

Solucionar problemas de estado de "JOINING" de VMs CPS-DRA no cluster de encaixe

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Problema](#)

[Procedimento para recuperar VMs CPS-DRA do estado JOINING](#)

Introdução

Este documento descreve como solucionar problemas de estado JOINING com a Máquina virtual (VM) do Cisco Policy Suite (CPS)-Diameter Routing Agent (DRA).

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Linux
- CPS



Observação: a Cisco recomenda que você tenha acesso de raiz privilegiado à CLI DRA do CPS.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- CPS-DRA 22.2
- Unified Computing System (UCS)-B

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

O Agente de Roteamento de Diâmetro Virtual (vDRA - Virtual Diameter Routing Agent) do CPS serve como o componente operacional dentro de uma rede, guiando as mensagens para os nós de destino pretendidos através da utilização de algoritmos de roteamento.

O papel central do CPS vDRA envolve o encaminhamento de mensagens e a transmissão subsequente de respostas aos seus pontos de origem originais.

Compreendendo uma coleção de máquinas virtuais (VMs) orquestradas como um cluster usando mecanismos Docker, o CPS vDRA consiste em entidades distintas, ou seja, MVs Master, Control, Diretor, Distribuidor 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]#
```

Status - Indica se a aplicação de agendamento está conectada ao mecanismo docker e é executada em um host.

Pings perdidos - O número de pings perdidos consecutivos para um determinado host.

Problema

Às vezes, a VM vDRA do CPS fica presa no estado JOINING devido a vários motivos.

```
<#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]#
```

Os possíveis motivos para a VM ficar presa no estado JOINING,

1. A VM não pode ser alcançada pela VM mestre.

- 1.1. Verifique se o status das conexões de tecelagem na VM afetada está no modo de encarte.



Observação: a Weave Net cria uma rede virtual que conecta contêineres Docker implantados em vários hosts e permite sua descoberta automática. Com a Weave Net, os aplicativos portáteis baseados em microsserviços que consistem em vários contêineres podem ser executados em qualquer lugar: em um host, vários hosts ou até mesmo em provedores de nuvem e data centers. Os aplicativos usam a rede como se todos os contêineres estivessem conectados ao mesmo switch de rede, sem configurar mapeamentos de porta, embaixadores ou links.

O CPS-DRA tem dois estados principais de conexões de tecelagem: fastdp e manga. A preferência dentro do cluster CPS-DRA é consistentemente orientada para ter conexões de tecelagem no fastdp estado.

<#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. Verifique se essas mensagens de log estão presentes no journalctl log na VM afetada.

```
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. O espaço em disco da VM se esgota.

2.1. Verifique o uso de espaço em disco na VM afetada e identifique a partição com alto uso de espaço em 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/a86da2c7a289dc2b71359654c5160a9a8ae334960e78def78e6eece
overlay 97G 97G 0 100% /var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f3
overlay 97G 97G 0 100% /var/lib/docker/overlay2/49ee42311e82974707a6041d82e6c550004d1ce25349478bb974cc0
cps@control-2:~$
```

Procedimento para recuperar VMs CPS-DRA do estado JOINING

Abordagem 1.

Se a VM não for alcançável a partir da VM principal, use essa abordagem.

1. Verifique o status da conexão de tecelagem nas VM/s afetadas, se estiver no modo de encarte.

```
#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. Reinicie o weave nas respectivas VMs.

```
#docker restart weave
```

3. Verifique se o status da conexão de tecelagem foi movido para o fastdp estado e se a VM afetada foi movida para o estadoCONNECTED.

4. Se as VMs ainda permanecerem no JOINING estado, reinicialize-as para causar impacto nas VMs.

```
<#root>
```

```
#sudo reboot now
```

```
or
```

```
#init 6
```

5. Agora, verifique se a VM afetada foi movida para o estado "CONNECTED".

```
<#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. Verifique se o vPAS inicia o tráfego de fornecimento e se todos os contêineres estão ATIVADOS (especialmente o endpoint de diâmetro); caso contrário, reinicie orchestrator-backup-a o contêiner na VM drc01.

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

7. Agora, verifique se o vPAS começou a processar o tráfego.

Abordagem 2.

Se o espaço em disco da VM se esgotar.

1. Identifique o diretório que consome espaço em disco.

```
<#root>
```

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

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

```
176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

2. Verifique os arquivos/logs/dumps que consomem muito espaço em 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. Identifique os contêineres executados na VM afetada (especialmente contêineres não íntegros).

```
<#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. Identifique o contêiner que aciona os arquivos principais volumosos para inspecionar cada contêiner hospedado na VM afetada, um por um.

```
<#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. Verifique se você tem acesso a esse contêiner específico ou não.

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

6. Se você não puder acessar esse contêiner, remova os arquivos principais volumosos para liberar espaço.

```
<#root>
```

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

```
rm -rf core*
```

7. Faça login no contêiner afetado a partir da VM afetada.

```
<#root>
```

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

8. Reinicie o processoapp no contêiner para interromper a geração de arquivos principais volumosos.

```
<#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 a Etapa 8. não ajudar a interromper a geração de arquivos principais em massa, reinicie o contêiner afetado.

```
<#root>
```

```
#
```

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

10. Verifique se a geração do arquivo principal em massa foi interrompida.

11. Para trazer a VM afetada de volta ao estado CONNECTED, efetue login no orchestrator contêiner e execute orchestration-engine reinicialização.

```
<#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 a Etapa 11. não ajudar a restaurar a VM, vá para reinicialização do engine-proxy na VM afetada.

```
<#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. Agora, verifique se a VM afetada foi movida para o estado "CONNECTED".

```
<#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]#
```

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.