

Solución de problemas y verificación del fabric con comandos de una línea

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Herramientas utilizadas para obtener la información](#)

[Lista de todos los comandos de una sola línea](#)

[Obtenga solo los ID de nodo de las HOJAS dentro del entramado:](#)

[Verifique si hay restablecimientos de interfaz:](#)

[Compruebe la interfaz con la clasificación más alta:](#)

[Buscar todos los cambios de topología STP:](#)

[Asegúrese de que haya caídas de RPF multidifusión:](#)

[Verifique si hay demasiados paquetes que reciben Glean:](#)

[Estadísticas de caída de QoS:](#)

[caídas de la interfaz:](#)

[Errores de FCS \(errores CRC sin comprimir\)](#)

[FCS + errores CRC detenidos](#)

[Descartes del búfer de salida](#)

[errores de salida](#)

[Borrar todos los contadores de interfaz](#)

[Problemas de sesión BGP:](#)

[Problemas de sesión OSPF:](#)

[Paquetes principales que se dirigen a la CPU](#)

[Comprobar el fabric completo desde apic, donde se implementa toda una relación de contrato específica.](#)

[Verifique dónde va está implementado un encap y obtenga el epq correspondiente](#)

[Uso de memoria de todos los nodos en el fabric:](#)

[Verificar caídas en la pila \(los paquetes de excepción son impulsados\)](#)

[Validación de contratos](#)

Introducción

Este documento describe diferentes maneras en que podemos detectar un problema general en el fabric.

Prerequisites

Requirements

- Cisco recomienda conocer la ACI
- Conocimientos básicos de bash

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

Dispositivos utilizados:

- Cisco ACI versión 4.2(3)

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

Herramientas utilizadas para obtener la información

- Usar para sustituir: La utilización "s/<old word >/<new word>/g" g define que se realiza más de una vez.
- Enviar para agarrar líneas de principio a fin: se ha utilizado "/<inicio/,/fin/p>". Combine la opción -n (no imprimir) con el indicador de impresión /p para duplicados de la funcionalidad de grep.
- Sed puede utilizarse más de una vez utilizando ";", por ejemplo: lo haría: sed "s/<old word >/<new word>/g; s/<old word2 >/<new word2>/g"
- awk -F '<field_separator>' '{print \$2}' En este ejemplo específico, dividió la línea por el FIELD_SEPARATOR definido e imprimió el 2º fragmento delimitado. Ambas sintaxis hacen exactamente lo mismo.
- awk '{ print \$1, \$2 }' imprime los dos primeros campos de cada registro de entrada, con un espacio entre ellos.
- clasificar | uniq Informa sobre las líneas repetidas. Usando las líneas de prefijo -c por el número de apariciones.
- sort -nrk <column> ordena las líneas más altas. -n para una ordenación numérica, -k para una clave para que podamos modificar la columna y si desea definir la más baja, puede eliminar -r
- python -m json.tool Muestra JSON en un formato bonito.

Lista de todos los comandos de una sola línea

Obtenga solo los ID de nodo de las HOJAS dentro del entramado:

1. En una lista:

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

2. En una línea con comas como separadores:

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

Verifique si hay restablecimientos de interfaz:

La información se ordena en las interfaces con mayor número de reinicios.

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

opción más lenta

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

Verifique la interfaz con la clasificación más alta:

Para encontrar dónde se recibe la mayor cantidad de tráfico:

Consulte el fabric para todas las interfaces con rendimiento de salida por encima del valor específico (b). El valor de m define si b es gb, mb o kb.

Para filtrar por gb establezca m en 125000000. Para filtrar por mb establezca m en 125000. Para filtrar por kb establezca m en 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 "
```

Buscar todos los cambios de topología STP:

1. El comando entra en cada hoja y verifica si hay cambios recientes en la topología y en qué interfaces:

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

2. El comando entra en cada hoja y verifica el conteo más alto en los cambios de PVRSTP y qué interfaces se ven:

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

Asegúrese de que haya caídas de RPF multidifusión:

Esto debe ser para la hoja individual en el momento y verifica todo el VRF pim habilitado para cualquier caída de RPF:

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

Verifique si hay demasiados paquetes que reciben Glean:

1. El ARP de fabric obtiene :

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

2. ARP obtiene paquetes recibidos:

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

Estadísticas de caída de QoS:

Verifique las caídas recibidas de QoS para todo el fabric :

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

Verifique las caídas de transmisión de QoS para todo el fabric :

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

caídas de la interfaz:

Errores de FCS (errores CRC sin comprimir)

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

FCS + errores CRC detenidos

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

Descartes del búfer de salida

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

errores de salida

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

Borrar todos los contadores de interfaz

Comando para obtener una lista de nodos de fabric

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

Comando para borrar contadores de la lista anterior

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

Problemas de sesión BGP:

Para verificar si hay alguna sesión BGP que tenga problemas en la estructura subyacente

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

Para verificar cualquier sesión BGP

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

Problemas de sesión OSPF:

Identifique las sesiones que no están en estado completo.

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

Identifique las sesiones que cambian constantemente y ordene la información según el recuento más alto:

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

Paquetes principales que se envían a la CPU

 Tenga en cuenta: este comando depura 500 paquetes. Para una cantidad menor, modifique el número después de -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
```

Directamente sin crear un archivo nuevo:

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

Comprobar el fabric completo desde apic, donde se implementa toda una relación de contrato específica.

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

Verifique dónde ya está implementado un encap y obtenga el epg correspondiente

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

Uso de memoria de todos los nodos en el fabric:

```
APIC#bash
```

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

Verificar caídas en la pila (los paquetes de excepción son impulsados)

Revise las caídas de TX. Utilice el comando con sort -nk 6 -r :

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

Validación de contratos

Revise todas las relaciones de un contrato específico. Utilice la secuencia de comandos para sustituir el nombre del contrato y del arrendatario:

```
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&target=children'

#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&target=children'

#To get the count of vzany consuming a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&target=children'

#To list all vzany objects consuming a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&target=children'

#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&target=children'

#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&target=children'

#To get the count of vzany providing a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&target=children'

#To list all vzany objects providing a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&target=children'
```

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).