

Mémoire et processeur Nexus Monitor utilisant Python Cli

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Configuration](#)

[Vérification](#)

[Programmer le script](#)

[Utilisation d'EEM](#)

[Utilisation du planificateur](#)

[Dépannage](#)

Introduction

Ce document décrit comment surveiller le processeur et la mémoire avec l'utilisation du module cli python intégré et planifier les notifications syslog.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- [Gestionnaire d'événements intégré](#)
- [Planificateur Nexus](#)

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Nexus 3000 - 7.0(3)I4(7)
- Nexus 9000 - 7.0(3)I7(1)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Configuration

Pour cet exemple, surveillez le processus PTP pour son processeur et sa mémoire.

Assurez-vous de configurer.

```
feature ptp
```

```
feature scheduler
```

Le script python est utilisé pour surveiller le processeur et la mémoire pour un processus particulier qui s'exécute sur le périphérique.

```
import cisco
import sys
from cli import *

def main(process):

mem = ""
cpu = ""

ptp_cpu = cli('sh proc cpu | i '+process)
ptp_mem = cli('sh proc mem | i '+process)

if ptp_cpu:
csplit = ptp_cpu.split()
cpu = csplit[4]
if ptp_mem:
msplit = ptp_mem.split()
mem = msplit[3]

cli('syslog priority notifications msg for '+process+' mem: '+mem+' and cpu '+cpu)

return

if __name__ == "__main__":
main(sys.argv[1])
```

Enregistrez le fichier en tant que **ptp_alert_mem_cpu.py**.

Vérification

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Exécutez le fichier à partir de l'interface de ligne de commande.

Syntaxe pour exécuter le script :

python <fichier script> <nom du processus>

```
Nexus# python bootflash:ptp_alert_mem_cpu.py ptp
```

```
Nexus# show logg last 5
```

```
2018 Dec 13 10:59:30 Nexus %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured from vty by admin on vsh.28744
```

```
2018 Dec 13 11:02:30 Nexus %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured from vty by admin on vsh.28867
```

2018 Dec 13 11:03:37 Nexus %EEM_ACTION-5-NOTIF: for ptp mem: 649089024 and cpu 0.00%

Comme nous l'avons vu ici, cette alerte syslog est déclenchée pour le processus PTP et inclut l'utilisation du processeur et de la mémoire.

La même sortie est vue à partir de l'interface de ligne de commande Nexus.

```
Nexus(config-schedule)# show proc cpu | i ptp  
21037          233          4586          50    0.00%  ptp
```

```
Nexus(config-schedule)# show proc mem | i ptp  
21037  4669440  1018201484649089024 fff201d0/ffff1e65c  ptp
```

Voici un autre exemple qui montre comment le script est utilisé pour tuer un processus dès que la mémoire dépasse le seuil spécifié.

Le script prend deux entrées, le nom du processus et la taille de la mémoire, après quoi le processus doit être exécuté.

```
Nexus# python bootflash:Check_mem_kill_process.py ptp 23423
```

```
Nexus# show logg last 5
```

```
2018 Dec 20 07:00:09 BGL14.1-G.17-N3K-C31108PC-1 %EEM_ACTION-5-NOTIF: Killing ptp mem: 691027968  
2018 Dec 20 07:00:09 BGL14.1-G.17-N3K-C31108PC-1 %VSHD-5-VSHD_SYSLOG_CMD_EXEC: User:admin  
executed the command:run bash  
2018 Dec 20 07:00:09 BGL14.1-G.17-N3K-C31108PC-1 %SYSMGR-2-SERVICE_CRASHED: Service "ptp" (PID  
29107) hasn't caught signal 6 (core will be saved).
```

Programmer le script

Utilisation d'EEM

Ce script EEM est déclenché toutes les minutes et le message syslog est généré.

```
Nexus(config)# event manager applet mem_cpu  
Nexus(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 get-type exact entry-op le  
entry-val 12345 poll-interval 60  
Nexus(config-applet)# action 1.0 syslog priority notifications msg Running_mem_cpu_script  
Nexus(config-applet)# action 2.0 cli command python bootflash:ptp_alert_mem_cpu.py ptp
```

```
Nexus(config-schedule)# show event manager policy internal mem_cpu  
  
Name : mem_cpu  
Policy Type : applet  
Event Specification : event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 get-type exact  
entry-op le entry-val 12345 poll-interval 60  
Action : 1.0,sup:syslog priority notifications msg  
Running_mem_cpu_script;2.0,vsh,sup:command python bootflash:ptp_alert_mem_cpu.py  
Event Specification active on : Active
```

Utilisation du planificateur

Cette tâche de planificateur est déclenchée toutes les minutes, puis le message syslog est généré.

```
Nexus(config)# feature scheduler
Nexus(config)# scheduler job name ptp
Nexus(config-job)# python bootflash:ptp_alert_mem_cpu.py ptp
Nexus(config-job)# exit
```

```
Nexus(config)# scheduler schedule name ptp
Nexus(config-schedule)# job name ptp
Nexus(config-schedule)# time start now repeat 0:0:1
```

Schedule starts from Thu Dec 13 11:21:13 2018

```
Nexus(config-schedule)# show scheduler schedule
```

```
Schedule Name      : ptp
-----
User Name          : admin
Schedule Type      : Run every 0 Days 0 Hrs 1 Mins
Start Time         : Thu Dec 13 11:21:13 2018
Last Execution Time : Thu Dec 13 11:21:13 2018
Last Completion Time: Thu Dec 13 11:21:17 2018
```

```
Execution count    : 1
```

```
-----
Job Name           Last Execution Status
-----
ptp                Success (0)
```

```
Nexus(config-schedule)# sh logg last 5
2018 Dec 13 11:20:19 Nexus %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured from vty by admin on
vsh.29770
2018 Dec 13 11:20:31 Nexus %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured from vty by admin on
vsh.29777
2018 Dec 13 11:21:17 Nexus %EEM_ACTION-5-NOTIF: for ptp mem: 649089024 and cpu 0.00%
```

Dépannage

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.