

# Troubleshoot "JOINING" State Issues of CPS-DRA VMs in Docker Cluster

## Contents

---

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Problem](#)

[Procedure to Recover CPS-DRA VMs from JOINING State](#)

---

## Introduction

This document describes how to troubleshoot JOINING state issues with Cisco Policy Suite (CPS)-Diameter Routing Agent (DRA) Virtual Machine (VM).

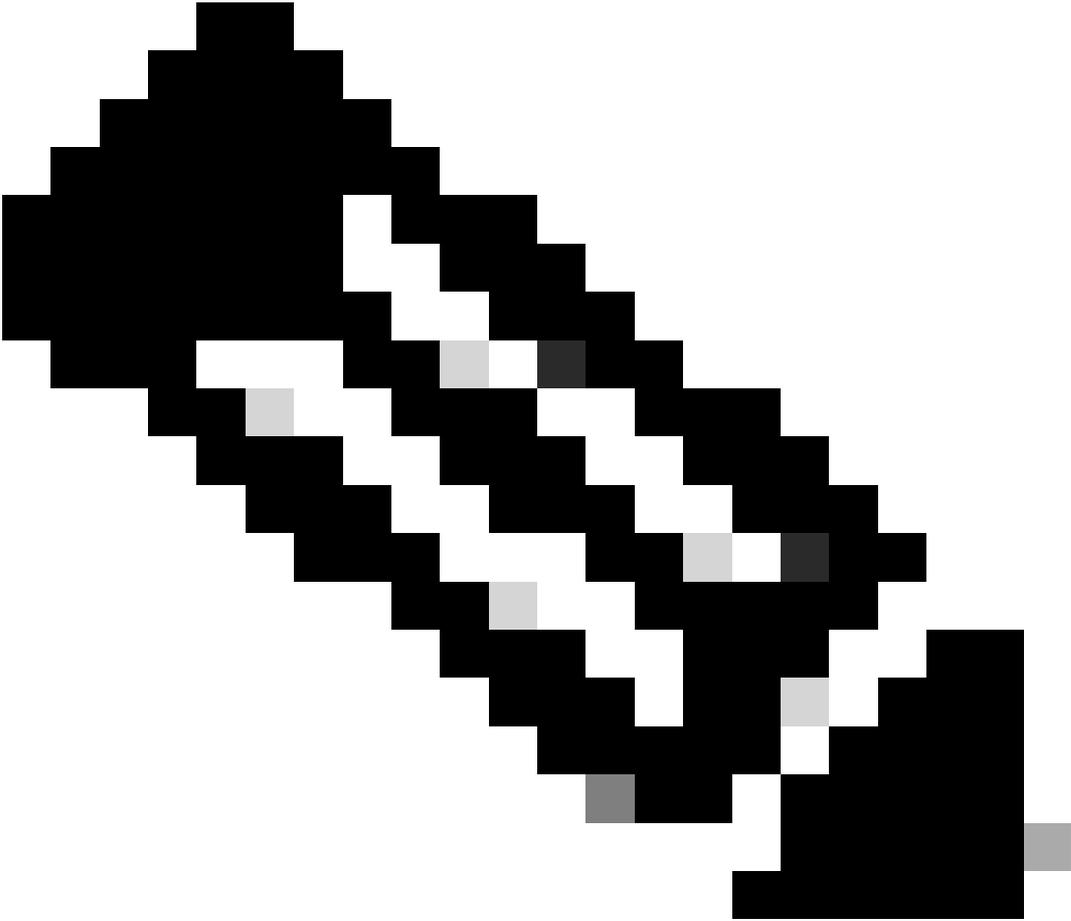
## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Linux
- CPS

---



**Note:** Cisco recommends that you must have privilege root access to CPS DRA CLI.

---

## Components Used

The information in this document is based on these software and hardware versions:

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

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## Background Information

The CPS Virtual Diameter Routing Agent (vDRA) serves as the operational component within a network, guiding messages to their intended destination nodes through the utilization of routing algorithms.

The central role of CPS vDRA involves message routing and the subsequent transmission of responses to

their original points of origin.

Comprising a collection of virtual machines (VMs) orchestrated as a cluster using Docker engines, CPS vDRA consists of distinct entities, namely Master, Control, Director, Distributor, and Worker VMs.

```
<#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 - Indicates if the scheduling application is connected to the docker engine and runs on a host.

Missed pings - The number of consecutive missed pings for a given host.

## Problem

Sometimes CPS vDRA VM gets stuck in JOINING State due to various reasons.

```
<#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
```

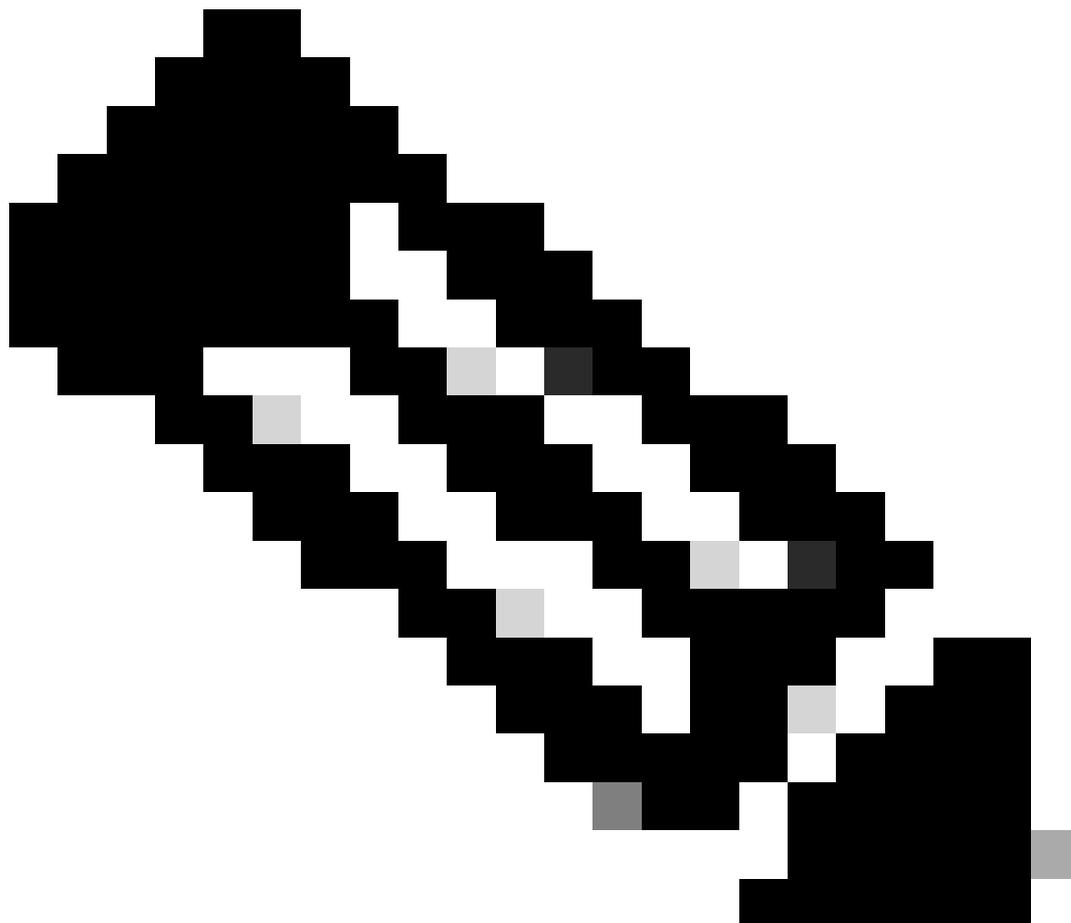
```
director-1 JOINING 57
director-2 JOINING 130
director-3 JOINING 131
director-4 JOINING 130
director-5 JOINING 30
director-6 JOINING 129
distributor-1 CONNECTED 0
distributor-2 CONNECTED 0
distributor-3 CONNECTED 0
distributor-4 CONNECTED 0
master-1 CONNECTED 0
worker-1 CONNECTED 0
worker-2 CONNECTED 0
worker-3 CONNECTED 0
admin@orchestrator[master-1]#
```

The possible reasons for VM to get stuck in the JOINING state,

1. VM is not reachable from the master VM.

- 1.1. Verify if the weave connections status on the impacted VM are in sleeve mode.

---



---

**Note:** Weave Net creates a virtual network that connects Docker containers deployed across multiple hosts and enables their automatic discovery. With Weave Net, portable microservices-based applications consisting of multiple containers can run anywhere: on one host, multiple hosts or even across cloud providers and data centers. Applications use the network just as if the containers were all plugged into the same network switch, without configuring port mappings, ambassadors or links.

---

CPS-DRA has two primary states of weave connections: `fastdp` and `sleeve`. The preference within the CPS-DRA cluster is consistently oriented towards having weave connections in the `fastdp` state.

```
<#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. Verify if these log messages are present in `journalctl` log on the impacted VM.

```
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
INFO: 2023/08/02 20:46:26.297307 overlay_switch ->[ee:87:68:44:fc:6a(worker-3)] using sleeve
```

2. VM Disk space gets exhausted.

2.1. Verify the disk space usage on the impacted VM and identify the partition with high disk space usage.

```
<#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:~$
```

## Procedure to Recover CPS-DRA VMs from JOINING State

Approach 1.

If the VM is not reachable from the master VM, use this approach.

1. Verify weave connection status on impacted VM/s, if it is sleeve mode.

```
#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. Restart weave on respective VMs.

```
#docker restart weave
```

3. Verify if the weave connection status has moved to `fastdp` state and the impacted VM has moved to `CONNECTED` state.

4. If VMs are still stuck in `JOINING` state, reboot those impact VMs.

```
<#root>
```

```
#sudo reboot now
```

```
or
```

```
#init 6
```

5. Now verify if the impacted VM has moved to CONNECTED state.

```
<#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. Verify if the vPAS starts catering traffic and all containers are UP (especially diameter endpoint), else restart orchestrator-backup-a container in drc01 VM.

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

7. Now, verify if vPAS began to process the traffic.

Approach 2.

If the Disk space of the VM gets exhausted.

1. Identify the directory that consumes high disk space.

```
<#root>
```

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

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

```
176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

2. Verify the files/logs/dumps that consume huge disk space.

```
<#root>
```

```
root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b
```

```
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. Identify the containers that run on impacted VM (especially unhealthy containers).

```
<#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. Identify the container that triggers bulky core files, to do so inspect each container hosted on the impacted VM, one by one.

```
<#root>
```

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

```
root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b
```

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

```
"MergedDir": "/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

```
root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b
```

5. Check if you have access to that particular container or not.

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

6. If you cannot access that container, remove bulky core files to free some space.

```
<#root>
```

```
root@control-2:/var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b
```

```
rm -rf core*
```

7. Login into the impacted container from the impacted VM.

```
<#root>
```

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

8. Restart the app process in the container to stop generation of bulky core files.

```
<#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. If Step 8. does not help to stop the generation of bulk core files, restart the impacted container.

```
<#root>
```

```
#
```

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

10. Check if bulk core file generation has stopped.

11. To bring the impacted VM back to the CONNECTED state, log in to orchestrator container and perform orchestration-engine restart.

```
<#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. If Step 11. does not help to restore the VM, go for engine-poxy restart in the impacted VM.

```
<#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. Now, verify if the impacted VM has moved to CONNECTED state.

```
<#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]#
```