Troubleshoot Common Data Layer (CDL)

Contents

1. Introduction

This article will cover basics of troubleshooting Common Data Layer (CDL) in SMF environment. Documentation you can find on this link.

2. Overview

The Cisco Common Data Layer (CDL) is a high-performance next generation KV (Key-value) data store layer for all the Cloud Native applications.

CDL is currently used as a state management component with HA (High Availability) and Geo HA functions.

The CDL provides:

- A Common Data Store Layer across different Network Functions (NFs).
- Low latency read and write (in memory session storage)
- Notify the NFs to block the subscriber when a DoS (Denial of Service) attack on the same session is reported.
- High Availability - Local redundancy with at least 2 replicas.
- Geo Redundancy with 2 sites.
- No primary/secondary concept all slots available for write operations. Improves failover time as no primary election takes place.

3. Components

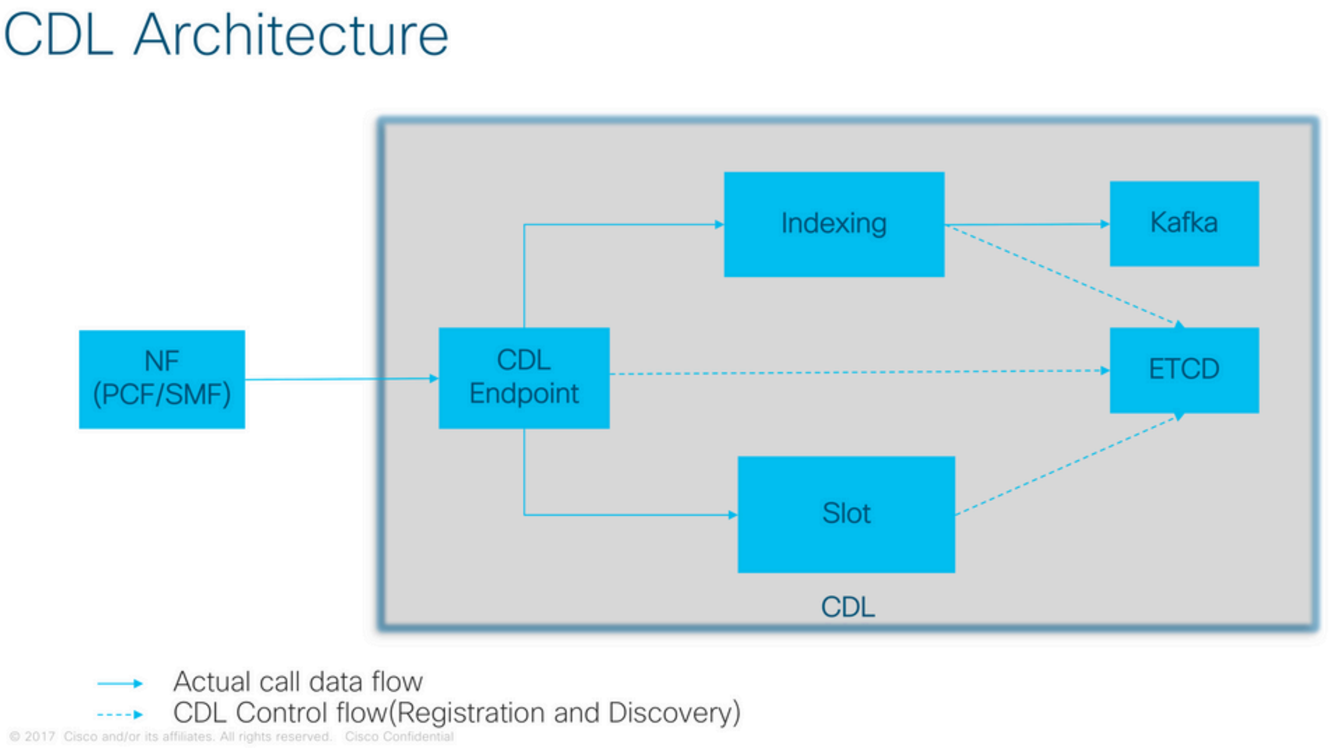
- Endpoint: (cdl-ep-session-c1-d0-7c79c87d65-xpm5v)

  - The CDL endpoint is a Kubernetes (K8s) POD. It is deployed for exposing gRPC over HTTP2 interface towards the NF client is for processing database service requests and acts as an entry point for the north-bound applications.
- Slot: (cdl-slot-session-c1-m1-0)
  - The CDL endpoint supports multiple Slot microservices. These microservices are K8s POD deployed for exposing internal gRPC interface towards the Cisco Data Store
  - Each Slot POD holds a finite number of sessions. These sessions is the actual session data in byte array format
- Index: (cdl-index-session-c1-m1-0)
  - The Index microservice holds the indexing related data
  - This indexing data is then used to retrieve the actual session data from the slot microservices
- ETCD: (etcd-smf-etcd-cluster-0)
  - CDL uses the ETCD (an open-source key-value store) as the DB service discovery. When the Cisco Data Store EP is started, killed or shutdown, it results in the addition of an event by the publishing state. Therefore, notifications are sent to each of the PODs subscribed to these events. Moreover, when a key event is added or removed, it refreshes the local map.
- Kafka: (kafka-0)
  - The Kafka POD replicates data between the local replicas and across sites for Indexing. For replication across sites, Kafak uses MirrorMaker.
- Mirror Maker: (mirror-maker-0)
  - The Mirror Maker POD geo-replicates the indexing data to the remote CDL sites. It takes data from the remote sites and publishes it to the local Kafka site for the appropriate indexing instances to pick up.

Example:

```
master-1:~$ kubectl get pods -n smf-smf -o wide
NAME                                                        READY   STATUS    RESTARTS   AGE   IP
cdl-ep-session-c1-d0-7889db4d87-5mln5                       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
```



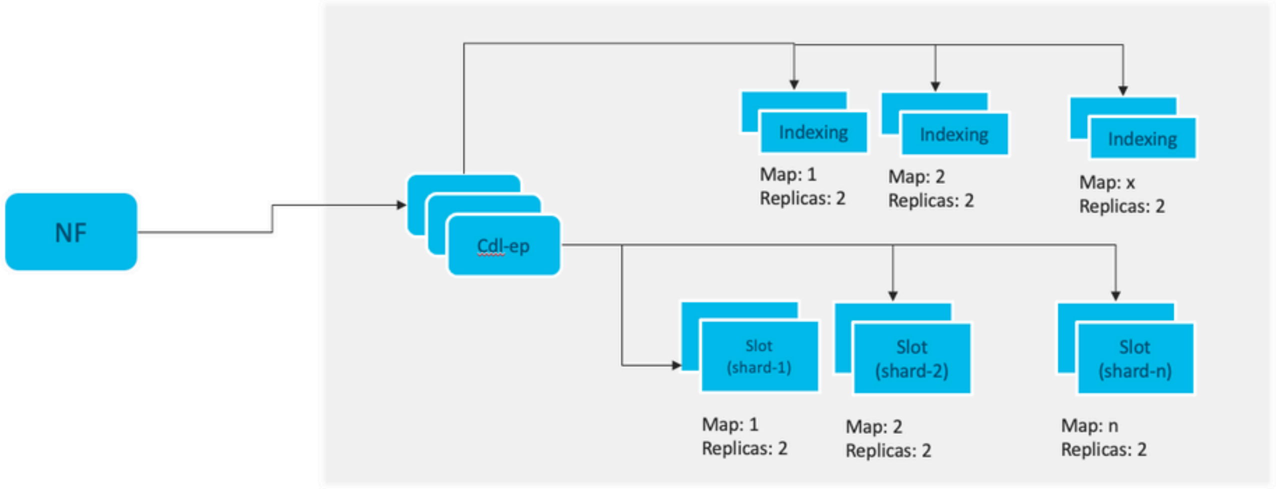
CDL Architecture

→ Actual call data flow

⇢ CDL Control flow(Registration and Discovery)

CDL Architecture

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.

Note: No primary/secondary concept all slots available for write operations. Improves failover time as no primary election takes place.

Note: By default CDL is deployed with 2 replicas for db-ep, 1 slot map (2 replicas per map) and 1 index map (2 replicas per map).

4. Configuration Walkthrough

```
smf# show running-config cdl
cdl system-id          1          /// unique across the site, system-id 1 is the primary site ID for 
cdl node-type          db-data    /// node label to configure the node affinity
cdl enable-geo-replication true   /// CDL GR Deployment with 2 RACKS
cdl remote-site 2
 db-endpoint host x.x.x.x      /// Remote site cdl-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
cdl 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                  /// number of cdl-ep pods
 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 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. Troubleshoot

5.1 Pod Failures

Operation of CDL is straightforward Key > Value db.

- All requests come to the cdl-endpoint pods.
- In cdl-index pods we store keys, round robin.
- In cdl-slot we store value (session info), round robin.
- We define backup (number of replicas) for each pod map (type).
- Kafka pod is used as transport bus.
- mirror maker is used as transport bus to different rack (Geo redundancy).

Failiure for each could be translated as, that is if all pods of this type/map went down at same time:

- cdl-endpont - errors of communicating with CDL
- cdl-index - loosing keys to session data
- cdl-slot - loosing session data
- Kafka - loosing sync option between the pod type maps
- mirror maker - losing sync with other geo redudand node

We can always collect logs from relevant pods because cdl pod logs do not rollover as fast, so there is extra value to collect them.

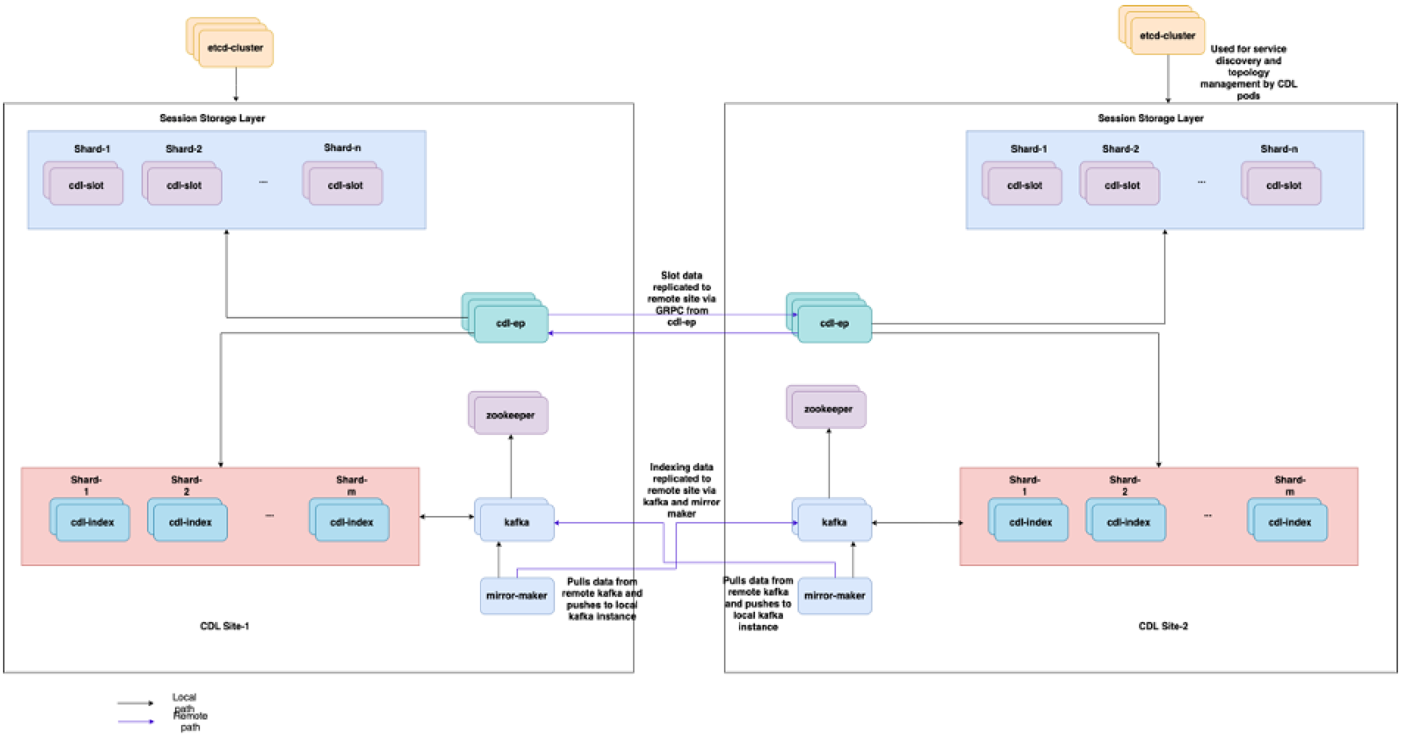
Remamber tac-debug collects snapshot in time while logs print out all data since it is stored.

Describe pods

```
kubectl describe pod cdl-ep-session-c1-d0-7889db4d87-5mln5 -n smf-rcdn
```

Collect pod logs

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



5.2 CDL How to Get Session Information from Session Keys

Inside CDL each session has a field called unique-keys that identifies this session.

If we compare session printout from show subscriber supi and cdl show sessions summary slice-name slice1 db-name session filter

- ipv4 session address combined with supi = "1#/#imsi-123969789012404:10.0.0.3"
- ddn + ip4 address = "1#/#lab:10.0.0.3"
- ipv6 session address combined with supi = "1#/#imsi-123969789012404:2001:db0:0:2::"
- ddn + ipv6 address from session = "1#/#lab:2001:db0:0:2::"
- smfTeid also N4 Session Key = "1#/#293601283" This is really useful when troubleshooting errors on UPF, you can search through session logs and find session related information.
- supi + ebi = "1#/#imsi-123969789012404:ebi-5"
- supi + ddn= "1#/#imsi-123969789012404:lab"

```
[smf/data] smf# cdl 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#
```

If we compare it to the printout from the 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"
    ]
  ]
}
```

Check CDL status on SMF:

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

5.3 CDL Pods are Not Up

How to identify

Check the describe pods output (containers/member/State/Reason, events).

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

How to fix

1. Pods are in pending state Check the if any k8s node with the label values equal to value of cdl/node-type number of replicas are less than or equal to number of k8s nodes with the label values equal to value of cdl/node-type

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

2. Pods are in CrashLoopBackOff failure State Check the etcd pods status. If etcd pods are not running, fix the etcd issues.

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

3. Pods are in ImagePullBack failure state Check if the helm repository and image registry is accessible. Check if required proxy and dns servers are configured.

5.4 Mirror Maker pods are in init state

Check the describe pods output and pod logs

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

How to fix

- Check if the external IPs configured for Kafka is correct
- Check the availability of remote site kafka through external IPs

5.5 CDL Index are not replicated properly

How to identify

Data which is added at one site is not accessible from other site.

How to fix

- Check the Local system id configuration and remote site configuration.
- Check the reachability of CDL endpoints and kafka between each sites.
- Check the map, replica of index and slot on each site. It can be identical across all site.

5.6 CDL operations are failing, but connection success

How to fix

- Check all pods are in ready and running state.
- Index pods are in ready state only if they sync is complete with peer replica (local or remote if available)
- Slot pods are in ready state only if they sync is complete with peer replica (local or remote if available)
- Endpoint are NOT in ready state if at least one slot and one index pods is not available. Even if it is not ready grpc connection will accepted from the client.

5.7 Notification for purging record came early/delayed from CDL

How to fix

- In a k8s cluster all nodes can be time synchronized
- Check NTP sync status on all k8s nodes. If there are any issues fix it.

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

6. Alerts

ALARM	severity	summary
cdlLocalRequestFailure	critical	If local requests success rate is less than 90% for more than 5 minutes, triggers the alarm
cdlRemoteConnectionFailure	critical	If active connections from endpoint pod to remote site reached 0 for longer than 5 minutes , then alarm is raised (only for GR enabled system)
cdlRemoteRequestFailure	critical	If incoming remote requests success rate is less than 90% for more than 5 minutes, triggers the alarm (only for GR enabled

ALARM	severity	summary
		system)
cdlReplicationError	critical	If ratio of outgoing replication requests to local requests in cdl-global namespace has gone under 90% for more than 5 minutes.(only for GR enabled system). These alerts are expected during upgrade activity and hence you can ignore them.
cdlKafkaRemoteReplicationDelay	critical	If kafka replication delay to remote site, crosses 10 seconds for longer than 5 minutes then the alarm is raised (only for GR enabled system)
cdlOverloaded - major	major	If the CDL system reaches configured percentage(default 80%) of its capacity, then the system triggers the alarm (only if Overload Protection feature is enabled)
cdlOverloaded - critical	critical	If the CDL system reaches the configured percentage(default 90%)of its capacity, then the system triggers the alarm (only if Overload Protection feature is enabled)
cdlKafkaConnectionFailure	critical	If CDL index pods are disconnected from kafka for longer than 5 minutes

7. Most Common Problems

7.1 cdlReplicationError

This alert is usually seen during bringing up the ops center or system upgrade, try to find CR for it, try to check on CEE occurrence of alert and was it cleared already.

7.2 cdlRemoteConnectionFailure & GRPC_Connections_Remote_Site

The explanation is applicable for all the “cdlRemoteConnectionFailure” and “GRPC_Connections_Remote_Site” alerts.

For cdlRemoteConnectionFailure alerts:

From CDL endpoint logs, we see the connection to the remote host from CDL endpoint pod was lost:

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

We could see the CDL endpoint pod trying to connect to the remote server but its refused by the remote host:

```
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
```

Since the remote host continued to be unreachable for 5 min - the alert got raised as below:

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

Connection to remote host was successful at 02:10:32:

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

Config present in SMF for CDL remote site:

```
cdl 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
```

For Alert GRPC_Connections_Remote_Site:

The same explanation is applicable for “GRPC_Connections_Remote_Site” as well since it’s also from the same CDL endpoint pod.

```
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 cdl-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" "
```

From CDL endpoint pod logs, The alert started when the connection to remote host was refused:

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

The alert got cleared when the connection to remote site was successful:

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

8. Grafana

CDL Dashboard is part of every SMF Deployment.