

Solución de problemas de Common Data Layer (CDL)

Contenido

[1. Introducción](#)

[2. Descripción general](#)

[3. Componentes](#)

[4. Tutorial sobre configuración](#)

[5. Solución de problemas](#)

[5.1 Fallas de POD](#)

[5.2 CDL Cómo obtener información de sesión de las claves de sesión](#)

[5.3 Las cápsulas de CDL no funcionan](#)

[5.4 Las vainas de Mirror Maker están en estado init](#)

[5.5 El índice de CDL no se replica adecuadamente](#)

[5.6 Las operaciones de CDL fallan, pero la conexión es correcta](#)

[5.7 La notificación para el registro de purga llegó antes o se retrasó desde la CDL](#)

[6. Alertas](#)

[7. Problemas más comunes](#)

[7.1 errorDeReplicaciónDeCdl](#)

[7.2 cdlRemoteConnectionFailure y GRPC Connections Remote Site](#)

[8. Grafana](#)

1. Introducción

Este artículo trata los aspectos básicos de la resolución de problemas de Common Data Layer (CDL) en entornos SMF. La documentación que puede encontrar en este [enlace](#).

2. Descripción general

Cisco Common Data Layer (CDL) es una capa de almacenamiento de datos KV (valor clave) de última generación y alto rendimiento para todas las aplicaciones nativas de la nube.

Actualmente, las CDL se utilizan como un componente de gestión estatal con funciones de HA (alta disponibilidad) y Geo HA.

La CDL proporciona:

- Una capa de almacén de datos común a través de diferentes funciones de red (NF).
- Lectura y escritura de baja latencia (en almacenamiento de sesión en memoria)
- Notifique a los NF que bloqueen al suscriptor cuando se informe de un ataque DoS (denegación de servicio) en la misma sesión.

- Alta disponibilidad: redundancia local con al menos 2 réplicas.
- Redundancia geográfica con 2 sitios.
- No hay ningún concepto principal/secundario para todas las ranuras disponibles para las operaciones de escritura. Mejora el tiempo de conmutación por fallo ya que no se lleva a cabo ninguna elección principal.

3. Componentes

- Terminal: (cdl-ep-session-c1-d0-7c79c87d65-xpm5v)
 - El criterio de valoración de CDL es un POD de Kubernetes (K8). Se implementa para exponer gRPC a través de la interfaz HTTP2 hacia el cliente NF, es para procesar solicitudes de servicio de base de datos y actúa como punto de entrada para las aplicaciones ascendentes.
- Ranura: (cdl-slot-session-c1-m1-0)
 - El terminal CDL admite microservicios de varias ranuras. Estos microservicios son POD K8s implementados para exponer la interfaz gRPC interna hacia el almacén de datos de Cisco
 - Cada POD de ranura contiene un número finito de sesiones. Estas sesiones son los datos reales de la sesión en formato de matriz de bytes
- Índice: (cdl-index-session-c1-m1-0)
 - El microservicio Index contiene los datos relacionados con la indexación
 - Estos datos de indexación se utilizan para recuperar los datos de sesión reales de los microservicios de ranura
- ETCD: (etcd-smf-etcd-cluster-0)
 - CDL utiliza ETCD (un almacén de valor de clave de código abierto) como detección de servicio de base de datos. Cuando se inicia, se cierra o se cierra el PE del almacén de datos de Cisco, el estado de publicación agrega un evento. Por lo tanto, se envían notificaciones a cada uno de los POD suscritos a estos eventos. Además, cuando se agrega o quita un evento de clave, actualiza el mapa local.
- Kafka: (kafka-0)
 - El POD Kafka replica los datos entre las réplicas locales y a través de los sitios para la indexación. Para la replicación entre sitios, Kafak utiliza MirrorMaker.
- Espejo Maker: (espejo-maker-0)
 - El POD Mirror Maker replica geográficamente los datos de indexación en los sitios CDL remotos. Toma los datos de los sitios remotos y los publica en el sitio local de Kafka para que las instancias de indexación apropiadas los recojan.

Ejemplo:

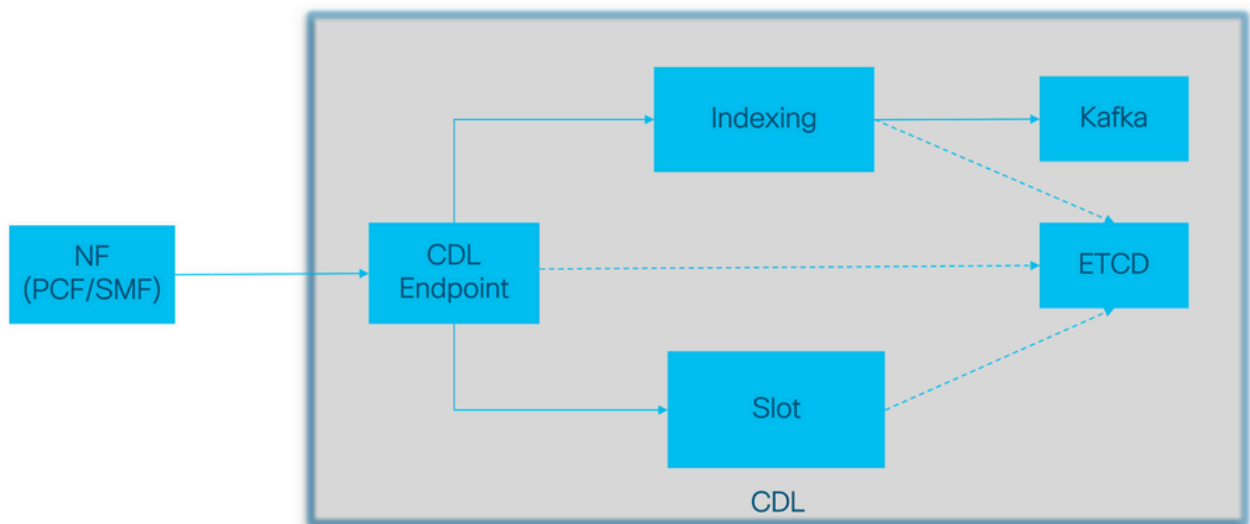
```

master-1:~$ kubectl get pods -n smf-smf -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
cdl-ep-session-c1-d0-7889db4d87-5m1n5 1/1     Running   0           80d   192.168.
cdl-ep-session-c1-d0-7889db4d87-8q7hg 1/1     Running   0           80d   192.168.
cdl-ep-session-c1-d0-7889db4d87-fj2nf 1/1     Running   0           80d   192.168.
cdl-ep-session-c1-d0-7889db4d87-z6c2z 1/1     Running   0           34d   192.168.
cdl-ep-session-c1-d0-7889db4d87-z7c89 1/1     Running   0           80d   192.168.
cdl-index-session-c1-m1-0             1/1     Running   0           80d   192.168.

```

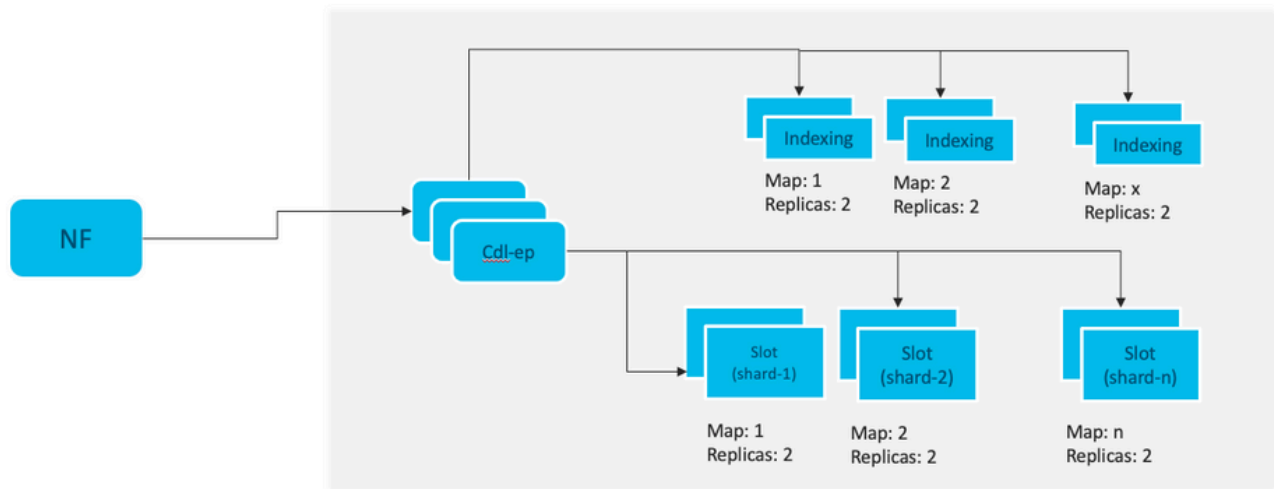
cdl-index-session-c1-m1-1	1/1	Running	0	80d	192.168.
cdl-index-session-c1-m2-0	1/1	Running	0	49d	192.168.
cdl-index-session-c1-m2-1	1/1	Running	0	80d	192.168.
cdl-index-session-c1-m3-0	1/1	Running	0	80d	192.168.
cdl-index-session-c1-m3-1	1/1	Running	0	80d	192.168.
cdl-index-session-c1-m4-0	1/1	Running	0	80d	192.168.
cdl-index-session-c1-m4-1	1/1	Running	0	49d	192.168.
cdl-slot-session-c1-m1-0	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m1-1	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m2-0	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m2-1	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m3-0	1/1	Running	0	34d	192.168.
cdl-slot-session-c1-m3-1	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m4-0	1/1	Running	0	34d	192.168.
cdl-slot-session-c1-m4-1	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m5-0	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m5-1	1/1	Running	0	34d	192.168.
cdl-slot-session-c1-m6-0	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m6-1	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m7-0	1/1	Running	0	80d	192.168.
cdl-slot-session-c1-m7-1	1/1	Running	0	49d	192.168.
cdl-slot-session-c1-m8-0	1/1	Running	0	49d	192.168.
cdl-slot-session-c1-m8-1	1/1	Running	0	80d	192.168.
etcd-smf-smf-etcd-cluster-0	2/2	Running	0	80d	192.168.11.17
etcd-smf-smf-etcd-cluster-1	2/2	Running	0	48d	192.168.7.59
etcd-smf-smf-etcd-cluster-2	2/2	Running	1	34d	192.168.11.66
georeplication-pod-0	1/1	Running	0	80d	10.10.1.
georeplication-pod-1	1/1	Running	0	48d	10.10.1.
grafana-dashboard-cdl-smf-smf-77bd69cff7-qbvmv	1/1	Running	0	34d	192.168.7.41
kafka-0	2/2	Running	0	80d	192.168.
kafka-1	2/2	Running	0	49d	192.168.
mirror-maker-0	1/1	Running	1	80d	192.168.
zookeeper-0	1/1	Running	0	34d	192.168.
zookeeper-1	1/1	Running	0	48d	192.168.
zookeeper-2					

CDL Architecture



→ Actual call data flow
 - - - - - CDL Control flow(Registration and Discovery)

CDL Architecture with primary components



x: Indexing maps /shards
 n: Slot maps /shards
 For HA purpose, each map should have 2 replicas minimum
 Data distributed in multiple maps. Data in each map is different.
 Each replica in a map has identical data for redundancy.

Nota: No hay ningún concepto principal/secundario para todas las ranuras disponibles para las operaciones de escritura. Mejora el tiempo de conmutación por fallo ya que no se lleva a cabo ninguna elección principal.

Nota: De forma predeterminada, la CDL se implementa con 2 réplicas para db-ep, 1 mapa de ranuras (2 réplicas por mapa) y 1 mapa de índice (2 réplicas por mapa).

4. Tutorial sobre configuración

```
smf# show running-config cd1
cd1 system-id 1 // unique across the site, system-id 1 is the primary site ID for
cd1 node-type db-data // node label to configure the node affinity
cd1 enable-geo-replication true // CDL GR Deployment with 2 RACKS
cd1 remote-site 2
db-endpoint host x.x.x.x // Remote site cd1-ep configuration on site-1
db-endpoint port 8882
kafka-server x.x.x.x 10061 // Remote site kafka configuration on site-1
exit
kafka-server x.x.x.x 10061
exit
exit
cd1 label-config session // Configures the list of label for CDL pods
endpoint key smi.cisco.com/node-type-3
endpoint value session
slot map 1
```

```

key smi.cisco.com/node-type-3
value session
exit
slot map 2
key smi.cisco.com/node-type-3
value session
exit
slot map 3
key smi.cisco.com/node-type-3
value session
exit
slot map 4
key smi.cisco.com/node-type-3
value session
exit
slot map 5
key smi.cisco.com/node-type-3
value session
exit
slot map 6
key smi.cisco.com/node-type-3
value session
exit
slot map 7
key smi.cisco.com/node-type-3
value session
exit
slot map 8
key smi.cisco.com/node-type-3
value session
exit
index map 1
key smi.cisco.com/node-type-3
value session
exit
index map 2
key smi.cisco.com/node-type-3
value session
exit
index map 3
key smi.cisco.com/node-type-3
value session
exit
index map 4
key smi.cisco.com/node-type-3
value session
exit

```

```

exit
cdl datastore session // unique with in the site

```

```

label-config session
geo-remote-site [ 2 ]
slice-names [ SMF1 SMF2 ]
endpoint cpu-request 2000
endpoint go-max-procs 16
endpoint replica 5
endpoint external-ip x.x.x.x
endpoint external-port 8882
index cpu-request 2000
index go-max-procs 8
index replica 2

```

```

// number of cdl-ep pods

```

```

// number of replicas per mop for cdl-index, can not be changed a

```

```

NOTE: If you need to change number of index replica, set the system mode to shutdown from respective op
index map 4 // number of mops for cdl-index

```

```

index write-factor 1          /// number of copies to be written before a successful response
slot cpu-request 2000
slot go-max-procs 8
slot replica 2                /// number of replicas per mop for cdl-slot
slot map 8                     /// number of mops for cdl-slot
slot write-factor 1
slot metrics report-idle-session-type true
features instance-aware-notification enable true    /// This enables GR failover notification
features instance-aware-notification system-id 1
  slice-names [ SMF1 ]
exit
features instance-aware-notification system-id 2
  slice-names [ SMF2 ]
exit
exit
cdl kafka replica 2
cdl kafka label-config key smi.cisco.com/node-type-3
cdl kafka label-config value session
cdl kafka external-ip x.x.x.x 10061
exit
cdl kafka external-ip x.x.x.x 10061
exit

```

5. Solución de problemas

5.1 Fallas de POD

El funcionamiento de la CDL es sencillo: clave > base de datos de valor.

- Todas las solicitudes llegan a las vainas cdl-endpoint.
- En las vainas cdl-index almacenamos las llaves, ordenamiento cíclico.
- En cdl-slot almacenamos valor (información de sesión), ordenamiento cíclico.
- Definimos una copia de seguridad (número de réplicas) para cada mapa de grupo de dispositivos (tipo).
- La cápsula Kafka se utiliza como autobús de transporte.
- El fabricante de espejos se utiliza como bus de transporte a diferentes racks (redundancia geo).

La falla para cada uno podría traducirse como, es decir, si todas las vainas de este tipo/mapa cayeran al mismo tiempo:

- cdl-terminal - errores de comunicación con CDL
- cdl-index - pérdida de claves para datos de sesión
- cdl-slot - pérdida de datos de sesión
- Kafka - perdiendo la opción de sincronización entre los mapas de tipo pod
- mirror maker - pérdida de sincronización con otro nodo geo redundand

Siempre podemos recopilar registros de grupos de dispositivos relevantes porque los registros de grupos de dispositivos en cdl no se traspasan tan rápido, por lo que hay un valor adicional para recopilarlos.

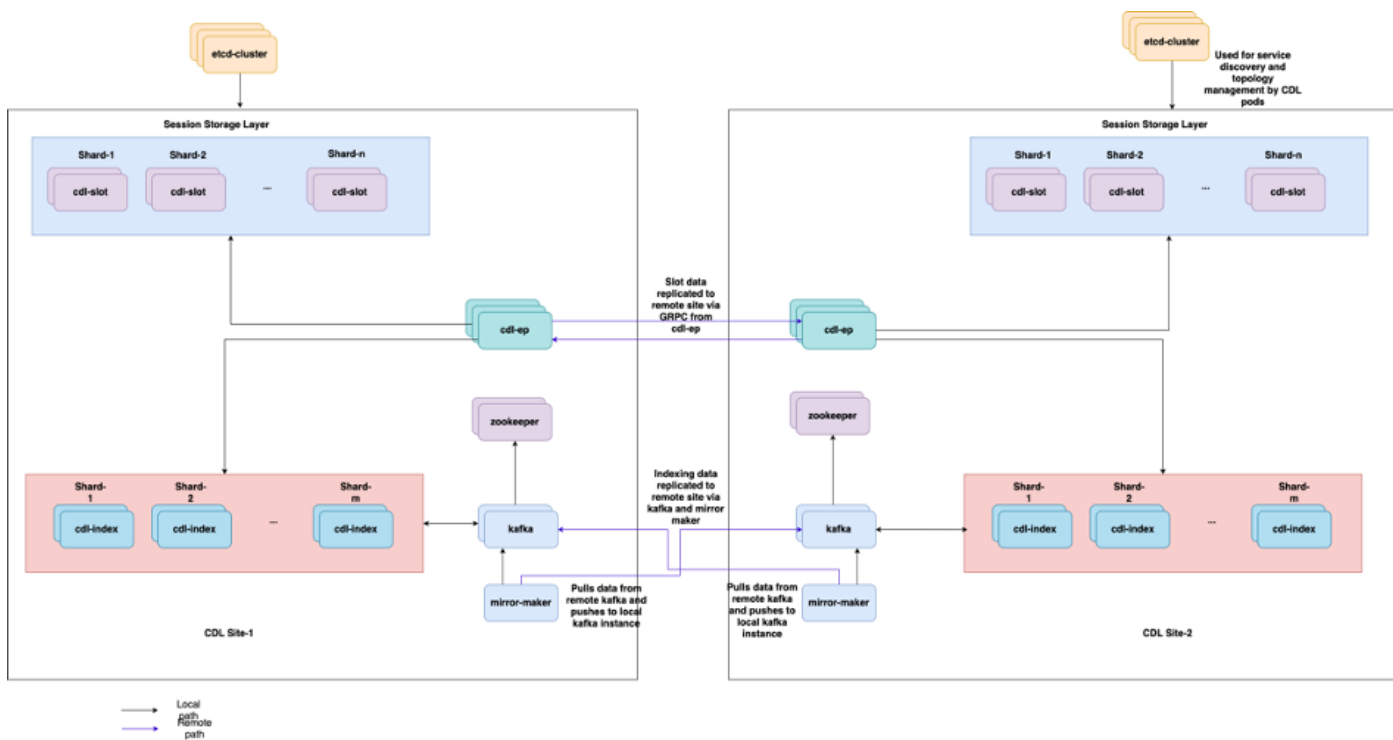
Remember tac-debug recopila la instantánea en el tiempo mientras los registros imprimen todos los datos desde que se almacenan.

Describir grupos de dispositivos

```
kubectl describe pod cdl-ep-session-c1-d0-7889db4d87-5m1n5 -n smf-rcdn
```

Recopilar registros de grupos de dispositivos

```
kubectl logs cdl-ep-session-c1-d0-7c79c87d65-xpm5v -n smf-rcdn
```



5.2 CDL Cómo obtener información de sesión de las claves de sesión

Dentro de la CDL cada sesión tiene un campo llamado llaves únicas que identifica esta sesión.

Si comparamos la impresión de la sesión de show subscriber supi y cdl show sessions summary slice-name slice1 db-name session filter

- dirección de sesión ipv4 combinada con supi = "1##/##imsi-123969789012404:10.0.0.3"
- ddn + dirección ip4 = "1##/##lab:10.0.0.3"
- dirección de sesión ipv6 combinada con supi = "1##/##imsi-123969789012404:2001:db0:0:2::"
- ddn + dirección ipv6 de la sesión = "1##/##lab:2001:db0:0:2::"
- smfTeid también N4 Session Key = "1##/##293601283" Esto es realmente útil cuando se solucionan errores en UPF, puede buscar en los registros de sesión y encontrar información

relacionada con la sesión.

- supi + ebi = "1##imsi-123969789012404:ebi-5"
- supi + ddn= "1##imsi-123969789012404:lab"

```
[smf/data] smf# cd| show sessions summary slice-name slice1 db-name session filter { condition match ke
Sun Mar 19 20:17:41.914 UTC+00:00
message params: {session-summary cli session {0 100 1##/293601283 0 [{0 1##/293601283}] [] 0 0 false 40
session {
  primary-key 1##/imsi-123969789012404:1
  unique-keys [ "1##/imsi-123969789012404:10.0.0.3" "1##/lab:10.0.0.3" "1##/imsi-123969789012404:2001
  non-unique-keys [ "1##/roaming-status:visitor-lbo" "1##/ue-type:nr-capable" "1##/supi:imsi-12396978
  flags [ flag3:peerGtpuEpKey:10.10.10.150:20.0.0.1 session-state-flag:smf_active ]
  map-id 2
  instance-id 1
  app-instance-id 1
  version 1
  create-time 2023-03-19 20:14:14.381940117 +0000 UTC
  last-updated-time 2023-03-19 20:14:14.943366502 +0000 UTC
  purge-on-eval false
  next-eval-time 2023-03-26 20:14:14 +0000 UTC
  session-types [ rat_type:NR wps:non_wps emergency_call:false pdu_type:ipv4v6 dnn:lab qos_5qi_1_rat_
  data-size 2866
}
[smf/data] smf#
```

Si lo comparamos con la impresión del SMF:

```
[smf/data] smf# show subscriber supi imsi-123969789012404 gr-instance 1 namespace smf
Sun Mar 19 20:25:47.816 UTC+00:00
subscriber-details
{
  "subResponses": [
    [
      "roaming-status:visitor-lbo",
      "ue-type:nr-capable",
      "supi:imsi-123969789012404",
      "gpsi:msisdn-22331010101010",
      "pei:imei-123456789012381",
      "psid:1",
      "snssai:001000003",
      "dnn:lab",
      "emergency:false",
      "rat:nr",
      "access:3gpp access",
      "connectivity:5g",
      "udm-uecm:10.10.10.215",
      "udm-sdm:10.10.10.215",
      "auth-status:unauthenticated",
      "pcfGroupId:PCF-dnn=lab;",
      "policy:2",
      "pcf:10.10.10.216",
      "upf:10.10.10.150",
      "upfEpKey:10.10.10.150:10.10.10.202",
    ]
  ]
}
```



```

    "ipv4-addr:pool1/10.0.0.3",
    "ipv4-pool:pool1",
    "ipv4-range:pool1/10.0.0.1",
    "ipv4-startrange:pool1/10.0.0.1",
    "ipv6-pfx:pool1/2001:db0:0:2::",
    "ipv6-pool:pool1",
    "ipv6-range:pool1/2001:db0::",
    "ipv6-startrange:pool1/2001:db0::",
    "id-index:1:0:32768",
    "id-value:2/3",
    "chfGroupId:CHF-dnn=lab;",
    "chf:10.10.10.218",
    "amf:10.10.10.217",
    "peerGtpuEpKey:10.10.10.150:20.0.0.1",
    "namespace:smf",
    "nf-service:smf"
  ]
}

```

Comprobar el estado de CDL en SMF:

```

cdl show status
cdl show sessions summary slice-name <slice name> | more

```

5.3 Las cápsulas de CDL no funcionan

Cómo identificar

Verifique el resultado de los grupos de dispositivos descritos (contenedores/miembro/Estado/Razón, eventos).

```
kubectl describe pods -n <namespace> <failed pod name>
```

Cómo arreglar

1. Pods están en estado pendiente Verifique si algún nodo k8s con los valores de etiqueta iguales al valor de cdl/node-type el número de réplicas es menor o igual al número de nodos k8s con los valores de etiqueta iguales al valor de cdl/node-type

```
kubectl get nodes -l smi.cisco.com/node-type=<value of cdl/node-type, default value is 'session' i
```

2. Pods están en estado de falla CrashLoopBackOff Compruebe el estado de los pods etcd. Si las vainas etcd no se están ejecutando, corrija los problemas de etcd.

```
kubectl describe pods -n <namespace> <etcd pod name>
```

3. Los grupos de dispositivos están en estado de error ImagePullBack. Comprobar si el repositorio de helm y el registro de imágenes están accesibles. Compruebe si se han configurado los servidores DNS y proxy necesarios.

5.4 Las vainas de Mirror Maker están en estado init

Verifique la salida de los pods descriptos y los registros de pod

```
kubectl describe pods -n <namespace> <failed pod name>  
kubectl logs -n <namespace> <failed pod name> [-c <container name>]
```

Cómo arreglar

- Verifique si las IP externas configuradas para Kafka son correctas
- Comprobar la disponibilidad de kafka de sitio remoto a través de IP externas

5.5 El índice de CDL no se replica adecuadamente

Cómo identificar

Los datos agregados a un sitio no son accesibles desde otro sitio.

Cómo arreglar

- Compruebe la configuración del ID del sistema local y la configuración del sitio remoto.
- Compruebe la disponibilidad de los terminales CDL y kafka entre cada sitio.
- Compruebe el mapa, la réplica del índice y la ranura en cada sitio. Puede ser idéntico en todos los sitios.

5.6 Las operaciones de CDL fallan, pero la conexión es correcta

Cómo arreglar

- Verifique que todas las vainas estén listas y en ejecución.
- Los grupos de indexación están en estado preparado sólo si la sincronización se ha completado con la réplica de par (local o remota si está disponible)
- Los grupos de dispositivos de ranura están en estado preparado sólo si la sincronización se ha completado con la réplica de par (local o remota si está disponible)
- Los terminales NO están en estado preparado si al menos una ranura y un grupo de dispositivos de índice no están disponibles. Incluso si no está listo, el cliente aceptará la conexión grpc.

5.7 La notificación para el registro de purga llegó antes o se retrasó desde la CDL

Cómo arreglar

- En un clúster k8, todos los nodos pueden sincronizarse a la hora
- Verifique el estado de sincronización de NTP en todos los nodos k8s. Si hay algún problema, soluciónelo.

```
chronyc tracking
chronyc sources -v
chronyc sourcestats -v
```

6. Alertas

ALARMA	gravedad	summary
cdlLocalRequestFailure	crítico	Si la tasa de éxito de las solicitudes locales es inferior al 90% durante más de 5 minutos, activa la alarma
cdlRemoteConnectionFailure	crítico	Si las conexiones activas del grupo de terminales al sitio remoto alcanzaron 0 durante más de 5 minutos, se activa la alarma (solo para el sistema habilitado para GR)
cdlRemoteRequestFailure	crítico	Si la tasa de éxito de las solicitudes remotas entrantes es inferior al 90% durante más de 5 minutos, activa la alarma (solo para sistemas con GR activado)
cdlReplicationError	crítico	Si la proporción de solicitudes de replicación salientes con respecto a las solicitudes locales en el espacio de nombres cdl-global ha descendido por debajo del 90% durante más de 5 minutos (sólo para el sistema habilitado para GR). Estas alertas se esperan durante la actividad de actualización y, por lo tanto, puede ignorarlas.
cdlKafkaRemoteReplicationDelay	crítico	Si el retraso de replicación kafka a sitio remoto, cruza

ALARMA	gravedad	summary
		10 segundos durante más de 5 minutos, entonces la alarma se activa (solo para el sistema habilitado GR)
cdlOverloaded - principal	principal	Si el sistema CDL alcanza el porcentaje configurado (valor predeterminado del 80%) de su capacidad, el sistema activará la alarma (solo si la función Protección contra sobrecarga está activada)
cdlOverloaded - crítico	crítico	Si el sistema CDL alcanza el porcentaje configurado (predeterminado 90%) de su capacidad, el sistema dispara la alarma (solo si la función Protección contra sobrecarga está habilitada)
cdlKafkaConnectionFailure	crítico	Si las vainas con índice de CDL se desconectan de kafka por más de 5 minutos

7. Problemas más comunes

7.1 errorDeReplicaciónDeCdl

Esta alerta se ve generalmente durante la activación del centro de operaciones o la actualización del sistema, intente encontrar CR para ella, intente verificar la aparición de la alerta CEE y si ya se borró.

7.2 cdlRemoteConnectionFailure y GRPC_Connections_Remote_Site

La explicación se aplica a todas las alertas "cdlRemoteConnectionFailure" y "GRPC_Connections_Remote_Site".

Para alertas de cdlRemoteConnectionFailure:

En los registros de terminales de CDL, se ve que se perdió la conexión con el host remoto desde el grupo de terminales de CDL:

```
2022/01/20 01:36:18.852 [ERROR] [RemoteEndpointConnection.go:572] [datastore.ep.session] Connection to r
```

Pudimos ver que el grupo de terminales CDL intentaba conectarse al servidor remoto pero el host remoto lo rechazó:

```
2022/01/20 01:37:08.730 [WARN] [checksum.go:836] [datastore.checksum.session] Could not get checksum fr
2022/01/20 01:37:08.732 [WARN] [checksum.go:836] [datastore.checksum.session] Could not get checksum fr
2022/01/20 01:37:08.742 [WARN] [checksum.go:836] [datastore.checksum.session] Could not get checksum fr
2022/01/20 01:37:08.742 [WARN] [checksum.go:836] [datastore.checksum.session] Could not get checksum fr
2022/01/20 01:37:08.752 [WARN] [checksum.go:836] [datastore.checksum.session] Could not get checksum fr
2022/01/20 01:37:08.754 [WARN] [checksum.go:836] [datastore.checksum.session] Could not get checksum fr
```

Dado que el host remoto siguió siendo inalcanzable durante 5 minutos, se generó la siguiente alerta:

```
alerts history detail cd1RemoteConnectionFailure f5237c750de6
severity critical
type "Processing Error Alarm"
startsAt 2025-01-21T01:41:26.857Z
endsAt 2025-01-21T02:10:46.857Z
source cd1-ep-session-c1-d0-6d86f55945-pxfx9
summary "CDL endpoint connections from pod cd1-ep-session-c1-d0-6d86f55945-pxfx9 and namespace smf-rcdn
Labels [ "alertname: cd1RemoteConnectionFailure" "cluster: smf-data-rcdn_cee" "monitor: prometheus" "na
annotations [ "summary: CDL endpoint connections from pod cd1-ep-session-c1-d0-6d86f55945-pxfx9 and nam
```

La conexión al host remoto se realizó correctamente a las 02:10:32:

```
2022/01/20 02:10:32.702 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.923 [WARN] [RemoteEndpointConnection.go:563] [datastore.ep.session] Cd1 status chang
2022/01/20 02:10:38.927 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.934 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.934 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.935 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
```

Configuración presente en SMF para el sitio remoto de CDL:

```
cd1 remote-site 2
db-endpoint host 10.10.10.141
db-endpoint port 8882
kafka-server 10.10.19.139 10061
exit
kafka-server 10.10.10.140 10061
exit
exit
```

Para la alerta GRPC_Connections_Remote_Site:

La misma explicación se aplica también a "GRPC_Connections_Remote_Site", ya que también procede del mismo grupo de terminales CDL.

```
alerts history detail GRPC_Connections_Remote_Site f083cb9d9b8d
severity critical
type "Communications Alarm"
startsAt 2025-01-21T01:37:35.160Z
endsAt 2025-01-21T02:11:35.160Z
source cd1-ep-session-c1-d0-6d86f55945-pxfx9
summary "GRPC connections to remote site are not equal to 4"
labels [ "alertname: GRPC_Connections_Remote_Site" "cluster: smf-data-rcdn_cee" "monitor: prometheus" "
```

En los registros de grupos de terminales de CDL, la alerta se inició cuando se rechazó la conexión al host remoto:

```
2022/01/20 01:36:18.852 [ERROR] [RemoteEndpointConnection.go:572] [datastore.ep.session] Connection to r
```

La alerta se borró cuando la conexión al sitio remoto se realizó correctamente:

```
2022/01/20 02:10:32.702 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.923 [WARN] [RemoteEndpointConnection.go:563] [datastore.ep.session] Cd1 status chang
2022/01/20 02:10:38.927 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.934 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.934 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
2022/01/20 02:10:38.935 [WARN] [RemoteEndpointConnection.go:437] [datastore.ep.session] Stream to remote
```

8. Grafana

El panel de CDL forma parte de cada implementación de SMF.

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).