

Dépanner l'exécution incomplète du script Diagnostics.sh

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Problème](#)

[Solution](#)

Introduction

Ce document décrit la procédure de dépannage de l'exécution de scripts diagnostics.sh incomplets dans Cisco Policy Suite (CPS).

Contribution d'Ullas Kumar E, ingénieur du centre d'assistance technique Cisco.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- CPS

Remarque : Cisco recommande que vous ayez un accès racine privilèges vers CPS CLI.

Composants utilisés

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

- CPS 21.1
- Centos 8.0
- Système d'informatique unifiée (UCS)-B

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. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

Diagnostics.sh est la commande de dépannage de base qui peut être exécutée dans le noeud pcrfclient ou installateur de CPS pour vérifier l'état actuel du système.

Il fournit une liste détaillée des paramètres dans le cadre de la vérification de l'intégrité de CPS.

Ce script s'exécute sur les différents points d'accès, de surveillance et de configuration d'un système CPS en cours d'exécution.

Dans les environnements à haute disponibilité (HA) ou à redondance géographique (GR), le script effectue toujours une vérification ping pour toutes les machines virtuelles (VM) avant toute autre vérification et ajoute à la variable IGNORED_HOSTS celles qui échouent au test ping. Cela permet de réduire le risque d'erreurs de fonction de script.

Exemples :

```
/var/qps/bin/diag/diagnostics.sh -q  
/var/qps/bin/diag/diagnostics.sh --basic_ports --clock_skew
```

Ce sont les contrôles proéminents que ce script fait.

```
--basic_ports : Run basic port checks  
For AIO: 80, 11211, 27017, 27749, 7070, 8080, 8090, 8182, 9091, 9092  
For HA/GR: 80, 11211, 7070, 8080, 8081, 8090, 8182, 9091, 9092, and Mongo DB ports based on /etc/broadh  
--clock_skew : Check clock skew between lb01 and all vms (Multi-Node Environment only)  
--diskspace : Check diskspace  
--get_active_alarms : Get the active alarms in the CPS  
--get_frag_status : Get fragmentation status for Primary members of DBs viz. session_cache, sk_cache, c  
--get_replica_status : Get the status of the replica-sets present in environment. (Multi-Node Environme  
--get_shard_health : Get the status of the sharded database information present in environment. (Multi-  
--get_sharding_status : Get the status of the sharding information present in environment. (Multi-Node  
--get_session_shard_health : Get the session shard health status information present in environment. (M  
--get_peer_status : Get the diameter peer information present in environment. (Multi-Node Environment c  
--get_sharded_replica_status : Get the status of the shards present in environment. (Multi-Node Environ  
--ha_proxy : Connect to HAProxy to check operation and performance statistics, and ports (Multi-Node En  
    http://lbvip01:5540/haproxy?stats  
    http://lbvip01:5540//haproxy-diam?stats  
--help -h : Help - displays this help  
--hostnames : Check hostnames are valid (no underscores, resolvable, in /etc/broadhop/servers) (AIO onl  
--ignored_hosts : Ignore the comma separated list of hosts. For example --ignored_hosts='portal01,porta  
    Default is 'portal01,portal02,portallb01,portallb02' (Multi-Node Environment only)  
--ping_check : Check ping status for all VM  
--policy_revision_status : Check the policy revision status on all QNS, LB, UDC VMs.  
--lwr_diagnostics : Retrieve diagnostics from CPS LWR kafka processes  
--qns_diagnostics : Retrieve diagnostics from CPS java processes  
--qns_login : Check qns user passwordless login  
--quiet -q : Quiet output - display only failed diagnostics  
--radius : Run radius specific checks  
--redis : Run redis specific checks  
--whisper : Run whisper specific checks  
--aido : Run Aido specific checks  
--svn : Check svn sync status between pcrfclient01 & pcrfclient02 (Multi-Node Environment only)  
--tacacs : Check Tacacs server reachability  
--swapspace : Check swap space  
--verbose -v : Verbose output - display *all* diagnostics (by default, some are grouped for readability)  
--virtual_ips : Ensure Virtual IP Addresses are operational (Multi-Node Environment only)  
--vm_allocation : Ensure VM Memory and CPUs have been allocated according to recommendations
```

Problème

Il est possible que, dans certaines situations, l'exécution des scripts `diagnostics.sh` se soit bloquée à un moment donné et qu'elle ne puisse pas se déplacer davantage ou terminer l'exécution du script.

Vous pouvez exécuter le script et observer que le script est coincé à « check forAuto Intelligent DB Operations (AIDO) Status" ne se poursuit pas pour la vérification du numéro de Subversion (SVN) et plus loin.

```
<#root>
```

```
[root@installer ~]#
```

```
diagnostics.sh
```

```
CPS Diagnostics HA Multi-Node Environment
```

```
-----  
Ping check for all VMs...  
Hosts that are not 'pingable' are added to the IGNORED_HOSTS variable...[PASS]  
Checking basic ports for all VMs...[PASS]  
Checking qns passwordless logins for all VMs...[PASS]  
Validating hostnames...[PASS]  
Checking disk space for all VMs...[PASS]  
Checking swap space for all VMs...[PASS]  
Checking for clock skew for all VMs...[PASS]  
Retrieving diagnostics from pcrfclient01:9045...[PASS]  
Retrieving diagnostics from pcrfclient02:9045...[PASS]  
Checking redis server instances status on lb01...[PASS]  
Checking redis server instances status on lb02...[PASS]  
Checking whisper status on all VMs...[PASS]  
Checking AIDO status on all VMs...[PASS]  
.  
.
```

Lorsque vous vérifiez le résultat détaillé de `diagnostics.sh`, il y a une étape pour vérifier l'état SVN, le script ne se déplace pas plus loin. Elle indique que le script `diagnostics.sh` s'est bloqué au niveau de la vérification factorielle.

```
<#root>
```

```
[[32mPASS[0m] AIDO Pass  
[[ -f /var/tmp/aido_extra_info ]]  
cat /var/tmp/aido_extra_info  
There is no provision to check AIDO service status of installer from this host  
/bin/rm -fr /var/tmp/aido_extra_info  
check_all_svn  
++ is_enabled true  
++ [[ ' ' == \t\r\u\e ]]  
++ [[ true != \f\a\l\s\e ]]  
++ echo true  
[[ true == \t\r\u\e ]]  
++ awk '{ $1=""; $2=""; print }'  
  
++ /usr/bin/ssh root@pcrfclient01 -o ConnectTimeout=2 /usr/bin/facter.
```

```
++ grep svn_slave_list
```

Le script se connecte à pcrfclient01 et vérifie la **svn_slave_list** à partir du résultat de la commande **factor**, qui n'est pas complètement exécutée.

Vous pouvez également vous connecter à pcrfclient01 et vérifier si la commande factor s'exécute correctement et affiche le résultat souhaité.

```
<#root>
[root@pcrfclient01 ]#
factor | grep eth

[root@installer ~]# ^C
```

Lorsque vous vérifiez la moyenne de charge de pcrfclient01, elle est très élevée.

```
<#root>
[root@pcrfclient01 pacemaker]#
top

top - 15:34:18 up 289 days, 14:55, 1 user, load average:
2094.68, 2091.77, 2086.36
```

Vérifiez que les processus liés aux facteurs sont bloqués et génèrent une charge moyenne élevée.

```
<#root>
[root@pcrfclient01 ~]#
ps -ef | grep factor | wc -l

2096
```

Solution

La meilleure solution pour supprimer ces processus bloqués et réduire la charge moyenne consiste à redémarrer la machine virtuelle pcrfclient01. La procédure pour effacer les processus bloqués de factor et résoudre le problème bloqué de l'exécution de diagnostics.sh :

Étape 1. Connectez-vous au noeud pcrfclient et exécutez la commande reboot.

```
<#root>
```

```
[root@pcrfclient01 ~]#
```

```
init 6
```

Étape 2. Vérifiez que la machine virtuelle pcrfclient01 est opérationnelle et stable.

```
<#root>
```

```
[root@pcrfclient01 ~]#
```

```
uptime
```

```
10:07:15 up 1 min, 4:09, 1 user, load average: 0.33, 0.33, 0.36
```

```
[root@pcrfclient01 ~]#
```

Étape 3. Vérifiez que la charge moyenne du pcrfclient01 est normale.

```
<#root>
```

```
[root@instapcrfclient01l1er ~]#
```

```
top
```

```
top - 10:07:55 up 1 min, 4:10, 1 user, load average: 0.24, 0.31, 0.35
```

Étape 4. Exécutez diagnostics.sh et vérifiez que l'exécution du script est terminée.

```
<#root>
```

```
[root@instapcrfclient01l1er ~]#
```

```
diagnostics.sh
```

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.