Risoluzione dei problemi relativi ai comandi CLI APIC NXOS Style

Sommario

Introduzione

In questo documento viene descritto come eseguire il debug dei comandi eseguiti dalla CLI di APIC.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- API REST ACI
- ACI Object Model

Il lettore deve avere una conoscenza precedente su come funziona il DME, anche su come il processo DME registra i loro messaggi.

In questi documenti vengono fornite informazioni più dettagliate sul modello ACI APIC e Object:

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

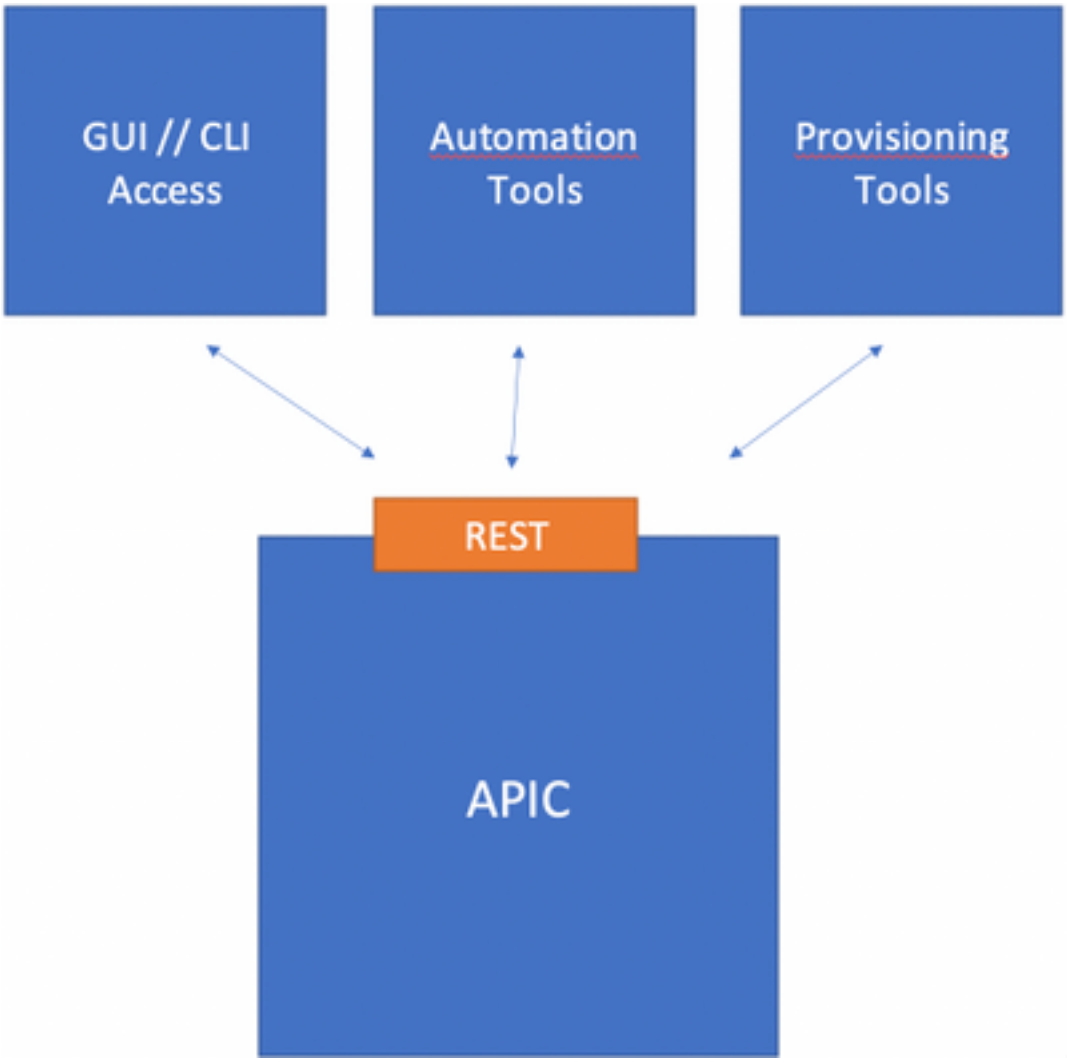
https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/policy-model-guide/b-Cisco-ACI-Policy-Model-Guide.html

Premesse

Il controller dei criteri dell'applicazione Cisco contiene un'unica API Northbound utilizzata per la gestione dei criteri.

Ogni interazione basata su criteri con un APIC si riduce a un'interazione HTTP/S API Request/Responded. Ciò vale per la GUI, gli script Python personalizzati e i comandi CLI presenti sugli APIC e sugli switch.

Questa immagine riepiloga il modo in cui gli utenti e gli strumenti interagiscono con l'API APIC



Tutti i comandi **show** eseguiti su un APIC richiamano la **CLI NXOS Style.** Questi comandi attivano una serie di script Python che generano le richieste API necessarie per raccogliere le informazioni richieste. Quando si riceve una risposta, questa viene analizzata con python e quindi consegnata all'utente in un formato grazioso.

NGINX

NGINX è un server Web open source. Ogni APIC esegue il proprio processo NGINX che serve l'API RESTful. Su un'APIC, NGINX include la registrazione tramite i file **/var/log/dme/log/nginx.bin.log** e **/var/log/dme/log/access.log**.

Il file **nginx.bin.log** mostra i dettagli di tutte le richieste API e le interazioni DME.

Il file **access.log** registra ogni richiesta API gestita da NGINX.

Poiché tutte le richieste API sono richieste HTTP, è possibile fare riferimento ai codici di risposta HTTP standard per la gestione delle chiamate API NGINX:

- **Il** codice **200** è OK
- I codici **4XX** sono errori client
- **I** codici **5XX** sono errori del server, in questo caso l'APIC

Servizio Decoy

Il servizio Decoy serve una chiamata API speciale tramite un modulo Python ospitato al di fuori di NGINX.

Questo servizio:

1. Accetta il comando CLI richiesto
2. Identifica gli script python necessari per compilare le chiamate dell'API REST
3. Invia le chiamate API all'API RESTful
4. Accetta la risposta
5. Formatta la risposta
6. Invia la risposta formattata all'utente.

Il servizio include i seguenti file di registro:

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

Il file decoy.log registra i comandi CLI eseguiti.

Il file **access.log** di Index utilizza il formato **"POST /decoy/exec/cmd.cli HTTP/1.1"** insieme al codice HTTP associato alla richiesta. . Il file registra le chiamate dell'API REST dal comando.

> **Nota:** I registri di inginx e decoy citati sono raccolti nell'APIC 3of3 Techsupport.

Risoluzione dei problemi relativi alla CLI di stile NXOS

Tracciare il comando CLI tramite i log

Eseguire il comando CLI

Per questo esempio, il comando "show controller" viene emesso dalla CLI di 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
```

Il comando restituisce un output formattato.

Controllare decoy.log per l'esecuzione di CMD

Il file **decoy.log** può essere controllato per trovare il CMD "**show controller**" richiamato:

```
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 lo invia come dizionario, lo possiamo vedere dalla riga evidenziata. La versione formattata avrà il seguente aspetto:

```
{
   "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 riga con '_%CMD_TERM%_' mostra il comando eseguito, insieme al modulo utilizzato, in questo esempio:

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

Il modulo CLI deve quindi tradurlo in chiamate API REST.

Controllare access.log per le chiamate API

Il file **access.log** visualizza le chiamate API ricevute dopo che il comando decoy ha elaborato il 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..."
```

Controlla nginx.bin.log

Il file *nginx.bin.log* include dettagli aggiuntivi su tutte le richieste API gestite e deve includere informazioni sul payload ricevuto da vari DME prima di essere inviato al richiedente.

Nello stesso file, dopo la chiamata a **decoy/exec/cmd.cli**, vengono registrate diverse chiamate 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
...
```

Rieseguire le singole chiamate API

Access.logs contiene un elenco delle chiamate API inviate per l'elaborazione.

L'obiettivo di questo passaggio è eseguire nuovamente ogni chiamata API per determinare:

1. Esiste un problema con la chiamata API stessa?
2. C'è qualcosa di diverso in una chiamata API riuscita rispetto a una chiamata non riuscita?

Ad esempio, una delle chiamate API generate dal comando "show controller" è:

```
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..."
```

Via icurlo

Questa richiesta API può essere rieseguita con l'istruzione use su un APIC:

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



        '
```

> **Nota:** La porta 7777 viene utilizzata specificamente per consentire all'APIC di eseguire query su se stesso.

Con la chiamata specifica come esempio:

```
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 quantità e la complessità delle chiamate API richieste per ogni comando CLI di NXOS può variare notevolmente. s. In ogni caso, le query possono essere rieseguite tramite l'ICURL per la convalida di richieste e risposte.

Tramite moquery

Dalle precedenti chiamate API mostrate, il modulo analizza l'output del comando "show controller" con le informazioni dei MO richiesti.

```
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:..." .../>
```

Eseguire nuovamente il comando CLI tramite Python

Come accennato, tutti i comandi "show" sono in realtà una chiamata API REST a una serie di script Python.

Per questo motivo, un utente può tecnicamente richiamare manualmente lo script python che esegue il cmd. L'obiettivo qui è quello di vedere se Python fornisce maggiori informazioni sul motivo per cui un CMD specifico ha dei problemi.

Per qualsiasi comando "**show**" che abbia un problema, richiamare il comando tramite Python per confrontare un APIC riuscito con uno non riuscito:

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

Esempi di esecuzioni:

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

Esempio in cui lo script Python diretto fornisce ulteriori informazioni di debug su un errore:

```
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 XXXXXXXXXX leaf101
...
205 ...



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

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

Trace Object Modification via APIC CLI

Creazione di oggetti

L'APIC utilizza il **terminale di configurazione** per modificare il MIT in modo simile a NXOS.

I risultati di queste operazioni possono essere visualizzati nel file **decoy.log**.

Esempio di creazione di VRF:

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

Output da **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=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'
'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'
```

Osservare gli oggetti fvTenant e fvCtx impostati entrambi con status="creating,modified".Eliminazione oggettoÈpossibile esaminare le operazioni REST coinvolte quando un MO viene eliminato.Esempio di eliminazione 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)#
```

Output da decoy.log:

```
'18279||2021-05-25 23:32:40,009||decoy||INFO||AUTH
COOKIE="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0
In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMU1lhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVta
CbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM51UmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
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
jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMU1lhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM51UmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
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="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidH1wIjoiand0
In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5Y1dMU1lhb0FnPT0ifQ.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=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidH1wIjoiand0I
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'
```

Osservare l'oggetto fvCtx impostato su status="Deleted".

Riferimenti e link utili

- Programmabilità ACI
- Cisco ACI Policy Model Guide
- Supporto Cisco ACI per NGINX Rate Limit

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).