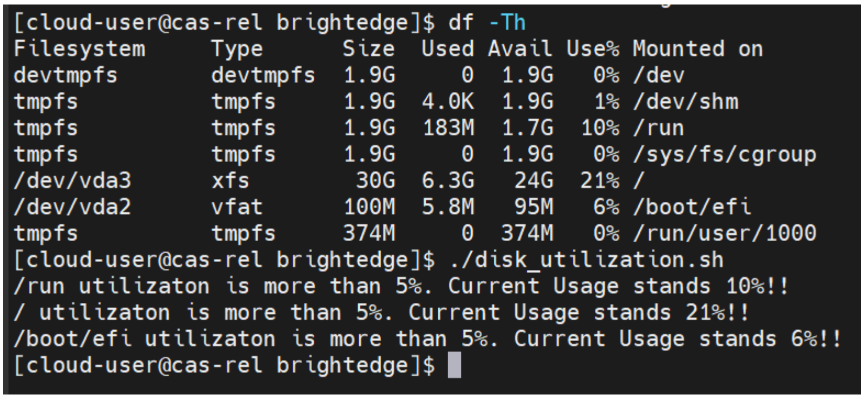
Task 1) Write a Linux shell script that determines if any of the mounted file system has less than 20% disk free

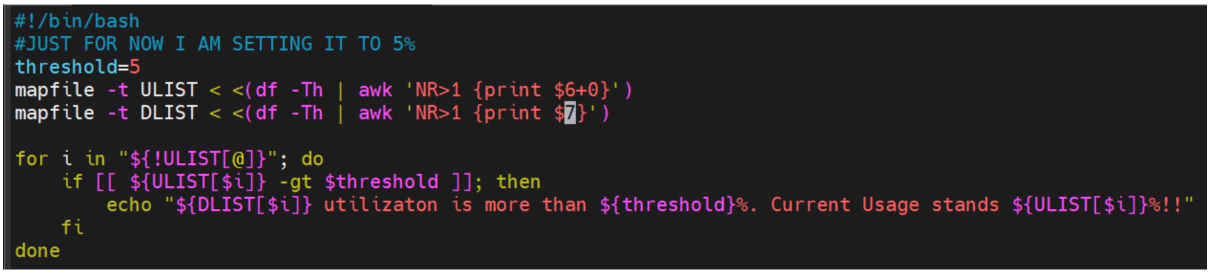
⇨ disk_utilization.sh

So, we'll get disk utilization like in the following way.



```
[cloud-user@cas-rel brightedge]$ df -Th
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.9G     0  1.9G   0% /dev
tmpfs          tmpfs     1.9G  4.0K  1.9G   1% /dev/shm
tmpfs          tmpfs     1.9G  183M  1.7G  10% /run
tmpfs          tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/vda3      xfs        30G  6.3G   24G  21% /
/dev/vda2      vfat      100M  5.8M   95M   6% /boot/efi
tmpfs          tmpfs     374M     0  374M   0% /run/user/1000
[cloud-user@cas-rel brightedge]$ ./disk_utilization.sh
/run utilizaton is more than 5%. Current Usage stands 10%!!
/ utilizaton is more than 5%. Current Usage stands 21%!!
/boot/efi utilizaton is more than 5%. Current Usage stands 6%!!
[cloud-user@cas-rel brightedge]$ 
```

Below is the shell script for performing the task of getting the list mounted Filesystem which has been utilized more than 80%. For current purpose I've set the threshold to 5%.



```
#!/bin/bash
#JUST FOR NOW I AM SETTING IT TO 5%
threshold=5
mapfile -t ULIST < <(df -Th | awk 'NR>1 {print $6+0}')
mapfile -t DLIST < <(df -Th | awk 'NR>1 {print $7}')

for i in "${!ULIST[@]}"; do
    if [[ ${ULIST[$i]} -gt $threshold ]]; then
       echo "${DLIST[$i]} utilizaton is more than ${threshold}%. Current Usage stands ${ULIST[$i]}%!!"
    fi
done
```

The code performs the following tasks:

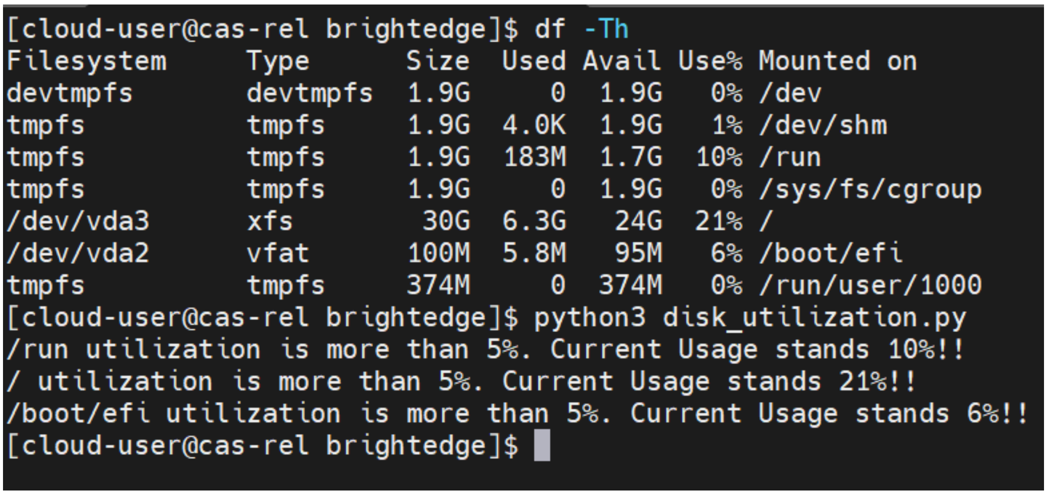
1. Get the list of Used% of all the mounted disks in an array (ULIST) and removes the % symbol after it.
2. Get the list of all the Corresponding Mounted files in another array (DLIST).
3. Next for Every Element in the array ULIST It is going to generate an output if the usage is greater than the threshold.

Note. I would like to add a counter to get an email alert using mailx if any of the file system is above threshold. I've edited the code to 80% and attached in a zip.

TASK 2) Write a python equivalent version of a script that determines if any of the mounted file system has less than 20% disk free

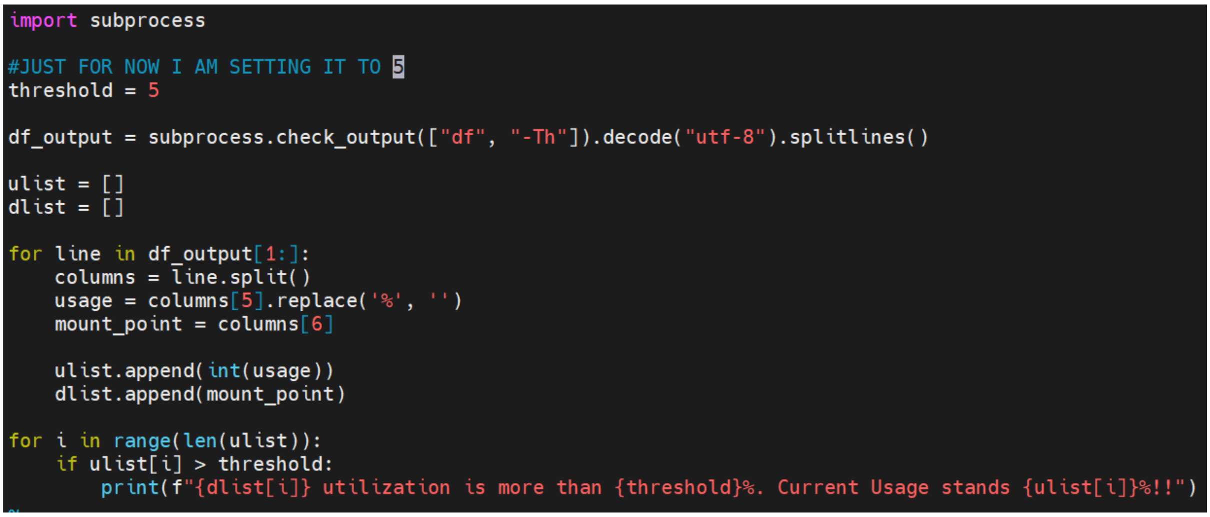
- ⇨ disk_utilization.py

So, we’ll get disk utilization like in the following way.



```
[cloud-user@cas-rel brightedge]$ df -Th
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.9G     0  1.9G   0% /dev
tmpfs          tmpfs     1.9G  4.0K  1.9G   1% /dev/shm
tmpfs          tmpfs     1.9G  183M  1.7G  10% /run
tmpfs          tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/vda3      xfs        30G  6.3G   24G  21% /
/dev/vda2      vfat      100M  5.8M   95M   6% /boot/efi
tmpfs          tmpfs     374M     0  374M   0% /run/user/1000
[cloud-user@cas-rel brightedge]$ python3 disk_utilization.py
/run utilization is more than 5%. Current Usage stands 10%!!
/ utilization is more than 5%. Current Usage stands 21%!!
/boot/efi utilization is more than 5%. Current Usage stands 6%!!
[cloud-user@cas-rel brightedge]$
```

Below is the equivalent script for performing the task of getting the list mounted Filesystem which has been utilized more than 80%. For current purpose I’ve set the threshold to 5%.



```python
import subprocess

#JUST FOR NOW I AM SETTING IT TO 5
threshold = 5

df_output = subprocess.check_output(["df", "-Th"]).decode("utf-8").splitlines()

ulist = []
dlist = []

for line in df_output[1:]:
    columns = line.split()
    usage = columns[5].replace('%', '')
    mount_point = columns[6]

    ulist.append(int(usage))
    dlist.append(mount_point)

for i in range(len(ulist)):
    if ulist[i] > threshold:
        print(f"{dlist[i]} utilization is more than {threshold}%. Current Usage stands {ulist[i]}%!!")
```

The code performs the following tasks:

1. Registers the output of “df -Th” as an array by splitting them line by line. Which means each row is currently an element of array df_output.
2. Since the first row is the header we really don’t need it So it’ll iterate from 2nd element to last splitting each word of lines word by word. We’ll have something like [“/dev/vda3”,” xfs”,”30G”,”6.3G”,”24G” ….]
3. We’ll pick the 6th element that is the usage and replace % symbol and shove it in ulist and similary 7th element in dlist.
4. The for loop subsequently checks for threshold.

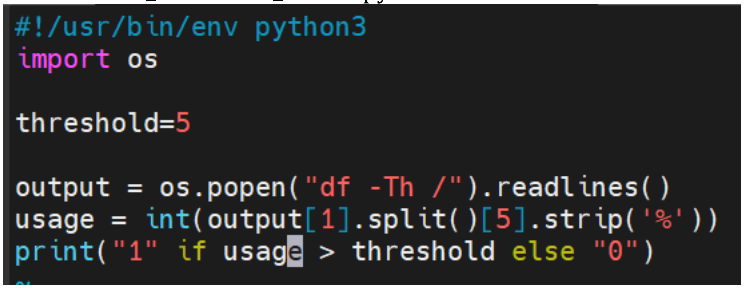
Note. I would like to add a counter to get an email alert using MIMETEXT if any of the file system is above threshold. I’ve edited the code to 80% and attached in a zip.

ZABBIX VERSION USED IS 6.0.6 LTS
TASK 3)

- Build and install zabbix server and agent on same machine.
- Uses zabbix user parameters that returns a single item boolean item that returns whether or not the system has any filesystem that is using over 20% of disk free
- Configure the item on the zabbix server

I am working on zabbix for the first time so for better understanding I have done this in 2 ways. Using python script since that what is required in task and another is direct shell command.

⇨ disk_utilization_zabbix.py



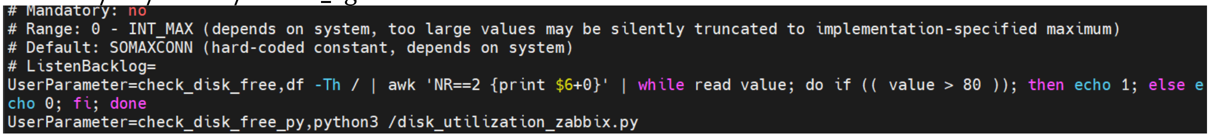
```
#!/usr/bin/env python3
import os

threshold=5

output = os.popen("df -Th /").readlines()
usage = int(output[1].split()[5].strip('%'))
print("1" if usage > threshold else "0")
```

Checking is the mounted directory / is more than 80% used or not. For current purpose I am setting it to 5%.

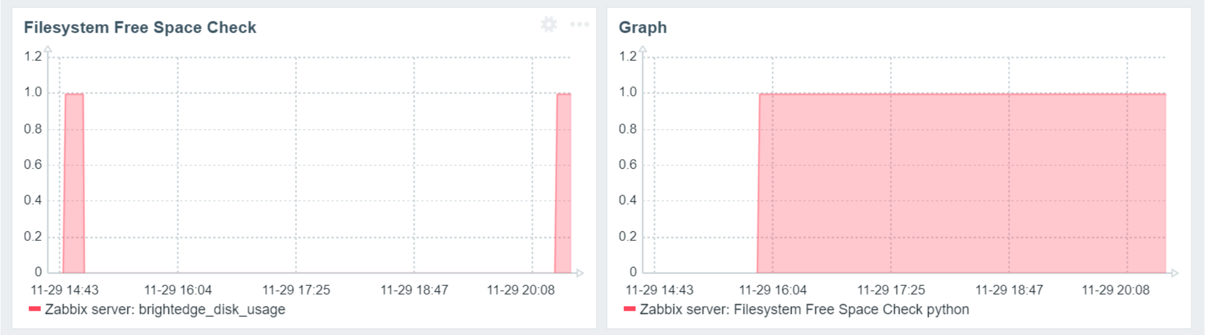
⇨ /etc/zabbix/zabbix_agentd.conf



```
# Mandatory: no
# Range: 0 - INT_MAX (depends on system, too large values may be silently truncated to implementation-specified maximum)
# Default: SOMAXCONN (hard-coded constant, depends on system)
# ListenBacklog=
UserParameter=check_disk_free,df -Th / | awk 'NR==2 {print $6+0}' | while read value; do if (( value > 80 )); then echo 1; else e
cho 0; fi; done
UserParameter=check_disk_free_py,python3 /disk_utilization_zabbix.py
```

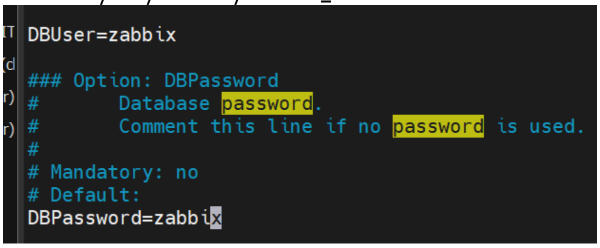
I have defined the user parameters for both the ways one is using direct command and other is using the python script.

Graph:



I have kept different threshold for demonstration.

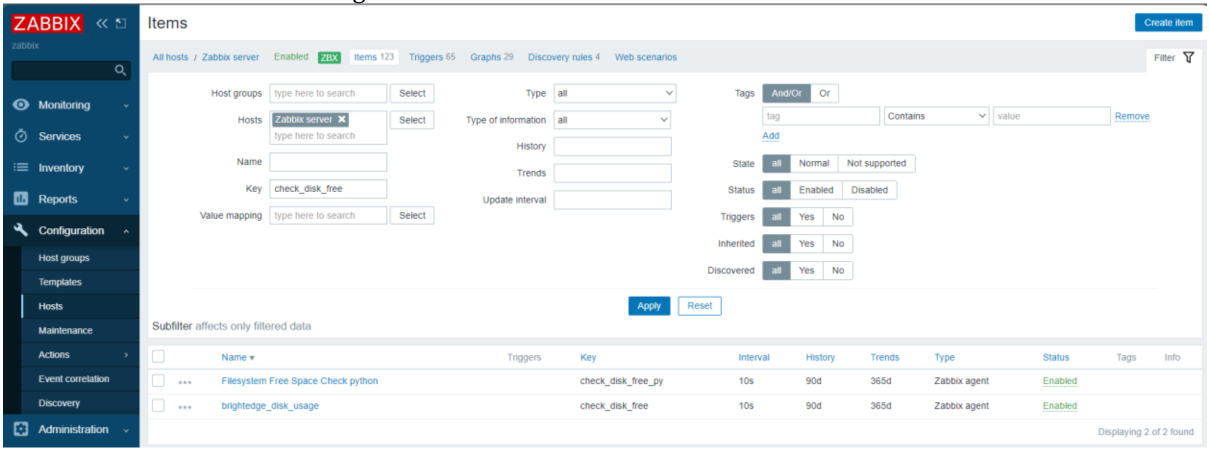
⇨ /etc/zabbix/zabbix_server.conf



```
DBUser=zabbix

### Option: DBPassword
#       Database password.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=zabbix
```

⇨ Zabbix front-end configuration:



Host groups: type here to search

Hosts: Zabbix server

Name:

Key: check_disk_free

Value mapping: type here to search

Type: all

Type of information: all

History:

Trends:

Update interval:

Tags: And/Or | Or — tag | Contains | value

State: all | Normal | Not supported

Status: all | Enabled | Disabled

Triggers: all | Yes | No

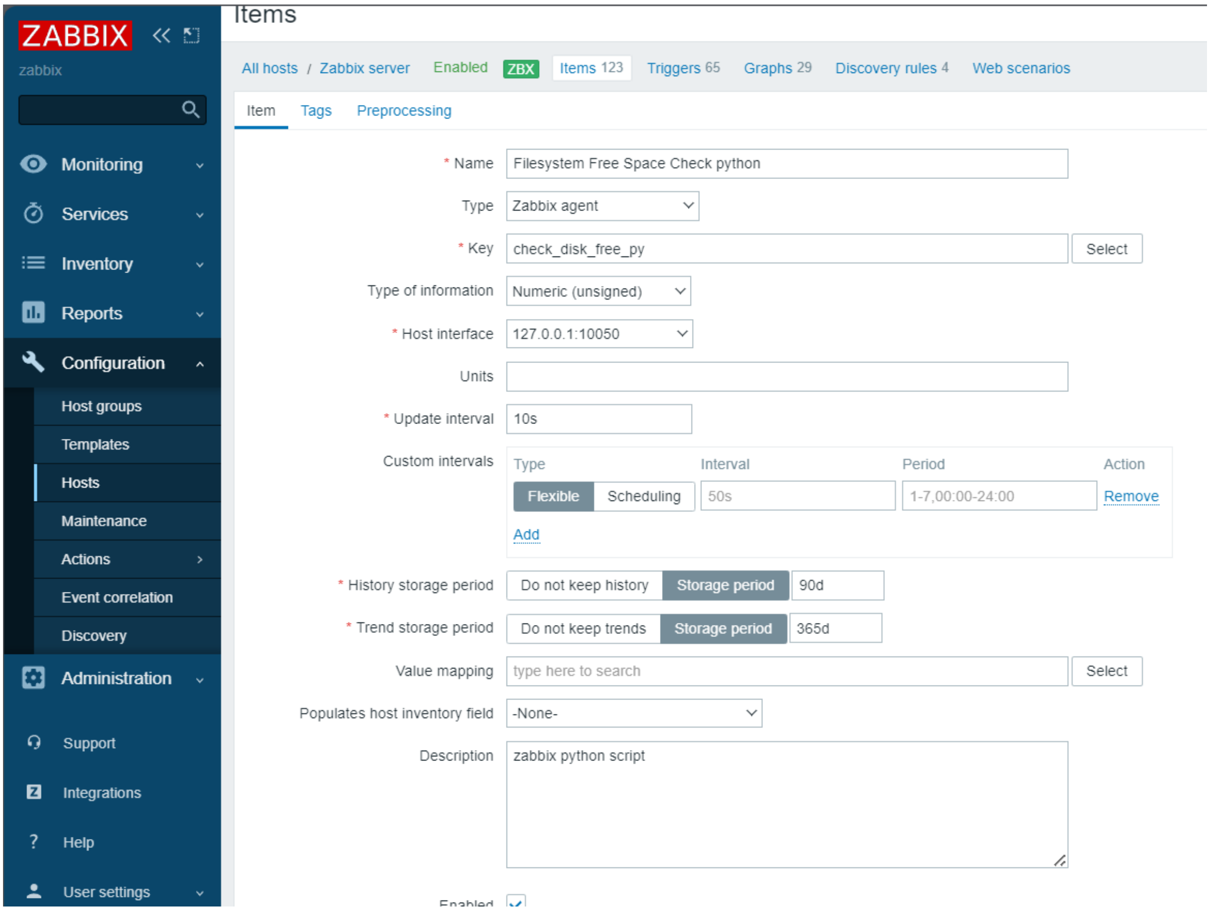
Inherited: all | Yes | No

Discovered: all | Yes | No

Subfilter affects only filtered data

	Name ▾	Triggers	Key	Interval	History	Trends	Type	Status	Tags	Info
□	Filesystem Free Space Check python		check_disk_free_py	10s	90d	365d	Zabbix agent	Enabled		
□	brightedge_disk_usage		check_disk_free	10s	90d	365d	Zabbix agent	Enabled		

Displaying 2 of 2 found



* Name: Filesystem Free Space Check python

Type: Zabbix agent

* Key: check_disk_free_py

Type of information: Numeric (unsigned)

* Host interface: 127.0.0.1:10050

Units:

* Update interval: 10s

Custom intervals:

Type	Interval	Period	Action
Flexible | Scheduling	50s	1-7,00:00-24:00	Remove

* History storage period: Do not keep history | Storage period: 90d

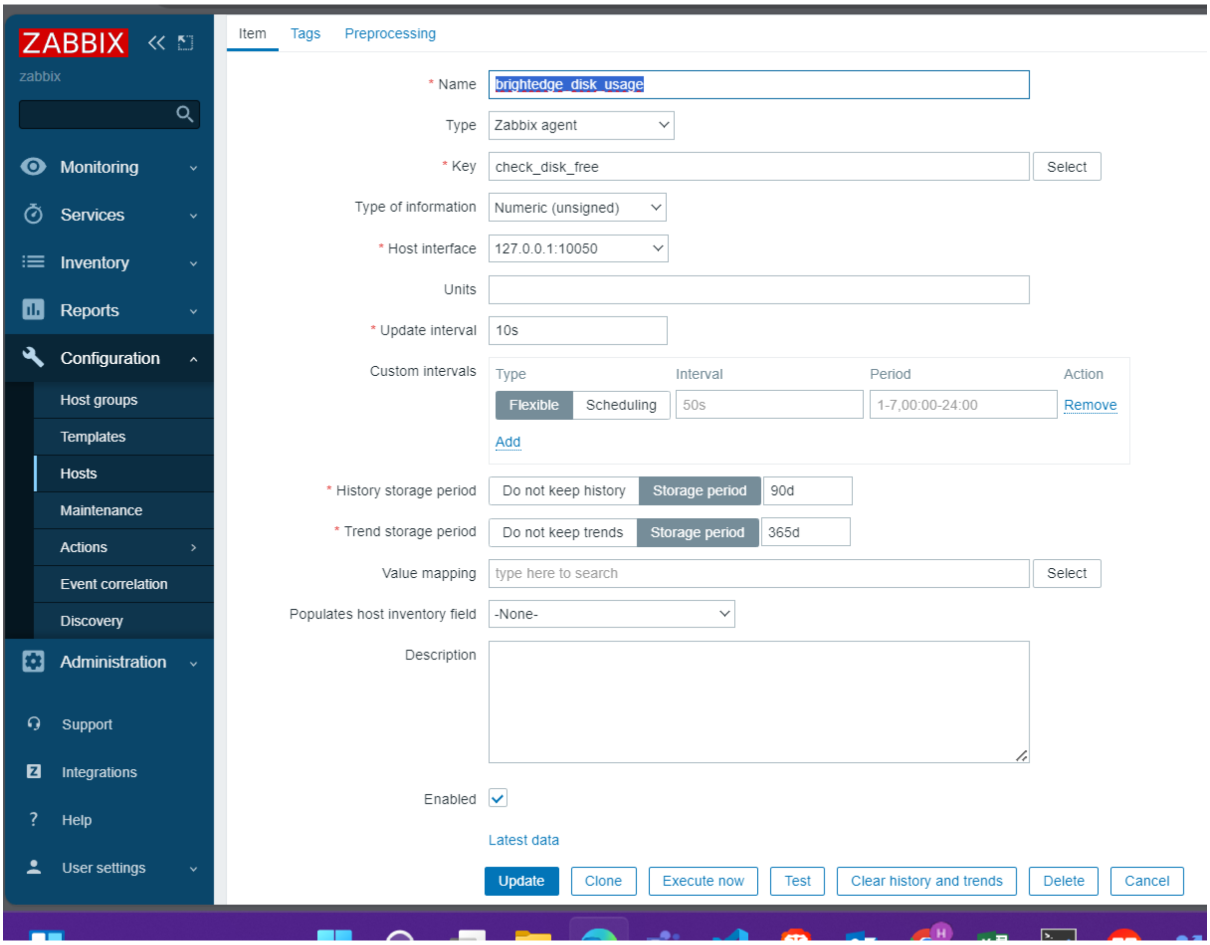
* Trend storage period: Do not keep trends | Storage period: 365d

Value mapping: type here to search

Populates host inventory field: -None-

Description: zabbix python script

Enabled ☑



* Name: brightedge_disk_usage

Type: Zabbix agent

* Key: check_disk_free

Type of information: Numeric (unsigned)

* Host interface: 127.0.0.1:10050

Units:

* Update interval: 10s

Custom intervals:

Type	Interval	Period	Action
Flexible / Scheduling	50s	1-7,00:00-24:00	Remove

* History storage period: Do not keep history [✓ Storage period] 90d

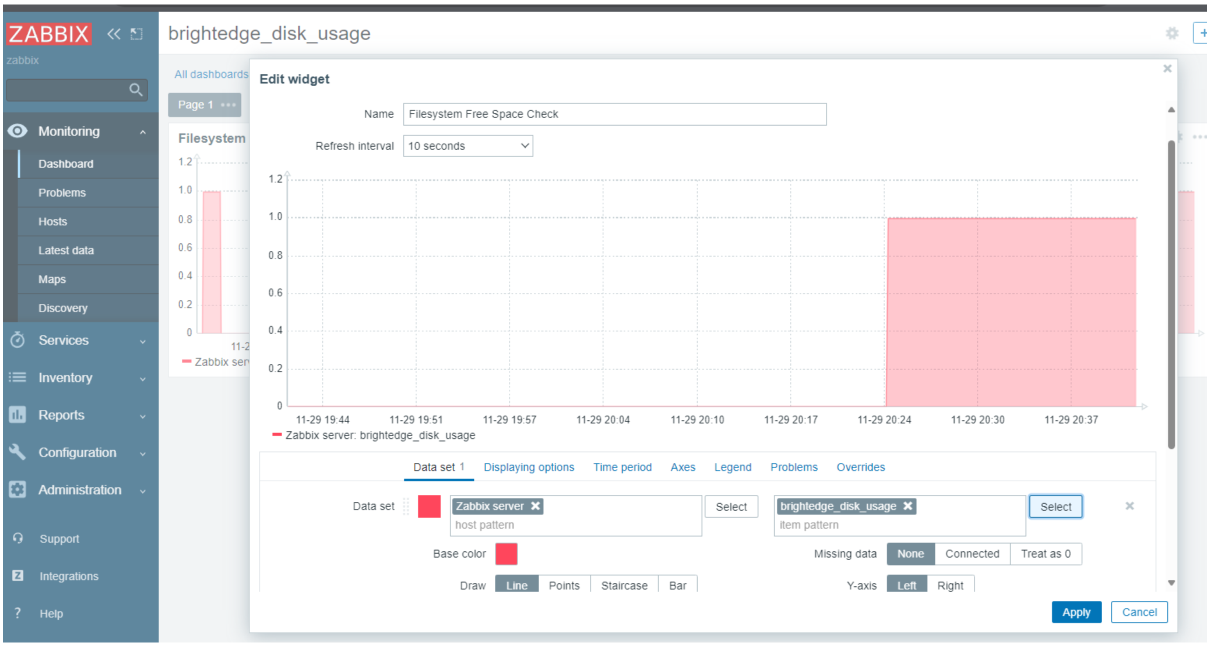
* Trend storage period: Do not keep trends [✓ Storage period] 365d

Value mapping: type here to search

Populates host inventory field: -None-

Description:

Enabled: ☑



Name: Filesystem Free Space Check

Refresh interval: 10 seconds

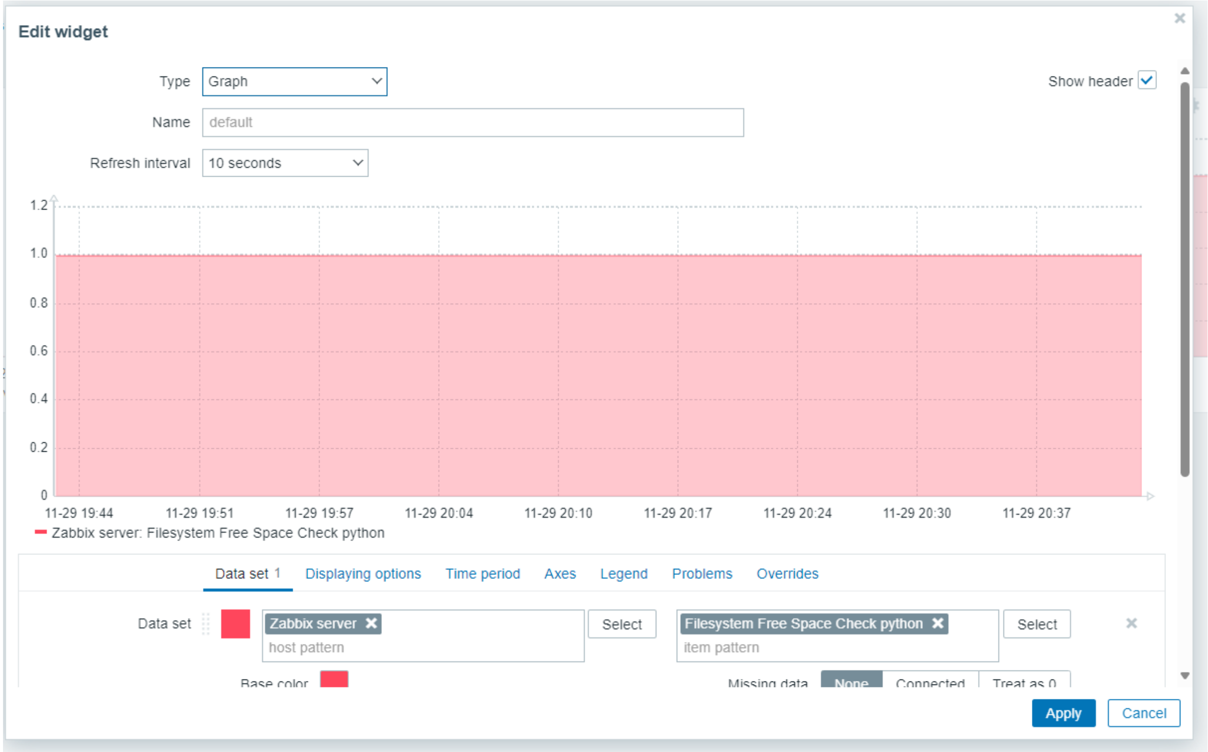
Data set: Zabbix server (host pattern) | brightedge_disk_usage (item pattern)

Base color: 🟥

Missing data: [✓ None] Connected Treat as 0

Draw: [✓ Line] Points Staircase Bar

Y-axis: [✓ Left] Right



Edit widget

Type: Graph

Show header ☑

Name: default

Refresh interval: 10 seconds

🟥 Zabbix server: Filesystem Free Space Check python

Data set: 🟥 Zabbix server ✖ (host pattern) Select | Filesystem Free Space Check python ✖ (item pattern) Select

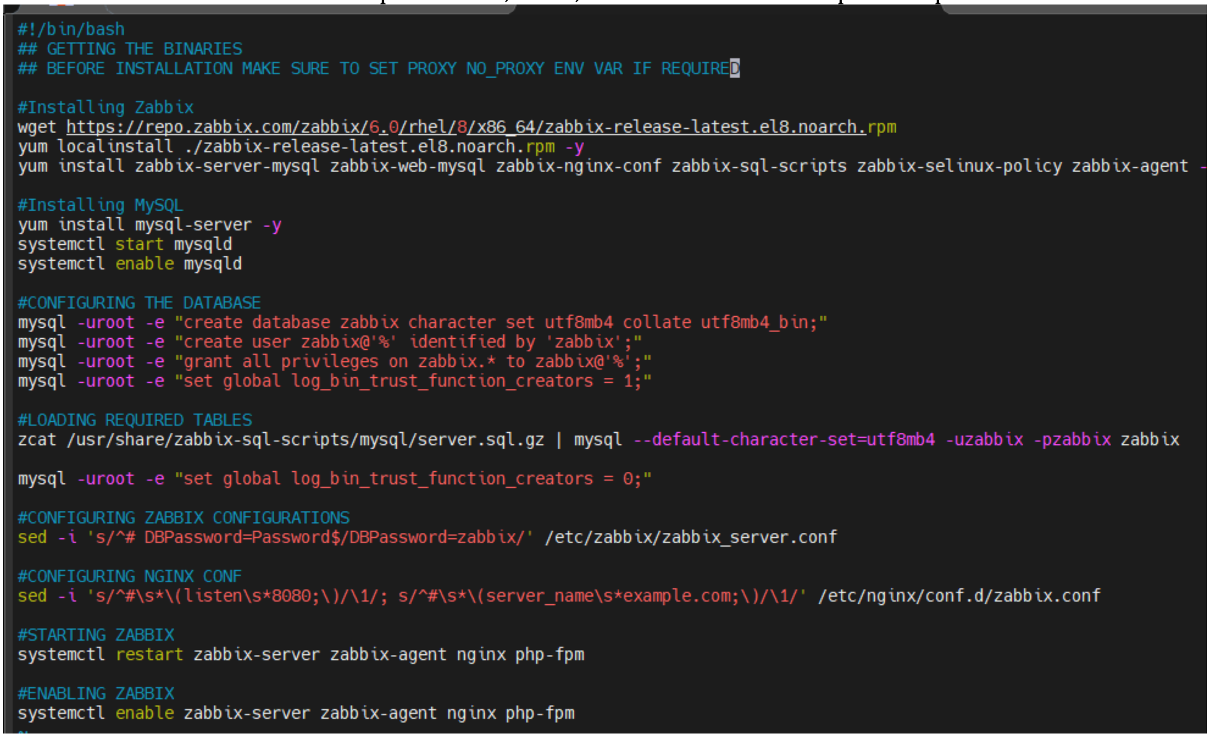
Base color: 🟥

Missing data: None | Connected | Treat as 0

BONUS

⇨ automate_deploy.sh

After configuring once, the steps were pretty straightforward. This is obviously not a production grade automation. Make sure there are ports 3306, 8080, 80 and 10050 are not pre occupied.



```
#!/bin/bash
## GETTING THE BINARIES
## BEFORE INSTALLATION MAKE SURE TO SET PROXY NO_PROXY ENV VAR IF REQUIRED

#Installing Zabbix
wget https://repo.zabbix.com/zabbix/6.0/rhel/8/x86_64/zabbix-release-latest.el8.noarch.rpm
yum localinstall ./zabbix-release-latest.el8.noarch.rpm -y
yum install zabbix-server-mysql zabbix-web-mysql zabbix-nginx-conf zabbix-sql-scripts zabbix-selinux-policy zabbix-agent -

#Installing MySQL
yum install mysql-server -y
systemctl start mysqld
systemctl enable mysqld

#CONFIGURING THE DATABASE
mysql -uroot -e "create database zabbix character set utf8mb4 collate utf8mb4_bin;"
mysql -uroot -e "create user zabbix@'%' identified by 'zabbix';"
mysql -uroot -e "grant all privileges on zabbix.* to zabbix@'%';"
mysql -uroot -e "set global log_bin_trust_function_creators = 1;"

#LOADING REQUIRED TABLES
zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -pzabbix zabbix

mysql -uroot -e "set global log_bin_trust_function_creators = 0;"

#CONFIGURING ZABBIX CONFIGURATIONS
sed -i 's/^# DBPassword=Password$/DBPassword=zabbix/' /etc/zabbix/zabbix_server.conf

#CONFIGURING NGINX CONF
sed -i 's/^#\s*\(listen\s*8080;\)/\1/; s/^#\s*\(server_name\s*example.com;\)/\1/' /etc/nginx/conf.d/zabbix.conf

#STARTING ZABBIX
systemctl restart zabbix-server zabbix-agent nginx php-fpm

#ENABLING ZABBIX
systemctl enable zabbix-server zabbix-agent nginx php-fpm
```

Similarly, we can create an ansible equivalent playbook for this.