

# 불완전한 Diagnostics.sh 스크립트 실행 문제 해결

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[솔루션](#)

---

## 소개

이 문서에서는 CPS(Cisco Policy Suite)에서 불완전한 diagnostics.sh 스크립트 실행 문제를 해결하는 절차에 대해 설명합니다.

기고자: Ullas Kumar E, Cisco TAC 엔지니어

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Linux
- CPS

참고: 루트 액세스 권한이 있어야 합니다. 권한 CPS CLI로 이동합니다.

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 배경 정보

Diagnostics.sh는 pcrfclient 또는 CPS의 설치 관리자 노드에서 실행하여 시스템의 현재 상태를 확인할 수 있는 기본 문제 해결 명령입니다.

CPS의 상태 점검의 일환으로 자세한 매개변수 목록을 제공합니다.

이 스크립트는 실행 중인 CPS 시스템의 다양한 액세스, 모니터링 및 컨피그레이션 포인트에 대해 실행됩니다.

HA(High Availability) 또는 GR(Geo-Redundant) 환경에서 스크립트는 항상 모든 VM(Virtual Machine)에 대해 ping 검사를 수행하고, 다른 검사에 앞서 ping 테스트에 실패한 모든 VM을 IGNORED\_HOSTS 변수에 추가합니다. 이렇게 하면 스크립트 함수 오류가 발생할 가능성을 줄일 수 있습니다.

Examples:

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

이 스크립트는 눈에 띄는 체크 기능을 제공합니다.

```
--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
--swap space : 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
```

## 문제

경우에 따라 diagnostics.sh 스크립트 실행이 한 지점에서 정지되어 더 이상 이동하거나 스크립트 실행을 완료할 수 없습니다.

스크립트를 실행하고 스크립트가 "확인 중"에 머물러 있는지 확인할 수 있습니다. Auto Intelligent DB Operations (AIDO) Status"는 Subversion Number (SVN) 검사 및 그 이상 진행되지 않습니다.

```
<#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.sh의 자세한 출력을 확인할 때 SVN 상태를 확인하는 단계가 있으며 스크립트가 여기에서 더 이상 이동하지 않습니다. 이는 diagnostics.sh 스크립트가 팩터 검사에서 멈춤을 나타냅니다.

```
<#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
```

스크립트는 pcrfclient01에 로그인하고 완전히 실행되지 않은 facter 명령 출력에서 svn\_slave\_list를 확인합니다.

또한 pcrfclient01에 로그인하여 facter 명령이 제대로 실행되고 원하는 출력을 제공하는지 확인할 수 있습니다.

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

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

pcrfclient01의 부하 평균을 확인하면 매우 높게 관찰된다.

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

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

계수와 관련된 프로세스가 고착되어 평균 로드가 높은지 확인합니다.

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

## 솔루션

이러한 중단된 프로세스를 제거하고 로드 평균을 줄이기 위한 궁극적인 솔루션은 pcrfclient01 VM을 재부팅하는 것입니다. 중단된 프로세스의 요소를 지우고 diagnostics.sh 실행의 중단된 문제를 해결하는 절차:

1단계. pcrfclient 노드에 로그인하고 reboot 명령을 실행합니다.

```
<#root>
[root@pcrfclient01 ~]#
init 6
```

2단계. pcrfclient01 VM이 작동 및 안정되었는지 확인합니다.

```
<#root>
[root@pcrfclient01 ~]#
uptime

10:07:15 up 1 min, 4:09, 1 user, load average: 0.33, 0.33, 0.36
[root@pcrfclient01 ~]#
```

3단계. pcrfclient01의 로드 평균이 정상인지 확인합니다.

```
<#root>
[root@instapcrfclient0111er ~]#
top

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

4단계. diagnostics.sh를 실행하고 스크립트 실행이 완료되었는지 확인합니다.

```
<#root>
[root@instapcrfclient0111er ~]#
diagnostics.sh
```



이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.