

Probleemoplossing CloudCenter-fout "Kan niet met orchestrator communiceren" met fout 408

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Probleem](#)

[Oplossing](#)

[AMP](#)

[CCO](#)

[CCM](#)

[Verifiëren](#)

[CCM](#)

[AMP](#)

[CCO](#)

Inleiding

In dit document wordt beschreven hoe CloudCenter-fout "kan niet met orchestrator communiceren" met fout 408

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Linux-interface
- Virtuele machinemodaliteit
- VIM

Gebruikte componenten

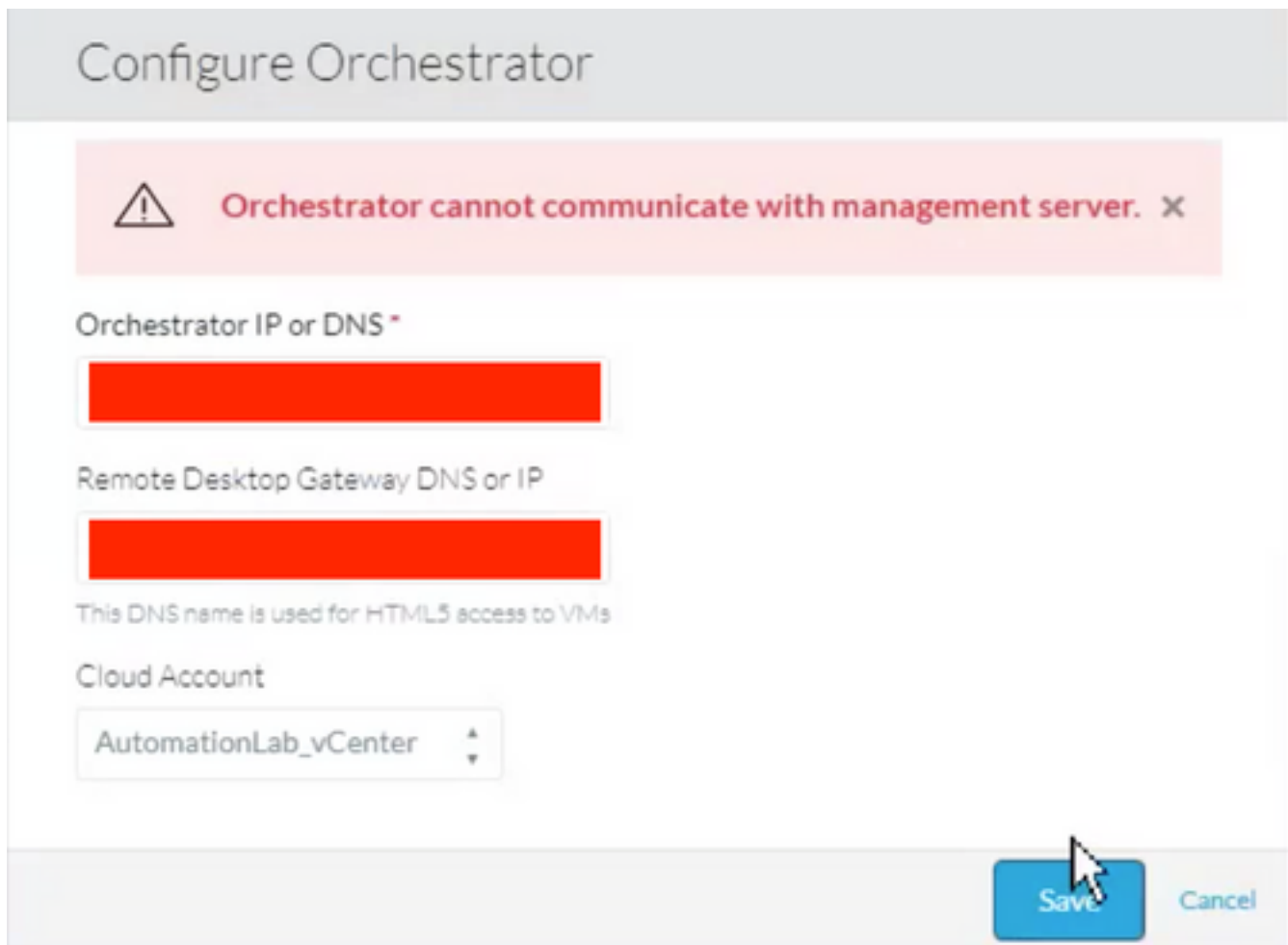
Cisco raadt kennis aan in:

- CloudCenter-apparaten
- CloudCenter-architectuur
- Linux O.S.
- CCM (CloudCenter Management)
- CCO (CloudCenter Orchestrator)
- Advanced Malware Protection (Advanced Message Queuing Protocol)


De informatie in dit document werd gemaakt van de apparaten in een specifieke privé-labomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de mogelijke impact van om het even welke opdracht begrijpt.

Probleem

Voedingsstroomuitval, onverwachte herstart of netwerkstoringen voor langere tijd kunnen ertoe leiden dat het CloudCenter-apparaat verdwijnt. Het is nodig de volgende controles uit te voeren om te weten dat apparaten correct zijn aangesloten bij het configureren van de orchestrator op de grafische gebruikersinterface van CloudCenter Manager (CCM GUI) gebruikers de fout zoals in de afbeelding kunnen krijgen.



Configure Orchestrator

 **Orchestrator cannot communicate with management server.** X

Orchestrator IP or DNS *

Remote Desktop Gateway DNS or IP

This DNS name is used for HTML5 access to VMs

Cloud Account

AutomationLab_vCenter

Save Cancel

Tijdens het controleren van de CCO-bestanden kan de volgende fout worden weergegeven:

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(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)
```

Oplossing

U dient de componenten van CloudCenter één voor één opnieuw te starten om de handdruk tussen de onderdelen te verfrissen

AMP

Stap 1. Meld u aan bij wortel

Stap 2. Start de AMQP-service opnieuw

Bij alle versies tot en met 4.8.1.2

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

Op versies vanaf 4.8.2

```
# systemctl restart rabbit
```

CCO

Stap 1. Meld u aan bij wortel

Stap 2. Start de CO-service opnieuw

Bij alle versies tot en met 4.8.1.2

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

Op versies vanaf 4.8.2

```
# systemctl restart cco
```

CCM

Stap 1. Meld u aan bij wortel

Stap 2. Start de CCM-service opnieuw

Bij alle versies tot en met 4.8.1.2

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

Op versies vanaf 4.8.2

```
# systemctl restart ccm
```

Verifiëren

Het is belangrijk dat alle apparaten correct worden aangesloten, omdat dit is nodig om in elk van de componenten van CloudCenter te controleren.

CCM

Stap 1. Meld u aan bij wortel

Stap 2. Controleer of tomcat (voorafgaand aan 4.8.2) of CCM-service (post 4.8.2) daadwerkelijk actief is

Bij alle versies tot en met 4.8.1.2

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

Op versies vanaf 4.8.2

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

Stap 3. Als telnet is geïnstalleerd, kan er een poging worden gedaan van CCO naar CCM, waardoor wordt begrepen dat communicatie mogelijk is

```
[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 '^]'.

```

Als er een fout optreedt, is er geen communicatie mogelijk. Dit moet worden vastgesteld.

Stap 4. Als u een orchestrator op CCM GUI wilt configureren, wordt een hostname gebruikt, zorg er dan voor dat de hostname in /etc/hosts-bestand aanwezig is

```
[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

```

AMP

Stap 1. Meld u aan bij wortel

Stap 2. Controleer of er een verbinding is die via de Advanced Malware Protection op elk van de bestaande CO's is aangesloten.

```
[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

```

In de vorige opdracht kunnen de verbindingen naar de CCO worden gezien in de lijst met de klant gebruiker (in dit geval is er slechts één CCO)

Bij een hoge beschikbaarheid (HA) en een AMQP onder een taakverdeler, wordt er één verbinding per CCO gezien die is aangesloten op de IP van de taakverdeling van de AMQP (In het volgende voorbeeld zijn er 2 CCO's)

```
[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

```

Indien dit niet het geval is, start dan het tomcatgua-proces (voorafgaand aan 4.8.2) of de konijnendienst (post 4.8.2) opnieuw op

CCO

Stap 1. Meld u aan bij wortel

Bij alle versies tot en met 4.8.1.2

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

Op versies vanaf 4.8.2

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

Stap 3. Controleer of verbindingen naar het CCM tot stand zijn gebracht. Ook deze zou in CLOSE_WAIT status moeten verschijnen (In dit geval is onze CCM op 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