疑難排解"；加入"；Docker叢集中的CPS-DRA VM狀態問題

目錄

簡介

本檔案介紹如何疑難排解Cisco Policy Suite (CPS)-Diameter Routing Agent (DRA)虛擬機器器 (VM)的狀態問題JOINING。

必要條件

需求

思科建議您瞭解以下主題：

- Linux

- CPS

注意：思科建議您必須具有對CPS DRA CLI的根使用者訪問許可權。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- CPS-DRA 22.2

- 整合運算系統(UCS)-B

本文中的資訊是根據特定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。

背景資訊

CPS Virtual Diameter Routing Agent (vDRA)是網路中的操作元件，透過使用路由演算法將消息引導到其預定的目標節點。

CPS vDRA的核心作用包括消息路由和隨後將響應傳輸到它們的原始源點。

CPS vDRA包含使用Docker引擎作為群集進行協調的一組虛擬機器(VM)，由不同的實體組成，即主虛擬機器、控制虛擬機器、指揮虛擬機器、分發虛擬機器和工作虛擬機器。

<#root>

admin@orchestrator[master-1]#

show docker engine

```
Fri Jul 14 09:36:18.635 UTC+00:00
MISSED
ID STATUS PINGS
--------------------------------
control-1 CONNECTED 0
control-2 CONNECTED 0
director-1 CONNECTED 0
director-2 CONNECTED 0
director-3 CONNECTED 0
director-4 CONNECTED 0
director-5 CONNECTED 0
director-6 CONNECTED 0
director-7 CONNECTED 0
director-8 CONNECTED 0
distributor-1 CONNECTED 0
distributor-2 CONNECTED 0
distributor-3 CONNECTED 0
distributor-4 CONNECTED 0
master-1 CONNECTED 0
worker-1 CONNECTED 0
worker-2 CONNECTED 0
worker-3 CONNECTED 0
admin@orchestrator[master-1]#
```

狀態-指示排程應用程式是否連線至docker引擎並在主機上執行。

未接ping -給定主機的連續未接ping數。

問題

有時CPS vDRA VM由於各種原因而停滯在JOINING狀態。

<#root>

admin@orchestrator[master-1]#

show docker engine

```
Fri Jul 14 09:36:18.635 UTC+00:00
MISSED
ID STATUS PINGS
--------------------------------
control-1 CONNECTED 0
control-2 CONNECTED 0
director-1 JOINING 57
director-2 JOINING 130
director-3 JOINING 131
director-4 JOINING 130
director-5 JOINING 30
director-6 JOINING 129
distributor-1 CONNECTED 0
distributor-2 CONNECTED 0
distributor-3 CONNECTED 0
distributor-4 CONNECTED 0
master-1 CONNECTED 0
worker-1 CONNECTED 0
worker-2 CONNECTED 0
worker-3 CONNECTED 0
admin@orchestrator[master-1]#
```

虛擬機器停滯在JOINING狀態的可能原因，

1. 無法從主虛擬機器訪問VM。

1.1.驗證受影響的虛擬機器上的編織連線狀態是否處於套筒模式。

注意： Weave Net會建立一個虛擬網路，將跨多部主機部署的Docker容器連線，並啟用自動探索。使用Weave Net，包含多個容器的基於攜帶型微服務的應用可以在任何地方運行：在一台主機、多台主機，甚至跨雲提供商和資料中心運行。應用使用網路的方式，就像所有容器都插入同一網路交換機一樣，無需配置埠對映、大使或鏈路。

CPS-DRA有兩種主要的編織連線狀態：fastdp和套筒。CPS-DRA集群內的首選項始終朝向在fastdp 狀態下具有編織連線。

<#root>

cps@director-1:~$

```
weave status connections
```

```
-> xx.xx.xx.xx:6783 established sleeve 4e:5f:58:99:d5:65(worker-1) mtu=1438
-> xx.xx.xx.xx:6783 established sleeve 76:33:17:3a:c7:ec(worker-2) mtu=1438
<- xx.xx.xx.xx:54751 established sleeve 76:3a:e9:9b:24:84(director-1) mtu=1438
-> xx.xx.xx.xx:6783 established sleeve 6e:62:58:a3:7a:a0(director-2) mtu=1438
-> xx.xx.xx.xx:6783 established sleeve de:89:d0:7d:b2:4e(director-3) mtu=1438
```

1.2.驗證受影響的虛擬機器上的journalctl 日誌中是否存在這些日誌消息。

2023-08-01T10:20:25.896+00:00 docker-engine Docker engine control-1 is unreachable
2023-08-01T10:20:25.897+00:00 docker-engine Docker engine control-2 is unreachable
2023-08-01T10:20:25.935+00:00 docker-engine Docker engine distributor-1 is unreachable
2023-08-01T10:20:25.969+00:00 docker-engine Docker engine worker-1 is unreachable

INFO: 2023/08/02 20:46:26.297275 overlay_switch ->[ee:87:68:44:fc:6a(worker-3)] fastdp timed out waiting for vxlan heartbeat
INFO: 2023/08/02 20:46:26.297307 overlay_switch ->[ee:87:68:44:fc:6a(worker-3)] using sleeve

2. VM磁碟空間耗盡。

2.1.驗證受影響虛擬機器上的磁碟空間使用情況，並確定磁碟空間使用率高的分割槽。

<#root>

 cps@control-2:~$

df -h

```
Filesystem Size Used Avail Use% Mounted on
udev 32G 0 32G 0% /dev
tmpfs 6.3G 660M 5.7G 11% /run
/dev/sda3 97G 97G 0 100% /
tmpfs 32G 0 32G 0% /dev/shm
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 32G 0 32G 0% /sys/fs/cgroup
/dev/sdb1 69G 4.7G 61G 8% /data
/dev/sda1 180M 65M 103M 39% /boot
/dev/sdb2 128G 97G 25G 80% /stats
overlay 97G 97G 0 100% /var/lib/docker/overlay2/63854e8173b46727e11de3751c450037b5f5565592b83112a3863fel
overlay 97G 97G 0 100% /var/lib/docker/overlay2/a86da2c7a289dc2b71359654c5160a9a8ae334960e78def78e6eece
overlay 97G 97G 0 100% /var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f3
overlay 97G 97G 0 100% /var/lib/docker/overlay2/49ee42311e82974707a6041d82e6c550004d1ce25349478bb974cc0
cps@control-2:~$
```

將CPS-DRA VM從加入狀態恢復的過程

方法1.

如果無法從主VM訪問VM，請使用此方法。

1. 驗證受影響虛擬機器上的編織連線狀態（如果為套筒模式）。

#weave connection status

<#root>

Sample output:

cps@director-1:~$

weave status connections

```
-> xx.xx.xx.xx:6783 established sleeve 4e:5f:58:99:d5:65(worker-1) mtu=1438
-> xx.xx.xx.xx:6783 established sleeve 76:33:17:3a:c7:ec(worker-2) mtu=1438
<- xx.xx.xx.xx:54751 established sleeve 76:3a:e9:9b:24:84(director-1) mtu=1438
-> xx.xx.xx.xx:6783 established sleeve 6e:62:58:a3:7a:a0(director-2) mtu=1438
-> xx.xx.xx.xx:6783 established sleeve de:89:d0:7d:b2:4e(director-3) mtu=1438
```

2. 在各自的虛擬機器上重新啟動編織。

#docker restart weave

3. 驗證編織連線狀態是否已移至fastdp狀態，且受影響的VM是否已移至CONNECTED狀態。

4. 如果VM仍然處於JOINING停滯狀態，請重新啟動那些影響VM的VM。

<#root>

#sudo reboot now

or

#init 6

5. 現在驗證受影響的VM是否已移至CONNECTED狀態。

<#root>

admin@orchestrator[master-1]#

show docker engine

```
Fri Jul 14 09:36:18.635 UTC+00:00
MISSED
```

```
ID STATUS PINGS
--------------------------------
control-1 CONNECTED 0
control-2 CONNECTED 0
director-1 CONNECTED 0
director-2 CONNECTED 0
director-3 CONNECTED 0
director-4 CONNECTED 0
distributor-1 CONNECTED 0
distributor-2 CONNECTED 0
distributor-3 CONNECTED 0
distributor-4 CONNECTED 0
master-1 CONNECTED 0
worker-1 CONNECTED 0
worker-2 CONNECTED 0
worker-3 CONNECTED 0
admin@orchestrator[master-1]#
```

6. 驗證vPAS是否啟動餐飲流量，並且所有容器均處於UP狀態（尤其是diameter endpoint），否則在drc01 VM中重新啟動容器 orchestrator-backup-a。

#docker restart orchestrator-backup-a

7. 現在，驗證vPAS是否開始處理流量。

方法2.

如果VM的磁碟空間耗盡。

1. 辨識耗用大量磁碟空間的目錄。

<#root>

root@control-2:/var/lib/docker/overlay2#

du -ah / --exclude=/proc | sort -r -h | head -n 10

176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7

2. 驗證耗用大量磁碟空間的檔案/記錄/傾印。

<#root>

root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7/diff#

ls -lrtha | grep G

```
total 88G
-rw------- 1 root root 1.1G Jul 12 18:10 core.22781
-rw------- 1 root root 1.2G Jul 12 18:12 core.24213
-rw------- 1 root root 1.2G Jul 12 18:12 core.24606
-rw------- 1 root root 1.1G Jul 12 18:12 core.24746
-rw------- 1 root root 1.1G Jul 12 18:13 core.25398
```

3. 確定在受影響的虛擬機器上運行的容器（尤其是不健康的容器）。

<#root>

admin@orchestrator[master-1]#

show docker service | exclude HEALTHY

```
Fri Jul 14 09:37:20.325 UTC+00:00
PENALTY
MODULE INSTANCE NAME VERSION ENGINE CONTAINER ID STATE BOX MESSAGE
----------------------------------------------------------------------------------
cc-monitor 103 cc-monitor 22.1.1-release control-2 cc-monitor-s103 STARTED true Pending health check
mongo-node 103 mongo-monitor 22.1.1-release control-2 mongo-monitor-s103 STARTED true Pending health ch
mongo-status 103 mongo-status 22.1.1-release control-2 mongo-status-s103 STARTED false -
policy-builder 103 policy-builder 22.1.1-release control-2 policy-builder-s103 STARTED true Pending hea
prometheus 103 prometheus-hi-res 22.1.1-release control-2 prometheus-hi-res-s103 STARTED true Pending h
prometheus 103 prometheus-planning 22.1.1-release control-2 prometheus-planning-s103 STARTED false -

admin@orchestrator[master-1]#
```

4. 確定觸發大量核心檔案的容器，並逐個檢查受影響虛擬機器上託管的每個容器。

<#root>

Sample output for container "cc-monitor-s103":

root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7/merged#

docker inspect cc-monitor-s103| grep /var/lib/docker/overlay2/| grep merged

```
"MergedDir": "/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7/
root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

5. 檢查您是否有權存取該特定貨櫃。

 #admin@orchestrator[master-0]# docker connect cc-monitor-s103

6. 如果無法訪問容器，請刪除大塊的核心檔案以釋放一些空間。

<#root>

root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7/diff#

rm -rf core*

7. 從受影響的VM登入到受影響的容器。

<#root>

#docker exec -it cc-monitor-s103 bash

8. 重新啟動容器中的app處理作業，以停止產生大量核心檔案。

<#root>

root@cc-monitor-s103:/#

supervisorctl status

app STARTING

```
app-logging-status RUNNING pid 30, uptime 21 days, 23:02:17
consul RUNNING pid 26, uptime 21 days, 23:02:17
consul-template RUNNING pid 27, uptime 21 days, 23:02:17
haproxy RUNNING pid 25, uptime 21 days, 23:02:17
root@cc-monitor-s103:/#
```

root@cc-monitor-s103:/# date;

supervisorctl restart app

```
Fri Jul 14 09:08:38 UTC 2023
app: stopped
app: started
root@cc-monitor-s103:/#
```

root@cc-monitor-s103:/#

supervisorctl status

```
app RUNNING pid 26569, uptime 0:00:01
app-logging-status RUNNING pid 30, uptime 21 days, 23:02:44
consul RUNNING pid 26, uptime 21 days, 23:02:44
consul-template RUNNING pid 27, uptime 21 days, 23:02:44
haproxy RUNNING pid 25, uptime 21 days, 23:02:44
root@cc-monitor-s103:/#
```

9. 如果步驟8.無法停止產生大量核心檔案，請重新啟動受影響的容器。

<#root>

```
   #
```

docker restart cc-monitor-s103

10. 檢查批次核心檔案生成是否已停止。

11. 要使受影響的虛擬機器恢復連線狀態，請登入orchestrator container並執行orchestration-engine 重新啟動。

<#root>

cps@master-1:~$ date;

docker exec -it orchestrator bash

```
Fri Jul 14 09:26:12 UTC 2023
root@orchestrator:/#
```

<#root>

root@orchestrator:/#

supervisorctl status

```
confd RUNNING pid 20, uptime 153 days, 23:33:33
consul RUNNING pid 19, uptime 153 days, 23:33:33
consul-template RUNNING pid 26, uptime 153 days, 23:33:33
haproxy RUNNING pid 17, uptime 153 days, 23:33:33
mongo RUNNING pid 22, uptime 153 days, 23:33:33
monitor-elastic-server RUNNING pid 55, uptime 153 days, 23:33:33
monitor-log-forward RUNNING pid 48, uptime 153 days, 23:33:33
orchestration-engine RUNNING pid 34, uptime 153 days, 23:33:33
orchestrator_back_up RUNNING pid 60, uptime 153 days, 23:33:33
remove-duplicate-containers RUNNING pid 21, uptime 153 days, 23:33:33
rolling-restart-mongo RUNNING pid 18, uptime 153 days, 23:33:33
simplehttp RUNNING pid 31, uptime 153 days, 23:33:33
root@orchestrator:/#
```

<#root>

root@orchestrator:/# date;

supervisorctl restart orchestration-engine

```
Fri Jul 14 09:26:39 UTC 2023
orchestration-engine: stopped
orchestration-engine: started
root@orchestrator:/#
```

12. 如果步驟11.對恢復虛擬機器沒有幫助，請在受影響的虛擬機器中進行引擎代理重新啟動。

<#root>

cps@control-2:~$

docker ps | grep engine

0b778fae2616 engine-proxy:latest "/w/w /usr/local/bin…" 5 months ago Up 3 weeks

engine-proxy-ddd7e7ec4a70859b53b24f3926ce6f01

<#root>

cps@control-2:~$

docker restart engine-proxy-ddd7e7ec4a70859b53b24f3926ce6f01

engine-proxy-ddd7e7ec4a70859b53b24f3926ce6f01
cps@control-2:~$

<#root>

cps@control-2:~$

docker ps | grep engine

0b778fae2616 engine-proxy:latest "/w/w /usr/local/bin…" 5 months ago Up 6 seconds engine-proxy-ddd7e7ec
cps@control-2:~$

13. 現在，驗證受影響的VM是否已移至CONNECTED狀態。

<#root>

admin@orchestrator[master-1]#

show docker engine

Fri Jul 14 09:36:18.635 UTC+00:00
ID STATUS MISSED PINGS
----
control-1 CONNECTED 0
control-2 CONNECTED 0
director-1 CONNECTED 0
director-2 CONNECTED 0
director-3 CONNECTED 0
director-4 CONNECTED 0
distributor-1 CONNECTED 0
distributor-2 CONNECTED 0

```
distributor-3 CONNECTED 0
distributor-4 CONNECTED 0
master-1 CONNECTED 0
worker-1 CONNECTED 0
worker-2 CONNECTED 0
worker-3 CONNECTED 0
admin@orchestrator[master-1]#
```