

CloudCenterエラー「Unable to communicate with orchestrator」のエラー408のトラブルシューティング

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[問題](#)

[解決方法](#)

[AMQP](#)

[CCO](#)

[CCM](#)

[確認](#)

[CCM](#)

[AMQP](#)

[CCO](#)

概要

このドキュメントでは、CloudCenterエラー「Unable to communicate with orchestrator」（エラー408）のトラブルシューティング方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux インターフェイス
- 仮想マシン環境
- VIM

使用するコンポーネント

シスコでは、次の分野の知識を推奨しています。

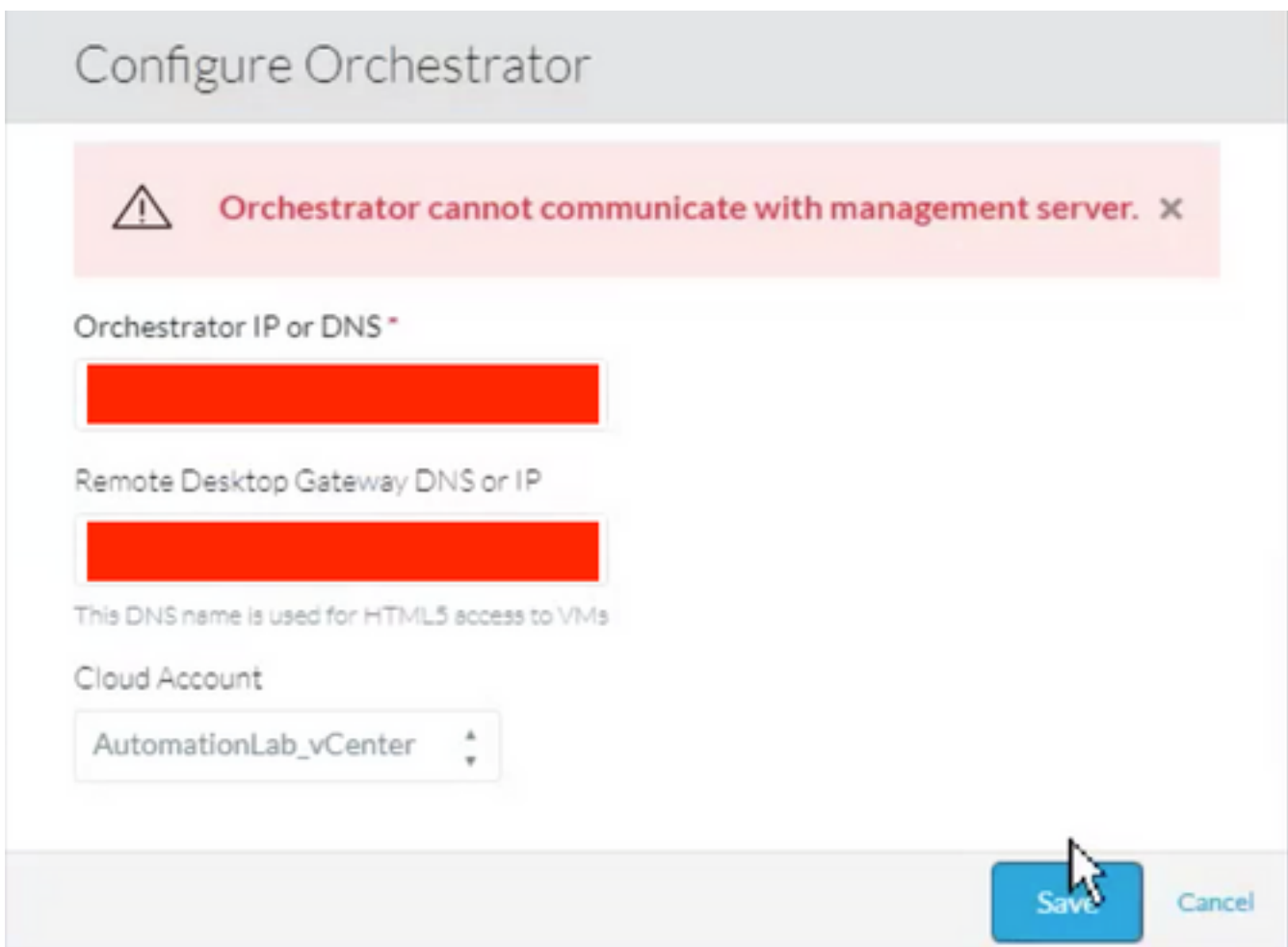
- CloudCenterアプライアンス
- CloudCenterアーキテクチャ
- Linux O.S
- CCM(CloudCenter Management)

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

このドキュメントの情報は、特定のプライベートラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

問題

停電、予期しない再起動、または長時間のネットワーク障害が原因で、CloudCenterアプライアンスの同期が解除されることがあります。アプライアンスが正しく接続されていることを確認するには、次のチェックを行う必要があります。CloudCenter ManagerのGraphical User Interface(CCM GUI)でオーケストレータを設定すると、図に示すようにエラーが表示されることがあります。



CCOログを確認すると、次のエラーが表示されます。

```
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)
```

解決方法

コンポーネント間のハンドシェイクを更新するために、CloudCenterコンポーネントを1つずつ再起動する必要があります

AMQP

ステップ1:rootとしてログインします。

ステップ2:AMQPサービスを再起動します。

4.8.1.2までのすべてのバージョン

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

4.8.2以降のバージョン

```
# systemctl restart rabbit
```

CCO

ステップ1:rootとしてログインします。

ステップ2:CCOサービスの再起動

4.8.1.2までのすべてのバージョン

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

4.8.2以降のバージョン

```
# systemctl restart cco
```

CCM

ステップ1:rootとしてログインします。

ステップ2:CCMサービスの再起動

4.8.1.2までのすべてのバージョン

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

4.8.2以降のバージョン

```
# systemctl restart ccm
```

確認

すべてのアプライアンスが正しく接続されていることが重要です。そのため、CloudCenterの各コンポーネントをチェックインする必要があります。

CCM

ステップ1:rootとしてログインします。

ステップ2:tomcat(4.8.2より前)またはCCMサービス(4.8.2以降)が実際に実行されていることを確認します

4.8.1.2までのすべてのバージョン

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

4.8.2以降のバージョン

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

ステップ3:Telnetがインストールされている場合、CCOからCCMに向けて試行を行うことができます。これにより、通信が可能であることを理解できます

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

エラーが発生すると、通信できません。これは修正する必要があります。

ステップ4:CCM GUIでオーケストレータを設定する場合は、ホスト名が/etc/hostsファイルに存在することを確認します

```
[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

ステップ1:rootとしてログインします。

ステップ2:AMQPから既存の各CCOへの接続が確立されていることを確認します。

```
[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
```

前のコマンドでは、CCOへの接続はcliqrユーザの行に表示されます(この場合、CCOは1つだけです)

ロードバランサでハイアベイラビリティ(HA)とAMQPを使用すると、AMQPのロードバランサIPに接続されたCCOごとに1つの接続が表示されます(次の例では2つのCCOがあります)

```
[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
```

そうでない場合は、tomcatguaプロセス(4.8.2以前)またはrabbitサービス(4.8.2以降)を再起動します

CCO

ステップ1:rootとしてログインします。

4.8.1.2までのすべてのバージョン

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

4.8.2以降のバージョン

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

ステップ3:CCMへの接続が確立されていることを確認します。CLOSE_WAITステータスにも表示されます(この場合、CCMは10.31.127.41です)

```
[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