

不完全な診断のトラブルシューティング.shスクリプトの実行

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[問題](#)

[解決方法](#)

はじめに

このドキュメントでは、Cisco Policy Suite(CPS)での不完全なdiagnostics.shスクリプトの実行をトラブルシューティングする手順について説明します。

著者 : Cisco TACエンジニア、Ullas Kumar E

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux
- CPS

注 : ルートアクセスが必要です 特権 CPS CLIに追加します。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- CPS 21.1
- Centos 8.0
- ユニファイドコンピューティングシステム(UCS)-B

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな(デフォルト)設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

Diagnostics.shは、CPSのpcrfclientノードまたはインストーラノードで実行して、システムの現在のステータスを確認できる基本的なトラブルシューティングコマンドです。

CPSのヘルスチェックの一環として、パラメータの詳細なリストが提供されます。

このスクリプトは、実行中のCPSシステムのさまざまなアクセス、モニタリング、および設定ポイントに対して実行されます。

高可用性(HA)環境または地理的冗長(GR)環境では、スクリプトは常にすべての仮想マシン(VM)のpingチェックを他のチェックよりも先に実行し、pingテストに失敗したVMをIGNORED_HOSTS変数に追加します。これにより、スクリプト関数のエラーが発生する可能性を減らすことができます。

Examples:

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

このスクリプトが実行する重要なチェックは次のとおりです。

```
--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/broad  
--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 Environm  
--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. (C  
--get_peer_status : Get the diameter peer information present in environment. (Multi-Node Environment  
--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 E  
    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 on  
--ignored_hosts : Ignore the comma separated list of hosts. For example --ignored_hosts='portal01,port  
    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
```

問題

状況によっては、diagnostics.shスクリプトの実行が1箇所でハングし、スクリプトの実行をそれ以上進めたり、終了させたりできない場合があります。

スクリプトを実行すると、スクリプトが「checking forAuto Intelligent DB Operations (AIDO) Status」は、Subversion Number (SVN)チェックとそれ以上には進みません。

```
<#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]
.
.
```

diagnostics.shの詳細な出力を確認する場合は、SVNステータスを確認する手順があります。スクリプトはそこから先には進みません。diagnostics.shスクリプトがフアクタチェックでスタックしたことを示しています。

```
<#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
```

スクリプトはpcrfclient01にログインし、facterコマンドの出力からsvn_slave_listをチェックしますが、これは完全には実行されません。

また、pcrfclient01にログインして、facterコマンドが正しく実行されているかどうかを確認し、必要な出力を表示できます。

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

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

pcrfclient01の負荷平均を調べると、非常に高いことが確認されています。

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

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

ファクタ関連のプロセスがスタックしていて、その結果として高い負荷平均が発生していることを確認します。

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

解決方法

これらのスタック状態のプロセスをクリアし、負荷平均を削減する最終的なソリューションは、pcrfclient01 VMをリブートすることです。Factorの停止したプロセスをクリアし、diagnostics.sh実行のハングした問題を解決する手順は次のとおりです。

ステップ 1 : pcrfclientノードにログインし、rebootコマンドを実行します。

```
<#root>
```

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

```
init 6
```

ステップ 2 : pcrfclient01 VMが起動して安定していることを確認します。

```
<#root>
```

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

```
uptime
```

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

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

ステップ 3 : pcrfclient01の負荷平均が正常であることを確認します。

```
<#root>
```

```
[root@instapcrfclient0111er ~]#
```

```
top
```

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

ステップ 4 : diagnostics.shを実行し、スクリプトの実行が終了したことを確認します。

```
<#root>
```

```
[root@instapcrfclient0111er ~]#
```

diagnostics.sh

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。