

Fehlerbehebung für den CloudCenter-Fehler "Keine Kommunikation mit dem Orchestrator möglich" mit Fehler 408

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Problem](#)

[Lösung](#)

[AMQP](#)

[CCO](#)

[CCM](#)

[Überprüfen](#)

[CCM](#)

[AMQP](#)

[CCO](#)

Einführung

In diesem Dokument wird beschrieben, wie der CloudCenter-Fehler "Kommunikation mit Orchestrator nicht möglich" mit Fehler 408 behoben wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Linux-Schnittstelle
- Umgebung virtueller Systeme
- VIM

Verwendete Komponenten

Cisco empfiehlt Kenntnisse in folgenden Bereichen:

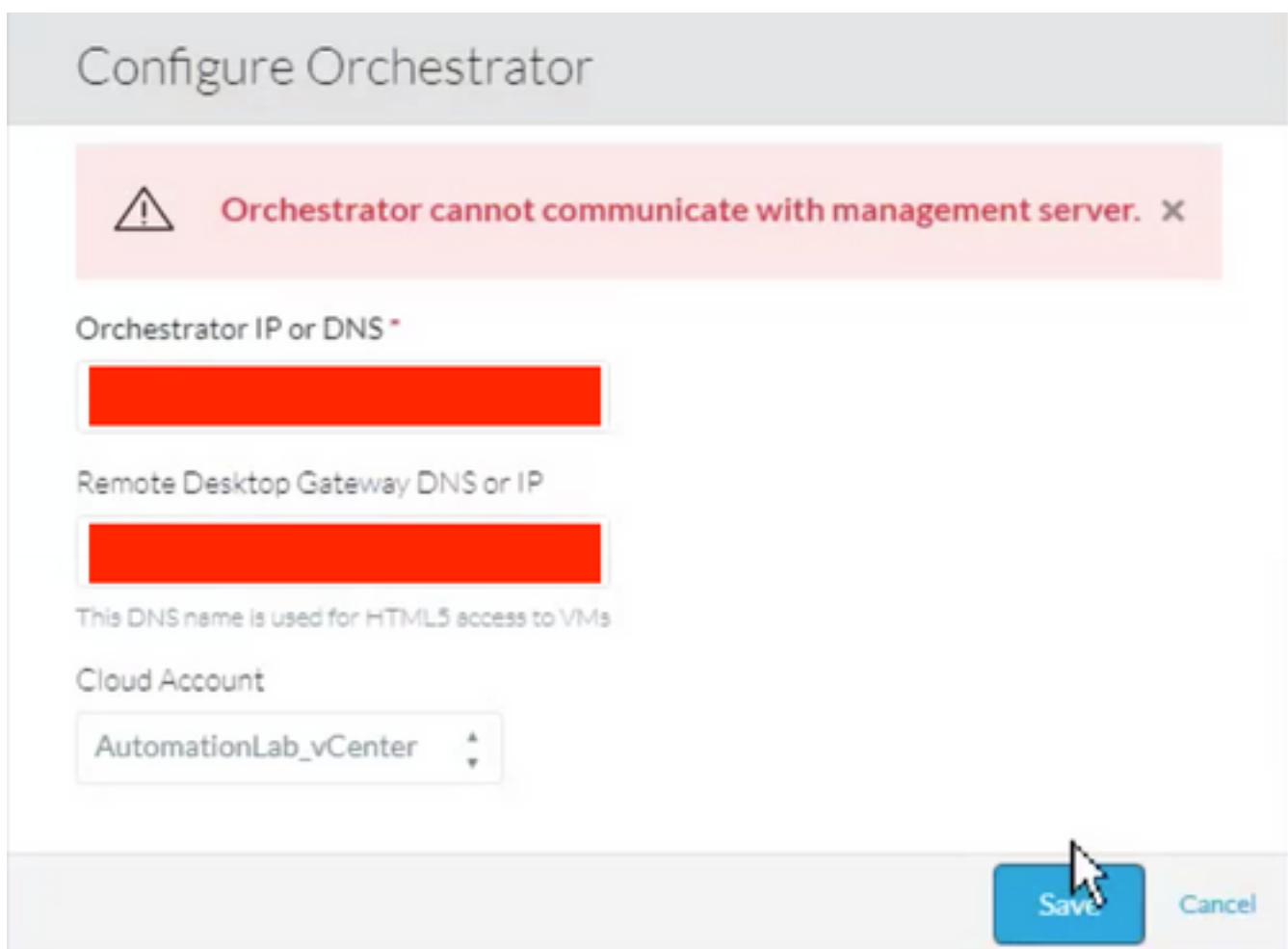
- CloudCenter-Appliances
- CloudCenter-Architektur
- Linux-Betriebssystem
- CCM (CloudCenter-Management)

- CCO (CloudCenter Orchestrator)
- AMQP (Advanced Message Queuing Protocol)

Die Informationen in diesem Dokument wurden von den Geräten in einer spezifischen privaten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Problem

Stromausfälle, unerwartete Neustarts oder Netzwerkausfälle über längere Zeiträume können dazu führen, dass die CloudCenter-Appliance synchronisiert wird. Die folgenden Prüfungen müssen durchgeführt werden, um sicherzustellen, dass die Einheiten ordnungsgemäß verbunden sind. Wenn der Orchestrator auf der grafischen Benutzeroberfläche (CCM GUI) des CloudCenter Managers konfiguriert wird, kann der Fehler wie im Bild dargestellt auftreten.



Beim Überprüfen der CCO-Protokolle kann der folgende Fehler angezeigt werden:

```
Caused by: java.net.ConnectException: Connection refused (Connection refused)
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at
org.apache.http.conn.ssl.SSLConnectionSocketFactory.connectSocket(SSLConnectionSocketFactory.jav
```

```
a:337)
at
org.apache.http.impl.conn.DefaultHttpClientConnectionOperator.connect(DefaultHttpClientConnectionOperator.java:134)
... 87 more
java.lang.RuntimeException: Failed to connect to CCM, please check network connection between CCM and CCO. JobId: 21912
at
com.osmosix.commons.mgmtserver.impl.MgmtServerServiceImpl.getUserCloudAccountByJobId(MgmtServerServiceImpl.java:236)
at
com.osmosix.gateway.persistence.impl.hazelcast.AbstractDistributedJobDaoImpl.find(AbstractDistributedJobDaoImpl.java:109)
at
com.osmosix.gateway.persistence.impl.hazelcast.AbstractDistributedJobDaoImpl.find(AbstractDistributedJobDaoImpl.java:17)
at com.osmosix.gateway.lifecycle.impl.AbstractLifecycle.getJob(AbstractLifecycle.java:207)
at com.osmosix.gateway.lifecycle.helpers.LifecycleReaper.reapApp(LifecycleReaper.java:62)
at com.osmosix.gateway.lifecycle.helpers.LifecycleReaper.reapDeadApps(LifecycleReaper.java:45)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at
org.springframework.scheduling.support.ScheduledMethodRunnable.run(ScheduledMethodRunnable.java:65)
at
org.springframework.scheduling.support.DelegatingErrorHandlingRunnable.run(DelegatingErrorHandlingRunnable.java:54)
at
org.springframework.scheduling.concurrent.ReschedulingRunnable.run(ReschedulingRunnable.java:81)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201(ScheduledThreadPoolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:293)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:748)
```

Lösung

Ist ein Neustart der CloudCenter-Komponenten einzeln erforderlich, um den Handshake zwischen den Komponenten zu aktualisieren?

AMQP

Schritt 1: Melden Sie sich als root an.

Schritt 2: AMQP-Dienst neu starten

In allen Versionen bis 4.8.1.2

```
# /etc/init.d/tomcatgua restart
```

Versionen ab 4.8.2

```
# systemctl restart rabbit
```

CCO

Schritt 1: Melden Sie sich als root an.

Schritt 2: CCO-Service neu starten

In allen Versionen bis 4.8.1.2

```
# /etc/init.d/tomcat restart
```

Versionen ab 4.8.2

```
# systemctl restart cco
```

CCM

Schritt 1: Melden Sie sich als root an.

Schritt 2: Neustart des CCM-Service

In allen Versionen bis 4.8.1.2

```
# /etc/init.d/tomcat restart
```

Versionen ab 4.8.2

```
# systemctl restart ccm
```

Überprüfen

Es ist wichtig, dass alle Appliances korrekt angeschlossen sind. Hierzu muss jede der CloudCenter-Komponenten eingecheckt werden.

CCM

Schritt 1: Melden Sie sich als root an.

Schritt 2: Überprüfen Sie, ob der Tomcat-Service (vor 4.8.2) oder der CCM-Service (nach 4.8.2) tatsächlich ausgeführt werden.

In allen Versionen bis 4.8.1.2

```
[root@localhost ~]# ps -ef | grep -i tomcat
```

Versionen ab 4.8.2

```
[root@localhost ~]# systemctl status ccm
```

Schritt 3: Wenn Telnet installiert ist, kann vom CCO versucht werden, den CCM anzuzeigen. Auf diese Weise wird die Kommunikation ermöglicht.

```
[root@cliqr-centos7-base-image ~]# telnet 10.31.127.41 8443
Trying 10.31.127.41...
Connected to 10.31.127.41.
Escape character is '^'].
```

Tritt ein Fehler auf, ist keine Kommunikation möglich. Das muss behoben werden.

Schritt 4: Wenn der Orchestrator in der CCM-GUI konfiguriert werden soll, wird ein Hostname verwendet, stellen Sie sicher, dass der Hostname in der Datei "/etc/hosts" vorhanden ist.

```
[root@cliqr-centos7-base-image ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1   devCC
10.31.127.42 cco
```

AMQP

Schritt 1: Melden Sie sich als root an.

Schritt 2: Stellen Sie sicher, dass vom AMQP eine Verbindung zu jedem der vorhandenen CCOs hergestellt wurde.

```
[root@localhost ~]# rabbitmqctl list_connections -p /cliqr
Listing connections ...
cliqr10.31.127.4233062running
cliqr_worker10.31.127.4233130running
cliqr_worker10.31.127.5938596running
cliqr_worker10.31.127.6749781running
cliqr_worker10.31.127.7949778running
cliqr_worker10.31.127.8549786running
```

Im vorherigen Befehl werden die Verbindungen zum CCO in der Zeile mit dem Benutzer cliqr angezeigt (in diesem Fall ist nur ein CCO vorhanden).

Im Fall von Hochverfügbarkeit (HA) und AMQP unter einem Load Balancer wird eine Verbindung pro CCO angezeigt, die mit der Load Balancer-IP des AMQP verbunden ist (im folgenden Beispiel gibt es 2 CCOs).

```
[root@amqp-azrel ~]# rabbitmqctl list_connections -p /cliqr
Listing connections ...
cliqr15.1.0.1035788running
cliqr15.1.0.1036212running
cliqr_worker15.1.0.1037714running
cliqr_worker15.1.0.1038362running
cliqr_worker15.1.0.1041102running
```

Ist dies nicht der Fall, starten Sie den Tomcatgua-Prozess (vor 4.8.2) oder den Kaninchendienst (nach 4.8.2) neu

CCO

Schritt 1: Melden Sie sich als root an.

In allen Versionen bis 4.8.1.2

```
[root@localhost ~]# ps -ef | grep -i tomcat
```

Versionen ab 4.8.2

```
[root@localhost ~]# systemctl status cco
```

Schritt 3: Überprüfen Sie, ob Verbindungen zum CCM hergestellt wurden. Er sollte ebenfalls im Status CLOSE_WAIT angezeigt werden (in diesem Fall ist unser CCM am 10.31.127.41).

```
[root@cliqr-centos7-base-image ~]# netstat -anp | grep 10.31.127.41
```

tcp	86	0	10.31.127.42:38542	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38562	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38546	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38566	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38556	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38554	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38550	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38564	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38560	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38568	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38552	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38558	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38570	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38548	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38572	10.31.127.41:8443	CLOSE_WAIT	1330/java
tcp	86	0	10.31.127.42:38544	10.31.127.41:8443	CLOSE_WAIT	1330/java