# Cisco Common Data Layer



# Feature Summary and Revision History

## Summary Data

***Table 1: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | SMF |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled – Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 2: Revision History***

| Revision Details | Release |
|---|---|
| Added the failure handling support. | 2023.03.0 |
| Added the procedures for configuration and verification of the event trace data in the CDL database record. | 2021.02.0 |

| Revision Details | Release |
|---|---|
| First introduced. | Pre-2020.02.0 |

# Feature Description

The SMF extends support to the Geo Redundant (GR) version of the Cisco Common Data Layer (CDL). When the primary CDL endpoint fails, the SMF attempts the same operation on the next highly rated secondary endpoint thus providing a non-disrupted N7 or Diameter message handling. If the next rated endpoint is unavailable, then the SMF reattempts the operation on the subsequent endpoint that has the highest rating and so on.

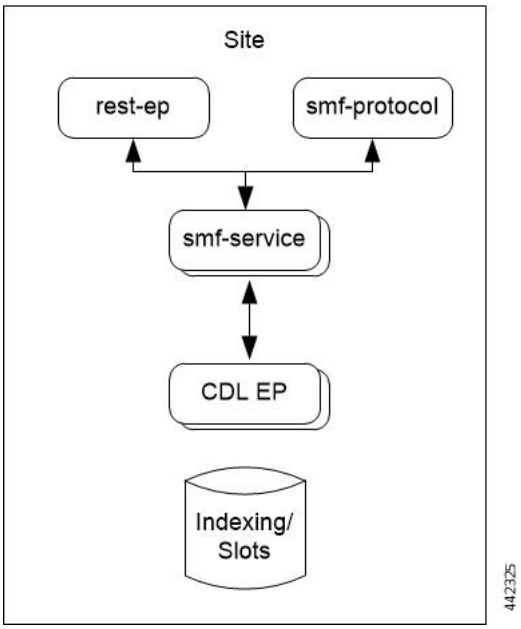
In case of specific errors or failures from CDL to SMF, the retry mechanism is used for the Create, Find, and Delete requests as well along with the earlier supported Update requests. The number of retries are as per the SLA.

When the SMF receives the successful response during the retries, then the retries are stopped and the response is sent to the application.

# Architecture

The following figure depicts the failover that happens when the SMF service is unable to access the CDL datastore endpoint.

***Figure 1: CDL Datastore Architecture***




With relevance to this architecture, you can configure CDL through SMF Ops Center. When the SMF connects to the CDL, it uses the local endpoints.

# How it Works

When CDL is configured in SMF through the SMF Ops Center, SMF gets enabled to support multiple CDL datastore endpoints. You can configure the endpoints by specifying the IP addresses, ports, and assigning ratings to each endpoint. By default, SMF considers the local endpoint as the primary endpoint, which has the maximum rating. SMF performs CDL API operations on the primary endpoint. If this endpoint is unavailable, then SMF routes the operations to the next maximum rated endpoint. SMF keeps failing over to the accessible secondary endpoint or until all the configured secondaries are exhausted. It does not reattempt a query on the next rated endpoint if the endpoint is reachable but responds with error or timeout.

If SMF is unable to access any of the endpoints in the cluster, then CDL operation fails with the "Datastore Unavailable" error.

The following table lists the scenarios, expected behavior, and the related error responses from CDL.

***Table 3: Scenarios, Expected Behavior, and Error Responses from CDL***

| Scenario | Expected Behavior | Error Response from CDL |
|---|---|---|
| All pods are active and healthy and trigger the CDL operations. | CDL Create, Read, Update, Delete (CRUD) operations are not retried. | Not applicable |
| CDL endpoint pod restarts and triggers CDL operations. | CDL CRUD operations are retried. | CDL sends the following error responses with the error code:<br>• ERROR_FROM_INDEXING [501]<br>• ERROR_FROM_SLOT [502]<br>• DATASTORE_EP_QUEUE_FULL [503]<br>• MAX_CAPACITY_REACHED [507 |

# Call Flows

This section describes the call flow that is associated with this feature.

• CDL Endpoint Failure Call Flow, on page 3

## CDL Endpoint Failure Call Flow

This section describes the SMF local data store endpoint failure call flow.

*Figure 2: CDL Endpoint Failure Call Flow*

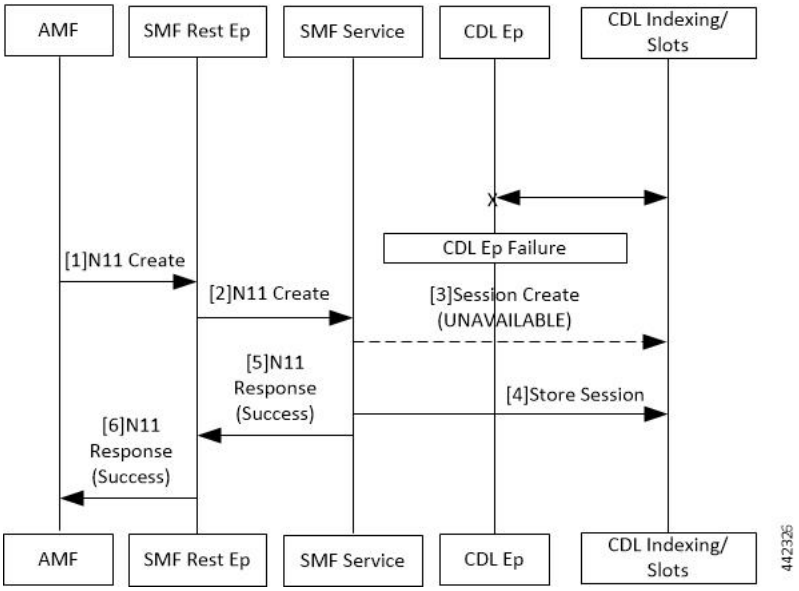



*Table 4: CDL Endpoint Failure Call Flow Description*

| Step | Description |
|---|---|
| 1 | The AMF sends a Create Request to SMF REST endpoint over the N11 interface. |
| 2 | After receiving the request, the SMF REST endpoint forwards the Create Request to the SMF service. |
| 3 | The SMF service attempts to reach the CDL endpoint to send the session creation request. However, the CDL endpoint is unreachable. |
| 4 | The Create Request is evaluated in the stored session and the SMF service forwards the request to the CDL endpoint. |
| 5 | After the call request is successful, the SMF service notifies the Success Message to the SMF REST endpoint. |
| 6 | The SMF REST endpoint forwards the Success Message to the AMF. |

# Limitations

The CDL configuration in SMF has the following limitations:

- The SMF service attempts to reroute the calls only when it encounters gRPC errors, such as UNAVAILABLE. It does not acknowledge errors that the datastore endpoint returns and actual gRPC timeouts, such as DEADLINE_EXCEEDED gRPC status code.
- The SMF service does not resolve failures occurring with the datastore such as indexing and slot failures. The CDL layer must resolve these failures and if necessary, send an API call on the remote.
- By default, the maximum retry limit for a request is three. After reaching this limit, the retry is stopped and the response is sent to the application.

- On receiving the response message as success during the retries, the further retry is stopped and the response is sent to the application.

# Configuring the CDL Through SMF Ops Center

The configuration of the CDL using SMF Ops Center involves the following steps:

1. Configuring the CDL Session Database and Defining the Base Configuration, on page 5
2. Configuring the Zookeeper in CDL, on page 6

# Configuring the CDL Session Database and Defining the Base Configuration

Use the following sample configuration to configure the CDL session database and define the base configuration in SMF:

```
config
   cdl system-id system_id
   cdl node-type node_type
   cdl zookeeper replica zookeeper_replica_id
   exit
   cdl logging default-log-level debug_level
   cdl datastore session
      cluster-id cluster_id
      endpoint replica 1
      endpoint replica num_replica
      index map map_value
      slot replica num_replica
      slot map num_map/shards
      slot write-factor write_factor
      slot notification host host
      slot notification port port
      slot notification limit tps
      index replica num_replica
      index map num_map/shards
      index write-factor write_factor
      slice-names cdl_slice_name
      end
```

**NOTES:**

- **cdl system-id** *system_id*: This is an optional command. Specifies the system or Kubernetes cluster identity. The default value is 1.
- **cdl node-type** *node_type*: This is an optional command. Specifies the Kubernetes node label to configure the node affinity. The default value is "session." Accepted length of the value is 0–64 alphabets.
- **cdl zookeeper replica** *zookeeper_replica_id*: Specifies the zookeeper replica server's ID.
- **endpoint replica** *num_replica*: This is an optional command. Specifies the number of replicas to be created. The default value is 1. Must be an integer in the range of 1–16.

- **slot replica** *num_replica*: This is an optional command. Specifies the number of replicas to be created. The default value is 1. num_replica must be an integer in the range of 1–16.
- **slot map** *num_map/shards*: This is an optional command. Specifies the number of partitions in a slot. The default value is 1. num_map/shards must be an integer in the range of 1–1024.
- **slot write-factor** *write_factor*: This is an optional command. Specifies the number of copies to be written before successful response. The default value is 1. write_factor must be an integer in the range of 0–16. Make sure that the value is lower than or equal to the number of replicas.
- **slot notification host** *host*: This is an optional command. Specifies the notification server hostname or IP address. The default value is datastore-notification-ep.
- **slot notification port** *port*: This is an optional command. Specifies the notification server Port number. The default value is 8890.
- **slot notification limit** *tps*: This is an optional command. Specifies the notification limit per second. The default value is 2000.
- **index replica** *num_replica*: This is an optional command. Specifies the number of replicas to be created. The default value is 2. num_replica must be an integer in the range of 1–16.
- **index map** *num_map/shards*: This is an optional command. Specifies the number of partitions in a slot. The default value is 1. num_map/shards must be an integer in the range of 1–1024. Avoid modifying this value after deploying the CDL.
- **index write-factor** *write_factor*: This is an optional command. Specifies the number of copies to be written before successful response. The default value is 1. write_factor must be an integer in the range of 0–16.
- **slice-names** *cdl_slice_name*: Specify the CDL slice names. *cdl_slice_name* must be an alphanumeric string from 1 to 16 characters in length.

# Configuring the Zookeeper in CDL

Use the following sample configuration to define the Zookeeper in CDL:

```
config
  cdl zookeeper data-storage-size data_storage_size_in_gb
    log-storage-size log_storage_size_in_gb
    replica number_of_replicas
    enable-JMX-metrics boolean_value
    enable-persistence boolean_value
    end
```

**NOTES:**

All the following parameters are optional.

- **cdl zookeeper data-storage-size** *data_storage_size_in_gb*: Specifies the size of the Zookeeper data storage in gigabyte. The default value is 20 GB. Accepted value is an integer in the range of 1-64.
- **log-storage-size** *log_storage_size_in_gb*: Specifies the size of the Zookeeper data log's storage in gigabyte. The default value is 20 GB. Accepted value is an integer in the range of 1-64.

- **replica** *num_replicas*: Specifies the number of replicas that must be created. The default value is 3. Accepted value is an integer in the range of one to 16.
- **enable-JMX-metrics** *boolean_value*: Specifies the status of the JMX metrics. The default value is true.
- **enable-persistence** *boolean_value*: Specifies the status of the persistent storage for Zookeeper data. The default value is *false*.

## Sample Configuration

This section shows a sample configuration of CDL in a HA environment.

```
config
cdl system-id system_id
cdl zookeeper replica num_zk_replica
cdl datastore session
 endpoint replica ep_replica
 index map index_shard_count
 slot replica slot_replica
 slot map slot_shard_count
 slice-names cdl_slice_name
exit
```

# Configuring Event Trace Data

This section describes how to configure the SMF to store event trace data in CDL database record. With this configuration, the SMF allows to enable or disable the storage of event trace data in the CDL database record. The event trace data shows the call flow event for the subscribers.

**Note** Configuring the event trace to disabled saves approximately 1 KB of database storage for each SMF database record.

To enable or disable the storage of event trace data in the CDL database record, use the following sample configuration:

```
config
   system-diagnostics event-trace [ enable | disable ]
      end
```

**NOTES:**

- **system-diagnostics event-trace [ enable | disable ]**: Enable or disable the storage of event trace data in the CDL database record for system diagnostics.

# Verifying Event Trace Data

This section describes how to verify the event trace data in SMF.

Use the show running-config command system-diagnostics event-trace CLI command to view if the event trace data is enabled or disabled.

The following is a sample output of the show running-config system-diagnostics event-trace CLI command.

```
show running-config system-diagnostics event-trace
system-diagnostics event-trace enabled
```