Resolución de problemas de comandos CLI de APIC NXOS Style

Contenido

Introducción

Este documento describe los pasos para depurar comandos ejecutados desde la CLI de APIC.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- API REST ACI
- Modelo de objetos de ACI

El lector debe tener un conocimiento previo sobre cómo funciona el dispositivo, así como sobre cómo el proceso DME registra sus mensajes.

Estos documentos explican con más detalle el modelo de objetos y ACI APIC:

https://developer.cisco.com/docs/aci/

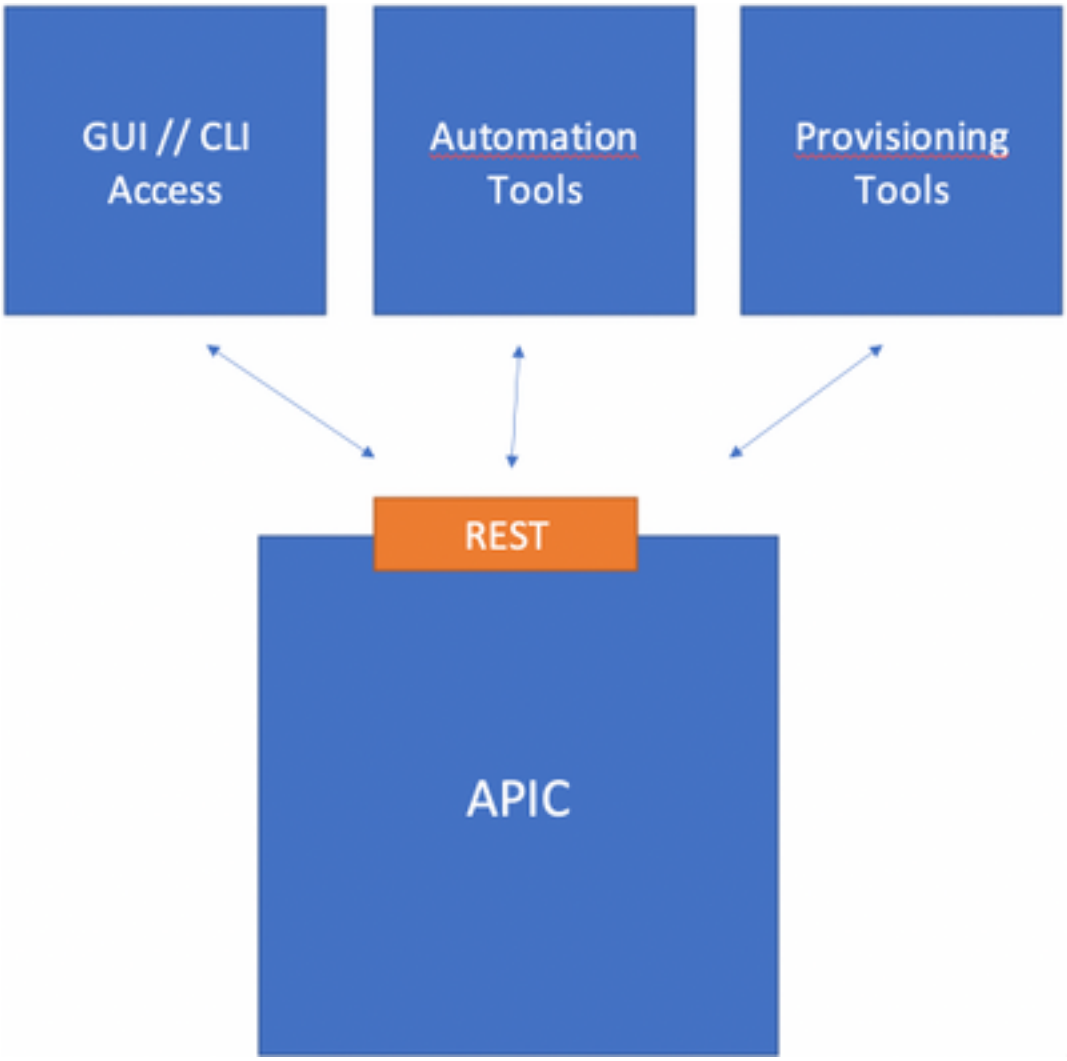
https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/policy-model-guide/b-Cisco-ACI-Policy-Model-Guide.html

Antecedentes

Cisco Application Policy Controller contiene una única API ascendente que se utiliza para la administración de políticas.

Cada interacción basada en políticas con un APIC se reduce a una interacción HTTP/S API Request/Responded. Esto es así para la GUI, las secuencias de comandos de python personalizadas y los comandos CLI que existen tanto en los APIC como en los switches.

Esta imagen resume la forma en que los usuarios y las herramientas interactúan con la API de APIC



Todos los comandos **show** se ejecutan en un APIC invocan la **CLI de estilo NXOS.** Estos comandos activan una serie de scripts python que generan las solicitudes de API necesarias para recopilar la información solicitada. Cuando se recibe una respuesta, se analiza con python y luego se entrega al usuario en un formato bonito.

NGINX

NGINX es un servidor web de código abierto. Cada APIC ejecuta su propio proceso NGINX, que

sirve a la API RESTful. En un APIC, NGINX incluye el registro a través de los archivos **/var/log/dme/log/nginx.bin.log** y **/var/log/dme/log/access.log**.

El archivo **nginx.bin.log** muestra los detalles de todas las solicitudes de API e interacciones de DME.

El archivo **access.log** registra todas las solicitudes de API gestionadas por NGINX.

Dado que todas las solicitudes de API son solicitudes HTTP, se puede hacer referencia a los códigos de respuesta HTTP estándar para la gestión de llamadas de la API NGINX:

- **200** es correcto
- Los códigos **4XX** son errores de cliente
- Los códigos **5XX** son errores del servidor, que es el APIC en este caso

El servicio de señalización

El servicio Decoy sirve una llamada API especial a través de un módulo python alojado fuera de NGINX.

Este servicio:

1. Acepta el comando CLI solicitado
2. Identifica los scripts de Python necesarios para generar las llamadas API REST
3. Envía las llamadas API a la API RESTful
4. Acepta la respuesta
5. Formatea la respuesta
6. Sirve la respuesta con formato al usuario.

Este servicio incluye estos archivos de registro:

- /var/log/dme/log/decoy.log
- /var/log/dme/log/decoy.error.log
- /var/log/dme/log/decoy_server.log

El archivo decoy.log registra los comandos CLI que se ejecutaron.

El archivo **access.log** de nginx utiliza el formato "**POST /decoy/exec/cmd.cli HTTP/1.1**" junto con el código HTTP asociado a la solicitud. . El archivo registra las llamadas de la API REST desde el comando.

> **Nota:** Los registros nginx y decoy mencionados se recopilan en el soporte técnico APIC 3of3.

Solución de problemas de la CLI de estilo NXOS

Seguimiento del comando CLI mediante registros

Ejecutar el comando CLI

Para este ejemplo, el "show controller" se ejecuta a través de la CLI de APIC:

```
APIC-1# show controller
Fabric Name          : ACI-POD1
Operational Size     : 2
Cluster Size         : 3
Time Difference      : 0
Fabric Security Mode : PERMISSIVE

 ID    Pod   Address          In-Band IPv4     In-Band IPv6               OOB IPv4         OOB
IPv6                        Version             Flags  Serial Number     Health
 ----  ----  ---------------  ---------------  -------------------------  ---------------  -----
------------------------  ------------------  -----  ----------------  ------------------
 1*    1     10.0.0.1         0.0.0.0        fc00::1                    192.168.1.1
4.2(6h)             crva-  XXXXXXXXXX fully-fit
 2     1     10.0.0.2         0.0.0.0          fc00::1                    192.168.1.1
        4.2(6h)             crva-  XXXXXXXXXX       fully-fit
Flags - c:Commissioned | r:Registered | v:Valid Certificate | a:Approved | f/s:Failover
fail/success
(*)Current (~)Standby (+)AS
```

El comando devuelve un resultado con un formato bonito.

Verifique decoy.log para la ejecución de CMD

El archivo **decoy.log** se puede verificar para encontrar el CMD "**show controller**" que se invocó:

```
APIC-1# tail /var/log/dme/log/decoy.log
...
...||AUTH COOKIE="XXXXXX"||...
...||CLI: {"option": "server", "loglevel": ["disable"], "cols": 171, "mode": [["exec"]], "port":
51719, "cli": ["show", "controller"]}||...
...||port: 51719||...
...||Mode: [[u'exec']]||...
...||Command: [u'show', u'controller']||...
...||CommandCompleter: add exec||...
...||CommandCompleter: add show||...
...||CommandCompleter: add controller||...
...||last tokens: ['show', 'controller']||...
...||modeCmd: Mode: exec, fulltree: False teminal context is : {'mode-module': 'yaci._cfgroot',
'module': 'show._controllers', 'inherited': False}||.
...||('_%CMD_TERM%_', {'mode-module': 'yaci._cfgroot', 'module': 'show._controllers',
'inherited': False}, {'prompt': '# ', 'mode': [[u'exec']]}, None)||...
...||terminal command module: {"mode-module": "yaci._cfgroot", "module": "show._controllers",
"inherited": false}|...
```

La CLI envía esto como un diccionario, podemos verlo desde la línea resaltada. La versión formateada tendría este aspecto:

```
{
   "option": "server",
   "loglevel": [
       "disable" <-- Can be modified to debug the interaction further.
   ],
   "cols": 171,
   "mode": [     <-- Command ran by admin
       [
           "exec"
       ]
```

```
    ],
    "port": 51719, <-- Random TCP port from session.
    "cli": [      <-- Actual command
        "show",
        "controller"
    ]
}
```

La línea con '_%CMD_TERM%_' muestra el comando ejecutado, junto con el módulo utilizado, en este ejemplo:

```
('_%CMD_TERM%_', {'mode-module': 'yaci._cfgroot', 'module': 'show._controllers', 'inherited':
False}, {'prompt': '# ', 'mode': [[u'exec']]}, None)
```

El módulo CLI debe traducirlo a llamadas API REST.

Comprobar access.log para llamadas API

El archivo **access.log** muestra las llamadas de API recibidas después de que el señuelo procese el CMD:

```
APIC-1# tail /var/log/dme/log/access.log
...
127.0.0.1 - - [24/May/2021:18:43:12 +0000] "POST /decoy/exec/cmd.cli HTTP/1.1" 200 0 "-"
"python-requests/2.7.0..."
127.0.0.1 - - [24/May/2021:18:43:19 +0000] "GET /api/mo/topology/pod-1/node-1/sys.xml HTTP/1.1"
200 1273 "-" "python-requests/2.7.0..."
127.0.0.1 - - [24/May/2021:18:43:19 +0000] "GET /api/mo/uni/fabsslcomm/ifmcertnode-1.xml
HTTP/1.1" 200 2391 "-" "python-requests/2.7.0..."
127.0.0.1 - - [24/May/2021:18:43:19 +0000] "GET /api/mo/uni/fabsslcomm/ifmcertnode-2.xml
HTTP/1.1" 200 2508 "-" "python-requests/2.7.0..."
127.0.0.1 - - [24/May/2021:18:43:19 +0000] "GET /api/mo/topology/pod-1/node-2/sys.xml HTTP/1.1"
200 1265 "-" "python-requests/2.7.0..."
```

Compruebe nginx.bin.log

El archivo *nginx.bin.log* incluye detalles adicionales sobre todas las solicitudes de API gestionadas y debe incluir información sobre la carga útil recibida de varios DME antes de enviarla de vuelta al solicitante.

En el mismo archivo, después de llamar a **decoy/exec/cmd.cli**, se registran varias llamadas de API:

```
admin@APIC-1:log> cat nginx.bin.log| grep "ifmcertnode|sys.xml"
17567||2021-05-24T18:43:19.817094383+00:00||nginx||DBG4||||Request received
/api/mo/topology/pod-1/node-1/sys.xml||../common/src/rest/./Rest.cc||67   bico 11.322
17567||2021-05-24T18:43:19.817184335+00:00||nginx||DBG4||||httpmethod=1; from 127.0.0.1;
url=/api/mo/topology/pod-1/node-1/sys.xml; url options=||../common/src/rest/./Request.cc||133

17567||2021-05-24T18:43:19.817409193+00:00||nginx||DBG4||||Request received
/api/mo/topology/pod-1/node-2/sys.xml||../common/src/rest/./Rest.cc||67
17567||2021-05-24T18:43:19.817466216+00:00||nginx||DBG4||||httpmethod=1; from 127.0.0.1;
url=/api/mo/topology/pod-1/node-2/sys.xml; url options=||../common/src/rest/./Request.cc||133

17567||2021-05-24T18:43:19.817589102+00:00||nginx||DBG4||||Request received
/api/mo/uni/fabsslcomm/ifmcertnode-1.xml||../common/src/rest/./Rest.cc||67
17567||2021-05-24T18:43:19.817641070+00:00||nginx||DBG4||||httpmethod=1; from 127.0.0.1;
url=/api/mo/uni/fabsslcomm/ifmcertnode-1.xml; url options=||../common/src/rest/./Request.cc||133
```

```
17567||2021-05-24T18:43:19.819268449+00:00||nginx||DBG4||||Request received
/api/mo/uni/fabsslcomm/ifmcertnode-2.xml||../common/src/rest/./Rest.cc||67
17567||2021-05-24T18:43:19.819340589+00:00||nginx||DBG4||||httpmethod=1; from 127.0.0.1;
url=/api/mo/uni/fabsslcomm/ifmcertnode-2.xml; url options=||../common/src/rest/./Request.cc||133
...
```

Volver a ejecutar las llamadas API individuales

Los access.logs contienen una lista de exactamente qué llamadas API se han enviado para su procesamiento.

El objetivo de este paso es volver a ejecutar cada llamada API para determinar:

1. ¿Hay algún problema con la llamada de la API en sí?
2. ¿Hay alguna diferencia entre una llamada API exitosa y una fallida?

Por ejemplo, una de las llamadas API generadas a partir del comando "show controllers" es:

```
127.0.0.1 - - [24/May/2021:18:43:19 +0000] "GET /api/mo/topology/pod-1/node-1/sys.xml HTTP/1.1"
200 1273 "-" "python-requests/2.7.0..."
```

Vía icurl

Esta solicitud de API se puede volver a ejecutar con el uso de icurl en un APIC:

```
icurl 'http://localhost:7777/




        '
```

> **Nota:** El puerto 7777 se utiliza específicamente para permitir que el APIC se consulte a sí mismo.

Con la llamada específica como ejemplo:

```
APIC-1# bash
admin@APIC-1:~> icurl 'http://localhost:7777/api/mo/topology/pod-1/node-1/sys.xml'

<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="1">
<topSystem address="10.0.0.1" bootstrapState="none" childAction="" dn="topology/pod-1/node-
1/sys" ... />
</imdata>
```

La cantidad y complejidad de las llamadas API requeridas para cada comando CLI de NXOS puede diferir enormemente. s En todos los casos, las consultas se pueden reproducir a través de icurl para la validación de solicitudes y respuestas.

Vía moquery

A partir de las llamadas API anteriores mostradas, el módulo analiza la salida del comando ‘show controller’ con la información de los MO solicitados.

```
admin@APIC-1:~> moquery -d topology/pod-1/node-1/sys -o xml
...
 <topSystem address="10.0.0.1" dn="topology/pod-1/node-1/sys" fabricDomain="ACI-POD1"
 id="1" inbMgmtAddr="0.0.0.0" inbMgmtAddr6=""
 oobMgmtAddr="192.168.1.1" oobMgmtAddr6="" podId="1" serial="..."
 state="in-service" version="4.2(6h)"/>


admin@APIC-1:~> moquery -d topology/pod-1/node-2/sys -o xml
...
 <topSystem address="10.0.0.2" dn="topology/pod-1/node-2/sys" fabricDomain="ACI-POD1"
 id="2" inbMgmtAddr="0.0.0.0" inbMgmtAddr6=""
 oobMgmtAddr="192.168.1.2" oobMgmtAddr6="" podId="1" serial="..."
 state="in-service" version="4.2(6h)"/>


admin@APIC-1:~> moquery -d uni/fabsslcomm/ifmcertnode-1  -o xml
...
 <pkiFabricNodeSSLCertificate nodeId="1" serialNumber="..." subject="/serialNumber=PID:APIC-
SERVER-M2 SN:..." .../>


admin@APIC-1:~> moquery -d uni/fabsslcomm/ifmcertnode-2  -o xml
...
 <pkiFabricNodeSSLCertificate nodeId="2" serialNumber="..." subject="/serialNumber=PID:APIC-
SERVER-M2 SN:..." .../>
```

Vuelva a ejecutar el comando CLI mediante Python

Como se mencionó, todos los comandos "show" son en realidad una llamada API REST a una serie de scripts python.

Debido a esto, un usuario técnicamente puede invocar manualmente el script python que ejecuta el cmd. El objetivo aquí es ver si Python brinda más información sobre por qué un CMD específico tiene problemas.

Para cualquier comando "**show**" que tenga un problema, invoque el comando a través de Python para comparar un APIC exitoso con uno fallido:

```
apic1# ${PYTHON} -m pyclient.remote exec terminal ${COLUMNS} <some show command>
```

Ejemplo de ejecución:

```
apic1# ${PYTHON} -m pyclient.remote exec terminal ${COLUMNS} show switch
apic1# ${PYTHON} -m pyclient.remote exec terminal ${COLUMNS} show controller
```

Ejemplo donde el script directo de Python proporciona información de depuración adicional sobre una falla:

```
a-apic1# ${PYTHON} -m pyclient.remote exec terminal ${COLUMNS} show switch

Process Process-2:
Traceback (most recent call last):
File "/usr/lib64/python2.7/multiprocessing/process.py", line 267, in _bootstrap
self.run()
```

```
File "/usr/lib64/python2.7/multiprocessing/process.py", line 114, in run
self._target(*self._args, **self._kwargs)
File "/controller/yaci/execmode/show/_switch_nodes.py", line 75, in _systemQuery
mo = ctx.moDir.lookupByDn(dn)
File "/controller/ishell/cobra/mit/access.py", line 80, in lookupByDn
mos = self.query(dnQuery)
File "/controller/yaci/pyclient/local.py", line 36, in query
raise e
QueryError: Unable to deliver the message, Resolve timeout from (type/num/svc/shard) =
switch:205:4:0

ID Pod Address In-Band IPv4 In-Band IPv6 OOB IPv4 OOB IPv6 Version Flags Serial Number Name
---- ---- --------------- --------------- ------------------------ --------------- ------------
------------ ------------------ ----- ---------------- ------------------
101 1 10.0.40.65 192.168.2.231 :: 192.168.1.101 :: n9000-14.2(6o) aliv XXXXXXXXXXX leaf101
...
205 ...



206 2 10.0.156.65 192.168.2.236 :: 192.168.1.106 :: n9000-14.2(6l) aliv XXXXXXXXXXX leaf206
...

Flags - a:Active | l/s:Leaf/Spine | v:Valid Certificate | i:In-Service
```

Modificación de objetos de seguimiento mediante APIC CLI

Creación de objetos

El APIC utiliza **configure terminal** para modificar el MIT de una manera similar a NXOS.

Los resultados de estas operaciones se pueden ver en el archivo **decoy.log**.

Ejemplo de Creación de VRF:

```
APIC-1# configure terminal
APIC-1(config)# tenant TestTn
APIC-1(config-tenant)# vrf context Test-CTX-1
APIC-1(config-tenant-vrf)#
```

Salida de **decoy.log**:

```
'18279||2021-05-25 22:52:11,877||decoy||INFO||AUTH
COOKIE="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0
In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVta
CbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-
OYBrwg"||/mgmt/opt/controller/decoy/decoy/_app.py||32'
'18279||2021-05-25 22:52:11,878||decoy||INFO||CLI: {"option": "server", "loglevel": ["disable"],
"cols": 100, "mode": [["exec"], ["configure", "terminal"]], "port": 60003, "cli": ["tenant",
"TestTn"]}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||88'
'18279||2021-05-25 22:52:11,878||decoy||INFO||port:
60003||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||107'
'18279||2021-05-25 22:52:11,879||decoy||INFO||Mode: [[u'exec'], [u'configure',
```

u'terminal']]||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||120'
'18279||2021-05-25 22:52:11,879||decoy||INFO||Command: [u'tenant',
u'**TestTn**']||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||121'
'18279||2021-05-25 22:52:11,879||decoy||DEBUG||CommandCompleter: add
exec||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18279||2021-05-25 22:52:11,880||decoy||DEBUG||CommandCompleter: add
configure||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18279||2021-05-25 22:52:11,880||decoy||DEBUG||OptionKeyword: add
terminal||/mgmt/opt/controller/yaci/yaci/_completer.py||1036'
'11818||2021-05-25 22:52:11,910||decoy||DEBUG||CommandCompleter: add
tenant||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'11818||2021-05-25 22:52:11,911||decoy||DEBUG||ArgCompleter: add
TestTn||/mgmt/opt/controller/yaci/yaci/_completer.py||531'
'11818||2021-05-25 22:52:11,912||decoy||INFO||last tokens: ['tenant',
u'TestTn']||/mgmt/opt/controller/yaci/yaci/_ctx.py||617'
'11818||2021-05-25 22:52:11,912||decoy||INFO||TID: 98||CLI Command: 'tenant
TestTn'||/mgmt/opt/controller/yaci/yaci/_transaction.py||67'

'11818||2021-05-25 22:52:14,331||decoy||DEBUG||[commit]: <?xml version="1.0" **encoding="UTF-8"?>**

status='created,modified'></fvTenant>||/mgmt/opt/controller/yaci/yaci/_ctx.py||1024'

'11818||2021-05-25 22:52:14,349||decoy||INFO||CLIENT-HEADERS: {'APIC-Client': u'tenant TestTn',
'Cookie': 'APIC-
cookie=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxM
jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-OYBrwg',
'Client_Name': 'APIC-CLI', 'Request-Tag':
'tag0'}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||31'
'11818||2021-05-25 22:52:14,350||decoy||INFO||Starting new HTTP connection (1):
127.0.0.1||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||203'
'11818||2021-05-25 22:52:14,381||decoy||DEBUG||" POST /api/mo/uni/tn-TestTn.xml HTTP/1.1" 200
70||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||383'

'18279||2021-05-25 22:52:14,383||decoy||DEBUG||(None, {}, {'prompt': '(config-tenant)# ',
'mode': [[u'exec'], [u'configure', u'terminal'], ['tenant', u'TestTn']]},
None)||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||144'

'18280||2021-05-25 22:52:48,569||decoy||INFO||AUTH
COOKIE="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0
In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVta
CbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-
OYBrwg"||/mgmt/opt/controller/decoy/decoy/_app.py||32'
'18280||2021-05-25 22:52:48,571||decoy||INFO||CLI: {"option": "server", "loglevel": ["disable"],

"cols": 100, "mode": [["exec"], ["configure", "terminal"], ["tenant", "TestTn"]], "port": 60821, "cli": ["vrf", "context", "Test-CTX-1"]}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||88'

'18280||2021-05-25 22:52:48,571||decoy||INFO||port: 60821||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||107'
'18280||2021-05-25 22:52:48,572||decoy||INFO||Mode: [[u'exec'], [u'configure', u'terminal'], [u'tenant', u'TestTn']]||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||120'
'18280||2021-05-25 22:52:48,572||decoy||INFO||Command: [u'vrf', u'context', u' Test-CTX-1']||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||121'

'18280||2021-05-25 22:52:48,572||decoy||DEBUG||CommandCompleter: add exec||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18280||2021-05-25 22:52:48,572||decoy||DEBUG||CommandCompleter: add configure||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18280||2021-05-25 22:52:48,573||decoy||DEBUG||OptionKeyword: add terminal||/mgmt/opt/controller/yaci/yaci/_completer.py||1036'
'18280||2021-05-25 22:52:48,574||decoy||DEBUG||CommandCompleter: add tenant||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18280||2021-05-25 22:52:48,574||decoy||DEBUG||ArgCompleter: add TestTn||/mgmt/opt/controller/yaci/yaci/_completer.py||531'
'13482||2021-05-25 22:52:48,599||decoy||DEBUG||CommandCompleter: add vrf||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'13482||2021-05-25 22:52:48,600||decoy||DEBUG||Keyword: add context||/mgmt/opt/controller/yaci/yaci/_completer.py||983'
'13482||2021-05-25 22:52:48,601||decoy||DEBUG||ArgCompleter: add Test-CTX-1||/mgmt/opt/controller/yaci/yaci/_completer.py||531'
'13482||2021-05-25 22:52:48,602||decoy||INFO||last tokens: ['vrf', 'context', u'Test-CTX-1']||/mgmt/opt/controller/yaci/yaci/_ctx.py||617'
'13482||2021-05-25 22:52:48,602||decoy||INFO||TID: 105||CLI Command: 'vrf context Test-CTX-1'||/mgmt/opt/controller/yaci/yaci/_transaction.py||67'

'13482||2021-05-25 22:52:51,393||decoy||DEBUG||[commit]: <?xml version="1.0" encoding="UTF-8"?>

status='created,modified' name='

Test-CTX-1'></fvCtx>||/mgmt/opt/controller/yaci/yaci/_ctx.py||1024'

'13482||2021-05-25 22:52:51,405||decoy||INFO||CLIENT-HEADERS: {'APIC-Client': u'tenant TestTn; vrf context Test-CTX-1', 'Cookie': 'APIC-cookie=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxMjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaCbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWwO64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-OYBrwg', 'Client_Name': 'APIC-CLI', 'Request-Tag': 'tag0'}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||31'
'13482||2021-05-25 22:52:51,407||decoy||INFO||Starting new HTTP connection (1):

```
127.0.0.1||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||203'
'13482||2021-05-25 22:52:51,452||decoy||DEBUG||" POST /api/mo/uni/tn-TestTn/ctx-Test-CTX-1.xml
HTTP/1.1" 200 70||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||383'



'18280||2021-05-25 22:52:51,455||decoy||DEBUG||(None, {}, {'prompt': '(config-tenant-vrf)# ',
'mode': [[u'exec'], [u'configure', u'terminal'], [u'tenant', u'TestTn'], ['vrf', 'context',
u'Test-CTX-1']]}, None)||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||144'
```

Observe los objetos fvTenant y fvCtx establecidos con status="created,modified".Eliminación de objetosPodemos revisar las operaciones REST involucradas cuando se elimina cualquier MO.Ejemplo de Borrado de VRF:

```
APIC-1# configure terminal
APIC-1(config)# tenant TestTn
APIC-1(config-tenant)# no vrf context Test-CTX-1
APIC-1(config-tenant-vrf)#
```

Salida de decoy.log:

```
'18279||2021-05-25 23:32:40,009||decoy||INFO||AUTH
COOKIE="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0
In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMU1lhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVta
CbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-
OYBrwg"||/mgmt/opt/controller/decoy/decoy/_app.py||32'
'18279||2021-05-25 23:32:40,010||decoy||INFO||CLI: {"option": "server", "loglevel": ["disable"],
"cols": 100, "mode": [["exec"], ["configure", "terminal"]], "port": 32789, "cli": ["tenant",
"TestTn"]}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||88'
'18279||2021-05-25 23:32:40,011||decoy||INFO||port:
32789||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||107'
'18279||2021-05-25 23:32:40,012||decoy||INFO||Mode: [[u'exec'], [u'configure',
u'terminal']]||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||120'
'18279||2021-05-25 23:32:40,012||decoy||INFO||Command: [u'tenant',
u'TestTn']||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||121'
'18279||2021-05-25 23:32:40,012||decoy||DEBUG||CommandCompleter: add
exec||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18279||2021-05-25 23:32:40,013||decoy||DEBUG||CommandCompleter: add
configure||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18279||2021-05-25 23:32:40,013||decoy||DEBUG||OptionKeyword: add
terminal||/mgmt/opt/controller/yaci/yaci/_completer.py||1036'
'27163||2021-05-25 23:32:40,042||decoy||DEBUG||CommandCompleter: add
tenant||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'27163||2021-05-25 23:32:40,043||decoy||DEBUG||ArgCompleter: add
TestTn||/mgmt/opt/controller/yaci/yaci/_completer.py||531'
'27163||2021-05-25 23:32:40,043||decoy||INFO||last tokens: ['tenant',
u'TestTn']||/mgmt/opt/controller/yaci/yaci/_ctx.py||617'
'27163||2021-05-25 23:32:40,044||decoy||INFO||TID: 106||CLI Command: 'tenant
TestTn'||/mgmt/opt/controller/yaci/yaci/_transaction.py||67'

'27163||2021-05-25 23:32:42,467||decoy||DEBUG||[commit]: <?xml version="1.0" encoding="UTF-8"?>
<fvTenant name='TestTn'
status='created,modified'></fvTenant>||/mgmt/opt/controller/yaci/yaci/_ctx.py||1024'
'27163||2021-05-25 23:32:42,485||decoy||INFO||CLIENT-HEADERS: {'APIC-Client': u'tenant TestTn',
'Cookie': 'APIC-
cookie=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
```

```
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxM
jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-OYBrwg',
'Client_Name': 'APIC-CLI', 'Request-Tag':
'tag0'}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||31'
'27163||2021-05-25 23:32:42,487||decoy||INFO||Starting new HTTP connection (1):
127.0.0.1||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||203'
'27163||2021-05-25 23:32:42,517||decoy||DEBUG||"POST /api/mo/uni/tn-TestTn.xml HTTP/1.1" 200
70||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||383'
'18279||2021-05-25 23:32:42,520||decoy||DEBUG||(None, {}, {'prompt': '(config-tenant)# ',
'mode': [[u'exec'], [u'configure', u'terminal'], ['tenant', u'TestTn']]},
None)||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||144'

'18280||2021-05-25 23:32:53,702||decoy||INFO||AUTH
COOKIE="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0
In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVta
CbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-
OYBrwg"||/mgmt/opt/controller/decoy/decoy/_app.py||32'
'18280||2021-05-25 23:32:53,704||decoy||INFO||CLI: {"option": "server", "loglevel": ["disable"],
"cols": 100, "mode": [["exec"], ["configure", "terminal"], ["tenant", "TestTn"]], "port": 32805,
"cli": ["no", "vrf", "context", "Test-CTX-
1"]}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||88'
'18280||2021-05-25 23:32:53,704||decoy||INFO||port:
32805||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||107'
'18280||2021-05-25 23:32:53,705||decoy||INFO||Mode: [[u'exec'], [u'configure', u'terminal'],
[u'tenant', u'TestTn']]||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||120'
'18280||2021-05-25 23:32:53,705||decoy||INFO||Command: [u'no', u'vrf', u'context', u'Test-CTX-
1']||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||121'
'18280||2021-05-25 23:32:53,705||decoy||DEBUG||CommandCompleter: add
exec||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18280||2021-05-25 23:32:53,706||decoy||DEBUG||CommandCompleter: add
configure||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18280||2021-05-25 23:32:53,706||decoy||DEBUG||OptionKeyword: add
terminal||/mgmt/opt/controller/yaci/yaci/_completer.py||1036'
'18280||2021-05-25 23:32:53,708||decoy||DEBUG||CommandCompleter: add
tenant||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'18280||2021-05-25 23:32:53,708||decoy||DEBUG||ArgCompleter: add
TestTn||/mgmt/opt/controller/yaci/yaci/_completer.py||531'
'27771||2021-05-25 23:32:53,736||decoy||DEBUG||CommandCompleter: add
vrf||/mgmt/opt/controller/yaci/yaci/_completer.py||255'
'27771||2021-05-25 23:32:53,737||decoy||DEBUG||Keyword: add
context||/mgmt/opt/controller/yaci/yaci/_completer.py||983'
'27771||2021-05-25 23:32:53,738||decoy||DEBUG||ArgCompleter: add Test-CTX-
1||/mgmt/opt/controller/yaci/yaci/_completer.py||531'
'27771||2021-05-25 23:32:53,738||decoy||INFO||last tokens: ['no', 'vrf', 'context', u'Test-CTX-
1']||/mgmt/opt/controller/yaci/yaci/_ctx.py||617'
'27771||2021-05-25 23:32:53,738||decoy||INFO||TID: 113||CLI Command: 'no vrf context Test-CTX-
1'||/mgmt/opt/controller/yaci/yaci/_transaction.py||67'
'27771||2021-05-25 23:32:53,740||decoy||INFO||CLIENT-HEADERS: {'APIC-Client': u'tenant TestTn;
no vrf context Test-CTX-1', 'Cookie': 'APIC-
cookie=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxM
```

jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMU1lhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-OYBrwg',
'Client_Name': 'APIC-CLI', 'Request-Tag':
'tag0'}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||31'
'27771||2021-05-25 23:32:53,743||decoy||INFO||Starting new HTTP connection (1):
127.0.0.1||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||203'
'27771||2021-05-25 23:32:53,750||decoy||DEBUG||"GET /api/mo/uni/tn-TestTn.xml?target-subtree-
class=l3extRsEctx&query-target-filter=eq(l3extRsEctx.tnFvCtxName,%22Test-CTX-1%22)&query-
target=subtree HTTP/1.1" 200 70||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||383'
'27771||2021-05-25 23:32:53,753||decoy||DEBUG||dnListLen:
0||/mgmt/opt/controller/yaci/lib/utils/l3ext.py||97'

'27771||2021-05-25 23:32:56,375||decoy||DEBUG||[commit]: <?xml version="1.0" encoding="UTF-8"?>
<fvCtx status='deleted' name='Test-CTX-
1'></fvCtx>||/mgmt/opt/controller/yaci/yaci/_ctx.py||1024'
'27771||2021-05-25 23:32:56,386||decoy||INFO||CLIENT-HEADERS: {'APIC-Client': u'tenant TestTn;
no vrf context Test-CTX-1', 'Cookie': 'APIC-
cookie=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxM
jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMU1lhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
O64LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFZb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-OYBrwg',
'Client_Name': 'APIC-CLI', 'Request-Tag':
'tag0'}||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||31'
'27771||2021-05-25 23:32:56,426||decoy||DEBUG||"POST /api/mo/uni/tn-TestTn/ctx-Test-CTX-1.xml
HTTP/1.1" 200 70||/mgmt/opt/controller/decoy/decoy-env/lib/python2.7/site-
packages/requests/packages/urllib3/connectionpool.py||383'
'18280||2021-05-25 23:32:56,428||decoy||DEBUG||(None, {}, {'prompt': '(config-tenant)# ',
'mode': [[u'exec'], [u'configure', u'terminal'], [u'tenant', u'TestTn']]},
None)||/mgmt/opt/controller/decoy/apps/execserver/execapp.py||144'

Observe el conjunto de objetos fvCtx con status="delete".

Referencias y enlaces útiles

- Programabilidad de ACI
- Guía del modelo de políticas de Cisco ACI
- Compatibilidad de Cisco ACI con límite de velocidad NGINX

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).