Zabbix

task

- ~~dashboard con conteggi aggregati (e altri grafici interessanti)~~
- ~~trigger di allarme: * luce accesa da troppo tempo~~
- ~~export dati csv con API~~
- ~~gestire utente non privilegiato su zabbix~~
- ~~VPN robusta che riparte quando il cloud non è raggiungibile~~
- gestire il mancato boot della componente zabbix di hass
  - /etc/openvpn/iotaiuto.conf ?
  - come far ripartire la connessione zabbix all'interno di hass ?
- esphome:
  - costruire in power meter con allarme su alexa
  - costruire un citofono con ESP32 CAM

API

```
#!/bin/sh


# 1. set connection details
url=http://monitor.iotaiuto.it/api_jsonrpc.php
user=xxx
password=xxx

# 2. get authorization token
auth=$(curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
 \"method\": \"user.login\",
 \"params\": {
  \"user\": \"$user\",
  \"password\": \"$password\"
 },
 \"id\": 1,
 \"auth\": null
}
" $url | \
jq -r '.result'
)

# 3. show triggers in problem state
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
    \"method\": \"trigger.get\",
    \"params\": {
        \"output\": \"extend\",
        \"selectHosts\": \"extend\",
        \"filter\": {
            \"value\": 1
        },
        \"sortfield\": \"priority\",
        \"sortorder\": \"DESC\"
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url | \
jq -r '.result'

# 4. logout user
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"user.logout\",
    \"params\": [],
    \"id\": 1,
    \"auth\": \"$auth\"
}
" $url
 ~ □ Scaricati □
 → □# 1. set connection details
    url=http://127.0.0.1/api_jsonrpc.php
    user=api
    password=zabbix

    # 2. get authorization token
    auth=$(curl -s -X POST \
    -H 'Content-Type: application/json-rpc' \
    -d " \
    {
     \"jsonrpc\": \"2.0\",
     \"method\": \"user.login\",
     \"params\": {
      \"user\": \"$user\",
      \"password\": \"$password\"
     },
     \"id\": 1,
    # 1. set connection details
    url=http://127.0.0.1/api_jsonrpc.php
    user=api \
    password=zabbix
    )
    # 2. get authorization token
    auth=$(curl -s -X POST \oblem state
    -H 'Content-Type: application/json-rpc' \
    -d " \ 'Content-Type: application/json-rpc' \
    {   -d " \
     \"jsonrpc\": \"2.0\",
     \"method\": \"user.login\",
     \"params\": {: \"trigger.get\",
      \"user\": \"$user\",
      \"password\": \"$password\"",
     },     \"selectHosts\": \"extend\",
     \"id\": 1,ilter\": {
     \"auth\": nullalue\": 1
    }       },
    " $url | \sortfield\": \"priority\",
    jq -r '.result'rder\": \"DESC\"
    )   },
        \"auth\": \"$auth\",
    # 3. show triggers in problem state
    curl -s -X POST \
    " $u-H 'Content-Type: application/json-rpc' \
        -d " \'.result'
    {
     \"jsonrpc\": \"2.0\",
        \"method\": \"trigger.get\",
        \"params\": {ype: application/json-rpc' \
            \"output\": \"extend\",
            \"selectHosts\": \"extend\",
            \"filter\": {0\",
                \"value\": 1ogout\",
            },ms\": [],
            \"sortfield\": \"priority\",
            \"sortorder\": \"DESC\"
        },
        \"auth\": \"$auth\",
        \"id\": 1
    }
    " $url | \
        jq -r '.result'

    # 4. logout user
    curl -s -X POST \
        -H 'Content-Type: application/json-rpc' \
        -d " \
    {
        \"jsonrpc\": \"2.0\",
        \"method\": \"user.logout\",
        \"params\": [],
        \"id\": 1,
        \"auth\": \"$auth\"
    }
    " $url
```

```
#!/bin/bash

# 1. set connection details
url=http://monitor.iotaiuto.it/api_jsonrpc.php
user=admin
password=Galileo2018.

# 2. get authorization token
auth=$(curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
 \"method\": \"user.login\",
 \"params\": {
  \"user\": \"$user\",
  \"password\": \"$password\"
 },
 \"id\": 1,
 \"auth\": null
}
" $url | \
jq -r '.result'
)

# 3. show triggers in problem state
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
 \"jsonrpc\": \"2.0\",
    \"method\": \"trigger.get\",
    \"params\": {
        \"output\": \"extend\",
        \"selectHosts\": \"extend\",
        \"filter\": {
            \"value\": 1
        },
        \"sortfield\": \"priority\",
        \"sortorder\": \"DESC\"
    },
    \"auth\": \"$auth\",
    \"id\": 1
}
" $url | \
jq -r '.result' > ../flussi/dati.json
jq -r 'del(.[].hosts) | (map(keys) | add | unique) as $cols | map(. as $row
| $cols | map($row[.])) as $rows | $cols, $rows[] | @csv'
../flussi/dati.json > ../flussi/dati_trigger.csv
jq -r '(map(.hosts[] + {"triggerid": .triggerid} | keys) | add | unique) as
$cols | map(. as $row | $cols | map($row[.])) as $rows | $cols, $rows[] |
@csv' ../flussi/dati.json > ../flussi/dati_host.csv
rm ../flussi/dati.json
cat ../flussi/dati_trigger.csv & ../flussi/dati_host.csv

# 4. logout user
curl -s -X POST \
-H 'Content-Type: application/json-rpc' \
-d " \
{
    \"jsonrpc\": \"2.0\",
    \"method\": \"user.logout\",
    \"params\": [],
    \"id\": 1,
    \"auth\": \"$auth\"
}
" $url
```