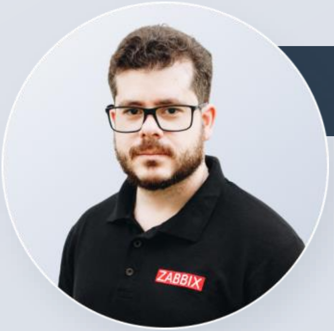
ZABBIX
SUMMIT
2024

zabbix_utils
API Library for
Enterprise Use Cases



Victor Breda Credidio

Support Engineer and Certified Zabbix trainer

zabbix_utils library

What is it? What is it for? How does it work?

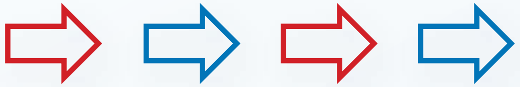
- What is it?
  - An officially supported Python library maintained by our Integration team

  - Why Python?
- What is it for?
  - To simplify the interaction with the Zabbix API, with the Zabbix server/proxy, and with agents

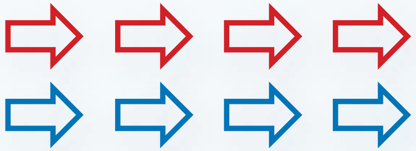
zabbix_utils library

How does it work?

- We have different classes for specific purposes:
  - ZabbixAPI
  - Getter
  - Sender
  - ItemValue



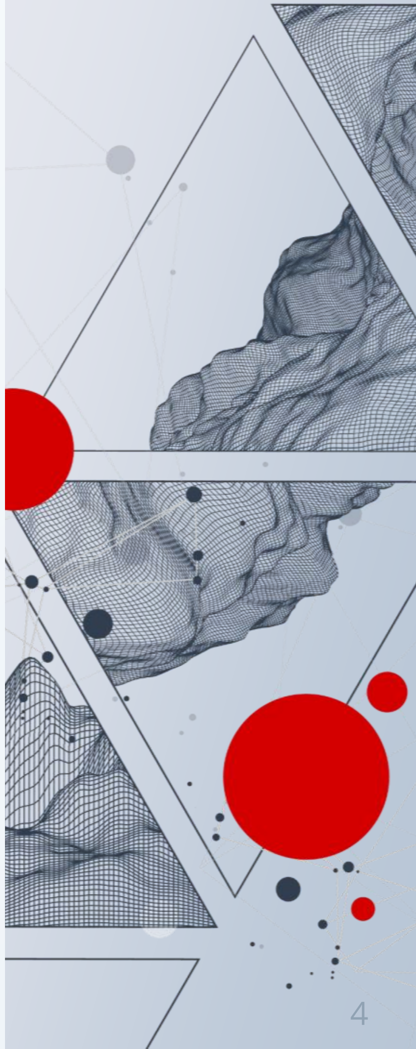
- Asynchronous classes:
  - AsyncZabbixAPI
  - AsyncGetter
  - AsyncSender



zabbix_utils library

Requirements and installation

- Supported versions:
  - Zabbix 5.0+
  - Python 3.8+
- Tested on:
  - Zabbix 5.0, 6.0, 6.4, and 7.0
  - Python 3.8, 3.9, 3.10, 3.11, and 3.12
- Dependencies:
  - aiohttp library (in case you’ll use async mode)



zabbix_utils library

Requirements and installation

- Installation from Python Package Index (PyPI):
  - Using pip:

```
$ pip install zabbix_utils
$ pip install zabbix_utils[async] #For async mode
```

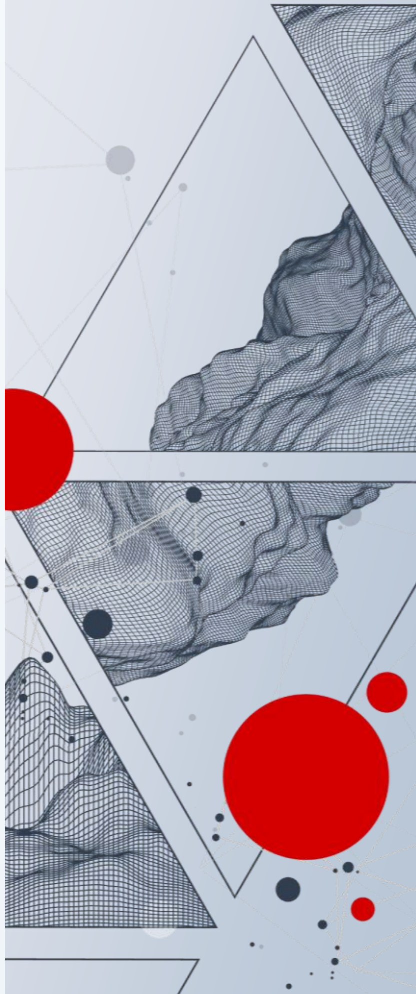
- Installation from Zabbix repository:

```
# dnf install python3-zabbix-utils
```

```
# apt install python3-zabbix-utils
```

- Installation from GitHub:

```
$ git clone https://github.com/zabbix/python-zabbix-utils
$ cd python-zabbix-utils-master/
$ python3 setup.py install
```



Use cases

Use case #1

Scenario: you’re migrating from a legacy tool to Zabbix and want to feed your Host on Zabbix with at least the last month of data using a .rrd file.

- Key points:
  - Fetch data from .rrd file
  - Create a new Host and the corresponding items on Zabbix via API
  - Push historic data to Zabbix
  - Update item types to Zabbix agent and adjust the keys accordingly

Use case #1

- Overview:
  - The Python script integrates RRDTool and zabbix_utils to automate pushing metrics into Zabbix
- Key components:
  - RRDTool
  - ZabbixAPI class
- Authentication and setup:
  - Pre-configured API Token
  - Zabbix Host creation
- Item Mapping:
  - 'cpu_util': 'system.cpu.util',
  - 'memory_usage': 'vm.memory.size[pused]',
  - 'network_in': 'net.if.in[eth0]',
  - 'network_out': 'net.if.out[eth0]'

Use case #1

- Creating Zabbix items:
  - Based on the mapped items, it will create corresponding items on Zabbix with those keys, but with the type of Zabbix trapper
- RRD File processing:
  - The RRD file have all the metrics read by the script
  - With a “for” clause, the metrics will be sent to Zabbix server using the history.push method
- Asynchronous execution:
  - Synchronous mode was taking way too long to be usable in practice
  - The data don’t need to be written in sequence, so we can use async mode
- Error handling and adaptations:
  - Each environment can present their own requirements
  - Homologation/Dev environment FTW

Use case #2

Scenario: you have a scheduled routine been executed (with Python) every now and then to achieve some specific task and want to keep track of the results of the execution.

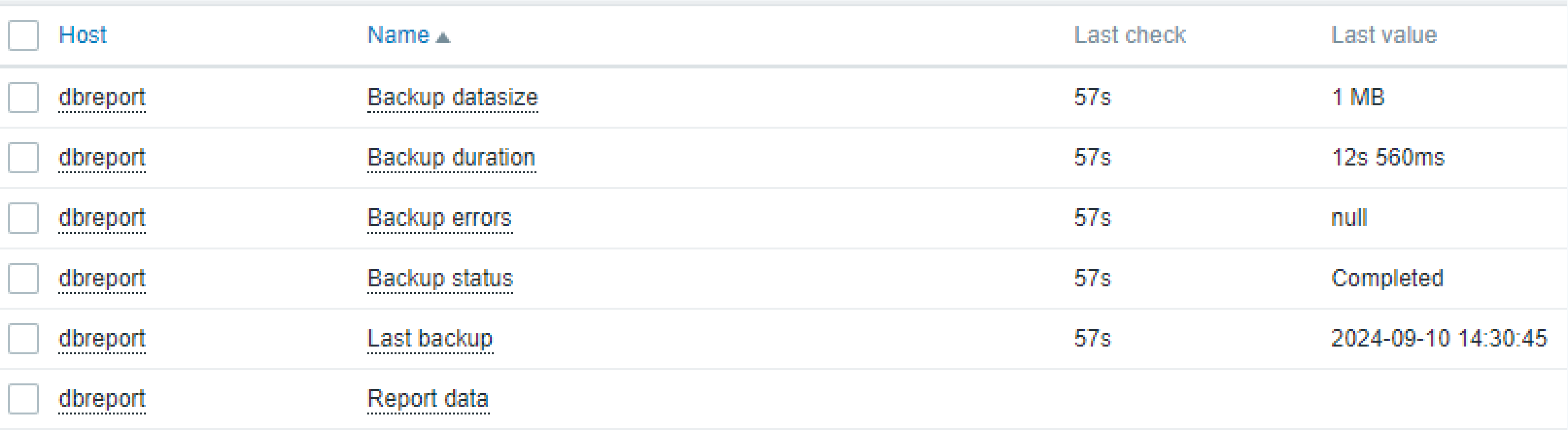
- Key points:
  - Add Sender class on from zabbix_utils on the script
  - Make sure that the script will generate the desired report data
  - Push the report data to Zabbix using the Sender class
  - Use the content from the report as a master item

Use case #2

```
{

    "LastBackup": "2024-09-10 14:30:45",
    "Duration": "12.56 seconds",
    "Status": "Completed",
    "DataBackedUp": "1048576 bytes",
    "ErrorsFound": null

}
```

```
from zabbix_utils import Sender

value = '{ "LastBackup": "2024-09-10 14:30:45", "Duration": "12.56
seconds", "Status": "Completed", "DataBackedUp": "1048576 bytes",
"ErrorsFound": null }'

sender = Sender(server='zabbix-server.vbclab.lan', port=10051)

response = sender.send_value('dbreport', 'report.data', value)
```

Use case #2

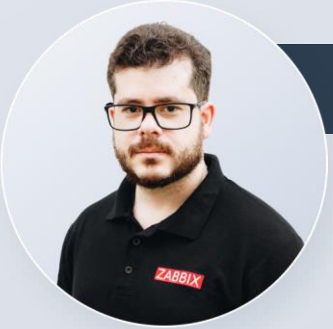


□	Host	Name ▲	Last check	Last value
□	dbreport	Backup datasize	57s	1 MB
□	dbreport	Backup duration	57s	12s 560ms
□	dbreport	Backup errors	57s	null
□	dbreport	Backup status	57s	Completed
□	dbreport	Last backup	57s	2024-09-10 14:30:45
□	dbreport	Report data		

```python
from zabbix_utils import Sender

value = '{ "LastBackup": "2024-09-10 14:30:45", "Duration": "12.56 seconds", "Status": "Completed", "DataBackedUp": "1048576 bytes", "ErrorsFound": null }'

sender = Sender(server='zabbix-server.vbclab.lan', port=10051)

response = sender.send_value('dbreport', 'report.data', value)
```

ZABBIX SUMMIT 2024

Thank you!



Victor Breda Credidio

Support Engineer and Certified Zabbix trainer