

Fehlerbehebung bei Befehlen der APIC NXOS-CLI

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Hintergrundinformationen](#)

[NGINX](#)

[Der Köder-Dienst](#)

[Fehlerbehebung für die Kommandozeile im NXOS-Stil](#)

[Verfolgen des CLI-Befehls über Protokolle](#)

[Ausführen des CLI-Befehls](#)

[Überprüfen Sie decoy.log auf CMD-Ausführung](#)

[API-Aufrufe auf access.log überprüfen](#)

[Überprüfen Sie nginx.bin.log](#)

[Führen Sie die einzelnen API-Aufrufe erneut aus.](#)

[Via icurl](#)

[via moquery](#)

[Führen Sie den CLI-Befehl erneut über Python aus.](#)

[Änderung von Trace-Objekten über APIC CLI](#)

[Objekterstellung](#)

[Objektlöschung](#)

[Referenzen & Nützliche Links](#)

Einleitung

In diesem Dokument werden die Schritte zum Debuggen von Befehlen beschrieben, die über die APIC-CLI ausgeführt werden.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- REST-API der ACI
- ACI-Objektmodell

Der Leser muss über Vorkenntnisse in der Funktionsweise, auch in der Protokollierung der Meldungen durch den DME-Prozess verfügen.

In diesen Dokumenten werden der ACI-APIC und das Objektmodell genauer erläutert:

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

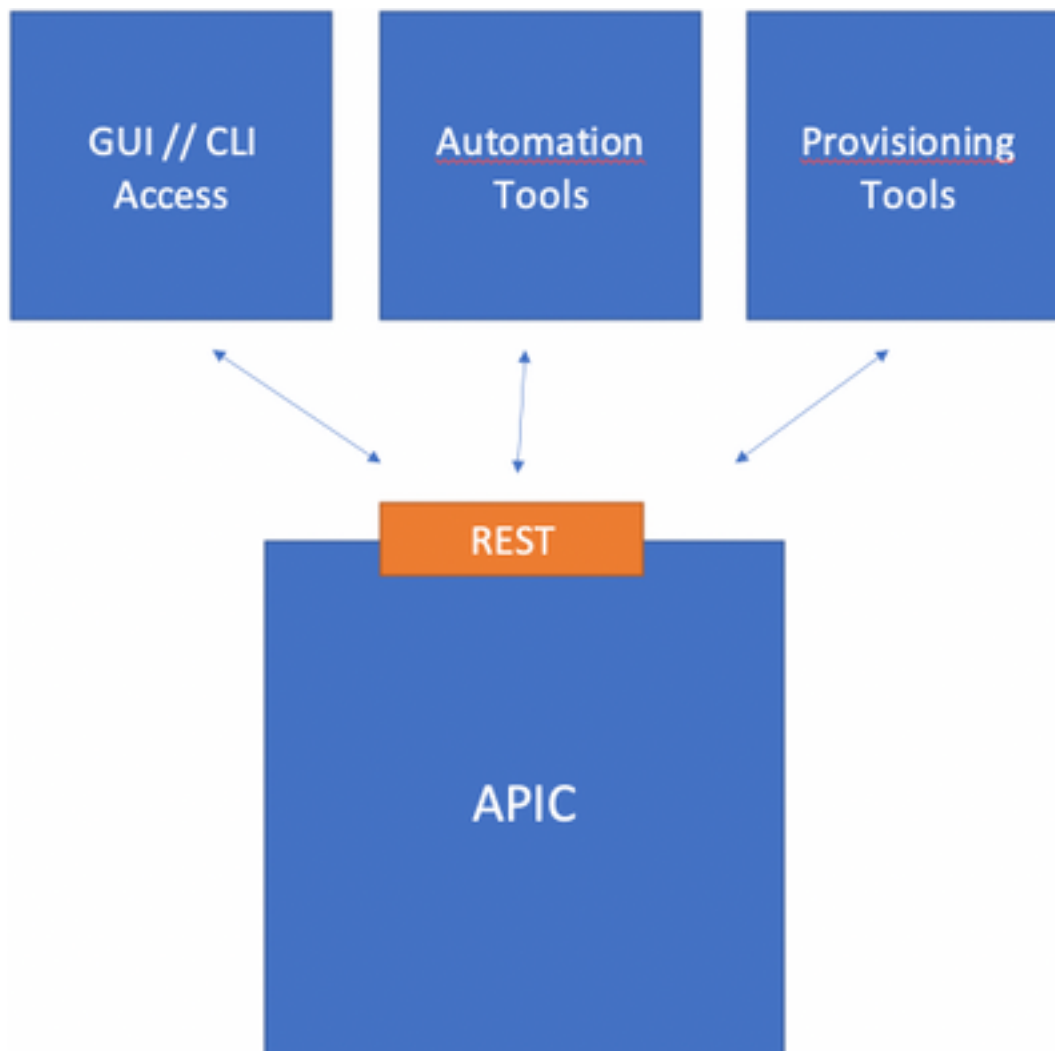
<https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/policy-model-guide/b-Cisco-ACI-Policy-Model-Guide.html>

Hintergrundinformationen

Der Cisco Application Policy Controller enthält eine einzelne Northbound-API, die für das Richtlinienmanagement verwendet wird.

Jede richtlinienbasierte Interaktion mit einem APIC läuft auf eine HTTP/S API Request/Response-Interaktion hinaus. Dies gilt für die grafische Benutzeroberfläche (GUI), benutzerdefinierte Python-Skripts und die CLI-Befehle, die auf den APICs und Switches vorhanden sind.

Dieses Bild fasst die Interaktion von Benutzern und Tools mit der APIC-API zusammen.



Alle **show**-Befehle werden auf einem APIC ausgeführt und rufen die **Kommandozeile im NXOS-Stil auf**. Diese Befehle lösen eine Reihe von Python-Skripten aus, die die API-Anforderungen erstellen, die zum Erfassen der angeforderten Informationen erforderlich sind. Wenn eine Antwort eingeht, wird sie mit Python analysiert und dann in einem hübschen Format an den Benutzer gesendet.

NGINX

NGINX ist ein Open-Source-Webserver. Jeder APIC führt seinen eigenen NGINX-Prozess aus, der die RESTful-API bereitstellt. Auf einem APIC umfasst NGINX die Protokollierung über die Dateien `/var/log/dme/log/nginx.bin.log` und `/var/log/dme/log/access.log`.

In der Datei `nginx.bin.log` werden die Details aller API-Anforderungen und DME-Interaktionen angezeigt.

Die Datei `access.log` protokolliert alle von NGINX verarbeiteten API-Anfragen.

Da es sich bei allen API-Anfragen um HTTP-Anfragen handelt, können Standard-HTTP-Antwortcodes für die NGINX API-Anrufbearbeitung herangezogen werden:

- **200** Code ist OK
- **4XX**-Codes sind Client-Fehler
- **5XX**-Codes sind Serverfehler, in diesem Fall der APIC

Der Köder-Dienst

Der Decoy-Service bedient einen speziellen API-Anruf über ein Python-Modul, das von NGINX gehostet wird.

Dieser Service:

1. Akzeptiert den angeforderten CLI-Befehl
2. Identifiziert die Python-Skripte, die zum Ausbau der REST-API-Aufrufe erforderlich sind
3. Sendet die API-Aufrufe an die RESTful-API
4. Akzeptiert die Antwort
5. Formatiert die Antwort
6. Gibt die formatierte Antwort an den Benutzer weiter.

Dieser Dienst enthält die folgenden Protokolldateien:

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

Die Datei `decoy.log` registriert die ausgeführten CLI-Befehle.

Die Datei `access.log` von nginx verwendet das `"POST /decoy/exec/cmd.cli HTTP/1.1"`-Format zusammen mit dem HTTP-Code, der der Anforderung zugeordnet ist. Die Datei protokolliert die REST-API-Aufrufe über den Befehl.

Anmerkung: Die genannten Protokolle "nginx" und "decoy" werden im Rahmen des technischen Supports für APIC 3of3 erfasst.

Fehlerbehebung für die Kommandozeile im NXOS-Stil

Verfolgen des CLI-Befehls über Protokolle

Ausführen des CLI-Befehls

In diesem Beispiel wird der "show controller" über die APIC-CLI bereitgestellt:

```
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-	XXXXXXXXXXXX	fully-fit		
2	1	10.0.0.2	0.0.0.0	fc00::1	192.168.1.1	
		4.2(6h)	crva-	XXXXXXXXXXXX	fully-fit	

Flags - c:Commissioned | r:Registered | v:Valid Certificate | a:Approved | f/s:Failover fail/success
(*)Current (~)Standby (+)AS

Der Befehl gibt eine ziemlich formatierte Ausgabe zurück.

Überprüfen Sie decoy.log auf CMD-Ausführung

Die Datei `decoy.log` kann überprüft werden, um den CMD "show controller" zu finden, der aufgerufen wurde:

```
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: [{"exec"}] ||...
...| Command: [u'show', u'controller'] ||...
...| CommandCompleter: add exec ||...
...| CommandCompleter: add show ||...
...| CommandCompleter: add controller ||...
...| last tokens: ['show', 'controller'] ||...
...| modeCmd: Mode: exec, fulltree: False terminal context is : {'mode-module': 'yaci._cfgroot', 'module': 'show._controllers', 'inherited': False} ||...
...| ('_%CMD_TERM%', {'mode-module': 'yaci._cfgroot', 'module': 'show._controllers', 'inherited': False}, {'prompt': '# ', 'mode': [{"exec"}]}, None) ||...
...| terminal command module: {"mode-module": "yaci._cfgroot", "module": "show._controllers", "inherited": false} ||...
```

Die Kommandozeile sendet dies als Wörterbuch, wie wir an der hervorgehobenen Zeile sehen können. Die formatierte Version sieht wie folgt aus:

```
{
  "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"
]
}

```

Die Zeile mit `"_%CMD_TERM%_"` zeigt den ausgeführten Befehl zusammen mit dem in diesem Beispiel verwendeten Modul:

```

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

```

Das CLI-Modul muss es dann in REST-API-Aufrufe übersetzen.

API-Aufrufe auf access.log überprüfen

In der Datei `access.log` werden die API-Aufrufe angezeigt, die nach der Verarbeitung des CMD durch Decoy empfangen wurden:

```

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..."

```

Überprüfen Sie nginx.bin.log

Die Datei `"nginx.bin.log"` enthält weitere Details zu allen verarbeiteten API-Anforderungen und muss Informationen über die Nutzlast enthalten, die von verschiedenen DMEs empfangen wurde, bevor sie an den Anforderer zurückgesendet wird.

In derselben Datei werden nach dem Aufruf von `decoy/exec/cmd.cli` mehrere API-Aufrufe protokolliert:

```

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
...

```

Führen Sie die einzelnen API-Aufrufe erneut aus.

Die Datei access.logs enthält eine Liste der API-Aufrufe, die zur Verarbeitung gesendet wurden.

In diesem Schritt soll jeder API-Aufruf erneut ausgeführt werden, um Folgendes zu ermitteln:

1. Besteht ein Problem mit dem API-Aufruf selbst?
2. Besteht ein Unterschied zwischen einem erfolgreichen und einem fehlgeschlagenen API-Aufruf?

Einer der API-Aufrufe, die mit dem Befehl "show controllers" generiert werden, lautet beispielsweise:

```

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

Diese API-Anforderung kann mit der icurl auf einem APIC erneut ausgeführt werden:

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

,

Anmerkung: Port 7777 wird speziell verwendet, um dem APIC die Abfrage zu ermöglichen.

Beispiel für einen bestimmten Anruf:

```

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>

```

Die Anzahl und Komplexität der API-Aufrufe, die für die einzelnen NX-OS-CLI-Befehle erforderlich sind, kann stark variieren. s. In allen Fällen können die Abfragen zur Anforderungs- und Antwortvalidierung über icurl wiedergegeben werden.

via moquery

Aus den vorherigen API-Aufrufen analysiert das Modul die Ausgabe für den Befehl "show controller" mit den Informationen aus den angeforderten MOs.

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

Führen Sie den CLI-Befehl erneut über Python aus.

Wie erwähnt, sind alle "show"-Befehle tatsächlich ein REST-API-Aufruf für eine Reihe von Python-Skripten.

Dadurch kann ein Benutzer das Python-Skript, das den Befehl cmd ausführt, technisch manuell aufrufen. Das Ziel ist hier, zu sehen, ob Python weitere Informationen gibt, warum ein bestimmtes CMD Probleme hat.

Rufen Sie für jeden "show"-Befehl, bei dem ein Problem vorliegt, den Befehl über Python auf, um einen erfolgreichen mit einem ausgefallenen APIC zu vergleichen:

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

Beispielausführungen:

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

Beispiel, bei dem das direkte Python-Skript zusätzliche Debuginformationen zu einem Fehler liefert:

```
a-apicl# ${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(60) aliv XXXXXXXXXXXX leaf101
...
205 ...
```

```
206 2 10.0.156.65 192.168.2.236 :: 192.168.1.106 :: n9000-14.2(61) aliv XXXXXXXXXXXX leaf206
...
```

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

Änderung von Trace-Objekten über APIC CLI

Objekterstellung

Der APIC verwendet das **konfigurierte Terminal**, um das MIT NX-OS-artig zu modifizieren.

Die Ergebnisse dieser Vorgänge sind in der Datei **decoy.log** zu sehen.

Beispiel zur VRF-Erstellung:

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

Ausgabe von **decoy.log**:

```
'18279||2021-05-25 22:52:11,877||decoy||INFO||AUTH
COOKIE="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyYXl1NmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0
In0.eyJyYmFjIjpbeyJkb21hYW4iOiJhbGwiLCJyb2xlcl1iOiJEsInJvbnVzVjYyI6ImN1dCJpc3MiOiJBQ0kgQVBJQyIsInVz
ZXJyYW1lIjoiYWRtaW4iLCJlc2VyaWQiOiJlMzc0LCJlc2VyZmxhZ3MiOiJQsImldhdCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIx
MjgyMjY2LCJzZXNzaW9uaWQiOiJUUWw9pZGgyMFRlUc5YldMUl1hb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVta
CbamxFaBlLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd28leaSQPPWit-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFhxZgYI2Lx8gg0P6sLoUydcShKKcUNGRmGWw
O64LH7rMEpyCTapJBXdkzUhJ-zm98fmOyloGGTesBteSWP_ksH14Xq411klebJ83sV4tL6-
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=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyYmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbkVzVzV6IjE6Mk1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMzc0LCJ1c2VzZm9hZ3M3MiOiJ0fQ.ksnCeOxnRnQeuNaQnmpauUG_eja70nVtaC
bamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VrauKwW8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFzhZgYI2Lx8gg0P6sLoUydcSHKKcUNGmGWw
O64LH7rMEpyzCTapJBXdKzUhJ-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="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyYmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbkVzVzV6IjE6Mk1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
ZXJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMzc0LCJ1c2VzZm9hZ3M3MiOiJ0fQ.ksnCeOxnRnQeuNaQnmpauUG_eja70nVta
CbamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VrauKwW8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFzhZgYI2Lx8gg0P6sLoUydcSHKKcUNGmGWw
O64LH7rMEpyzCTapJBXdKzUhJ-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=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyYzI1MDI1YmR3MG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiaW00
n0.eyJyYmFJb2JpbGciOiJSUzI1NiIsImtpZCI6ImgyYzI1MDI1YmR3MG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiaW00
XJYyYmFJb2JpbGciOiJSUzI1NiIsImtpZCI6ImgyYzI1MDI1YmR3MG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiaW00
jgyYmFJb2JpbGciOiJSUzI1NiIsImtpZCI6ImgyYzI1MDI1YmR3MG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiaW00
b2JpbGciOiJSUzI1NiIsImtpZCI6ImgyYzI1MDI1YmR3MG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiaW00
ieCjYxIm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHjvXpPctDsBYi9YM5lUmFzhZgYI2Lx8gg0P6sLoUydcShKKcUNGmGwW
O64LH7rMEpzyCTapJBXdKzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUwWZFzb5tY4JUJxrSnahKfwyidNXt5m8LCIc8pt-xbBtVihAFkAYBoXKI-OYBrwg',

```



```
'Cookie': 'APIC-
cookie=eyJhbGciOiJIUzU1NiIsImtpZCI6ImgyYDE1NmMjYXlzMGM5Oemx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwlcjYybnxlc1IiOiJEsInJvbGVzV3R1I6Mk1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYVllIjoiYWRtaW4iLCJ1c2VyaWQlOiElMz00LCJ1c2VyZmRhZ3MlOiJQsImldCI6MTFYMTI4MTY2NiwiZXhwIjozNjIxM
jgyMjY2LCJzZXNzaW9uaWQiOiJ0UW9pZGgyMFRlUc5YldMUI1hb0FnPT0ifQ.ksnCeOxnRNQeuNaQnmpauUG_eja70nVtaC
bamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VrauKwW8Fztd281easQPPWiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUJvXpPctDsBYi9YM5lUmFkhZgYI2Lx8gg0P6sLoUydcShKKcUNGRmGWw
064LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTebTeSWP_ksH14Xq411klebJ83sV4tL6-
FJLhcPNlKwqYJ87fqUWwZFzb5tY4JUJxrSnahKfwyidNXt5m8LCic8pt-xbBtVihAFkAYBoXXI-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="eyJhbGciOiJIUzU1NiIsImtpZCI6ImgyYDE1NmMjYXlzMGM5Oemx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwlcjYybnxlc1IiOiJEsInJvbGVzV3R1I6Mk1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYVllIjoiYWRtaW4iLCJ1c2VyaWQlOiElMz00LCJ1c2VyZmRhZ3MlOiJQsImldCI6MTFYMTI4MTY2NiwiZXhwIjozNjIxM
jgyMjY2LCJzZXNzaW9uaWQiOiJ0UW9pZGgyMFRlUc5YldMUI1hb0FnPT0ifQ.ksnCeOxnRNQeuNaQnmpauUG_eja70nVta
CbamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VrauKwW8Fztd281easQPPWiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUJvXpPctDsBYi9YM5lUmFkhZgYI2Lx8gg0P6sLoUydcShKKcUNGRmGWw
064LH7rMEpzyCTapJBXdkzUhJ-zm98fmOy1oGGTebTeSWP_ksH14Xq411klebJ83sV4tL6-
FJLhcPNlKwqYJ87fqUWwZFzb5tY4JUJxrSnahKfwyidNXt5m8LCic8pt-xbBtVihAFkAYBoXXI-
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=eyJhbGciOiJIUzU1NiIsImtpZCI6ImgyYDE1NmMjYXlzMGM5Oemx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiand0I
```

```

n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbGVzVvyI6MK1dLCJpc3MiOiJBQ0kgQVBjQyIsInVzZ
XJuYW11IjoiYWRtaW4iLCJ1c2VyaWQiOiE1Mzc0LCJ1c2VyZm9zZ3MiOiJQsImldhdCI6MTYyMTI4MTY2NiwiZm9kIjojNjIiXm
jgyMjY2LCJzZXNzaW9uaWQiOiJUUW9pZGgyMFRlUc5YldMU11hb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFzhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
064LH7rMEpzyCTapJBXdKzUhJ-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=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyZDE1NmQjYXkiMG50emx3YmR3MDFjNnV5dGEycXQzIiwidHlwIjoiaW90Ii
n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbGVzVvyI6MK1dLCJpc3MiOiJBQ0kgQVBjQyIsInVzZ
XJuYW11IjoiYWRtaW4iLCJ1c2VyaWQiOiE1Mzc0LCJ1c2VyZm9zZ3MiOiJQsImldhdCI6MTYyMTI4MTY2NiwiZm9kIjojNjIiXm
jgyMjY2LCJzZXNzaW9uaWQiOiJUUW9pZGgyMFRlUc5YldMU11hb0FnPT0ifQ.ksnCeOxnrNQeuNaQnmpauUG_eja70nVtaC
bamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHUjvXpPctDsBYi9YM5lUmFzhZgYI2Lx8gg0P6sLoUydcShKKcUNgRmGWw
064LH7rMEpzyCTapJBXdKzUhJ-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'

```

Beobachten Sie den fvCtx-Objektsatz mit status="deleted". Referenzen & Nützliche Links

- [ACI-Programmierbarkeit](#)
- [Leitfaden für Cisco ACI-Richtlinien](#)
- [Cisco ACI-Unterstützung für NGINX-Durchsatzgrenze](#)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.