Dépannage des commandes CLI de type APIC NXOS

Table des matières

Introduction

Ce document décrit les étapes à suivre pour déboguer les commandes exécutées à partir de l'interface CLI du contrôleur APIC.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- API REST ACI
- Modèle objet ACI

Le lecteur doit avoir une connaissance préalable du fonctionnement de l', ainsi que de la façon dont le processus DME enregistre ses messages.

Ces documents expliquent plus en détail le modèle ACI APIC et le modèle d'objet :

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

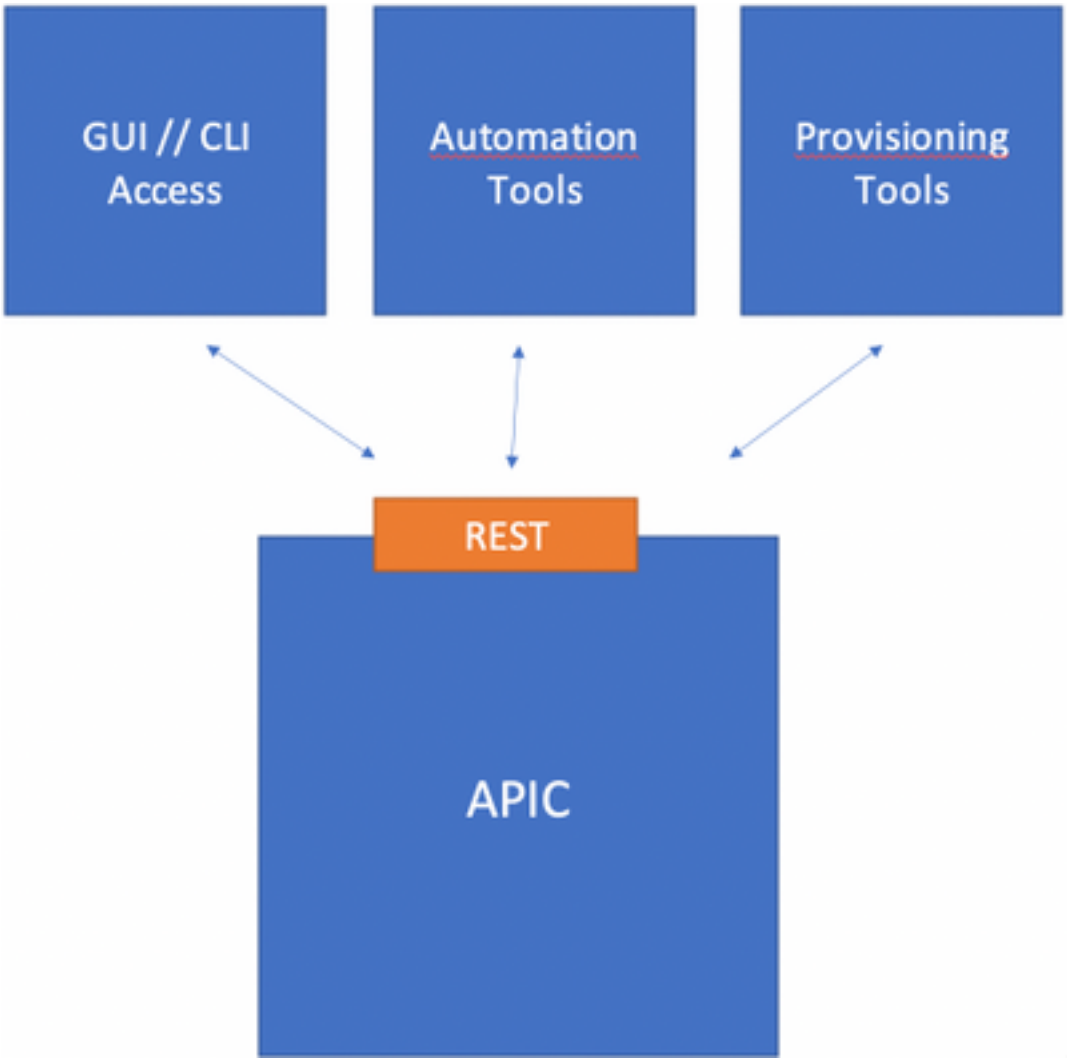
https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/policy-model-guide/b-Cisco-ACI-Policy-Model-Guide.html

Informations générales

Le contrôleur de stratégie d'application Cisco contient une API Northbound unique qui est utilisée pour la gestion des stratégies.

Chaque interaction basée sur des politiques avec un APIC se résume à une interaction HTTP/S API Request/Responded. Cela est vrai pour l'interface graphique utilisateur, les scripts python personnalisés et les commandes CLI qui existent sur les APIC et les commutateurs.

Cette image résume la manière dont les utilisateurs et les outils interagissent avec l'API APIC



Toutes les commandes **show** exécutées sur un APIC appellent l'interface de ligne de commande de **style NXOS.** Ces commandes déclenchent une série de scripts python qui construisent les requêtes d'API requises pour collecter les informations demandées. Quand une réponse est reçue, elle est analysée avec python puis remise à l'utilisateur dans un joli format.

NGINX

NGINX est un serveur Web open source. Chaque APIC exécute son propre processus NGINX qui sert l'API RESTful. Sur un APIC, NGINX inclut la journalisation via les fichiers **/var/log/dme/log/nginx.bin.log** et **/var/log/dme/log/access.log**.

Le fichier **nginx.bin.log** affiche les détails de toutes les requêtes API et interactions DME.

Le fichier **access.log** consigne chaque requête API traitée par NGINX.

Comme toutes les requêtes API sont des requêtes HTTP, les codes de réponse HTTP standard peuvent être référencés pour la gestion des appels de l'API NGINX :

- **Le** code **200** est OK
- **Les** codes **4XX** sont des erreurs client
- **Les** codes **5XX** sont des erreurs de serveur, qui sont le contrôleur APIC dans ce cas

Le service Leurre

Le service Leurre sert un appel d'API spécial via un module python hébergé hors de NGINX.

Ce service :

1. Accepte la commande CLI demandée
2. Identifie les scripts python requis pour générer les appels de l'API REST
3. Envoie les appels API à l'API RESTful
4. Accepte la réponse
5. Formate la réponse
6. Envoie la réponse formatée à l'utilisateur.

Ce service inclut les fichiers journaux suivants :

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

Le fichier leurre.log enregistre les commandes CLI qui ont été exécutées.

Le fichier **access.log** de nginx utilise le format « **POST /decoy/exec/cmd.cli HTTP/1.1** » avec le code HTTP associé à la requête. . Le fichier consigne les appels de l'API REST à partir de la commande.

> **Note**: Les journaux nginx et leurres mentionnés sont collectés dans l'APIC 3of3 Techsupport.

Dépannage de l'interface de ligne de commande NXOS

Suivre la commande CLI via les journaux

Exécuter la commande CLI

Dans cet exemple, la commande « show controller » est émise via l'interface de ligne de commande 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
```

La commande retourne un résultat plutôt formaté.

Vérifier l'exécution CMD dans le fichier leurre.log

Le fichier **leurre.log** peut être vérifié pour trouver le CMD "**show controller**" qui a été appelé :

```
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}|...
```

L'interface de ligne de commande envoie ce message sous forme de dictionnaire. Nous pouvons le voir à partir de la ligne mise en surbrillance. La version formatée ressemblerait à ceci :

```
{
   "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 ligne avec '_%CMD_TERM%_' affiche la commande exécutée, ainsi que le module utilisé, dans cet exemple :

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

Le module CLI doit ensuite le traduire en appels REST API.

Vérifiez access.log pour les appels API

Le fichier **access.log** affiche les appels d'API reçus après que le leurre a traité le 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..."
```

Vérifiez nginx.bin.log

Le fichier *nginx.bin.log* inclut des détails supplémentaires sur toutes les requêtes API traitées et doit inclure des informations sur la charge utile reçue de divers DME avant d'être renvoyée au demandeur.

Dans le même fichier, après l'appel de **decoy/exec/cmd.cli**, il y a plusieurs appels d'API enregistrés :

```
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
...
```

Réexécuter les appels d'API individuels

Le fichier access.logs contient une liste des appels API envoyés pour traitement.

L'objectif de cette étape est de réexécuter chaque appel d'API pour déterminer :

1. Y a-t-il un problème avec l'appel API lui-même ?
2. Y a-t-il une différence entre un appel API réussi et un appel en échec ?

Par exemple, l'un des appels d'API générés par la commande « show controllers » est :

```
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 icurl

Cette demande d'API peut être réexécutée avec l'URL d'utilisation sur un APIC :

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



        '
```

> **Note**: Le port 7777 est spécifiquement utilisé pour permettre au contrôleur APIC de s'interroger lui-même.

Avec l'appel spécifique comme exemple :

```
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 quantité et la complexité des appels API requis pour chaque commande CLI NXOS peuvent varier considérablement. s. Dans tous les cas, les requêtes peuvent être relues via icurl pour la validation des demandes et des réponses.

Via moquery

Àpartir des appels d'API précédents affichés, le module analyse le résultat de la commande « show controller » avec les informations des MO demandés.

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

Réexécutez la commande CLI via Python

Comme mentionné, toutes les commandes « show » sont en fait un appel d'API REST à une série de scripts python.

De ce fait, un utilisateur peut techniquement appeler manuellement le script python qui exécute la cmd. L'objectif ici est de voir si Python donne plus d'informations sur la raison pour laquelle un CMD spécifique a des problèmes.

Pour toute commande "**show**" qui a un problème, appelez la commande via Python pour comparer un APIC réussi et un APIC échoué :

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

Exemples d'exécution :

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

Exemple où le script Python direct donne des informations de débogage supplémentaires sur un échec :

```
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
```

Modification d'objet trace via CLI APIC

Création d'objets

L'APIC utilise **configure terminal** pour modifier le MIT d'une manière similaire à NXOS.

Les résultats de ces opérations peuvent être vus dans le fichier **leurre.log**.

Exemple de création VRF :

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

Résultat de **leurre.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=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'
'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="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgycDE1NmdjYXIxMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVyI6MX1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZXJuYW1lIjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxMjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMUllhb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaCbamxFaB1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFxhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWwO64LH7rMEpzyCTapJBXdkzUhJ-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'
```

Observez les objets fvTenant et fvCtx définis avec status="créé, modifié".Suppression d'objetsNous pouvons examiner les opérations REST impliquées quand un MO est supprimé.Exemple de suppression 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)#
```

Résultat de leurre.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
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'
'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
ZXJuYW11IjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMU1lhb0FnPT0ifQ.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
XJuYW11IjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZmxhZ3MiOjQsImlhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIxM
jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMU1lhb0FnPT0ifQ.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
jgyMjY2LCJzZXNzaW9uaWQiOiJUWW9pZGgyMFRIeUc5YldMU1lhb0FnPT0ifQ.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'
```

Observez l'ensemble d'objets fvCtx avec status="deleted".

Références et liens utiles

- Programmabilité ACI
- Guide du modèle de politique Cisco ACI
- Assistance Cisco ACI pour NGINX Rate Limit

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.