

Fehlerbehebung für DNA-Center und CMX-Server mit Wiedergabe

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Fehlerbehebung](#)

[DNA-Center](#)

[Schritt 1: Überprüfen Sie den Dienststatus der DNA-Maps auf der DNAC CLI.](#)

[Schritt 2: Überprüfen Sie, ob der CMX-Server auf der Web-GUI des DNA-Zentrums konfiguriert ist und durch Ping erreichbar ist.](#)

[Schritt 3: DnaMaps-Debugging-Ebene aktivieren](#)

[Schritt 4: Erfassen von Serviceprotokollen für die Analyse](#)

[Schritt 5: Trigger der DNAC API zum Senden der GET-Anforderung an den CMX-Server. Auf DNAC Web GUI](#)

[Zugehörige Informationen](#)

Einführung

Dieses Dokument beschreibt die grundlegende Konfiguration und Fehlerbehebung für die Wiedergabefunktion bei einer Integration von Digital Network Architecture Center (DNAC)- und Connected Mobile Experience (CMX)-Servern.

Die Wiedergabe ist Teil der CMX-Standortdienste, die den End-Host-Live-Standort und die Verfolgung über Wireless-Grundrisse bereitstellen, die in die DNA-Analytik-Funktion integriert werden können.

Voraussetzungen

Anforderungen

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

- DNA Center und CMX müssen bereits integriert sein.
- Fußbodenzuordnung wurde hinzugefügt oder in DNAC WebGUI > Design > Network Hierarchy Abschnitt importiert.
- Endhosts werden auf der entsprechenden Karte angezeigt.
- Grundlegende Linux-Grundlage.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- DNA Center mit Version 2.1.2.5

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle in diesem Dokument verwendeten Geräte wurden zu Testzwecken konfiguriert. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Fehlerbehebung

Dieser Abschnitt enthält Informationen, die Sie zur Fehlerbehebung bei Ihrer Konfiguration verwenden können.

DNA-Center

Schritt 1: Überprüfen Sie den Dienststatus der DNA-Maps auf der DNAC CLI.

```
$ magctl appstack status | egrep "STATUS|dna-maps-service"
NAMESPACE NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE
fusion dna-maps-service-7dff9d6b6-58qjb 1/1 Running 3 133d
<none>
```

Der Service muss den Status **Ready** und **Running** (Bereit) aufweisen.

```
$ magctl service status dna-maps-service
Name: dna-maps-service-7dff9d6b6-58qjb
Namespace: fusion
Node:
Start Time: Tue, 01 Dec 2020 19:43:27 +0000
Labels: pod-template-hash=3899858262
serviceName=dna-maps-service
version=7.14.117.62009
Annotations: <none>
Status: Running
IP:
Controlled By: ReplicaSet/dna-maps-service-7dff9d6b6
Containers:
dna-maps-service:
Container ID: docker://ddbe6999823a6830983611c1900c4a5d255b40b5a1957bef2d2ecddcd606a0b9
Image: maglev-registry.maglev-system.svc.cluster.local:5000/fusion/dna-maps-
service:7.14.117.62009
Image ID: docker-pullable://maglev-registry.maglev-system.svc.cluster.local:5000/fusion/dna-
maps-service@sha256:0b6510c1c29d260492647b586ffb714f1689ae7ec9d5f63905bb0ad4dac738c9
Ports: 22222/TCP, 11111/TCP
Host Ports: 0/TCP, 0/TCP
State: Running
.
```

```
Conditions:
Type Status
Initialized True
Ready True
ContainersReady True
PodScheduled True
```

```
.
.
.
.
```

```
Events: <none>
```

Hinweis: Es wird empfohlen, diesen Befehlssatz mehrmals (alle 10 Minuten) auszuführen, um sicherzustellen, dass der Dienst nicht ständig neu gestartet wird. Wenn der Dienst ständig neu geladen wird, überprüfen Sie den zweiten Befehl, um den letzten Abschnitt **Ereignisse** des Dienstes genauer zu überprüfen.

Schritt 2: Überprüfen Sie, ob der CMX-Server auf der Web-GUI des DNA-Zentrums konfiguriert ist und durch Ping erreichbar ist.

1. Klicken Sie auf  Symbol links oben in der DNAC Web GUI.
2. Gehen Sie zu **System > Einstellungen > DNS Spaces/CMX Servers**.
3. Bestätigen Sie, dass die CMX-Server dem korrekten Abschnitt hinzugefügt und als **registrierter** Status markiert wurden.
4. Von DNAC CLI aus pingen Sie den CMX-Server.
5. Klicken Sie auf  Symbol links oben in der DNAC Web GUI.
6. Gehen Sie zu **Design > Network Settings > Wireless**.
7. Bestätigen Sie im Abschnitt **DNA Spaces/CMX Servers**, dass der gewünschte Server in der globalen Hierarchie oder dem gewünschten Standort **ausgewählt** und **gespeichert** ist.

Schritt 3: DnaMaps-Debugging-Ebene aktivieren

1. Klicken Sie auf  Symbol links oben in der DNAC Web GUI
2. Gehen Sie zu **System > Einstellungen > Debugging Logs**.
3. Wählen Sie in der Liste Service die Option **dna-maps-service aus**. Wählen Sie anschließend **Debuggen** als Protokollierungsebene aus, und definieren Sie einen Zeitstempel.

Search Settings

- steairnwatch
- Umbrella
- vManage
- System Configuration ▼
- Debugging Logs
- High Availability
- Integration Settings
- Login Message
- Proxy Config
- System Health Notifications
- Terms and Conditions ▼
- Telemetry Collection
- Trust & Privacy ▼

Settings / System Configuration

Debugging Logs

Use this form to configure the logging of Cisco DNA Center internal processes and errors.

Service*
dna-maps-service ▼

Logger Name*
com.cisco Info

Logging Level
Debug ▼

Time Out
30 Mins ▼

[Save](#)

4. Klicken Sie auf **Speichern**, um den ausgewählten Dienst automatisch zu debuggen.

Hinweis: Vor der DNAC-Version 2.1.2.x (Wolverine) wird das Debuggen von dna-maps über die DNAC Web GUI nicht unterstützt. Es ist daher erforderlich, Service-Debug manuell über die DNAC CLI zu aktivieren.

Um den Debugging-Level auf dem DnaMaps-Service zu generieren, müssen wir eine Verbindung zur DNAC CLI herstellen und die API-Parameter auf den erforderlichen Wert ändern:

Authentifizierungstoken generieren

```
$ curl -s -k -u admin -X POST https:///dna/system/api/v1/auth/token | jq -r .Token
```

Verwenden Sie den folgenden CURL-Befehl, um das Debuggen für den Dienst dna-maps-zu aktivieren. Stellen Sie sicher, dass die Antwort als 200 OK zurückgegeben wird.

```
$ curl -k -i -X POST 'https://
```

```
    /api/v1/dna-maps-service/testing/logging/level?loggingLevel=5 ' --header 'X-Auth-Token:'  
--form 'Content-Type='
```

Beispiel:

```
$ curl -k -i -X POST 'https://x.x.x.x/api/v1/dna-maps-  
service/testing/logging/level?loggingLevel=5 ' --header 'X-Auth  
Token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiI1ZjQ2NDZkOGE3NmQ4ZDAwY2UzNzA4ZWYiLCJhdXRoU  
291cmNlIjoiaW50ZXJuYmwiLCJ0ZW5hbnROYWllIjoive5UMCIsInJvbGVzIjpbIjVmNDY0NmQ4YTc2ZDhkMDBjZTM3MDhlZ  
Sjdlc2ZDhkMDBjZTM3MDhlYyIsImV4cCI6MTYxODQyNDE3OCwiaWF0IjoxNjE4NDIwN  
Tc4LCJqdGkiOiIwYTU4YWVmYS03ZTgwLTRlOGUtODg4OC0wODBiNTk4ZW00NTciLCJlc2VybmFtZSI6ImFkbWluIn0.Li0BG  
N3VeVRiEwlrUsd94hnQt0xlcx0fJHCAtg0pQ7wx9MMC1UMImcuFabHXUgVmHcDrIAgds5GyBnNaPKfgs9qsvmjhJyVHZdT7_
```

```
f8YJ2BihkgEokfJbIkcb7Ulp7AqIzceACYpZXeBmfQtDCNDyJveoz1XLaKu69JYzArf8UaPzg3jHVoq9m6N5ohypMC-  
pmsp87-SbOoD-2x660K7AnkzqqtXw4vhyAp0atcYujPg0-8G4fkOPrLE-  
Cw6SXb8YonrjWVPbrBwfqENTr6sUj7SrPlH_CVBGzRG20YxaYvn_yeGG8E1pbCgDEK2UjyRaH-FM9BHIqY3TNUIf0mdvQ' -  
-form 'Content-Type='  
HTTP/1.1 100 Continue
```

```
HTTP/1.1 200 OK
```

Schritt 4: Erfassen von Serviceprotokollen für die Analyse

```
$ magctl service logs -rf dna-maps-service | lql
```

Dadurch werden die Protokolle in der CLI angezeigt. Es wird empfohlen, die Protokollierung im Anwendungsterminal zu speichern oder die Option `> log.txt` am Ende des Befehls hinzuzufügen, um die Ausgaben automatisch in einer Protokolldatei im Pfad `/home/maglev` zu speichern.

Schritt 5: Trigger der DNAC API zum Senden der GET-Anforderung an den CMX-Server. Auf DNAC Web GUI

1. Wechseln Sie in die Fußbodenübersicht, in der sich ein nicht funktionierender End-Host befindet.

2. **Klicken Sie** auf das blaue Symbol des End-Hosts. Daraufhin wird eine neue Registerkarte mit Informationen zum Client 360 mit der MAC-Adresse, der IP-Adresse und den zugehörigen Informationen geöffnet.

3. **Klicken Sie** erneut auf die Registerkarte **Wiedergabe**.

An diesem Punkt würde das DNA Center auf der CLI die Backend-Service-Debug mit der Konvertierung zwischen DNAC und CMX Server anzeigen.

Beispiel:

```
[2021-03-31 16:23:13,024 | DEBUG | scheduler-12 | | o.s.web.client.RestTemplate | - Created GET  
request for "https://x.x.x.x/api/location/v3/clients/count?associatedOnly=true "
```

```
[2021-03-31 16:23:13,024 | DEBUG | scheduler-12 | | o.s.web.client.RestTemplate | - Setting  
request Accept header to [application/xml, text/xml, application/json, application/*+xml,  
application/*+json]
```

```
[2021-03-31 16:23:16,028 | DEBUG | scheduler-12 | | o.s.web.client.RestTemplate | - GET request  
for "https://x.x.x.x/api/location/v3/clients/count?associatedOnly=true " resulted in 503  
(Service Unavailable); invoking error handler
```

```
[2021-03-31 16:23:16,030 | ERROR | scheduler-12 | | c.c.a.m.c.p.i.CMXVerificationRestClientImpl  
| - Error occurred during interaction with CMX to get client counts for CMX IpAddr x.x.x.x  
org.springframework.web.client.HttpServerErrorException: 503 Service Unavailable at  
org.springframework.web.client.DefaultResponseErrorHandler.handleError(DefaultResponseErrorHandl  
er.java:89)
```

```
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at  
org.springframework.web.client.RestTemplate.handleResponse(RestTemplate.java:708)
```

```
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at  
org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:661)
```

```
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at
```

