

Site Traffic Monitoring Service README

This directory contains the **Site Traffic Monitoring Service** example and is intended to be deployed at WLCG sites to gather simple network statistics from the site's network border device(s). It contains a python3 script that can be configured to read multiple interfaces via SNMP, add the IN and OUT traffic (In Bytes/sec) up and save the output in a standard formatted JSON file for use by WLCG Monitoring. We desire relatively fine-grained monitoring, if possible, and **should have an update interval of 60 seconds**, unless there is a technical reason not to.

Installation Considerations

This software should be installed on a system that has:

- Network access to the site's border devices via SNMP
- A web server (alternatively the output could be copied to a web server via a suitable script)
- Python3, net-snmp and git installed

Pick a location on that system to deploy to which we will call `INSTALL_LOC`, e.g.,

```
export INSTALL_LOC=~/.my-site-monitoring
mkdir -p ${INSTALL_LOC}
```

Install Pre-Requisites

You should also make sure you have installed any dependencies if needed:

```
yum install net-snmp git net-snmp-devel python3-devel gcc
pip3 install easysnmp
```

Installing

Then you can get the project and copy the relevant director to `INSTALL_LOC`

```
cd /tmp
git clone https://gitlab.cern.ch/wlcg-doma/site-network-information.git
cp -ar site-network-information/WLCG-site-snmp/* ${INSTALL_LOC}/
```

After getting the starting files in place in `INSTALL_LOC`, you should make some working copies of the relevant files:

```
cd ${INSTALL_LOC}
cp site-traffic-monitor.service-example site-traffic-monitor.service
cp WLCG-site-snmp-example.py WLCG-site-snmp.py
```

You will need to edit `WLCG-site-snmp.py` to customize for your use-case. The variables to be set are described in the table below.

Variable name	Description	Example
INSTALL_LOC	The location the code is installed at	See above
INTERVAL	The update interval in seconds	60
JSONOUTFILE	The path for the output file	<code>/var/www/html/aglt2-netmon.json</code>
INDICES	A python dictionary holding the devices and interfaces to monitor	See below
COMM	A python dictionary holding device snmp community strings	See below

Configuring

To configure your installation, you will need to identify the network interfaces that represent your site's border. There may be more than one interface and more than one device hosting border interfaces. You will likely need to work with your network team to ensure you correctly identify the appropriate interface(s) and get readonly access via SNMP. The next step is to confirm that snmp is working for accessing your devices. Once you have the SNMP community to use, set it as an environment variable:

```
export SNMPCOMM=myreadonlycomm
```

Then we can test access to the switch(es) or router(s) that host the site border interface(s):

```
snmpwalk -v2c -c ${SNMPCOMM} <device> .sysDescr
```

This should return a value:

```
[root@sysprov02 ~]# snmpwalk -v2c -c ${SNMPCOMM} aglt2-rtr-1.local .sysDescr
SNMPv2-MIB::sysDescr.0 = STRING: Cisco NX-0S(tm) nxos.9.3.5.bin, Software (nxos), Version 9.3(5),
RELEASE SOFTWARE Copyright (c) 2002-2020 by Cisco Systems, Inc. Compiled 7/20/2020 20:00:00
```

Please check each device that hosts border interfaces to ensure snmp access works.

Once that is verified, you can configure the `COMM` dictionary in your `WLCG-site-snmp.py` file. Here is an example for a site with two border devices:

```
COMM = {"aglt2-rtr-1.local": "mycomm1", "aglt2-rtr-2.local": "mycomm2"}
```

We now need to identify the correct SNMP indices for the border interfaces we want to monitor. Continuing with the example of `aglt2-rtr-1.local` and `aglt2-rtr-2.local`, let's say the border interfaces are `Ethernet1/51` and `Ethernet1/52` on each host. We can use the following command to identify the correct index to configure in

WLCG-site-snmp.py for aglt2-rtr-1.local:

```
snmpwalk -v2c -c ${SNMPCOMM} aglt2-rtr-1.local IF-MIB::ifDescr
```

The example output lists all the indices and interfaces:

```
...
IF-MIB::ifDescr.436233216 = STRING: Ethernet1/51
IF-MIB::ifDescr.436233728 = STRING: Ethernet1/52
      ^^^^^^^^^          ^^^^^^^^^^^^^
      Index              Interface description
```

You should find the appropriate index values for all devices hosting border interfaces. Below is the example from AGLT2:

```
INDICES = {
  "aglt2-rtr-1.local": {
    "Ethernet1/51": 436233216,
    "Ethernet1/52": 436233728,
  },
  "aglt2-rtr-2.local": {
    "Ethernet1/51": 436233216,
    "Ethernet1/52": 436233728
  },
}
```

Feel free to adjust any of the variables as desired. Once you have them configured, you can run a test:

```
# python3 WLCG-site-snmp.py
WLCG site traffic monitor started at 2022-05-09T14:41:45.955477+00:00
----- traffic monitor directory /root/site-network-information/WLCG-site-snmp/
----- traffic JSON output /var/www/html/aglt2-netmon.json
```

Note that the code is set to run forever. You need to wait for at least `INTERVAL` , CTRL-C and then verify a new output file `JSONOUTFILE` was created.

For AGLT2 it looks like:

```
{
  "Description": "Network statistics for AGLT2",
  "UpdatedLast": "2022-05-09T14:47:23.057942+00:00",
  "InBytesPerSec": 1275982208.1281207,
  "OutBytesPerSec": 1526111205.30333,
  "UpdateInterval": "60 seconds",
  "MonitoredInterfaces": [
    "aglt2-rtr-1.local_Ethernet1/51",
    "aglt2-rtr-1.local_Ethernet1/52",
    "aglt2-rtr-2.local_Ethernet1/51",
    "aglt2-rtr-2.local_Ethernet1/52"
  ]
}
```

If things look OK we can set it up as a systemd service

Implementing as a systemd service

You need to edit the systemd service file we created by copying the example: `site-traffic-monitor.service`. The only change needed is edit that file and replace `<INSTALL_LOC>` with the actual install location.

Then you can copy this file (as 'root') to `/etc/systemd/system/site-traffic-monitor.service`

```
sudo cp ${INSTALL_LOC}/site-traffic-monitor.service /etc/systemd/system/site-traffic-monitor.service'
```

You then need to reload systemd, enable the new service and start it

```
sudo systemctl daemon-reload
sudo systemctl enable site-traffic-monitor.service
sudo systemctl start site-traffic-monitor.service
```

While this service runs it should create a new `JSONOUTFILE` every `INTERVAL` seconds. NOTE: set `INTERVAL` at 60 seconds unless technical issues preclude this.

If the location of `JSONOUTFILE` is NOT accessible via a web URL, you will need to have some mechanism to move it to a web accessible location.

Register in CRIC

The last thing to do is register the URL for the `JSONOUTFILE` in CRIC. Each network site in CRIC is shown at [CRIC Network Site](#). Find the appropriate network site and update the monitoring URL with the correct location allowing access to your `JSONOUTFILE`

For example

```
Monitoring URL
Monitoring URL that shows real-time network traffic into and out-of this site
https://head01.aglt2.org/aglt2-netmon.json
```

The WLCG Monitoring Task Force plans to use these URLs to grab the JSON monitoring for all participating sites.