [SET]
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : Connection data complete. Trying to connect and store data!
DEBUG (Thu Jul 27 2023 11:16:12 GMT+0200 (Mitteleuropäische Sommerzeit)) : REST output is ok! Connection file created
  
```

PlugIn Proxy Debug with successfully connection

When you see the “Connection definition data” the connection between the PlugIn and the PlugIn Proxy is fine. When you see the “REST output is ok” also the connection to BHOM is fine!

[If you can't make it work anymore contact support@tabuso.com!!!](mailto:support@tabuso.com)

The Proxy cannot connect to BHOM

When the PlugIn cannot successfully connect to the BHOM system configured in the Data Source, but the connection to PlugIn Proxy is fine follow the steps below:

- 1.) Log in to the PlugIn Proxy Server.
- 2.) Open a Command Shell with Administrator privileges
- 3.) Go to installation directory and subdirectory modules\rest-client
- 4.) Execute the following command

```
bh-rest-client.exe -cs configs\[System name].cfg -f getconnectionstatus
```

The System name is the logical name you defined in the Grafana Data Source PlugIn for the BHOM System.

```
C:\_NodeApps\BHPuginProxy\BHPuginProxy_V1.x.xx\modules\rest-client>bh-rest-client.exe -cs configs\BMC-BHOM-SAS-Dev.cfg -f getconnectionstatus
(8)MC (H)elix (R)EST (API) (C)lient NG VB.9.01 for TAUBER Grafana PlugIn Proxy
(c) TAUBER Business Solutions UG
written by Timo Schmidt / http://www.tabuso.com/
Subscription license expires on [01.01.2025]
{"state": "OK", "licState": "OK", "message": "Subscription license expires on [01.01.2025]", "licenseType": "temp", "licenseProductCode": "BGP0", "licenseProduct": "TAUBER Grafana Device List Panel PlugIn" }
C:\_NodeApps\BHPuginProxy\BHPuginProxy_V1.x.xx\modules\rest-client>
```

Test REST Client command with result

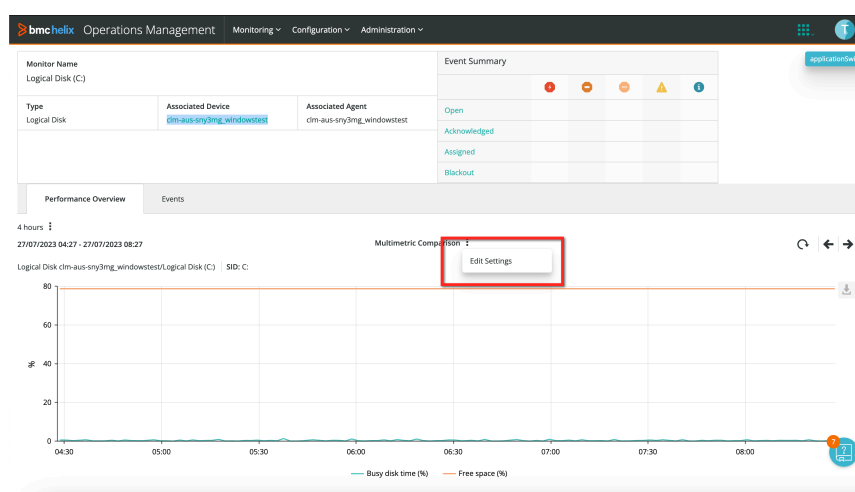
If the connection fails, make sure that the BHOM Presentation Layer is accessible on the HTTPS Port from the Server running the PlugIn Proxy.

If you can't make it work anymore contact support@tabuso.com!!!

No Metric or Event data in the Panel

If you can connect to the BHOM System but the Panel does not show any data you have to make sure, that you can graph the Attribute within BHOM. Please check that first! Next step is to copy the right information from BHOM into the Query Editor. Follow the steps below:

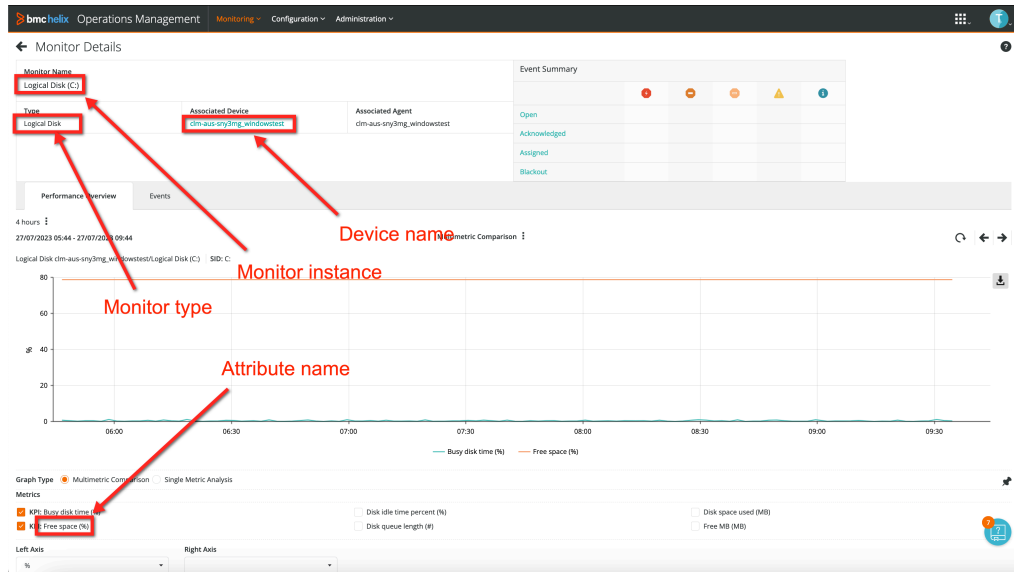
- 1.) Login to BHOM and display the Graph for the Attribute you want to show in a Panel in Grafana
- 2.) In a Multi Attribute Graph select the “Edit Settings” button to show all available Attributes



BHOM Graph with multiple Attributes

- 3.) Copy the Device name to the “Device name” (1. in the picture below) input field in the Query Editor.
- 4.) Copy the Monitor Type Name (2. in the picture below) to the “Monitor Type” input field in the Query Editor.
- 5.) Copy the Monitor Instance Name (3. in the picture below) to the “Monitor Inst.” Input field in the Query Editor.
- 6.) Copy the Attribute name (4. in the picture below) to the “Attribute” Input field in the Query Editor.

See Picture on the next page!



BHOM Graph with marked information for Query Editor

Known problems

No problems known yet!

ID	Status	Affected Version	Description
00000004	Open	All	When adding a Data link to the Table Panel clicking the link might result in a "403" error. Workaround Copy and paste the link into a new Browser tab