

Solucionar problemas de "JOINING" de estado de VM CPS-DRA en clúster de Docker

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Problema](#)

[Procedimiento para recuperar máquinas virtuales CPS-DRA del estado JOINING](#)

Introducción

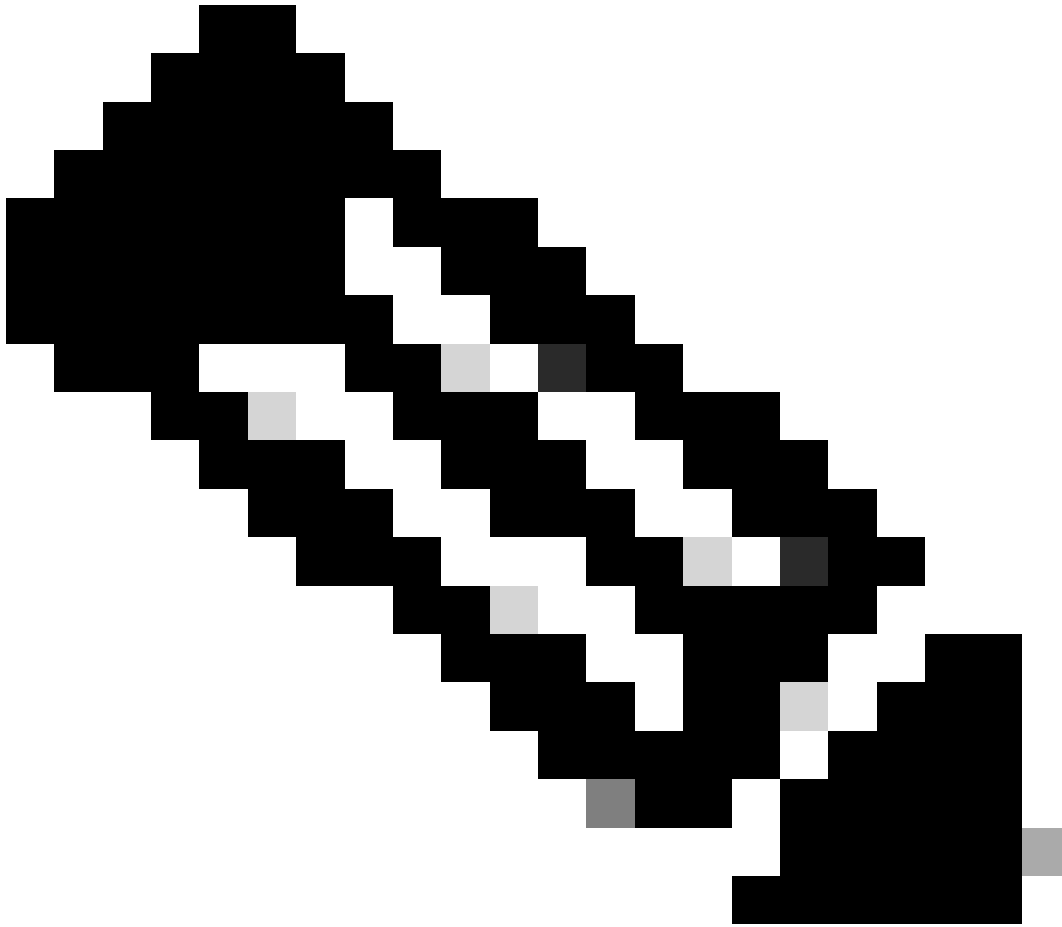
Este documento describe cómo resolver JOINING problemas de estado con la máquina virtual (VM) de Cisco Policy Suite (CPS)-Diameter Routing Agent (DRA).

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Linux
- CPS



Nota: Cisco recomienda que tenga acceso de raíz con privilegios a CPS DRA CLI.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

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

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

El agente de routing de diámetro virtual (vDRA) de CPS actúa como componente operativo dentro de una red y guía los mensajes a los nodos de destino deseados mediante el uso de algoritmos de routing.

La función central de CPS vDRA implica el enrutamiento de mensajes y la posterior transmisión de respuestas a sus puntos de origen originales.

CPS vDRA, que consta de un conjunto de máquinas virtuales (VM) orquestadas como un clúster mediante motores Docker, consta de distintas entidades, a saber, máquinas virtuales Master, Control, Director, Distributor y 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]#
```

Estado: indica si la aplicación de planificación está conectada al motor de acoplamiento y se ejecuta en un host.

Ping perdidos - La cantidad de pings perdidos consecutivos para un host determinado.

Problema

A veces, la VM vDRA de CPS se bloquea en el estado JOINING debido a varias razones.

<#root>

admin@orchestrator[master-1]#

show docker engine

Fri Jul 14 09:36:18.635 UTC+00:00

MISSED

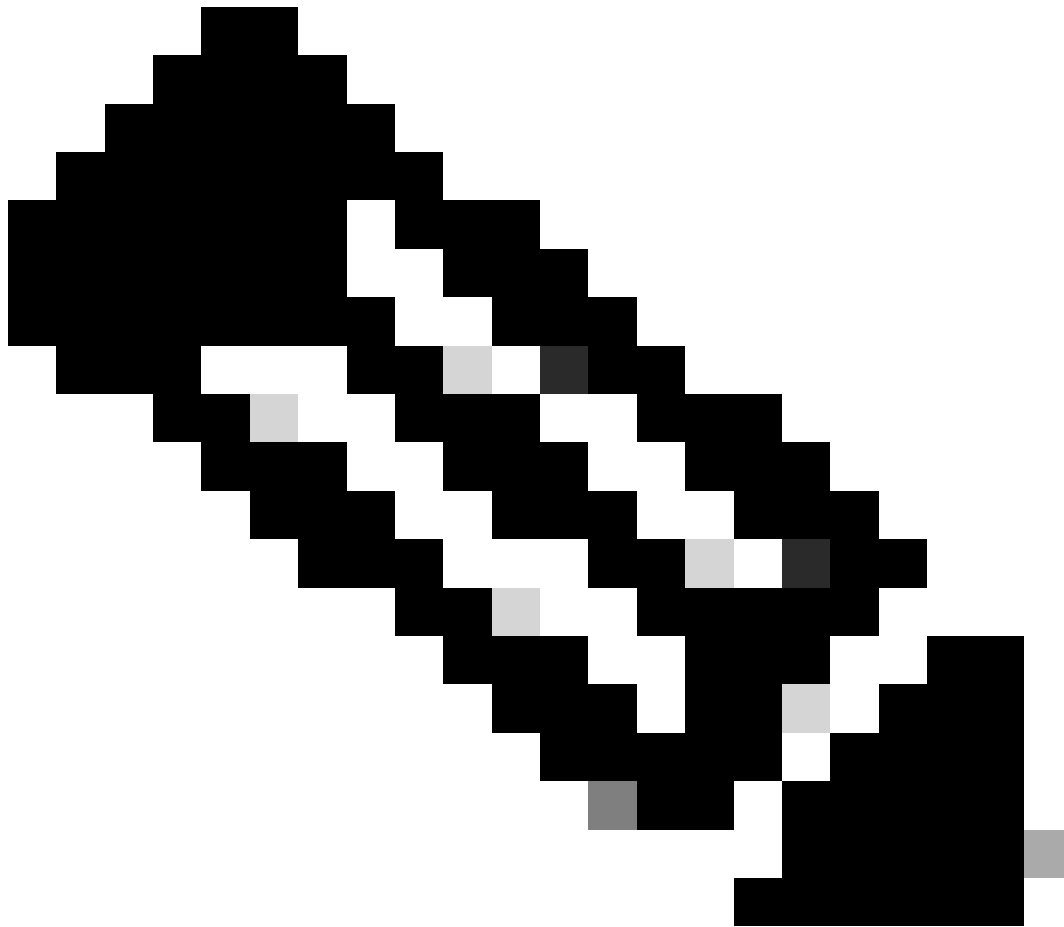
ID STATUS PINGS

```
-----  
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]#
```

Las posibles razones para que la VM se atasque en el JOINING estado,

1. No se puede acceder a la VM desde la VM maestra.

1.1. Verifique si el estado de las conexiones de tejido en la máquina virtual afectada está en modo de manguito.



Nota: Weave Net crea una red virtual que conecta contenedores Docker implementados en varios hosts y permite su detección automática. Con Weave Net, las aplicaciones portátiles basadas en microservicios que constan de varios contenedores pueden ejecutarse en cualquier lugar: en un host, varios hosts o incluso entre proveedores de nube y Data Centers. Las aplicaciones utilizan la red como si todos los contenedores estuvieran conectados al mismo switch de red, sin configurar los mapeos de puertos, embajadores o links.

CPS-DRA tiene dos estados principales de conexiones de tejido: `fastdp` y `sleeve`. La preferencia dentro del clúster CPS-DRA se orienta de forma coherente hacia la creación de conexiones de trama en el `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 si estos mensajes de registro están presentes en el journalctl registro en la máquina virtual afectada.

```
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. Se agota el espacio de disco de VM.

2.1. Verifique el uso de espacio en disco en la VM afectada e identifique la partición con uso de espacio en disco elevado.

```
<#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:~$
```

Procedimiento para recuperar máquinas virtuales CPS-DRA del estado JOINING

Enfoque 1.

Si la VM no es accesible desde la VM maestra, utilice este enfoque.

1. Verifique el estado de la conexión de trama en las VM afectadas, si es el modo de manguito.

```
#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 el tejido en las VM respectivas.

```
#docker restart weave
```

3. Compruebe si el estado de la conexión de tejido se ha movido al fastdp estado y la VM afectada se ha movido al CONNECTED estado.

4. Si las VM siguen atascadas en el JOINING estado, reinicie esas VM afectadas.

```
<#root>
```

```
#sudo reboot now
```

```
or
```

```
#init 6
```

5. Ahora verifique si la VM afectada se ha movido al CONNECTED estado.

```
<#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 si vPAS comienza a abastecer el tráfico y todos los contenedores están **ACTIVOS** (especialmente el punto final de diámetro); de lo contrario, reinicie el orchestrator-backup-a contenedor en drc01 VM.

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

7. Ahora, verifique si vPAS comenzó a procesar el tráfico.

Enfoque 2.

Si se agota el espacio en disco de la máquina virtual.

1. Identifique el directorio que consume mucho espacio en disco.

```
<#root>
```

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

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

```
176G 9dfd1bf36282c4e707a3858beba91bfaa383c78b5b9eb3acf0e58f335126d9b7
```

2. Verifique los archivos/registros/volcados que consumen un enorme espacio en 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. Identificar los contenedores que se ejecutan en VM impactada (especialmente contenedores insalubres).

```
<#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 el contenedor que desencadena archivos de núcleo voluminosos, para hacerlo inspeccione cada contenedor alojado en la máquina virtual afectada, uno por 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. Compruebe si tiene o no acceso a ese contenedor concreto.

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

6. Si no puede acceder a ese contenedor, elimine los archivos de núcleo voluminosos para liberar espacio.

```
<#root>
```

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

```
rm -rf core*
```

7. Inicie sesión en el contenedor afectado desde la máquina virtual afectada.

```
<#root>
```

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

8. Reinicie el app proceso en el contenedor para detener la generación de archivos de núcleo voluminosos.

```
<#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. Si el paso 8. no ayuda a detener la generación de archivos de núcleo masivo, reinicie el contenedor afectado.

```
<#root>
```

```
#
```

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

10. Compruebe si se ha detenido la generación masiva de archivos de núcleo.

11. Para devolver la máquina virtual afectada al estado CONNECTED, inicie sesión en el orchestrator contenedor y realice el orchestration-engine reinicio.

```
<#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. Si el paso 11. no ayuda a restaurar la VM, vaya a reinicio de proxy de motor en la VM afectada.

```
<#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. Ahora, verifique si la VM afectada se ha movido al CONNECTED estado.

```
<#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]#
```

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).