

TESTINGGGGGGGGGGGGGGGGGGG

Cisco Application Control Engine (ACE) Troubleshooting Guide - Troubleshooting ACE Health Monitoring

Inhalt

- 1 Übersicht über die ACE-Integritätsüberwachung
 - 1.1 Sondentypen
 - 1.2 Konfigurieren von Tests
 - 1.3 Beispiel einer Testkonfiguration
- 2 Fehlerbehebung ACE-Integritätsüberwachung
 - 2.1 Grundlegende Fehlerbehebung bei Tests
 - 2.2 Fehlerbehebung bei einem TCP-Testfehler
 - 2.3 Fehlerbehebung bei einem UDP-Testfehler
 - 2.4 Fehlerbehebung bei HTTP-Testfehlern
 - 2.5 Fehlerbehebung bei HTTPS-Testfehlern
 - 2.6 Fehlerbehebung bei SNMP-Testproblemen
 - 2.7 Verwenden des Felds "Letzter Statuscode"
 - 2.8 Testdebugs

Übersicht über die ACE-Integritätsüberwachung

In diesem Abschnitt wird die Systemüberwachung auf dem ACE beschrieben.

Der ACE verwendet Tests als OOB-Statusüberwachung (Out-of-Band), um den Status eines Servers zu verfolgen. Standardmäßig sind im ACE keine Tests konfiguriert. Der ACE verifiziert die Serverantwort auf eine Anfrage (oder wenn überhaupt keine Antwort erfolgt), um Netzwerk- oder Anwendungsprobleme zu überprüfen, die einen Client daran hindern können, einen Server zu erreichen. Auf Basis der Serverantwort kann der ACE den Server in Betrieb nehmen oder außer Betrieb setzen und zuverlässige Entscheidungen zum Lastenausgleich treffen.

Sie können die Systemüberwachung auch verwenden, um Fehler für ein Gateway oder einen Host in hochverfügbaren (redundanten) Konfigurationen zu erkennen. Weitere Informationen finden Sie im [Cisco Application Control Engine Module Administration Guide](#).

Der ACE bewertet den Zustand eines Servers, indem er die Tests wie folgt markiert:

- Bestanden - Der Server gibt eine gültige Antwort zurück.
- Fehlgeschlagen — Der Server kann keine gültige Antwort an den ACE übermitteln, und der ACE kann einen Server für eine bestimmte Anzahl von Wiederholungen nicht erreichen.

Wenn der ACE für die Integritätsüberwachung konfiguriert wird, sendet er regelmäßig aktive Tests, um den Serverstatus zu ermitteln. Der ACE unterstützt 4.096 eindeutige Diagnosekonfigurationen, darunter ICMP, TCP, HTTP und andere vordefinierte Diagnosetools. Der ACE kann jeweils nur bis zu 200 skriptgesteuerte Tests gleichzeitig ausführen. Der ACE ermöglicht auch das gleichzeitige Öffnen von 2048 Sockeln.

Sie können den gleichen Test mehreren echten Servern oder Serverfarmen zuordnen. Jedes Mal, wenn Sie dieselbe Probe erneut verwenden, zählt der ACE sie als eine weitere Probe-Instanz. Sie können maximal 16.000 Testinstanzen zuordnen.

Der ACE unterstützt außerdem eine In-Band-Integritätsüberwachungsfunktion (standardmäßig deaktiviert), die Verbindungsfehler auf den echten Servern in einer Serverfarm verfolgt:

- Bei TCP werden Resets (RSTs) vom Server oder SYN-Timeouts durchgeführt.
- Für nicht erreichbare Nachrichten von UDP, ICMP-Host, Netzwerk, Port, Protokoll und Quellroute

Wenn Sie einen Schwellenwert für die Fehleranzahl konfigurieren und die Anzahl dieser Fehler den Schwellenwert innerhalb des Intervalls zum Zurücksetzen überschreitet, markiert der ACE den Server sofort als ausgefallen, nimmt ihn außer Betrieb und entfernt ihn aus dem Lastenausgleich. Der Server wird erst dann für den Lastenausgleich in Betracht gezogen, wenn das optionale Fortsetzungsdienstintervall abläuft.

Weitere Informationen zur In-Band-Integritätsüberwachung finden Sie im [Cisco Application Control Engine Module Server Load Balancing Guide](#).

Sondentypen

- ICMP-Anfrage
 - Sendet Echoanfrage
 - Erwartet Echo-Antwort
- Generischer TCP-Prüfpunkt
 - Verbindung mit Server öffnen und Verbindung mit TCP FIN trennen (Standard)
 - TCP RST kann verwendet werden, um die TCP-Überprüfung bei Bedarf zu schließen.
- Generischer UDP-Prüfpunkt
 - Sendet ein Paket, Test wird als erfolgreich angesehen, wenn kein icmp-Fehler empfangen wurde
- HTTP-Anfrage
 - Sendet eine GET/HTTP 1.1-Anforderung

- Erwartet eine HTTP-Antwort mit dem konfigurierten Rückgabecode

- HTTPS-Anfrage

- Stellt eine SSL-Verbindung her, sendet HTTP-Anfrage und trennt sie ab
- HTTP-Anfrage/-Antwort funktioniert wie bei HTTP-Anfrage

- FTP-Datensammlung

- Ähnlich wie bei TCP-Anfrage, eine "QUIT"-Nachricht senden

- Telnet-Anfrage

- Stellt eine Verbindung her, sendet eine "QUIT"-Nachricht

- DNS-Probe

- Sendet DNS-Abfrage, verwendet die Standarddomäne " www.cisco.com
- erwartet eine IP-Antwort für die abgefragte Domäne

- SMTP-Anfrage

- Sendet eine "HELO"-Nachricht gefolgt von einer "QUIT"-Nachricht

- POP3-Sonde

- Ähnlich wie bei TCP-Anfrage, kann Anmeldeinformationen senden

- IMAP-Anfrage

- Ähnlich wie bei TCP-Anfrage, kann Anmeldeinformationen senden

- SIP-Anfrage

- TCP- oder UDP-basiert S OPTIONS-Anfrage senden
- Kann die Antwort mit konfiguriertem Antwortcode oder erwarteter Zeichenfolge oder mit beiden vergleichen

- RTSP-Probe

- Ähnlich wie bei HTTP Probe. Senden einer OPTIONS- oder DESCRIPTION-Anforderung

- Radius-Datensammlung

- Ähnlich wie UDP-Probe. NAS-IP kann konfiguriert werden

Konfigurieren von Sonden

Sie können Diagnosetools auf dem ACE konfigurieren, um aktiv Verbindungen herzustellen und explizit Datenverkehr an Server zu senden. Die Tests bestimmen, ob der Diagnosestatus eines Servers durch die Serverantwort unterbrochen wird oder fehlschlägt.

Die Konfiguration aktiver Tests erfolgt in drei Schritten:

1. Konfigurieren Sie den Diagnosetest mit einem Namen, einem Typ und Attributen.
2. Ordnen Sie die Sonde einem der folgenden Anschlüsse zu:
 - Ein echter Server.
 - Eine echte Serverinstanz innerhalb einer bestimmten Serverfarm. Sie können einen einzelnen Prüfpunkt oder mehrere Prüfpunkte realen Servern innerhalb einer Serverfarm zuordnen.
 - Eine Serverfarm. Alle Server in der Serverfarm empfangen Tests der verknüpften Testtypen.
3. Den echten Server oder die Serverfarm in Dienst stellen.

Weitere Informationen finden Sie im [Cisco Application Control Engine Module Server Load Balancing Guide](#).

Beispiel einer Testkonfiguration

Das folgende Beispiel zeigt eine aktuelle Konfiguration, bei der der DNS-Datenverkehr über mehrere echte Server verteilt und UDP-Daten übertragen und empfangen werden, die mehrere Pakete umfassen. Für die Konfiguration wird ein UDP-Diagnosetest verwendet.

```
access-list ACL1 line 10 extended permit ip any any
```

```
probe udp UDP
  interval 5
  passdetect interval 10
  description THIS PROBE IS INTENDED FOR LOAD BALANCING DNS TRAFFIC
  port 53
  send-data UDP_TEST
```

```
rserver host SERVER1
  ip address 192.168.10.45
  inservice
rserver host SERVER2
  ip address 192.168.10.46
  inservice
```

```

rserver host SERVER3
  ip address 192.168.10.47
  inservice

serverfarm host SFARM1
  probe UDP
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
    inservice

class-map match-all L4UDP-VIP_114:UDP_CLASS
  2 match virtual-address 192.168.120.114 udp eq 53

policy-map type loadbalance first-match L7PLBSF_UDP_POLICY
  class class-default
    serverfarm SFARM1
policy-map multi-match L4SH-Gold-VIPs_POLICY
  class L4UDP-VIP_114:UDP_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_UDP_POLICY
    loadbalance vip icmp-reply
  nat dynamic 1 vlan 120
  connection advanced-options 1SECOND-IDLE

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.70 netmask 255.255.255.0 pat
  service-policy input L4SH-Gold-VIPs_POLICY
  no shutdown

ip route 10.1.0.0 255.255.255.0 192.168.120.254

```

Fehlerbehebung ACE-Integritätsüberwachung

In diesem Abschnitt wird die Fehlerbehebung bei allgemeinen Problemen mit der Testkonfiguration beschrieben.

Grundlegende Fehlerbehebung bei Tests

Die wichtigste Befehlsausgabe, die für die grundlegende Fehlerbehebung für den Prüfpunkt überprüft werden muss, ist `show probe detail`:

```
<#root>
```

```
ACE_module5/Admin#
```

```
show probe detail
```

probe : icmp-probe
type : ICMP, state :

ACTIVE

description :

```
-----  
port      : 0      address      : 0.0.0.0      addr type : -  
interval  : 10     pass intvl  : 10      pass count : 3  
fail count: 5     recv timeout: 5
```

```
----- probe results -----  
probe association  probed-address  probes
```

failed

passed

health

```
-----+-----+-----+-----+-----  
rserver      : rs1  
10.7.107.51   230      6      224
```

FAILED

Socket state : RESET

No. Passed states : 1 No. Failed states : 1

No. Probes skipped : 0 Last status code : 0

Last disconnect err :

Host Unreachable, no route found to destination

Last probe time : Sat Feb 18 18:24:18 2006

Last fail time : Sat Feb 18 18:24:08 2006

Last active time : Sat Feb 18 17:46:08 2006

- Diagnosezustand: "INIT", "FAILED", "PASSED", "DISABLED"
- Socket-Status: "RESET", "OPENING", "RECEIVING", "CLOSED"
- Letzter Trennungsfehler: Generische oder typspezifische Fehlermeldungen:
 - "Verbindung vom Server zurückgesetzt"
 - "Verbindung vom Server abgelehnt"
 - "Unbekannte oder ungültige Antwort"
 - "Zeitüberschreitung bei Serveröffnung (kein SYN ACK)"
 - "Serverantwort-Timeout (keine Antwort)"
 - "Erwartung einer Inkongruenz des Statuscodes"
 - "Ungültiger Statuscode empfangen"
 - "Interner Fehler: Skript wurde wegen Zeitüberschreitung beendet."
 - "ICMP Internal error: Kein Speicherplatz, Übertragungspfad ist voll"
 - "Interner Fehler: Fehler beim Erstellen der SSL-Sitzung."
 - "Empfangener veralteter ICMP-Pakt"
 - ...

Fehlerbehebung bei einem TCP-Testfehler

Die Felder "Socket State" (Sockelzustand) und "Last disconnect err" (Letzter Trennungsfehler) in der Ausgabe von `show probe detail` sollten die meisten erforderlichen Informationen bereitstellen. Überprüfen Sie dies zusammen mit einer Sniffer-Spur des Sonden-Datenverkehrs.

Fehlerbehebung bei einem UDP-Testfehler

Die Felder "Socket State" (Sockelzustand) und "Last disconnect err" (Letzter Trennungsfehler) in der Ausgabe von `show probe detail` sollten die meisten erforderlichen Informationen bereitstellen. Überprüfen Sie dies zusammen mit einer Sniffer-Spur des Sonden-Datenverkehrs.

Fehlerbehebung bei einem HTTP-Testfehler

In diesem ersten Szenario haben Sie einen HTTP-Test konfiguriert, aber der Status des tatsächlichen Servers wird als FEHLGESCHLAGEN angezeigt, und das Feld Letzter Trennungsfehler gibt an, dass ein ungültiger Statuscode empfangen wurde, wie in der Befehlsausgabe `show probe detail` angezeigt wird. Sie haben Ihren Server überprüft und er ist betriebsbereit. Eine Paketerfassung auf dem Server zeigt auch, dass alles in Ordnung ist. Wo liegt das Problem?

1. Zeigen Sie die Sondenstatusdetails an, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/Admin#

show probe detail

probe : HTTP_PROBE
type : HTTP
state : ACTIVE
description :

```
-----
port      : 80      address      : 0.0.0.0      addr type : -
interval  : 10      pass intvl   : 10      pass count : 3
fail count: 3      rcv timeout: 10
http method : GET
http url    : /
conn termination : GRACEFUL
expect offset : 0      , open timeout : 1
expect regex : -
send data   : -
```

```
----- probe results -----
associations ip-address      port porttype probes  failed  passed  health
-----+-----+-----+-----+-----+-----+-----
rserver      : SERVER1
              192.168.10.45  80    --      2        2        0      FAILED
```

Socket state : CLOSED
No. Passed states : 0 No. Failed states : 1

No. Probes skipped : 0 Last status code : 200 <----- Last status code from server

No. Out of Sockets : 0 No. Internal error: 0

Last disconnect err : Received invalid status code <-----

Last probe time : Tue Apr 7 16:17:26 2009
Last fail time : Tue Apr 7 16:17:16 2009
Last active time : Never

Das Feld Letzter Verbindungsabbruch zeigt an, dass der ACE einen ungültigen Statuscode erhalten hat. Dieser Fehler bedeutet, dass Sie den Befehl erwartungsstatus für den Prüfpunkt nicht konfiguriert haben.

2. Bestätigen Sie diese Feststellung, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/Admin#

show running-config probe

Generating configuration....

```
probe http HTTP_PROBE
  interval 10
  passdetect interval 10
  open 1
```

3. Korrigieren Sie das Problem, indem Sie die folgenden Befehle eingeben:

<#root>

ACE_module5/Admin#

config

Enter configuration commands, one per line. End with CNTL/Z.
ACE_module5/Admin(config)#

probe http HTTP_PROBE

ACE_module5/Admin(config-probe-http)#

expect status 200 200

<----- 200 indicates the 200 OK message from the server
ACE_module5/Admin(config-probe-http)#

end

4. Bestätigen Sie die Konfiguration, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/Admin#

show running-config probe

Generating configuration....

```
probe http HTTP_PROBE
  interval 10
  passdetect interval 10
  expect status 200 200
  open 1
```

5. Zeigen Sie die Teststatusdetails erneut an, und stellen Sie fest, dass der Serverstatuswert ERFOLGREICH lautet, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/Admin#

show probe HTTP_PROBE detail

probe : HTTP_PROBE
type : HTTP
state : ACTIVE
description :

port : 80 address : 0.0.0.0 addr type : -
interval : 10 pass intvl : 10 pass count : 3
fail count: 3 recv timeout: 10
http method : GET
http url : /
conn termination : GRACEFUL
expect offset : 0 , open timeout : 1
expect regex : -
send data : -

----- probe results -----
associations ip-address port porttype probes failed passed health
-----+-----+-----+-----+-----+-----+-----
rserver : SERVER1
192.168.10.45 80 -- 24 15 9 SUCCESS

Socket state : CLOSED
No. Passed states : 1 No. Failed states : 1
No. Probes skipped : 0 Last status code : 200
No. Out of Sockets : 0 No. Internal error: 0

Last disconnect err : - <----- No error indicated now. The probe is successful.

Last probe time : Tue Apr 7 16:21:05 2009
Last fail time : Tue Apr 7 16:17:16 2009
Last active time : Tue Apr 7 16:20:05 2009

Fehlerbehebung bei einem HTTPS-Testfehler

Zusätzlich zu den Methoden zur [Fehlerbehebung bei HTTP-Testfehlern](#) können Sie zur Fehlerbehebung bei HTTPS-Testfehlern eine SSL-Statistik verwenden. HTTPS-Prüfdatenverkehr wird im virtuellen Admin-Kontext ausgeführt. Zeigen Sie daher die Ausgabe des Befehls [show stats crypto client](#) in diesem Kontext an.

Anders als bei allen anderen Tests werden HTTPS-Tests auf der Datenebene des ACE ausgeführt. Sie sind daher Ressourcenzuweisungen für SSL-Datenverkehr auf dem ACE unterworfen.

Fehlerbehebung bei einem SNMP-Testproblem

In diesem Szenario haben Sie einen SNMP-Prüfpunkt konfiguriert, aber das Feld Letzter Trennungsfehler zeigt an, dass die Summe der Gewichtungen nicht den maximalen Gewichtungswert ergibt, wie in der Ausgabe des Befehls show probe detail angezeigt wird.

1. Zeigen Sie die Sondenstatusdetails an, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/test#

show probe detail

probe : SNMP_PROBE
type : SNMP
state : ACTIVE
description : snmp probe

port : 161 address : 0.0.0.0 addr type : -
interval : 15 pass intvl : 10 pass count : 3
fail count: 3 recv timeout: 10
version : 2c community : test_comm
oid string #1 : .1.3.6.1.2.1.4.3.0
type : ABSOLUTE max value : 1000000000
weight : 10000 threshold : 1000000000

----- probe results -----
associations ip-address port porttype probes failed passed health
-----+-----+-----+-----+-----+-----+-----
serverfarm : least-loaded, predictor least-loaded
real : SERVER1[0]
192.168.10.45 161 -- 0 0 0 INIT

Socket state : CLOSED
No. Passed states : 0 No. Failed states : 0
No. Probes skipped : 0 Last status code : 0
No. Out of Sockets : 0 No. Internal error: 30

Last disconnect err : Sum of weights don't add up to max weight value <----- Error condition

Last probe time : Never
Last fail time : Never
Last active time : Never

Server load : 16000 <----- Note the server load value

Der Grund für diesen Fehler ist, dass der Befehl weight konfiguriert werden muss, wenn mehrere OIDs für einen einzelnen Prüfpunkt konfiguriert sind, und von diesen OIDs, wenn Sie einer bestimmten OID Priorität einräumen möchten.

Die Summe der Gewichtungen muss 16000 ergeben (siehe Feld "Server Load"). Für eine einzelne OID hat der Befehl weight keine Bedeutung.

2. Zeigen Sie die Sondenkonfiguration an, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/Admin#

show running-config probe

probe snmp SNMP_PROBE
description snmp probe

```

port 161
interval 15
passdetect interval 10
version 2c
community TEST_COMM
oid .1.3.6.1.2.1.4.3.0
    type absolute max 1000000000
    weight 10000 <-----

```

In der obigen Konfiguration ist die Gewichtung für eine einzelne OID auf 10000 festgelegt. Der ACE erwartet, dass eine weitere OID in der Sonde konfiguriert wird, und die Summe beider Gewichtungen sollte 16000 betragen.

Die Konfiguration ist unvollständig, und der ACE erwartet zusätzliche Parameter in der Sondenkonfiguration. Da es keine weitere OID in der Konfiguration gibt, kann der ACE die Last nicht berechnen und deshalb erscheint die Fehlermeldung "Summe der Gewichte addiert sich nicht zum maximalen Gewichtswert".

3. Lösen Sie das Problem, indem Sie die Sondenkonfiguration wie folgt ändern:

<#root>

```

probe snmp SNMP_PROBE
    description test
    port 161
    interval 15
    passdetect interval 60
    version 2c
    community test_comm
    oid .1.3.6.1.2.1.4.3.0
        type absolute max 1000000000
        weight 10000
    oid .1.3.6.1.2.1.4.10.0
        type absolute max 1000000000

```

weight 6000 <----- 10000 + 6000 = 16000

4. Zeigen Sie die Sondenstatusdetails erneut an, indem Sie den folgenden Befehl eingeben:

<#root>

ACE_module5/test#

show probe SNMP_PROBE detail

```

probe      : snmp1
type       : SNMP
state      : ACTIVE
description : snmp probe

```

```

port      : 161    address      : 0.0.0.0    addr type : -

```

```

interval : 15      pass intvl : 10      pass count : 3
fail count: 3      recv timeout: 10
version : 2c      community : test_comm

oid string #1 : .1.3.6.1.2.1.4.3.0
type : ABSOLUTE      max value : 1000000000
weight : 10000      threshold : 1000000000

oid string #2 : .1.3.6.1.2.1.4.10.0
type : ABSOLUTE      max value : 1000000000
weight : 6000      threshold : 1000000000

----- probe results -----
associations ip-address      port porttype probes  failed  passed  health
-----+-----+-----+-----+-----+-----+-----
serverfarm : least-loaded, predictor least-loaded
  real : SERVER1[0]
        192.168.10.45  161  --      4143    0      4143    SUCCESS

Socket state : CLOSED
No. Passed states : 1      No. Failed states : 0
No. Probes skipped : 0      Last status code : 0
No. Out of Sockets : 0      No. Internal error: 0

Last disconnect err : - <----- No error indicated now. The probe is successful.

Last probe time : Mon Apr 6 09:12:54 2009
Last fail time : Never
Last active time : Sun Apr 5 15:57:28 2009
Server load : 0

```

Verwenden des Felds für den letzten Statuscode

Details zum letzten Statuscodefeld können für nicht triviale Tests bereitgestellt werden. Im Fall der skriptgesteuerten Sonde PROBENOTICE_PROBE bedeutet der Statuscode 30001, dass die Sonde erfolgreich ist und der Wert 30002 einen Fehler in den Sondenargumenten anzeigt. Der letzte Trennungsfehler für den Statuscode 30002 zeigt Folgendes an: "Die richtige Antwort vom Server wurde nicht empfangen". Das eigentliche Problem bezieht sich jedoch auf Argumente in der Testkonfiguration, die im Skript für den Test überprüft werden können.

<#root>

ACE_module5/Admin#

show probe TEST detail

```

probe : TEST
type : SCRIPTED
state : ACTIVE
description :
-----
port : 0      address : 0.0.0.0      addr type : -
interval : 15      pass intvl : 20      pass count : 3
fail count: 3      recv timeout: 10
script filename : PROBENOTICE_PROBE
----- probe results -----

```

probe association	probed-address	probes	failed	passed	health
serverfarm : sf1					
real : rs2[0]	23.0.0.5	4082	54	4028	SUCCESS

Socket state : RESET
 No. Passed states : 6 No. Failed states : 5
 No. Probes skipped : 8

Last status code : 30001 <----- Indicates success

No. Out of Sockets : 0 No. Internal error: 0
 Last disconnect err : -
 Last probe time : Wed Apr 8 04:44:41 2009
 Last fail time : Tue Apr 7 12:02:10 2009
 Last active time : Tue Apr 7 12:03:45 2009

Testdebugs

Für eine eingehendere Untersuchung von Fehlern bei Prüfungen kann folgender Aktionsplan durchgeführt werden:

- Erhalten Sie eine Show-Tech aus dem ACE-Admin-Kontext sowie aus dem Kontext, in dem die Tests konfiguriert sind.
- Sniffer-Traces zur Erfassung von Sondendatenverkehr starten
- Prüfpunktdebugs sammeln:

1) Erstellen Sie eine Debug-Protokolldatei:

```
debug logfile probe.log
```

2) Aktivieren Sie das Debugging:

```

hm-Fehler debuggen
hm-Ereignisse debuggen
hm info debuggen

```

3) Testfehler erfassen, Sniffer-Traces stoppen und Debugging deaktivieren:

```

keine debug hm all
oder
Unbug alle

```

4) Um die Protokollausgabe zu erfassen, geben Sie den folgenden Befehl ein, und kopieren Sie die Ausgabe in eine Textdatei:

show debug logfile probe.log

5) Löschen Sie die Datei aus dem ACE, sobald Sie sie kopiert haben:

debug-logfile test.log löschen

- Weitere Show-Tech aus dem ACE-Admin-Kontext sowie aus dem Kontext, in dem nach dem Test die Tests konfiguriert werden

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.