

Solucionar problemas de execução de script incompleta do Diagnostics.sh

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Problema](#)

[Solução](#)

Introdução

Este documento descreve o procedimento para solucionar problemas de execução de script diagnostics.sh incompleto no Cisco Policy Suite (CPS).

Contribuição de Ullas Kumar E, Engenheiro do TAC da Cisco.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Linux
- CPS

Observação: a Cisco recomenda que você tenha acesso raiz privilégios à CLI do CPS.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- CPS 21.1
- Centos 8,0
- Unified Computing System (UCS)-B

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Diagnostics.sh é o comando básico de solução de problemas que pode ser executado no nó prfclient ou installer do CPS para verificar o status atual do sistema.

Ele fornece uma lista detalhada de parâmetros como parte da verificação de integridade do CPS.

Esse script executa os vários pontos de acesso, monitoramento e configuração de um sistema CPS em execução.

Em ambientes de Alta Disponibilidade (HA) ou Geo-Redundant (GR), o script sempre faz uma verificação de ping para todas as Máquinas Virtuais (VMs) antes de qualquer outra verificação e adiciona qualquer um que falhe no teste de ping à variável IGNORED_HOSTS. Isso ajuda a reduzir a possibilidade de erros de função de script.

Examples:

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

Estas são as verificações proeminentes que este script faz.

```
--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
```

Problema

É possível que, em algumas situações, a execução de scripts `diagnostics.sh` tenha sido interrompida em um ponto e não seja possível ir além ou concluir a execução do script.

Você pode executar o script e observar que o script está travado em "verificandoAuto Intelligent DB Operations (AIDO) Status" não prossegue para verificação do Número de subversão (SVN) e mais.

```
<#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]
```

```
.
```

```
.
```

Quando você verifica a saída detalhada do `diagnostics.sh`, há uma etapa para verificar o status do SVN, o script não se move mais de lá. Ele está indicando que o script `diagnostics.sh` ficou preso na verificação do fator.

```
<#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
```

O script efetua login no pcrfclient01 e verifica a saída do comando **svn_slave_list** a partir do **fator**, que não é completamente executado.

Além disso, você pode fazer login no pcrfclient01 e verificar se o comando fator é executado corretamente e fornece a saída desejada.

```
<#root>
```

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

```
factor | grep eth
```

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

Quando você verifica a média de carga de pcrfclient01, ela é observada como muito alta.

```
<#root>
```

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

```
top
```

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

```
2094.68, 2091.77, 2086.36
```

Verifique se os processos relacionados ao fator estão travados e resultam em uma média de carga alta.

```
<#root>
```

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

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

```
2096
```

Solução

A solução definitiva para limpar esses processos travados e reduzir a média de carga é reinicializar a VM pcrfclient01. O procedimento para limpar processos presos de factor e resolver o problema suspenso de execução de diagnostics.sh:

Etapa 1. Efetue login no nó pcrfclient e execute o comando reboot.

```
<#root>
```

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

```
init 6
```

Etapa 2. Verifique se a VM pcrfclient01 está Ativa e estável.

```
<#root>
```

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

```
uptime
```

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

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

Etapa 3. Verifique se a carga média de pcrfclient01 é normal.

```
<#root>
```

```
[root@instapcrfclient0111ler ~]#
```

```
top
```

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

Etapa 4. Execute diagnostics.sh e verifique se a execução do script foi concluída.

```
<#root>
```

```
[root@instapcrfclient0111ler ~]#
```

```
diagnostics.sh
```

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.