

Risoluzione dei problemi di "JOINING" stato delle VM CPS-DRA nel cluster Docker

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Problema](#)

[Procedura per il ripristino delle VM CPS-DRA dallo stato di JOIN](#)

Introduzione

In questo documento viene descritto come risolvere i problemi di stato con la macchina virtuale (VM) Cisco Policy Suite (CPS)-Diameter Routing Agent (DRA)JOINING.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Linux
- CPS



Nota: Cisco consiglia di disporre dell'accesso privilegiato root alla CLI di CPS DRA.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- CPS-DRA 2.2
- Unified Computing System (UCS)-B

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Il vDRA (Virtual Diameter Routing Agent) di CPS funge da componente operativo all'interno di una rete, guidando i messaggi verso i nodi di destinazione previsti attraverso l'utilizzo di algoritmi di routing.

Il ruolo centrale di CPS vDRA implica l'instradamento dei messaggi e la successiva trasmissione delle risposte ai loro punti di origine originali.

Composto da una raccolta di macchine virtuali (VM) orchestrate come cluster utilizzando i motori Docker, CPS vDRA è costituito da entità distinte, ovvero VM Master, Control, Director, Distributore e Worker.

<#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]#
```

Stato: indica se l'applicazione di pianificazione è connessa al motore docker ed è in esecuzione su un host.

Ping mancati: il numero di ping mancati consecutivi per un determinato host.

Problema

A volte la VM vDRA CPS rimane bloccata nello stato di JOIN per vari motivi.

<#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]#
```

Possibili motivi per cui la VM rimane bloccata nello stato in cui si trova JOINING,

1. La VM non è raggiungibile dalla VM master.

1.1. Verificare se lo stato delle connessioni a trama sulla VM interessata è in modalità sleeve.



Nota: Weave Net crea una rete virtuale che connette i contenitori Docker distribuiti su più host e ne consente l'individuazione automatica. Con Weave Net, le applicazioni portatili basate su microservizi, costituite da più contenitori, possono essere eseguite ovunque: su un host, su più host o anche su provider di cloud e centri dati. Le applicazioni utilizzano la rete come se tutti i contenitori fossero collegati allo stesso switch di rete, senza configurare mapping delle porte, ambasciatori o collegamenti.

CPS-DRA ha due stati principali di connessioni a trama: fastdp e sleeve. La preferenza all'interno del cluster CPS-DRA è costantemente orientata verso l'avere connessioni a trama nello stato fastdp .

```
<#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. Verificare se questi messaggi di registro sono presenti nel journalctl registro della macchina virtuale interessata.

```
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. Lo spazio su disco della VM è esaurito.

2.1. Verificare l'utilizzo dello spazio su disco nella macchina virtuale interessata e identificare la partizione con un utilizzo elevato dello spazio su disco.

<#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/overlay2/63854e8173b46727e11de3751c450037b5f5565592b83112a3863fe
overlay 97G 97G 0 100% /var/lib/docker/overlay2/a86da2c7a289dc2b71359654c5160a9a8ae334960e78def78e6eeca
overlay 97G 97G 0 100% /var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f3
overlay 97G 97G 0 100% /var/lib/docker/overlay2/49ee42311e82974707a6041d82e6c550004d1ce25349478bb974cc0
cps@control-2:~$
```

Procedura per il ripristino delle VM CPS-DRA dallo stato di JOIN

Approccio 1.

Se la VM non è raggiungibile dalla VM master, utilizzare questo approccio.

1. Verificare lo stato della connessione alla trama sulle macchine virtuali interessate, se è in modalità sleeve.

```
#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. Riavviare la tessitura sulle rispettive VM.

```
#docker restart weave
```

3. Verificare se lo stato della connessione a trama è passato allo stato attuale e se la macchina virtuale interessata è passata allo stato attualefastdp CONNECTED.

4. Se le VM sono ancora bloccate nello JOINING stato, riavviare le VM interessate.

```
<#root>
```

```
#sudo reboot now
```

```
or
```

```
#init 6
```

5. Verificare ora se la macchina virtuale interessata è passata allo statoCONNECTED.

```
<#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. Verificare se il vPAS avvia il traffico di catering e tutti i contenitori sono attivi (in particolare l'endpoint di diametro), altrimenti riavviare il contenitore nella macchina virtuale drc01orchestrator-backup-a.

```
#docker restart orchestrator-backup-a
```

7. Verificare se vPAS ha iniziato a elaborare il traffico.

Metodo 2.

Se lo spazio su disco della macchina virtuale si esaurisce.

1. Identificare la directory che utilizza molto spazio su disco.

```
<#root>
```

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

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

```
176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

2. Verificare i file/log/dump che occupano molto spazio su disco.

```
<#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. Identificare i contenitori in esecuzione sulla macchina virtuale interessata (in particolare i contenitori non integri).

```
<#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 check
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 health check
prometheus 103 prometheus-hi-res 22.1.1-release control-2 prometheus-hi-res-s103 STARTED true Pending health check
prometheus 103 prometheus-planning 22.1.1-release control-2 prometheus-planning-s103 STARTED false -
```

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

4. Identificare il contenitore che attiva file di base ingombranti, per fare in modo che ispezioni ogni contenitore ospitato sulla VM interessata, uno per uno.

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

5. Verificare se si dispone o meno dell'accesso a quel particolare container.

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

6. Se non puoi accedere al contenitore, rimuovi i file di base più voluminosi per liberare spazio.

```
<#root>
```

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

```
rm -rf core*
```

7. Accedere al contenitore interessato dalla macchina virtuale interessata.

```
<#root>
```

```
#docker exec -it cc-monitor-s103 bash
```

8. Riavviare il processo nelapp contenitore per interrompere la generazione di file di base voluminosi.

```
<#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. Se il passo 8. non consente di interrompere la generazione di file di base in blocco, riavviare il contenitore interessato.

```
<#root>
```

```
#
```

```
docker restart cc-monitor-s103
```

10. Verificare se la generazione di file di base in blocco è stata interrotta.

11. Per riportare la macchina virtuale interessata allo stato CONNESSO, accedere al orchestrator contenitore ed eseguire il orchestration-engine riavvio.

```
<#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. Se il passaggio 11. non consente di ripristinare la macchina virtuale, avviare il riavvio del proxy del motore nella macchina virtuale interessata.

```
<#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. Verificare ora se la macchina virtuale interessata è passata allo statoCONNECTED.

```
<#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]#
```

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).