排除Docker群集中CPS-DRA VM的状态问题(&Q)

目录

简介

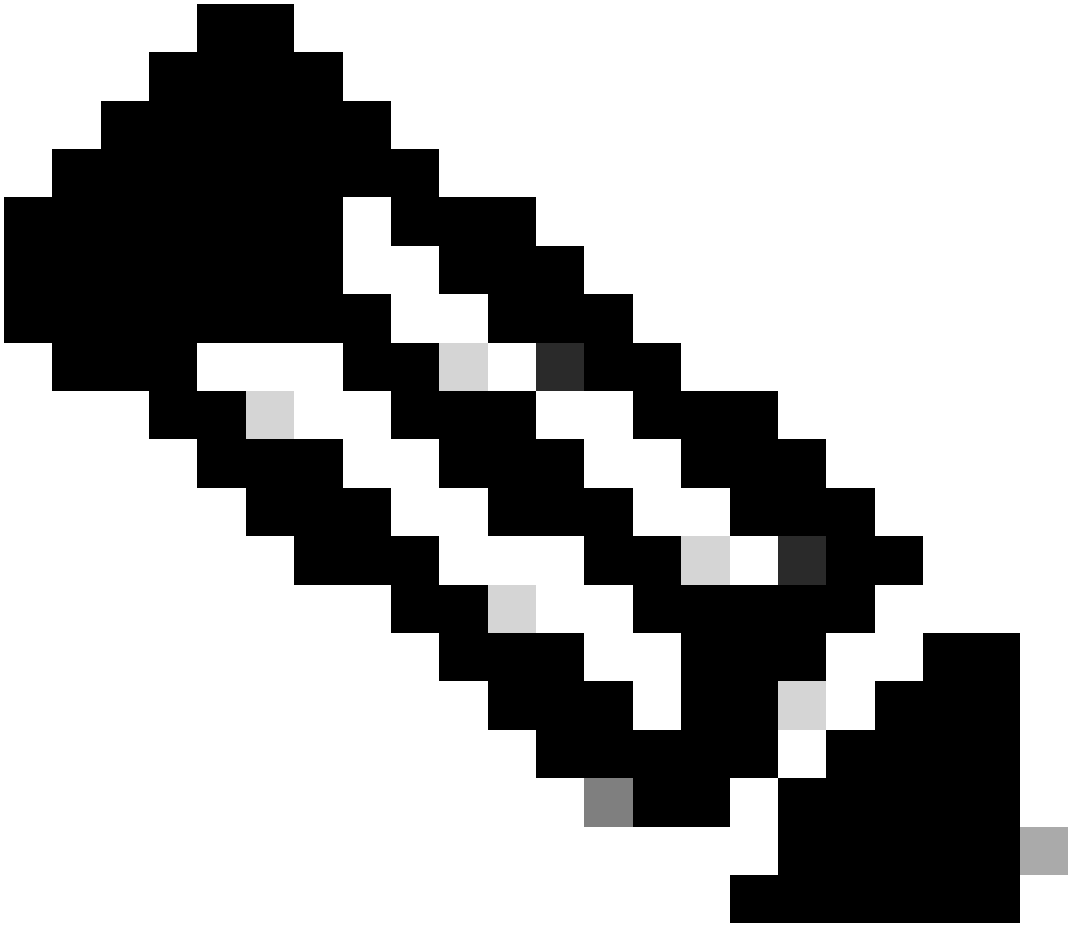
本文档介绍如何排除Cisco Policy Suite (CPS)-Diameter Routing Agent (DRA)虚拟机(VM)的状态问题JOINING。

先决条件

要求

Cisco 建议您了解以下主题：

- Linux
- CPS



注意：思科建议您必须具有对CPS DRA CLI的超级用户访问权限。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- CPS-DRA 22.2
- 统一计算系统(UCS)-B

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

CPS Virtual Diameter Routing Agent (vDRA)充当网络中的操作组件，通过使用路由算法将消息引导至其预定目的节点。

CPS vDRA的核心作用是消息路由和随后将响应传输到原始源点。

CPS vDRA包含使用Docker引擎作为集群协调的虚拟机(VM)集合，由不同的实体组成，即主虚拟机、控制虚拟机、导向虚拟机、分发服务器虚拟机和工作服务器VM。

<#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]#
```

状态-指示计划应用程序是否连接到Docker引擎并在主机上运行。

错过的ping -给定主机连续错过的ping数。

问题

有时，CPS vDRA VM由于各种原因而停滞在JOINING状态。

<#root>

admin@orchestrator[master-1]#

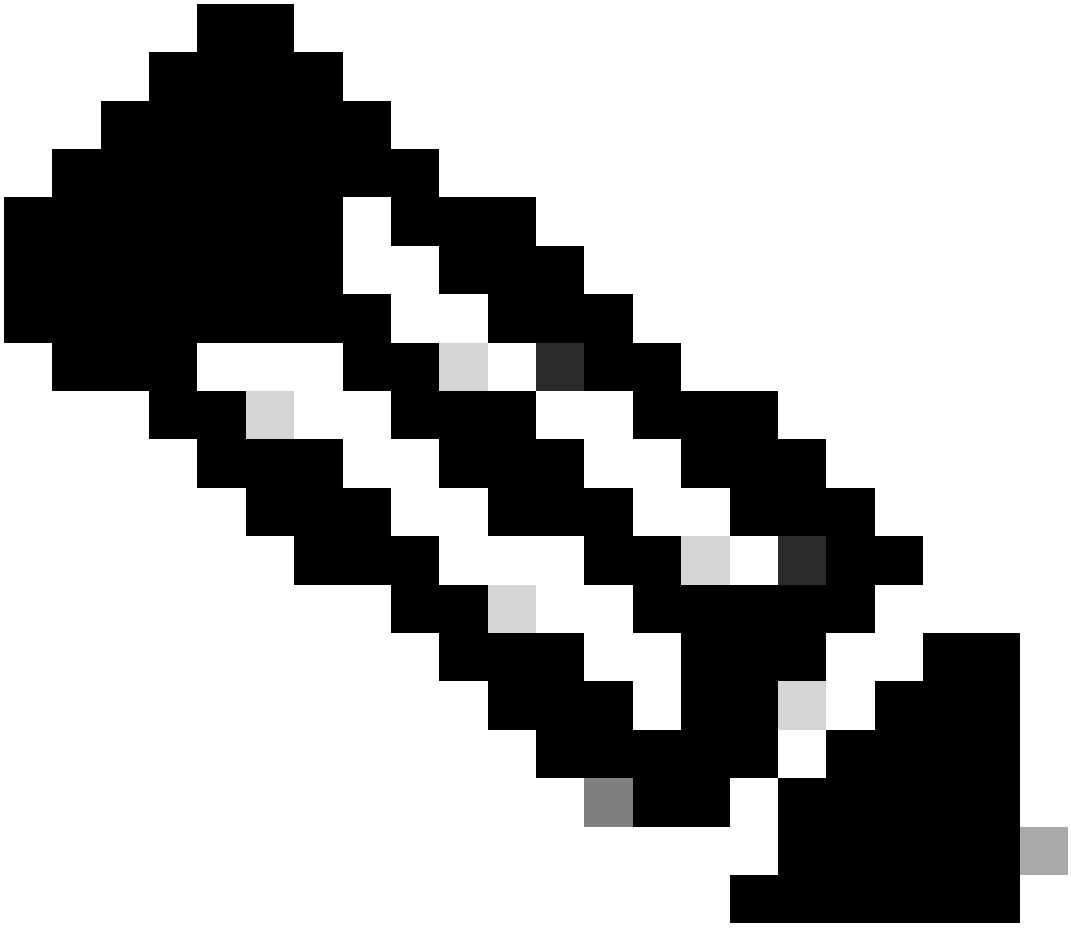
show docker engine

```
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]#
```

导致VM停滞在JOINING状态的可能原因

1. 无法从主VM访问虚拟机。

1.1.验证受影响虚拟机上的编织连接状态是否为套筒模式。



注意：Weave Net会创建一个虚拟网络，用于连接部署在多个主机上的Docker容器并启用其自动发现。借助Weave Net，由多个容器组成的便携式微服务应用可以在任何地方运行：一台主机、多台主机，甚至跨云提供商和数据中心运行。应用程序使用网络的方式与将容器全部插入同一网络交换机一样，无需配置端口映射、大使或链路。

CPS-DRA有两种主要的编织连接状态：fastdp和套筒。CPS-DRA集群中的首选项始终倾向于使编织连接处于fastdp 状态。

<#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.验证受影响的VM上的journalctl 日志中是否存在这些日志消息。

```
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磁盘空间耗尽。

2.1.验证受影响虚拟机上的磁盘空间使用情况，并确定磁盘空间使用率较高的分区。

<#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/a86da2c7a289dc2b71359654c5160a9a8ae334960e78def78e6ece
overlay 97G 97G 0 100% /var/lib/docker/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f3
overlay 97G 97G 0 100% /var/lib/docker/overlay2/49ee42311e82974707a6041d82e6c550004d1ce25349478bb974cc0
cps@control-2:~$
```

将CPS-DRA VM从加入状态恢复的过程

方法1.

如果无法从主VM访问VM，请使用此方法。

1. 验证受影响虚拟机上的编织连接状态（如果为套筒模式）。

#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. 在各自的虚拟机上重新启动编织。

#docker restart weave

3. 验证编织连接状态是否已移至fastdp状态，受影响虚拟机是否已移至CONNECTED状态。

4. 如果VM仍停滞在JOINING状态，请重新启动那些受影响的VM。

<#root>

#sudo reboot now

or

#init 6

5. 现在验证受影响的VM是否已变为CONNECTED状态。

<#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. 验证vPAS是否启动餐饮流量，并且所有容器均处于UP状态（尤其是diameter终端），否则在drc01 VM中重新启动orchestrator-backup-a容器。

#docker restart orchestrator-backup-a

7. 现在，验证vPAS是否开始处理流量。

方法2.

如果VM的磁盘空间耗尽。

1. 确定占用高磁盘空间的目录。

<#root>

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

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

176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7

2. 验证占用大量磁盘空间的文件/日志/转储。

<#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. 确定在受影响虚拟机上运行的容器（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. 确定触发大容量核心文件的容器，以逐个检查受影响虚拟机上托管的每个容器。

<#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/overlay2/9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b
```

5. 检查您是否有权访问该特定容器。

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

6. 如果无法访问容器，请删除大块的核心文件以释放一些空间。

<#root>

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

rm -rf core*

7. 从受影响的VM登录受影响的容器。

<#root>

#docker exec -it cc-monitor-s103 bash

8. 重新启动容器中的app进程，以停止生成大容量核心文件。

<#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. 如果步骤8.无助于停止生成批量核心文件，请重新启动受影响的容器。

<#root>

#

docker restart cc-monitor-s103

10. 检查批量核心文件生成是否已停止。

11. 要将受影响的VM恢复为“已连接”状态，请登录orchestrator容器并执行重orchestration-engine 启。

<#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. 如果步骤11.对恢复VM没有帮助，请在受影响的VM中执行引擎代理重新启动。

<#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. 现在，验证受影响的VM是否已变为CONNECTED状态。

<#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]#
```

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。