

Docker 클러스터에 있는 CPS-DRA VM의 "JOINING" 상태 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제](#)

[결합 상태에서 CPS-DRA VM을 복구하는 절차](#)

소개

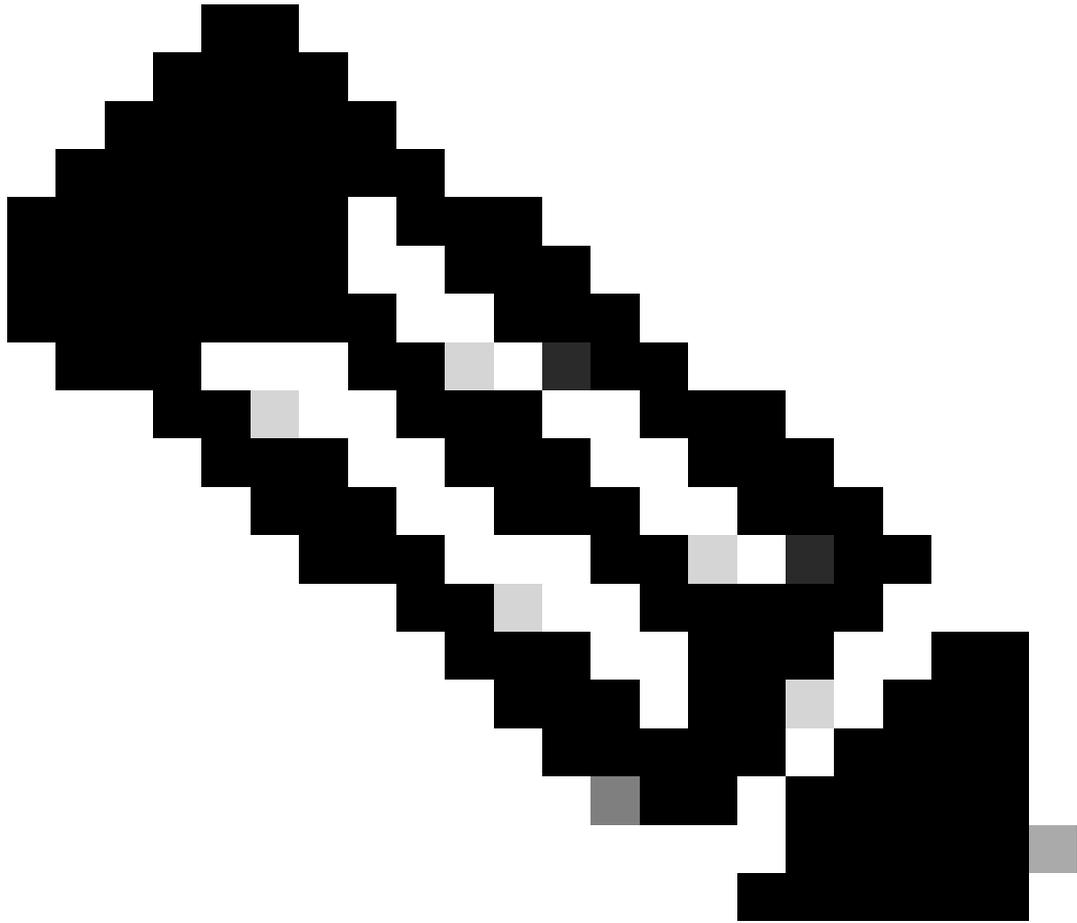
이 문서에서는 Cisco CPS(Policy Suite)-DRA(Diameter Routing Agent) VM(Virtual Machine)의 상태 문제를 해결하는 방법에 대해 설명합니다JOINING.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Linux
- CPS



참고: CPS DRA CLI에 대한 루트 액세스 권한이 있어야 합니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

CPS vDRA(Virtual Diameter Routing Agent)는 네트워크 내에서 운영 구성 요소 역할을 하며 라우팅 알고리즘을 활용하여 메시지를 원하는 대상 노드로 안내합니다.

CPS vDRA의 중심 역할은 메시지 라우팅 및 원래 발신지로의 후속 응답 전송을 포함합니다.

CPS vDRA는 Docker 엔진을 사용하여 클러스터로 오케스트레이션된 VM(가상 머신) 컬렉션으로 구성되며, Master, Control, Director, Distributor, Worker VM과 같은 고유한 엔티티로 구성됩니다.

<#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(상태) - 스케줄링 애플리케이션이 Docker 엔진에 연결되어 호스트에서 실행되는지 여부를 나타냅니다.

Missed ping(누락된 ping) - 지정된 호스트에 대한 연속 누락된 ping 수입니다.

문제

때로는 CPS vDRA VM이 여러 가지 이유로 인해 JOINING State에 갇히기도 합니다.

<#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]#
```

VM이 이 상태로 유지되는 JOINING 가능한 이유,

1. 마스터 VM에서 VM에 연결할 수 없습니다.

1.1. 영향받는 VM의 weave 연결 상태가 슬리브 모드인지 확인합니다.



참고: Weave Net은 여러 호스트에 구축된 Docker 컨테이너를 연결하고 해당 컨테이너의 자동 검색을 활성화하는 가상 네트워크를 생성합니다. Weave Net을 사용하면 여러 컨테이너로 구성된 휴대용 마이크로서비스 기반 애플리케이션을 하나의 호스트, 여러 호스트, 심지어 클라우드 제공업체와 데이터 센터 전체에서 실행할 수 있습니다. 애플리케이션은 포트 매핑, 대사 또는 링크를 구성하지 않고 마치 컨테이너가 모두 동일한 네트워크 스위치에 연결된 것처럼 네트워크를 사용합니다.

CPS-DRA는 fastdp와 sleeve라는 두 가지 주요 직조 연결 상태가 있습니다. CPS-DRA 클러스터 내의 기본 설정은 지속적으로 해당 상태에서 직조 연결을 하는 방향으로 fastdp 향합니다.

<#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. 이러한 로그 메시지가 영향을 받는 VM의 journalctl 로그에 표시되는지 확인합니다.

```
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 디스크 공간이 부족합니다.

2.1. 영향을 받는 VM의 디스크 공간 사용량을 확인하고 디스크 공간 사용량이 많은 파티션을 식별합니다.

```
<#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:~$
```

결합 상태에서 CPS-DRA VM을 복구하는 절차

접근 방식 1.

마스터 VM에서 VM에 연결할 수 없는 경우 이 방법을 사용합니다.

1. 슬리브 모드인 경우 영향을 받는 VM에 대한 weave 연결 상태를 확인합니다.

```
#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. 각 VM에서 weave를 다시 시작합니다.

```
#docker restart weave
```

3. weave 연결 상태가 fastdp 상태로 변경되었고 영향받는 VM이 상태로 변경되었는지 CONNECTED 확인합니다.

4. VM이 여전히 JOINING 고착 상태인 경우 영향을 주는 VM을 재부팅합니다.

```
<#root>
```

```
#sudo reboot now
```

```
or
```

```
#init 6
```

5. 이제 영향받는 VM이 상태로 이동했는지 확인합니다CONNECTED.

```
<#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. vPAS가 케이터링 트래픽을 시작하며 모든 컨테이너가 UP(특히 직경 엔드포인트)인지, 그렇지 않으면 drc01 VM에서 컨테이너를 다시 시작하는지 orchestrator-backup-a 확인합니다.

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

7. 이제 vPAS가 트래픽 처리를 시작했는지 확인합니다.

접근 2.

VM의 디스크 공간이 소진되는 경우

1. 높은 디스크 공간을 사용하는 디렉토리를 식별합니다.

<#root>

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

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

```
176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

2. 대용량 디스크 공간을 사용하는 파일/로그/덤프를 확인합니다.

<#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. 영향을 받는 VM에서 실행되는 컨테이너(특히 비정상 컨테이너)를 식별합니다.

<#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. 대용량 코어 파일을 트리거하는 컨테이너를 식별하여 영향을 받는 VM에 호스팅된 각 컨테이너를 하나씩 검사합니다.

<#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. 특정 컨테이너에 대한 액세스 권한이 있는지 확인합니다.

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

6. 컨테이너에 액세스할 수 없는 경우, 부피가 큰 핵심 파일을 제거하여 공간을 확보하십시오.

<#root>

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

```
rm -rf core*
```

7. 영향을 받는 VM에서 영향을 받는 컨테이너에 로그인합니다.

```
<#root>
```

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

8. 대용량 app 코어 파일의 생성을 중지하려면 컨테이너에서 프로세스를 다시 시작합니다.

```
<#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. 8단계가 벌크 코어 파일의 생성을 중지하는 데 도움이 되지 않으면 영향을 받은 컨테이너를 다시 시작합니다.

```
<#root>
```

```
#
```

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

10. 벌크 코어 파일 생성이 중지되었는지 확인합니다.

11. 영향을 받은 VM을 다시 CONNECTED 상태로 만들려면 컨테이너에 로그인하고 orchestrator 다시 orchestration-engine 시작합니다.

```
<#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. 11단계에서 VM을 복원하는 데 도움이 되지 않는 경우 영향받는 VM에서 엔진 프록시 재시작으로 이동합니다.

```
<#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. 이제 영향받는 VM이 상태로 이동했는지 CONNECTED 확인합니다.

```
<#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]#
```

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.