Zabbix Autoregistration :

1-action → autoregistration action

2- hostmetadata → contain linux

3-operation → add host – link to template – enable host

4- edit /etc/Zabbix/Zabbix_agentd.conf

Server : (ZABBIX_SERVER_IP)
Server Active : (ZABBIX_SERVER_IP)
Hostmetadata → uncomment it and ad linux pharase

5- systemctl restart Zabbix-agent

User Parameters :

1-Edit agent config file (UserParameters Section):
Format:
UserParameters=<key>,<Shell command>
Example:
UserParameters=isalive,echo 1

2- restart agent service :
#systemctl restart Zabbix-agent

3-Create Item in Zabbix UI:
Name:isalive
Type: Zabbix agent
Key:isalive
Test it !

4- another way to test userpaprameters: (Does not need restart Zabbix agent)
#Zabbix_agentd –t isalive

Dependent and calculated Item :
First :
https://sbcode.net/zabbix/log-file-monitoring-http-codes/

Creating dependent items means that the agent doesn't need to run possibly identical queries on a host many times in order to extract parts of a value. The master item runs once on the host, and then the Zabbix server (or Zabbix proxy if host managed by proxy) updates the dependent items each time the master item gets its new values **.**

1- create template nginx access log

2- host group → template module (not necessary)

3-create item in template (Master Item) : name →HTTP Access log
Type : Zabbix agent active
Key : log[/var/log/nginx/access.log,"^.*",,,skip,\0,,,]
Type of information : log

4- add created template to host that gather nginx log

5- select three . (…) from item that we created in template and create dependent item

Name : status code

Key: HTTPStatusCode

Preprocceccing : ^(\S+) (\S+) (\S+) \[([\w:\/]+\s[+\-]\d{4})\] \"(\S+)\s?(\S+)?\s?(\S+)?\" (\d{3}|-) (\d+|-
)\s?\"?([^\"]*)\"?\s?\"?([^\"]*)\“

Output : \8 → status code is here in nginx log

Zabbix-get command & Zabbix sender and trapper :

Utility to get data from agent . When you want to check item from command line .

1- install : apt install Zabbix-get
Help : Zabbix_get –h
2- get ping form other host:
Zabbix_get –s IP –p 10050 –k agent.ping (Get Error ! Permission Problem)
On Destination : vi /etc/Zabbix/Zabbix_agentd.conf and add source server :
Server=(SOURCE_IP)
TLSAccept=psk,uneccrypted

3- systemctl restart Zabbix-agent

Zabbix_get –s IP –p 10050 –k agent.version

Zabbix_get –s IP –p 10050 –k agent.hostname

Zabbix_get –s IP –p 10050 –k system.cpu.load[all,avg1]

Zabbix sender is a command line utility that may be used to send performance data to Zabbix server for processing .

1- sudo apt install Zabbix-sender

2- Zabbix_sender –z (PROXY_IP) –s “swarm-1” –k temp –o 23.8
Zabbix_sender –z (127.0.0.1) –s “swarm-1” –k temp –o 1.23

Create item on swarm-1 in Zabbix ui with name temp and type : Zabbix trapper key : temp

Note : Zabbix trapper item does not have interval and just listen to Zabbix sender to receive data

ZABBIX

Create Custom Low level Discovery (LLD)

Discover rule : the script that will run , find group of sth on our host . There is some built in discovery rule :
vfs.fs.discovery : Used by Mounted Filesystem Discovery
net.if.discovery : Used by Network Interface Discovery
vfs.dev.discovery : Used by Block Devices Discovery

```
zabbix_agentd -t vfs.dev.discovery
zabbix_agentd -t net.if.discovery
zabbix_agentd -t vfs.fs.discovery
```

```
[
 {
  "{#DEVNAME}": "loop8",
  "{#DEVTYPE}": "disk"
 },
 {
  "{#DEVNAME}": "vda15",
  "{#DEVTYPE}": "partition"
 },
 {
  "{#DEVNAME}": "vda14",
  "{#DEVTYPE}": "partition"
 },
 {
  "{#DEVNAME}": "vda1",
  "{#DEVTYPE}": "partition"
 },
 {
  "{#DEVNAME}": "vda",
  "{#DEVTYPE}": "disk"
 },
]
```

```
systemctl list-unit-files

Cd /home

sudo nano service_discovery.py

python3 /home/zabbix/service_discovery.py

sudo -H -u zabbix bash -c 'python3 /home/zabbix/service_discovery.py‘

sudo nano /etc/zabbix/zabbix_agentd.conf

UserParameter=service.discovery,python3 /home/service_discovery.py
UserParameter=service.isactive[*],systemctl is-active --quiet '$1' && echo 1 || echo 0

zabbix_agentd -t service.discovery

sudo service zabbix-agent restart
sudo service zabbix-agent status
```

Configuration → Templates → Create Template → linux services group:linux servers

Create A New Discovery Rule

Name: Service Discovery

Type: Zabbix Agent (active)

Key : service.discovery

Update interval : 1h

Create Prototypes

```
zabbix_agentd -t service.isactive[zabbix-agent]
```

Item Prototypes

Now to create some item prototypes

Key	Value
Name	Service Active : {#SERVICE}
Type	Zabbix Agent (active)
Key	service.isactive["{#SERVICE}"]
Type of info	Numeric (unsigned)
Update Interval	1m
Create Enabled	Un-ticked (optional)

Trigger Prototypes

Key	Value
Name	Service Not Active : {#SERVICE}
Severity	Disaster (Optional)
Expression	last(/Linux Services /service.isactive["{#SERVICE}"])=0
OK event generation	Expression
Create Enabled	Un-ticked (optional)

Add The Template To A Host:

Go into **Configuration → Hosts**, select your host, and add the *Service Discovery* template.
After some time, you can inspect the new data using the **Monitoring → Latest Data** tab and also see the Host items.

Note : if get this error :

Special characters "\, ', ", `, *, ?, [,], {, }, ~, $, !, &, ;, (,), <, >, |, #, @, 0x0a" are not allowed in the parameters

Do : UnsafeUserParameters=1 in agent config file

Convert MIB file to Zabbix Template:

1- Debian/Ubuntu :

```
sudo apt-get update
sudo apt install snmp
sudo apt install snmp-mibs-downloader
```

```
yum check-update
 yum install net-snmp-utils
yum install net-snmp-libs
```

2- `sudo nano /etc/snmp/snmp.conf`

Comment out mibs lines:

```
#mibs
```

3- `snmptranslate 1.3.6.1.2.1.1`

The response should be :

```
SNMPv2-MIB::system
```

4- mib2zabbix.pl is written in perl, so you will need to install the perl dependencies.

Ubuntu:
```
apt-get install perl libxml-simple-perl libsnmp-perl
```
Centos:
```
yum install "perl(SNMP)" "perl(XML::Simple)“
```
Centos8:
```
rpm -ivh http://repo.okay.com.mx/centos/8/x86_64/release/okay-release-1-3.el8.noarch.rpm
```
5- Next, get the **mib2zabbix** perl script:
```
curl https://raw.githubusercontent.com/cavaliercoder/mib2zabbix/master/mib2zabbix.pl > mib2zabbix
```
Or :
```
wget -O mib2zabbix https://raw.githubusercontent.com/cavaliercoder/mib2zabbix/master/mib2zabbix.pl
chmod a+x mib2zabbix
```
Tests it :
```
./mib2zabbix –h
```

6- Test Huawei.MIB
```
curl http://www.circitor.fr/Mibs/Mib/H/HUAWEI-MIB.mib > HUAWEI-MIB.mib
```
Or:
```
wget -O HUAWEI-MIB.mib http://www.circitor.fr/Mibs/Mib/H/HUAWEI-MIB.mib
```

Next, see if **snmptranslate** can read this file.

```
snmptranslate -Tz -m ./HUAWEI-MIB.mib
```

```
snmptranslate -Tz -m ./HUAWEI-MIB.mib | ./mib2zabbix -o .1.3.6.1.4.1 -f template-huawei-mib.xml -N huawei-mib
```

This command will create file : **template-huawei-mib.xml**

Web Senario :

To monitor remote http server with steps:

1- configuaration → host (select host) → web → create web scenario

Name: http check website

Agent : zabbix

2-Steps:
Name check website
URL: https://zabbix.com

3- check latest data :

Send sms in zabbix :

1- create an account in sms service provider

2- get config.sh form site and put it on /usr/lib/zabbix/alertscripts

3- chmod +777 config.sh

4- in zabbix ui create media type .

Insert name

Type : script

Scriptname : name of script

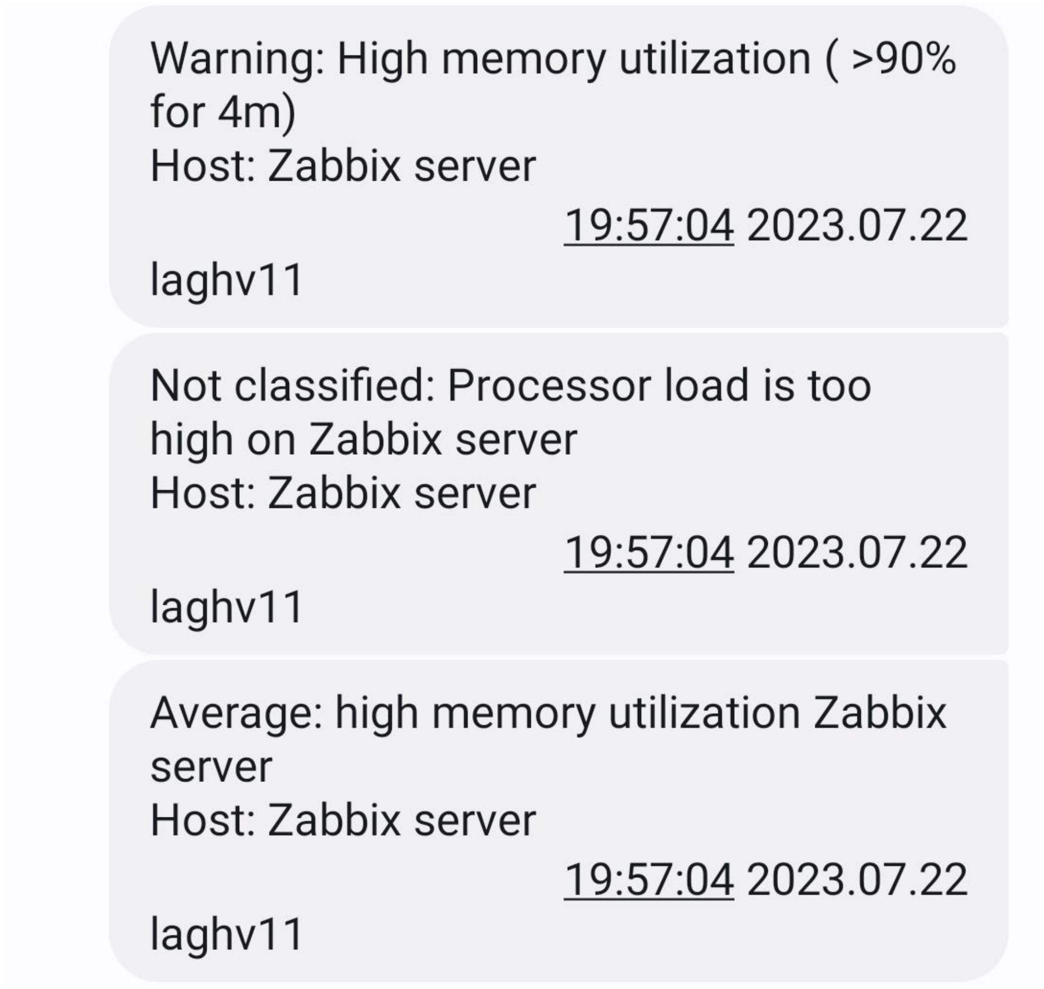
Add 5 parameters to send :

Username-password-number-{ALERT.SENDTO}-{ALERT.SUBJECT}%0a{ALERT.MESSAGE}

6- test sms :

Output like this :

Config administration →user → media → sms



```
Warning: High memory utilization ( >90%
for 4m)
Host: Zabbix server
                    19:57:04 2023.07.22
laghv11
```

```
Not classified: Processor load is too
high on Zabbix server
Host: Zabbix server
                    19:57:04 2023.07.22
laghv11
```

```
Average: high memory utilization Zabbix
server
Host: Zabbix server
                    19:57:04 2023.07.22
laghv11
```

Send zabbix Alerts With telegram :

1- create bot with @Botfather in telegram

2-Copy Api Access token

3- go to zabbix → media type →paste your token in part token and enable and update it .

4- start your bot by searching in telegeam .

5- create group and add your bot to group

6- [https://api.telegram.org/bot*XXX:YYY*/getUpdates](https://api.telegram.org/botXXX:YYY/getUpdates) (replace the XXX: YYY with your **HTTP API Token** you just got from Telegram)

7- Search it on google : should get ok:true

8- for test type sth in group and get that message in browser .if you did not get it , remove you bot and add it again .

9- copy chat id (negative number)

9-test telegram media → put chat id to section To:-912599753

10-you get message in telegram group

How to connect zabbix to promethous ?

1- install promethous node_exporter on machine

2-create host for your machine in zabbix and add template os linux by prom (install zabbix agent on it and ..)

3-in host macro you can see macro :

{$NODE_EXPORTER_PORT} 9100

4- there is item in this host : Get node_exporter metrics

5- check latest data of host

6-you can create discovery rule for finding all node_exporter in network

Install and manage Grafana with bash and terraform :

1- bash Grafana.sh

2-$ terraform init
$ terraform validate
$ terraform plan
$ terraform apply

3- download and install zabbix plugin for grafana :

a) grafana-cli plugins install alexanderzobnin-zabbix-app

b) https://grafana.com/api/plugins/alexanderzobnin-zabbix-app/versions/4.3.1/download

4- put it on /var/lib/Grafana/plugins

5- Restart Grafana-server service

6- add zabbix data source and config it .

Integration zabbix with jira :
https://www.zabbix.com/integrations/jira

1- install and crack jira

2-in zabbix ui , import jira_media.yaml as media type

3-
jira_url - actual URL of your Jira instance,
jira_user - Jira user login,
jira_password - password or API token (for Jira Cloud installations an API token can be obtained at https://id.atlassian.com/manage/api-tokens),
jira_project_key - text key of the Jira project (not to be mistaken with an ID!),
jira_issue_type - name of the issue type to be used when creating new issues from Zabbix notifications.