

Risoluzione dei problemi e verifica del fabric con i comandi a linea singola

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Strumenti utilizzati per acquisire le informazioni](#)

[Elenco di tutti i comandi a riga singola](#)

[Ottenere solo gli ID dei nodi delle FOGLIA all'interno dell'infrastruttura:](#)

[Verificare se sono presenti reimpostazioni dell'interfaccia:](#)

[Controlla se l'interfaccia ha la classificazione più alta:](#)

[Trova tutte le modifiche alla topologia STP:](#)

[Verificare se sono presenti interruzioni RPF multicast:](#)

[Verificare se ci sono troppi pacchetti che ottengono Glean:](#)

[Statistiche rilascio QoS:](#)

[Interfaccia scartata:](#)

[Errori FCS \(errori CRC non-stallati\)](#)

[FCS + errori CRC stomped](#)

[Perdite buffer di output](#)

[output errors](#)

[Cancella tutti i contatori di interfaccia](#)

[Problemi della sessione BGP:](#)

[Problemi della sessione OSPF:](#)

[Pacchetti primari reindirizzati alla CPU](#)

[Controllare l'infrastruttura dall'apic in cui viene distribuita una relazione contrattuale specifica](#)

[Controllare se un Encrypt è già stato distribuito e ottenere l'epg corrispondente](#)

[Utilizzo memoria di tutti i nodi nella struttura:](#)

[Verificare le cadute sullo stack \(i pacchetti di eccezione vengono puntati\)](#)

[Convalida contratto](#)

Introduzione

In questo documento vengono descritte le diverse modalità di rilevamento di un problema generale sul fabric.

Prerequisiti

Requisiti

- Cisco raccomanda la conoscenza di ACI
- Conoscenze base di bash

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Dispositivi usati:

- Cisco ACI con versione 4.2(3)

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Strumenti utilizzati per acquisire le informazioni

- Inviato per sostituire: L'uso di "s/<oldword >/<new word>/g" g definisce che è fatto più di una volta.
- Inviare per acquisire le righe dall'inizio alla fine: Utilizzare "/<begin/,/end/p>". Combinare l'opzione -n (noprnt) con il flag di stampa /p per duplicare le funzionalità di grep.
- Sed può essere utilizzato più volte utilizzando ";", ad esempio: potrebbe: sed "s/<oldword >/<nuova parola>/g; s/<oldword2 >/<nuova parola2>/g"
- awk -F '<separatore_campo>' '{print \$2}' In questo esempio specifico, la riga viene divisa in base al FIELD_SEPARATOR definito e viene stampato il secondo blocco delimitato. Entrambe le sintassi fanno esattamente la stessa cosa.
- awk '{ print \$1, \$2 }' stampa i primi due campi di ogni record di input, con uno spazio tra di essi.
- ordinare | uniq Riporta le righe ripetute. Utilizzo delle righe del prefisso -c per il numero di occorrenze.
- sort -nrk <colonna> ordina le righe più in alto. -n per un ordinamento numerico, -k per una chiave in modo da poter modificare la colonna e se si desidera definire il valore più basso è possibile rimuovere -r
- python -m json.tool Visualizza JSON in un formato grazioso.

Elenco di tutti i comandi a riga singola

Ottenere solo gli ID dei nodi delle FOGLIA all'interno dell'infrastruttura:

1. In un elenco:

```
APIC#acidiag fnvread | grep leaf | awk '{print $1}'
```

2. In una riga con le virgole come separatori:

```
APIC#acidiag fnvread | grep leaf | awk '{print $1}' | sed -z 's/\n/,/g;s/,$/\n/'
```

Verificare se sono presenti reimpostazioni dell'interfaccia:

Le informazioni vengono ordinate sulle interfacce con il maggior numero di ripristini.

```
APIC#moquery -c ethpm.PhysIf | egrep "dn|lastLinkStChg|resetCtr" | tr -d "\n" | sed "s/dn/\ndn/g;s/last
```

opzione più lenta

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf "
```

Controllare l'interfaccia con la classificazione più alta:

Per individuare il punto in cui viene ricevuto il maggior numero di traffico:

Eseguire una query sull'infrastruttura per tutte le interfacce con throughput di output su un valore specifico (b). Il valore di m definisce se b è gb, mb o kb.

Per filtrare in base a gb impostare m su 12500000. Per filtrare in base a mb impostare m su 125000. Per filtrare in base a kb impostare m su 125.

```
APIC#bash
```

```
APIC#b=1; m=125000;b=$((b*m)); printf "%-65s %25s\n", "Node/Interface" "Bits/Second"; icurl 'http://loc
```

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf "
```

Trova tutte le modifiche alla topologia STP:

1. Il comando passa in ciascuna foglia e verifica se sono state apportate modifiche recenti alla

topologia e in quali interfacce:

```
APIC#for node in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo "node ID: $node "; fab
```

2. Il comando passa in ciascuna foglia e verifica il conteggio più alto in base alle modifiche PVRSTP e le interfacce visualizzate:

```
APIC#for node in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo "node ID: $node "; fab
```

Verificare se sono presenti interruzioni RPF multicast:

Questa operazione deve essere eseguita per la singola foglia in quel momento e verifica tutte le VRF PIM abilitate per ogni perdita di RPF:

```
APIC#for vrf in `show ip mroute summary vrf all | grep 'IP Multicast Routing Table for VR' | awk '{prin
```

Verificare se ci sono troppi pacchetti che ottengono Glean:

1. Glean ARP fabric:

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf "
```

2. ARP glean Pacchetti ricevuti:

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf "
```

Statistiche rilascio QoS:

Verificare che QoS abbia ricevuto cali per l'intera struttura:

```
APIC#moquery -c qosmIfClass -f 'qosm.IfClass.RxDropPacketsCount!="0"' | egrep "RxDropPacketsCount|dn"
```

Verificare le perdite di trasmissione QoS per l'intero fabric:

```
APIC#moquery -c qosmIfClass -f 'qosm.IfClass.TxDropPacketsCount!="0"' | egrep "TxDropPacketsCount|dn" |
```

Interfaccia scartata:

Errori FCS (errori CRC non-stallati)

```
APIC#moquery -c rmonDot3Stats -f 'rmon.Dot3Stats.fCSErrors>="1"' | egrep "dn|fCSErrors"
```

FCS + errori CRC stomped

```
APIC#moquery -c rmonEtherStats -f 'rmon.EtherStats.cRCAAlignErrors>="1"' | egrep "dn|cRCAAlignErrors"
```

Perdite buffer di output

```
APIC#moquery -c rmonEgrCounters -f 'rmon.EgrCounters.bufferdropkts>="1"' | egrep "dn|bufferdropkts"
```

output errors

```
APIC#moquery -c rmonIfOut -f 'rmon.IfOut.errors>="1"' | egrep "dn|errors"
```

Cancella tutti i contatori di interfaccia

Comando per ottenere un elenco di nodi dell'infrastruttura

```
APIC# acidiag fmvread | egrep " active" | egrep "leaf|spine" | awk '{print $1}' | sed -e 'H;${x;s/\n/,/;x};' | tail -n 101,102,103,204,205,206,301,1001,1002,2001,2002
```

Comando per cancellare i contatori per l'elenco precedente

```
APIC# fabric 101,102,103,204,205,206,301,1001,1002,2001,2002 clear counters interface all
```

Problemi della sessione BGP:

Per controllare se vi sono sessioni BGP che presentano problemi nell'alloggiamento della struttura

```
APIC#moquery -c bgpPeerEntry -f 'bgp.PeerEntry.operSt!="established" and bgp.PeerEntry.dn*"overlay-1"'
```

Per controllare una sessione BGP

```
APIC#moquery -c bgpPeerEntry -f 'bgp.PeerEntry.operSt!="established" | egrep "dn|operSt" | tr -d "\n" |
```

Problemi della sessione OSPF:

Identificare le sessioni che non sono in stato completo.

```
APIC#moquery -c ospfAdjEp -f 'ospf.AdjEp.operSt!="full"' | egrep "dn|peerIp" | tr -d '\n' | sed "s/dn -
```

Identificare le sessioni che si alternano continuamente e ordinare le informazioni in base al conteggio più alto:

```
APIC#moquery -c ospfAdjStats -f 'ospf.AdjStats.stChgCnt!="0"' | egrep "dn|stChgCnt" | tr -d "\n" | tr -
```

Pacchetti primari inviati alla CPU

 Tener conto: questo comando esegue il debug di 500 pacchetti. Per un importo inferiore, modificare il numero dopo -c.

```
LEAF#tcpdump -i kpm_inb -c 500 > /tmp/cpu-dp.txt
```

```
LEAF#cat /tmp/cpu-dp.txt | grep IP | awk '{print $3 , $4 , $5}' | grep -v $HOSTNAME | awk -F ':' '{prin
```

Diretto senza creare un nuovo file:

```
LEAF#tcpdump -i kpm_inb -c 500 | grep IP | awk '{print $3 , $4 , $5}' | grep -v $HOSTNAME | awk -F ':'
```

Controllare l'infrastruttura dall'apic in cui viene distribuita una relazione contrattuale specifica

```
APIC#moquery -c actrlRule -f 'actrl.Rule.sPcTag=="32783" and actrl.Rule.dPcTag=="46" and actrl.Rule.sco
```

Controllare se un Encrypt è già stato distribuito e ottenere l'epg corrispondente

```
APIC#moquery -c l2CktEp -f 'l2.CktEp.encap=="vlan-3018"'
```

Utilizzo memoria di tutti i nodi nella struttura:

```
APIC#bash
```

```
APIC# clear ; echo -e "Node ID\t\tFree Memory\tUsed Memory" ; moquery -c procSysMemHist15min -f 'proc.S
```

Verificare le cadute sullo stack (i pacchetti di eccezione vengono puntati)

Esaminare le perdite di TX. Utilizzare il comando sort -nk 6 -r :

```
APIC# cat istack_debug | egrep "Protocol:|x_pkts_dropped" | tr -d "\n" | sed "s/Protocol/\nProtocol/g"
```

Convalida contratto

Esamina tutte le relazioni per un contratto specifico. Utilizzare lo script che sostituisce il nome del contratto e del tenant:

```
APIC# CONTRACT='brc-<contract-name>'
```

```
APIC# TN='<tenant>'
```

```
#CHECK CONSUMERS
```

```
#To get the count of epgs consuming a contract (excluding vzany consumer):
```

```
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar
```

```
#To list all epg objects consuming a contract (excluding vzany consumers):
```

```
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar
```

```
#To get the count of vzanys consuming a contract:
```

```
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe
```

```
#To list all vzany objects consuming a contract:
```

```
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe
```

```
#CHECK PROVIDERS
```

```
#To get the count of epgs providing a contract (excluding vzany consumer):
```

```
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar
```

```
#To list all epg objects providing a contract (excluding vzany consumers):
```

```
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar
```

```
#To get the count of vzanys providing a contract:
```

```
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe
```

```
#To list all vzany objects providing a contract:
```

```
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe
```

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).