Troubleshoot DNA Center And CMX Server With Playback

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Troubleshoot

- DNA Center
  - Step 1. Verify DNA Maps Service status on DNAC CLI
  - Step 2. Verify CMX Server is configured on DNA Center Web GUI and it is reachable by ping
  - Step 3. Turn ON dna-maps debugging level
  - Step 4. Start collecting service logs for analysis
  - Step 5. Trigger the DNAC API to send the GET request to the CMX Server. On DNAC Web GUI

Related Information

Introduction

This document describes the basic configuration and troubleshooting for Playback feature on a Digital Network Architecture Center (DNAC) and Connected Mobile Experience (CMX) Server integration.

Playback is part of the CMX location services which provides end host live location and tracking over Wireless floor maps which can be integrated with DNA Analytics functionality.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- DNA Center and CMX must be already integrated.

- Floor map have been added or imported on DNAC **WebGUI > Design > Network Hierarchy** section.

- End hosts are displayed on the corresponding map.

- Basic linux foundation.

Components Used

The information in this document is based on these software and hardware versions:

- DNA Center running version 2.1.2.5

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure

that you understand the potential impact of any command.

Troubleshoot

This section provides information you can use in order to troubleshoot your configuration.

DNA Center

Step 1. Verify DNA Maps Service status on DNAC CLI

<#root>

$

magctl appstack status | egrep "STATUS|dna-maps-service"

NAMESPACE NAME READY STATUS RESTARTS AGE IP
fusion

dna-maps-service-7dffd9d6b6-58qjb

1

/1

Running

3 133d <none>

It would be required to have service in **Ready** and **Running** Status.

<#root>

$

magctl service status dna-maps-service

Name: dna-maps-service-7dffd9d6b6-58qjb
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-7dffd9d6b6
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-servic
Ports: 22222/TCP, 11111/TCP
Host Ports: 0/TCP, 0/TCP
State:
```

Running

.

.
<Output omitted>
.
.

```
Conditions:
Type              Status
Initialized
```

True

```
Ready
```

True

```
ContainersReady
```

True

```
PodScheduled
```

True

.

.
<Output omitted>
.
.

```
Events: <none>
```

Note: It is recommended to run this set of commands a couple of times (each 10 minutes) to ensure the service is not having constant restarts. In case the service is reloading constantly, refer to the second command to verify the last service **Events** section for more detail.

Step 2. Verify CMX Server is configured on DNA Center Web GUI and it is reachable by ping

1. Click on the ☰ icon on the top left of the DNAC Web GUI.

2. Go to **System** > **Settings** > **DNA Spaces/CMX Servers.**

3. Confirm the CMX Server/s are added into the correct section and marked as **Registered** status.

4. From DNAC CLI ping the CMX Server.

5. Click on the ☰ icon on the top left of the DNAC Web GUI.

6. Go to **Design** > **Network Settings** > **Wireless.**

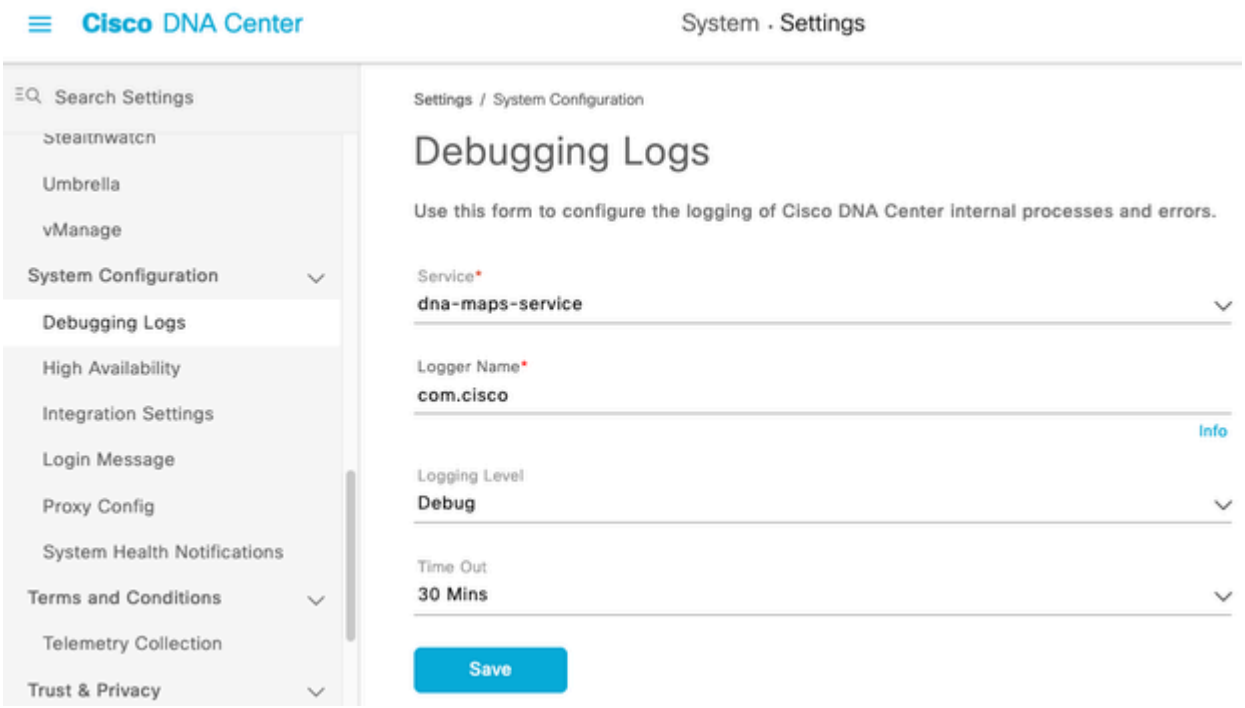
7. Confirm on the **DNA Spaces/CMX Servers** section that the desired server is **Selected** and **Saved** on the Global Hierarchy or the desired site.

Step 3. Turn ON dna-maps debugging level

1. Click on the ☰ icon on the top left of the DNAC Web GUI

2. Go to **System** > **Settings** > **Debugging logs.**

3. On Service list, select **dna-maps-service**. Then select **Debug** as Logging Level and define a timestamp.



4. Click on **Save** and this automatically starts debugging our selected service.

Note: Before DNAC version 2.1.2.x (Wolverine), running dna-maps debugging is not supported through DNAC Web GUI. So, it is required to turn ON service debugs manually on DNAC CLI

To generate debug level on dna-maps-service we would need to connect to DNAC CLI and modify the API parameters to the needed value:

Generate Authentication Token

<#root>

```
$ curl -s -k -u admin -X POST https://
```

```
<DNAC_IP>
```

```
/dna/system/api/v1/auth/token  | jq -r .Token
```

Use this CURL command to enable debugging for the dna-maps-service. Ensure that the response comes back as a 200 OK

<#root>

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

```
<DNAC_IP>
```

```
/api/v1/dna-maps-service/testing/logging/level?loggingLevel=
```

```
5
```

```
 ' --header 'X-Auth-Token:
```

```
<TOKEN_FROM_PREVIOUS_STEP>
```

```
' --form 'Content-Type='
```

Example:

<#root>

```
$ curl -k -i -X POST 'https://x.x.x.x/api/v1/dna-maps-service/testing/logging/level?loggingLevel=5 ' --
HTTP/1.1 100 Continue

HTTP/1.1

200 OK
```

Step 4. Start collecting service logs for analysis

<#root>

$

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

This starts displaying the logs on the CLI, it is recommended to save the logging on the application terminal or add the option **> log.txt** at the end of the command to automatically store the outputs into a log file on /home/maglev path.

Step 5. Trigger the DNAC API to send the GET request to the CMX Server. On DNAC Web GUI

1. Go into the Floor map where a non working end host is located at.

2. **Click** on the end host blue icon. This opens a new side tab with the Client 360 information with the MAC Address, IP and related information.

3. **Click** again on the **Playback** tab.

At this point, the DNA Center would be displaying on CLI the backend service debugs with the conversion between DNAC and CMX Server.

Example:

<#root>

```
|2021-03-31 16:23:13,024 | DEBUG | scheduler-12 | | o.s.web.client.RestTemplate | - Created GET request

|2021-03-31 16:23:13,024 | DEBUG | scheduler-12 | | o.s.web.client.RestTemplate | - Setting request Acc

|2021-03-31 16:23:16,028 | DEBUG | scheduler-12 | | o.s.web.client.RestTemplate | - GET request for "ht
```

```
|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.springframewo

~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at org.springframework.web.client.RestTemplate.handleRe
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at org.springframework.web.client.RestTemplate.doExecut
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at org.springframework.web.client.RestTemplate.execute(
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at org.springframework.web.client.RestTemplate.exchange
~[spring-web-4.3.19.RELEASE.jar:4.3.19.RELEASE] at com.cisco.apicem.maps.cmx.proxy.impl.CMXVerificationI
~[cmx-proxy-7.14.264.62702.jar:7.14.264.62702] at com.cisco.csg.ngmaps.impl.apicem.integration.diagnost
~[classes/:na] at jdk.internal.reflect.GeneratedMethodAccessor118.invoke(Unknown Source) ~[na:na]
```

⚠ **Caution**: Once troubleshooting is done, it is required to set again debugging level to default in case it was configured on DNAC CLI

<#root>

```
$ curl -k -i -X POST 'https://
<DNAC_IP>
/api/v1/dna-maps-service/testing/logging/level?loggingLevel=
3
 ' --header 'X-Auth-Token:
<TOKEN_FROM_PREVIOUS_STEP>
' --form 'Content-Type='
```

Example:

<#root>

```
$ curl -k -i -X POST 'https://x.x.x.x/api/v1/dna-maps-service/testing/logging/level?loggingLevel=
```

3

```
' --header 'X-Auth-Token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiI1ZjQ2NDZkOGE3NmQ4ZDAwY2UzNzA4Z
HTTP/1.1 100 Continue

HTTP/1.1
```

200 OK

Related Information

- **Cisco DNA Assurance Release Notes**