

Opdrachten voor CLI in APIC NXOS-stijl oplossen

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Achtergrondinformatie](#)

[NGINX](#)

[De lokalisatieservice](#)

[Probleemoplossing voor de NXOS-stijl CLI](#)

[De CLI-opdracht overtrekken met logbestanden](#)

[De CLI-opdracht uitvoeren](#)

[Controleer decoy.log op CMD Execution](#)

[Controleer access.log op API-oproepen](#)

[Controleer nginx.bin.log](#)

[De individuele API-oproepen opnieuw uitvoeren](#)

[Via icurl](#)

[Via moquery](#)

[Herstart de CLI-opdracht via Python](#)

[Wijziging van overtrekobjecten via APIC CLI](#)

[Creatie van objecten](#)

[Objectverwijdering](#)

[Referenties en nuttige links](#)

Inleiding

Dit document beschrijft stappen om opdrachten te debuggen die vanaf de APIC-CLI worden uitgevoerd.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- ACI REST API
- ACI-objectmodel

De lezer moet een eerdere kennis hebben over hoe het werk, ook hoe het DME-proces hun berichten registreert.

In deze documenten wordt meer gedetailleerd uitleg gegeven over het ACI APIC- en Object-model:

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

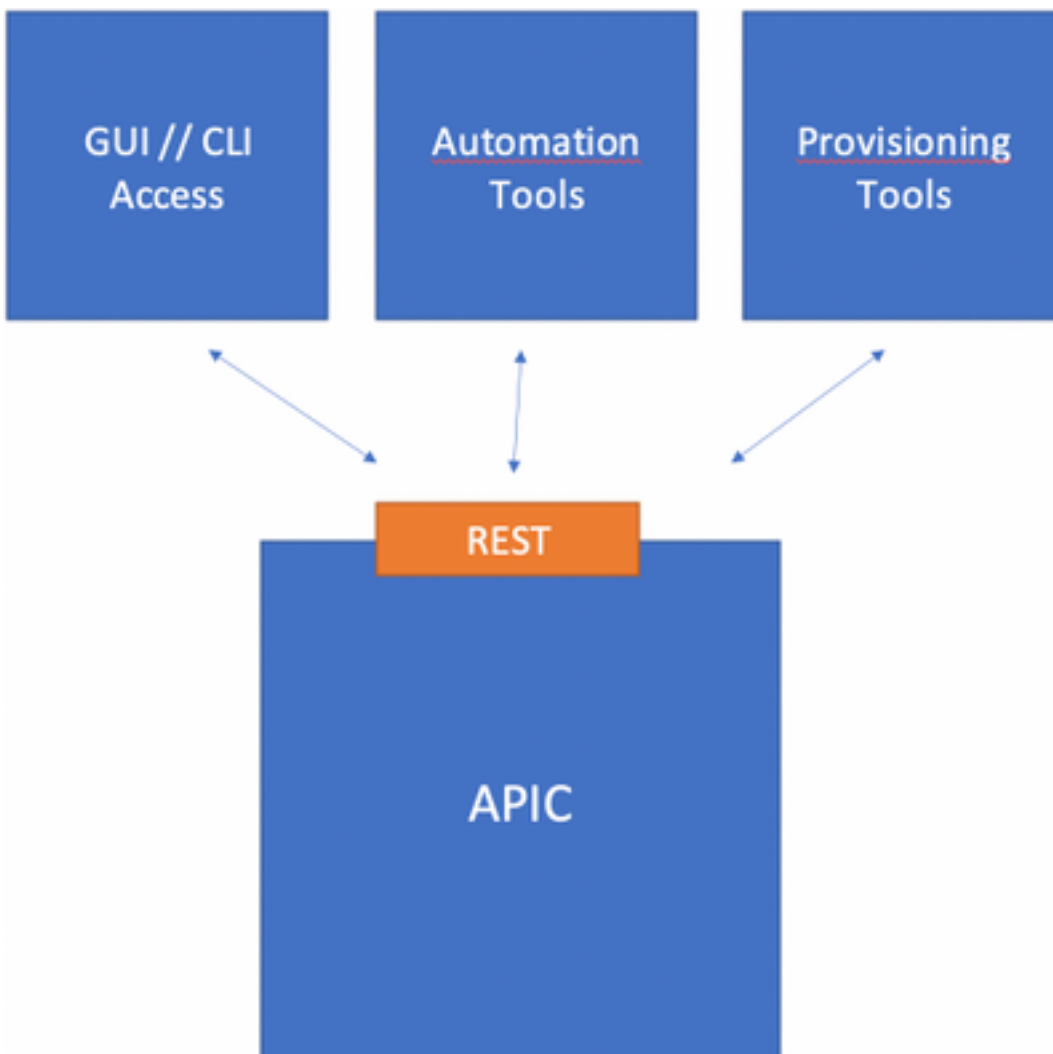
<https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/policy-model-guide/b-Cisco-ACI-Policy-Model-Guide.html>

Achtergrondinformatie

De Cisco Application Policy Controller bevat één noordelijke API die wordt gebruikt voor beleidsbeheer.

Elke op beleid gebaseerde interactie met een APIC komt neer op een HTTP/S API Aanvraag/Gereageerd interactie. Dit geldt voor de GUI, aangepaste python-scripts en de CLI-opdrachten die op zowel de APIC's als de Switches bestaan.

Deze afbeelding vat de manier samen waarop gebruikers en tools communiceren met de APIC's API



Alle **show** commando's uitgevoerd op een APIC halen de **NXOS Style CLI**. Deze opdrachten starten een serie python-scripts die de API-aanvragen uitbouwen die nodig zijn om de gevraagde informatie te verzamelen. Wanneer een antwoord wordt ontvangen, wordt het geparst met python dan geleverd aan de gebruiker in een vrij formaat.

NGINX

NGINX is een open source webserver. Elke APIC voert zijn eigen NGINX-proces uit dat de RESTful API dient. Op een APIC, NGINX omvat het registreren via zowel de `/var/log/dme/log/nginx.bin.log` als `/var/log/dme/log/access.log` bestanden.

Het bestand `nginx.bin.log` toont de details van alle API-aanvragen en DME-interacties.

Het `access.log`-bestand logt elke API-aanvraag af die door NGINX wordt verwerkt.

Aangezien alle API-verzoeken betrekking hebben op HTTP-verzoeken, kunnen standaard HTTP-responscodes worden gebruikt voor NGINX API Call handling:

- **200** code is OK
- **4XX** codes zijn clientfouten
- **5X**-codes zijn serverfouten, in dit geval het APIC

De lokalisatieservice

De Decoy-service verzorgt een speciale API-oproep via een python-module die wordt gehost vanaf NGINX.

Deze service:

1. Accepteert de gevraagde CLI-opdracht
2. Identificeert de python scripts die nodig zijn om de REST API gesprekken uit te bouwen
3. De API-oproepen naar de RESTful API verzenden
4. aanvaardt het antwoord
5. Formateert de respons
6. Dient voor het geformatteerde antwoord op de gebruiker.

Deze service omvat deze logbestanden:

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

Het bestand `decoy.log` registreert de CLI-opdrachten die zijn uitgevoerd.

Het `access.log` bestand van nginx gebruikt het "`POST /decoy/exec/cmd.cli HTTP/1.1`" formaat samen met de HTTP code geassocieerd met het verzoek. Het bestand registreert de REST API-oproepen vanuit de opdracht.

Opmerking: De genoemde nginx- en decoy-logs worden verzameld binnen de APIC 3of3 Techsupport.

Probleemoplossing voor de NXOS-stijl CLI

De CLI-opdracht overtrekken met logbestanden

De CLI-opdracht uitvoeren

In dit voorbeeld wordt de "show controller" uitgegeven via de APIC CLI:

```
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

Het commando geeft een vrij geformatteerde output terug.

Controleer decoy.log op CMD Execution

Het bestand **decoy.log** kan worden gecontroleerd om de CMD "show controller" te vinden die is aangeroepen:

```
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 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': [{"exec"}]}, None)||...
...|terminal command module: {"mode-module": "yaci._cfgroot", "module": "show._controllers", "inherited": false}||...
```

De CLI stuurt dit als een woordenboek, we kunnen het zien vanaf de gemarkeerde regel. De geformatteerde versie zou er als volgt uitzien:

```
{
  "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"
  ]
}

```

De regel met `"_%CMD_TERM%"` toont de uitgevoerde opdracht, samen met de gebruikte module, in dit voorbeeld:

```

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

```

De CLI module moet het vervolgens vertalen naar REST API-oproepen.

Controleer access.log op API-oproepen

Het bestand `access.log` toont de API-oproepen die worden ontvangen nadat de decoy de CMD verwerkt:

```

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..."

```

Controleer nginx.bin.log

Het bestand `nginx.bin.log` bevat aanvullende informatie over alle verwerkte API-aanvragen en moet informatie bevatten over de payload die van verschillende DME's is ontvangen voordat deze naar de aanvrager wordt teruggestuurd.

In hetzelfde bestand, nadat `decoy/exec/cmd.cli` is aangeroepen, zijn er verschillende API-oproepen gelogd:

```

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
...
```

De individuele API-oproepen opnieuw uitvoeren

Access.logs bevatten een lijst van precies welke API-oproepen voor verwerking werden verzonden.

Het doel van deze stap is elke API-aanroep opnieuw uit te voeren om te bepalen:

1. Is er een probleem met de API-aanroep zelf?
2. Is er iets anders aan een succesvolle API call versus een mislukte?

Een van de API-oproepen die worden gegenereerd vanuit de opdracht "show controllers" is bijvoorbeeld:

```
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

Deze API-aanvraag kan opnieuw worden uitgevoerd met het gebruiksprofiel op een APIC:

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

Opmerking: Port 7777 wordt specifiek gebruikt om de APIC in staat te stellen zichzelf te bevragen.

Met de specifieke oproep als voorbeeld:

```
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>
```

De hoeveelheid en complexiteit van API-aanroepen die vereist zijn voor elke NXOS CLI-opdracht kan enorm verschillen. s. In alle gevallen kunnen de queries opnieuw worden afgespeeld via icurl voor verzoek- en antwoordvalidatie.

Via moquery

Van de vorige getoonde API-oproepen parseert de module de uitvoer voor de opdracht "show controller" met de informatie van de gevraagde MO's.

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

Herstart de CLI-opdracht via Python

Zoals vermeld, zijn alle "show" commando's eigenlijk een REST API call naar een reeks python scripts.

Hierdoor kan een gebruiker technisch handmatig het python script aanroepen dat de cmd draait. Het doel is om te zien of Python meer informatie geeft over waarom een specifieke CMD problemen heeft.

Voor elke "show" opdracht die een probleem heeft, haal de opdracht via Python aan om een succesvolle vs een mislukte APIC te vergelijken:

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

Voorbeeld draait:

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

Voorbeeld waar het directe Python-script aanvullende debug-informatie geeft over een fout:

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

```
Process Process-2:
Traceback (most recent call last):
```



```
'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.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbGVzVyI6MX1dLkZjc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMzc0LCJ1c2VyZm9zZ3MiOiJQsImldCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIyMjYy
jgyMjY2LkZzZXNzaW9uaWQiOiJlMzc0LCJ1c2VyZm9zZ3MiOiJQsImldCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIyMjYy
bamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHJuvXpPctDsBYi9YM5lUmFxxhZgYI2Lx8gg0P6sLoUydcSHKkCUNGmGWw
064LH7rMEpyCTapJBXdkzUhJ-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.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbGVzVyI6MX1dLkZjc3MiOiJBQ0kgQVBJQyIsInVzZ
ZXJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMzc0LCJ1c2VyZm9zZ3MiOiJQsImldCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIyMjYy
MjYyMjY2LkZzZXNzaW9uaWQiOiJlMzc0LCJ1c2VyZm9zZ3MiOiJQsImldCI6MTYyMTI4MTY2NiwiZXhwIjoxNjIyMjYy
CbamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPWiT-
ieCjWxim8Sw4spYS8XBrBBx62tot201TIEJ8mUFHJuvXpPctDsBYi9YM5lUmFxxhZgYI2Lx8gg0P6sLoUydcSHKkCUNGmGWw
064LH7rMEpyCTapJBXdkzUhJ-zm98fmOy1oGGTesBteSWP_ksH14Xq411k1ebJ83sV4tL6-
```


'Cookie': 'APIC-
cookie=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyZDE1NmMjYXl1MG50emx3YmR3MDFjNmV5dGEyYXQzIiwidHlwIjoiand0In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbkVzVyI6Mk1dLCJpc3MiOiJBQ0kgQVBjQyIsInVzZXJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMz00LCJ1c2VyZmxhZ3MiOiJQsImlhdCI6MTFYMTI4MTY2NiwiZlZlIjoiXzNmJyY2LCJzZXNzaW9uaWQiOiJUW9pZGgyMFRlUc5YldMUI1hb0FnPT0ifQ.ksnCeOxnrNQeUNaQnmpauUG_eja70nVtaCbamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPwiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHujvXpPctDsBYi9Ym5lUmFzhZgYI2Lx8gg0P6sLoUydcSHKkcUNGRmGWw064LH7rMEpzyCTapJBXdkzUhJ-zm98fmOyloGGTesBteSWP_ksH14Xq411klebJ83sV4tL6-
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="eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyZDE1NmMjYXl1MG50emx3YmR3MDFjNmV5dGEyYXQzIiwidHlwIjoiand0In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbkVzVyI6Mk1dLCJpc3MiOiJBQ0kgQVBjQyIsInVzZXJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMz00LCJ1c2VyZmxhZ3MiOiJQsImlhdCI6MTFYMTI4MTY2NiwiZlZlIjoiXzNmJyY2LCJzZXNzaW9uaWQiOiJUW9pZGgyMFRlUc5YldMUI1hb0FnPT0ifQ.ksnCeOxnrNQeUNaQnmpauUG_eja70nVtaCbamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPwiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHujvXpPctDsBYi9Ym5lUmFzhZgYI2Lx8gg0P6sLoUydcSHKkcUNGRmGWw064LH7rMEpzyCTapJBXdkzUhJ-zm98fmOyloGGTesBteSWP_ksH14Xq411klebJ83sV4tL6-
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=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyZDE1NmMjYXl1MG50emx3YmR3MDFjNmV5dGEyYXQzIiwidHlwIjoiand0In0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOiJEsInJvbkVzVyI6Mk1dLCJpc3MiOiJBQ0kgQVBjQyIsInVzZXJuYWI1IjoiYWRtaW4iLCJ1c2VyaWQiOiJlMz00LCJ1c2VyZmxhZ3MiOiJQsImlhdCI6MTFYMTI4MTY2NiwiZlZlIjoiXzNmJyY2LCJzZXNzaW9uaWQiOiJUW9pZGgyMFRlUc5YldMUI1hb0FnPT0ifQ.ksnCeOxnrNQeUNaQnmpauUG_eja70nVtaCbamxFab1LLkMIqzJ_wk_GMN1h4eM1WLS41VraukWw8Fztd281eaSQPPwiT-
ieCjWxlm8Sw4spYS8XBrBBx62tot201TIEJ8mUFHujvXpPctDsBYi9Ym5lUmFzhZgYI2Lx8gg0P6sLoUydcSHKkcUNGRmGWw064LH7rMEpzyCTapJBXdkzUhJ-zm98fmOyloGGTesBteSWP_ksH14Xq411klebJ83sV4tL6-
FJLhcPNIKwqYJ87fqUWwZFzb5tY4JUJxrSnahKfwyidNXt5m8LCic8pt-xbBtVihAFkAYBoXKI-OYBrwg',

```

n0.eyJyYmFjIjpbeyJkb21haW4iOiJhbGwiLCJyb2xlc1IiOjEsInJvbGVzVvyI6MK1dLCJpc3MiOiJBQ0kgQVBJQyIsInVzZ
XJuYW11IjoiYWRtaW4iLCJ1c2VyaWQiOjE1Mzc0LCJ1c2VyZm9zZ3MiOjQsIm1hdCI6MTYyMTI4MTY2NiwiZm9wZXhwIjoxNjI4M
jgyMjY2LCJzZXNzaW9uaWQiOiJlUW9pZGgyMFRlUc5YldMU11hb0FnPT0ifQ.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=eyJhbGciOiJSUzI1NiIsImtpZCI6ImgyYmFjYXN1c2VyaWQiOjE1Mzc0LCJ1c2VyZm9zZ3MiOjQsIm1hdCI6MTYyMTI4MTY2NiwiZm9wZXhwIjoxNjI4M
jgyMjY2LCJzZXNzaW9uaWQiOiJlUW9pZGgyMFRlUc5YldMU11hb0FnPT0ifQ.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'

```

Neem het fvCtx object waar dat is ingesteld met status="deleted". Referenties en nuttige links

- [ACI-programmeerbaarheid](#)
- [Handleiding voor Cisco ACI-beleidsmodellen](#)
- [Cisco ACI-ondersteuning voor NGINX-snelheidsbeperking](#)

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.