

Fehlerbehebung unvollständige Diagnose.sh Skriptausführung

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Problem](#)

[Lösung](#)

Einleitung

In diesem Dokument wird das Verfahren zur Fehlerbehebung bei unvollständiger diagnostics.sh-Skriptausführung in Cisco Policy Suite (CPS) beschrieben.

Beitrag von Ullas Kumar E, Cisco TAC Engineer.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Linux
- CPS

Hinweis: Cisco empfiehlt, dass Sie Root-Zugriff benötigen. Berechtigungen auf die CPS-CLI.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- CPS 21.1
- Centos 8.0
- Unified Computing System (UCS)-B

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Diagnostics.sh ist der grundlegende Fehlerbehebungsbefehl, der im pcrfclient- oder im Installationsknoten des CPS ausgeführt werden kann, um den aktuellen Status des Systems zu überprüfen.

Es bietet eine detaillierte Liste von Parametern als Teil der Statusprüfung von CPS.

Dieses Skript wird auf den verschiedenen Zugriffs-, Überwachungs- und Konfigurationspunkten eines laufenden CPS-Systems ausgeführt.

In Umgebungen mit hoher Verfügbarkeit (HA) oder Geo-Redundanz (GR) führt das Skript vor allen anderen Prüfungen immer eine Ping-Prüfung für alle virtuellen Systeme (VMs) durch und fügt alle Prüfungen, die den Ping-Test nicht bestehen, der IGNORED_HOSTS-Variablen hinzu. Dadurch wird die Wahrscheinlichkeit von Skriptfunktionsfehlern verringert.

Examples:

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

Dies sind die herausragenden Prüfungen, die dieses Skript durchführt.

```
--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
```

Problem

Es ist möglich, dass in einigen Situationen die Ausführung von diagnostics.sh-Skripten an einem Punkt hängen geblieben ist, und es kann nicht weiter gehen oder die Skriptausführung beenden.

Sie können das Skript ausführen und beobachten, dass das Skript bei "Check forAuto Intelligent DB Operations (AIDO) Status" wird nicht für Subversion Number (SVN)-Prüfung und weiter fortgesetzt.

```
<#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]  
.  
.
```

Wenn Sie die ausführliche Ausgabe von diagnostics.sh überprüfen, gibt es einen Schritt, um den SVN-Status zu überprüfen, das Skript bewegt sich nicht weiter von dort. Es weist darauf hin, dass das Skript diagnostics.sh bei der Faktorprüfung hängen geblieben ist.

```
<#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
```

Das Skript meldet sich bei pcrfclient01 an und überprüft die `svn_slave_list` anhand der `factor`-Befehlsausgabe, die nicht vollständig ausgeführt wird.

Außerdem können Sie sich bei pcrfclient01 anmelden und überprüfen, ob der Faktor-Befehl ordnungsgemäß ausgeführt wird und die gewünschte Ausgabe liefert.

```
<#root>
```

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

```
factor | grep eth
```

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

Wenn Sie den Lastdurchschnitt von pcrfclient01 überprüfen, wird er als sehr hoch angesehen.

```
<#root>
```

```
[root@pcrfclient01 pacemaker]#
```

```
top
```

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

```
2094.68, 2091.77, 2086.36
```

Überprüfen Sie, ob die faktorbezogenen Prozesse hängen bleiben und zu einem hohen Lastdurchschnitt führen.

```
<#root>
```

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

```
ps -ef | grep factor | wc -l
```

```
2096
```

Lösung

Die ultimative Lösung, um diese festgefahrenen Prozesse zu beseitigen und den Lastdurchschnitt zu reduzieren, ist der Neustart des virtuellen pcrfclient01-Systems. Das Verfahren, um festgefahrene Prozesse des Faktors zu löschen und festgefahrene Probleme bei der Ausführung von `diagnostics.sh` zu beheben:

Schritt 1: Melden Sie sich beim pcrfclient-Knoten an, und führen Sie den Befehl `reboot` aus.

```
<#root>
```

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

```
init 6
```

Schritt 2: Stellen Sie sicher, dass die virtuelle pcrfclient01-Maschine aktiv und stabil ist.

```
<#root>
```

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

```
uptime
```

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

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

Schritt 3: Überprüfen Sie, ob der Lastdurchschnitt von pcrfclient01 normal ist.

```
<#root>
```

```
[root@instapcrfclient01lller ~]#
```

```
top
```

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

Schritt 4: Führen Sie diagnostics.sh aus, und stellen Sie sicher, dass die Skriptausführung abgeschlossen ist.

```
<#root>
```

```
[root@instapcrfclient01lller ~]#
```

```
diagnostics.sh
```

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.