排除CloudCenter错误“Unable to communication with orchestrator”，错误为408

目录

[简介](#)
[先决条件](#)
[要求](#)
[使用的组件](#)
[问题](#)
[解决方案](#)
[AMQP](#)
[CCO](#)
[CCM](#)
[验证](#)
[CCM](#)
[AMQP](#)
[CCO](#)

简介

本文档介绍如何排除CloudCenter错误“无法与协调器通信”(Unable to communication with orchestrator)，错误为408

先决条件

要求

Cisco 建议您了解以下主题：

- Linux界面
- 虚拟机环境
- VIM

使用的组件

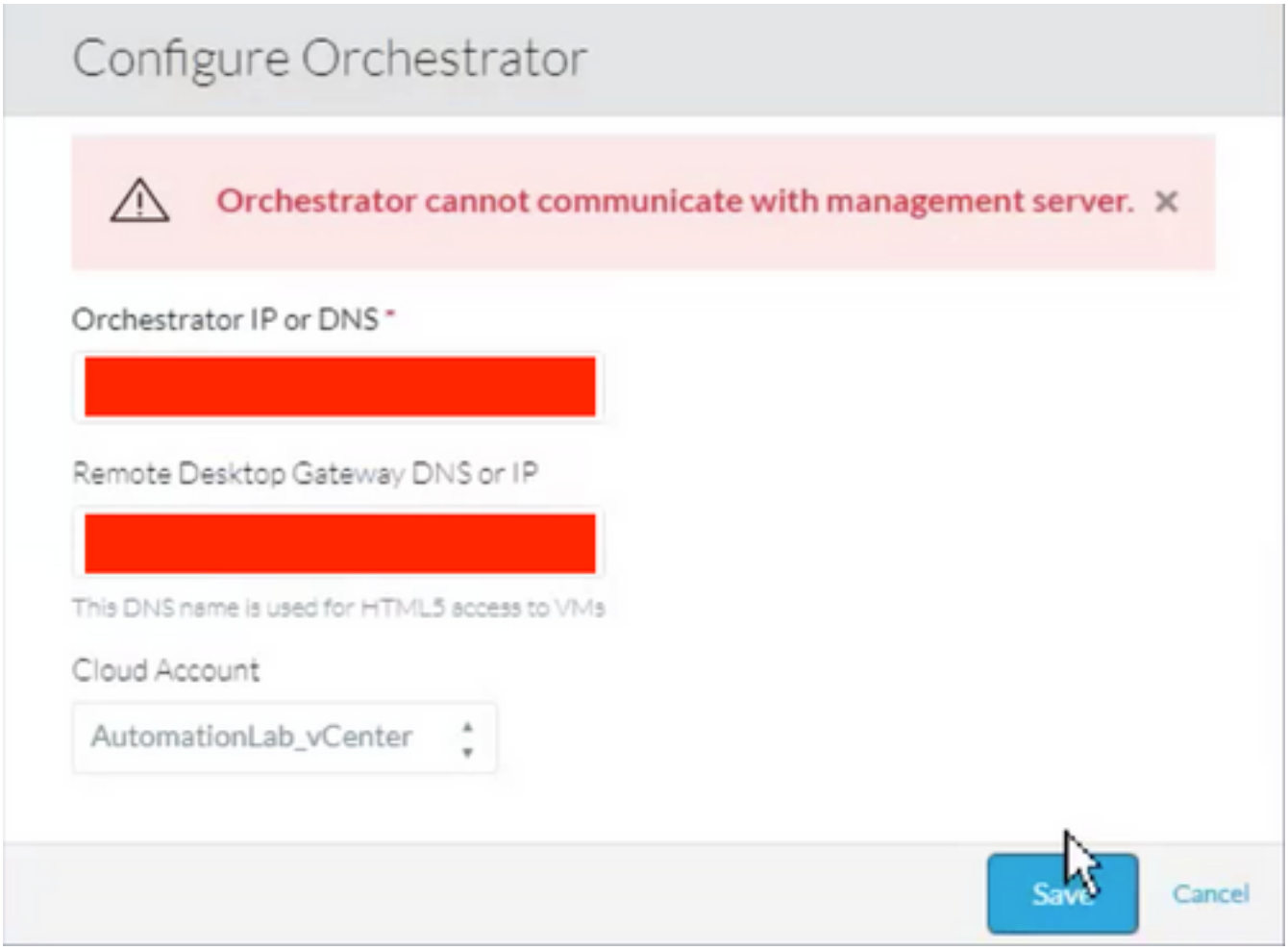
思科推荐以下方面的知识：

- CloudCenter设备
- CloudCenter架构
- Linux操作系统
- CCM（CloudCenter管理）
- CCO(CloudCenter Orchestrator)
- AMQP（高级消息队列协议）

本文档中的信息基于特定专用实验室环境中的设备。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

问题

停电、意外重新启动或长时间的网络故障可能导致CloudCenter设备不同步。必须执行以下检查，才能知道设备已正确连接。在CloudCenter Manager图形用户界面(CCM GUI)上配置协调器时，用户可能会收到如图所示的错误。



检查CCO日志时，可能显示以下错误：

```
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(DefaultHttpClientConnectio
nOperator.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)
```

解决方案

是否需要逐个重新启动CloudCenter组件以刷新组件之间的握手

AMQP

步骤1.以root用户身份登录

步骤2.重新启动AMQP服务

在所有版本上，最高版本为4.8.1.2

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

从4.8.2开始的版本

```
# systemctl restart rabbit
```

CCO

步骤1.以root用户身份登录

步骤2.重新启动CCO服务

在所有版本上，最高版本为4.8.1.2

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

从4.8.2开始的版本

```
# systemctl restart cco
```

CCM

步骤1.以root用户身份登录

步骤2.重新启动CCM服务

在所有版本上，最高版本为4.8.1.2

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

从4.8.2开始的版本

```
# systemctl restart ccm
```

验证

所有设备都正确连接非常重要，因此必须检入每个CloudCenter组件。

CCM

步骤1.以root用户身份登录

步骤2.检查tomcat（4.8.2之前）或CCM服务（4.8.2之后）是否实际运行

在所有版本上，最高版本为4.8.1.2

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

从4.8.2开始的版本

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

步骤3.如果安装了telnet，则可以尝试从CCO到CCM，这将使您了解通信是可能的

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

如果发生错误，则无法通信。这必须得解决。

步骤4.如果要在CCM GUI上配置协调器，将使用主机名，请确保该主机名在/etc/hosts文件中存在

```
[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

步骤1.以root用户身份登录

步骤2.检查是否已建立从AMQP到每个现有CCO的连接。

```
[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
```

在上一命令中，可以在cliqr用户（本例中只有一个CCO）的行中看到指向CCO的连接

在负载均衡器下的高可用性(HA)和AMQP中，您将看到每个CCO与AMQP的负载均衡器IP连接一个连接（在以下示例中，有2个CCO）

```
[root@amqp-azre1 ~]# 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
```

如果情况并非如此，请重新启动tomcatgua进程（在4.8.2之前）或兔服务（在4.8.2之后）

CCO

步骤1.以root用户身份登录

在所有版本上，最高版本为4.8.1.2

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

从4.8.2开始的版本

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

步骤3.检查是否已建立到CCM的连接。它也应显示为CLOSE_WAIT状态（在本例中，我们的CCM位于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
```