

# 對CloudCenter錯誤「無法與協調器通訊」進行故障排除，錯誤為408

## 目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[問題](#)

[解決方案](#)

[AMQP](#)

[CCO](#)

[CCM](#)

[驗證](#)

[CCM](#)

[AMQP](#)

[CCO](#)

## 簡介

本文檔介紹如何對CloudCenter錯誤「Unable to communication with orchestrator」進行故障排除，錯誤為408

## 必要條件

### 需求

思科建議您瞭解以下主題：

- Linux介面
- 虛擬機器環境
- VIM

### 採用元件

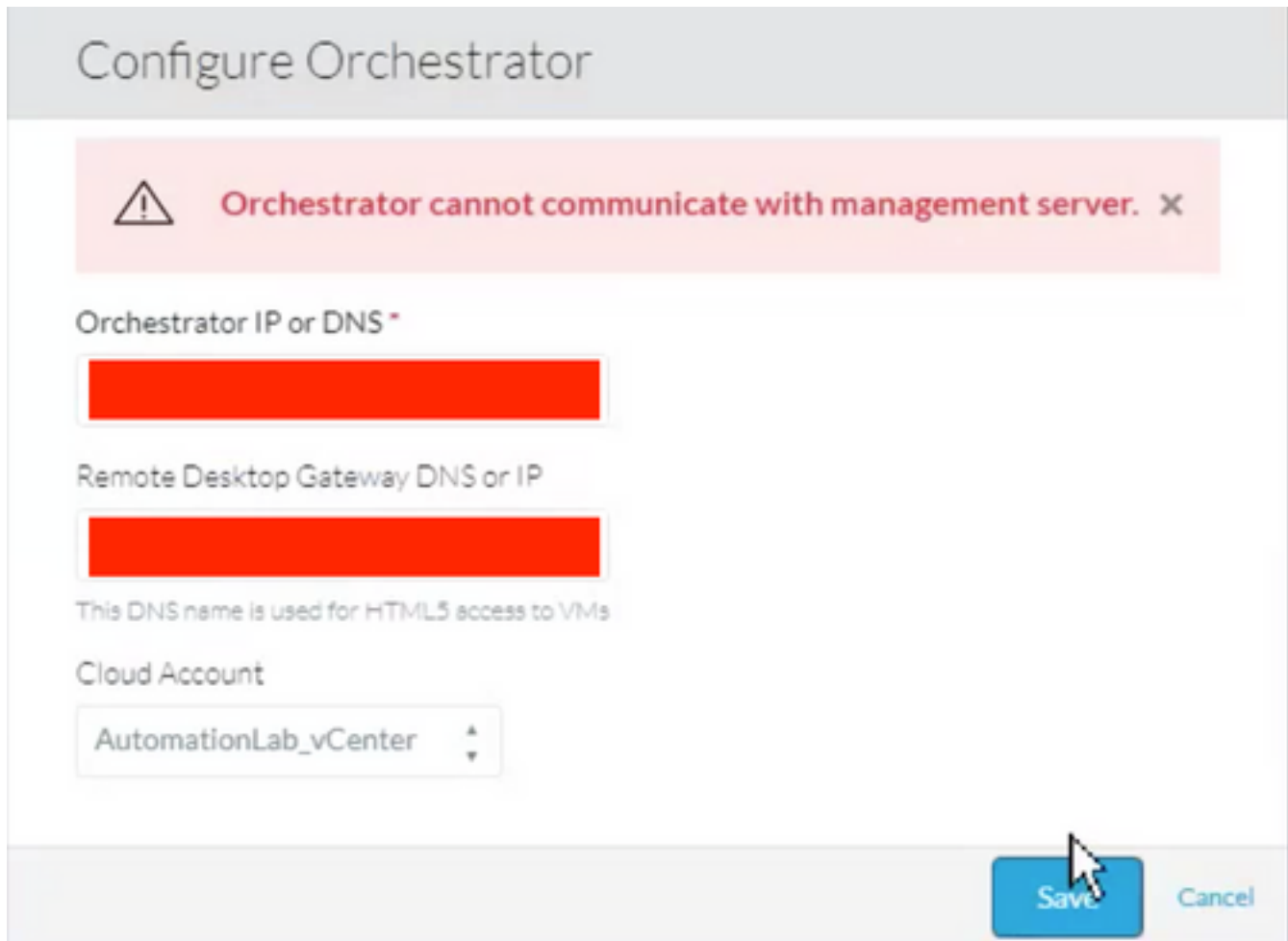
思科建議瞭解以下內容：

- CloudCenter裝置
- CloudCenter架構
- Linux作業系統
- CCM(CloudCenter Management)
- CCO(CloudCenter Orchestrator)
- AMQP (進階訊息佇列通訊協定)

本文中的資訊是根據特定私人實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

## 問題

停電、意外重啟或長時間網路故障可能導致CloudCenter裝置不同步。必須執行以下檢查才能知道裝置是否正確連線在CloudCenter Manager圖形使用者介面(CCM GUI)上配置協調器時，使用者可能會收到如下圖所示的錯誤。



檢查CCO日誌時，可能會顯示以下錯誤：

```
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.java: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(MgmtServerS
erviceImpl.java:236)
at
com.osmosix.gateway.persistence.impl.hazelcast.AbstractDistributedJobDaoImpl.find(AbstractDistri
butedJobDaoImpl.java:109)
at
com.osmosix.gateway.persistence.impl.hazelcast.AbstractDistributedJobDaoImpl.find(AbstractDistri
butedJobDaoImpl.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(DelegatingErrorHandli
ngRunnable.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$201(ScheduledThreadP
oolExecutor.java:180)
at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExec
utor.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)
```

## 解決方案

必須逐個重新啟動CloudCenter元件以刷新元件之間的握手

### AMQP

步驟1.以根使用者身份登入

步驟2.重新啟動AMQP服務

在所有版本上，最高為4.8.1.2

```
# /etc/init.d/tomcatgua restart
從4.8.2開始的版本
```

```
# systemctl restart rabbit
```

### CCO

步驟1.以根使用者身份登入

## 步驟2.重新啟動CCO服務

在所有版本上，最高為4.8.1.2

```
# /etc/init.d/tomcat restart  
從4.8.2開始的版本
```

```
# systemctl restart cco
```

## CCM

步驟1.以根使用者身份登入

步驟2.重新啟動CCM服務

在所有版本上，最高為4.8.1.2

```
# /etc/init.d/tomcat restart  
從4.8.2開始的版本
```

```
# systemctl restart ccm
```

## 驗證

所有裝置必須正確連線，為此，需要檢查每個CloudCenter元件。

## CCM

步驟1.以根使用者身份登入

步驟2.檢查tomcat ( 4.8.2之前的版本 ) 或CCM服務 ( 4.8.2之後的版本 ) 是否實際正在運行

在所有版本上，最高為4.8.1.2

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

從4.8.2開始的版本

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

步驟3.如果安裝了telnet，則可以嘗試從CCO訪問CCM，這將有助於瞭解通訊是否可行

```
[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 '^]'.  
^C
```

如果發生錯誤，則無法通訊。這個問題必須解決。

步驟4.如果要在CCM GUI上配置orchestrator要使用主機名，請確保主機名存在於/etc/hosts檔案中

```
[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

步驟1.以根使用者身份登入

步驟2.檢查是否存在從AMQP到每個現有CCO的連線。

```
[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
```

在上一個命令中，與CCO的連線可以在cliqr使用者的行中看到（這種情況下只有一個CCO）

如果高可用性(HA)和AMQP在負載平衡器之下，那麼您將看到每個CCO有一個連線與AMQP的負載平衡器IP連線（在以下示例中，有2個CCO）

```
[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
```

如果不是，請重新啟動tomcatgua流程（4.8.2之前的版本）或Rabbit服務（4.8.2之後的版本）

## CCO

步驟1.以根使用者身份登入

在所有版本上，最高為4.8.1.2

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

從4.8.2開始的版本

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

步驟3.檢查是否已建立與CCM的連線。它也應出現在CLOSE\_WAIT狀態（本例中我們的CCM位於10.31.127.41上）

```
[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